ΔΠΜΣ Επιστήμη Δεδομένων και Μηχανική Μάθηση

Επεξεργασία Φωνής και Φυσικής Γλώσσας

3η Εργαστηριακή Άσκηση

Ονοματεπώνυμο: Ευάγγελος Τσόγκας

Αριθμός Μητρώου: 03400120

Προεπεξεργασία δεδομένων

Πριν γίνει η εκπαίδευση των μοντέλων για αναγνώριση συναισθήματος είναι πολύ σημαντικό να γίνει σωστή προεπεξεργασία των δεδομένων, ώστε να μπορούν να δοθούν σαν είσοδος σε κάποιο νευρωνικό δίκτυο.

Το πρώτο βήμα αφορά την **κωδικοποίηση των κλάσεων** που είναι σε μορφή κειμένου, έτσι ώστε να αντιστοιχούν σε έναν αριθμό. Στα παρακάτω screenshots φαίνονται τα 10 πρώτα labels από τα δεδομένα εκπαίδευσης για τα δύο datasets και η αντιστοιχία τους σε αριθμούς.

MR dataset

```
First 10 labels of training data:
 ['positive', 'positive', 'positive', 'positive', 'positive', 'positive', 'positive', 'positive', 'positive', 'positive']

First 10 labels of training data, encoded:
 [1 1 1 1 1 1 1 1 1 1]
```

Semeval2017A dataset

```
First 10 labels of training data:
 ['neutral', 'positive', 'neutral', 'positive', 'positive', 'positive', 'neutral', 'positive', 'negative', 'neutral']

First 10 labels of training data, encoded:
 [1 2 1 2 2 2 1 2 0 1]
```

Το επόμενο βήμα είναι η **λεκτική ανάλυση (tokenization)**. Πριν γίνει το tokenization γίνεται μια μικρή προεπεξεργασία των κειμένων με τον εξής τρόπο. Μετατρέπουμε όλους τους χαρακτήρες σε πεζούς, διαγράφουμε urls, κάνουμε expand τα contractions και διαγράφουμε λέξεις που αποτελούνται από έναν μόνο χαρακτήρα. Στη συνέχεια

χρησιμοποιούμε το RegexpTokenizer του NLTK και κάνουμε tokenize το κείμενο κρατώντας μόνο αλφαριθμητικούς χαρακτήρες.

Παρακάτω φαίνονται μερικά παραδείγματα των αποτελεσμάτων αυτής της επεξεργασίας για τα δύο datasets.

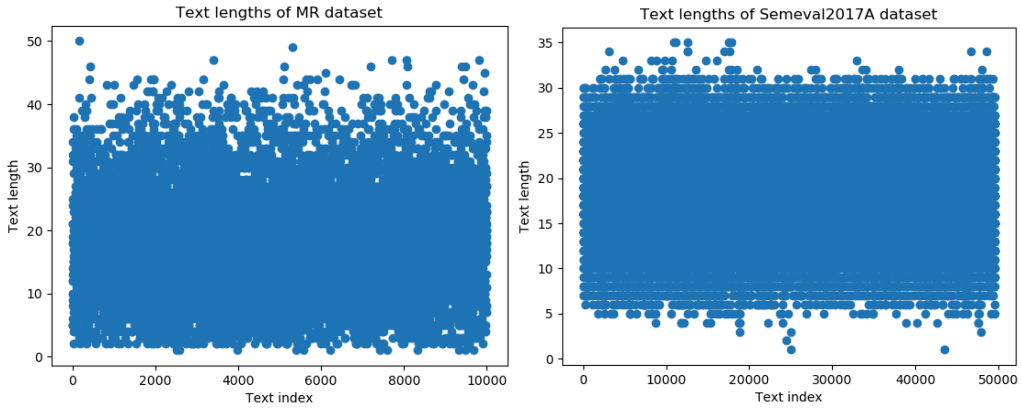
MR dataset

```
First 10 training examples after preprocessing:
['the', 'rock', 'is', 'destined', 'to', 'be', 'the', '21st', 'century', 'new', 'conan', 'and', 'that', 'he', 'is', 'going'
['the', 'gorgeously', 'elaborate', 'continuation', 'of', 'the', 'lord', 'of', 'the', 'rings', 'trilogy', 'is', 'so', 'huge
['effective', 'but', 'too', 'tepid', 'biopic']
['if', 'you', 'sometimes', 'like', 'to', 'go', 'to', 'the', 'movies', 'to', 'have', 'fun', 'wasabi', 'is', 'good', 'place'
['emerges', 'as', 'something', 'rare', 'an', 'issue', 'movie', 'that', 'is', 'so', 'honest', 'and', 'keenly', 'observed',
['the', 'film', 'provides', 'some', 'great', 'insight', 'into', 'the', 'neurotic', 'mindset', 'of', 'all', 'comics', 'even
['offers', 'that', 'rare', 'combination', 'of', 'entertainment', 'and', 'education']
['perhaps', 'no', 'picture', 'ever', 'made', 'has', 'more', 'literally', 'showed', 'that', 'the', 'road', 'to', 'hell', 'i
['steers', 'turns', 'in', 'snappy', 'screenplay', 'that', 'curls', 'at', 'the', 'edges', 'it', 'is', 'so', 'clever', 'you'
['take', 'care', 'of', 'my', 'cat', 'offers', 'refreshingly', 'different', 'slice', 'of', 'asian', 'cinema']
```

Semeval2017A dataset

```
First 10 training examples after preprocessing:
['05', 'beat', 'it', 'michael', 'jackson', 'thriller', '25th', 'anniversary', 'edition', 'hd']
['jay', 'joins', 'instagram', 'with', 'nostalgic', 'tribute', 'to', 'michael', 'jackson', 'jay', 'apparently', 'joined
['michael', 'jackson', 'bad', '25th', 'anniversary', 'edition', 'picture', 'vinyl', 'this', 'unique', 'picture', 'disc
['liked', 'youtube', 'video', 'one', 'direction', 'singing', 'man', 'in', 'the', 'mirror', 'by', 'michael', 'jackson',
['18th', 'anniv', 'of', 'princess', 'diana', 'death', 'still', 'want', 'to', 'believe', 'she', 'is', 'living', 'on', '
['oridaganjazz', 'the', '1st', 'time', 'heard', 'michael', 'jackson', 'sing', 'was', 'in', 'honolulu', 'hawaii', 'rest
['michael', 'jackson', 'appeared', 'on', 'saturday', '29', 'at', 'the', '9th', 'place', 'in', 'the', 'top20', 'of', 'm
['are', 'you', 'old', 'enough', 'to', 'remember', 'michael', 'jackson', 'attending', 'the', 'grammys', 'with', 'brooke
['etbowser', 'do', 'you', 'enjoy', 'his', '2nd', 'rate', 'michael', 'jackson', 'bit', 'honest', 'ques', 'like', 'the',
['the', 'weeknd', 'is', 'the', 'closest', 'thing', 'we', 'may', 'get', 'to', 'michael', 'jackson', 'for', 'long', 'tim
```

Τέλος, κάνουμε **κωδικοποίηση των λέξεων**, έτσι ώστε κάθε όρος να χαρτογραφηθεί σε έναν αριθμό με τον οποίο το embedding layer θα μπορεί να τον αντιστοιχίσει στη σωστή διανυσματική αναπαράσταση. Σημειώνουμε ότι χρησιμοποιούμε τα glove.twitter embeddings 25 διαστάσεων για λόγους υπολογιστικής ταχύτητας και μειωμένων απαιτήσεων σε μνήμη. Επειδή όμως θέλουμε όλες οι ακολουθίες να έχουν ίδιο μήκος πρέπει να βρούμε έναν καλό μέγιστο αριθμό για κάθε πρόταση και είτε να αφαιρούμε τις επιπλέον λέξεις, είτε να κάνουμε zero-padding αν είναι λιγότερες. Στις παρακάτω εικόνες φαίνονται τα scatter plots για τα μήκη των κειμένων στα 2 datasets (μετά το tokenization) και συμπεραίνουμε ότι το μέγιστο μήκος 35 λέξεων είναι καλό για τις ανάγκες μας.



Παρακάτω φαίνονται μερικά παραδείγματα από το Semeval2017A dataset, όπως είναι στην αρχική τους μορφή (μετά το tokenization) και όπως τα επιστρέφει το SentenceDataset, κωδικοποιημένα.

```
dataitem = ['05', 'beat', 'it', 'michael', 'jackson', 'thriller', '25th', 'anniversary', 'edition', 'hd']
Return values:
example = [1193515, 1079, 34, 2018, 3912, 19747, 1193515, 4285, 4078, 2995, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
label = 1
length = 10

dataitem = ['jay', 'joins', 'instagram', 'with', 'nostalgic', 'tribute', 'to', 'michael', 'jackson', 'jay', 'apparent
Return values:
example = [3904, 16210, 1157, 59, 49486, 10648, 17, 2018, 3912, 3904, 3323, 8935, 1157, 47, 1341, 27, 0, 0, 0, 0, 0,
label = 2
length = 16

dataitem = ['michael', 'jackson', 'bad', '25th', 'anniversary', 'edition', 'picture', 'vinyl', 'this', 'unique', 'pic
Return values:
example = [2018, 3912, 356, 1193515, 4285, 4078, 921, 17557, 54, 7262, 921, 15682, 17557, 14771, 14, 2898, 1193515, 0
label = 1
length = 17

dataitem = ['liked', 'youtube', 'video', 'one', 'direction', 'singing', 'man', 'in', 'the', 'mirror', 'by', 'michael'
Return values:
example = [1305, 1506, 287, 97, 792, 2081, 247, 36, 14, 4306, 153, 2018, 3912, 36, 6616, 295, 3221, 1193515, 0, 0, 0,
label = 2
length = 18

dataitem = ['18th', 'anniv', 'of', 'princess', 'diana', 'death', 'still', 'want', 'to', 'believe', 'she', 'is', 'livi
Return values:
example = [1193515, 7615, 40, 2993, 10095, 1782, 233, 130, 17, 553, 148, 33, 1705, 47, 4735, 2647, 550, 134, 14, 2573
label = 2
length = 23
```

Μοντέλο DNN

Έχοντας φέρει τα δεδομένα στη μορφή που χρειάζεται για να μπουν ως είσοδος στο νευρωνικό δίκτυο, χτίζουμε το μοντέλο. Το πρώτο layer του δικτύου είναι το **embedding layer** το οποίο αρχικοποιούμε με τα προ-εκπαιδευμένα word-embeddings (glove.twitter.27B.25d) και παγώνουμε τα βάρη. Ο λόγος που αρχικοποιούμε προ-εκπαιδευμένα embeddings τα οποία δεν τα ενημερώνουμε κατά την εκπαίδευση είναι επειδή τα datasets που έχουμε είναι μικρά και δεν μπορεί το μοντέλο να μάθει ποιοτικές αναπαραστάσεις των λέξεων ενώ ταυτόχρονα προσπαθεί να κάνει classification. Από την άλλη, τα embeddings που χρησιμοποιούμε είναι εκπαιδευμένα σε πολύ μεγάλα σύνολα δεδομένων και έτσι έχουν σχηματίσει έναν αξιόλογο διανυσματικό χώρο για τις λέξεις τον οποίο μπορούμε να χρησιμοποιήσουμε έτοιμο και τον οποίο ακόμα και η απλή ρύθμιση των ήδη αρχικοποιημένων βαρών μπορεί να επηρεάσει αρνητικά.

Στη συνέχεια ορίζουμε ένα layer με συνάρτηση ενεργοποίησης τη **ReLu** και τελικά το layer που θα κάνει το classification. Ο λόγος που βάζουμε μια μη γραμμική συνάρτηση ενεργοποίησης στο προ-τελευταίο layer είναι ώστε να μπορεί το δίκτυο να μοντελοποιεί πιο περίπλοκα δεδομένα και να εκπαιδεύεται σε μεγάλα datasets. Αν είχαμε γραμμικούς μετασχηματισμούς στη σειρά τότε το δίκτυο θα μάθαινε απλά γραμμικά μοντέλα, δηλαδή, θα έκανε γραμμική παλινδρόμηση και έτσι δεν θα είχε καλά αποτελέσματα στην κατηγοριοποίηση δεδομένων που δεν διαχωρίζονται γραμμικά.

Τέλος, στη forward του δικτύου δημιουργούμε **καινούριες αναπαραστάσεις** για κάθε κείμενο ως τον μέσο όρο των αναπαραστάσεων των επιμέρους λέξεων τους. Διαισθητικά, ενώ ο embedding χώρος αρχικά αφορούσε αναπαραστάσεις λέξεων, παίρνοντας το κέντρο βάρος τους ουσιαστικά έχουμε πλέον έναν embedding χώρο κειμένων, δηλαδή κάθε embedding διάνυσμα περιγράφει ένα κείμενο της συλλογής. Μια πιθανή αδυναμία που έχει αυτού του είδους η αναπαράσταση κειμένων είναι ότι χάνεται η σειρά των λέξεων που μπορεί να παίζει σημαντικό ρόλο. Επίσης, όπως είναι λογικό, υπάρχει μεγάλη μείωση της πληροφορίας αφού αρχικά κάθε λέξη ξεχωριστά είχε τη δική της αναπαράσταση και έτσι η ακρίβεια αναπαράστασης των κειμένων θα ήταν σαφέστατα μεγαλύτερη.

Διαδικασία Εκπαίδευσης

Προκειμένου να εκπαιδεύσουμε το μοντέλο που κατασκευάσαμε, φορτώνουμε τα δεδομένα και τα **χωρίζουμε σε mini-batches**. Συνήθως επιλέγουμε mini-batches μεγέθους 32, 64, 128 ή 256 όπου μεγαλύτερα προτιμώνται σε εκπαίδευση με GPU ώστε να εκμεταλλευόμαστε τις γρήγορες πράξεις πινάκων. Πολύ μεγάλα mini-batches, όμως, έχουν αρνητική επίδραση στα αποτελέσματα, επειδή μπορεί να οδηγήσουν το μοντέλο στο να συγκλίνει σε μη βέλτιστο ελάχιστο της συνάρτησης κόστους. Μικρότερα mini-batches εισάγουν θόρυβο και αυτό βοηθάει στο να βρεθεί το βέλτιστο ελάχιστο κάνοντας μικρά βήματα στο gradient descent και αποφεύγοντας το να παγιδευτούν σε τοπικά ελάχιστα. Από την άλλη βέβαια δεν πρέπει ούτε να είναι πολύ μικρά, γιατί σε αυτή την περίπτωση ο θόρυβος είναι μεγάλος και μπορεί να αποτρέπει την εύρεση του ελαχίστου.

Επιπλέον, συνήθως ανακατεύουμε τη σειρά των batches, καθώς έχει παρατηρηθεί πως το μοντέλο έτσι συγκλίνει πιο γρήγορα και επίσης εισάγοντας θόρυβο το βοηθάμε να μπορεί να γενικεύσει καλύτερα.

Ως optimizer για την εκπαίδευση ορίζουμε τον Adam και του δίνουμε να εκπαιδεύσει μόνο τις παραμέτρους που θέλουμε, δηλαδή όλες εκτός από τα embeddings. Στο παρακάτω screenshot φαίνεται το μοντέλο όπως τυπώνεται στην περίπτωση του multi-class classification.

```
BaselineDNN(
  (embedding): Embedding(1193516, 25)
  (fc): Linear(in_features=25, out_features=16, bias=True)
  (relu): ReLU()
  (clf): Linear(in_features=16, out_features=3, bias=True)
)
```

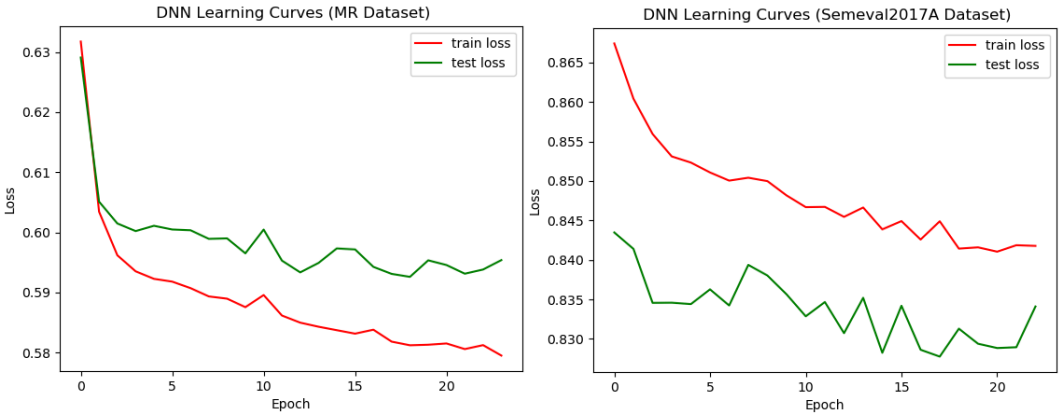
Τέλος, εκπαιδεύουμε το μοντέλο και με τα δύο datasets, χρησιμοποιώντας early stopping, όπου αν το test loss δεν βελτιώνεται για 5 συνεχόμενες εποχές τότε σταματάμε την εκπαίδευση ώστε να μην γίνει over-fit.

Αξιολόγηση

Στον παρακάτω πίνακα φαίνονται οι μετρικές accuracy, macro avg f1-score και macro avg recall που πετύχαμε στα test δεδομένα για τα δύο datasets.

Dataset	Accuracy	F1-score	Recall
MR	0.67	0.67	0.67
Semeval2017A	0.60	0.58	0.59

Επίσης, στα παρακάτω διαγράμματα φαίνονται και οι καμπύλες εκμάθησης του μοντέλου για τα δύο datasets.



Μοντέλα LSTM

Αφού εκπαιδεύσαμε ένα DNN, τώρα θα εκπαιδεύσουμε μερικά μοντέλα LSTM πειραματιζόμενοι με διαφορετικές αναπαραστάσεις των κειμένων, ώστε να δούμε ποια φαίνεται να είναι η καλύτερη. Θα χρησιμοποιήσουμε τα ακόλουθα μοντέλα/αναπαραστάσεις:

1. <u>LSTM:</u> Η αναπαράσταση του κειμένου που παίρνουμε από το LSTM αντιστοιχεί στο τελευταίο output h_N, εξαιρώντας το zero-padding.

2. <u>LSTM + Pooling:</u> Στο DNN εφαρμόσαμε mean pooling στα embeddings, οπότε μια αναπαράσταση που μπορούμε να δημιουργήσουμε είναι το να κάνουμε concatenate την έξοδο του LSTM με το mean και max pooling των embeddings, δηλαδή

$$u = [h_N||mean(E)||\max(E)]$$

Με το mean pooling αναπαριστούμε μια πρόταση ως τον μέσο όρο των αναπαραστάσεων των λέξεων της όπως εξηγήσαμε προηγουμένως, δηλαδή η έννοια μιας πρότασης συμπεριλαμβάνει κατά κάποιον τρόπο τις έννοιες όλων των λέξεων. Προσθέτοντας και το max pooling, παίρνουμε πληροφορία για τις πιο ισχυρές τιμές των λέξεων σε κάθε διάσταση των embeddings, μια αναπαράσταση που δίνει σημασία σε κάθε πρόταση με διαφορετικό τρόπο, αφού πλέον κάθε τιμή του διανύσματος της πρότασης θα προκύπτει από διαφορετικές λέξεις δίνοντας σε μια πρόταση τη σημασία των πιο σημαντικών χαρακτηριστικών των λέξεων της. Προσφέροντας τέτοιες διαφορετικές αναπαραστάσεις το

δίκτυο πιθανόν να βοηθηθεί ώστε να εξάγει την πιο χρήσιμη πληροφορία για την αναγνώριση συναισθήματος.

3. Embeddings + Attention: Μια διαφορετική αναπαράσταση που θα χρησιμοποιήσουμε είναι εφαρμόζοντας μηχανισμό attention στα embeddings, χωρίς κάποιο δίκτυο LSTM, δηλαδή η αναπαράσταση ενός κειμένου θα είναι το σταθμισμένο άθροισμα των embeddings, των λέξεων του.

4. LSTM + Attention: Ομοίως, μπορούμε να χρησιμοποιήσουμε τον μηχανισμό attention στα outputs του LSTM για να πάρουμε το σταθμισμένο άθροισμά τους.

5. BiLSTM + Pooling: Ένα άλλο μοντέλο που μπορούμε να χρησιμοποιήσουμε είναι ένα Bidirectional LSTM, το οποίο επεξεργάζεται την ακολουθία των λέξεων μιας πρότασης από δύο κατευθύνσεις. Στο output θα συμπεριλάβουμε την πληροφορία του mean και max pooling.

6. BiLSTM + Attention: Τέλος, θα χρησιμοποιήσουμε ένα Bidirectional LSTM, εφαρμόζοντας τον μηχανισμό Attention στα outputs.

Στον παρακάτω πίνακα φαίνονται τα αποτελέσματα των μοντέλων εκπαιδευμένα (με χρήση early stopping) πάνω στο MR dataset και χρησιμοποιώντας τα glove.twitter.27B.25d word embeddings. Επίσης, το μοντέλο LSTM στον κώδικα είναι ένα και μοναδικό, αλλά δέχεται ως παραμέτρους τις λειτουργίες που αναφέραμε ώστε να προσαρμοστεί στις παραλλαγές.

Μοντέλο	Accuracy	F1-score	Recall
LSTM	0.711	0.710	0.716
LSTM + Pooling	0.725	0.722	0.734
Embeddings + Attention	0.693	0.693	0.694
LSTM + Attention	0.730	0.729	0.733
BiLSTM + Pooling	0.734	0.732	0.741
BiLSTM + Attention	**0.742**	**0.740**	**0.749**

Όπως παρατηρούμε το Bidirectional LSTM με Attention τα πηγαίνει καλύτερα από τις άλλες παραλλαγές. Ο Bidirectional μηχανισμός φαίνεται να βελτιώνει κατά περίπου 0.01 το accuracy παρατηρώντας τις διαφορές μεταξύ των ίδιων αναπαραστάσεων με ή χωρίς αυτόν, αλλά ο Attention μηχανισμός φαίνεται ακόμα πιο αποτελεσματικός με βελτίωση 0.02. Επίσης, όλα αυτά τα μοντέλα τα πηγαίνουν αρκετά καλύτερα σε σχέση με το απλό DNN.

Χαρακτηριστικά BoW

Μια άλλη αναπαράσταση που θα μπορούσαμε να χρησιμοποιήσουμε είναι Bag of Words χαρακτηριστικά, όπως για παράδειγμα tf-idf, αντί για αναπαραστάσεις όπως ο μέσος όρος των word embeddings. Για την ακρίβεια αυτά είναι τα χαρακτηριστικά που χρησιμοποιούνταν ευρέως πριν τα word-embeddings, αλλά έχουν το μειονέκτημα ότι δημιουργούν μεγάλες sparse αναπαραστάσεις οι οποίες είναι σπάταλες σε μνήμη, σε αντίθεση με τα dense vectors των embeddings. Παρ' όλα αυτά υπάρχουν περιπτώσεις που μπορεί τα BoW χαρακτηριστικά να οδηγήσουν σε καλύτερα αποτελέσματα απ' ότι τα embeddings. Για παράδειγμα, όταν τα δεδομένα είναι πολύ λίγα είναι πιο εύκολο το

νευρωνικό δίκτυο να κάνει overfit με τα word-embeddings οπότε ίσως τα πάει καλύτερα με τα BoW χαρακτηριστικά. Επίσης, οι dense αναπαραστάσεις των word-embeddings είναι περίπλοκες και κρύβουν patterns τα οποία μπορεί αν μάθει ένα βαθύ νευρωνικό δίκτυο. Πιο ρηχά νευρωνικά δίκτυα όπως ένα απλό MLP ή και κλασικά μοντέλα μηχανικής μάθησης όπως ένα Gradient Boosted Decision Tree, πολύ πιθανόν να τα πάνε καλύτερα με τα BoW χαρακτηριστικά σε σχέση με τα word embeddings, ειδικά όταν τα datasets είναι μικρά.