

1 Εισαγωγή

Στην προπαρασκευή, σχεδιάσατε απλές αρχιτεκτονικές νευρωνικών δικτύων για κατηγοριοποίηση κειμένων, χρησιμοποιώντας προ-εκπαιδευμένες διανυσματικές αναπαραστάσεις λέξεων (pretrained word embeddings). Σκοπός αυτής της εργαστηριακής άσκησης είναι να εμβαθύνετε χρησιμοποιώντας Ανατροφοδοτούμενα Νευρωνικά Δίκτυα (Recurrent Neural Networks), τεχνικές Μεταφοράς Γνώσης (Transfer Learning) και μηχανισμούς Προσοχής (Attention mechanisms).

Στην προπαρασκευή του εργαστηρίου, η αρχιτεκτονική η οποία σας έχει ζητηθεί να φτιάξετε είναι η εξής:

$$e_i = \text{embed}(x_i) \quad \text{διανυσματική αναπαράσταση κάθε λέξης (embedding)} \quad (1)$$

$$u = \frac{1}{N} \sum_{i=1}^N e_i \quad \text{αναπαράσταση κειμένου: μέσος όρος embeddings} \quad (2)$$

$$r = \tanh(Wu + b) \quad \text{μη-γραμμικός μετασχηματισμός} \quad (3)$$

όπου r η διανυσματική αναπαράσταση ενός κειμένου (feature vector). Στόχος είναι να δημιουργήσετε καλύτερες αναπαραστάσεις χρησιμοποιώντας τις προαναφερθείσες τεχνικές.

2 Θεωρητικό Υπόβαθρο

2.1 Ανατροφοδοτούμενα Νευρωνικά Δίκτυα

Τα Ανατροφοδοτούμενα Νευρωνικά Δίκτυα (Recurrent Neural Networks - RNN) σε αντίθεση με άλλες αρχιτεκτονικές έχουν την ιδιότητα ότι σχηματίζουν συνδέσεις με ανάδραση (feedback). Τα περισσότερα δίκτυα απαιτούν εισόδους σταθερών διαστάσεων ενώ τα RNNs μπορούν να επεξεργάζονται με ευκολία δεδομένα μεταβλητού μήκους. Αυτή η ευελιξία, τα κάνει ιδανικά για την επεξεργασία ακολουθιών, όπως σε προβλήματα επεξεργασίας φυσικής γλώσσας. Ο τρόπος λειτουργίας τους μοιάζει με τον τρόπο που ο άνθρωπος επεξεργάζεται την γλώσσα (κειμένο, ομιλία), δηλαδή σειριακά.

Η βασική λειτουργία ενός RNN είναι η εξής: δέχεται ως είσοδο μία ακολουθία από διανύσματα $x = (x_1, x_2, \dots, x_n)$ και παράγει ως έξοδο ένα μοναδικό διάνυσμα y . Όμως, η κρίσιμη διαφορά είναι ότι σε κάθε βήμα, για την παραγωγή του αποτελέσματος, λαμβάνεται υπόψη το αποτέλεσμα του προηγούμενου βήματος. Το απλό RNN, το οποίο εξετάζουμε τώρα, έχει ορισμένες παραλλαγές [Hopfield, 1982, Elman, 1990, Jordan, 1997]. Η παραλλαγή η οποία εξετάζουμε εδώ είναι το δίκτυο Elman [Elman, 1990].

Λειτουργία Πιο συγκεκριμένα ένα RNN, επεξεργάζεται τα στοιχεία μίας ακολουθίας σειριακά (ένα-προς-ένα), εκτελώντας τον ίδιο μετασχηματισμό, σε κάθε στοιχείο της ακολουθίας. Επίσης, διατηρεί μία εσωτερική κατάσταση, η οποία λειτουργεί σαν ένα είδος μνήμης, με τη μορφή ενός διανύσματος h , το οποίο ονομάζεται *κρυφή κατάσταση* (hidden state). Σε κάθε βήμα, το RNN ενημερώνει το h , λαμβάνοντας υπόψη του την τιμή του τρέχοντος στοιχείου x_t και της προηγούμενης

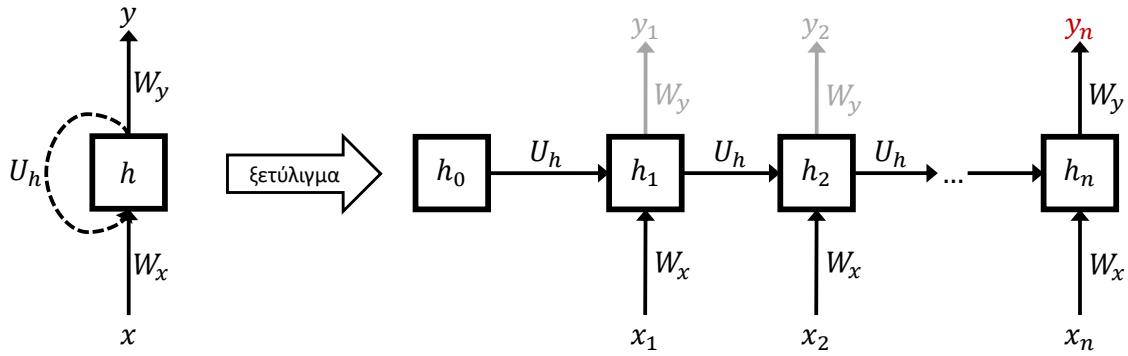


Figure 1: Διάγραμμα λειτουργίας ενός RNN. Υπάρχουν δύο τρόποι ώστε να σκέφτεται κανείς το πως λειτουργεί ένα RNN. Στην αριστερή εικόνα φαίνεται η αναδρομική λειτουργία του RNN. Το RNN δέχεται το ένα μετά το άλλο τα στοιχεία της ακολουθίας και ενημερώνει την εσωτερική ή κρυφή του κατάσταση. Ένας άλλος τρόπος αναπαράστασης είναι με το “ξετύλιγμα” του δικτύου στο χρόνο. Ουσιαστικά ένα RNN είναι ένα feed-forward NN (FFNN) με ανάδραση. Στην ξετυλιγμένη μορφή, ένα RNN μοιάζει με ένα πολυεπίπεδο FFNN.

τιμής του h . Έτσι για κάθε χρονική στιγμή t , έχουμε:

$$h_t = f_h(W_h x_t + U_h h_{t-1} + b_h) \quad (4)$$

$$y_t = f_y(W_y h_t + b_y) \quad (5)$$

όπου:

- h_t η κρυφή κατάσταση (ή κρυφό διάνυσμα), τη χρονική στιγμή t .
- x_t το διάνυσμα (εισόδου) του στοιχείου της ακολουθίας τη χρονική στιγμή t .
- y_t η έξοδος (διάνυσμα) τη χρονική στιγμή t .
- W_x, U_h, W_y τα βάρη¹ του δικτύου για τα h, x και y αντίστοιχα.
- b_h το bias για το h .
- f_x, f_h οι συναρτήσεις ενεργοποίησης (συνήθως tanh).

Η Εικόνα 1 ξεκαθαρίζει τον τρόπο λειτουργίας ενός RNN. Όπως φαίνεται και από το σχήμα, ένα RNN μπορεί να δεχθεί μία ακολουθία οποιουδήποτε μήκους. Αφού επεξεργαστεί ένα-προς-ένα τα στοιχεία, παράγει το τελικό αποτέλεσμα y_n , το οποίο είναι μία σταθερή διανυσματική αναπαράσταση για όλη την ακολουθία. Μπορείτε να σκεφτείτε αυτή την αναπαράσταση σαν τη σύνοψη της ακολουθίας. Σε αυτή την περίπτωση, εκτελούνται μόνο οι ενημερώσεις της κρυφής κατάστασης (Εξίσωση (4)) και στο τέλος παράγεται η έξοδος (Εξίσωση (5)).

Παράδειγμα Ας δούμε ένα συνηθισμένο παράδειγμα εφαρμογής ενός RNN στην επεξεργασία φυσικής γλώσσας. Σε ένα πρόβλημα κατηγοριοποίησης κειμένου, το RNN επεξεργάζεται τις λέξεις του εγγράφου, τη μία μετά την άλλη, και στο τέλος παράγει την διανυσματική αναπαράσταση του εγγράφου y_n . Η αναπαράσταση αυτή χρησιμοποιείται σαν διάνυσμα χαρακτηριστικών για την κατηγοριοποίηση του κειμένου, π.χ. βάση του συναισθηματικού του προσανατολισμού.

Πιο συγκεκριμένα, το έγγραφο αναπαριστάται από μία ακολουθία λέξεων. Κάθε λέξη αναπαριστάται από ένα διάνυσμα (word embedding) x_i , με $x_i \in R^E$, όπου E οι διαστάσεις των διανυσμάτων λέξεων. Έτσι έχουμε την ακολουθία $X = (x_1, x_2, \dots, x_T)$, όπου T το πλήθος των λέξεων

¹Συνήθως χρησιμοποιείται το W για να δηλώσει τα βάρη στην κανονικές συνδέσεις ενός δικτύου και το U για να δηλώσει τα βάρη σε αναδρομικές συνδέσεις.

στο έγγραφο. Το RNN επεξεργάζεται σειριακά τις λέξεις, διατηρώντας στο εσωτερικό του, μία περίληψη όσων έχει διαβάσει μέχρι τη χρονική στιγμή t . Στο τέλος, περιέχει την περίληψη όλης της πληροφορίας του εγγράφου και από αυτή αποτελεί την τελική διανυσματική αναπαράσταση για το έγγραφο.

2.1.1 Αμφίδρομο RNN

Ένα αμφίδρομο RNN (bidirectional RNN ή BiRNN) αποτελείται από τον συνδυασμό δύο διαφορετικών RNN, όπου το κάθε ένα επεξεργάζεται την ακολουθία με διαφορετική φορά. Το κίνητρο αυτής της τεχνικής, είναι η δημιουργία μία περίληψης του εγγράφου και από τις δύο κατευθύνσεις, ώστε να σχηματιστεί μία καλύτερη αναπαράσταση. Έτσι έχουμε ένα δεξιόστροφο RNN \vec{f} , το οποίο διαβάζει μία πρόταση από το x_1 προς x_T και ένα αριστερόστροφο RNN \overleftarrow{f} , το οποίο διαβάζει μία πρόταση από το x_T προς x_1 . Έτσι, κάθε χρονική στιγμή t , έχουμε:

$$h_i = \vec{h}_i \parallel \overleftarrow{h}_i, \quad h_i \in R^{2N} \quad (6)$$

όπου το \parallel συμβολίζει την πράξη της ένωσης δύο διανυσμάτων και N είναι οι διαστάσεις του κάθε RNN.

2.1.2 Βαθύ RNN

Όπως και με τα απλά FFNN μπορούμε να στοιχίσουμε ένα RNN σε επίπεδα για την δημιουργία βαθιών δικτύων.

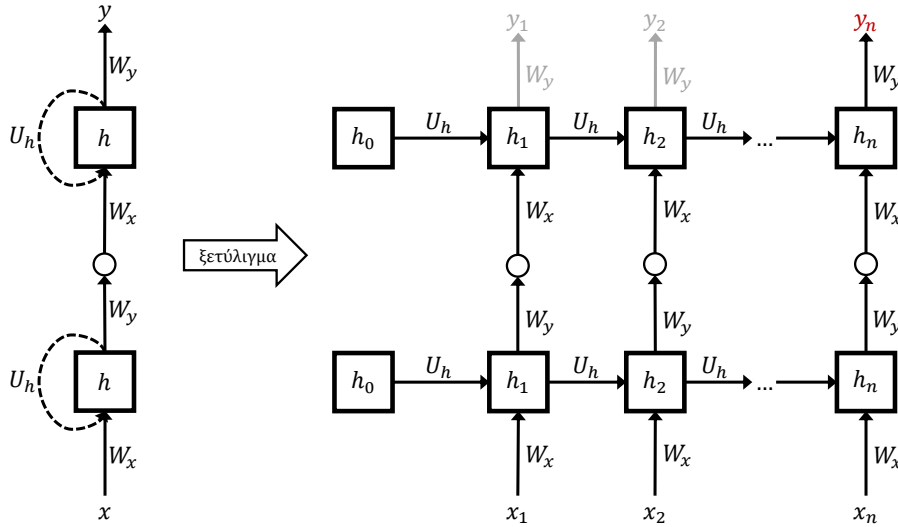


Figure 2: Ένα βαθύ RNN με δύο επίπεδα.

2.1.3 Long Short-Term Memory (LSTM)

Πλέον δεν χρησιμοποιείται σχεδόν καθόλου το απλό RNN αλλά παραλλαγές του, οι οποίες επιχειρούν να ξεπεράσουν ορισμένα προβλήματα, με σημαντικότερο αυτό του vanishing gradient κατά την διάρκεια της εκπαίδευσης του δικτύου [Bengio et al., 1994, Hochreiter et al., 2001]. Η πιο δημοφιλής παραλλαγή είναι το δίκτυο Long Short-Term Memory (LSTM) [Hochreiter and Schmidhuber, 1997].

Χρησιμοποιεί έναν εξεζητημένο μηχανισμό, ο οποίος του επιτρέπει να ξεπεράσει το πρόβλημα του RNN, σχετικά με την αναγνώριση απομακρυσμένων εξαρτήσεων. Το LSTM έχει δύο βασικές διαφορές από το απλό RNN:

- Δεν εφαρμόζει συνάρτηση ενεργοποίησης στις αναδρομικές συνδέσεις. Αυτό σημαίνει ότι οι ενημερώσεις θα είναι γραμμικές. Έτσι εγγυάται ότι τα σφάλματα (gradients), δεν θα εξαφανίζονται από την επαναληπτική εφαρμογή των ενημερώσεων (backpropagation through-time). Συνεπώς εξασφαλίζει τη ροή της πληροφορίας στο δίκτυο.
- Μηχανισμός με θύρες. Ο μηχανισμός αυτός εισάγει θύρες, οι οποίες ρυθμίζουν το πόσο θα ενημερώνεται κάθε διάνυσμα του δικτύου (εσωτερική κατάσταση, έξοδος κλπ.). Με αυτό τον τρόπο το δίκτυο αφομοιώνει και διατηρεί τις πιο σημαντικές πληροφορίες καλύτερα.

2.2 Μηχανισμός Προσοχής

Όπως είδαμε και παραπάνω, ένα RNN χρησιμοποιεί την τελευταία τιμή της εσωτερικής του κατάστασης ως την διανυσματική αναπαράσταση όλης της ακολουθίας. Η αναπαράσταση αυτή ενημερώνεται καθώς το RNN διαβάζει την ακολουθία και στο τέλος περιέχει μία σύνοψη της ακολουθίας. Όμως, ειδικά όταν η ακολουθία είναι αρκετά μεγάλη, υπάρχει η περίπτωση το δίκτυο να μην μπορεί να συγκρατήσει όλες τις σημαντικές πληροφορίες στην εσωτερική του κατάσταση. Για να αντιμετωπίσουμε το πρόβλημα στο οποίο τα RNN τείνουν να είναι “μυωπικά”, μπορούμε να χρησιμοποιήσουμε ένα μηχανισμό προσοχής (attention), ο οποίος επιχειρεί να ενισχύσει την συνεισφορά των πιο σημαντικών στοιχείων στην τελική αναπαράσταση.

Αυτό το πετυχαίνει αξιοποιώντας όλες τις ενδιάμεσες καταστάσεις του RNN. Έτσι κάθε χρονική στιγμή t , η εσωτερική κατάσταση του RNN y_i , χρησιμοποιείται ως την ερμηνεία της λέξης x_i , καθώς το RNN κωδικοποιεί κάθε λέξη βάση των συμφραζόμενων. Για την παραγωγή της διανυσματικής αναπαράστασης ολόκληρου του κειμένου, χρησιμοποιούμε το σταθμισμένο άθροισμα των ερμηνειών των λέξεων [Graves, 2013, Bahdanau et al., 2014], όπου ο συντελεστής κάθε λέξης καθορίζεται από τον μηχανισμό προσοχής. Ουσιαστικά υπολογίζει μία κατανομή πιθανότητας πάνω στα y_i και η τιμή a_i αντιστοιχεί στην σημαντικότητα κάθε λέξης. Ο μηχανισμός προσοχής, αποτελεί ένα επίπεδο του δικτύου, το οποίο εκπαιδεύεται μαζί με τα υπόλοιπα.

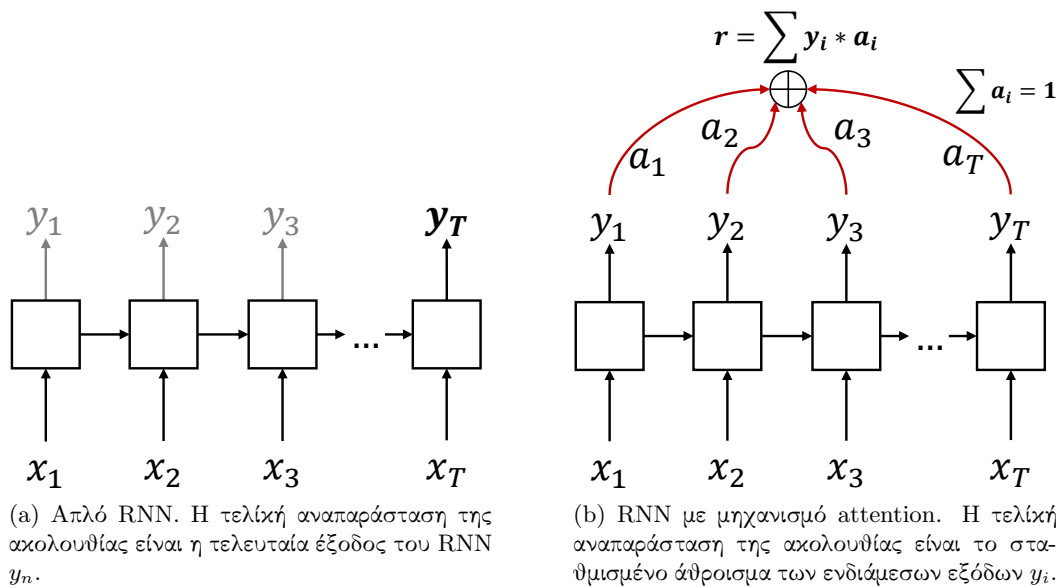


Figure 3: Σύγκριση μεταξύ του απλού RNN και ενός RNN με attention. Στο RNN με attention, Η τελική αναπαράσταση r είναι το σταθμισμένο άθροισμα όλων των εξόδων του RNN. Τα βάρη κάθε βήματος a_i , ορίζονται από το attention layer.

Παραδείγματα. Όταν η ακολουθία είναι ένα κείμενο, αυτό σημαίνει ότι ο μηχανισμός δίνει μεγαλύτερη προσοχή στις πιο σημαντικές λέξεις του κειμένου. Στο πλαίσιο της συναισθηματικής

ανάλυση κειμένου (sentiment analysis), το δίκτυο μαθαίνει να αποδίδει μεγαλύτερα βάρη στις λέξεις οι οποίες είναι καθοριστικές για τον προσδιορισμό του συναισθήματος σε ένα κείμενο. Αντίστοιχα, στο πρόβλημα της παραγωγής περιγραφών από εικόνες [Xu et al., 2015], θέλουμε να δώσουμε έμφαση σε διαφορετικές περιοχές μίας εικόνας, ανάλογα με την έννοια που θέλουμε να περιγράψουμε. Ομοίως, στο πλαίσιο της μηχανικής μετάφρασης [Bahdanau et al., 2014], όταν παράγουμε το μεταφρασμένο κείμενο, θέλουμε μετά από κάθε λέξη της μετάφρασης να δώσουμε έμφαση σε διαφορετικές λέξεις του πηγαιού κειμένου.

2.3 Μεταφοράς Γνώσης

Κατά την διαδικασία Μεταφοράς Γνώσης (Transfer Learning) εκπαιδεύουμε ένα στατιστικό μοντέλο σε ένα πρόβλημα μηχανική μάθησης και το εφαρμόζουμε (κατόπιν κατάλληλης επανεκπαίδευσης) σε μια άλλη εφαρμογή. Στα πλαίσια των νευρωνικών παρατηρήθηκε ότι τα βάρη δικτύων δικτύων εκπαιδευμένα σε τεράστια σύνολα δεδομένων για αναγνώρισης εικόνων (ImageNet), μπορούσαν να χρησιμοποιηθούν για την αρχικοποίηση δικτύων με στόχο την επίλυση άλλων προβλημάτων ανάλυσης εικόνας. Τα δίκτυα αυτά είχαν μάθει να αναγνωρίζουν πολύ γενικά χαρακτηριστικά στα χαμηλότερα επίπεδά τους, τα οποία ήταν χρήσιμα σε πολλά διαφορετικά προβλήματα. Στην απλούστερη περίπτωση μπορεί κανείς να χρησιμοποιήσει ένα προεκπαιδευμένο δίκτυο σε ένα πρόβλημα μηχανική μάθησης με πολλά δεδομένα, και να το προσαρμόσει εκπαιδεύοντάς το για λίγες εποχές μια άλλη εφαρμογή, στο οποίο συνήθως δεν υπάρχουν αρκετά δεδομένα εκπαίδευσης.

Η ίδια προσέγγιση μπορεί να εφαρμοστεί και σε προβλήματα επεξεργασίας φυσικής γλώσσας όταν έχουμε λίγα δεδομένα εκπαίδευσης. Η χρήση προεκπαιδευμένων word embeddings είναι μία μορφή Μεταφοράς Γνώσης. Όμως μπορούμε να χρησιμοποιήσουμε και μεταφορά ολόκληρου του δικτύου [Baziotis et al., 2018], το οποίο έχει μάθει να κωδικοποιεί και τα συμφραζόμενα των λέξεων (context).

3 Ερωτήματα

Ερώτημα 1

1.1 Υπολογίστε την αναπαράσταση κάθε πρότασης u (Εξίσωση 2) ως την συνένωση (concatenation) του μέσου όρου (mean pooling) και του μεγίστου ανά διάσταση (max pooling) των word embeddings κάθε πρότασης, $E = (e_1, e_2, \dots, e_N)$.

$$u = [\text{mean}(E) || \text{max}(E)] \quad (7)$$

1.2 Τι διαφορά(ές) έχει αυτή η αναπαράσταση με την αρχική; Τι παραπάνω πληροφορία θα μπορούσε να εξάγει; Απαντήστε συνοπτικά.

Ερώτημα 2

Σε αυτό το ερώτημα θα πρέπει να χρησιμοποιήσετε ένα LSTM για να κωδικοποιήσετε την πρόταση. Το LSTM θα διαβάζει τα word embeddings e_i και θα παράγει μία νέα αναπαράσταση για κάθε λέξη h_i , η οποία θα λαμβάνει υπόψη της και τα συμφραζόμενα. Μπορείτε να παραλείψετε τον μη-γραμμικό μετασχηματισμό (Εξίσωση 3).

2.1 Χρησιμοποιήστε την τελευταία έξοδο του LSTM h_N ως την αναπαράσταση του κειμένου u .

Προσοχή: πρέπει να χρησιμοποιήσετε το πραγματικό τελευταίο timestep, εξαιρώντας τα zero-padded timesteps. Χρησιμοποιήστε τα πραγματικά μήκη των προτάσεων, όπως υπολογίσατε τον μέσο όρο στην άσκηση της προπαρασκευής.

2.2 Χρησιμοποιήστε ως αναπαράσταση του κειμένου u : την συνένωση της τελευταίας εξόδου του LSTM h_N , του μέσου όρου των εξόδων του LSTM, και του μεγίστου ανά διάσταση των εξόδων του LSTM.

$$u = [h_t || \text{mean}(E) || \text{max}(E)] \quad (8)$$

Ερώτημα 3

Σε αυτό το ερώτημα θα χρησιμοποιήσετε ένα μηχανισμό attention. Μπορείτε να υλοποιήσετε ένα δικό σας μηχανισμό ή να χρησιμοποιήσετε έτοιμη μία υλοποίηση ².

3.1 Χρησιμοποιήστε ένα μηχανισμό attention, για να υπολογίσετε την αναπαράσταση ενός κειμένου, ως το σταθμισμένο άθροισμα των word embeddings.

$$v_i = \tanh(W\mathbf{e}_i + b) \quad (9)$$

$$a_i = \frac{\exp(v_i)}{\sum_{t=1}^N \exp(v_t)} \quad (10)$$

$$\mathbf{u} = \sum_{i=1}^N a_i \mathbf{e}_i \quad (11)$$

3.2 Χρησιμοποιήστε ένα μηχανισμό attention, για να υπολογίσετε την αναπαράσταση ενός κειμένου, ως το σταθμισμένο άθροισμα των εξόδων ενός LSTM.

$$v_i = \tanh(W\mathbf{h}_i + b) \quad (12)$$

$$a_i = \frac{\exp(v_i)}{\sum_{t=1}^N \exp(v_t)} \quad (13)$$

$$\mathbf{u} = \sum_{i=1}^N a_i \mathbf{h}_i \quad (14)$$

Ερώτημα 4

Σε αυτό το ερώτημα θα υλοποιήσετε τα ζητούμενα του ερωτήματος 3 αλλά χρησιμοποιώντας ένα **αμφίδρομο LSTM** για να κωδικοποιήσετε την πρόταση. Προσοχή, στα zero-padded timesteps. Το τελευταίο timestep θα υπολογίζεται ξεχωριστά ανά κατεύθυνση!

4.1 Όπως στο ερώτημα 3.1, αλλά χρησιμοποιώντας αμφίδρομο LSTM.

4.2 Όπως στο ερώτημα 3.2, αλλά χρησιμοποιώντας αμφίδρομο LSTM.

Ερώτημα 5

5.1 Αποθηκεύστε σε ένα checkpoint στο δίσκο το μοντέλο με το χαμηλότερο loss στο validation set. Χρησιμοποιήστε το αποθηκευμένο checkpoint και υπολογίστε τις προβλέψεις του δικτύου στο validation set, αποθηκεύοντας τα αποτελέσματα σε ένα .TXT αρχείο.

5.2 Χρησιμοποιήστε το αποθηκευμένο checkpoint και οπτικοποιήστε τα βάρη των κατανομών του attention. Μπορείτε να χρησιμοποιήσετε το NeAt-vision³.

5.3 Συγκρίνετε τα αποτελέσματα των κατανομών του attention, ανάμεσα στις εξόδους ενός μοντέλου εκπαιδευμένου όπως στο ερώτημα 3.1 και το 3.2, αντίστοιχα. Τι παρατηρείτε; Βρείτε 3 ενδιαφέροντα παραδείγματα και σχολιάστε τα.

Ερώτημα 6

Χρησιμοποιήστε χαρακτηριστικά Bag-of-Words (BoW) σε συνδυασμό με τα χαρακτηριστικά που εξάγει το νευρωνικό δίκτυο (αναπαράσταση). Για τον υπολογισμό των BoW χαρακτηριστικών, μπορείτε να χρησιμοποιήσετε το scikit-learn⁴ (CountVectorizer ή TfidfVectorizer).

6.1 Υπολογίστε την τελική αναπαράσταση ως συνάρτηση των δύο επιμέρους αναπαραστάσεων (με όποιον τρόπο θέλετε). Επίσης, είστε ελεύθεροι να δημιουργήσετε με όποιον τρόπο θέλετε τα BoW χαρακτηριστικά.

6.2 (bonus) Σε ποιες περιπτώσεις θα μπορούσαν τα BoW χαρακτηριστικά να οδηγήσουν σε καλύτερα αποτελέσματα σε σχέση με αναπαραστάσεις όπως ο μέσος όρος των word embeddings?

²<https://gist.github.com/cbaziotis/94e53bdd6e4852756e0395560ff38aa4>

³<https://github.com/cbaziotis/neat-vision>

⁴https://scikit-learn.org/stable/tutorial/text_analytics/working_with_text_data.html

Ερώτημα 7 (bonus)

Σε αυτό το ερώτημα θα εφαρμόσετε μεταφορά γνώσης για να βελτιώσετε τις επιδόσεις ενός μοντέλου για τον υπολογισμό έντασης συναισθήματος. Ως source dataset, θα χρησιμοποιήσετε αυτό του Semeval 2017 Task4-A⁵ [Rosenthal et al., 2017] (από την προπαρασκευή) και ως target dataset αυτά του EI-reg (emotion intensity regression) Subtask από το SemEval-2018 Task 1: Affect in Tweets⁶. Συμβουλευτείτε το [Baziotis et al., 2018].

7.1 Υλοποιήστε ένα regression μοντέλο για το EI-reg Subtask. Χρησιμοποιείτε την αρχιτεκτονική του ερωτήματος 4.2.

7.2 Εκτελέστε μεταφορά γνώσης σε κάθε ένα από τα συναισθήματα του EI-reg Subtask (anger, fear, joy, and sadness). Αρχικοποιήστε τα βάρη του νευρωνικού σας, εξαιρουμένου του τελευταίου layer⁷, από ένα μοντέλο προεπαιδευμένο στο Semeval 2017 Task4-A.

Βοηθητικό Υλικό

Το παρακάτω υλικό αποτελείται από blog posts, τα οποία είναι αρκετά αφαιρετικά και θα σας βοηθήσουν στην κατανόηση των εννοιών που επιγραμματικά καλύφθηκαν στην Θεωρητικό Υπόβαθρο.

- Λειτουργία των RNNs [Karpathy, 2015].
- Λειτουργία των LSTMs [Olah, 2015].
- Μηχανισμοί attention [Olah, 2016].
- Μεταφορά γνώσης για επεξεργασία φυσικής γλώσσας [Ruder, 2018a]. Επίσης συμβουλευτείτε την σχετική εργασία της ομάδας μας [Baziotis et al., 2018].
- Ιστορική αναδρομή της επεξεργασία φυσικής γλώσσας με την χρήση βαθιών νευρωνικών δικτύων [Ruder, 2018b]. Το συγκεκριμένο άρθρο δεν σχετίζεται με την εργασία, όμως θα σας δώσει μία εικόνα για το πως συνδέονται οι τεχνικές που χρησιμοποιήσατε στην πρόσφατη πορεία του πεδίου.

Παραδοτέα

Στα ερωτήματα 1-6 μπορείτε να χρησιμοποιήσετε ένα από τα δύο datasets τα οποία αναφέρονται στην προπαρασκευή. Διαλέξτε όποιο θέλετε απλά αναφέρετέ το στην αναφορά σας. Για κάθε παραλλαγή του μοντέλου, αναφέρετε στην αναφορά σας τις επιδόσεις του μοντέλου στις μετρικές: accuracy, F1_score (macro average), recall (macro average). Τα ερωτήματα 6.2, 7.1, 7.2 είναι προαιρετικά.

Σε ότι αφορά την βαθμολόγηση της άσκησης, δεν θα αξιολογηθείτε ως προς τις επιδόσεις του μοντέλου σας, αλλά ως προς την ορθότητα των απαντήσεων σας. Θα πρέπει να παραδώσετε τα εξής:

1. Σύντομη αναφορά (σε pdf ή jupyter notebook) που θα περιέχει τις απαντήσεις σε κάθε ερώτημα και τα αντίστοιχα αποτελέσματα.
2. Κώδικας Python, συνοδευόμενος από σύντομα σχόλια.

Συγκεντρώστε τα (1) και (2) σε ένα .zip αρχείο το οποίο πρέπει να αποσταλεί μέσω του my-courses.ntua.gr.

⁵<http://alt.qcri.org/semeval2017/task4/index.php?id=data-and-tools>

⁶<https://competitions.codalab.org/competitions/17751>

⁷Δηλαδή μόνο τα: Embedding, LSTM, Attention

References

- [Bahdanau et al., 2014] Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv:1409.0473 [cs, stat]*. 02127.
- [Baziotis et al., 2018] Baziotis, C., Nikolaos, A., Chronopoulou, A., Kolovou, A., Paraskevopoulos, G., Ellinas, N., Narayanan, S., and Potamianos, A. (2018). Ntua-slp at semeval-2018 task 1: Predicting affective content in tweets with deep attentive rnns and transfer learning. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 245–255.
- [Bengio et al., 1994] Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.
- [Elman, 1990] Elman, J. L. (1990). Finding Structure in Time. *Cognitive Science*, 14(2):179–211. 08621.
- [Graves, 2013] Graves, A. (2013). Generating Sequences With Recurrent Neural Networks. *arXiv:1308.0850 [cs]*. 00570.
- [Hochreiter et al., 2001] Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J., et al. (2001). Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.
- [Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.
- [Hopfield, 1982] Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558. 18706.
- [Jordan, 1997] Jordan, M. I. (1997). Serial Order: A Parallel Distributed Processing Approach. In *Advances in Psychology*, volume 121, pages 471–495. Elsevier. 01033.
- [Karpathy, 2015] Karpathy, A. (2015). The unreasonable effectiveness of recurrent neural networks. <http://karpathy.github.io/2015/05/21/rnn-effectiveness/>.
- [Olah, 2015] Olah, C. (2015). Understanding lstm networks. <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>.
- [Olah, 2016] Olah, C. (2016). Attention and augmented recurrent neural networks. <https://distill.pub/2016/augmented-rnns/>.
- [Rosenthal et al., 2017] Rosenthal, S., Farra, N., and Nakov, P. (2017). SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation*, SemEval ’17, Vancouver, Canada. Association for Computational Linguistics.
- [Ruder, 2018a] Ruder, S. (2018a). Nlp’s imagenet moment has arrived. <http://ruder.io/nlp-imagenet/>.
- [Ruder, 2018b] Ruder, S. (2018b). A review of the recent history of natural language processing. <http://ruder.io/a-review-of-the-recent-history-of-nlp/>.
- [Xu et al., 2015] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A. C., Salakhutdinov, R., Zemel, R. S., and Bengio, Y. (2015). Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *ICML*, volume 14, pages 77–81. 01037.