



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΣΗΜΑΤΩΝ, ΕΛΕΓΧΟΥ ΚΑΙ ΡΟΜΠΟΤΙΚΗΣ

Ανάλυση βιομετρικών δεικτών με χρήση
αυτο-επιβλεπόμενης μάθησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Ευάγγελου Φέκα

Επιβλέπων: Πέτρος Μαραγκός
Καθηγητής Ε.Μ.Π.

ΕΡΓΑΣΤΗΡΙΟ ΟΡΑΣΗΣ ΥΠΟΛΟΓΙΣΤΩΝ, ΕΠΙΚΟΙΝΩΝΙΑΣ ΛΟΓΟΥ ΚΑΙ ΕΠΕΞΕΡΓΑΣΙΑΣ ΣΗΜΑΤΩΝ
Αθήνα, Ιούλιος 2022



Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Σημάτων, Ελέγχου και Ρομποτικής
Εργαστήριο Όρασης Υπολογιστών, Επικοινωνίας Λόγου και Επεξεργασίας
Σημάτων

Ανάλυση βιομετρικών δεικτών με χρήση αυτο-επιβλεπόμενης μάθησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Ευάγγελου Φέκα

Επιβλέπων: Πέτρος Μαραγκός
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 14^η Ιουλίου, 2022.

.....
Πέτρος Μαραγκός
Καθηγητής Ε.Μ.Π.

.....
Αθανασιος Ροντογιάννης
Αναπληρωτής Καθηγητής Ε.Μ.Π.

.....
Γεράσιμος Ποταμιάνος
Αναπληρωτής Καθηγητής
Παν/μιο Θεσσαλίας

Αθήνα, Ιούλιος 2022

.....
ΕΥΑΓΓΕΛΟΣ ΦΕΚΑΣ
Διπλωματούχος Ηλεκτρολόγος Μηχανικός
και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © – All rights reserved Ευάγγελος Φέκας, 2022.
Με επιφύλαξη παντός δικαιώματος.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Περίληψη

Είναι γεγονός ότι η Τεχνητή Νοημοσύνη έχει γίνει κομμάτι της καθημερινότητας μας, βελτιώνοντας την ποιότητα ζωής σε πολλά επίπεδα. Από τη συνθήκη αυτή, δεν θα μπορούσε να λείπει ο τομέας της υγείας και ειδικότερα της ψυχικής, η οποία δοκιμάζεται καθημερινά από το σύγχρονο τρόπο ζωής. Από τις πιο επώδυνες για τον ασθενή ψυχικές ασθένειες είναι αυτές του φάσματος της σχιζοφρένειας, των οποίων οι αιτίες παραμένουν ασαφείς μέχρι σήμερα, πάρα την εκτενή έρευνα των τελευταίων ετών. Ευτυχώς, σημαντικά βήματα έχουν γίνει για τη μετάβαση από την μέχρι τώρα νοσοκομειοκεντρική προσέγγιση τέτοιων ασθενειών σε προληπτική, εξατομικευμένη και πραγματικού χρόνου. Ένας από τους τρόπους να συμβεί αυτό είναι με τη χρήση φορητών συσκευών (wearables) και αξιοποιώντας την παθητική καταγραφή βιομετρικών δεικτών που αυτά μας παρέχουν.

Δυστυχώς, η σύγχρονη τάση στην Μηχανική μάθηση (supervised deep-learning) απαιτεί τεράστιες ποσότητες προσεκτικά επισημειωμένων δεδομένων, κάτι το οποίο δεν μπορεί να γίνει εύκολα με τα wearables, αφού οι μετρήσεις τους συλλέγονται παθητικά. Πράγματι, για την εκτέλεση ενός απλού πειράματος (για παράδειγμα sleep-staging), χρειάζεται εξειδικευμένοι τεχνικοί να επισημειώσουν χειροκίνητα ώρες δεδομένων.

Μια πολλά υποσχόμενη λύση στο πρόβλημα αυτό, είναι ο αναδυόμενος τομέας της αυτο-επιβλεπόμενης μάθησης (SSL). Στο SSL, η δομή των δεδομένων χρησιμοποιείται για να μετατρέψει το unsupervised πρόβλημα σε supervised, το οποίο ονομάζεται ονομάζεται “pretext task”. Η αναπαράσταση που μαθαίνεται από το pretext task μπορεί στη συνέχεια να ξαναχρησιμοποιηθεί σε μια supervised εργασία, μειώνοντας δυνητικά τον απαιτούμενο αριθμό επισημειωμένων παραδειγμάτων.

Στην παρούσα μελέτη εφαρμόζεται η τεχνική της αυτο-επιβλεπόμενης μάθησης για την εξαγωγή αναπαραστάσεων από βιομετρικά δεδομένα, που συλλέχθηκαν από έξυπνα ρολόγια (smartwatches). Στη συνέχεια, οι αναπαραστάσεις αυτές χρησιμοποιούνται σε τέσσερις τελικές εργασίες και τα αποτελέσματα συγκρίνονται με αυτά άλλων μη-επιβλεπόμενων αλλά και πλήρως-επιβλεπόμενων τεχνικών. Οι τελικές εργασίες που μελετήσαμε είναι η ανίχνευση ύπνου, η ανίχνευση δραστηριότητας, η αναγνώριση χρήστη και πρόβλεψη χρονικού διαστήματος μέχρι την επόμενη υποτροπή.

Τα αποτελέσματα είναι ενθαρρυντικά, αφού στις πρώτες τρεις εργασίες η SSL αγγίζει την ακρίβεια πλήρως-επιβλεπόμενων τεχνικών μόνο με ένα κλάσμα των δεδομένων για προεκπαίδευση, ενώ εξαιρετική είναι και η απόδοση των τυχαίων συνελικτικών πυρήνων, ως βήμα εξαγωγής χαρακτηριστικών. Όμοια, στο πρόβλημα πρόβλεψης της υποτροπής ξεχώρισαν τα hand-crafted χαρακτηριστικά για την απόδοσή τους σε λίγα δεδομένα εκπαίδευσης και τη δυνατότητα ερμηνευσιμότητας που μας παρέχουν. Εξαιρετική είναι, επίσης, η ποιότητα των αναπαραστάσεων συνελικτικών πυρήνων και πολλά υποσχόμενες οι SSL τεχνικές, λόγω της ικανότητας να βελτιώνουν τις αναπαραστάσεις τους, όσο αυξάνουμε τα μη-επισημειωμένα δεδομένα προεκπαίδευσης.

Λέξεις Κλειδιά — Ψυχωτικές Διαταραχές, Survival Analysis, Αυτο-επιβλεπόμενη μάθηση, Transformers, Convolutional Neural Networks, Random Convolutional Kernels, Χρονοσειρές

Abstract

It is a fact that Artificial Intelligence (AI) has become part of our daily lives, improving the quality of life on many different levels. The health sector in particular could be also benefited from AI, especially regarding mental health, which is tested daily due to the modern way of life. One of the most infictive, for the patients, mental disorders are those belonging to the spectrum of schizophrenia, the causes of which remain until today unclear, despite the extensive research in recent years. Fortunately, important steps have been taken for the transformation of the hospital-centered healthcare practice to proactive, individualized care, through the use of wearables that can assist in collecting passive recording of biometric indices.

The current trend in supervised deep-learning at the moment requires huge amounts of carefully labeled data, something that unfortunately cannot be easily obtained when using wearables, since their measurements are collected passively. Indeed, in order to be able to perform a simple experiment (for example sleep-staging), specialized technicians are needed to manually annotate many hours of data.

A promising solution to this problem is the emerging field of self-supervised learning (SSL). In SSL, the data structure is used to convert the unsupervised problem to supervised, which is called a “pretext task”. The representation learned from the pretext task can then be reused in a supervised task, potentially reducing the required number of annotated examples.

In our study, the technique of self-supervised learning is applied to extract representations from biometric data, collected from smartwatches. These representations are then used in four different tasks and the results are compared with those of other non-supervised as well as fully-supervised techniques. The final tasks we studied are sleep detection, activity detection, user identification and prediction of the time until the next relapse of patients’ with mental disorder.

The results are encouraging; specifically, in the first three tasks, SSL approaches the accuracy of fully-supervised techniques using only a fraction of the data for pre-training, while the performance of random convolutional kernels is excellent as a feature extraction step. Similarly, in the problem of predicting relapses, handcrafted features yielded good results using only a few training data in addition to their provided interpretation ability. The quality of convolutional kernel representations is, again, great and the SSL techniques are also promising, due to their ability to enhance their representations as we increase unlabeled pre-training data.

Keywords — Psychotic Disorders, Survival Analysis, Self-Supervised Learning, Transformers, Convolutional Neural Networks, Random Convolutional Kernels, Time-series

Ευχαριστίες

Καταρχάς, θα ήθελα να ευχαριστήσω θερμά τον καθηγητή κ. Πέτρο Μαραγκό, για την εμπιστοσύνη που μου έδειξε με την ανάθεση αυτής της διπλωματικής εργασίας. Επίσης, θα ήθελα να ευχαριστήσω τη μεταδιδακτορική ερευνήτρια Δρ. Νάνσυ Ζλατίντση για τη συνεχή καθοδήγηση και τις ουσιαστικές διορθώσεις/προτάσεις και τον διδακτορικό φοιτητή Παναγιώτη Φιλντίση για όλη την προεργασία που έχει κάνει στα δεδομένα και στο στήσιμο των μηχανημάτων. Τέλος, θα ήθελα να ευχαριστήσω τους γονείς μου και τον αδερφό μου, που με στήριξαν και με βοήθησαν καθ' όλη την διάρκεια των φοιτητικών μου χρόνων.

Φέκας Ευάγγελος
Ιούλιος 2022

Περιεχόμενα

Περιεχόμενα	xī
Λίστα Σχημάτων	xiv
Κατάλογος Πινάκων	xviii
1 Εισαγωγή	1
1.1 Βιολογικά σήματα και Ψυχική Υγεία	2
1.1.1 Προϋποθέσεις εφαρμογής	2
1.2 Πρόβλημα	3
1.2.1 Περιγραφή του προβλήματος	3
1.2.2 Αντιμετώπιση του προβλήματος	4
1.3 Μηχανική μάθηση και Δεδομένα χρονοσειρών	4
1.4 Κίνητρα και Συνεισφορές	6
1.5 Διάρθρωση της Διπλωματικής εργασίας	6
2 Θεωρητικό Υπόβαθρο	9
2.1 Μηχανική μάθηση και no free lunch theorem	10
2.2 Αναπαράσταση δεδομένων σε χώρους πολλών διαστάσεων	10
2.2.1 Blessings of dimensionality	10
2.2.2 Curse of Dimensionality	11
2.3 The manifold hypothesis	14
2.4 Επέκταση των διαστάσεων της εισόδου	17
2.5 Νευρωνικά δίκτυα ως universal approximators	17
2.6 Ιεραρχική αναπαράσταση της εισόδου	18
2.7 Μετασχηματισμοί που συμβαίνουν σε κάθε στρώμα	20
3 Ιδιότητες των φυσικών σημάτων - Αρχιτεκτονικές Δικτύων	25
3.1 Ιδιότητες των φυσικών σημάτων	26
3.2 Ο πίνακας ενός πλήρως συνδεδεμένου δικτύου	27
3.3 Locality \Rightarrow sparsity	28
3.4 Stationarity \Rightarrow parameters sharing	29
3.4.1 Πίνακες Toeplitz	29
3.4.2 Στρώματα από Toeplitz πίνακες	29
3.5 Πλεονεκτήματα αρχιτεκτονικής	30
3.6 Σύνοψη CNN αρχιτεκτονικής	30
3.6.1 Pooling	31
3.6.2 Batch normalization και residual bypass connections	31
3.6.3 Padding	32
3.7 Transformers	32
3.7.1 Soft associative memories	32
3.7.2 Attention	33
3.7.3 Queries, Keys και Values	34
3.7.4 Multi-head attention	36

3.7.5	The Transformer	36
3.7.6	Encoder	37
3.7.7	Decoder	37
3.7.8	Positional encoding	39
3.7.9	Τα τελικά Linear και Softmax Layers	39
4	Self-supervised learning	41
4.1	Αυτο-επιβλεπόμενη μάθηση (Self-supervised Learning)	42
4.1.1	Μοντελοποιώντας την αβεβαιότητα πρόβλεψης	43
4.1.2	Ενοποιημένη οπτική στη χρήση αυτο-επιβλεπόμενης μάθησης	44
4.1.3	Αρχιτεκτονικές στο Self-Supervised Learning	45
4.1.4	Απλές επιλογές για pretext task	48
4.1.5	Contrastive energy-based SSL	50
4.1.6	Non-contrastive energy-based SSL	56
5	Περιγραφή μεθόδων	65
5.1	Εισαγωγή	66
5.2	Hand-Crafted Feature Engineering	66
5.3	Random convolutional kernels	66
5.4	InceptionTime	67
5.4.1	Inception modules	67
5.4.2	Inception network	68
5.4.3	InceptionTime: a neural network ensemble for TSC	68
5.5	Self-supervised time series representation learning by inter-intra relational reasoning	69
5.5.1	Μέθοδος	70
5.6	A Transformer-based Framework for Multivariate Time Series Representation Learning	72
5.6.1	Πλεονεκτήματα των transformers έναντι άλλων αρχιτεκτονικών	73
5.6.2	Μεθοδολογία	73
5.6.3	Πειράματα & Αποτελέσματα των Zerveas et al.	79
5.7	Encoding Time Series as Images for Visual Inspection and Classification	80
5.7.1	Gramian Angular Field	81
5.7.2	Markov Transition Field	83
5.7.3	Σύγκριση και ανάλυση	84
5.8	Survival Analysis	85
5.8.1	Εισαγωγή	85
5.8.2	Ορισμοί	87
5.8.3	Εφαρμογές του survival analysis	87
6	Πειράματα και Αποτελέσματα	95
6.1	Εισαγωγή	96
6.1.1	Περιγραφή δεδομένων	96
6.1.2	Επιλογή μεθοδολογίας	96
6.2	Περιγραφή Πειραμάτων	97
6.3	Προβλήματα κατηγοριοποίησης	97
6.3.1	Μεθοδολογία	98
6.3.2	Αποτελέσματα	100
6.3.3	Αυξάνοντας σταδιακά τη διαθεσιμότητα των labels	101
6.3.4	Αυξάνοντας σταδιακά τη διαθεσιμότητά των unlabeled δεδομένων για pretrain	101
6.3.5	Αποτελέσματα όταν έχουμε fully-labeled dataset	103
6.4	Survival Analysis	104
6.4.1	Μεθοδολογία	108
6.4.2	Αποτελέσματα και σχολιασμός	111
7	Επίλογος	115
7.1	Σύνοψη και Συμπεράσματα	115
7.2	Μελλοντικές Επεκτάσεις	116

Λίστα Σχημάτων

1.3.1	Mean rank διαφόρων classifiers στα 85 “bake off” datasets του UCR archive [Dau+19], το οποίο είναι το πιο ευρέως χρησιμοποιούμενο “benchmark dataset” σε δεδομένα χρονοσειρών.	5
2.2.1	Ποσοστό διαχωρίσιμων σημείων [SB11]	11
2.2.2	Ο αριθμός των περιοχών ενός κανονικού πλέγματος μεγαλώνει εκθετικά με τη διάσταση d του χώρου [Bis06].	12
2.2.3	Μοναδιαίος υπερκύβος και κύβος ακμής μήκους ℓ που περιέχει τους 10 κοντινότερους γείτονες [Wei21].	12
2.2.4	Ιστογράμματα εμφανίσεων pairwise-αποστάσεων για κάθε d . Βλέπουμε ότι όσο οι διαστάσεις αυξάνονται: α) Δεν υπάρχουν γείτονες (σχεδόν κανένα σημείο απόσταση 0). β) Όλα τα σημεία έχουν ίδια απόσταση ίση με τη μέγιστη απόσταση μεταξύ των κορυφών του κύβου [Wei21].	13
2.2.5	Η απόσταση μεταξύ σημείων αυξάνεται αφού τα σημεία αυτά απέχουν επιπλέον Δz ως προς τον άξονα z . Η απόστασή τους ως προς το επίπεδο παραμένει σταθερή [Wei21].	14
2.2.6	Αριστερά: Τυχαία δειγματοληπτημένα σημεία στο διδιάστατο επίπεδο. Δεξιά: Προσθήκη μίας τρίτης διάστασης με τυχαίες συντεταγμένες. Βλέπουμε ότι η απόσταση μεταξύ των σημείων αυξάνεται, ενώ η απόστασή τους από το υπερεπίπεδο παραμένει σταθερή [Wei21].	14
2.3.1	Αριστερά: βλέπουμε σημεία από μια κατανομή σε έναν διδιάστατο χώρο που είναι στην πραγματικότητα συγκεντρωμένα ένα μονοδιάστατο manifold, που προσομοιάζει μία χορδή. Η συμπαγής γραμμή δείχνει το manifold που πρέπει να μάθει ο αλγόριθμος [GBC16]. Δεξιά: φαίνεται το ίδιο για 3-διάστατο χώρο και 2-διάστατο manifold [Wei21].	15
2.3.2	Δειγματοληψία εικόνων ομοιόμορφα τυχαία (επιλέγοντας τυχαία κάθε pixel σύμφωνα με ομοιόμορφη κατανομή) δημιουργεί θορυβώδεις εικόνες [GBC16].	16
2.3.3	Παραδείγματα δειγμάτων εκπαίδευσης από το QMUL Multiview Face Dataset [GMP00] για το οποίο ζητήθηκε από τους συμμετέχοντες να κινηθούν με τέτοιο τρόπο ώστε να διασχίσουν το διδιάστατο manifold που αντιστοιχεί σε δύο γωνίες περιστροφής.	16
2.5.1	Κάθε hidden unit καθορίζει πού θα διπλωθεί ο χώρος. Συνδυάζοντάς τέτοιους μετασχηματισμούς έχουμε εκθετικά μεγάλο αριθμό από piecewise linear περιοχές [GBC16].	18
2.5.2	Εμπειρικά αποτελέσματα που δείχνουν ότι τα βαθύτερα δίκτυα γενικεύουν καλύτερα. Το συγκεκριμένο πείραμα αφορά αναγνώριση πολυψήφων αριθμών από φωτογραφίες διευθύνσεων [Goo+13]. Η ακρίβεια του συνόλου δοκιμής αυξάνεται σταθερά με την αύξηση του βάθους. Επίσης το Σχήμα 2.5.3 δείχνει ότι η αύξηση στο μέγεθος του μοντέλου δεν έχει το ίδιο αποτέλεσμα.	19
2.5.3	Τα πιο βαθιά μοντέλα τείνουν να έχουν καλύτερη απόδοση. Αυτό δεν συμβαίνει μόνο επειδή το μοντέλο είναι μεγαλύτερο. Αυτό το πείραμα από τους Goodfellow, I. J. et al. [Goo+13] δείχνει ότι η αύξηση του αριθμού των παραμέτρων σε στρώματα συνελικτικών δικτύων χωρίς αύξηση του βάθους τους δεν έχει αποτέλεσμα στην αύξηση της απόδοσης του συνόλου δοκιμής. Στη λεζάντα γράφεται το βάθος του δικτύου που χρησιμοποιείται για τη δημιουργία κάθε καμπύλης και αν είναι συνελικτικό ή τα πλήρως συνδεδεμένο δίκτυο. Παρατηρούμε ότι ρηχά μοντέλα κάνουν overfit στις περίπου 20 εκατομμύρια παραμέτρους, ενώ τα βαθιά μπορούν να επωφεληθούν από την ύπαρξη πάνω από 60 εκατομμύρια παραμέτρων.	19
2.6.1	Ιεραρχική δομή που μαθαίνει το δίκτυο και ταιριάζει με τη διαίσθησή μας για το από τι αποτελείται μία εικόνα [OMS17].	19
2.7.1	Μπλοκ Διάγραμμα ενός απλού νευρωνικού δικτύου	20

2.7.2	Στο 2.7.2a φαίνονται τα αρχικά σημεία. Στη συνέχεια εφαρμόζονται με τη σειρά οι μετασχηματισμοί με τους αντίστοιχους πίνακες: $[2 \ 00 \ 2]$, $[2 \ 00 \ 0.5]$, $[-1 \ 00 \ 1]$, $[\cos(45^\circ) \ -\sin(45^\circ)\sin(45^\circ) \ \cos(45^\circ)]$, $[1 \ 0.50 \ 1]$, $[1 \ 0 \ 20 \ 1 \ 10 \ 0 \ 1]$ Φαίνονται επίσης τα διανύσματα βάσης πριν και μετά το μετασχηματισμό	23
2.7.3	Οπτικοποίηση ενός νευρωνικού δικτύου με δύο νευρώνες εισόδου, ένα κρυφό στρώμα με 3 νευρώνες και ReLU συνάρτηση ενεργοποίησης και 2 νευρώνες εξόδου.	24
3.3.1	Πληρως συνδεδεμένο δίκτυο	28
3.3.2	Αριστερά: Βλέπουμε ένα πλήρως-συνδεδεμένο δίκτυο. Δεξιά: Βλέπουμε το ίδιο δίκτυο αν εκμεταλλευτούμε το locality. Φαίνονται επίσης και τα receptive field (RF) των νευρώνων.	29
3.4.1	Αριστερά: Αμέσως μετά την εφαρμογή του sparsity Δεξιά: Αμέσως μετά την εφαρμογή του Parameter Sharing.	30
3.5.1	Αριστερά: Εφαρμογή kernels σε 1D δεδομένα. Δεξιά: Το ίδιο με Zero Padding.	30
3.6.1	Οπτικοποίηση του Pooling operator.	31
3.6.2	Στο σχήμα βλέπουμε ότι η μετατρέπεται από χωρική, δηλαδή διάσπαρτη ως προς το πλάτος και μήκος της εικόνας, γίνεται μία πυκνή αναπαράσταση από features.	32
3.7.1	Soft associative memory	33
3.7.2	Αριστερά: Scaled Dot-Product Attention, Δεξιά: Multi-Head Attention	35
3.7.3	Στοιβες από Encoders - decoders	37
3.7.4	Κάθε λέξη (σειρά) επιτρέπεται να κάνει attend σε λέξεις (στήλες) που ανήκουν στο παρελθόν αυτής.	37
3.7.5	Αριστερά: Encoders, Δεξιά: Decoders.	38
3.7.6	Οπτικοποίηση των Positional embeddings.	39
4.1.1	Στην αυτο-επιβλεπόμενη μάθηση, το σύστημα εκπαιδεύεται για να προβλέψει κρυμμένα μέρη της εισόδου (σε γκρι) από ορατά μέρη της εισόδου (με πράσινο χρώμα).	42
4.1.2	Οπτικοποίηση μοντέλων που χρησιμοποιούνται στην πράξη στους άξονες αβεβαιότητας, διακριτότητας και μεγέθους διάστασης των δεδομένων εισόδου	44
4.1.3	Ένα μοντέλο που βασίζεται στην ενέργεια (EBM) μετρά τη συμβατότητα μεταξύ μιας παρατήρησης x και μιας προτεινόμενης πρόβλεψης y . Εάν τα x και y είναι συμβατά, η ενέργεια είναι ένας μικρός αριθμός. εάν είναι ασύμβατα, η ενέργεια είναι μεγαλύτερος αριθμός.	45
4.1.4	Joint embedding architecture. Η συνάρτηση C στην κορυφή παράγει ένα βαθμωτό μέγεθος (ενέργεια) που μετρά την απόσταση μεταξύ των διανυσμάτων αναπαράστασης (embeddings) που παράγονται από δύο πανομοιότυπα δίδυμα δίκτυα που μοιράζονται τις ίδιες παραμέτρους (w). Όταν τα x και y είναι ελαφρώς διαφορετικές εκδόσεις της ίδιας εικόνας, το σύστημα έχει εκπαιδευτεί να παράγει χαμηλή ενέργεια, γεγονός που αναγκάζει το μοντέλο να παράγει παρόμοια διανύσματα ενσωμάτωσης για τις δύο εικόνες. Το δύσκολο μέρος είναι να εκπαιδύσουμε το μοντέλο έτσι ώστε να παράγει υψηλή ενέργεια (δηλαδή, διαφορετικές ενσωματώσεις) για εικόνες που είναι διαφορετικές.	46
4.1.5	Με τη σειρά: Rotation, Relative Position, Jigsaw, Square Crop, Colorization	49
4.1.6	Αριστερά: Καθώς αυξάνουμε τον αριθμό των permutations το mAP αυξάνει και στη συνέχεια πέφτει. Δεξιά: Καθώς κάνουμε probing σε πιο βαθιά layers του ResNet50 το mAP αυξάνει μέχρι ένα σημείο και έπειτα πέφτει.	50
4.1.7	Η ενέργεια είναι χαμηλή για τα συμβατά ζεύγη (μαύρες κουκκίδες), ενώ υψηλή για μη συμβατά x, y	51
4.1.8	Ένα masked language model, είναι ένα instance των denoising auto-encoders, Η μεταβλητή y είναι ένα κομμάτι κειμένου. Το x είναι το ίδιο κείμενο με κάποιες από τις λέξεις να είναι masked. Το δίκτυο εκπαιδεύεται για να κάνει ανακατασκευή το κείμενο.	52
4.1.9	Μια latent-variable predictive αρχιτεκτονική. Με δεδομένη μια παρατήρηση x , το μοντέλο πρέπει να μπορεί να παράγει ένα σύνολο πολλαπλών συμβατών προβλέψεων που συμβολίζονται από το σχήμα S στο διάγραμμα. Καθώς η latent μεταβλητή z ποικίλλει μέσα σε ένα σύνολο, που συμβολίζεται με ένα γκρι τετράγωνο, η έξοδος ποικίλλει σε σχέση με το σύνολο των προβλέψεων που θέλουμε.	52
4.1.10	53
4.1.11	Πάνω: Block διάγραμμα Κάτω: Διαισθητική απεικόνιση στο manifold των δεδομένων.	54
4.1.12	End-to-end vs Memory bank vs Momentum update.	55
4.1.13	Αριστερά: BYOL Δεξιά: SimSiam	57
4.1.14	Οπτικοποίηση της μεθόδου Barlow Twins.	58

4.1.15	Αριστερά: Contrastive learning στο feature space. Π.χ. τα μπλε σχήματα είναι μία εικόνα και οι μετασχηματισμοί (augmentations) της. Δεξιά: Μπορούμε να πετύχουμε το ίδιο αποτέλεσμα με clustering.	59
4.1.16	Οπτικοποίηση της μεθόδου SwAV.	59
4.1.17	Ελαχιστοποίηση της χωρητικότητας της latent μεταβλητής μέσω της προσθήκης Gaussian θορύβου. Ο όρος $k\ z - \tilde{z}\ ^2$ μπορεί να θεωρηθεί σαν log μίας prior από όπου κάνουμε sample την z . Η έξοδος του encoder γίνεται regularized, ώστε να έχει mean = 0.	60
4.1.18	Πάνω: Block διάγραμμα Κάτω: Διαισθητική απεικόνιση στο manifold των δεδομένων.	61
4.1.19	Διαισθητική απεικόνιση του regularization στον VAE.	63
5.4.1	Inception module. Για απλότητα παρουσιάζεται από τους Fawaz et al. bottleneck layer μεγέθους $m = 1$	68
5.4.2	Ολόκληρο το Inception network.	68
5.5.1	Αριστερά: Οπτικοποίηση του inter-sample relation. Δίνεται ο anchor, ο μετασχηματισμός του (positive sample) και ο μετασχηματισμός ενός άλλου sample (negative sample) Δεξιά: Οπτικοποίηση του multi-scale intra-temporal relation. Εδώ έχουμε 3-scale temporal relations δηλαδή short-term, middle-term και long-term.	70
5.5.2	Αρχιτεκτονική του SelfTime.	70
5.5.3	Η μπλε γραμμή είναι το αρχικό σήμα και η κόκκινη το μετασχηματισμένο.	72
5.6.1	Αριστερά: Το διάνυσμα χαρακτηριστικών \mathbf{x}_t σε κάθε χρονική στιγμή t προβάλλεται σε διάνυσμα \mathbf{u}_t ίδιας διάστασης d . Στο τελευταίο προστίθενται τα positional encodings και αυτή αποτελεί την τελική αναπαράσταση στο πρώτο self-attention στρώμα για να μετατραπεί σε keys, queries και values. Δεξιά: Training setup of the unsupervised pre-training task. Κάνουμε mask ένα τμήμα r κάθε μεταβλητής της χρονοσειράς ανεξάρτητα, έτσι ώστε να έχουμε τμήματα μέσου μήκους l_m masked, ακολουθούμενα από unmasked κομμάτια μέσου μήκους $l_u = \frac{1-r}{r}l_m$. Στη συνέχεια χρησιμοποιούμε ένα linear layer με είσοδο τις διανυσματικές αναπαραστάσεις \mathbf{z}_t , σε κάθε στιγμή και ως labels τα uncorrupted input vectors \mathbf{x}_t . Τέλος υπολογίζεται το Μέσο Τετραγωνικό Σφάλμα μετρώντας μόνο τις masked τιμές.	74
5.6.2	Οπτικοποίηση διαφόρων σχημάτων κατασκευής μάσκας. Στην κορυφή (5.6.2a) φαίνεται η μάσκα με τις default τιμές $r = 0.15$, $l_m = 3$, stateful=True, sync=False. Στη συνέχεια σε κάθε σχήμα αλλάζουμε μία από τις μεταβλητές και κρατάμε τις υπόλοιπες στις default για να δούμε τις διαφορές. Μεγαλώνοντας το l_m (Σχήμα 5.6.2b) βλέπουμε ότι μεγαλώνει το μέσο μήκος που γίνεται masked δηλαδή τα μαύρα κομμάτια. Στη συνέχεια θέτοντας το sync=True (Σχήμα 5.6.2c) βλέπουμε ότι γίνετονται masked όλες οι μεταβλητές της χρονοσειράς σύγχρονα. Στη συνέχεια αλλάζοντας το Stateful σε True (Σχήμα 5.6.2d) βλέπουμε ότι πλέον έχουμε κατανομή bernoulli και άρα δεν πετυχαίνουμε τόσο συχνά συνεχόμενες ακολουθίες στη μάσκα για δεδομένο r . Τέλος μεγαλώνοντας το r σε 0.5 βλέπουμε ότι αλλάζει το γενικό κλάσμα masked σημείων σε όλη τη χρονοσειρά και όλες τις μεταβλητές χωρίς να αλλάζει η κατανομή του μήκους. Επίσης στον τίτλο του κάθε σχήματος φαίνεται το sample mean masked σημείων για κάθε δείγμα μάσκας. Τα σχήματα δημιουργήθηκαν χρησιμοποιώντας τη βιβλιοθήκη [Ogu20].	77
5.6.3	Πάνω: Imputation of missing values στο test set του BenzeneConcentration dataset. Η συνεχόμενη μπλε γραμμή είναι το ground truth σήμα, οι ανοιχτοί μπλε κύκλοι είναι οι τιμές που έγιναν mask και με πορτοκαλί οι προβλέψεις του συστήματος. Βλέπουμε ότι το μοντέλο πηγαίνει πολύ καλά ακόμη και σε απότομες αλλαγές ή όταν λείπουν πολλές συνεχόμενες τιμές. Κάτω: Το ίδιο για 5 διαφορετικές μεταβλητές της χρονοσειράς.	78
5.6.4	Σχήμα μάσκας για forecasting (αριστερά), sliding window (δεξιά).	79
5.6.5	Αριστερά: MSE στο test set πλήρως supervised μοντέλου (πορτοκαλί κύκλοι) και του ίδιου προεκπαιδευμένου μοντέλου (μπλε διαμάντια), καθώς αυξάνουμε το ποσοστό στο οποίο κάνουμε supervised learning μετά την προεκπαίδευση. Δεξιά: MSE στο test set ενός δεδομένου μοντέλου ως συνάρτηση του αριθμού των δειγμάτων που χρησιμοποιούνται για προεκπαίδευση χωρίς επίβλεψη. Για την εποπτευόμενη μάθηση που ακολουθεί, απεικονίζονται δύο επίπεδα διαθεσιμότητας ετικετών: 10% (μαβ κύκλοι) και 20% (πράσινα τετράγωνα). Όταν ο οριζόντιος άξονας είναι 0 έχουμε μόνο supervised εκμάθηση, ενώ όλες οι άλλες τιμές αντιστοιχούν σε προεκπαίδευση ακολουθούμενη από finetuning.	81

5.7.1	Διαδικασία κατασκευής του Gramian Angular Fields. Το X είναι η rescaled χρονοσειρά. Στη συνέχεια μετατρέπουμε το rescaled X σε πολικές συντεταγμένες μέσω της Εξ. (5.7.3) και υπολογίζουμε τις GASF/GADF εικόνες από τις Εξ. (5.7.5) και (5.7.7). Στη συνέχεια κάνουμε PAA smoothing, όμως στο παράδειγμα εδώ δεν έγινε για να έχουμε υψηλή ανάλυση.	82
5.7.2	Οπτικοποίηση κατασκευής του Markov Transition Fields. Το X είναι μία χρονοσειρά που γίνεται discretized σε Q quantile bins. Στη συνέχεια υπολογίζουμε τον Markov Transition Matrix W και έπειτα το MTF μέσω της Εξ. (5.7.9).	84
5.7.3	Αρχικό GASF \rightarrow “broken” GASF \rightarrow recovered GASF (πάνω). Αρχική χρονοσειρά \rightarrow corrupted χρονοσειρά με missing values \rightarrow predicted χρονοσειρά (κάτω).	85
5.8.1	Οπτικοποίηση του censoring. Σχήμα από [Pö120]	86
5.8.2	Σύγκριση Uno’s c με Harrel’s c . Βλέπουμε ότι ο εκτιμητής του Uno είναι σταθερός ενώ εκείνος του Harrel υπερεκτιμά το c για μεγάλο ποσοστό του censoring.	91
5.8.3	Τυπική καμπύλη ROC.	92
6.3.1	Οπτικοποίηση standardized δεδομένων και οι αντίστοιχες ετικέτες για το Sleep task . Σειρές: Αξελερόμετρο, Γυροσκόπιο, Καρδιακοί παλμοί. Στήλες: Sleeping, awake, transition. Βλέπουμε ότι τα πλάτη είναι μεγαλύτερα όταν ο χρήστης είναι awake, σε σύγκριση με όταν κοιμάται, ενώ όταν ο χρήστης είναι σε transition βλέπουμε κάτι ενδιάμεσο.	98
6.3.2	Οπτικοποίηση της κατανομής των κατηγοριών για κάθε τελική εργασία. Προφανώς το transition έχει πολύ λίγα δείγματα αφού πρέπει να τύχει μέσα σε συγκεκριμένο διάστημα ενός λεπτού ο χρήστης να μεταβεί από μια κατάσταση σε μία άλλη.	99
6.3.3	Boxplots που προκύπτουν από όλα τα πειράματα που τρέξαμε. Στον άξονα y απεικονίζεται το accuracy. Για κάθε ζευγάρι μεταβλητών αναγράφεται αν το αποτέλεσμα που φαίνεται στο σχήμα είναι στατιστικά σημαντικό σύμφωνα με το <code>ttest_ind</code>	100
6.3.4	Αριστερά: Αυξάνοντας σταδιακά τη διαθεσιμότητα των labels. Δεξιά: Αυξάνοντας σταδιακά τη διαθεσιμότητα των unlabeled δεδομένων για pretrain.	102
6.3.5	Confusion Matrix του καλύτερου μοντέλου (MiniRocket) για τα 3 tasks. Sleep: 0:awake, 1:sleeping, 2:transition, Step: 0:walking, 1:not-walking, 2:transition.	104
6.3.6	Αποτελέσματα όταν έχουμε όλα τα διαθέσιμα labels. Στον άξονα x φαίνεται σε ποιο global step έκανε early stopping ο γραμμικός ταξινομητής και στον άξονα y το αντίστοιχο accuracy.	105
6.3.7	Εφαρμογή 4 τεχνικών dimensionality reduction, συγκεκριμένα PCA, t-SNE, UMAP και densMAP στα embeddings του καλύτερου μοντέλου (SelfTime) στα step και id tasks.	106
6.3.8	Εφαρμογή 4 τεχνικών dimensionality reduction, συγκεκριμένα PCA, t-SNE, UMAP και densMAP στα embeddings του καλύτερου μοντέλου για το sleep task (MiniRocket).	107
6.4.1	Έχουμε συνολικά 1909 samples/κουκκίδες και 37 υποτροπές (μπλε διαστήματα). Με πορτοκαλί χρώμα φαίνονται τα samples όπου έχουν κάποια υποτροπή στα δεξιά τους, ενώ με πράσινο εκείνα που βρίσκονται δεξιά από κάθε υποτροπή του χρήστη.	108
6.4.2	(a) Οπτικοποίηση ενός sample του dataset, δηλαδή 12 ώρες συνεχόμενων μετρήσεων από τους τρεις sensors και τα cyclically-encoded χαρακτηριστικά της ώρας. (b) Ιστόγραμμα των time-to-event χρόνων για censored/non-censored δείγματα.	110
6.4.3	Αποτελέσματα 5-fold-cross-validation, ως προς IPCW-AUC (higher is better) και Brier (lower is better) μετρικές, για κάθε μοντέλο και κάθε embedding. Στην πάνω σειρά φαίνονται οι μέσοι όροι των μετρικών για τις 5 επαναλήψεις, ενώ στην κάτω σειρά οι αντίστοιχες τυπικές αποκλίσεις.	112
6.4.4	Feature Importances για τα embeddings του TSFresh, για το μοντέλο Random S-Forest μέσω αλγορίθμου Permutation Importance [Bre01].	113

Κατάλογος Πινάκων

2.1	Αντιστοιχία d με ℓ [Wei21].	12
2.2	Μορφές που μπορεί να πάρει ο πίνακας ενός αφινικού μετασχηματισμού και οπτικοποίησή τους.	21
5.1	Σύγκριση layer normalization και batch normalization (Batch size=128).	75
5.2	Σύγκριση επιδόσεων μεταξύ fine-tuned και frozen (όλων των layers εκτός του τελευταίου). Οι χρόνοι αφορούν: per-epoch training time σε GPU.	79
5.3	Απόδοση σε multivariate regression , μετρική: Root Mean Squared Error. Bold σημαίνει οι καλύτερες τιμές και υπογραμμισμένες σημαίνει δεύτερες καλύτερες.	80
6.1	Accuracy για κάθε task, κάθε sensor και κάθε μοντέλο. Επίσης φαίνονται με bold το ελάχιστο και μέγιστο accuracy για κάθε sensor και task. Επίσης βλέπουμε με πράσινο χρώμα το μέγιστο accuracy για κάθε task και με κόκκινο χρώμα το ελάχιστο. Στην περίπτωση του step task έχουμε ισοβαθμία στο μέγιστο.	100
6.2	IPCW-AUC (higher is better) και IBS (lower is better) μετρικές για κάθε μοντέλο και κάθε embedding. Επίσης, φαίνεται με bold το καλύτερο embedding για κάθε συνδυασμό μετρικής/τελικού survival μοντέλου. Επίσης, βλέπουμε με πράσινο χρώμα τον καλύτερο συνδυασμό survival-μοντέλου/embedding-εισόδου, ως προς κάθε μετρική και με κόκκινο χρώμα τον χειρότερο. Στην περίπτωση του step task έχουμε ισοβαθμία στο μέγιστο.	111

Κεφάλαιο 1

Εισαγωγή

1.1	Βιολογικά σήματα και Ψυχική Υγεία	2
1.1.1	Προϋποθέσεις εφαρμογής	2
1.2	Πρόβλημα	3
1.2.1	Περιγραφή του προβλήματος	3
1.2.2	Αντιμετώπιση του προβλήματος	4
1.3	Μηχανική μάθηση και Δεδομένα χρονοσειρών	4
1.4	Κίνητρα και Συνεισφορές	6
1.5	Διάρθρωση της Διπλωματικής εργασίας	6

1.1 Βιολογικά σήματα και Ψυχική Υγεία

Τεράστια πρόοδος έχει σημειωθεί τα τελευταία χρόνια στον τομέα των φορητών συσκευών που φοριούνται στο χέρι (wearables), όπως έξυπνα ρολόγια (smartwatches) και fitness-trackers. Πλέον, μπορούμε να έχουμε αξιόπιστη, διακριτική, απομακρυσμένη και εξατομικευμένη συλλογή βιομετρικών σημάτων μέσω των αισθητήρων τέτοιων συσκευών [Pat+12; BML15]. Η εκμετάλλευση τέτοιων σημάτων για την παρακολούθηση της υγείας ενός ατόμου, μπορεί να αποτελέσει ένα πρώτο βήμα για τη μετάβαση από την τωρινή νοσοκομειοκεντρική πρακτική υγειονομικής περίθαλψης σε προληπτική, εξατομικευμένη και πραγματικού-χρόνου. Πράγματι, η χρήση τέτοιων σημάτων σε πολλά πεδία της ιατρικής είναι πραγματικότητα, ενώ συνεχώς αυξανόμενη είναι βιβλιογραφία στον τομέα της ψυχικής υγείας [Tor+14; EH15; Tor+15; Sae+15]. Πιο συγκεκριμένα, ο όρος ‘ψηφιακός φαινοτυπισμός’ (digital phenotyping) [Tor+16] διευθετήθηκε για να περιγράψει ακριβώς αυτό, δηλαδή την εξατομικευμένη, παθητική συλλογή δεδομένων από ψηφιακές συσκευές με σκοπό την εκτίμηση της ψυχικής κατάστασης του χρήστη στιγμή προς στιγμή. Τέτοια δεδομένα μπορεί να είναι: συντεταγμένες GPS, σχόλια σε κοινωνικά δίκτυα, βιομετρικά σήματα από τους αισθητήρες των wearables κ.α.. Οι δείκτες που αποκτώνται από αυτά τα σήματα, αξιοποιούνται ήδη στη γενική ιατρική και έχει γίνει και η εισαγωγή τους στην κλινική ψυχιατρική [AMC17].

Οι ψυχικές διαταραχές επιβαρύνουν τους πάσχοντες με φορτίο που είναι δύσκολο να μετρηθεί. Μία εκτίμηση αυτού του φορτίου, είναι ο δείκτης Years of healthy Life lost due to Disability (YLD), ο οποίος προτείνεται από τον παγκόσμιο οργανισμό υγείας. Όπως ορίζει και το όνομά του, ένα YLD ισοδυναμεί με ένα έτος υγιούς ζωής που χάνεται λόγω κάποιας ασθένειας. Η συνολική εκτιμώμενη επιβάρυνση που σχετίζεται με ψυχικές διαταραχές αντιστοιχεί στο 32,4% όλων των YLDs από όλες τις ασθένειες συνολικά [VTA16]. Μία από τις πιο σοβαρές, χρόνιες ψυχικές διαταραχές είναι η διπολική διαταραχή (Bipolar disorder - BD), η οποία έχει επιπολασμό 0,6% [SR21]. Οι ασθενείς που έχουν διαγνωστεί με BD παρουσιάζουν ακραίες εναλλαγές διάθεσης και διακυμάνσεις από υπερκινητικότητα σε πλήρη αδράνεια. Οι ασθενείς με BD υποφέρουν συχνά από δυσκολίες στον ύπνο και μπορεί να δυσκολεύονται ακόμη και στις καθημερινές τους εργασίες [Man+07]. Η σχιζοφρένεια είναι, επίσης, μια σοβαρή χρόνια ψυχική διαταραχή με αναφερόμενο επιπολασμό 0,3% [SR21]. Η διαταραχή προκαλεί παραισθήσεις, ψευδαισθήσεις, αλλαγές στη βούληση και τη νευρογνώσια και συναισθηματική δυσρύθμιση [SKK18]. Οι ασθενείς με σχιζοφρένεια αντιμετωπίζουν, επίσης, παράπλευρες προκλήσεις στη ζωή τους, όπως το χαμηλό ποσοστό απασχόλησης (κάτω από 20%) και υψηλό ποσοστό έλλειψης στέγης (έως 20%) [MBR05]. Τέλος, η BD και η σχιζοφρένεια, συχνά ερευνώνται μαζί λόγω των ομοιοτήτων τους στα συμπτώματα [BH06].

Δυστυχώς, παρά την εκτεταμένη έρευνα τα τελευταία 60 χρόνια στη νευροβιολογία και τη νευροφυσιολογία των ψυχωσικών διαταραχών, η αιτία τους παραμένει ασαφής και αξιόπιστοι βιομετρικοί δείκτες για τη διάγνωση και πρόβλεψη της πορείας της ψυχωτικής συμπτωματολογίας δεν έχουν βρεθεί ακόμη. Η χρήση βιομετρικών σημάτων για την ανίχνευση και έγκαιρη διάγνωση και πρόληψη ψυχωσικών υποτροπών είναι πλέον μία από τις σημαντικότερες ερευνητικές περιοχές στην ψυχιατρική [Kou+12; McG+14].

1.1.1 Προϋποθέσεις εφαρμογής

Ένα πρόβλημά που προκύπτει, λόγω του τεράστιου όγκου δεδομένων που παράγεται από τα wearables, είναι ότι δεν είναι όλα σχετικά με την κλινική κατάσταση του ασθενή [FW22]. Γι’ αυτό το λόγο, συνήθως χρησιμοποιούνται αλγόριθμοι μηχανικής μάθησης που εκπαιδεύονται να αναγνωρίζουν πρότυπα μέσα στα δεδομένα αυτά και να τα συσχετίσουν με διάφορες κλινικές καταστάσεις, όπως η έναρξη ενός μανιακού ή καταθλιπτικού επεισοδίου. Επειδή όμως οι αλγόριθμοι αυτοί έχουν περιθώριο σφάλματος και οι αποφάσεις τους αφορούν την υγεία των ατόμων, πρέπει πάντοτε να διασφαλίζουμε ότι ισχύουν κάποιες βασικές προϋποθέσεις.

Ηθικές προϋποθέσεις

Λαμβάνοντας υπόψη την ευαλωτότητα των ατόμων με ψυχικές διαταραχές, σε συνδυασμό με το γεγονός ότι τα δεδομένα που λαμβάνονται είναι άκρως προσωπικά, είναι απαραίτητο να δοθεί προτεραιότητα στις ηθικές αρχές σε όλες τις σχετικές έρευνες [TK18; Neb+16]. Οι Chivilgina et al. [CEJ21] επισημαίνουν τις πιο επείγουσες ανησυχίες, όπως για παράδειγμα την εμπιστευτικότητα των δεδομένων και την ύπαρξη σαφών προτύπων ασφαλείας. Επισημαίνουν, επίσης, ότι έχουμε ανεπαρκή στοιχεία σχετικά με τον αντίκτυπο της τεχνολογίας των wearables στην κατάσταση των ασθενών, καθώς επίσης και της αντίληψής των ασθενών για τα wearables. Η θέσπιση δεοντολογικών κατευθυντήριων γραμμών για την συνεχή, διακριτική, παθητική συλλογή δεδομένων που εκτελείται από τα wearables, αποτελεί βασική προϋπόθεση για την προστασία των ασθενών.

Πρόσφατες μελέτες έχουν αξιολογήσει τις αντιλήψεις των ασθενών ως προς φορητές συσκευές. Οι Dewa et al. [Dew+19] διαπίστωσαν ότι οι νέοι με κάποιο επεισόδιο στο παρελθόν, είχαν θετική αντίληψη για τα wearables και την προσδοκία ότι η χρήση τους θα προέβλεπε ένα μελλοντικό επεισόδιο. Συγκεκριμένα για τη σχιζοφρένεια, τα wearables έχουν δείχθει αποδεκτά από τους ασθενείς και πως δεν προκαλούν κάποια σημαντική επιδείνωση των συμπτωμάτων [Mey+18].

Προϋποθέσεις εγκυρότητας

Προτού τα wearables αποκτήσουν σημαντικό κλινικό ρόλο, πρέπει οι μετρήσεις να επικυρωθούν. Ένα εμπόδιο σε αυτό είναι το ευρύ φάσμα συσκευών που χρησιμοποιείται στη βιβλιογραφία, το οποίο περιλαμβάνει από καταναλωτικά προϊόντα, όπως το Fitbit (Fitbit LLC), μέχρι πιο τυποποιημένες συσκευές ακτιγραφίας που προσφέρουν αυξημένη ακρίβεια. Ένα πρώτο βήμα επικύρωσης έχει αρχίσει να γίνεται στη βιβλιογραφία και τα αποτελέσματα είναι ενθαρρυντικά για τις περισσότερες συσκευές.

Αρκετές μελέτες είχαν ως στόχο να επικυρώσουν τις επισημειώσεις ύπνου που παράγουν καταναλωτικές συσκευές. Σε μία από αυτές οι Rookham et al. [Roo+19] συνέκριναν το Apple Watch (Apple Inc) με το κλινικά επικυρωμένο Philips Actiwatch Spectrum Pro (Koninklijke Philips NV) σε 14 υγιείς ενήλικες. Η μελέτη έδειξε ότι η το ρολόι έχει accuracy 97% και sensitivity 99% στην αναγνώριση του ύπνου και 79% specificity στην ανίχνευση wakefulness. Το ρολόι έτεινε να υποτιμά το wakefulness μετά τον ύπνο κατά 5, 74 λεπτά και να υπερεκτιμά το συνολικό χρόνο ύπνου κατά 6, 31 λεπτά. Αυτά τα αποτελέσματα υποδηλώνουν ότι το ρολόι είναι παρόμοιο, όσον αφορά την ικανότητα παρακολούθησης ύπνου, με τη συσκευή ακτιγραφίας, αν και θα πρέπει να γίνουν μελλοντικές συγκρίσεις με την πολυυπνογραφία (Polysomnography - PSG), η οποία αποτελεί την πιο αξιόπιστη μέθοδο μελέτης του ύπνου. Παρόμοιες έρευνες [Stu+21] που συγκρίνουν Fitbit Charge 2 (Fitbit LLC) με at-home polysomnography δείχνουν ότι και αυτό το wearable έχει ακριβή αποτελέσματα σχετικά με τον ύπνο και τις μετρήσεις των καρδιακών παλμών.

1.2 Πρόβλημα

1.2.1 Περιγραφή του προβλήματος

Τα δεδομένα που χρησιμοποιούνται στην παρούσα εργασία συλλέχθηκαν στο Πανεπιστήμιο Ψυχικής Υγείας, Ινστιτούτο Ερευνών Νευροεπιστημών και Ιατρικής Ακρίβειας 'Κώστας Στεφανής' (UMHRI) στην Αθήνα, Ελλάδα. Αφού ενημερώθηκαν οι συμμετέχοντες για τους στόχους του έργου, υπέγραψαν γραπτή συγκατάθεση και άδεια για τη συμμετοχή τους και τη χρήση των προσωπικών τους (ανώνυμων) δεδομένων, σύμφωνα με τις διατάξεις του Γενικού Κανονισμού (EU) 2016/679. Επιπρόσθετα, όλα τα πρωτόκολλα του έργου e-Prevention¹ έχουν εγκριθεί από την Επιτροπή Δεοντολογίας του Ιδρύματος.

Ειδικότερα, έχουμε δεδομένα που καταγράφονται συνεχώς (24/7 - εκτός περίπου 2 ωρών που το ρολόι είναι σε φόρτιση), από ένα έξυπνο ρολόι Samsung Gear S3 Frontier που διαθέτουν όλοι οι συμμετέχοντες και τα οποία συλλέγονται σε μια πλατφόρμα που βασίζεται σε cloud [Mag+20]. Η συλλογή των δεδομένων ξεκίνησε τον Νοέμβριο του 2019 και συνεχίζεται μέχρι και σήμερα.

Συγκεκριμένα, έχουμε μετρήσεις της γραμμικής επιτάχυνσης σε 3 άξονες (acc), της γωνιακής επιτάχυνσης σε 3 άξονες (gyr), τους καρδιακούς παλμούς κάθε λεπτό (hrm) και τα RR διαστήματα (χρονικά διαστήματα μεταξύ δύο διαδοχικών καρδιακών παλμών). Οι δύο πρώτες μετρήσεις δειγματοληπτούνται στα 20Hz, ενώ μετρήσεις από τον αισθητήρα του καρδιακού παλμού δειγματοληπτούνται στα 5Hz. Επιπλέον έχουμε ετικέτες για τα βήματα, τη διανυόμενη απόσταση και το πρόγραμμα του ύπνου οι οποίες προκύπτουν από το Tizen API [Mag+20] που έχει το έξυπνο ρολόι.

Συνολικά στην έρευνα συμμετέχουν 64 άτομα (26 controls και 38 patients). Οι ασθενείς υποβάλλονται σε μηνιαίες αξιολογήσεις από τους κλινικούς ιατρούς του έργου. Οι ιατροί σημειώνουν τις περιόδους του ασθενή που δεν έχει εμφανίσει συμπτώματα ψυχωτικής υποτροπής ως 'Φυσιολογική' (Normal-N) και ως 'Υποτροπή' (Relapse-R) όταν ο ασθενής έχει εμφανίσει συμπτώματα. Όπως θα δούμε και πιο αναλυτικά στην Παράγραφο 6.1.2 το παραπάνω dataset έχει κάποιες ιδιαιτερότητες. Συγκεκριμένα:

¹Περισσότερες πληροφορίες: <http://eprevention.gr/>

1. Σε αντίθεση με παλαιότερες εργασίες, οι οποίες κράτησαν από μερικές ώρες μέχρι μερικές εβδομάδες [Bar+18; Cel+18; Val+13], με κάποιες εξαιρέσεις ([Adl+20]), το συγκεκριμένο dataset έχει μετρήσεις για πάνω από τρία έτη (Νοέμβριος του 2019 μέχρι και σήμερα).
2. Παρότι υπάρχουν επισημειώσεις για τις περιόδους που ο χρήστης περπατά, κοιμάται, ή βρίσκεται σε υποτροπή, αυτές είναι πολύ λίγες σε σχέση με τον τεράστιο όγκο δεδομένων που διαθέτουμε. Είναι λοιπόν, πλεονασμός να σπαταλήσουμε ένα τόσο μεγάλο όγκο δεδομένων για ένα classification πρόβλημα με δύο μόνο κλάσεις, όπως είναι το sleep-awake πρόβλημα.
3. Υπάρχουν tasks, όπως για παράδειγμα η αναγνώριση δραστηριότητας (human activity recognition - HAR), τα οποία έχειδειχθεί ότι μπορούν να μοντελοποιηθούν από δεδομένα μας [MJ20], όμως δεν έχουμε καθόλου επισημειώσεις γι' αυτά.

Με βάση τα παραπάνω, σκοπός της παρούσας εργασίας είναι να εκμεταλλευτεί όσο το δυνατόν καλύτερα τον τεράστιο όγκο δεδομένων που έχουμε, χωρίς να βασιστεί σε επισημειώσεις. Ίδανικά θέλουμε να εκπαιδύσουμε μοντέλα τα οποία θα παράγουν αναπαραστάσεις, τέτοιες ώστε:

1. Αν μελλοντικά θέλουμε να εκπαιδύσουμε ένα μοντέλο σε ένα καινούριο task (π.χ. sleep stage recognition), τότε να χρειαζόμαστε πολύ λίγα labels γι' αυτό. Η χρήση των embeddings, δηλαδή, θα μας επιτρέψει να συλλέξουμε πολύ λιγότερες επισημειώσεις από ότι θα χρειαζόταν αν κάναμε 'end-to-end' εκπαίδευση. Πράγματι, η συλλογή labels για κάποιο task είναι δύσκολη διαδικασία, οπότε θέλουμε με λίγες επισημειώσεις (π.χ. ένα πείραμα λίγων ωρών) να πετυχαίνουμε ικανοποιητική απόδοση.
2. Αφού έχει γίνει συλλογή επισημειώσεων για κάποιο task θέλουμε να μπορούμε να αξιοποιήσουμε τη γνώση αυτή και σε άλλα tasks (transferability π.χ. από sleep-stage σε human activity recognition).
3. Όσο αυξάνεται ο όγκος των μη-επισημειωμένων δεδομένων, να βελτιώνεται η ποιότητα των αναπαραστάσεων. Αυτό γιατί, η παθητική συλλογή μη-επισημειωμένων μετρήσεων από τα ρολόγια είναι σχετικά εύκολη, επομένως, θέλουμε να αξιοποιήσουμε αυτόν τον τεράστιο όγκο δεδομένων στο μέγιστο.

1.2.2 Αντιμετώπιση του προβλήματος

Τα πειράματα που εκτελέσαμε στην παρούσα εργασία έχουν ως κύριο στόχο τη σύγκριση των αναπαραστάσεων που παράγονται από τεχνικές μηχανικής μάθησης που δεν βασίζονται σε επισημειώσεις. Βεβαίως, για να γίνει τελικά η σύγκριση θα αξιοποιήσουμε τα labels που έχουμε στο dataset, αλλά σε δεύτερο χρόνο, μετά δηλαδή την προ-εκπαίδευση των μοντέλων. Άρα:

- Αρχικά, υποθέτουμε ότι δεν έχουμε καθόλου labels και θα προ-εκπαιδύσουμε διάφορες τεχνικές unsupervised και self-supervised learning, όπως αυτές περιγράφονται στην Παράγραφο 4.1.
- Στη συνέχεια, θα συγκρίνουμε τις αναπαραστάσεις που προέκυψαν ως προς 4 είδη labels που έχει το συγκεκριμένο dataset, δηλαδή:
 - Sleep: 3 πιθανές κλάσεις sleeping, awake, transition
 - Step: 3 πιθανές κλάσεις walking, not walking, transition
 - Id: 10 πιθανές κλάσεις: το μοναδικό id του χρήστη που φοράει το smartwatch
 - Time-to-event (Relapse): Πρόβλεψη χρονικού διαστήματος, από τυχαία στιγμή μέχρι το επόμενο relapse του χρήστη.

Όπως θα δούμε στην επόμενη Παράγραφο 1.3, η συνθήκη του να έχουμε πολύ λίγες επισημειώσεις σε σύγκριση με τον όγκο των δεδομένων είναι αρκετά συνηθισμένη στα δεδομένα χρονοσειρών. Οπότε, καλό είναι να αναλύσουμε τη βιβλιογραφία του συγκεκριμένου τομέα και να δούμε ποιος είναι ο βέλτιστος τρόπος να διαχειριστούμε αυτή την ιδιαιτερότητα.

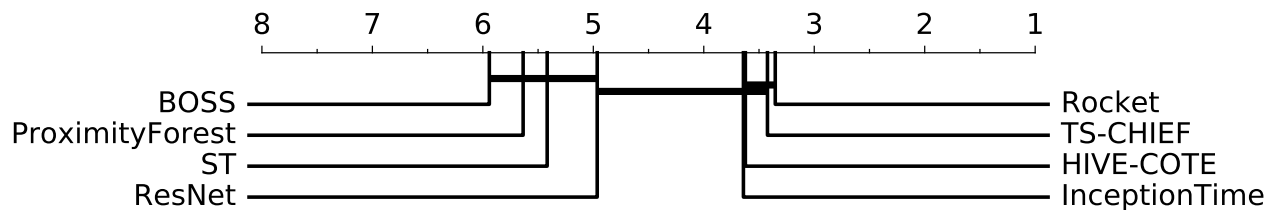
1.3 Μηχανική μάθηση και Δεδομένα χρονοσειρών

Η τεχνολογική πρόοδος των τελευταίων ετών έχει καταστήσει εύκολη την εξόρυξη δεδομένων χρονοσειρών από διάφορους τομείς. Παράλληλα, σημαντική πρόοδος έχει σημειωθεί στην ανάλυση χρονοσειρών, στη μηχανική

μάθηση, την επεξεργασία σήματος και άλλους συναφείς κλάδους, με πολλές πρακτικές εφαρμογές, όπως η υγειονομική περίθαλψη [Ste+19], η βιομηχανική διάγνωση [Kan+15] και η οικονομική πρόβλεψη [SYD19].

Ένας ακόμη τομέας στον οποίο, επίσης, έχει γίνει τεράστια πρόοδος τα τελευταία έτη είναι η Τεχνητή Νοημοσύνη (AI). Συγκεκριμένα, η πρόοδος αφορά την ανάπτυξη συστημάτων τα οποία μπορούν να μάθουν από τεράστιες βάσεις προσεκτικά επισημειωμένων (labeled) δεδομένων. Ειδικότερα, το παράδειγμα της επιβλεπόμενης μάθησης (supervised learning) σε συνδυασμό με τη βαθιά μάθηση (deep learning) έχει αποδειχθεί ο καλύτερος τρόπος για την ανάπτυξη μοντέλων, που αποδίδουν εξαιρετικά σε ένα πολύ συγκεκριμένο πρόβλημα (task) πάνω στο οποίο έχουν εκπαιδευτεί [SM19a]. Πράγματι, τα μοντέλα βαθιάς μάθησης έχουν αναδειχθεί ως επιτυχημένα μοντέλα και στην ανάλυση χρονοσειρών [HS97; GMH13; SM19b; For+19; Ore+20] και κατά καιρούς αντικαθιστούν την τελευταία λέξη της τεχνολογίας σε εργασίες όπως η πρόβλεψη (forecasting), η παλινδρόμηση (regression) και η ταξινόμηση (classification) [Bro+19; Tan+20; Ism+19a].

Ωστόσο, σε αντίθεση με τους τομείς του Computer Vision και του Natural Language Processing (NLP), η κυριαρχία της βαθιάς μάθησης για χρονοσειρές απέχει πολύ από το να έχει εδραιωθεί. Στην πραγματικότητα, μέθοδοι μη βαθιάς μάθησης όπως TS-CHIEF [Shi+20], HIVE-COTE [LTB18] και ROCKET [DPW20], οι οποίες εξηγούνται στην Παραγράφο 5, κατέχουν επί του παρόντος το ρεκόρ στα προβλήματα παλινδρόμησης και ταξινόμησης [Tan+20; Bag+17], που ταιριάζουν ή έχουν ακόμη καλύτερη απόδοση από εξελιγμένες βαθιές αρχιτεκτονικές όπως το InceptionTime [Ism+20] και το ResNet [Ism+19b] που επίσης εξηγούνται στην Παραγράφο 5 (βλ. Σχήμα 1.3.1).



Σχήμα 1.3.1: Mean rank διαφόρων classifiers στα 85 “bake off” datasets του UCR archive [Dau+19], το οποίο είναι το πιο ευρέως χρησιμοποιούμενο “benchmark dataset” σε δεδομένα χρονοσειρών.

Επιπλέον, τα υπάρχοντα μοντέλα επιβλεπόμενης βαθιάς μάθησης δεν είναι κατάλληλα για δεδομένα μεγάλων διαστάσεων χρονοσειρών με περιορισμένο αριθμό δειγμάτων εκπαίδευσης. Οι προσεγγίσεις αυτές βασίζονται στην επισημείωση των δεδομένων, διαδικασία αρκετά χρονοβόρα και μερικές φορές αδύνατη για ορισμένους τύπους δεδομένων που είναι πολύ περίπλοκοι ή θορυβώδεις, με αποτέλεσμα ανθρώπινους σχολιασμούς κακής ποιότητας. Πράγματι, ετικέτες για βιολογικά σήματα π.χ. ηλεκτροεγκεφαλογραφία (EEG) είναι δύσκολο να ληφθούν, καθώς απαιτούν ειδική γνώση. Για παράδειγμα, στο sleep staging, απαιτείται η καταγραφή των διαφορετικών σταδίων ύπνου στα recordings του ύπνου. Εκπαιδευμένοι τεχνικοί κάνουν επισημείωση χειροκίνητα σε ώρες δεδομένων [You17]. Επίσης για παθολογικές καταστάσεις, τα σήματα πρέπει να επισημειωθούν από νευρολόγους και άλλους επαγγελματίες του ιατρικού τομέα.

Σαν αποτέλεσμα ο αριθμός των διαθέσιμων, για ερευνητικούς σκοπούς, επισημειωμένων συνόλων φυσιολογικών σημάτων είναι περιορισμένος και τα υπάρχοντα σύνολα είναι σχετικά μικρά. Επιπλέον, συνήθως είναι μη συμβατά μεταξύ τους, δηλαδή έχουν διαφορετικό αριθμό καναλιών, ρυθμούς δειγματοληψίας, τύπους αισθητήρων κλπ. Αυτό τα καθιστά δύσκολο να συγχωνευτούν για την απόκτηση ενός μεγαλύτερου συνόλου δεδομένων. Γι’ αυτό και υπάρχει μεγάλο ενδιαφέρον μεθόδων που μπορούν να αξιοποιήσουν την υπάρχουσα πληθώρα δεδομένων χωρίς ετικέτα.

Μία από τις πιο στοιχειώδεις μορφές εύρεσης αναπαραστάσεων χωρίς επίβλεψη είναι το hand-crafted feature engineering. Δυστυχώς όμως οι αναπαραστάσεις αποδεικνύονται σε μεγάλο βαθμό πλεονάζουσες, με περιορισμένη διαχωριστική δύναμη για την κατασκευή μοντέλων υψηλής απόδοσης [LBH15].

Ένας άλλος τομέας έρευνας που διερευνάται σημαντικά επικεντρώνεται σε προσεγγίσεις βασισμένες στο reconstruction της εισόδου με στόχο την εξαγωγή low-dimensional embeddings με τη χρήση deep autoencoders [Vin+08]. Το μειονέκτημα αυτής της μεθόδου είναι ότι σπαταλάει το capacity του μοντέλου για να μάθει low-level details της εισόδου, αφού προσπαθεί να προβλέψει κάθε bit του σήματος. Κάτι τέτοιο δεν χρειάζεται αφού τελικός στόχος μας είναι να μάθουμε features που γενικεύουν στα τελικά (downstream/end) tasks, για παράδειγμα sleep stage classification με EEG.

Μια πολλά υποσχόμενη λύση στο πρόβλημα αυτό, είναι ο αναδυόμενος τομέας της αυτο-επιβλεπόμενης μάθησης (SSL) [Sa94], η οποία αναλύεται στην Παράγραφο 4.1. Με το SSL, η δομή των δεδομένων χρησιμοποιείται για να μετατρέψει το unsupervised πρόβλημα σε supervised, το οποίο ονομάζεται ονομάζεται “pretext task” [JT19]. Η αναπαράσταση που μαθαίνεται από το unsupervised pretext task μπορεί στη συνέχεια να ξαναχρησιμοποιηθεί σε μια supervised εργασία, μειώνοντας δυννητικά τον απαιτούμενο αριθμό επισημειωμένων παραδειγμάτων.

1.4 Κίνητρα και Συνεισφορές

Κύριο κίνητρο αυτής της διπλωματικής εργασίας είναι η συνεισφορά στο έργο e-Prevention και κατ’ επέκταση η προσφορά βοήθειας σε ασθενείς που πάσχουν από διαταραχές του φάσματος της σχιζοφρένειας. Συγκεκριμένα, στηριχθήκαμε στις ενδείξεις από έρευνες του ίδιου έργου όπως οι [Fil+20; Mag+20], στις οποίες έχουν εντοπιστεί διάφορα χαρακτηριστικά που επηρεάζουν τους ασθενείς με ψυχωτικές διαταραχές. Έτσι, ακολουθώντας παρόμοια προεπεξεργασία δεδομένων, προσπαθήσαμε να λύσουμε το ίδιο πρόβλημα, δηλαδή την πρόβλεψη υποτροπής, μέσα από τη σκοπιά του Survival-Analysis όπως αυτό εξηγείται αναλυτικά στην Παράγραφο 5.8. Η έγκαιρη ανίχνευση υποτροπών είναι πρόβλημα τεράστιας σημασίας, αφού οι ασθενείς πολλές φορές αμελούν να ενημερώνουν, όταν τα συμπτώματα αρχίζουν να επανεμφανίζονται ή να επιδεινώνονται [CLE90]. Στην πραγματικότητα, μια τέτοια πρόβλεψη, θα μπορούσε να βοηθήσει στη μείωση της σοβαρότητας της υποτροπής ή ακόμη και να την αποτρέψει τελείως [CLE90].

Παράλληλα, λύσαμε προβλήματα κατηγοριοποίησης, που αφορούν τον ύπνο του χρήστη και τη δραστηριότητά του. Αυτό είναι σημαντικό, γιατί και ο ύπνος και η δραστηριότητα σχετίζονται άμεσα με την κατάσταση που βρίσκεται ο ασθενής από σχιζοφρένεια ή διπολική διαταραχή [Coh08; Cal04].

Τέλος, δημιουργήσαμε ένα νέο πλαίσιο, που βασίζεται στο Self-Supervised Learning, το οποίο εκπαιδεύεται χωρίς τη χρήση επισημειώσεων και στη συνέχεια μπορεί να ξαναχρησιμοποιηθεί σε μια supervised εργασία, μειώνοντας δυννητικά τον απαιτούμενο αριθμό επισημειωμένων παραδειγμάτων. Οπότε, σε περίπτωση που χρειαστεί να εκτελέσουμε κάποια συλλογή επισημειώσεων στο μέλλον, αυτές θα είναι σημαντικά λιγότερες από όσες θα χρειαζόμασταν χωρίς την τεχνική αυτή.

1.5 Διάρθρωση της Διπλωματικής εργασίας

Η δομή της διπλωματικής εργασίας οργανώνεται ως εξής:

- Στο κεφάλαιο 2 παρουσιάζονται οι βασικές αρχές της Μηχανικής Μάθησης και ειδικότερα του υποτομέα του Representation Learning πάνω στον οποίο βασίζεται η εργασία.
- Στο κεφάλαιο 3 περιγράφονται βασικές ιδιότητες που έχουν τα φυσικά σήματα, όπως η εικόνα, ο ήχος ή τα βιολογικά σήματα που χρησιμοποιούμε. Στη συνέχεια, εξηγείται πώς οι σύγχρονες αρχιτεκτονικές νευρωνικών δικτύων αξιοποιούν τις ιδιότητες αυτές με στόχο να μειώσουν τον αριθμό των παραμέτρων τους.
- Στο κεφάλαιο 4 αναφέρονται τα όρια στο πόσο μακριά μπορεί να φτάσει η Τεχνητή Νοημοσύνη μόνο με την επιβλεπόμενη μάθηση, πάνω στην οποία βασίζονται τα περισσότερα σύγχρονα συστήματα. Επίσης, γίνεται εισαγωγή στην έννοια της Αυτο-επιβλεπόμενης μάθησης (SSL), μια τεχνική που ίσως είναι ικανή να ξεπεράσει τα παραπάνω όρια. Τέλος, παρουσιάζονται οι πιο επιτυχημένες αρχιτεκτονικές αυτο-επιβλεπόμενης μάθησης, οι οποίες αφορούν κυρίως τον τομέα της όρασης υπολογιστών.
- Στο κεφάλαιο 5 παρουσιάζονται οι τεχνικές που χρησιμοποιήθηκαν στην παρούσα εργασία. Αυτές περιλαμβάνουν: Hand-crafted Engineering σε συνδυασμό με Feature-selection, Random Convolutional Kernels, και δύο τεχνικές SSL με εφαρμογή στις χρονοσειρές: SelfTime και TSBert. Επίσης παρουσιάζεται μια state-of-the-art supervised αρχιτεκτονική (InceptionTime), η οποία χρησιμοποιείται για σύγκριση με τα προηγούμενα, καθώς επίσης και μία τεχνική μετατροπής δεδομένων χρονοσειρών σε δεδομένα εικόνων. Τέλος, παρουσιάζεται για πληρότητα η τεχνική του Survival Analysis, καθώς δεν είναι τόσο διαδεδομένη όσο οι τεχνικές του classification και regression.
- Στο κεφάλαιο 6 γίνεται παρουσίαση και αξιολόγηση των αποτελεσμάτων, έπειτα από εφαρμογή των παραπάνω τεχνικών στα δεδομένα του e-Prevention.

- Στο κεφάλαιο 7 παρουσιάζονται τα συμπεράσματα και πιθανές μελλοντικές επεκτάσεις αυτής της διπλωματικής εργασίας.

Κεφάλαιο 2

Θεωρητικό Υπόβαθρο

2.1	Μηχανική μάθηση και no free lunch theorem	10
2.2	Αναπαράσταση δεδομένων σε χώρους πολλών διαστάσεων	10
2.2.1	Blessings of dimensionality	10
2.2.2	Curse of Dimensionality	11
2.3	The manifold hypothesis	14
2.4	Επέκταση των διαστάσεων της εισόδου	17
2.5	Νευρωνικά δίκτυα ως universal approximators	17
2.6	Ιεραρχική αναπαράσταση της εισόδου	18
2.7	Μετασχηματισμοί που συμβαίνουν σε κάθε στρώμα	20

2.1 Μηχανική μάθηση και no free lunch theorem

Μηχανική μάθηση είναι ο τομέας στον οποίο αναπτύσσονται μοντέλα τα οποία μαθαίνουν από πεπερασμένο σύνολο δεδομένων (training set) με σκοπό να γενικεύουν σωστά σε δεδομένα που δεν έχουν ξαναδεί. Αυτό αρχικά φαίνεται να έρχεται σε αντίθεση με τις βασικές αρχές της Μαθηματικής Λογικής. Για να συμπεράνουμε λογικά έναν κανόνα που περιγράφει κάθε μέλος ενός συνόλου, πρέπει να έχουμε πληροφορίες για κάθε μέλος αυτού του συνόλου. Εν μέρει, η μηχανική μάθηση προσπερνά αυτό το πρόβλημα προσφέροντας μόνο πιθανοτικούς κανόνες. Ψάχνει δηλαδή να βρει κανόνες που είναι **πιθανώς** σωστοί για τα περισσότερα μέλη του συνόλου. Ωστόσο, αυτή η λογική έχει ως εμπόδιο το παρακάτω θεώρημα 2.1.1:

Θεώρημα 2.1.1: No free lunch theorem [WM97; Wol96]

Αν πάρουμε τον μέσο όρο από όλες οι πιθανές κατανομές παραγωγής δεδομένων, τότε κάθε αλγόριθμος που κάνει κατηγοριοποίηση έχει ίδιο ποσοστό σφάλματος στο test set. Με άλλα λόγια, κανένας αλγόριθμος μηχανικής μάθησης δεν είναι καθολικά καλύτερος. Ο πιο εξελιγμένος αλγόριθμος που μπορούμε να φανταστούμε έχει τον ίδιο μέσο όρο σφάλματος (ως προς όλα τα tasks που υπάρχουν) σαν να ανέθετε κάθε σημείο στην ίδια κατηγορία.

Ευτυχώς, το παραπάνω ισχύει μόνο όταν υπολογίζουμε τον μέσο όρο όλων των πιθανών κατανομών δημιουργίας δεδομένων. Αν κάνουμε υποθέσεις για το είδος της κατανομής, για κάθε τύπο δεδομένων που συναντάμε, τότε μπορούμε να σχεδιάσουμε αλγόριθμους που έχουν καλή απόδοση σε αυτές τις κατανομές. Αυτό σημαίνει ότι ο στόχος της έρευνας της μηχανικής μάθησης δεν πρέπει να είναι η αναζήτηση του καλύτερου μοντέλου-αλγόριθμου. Αντίθετα, πρέπει να βρούμε ποια είδη αλγορίθμων έχουν καλή απόδοση, αναλόγως των δεδομένων που μας ενδιαφέρουν κάθε φορά.

2.2 Αναπαράσταση δεδομένων σε χώρους πολλών διαστάσεων

Ο τομέας της Τεχνητής Νοημοσύνης και ειδικότερα ο υποτομέας του Representation Learning βασιζόταν για πολλές δεκαετίες στα hand-crafted, task-specific χαρακτηριστικά. Αντιθέτως τα τελευταία χρόνια η έρευνα έχει στραφεί σε χαρακτηριστικά γενικού σκοπού τα οποία, μάλιστα, εκπαιδεύονται από τα ίδια τα δεδομένα. Και οι δύο μέθοδοι μετασχηματίζουν τα δεδομένα σε ένα χώρο μεγαλύτερης διάστασης. Όπως θα δούμε παρακάτω υπάρχει θεωρητικός λόγος που κάνουμε κάτι τέτοιο και δυστυχώς ο μετασχηματισμός αυτός εμπεριέχει και πολλά προβλήματα. Έχουμε λοιπόν από τη μία τα “blessings of dimensionality” και από την άλλη τις “curses of dimensionality” όπως αναφέρονται στη βιβλιογραφία [Don+00].

2.2.1 Blessings of dimensionality

Από πολύ νωρίς (1965) ο Cover διατύπωσε το παρακάτω θεώρημα 2.2.2, το οποίο τονίζει την ανάγκη να πάμε σε χώρους μεγαλύτερης διάστασης, καθώς εκεί τα δεδομένα είναι πιθανότερο να είναι γραμμικά διαχωρίσιμα. Δίνεται, επίσης τυπικός ορισμός 2.2.1 του τι σημαίνει γραμμικά διαχωρίσιμα σύνολα:

Ορισμός 2.2.1: Γραμμικά διαχωρίσιμα σύνολα

Ένα σύνολο $S \subset \mathbb{R}^n$ είναι γραμμικά διαχωρίσιμο αν για κάθε $x \in S$ υπάρχει γραμμικό συναρτησιακό l τέτοιο ώστε $l(x) > l(y)$ για κάθε $y \in S$, $y \neq x$.

Θεώρημα 2.2.2: Θεώρημα του Cover [Cov65]

Σε έναν Ευκλείδειο χώρο d διαστάσεων υπάρχουν $C(N, d)$ τρόποι να χωρίσουμε N σημεία σε 2 κλάσεις και τελικά αυτά να είναι γραμμικά διαχωρίσιμα, όπου:

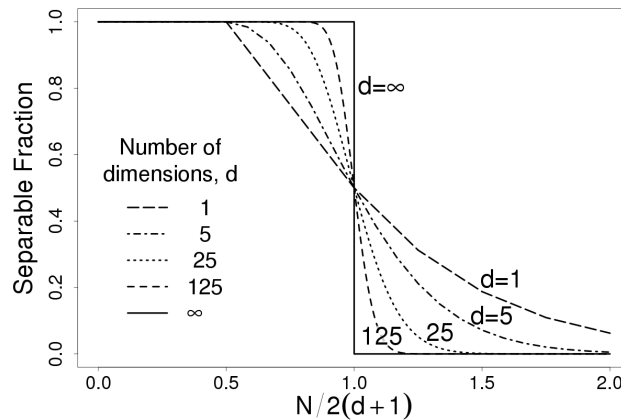
$$C(N, d) = 2 \sum_{k=0}^d \binom{N-1}{k}$$

Αυτό, με τον περιορισμό τα σημεία να μην «παρατάσσονται», δηλαδή να μην βρίσκονται πάνω από δύο σημεία στην ίδια γραμμή σε ένα διδιάστατο χώρο, όχι περισσότερα από τρία στο ίδιο επίπεδο σε έναν

τριδιάστατο χώρο κλπ.

Είναι σημαντικό να τονίσουμε ότι το αποτέλεσμα του παραπάνω θεωρήματος είναι ανεξάρτητο από την κατανομή των N σημείων, δηλαδή έχουμε πάντα ακριβώς $C(N, d)$. Συμβολίζοντας με $f(N, d)$ το κλάσμα των διχοτομήσεων, προκύπτει το παρακάτω Σχήμα 2.2.1. Διαισθητικά: Παίρνοντας N στοιχεία από οποιαδήποτε d -διάστατη κατανομή και χωρίζοντάς τα σε 2 κλάσεις στο d -διάστατο χώρο, τότε:

- αν $N < 2(d + 1)$, τότε με μεγάλη πιθανότητα θα είναι γραμμικά διαχωρίσιμα. **Δηλαδή σε χώρους πολλών διαστάσεων είναι πιο πιθανό τα σημεία να είναι γραμμικά διαχωρίσιμα.**
- αν $N > 2(d + 1)$, τότε με μεγάλη πιθανότητα δεν θα είναι γραμμικά διαχωρίσιμα



Σχήμα 2.2.1: Ποσοστό διαχωρίσιμων σημείων [SB11]

Οι χώροι πολλών διαστάσεων μελετήθηκαν από τους Maxwell, Boltzmann, Gibbs Einstein κατά την ανάπτυξη του στατιστικού υποβάθρου της θερμοδυναμικής [Gib16]. Συγκεκριμένα είδαν ότι για πολλά σωματίδια οι συναρτήσεις κατανομής έχουν ξεχωριστές ιδιότητες, οι οποίες ονομάστηκαν στη συνέχεια “measure of concentration” [GT18; Ver18; GM00; Led01]. Τα κλασικά θεωρήματα “measure of concentration” δηλώνουν ότι τυχαία σημεία σε χώρους πολλών διαστάσεων συγκεντρώνονται σε ένα λεπτό στρώμα κοντά σε μια επιφάνεια (μια σφαίρα, ένα σύνολο μέσου ή διάμεσου επιπέδου ενέργειας κτλ.). Αυτό θα το δούμε και στην επόμενη παράγραφο. Τα θεωρήματα “stochastic separation theorems” [GT17] περιγράφουν τη δομή αυτών των λεπτών στρωμάτων: Τα τυχαία σημεία δεν συγκεντρώνονται απλώς σε ένα λεπτό στρώμα, αλλά είναι όλα γραμμικά διαχωρισμένα από το υπόλοιπο του συνόλου ακόμη και για εκθετικά μεγάλα τυχαία σύνολα.

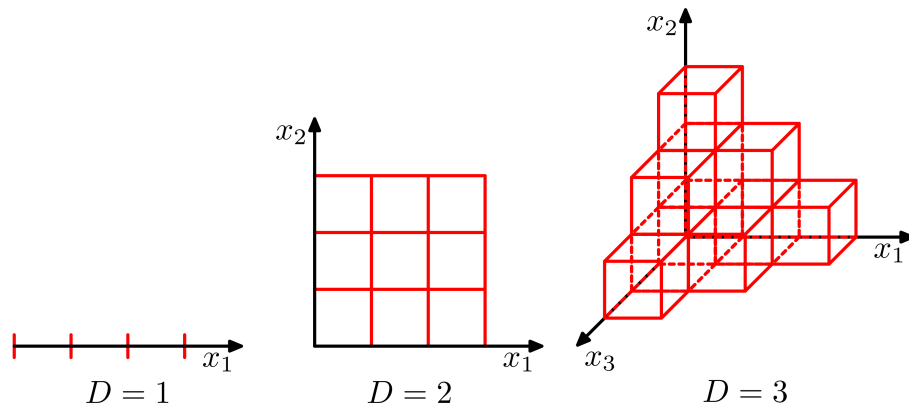
2.2.2 Curse of Dimensionality

Απόσταση μεταξύ σημείων

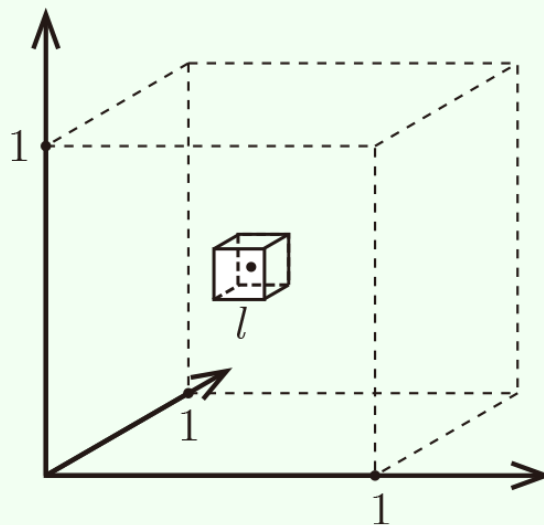
Παρά την ευκολία που μας δίνουν οι χώροι πολλών διαστάσεων, ως προς το γραμμικό διαχωρισμό των σημείων, προκύπτει το εξής πρόβλημα: Καθώς αυξάνουμε τον αριθμό των διαστάσεων ο όγκος μέσα στον οποίο μπορεί να βρεθεί ένα σημείο αυξάνεται εκθετικά και άρα τα δεδομένα μας βρίσκονται μακριά το ένα από το άλλο. Μία οπτικοποίηση μέχρι τις 3 διαστάσεις φαίνεται στο Σχήμα 2.2.2. Άρα, οποιαδήποτε ανάλυση και αν εφαρμόσουμε, τα δεδομένα που απαιτούνται για να έχουμε στατιστική σημαντικότητα αυξάνονται εκθετικά σε σχέση με τον αριθμό των διαστάσεων. Το πρόβλημα γίνεται εμφανές στο παράδειγμα 2.2.3.

Παράδειγμα 2.2.3: 10 κοντινότεροι γείτονες τυχαίου σημείου στον μοναδιαίο υπερκύβο

Δειγματοληπτούμε n σημεία από μία ομοιόμορφη κατανομή μέσα στον μοναδιαίο υπερκύβο που φαίνεται στο Σχήμα 2.2.3. Στη συνέχεια υπολογίζουμε πόσο χώρο μέσα σε αυτόν καταλαμβάνουν οι $k=10$ κοντινότεροι γείτονες ενός τυχαίου σημείου. Έστω ο μοναδιαίος υπερκύβος διάστασης d $[0, 1]^d$. Όλα τα σημεία δειγματοληπτούνται από ομοιόμορφη κατανομή μέσα σε αυτόν τον κύβο, π.χ. $\forall i, x_i \in [0, 1]^d$, και θεωρούμε τους $k = 10$ κοντινότερους γείτονες ενός τυχαίου σημείου.



Σχήμα 2.2.2: Ο αριθμός των περιοχών ενός κανονικού πλέγματος μεγαλώνει εκθετικά με τη διάσταση d του χώρου [Bis06].



Σχήμα 2.2.3: Μοναδιαίος υπερκύβος και κύβος ακμής μήκους ℓ που περιέχει τους 10 κοντινότερους γείτονες [Wei21].

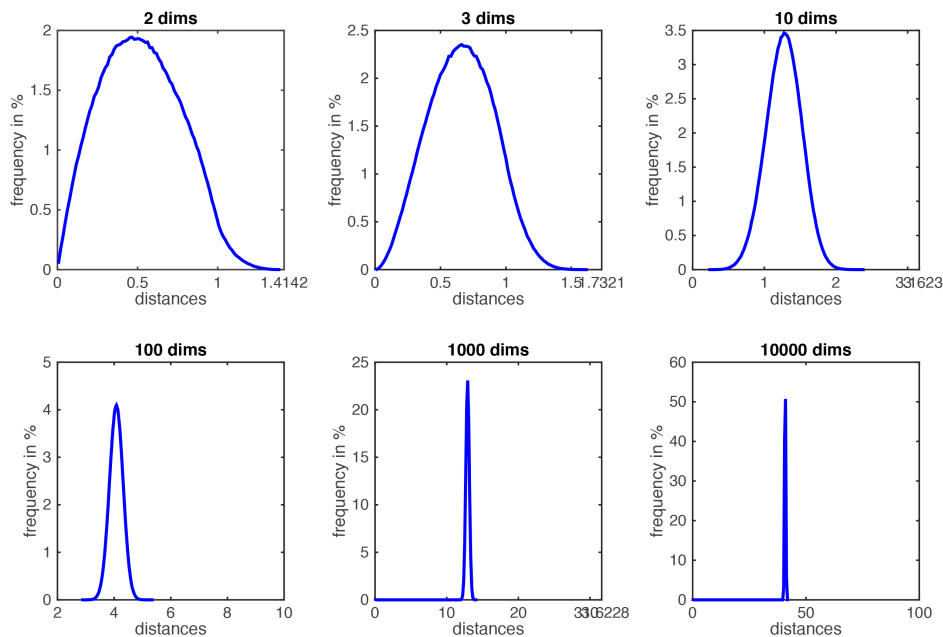
Έστω ℓ το μήκος της ακμής του μικρότερου υπερκύβου που περιλαμβάνει τους k -κοντινότερους γείτονες του επιλεγμένου σημείου. Τότε $\ell^d \approx \frac{k}{n}$ (λόγω ομοιόμορφης κατανομής) και $\ell \approx \left(\frac{k}{n}\right)^{1/d}$. Για διάφορες τιμές του d οι αντίστοιχες τιμές του ℓ είναι:

d	ℓ
2	0.1
10	0.63
100	0.955
1000	0.9954

Πίνακας 2.1: Αντιστοιχία d με ℓ [Wei21].

Άρα όσο $d \gg 0$ σχεδόν ολόκληρος ο υπερκύβος χρειάζεται για να χωρέσουν οι 10-κοντινότεροι-γείτονες. Δηλαδή οι κοντινότεροι γείτονες έχουν την ίδια απόσταση με τα υπόλοιπα σημεία. Ειδικότερα, όσο οι διαστάσεις αυξάνονται:

1. Δεν υπάρχουν γείτονες, αφού δεν έχουμε σχεδόν κανένα ζευγάρι σημείων με απόσταση 0.
 2. Όλα τα σημεία έχουν την ίδια απόσταση, ίση με τη μέγιστη απόσταση μεταξύ των κορυφών του κύβου, όπως φαίνεται και στο Σχήμα 2.2.4.
 3. Σχεδόν όλο το εσωτερικό του κύβου άδειο και όλα τα σημεία βρίσκονται στις ακμές και κορυφές του.
- Αν αυξάναμε τον αριθμό των δεδομένων εκπαίδευσης, n , μέχρι οι κοντινότεροι γείτονες να ήταν πραγματικά κοντά στο επιλεγμένο σημείο, πόσα δεδομένα θα θέλαμε ℓ να γίνει σχετικά μικρό?
 - Έστω $\ell = \frac{1}{10} = 0.1 \Rightarrow n = \frac{k}{\ell^d} = k \cdot 10^d$, που αυξάνεται εκθετικά. Για $d > 100$ θα θέλαμε περισσότερα δείγματα εκπαίδευσης από τον αριθμό των ηλεκτρονίων στο σύμπαν!



Σχήμα 2.2.4: Ιστογράμματα εμφανίσεων pairwise-αποστάσεων για κάθε d . Βλέπουμε ότι όσο οι διαστάσεις αυξάνονται: α) Δεν υπάρχουν γείτονες (σχεδόν κανένα σημείο απόσταση 0). β) Όλα τα σημεία έχουν ίδια απόσταση ίση με τη μέγιστη απόσταση μεταξύ των κορυφών του κύβου [Wei21].

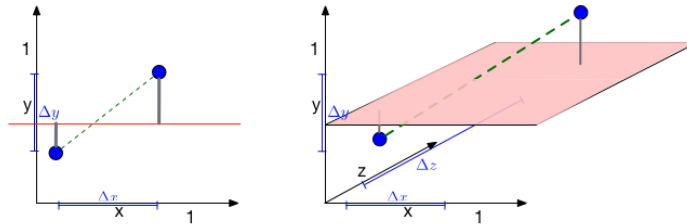
Απόσταση σημείου με υπερεπίπεδο

Είδαμε ότι η απόσταση μεταξύ δύο τυχαίων σημείων αυξάνει όταν αυξάνονται οι διαστάσεις. Άραγε συμβαίνει το ίδιο για την απόσταση σημείου από υπερεπίπεδο?

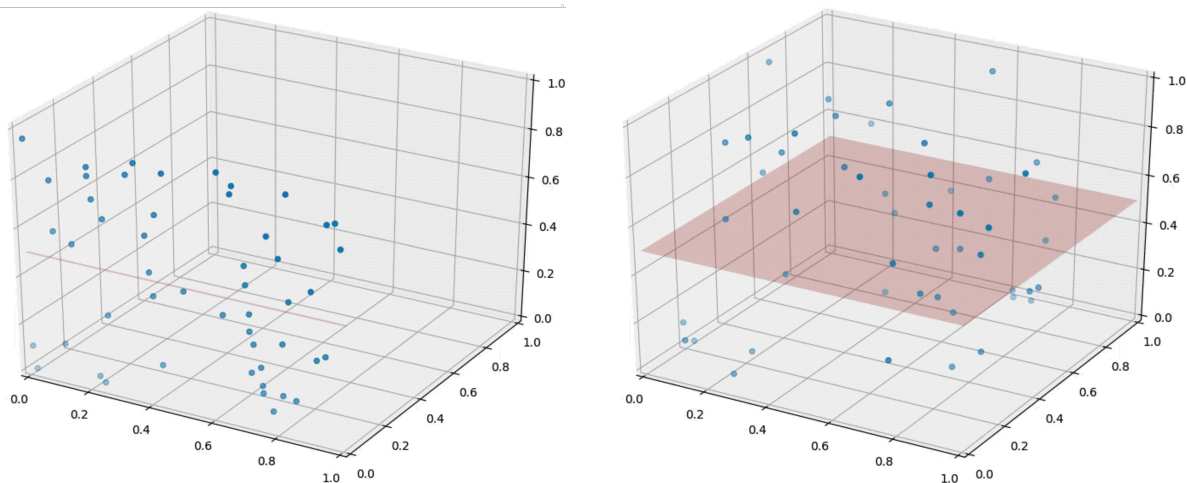
Η απάντηση βρίσκεται στο Σχήμα 2.2.5 και είναι όχι. Υπάρχουν δύο μπλε σημεία και ένα κόκκινο υπερεπίπεδο. Όταν $d = 2$, η απόσταση μεταξύ δύο σημείων είναι $\sqrt{\Delta x^2 + \Delta y^2}$. όταν πάμε στις τρεις διαστάσεις η απόσταση γίνεται $\sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2}$, η οποία είναι μεγαλύτερη ή ίση με αυτή των δύο διαστάσεων. Αυτό επιβεβαιώνει ότι οι pairwise-αποστάσεις μεγαλώνουν όσο αυξάνεται ο αριθμός των διαστάσεων. Από την άλλη, η απόσταση ως προς το κόκκινο υπερεπίπεδο παραμένει ίδια όταν πάμε στις 3 διαστάσεις. Ο λόγος που συμβαίνει αυτό είναι ότι το διάνυσμα του υπερεπιπέδου είναι κάθετο στη νέα διάσταση. Γενικά σε d διαστάσεις, $d - 1$ διαστάσεις θα είναι κάθετες στο διάνυσμα κάθε υπερεπιπέδου. Όπως φαίνεται στο Σχήμα 2.2.6 όσο κινούμαστε ως προς την

τρίτη διάσταση η απόσταση μεταξύ των σημείων αυξάνεται ενώ η απόστασή τους από το υπερεπίπεδο παραμένει σταθερή.

Όσο οι pairwise αποστάσεις γίνονται πολύ μεγάλες στις πολλές διαστάσεις, οι αποστάσεις από υπερεπίπεδα γίνονται μικροσκοπικές σε σύγκριση με αυτές (αφού παραμένουν σταθερές). Όμως πολλά μοντέλα χρησιμοποιούν υπερεπίπεδα μεταξύ clusters διαφορετικών κατηγοριών. Άρα τα περισσότερα σημεία τείνουν να είναι πολύ κοντά σε αυτά τα υπερεπίπεδα. Μία συνέπεια αυτού είναι ότι αν διαταραχθεί ελαφρά η είσοδος μπορεί να αλλάξει το αποτέλεσμα της ταξινόμησης. Αυτό το πρόβλημα αναφέρεται συχνά και ως δημιουργία adversarial δείγματα [Sze+14].



Σχήμα 2.2.5: Η απόσταση μεταξύ σημείων αυξάνεται αφού τα σημεία αυτά απέχουν επιπλέον Δz ως προς τον άξονα z . Η απόστασή τους ως προς το επίπεδο παραμένει σταθερή [Wei21].



Σχήμα 2.2.6: **Αριστερά:** Τυχαία δειγματοληπτημένα σημεία στο διδιάστατο επίπεδο. **Δεξιά:** Προσθήκη μίας τρίτης διάστασης με τυχαίες συντεταγμένες. Βλέπουμε ότι η απόσταση μεταξύ των σημείων αυξάνεται, ενώ η απόστασή τους από το υπερεπίπεδο παραμένει σταθερή [Wei21].

2.3 The manifold hypothesis

Είδαμε στην Παράγραφο 2.2.1 ότι αν και στους χώρους πολλών διαστάσεων τα δεδομένα μας είναι εύκολα γραμμικά διαχωρίσιμα, δυστυχώς ταυτόχρονα καταργείται και η έννοια της απόστασης και απέχουν όλα μεταξύ τους το ίδιο. Επίσης τα δεδομένα που χρειαζόμαστε για να περιγράψουμε το χώρο αυξάνουν εκθετικά με τη διάστασή του. Όμως παρόλα αυτά παρατηρούμε στη βιβλιογραφία αλγορίθμους μηχανικής μάθησης όπου πετυχαίνουν εξαιρετικά αποτελέσματα. Δηλαδή βρίσκουν συναρτήσεις που περιγράφουν ολόκληρο τον χώρο \mathbb{R}^n για n -διάστατη είσοδο. Πώς συμβαίνει αυτό;

- Τα δεδομένα **υποθέτουμε** ότι ζουν σε *manifolds*. Δηλαδή όλος ο \mathbb{R}^n περιέχει μη έγκυρες εισόδους (θόρυβο) και οι εισόδου που μας ενδιαφέρουν ζουν σε μία συλλογή από *manifolds*. Άρα και η συνάρτηση που θα πρέπει να μάθουμε έχει ενδιαφέρουσες μεταβολές μόνο όταν κινούμαστε επάνω στο *manifold* (ή από *manifold* σε *manifold*)

Ορισμός 2.3.1: Manifold

Παρόλο που υπάρχει αυστηρός ορισμός για τον όρο manifold, για το πλαίσιο της μηχανικής μάθησης αρκεί ο εξής ορισμός:

Είναι ένα σύνολο σημείων που μπορεί να προσεγγιστεί λαμβάνοντας υπόψη μόνο ένα μικρό αριθμό βαθμών-ελευθερίας/διαστάσεων ενώ εμπεριέχεται σε ένα χώρο μεγαλύτερων βαθμών-ελευθερίας/διαστάσεων. Σε κάθε σημείο το manifold τοπικά είναι ένας ευκλείδειος χώρος.

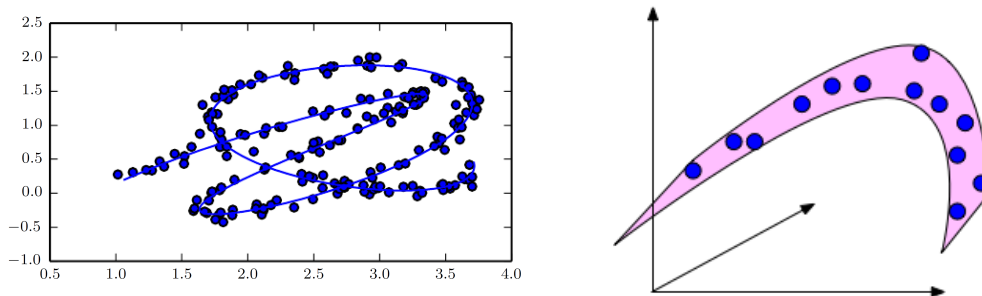
Άρα ένα μοντέλο μηχανικής μάθησης μπορεί να αναπαραστήσει τα δεδομένα μας σε συντεταγμένες στο manifold, αντί για συντεταγμένες στο \mathbb{R}^n . Αυτό γίνεται σαφές και από το Σχήμα 2.3.1 και από το παρακάτω διαισθητικό παράδειγμα:

Παράδειγμα 2.3.2: Επιφάνεια της γης και δρόμοι της πόλης ως manifolds

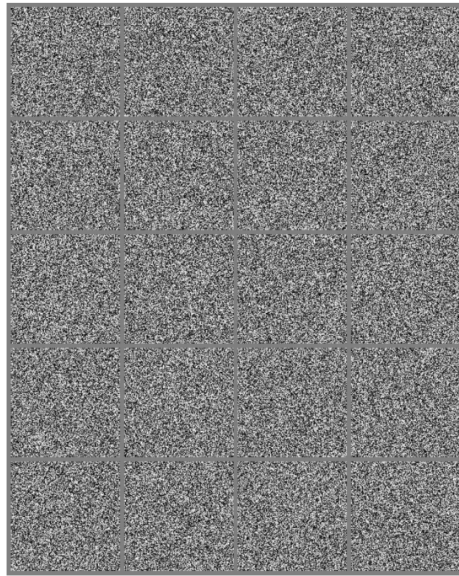
- Στην καθημερινή ζωή, βιώνουμε την επιφάνεια του κόσμου ως δισδιάστατο επίπεδο, αλλά στην πραγματικότητα είναι ένα σφαιρικό manifold ενσωματωμένο στον τρισδιάστατο χώρο. Έτσι μπορούμε να περπατήσουμε π.χ. βόρεια ή νότια δηλαδή με μετασχηματισμούς που αναφέρονται στο 2-διάστατο επίπεδο.
- Επίσης οι δρόμοι της πόλης μπορούν να περιγραφούν από ένα μονοδιάστατο σύνολο σημείων ενσωματωμένο στον 3-διάστατο χώρο. Έτσι μπορούμε να δώσουμε π.χ. οδηγίες για μία διεύθυνση δίνοντας διαδοχικές διευθύνσεις, αντί για διαδοχικές συντεταγμένες στον 3-διάστατο ή 2-διάστατο χώρο.

Η υπόθεση ότι τα δεδομένα ζουν σε ένα manifold χαμηλότερων διαστάσεων μπορεί να μην ισχύει πάντοτε. Ωστόσο όταν διαχειριζόμαστε δεδομένα που παρουσιάζουν τον πραγματικό κόσμο όπως εικόνες, ήχος ή κείμενο, τότε αυτή η υπόθεση φαίνεται να είναι -τουλάχιστον προσεγγιστικά- αληθής. Αυτό προκύπτει από τις εξής δύο παρατηρήσεις:

- Η κατανομή των δεδομένων εικόνων, ήχου, κειμένου κλπ. είναι εξαιρετικά συγκεντρωμένη σε συγκεκριμένα σημεία. Δηλαδή ομοιόμορφα τυχαίες εισδοδοί είναι θόρυβος σε όλους αυτούς τους τομείς. Αν φτιάξουμε κείμενο από τυχαία γράμματα στη σειρά η πιθανότητα να έχουμε μια σωστή λέξη είναι σχεδόν μηδενική. Το ίδιο αν διαλέξουμε τυχαίες λέξεις με σκοπό να δημιουργήσουμε μία πρόταση. Άρα η κατανομή της φυσικής γλώσσας καταλαμβάνει ένα πολύ μικρό όγκο από όλους τους πιθανούς συνδυασμούς που υπάρχουν. Επιπλέον, αν πάρουμε ομοιόμορφα τυχαίες τιμές για pixels εικόνας παίρνουμε με μεγάλη πιθανότητα θόρυβο όπως στο Σχήμα 2.3.2.
- Φυσικά η συγκέντρωση στην κατανομή των δεδομένων δεν συνεπάγεται ότι τα δεδομένα ζουν σε manifolds. Πρέπει, επιπλέον, να δείξουμε ότι αυτά συνδέονται μεταξύ τους. Να δείξουμε, δηλαδή ότι παρόμοια παραδείγματα μπορούν να παραχθούν διασχίζοντας το manifold. Τέτοια φαινόμενα έχουν παραχθεί και στους 3 προαναφερθέντες τομείς πειραματικά [Cay05; NM10; DG03; WSS04]. Ένα παράδειγμα σε εικόνα φαίνεται στο Σχήμα 2.3.3.



Σχήμα 2.3.1: **Αριστερά:** βλέπουμε σημεία από μια κατανομή σε έναν δισδιάστατο χώρο που είναι στην πραγματικότητα συγκεντρωμένα ένα μονοδιάστατο manifold, που προσομοιάζει μία χορδή. Η συμπαγής γραμμή δείχνει το manifold που πρέπει να μάθει ο αλγόριθμος [GBC16]. **Δεξιά:** φαίνεται το ίδιο για 3-διάστατο χώρο και 2-διάστατο manifold [Wei21].



Σχήμα 2.3.2: Δειγματοληψία εικόνων ομοιόμορφα τυχαία (επιλέγοντας τυχαία κάθε pixel σύμφωνα με ομοιόμορφη κατανομή) δημιουργεί θορυβώδεις εικόνες [GBC16].



Σχήμα 2.3.3: Παραδείγματα δειγμάτων εκπαίδευσης από το QMUL Multiview Face Dataset [GMP00] για το οποίο ζητήθηκε από τους συμμετέχοντες να κινηθούν με τέτοιο τρόπο ώστε να διασχίσουν το δισδιάστατο manifold που αντιστοιχεί σε δύο γωνίες περιστροφής.

2.4 Επέκταση των διαστάσεων της εισόδου

Στις προηγούμενες παραγράφους είδαμε την ανάγκη να προβάλλουμε τα δεδομένα μας σε χώρους πολλών διαστάσεων. Η εξαγωγή χαρακτηριστικών (feature extraction) είναι η διαδικασία κατά την οποία το πετυχαίνουμε αυτό, με τελικό στόχο στη νέα αναπαράσταση τα δεδομένα να είναι γραμμικά διαχωρίσιμα (linearly separable). Μαθαίνουμε δηλαδή μια συνάρτηση $f(x) = \sum_i w_i \phi_i(x)$, όπου w_i είναι οι συντελεστές προς εκπαίδευση, και ϕ_i είναι συναρτήσεις που επιλέγουμε εμείς. Για την επιλογή των $\phi_i(x)$, οι πιο συνηθισμένες τεχνικές είναι:

1. **Hand-crafted/Engineered-features:** Χρησιμοποιούμε γραμμικούς συνδυασμούς χαρακτηριστικών $\phi_i(x) = \phi(x, u_i)$, όπου η μεταβλητή u_i συμβολίζει τον τύπο των δεδομένων (π.χ. εικόνα ή ήχος). Δυστυχώς αυτή η μέθοδος δεν αποδίδει καλά, όσο αυξάνονται οι διαστάσεις d , αφού χρειάζονται n^d υπο-συναρτήσεις για να καλύψουμε το πλέγμα μήκους n σε κάθε διάσταση.
2. **Τυχαίες προβολές:** Χρησιμοποιούμε πίνακες με τυχαία βάρη ώστε να μετασχηματίσουμε την είσοδο σε features. Έτσι αξιοποιούμε το blessing of dimensionality που είδαμε. Αυτές οι μέθοδοι φαίνονται να πηγαίνουν καλά στην πράξη [SW17].
3. **Polynomial classifier:** Χρησιμοποιούμε μονώνυμα ως ϕ_i . Αν τα δεδομένα είναι διάστασης d , μπορούμε να κατασκευάσουμε πολυώνυμα όπου οι όροι τους είναι γινόμενα συγκεκριμένων διαστάσεων. Για παράδειγμα, για $d = 2$ (άρα έχω (x_1, x_2)), το πολυώνυμο έχει μορφή:

$$w_0 + \sum_{a \in \mathbb{N}^*} w_{1,a} x_1^a + \sum_{b \in \mathbb{N}^*} w_{2,b} x_2^b + \sum_{c_1, c_2 \in \mathbb{N}^*} w_{3,c_1,c_2} x_1^{c_1} x_2^{c_2}.$$

Όμως έτσι έχουμε πολυωνυμικά πολλούς συνδυασμούς όσο ανεβαίνει η διάσταση. Μια λύση είναι να περιορίσουμε το βαθμό του πολυωνύμου π.χ., για $d = 2$ να έχουμε πολυώνυμα μέχρι 2ου βαθμού: $w_0 + w_{1,1}x_1 + w_{2,1}x_2 + w_{1,2}x_1^2 + w_{2,2}x_2^2 + w_{3,1,1}x_1x_2$.

4. **Radial Basis Functions (RBF):** Είναι συναρτήσεις των οποίων η τιμή εξαρτάται από την απόσταση της μεταβλητής από ένα επιλεγμένο σημείο. Μία τέτοια συνάρτηση που χρησιμοποιείται συχνά είναι: $\phi_i(x) = e^{-\|x-u_i\|^2}$.
5. **Kernel machines:** Χρησιμοποιούμε kernel συναρτήσεις που ικανοποιούν την “Mercer’s condition”. Συγκεκριμένα χρησιμοποιούμε $\phi_i(x) = K(x, u_i)$, όπου K συνεχής, συμμετρική, θετικά ορισμένη kernel συνάρτηση (δηλαδή ο πίνακας $[K(x_i, x_j)]_{i,j}$ είναι θετικά ορισμένος, όπου x_i είναι τα δεδομένα μας). Στην πραγματικότητα παίρνοντας $\phi_i(x) = K(x, x_i)$ είναι ισοδύναμο με ένα νευρωνικό δίκτυο 2 στρωμάτων.

Όπως βλέπουμε ένας γραμμικός ταξινομητής, πάνω στα features των παραπάνω μεθόδων θέλει θεωρητικά άπειρες διαστάσεις, άρα πρέπει να ψάξουμε για κάτι καλύτερο.

Ορισμός 2.4.1: Mercer’s condition

- Μία συνάρτηση $K(x, y)$ ικανοποιεί την “Mercer’s condition” αν για όλες τις συναρτήσεις $g(x)$ που έχουν ολοκλήρωμα του τετραγώνου της απόλυτης τιμής πεπερασμένο, ισχύει:

$$\iint g(x)K(x, y)g(y) dx dy \geq 0.$$

- Το διακριτό ανάλογο είναι ένας θετικά ημιορισμένος πίνακας K διάστασης N , όπου για όλα τα διανύσματα g , ισχύει:

$$(g, Kg) = g^T \cdot Kg = \sum_{i=1}^N \sum_{j=1}^N g_i K_{ij} g_j \geq 0$$

2.5 Νευρωνικά δίκτυα ως universal approximators

Ένα γραμμικό μοντέλο όπου το μόνο που κάνει είναι πολλαπλασιασμοί πινάκων με την είσοδο μπορεί να μάθει μόνο γραμμικές συναρτήσεις. Στην πλειοψηφία τους, όμως, οι συναρτήσεις που θέλουμε να μάθουμε είναι

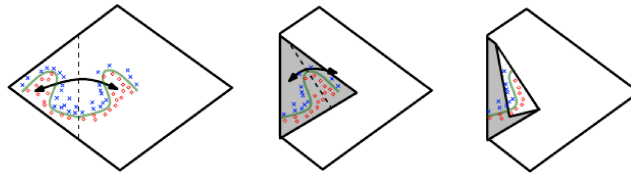
μη-γραμμικές. Κάποιος θα πίστευε ότι ανάλογα με το nonlinearity που θέλουμε να αναπαραστήσουμε θα χρειαζόταν μοντέλο σχεδιασμένο ειδικά γι' αυτό. Ωστόσο, έχει αποδειχθεί ότι, τα νευρωνικά δίκτυα με τουλάχιστον ένα κρυφό στρώμα στο οποίο εφαρμόζεται μη-γραμμικός (nonpolynomial) μετασχηματισμός, μπορούν να αναπαραστήσουν οποιαδήποτε συνάρτηση ¹. Επίσης μπορούν να αναπαραστήσουν οποιοδήποτε διακριτό μετασχηματισμό από ένα πεπερασμένο χώρο διαστάσεων σε έναν άλλο. Πολλές μελέτες απέδειξαν το universal approximation theorem [Les+93; Cyb89; HSW89] για διάφορες συναρτήσεις ενεργοποίησης με τελευταία την rectified linear unit ReLU που χρησιμοποιείται τα τελευταία χρόνια.

Ωστόσο, το universal approximation theorem εγγυάται μόνο ότι μπορούμε να αναπαραστήσουμε οποιαδήποτε συνάρτηση. Δεν μας εγγυάται ότι θα μπορέσουμε τελικά να μάθουμε τη συνάρτηση. Το δεύτερο μπορεί να αποτύχει για δύο λόγους:

- Ο αλγόριθμος βελτιστοποίησης που θα χρησιμοποιήσουμε μπορεί να μην καταφέρει να βρει τις παραμέτρους που αναπαριστούν την συγκεκριμένη συνάρτηση
- Ο αλγόριθμος βελτιστοποίησης μπορεί να διαλέξει λάθος συνάρτηση, λόγω overfitting.

Το δεύτερο επιβεβαιώνει και το No free lunch θεώρημα 2.1.1, με την έννοια ότι αν δεν ίσχυε, θα είχαμε έναν αλγόριθμο όπου για κάθε training set θα έβρισκε τη σωστή συνάρτηση που θα γενίκευε καλύτερα στο test set. Ένα πρόβλημα, επίσης, είναι ότι ο αριθμός των νευρώνων που μπορεί να χρειαστούν στο πρώτο επίπεδο είναι εκθετικός ως προς τη διάσταση της εισόδου [Mon+14].

Έχει όμωςδειχτεί ότι στοιβάζοντας στρώματα νευρώνων στη σειρά υπάρχουν οικογένειες συναρτήσεων που για βάθος d χρειάζονται πολύ λιγότερες παραμέτρους για να αναπαρασταθούν από ότι αν το βάθος ήταν μικρότερο του d . Τα [Maa97; MSS94] αποδεικνύουν το παραπάνω για sigmoid activations. Επιπλέον, στο [Mon+14] αποδεικνύεται ότι για rectifier nonlinearities (όπως η relu ή η απόλυτη τιμή) μπορούμε να αναπαραστήσουμε συναρτήσεις με αριθμό περιοχών εκθετικό ως προς το βάθος του δικτύου. Μία γεωμετρική αναπαράσταση για συνάρτηση την απόλυτη τιμή δίνεται στο Σχήμα 2.5.1. Κάθε hidden unit καθορίζει πού θα διπλωθεί ο χώρος. Συνδυάζοντάς τέτοιους μετασχηματισμούς έχουμε εκθετικά μεγάλο αριθμό από piecewise linear περιοχές [GBC16].



Σχήμα 2.5.1: Κάθε hidden unit καθορίζει πού θα διπλωθεί ο χώρος. Συνδυάζοντάς τέτοιους μετασχηματισμούς έχουμε εκθετικά μεγάλο αριθμό από piecewise linear περιοχές [GBC16].

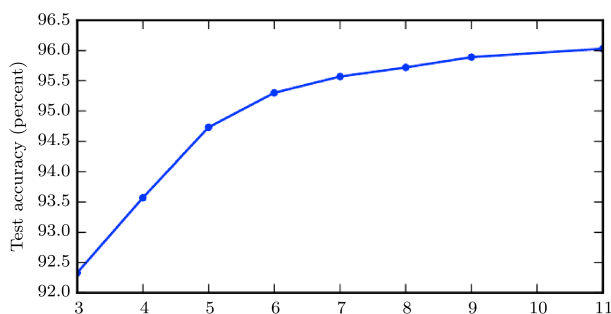
Τέλος, μπορεί επίσης να θέλουμε να επιλέξουμε ένα βαθύ μοντέλο για στατιστικούς λόγους. Η ιεραρχική αναπαράσταση που περιγράφεται στο κεφάλαιο 2.6, περιγράφει αναλυτικά τι στατιστικές υποθέσεις κάνουμε για τα δεδομένα.

2.6 Ιεραρχική αναπαράσταση της εισόδου

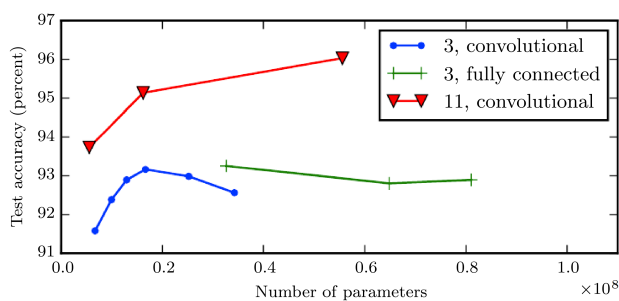
Οποτεδήποτε επιλέγουμε έναν συγκεκριμένο αλγόριθμο μηχανικής μάθησης, δηλώνουμε εμμέσως ένα σύνολο πεποιθήσεων που έχουμε σχετικά με το είδος της συνάρτησης που πρέπει να μάθουμε. Η επιλογή ενός μοντέλου με πολλά στρώματα εμπεριέχει μια πολύ γενική πεποίθηση ότι η συνάρτηση που θέλουμε να μάθουμε, περιλαμβάνει σύνθεση πολλών απλούστερων συναρτήσεων. Η βασική ιδέα πίσω από αυτό είναι ότι και τα δεδομένα δημιουργήθηκαν από τη σύνθεση χαρακτηριστικών.

Αναλογιστείτε τους εξής τύπους δεδομένων και τις συναρτήσεις κατανομής που τα δημιούργησαν. Παρακάτω εξηγείται διαισθητικά, η ιεραρχική φύση τους:

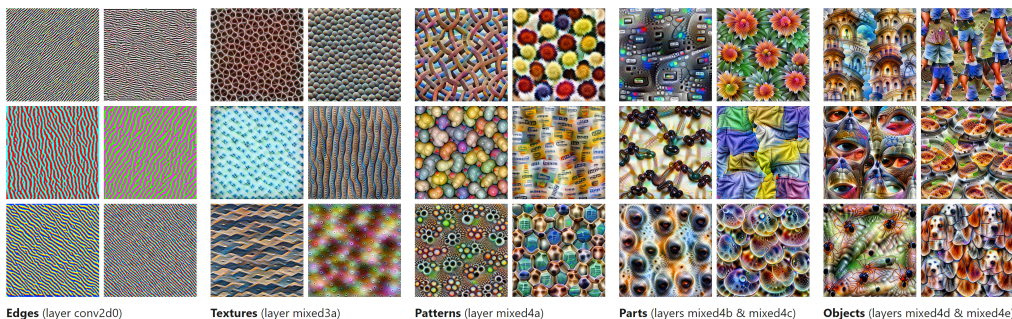
¹Συγκεκριμένα συνεχή συνάρτηση σε ένα κλειστό, πεπερασμένο σύνολο του \mathbb{R}^n



Σχήμα 2.5.2: Εμπειρικά αποτελέσματα που δείχνουν ότι τα βαθύτερα δίκτυα γενικεύουν καλύτερα. Το συγκεκριμένο πείραμα αφορά αναγνώριση πολυψήφιων αριθμών από φωτογραφίες διευθύνσεων [Goo+13]. Η ακρίβεια του συνόλου δοκιμής αυξάνεται σταθερά με την αύξηση του βάρους. Επίσης το Σχήμα 2.5.3 δείχνει ότι η αύξηση στο μέγεθος του μοντέλου δεν έχει το ίδιο αποτέλεσμα.



Σχήμα 2.5.3: Τα πιο βαθιά μοντέλα τείνουν να έχουν καλύτερη απόδοση. Αυτό δεν συμβαίνει μόνο επειδή το μοντέλο είναι μεγαλύτερο. Αυτό το πείραμα από τους Goodfellow, I. J. et al. [Goo+13] δείχνει ότι η αύξηση του αριθμού των παραμέτρων σε στρώματα συνελικτικών δικτύων χωρίς αύξηση του βάρους τους δεν έχει αποτέλεσμα στην αύξηση της απόδοσης του συνόλου δοκιμής. Στη λεζάντα γράφεται το βάθος του δικτύου που χρησιμοποιείται για τη δημιουργία κάθε καμπύλης και αν είναι συνελικτικό ή τα πλήρως συνδεδεμένο δίκτυο. Παρατηρούμε ότι ρηγά μοντέλα κάνουν overfit στις περίπου 20 εκατομμύρια παραμέτρους, ενώ τα βαθιά μπορούν να επωφεληθούν από την ύπαρξη πάνω από 60 εκατομμύρια παραμέτρων.



Σχήμα 2.6.1: Ιεραρχική δομή που μαθαίνει το δίκτυο και ταιριάζει με τη διαίσθησή μας για το από τι αποτελείται μία εικόνα [OMS17].

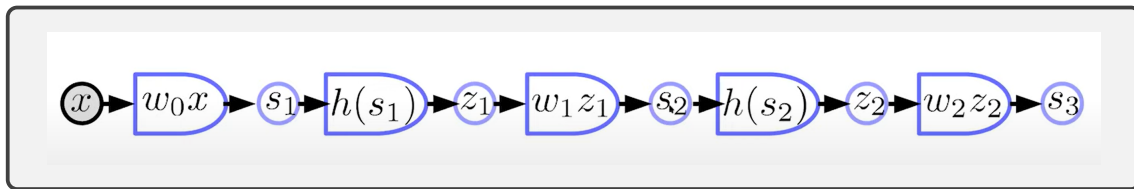
Παράδειγμα 2.6.1: Παραδείγματα ιεραρχικής φύσης σε δεδομένα

- **Εικόνες:** Τα pixels συγκροτούν ακμές, οι οποίες με τη σειρά τους δημιουργούν “textons”. Τα “textons” συνθέτουν μοτίβα, τα οποία δημιουργούν μέρη αντικειμένων, που τελικά συγκροτούν τα τελικά αντικείμενα. Στο Σχήμα 2.6.1 φαίνεται ότι σε κάθε στρώμα η αναπαράσταση που μαθαίνεται ταιριάζει με αυτή που περιγράψαμε.
- **Γλώσσα:** Ένα βιβλίο είναι η ένωση κεφαλαίων, παραγράφων, προτάσεων, φράσεων, γραμμάτων και τελικά χαρακτήρων.
- **Φωνή:** Ένα δείγμα ήχου είναι μία ακολουθία αριθμών. Το συχνотικό περιεχόμενο μιας κυματομορφής μπορεί να αναπαρασταθεί από ένα διάνυσμα χαρακτηριστικών (π.χ. MFCC’s), και αυτά να ενωθούν για να σχηματίσουν ήχους, οι οποίοι στη συνέχεια μπορούν να ενωθούν για να σχηματίσουν λέξεις κλπ.

2.7 Μετασχηματισμοί που συμβαίνουν σε κάθε στρώμα

Η παρούσα παράγραφος βασίζεται στο [Can]. Τα σχήματα έχουν προκύψει είτε μέσω του πακέτου [Hun07], είτε μέσω του [The22].

Ένα νευρωνικό δίκτυο στην πιο απλή του μορφή έχει τη μορφή εναλλαγής μεταξύ δύο μπλοκ: Αφινικών μπλοκ και μη γραμμικών μπλοκ. Παρακάτω δίνεται ένα διάγραμμα ενός τέτοιου δικτύου:



Σχήμα 2.7.1: Μπλοκ Διάγραμμα ενός απλού νευρωνικού δικτύου

Συγκεκριμένα τα αφινικά μπλοκ είναι συναρτήσεις της μορφής:

$$\mathbf{y} = f(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{b}.$$

Το οποίο ισοδύναμα μπορεί να γραφτεί:

$$\begin{bmatrix} \mathbf{y} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}$$




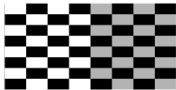


Το παραπάνω ονομάζεται Αφινικός Μετασχηματισμός και διατηρεί τις γραμμές και τον παραλληλισμό (αλλά όχι απαραίτητα αποστάσεις και γωνίες). Οπότε, τα αφινικά μπλοκ μπορούν να γραφούν:

$$\mathbf{s}_{k+1} = \mathbf{W}_k \mathbf{z}_k \quad (2.7.1)$$

Και τα μη γραμμικά:

$$\mathbf{z}_k = h(\mathbf{s}_k) \quad (2.7.2)$$

Στο διάγραμμα 2.7.1 και τις εξισώσεις (2.7.1) και (2.7.2), το $x \in \mathbb{R}^n$ αντιπροσωπεύει το διάνυσμα εισόδου. Το $\mathbf{W}_k \in \mathbb{R}^{n_k \times n_{k-1}}$ αντιπροσωπεύει τον πίνακα ενός αφινικού μετασχηματισμού που αντιστοιχεί στο $k^{\text{οστό}}$ μπλοκ. Η συνάρτηση h ονομάζεται συνάρτηση ενεργοποίησης και αυτή η συνάρτηση σχηματίζει το μη γραμμικό μπλοκ του νευρωνικού δικτύου. Μετά από διαδοχικές εφαρμογές αφινικών και μη γραμμικών μπλοκ, το δίκτυο παράγει ένα διάνυσμα εξόδου $s_k \in \mathbb{R}^{n_k-1}$. Ανάλογα με τη μορφή του πίνακα μετασχηματισμού, ένας αφινικός μετασχηματισμός μπορεί να κατηγοριοποιηθεί σε 6 κατηγορίες που φαίνονται στον Πίνακα 2.2.

Ταυτοτικός πίνακας	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	
Μετατόπιση	$\begin{bmatrix} 1 & 0 & v_x < 0 \\ 0 & 1 & v_y = 0 \\ 0 & 0 & 1 \end{bmatrix}$	
Αντανάκλαση (αρνητική οριζουσα)	$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	
Κλιμάκωση (διαγώνιος πίνακας)	$\begin{bmatrix} c_x = 2 & 0 & 0 \\ 0 & c_y = 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	
Περιστροφή (ορθοκανονικός πίνακας)	$\begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$	
Διάτμηση	$\begin{bmatrix} 1 & c_x = 0.5 & 0 \\ c_y = 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	

Πίνακας 2.2: Μορφές που μπορεί να πάρει ο πίνακας ενός αφινικού μετασχηματισμού και οπτικοποίησή τους.

Παράδειγμα 2.7.1: Ταξινόμηση εικόνων

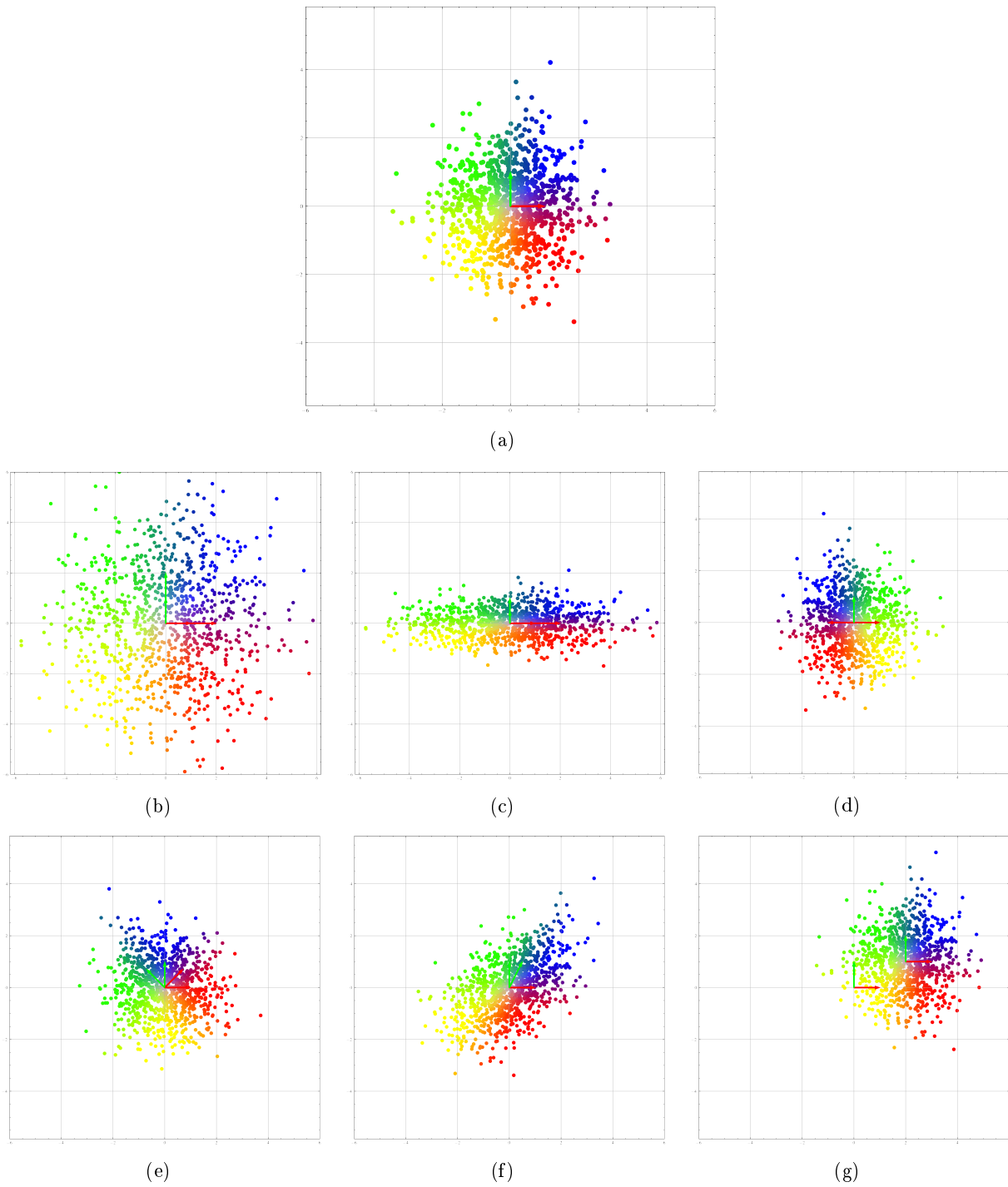
Ας υποθέσουμε τώρα, ότι τραβάμε χίλιες φωτογραφίες με κάμερα 1 megapixel. Κάθε φωτογραφία θα έχει περίπου 1.000 pixel κάθετα και 1.000 pixel οριζόντια και κάθε pixel θα έχει τρεις χρωματικές διαστάσεις για κόκκινο, πράσινο και μπλε (RGB). Οπότε, κάθε φωτογραφία μπορεί να θεωρηθεί ως ένα σημείο σε έναν χώρο 3 εκατομμυρίων διαστάσεων. **Είδαμε όμως ότι ο όγκος του χώρου αυξάνει εκθετικά με τις διαστάσεις άρα είναι πολύ πιθανό οι εικόνες αυτές να βρίσκονται σε ένα συγκεκριμένο σημείο του χώρου** (και να έχουν μεταξύ τους ίσες αποστάσεις όπως είδαμε από την κατάρτα της διαστατικότητας). Οπότε μια καλή αρχή σε κάθε τέτοιο πρόβλημα είναι να αφαιρέσουμε το μέσο των εικόνων και να διαιρέσουμε με την τυπική απόκλιση τους, ώστε να φέρουμε τις εικόνες στην αρχή του άξονα. Στη συνέχεια εφαρμόζουμε διαδοχικά τα παραπάνω μπλοκ με σκοπό να μετατρέψουμε τα δεδομένα μας σε γραμμικά διαχωρίσιμα ως προς το πρόβλημα που λύνουμε^a.

Για να δούμε τι συμβαίνει σε κάθε στρώμα αναπαριστούμε κάθε φωτογραφία με 2 συντεταγμένες, για παράδειγμα τις δύο πρώτες συνιστώσες του PCA [Pea01], οπότε έχουμε 1000 σημεία στο 2-διάστατο χώρο. Αρχικά, αφαιρούμε τον μέσο όρο των σημείων από κάθε σημείο και διαιρούμε με την τυπική τους απόκλιση, άρα έχουμε κέντρο την αρχή των αξόνων και ακτίνα περίπου 3 ($std = 1$). Οι μετασχηματισμοί που μπορούν να γίνουν από το αφινικό στρώμα φαίνονται στο Σχήμα 2.7.2. Βλέπουμε ότι η ιδιότητα του αφινικού μετασχηματισμού να διατηρεί τις γραμμές και τον παραλληλισμό μας εμποδίζει από το να μετασχηματίσουμε με αυθαίρετους τρόπους τα δεδομένα. Γι' αυτό χρησιμοποιούμε συναρτήσεις ενεργοποίησης.

Στο Σχήμα 2.7.3 φαίνεται η οπτικοποίηση ενός νευρωνικού δικτύου με δύο νευρώνες εισόδου, ένα κρυφό στρώμα με 3 νευρώνες και ReLU συνάρτηση ενεργοποίησης και 2 νευρώνες εξόδου. Μερικά σημεία που αξίζει να σταθούμε είναι τα εξής:

1. Βλέπουμε στο Σχήμα 2.7.3b πόσο λιγότερο χώρο καταλαμβάνουν τα δεδομένα στις τρεις διαστάσεις σε σχέση με τις δύο.
2. Στο Σχήμα 2.7.3c η ReLU μηδενίζει τις αρνητικές τιμές και ως προς τις 3 διαστάσεις.
3. Τέλος, στο Σχήμα 2.7.3d βλέπουμε ότι τα δεδομένα μας μετασχηματίστηκαν με τρόπο που δεν μπορούσαμε να πετύχουμε με αφινικούς μετασχηματισμούς, αφού στο Σχήμα 2.7.3d δεν διατηρήθηκαν οι παράλληλες γραμμές).

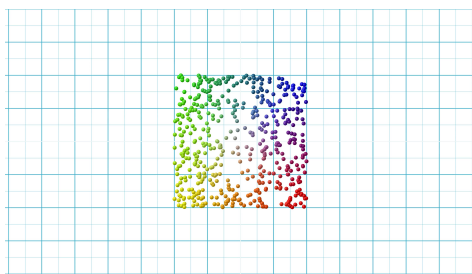
^aΣτην πραγματικότητα στα προβλήματα εικόνων είμαστε ήδη σε πολύ υψηλό αριθμό διαστάσεων άρα το να πάμε σε ακόμη μεγαλύτερες διαστάσεις κάνει το πρόβλημα αδύνατο υπολογιστικά. Οπότε χρησιμοποιούνται άλλου είδους στρώματα (συνελικτικά), τα οποία προσπερνούν αυτό το πρόβλημα μέσω του parameter sharing.



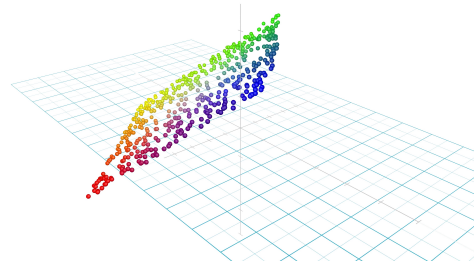
Σχήμα 2.7.2: Στο 2.7.2a φαίνονται τα αρχικά σημεία. Στη συνέχεια εφαρμόζονται με τη σειρά οι μετασχηματισμοί με τους αντίστοιχους πίνακες:

$$\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, \begin{bmatrix} 2 & 0 \\ 0 & 0.5 \end{bmatrix}, \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} \cos(45^\circ) & -\sin(45^\circ) \\ \sin(45^\circ) & \cos(45^\circ) \end{bmatrix}, \begin{bmatrix} 1 & 0.5 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

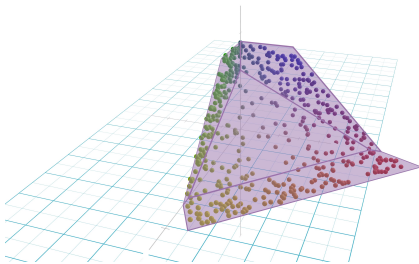
Φαίνονται επίσης τα διανύσματα βάσης πριν και μετά το μετασχηματισμό



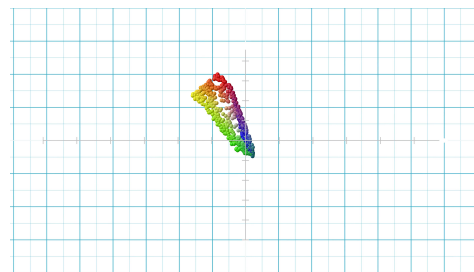
(a) Αρχικά σημεία



(b) Τυχαίος αφινικός μετασχηματισμός από τις δύο στις τρεις διαστάσεις



(c) Εφαρμογή ReLU



(d) Τυχαίος αφινικός μετασχηματισμός από τις τρεις στις δύο διαστάσεις

Σχήμα 2.7.3: Οπτικοποίηση ενός νευρωνικού δικτύου με δύο νευρώνες εισόδου, ένα κρυφό στρώμα με 3 νευρώνες και ReLU συνάρτηση ενεργοποίησης και 2 νευρώνες εξόδου.

Κεφάλαιο 3

Ιδιότητες των φυσικών σημάτων - Αρχιτεκτονικές Δικτύων

3.1	Ιδιότητες των φυσικών σημάτων	26
3.2	Ο πίνακας ενός πλήρως συνδεδεμένου δικτύου	27
3.3	Locality \Rightarrow sparsity	28
3.4	Stationarity \Rightarrow parameters sharing	29
3.4.1	Πίνακες Toeplitz	29
3.4.2	Στρώματα από Toeplitz πίνακες	29
3.5	Πλεονεκτήματα αρχιτεκτονικής	30
3.6	Σύνοψη CNN αρχιτεκτονικής	30
3.6.1	Pooling	31
3.6.2	Batch normalization και residual bypass connections	31
3.6.3	Padding	32
3.7	Transformers	32
3.7.1	Soft associative memories	32
3.7.2	Attention	33
3.7.3	Queries, Keys και Values	34
3.7.4	Multi-head attention	36
3.7.5	The Transformer	36
3.7.6	Encoder	37
3.7.7	Decoder	37
3.7.8	Positional encoding	39
3.7.9	Τα τελικά Linear και Softmax Layers	39

3.1 Ιδιότητες των φυσικών σημάτων

Η παρούσα παράγραφος βασίζεται στο [Can]. Όλα τα σχήματα προέρχονται από εκεί, εκτός και αν αναφέρεται διαφορετικά.

Στο προηγούμενο κεφάλαιο, είδαμε την ανάγκη να αναπαραστήσουμε την είσοδο σε χώρους πολλών διαστάσεων ή ισοδύναμα να αναπαραστήσουμε την είσοδο διάστασης n σε ένα χώρο διάστασης m , όπου $m > n$. Ο πιο απλός τρόπος για να γίνει αυτό είναι να δημιουργήσουμε m κόμβους και κάθε ένας από αυτούς να είναι γραμμικός συνδυασμός των n σημείων της εισόδου, όπως φαίνεται στο Σχήμα 3.3.1. Η αρχιτεκτονική αυτή ονομάζεται πλήρως-συνδεδεμένη (fully-connected). Όμως, τα σήματα που χρησιμοποιούμε στην πράξη είναι συνήθως ήδη πολλών διαστάσεων και αυτό κάνει το πρόβλημα αδύνατο. Πρέπει, λοιπόν, να βρούμε κάποιον τρόπο να βελτιώσουμε την πλήρως-συνδεδεμένη αρχιτεκτονική και να μειώσουμε τις παραμέτρους του δικτύου. Την κατεύθυνση για αυτό μας την δίνουν οι ίδιες οι ιδιότητες των φυσικών σημάτων. Πρώτα όμως πρέπει να ορίσουμε τι σημαίνει φυσικά σήματα.

Όλα τα σήματα μπορούν να αναπαρασταθούν ως διανύσματα. Για παράδειγμα ένα σήμα ήχου είναι ένα 1D διάνυσμα $\mathbf{x} = [x_1, x_2, \dots, x_T]$ όπου κάθε τιμή x_t αντιπροσωπεύει το πλάτος της κυματομορφής τη χρονική στιγμή t . Στη μουσική, για παράδειγμα, ένα στερεοφωνικό σήμα έχει 2 κανάλια αλλά το σήμα παραμένει μονοδιάστατο γιατί έχουμε μεταβολές μόνο ως προς τον άξονα του χρόνου. Μία εικόνα είναι ένα 2-διάστατο σήμα επειδή έχουμε μεταβολές και ως προς τους 2 άξονες. Κάθε σημείο μπορεί να είναι ένα διάνυσμα από μόνο του. Αυτό σημαίνει ότι αν έχουμε c κανάλια σε μια εικόνα, κάθε χωρικό σημείο στην εικόνα είναι ένα διάνυσμα της διάστασης c . Μια έγχρωμη εικόνα έχει επίπεδα RGB, που σημαίνει $c = 3$. Για οποιοδήποτε σημείο $x_{i,j}$, αυτό αντιστοιχεί στην ένταση των χρωμάτων κόκκινου, πράσινου και μπλε αντίστοιχα. Μια πρόταση της γλώσσας μπορεί να αναπαρασταθεί με τον παραπάνω τρόπο. Κάθε λέξη της γλώσσας είναι μία one-hot αναπαράσταση (διάνυσμα) όπου έχει 1 στην αντίστοιχη θέση του λεξιλογίου και 0 αλλού. Αυτό σημαίνει ότι κάθε λέξη έχει μήκος το μήκος του λεξιλογίου. Πιο τυπικά μπορούμε να σκεφτούμε κάθε φυσικό σήμα εισόδου ως ένα σύνολο X που αποτελείται από συναρτήσεις που κάνουν mapping το domain Ω στα κανάλια c .

Ορισμός 3.1.1: Τυπικός ορισμός φυσικών σημάτων

$$X = \{x^{(i)} \in \mathbb{R}^n | x^{(i)} \text{ ένα data sample} \}_{i=1}^m$$

$$X = \{x^{(i)} : \Omega \rightarrow \mathbb{R}^c, \omega \mapsto x^{(i)}(\omega)\}_{i=1}^m$$

όπου Ω το domain από όπου δημιουργήθηκαν τα δεδομένα και c ο αριθμός των καναλιών. Παραδείγματα:

Παράδειγμα 3.1.2: Παραδείγματα του παραπάνω mapping

- **Ήχος:**

$$\Omega = \{1, 2, \dots, T/\Delta t\} \subset \mathbb{N} \quad c \in \{1, 2, 5 + 1, \dots\}$$

όπου T ο συνολικός χρόνος Δt το sampling interval και $c = 1$: mono, $c = 2$: stereo, $c = 5 + 1$: Dolby 5.1.

- **Εικόνα:**

$$\Omega = \{1, \dots, h\} \times \{1, \dots, w\} \subset \mathbb{N}^2 \quad c \in \{1, 3, 20, \dots\}$$

όπου h το ύψος w το πλάτος και $c = 1$: grayscale, $c = 3$: RGB, $c = 20$: hyperspectral, κλπ.

Οι πιο σημαντικές ιδιότητες που συναντάμε στα φυσικά σήματα και οι οποίες μπορούν να χρησιμοποιηθούν για να βελτιώσουμε την fully connected αρχιτεκτονική είναι οι εξής:

Ορισμός 3.1.3: Locality

Υπάρχει ισχυρή τοπική συσχέτιση μεταξύ των τιμών. Αν πάρουμε δύο κοντινά pixels μιας φυσικής εικόνας, αυτά είναι πολύ πιθανό να έχουν το ίδιο χρώμα. Καθώς τα δύο pixel απομακρύνονται, η ομοιότητα μεταξύ τους μειώνεται. Άρα η τοπική συσχέτιση δημιουργεί την ανάγκη για τοπικές συνδέσεις.

Ορισμός 3.1.4: Stationarity

Διάφορα μοτίβα μπορούν να εμφανιστούν οπουδήποτε στην εικόνα, πράγμα που σημαίνει ότι πρέπει να επαναλάβουμε την ανίχνευση χαρακτηριστικών για κάθε θέση στην εικόνα εισόδου. Έτσι δικαιολογείται η ανάγκη για κοινά βάρη.

Ορισμός 3.1.5: Compositionality

Τα φυσικά σήματα έχουν ιεραρχική φύση, όπως εξηγήθηκε στην προηγούμενη παράγραφο. Αυτό δικαιολογεί τη χρήση πολλαπλών στρωμάτων νευρώνων.

3.2 Ο πίνακας ενός πλήρως συνδεδεμένου δικτύου

Έστω ότι έχουμε ένα μόνο κρυφό στρώμα h :

$$h = f(z)$$

Η έξοδος είναι μια μη γραμμική συνάρτηση $f(\cdot)$ που εφαρμόζεται σε ένα διάνυσμα z . Εδώ το z είναι η έξοδος ενός αφινικού μετασχηματισμού A στο διάνυσμα εισόδου x :

$$z = Ax$$

$$Ax = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} -\mathbf{a}^{(1)}- \\ -\mathbf{a}^{(2)}- \\ \vdots \\ -\mathbf{a}^{(m)}- \end{pmatrix} \Big| \mathbf{x} = \begin{pmatrix} \mathbf{a}^{(1)}\mathbf{x} \\ \mathbf{a}^{(2)}\mathbf{x} \\ \vdots \\ \mathbf{a}^{(m)}\mathbf{x} \end{pmatrix}_{m \times 1}$$

Όπου $\mathbf{a}^{(i)}$ είναι i -ή σειρά του πίνακα A .

Για να καταλάβουμε το νόημα αυτού του μετασχηματισμού, ας πάρουμε ένα μέρος του z και αναλύσουμε το $\mathbf{a}^{(1)}\mathbf{x}$. Έστω επίσης $n = 2$, τότε $\mathbf{a} = (a_1, a_2)$ και $\mathbf{x} = (x_1, x_2)$. Τα \mathbf{a} και \mathbf{x} είναι διανύσματα του διδιάστατου επιπέδου. Αν η γωνία μεταξύ του \mathbf{a} και \hat{i} είναι α και η γωνία μεταξύ \mathbf{x} και \hat{i} είναι ξ , τότε η $\mathbf{a}^\top \mathbf{x}$ γράφεται:

$$\begin{aligned} \mathbf{a}^\top \mathbf{x} &= a_1 x_1 + a_2 x_2 \\ &= \|\mathbf{a}\| \cos(\alpha) \|\mathbf{x}\| \cos(\xi) + \|\mathbf{a}\| \sin(\alpha) \|\mathbf{x}\| \sin(\xi) \\ &= \|\mathbf{a}\| \|\mathbf{x}\| (\cos(\alpha) \cos(\xi) + \sin(\alpha) \sin(\xi)) \\ &= \|\mathbf{a}\| \|\mathbf{x}\| \cos(\xi - \alpha) \end{aligned}$$

Η έξοδος μετρά την ευθυγράμμιση της εισόδου σε μια συγκεκριμένη γραμμή του πίνακα A . Αυτό μπορεί να γίνει κατανοητό παρατηρώντας τη γωνία μεταξύ των δύο διανυσμάτων, $\xi - \alpha$. Όταν $\xi = \alpha$, τα δύο διανύσματα είναι τέλεια ευθυγραμμισμένα και επιτυγχάνεται το μέγιστο. Αν $\xi - \alpha = \pi$, τότε $\mathbf{a}^\top \mathbf{x}$ φτάνει στο ελάχιστο και τα δύο διανύσματα δείχνουν προς αντίθετες κατευθύνσεις. Στην ουσία, ο γραμμικός μετασχηματισμός επιτρέπει σε κάποιον να δει την προβολή μιας εισόδου σε διάφορους προσανατολισμούς που εμπεριέχονται στον A . Αυτή η διαίσθηση μπορεί να επεκταθεί και σε υψηλότερες διαστάσεις.

Ένας ακόμη τρόπος να καταλάβουμε τον γραμμικό μετασχηματισμό είναι:

$$Ax = \begin{pmatrix} | & | & \cdots & | \\ \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_n \\ | & | & & | \end{pmatrix} \mathbf{x} = x_1 \mathbf{a}_1 + x_2 \mathbf{a}_2 + \cdots + x_n \mathbf{a}_n$$

Η έξοδος, δηλαδή, είναι το weighted sum των στηλών του πίνακα \mathbf{A} . Άρα η έξοδος είναι σύνθεση της εισόδου. Ας δούμε τώρα πώς αυτός ο πίνακας παίρνει ειδική μορφή αξιοποιώντας τις ιδιότητες των φυσικών σημάτων:

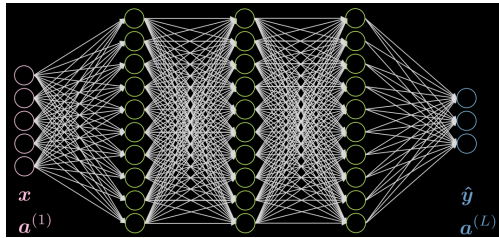
3.3 Locality \Rightarrow sparsity

Το Σχήμα 3.3.1 δείχνει ένα πλήρως συνδεδεμένο δίκτυο. Κάθε βέλος αντιπροσωπεύει ένα βάρος που πρέπει να πολλαπλασιαστεί με τις εισόδους. Όπως μπορούμε να δούμε, όσο μεγαλώνει η είσοδος τόσο μεγαλύτερο πίνακα θέλουμε αφού υπάρχουν παντού συνδέσεις.

Άρα ξεκινάμε με έναν πίνακα με μεγάλο αριθμό στηλών:

$$\begin{bmatrix} w_{11} & w_{12} & w_{13} & w_{14} & \cdots & w_{1k} & \cdots & w_{1n} \\ w_{21} & w_{22} & w_{23} & w_{24} & \cdots & w_{2k} & \cdots & w_{2n} \\ w_{31} & w_{32} & w_{33} & w_{34} & \cdots & w_{3k} & \cdots & w_{3n} \\ w_{41} & w_{42} & w_{43} & w_{44} & \cdots & w_{4k} & \cdots & w_{4n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_k \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ \vdots \\ y_k \\ \vdots \\ y_n \end{bmatrix}$$

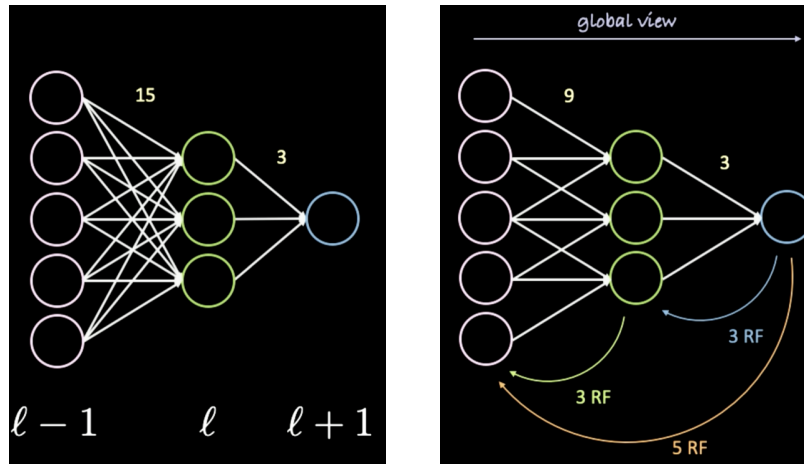
Επίσης, εάν κάνουμε τυχαία αναδιάταξη στην είσοδο η έξοδος του πλήρως-συνδεδεμένου δικτύου δεν θα επηρεαστεί. Αυτό μας δείχνει ότι μερικά βάρη του δικτύου είναι περιττά, όπως θα δούμε παρακάτω.



Σχήμα 3.3.1: Πλήρως συνδεδεμένο δίκτυο

Εάν τα δεδομένα μας παρουσιάζουν locality, κάθε νευρώνας πρέπει να συνδεθεί μόνο με λίγους τοπικούς νευρώνες του προηγούμενου στρώματος. Έτσι, ορισμένες συνδέσεις δεν χρειάζονται, όπως φαίνεται στο Σχήμα 3.3.2. Αριστερά έχουμε ένα πλήρως-συνδεδεμένο δίκτυο. Εκμεταλλευόμενοι την ιδιότητα του locality, σβήνουμε συνδέσεις μεταξύ μακρινών νευρώνων (δεξιά εικόνα). Αν και οι νευρώνες του κρυφού στρώματος (πράσινοι) στο Σχήμα 3.3.2 δεν καλύπτουν ολόκληρη την είσοδο, η συνολική αρχιτεκτονική θα μπορεί να λαμβάνει υπόψη όλους τους νευρώνες εισόδου. Το receptive field (RF) είναι ο αριθμός των νευρώνων των προηγούμενων στρωμάτων, που μπορεί να λάβει υπόψη κάθε νευρώνας ενός συγκεκριμένου στρώματος. Επομένως, το RF του στρώματος εξόδου προς το κρυφό στρώμα είναι 3, το RF του κρυφού στρώματος με το επίπεδο εισόδου είναι 3, αλλά το RF του στρώματος εξόδου με το στρώμα εισόδου είναι 5, δηλαδή όλη η είσοδος. Αντίστοιχα στον πίνακα δεν μας ενδιαφέρουν μακρινά σημεία άρα τα w_{1k} μπορούν να γίνουν 0 για μεγάλο k . Άρα η πρώτη στήλη του πίνακα μπορεί να γίνει π.χ. ένας kernel μήκους 3. Ας ονομάσουμε τον kernel: $\mathbf{a}^{(1)} = \begin{bmatrix} a_1^{(1)} & a_2^{(1)} & a_3^{(1)} \end{bmatrix}$.

$$\begin{bmatrix} a_1^{(1)} & a_2^{(1)} & a_3^{(1)} & 0 & \cdots & 0 & \cdots & 0 \\ w_{21} & w_{22} & w_{23} & w_{24} & \cdots & w_{2k} & \cdots & w_{2n} \\ w_{31} & w_{32} & w_{33} & w_{34} & \cdots & w_{3k} & \cdots & w_{3n} \\ w_{41} & w_{42} & w_{43} & w_{44} & \cdots & w_{4k} & \cdots & w_{4n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_k \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ \vdots \\ y_k \\ \vdots \\ y_n \end{bmatrix}$$



Σχήμα 3.3.2: **Αριστερά:** Βλέπουμε ένα πλήρως-συνδεδεμένο δίκτυο. **Δεξιά:** Βλέπουμε το ίδιο δίκτυο αν εκμεταλλευτούμε το locality. Φαίνονται επίσης και τα receptive field (RF) των νευρώνων.

3.4 Stationarity \Rightarrow parameters sharing

3.4.1 Πίνακες Toeplitz

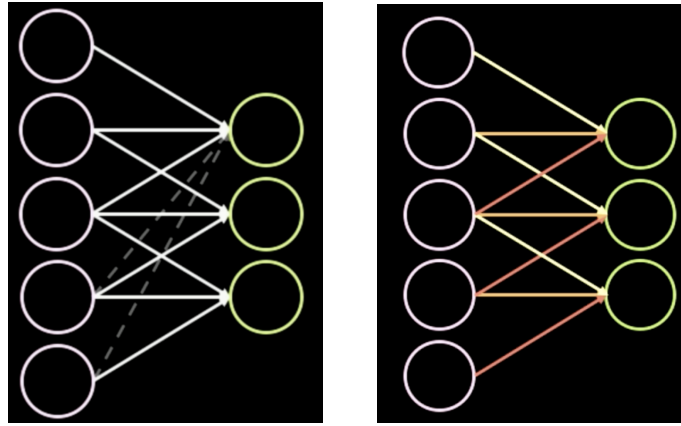
Εάν τα δεδομένα μας παρουσιάζουν Stationarity, θα μπορούσαμε να επαναχρησιμοποιήσουμε ένα μικρό σύνολο παραμέτρων πολλές φορές στην αρχιτεκτονική του δικτύου. Για παράδειγμα, στο sparse δίκτυό του προηγούμενου βήματος, μπορούμε να χρησιμοποιήσουμε ένα σύνολο 3 κοινών παραμέτρων (Σχήμα 3.4.1) (κίτρινο, πορτοκαλί και κόκκινο). Έτσι, ο αριθμός των παραμέτρων θα πέσει από 9 σε 3. Έτσι έχουμε επίσης περισσότερα δεδομένα για την εκπαίδευση, ως προς τον αριθμό των βαρών. Τα τελικά βάρη μετά την εφαρμογή του sparsity και του parameters sharing ονομάζονται πυρήνας συνέλιξης. Άρα, μπορούμε στον πίνακα να επαναχρησιμοποιήσουμε τον kernel $\mathbf{a}^{(1)}$ π.χ. με βήμα 1:

$$\begin{bmatrix} a_1^{(1)} & a_2^{(1)} & a_3^{(1)} & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & a_1^{(1)} & a_2^{(1)} & a_3^{(1)} & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & a_1^{(1)} & a_2^{(1)} & a_3^{(1)} & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & a_1^{(1)} & a_2^{(1)} & a_3^{(1)} & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & a_1^{(1)} & a_2^{(1)} & a_3^{(1)} & \cdots & 0 \\ \vdots & & \vdots & & \vdots & & \vdots & & \vdots \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_k \\ \vdots \\ x_n \end{bmatrix}$$

Αυτός ο τύπος πίνακα ονομάζεται πίνακας Toeplitz. Σε κάθε πίνακα Toeplitz, κάθε φθίνουσα διαγώνιος από αριστερά προς τα δεξιά είναι σταθερή. Οι πίνακες Toeplitz που χρησιμοποιούμε εδώ είναι επίσης αραιοί πίνακες.

3.4.2 Στρώματα από Toeplitz πίνακες

Μετά την τελευταία αλλαγή έχουμε μόλις 3 παραμέτρους (a_1, a_2, a_3), ενώ μετά την αξιοποίηση του locality είχαμε 9 (Σχήμα 3.3.2). Αυτό μας δίνει το περιθώριο να αυξήσουμε τον αριθμό παραμέτρων του δικτύου. Έτσι, αν το παραπάνω θεωρηθεί, ως convolutional στρώμα με kernel $\mathbf{a}^{(1)}$, μπορούμε να στοιβάξουμε πολλά τέτοια στρώματα $\mathbf{a}^{(2)}, \mathbf{a}^{(3)} \dots$, για να αυξήσουμε τις παραμέτρους του δικτύου.



Σχήμα 3.4.1: **Αριστερά:** Αμέσως μετά την εφαρμογή του sparsity **Δεξιά:** Αμέσως μετά την εφαρμογή του Parameter Sharing.

3.5 Πλεονεκτήματα αρχιτεκτονικής

Μερικά πλεονεκτήματα που προσφέρουν το sparsity και το parameter sharing είναι:

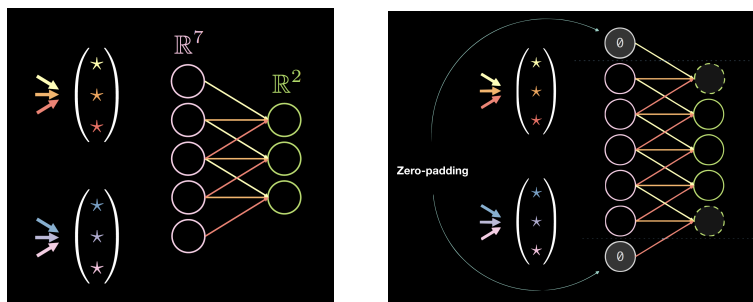
Parameter sharing

- Ταχύτερη σύγκλιση του αλγόριθμου βελτιστοποίησης.
- Δεν υπάρχει περιορισμός στο μέγεθος εισόδου.
- Kernel independence, οπότε υψηλή παραλληλοποίηση στο hardware, με χρήση κάρτας γραφικών.

Connection sparsity

- Λιγότεροι υπολογισμοί, λόγω των πολλών μηδενικών στον πίνακα βαρών.

Στο Σχήμα 3.5.1 φαίνεται η εφαρμογή τέτοιων πυρήνων σε μονοδιάστατα δεδομένα. Η επιλογή του μεγέθους του πυρήνα είναι εμπειρική. Η συνέλιξη 3×3 φαίνεται να είναι το ελάχιστο μέγεθος για χωρικά δεδομένα. Η συνέλιξη μεγέθους 1 μπορεί να χρησιμοποιηθεί πριν το τελικό στρώμα. Αν το μέγεθος πυρήνα είναι ζυγός αριθμός μπορεί να δημιουργηθεί θόρυβος, επομένως έχουμε πάντα μέγεθος πυρήνα περιττό αριθμό.



Σχήμα 3.5.1: **Αριστερά:** Εφαρμογή kernels σε 1D δεδομένα. **Δεξιά:** Το ίδιο με Zero Padding.

3.6 Σύνοψη CNN αρχιτεκτονικής

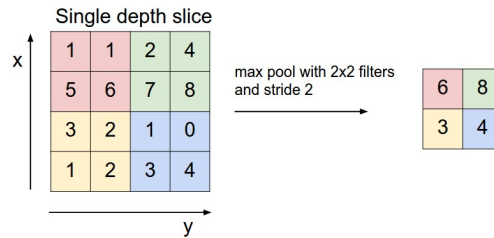
Ένα συνηθισμένο CNN ανεξαρτήτως της διάστασης εισόδου, αποτελείται από στρώματα στρώματα με:

- Convolution kernels
- Non-linearity (ReLU ή Leaky ReLU)
- Pooling

- Batch normalisation
- Residual bypass connection

Τα τρία τελευταία εξηγούνται στις παραγράφους 3.6.1 και 3.6.2:

3.6.1 Pooling



Σχήμα 3.6.1: Οπτικοποίηση του Pooling operator.

Βασικό συστατικό του pooling είναι ένας τελεστής, ο οποίος κάνει κάποιου είδους aggregate σε ένα κομμάτι της εισόδου, και είναι permutation invariant, δηλαδή αν ανακατέψουμε την είσοδο μέσα στο παράθυρο εφαρμογής, το αποτέλεσμα θα είναι ίδιο. Μία τέτοια συνάρτησή είναι π.χ. η L_p -norm (πιο συνηθισμένα $p = +\infty$ άρα $L_p = \max$), η οποία εφαρμόζεται σε ένα παράθυρο του σήματος όπως φαίνεται στο Σχήμα 3.6.1. Στη συνέχεια επαναλαμβάνουμε διαδοχικά για το σύνολο του σήματος ανά περιοχή, μετακινούμενοι με σταθερό αριθμό βημάτων (stride). Αν ξεκινήσουμε με $m*n$ δεδομένα με c κανάλια, θα καταλήξουμε με $\frac{m}{2} * \frac{n}{2}$ δεδομένα ακόμα με c κανάλια. Ο κύριος σκοπός του pooling είναι ότι μειώνει τον όγκο των δεδομένων και κυρίως **εξαλείφει ορισμένες πληροφορίες σχετικά με την ακριβή θέση όπου εμφανίζεται ένα συγκεκριμένο μοτίβο**. Έτσι το pooling βοηθά να γίνει η τελική αναπαράσταση approximately invariant σε μικρές μετατοπίσεις της εισόδου.

Πιο συγκεκριμένα δίνεται ο παρακάτω ορισμός:

Ορισμός 3.6.1: Invariance και equivariance

Για δύο μετασχηματισμούς $f(\cdot)$ και $T(\cdot)$ ορίζουμε:

- Ο f είναι invariant ως προς τον T αν:

$$f(T(x)) = f(x)$$

- Ο f είναι equivariant ως προς τον T αν:

$$f(T(x)) = T'f(x)$$

Σύμφωνα με τον ορισμό 3.6.1 έχουμε:

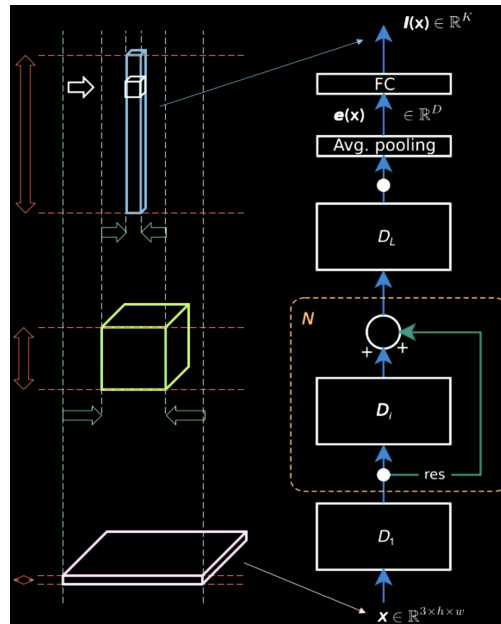
- Η συνέλιξη είναι equivariant ως προς τη μετατόπιση
- Το pooling είναι locally (approximately) invariant ως προς τη μετατόπιση

3.6.2 Batch normalization και residual bypass connections

Το Batch normalization και οι residual bypass connections είναι πολύ χρήσιμα για την καλή εκπαίδευση του δικτύου. Διάφορα μέρη ενός σήματος μπορούν να χαθούν εάν έχουν στοιβαχτεί πάρα πολλά στρώματα λόγω των πολλαπλασιασμών με μικρά βάρη. Οι residual συνδέσεις, εγγυώνται μια διαδρομή που εγγυάται ότι το σήμα θα φτάσει χωρίς πολλές τροποποιήσεις στην έξοδο (και το ίδιο για τα gradients στην αντίστροφη διαδρομή).

Στο Σχήμα 3.6.2, βλέπουμε ότι η πληροφορία στην εικόνα εισόδου είναι διάσπαρτη ως προς το πλάτος και το μήκος της εικόνας. Η μόνη πληροφορία στο ύψος της είναι τα διαφορετικά χρώματα. Άρα το παραλληλό-

γραμμο είναι λεπτό ως προς το ύψος του. Αντίθετα, καθώς ανεβαίνουμε στην ιεραρχία, παίρνουμε πιο πυκνή αναπαράσταση καθώς χάνουμε τις χωρικές πληροφορίες.



Σχήμα 3.6.2: Στο σχήμα βλέπουμε ότι η μετατρέπεται από χωρική, δηλαδή διάσπαρτη ως προς το πλάτος και μήκος της εικόνας, γίνεται μία πυκνή αναπαράσταση από features.

3.6.3 Padding

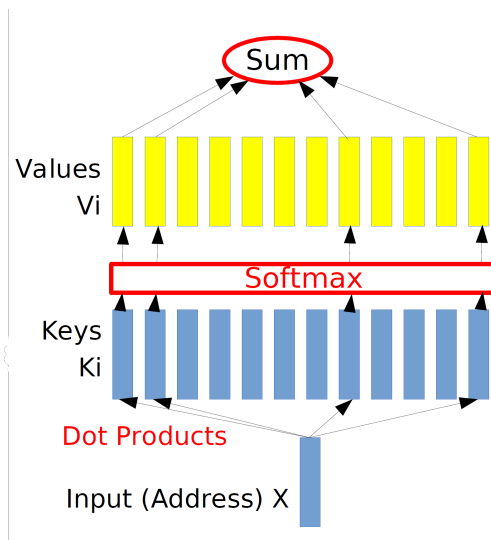
Τέλος, μια τεχνική που χρησιμοποιείται συχνά είναι το padding. Το padding γενικά βλάπτει τα τελικά αποτελέσματα, αφού προσθέτουμε πληροφορία η οποία είναι θόρυβος (μηδενικά), αλλά είναι βολικό προγραμματιστικά. Συνήθως χρησιμοποιούμε τον τύπο: $\text{size} = \frac{\text{μέγεθος πυρήνα} - 1}{2}$.

3.7 Transformers

Τα fully-connected και τα convolutional δίκτυα που περιγράφηκαν στις παραγράφους 3.6 και 3.2, είναι πολύ καλά στην “κατανόηση των δεδομένων” (perception) π.χ. αναγνώριση εικόνας, εξαγωγή ιεραρχικής αναπαράστασης των δεδομένων κλπ. Αυτό προκύπτει από τη feed-forward μορφή τους, αφού πρώτα εκπαιδεύονται στα δεδομένα και σε δεύτερο χρόνο (κατά το inference) βγάζουν συμπεράσματα πάνω σε αυτά με ένα μόνο πέρασμα. Αυτό όμως δεν αρκεί για να εκτελεστεί μία διεργασία (task) η οποία απαιτεί συλλογισμό. Όταν εκτελούμε σαν άνθρωποι ένα task, έχουμε μία μνήμη εργασίας, δρούμε με βάση αυτή, την ανανεώνουμε και με βάση αυτή σχεδιάζουμε την επόμενη κίνηση. Άρα, λοιπόν, θέλουμε να επεκτείνουμε τις μέχρι τώρα αρχιτεκτονικές, προσθέτοντας κάποιου είδους μνήμη.

3.7.1 Soft associative memories

Ένα κομμάτι δικτύου που προσομοιάζει ένα είδος μνήμης φαίνεται στο Σχήμα 3.7.1 και θυμίζει πολύ μία συνηθισμένη RAM. Η είσοδος είναι μία διεύθυνση, η οποία συγκρίνεται με όλες τις διευθύνσεις (κλειδιά) και όταν γίνει match με κάποιο κλειδί δίνει στην έξοδο την τιμή της μνήμης στην αντίστοιχη διεύθυνση που έγινε το match. Στο πλαίσιο των νευρωνικών δικτύων το match γίνεται με βάση κάποια συνάρτηση ομοιότητας (π.χ. εσωτερικό γινόμενο) ενώ μπορούμε να έχουμε ένα (hard) ή περισσότερα (soft) matches. Στη δεύτερη περίπτωση παίρνουμε τον γραμμικό συνδυασμό των αντίστοιχων τιμών με βάση το βάρος που έδωσε το μοντέλο σε κάθε τιμή. Η παραπάνω αρχιτεκτονική είναι διαφορίσιμη ως προς τις τιμές v_i αλλά και ως προς τα κλειδιά k_i , άρα και τα δύο μπορούν να εκπαιδευτούν.



Σχήμα 3.7.1: Soft associative memory

Εισάγουμε την έννοια του attention πριν μιλήσουμε για την αρχιτεκτονική του Transformer. Υπάρχουν δύο κύριοι τύποι attention: **self attention** και **cross attention**, και για κάθε μία από αυτές τις κατηγορίες, μπορούμε να έχουμε **hard attention** και **soft attention**.

Όπως θα δούμε αργότερα, οι transformers αποτελούνται από μονάδες attention, οι οποίες είναι αντιστοιχίσεις μεταξύ συνόλων, αντί για ακολουθίες, πράγμα που σημαίνει ότι δεν επιβάλλουμε μια σειρά στις εισόδους/εξόδους μας. Πιο τυπικά είναι μοντέλα “equivariant to permutation”, δηλαδή για αναδιατεταγμένη είσοδο η έξοδος θα έχει την ίδια αναδιάταξη.

3.7.2 Attention

Έστω σύνολο με t εισόδους \mathbf{x}_i :

$$\{\mathbf{x}_i\}_{i=1}^t = \{\mathbf{x}_1, \dots, \mathbf{x}_t\}$$

όπου κάθε \mathbf{x}_i είναι n -διάστατο διάνυσμα. Αφού το σύνολο έχει t στοιχεία, όπου το καθένα ανήκει στο \mathbb{R}^n , μπορούμε να τα στοιβάξουμε σε στήλες ενός πίνακα $\mathbf{X} \in \mathbb{R}^{n \times t}$.

Ορισμός 3.7.1: Attention

Ονομάζουμε attention, το κομμάτι δικτύου με έξοδο το representation h . Το h είναι γραμμικός συνδυασμός των εισόδων (στηλών):

$$\alpha_1 \mathbf{x}_1 + \alpha_2 \mathbf{x}_2 + \dots + \alpha_t \mathbf{x}_t$$

Απορούμε να γράψουμε το παραπάνω ως γινόμενο πινάκων:

$$\mathbf{X}\boldsymbol{\alpha} = \begin{pmatrix} | & | & \dots & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_t \\ | & | & \dots & | \end{pmatrix} \boldsymbol{\alpha}$$

όπου $\boldsymbol{\alpha} \in \mathbb{R}^t$ διάνυσμα στήλη με στοιχεία τα α_i .

Βλέπουμε ότι αυτό είναι διαφορετικό από το hidden representation που παράγει π.χ. ένα απλό feed-forward δίκτυο, όπου πολλαπλασιάζει instance της εισόδου με έναν πίνακα από βάρη. Εδώ διαλέγουμε (δίνουμε attention) σε διάφορα σημεία της εισόδου ανάλογα με κάποιο βάρος.

- **Feed forward:**

$$\mathbf{A}\mathbf{x} = \begin{pmatrix} | & | & \cdots & | \\ \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_n \\ | & | & & | \end{pmatrix} \mathbf{x} = x_1 \mathbf{a}_1 + x_2 \mathbf{a}_2 + \cdots + x_n \mathbf{a}_n$$

- **Attention:**

$$\mathbf{X}\boldsymbol{\alpha} = \begin{pmatrix} | & | & \cdots & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_t \\ | & | & & | \end{pmatrix} \boldsymbol{\alpha} = \alpha_1 \mathbf{x}_1 + \alpha_2 \mathbf{x}_2 + \cdots + \alpha_t \mathbf{x}_t$$

Ανάλογα με τους περιορισμούς που επιβάλλουμε στο διάνυσμα \mathbf{a} , μπορούμε να πετύχουμε hard ή soft attention.

Ορισμός 3.7.2: Hard Attention

Hard-attention σημαίνει ότι επιβάλλουμε $\|\mathbf{a}\|_0 = 1$. Αυτό σημαίνει ότι το \mathbf{a} είναι ένα one-hot διάνυσμα. Άρα το hidden representation είναι το κομμάτι της εισόδου \mathbf{x}_i που αντιστοιχεί στο $\alpha_i = 1$.

Ορισμός 3.7.3: Soft Attention

Soft attention σημαίνει ότι επιβάλλουμε $\|\mathbf{a}\|_1 = 1$. Άρα το hidden representation είναι γραμμικός συνδυασμός των εισόδων με συντελεστές που αθροίζονται στη μονάδα.

Είδαμε από το Σχήμα 3.7.1 ότι τα α_i προκύπτουν από μία συνάρτηση ομοιότητας της εισόδου με όλες τις άλλες εισόδους (π.χ. additive attention [BCB16] που υπολογίζει την συνάρτηση ομοιότητας μέσω feed-forward δικτύου με ένα hidden layer). Ωστόσο, στην πράξη κυρίως χρησιμοποιείται η scaled-dot-product attention αφού είναι space-efficient και πιο γρήγορη λόγω των optimized αλγορίθμων για πολλαπλασιασμούς πινάκων. Στη συνέχεια καλούμε τη συνάρτηση softmax στο αποτέλεσμα για να μετατρέψουμε τα scores σε pseudoprobabilities. Άρα τελικά το $\mathbf{a} \in \mathbb{R}^t$ προκύπτει ως εξής:

$$\mathbf{a} = \text{softmax}_\beta(\mathbf{X}^\top \mathbf{x})$$

Όπου β αντιπροσωπεύει την αντίστροφη παράμετρο θερμοκρασίας της συνάρτησης softmax(\cdot). Όταν $\beta \rightarrow \infty$ η συνάρτηση είναι η απλή argmax(\cdot) και άρα έχουμε one-hot encoded έξοδο και hard attention. Συνήθως επιλέγουμε $\beta = \frac{1}{\sqrt{n}}$. Το γιατί εξηγείται στο ακόλουθο παράδειγμα:

Παράδειγμα 3.7.4: Variance εσωτερικού γινομένου ως προς αριθμό διαστάσεων

Έστω q και k ανεξάρτητες τυχαίες μεταβλητές με μέσο όρο 0 και variance 1. Το εσωτερικό γινόμενο τους $q \cdot k = \sum_{i=1}^{n_k} q_i k_i$, έχει μέσο όρο 0 και variance n_k .

Για μεγάλες τιμές του n η additive attention νικά την dot product attention αν δεν κάνουμε scale με το προτεινόμενο β [Bri+17]. Αυτό γιατί το εσωτερικό γινόμενο αυξάνει γραμμικά όσο αυξάνουν οι διαστάσεις με αποτέλεσμα η softmax να δίνει πολύ μικρά gradients.

Μία λύση είναι να διαρέσουμε με το $n = \|\mathbf{1}\|_2$, όπου $\mathbf{1} \in \mathbb{R}^n$.

Έτσι έχουμε παρόμοιο scaling ανεξάρτητα από τον αριθμό των διαστάσεων n και ως αποτέλεσμα πιο stable gradients.

Άρα προκύπτει ένα σύνολο από \mathbf{a} με κάθε στοιχείο του να αντιστοιχεί σε ένα \mathbf{x}_i . Κάθε $\mathbf{a}_i \in \mathbb{R}^t$, άρα μπορούμε να τα στοιβάζουμε σε έναν πίνακα $\mathbf{A} \in \mathbb{R}^{t \times t}$. Οπότε, το τελικό hidden state έχει διάσταση $\mathbf{H} \in \mathbb{R}^{n \times t}$ και:

$$\mathbf{H} = \mathbf{X}\mathbf{A}$$

3.7.3 Queries, Keys και Values

Μέχρι στιγμής το μοντέλο δεν έχει προς εκπαίδευση παραμέτρους και τα attention scores προκύπτουν με βάση την ομοιότητα μεταξύ των εισόδων. Σαν επόμενο βήμα μπορούμε να προβάλλουμε την είσοδο σε ένα χώρο, οι άξονές του οποίου να είναι εκπαιδευόμενοι. Αυτό ακριβώς κάνουμε στους ορισμούς 3.7.5 και 3.7.6:

Ορισμός 3.7.5: Self-Attention

Τα διανύσματα queries, keys και values προκύπτουν ως εξής:

$$\mathbf{q} = \mathbf{W}_q \mathbf{x}$$

$$\mathbf{k} = \mathbf{W}_k \mathbf{x}$$

$$\mathbf{v} = \mathbf{W}_v \mathbf{x}$$

Ορισμός 3.7.6: Cross-Attention

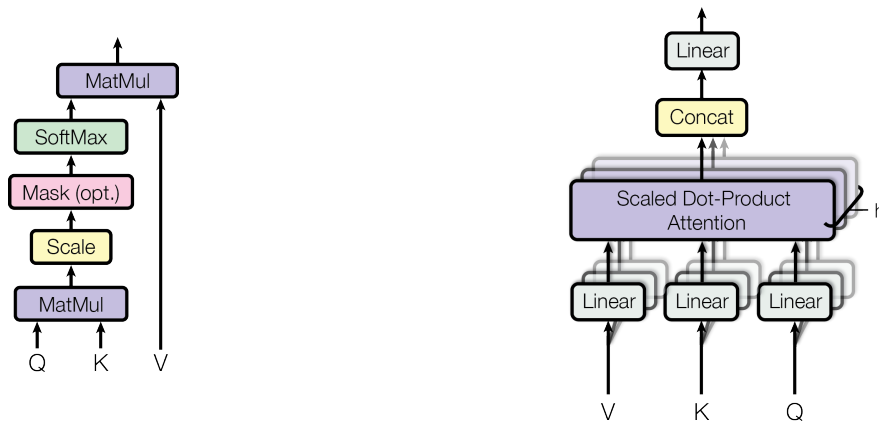
Τα διανύσματα queries, keys και values προκύπτουν ως εξής:

$$\mathbf{q} = \mathbf{W}_q \mathbf{y}$$

$$\mathbf{k} = \mathbf{W}_k \mathbf{h}_{enc}$$

$$\mathbf{v} = \mathbf{W}_v \mathbf{h}_{enc}$$

Βλέπουμε ότι και το self και το cross attention ακολουθούν τη μορφή του Σχήματος 3.7.2 (αριστερά). Η μόνη διαφορά είναι ότι στο cross attention το Query προκύπτει από ένα block (συνήθως του decoder) ενώ τα keys και values από άλλο (συνήθως η έξοδος του encoder).



Σχήμα 3.7.2: **Αριστερά:** Scaled Dot-Product Attention, **Δεξιά:** Multi-Head Attention

Επιπλέον, δεν χρειαζόμαστε μη-γραμμικότητες καθώς το attention βασίζεται κυρίως στη γωνία μεταξύ διανυσμάτων (εσωτερικό γινόμενο). Για να μπορέσουμε να συγκρίνουμε το query με όλα τα πιθανά keys πρέπει τα \mathbf{q} και \mathbf{k} να είναι ίδιας διάστασης $\mathbf{q}, \mathbf{k} \in \mathbb{R}^{d'}$. Το \mathbf{v} μπορεί να έχει οποιαδήποτε διάσταση $\mathbf{v} \in \mathbb{R}^{d''}$. Για λόγους απλότητας θέτουμε για το υπόλοιπο της παραγράφου $d' = d'' = d$. Άρα έχουμε ένα σύνολο από \mathbf{x} από όπου προκύπτει ένα σύνολο από queries, ένα σύνολο από keys και ένα σύνολο από values. Μπορούμε να τα στοιβάξουμε σε πίνακες, όπου κάθε ένας έχει t στήλες από τα t διανύσματα. Κάθε διάνυσμα έχει διάσταση d , άρα τελικά:

$$\{\mathbf{x}_i\}_{i=1}^t \rightsquigarrow \{\mathbf{q}_i\}_{i=1}^t, \{\mathbf{k}_i\}_{i=1}^t, \{\mathbf{v}_i\}_{i=1}^t \rightsquigarrow \mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{d \times t}$$

Με την ίδια λογική με την προηγούμενη παράγραφο, συγκρίνουμε το query \mathbf{q} με όλα τα keys \mathbf{K} :

$$\mathbf{a} = \text{softmax}_\beta(\mathbf{K}^\top \mathbf{q}) \in \mathbb{R}^t$$

Τέλος, το hidden layer θα είναι ένας γραμμικός συνδυασμός των στηλών του \mathbf{V} με συντελεστές αυτούς του \mathbf{a} :

$$\mathbf{h} = \mathbf{V}\mathbf{a} \in \mathbb{R}^d$$

Σε μορφή πινάκων αφού έχουμε t θα έχουμε t και άρα πίνακα \mathbf{A} διάστασης $t \times t$. Δηλαδή:

$$\{\mathbf{q}_i\}_{i=1}^t \rightsquigarrow \{\mathbf{a}_i\}_{i=1}^t, \rightsquigarrow \mathbf{A} \in \mathbb{R}^{t \times t}$$

$$\mathbf{H} = \mathbf{V}\mathbf{A} \in \mathbb{R}^{d \times t}$$

Στην υλοποίηση, μπορούμε να επιταχύνουμε τους υπολογισμούς στοιβάζοντας όλα τα \mathbf{W} σε έναν πίνακα \mathbf{W} και κάνοντας μόνο έναν πολλαπλασιασμό πινάκων:

$$\begin{bmatrix} \mathbf{q} \\ \mathbf{k} \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} \mathbf{W}_q \\ \mathbf{W}_k \\ \mathbf{W}_v \end{bmatrix} \mathbf{x} \in \mathbb{R}^{3d}$$

3.7.4 Multi-head attention

Τέλος, μπορούμε να προβάλλουμε την είσοδο σε περισσότερους από έναν χώρους και να υπολογίζουμε την ομοιότητα σε κάθε έναν από αυτούς. Έχουμε δηλαδή πολλαπλές κεφαλές “heads” όπως φαίνεται στο Σχήμα 3.7.2 (δεξιά). Π.χ για h heads έχουμε h \mathbf{q} , \mathbf{k} και \mathbf{v} άρα καταλήγουμε σε διάνυσμα \mathbb{R}^{3hd} .

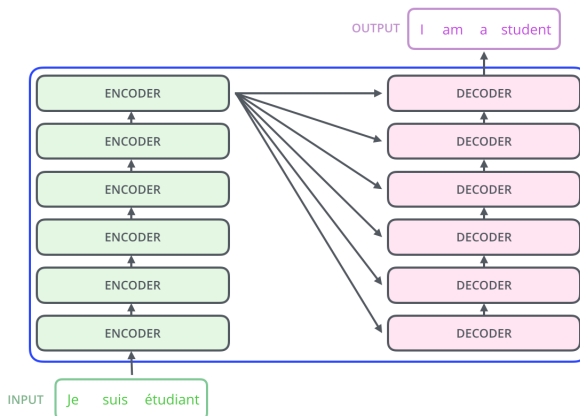
$$\begin{bmatrix} \mathbf{q}^1 \\ \mathbf{q}^2 \\ \vdots \\ \mathbf{q}^h \\ \mathbf{k}^1 \\ \mathbf{k}^2 \\ \vdots \\ \mathbf{k}^h \\ \mathbf{v}^1 \\ \mathbf{v}^2 \\ \vdots \\ \mathbf{v}^h \end{bmatrix} = \begin{bmatrix} \mathbf{W}_q^1 \\ \mathbf{W}_q^2 \\ \vdots \\ \mathbf{W}_q^h \\ \mathbf{W}_k^1 \\ \mathbf{W}_k^2 \\ \vdots \\ \mathbf{W}_k^h \\ \mathbf{W}_v^1 \\ \mathbf{W}_v^2 \\ \vdots \\ \mathbf{W}_v^h \end{bmatrix} \mathbf{x} \in \mathbb{R}^{3hd}$$

Στη συνέχεια μπορούμε να προβάλλουμε τις multi-headed values στην αρχική διάσταση \mathbb{R}^d χρησιμοποιώντας έναν πίνακα $\mathbf{W}_h \in \mathbb{R}^{d \times hd}$.

3.7.5 The Transformer

Το μοντέλο Transformer πρωτοεφαρμόστηκε σε εφαρμογές sequence-to-sequence μετάφρασης και είχε τη μορφή ενός encoder-decoder, αφού αυτή ακολουθούσαν όλα τα μέχρι τότε μοντέλα που έλυναν το συγκεκριμένο πρόβλημα [BCB16]. Ο κωδικοποιητής (encoder) αντιστοιχίζει ένα σύνολο εισόδων $\{\mathbf{x}_i\}_{i=1}^t = \{\mathbf{x}_1, \dots, \mathbf{x}_t\}$ σε ένα σύνολο αναπαραστάσεων $\{\mathbf{z}_i\}_{i=1}^t = \{\mathbf{z}_1, \dots, \mathbf{z}_t\}$. Δεδομένου του \mathbf{z} , ο αποκωδικοποιητής δημιουργεί ένα σύνολο εξόδου $\{\mathbf{y}_i\}_{i=1}^t = \{\mathbf{y}_1, \dots, \mathbf{y}_t\}$ με σύμβολα, ένα στοιχείο τη φορά. Σε κάθε βήμα το μοντέλο είναι auto-regressive [Gra14], χρησιμοποιώντας τα σύμβολα που δημιουργήθηκαν προηγουμένως ως πρόσθετη είσοδο κατά τη δημιουργία του επόμενου.

Ο encoder στην πραγματικότητα είναι μια στοίβα κωδικοποιητών (encoders), όπου όλοι έχουν την ίδια δομή (όμως όχι κοινά βάρη). Όμοια ο decoder είναι μια στοίβα αποκωδικοποιητών (decoders) του ίδιου αριθμού. (βλ. Σχήμα 3.7.3).



Σχήμα 3.7.3: Στοιβές από Encoders - decoders

3.7.6 Encoder

Κάθε encoder αποτελείται από δύο Sublayers(-), όπως φαίνεται στο Σχήμα 3.7.5 (αριστερά), δηλαδή:

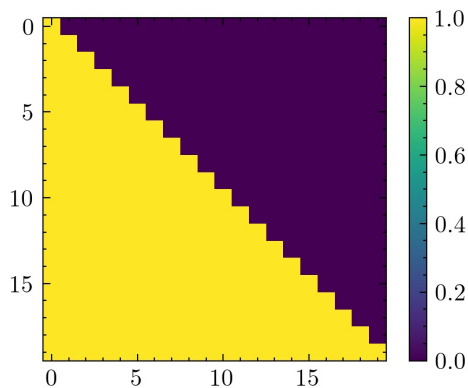
- Ένα στρώμα multi-head self-attention (Σχήμα 3.7.2), που βοηθά τον κωδικοποιητή να συσχετίσει τις εισόδους μεταξύ τους.
- Ένα position-wise fully connected feed-forward νευρωνικό δίκτυο (ισοδύναμο με 1D-convolution).

Οι έξοδοι κάθε υποστρώματος έχουν residual connections [He+15] με το προηγούμενο και Layer-normalization [BKH16]. Άρα έχουμε για κάθε Sublayer έξοδο ίση με: $\text{LayerNorm}(x + \text{Sublayer}(x))$. Τέλος εφαρμόζουμε dropout [Sri+14] στην έξοδο κάθε υποστρώματος, πριν προστεθεί στην είσοδο του επόμενου υποστρώματος.

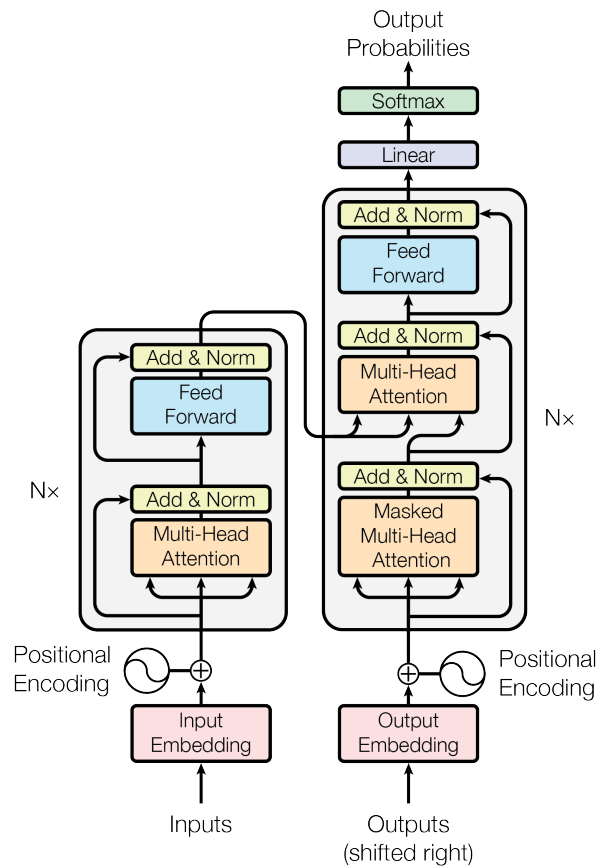
3.7.7 Decoder

Κάθε decoder έχει τα δύο παραπάνω sublayers συν ένα επιπλέον multi-head attention με queries και values από την έξοδο του encoder stack (cross-attention, Σχήμα 3.7.3). Παρόμοια με τον κωδικοποιητή, χρησιμοποιούμε residual connections σε κάθε ένα από τα υποστρώματα, ακολουθούμενες από layer normalization.

Τροποποιούμε, επίσης, το self-attention υποστρώμα στη στοίβα του αποκωδικοποιητή για να αποτρέψουμε να δίνει προσοχή σε επόμενες θέσεις. Άρα αφού τα embeddings εξόδου παράγονται με auto-regressive τρόπο (δηλ. αυξάνεται κατά ένα η θέση i), αρκεί ο decoder να μην μπορεί να δώσει προσοχή σε θέσεις με μεγαλύτερο i . Αυτό επιτυγχάνεται με μία άνω-τριγωνική μάσκα όπως στο Σχήμα 3.7.4.

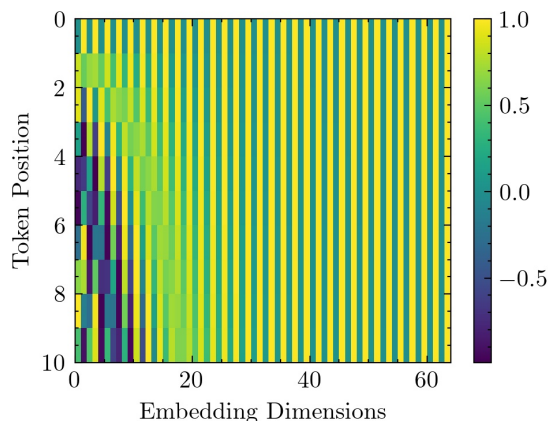


Σχήμα 3.7.4: Κάθε λέξη (σειρά) επιτρέπεται να κάνει attend σε λέξεις (στήλες) που ανήκουν στο παρελθόν αυτής.



Σχήμα 3.7.5: **Αριστερά:** Encoders, **Δεξιά:** Decoders.

3.7.8 Positional encoding



Σχήμα 3.7.6: Οπτικοποίηση των Positional embeddings.

Το μοντέλο μας είδαμε ότι είναι equivariant ως προς το permutation της εισόδου, δηλαδή με απλά λόγια αγνοεί τη σειρά των εισόδων. Για να το χρησιμοποιήσουμε όμως σε δεδομένα όπου η χρονική συσχέτιση μεταξύ των εισόδων είναι σημαντική πρέπει να περάσουμε επιπλέον πληροφορία που να κωδικοποιεί τη συσχέτιση αυτή. Αυτή η πληροφορία ονομάζεται positional embeddings ή encodings και προστίθεται (μπορούν να γίνουν και άλλες πράξεις όπως concatenation) στα embeddings της εισόδου. Άρα έχουν την ίδια διάσταση d_{model} με τα embeddings εισόδου, για να μπορούν να προστεθούν. Μερικές επιλογές γι' αυτά αναφέρονται στο άρθρο [Geh+17]. Στο πρώτο άρθρο που παρουσιάστηκε η αρχιτεκτονική του transformer [Vas+17] χρησιμοποιήθηκαν ημίτονα και συνημίτονα διαφορετικών συχνοτήτων (Σχήμα 3.7.6):

$$PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{\text{model}}})$$

$$PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{\text{model}}})$$

όπου pos η θέση και i η διάσταση. Άρα κάθε διάσταση του positional encoding αντιστοιχεί σε ένα ημιτονοειδές. Οι κυματομορφές ακολουθούν γεωμετρική πρόοδο από το 2π στο $10000 \cdot 2\pi$. Οι συναρτήσεις αυτές επιτρέπουν στο μοντέλο να μάθει εύκολα να κάνει attend σχετικές θέσεις, αφού για οποιαδήποτε σταθερή μετατόπιση k , η PE_{pos+k} είναι γραμμική συνάρτηση του PE_{pos} . Επιπλέον, εφαρμόζεται dropout στο τελικό άθροισμα των embeddings εισόδου με τα positional encodings και στον encoder και στον decoder. Επιλέγεται $P_{\text{drop}} = 0.1$.

3.7.9 Τα τελικά Linear και Softmax Layers

Ο decoder παράγει ένα διάνυσμα από πραγματικούς αριθμούς. Στην περίπτωση που κάνουμε μετάφραση πρέπει να έχουμε ως έξοδο μία λέξη κάθε στιγμή. Γι' αυτό το λόγο έχουμε ένα τελευταίο linear layer που ακολουθείται από ένα Softmax Layer. Το Linear layer προβάλλει την έξοδο του decoder σε ένα πολύ μεγαλύτερης διάστασης διάνυσμα που ονομάζεται logits vector. Η διάσταση αυτού του διανύσματος είναι ίση με τη διάσταση του λεξιλογίου για το task της μετάφρασης (π.χ. 100.000 για 100.000 λέξεις). Η softmax, στη συνέχεια, μετατρέπει τα logits σε pseudoprobabilities με κάθε cell να αναπαριστά την πιθανότητα η επόμενη λέξη στη μετάφραση να είναι αυτή που αντιστοιχεί σε αυτό.

Τέλος, επειδή το μοντέλο παράγει τις εξόδους μία κάθε φορά, μπορούμε να υποθέσουμε ότι το μοντέλο επιλέγει τη λέξη με την υψηλότερη πιθανότητα από αυτήν την κατανομή πιθανότητας και απορρίπτει τα υπόλοιπα. Αυτός είναι ένας μόνο τρόπος αποκωδικοποίησης (greedy decoding). Ένας άλλος τρόπος είναι να κρατήσουμε τις δύο κορυφαίες λέξεις και στη συνέχεια, στο επόμενο βήμα, να εκτελέσουμε το pass του decoder δύο φορές: μια φορά υποθέτοντας ότι η σωστή λέξη ήταν αυτή με τη μεγαλύτερη πιθανότητα και μια άλλη φορά υποθέτοντας ότι η λέξη ήταν αυτή με τη δεύτερη μεγαλύτερη πιθανότητα. Το επαναλαμβάνουμε αυτό για τις θέσεις 2 και 3 κ.λπ. Αυτή η μέθοδος ονομάζεται beam search, όπου στο παράδειγμά μας, $beam_size = 2$ (που σημαίνει ότι ανά πάσα στιγμή, δύο μερικές υποθέσεις (ημιτελείς μεταφράσεις) διατηρούνται στη μνήμη) και το $top_beams = 2$ (που σημαίνει ότι θα επιστρέψουμε δύο μεταφράσεις). Αυτές είναι υπερπαράμετροι που μπορούμε να δοκιμάσουμε.

Κεφάλαιο 4

Self-supervised learning

4.1	Αυτο-επιβλεπόμενη μάθηση (Self-supervised Learning)	42
4.1.1	Μοντελοποιώντας την αβεβαιότητα πρόβλεψης	43
4.1.2	Ενοποιημένη οπτική στη χρήση αυτο-επιβλεπόμενης μάθησης	44
4.1.3	Αρχιτεκτονικές στο Self-Supervised Learning	45
4.1.4	Απλές επιλογές για pretext task	48
4.1.5	Contrastive energy-based SSL	50
4.1.6	Non-contrastive energy-based SSL	56

4.1 Αυτο-επιβλεπόμενη μάθηση (Self-supervised Learning)

Η παρούσα παράγραφος βασίζεται στα [LM; Can]. Όλα τα σχήματα προέρχονται από εκεί, εκτός και αν αναφέρεται διαφορετικά.

Ο τομέας της Τεχνητής Νοημοσύνης (AI) έχει σημειώσει τεράστια πρόοδο στην ανάπτυξη συστημάτων τα οποία μπορούν να μάθουν από τεράστιες βάσεις προσεκτικά επισημειωμένων (labeled) δεδομένων. Ειδικότερα, το παράδειγμα της επιβλεπόμενης μάθησης (supervised learning) σε συνδυασμό με τη βαθιά μάθηση (deep learning) έχει αποδειχθεί ο καλύτερος τρόπος για την ανάπτυξη μοντέλων, που αποδίδουν εξαιρετικά σε ένα πολύ συγκεκριμένο πρόβλημα (task) πάνω στο οποίο έχουν εκπαιδευτεί [SM19a]. Δυστυχώς, υπάρχει ένα όριο στο πόσο μακριά μπορεί να φτάσει το πεδίο της τεχνητής νοημοσύνης μόνο με την επιβλεπόμενη μάθηση. Αυτό γιατί, η μάθηση με επίβλεψη αποτυγχάνει στην ανάπτυξη μοντέλων που γενικεύουν και μπορούν να μαθαίνουν συνεχώς νέες δεξιότητες, χωρίς την ανάγκη για τεράστιες ποσότητες δεδομένων.

Μια υπόθεση που ακόμη διερευνάται, είναι ότι η γενικευμένη γνώση για τον κόσμο, ή αλλιώς κοινή λογική, αποτελεί το μεγαλύτερο μέρος της βιολογικής νοημοσύνης τόσο στους ανθρώπους όσο και στα ζώα [Sha+20]. Αυτή η κοινή λογική θεωρείται δεδομένη σε ανθρώπους και ζώα, αλλά παραμένει ανοιχτό πρόβλημα στην έρευνα για την τεχνητή νοημοσύνη από την έναρξή της. Η κοινή λογική βοηθά τους ανθρώπους να μάθουν νέες δεξιότητες χωρίς να απαιτούν τεράστιο όγκο δεδομένων για κάθε νέα εργασία. Για παράδειγμα, αν δείξουμε μόνο μερικά σχέδια ελέφαντα σε μικρά παιδιά, θα μπορούν πλέον να αναγνωρίσουν οποιοδήποτε ελέφαντα βλέπουν. Αντίθετα, τα μοντέλα που εκπαιδεύονται με επιβλεπόμενη μάθηση απαιτούν πολλά παραδείγματα εικόνων ελέφαντα και ενδέχεται να αποτύχουν να αναγνωρίσουν ελέφαντες σε ασυνήθιστες καταστάσεις, όπως το να βρίσκεται σε μια παραλία. Όμοια, οι άνθρωποι μαθαίνουν να οδηγούν ένα αυτοκίνητο σε περίπου 20 ώρες πρακτικής με πολύ μικρή επίβλεψη, ενώ η πλήρως αυτόνομη οδήγηση εξακολουθεί να δυσκολεύει τα καλύτερα συστήματα AI που έχουν εκπαιδευτεί με χιλιάδες ώρες δεδομένων από ανθρώπους οδηγούς. Ο λόγος είναι ότι οι άνθρωποι βασίζονται στις προηγούμενες γνώσεις τους σχετικά με τον τρόπο λειτουργίας του κόσμου. Η αυτο-επιβλεπόμενη μάθηση (Self-supervised learning - SSL) είναι ένας από τους πιο πολλά υποσχόμενους τρόπους για την προσέγγιση μιας μορφής κοινής λογικής στα συστήματα AI.

Ορισμός 4.1.1: Self Supervised Learning

Η αυτο-επιβλεπόμενη μάθηση επιτρέπει στα συστήματα AI να εκπαιδεύονται από μεγάλες βάσεις μη επισημειωμένων δεδομένων. Μέσα σε αυτές είναι πολύ πιθανότερο να εμπεριέχονται λιγότερο συνηθισμένα στιγμιότυπα του κόσμου.



Σχήμα 4.1.1: Στην αυτο-επιβλεπόμενη μάθηση, το σύστημα εκπαιδεύεται για να προβλέψει κρυμμένα μέρη της εισόδου (σε γκρι) από ορατά μέρη της εισόδου (με πράσινο χρώμα).

Η αυτο-επιβλεπόμενη μάθηση λαμβάνει σήματα επίβλεψης από τα ίδια τα δεδομένα, αξιοποιώντας συχνά την υποκείμενη δομή τους. Η γενική τεχνική της αυτο-επιβλεπόμενης μάθησης είναι η πρόβλεψη οποιουδήποτε μη παρατηρήσιμου - κρυμμένου τμήματος της εισόδου από οποιοδήποτε παρατηρήσιμο - μη κρυφό μέρος της όπως φαίνεται στο Σχήμα 4.1.1.

Για παράδειγμα, είναι συνηθισμένο στην επεξεργασία φυσικής γλώσσας (NLP), να κρύβουμε μέρος μιας πρότασης και να προσπαθούμε να προβλέψουμε τις κρυφές λέξεις από τις υπόλοιπες. Μπορούμε επίσης να προβλέψουμε προηγούμενα ή μελλοντικά καρέ σε ένα βίντεο (κρυφά δεδομένα) από τα τρέχοντα (παρατηρούμενα δεδομένα). Δεδομένου ότι η αυτο-επιβλεπόμενη μάθηση χρησιμοποιεί τη δομή των ίδιων των δεδομένων, έχουμε τα εξής πλεονεκτήματα όταν τη χρησιμοποιούμε:

- μπορούμε να χρησιμοποιήσουμε σήματα επίβλεψης από διαφορετικές πηγές (π.χ. βίντεο και ήχο) ταυτόχρονα.
- Αφού πλέον δεν μας ενδιαφέρουν οι πολύ συγκεκριμένες ετικέτες των μεγάλων επισημειωμένων βάσεων δεδομένων μπορούμε να κάνουμε συνένωση βάσεων είτε αυτές είναι επισημειωμένες είτε όχι.

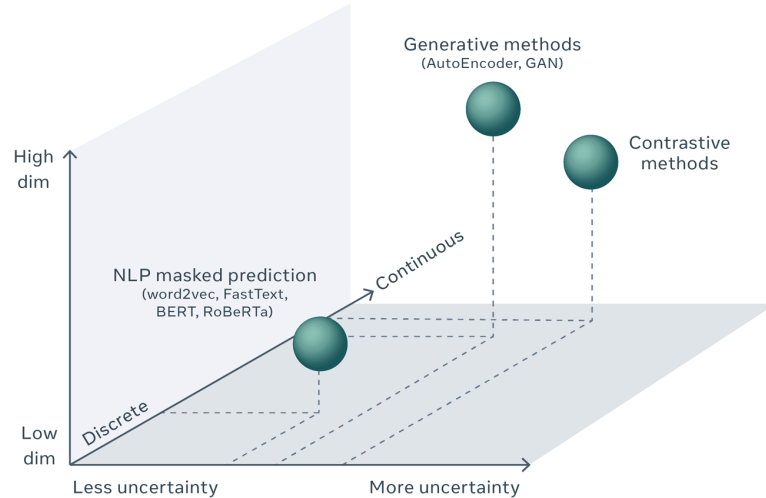
Η μη επιβλεπόμενη μάθηση (unsupervised learning) είναι ένας όρος που εσφαλμένα χρησιμοποιείται σε συστήματα που εκπαιδεύονται με την παραπάνω λογική, αντί του σωστού “αυτο-επιβλεπόμενη μάθηση” και υποδηλώνει ότι η μάθηση δεν χρησιμοποιεί καθόλου επίβλεψη [LM]. Στην πραγματικότητα, η αυτο-επιβλεπόμενη όχι απλώς χρησιμοποιεί επίβλεψη, αλλά τα σήματα επίβλεψης που χρησιμοποιεί περιέχουν πολύ περισσότερη πληροφορία από ότι οι τυπικές μέθοδοι επίβλεψης. Για παράδειγμα σε ένα κλασικό supervised πρόβλημα κατηγοριοποίησης (classification) έχουμε ως ετικέτα την κατηγορία ενός βίντεο ($\sim \frac{10-10000 \text{ bits}}{\text{sample}}$), ενώ η αυτο-επίβλεψη έχει ως ετικέτα μελλοντικά/μη παρατηρούμενα frames ενός βίντεο ($\sim \frac{1000000 \text{ bits}}{\text{sample}}$).

Η αυτο-επιβλεπόμενη μάθηση έχει εξαιρετικά αποτελέσματα στην επεξεργασία φυσικής γλώσσας, επιτρέποντάς μας να εκπαιδεύουμε μοντέλα όπως τα BERT [Dev+19], RoBERTa [Liu+19], XLM-R [Con+20] και άλλα σε μεγάλα σύνολα δεδομένων κειμένου χωρίς ετικέτες και στη συνέχεια να χρησιμοποιούμε αυτά τα μοντέλα για πιο συγκεκριμένα tasks. Στη φάση αυτο-επιβλεπόμενης προ-εκπαίδευσης, εμφανίζεται στο σύστημα ένα σύντομο κείμενο (συνήθως 1.000 λέξεις) στο οποίο ορισμένες από τις λέξεις έχουν σβηστεί ή αντικατασταθεί. Το σύστημα εκπαιδεύεται -στη συνέχεια- να προβλέπει τις λέξεις που είχαν καλυφθεί ή αντικατασταθεί. Για να ολοκληρώσετε -για παράδειγμα- μια πρόταση όπως «Το (κενό) κυνηγά το (κενό) στη σαβάνα», το σύστημα πρέπει να μάθει ότι τα λιοντάρια ή τα σιτάχ μπορούν να κυνηγήσουν την αντιλόπη. Επίσης ότι οι γάτες κυνηγούν ποντίκια αλλά στην κουζίνα, όχι στη σαβάνα. Ως συνέπεια της εκπαίδευσης, το σύστημα μαθαίνει την έννοια των λέξεων, τον συντακτικό ρόλο τους και το νόημα ολόκληρων των κειμένων. Οι ίδιες τεχνικές, ωστόσο, δεν είναι το ίδιο εύκολο να εφαρμοστούν σε άλλους τύπους δεδομένων π.χ. εικόνα, βίντεο και χρονοσειρές. Παρά τα πολλά υποσχόμενα πρώτα αποτελέσματα, το SSL δεν έχει επιφέρει ακόμη τις ίδιες βελτιώσεις σε αυτούς τους τομείς σε σύγκριση με αυτές που έχουμε δει στο NLP. Ο κύριος λόγος είναι ότι είναι πολύ πιο δύσκολο να απεικονιστεί η αβεβαιότητα της πρόβλεψης για εικόνες από ότι για τις λέξεις. Όταν η λέξη που λείπει δεν μπορεί να προβλεφθεί ακριβώς (είναι «λιοντάρι» ή «σίτα»);), το σύστημα μπορεί να αντιστοιχίσει ένα σκορ ή μια πιθανότητα σε όλες τις πιθανές λέξεις στο λεξιλόγιο: υψηλή βαθμολογία για «λιοντάρι», «σίτα» και σε μερικούς άλλους θηρευτές και χαμηλές βαθμολογίες για όλες τις άλλες λέξεις στο λεξιλόγιο. Αλλά δεν ξέρουμε πώς να αναπαραστήσουμε την αβεβαιότητα όταν προβλέπουμε καρέ σε ένα βίντεο ή όταν λείπουν κρυμμένα κομμάτια σε μια εικόνα. Δεν μπορούμε να απαριθμήσουμε όλα τα πιθανά καρέ βίντεο και να συσχετίσουμε μια βαθμολογία σε καθένα από αυτά, επειδή υπάρχει ένας άπειρος αριθμός από αυτά.

4.1.1 Μοντελοποιώντας την αβεβαιότητα πρόβλεψης

Για να κατανοήσουμε καλύτερα αυτήν την πρόκληση, πρέπει πρώτα να κατανοήσουμε με μεγαλύτερη λεπτομέρεια την αβεβαιότητα της πρόβλεψης και τον τρόπο με τον οποίο διαμορφώνεται στην επεξεργασία φυσικής γλώσσας σε σύγκριση π.χ. με την όραση (βλ. Σχήμα 4.1.1). Στην επεξεργασία φυσικής γλώσσας, η πρόβλεψη των λέξεων που λείπουν περιλαμβάνει τον υπολογισμό ενός σκορ προβλέψεων για κάθε πιθανή λέξη στο λεξιλόγιο.

Αν και το λεξιλόγιο είναι μεγάλο και η πρόβλεψη της λέξης που λείπει περιλαμβάνει κάποια αβεβαιότητα, είναι δυνατό να δημιουργήσουμε μια λίστα με όλες τις λέξεις στο λεξιλόγιο μαζί με μια εκτίμηση πιθανότητας για την εμφάνιση της κάθε λέξης σε κάθε κενή-κρυφή θέση. Στα κλασικά συστήματα μηχανικής μάθησης αυτό γίνεται αντιμετωπίζοντας το πρόβλημα πρόβλεψης ως πρόβλημα κατηγοριοποίησης και υπολογίζοντας ένα σκορ



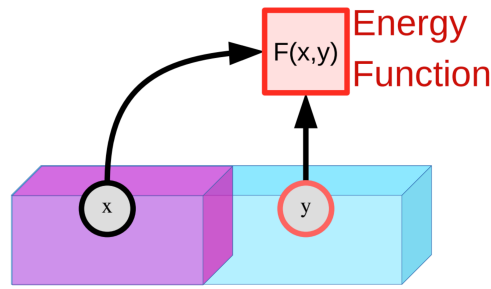
Σχήμα 4.1.2: Οπτικοποίηση μοντέλων που χρησιμοποιούνται στην πράξη στους άξονες αβεβαιότητας, διακριτότητας και μεγέθους διάστασης των δεδομένων εισόδου

για κάθε θέση και κάθε λέξη χρησιμοποιώντας ένα γιγαντιαίο softmax layer, το οποίο μετατρέπει τα σκόρ σε μία κατανομή πιθανότητας πάνω στις λέξεις. Έτσι η αβεβαιότητα της πρόβλεψης αντιπροσωπεύεται από μια κατανομή πιθανότητας για όλα τα πιθανά αποτελέσματα, υπό την προϋπόθεση ότι υπάρχει ένας πεπερασμένος αριθμός πιθανών αποτελεσμάτων. Στην όραση, από την άλλη πλευρά, το ανάλογο πρόβλημα της πρόβλεψης “κρυμμένων” καρτέ σε ένα βίντεο αποτελεί πρόβλεψη συνεχών σημάτων υψηλής διάστασης και όχι διακριτών αποτελεσμάτων. Δεν είναι δυνατή η ρητή αναπαράσταση όλων των πιθανών καρτέ βίντεο και η συσχέτιση βαθμολογίας πρόβλεψης με αυτά καθώς τα πρώτα είναι άπειρα. Στην πραγματικότητα, ενδέχεται να μην βρούμε ποτέ τεχνικές που να αναπαριστούν κατανομές πιθανότητας για συνεχόμενα σήματα υψηλών διαστάσεων, όπως το σύνολο όλων των πιθανών καρτέ ενός βίντεο.

4.1.2 Ενοποιημένη οπτική στη χρήση αυτο-επιβλεπόμενης μάθησης

Υπάρχει ένας τρόπος να σκεφτούμε την αυτο-επιβλεπόμενη μάθηση ως ένα ενοποιημένο πλαίσιο ενός μοντέλου που βασίζεται στην ενέργεια (Energy-Based Models ή EBM). Το EBM είναι ένα προς εκπαίδευση σύστημα με δύο εισόδους, x και y , που προβλέπει πόσο ασύμβατες είναι οι εισόδοι μεταξύ τους. Για παράδειγμα, αν x και y δύο κομμάτια βίντεο, το EBM θα μας έλεγε σε ποιο βαθμό το y είναι μια καλή συνέχεια για το x . Για να δείξει την ασυμβατότητα μεταξύ x και y , το EBM παράγει έναν μόνο αριθμό, που ονομάζεται ενέργεια. Εάν η ενέργεια είναι χαμηλή, τα x και y θεωρούνται συμβατά. εάν είναι υψηλή, θεωρούνται ασύμβατα.

Η εκπαίδευση ενός EBM αποτελείται από δύο μέρη: (1) Δίνουμε στο σύστημα παραδείγματα x και y που είναι συμβατά και το εκπαιδεύουμε να παράγει χαμηλή ενέργεια και (2) Βρίσκουμε τρόπο, ώστε για ένα συγκεκριμένο x , οι τιμές y που είναι ασύμβατες με το x παράγουν υψηλότερη ενέργεια από τις τιμές y που είναι συμβατές με το x (βλ. Σχήμα 4.1.3). Το πρώτο μέρος είναι απλό, αλλά το δεύτερο μέρος είναι όπου βρίσκεται η δυσκολία. Για αναγνώριση εικόνας π.χ., το μοντέλο μας λαμβάνει δύο εικόνες, x και y , ως εισόδους. Εάν τα x και y είναι ελαφρώς παραμορφωμένες εκδόσεις της ίδιας εικόνας, το μοντέλο έχει εκπαιδευτεί για να παράγει χαμηλή ενέργεια στην έξοδό του. Για παράδειγμα, το x θα μπορούσε να είναι μια φωτογραφία ενός αυτοκινήτου και το y μια φωτογραφία του ίδιου αυτοκινήτου που τραβήχτηκε από σε μια διαφορετική τοποθεσία και διαφορετική ώρα της ημέρας, έτσι ώστε το αυτοκίνητο στην y να έχει μετατοπιστεί, περιστραφεί, μεγεθυνθεί, ή μικρύνει ή να εμφανίζει ελαφρώς διαφορετικά χρώματα και σκιές από το αυτοκίνητο στη x .



Σχήμα 4.1.3: Ένα μοντέλο που βασίζεται στην ενέργεια (EBM) μετρά τη συμβατότητα μεταξύ μιας παρατήρησης x και μιας προτεινόμενης πρόβλεψης y . Εάν τα x και y είναι συμβατά, η ενέργεια είναι ένας μικρός αριθμός. εάν είναι ασύμβατα, η ενέργεια είναι μεγαλύτερος αριθμός.

4.1.3 Αρχιτεκτονικές στο Self-Supervised Learning

Joint embedding, Siamese networks

Μια αρχιτεκτονική βαθιάς μάθησης η οποία υλοποιεί την παραπάνω ιδέα είναι τα λεγόμενα Siamese δίκτυα (Siamese networks) ή joint embedding αρχιτεκτονικές (joint embedding architecture). Η αρχιτεκτονική αναφέρεται σε άρθρα από το εργαστήριο του Geoff Hinton και την ομάδα του Yann LeCun στις αρχές της δεκαετίας του 1990 [BH92; BRO+93] και στα μέσα της δεκαετίας του 2000 [CHL05; HCL06; Gol+05]. Η ιδέα αγνοήθηκε για μεγάλο χρονικό διάστημα, αλλά επανήλθε στο προσκήνιο από τα τέλη του 2019. Ένα joint embedding δίκτυο αποτελείται από δύο πανομοιότυπα (ή σχεδόν πανομοιότυπα) αντίγραφα του ίδιου δικτύου. Το ένα δίκτυο τροφοδοτείται με x και το άλλο με y . Τα δίκτυα παράγουν διανύσματα εξόδου που ονομάζονται embeddings, τα οποία αναπαριστούν τα x και y σε μία μικρότερη διάσταση. Ένα τρίτο δίκτυο, που συνδέει τα δύο δίκτυα, υπολογίζει την ενέργεια ως την απόσταση μεταξύ των δύο διανυσμάτων (embeddings). Όταν το μοντέλο παίρνει ως είσοδο παραμορφωμένες εκδόσεις της ίδιας εικόνας, οι παράμετροι των δύο πρώτων δικτύων μπορούν εύκολα να ρυθμιστούν έτσι ώστε οι έξοδοι τους να βρίσκονται κοντά. Αυτό διασφαλίζει ότι το δίκτυο παράγει σχεδόν πανομοιότυπες αναπαραστάσεις (embeddings) ενός αντικειμένου, ανεξάρτητα από τη συγκεκριμένη λήψη της φωτογραφίας αυτού του αντικειμένου.

Η δυσκολία είναι να βεβαιωθούμε ότι τα δίκτυα παράγουν υψηλή ενέργεια, δηλαδή διαφορετικά embeddings, όταν τα x και y είναι διαφορετικές εικόνες. Αν δεν θέσουμε κάποιο περιορισμό, τα δύο δίκτυα μπορούν να αγνοήσουν τις εισόδους τους και να παράγουν πάντα πανομοιότυπες αναπαραστάσεις εξόδου. Αυτό το φαινόμενο ονομάζεται κατάρρευση (collapse). Όταν συμβαίνει η κατάρρευση, η ενέργεια δεν είναι υψηλότερη για μη συμβατά x και y από ότι για συμβατά. Υπάρχουν δύο κατηγορίες τεχνικών για την αποφυγή κατάρρευσης: μέθοδοι αντίθεσης (contrastive methods) και μέθοδοι κανονικοποίησης (regularization methods).

Auto-encoders

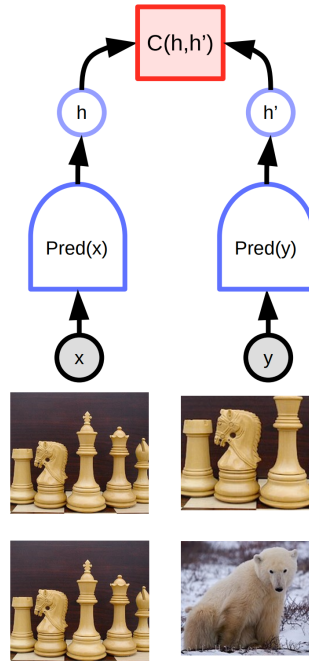
Ένα υποσύνολο αρχιτεκτονικών που μπορούν εύκολα να αναπαρασταθούν ως EBM's είναι οι autoencoders.

Ορισμός 4.1.2: Autoencoder: Ορισμοί μεταβλητών

Για το υπόλοιπο του κεφαλαίου ορίζουμε τις εξής μεταβλητές:

- x : Σημαίνει ότι η μεταβλητή είναι παρατηρήσιμη και κατά το training και κατά το testing.
- y : Σημαίνει ότι η μεταβλητή είναι παρατηρήσιμη κατά το training αλλά όχι κατά το testing.
- z : Δεν είναι παρατηρήσιμη ούτε κατά το training ούτε κατά το testing.
- h : Παράγεται από την είσοδο (hidden).
- \tilde{y} : Παράγεται από το hidden layer.

Αυτού του είδους τα δίκτυα χρησιμοποιούνται για την εκμάθηση της εσωτερικής δομής της εισόδου και την κωδικοποίησή της σε μια κρυφή εσωτερική αναπαράσταση h , η οποία εκφράζει την είσοδο. Στην πιο απλή του μορφή ο autoencoder:



Σχήμα 4.1.4: Joint embedding architecture. Η συνάρτηση C στην κορυφή παράγει ένα βαθμωτό μέγεθος (ενέργεια) που μετρά την απόσταση μεταξύ των διανυσμάτων αναπαράστασης (embeddings) που παράγονται από δύο πανομοιότυπα δίδυμα δίκτυα που μοιράζονται τις ίδιες παραμέτρους (w). Όταν τα x και y είναι ελαφρώς διαφορετικές εκδόσεις της ίδιας εικόνας, το σύστημα έχει εκπαιδευτεί να παράγει χαμηλή ενέργεια, γεγονός που αναγκάζει το μοντέλο να παράγει παρόμοια διανύσματα ενσωμάτωσης για τις δύο εικόνες. Το δύσκολο μέρος είναι να εκπαιδεύσουμε το μοντέλο έτσι ώστε να παράγει υψηλή ενέργεια (δηλαδή, διαφορετικές ενσωματώσεις) για εικόνες που είναι διαφορετικές.

- Αρχικά, λαμβάνει μια είσοδο \mathbf{y} και την αντιστοιχίζει σε μια κρυφή κατάσταση \mathbf{h} μέσω ενός αφινικού μετασχηματισμού ακολουθούμενου από μίας συνάρτησης ενεργοποίησης $f(\cdot)$.

$$\mathbf{h} = f(\mathbf{W}_h \mathbf{y} + \mathbf{b}_h)$$

Αυτό είναι το στάδιο του encoder. Επίσης το \mathbf{h} ονομάζεται επίσης code.

- Στη συνέχεια, εκτελείται το βήμα του decoder δηλαδή:

$$\tilde{\mathbf{y}} = g(\mathbf{W}_y \mathbf{h} + \mathbf{b}_y)$$

- Τέλος, η ενέργεια είναι το άθροισμα ενός reconstruction και ενός regularization όρου.

$$\mathbf{F}(\mathbf{y}) = \mathbf{C}(\mathbf{y}, \tilde{\mathbf{y}}) + \mathbf{R}(\mathbf{h})$$

Τα \mathbf{y} και $\tilde{\mathbf{y}}$ έχουν ίδια διάσταση, ενώ το \mathbf{h} ανήκει στο \mathbb{R}^d . Επίσης τα \mathbf{W}_h και \mathbf{W}_y είναι τα βάρη των δικτύων.

$$\mathbf{y}, \tilde{\mathbf{y}} \in \mathbb{R}^n$$

$$\mathbf{h} \in \mathbb{R}^d$$

$$\mathbf{W}_h \in \mathbb{R}^{d \times n}$$

$$\mathbf{W}_y \in \mathbb{R}^{n \times d}$$

Κόστος ανακατασκευής:

Όταν έχουμε εισόδους πραγματικών μεταβλητών, ορίζεται ως το τετράγωνο της ευκλείδειας απόστασης μεταξύ \mathbf{y} και $\tilde{\mathbf{y}}$:

$$\mathbf{C}(\mathbf{y}, \tilde{\mathbf{y}}) = \|\mathbf{y} - \tilde{\mathbf{y}}\|^2 = \|\mathbf{y} - \text{Dec}[\text{Enc}(\mathbf{y})]\|^2$$

Όμοια, όταν έχουμε δυαδικές εισόδους, ορίζουμε το Binary cross-entropy:

$$\mathbf{C}(\mathbf{y}, \tilde{\mathbf{y}}) = - \sum_{i=1}^n \mathbf{y}_i \log(\tilde{\mathbf{y}}_i) + (1 - \mathbf{y}_i) \log(1 - \tilde{\mathbf{y}}_i)$$

Τέλος ορίζουμε τα συναρτησιακά του loss, ως τον μέσο όρος των per sample loss functions:

$$\mathcal{L}(\mathbf{F}(\cdot), \mathbf{Y}) = \frac{1}{m} \sum_{j=1}^m \ell(\mathbf{F}(\cdot), \mathbf{y}^{(j)}) \in \mathbb{R}$$

$$\ell_{\text{energy}}(\mathbf{F}(\cdot), \mathbf{y}) = \mathbf{F}(\mathbf{y})$$

Over-Under complete autoencoders

Το μέγεθος της κρυφής αναπαράστασης \mathbf{h} σε αυτά τα δίκτυα μπορεί να είναι μικρότερο **Under Complete Autoencoder** ή και μεγαλύτερο από το μέγεθος εισόδου **Over Complete Autoencoder**. Εάν επιλέξουμε μικρότερο \mathbf{h} , το δίκτυο κάνει μη γραμμική μείωση διαστατικότητας.

Έχουμε δει ότι σε μεγαλύτερες διαστάσεις τα δεδομένα είναι πιο εύκολο να διαχωριστούν γραμμικά. Άρα πολλές φορές χρειάζεται \mathbf{h} με διάσταση μεγαλύτερη της εισόδου. Ωστόσο, σε αυτό το σενάριο, ένας απλός autoencoder θα κατέρρευε (collapse). Αυτό γιατί για μεγάλο \mathbf{h} το μοντέλο μπορεί να αντιγράψει όλα τα χαρακτηριστικά εισόδου στο κρυφό επίπεδο και να τα μεταβιβάσει ως έξοδο και έτσι ουσιαστικά να συμπεριφέρεται ως ταυτοτική συνάρτηση. Έχουμε δηλαδή ένα μοντέλο που δίνει ίδια μηδενική ενέργεια σε κάθε sample αφού το reconstruction loss είναι μηδενικό για όλα.

Για να αποτρέψουμε την κατάρρευση του μοντέλου, πρέπει να χρησιμοποιήσουμε τεχνικές που περιορίζουν τις περιοχές που μπορεί να λάβει μηδενικές ή χαμηλές τιμές η ενέργεια. Αυτές οι τεχνικές μπορεί να είναι κάποιου είδους regularization, όπως sparsity, προσθήκη τεχνητού θορύβου ή δειγματοληψία και θα τις δούμε παρακάτω.

4.1.4 Απλές επιλογές για pretext task

Οι παρακάτω παράγραφοι αφορούν την εφαρμογή του SSL στον τομέα της όρασης υπολογιστών. Αυτό γιατί σε αυτόν τον τομέα είναι πιο διασητικοί οι λόγοι που μας οδήγησαν σε αλλαγές στις διάφορες τεχνικές που χρησιμοποιούμε. Ωστόσο, με πολύ μικρές αλλαγές οι ίδιες τεχνικές μπορούν να χρησιμοποιηθούν (και χρησιμοποιούνται) σε οποιοδήποτε τύπο δεδομένων.

Στην αρχή της εφαρμογής του SSL στον τομέα της όρασης όλα τα pretext tasks είχαν την παρακάτω μορφή:

- Εφαρμογή ενός μετασχηματισμού t στην εικόνα.
- Πέρασμα της μετασχηματισμένης εικόνας I^t από κάποιο συνελικτικό δίκτυο.
- Πρόβλεψη κάποιας ιδιότητας του μετασχηματισμού t ως ένα πρόβλημα classification.

Μερικά επιτυχημένα παραδείγματα φαίνονται στο Σχήμα 4.1.5 και αναλύονται παρακάτω. Η σειρά τους είναι αύξουσα με βάση την πληροφορία που δίνουμε ως supervisory signal στην έξοδο:

- Εφαρμόζουμε ένα rotation στην εικόνα, την περνάμε μέσα από τα συνελικτικό δίκτυο και προβλέπουμε το rotation (4-class classification για 0, 90, 180 και 270 μοίρες) [GSK18].
- Χωρίζουμε την εικόνα σε ένα 3x3 patches. Παίρνουμε το μεσαίο patch και ένα τυχαίο ακόμη patch, τα περνάμε μέσα από το ίδιο δίκτυο, κάνουμε concatenate τα representations και κάνουμε classify τη σχετική τους θέση (8-class classification) [DGE16].
- Χωρίζουμε την εικόνα σε 3x3 patches. Στη συνέχεια ανακατεύουμε τα patches και κάνουμε classify ποιά από τα 9! permutations έγινε (στην πραγματικότητα κρατάμε περίπου 2000 permutations, αλλιώς 9!-classes classification, κάτι που απαιτεί πολλά δεδομένα) [NF17].
- Εφαρμογή τετραγωνικού crop τυχαία στην εικόνα και πρόβλεψη των masked pixels [Pat+16].
- Παίρνουμε ως είσοδο την grayscale εικόνα (L channel) και προσπαθούμε να προβλέψουμε ολόκληρο το (L,ab) concatenated (colorization) [ZIE16].

Όπως βλέπουμε και στα 5 παραδείγματα το supervisory signal προκύπτει από τα ίδια τα δεδομένα και όχι από ανθρώπινη παρέμβαση, όπως συμβαίνει στο κλασικό supervised learning.

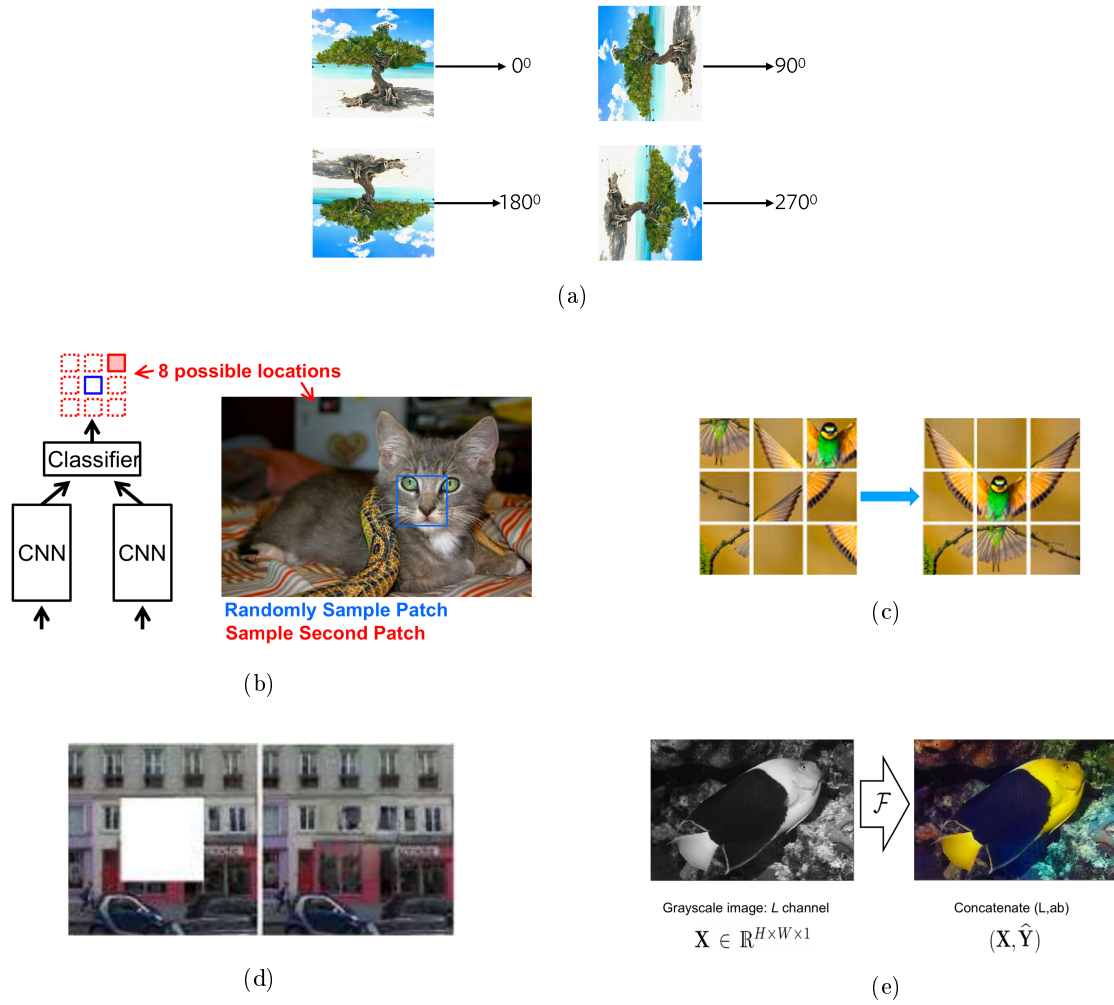
Τα παραπάνω παραδείγματα δίνουν πολύ καλά representations τα οποία μπορούμε να χρησιμοποιήσουμε ως features σε διάφορα downstream tasks. Ένας τρόπος να αξιολογήσουμε την ποιότητα των representations είναι να εκπαιδύσουμε έναν γραμμικό ταξινομητή σε διάφορα downstream tasks πάνω στα representations που δίνει η κάθε έξοδος του συνελικτικού δικτύου με παγωμένα βάρη (linear probing). Ωστόσο, τα εν λόγω representations δεν καταφέρνουν να νικήσουν τα representations που έχουν προκύψει μέσω supervised training, με αυτή τη λογική. Δηλαδή τα representations ενός Resnet50 εκπαιδευμένου στο ImageNet δίνουν καλύτερο αποτέλεσμα στα περισσότερα tasks, όταν πάνω σε αυτά εκπαιδύουμε έναν γραμμικό ταξινομητή.

Για πολύ καιρό η απάντηση στο γιατί συμβαίνει αυτό ήταν ότι το δίκτυο παίρνει πολύ λίγη πληροφορία στην έξοδό του (π.χ. 2 bits για το rotation task) κάτι που απαιτεί πάρα πολλά training samples για να μάθει γενικά representations. Στο άρθρο [Goy+19], όμως αποδείχθηκε πως δεν φταίει ο όγκος των δεδομένων αλλά το ίδιο το task. Συγκεκριμένα, πήραν το καλύτερο σε απόδοση task (jigsaw) και:

- Το εκπαιδύσαν σε περισσότερα δεδομένα (100 εκατομμύρια εικόνες).
- Επιπλέον, αύξησαν το capacity του backbone μοντέλου κατά πολύ δηλαδή έβαλαν ResNet50 ως backbone.

Το αποτέλεσμα στο downstream VOC07 έδειξε ότι (Σχήμα 4.1.6):

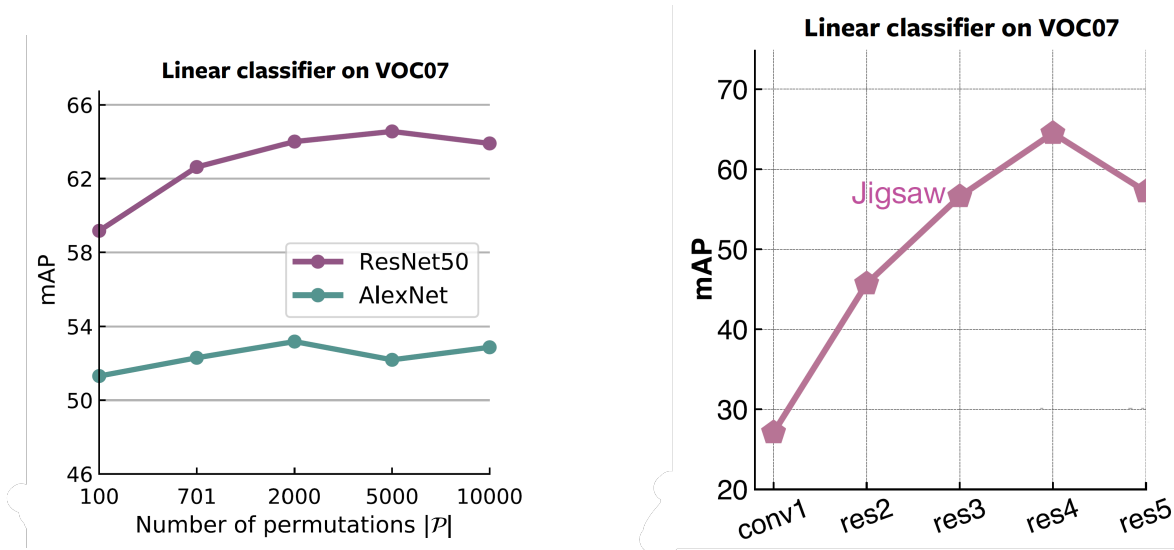
- Καθώς κάνουμε probing σε πιο βαθιά layers του ResNet50 το mAP αυξάνει μέχρι ένα σημείο και έπειτα πέφτει. Κάποιος θα μπορούσε να ισχυριστεί ότι είναι καλή ιδέα να πετάξουμε τα layers μετά το Res4 όπου μεγιστοποιείται το mAP. Τότε σε αυτή την περίπτωση παρατηρείται η ίδια συμπεριφορά (αύξηση και έπειτα μείωση) όμως ένα layer πριν. Αυτό υποδεικνύει ότι τα τελευταία layers έχουν εξειδικευτεί στο jigsaw puzzle και δεν γενικεύουν.
- Καθώς αυξάνουμε τον αριθμό των permutations το mAP αυξάνει και στη συνέχεια πέφτει. Άρα το ίδιο το jigsaw task έχει ένα όριο στο να γενικεύει και δεν φταίει ο όγκος των δεδομένων ή το backbone που χρησιμοποιούμε.



Σχήμα 4.1.5: Με τη σειρά: Rotation, Relative Position, Jigsaw, Square Crop, Colorization

Πράγματι, αν σκεφτούμε πιο προσεκτικά τη λογική του να κάνουμε μετασχηματισμό σε μια εικόνα και στη συνέχεια να προβλέψουμε τον μετασχηματισμό (ή κάποια ιδιότητά του) θα δούμε ότι αυτό αντιφάσκει με τη λογική της μηχανικής μάθησης. Συγκεκριμένα είδαμε ότι στα CNN's θέλουμε η έξοδος να είναι invariant π.χ. στη μετατόπιση της εισόδου, δηλαδή αν μετατοπίσουμε την είσοδο η έξοδος να είναι ίδια. Το ίδιο και σε κλασσικές μεθόδους όπως SIFT [Low99] και HOG [DT05] θέλουμε να έχουμε invariance ως προς τη μετατόπιση. Όμοια στο deep-learning χρησιμοποιώντας data-augmentation πετυχαίνουμε invariance σε παράγοντες όπως ακριβές χρώμα, φωτισμός, ακριβής τοποθεσία ενός αντικειμένου στην εικόνα. Ωστόσο, εμείς κάνουμε ακριβώς το αντίθετο, δηλαδή τα τελευταία layers, όχι μόνο δεν είναι invariant των μετασχηματισμών, αλλά αλλάζουν σύμφωνα με αυτούς με σκοπό να τους προβλέψουν.

Από την διαπίστωση αυτή και έπειτα η έρευνα επικεντρώθηκε από το να αναγνωρίζει έναν τυχαίο μετασχηματισμό στο να βρίσκει invariant representations για συγκεκριμένους μετασχηματισμούς. Αυτό το πετυχαίνει με contrastive μεθόδους που εξηγούνται στην επόμενη παράγραφο.



Σχήμα 4.1.6: **Αριστερά:** Καθώς αυξάνουμε τον αριθμό των permutations το mAP αυξάνει και στη συνέχεια πέφτει. **Δεξιά:** Καθώς κάνουμε probing σε πιο βαθιά layers του ResNet50 το mAP αυξάνει μέχρι ένα σημείο και έπειτα πέφτει.

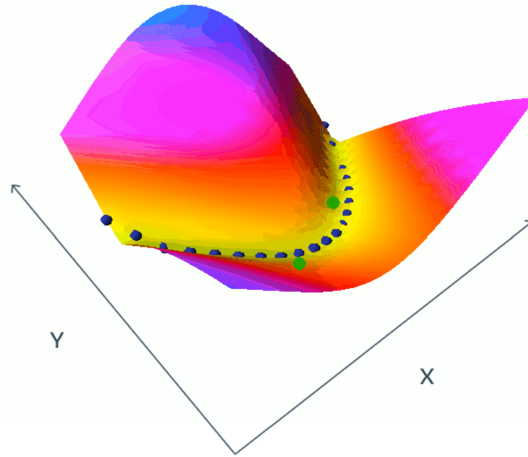
4.1.5 Contrastive energy-based SSL

Οι μέθοδοι αντίθεσης (contrastive methods) βασίζονται στην απλή ιδέα ότι μπορούμε να κατασκευάσουμε μη συμβατά ζεύγη (x, y) , και να προσαρμόσουμε τις παραμέτρους του μοντέλου έτσι ώστε η αντίστοιχη ενέργεια εξόδου να είναι μεγάλη.

Η εκπαίδευση ενός EBM με μια μέθοδο αντίθεσης συνίσταται στην ταυτόχρονη μείωση της ενέργειας των συμβατών ζευγών (x, y) από το σύνολο εκπαίδευσης (μπλε κουκίδες στο Σχήμα 4.1.7) και την αύξηση της ενέργειας ζευγών (x, y) που δεν είναι συμβατά (πράσινες κουκίδες). Έτσι πετυχαίνουμε features που (π.χ. για εικόνες):

- Αναπαριστούν πώς σχετίζονται οι εικόνες μεταξύ τους (ομοιότητα).
- Είναι ανθεκτικά σε παράγοντες όπως: ακριβής θέση αντικειμένων, φωτισμός, ακριβές χρώμα κλπ.

Στο Σχήμα 4.1.7, τα x και y είναι βαθμωτά μεγέθη, αλλά σε πραγματικές συνθήκες, τα x και y θα μπορούσαν να είναι μια εικόνα ή ένα βίντεο με εκατομμύρια διαστάσεις. Η δημιουργία ασύμβατων ζευγών που θα διαμορφώσουν την ενέργεια με κατάλληλους τρόπους είναι μια δύσκολη και ακριβή υπολογιστικά διαδικασία.



Σχήμα 4.1.7: Η ενέργεια είναι χαμηλή για τα συμβατά ζεύγη (μαύρες κουκκίδες), ενώ υψηλή για μη συμβατά x, y .

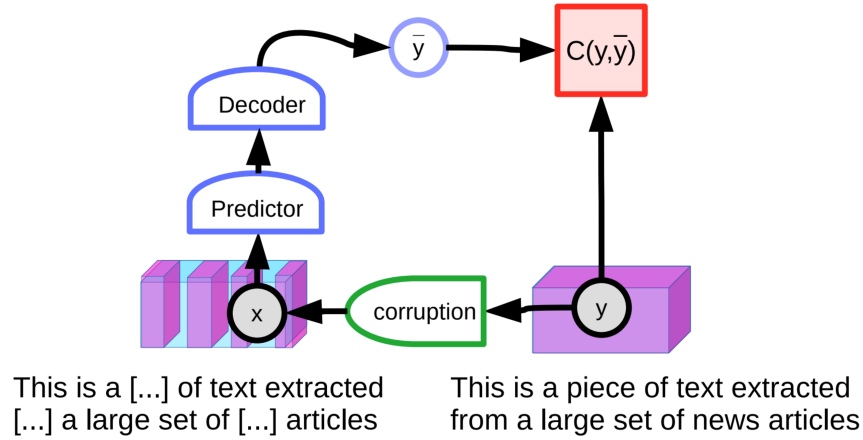
Παράδειγμα 4.1.3: BERT [Dev+19] ως contrastive EBM

Μια μεγάλη επιτυχία της αρχιτεκτονικής Transformer που είδαμε στην προηγούμενη παράγραφο είναι η χρήση του για εκπαίδευση συστημάτων NLP με κάλυψη ή αντικατάσταση ορισμένων λέξεων. Στην πραγματικότητα αυτή η μέθοδος είναι ένα contrastive EBM και ως μη χρησιμοποιεί joint embedding αρχιτεκτονική. Αντίθετα, χρησιμοποιείται predictive αρχιτεκτονική στην οποία το μοντέλο παράγει απευθείας μια πρόβλεψη για το y . Συγκεκριμένα ξεκινάμε με πλήρες τμήμα του κειμένου y και στη συνέχεια το αλλοιώνουμε, π.χ., καλύπτοντας κάποιες λέξεις για να παραχθεί ένα x . Η αλλοιωμένη είσοδος τροφοδοτείται στον transformer με σκοπό να ανακατασκευάσει το αρχικό κείμενο y . Επομένως:

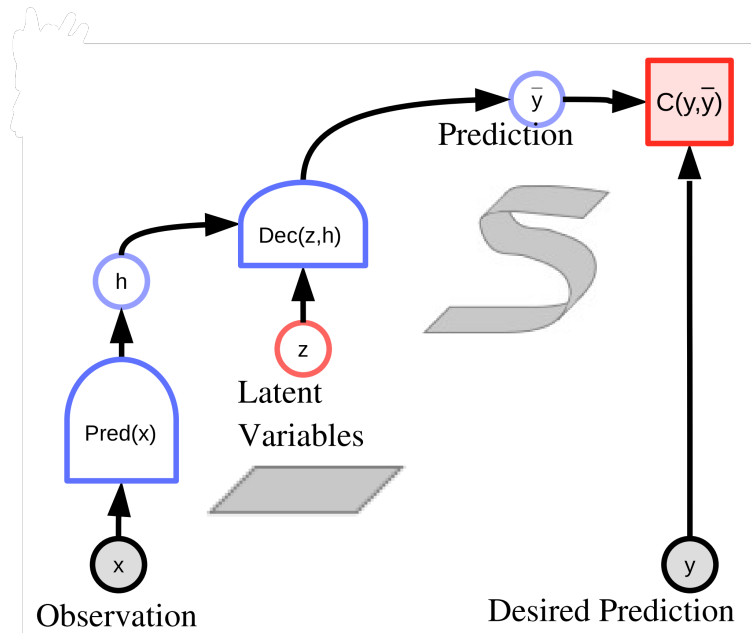
- Ένα μη αλλοιωμένο κείμενο θα ανακατασκευαστεί ως το ίδιο (χαμηλό σφάλμα ανακατασκευής), ενώ ένα κατεστραμμένο κείμενο θα ανακατασκευαστεί ως μη αλλοιωμένη έκδοση του εαυτού του (μεγάλο σφάλμα ανακατασκευής). Αν κάποιος ερμηνεύσει το σφάλμα ανακατασκευής ως ενέργεια, θα έχει την επιθυμητή ιδιότητα: χαμηλή ενέργεια για «καθαρό» κείμενο και υψηλότερη ενέργεια για «καθαρισμένο» κείμενο (Σχήμα 4.1.8).
- Η παραπάνω τεχνική δηλαδή η αλλοίωση της εισόδου με σκοπό την ανακατασκευή της ονομάζεται denoising auto-encoder [Vin+08; Col+11; Dev+19]. Η πρόβλεψη του παραπάνω μοντέλου δεν είναι ένα σύνολο λέξεων αλλά μια σειρά βαθμολογιών για κάθε λέξη στο λεξιλόγιο για κάθε θέση λέξης που λείπει.

Ορισμός 4.1.4: Denoising Auto-encoder ως EBM

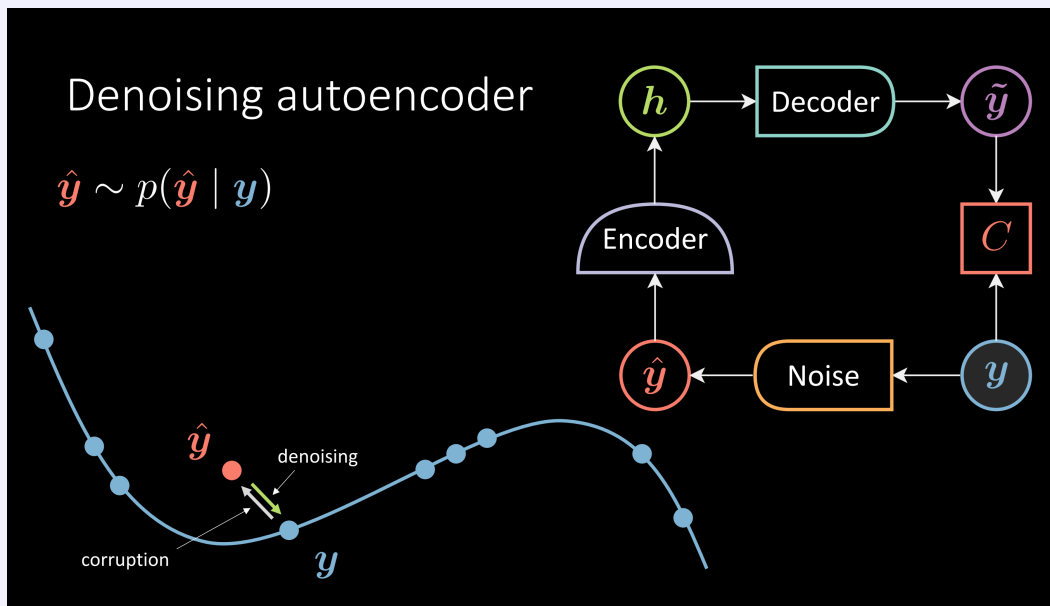
Είναι μία contrastive τεχνική, κατά την οποία τα negative samples παράγονται προσθέτοντας θόρυβο σε ένα sample. Στη συνέχεια προσπαθούμε να ανακατασκευάσουμε το corrupted sample, στην αρχική του τιμή. Η ενέργεια που δίνει το μοντέλο στο συγκεκριμένο sample είναι: $\|\mathbf{y} - \tilde{\mathbf{y}}\|^2$. Στην πραγματικότητα αποτρέπουμε το μοντέλο από το να κάνει collapse (ισοδύναμα να μπορεί να ανακατασκευάσει κάθε είσοδο) προσθέτοντας τεχνητό θόρυβο. Τύποι θορύβου μπορεί να είναι: Gaussian θόρυβος, ή μία μάσκα (dropout) που σβήνει λέξεις σε μία πρόταση κλπ. Ο θόρυβος που προστίθεται στην αρχική είσοδο θα πρέπει να είναι παρόμοιος με αυτόν που περιμένουμε στην πραγματικότητα.



Σχήμα 4.1.8: Ένα masked language model, είναι ένα instance των denoising auto-encoders. Η μεταβλητή y είναι ένα κομμάτι κειμένου. Το x είναι το ίδιο κείμενο με κάποιες από τις λέξεις να είναι masked. Το δίκτυο εκπαιδεύεται για να κάνει ανακατασκευή το κείμενο.



Σχήμα 4.1.9: Μια latent-variable predictive αρχιτεκτονική. Με δεδομένη μια παρατήρηση x , το μοντέλο πρέπει να μπορεί να παράγει ένα σύνολο πολλαπλών συμβατών προβλέψεων που συμβολίζονται από το σχήμα S στο διάγραμμα. Καθώς η latent μεταβλητή z ποικίλλει μέσα σε ένα σύνολο, που συμβολίζεται με ένα γκρι τετράγωνο, η έξοδος ποικίλλει σε σχέση με το σύνολο των προβλέψεων που θέλουμε.



Σχήμα 4.1.10

Συγκεκριμένα στο Σχήμα 4.1.10 ξεκινάμε από το y , το οποίο έχει χαμηλή ενέργεια και βρίσκεται επάνω στο manifold των training δεδομένων. Στη συνέχεια προσθέτουμε θόρυβο και παίρνουμε το \hat{y} το οποίο έχει υψηλή ενέργεια και δεν βρίσκεται πλέον στο manifold των δεδομένων. Το υπόλοιπο δίκτυο είναι ένας autoencoder που έχει σκοπό να ανακατασκευάσει την αρχική είσοδο ή ισοδύναμα να φέρει το \hat{y} πίσω στο manifold των δεδομένων.

Ωστόσο, δεν μπορούμε να χρησιμοποιήσουμε το τέχνασμα του παραδείγματος για εικόνες γιατί δεν μπορούμε να απαριθμήσουμε όλες τις πιθανές εικόνες. Δύο ενδιαφέρουσες λύσεις είναι οι joint embedding αρχιτεκτονικές και οι latent-variable predictive αρχιτεκτονικές.

Latent-variable predictive αρχιτεκτονικές

Στις latent-variable predictive αρχιτεκτονικές δίνεται μια παρατήρηση x και το μοντέλο μπορεί να παράγει ένα σύνολο πολλαπλών συμβατών προβλέψεων που συμβολίζονται από το σχέδιο S στο Σχήμα 4.1.9. Καθώς η latent μεταβλητή z παίρνει τιμές μέσα στο γκρι παραλληλόγραμμο του Σχήματος 4.1.9, η αντίστοιχη έξοδος κινείται στο σχήμα S . Συγκεκριμένα η z ονομάζεται latent γιατί η τιμή της δεν παρατηρείται ποτέ. Η λογική είναι ότι ένα εκπαιδευμένο μοντέλο, παράγει τέτοια h ώστε, όταν αυτά ενώνονται με την αντίστοιχη z μπορούν να γίνουν decode σε τιμές πάνω στο manifold των δεδομένων του training set.

Τα Latent-variable μοντέλα μπορούν να εκπαιδευτούν με contrastive μεθόδους. Ένα παράδειγμα αυτού είναι το GAN [ACB17; ZML17]. Ο critic (ή discriminator) υπολογίζει μια ενέργεια που υποδεικνύει εάν η είσοδος y φαίνεται καλή. Ο generator από την άλλη εκπαιδεύεται να παράγει δείγματα στα οποία ο discriminator έχει εκπαιδευτεί να συσχετίζει (λανθασμένα) υψηλή ενέργεια.

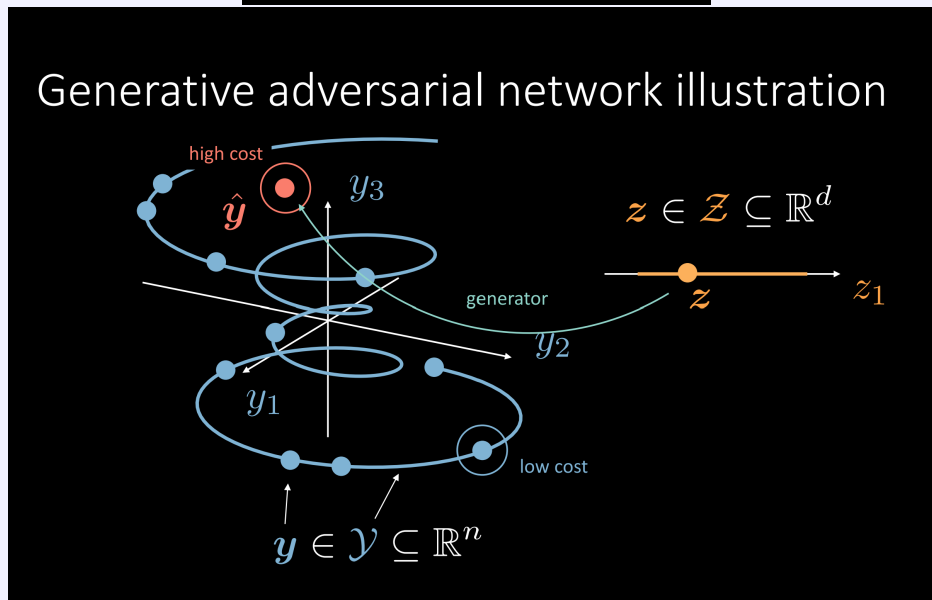
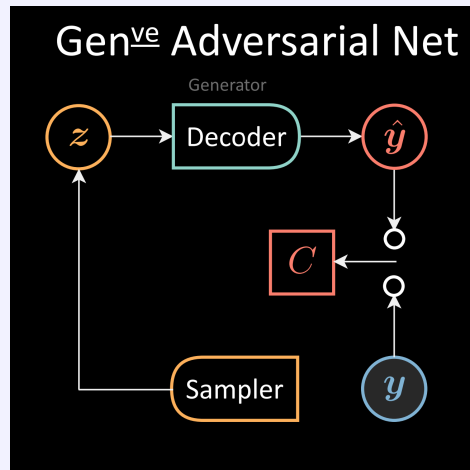
Ορισμός 4.1.5: Generative Adversarial Networks (GAN) ως EBM

Τα GAN έχουν την ίδια λογική με τους Denoising Auto-encoders (DAE) με κάποιες τροποποιήσεις.

- Ο DAE δημιουργεί negative δείγματα αλλοιώνοντας την είσοδο σύμφωνα με μία κατανομή θορύβου. Αντίθετα, τα GAN προσπερνούν το corruption της εισόδου και τα negative samples \hat{y} προκύπτουν ως έξοδος ενός δικτύου, το οποίο παίρνει ως είσοδο θόρυβο (Generator).
- Στον DAE το $C(\cdot)$ είναι η συνάρτηση $\|y - \tilde{y}\|^2$ με εισόδους την αρχική είσοδο και την ανακατασκευη \tilde{y} της (ταυτόχρονα).
- Στα GANs το $C(\cdot)$ είναι ένα νευρωνικό δίκτυο με εισόδους την αρχική είσοδο και την

“τεχνητή” generated είσοδο \hat{y} . Επίσης εδώ οι είσοδοι δεν μπαίνουν ταυτόχρονα στη συνάρτηση-νευρωνικό αλλά είτε η μία είτε η άλλη.

- Άρα έχουμε έναν discriminator που μας λέει αν η αντίστοιχη είσοδός του προέρχεται από το data manifold (έχει χαμηλή ενέργεια) ή το αντίθετο.



Σχήμα 4.1.11: **Πάνω:** Block διάγραμμα **Κάτω:** Διαισθητική απεικόνιση στο manifold των δεδομένων. Ο Generator απεικονίζει το latent space πίσω στο data space:

$$G: \mathcal{Z} \rightarrow \mathbb{R}^n, z \rightarrow \hat{y}$$

Η είσοδος y και το ανακατασκευασμένο \hat{y} τροφοδοτούνται στο δίκτυο κόστους (Discriminator) για να μετρηθεί η ασυμβατότητα:

$$C: \mathbb{R}^n \rightarrow \mathbb{R}, y \vee \hat{y} \rightarrow c$$

Το loss functional του Discriminator είναι:

$$\ell_C(y, \hat{y}) = C(y) + [m - C(\hat{y})]^+$$

Θέλουμε, δηλαδή να αυξήσουμε την ενέργεια του y (πραγματική είσοδος) και να μειώσουμε την ενέργεια του \hat{y} (το αποτέλεσμα του generator). Αυτό μέχρι να έχουν απόσταση m (αν $C \geq m$ δεν λαμβάνουμε

gradients, αφού η $\text{ReLU}(\cdot)$ δίνει έξοδο 0). Για το training του generator το αντίστοιχο συναρτησιακό είναι απλώς:

$$\ell_G(\mathbf{z}) = C(\mathbf{G}(\mathbf{z}))$$

Μια επιλογή για το $C(\mathbf{y})$ μπορεί να είναι:

$$C(\mathbf{y}) = \|\text{Dec}(\text{Enc}(\mathbf{y})) - \mathbf{y}\|^2$$

Συνοψίζοντας ο discriminator, ωθεί τα αυθεντικά δείγματα στο 0 και τα δείγματα του generator στο ενεργειακό επίπεδο m . Χρησιμοποιώντας το παραπάνω $C(\mathbf{y})$, έχουμε τετραγωνική απόσταση των σημείων στο manifold \mathbf{y} και των σημείων που δημιουργούνται από τον generator $\hat{\mathbf{y}}$. Κατά τη διάρκεια της εκπαίδευσης, ο generator ενημερώνεται για να προσπαθήσει να παράγει δείγματα που θα έχουν σταδιακά χαμηλή ενέργεια και άρα παράγει δείγματα όλο και πιο κοντά στο manifold των δεδομένων.

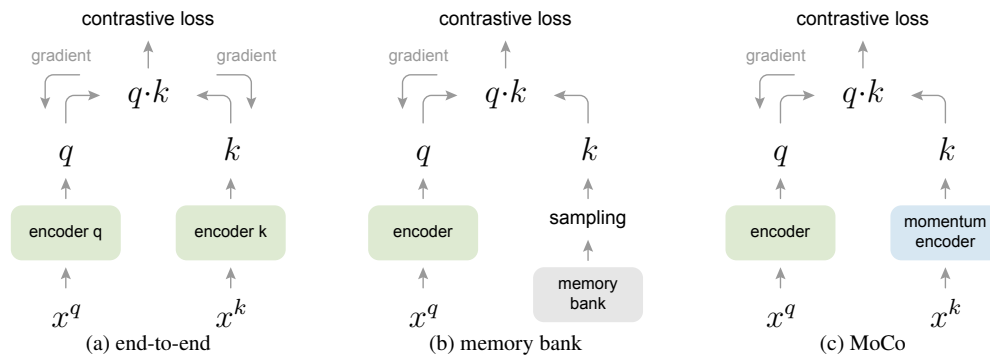
Joint embedding αρχιτεκτονικές

Σε αυτή την περίπτωση οι contrastive μέθοδοι, στην προσπάθειά τους να αποφύγουν το collapse, βασίζονται αποκλειστικά στα negative pairs. Συγκεκριμένα η διαδικασία μπορεί να θεωρηθεί ως εκπαίδευση ενός encoder για μια εργασία *dictionary look-up*, όπως περιγράφεται στη συνέχεια.

Έστω ένα κωδικοποιημένο query q και ένα σύνολο κωδικοποιημένων δειγμάτων $\{k_0, k_1, k_2, \dots\}$ που είναι τα keys ενός λεξικού. Ας υποθέσουμε ότι υπάρχει ένα μεμονωμένο κλειδί (που συμβολίζεται ως k_+) στο λεξικό που ταιριάζει με το q . Μία contrastive loss συνάρτηση είναι αυτή της οποίας η τιμή είναι χαμηλή όταν το q είναι παρόμοιο με το θετικό κλειδί k_+ και ανόμοιο με όλα τα άλλα κλειδιά (θεωρούνται αρνητικά κλειδιά για το q). Με την ομοιότητα να μετράται με το εσωτερικό γινόμενο, μια μορφή μιας contrastive loss function είναι η InfoNCE [OLV18].

$$\mathcal{L}_q = -\log \frac{\exp(q \cdot k_+ / \tau)}{\sum_{i=0}^K \exp(q \cdot k_i / \tau)}, \quad (4.1.1)$$

όπου τ είναι η υπερπαραμέτρος θερμοκρασίας. Το άθροισμα αφορά το ένα θετικό δείγμα και τα K αρνητικά δείγματα. Διαισθητικά, είναι το σφάλμα ενός $(K+1)$ -way softmax ταξινομητή, που προσπαθεί να ταξινομήσει το q ως k_+ .



Σχήμα 4.1.12: *End-to-end* vs *Memory bank* vs *Momentum update*.

Άρα, το contrastive learning μπορεί να θεωρηθεί ως ένας τρόπος δημιουργίας ενός λεξικού για συνεχείς εισόδους υψηλών διαστάσεων, όπως εικόνες. Το λεξικό είναι δυναμικό με την έννοια ότι τα κλειδιά δειγματοληπτούνται τυχαία και ότι ο κωδικοποιητής κλειδιών εξελίσσεται κατά τη διάρκεια της εκπαίδευσης. Για να μάθουμε χρήσιμα χαρακτηριστικά χρειαζόμαστε **μεγάλο** λεξικό που να καλύπτει ένα πλούσιο σύνολο αρνητικών δειγμάτων, ενώ ο κωδικοποιητής για τα κλειδιά του λεξικού να διατηρείται όσο το δυνατόν **συνεπής** παρά την εξέλιξη κατά την εκπαίδευση. Τρεις κατηγορίες αρχιτεκτονικών που πετυχαίνουν αυτό αναλύονται παρακάτω και οπτικοποιούνται στο Σχήμα 4.1.12.

- **End-to-end**, όπως για παράδειγμα στο SimCLR των Chen, T. et al. [Che+20]. Τα negative pairs είναι όλες οι εικόνες μέσα στο batch. **Μειονεκτήματα:** Θέλουμε μεγάλο batch-size (>8192) άρα πολλές GPU. Χρειάζεται επίσης ειδική μεταχείριση κατά το training (large mini-batch optimization [Goy+18]).
- **Memory bank**, όπως στο παράδειγμα των Wu, Z. et al. [Wu+18] Ικανοποιεί την ανάγκη για μεγάλο λεξικό με memory banks. Κρατάμε δηλαδή για κάθε παράδειγμα στο training set το momentum των μέχρι τώρα activations και σε κάθε forward pass παίρνουμε τα negatives από εκεί. Ο σκοπός του momentum update είναι για να κρατηθεί η συνέπεια των αναπαραστάσεων αφού σε κάθε forward pass τα representations ενημερώνονται. **Μειονεκτήματα:** Χρειάζεται πολύ GPU memory, επειδή κρατάμε για κάθε sample από ένα representation.
- **Momentum update.** Παράδειγμα το MoCo των He, K. et al. [He+20]. Χρησιμοποιεί ουρά (queue) και σαν αποτέλεσμα, μπορεί να έχει μεγάλο λεξικό. Τα δείγματα στο λεξικό αντικαθίστανται σταδιακά. Σε κάθε επανάληψη το τρέχον mini-batch μπαίνει στην ουρά ενώ το παλαιότερο mini-batch αφαιρείται. Αυτό μας επιτρέπει να επαναχρησιμοποιούμε τα κωδικοποιημένα κλειδιά από τα προηγούμενα mini-batches και κάνει ανεξάρτητο το μέγεθος του λεξικού από το μέγεθος του mini-batch. Επίσης, η αφαίρεση του παλαιότερου mini-batch μπορεί να είναι ωφέλιμη, επειδή τα κωδικοποιημένα κλειδιά του είναι τα πιο παλιά και επομένως τα λιγότερο συνεπή με τα νεότερα. Σε κάθε επανάληψη το λεξικό αντιπροσωπεύει πάντα ένα υποσύνολο όλων των δεδομένων.

Ωστόσο, δεν μπορούμε να κάνουμε backpropagation λόγω της ουράς. Μια λύση είναι να αντιγράψουμε τα βάρη του κωδικοποιητή κλειδιών f_k από τον κωδικοποιητή των queries f_q . Όμως έτσι έχουμε ταχέως μεταβαλλόμενο κωδικοποιητή που μειώνει τη συνέπεια των αναπαραστάσεων. Άρα χρησιμοποιούμε momentum update, δηλαδή για τις παραμέτρους του f_k , θ_k και εκείνες του f_q , θ_q , ενημερώνουμε το θ_k ως:

$$\theta_k \leftarrow m\theta_k + (1 - m)\theta_q.$$

Εδώ το $m \in [0, 1)$ είναι ο συντελεστής ορμής. Μόνο οι παράμετροι θ_q ενημερώνονται με backpropagation. Η ενημέρωση ορμής κάνει το θ_k να εξελίσσεται πιο ομαλά από το θ_q .

4.1.6 Non-contrastive energy-based SSL

Παρακάτω θα δούμε μεθόδους που μέσω περιορισμών στην αρχιτεκτονική του μοντέλου, επιβεβαιώνουν ότι η ενέργεια των ασύμβατων ζευγών είναι υψηλότερη από εκείνη των συμβατών ζευγών, χωρίς την ανάγκη των negative samples.

Clustering:

Παραδείγματα: DeeperCluster [Car+19], ClusterFit [Yan+19], SwAV [Car+21], SEER (SwAV on 1.3 billion random/non-filtered images) [Goy+21]. Η ιδέα είναι ότι οι contrastive learning μέθοδοι δημιουργούν συστάδες στον χώρο των embeddings, όπως φαίνεται στο Σχήμα 4.1.15. Η πιο πρόσφατη μέθοδος είναι το SwAV [Car+21] και είναι αυτή που δίνει τα καλύτερα αποτελέσματα. Συγκεκριμένα, έχουμε prototypes (cluster centers) και ελέγχουμε σε ποιο cluster ανήκει κάθε sample. Ίδανικά θέλουμε οι όμοιες εικόνες να μπου στο ίδιο cluster ανεξαρτήτως του augmentation που εφαρμόζουμε όπως στο Σχήμα 4.1.16a. Δυστυχώς όμως το μοντέλο εύκολα μπορεί να πέσει σε trivial solution ή ισοδύναμα το EBM να κάνει collapse 4.1.16b. Εδώ έχουμε non-contrastive μέθοδο δηλαδή δεν έχουμε negative samples, οπότε το collapse αποφεύγεται μέσω περιορισμών στην αρχιτεκτονική του μοντέλου. Συγκεκριμένα στο SwAV αυτό λύνεται με τον περιορισμό ότι κάθε prototype θα έχει το μέγιστο N/K samples, όπου N : αριθμός των samples και K : αριθμός των prototypes. Αυτό λύνεται μέσω της μεθόδου Optimal Transport και συγκεκριμένα του αλγορίθμου Sinkhorn-Knopp [Cut13] για λόγους ταχύτητας. Τέλος αντί να κάνουμε hard assignment δηλαδή κάθε sample να ανήκει μόνο σε ένα prototype έχουμε μία κατανομή από assignments (soft-assignment) όπως φαίνεται στο Σχήμα 4.1.16c. Έτσι πετυχαίνουμε να έχουμε πολλές περισσότερες κλάσεις από τις αρχικές K (όπου κάθε sample ανήκει σε ένα prototype), δηλαδή έχουμε κλάσεις της μορφής $[0.8, 0.1, 0.1]$ όπου το sample ανήκει με τα αντίστοιχα βάρη στα αντίστοιχα prototypes. Ως αποτέλεσμα αποφεύγουμε class imbalances και έχουμε πιο dense codes στο prototype space.

Τέλος σε κάθε βήμα μετά το clustering προσπαθούμε να προβλέψουμε το code της augmented εικόνας από το embedding της αυθεντικής και αντίστροφα. Αυτό με τη λογική ότι παρόμοιες εικόνες έχουν παρόμοια codes.

Έτσι μπορούμε να κάνουμε backpropagation και ως προς τα embeddings και ως προς τα prototypes άρα το clustering είναι online.

Distillation:

Παραδείγματα: BYOL [Gri+20] και SimSiam [CH20]. Βασίζονται στο Knowledge distillation [HVD15] άρα βλέπουν το πρόβλημα ως ένα student-teacher πρόβλημα:

$$f_{\theta}^{\text{student}}(I) = f_{\theta}^{\text{teacher}}(\text{augment}(I))$$

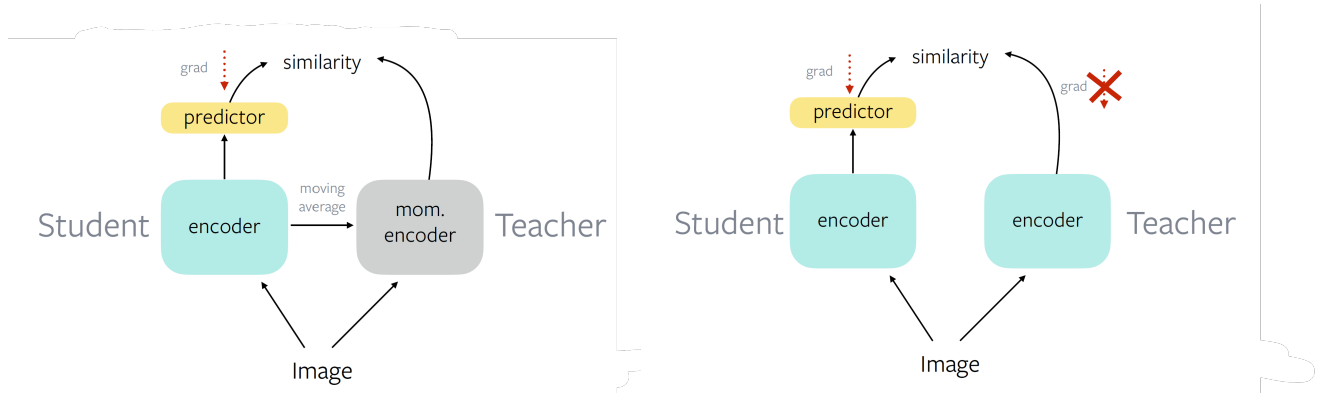
Είδαμε στην Παράγραφο 4.1.5 ότι χωρίς negative samples και εάν τα δύο δίκτυα στο siamese network έχουν ίδια αρχιτεκτονική τότε έχουμε trivial solution. Τα Distillation μοντέλα αποφεύγουν το collapse του EBM μέσω:

- Ασύμμετρο κανόνας μάθησης μεταξύ μαθητή δασκάλου.
- Ασύμμετρης αρχιτεκτονικής μεταξύ μαθητή και δασκάλου.

Ένα τέτοιο παράδειγμα είναι το BYOL [Gri+20] (Σχήμα 4.1.13 αριστερά), στο οποίο έχουμε:

- Ένα επιπλέον prediction head μετά τον encoder του Student ενώ ο teacher έχει μόνο encoder, άρα έχουμε **ασυμμετρία στην αρχιτεκτονική**.
- Τα gradients περνούν μόνο μέσα από τον Student άρα έχουμε **ασυμμετρία στον κανόνα μάθησης**.
- Τέλος, τα βάρη του Teacher ορίζονται ως moving average αυτών του Student, όπως στο MoCo [He+20], που περιγράφεται στην Παράγραφο 4.1.5. Άρα έχουμε **ασυμμετρία ως προς τα βάρη των μοντέλων**.

Μια βελτίωση του παραπάνω είναι το SimSiam (Σχήμα 4.1.13 δεξιά) που απέδειξε ότι δεν χρειαζόμαστε και τα τρία είδη ασυμμετρίας. Συγκεκριμένα ότι είναι αχρείαστο να έχουμε ασυμμετρία στα βάρη των δύο μοντέλων.



Σχήμα 4.1.13: Αριστερά: BYOL Δεξιά: SimSiam

Redundancy reduction:

Παράδειγμα: Barlow Twins [Zbo+21]. Είναι μία πολύ απλή μέθοδος που δεν χρησιμοποιεί ούτε negative sampling, ούτε κάποιου είδους ασυμμετρία. Η ιδέα είναι ότι αφού έχουμε περιορισμένο αριθμό νευρώνων θέλουμε ο κάθε ένας από αυτούς να κωδικοποιεί διαφορετική πληροφορία, δηλαδή αν N νευρώνες παράγουν μια αναπαράσταση (embedding), θέλουμε κάθε νευρώνας να ικανοποιεί:

- invariance: Να είναι αμετάβλητος υπό διαφορετικό data augmentation.
- independence: Να είναι ανεξάρτητος από τους άλλους νευρώνες.

Ισοδύναμα:

$$f_{\theta}(I)[i] = f_{\theta}(\text{augment}(I))[i]$$

$$f_{\theta}(I)[i] \neq f_{\theta}(\text{augment}(I))[j]$$

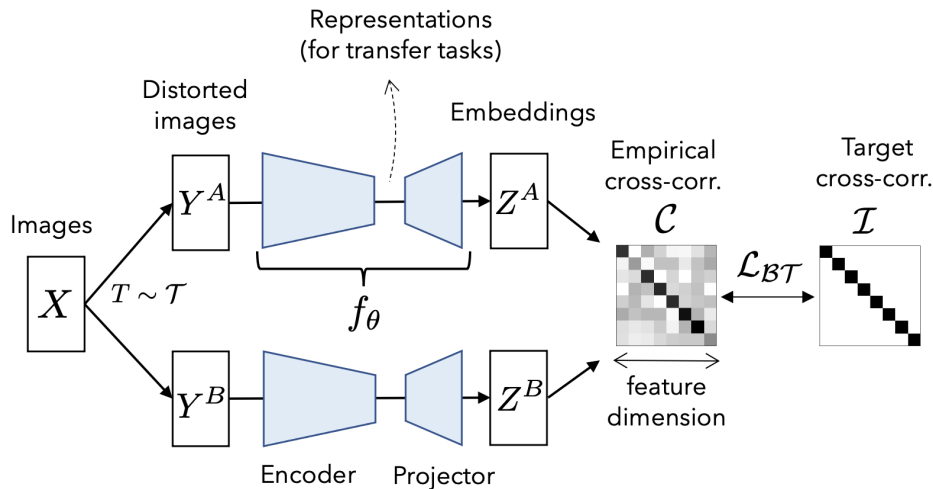
Δηλαδή το collapse του EBM αποφεύγεται σε αυτή τη μέθοδο μέσω του περιορισμού για ανεξαρτησία μεταξύ των νευρώνων. Όπως φαίνεται από το Σχήμα 4.1.14 προσπαθούμε να ελαχιστοποιήσουμε την απόσταση μεταξύ του cross-correlation πίνακα των embeddings μεταξύ του μοναδιαίου πίνακα. Αυτό το πετυχαίνουμε μέσω της συνάρτησης \mathcal{L}_{BT} :

$$\mathcal{L}_{BT} \triangleq \underbrace{\sum_i (1 - C_{ii})^2}_{\text{invariance term}} + \lambda \underbrace{\sum_i \sum_{j \neq i} C_{ij}^2}_{\text{redundancy reduction term}} \quad (4.1.2)$$

όπου λ είναι μια θετική σταθερά που καθορίζει τη σημασία του πρώτου και του δεύτερου όρου στο loss. Αυτό γιατί έχουμε μόνο N invariance όρους (διαγώνιος), ενώ $N^2 - N$ redundancy όρους. Άρα το λ προσπαθεί να αποτρέψει το δίκτυο από το να εστιάσει μόνο στους redundancy όρους. Τέλος, C είναι ο cross-correlation πίνακας, που υπολογίζεται μεταξύ των εξόδων των δύο πανομοιότυπων δικτύων κατά μήκος του batch dimension:

$$C_{ij} \triangleq \frac{\sum_b z_{b,i}^A z_{b,j}^B}{\sqrt{\sum_b (z_{b,i}^A)^2} \sqrt{\sum_b (z_{b,j}^B)^2}} \quad (4.1.3)$$

όπου b το batch index, i, j indices των εξόδων των δικτύων.



Σχήμα 4.1.14: Οπτικοποίηση της μεθόδου Barlow Twins.

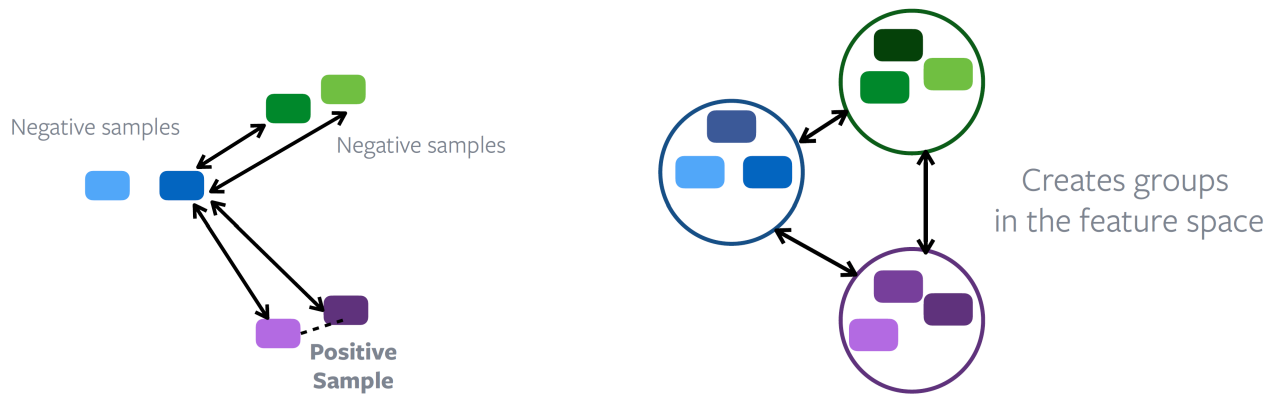
Variational Autoencoders (VAEs):

Μία ενδιαφέρουσα αρχιτεκτονική είναι οι contrastive μέθοδοι με latent μεταβλητές. Η ιδέα είναι ότι πρέπει να βρούμε κάποιον τρόπο για την ελαχιστοποίηση της χωρητικότητας της latent μεταβλητής. Ο όγκος του συνόλου στο οποίο μπορεί να ποικίλλει η latent μεταβλητή περιορίζει τον όγκο των εξόδων που λαμβάνουν χαμηλή ενέργεια. Ελαχιστοποιώντας αυτόν τον όγκο, διαμορφώνει κανείς αυτόματα την ενέργεια με τον σωστό τρόπο και αποφεύγει το collapse του EBM. Ένα επιτυχημένο παράδειγμα μιας τέτοιας μεθόδου είναι οι VAE's που εξηγούνται παρακάτω:

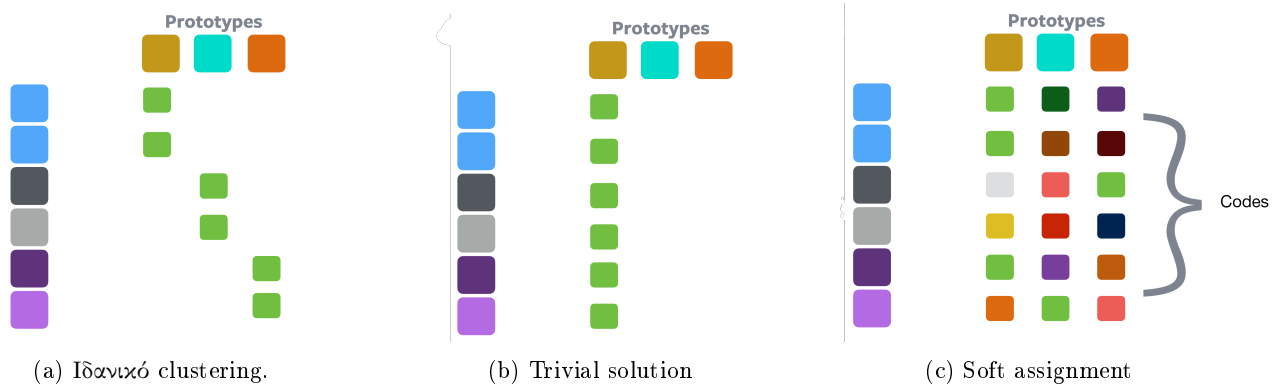
Ορισμός 4.1.6: Variational Auto-Encoder (VAE) ως EBM

Διαφορές του VAE από vanilla AE, που περιγράφηκε στην Παράγραφο 4.1.3:

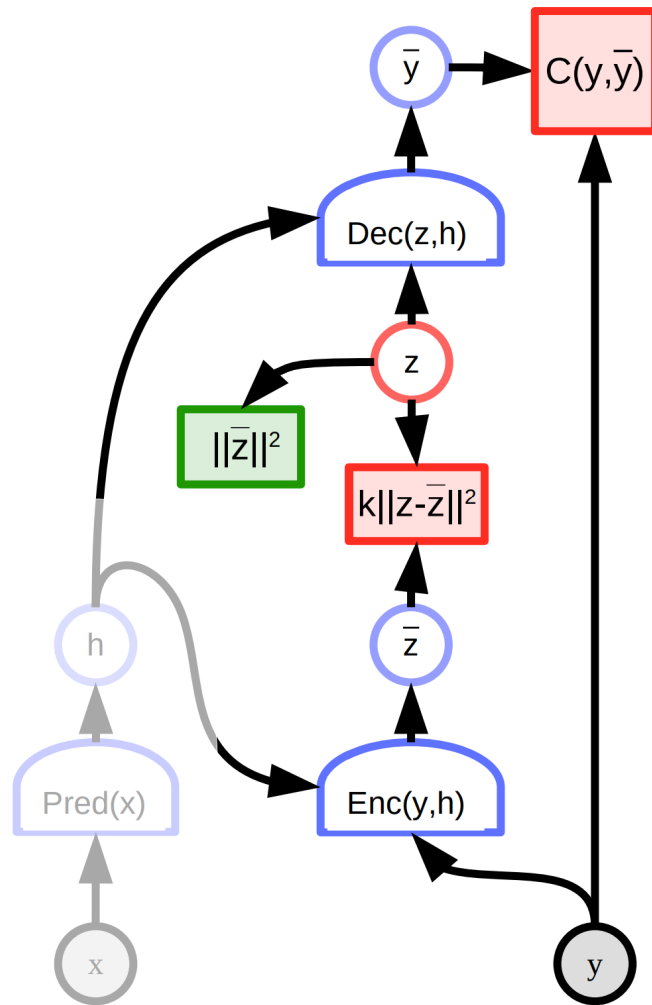
- Στον VAE περνάμε την είσοδο y στον encoder. Αυτός, αντί να επιστρέψει ένα διάνυσμα h όπως στον απλό AE, επιστρέφει δύο concatenated hidden representations: μ και v . Το κάθε ένα από



Σχήμα 4.1.15: **Αριστερά:** Contrastive learning στο feature space. Π.χ. τα μπλε σχήματα είναι μία εικόνα και οι μετασχηματισμοί (augmentations) της. **Δεξιά:** Μπορούμε να πετύχουμε το ίδιο αποτέλεσμα με clustering.



Σχήμα 4.1.16: Οπτικοποίηση της μεθόδου SwAV.

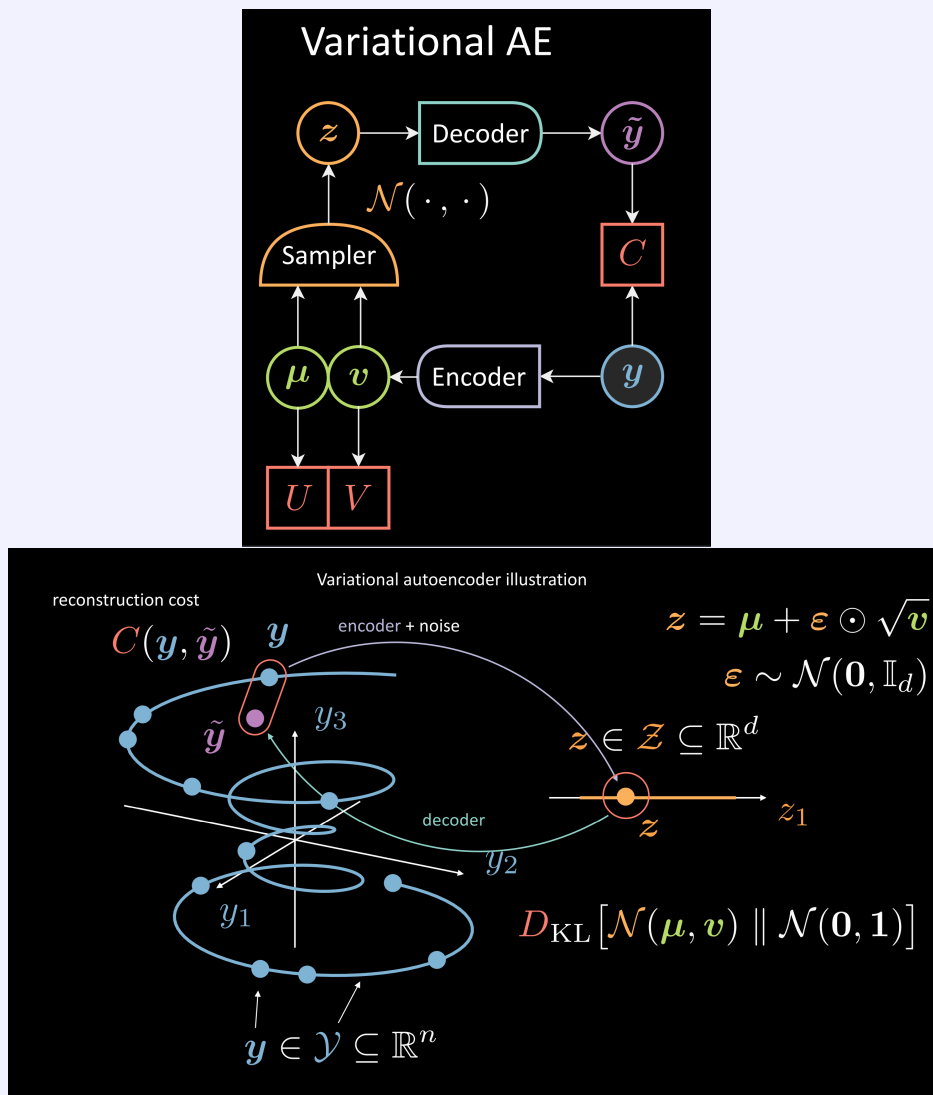


Σχήμα 4.1.17: Ελαχιστοποίηση της χωρητικότητας της latent μεταβλητής μέσω της προσθήκης Gaussian θορύβου. Ο όρος $k\|z - \bar{z}\|^2$ μπορεί να θεωρηθεί σαν log μίας prior από όπου κάνουμε sample την z . Η έξοδος του encoder γίνεται regularized, ώστε να έχει mean = 0.

αυτά έχει από έναν regularisation όρο, συγκεκριμένα U και V για τα μ και v , αντίστοιχα.

- Στη συνέχεια χρησιμοποιούμε τον sampler για να κάνουμε sample το z το οποίο είναι μια latent random μεταβλητή που ακολουθεί Gaussian κατανομή με μέσο μ και variance v .
- Έπειτα το z μπαίνει σαν είσοδο στον decoder για να δώσει ως έξοδο το \tilde{y} .
- Ο decoder είναι μία συνάρτηση από το Z στο \mathbb{R}^n : $z \mapsto \tilde{y}$.

Από το παραπάνω προκύπτει ότι ο vanilla autoencoder που είδαμε έχει variance 0 δηλαδή $h = \mu$. Αντίθετα ο VAE έχει fuzzy latent space πράγμα που περιορίζει το capacity (δεν μπορεί απλώς να απομνημονεύσει την είσοδο), δηλαδή δρα σαν επιπλέον regularization, πράγμα που θέλουμε.



Σχήμα 4.1.18: Πάνω: Block διάγραμμα Κάτω: Διαισθητική απεικόνιση στο manifold των δεδομένων.

Σύγκριση VAE με DAE:

Στον DAE το sampling λαμβάνει χώρα μεταξύ y και \hat{y} . Δηλαδή μετακινούμε την είσοδο εκτός του training manifold 4.1.10 και ο encoder-decoder επαναφέρει το corrupted \hat{y} στο \tilde{y} . Αντίθετα στον VAE, κωδικοποιούμε την είσοδο και προσθέτουμε θόρυβο στη συνέχεια. Τέλος, ο decoder αποκωδικοποιεί το z στο \tilde{y} .

Συνάρτηση σφάλματος:

Η συνάρτηση λάθους, στον VAE αποτελείται από έναν όρο ανακατασκευής, καθώς και από έναν regularization όρο.

- Ο όρος ανακατασκευής είναι το τετράγωνο $C(\mathbf{y}, \tilde{\mathbf{y}})$ του Σχήματος 4.1.18.
- Ο regularization όρος, αφορά το latent layer και επιβάλλει Gaussian δομή στον latent χώρο. Αυτό γίνεται μέσω της συνάρτησης $D_{KL}(\mathcal{N}(\boldsymbol{\mu}, \mathbf{v}) \parallel \mathcal{N}(\mathbf{0}, \mathbf{1}))$ που κάνει την κατανομή της \mathbf{z} να μην αποκλίνει από την $\mathcal{N}(\mathbf{0}, \mathbf{1})$. Χωρίς αυτό τον όρο το \mathbf{z} θα μπορούσε να πάρει αυθαίρετα μεγάλες μέσες τιμές και variances που οδηγούν σε overfitting και τελικά σε collapse του συστήματος.

Οπτικοποίηση συνάρτησης σφάλματος:

Στο Σχήμα 4.1.19, κάθε κύκλος αντιπροσωπεύει μια εκτιμώμενη περιοχή του \mathbf{z} . Όπως είπαμε πριν θέλουμε να κάνουμε ελαχιστοποίηση δύο όρων: τον reconstruction όρο και τον regularization όρο. Αυτό γράφεται:

$$\tilde{F}(\mathbf{y}) = C(\mathbf{y}, \tilde{\mathbf{y}}) + \beta D_{KL}(\mathcal{N}(\boldsymbol{\mu}, \mathbf{v}) \parallel \mathcal{N}(\mathbf{0}, \mathbf{1}))$$

Για να οπτικοποιήσουμε τον σκοπό κάθε όρου, μπορούμε να σκεφτούμε το \mathbf{z} ως κύκλο στον $2d$, χώρο, όπου το κέντρο του κύκλου είναι $\boldsymbol{\mu}$ και η γύρω περιοχή είναι οι πιθανές τιμές του \mathbf{z} με ακτίνα ανάλογη του \mathbf{v} . Εάν υπάρχει επικάλυψη μεταξύ οποιωνδήποτε δύο εκτιμήσεων του \mathbf{z} (οπτικά, εάν επικάλυπτονται δύο κύκλοι), αυτό δημιουργεί ασάφεια για την ανακατασκευή επειδή τα σημεία στην επικάλυψη μπορούν να αντιστοιχίσουν και στις δύο αρχικές εισόδους. Άρα ο όρος ανακατασκευής $C(\mathbf{y}, \tilde{\mathbf{y}})$ προσπαθεί διαισθητικά:

- Να απομακρύνει τα σημεία το ένα από το άλλο, ώστε να μην υπάρχει επικάλυψη.
- Να μηδενίσει τη διακύμανση \mathbf{v} και οι κύκλοι να γίνουν σημεία, ώστε να μην υπάρχει και πάλι επικάλυψη.

Άρα, εάν χρησιμοποιήσουμε μόνο τον όρο ανακατασκευής, οι εκτιμήσεις θα απομακρύνονται αυθαίρετα μακριά η μία από την άλλη. Οπότε χρειάζεται ο regularization όρος. Ο δεύτερος όρος είναι η relative entropy (ένα μέτρο της απόστασης μεταξύ δύο κατανομών) μεταξύ μίας Gaussian με μέσο όρο $\boldsymbol{\mu}$ και διακύμανση \mathbf{v} , και της $\mathcal{N}(\mathbf{0}, \mathbf{1})$. Συγκεκριμένα:

$$D_{KL}(\mathcal{N}(\boldsymbol{\mu}, \mathbf{v}) \parallel \mathcal{N}(\mathbf{0}, \mathbf{1})) = \frac{1}{2} \sum_{i=1}^d v_i - \log(v_i) - 1 + \mu_i^2$$

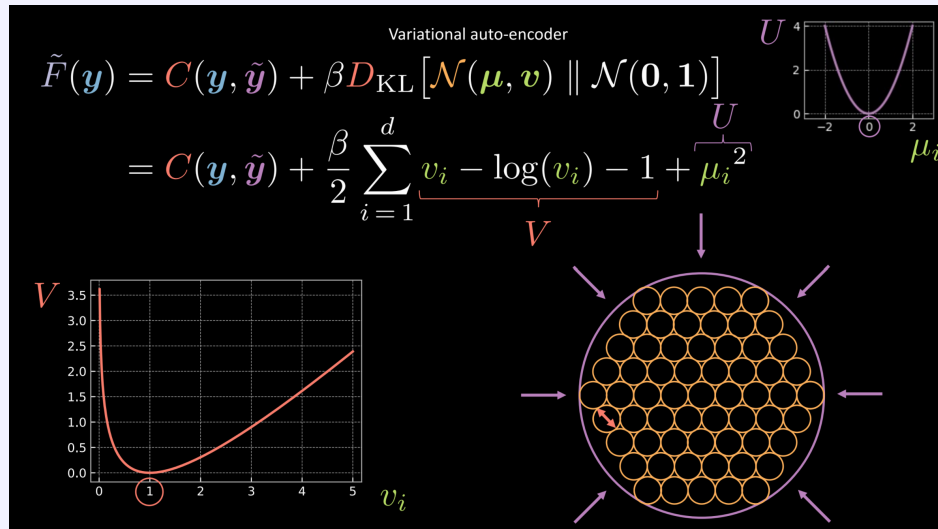
Αν θέσουμε:

$$V = v_i - \log(v_i) - 1$$

Φαίνεται και στο Σχήμα 4.1.19 ότι αυτή η έκφραση ελαχιστοποιείται όταν το $v_i = 1$. Επομένως, ο regularization όρος θα διατηρήσει τη διακύμανση των εκτιμώμενων latent μεταβλητών μας περίπου στο 1 ή ισοδύναμα οι κύκλοι θα έχουν ακτίνα περίπου 1. Όμοια ο μ_i^2 , προσπαθεί να ελαχιστοποιήσει την απόσταση μεταξύ των κύκλων και επομένως αποτρέπει τον όρο ανακατασκευής να να τους μετατοπίσει αυθαίρετα μακριά. Τέλος, ο όρος β είναι μία υπερπαραμέτρος που αφορά το πόσο βάρος θα δώσουμε στον reconstruction όρο και πόσο στον regularization όρο.

Reparameterisation trick:

Όταν εφαρμόζουμε gradient descent για να κάνουμε train τον VAE δεν υπάρχει τρόπος να κάνουμε backpropagation στο sampling module. Αυτό λύνεται εύκολα αν πούμε ότι το \mathbf{z} ισούται με $\boldsymbol{\mu}$ που είναι ο sampled μέσος όρος, συν ϵ που γίνεται sample από μία Gaussian πολλαπλασιασμένο με την τετραγωνική ρίζα του variance \mathbf{v} .



Σχήμα 4.1.19: Διασθητική απεικόνιση του regularization στον VAE.

Κεφάλαιο 5

Περιγραφή μεθόδων

5.1	Εισαγωγή	66
5.2	Hand-Crafted Feature Engineering	66
5.3	Random convolutional kernels	66
5.4	InceptionTime	67
5.4.1	Inception modules	67
5.4.2	Inception network	68
5.4.3	InceptionTime: a neural network ensemble for TSC	68
5.5	Self-supervised time series representation learning by inter-intra relational reasoning	69
5.5.1	Μέθοδος	70
5.6	A Transformer-based Framework for Multivariate Time Series Representation Learning	72
5.6.1	Πλεονεκτήματα των transformers έναντι άλλων αρχιτεκτονικών	73
5.6.2	Μεθοδολογία	73
5.6.3	Πειράματα & Αποτελέσματα των Zerveas et al.	79
5.7	Encoding Time Series as Images for Visual Inspection and Classification	80
5.7.1	Gramian Angular Field	81
5.7.2	Markov Transition Field	83
5.7.3	Σύγκριση και ανάλυση	84
5.8	Survival Analysis	85
5.8.1	Εισαγωγή	85
5.8.2	Ορισμοί	87
5.8.3	Εφαρμογές του survival analysis	87

5.1 Εισαγωγή

Σε αυτή την παράγραφο παρουσιάζονται οι μέθοδοι τις οποίες εφαρμόσαμε στα δεδομένα του e-Prevention. Ως είσοδο σε αυτές τις μεθόδους δίνουμε πολυμεταβλητές χρονοσειρές ή χρονοσειρές μίας μεταβλητής όπως ορίζονται τυπικά παρακάτω:

Ορισμός 5.1.1: Τυπικοί ορισμοί:

- Μια Πολυμεταβλητή χρονοσειρά (Multivariate Time Series - MTS) $X = [X_1, X_2, \dots, X_T]$ με M διαστάσεις, αποτελείται από T διατεταγμένες τιμές $X_i \in \mathbb{R}^M$.
- Μια χρονοσειρά μίας μεταβλητής (Univariate) X μήκους T είναι απλώς μία MTS με $M = 1$.
- Ένα σύνολο δεδομένων $D = \{(X^1, Y^1), (X^2, Y^2), \dots, (X^N, Y^N)\}$, περιέχει ζεύγη (X^i, Y^i) όπου X^i είναι χρονοσειρά μίας ή πολλών μεταβλητών και Y^i η αντίστοιχη επισήμειωση.

Το πρόβλημα που ορίζεται στο dataset D ονομάζεται supervised timeseries classification (TSC).

5.2 Hand-Crafted Feature Engineering

Έχουμε ένα τέτοιο σύνολο χρονοσειρών D , τις οποίες θέλουμε να αναπαραστήσουμε και για τις οποίες δεν διαθέτουμε κάποια επισήμειωση. Ένας τρόπος να αναπαραστήσουμε μία χρονοσειρά X^i , είναι μέσω συναρτήσεων $\theta_k : \mathbb{R}^{n_t} \rightarrow \mathbb{R}$ που περιγράφουν την κατανομή των σημείων της. Κάθε θ_k αποτυπώνει συγκεκριμένο χαρακτηριστικό της χρονοσειράς και στη συνέχεια η ένωση τους αποτελεί την τελική αναπαράσταση.

Έστω, τώρα, ότι μας δίνονται ετικέτες $Y = (y_1, \dots, y_N)^T$ για κάποιο πρόβλημα. Δυστυχώς, έχει δείχθει συγκεκριμένες συναρτήσεις αποδίδουν καλά σε διαφορετικούς τύπους προβλημάτων [Dau+19]. Μερικά τέτοια παραδείγματα είναι:

- **Αρχεία ήχου:** MFCC's και χαρακτηριστικά που έχουν να κάνουν με τις κορυφές του σήματος [MM05].
- **Vibration Monitoring:** Χαρακτηριστικά βασισμένα στον Wavelet μετασχηματισμό [YL00].
- **Υπολειπόμενη διάρκεια ζωής των ρουλεμάν:** Χαρακτηριστικά που προκύπτουν από την fitted exponential επάνω στη χρονοσειρά [Geb+04].
- **Ανίχνευση τόξων στις λωρίδες επαφής των τρένων υψηλής ταχύτητας:** Χαρακτηριστικά που προκύπτουν από το περιοδόγραμμα σε λογαριθμική κλίμακα [Bar+14].

Στις εργασίες των Fulcher et al. [FJ14] και Nun et al. [Nun+15] γίνεται εκτενής ανάλυση τέτοιων χαρακτηριστικών και για διαφορετικά domains και tasks. Επομένως, εφόσον δεν γνωρίζουμε εξ' αρχής το τελικό πρόβλημα, δεν υπάρχει συγκεκριμένο σύνολο συναρτήσεων που να αποδίδει καλά σε οποιοδήποτε "downstream task" [Bag+15]. Δύο λύσεις σε αυτό είναι, όταν τελικά μας δοθεί το τελικό task:

- Να εκπαιδύσουμε διαφορετικούς classifiers με είσοδο διαφορετικά representations και να συνδυάσουμε τις προβλέψεις τους μέσω ensembling. Η μέθοδος αυτή ονομάζεται COTE [Bag+15] και η βελτίωσή της HIVE-COTE [LTB18] πετυχαίνει εξαιρετικά ακριβή αποτελέσματα στο UCR archive [Dau+19]. Δυστυχώς όμως ο χρόνος εκπαίδευσής της είναι $\mathcal{O}(N^2 * T^4)$, πράγμα που την καθιστά μη εφαρμόσιμη σε σύνολα χρονοσειρών με πολλά samples ή με μεγάλης διάρκειας χρονοσειρές.
- Να παράγουμε όλα τα features που έχουν αποδειχθεί ότι αποδίδουν για διαφορετικά tasks [Chr+18] και στη συνέχεια να κάνουμε selection με βάση την προγνωστική ισχύ των features στο συγκεκριμένο task. Ένας αλγόριθμος που ειδικεύεται σε features χρονοσειρών για selection σε tasks κατηγοριοποίησης και παλινδρόμησης είναι ο αλγόριθμος fresh [CKF17].

5.3 Random convolutional kernels

Οι πυρήνες συνέλιξης που είδαμε στην παράγραφο 3.6 μπορούν να αποτυπώσουν διάφορα μοτίβα σε μία χρονοσειρά, δίνοντας μεγάλες τιμές εξόδου στις περιοχές όπου η είσοδος ταιριάζει με το κάθε μοτίβο. Επιπλέον, η τεχνική του dilation, επιτρέπει στον πυρήνα να αποτυπώνει το ίδιο pattern σε διαφορετικά scales [YK15] και να αντλεί πληροφορία από πεδίο της συχνότητας: μεγαλύτερα dilations αντιστοιχούν σε χαμηλές συχνότητες, ενώ

μικρότερα σε υψηλότερες συχνότητες. Τέλος, ο pooling μηχανισμός κάνει τους πυρήνες invariant ως προς την ακριβή θέση του μοτίβου στη χρονοσειρά και ο συνδυασμός πολλών kernels μπορεί να αναγνωρίσει πολύπλοκα patterns της εισόδου. Οι kernels που μαθαίνονται στα συνελικτικά νευρωνικά δίκτυα πολλές φορές αντιστοιχούν σε φίλτρα στο πεδίο της συχνότητας [KSH12b; Yos+14; ZF14]. Επιπλέον, οι Saxe et al. [Sax+11] αποδεικνύουν ότι ακόμη και οι τυχαίοι πυρήνες αποτελούν frequency selective φίλτρα.

Τα βάρη των kernels στις περισσότερες μεθόδους εκπαιδεύονται. Ωστόσο, δεν είναι λίγες οι μέθοδοι που χρησιμοποιούν τυχαίους συνελικτικούς πυρήνες [Jar+09; Pin+09; Sax+11; CP11]. Μια παρατήρηση των Ismail Fawaz et al. [Ism+20] είναι ότι τα συνελικτικά δίκτυα παρουσιάζουν μεγάλη διακύμανση ως προς το classification accuracy στο UCR archive και η λύση που προτείνουν είναι χρήση πολλών διαφορετικών kernels (και ensembles πολλών μοντέλων). Φαίνεται, λοιπόν, ότι είναι δύσκολο να βρεθούν χρήσιμοι πυρήνες για μικρά datasets. Οπότε είναι λογικό να δοκιμάσει κανείς πολλούς, τυχαίους συνελικτικούς πυρήνες [Jar+09; Yos+14].

Στην παραπάνω απλή αρχή βασίζεται το μοντέλο ROCKET [DPW20], με τις εξής όμως διαφορές σε σχέση με τα συνελικτικά δίκτυα:

- Το Rocket χρησιμοποιεί πολύ μεγάλο αριθμό πυρήνων. Καθώς υπάρχει μόνο ένα “στρώμα” από πυρήνες, και καθώς τα βάρη του πυρήνα δεν μαθαίνονται, το υπολογιστικό κόστος του υπολογισμού των συνελίξεων είναι χαμηλό και είναι δυνατό να χρησιμοποιηθεί ένας πολύ μεγάλος αριθμός πυρήνων.
- Επιπλέον, χρησιμοποιεί μια τεράστια ποικιλία πυρήνων. Σε αντίθεση με τα τυπικά συνελικτικά δίκτυα, όπου είναι σύνηθες τα στρώματα πυρήνων να μοιράζονται το ίδιο μέγεθος και padding, για το Rocket κάθε πυρήνας έχει τυχαίο μήκος, padding και φυσικά τυχαία βάρη και biases.
- Παράλληλα, χρησιμοποιεί με ξεχωριστό τρόπο το dilation. Στα τυπικά συνελικτικά νευρωνικά δίκτυα, το dilation αυξάνεται εκθετικά με βάθος, ενώ στο Rocket έχουμε τυχαίο dilation για κάθε πυρήνα. Έτσι μπορούμε να αναγνωρίσουμε μοτίβα σε διαφορετικές συχνότητες και κλίμακες, κάτι το οποίο είναι κρίσιμο για την απόδοση της μεθόδου.
- Εκτός από τη χρήση της μέγιστης τιμής των feature maps που προκύπτουν (global max pooling), η Rocket χρησιμοποιεί ένα πρόσθετο feature: το ποσοστό των θετικών τιμών (ή prvn). Αυτό επιτρέπει σε έναν ταξινομητή να σταθμίσει την επικράτηση ενός δεδομένου μοτίβου μέσα σε μία χρονοσειρά. Αυτό κρίνεται και από τους συγγραφείς ως το στοιχείο της αρχιτεκτονικής Rocket που είναι περισσότερο κρίσιμο για την εξαιρετική του ακρίβεια.

5.4 InceptionTime

Η αρχιτεκτονική αυτή εκπαιδεύεται με supervised τρόπο, δηλαδή υποθέτει ότι έχουμε εξ’ αρχής τις ετικέτες $Y = (y_1, \dots, y_N)^T$. Παρόλα αυτά χρησιμοποιήθηκε στην παρούσα εργασία, για τη σύγκριση των επιδόσεων supervised/self-supervised μεθόδων.

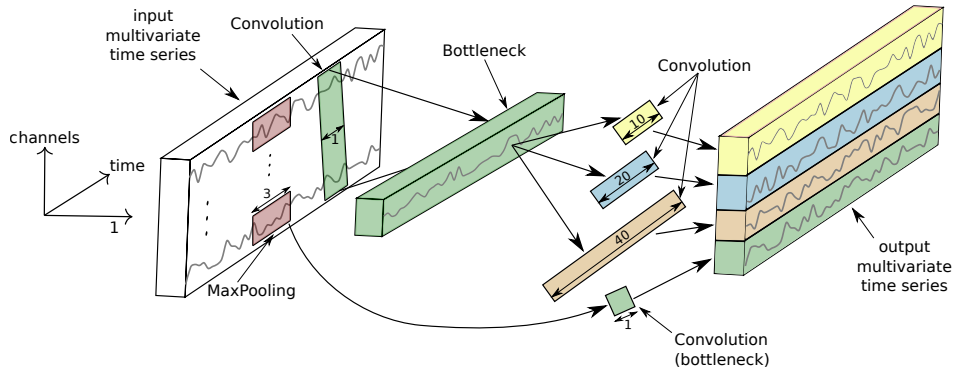
Η εφαρμογή των συνελικτικών δικτύων που είδαμε στην Παράγραφο 3.6 σε δεδομένα πολυμεταβλητών χρονοσειρών είναι απλή. Ένας μονοδιάστατος πυρήνας ολισθαίνει στην χρονοσειρά, επιτρέποντας έτσι στο δίκτυο να εξάγει χαρακτηριστικά που είναι χρονικά αμετάβλητα και χρήσιμα στην ταξινόμηση. Με τη διαδοχή πολλαπλών επιπέδων, το δίκτυο είναι σε θέση να εξαγάγει περαιτέρω ιεραρχικά χαρακτηριστικά που θεωρητικά θα πρέπει να βελτιώσουν την πρόβλεψη του δικτύου. Μόνη διαφορά είναι, αφού τα δεδομένα χρονοσειρών παρουσιάζουν μία διάσταση λιγότερη από αυτά των εικόνων είναι δυνατό να χρησιμοποιηθούν πιο πολύπλοκα μοντέλα που είναι συνήθως ανέφικτο να εκπαιδευτούν στα προβλήματα αναγνώρισης εικόνων. Για παράδειγμα μπορούμε να αφαιρέσουμε το στρώμα του pooling το οποίο απορρίπτει χρήσιμη πληροφορία με σκοπό τη μείωση της διάστασης του μοντέλου.

5.4.1 Inception modules

Αντί των τυπικών συνελικτικών στρωμάτων, το μοντέλο InceptionTime χρησιμοποιεί Inception modules, όπως αυτά φαίνονται στο Σχήμα 5.4.1. Ας δούμε πώς λειτουργεί αυτό, όταν έχουμε είσοδο μία πολυμεταβλητή χρονοσειρά M διαστάσεων. Το πρώτο σημαντικό στάδιο είναι το “bottleneck” στρώμα. Σε αυτό ολισθαίνουν m φίλτρα μήκους 1 και με stride μήκους 1 πάνω στη χρονοσειρά. Αυτό έχει σαν αποτέλεσμα η MTS να γίνει

διάστασης m όπου $m \ll M$. Έτσι μπορούμε να έχουμε μεγάλου μήκους φίλτρα κρατώντας μικρή τη διάσταση του μοντέλου, πράγμα που λύνει το overfitting στα μικρά datasets.

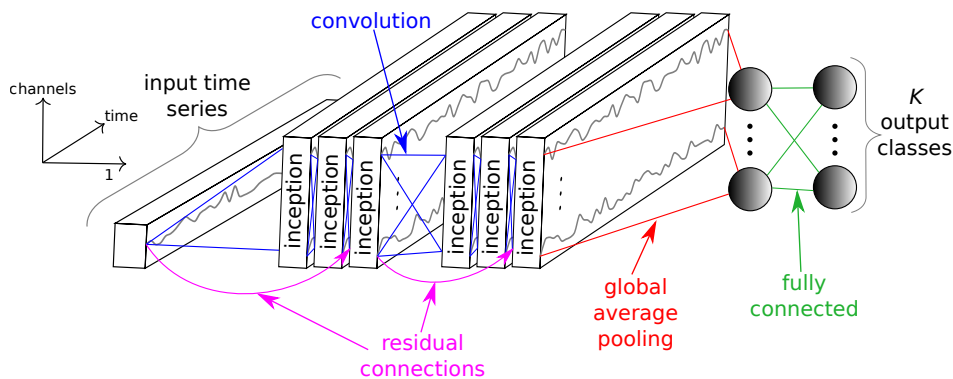
Στο δεύτερο στάδιο του Inception module εφαρμόζουμε φίλτρα διαφορετικού μήκους στην ίδια χρονοσειρά. Για παράδειγμα στο Σχήμα 5.4.1 τρία convolution blocks με μήκη $l \in \{10, 20, 40\}$ εφαρμόζονται στην έξοδο του “bottleneck” στρώματος. Επιπλέον, με σκοπό να κάνουμε το μοντέλο invariant σε μικρές μετατοπίσεις στην είσοδο, εφαρμόζουμε τον τελεστή MaxPooling, ακολουθούμενο από ένα ακόμη “bottleneck” στρώμα. Τα παραπάνω στάδια εκτελούνται σε κάθε Inception Module



Σχήμα 5.4.1: Inception module. Για απλότητα παρουσιάζεται από τους Fawaz et al. bottleneck layer μεγέθους $m = 1$.

5.4.2 Inception network

Το συνολικό δίκτυο φαίνεται στο Σχήμα 5.4.2 και αποτελείται από δύο residual blocks. Κάθε τέτοιο block έχει τρία Inception modules όπως τα περιγράψαμε πριν. Κάθε είσοδος του residual block μεταφέρεται με μία σύνδεση στο επόμενο block αποτρέποντας το πρόβλημα των vanishing gradients. Στην έξοδο του τελευταίου block εφαρμόζεται Global Average Pooling (GAP) και τέλος ένα fully-connected softmax στρώμα με αριθμό νευρώνων ίσο με τις κλάσεις του dataset.



Σχήμα 5.4.2: Ολόκληρο το Inception network.

5.4.3 InceptionTime: a neural network ensemble for TSC

Το τελικό InceptionTime είναι ένα ensemble που αποτελείται από 5 Inception networks, στα οποία δίνεται το ίδιο βάρος. Αυτό γιατί κατά την εκπαίδευση οι συγγραφείς παρατήρησαν μεγάλο standard deviation στο accuracy, κάτι που υποθέτουν προκύπτει από την τυχαία αρχικοποίηση των βαρών και την στοχαστικότητα που περιέχει ο ίδιος ο αλγόριθμος εκπαίδευσης. Οπότε με βάση την ακόλουθη εξίσωση γίνεται το ensembling 5 διαφορετικών δικτύων με διαφορετικές αρχικοποιήσεις:

$$\hat{y}_{i,c} = \frac{1}{n} \sum_{j=1}^n \sigma_c(x_i, \theta_j) \quad | \quad \forall c \in [1, C] \quad (5.4.1)$$

όπου $\hat{y}_{i,c}$ είναι η πιθανότητα που δίνει το ensemble στη χρονοσειρά x_i να ανήκει στην κλάση c , η οποία είναι ίση ο μέσος όρος των logistic outputs σ_c μεταξύ των n αρχικοποιημένων μοντέλων.

5.5 Self-supervised time series representation learning by inter-intra relational reasoning

Η παρούσα παράγραφος βασίζεται στο άρθρο των *H. Fan et al. [Hao21]*. Όλα τα σχήματα προέρχονται από εκεί, εκτός και αν αναφέρεται διαφορετικά.

Οι περισσότερες SSL μέθοδοι χρονοσειρών επικεντρώνονται κυρίως στην εξερεύνηση των σχέσεων μεταξύ δειγμάτων (inter-sample) ενώ λιγότερες προσπάθειες έχουν επικεντρωθεί στην ενδοχρονική (intra-temporal) σχέση, η οποία είναι σημαντική για δεδομένα χρονοσειρών. Συγκεκριμένα, το μοντέλο που χρησιμοποιείται στην παρούσα εργασία [Hao21], αξιοποιεί και των δύο ειδών σχέσεις. Μερικές βασικές έννοιες που είναι απαραίτητες για την κατανόηση του κειμένου, ορίζονται παρακάτω και οπτικοποιούνται στο Σχήμα 5.5.1:

Ορισμός 5.5.1: Ορισμοί μεταβλητών και εννοιών:

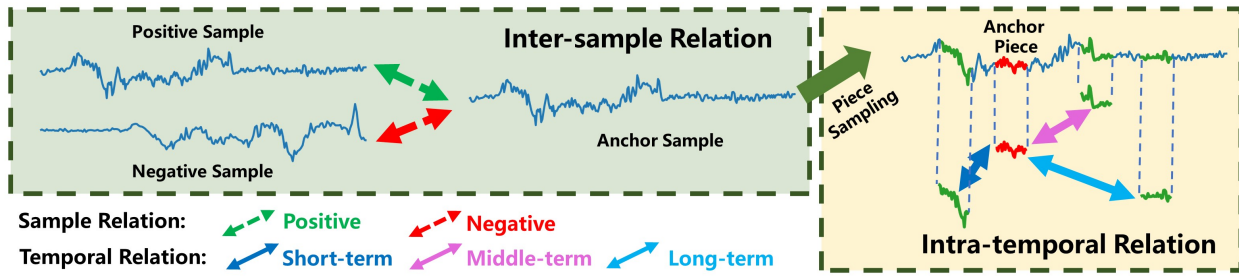
- **Σχέση μεταξύ δειγμάτων (inter-sample):** Μία δυαδική μεταβλητή με τιμές Positive και Negative. Για παράδειγμα, αν έχουμε samples EEG μετρήσεων από διαφορετικά άτομα, τότε η μεταβλητή παίρνει τιμές Positive και Negative για samples από το ίδιο και διαφορετικά άτομα αντίστοιχα.
- **Ενδοχρονική (intra-temporal) σχέση:** Είναι μια μεταβλητή της οποίας ο αριθμός των τιμών που μπορεί να πάρει (C) είναι υπερπαράμετρος του μοντέλου και αντιπροσωπεύει το πόσο μακριά βρίσκονται δύο κομμάτια της ίδιας χρονοσειράς. Για παράδειγμα, για $C = 3$, έχουμε 3 σχέσεις Short-term, Middle-term και Long-term όπως φαίνεται στο Σχήμα 5.5.1.
- **Δείγμα-άγκυρα (anchor):** Ορίζεται με διαφορετικό τρόπο για την inter σχέση και με διαφορετικό για την intra:
 - Για την inter-sample σχέση: έχουμε **anchor sample** μία ολόκληρη χρονοσειρά. Όταν αυτό συγκρίνεται με διαφορετική χρονοσειρά έχουμε αρνητική σχέση, ενώ όταν συγκρίνεται με τον εαυτό του (η μετασχηματισμούς του) έχουμε θετική σχέση.
 - Για την intra-temporal σχέση, έχουμε **anchor piece** ένα κομμάτι μήκους L της χρονοσειράς. Στη συνέχεια δειγματοληπτούμε κομμάτια της ίδιας χρονοσειράς και με βάση το πόσο μακριά βρίσκονται από το **anchor piece** προκύπτει η αντίστοιχη σχέση.
- **Θετική-αρνητική δειγματοληψία:** Ορίζεται για την inter-sample σχέση και δηλώνει τη διαδικασία κατά την οποία διαλέγουμε ως χρονοσειρά το ίδιο το **anchor sample** ή μετασχηματισμούς του (θετική δειγματοληψία) ή διαφορετικές χρονοσειρές (αρνητική δειγματοληψία).

Συνοψίζοντας, για τη σχέση μεταξύ δειγμάτων (inter-sample) γίνεται θετική και αρνητική δειγματοληψία (positive-negative sampling) με βάση ένα δείγμα-άγκυρα (anchor). Αντίστοιχα για την ενδοχρονική (intra-temporal) σχέση γίνεται δειγματοληψία χρονικών κομματιών μέσα στην ίδια χρονοσειρά.

Στη συνέχεια, μια κοινή ραχοκοκαλιά εξαγωγής χαρακτηριστικών (shared feature extraction backbone) χρησιμοποιείται σε συνδυασμό με δύο ξεχωριστές κεφαλές συλλογιστικής σχέσης οι οποίες εκπαιδεύονται: A) από τις σχέσεις μεταξύ των ζευγών δειγμάτων για συλλογιστική σχέση μεταξύ δειγμάτων, B) από τις σχέσεις των ζευγών κομματιών χρονοσειρών για τη συλλογιστική ενδοχρονικής σχέσης, αντίστοιχα.

Συγκεκριμένα, όπως φαίνεται στο Σχήμα 5.5.2, για σχέση μεταξύ δειγμάτων (inter-sample), δεδομένου ενός δείγματος άγκυρας, παράγουμε θετικά ζευγάρια μεταξύ της μετασχηματισμένης άγκυρας και διαφόρων μετασχηματισμών της. Όμοια, αρνητικά ζευγάρια μεταξύ της μετασχηματισμένης άγκυρας και άλλων μετασχηματισμένων μεμονωμένων δειγμάτων.

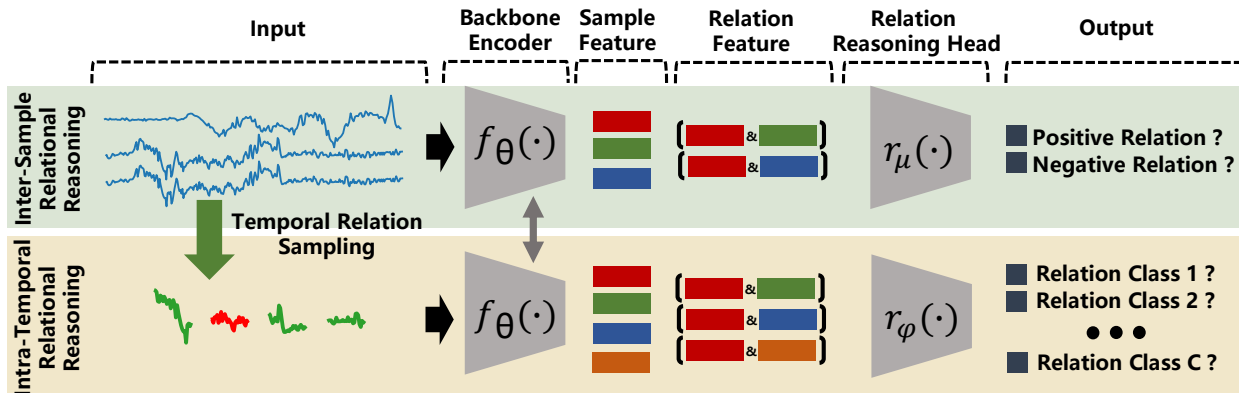
Για τον συλλογισμό ενδο-χρονικής (inter-temporal) σχέσης, δημιουργούμε πρώτα την άγκυρα. Στη συνέχεια, δειγματοληπτούνται διάφορα κομμάτια εντός του σήματος για την κατασκευή ζευγαριών μεταξύ της άγκυρας και του αντίστοιχου κομματιού. Τα ζευγάρια έχουν ως label τη χρονική απόσταση. Στο Σχήμα 5.5.1, φαίνεται παράδειγμα χρονικής σχέσης σε 3 κλίμακες: βραχυπρόθεσμη, μεσοπρόθεσμη και μακροπρόθεσμη καθαρά για λόγους απεικόνισης. Σε διαφορετικά σενάρια, θα μπορούσε να υπάρχει διαφορετικός αριθμός κλίμακας χρονικής σχέσης.



Σχήμα 5.5.1: **Αριστερά:** Οπτικοποίηση του inter-sample relation. Δίνεται ο anchor, ο μετασχηματισμός του (positive sample) και ο μετασχηματισμός ενός άλλου sample (negative sample) **Δεξιά:** Οπτικοποίηση του multi-scale intra-temporal relation. Εδώ έχουμε 3-scale temporal relations δηλαδή short-term, middle-term και long-term.

5.5.1 Μέθοδος

Δίνεται ένα σύνολο μη επισημειωμένων χρονοσειρών $\mathcal{T} = \{t_n\}_{n=1}^N$, όπου η κάθε χρονοσειρά $t_n = (t_{n,1}, \dots, t_{n,T})^T$ αποτελείται από T διατεταγμένες πραγματικές τιμές. Στόχος μας είναι να εξάγουμε μία χρήσιμη αναπαράσταση $z_n = f_\theta(t_n)$ από το πρώτο δίκτυο - κωδικοποιητή $f_\theta(\cdot)$, όπου θ είναι τα προς εκπαίδευση βάρη του νευρωνικού δικτύου. Η αρχιτεκτονική ολόκληρου του δικτύου φαίνεται στο Σχήμα 5.5.2, και αποτελείται από μια διακλάδωση που αφορά τη σχέση μεταξύ δειγμάτων (inter-sample) και άλλη μία για την ενδο-χρονική (intra-temporal) σχέση. Παίρνοντας την αρχική χρονοσειρά και τα δειγματοληπτημένα κομμάτια χρονοσειρών ως εισόδους, ο κοινός κωδικοποιητής $f_\theta(\cdot)$ εξάγει αναπαραστάσεις για την κάθε είσοδο. Στη συνέχεια αυτές συναθροίζονται και χρησιμοποιούνται ως εισόδοι στα δίκτυα $r_\mu(\cdot)$ και $r_\varphi(\cdot)$.



Σχήμα 5.5.2: Αρχιτεκτονική του SelfTime.

Inter-sample Relation Reasoning

Δοθέντων δύο διαφορετικών χρονοσειρών t_m και t_n από το \mathcal{T} , δημιουργούμε δύο σύνολα με K μετασχηματισμούς τους: $\mathcal{A}(t_m) = \{t_m^{(i)}\}_{i=1}^K$ και $\mathcal{A}(t_n) = \{t_n^{(i)}\}_{i=1}^K$, όπου $t_m^{(i)}$ και $t_n^{(i)}$ είναι ο i -ός μετασχηματισμός της t_m και t_n αντίστοιχα. Στη συνέχεια, κατασκευάζουμε θετικά και αρνητικά ζευγάρια. Ένα θετικό ζευγάρι $(t_m^{(i)}, t_m^{(j)})$ έχει δειγματοληπτηθεί από το ίδιο σύνολο μετασχηματισμών $\mathcal{A}(t_m)$, ενώ ένα αρνητικό ζευγάρι $(t_m^{(i)}, t_n^{(j)})$ από διαφορετικά σύνολα $\mathcal{A}(t_m)$ και $\mathcal{A}(t_n)$. Έπειτα εξάγουμε από τον αποκωδικοποιητή f_θ τα χαρακτηριστικά $z_m^{(i)} = f_\theta(t_m^{(i)})$, $z_m^{(j)} = f_\theta(t_m^{(j)})$, και $z_n^{(j)} = f_\theta(t_n^{(j)})$ και τα χρησιμοποιούμε για να δημιουργήσουμε θετικά ζευγάρια αναπαραστάσεων $[z_m^{(i)}, z_m^{(j)}]$, και αρνητικά ζευγάρια αναπαραστάσεων $[z_m^{(i)}, z_n^{(j)}]$, όπου $[\cdot, \cdot]$ συμβολίζει τη συνένωση διανυσμάτων. Το δίκτυο $r_\mu(\cdot)$ παίρνει το ζευγάρι αναπαραστάσεων ως είσοδο και υπολογίζει το τελικό σκορ συσχέτισης $h_{2m-1}^{(i,j)} = r_\mu([z_m^{(i)}, z_m^{(j)}])$ για θετική συσχέτιση $h_{2m}^{(i,j)} = r_\mu([z_m^{(i)}, z_n^{(j)}])$ για αρνητική συσχέτιση αντίστοιχα. Τέλος, το δίκτυο εκπαιδεύεται ως ένα δυαδικό πρόβλημα κατηγοριοποίησης ελαχιστοποιώντας το

cross-entropy σφάλμα \mathcal{L}_{inter} :

$$\mathcal{L}_{inter} = - \sum_{n=1}^{2N} \sum_{i=1}^K \sum_{j=1}^K (y_n^{(i,j)} \cdot \log(h_n^{(i,j)}) + (1 - y_n^{(i,j)}) \cdot \log(1 - h_n^{(i,j)})) \quad (5.5.1)$$

όπου $y_n^{(i,j)} = 1$ για θετική συσχέτιση και $y_n^{(i,j)} = 0$ για αρνητική συσχέτιση.

Intra-temporal Relation Reasoning

Για να εξερευνήσουμε τη δομή που κρύβεται κατά τη διάσταση του χρόνου, ζητούμε από το μοντέλο να προβλέψει διαφορετικούς τύπους χρονικής συσχέτισης μεταξύ κομματιών της χρονοσειράς. Συγκεκριμένα, δεδομένης μίας χρονοσειράς $\mathbf{t}_n = (t_{n,1}, \dots, t_{n,T})^T$, ορίζουμε ως L -μήκους χρονικό κομμάτι $\mathbf{p}_{n,u}$ της \mathbf{t}_n τη χρονοσειρά $\mathbf{p}_{n,u} = (t_{n,u}, t_{n,u+1}, \dots, t_{n,u+L-1})^T$ που ξεκινά από το χρόνο u και έχει μήκος L . Κατ' αρχάς κάνουμε τυχαία δειγματοληψία δύο L -μήκους κομματιών $\mathbf{p}_{n,u}$ και $\mathbf{p}_{n,v}$ της \mathbf{t}_n τα οποία ξεκινούν από τους χρόνους u και v αντίστοιχα. Στη συνέχεια η χρονική συσχέτιση μεταξύ των $\mathbf{p}_{n,u}$ και $\mathbf{p}_{n,v}$ καθορίζεται με βάση τη χρονική απόστασή τους $d_{u,v}$, π.χ. μια επιλογή για τη $d_{u,v}$ είναι η $d_{u,v} = |u - v|$ απόλυτη διαφορά των αρχικών χρόνων u και v . Έπειτα, κατατάσσουμε κάθε ζευγάρι κομματιών σε έναν από τους C τύπους συσχέτισης ως εξής: καθορίζουμε ένα χρονικό κατώφλι $D = \lfloor T/C \rfloor$, και αν η απόσταση $d_{u,v}$ μεταξύ των κομματιών του ζευγαριού D , ορίζουμε ως ετικέτα της σχέσης 0, αν $d_{u,v}$ είναι D μικρότερη από $2D$, ορίζουμε ως ετικέτα 1 και ούτω καθεξής μέχρι να φτάσουμε τους C τύπους χρονικής συσχέτισης. Παρακάτω φαίνεται ο ακριβής Αλγόριθμος 1 δειγματοληψίας και ορισμού ετικετών.

Algorithm 1 Temporal Relation Sampling.

Require:

- \mathbf{t}_n : A T -length time series.
- $\mathbf{p}_{n,u}, \mathbf{p}_{n,v}$: two L -length pieces of \mathbf{t}_n .
- C : Number of relation classes.

Ensure:

- $y_n^{(u,v)} \in \{1, 2, \dots, C\}$: The label of the temporal relation between $\mathbf{p}_{n,u}$ and $\mathbf{p}_{n,v}$.
 - 1: $d_{u,v} = |u - v|, D = \lfloor T/C \rfloor$
 - 2: **if** $d_{u,v} \leq D$ **then**
 - 3: $y_n^{(u,v)} = 0$
 - 4: **else if** $d_{u,v} \leq 2 * D$ **then**
 - 5: $y_n^{(u,v)} = 1$
 - 6: ...
 - 7: **else if** $d_{u,v} \leq (C - 1) * D$ **then**
 - 8: $y_n^{(u,v)} = C - 2$
 - 9: **else**
 - 10: $y_n^{(u,v)} = C - 1$
 - 11: **end if**
 - 12: **return** $y_n^{(u,v)}$
-

Τέλος χρησιμοποιούμε τα δειγματοληπτημένα ζευγάρια και τις ετικέτες που ορίσαμε γι' αυτά ως εξής. Χρησιμοποιούμε τον αρχικό κωδικοποιητή f_θ για να εξάγουμε χρήσιμες αναπαραστάσεις για κάθε κομμάτι χρονοσειράς, όπου $\mathbf{z}_{n,u} = f_\theta(\mathbf{p}_{n,u})$ και $\mathbf{z}_{n,v} = f_\theta(\mathbf{p}_{n,v})$. Συνενώνουμε τις αναπαραστάσεις για μία συνολική αναπαράσταση του ζευγαριού $[\mathbf{z}_{n,u}, \mathbf{z}_{n,v}]$. Στη συνέχεια, το δίκτυο $r_\varphi(\cdot)$ παίρνει τη συνολική αναπαράσταση ως είσοδο και παράγει την τελική πρόβλεψη $h_n^{(u,v)} = r_\varphi([\mathbf{z}_{n,u}, \mathbf{z}_{n,v}])$. Το δίκτυο εκπαιδεύεται ως multi-class πρόβλημα κατηγοριοποίησης ελαχιστοποιώντας το cross-entropy loss \mathcal{L}_{intra} :

$$\mathcal{L}_{intra} = - \sum_{n=1}^N y_n^{(u,v)} \cdot \log \frac{\exp(h_n^{(u,v)})}{\sum_{c=1}^C \exp(h_n^{(u,v)})} \quad (5.5.2)$$

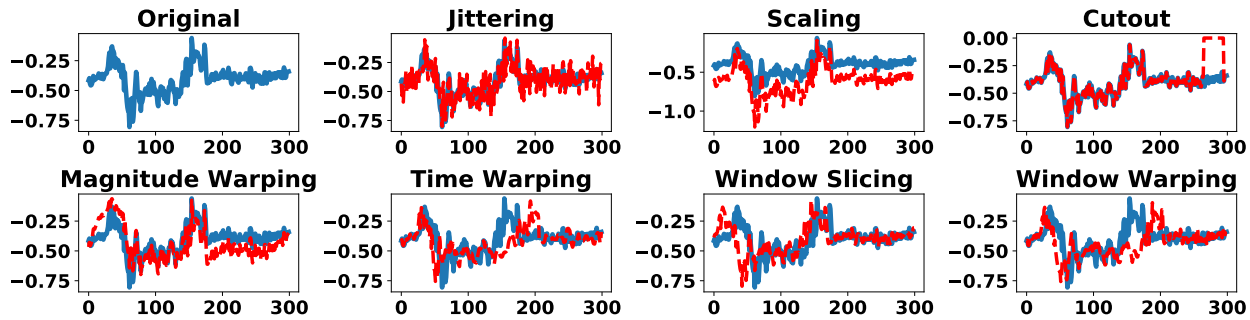
Τέλος, κάνουμε βελτιστοποίηση συνδυαστικά για το inter-sample σφάλμα (Εξ. 5.5.1) και το intra-temporal σφάλμα (Εξ. 5.5.2), οπότε το συνολικό loss είναι:

$$\mathcal{L} = \mathcal{L}_{inter} + \mathcal{L}_{intra} \quad (5.5.3)$$

Time Series Augmentation

Τα data augmentations στον τομέα των χρονοσειρών βασίζονται σε τυχαίους μετασχηματισμούς σε δύο πεδία [IU20]: στον πεδίο του πλάτους και στο πεδίο του χρόνου. Στον τομέα του πλάτους οι μετασχηματισμοί εφαρμόζονται στις τιμές της χρονοσειράς ενώ τα timestamps παραμένουν αμετάβλητα. Οι πιο συνηθισμένοι μετασχηματισμοί πλάτους είναι τα jittering, scaling, magnitude warping [Um+17], και cutout [DT17]. Στον τομέα

του χρόνου οι μετασχηματισμοί αλλάζουν τα timestamps της χρονοσειράς ενώ οι αντίστοιχες τιμές παραμένουν αμετάβλητες. Τέτοιοι μετασχηματισμοί είναι τα: time warping [Um+17], window slicing, and window warping [LMT16]. Μία οπτικοποίηση των παραπάνω φαίνεται στο Σχήμα 5.5.3 και εξηγούνται περισσότερο παρακάτω:



Σχήμα 5.5.3: Η μπλε γραμμή είναι το αρχικό σήμα και η κόκκινη το μετασχηματισμένο.

Jittering: Προσθέτουμε Γκαουσιανό θόρυβο στις τιμές του αρχικού σήματος με $\mathcal{N}(0, 0.2)$.

Scaling: Πολλαπλασιάζουμε όλες τις τιμές του αρχικού σήματος με μία τυχαία σταθερά από $\mathcal{N}(0, 0.4)$.

Cutout: Αντικαθιστούμε ένα τυχαίο 10% κομμάτι της αρχικής χρονοσειράς με μηδενικά, ενώ το υπόλοιπο σήμα μένει ίδιο.

Magnitude Warping: Το πλάτος της χρονοσειράς πολλαπλασιάζεται με μία cubic spline με 4 knots με τυχαία πλάτη $\mu = 1$ και $\sigma = 0.3$ [Um+17].

Time Warping: Κάνουμε scale το χρόνο με μία cubic spline curve με 8 knots, με τυχαία πλάτη $\mu = 1$ και $\sigma = 0.2$ [Um+17].

Window Slicing: Κόβουμε ένα παράθυρο με 80% της αρχικής χρονοσειράς και στη συνέχεια κάνουμε interpolate το cropped κομμάτι πίσω στην αρχική χρονοσειρά [LMT16].

Window Warping: Διαλέγουμε τυχαία ένα συνεχόμενο παράθυρο μήκους 30% του αρχικού σήματος και κάνουμε warp το χρόνο είτε με παράγοντα 0.5 είτε με 2 [LMT16].

5.6 A Transformer-based Framework for Multivariate Time Series Representation Learning

Η παράγραφος βασίζεται στο άρθρο των Zerveas, G. et al. [Zer+20]. Όλα τα σχήματα και οι πίνακες προέρχονται από εκεί, εκτός και αν αναφέρεται διαφορετικά.

Η παρακάτω μέθοδος διερευνά τη χρήση transformer encoder στο βήμα της προ εκπαίδευσης. Επίσης, μία διαφορά σε σχέση με τη μέθοδο της Παραγράφου 5.5 είναι ότι μπορεί να χρησιμοποιηθεί σε πολυμεταβλητές χρονοσειρές χωρίς πολλές τροποποιήσεις. Οι αναπαραστάσεις που προκύπτουν από το βήμα της προ-εκπαίδευσης μπορούν στη συνέχεια να χρησιμοποιηθούν σε προβλήματα παλινδρόμησης και ταξινόμησης. Οι transformers είναι μια σημαντική, πρόσφατα αναπτυγμένη κατηγορία μοντέλων βαθιάς μάθησης, που προτάθηκαν για πρώτη φορά για μετάφραση φυσικής γλώσσας [Vas+17] αλλά έχουν από τότε μονοπωλήσει σε όλες σχεδόν τις εργασίες NLP [Raf+20]. Ένας βασικός παράγοντας για την ευρεία επιτυχία των transformers στο NLP είναι η ικανότητά τους για εκμάθηση του τρόπου αναπαράστασης της φυσικής γλώσσας μέσω της προ-εκπαίδευσης χωρίς επίβλεψη [Bro+20; Raf+20].

Εμπνευσμένη από τα εντυπωσιακά αποτελέσματα που επιτυγχάνονται μέσω της προ εκπαίδευσης χωρίς επίβλεψη των μοντέλων transformers στο NLP η παρακάτω μέθοδος αναπτύσσει μια γενικά εφαρμόσιμη μεθοδολογία που μπορεί να αξιοποιήσει μη επισημειωμένα δεδομένα, εκπαιδύοντας πρώτα ένα μοντέλο transformer για εξαγωγή πυκνών διανυσματικών αναπαραστάσεων πολυμεταβλητών χρονοσειρών, μέσω αποθορυβοποίησης της εισόδου. Το προ-εκπαιδευμένο μοντέλο μπορεί στη συνέχεια να εφαρμοστεί σε διάφορες εργασίες, όπως παλινδρόμηση, ταξινόμηση, και πρόβλεψη. Σε πολλά ανοιχτά σύνολα δεδομένων η μέθοδος αυτή μπορεί να ξεπεράσει όλες τις τρέχουσες μοντελοποιήσεις αιχμής, ακόμη και όταν έχει πρόσβαση σε πολύ περιορισμένο αριθμό δειγμάτων δεδομένων εκπαίδευσης (τάξη των εκατοντάδων δειγμάτων), μια άνευ προηγουμένου επιτυχία για μοντέλα βαθιάς

μάθησης. Επίσης είναι σημαντικό ότι, παρά τις κοινές προκαταλήψεις για τους transformers από τον τομέα του NLP, όπου τα μοντέλα κορυφαίας απόδοσης έχουν δισεκατομμύρια παραμέτρους και απαιτούνται ημέρες έως εβδομάδες προ εκπαίδευσης σε πολλές παράλληλες GPU ή TPU, αποδεικνύεται ότι η μέθοδος αυτή, χρησιμοποιώντας το πολύ εκατοντάδες χιλιάδες παραμέτρους, μπορεί να εκπαιδευτεί πρακτικά ακόμη και σε CPU.

Η μέθοδος που παρουσιάζεται φιλοδοξεί να γενικεύσει τη χρήση transformers από λύσεις σε συγκεκριμένες εργασίες (που απαιτούν την πλήρη αρχιτεκτονική κωδικοποιητή-αποκωδικοποιητή) σε ένα πλαίσιο που ξεκινάει από προ-εκπαίδευση χωρίς επίβλεψη και στη συνέχεια με μικρές τροποποιήσεις να μπορεί να χρησιμοποιηθεί εύκολα για μια μεγάλη ποικιλία downstream εργασιών. Αυτό είναι ανάλογο με τον τρόπο που το BERT [Dev+19] μετέτρεψε ένα μοντέλο μετάφρασης σε ένα γενικό πλαίσιο που βασίζεται σε μάθηση χωρίς επίβλεψη, μια προσέγγιση που καθιέρωσε την κυριαρχία των transformers στο NLP.

5.6.1 Πλεονεκτήματα των transformers έναντι άλλων αρχιτεκτονικών

Τα μοντέλα transformers βασίζονται σε έναν μηχανισμό attention πολλαπλών κεφαλών που προσφέρει πολλά πλεονεκτήματα και τα καθιστά ιδιαίτερα κατάλληλα για δεδομένα χρονοσειρών:

- Μπορούν να λάβουν υπόψη συσχετίσεις μεταξύ σημείων που απέχουν πολύ χρονικά μεταξύ τους. Αυτό γιατί ο τρόπος που αναπαριστούν κάθε σημείο βασίζεται στο attention που έχει μάθει το μοντέλο ως προς τα υπόλοιπα σημεία της χρονοσειράς, όπου κι αν βρίσκονται αυτά. Μάλιστα αυτό το κάνουν χωρίς κάποιο position-dependent prior bias. Σε αντίθεση, τα RNN-based μοντέλα:
 - Μεταχειρίζονται τα σημεία στη μέση της ακολουθίας διαφορετικά από τα άκρα της (το ίδιο συμβαίνει ακόμη και στα bi-directional RNNs).
 - Ακόμη και τα καλοσχεδιασμένα LSTM (Long Short Term Memory) και GRU (Gated Recurrent Unit), έχουν κάποιο όριο στο πόσο μακριά βρίσκεται μία σχετική πληροφορία για να μπορέσουν να την αποθηκεύσουν στο hidden state τους (vanishing gradient problem [Hoc98; PMB13]). Άρα το context που χρησιμοποιούν για να αναπαραστήσουν κάθε σημείο της ακολουθίας είναι περιορισμένο/
- Τα διαφορετικά attention heads μπορούν να μάθουν διαφορετικούς υποχώρους και άρα διαφορετικούς τρόπους όπου σχετίζονται τα σημεία μεταξύ τους. Για παράδειγμα, έστω ένα σήμα με δύο κύριες συχνότητες: $1/T_1$ και $1/T_2$. Τότε μία κεφαλή μπορεί να κοιτάζει τα γειτονικά σημεία, μία άλλη σημεία που απέχουν T_1 ενώ μία τρίτη σημεία που απέχουν T_2 κλπ. Αυτό δεν μπορεί να συμβεί ούτε στα κλασσικά RNN ούτε στα RNN's με attention στα οποία μαθαίνεται μία attention distribution.
- Μετά από κάθε layer του encoder, μαθαίνουμε όλο και πιο abstract κατανομές attention, αφού έχουμε και πιο abstract αναπαραστάσεις της εισόδου. Σε αντίθεση τα RNN's με attention χρησιμοποιούν συνήθως ως είσοδο τα hidden states του RNN και άρα μαθαίνουν μία μόνο κατανομή attention επάνω σε αυτά.

5.6.2 Μεθοδολογία

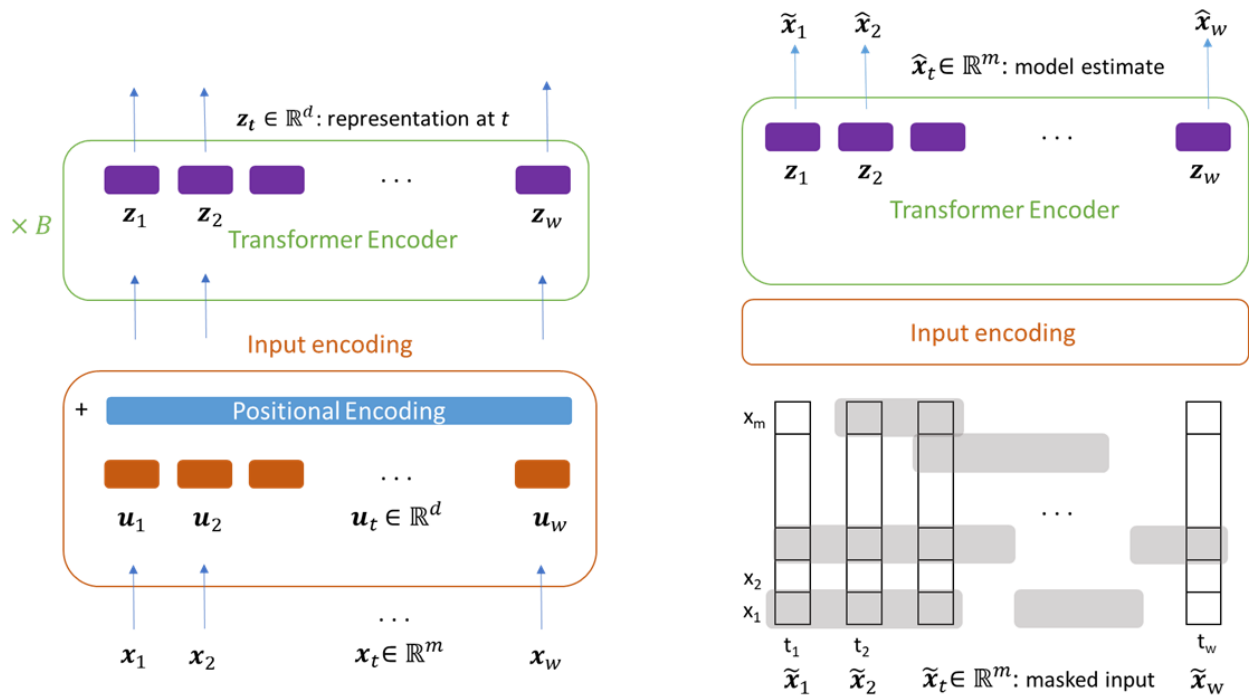
Βασικό μοντέλο

Η μέθοδος βασίζεται στο encoder κομμάτι ενός transformer, όπως αυτός προτάθηκε στο [Vas+17], χωρίς, ωστόσο, να γίνεται χρήση του decoder μέρους της αρχιτεκτονικής. Το διάγραμμα της αρχιτεκτονικής φαίνεται στο Σχήμα 5.6.1. Παρακάτω αναφέρονται οι τροποποιήσεις του πρωτότυπου transformer μοντέλου έτσι ώστε να είναι συμβατό με δεδομένα multivariate χρονοσειρών, αντί για ακολουθίες διακριτών λέξεων.

Κάθε training sample $\mathbf{X} \in \mathbb{R}^{w \times m}$, είναι μια χρονοσειρά m μεταβλητών μήκους w . Έτσι έχουμε μία ακολουθία w feature vectors $\mathbf{x}_t \in \mathbb{R}^m$: $\mathbf{X} \in \mathbb{R}^{w \times m} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_w]$. Κάθε \mathbf{x}_t γίνεται normalize ως προς κάθε μεταβλητή και στη συνέχεια προβάλλονται σε ένα d -διάστατο χώρο, όπου d είναι η διάσταση και των τελικών διαστάσεων αναπαράστασης του transformer (συνήθως λέγεται *model dimension*):

$$\mathbf{u}_t = \mathbf{W}_p \mathbf{x}_t + \mathbf{b}_p \quad (5.6.1)$$

όπου $\mathbf{W}_p \in \mathbb{R}^{d \times m}$, $\mathbf{b}_p \in \mathbb{R}^d$ είναι παράμετροι προς εκπαίδευση και $\mathbf{u}_t \in \mathbb{R}^d, t = 0, \dots, w$ είναι τα τελικά διανύσματα εισόδου (τα αντίστοιχα word vectors ενός NLP transformer). Στο τελευταίο προστίθενται τα positional encodings και αυτή αποτελεί την τελική αναπαράσταση που θα μπει στο πρώτο self-attention στρώμα



Σχήμα 5.6.1: **Αριστερά:** Το διάνυσμα χαρακτηριστικών x_t σε κάθε χρονική στιγμή t προβάλλεται σε διάνυσμα u_t ίδιας διάστασης d . Στο τελευταίο προστίθενται τα positional encodings και αυτή αποτελεί την τελική αναπαράσταση στο πρώτο self-attention στρώμα για να μετατραπεί σε keys, queries και values. **Δεξιά:** Training setup of the unsupervised pre-training task. Κάνουμε mask ένα τμήμα r κάθε μεταβλητής της χρονοσειράς ανεξάρτητα, έτσι ώστε να έχουμε τμήματα μέσου μήκους l_m masked, ακολουθούμενα από unmasked κομμάτια μέσου μήκους $l_u = \frac{1-r}{r}l_m$. Στη συνέχεια χρησιμοποιούμε ένα linear layer με είσοδο τις διανυσματικές αναπαραστάσεις z_t , σε κάθε στιγμή και ως labels τα uncorrupted input vectors x_t . Τέλος υπολογίζεται το Μέσο Τετραγωνικό Σφάλμα μετρώντας μόνο τις masked τιμές.

Dataset	Task (Metric)	LayerNorm	BatchNorm
Heartbeat	Classif. (Accuracy)	0.741	0.776
InsectWingbeat	Classif. (Accuracy)	0.658	0.684
SpokenArabicDigits	Classif. (Accuracy)	0.993	0.993
PEMS-SF	Classif. (Accuracy)	0.832	0.919
BenzeneConcentration	Regress. (RMSE)	2.053	0.516
BeijingPM25Quality	Regress. (RMSE)	61.082	60.357
LiveFuelMoistureContent	Regress. (RMSE)	42.993	42.607

Πίνακας 5.1: Σύγκριση layer normalization και batch normalization (Batch size=128).

για να μετατραπεί σε keys, queries και values μετά τον πολλαπλασιασμό με τους αντίστοιχους πίνακες. Είναι σημαντικό να ειπωθεί ότι τα input vectors \mathbf{u}_t δεν είναι απαραίτητο να βγουν μέσω της (5.6.1) σε περίπτωση όπου το temporal resolution είναι πολύ λεπτομερές. Αυτό γιατί το υπολογιστικό κόστος είναι της τάξης $O(w^2)$ ενώ ο αριθμός των παραμέτρων ¹ $O(w)$, όπου w το μήκος της ακολουθίας. Ένας άλλος τρόπος υπολογισμού του \mathbf{u}_t στην περίπτωση που έχουμε υψηλό temporal resolution είναι να χρησιμοποιήσουμε 1D-convolutional layer με 1 input και d output κανάλια και kernels K_i μεγέθους (k, m) , όπου k το πλάτος σε αριθμό time steps and i το κανάλι εξόδου:

$$u_t^i = u(t, i) = \sum_j \sum_h x(t + j, h) K_i(j, h), \quad i = 1, \dots, d \quad (5.6.2)$$

Με αυτό τον τρόπο μπορούμε να αλλάξουμε το temporal resolution χρησιμοποιώντας stride ή dilation factor μεγαλύτερο του 1. Επιπλέον αν και στο paper χρησιμοποιείται μόνο η Εξίσωση (5.6.1), μπορεί εναλλακτικά να χρησιμοποιηθεί η (5.6.2) για τον υπολογισμό των keys και queries και η (5.6.1) για τον υπολογισμό των values στο self-attention layer. Αυτό είναι χρήσιμο στις univariate χρονοσειρές, όπου το self-attention θα θεωρούσε ως match όλα τα time steps όπου έχουν κοινές τιμές με την ανεξάρτητη μεταβλητή, όπως περιγράφεται στο [Li+20].

Τέλος επειδή οι transformers είναι μία feed-forward αρχιτεκτονική που αγνοεί τη σειρά των εισόδων, χρειάζεται να προσθέσουμε positional encodings ως επιπλέον πληροφορία που κωδικοποιεί την διαδοχική φύση των δεδομένων χρονοσειρών. Άρα προσθέτουμε στα διανύσματα εισόδου $U \in \mathbb{R}^{w \times d} = [\mathbf{u}_1, \dots, \mathbf{u}_w]$ τα positional encodings $W_{\text{pos}} \in \mathbb{R}^{w \times d}$. Τελικά: $U' = U + W_{\text{pos}}$.

Αντί όμως για τα ντετερμινιστικά sinusoidal encodings, που προτείνονται στο [Vas+17], χρησιμοποιήθηκαν εκπαιδευόμενα positional embeddings καθώς αυτά έδιναν καλύτερα αποτελέσματα σε όλα τα datasets που δοκιμάστηκαν. Επίσης, αν και τα positional encodings προστίθενται στο τελικό προβαλλόμενο διάνυσμα φαίνεται να μην παρεμβαίνουν σημαντικά στις αριθμητικές πληροφορίες του. Μία υπόθεση είναι ότι αυτό συμβαίνει επειδή τα embeddings μαθαίνονται με τέτοιο τρόπο, ώστε να καταλαμβάνουν έναν διαφορετικό, περίπου ορθογώνιο, υπόχωρο από αυτόν στον οποίο βρίσκονται τα προβαλλόμενα δείγματα χρονοσειρών.

Τα διάφορα samples χρονοσειρών δεν έχουν πάντοτε το ίδιο μήκος μεταξύ τους. Αυτό μπορεί να αντιμετωπιστεί με τον εξής τρόπο: Θέτουμε ένα μέγιστο μήκος ακολουθίας w και οι μικρότερες ακολουθίες καλύπτονται με τυχαίες τιμές. Στη συνέχεια δημιουργούμε μία padding mask η οποία προσθέτει έναν μεγάλο αρνητικό αριθμό στα attention scores των padded θέσεων. Άρα κατά τον υπολογισμό της self-attention distribution η συνάρτηση softmax θα αγνοήσει τελείως τις padded τιμές.

Οι transformers στο NLP χρησιμοποιούν layer normalization αμέσως μετά το self-attention μέρος και αμέσως μετά το feed-forward μέρος, αντί για batch normalization και αυτό οδηγεί σε καλύτερα αποτελέσματα [Vas+17]. Παρ' όλα αυτά στα δεδομένα χρονοσειρών έχουμε συχνά outlier τιμές κάτι το οποίο δεν υπάρχει στα word embeddings του NLP και αυτό κάνει τη χρήση batch normalization προτιμότερη. Επίσης σύμφωνα με το [She+20], η κατώτερες επιδόσεις του batch normalization στο NLP αποδίδονται κυρίως στις μεγάλες διακυμάνσεις στο sample length (τα samples είναι προτάσεις), ενώ στα dataset χρονοσειρών το sample variation είναι συνήθως μικρότερο. Στον Πίνακα 5.1 φαίνεται πειραματικά ότι το batch normalization λειτουργεί καλύτερα σε δεδομένα χρονοσειρών.

¹ Δηλαδή των παραμέτρων του learnable positional encoding, batch normalization και των output layers

Pretext task

Ως task για το unsupervised pre-training του μοντέλου ορίζεται το autoregressive task της αποθρομβοποίησης της εισόδου. Συγκεκριμένα, θέτουμε ένα μέρος της εισόδου ίσο με 0 και ζητούμε από το μοντέλο να προβλέψει τις masked τιμές. Η παραπάνω διαδικασία φαίνεται και στο Σχήμα 5.6.1. Μία δυαδική μάσκα θορύβου $\mathbf{M} \in \mathbb{R}^{w \times m}$, δημιουργείται ανεξάρτητα για κάθε training sample, και πολλαπλασιάζεται στοιχείο-προς-στοιχείο με την είσοδο: $\tilde{\mathbf{X}} = \mathbf{M} \odot \mathbf{X}$. Κατά μέσο όρο ένα ποσοστό r κάθε στήλης μήκους w της μάσκας (η οποία αποτελεί μια από τις μεταβλητές της χρονοσειράς) τίθεται σε 0. Οι πιθανότητες μετάβασης κατάστασης είναι τέτοιες, ώστε κάθε τμήμα μηδενικών να έχει μήκος που ακολουθεί γεωμετρική κατανομή με μέσο μήκος l_m και να ακολουθείται από ένα τμήμα άσων με μέσο μήκος $l_u = \frac{1-r}{r}l_m$. Το l_m επιλέγεται ίσο με: $l_m = 3$. Ο λόγος που γίνεται αυτό, αντί να θέτουμε τα στοιχεία της μάσκας ανεξάρτητα σύμφωνα με μία κατανομή Bernoulli με παράμετρο r είναι για να ελέγχουμε το μήκος της masked ακολουθίας. Αυτό γιατί πολύ μικρές masked ακολουθίες (π.χ. μήκους 1) είναι πολύ εύκολο να προβλεφθούν από το μοντέλο π.χ. παίρνοντας το μέσο όρο των γειτονικών σημείων. Δεδομένης της κατανομής Bernoulli η αποφυγή τέτοιων ακολουθιών θα γινόταν μόνο μεγαλώνοντας το r κάτι το οποίο θα έκανε το πρόβλημα αδύνατο. Με αυτό τον τρόπο, σε κάθε time step θα έχουμε $r \cdot m$ μεταβλητές masked. Εμπειρικά επιλέγεται $r = 0.15$ για όλα τα πειράματα. Αυτό το σχήμα κατασκευής μάσκας είναι διαφορετικό από εκείνο που χρησιμοποιείται π.χ. στο BERT [Dev+19], όπου ένα ειδικό token αντικαθιστά ολόκληρο το feature vector στα masked τμήματα. Το προτεινόμενο σχήμα κατασκευής μάσκας κάνει το μοντέλο να μαθαίνει να προβλέπει ένα masked σημείο μίας μεταβλητής κοιτάζοντας το παρελθόν και το μέλλον της ίδιας μεταβλητής αλλά και κοιτώντας τις εξαρτήσεις με τις υπόλοιπες μεταβλητές τη συγκεκριμένη στιγμή. Το συγκεκριμένο σχήμα έχει αποδειχθεί το αποτελεσματικότερο μεταξύ των άλλων που συζητούνται. Επίσης στο Σχήμα 5.6.2 φαίνεται η οπτικοποίηση της μάσκας και πώς αυτές αλλάζουν με τις αλλαγές των παραμέτρων.

Τέλος χρησιμοποιώντας ένα linear layer με παραμέτρους $\mathbf{W}_o \in \mathbb{R}^{m \times d}$, $\mathbf{b}_o \in \mathbb{R}^m$ με είσοδο τις διανυσματικές αναπαραστάσεις $\mathbf{z}_t \in \mathbb{R}^d$, για κάθε στιγμή προβλέπουμε μία εκτίμηση $\hat{\mathbf{x}}_t$ των unmasked input vectors \mathbf{x}_t . Παρ' όλα αυτά μόνο οι προβλέψεις των masked τιμών (με δείκτες που ανήκουν στο σύνολο $M \equiv \{(t, i) : m_{t,i} = 0\}$, όπου $m_{t,i}$ είναι στοιχεία της μάσκας \mathbf{M}), προσμετρώνται στον υπολογισμό του Ελάχιστου Τετραγωνικού Σφάλματος:

$$\hat{\mathbf{x}}_t = \mathbf{W}_o \mathbf{z}_t + \mathbf{b}_o \quad (5.6.3)$$

$$\mathcal{L}_{\text{MSE}} = \frac{1}{|M|} \sum_{(t,i) \in M} (\hat{x}(t,i) - x(t,i))^2 \quad (5.6.4)$$

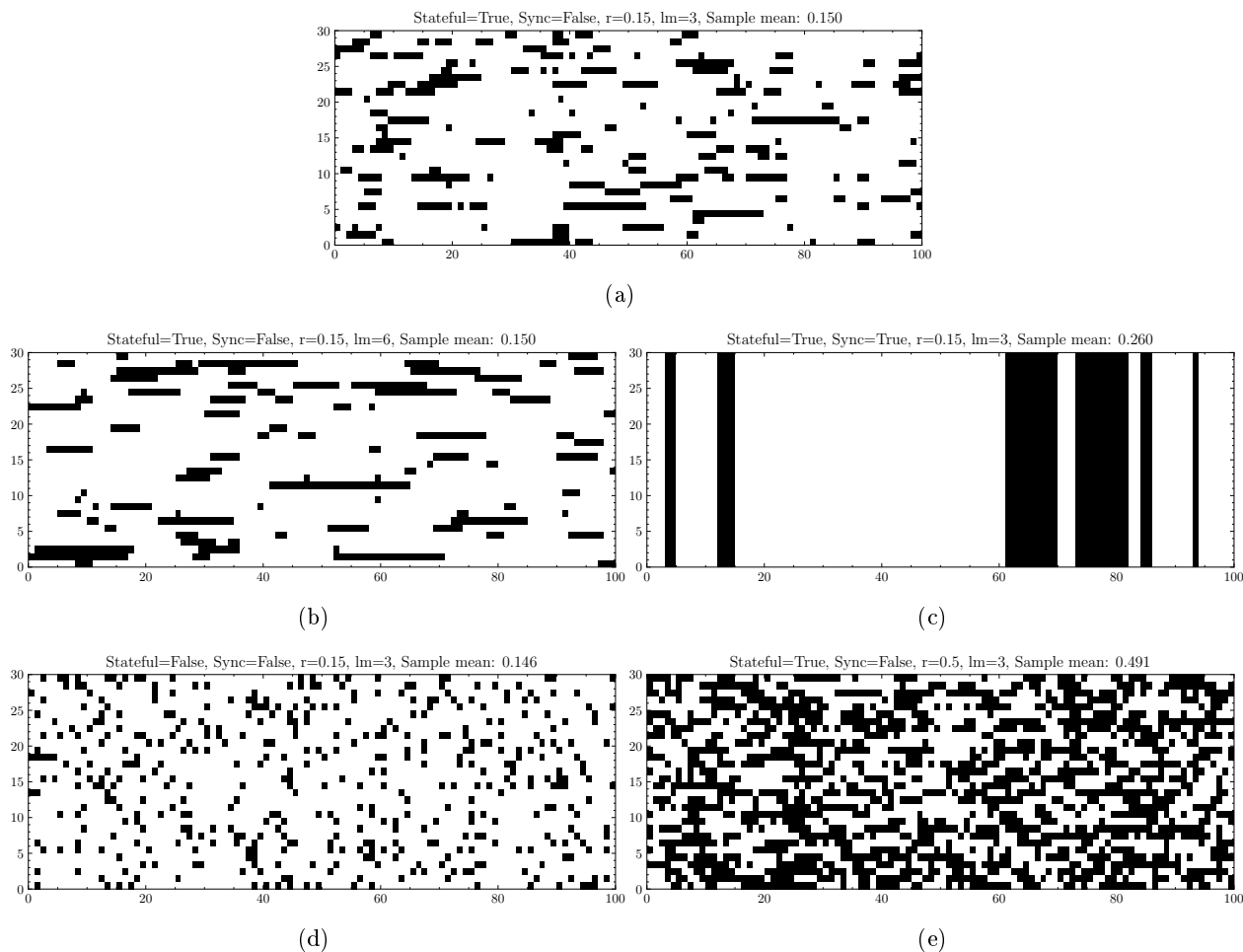
Αυτή η προσέγγιση διαφέρει από εκείνη που χρησιμοποιείται στους denoising autoencoders, όπου το σφάλμα υπολογίζεται σε όλη την είσοδο και χρησιμοποιείται Gaussian θόρυβος. Επίσης διαφέρει από τη χρήση dropout στα input embeddings και από την πλευρά του mask distribution, αλλά και από το γεγονός ότι η μάσκα καθορίζει και τη συνάρτηση σφάλματος (loss function). Στην πραγματικότητα χρησιμοποιείται dropout 10% κατά την εκπαίδευση.

Επίσης αξίζει να σημειωθεί ότι το pretext-task που παρουσιάστηκε είναι το ίδιο με το να λύναμε το πρόβλημα εποπτευόμενης μάθησης, στο οποίο κάνουμε imputate τις απύσες τιμές. Στην πραγματικότητα το μοντέλο φτάνει Root Mean Square Error πολύ κοντά στο 0 στα test sets ανοιχτών dataset για imputation όταν έχει γίνει pretrain με την παραπάνω τεχνική. Μια οπτικοποίηση επάνω στο test set του BenzeneConcentration φαίνεται στο σχήμα 5.6.3.

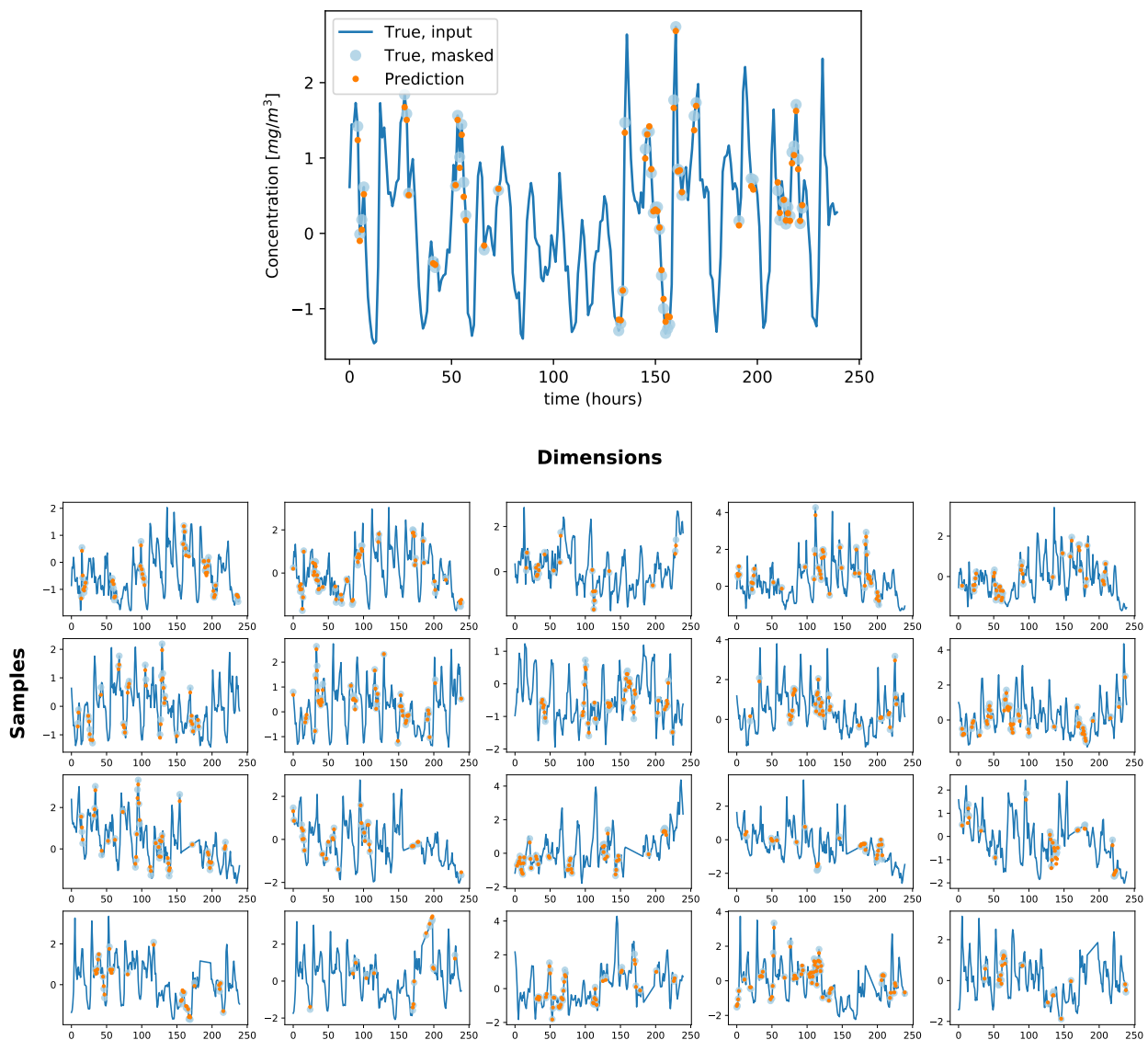
Τέλος, χρησιμοποιώντας διαφορετικά patterns στις μάσκες, μπορεί κάποιος να πετύχει καλύτερα αποτελέσματα σε διαφορετικά downstream tasks κρατώντας ίδιο το υπόλοιπο μοντέλο. Π.χ. αν κάποιος κάνει mask τα τελευταία time-steps της χρονοσειράς, σύγχρονα για όλες τις μεταβλητές τότε θα βγουν αναπαραστάσεις που θα λειτουργήσουν καλύτερα στο forecasting χρονοσειρών. Αν επιπλέον η χρονοσειρά έχει αρκετά μεγάλο μήκος, το ίδιο σχήμα μπορεί να γίνει και με sliding window όπως παρουσιάζεται στο Σχήμα 5.6.4.

Downstream tasks

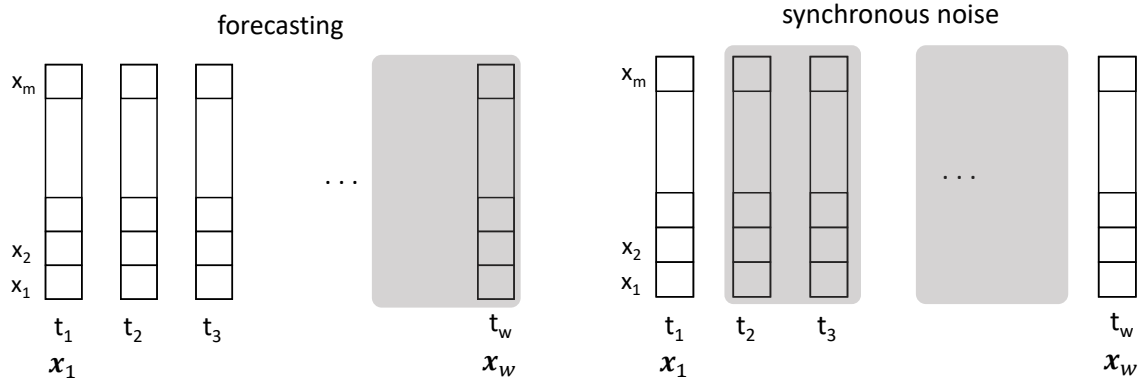
Το μοντέλο όπως περιγράφεται στην 5.6.2 και φαίνεται στο Σχήμα 5.6.1 μπορεί να χρησιμοποιηθεί για σκοπούς regression και classification με τις εξής αλλαγές: τα τελικά διανύσματα αναπαραστάσεως $\mathbf{z}_t \in \mathbb{R}^d$ που αντιστοιχούν σε διαδοχικά time steps γίνονται concatenate σε ένα διάνυσμα $\bar{\mathbf{z}} \in \mathbb{R}^{d \cdot w} = [\mathbf{z}_1; \dots; \mathbf{z}_w]$, το οποίο αποτελεί και την



Σχήμα 5.6.2: Οπτικοποίηση διαφόρων σχημάτων κατασκευής μάσκας. Στην κορυφή (5.6.2a) φαίνεται η μάσκα με τις default τιμές $r = 0.15$, $lm = 3$, $stateful=True$, $sync=False$. Στη συνέχεια σε κάθε σχήμα αλλάζουμε μία από τις μεταβλητές και κρατάμε τις υπόλοιπες στις default για να δούμε τις διαφορές. Μεγαλώνοντας το lm (Σχήμα 5.6.2b) βλέπουμε ότι μεγαλώνει το μέσο μήκος που γίνεται masked δηλαδή τα μαύρα κομμάτια. Στη συνέχεια θέτοντας το $sync=True$ (Σχήμα 5.6.2c) βλέπουμε ότι γίνονται masked όλες οι μεταβλητές της χρονοσειράς σύγχρονα. Στη συνέχεια αλλάζοντας το $Stateful$ σε $True$ (Σχήμα 5.6.2d) βλέπουμε ότι πλέον έχουμε κατανομή bernoulli και άρα δεν πετυχαίνουμε τόσο συχνά συνεχόμενες ακολουθίες στη μάσκα για δεδομένο r . Τέλος μεγαλώνοντας το r σε 0.5 βλέπουμε ότι αλλάζει το γενικό κλάσμα masked σημείων σε όλη τη χρονοσειρά και όλες τις μεταβλητές χωρίς να αλλάζει η κατανομή του μήκους. Επίσης στον τίτλο του κάθε σχήματος φαίνεται το sample mean masked σημείων για κάθε δείγμα μάσκας. Τα σχήματα δημιουργήθηκαν χρησιμοποιώντας τη βιβλιοθήκη [Ogu20].



Σχήμα 5.6.3: **Πάνω:** Imputation of missing values στο test set του BenzeneConcentration dataset. Η συνεχόμενη μπλε γραμμή είναι το ground truth σήμα, οι ανοιχτοί μπλε κύκλοι είναι οι τιμές που έγιναν mask και με πορτοκαλί οι προβλέψεις του συστήματος. Βλέπουμε ότι το μοντέλο πηγαίνει πολύ καλά ακόμη και σε απότομες αλλαγές ή όταν λείπουν πολλές συνεχόμενες τιμές. **Κάτω:** Το ίδιο για 5 διαφορετικές μεταβλητές της χρονοσειράς.



Σχήμα 5.6.4: Σχήμα μάσκας για forecasting (αριστερά), sliding window (δεξιά).

Dataset	Task (Metric)	Static		Fine-tuned	
		Metric	Epoch time (s)	Metric	Epoch time (s)
Heartbeat	Classif. (Accuracy)	0.756	0.082	0.776	0.14
InsectWingbeat	Classif. (Accuracy)	0.236	4.52	0.687	6.21
SpokenArabicDigits	Classif. (Accuracy)	0.996	1.29	0.998	2.00
PEMS-SF	Classif. (Accuracy)	0.844	0.208	0.896	0.281
BenzeneConcentration	Regress. (RMSE)	4.684	0.697	0.494	1.101
BeijingPM25Quality	Regress. (RMSE)	65.608	1.91	53.492	2.68
LiveFuelMoistureContent	Regress. (RMSE)	48.724	1.696	43.138	3.57

Πίνακας 5.2: Σύγκριση επιδόσεων μεταξύ fine-tuned και frozen (όλων των layers εκτός του τελευταίου). Οι χρόνοι αφορούν: per-epoch training time σε GPU.

είσοδο σε ένα linear output layer παραμέτρων: $\mathbf{W}_o \in \mathbb{R}^{n \times (d \cdot w)}$, $\mathbf{b}_o \in \mathbb{R}^n$, όπου n ο αριθμός των κλιμακωτών που προσεγγίζονται στο regression πρόβλημα (συνήθως $n = 1$), ή ο αριθμός κλάσεων στο classification πρόβλημα:

$$\hat{\mathbf{y}} = \mathbf{W}_o \bar{\mathbf{z}} + \mathbf{b}_o \quad (5.6.5)$$

Στην περίπτωση του regression, το σφάλμα για ένα data sample είναι το τετραγωνικό σφάλμα $\mathcal{L} = \|\hat{\mathbf{y}} - \mathbf{y}\|^2$, όπου $\mathbf{y} \in \mathbb{R}^n$ οι πραγματικές τιμές (ground truth). Στην περίπτωση του classification, οι προβλέψεις $\hat{\mathbf{y}}$ περνούν από μια συνάρτηση softmax για να πάρουμε μία κατανομή επάνω στις κλάσεις, και στη συνέχεια το sample loss προκύπτει από τον υπολογισμό του cross-entropy της κατανομής αυτής με τις πραγματικές τιμές.

Μία επιλογή είναι το fine-tuning των pre-trained μοντέλων, δηλαδή να αφήσουμε όλα τα βάρη να εκπαιδευτούν περαιτέρω. Διαφορετικά μπορούμε να παγώσουμε όλα τα layers εκτός από το τελευταίο output layer. Σε αυτή την περίπτωση χρησιμοποιούμε τα προεκπαιδευμένα representations του μοντέλου τα εξετάζουμε ως προς το separability τους. Στον Πίνακα 5.2 φαίνονται τα trade-offs με βάση τις επιδόσεις σε θέμα ταχύτητας και ποιότητας των αποτελεσμάτων μεταξύ των δύο μεθόδων.

5.6.3 Πειράματα & Αποτελέσματα των Zerveas et al.

Ερώτημα 1: Αν έχουμε ένα μερικώς επισημειωμένο σύνολο δεδομένων συγκεκριμένου μεγέθους, πώς θα επηρεάσουν οι πρόσθετες ετικέτες την απόδοση; Αυτή είναι μια από τις πιο σημαντικές αποφάσεις που αντιμετωπίζουν όσοι ασχολούνται με την δημιουργία συνόλων δεδομένων, δηλαδή σε ποιο βαθμό θα βοηθήσει η περαιτέρω επισημείωση δεδομένων. Για να εξετάσουν αυτό το ερώτημα οι Zerveas et al., επιλέγουν το μεγαλύτερο σύνολο δεδομένων (12,5 χιλιάδες δείγματα), προκειμένου να αποφύγουν τη διακύμανση που εισάγεται από μικρά μεγέθη συνόλων. Η αριστερή εικόνα του Σχήματος 5.6.5 (όπου κάθε κουκκίδα είναι ένα πείραμα) δείχνει πώς αλλάζει η απόδοση στο test set, καθώς αυξάνει το ποσοστό δεδομένων στα οποία γίνεται supervised training. Όπως αναμενόταν, με ένα αυξανόμενο ποσοστό διαθέσιμων ετικετών, η απόδοση

Dataset	Root MSE											Zerveas, G. et al.	
	SVR	Random Forest	XGBoost	1-NN-ED	5-NN-ED	1-NN-DTWD	5-NN-DTWD	Rocket	FCN	ResNet	Inception	TST (sup. only)	TST (pretrained)
AppliancesEnergy	3.457	3.455	3.489	5.231	4.227	6.036	4.019	2.299	2.865	3.065	4.435	2.228	2.375
BenzeneConcentr.	4.790	0.855	0.637	6.535	5.844	4.983	4.868	3.360	4.988	4.061	1.584	0.517	0.494
BeijingPM10	110.574	94.072	93.138	139.229	115.669	139.134	115.502	120.057	94.348	95.489	96.749	91.344	86.866
BeijingPM25	75.734	63.301	59.495	88.193	74.156	88.256	72.717	62.769	59.726	64.462	62.227	60.357	53.492
LiveFuelMoisture	43.021	44.657	44.295	58.238	46.331	57.111	46.290	41.829	47.877	51.632	51.539	42.607	43.138
IEEPPPG	36.301	32.109	31.487	33.208	27.111	37.140	33.572	36.515	34.325	33.150	23.903	25.042	27.806
Avg Rel. diff. from mean	0.097	-0.172	-0.197	0.377	0.152	0.353	0.124	-0.048	0.021	0.005	-0.108	-0.301	-0.303
Avg Rank	7.166	4.5	3.5	10.833	8	11.167	7.667	5.667	6.167	6.333	5.666	1.333	

Πίνακας 5.3: Απόδοση σε **multivariate regression**, μετρική: Root Mean Squared Error. Bold σημαίνει οι καλύτερες τιμές και υπογραμμισμένες σημαίνει δεύτερες καλύτερες.

βελτιώνεται τόσο για το fully supervised μοντέλο, όσο και για το μοντέλο που έχει πρώτα προεκπαιδευτεί σε ολόκληρο το training set και στη συνέχεια έγινε finetune. Είναι ενδιαφέρον ότι το προεκπαιδευμένο μοντέλο υπερτερεί του πλήρως εποπτευόμενου, και μάλιστα το όφελος παραμένει σε όλο το εύρος της διαθεσιμότητας ετικετών, ακόμη και όταν τα μοντέλα επιτρέπεται να χρησιμοποιούν όλες τις ετικέτες. Άρα υπάρχει όφελος όταν επαναχρησιμοποιούμε δείγματα αρχικά χωρίς τις ετικέτες τους στο pretraining και έπειτα κάνοντας supervised learning στις ίδιες. Το παραπάνω επιβεβαιώνεται και στον Πίνακα 5.3.

Ερώτημα 2: Αν έχουμε ένα επισημειωμένο σύνολο δεδομένων, πώς θα επηρεάσουν τα πρόσθετα unlabeled δείγματα την απόδοση; Με άλλα λόγια, σε ποιο βαθμό η μάθηση χωρίς επίβλεψη βελτιώνεται από τη συλλογή περισσότερων δεδομένων, ακόμη και αν δεν υπάρχουν διαθέσιμες ετικέτες σε αυτά;

Αυτή η ερώτηση διαφέρει από την παραπάνω, καθώς τώρα κλιμακώνεται η διαθεσιμότητα των δειγμάτων δεδομένων μόνο για *unsupervised* προεκπαίδευση, ενώ ο αριθμός των δειγμάτων με ετικέτα είναι σταθερός. Το δεξί πλαίσιο της Εικόνας 5.6.5 (όπου κάθε κουκκίδα είναι ένα πείραμα) δείχνει ότι, για έναν δεδομένο αριθμό ετικετών (εμφανίζεται ως ποσοστό των συνολικά διαθέσιμων ετικετών), όσο περισσότερα δείγματα δεδομένων χρησιμοποιούνται για μη εποπτευόμενη μάθηση, τόσο χαμηλότερο είναι το σφάλμα στο test-set (όταν ο οριζόντιος άξονας είναι 0 τότε έχουμε πλήρως εποπτευόμενη μάθηση, ενώ σε όλες οι άλλες τιμές έχουμε προεκπαίδευση ακολουθούμενη από finetuning). Αυτή η τάση είναι γραμμική στην περίπτωση της εποπτευόμενης μάθησης στο 20% των ετικετών (περίπου 2500 δείγματα). Ωστόσο, λόγω των λίγων δειγμάτων στο σύνολο, στην περίπτωση του 10% των ετικετών (περίπου 1250 δείγματα), το σφάλμα πρώτα μειώνεται γρήγορα καθώς χρησιμοποιούμε περισσότερα δείγματα για μη εποπτευόμενη προ-εκπαίδευση, στη συνέχεια αυξάνεται στιγμιαία και τελικά μειώνεται ξανά.

Σε συμφωνία με τις παραπάνω παρατηρήσεις, είναι ενδιαφέρον να σημειωθεί και πάλι ότι, για έναν δεδομένο αριθμό δειγμάτων με ετικέτα, η επαναχρησιμοποίηση ενός υποσυνόλου των *ιδίων samples* για προεκπαίδευση χωρίς επίβλεψη βελτιώνει την απόδοση:

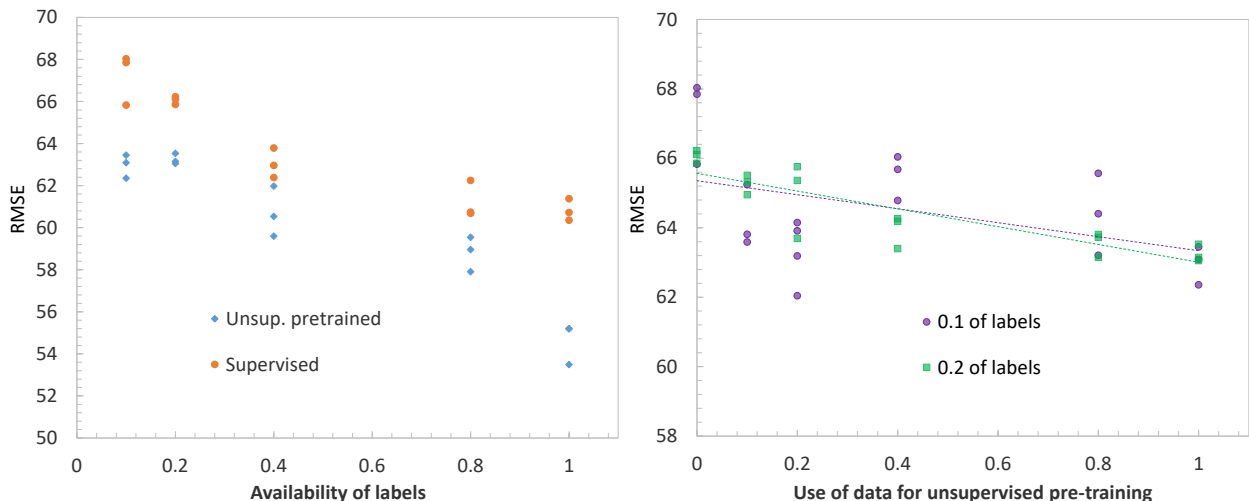
- για τις 1250 ετικέτες (μπλε διαμάντια στο δεξί πλαίσιο της Εικόνας 5.6.5 αυτό μπορεί να παρατηρηθεί στην περιοχή οριζόντιου άξονα $[0, 0, 1]$)
- για το 2500 ετικέτες (μπλε διαμάντια του δεξιού πίνακα της Εικόνας 5.6.5 στην περιοχή οριζόντιου άξονα $[0, 0, 2]$).

Τέλος, είναι ενδιαφέρον να παρατηρήσουμε ότι η αύξηση της διαθεσιμότητας unlabeled δειγμάτων από 0 σε 1250 φαίνεται να έχει ισχυρότερη επίδραση στη βελτίωση της απόδοσης σε σύγκριση με την αύξηση της διαθεσιμότητας ετικετών για εποπτευόμενη μάθηση από 1250 σε 2500.

5.7 Encoding Time Series as Images for Visual Inspection and Classification

Η παράγραφος βασίζεται στο άρθρο των Wang, Z. et al. [W014]. Όλα τα σχήματα προέρχονται από εκεί, εκτός και αν αναφέρεται διαφορετικά.

Είναι σαφές ότι ο η μεγαλύτερη εξέλιξη στον τομέα του self-supervised learning έχει γίνει στον τομέα της όρασης υπολογιστών. Άρα είναι λογικό να αναζητήσει κάποιος τρόπους μετατροπής χρονοσειρών σε εικόνες.



Σχήμα 5.6.5: **Αριστερά:** MSE στο test set πλήρως supervised μοντέλου (πορτοκαλί κύκλοι) και του ίδιου προεκπαιδευμένου μοντέλου (μπλε διαμάντια), καθώς αυξάνουμε το ποσοστό στο οποίο κάνουμε supervised learning μετά την προεκπαίδευση. **Δεξιά:** MSE στο test set ενός δεδομένου μοντέλου ως συνάρτηση του αριθμού των δειγμάτων που χρησιμοποιούνται για προεκπαίδευση χωρίς επίβλεψη. Για την εποπτευόμενη μάθηση που ακολουθεί, απεικονίζονται δύο επίπεδα διαθεσιμότητας ετικετών: 10% (μωβ κύκλοι) και 20% (πράσινα τετράγωνα). Όταν ο οριζόντιος άξονας είναι 0 έχουμε μόνο supervised εκμάθηση, ενώ όλες οι άλλες τιμές αντιστοιχούν σε προεκπαίδευση ακολουθούμενη από finetuning.

Επιπλέον λόγω των αρχιτεκτονικών που χρησιμοποιούνται στον τομέα πετυχαίνουμε invariance σε μετατοπίσεις στο χρόνο και στη συχνότητα. Δυστυχώς, όμως οι υπάρχουσες μέθοδοι μετατροπής χρονοσειρών σε εικόνα που χρησιμοποιούνται στα συστήματα αναγνώρισης φωνής όπως τα Mel frequency cepstral coefficients (MFCCs) [Xu+04] ή το perceptual linear predictive coefficient (PLPs) [Her90] είναι πολύ συγκεκριμένα στο task και αποτυγχάνουν να γενικεύσουν για τυχαίες χρονοσειρές.

5.7.1 Gramian Angular Field

Ο πρώτος μετασχηματισμός χρονοσειράς σε εικόνα, είναι το Gramian Angular Field (GAF), στο οποίο αναπαριστούμε χρονοσειρές σε πολικό σύστημα συντεταγμένων αντί για τις τυπικές καρτεσιανές συντεταγμένες. Στον πίνακα Gramian, το καθένα στοιχείο είναι στην πραγματικότητα το συνημίτονο του αθροίσματος των γωνιών.

Ορισμός 5.7.1: Χώρος εσωτερικού γινομένου

Ένας χώρος εσωτερικού γινομένου είναι ένας διανυσματικός χώρος F σε ένα πεδίο (π.χ. \mathbb{R} ή \mathbb{C}) στο οποίο ορίζεται η πράξη εσωτερικού γινομένου:

$$\langle \cdot, \cdot \rangle : V \times V \rightarrow F$$

η οποία ικανοποιεί τις παρακάτω τρεις ιδιότητες για όλα τα διανύσματα $x, y, z \in V$ και βαθμωτά $a \in F$. Το εσωτερικό γινόμενο ενός ζεύγους στοιχείων είναι ίσο με τον συζυγή μιγαδικό του εσωτερικού γινομένου των ανταλλαγμένων στοιχείων:

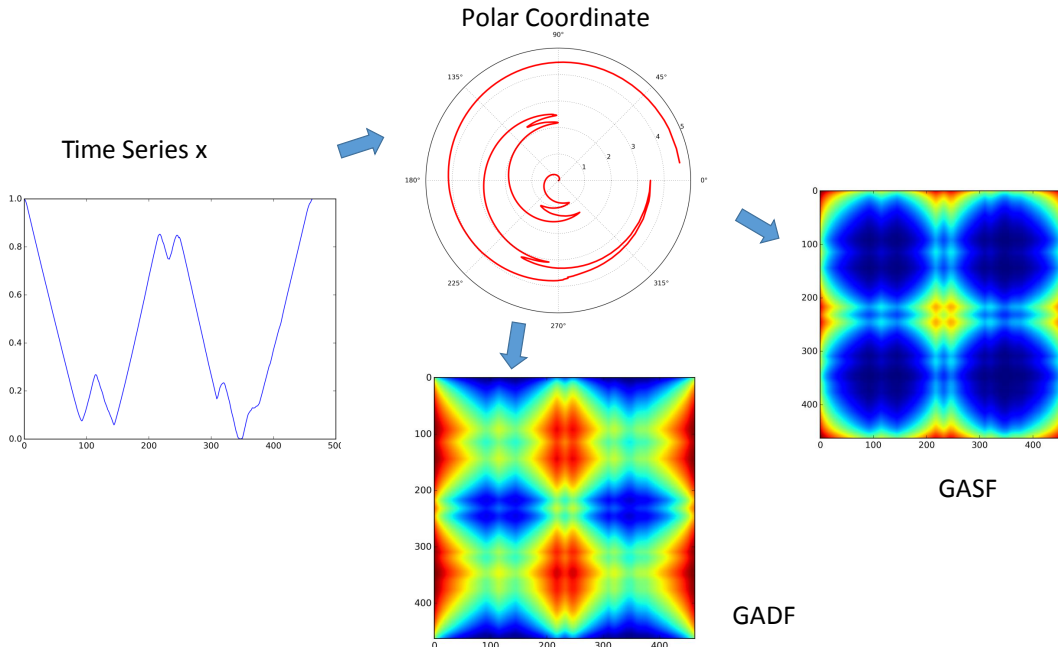
$$\langle y, x \rangle = \overline{\langle x, y \rangle}$$

Το εσωτερικό γινόμενο είναι γραμμικό ως προς την αριστερή μεταβλητή. Για όλους τους μιγαδικούς αριθμούς a και β

$$\langle ax_1 + \beta x_2, y \rangle = a\langle x_1, y \rangle + \beta\langle x_2, y \rangle$$

Το εσωτερικό γινόμενο ενός στοιχείου με τον εαυτό της είναι θετικά ορισμένο:

$$\langle x, x \rangle \geq 0$$



Σχήμα 5.7.1: Διαδικασία κατασκευής του Gramian Angular Fields. Το X είναι η rescaled χρονοσειρά. Στη συνέχεια μετατρέπουμε το rescaled X σε πολικές συντεταγμένες μέσω της Εξ. (5.7.3) και υπολογίζουμε τις GASF/GADF εικόνες από τις Εξ. (5.7.5) και (5.7.7). Στη συνέχεια κάνουμε PAA smoothing, όμως στο παράδειγμα εδώ δεν έγινε για να έχουμε υψηλή ανάλυση.

όπου η ισότητα ισχύει όταν $x = 0$.

Ορισμός 5.7.2: Gramian πίνακας

Ο Gramian πίνακας ενός συνόλου από διανύσματα v_1, \dots, v_n σε ένα χώρο εσωτερικού γινομένου είναι ο Ερμιτιανός πίνακας του οποίου οι τιμές δίνονται από την $G_{ij} = \langle v_i, v_j \rangle$. Αν τα v_1, \dots, v_n είναι στήλες του πίνακα X τότε ο Gramian είναι $X^T X$ στη γενική περίπτωση που έχουμε μιγαδικές τιμές, ή $X^T X$ όταν έχουμε πραγματικές τιμές.

Δίνεται η χρονοσειρά $X = \{x_1, x_2, \dots, x_n\}$ με n πραγματικές τιμές. Κάνουμε rescale την X είτε στο $[-1, 1]$ είτε στο $[0, 1]$ μέσω των:

$$\tilde{x}_{-1}^i = \frac{(x_i - \max(X)) + (x_i - \min(X))}{\max(X) - \min(X)} \quad (5.7.1)$$

$$\text{και} \quad \tilde{x}_0^i = \frac{x_i - \min(X)}{\max(X) - \min(X)} \quad (5.7.2)$$

Στη συνέχεια αναπαριστούμε τη rescaled χρονοσειρά \tilde{X} σε πολικές συντεταγμένες αναπαριστώντας την κάθε τιμή ως το angular cosine της τιμής και το timestamp ως την ακτίνα μέσω των εξισώσεων:

$$\begin{cases} \phi = \arccos(\tilde{x}_i), -1 \leq \tilde{x}_i \leq 1, \tilde{x}_i \in \tilde{X} \\ r = \frac{t_i}{N}, t_i \in \mathbb{N} \end{cases} \quad (5.7.3)$$

Όπου, t_i το timestamp και N μια σταθερά με σκοπό να κάνει regularize το span των πολικών συντεταγμένων. Οι εξισώσεις 5.7.3 έχουν δύο σημαντικές ιδιότητες:

- Είναι ένα προς ένα αφού το $\cos(\phi)$ είναι μονότονη συνάρτηση στο $\phi \in [0, \pi]$. Άρα για κάποια χρονοσειρά, ο μετασχηματισμός παράγει μία και μόνη εικόνα με μοναδικό αντίστροφο μετασχηματισμό.
- Επιπλέον, σε αντίθεση με τις καρτεσιανές, οι πολικές συντεταγμένες διατηρούν τις απόλυτες χρονικές συσχετίσεις. Δηλαδή το εμβαδόν από τη χρονική στιγμή i μέχρι τη στιγμή j , δεν εξαρτάται μόνο από

την απόλυτη διαφορά $|i - j|$, αλλά και από τις ίδιες τις i και j . Απόδειξη για τη δεύτερη ιδιότητα δίνεται παρακάτω:

Σε καρτεσιανές συντεταγμένες το εμβαδόν δίνεται από $S_{i,j} = \int_{x(i)}^{x(j)} f(x(t))dx(t)$ και όταν ισχύει ότι η $f(x(t))$ έχει ίδιες τιμές στα $[i, i + k]$ και $[j, j + k]$, τότε $S_{i,i+k} = S_{j,j+k}$. Ωστόσο, σε πολικές συντεταγμένες έχουμε: $S'_{i,j} = \int_{\phi(i)}^{\phi(j)} r[\phi(t)]^2 d(\phi(t))$, άρα $S'_{i,i+k} \neq S'_{j,j+k}$.

Μετά τη μετατροπή της rescaled χρονοσειράς σε πολικές συντεταγμένες ορίζουμε τα συνημίτονα αθροισμάτων/ημίτονα διαφορών μεταξύ σημείων, για να αναπαραστήσουμε τις χρονικές συσχετίσεις για διαφορετικά χρονικά διαστήματα. Το Gramian Summation Angular Field (GASF) και το Gramian Difference Angular Field (GADF) ορίζονται ως εξής:

$$GASF = \begin{bmatrix} \cos(\phi_1 + \phi_1) & \cdots & \cos(\phi_1 + \phi_n) \\ \cos(\phi_2 + \phi_1) & \cdots & \cos(\phi_2 + \phi_n) \\ \vdots & \ddots & \vdots \\ \cos(\phi_n + \phi_1) & \cdots & \cos(\phi_n + \phi_n) \end{bmatrix} \quad (5.7.4)$$

$$= \tilde{X}' \cdot \tilde{X} - \sqrt{I - \tilde{X}^2}' \cdot \sqrt{I - \tilde{X}^2} \quad (5.7.5)$$

$$GADF = \begin{bmatrix} \sin(\phi_1 - \phi_1) & \cdots & \sin(\phi_1 - \phi_n) \\ \sin(\phi_2 - \phi_1) & \cdots & \sin(\phi_2 - \phi_n) \\ \vdots & \ddots & \vdots \\ \sin(\phi_n - \phi_1) & \cdots & \sin(\phi_n - \phi_n) \end{bmatrix} \quad (5.7.6)$$

$$= \sqrt{I - \tilde{X}^2}' \cdot \tilde{X} - \tilde{X}' \cdot \sqrt{I - \tilde{X}^2} \quad (5.7.7)$$

Όπου I το μοναδιαίο διάνυσμα γραμμή $[1, 1, \dots, 1]$. Στη συνέχεια ορίζουμε τα εσωτερικά γινόμενα: $\langle x, y \rangle = x \cdot y - \sqrt{1 - x^2} \cdot \sqrt{1 - y^2}$ και $\langle x, y \rangle = \sqrt{1 - x^2} \cdot y - x \cdot \sqrt{1 - y^2}$. Τότε τα Gramian Angular Fields (GAFs) σύμφωνα με τους προηγούμενους ορισμούς είναι quasi-Gramian πίνακες $[\langle \tilde{x}_1, \tilde{x}_1 \rangle]$.²

$$\begin{bmatrix} \langle \tilde{x}_1, \tilde{x}_1 \rangle & \cdots & \langle \tilde{x}_1, \tilde{x}_n \rangle \\ \langle \tilde{x}_2, \tilde{x}_1 \rangle & \cdots & \langle \tilde{x}_2, \tilde{x}_n \rangle \\ \vdots & \ddots & \vdots \\ \langle \tilde{x}_n, \tilde{x}_1 \rangle & \cdots & \langle \tilde{x}_n, \tilde{x}_n \rangle \end{bmatrix} \quad (5.7.8)$$

Τα GAFs έχουν πολλά πλεονεκτήματα:

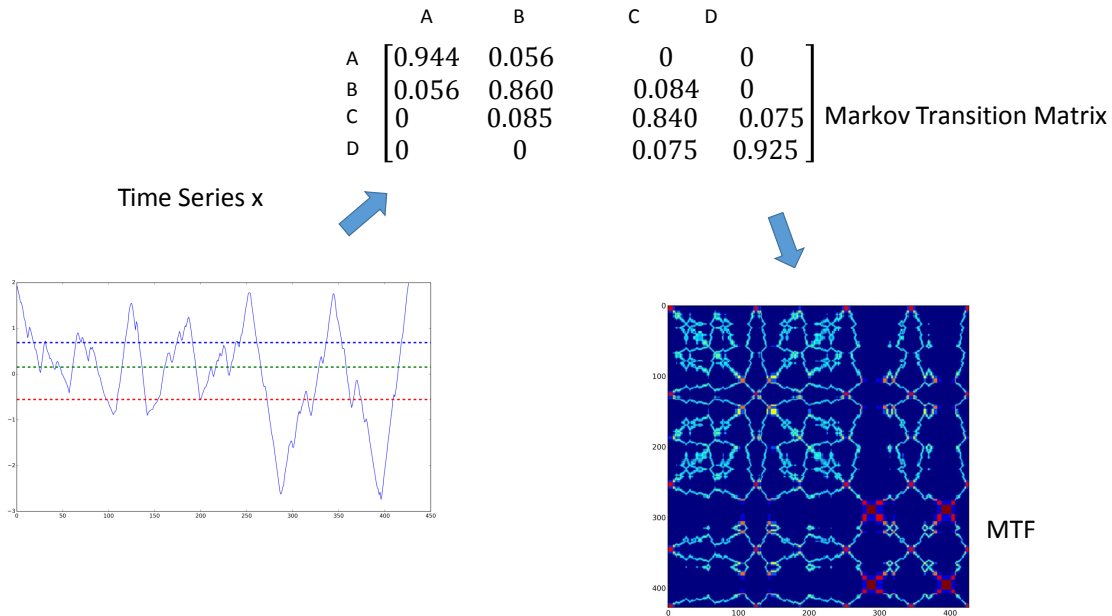
- Πρώτον, διατηρούν την χρονική αναλογία, αφού καθώς ο χρόνος αυξάνεται μετακινούμαστε στον πίνακα από πάνω-αριστερά προς τα κάτω-δεξιά.
- Εμπεριέχουν πληροφορία για τις χρονικές συσχετίσεις μεταξύ σημείων, αφού το $G_{(i,j||i-j|=k)}$ αντιπροσωπεύει τη συσχέτιση από ως προς το χρονικό διάστημα k .
- Η κύρια διαγώνιος $G_{i,i}$ είναι ειδική περίπτωση όπου $k = 0$, η οποία περιέχει την αρχική πληροφορία από την οποία μπορούμε να κάνουμε και reconstruction.

Ωστόσο, ο πίνακας των GAFs έχει μέγεθος $n \times n$, όπου n το μήκος της χρονοσειράς το οποίο είναι αρκετά μεγάλο. Για να μειώσουμε το μέγεθος αυτό εφαρμόζουμε Piecewise Aggregation Approximation (PAA) [KP00] για να εξομαλύνουμε τις χρονοσειρές, ενώ παράλληλα διατηρούμε τα trends τους. Ολόκληρη η διαδικασία φαίνεται στο Σχήμα 5.7.1.

5.7.2 Markov Transition Field

Μία δεύτερη τεχνική είναι το Markov Transition Field το οποίο κυρίως κωδικοποιεί στατιστικά τη δυναμική των μεταβάσεων (σε αντίθεση με το GAF που ήταν στατική μέθοδος).

²“quasi” επειδή τα $\langle x, y \rangle$ δεν ικανοποιούν την προϋπόθεση της γραμμικότητας.



Σχήμα 5.7.2: Οπτικοποίηση κατασκευής του Markov Transition Fields. Το X είναι μία χρονοσειρά που γίνεται discretized σε Q quantile bins. Στη συνέχεια υπολογίζουμε τον Markov Transition Matrix W και έπειτα το MTF μέσω της Εξ. (5.7.9).

Για μία χρονοσειρά X , βρίσκουμε αρχικά τα Q quantile bins και αναθέτουμε κάθε x_i στο αντίστοιχο bin q_j ($j \in [1, Q]$). Οπότε έχουμε έναν $Q \times Q$ πίνακα γειτνίασης με βάρη W , όπου μετρά μεταβάσεις μεταξύ quantile bins, δηλαδή μία Μαρκοβιανή αλυσίδα πρώτου βαθμού. Τα $w_{i,j}$ προκύπτουν από τη συχνότητα με την οποία ένα σημείο στο quantile q_j ακολουθείται από ένα σημείο στο quantile q_i . Έπειτα κάνουμε normalization με το $\sum_j w_{i,j} = 1$ και προκύπτει ο πίνακας μεταβάσεων W . Όμως αυτός είναι ανεξάρτητος από την κατανομή της X και ανεξαρτήτως των χρονικών εξαρτήσεων των t_i , αφού προκύπτει μόνο από τις τιμές της χρονοσειράς και αγνοεί τελείως το χρόνο. Αυτό είναι μεγάλη απώλεια πληροφορίας και για να το προσπεράσουμε, ορίζουμε το Markov Transition Field (MTF) ως εξής:

$$M = \begin{bmatrix} w_{ij|x_1 \in q_i, x_1 \in q_j} & \cdots & w_{ij|x_1 \in q_i, x_n \in q_j} \\ w_{ij|x_2 \in q_i, x_1 \in q_j} & \cdots & w_{ij|x_2 \in q_i, x_n \in q_j} \\ \vdots & \ddots & \vdots \\ w_{ij|x_n \in q_i, x_1 \in q_j} & \cdots & w_{ij|x_n \in q_i, x_n \in q_j} \end{bmatrix} \quad (5.7.9)$$

Έστω q_i και q_j ($q \in [1, Q]$) τα quantile bins που περιέχουν την τιμή της χρονικής στιγμής i και j αντίστοιχα. Το M_{ij} είναι η πιθανότητα μετάβασης $q_i \rightarrow q_j$. Άρα αναδιατάσσουμε τον W ο οποίος περιέχει τις πιθανότητες μετάβασης αγνοώντας την χρονική συσχέτιση σε έναν μεγαλύτερο πίνακα (MTF) που τις λαμβάνει υπόψιν.

Αναθέτοντας σε κάθε pixel M_{ij} την πιθανότητα μετάβασης από το quantile τη χρονική στιγμή i προς το quantile τη χρονική στιγμή j το MTF M το $M_{i,j||i-j|=k}$ κωδικοποιεί την πιθανότητα μετάβασης μεταξύ σημείων με χρονική απόσταση k . Η κύρια διαγώνιος M_{ii} ($k = 0$) κωδικοποιεί την πιθανότητα μετάβασης από κάθε quantile στον εαυτό του τη χρονική στιγμή i . Επειδή ο νέος πίνακας είναι $n \times n$, μειώνουμε το μέγεθος του με non-overlapping $m \times m$ blurring kernel $\{\frac{1}{m^2}\}_{m \times m}$. Ολόκληρη η διαδικασία φαίνεται στο Σχήμα 5.7.2.

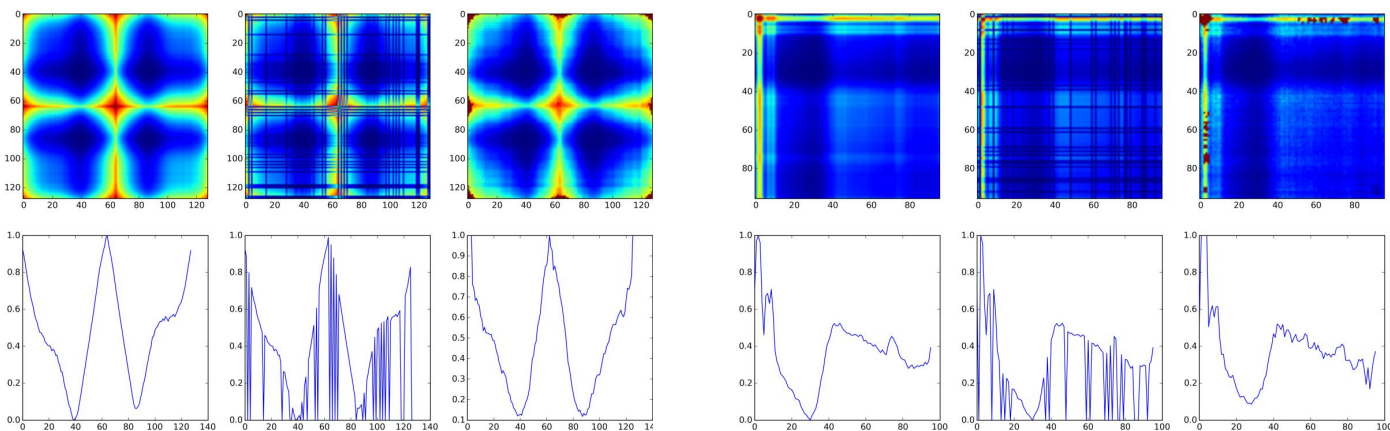
5.7.3 Σύγκριση και ανάλυση

Το MTF έχει μεγαλύτερα error rates από τα GAFs. Αυτό κυρίως οφείλεται στην αβεβαιότητα του αντίστροφου μετασχηματισμού στο MTF. Όταν κάνουμε rescale στο $[-1, 1]$ το cosine έχει γωνίες στο $[0, \pi]$ άρα και οι δύο μετασχηματισμοί δεν έχουν μοναδικό αντίστροφο (παράγουν, ωστόσο μοναδικές εικόνες). Αντίθετα, όταν έχουμε rescale στο $[0, 1]$ το cosine έχει γωνίες στο $[0, \frac{\pi}{2}]$ και ο GAF έχει μοναδικό αντίστροφο, δηλαδή μπορούμε

να κάνουμε reconstruct τη χρονοσειρά. Συγκεκριμένα μέσω της κύριας διαγωνίου $\{G_{ii}\} = \{\cos(2\phi_i)\}$ μπορούμε να ανακατασκευάσουμε την αρχική χρονοσειρά μέσω της:

$$\cos(\phi) = \sqrt{\frac{\cos(2\phi) + 1}{2}} \quad \phi \in [0, \frac{\pi}{2}] \quad (5.7.10)$$

Επίσης παρατηρούμε ότι τα GAFs κωδικοποιούν στατική πληροφορία ενώ τα MTF δυναμική. Άρα μπορούμε να τα συνδυάσουμε σαν “ορθογώνια” κανάλια αφού κωδικοποιούν ανεξάρτητες πληροφορίες. Πράγματι όταν συνδυάζουμε τα (GASF-GADF-MTF) έχουμε καλύτερες επιδόσεις σε classification tasks.



Σχήμα 5.7.3: Αρχικό GASF → “broken” GASF → recovered GASF (πάνω). Αρχική χρονοσειρά → corrupted χρονοσειρά με missing values → predicted χρονοσειρά (κάτω).

Τέλος, αφού μπορούμε να κάνουμε reconstruct μπορούμε να βάλουμε “salt-and-pepper” θόρυβο στην αρχική χρονοσειρά (missing values) και να προσπαθήσουμε να ανακατασκευάσουμε την την πρώτη. Αυτός ο denoising autoencoder μπορεί να χρησιμοποιηθεί για Time Series Imputation όπως φαίνεται στο Σχήμα 5.7.3). Μάλιστα, έχει πιο σταθερή απόδοση από το να κάναμε denoising στα raw data. Αυτό γιατί μέσω του kernel-εσωτερικού γινομένου $k(x_i, x_j)$ αυξάνουμε τη διαστατικότητα των δεδομένων. Έτσι ο denoising autoencoder λαμβάνει υπόψιν και χρονικές και χωρικές συσχετίσεις για την πρόβλεψη των missing values.

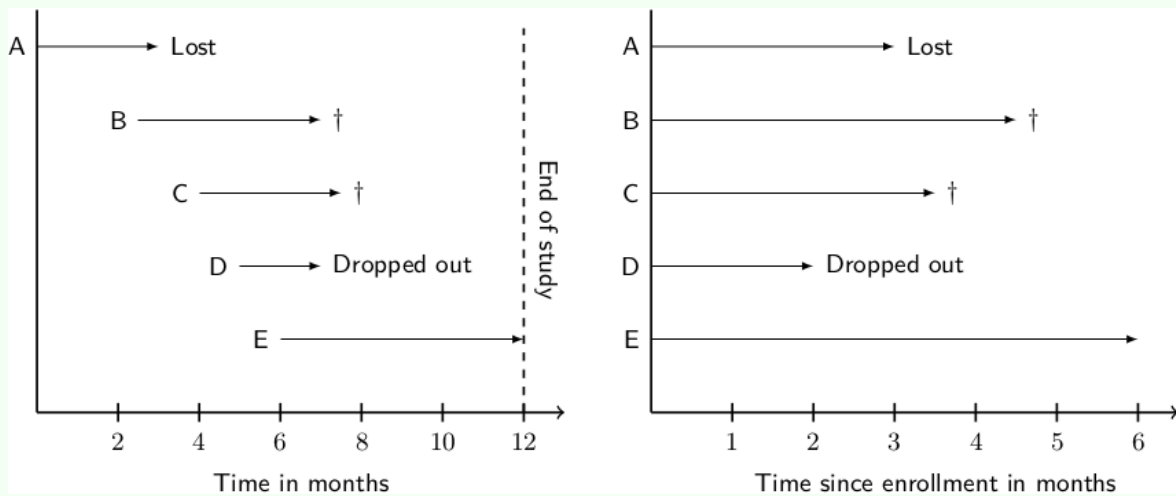
5.8 Survival Analysis

5.8.1 Εισαγωγή

Το survival analysis (αλλιώς reliability analysis ή time-to-event analysis) είναι ο κλάδος που ασχολείται με ανάλυση time-to-event δεδομένων, δηλαδή δεδομένων που αφορούν το χρόνο μέχρι κάποιο συμβάν. Ο όρος ‘survival’ υπάρχει γιατί η τεχνική αυτή χρησιμοποιήθηκε, αρχικά, σε κλινικές έρευνες, όπου κύριος στόχος ήταν η πρόβλεψη του χρόνου επιβίωσης ενός ασθενή (survival time). Το survival analysis είναι ένας τύπος προβλήματος παλινδρόμησης (regression), αφού θέλουμε να προβλέψουμε μια συνεχή τιμή, αλλά με κάποιες διαφορές τις οποίες δεν μπορούμε να διαχειριστούμε με την κλασική παλινδρόμηση. Κύρια διαφορά, είναι το γεγονός ότι τμήματα των δεδομένων εκπαίδευσης μπορούν να παρατηρηθούν μόνο εν μέρει, είναι δηλαδή ‘censored’. Αυτό φαίνεται πιο αναλυτικά στο παρακάτω παράδειγμα:

Παράδειγμα 5.8.1: Κλινική μελέτη, η οποία διερευνά τη στεφανιαία νόσο

Ας πάρουμε για παράδειγμα μια κλινική μελέτη, η οποία διερευνά τη στεφανιαία νόσο και έχει διεξαχθεί για περίοδο ενός έτους, όπως φαίνεται στο Σχήμα 5.8.1.



Σχήμα 5.8.1: Οπτικοποίηση του censoring. Σχήμα από [Pö120]

Ο ασθενής A χάθηκε από την παρακολούθηση μετά από τρεις μήνες χωρίς καταγεγραμμένο καρδιαγγειακό συμβάν, ο ασθενής B παρουσίασε ένα συμβάν τεσσεράμισι μήνες μετά την εγγραφή του και ο ασθενής C τριάντισι μήνες μετά την εγγραφή του. Όμοια, ο ασθενής D αποσύρθηκε από τη μελέτη δύο μήνες μετά την εγγραφή του και ο ασθενής E δεν παρουσίασε κανένα συμβάν πριν τελειώσει η μελέτη. Κατά συνέπεια, ο ακριβής χρόνος ενός καρδιαγγειακού συμβάντος μπορεί να καταγραφεί μόνο για τους ασθενείς B και C, των οποίων τα δεδομένα είναι 'uncensored'. Για τους υπόλοιπους ασθενείς είναι άγνωστο εάν εμφάνισαν ή όχι κάποιο συμβάν μετά τον τερματισμό της μελέτης. Οι μόνες έγκυρες πληροφορίες που είναι διαθέσιμες για τους ασθενείς A, D και E είναι ότι ήταν χωρίς συμβάντα μέχρι την τελευταία τους παρακολούθηση. Ως εκ τούτου, τα αρχικά τους είναι 'censored' και ειδικότερα 'right censored', γιατί το σημείο που έγινε το censoring είναι δεξιά του αρχικού σημείου.

Μερικά σημεία που πρέπει να προσέξουμε όταν εκτελούμε survival analysis και ισχύουν και στο παράδειγμα 5.8.1 είναι τα εξής:

- Το αρχικό και το τελικό σημείο μπορεί να είναι διαφορετικά από άτομο σε άτομο, συμβολίζουν όμως, το ίδιο γεγονός για όλους. Στο παράδειγμα 5.8.1 έχουμε αρχικό σημείο το χρόνο έναρξης στην έρευνα και τελικό το καρδιαγγειακό γεγονός ή τη χρονική στιγμή που έγινε το 'censoring'. Με αυτόν τον τρόπο μπορούμε να πάμε από το αριστερό σχήμα του 5.8.1 στο δεξί, δηλαδή να έχουμε κοινή αρχή για όλους.
- Θα μπορούσε κανείς να απορρίψει τις 'censored' εγγραφές απ' την παραπάνω έρευνα και να υπολογίσει την πιθανότητα ένα άτομο να επιβιώσει πέρα από το χρόνο t ως εξής: $\hat{S}(t) = \frac{\text{number of patients surviving beyond } t}{\text{total number of patients}}$. Αν οι 'censored' εγγραφές είχαν survival time κοντά στον μέσο όρο των 'uncensored', τότε αυτό δεν θα ήταν πρόβλημα. Όμως είναι λάθος να υποθέσουμε ότι κάτι τέτοιο ισχύει και ειδικά για datasets με πολλά censored δεδομένα. Οπότε τα 'censored' δεδομένα **πρέπει να συμπεριληφθούν στο dataset**.
- Το censoring θα πρέπει να είναι ανεξάρτητο (non-informative) από το συμβάν που θα είχε παρατηρηθεί διαφορετικά, λαμβανομένων υπόψη τυχόν επεξηγηματικών μεταβλητών που περιλαμβάνονται στην ανάλυση, διαφορετικά το συμπέρασμα θα είναι μεροληπτικό. Ένα παράδειγμα informative censoring έχει ως εξής: Σε μια μελέτη επιβίωσης μετά από τη διάγνωση μίας ασθένειας, οι ασθενείς μπορεί να χάσουν την παρακολούθηση επειδή η κατάσταση τους έχει επιδεινωθεί και δεν είναι πλέον σε θέση να παρευρεθούν. Αντίθετα, σε μια μελέτη θεραπειών για μια κατάσταση που δεν είναι απειλητική για τη ζωή, ορισμένοι ασθενείς μπορεί να εγκαταλείψουν τη θεραπεία και να χάσουν την παρακολούθηση, γιατί η κατάστασή τους έχει βελτιωθεί σημαντικά.
- Ένα άλλο χαρακτηριστικό των time-to-event δεδομένων είναι ότι οι κατανομές τους είναι συχνά ασύμμετρες (skewed/asymmetric) και επομένως απλές τεχνικές που υποθέτουν ότι τα δεδομένα έχουν κανονική κατανομή δεν μπορούν να χρησιμοποιηθούν.
- Το dataset μπορεί να είναι **truncated**, όταν δεν μπορούμε να παρατηρήσουμε δεδομένα με πολύ μικρό

(left-truncated) ή πολύ μεγάλο (right truncated) time-to-event. Ένα παράδειγμα left truncated dataset παρουσιάζεται από τους Gilbert, S. L. et al. [Gil+14] οι οποίοι εξερευνούν το survival-time νεογνών ζώων. Σε αυτή την περίπτωση τα ζώα που έχασαν τη ζωή πολύ νωρίς είναι πολύ σπάνιο να δειγματοληπτηθούν. Αντίστοιχα right truncated datasets είναι αυτά στα οποία είναι δύσκολο να δειγματοληπτήσουμε μεγάλους survival χρόνους.

5.8.2 Ορισμοί

Ορισμός 5.8.2: Τυπικοί ορισμοί [Kar16]

Έστω $T \geq 0$ μια τυχαία μεταβλητή που αντιπροσωπεύει το χρόνο επιβίωσης (ή συμβάντος). Η συνάρτηση επιβίωσης (survival function) είναι η πιθανότητα ένα άτομο να επιβιώσει πέρα από το χρόνο t ,

$$S(t) = \mathbb{P}(T > t), \quad 0 < t < \infty$$

Η συνάρτηση πυκνότητας πιθανότητας $f(t)$ είναι η συχνότητα των γεγονότων ανά μονάδα χρόνου. Σχετίζεται με τη συνάρτηση επιβίωσης ως:

$$f(t) = -\frac{dS(t)}{dt}$$

Η hazard function είναι ο στιγμιαίος ρυθμός με τον οποίο συμβαίνουν γεγονότα για άτομα που επιβιώνουν τη χρονική στιγμή t :

$$h(t) = \lim_{\delta t \rightarrow 0^+} \frac{\mathbb{P}(t \leq T < t + \delta t \mid T \geq t)}{\delta t}$$

Αντίστοιχα η cumulative hazard function, ορίζεται ως:

$$H(t) = \int_0^t h(u) du$$

Η cumulative hazard function σχετίζεται με την survival function μέσω της:

$$S(t) = e^{-H(t)}$$

Δηλαδή, όσο μεγαλύτερος είναι ο κίνδυνος, τόσο μικρότερη είναι η επιβίωση.

Έστω δ_i ίσο με 1 για κάποιο άτομο i εάν είχε συμβάν και 0 το αρχείο του i είναι ‘censored’. Στη συνέχεια, για ένα σύνολο right-censored δεδομένων, τα αρχεία του ατόμου i μπορούν να αναπαρασταθούν ως (t_i, δ_i, x_i) , όπου t_i είναι η ώρα του συμβάντος ή του censoring, δ_i είναι ένας δείκτης για το αν έχει γίνει censoring και x_i είναι ένα σύνολο από covariates, δηλαδή ένα σύνολο μεταβλητών που αντιπροσωπεύουν οποιαδήποτε άλλη πληροφορία αφορά το άτομο.

Τότε η συνάρτηση πιθανοφάνειας είναι:

$$L = \prod_{j: \text{had event}} f(t_j) \prod_{k: \text{censored}} S(t_k) = \prod_{i=1}^N h(t_i)^{\delta_i} S(t_i)$$

Δηλαδή, κάθε άτομο με ένα παρατηρούμενο χρόνο συμβάντος t_i συνεισφέρει την τιμή της hazard function τη χρονική στιγμή t_i πολλαπλασιασμένη με την τιμή της survival function την ίδια στιγμή. Παρόμοια κάθε άτομο που είναι censored την στιγμή t_i συνεισφέρει την τιμή της survival function τη χρονική στιγμή t_i .

5.8.3 Εφαρμογές του survival analysis

Το survival analysis μπορεί να χρησιμοποιηθεί ως εργαλείο για:

- Εκτίμηση και γραφική αναπαράσταση της survival function, δεδομένου ενός time-to-event dataset. Η πιο συνηθισμένη τεχνική γι' αυτό το task είναι η Kaplan-Meier η οποία είναι μη-παραμετρική μέθοδος.

Αντίστοιχες παραμετρικές κατανομές είναι οι: exponential, η Weibull και η log-logistic.

- Σύγκριση survival curves για δύο ή περισσότερα groups. Πιο συνηθισμένα εργαλεία είναι τα log-rank και Mantel-Haenzel tests.
- **Survival regression:** Δηλαδή η εξέταση, αν οι survival times μπορούν να προβλεφθούν από άλλα features/covariates. Παραδείγματα τέτοιων τεχνικών είναι το Cox proportional hazards μοντέλο και το accelerated failure time (ή accelerated life) μοντέλο.

Στην παρούσα εργασία χρησιμοποιούμε μόνο το survival regression γι' αυτό και το αναλύουμε με περισσότερη λεπτομέρεια παρακάτω:

Survival regression:

Το πιο συνηθισμένο μοντέλο για multivariate survival regression είναι το Cox proportional hazards μοντέλο. Χρησιμοποιείται συνήθως στην ιατρική έρευνα και είναι εύκολο να ερμηνευτεί (interpretable) με παρόμοιο τρόπο που ερμηνεύεται η έξοδος ενός linear regression μοντέλου. Δυστυχώς, όμως δύο σημαντικοί περιορισμοί του Cox μοντέλου είναι ότι: α) δεν μπορεί να μάθει μη-γραμμικές σχέσεις στα δεδομένα και β) απαιτεί αρκετές υποθέσεις στα δεδομένα [Kas]. Ένα τέτοιο μοντέλο έχει συνάρτηση κινδύνου:

$$\lambda(t|X_i) = \lambda_0(t) \exp(\beta_1 X_{i1} + \dots + \beta_p X_{ip}) = \lambda_0(t) \exp(X_i \cdot \beta)$$

και συνάρτηση πιθανοφάνειας για το άτομο i (partial γιατί η επίδραση των covariates μπορεί να εκτιμηθεί χωρίς την ανάγκη μοντελοποίησης της μεταβολής του κινδύνου με την πάροδο του χρόνου):

$$PL_i(\beta) = \frac{\lambda(Y_i | X_i)}{\sum_{j: Y_j \geq Y_i} \lambda(Y_j | X_j)} = \frac{\lambda_0(Y_i) \theta_i}{\sum_{j: Y_j \geq Y_i} \lambda_0(Y_j) \theta_j} = \frac{\theta_i}{\sum_{j: Y_j \geq Y_i} \theta_j}$$

Βλέπουμε ότι η συνάρτηση κινδύνου αποτελείται από δύο μέρη:

- Την baseline hazard function που συμβολίζεται με λ_0 και περιγράφει πως μεταβάλλεται η πιθανότητα ενός συμβάντος ανά μονάδα χρόνου, όταν τα covariates είναι μηδέν.
- Τις effect parameters $\exp(X_i \cdot \beta)$ που περιγράφουν πώς μεταβάλλεται ο κίνδυνος σαν συνάρτηση των explanatory covariates.

Μια περιοριστική υπόθεση που κάνει το Cox μοντέλο για τα δεδομένα και φαίνεται στην παραπάνω σχέση είναι ότι ισχύει η proportional hazards condition. Η συνθήκη αυτή δηλώνει ότι τα covariates σχετίζονται πολλαπλασιαστικά με τον κίνδυνο hazard ή ισοδύναμα ότι τα covariates είναι σταθερά στο χρόνο. Επεκτάσεις του Cox μοντέλου που επιτρέπουν χρονικά μεταβαλλόμενα covariates και πολλαπλά γεγονότα ανά άτομο φαίνονται στην εργασία των Andersen, P. K. et al. [AG82].

Επιπλέον, όπως και στο απλό linear regression, όταν ο αριθμός των συμμεταβλητών (covariates) p είναι μεγάλος σε σύγκριση με το μέγεθος του δείγματος n , ο πίνακας $X'X$ γίνεται non-singular λόγω των γραμμικών συσχετίσεων μεταξύ των features. Αυτό είναι πολύ συνηθισμένο πρόβλημα όταν έχουμε δεδομένα σε χώρους πολλών διαστάσεων και μπορεί να λυθεί μέσω των Penalized Cox Models. Τρία παραδείγματα τέτοιων μοντέλων είναι:

- **Ridge:** Εδώ προσθέτουμε έναν ℓ_2 -penalty όρο στα coefficients β_1, \dots, β_p ο οποίος συρρικνώνει το μέτρο τους. Αυτό το πετυχαίνουμε λύνοντας το εξής πρόβλημα βελτιστοποίησης:

$$\arg \max_{\beta} \log PL(\beta) - \frac{\alpha}{2} \sum_{j=1}^p \beta_j^2,$$

όπου $PL(\beta)$ η partial likelihood συνάρτηση του Cox μοντέλου β_1, \dots, β_p τα coefficients των p features και $\alpha \geq 0$ υπερπαράμετρος που ρυθμίζει το μέγεθος της συρρίκνωσης.

- **LASSO:** Αν και το ℓ_2 -penalty λύνει το μαθηματικό πρόβλημα του non-singularity και μας επιτρέπει να εκπαιδύσουμε ένα μοντέλο Cox, ιδανικά, θα θέλαμε να επιλέξουμε ένα μικρό υποσύνολο χαρακτηριστικών που έχουν την περισσότερη προγνωστική ισχύ και να αγνοήσουμε τα υπόλοιπα. Αυτό ακριβώς κάνει το

LASSO (Least Absolute Shrinkage and Selection Operator) αντικαθιστώντας το ℓ_2 -penalty με ℓ_1 , δηλαδή λύνοντας το εξής πρόβλημα βελτιστοποίησης:

$$\arg \max_{\beta} \log \text{PL}(\beta) - \alpha \sum_{j=1}^p |\beta_j|.$$

- **Elastic Net:** Το LASSO είναι ένα εξαιρετικό εργαλείο για να επιλέξουμε ένα υποσύνολο από discriminative features, αλλά έχει δύο βασικά μειονεκτήματα. Πρώτον, δεν μπορεί να επιλέξει περισσότερα χαρακτηριστικά από τον αριθμό των δειγμάτων στα δεδομένα εκπαίδευσης, κάτι που είναι προβληματικό όταν έχουμε δεδομένα πολύ υψηλών διαστάσεων. Δεύτερον, εάν τα δεδομένα περιέχουν μια ομάδα χαρακτηριστικών που έχουν υψηλή συσχέτιση, η ποινή LASSO θα επιλέξει τυχαία ένα χαρακτηριστικό από αυτήν την ομάδα. Η ποινή Elastic Net ξεπερνά αυτά τα προβλήματα χρησιμοποιώντας έναν σταθμισμένο συνδυασμό των ποινών ℓ_1 και ℓ_2 με επίλυση του:

$$\arg \max_{\beta} \log \text{PL}(\beta) - \alpha \left(r \sum_{j=1}^p |\beta_j| + \frac{1-r}{2} \sum_{j=1}^p \beta_j^2 \right),$$

όπου $r \in [0; 1[$ το σχετικό βάρος μεταξύ ℓ_1 και ℓ_2 penalties.

Random Survival Forests

Τα Random Survival Forests είναι ensembles μοντέλων, τα οποία έχουν δέντρα αποφάσεων ως base learners και αποτελούν επέκταση των κλασικών Random Forests [Ho98] για right-censored δεδομένα. Στην εργασία των Ishwaran et al. [Ish+08] αναφέρεται ένας γενικός αλγόριθμος που μπορεί να χρησιμοποιηθεί για να εκπαιδεύσουμε Survival Forest μοντέλα:

1. Δειγματοληπτούμε τυχαία υποσύνολα ίδιου μεγέθους από το αρχικό σύνολο δεδομένων με αντικατάσταση. Τα δείγματα που περισσεύουν σε κάθε δειγματοληψία, λέγονται out-of-bag (OOB).
2. Εκπαιδεύουμε ένα survival tree σε κάθε ένα από τα $b = 1, \dots, B$ υποσύνολα.
 - (a) Σε κάθε κόμβο, επιλέγουμε ένα τυχαίο υποσύνολο των features. Στη συνέχεια, βρίσκουμε την καλύτερη τιμή c^* που χωρίζει το σύνολο σε δύο υποσύνολα (τους θυγατρικούς κόμβους), έτσι ώστε να μεγιστοποιείται η διαφορά σε μία δεδομένη συνάρτηση στόχο (objective function).
 - (b) Επαναλαμβάνουμε το (α) αναδρομικά σε κάθε θυγατρικό κόμβο μέχρι να ικανοποιηθεί ένα δεδομένο κριτήριο διακοπής.
3. Υπολογίζουμε την αθροιστική συνάρτηση κινδύνου (CHF) για κάθε δέντρο και παίρνουμε τον μέσο όρο τους για όλα τα B δέντρα, ο οποίος λέγεται και ensemble CHF.
4. Υπολογίζουμε το σφάλμα πρόβλεψης που δίνει η ensemble CHF χρησιμοποιώντας μόνο τα OOB δεδομένα.

Συγκεκριμένα για το βήμα 2. που είναι και το πιο σημαντικό:

Σε κάθε κόμβο, επιλέγουμε ένα feature x από το υποσύνολο των τυχαία επιλεγμένων features και μια τιμή διαχωρισμού c . Το c είναι μια από τις μοναδικές τιμές του x στο training-set.

Αναθέτουμε κάθε μεμονωμένο δείγμα i είτε στον δεξιό θυγατρικό κόμβο, αν $x_i \leq c$ ή στον αριστερό αν $x_i > c$. Στη συνέχεια υπολογίζουμε την τιμή του log-rank test έτσι ώστε:

$$L(x, c) = \frac{\sum_{i=1}^N \left(d_{i,1} - Y_{i,1} \frac{d_i}{Y_i} \right)}{\sqrt{\sum_{i=1}^N \frac{Y_{i,1}}{Y_i} \left(1 - \frac{Y_{i,1}}{Y_i} \right) \left(\frac{Y_i - d_i}{Y_i - 1} \right) d_i}} \quad (5.8.1)$$

όπου:

- j : Θυγατρικοί κόμβοι $j \in \{1, 2\}$.
- $d_{i,j}$: Αριθμός γεγονότων τη στιγμή t_i στον Θυγατρικό κόμβο j .

- $Y_{i,j}$: Αριθμός ατόμων που βίωσαν ένα γεγονός ή βρίσκονται σε κίνδυνο τη στιγμή t_i στον Θυγατρικό κόμβο j .
- d_i : Αριθμός γεγονότων τη στιγμή t_i , έτσι ώστε: $d_i = \sum_j d_{i,j}$.
- Y_i : Αριθμός ατόμων που αντιμετώπισαν ένα γεγονός ή σε κίνδυνο τη στιγμή t_i έτσι $Y_i = \sum_j Y_{i,j}$.

Επαναλαμβάνουμε για κάθε x και c , μέχρι να βρούμε x^* και c^* που να ικανοποιούν $|L(x^*, c^*)| \geq |L(x, c)|$ για κάθε x και c . Τέλος, τα μοντέλα **Extra Survival Trees** χρησιμοποιούν την objective function με τα μοντέλα Random Survival Forest. Όμως για κάθε feature x , αντί να χρησιμοποιούμε τις μοναδικές τιμές του x για να βρούμε την καλύτερη τιμή διαχωρισμού c^* , χρησιμοποιούμε N_{splits} τιμές που προέρχονται από μια ομοιόμορφη κατανομή στο διάστημα $[\min(x), \max(x)]$.

Survival Support Vector Machine

Τα Survival Support Vector Machines είναι μια επέκταση των κλασικών SVM's [CV95; BGV92] σε right-censored time-to-event data. Το κύριο πλεονέκτημα της μεθόδου, είναι ότι μπορεί να μάθει πολύπλοκες, μη γραμμικές σχέσεις μεταξύ των χαρακτηριστικών και της του χρόνου επιβίωσης μέσω του λεγόμενου kernel trick. Μια kernel συνάρτηση προβάλλει τα χαρακτηριστικά εισόδου σε χώρους υψηλών διαστάσεων όπου η survival function μπορεί να περιγραφεί από ένα υπερεπίπεδο. Αυτό κάνει τα Survival Support Vector Machines εξαιρετικά ευέλικτα και εφαρμόσιμα σε ένα ευρύ φάσμα δεδομένων. Ένα δημοφιλές παράδειγμα για μια τέτοια συνάρτηση πυρήνα είναι η Radial Basis Function.

Το πρόβλημα του Survival Analysis στο πλαίσιο των Survival Support Vector Machines, μπορεί να περιγραφεί με δύο διαφορετικούς τρόπους:

- Ως πρόβλημα κατάταξης (ranking): το μοντέλο μαθαίνει να αναθέτει σε χαμηλότερη κατάταξη δείγματα με μικρότερους χρόνους επιβίωσης, λαμβάνοντας υπόψη όλα τα πιθανά ζεύγη δειγμάτων στα δεδομένα εκπαίδευσης.
- Ως πρόβλημα παλινδρόμησης (regression): το μοντέλο μαθαίνει να προβλέπει άμεσα τον log-χρόνο επιβίωσης.

Στην περίπτωση του Linear Survival Support Vector Machine, δηλαδή ακριβώς πριν γίνει το kernel trick και αν ορίσουμε τα training data ως triplets της μορφής $(\mathbf{x}_i, y_i, \delta_i)$, όπου \mathbf{x}_i ένα d -διάστατο feature vector, $y_i > 0$ το survival time ή ο χρόνος του censoring και $\delta_i \in \{0, 1\}$ δυαδική μεταβλητή που δηλώνει εάν έγινε event. Χρησιμοποιώντας τα δεδομένα εκπαίδευσης, ο στόχος είναι να ελαχιστοποιηθεί η ακόλουθη συνάρτηση:

$$\arg \min_{\mathbf{w}, b} \frac{1}{2} \mathbf{w}^T \mathbf{w} + \frac{\alpha}{2} \left[r \sum_{i,j \in \mathcal{P}} \max(0, 1 - (\mathbf{w}^T \mathbf{x}_i - \mathbf{w}^T \mathbf{x}_j))^2 + (1 - r) \sum_{i=0}^n (\zeta_{\mathbf{w}, b}(y_i, x_i, \delta_i))^2 \right] \quad (5.8.2)$$

$$\zeta_{\mathbf{w}, b}(y_i, \mathbf{x}_i, \delta_i) = \begin{cases} \max(0, y_i - \mathbf{w}^T \mathbf{x}_i - b) & \text{if } \delta_i = 0, \\ y_i - \mathbf{w}^T \mathbf{x}_i - b & \text{if } \delta_i = 1, \end{cases} \quad (5.8.3)$$

$$\mathcal{P} = \{(i, j) \mid y_i > y_j \wedge \delta_j = 1\}_{i,j=1,\dots,n} \quad (5.8.4)$$

Η υπερ-παράμετρος $\alpha > 0$ καθορίζει το ποσό του regularization που θα εφαρμοστεί: μικρότερη τιμή αυξάνει το regularization και υψηλότερη τιμή μειώνει το regularization. Η υπερπαράμετρος $r \in [0; 1]$ καθορίζει το trade-off μεταξύ του στόχου κατάταξης και του στόχου παλινδρόμησης. Αν $r = 1$ έχουμε μόνο στόχο κατάταξης και αν $r = 0$ στόχο παλινδρόμησης.

Τέλος, το Kernel Survival Support Vector Machine είναι μια γενίκευση του Linear Survival Support Vector Machine, όπου πρώτα προβάλλουμε τα δεδομένα μας μέσω ενός kernel function (π.χ. του Radial Basis Function $k(x, x') = \exp(-\gamma \|x - x'\|^2)$). Το μειονέκτημα είναι ότι η επιλογή της συνάρτησης πυρήνα και των υπερπαραμέτρων της συχνά δεν είναι απλή και απαιτεί tuning για να ληφθούν καλά αποτελέσματα (π.χ. tuning της υπερπαραμέτρου γ στη Radial Basis Function).

Evaluation metrics:

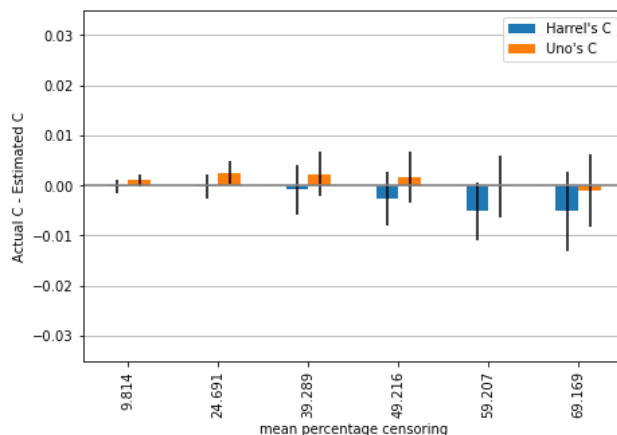
Τα test δεδομένα μας μπορεί επίσης να έχουν censored εγγραφές, επομένως μετρικές που χρησιμοποιούνται στο απλό regression, όπως το mean-squared-error είναι αδύνατο να εφαρμοστούν.

Concordance index (c index ή c statistic)

Η πιο συχνά χρησιμοποιούμενη μετρική αξιολόγησης των survival μοντέλων είναι ο concordance index. Είναι μία επέκταση του Kendall τ , δηλαδή ένα μέτρο του rank correlation μεταξύ των προβλεπόμενων σκόρ κινδύνου \hat{f} και των παρατηρούμενων χρονικών σημείων y . Ορίζεται ως η αναλογία σωστά διατεταγμένων (concordant) ζευγών προς συγκρίσιμα (comparable) ζεύγη. Δύο δείγματα i και j είναι συγκρίσιμα εάν το δείγμα με χαμηλότερο παρατηρούμενο χρόνο y είχε ένα γεγονός, δηλαδή εάν $y_j > y_i$ και $\delta_i = 1$, όπου δ_i είναι ο δυαδικός δείκτης συμβάντος. Ένα συγκρίσιμο ζεύγος (i, j) είναι concordant, εάν ο εκτιμώμενος κίνδυνος \hat{f} από ένα survival μοντέλο είναι υψηλότερος για άτομα με χαμηλότερο χρόνο επιβίωσης, δηλαδή $\hat{f}_i > \hat{f}_j \wedge y_j > y_i$, διαφορετικά το ζεύγος είναι discordant.

Ενώ η παραπάνω μετρική είναι εύκολο να ερμηνευτεί και να υπολογιστεί, έχει ορισμένες ελλείψεις:

- Έχει αποδειχθεί ότι είναι optimistic, καθώς αυξάνεται το ποσοστό των censored εγγραφών [Uno+11]. Στην ίδια εργασία οι Uno et al. προτείνουν μια παραλλαγή που ανταποκρίνεται καλύτερα όσο αυξάνεται το mean percentage censoring. Στο Σχήμα 5.8.2 φαίνεται η διαφορά του πραγματικού C που έχει προκύψει από simulation μείον το C που προκύπτει από τις δύο μεθόδους. Βλέπουμε ότι η μέθοδος των Uno et al. δεν υπερεκτιμά το C καθώς αυξάνεται το mean percentage censoring.
- Δεν μπορεί να χρησιμοποιηθεί σε περιπτώσεις που θέλουμε να αξιολογήσουμε για συγκεκριμένο διάστημα (π.χ. πρόβλεψη συμβάντος εντός 2 ετών). Γι αυτό το λόγο χρησιμοποιούνται εναλλακτικές όπως η Time-dependent Area under the ROC που εξηγείται παρακάτω.



Σχήμα 5.8.2: Σύγκριση Uno's c με Harrel's c . Βλέπουμε ότι ο εκτιμητής του Uno είναι σταθερός ενώ εκείνος του Harrel υπερεκτιμά το c για μεγάλο ποσοστό του censoring.

Time-dependent Area under the ROC

Το εμβαδόν κάτω από την καμπύλη receiver operating characteristics (ROC) είναι ένα δημοφιλές μέτρο απόδοσης για tasks δυαδικής ταξινόμησης. Συγκεκριμένα αν ορίσουμε:

Ορισμός 5.8.3: Βασικές έννοιες

True Positive (TP): Όταν το αποτέλεσμα του αλγορίθμου προβλέπει σωστά την παρουσία ενός χαρακτηριστικού.

True Negative (TN): Όταν το αποτέλεσμα του αλγορίθμου προβλέπει σωστά την απουσία ενός χαρακτηριστικού.

False Positive (FP): Όταν το αποτέλεσμα του αλγορίθμου προβλέπει ότι υπάρχει ένα χαρακτηριστικό, ενώ στην πραγματικότητα, αυτό, απουσιάζει.

False Negative (FN): Όταν το αποτέλεσμα του αλγορίθμου προβλέπει ότι απουσιάζει ένα χαρακτηριστικό.

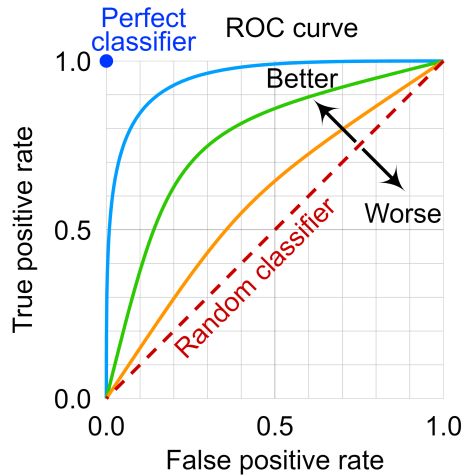
τικό, ενώ στην πραγματικότητα, αυτό, υπάρχει.

$$\text{False Positive Rate: } \text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$

$$\text{True Positive Rate: } \text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Τότε η καμπύλη ROC είναι η μεταβολή των TPR (probability of detection) και FPR (probability of false alarm), καθώς αλλάζει το discrimination threshold του ταξινομητή (Σχήμα 5.8.3). Στον ιατρικό τομέα, χρησιμοποιείται συχνά για τον προσδιορισμό του πόσο καλά μπορούν τα risk scores να διαχωρίσουν τους ασθενείς (cases) από τους υγιείς (controls). Δεδομένου του risk score \hat{f} , η καμπύλη ROC συγκρίνει το false positive rate με το true positive rate για κάθε πιθανή τιμή του \hat{f} .

Επεκτείνοντας την καμπύλη ROC σε συνεχή αποτελέσματα, και συγκεκριμένα σε survival χρόνους, η κατάσταση της νόσου ενός ασθενούς μπορεί να αλλάζει με την πάροδο του χρόνου: κατά την εγγραφή ένα άτομο είναι συνήθως υγιές, αλλά μπορεί να νοσήσει σε κάποια μεταγενέστερη χρονική στιγμή. Κατά συνέπεια, το sensitivity και το specificity γίνονται time-dependent μετρικές. Οπότε, εξετάζουμε τα αθροιστικά cases και αντίστοιχα τους αθροιστικούς controls σε ένα δεδομένο χρονικό σημείο t , και άρα έχουμε αθροιστική ROC για κάθε στιγμή t . Αθροιστικά cases είναι όλα τα άτομα που βίωσαν ένα συμβάν πριν ή τη στιγμή t ($t_i \leq t$), ενώ τα αθροιστικά controls είναι αυτά με $t_i > t$. Υπολογίζοντας το εμβαδόν κάτω από την αθροιστική ROC τη χρονική στιγμή t , μπορούμε να προσδιορίσουμε πόσο καλά ένα μοντέλο μπορεί να διακρίνει τα άτομα (i) που βιώνουν κάποιο γεγονός πριν τη στιγμή t ($t_i \leq t$) από εκείνα που βιώνουν μετά την t ($t_i > t$). Ως εκ τούτου, η μετρική αυτή μπορεί να χρησιμοποιηθεί, εάν κάποιος θέλει να προβλέψει την εμφάνιση ενός γεγονότος σε μια περίοδο μέχρι το χρόνο t . Η ερμηνεία είναι πανομοιότυπη με την παραδοσιακή area under the ROC curve (AUC) για δυαδική ταξινόμηση: η τιμή 0,5 υποδηλώνει ένα τυχαίο μοντέλο, η τιμή 1,0 υποδηλώνει ένα τέλει μοντέλο και η τιμή 0,0 υποδηλώνει ένα εντελώς λάθος μοντέλο.



Σχήμα 5.8.3: Τυπική καμπύλη ROC.

Time-dependent Brier Score

Το time-dependent Brier Score είναι μια επέκταση του μέσου τετραγώνου του σφάλματος στα right-censored δεδομένα. Με δεδομένο ένα χρονικό σημείο t , ορίζεται ως:

$$\text{BS}^c(t) = \frac{1}{n} \sum_{i=1}^n I(y_i \leq t \wedge \delta_i = 1) \frac{(0 - \hat{\pi}(t|\mathbf{x}_i))^2}{\hat{G}(y_i)} + I(y_i > t) \frac{(1 - \hat{\pi}(t|\mathbf{x}_i))^2}{\hat{G}(t)},$$

όπου $\hat{\pi}(t|\mathbf{x})$ η πιθανότητα που δίνει το μοντέλο να μην έχουμε γεγονός μέχρι τη στιγμή t , δεδομένου του feature-vector \mathbf{x} και $1/\hat{G}(t)$ η αντίστροφη πιθανότητα του censoring weight. Η μετρική Brier χρησιμοποιείται συχνά για την αξιολόγηση του μοντέλου ως προς το calibration. Εάν ένα μοντέλο προβλέπει κίνδυνο 10% να συμβεί ένα γεγονός τη χρονική στιγμή t , η παρατηρούμενη συχνότητα στα δεδομένα θα πρέπει να ταιριάζει με αυτό το ποσοστό για ένα calibrated μοντέλο. Επιπλέον, η μετρική Brier είναι μετρική διάκρισης (measure of discrimination), δηλαδή αυξάνεται όσο το μοντέλο προβλέπει σωστά τη σειρά των γεγονότων. Αντίθετα το concordance index είναι μόνο μετρική διάκρισης. Πράγματι, το c-index αγνοεί τις πραγματικές τιμές των

προβλεπόμενων βαθμολογιών κινδύνου –αφού είναι ranking metric– άρα δεν είναι σε θέση να μας πει τίποτα σχετικά με το calibration του μοντέλου.

Κεφάλαιο 6

Πειράματα και Αποτελέσματα

6.1	Εισαγωγή	96
6.1.1	Περιγραφή δεδομένων	96
6.1.2	Επιλογή μεθοδολογίας	96
6.2	Περιγραφή Πειραμάτων	97
6.3	Προβλήματα κατηγοριοποίησης	97
6.3.1	Μεθοδολογία	98
6.3.2	Αποτελέσματα	100
6.3.3	Αυξάνοντας σταδιακά τη διαθεσιμότητα των labels	101
6.3.4	Αυξάνοντας σταδιακά τη διαθεσιμότητά των unlabeled δεδομένων για pretrain	101
6.3.5	Αποτελέσματα όταν έχουμε fully-labeled dataset	103
6.4	Survival Analysis	104
6.4.1	Μεθοδολογία	108
6.4.2	Αποτελέσματα και σχολιασμός	111

6.1 Εισαγωγή

6.1.1 Περιγραφή δεδομένων

Τα δεδομένα της παρούσας εργασίας, καταγράφονται συνεχώς από ένα έξυπνο ρολόι Samsung Gear S3 Frontier που διαθέτουν όλοι οι συμμετέχοντες και τα οποία συλλέγονται σε μια πλατφόρμα που βασίζεται σε cloud [Mag+20]. Η συλλογή των δεδομένων ξεκίνησε τον Νοέμβριο του 2019 και συνεχίζεται μέχρι και σήμερα.

Συγκεκριμένα, έχουμε μετρήσεις της γραμμικής επιτάχυνσης σε 3 άξονες (αξελερόμετρο/accelerometer - acc), της γωνιακής επιτάχυνσης σε 3 άξονες (γυροσκόπιο/gyroscope - gyr), τους καρδιακούς παλμούς κάθε λεπτό (hrm) και τα RR διαστήματα, τα οποία είναι χρονικά διαστήματα μεταξύ δύο διαδοχικών καρδιακών παλμών. Οι δύο πρώτες μετρήσεις δειγματοληπτούνται στα 20 Hz, ενώ μετρήσεις από τον αισθητήρα του καρδιακού παλμού δειγματοληπτούνται στα 5 Hz. Επιπλέον έχουμε ετικέτες για τα βήματα, τη διανυόμενη απόσταση και το πρόγραμμα του ύπνου οι οποίες προκύπτουν από το Tizen API [Mag+20] που έχει το έξυπνο ρολόι.

Συνολικά στην έρευνα συμμετέχουν 64 άτομα (26 controls και 38 patients). Οι ασθενείς υποβάλλονται σε μηνιαίες αξιολογήσεις από τους κλινικούς ιατρούς του έργου. Οι ιατροί σημειώνουν τις περιόδους του ασθενή που δεν έχει εμφανίσει συμπτώματα ψυχωτικής υποτροπής ως ‘Φυσιολογική’ (Normal-N) και ως ‘Υποτροπή’ (Relapse-R) όταν ο ασθενής έχει εμφανίσει συμπτώματα.

6.1.2 Επιλογή μεθοδολογίας

Το παραπάνω dataset έχει τις εξής ιδιαιτερότητες:

1. Έχει τεράστιο όγκο δεδομένων. Συγκεκριμένα έχουμε δεδομένα από το 2019 μέχρι και σήμερα, για 64 χρήστες, με 3 σένσορες για κάθε χρήστη και ρυθμούς δειγματοληψίας 20 Hz για τα acc, gyr και 5 Hz για το hrm.
2. Παρότι υπάρχουν ετικέτες για τις περιόδους που ο χρήστης περπατά, κοιμάται, ή βρίσκεται σε υποτροπή, αυτές είναι πολύ λίγες σε σχέση με τον τεράστιο όγκο δεδομένων που διαθέτουμε. Πράγματι είναι πλεονασμός να σπαταλήσουμε ένα τόσο μεγάλο όγκο δεδομένων για ένα classification πρόβλημα με δύο μόνο κλάσεις (π.χ. sleep-awake).
3. Υπάρχουν εργασίες, όπως για παράδειγμα η αναγνώριση δραστηριότητας (human activity recognition), οι οποίες έχειδειχθεί ότι μπορούν να μοντελοποιηθούν με μεγάλη ακρίβεια χρησιμοποιώντας δεδομένα από wearables [MJ20], σαν τα δικά μας. Δυστυχώς, όμως, δεν καθόλου επισημειώσεις για τέτοια tasks.

Οπότε, σκοπός της παρούσας εργασίας είναι να αξιοποιήσει όσο το δυνατόν καλύτερα τον τεράστιο όγκο δεδομένων που έχουμε, χωρίς να βασιστεί στις ετικέτες. Αυτό μπορούμε να το πετύχουμε υποθέτοντας ότι δεν έχουμε καθόλου επισημειώσεις και ψάχνουμε να βρούμε την καλύτερη μέθοδο όπου πετυχαίνει τα εξής:

1. Όταν τελικά αποκτήσουμε labels για κάποιο task, να μπορεί προσαρμόζεται σε αυτό μέσω finetuning. Καλύτερη θεωρείται η μέθοδος που προσαρμόζεται με τα λιγότερα labels.
2. Αν αποκτήσουμε labels για δύο ή παραπάνω tasks να πετυχαίνει καλές επιδόσεις για τα περισσότερα από αυτά (generalizability). Νικήτρια η μέθοδος που πετυχαίνει καλές επιδόσεις για τον μεγαλύτερο αριθμό από tasks.
3. Αν αποκτήσουμε περισσότερα unlabeled δεδομένα να υπάρχει τρόπος αξιοποίησής τους. Νικήτρια η μέθοδος που η απόδοσή της αυξάνεται (συνεχώς) όσο αυξάνεται ο αριθμός των unlabeled δεδομένων που κάνουμε pretrain.

Κάθε μία από τις παραπάνω ιδιότητες προσφέρει και πλεονεκτήματα στη διαδικασία συλλογής των δεδομένων. Συγκεκριμένα, με την ίδια σειρά που αναφέρθηκαν οι ιδιότητες, προκύπτουν τα εξής πλεονεκτήματα:

1. Είναι δύσκολο να συλλέξουμε labels για κάποιο task οπότε θέλουμε με πολύ λίγες επισημειώσεις (π.χ. ένα πείραμα λίγων ωρών για sleep stage recognition) να πετυχαίνουμε ικανοποιητική απόδοση.
2. Αφού έχει γίνει συλλογή δεδομένων για κάποιο task θέλουμε να μπορούμε να αξιοποιήσουμε τη γνώση αυτή και σε άλλες εργασίες (για παράδειγμα από sleep-stage-recognition σε human-activity-recognition εργασία).

3. Η συλλογή unlabeled δεδομένων από τα ρολόγια είναι σχετικά εύκολη, οπότε θέλουμε η ποιότητα των αναπαραστάσεων να βελτιώνεται όσο αποκτούμε περισσότερη πληροφορία.

6.2 Περιγραφή Πειραμάτων

Τα πειράματα που πραγματοποιήθηκαν μπορούν να χωριστούν σε δύο σκέλη:

- Μία σειρά πειραμάτων που εξετάζει προβλήματα κατηγοριοποίησης.
- Μία σειρά πειραμάτων που εξετάζει το πρόβλημα του survival-analysis.

Το πρώτο σκέλος μπορεί να χωριστεί σε 3 tasks, όσα είναι και τα είδη των επισημειώσεων που έχουμε. Ειδικότερα, έχουμε labels για το πρόγραμμα ύπνου, για το αν ο χρήστης περπατά και για το μοναδικό id του χρήστη που φοράει το ρολόι.

Το δεύτερο σκέλος, αφορά τα labels που έχουμε για τις περιόδους όπου ο χρήστης βρίσκεται σε υποτροπή. Σε αυτό προσπαθούμε με βάση τις μετρήσεις του ρολογιού σε μία τυχαία μέρα να προβλέψουμε το χρονικό διάστημα μέχρι την υποτροπή του χρήστη.

Και τα δύο σκέλη, αφορούν σύγκριση μεθόδων ως προς τις τρεις ιδιότητες που αναφέρονται στην Παράγραφο 6.1.2, δηλαδή ψάχνουν την καλύτερη μέθοδο που αξιοποιεί με τον καλύτερο τρόπο τα δεδομένα, χωρίς τη χρήση επισημειώσεων. Η προεπεξεργασία που έγινε στα δεδομένα είναι διαφορετική για το κάθε σκέλος και παρουσιάζεται στις επόμενες παραγράφους.

6.3 Προβλήματα κατηγοριοποίησης

Έχουμε τριών ειδών επισημειώσεις, ως προς τις οποίες μπορούμε να κάνουμε κατηγοριοποίηση:

1. **sleep:** Με 3 πιθανές κλάσεις: sleeping, awake, transition.
2. **step:** Το οποίο έχει, επίσης, 3 πιθανές κλάσεις: walking, not walking, transition.
3. **id:** Για το υποσύνολο του dataset που χρησιμοποιήθηκε σε αυτό το πρόβλημα έχουμε 10 πιθανές κλάσεις: το μοναδικό id του χρήστη που φοράει το smartwatch.

Οι κύριες αποφάσεις που πρέπει να πάρουμε σχετικά με τη δημιουργία του dataset είναι:

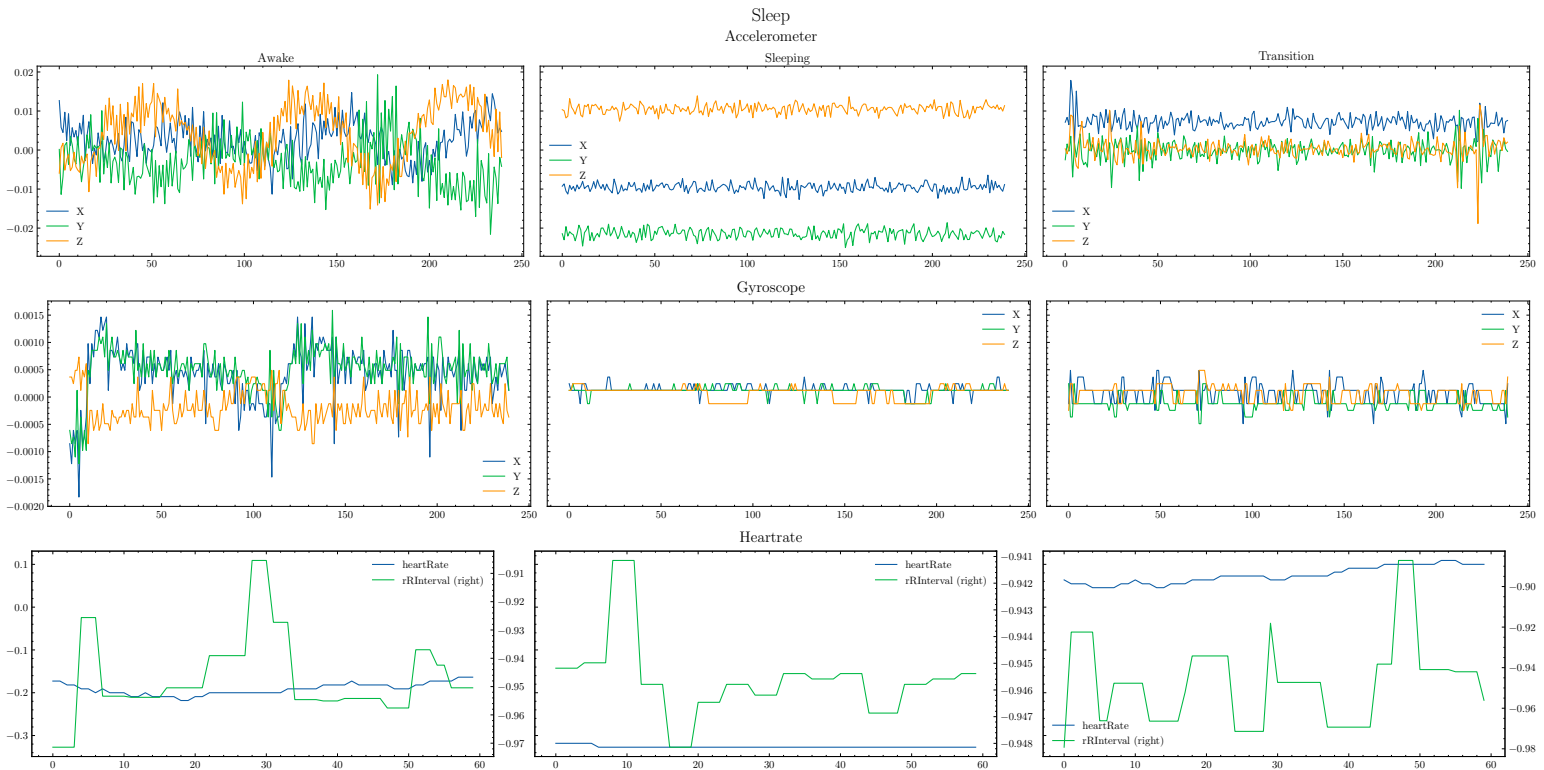
- **Η επιλογή του χρονικού διαστήματος που θα έχει ένα training sample.** Ιδανικά, για το task του person identification, θέλουμε διάστημα αρκετά μικρό για να μάθουμε μικροσυμπεριφορές και όχι καθημερινές συνήθειες, και αρκετά μεγάλο ώστε να καταγράψουμε βασικές δραστηριότητες, όπως περπάτημα, τρέξιμο κ.λπ. Όμοια, για τα sleep και step tasks θέλουμε αρκετά μικρό διάστημα για να αποφύγουμε την επικάλυψη ετικετών (π.χ. ο χρήστης να περπάτησε για λίγο, να σταμάτησε και στη συνέχεια να περπάτησε ξανά). Τελικά, καταλήγουμε στο ένα λεπτό όπου βλέπουμε ότι έχουμε ικανοποιητική απόδοση στο πιο δύσκολο από τα τρία tasks (identification), σύμφωνα με την εργασία των Retsinas, G. et al. [Ret+20].
- **Ο συνολικός αριθμός των training samples.** Εδώ, εφόσον το dataset που έχουμε δεν μας περιορίζει, πήραμε το μέγιστο αριθμό από samples από όλα τα datasets του UCR archive [Dau+19], αφού πάνω σε αυτό συγκρίνονται οι μέθοδοι που εφαρμόσαμε.

Τελικά καταλήξαμε σε 92.131 raw samples του ενός λεπτού. Αυτά προέκυψαν με τέτοιο τρόπο, ώστε να έχουμε μη-κενά δεδομένα και για τους 3 άξονες (acc, gyr, hrm) και για τα 3 labels (sleep, step, id). Επειδή το smartwatch δίνει τις τιμές του sleeping/walking σε διαστήματα δεν είναι σίγουρο ότι στο διάστημα ενός λεπτού που ορίσαμε θα έχουμε ένα συγκεκριμένο label. Οπότε κάνουμε τους εξής συμβιβασμούς:

- Αν ο χρήστης φαίνεται να είναι awake ανάμεσα σε δύο sleeping διαστήματα για χρόνο μικρότερο από 2 λεπτά, τότε συνενώνουμε τα 3 διαστήματα [sleeping, awake, sleeping] και θέτουμε ότι ο χρήστης είναι σε κατάσταση sleeping για όλο το διάστημα.

- Αν ο χρήστης φαίνεται να είναι σε κατάσταση not walking ανάμεσα σε δύο walking διαστήματα για χρόνο μικρότερο από 10 δευτερόλεπτα, τότε συνενώνουμε τα 3 διαστήματα [walking, not walking, walking] και θέτουμε ότι ο χρήστης είναι σε κατάσταση walking για όλο το διάστημα.
- Τέλος, αν για το κάθε διάστημα ενός λεπτού που εξετάζουμε ο χρήστης αλλάζει από sleeping/walking σε κατάσταση awake/not walking αντίστοιχα τότε θέτουμε κατάσταση transition. Διαφορετικά είναι σε μοναδική κατάσταση sleep-awake/walking-not walking.

Στο Σχήμα 6.3.1 φαίνεται ένα τυχαίο standardized sample του ενός λεπτού από κάθε πιθανή κλάση (sleep/awake/transition) για το task του ύπνου, για κάθε σένσορα. Βλέπουμε ότι τα πλάτη είναι μεγαλύτερα όταν ο χρήστης είναι awake, σε σύγκριση με όταν κοιμάται, ενώ, όταν ο χρήστης είναι σε transition βλέπουμε κάτι ενδιάμεσο. Η κατανομή των κλάσεων στο training set φαίνεται στο Σχήμα 6.3.2. Προφανώς το transition έχει πολύ λίγα δείγματα αφού πρέπει να τύχει μέσα σε συγκεκριμένο διάστημα ενός λεπτού ο χρήστης να μεταβεί από μια κατάσταση σε μία άλλη.

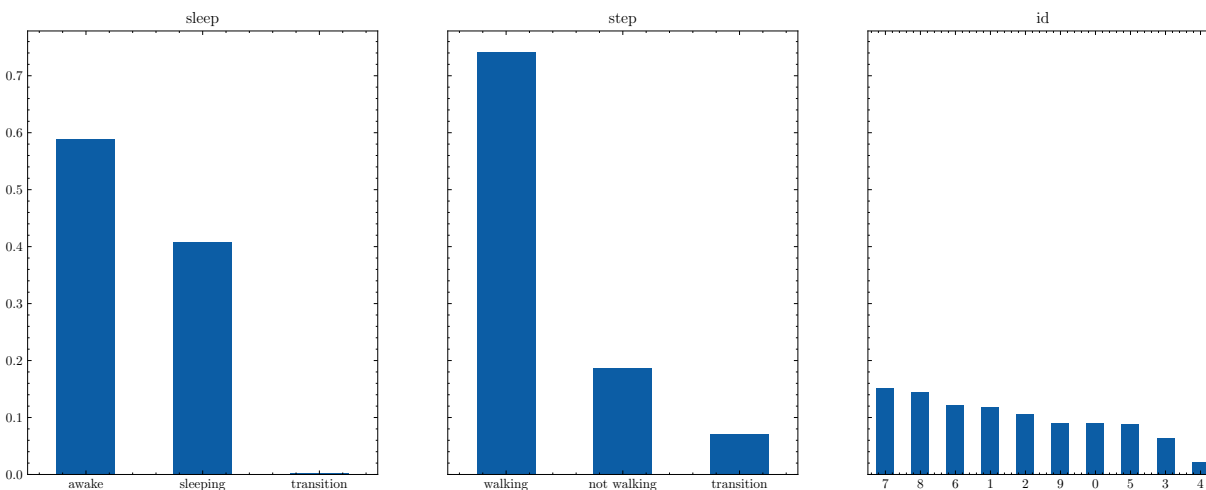


Σχήμα 6.3.1: Οπτικοποίηση standardized δεδομένων και οι αντίστοιχες ετικέτες για το **Sleep task**. Σειρές: Αξελερόμετρο, Γυροσκόπιο, Καρδιακοί παλμοί. Στήλες: Sleeping, awake, transition. Βλέπουμε ότι τα πλάτη είναι μεγαλύτερα όταν ο χρήστης είναι awake, σε σύγκριση με όταν κοιμάται, ενώ όταν ο χρήστης είναι σε transition βλέπουμε κάτι ενδιάμεσο.

6.3.1 Μεθοδολογία

Για κάθε έναν από τους τρεις sensors και κάθε ένα από τα τρία tasks εφαρμόσαμε το linear evaluation πρωτόκολλο [Zer+20; Hao21] που εφαρμόζεται για την αξιολόγηση self-supervised μεθόδων:

- Από τα 92.131 samples κρατήσαμε ένα 10%(= 9.213 δείγματα) για test set στο οποίο δεν γίνεται ούτε supervised training, **αλλά ούτε self-supervised pretraining**
- Στο υπόλοιπο 90%(= 82.918 δείγματα) δοκιμάσαμε τις εξής τεχνικές οι οποίες διαλέχτηκαν με την προϋπόθεση να έχουν μία pretraining φάση στην οποία δεν βλέπουν καθόλου labels:
 - **Hand-crafted:** Μέσω του πακέτου tsfresh [Chr+18], όπου ειδικεύεται στο automatic time series feature extraction και selection. Παράγει δηλαδή περίπου 7700 time series features, και από αυτά επιλέγονται τα πιο σχετικά με το task με βάση τον αλγόριθμο fresh [CKF17]. Το βήμα του selection



Σχήμα 6.3.2: Οπτικοποίηση της κατανομής των κατηγοριών για κάθε τελική εργασία. Προφανώς το transition έχει πολύ λίγα δείγματα αφού πρέπει να τύχει μέσα σε συγκεκριμένο διάστημα ενός λεπτού ο χρήστης να μεταβεί από μια κατάσταση σε μία άλλη.

δεν είναι unsupervised, αλλά είναι προαιρετικό και επίσης μπορεί να γίνει σε μικρό υποσύνολο του training set.

- **Random Projections:** Rocket και MiniRocket [DPW20; DSW21] τα οποία πετυχαίνουν state-of-the-art απόδοση χρησιμοποιώντας πολλούς (> 10.000) τυχαίους convolution kernels (τυχαίοι ως προς το μήκος, dilation, padding, βάρη και bias). Επίσης, αντί του global max pooling χρησιμοποιεί proportion of positive values (ppv), όπως περιγράφεται στην Παράγραφο 5.3.
- **Self-supervised:** SelfTime [Hao21] και TSBert [Zer+20] τα οποία περιγράφονται στις παραγράφους 5.5 και 5.6.
- Κάναμε linear evaluation (linear probing) στα embeddings των προηγούμενων μεθόδων. Δηλαδή, για όλο το training set, χρησιμοποιήσαμε τα embeddings των unsupervised μεθόδων ως είσοδο για να εκπαιδύσουμε έναν γραμμικό ταξινομητή.
- Τέλος, εκπαιδύσαμε ένα InceptionTime [Ism+20] (state-of-the-art μοντέλο στο time-series classification που περιγράφεται στην Παράγραφο 5.4.2) για να συγκρίνουμε την απόδοση με το supervised training. Γενικά περιμένουμε το supervised learning να έχει καλύτερη απόδοση αφού βλέπει όλα τα labels κατά το training, σε αντίθεση με τα υπόλοιπα μοντέλα, όπου τα labels τα βλέπει μόνο ο τελικός linear classifier.

Στον Πίνακα 6.1 φαίνονται τα συγκεντρωτικά αποτελέσματα ως προς το accuracy για κάθε task, κάθε sensor και κάθε μοντέλο. Επειδή ο αριθμός των πειραμάτων που τρέξαμε είναι αρκετά μεγάλος μπορούμε να εξετάσουμε αν κάποια από τις διαισθητικές παρατηρήσεις είναι στατιστικά σημαντικές. Στο Σχήμα 6.3.3 φαίνονται τα αποτελέσματα του πακέτου ttest_ind που έχει μηδενική υπόθεση ότι 2 ανεξάρτητα δείγματα έχουν ταυτόσημες μέσες τιμές. Κάποιες συνολικές παρατηρήσεις είναι:

- **Ως προς τα μοντέλα:**
 - Το MiniRocket πετυχαίνει την καλύτερη απόδοση σε 5 από τους 9 συνδυασμούς (sensor, task), χωρίς να έχει δει labels κατά το pretrain. Στη συνέχεια το SelfTime πετυχαίνει σε 3 στα 9 tasks την καλύτερη απόδοση, χωρίς επίσης να έχει δει labels κατά το pretrain.
 - Τη μικρότερη απόκλιση ως προς το μέσο accuracy έχει το MiniRocket, που σημαίνει ότι είναι το πιο robust μοντέλο σε διαφορετικούς συνδυασμούς sensor/tasks.
- **Ως προς τα tasks:**
 - Με αύξουσα σειρά δυσκολίας (φθίνουσα ως προς το μέσο accuracy) έχουμε: sleep, step, id. Το

αποτέλεσμα είναι στατιστικά σημαντικό, για κάθε ζευγάρι από tasks με $p < 10^{-4}$.

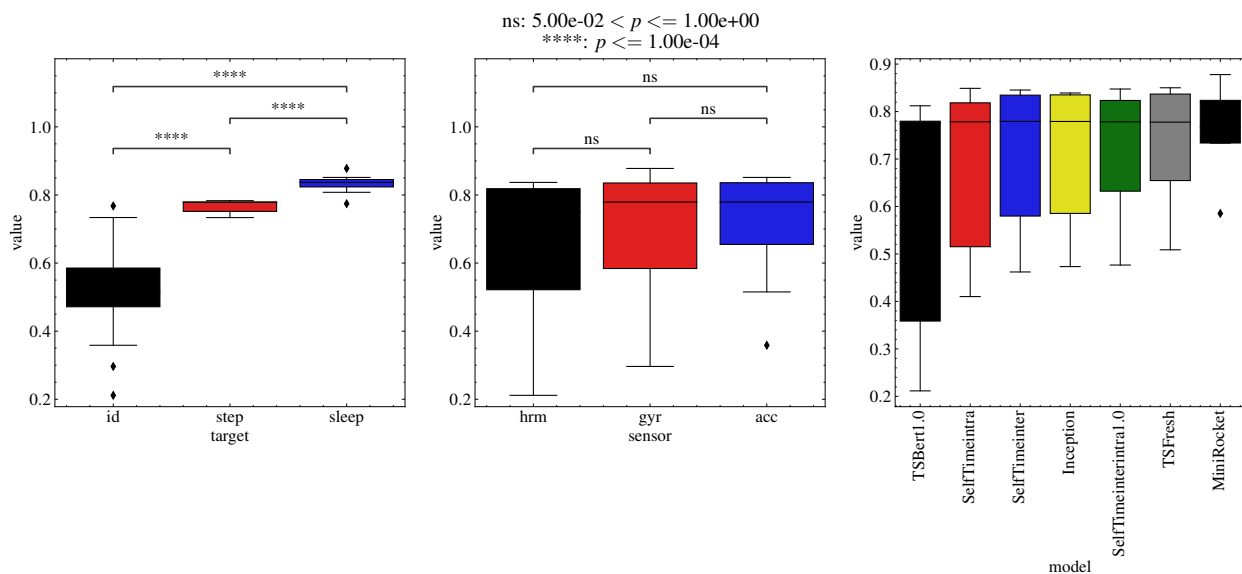
- Το id task έχει το μεγαλύτερο range, που δείχνει ότι συγκεκριμένοι μόνο συνδυασμοί sensor-μοντέλων μπορούν να εξάγουν την πληροφορία που απαιτεί το συγκεκριμένο task. Όντως, βλέποντας τα αποτελέσματα του Πίνακα 6.1, βλέπουμε ότι οι το hrm μάλλον δεν διαθέτει αρκετή πληροφορία για το συγκεκριμένο task, αφού έχει μέγιστο accuracy = 0.59.

- Ως προς τους sensors:

- Έχουμε με αυξουσα σειρά acc, gyr, hrm ως προς το μέσο accuracy. Το αποτέλεσμα δεν είναι, όμως, στατιστικά σημαντικό.
- Η ίδια σειρά ισχύει και για τα ranges του accuracy. Οπότε το αξελερόμετρο περιέχει πληροφορία χρήσιμη και για τα τρία tasks.

Sensor Target	acc			gyr			hrm		
	id	sleep	step	id	sleep	step	id	sleep	step
Model									
Inception	0.585	0.839	0.782	0.525	0.835	0.779	0.473	0.836	0.747
MiniRocket	0.734	0.852	0.779	0.768	0.878	0.746	0.585	0.824	0.734
SelfTimeinter	0.580	0.840	0.779	0.462	0.845	0.783	0.522	0.835	0.753
SelfTimeinterintra1.0	0.632	0.839	0.783	0.477	0.847	0.778	0.478	0.823	0.754
SelfTimeintra	0.515	0.836	0.780	0.472	0.849	0.778	0.410	0.818	0.748
TSBert1.0	0.359	0.812	0.777	0.296	0.808	0.780	0.212	0.774	0.741
TSFresh	0.655	0.838	0.778	0.584	0.850	0.781	0.509	0.837	0.752

Πίνακας 6.1: Accuracy για κάθε task, κάθε sensor και κάθε μοντέλο. Επίσης φαίνονται με **bold** το ελάχιστο και μέγιστο accuracy για κάθε sensor και task. Επίσης βλέπουμε με **πράσινο** χρώμα το μέγιστο accuracy για κάθε task και με **κόκκινο** χρώμα το ελάχιστο. Στην περίπτωση του step task έχουμε ισοβαθμία στο μέγιστο.



Σχήμα 6.3.3: Boxplots που προκύπτουν από όλα τα πειράματα που τρέξαμε. Στον άξονα y απεικονίζεται το accuracy. Για κάθε ζευγάρι μεταβλητών αναγράφεται αν το αποτέλεσμα που φαίνεται στο σχήμα είναι στατιστικά σημαντικό σύμφωνα με το ttest_ind.

6.3.2 Αποτελέσματα

Έστω ότι, όπως υποθέσαμε και στην παράγραφο 6.1.2 έχουμε όλα τα δεδομένα όπως περιγράφηκαν, αλλά καθόλου labels. Τότε, μπορούμε να κάνουμε pretrain τις 4 μεθόδους που αναφέρονται στην παράγραφο 6.3.1. Έστω τώρα, ότι μετά το pretrain, αποκτήσαμε labels για το αν ο χρήστης κοιμάται, είναι ξύπνιος ή βρίσκειται

σε transition σε κάθε ένα από τα διαστήματα ενός λεπτού του dataset. Συγκεκριμένα, θέλουμε να εξετάσουμε:

1. Ποια μέθοδος πετυχαίνει καλύτερα αποτελέσματα με τα λιγότερα διαθέσιμα labels.
2. Κατά πόσο βελτιώνεται η κάθε μέθοδος όταν αυξάνουμε τα **unlabeled** δεδομένα.
3. Ποια μέθοδος πετυχαίνει την καλύτερη απόδοση όταν έχουμε τα περισσότερα δυνατά labels (δηλαδή 92.131 συνολικά train+test).

6.3.3 Αυξάνοντας σταδιακά τη διαθεσιμότητα των labels

Αρχικά, κάνουμε self-supervised-pretrain σε όλο το unlabeled dataset (82.918 δείγματα) τα self-supervised μοντέλα, δηλαδή το SelfTime και το TSBert. Στη συνέχεια, με είσοδο τα embeddings της pretrain φάσης εκπαιδεύουμε έναν γραμμικό ταξινομητή. Στο Σχήμα 6.3.4a (Αριστερά) φαίνονται τα αποτελέσματα, καθώς αυξάνουμε σταδιακά τη διαθεσιμότητα των labels για γραμμική ταξινόμηση στο sleep task. Βλέπουμε στην πράσινη καμπύλη ότι αρκεί 10% των labels (δηλαδή περίπου 8291 δείγματα) για να πετύχουμε αποτέλεσμα που συγκρίνεται με το fully supervised μοντέλο, το οποίο έχει δει όλα τα labels. Πράγματι, στον Πίνακα 6.1 το Inception (fully-supervised μοντέλο) έχει accuracy = 0.83, ίσο με αυτό του SelfTime στον άξονα του γυροσκόπου στο 0.1 του άξονα- x . Επίσης, με μόλις 1% των labels (δηλαδή με μόλις 800 λεπτά labeled δεδομένων) μπορούμε να πετύχουμε ικανοποιητικά αποτελέσματα, με 4%, μόνο, απόκλιση από το μέγιστο accuracy.

Επιπλέον, βλέπουμε ότι το TSBert με το SelfTime έχουν παρόμοια συμπεριφορά καθώς αυξάνουμε τη διαθεσιμότητα των labels. Συγκεκριμένα από το 1% στο 10% διαθεσιμότητα, η απόδοση αυξάνεται περίπου 3 – 4% και στη συνέχεια μένει σταθερή.

Στο Σχήμα 6.3.4b (Αριστερά) φαίνεται η αντίστοιχη καμπύλη, για το step task, καθώς αυξάνουμε την διαθεσιμότητα των labels. Επειδή το task όπως είδαμε είναι πιο δύσκολο έχουμε δύο διαφοροποιήσεις ως προς το sleep task:

1. Πολλά μοντέλα χρειάζονται μεγαλύτερο αριθμό από labels για να φτάσουν το μέγιστο accuracy, για παράδειγμα το SelfTime στον άξονα του hrm (γκρι γραμμή) χρειάζεται 40% των labels έναντι του 10% που είδαμε για το sleep task.
2. Πλέον, η απόκλιση του accuracy από το μέγιστο, όταν έχουμε 1% των labels, δεν είναι της τάξης του 4%, αλλά φτάνει έως και 8% για τη γκρι γραμμή.

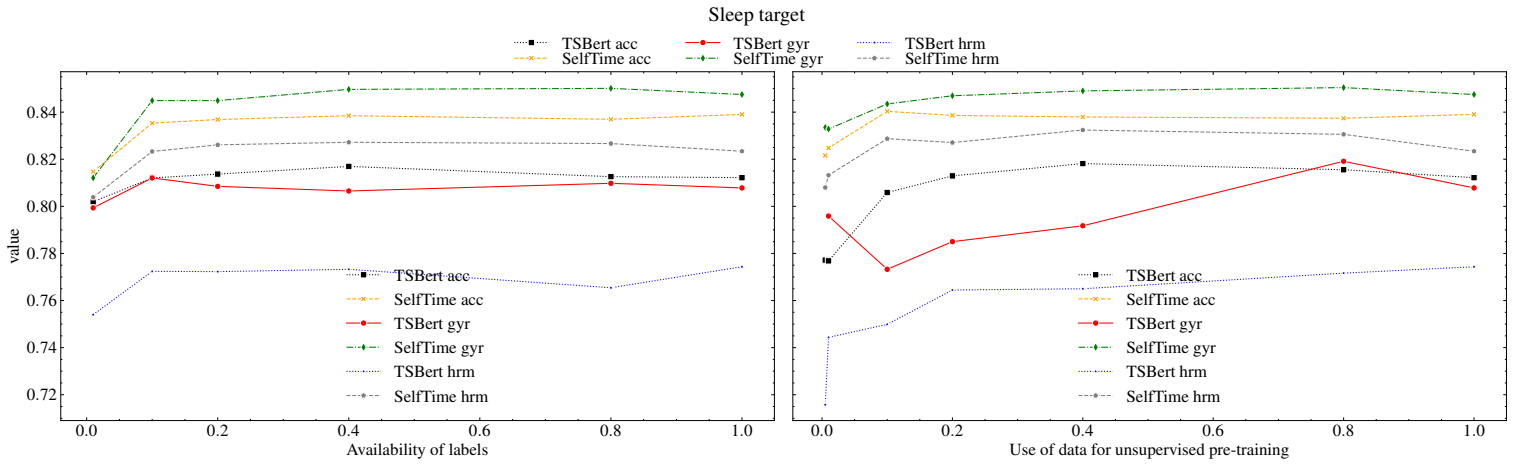
Όμοια, για το id task που είναι ακόμη πιο δύσκολο, έχουμε το Σχήμα 6.3.4c (Αριστερά). Βλέπουμε ότι τα περισσότερα μοντέλα θέλουν 20% των labels (δηλαδή περίπου 16000 δείγματα) για να φτάσουν τη μέγιστη απόδοσή τους. Και πάλι η απόκλιση στο accuracy από το μέγιστο, όταν έχουμε 1% των labels, είναι περίπου 8%. Τέλος, σε αντίθεση με τα προηγούμενα δύο tasks, όταν έχουμε 1% των labels (δηλαδή 800 λεπτά labeled δεδομένων) πετυχαίνουμε απόδοση κοντά στο random που είναι 0.1 για τις 10 κλάσεις του προβλήματος.

6.3.4 Αυξάνοντας σταδιακά τη διαθεσιμότητά των unlabeled δεδομένων για pretrain

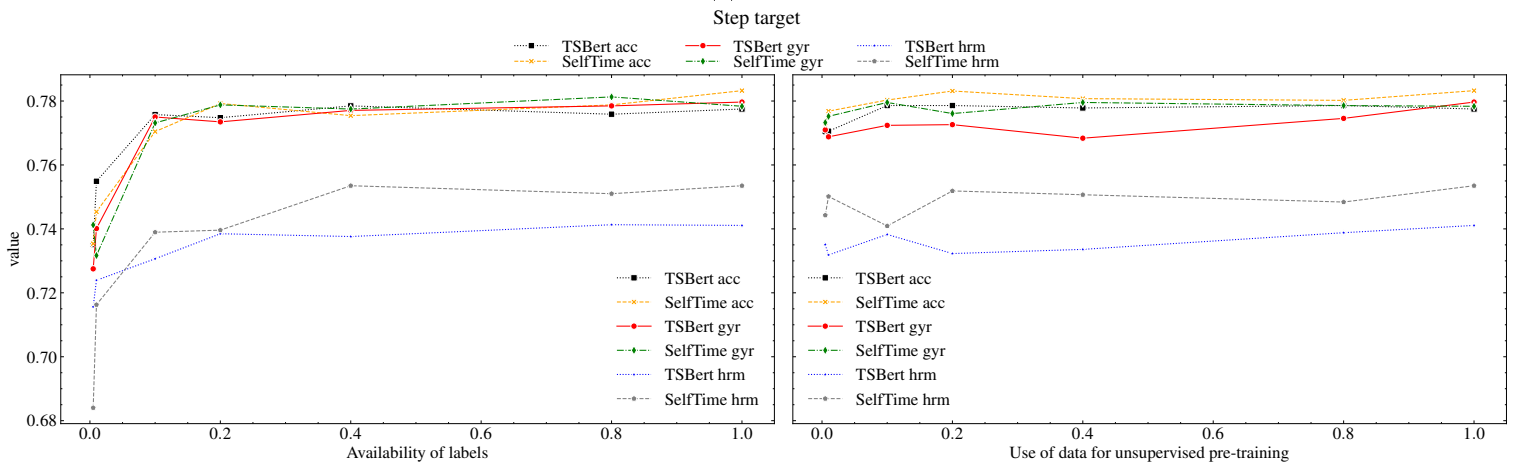
Στο Σχήμα 6.3.4a (Δεξιά) παρατηρούμε ότι τα περισσότερα μοντέλα φτάνουν την μέγιστη απόδοση τους όταν κάνουν pretrain στο 20% των labels. Επιπλέον, βλέπουμε ότι για το SelfTime οι καμπύλες είναι πιο ομαλές, δηλαδή το μοντέλο είναι πιο robust στα λίγα unlabeled δεδομένα. Αντίθετα η απόδοση του TSBert πέφτει σημαντικά (−10%) για λίγα (= 1%) unlabeled δεδομένα. Αυτό, μάλλον οφείλεται στο μέγεθος των δικτύων, αφού το SelfTime αποτελείται από μόλις 4 Conv1D στρώματα, ενώ το TSBert έχει αρχιτεκτονική Transformer με πολύ περισσότερες παραμέτρους.

Όμοια, στο Σχήμα 6.3.4b (Δεξιά) φαίνεται η αντίστοιχη καμπύλη στην οποία αυξάνουμε τα unlabeled δεδομένα για το step task. Βλέπουμε ότι, επειδή το πρόβλημα είναι σχετικά εύκολο, το 10% των unlabeled δεδομένων αρκούν για να φτάσουμε την μέγιστη ακρίβεια. Επίσης, σε σχέση με το sleep task, το TSBert είναι πιο robust για το step στα λίγα δεδομένα.

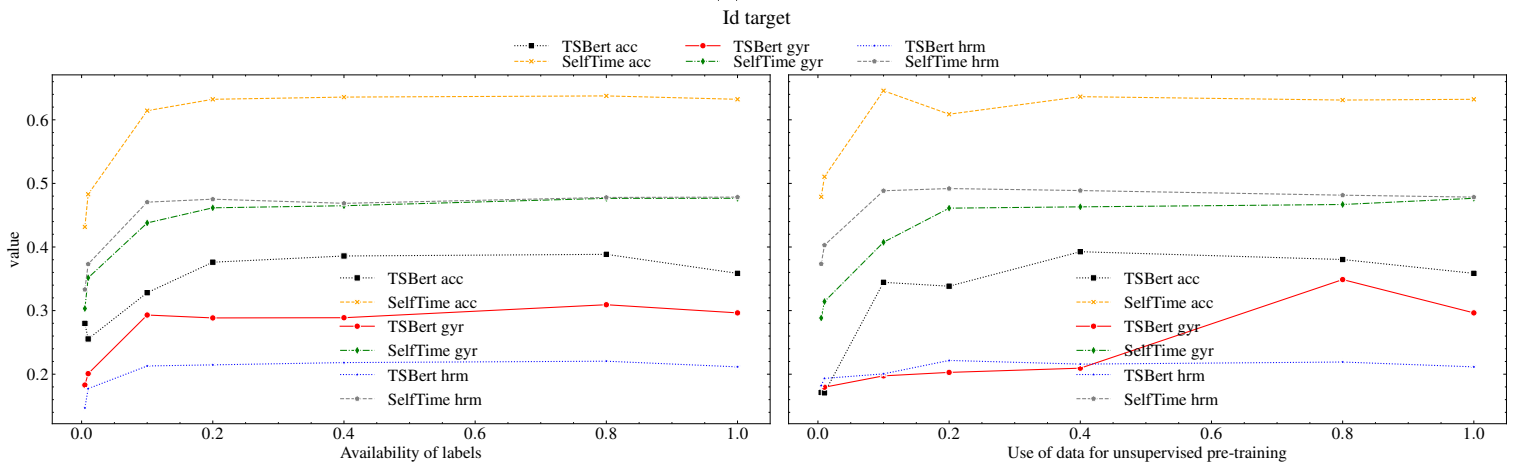
Στο Σχήμα 6.3.4c (Δεξιά) βλέπουμε, ότι αφού το task είναι αρκετά δυσκολότερο, ακόμη και το Self_time που στα δύο πρώτα task συνέκλινε πολύ γρήγορα, τώρα συγκλίνει μετά το 10% των labels. Τέλος, βλέπουμε ότι για



(a) Sleep task.



(b) Step task.



(c) Identification task.

Σχήμα 6.3.4: **Αριστερά:** Αυξάνοντας σταδιακά τη διαθεσιμότητα των labels. **Δεξιά:** Αυξάνοντας σταδιακά τη διαθεσιμότητα των unlabeled δεδομένων για pretrain.

το TSBert οι καμπύλες ξεκινούν από το random για πολύ λίγα (1%) unlabeled δεδομένα, ενώ στο SelfTime από πιο ψηλά. Άρα και πάλι το SelfTime είναι πιο robust στα λίγα unlabeled δεδομένα.

6.3.5 Αποτελέσματα όταν έχουμε fully-labeled dataset

Σύγκριση μοντέλων ανά task

Sleep task:

Στο Σχήμα 6.3.6a φαίνονται αναλυτικά οι επιδόσεις των μοντέλων καθώς και το global step που έκαναν early stopping για τους 3 sensors. Μερικές παρατηρήσεις είναι:

1. Το γυροσκοπιο φαίνεται να έχει περισσότερη πληροφορία για το πρόβλημα του ύπνου, αφού 5 από τα 7 μοντέλα πετυχαίνουν καλύτερες επιδόσεις, όταν εκπαιδεύονται στα δεδομένα αυτού του sensor. Αυτό συμπίπτει με τα αποτελέσματα των Retsinas, G. et al. [Ret+20].
2. Το MiniRocket φαίνεται να είναι με διαφορά το καλύτερο μοντέλο για τους άξονες του γυροσκοπίου και αξελερόμετρου με accuracies 87,8% και 85.2% αντίστοιχα. Επίσης, αποδίδει αρκετά καλά και στον άξονα του HRM με accuracy 82.4%. Τέλος, τη χειρότερη απόδοση την βλέπουμε από το μοντέλο TSBert εκπαιδευμένο στον άξονα του HRM με accuracy 77.4%.
3. Τα embeddings του TSBert φαίνεται να μην δουλεύουν σε αυτό το task σε αντίθεση με αυτά του SelfTime. Μια βασική διαφορά του TSBert με το SelfTime είναι ότι το πρώτο προβλέπει τεχνητά gaps μέσα σε ένα sample χρονοσειράς, χωρίς να συγκρίνει ξένα samples. Άρα, η χαμηλή του απόδοση, μας δείχνει ότι η πρόβλεψη σχέσεων μέσα στο δείγμα (intra relationship) δεν αρκεί για το sleep task. Παίζει, δηλαδή σημαντικό ρόλο η σύγκριση περιόδων όπου ο χρήστης κοιμάται με αυτές που είναι ξύπνιος (inter-sample relationship).
4. Το SelfTime και στις τρεις παραλλαγές του πετυχαίνει κοντινή απόδοση με το fully supervised μοντέλο (InceptionTime). Επιπλέον, μπορούμε να δούμε στον άξονα του HRM, ότι η inter παραλλαγή νικάει κατά πολύ την intra πράγμα που επιβεβαιώνει το 3., δηλαδή η inter σχέση είναι σημαντική στο συγκεκριμένο task. Πράγματι, στον ίδιο άξονα το TSBert που δεν αξιοποιεί αυτή τη σχέση πετυχαίνει τα χειρότερα αποτελέσματα.

Step task:

Παρόμοια, στο Σχήμα 6.3.6b φαίνονται οι επιδόσεις των μοντέλων το step task. Οι παρατηρήσεις εδώ είναι οι εξής:

1. Τα kinetic δεδομένα, δηλαδή το γυροσκοπιο και το αξελερόμετρο έχουν παρόμοιες επιδόσεις, με ελάχιστο accuracy για όλα τα μοντέλα στους δύο άξονες 74.6% (μοντέλο MiniRocket στον άξονα του γυροσκοπίου). Αντίθετα, το HRM σε αυτό το task είναι σημαντικά χειρότερο με μέγιστο accuracy 75.4%, δηλαδή μόλις 0.8% υψηλότερο από το ελάχιστο των δύο kinetic αξόνων. Και πάλι τα αποτελέσματα συμπίπτουν με το [Ret+20].
2. Το MiniRocket, ενώ στο πρόβλημα του ύπνου πετυχαίνει εξαιρετικές επιδόσεις, σε αυτό το task βγαίνει τελευταίο σε accuracy στους άξονες του γυροσκοπίου και του HRM με τιμές 74.6% και 74.4% αντίστοιχα.
3. Το SelfTime πετυχαίνει τις καλύτερες επιδόσεις για το συγκεκριμένο task για όλους τους sensors. Μάλιστα, δεν υπάρχει διαφοροποίηση από την intra στην inter εκδοχή του, όσο αφορά την απόδοση. Αυτό σημαίνει ότι και των δύο ειδών σχέσεις είναι εξίσου σημαντικές στο task αυτό ή ισοδύναμα είναι σημαντική η πρόβλεψη gaps μέσα στο ίδιο δείγμα χρονοσειράς (intra-σχέση).
4. Το TSBert πηγαίνει αρκετά καλά σε σύγκριση με το task του ύπνου. Πράγματι, αν συγκρίνει κανείς τα Σχήματα 6.3.6b και 6.3.6a, θα δει ότι στο task του ύπνου το TSBert βρίσκεται με διαφορά κάτω από όλα τα υπόλοιπα μοντέλα, κάτι το οποίο δεν συμβαίνει στο step task. Οπότε, επιβεβαιώνεται το 3., δηλαδή στο πρόβλημα του step η intra σχέση έχει επίσης σημαντική πληροφορία την οποία εκμεταλλεύεται το TSBert.

Id task:

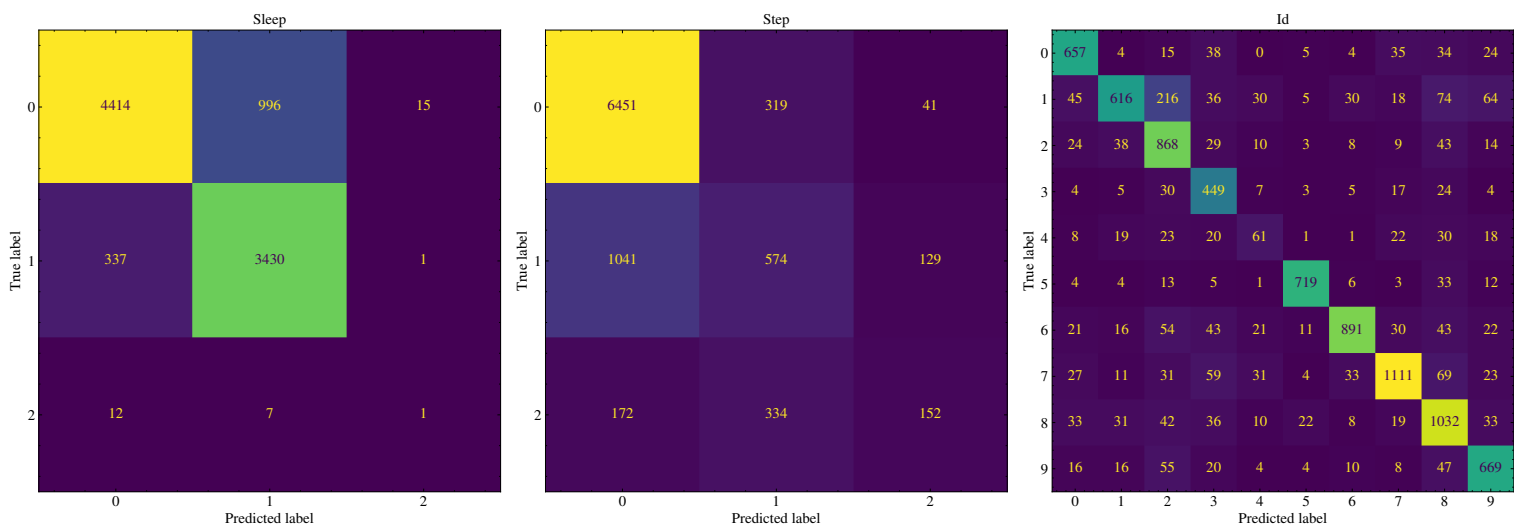
Τέλος, στο Σχήμα 6.3.6c φαίνονται οι επιδόσεις των μοντέλων για το id task. Οι παρατηρήσεις είναι:

1. Σε αυτό το task φαίνεται ότι το αξελερόμετρο περιέχει την περισσότερη πληροφορία, αφού αν κάνουμε την σύγκριση ανά μοντέλο ως προς το accuracy, τότε τα περισσότερα ζευγάρια (πλην του MiniRocket) έχουν μεγαλύτερο accuracy στο αξελερόμετρο.
2. Το MiniRocket είναι και πάλι με διαφορά το καλύτερο μοντέλο και για τους τρεις σένσορες, καθώς πετυχαίνει accuracies 74.4%, 76.8% και 58.5% στους acc, gyi και hrm άξονες αντίστοιχα.
3. Το SelfTime και συγκεκριμένα η παραλλαγή που εξερευνά τις inter σχέσεις πηγαίνει εξαιρετικά καλά. Αυτό είναι διαισθητικά σωστό, αφού γι' αυτό το task παίζει σημαντικό ρόλο η σύγκριση μεταξύ χρονοσειρών, η οποία εξαρτάται άμεσα με το label.
4. Τα embeddings του TSBert φαίνεται να μην δουλεύουν σε αυτό το task σε αντίθεση με αυτά του SelfTime και αυτό φαίνεται στο Σχήμα 6.3.6c, αφού το TSBert είναι με διαφορά πιο κάτω από τα υπόλοιπα μοντέλα. Όπως είδαμε ήδη, το TSBert δε λαμβάνει υπόψιν τις intra σχέσεις, οπότε επιβεβαιώνεται το 3. με τη χαμηλή του απόδοση.

Επίσης, στο Σχήμα 6.3.5 (Αριστερά) φαίνεται το Confusion Matrix του καλύτερου μοντέλου (MiniRocket) για το sleep task. Δεν παρατηρούμε κάποιο σφάλμα, για παράδειγμα λάθος αναθέσεις στην κλάση πλειοψηφίας (awake). Στη συνέχεια στο μεσαίο πλαίσιο φαίνεται το Confusion Matrix του καλύτερου μοντέλου (SelfTimeinter) για το step task. Εδώ βλέπουμε ότι ανατίθενται πολλά instances στην κλάση πλειοψηφίας. Τέλος, στο δεξιά πλαίσιο βλέπουμε το καλύτερο μοντέλο (MiniRocket) για το id task, χωρίς και εδώ να γίνεται κάποιο συστηματικό σφάλμα.

Οπτικοποίηση των pretrained-embeddings

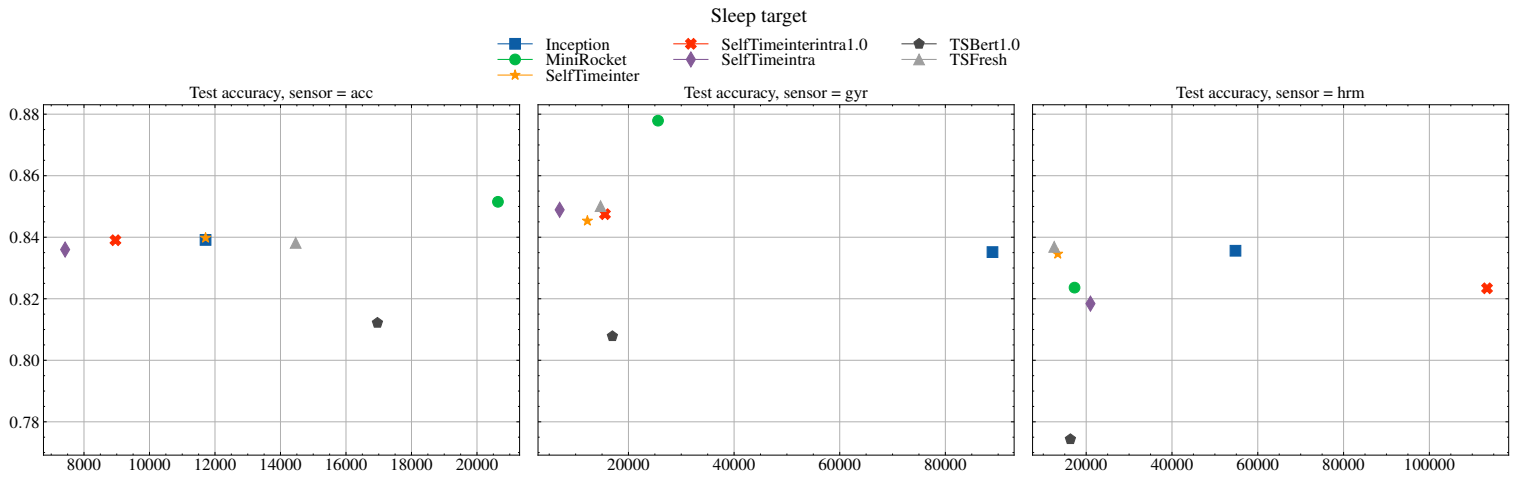
Τέλος, στα Σχήματα 6.3.8, 6.3.7a και 6.3.7b φαίνονται οπτικοποιήσεις τεσσάρων τεχνικών dimensionality reduction, συγκεκριμένα PCA, t-SNE [MH08], UMAP και densMAP [MHM18] στα embeddings των καλύτερων μοντέλων για κάθε task αντίστοιχα. Συγκεκριμένα του SelfTime για τα **step** και **identification** tasks και το MiniRocket για το **sleep** task. Είναι εντυπωσιακό ότι χωρίς να έχει δει καθόλου labels το μοντέλο, μπορεί να ξεχωρίζει τόσο καλά τις κλάσεις για όλα τα tasks.



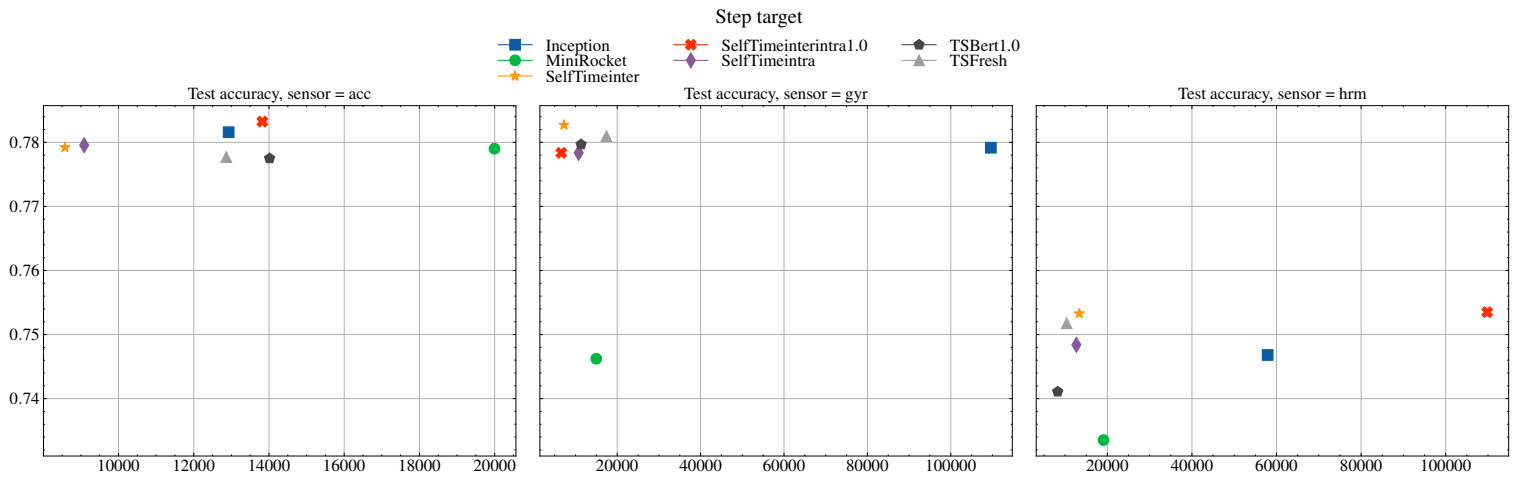
Σχήμα 6.3.5: Confusion Matrix του καλύτερου μοντέλου (MiniRocket) για τα 3 tasks. **Sleep:** 0:awake, 1:sleeping, 2:transition, **Step:** 0:walking, 1:not-walking, 2:transition.

6.4 Survival Analysis

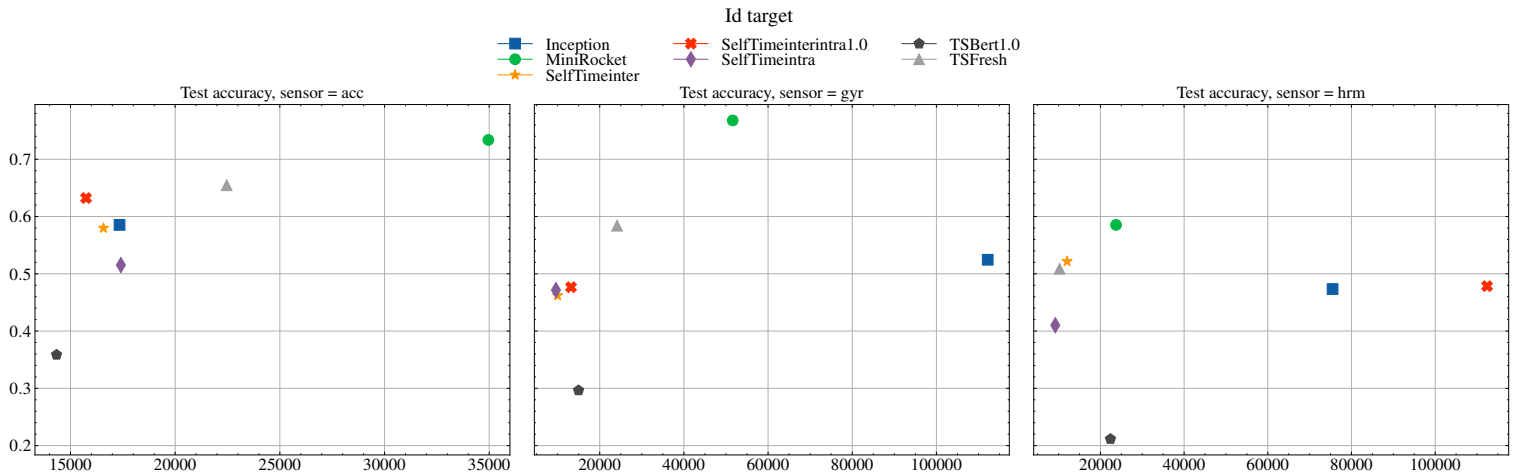
Από τους 64 συμμετέχοντες στο πρόγραμμα, οι 38 είναι patients. Στους 21 από τους patients, έχουμε δεδομένα για το χρονικό διάστημα στο οποίο βρισκόταν σε υποτροπή και συνολικά έχουμε 37 διαστήματα υποτροπής. Στόχος αυτού του πειράματος είναι: Δεδομένων των μετρήσεων μίας ημέρας από το smartwatch, να προβλέψει το χρονικό διάστημα μέχρι την έναρξη της υποτροπής.



(a) Sleep task.

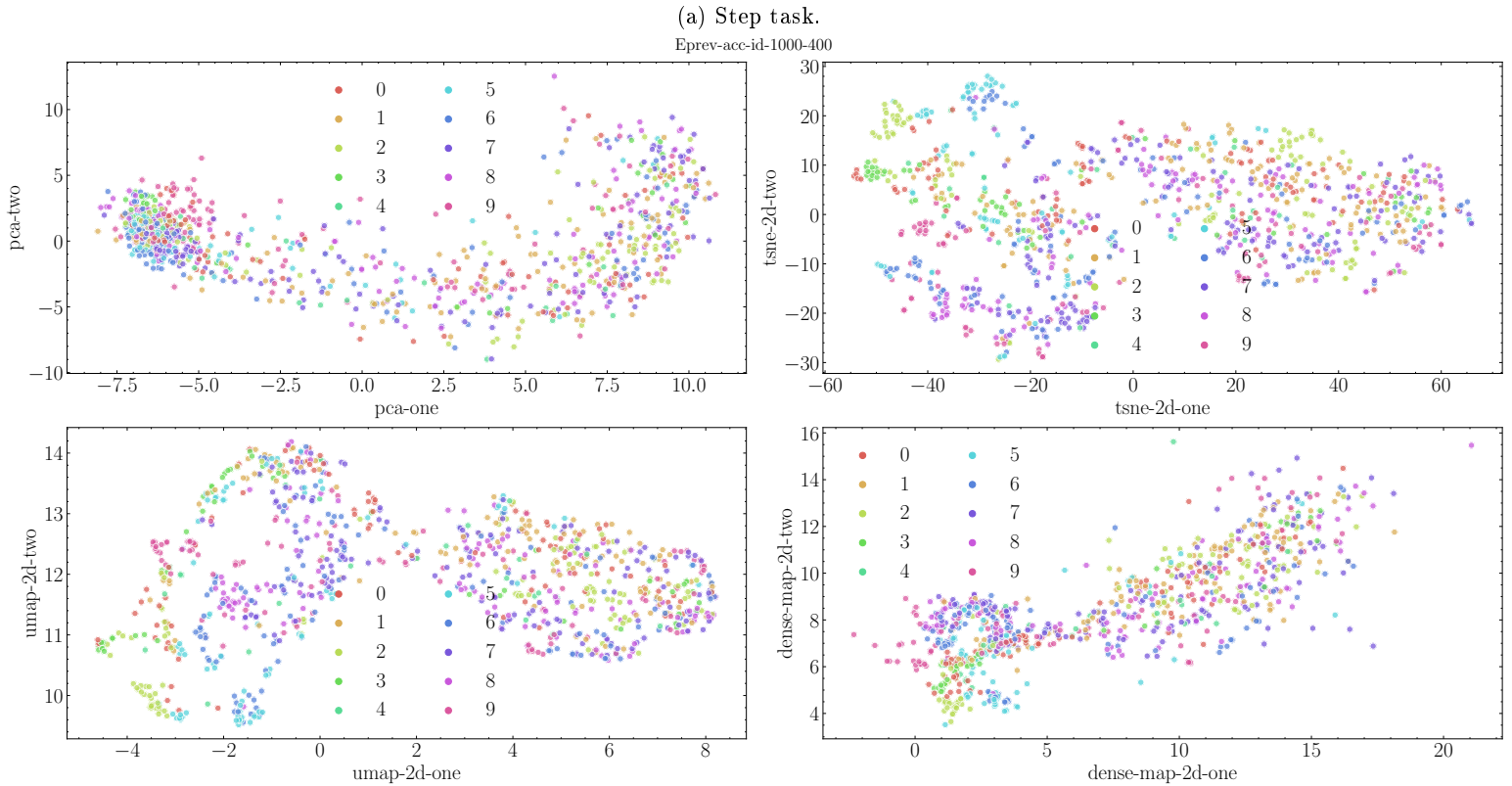
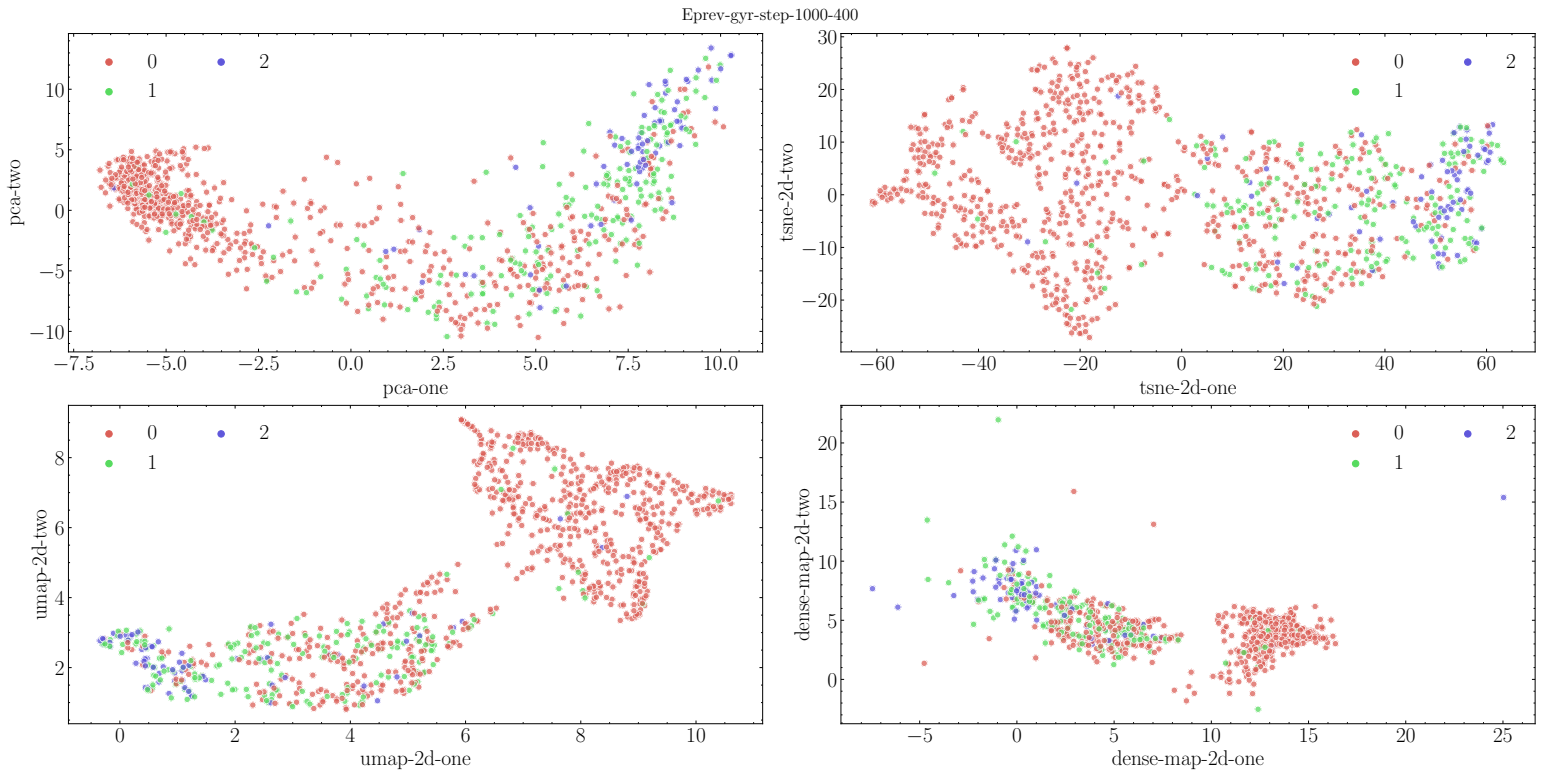


(b) Step task.

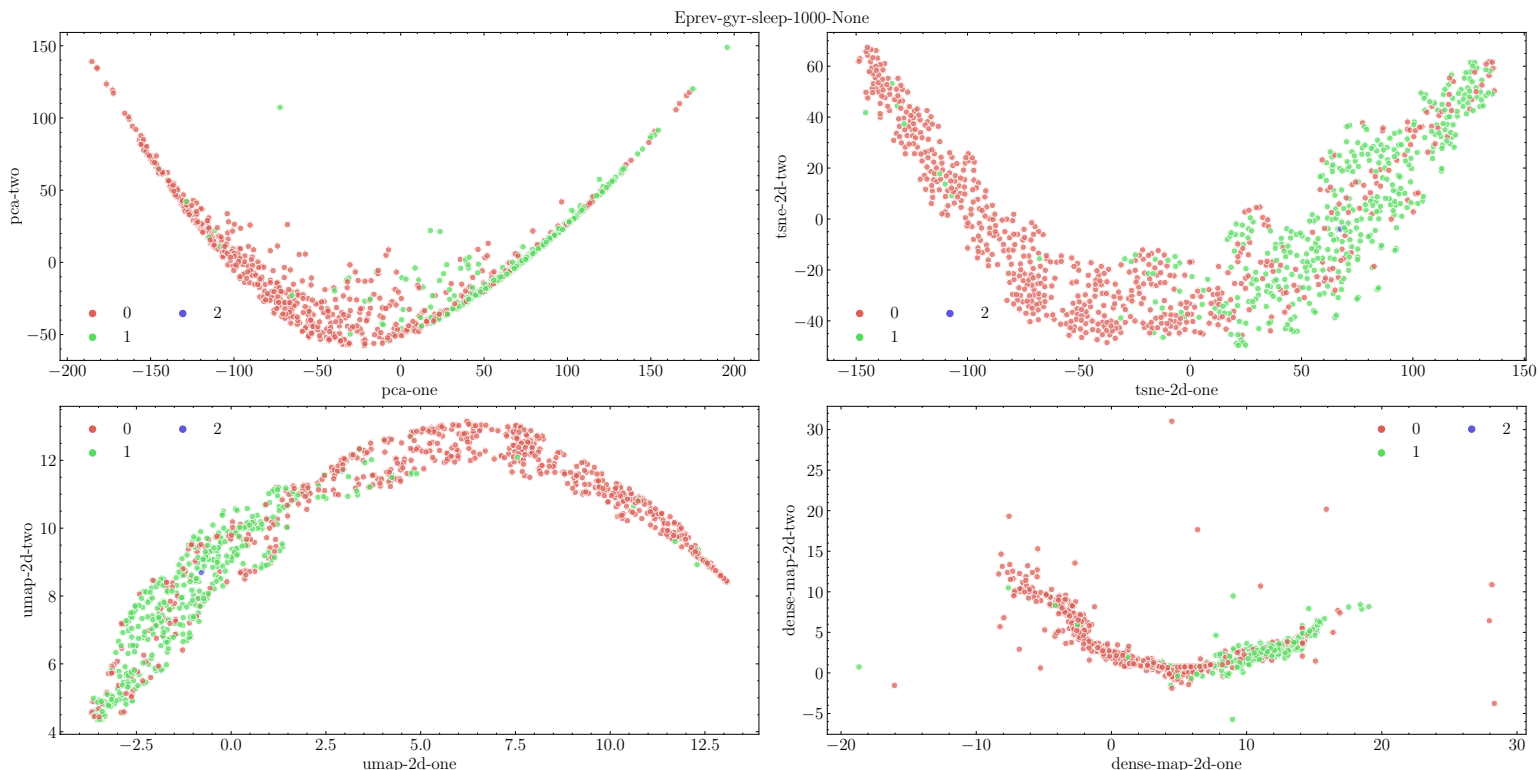


(c) Identification task.

Σχήμα 6.3.6: Αποτελέσματα όταν έχουμε όλα τα διαθέσιμα labels. Στον άξονα x φαίνεται σε ποιο global step έκανε early stopping ο γραμμικός ταξινομητής και στον άξονα y το αντίστοιχο accuracy.



Σχήμα 6.3.7: Εφαρμογή 4 τεχνικών dimensionality reduction, συγκεκριμένα PCA, t-SNE, UMAP και densMAP στα embeddings του καλύτερου μοντέλου (SelfTime) στα **step** και **id** tasks.



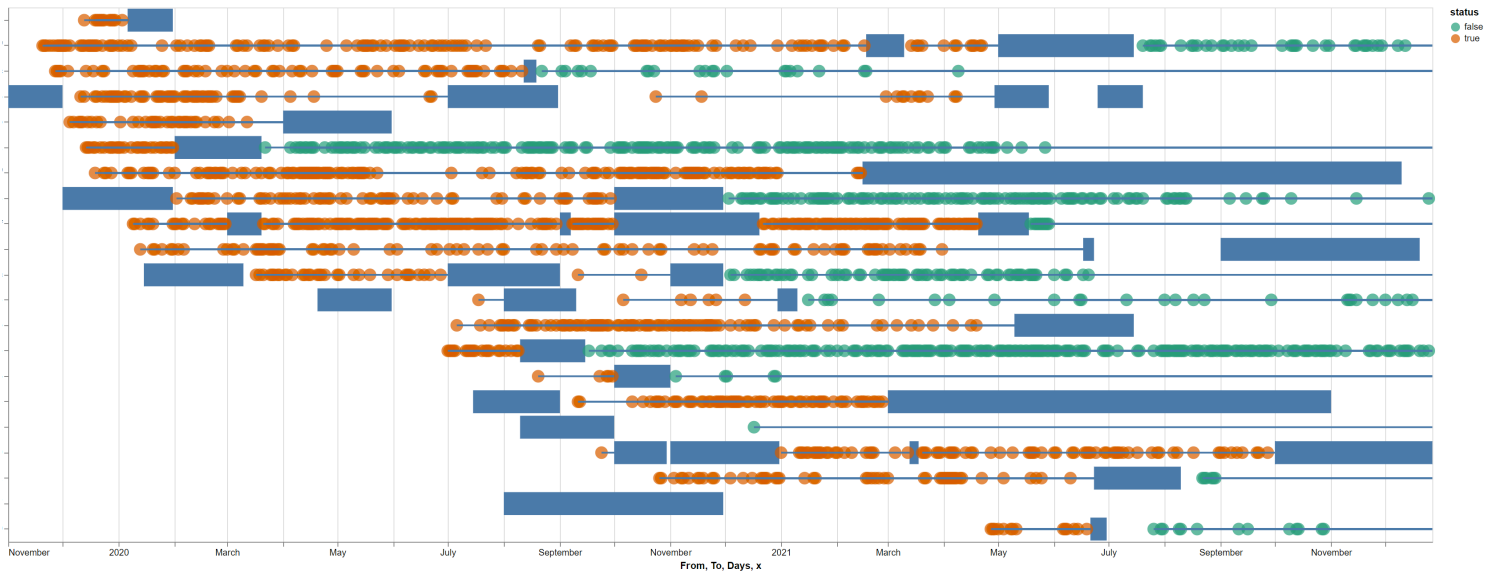
Σχήμα 6.3.8: Εφαρμογή 4 τεχνικών dimensionality reduction, συγκεκριμένα PCA, t-SNE, UMAP και densMAP στα embeddings του καλύτερου μοντέλου για το **sleep** task (MiniRocket).

Όπως και στην Παράγραφο 6.3, πρέπει να πάρουμε κάποιες αποφάσεις, σχετικά με τη δημιουργία του dataset. Συγκεκριμένα, οι αποφάσεις που θα πάρουμε είναι ίδιες με το αν προσεγγίζαμε το πρόβλημα σαν anomaly detection task:

- **Η επιλογή του χρονικού διαστήματος που θα έχει ένα training sample.** Το διάστημα για να μπορούμε να αξιολογήσουμε αν ο χρήστης βρίσκεται κοντά σε relapse, πρέπει να είναι αρκετά μεγάλο, έτσι ώστε το μοντέλο να μπορεί να εντοπίσει “ασυνήθιστες” συμπεριφορές σε σύγκριση με τις “κανονικές” συμπεριφορές. Από την άλλη όσο μεγαλώνει το διάστημα, τόσο πιο πιθανό είναι να έχουμε gaps, είτε λόγω φόρτισης του ρολογιού είτε σφαλμάτων κατά τη συλλογή των δεδομένων. Δεδομένου, ότι τα μοντέλα που χρησιμοποιούμε εκμεταλλεύονται τη χρονική αλληλουχία των μετρήσεων, το να συνενώναμε ή να κάναμε interpolation μεγάλα κενά θα εισήγαγε θόρυβο. Τέτοιο θόρυβο το μοντέλο μπορεί να τον κατηγοριοποιούσε ως “anomaly”, πράγμα το οποίο δεν το θέλουμε. Τελικά, καταλήξαμε στο διάστημα των 12 ωρών συνεχόμενων μετρήσεων. Τέλος, επειδή σε αυτό το task δεν μας ενδιαφέρουν οι μικροσυμπεριφορές, αλλά η καθημερινότητα του χρήστη και επειδή το διάστημα 12 ωρών είναι πολύ μεγάλο αν δειγματοληπτούμε με την αρχική συχνότητα, κάναμε resample παίρνοντας τον μέσο όρο των μετρήσεων ανά ένα λεπτό.
- **Ο συνολικός αριθμός των training samples.** Εδώ, σε αντίθεση με την Παράγραφο 6.3, ο αριθμός των ατόμων με μετρήσεις είναι περιορισμένος και ο αριθμός των συνεχόμενων 12-ωρων μετρήσεων το ίδιο. Οπότε, κρατάμε όλα τα δείγματα όπου έχουν 12 ώρες συνεχόμενες μετρήσεις, μετά από γραμμικό interpolation 5 δείγματα ως προς κάθε κατεύθυνση (= 10 λεπτά), σε περίπτωση που έχουμε μικρά κενά. Τελικά, προέκυψαν 1909 samples (ή 1909 μέρες 12-ωρων συνεχόμενων μετρήσεων), όπως φαίνονται στο Σχήμα 6.4.1.
- **Εξαγωγή επισημειώσεων.** Η πρώτη ημερομηνία που έχουμε κάποιο γεγονός είναι 2019-11-20, ενώ η ημερομηνία που τελειώνει και το τελευταίο γεγονός είναι 2021-12-26. Στο Σχήμα 6.4.1 φαίνονται τα χρονικά διαστήματα των relapses για κάθε χρήστη. Επίσης, σημειώνονται με κουκκίδες οι ημέρες που έχουμε 12 ώρες συνεχόμενων μετρήσεων, με πορτοκαλί για τις ημέρες που έχουν κάποια υποτροπή δεξιά τους, ενώ με πράσινο διαφορετικά. Θα μπορούσε κανείς να πάρει μόνο τις πορτοκαλί κουκκίδες και να

κάνει regression ως προς το χρονικό διάστημα από την κουκκίδα μέχρι την αρχή του επόμενου συμβάντος. Αυτό είναι λάθος, γιατί πετάμε την χρήσιμη πληροφορία ότι οι πράσινες κουκκίδες είναι για τουλάχιστον t διάστημα χωρίς συμβάν, όπου t το διάστημα από την κουκκίδα μέχρι τις 2021-12-26. Άρα έχουμε right-censored δεδομένα, όπως περιγράφηκε στην Παράγραφο 5.8 και άρα survival-analysis πρόβλημα.

- **Χρονοσειρές εισόδου.** Το πρόβλημα που έχουμε να λύσουμε είναι αρκετά πιο δύσκολο από αυτά της Παραγράφου 6.3, οπότε θα χρειαστούμε και τους 3 sensors συνδυαστικά. Συγκεκριμένα, για κάθε sensor παίρνουμε την ℓ_2 -νόρμα ως προς τους τρεις άξονες, δηλαδή: $\|\text{sensor}_i\|_2 = \sqrt{x^2 + y^2 + z^2}$. Επίσης, διαισθητικά, αν κάποιος ήθελε να περιγράψει την καθημερινότητα του χρήστη, θα έπρεπε με κάποιον τρόπο να αντιστοιχίσει την ώρα της ημέρας με την κάθε του δραστηριότητα. Ευτυχώς, την πληροφορία αυτή την έχουμε από το Timestamp του ρολογιού και μπορούμε να την κάνουμε encode ως cyclical feature. Συγκεκριμένα, αντιστοιχίζουμε κάθε ώρα του Timestamp στο μοναδιαίο κύκλο, σαν να είχαμε ένα ρολόι με 24 (αντί για 12) ώρες. Άρα για να περιγράψουμε την κάθε ώρα ακούν δύο τιμές: $x_{\sin} = \sin\left(\frac{2\pi * \text{hour}}{24}\right)$ και $x_{\cos} = \cos\left(\frac{2\pi * \text{hour}}{24}\right)$.



Σχήμα 6.4.1: Έχουμε συνολικά 1909 samples/κουκκίδες και 37 υποτροπές (μπλε διαστήματα). Με πορτοκαλί χρώμα φαίνονται τα samples όπου έχουν κάποια υποτροπή στα δεξιά τους, ενώ με πράσινο εκείνα που βρίσκονται δεξιά από κάθε υποτροπή του χρήστη.

Ένα στιγμιότυπο του dataset όπως περιγράφηκε, καθώς και το ιστόγραμμα των time-to-event χρόνων για censored/non-censored δείγματα φαίνεται στο Σχήμα 6.4.2. Βλέπουμε ότι η κατανομή των time-to-event χρόνων είναι ίδια στα censored και non-censored δεδομένα, επομένως δεν χρειάζεται να κάνουμε κάποιου είδους stratification, ως προς το censoring κατά το train-test split του dataset.

6.4.1 Μεθοδολογία

Προ-εκπαίδευση

Στόχος αυτού του πειράματος είναι να συγκρίνουμε τα embeddings που προκύπτουν από τους παρακάτω τρόπους ως προς την αποτελεσματικότητά τους στην πρόβλεψη του χρόνου μέχρι την επόμενη υποτροπή:

- **Hand-crafted:** Χρησιμοποιούμε, όπως και στην Παράγραφο 6.3.1 το πακέτο TSFresh [Chr+18] για την παραγωγή Hand-crafted features. Επειδή όπως είδαμε παράγονται συνολικά 7700 features χρονοσειρών και επειδή οι αλγόριθμοι survival-analysis που εφαρμόσαμε δεν τρέχουν στην κάρτα γραφικών και άρα έχουν μεγάλους χρόνους εκπαίδευσης, ήταν σημαντικό να εκτελέσουμε και το βήμα του selection για να μειώσουμε τη διάσταση του χώρου των features. Αυτό έγινε μέσω του αλγόριθμου fresh [CKF17] και συγκεκριμένα μόνο στα uncensored δεδομένα, σαν να είχαμε regression πρόβλημα.
- **Random Projections:** Όπως και στην Παράγραφο 6.3.1 χρησιμοποιήθηκε το μοντέλο

MiniRocket [DSW21] για εξαγωγή χαρακτηριστικών, το οποίο χρησιμοποιεί 10.000 τυχαίους convolution kernels (τυχαίοι ως προς το μήκος, dilation, padding, βάρη και bias).

- **Self-supervised:** Επειδή το SelfTime [Hao21] που εφαρμόσαμε στην Παράγραφο 6.3.1 δεν έχει σχεδιαστεί για multivariate χρονοσειρές, δεν χρησιμοποιήθηκε. Οπότε χρησιμοποιήθηκε μόνο το TS-Bert [Zer+20], το οποίο περιγράφεται στην παράγραφο 5.6. Ειδικότερα, τα self-supervised μοντέλα έχουν το πλεονέκτημα ότι μπορούμε να κάνουμε pretrain και σε δεδομένα για τα οποία δεν έχουμε labels σχετικά με την υποτροπή ή και με δεδομένα των controls. Πράγματι, δοκιμάσαμε να κάνουμε pre-train σε δεδομένα όλων των ασθενών, ανεξάρτητα αν είχαν υποτροπή η όχι και αυτό βελτίωσε την απόδοση. Επιπλέον, δοκιμάσαμε να κάνουμε προεκπαίδευση σε δεδομένα από controls, αλλά δυστυχώς αυτά ήταν πολύ λίγα (3 μήνες μετρήσεων).

Με βάση τα παραπάνω προκύπτουν πέντε είδη embeddings. Ειδικότερα, ορίζουμε ως n τον αριθμό των δειγμάτων εκπαίδευσης και m τον αριθμό των μεταβλητών (6 στο σύνολο, δηλαδή τα: acc, gyr, hrm, RR και sin/cos της ώρας της ημέρας). Επίσης, ορίζουμε num_samples τον αριθμό των χρονικών δειγμάτων που έχει ένα sample (12 ώρες ή 720 samples του ενός λεπτού) και p τη διάσταση των embeddings εξόδου. Οπότε, έχουμε:

- **Hand-crafted embeddings:** Το αρχικό σύνολο δεδομένων εκπαίδευσης, διάστασης $(n, m, \text{num_samples}) = (1909, 6, 720)$, μετατρέπεται σε $(n, p) = (1909, 7700)$, δηλαδή για κάθε MTS εισόδου παράγονται 7700 χαρακτηριστικά. Στη συνέχεια, το βήμα του feature selection μειώνει τη διάσταση αυτή σε $(n, p) = (1909, 898)$.
- **Τα embeddings του minirocket:** Όπου για κάθε MTS εισόδου παράγονται 10000 features μέσω τυχαίων συνελικτικών πυρήνων, οπότε έχουμε $(n, p) = (1909, 10000)$.
- **Τα embeddings του TSBert:** Εδώ έχουμε την ευελιξία να κάνουμε pretrain με τρία διαφορετικά datasets. Συγκεκριμένα έχουμε:
 - Dataset χωρίς τις μέρες των υποτροπών: $(n, m, \text{num_samples}) = (1909, 6, 720)$. Διάσταση εξόδου: $(n, p) = (1909, 256)$.¹ Στο Σχήμα 6.4.3 και στον Πίνακα 6.2, αναφέρεται ως: TSBert.
 - Dataset που περιλαμβάνει τις μέρες των υποτροπών: $(n, m, \text{num_samples}) = (2402, 6, 720)$. Διάσταση εξόδου: $(n, p) = (2402, 256)$. Στο Σχήμα 6.4.3 και στον Πίνακα 6.2, αναφέρεται ως: TSBertAD (All Days).
 - Dataset που περιλαμβάνει τις μέρες των υποτροπών και επιπλέον δεδομένα από τους controls για όλα τα έτη: $(n, m, \text{num_samples}) = (2402+840 = 3242, 6, 720)$. Διάσταση εξόδου: $(n, p) = (3242, 256)$. Στο Σχήμα 6.4.3 και στον Πίνακα 6.2, αναφέρεται ως: TSBertADC (All Days + Controls). Όπως βλέπουμε τα δεδομένα των controls είναι, δυστυχώς, λίγα, δηλαδή 720 samples των 12 ωρών.

Εκπαίδευση τελικών survival-models

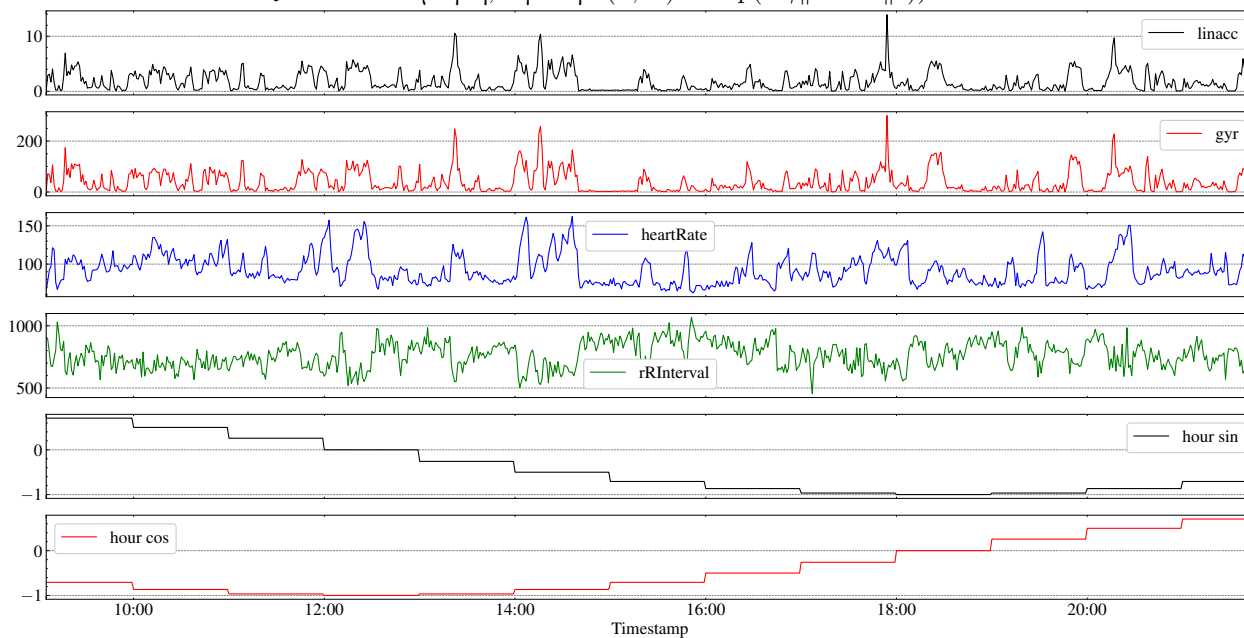
Αφού κάνουμε προ-εκπαίδευση τα embeddings των παραπάνω μεθόδων, μπορούμε να εκπαιδύσουμε με είσοδο αυτά τα εξής survival-analysis μοντέλα:

- **Penalized Cox Model** και συγκεκριμένα **Elastic Net:** Ο αριθμός των covariates p είναι πολύ μεγάλος σε σχέση με τον αριθμό των samples n για μερικά μοντέλα (για παράδειγμα MiniRocket: $p = 10000, n = 2321$). Οπότε, θα ήταν λάθος να χρησιμοποιήσουμε μοντέλο χωρίς regularization. Γι'αυτό το λόγο επιλέχθηκε το Elastic Net που είναι συνδυασμός των LASSO και Ridge και άρα μπορούμε να κάνουμε grid-search στο κατά πόσο θα χρησιμοποιήσουμε το ℓ_1 σε σχέση με το ℓ_2 -penalty, ώστε να μεγιστοποιήσουμε την απόδοση.
- **Random Survival Forest** και **Gradient Boosting Survival Model:** Δύο παρόμοια μοντέλα που βασίζονται σε δέντρα αποφάσεων και άρα μπορούν να μάθουν μη γραμμικές εξαρτήσεις στα δεδομένα μας. Αν η απόδοσή τους σε σχέση με το γραμμικό Cox είναι κατά πολύ καλύτερη, τότε μπορούμε να συμπεράνουμε ότι τα embeddings μας δεν είναι γραμμικά διαχωρίσιμα ως προς το τελικό πρόβλημα. Επίσης, τα συγκεκριμένα μοντέλα, έχουν την δυνατότητα να παράγουν feature-importances, οπότε μπορούν να μας δώσουν interpretation για το ποια χαρακτηριστικά είναι σημαντικά ως προς την πρόβλεψη του χρόνου

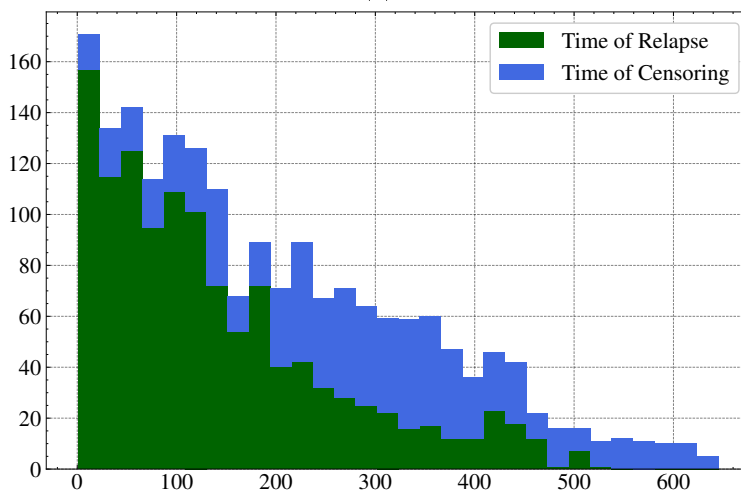
¹Στην πραγματικότητα το TSBert είναι ένας transformer, ο οποίος παράγει 256 features για κάθε timestep. Οπότε κάναμε Global AdaptiveConcatPool + Flatten [HSK19]. Μία εναλλακτική θα ήταν μόνο το Flatten, όμως τότε θα είχαμε πολύ μεγάλο feature-space.

υποτροπής. Βέβαια, αυτό είναι χρήσιμο μόνο για το TSFresh όπου τα features είναι συναρτήσεις της χρονοσειράς και άρα έχουν κάποια σημασιολογία (για παράδειγμα: $\|\cdot\|_2$ = ενέργεια).

- **Linear Survival SVM:** Από αυτή την κλάση εκπαιδεύσαμε δύο μοντέλα: ένα με $r = 0$, δηλαδή πρόβλημα regression και ένα με $r = 1$, δηλαδή πρόβλημα ranking, όπως περιγράφονται στην Παράγραφο 5.8.3.
- **Kernel Survival SVM:** Τέλος, εκπαιδεύσαμε ένα Kernel Survival Support Vector Machine με Radial Basis Function ως kernel συνάρτηση, δηλαδή $k(x, x') = \exp(-\gamma\|x - x'\|^2)$.



(a)



(b)

Σχήμα 6.4.2: (a) Οπτικοποίηση ενός sample του dataset, δηλαδή 12 ώρες συνεχόμενων μετρήσεων από τους τρεις sensors και τα cyclically-encoded χαρακτηριστικά της ώρας. (b) Ιστόγραμμα των time-to-event χρόνων για censored/non-censored δείγματα.

Επιλογή μετρικών για αξιολόγηση

Κάθε μοντέλο από τα παραπάνω αξιολογήθηκε ως προς τα: **Uno's concordance index (IPCW)** και **Time-dependent Area under the ROC** για να βγάλουμε συμπεράσματα για τη διαχωριστική του ικανότητα και ως προς το **Integrated Brier Score (IBS)**, για να βγάλουμε συμπεράσματα σχετικά με το calibration του μοντέλου. Οι μετρικές αυτές περιγράφονται αναλυτικά στην Παράγραφο 5.8.3. Πολύ σύντομα: Η ερμηνεία

των IPCW και Time-dependent Area under the ROC (AUC), είναι πανομοιότυπη με την AUC για δυαδική ταξινόμηση: η τιμή 0,5 υποδηλώνει ένα μοντέλο με τυχαίες προβλέψεις, η τιμή 1 υποδηλώνει ένα τέλει μοντέλο και η τιμή 0 υποδηλώνει ένα εντελώς λάθος μοντέλο. Όμοια, το Time-dependent Brier Score είναι μια επέκταση του μέσου τετραγώνου σφάλματος για right-censored δεδομένα, αφού μετρά την ευκλείδεια απόσταση μεταξύ του πραγματικού survival status και της προβλεπόμενης survival probability. Έτσι, μας δίνει συμπεράσματα ως προς την διαχωριστική ικανότητα αλλά και ως προς το calibration των πιθανοτήτων του μοντέλου. Επειδή, όπως περιγράφηκε στην Παράγραφο 5.8.3 η μετρική αυτή παράγει αποτέλεσμα για κάθε χρονική στιγμή της survival συνάρτησης, χρησιμοποιήσαμε το IBS, δηλαδή το ολοκλήρωμά της για όλες τις χρονικές στιγμές:

$$\text{IBS}(t_{\max}) = \frac{1}{t_{\max}} \int_0^{t_{\max}} \text{BS}(t) dt \quad (6.4.1)$$

Ένα σημείο που χρειάζεται ιδιαίτερη προσοχή, είναι ότι οι παραπάνω μετρικές εξαρτώνται από την αντίστροφη πιθανότητα του censoring-weight, άρα για να έχουμε αποτέλεσμα, αυτή η πιθανότητα θα πρέπει να είναι μη μηδενική. Οπότε, είναι σημαντικό να επιλέξουμε τέτοια χρονικά σημεία αξιολόγησης, έτσι ώστε η πιθανότητα να έχουμε censoring μετά το τελευταίο χρονικό σημείο να είναι μη μηδενική. Στην εργασία μας ορίσαμε ακολουθήσαμε τις οδηγίες του γνωστού πακέτου scikit-survival [Pöl120] και ορίσαμε το ανώτερο σημείο που κάνουμε evaluation ως το 80% percentile των παρατηρούμενων χρονικών σημείων στο σύνολο εκπαίδευσης. Επίσης, η μετρική Integrated Brier-Score μπορεί να προκύψει μόνο για τα μοντέλα που παράγουν πιθανότητες και όχι μόνο “hard labels”. Τέλος, η αξιολόγηση έγινε με 5-fold-cross-validation (χωρίς stratification, ως προς το censoring, αφού οι κατανομές censored/non-censored είναι παρόμοιες) και τα αποτελέσματα φαίνονται στο Σχήμα 6.4.3.

6.4.2 Αποτελέσματα και σχολιασμός

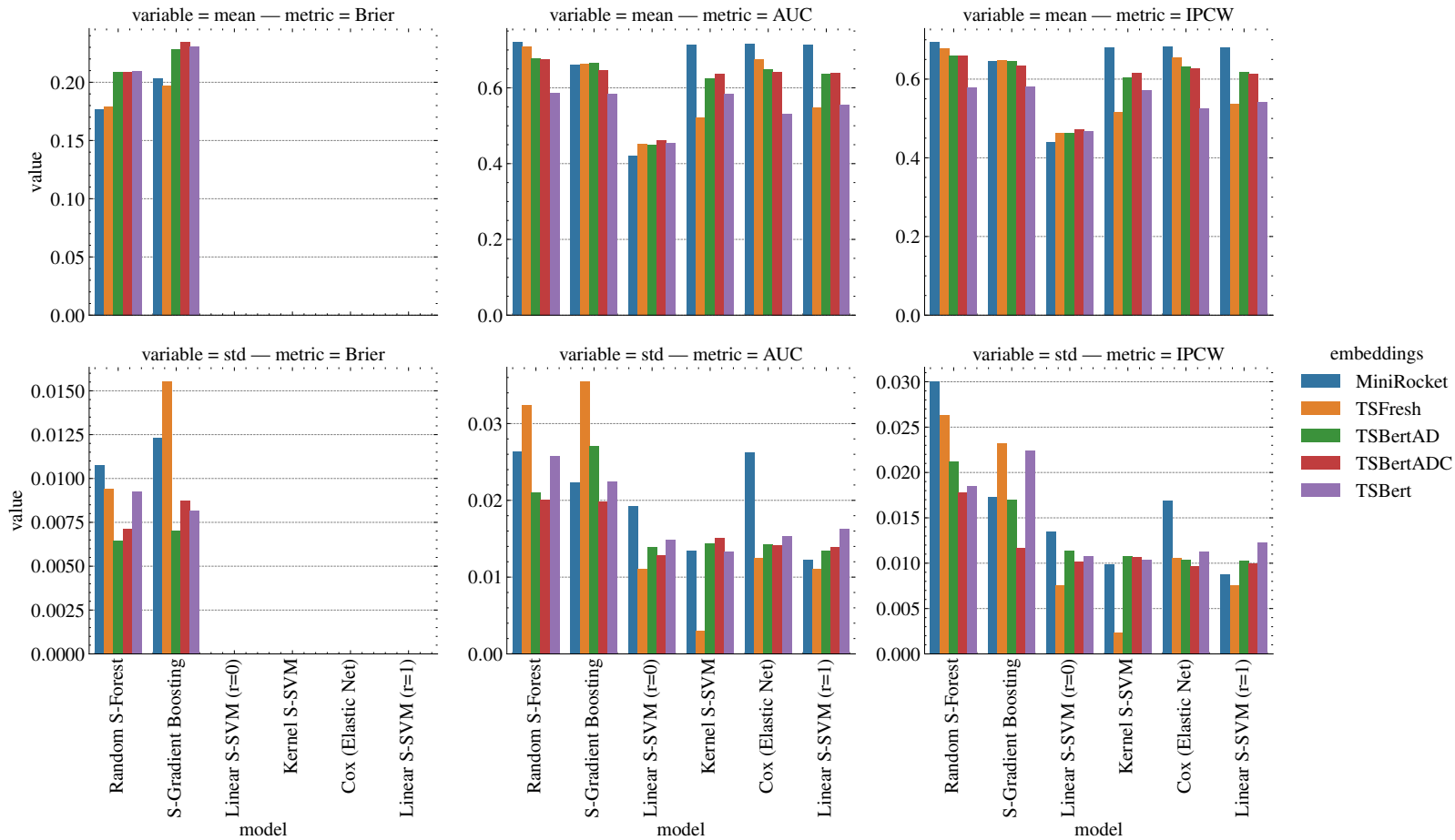
Στο Σχήμα 6.4.3 και στον Πίνακα 6.2 φαίνονται συγκεντρωτικά τα 5-fold-cross-validation αποτελέσματα πρόβλεψης του χρονικού διαστήματος μέχρι την επόμενη υποτροπή. Αυτά μπορούν να αναλυθούν ως προς τρεις άξονες:

metric	embeddings variable model	TSFresh	TSBert	MiniRocket	TSBertAD	TSBertADC	TSFresh	TSBert	MiniRocket	TSBertAD	TSBertADC
		mean	mean	mean	mean	mean	std	std	std	std	std
IPCW	Random S-Forest	0.678	0.579	0.694	0.659	0.659	0.026	0.018	0.030	0.021	0.018
	Cox (Elastic Net)	0.655	0.526	0.683	0.631	0.628	0.011	0.011	0.017	0.010	0.010
	S-Gradient Boosting	0.648	0.580	0.645	0.647	0.633	0.023	0.022	0.017	0.017	0.012
	Linear S-SVM (r=1)	0.538	0.542	0.680	0.617	0.613	0.008	0.012	0.009	0.010	0.010
	Kernel S-SVM	0.516	0.572	0.680	0.605	0.615	0.002	0.010	0.010	0.011	0.011
IBS	Linear S-SVM (r=0)	0.462	0.468	0.440	0.464	0.471	0.008	0.011	0.013	0.011	0.010
	S-Gradient Boosting	0.197	0.231	0.203	0.228	0.234	0.016	0.008	0.012	0.007	0.009
AUC	Random S-Forest	0.179	0.209	0.177	0.209	0.209	0.009	0.009	0.011	0.006	0.007
	S-Gradient Boosting	0.197	0.231	0.203	0.228	0.234	0.016	0.008	0.012	0.007	0.009
	Random S-Forest	0.710	0.587	0.721	0.678	0.676	0.032	0.026	0.026	0.021	0.020
	Cox (Elastic Net)	0.676	0.532	0.716	0.648	0.642	0.013	0.015	0.026	0.014	0.014
	S-Gradient Boosting	0.663	0.585	0.660	0.665	0.647	0.036	0.022	0.022	0.027	0.020
	Linear S-SVM (r=1)	0.547	0.555	0.714	0.637	0.639	0.011	0.016	0.012	0.013	0.014
Kernel S-SVM	0.522	0.584	0.714	0.624	0.636	0.003	0.013	0.013	0.014	0.015	
	Linear S-SVM (r=0)	0.452	0.455	0.421	0.450	0.462	0.011	0.015	0.019	0.014	0.013

Πίνακας 6.2: IPCW-AUC (higher is better) και IBS (lower is better) μετρικές για κάθε μοντέλο και κάθε embedding. Επίσης, φαίνεται με **bold** το καλύτερο embedding για κάθε συνδυασμό μετρικής/τελικού survival μοντέλου. Επίσης, βλέπουμε με **πράσινο** χρώμα τον καλύτερο συνδυασμό survival-μοντέλου/embedding-εισόδου, ως προς κάθε μετρική και με **κόκκινο** χρώμα τον χειρότερο. Στην περίπτωση του step task έχουμε ισοβαθμία στο μέγιστο.

1. **Διαχωριστική ισχύς των μοντέλων**, εστιάζοντας στις μετρικές IPCW και AUC. Συγκρίνοντας ως προς τα embeddings:

- Βλέπουμε ότι το MiniRocket αποδίδει εξαιρετικά ως προς τις μετρικές AUC και IPCW, αφού έχει την καλύτερη απόδοση για όλα τα survival μοντέλα, εκτός από τα S-Gradient Boosting και Linear S-SVM ($r = 0$) στα οποία δεν αποδίδει καλά.
- Επίσης, για όλα τα τελικά survival μοντέλα, όταν προ-εκπαιδύουμε με περισσότερα δεδομένα (TSBertAD ή TSBertADC σε σύγκριση με TSBert) οι μετρικές IPCW και AUC αυξάνονται, χωρίς



Σχήμα 6.4.3: Αποτελέσματα 5-fold-cross-validation, ως προς IPCW-AUC (higher is better) και Brier (lower is better) μετρικές, για κάθε μοντέλο και κάθε embedding. Στην πάνω σειρά φαίνονται οι μέσοι όροι των μετρικών για τις 5 επαναλήψεις, ενώ στην κάτω σειρά οι αντίστοιχες τυπικές αποκλίσεις.

όμως να φτάνουμε την απόδοση του MiniRocket. Το ίδιο δεν συμβαίνει, όταν συγκρίνουμε το TS-BertAD με το TSBertADC, δηλαδή όταν εκπαιδεύουμε σε παραπάνω δεδομένα από controls, αφού η απόδοση παραμένει σταθερή. Αυτό είναι σημείο προς εξερεύνηση και πρέπει να δοκιμάσουμε την προεκπαίδευση με περισσότερα control δεδομένα (συγκρίσιμα στο μέγεθος με αυτά των patients) για να επιβεβαιωθεί. Δυστυχώς, όμως, όπως είδαμε στο dataset μας έχουμε περιορισμένα δεδομένα από controls.

- Το TSFresh δίνει ελαφρώς καλύτερα αποτελέσματα από τις τρεις παραλλαγές του TSBERT, αφού έχει μέγιστα $(IPCW, AUC) = (0.678, 0.710)$, ενώ το δεύτερο $(IPCW, AUC) = (0.659, 0.678)$. Έχει όμως το μειονέκτημα ότι όσα unlabeled δεδομένα και αν έχουμε η απόδοση αυτή θα παραμείνει δεν μπορεί να βελτιωθεί.

Συγκρίνοντας προς τα survival-μοντέλα:

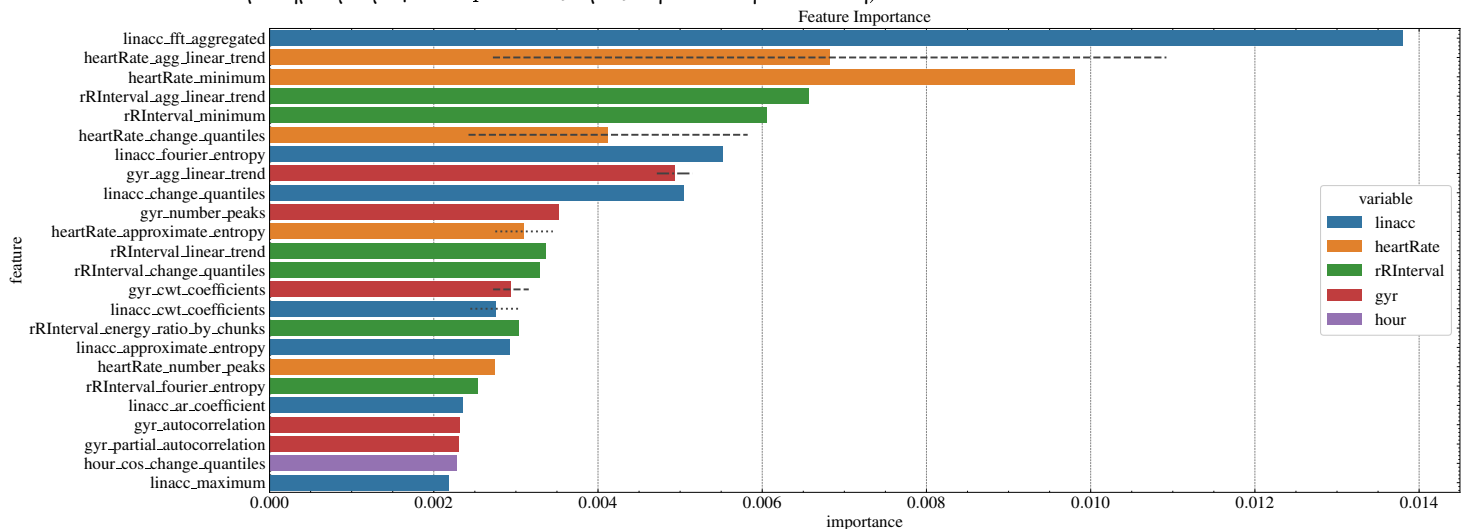
- Την καλύτερη απόδοση την έχει το Random S-Forest μοντέλο, πράγμα που αποδεικνύει την μη γραμμική σχέση μεταξύ embeddings και τελικού task.
- Βλέπουμε ότι το Linear-SVM με $r = 0$, το οποίο λύνει το πρόβλημα βελτιστοποίησης του regression, αποδίδει χειρότερα και στις δύο μετρικές. Μάλιστα σύμφωνα με τον Πίνακα 6.2, βλέπουμε ότι τα IPCW, AUC παίρνουν τιμές μικρότερες από το 0.5, δηλαδή το μοντέλο αυτό είναι οριακά χειρότερο από το random.

2. **Calibration**, εστιάζοντας στο IBS. Εδώ μπορούμε να συγκρίνουμε μόνο τα μοντέλα που μπορούν να παράγουν συνάρτηση κινδύνου για κάθε χρονική στιγμή, δηλαδή τα Random S-Forest και S-Gradient Boosting. Ως προς τα embeddings:

- Βλέπουμε ότι το Minirocket έχει το μικρότερο IBS, αλλά με όχι μεγάλη διαφορά από το TSFresh.
- Τα embeddings των TSBerts δεν παράγουν calibrated πιθανότητες και δυστυχώς αυτό δεν φαίνεται να βελτιώνεται όταν αυξάνουμε τα δεδομένα πάνω στα οποία κάνουμε προεκπαίδευση.

Επίσης, ως προς τα survival-μοντέλα βλέπουμε ότι τα Random S-Forest έχει μικρότερο IBS, άρα καλύτερο calibration.

3. **Robustness**, εστιάζοντας στις τυπικές αποκλίσεις των μετρικών. Παρατηρούμε ότι τα μοντέλα που βασίζονται σε δέντρα αποφάσεων, δηλαδή τα Random S-Forest και S-Gradient Boosting, παρά την εξαιρετική τους διαχωριστική ικανότητα και τη δυνατότητα να παράγουν συνάρτηση κινδύνου για κάθε χρονική στιγμή, έχουν αρκετά μεγάλη τυπική απόκλιση για τα 5 cross-validation πειράματα που εκτελέσαμε. Αυτό γίνεται ακόμη πιο έντονο, όταν αυτά εκπαιδεύονται πάνω στα embeddings του TSFresh (στο Σχήμα 6.4.3 παρατηρούμε μεγάλα spikes ως προς την τυπική απόκλιση).



Σχήμα 6.4.4: Feature Importances για τα embeddings του TSFresh, για το μοντέλο Random S-Forest μέσω αλγορίθμου Permutation Importance [Bre01].

Επιπλέον, στο Σχήμα 6.4.4 βλέπουμε τα 30 πιο σημαντικά features που προέκυψαν από το μοντέλο Random S-Forest για τα embeddings του TSFresh. Το feature importance προέκυψε με Permutation Importance [Bre01] αλγόριθμο, δηλαδή συγκρίνοντας πώς αλλάζει το c-index (IPCW) καθώς αφαιρούμε features από το survival μοντέλο. Βλέπουμε ότι και οι τρεις σενσορες θεωρούνται εξίσου σημαντικοί από το μοντέλο, ενώ τα features που επιλέγονται είναι κλασσικά time-series χαρακτηριστικά όπως: συντελεστές fourier/cwt και aggregations τους number of peaks, κ.α.

Τελικά, βλέποντας το πρόβλημα από πολλές οπτικές γωνίες, βλέπουμε ότι δεν υπάρχει μοναδική καλύτερη μέθοδος, αλλά ιδανικά θα μπορούσαμε να συνδυάσουμε τις επιμέρους προβλέψεις ως είσοδο σε μία καινούρια, πιο σύνθετη μέθοδο.

Κεφάλαιο 7

Επίλογος

7.1 Σύνοψη και Συμπεράσματα

Η παρούσα διπλωματική μπορεί να χωριστεί σε δύο σκέλη: Τα προβλήματα κατηγοριοποίησης, συγκεκριμένα:

- Κατηγοριοποίηση sleep/awake/transition
- Κατηγοριοποίηση walking/not-walking/transition
- Ταυτοποίηση χρήστη

και το πρόβλημα του survival-analysis, δηλαδή: time-to-relapse estimation. Και τα δύο προβλήματα αντιμετωπίστηκαν με τεχνικές οι οποίες μπορούν να χωριστούν σε δύο στάδια: Ένα πρώτο στάδιο unsupervised προεκπαίδευσης, το οποίο παράγει πυκνές αναπαραστάσεις για κάθε είσοδο και ένα δεύτερο, στο οποίο εκπαιδεύουμε ένα γραμμικό ταξινομητή, ως προς το τελικό task, με είσοδο τις αναπαραστάσεις αυτές.

Για τα προβλήματα κατηγοριοποίησης:

- Είδαμε ότι οι αναπαραστάσεις που προκύπτουν από τυχαίους συνελικτικούς πυρήνες αποδίδουν εξαιρετικά και σταθερά, αφού και στα τρία προβλήματα έχουν ίσο ή και μεγαλύτερο accuracy από το fully-supervised μοντέλο.
- Είδαμε ότι το SSL μοντέλο SelfTime πετυχαίνει το ίδιο accuracy με το fully-supervised Inception, όταν έχει δει μόλις 20% των επισημειώσεων και για τις τρεις εργασίες κατηγοριοποίησης. Το ίδιο ισχύει και για τη φάση του pretraining, δηλαδή, και τα δύο SSL μοντέλα συγκλίνουν ως προς την απόδοσή τους στο τελικό πρόβλημα, όταν έχουν προεκπαιδευτεί με το 20% των δεδομένων.
- Τέλος, είδαμε ότι το μοντέλο TSBert, παρά το μεγάλο capacity του λόγω της αρχιτεκτονικής transformer και την ευκολία του να χειρίζεται εισόδους χρονοσειρές πολλών μεταβλητών, δεν εξερευνά την σχέση μεταξύ δειγμάτων (inter-sample relationship) και αυτό του στοιχίζει στο task του ύπνου και του person identification.

Για το πρόβλημα της πρόβλεψης του χρονικού διαστήματος μέχρι την υποτροπή:

- Ξεχώρισαν τα embeddings του Minirocket, για την εξαιρετική τους απόδοση και τις calibrated πιθανότητες που μπορούμε να παράγουμε από αυτά.
- Παράλληλα, ξεχωρίσαμε το TFresh για τη σταθερή απόδοση σε λίγα δεδομένα και το interpretability που μπορεί να μας παρέχει, αφού τα embeddings του είναι συναρτήσεις της εισόδου.
- Επιπλέον, παρατηρήσαμε ότι υπάρχει περιθώριο βελτίωσης των embeddings του TSBert, όταν αυτό προεκπαιδεύεται με παραπάνω unlabeled δεδομένα, κάτι το οποίο δεν συμβαίνει με τις άλλες δύο μεθόδους.
- Τέλος, είδαμε ότι, τα μοντέλα που βασίζονται σε δέντρα αποφάσεων, δηλαδή τα Random S-Forest και S-Gradient Boosting δίνουν τα καλύτερα αποτελέσματα, λόγω της ικανότητάς τους να μαθαίνουν μη γραμμικές σχέσεις πάνω στα embeddings. Δυστυχώς, παρατηρήσαμε ότι έχουν μεγάλη τυπική απόκλιση,

όταν εκτελούμε την εκπαίδευση πολλές φορές, κάτι το οποίο δεν μπορεί να αγνοηθεί σε ένα τόσο σοβαρό πρόβλημα όπως η πρόβλεψη μίας υποτροπής.

7.2 Μελλοντικές Επεκτάσεις

Το σύνολο δεδομένων που χρησιμοποιήθηκε στην παρούσα μελέτη είναι τόσο μεγάλο σε όγκο και με τόσες επεκτάσεις που δύσκολα χωρούν σε μία εργασία. Μερικές από αυτές μπορεί να είναι:

- Ίσως για πρόβλημά της πρόβλεψης της υποτροπής, η μία ημέρα μετρήσεων ανά δείγμα, που επιλέξαμε να μην αρκεί να προβλέψει κάτι τέτοιο. Όντως, η δυναμική της πορείας της ασθένειας από μέρα σε μέρα, ίσως έδινε επιπλέον πληροφορία στο μοντέλο. Άρα, μια πιθανή επέκταση είναι η ανάλυση του προβλήματος σε πολλές χρονικές κλίμακες (Multi-scale / Multi-Resolution).
- Έχειδειχθεί ότι η μοντελοποίηση του προβλήματος της υποτροπής μέσα από τη σκοπιά του anomaly detection αποδίδει εξαιρετικά καλά [Hen+21]. Η σύγκριση της αναπαράστασης που παράγεται μέσω του anomaly detection με αυτή της αυτοεπιβλεπόμενης μάθησης, ως προς το task της πρόβλεψης της υποτροπής θα είχε μεγάλο ενδιαφέρον. Το ίδιο ισχύει και για το συνδυασμό τους και κατά πόσο αυτός αποδίδει.
- Είναι γεγονός ότι η μεγαλύτερη έρευνα στον τομέα του SSL γίνεται στον τομέα της όρασης υπολογιστών. Αν μπορούσε κανείς με κάποιον τρόπο να μετατρέψει δεδομένα χρονοσειρών σε εικόνες τότε θα μπορούσε να εκμεταλλευτεί όλες αυτές τις τεχνικές. Ένας τέτοιος τρόπος, παρουσιάζεται στην Παράγραφο 5.7.
- Επίσης, για τα προβλήματα κατηγοριοποίησης μπορούμε να δοκιμάσουμε συνδυασμούς αισθητήρων, αλλά και τους τρεις άξονες συνδυαστικά. Επίσης, μπορεί να συγκρίνουμε διάφορους συνδυασμούς διαστήματος μετρήσεων με resampling rates ή και να εκπαιδεύσουμε κάποιο multi-scale μοντέλο.
- Τέλος, για τα προβλήματα κατηγοριοποίησης μπορούμε να σχεδιάσουμε μια αρχιτεκτονική που να συνδυάζει τα θετικά του SelfTime και του TSBert. Συγκεκριμένα την διερεύνηση σχέσεων μεταξύ δειγμάτων (inter-sample relationship) που κάνει το SelfTime και τη χρήση αρχιτεκτονικής Transformer του TSBert.

Παράρτημα Α

Βιβλιογραφία

- [Adl+20] Adler, D. A. et al. “Predicting early warning signs of psychotic relapse from passive sensing data: an approach using encoder-decoder neural networks”. In: *JMIR mHealth and uHealth* 8.8 (2020), e19962.
- [AG82] Andersen, P. K. and Gill, R. D. “Cox’s Regression Model for Counting Processes: A Large Sample Study”. In: *The Annals of Statistics* 10.4 (Dec. 1982). DOI: [10.1214/aos/1176345976](https://doi.org/10.1214/aos/1176345976). URL:
- [ACB17] Arjovsky, M., Chintala, S., and Bottou, L. *Wasserstein GAN*. 2017. arXiv: [1701.07875](https://arxiv.org/abs/1701.07875) [[stat.ML](https://arxiv.org/abs/1701.07875)].
- [AMC17] Aung, M. H., Matthews, M., and Choudhury, T. “Sensing behavioral symptoms of mental health and delivering personalized interventions using mobile technologies”. In: *Depression and anxiety* 34.7 (2017), pp. 603–609.
- [BKH16] Ba, J. L., Kiros, J. R., and Hinton, G. E. *Layer Normalization*. 2016. arXiv: [1607.06450](https://arxiv.org/abs/1607.06450) [[stat.ML](https://arxiv.org/abs/1607.06450)].
- [Bag+15] Bagnall, A. et al. “Time-series classification with COTE: the collective of transformation-based ensembles”. In: *IEEE Transactions on Knowledge and Data Engineering* 27.9 (2015), pp. 2522–2535.
- [Bag+17] Bagnall, A. et al. “The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances”. en. In: *Data Mining and Knowledge Discovery* 31.3 (May 2017), pp. 606–660. ISSN: 1384-5810, 1573-756X. DOI: [10.1007/s10618-016-0483-9](https://doi.org/10.1007/s10618-016-0483-9). URL: (visited on 10/25/2021).
- [BCB16] Bahdanau, D., Cho, K., and Bengio, Y. *Neural Machine Translation by Jointly Learning to Align and Translate*. 2016. arXiv: [1409.0473](https://arxiv.org/abs/1409.0473) [[cs.CL](https://arxiv.org/abs/1409.0473)].
- [Bar+14] Barmada, S. et al. “Arc detection in pantograph-catenary systems by the use of support vector machines-based classification”. In: *IET Electrical Systems in Transportation* 4.2 (2014), pp. 45–52.
- [Bar+18] Barnett, I. et al. “Relapse prediction in schizophrenia through digital phenotyping: a pilot study”. In: *Neuropsychopharmacology* 43.8 (2018), pp. 1660–1666.
- [BH92] Becker, S. and Hinton, G. E. “Self-organizing neural network that discovers surfaces in random-dot stereograms”. In: *Nature* 355.6356 (Jan. 1992), pp. 161–163. DOI: [10.1038/355161a0](https://doi.org/10.1038/355161a0). URL:
- [Ben09] Bengio, Y. *Learning deep architectures for AI*. Now Publishers Inc, 2009.
- [Bis06] Bishop, C. M. *Pattern Recognition and Machine Learning*. Springer, 2006.
- [BML15] Boletsis, C., McCallum, S., and Landmark, B. F. “The use of smartwatches for health monitoring in home-based dementia care”. In: *International Conference on Human Aspects of IT for the Aged Population*. Springer. 2015, pp. 15–26.
- [BGV92] Boser, B. E., Guyon, I. M., and Vapnik, V. N. “A training algorithm for optimal margin classifiers”. In: *Proceedings of the fifth annual workshop on Computational learning theory - COLT '92*. ACM Press, 1992. DOI: [10.1145/130385.130401](https://doi.org/10.1145/130385.130401). URL:
- [BH06] Bowie, C. R. and Harvey, P. D. “Cognitive deficits and functional outcome in schizophrenia”. In: *Neuropsychiatric disease and treatment* 2.4 (2006), p. 531.
- [Bre01] Breiman, L. In: *Machine Learning* 45.1 (2001), pp. 5–32. DOI: [10.1023/a:1010933404324](https://doi.org/10.1023/a:1010933404324). URL:

- [Bri+17] Britz, D. et al. *Massive Exploration of Neural Machine Translation Architectures*. 2017. arXiv: [1703.03906](#) [cs.CL].
- [BRO+93] BROMLEY, J. et al. “SIGNATURE VERIFICATION USING A “SIAMESE” TIME DELAY NEURAL NETWORK”. In: *International Journal of Pattern Recognition and Artificial Intelligence* 07.04 (1993), pp. 669–688. DOI: [10.1142/S0218001493000339](#). eprint: URL:
- [Bro+19] Brouwer, E. D. et al. *GRU-ODE-Bayes: Continuous modeling of sporadically-observed time series*. 2019. arXiv: [1905.12374](#) [cs.LG].
- [Bro+20] Brown, T. B. et al. *Language Models are Few-Shot Learners*. 2020. arXiv: [2005.14165](#) [cs.CL].
- [Cal04] Callaghan, P. “Exercise: a neglected intervention in mental health care?” In: *Journal of psychiatric and mental health nursing* 11.4 (2004), pp. 476–483.
- [Can] Canziani, Y. L. bibinitperiod A. *DS-GA 1008 · SPRING 2020 · NYU CENTER FOR DATA SCIENCE*. URL:
- [Car+19] Caron, M. et al. “Unsupervised Pre-Training of Image Features on Non-Curated Data”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2019.
- [Car+21] Caron, M. et al. *Unsupervised Learning of Visual Features by Contrasting Cluster Assignments*. 2021. arXiv: [2006.09882](#) [cs.CV].
- [Cay05] Cayton, L. “Algorithms for manifold learning”. In: *Univ. of California at San Diego Tech. Rep* 12.1-17 (2005), p. 1.
- [Cel+18] Cella, M. et al. “Using wearable technology to detect the autonomic signature of illness severity in schizophrenia”. In: *Schizophrenia research* 195 (2018), pp. 537–542.
- [Che+20] Chen, T. et al. *A Simple Framework for Contrastive Learning of Visual Representations*. 2020. arXiv: [2002.05709](#) [cs.LG].
- [CH20] Chen, X. and He, K. *Exploring Simple Siamese Representation Learning*. 2020. arXiv: [2011.10566](#) [cs.CV].
- [CEJ21] Chivilgina, O., Elger, B. S., and Jotterand, F. “Digital Technologies for Schizophrenia Management: A Descriptive Review”. In: *Science and Engineering Ethics* 27.2 (Apr. 2021). DOI: [10.1007/s11948-021-00302-z](#). URL:
- [CHL05] Chopra, S., Hadsell, R., and LeCun, Y. “Learning a similarity metric discriminatively, with application to face verification”. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*. Vol. 1. 2005, 539–546 vol. 1. DOI: [10.1109/CVPR.2005.202](#).
- [CKF17] Christ, M., Kempa-Liehr, A. W., and Feindt, M. *Distributed and parallel time series feature extraction for industrial big data applications*. 2017. arXiv: [1610.07717](#) [cs.LG].
- [Chr+18] Christ, M. et al. “Time Series Feature Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package)”. In: *Neurocomputing* 307 (2018), pp. 72–77. ISSN: 0925-2312. DOI: <https://doi.org/10.1016/j.neucom.2018.03.067>. URL:
- [Coh08] Cohrs, S. “Sleep disturbances in patients with schizophrenia”. In: *CNS drugs* 22.11 (2008), pp. 939–962.
- [Col+11] Collobert, R. et al. “Natural Language Processing (Almost) from Scratch”. In: *Journal of Machine Learning Research* 12.76 (2011), pp. 2493–2537. URL:
- [Con+20] Conneau, A. et al. *Unsupervised Cross-lingual Representation Learning at Scale*. 2020. arXiv: [1911.02116](#) [cs.CL].
- [CLE90] Corrigan, P. W., Liberman, R. P., and Engel, J. D. “From noncompliance to collaboration in the treatment of schizophrenia”. In: *Psychiatric Services* 41.11 (1990), pp. 1203–1211.
- [CV95] Cortes, C. and Vapnik, V. “Support-vector networks”. In: *Machine Learning* 20.3 (Sept. 1995), pp. 273–297. DOI: [10.1007/bf00994018](#). URL:
- [Cov65] Cover, T. M. “Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition”. In: *IEEE Trans. Electron. Comput.* 14 (1965), pp. 326–334.
- [CP11] Cox, D. and Pinto, N. “Beyond simple features: A large-scale feature search approach to unconstrained face recognition”. In: *2011 IEEE International Conference on Automatic Face & Gesture Recognition (FG)*. IEEE. 2011, pp. 8–15.
- [Cut13] Cuturi, M. “Sinkhorn Distances: Lightspeed Computation of Optimal Transport”. In: *Advances in Neural Information Processing Systems*. Ed. by C. J. C. Burges et al. Vol. 26. Curran Associates, Inc., 2013. URL:

-
- [Cyb89] Cybenko, G. “Approximation by superpositions of a sigmoidal function”. In: *Mathematics of control, signals and systems* 2.4 (1989), pp. 303–314.
- [DT05] Dalal, N. and Triggs, B. “Histograms of oriented gradients for human detection”. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*. Vol. 1. 2005, 886–893 vol. 1. DOI: [10.1109/CVPR.2005.177](https://doi.org/10.1109/CVPR.2005.177).
- [Dau+19] Dau, H. A. et al. “The UCR time series archive”. In: *IEEE/CAA Journal of Automatica Sinica* 6.6 (2019), pp. 1293–1305.
- [DPW20] Dempster, A., Petitjean, F., and Webb, G. I. “ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels”. In: *Data Mining and Knowledge Discovery* 34.5 (July 2020), pp. 1454–1495. ISSN: 1573-756X. DOI: [10.1007/s10618-020-00701-z](https://doi.org/10.1007/s10618-020-00701-z). URL:
- [DSW21] Dempster, A., Schmidt, D. F., and Webb, G. I. “MiniRocket”. In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* (Aug. 2021). DOI: [10.1145/3447548.3467231](https://doi.org/10.1145/3447548.3467231). URL:
- [Dev+19] Devlin, J. et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: [1810.04805](https://arxiv.org/abs/1810.04805) [cs.CL].
- [DT17] DeVries, T. and Taylor, G. W. “Improved regularization of convolutional neural networks with cutout”. In: *arXiv preprint arXiv:1708.04552* (2017).
- [Dew+19] Dewa, L. H. et al. “Young adults’ perceptions of using wearables, social media and other technologies to detect worsening mental health: A qualitative study”. In: *PLOS ONE* 14.9 (Sept. 2019). Ed. by Q. Grundy, e0222655. DOI: [10.1371/journal.pone.0222655](https://doi.org/10.1371/journal.pone.0222655). URL:
- [DGE16] Doersch, C., Gupta, A., and Efros, A. A. *Unsupervised Visual Representation Learning by Context Prediction*. 2016. arXiv: [1505.05192](https://arxiv.org/abs/1505.05192) [cs.CV].
- [Don+00] Donoho, D. L. et al. “High-dimensional data analysis: The curses and blessings of dimensionality”. In: (2000).
- [DG03] Donoho, D. L. and Grimes, C. “Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data”. In: *Proceedings of the National Academy of Sciences* 100.10 (2003), pp. 5591–5596.
- [EH15] East, M. L. and Havard, B. C. “Mental Health Mobile Apps: From Infusion to Diffusion in the Mental Health Social System”. In: *JMIR Mental Health* 2.1 (Mar. 2015), e10. DOI: [10.2196/mental.3954](https://doi.org/10.2196/mental.3954). URL:
- [Fil+20] Filntisis, P. P. et al. “Identifying differences in physical activity and autonomic function patterns between psychotic patients and controls over a long period of continuous monitoring using wearable sensors”. In: *CoRR* abs/2011.02285 (2020). arXiv: [2011.02285](https://arxiv.org/abs/2011.02285). URL:
- [FW22] Fonseka, L. N. and Woo, B. K. P. “Wearables in Schizophrenia: Update on Current and Future Clinical Applications”. In: *JMIR mHealth and uHealth* 10.4 (Apr. 2022), e35600. DOI: [10.2196/35600](https://doi.org/10.2196/35600). URL:
- [For+19] Fortuin, V. et al. *SOM-VAE: Interpretable Discrete Representation Learning on Time Series*. 2019. arXiv: [1806.02199](https://arxiv.org/abs/1806.02199) [cs.LG].
- [FJ14] Fulcher, B. D. and Jones, N. S. “Highly comparative feature-based time-series classification”. In: *IEEE Transactions on Knowledge and Data Engineering* 26.12 (2014), pp. 3026–3037.
- [Geb+04] Gebraeel, N. et al. “Residual life predictions from vibration-based degradation signals: a neural network approach”. In: *IEEE Transactions on industrial electronics* 51.3 (2004), pp. 694–700.
- [Geh+17] Gehring, J. et al. *Convolutional Sequence to Sequence Learning*. 2017. arXiv: [1705.03122](https://arxiv.org/abs/1705.03122) [cs.CL].
- [GM00] Giannopoulos, A. and Milman, V. “Concentration property on probability spaces”. In: *Advances in Mathematics* 156 (2000), pp. 77–106. URL:
- [Gib16] Gibbs, J. W. “Elementary Principles of Statistical Mechanics Developed with Especial Reference to the Rational Foundation of Thermodynamics (Charles Scribner’s Sons, New York, 1902)”. In: *The Project Gutenberg EBook* 50992 (2016).
- [GSK18] Gidaris, S., Singh, P., and Komodakis, N. *Unsupervised Representation Learning by Predicting Image Rotations*. 2018. arXiv: [1803.07728](https://arxiv.org/abs/1803.07728) [cs.CV].
- [Gil+14] Gilbert, S. L. et al. “Dead before detection: addressing the effects of left truncation on survival estimation and ecological inference for neonates”. In: *Methods in Ecology and Evolution* 5.10 (Aug. 2014). Ed. by R. B. O’Hara, pp. 992–1001. DOI: [10.1111/2041-210x.12234](https://doi.org/10.1111/2041-210x.12234). URL:
-

- [Gol+05] Goldberger, J. et al. “Neighbourhood Components Analysis”. In: *Advances in Neural Information Processing Systems*. Ed. by L. Saul, Y. Weiss, and L. Bottou. Vol. 17. MIT Press, 2005. URL:
- [GMP00] Gong, S., McKenna, S. J., and Psarrou, A. *Dynamic vision: from images to face recognition*. World Scientific, 2000.
- [Goo+13] Goodfellow, I. J. et al. “Multi-digit number recognition from street view imagery using deep convolutional neural networks”. In: *arXiv preprint arXiv:1312.6082* (2013).
- [GBC16] Goodfellow, I. J., Bengio, Y., and Courville, A. *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
- [GT18] Gorban, A. N. and Tyukin, I. Y. “Blessing of dimensionality: mathematical foundations of the statistical physics of data”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 376.2118 (Mar. 2018), p. 20170237. ISSN: 1471-2962. DOI: [10.1098/rsta.2017.0237](https://doi.org/10.1098/rsta.2017.0237). URL:
- [GT17] Gorban, A. and Tyukin, I. “Stochastic separation theorems”. In: *Neural Networks* 94 (Oct. 2017), pp. 255–259. ISSN: 0893-6080. DOI: [10.1016/j.neunet.2017.07.014](https://doi.org/10.1016/j.neunet.2017.07.014). URL:
- [Goy+18] Goyal, P. et al. *Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour*. 2018. arXiv: [1706.02677](https://arxiv.org/abs/1706.02677) [cs.CV].
- [Goy+19] Goyal, P. et al. *Scaling and Benchmarking Self-Supervised Visual Representation Learning*. 2019. arXiv: [1905.01235](https://arxiv.org/abs/1905.01235) [cs.CV].
- [Goy+21] Goyal, P. et al. *Self-supervised Pretraining of Visual Features in the Wild*. 2021. arXiv: [2103.01988](https://arxiv.org/abs/2103.01988) [cs.CV].
- [Gra14] Graves, A. *Generating Sequences With Recurrent Neural Networks*. 2014. arXiv: [1308.0850](https://arxiv.org/abs/1308.0850) [cs.NE].
- [GMH13] Graves, A., Mohamed, A.-r., and Hinton, G. “Speech recognition with deep recurrent neural networks”. In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. 2013, pp. 6645–6649. DOI: [10.1109/ICASSP.2013.6638947](https://doi.org/10.1109/ICASSP.2013.6638947).
- [Gri+20] Grill, J.-B. et al. *Bootstrap your own latent: A new approach to self-supervised Learning*. 2020. arXiv: [2006.07733](https://arxiv.org/abs/2006.07733) [cs.LG].
- [HCL06] Hadsell, R., Chopra, S., and LeCun, Y. “Dimensionality Reduction by Learning an Invariant Mapping”. In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*. Vol. 2. 2006, pp. 1735–1742. DOI: [10.1109/CVPR.2006.100](https://doi.org/10.1109/CVPR.2006.100).
- [Hao21] Haoyi Fan Fengbin Zhang, Y. G. “Self-Supervised Time Series Representation Learning by Inter-Intra Relational Reasoning”. In: *Submitted to International Conference on Learning Representations*. under review. 2021. URL:
- [He+15] He, K. et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: [1512.03385](https://arxiv.org/abs/1512.03385) [cs.CV].
- [He+20] He, K. et al. “Momentum Contrast for Unsupervised Visual Representation Learning”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.
- [Hen+21] Henson, P. et al. “Anomaly detection to predict relapse risk in schizophrenia”. In: *Translational psychiatry* 11.1 (2021), pp. 1–6.
- [Her90] Hermansky, H. “Perceptual linear predictive (PLP) analysis of speech.” In: *The Journal of the Acoustical Society of America* 87 4 (1990), pp. 1738–52.
- [HVD15] Hinton, G., Vinyals, O., and Dean, J. *Distilling the Knowledge in a Neural Network*. 2015. arXiv: [1503.02531](https://arxiv.org/abs/1503.02531) [stat.ML].
- [Ho98] Ho, T. K. “The random subspace method for constructing decision forests”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20.8 (1998), pp. 832–844. DOI: [10.1109/34.709601](https://doi.org/10.1109/34.709601).
- [Hoc98] Hochreiter, S. “The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions”. In: *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* 6.2 (Apr. 1998). Place: River Edge, NJ, USA Publisher: World Scientific Publishing Co., Inc., pp. 107–116. ISSN: 0218-4885. DOI: [10.1142/S0218488598000094](https://doi.org/10.1142/S0218488598000094).
- [HS97] Hochreiter, S. and Schmidhuber, J. “Long Short-Term Memory”. In: *Neural Computation* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: [10.1162/neco.1997.9.8.1735](https://doi.org/10.1162/neco.1997.9.8.1735). eprint: URL:
- [HSW89] Hornik, K., Stinchcombe, M., and White, H. “Multilayer feedforward networks are universal approximators”. In: *Neural networks* 2.5 (1989), pp. 359–366.

-
- [Hun07] Hunter, J. D. “Matplotlib: A 2D graphics environment”. In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: [10.1109/MCSE.2007.55](https://doi.org/10.1109/MCSE.2007.55).
- [HSK19] Hyun, J., Seong, H., and Kim, E. *Universal Pooling – A New Pooling Method for Convolutional Neural Networks*. 2019. DOI: [10.48550/ARXIV.1907.11440](https://doi.org/10.48550/ARXIV.1907.11440). URL:
- [Ish+08] Ishwaran, H. et al. “Random survival forests”. In: *The Annals of Applied Statistics* 2.3 (Sept. 2008). DOI: [10.1214/08-aos169](https://doi.org/10.1214/08-aos169). URL:
- [Ism+19a] Ismail Fawaz, H. et al. “Deep learning for time series classification: a review”. In: *Data Mining and Knowledge Discovery* 33.4 (Mar. 2019), pp. 917–963. ISSN: 1573-756X. DOI: [10.1007/s10618-019-00619-1](https://doi.org/10.1007/s10618-019-00619-1). URL:
- [Ism+19b] Ismail Fawaz, H. et al. “Deep learning for time series classification: a review”. In: *Data Mining and Knowledge Discovery* 33.4 (Mar. 2019), pp. 917–963. ISSN: 1573-756X. DOI: [10.1007/s10618-019-00619-1](https://doi.org/10.1007/s10618-019-00619-1). URL:
- [Ism+20] Ismail Fawaz, H. et al. “InceptionTime: Finding AlexNet for time series classification”. In: *Data Mining and Knowledge Discovery* 34.6 (Nov. 2020), pp. 1936–1962. DOI: [10.1007/s10618-020-00710-y](https://doi.org/10.1007/s10618-020-00710-y). URL:
- [IU20] Iwana, B. K. and Uchida, S. “Time Series Data Augmentation for Neural Networks by Time Warping with a Discriminative Teacher”. In: *arXiv preprint arXiv:2004.08780* (2020).
- [Jar+09] Jarrett, K. et al. “What is the best multi-stage architecture for object recognition?” In: *2009 IEEE 12th international conference on computer vision*. IEEE, 2009, pp. 2146–2153.
- [JT19] Jing, L. and Tian, Y. *Self-supervised Visual Feature Learning with Deep Neural Networks: A Survey*. 2019. arXiv: [1902.06162](https://arxiv.org/abs/1902.06162) [cs.CV].
- [Kah+13] Kahou, S. E. et al. “Combining modality specific deep neural networks for emotion recognition in video”. In: *Proceedings of the 15th ACM on International conference on multimodal interaction*. 2013, pp. 543–550.
- [Kan+15] Kang, M. et al. “Time-Varying and Multiresolution Envelope Analysis and Discriminative Feature Analysis for Bearing Fault Diagnosis”. In: *IEEE Transactions on Industrial Electronics* 62.12 (2015), pp. 7749–7761. DOI: [10.1109/TIE.2015.2460242](https://doi.org/10.1109/TIE.2015.2460242).
- [Kar16] Kartsonaki, C. “Survival analysis”. In: *Diagnostic Histopathology* 22.7 (2016). Mini-Symposium: Medical Statistics, pp. 263–270. ISSN: 1756-2317. DOI: <https://doi.org/10.1016/j.mpdhp.2016.06.005>. URL:
- [Kas] Kassambara, A. *Cox model assumptions*. URL:
- [KP00] Keogh, E. J. and Pazzani, M. J. “Scaling up Dynamic Time Warping for Datamining Applications”. In: *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’00. Boston, Massachusetts, USA: Association for Computing Machinery, 2000, pp. 285–289. ISBN: 1581132336. DOI: [10.1145/347090.347153](https://doi.org/10.1145/347090.347153). URL:
- [Kou+12] Koutsouleris, N. et al. “Early recognition and disease prediction in the at-risk mental states for psychosis using neurocognitive pattern classification”. In: *Schizophrenia bulletin* 38.6 (2012), pp. 1200–1215.
- [KSH12a] Krizhevsky, A., Sutskever, I., and Hinton, G. E. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems* 25 (2012), pp. 1097–1105.
- [KSH12b] Krizhevsky, A., Sutskever, I., and Hinton, G. E. “Imagenet classification with deep convolutional neural networks”. In: *Advances in neural information processing systems* 25 (2012).
- [LMT16] Le Guennec, A., Malinowski, S., and Tavenard, R. “Data Augmentation for Time Series Classification using Convolutional Neural Networks”. In: *ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data*. 2016.
- [LBH15] LeCun, Y., Bengio, Y., and Hinton, G. “Deep learning”. In: *Nature* 521.7553 (May 2015), pp. 436–444. DOI: [10.1038/nature14539](https://doi.org/10.1038/nature14539). URL:
- [LM] LeCun, Y. and Misra, I. *Self-supervised learning: The dark matter of intelligence*. URL:
- [Led01] Ledoux, M. *The concentration of measure phenomenon*. 89. American Mathematical Soc., 2001.
- [Les+93] Leshno, M. et al. “Multilayer feedforward networks with a nonpolynomial activation function can approximate any function”. In: *Neural networks* 6.6 (1993), pp. 861–867.
- [Li+20] Li, S. et al. *Enhancing the Locality and Breaking the Memory Bottleneck of Transformer on Time Series Forecasting*. 2020. arXiv: [1907.00235](https://arxiv.org/abs/1907.00235) [cs.LG].
-

- [LTB18] Lines, J., Taylor, S., and Bagnall, A. “Time Series Classification with HIVE-COTE: The Hierarchical Vote Collective of Transformation-Based Ensembles”. In: *ACM Trans. Knowl. Discov. Data* 12.5 (July 2018). ISSN: 1556-4681. DOI: [10.1145/3182382](https://doi.org/10.1145/3182382). URL:
- [Liu+19] Liu, Y. et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019. arXiv: [1907.11692](https://arxiv.org/abs/1907.11692) [cs.CL].
- [Low99] Lowe, D. “Object recognition from local scale-invariant features”. In: *Proceedings of the Seventh IEEE International Conference on Computer Vision*. Vol. 2. 1999, 1150–1157 vol.2. DOI: [10.1109/ICCV.1999.790410](https://doi.org/10.1109/ICCV.1999.790410).
- [Maa97] Maass, W. “Bounds for the computational power and learning complexity of analog neural nets”. In: *SIAM Journal on Computing* 26.3 (1997), pp. 708–732.
- [MSS94] Maass, W., Schnitger, G., and Sontag, E. D. “A comparison of the computational power of sigmoid and Boolean threshold circuits”. In: *Theoretical Advances in Neural Computation and Learning*. Springer, 1994, pp. 127–151.
- [MH08] Maaten, L. van der and Hinton, G. “Visualizing data using t-SNE”. In: *Journal of Machine Learning Research* 9 (Nov. 2008), pp. 2579–2605.
- [Mag+20] Maglogiannis, I. et al. “An Intelligent Cloud-Based Platform for Effective Monitoring of Patients with Psychotic Disorders”. In: *IFIP Advances in Information and Communication Technology*. Springer International Publishing, 2020, pp. 293–307. DOI: [10.1007/978-3-030-49186-4_25](https://doi.org/10.1007/978-3-030-49186-4_25). URL:
- [Man+07] Mansell, W. et al. “The interpretation of, and responses to, changes in internal states: an integrative cognitive model of mood swings and bipolar disorders”. In: *Behavioural and Cognitive psychotherapy* 35.5 (2007), pp. 515–539.
- [McG+14] McGorry, P. et al. “Biomarkers and clinical staging in psychiatry”. In: *World Psychiatry* 13.3 (2014), pp. 211–223.
- [MHM18] McInnes, L., Healy, J., and Melville, J. *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*. 2018. DOI: [10.48550/ARXIV.1802.03426](https://doi.org/10.48550/ARXIV.1802.03426). URL:
- [MBR05] McNiel, D. E., Binder, R. L., and Robinson, J. C. “Incarceration associated with homelessness, mental disorder, and co-occurring substance abuse”. In: *Psychiatric Services* 56.7 (2005), pp. 840–846.
- [MJ20] Mekruksavanich, S. and Jitpattanakul, A. “Smartwatch-based Human Activity Recognition Using Hybrid LSTM Network”. In: *2020 IEEE SENSORS*. 2020, pp. 1–4. DOI: [10.1109/SENSORS47125.2020.9278630](https://doi.org/10.1109/SENSORS47125.2020.9278630).
- [Mey+18] Meyer, N. et al. “Capturing Rest-Activity Profiles in Schizophrenia Using Wearable and Mobile Technologies: Development, Implementation, Feasibility, and Acceptability of a Remote Monitoring Platform”. In: *JMIR mHealth and uHealth* 6.10 (Oct. 2018), e188. DOI: [10.2196/mhealth.8292](https://doi.org/10.2196/mhealth.8292). URL:
- [MM05] Mierswa, I. and Morik, K. “Automatic feature extraction for classifying audio data”. In: *Machine learning* 58.2 (2005), pp. 127–149.
- [Mon+14] Montúfar, G. et al. *On the Number of Linear Regions of Deep Neural Networks*. 2014. arXiv: [1402.1869](https://arxiv.org/abs/1402.1869) [stat.ML].
- [NM10] Narayanan, H. and Mitter, S. “Sample complexity of testing the manifold hypothesis”. In: *Proceedings of the 23rd International Conference on Neural Information Processing Systems-Volume 2*. 2010, pp. 1786–1794.
- [Neb+16] Nebeker, C. et al. “Engaging research participants to inform the ethical conduct of mobile imaging, pervasive sensing, and location tracking research”. In: *Translational Behavioral Medicine* 6.4 (Sept. 2016), pp. 577–586. DOI: [10.1007/s13142-016-0426-4](https://doi.org/10.1007/s13142-016-0426-4). URL:
- [NF17] Noroozi, M. and Favaro, P. *Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles*. 2017. arXiv: [1603.09246](https://arxiv.org/abs/1603.09246) [cs.CV].
- [Nun+15] Nun, I. et al. “Fats: Feature analysis for time series”. In: *arXiv preprint arXiv:1506.00010* (2015).
- [Ogu20] Oguiza, I. *tsai - A state-of-the-art deep learning library for time series and sequential data*. Github. 2020. URL:
- [OMS17] Olah, C., Mordvintsev, A., and Schubert, L. “Feature Visualization”. In: *Distill* (2017). <https://distill.pub/2017/feature-visualization>. DOI: [10.23915/distill.00007](https://doi.org/10.23915/distill.00007).
- [OLV18] Oord, A. v. d., Li, Y., and Vinyals, O. “Representation learning with contrastive predictive coding”. In: *arXiv preprint arXiv:1807.03748* (2018).

-
- [Ore+20] Oreshkin, B. N. et al. *N-BEATS: Neural basis expansion analysis for interpretable time series forecasting*. 2020. arXiv: [1905.10437 \[cs.LG\]](#).
- [PMB13] Pascanu, R., Mikolov, T., and Bengio, Y. “On the difficulty of training recurrent neural networks”. In: *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*. ICML’13. Atlanta, GA, USA: JMLR.org, June 2013, pp. III–1310–III–1318. (Visited on 10/01/2020).
- [Pat+12] Patel, S. et al. “A review of wearable sensors and systems with application in rehabilitation”. In: *Journal of neuroengineering and rehabilitation* 9.1 (2012), pp. 1–17.
- [Pat+16] Pathak, D. et al. *Context Encoders: Feature Learning by Inpainting*. 2016. arXiv: [1604.07379 \[cs.CV\]](#).
- [Pea01] Pearson, K. *LIII. On lines and planes of closest fit to systems of points in space*. Nov. 1901. DOI: [10.1080/14786440109462720](#). URL:
- [Pin+09] Pinto, N. et al. “A high-throughput screening approach to discovering good forms of biologically inspired visual representation”. In: *PLoS computational biology* 5.11 (2009), e1000579.
- [Pöl20] Pölsterl, S. “scikit-survival: A Library for Time-to-Event Analysis Built on Top of scikit-learn”. In: *Journal of Machine Learning Research* 21.212 (2020), pp. 1–6. URL:
- [Raf+20] Raffel, C. et al. *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. 2020. arXiv: [1910.10683 \[cs.LG\]](#).
- [Ret+20] Retsinas, G. et al. “Person Identification Using Deep Convolutional Neural Networks on Short-Term Signals from Wearable Sensors”. In: *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2020, pp. 3657–3661. DOI: [10.1109/ICASSP40776.2020.9053910](#).
- [Roo+19] Roomkham, S. et al. “Sleep monitoring with the apple watch: comparison to a clinically validated actigraph”. In: *F1000Research* 8 (2019), Article–number.
- [Sa94] Sa, V. R. de. “Learning classification with unlabeled data”. In: *Advances in neural information processing systems*. Citeseer. 1994, pp. 112–119.
- [Sae+15] Saeb, S. et al. “Mobile Phone Sensor Correlates of Depressive Symptom Severity in Daily-Life Behavior: An Exploratory Study”. In: *Journal of Medical Internet Research* 17.7 (July 2015), e175. DOI: [10.2196/jmir.4273](#). URL:
- [SR21] Saloni Dattani, H. R. and Roser, M. “Mental Health”. In: *Our World in Data* (2021). <https://ourworldindata.org/mental-health>.
- [SB11] Samuelson, F. and Brown, D. G. “Application of Cover’s theorem to the evaluation of the performance of CI observers”. In: *The 2011 International Joint Conference on Neural Networks*. 2011, pp. 1020–1026. DOI: [10.1109/IJCNN.2011.6033334](#).
- [Sax+11] Saxe, A. M. et al. “On random weights and unsupervised feature learning”. In: *Icml*. 2011.
- [SW17] Scardapane, S. and Wang, D. “Randomness in Neural Networks: An Overview”. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 7 (Jan. 2017). DOI: [10.1002/widm.1200](#).
- [SYD19] Sen, R., Yu, H.-F., and Dhillon, I. *Think Globally, Act Locally: A Deep Neural Network Approach to High-Dimensional Time Series Forecasting*. 2019. arXiv: [1905.03806 \[stat.ML\]](#).
- [Sha+20] Shanahan, M. et al. “Artificial Intelligence and the Common Sense of Animals”. In: *Trends in Cognitive Sciences* 24.11 (2020), pp. 862–872. ISSN: 1364-6613. DOI: <https://doi.org/10.1016/j.tics.2020.09.002>. URL:
- [She+20] Shen, S. et al. *PowerNorm: Rethinking Batch Normalization in Transformers*. 2020. arXiv: [2003.07845 \[cs.CL\]](#).
- [Shi+20] Shifaz, A. et al. “TS-CHIEF: a scalable and accurate forest algorithm for time series classification”. In: *Data Mining and Knowledge Discovery* 34 (May 2020). DOI: [10.1007/s10618-020-00679-8](#).
- [SM19a] Shrestha, A. and Mahmood, A. “Review of Deep Learning Algorithms and Architectures”. In: *IEEE Access* 7 (2019), pp. 53040–53065. DOI: [10.1109/ACCESS.2019.2912200](#).
- [SM19b] Shukla, S. N. and Marlin, B. M. *Interpolation-Prediction Networks for Irregularly Sampled Time Series*. 2019. arXiv: [1909.07782 \[cs.LG\]](#).
- [Sri+14] Srivastava, N. et al. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *Journal of Machine Learning Research* 15.56 (2014), pp. 1929–1958. URL:
-

- [SKK18] Stepnicki, P., Kondej, M., and Kaczor, A. A. “Current Concepts and Treatments of Schizophrenia”. In: *Molecules* 23.8 (Aug. 2018), p. 2087. DOI: [10.3390/molecules23082087](https://doi.org/10.3390/molecules23082087). URL:
- [Ste+19] Stevner, A. B. A. et al. “Discovery of key whole-brain transitions and dynamics during human wakefulness and non-REM sleep”. In: *Nature Communications* 10.1 (Mar. 2019). DOI: [10.1038/s41467-019-08934-3](https://doi.org/10.1038/s41467-019-08934-3). URL:
- [Stu+21] Stucky, B. et al. “Validation of Fitbit Charge 2 Sleep and Heart Rate Estimates Against Polysomnographic Measures in Shift Workers: Naturalistic Study”. In: *Journal of Medical Internet Research* 23.10 (Oct. 2021), e26476. DOI: [10.2196/26476](https://doi.org/10.2196/26476). URL:
- [Sze+14] Szegedy, C. et al. *Intriguing properties of neural networks*. 2014. arXiv: [1312.6199](https://arxiv.org/abs/1312.6199) [cs.CV].
- [Tan+20] Tan, C. W. et al. *Monash University, UEA, UCR Time Series Extrinsic Regression Archive*. 2020. arXiv: [2006.10996](https://arxiv.org/abs/2006.10996) [cs.LG].
- [The22] The Manim Community Developers. *Manim – Mathematical Animation Framework*. Version v0.15.1. Mar. 2022. URL:
- [TK18] Torous, J. and Keshavan, M. “A new window into psychosis: The rise digital phenotyping, smartphone assessment, and mobile monitoring”. In: *Schizophrenia Research* 197 (July 2018), pp. 67–68. DOI: [10.1016/j.schres.2018.01.005](https://doi.org/10.1016/j.schres.2018.01.005). URL:
- [Tor+14] Torous, J. et al. “Patient Smartphone Ownership and Interest in Mobile Apps to Monitor Symptoms of Mental Health Conditions: A Survey in Four Geographically Distinct Psychiatric Clinics”. In: *JMIR Mental Health* 1.1 (Dec. 2014), e5. DOI: [10.2196/mental.4004](https://doi.org/10.2196/mental.4004). URL:
- [Tor+15] Torous, J. et al. “Utilizing a Personal Smartphone Custom App to Assess the Patient Health Questionnaire-9 (PHQ-9) Depressive Symptoms in Patients With Major Depressive Disorder”. In: *JMIR Mental Health* 2.1 (Mar. 2015), e8. DOI: [10.2196/mental.3889](https://doi.org/10.2196/mental.3889). URL:
- [Tor+16] Torous, J. et al. “New Tools for New Research in Psychiatry: A Scalable and Customizable Platform to Empower Data Driven Smartphone Research”. In: *JMIR Mental Health* 3.2 (May 2016), e16. ISSN: 2368-7959. DOI: [10.2196/mental.5165](https://doi.org/10.2196/mental.5165). URL:
- [Um+17] Um, T. T. et al. “Data augmentation of wearable sensor data for parkinson’s disease monitoring using convolutional neural networks”. In: *Proceedings of the 19th ACM International Conference on Multimodal Interaction*. 2017, pp. 216–220.
- [Uno+11] Uno, H. et al. “On the C-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data”. In: *Statistics in Medicine* 30.10 (2011), pp. 1105–1117. DOI: <https://doi.org/10.1002/sim.4154>. eprint: URL:
- [Val+13] Valenza, G. et al. “Wearable monitoring for mood recognition in bipolar disorder based on history-dependent long-term heart rate variability analysis”. In: *IEEE Journal of Biomedical and Health Informatics* 18.5 (2013), pp. 1625–1635.
- [Vas+17] Vaswani, A. et al. *Attention Is All You Need*. 2017. arXiv: [1706.03762](https://arxiv.org/abs/1706.03762) [cs.CL].
- [Ver18] Vershynin, R. *High-dimensional probability: An introduction with applications in data science*. Vol. 47. Cambridge university press, 2018.
- [VTA16] Vigo, D., Thornicroft, G., and Atun, R. “Estimating the true global burden of mental illness”. In: *The Lancet Psychiatry* 3.2 (Feb. 2016), pp. 171–178. DOI: [10.1016/s2215-0366\(15\)00505-2](https://doi.org/10.1016/s2215-0366(15)00505-2). URL:
- [Vin+08] Vincent, P. et al. “Extracting and Composing Robust Features with Denoising Autoencoders”. In: *Proceedings of the 25th International Conference on Machine Learning*. ICML ’08. Helsinki, Finland: Association for Computing Machinery, 2008, pp. 1096–1103. ISBN: 9781605582054. DOI: [10.1145/1390156.1390294](https://doi.org/10.1145/1390156.1390294). URL:
- [WO14] Wang, Z. and Oates, T. “Encoding Time Series as Images for Visual Inspection and Classification Using Tiled Convolutional Neural Networks”. In: 2014.
- [Wei21] Weinberger, K. *Lecture 2: k-nearest neighbors*. 2021. URL:
- [WSS04] Weinberger, K. Q., Sha, F., and Saul, L. K. “Learning a kernel matrix for nonlinear dimensionality reduction”. In: *Proceedings of the twenty-first international conference on Machine learning*. 2004, p. 106.
- [Wol96] Wolpert, D. H. “The lack of a priori distinctions between learning algorithms”. In: *Neural computation* 8.7 (1996), pp. 1341–1390.
- [WM97] Wolpert, D. H. and Macready, W. G. “No free lunch theorems for optimization”. In: *IEEE transactions on evolutionary computation* 1.1 (1997), pp. 67–82.

-
- [Wu+18] Wu, Z. et al. *Unsupervised Feature Learning via Non-Parametric Instance-level Discrimination*. 2018. arXiv: [1805.01978](#) [[cs.CV](#)].
- [Xu+04] Xu, M. et al. “HMM-based audio keyword generation”. In: *Pacific-Rim Conference on Multimedia*. Springer. 2004, pp. 566–574.
- [Yan+19] Yan, X. et al. *ClusterFit: Improving Generalization of Visual Representations*. 2019. arXiv: [1912.03330](#) [[cs.CV](#)].
- [YL00] Yen, G. G. and Lin, K.-C. “Wavelet packet feature extraction for vibration monitoring”. In: *IEEE transactions on industrial electronics* 47.3 (2000), pp. 650–667.
- [Yos+14] Yosinski, J. et al. “How transferable are features in deep neural networks?” In: *Advances in neural information processing systems* 27 (2014).
- [You17] Younes, M. “The case for using digital EEG analysis in clinical sleep medicine”. In: *Sleep Science and Practice* 1.1 (Feb. 2017). DOI: [10.1186/s41606-016-0005-0](#). URL:
- [YK15] Yu, F. and Koltun, V. “Multi-scale context aggregation by dilated convolutions”. In: *arXiv preprint arXiv:1511.07122* (2015).
- [Zbo+21] Zbontar, J. et al. *Barlow Twins: Self-Supervised Learning via Redundancy Reduction*. 2021. arXiv: [2103.03230](#) [[cs.CV](#)].
- [ZF14] Zeiler, M. D. and Fergus, R. “Visualizing and understanding convolutional networks”. In: *European conference on computer vision*. Springer. 2014, pp. 818–833.
- [Zer+20] Zerveas, G. et al. *A Transformer-based Framework for Multivariate Time Series Representation Learning*. 2020. arXiv: [2010.02803](#) [[cs.LG](#)].
- [ZIE16] Zhang, R., Isola, P., and Efros, A. A. *Colorful Image Colorization*. 2016. arXiv: [1603.08511](#) [[cs.CV](#)].
- [ZML17] Zhao, J., Mathieu, M., and LeCun, Y. *Energy-based Generative Adversarial Network*. 2017. arXiv: [1609.03126](#) [[cs.LG](#)].