CS772 - DL4NLP
Sentiment analysis

Group - 24

170050003 - Jigar Makwana
170050031 - Shashank Kumar
170050052 - Devki Nandan Malav
170050057 - Arpit Menaria

13 March 2021

Problem Description

Continuing from the previous assignment,

1. Introduce hidden layer(s)
2. Use sigmoid as well as RELU for the hidden neurons and study the difference.
3. Develop a good GUI for input mechanism.
4. Use word2vec, GloVe, FastText for word embedding and compare results.
5. Solve the data unbalance problem (class V has 65% of the data).
6. Achieve at least 85% accuracy (currently for many groups it is in the range of 40s).

FFNN details

```
Layer (type)                 Output Shape              Param #
=================================================================
Embedding (Embedding)        (None, 15, 300)           5404200
flatten (Flatten)            (None, 4500)              0
dropout (Dropout)            (None, 4500)              0
hidden_1(Dense)              (None, 256)               1152256
dense (Dense)                (None, 5)                 1285
softmax (Activation)         (None, 5)                 0
=================================================================
Total params: 6,557,741
Trainable params: 1,153,541
Non-trainable params: 5,404,200
```

Library Functions Used

- sklearn, tensorflow.keras for the neural-net
- gensim for the loading and reading embeddings
- csv, pandas, nltk, numpy, os, random, re for preprocessing and other calculative tasks
- flask for GUI (web app)

Sigmoid Functions Used

Score are in order (Accuracy, Precision, Recall, F1 score)

	Word2Vec	glove	Fasttext
1 hidden layer	precision recall f1-score 1 0.58 0.53 0.56 2 0.20 0.11 0.14 3 0.35 0.15 0.21 4 0.30 0.05 0.09 5 0.70 0.95 0.80 accuracy 0.65	precision recall f1-score 1 0.54 0.53 0.54 2 0.21 0.05 0.08 3 0.33 0.14 0.20 4 0.30 0.11 0.16 5 0.70 0.93 0.80 accuracy 0.64	precision recall f1-score 1 0.61 0.54 0.57 2 0.14 0.00 0.01 3 0.31 0.29 0.30 4 0.38 0.06 0.11 5 0.70 0.94 0.81 accuracy 0.65

Imbalance handling with Sigmoid

Score are in order (Accuracy, Precision, Recall, F1 score)

	Word2Vec			glove			Fasttext		
1 hidden layer		precision	recall	f1-score		precision	recall	f1-score		precision	recall	f1-score
	1	0.51	0.53	0.52	1	0.53	0.52	0.53	1	0.47	0.55	0.51
	2	0.16	0.24	0.19	2	0.20	0.17	0.18	2	0.13	0.28	0.18
	3	0.21	0.38	0.27	3	0.22	0.25	0.24	3	0.20	0.37	0.26
	4	0.26	0.28	0.27	4	0.23	0.22	0.23	4	0.27	0.19	0.23
	5	0.84	0.67	0.74	5	0.77	0.79	0.78	5	0.84	0.67	0.75
	accuracy			0.54	accuracy			0.59	accuracy			0.53

Note: Data imbalance problem is handled by oversampling, undersampling and class weights inverse proportional to class samples but finally we settled on oversampling method by duplicating minority classes. Simultaneous over-sampling and under-sampling will be explored in next step.

Relu Functions Used

Score are in order (Accuracy, Precision, Recall, F1 score)

	Word2Vec	glove	Fasttext
1 hidden layer	precision recall f1-score 1 0.62 0.51 0.56 2 0.18 0.10 0.13 3 0.28 0.23 0.25 4 0.25 0.15 0.19 5 0.74 0.89 0.81 accuracy 0.63	precision recall f1-score 1 0.59 0.47 0.52 2 0.21 0.09 0.13 3 0.28 0.18 0.22 4 0.27 0.16 0.20 5 0.71 0.90 0.80 accuracy 0.62	precision recall f1-score 1 0.54 0.60 0.57 2 0.16 0.03 0.04 3 0.34 0.20 0.25 4 0.38 0.09 0.15 5 0.71 0.94 0.81 accuracy 0.65

Imbalance handling with Relu

Score are in order (Accuracy, Precision, Recall, F1 score)

	Word2Vec	glove	Fasttext
1 hidden layer	precision recall f1-score 1 0.54 0.54 0.54 2 0.18 0.10 0.13 3 0.25 0.19 0.22 4 0.25 0.20 0.22 5 0.74 0.84 0.79 accuracy 0.61	precision recall f1-score 1 0.51 0.48 0.49 2 0.17 0.15 0.16 3 0.24 0.19 0.21 4 0.25 0.19 0.22 5 0.75 0.83 0.79 accuracy 0.60	precision recall f1-score 1 0.55 0.51 0.53 2 0.19 0.11 0.14 3 0.22 0.30 0.26 4 0.27 0.26 0.27 5 0.78 0.79 0.79 accuracy 0.59

Note: Data imbalance problem is handled by oversampling, undersampling and class weights inverse proportional to class samples but finally we settled on oversampling method by duplicating minority classes. Simultaneous over-sampling and under-sampling will be explored in next step.

Confusion Matrix for Best Model

This is the confusion matrix for the model with class imbalance handling, single hidden layer with 256 neurons, sigmoid activation and glove embeddings.

	1	2	3	4	5
1	665	139	167	93	207
2	155	104	160	74	137
3	145	87	228	182	269
4	73	80	204	306	741
5	209	109	269	648	4549

Qualitative Analysis with some examples

“Rado is very expensive watch but it is very classy.”

- Predicted class - 3
- Probabilities = [0.05687244, 0.011874935, 0.5287177, 0.14924818, 0.25328678]

“Titan is very cheap watch, also shows time accurate.”

- Predicted class: 2
- Probabilities = [0.09363254, 0.37257537, 0.25029543, 0.22489668, 0.0586]

“Titan’s watch are cheap yet shows accurate time.”

- Predicted class: 1
- Probabilities = [0.7245896, 0.03729118, 0.121905275, 0.066225745, 0.04998817]

GUI Interface

Implemented using Flask and allows dynamic model selection based on embeddings and activations