Problem Set 09

Wen Sheng - April 29, 2016

Approach Description

I use Word Embedding techniques as the first layer in my final model to do pre-processing, then use Recurrent Neural Network(RNN) Model to fit and predict classification of reviews. As we know, compared to other tradition neural networks, RNN is good at handle variable length input. Also I did a lot of Single Variable Comparison experiment for tuning the RNN model to achieve my highest classification rate of 0.8745.

- Word Embedding Techniques
- Text Tokenizer
- Recurrent Neural Network
- Experiment using Single Variable Comparison on 5 critical tuning choices in the model construction

2. Final Model Summary

2.1 Overall model description

I build the model with an embedding layer to accept max length of 250 words for each review with an dropout of 0.25; Then use a type of LSTM RNN model layer and one dense layer of (256 nodes, dropout 0.25, activation 'relu'); The a dense layer with 'sigmoid' activation. The model optimizer is using Adadelta.

```
summary
____________________________________________________________________________________________________
Initial input shape: (None, 5000)
____________________________________________________________________________________________________
Layer (name)                     Output Shape                  Param #
____________________________________________________________________________________________________
Embedding (embedding)            (None, 250, 32)               160000
Dropout (dropout)                (None, 250, 32)               0
LSTM (lstm)                      (None, 16)                    3136
Dense (dense)                    (None, 256)                   4352
Dropout (dropout)                (None, 256)                   0
Activation (activation)          (None, 256)                   0
Dense (dense)                    (None, 1)                     257
Activation (activation)          (None, 1)                     0
____________________________________________________________________________________________________
Total params: 167745
```

2.2 Tokenize and 50 Top weight token words

After using Tokenizer to fit on negative and positive reviews, I get 5000 tokens and represent here with 50 highest weight words which are listed here:

('loyalities', 102698)	('sathla', 102686)	("ernest's", 102672)	('someome', 102659)
('bootleggers', 102697)	('ballz', 102685)	('braiding', 102671)	('"fix", 102658)
('montevallo', 102696)	('mopping', 102684)	('comfirming', 102670)	('brillian', 102657)
('healthfood', 102695)	('arisotle', 102683)	('rectified', 102669)	('reacheable', 102656)
('develolpment', 102694)	('friedn', 102682)	('lindberg', 102668)	('aamc', 102655)
('moking', 102693)	('caboose', 102681)	('gallico', 102667)	('clearl', 102654)
('chockful', 102692)	('jeuvenile', 102680)	('stingers', 102666)	('wheedle', 102653)
('cerevantes', 102691)	('restak', 102679)	('untimately', 102665)	('shim', 102652)
('reverences', 102690)	('mightiest', 102678)	("eng's", 102664)	("cervera's", 102651)
('unapralleled', 102689)	("susanna's", 102677)	("rip'", 102663)	('preiss', 102650)
("knowlson's", 102688)	('memorites', 102676)	('vicously', 102662)	('logarithmically', 102649)
('unspiced', 102687)	("vernes'", 102675)	("wallpaper's", 102661)	('noteably', 102648)
	('rothenberg', 102674)	('tannhauser', 102660)	('jocularity', 102647)
	("photius'", 102673)		

3. Experiments & Model Tuning

Without special specification, the model used in experiments is specified here:

- Experiment Model Description Here:
- Embedding Layer (drop out 0.25)
- SimpleRNN layer
- Dense Layer(256, dropout 0.25, relu)
- Output Dense Layer (32 batches, 10 epochs, early stop , optimizer: SGD)
- Total number of top words: 5000
- Max length of each review: 250

3.1 Choosing Max Length (number of words) for each review

We can see the pattern in this experiment: With the max length of each review increases, the classification also increases. Because we want the highest classification rate and using max length of 250 will also finish in acceptable time span, I choose 250 as the max length of each review.

Max Length	50	100	150	200	250
Classification rate	0.8104	0.8337	0.8375	0.8363	**0.8485**

3.2 Choosing Total number of top words

The result shows when the total number of top words increases from 500 to 8000, the classification start to increase first, then it get to a steady rate around 0.835 - 0.845. Because of this pattern, I choose 5000 as the total number of words in the final model because it both has **high classification rate** in considerable **fast** in computing process.

Total Number	500	1000	2500	3000	4000	5000	6000	7000	8000
Classification rate	0.7901	0.8052	0.8384	0.8403	0.8371	**0.8436**	0.8192	0.8428	0.8389

3.3 Choosing types of RNN model

According to the result, the type of LSTM RNN model has the highest classification rate.

TYPE	Simple RNN(16)	LSTM(32)	GRU(32)
Classification rate	0.8446	**0.8538**	0.8461

3.4 Choosing Number of Nodes in RNN layer

- RNN type: LSTM type

As we can see in the result, the pattern is the classification rate begins to increase then get to nodes of 32 and decreases. The reason for this is **over-fitting** in around 10 epochs. Even though I set the call back function of Early Stopping with patience of 2, when the model get to around 8 - 9 epochs, the experiments shows a decrease in classification rate appears in every model listed here.

Number of Nodes	4	8	16	32	64	128
Classification rate	0.8459	0.8608	**0.8663**	0.8538	0.8595	0.8446

3.5 Choosing Optimizer Function

- RNN type: LSTM type

As we can see in this experiment, Adadelta optimizer have the highest classification rate. The reason for this high performance is because it's and adaptive learning rate method, which is suitable for here when over-fitting happens in almost every experiment listed here.

TYPE	SGD	RMSPROP	Addgrad	Adadelta	Adam	Adamax
Classification rate	0.7513	0.8486	0.8531	**0.8745**	0.8513	0.8564