Exercise 2
Deep Learning 097200

Roni Fridman – ID:205517097

Gadiel Eitan – ID:211879051

1. Data Loading and Preprocessing:

- We decided to use GloVe pre-trained embedding(Google News 6B)
  . Also tried word2vec embeddings(English Wiki), which gave even worst results.
- After trying out different dimensions lengths, we decided to go with the 100d version (gave us the best validation accuracy for the same hyperparameters = 0.429)
- Because the tweets contain a lot of noisy use of punctuation (slangs, emojies etc…) we decided to take only the words in every tweet. We used a regular expression for that: \w+
- We used word2vec in order to tokenize the sentences into tensors. We then fine-tuned our word vocabulary combining our embeddings and words of our training and test set (because if the word is not in the vocabulary, it will not be recognized and the model will crash).
  - Note : we didn't use word from the test set while training, only while using predict.
- We then found the longest tweet in both out training and test sets, which was 131, and used padding in order to get the tokens We then configured this maximum length into our functions, and should we meet a sentence longer then 131, we slice it into 131.
- We took 3000 samples from our train set for validation which is about 21% of the entire train set.

2. Architecture : LSTM with hidden size of 128, with two LSTM cells.

The input size is the same as the dimension of the GloVe embeddings we use : 100d. We used dropout of 0.4 to avoid overfitting.

We then used ReLU to break linearity, instead of sigmoid function.

After that we use one fully connected layer, ReLU again, and then one more linear layer that outputs three dimensions. Those three go through a LogSoftmax layer, to get their probabilities.

3. We used the NLL - loss function.

4. Optimization: we ran a simulation calculating the validation accuracy using different optimizer. We tried: Adam, RMSDrop and SGD.

Out of the three, Adam preformed better than RMSDrop considerably, and slightly better then SGD.

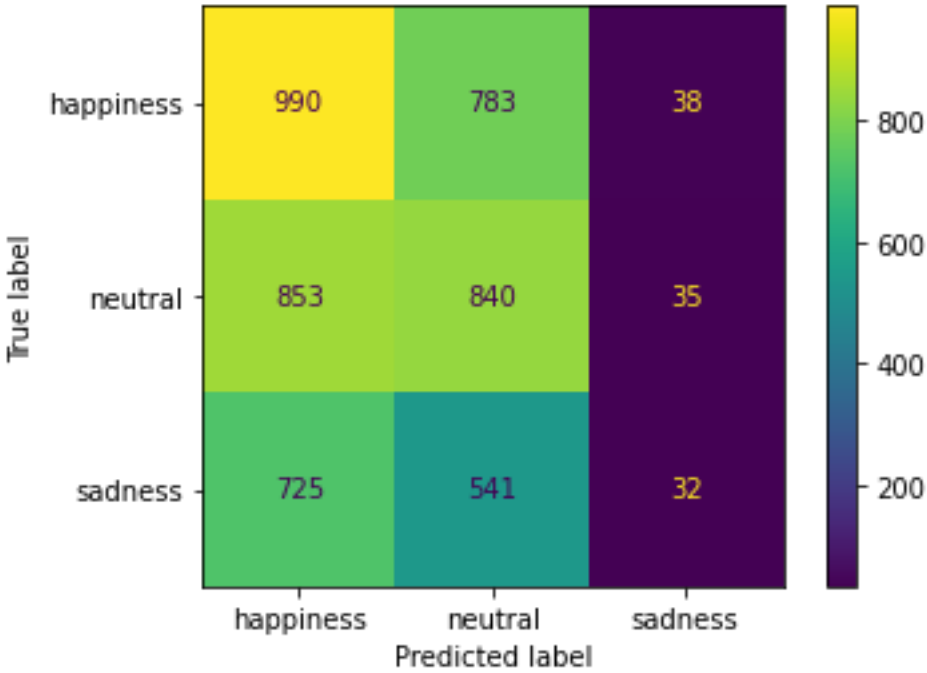
5.Regularization : We used dropout with p=0.4, which was the best out of the values: 0.1,0.2,0.3,0.4,0.05

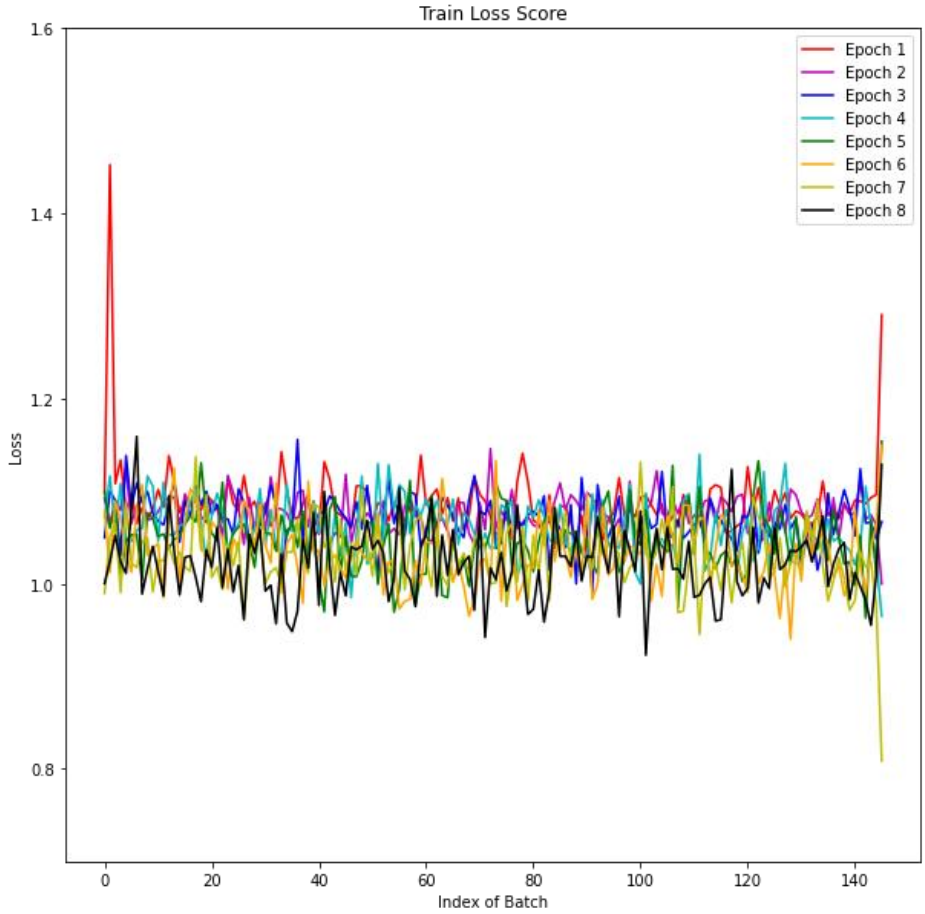
6. We later ran another simulation to see the accuracy based on learning rates and epochs. The results were:

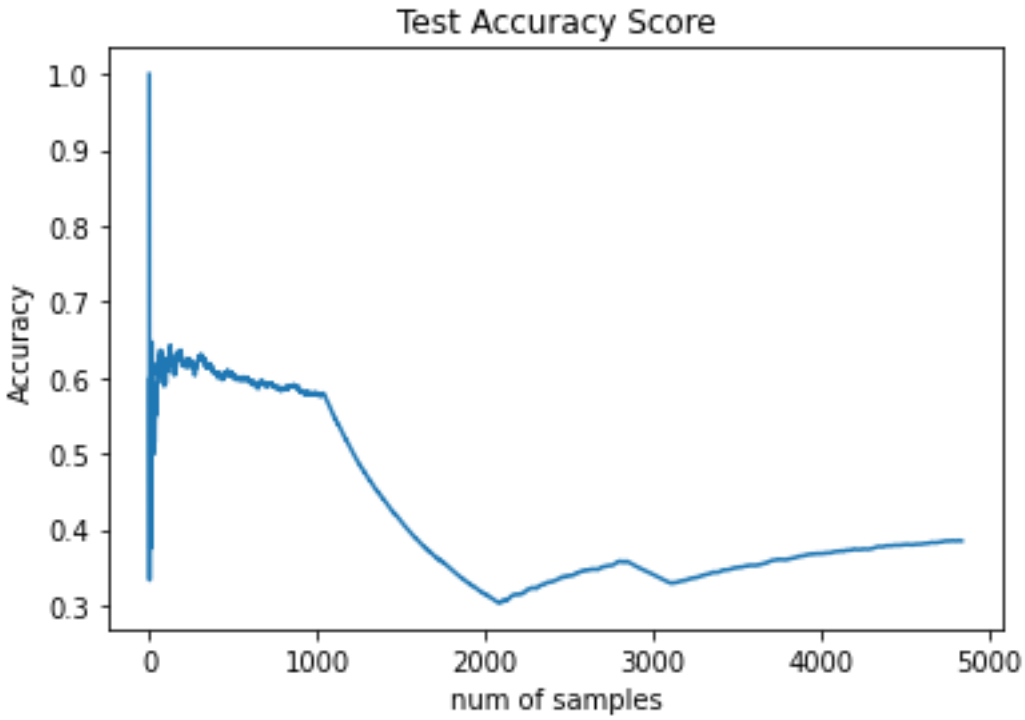
learning rate\ epochs	5	10	15	20
0.001	0.418333	0.41666	0.425666	0.415
0.002	0.426	0.438	0.422333	0.41
0.003	0.428	0.41733	0.41666	0.414
0.004	0.432333	0.419	0.41	0.393
0.005	0.423666	0.434	0.422	0.401666
0.006	0.420333	0.416	0.416	0.40666
0.007	0.411	0.408666	0.358333	0.418
0.008	0.352333	0.425666	0.419	0.408
0.009	0.428333	0.426666	0.409333	0.378
0.01	0.417	0.40666	0.37666	0.399
0.02	0.391666	0.401333	0.422666	0.411333
0.03	0.378	0.378	0.42333	0.35233
0.04	0.398333	0.384	0.379666	0.378
0.05	0.382	0.378	0.378	0.39033
0.06	0.378	0.378	0.377666	0.384
0.07	0.35222	0.386333	0.378	0.378
0.08	0.35233	0.352333	0.378	0.378
0.09	0.378	0.352333	0.378	0.378
0.1	0.352333	0.352333	0.352333	0.36333

As we can see, around 10 epochs gave the best results, and the best learning rates were 0.002 and 0.005 – but we chose 0.002 after running those two alone, seeing which had the best expected result. We later found out that 8 epochs gave the best result.

Graphs:


happiness
neutral
sadness
True label
990
783
38
853
840
35
725
541
32
800
600
400
200
happiness
neutral
sadness
Predicted label


Train Loss Score
Epoch 1
Epoch 2
Epoch 3
Epoch 4
Epoch 5
Epoch 6
Epoch 7
Epoch 8
Loss
1.6
1.4
1.2
1.0
0.8
0
20
40
60
80
100
120
140
Index of Batch



Conclusion : We think RNN in general aren't very good for sentiment analysis since it required a large amount of data from the same field (using embedding from places like Wikipedia, News, or other sources with "clean" and formal English, makes the system preform poorly) so the best case scenario will be to have a large training set from social media the was embedded by us personally.

We can see that our network incline towards 'sadness' and we notice that sad people tent to write in much proper English then others, which makes them a good fit for our GloVe embeddings.

In general , we had fun, but never again.