Καρακόλης Ιωάννης sdi1800065

Ασκ2:

Ασκ 2)

$f(x, y^*, m, b) = (\max(0, (x \cdot m + b)) - y^*)^2$

$k = x \cdot m$
$a = k + b$
$y = \max(0, a)$
$f = (y - y^*)^2$

Έστω $x = 1,\ y^* = 1,\ m = 2,\ b = 1$
$k = 2,\ y = 3,\ f = 4$

Local gradients:

$\frac{\partial k}{\partial x} = m = 2 \qquad \frac{\partial k}{\partial m} = x = 1$

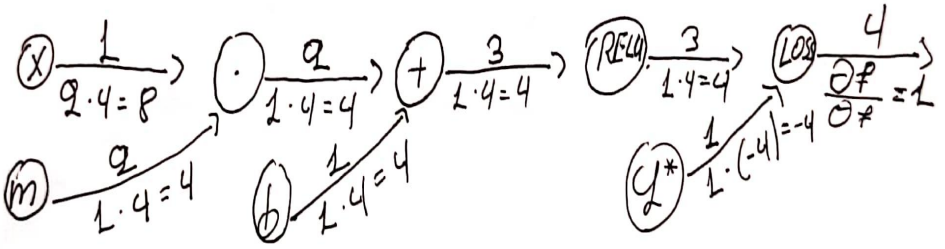
$\frac{\partial a}{\partial k} = 1 \qquad \frac{\partial a}{\partial b} = 1$

$\frac{\partial y}{\partial a} = 1 \cdot (a > 0) = 1$

$\frac{\partial f}{\partial y} = 2(y - y^*) \cdot (1 - 0) = 4$

$\frac{\partial f}{\partial y^*} = 2(y - y^*) \cdot (0 - 1) = -4$

upstream * local = downstream



Report:

1. Loading and exploring the dataset && Data pre-processing:

Αρχικά έχοντας διαβάσει τα datasets κάνω μια διαδικασία καθαρισμού των tweet , αφαιρώ δηλαδή ειδικούς χαρακτήρες , links tags και mentions τα οποία δεν βοηθάν το νευρωνικό και στην συνέχεια κάνοντας tokenize φτιάχνω μέσω του glove τα train/test set που θα χρησιμοποιηθούν στην συνέχεια . Επέλεξα τα pretrained του glove και για τα δυο μοντέλα καθως έχουμε από την πρώτη εργασία μια εικόνα πάνω στους vectorizer, όπου και φαίνεται να δίνουν καλυτέρα αποτελέσματα καθως ο τρόπος που χρησιμοποιούνται τα pre trained με το average για την δημιουργία ίσων διαστάσεων που είναι απαραίτητο στα γραμμικά μοντέλα χάνει αρκετή πληροφορία και οδηγεί σε αναμενόμενα μικρότερα scores.

2. Experimenting with different choices:

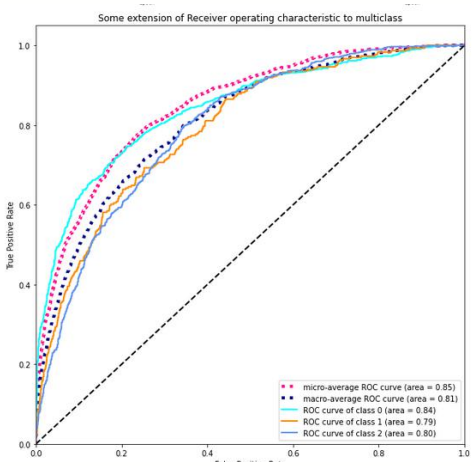
Χρησιμοποιώ δυο παρόμοια μοντέλα με τις εξής διαφορές :

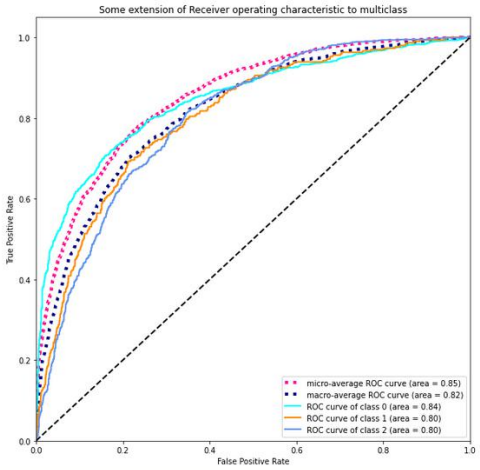
Το δεύτερο μοντέλο είναι μια πιο deep υλοποίηση καθως έχει ένα παραπάνω relu layer , ένα παραπάνω linear layer αλλά και ένα SoftMax layer στο output. Το τελευταίο δεν χεριάζετε στο πρώτο μοντέλο καθως χρησιμοποιεί την CrossEntropyLoss() η οποία κάνει από μονή της την διαδικασία του softmax σε αντίθεση με την MSELoss() που χρησιμοποιεί το δεύτερο μοντέλο. Η CrossEtropyLoss δέχεται όπως είναι το batch και το output από το linear layer που επιστρέφει το πρώτο μοντέλο ενώ στο δεύτερο μοντέλο πρέπει να γίνει μια διαδικασία μετατροπής του batch (label_binarize) κατάλληλη για την MSEloss().Όσο αναφορά τους optimizer επέλεξα nadam για το πρώτο μοντέλο και adam για το δεύτερο καθως φαίνεται να έχουν τα καλυτέρα αποτελέσματα κατά περίπτωση με μικρές διαφορές μεταξύ τους ωστόσο για όλες τις δοκιμές το learning rate ήταν πάντα 0.0001 καθως οποιαδήποτε προσπάθεια για αλλαγή έδινε εικόνες μεγάλου overfit η underfit η και εντελώς απρόσμενα αποτελέσματα. Δοκιμάζοντας άλλες παραμέτρους όπως weight_decay δεν έδωσαν καλύτερα αποτελέσματα. Τέλος για τον αριθμό των node στο πρώτο μοντέλο κατέληξα στα καλύτερα αποτελέσματα με H1 = 90 , H2=60 και παρομοίως στο δεύτερο με H1=90 H2=60 και H3=30.

3. Presenting the results

Μέσω διαφόρων γραφικών παραστάσεων αλλά και του classification report της sklearn φαίνονται εύκολα οι διαφορές μεταξύ των υλοποιήσεων των μοντέλων με το μοντέλο της πρώτης εργασίας να παρουσιάζει τα καλύτερα αποτελέσματα αφού είναι και το μόνο με tf_idf vectorizer όπως αναφέρθηκε και παραπάνω.

Η διαφορά των δυο μοντέλων της δεύτερης εργασίας στο ROC Curve είναι μικρή καθως το πιο βαθύ μοντέλο συμπεριφέρεται ελαφρώς καλύτερα στο αρνητικό συναίσθημα (class 1) από το πρώτο μοντέλο





Στο classification report φαίνεται όμως πως ο vectorizer σε σύγκριση με την συγκεκριμένη υλοποίηση του glove με το average έχει εμφανώς καλύτερα αποτελέσματα στο αρνητικό συναίσθημα και από τα δυο μοντέλα της δεύτερης εργασίας

Homework 1 model

	precision	recall	f1-score	support
0	0.77	0.79	0.78	1065
1	0.68	0.38	0.49	296
2	0.68	0.76	0.72	921
accuracy			0.72	2282
macro avg	0.71	0.64	0.66	2282
weighted avg	0.72	0.72	0.72	2282

Second neural net model

	precision	recall	f1-score	support
0	0.78	0.72	0.75	1065
1	0.47	0.25	0.33	296
2	0.62	0.77	0.68	921
accuracy			0.68	2282
macro avg	0.62	0.58	0.59	2282
weighted avg	0.67	0.68	0.67	2282

Τέλος η δοκιμή που ζητήθηκε με το όρισμα reduction=’sum’ έχει ελάχιστα μειωμένο accuracy αλλά το loss ανεβαίνει κατά πολύ.