National Technical University of Athens
Data Science and Machine Learning
Speech and Language Processing Laboratory

Speech Emotion Recognition using Contrastive Learning

MSc Thesis
of
Iatropoulos Petros

Supervisor: Alexandros Potamianos
NTUA Professor

January 19, 2023

Abstract

Contrastive learning is a machine learning technique that aims to optimize the similarity between different data points. It has gained attention in various domains, including speech emotion recognition (SER), which refers to the task of identifying the emotional state of a speaker from their speech. In this work, the authors investigate the effectiveness of various contrastive learning methods for SER, including supervised contrastive losses (Triplet, NT-Xent, SupCon), self-supervised contrastive pre-training, and a combination of self-supervised pre-training and supervised fine-tuning.

Early work in SER focused on extracting a set of emotion features and determining the optimal time-scale for emotional context extraction. These features were extracted from speech frames using Low Level Descriptors (LLDs) such as Mel Frequency Cepstral Coefficients (MFCCs), pitch, short-time energy, Zero Crossing Rate (ZCR), and Harmonic to Noise Ratio (HNR). These LLDs were aggregated using statistical functionals or by training neural networks on top of them and summarizing the results through an attention mechanism. With the increase in computational power, SER systems began to perform feature extraction using neural networks that were trained on spectrograms or even raw speech signals.

Recent works in SER have attempted to improve performance using contrastive learning techniques. In some cases, this has involved pre-training models using a self-supervised contrastive loss (e.g., NT-Xent or Barlow Twins) and then fine-tuning them using a supervised triplet loss. Others have used features from pre-trained models that were trained using a self-supervised contrastive loss (e.g., wav2vec 2.0).

In this work, we applied several contrastive learning methods and measured their effect on SER. We found that supervised contrastive losses did not significantly improve performance compared to supervised cross-entropy training. However, self-supervised pre-training and supervised fine-tuning with cross-entropy performed better than simply training with cross-entropy. Pre-training with the NT-Xent loss and fine-tuning with the cross-entropy loss resulted in the best performance. The authors also found that using a larger dataset for pre-training improved performance, and that the combination of self-supervised pre-training and supervised fine-tuning was more effective than either approach alone.

Overall, the results of this work suggest that self-supervised pre-training and supervised fine-tuning with cross-entropy is a promising approach for SER, and that using a larger dataset for pre-training can further improve performance. Further research is needed to fully understand the benefits and limitations of contrastive learning for SER.

Περίληψη στα Ελληνικά

Η αντιθετική μάθηση (Contrastive Learning) είναι μια τεχνική μηχανικής μάθησης που στοχεύει στη βελτιστοποίηση της ομοιότητας μεταξύ διαφορετικών σημείων δεδομένων. Έχει κερδίσει την προσοχή σε διάφορους τομείς, συμπεριλαμβανομένης της αναγνώρισης συναισθημάτων ομιλίας (Speech Emotion Recognition - SER), η οποία αναφέρεται στο έργο της αναγνώρισης της συναισθηματικής κατάστασης ενός ομιλητή από την ομιλία του. Σε αυτή την εργασία, οι συγγραφείς διερευνούν την αποτελεσματικότητα διαφόρων μεθόδων αντιθετικής μάθησης για το SER, συμπεριλαμβανομένων των εποπτευόμενων απωλειών αντίθεσης (contrastive losses) (Triplet, NT-Xent, SupCon), της αυτοεποπτευόμενης προεκπαίδευσης αντίθεσης και ενός συνδυασμού αυτοεπιβλεπόμενης προεκπαίδευσης και εποπτευόμενης μικρορύθμισης.

Η πρώιμη εργασία στο SER επικεντρώθηκε στην εξαγωγή ενός συνόλου χαρακτηριστικών συναισθημάτων και στον καθορισμό της βέλτιστης χρονικής κλίμακας για την εξαγωγή συναισθηματικού πλαισίου. Αυτά τα χαρακτηριστικά εξήχθησαν από πλαίσια ομιλίας χρησιμοποιώντας Περιγραφείς Χαμηλού Επιπέδου (Low Level Descriptors - LLD) όπως Συντελεστές Mel Cepstral Συχνότητας (MFCCs), Θεμελιώδης συχνότητα (F0), βραχυχρόνια ενέργεια (Short-term Energy), ρυθμός διέλευσης από το μηδέν (Zero Crossing Rate - ZCR) και λόγος αρμονικών προς θόρυβο (Harmonic to Noise Ration - HNR). Αυτά τα LLD συγκεντρώθηκαν χρησιμοποιώντας στατιστικές συναρτήσεις ή εκπαιδεύοντας νευρωνικά δίκτυα πάνω τους και συνοψίζοντας τα αποτελέσματα μέσω ενός μηχανισμού προσοχής. Με την αύξηση της υπολογιστικής ισχύος, τα συστήματα SER άρχισαν να εκτελούν εξαγωγή χαρακτηριστικών χρησιμοποιώντας νευρωνικά δίκτυα που είχαν εκπαιδευτεί σε φασματογράμματα ή ακόμα και ακατέργαστα σήματα ομιλίας.

Πρόσφατες εργασίες στο SER προσπάθησαν να βελτιώσουν την απόδοση χρησιμοποιώντας τεχνικές αντιθετικής μάθησης. Σε ορισμένες περιπτώσεις, αυτό περιλάμβανε μοντέλα προ-εκπαίδευσης χρησιμοποιώντας μια αυτοεπιβλεπόμενη συνάρτηση σφάλματος αντίθεσης (π.χ. NT-Xent ή Barlow Twins) και στη συνέχεια βελτίωσή τους χρησιμοποιώντας μια εποπτευόμενη συνάρτηση σφάλματος τριπλής. Άλλοι έχουν χρησιμοποιήσει χαρακτηριστικά από προεκπαιδευμένα μοντέλα που εκπαιδεύτηκαν χρησιμοποιώντας μια αυτοεπιβλεπόμενη συνάρτηση σφάλματος αντίθεσης (π.χ. wav2vec 2.0).

Σε αυτή την εργασία, εφαρμόσαμε διάφορες μεθόδους αντιθετικής μάθησης και μετρήσαμε την επίδρασή τους στο SER. Διαπιστώσαμε ότι οι εποπτευόμενες συναρτήσεις σφάλματος αντίθεσης δεν βελτίωσαν σημαντικά την απόδοση σε σύγκριση με την εποπτευόμενη εκπαίδευση διασταυρούμενης εντροπίας (cross entropy). Ωστόσο, η αυτο-εποπτευόμενη προ-εκπαίδευση και η εποπτευόμενη μικρορύθμιση με διασταυρούμενη εντροπία απέδωσαν καλύτερα από την απλή εκπαίδευση με διασταυρούμενη εντροπία. Η προ-εκπαίδευση με την NT-Xent και η μικρορύθμιση με τη συνάρτηση σφάλματος διασταυρούμενης εντροπίας είχαν ως αποτέλεσμα την καλύτερη απόδοση. Οι συγγραφείς διαπίστωσαν επίσης ότι η χρήση ενός μεγαλύτερου συνόλου δεδομένων για προ-εκπαίδευση βελτίωσε την απόδοση και ότι ο συνδυασμός της αυτο-εποπτευόμενης προ-εκπαίδευσης και της εποπτευόμενης μικρορύθμισης ήταν πιο αποτελεσματικός από κάθε προσέγγιση μόνη της.

Συνολικά, τα αποτελέσματα αυτής της εργασίας υποδηλώνουν ότι η αυτο-εποπτευόμενη προ-εκπαίδευση και η εποπτευόμενη μικρορύθμιση με διασταυρούμενη εντροπία είναι μια πολλά υποσχόμενη προσέγγιση για το SER και ότι η χρήση ενός μεγαλύτερου συνόλου δεδομένων για προεκπαίδευση μπορεί να βελτιώσει περαιτέρω την απόδοση. Απαιτείται

περαιτέρω έρευνα για την πλήρη κατανόηση των πλεονεκτημάτων και των περιορισμών της αντιθετικής μάθησης για το SER.

Εκτεταμένη περίληψη στα Ελληνικά

0.0.1 Εισαγωγή

Η διαδικασία παραγωγής λόγου στον άνθρωπο είναι μια πολύπλοκη και συντονισμένη προσπάθεια που περιλαμβάνει πολλά διαφορετικά συστήματα στο σώμα, συμπεριλαμβανομένων των αναπνευστικών, φωνητικών και αρθρωτικών συστημάτων, καθώς και γνωστικές διεργασίες στον εγκέφαλο. Το αναπνευστικό σύστημα παρέχει τον αέρα που χρειάζεται για την ομιλία, το φωνητικό σύστημα παράγει ηχητικά κύματα δονώντας τις φωνητικές χορδές στον λάρυγγα και το αρθρικό σύστημα διαμορφώνει τα ηχητικά κύματα σε αναγνωρίσιμους ήχους ομιλίας μετακινώντας τα χείλη, τη γλώσσα και άλλους αρθρωτές. Οι κινήσεις των αρθρώσεων ελέγχονται από τον κινητικό φλοιό, μια περιοχή του εγκεφάλου που είναι υπεύθυνη για τον έλεγχο των εκούσιων κινήσεων. Επιπλέον, άλλες περιοχές του εγκεφάλου, όπως η περιοχή του Broca και η περιοχή του Wernicke, που είναι γλωσσικά κέντρα που βρίσκονται στον μετωπιαίο και κροταφικό λοβό, αντίστοιχα, εμπλέκονται επίσης στην παραγωγή και την κατανόηση του λόγου. Αυτές οι γνωστικές διαδικασίες μας βοηθούν να κατανοήσουμε τι θέλουμε να πούμε και να επιλέξουμε τις κατάλληλες λέξεις και ήχους που θα χρησιμοποιήσουμε στην ομιλία μας.

Εκτός από τους μηχανισμούς παραγωγής λόγου, σημαντικό ρόλο στην επικοινωνία παίζει και η έκφραση και η αντίληψη των συναισθημάτων. Η παραγωγή συναισθημάτων κατά την ομιλία αναφέρεται στον τρόπο με τον οποίο τα συναισθήματα μεταδίδονται μέσω της ομιλίας και μπορεί να περιλαμβάνει παράγοντες όπως ο τόνος της φωνής, ο τόνος και ο ρυθμός. Η αντίληψη του συναισθήματος της ομιλίας αναφέρεται στην ικανότητα του ακροατή να κατανοεί και να ερμηνεύει τα συναισθήματα που μεταφέρονται μέσω της ομιλίας. Αυτή η ικανότητα επηρεάζεται από διάφορους παράγοντες όπως η συναισθηματική κατάσταση του ακροατή, το πολιτισμικό υπόβαθρο και οι προηγούμενες εμπειρίες.

Τα συναισθήματα μπορούν να μεταφέρουν σημαντικές πληροφορίες για τις προθέσεις και τις στάσεις του ομιλητή και μπορούν επίσης να βοηθήσουν στη δημιουργία κοινωνικών συνδέσεων και σχέσεων. Ωστόσο, η ικανότητα αντίληψης των συναισθημάτων στην ομιλία δεν είναι πάντα ακριβής και μπορεί να επηρεαστεί από διάφορους παράγοντες. Για παράδειγμα, άτομα με ορισμένες διαταραχές όπως η διαταραχή του φάσματος του αυτισμού μπορεί να έχουν δυσκολία στην αντίληψη των συναισθημάτων στην ομιλία. Επιπλέον, το πολιτιστικό και γλωσσικό υπόβαθρο μπορεί επίσης να παίξει ρόλο στον τρόπο με τον οποίο τα συναισθήματα γίνονται αντιληπτά και εκφράζονται στην ομιλία.

Η έκφραση και η αντίληψη των συναισθημάτων στην ομιλία είναι ένας σημαντικός τομέας μελέτης καθώς έχει εφαρμογές σε τομείς όπως η ψυχολογία, η γλωσσολογία και η επιστήμη των υπολογιστών. Στην ψυχολογία, χρησιμοποιείται για να κατανοήσει πώς τα συναισθήματα επηρεάζουν την επικοινωνία και τις κοινωνικές αλληλεπιδράσεις. Στη γλωσσολογία, χρησιμοποιείται για να κατανοήσει πώς εκφράζονται και αντιλαμβάνονται τα συναισθήματα σε διαφορετικές γλώσσες και πολιτισμούς. Στην επιστήμη των υπολογιστών, χρησιμοποιείται για την ανάπτυξη πιο φυσικών και ανθρωπογενών συστημάτων αναγνώρισης ομιλίας και ομιλίας που δημιουργούνται από υπολογιστή.

Συνολικά, η διαδικασία παραγωγής λόγου είναι μια πολύπλοκη και συντονισμένη προσπάθεια που περιλαμβάνει πολλά διαφορετικά συστήματα στο σώμα και γνωστικές διεργασίες στον εγκέφαλο. Η έκφραση και η αντίληψη των συναισθημάτων παίζουν επίσης σημαντικό ρόλο στην επικοινωνία, μεταφέροντας σημαντικές πληροφορίες για τις προθέσεις

και τις στάσεις του ομιλητή και βοηθώντας στη δημιουργία κοινωνικών συνδέσεων και σχέσεων. Αυτοί οι τομείς σπουδών έχουν εφαρμογές σε διάφορους τομείς και μπορούν να παρέχουν πληροφορίες για το πώς επικοινωνούμε και κατανοούμε τα συναισθήματα.

Το τμήμα τεχνικού υποβάθρου της τρέχουσας εργασίας εστιάζει στην ψηφιακή αναπαράσταση των σημάτων ομιλίας. Το αναλογικό σήμα ομιλίας μετατρέπεται πρώτα σε σήμα διακριτού χρόνου δειγματοληπτώντας το σε τακτά χρονικά διαστήματα και στη συνέχεια κβαντοποιείται χαρτογραφώντας το πλάτος κάθε δείγματος σε μια ψηφιακή τιμή από ένα πεπερασμένο σύνολο πιθανών τιμών. Οι ψηφιακές τιμές αντιπροσωπεύονται τυπικά με χρήση δυαδικού κώδικα. Ο ρυθμός δειγματοληψίας και ο αριθμός των bit που χρησιμοποιούνται για την κβαντοποίηση καθορίζουν την ποιότητα της ψηφιακής αναπαράστασης.

Μόλις το αναλογικό σήμα ομιλίας μετατραπεί σε ψηφιακό σήμα, μπορεί εύκολα να επεξεργαστεί, να μεταδοθεί και να αποθηκευτεί χρησιμοποιώντας ψηφιακές συσκευές και συστήματα. Αυτό επιτρέπει ένα ευρύ φάσμα εφαρμογών, όπως η επεξεργασία ψηφιακού σήματος, η αναγνώριση ομιλίας και οι τηλεπικοινωνίες. Επιπλέον, η ψηφιακή αναπαράσταση των σημάτων ομιλίας επιτρέπει τη χρήση κωδικών διόρθωσης σφαλμάτων, οι οποίοι μπορούν να βοηθήσουν στη μείωση της επίδρασης του θορύβου και άλλων πηγών παραμόρφωσης στο σήμα.

0.0.2 Τεχνικό υπόβαθρο και προαπαιτούμενα

Η ψηφιακή αναπαράσταση των σημάτων ομιλίας μπορεί να γίνει στο πεδίο χρόνου ή στο πεδίο συχνότητας. Στην ψηφιακή αναπαράσταση στον τομέα του χρόνου, το σήμα ομιλίας αναπαρίσταται ως μια ακολουθία ψηφιακών δειγμάτων στο χρόνο. Αυτό σημαίνει ότι το πλάτος του σήματος ομιλίας δειγματοληπτείται σε τακτά χρονικά διαστήματα και τα δείγματα κβαντίζονται και κωδικοποιούνται χρησιμοποιώντας έναν δυαδικό κώδικα. Η προκύπτουσα ακολουθία ψηφιακών τιμών μπορεί να θεωρηθεί ως μια σειρά από στιγμιότυπα του σήματος ομιλίας σε διαφορετικά χρονικά σημεία. Ένα πλεονέκτημα της αναπαράστασης τομέα χρόνου είναι ότι αντανακλά άμεσα τη χρονική δομή του σήματος ομιλίας. Ωστόσο, μπορεί να είναι δύσκολο να αναπαραστήσουμε ορισμένους τύπους σημάτων χρησιμοποιώντας μια αναπαράσταση τομέα χρόνου με περιορισμένο ρυθμό δειγματοληψίας.

Στην ψηφιακή αναπαράσταση του τομέα συχνότητας, το σήμα ομιλίας αναπαρίσταται ως προς το περιεχόμενο συχνότητάς του. Αυτό γίνεται συνήθως χρησιμοποιώντας μια διαδικασία που ονομάζεται μετασχηματισμός Fourier, η οποία αποσυνθέτει το σήμα σε ένα σύνολο ημιτονοειδών σημάτων με διαφορετικές συχνότητες, πλάτη και φάσεις. Η προκύπτουσα αναπαράσταση τομέα συχνότητας του σήματος ομιλίας ονομάζεται συχνά φάσμα του σήματος. Ένα πλεονέκτημα της αναπαράστασης τομέα συχνότητας είναι ότι αντανακλά άμεσα το περιεχόμενο συχνότητας του σήματος ομιλίας. Ωστόσο, μπορεί να είναι δύσκολο να αναπαραστήσουμε ορισμένους τύπους σημάτων, όπως σήματα με ταχεία χρονική αλλαγή, χρησιμοποιώντας μια αναπαράσταση τομέα συχνότητας.

0.0.3 Μηχανική και Βαθιά Μηχανική Μάθηση

Το τεχνικό υπόβαθρο αυτής της εργασίας περιλαμβάνει τη χρήση της βαθιάς μάθησης και των νευρωνικών δικτύων. Η βαθιά μάθηση είναι ένα υποπεδίο της μηχανικής μάθησης που χρησιμοποιεί πολύ βαθιά νευρωνικά δίκτυα, τα οποία βελτιστοποιούνται χρησιμοποιώντας παραλληλισμένους αλγόριθμους κυρίως σε GPU ή TPU. Αυτά τα νευρωνικά δίκτυα

αποτελούνται από δομικά στοιχεία που μετασχηματίζουν διαδοχικά τα δεδομένα. Αυτές οι συναρτήσεις είναι μαθησιακά επίπεδα λειτουργιών που συντίθενται για να σχηματίσουν ένα δίκτυο. Ανάλογα με τη δομή των δεδομένων και την εργασία στο χέρι, έχουν προταθεί διαφορετικές αρχιτεκτονικές επιπέδων.

Μια τέτοια αρχιτεκτονική είναι τα συνελικτικά δίκτυα. Τα συνελικτικά επίπεδα υλοποιούν τη λειτουργία συνέλιξης μεταξύ του σήματος εισόδου και ενός συνόλου παραμέτρων που μπορούν να μάθουν, που συνήθως αναφέρονται ως πυρήνες ή φίλτρα. Η έξοδος αυτών των λειτουργιών αναφέρεται συνήθως ως χάρτης χαρακτηριστικών. Η λειτουργία συνέλιξης είναι μια λειτουργία ισοδύναμη μετάφρασης, που σημαίνει ότι μετακινείται με τη λειτουργία μετάφρασης. Η ισοδυναμία μετάφρασης είναι μια εξαιρετικά χρήσιμη ιδιότητα για νευρωνικά δίκτυα που χρησιμοποιούνται για εργασίες για τις οποίες δεν απαιτείται συγκεκριμένα ¨πού/πότε' υπάρχει ένα χαρακτηριστικό αλλά ¨εάν' υπάρχει ένα χαρακτηριστικό. Επιτρέπει επίσης στα νευρωνικά δίκτυα να λειτουργούν σε εισόδους με διαφορετικό μήκος.

Μια άλλη αρχιτεκτονική είναι τα αναδρομικά δίκτυα. Τα αναδρομικά νευρωνικά δίκτυα (RNN) χρησιμοποιούνται για την επεξεργασία διαδοχικών δεδομένων και τη μοντελοποίηση δυναμικών συστημάτων. Η περιγραφή της κατάστασης ενός δυναμικού συστήματος εξαρτάται από ένα σήμα εισόδου με παραμέτρους. Οι εξισώσεις της προς τα εμπρός διάδοσης ενός RNN είναι, όπου h είναι η κρυφή κατάσταση, ο είναι η έξοδος και πηι είναι μια συνάρτηση ενεργοποίησης. Η κύρια πρόκληση στην εκπαίδευση RNN είναι ότι οι διαβαθμίσεις των παραμέτρων μπορεί να εξαφανιστούν ή να εκραγούν καθώς το σφάλμα διαδίδεται πίσω στο χρόνο. Μια παραλλαγή των RNN είναι τα LSTM (μακροπρόθεσμη μνήμη) τα οποία ξεπερνούν αυτό το πρόβλημα εισάγοντας ένα κελί μνήμης, πύλες εισόδου, πύλες λήθης και πύλες εξόδου.

Υπάρχουν πολλές άλλες αρχιτεκτονικές όπως αυτοκωδικοποιητές (autoencoders), Transformers και Self-Attention που χρησιμοποιούνται στη βαθιά εκμάθηση. Κάθε αρχιτεκτονική χρησιμοποιείται σε συγκεκριμένες καταστάσεις και εφαρμογές. Η επιλογή της αρχιτεκτονικής εξαρτάται από τα διαθέσιμα δεδομένα, το πρόβλημα και τους διαθέσιμους υπολογιστικούς πόρους.

0.0.4 Αντιθετική μάθηση (Contrastive Learning)

Σίγουρα, η εκμάθηση μετρήσεων είναι ένας υποτομέας της μηχανικής μάθησης που στοχεύει να μάθει μια αντιστοίχιση από έναν τομέα $\mathcal{X}$ σε έναν χώρο υψηλών διαστάσεων, έτσι ώστε παρόμοια σημεία να αντιστοιχίζονται κοντά το ένα στο άλλο. Ο στόχος είναι να βρεθεί μια αντιστοίχιση f τέτοια ώστε η απόσταση μεταξύ παρόμοιων σημείων x και x^+ να είναι μικρότερη από την απόσταση μεταξύ ανόμοιων σημείων x και x^-. Η συνάρτηση απόστασης d που χρησιμοποιείται στη χαρτογράφηση μπορεί να είναι η Ευκλείδεια απόσταση ή η απόσταση συνημιτόνου. Ο ορισμός της ομοιότητας στη χαρτογράφηση απαιτεί πλαίσιο, καθώς δύο δείγματα μπορεί να θεωρηθούν παρόμοια σε ένα πλαίσιο και ανόμοια σε ένα άλλο. Οι μετρικές μέθοδοι εκμάθησης μπορούν να είναι υπό επίβλεψη ή χωρίς επίβλεψη, ανάλογα με τον ορισμό της ομοιότητας.

Οι μέθοδοι γραμμικής μετρικής εκμάθησης χρησιμοποιούν μια γραμμική αντιστοίχιση $f(x) = Wx$, και όταν χρησιμοποιείται η Ευκλείδεια απόσταση, αυτές οι μέθοδοι αναφέρονται ως «Μάθηση μετρικής απόστασης Mahalanobis». Η απόσταση Mahalanobis ορίζεται ως η απόσταση μεταξύ δύο σημείων σε έναν πολυδιάστατο χώρο, σε σχέση με έναν πίνακα συνδιακύμανσης. Η απόσταση Mahalanobis είναι χρήσιμη σε περιπτώσεις

όπου η κατανομή των δεδομένων είναι μη ισοτροπική (δηλαδή τα δεδομένα δεν κατανέμονται εξίσου προς όλες τις κατευθύνσεις).

Ένα παράδειγμα μεθόδου γραμμικής μετρικής εκμάθησης είναι η Ανάλυση Στοιχείων Γειτονίας (NCA), η οποία εκπαιδεύει έναν ταξινομητή «μαλακού» πλησιέστερου γείτονα για να μεγιστοποιήσει την πιθανότητα εκχώρησης της σωστής κλάσης. Η NCA μαθαίνει έναν γραμμικό μετασχηματισμό έτσι ώστε τα μετασχηματισμένα δεδομένα να έχουν μια δομή χαμηλότερης διάστασης που διατηρεί τις τοπικές πληροφορίες γειτονιάς των δεδομένων, κάτι που είναι επωφελές για ταξινόμηση.

Οι μέθοδοι εκμάθησης βαθιάς μετρικής, από την άλλη πλευρά, χρησιμοποιούν μια μη γραμμική χαρτογράφηση f που μαθαίνεται από ένα νευρωνικό δίκτυο. Αυτές οι μέθοδοι είναι πιο ισχυρές από τις γραμμικές μεθόδους επειδή μπορούν να μάθουν πιο σύνθετες αναπαραστάσεις των δεδομένων. Ένα παράδειγμα μεθόδου εκμάθησης βαθιάς μετρικής είναι τα σιαμαία δίκτυα, το οποίο είναι μια αρχιτεκτονική νευρωνικών δικτύων που αποτελείται από δύο πανομοιότυπα υποδίκτυα, καθένα από τα οποία επεξεργάζεται ένα από τα δείγματα εισόδου. Η έξοδος κάθε υποδικτύου τροφοδοτείται στη συνέχεια σε μια συνάρτηση απώλειας αντίθεσης, η οποία συγκρίνει την ομοιότητα των δύο δειγμάτων εισόδου. Η λειτουργία απώλειας αντίθεσης έχει σχεδιαστεί για να ελαχιστοποιεί την απόσταση μεταξύ της εξόδου των δύο υποδικτύων για παρόμοια δείγματα εισόδου και να μεγιστοποιεί την απόσταση για ανόμοια δείγματα εισόδου.

Ένα άλλο παράδειγμα μεθόδου εκμάθησης βαθιάς μετρικής είναι το δίκτυο Triplet, το οποίο είναι μια αρχιτεκτονική νευρωνικού δικτύου που λαμβάνει τρία δείγματα εισόδου: ένα δείγμα άγκυρα, ένα θετικό δείγμα και ένα αρνητικό δείγμα. Το δείγμα άγκυρας και το θετικό δείγμα θεωρούνται όμοια, ενώ το δείγμα άγκυρας και το αρνητικό δείγμα θεωρούνται ανόμοια. Ο στόχος του δικτύου Triplet είναι να μάθει μια μη γραμμική απεικόνιση που απεικονίζει το δείγμα αγκύρας και το θετικό δείγμα πιο κοντά μεταξύ τους από το δείγμα αγκύρωσης και το αρνητικό δείγμα.

Η αντιθετική μάθηση είναι μια μέθοδος που χρησιμοποιεί συναρτήσεις απώλειας αντίθεσης για να μάθει αναπαραστάσεις των δεδομένων. Η λειτουργία απώλειας αντίθεσης έχει σχεδιαστεί για να ελαχιστοποιεί την απόσταση μεταξύ της εξόδου των δύο υποδικτύων για παρόμοια δείγματα εισόδου και να μεγιστοποιεί την απόσταση για ανόμοια δείγματα εισόδου. Η κύρια ιδέα πίσω από την αντιθετική μάθηση είναι ότι παρόμοια δείγματα πρέπει να έχουν παρόμοιες αναπαραστάσεις, ενώ τα ανόμοια δείγματα πρέπει να έχουν ανόμοιες αναπαραστάσεις. Η αντιθετική μάθηση μπορεί να χρησιμοποιηθεί για μάθηση χωρίς επίβλεψη ή αυτο-επίβλεψη, όπου ο στόχος είναι να μάθουμε μια καλή αναπαράσταση των δεδομένων χωρίς δεδομένα με ετικέτα.

Αυτές οι τεχνικές χρησιμοποιούνται συνήθως σε διάφορες εργασίες, όπως η αναγνώριση εικόνας, η αναγνώριση προσώπου και η επαλήθευση υπογραφής.

Η εποπτευόμενη προσέγγιση είναι μια μέθοδος όπου η θετική κατανομή ορίζεται ως το σύνολο όλων των άλλων δειγμάτων από την παρτίδα που μοιράζονται την ίδια κατηγορία με το δείγμα αγκύρωσης. Μαθηματικά, αυτό μπορεί να αναπαρασταθεί ως:

$$P(a) = \{p \in A | y_a = y_p, a \neq p\}$$

Όπου a είναι το δείγμα αγκύρωσης, το y_a είναι η ετικέτα για αυτό το δείγμα και το $P(a)$ είναι το σύνολο όλων των άλλων δειγμάτων της παρτίδας που μοιράζονται την ίδια ετικέτα με την άγκυρα. Από την άλλη πλευρά, τα αρνητικά ορίζονται ως τα δείγματα που

έχουν διαφορετική ετικέτα από το δείγμα αγκύρωσης. Αυτό μπορεί να αναπαρασταθεί μαθηματικά ως:

$$N(a) = \{n \in A | y_a \neq y_n\}$$

Στην περίπτωση των σιαμαίων δικτύων, για κάθε δείγμα x_i, το δίκτυο παρέχεται επίσης με ένα άλλο δείγμα x_j. Το δίκτυο εφαρμόζεται και στα δύο δείγματα και παράγει δύο κωδικοποιημένα διανύσματα $\mathbf{z}_i, \mathbf{z}_j$. Η απώλεια υπολογίζεται συγκρίνοντας την απόσταση μεταξύ των δύο κωδικοποιημένων διανυσμάτων. Εάν τα δύο δείγματα θεωρούνται παρόμοια, η απώλεια είναι η απόσταση μεταξύ των διανυσμάτων κωδικοποίησης, ενώ εάν θεωρούνται διαφορετικά η απώλεια είναι η άρνηση της απόστασής τους προσθέτοντας ένα θετικό περιθώριο. Επίσημα αυτό μπορεί να εκπροσωπηθεί ως:

$$\ell_{ij} = y_{ij} d(f_\theta(x_i), f_\theta(x_j)) + (1 - y_{ij})[m - d(f_\theta(x_i), f_\theta(x_j))]+$$

Όπου y_{ij} είναι ένας δείκτης που ισούται με 1 εάν τα δείγματα x_i, τα x_j είναι παρόμοια και 0 αλλού και $m > 0$ είναι η παράμετρος περιθωρίου.

Η απώλεια τριπλής είναι μια επέκταση της αρχιτεκτονικής σιαμαίων δικτύων. Αντί να συγκρίνει δύο εισόδους, συγκρίνει τρεις εισόδους: ένα δείγμα αγκύρωσης, ένα θετικό δείγμα και ένα αρνητικό δείγμα. Ο στόχος είναι να γίνει η απόσταση μεταξύ της άγκυρας και του θετικού δείγματος όσο το δυνατόν μικρότερη, ενώ η απόσταση μεταξύ της άγκυρας και του αρνητικού δείγματος όσο το δυνατόν μεγαλύτερη. Αυτό μπορεί να αναπαρασταθεί μαθηματικά ως:

$$\mathcal{L}(A, P, N) = \max\left(d(A, P) - d(A, N) + \alpha, 0\right)$$

Όπου τα A, P και N αντιπροσωπεύουν τα δείγματα άγκυρας, θετικά και αρνητικά αντίστοιχα, το $d(\cdot, \cdot)$ αντιπροσωπεύει τη μέτρηση της απόστασης και το α είναι μια υπερπαράμετρος περιθωρίου.

Ο παρεχόμενος κώδικας συζητά τη μέθοδο εκμάθησης αναπαράστασης χωρίς επίβλεψη που ονομάζεται InfoNCE/NT-Xent. Αυτή η μέθοδος είναι μια διατύπωση πολλαπλών αρνητικών, που σημαίνει ότι αντί να δειγματίζονται ρητά αρνητικά παραδείγματα, τα άλλα παραδείγματα σε μια μίνι παρτίδα αντιμετωπίζονται ως αρνητικά. Αυτή η μέθοδος χρησιμοποιείται σε μάθηση χωρίς επίβλεψη, πράγμα που σημαίνει ότι δεν βασίζεται σε δεδομένα με ετικέτα για την εκμάθηση των αναπαραστάσεων.

Η μέθοδος βασίζεται στην ιδέα του ορισμού θετικών και αρνητικών κατανομών (P(a), N(a)) με μια άγκυρα α. Η επιλογή αυτών των διανομών ρυθμίζεται εξ ολοκλήρου από τον χρήστη. Η διαφορά μεταξύ των διαφορετικών μεθόδων έγκειται στον τρόπο ορισμού αυτών των δύο κατανομών.

Η συγκεκριμένη μέθοδος που συζητείται στον κώδικα είναι η InfoNCE, η οποία εισήχθη στην εργασία "Representation Learning with Contrastive Predictive Coding" από τους Oord et al. και η απώλεια NT-Xent, η οποία εισήχθη στην εργασία "Improving Language Understanding by Generative Pre-Training" από τους Chen et al. Αυτές οι μέθοδοι έχουν χρησιμοποιηθεί σε πολλές άλλες εργασίες, όπως CPC, CMC, MoCo και CPC v2 για εκμάθηση αναπαράστασης χωρίς επίβλεψη.

Η συνάρτηση απώλειας που ελαχιστοποιείται για την εκμάθηση των αναπαραστάσεων έχει τη μορφή:

$$\mathcal{L} = -\mathbb{E}_X[\log \frac{g(x_{t+k}, \mathbf{c}_t)}{\sum_{x \in X} g(x, \mathbf{c}_t)}]$$

Όπου γ είναι ένας εκτιμητής του μη κανονικοποιημένου λόγου πιθανότητας:

$$g_k(x_{t+k}, \mathbf{c}_t) \propto \frac{p(x_{t+k}|\mathbf{c}_t)}{p(x_{t+k})}$$

Η λειτουργική μορφή του γ επιλέγεται να είναι ένας λογ-διγραμμικός μετασχηματισμός:

$$g_k(x_{t+k}, \mathbf{c}_t) = \exp(\mathbf{z}_{t+k} \cdot W_k \mathbf{c}_t)$$

Με αυτήν την επιλογή του γ, η βελτιστοποιημένη απώλεια παίρνει τη μορφή μιας απώλειας αντίθεσης softmax:

$$\mathcal{L} = -\mathbb{E}_X[\log\left(\frac{e^{\mathbf{z}_{t+k} \cdot W_k \mathbf{c}_t}}{e^{\mathbf{z}_{t+k} \cdot W_k \mathbf{c}_t} + \sum_{x \neq x_{t+k}} e^{\mathbf{z} \cdot W_k \mathbf{c}_t}}\right)]$$

Ο κώδικας περιλαμβάνει επίσης μια εικόνα που απεικονίζει τη μέθοδο απώλειας NTXent, η οποία δείχνει ότι μια μικρή παρτίδα N παραδειγμάτων δειγματοληπτείται τυχαία και η εργασία αντίθεσης πρόβλεψης ορίζεται σε ζεύγη επαυξημένων παραδειγμάτων που προέρχονται από τη μίνι παρτίδα, με αποτέλεσμα 2N σημεία δεδομένων. Η απώλεια μεταξύ δύο δειγμάτων i, j είναι:

$$\ell_{i,j} = -\log\left(\frac{e^{\mathbf{z}i \cdot \mathbf{z}j/\tau}}{\sum_{k=1,k\neq i}^{2N} e^{\mathbf{z}i \cdot \mathbf{z}k/\tau}}\right)$$

Εφαρμογή Αντιθετικής Μάθησης σε Αναγνώριση Συναισθήματος στην Ομιλία

Η αντιθετική μάθηση (Contrastive Learning) είναι μια τεχνική μηχανικής μάθησης που στοχεύει στη βελτιστοποίηση της ομοιότητας μεταξύ διαφορετικών σημείων δεδομένων. Έχει κερδίσει την προσοχή σε διάφορους τομείς, συμπεριλαμβανομένης της αναγνώρισης συναισθημάτων ομιλίας (Speech Emotion Recognition - SER), η οποία αναφέρεται στο έργο της αναγνώρισης της συναισθηματικής κατάστασης ενός ομιλητή από την ομιλία του. Σε αυτή την εργασία, οι συγγραφείς διερευνούν την αποτελεσματικότητα διαφόρων μεθόδων αντιθετικής μάθησης για το SER, συμπεριλαμβανομένων των εποπτευόμενων απωλειών αντίθεσης (contrastive losses) (Triplet, NT-Xent, SupCon), της αυτοεποπτευόμενης προεκπαίδευσης αντίθεσης και ενός συνδυασμού αυτοεπιβλεπόμενης προεκπαίδευσης και εποπτευόμενης μικρορύθμισης.

Η πρώιμη εργασία στο SER επικεντρώθηκε στην εξαγωγή ενός συνόλου χαρακτηριστικών συναισθημάτων και στον καθορισμό της βέλτιστης χρονικής κλίμακας για την εξαγωγή συναισθηματικού πλαισίου. Αυτά τα χαρακτηριστικά εξήχθησαν από πλαίσια ομιλίας χρησιμοποιώντας Περιγραφείς Χαμηλού Επιπέδου (Low Level Descriptors - LLD) όπως Συντελεστές Mel Cepstral Συχνότητας (MFCCs), Θεμελιώδης συχνότητα (F0), βραχυχρόνια ενέργεια (Short-term Energy), ρυθμός διέλευσης από το μηδέν (Zero Crossing Rate - ZCR) και λόγος αρμονικών προς θόρυβο (Harmonic to Noise Ration - HNR). Αυτά τα LLD συγκεντρώθηκαν χρησιμοποιώντας στατιστικές συναρτήσεις ή εκπαιδεύοντας νευρωνικά δίκτυα πάνω τους και συνοψίζοντας τα αποτελέσματα μέσω ενός μηχανισμού

προσοχής. Με την αύξηση της υπολογιστικής ισχύος, τα συστήματα SER άρχισαν να εκτελούν εξαγωγή χαρακτηριστικών χρησιμοποιώντας νευρωνικά δίκτυα που είχαν εκπαιδευτεί σε φασματογράμματα ή ακόμα και ακατέργαστα σήματα ομιλίας.

Πρόσφατες εργασίες στο SER προσπάθησαν να βελτιώσουν την απόδοση χρησιμοποιώντας τεχνικές αντιθετικής μάθησης. Σε ορισμένες περιπτώσεις, αυτό περιλάμβανε μοντέλα προ-εκπαίδευσης χρησιμοποιώντας μια αυτοεπιβλεπόμενη συνάρτηση σφάλματος αντίθεσης (π.χ. NT-Xent ή Barlow Twins) και στη συνέχεια βελτίωσή τους χρησιμοποιώντας μια εποπτευόμενη συνάρτηση σφάλματος τριπλής. Άλλοι έχουν χρησιμοποιήσει χαρακτηριστικά από προεκπαιδευμένα μοντέλα που εκπαιδεύτηκαν χρησιμοποιώντας μια αυτοεπιβλεπόμενη συνάρτηση σφάλματος αντίθεσης (π.χ. wav2vec 2.0).

Σε αυτή την εργασία, εφαρμόσαμε διάφορες μεθόδους αντιθετικής μάθησης και μετρήσαμε την επίδρασή τους στο SER. Διαπιστώσαμε ότι οι εποπτευόμενες συναρτήσεις σφάλματος αντίθεσης δεν βελτίωσαν σημαντικά την απόδοση σε σύγκριση με την εποπτευόμενη εκπαίδευση διασταυρούμενης εντροπίας (cross entropy). Ωστόσο, η αυτο-εποπτευόμενη προ-εκπαίδευση και η εποπτευόμενη μικρορύθμιση με διασταυρούμενη εντροπία απέδωσαν καλύτερα από την απλή εκπαίδευση με διασταυρούμενη εντροπία. Η προ-εκπαίδευση με την NT-Xent και η μικρορύθμιση με τη συνάρτηση σφάλματος διασταυρούμενης εντροπίας είχαν ως αποτέλεσμα την καλύτερη απόδοση. Οι συγγραφείς διαπίστωσαν επίσης ότι η χρήση ενός μεγαλύτερου συνόλου δεδομένων για προ-εκπαίδευση βελτίωσε την απόδοση και ότι ο συνδυασμός της αυτο-εποπτευόμενης προ-εκπαίδευσης και της εποπτευόμενης μικρορύθμισης ήταν πιο αποτελεσματικός από κάθε προσέγγιση μόνη της.

0.1 Πειράματα και αποτελέσματα

0.1.1 Βάση σύγκρισης

Προκειμένου να έχουμε μια βάση σύγκρισης για να συγκρίνουμε τα ακόλουθα πειράματα, ένας ταξινομητής h εκπαιδεύεται από άκρο σε άκρο χρησιμοποιώντας την τυπική απώλεια διασταυρούμενης εντροπίας. Ο κωδικοποιητής αποτελείται από έναν μη γραμμικό κωδικοποιητή f που έχει την αρχιτεκτονική που περιγράφηκε στο 4.3.2 (χωρίς $l2$-normalization) και ακολουθείται από μια εκμαθήσιμη γραμμική αντιστοίχιση g (βλ. επίσης σχήμα 4.2). Όσον αφορά την αναγνώριση συναισθημάτων ομιλίας, η οποία είναι ο βασικός στόχος, το βασικό μοντέλο έδωσε τα ακόλουθα αποτελέσματα που φαίνονται στον πίνακα 4.1.

0.1.2 Εποπτευόμενη εκμάθηση αντίθεσης

Για αυτό το σύνολο πειραμάτων, ένας κωδικοποιητής, βαθύ νευρωνικό δίκτυο f, που απεικονίζει την ακολουθία εισόδου σε ένα διάνυσμα εκπαιδεύεται πρώτα χρησιμοποιώντας μια εποπτευόμενη αντιθετική συνάρτηση σφάλματος και στη συνέχεια οι διανυσματικές αναπαραστάσεις που προκύπτουν χρησιμοποιούνται για την αναγνώριση συναισθημάτων. Κάθε συνάρτηση σφάλματος σχηματίζεται με επίβλεψη που σημαίνει ότι τα θετικά δείγματα $P(a)$ είναι αυτά που ανήκουν στην ίδια κατηγορία

$$P(a) = \{p \in A | y_a = y_p, a \neq p\}$$

και τα αρνητικά $N(a)$ σε διαφορετική

$$N(a) = \{n \in A | y_a \neq y_n\}$$

όπου A είναι το σύνολο των δεικτών αγκύρωσης.

Η μέτρηση μεσολάβησης για την ποιότητα των αναπαραστάσεων επιλέγεται ως Ακρίβεια κλάσης Μη σταθμισμένη (ΥΑ). Προκειμένου να αξιολογηθεί το ΥΑ, ένας ταξινομητής g τοποθετείται πάνω από τον κωδικοποιητή f. Η σύνθεση των δύο μαθησιακών στοιχείων θα μας οδηγήσει σε έναν ταξινομητή από άκρο σε άκρο $h = g \circ f$. Ανάλογα με το πείραμα είτε τοποθετούμε ένα γραμμικό SVM πάνω από τον εκπαιδευμένο κωδικοποιητή είτε ρυθμίζουμε με ακρίβεια τον κωδικοποιητή χρησιμοποιώντας έναν εκμαθήσιμο γραμμικό ταξινομητή g (βλ. εικ. 4.3). Και στις δύο περιπτώσεις λαμβάνουμε ένα σύστημα SER από άκρο σε άκρο h.

0.1.2.1 Συνάρτηση σφάλματος τριπλετών (Triplet Loss)

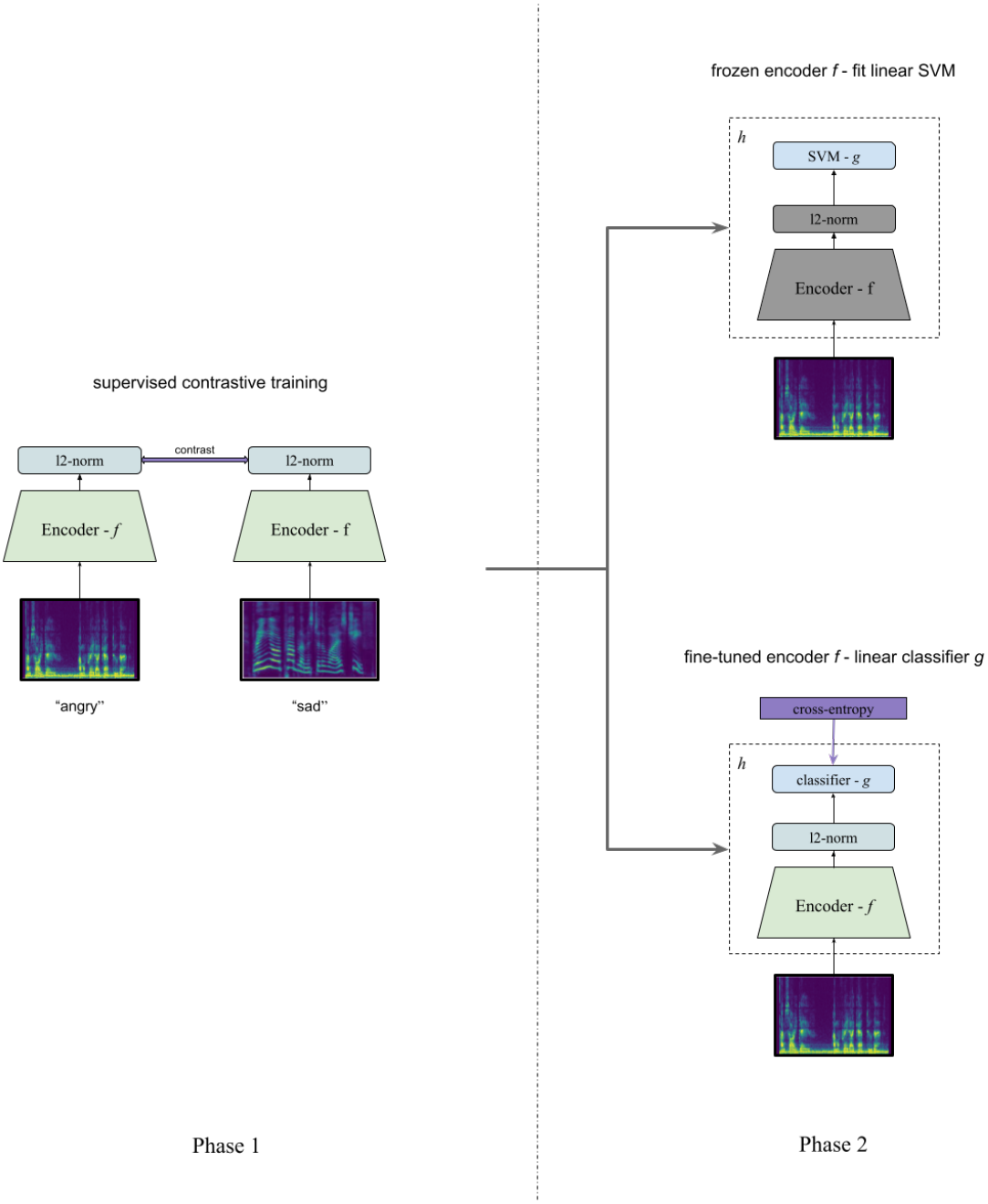
Το πρώτο πείραμα είναι να εκπαιδεύσουμε τον κωδικοποιητή f χρησιμοποιώντας την Triplet Loss με την απόσταση συνημιτόνου

$$\mathcal{L}_m^{\text{triplet}} = \frac{1}{\sum_a |\mathcal{T}_a'|} \sum_{a \epsilon A} \sum_{(p,n) \in \mathcal{T}_a} \max(0, \mathbf{z}_a \cdot \mathbf{z}_n - \mathbf{z}_a \cdot \mathbf{z}_p + m)$$

όπου $\mathcal{T}_a = P(a) \times N(a)$ είναι το σύνολο των σχηματισμένων τριπλετς με βάση την άγκυρα a και $\mathcal{T}_a' = \{(p,n) \in \mathcal{T}_a : \mathbf{z}_a \cdot \mathbf{z}_p < \mathbf{z}_a \cdot \mathbf{z}_n + m\}$ είναι το υποσύνολο των σχηματισμένων τριπλετών με βάση την άγκυρα a που συμβάλλουν στον υπολογισμό του σφάλματος.
Οι δύο πιο σημαντικές υπερπαράμετροι αυτής της μεθόδου είναι α) το περιθώριο m και β) η επιλογή των ζευγών (p,n) δίνοντας μια άγκυρα a που θα χρησιμοποιηθεί για να σχηματίσει τις τριπλέτες και να υπολογίσει το σφάλμα.

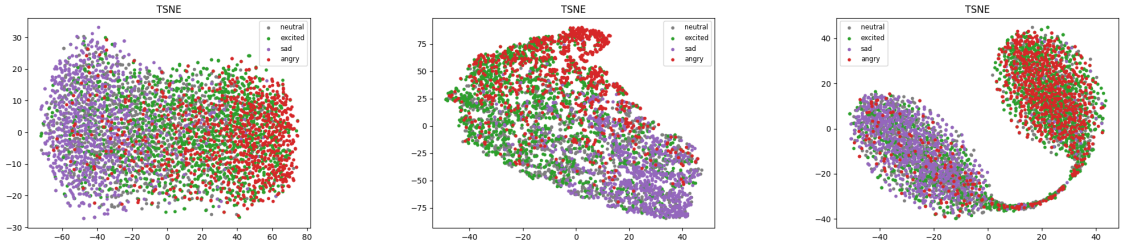
Περιθώριο

Για το πρώτο πείραμα, χρησιμοποιούνται όλες οι τριπλέτες που μπορούν να σχηματιστούν από την παρτίδα και δοκιμάζονται οι ακόλουθες 8 τιμές για το περιθώριο: $\{0,01,\ 0,02,\ 0,05,\ 0,1,\ 0,2,\ 0,5,\ 1\ ,\ 2\}$. Δεν χρειάζεται να υπερβούμε το περιθώριο 2, καθώς είναι η μέγιστη διαφορά που μπορεί να επιτευχθεί (αν και μόνο αν $\mathbf{z}_a^T \mathbf{z}_n = 1, \mathbf{z}_a^T \mathbf{z}_p = -1$) αφού τα διανύσματα αναπαράστασης είναι $l2$-κανονικοποιημένα. Μετά την εκπαίδευση των κωδικοποιητών f, ένα Σ΄Μ με γραμμικό πυρήνα εκπαιδεύεται στην κορυφή ως ταξινομητής g για να μετρήσει την ακρίβεια χρησιμοποιώντας τις μαθημένες αναπαραστάσεις. Η σύγκριση μεταξύ της μοντέλου που χρησιμοποιείται ως βάση και του καλύτερου μοντέλου που εκπαιδεύτηκε χρησιμοποιώντας triplet loss μπορεί να φανεί στον πίνακα 4.2. Το βασικό μοντέλο είναι καλύτερο σε όρους ΥΑ (53,07%), αλλά δεν απέχει πολύ από το μοντέλο αντίθεσης σε συνδυασμό με το Σ΄Μ (54,91%). Όπως μπορούμε να δούμε από τα διαγράμματα, τα καλύτερα αποτελέσματα επιτυγχάνονται χρησιμοποιώντας μια τιμή περιθωρίου μεταξύ $0,05$ και $0,5$. Αυτή η συμπεριφορά είναι αναμενόμενη επειδή το περιθώριο ελέγχει τη στεγανότητα των συστάδων. Η πολύ χαμηλή τιμή περιθωρίου αντιστοιχεί σε πολύ χαλαρες συστάδες και αντίθετα πολύ υψηλό περιθώριο σημαίνει πολύ σφιχτές συστάδες. Όπως φαίνεται από τα οικόπεδα στο σχ. 4.5, όσο μεγαλύτερο είναι το περιθώριο, τόσο πιο σφιχτές είναι οι συστάδες που σχηματίζονται.



Σχήμα 1: Η διαδικασία της εποπτευόμενης αντιθετικής εκπαίδευσης ενός κωδικοποιητή f. Αποτελείται από δύο φάσεις: εκπαιδεύστε τον κωδικοποιητή f χρησιμοποιώντας μια απώλεια αντίθεσης (αριστερά) και στη συνέχεια τοποθετήστε ένα γραμμικό SVM στην κορυφή ως ταξινομητή g (δεξιά επάνω) ή βελτιστοποιήστε με έναν ταξινομητή g (δεξιά) που μπορείτε να μάθετε κάτω)

Σχήμα 2: Επίδραση της παραμέτρου *περιθωρίου (m)* κατά την εκπαίδευση με την Triplet Loss χρησιμοποιώντας όλες της τριπλέτες της παρτίδας. Όσον αφορά την αναγνώριση, οι βέλτιστες τιμές φαίνεται να είναι στην περιοχή από $0,05$ έως $0,5$.



(α′) $m = 0,02$ (β′) $m = 0,2$ (γ′) $m = 2$

Σχήμα 3: Επίδραση του περιθωρίου στη στεγανότητα των συστάδων

Στρατηγική δειγματοληψίας

Οι τριπλέτες μπορούν να χωριστούν σε τρεις περιοχές:

1. **Easy**: Τριπλέτες που ήδη ικανοποιούν τον περιορισμό και επομένως δεν συμβάλλουν στο σφάλμα

$$\{(a, p, n) \in A \times P(a) \times N(a) \ \mid \mathbf{z}_a^T \cdot \mathbf{z}_p > \mathbf{z}_a^T \cdot \mathbf{z}_n + m\}$$

2. **Semi-Hard**: Οι τριπλέτες που *τα θετικά ζεύγη είναι ήδη πιο όμοια από τα αρνητικά με την άγκυρα αλλά όχι περισσότερο από το περιθώριο.*

$$\{(a, p, n) \in A \times P(a) \times N(a) \mid \mathbf{z}_a^T \cdot \mathbf{z}_n + m > \mathbf{z}_a^T \cdot \mathbf{z}_p > \mathbf{z}_a^T \cdot \mathbf{z}_n\}$$

3. **Hard**: Τριπλέτες που *τα αρνητικά ζεύγη μοιάζουν περισσότερο με την άγκυρα παρά τα αρνητικά*

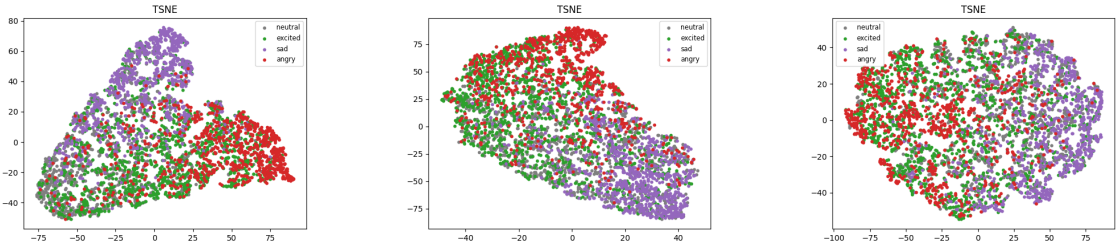
$$\{(a, p, n) \in A \times P(a) \times N(a) \mid \mathbf{z}_a^T \cdot \mathbf{z}_p < \mathbf{z}_a^T \cdot \mathbf{z}_n\}$$

Εξετάζουμε τρεις διαφορετικές στρατηγικές για την εξόρυξη τριπλετών:

1. **semi-hard**: Χρησιμοποιούνται μόνο **semi-hard** τριπλέτες
2. **hard**: Χρησιμοποιούνται μόνο **hard** τριπλέτες
3. **all**: Οι **semi-hard** και **hard** τριπλέτες συνδυάζονται

Για να εξεταστεί η επίδραση της αρνητικής δειγματοληψίας, επιλέγεται μια σταθερή τιμή για το περιθώριο, εδώ $m = 0,2$ και εφαρμόζεται κάθε μία από τις τρεις διαφορετικές στρατηγικές. Δοκιμάζεται επίσης μια μικτή στρατηγική όπου για το πρώτο 80% των εποχών το δίκτυο εκπαιδεύεται με semi-hard τριπλέτες και για το υπόλοιπο 20% του παρέχονται μόνο με hard τριπλέτες. Τα αποτελέσματα για την εργασία αναγνώρισης σε όρους ΥΑ εμφανίζονται στον πίνακα 4.3. Όπως φαίνεται από τον πίνακα, χρησιμοποιώντας μόνο σκληρές τριπλέτες το δίκτυο δεν μπορεί να συγκλίνει πιθανώς επειδή έχει κολλήσει στα τοπικά ελάχιστα. Τα ημίσκληρα τρίδυμα βελτιώνουν τη στρατηγική **all** και τα αποτελέσματα γίνονται ελαφρώς καλύτερα όταν αλλάζουν σε **hard** στις τελευταίες εποχές της εκπαίδευσης.

Σχήμα 4: Επίδραση διαφορετικών στρατηγικών εξόρυξης τριπλετών κατά την εκπαίδευση με την Triplet loss



(α′) semi-hard (β′) all (γ′) hard

Σχήμα 5: Επίδραση της στρατηγικής δειγματοληψίας τριπλετών στη στεγανότητα των συστάδων

0.1.2.2 Ομαλή συνάρτηση σφάλματος τριπλέτας

Επαναλαμβάνουμε το ίδιο πείραμα χρησιμοποιώντας την ομαλή συνάρτηση σφάλματος τριπλέτας

$$\mathcal{L}_{\tau}^{\text{τριπλετ-σμοοτη}} = \frac{1}{\sum_{a\in A}|\mathcal{T}_a|}\sum_{a\in A}\sum_{(p,n)\in\mathcal{T}_a}\log\left(1+e^{(\mathbf{z}_a\cdot\mathbf{z}_n-\mathbf{z}\cdot\mathbf{z}_p)/\tau}\right)$$

Δεδομένου ότι δεν υπάρχει σκληρό περιθώριο εδώ, σχηματίζονται όλες οι τριπλέτες της παρτίδας και η θερμοκρασία τ επιλέγεται από τις ακόλουθες τιμές $\{0,01,\ 0,02,\ 0,05,\ 0,1,\ 0,2,\ 0,5,\ 1,\ 2,\ 5,\ 10\}$. Τα αποτελέσματα ομαδοποιούνται και παρουσιάζονται μαζί με την επόμενη ενότητα.

0.1.2.3 Πολλαπλά αρνητικά και/ή θετικά ανά άγκυρα

Σε αυτό το σύνολο πειραμάτων, συγκρίνουμε και αντιπαραβάλλουμε κάθε δείγμα άγκυρα με περισσότερα από ένα θετικά ή/και αρνητικά δείγματα. Αναμένουμε ότι το δίκτυο κωδικοποιητή θα επωφεληθεί από την αύξηση του αριθμού των αντιθετικών δειγμάτων και θα μάθει μια καλύτερη δομή του χώρου αναπαράστασης. Για να το κάνουμε αυτό, χρησιμοποιούμε τις συναρτήσεις σφάλματος NT-Xent [1] και SupCon [2]. Εδώ, δοκιμάζουμε την NT-Xent loss [1] με εποπτευόμενο τρόπο, δηλαδή χρησιμοποιώντας ως αρνητικά δείγματα, τα δείγματα της παρτίδας που ανήκουν σε διαφορετική κατηγορία από το ζεύγος anchor-positive

$$\mathcal{L}_{\tau}^{\text{NT-Ξεντ}} = \frac{1}{|A|}\sum_{a\in A}\frac{1}{|P(a)|}\sum_{p\in P(a)}-\log\left(\frac{e^{\mathbf{z}_a\cdot\mathbf{z}_p/\tau}}{e^{\mathbf{z}_a\cdot\mathbf{z}_p/\tau}+\sum_{n\in N(a)}e^{\mathbf{z}_a\cdot\mathbf{z}_n/\tau}}\right)$$

Εφόσον δεν χρησιμοποιούμε επαυξήσεις, η απώλεια είναι πρακτικά η ίδια με την απώλεια N-Pair [3] αλλά χωρίς να περιορίζεται η παρτίδα ώστε να περιέχει μόνο ένα δείγμα ανά κατηγορία συναισθημάτων. Ο λόγος για τον περιορισμό της παρτίδας ώστε να περιέχει μόνο ένα δείγμα ανά κλάση ήταν ότι ο αριθμός των κλάσεων ήταν μεγάλος και επομένως ήταν υπολογιστικά απαγορευτικός για να σχηματιστεί η παρτίδα. Πειραματιζόμαστε με τις ίδιες τιμές τ με τις προηγούμενες $\{0,01,\ 0,02,\ 0,05,\ 0,1,\ 0,2,\ 0,5,\ 1,\ 2,\ 5,\ 10\}$ και αξιολογούμε το παραστάσεις με τον ίδιο τρόπο.

Σχήμα 6: Επίδραση της παραμέτρου θερμοκρασίας (τ) για κάθε συνάρτηση σφάλματος

Πειραματιζόμαστε επίσης με την απώλεια εποπτευόμενης αντίθεσης (SupCon) που παρουσιάζεται στο [2] όπου χρησιμοποιούνται πολλαπλά αρνητικά και πολλαπλά θετικά δείγματα για τον υπολογισμό της απώλειας και η άθροιση εκτελείται εκτός του λογαρίθμου

$$\mathcal{L}_{out}^{\text{συπ}} = \sum_{a \in A} \frac{1}{|P(a)|} \sum_{p \in P(a)} -\log\left(\frac{e^{\mathbf{z}_a \cdot \mathbf{z}_p/\tau}}{\sum_{p \in P(a)} e^{\mathbf{z}_a \cdot \mathbf{z}_p/\tau} + \sum_{n \in N(a)} e^{\mathbf{z}_a \ cdot \mathbf{z}_n/\tau}}\right)$$

Σημειώστε ότι σε αυτό το πείραμα δεν χρησιμοποιούνται επαυξήσεις όπως στο αρχικό και ως εκ τούτου δεν σχηματίζεται παρτίδα ¨πολλαπλής προβολής¨. Αυξάνουμε μόνο τον αριθμό των θετικών που απαιτούνται για τον υπολογισμό του αριθμητή. Οι τιμές της θερμοκρασίας τ που χρησιμοποιούνται είναι όπως οι προηγούμενες $\{0,01,\ 0,02,\ 0,05,\ 0,1,\ 0,2,\ 0,5,\ 1,\ 2,\ 5,\ 10\}$ και το οι αναπαραστάσεις αξιολογούνται με τον ίδιο τρόπο.

Εξετάζουμε την επίδραση της παραμέτρου θερμοκρασίας τ για όλες τις απώλειες αντίθεσης (Smooth Triplet, NT-Xent, SupCon) σε μια κοινή γραφική παράσταση (σχήμα 4.8).

Τα καλύτερα μοντέλα από άποψη UA για κάθε συνάρτηση σφάλματος αντίθεσης παρουσιάζονται επίσης στον πίνακα 4.4

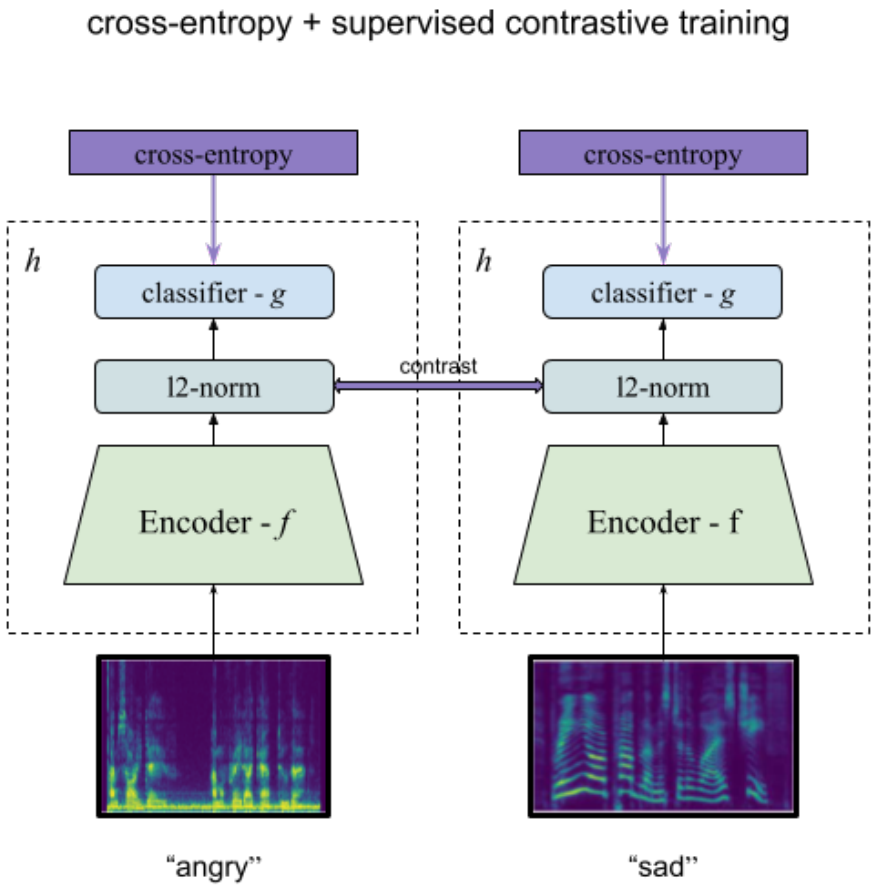
Συνεχίζουμε με τη μικρο-ρύθμιση ενός γραμμικού ταξινομητή g πάνω από τον κωδικοποιητή f. Όλο το δίκτυο εκπαιδεύεται για 20 εποχές. Τις πρώτες 10 εποχές ο κωδικοποιητής f διατηρείται παγωμένος και στη συνέχεια ολόκληρο το δίκτυο εκπαιδεύεται μαζί με τρόπο από άκρο σε άκρο με Cross Entropy. Ξεκινάμε με χαμηλότερο ρυθμό μάθησης $0,0001$ και οι υπόλοιπες παράμετροι βελτιστοποίησης είναι ίδιες με τη βασική μας μέθοδο. Αντί για το γραμμικό SVM που χρησιμοποιήθηκε πριν για την αξιολόγηση των αναπαραστάσεων, τώρα μετράμε το UA χρησιμοποιώντας τον γραμμικό ταξινομητή g που χρησιμοποιήθηκε για τη μικρο-ρύθμιση του κωδικοποιητή f. Τα αποτελέσματα εμφανίζονται στο 4.5

0.1.2.4 Θετικά δείγματα από επαύξηση

Όπως και στην αρχική πρόταση του [2], σχηματίζεται μια ¨παρτίδα πολλαπλών προβολών¨. Για ένα σύνολο N τυχαίων ζευγών δείγματος/ετικέτας, $A = \{(x_k, y_k), k = 1...N\}$, η αντίστοιχη παρτίδα που χρησιμοποιείται για την εκπαίδευση αποτελείται από $2N$ ζεύγη, $\tilde{A} = \{(\tilde{x}_i, \tilde{y}_i), i = 1...2N\}$, όπου x_{2k} και Οι x_{2k-1} είναι δύο τυχαίες αυξήσεις (γνωστές και ως ¨προβολές¨) των x_k με $(k = 1...N)$ και $y_{2k} = y_{2k-1} = y_k$. Το ίδιο σκεύασμα χρησιμοποιείται για τον υπολογισμό της συνάρτησης σφάλματος, αλλά χρησιμοποιώντας επαυξημένα δείγματα από την ¨παρτίδα πολλαπλής προβολής'

$$\mathcal{L}_{out}^{\text{συπ}} = \sum_{a \in \tilde{A}} \frac{1}{|P(a)|} \sum_{p \in P(a)} -\log\left(\frac{e^{\mathbf{z}_a \cdot \mathbf{z}_p/\tau}}{\sum_{p \in P(a)} e^{\mathbf{z}_a \cdot \mathbf{z}_p/\tau} + \sum_{n \in N(a)} e^{\mathbf{z}_a \cdot \mathbf{z}_n/\tau}}\right)$$

Οι επαυξήσεις που χρησιμοποιούνται σε αυτό το πείραμα είναι η κάλυψη διαδοχικών κομματιών στο χρόνο ή/και στη συχνότητα ακολουθώντας την εργασία του [4]. Για χρονική κάλυψη, ένα διαδοχικό κομμάτι έως και 5% του συνολικού χρονικού μήκους του φασματογράμματος μηδενίζεται. Για την απόκρυψη συχνότητας, ένα διαδοχικό κομμάτι έως και 10 συχνότητες μηδενίζεται.



Σχήμα 7: Η διαδικασία της εποπτευόμενης εκπαίδευσης αντίθεσης ενός κωδικοποιητή f μαζί με απώλεια διασταυρούμενης εντροπίας που εφαρμόζεται σε ολόκληρο το δίκτυο h.

Προκειμένου να παρέχονται δίκαιες συγκρίσεις, εκπαιδεύεται και αξιολογείται μια έκδοση βασικού μοντέλου που χρησιμοποιεί επαυξήσεις. Οι αναπαραστάσεις του κωδικοποιητή που έχει εκπαιδευτεί με ¨παρτίδες πολλαπλών προβολών' αξιολογούνται όπως πριν. Ένα γραμμικό SVM τοποθετείται πάνω από τον κωδικοποιητή και ένας γραμμικός ταξινομητής είναι ρυθμισμένος με ακρίβεια μαζί με τον κωδικοποιητή. Η λεπτομέρεια εκτελείται χρησιμοποιώντας τα καθαρά δεδομένα και τα επαυξημένα δεδομένα. Τα αποτελέσματα εμφανίζονται στον πίνακα 4.7. Όπως φαίνεται από τον πίνακα, τα καλύτερα αποτελέσματα δόθηκαν με προ-προπόνηση με εποπτευόμενη απώλεια αντίθεσης χρησιμοποιώντας παρτίδες πολλαπλής προβολής και λεπτομέρεια με διασταυρούμενη εντροπία χρησιμοποιώντας αυξήσεις δεδομένων. Ωστόσο, το UA απέχει πολύ από το βασικό μοντέλο με επαυξήσεις.

0.1.3 Συνδυασμός διασταυρούμενης εντροπίας και εποπτευόμενων σημάτων αντίθεσης

Δεδομένου ότι υπάρχει μια βελτίωση κατά τη λεπτομερή ρύθμιση του κωδικοποιητή όταν εκπαιδεύουμε έναν γραμμικό ταξινομητή πάνω του, προσπαθούμε να εφαρμόσουμε και

τις δύο απώλειες (επιβλεπόμενη αντίθεση και διασταυρούμενη εντροπία) ταυτόχρονα (βλ. εικόνα 4.9). Αναμένουμε ότι αυτή η στρατηγική θα ενθαρρύνει τον κωδικοποιητή να μάθει μια δομή του χώρου και θα βοηθήσει τον γραμμικό ταξινομητή να επιτύχει καλύτερη γενίκευση. Για το ακόλουθο σύνολο πειραμάτων, η εποπτευόμενη απώλεια αντίθεσης $\mathcal{L}_{sc}$ χρησιμοποιείται ως βοηθητική απώλεια στην κύρια απώλεια διασταυρούμενης εντροπίας $\mathcal{L}_{ce}$ σταθμισμένη με τιμή $\lambda > 0$

$$\mathcal{L} = \mathcal{L}_{ce} + \lambda \mathcal{L}_{sc}$$

Η επιλεγμένη εποπτευόμενη απώλεια αντίθεσης είναι η ίδια όπως πριν, δηλαδή η απώλεια Συπὸν με $\tau = 0,1$. Ο ρυθμός μάθησης ξεκινά από μια χαμηλότερη τιμή $0,0001$. Το δίκτυο εκπαιδεύεται χρησιμοποιώντας επαυξήσεις με παρτίδες ¨πολλαπλής προβολής' όπως περιγράφηκε προηγουμένως. Το αρχικό μέγεθος παρτίδας έχει οριστεί σε 32, επομένως το πραγματικό μέγεθος παρτίδας είναι 64. Όλες οι άλλες παράμετροι της διαδικασίας βελτιστοποίησης παραμένουν ίδιες με το βασικό μοντέλο. Τα αποτελέσματα εμφανίζονται στον πίνακα 4.8. Η καλύτερη απόδοση επιτυγχάνεται όταν οι απώλειες σταθμίζονται εξίσου ($\lambda = 1$), αλλά εξακολουθεί να μην φτάνει το βασικο μοντέλο.

0.1.4 Αυτοεποπτευόμενη αντιθετική (προ-)εκπαίδευση

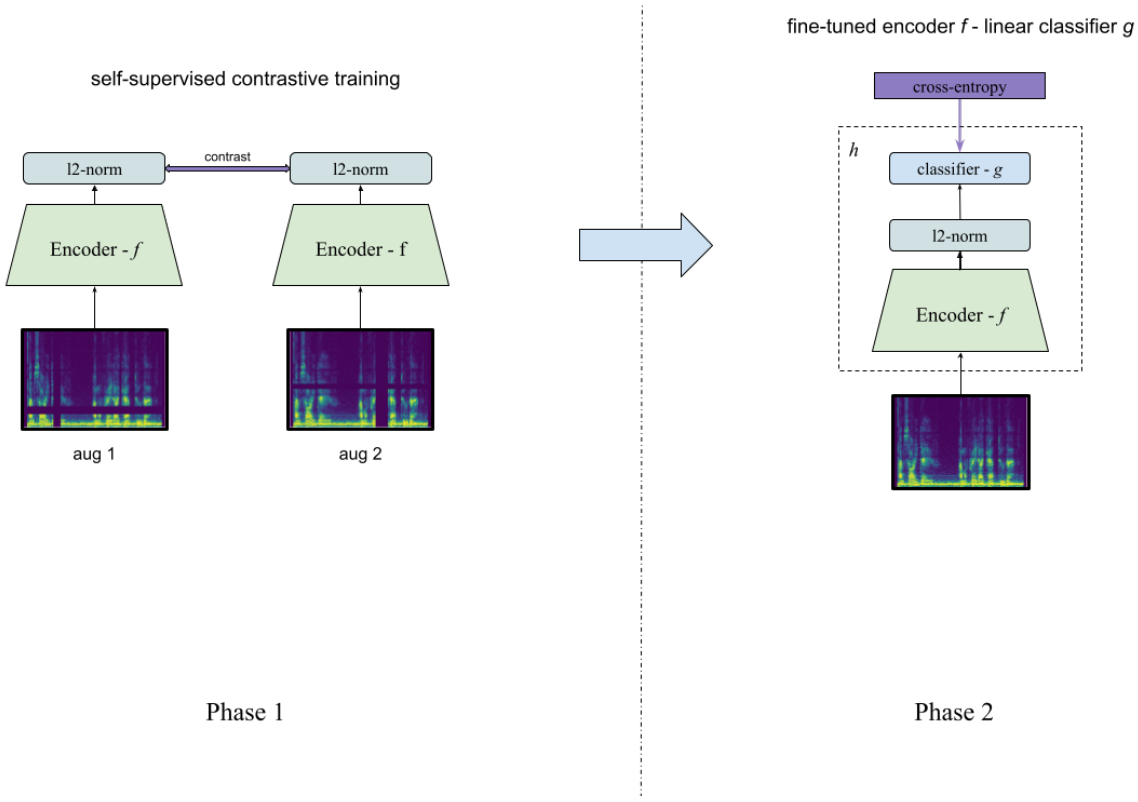
Προχωράμε με την προσθήκη μιας μορφής αυτο-επίβλεψης στη φάση της εκπαίδευσης ως προεκπαιδευτικό βήμα. Η αυτοεποπτευόμενη προ-προπονητική διαδικασία καθοδηγείται επίσης από απώλεια αντίθεσης. Σε αντίθεση με τις προηγούμενες ενότητες, σχηματίζουμε και υπολογίζουμε την απώλεια χωρίς τη χρήση των ετικετών. Ακολουθούμε ακριβώς την ίδια διαδικασία όπως στο αρχικό χαρτί του NT-Xent loss [1] σχηματίζοντας παρτίδες ¨πολλαπλής προβολής' και απορρίπτοντας τις ετικέτες. Συντονίζουμε πρώτα τη θερμοκρασία εκπαιδεύοντας τον κωδικοποιητή μόνο με αυτο-επίβλεψη και στη συνέχεια τοποθετούμε ένα γραμμικό SVM πάνω από τις μαθημένες αναπαραστάσεις. Επιλέγεται η θερμοκρασία που δίνει την καλύτερη απόδοση από άποψη UA. Δεν υπήρχαν μεγάλες αποκλίσεις μεταξύ των διαφόρων τιμών της θερμοκρασίας, έτσι για τα υπόλοιπα πειράματα ορίσαμε τη θερμοκρασία της αυτοεπιβλεπόμενης απώλειας NT-Xent σε $\tau = 0,5$. Προκειμένου να αξιολογήσουμε περαιτέρω την πιθανή προστιθέμενη αξία της αυτο-εποπτευόμενης προεκπαίδευσης, προχωράμε σε μικροσυντονισμό του κωδικοποιητή με εποπτευόμενα σήματα.

Οι μέθοδοι μικρορύθμισης που δοκιμάζουμε είναι οι επόμενες:

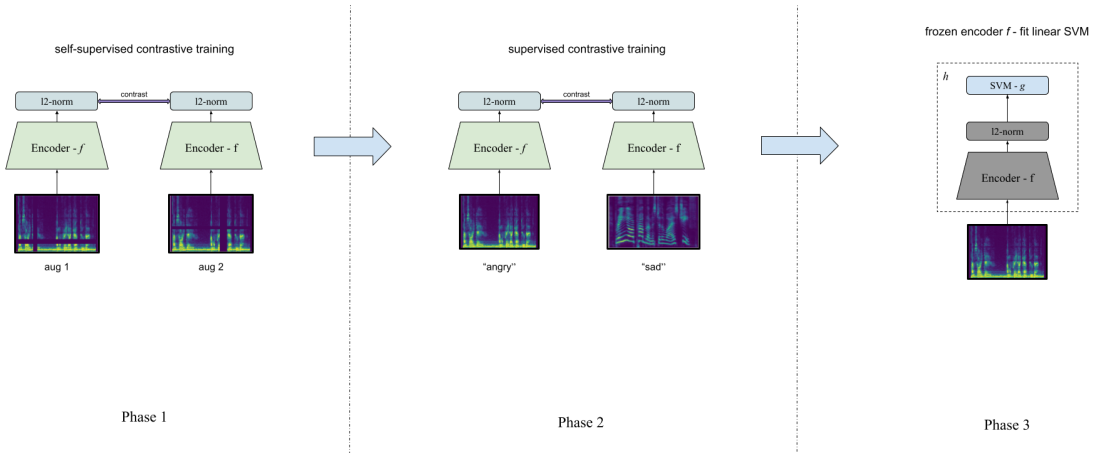
1. ce (ft-aug): Βελτιστοποιήστε έναν ταξινομητή από άκρο σε άκρο h προσθέτοντας μια γραμμική κεφαλή g πάνω από το εκμαθημένο δίκτυο κωδικοποιητών διατηρώντας τις επαυξήσεις χρησιμοποιώντας μόνο την απώλεια ῞ροσς Εντροπψ $\mathcal{L}_{ce}$ (βλ. εικόνα 4.12)

2. scl (ft-multi-aug) → svm: Ρυθμίστε με ακρίβεια τον κωδικοποιητή f χρησιμοποιώντας μια εποπτευόμενη απώλεια αντίθεσης $\mathcal{L}_{sc}$ και στη συνέχεια τοποθετήστε ένα SVM πάνω από τον κωδικοποιητή (βλ. εικόνα 4.11)

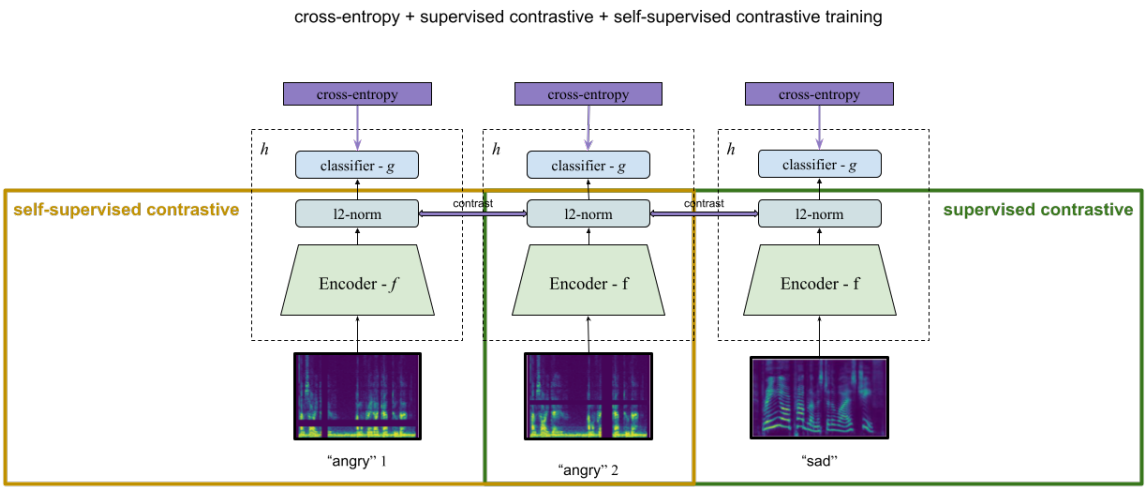
Τα αποτελέσματα εμφανίζονται στον πίνακα 4.9. Μπορεί να φανεί ότι η ύπαρξη ενός κωδικοποιητή προεκπαιδευμένου με αυτο-επίβλεψη και άμεσης λεπτομέρειας με διασταυρούμενη εντροπία από άκρο σε άκρο με αυξήσεις οδηγεί στα καλύτερα αποτελέσματα που βελτιώνονται στη βασική γραμμή +1% όσον αφορά το UA .



Σχήμα 8: Ακολούθησε η διαδικασία αυτοεποπτευόμενης εκπαίδευσης αντίθεσης ενός κωδικοποιητή f με λεπτομέρεια ολόκληρου του δικτύου h χρησιμοποιώντας απώλεια διασταυρούμενης εντροπίας



Σχήμα 9: Η διαδικασία της αυτοεποπτευόμενης εκπαίδευσης αντίθεσης ενός κωδικοποιητή f ακολούθησε με λεπτομέρεια του κωδικοποιητή f με εποπτευόμενη απώλεια αντίθεσης και στη συνέχεια τοποθέτηση ενός γραμμικού ΣΜ g από πάνω



Σχήμα 10: Η διαδικασία συνδυασμένης αυτοεποπτευόμενης και εποπτευόμενης εκπαίδευσης αντίθεσης ενός κωδικοποιητή f μαζί με απώλεια διασταυρούμενης εντροπίας που εφαρμόζεται σε ολόκληρο το δίκτυο h

0.1.5 Συνδυασμός εποπτευόμενων και αυτοεποπτευόμενων σημάτων

Αντί να χωρίσουμε τη διαδικασία σε 2 βήματα (αυτοεποπτευόμενη προεκπαίδευση και εποπτευόμενη μικρορύθμιση), προσπαθούμε να τα συνδυάσουμε σε μια ενιαία διαδικασία εκπαίδευσης. Αναμένουμε ότι ο συνδυασμός των εποπτευόμενων και των αυτοεποπτευόμενων απωλειών θα ωθήσει το δίκτυο για την εκμάθηση διακριτικών χαρακτηριστικών και αμετάβλητης κατηγορίας. Η συνδυασμένη απώλεια είναι η ίδια με το 4.1 με μια σταθμισμένη προσθήκη μιας αυτοεποπτευόμενης απώλειας $\mathcal{L}_{ssc}$

$$\mathcal{L} = \mathcal{L}_{ce} + \lambda \mathcal{L}_{sc} + \beta \mathcal{L}_{ssc}$$

Η υπέρ-παράμετρος λ έχει καθοριστεί σε 1 που είναι η τιμή που έδωσε τα καλύτερα αποτελέσματα στο 4.4.3. Συντονίζουμε την υπερπαράμετρο β επιλέγοντας την τιμή μεταξύ $\{0.1, 0.5, 1\}$. Τα αποτελέσματα φαίνονται στον πίνακα. Όπως μπορούμε να δούμε από τα αποτελέσματα, δεν υπάρχει κανένα όφελος από την ανάμειξη αυτοεπιβλεπόμενων και εποπτευόμενων σημάτων στην ίδια απώλεια.

0.1.6 Σύγκριση με άλλα έργα

Η εκτίμηση απόδοσης ενός συστήματος SER μπορεί να γίνει τόσο με τρόπο εξαρτώμενο από τον ομιλητή (Speaker Dependent - SD) όσο και με τρόπο ανεξάρτητο από τον ομιλητή (Speaker Independent - SI). Η αξιολόγηση εξαρτώμενη από ομιλητή σημαίνει ότι τα σήματα ομιλίας που χρησιμοποιούνται για τη δοκιμή έχουν παραχθεί από ομιλητές που έχουν επίσης συμμετάσχει στη διαδικασία εκπαίδευσης. Η ανεξάρτητη αξιολόγηση ομιλητών σημαίνει ότι δεν έχουν παρασχεθεί πληροφορίες στο σύστημα κατά τη διάρκεια της φάσης εκπαίδευσής του για τους ομιλητές που χρησιμοποιούνται για τη δοκιμή. Δεδομένου ότι η προσέγγιση της ανάλυσης που εξαρτάται από ομιλητή είναι η λιγότερο συχνή σε πραγματικές περιπτώσεις (πιθανότατα δεν θα έχετε καμία προηγούμενη ενημέρωση για τους

ομιλητές), αξιολογούμε το σύστημά μας σε έναν ανεξάρτητο ομιλητή. Στη βιβλιογραφία, υπάρχουν δύο προσεγγίσεις για την ανεξάρτητη αξιολόγηση του ομιλητή του συνόλου δεδομένων IEMOCAP. Η πρώτη προσέγγιση είναι μια στρατηγική διασταυρούμενης επικύρωσης 5 φορές που ονομάζεται Leave-One-Session-Out και η δεύτερη προσέγγιση είναι μια δεκαπλάσια στρατηγική αλληλοεπικύρωσης που ονομάζεται Leave-One-Speaker-Out. Αποφασίσαμε να ακολουθήσουμε τη δεύτερη προσέγγιση ανεξάρτητου ομιλητή (Λεαε-Ονε-Σπεακερ-Ουτ). Στον πίνακα 4.11, υπάρχει σύγκριση με άλλες υλοποιήσεις συστημάτων SER.

Συνολικά, τα αποτελέσματα αυτής της εργασίας υποδηλώνουν ότι η αυτο-εποπτευόμενη προ-εκπαίδευση και η εποπτευόμενη μικρορύθμιση με διασταυρούμενη εντροπία είναι μια πολλά υποσχόμενη προσέγγιση για το SER και ότι η χρήση ενός μεγαλύτερου συνόλου δεδομένων για προεκπαίδευση μπορεί να βελτιώσει περαιτέρω την απόδοση. Απαιτείται περαιτέρω έρευνα για την πλήρη κατανόηση των πλεονεκτημάτων και των περιορισμών της αντιθετικής μάθησης για το SER.

Acknowledgements

First of all, I would like to thank my supervisor professor Alexandros Potamianos who embraced my as a member of his lab team from the first day and gave me invaluable advice which played a crucial role for completing my thesis. Special thanks to Efthymis Georgiou and Giorgos Paraskevopoulos who provided me with ideas and guidance through the way as well as the for all the fun we had. I am also feeling grateful to professor Dimitris Fotakis who explicitly expressed his belief in my potential and pushed me to work at the best of my capabilities. During the time of working on my Master's Thesis, I would like to thank everyone who showed me the unconditional love and support I needed.

Speech Emotion Recognition using Contrastive Learning

Petros Iatropoulos
iatrop.petros@gmail.com

January 19, 2023

Contents

Chapter 1

Introduction

As an introduction to our work, we begin by providing some general information about the mechanics of speech and about the expression and perception of emotions.

1.1 Speech production

The process of speech production in humans involves the coordination of several different systems in the body, including the respiratory system, the phonatory system, and the articulatory system. The respiratory system supplies the air that is necessary for speech, the phonatory system produces sound waves by vibrating the vocal folds in the larynx, and the articulatory system shapes the sound waves into recognizable speech sounds by moving the lips, tongue, and other articulators.

The process of speech production begins with the respiratory system, which supplies the air needed for speech. When we inhale, air is drawn into the lungs and then passes through the trachea, or windpipe. From the trachea, the air moves into the larynx, which is located at the top of the trachea. The larynx contains the vocal folds, which are two bands of smooth muscle tissue that vibrate to produce sound.

When we exhale, air from the lungs passes through the vocal folds, causing them to vibrate. The vibrations of the vocal folds produce sound waves, which are then shaped into recognizable speech sounds by the articulatory system. The articulatory system includes the lips, tongue, and other muscles in the mouth and throat, which move to produce different speech sounds. For example, the position of the lips and tongue can be varied to produce sounds like "m," "b," and "v," while the shape of the mouth can be changed to produce sounds like "o" and "u."

By coordinating the respiratory, phonatory, and articulatory systems, we are able to produce a wide range of speech sounds, which can be combined to form words and sentences.

The process of speech production is a complex and coordinated effort that involves several different systems in the body. While the respiratory system supplies the air needed for speech, the phonatory system is responsible for producing sound waves by vibrating the vocal folds in the larynx. The articulatory system then shapes these sound waves into recognizable speech sounds by moving the lips, tongue, and other

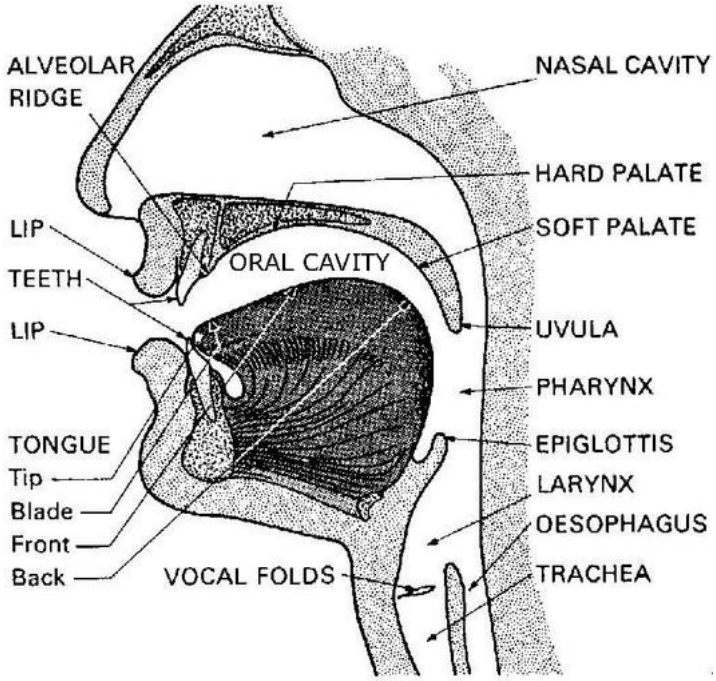


Figure 1.1: Speech production system

articulators.

The movements of the articulators are controlled by the motor cortex, a region of the brain that is responsible for controlling voluntary movements. The motor cortex sends signals to the muscles in the mouth and throat, instructing them to move in specific ways to produce the desired speech sounds. In addition to the motor cortex, other brain regions are also involved in speech production. For example, the Broca's area and the Wernicke's area are two important language centers in the brain that are involved in speech production and comprehension. The Broca's area, located in the frontal lobe, is important for the production of speech, while the Wernicke's area, located in the temporal lobe, is important for the comprehension of speech.

The process of speech production is also influenced by cognitive processes in the brain, such as language comprehension and production. These cognitive processes help us to understand what we want to say and to select the appropriate words and sounds to use in our speech.

Overall, speech production is a highly complex process that involves the coordination of several different systems in the body, including the respiratory, phonatory, and articulatory systems, as well as cognitive processes in the brain.

1.2 Emotion expression and perception

Speech emotion production refers to the way in which emotions are conveyed through speech. This can include factors such as tone of voice, pitch, and tempo.

Perception of speech emotion refers to the ability of a listener to understand and interpret the emotions conveyed through speech. This is a complex process that involves both verbal and non-verbal cues, and can be influenced by a variety of factors including

Figure 1.2: Emotion wheel from Robert Plutchik in 1980 [5]

the listener's emotional state, the context of the conversation, and the relationship between the speaker and listener.

When we produce speech, the way we convey emotions is influenced by our physiological arousal. For example, if we are feeling angry, we might speak in a louder tone with more force, or if we are feeling sad, our voice might become softer and lower pitched.

Plutchik's wheel of emotions is a model of emotion that was proposed by psychologist Robert Plutchik [5]. It is based on the idea that there are eight primary emotions, which Plutchik called "primary affective states": joy, trust, fear, surprise, sadness, disgust, anger, and anticipation. These primary emotions are arranged in a circular diagram 1.2, with each emotion located at a specific point on the circle. According to Plutchik, the primary emotions can combine in various ways to produce more complex and nuanced emotions, such as love, pride, and shame. The wheel of emotions is a useful tool for understanding the relationships between different emotions and how they can be combined to produce a wide range of emotional experiences.

The valence-arousal model of emotion was proposed by psychologist Russell (1980) in his "circumplex model of affect". This model suggests that emotions can be represented in terms of two orthogonal dimensions: valence and arousal. Valence refers to the positivity or negativity of an emotion, while arousal refers to its intensity or level of activation. According to this model, emotions can be represented as points in a circular space, with the two dimensions forming the x and y axes. For example, happiness would be represented by a high-valence, low-arousal point in the top-right quadrant of the circle, while fear would be represented by a low-valence, high-arousal point in the bottom-left quadrant. The 2d model of Russel is shown in figure 1.3 as it was originally described in his paper.

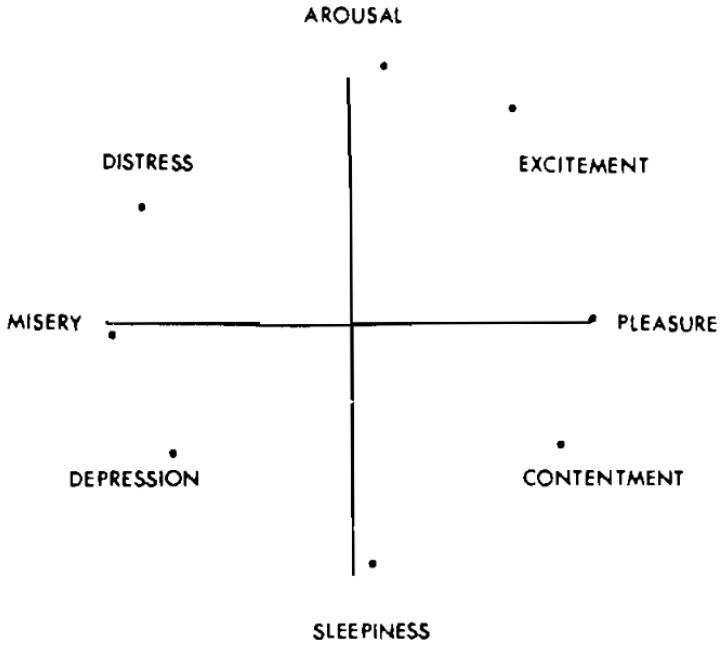


Figure 1.3: The two dimensional model of valence and arousal as it was depicted in the original work of Russell in 1980 [6]

In addition to the physiological arousal, our emotional state also influences the words we choose and the way we structure our sentences. For example, if we are feeling happy, we might use more positive words and have a more upbeat cadence to our speech.

Perception of speech emotion involves both verbal and non-verbal cues. Verbal cues include the words we use and the way we structure our sentences, while non-verbal cues include our tone of voice, pitch, tempo, and facial expressions.

The ability to perceive speech emotion can vary depending on the listener's emotional state, the context of the conversation, and the relationship between the speaker and listener. For example, if the listener is in a positive emotional state, they may be more likely to perceive positive emotions in the speaker's speech.

Research has shown that the ability to perceive speech emotion is important for effective communication and social interactions. It allows us to understand the emotional state of the speaker and respond in a way that is appropriate and empathetic.

Overall, speech emotion production and perception are complex processes that play a crucial role in human communication and social interactions. Understanding these processes can help us improve our ability to communicate effectively and build stronger relationships with others.

1.3 Speech emotion recognition

Speech emotion recognition (SER) is a task in the field of artificial intelligence that involves using machine learning algorithms to analyze speech signals and predict the emotions that a person may be expressing. The goal of this task is to develop systems

that can automatically recognize and interpret the emotional content of a person's speech, which can be useful in a variety of applications, such as virtual assistants, customer service chatbots, or even in psychological therapy.

The domain of speech emotion recognition includes a number of different subfields, such as natural language processing, speech processing, and machine learning. To perform this task, researchers often use a combination of techniques from these fields, such as feature extraction, acoustic modeling, and classification algorithms, to build systems that can accurately recognize and interpret emotions in speech. These systems typically rely on large datasets of labeled speech samples to train and evaluate their performance, and the quality and diversity of these datasets can have a significant impact on the accuracy of the resulting models.

Chapter 2

Technical Background

The following section includes some general technical background that the current work is based onto.

2.1 Speech signal digital representation

In digitization, the analog speech signal is first converted into a discrete-time signal by sampling it at regular intervals. The samples are then quantized, which involves mapping the amplitude of each sample to a digital value from a finite set of possible values. The digital values are typically represented using a binary code, where each value is represented by a sequence of binary digits (bits).

The sampling rate and the number of bits used for quantization determine the quality of the digital representation. In general, a higher sampling rate and a larger number of bits per sample result in a digital signal that more closely resembles the original analog signal. However, this also increases the amount of data that needs to be processed, transmitted, and stored, so there is a trade-off between the quality and efficiency of the digital representation.

Once the analog speech signal has been converted into a digital signal, it can be easily processed, transmitted, and stored using digital devices and systems. This allows for a wide range of applications, such as digital signal processing, speech recognition, and telecommunication. In addition, the digital representation of speech signals enables the use of error-correcting codes, which can help to reduce the impact of noise and other sources of distortion on the signal.

2.1.1 Time domain representation

In time-domain digital representation of a speech signal, the speech signal is represented as a sequence of digital samples in time. This means that the amplitude of the speech signal is sampled at regular intervals, and the samples are quantized and encoded using a binary code. The resulting sequence of digital values can be thought of as a series of snapshots of the speech signal at different points in time.

One advantage of time-domain representation is that it directly reflects the temporal structure of the speech signal. This makes it easy to analyze the signal in terms of

its temporal characteristics, such as its duration, periodicity, and pitch. In addition, time-domain representation is often used in speech processing applications that involve time-domain processing, such as filtering and time-varying signal manipulation.

However, time-domain representation has some limitations. For example, it can be difficult to represent certain types of signals, such as signals with wide frequency spectra, using a time-domain representation with a limited sampling rate. In addition, time-domain representation does not directly reflect the frequency content of the signal, so it is not well-suited for applications that involve frequency-domain processing, such as spectral analysis and frequency-domain filtering.

2.1.2 Frequency domain representation

In frequency-domain digital representation of a speech signal, the speech signal is represented in terms of its frequency content. This is typically done using a process called Fourier transformation, which decomposes the signal into a set of sinusoidal signals with different frequencies, amplitudes, and phases. The resulting frequency-domain representation of the speech signal is often called the spectrum of the signal.

One advantage of frequency-domain representation is that it directly reflects the frequency content of the speech signal. This makes it easy to analyze the signal in terms of its spectral characteristics, such as its spectral shape and the distribution of energy across different frequency bands. In addition, frequency-domain representation is often used in speech processing applications that involve frequency-domain processing, such as spectral analysis and frequency-domain filtering.

However, frequency-domain representation has some limitations. For example, it can be difficult to represent certain types of signals, such as signals with rapid temporal changes, using a frequency-domain representation with a limited frequency resolution. In addition, frequency-domain representation does not directly reflect the temporal structure of the signal, so it is not well-suited for applications that involve time-domain processing, such as filtering and time-varying signal manipulation.

2.2 Machine Learning

2.2.1 General

What is Machine Learning (ML)? We could describe ML as the collection of algorithms that are able to learn from the data. But what do we mean learn from the data? A definition is given in [7]:
"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."
There are too many aspects to cover on all three (experience E, task T, performance P) but in this short section we will provide the basic and necessary background theory included in the current work.

2.2.1.1 Task T

Machine learning algorithms after their execution produce models that solve the task at hand. In the following section we describe some of the tasks.

Classification

For this task, we want the model that has been produced by the algorithm to be able to specify into which category / class, from a total of C classes, a sample input belongs to. In order to do this, the produced model should be a function of the form $f : \mathcal{X} \to \{1, 2, \dots C\}$ where $\mathcal{X}$ is the input domain. A usual variant of the classification task is to output a function that emits a probability vector equal to the number of class, $f : \mathcal{X} \to \mathbb{R}^C$.

Clustering

Clustering is the task of grouping similar samples into the same group while dissimilar ones are assigned into different groups.

Retrieval

Given a query sample, retrieve its most similar one that has already been processed

2.2.1.2 Experience E

Depending on the kind of the experience an algorithm is allowed to have during training, they are separated into two categories, **supervised** and **unsupervised**. Although, the line between them is sometimes blurry, we will try to give some insights into each category.

Unsupervised

The algorithms that fall into this category are the ones that attempt to learn the underlying structure of the given data. The most common purpose is the estimation of the probability distribution that generated the data at hand usually through density estimation. Clustering of the data is also a task that is usually learned without supervision. Unsupervised learning involves observing several examples of a random vector $\mathbf{x}$ and attempting to implicitly or explicitly learn the probability distribution $p(\mathbf{x})$

Supervised

The supervised algorithms are provided with extra information (usually called **label** or **target**) about the task they need to solve. Classification and Regression are most of the times considered supervised tasks. Supervised learning involves observing several examples of a random vector $\mathbf{x}$ and an associated value or vector $\mathbf{y}$, then learning to predict $\mathbf{y}$ from $\mathbf{x}$, usually by estimating $\mathrm{p}(\mathbf{y} \mid \mathbf{x})$. The term supervised learning originates from the view of the target $\mathbf{y}$ being provided by an instructor or teacher who shows the machine learning system what to do.

2.2.2 Deep Learning and Neural Networks

What separates the traditional Machine Learning methods from Deep Learning is the use of very 'deep' neural networks (i.e. neural networks with many layers) that are optimized using parallelized algorithms mainly on GPUs or TPUs. Deep neural networks consist of building blocks that sequentially transform the data. These functions are learnable layers of operations that are composed to form a network. Depending on the structure of the data and the task at hand there have been proposed different layer architectures and some of them are presented in the following sections.

2.2.2.1 Convolution

The convolution operation of two one-dimentional continuous functions $x, w : \mathbb{R} \to \mathbb{R}$ is defined as

$$s(t) = (x * w)(t) = \int x(a)w(t-a)\,da$$

For discrete one-dimentional functions $\mathbb{N} \to \mathbb{R}$ the integral is substituted with a summation

$$s(n) = (x * w)(n) = \sum_m x(m)w(n-m)$$

Convolutional layers implement this operation between the input signal x and a set of learnable parameters w usually referred to as the **kernels** or **filters**. The output s of these operations is usually referred to as **feature map**.

Convolution operation is a **translation equivariant** operation. Translation of a function f by β is defined as

$$\tau_\beta f = f(x - \beta)$$

Equivariance with respect to translation means that convolution commutes with translation operation

$$\tau_\beta(x * w) = (\tau_\beta x * w) = (x * \tau_\beta w)$$

Translation equivariance is an extremely useful property for neural networks which are used for tasks that it is not required to specifically *where/when* a feature is present but *if* a feature is present. It also enables neural networks to operate on inputs with varying length.

In the context of gradient based learning, convolutional neural networks made their first appearance in [8].

2.2.2.2 Recurrence

Recurrent networks are first mentioned in [9]. Recurrent neural networks (RNNs) are used to process sequential data and model dynamical systems. The description of the state of a dynamical system $\mathbf{s}_t$ dependent on an input signal $\mathbf{x}_t$ with parameters θ is

$$\mathbf{s}_t = f(\mathbf{s}_{t-1}, \mathbf{x}_t; \theta) \tag{2.1}$$

The above equation 2.1 simply describes that the system's current state $\mathbf{s}_t$ depends on the previous state of the system $\mathbf{s}_{t-1}$ and the current input $\mathbf{x}_t$. The equations of

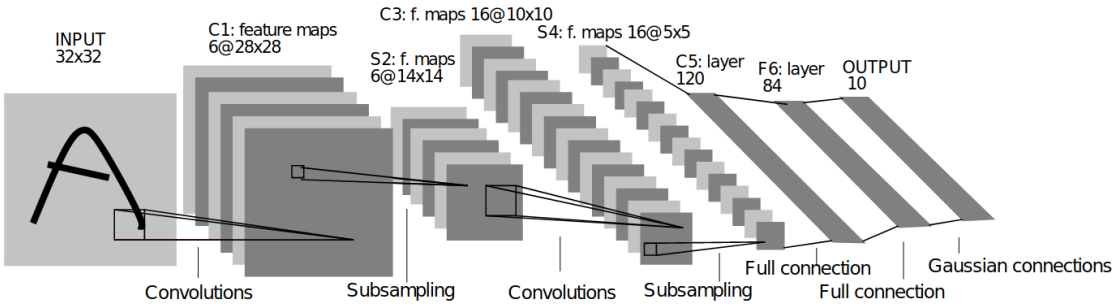


Figure 2.1: LeNet architecture from the paper [8]

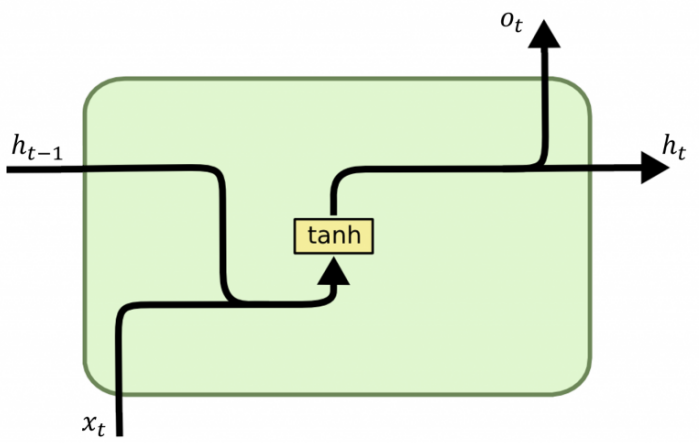


Figure 2.2: RNN's cell architechture

forward propagation of a RNN are

$$\mathbf{h}_t = \phi(\mathbf{b} + W\mathbf{h}_{t-1} + U\mathbf{x}_t)$$
$$\mathbf{o}_t = \mathbf{c} + V\mathbf{h}_t$$

where ϕ is an activation function, usually tanh, and $\mathbf{b}, W, U, \mathbf{c}, V$ are learn-able parameters that are shared across all time-steps. Here, $\mathbf{b}$ is the bias to the new hidden state, W is the hidden-to-hidden matrix, U is the input-to-hidden matrix, $\mathbf{c}$ is the bias vector to the output and V is the hidden-to-output matrix. The RNN cell is shown in figure 2.2

In order to alleviate the problem of vanishing / exploding gradients, two variants have been proposed: Long Short-Term Memory (LSTM) [10] networks and Gated Recurrent Units (GRU) [11] networks. The modified forward equations of LSTM networks are

$$\mathbf{i}_t = \sigma(W_{ii}\mathbf{x}_t + \mathbf{b}_{ii} + W_{hi}\mathbf{h}_{t-1} + \mathbf{b}_{hi})$$
$$\mathbf{f}_t = \sigma(W_{if}\mathbf{x}_t + \mathbf{b}_{if} + W_{hf}\mathbf{h}_{t-1} + \mathbf{b}_{hf})$$
$$\mathbf{g}_t = \tanh(W_{ig}\mathbf{x}_t + \mathbf{b}_{ig} + W_{hg}\mathbf{h}_{t-1} + \mathbf{b}_{hg})$$
$$\mathbf{o}_t = \sigma(W_{io}\mathbf{x}_t + \mathbf{b}_{io} + W_{ho}\mathbf{h}_{t-1} + \mathbf{b}_{ho})$$
$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t$$
$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t)$$

The LSTM's cell architecture is shown in 2.3.

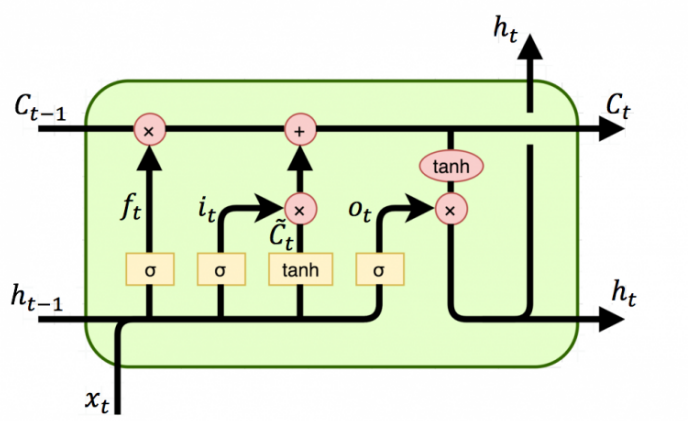


Figure 2.3: LSTM's cell architechture

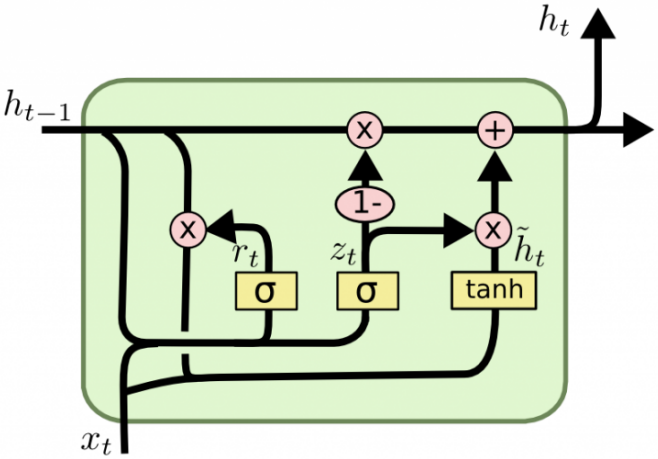


Figure 2.4: GRU's cell architechture

The modified forward equations of GRU networks are

$$\begin{aligned}
\mathbf{r}_t &= \sigma(W_{ir}\mathbf{x}_t + \mathbf{b}_{ir} + W_{hr}\mathbf{h}_{(t-1)} + b_{hr}) \\
\mathbf{z}_t &= \sigma(W_{iz}\mathbf{x}_t + \mathbf{b}_{iz} + W_{hz}\mathbf{h}_{(t-1)} + \mathbf{b}_{hz}) \\
\mathbf{n}_t &= \tanh\big(W_{in}\mathbf{x}_t + \mathbf{b}_{in} + \mathbf{r}_t * (W_{hn}\mathbf{h}_{(t-1)} + \mathbf{b}_{hn})\big) \\
\mathbf{h}_t &= (1 - \mathbf{z}_t) * \mathbf{n}_t + \mathbf{z}_t * \mathbf{h}_{(t-1)}
\end{aligned}$$

The GRU's cell architecture is shown in 2.4.

2.2.2.3 Attention

In [12], attention mechanism has first been used for aligning and translating. The method was named after the first author, Bahdanau attention. In summary, the attention algorithm proposed by Bahdanau performs the following operations:

1. The encoder generates a set of annotations, $\mathbf{h}_i$, from the input sentence.
2. These annotations are fed to an alignment model together with the previous hidden decoder state. The alignment model uses this information to generate the attention scores, $e_{t,i}$.
3. A softmax function is applied to the attention scores, effectively normalizing them into weight values, $\alpha_{t,i}$, in a range between 0 and 1.

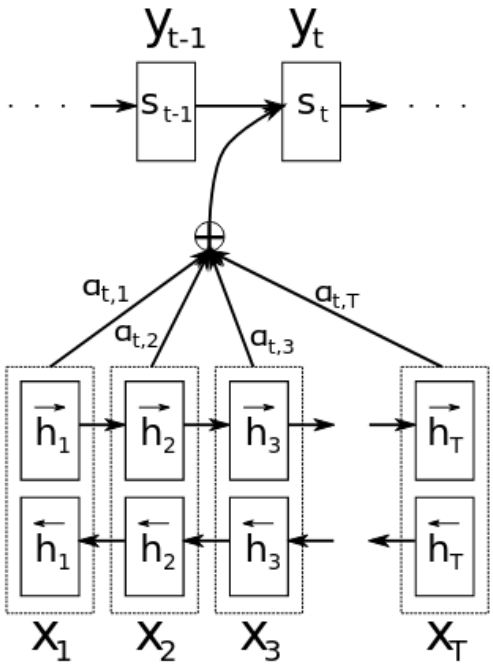


Figure 2.5: Bahdanau attention mechanism

4. These weights together with the previously computed annotations are used to generate a context vector, $\mathbf{c}_t$, through a weighted sum of the annotations.
5. The context vector is fed to the decoder together with the previous hidden decoder state and the previous output, to compute the final output, y_t.
6. Steps 2-5 are repeated until the end of the sequence.

The authors had tested their architecture on the task of English-to-French translation, and had reported that their model outperformed the conventional encoder-decoder model significantly, irrespective of the sentence length.

There had been several improvements over the Bahdanau attention that had been proposed thereafter, such as those of Luong in [13].

2.2.2.4 Self-Attention

The self-attention mechanism has first been proposed in [14] as a part of Transformer's model architecture for the task of machine translation.

Self-attention is a mechanism used by some neural networks to help capture long-range dependencies in input data. It works by allowing the network to focus on different parts of the input at different times, rather than processing the entire input simultaneously. This is done by computing a weighted sum of the input features, where the weights are learned by the network and determined by the input itself. This weighted sum is then used as input to the next layer of the network.

In order to understand how self-attention works, it's helpful to first understand the concept of attention. In general, attention is the ability to focus on a specific part of

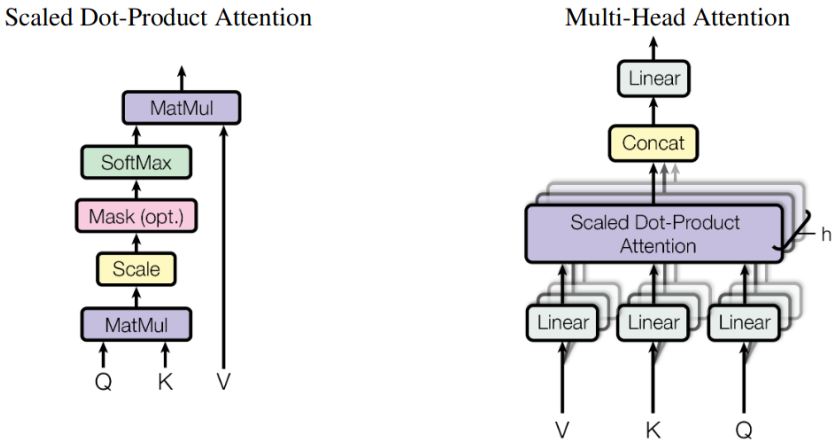


Figure 2.6: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel. (image and caption from [14]

something, and in the context of neural networks, it refers to the ability of a network to focus on specific parts of its input. This is useful because it allows the network to focus on the most important parts of the input and ignore irrelevant information, which can improve its performance.

The self-attention mechanism works by first computing a set of dot products between the input features and a set of learned parameters called keys and values. These dot products are then used to compute a set of weights, which indicate how much attention the network should pay to each input feature. These weights are then used to compute a weighted sum of the input features, which is used as input to the next layer of the network.

Mathematically, this can be expressed as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Here, Q, K, and V are the query, key, and value matrices, respectively, and d_k is the dimension of the keys. The dot product between the query and key matrices is scaled by the square root of the key dimension, and then passed through a softmax function to compute the attention weights. These weights are then used to compute the weighted sum of the value matrix.

The key advantage of self-attention is that it allows the network to focus on different parts of the input at different times, rather than processing the entire input simultaneously. This is useful because it allows the network to capture long-range dependencies in the data, which can be difficult for other types of networks to handle. It also makes the network more interpretable, since the attention weights can be used to understand which parts of the input the network is focusing on at any given time. Overall, self-attention is a powerful tool for improving the performance and interpretability of neural

networks.

Chapter 3

Deep metric and Contrastive learning

3.1 Metric Learning

Metric learning is a subdomain of machine learning whose primary goal is to find a mapping $f : \mathcal{X} \to \mathbb{R}^d$ such that points from domain $\mathcal{X}$ that are recognized similar, are mapped 'close' to each other. Formally, if two points x, x^+ are considered similar and another point x^- is considered different from them, we would like to retrieve a mapping such that

$$d(f(x), f(x^+)) < d(f(x), f(x^-)), \forall x, x^+, x^- \in \mathcal{X} \tag{3.1}$$

where $d : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a distance function

Equation 3.1 states that the distance between every two mapped similar samples x, x^+ is less than the distance between every two mapped dissimilar samples x, x^-.

Since all points of our domain $\mathcal{X}$ accept the same mapping f, it is considered as *global metric learning*

For every metric learning method, two things should be determined:

- The distance function $d : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$
- When are two samples considered similar?

Usually, the distance functions that are used in practice are the Euclidean distance

$$d_{Eucl}(x, y) = ||x - y||_2 \tag{3.2}$$

or the cosine distance

$$d_{\cos}(x, y) = 1 - \frac{x \cdot y}{||x||_2 \cdot ||y||_2} \tag{3.3}$$

The definition of similarity requires **context**. Two samples can be considered similar in a specific context but dissimilar in another context. For example, two speech utterances might be considered similar if they come from the same speaker but in the context of emotion can be considered dissimilar since they may express different emotions. Every method should define which samples are considered similar and dissimilar to a sample.

The metric learning methods can be formulated with and without supervision. The kind of supervision is entirely determined by the definition of similarity.

In the next sections, linear and deep metric learning methods as well as contrastive losses are presented.

3.1.1 Linear metric learning

If the mapping f is chosen to be linear $f(x) = Wx$, the method belongs to linear metric learning methods. Where the Euclidean distance is chosen, sometimes, these methods are referred to as 'Mahalanobis distance metric learning'.

$$d(x, y) = \|f(x) - f(y)\|_2 = \|Wx - Wy\|_2 = \|W(x - y)\|_2 = \sqrt{(x - y)^T W^T W(x - y)} \tag{3.4}$$

The final equation resembles a Mahalanobis distance which is defined as

$$d(x, y) = \sqrt{(x - y)^T \Sigma^{-1} (x - y)} \tag{3.5}$$

where Σ is a positive semi-definite matrix

3.1.1.1 Neighbourhood Component Analysis (NCA)

In Neighbourhood Component Analysis (NCA) [15], a 'soft' nearest neighbour classifier is trained to maximize the probability of assigning the correct class. The probability of two samples x_i, x_j belonging to the same class (x_j being a neighbour of x_i) is defined as

$$p_{ij} = \frac{\exp(-\|Ax_i - Ax_j\|)}{\sum_{k \neq i} \exp(-\|Ax_i - Ax_k\|)}, p_{ii} = 0 \tag{3.6}$$

The probability p_i of sample x_i being classified correctly is the sum of probabilities taken on the similar (same class) samples x_j

$$p_i = \sum_{j : y_i = y_j} p_{ij} \tag{3.7}$$

where y_i denotes the class of sample x_i

The goal of this method is to find the matrix A that maximizes the probability of all samples to be classified correctly

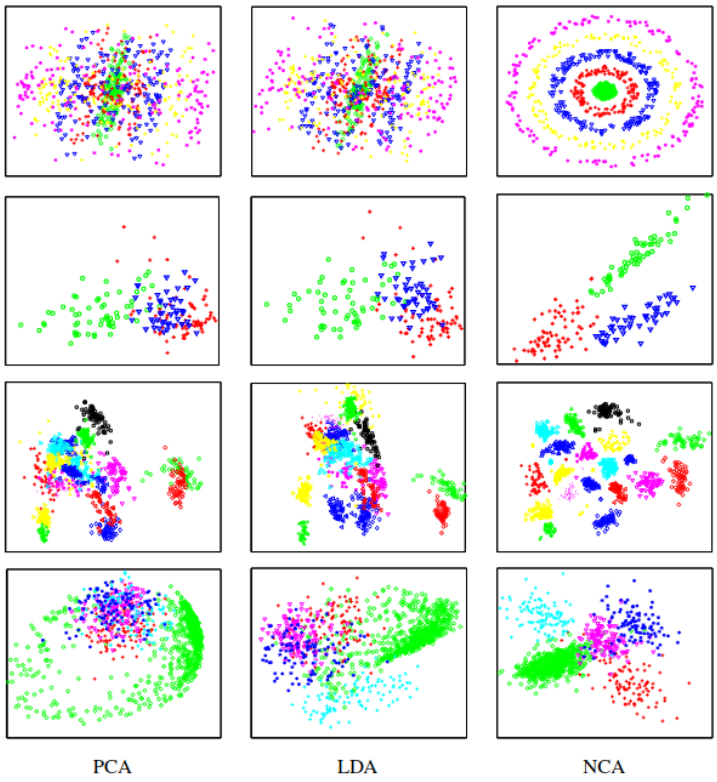


Figure 3.1: Dataset visualization results of PCA, LDA and NCA applied to (from top) the "concentric rings", "wine", "faces" and "digits" datasets. The data are reduced from their original dimensionalities (D=3,D=13,D=560,D=256 respectively) to the d=2 dimensions (image and caption taken from [15])

$$\max \sum_i p_i \tag{3.8}$$

Authors also considered the alternative of maximizing the following expression

$$\max \sum_i \log(p_i) \tag{3.9}$$

The produced representations and the comparison with other methods from the original paper are shown in 3.1

3.1.1.2 Large Margin Nearest Neighbour (LMNN)

The main concept in Large Margin Nearest Neighbour [16] method is to pull near the similar samples while pushing far away the different. The criterion that is minimized for each sample x_i is the following

$$L_i = \mu \cdot \sum_{j:y_i=y_j} d(Ax_i, Ax_j) + (1-\mu) \cdot \sum_{j,l:y_i=y_j\neq y_l} [d(Ax_i, Ax_j) - d(Ax_i, Ax_l) + 1]_+ \tag{3.10}$$

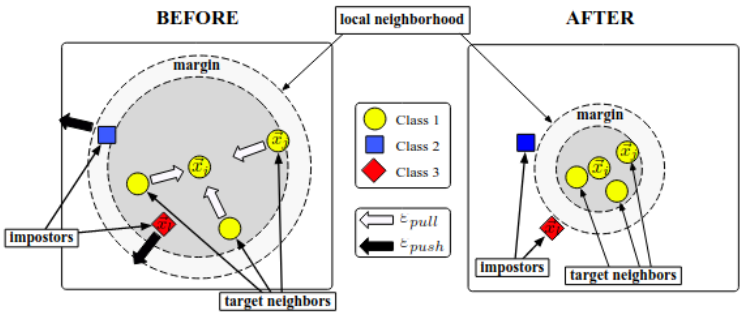


Figure 3.2: Schematic illustration of one input's neighborhood x_i before training (left) versus after training (right) (image and caption taken from [16]).

where d is the Euclidean distance, $[x]_+ = max(x, 0)$, y_i denotes the label of sample x_i and μ is a parameter that controls the weighting of each term. Note that the term $\epsilon_{pull}^{(i)} = \sum_{j:y_i=y_j} d(Ax_i, Ax_j)$ pulls all similar samples to x_i near it whereas the term $\epsilon_{push}^{(i)} = \sum_{j,l:y_i=y_j\neq y_l}[d(Ax_i, Ax_j) - d(Ax_i, Ax_l) + 1]_+$ pushed the different ones further than the similar.

$$L_i = \mu\epsilon_{pull}^{(i)} + (1-\mu)\epsilon_{push}^{(i)}$$

The complete loss is obtained after summing on all available data

$$L = \sum_i L_i$$

In 3.2, the procedure is shown schematically.

3.1.2 Deep Metric Learning / Contrastive Learning

If the mapping f is a parameterized by θ deep neural network f_θ, the methods are usually referred to as Deep Metric Learning or Contrastive Learning methods.

Deep metric learning and contrastive learning are similar techniques that are used to learn a distance metric between data points in a dataset.

In deep metric learning, the goal is to learn a function that maps input data points to a feature space, such that the distance between points in the feature space reflects the similarity between the original data points. This is typically done by minimizing a loss function that measures the difference between the distances of similar pairs of points and dissimilar pairs of points in the feature space.

Contrastive learning is a specific type of deep metric learning that uses a contrastive loss function to train the model. In contrastive learning, the model is trained to pull similar data points together and push dissimilar data points apart in the feature space. The contrastive loss function is designed to maximize the similarity between similar pairs of points and minimize the similarity between dissimilar pairs of points.

Overall, both deep metric learning and contrastive learning are methods for learning a distance metric between data points, with contrastive learning being a specific type of deep metric learning that uses a contrastive loss function to train the model.

For the following section, $\mathbf{z}$ is assumed to be the encoding vector of sample x through the network f_θ.

$$\mathbf{z} = f_\theta(x) \tag{3.11}$$

We proceed by presenting some of the proposed contrastive losses in the literature. Although that most of the following losses can be used with and without supervision, we organize them according to their original proposal. Usually, in the supervised set-up, samples belonging to the same class are considered similar. In the unsupervised losses a specific definition of similarity is required depending on the method.

For the following, we denote as $\mathcal{A}$ the set of the samples in a batch. The set of indices in the batch is also denoted as $A = \{1, 2, \ldots, |\mathcal{A}|\}$. The set of positive samples given an anchor index a is denoted as $P(a)$ and the set of negatives as $N(a)$.

3.1.3 Supervised

In the supervised case, the positive distribution is the one that contains all the other samples from the batch that share the same class with the anchor

$$P(a) = \{p \in A | y_a = y_p, a \neq p\}$$

and the negatives $N(a)$ different

$$N(a) = \{n \in A | y_a \neq y_n\}$$

3.1.3.1 Siamese Networks

Siamese networks are firstly introduced in [17] for the task of signature verification. They operate in the following way: For each sample x_i, the network is also provided with another sample x_j. The network is applied on both samples and provides two encoded vectors $\mathbf{z}_i, \mathbf{z}_j$. If the two samples are considered similar, the loss is the distance between the encoding vectors, whereas if they are considered different the loss is the negation of their distance adding a positive margin. Formally

$$\ell_{ij} = y_{ij} d(f_\theta(x_i), f_\theta(x_j)) + (1 - y_{ij})[m - d(f_\theta(x_i), f_\theta(x_j))]_+ \tag{3.12}$$

where y_{ij} is an indicator that is equal to 1 if samples x_i, x_j are similar and 0 elsewhere, and $m > 0$ is the margin parameter.

Substituting with the encoded vectors, the loss can be rewritten as

$$\ell_{ij} = y_{ij} d(\mathbf{z}_i, \mathbf{z}_j) + (1 - y_{ij})[m - d(\mathbf{z}_i, \mathbf{z}_j)]_+ \tag{3.13}$$

The complete loss is calculated by averaging over all the pairs that can be formed from the batch

$$\mathcal{L}_{Siamese} = \frac{1}{|A|(|A| - 1)} \sum_{i \in A} \sum_{j \in A \setminus \{i\}} y_{ij} d(\mathbf{z}_i, \mathbf{z}_j) + (1 - y_{ij})[m - d(\mathbf{z}_i, \mathbf{z}_j)]_+ \tag{3.14}$$

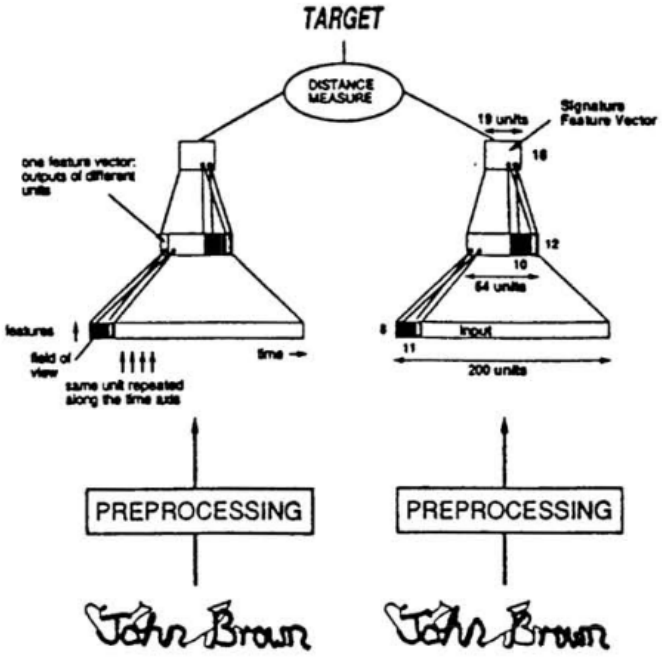


Figure 3.3: Siamese networks architecture (image taken from [17])

Schematically, the Siamese networks are depicted in figure 3.3

3.1.3.2 Triplet Loss

The term "Triplet Network" (fig. 3.4) first shows up in [18] inspired from [19] and [20]. Extending the idea of Siamese networks, each sample x_a (which is usually referred to as anchor) is now compared with two other samples x_p, x_n where x_p is similar to the anchor x_a (positive sample) and x_n is different (negative sample). The formulation of the loss is

$$\ell_{apn} = [d(\mathbf{z}_a, \mathbf{z}_p) - d(\mathbf{z}_a, \mathbf{z}_n) + m]_+ \tag{3.15}$$

where $m > 0$ is the margin parameter.

A visualized representation of the method is shown in 3.5.

The total calculation of the triplet loss involves averaging over the triplets with non-zero loss

$$\mathcal{L}_m^{\text{triplet}} = \frac{1}{\sum_a |\mathcal{T}_a'|} \sum_{a \in A} \sum_{(p,n) \in \mathcal{T}_a} [d(\mathbf{z}_a, \mathbf{z}_p) - d(\mathbf{z}_a, \mathbf{z}_n) + m]_+$$

where $\mathcal{T}_a = P(a) \times N(a)$ is the set of the formed triplets based on anchor a and $\mathcal{T}_a' = \{(p,n) \in \mathcal{T}_a : d(\mathbf{z}_a, \mathbf{z}_n) < d(\mathbf{z}_a, \mathbf{z}_p) + m\}$ is the subset of the formed triplets based on anchor a that contribute to the loss.

Instead of minimizing a notion of distance, a contrastive loss can be employed as a

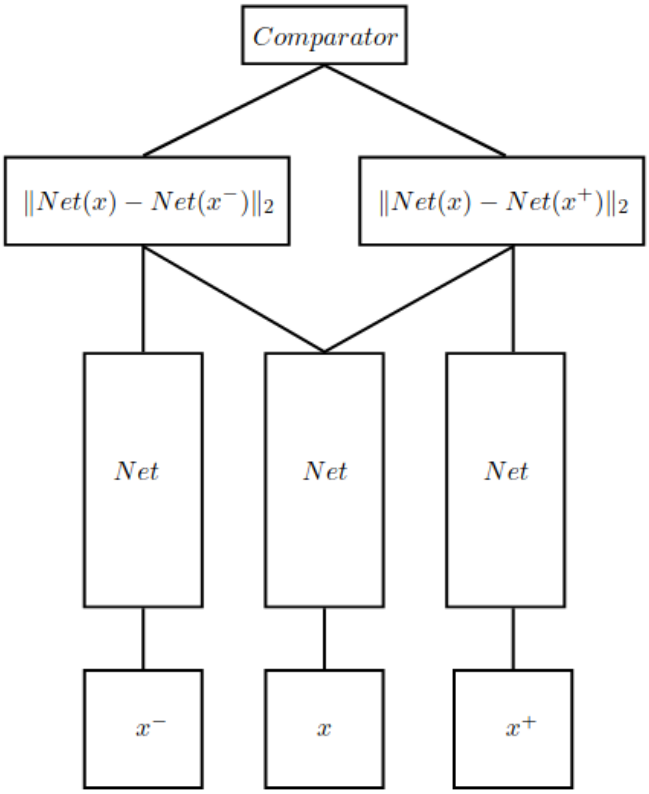


Figure 3.4: Triplet network [18]

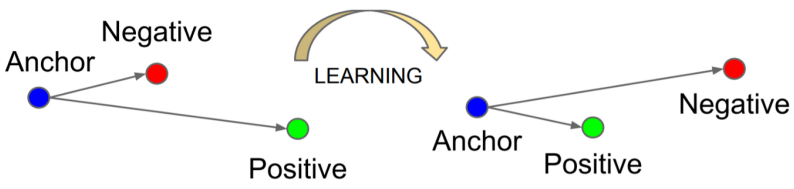


Figure 3.5: Effect after training with the Triplet loss (image taken from [21])

smooth alternative for triplet loss:

$$\ell_{a,p,n}^{triplet-smooth} = -\log\left(\frac{e^{\mathbf{z}_a\cdot\mathbf{z}_p}}{e^{\mathbf{z}_a\cdot\mathbf{z}_p}+e^{\mathbf{z}_a\cdot\mathbf{z}_n}}\right) = -\log\left(\frac{1}{1+e^{-(\mathbf{z}_a\cdot\mathbf{z}_p-\mathbf{z}_a\cdot\mathbf{z}_n)}}\right) = -\log(\sigma(\mathbf{z}_a\cdot\mathbf{z}_p-\mathbf{z}_a\cdot\mathbf{z}_n))$$

where σ is the sigmoid function. Usually, a temperature parameter τ is introduced to control the sharpness of the softmax distribution:

$$\ell_{a,p,n;\tau}^{triplet-smooth} = \log\big(1+e^{(\mathbf{z}_a\cdot\mathbf{z}_n-\mathbf{z}\cdot\mathbf{z}_p)/\tau}\big)$$

Assuming $\mathbf{z}_a\cdot\mathbf{z}_p \gg \mathbf{z}_a\cdot\mathbf{z}_n$ the hard triplet loss can be derived from the smooth contrastive variant as follows:

$$\begin{aligned}
\ell_{a,p,n;\tau}^{triplet-smooth} &= \log\big(1+e^{(\mathbf{z}_a\cdot\mathbf{z}_n-\mathbf{z}_a\cdot\mathbf{z}_p)/\tau}\big) \\
&\approx e^{(\mathbf{z}_a\cdot\mathbf{z}_n-\mathbf{z}_a\cdot\mathbf{z}_p)/\tau} && \text{(Taylor expansion: } \ln(1+x)\approx x\text{)} \\
&\approx 1+\frac{1}{\tau}(\mathbf{z}_a\cdot\mathbf{z}_n-\mathbf{z}_a\cdot\mathbf{z}_p) && \text{(Taylor expansion:} e^x\approx 1+x\text{)} \\
&= 1+\frac{1}{2\tau}(||\mathbf{z}_a-\mathbf{z}_p||_2^2-||\mathbf{z}_a-\mathbf{z}_n||_2^2) \\
&\propto ||\mathbf{z}_a-\mathbf{z}_p||_2^2-||\mathbf{z}_a-\mathbf{z}_n||_2^2+2\tau
\end{aligned}$$

Note that the temperature here acts like the margin controlling the inter- and intra-cluster variability.

3.1.3.3 N-Pair Loss

A generalization of the smooth triplet loss involves contrasting each (a,p) pair with all the negatives n inside the log calculation:

$$\ell_{a,p} = -\log\left(\frac{e^{\mathbf{z}_a\cdot\mathbf{z}_p}}{e^{\mathbf{z}_a\cdot\mathbf{z}_p}+\sum_{n\in N(a)}e^{\mathbf{z}_a\cdot\mathbf{z}_n}}\right) \tag{3.16}$$

Again, usually a temperature parameter τ is used to control the sharpness of the softmax

$$\ell_{a,p;\tau} = -\log\left(\frac{e^{\mathbf{z}_a\cdot\mathbf{z}_p/\tau}}{e^{\mathbf{z}_a\cdot\mathbf{z}_p/\tau}+\sum_{n\in N(a)}e^{\mathbf{z}_a\cdot\mathbf{z}_n/\tau}}\right) \tag{3.17}$$

The summation here is performed on the positive pairs

$$\mathcal{L} = \frac{1}{|A|}\sum_{a\in A}\frac{1}{|P(a)|}\sum_{p\in P(a)}\ell_{a,p}$$

In [3], where N-Pair loss was introduced, in order to compute the loss, the batch is formed in the following way. Two samples are selected for each of B disjoint class forming a batch of size $2B$. The two samples of each class are the anchor-positive pairs. The negatives for each pair are the rest $B-1$ samples from the $B-1$ different classes. The batch construction procedure is shown in 3.6

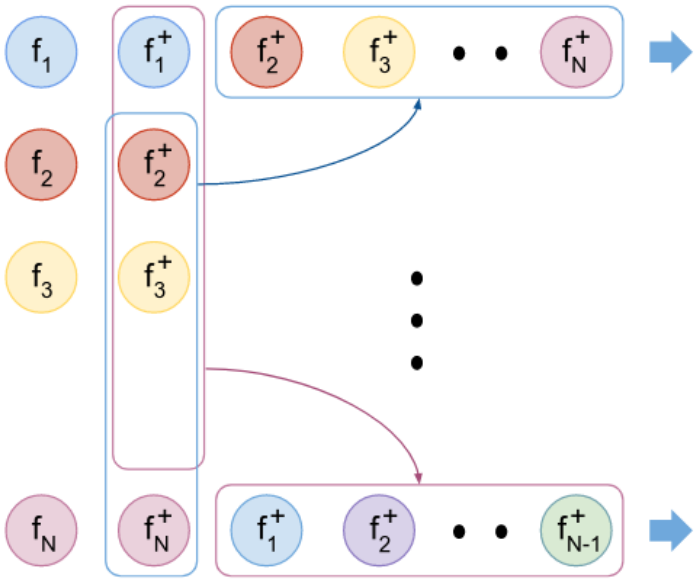


Figure 3.6: Batch construction of N-Pair loss (image taken from [3])

3.1.4 Unsupervised

As stated before, the type of supervision is entirely configured by the choice of positive and negative distribution $(P(a), N(a))$ given the anchor a. The difference between the following methods actually lies in the way of defining those two distributions.

3.1.4.1 InfoNCE / NT-Xent

The multiple negatives formulation has been used under the names of InfoNCE (introduced in [22]) and NT-Xent loss (introduced in [1]) for unsupervised representation learning: CPC [22], CMC [23], MoCo [24], CPC v2 [25]

In [22] the loss that is minimized is of the form of

$$\mathcal{L} = -\mathbb{E}_X[\log \frac{g(x_{t+k}, \mathbf{c}_t)}{\sum_{x \in X} g(x, \mathbf{c}_t)}]$$

where g is an estimator of the unnormalized probability ratio

$$g_k(x_{t+k}, \mathbf{c}_t) \propto \frac{p(x_{t+k}|\mathbf{c}_t)}{p(x_{t+k})}$$

The functional form of g is chosen to be a log-bilinear transformation

$$g_k(x_{t+k}, \mathbf{c}_t) = \exp(\mathbf{z}_{t+k} \cdot W_k \mathbf{c}_t)$$

With this choice of g, the optimized loss takes the form of a softmax contrastive loss

$$\mathcal{L} = -\mathbb{E}_X[\log \left(\frac{e^{\mathbf{z}_{t+k} \cdot W_k \mathbf{c}_t}}{e^{\mathbf{z}_{t+k} \cdot W_k \mathbf{c}_t} + \sum_{x \neq x_{t+k}} e^{\mathbf{z} \cdot W_k \mathbf{c}_t}} \right)]$$

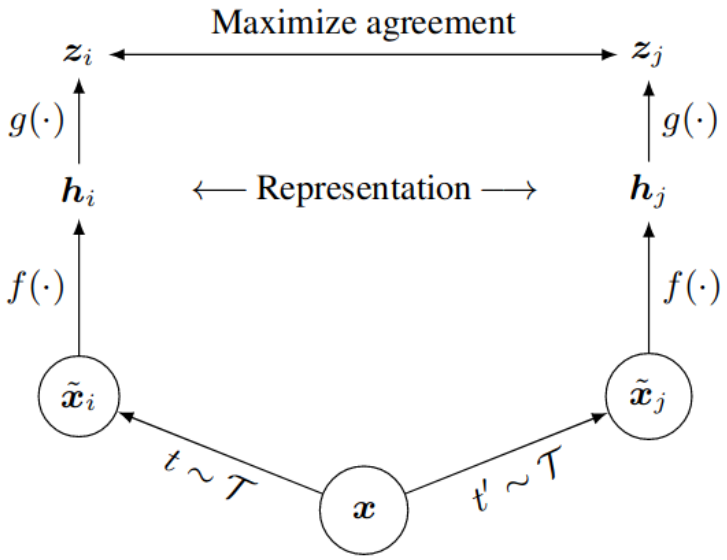


Figure 3.7: Two separate data augmentation operators are sampled from the same family of augmentations ($t - T$ and $t^{'}$ - T) and applied to each data example to obtain two correlated views. A base encoder network $f(\cdot)$ and a projection head $g(\cdot)$ are trained to maximize agreement using a contrastive loss. After training is completed, we throw away the projection head $g(\cdot)$ and use encoder $f(\cdot)$ and representation h for downstream tasks. (image and caption taken from [1])

The NTXent loss can be summarized in the figure 3.7.

A minibatch of N examples is randomly sampled and define the contrastive prediction task on pairs of augmented examples derived from the minibatch, resulting in $2N$ data points. Negative examples are not sampled explicitly. Instead, given a positive pair, the other $2(N-1)$ augmented examples within a minibatch are treated as negative examples. The loss between two samples i, j is

$$\ell_{i,j} = -\log\left(\frac{e^{\mathbf{z}_i \cdot \mathbf{z}_j/\tau}}{\sum_{k=1,k\neq i}^{2N} e^{\mathbf{z}_i \cdot \mathbf{z}_k/\tau}}\right)$$

Note that the above equation can actually be derived from the general form of 3.17 by forming the batch in the way it was described and selecting all other samples in the batch as negatives.

3.1.4.2 SupCon

There have also been proposed variants that include the summation of positives inside the logarithm calculation as well, leading to a general form of:

$$\ell_a = -\log\left(\frac{\sum_p e^{\mathbf{z}_a \cdot \mathbf{z}_p}}{\sum_p e^{\mathbf{z}_a \cdot \mathbf{z}_p} + \sum_n e^{\mathbf{z}_a \cdot \mathbf{z}_n}}\right)$$

The summation is performed over the anchors in the sample

$$\mathcal{L} = \frac{1}{|A|}\sum_{a\in A} \ell_a$$

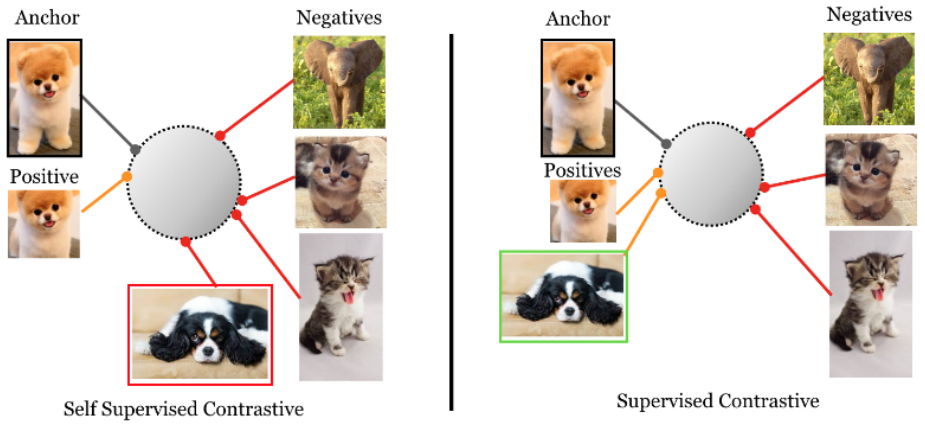


Figure 3.8: Supervised vs. self-supervised contrastive losses: The self-supervised contrastive loss (left) contrasts a single positive for each anchor (i.e., an augmented version of the same sample) against a set of negatives consisting of the entire remainder of the batch. The supervised contrastive loss (right), however, contrasts the set of all samples from the same class as positives against the negatives from the remainder of the batch. (image taken from [2])

The idea of [1] is used with supervision in [2] and it is extended by using multiple positives per anchor in two different versions of the SupCon loss

$$\mathcal{L}_{out}^{sup} = \sum_{a \in A} \frac{1}{|P(a)|} \sum_{p \in P(a)} -\log \left(\frac{e^{\mathbf{z}_a \cdot \mathbf{z}_p / \tau}}{\sum_{p \in P(a)} e^{\mathbf{z}_a \cdot \mathbf{z}_p / \tau} + \sum_{n \in N(a)} e^{\mathbf{z}_a \cdot \mathbf{z}_n / \tau}} \right)$$

and

$$\mathcal{L}_{in}^{sup} = \sum_{a \in A} -\log \left(\frac{1}{|P(a)|} \sum_{p \in P(a)} \frac{e^{\mathbf{z}_a \cdot \mathbf{z}_p / \tau}}{\sum_{p \in P(a)} e^{\mathbf{z}_a \cdot \mathbf{z}_p / \tau} + \sum_{n \in N(a)} e^{\mathbf{z}_a \cdot \mathbf{z}_n / \tau}} \right)$$

In the paper it is shown that $\mathcal{L}_{out}^{sup}$ behaves better than $\mathcal{L}_{in}^{sup}$ since it is upper bounds ($\mathcal{L}_{in}^{sup} \leq \mathcal{L}_{out}^{sup}$) it because of logarithm's concavity.

The idea of SupCon loss is visually shown in figure 3.8.

A similar approach with multiple positives but with Euclidean distance is taken in [**?**] where the loss is formulated as:

$$\mathcal{L}^{snnl} = \frac{1}{|A|} \sum_{a \in A} -\log \frac{\sum_{p \in P(a)} e^{-\frac{||\mathbf{z}_a - \mathbf{z}_p||^2}{T}}}{\sum_{p \in P(a)} e^{-\frac{||\mathbf{z}_a - \mathbf{z}_p||^2}{T}} + \sum_{n \in N(a)} e^{-\frac{||\mathbf{z}_a - \mathbf{z}_n||^2}{T}}}$$

In [**?**] a different approach is taken by maximizing the loss instead of minimizing and it is shown that it provides adversarial robustness.

3.1.5 Relationship with Cross Entropy Loss

In the classification problem with C classes, we seek to find a mapping from the input distribution $\mathcal{X}$ to the true vector of class probabilities $\mathbf{p} \in \mathbb{R}^C$. Assume we have a

sample $x_i \in \mathcal{X}$ which belongs to the class $y_i \in [C]$. The true class distribution is a vector $\mathbf{p}_i \in \mathbb{R}^C$ with 1 at position $c = y_i$ and 0 elsewhere. The predicted class distribution is a vector $\mathbf{q}_i \in \mathbb{R}^C$ which is obtained from the mapping $\mathbf{q}_i = \text{softmax}(\mathbf{u}_i)$, where $\mathbf{u}_i = h_\theta(x_i)$ and $h_\theta(\cdot)$ is a neural network parameterized by θ and the softmax function of a vector $\mathbf{z}$ is defined as $\text{softmax}(\mathbf{u})_i = \frac{e^{u_i}}{\sum_j e^{u_j}}$. Typically, the last layer of the network is a linear mapping $g(\mathbf{z}) = W\mathbf{z} + \mathbf{b}$ of the last representation $\mathbf{z}$ to the unnormalized probabilities $\mathbf{u}$. The unnormalized probability of class c is then given by $u_c = \mathbf{w}_c \cdot \mathbf{z} + b_c$. Usually, the loss that is applied for the i-th sample is the cross entropy between the true and the predicted distributions

$$
\begin{aligned}
\mathcal{L}_{ce} &= \sum_{x_i \in \mathcal{X}} H(\mathbf{p}_i, \mathbf{q}_i) \\
&= -\sum_{x_i \in \mathcal{X}} \mathbf{p}_i \cdot \log(\mathbf{q}_i) \\
&= -\sum_{x_i \in \mathcal{X}} \sum_{c \in [C]} \mathbf{p}_i[c] \cdot \log(\mathbf{q}_i[c]) && \text{Note that: } \mathbf{p}_i[c = y_i] = 1 \text{ and 0 elsewhere} \\
&= -\sum_{x_i \in \mathcal{X}} \log(\mathbf{q}_i[y_i]) \\
&= -\sum_{x_i \in \mathcal{X}} \log(\text{softmax}(\mathbf{u}_i)[y_i]) \\
&= -\sum_{x_i \in \mathcal{X}} \log\left(\frac{e^{u_{y_i}}}{\sum_{c \in [C]} e^{u_c}}\right) \\
&= -\sum_{x_i \in \mathcal{X}} \log\left(\frac{e^{\mathbf{w}_{y_i} \cdot \mathbf{z}_i + b_{y_i}}}{e^{\mathbf{w}_{y_i} \cdot \mathbf{z}_i + b_{y_i}} + \sum_{c \neq y_i} e^{\mathbf{w}_c \cdot \mathbf{z}_i + b_c}}\right)
\end{aligned}
$$

Ignoring the bias terms the loss becomes:

$$
-\sum_{x_i \in \mathcal{X}} \log\left(\frac{e^{\mathbf{w}_{y_i} \cdot \mathbf{z}_i}}{e^{\mathbf{w}_{y_i} \cdot \mathbf{z}_i} + \sum_{c \neq y_i} e^{\mathbf{w}_c \cdot \mathbf{z}_i}}\right) \tag{3.18}
$$

In order to minimize the loss, we have to maximize the ratio inside the logarithm. Looking closely on the expression of the ratio, we can see that the network must learn for each class c a vector w_c that has maximal dot product similarity with the representation of each sample of the same class while maintaining low similarity with the samples of different classes. Specifically, the nominator $e^{\mathbf{w}_{y_i} \cdot \mathbf{z}_i}$ pulls the class representative vector $\mathbf{w}_{y_i}$ near $\mathbf{z}_i$ whilst the second term of the denominator $\sum_{c \neq y_i} e^{\mathbf{w}_c \cdot \mathbf{z}_i}$ pushes the other class vectors $w_c, (c \neq y_i)$ away from $\mathbf{z}_i$

Note that this variant bears resemblance with the cross-entropy loss ignoring the bias terms as presented in section **??**

$$
-\log\left(\frac{e^{\mathbf{w}_{y_i} \cdot \mathbf{z}_i}}{e^{\mathbf{w}_{y_i} \cdot \mathbf{z}_i} + \sum_{c \neq y_i} e^{\mathbf{w}_c \cdot \mathbf{z}_i}}\right) \tag{3.19}
$$

Equation 3.17 is non-parameterized softmax since there are no learned parameters like 3.19. In cross entropy the learned vector of the class similar to the sample $\mathbf{w}_{y_i}$ acts as another positive sample and the different class vectors $\mathbf{w}_c (c \neq y_i)$ as the negative.

Chapter 4

Applying Contrastive Learning Methods to SER

In this section we present the main part of our work and contribution. At first, we provide an overview of the existing work in the SER and in the Contrastive Learning domain. We proceed by explaining our contribution and describing the experiments we have conducted to verify it.

4.1 Related work

Early works on SER have mainly focused on extracting finding a representative set of emotion features and the optimal time-scale for emotional context extraction. Prosodic, spectral and voice quality Low Level Descriptors (LLDs), extracted from speech frames, have been extensively used for SER [26], [27]. Some of the most widely used LLDs or local features are Mel Frequency Cepstral Coefficients (MFCCs), pitch or fundamental frequency, short-time energy from frames, Zero Crossing Rate (ZCR) and Harmonic to Noise Ratio (HNR). All these LLDs are extracted directly from speech frames corresponding to 23 windows of 20-100ms. Since these feature sets are extracted per frame, an aggregation should be performed in order to obtain a global representation of the whole utterance. This aggregation initially was performed by applying statistical functionals over the LLDs on the whole sequence. Later, the aggregation was made by training cpnvolutional (CNN) and/or recurrent networks (RNN, LSTM, GRU) on top of the extracted LDDs and then summarizing through an attention mechanism [28], [29], [30], [31], [32]. As the computational power increased, the SER systems started to perform feature extraction with minimal signal processing. No LLDs were extracted but the neural networks performed themselves the feature extraction step using spectrograms [33], [34], [35] or Mel-spectrograms [36], [37] as their input or even the raw speech signal as a waveform [38], [39], [40], [41], [42].

Deep metric learning has gained a lot of attention and lots of works have been produced using contrastive learning methods. In the domain of SER, there have also been attempts to improve the performance using contrastive learning techniques. In [42] the authors combine a self-supervised NT-Xent loss with a reconstruction loss in order to pre-train an encoder on LibriSpeech dataset which is then fine-tuned on IEMOCAP. In [43] pre-trained models are fine-tuned using supervised triplet loss and unsupervised

Barlow Twins [44] loss. Recent works [45], [46] use features from a pre-trained wav2vec 2.0 [47] model which is also a model that is trained using a self-supervised contrastive loss in an auto-regressive manner.

4.2 Goals and Contribution of the current work

The motivation of this work is to aid the building of successful SER systems. Towards this goal we believe that the contribution of this work is two-fold:

1. **Thorough investigation on the effectiveness of contrastive learning methods on the task of SER**: In this work we have applied several contrastive learning methods and measured their effect. We have tried several supervised contrastive losses (Triplet [18], NT-Xent [1], SupCon [2]) where our results suggest that the performance is similar with that of the supervised cross-entropy training.
2. **Self-supervised pre-training and supervised fine-tuning with cross-entropy performs better than simply training with cross-entropy**: The results of our experimetns showed that it is preferable to pre-train the network using NT-Xent loss instead of adding it as a regularized to cross-entropy. Actually, if the network is pre-trained with self-supervision the accuracy of a fine-tuned classifier using augmentations and cross-entropy increases (see table 4.9).

4.3 General description of the methodology and experimental setup

In the current work, the contrastive learning paradigm is tested on the task of speech emotion recognition (SER). Before proceeding to the detailed description of the specific experiments we proceed by giving information about the data, the models and the optimization procedure that is common across experiments.

Our baseline model is a deep neural network classifier h that is trained using the cross-entropy loss on a source dataset, here IEMOCAP. As mentioned before, the classifier can be decomposed into two networks, a non-linear sequence to vector encoder f and a linear classifier g, $h = g \circ f$. Our baseline is the same encoder model with an added linear layer that is trained using Cross-Entropy loss.

4.3.1 Dataset and data preprocessing

The benchmark dataset that is used for our experiments is the Interactive EMOtional dyadic motion CAPture database (IEMOCAP [48]) provided by Signal Analysis and Interpretation Laboratory (SAIL) lab at University of Southern California (USC). The dataset contains about 12 hours of scripted and improvised speech. It consists of 5 sessions where in each one of them 2 speakers, one male and one female, interact with each other. Each utterance is annotated with one of 5 emotion categories (neutral, sad, happy, angry, excited). In order to be aligned and comparable with other works, we merge the 'happy' and the 'excited' into one category. The total number of utterances

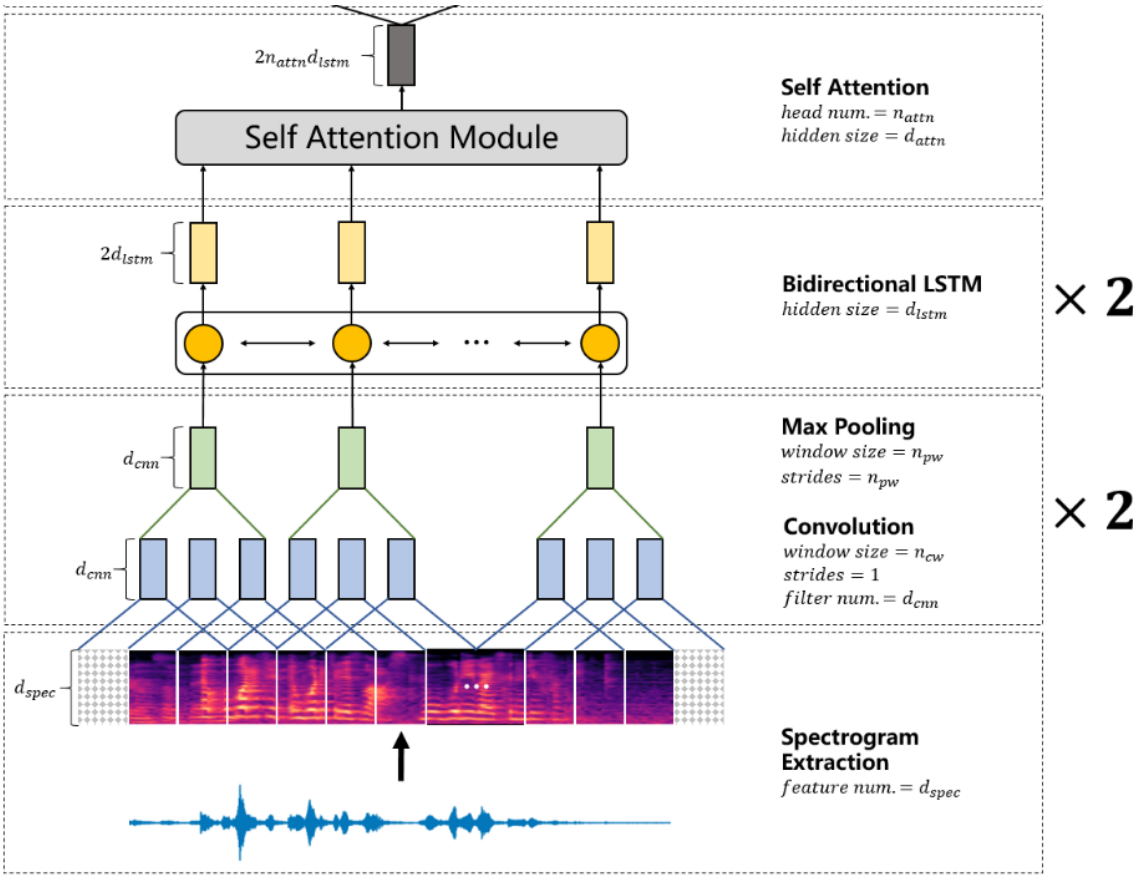


Figure 4.1: The architecture of the encoder network (image taken from [35])

in the dataset is 5531 including distributed as 1103 angry, 1636 happy, 1708 neutral and 1084 sad.

The neural networks that are used in this work operate using the spectrogram of the speech utterance as input. It is a design choice driven by the size of the dataset and the computational resources since neural networks that use raw waveform as input are large and difficult to optimize. Each speech utterance is sampled at 16kHz and the log-spectrogram is extracted using an FFT size of 512 with a time window of 25ms (400 samples) every 10ms (160 samples). No normalization is applied on the extracted features. The same pre-processing steps are applied on the samples.

4.3.2 Model

A common architecture is used for the sequence encoder f across all experiments in order to provide fair comparisons between the different methods of optimization and hyper-parameters. The architecture of the sequence encoder is the same with [35] which is shown in image 4.1 and described in the table 4.3.2.

Sequence Encoder Architecture
Conv2D (C=64, K=(41, 21), S=(2, 2))
BatchNorm2D
Hard-tanh(0, 20)
Conv2D (C=64, K=(21, 11), S=(2, 2))
BatchNorm2D
Hard-tanh(0, 20)
Bi-GRU (H=128, drop=0.2, bi-directional) x 2
TransformerEncoder ($d_{model} = 256, d_{ff} = 512, n_{head} = 8$)

The encoder outputs are l2-normalized i.e. $||\mathbf{z}||_2 = 1$ when training with contrastive losses.

4.3.2.1 Optimization

In order to provide **speaker independent** performance analysis and comparable results with other works, we follow the standard **Leave-One-Session-Out (LOSO)** procedure when training with the IEMOCAP dataset. The dataset has 5 sessions with 2 speakers in each session. Following the LOSO procedure, for each training round we use 4 sessions for training and we leave one out from which we use the one speaker for validation and the other for testing. We then swap the validation and the test speaker and repeat the training. This leads in a total of **10** different folds for cross validation. In order to form the batches that are presented to the network, the sequences either **padded or cropped to match the mean sequence length in the batch**. This procedure ensures that there will not be many zeros because of padding small utterances to reach the same length with long ones and will not have great data loss because of cropping longer sequences into small lengths. It also exposes the network during training into different sequence lengths. Unless stated otherwise, every model in the following experiments is trained for **50 epochs** using **Adam** as the optimization algorithm and an **initial learning rate** of **0.001** using a batch with **32 samples for the baseline** and **128 for the contrastive losses**. The learning rate is **divided by 10** if no improvement is shown in the monitored validation metric **after 3 epochs**. For the baseline, the monitored metric is the Unweighted Accuracy (UA) and for the contrastive losses the loss serves as the validation metric.

4.3.2.2 Evaluation

The primary task that representations are tested is emotion recognition but they are evaluated on the tasks of retrieval and clustering.

- Recognition: For the baseline, since the training is done end-to-end, the classifier is evaluated on the validation and test set. For the encoders obtained after training with contrastive losses, an additional SVM classifier with a linear kernel on the representations of the train set after applying the trained encoder. The evaluation metrics that are calculated is the Unweighted Accuracy (UA) and the Weighted Accuracy (WA).
- Retrieval: For each sample in the validation and test set, the nearest neighbor is selected using the Euclidean distance model the average recall for each class is computed (Recall@1).

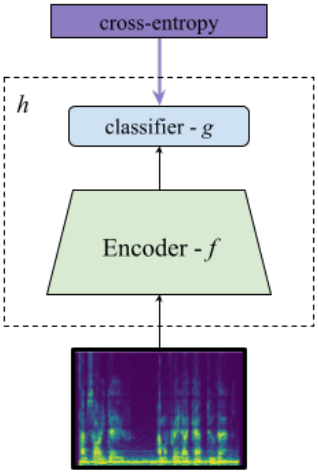


Figure 4.2: Training classifier h end-to-end with cross-entropy. The whole system is composed of an encoder f and a linear mapping g.

Metric	Validation	Test
Unweighted Accuracy (UA)	$54.79 \pm 2.76\%$	$54.07 \pm 4.03\%$
Weighted Accuracy (WA)	$51.46 \pm 4.20\%$	$51.64 \pm 3.69\%$

Table 4.1: Baseline results

- Clustering: The clustering algorithm of k-means is run on the representations given the true number of emotions classes. The F1 and Natural Mutual Information (NMI) scores are computed.

4.4 Experiments and results

4.4.1 Baseline

In order to have a baseline to compare the following experiments, a classifier h is trained end-to-end using the standard cross entropy loss. The encoder consists of a non-linear encoder f which has the architecture that was described in 4.3.2 (without $l2$-normalization) and it is followed by a learnable linear mapping g (see also figure 4.2). In terms of speech emotion recognition which is the main task at hand, the baseline model gave the following results that are shown in table 4.1.

4.4.2 Supervised Contrastive Learning

For this set of experiments, a deep neural network encoder f that maps the input sequence into a vector is first trained using a supervised contrastive loss and subsequently the obtained representations are used for emotion recognition. Each loss is formed with supervision meaning that the positive samples $P(a)$ are the ones sharing the same class

$$P(a) = \{p \in A | y_a = y_p, a \neq p\}$$

and the negatives $N(a)$ different

$$N(a) = \{n \in A | y_a \neq y_n\}$$

where A is the set of the anchor indices.

The proxy metric for the quality of the representations is chosen to be the Unweighted class Accuracy (UA). In order to evaluate the UA, a classifier g is fitted on top of the encoder f. The composition of the two learned components will lead us to an end-to-end classfier $h = g \circ f$. Depending on the experiment we either fit a linear SVM on top of the trained encoder or we fine-tune the encoder using a learnable linear classifier g (see fig. 4.3). In both cases we obtain an end-to-end SER system h.

4.4.2.1 Triplet Loss

The first experiment is to train the encoder f using the triplet loss with the cosine distance

$$\mathcal{L}_m^{\text{triplet}} = \frac{1}{\sum_a |\mathcal{T}_a'|} \sum_{a \in A} \sum_{(p,n) \in \mathcal{T}_a} \max(0, \mathbf{z}_a \cdot \mathbf{z}_n - \mathbf{z}_a \cdot \mathbf{z}_p + m)$$

where $\mathcal{T}_a = P(a) \times N(a)$ is the set of the formed triplets based on anchor a and $\mathcal{T}_a' = \{(p,n) \in \mathcal{T}_a : \mathbf{z}_a \cdot \mathbf{z}_p < \mathbf{z}_a \cdot \mathbf{z}_n + m\}$ is the subset of the formed triplets based on anchor a that contribute to the loss. Note that this formulation means that the loss is averaged only on the non-zero triplets.
The two most important hyper-parameters of this method is a) the margin m and b) the choice of (p, n) pairs given an anchor a that will be used to form the triplets and compute the loss. We begin by investigating the effect of the margin m hyper-parameter without using a specific strategy on triplet sampling and then we investigate the effect of the sampling strategy on a fixed margin.

Margin

For the first experiment, all the triplets that can be formed from the batch are used and the following 8 values are tried for the margin: {0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2}. There is no need to go beyond a margin of 2 since it is the maximum difference (if and only if $\mathbf{z}_a^T \mathbf{z}_n = 1, \mathbf{z}_a^T \mathbf{z}_p = -1$) that can be achieved since the representation vectors are $l2$-normalized. After training the encoders f, a SVM with linear kernel is trained on top as classifier g to measure the accuracy using the learned representations. The comparison between the baseline and the best model that was trained using Triplet loss can be seen in table 4.2. The baseline model is better in terms of UA (53.07%) but not far from the contrastive model combined with the SVM (54.91%). Complete results for all tasks and metrics are shown in 4.4. As we can see from the plots, the best results are obtained using a margin value between 0.05 and 0.5. This behaviour is expected because the margin controls the tightness of the clusters. Very low margin value corresponds to very loose clusters and on the contrary very high margin means

Figure 4.3: The procedure of supervised contrastive training an encoder f. It consists of two phases: train the encoder f using a contrastive loss (left) and then fit a linear SVM on top as a classifier g (right up) or fine-tune with a learnable classifer g (right down)

Loss	Validation	Test
Cross-Entropy	$54.79 \pm 2.76\%$	$54.07 \pm 4.03\%$
Triplet - m=0.05 (all)	$52.18 \pm 2.54\%$	$53.23 \pm 2.88\%$

Table 4.2: Comparison between the baseline and the best model trained using Triplet loss in terms of UA

very tight clusters. As it can be seen from the plots in fig. 4.5, the greater the margin, the tighter the clusters that are formed.

Sampling strategy

The triplets can be divided into three regions:

1. **Easy**: Triplets that are already satisfying the constraint and thus not contributing to the loss

$$\{(a, p, n) \in A \times P(a) \times N(a) \ \mid \mathbf{z}_a^T \cdot \mathbf{z}_p > \mathbf{z}_a^T \cdot \mathbf{z}_n + m\}$$

2. **Semi-Hard**: Triplets that *positive are already more similar from the anchor than the negative but not more than the margin.*

$$\{(a, p, n) \in A \times P(a) \times N(a) \mid \mathbf{z}_a^T \cdot \mathbf{z}_n + m > \mathbf{z}_a^T \cdot \mathbf{z}_p > \mathbf{z}_a^T \cdot \mathbf{z}_n\}$$

3. **Hard**: Triplets that *negative is more similar to the anchor than the positive*

$$\{(a, p, n) \in A \times P(a) \times N(a) \mid \mathbf{z}_a^T \cdot \mathbf{z}_p < \mathbf{z}_a^T \cdot \mathbf{z}_n\}$$

We examine three different strategies on triplet mining:

1. **semi-hard**: Only Semi-Hard triplets are used
2. **hard**: Only Hard triplets are used
3. **all**: Semi-Hard and Hard triplets are combined

In order to examine the effect of the negative sampling, a fixed value for the margin is selected, here $m = 0.2$ and each of the three different strategies are applied. A mixed strategy is also tested where for the first 80% of epochs the network is trained with semi-hard triplets and for the rest 20% it is provided only with hard triplets. The results for the recognition task in terms of UA are shown in table 4.3. As it can be seen from the table, using only hard triplets the network cannot converge possibly because it is stuck in local minima. Semi-hard triplets improve upon the **all** strategy and results get slightly better when changing to **hard** in the last epochs of the training. Complete results for all tasks and metrics are shown in 4.6. Again, we provide visual material for the structure of the learned representation space depending on the triplet mining strategy 4.7.

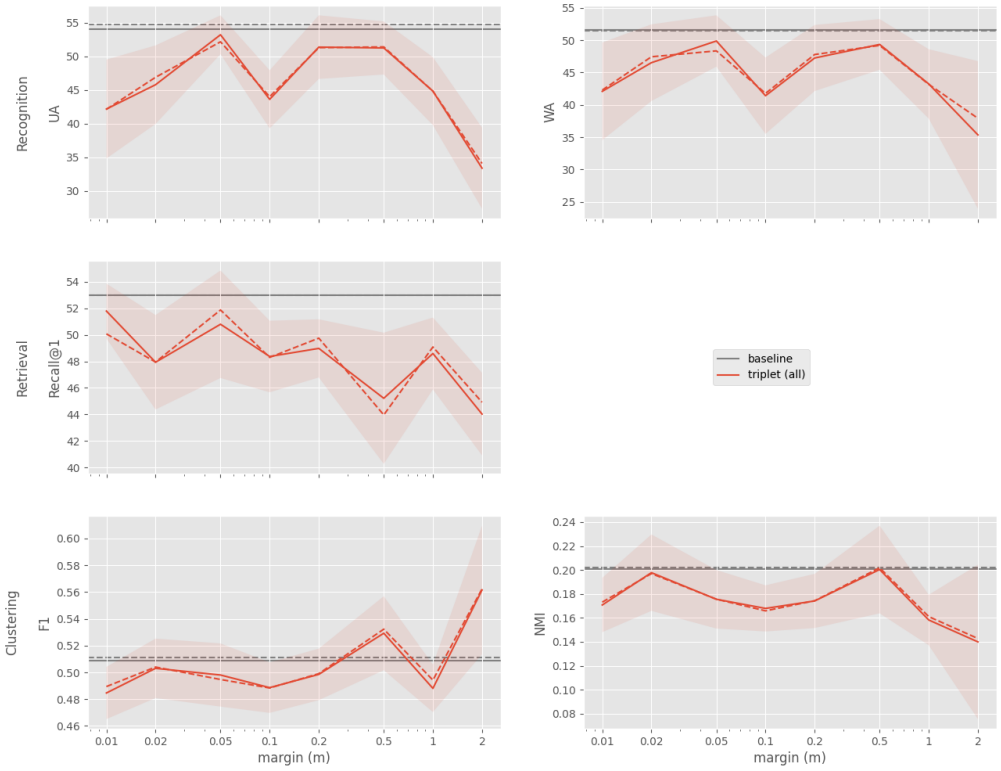
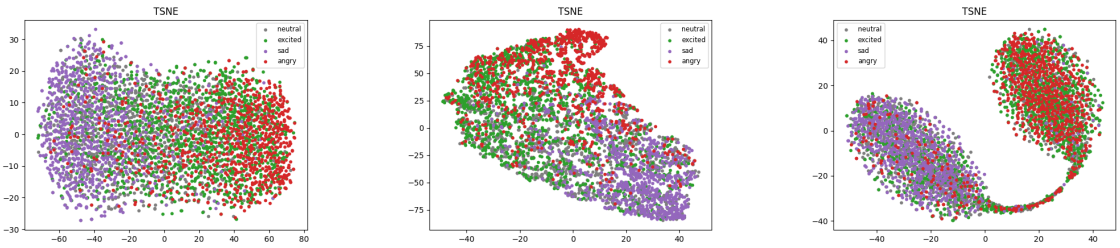


Figure 4.4: Effect of *margin (m)* parameter when training with the Triplet loss using all the triplets in the batch. In terms of recognition, the optimal values seem to be in the range of 0.05 up to 0.5.



(a) $m = 0.02$ (b) $m = 0.2$ (c) $m = 2$

Figure 4.5: Effect of the margin to the tightness of the clusters

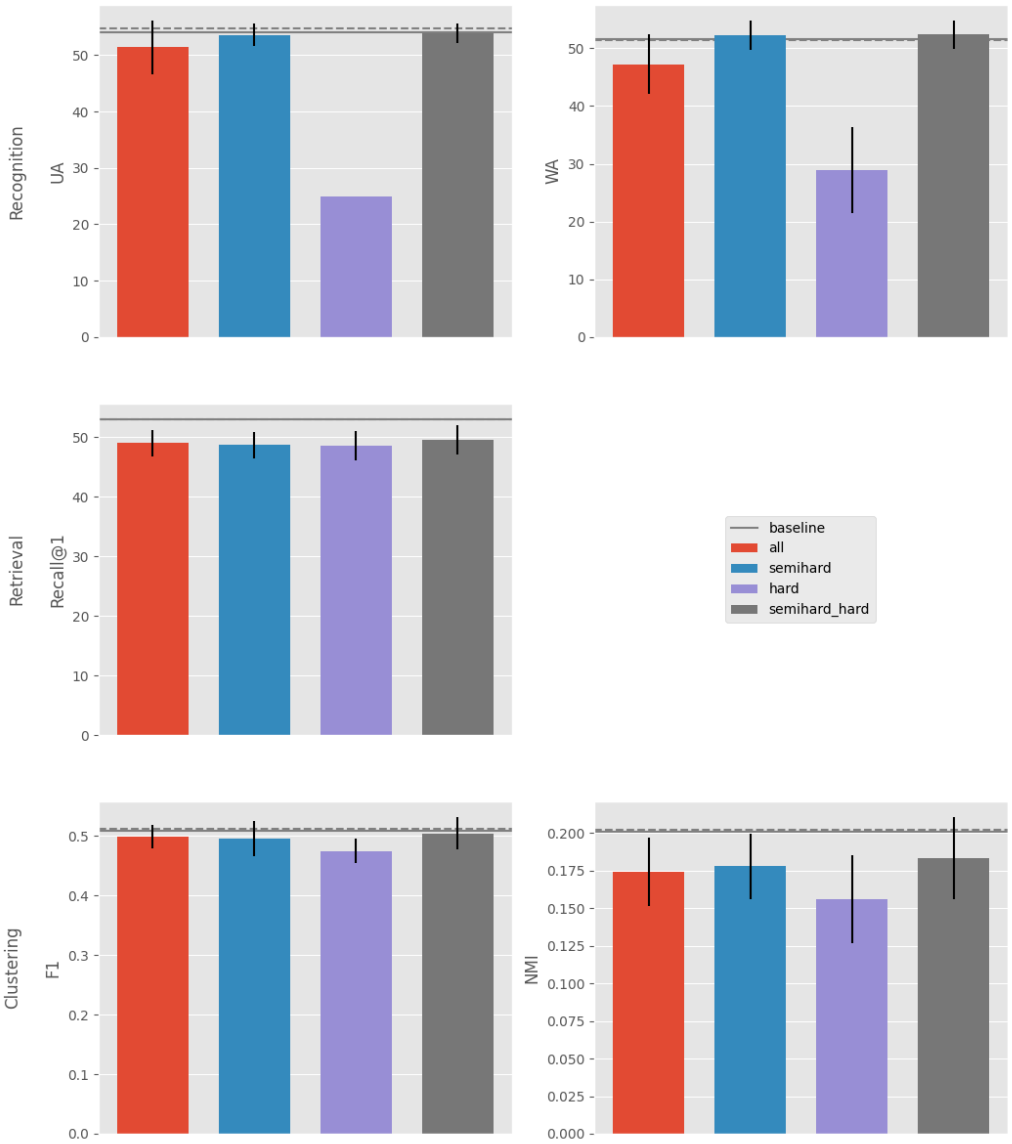
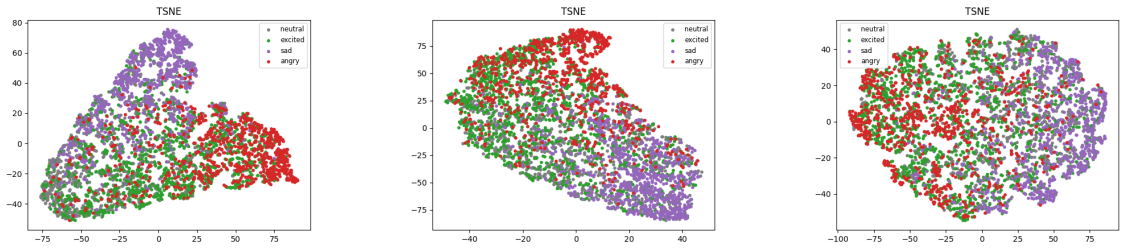


Figure 4.6: Effect of different triplet mining strategies when training with the Triplet loss

Loss	Validation	Test
Cross-Entropy	**54.79** $\pm$ 2.76%	**54.07** $\pm$ 4.03%
Triplet - m=0.2 (all)	52.18 $\pm$ 2.54%	53.23 $\pm$ 2.88%
Triplet - m=0.2 (semihard)	53.77 $\pm$ 2.69%	53.62 $\pm$ 2.05%
Triplet - m=0.2 (hard)	25.00 $\pm$ 0.00%	25.00 $\pm$ 0.00%
Triplet - m=0.2 (semihard, hard)	**54.62** $\pm$ 2.79%	**53.91** $\pm$ 1.71%

Table 4.3: Comparison between the baseline and different mining strategies of Triplet loss in terms of UA. Although, none of the triplet losses configurations exceeds the cross-entropy baseline, the results are very close.



(a) semi-hard (b) all (c) hard

Figure 4.7: Effect of the triplet sampling strategy to the tightness of the clusters

4.4.2.2 Smooth variant of triplet loss

We repeat the same experiment using the smooth variant of triplet loss

$$\mathcal{L}_{\tau}^{\text{triplet-smooth}} = \frac{1}{\sum_{a \in A} |\mathcal{T}_a|} \sum_{a \in A} \sum_{(p,n) \in \mathcal{T}_a} \log\left(1 + e^{(\mathbf{z}_a \cdot \mathbf{z}_n - \mathbf{z} \cdot \mathbf{z}_p)/\tau}\right)$$

Since there is no hard margin here, all the triplets in the batch are harvested and the temperature τ is selected from the following values {0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10}. Results are grouped and presented together along with the next section.

4.4.2.3 Multiple negative and/or positives per anchor

In this set of experiments, we compare and contrast each anchor sample with more than one positive and/or negative samples. We expect that the encoder network will benefit from increasing the number of contrastive samples and it will learn a better structure of the representation space. In order to do this, we use the NT-Xent [1] and SupCon [2] loss. In this section, no augmentations are used and the use of augmentations will be studied in the next section. Here, we try the NT-Xent loss [1] in a supervised way i.e. by using as negative samples, the samples in the batch that belong to a different class from the anchor-positive pair

$$\mathcal{L}_{\tau}^{\text{NT-Xent}} = \frac{1}{|A|} \sum_{a \in A} \frac{1}{|P(a)|} \sum_{p \in P(a)} -\log\left(\frac{e^{\mathbf{z}_a \cdot \mathbf{z}_p/\tau}}{e^{\mathbf{z}_a \cdot \mathbf{z}_p/\tau} + \sum_{n \in N(a)} e^{\mathbf{z}_a \cdot \mathbf{z}_n/\tau}}\right)$$

Since we do not use augmentations, the loss is practically the same with N-Pair loss [3] but without restricting the batch to contain only one sample per emotion class. The

Loss	Validation	Test
Cross-Entropy	$\mathbf{54.79} \pm 2.76\%$	$\mathbf{54.07} \pm 4.03\%$
Triplet Smooth - $\tau = 0.5$	$54.62 \pm 2.89\%$	$54.52 \pm 1.98\%$
NT-Xent - $\tau = 0.1$	$53.08 \pm 1.68\%$	$52.62 \pm 2.29\%$
SupCon - $\tau = 0.1$	$52.62 \pm 2.97\%$	$52.40 \pm 3.27\%$

Table 4.4: Comparison between the baseline and different smooth contrastive losses with the best value of temperature (τ) in terms of UA given from an SVM

reason for restricting the batch to contain only one sample per class was that the number of classes was large and thus computationally prohibitive to form the batch. We experiment with the same values of τ as previous {0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10} and we evaluate the representations with the same manner.

We are also experimenting with the Supervised Contrastive (SupCon) loss presented in [2] where both multiple negative and multiple positive samples are used to calculate the loss and the summing is performed outside the logarithm

$$\mathcal{L}_{out}^{\text{sup}} = \sum_{a \in A} \frac{1}{|P(a)|} \sum_{p \in P(a)} -\log \left(\frac{e^{\mathbf{z}_a \cdot \mathbf{z}_p / \tau}}{\sum_{p \in P(a)} e^{\mathbf{z}_a \cdot \mathbf{z}_p / \tau} + \sum_{n \in N(a)} e^{\mathbf{z}_a \cdot \mathbf{z}_n / \tau}} \right)$$

Note that in this experiment no augmentations are used as in the original paper and hence no "multi-view" batch is formed. We only increase the number of positives required to compute the denumerator. The values of the temperature τ that are used are as previous {0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2, 5, 10} and the representations are evaluated with the same manner.

We examine the effect of the temperature parameter τ for all the contrastive losses (Smooth Triplet, NT-Xent, SupCon) in a common plot (figure 4.8).

The best models in terms of UA for each contrastive loss is also presented in table 4.4

We proceed with fine-tuning a linear classifier g on top of the encoder f. The whole network is trained for 20 epochs. The first 10 epochs the encoder f is kept frozen and then the whole network is trained together in an end-to-end fashion with Cross Entropy loss. We start with a lower learning rate of 0.0001 and the rest of the optimization parameters are the same with our baseline method. Instead of the linear SVM that used before for evaluating the representations, we now measure the UA using the linear classifier g that was used for fine-tuning the encoder f. The results are shown in 4.5

4.4.2.4 Augmentation based positive samples

As in the original proposal of [2], a "multi-viewed batch" is formed. For a set of N randomly sampled sample/label pairs, $A = \{(x_k, y_k), k = 1...N\}$, the corresponding batch used for training consists of $2N$ pairs, $\tilde{A} = \{(\tilde{x}_i, \tilde{y}_i), i = 1...2N\}$, where x_{2k} and x_{2k-1} are two random augmentations (a.k.a., "views") of x_k with $(k = 1...N)$ and $y_{2k} = y_{2k-1} = y_k$. The same formulation is used for computing the loss but using augmented samples from the "multiviewed batch"

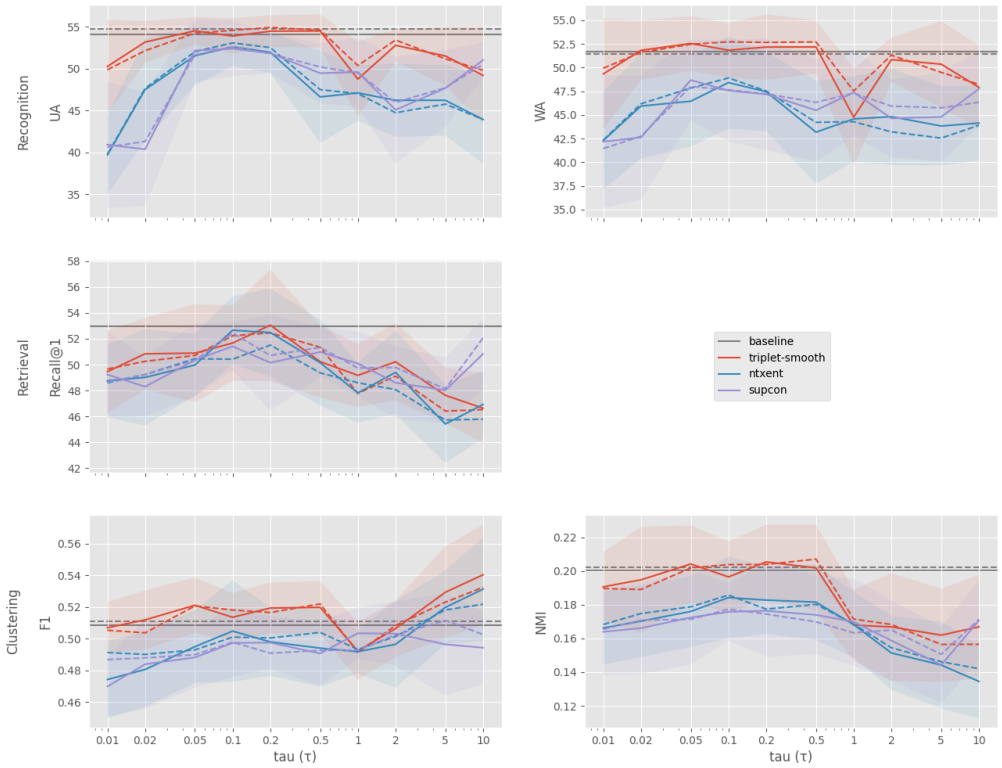


Figure 4.8: Effect of *temperature* (τ) parameter for each loss

Loss	Validation	Test
Cross-Entropy	$54.79 \pm 2.76\%$	$54.07 \pm 4.03\%$
Triplet Smooth - $\tau = 0.5$	$54.68 \pm 2.42\%$	$55.32 \pm 2.51\%$
NT-Xent - $\tau = 0.1$	$54.21 \pm 2.23\%$	$55.31 \pm 2.43\%$
SupCon - $\tau = 0.1$	$\mathbf{56.4} \pm 2.35\%$	$\mathbf{56.36} \pm 2.93\%$

Table 4.5: Comparison between the baseline and different values of temperature (τ) for different smooth contrastive losses in terms of UA after fine-tuning the network with Cross-Entropy. Fine-tuning the encoder f with a linear classifier g and training with cross-entropy increased the performance by $1 - 4\%$ (comparing with the linear SVMs). The fine-tuned encoder (SupCon loss pre-trained) exceeds the performance of the baseline by 2%

Abbreviation	Description
ce	Cross-Entropy
scl	Supervised Contrastive Loss (here: SupCon with $\tau = 0.1$)
svm	Linear SVM is fitted
clean	No augmentations used
aug	Augmentations used
multi	"Multi-view" batches
ft	Fine-Tuning
$\rightarrow$	The right operation follows the left operation as a separate step

Table 4.6: Abbreviation and notations explained

Method	Validation	Test
ce (aug)	$\mathbf{59.44} \pm 3.15\%$	$\mathbf{58.82} \pm 4.01\%$
scl (multi-aug) $\rightarrow$ svm (clean)	$51.99 \pm 3.38\%$	$52.15 \pm 2.93\%$
scl (multi-aug) $\rightarrow$ ce (ft-clean)	$55.82 \pm 3.00\%$	$54.4 \pm 2.04\%$
scl (multi-aug) $\rightarrow$ ce (ft-aug)	$55.88 \pm 2.32\%$	$55.56 \pm 2.29\%$

Table 4.7: Augmentation results

$$\mathcal{L}_{out}^{\text{sup}} = \sum_{a \in \tilde{A}} \frac{1}{|P(a)|} \sum_{p \in P(a)} -\log \left(\frac{e^{\mathbf{z}_a \cdot \mathbf{z}_p / \tau}}{\sum_{p \in P(a)} e^{\mathbf{z}_a \cdot \mathbf{z}_p / \tau} + \sum_{n \in N(a)} e^{\mathbf{z}_a \cdot \mathbf{z}_n / \tau}} \right)$$

The augmentations that are employed in this experiment are time and frequency masking following the work of [4]. For time masking a consecutive piece up to 5% of the total time length of the spectrogram is zero-ed out. For frequency masking a consecutive piece up to 10 frequency bins is zero-ed out.
In order to provide fair comparisons, a version of baseline using augmentations is trained and evaluated. The representations of the encoder trained with "multi-viewed batches" are evaluated as before. A linear SVM is fitted on top of the encoder and a linear classifier is fine-tuned along with the encoder. The fine-tuning is performed using the clean data and augmented data. The results are shown in table 4.7. A summary of the abbreviations that were used in the table 4.7 can be found in 4.6. As it can be ssen from the table, the best results were given by pre-training with a supervised contrastive loss using multi-view batches and fine-tuning with cross-entropy using data augmentations. Still, the UA is far from the baseline with augmentations.

4.4.3 Combining Cross Entropy and Supervised Contrastive signals

Since there is an improvement when fine-tuning the encoder when training a linear classifier on top of it, we attempt to apply both losses (supervised contrastive and cross entropy) simultaneously (see figure 4.9). We expect that this strategy will both encourage the encoder to learn a structure of the space and aid the linear classifier to achieve better generalizability. For the following set of experiments the supervised contrastive loss $\mathcal{L}_{sc}$ is used as an auxiliary loss to the primary Cross Entropy loss $\mathcal{L}_{ce}$

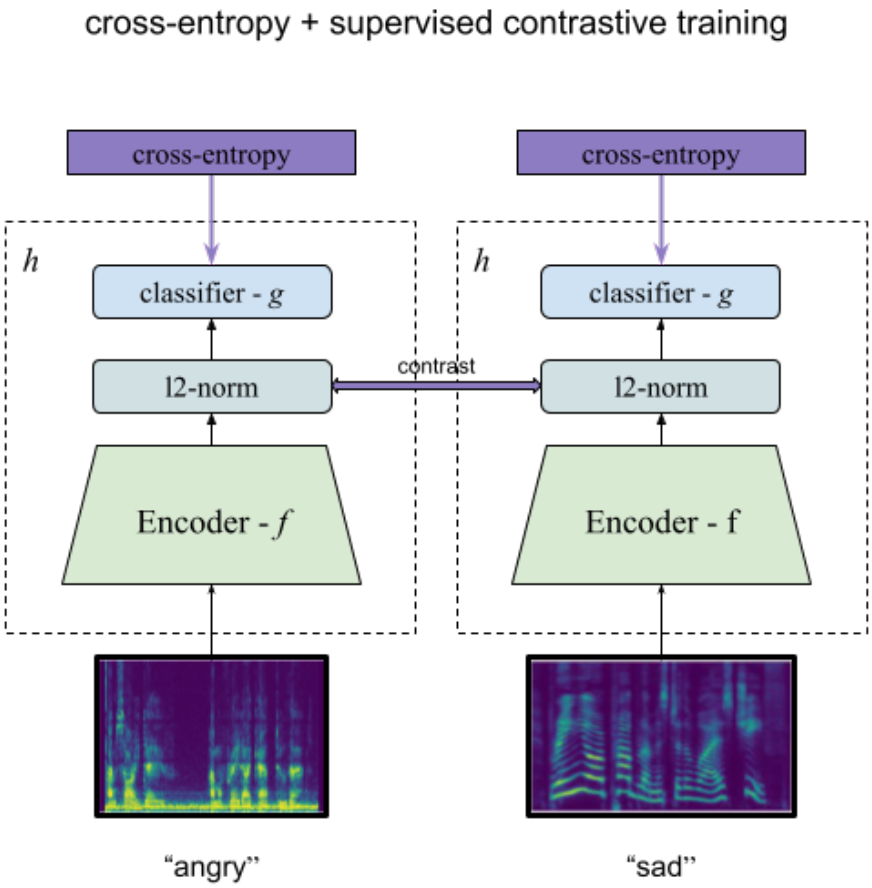


Figure 4.9: The procedure of supervised contrastive training an encoder f along with cross-entropy loss applied on the whole network h.

λ	Validation	Test
0.01	$44.17 \pm 5.01\%$	$42.50 \pm 5.44\%$
0.1	$44.17 \pm 5.01\%$	$42.50 \pm 5.44\%$
0.5	$\mathbf{55.44} \pm 3.47\%$	$\mathbf{53.23} \pm 4.94\%$
1	$\mathbf{54.79} \pm 2.40\%$	$\mathbf{53.60} \pm 3.34\%$
2	$52.13 \pm 2.36\%$	$51.91 \pm 3.92\%$

Table 4.8: Comparison between different values of weighting (λ) of the auxiliary supervised contrastive loss (here: SupCon with $\tau = 0.1$) in terms of UA

weighted with a value $\lambda > 0$

$$\mathcal{L} = \mathcal{L}_{ce} + \lambda \mathcal{L}_{sc} \tag{4.1}$$

The chosen supervised contrastive loss is the same as before i.e. SupCon loss with $\tau = 0.1$. The learning rate is started from a lower value of 0.0001. The network is trained using augmentations with "multi-view" batches as described before. The initial batch size is set to 32 so the effective batch size is 64. All the other parameters of the optimization process remain the same with the baseline model. The results are shown in table 4.8. The best performance is obtained when the losses are equally weighted ($\lambda = 1$) but it is still no match for the baseline.

4.4.4 Self-Supervised contrastive (pre-)training

We proceed with adding a form of self-supervision in the training phase as a pre-training step. The self-supervised pre-training procedure is also instructed by a contrastive loss. In contrary with the previous sections, we form and calculate the loss without using the labels. We follow exactly the same procedure as in the original paper of NT-Xent loss [1] by forming "multi-view" batches and discarding the labels. We first tune the temperature by training the encoder only with self-supervision and then fitting a linear SVM on top of the learned representations. The temperature that gives the best performance in terms of UA is selected. There were no big discrepancies between the various values of the temperature, so for the rest of the experiments we set the temperature of self-supervised NT-Xent loss to $\tau = 0.5$. In order to further evaluate the possible added value of self-supervised pre-training we proceed by fine-tuning the encoder with supervised signals.

The fine-tuning methods we try are the following:

1. ce (ft-aug): Fine-tune an end-to-end classifier h by adding a linear head g on top of the learned encoder network keeping the augmentations using only the Cross Entropy loss $\mathcal{L}_{ce}$ (see figure 4.12)
2. scl (ft-multi-aug) $\rightarrow$ svm: Fine-tune the encoder f using a supervised contrastive loss $\mathcal{L}_{sc}$ and afterwards fit an SVM on top of the fine-tuned encoder (see figure 4.11)

The results are shown in table 4.9. It can be seen that having an encoder pre-trained with self-supervision and directly fine-tuning with cross-entropy end-to-end with augmentations leads to the best results improving upon the baseline +1% in terms of UA.

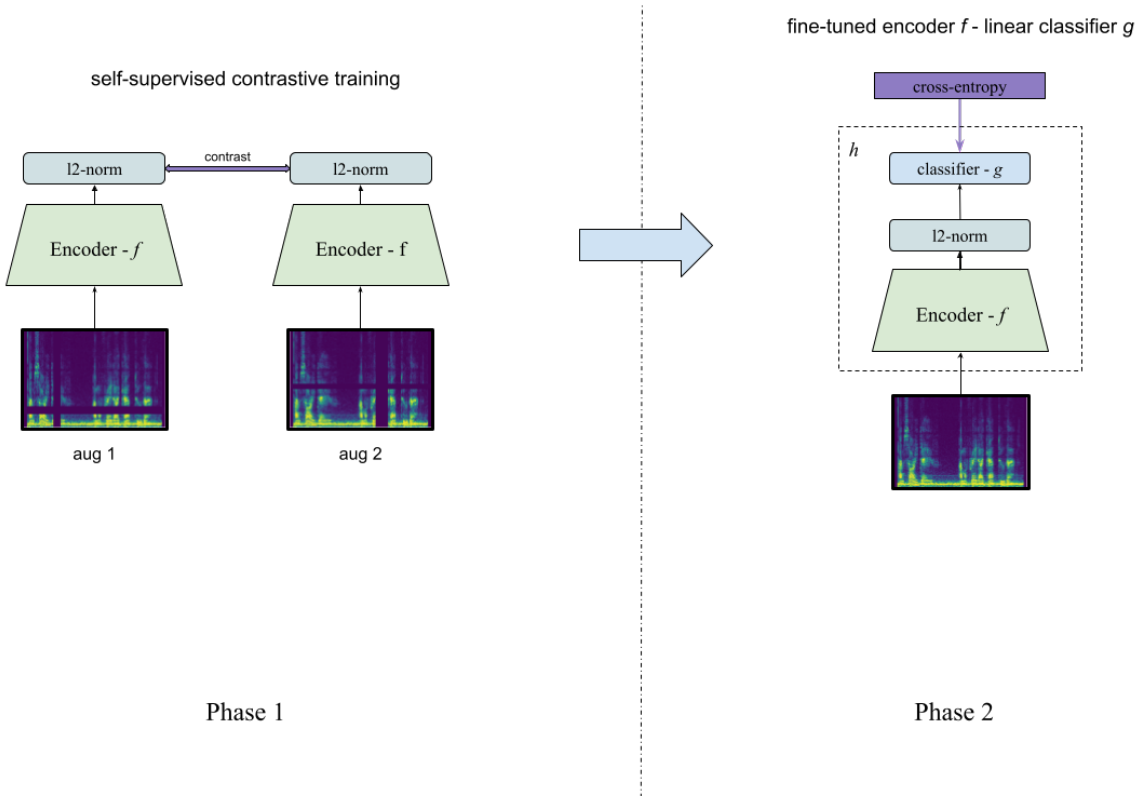


Figure 4.10: The procedure of self-supervised contrastive training an encoder f followed with fine-tuning the whole network h using cross-entropy loss

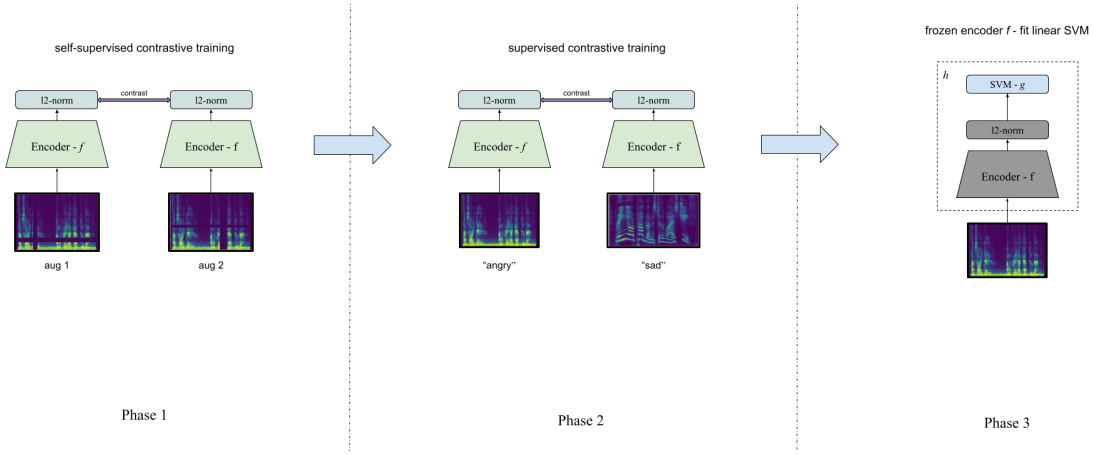


Figure 4.11: The procedure of self-supervised contrastive training an encoder f followed with fine-tuning the encoder f with supervised contrastive loss and then fitting a linear SVM g on top

Loss	Validation	Test
Cross-Entropy (aug)	$59.44 \pm 3.15\%$	$58.82 \pm 4.01\%$
sscl (pt) $\rightarrow$ ce (ft-aug)	$\mathbf{60.11} \pm 3.00\%$	$\mathbf{59.53} \pm 4.32\%$
sscl (pt) $\rightarrow$ scl (ft-multi-aug) $\rightarrow$ svm	$50.46 \pm 4.06\%$	$50.35 \pm 3.28\%$

Table 4.9: Results of classifiers on pre-trained with self-supervision model

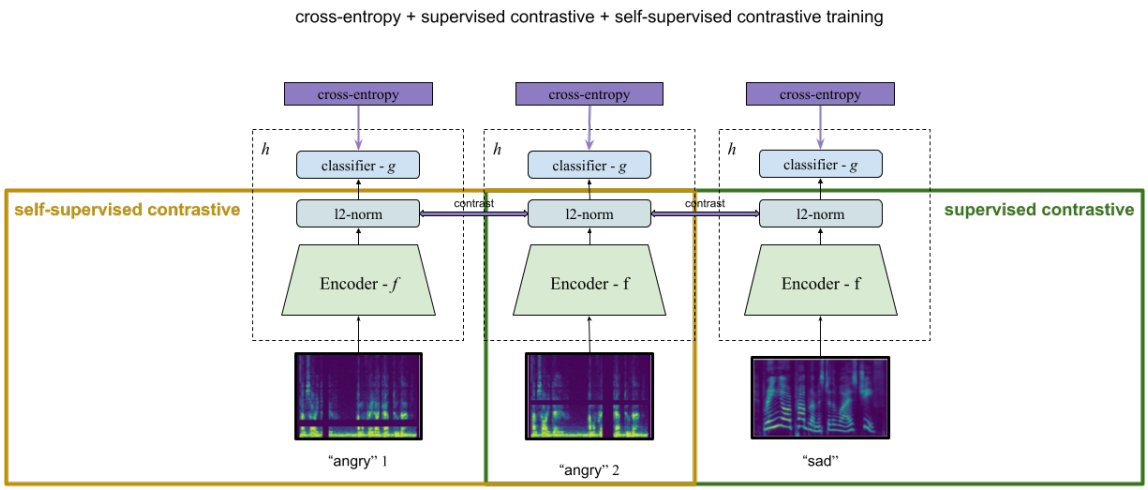


Figure 4.12: The procedure of combined self-supervised and supervised contrastive training an encoder f along with cross-entropy loss applied on the whole network h

β	Validation	Test
0.1	$53.01 \pm 3.26\%$	$52.01 \pm 3.47\%$
0.5	$54.85 \pm 2.76\%$	$54.13 \pm 3.84\%$
1	$50.30 \pm 5.32\%$	$49.54 \pm 4.67\%$

Table 4.10: Comparison between different values of weighting (β) of the self-supervised contrastive loss (here: NTXent with $\tau = 0.5$) in terms of UA

4.4.5 Combining supervised and self-supervised signals

Instead, of splitting the procedure into 2 steps (self-supervised pre-training and supervised fine-tuning), we attempt to combine them into a single training procedure (. We expect that the combination of the supervised and self-supervised losses will push the network into learning both class discriminative and class invariant features. The combined loss is the same with 4.1 with a weighted adding of a self-supervised loss $\mathcal{L}_{ssc}$

$$\mathcal{L} = \mathcal{L}_{ce} + \lambda \mathcal{L}_{sc} + \beta \mathcal{L}_{ssc} \tag{4.2}$$

The λ hyper-parameter is fixed to 1 which is the value that gave the best results in 4.4.3. We tune the β hyper-parameter by selecting the value between $\{0.1, 0.5, 1\}$. The results are shown in table. As we can see from the results, there is no benefit from mixing self-supervised and supervised signals in the same loss.

4.4.6 Comparison with other works

The performance estimation of a SER system can be done both in a Speaker Dependent (SD) way and in a Speaker Independent (SI) way. Speaker Dependent evaluation means that the speech signals which are used for testing have been produced from speakers that have also been part of the training procedure. Speaker Independent evaluation means that no information has been provided to the system during its training phase for the speakers that are used for testing. Since the approach of Speaker

Paper	Input	Other tasks	CV	UA (%)
Speech SimCLR [42]	MelSpec	ASR	5-fold	65.1
Lux22Arx [43]	w2v2 [47]	-	5-fold	67.2
Pep21IS [45]	w2v2	-	5-fold	67.2
SUPERB [49]	HuBERT [50]	-	5-fold	67.62
Zou22IC [46]	MFCCs,Spec,w2v2	-	5-fold	71.05
San21IS [51]	MFCC+CQT+F0	-	5-fold	75.9
Ours [sscl (pt) + ce (ft-aug)]	Spec	-	10-fold	59.53
Feng20IS [52]	MFCCs	ASR	10-fold	69.67
Light-SERNet [53]	MFCCs	-	10-fold	70.76
Zou22IC [46]	MFCCs,Spec,w2v2	-	10-fold	72.70

Table 4.11: Caption

Dependent analysis is the least frequent in real case scenarios (most probably you will not have any prior information about the speakers), we evaluate our system in a Speaker Independent manner. In the literature, there are two approaches for Speaker Independent evaluation of IEMOCAP dataset. The first approach is a 5-fold cross-validation strategy that is called Leave-One-Session-Out and for every fold 4 sessions are used for training and 1 session is used for testing. Since there are 5 total session it produces 5 speaker independent folds for cross-validation. The second approach is a 10-fold cross-validation strategy called Leave-One-Speaker-Out. For each fold 1 session is left-out which contains 2 speakers, 1 female and 1 male. One speaker is used as validation set and the other as test set. The speakers are swapped and the procedure is repeated. This is done for every session so 10 speaker independent folds are produced. We decided to go with the second Speaker Independent approach (Leave-One-Speaker-Out) because we believe is the most fair in estimating the validation and test scores. In table 4.11, there is a comparison with other works

Chapter 5

Conclusion and Future Work

This is the last section of our work where we draw the main conclusions and provide insights for further research to be conducted in the future.

5.1 Conclusions

As we have already stated in section 4.2, the goals of this work was to provide an investigation on the use of contrastive learning for the task of SER and find ways to increase a system's performance. The main conclusions of our work can be summarized as follows:

5.1.1 Conclusion 1

The use of supervised contrastive losses yields approximately the same performance as with the cross-entropy loss (sections 4.4.2, 4.4.3)

We have investigated four supervised contrastive learning losses (Triplet rough and smooth, NT-Xent and SupCon). None of them yielded better results than cross-entropy loss and the increase of positive and negative samples seems to have no effect on the performance. The most possible explanation is that the number of classes is small and the negative sampling is adequate even for small batch sizes.

5.1.2 Conclusion 2

Self-supervised pre-training of an encoder can improve the performance of the system (sections 4.4.4, 4.4.5)

The self-supervised pre-training helped the subsequent training with cross-entropy loss to increase its performance. Self-supervision helped the network identify and incorporate the invariances in the representation space which made the subsequent training converge to a more robust solution.

In every case, the contrastive learning seems to be more efficient as a pre-training step rather than an auxiliary task to classification (sections 4.4.3, 4.4.5)

5.2 Future work

The next steps in this area should be focused on the self-supervised methods of pre-training an encoder network in order to build a more robust representation space. Other methods that have been proposed in the literature can be applied on the specific task of SER such as MoCo [24], Barlow Twins [44] and Deep InfoMax (DIM) [54], Augmented Multiscale DIM (AMDIM) [55].

The type and the intensity of the augmentation is crucial for the success of the contrastive losses. We would like to conduct further experiments on data augmentation types and intensity on the spectrograms like [4]. Augmentations such as noise addition, pitch shifting, reverberation and speed altering could be applied directly on the raw waveform before converting it to a spectrogram.

We could also pre-train the network on a larger emotionally salient dataset such as CMU MOSI [56] and MOSEI [57] and MSP Podcast [58] and fine-tune on IEMOCAP to increase the performance [48].

The method should be also tested on state-of-the-art speech signal encoders from raw signal such as wav2vec 2.0 [47] and HuBERT [50].

Appendix A

Losses and Gradients

- Cross-Entropy

$$\ell_a = -\log\left(\frac{\exp(\mathbf{z}_a \cdot \mathbf{w}_{y_a})}{\sum_{i=1}^{C}\exp(\mathbf{z}_a \cdot \mathbf{w}_i)}\right)$$

$$\nabla_{\mathbf{z}_a}\ell_a = \frac{\sum_{k=1}^{C}(\mathbf{w}_k - \mathbf{w}_{y_a})\exp(\mathbf{z}_a \cdot \mathbf{w}_k)}{\sum_{i=1}^{C}\exp(\mathbf{z}_a \cdot \mathbf{w}_i)}$$

- Triplet

$$\ell_{a,p,n} = \max(0, \mathbf{z}_a \cdot \mathbf{z}_n - \mathbf{z}_a \cdot \mathbf{z}_p + m)$$

$$\nabla_{\mathbf{z}_a}\ell_{a,p,n} = \mathbf{z}_n - \mathbf{z}_p \text{ if } \mathbf{z}_p - \mathbf{z}_n < m \text{ else } \mathbf{0}$$

- Triplet (smooth)

$$\ell_{a,p,n} = -\log\left(\frac{\exp(\mathbf{z}_a \cdot \mathbf{z}_p/\tau)}{\exp(\mathbf{z}_a \cdot \mathbf{z}_p/\tau) + \exp(\mathbf{z}_a \cdot \mathbf{z}_n/\tau)}\right)$$

$$\nabla_{\mathbf{z}_a}\ell_{a,p} = -\frac{1}{\tau}[(1 - \frac{\exp(\mathbf{z}_a \cdot \mathbf{z}_p/\tau)}{Z})\mathbf{z}_p - \frac{\exp(\mathbf{z}_a \cdot \mathbf{z}_n/\tau))}{Z}\mathbf{z}_n]$$

- NT-Xent

$$\ell_{a,p} = -\log\left(\frac{\exp(\mathbf{z}_a \cdot \mathbf{z}_p/\tau)}{\exp(\mathbf{z}_a \cdot \mathbf{z}_p/\tau) + \sum_{n\in N(a)}\exp(\mathbf{z}_a \cdot \mathbf{z}_n/\tau)}\right)$$

$$\nabla_{\mathbf{z}_a}\ell_{a,p} = -\frac{1}{\tau}[(1 - \frac{\exp(\mathbf{z}_a \cdot \mathbf{z}_p/\tau)}{Z})\mathbf{z}_p - \sum_{n\in N(a)}\frac{\exp(\mathbf{z}_a \cdot \mathbf{z}_n/\tau))}{Z}\mathbf{z}_n]$$

- SupCon

$$\ell_{a,p} = -\log\left(\frac{\exp(\mathbf{z}_a \cdot \mathbf{z}_p/\tau)}{\sum_{p\in P(a)}\exp(\mathbf{z}_a \cdot \mathbf{z}_p/\tau) + \sum_{n\in N(a)}\exp(\mathbf{z}_a \cdot \mathbf{z}_n/\tau)}\right)$$

Bibliography

[1] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," 2020.

[2] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," 2021.

[3] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2016, pp. 1857–1865.

[4] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *Proc. Interspeech 2019*, pp. 2613–2617, 2019.

[5] R. Plutchik, "A general psychoevolutionary theory of emotion," in *Theories of emotion*. Elsevier, 1980, pp. 3–33.

[6] J. A. Russell, "A circumplex model of affect." *Journal of personality and social psychology*, vol. 39, no. 6, p. 1161, 1980.

[7] T. M. Mitchell, *Machine learning*, vol. 1, no. 9.

[8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[9] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," California Univ San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," in *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014.

[12] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014. [Online]. Available: https://arxiv.org/abs/1409.0473

[13] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[15] J. Goldberger, G. E. Hinton, S. Roweis, and R. R. Salakhutdinov, "Neighbourhood components analysis," *Advances in neural information processing systems*, vol. 17, 2004.

[16] K. Q. Weinberger, J. Blitzer, and L. Saul, "Distance metric learning for large margin nearest neighbor classification," *Advances in neural information processing systems*, vol. 18, 2005.

[17] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a" siamese" time delay neural network," *Advances in neural information processing systems*, vol. 6, 1993.

[18] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *International workshop on similarity-based pattern recognition*. Springer, 2015, pp. 84–92.

[19] G. Chechik, V. Sharma, U. Shalit, and S. Bengio, "Large scale online learning of image similarity through ranking." *Journal of Machine Learning Research*, vol. 11, no. 3, 2010.

[20] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, "Learning fine-grained image similarity with deep ranking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1386–1393.

[21] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2015. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2015.7298682

[22] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," 2019.

[23] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," 2019.

[24] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," 2019.

[25] O. J. Hénaff, A. Srinivas, J. D. Fauw, A. Razavi, C. Doersch, S. M. A. Eslami, and A. van den Oord, "Data-efficient image recognition with contrastive predictive coding," 2019.

[26] C.-N. Anagnostopoulos, T. Iliou, and I. Giannoukos, "Features and classifiers for emotion recognition from speech: a survey from 2000 to 2011," *Artificial Intelligence Review*, vol. 43, no. 2, pp. 155–177, 2015.

[27] S. G. Koolagudi and K. S. Rao, "Emotion recognition from speech: a review," *International journal of speech technology*, vol. 15, no. 2, pp. 99–117, 2012.

[28] S. Mirsamadi, E. Barsoum, and C. Zhang, "Automatic speech emotion recognition using recurrent neural networks with local attention," in *2017 IEEE International*

conference on acoustics, speech and signal processing (ICASSP). IEEE, 2017, pp. 2227–2231.

[29] G. Ramet, P. N. Garner, M. Baeriswyl, and A. Lazaridis, "Context-aware attention mechanism for speech emotion recognition," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 126–131.

[30] M. Neumann and N. T. Vu, "Attentive convolutional neural network based speech emotion recognition: A study on the impact of input features, signal length, and acted speech," *Proc. Interspeech 2017*, pp. 1263–1267, 2017.

[31] R. Li, Z. Wu, J. Jia, S. Zhao, and H. Meng, "Dilated residual network with multi-head self-attention for speech emotion recognition," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6675–6679.

[32] L. Tarantino, P. N. Garner, A. Lazaridis *et al.*, "Self-attention for speech emotion recognition." in *Interspeech*, 2019, pp. 2578–2582.

[33] C. Etienne, G. Fidanza, A. Petrovskii, L. Devillers, and B. Schmauch, "Cnn+ lstm architecture for speech emotion recognition with data augmentation," in *Proc. Workshop on Speech, Music and Mind 2018*, 2018, pp. 21–25.

[34] Y. Zhang, J. Du, Z. Wang, J. Zhang, and Y. Tu, "Attention based fully convolutional network for speech emotion recognition," in *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2018, pp. 1771–1775.

[35] Y. Li, T. Zhao, and T. Kawahara, "Improved end-to-end speech emotion recognition using self attention mechanism and multitask learning," 09 2019, pp. 2803–2807.

[36] Z. Aldeneh and E. M. Provost, "Using regional saliency for speech emotion recognition," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 2741–2745.

[37] A. Nediyanchath, P. Paramasivam, and P. Yenigalla, "Multi-head attention for speech emotion recognition with auxiliary learning of gender recognition," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7179–7183.

[38] G. Trigeorgis, F. Ringeval, R. Brueckner, E. Marchi, M. A. Nicolaou, B. Schuller, and S. Zafeiriou, "Adieu features? end-to-end speech emotion recognition using a deep convolutional recurrent network," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 5200–5204.

[39] M. Sarma, P. Ghahremani, D. Povey, N. K. Goel, K. K. Sarma, and N. Dehak, "Emotion identification from raw speech signals using dnns." in *Interspeech*, 2018, pp. 3097–3101.

[40] S. Latif, R. Rana, S. Khalifa, R. Jurdak, and J. Epps, "Direct Modelling of Speech Emotion from Raw Speech," in *Proc. Interspeech 2019*, 2019, pp. 3920–3924. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-3252

[41] Z. Zhang, B. Wu, and B. Schuller, "Attention-augmented end-to-end multi-task learning for emotion prediction from speech," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6705–6709.

[42] D. Jiang, W. Li, M. Cao, W. Zou, and X. Li, "Speech SimCLR: Combining Contrastive and Reconstruction Objective for Self-Supervised Speech Representation Learning," in *Proc. Interspeech 2021*, 2021, pp. 1544–1548.

[43] F. Lux, C.-Y. Chen, and N. T. Vu, "Combining contrastive and non-contrastive losses for fine-tuning pretrained models in speech analysis," *arXiv preprint arXiv:2211.01964*, 2022.

[44] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," in *International Conference on Machine Learning*. PMLR, 2021, pp. 12 310–12 320.

[45] L. Pepino, P. Riera, and L. Ferrer, "Emotion Recognition from Speech Using wav2vec 2.0 Embeddings," in *Proc. Interspeech 2021*, 2021, pp. 3400–3404.

[46] H. Zou, Y. Si, C. Chen, D. Rajan, and E. S. Chng, "Speech emotion recognition with co-attention based multi-level acoustic information," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 7367–7371.

[47] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 449–12 460, 2020.

[48] C. Busso, M. Bulut, C.-C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. N. Chang, S. Lee, and S. S. Narayanan, "Iemocap: Interactive emotional dyadic motion capture database," *Language resources and evaluation*, vol. 42, no. 4, pp. 335–359, 2008.

[49] S. wen Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin, T.-H. Huang, W.-C. Tseng, K. tik Lee, D.-R. Liu, Z. Huang, S. Dong, S.-W. Li, S. Watanabe, A. Mohamed, and H. yi Lee, "SUPERB: Speech Processing Universal PERformance Benchmark," in *Proc. Interspeech 2021*, 2021, pp. 1194–1198.

[50] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[51] J. Santoso, T. Yamada, S. Makino, K. Ishizuka, and T. Hiramura, "Speech Emotion Recognition Based on Attention Weight Correction Using Word-Level Confidence Measure," in *Proc. Interspeech 2021*, 2021, pp. 1947–1951.

[52] H. Feng, S. Ueno, and T. Kawahara, "End-to-End Speech Emotion Recognition Combined with Acoustic-to-Word ASR Model," in *Proc. Interspeech 2020*, 2020, pp. 501–505.

[53] A. Aftab, A. Morsali, S. Ghaemmaghami, and B. Champagne, "Light-sernet: A lightweight fully convolutional neural network for speech emotion recognition," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 6912–6916.

[54] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio, "Learning deep representations by mutual information estimation and maximization," in *International Conference on Learning Representations*, 2018.

[55] P. Bachman, R. D. Hjelm, and W. Buchwalter, "Learning representations by maximizing mutual information across views," *Advances in neural information processing systems*, vol. 32, 2019.

[56] A. Zadeh, R. Zellers, E. Pincus, and L.-P. Morency, "Mosi: multimodal corpus of sentiment intensity and subjectivity analysis in online opinion videos," *arXiv preprint arXiv:1606.06259*, 2016.

[57] A. Zadeh and P. Pu, "Multimodal language analysis in the wild: Cmu-mosei dataset and interpretable dynamic fusion graph," in *Proceedings of the 56th annual meeting of the association for computational linguistics (Long Papers)*, 2018.

[58] L. Martinez-Lucas, M. Abdelwahab, and C. Busso, "The msp-conversation corpus," *Interspeech 2020*, 2020.