Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών
Τομέας Μαθηματικών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Μέθοδοι παραγωγής αντιπαραδειγμάτων σε δεδομένα χρονοσειρών»

Στελλα Γεραντώνη

Επιβλέπων Καθηγητής:
Γεώργιος Στάμου

Αθήνα, Μάρτιος 2023

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών
Τομέας Μαθηματικών

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«Μέθοδοι παραγωγής αντιπαραδειγμάτων σε δεδομένα χρονοσειρών»

Στελλα Γεραντώνη

Επιβλέπων : Γεώργιος Στάμου
Αν. Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την .

.....	.....	.....
Γεώργιος Στάμου	Αντώνιος Συμβώνης	Αθανάσιος Βουλόδημος

Αθήνα, Μάρτιος 2023

.....

Στελλα Γεραντώνη

Διπλωματούχος της Σχολής Εφαρμοσμένων Μαθηματικών και Φυσικών Επιστημών

Περίληψη

Η διπλωματική εργασία ασχολείται με αντικειμενικές εξηγήσεις στο πεδίο της μηχανικής μάθησης. Οι αντικειμενικές εξηγήσεις μελετάνε «Τίς απαιτούμενες αλλαγές ώστε ένα δείγμα δεδομένων να κατηγοριοποιείται στην αντίθετη κλάση». Όσο η απαίτηση διαύγειας των νευρωνικών δικτύων αυξάνεται, είτε επειδή υπάρχει φόβος της ανεξέλεγκτης συμπεριφοράς τους, είτε γιατί η κατανόησή τους καθιστά ικανή την σημαντική βελτίωση τους, οι αντικειμενικές εξηγήσεις αποτελούν μία από τις πιο ασφαλής και ακριβής μεθόδους εξηγήσεις της συμπεριφοράς μοντέλων μηχανικής μάθησης.

Η μελέτη που πραγματοποιήθηκε στην συγκεκριμένη εργασία αφορά την μέθοδο LatentCF, ή οποία παράγει αντιπαραδείγματα μέσω του συμπιεσμένου χώρου ενός αυτοκωδικοποιητή και βασίζεται πάνω σε δύο διατριβές, την «Latent-CF: A Simple Baseline for Reverse Counterfactual Explanations» των Rachana Balasubramanian, Sam Sharpe, Brian Barr και C. Bayan Bruss (1) και την «Learning Time Series Counterfactuals via Latent Space Representations» των Zhendong Wang, Rami Mochaourab και Παναγιώτης Παπαπέτρου (2). Συγκεκριμένα μελετά την συμπεριφορά της LatentCF με πολυδιάστατες χρονοσειρές και προτείνεται μία βελτιωμένη εκδοχή της που συγκρίνει τις κατανομές των δεδομένων, μέσω του Kernel Dencity Estimation. Σκοπός αυτής της προσθήκης είναι τα αντιπαραδείγματα να έχουν κοινή κατανομή με τα δεδομένα της αντίθετης κλάσης στην οποία πλέον ανήκουν.

Όλα τα πειράματα έχουν πραγματοποιηθεί με την προγραμματιστική γλώσσα Python (3), έχουν υλοποιηθεί στην πλατφόρμα google colab (4) και είναι διαθέσιμα στο github με URL: https://github.com/stellagerantoni/LatentCfMultivariate.

Λέξεις Κλειδιά: Μηχανική Μάθηση, αυτοκωδικοποιητή, συμπιεσμένη διάσταση, αντιπαραδείγματα, Κατανομή KDE.

Abstract

In this thesis, the primary area of study is counterfactual explanations within the field of Machine Learning. Counterfactual explanations describe “What has to change in the input, so that the output is of the opposite class”. The need for explaining the behavior of machine learning models is becoming more important, because of two reasons. A growing fear of what machine learning models are capable of and the fact that understanding their logic makes their massive improvement possible. Counterfactual explanations are one of the most robust methods for explaining the behavior of the models and thus its improvement is important.

The study is based on LatentCF. A method of generating counterfactual explanations through the latent space of an autoencoder. It is based on two Thesis: «Latent-CF: A Simple Baseline for Reverse Counterfactual Explanations» written by Rachana Balasubramanian, Sam Sharpe, Brian Barr and C. Bayan Bruss (1) and «Learning Time Series Counterfactuals via Latent Space Representations» written by Zhendong Wang, Rami Mochaourab and Panagiotis Papapetrou (2). Specifically, it studies the behavior of LatentCF with multidimensional time series and proposes an improved version that compares the data distributions through Kernel Density Estimation (KDE). The purpose of this addition is to ensure that the counterexamples share a common distribution with the data of the opposite class to which they belong.

Everything has been implemented using the Programming Language Python (3), the platform google colab (4) and the code is available on the github site with URL: https://github.com/stellagerantoni/LatentCfMultivariate.

Keywords: Machine Learning, counterfactual explanations, autoencoder, latent space,

Kernel Density Estimation, KDE.

Ευχαριστίες

Θα ήθελα να εκφράσω τις ευχαριστίες μου σε όλους όσους στάθηκαν πλάι μου και συνέβαλαν στην ολοκλήρωση την διπλωματικής εργασίας. Η καθοδήγηση και η υποστήριξή τους έπαιξε καταλυτικό ρόλο στην εκπόνηση της εργασίας αυτής.

Αρχικά θα ήθελα να εκφράσω την ευγνωμοσύνη μου στον επιβλέπων καθηγητή μου κ. Γιώργο Στάμου που μου έδωσε την ευκαιρία να εμπλακώ στο εργαστήριο της Τεχνητής Νοημοσύνης, και να ερευνήσω το συγκεκριμένο επιστημονικό πεδίο. Επιπλέον, θα ήθελα να ευχαριστήσω τον Γιώργο Φιλανδριανό και τον Κωνσταντίνο Θωμά για την πολύτιμη βοήθεια και καθοδήγησή τους κατά τη διάρκεια της εκπόνησης της διπλωματικής εργασίας. Τέλος θα ήθελα να ευχαριστήσω τους γονείς μου για την στήριξη τους σε όλη τη διάρκεια των σπουδών μου και τους συμφοιτητές και φίλους μου για την συμπαράσταση τους.

Στέλλα Γεραντωνη

Μάρτιο 2023

Contents

1.Εισαγωγή

1.1 Μηχανική Μάθηση και Ερμηνευσιμότητα των Αλγορίθμων

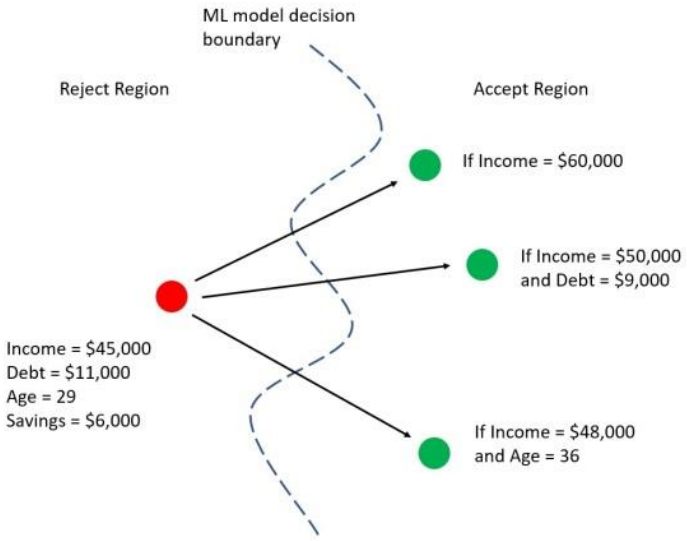
Η μηχανική Μάθηση είναι το επιστημονικό πεδίο το οποίο ασχολείται με την ανάπτυξη αλγορίθμων που δίνουν τη δυνατότητα στους υπολογιστές να λαμβάνουν αποφάσεις χωρίς να χρειαστεί ο άνθρωπος να προγραμματίσει την κάθε κίνηση του υπολογιστή. Ο κάθε αλγόριθμος μαθαίνει να αναγνωρίζει μοτίβα που παρουσιάζονται στα δεδομένα και τα χρησιμοποιεί για να κάνει προβλέψεις.

Οι αλγόριθμοι μηχανικής μάθησης έχουν αποδειχθεί τόσο αποτελεσματικοί τα τελευταία χρόνια που πολύ γρήγορα άρχισαν να διαδραματίζουν μεγάλο ρόλο στη λήψη αποφάσεων στους περισσότερους τομείς της αγοράς. Η ραγδαία ανάπτυξή τους βελτίωσε την αποτελεσματικότητά και την πολυπλοκότητά τους σε βαθμό που ο άνθρωπος δεν κατανοεί την ακριβή διαδικασία που ακολουθούν. Έτσι τους δόθηκε η ονομασία “black boxes” η οποία αναδεικνύει το πρόβλημα αυτό και πυροδότησε την ανάγκη ανάπτυξης μεθόδων που θα προσπαθούν να αναλύσουν και να εξηγήσουν τι συμβαίνει μέσα στα νευρωνικά δίκτυα.

Η ανάγκη για την ανάπτυξη τέτοιων μεθόδων προήρθε από περιπτώσεις όπου η μη διαφάνεια των νευρωνικών δικτύων αποδείχθηκε καταστροφική. Σε ένα περιστατικό που εκτυλίχθηκε στις ένοπλες δυνάμεις των ΗΠΑ (5), αναπτύχθηκε ένας ταξινομητής με τον σκοπό να διακρίνει εχθρικά και φιλικά τανκς, ενισχύοντας έτσι την ακρίβεια και την ταχύτητα της απόκρισης σε εχθρικές απειλές. Φαινομενικά, κατά την αξιολόγηση του ταξινομητή, η απόδοση του ήταν αρκετά ακριβής. Όμως, η χρήση του σε πραγματική πολεμική συνθήκη απέδειξε το αντίθετο. Δεν αναγνώριζε καμία διαφορά στα τανκς που βρίσκονταν στο πεδίο της μάχης. Όταν έγινε η προσπάθεια να κατανοηθεί αυτή η συμπεριφορά του ταξινομητή, ανακαλύφθηκε ότι κατά την εκπαίδευση του οι εικόνες που λάμβανε ως είσοδο και απεικόνιζαν τα τεθωρακισμένα της αντίπαλης πλευράς, είχαν ληφθεί σε ημέρες κακοκαιρίας με περιορισμένο φως, σε αντίθεση με αυτές της φιλικής πλευράς. Ο ταξινομητής λοιπόν

λάμβανε αποφάσεις με βασικό κριτήριο τον καιρό και δεν επεξεργαζόταν τα ίδια τα τανκς.

Ωστόσο, η διαφάνεια της λήψης αποφάσεων μπορεί να φανεί χρήσιμη στον άνθρωπο για πολλούς λόγους. Δίνεται ένα ακόμη παράδειγμα όπου μια τράπεζα χρησιμοποιεί τεχνητή νοημοσύνη για τη χορήγηση δανείων. οι λόγοι για του οποίους γίνεται δεκτό η απορρίπτεται ένα δάνειο ενδιαφέρουν τόσο την τράπεζα όσο και τον άνθρωπο (η την εταιρεία) που κάνει την αίτηση για το δάνειο. Η τράπεζα από την πλευρά της μπορεί να ελέγξει αν η απόφαση που πήρε ήταν ορθή, δίκαιη και αν συμμορφώνεται με τους κανονισμούς των μηχανισμών, ενώ ο αιτούμενος μπορεί ανακαλύπτοντας το «γιατί» η αίτησή του απορρίφθηκε να δράσει διαφορετικά έτσι ώστε να γίνει δεκτή μια παρόμοια αίτηση στο μέλλον. Έτσι παρουσιάζονται οι ερμηνείες με αντιπαραδείγματα ως ένας αποτελεσματικός τρόπος να αποκτηθεί αυτή η διαύγεια.



Εικόνα 2.1: Ένα μοντέλο που παράγει «λύσεις» προς τον δανειολήπτη για την απόρριψη της αίτησης του για δάνειο.

1.2 Μέθοδος ερμηνείας με αντιπαραδείγματα.

(Counterfactual explanations)

Οι ερμηνείες με αντιπαραδείγματα υποδεικνύουν τί πρέπει να αλλάξει στα χαρακτηριστικά των δεδομένων εισόδου έτσι ώστε να αλλάξει η κλάση που επιστρέφεται από την έξοδο του ταξινομητή.

Ο Riccardo Guidotti (6) σε συμφωνία με τον Molnar (7) έδωσε ορισμό για την ερμηνεία με αντιπαραδείγματα:

ΟΡΙΣΜΟΣ:

Ονομάζουμε έναν ταξινομητή b, ένα παράδειγμα των δεδομένων εισόδου ως x και την απόφαση που παίρνει ο ταξινομητής ως y. Έτσι έχουμε την απόφαση του ταξινομητή να ισούται με y=b(x). Μια ερμηνεία με αντιπαράδειγμα αποτελείται από ένα καινούργιο παράδειγμα x′ τέτοιο ώστε να απόφαση του b για το x′ να είναι διαφορετική από το y. Ορίζεται έτσι : $b(x') \neq y$, τέτοιο ώστε η διαφορά του x και x′ να είναι ελάχιστη.

Οι ερμηνείες με αντιπαραδείγματα στη μηχανική μάθηση είναι φιλικές προς τον άνθρωπο διότι αντικατοπτρίζουν τον τρόπο με τον οποίο σκέφτεται και μαθαίνει. Καθώς οι άνθρωποι αντιμετωπίζουν καθημερινά προκλήσεις και λαμβάνουν αποφάσεις, συχνά φαντάζονται πώς θα μπορούσαν να έχουν αλλάξει τα πράγματα αν είχαν ενεργήσει διαφορετικά. Αυτός ο τρόπος σκέψης έχει μελετηθεί και από άλλες επιστήμες όπως η φιλοσοφία, η ψυχολογία και η κοινωνιολογία. Οι ερμηνείες με αντιπαραδείγματα προσφέρουν τη δυνατότητα να αλλάξει το αποτέλεσμα.

Επιστρέφοντας στο παράδειγμα με την αίτηση για δάνειο γίνεται παράθεση όλων των τρόπων με τους οποίους οι ερμηνείες με αντιπαραδείγματα μπορούνε να διαλευκάνουν την απόφαση του ταξινομητή αυτό όπως αναφέρεται και στο (5) για να κατανοηθεί η ανάγκη μελέτης των αντιπαραδειγματικών εξηγήσεων.

1) Η εξήγηση μπορεί να φανεί σημαντικά χρήσιμη για το δέκτη. Υπάρχουν περιπτώσεις όπου είναι σημαντικό να γνωρίζει ποια χαρακτηριστικά

επέδρασσαν καθοριστικά στην περίπτωση που γίνεται δεκτό το αίτημα ώστε να μπορεί να κατανοήσει τα δυνατά του χαρακτηριστικά.

2) Οι ερμηνείες με αντιπαραδείγματα μπορούν να βελτιώσουν την εμπιστοσύνη του ταξινομητή.
3) Αν ο δέκτης από τα αντιπαραδείγματα ανακαλύψει ότι έχει απορριφθεί το αίτημα του για λόγους που δεν θεωρεί δίκαιους (για παράδειγμα το να είναι μετανάστης) μπορεί να κάνει ένσταση.
4) Ο Αιτούντα μαθαίνοντας τους λόγους που απορρίφθηκε η αίτηση του, μπορεί να ενεργήσει αντίστοιχα ώστε να έχει ευκαιρία να γίνει δεκτή στο μέλλον.
5) Οι ερμηνείες με αντιπαραδείγματα μπορούν να εντοπίσουν λάθη στη διαδικασία λήψης αποφάσεων του ταξινομητή και να καταστήσουν δυνατή την αντιμετώπιση των αδύναμων του σημείων.
6) Ορισμένες κυβερνήσεις έχουν εγκρίνει νόμους όπως ο Γενικός Κανονισμός για την Προστασία των Δεδομένων (GDPR) της Ευρωπαϊκής Ένωσης, προκειμένου να παρέχουν το δικαίωμα πληροφόρησης σχετικά με το πώς λειτουργούν αυτόματα συστήματα λήψης αποφάσεων. Ο νόμος ενθαρρύνει τους δημιουργούς ταξινομητών να αποκτούν διαφάνεια γύρω από τα συστήματα αυτά. (8)

1.3 Τοπικές αντιπαραδειγματικές εξηγήσεις

Οι αντικειμενικές εξηγήσεις που δίνονται στην συγκεκριμένη έρευνα είναι τοπικές. Αυτό σημαίνει πως η εξήγηση που δίνεται αναφέρεται αποκλειστικά στο συγκεκριμένο δεδομένο σε αντίθεση με τις παγκόσμιες εξηγήσεις που είναι γενικές και σκοπός τους είναι να δοθεί εξήγηση για όλη την κλάση.

Θα ορίσουμε τις Τοπικές Εξηγήσεις σύμφωνα με την διατριβή (9).

Ορισμός: Τοπική Εξήγηση σε Χρονοσειρές

Σε ένα σύστημα ανάλυσης χρονοσειρών, μια τοπική εξήγηση εστιάζει στο να προσδιορίσει τη λογική πίσω από την απόφαση του ταξινομητή για ένα συγκεκριμένο δείγμα μέσα στο σύνολο δεδομένων. Η τοπική εξήγηση αποσκοπεί στην ανάδειξη των λεπτομερειών αυτής της διαδικασίας, αποκαλύπτοντας τις συγκεκριμένες

χαρακτηριστικές ιδιότητες ή παράμετρους που οδήγησαν στην απόφαση του ταξινομητή για το δοθέν δείγμα.

1.4 Τρόπος εύρεσης ερμηνείας με αντιπαραδείγματα

Μία απλή απάντηση στο πώς βρίσκονται ερμηνείες με αντιπαραδείγματα είναι εύκολη (7). Αρκεί να αλλάζουν τα χαρακτηριστικά των δεδομένου εισόδου με τυχαίο τρόπο επαναλαμβανόμενα μέχρι οι αλλαγές αυτές να πετύχουν το ζητούμενο. Μέχρι δηλαδή ο ταξινομητής να το θεωρεί δεδομένο της αντίθετης κλάσης. Το ζήτημα όμως είναι ότι αν δεν υπάρχει κάποια κατεύθυνση προς την οποία θα κινηθεί το μοντέλο είναι προφανές ότι αυτή η διαδικασία μπορεί να πάρει πολύ καιρό και να μην είναι αποτελεσματική. Έτσι ορίζεται η συνάρτηση απώλειας που υπολογίζει την απόσταση μεταξύ του δεδομένου και κάποιου επιθυμητού αντιπαραδείγματος. Με βάση αυτή την συνάρτηση απώλειας λοιπόν κινείται το μοντέλο με σκοπό να την μειώσει όσο περισσότερο δυνατόν. Όμως ο χώρος στον οποίο έχει τη δυνατότητα να ψάχνει μέχρι να μειώσει την συνάρτηση απώλειας συνεχίζει να είναι μεγάλος. Έτσι προτείνεται από τους *Rachana Balasubramanian, Sam Sharpe, Brian Barr, C. Bayan Bruss* (1) να γίνεται η εύρεση αντιπαραδειγμάτων στον συμπιεσμένο χώρο ενός αυτοκωδικοποιητή για μεγαλύτερη αποτελεσματικότητα και λιγότερο χρονοβόρα απόδοση από ότι προσέφεραν οι προ υπάρχουσες μέθοδοι.

2. Θεωρητικό Μέρος

2.1 Κατηγορίες Τεχνητής Μηχανικής Μάθησης

Στην Μηχανική Μάθηση κατατάσσονται όλοι οι αλγόριθμοι που αναπτύσσονται με σκοπό την επίλυση προβλημάτων και περιλαμβάνουν την εκμάθησή των χαρακτηριστικών ενός σετ δεδομένων χωρίς να απαιτείται η ρητή προγραμματιστική οδηγία για την επίτευξη αυτού.

Βασικές κατηγορίες μοντέλων Μηχανική Μάθησης είναι η επιβλεπόμενη και η μη επιβλεπόμενη.

2.1.1 Επιβλεπόμενη Μηχανική Μάθηση

Βασικό χαρακτηριστικό της επιβλεπόμενης Μάθησης (10) είναι οι ταμπέλες που δίνονται μαζί με τα δεδομένα για καθοδήγηση. Ο αλγόριθμος κατά την εκπαίδευση κατανοεί τη δομή και τις σχέσεις του συνόλου δεδομένων, με απώτερο σκοπό την εκμάθηση της σχέσης εισόδου-εξόδου. Αφού ολοκληρωθεί η εκπαίδευση, ο αλγόριθμος θα έχει την ικανότητα να λαμβάνει νέα, απροσδιόριστα δεδομένα εισόδου και να προβλέπει τα αντίστοιχα αποτελέσματα, διευκολύνοντας την εφαρμογή του σε διάφορες προκλήσεις, όπως η πρόβλεψη σειρών δεδομένων. Αυτή η διαδικασία είναι καθοριστική για την κατανόηση και την ερμηνεία ευρύτερων μοτίβων και δυναμικών στα δεδομένα, παρέχοντας έτσι τη βάση για πιο προηγμένες αναλύσεις και εφαρμογές.

2.1.2 Μη επιβλεπόμενη Μηχανική Μάθηση

Η μη επιβλεπόμενη μάθηση (11) είναι τα μοντέλα που, σε αντίθεση με τους ταξινομητές όπου περιλαμβάνουν τα δεδομένα εισόδου με τις αντίστοιχες κλάσεις τους, δεν έχουν ετικέτες για καθοδήγηση. Η βασική τους λειτουργεία έχει σκοπό την ομαδοποίηση (clustering) εντοπίζοντας τα κοινά χαρακτηριστικά των δεδομένων και κατατάσσοντας τα σε κατηγορίες με βάση αυτά τα χαρακτηριστικά. Υπάρχουν όμως διάφορα είδη μοντέλων μη επιβλεπόμενης μάθησης και αναλόγως η λειτουργία διαφέρει.

Έτσι δημιουργείται μια συνάρτηση f που μπορεί να εκφράζει τα δεδομένα και με βάση τα κοινά χαρακτηριστικά τους να τα ομαδοποιεί. Αυτή η διαδικασία είναι πιο

περίπλοκη από την επιβλεπόμενη μάθηση καθώς δεν υπάρχει καμία καθοδήγηση λόγο την μη ύπαρξης ετικετών. Το μοντέλο ομαδοποιεί με βάση τα κοινά χαρακτηριστικά που θα παρατηρήσει χωρίς να υπάρχει πάντα κάποια ρητή οδηγία.

Στα επόμενα κεφάλαια αναπτύσσονται λεπτομερώς οι Ταξινομητές που κατατάσσονται στην επιβλεπόμενη Μηχανική Μάθηση και οι Αυτοκωδικοποιητές που κατατάσσονται αντίστοιχα στην μη επιβλεπόμενη, καθώς αυτά τα δύο μοντέλα χρησιμοποιούνται εκτενώς στο πειραματικό μέρος.

Πριν την ανάλυση αυτών των δύο κρίνεται σκόπιμο η ανάπτυξη της θεωρίας των νευρώνων καθώς είναι βασικός πυλώνας της λειτουργίας όλων των προαναφερόμενων μοντέλων.

2.2 Νευρώνες

Οι νευρώνες αποτελούν τη δομική μονάδα των νευρωνικών δικτύων, όπως φαίνεται από το όνομα. Οι νευρώνες σαν όνομα αλλά και σαν λειτουργία είναι εμπνευσμένοι από τον ίδιο τον ανθρώπινο εγκέφαλο και επιτρέπουν την επεξεργασία των δεδομένων με πολύ περίπλοκο τρόπο που οι άνθρωποι δεν είναι δυνατόν να ακολουθήσουν τη ροή της εκμάθησης ή της επεξεργασίας ενός ακόμα δεδομένου. Έτσι προκύπτει και το πρόβλημα των «black boxes».

Οι νευρώνες συνδέονται μέσω συναρτήσεων με τους υπόλοιπους νευρώνες, τα δεδομένα εισόδου και τα δεδομένα εξόδου. Οι συναρτήσεις αυτές καθορίζονται από το ίδιο το μοντέλο κατά την εκπαίδευση ,που μαθαίνει τις πιο ικανοποιητικές συναρτήσεις για τα χαρακτηριστικά των δεδομένων, αλλά και από το είδος του μοντέλου, συνελικτικό ή LSTM, και τις συναρτήσεις ενεργοποίησης που αναλύονται παρακάτω.

Τα τρία βασικά χαρακτηριστικά των νευρώνων είναι (12):

1. **Βάρη [Wij]:** Τα βάρη αποκτούν μια διαφορετική τιμή για κάθε ζεύγος νευρώνων (εκφράζεται έτσι η «σχέση» μεταξύ τους) και πολλαπλασιάζονται με το δεδομένο πριν αυτό εισέλθει στον ίδιο το νευρώνα.

2. **Η αθροιστική συνάρτηση [Sj]** : Είναι μια συνάρτηση που αθροίζει τα σήματα εισόδου πολλαπλασιασμένα με τα βάρη.
3. **Συνάρτηση Ενεργοποίησης (Activation Function) (Oj):** Ορίζει πεπερασμένα όρια στα σήματα εξόδου συνήθως από τιμές [0,1] ή [-1,1].
4. **Bias [b]:** Το bias είναι μια τιμή που αποτελεί εξωτερική πόλωση. Προστίθεται στην αθροιστική συνάρτηση με σκοπό την αύξηση ή την μείωση της διέγερσης της συνάρτησής ενεργοποίησης.

Η μαθηματική σχέση της αθροιστικής συνάρτησης (13):

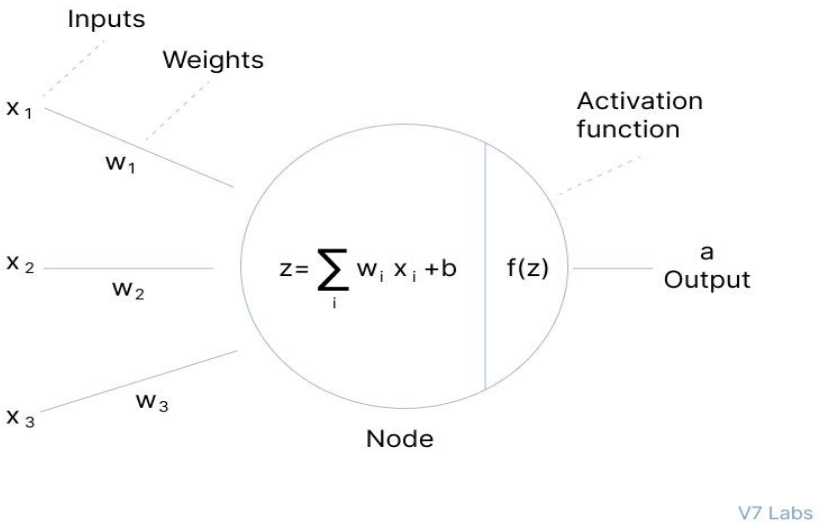
$$s_j = \sum_{i=1}^{n} x_i w_{ij}$$

Όπου x_i όλα τα σήματα που εισέρχονται στον συγκεκριμένο νευρώνα j, n το σύνολο των σημάτων εισόδου, w_{ij} τα βάρη που πολλαπλασιάζονται με το σήμα εισόδου i στον νευρώνα j.

Η σχέση της Τελικής συνάρτησης εξόδου:

$$f_j = O(S_j + b_j)$$

Όπου O η συνάρτηση εξόδου και b_j το bias του νευρώνα.



Εικόνα αριθμός: Μια αναπαράσταση της λειτουργίας του νευρώνα (14)

2.3 Συνάρτηση ενεργοποίησης (Activation Function)

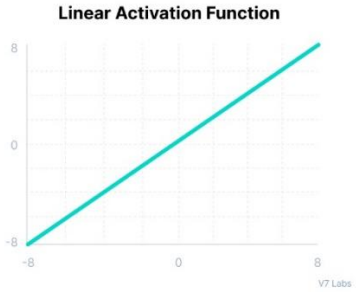
Ένα ακόμα πολύ βασικό κομμάτι των μοντέλων μηχανικής μάθησης είναι οι συναρτήσεις ενεργοποίησης. Οι συναρτήσεις ενεργοποίησης είναι εκείνες που σε κάθε επίπεδο του μοντέλου αποφασίζουν αν θα ενεργοποιηθεί ο νευρώνας η όχι. Μετατρέπει τα δεδομένα εισόδου του νευρώνα που είναι πολλαπλασιασμένα με τα βάρη σε δεδομένα εξόδου για τον επόμενο νευρώνα η για την έξοδο του μοντέλο ολοκληρωτικά. (14)

Στην συνέχεια ακολουθούνε οι συναρτήσεις ενεργοποίησης οι οποίες είναι απαραίτητες για το χτίσιμο των μοντέλων που με τη σειρά τους χρειάζονται για το πειραματικό μέρος της εργασίας:

Linear Activation Function (Γραμμική Συνάρτηση Ενεργοποίησης)

Η Γραμμική συνάρτηση ενεργοποίησης, επιστρέφει ως έξοδο την αντίστοιχη είσοδο. Ασχέτως με τα βάρη που έχουν δοθεί σε κάθε χαρακτηριστικό των δεδομένων εδώ η έξοδος είναι αυτούσια ή είσοδος.

$$f(x) = x$$



Σχήμα (αριθμός) : Γραφική παράσταση της γραμμικής συνάρτησης ενεργοποίησης.

Sigmoid/ Logistic

Η συνάρτηση ενεργοποίησης Sigmoid αλλάζει κλίμακα στα δεδομένα. Η έξοδος παίρνει πάντα τιμές από μηδέν μέχρι ένα. Όσο πιο μεγάλο το στοιχείο εισόδου, τόσο πιο κοντά στο 1 η έξοδος και αντίστοιχα όσο πιο μικρό το στοιχείο εισόδου τόσο πιο κοντά στο 0 η έξοδος. Η συνάρτηση αυτή χρησιμοποιείται συνήθως στους τελικούς νευρώνες δυαδικών ταξινομητών και για κάθε κλάση επιστρέφετε μία τιμή από την σιγμοειδή συνάρτηση η οποία αποτελεί την πιθανότητα το δεδομένο να ανήκει στην κλάση αυτή.

$$f(x) = \frac{1}{1 + e^{-x}}$$

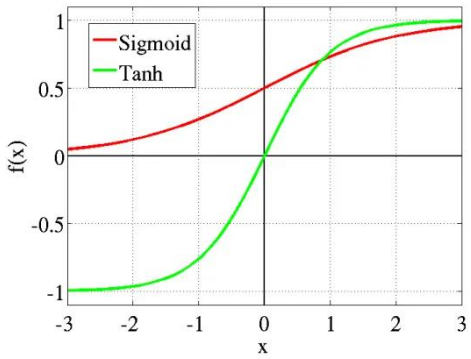
Εικόνα (αριθμός) : Η γραφική παράσταση της Σιγμοειδής συνάρτησης ενεργοποίησης.

Tanh

Η Tanh (Hyperbolic Tangent Function) είναι μία συνάρτηση ενεργοποίησης που στην εργασία αυτή χρησιμοποιείται στους νευρώνες του LSTM μοντέλου. Μοιάζει πάρα πολύ με την σιγμοειδής συνάρτηση ενεργοποίησης όμως επιστρέφει τιμές στο διάστημα [-1,1] (σε αντίθεση με το διάστημα [0,1] της σιγμοειδούς). (15)

$$f(x) = 2\sigma(2x) - 1$$

Όπου σ είναι η σιγμοειδής συνάρτηση.

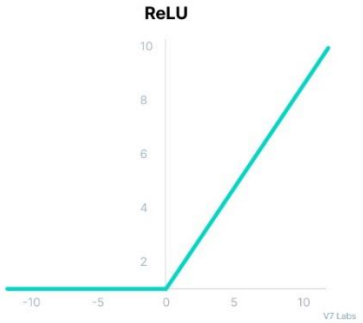


Εικόνα αριθμός: Στην εικόνα απεικονίζεται η σιγμοειδής και η tanh συνάρτηση.

ReLu (Rectified Linear Unit)

Η συνάρτηση ReLu έχει καταστεί εξαιρετικά δημοφιλής καθώς αποδεικνύεται αποδοτική σε ποικίλα μοντέλα βαθιάς μάθησης. Εφαρμόζει γραμμική ενεργοποίηση όταν η είσοδος είναι θετική και μηδενική όταν η είσοδος είναι αρνητική. Είναι γρήγορη και υπολογιστικά αποδοτική για τους παρακάτω λόγους. Αρχικά όταν κάποιοι νευρώνες επιστρέφουν μηδέν αποκρύπτεται ένα μεγάλο μέρος της πληροφορίας το οποίο καθιστά την διαδικασία πιο γρήγορη. Επιπλέον οι συναρτήσεις που δίνουν τιμές από το διάστημα (0,1) καθυστερούν την διαδικασία, λόγο της παραγοντοποίησες με πολύ μικρούς αριθμούς. Η ReLu προχωράει πιο γρήγορα καθώς έχει την ικανότητα με λίγες επαναλήψεις να μεγαλώσει τους παραγώγους και να εκκινήσει τη διαδικασία εκπαίδευσης.

$$f(x) = \max(0, x)$$



Σχήμα (αριθμός) : Γραφική παράσταση της συνάρτησης ενεργοποίησης Relu.

Softmax

Η Softmax είναι αντίστοιχη της Sigmoid αλλά για προβλήματα περισσότερων κλάσεων (Ενώ η Sigmoid απευθύνεται σε δυαδικά). Χρησιμοποιείται στο τελευταίο νευρώνα ενός μοντέλου που έχει ως στόχο να ταξινομήσει τα δεδομένα σε πολλές κλάσεις. Παράγει πιθανότητα για την κάθε κλάση και εκφράζεται από τον τύπο:

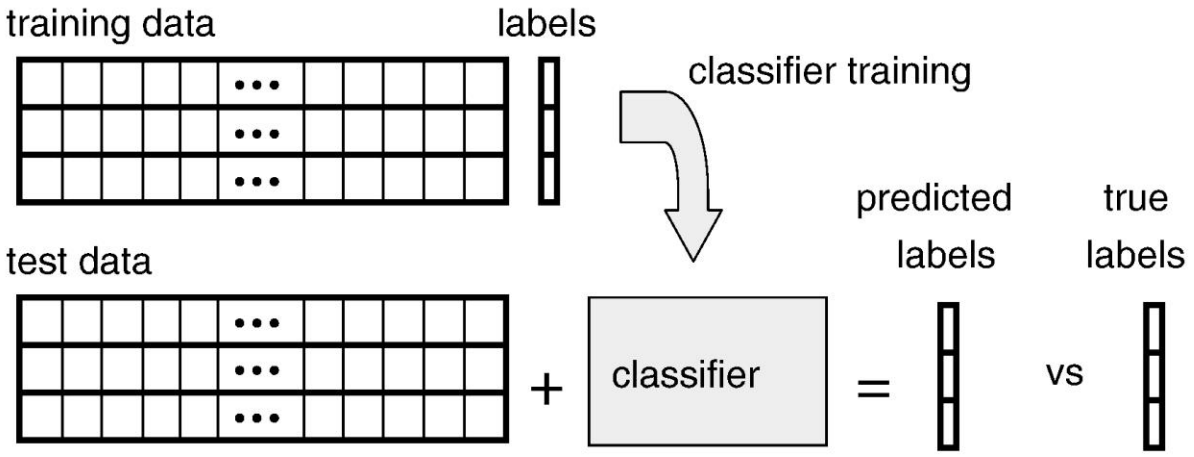
$$softmax(z_i) = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}}, για\ i = 1,2, \dots, K$$

Όπου K είναι το πλήθος κλάσεων. Ένας νευρώνας που ακολουθεί αυτή τη συνάρτηση δέχεται ένα δεδομένο και επιστρέφει έναν πίνακα διαστάσεων 1xK που περιέχει πιθανότητες για κάθε κλάση και όποια κλάση έχει την μεγαλύτερη πιθανότητα είναι εκείνη στην οποία ανήκει το δεδομένο. Το άθροισμα όλων των πιθανοτήτων ισούται με ένα.

2.4 Ταξινομιτές

Ταξινομητής (16), κατά την επιβλεπόμενη μάθηση, ορίζεται η διαδικασία που δέχεται ως είσοδο τις τιμές των διαφόρων χαρακτηριστικών ενός δείγματος και προβλέπει την κλάση στην οποία ανήκει το δείγμα. Ένα παράδειγμα ταξινόμησης μπορεί να έχει ώς δείγμα τα οικονομικά και κοινωνικά χαρακτηριστικά πολλών ατόμων και οι κατηγορίες να είναι αν θα γίνει δεκτό το αίτημα τους για δάνειο ή όχι. Ένα δείγμα θα συμβολίζεται με το διάνυσμα γραμμών X = [x1…xn] και η κατηγορία του με την ετικέτα y. Ο ταξινομητής διαθέτει μια σειρά παραμέτρων που καθορίζονται μέσα από τα δεδομένα εκπαίδευσης. Ο ταξινομητής που έχει εκπαιδευτεί αποτελεί ουσιαστικά ένα μοντέλο που περιγράφει τη σχέση μεταξύ των χαρακτηριστικών και της ετικέτας στο εκπαιδευτικό σύνολο. Ειδικότερα, για ένα δείγμα x, ο ταξινομητής είναι μια συνάρτηση f που προβλέπει την ετικέτα.

Ένας ταξινομητής κατά την εκπαίδευση του επαναλαμβάνει μία ίδια διαδικασία πολλαπλές φορές. Αυτή η διαδικασία εμπεριέχει την επεξεργασία των δεδομένων εκπαίδευσης με σκοπό τη δημιουργία της συνάρτησης f, την διαδικασία evaluation κατά την οποία εκτιμάει αν η συνάρτηση ικανοποιεί όντως τα δεδομένα και την βελτιστοποιεί αναλόγως και την διαδικασία testing κατά την οποία εκτιμάται η αποτελεσματικότητα του με καινούργιο σετ δεδομένων. Η επιτυχία της διαδικασίας αυτής κρίνεται από το accuracy που είναι το ποσοστό που εκφράζει πόσα δείγματα ταξινομήθηκαν σωστά. Όταν είναι εκπαιδευμένος, αν η διαδικασία έχει γίνει αποτελεσματικά, είναι σε θέση να ταξινομήσει δεδομένα ίδιας φύσης που δεν έχει ξαναδεί. Έτσι στην εργασία αυτή ο ταξινομητής χρησιμοποιείται για να εκτιμάει την κλάση που ανήκει το δεδομένο που υπόκειται αλλαγές από τον αυτοκωδικοποιητή με σκοπό να περάσει στην άλλη κλάση και να αποτελεί πλέον αντιπαράδειγμα.



Εικόνα αριθμός: Η βασική διαδικασία εκπαίδευσης ενός ταξινομιτή.

Στην εικόνα απεικονίζεται γλαφυρά η διαδικασία κατά την οποία δίνονται στον ταξινομιτή τα σετ δεδομένων εκπαίδευσης μαζί με τις ταμπέλλες (μέσα στα οποία βρίσκεται και το validation σετ δεδομένων) και εκπαιδεύεται ο ταξινομητής. Έπειτα δίνονται τα testing σετ δεδομένων και αφού ο ταξινομητής κάνει προβλέψεις, που κατατάσσονται στον πίνακα predicted labels, συγκρίνονται με τις πραγματικές ταμπέλλες των δεδομένων και υπολογίζεται το ποσοστό του validation που ορίζει πόσο επιτυχημένο είναι το μοντέλο.

2.5 Αυτοκωδικοποιητές

Ένας αυτοκωδικοποιητής, η πιο συνηθισμένα στα αγγλικά (autoencoder), είναι ένα νευρωνικό δίκτυο που λειτουργεί χωρίς επίβλεψη. Αυτό σημαίνει πως δεν δίνονται ετικέτες στο μοντέλο ως στόχος εκπαίδευσης (πχ. Spam η not Spam) αλλά, αντιθέτως, με τα δεδομένα και μόνο το μοντέλο έχει την δυνατότητα να εξάγει αποτελέσματα παρατηρώντας μοτίβα η κρυμμένες δομές. Τα δεδομένα εισόδου είναι τα ίδια με τα δεδομένα εξόδου.

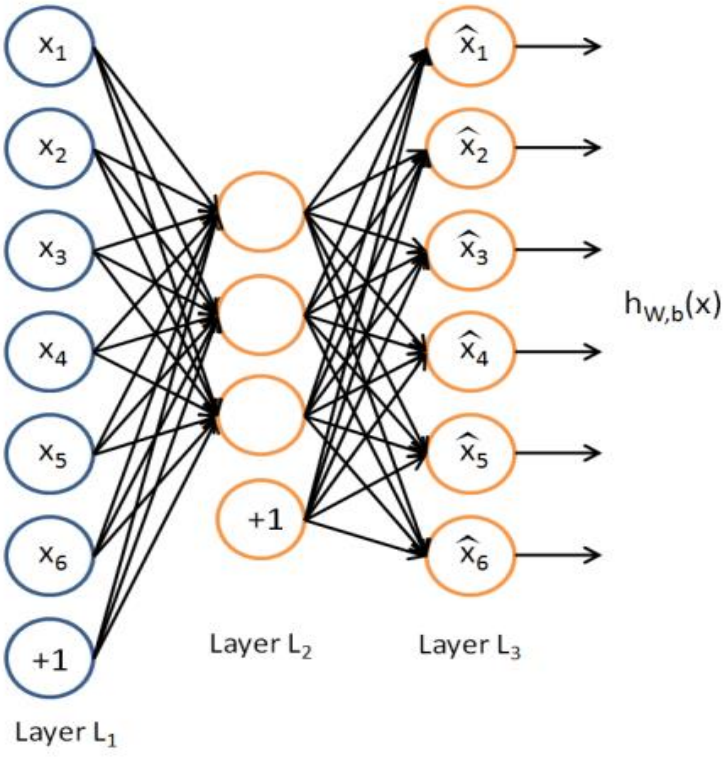
Ο βασικός στόχος του αυτοκωδικοποιητή είναι να δημιουργήσει μια συμπιεσμένη μορφή των δεδομένων η οποία ενέχει μόνο τις σημαντικές πληροφορίες και στην πορεία να αποσυμπιέσει τα δεδομένα με σκοπό να τα ξαναχτίσει όσο πιο κοντά γίνεται στην αρχική τους μορφή. Συνεπώς, ο αυτοκωδικοποιητής εκπαιδεύεται στις σημαντικές πληροφορίες των δεδομένων, με αποτέλεσμα η συμπιεσμένη μορφή να διατηρεί τα μοτίβα και τις πληροφορίες που στην πραγματικότητα «χαρακτηρίζουν» το δεδομένο.

Έτσι ορίζονται τα τρία μέρη του αυτοκωδικοποιητή:

Ο Κωδικοποιητής (encoder): που αποτελείται από νευρώνες, και είναι υπεύθυνος για την συμπίεση των διαστάσεων των δεδομένων με τέτοιο τρόπο ώστε να διατηρηθούν οι σημαντικές πληροφορίες. Σαν είσοδο λαμβάνει τα αρχικά δεδομένα και τα επιστρέφει σε συμπιεσμένη μορφή.

Ο Χώρος ενδιάμεσης Αναπαράστασης (Latent Space Representation) ή κρυφός χώρος ονομάζεται η συμπυκνωμένη αναπαράσταση που περιέχει τις σημαντικές πληροφορίες των δεδομένων. Αποτελεί την έξοδο του κωδικοποιητή η οποία δίνεται σαν είσοδος στον αποκωδικοποιητή.

Ο Αποκωδικοποιητής (Decoder): λαμβάνει σαν είσοδο τον χώρο ενδιάμεσης αναπαράστασης και προσπαθεί, περνώντας τα δεδομένα από νευρώνες, να ανακτήσει ή να αναπαραστήσει τα αρχικά δεδομένα εισόδου. Στόχος του είναι να παράξει εξόδους που είναι όσο το δυνατόν πιο κοντά στα αρχικά δεδομένα εισόδου.



Εικόνα: (17) Στο παράδειγμα αυτό δίνονται στο μοντέλο δεδομένα {Δ(1), Δ(2), ...} όπου το κάθε δεδομένο είναι της μορφής Δ(i) = $\{x^{(1)}, x^{(2)}, x^{(3)}, \dots, x^{(6)}\}$ και το κάθε $x^{(\iota)} \in \mathbb{R}^n$.

Σύμφωνα με το εκπαιδευτικό υλικό : (17) η ουσία του αυτοκωδικοποιητή είναι να μάθει την συνάρτηση

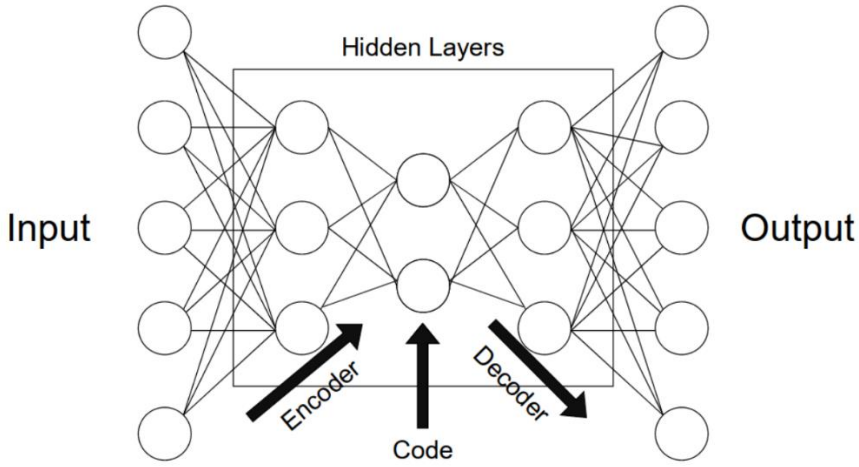
$$h_{W,b}(\mathrm{x}) \approx \mathrm{x}$$

Η ταυτοτική αυτή συνάρτηση μοιάζει απλή για να την υλοποιήσει κάποιο μοντέλο αλλά μέσω της επιβολής περιορισμών του δικτύου στο κρυφό επίπεδο, γίνεται όσο περίπλοκη είναι η δομή των δεδομένων.

Ας υποτεθεί ότι τα δεδομένα είναι χρονοσειρά με 100 στιγμές. Στο ενδιάμεσο επίπεδο, το οποίο αποτελεί την συμπιεσμένη αναπαράσταση των δεδομένων, θα υπάρχουν 50 στιγμές. Στο τρίτο και τελικό επίπεδο θα πρέπει να ξαναχτίσει δεδομένα, ώστε να φτάσει πάλι τις 100 στιγμές, όμοια με το αρχικό. Αν τα 100 σημεία στο χρόνο του δεδομένου ήταν τελείως τυχαία, λόγου χάρη, κάθε $x^{(i)}$ προέρχεται από μια Gaussian κατανομή ανεξάρτητα από τα άλλα χαρακτηριστικά, δεν θα μπορούσε το

μοντέλο από τα μισά δεδομένα να ξαναχτίσει κάτι που μοιάζει με το αρχικό. Αν όμως υπάρχει συσχέτιση μεταξύ των δεδομένων τότε γίνεται εφικτό να αναγνωρίσει ο αλγόριθμος κομμάτια της συσχέτισης αυτής και να ξαναχτίσει τα δεδομένα.

Ένα απλουστευμένο παράδειγμα αυτοκωδικοποιητή λειτουργεί παρόμοια με την μέθοδο Ανάλυσης Κυρίων Συνιστωσών (PCA), καθώς οι γραμμικοί συνδυασμοί που ανιχνεύουν τις κύριες κατευθύνσεις της διακύμανσης στα δεδομένα υπολογίζονται με παρόμοιο τρόπο στις δυο μεθόδους.



Εικόνα 2: (18) Πολύπλοκη μορφή τις διαδικασίας κωδικοποίησης και αποκωδικοποίησης.

Η χρησιμότητα της συμπιεσμένης διάστασης

Σε αυτήν την υπό ενότητα εξηγείται γιατί είναι χρήσιμο και αποτελεσματικό η γίνεται η εύρεση αντιπαραδειγμάτων στο χώρο της ενδιάμεσης αναπαράστασης ενός αυτοκωδικοποιητή.

Τα αντιπαραδείγματα βελτιώνουν την κατανόησή για τα μοντέλα μηχανικής μάθησης, δείχνοντας πώς μια είσοδος μπορεί να αλλάξει για να επιτευχθεί η επιθυμητή πρόβλεψη. Σε αυτό το πλαίσιο, ο ενδιάμεσος χώρος των αυτοκωδικοποιητών μπορεί να είναι πολύτιμος. Περιλαμβάνει την ουσία των

δεδομένων εισόδου σε συμπαγή μορφή, επιτρέποντάς την ουσιαστική και δραστική αλλαγή στα χαρακτηριστικά τους, καθώς έχουμε απαλλαγεί από την «περιττή» πληροφορία, έτσι ώστε να αλλάξουν κλάση.

Για τη δημιουργία μιας αντιπαραδειγματικής εξήγησης, το αρχικό βήμα είναι η κωδικοποίηση των δεδομένων σε αυτόν τον ενδιάμεσο χώρο. Στη συνέχεια, εισάγονται παρεμβάσεις σε αυτή την κωδικοποιημένη αναπαράσταση. Αποκωδικοποιώντας αυτές τις τροποποιημένες αναπαραστάσεις, μπορούμε να παράγουμε παραλλαγές δεδομένων. Δοκιμάζοντας αυτές τις παραλλαγές στο μοντέλο, μπορούν να προσδιοριστούν εκείνες που οδηγούν στην επιθυμητή αντιπαραδειγματική πρόβλεψη.

Αυτή η μέθοδος προσφέρει διάφορα πλεονεκτήματα. Πρώτον, είναι αποτελεσματική, καθώς επικεντρώνεται σε έναν συμπιεσμένο χώρο, αντί για τον τεράστιο χώρο όλων των πιθανών εισόδων. Δεύτερον, η εγγενής συνέχεια αυτού του χώρου εξασφαλίζει ότι οι μικρές αλλαγές παράγουν ρεαλιστικές μετατοπίσεις στα δεδομένα, καθιστώντας τις παραγόμενες αντιπαραδειγματικές εξηγήσεις πιο κατανοητές.

Είδη νευρωνικών δικτύων

Ακολουθεί μια ανάλυση των συνελικτικών στρωμάτων (Convolutional Neural Networks, CNN), και των LSTM (Long Short Term Memory) στρωμάτων καθώς χρησιμοποιούνται εκτενώς στους ταξινομητές και αυτοκωδικοποιητές του πειραματικού μέρους.

Συνελικτικά στρώματα (CNN)

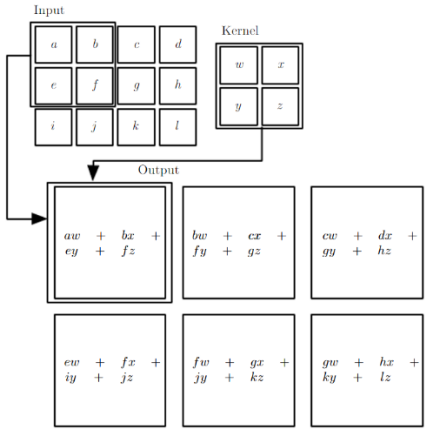
Τα CNN έχουν διακριθεί από άλλα νευρωνικά δίκτια χάρη στην βέλτιστη απόδοση τους σε δεδομένα εικόνας, ομιλίας, ήχου και χρονοσειρών. Έχουν τη δυνατότητα να "βλέπουν" μοτίβα στα δεδομένα μέσω τριών βασικών στρωμάτων (19), που εφαρμόζονται διαδοχικά όσες φορές ο χρήστης το προγραμματίσει.

1) Συνελικτικό στρώμα (Convolutional layer)

Το συνελικτικό στρώμα είναι βασικό στα δίκτυα αυτά και έχει σκοπό να εντοπίσει μοτίβα στα δεδομένα. Αυτό συμβαίνει πολλαπλασιάζοντας κάποιο πυρήνα (Kernel) με τον πίνακα εισόδου. Ο πυρήνας είναι ένας πίνακας ορισμένων διαστάσεων (συνήθως 3*3) και έχει μέσα τιμές ανάλογα με το μοτίβο που επιθυμεί να εντοπίσει ο αλγόριθμος. Οι διαστάσεις του πυρήνα είναι σαφώς μικρότερες από τον πίνακα εισόδου και έτσι, αν τα δεδομένα ήταν εικόνα, πολλαπλασιάζεται ο πχ. 3*3 πυρήνας με όλη την εικόνα χωρισμένη σε 3*3 Pixels διαδοχικά. Αυτή η διαδικασία ακολουθεί, σύμφωνα με τον (20) , την σχέση:

$$s(t) = (x * w)(t)$$

Όπου s είναι η συνελικτική συνάρτηση, x είναι ο πίνακας δεδομένων, w είναι ο πυρήνας και t είναι η μεταβλητή των δεδομένων.



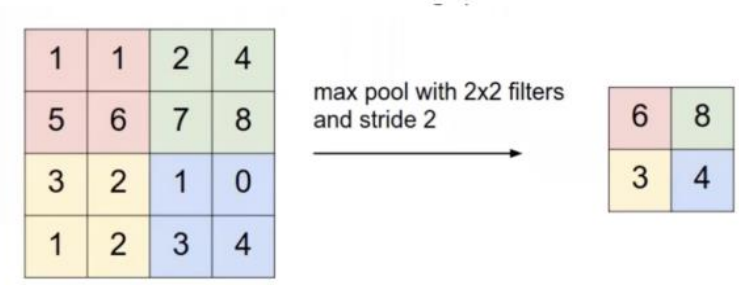
Εικόνα αριθμός (20): Εφαρμογή συνελικτική συνάρτησης με πίνακα διαστάσεων (2*2).

Οι βέλτιστες τιμές του κάθε πυρήνα εκμαθαίνονται από το μοντέλο, με τέτοιο τρόπο, ώστε να αντλούν κρίσιμες πληροφορίες από το δεδομένο όπως δομικά μοτίβα ή κυρίαρχες γραμμές στο δεδομένο. (21)

Εικόνα: Συγκεκριμένοι πυρήνες σε μοντέλα αναγνώρισης εικόνας που έχουν σκοπό στην δεξιά να εντοπίσουν τις άκρες όπου υπάρχουν στην εικόνα και στην αριστερή να την κάνουν πιο έντονη για καλύτερη επεξεργασία μετέπειτα. (21)

2) Στρώμα συσσώρευσης (Pooling layer)

Τα στρώματα συσσώρευσης εκτελούν μείωση διάστασης με σκοπό την διατήρηση μόνο της σημαντικής πληροφορίας. Ορίζεται κάποιο φίλτρο ορισμένων διαστάσεων, και περνιέται από τα δεδομένα με διαδοχικό τρόπο. Σε κάθε θέση στα δεδομένα, όπου βρίσκεται το προκαθορισμένο φίλτρο, μπορεί να επιλέγεται το μεγαλύτερο Pixel στο παράδειγμα της εικόνας ή ο μέσος όρος τους, όπως περιγράφεται στην εικόνα (νούμερο εικόνας). Σκοπός αυτού του στρώματος είναι να δημιουργηθεί ένα νέο δεδομένο, μικρότερων διαστάσεων, που δεν περιλαμβάνει θορυβώδης και περιττές λεπτομέρειες. Κατά την ταξινόμηση δεν είναι θεμιτό να λαμβάνεται υπόψη πχ σε ένα πρόσωπο που ακριβώς είναι τα μάτια άλλα η γενική τους τοποθεσία και ότι υπάρχουν.



Εικόνα: Φίλτρο που επιλέγει το μεγαλύτερο δεδομένο σε κάθε 2x2 πίνακα. (21)

3) Πλήρως συνδεδεμένο στρώμα (Fully-connected layer)

Αυτό το στρώμα αυτό είναι ένας γραμμικός ταξινομητής, που χρησιμοποιείται για να ταξινομήσει τα δεδομένα. Τα δύο προαναφερόμενα στρώματα εξαγάγουν αποτελέσματα σε σχέση με τα δεδομένα που εκφράζονται και ως χαρακτηριστικά των δεδομένων. Το πλήρως συνδεδεμένο στρώμα έρχεται να επεξεργαστεί τα χαρακτηριστικά αυτά και να βγάλει συμπεράσματα

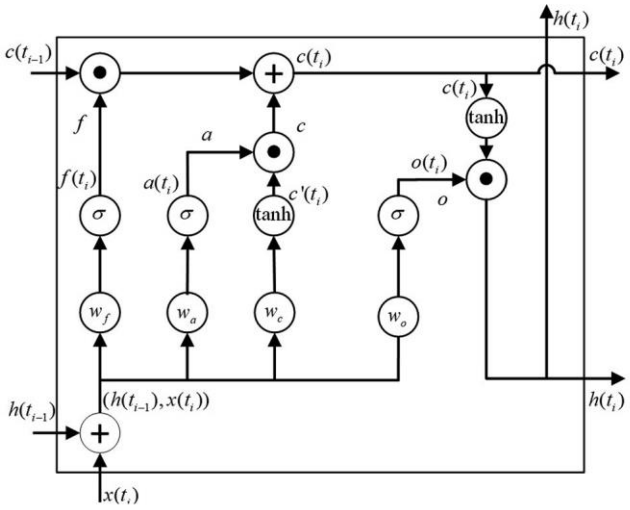
Νευρωνικά Δίκτυα μακράς και βραχείας μνήμης

Τα νευρωνικά δίκτυα μακράς και βραχείας μνήμης, Long Short-Term Memory (LSTM) αποδεικνύονται πολύ αποτελεσματικά σε προβλήματα όπως οι χρονοσειρές καθώς έχουν «μνήμη» που τα καθιστά ικανά να διαχειρίζονται μεγάλες ακολουθίες. Αυτό γίνεται μέσω της δυνατότητας τους να αποθηκεύουν δεδομένα σε νευρώνες, που θεωρούν ότι είναι σημαντικό χαρακτηριστικό.

Αναπτύχθηκαν από τους Sepp Hochreiter και Jürgen Schmidhuber στη μελέτη (22) για την αντιμετώπιση των εξαφανιζόμενων gradients (vanishing gradient problem). Το πρόβλημα του εξαφανιζόμενου gradient αποτελεί μια σημαντική πρόκληση στην εκπαίδευση αναδρομικών νευρωνικών δικτύων (RNNs). Η δυσκολία προκύπτει όταν οι διαδοχικές παράγωγοι της συνάρτησης σφάλματος μειώνονται εκθετικά κατά τη διάδοση προς τα πίσω (backpropagation), κάνοντας την ανανέωση των βαρών στα αρχικά επίπεδα του δικτύου ανεπαρκή. Αυτό οδηγεί σε μια σημαντική επιβράδυνση της σύγκλισης του μοντέλου κατά την εκπαίδευση ή ακόμα και στην πλήρη αδυναμία εκμάθησης βαθύτερων συσχετίσεων στα δεδομένα καθώς οι παράγωγοι μηδενίζονται.

Τα μοντέλα Long Short-Term Memory (LSTM) (23) αποτελούν μια εξελιγμένη μορφή των RNNs που σχεδιάστηκαν για να αντιμετωπίζουν το προαναφερθέν πρόβλημα. Τα LSTM εισάγουν την έννοια των πυλών (gates), που ελέγχουν τη ροή της πληροφορίας μέσα στη μνήμη της συσκευής. Αυτές οι πύλες - η πύλη εισόδου, η πύλη εξόδου, και η πύλη ξεχασμού - επιτρέπουν στο μοντέλο να διατηρεί ή να διαγράφει πληροφορίες σε μακροχρόνια ή βραχυχρόνια βάση. Επιπλέον, η δομή των LSTM διευκολύνει τη διατήρηση σταθερών gradient κατά τη διάρκεια της εκπαίδευσης, αντιμετωπίζοντας έτσι το πρόβλημα του εξαφανιζόμενου gradient με τη χρήση αθροιστικών στοιχείων. Είναι ιδανικά για εφαρμογές που απαιτούν την επεξεργασία σύνθετων και μακροχρόνιων χρονοσειρών.

Αναλυτικά, η λειτουργία των LSTMs, παρουσιάζεται στην εικόνα αριθμός.



Όπου:

- σ: σιγνοειδής συνάρτηση ενεργοποίησης
- tanh: συνάρτηση ενεργοποίησης tanh
- ⊙ : Πολλαπλασιασμός πινάκων
- ⊕: Αθροισμα πινάκων

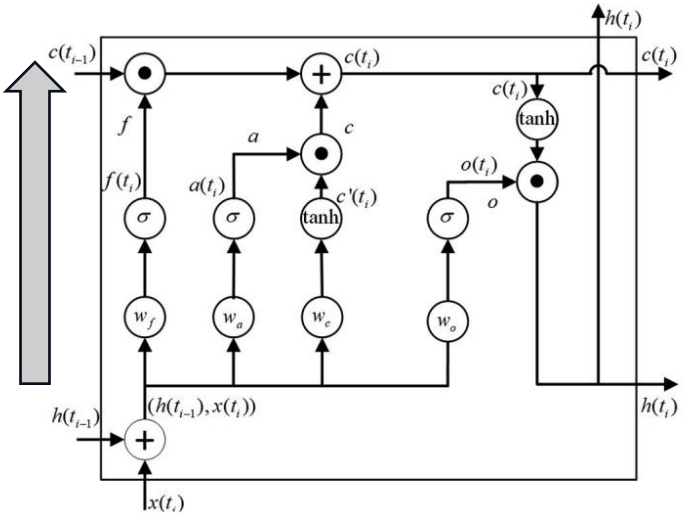
Εικόνα: Το κύταρρο κατάστασης ενός LSTM δικτύου. (24)

Στην εικόνα αριθμός απεικονίζεται η διαδικασία επεξεργασίας των δεδομένων σε ένα κύτταρο κατάστασης του LSTM μοντέλου.

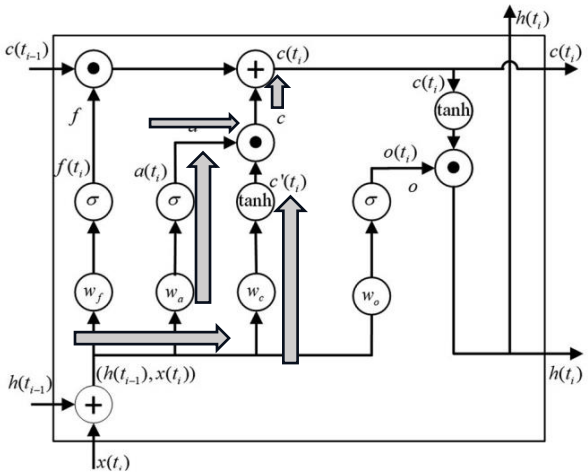
Ως είσοδος δίνεται ένα καινούργιο δεδομένο αθροισμένο με τα κομμάτια εκείνα που επιλέγει να «θυμάται». Στο τέλος της επεξεργασίας εξάγεται το καινούργιο κομμάτι που επιλέγει να «θυμάται» και αυτό με τη σειρά του αποτελεί κομμάτι της καινούργια εισόδου στο επόμενο κύτταρο κατάστασης. Αυτή η διαδικασία επαναλαμβάνεται για όλα τα δεδομένα. Παρακάτω παρουσιάζεται αναλυτικά κάθε πύλη. (25) (26)

<u>Input Gate</u>

Έπειτα τα δεδομένα περνάνε ταυτόχρονα από μια σιγμοειδούς συνάρτηση και μία tanh. Καθώς η σιγμοειδής συνάρτηση επιστρέφει τιμές μεταξύ [0,1] θεωρείται βάρος και πολλαπλασιάζεται με την τιμή που επιστρέφει η συνάρτηση tanh που επιστρέφει [-1,1]. Η έξοδο της input gate είναι το γινόμενο των εξόδων των δύο συναρτήσεων ενεργοποίησης.

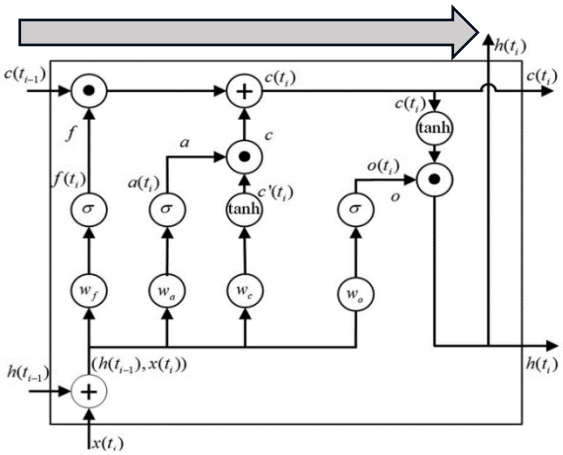


σιγμοειδούς συνάρτησης ξεχνιούνται όσα φέρουν την τιμή μηδέν. Αυτό το στρώμα ονομάζεται forget gate.

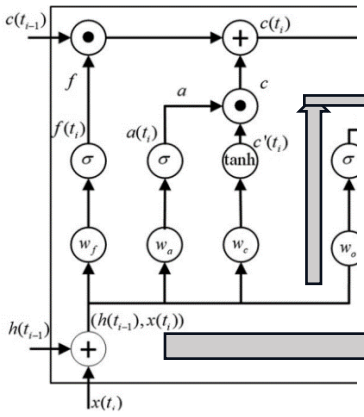


Cell State

Πριν τη συγκεκριμένη πύλη ξανά εισέρχονται τα παλιά δεδομένα που θυμάται και πολλαπλασιάζονται με την έξοδο της forget gate. Το γινόμενο αυτό έχει πιθανότητα είναι μηδέν, όμως για την αποφυγή της πιθανότητας να μηδενιστούν οι παράγωγοι αθροίζεται με την έξοδο της input gate. Η έξοδος του cell state είναι η τιμή του γινομένου αυτού.



Output Gate



Αυτή είναι η τελευταία πύλη που καθορίζει την συνολική έξοδο. Το καινούργιο δεδομένο αθροισμένο με το δεδομένο που θυμάται περνάνε απο την σιγμοειδή συνάρτηση. Έπειτα η έξοδος του cell state περνάει απο την tanh συνάρτηση ενεργοποίησης. Οι έξοδοι των δύο συναρτήσων ενεργοποίησης πολλαπλασιάζονται και αυτό αποτελεί την έξοδο που ονομάζεται hidden state που προορίζεται να εισέλθει ξανα στο επόμενο LSTM cell.

Συναρτήσεις απώλειας (loss-functions)

Η συναρτήσεις απώλειας, loss functions, περιγράφουν τη διαφορά ανάμεσα στην προβλεπόμενη και την πραγματική τιμή του δεδομένου. Είναι η βασική μετρική που χρησιμοποιείται για να βελτιωθεί η απόδοση του νευρωνικού δικτύου. Αποτελείται από μία συνάρτηση που υπολογίζει, με διάφορους τρόπους, την απόσταση της πραγματικής και προβλεπόμενης τιμής. Στην ουσία την απόκλιση του μοντέλου.

Αυτό γίνεται παράλληλα με την διαδικασία της forward και back propagation, που είναι υπεύθυνες για την διαδικασία αναβάθμισης των βαρών του μοντέλου. Για κάθε επανάληψη, και για κάθε νέο δεδομένο που το μοντέλο επεξεργάζεται υπολογίζεται η συνάρτηση απώλειας και αναβαθμίζονται τα βάρη με σκοπό να ελαχιστοποιηθεί η συνάρτηση αυτή.

Παρακάτω, παραθέτετέ, οι συναρτήσεις απώλειας που χρησιμοποιούνται σε όλη τη διαδικασία των πειραμάτων. Αρχικά η συνάρτηση Binary Cross-Entropy που χρησιμοποιείται για την εκπαίδευση όλων των ταξινομητών και των αυτοκωδικοποιητών. Έπειτα παρουσιάζονται και οι συναρτήσεις κόστους που χρησιμοποιούνται στην διαδικασία παραγωγής αντιπαραδειγμάτων, και στην τελική τους εκτίμηση.

Binary cross-entropy (27):

Χρησιμοποιείται σε δυαδικά προβλήματα με δύο κλάσεις. Συγκρίνει τις προβλεπόμενες πιθανότητες με την πραγματική έξοδο η οποία μπορεί να είναι είτε 0 είτε 1. Η συνάρτηση αυτή εκφράζει την διαφορά εντροπίας των δύο εξόδων (11). Η εντροπία εκφράζει την πιθανότητα να προβλέψει κάτι συγκεκριμένο ο ταξινομητής (να ακολουθούν οι έξοδοι μια συγκεκριμένη ακολουθία ή όχι). Αν οι πραγματικές τιμές των δεδομένων είναι τυχαίες, το loss τείνει και τον ταξινομητή να επιστρέφει «τυχαίες» (μεταξύ τους και όχι με τα δεδομένα) τιμές. Διαφορετικά αν ακολουθούν μια συγκεκριμένη κατανομή, για παράδειγμα Gauss, η συνάρτηση Binary Cross-Entropy θα είναι μικρό όταν και οι προβλεπόμενες ακολουθούν την αντίστοιχη κατανομή.

$$\log(Loss) = -\frac{1}{N}\sum_{i=1}^{N} Yi \log \hat{Y}i + (1 - Yi) \log\left(1 - \hat{Y}i\right)$$

MSE – Μέσο Τετραγωνικό Σφάλμα (Mean Square Error):

Αποτελεί την πιο απλή συνάρτηση απώλειας. Αρχικά υπολογίζεται η διαφορά της τιμής πρόβλεψης και της πραγματικής τιμής, υψώνεται στο τετράγωνο και κατόπι υπολογίζεται το μέσος όρος όλων αυτών των τιμών. Είναι εύκολη στην ερμηνεία λόγο της απλού τύπου υπολογισμού, είναι πάντα διαφορίσιμη λόγω του τετραγώνου και παρουσιάζει μόνο ένα τοπικό ελάχιστον οπότε λειτουργεί καλά πλάι στον αλγόριθμο βελτιστοποίησης Gradient Descent (27).

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(Yi - \hat{Y}i\right)^2$$

Όπου N το πλήθος των δεδομένων, Yi η πραγματική τιμή και $\hat{Y}i$ η προβλεπόμενη τιμή.

MAE- Μέσο Απόλυτο Σφάλμα (Mean absolute error):

Μοιάζει με την MSE στη συνάρτηση, στην απλότητα και στο πόσο συχνά χρησιμοποιείται. Υπολογίζεται ως το απόλυτο του μέσου όρου της διαφοράς της πραγματικής τιμής και της προβλεπόμενης τιμής και υπολογίζεται από τον τύπο (27):

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|Yi - \hat{Y}i\right|$$

Ευκλείδια απόσταση

Όπως είναι προφανές από την ονομασία η μέθοδος αυτή υπολογίζει την ευκλείδεια απόσταση μεταξύ των πραγματικών τιμών και των προβλεπόμενων τιμών ως εξής:

$$euc_{dist} = \sqrt{\sum_{i=1}^{N} \left(Yi - \hat{Y}i\right)^2}$$

Normalized l2(NED- Normalized Euclidean distance):

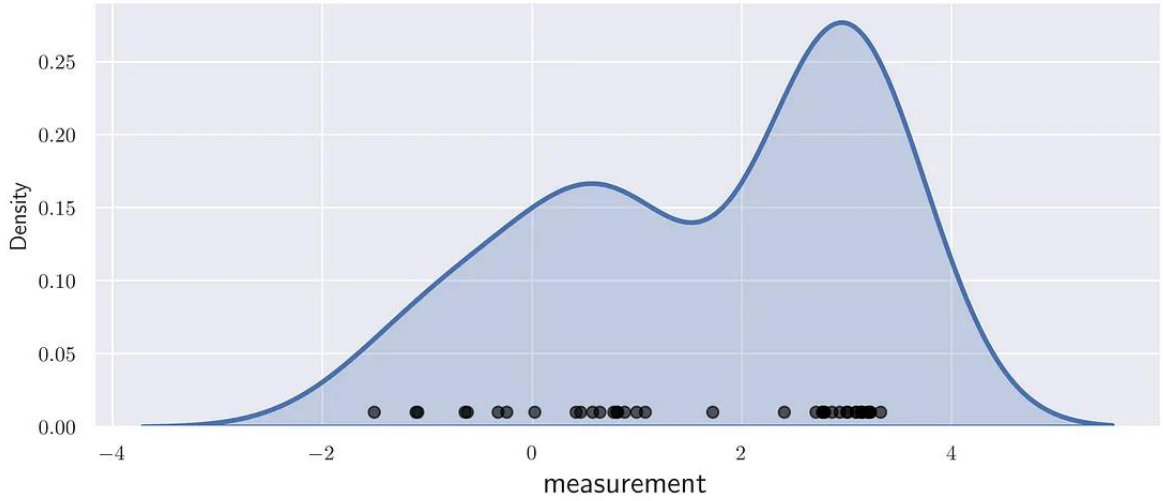
Η συνάρτηση αυτή είναι μια παραλλαγή της ευκλείδειας απόστασης που κανονικοποιεί με βάση τη διακύμανση των δεδομένων. Η συγκεκριμένη συνάρτηση υπολογίζεται στα αρχικά δεδομένα (στα X) και όχι στις κλάσεις και προβλέψεις των δεδομένων όπως οι συναρτήσεις σφάλματος που έχουν παρουσιαστεί ως τώρα. Χρησιμοποιείται για τον υπολογισμό της απόστασης του αντιπαραδείγματος και του αρχικού δείγματος ως εξής:

$$NED^2\left[Xi, \hat{X}i\right] = \frac{0.5 \times Var\left[Xi - \hat{X}i\right]}{Var[Xi] + Var\left[\hat{X}i\right]}$$

Οπου Xi είναι η αρχική χρονοσειρά $\hat{X}i$ το αντιπαράδειγμα και Var η διασπορά.

KDE (Kernel Density Estimation) (28)

Το KDE είναι μία στατιστική μέθοδος που υπολογίζει την συνάρτηση πυκνότητας πιθανότητας των δεδομένων, όταν τα δεδομένα δεν ακολουθούνε μια γνωστή κατανομή όπως την Κανονική, Poisson κτλ.



Εικόνα αριθμός από (28): Στην εικόνα φαίνεται ένα διάγραμμα εφαρμοσμένο στην κατανομή των δεδομένων. Η κορυφή της καμπύλης είναι σημείο που ομαλά συνδέεται με την υπόλοιπη καμπύλη στην οποία βρίσκονται τα περισσότερα δεδομένα, και μέσω ένός τέτοιου διαγράμματος παρατηρόντας που βρίσκονται τα τοπικά ακρότατα είναι εύκολο να κατανοήσουμε που υπάρχει πληθώρα δεδομένων και που αραιώνουν.

Η συγκεκριμένη μέθοδος υπολογίζει για κάθε δεδομένο ξεχωριστά μια καμπύλη με κορυφή το σημείο που βρίσκεται το δεδομένο και στη συνέχεια προσθέτει όλες τις καμπύλες και προκείπτει μία ενιαία που εκφράζει την κατανομή των δεδομένων.

Η καμπύλη αυτή ονομάζεται Kernel (K) ή αλλιώς πυρήνας και υπολογίζεται απο την κανονική κατανομή ως εξής:

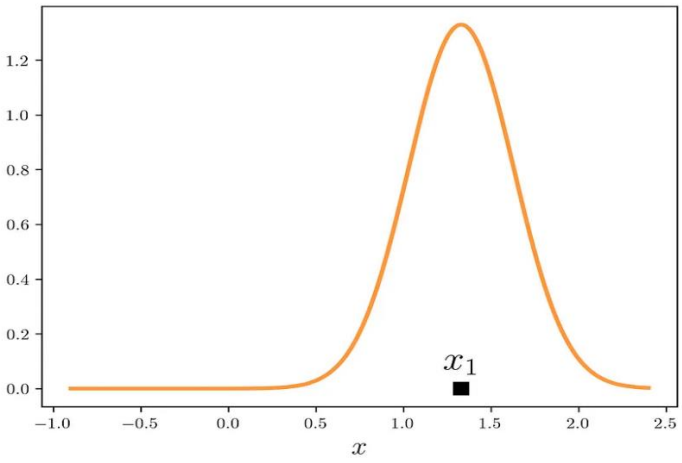
$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

Όπου x το δεδομένο που κατατάσεται.

Στη συνέχεια υπολογίζεται ο τύπος:

$$f(x) = \frac{1}{h} K\left(\frac{x - x_i}{h}\right)$$

Όπου x είναι το σύνολο των δεδομένων , x_i είναι το επιλεγμένο δεδομένο που επιθύμείται να βρεθεί η κατανομή του σε σχέση με την υπόλοιπη βάση δεδομένων και h ονομάζεται το bandwidth το οποίο εφράζει πόσο ομαλή η απότομή θα είναι η γραφική της κατανομής. (ανάλογα τα χαρακτηριστικά των δεδομένων είναι αναγκαίο κάποιο διαφορετικό bandwidth) Η γραφική του έχει τη μορφή:

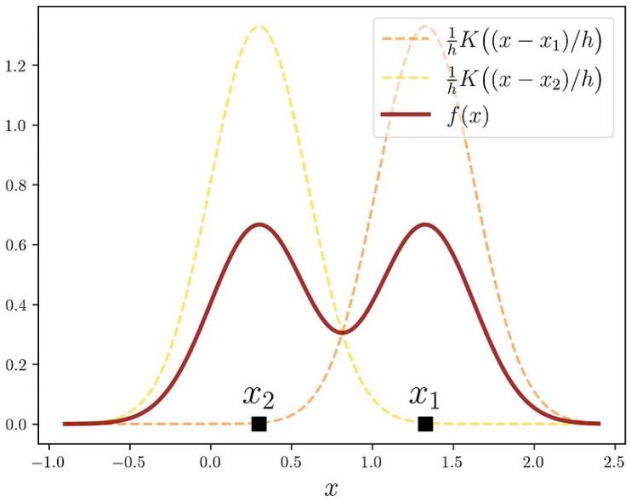


Εικόνα αριθμός απο (28): Γραφική παράσταση της κατανομής του x_i απο τον τύπο (αριθμός)

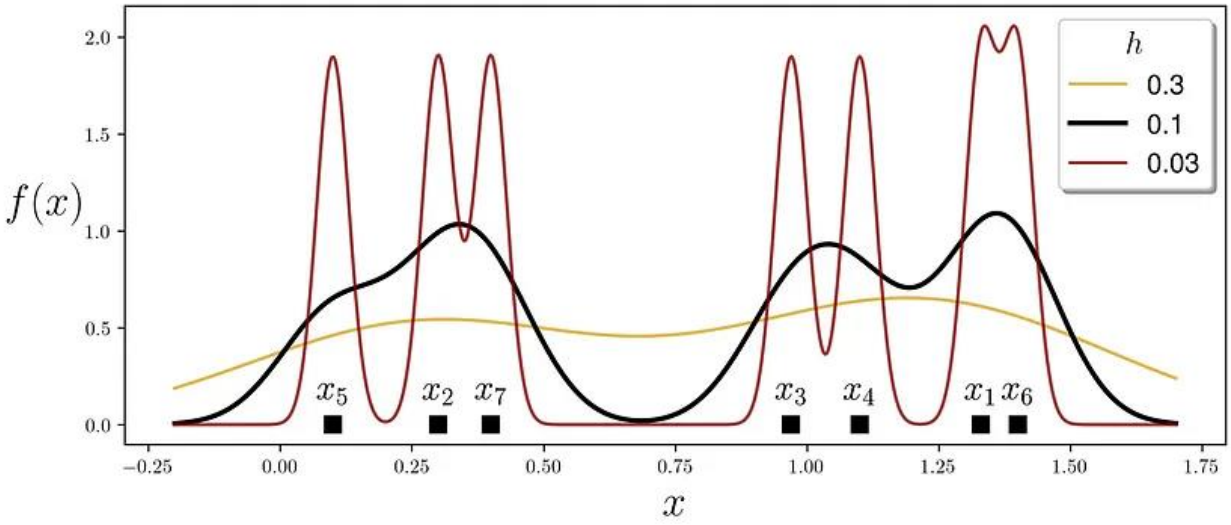
Για δύο δεδομένα υπολογίζεται η γραφική ως εξής:

$$f(x) = \frac{1}{h}\sum_{i=1}^{2} K\left(\frac{x - x_i}{h}\right)$$

Και έχει τη μορφή:



Εικόνα αριθμός απο (28): Η γραφική παράσταση του KDE με χρήση δύο δεδομένων



Εικόνα αριθμός απο (28): Η γραφική παράσταση πολλών δεδομένων με τη χρήση διαφορετικών bandwidths.

Στη συγκεκριμένη εργασία εφαρμόζεται το KDE σε κάθε διάσταση των χρωνοεξαρτούμενων δεδομένων ξεχωριστά. Έπειτα συγκρίνεται με το μέσο όρο του αθροίσματος της λογαριθμικής κατανομής όλων των δεδομένων στην αντίστοιχη διάσταση. Αυτό συμβαίνει καθώς η λογαριθμική τιμή της διαφοράς της κατανομής αποτρέπει τιμές υπερ-επηρεασμένες απο λίγα δεδομένα πολύ μακρια, ή φέρνει σε πιο

κατανοητή κλίμακα πολύ μικρές τιμές που τείνουν να μυδενίζονται ή πολύ μεγάλες που τείνουν στο άπειρο.

Στη δημιουργία αντιπαραδειγμάτων είναι σημαντικό να κρατάτε η μορφή της κατανομής καθως βοηθάει το μοντέλο στη «μορφή» των αντιπαραδειγμάτων που καλείται να δημιουργήσει.

Ο αλγόριθμος βελτιστοποιησης Adam

Ο αλγόριθμος Adam (Adaptive Moment Estimation) προτάθηκε από τους Kigma και Ba το 2015 (29) και έκτοτε έχει γίνει πολύ γνωστός στον κλάδο της μηχανικής μάθησης καθώς έχει αποδειχθεί από τους πιο χρήσιμους και αποτελεσματικούς αλγόριθμους στα μοντέλα βαθιάς μάθησης. Είναι συνέχεια της Στοχαστικής Κατάβασης Κλίσης (stochastic gradient descent) που είναι ο πιο δημοφιλής αλγόριθμος βελτιστοποίησης σήμερα, και βασίζεται στους αλγορίθμους AdaGrad και AMS-Prop.

Οι αλγόριθμοι βελτιστοποίησης είναι χρήσιμοι καθώς περιγράφουν την διαδικασία ελαχιστοποίησης των συναρτήσεων απώλειας.

Ο αλγόριθμος Adam έχει πολλά θετικά (29):

- Είναι απλός στην υλοποίηση του.
- Είναι υπολογιστικά αποδοτικός.
- Έχει μικρές απαιτήσεις στη μνήμη οπότε είναι και αποτελεσματικός σε μεγάλο πλήθος δεδομένων ή παραμέτρων.
- Είναι κατάλληλος για μη σταθερούς στόχους και για προβλήματα με θόρυβο στα δεδομένα.

Ο κυρίαρχος λόγος για τον οποίο χρησιμοποιείται σε αυτήν την εργασία είναι γιατί αντιμετωπίζει το πρόβλημα του τοπικού ελαχίστου σε αντίθεση με τον αλγόριθμο Gradient Descent. Η μέθοδος που μελετάται σε αυτή την εργασία βασίζεται στην LatentCF. Μία βάση για την δημιουργία εξηγήσεων μέσω αντιπαραδειγμάτων που χρησιμοποιεί Gradient Descent για να ελαχιστοποιήσει τη συνάρτηση απώλειας των αντιπαραδειγμάτων. Η επιλογή συνάρτηση απώλειας καθιστά τη μέθοδο χρονοβόρα και απαιτητική μέχρι να βρεθεί ο κατάλληλος ρυθμός εκμάθησης καθώς είναι πολύ εύκολο στην κατάβαση να κολλήσει στο τοπικό ελάχιστο. Αυτό μπορεί να έχει ως αποτέλεσμα να μην δημιουργούνται αντιπαραδείγματα (να επιστρέφονται χωρίς να έχουν αλλάξει κλάση τα δεδομένα) διότι δεν βρίσκει το ολικό ελάχιστο του σφάλματος. Πάνω σε αυτό ήρθε και χτίστηκε η μέθοδος LatentCF++ (2) η οποία με την εισαγωγή του Adam έλυσε αυτήν την αδυναμία.

Ο Adam βασίζεται πάνω στην ορμή (Momentum) η οποία έχει τη δυνατότητα να κατευθύνει τον αλγόριθμο ελαττώνοντας τις κινήσεις προς άσχετες κατευθύνσεις. Αυτό γίνεται επειδή ο αλγόριθμος αυτός υπολογίζει την κλίση με βάση μια προσέγγισης της μελλοντικής τιμής του μοντέλου σε αντίθεση με τον Gradient Descent

που το υπολογίζει με βάση τις παραμέτρους. Η ορμή επιτρέπει την πρόβλεψη της μελλοντικής τιμής.

Για να γίνει κατανοητό πως ακριβώς λειτουργεί ο Adam πρέπει αρχικά να οριστούνε οι παράμετροι. Το α = 0.001 που ορίζει το βήμα, τα $\beta_1 = 0.9,\ \ \beta_2 = 0.999\ \ και\ \varepsilon = 10^{-8}$, ο οποίος είναι ένας εξαιρετικά μικρός αριθμός για την αποφυγή διαίρεσης με το μηδέν. Να σημειωθεί ότι οι τιμές που έχουν δοθεί στις παραμέτρους είναι προτεινόμενες από τους συντάκτες του αλγορίθμου και όχι απόλυτες τιμές τις οποίες *πρέπει* να λάβουν οι παράμετροι.

Βασικό ρόλο στον αλγόριθμο έχουν τα δύο διανύσματα $m_t\ και\ u_t$ τα οποία αρχικά έχουν την τιμή μηδέν αλλά με κάθε βήμα υπολογίζεται καινούργια τιμή με βάση τις δύο εξισώσεις:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$u_t = \beta_1 u_{t-1} + (1 - \beta_2) g_t^2$$

Όπου m_t είναι η μέση τιμή της κλίσης και u_t είναι η μέση τιμή της τετραγωνικής κλίσης, t είναι το τρέχον βήμα εκπαίδευσης και g_t ορίζεται ως η κλίση (κατεύθυνση και μέγεθος) της αλλαγής στο θ και εκφράζεται από τον τύπο:

$$g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$$

Όπου $f_t(\theta)$ εκφράζει την απώλεια προς ελαχιστοποίηση, όπως το mean squared error και το cross-entropy.

Καθώς τα δύο διανύσματα $m_t\ και\ u_t$ αρχικοποιούνται στο μηδέν, για αρκετές επαναλήψεις συνεχίζουν να είναι πολύ κοντά στο μηδέν. Αυτό έχει ώς αποτέλεσμα να αργεί πολύ να συγκλίνει ο αλγόριθμος. Έτσι ορίζονται τα διορθωμένα διανύσματα και υπολογίζονται ως εξής:

$$\widehat{m}_t = \frac{m_t}{1 - \beta_1^t}$$

$$\hat{u}_t = \frac{u_t}{1 - \beta_2^t}$$

Με αυτόν τον τρόπο, στα πρώτα βήματα όπου το t είναι μικρότερο τα δύο διανύσματα θα είναι πιο κοντά στις πραγματικές τιμές των κλίσεων, επιτρέποντας στον αλγόριθμο να ενημερώσει τις παραμέτρους του μοντέλου πιο αποτελεσματικά.

Έτσι υπολογίζεται το θ που είναι και η βασική παράμετρος των αλγορίθμων βελτιστοποίησης ως εξής:

$$\theta_t \leftarrow \frac{\theta_{t-1} - \alpha \cdot \hat{m}_t}{\left(\sqrt{\hat{u}_t} + \varepsilon\right)}$$

Ενώ οι παραδοσιακοί μέθοδοι βελτιστοποίησης μπορούν συχνά να παγιδεύονται σε τοπικά ελάχιστα, οδηγώντας σε υπό βέλτιστη απόδοση του μοντέλου, ο ρυθμός μάθησης του Adam και οι τύποι ενημέρωσης παρέχουν ένα πιο σίγουρο μονοπάτι προς το παγκόσμιο ελάχιστο. Αυτό το χαρακτηριστικό όχι μόνο ενισχύει την αποδοτικότητα της διαδικασίας εκπαίδευσης, αλλά συχνά οδηγεί και σε καλύτερη γενίκευση στο τελικό μοντέλο.

Πρακτικό Μέρος

Το πειραματικό μέρος ξεκινάει με την παρουσίαση της ιστορίας της μέθοδος LatentCF++KDE. Είναι η μέθοδος που προτείνεται για υπολογιστικά γρήγορη και αποτελεσματική δημιουργία αντιπαραδειγμάτων σε δεδομένα χρονοσειράς. Είναι γρήγορη καθώς η επεξεργασία γίνεται σε συμπιεσμένο χώρο, και αποτελεσματική χάρη στην εισαγωγή του κριτηρίου της κατανομής. Οι συζητήσεις στο χώρο του explainable AI προτείνουν μια μετατόπιση από την εστίαση στις μικρότερες αλλαγές προς μια πιο σημασιολογική κατεύθυνση των αλλαγών. Αναπτύσσονται μελέτες, όπως η (30) που στοχεύει στη δημιουργία πιο κατανοητών και «λογικών» αντιπαραδειγμάτων, με τη χρήση γράφων και την αξιοποίηση των περίπλοκων σχέσεων που μπορούν και εκφράζουν, και την (31) που όμοια κινείται για την επεξεργασία Φυσικής Γλώσσας. Στην κίνηση που παρατηρείται προς πιο «λογικά» αντιπαραδείγματα στηρίζεται και η προτεινόμενη μέθοδος.

Παράθεση της μεθόδου

Στόχος σε αυτήν την εργασίας είναι να μελετηθεί ή μέθοδος LatentCF++ που δημιουργεί αντιπαραδείγματα για πολυδιάστατες χρονοσειρές. Ορίζεται το πρόβλημα των αντιπαραδειγμάτων παρόμοια με τις έρευνες (32) και (2):

Έχοντας ένα ταξινομητή f που αποτελεί μαύρο κουτί (black-box classifier), ο οποίος είναι εκπαιδευμένος με τις χρονοσειρές υπό μελέτη, και ως έξοδο επιστρέφει τις πιθανότητες να ανήκει το κάθε δεδομένο στην κάθε κλάση, η μέθοδος αυτή μπορεί και επιστρέφει τις αλλαγές εκείνες που είναι απαραίτητες να γίνουν στα χαρακτηριστικά των δεδομένων έτσι ώστε να μεταβούμε από την κλάση 0 (undesired state) στην κλάση 1 (desired state). Εισάγεται ένα δείγμα x_0 μέσω του ταξινομιτή f : X → [0,1] και ενημερώνεται το δείγμα απο τη μέθοδο μέχρι να είναι κοντά στο στοχευμένο όριο απο το οποίο μετράτε απο μια συνάρτηση απώλειας L.

Για τη μέθοδο αυτή απαιτείται η ύπαρξη ενός προ εκπαιδευμένου ταξινομητή και ενός προ εκπαιδευμένου αυτοκωδικοποιητή που να έχει τη δυνατότητα να συμπιέσει τα δεδομένα με σκοπό την εύρεση αντιπαραδειγμάτων στο συμπιεσμένο χώρο. Είναι αξιοσημείωτο ότι ο ταξινομητής και ο αυτοκωδικοποιητής λειτουργούν ως «μαύρο κουτί». Αυτό σημαίνει ότι η μέθοδος αυτή είναι ανεξάρτητή από την εσωτερική δομή των δύο μοντέλων (model-agnostic), και δεν λαμβάνει υπόψη της τα ειδικά χαρακτηριστικά των μοντέλων, όπως τα βάρη. Το μόνο που είναι απαραίτητο

από την πλευρά του ταξινομητή είναι η πρόβλεψη του σε κάθε στάδιο της διαδικασίας και από την πλευρά του αυτοκωδικοποιητή η συμπίεση των δεδομένα. Παρακάτω θα οριστεί το πρόβλημα επίσημα σύμφωνα με την έρευνα (2).

LatentCF++

Δίνεται ένας ταξινομητής f(·) και ένα δείγμα χρονοσειράς x που περιέχει t χρονικά βήματα, τέτοια ώστε η έξοδος να αναπαρίσταται ως f(x)=′−′ με πιθανότητα $\hat{y}$. Στο πρόβλημα, $\hat{y}$ είναι μικρότερο από το όριο απόφασης τ (δηλ. $\hat{y}$<τ) καθώς καθορίζει αρνητικό ή αλλιώς ανώμαλο (που στόχος είναι να μετατραπεί σε ομαλό). Ο στόχος είναι να χρησιμοποιηθεί ένας αυτοκωδικοποιητής αποτελούμενος από μια συνάρτηση κωδικοποίησης E(·) και μια συνάρτηση αποκωδικοποίησης D(·) για να βρεθεί το παραγόμενο αντιπαράδειγμα x′ με το επιθυμητό θετικό αποτέλεσμα. Διαταράσσεται η κωδικοποιημένη ενδιάμεση αναπαράσταση z=E(x) μέσω μιας συνάρτησης βελτιστοποίησης ADAM επαναληπτικά, ώστε να παραχθεί ένα νέο δείγμα χρονοσειράς x′=D(z) τέτοιο ώστε το στόχο της εξόδου f(x′)=′+′. Τέλος, ελαχιστοποιείται η συνάρτηση που δείχνει την απώλεια μεταξύ $\hat{y}$ και τ.

LatentCF++_KDE

Αυτή η έρευνα επεκτείνει τις βάσεις που έθεσε η μεθοδολογία LatentCF++ και ασχολείται ενδελεχώς με την ανάλυση πολυδιάστατων χρονοσειρών. Προσθέτει έναν καινοτόμο μηχανισμό στη διαδικασία υπολογισμού της απώλειας, ο οποίος ενσωματώνει τη μέθοδο KDE (Kernel Density Estimation). Μέσα από αυτή την προσθήκη, η μεθοδολογία επιχειρεί να συγκρίνει τις διαφορές στις κατανομές των δεδομένων των δύο κλάσεων - της κλάσης με τα ομαλά και της κλάσης με τα ανώμαλα δεδομένα - σε σχέση με τα δεδομένα που υπόκεινται σε διαδικασίες αλλαγής.

Η προσθήκη αυτή βασίζεται στην LatentCF από τη διατριβή (1) που χρησιμοποιεί την μέθοδο KDE για την εκτίμηση των αποτελεσμάτων και στο θεωρητικό υπόβαθρο που ορίζει η διατριβή (33) και συμπυκνώνεται στη φράση ότι η τροποποίηση που υπόκεινται τα δεδομένα θα πρέπει να είναι εφικτή ως προς την κατανομή τους, μια παρατήρηση που προσθέτει περιορισμούς στις εναλλακτικές εισόδους.

Με αυτόν τον τρόπο, ο στόχος δεν περιορίζεται απλώς στην αναγνώριση των αντιπαραδειγμάτων από τον ταξινομητής ως δεδομένα της «ομαλής» κλάσης, αλλά επιχειρείται επίσης η επίτευξη μιας ομοιομορφίας στην κατανομή τους σε σύγκριση με την «ομαλή» κλάση. Αυτή η προσέγγιση φιλοδοξεί να αποτρέψει σενάρια όπου ο ταξινομητής καταφέρνει να ταξινομήσει σωστά τα δεδομένα, ωστόσο αυτά δεν εμφανίζουν πλέον ουσιαστική ομοιότητα με τα χαρακτηριστικά της ομαλής κλάσης, όπως θα καταλάβαινε ο ανθρώπινος παράγοντας. Αντίθετα, τα δεδομένα αυτά καταλήγουν να είναι μια ασαφής ενδιάμεση μορφή που δεν ανήκει σαφώς σε καμία από τις δύο κλάσεις.

Με την νέα προσθήκη η συνάρτηση απώλειας ορίζεται από τρία μέρη και δίνεται η δυνατότητα στον χρήστη να ορίσει τα βάρη που θα λαμβάνει η κάθε μία από τις υπό συναρτήσεις της απώλειας. (Τα τρία βάρη των τριών υπό συναρτήσεων έχουν πάντα άθροισμα ίσο με ένα).

Margin Difference

Το πρώτο μέρος της συνάρτησης απώλειας είναι το Margin Difference. Ορίζεται ως η μέση τετραγωνική απόσταση της πιθανότητας του δεδομένο από τον ταξινομητή και του decision boundary (τ). Πιθανότητα του δεδομένου ορίζουμε την έξοδο του ταξινομητή, έναν αριθμό που παίρνει τιμές στο διάστημα [0,1] και αν είναι κάτω από 0.5 τότε το δεδομένο ανήκει στην ανώμαλη κλάση, διαφορετικά ανήκει στην ομαλή. Το Decision Boundary ορίζεται ένας αριθμός πάλι στο διάστημα [0,1] ο οποίος δηλώνει το «που θέλουμε να ψάχνει ο ταξινομητής για αντιπαράδειγμα», δηλαδή ποια είναι η ελάχιστη αποδεκτή τιμή της πιθανότητας του δεδομένου.

Τα δύο βήματα είναι τα εξής:

$$\mathrm{MSE} = (\mathrm{y}_{pred} - \tau)^2$$

$$loss += weight * MSE$$

Όπου y_{pred} είναι η πιθανότητα του δεδομένου, τ είναι το decision boundary και weight είναι το βάρος που ορίζεται από το χρήση για τη συνάρτηση αυτή.

Η υπό συνάρτηση αυτή, μετά από δοκιμές στα πειράματα, καταλαμβάνει βάρος 60%-70% διότι είναι ο βασικός παράγοντας που ορίζει αν το δεδομένο αποτελεί η όχι αντιπαράδειγμα.

Proximity

Δεύτερο μέρος της συνάρτησης της απώλειας είναι η μέση απόλυτη απόσταση μεταξύ του αρχικού δεδομένου X και του δεδομένου που υπέστη αλλαγές X′ πολλαπλασιασμένο με τα παγκόσμια βάρη των δεδομένων. Τα X, X′ και τα βάρη είναι πίνακες ιδίων διαστάσεων (timesteps, features).

Για την εύρεση των παγκόσμιων βαρών των δεδομένων, που ονομάζονται step_weights στην εργασία αυτή, χρησιμοποιείται μία μέθοδος η οποία κάνει μικρές αλλαγές στα δεδομένα και με τη βοήθεια του ταξινομητή καθορίζει ποιες στιγμές είναι αυτές που επηρεάζουν σημαντικά την έξοδο του ταξινομητή. Τα βάρη αυτά είναι ένας πίνακας διαστάσεων όσο το αρχικό δεδομένο και σε κάθε θέση του πίνακα βρίσκεται είτε το ψηφίο [0] είτε [1]. Όπου υπάρχει μηδέν το αντίστοιχο χρονικό σημείο στα δεδομένα δεν έχει βαρύτητα και αντίστοιχα ένα όπου τα σημεία είναι σημαντικά για τον ταξινομητή.

Έτσι οι τύποι έχουν τη μορφή:

$$MAE_{(X,X')} = \frac{1}{N}\sum_{i=1}^{N}|X_i - X'_i|$$

$$loss = MAE_{(X,X')} * step\_weights$$

Όπου το X είναι το αρχικό δεδομένο σε μορφή πίνακα με διαστάσεις (διαστάσεις, χρονικές στιγμές), X′ είναι το αντιπαράδειγμα και τέλος τα step_weights είναι τα βάρη. Έτσι το τελικό αποτέλεσμα είναι πίνακας διαστάσεων όσο το αρχικό, όπου έχει [0] στις στιγμές που δεν επηρεάζουν το αποτέλεσμα και [1] σε αυτές που το επηρεάζουν.

Το βάρος αυτής της υπό συνάρτησης είναι συνήθως 20%-30% (στα πειράματα αυτής της εργασίας). Η συνάρτηση αυτή εκφράζει την προσπάθεια να «γίνονται οι λιγότερο δυνατόν αλλαγές στο δεδομένο», όμως με βάρος μεγαλύτερο από το προαναφερόμενο τα δεδομένα θα αποτρέπονταν από το να μεταπηδήσουν κλάση.

KDE Diffrence

Τέλος, υπολογίζεται η διαφορά κατανομής (KDE) του δεδομένου που υπόκειται αλλαγές και όλου του σετ δεδομένων της «ομαλής» κλάσης. Για το υπολογισμό του KDE υπολογίζεται η διαφορά των συνολικών δεδομένων (data) και των δεδομένων υπό μελέτη (x). Έπειτα υπολογίζεται η διαφορά KDE με χρήση bandwidth d όπου d ορίζεται το πλήθος των στοιχείο σε κάθε διάσταση του δεδομένου. Με τη χρήση του τύπου

$$f(x) = \frac{1}{h}\sum_{i=1}^{2}\left(\frac{1}{\sqrt{2\pi}}e^{-\frac{x - x_i{}^2}{2h}}\right)$$

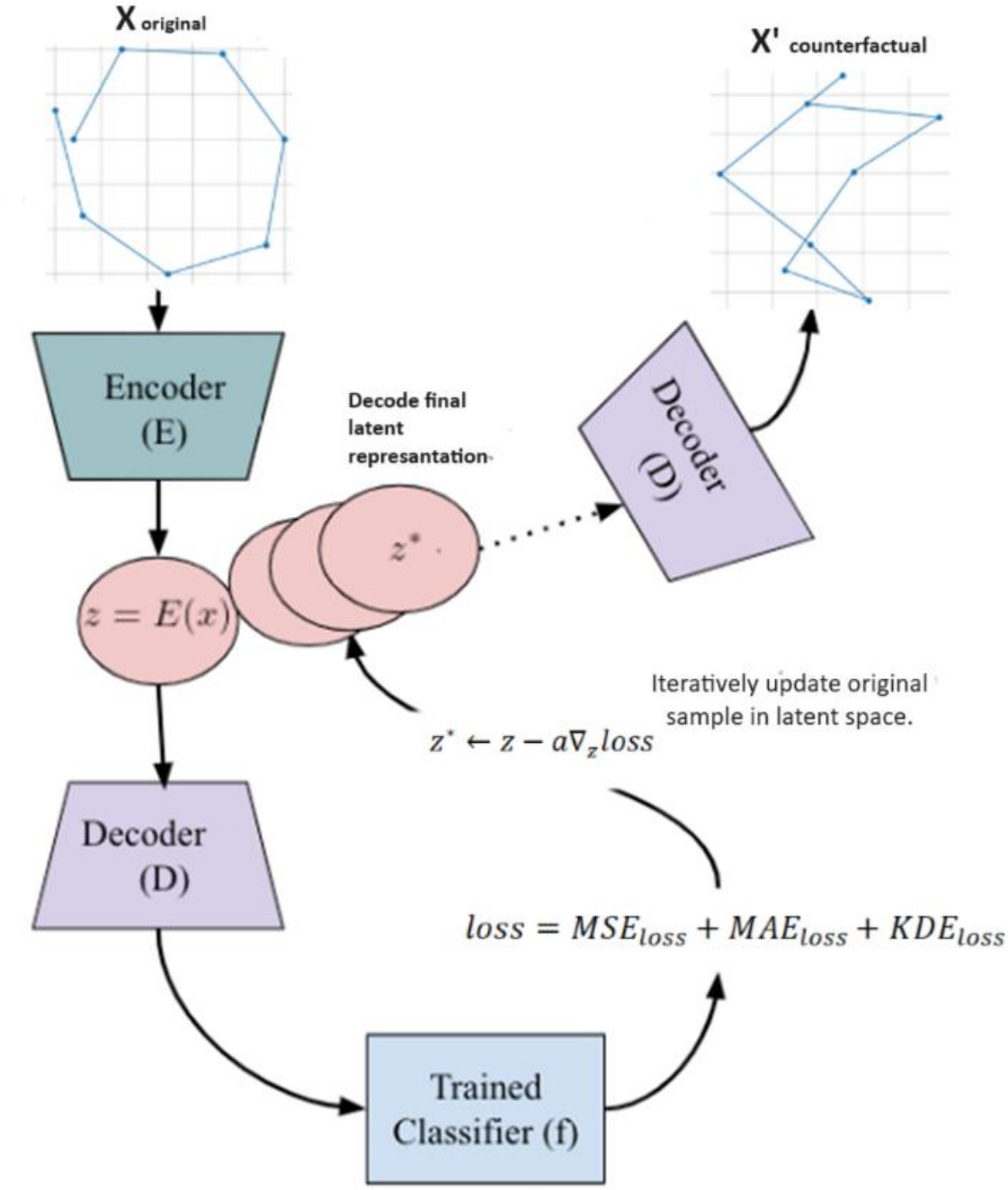
όπως έχει αναφερθεί και στο θεωρητικό μέρος.

Υπολογίζεται η μέση λογαριθμική τιμή της κατανομής που χρησιμοποιείται στη σύγκριση.

$$mean(\log(f(x)) = \frac{1}{n}\sum_{i=1}^{n}\log\left(f(x)\right)$$

Τέλος η απώλεια υπολογίζεται ως η διαφορά της κατανομής KDE του δεδομένου που υπόκειται αλλαγές και των δεδομένων που κατατάσσονται στην «ομαλή» κλάση, με σκοπό να ελαχιστοποιηθεί αυτή η διαφορά, πολλαπλασιασμένο με το ορισμένο από το χρήστη βάρος.

Σε αυτή τη μετρική το βάρος δεν προτείνεται να ξεπεράσει το 30%, καθώς με μεγαλύτερο βάρος, δεν είναι ξεκάθαρος ο στόχος της μεθόδου και παρατηρείται μια «χαοτική» έξοδος. Ένα αντιπαράδειγμα που από την μία δεν έχει αλλάξει κλάση αλλά από την άλλη προσπαθείτε η κατανομή να μοιάσει με αυτό της ομαλής.



Εικόνα: Παρόμοια με το (1), η αρχιτεκτονική της μεθόδου LatentCF++_KDE.

Βασικός Αλγόριθμος :

Ψευδογλώσσα του αλγορίθμου

Είσοδος:

- Πολυδιάστατο δεδομένο χρονοσειράς→ x,
- όριο απόφασης → τ,
- learning rate → a
- μέγιστη αποδεκτή απώλεια → tol
- όριο επαναλήψεων → max_iter
- decition boundaty → t (target)

"Εξοδος:

- αντιπαράδειγμα → x'

1. $z \leftarrow$ Encode(x)
2. $y_{pred} \leftarrow$ Predict(Decode(z))
3. loss ← loss_metric $[W_1 * f_1(y_{pred}, t) + W_2 f_2(X_{orig}, X_{cf}) + W_3 f_3(KDE_{norm}, KDE_{cf})]$
4. iter ← 0

while loss > tol **and** y_{pred} < t

and iter < max_iter **do**:

5. $z \leftarrow$ AdamOptimize(z, loss, α)
6. $y_{pred} \leftarrow$ Predict(Decode(z))
7. loss ← loss_metric
8. *iter* ← *iter* + 1
9. x' ← Decode(z)
10. return x'

Όπου loss_metric = $W_1 * f_1(y_{pred}, t) + W_2 f_2(X_{orig}, X_{cf}) + W_3 f_3(KDE_{norm}, KDE_{cf})$.

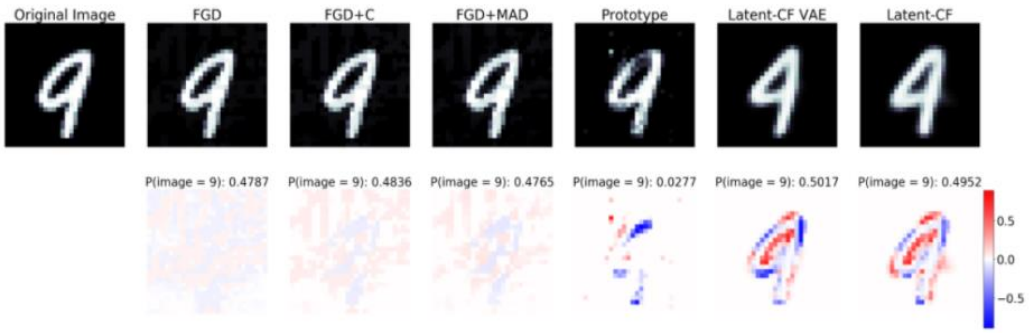
Στην Ψευδογλώσσα παρόμοια με την εργασία (2) φαίνεται η διαδικασία με την οποία αρχικά συμπιέζεται το δεδομένο. Στη συνέχεια υπολογίζεται η απώλεια που είναι το άθροισμα των τριών συναρτήσεων. Η βασική συνάρτηση είναι η $W_1 * f_1(y_{pred}, t)$ καθώς το W_1 λαμβάνει τιμές πάντα μεγαλύτερες ή ίσες με 0.6. Αυτό σημαίνει ότι το κυρίαρχο στοιχείο αξιολόγησής είναι το αν το δεδομένο άλλαξε ή όχι κλάση και αν εντέλει αποτελεί αντιπαράδειγμα. Στη λούπα της επανάληψης μπαίνει ο αλγόριθμος μέχρι κάποιο από τα τρία προ απαιτούμενα να ικανοποιηθούν. Είτε η απώλεια να γίνει μικρότερη από την οριακή απώλεια, είτε το αντιπαράδειγμα να ξεπεράσει το decision boundary, είτε να φτάσει το μέγιστος αριθμός επαναλήψεων.

Ιστορικό της μεθόδου

LatentCF (1)

Τον Ιούνιο του 2021 προτάθηκε η μέθοδος LatentCF που αποσκοπεί στην εύρεση αντιπαραδειγμάτων στο συμπιεσμένο χώρο ενός αυτοκωδικοποιητή. Πολύ παρόμοια με την μέθοδο που μελετάται σε αυτή την εργασία υπάρχει προ εκπαιδευμένος ταξινομητής και αυτοκωδικοποιητής που όμως χρησιμοποιεί Gradient Descent (FGD) για να ελαχιστοποιήσει το κόστος. Αποτελεί τη διατριβή που μελέτησε πρώτη την εύρεση αντιπαραδειγμάτων στο συμπιεσμένο χώρο και κατάφερε να αποδείξει ότι είναι πιο αποδοτική και πιο γρήγορη με τις μέχρι τώρα μεθόδους.

Παραθέτοντας παρακάτω τα αποτελέσματα της για να αναδειχθεί η ανάγκη ανάπτυξης αυτής της μεθόδου. Συγκεκριμένα συγκρίνονται οι μέθοδοι LatentCF και LatentCF VAE που αποτελεί μια παραλλαγή που αντικαθιστά τον αυτοκωδικοποιητή με ποικιλιακό αυτοκωδικοποιητή, με τις FGD (feature Gradient Descent) με διάφορες παραλλαγές και την εισαγωγή πρωτοτύπων (Prototypes) που χρησιμοποιεί αυτοκωδικοποιητές για να μετατρέψει τα δεδομένα σε ένα συμπιεσμένο χώρο, όπου καθορίζει "πρωτότυπα" για κάθε κλάση και, στη συνέχεια, προσεγγίζει αυτά τα πρωτότυπα για να δημιουργήσει αντιπαραδείγματα. (34)



Εικόνα: Σύγκριση της LatentCF με άλλες μεθόδους.

Παραθέτετε η σύγκριση της μεθόδου με κριτήριο αποτελεσματικότητας της LatentCF με προ υπάρχουσες μεθόδους δημιουργίας αντιπαραδειγμάτων χρησιμοποιώντας τη βάση δεδομένων αριθμών MNIST .

Τα αποτελέσματα των μοντέλων που φαίνονται στην εικόνα (αριθμός εικόνας) είναι η προσπάθεια τους να μετατρέψουν το 9 σε 4 και στην πραγματικότητα μόνο το LatentCF έχει ικανοποιητικά αποτελέσματα. Στην δεύτερη σειρά εικόνων σε μορφή heatmap φαίνονται οι αλλαγές που κάνει το εκάστοτε μοντέλο στο δεδομένο. Τα μοντέλα FGD, FGD+C, FGD+MAD έκαναν μικρές αλλαγές σε όλα σχεδόν τα pixels χωρίς να έχουν κριτήριο έτσι ώστε να μπορεί να αλλάξει το αποτέλεσμα του δεδομένου. Η Prototype κατάφερε πιο αποτελεσματικά να εντοπίσει που πρέπει να γίνουν οι αλλαγές αλλά δεν κατάφερε να κάνει βήματα τέτοια ώστε να γίνει όντως 4 και τελικώς «μουτζούρωσε» λίγο την εικόνα. Η LatentCF και η LatentCF VAE εντόπισε αποτελεσματικά και που πρέπει να γίνουν οι αλλαγές αλλά και είχε τη δυνατότητα να τις κάνει καθαρά και στενευμένα λόγο του ότι χειριζόταν μόνο μικρότερης διάστασης δεδομένα και μόνο τα χαρακτηριστικά pixels της εικόνας.

Method	dti	loan_amnt	int_rate	annual_inc	fico
FGD	-32 (32)	-37 (37)	-15 (18)	58 (59)	11 (11)
FGD+C	-31 (32)	-37 (37)	-15 (18)	58 (58)	11 (11)
FGD+MAD	-32 (32)	-37 (37)	-15 (18)	59 (59)	11 (11)
Prototype	0 (16)	-9 (25)	10 (16)	39 (43)	14 (14)
Latent-CF VAE	16 (23)	-1 (30)	2 (22)	-2 (25)	14 (14)
Latent-CF	-26 (29)	4 (26)	-6 (26)	24 (32)	13 (13)

Εικόνα (νουμερο εικόνας): LatentCF για tabular δεδομένα.

Στην εικόνα (αριθμός) παραθέτονται τα αποτελέσματα του (1) για tabular δεδομένα. Συγκεκριμένα, η μέση ποσοστιαία αλλαγή που έκανε ή κάθε μέθοδος στο κάθε χαρακτηριστικό των δεδομένων.

Συγκεκριμένα στην εικόνα (νούμερο εικόνας) μπορούμε να κατανοήσουμε τις κινήσεις της LatentCF σε σύγκρισή με άλλες μεθόδους που έχουν μεγάλες διαφορές. Οι μέθοδοι FGD, FGD+C, FGD+MAD δημιουργούνε πολύ μεγάλες αλλαγές που έχει ως αποτέλεσμα να ανήκει το δεδομένο στην επιθυμητή αντίθετη κλάση αλλά οι αλλαγές να είναι τόσο μεγάλες που να μην είναι πραγματοποιήσιμες για τους ανθρώπους που καλούνται να υλοποιήσουν τις προτάσεις των αλγορίθμων. Οι αλλαγές αυτές γιατί όπως έχει προαναφερθεί είναι πολύ σημαντικό να είναι φιλικά προσκείμενες προς τον άνθρωπο. Οι άλλες τρείς μέθοδοι που είναι εμφανές ότι κατανοούνε περισσότερο την κατανομή των δεδομένων δεν κάνουν άλματα στα βήματα τους αλλά αντιθέτους κάνουν αλλαγές μικρότερες και πιο στενευμένες.

LatentCF++ (2)

Σε συνέχεια της μεθόδου LatentCF ήρθαν τον Οκτώβρη του 2021 και έγινε προσπάθεια βελτίωσης της καθώς είδη αποδεδειγμένα είχε προοπτικές. Η βελτιωμένη εκδοχή ονομάστηκε LatentCF++ η οποία εισήγαγε τον αλγόριθμο βελτιστοποίησης Adam καθώς επέφερε καλύτερες αποδόσεις από ότι ο Gradient Descent. Ο αλγόριθμος Adam κινείται πιο στενευμένα στο ολικό ελάχιστο και αποφεύγει να «κολλάει» σε τοπικά ελάχιστα.

Τα αποτελέσματα τους βασίζονται σε ένα σύνολο 40 βάσεων δεδομένων αποτελούμενα από μονοδιάστατες χρονοσειρές. Οι 20 βάσεις δεδομένων, από τις 40, είναι εκείνες που έχουν περισσότερο από 500 δεδομένα και κωδικοποιούνται ως Subset avg, ενώ οι υπόλοιπες ως Total avg. Τα δεδομένα που χρησιμοποιήθηκαν ποικίλουν από σήματα εγκεφάλου μέχρι σήματα μηχανής αυτοκινήτου.

Οι μέθοδοι με τις οποίες συγκρίνεται η LatentCF++ είναι 1) η LatentCF, 2) η FGD (feature Gradient Descent), 3) η k-NN (k-κοντινότεροι γείτονες) και 4) η RSF (Random Survival Forest). Παρακάτω παρατίθενται οι πίνακες σύγκρισης για τρείς μετρικές. Το Validity, το Margin Difference και το Proximity που είναι μετρικές οι οποίες χρησιμοποιούνται και στην έρευνα αυτής της εργασίας και είναι αναλυτικά εξηγημένες παραπάνω.

Validity

	LatentCF++			LatentCF		Baseline		
	1dCNN	LSTM	1dCNN-C	1dCNN	LSTM	FGD	k-NN	RSF
Subset avg.	0.9841	0.8929	0.9720	0.2703	0.6058	0.1548	0.9180	**1.0000**
Total avg.	**0.9920**	0.8256	0.9615	0.1676	0.5779	0.0774	0.9496	0.9802

Πίνακας (αριθμός): Validity της LatentCF++.

Παρουσιάζεται η αποτελεσματικότητα των μεθόδων σε σχέση με το validity δηλαδή την ικανότητα του δεδομένου να αλλάξει αποτελεσματικά κλάση. Εδώ είναι ξεκάθαρο ότι τα καλύτερα μοντέλα είναι το LatentCF++, το k-NN και το RSF.

Margin Difference

	LatentCF++			LatentCF		Baseline		
	1dCNN	LSTM	1dCNN-C	1dCNN	LSTM	FGD	k-NN	RSF
Subset avg.	**0.0333**	0.0736	-0.0049	-0.0929	0.1196	-0.1973	0.3530	0.0811
Subset std.	0.0348	**0.0246**	0.0287	0.2072	0.0453	0.1848	0.1600	0.0623
Total avg.	**0.0168**	0.0580	-0.0120	-0.0896	0.0520	-0.1435	0.3608	0.0614
Total std.	**0.0175**	0.0234	0.0310	0.1193	0.0240	0.1081	0.1218	0.0576

Πίνακας (αριθμός): Το μέτρο σύγκρισης εδώ είναι το Margin Difference που αντιπροσωπεύει την απόσταση του αντιπαραδείγματος από το όριο απόφασης (decision boundary τ). Αρνητικός αριθμός είναι αυτός που ο μέσος όρος των δεδομένων δεν άλλαξε καν κλάση και δεν αποτελεί αντιπαράδειγμα. Εδώ τα καλύτερα αποτελέσματα τα παρουσιάζει το LatentCF++ καθώς είναι συγκρίσιμα με την RSF.

Proximity

	LatentCF++			LatentCF		Baseline		
	1dCNN	LSTM	1dCNN-C	1dCNN	LSTM	FGD	k-NN	RSF
Subset avg.	0.3891	2.2441	0.5088	0.1963	1.3451	0.0169	0.6592	**0.2873**
Total avg.	0.8926	2.5409	0.9179	0.4415	1.9841	0.0087	1.4613	**0.5241**

Πίνακας(αριθμός): Γίνεται αναπαράσταση των αποτελεσμάτων σε σχέση με το Proximity, δηλαδή την απόσταση μεταξύ του αντιπαραδείγματος και του αρχικού δεδομένου. Εδώ τα καλύτερα αποτελέσματα τα παρουσιάζει η μέθοδος RSF αλλά η LatentCF++ (εκτός από αυτήν που χρησιμοποιεί LSTM αυτοκωδικοποιητή) έχουν τα δεύτερα καλύτερα αποτελέσματα.

Εν κατακλείδι ή LatentCF++ παρουσιάζει άπταιστα στατιστικά δεδομένα αρκετά συγκρίσιμα με την RSF που όμως εξηγείται διότι η χρήση αυτοκωδικοποιητών

κάνει τη μέθοδο να εξαρτάται πολύ από την απώλεια στα δεδομένα κατά την αναδιαμόρφωση τους στον αρχικό χώρο. Αυτό καταλήγει στην δυσκολία του LatentCF++ να μπορεί να γενικεύει το ίδιο μοντέλο σε 40 βάσεις δεδομένων και ανάλογα τη βάση δεδομένων είναι σημαντικό να γίνεται εύρεση του αυτοκωδικοποιητή με τη λιγότερη απώλεια. Έτσι η LatentCF++ θα μπορεί να παρουσιάσει τα καλύτερα αποτελέσματα στο σύνολο των μεθόδων.

Στήσιμο πειράματος

Το πειραματικό μέρος αυτής της εργασίας στηρίζεται στη σύγκριση δύο μεθόδων της LatentCF++. Την μέθοδο που παρουσιάζεται στην εργασία αυτή και την μέθοδο προτεινόμενη από τους Zhendong Wang, Rami Mochaourab και Παναγιώτης Παπαπέτρου στην εργασία (2) και της βελτιωμένης εκδοχής που προτείνεται στην εργασία αυτή με την εισαγωγή της σύγκρισης κατανομών. Το στήσιμο είναι όμοιο και για τις δύο πλευρές. Δύο μοντέλα, ένας ταξινομητής και ένας αποκωδικοποιητής προ εκπαιδεύονται στα σετ δεδομένων. Τα σετ δεδομένων δίνονται στο κωδικοποιητή για συμπιέσει τις διαστάσεις τους και ακολουθεί η διαδικασία αλλαγής των δεδομένων με στόχο τη ελαχιστοποίηση της συνάρτησης την απώλειας. Αποκωδικοποιούνται και γίνεται evaluate. Για ευκολία θα ονομάσουμε την μέθοδο των Zhendong Wang, Rami Mochaourab, Παναγιώτης Παπαπέτρου ως LatentCF++ και την προτεινόμενη από την εργασία αυτή LatentCF++_KDE. Η μόνη διαφορά των δύο μεθόδων είναι η προσθήκη της συνάρτησης KDE στη συνάρτηση απώλειας η οποία όμως όπως προτείνεται στην εργασία αυτή είναι αρκετά βοηθητική στην διατήρηση της κατανομής των δεδομένων και στην αποφυγή δεδομένων που απλώς ικανοποιούν τον ταξινομητή.

Τα μοντέλα προς χρήση στο πείραμα

Για την περαιτέρω μελέτη της μεθόδου LatentCF++ χρησιμοποιούνται δυο συνδυασμοί ταξινομητών και αυτοκωδικοποιητών καθώς διαφορετικά δεν θα ήταν ξεκάθαρο τί ευθύνεται για τα αποτελέσματα, τα μοντέλα ή η μέθοδος παραγωγής αντιπαραδειγμάτων.

	Ταξινομητής	Αυτοκωδικοποιητής	Αλγόριθμος βελτιστοποίησης
LatentCF++	1D CNN - Classifier	1D CNN-Autoencoder	Adam
LatentCF++	LSTM - Classifier	LSTM- Autoencoder	Adam
LatentCF++_KDE	1D CNN - Classifier	1D CNN-Autoencoder	Adam
LatentCF++_KDE	LSTM - Classifier	LSTM- Autoencoder	Adam

Πίνακας(αριθμος): Τα μοντέλα προς σύγκριση.

Στο κεφάλαιο 2.3.1 θα εξηγήθούν οι δομές των δύο ταξινομιτών και στο 2.3.2 οι δομές των δύο αυτοκωδικοποιητών τα οποία δημιουργήθηκαν και εκπαιδευτηκαν χρησιμοποιόντας τις βιβλιοθήκες Python TensorFlow και Keras.

Δομή ταξινομητών.

1D-CNN Ταξινομητής

Κατασκευάζεται ένα νευρωνικά δίκτυο βαθιάς μάθησης που χρησιμοποιεί συνελικτικά επίπεδα (Convolutional Layers) και είναι κατάλληλος για την ταξινόμηση χρονοσειρών. Η χρήση των συνελικτικών επιπέδων καθιστά το μοντέλο αυτό κατάλληλο για την αναγνώριση τοπικών μοτίβων που χαρακτηρίζουν τα δεδομένα, ενώ χρησιμοποιείται ένα επίπεδο μέγιστης δειγματοληψίας με σκοπό να μειώσει τις διαστάσεις, διατηρώντας ταυτόχρονα τα σημαντικά χαρακτηριστικά τους. Παρακάτω παρατίθεται μια λεπτομερής περιγραφή.

1) Αρχικά υπάρχει ένα στρώμα **εισόδου** που δέχεται δεδομένα διαστάσεων (n_timesteps, n_features) δηλαδή (αριθμός στιγμών, αριθμός διαστάσεων).

2) Ένα **πυκνό** επίπεδο (Dence layer) εφόσον έχει ενεργοποιηθεί από το χρήστη, που αποτελείται από 128 νευρώνες και χρησιμοποιεί regularization methods που είναι μεθόδου για την αποφυγή του υπερεκπαίδευσης (την αδυναμία του μοντέλου να γενικεύσει) .

3) Δύο επίπεδα που εφαρμόζουν την **συνελικτική** λειτουργία της εισόδου και επαναλαμβάνονται όσες φορές έχει οριστεί από τον χρήστη μέσω της μεταβλητής n_conv_layers.

    1. Μονοδιάστατο **συνελικτικό** στρώμα με filters = 64 και kernel_size = (3, 3).
    2. **Bach Normalization** στρώμα που κανονικοποιεί τις τιμές σε κάθε δέσμη για να προσαρμόζεται καλύτερα το νευρωνικό δίκτυο στα δεδομένα.
    3. **Συνάρτηση ενεργοποίησης** ReLu.

4) Ένα στρώμα **μέγιστης συσσώρευσης** (MaxPooling1D) το οποίο μειώνει τις διαστάσεις παίρνοντας την μέγιστη τιμή κάθε (2,2) παραθύρου.

5) Ένα επίπεδο **Εξομάλυνσης** (Flatten) που μετατρέπει την δισδιάστατή είσοδο σε μονοδιάστατη ώστε να μπορεί να συνδεθεί με τα πυκνά επίπεδα.

6) Την **Έξοδο** που χρησιμοποιεί μια συνάρτηση ενεργοποίησης, η οποία ταξινομεί τα δεδομένα. Επιλέγει ανάμεσα σε Softmax για ταξινόμησης δεδομένων με πολλές κλάσεις, είτε sigmoid για δύο κλάσεις.

LSMT Ταξινομητής

Εκπαιδεύεται ένας ταξινομητής μακράς και βραχείας μνήμης σχεδιασμένος για την ταξινόμηση χρονοσειρών.
Αναλυτικά περιέχει:

1) Ένα στρώμα **εισόδου** όπου το δίκτυο δέχεται δεδομένα διαστάσεων (n_timesteps, n_features) δηλαδή (αριθμός διαδοχικών βημάτων, αριθμός παραμέτρων).

2) Ένα **LSTM** επίπεδο εφόσον έχει ενεργοποιηθεί από το χρήστη, που αποτελείται από 64 νευρώνες και χρησιμοποιεί τη συνάρτηση ενεργοποίησης **tanh**.

3) Εάν δεν ενεργοποιηθεί από το χρήστη το παραπάνω στρώμα χρησιμοποιείται ένα **LSTM** επίπεδο με 32 νευρώνες στη θέση του.

4) Ένα **Bach Normalization** στρώμα που κανονικοποιεί τις τιμές σε κάθε δέσμη για να προσαρμόζεται καλύτερα το νευρωνικό δίκτυο στα δεδομένα.

5) Το Τελευταίο **LSTM** επίπεδο με 16 νευρώνες που επιστρέφει ακολουθίες στη μορφή που είναι απαραίτητη για την τελική ταξινόμηση.

6) Το Επίπεδο **εξόδου** που χρησιμοποιεί την σιγμοειδή συνάρτηση ενεργοποίησης για δυαδική ταξινόμηση ή Softmax αν υπάρχουν παραπάνω από δύο διαστάσεις.

Δομή Αυτοκωδικοποιητών

1D-CNN Αυτοκωδικοποιητής

Ο πρώτος αυτοκωδικοποιητής χρησιμοποιεί συνελικτικά επίπεδα για να αναγνωρίζει χαρακτηριστικά και επίπεδα δειγματοληψίας για την αλλαγή διαστάσεων (για την μείωση τους στον κωδικοποιητή και την αύξησή του στον αυτοκωδικοποιητή μέχρι να φτάσει τις αρχικές διαστάσεις). Κατά την εκπαίδευση του μοντέλου αυτού χρησιμοποιείται η συνάρτηση βελτιστοποίησης Adam.

1. Κωδικοποιητής

Δέχεται σαν είσοδο ένα δεδομένο τη φορά με διαστάσεις (n_timesteps, n_features) και επιστρέφει ως έξοδο το δεδομένο σε συμπιεσμένη μορφή:

- Ένα **συνελικτικό** επίπεδο που λειτουργεί στα δεδομένα εισόδου, αποτελείται από 64 φίλτρα με μέγεθος πυρήνα (3,3) και συνάρτηση ενεργοποίησης ReLu.
- ‘Ενα επίπεδο **δειγματοληψίας** που μειώνει τις διαστάσεις των δεδομένων παίρνοντας την μέγιστη τιμή από κάθε παράθυρο (2,2) διαστάσεων του πίνακα δεδομένων.

- Ένα ακόμη **συνελικτικό** επίπεδο με τα μισά φίλτρα από ότι πριν, ίσα με 32.
- Και τέλος ένα επίπεδο **δειγματοληψίας** ακριβώς όπως το προηγούμενο για να ξαναειπωθούν στα μισά οι διαστάσεις.

2. Αποκωδικοποιητής

Δέχεται σαν είσοδο το συμπιεσμένο δεδομένο που επέστρεψε ο κωδικοποιητής και ξαναχτίζει τα δεδομένα έτσι ώστε να φτάσει τις αρχικές διαστάσεις.
Αποτελείται από:

- Ένα **συνελικτικό** επίπεδο με φίλτρα 32, πυρήνα (3,3) και συνάρτηση ενεργοποίησης ReLu.
- Ένα επίπεδο **δειγματοληψίας** που διπλασιάζει το μέγεθος τον δεδομένων. Αντιστρέφει την λειτουργία των δειγματοληπτικών επιπέδων του κωδικοποιητή.

- Ένα **συνελικτικό** επίπεδο με φίλτρα όσο ο αρχικών αριθμός των παραμέτρων έτσι ώστε να φτιαχτεί η δεύτερη διάσταση όπως ήταν αρχικά.
- Ένα επίπεδο **δειγματοληψίας** που διπλασιάζει τα δεδομένα έτσι ώστε να είναι οι χρονικές στιγμές όσες ήταν αρχικά.

LSTM Αυτοκωδικοποιητής

Ο αυτοκωδικοποιητής που παρουσιάζεται χρησιμοποιεί επίπεδα LSTM για να αναγνωρίζει τα χαρακτηριστικά από ακολουθίες δεδομένων και να ανακατασκευάζει τα αρχικά δεδομένα από τη συμπιεσμένη αναπαράστασή τους. Η εκπαίδευση του μοντέλου γίνεται χρησιμοποιώντας την συνάρτηση βελτιστοποίησης Adam για την εκμάθηση των βαρών.

Λεπτομερής ανάλυση των επιπέδων του μοντέλου:

1. Κωδικοποιητής

Δέχεται είσοδο με διαστάσεις (n_timesteps, n_features) και διαδοχικά εφαρμόζει δύο επίπεδα LSTM:

- Ένα επίπεδο **LSTM** με 64 μονάδες και συνάρτηση ενεργοποίησης "tanh", που διατηρεί τις ακολουθίες στην έξοδο για την επεξεργασία στο επόμενο επίπεδο.

- Ένα επίπεδο **LSTM** με 32 μονάδες και συνάρτηση ενεργοποίησης "tanh", που παράγει τη συμπιεσμένη αναπαράσταση των αρχικών δεδομένων και τις επιστρέφει ως έξοδο.

2. Αποκωδικοποιητής

Δέχεται τη συμπιεσμένη αναπαράσταση από τον κωδικοποιητή και διαδοχικά εφαρμόζει:

- Ένα επίπεδο **RepeatVector** για την επαναφορά της διάστασης ακολουθίας με βάση τον αριθμό των χρονικών βημάτων (n_timesteps).
- Ένα επίπεδο **LSTM** με 32 μονάδες και συνάρτηση ενεργοποίησης "tanh".
- Τέλος, ένα επίπεδο **LSTM** με 64 μονάδες Επεκτείνει την ικανότητα του μοντέλου να αποκωδικοποιεί πιο περίπλοκες σχέσεις μέσα στα δεδομένα, διατηρώντας παράλληλα την χρονική ακολουθία.

Μετρικές

Σε αυτή την ενότητα θα αναλυθούν οι μετρικές με τις οποίες αξιολογείται η κάθε μέθοδος και εν τέλη ο τρόπος με τον οποίο θα γίνει σύγκριση και ανάλυση των αποτελεσμάτων.

Εγκυρότητα

Αρχικά χρησιμοποιείται η μετρική της εγκυρότητας (validity). (2) Εκφράζει την τον αριθμό των δεδομένων που επιτυχημένα έγιναν αντιπαραδείγματα και αλλάξανε κλάση. Θεωρούμε την κλάση 0 την ανώμαλη και κλάση 1 την ομαλή κλάση. Υπολογίζεται ο αριθμός των δεδομένων που ανήκουν στην κλάση 1 (από την κλάση 0) και διαιρείται με τον συνολικό αριθμό των δεδομένων.

$$Validity\left(y_{cf},\tau\right)=\frac{\sum_{i=1}^{N}y_i\ \ (where\ y_i\geq\tau,y_i\in y_{cf})}{N}$$

Όπου $(y_1,y_2,y_3,\dots,y_N)\in y_{cf}$ με y_{cf} να είναι το σύνολο των εξόδων της μεθόδου ασχέτων αν όντως αλλάξανε κλάση ή όχι και N εκφράζει τον συνολικό αριθμό των δεδομένων.

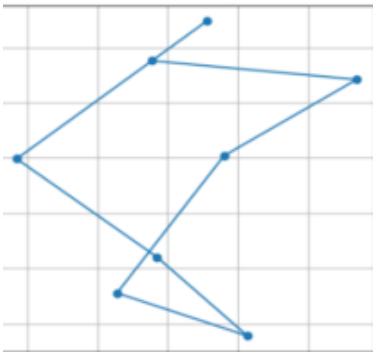
Η εγκυρότητα είναι η κυρίαρχη μετρική καθώς αν η εγκυρότητα δεν βγάλει ικανοποιητικά αποτελέσματα η μέθοδος είναι αυτομάτως αναποτελεσματική.

3.4.2) Περιθωριακή διαφορά (margin difference)

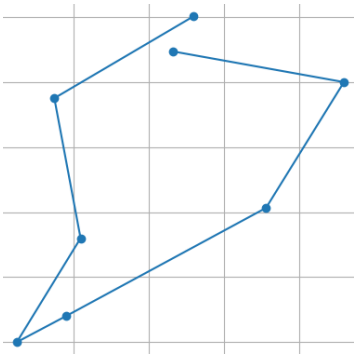
Η περιθωριακή διαφορά μετράει την απόσταση του αντιπαραδείγματος από το decision boundary που συμβολίζεται ως τ (2).

$$margin\_diff\left(y_{cf},d\right)=y_{cf}-\tau$$

Το decision boundary είναι το σημείο εκείνο στο χώρο όπου η μέθοδος «ψάχνει» για αντιπαράδειγμα. Στην ουσία είναι ένας αριθμός στο διάστημα [0,1] και αποτελεί όριο, απαιτείται δηλαδή το δεδομένο να ξεπεράσει το decision boundary για να το θεωρείται αντιπαράδειγμα. Στις περισσότερες περιπτώσεις ταυτίζεται με το 0.5, που είναι το σημείο τομής των κλάσεων, καθώς απαιτούνται οι λιγότερο δυνατές αλλαγές. Έτσι η μέθοδος επιστρέφει το δεδομένο με το που ξεπεράσει την πιθανότητα 0.5. Υπάρχουν όμως και περιπτώσεις που δεν είναι θεμιτό να είναι κατά 50% σίγουρος ο ταξινομητής αν ανήκει στην ομαλή κλάση, καθώς τα αντιπαραδείγματα δεν είναι ακόμα σωστά σχηματισμένα όπως στο παράδειγμα παρακάτω.



Εικόνα (αριθμός)



Εικόνα (αριμός)

Εικόνες (αρ της δεξ, αρ της αρ): Στις εικόνες αυτές απεικονίζεται δύο [8] που αποτελούν αντιπαραδείγματα προερχόμενα από [0]. Η πρώτη εικόνα απεικονίζει ένα αντιπαράδειγμα που υπολογίστηκε με τ = 0.9 και η δεύτερη με τ = 0.5. Είναι ξεκάθαρο ότι στη δεύτερη εικόνα παρόλο που το παράδειγμα θεωρείται επιτυχημένο καθώς ταξινομείται ως 8 (οριακά) λόγο των αναγκών της συγκεκριμένης βάσης δεδομένων δεν είναι.

Στην συγκεκριμένη εργασία χρησιμοποιείται και η απόλυτη τιμή των περιθωριακών διαφορών για τον υπολογισμό της απόστασης από το decision boundary αλλά και χωρίς απόλυτη τιμή για τον καθορισμό της κατεύθυνσης. Άμα η μετρική χωρίς την απόλυτη τιμή είναι αρνητική, ακόμα και αν είναι μικρή, τότε το δεδομένο δεν άλλαξε κλάση καθώς είναι κάτω από το όριο και αυτομάτως καθιστά το σετ δεδομένων με την μέθοδο μη επιτυχημένο.

Proximity

Σύμφωνα με την διατριβή (2) ως Proximity ορίζεται η ευκλείδεια απόσταση μεταξύ του αντιπαραδείγματος X′ και του αρχικού δεδομένου X.

$$SE(X', X) = \sqrt{\sum_{i=1}^{t} (X'[t] - X[t])^2}$$

Όπου SE ονομάζουμε την ευκλείδεια απόσταση μεταξύ των πινάκων που εκφράζουν τα δεδομένα X και X’ (SE απο την φράση Square Error), t είναι ή κάθε χρονική στιγμή των δεδομένων και έπειτα υπολογίζεται η μέση τιμή την ευκλείδειας απόστασης:

$$\frac{1}{n}\sum_{i=1}^{n} SE(Xi', Xi)$$

Όπου n είναι το μέγιστο πλήθος των δεδομένων, i χαρακτηρίζει το δεδομένο για το οποίο υπολογίζεται το Proximity σε κάθε επανάληψη, Xi είναι το αρχικό δεδομένο και Xi’ είναι το αντίστοιχο αντιπαράδειγμα.

Σκοπός αυτής της μετρικής είναι να υπολογιστεί η απόσταση που έχει διανύσει το δεδομένο μέχρι να γίνει αντιπαράδειγμα, καθώς ένα καλό αντιπαράδειγμα ορίζεται εκείνο το οποίο ενώ έχει αλλάξει κλάση απέχει την ελάχιστη δυνατή απόσταση από το αρχικό.

KDE *diffrence*

Τέλος, σύμφωνα με τη διατριβή (1), ορίζεται η μετρική της διαφοράς των KDE κατανομών. Για την υλοποίηση αυτής της μετρικής αρχικά ορίζεται ο τύπος της πυκνότητας-πιθανότητας ή αλλιώς πυρήνας των ομαλών και των ανώμαλων δεδομένων. Σκοπός είναι να υπάρχει μια τιμή που θα αναδεικνύει αν τα αντιπαραδείγματα έχουν κατανομή πιο όμοια με τα ομαλά ή τα ανώμαλα δεδομένα. (Ένα καλό αντιπαράδειγμα έχει κατατομή πιο κοντά στα ομαλά). Έτσι ορίζεται ο πυρήνας K ως εξής:

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

Στη συνέχεια υπολογίζεται ο τύπος:

$$f(x) = \frac{1}{h}\sum_{i=1}^{2} K\left(\frac{x - x_i}{h}\right)$$

Όπου x είναι το σύνολο των δεδομένων , x_i είναι το επιλεγμένο δεδομένο που επιθύμείται να βρεθεί η κατανομή του σε σχέση με την υπόλοιπη βάση δεδομένων και h ονομάζεται το bandwidth το οποίο εφράζει πόσο ομαλή η απότομή θα είναι η γραφική της κατανομής

Με αυτόν τον τρόπο ορίζεται ένας πυρήνας για τα ομαλά και ανώμαλα δεδομένα και έπειτα με τον τύπου (αριθμός τελευταίου τύπου) παράγεται μια καινούργια συνάρτηση για την οποία υπολογίζεται η λογαριθμική της τιμή. Η λογαριθμική τιμή, υπολογίζεται καθώς «κανονικοποιεί» το αποτέλεσμα. Τελικώς αυτή είναι που ορίζει ποσό κοντά ή όχι είναι τα αντιπαραδείγματα στην ομαλή ή αντίστοιχα την ανώμαλη κλάση. Έπειτα, καθώς με αυτόν τον τρόπο υπάρχει μία λογαριθμική τιμή για κάθε αντιπαράδειγμα, υπολογίζεται ο μέσος όρος τους.

$$mean(\log(f(x)) = \frac{1}{n}\sum_{i=1}^{n} \log\left(f(x)\right)$$

Αυτή η διαδικασία πραγματοποιείται για κάθε διάσταση ξεχωριστά για να γίνεται η σύγκριση μεταξύ αντίστοιχων διαστάσεων και να αποφευχθεί ένα μη διαχειριστικό συνονθύλευμα. Δοκιμάστηκε για όλες τις διαστάσεις ταυτόχρονα αλλά τα αποτελέσματα δεν ήταν ενδεικτικά.

Να σημειωθεί ότι όσο πιο κοντά στο μηδέν είναι η τιμή της λογαριθμικής κατανομής τόσο περισσότερο «μοιάζουν» τα δεδομένα, όμως το βασικό κριτήριο είναι αυτό της σύγκρισης. Άμα η λογαριθμική τιμή είναι μικρότερη όταν συγκρίνονται τα αντιπαραδείγματα με τα δείγματα της ομαλής κλάσης, παρά όταν την συγκρίνονται με την ανώμαλη τότε το συμπέρασμα είναι ότι τα δεδομένα μοιάζουν περισσότερο με την ομαλή κλάση που είναι και το ζητούμενο.

Ανάλυση των δεδομένων

Σετ δεδομένων

Σε αυτήν την ενότητα αναλύονται τα τρία σετ δεδομένων που χρησιμοποιήθηκαν για το πείραμα αυτήν της εργασίας. Τα τρία σετ δεδομένων είναι τα εξής:

PenDigits, WalkingSittingStanding, FingerMovemets

Συνολικά η μέθοδος υλοποιήθηκε με μια σειρά συνδυασμών που αφορούν τα διαφορετικά dataset και τα μοντέλα μηχανικής μάθησης (ταξινομητή και αυτοκωδικοποιητή) ως εξής:

	Ταξινομιτής	Αυτοκωδικοποιητής	Σετ Δεδομένων
LatentCF++	1D CNN - Classifier	1D CNN-Autoencoder	PenDigits
LatentCF++	LSTM - Classifier	LSTM- Autoencoder	PenDigits
LatentCF++_KDE	1D CNN - Classifier	1D CNN-Autoencoder	PenDigits
LatentCF++_KDE	LSTM - Classifier	LSTM- Autoencoder	PenDigits
LatentCF++	1D CNN - Classifier	1D CNN-Autoencoder	WalkingSittingStanding
LatentCF++	LSTM - Classifier	LSTM- Autoencoder	WalkingSittingStanding
LatentCF++_KDE	1D CNN - Classifier	1D CNN-Autoencoder	WalkingSittingStanding
LatentCF++_KDE	LSTM - Classifier	LSTM- Autoencoder	WalkingSittingStanding
LatentCF++	1D CNN - Classifier	1D CNN-Autoencoder	FingerMovemets
LatentCF++	LSTM - Classifier	LSTM- Autoencoder	FingerMovemets
LatentCF++_KDE	1D CNN - Classifier	1D CNN-Autoencoder	FingerMovemets
LatentCF++_KDE	LSTM - Classifier	LSTM- Autoencoder	FingerMovemets

Πίνακας αριθμός: Συνδυασμοί μεθόδων με μοντέλα του πειράματος.

Pen-digits

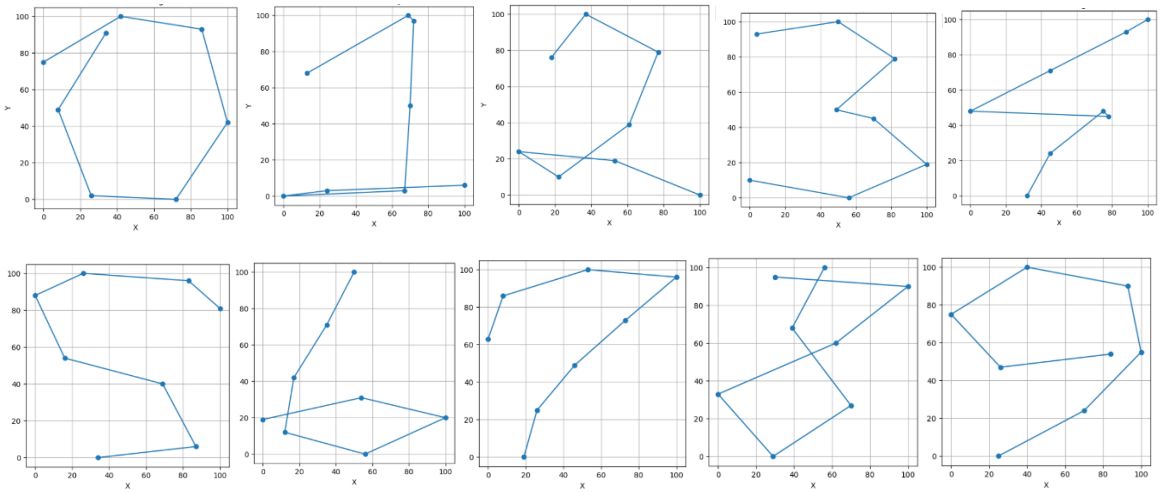
Αποτελεί την πρώτη βάση δεδομένων που χρησιμοποιήθηκε για την υλοποίηση της μεθόδου και των αλλαγών που υπέστη. Τα αποτελέσματα των συγκεκριμένων

δεδομένων PenDigits είναι εύκολα στην κατανόηση και την ανάλυση διότι είναι δεδομένα που οπτικοποιούνται.

Η βάση δεδομένων αυτή πάρθηκε από το UCI Archive και βρίσκεται στο timeseriesclassification.com (35), ονομάζεται PenDigits και αντικατοπτρίζει χειρόγραφα ψηφία. Οι διαστάσεις των δεδομένων είναι x και y συντεταγμένες 8 σημείων στο χώρο για κάθε ψηφίο. Αρχικά, καταγράφτηκαν σε εικόνες και στη συνέχεια πραγματοποιήθηκε επαναδειγματοληψία χωρικά, ώστε αντί οι x και y διαστάσεις να είναι συνεχόμενες, να έχουν χωρικά σταθερό βήμα. Οι ετικέτες ανήκουν στο σύνολο κλάσεων [0...9] που αντιστοιχεί το ψηφίο που σχεδιάστηκε. Έτσι είναι δυνατός ο σχεδιασμός συντεταγμένων στο αξονικό σύστημα συντεταγμένων και εμφανίζεται ψηφίο που μοιάζει με εικόνα ώστε να οπτικοποιείται εύκολα και σίγουρα.

Dataset size	**Length**	**Number of Classes**	**Number of Dimensions**	**Type**
10.992	8	10	2	(x,y) coordinates

Πίνακας αριθμός: Τα χαρακτηριστικά του Dataset PenDigits.



Εικόνα αριθμός: 10 παραδείγματα δεδομένων από κάθε κλάση

Στην εικόνα αριθμό φαίνονται σχεδιασμένα πάνω στο καρτεσιανό σύστημα αξόνων τα ψηφία [0,9] από το PenDigits Dataset. Ο πρώτος λόγος που επιλέχθηκε το

συγκεκριμένο σετ δεδομένων είναι λόγο της ευκολίας του καθώς έχει δύο διαστάσεις από 8 συντεταγμένες στο σετ δεδομένων (σύνολο 16 σε κάθε δεδομένο). Αυτό έχει ως αποτέλεσμα να είναι υπολογιστικά αποδοτικό και καθιστά δυνατή την εκτέλεση του κώδικα επαναλαμβανόμενα, έτσι ώστε να διορθώνεται με κάθε βήμα η μέθοδος. Ο δεύτερος, και πιο σημαντικός, λόγος είναι το γεγονός ότι μπορεί να αξιολογηθεί από το ανθρώπινο μάτι, καθώς είναι εύκολο και κατανοητό το αποτέλεσμα.

Η βασική μετατροπή που έχει μελετηθεί (τυχαία επιλεγμένο) είναι η μετατροπή του ψηφίου [0] (ανώμαλο) στο ψηφίο [8] (ομαλό).

WalkingSittingStanding

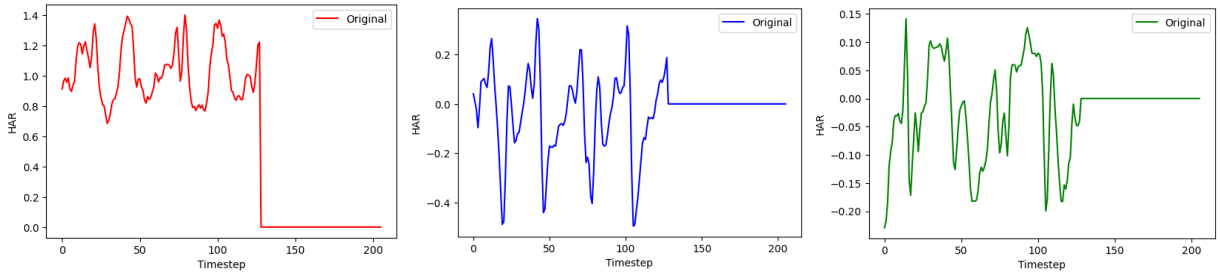
Το επόμενο σετ δεδομένων πραγματεύεται την αναγνώριση της ανθρώπινης δραστηριότητας (Human Activity Recognition - HAR) από το αρχείο UCR. Πρόκειται για μία συλλογή δεδομένων παρμένη από 30 εθελοντές που ήταν υγιείς ηλικίας 19 έως 48 ετών. Οι κλάσεις είναι έξι και είναι:

1. Περπάτημα
2. Ανάβαση σκαλοπατιών
3. Κατάβαση σκαλοπατιών
4. Κάθισμα
5. Όρθια στάση
6. Ξάπλωμα

Στους συμμετέχοντες είχαν δοθεί smartphones που διέθεταν αισθητήρες ικανούς να μετρούν τρείς άξονες της γραμμική επιτάχυνση και τρείς άξονες της γωνιακής ταχύτητας με ρυθμό 50 Hz. Έτσι περιλάμβανε αρχικά έξι κανάλια πληροφοριών αλλά υπέστη προ επεξεργασία για να απλοποιηθεί σε τρία κανάλια εστιάζοντας ειδικά στην γραμμική επιτάχυνση του σώματος. Συνολικά υπάρχουν 10.299 περιπτώσεις με τρείς διαστάσεις και 206 χρονικά βήματα σε κάθε διάσταση και έξι κλάσεις στις οποίες ανήκουν τα δεδομένα. Έτσι, καθώς στην περίπτωση της μελέτης αυτή είναι χρήσιμες μόνο δύο κλάσεις παρόμοια με το σετ δεδομένων PenDigits, διαλέγουμε το Περπάτημα ως την «ανώμαλη» κλάση και το Ξάπλωμα ως την «ομαλή».

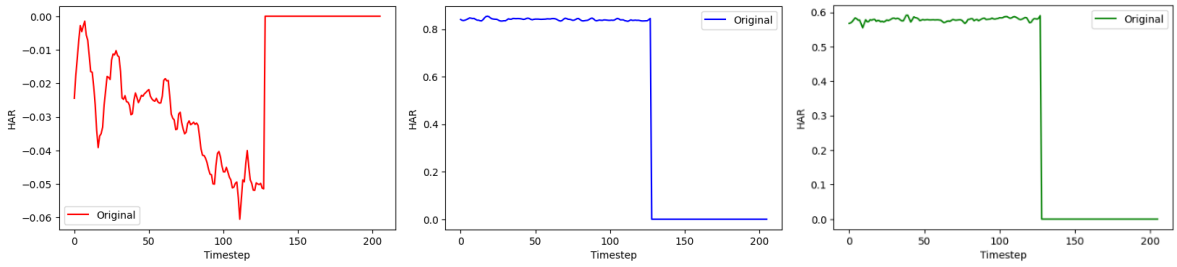
Dataset size	Length	Number of Classes	Number of Dimensions	Type
10.299	206	6	3	HAR

Πίνακας αριθμός: Τα χαρακτηριστικά του Dataset WalkingSittingStanding.



Εικόνα αριθμός: WalkingSittingStanding από την κλάση περπάτημα.

Περίπτωση απο έναν άνθρωπο που περπατάει και βρίσκεται στην «ανώμαλη» κατάσταση απο το σετ δεδομένων WalkingSittingStanding. Τρία διαγράμματα απο την κάθε διάσταση 0,1,2 αντίστοιχα εκτυπωμένα με την βιβλιοθήκη matplotlib.Pyplot.



Εικόνα αριθμός: WalkingSittingStanding από την κλάση ξάπλωμα.

Περίπτωση από έναν άνθρωπο που ξαπλώνει και βρίσκεται στην «ομαλή» κατάσταση από το σετ δεδομένων WalkingSittingStanding. Τρία διαγράμματα από την κάθε διάσταση 0,1,2 αντίστοιχα εκτυπωμένα με την βιβλιοθήκη matplotlib.Pyplot.

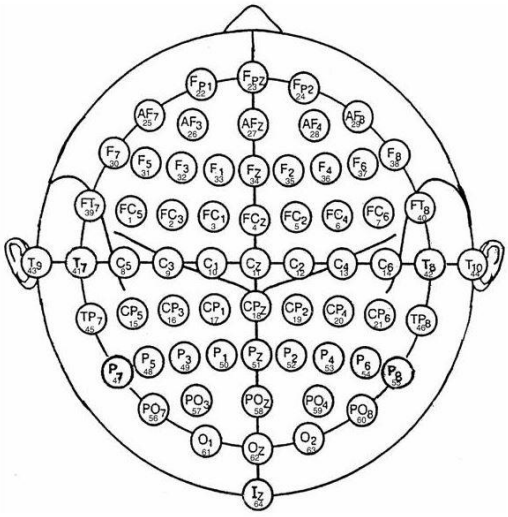
FingerMovemets Dataset

Αυτή η συλλογή δεδομένων πάρθηκε από την ιστοσελίδα timeseriesclassification.com (35) και δημιουργήθηκε από το ομάδα Intelligent Data Analysis του Fraunhofer-FIRST και το Τμήμα Νευρολογίας του Freie Universitat Berlin, υπό την εποπτεία του Gabriel Curio. Τα δεδομένα καταγράφηκαν από ένα υγιή άτομο κατά τη διάρκεια μιας συνεδρίας, ο οποίος καθόταν σε μια κανονική καρέκλα, με χαλαρά χέρια πάνω στο τραπέζι και τα δάχτυλα σε θέση πληκτρολόγησης. Το έργο ήταν να πατά τα πλήκτρα με τους δείκτες και τα μικρά δάχτυλα με οποιαδήποτε σειρά και χρονισμό ήθελε ο συμμετέχων. Υπήρχαν δύο κατηγορίες: 0 για μελλοντικές κινήσεις του αριστερού χεριού και 1 για το δεξί. Το πείραμα περιελάβανε 3 συνεδρίες των 6 λεπτών η κάθε μία και όλες διεξάχθηκαν την ίδια μέρα με μικρά διαλείμματα ανάμεσα. Η πληκτρολόγηση γινόταν με μέσο όρο 1 πλήκτρο ανά δευτερόλεπτο. Υπήρχαν 316 περιπτώσεις εκπαίδευσης και 100 δοκιμαστικές. Κάθε περίπτωση αποτελείται από μια καταγραφή 28 καναλιών EEG διάρκειας 500 ms, που τελειώνει 130 ms πριν από μια πληκτρολόγηση. (36)

Περιλαμβάνει 416 δεδομένα στο σύνολο με 28 διαστάσεις το κάθε ένα αντιπροσωπεύοντας ένα κανάλι EEG και σύνολο χρονικών σημείων 50.

Dataset size	**Length**	**Number of Classes**	**Number of Dimensions**	**Type**
416	50	2	28	ECG

Πίνακας αριθμός: Τα χαρακτηριστικά του Dataset FingerMovemets.



Εικόνα αριθμός: Διάγραμμα από έναν ανθρώπινο εγκέφαλο. (37)

Η παρουσίαση όλων των 28 διαστάσεων αυτού του σετ δεδομένων δεν θα ήταν πρακτική και χρήσιμη λόγο της πολυπλοκότητας του. Η εκτύπωση όλων αυτών των διαστάσεων θα δημιουργούσε μια τεράστια ποσότητα δεδομένων που θα ήταν δύσκολο να αναλυθεί ή να κατανοηθεί. Επίσης τα δεδομένα EEG συνήθως αναλύονται και επεξεργάζονται χρησιμοποιώντας ειδικούς αλγόριθμους και τεχνικές επεξεργασίας σημάτων. Η απλή εκτύπωση των αριθμητικών τιμών δεν παρέχει σημαντική επιστημονική ή διαγνωστική αξία. Έτσι είναι αρκετές οι μετρικές και το συγκεκριμένο σετ δεδομένων χρησιμοποιείται μόνο για την εκτίμηση των μεθόδων.

Επεξεργασία των δεδομένων

Σε αυτό το κεφάλαιο αναλύεται η επεξεργασία που υπέστη τα δεδομένα πριν δοθούν στα μοντέλα για εκπαίδευση και στις μεθόδους παραγωγής αντιπαραδειγμάτων.

Τα σημεία επεξεργασίας είναι τρία:

- *Up sampling* στη μειονοτική κλάση.
- *Κανονικοποίηση* των δεδομένων.
- *Padding* στην διάσταση των στιγμιότυπων.

Up sampling

Όταν τα σετ δεδομένων είναι ανισορροπημένα, δηλαδή μία κλάση είναι σημαντικά πιο κυρίαρχη από την άλλη, τα μοντέλα μηχανικής μάθησης τείνουν να μαθαίνουν καλύτερα την πλειοψηφική κλάση. Αυτό μπορεί να οδηγήσει σε ανακριβείς προβλέψεις και να εμποδίσει την απόδοση του μοντέλου. Το Upsampling είναι μία τεχνική που αντιμετωπίζει αυτό το πρόβλημα με τη δημιουργία συνθετικών δεδομένων για την μειονοτική κλάση ή με την αντιγραφή υπαρχόντων.

Εργαλεία για Upsampling

1. Τυχαίο Upsampling: Αυτή η μέθοδος περιλαμβάνει τυχαία επιλογή δεδομένων από τη μειονοτική κλάση, αναπαράγει αυτές τις περιπτώσεις και να τις προσθέσει στα δεδομένα έτσι ώστε με την αναπαραγωγή αυτή η μειονοτική κλάσει να έχει όσες περιπτώσεις έχει η πλειοψηφική. Είναι μια εύκολη στην υλοποίηση τεχνική και σίγουρη καθώς τα δεδομένα είδη υπάρχουν και το μόνο τασκ είναι να αναπαραχθούν. Ο κίνδυνος όμως αν αυτή η τεχνική δεν υλοποιηθεί σωστά είναι η υπερεκπαίδευση (overfiting). Αυτό σημαίνει ότι το μοντέλο το οποίο εκπαιδεύεται σε δεδομένα που έχουν καταστεί αυτήν την επεξεργασία προσαρμόζεται πολύ στενά στα χαρακτηριστικά των δεδομένων και αδυνατεί να γενικεύσει σε δεδομένα που δεν έχει γνωρίσει κατα την εκπαίδευση. Ο λόγος που συμβαίνει αυτό είναι γιατί άμα γίνει Upsamping σε πολλά δεδομένα θα υπάρχουν πολλά πανομοιότυπα με αποτέλεσμα το μοντέλο να βλέπει πολλές φορές κάποια μοτίβα στα πανομοιότυπα δεδομένα που μπορεί να μην είναι ενδεικτικά της κλάσης.

2. Τεχνική Συνθετικής Μειονοτικής Oversamping (SMOTE): Η μέθοδος αυτή είναι πολύ δημοφιλής όμως λίγο πιο περίπλοκη στην υλοποίηση. Χρησιμοποιεί την παρεμβολή χαρακτηριστικών για την δημιουργία συνθετικών δεδομένων ανάμεσα στις υπάρχουσες παρατηρήσεις της μειονοτικής κλάσης. Η μέθοδος αυτή όμως είναι πολύ αναγκαίο να εφαρμοστεί σωστά καθώς και αυτή με τη σειρά της ενέχει κινδύνους. Όπως και προηγούμενος ο ένας κίνδυνος είναι η υπερεκπαίδευση καθώς πάλι δημιουργούνται τα δεδομένα με βάση τα

χαρακτηριστικά των είδη υπαρχόντων. Εάν τα δεδομένα της μειονοτικής κλάσης δεν αντιπροσωπεύουν πιστά, υπάρχει ο κίνδυνος λανθασμένης ερμηνείας τους από την SMOTE και ως αποτέλεσμα παραμορφωμένα καινούργια δεδομένα. Τέλος δεν λειτουργεί καλά άμα τα δεδομένα έχουν πολλές διαστάσεις καθώς η ερμηνεία την κυρίαρχων χαρακτηριστικών τους είναι σαφώς πιο δύσκολη.

Στην εργασία αυτή προτιμάται η μέθοδος του τυχαίου Upsampling καθώς είναι εύκολη στην υλοποίηση και έχει θετικά αποτελέσματα. Δοκιμάστηκε να δημιουργηθούν συνθετικά δεδομένα, αλλά καθώς οι δύο μέθοδοι μεταξύ τους είχαν την ίδια αποδοτικότητα και είναι θεμιτό να γενικευτεί η διαδικασία για σετ δεδομένων με διάφερα μεγέθη, απορρίφθηκε.

Normalizing data

Το επόμενο βήμα που ακολουθείται στην επεξεργασία των δεδομένων είναι η κανονικοποίηση τους. Όλες οι τιμές των χρονοσειρών μπορεί να έχουν μεγάλες αποστάσεις μεταξύ τους κάτι που καθιστά την επεξεργασία τους αρκετά περίπλοκη. Είναι περισσότερο αποτελεσματικό, δεδομένα που παίρνουν τιμές από [-100,100] να αλλάζουν κλίμακα έτσι ώστε να κινούνται στο διάστημα [0,1]. Επιπροσθέτως, μπορεί να δημιουργηθεί και παραποίηση των αποτελεσμάτων, άμα λίγες τιμές είναι πολύ μεγαλύτερες ή μικρότερες από το μέσο όρο, επηρεάζονται τα βάρη από τις διαφορές στις τιμές αυτές αναντίστοιχα. Για αυτό το λόγο αλλάζουμε την κλίμακα έτσι ώστε όλες οι τιμές να βρίσκονται στο διάστημα [0,1] χωρίς όμως να χάνεται κάποια πληροφορία. Για να γίνει αυτό υλοποιείται ο παρακάτω τύπος σε κάθε δεδομένο ξεχωριστά.

$$X_{κανονικοποιημένο} = \frac{X - X_{ελάχιστο}}{X_{μέγιστο} - X_{ελάχιστο}}$$

Όπου X είναι το αρχικό δεδομένο, $X_{κανονικοποιημένο}$ είναι το καινούργιο δεδομένο που έχει τιμή από [0,1], $X_{ελάχιστο}$ είναι μικρότερο δεδομένο όλου του σετ και $X_{μέγιστο}$ είναι αντίστοιχα το μεγαλύτερο. Ως αποτέλεσμα της εφαρμογής αυτού του τύπου το μικρότερο δεδομένο έχει τιμή [0] και το μεγαλύτερο δεδομένο [1].

Padding

Τέλος εφαρμόζεται Padding στα δεδομένα. Η διαδικασία αυτή περιλαμβάνει την εφαρμογή μηδενικών τιμών στα άκρα των δεδομένων με σκοπό την καλύτερη λειτουργεία των μοντέλων καθώς επειδή αλλάζουν τις διαστάσεις στα δεδομένα αυτό μπορεί να προκαλέσει αδυναμίες.

Στους ταξινομητές και αυτοκωδικοποιητές τα δεδομένα περνάνε από διάφορα επίπεδα που μειώνουν τις διαστάσεις συνήθως στο μισό με τη μέθοδο δειγματολειψίας. Για να λειτουργήσει αυτή η διαδικασία οι διαστάσεις των δεδομένων εξαρχής είναι αναγκαίο να είναι διαιρετές του παράγοντα δειγματοληψίας. Στον αυτοκωδικοποιητή ακολουθείται μετά και η αντίστροφη διαδικασία επαναφοράς των διαστάσεων. Για να ανακτηθεί όμως η αρχική διάσταση, είναι σημαντικό να ξεκινούν από μία διάσταση που να επιτρέπει τη πολλαπλή διαίρεση της και στη συνέχεια την επαναφορά της.

Έτσι εξετάζονται οι διαστάσεις των δεδομένων, με σκοπό να διαιρούνται με το αριθμό τέσσερα, και προθέτονται μηδενικά στο τέλος της χρονοσειράς αν είναι απαραίτητο. Με αυτόν τον τρόπο διασφαλίζεται η ομοιομορφία των δεδομένων σε κάθε σημείο επεξεργασίας των διαστάσεων τους και αποφεύγεται η ανάγκη αναπροσαρμογής κάθε μοντέλου για διαφορετικές βάσεις δεδομένων με διαφορετικό μέγεθος ή διάσταση.

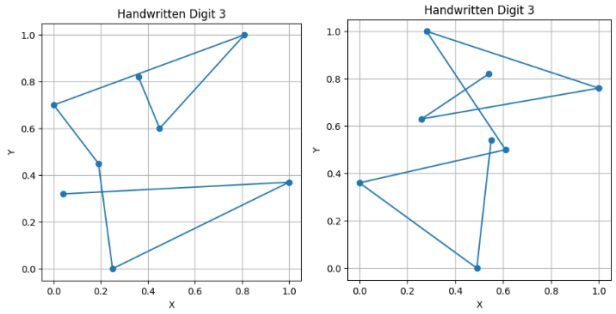
Να σημειωθεί σε αυτό το σημείο ότι θα ήταν αναγκαία η εφαρμογή μηδενικού Padding αν τα δεδομένα μεταξύ τους δεν είχαν ίδιο πλήθος στιγμών σε κάθε διάσταση, όμως στην εργασία αυτή δεν συναντιέται τέτοια περίπτωση.

Παρουσίαση Μεθόδων.

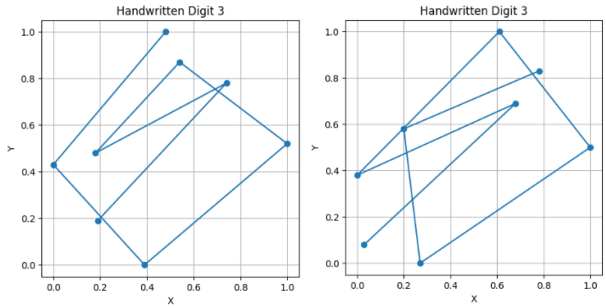
Σε αυτήν την ενότητα παρουσιάζονται τα αποτελέσματα των μεθόδων με την σειρά που υλοποιήθηκαν ξεκινώντας από την μέθοδο LatentCF++ των Zhendong, Rami Mochaourab, Παναγιώτης Παπαπέτρου, και έπειτα με τις αλλαγές που υλοποιήθηκαν με σκοπό τη βελτίωση, κάτι το οποίο κατέληξε στην μέθοδο LatentCF++_KDE.

Το Dataset που χρησιμοποιήθηκε για τις δοκιμές είναι το PenDigits σετ δεδομένων καθώς όπως έχει προαναφερθεί είναι κατάλληλο για την αποτίμηση όλων των αλλαγών αφού είναι εύκολα κατανοητά τα αποτελέσματα από τον ανθρώπο. Τα υπόλοιπα σετ δεδομένο χρησιμοποιούνται για να αποτιμηθούν τα αποτελέσματα τους στο επόμενο κεφάλαιο που παρουσιάζονται τα τελικά αποτελέσματα των δύο τελικών μεθόδων που πήραν μέρος στη σύγκριση και την εις βάθος ανάλυση των αποτελεσμάτων τους.

Πριν την παράθεση των μεθόδων θεωρείται σκόπιμο να αναφερθεί ότι καθώς το PenDigits σετ δεδομένων είναι ψηφιοποιημένοι αριθμοί γραμμένοι από ανθρώπους. Αυτό σημαίνει ότι υπάρχουν κάποια «κακής ποιότητας» δεδομένα τα οποία δεν φέρνουν στην όψη το ψηφείο της κλάσης στο οποίο ανήκουν. Συνεπώς καμία μέθοδος δεν μπορεί να έχει «τέλεια» αποτελέσματα και προσπαθείτε να παραχθεί η καλύτερη δυνατή μέθοδος με τα διαθέσιμα δεδομένα. Κάποια παραδείγματα από αυτά είναι:



Εικόνα αριθμός: «Κακής ποιότητας» δεδομένα από την κλάση [8].



Εικόνα αριθμός: «Κακής ποιότητας» δεδομένα από την κλάση [0].

Για όλα τα πειράματα έχουν χρησιμοποιηθεί τα συνελικτικά μοντέλα (ταξινομιτή και αυτοκωδικοποιητή) όπως ακριβώς έχουν περιγραφτεί στην ενότητα (**Γράψε ενότητα)**.

Ο ταξινομητής έχει απόδοση εγκυρότητας: 0.992

	Pred: [8]	**Pred: [0]**
True:[8]	*208*	*3*
True:[0]	*0*	*229*

Πίνακας αριθμός: Confusion Matrix του Συνελικτικού ταξινομιτή εκπαιδευμένο πάνω στις κλάσεις [0] και [8] του σετ δεδομένων Pen Digits.

Ο Αυτοκωδικοποιητής έχει απώλεια: 0.001 .

LatentCF++

Αυτή η υπο ενότητα ξεκινάει με την υλοποίηση της LatentCF++ όπως ακριβώς παρουσιάζεται στην διατριβή (2). Η μόνη διαφορά είναι τα σετ δεδομένων που επιλέχθηκαν. Στην διατριβή χρησιμοποιούνται μονοδιάστατες χρονοσειρές και εδώ πολυδιάστατες.

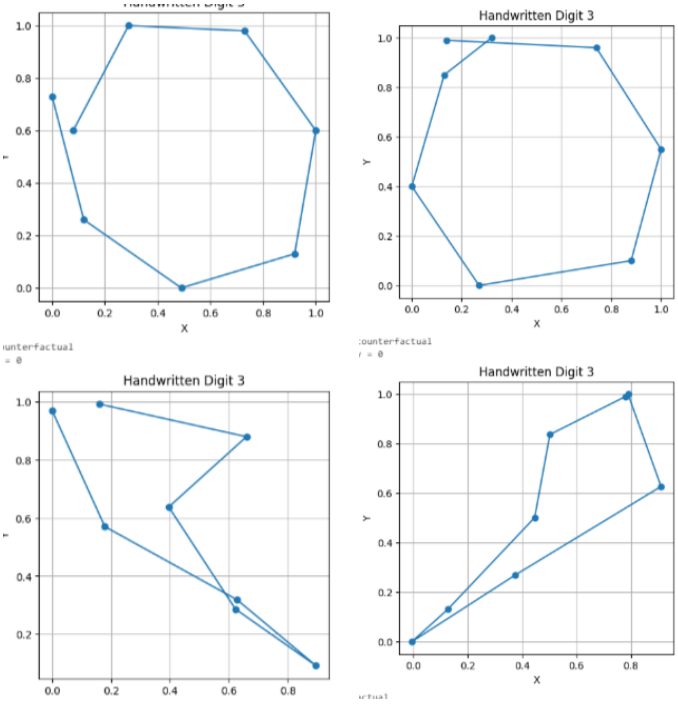
Το στήσιμο του πειράματος αυτού έχει ως εξής:

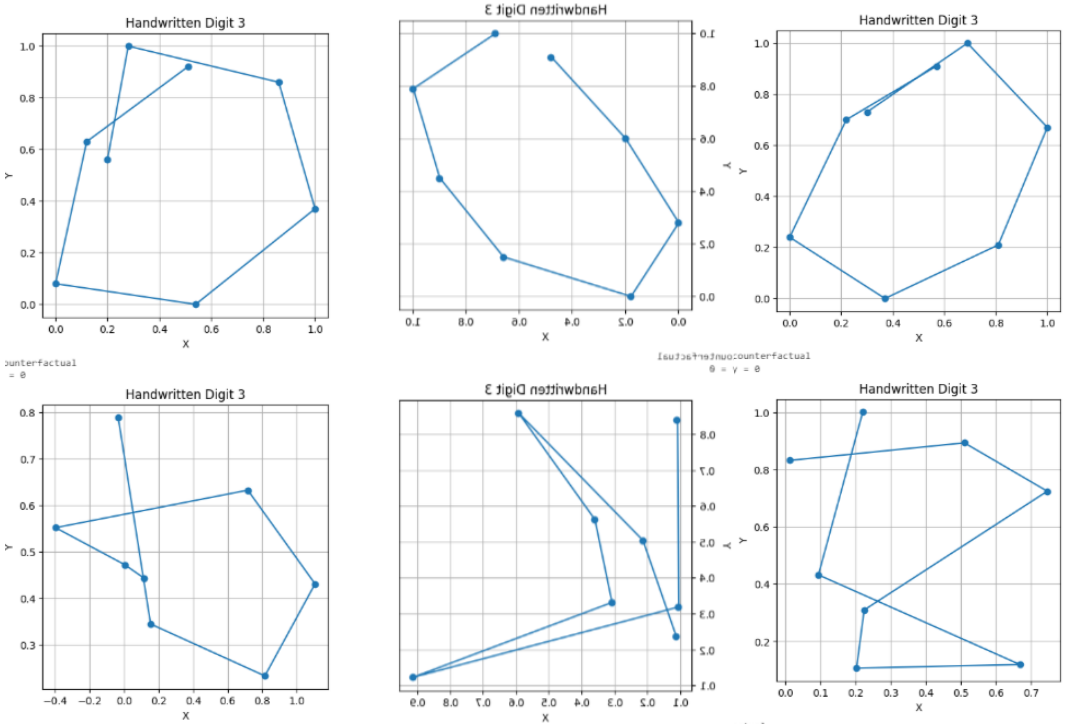
Η απώλεια υπολογίζεται απο τον συνδυασμό:

1. Της απόστασης ελάχιστων τετραγώνων της πιθανότητας που απέδωσε ο ταξινομητής στο δεδομένο που υπόκειται αλλαγές και του probability τ.
2. Της απόλυτης μέσης απόστασης μεταξύ του αρχικού δεδομένου X και του αντίστοιχου αντιπαραδείγματος του.

Το probability είναι ένα μέγεθος δοσμένο απο το χρήστη πριν γίνει η ευρεση αντιπαραδειγμάτων. Στα pen digits το probability έχει οριστεί στο 0.9 επειδή επιθυμείται τα αντιπαραδείγματα να αντιπροσωπεύουν την κλάση του με 90% επιτυχία για να είναι ξεκάθαρο και σίγουρο το αποτέλεσμα. Στα υπόλοιπα Dataset που διαχειρίζονται δεδομένα σημάτων εγκεφάλου όμως θα οριστεί στο 0.5 για να είναι στόχος να αποδίδει ο ταξινομητής με 50% σιγουριά έτσι ώστε να έχουν γίνει το λιγότερο δυνατόν αλλαγές.

Έτσι με την απώλεια να υπολογίζεται με αυτόν τον τρόπο δώθηκε στο μοντέλο πέντε δεδομένα (όλα μηδενικά) απο τα Pen Digits με εντολή να τα μετατρέψει σε οχτώ. Τα αποτελέσματα παρουσιάζονται παρακάτω. (Η εγκυρότητα σε αυτό το παράδειγμα είναι 0.9 δηλαδή 90% επιτυχία σύμφωνα με τον ταξινομιτή, όμως όταν γίνει συνολική αποτίμηση της μεθόδου με όλα τα δεδομένα τα στατιστικά δεν είναι τόσο ενθαρυντικά. Η πολύ καλή εγκυρότητα εδώ είναι τυχαία καθώς είναι πολύ μικρός ο αριθμός των δεδομένων προς αποτίμηση.





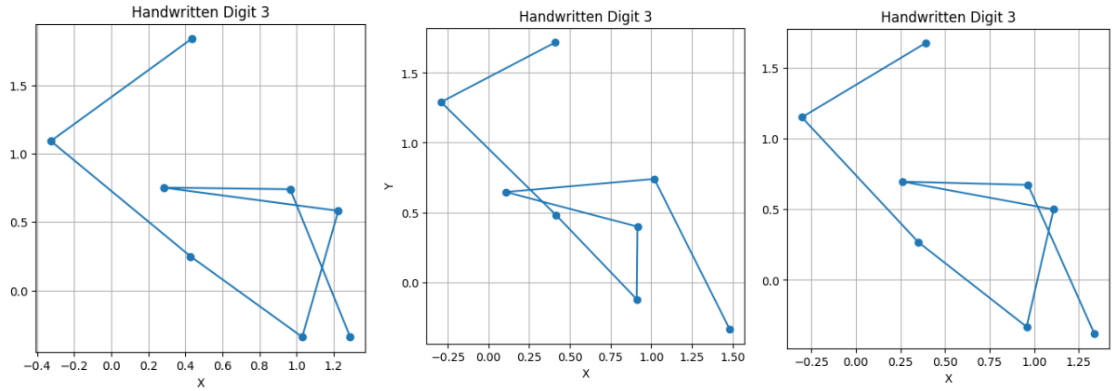
Εικόνα αριθμός: Πέντε παραδείγματα όπου πάνω παρουσιάζεται το μηδενικό που δόθηκε για βάση στη δημιουργία αντιπαραδείγματος και αντίστοιχα κάτω απο κάθε μυδενικό στοιχείο βρίσκεται το αντίστοιχο αντιπαράδειγμα του.

Αρχικά παρατηρείται ότι τα βήματα που γίνονται είναι τολμηρά. Στο πρώτο και το τελευταίο παράδειγμα από τα πέντε το αποτέλεσμα είναι πολύ ικανοποιητικό όμως στα υπόλοιπα τρία η κατεύθυνση δεν είναι τόσο σωστή και δεν υπάρχει καμία ποινή για να αναγνωριστεί και να ανακατευθυνθεί η μέθοδος. Έτσι τα αποτελέσματα που δόθηκαν για αντιπαραδείγματα δεν είναι αντιπρωροπευτικά της κλάσης τους. Ο ταξινομιτής όμως δίνει πιθανότητα 90% σε αυτά τα να είναι όχτώ, ένα αποτέλεσμα το οποίο είναι αναμενόμενο με βάση τον τρόπο λειτουργίας του ταξινομιτη. Ο εν λόγω ταξινομιτής έχει δύο κλάσεις απο τις οποίες καλείται να διαλέξει, το [0,8]. Τα παραδείγματα αυτά αν και όχι αντιπρωσοπευτικά της κλάσης τους όπως παρατηρείται με το ανθρώπινο μάτι, παρουσιάζει περισσότερα μοτίβα του οχτώ παρά του μηδέν.

Εκπαιδευμένα Μοντέλα σε Όλες τις Κλάσεις.

Έτσι δοκιμάστηκε η κίνηση της LatentCF++ με τη χρήση ενός αυτοκωδικοποιητή και ενός ταξινομιτή με τα ακριβώς ίδια χαρακτηριστικά που όμως έχουν εκπαιδευτεί και στις 10 κλάσεις των δεδομένων και όχι μόνο στο μηδέν και το οχτώ. Ο σκοπός αυτής την ενέργειας ήταν να παρατηρηθεί η αντίδραση της μεθόδου παραγωγής αντιπαραδειγμάτων πλησιάζοντας κάποιο άλλο νούμερο πέραν του μηδέν και του οχτώ με την ελπιδα ότι θα έχει την ικανότητα να δώσει καινούργια κατεύθυνση άμα το αντιπαράδειγμα αρχίζε και πλησιάζει κάποια άλλη κλάση του σετ δεδομένων και όχι το οχτώ που είναι το «ομαλό».

Τα αποτελέσματα όμως όπως φαίνεται και παρακάτω δεν είναι θετικά προσκείμενα.



Εικόνα αριθμός: Αναπαραστάσεις τριών αντιπαραδειγμάτων που αντιπροσωπευουν οχτώ.

Όπως γίνεται κατανοητό απο την εικόνα το σχήμα που αναγνωρίζεται ώς οχτώ έχει αλοιωθεί με παρομοιότυπο τρόπο σε όλα τα αντιπαραδείγματα. Όταν ο αυτοκωδικοποιητής και ο ταξινομητής εκπαιδεύονται μόνο σε δύο ψηφία (0 και 8), μαθαίνουν και προσαρμόζονται στα συγκεκριμένα χαρακτηριστικά και ιδιότητες αυτών των ψηφίων. Ο συμπιεσμένος χώρος γίνεται ειδικευμένος στην αναπαράσταση παραλλαγών των 0 και 8. Ως εκ τούτου, όταν προσπαθείτε να δημιουργηθεί ένα 8 από ένα μηδέν, ο αυτοκωδικοποιητής ήδη κατανοεί καλά τις διαφορές μεταξύ αυτών των δύο ψηφίων και μπορεί να κάνει ουσιαστικές τροποποιήσεις στην είσοδο (μηδέν) για να τη μετατρέψει σε ένα 8.

Αντίθετα, όταν ο αυτοκωδικοποιητής και ο ταξινομητής εκπαιδεύονται σε όλα τα ψηφία (0 έως 9), ο συμπιεσμένος χώρος γίνεται πιο γενικευμένος για να φιλοξενήσει την ευρύτερη γκάμα χαρακτηριστικών που παρουσιάζονται σε όλα τα

δέκα ψηφία. Αυτή η γενίκευση μπορεί να οδηγήσει σε λιγότερο ακριβή κατανόηση των ειδικών διαφορών μεταξύ κάθε δύο ψηφίων (όπως το 0 και το 8). Επομένως, όταν προσπαθείτε να δημιουργηθεί ένα 8 από ένα μηδέν σε αυτό το σενάριο, ο αυτοκωδικοποιητής μπορεί να δυσκολευτεί να κάνει συγκεκριμένες, στοχευμένες αλλαγές. Αντίθετα παρουσιάζει κομμάτια των μοτίβων που παρουσιάζονται σε άλλα ψηφείο οδηγώντας τις εικόνας τα αλοιώνονται με ακατανόητο τρόπο.

Χρήση Δεδομένου απο την Ομαλή Κλάση ως Κατευθυνση.

Εναπανπροσπαθείται η χρήση ταξινομιτή και αυτοκωδικοποιητή εκπαιδευμένα μόνο στίς δύο κλάσεις που είναι αναγκαίο, τις [0] και [8]. Στην προσπάθεια όμως να δωθεί μια πιο σωστή κατεύθυνση δοκιμάστηκε να δωθεί στη μέθοδο ένα δεδομένο απο την «ομαλή» κλάση ώς παράδειγμα προς μίμηση.

Η χρήση δεδομένου απο την «ομαλή κλάση» ώς παράδειγμα προς μίμηση στη μέθοδο λειτουργεί ώς εξής:

Γίνεται μια προσπάθεια να βελτιωθεί η ακρίβεια του μοντέλου χρησιμοποιώντας δεδομένα από την "ομαλή" κλάση. Αυτό σημαίνει ότι για κάθε νέο δεδομένο που εξετάζεται, το σύστημα αναζητά ένα δείγμα από την ομαλή κλάση που είναι το πλησιέστερο δυνατό σε αυτό. Η ιδέα είναι να επιλέγεται κάθε φορά ένα διαφορετικό δείγμα από την ομαλή κλάση, ώστε τα αντιπαραδείγματα που δημιουργούνται να μην είναι όμοια μεταξύ τους.

Με τη χρήση της LatentCF++ και των συνελικτικών επιπέδων της, ο ταξινομητής και ο αυτοκωδικοποιητής εντοπίζουν το πλησιέστερο δείγμα από την "ομαλή" κλάση, προκειμένου το νέο δεδομένο να αποκτήσει όσο το δυνατόν πιο ομαλές χαρακτηριστικές ιδιότητες, διατηρώντας ταυτόχρονα την αρχική του ταυτότητα. Αυτή η μέθοδος αποσκοπεί στη βελτίωση της ακρίβειας του συστήματος, καθώς και στη δημιουργία πιο ποικίλων και αντιπροσωπευτικών αντιπαραδειγμάτων.

Καταλληκτικά η μέθοδος κάνει τους εξής υπολογισμούς:

Αρχικά δίνεται κάποιο δεδομένο στη μέθοδο Χ με σκοπό να παραχθεί αντιπαράδειγμα Χ'. Έπειτα γίνεται εύρεση του κοντινότερου «ομαλού» δεδομένου στο αρχικό «ανώμαλο» δεδομένο Χ. Για την επίτευξη αυτού γίνεται χρήση λούπας που κοιτάει όλα τα δεδομένα που ανοίκουν στην «ομαλή» κλάση και υπολογίζεται η μέση απόλυτη απόσταση MAE μεταξύ των δύο δεδομένων σώζοντας το δεδομένο με την μικρότερη MAE τιμή πολλαπλασιασμένο κατα στήλη με τα βάρη των χρονικών βημάτων όπως ακριβώς παρουσιάζεται παρακάτω.

$$MAE_{(X_{orig},\ X_{sample\_abnormal})} = |Xorig - Xsample\_abnormal|$$

$$loss = MAE_{(X_{orig}, X_{sample\_abnormal})} * step\_weights$$

$$BestX_{abnorsample_{abnormalmal}} =$$

$$X_{sample\_abnormal}[MAE_{(X_{orig}, X_{abnormal})} == minMAE_{(X_{orig}, X_{sample\_abnormal})}] \ (3)$$

Όπου X_{orig} είναι το αρχικό δεδομένο που προκειται να μεταλαχθεί σε αντιπαράδειγμα, $X_{sample\_abnormal}$ είναι το δεδομένο της ομαλής κλάσεις που τίθεται προς σύγκριση, $step\_weights$ είναι τα βάρη που έχουν εντιστοιχηθεί σε κάθε χρονικό βήμα και η εξίσωση (3) είναι η εκφραση «Το ομαλό δεδομένο που έχει τη MAE ίσο με τη μικρότερη MAE τιμή».

Αφού έχει βρεθεί το καλύτερο «ομαλό» δεδομένο $X_{sample\_abnormal}$ το αρχικό δεδομένο Χ υπολογίζεται η απώλεια :

$$\mathrm{MSE} = (\mathrm{y}_{pred} - \tau)^2$$

$$loss += weight_{mse} * MSE$$

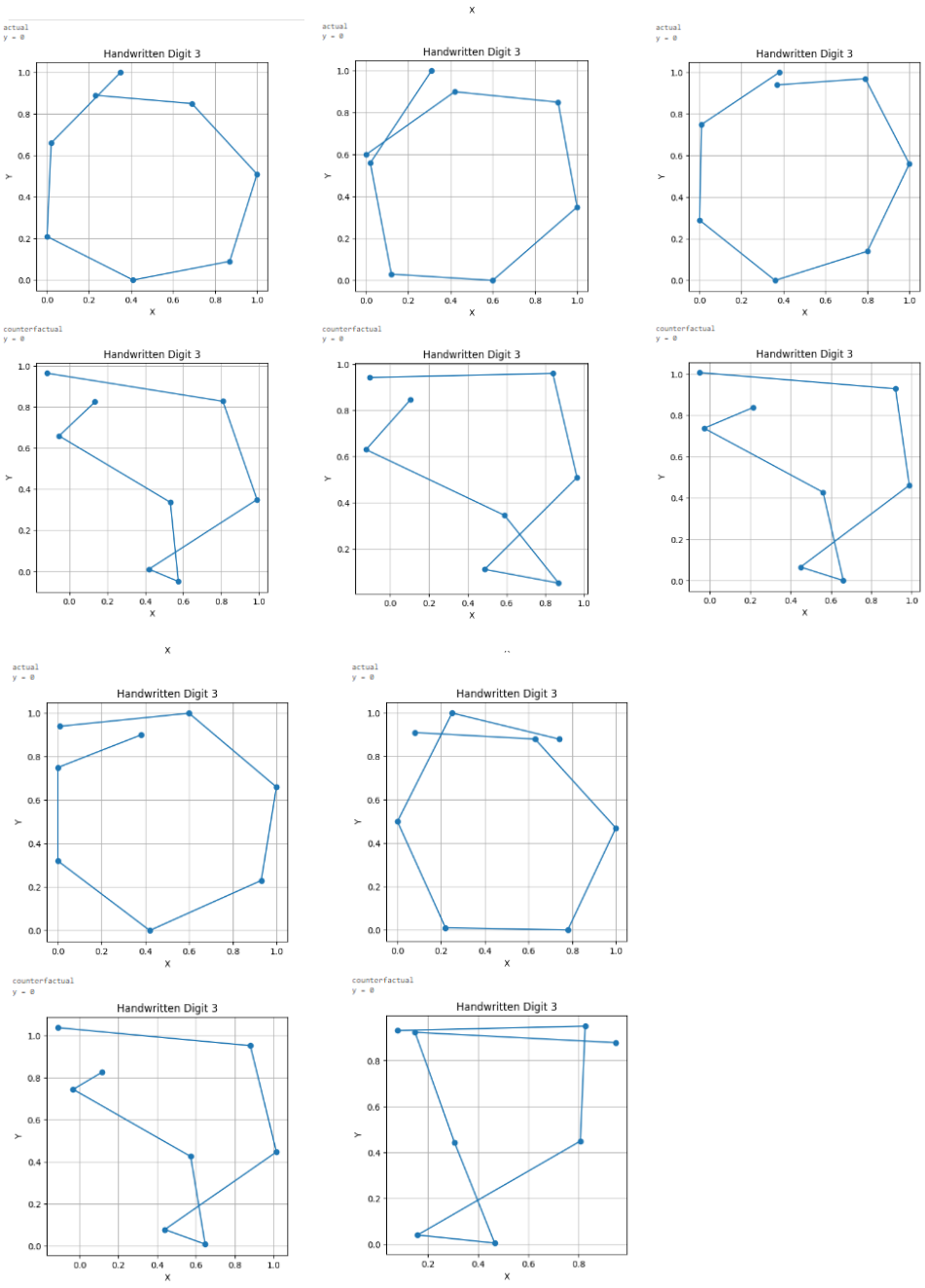
Όπως ακριβώς και σε όλες τις υπόλοιπες μεθόδους και:

$$MAE_{(X_{count},\ X_{sample\_abnormal})} = |Xcount - Xsample\_abnormal|$$

$$loss \mathrel{+}= weight_{mae} * MAE_{(X_{count}, X_{sample\_abnormal})} * step\_weights$$

Όπου $weight_{mse}$ βρέθηκε καταλληλότερα στο πείραμα με τα συγκεκριμένα δεδομένα να είναι ίσο με 0.5 και $weight_{mae}$ αντίστοιχα με 0.5.

Με την ανελυμένη πλεον μέθοδο τα αποτελέσματα παρουσιάζονται παρακάτω.



Εικόνα αριθμός: Στην εικόνα φαίνονται πέντε τυχαία αποτελέσματα απο την μέθοδο με την χρήση δεδομένου ως Κατευθυνση. Στην πάνω πλευρα παρουσιάζονται τα μηδενικά απο όπου εκκίνησε το κάθε αντιπαράδειγμα και αντίστοιχα απο κάτω βρίσεται το αντιπαράδειγμα που δημιουργήθηκε.

Όπως γίνεται κατανοητό απο την είκονα αριθμός, είναι εμφανέστατα πιο βελτιωμένα τα αποτελέσματα σε σχέση με προιγουμένως. Τα τέσσερα πρώτα αποτελέσματα όμως είναι πολύ πανομοιότυπα μεταξύ τους κάτι το οποίο συνιστά ότι το πιο κοντινό δεδομένο προς μίμηση που επιλέχθηκε απο το μοντέλο είναι ενδεχωμένος το ίδιο κάθε φορα ή αν όχι είναι παρόμοιο. Αυτό είναι λογικό καθώς το δεδομένο της κλάσης [8] που απέχει μικρότερη απόσταση απο κάποιο δεδομένο [0] θα είναι αυτό που θα απέχει μικρότερη απόσταση και απο τα υπόλοιπα. Είναι η μικρότερη απόσταση των δύο κλάσων συνολικά, η διαφορετικά είναι το [8] που μοιάζει περισσότερο με [0] είναι ένα συγκεκριμένο για τα περισσότερα δεδομένα.

Ετσι βγαίνει το αποτέλεσμα οτι η μέθοδος αυτή δεν μπορεί να θεωρηθεί ικανοποιητική καθώς δεν έχει τη δυνατότητα να γενικεύσει με το σωστό τρόπο. Περαιτέρω είναι σημαντικό να αναφερθεί οτι τα δετ δεδομένων που απεικονίζουν ιαρτική πληροφορία απο τη μέθοδο αυτή δεν θα δημιουργήσουν σωστό αντιπαράδειγμα καθώς θα ακολουθεί χαρακτηριστικά άλλων δεδομένων αλλων ασθενών κάτι το οποίο καταλλήγει σε λάθος συμπεράσματα.

Λαμβάνοντας αυτά υπόψη παρουσιάζουμε και αυτή τη μέθοδο αλλα με προσοχή καθώς δεν ακολουθεί σωστή λογική για μία μεγάλη γκάμα σετ δεδομένων. Η μέθοδος αυτή είναι ικανοποιητική μονο για περιπτώσεις που ακολουθούν ένα σετ δεδομένων όπως το Pen Digits που τα δεδομένα δεν μας ενδιαφέρει αν θα ακολουθούν την κατανομή και τις ιδιότητες ενός άλλου ξεχωριστής προέλευσης δεδομένου.

LatentCF++_KDE

Καθώς καμία απο τις προιγούμενες μεθόδους δεν έχει ικανοποιητικά αποτελέσματα (εκτώς απο την μέθοδο με Χρήση Δεδομένου απο την Ομαλή Κλάση ως Κατευθυνση η οποία όμως έχει αδυναμία γενίκευσης) επαναπροσπαθείται η

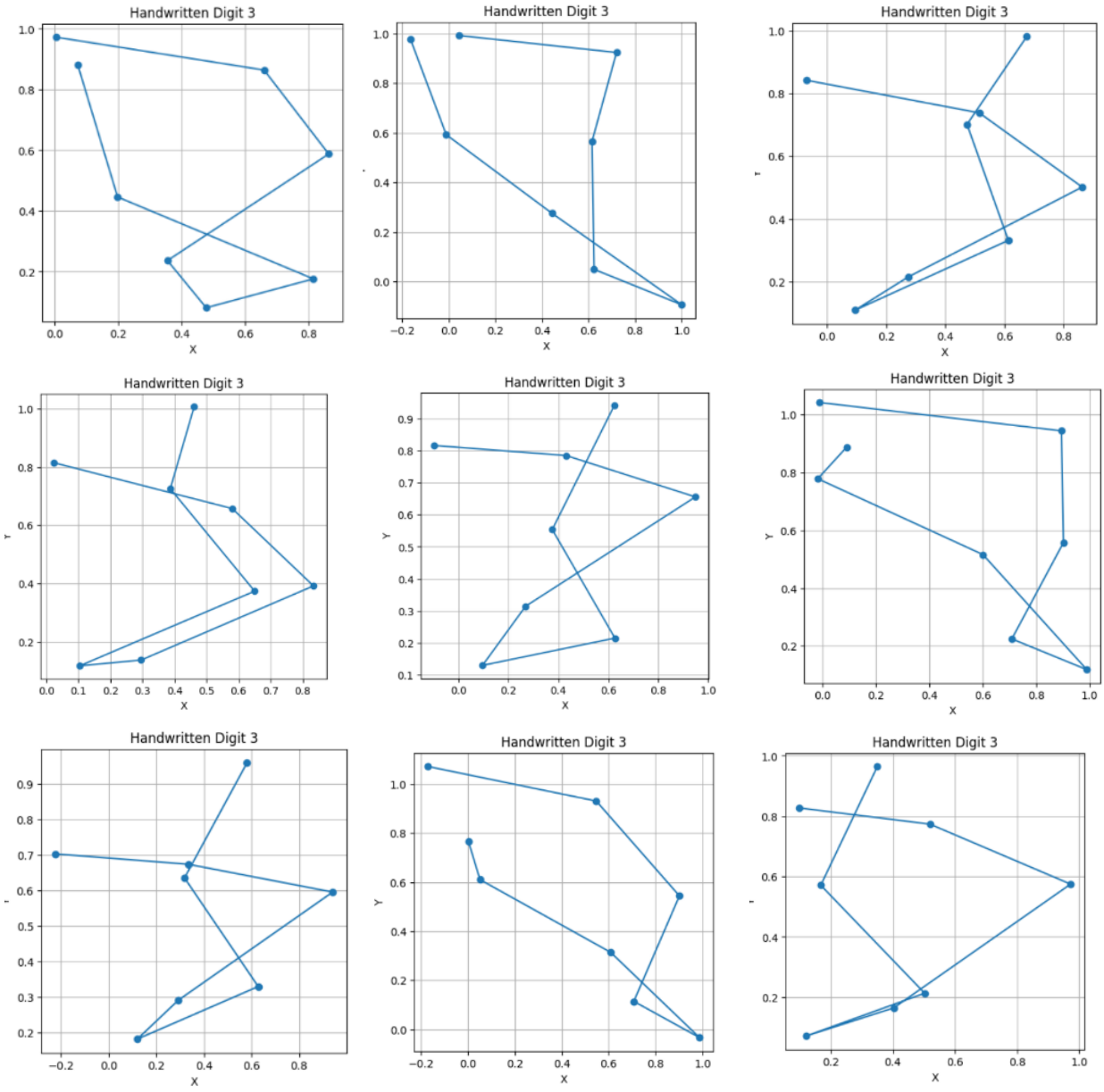
πρώτη μέθοδος LatentCF++ με την εισαγωγή της σύγκρισης κατανομών για να δωθεί η παραπάνω καρευθυντήρια στον αυτοκωδικοποιητή.

Η μέθοδος έχει αναλυθεί στην αρχική του Πρακτικού μέρους της εργασίας αυτής καθώς είναι η πιο πετυχημένη μέθοδος και είναι η τελική που παρουσιάζεται. Στην παράγραφο αυτή κρίνεται σκόπιμο να αναφερθεί ξανά το βασικότερο στοιχείο της μεθόδου, που είναι η συνάρτηση απώλειας.

Συνάρτηση απώλειας:

1. Η απόσταση ελαχίστων τετραγώνων μεταξύ του τ (decition boundary) και του pred, την πιθανότητα που απέδωσε στο αντιπαράδειγμα ο ταξινομιτής.
2. Η μέση απόλυτη απόσταση του αρχικού δεδομένου X και του αντιπαραδείγματος X’ πολλαπλασιασμένο με τα βάρη των χρονικών βημάτων.
3. Τη διαφορά της μέσης λογαριθμικής KDE κατανομής των δεδομένων την «ομαλής» κλάσης με την λογαριθμική κατανομή KDE του αντιπαραδείγματος X’.

Ετσι παρουσιάζονται τα αποτελέσματα του σετ δεδομένων PenDigits στην μέθοδο αυτή.



Εικόνα αριθμός: Στην εικόνα αυτή παρατίθονται 10 αντιπαραδείγματα απο τυχαία δείγματα για να γίνει κατανοητή η αποτελεσματικότητα της μεθόδου. Για την αναπαραγωγή αυτήν των οχταριών έχει χρησιμοποιηθεί η απώλεια των ελάχιστων τετραγώνων κατα 75% και κατα 25% η απώλεια της διαφοράς των κατανομών.

Όπως έχει προαναφερθει δεν είναι λογική η απαίτηση τα αντιπαραδείγματα να είναι τέλειοι ψηφιακοί αριθμοί καθώς οι αριθμοί που δίνονται για εκπαίδευση στον αυτοκωδικοποιητή όντας χειρόγραφοι αριθμοί αποκλείνουν απο την τυπική μορφή του αριθμού καθώς παρεμβάλεται ο ανθρώπινος παράγοντας.

Συνοψίζοντας, η μέθοδος LatentCF++ με την ενσωμάτωση της συγκρίσεως κατανομών στη συνάρτηση απώλειας είναι ικανή να επιδείξει μεγάλη αποδοτικότητα στο συγκεκριμένο σετ δεδομένων. Η εφαρμογή αυτής της μεθόδου, αν και με μή τέλειους ψηφειακούς αριθμούς, υποδεικνύει οτι είναι δυνατή η δημιουργία αντιπαραδειγμάτων που να αποτυπώνουν ικανοποιητικά το ζητούμενο, διατηρώντας παράλληλα την γενική ουσία των αρχικών δεδομένων μέσω της διατήρησης της κατανομής. Αυτό αποτελεί ένδειξη ότι η μέθοδος έχει την ικανότητα να παράγει αξιόπιστα αντιπαραδείγματα, ακόμα και όταν αντιμετωπίζεται με δεδομένα που είναι επηρεασμένα από την ανθρώπινη παρέμβαση και την ατέλεια. Επομένως, η LatentCF++_KDE αποτελεί μια προηγμένη και αποδοτική προσέγγιση για την επεξεργασία και ανάλυση δεδομένων, προσφέροντας σημαντικά οφέλη στην εκτίμηση της ποιότητας και της αξιοπιστίας των προβλεπόμενων αποτελεσμάτων.

Πριν όμως την τελική εκτίμηση της καλύτερης μεθόδου είναι σημαντικό να γίνει μια συνολική κριτική με μετρικές υπολογισμένες πάνω σε όλα τα δεδομένα γιατί τα αποτελέσματα που έχουν παρουσιαστεί μεχρι στιγμής είναι παρμένα απο ένα πολύ μικρό δείγμα δεδομένων.

Παρουσίαση Αποτελεσμάτων.

Στην συγκερκιμένη ενότητα παρουσιάζονται τα τελικά αποτελέσματα των σετ δεδομένων με την μέθοδο της LatentCF++ και την LatentCF++_KDE, όπως έχει περιγραφεί στην ανάλυση δεδομένων. Η κάθε μέθοδος έχει εφαρμοστεί με συνελικτικά και LSTM μοντέλα, παρατίθονται αρχικά η αποτελεσματικότητα τους με κάθε σετ δεδομένων.

Απόδοση Μοντέλων

Για του ταξινομιτές παρουσιάζεται η μετρική απόδωσης και για τους αυτοκωδικοποιητές η μετική απώλειας. Έτσι ένας επιτυχημένος ταξινομιτής έχει απόδωση 1.0 και ένας επιτυχημένο αυτοκωδικοποιητής έχει απώλεια 0.0.

	Ταξινομιτές		Αυτοκωδικοποιητές	
	Συνελικτικός	**LSTM**	**Συνελικτικός**	**LSTM**
PenDigits	0.9929	0.9976	0.0012	0.0035
WalkingSittingStanding	1.0	1.0	0.0004	0.0251
FingerMovements	0.4404	0.4761	0.0033	0.0092

Πίνακας αριθμός: Αποδόσεις των μοντέλων για κάθε σετ δεδομένων.

Τα σετ δεδομένων PenDigits και WalkingSittingStanding έχουν πολύ καλή απόδοση και με τα δύο ειδών μοντέλα. Το FingerMovements απο την άλλη έχει ικανοποιητική απόδοση στους αυτοκωδικοποιητές αλλα όχι στους ταξινομιτές. Αυτό μπορεί να συμβαίνει για διαφόρους λόγους.

Κάποιοι ενδεικτικά θα μπορούσε να είναι:

1. Ο μικρός αριθμός δεδομένων (200 σε κάθε κλαση) σε αντίθεση με τα άλλα σετ δεδομένων (που είχαν πάνω απο 1.000).
2. Πολλές διαστάσεις: Οι πολλές διαστάσεις καθιστούν το σετ δεδομένων περίπλοκο και σε συνδιασμό με τα λίγα δεδομένα δεν είναι εύκολο ο ταξινομιτής να παρατηρήσει τα κυρίαρχα διαχωριστικά χαρακτηριστικά των κλάσεων.
3. Το dataset να είναι θορυβώδες. Οι αυτοκωδικοποιητές λόγος της ικανότητας τους να συμπιέζουν τα δεδομένα με τέτοιο τρόπο όπου μόνο η σημαντική πληροφορία απομένει στη συμπιεσμένη μορφή, είναι πολύ χρήσιμοι όταν παρουσιάζεται θόρυβος στα δεδομένα. Άμα ενα σετ δεδομένων περιέχει υψηλά ποσοστά θορύβου, κατα την συμπίεση αυτά ακριβώς τα ποσοστά είναι κατασκευασμένος να διώξει καθως δεν είναι χρήσιμο κομμάτι της πληροφορίας. Οι ταξινομιτές, απο την άλλη, δεν ξεχωρίζουν τα δεδομένα κάτω απο το θόρυβο με αποτέλεσμα να απαιτείται η εμπολή μεθόδων καθαρισμού των δεδομένων ώστε να μπορούν να διακρίνουν τις κλάσεις.

Μετρικές απόδοσης LatentCF++ και LatentCF++_KDE

Παραθέτονται οι πίνακες των αποδόσεων για τις δυο μεθόλους και κάθε σετ δεδομένων και μοντέλων αντίστοιχα.

Validity

Με bold είναι εκείνα που είχαν το καλύτερο αποτέλεσμα και υπογραμμισμένα εκείνο με τα χειρότερα.

		PenDigits	WalkingSittingStanding	FingerMovements	Μέσος όρος
LatentCF_KDE	**1dCNN**	**1.0**	**1.0**	**1.0**	**1.0**
LatentCF_KDE	**LSTM**	0.95	**1.0**	0.3	0.75
LatentCF++	**1dCNN**	0.4851	**1.0**	**1.0**	0.8283
LatentCF++	**LSTM**	0.1	**1.0**	0.1634	0.4211

Πίνακας αριθμός: Αποτελέσματα Validity

Η πιο αποτελεσματική μέθοδος σε αυτήν τη μετρική είναι το LatentCF++_KDE με την χρήση συνελικτικών μοντέλων καθώς έχει 100% επιτυχία σε όλα τα σετ δεδομένων. Η δευτερη καλύτερη μέθοδος με βάση το μέσο όρο φαίνεται να είναι η LatentCF++ με συνελικτικά μοντέλα, όμως αυτό συμβαίνει λόγο της επιτυχίας των συνελικτικών μοντέλων. Τα LSTM δεν εχουν καλή απόδοση με καμία μέθοδο στα FingerMovements δεδομένα καθώς όπως έχει προαναφερθεί είναι σύνθετο σετ δεδομένων και καθιστάται αναγκαία η εκτενής προσπάθεια ευρεσης μοντέλων που θα είναι κατάλληλα για τα συγκεκριμένα περίπλοκα χαρακτηριστικά της μεθόδου.

Τα LSTM μοντέλα στα δεδομένα PenDigits με τη χρήση της LatentCF++_KDE μέθοδο έχουν καλύτερη απόδοση σε σύγκριση με τη χρήση της LatentCF++ και συνολικά μπορεί να βγεί το συμπέρασμα οτι η LatentCF++_KDE είναι η πιο αποτελεσματική μέθοδος.

Proximity (X-X’)

Με Bold τονίζονται τα δεδομένα με το καλύτερο Proximity. Υπογραμισμένα είναι εκείνα που λόγο της αποτυχίας τους στο Validity δεν λαμβάνονται υπόψη, καθώς είναι αναμενόμενο να απέχουν μικρή απόσταση απο τα αρχικά δεδομένα τα αντιπαραδείγματα εκείνα που δεν άλλαξαν κλάση.

(Ο Μέσος Όρος δεν περιλαμβάνει τα υπογραμμισμένα.)

		PenDigits	**WalkingSittingStanding**	**FingerMovements**	**Μέσος όρος**
LatentCF_KDE	**1dCNN**	0.5442	2.4172	**0.4791**	1.147
LatentCF_KDE	**LSTM**	0.6596	2.2362	0.6441	1.4479
LatentCF++	**1dCNN**	0.2458	2.2574	**0.4412**	1.349
LatentCF++	**LSTM**	0.1132	2.1505	-0.00422	2.1505

Πίνακας αριθμός: Αποτελέσματα Proximity

Χωρίς να λαμβάνονται υπόψη τα υπογραμμισμένα αποτελέσματα, τα μοντέλα που κατάφεραν να κάνουν τις μικρότερες δυνατές αλλαγές είναι τα Συνελικτικά μοντέλα στο σετ δεδομένων FingerMovements. Στα δεδομένα PenDigits με τη χρήση της μεθόδους LatentCF++_KDE τα αποτελέσματα είναι πολύ όμοια με αυτά των FingerMovements και έτσι συμπεραίνεται ότι είναι και αυτά γίνονται δεκτά ώς καλα αποτελέσματα Proximity. Στο σετ δεδομένων WalkingSittingStanding η απόσταση είναι πολύ μεγάλη για όλες τις μεθόδους με ελάχιστα μικρότερη στην LatentCF++ όμως με πολύ μικρή διαφορά για να αξίζει ανάλυση.

Margin Difference absolute

		PenDigits	**WalkingSittingStanding**	**FingerMovements**	**Μέσος όρος**

LatentCF_KDE	**1dCNN**	**0.0239**	0.4055	**0.0642**	0.1645
LatentCF_KDE	**LSTM**	**0.1008**	0.404	0.0024	0.1691
LatentCF++	**1dCNN**	0.4786	0.399	**0.0005**	0.2927
LatentCF++	**LSTM**	0.8919	0.4009	0.0043	0.4324

Πίνακας αριθμός: Η μέση απόλυτη απόσταση του X’ και τ.

Margin Difference without absolute

		PenDigits	**WalkingSittingStanding**	**FingerMovements**	**Μέσος όρος**
LatentCF_KDE	**1dCNN**	0.0007	0.4055	0.0064	0.1375
LatentCF_KDE	**LSTM**	0.0443	0.404	0	0.1494
LatentCF++	**1dCNN**	0.4183	0.399	0.0005	0.2726
LatentCF++	**LSTM**	0.8019	0.4009	-0.0042	0.3995

Πίνακας αριθμός: Η μέση απόλυτη απόσταση του τ και X’ (για κατεύθυνση).

Στον πίνακα (αριθμό) παρουσιάζονται οι αποστάσεις του X’ (αντιπαράδειγμα) και τ (decision boundary). Χρησιμοποιείται το απόλυτο για να μην υπάρξει αλληλοδιαγραφή αν κάποιο έχει αρνητικό πρόσημο, με ένα αντίστοιχο θετικό. Στον πίνακα (αριθμό) όμως παρουσιάζονται και χωρίς απόλυτο έτσι ώστε να φανεί η κατευθυνση του αντιπαραδείγματος. Αρνητικά είναι εκείνα που δεν έγιναν αντιπαραδείγματα οπότε δεν είναι αποδεκτά αποτελέσματα, αλλα υπάρχουν και εκείνα που έχουν μικρότερη τιμή στην μετρική με απόλυτο απο ότι με χωρίς. Αυτό συμβαίνει γιατι υπάρχουν αρνητικά στοιχεία (δεδομένα που δεν άλλαξαν κλάση) απλά όχι τόσα ώστε να είναι απορρυφθεί το σετ δεδομένων.

Η καλύτερη απόσταση απο το τ παρουσιάζεται στο σετ δεδομένων FingerMovements με συνελικτικά μοντέλα και στο σετ δεδομένων PenDigits με τη χρήση του LatentCF_KDE. Όλα τα αποτελέσματα όμως είναι αποδεκτά καθώς και στο WalkingSittingStanding οι αποστάσεις είναι μικρές και δεν υπάρχει σημαντική διαφορά ανάμεσα στις δύο μεθόδους για να μπορεί να βγεί καποιο συμπέρασμα. Τα σετ δεδομένων WalkingSittingStanding και FingerMovements έχουν ίδια απόσταση και με απόλυτο και χωρίς. Αυτό είναι λογικο καθώς αν υπήρχαν δεδομένα κάτω απο το 0 (που θα κατέληγε σε μικρότερη απόλυτη τιμή) το Validity τους δεν θα ήταν 1.0. Το σετ δεδομένων PenDigits έχει μικρότερες τιμές στη μετρική με απόλυτο, ενώ έχουν Validity 1.0, όμως και αυτο εξηγείται καθώς το τ ορίζεται στο 0.9 για την «καθαρότητα» των αποτελεσμάτων σε αντίθεση με τα άλλα σετ δεδομένων που ορίζεται απο το 0.5. Έτσι ενώ υπάρχουν αρνητικές τιμές που μικραίνουν την μετρική με απόλυτο αυτο δεν καταλήγει σε αντιπαραδείγματα που δεν άλλαξαν κλάση.

KDE ανώμαλων δεδομένων

		PenDigits	WalkingSittingStanding	FingerMovements
LatentCF_KDE	**1dCNN**	3.8086	**8.7929**	18.3317
LatentCF_KDE	**LSTM**	**6.1199**	**7.5362**	21.6666
LatentCF++	**1dCNN**	0.9835	6.5026	17.0007
LatentCF++	**LSTM**	1.262	**7.8935**	34.1804

Πίνακας αριθμός: KDE απόσταση αντιπαραδειγμάτων με τα ανώμαλα δεδομένα

KDE ομαλών δεδομένων

		PenDigits	WalkingSittingStanding	FingerMovements
LatentCF_KDE	**1dCNN**	4.2578	**0.9722**	54.9665
LatentCF_KDE	**LSTM**	**1.9465**	**2.2288**	62.5145
LatentCF++	**1dCNN**	7.5988	7.7153	58.6783
LatentCF++	**LSTM**	6.804	**1.8716**	87.5720

Πίνακας αριθμός: KDE απόσταση αντιπαραδειγμάτων με τα ομαλά δεδομένα

Στούς δύο παραπάνω πίνακες παρουσίαζεται η απόσταση της κατανομής των αντιπα ραδειγμάτων με τα ανώμαλα και τα ομαλά αντίστοιχα δεδομένα. Η συγκεκριμένη μετ ρική απαιτεί να γίνεται συγκριτική αξιολόγιση της απόστασης της κατανομής απο τα ομαλά δεδομένα και τα ανώμαλα και όχι των αποστάσων μεταξύ των μεθόδων. Για π αράδειγμα η LatentCF_KDE με χρήση LSTM μοντέλων του σετ δεδομένου PenDigits έχε ιαπόσταση κατανομής απο τα ανώμαλα δεδομένα **6.1199** ενώ απο τα ομαλά **1.9465**. Αυτό υποδηλώνει ότι η κατανομή είναι πιο κοντά στην επιθυμητή κλάση, κάτι που κα θιστά τη μετρική «καλή».

Παρατηρείται ότι η LatentCF_KDE έχει πολύ καλα αποτελέσματα σε τρείς συνδιασμούς μοντέλων και σετ δεδομένων, ενώ η LatentCF++ μόνο σε μία. Ακόμα και στις περιπτώσεις που η LatentCF_KDE δεν έχει καλά αποτελέσματα υπερυσχύει καθώς πλησιάζει περισσότερο την ομαλή κλάση απο την LatentCF++ παρόλο που μοιάζουν οι κατανομές και των δύο μεθόδων περισσότερο στην ανώμαλη κλάση. Για παράδειγμα, στο σετ δεδομένων Finger Movements, παρά το γεγονός οτι καμία μέθοδος δεν κατάφερε να πλησίασει την ομαλή κλάση, η LatentCF_KDE έχει πιο όμοια κατανομή. Επιπλέον, στην περίπτωση των PenDigits με την χρήση συνελικτικών μοντέλων η LatentCF_KDE έχει καλύτερα αποτελέσματα απο την LatentCF++.

Αξίζει επίσης να σημειωθεί οτι στην συγκεκριμένη μετρική δεν απορρύπτονται οι συνδιασμοί μεθόδων και μοντέλων που δεν είχαν καλό Validity καθώς έχει σημασία αν πλησίασαν ή όχι τα όμαλα δεδομένα. Παρόλο που δεν άλλαξαν κλάση υποδεικνύει σωστή ή λανθασμένη κατεύθυνση.

Γενικά συμπεράσματα

Βάσει των παρατηρήσεων και των αποτελεσμάτων όσον αφορά την απόσταση KDE, που περιγράφει πόσο κοντά ή όχι είναι η κατανομή των αντιπαραδειγμάτων στην επιθυμητή κλάση, αποδεικνύεται ότι η LatentCF_KDE είναι η καλύτερη μέθοδος παραγωγής αντιπαραδειγμάτων καθώς έχει την ικανότητα να προσομοιάζει περισσότερο την επιθυμητή κλάση. Τα αποτελέσματα του Validity, που υποδεικνύουν αν η κάθε μέθοδος κατάφερε όντως να παράξει αντιπαραδείγματα, δείχνουν ότι η προσθήκη σύγκρισης κατανομών στη μέθοδο LatentCF++ είναι ικανή να βελτιώσει ακόμα και την ίδια την παραγωγή αντιπαραδειγμάτων καθώς με τη σύγκριση KDE περισσότερα δεδομένα κατάφεραν να αλλάξουν κλάση.

Επομένως, η LatentCF_KDE

- παρουσιάζει την καλύτερη απόδοση σε μια γκάμα σετ δεδομένων και μοντέλων
- και προσφέρει μια πιο σωστή και αξιόπιστη λύση παραγωγής αντιπαραδειγμάτων

καθιστώντας την προτιμώμενη επιλογή για χρήση πολυδιάστατων χρονοσειρών .

Works Cited

1. **Rachana Balasubramanian, Sam Sharpe, Brian Barr, C. Bayan Bruss.** *Latent-CF: A Simple Baseline for Reverse Counterfactual Explanations.* New York : Center for Machine Learning, Capital One, 2021.

2. **Zhendong Wang, Panagiotis Papapetrou, Rami Mochaourab.** *Learning Time Series Counterfactuals via Latent Space Representations.* 2021.

3. **Van Rossum, G. & Drake.** *Python 3 Reference Manual, CA.* Scotts Valley : CreateSpace, 2009.

4. **Tiago, Carneiro., Raul, Victor, Medeiros, da, Nóbrega., Thiago, Nepomuceno., Gui-Bin, Bian., Victor, Hugo, C., de, Albuquerque., Pedro, Pedrosa, Rebouças, Filho.** *Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Aplications. .*

5. **Sahil Verma, Varich Boonsanong, Minh Hoang, Keegan E. Hines, John P. Dickerson, Chirag Shah.** *Counterfactual Explanations and Algorithmic Recourses for Machine Learning: A Review.* 2022.

6. **Guidotti, Riccardo.** *Counterfactual explanations and how to find them:.* s.l. : The Author(s), 2022.

7. **Molnar, Christoph.** *Interpretable Machine Learning.* s.l. : BOOKDOWN, 2023.

8. **Commission, European.** *REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 27 April 2016 on the protection.* 2015.

9. **Giorgos Filandrianos, Konstantinos Thomas, Edmund Dervakos, George Stamou.** *Conceptual Edits as Counterfactual Explanations.* s.l. : AILS lab, School of Electrical and Computer Engineering, National Technical University of Athens.

10. **ΜΑΣΤΡΟΜΙΧΑΛΑΚΗΣ, ΟΡΦΕΑΣ ΜΕΝΗΣ -.** *Αναγνώριση Τεχνοτροπίας Έργων Τέχνης με Συνελικτικά Νευρωνικά Δίκτυα.* Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Εθνικό Μετσόβιο Πολυτεχνείο. Αθήνα : s.n., 2019.

11. **ΜΙΑΜΗ, ΧΡΗΣΤΟΥ.** *Παρακολούθηση Διαλογικής Κατάστασης με αρχιτεκτονική Encoder-Decoder και χρήση Pointer - Generator Δικτύου.* s.l. : Εθνικό Μετσόβιο Πολυτεχνείο, 2019.

12. **ΜΑΡΚΟΥΛΕΣΚΟΥ, ΕΛΕΝΑΣ - ΜΠΙΑΝΚΑ Ν.** *Πρόβλεψη Χρονοσειρών μέσω Εικόνων με χρήση Γεννητικών Ανταγωνιστικών Νευρωνικών Δικτύων.* s.l. : Εθνικό Μετσόβιο Πολυτεχνείο, 2021. Διπλωματική Εργασία.

13. **Svajone Bekesiene ,Rasa Smaliukiene and Ramute Vaicaitiene.** *Using Artificial Neural Networks in Predicting the Level of Stress among Military Conscripts.* General Jonas Zemaitis Military Academy of Lithuania. Silo 5a, 10322 Vilnius, Lithuania : s.n., 2021.

14. **Baheti, Pragati.** *Activation Functions in Neural Networks [12 Types & Use Cases].* 2021.

15. **SHARMA, SAGAR.** *Activation Functions in Neural Networks.* s.l. : Medium, 2017.

16. **Francisco Pereira, Tom Mitchell, Matthew Botvinick.** *Machine learning classifiers and fMRI: A tutorial overview.* s.l. : NeuroImage, 2009. σσ. Volume 45, Issue 1, Supplement 1.

17. **Ng, Andrew.** *CS294A Lecture notes.*

18. **baeldung.** Autoencoders Explained. 2023.

19. **Mishra, Mayank.** *Convolutional Neural Networks, Explained.* s.l. : Towards Data Science, 2020.

20. **Ian Goodfellow, Yoshua Bengio, and Aaron Courville.** *Deep Learning.* s.l. : MIT Press, 2016.

21. **[Ηλεκτρονικό] https://courses.cs.washington.edu/courses/cse416/22su/lectures/10/lecture_10.pdf.**

22. Sepp Hochreiter, Jürgen Schmidhuber. ***Long Short-Term Memory.*** **s.l. : Neural Comput, 1997.**

23. Benjamin Lindemann, Benjamin Maschler, Nada Sahlab, Michael Weyrich. ***A survey on anomaly detection for technical systems using LSTM networks.*** **Institute of Industrial Automation and Software Engineering. Pfaffenwaldring 47, 70569, Stuttgart, Germany : University of Stuttgart, 2021.**

24. K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink and J. Schmidhuber. ***LSTM: A Search Space Odyssey.*** **s.l. : IEEE Transactions on Neural Networks and Learning Systems, vol. 28, no. 10, pp. 2222-2232, 2017.**

25. Ιωάννα, Τσιάκα. ***«Τεχνικές μηχανικής μάθησης για τη διάγνωση ατόμων που πάσχουν από τον ιό SARS-CoV-2 με χρήση ηχητικών καταγραφών βήχα.*** **Σχολή Ηλεκτρολόγων Και Μηχανικών Υπολογιστών. Αθήνα : Εθνικό Μετσόβιο Πολυτεχνείο, 2021. Διπλωματική Εργασία.**

26. Phi, Michael. ***Illustrated Guide to LSTM's and GRU's: A step by step explanation.*** **s.l. : Towards Data Science.**

27. Shankar297. ***Understanding Loss Function in Deep Learning.*** **s.l. : Analytics Vidhya, 2023.**

28. Drapala, Jaroslaw. ***Kernel Density Estimator explained step by step.*** **s.l. : Towards Data Science, 2016.**

29. ***ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION.*** **Diederik P. Kingma, Jimmy Lei Ba. 2015.**

30. Edmund Dervakos, Konstantinos Thomas, Giorgos Filandrianos, Giorgos Stamou. ***Choose your Data Wisely: A Framework for Semantic Counterfactuals.*** **s.l. : National Technical University of Athens, 2023.**

31. Giorgos Filandrianos, Edmund Dervakos, Orfeas Menis-Mastromichalakis, Chrysoula Zerva and Giorgos Stamou. ***Counterfactuals of Counterfactuals: a back-translation-inspired approach to analyse counterfactual editors.*** **s.l. : National Technical University of Athens, 2023.**

32. Emre Ates, , Burak Aksar, , Vitus J. Leung, Ayse K. Coskun. ***Counterfactual Explanations for Machine Learning on Multivariate Time Series Data.*** **Boston, MA, USA : Boston University, 2020.**

33. Maria Lymperaiou, Giorgos Filandrianos, Konstantinos Thomas and Giorgos Stamou. ***Counterfactual Edits for Generative Evaluation.*** **s.l. : AILS Laboratory, School of Electrical and Computer Engineering, National Technical University of Athens, 2023.**

34. Looveren, A. V.; and Klaise, J. ***Interpretable Counterfactual Explanations Guided by Prototypes.*** **2019.**

35. timeseriesclassification.com. [Ηλεκτρονικό] https://timeseriesclassification.com.

36. PenDigits Dataset . ***timeseriesclassification.com.*** **[Ηλεκτρονικό] https://www.timeseriesclassification.com/description.php?Dataset=PenDigits.**

37. FingerMovements Dataset. ***timeseriesclassification.com.*** **[Ηλεκτρονικό] https://www.timeseriesclassification.com/description.php?Dataset=FingerMovements.**

38. Maresa, Schröder. ***Explanations from the Latent Space: The Need for Latent Feature Saliency Detection in Deep Time Series Classification.*** **s.l. : Technical University of Munich, 2022. Master's Thesis in Mathematics.**

39. ΝΤΟΥΝΗ, ΠΕΤΡΟΥ. Surveillance system for Mask detection using AI. ***2023.***

40. timeseriesclassification.com. ***WalkingSittingStanding Dataset.*** **[Ηλεκτρονικό] https://www.timeseriesclassification.com/description.php?Dataset=WalkingSittingStanding.**

41. ΜΙΑΜΗ, ΧΡΗΣΤΟΥ. ***Παρακολούθηση Διαλογικής Κατάστασης με αρχιτεκτονική Encoder-Decoder και χρήση Pointer-Generator Δικτύου.*** **Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Εθνικό Μετσόβιο Πολυτεχνείο. Αθήνα : s.n., 2019. Διπλωματική Εργασία.**