



Εθνικό Μετσόβιο Πολυτεχνείο

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Επεξεργασία Φωνής και Φυσικής Γλώσσας, 2024-2025

Αναφορά 2ου Εργαστηρίου: Κατηγοριοποίηση κειμένων με Deep Neural Networks - DNN

Όνομα:	Γεώργιος
Επώνυμο:	Οικονόμου
ΑΜ:	03121103

Εισαγωγή

Ο κώδικας, συνοδευόμενος από σύντομα σχόλια, παρουσιάζεται στο ακόλουθο [αποθετήριο Git](#).

Προπαρασκευή 2ου Εργαστηρίου

Για τις ανάγκες του εργαστηρίου, χρησιμοποιήθηκε το εργαλείο Conda με Python 3.8, και το αποθετήριο Git του εργαστηρίου εγκαταστάθηκε με επιτυχία. Το περιβάλλον Conda που χρησιμοποιείται είναι το slp2. Τέλος, εγκαταστάθηκαν τα διανύσματα λέξεων Glove 6B και Glove Twitter στον φάκελο /embeddings.

Προεπεξεργασία Δεδομένων

Κωδικοποίηση Επισημειώσεων (Labels)

Χρησιμοποιήθηκε ο LabelEncoder της βιβλιοθήκης scikit-learn και κωδικοποιήθηκε κάθε κλάση ώστε να αντιστοιχεί σε έναν συγκεκριμένο αριθμό. Είναι αναγκαίο ο LabelEncoder να έχει πρώτα εκπαιδευτεί με `.fit()` στα δεδομένα του train set για να μετατρέψει τα labels με `.transform()`.

```
##### EX1 # MR #####
```

```
positive 1
positive 1
positive 1
positive 1
positive 1
positive 1
positive 1
positive 1
positive 1
positive 1
```

```
##### EX1 # Semeval2017A #####
```

```
neutral 1
positive 2
neutral 1
positive 2
positive 2
positive 2
positive 2
neutral 1
positive 2
negative 0
neutral 1
```

Οπότε, στο dataset του Semeval 2017 Task4-A οι κλάσεις {negative, neutral, positive} κωδικοποιούνται στους αριθμούς {0, 1, 2} και στο dataset του MR οι κλάσεις {negative, positive} κωδικοποιούνται στους αριθμούς {1, 2}.

Λεκτική Ανάλυση (Tokenization)

Χρησιμοποιήθηκε η κλάση SentenceDataset στο αρχείο dataloading.py και διαχωρίζει μια πρόταση σε ξεχωριστές λέξεις (tokens) βάση του χαρακτήρα του κενού. Είναι αναγκαίο να εκτελεστεί η εντολή nltk.download('punkt') για να κατέβει το προ-εκπαιδευμένο μοντέλο "punkt" από το NLTK, το οποίο είναι απαραίτητο για το tokenization των προτάσεων σε λέξεις.

```
##### EX2 # MR #####
```

```
[['the', 'rock', 'is', 'destined', 'to', 'be', 'the', '21st', 'century', 's', 'new', 'conan', 'and', 'that', 'he', 's', 'going', 'to', 'make', 'a', 'splash', 'even', 'greater', 'than', 'arnold', 'schwarzenegger', 'jean-claud', 'van', 'damme', 'or', 'steven', 'segal', 'the', 'gorgeously', 'elaborate', 'continuation', 'of', 'the', 'lord', 'of', 'the', 'rings', 'trilogy', 'is', 'so', 'huge', 'that', 'a', 'column', 'of', 'words', 'can', 'not', 'adequately', 'describe', 'co-writer/director', 'peter', 'jackson', 's', 'expanded', 'vision', 'of', 'j', 'r', 'r', 'tolkien', 's', 'middle-earth', 'effective', 'but', 'too-tepid', 'biopic'], ['if', 'you', 'sometim', 'es', 'like', 'to', 'go', 'to', 'the', 'movies', 'to', 'have', 'fun', 'wasabi', 'is', 'a', 'good', 'place', 'to', 's', 'tart', ''], ['emerges', 'as', 'something', 'rare', 'an', 'issue', 'movie', 'that', 's', 'so', 'honest', 'and', 'k', 'eenly', 'observed', 'that', 'it', 'does', 'n't', 'feel', 'like', 'one', 'the', 'film', 'provides', 'some', 'great', 'insight', 'into', 'the', 'neurotic', 'mindset', 'of', 'all', 'comics', 'even', 'those', 'who', 'have', 'reache', 'd', 'the', 'absolute', 'top', 'of', 'the', 'game', ''], ['offers', 'that', 'rare', 'combination', 'of', 'entertainment', 'and', 'education', ''], ['perhaps', 'no', 'picture', 'ever', 'made', 'has', 'more', 'literally', 'showed', 'that', 'the', 'road', 'to', 'hell', 'is', 'paved', 'with', 'good', 'intentions', ''], ['steers', 'turns', 'in', 'a', 'snappy', 'screenplay', 'that', 'curls', 'at', 'the', 'edges', 'it', 's', 'so', 'clever', 'you', 'want', 'to', 'hate', 'it', 'but', 'he', 'somehow', 'pulls', 'it', 'off', ''], ['take', 'care', 'of', 'my', 'cat', 'offers', 'a', 'refreshing', 'ly', 'different', 'slice', 'of', 'asian', 'cinema', ']]
```

```
##### EX2 # Semeval2017A #####
```

```
[['05', 'Beat', 'it', '-', 'Michael', 'Jackson', '-', 'Thriller', '(', '25th', 'Anniversary', 'Edition', ')', '['', 'HD', ''], 'http', ':', '//t.co/A4k2B86PBv'], ['Jay', 'Z', 'joins', 'Instagram', 'with', 'nostalgic', 'tribute', 'to', 'Micha', 'el', 'Jackson', ':', 'Jay', 'Z', 'apparently', 'joined', 'Instagram', 'on', 'Saturday', 'and', 'http', ':', '//t.c', 'o/0j9I4eCvXy'], ['Michael', 'Jackson', ':', 'Bad', '25th', 'Anniversary', 'Edition', '(', 'Picture', 'Vinyl', ')', ':', 'This', 'unique', 'picture', 'disc', 'vinyl', 'includes', 'the', 'original', '1', 'http', ':', '//t.co/fkXhToAAuW'], ['I', 'liked', 'a', '@', 'YouTube', 'video', 'http', ':', '//t.co/AaR3pjp2PI', 'One', 'Direction', 'singing', 'Man', 'in', 'the', 'Mirror', 'by', 'Michael', 'Jackson', 'in', 'Atlanta', 'GA', 'June', '26', '18th', 'anniv', 'of', 'Princess', 'Diana', 's', 'death', 'I', 'still', 'want', 'to', 'believe', 'she', 'is', 'living', 'on', 'a', 'private', 'island', 'away', 'from', 'the', 'public', 'With', 'Michael', 'Jackson', ''], ['@', 'oridaganja', 'zz', 'The', '1st', 'time', 'I', 'heard', 'Michael', 'Jackson', 'sing', 'was', 'in', 'Honolulu', 'Hawaii', '@', 'a', 'restaurant', 'on', 'radio', 'It', 'was', 'A.B.C', 'I', 'was', '13', 'I', 'loved', 'it', ''], ['Michae', 'l', 'Jackson', 'appeared', 'on', 'Saturday', '29', 'at', 'the', '9th', 'place', 'in', 'the', 'Top20', 'of', 'Miami', 's', 'Trends', 'http', ':', '//t.co/dXN2FWgUhb', '#', 'trndnl'], ['Are', 'you', 'old', 'enough', 'to', 'remember', 'Michael', 'Jackson', 'attending', 'the', 'Grammys', 'with', 'Brooke', 'Shields', 'and', 'Webster', 'sat', 'on', 'his', 'lap', 'during', 'the', 'show', '?'], ['@', 'etbowser', 'do', 'u', 'enjoy', 'his', '2nd', 'rate', 'Michael', 'Jackson', 'bit', '?', 'Honest', 'ques', 'Like', 'the', 'ca', 'n't', 'feel', 'face', 'song', 'but', 'god', 'it', 's', 'so', 'obvious', 'they', 'want', 'MJ', '2.0'], ['The', 'Weeknd', 'is', 'the', 'closest', 'thing', 'we', 'may', 'get', 'to', 'Michael', 'Jackson', 'for', 'a', 'long', 'time', '...', 'especially', 'since', 'he', 'damn', 'near', 'mimics', 'everyth', 'ing']]
```

Κωδικοποίηση Παραδειγμάτων (Λέξεων)

Υλοποιήθηκε περαιτέρω η μέθοδος __getitem__ της κλάσης SentenceDataset με μέγιστο μήκος πρότασης 25. Μετατρέπει το κείμενο σε αριθμούς δείκτες μέσω του λεξικού word2idx, επιστρέφει τη κλάση (label) και το πραγματικό μήκος της πρότασης. Το μήκος της πρότασης προσαρμόζεται στο μέγιστο μήκος με μηδενικά ή κόβοντας το κείμενο.

```
##### EX3 # MR #####

Original Sentence: the rock is destined to be the 21st century's new " conan " and that he's going to make a splash even
greater than arnold schwarzenegger , jean-claud van damme or steven segal .
Tokenized Sentence: [ 1 1138 15 10454 5 31 1 5034 590 10 51 29
18513 29 6 13 19 10 223 5 160 8 16807 152
1414]
Label: 1
Length: 36

Original Sentence: the gorgeously elaborate continuation of " the lord of the rings " trilogy is so huge that a column o
f words cannot adequately describe co-writer/director peter jackson's expanded vision of j . r . r . tolkien's middle-ea
rth .
Tokenized Sentence: [ 1 78616 5135 10117 4 29 1 2371 4 1 6820 29
12305 15 101 1325 13 8 3236 4 1375 87 37 12424
4467]
Label: 1
Length: 42

Original Sentence: effective but too-tepid biopic
Tokenized Sentence: [ 2038 35 400001 34277 0 0 0 0 0 0
0 0 0 0 0 0
0 0 0 0 0]
Label: 1
Length: 4

Original Sentence: if you sometimes like to go to the movies to have fun , wasabi is a good place to start .
Tokenized Sentence: [ 84 82 1072 118 5 243 5 1 2460 5 34 2906
2 66408 15 8 220 242 5 466 3 0 0 0
0]
Label: 1
Length: 21

Original Sentence: emerges as something rare , an issue movie that's so honest and keenly observed that it doesn't feel
like one .
Tokenized Sentence: [12398 20 646 2349 2 30 496 1006 13 10 101 6082
6 23499 4583 13 21 261 71 999 118 49 3 0
0]
Label: 1
Length: 23

##### EX3 # Semeval2017A #####

Original Sentence: 05 Beat it - Michael Jackson - Thriller (25th Anniversary Edition) [HD] http://t.co/A4K2B86PBv
Tokenized Sentence: [ 17261 400001 21 12 400001 400001 12 400001 24 8962
400001 400001 25 2824 400001 5281 33162 46 400001 0
0 0 0 0 0]
Label: 1
Length: 19

Original Sentence: Jay Z joins Instagram with nostalgic tribute to Michael Jackson: Jay Z apparently joined Instagram on
Saturday and.. http://t.co/Qj9I4eCvXy
Tokenized Sentence: [400001 400001 7698 400001 18 20557 5079 5 400001 400001
46 400001 400001 1897 1031 400001 14 400001 6 400001
33162 46 400001 0 0]
Label: 2
Length: 23

Original Sentence: Michael Jackson: Bad 25th Anniversary Edition (Picture Vinyl): This unique picture disc vinyl include
s the original 1 http://t.co/fKXhToAAuW
Tokenized Sentence: [400001 400001 46 400001 8962 400001 400001 24 400001 400001
25 46 400001 3007 1836 5977 11193 1013 1 930
177 33162 46 400001 0]
Label: 1
Length: 24

Original Sentence: I liked a @YouTube video http://t.co/AaR3pjp2PI One Direction singing "Man in the Mirror" by Michael
Jackson in Atlanta, GA [June 26,
Tokenized Sentence: [400001 5573 8 17528 400001 975 33162 46 400001 400001
400001 4100 29 400001 7 1 400001 28 22 400001
400001 7 400001 2 400001]
Label: 2
Length: 29

Original Sentence: 18th anniv of Princess Diana's death. I still want to believe she is living on a private island away
from the public. With Michael Jackson.
Tokenized Sentence: [ 4014 400001 4 400001 400001 10 337 3 400001 150
304 5 734 68 15 757 14 8 673 584
421 26 1 199 3]
Label: 2
Length: 29
```

Μοντέλο

Embedding Layer

Δημιουργείται ένα embedding layer, το οποίο αναπαριστά τις λέξεις σε έναν πολυδιάστατο χώρο, με τέτοιο τρόπο ώστε οι σημασιολογικά παρόμοιες λέξεις να τοποθετούνται κοντά η μία στην άλλη. Τα βάρη του embedding layer αρχικοποιούνται χρησιμοποιώντας τα

προεκπαιδευμένα word embeddings GloVe (glove.6B.50d), τα οποία έχουν διάσταση 50 (50-dimensional embeddings). Τα βάρη δεν θα ενημερωθούν περαιτέρω κατά την εκπαίδευση του μοντέλου.

Γιατί αρχικοποιούμε το embedding layer με τα προ-εκπαιδευμένα word embeddings?

Το embedding layer αρχικοποιείται με τα προ-εκπαιδευμένα word embeddings καθώς αυτά αναπαριστούν ήδη τις σημασιολογικά παρόμοιες λέξεις κοντά τη μία στην άλλη στον πολυδιάστατο χώρο. Με άλλα λόγια, τα προεκπαιδευμένα word embeddings έχουν ήδη προσαρμοστεί κατάλληλα πάνω σε δεδομένα και το μοντέλο ξεκινάει με καλύτερες αρχικές αναπαραστάσεις. Αντίθετα, οι τυχαίες τιμές απαιτούν περισσότερους υπολογιστικούς πόρους καθώς χρειάζονται εκπαίδευση από το μηδέν.

Γιατί κρατάμε παγωμένα τα βάρη του embedding layer κατά την εκπαίδευση?

Τα βάρη του embedding layer παραμένουν σταθερά κατά την εκπαίδευση, πράγμα που σημαίνει ότι δεν εκτελείται ο αλγόριθμος του backpropagation για αυτά. Αυτή η προσέγγιση βοηθά στην αποφυγή του overfitting. Με άλλα λόγια, είναι θεμιτό τα προεκπαιδευμένα word embeddings να μην προσαρμοστούν στο συγκεκριμένο πρόβλημα εκπαίδευσης αλλά να γενικευούν. Τέλος, μειώνει τον υπολογιστικό φόρτο, καθώς δεν χρειάζεται να εκπαιδεύσουμε τα embeddings από την αρχή.

Output Layer

Δημιουργείται το output layer του μοντέλου, το οποίο προβάλλει τις αναπαραστάσεις των κειμένων στον χώρο των κλάσεων. Ως μη γραμμική συνάρτηση ενεργοποίησης επιλέγεται η ReLU και ως διάσταση του hidden layer οι 512 νευρώνες. Τέλος, επιθυμητή είναι η αναζήτηση υπερπαραμέτρων (hyperparameter optimization) ώστε να βελτιστοποιηθεί η διάσταση και να επιλεγεί αυτή που έχει την καλύτερη απόδοση.

Γιατί βάζουμε μία μη γραμμική συνάρτηση ενεργοποίησης στο προτελευταίο layer; Τι διαφορά θα είχε αν είχαμε 2 ή περισσότερους γραμμικούς μετασχηματισμούς στη σειρά;

Οι μη γραμμικές συναρτήσεις ενεργοποίησης είναι θεμελιώδεις για την ικανότητά τους να μαθαίνουν και να μοντελοποιούν πολύπλοκες και σύνθετες σχέσεις στα δεδομένα. Χωρίς τη χρήση μη γραμμικών συναρτήσεων, όλοι οι γραμμικοί μετασχηματισμοί σε διαδοχικά στρώματα στη σειρά θα οδηγούσαν σε έναν μόνο γραμμικό μετασχηματισμό, περιορίζοντας τη δυνατότητα του μοντέλου να αναγνωρίσει μη γραμμικά μοτίβα.

Forward pass

Εκτελούνται οι απαραίτητοι μετασχηματισμοί για την εκτέλεση του forward propagation. Οι λέξεις προβάλλονται σε διανύσματα και δημιουργείται η αναπαράσταση κάθε πρότασης στον πολυδιάστατο χώρο, υπολογίζοντας τον μέσο όρο των word embeddings. Τέλος, η μη γραμμική συνάρτηση ReLU χρησιμοποιείται ώστε οι αναπαραστάσεις των προτάσεων να μετασχηματιστούν και να προβληθούν στο χώρο των κλάσεων.

Αν θεωρήσουμε ότι κάθε διάσταση του embedding χώρου αντιστοιχεί σε μία αφηρημένη έννοια, μπορείτε να δώσετε μία διαισθητική ερμηνεία για το τι περιγράφει η αναπαράσταση που φτιάξατε (κέντρο-βάρους);

Οι σημασιολογικά παρόμοιες λέξεις αναπαριστώνται κοντά η μία στην άλλη στον πολυδιάστατο χώρο. Ομοίως, αν κάθε διάσταση του χώρου αντιστοιχεί σε μια αφηρημένη

έννοια, η αναπαράσταση του κέντρου-βάρους δίνει την ευρύτερη έννοια της πρότασης στον πολυδιάστατο χώρο. Αν, παραδείγματος χάριν, η πρόταση έχει λέξεις που σχετίζονται με αρνητικά συναισθήματα, όπως αυτό της λύπης ή του θυμού, τότε το διάνυσμα της πρότασης θα έχει υψηλές τιμές στην διάσταση που αντιστοιχούν αφηρημένες έννοιες των αρνητικών συναισθημάτων. Συμπερασματικά, το κέντρο-βάρους δείχνει το γενικό νόημα της πρότασης.

Αναφέρετε πιθανές αδυναμίες της συγκεκριμένης προσέγγισης για να αναπαραστήσουμε κείμενα.

Η προσέγγιση αυτή δεν λαμβάνει υπόψη τη συντακτική δομή της πρότασης και τα σημεία στίξης ενώ επηρεάζεται εξίσου από ουδέτερα ή μη πληροφοριακά μέρη του λόγου, όπως άρθρα και σύνδεσμοι, γεγονός που μπορεί να υποβαθμίσει τη σημασιολογική ακρίβεια της αναπαράστασης.

Διαδικασία Εκπαίδευσης

Φόρτωση Παραδειγμάτων (DataLoaders)

Χρησιμοποιήθηκε η κλάση `DataLoader` στο αρχείο `main.py` προκειμένου τα `dataset` να διαχωριστούν σε `mini-batches` και να τροφοδοτούνται στο μοντέλο κατά την εκπαίδευση.

Τι συνέπειες έχουν τα μικρά και μεγάλα `mini-batches` στην εκπαίδευση των μοντέλων;

Η επιλογή του μεγέθους του `mini-batch` επηρεάζει σημαντικά την απόδοση και τη σύγκλιση του μοντέλου. Τα μικρά `mini-batches` προσφέρουν θορυβώδεις κλίσεις, βοηθώντας στη διαφυγή από τοπικά ελάχιστα και βελτιώνοντας τη γενίκευση, αλλά απαιτούν περισσότερες επαναλήψεις και έχουν χαμηλότερη απόδοση. Αντίθετα, τα μεγάλα `mini-batches` επιταχύνουν την εκπαίδευση ανά εποχή, αλλά μπορεί να χρειαστούν ρύθμιση του `learning rate` και να οδηγήσουν σε χειρότερη γενίκευση.

Συνήθως ανακατεύουμε την σειρά των `mini-batches` στα δεδομένα εκπαίδευσης σε κάθε εποχή. Μπορείτε να εξηγήσετε γιατί;

Το `shuffling` των `mini-batches` στα δεδομένα εκπαίδευσης σε κάθε εποχή εξυπηρετεί στην αποφυγή της υπερεκπαίδευσης σε συγκεκριμένη σειρά δεδομένων. Έτσι, το μοντέλο θα μπορεί να γενικεύει και να μην κάνει `overfitting`. Αν το μοντέλο βλέπει τα δεδομένα πάντα με την ίδια σειρά, μπορεί να μάθει μοτίβα που σχετίζονται με τη σειρά και όχι με τα ίδια τα χαρακτηριστικά των δεδομένων.

Βελτιστοποίηση

Χρησιμοποιήθηκε το μοντέλο `BaselineDNN` χρησιμοποιώντας τα προεκπαιδευμένα `word embeddings GloVe`. Ανάλογα με τον αριθμό των κλάσεων επιλέγεται η κατάλληλη συνάρτηση κόστους (`BCEWithLogitsLoss` για δυαδική ταξινόμηση, `CrossEntropyLoss` για περισσότερες κλάσεις). Τέλος, χρησιμοποιείται ο αλγόριθμος βελτιστοποίησης `Adam` με `learning rate 0.001`.

Εκπαίδευση

Χρησιμοποιήθηκε η συνάρτηση `train_dataset` και `eval_dataset` για να εκπαιδευτεί και αξιολογηθεί κάθε `mini-batch`. Για την περίπτωση δυαδικής ταξινόμησης, η πρόβλεψη γίνεται με βάση το πρόσημο του `logit` (`output > 0`), ώστε να συμβαδίζει με τη χρήση της συνάρτησης

BCEWithLogitsLoss. Διαφορετικά, η τελική πρόβλεψη προκύπτει με χρήση της `torch.argmax` στα logits του δικτύου.

Αξιολόγηση

Η ανάπτυξη του μοντέλου έχει ολοκληρωθεί. Έπειτα, αξιολογούνται οι επιδόσεις του μοντέλου στα δύο dataset Semeval 2017 Task4-A και MR με προεπιλεγμένη τιμή των 150 εποχών. Χρησιμοποιήθηκαν οι μετρικές accuracy, F1 score, recall και τα προεκπαιδευμένα word embeddings glove.6B.50d, τα οποία έχουν διάσταση 50. Ο παρακάτω πίνακας συνοψίζει τις επιδόσεις του μοντέλου για τον βέλτιστο συνδυασμό υπερπαραμέτρων, όπως προέκυψε από δοκιμές.

	MR	Semeval 2017 Task4-A
Train Loss	0.278	0.613
Test Loss	0.714	1.531
Train Accuracy	0.895	0.732
Test Accuracy	0.678	0.532
Train F1 Score	0.895	0.693
Test F1 Score	0.675	0.493
Train Recall	0.895	0.670
Test Recall	0.678	0.498

```
[=====] ...Epoch 150, Loss: 0.2845
Epoch 150
Train loss: 0.2782278800312477
Test loss: 0.7136167089144388
Train accuracy: 0.8949
Test accuracy: 0.6782477341389728
Train F1 score: 0.8946117325258607
Test F1 score: 0.67471388728047
Train Recall: 0.8949
Test Recall: 0.6782477341389728
```

MR

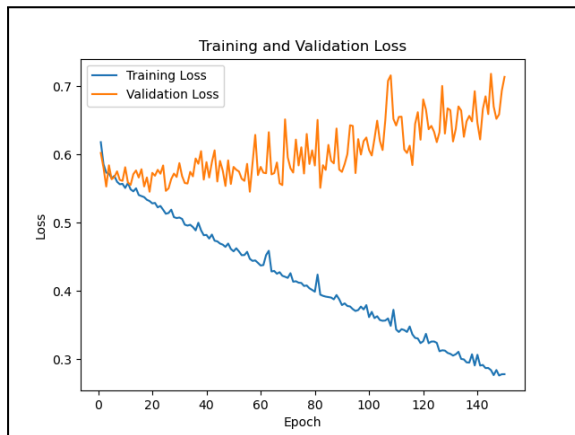
```
[=====] ...Epoch 150, Loss: 0.9306
Epoch 150
Train loss: 0.6129142244912914
Test loss: 1.5312071622659762
Train accuracy: 0.7317934234415977
Test accuracy: 0.5318308227938783
Train F1 score: 0.6929682452480499
Test F1 score: 0.4932214534507519
Train Recall: 0.6704137960171156
Test Recall: 0.4984143316130121
```

Semeval 2017 Task4-A

Για την επίτευξη της βέλτιστης απόδοσης - με στόχο τη μείωση του loss και τη βελτίωση των μετρικών (accuracy, F1 score, recall) - πραγματοποιήθηκαν δοκιμές με διάφορες τιμές υπερπαραμέτρων. Αρχικά, καθώς αυξανόταν η διάσταση των word embeddings από 50 σε 100 και 200, το μοντέλο εμφάνιζε ολοένα και εντονότερο φαινόμενο overfitting στα χαρακτηριστικά του train set, με αποτέλεσμα να μειώνεται η ικανότητά του να γενικεύει. Ομοίως, η υπερβολική αύξηση του αριθμού των εποχών εκπαίδευσης οδηγούσε σε φαινόμενο overfitting, καθώς μετά από ένα ορισμένο σημείο το μοντέλο συνέχιζε να βελτιώνεται στο train set, αλλά η απόδοσή του στο test set επιδεινωνόταν. Αυτό υποδηλώνει ότι το μοντέλο μάθαινε υπερβολικά καλά τα πρότυπα του train set.

Οι τιμές που προέκυψαν από το σύνολο δεδομένων Semeval 2017 Task 4-A είναι αρκετά ανησυχητικές, καθώς η υψηλή τιμή του test loss υποδεικνύει ότι το μοντέλο έχει κάνει overfitting στα χαρακτηριστικά του train set. Προκειμένου να βελτιωθούν οι τιμές αξίζει να

εξεταστεί η ενσωμάτωση τεχνικών όπως η χρήση regularization, η προσθήκη dropout ή/και η εκπαίδευση να σταματά όταν το loss του test set σταματάει να μειώνεται. Τέλος, αξίζει να αυξηθεί ο αριθμός των εποχών εφόσον οι 150 εποχές δεν επαρκούν για την πλήρη σύγκλιση του μοντέλου.



MR



Semeval 2017 Task4-A

Κατηγοριοποίηση με χρήση LLM

Δημιουργήθηκε το python script `load_sentences.py`. Ο συγκεκριμένος κώδικας φορτώνει τα δεδομένα από τα δύο dataset και μετατρέπει τις κατηγορικές ετικέτες των προτάσεων σε αριθμητικές μέσω του `LabelEncoder`. Στη συνέχεια, επιλέγει είκοσι κείμενα από κάθε κατηγορία και ανακατεύονται τυχαία σε αρχείο με όνομα `shuffled_sentences.txt`.

Τρία διαφορετικά prompts δίνονται στο μοντέλο ChatGPT για την αναγνώριση του συναισθήματος:

- Διάβασε τις παρακάτω προτάσεις και πες μου αν το συναίσθημα είναι θετικό, αρνητικό ή ουδέτερο για καθεμία ξεχωριστά. Απάντησε μου με 0 για αρνητικό, 1 για ουδέτερο, 2 για θετικό:

Dataset	Σωστές Προβλέψεις	Σύνολο Προτάσεων	Accuracy
Semeval 2017 Task4-A	40	60	66.67%
MR	37	40	92.50%

Βήμα 2: Υπολογισμός ακρίβειας

- Σύνολο σωστών προβλέψεων: 40
- Σύνολο προτάσεων: 60

`Accuracy = 40 / 60 = 66.67%`

Άρα:

`Accuracy = 37 / 40 = 0.925 → 92.5%`

✅ Η ακρίβεια μου ήταν 92.5%.


- Ποιο είναι το βασικό συναίσθημα σε κάθε μία από τις παρακάτω προτάσεις (θετικό, αρνητικό ή ουδέτερο); Απάντησε μου με 0 για αρνητικό, 1 για ουδέτερο, 2 για θετικό.

Εξήγησε την απάντησή σου και ανέφερε λέξεις ή φράσεις που σε οδήγησαν σε αυτό το συμπέρασμα:

Dataset	Σωστές Προβλέψεις	Σύνολο Προτάσεων	Accuracy
Semeval 2017 Task4-A	39	60	65%
MR	38	40	95%

✓ **Αποτελέσματα**

- Σωστές προβλέψεις: 39
- Λάθος προβλέψεις: 21
- Συνολικά δείγματα: 60

 **Accuracy**

$$\text{Accuracy} = \frac{39}{60} = 0.65 = 65\%$$

✓ **Βήμα 2: Υπολογισμός**

- Σωστές προβλέψεις: 38 από 40
- $\text{Accuracy} = \frac{38}{40} = 0.95$

- Ανέλυσε τις παρακάτω προτάσεις, την καθεμία ξεχωριστά, και προσδιόρισε ποιες λέξεις έχουν τη μεγαλύτερη συναισθηματική βαρύτητα. Πες μου αν είναι θετικό, αρνητικό ή ουδέτερο το γενικό συναίσθημα. Απάντησε μου με 0 για αρνητικό, 1 για ουδέτερο, 2 για θετικό:

Dataset	Σωστές Προβλέψεις	Σύνολο Προτάσεων	Accuracy
Semeval 2017 Task4-A	27	60	45%
MR	40	40	100%

Το μοντέλο ChatGPT παρουσιάζει σφάλματα στις προβλέψεις του, ιδιαίτερα στο dataset του Semeval 2017 Task 4-A. Αυτά τα σφάλματα αποδίδονται στην ασαφή ή αντιφατική φύση του περιεχομένου των προτάσεων. Συγκεκριμένα, οι σαρκαστικές ή ειρωνικές φράσεις είναι δύσκολο να εντοπιστούν αυτόματα, ενώ κάποιες προτάσεις περιέχουν τόσο θετικά όσο και αρνητικά στοιχεία, όπως μια κριτική που συνδυάζει θαυμασμό και ειρωνεία. Επίσης, ορισμένες λέξεις που φαίνονται ουδέτερες μπορεί να είναι φορτισμένες συναισθηματικά. Για παράδειγμα, λέξεις όπως "awful", "destroy", "ghoul" υποδεικνύουν έντονα αρνητικό συναίσθημα. Τέλος, φράσεις που φαίνονται περιγραφικές μπορεί να περιέχουν λεπτό συναίσθημα που χάνεται εύκολα, όπως στην περίπτωση της φράσης "Michael Jackson: Bad 25th Anniversary Edition...", που φαίνεται ουδέτερη, αλλά μπορεί να θεωρηθεί θετική αν την ερμηνεύσουμε ως φόρο τιμής.

Τέλος, υπάρχουν ορισμένες λέξεις που λειτουργούν ως ισχυροί δείκτες συναισθήματος. Για παράδειγμα, στο dataset του Semeval 2017 Task 4-A:

Θετικό (2): "happy birthday", "love", "was the man", "lit", "amazing", "touching", "tribute", "nostalgic", "loved it".

Αρνητικό (0): "awful", "stab", "lying", "destroy", "ghoul", "shit", "no talent", "ISIS", "fuckboys", "harassed".

Ουδέτερο (1): "anniversary", "album released", "was buried", "list", "trending", "listened to".

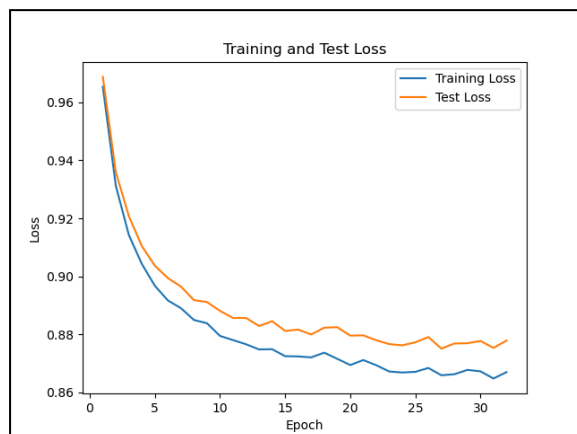
Ερωτήματα 2ου Εργαστηρίου

Για τις ανάγκες του εργαστηρίου, χρησιμοποιήθηκε το dataset του Semeval 2017 Task4-A και τα προεκπαιδευμένα word embeddings Twitter GloVe (glove.twitter.27B.50d), τα οποία έχουν διάσταση 50, καθώς είναι εκπαιδευμένα πάνω σε tweets.

Ερώτημα 1

1.1

Ως μη γραμμική συνάρτηση ενεργοποίησης επιλέγεται η ReLU. Η συνάρτηση forward του μοντέλου τροποποιείται στο αρχείο models.py, έτσι ώστε η αναπαράσταση κάθε πρότασης να υπολογίζεται ως η συνένωση του mean και max pooling των word embeddings της πρότασης. Η ενσωμάτωση του max pooling ενισχύει τη μεταβίβαση πληροφορίας στο επόμενο επίπεδο του δικτύου, διατηρώντας σημαντικά χαρακτηριστικά λέξεων με έντονο συναισθηματικό φορτίο.



```
[=====] ...Epoch 31, Loss: 0.9453
Epoch 31
Train loss: 0.8648142983836512
Test loss: 0.8753693088507041
Train accuracy: 0.58460258220698
Test accuracy: 0.5795844260641517
Train F1 score: 0.5164613635259202
Test F1 score: 0.5080684490040989
Train Recall: 0.508063295333447
Test Recall: 0.5017542688733355
[=====] ...Epoch 32, Loss: 0.8885
Epoch 32
Train loss: 0.8670183268285567
Test loss: 0.8778798809418311
Train accuracy: 0.5836443413354852
Test accuracy: 0.5787774863828928
Train F1 score: 0.49943199056297943
Test F1 score: 0.48704124580296443
Train Recall: 0.49418511073930266
Test Recall: 0.4858929698842145
Early stopping was activated.
Epoch 32/150, Loss at training set: 0.8670183268285567
Loss at validation set: 0.8778798809418311
Training has been completed.
```

Η διαδικασία εκπαίδευσης ολοκληρώνεται πριν από τη 150η εποχή, συγκεκριμένα κατά την 32η, λόγω της ενεργοποίησης του μηχανισμού early stopping.

1.2

Η αναπαράσταση με τη συνδυαστική προσέγγιση που ενώνει το mean και το max pooling εξάγει τις ισχυρότερες τιμές της πρότασης, δηλαδή, τις πιο έντονες σημασιολογικά λέξεις. Έτσι και διαφορετικά με την αρχική προσέγγιση του average pooling, οι λέξεις δεν συμβάλλουν ισότιμα στο embedding.

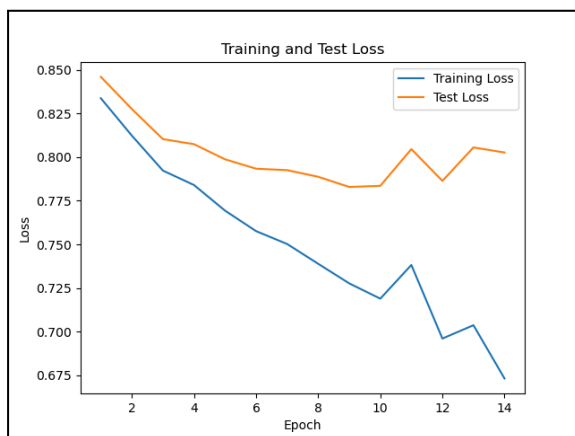
Ο παρακάτω πίνακας παρουσιάζει συνοπτικά τις επιδόσεις του τρέχοντος μοντέλου και τις συγκρίνει με εκείνες του μοντέλου, το οποίο είχε εκπαιδευτεί χρησιμοποιώντας αποκλειστικά την προσέγγιση του average pooling στο ίδιο σύνολο δεδομένων.

Semeval 2017 Task4-A	Mean-Max Pooling	Average Pooling
Train Loss	0.867	0.613
Test Loss	0.878	1.531
Train Accuracy	0.584	0.732
Test Accuracy	0.579	0.532
Train F1 Score	0.499	0.693
Test F1 Score	0.487	0.493
Train Recall	0.494	0.670
Test Recall	0.486	0.498

Γίνεται κατανοητό πως η χρήση της συνδυαστικής προσέγγισης που ενώνει το mean και το max pooling βελτιώνει σημαντικά την ικανότητα του μοντέλου να γενικεύει, συγκριτικά με τη μέθοδο του average pooling. Παρότι το μέσο loss στο training set είναι υψηλότερο για το mean-max pooling, το μοντέλο πετυχαίνει καλύτερες επιδόσεις στο test set ως προς το loss και την ακρίβεια. Αυτό υποδεικνύει ότι ενισχύθηκε η γενίκευση του μοντέλου σε μη γνωστά δεδομένα.

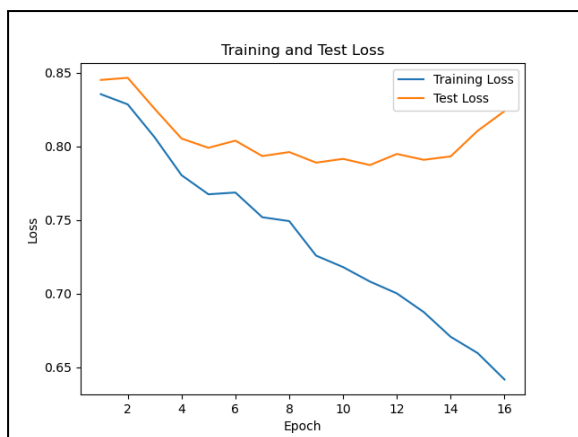
Ερώτημα 2

Η κλάση LSTM και στη μέθοδο αυτής forward τα embeddings συσκευάζονται με τη βοήθεια της `pack_padded_sequence`, που απαιτεί τις πραγματικές διάρκειες κάθε ακολουθίας. Εκεί γίνεται χρήση της `torch.clamp(lengths, max=max_length)` για να εξασφαλιστεί ότι καμία τιμή των `lengths` δεν ξεπερνά το μέγιστο επιτρεπτό μήκος `max_length` του batch. Στο τέλος, το δίκτυο συλλέγει την έξοδο του LSTM στο τελευταίο timestep κάθε ακολουθίας, εξαιρώντας τα zero-padded timesteps.



```
Epoch 13
Train loss: 0.7036820805841877
Test loss: 0.805554870611582
Train accuracy: 0.6825701028848093
Test accuracy: 0.626487795037321
Train F1 score: 0.6619209775076077
Test F1 score: 0.5993109221847172
Train Recall: 0.6566014308847897
Test Recall: 0.5947019068009859
[=====] ...Epoch 14, Loss: 0.7635
Epoch 14
Train loss: 0.6731477237516834
Test loss: 0.8026576645863361
Train accuracy: 0.6946741981036918
Test accuracy: 0.6325398426467621
Train F1 score: 0.661444977952033
Test F1 score: 0.586836965944294
Train Recall: 0.6416586161465604
Test Recall: 0.5728635392306155
Early stopping was activated.
Epoch 14/150, Loss at training set: 0.6731477237516834
Loss at validation set: 0.8026576645863361
Training has been completed.
```

Non-bidirectional LSTM



```
Epoch 15
Train loss: 0.659669747660237
Test loss: 0.8104983828086647
Train accuracy: 0.7054670163405286
Test accuracy: 0.6299172886826709
Train F1 score: 0.6913640587991655
Test F1 score: 0.6065376278844311
Train Recall: 0.6928506886209961
Test Recall: 0.6064111663542692
[=====] ...Epoch 16, Loss: 0.5924
Epoch 16
Train loss: 0.6417073035432447
Test loss: 0.8238977851011814
Train accuracy: 0.7119981843857172
Test accuracy: 0.624773048214646
Train F1 score: 0.6910801798859577
Test F1 score: 0.5868634368772977
Train Recall: 0.6783154970992945
Test Recall: 0.5780981431565854
Early stopping was activated.
Epoch 16/150, Loss at training set: 0.6417073035432447
Loss at validation set: 0.8238977851011814
Training has been completed.
```

Bidirectional LSTM

Η χρήση της `torch.clamp(lengths, max=max_length)` λειτουργεί ως πρόχειρη λύση για την αποφυγή σφαλμάτων διαστάσεων, καθώς περιορίζει τα μήκη των ακολουθιών στο επιτρεπόμενο όριο. Ωστόσο, ενέχει τον κίνδυνο απώλειας πληροφορίας, αφού αποκόπτει οποιοδήποτε στοιχείο υπερβαίνει το καθορισμένο μήκος. Αυτό μπορεί να επηρεάσει αρνητικά την απόδοση του μοντέλου, καθώς σημαντικά δεδομένα αγνοούνται. Σύμφωνα με τη βιβλιογραφία, η βέλτιστη πρακτική είναι η χρήση δυναμικού padding στο `collate_fn`, ώστε το padding να προσαρμόζεται στο πραγματικό μέγιστο μήκος κάθε batch.

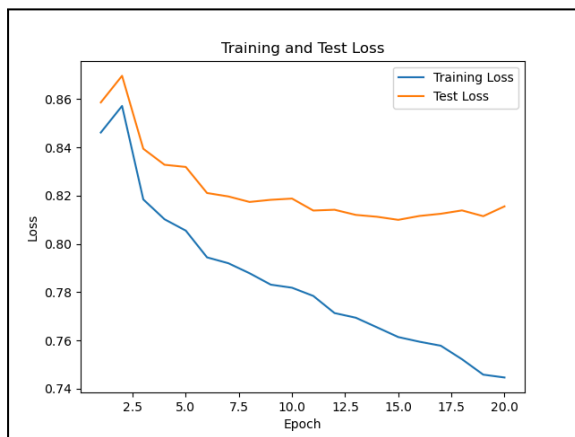
Ο παρακάτω πίνακας παρουσιάζει συνοπτικά τις επιδόσεις των τρεχόντων μοντέλων. Η διαδικασία εκπαίδευσης των μοντέλων ολοκληρώνεται πριν από τη 150η εποχή λόγω της ενεργοποίησης του μηχανισμού early stopping.

Semeval 2017 Task4-A	Non-bidirectional LSTM	Bidirectional LSTM
Train Loss	0.673	0.642
Test Loss	0.803	0.824
Train Accuracy	0.695	0.712
Test Accuracy	0.632	0.625
Train F1 Score	0.661	0.691
Test F1 Score	0.587	0.587
Train Recall	0.642	0.678
Test Recall	0.573	0.578

Ερώτημα 3

3.1

Αξιοποιείται το μοντέλο SimpleSelfAttentionModel και εφαρμόζεται average pooling στις αναπαραστάσεις των λέξεων πριν το τελευταίο layer, για να εξάγεται την αναπαράσταση της πρότασης. Προκαταρκτικά, προστίθεται ένας έλεγχος στο training.py ώστε να καλείται σωστά η forward μέθοδος του εκάστοτε μοντέλου: εάν το μοντέλο είναι το SimpleSelfAttentionModel, καλείται χωρίς το όρισμα lengths, ενώ για τα υπόλοιπα μοντέλα καλείται με inputs και lengths.



```
[=====] ...Epoch 19, Loss: 0.7590
Epoch 19
Train loss: 0.7458162732662693
Test loss: 0.8115026759795654
Train accuracy: 0.6566219497593302
Test accuracy: 0.6210409521888239
Train F1 score: 0.6195090149940383
Test F1 score: 0.5755018039085558
Train Recall: 0.6026704853882335
Test Recall: 0.5625036168608668
[=====] ...Epoch 20, Loss: 0.7307
Epoch 20
Train loss: 0.7446349641969127
Test loss: 0.8155708809693655
Train accuracy: 0.6550654428081501
Test accuracy: 0.6181157958442607
Train F1 score: 0.6170110383765054
Test F1 score: 0.5710966837022017
Train Recall: 0.5998076537382665
Test Recall: 0.5578384303811195
Early stopping was activated.
Epoch 20/150, Loss at training set: 0.7446349641969127
Loss at validation set: 0.8155708809693655
Training has been completed.
```

Η διαδικασία εκπαίδευσης ολοκληρώνεται πριν από τη 150η εποχή, συγκεκριμένα κατά την 20η, λόγω της ενεργοποίησης του μηχανισμού early stopping.

SimpleSelfAttentionModel	Accuracy	F1 Score	Recall
Test Set	0.618	0.571	0.558

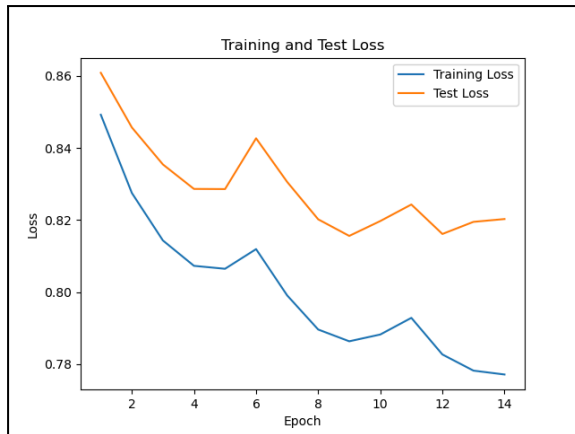
3.2

Στη κλάση attention.Head τα queries, keys και values είναι τρία διανύσματα που προκύπτουν από γραμμικούς μετασχηματισμούς της εισόδου x. Αναλυτικά, κάθε query αντιστοιχεί σε ένα token και δείχνει σε ποια άλλα tokens θα εστιάσει, τα keys υπολογίζουν πόσο σχετικό είναι το token με τα άλλα μέσω εσωτερικού γινομένου, ενώ τα values περιέχουν την πραγματική πληροφορία που θα διαχυθεί από το query. Στη συνέχεια, γίνεται ο υπολογισμός του weighted sum που είναι το βασικό score της κλάσης, το οποίο αποτελεί την προσοχή που δίνει κάθε query σε κάθε key. Συνολικά, ο μηχανισμός του attention επιτρέπει στο μοντέλο να αναλύει δυναμικά τις σχέσεις ανάμεσα σε όλα τα tokens της πρότασης.

Στη κλάση attention.SimpleSelfAttentionModel τα position_embeddings διακρίνουν τη σειρά των tokens στην πρόταση.

Ερώτημα 4

Αξιοποιείται το μοντέλο MultiHeadAttentionModel και ο παρακάτω πίνακας παρουσιάζει συνοπτικά τις επιδόσεις του τρέχοντος μοντέλου:



```
[-----] ...Epoch 13, Loss: 0.7525
Epoch 13
Train loss: 0.7782347519551555
Test loss: 0.8195010049220843
Train accuracy: 0.6357423845067581
Test accuracy: 0.6169053863223724
Train F1 score: 0.5834274280514453
Test F1 score: 0.5619581672945696
Train Recall: 0.5668884949261567
Test Recall: 0.5475446684103883
[-----] ...Epoch 14, Loss: 0.8496
Epoch 14
Train loss: 0.7771591330728224
Test loss: 0.8202834763588049
Train accuracy: 0.6360702037522695
Test accuracy: 0.6163001815614283
Train F1 score: 0.5837556088693515
Test F1 score: 0.5590131901894047
Train Recall: 0.5671600806506535
Test Recall: 0.5451768166671754
Early stopping was activated.
Epoch 14/150, Loss at training set: 0.7771591330728224
Loss at validation set: 0.8202834763588049
Training has been completed.
```

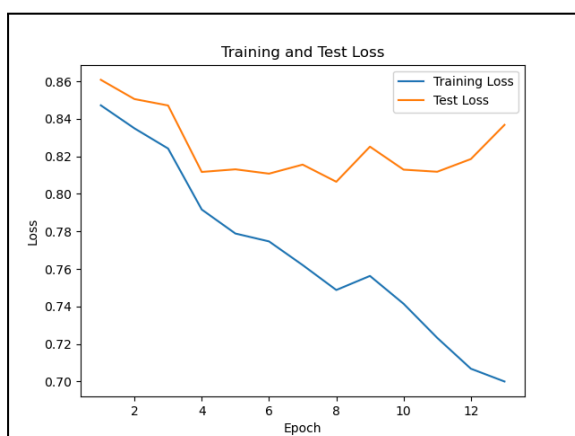
Η διαδικασία εκπαίδευσης ολοκληρώνεται πριν από τη 150η εποχή, συγκεκριμένα κατά την 14η, λόγω της ενεργοποίησης του μηχανισμού early stopping.

MultiHeadAttentionModel	Accuracy	F1 Score	Recall
Test Set	0.616	0.559	0.545

Ερώτημα 5

Το μοντέλο αξιοποιεί μια αρχιτεκτονική Transformer-Encoder για την κατηγοριοποίηση συναισθήματος, εφαρμόζοντας average pooling στις αναπαραστάσεις των λέξεων κάθε πρότασης.

Η διαφορά μεταξύ των μοντέλων MultiHeadAttentionModel και TransformerEncoderModel έγκειται στη δομή τους. Το πρώτο περιορίζεται σε ένα μόνο επίπεδο προσοχής και ένα feedforward δίκτυο, ενώ το δεύτερο ενσωματώνει μια βαθύτερη ιεραρχία με επαναλαμβανόμενα blocks. Κάθε τέτοιο block περιλαμβάνει όχι μόνο τον μηχανισμό προσοχής, αλλά και επιπλέον στρώματα κανονικοποίησης και feedforward. Ο παρακάτω πίνακας παρουσιάζει συνοπτικά τις επιδόσεις του τρέχοντος μοντέλου στο test set:



```
[-----] ...Epoch 12, Loss: 0.8085
Epoch 12
Train loss: 0.7068115628534748
Test loss: 0.8185856693830246
Train accuracy: 0.6807797054670164
Test accuracy: 0.6146863021989106
Train F1 score: 0.6441180906625171
Test F1 score: 0.5673989517232885
Train Recall: 0.6261550492918949
Test Recall: 0.5554126884064616
[-----] ...Epoch 13, Loss: 0.8033
Epoch 13
Train loss: 0.6999883024923264
Test loss: 0.8368057272373102
Train accuracy: 0.6850413556586645
Test accuracy: 0.610550736324592
Train F1 score: 0.6340235635410331
Test F1 score: 0.5448336145009725
Train Recall: 0.613055285193208
Test Recall: 0.5333793577428847
Early stopping was activated.
Epoch 13/150, Loss at training set: 0.6999883024923264
Loss at validation set: 0.8368057272373102
Training has been completed.
```

Transformer-Encoder	Accuracy	F1 Score	Recall
Test Set	0.610	0.545	0.533

Η διαδικασία εκπαίδευσης ολοκληρώνεται πριν από τη 150η εποχή, συγκεκριμένα κατά την 13η, λόγω της ενεργοποίησης του μηχανισμού early stopping. Μέχρι στιγμής, παρατηρείται πως λόγω της χρήσης της τεχνικής του early stopping, τα μοντέλα δεν εκπαιδεύονται για ικανοποιητικό αριθμό εποχών. Οι επιδόσεις των μοντέλων με μηχανισμούς attention είναι χαμηλότερες από τις αναμενόμενες, κάτι που ενδέχεται να οφείλεται στον περιορισμένο αριθμό εποχών και στην απλούστερη αρχιτεκτονική τους. Το μοντέλο τύπου Transformer έπρεπε να παρουσιάζει την καλύτερη απόδοση, καθώς διαθέτει βαθύτερη αρχιτεκτονική με επαναλαμβανόμενα blocks. Αυτό του επιτρέπει να μαθαίνει πιο σύνθετες εξαρτήσεις και σχέσεις στα δεδομένα, προσφέροντας καλύτερη γενίκευση και απόδοση.

Τέλος, σύμφωνα με το επιστημονικό άρθρο "[Attention Is All You Need](#)", οι default τιμές των υπερπαραμέτρων στην κλασική αρχιτεκτονική του Transformer είναι:

- embedding dimension = 512 (βλ. σελίδα 3)
- n_layer = 6 (βλ. σελίδα 3)
- n_head = 8 (βλ. σελίδα 5)
- dropout: 0.1 (βλ. σελίδα 8)

Ερώτημα 6

Χρησιμοποιήθηκαν έξι Pre-Trained Transformer μοντέλα στα dataset του Semeval 2017 Task4-A με κλάσεις {negative, neutral, positive} και του MR με κλάσεις {negative, positive}. Ο παρακάτω πίνακας παρουσιάζει συνοπτικά τις επιδόσεις των μοντέλων:

Semeval 2017 Task4-A	Accuracy	F1 Score	Recall
cardiffnlp/	0.724	0.722	0.723
finiteautomata/	0.718	0.718	0.730
j-hartmann/	0.698	0.649	0.669
MR	Accuracy	F1 Score	Recall
siebert/	0.924	0.925	0.925
distilbert/	0.891	0.891	0.891
facebook/	0.814	0.814	0.812

```
Dataset: Semeval2017A
Pre-Trained model: cardiffnlp/twitter-roberta-base-sentiment
Test set evaluation
accuracy: 0.7237870400521003
recall: 0.7229454214750545
f1-score: 0.7222115953560642
```

```
Dataset: Semeval2017A
Pre-Trained model: finiteautomata/bertweet-base-sentiment-analysis
Test set evaluation
accuracy: 0.7177629436665581
recall: 0.7301871228078923
f1-score: 0.718050644575488
```

```
Dataset: Semeval2017A
Pre-Trained model: j-hartmann/sentiment-roberta-large-english-3-classes
Test set evaluation
accuracy: 0.6984695538912407
recall: 0.6487496185621454
f1-score: 0.6693682259677627
```

```
Dataset: MR
Pre-Trained model: siebert/sentiment-roberta-large-english
Test set evaluation
accuracy: 0.9244712990936556
recall: 0.9244712990936557
f1-score: 0.924468541489818
```

```
Dataset: MR
Pre-Trained model: distilbert/distilbert-base-uncased-finetuned-sst-2-english
Test set evaluation
accuracy: 0.8912286786948641
recall: 0.8912386786948641
f1-score: 0.891213847582191
```

```
Dataset: MR
Pre-Trained model: facebook/bart-large-mnli
Test set evaluation
accuracy: 0.8141993957703928
recall: 0.8141993957703928
f1-score: 0.812037312760992
```

Τα μοντέλα `cardiffnlp/` και `finiteautomata/` παρουσιάζουν τις καλύτερες επιδόσεις στο Semeval 2017 Task4-A, με ακρίβεια πάνω από 0.7, καθώς είναι εκπαιδευμένα σε δεδομένα που σχετίζονται με tweets, κάτι που τα καθιστά ιδιαίτερα ικανά για sentiment analysis. Αντίθετα, το μοντέλο `j-hartmann/` υστερεί, πιθανώς λόγω περιορισμένων δεδομένων εκπαίδευσης. Στο MR, το `siebert/` και το `distilbert/` επιτυγχάνουν υψηλές επιδόσεις, ενώ το `facebook/` παρουσιάζει χαμηλότερη απόδοση, πιθανώς λόγω της γενικότερης φύσης του μοντέλου που δεν έχει εκπαιδευτεί ειδικά για sentiment analysis.

Ερώτημα 7

Κατά τη διαδικασία fine-tuning των Pre-Trained Transformer μοντέλων παρατηρήθηκε ότι, σε τοπική εκτέλεση με αυξημένο αριθμό εποχών (άνω των 150), το σύστημα παρουσίασε επαναλαμβανόμενα σφάλματα και κρασαρίσματα, πιθανότατα λόγω των περιορισμένων υπολογιστικών πόρων. Η προτεινόμενη στρατηγική θα περιλάμβανε μεγαλύτερο batch size, fine-tuning για 10–20 εποχές με early stopping και hyperparameter tuning (μέσω random search) ώστε να επιτευχθεί η βέλτιστη γενίκευση του μοντέλου, αξιοποιώντας παράλληλα τα περιορισμένα dataset.

Ερώτημα 8

Το μοντέλο ChatGPT έδωσε τις εξής εξηγήσεις και σχολιασμούς για τον κώδικα του tutorial “Let’s build GPT: from scratch, in code, spelled out”:

Φυσικά! Ο κώδικας αυτός υλοποιεί ένα μικρό γλωσσικό μοντέλο τύπου Transformer σε PyTorch, εκπαιδευμένο πάνω στο dataset TinyShakespeare. Πάμε να τον αναλύσουμε βήμα-βήμα:

- `batch_size`: Πόσα ανεξάρτητα sequence θα επεξεργαζόμαστε ταυτόχρονα.
- `block_size`: Το μήκος κάθε sequence (πόσο context βλέπει το μοντέλο).
- `device`: CUDA αν υπάρχει GPU.
- `n_embd`: Διάσταση embeddings (και μοντέλου γενικά).
- `n_head`: Πόσα attention heads (π.χ. για παράλληλη επεξεργασία).
- `n_layer`: Αριθμός Transformer blocks.
- `dropout`: Regularization (0.0 εδώ, για απλότητα).

Ο κώδικας αυτός φτιάχνει ένα μικρό γλωσσικό μοντέλο Transformer (σαν τον GPT) που μαθαίνει να προβλέπει χαρακτήρες, εκπαιδευόμενο πάνω στο κείμενο του TinyShakespeare. Πιο συγκεκριμένα:

- Δεδομένα: Μετατρέπει το κείμενο σε αριθμούς (ένα λεξικό χαρακτήρων) και το χωρίζει σε training και validation set.
- Μοντέλο: Περιλαμβάνει layers attention, position embeddings, normalization και feedforward. Μαθαίνει να προβλέπει τον επόμενο χαρακτήρα με βάση τους προηγούμενους.
- Εκπαίδευση: Γίνεται από το μηδέν, χωρίς προεκπαιδευμένα βάρη, και χρησιμοποιεί απώλεια cross-entropy και optimizer AdamW.
- Δημιουργία κειμένου: Μετά την εκπαίδευση, μπορεί να παράγει νέο κείμενο χαρακτήρα-χαρακτήρα με βάση ένα αρχικό prompt.

Έπειτα, το μοντέλο ChatGPT αξιολόγησε τον κώδικα:

Ο κώδικας είναι πολύ καλός για εκπαιδευτικούς σκοπούς και προσφέρει μια καθαρή, μινιμαλιστική υλοποίηση ενός Transformer-based γλωσσικού μοντέλου. Παρακάτω μια αξιολόγηση με τα δυνατά σημεία και πιθανές βελτιώσεις:

Δυνατά Σημεία

Απλότητα και καθαρότητα: Ο κώδικας είναι μικρός αλλά καλύπτει όλες τις βασικές έννοιες (embeddings, attention, layer norm, training loop).

Εκπαιδευτικός: Είναι ιδανικός για να κατανοήσει κάποιος πώς λειτουργεί ένα Transformer από την αρχή.

Χρήση PyTorch: Ενσωματώνει ορθά όλες τις βασικές αρχές PyTorch, από layers μέχρι το training loop και inference.

Αυτόνομο μοντέλο: Δεν χρειάζεται εξωτερικά αρχεία ή βιβλιοθήκες πέρα από το input.txt.

Πιθανά Σημεία Βελτίωσης

Όνομα μοντέλου: Το όνομα BigramLanguageModel είναι παραπλανητικό – το μοντέλο δεν είναι bigram, αλλά Transformer.

Απουσία Masking στο Projection Layer: Αν και γίνεται σωστό masking στο attention, δεν εξετάζεται ρητά αν το projection layer διατηρεί αιτιότητα.

Χρόνος εκπαίδευσης: 5.000 εποχές είναι λίγες για καλή παραγωγή. Το μοντέλο εκπαιδεύεται κυρίως για επίδειξη.

Δεν αποθηκεύεται το μοντέλο: Θα ήταν καλό να υπάρχει αποθήκευση/φόρτωση checkpoint.

Δεν χρησιμοποιούνται advanced τεχνικές:

Δεν υπάρχει learning rate scheduler.

Δεν υπάρχει gradient clipping.

Δεν αξιοποιούνται mixed precision ή άλλες τεχνικές επιτάχυνσης.

Απουσία τεκμηρίωσης / σχόλια σε επίπεδο αρχείου: Μερικά blocks θα μπορούσαν να έχουν inline περιγραφές για αρχάριους.

Αν θες, μπορώ να σου προτείνω μια «βελτιωμένη» έκδοση του ίδιου κώδικα με checkpointing, tensorboard logging ή υποστήριξη word-level μοντέλων. Θες κάτι τέτοιο;

Τέλος, το μοντέλο ChatGPT ξαναέγραψε τον κώδικα με καθαρή αρχιτεκτονική, υποστήριξη για checkpoints, logging (προαιρετικά με TensorBoard), modular κώδικα για εύκολες επεκτάσεις και σωστό όνομα (TransformerLanguageModel).

Αλλαγή 1: Όνομα Μοντέλου

```
class TransformerLanguageModel(nn.Module):
```

Αλλαγή 2: Ορισμός του Μοντέλου

```
model = TransformerLanguageModel().to(device)
```

Αλλαγή 3: Αρχιτεκτονική του Μοντέλου

```
class TransformerLanguageModel(nn.Module):
```

```
    def __init__(self):
        super().__init__()
        self.token_embedding_table = nn.Embedding(vocab_size, n_embd)
        self.position_embedding_table = nn.Embedding(block_size, n_embd)
        self.blocks = nn.Sequential(*[Block(n_embd, n_head) for _ in range(n_layer)])
        self.ln_f = nn.LayerNorm(n_embd)
        self.lm_head = nn.Linear(n_embd, vocab_size)

    def forward(self, idx, targets=None):
        B, T = idx.shape
        tok_emb = self.token_embedding_table(idx)
        pos_emb = self.position_embedding_table(torch.arange(T, device=idx.device))
        x = tok_emb + pos_emb
        x = self.blocks(x)
        x = self.ln_f(x)
        logits = self.lm_head(x)

        loss = None
        if targets is not None:
            logits = logits.view(-1, vocab_size)
            targets = targets.view(-1)
            loss = F.cross_entropy(logits, targets)

        return logits, loss

    def generate(self, idx, max_new_tokens):
        for _ in range(max_new_tokens):
            idx_cond = idx[:, -block_size:]
            logits, _ = self(idx_cond)
            logits = logits[:, -1, :]
            probs = F.softmax(logits, dim=-1)
            idx_next = torch.multinomial(probs, num_samples=1)
            idx = torch.cat((idx, idx_next), dim=1)
        return idx
```

Νέα Classes που πρέπει να προσθέσεις

```

class Head(nn.Module):
    def __init__(self, head_size):
        super().__init__()
        self.key = nn.Linear(n_embd, head_size, bias=False)
        self.query = nn.Linear(n_embd, head_size, bias=False)
        self.value = nn.Linear(n_embd, head_size, bias=False)
        self.register_buffer('tril', torch.tril(torch.ones(block_size, block_size)))
        self.dropout = nn.Dropout(dropout)

    def forward(self, x):
        B, T, C = x.shape
        k = self.key(x)
        q = self.query(x)
        wei = q @ k.transpose(-2, -1) * (C ** -0.5)
        wei = wei.masked_fill(self.tril[:T, :T] == 0, float('-inf'))
        wei = F.softmax(wei, dim=-1)
        wei = self.dropout(wei)
        v = self.value(x)
        return wei @ v

class MultiHeadAttention(nn.Module):
    def __init__(self, num_heads, head_size):
        super().__init__()
        self.heads = nn.ModuleList([Head(head_size) for _ in range(num_heads)])
        self.proj = nn.Linear(n_embd, n_embd)
        self.dropout = nn.Dropout(dropout)

    def forward(self, x):
        out = torch.cat([h(x) for h in self.heads], dim=-1)
        return self.dropout(self.proj(out))

class FeedForward(nn.Module):
    def __init__(self, n_embd):
        super().__init__()
        self.net = nn.Sequential(
            nn.Linear(n_embd, 4 * n_embd),
            nn.ReLU(),
            nn.Linear(4 * n_embd, n_embd),
            nn.Dropout(dropout),
        )

    def forward(self, x):
        return self.net(x)

class Block(nn.Module):
    def __init__(self, n_embd, n_head):
        super().__init__()
        head_size = n_embd // n_head
        self.sa = MultiHeadAttention(n_head, head_size)
        self.ffwd = FeedForward(n_embd)
        self.ln1 = nn.LayerNorm(n_embd)
        self.ln2 = nn.LayerNorm(n_embd)

    def forward(self, x):

```

```
x = x + self.sa(self.ln1(x))  
x = x + self.ffwd(self.ln2(x))  
return x
```

Οι απαντήσεις του μοντέλου ChatGPT ήταν σαφείς και ακριβείς, παρέχοντας κατανοητές εξηγήσεις. Ειδικότερα, χάρη στη λειτουργία του memory του και τις προηγούμενες συζητήσεις σχετικά με την κατανόηση των LSTM, το μοντέλο προχώρησε σε refactoring του κώδικα, εστιάζοντας στην ενσωμάτωση τεχνικών όπως το multi-head attention και τα feedforward layers, που έχουν συζητηθεί στα προηγούμενα ερωτήματα του εργαστηρίου. Εντοπίζει επιτυχώς λάθη και παραλείψεις του κώδικα.