UDACITY

‹ Return to Classroom

DISCUSS ON STUDENT HUB

Sentiment Analysis with Neural Networks

REVIEW

HISTORY

Meets Specifications

Great job, you are ready to go! 💯 Clearly, you have acquired all the important concepts from this project. Wish you all the best for the upcoming projects! 🤞
Tip: If you would like to know some practical tips about how to improve your RNN, you may find this and this tutorial useful.

Importing Twits

Print the number of twits in the dataset.

Well done! You get the correct number of tweets from the given data set, `1548010`. 👏

Preprocessing the Data

The function `preprocess` correctly lowercases, removes URLs, removes ticker symbols, removes punctuation, tokenizes, and removes any single character tokens.

Good job, the patterns are correct. Also, you correctly remove the single character tokens.💯

Preprocess all the twits into the `tokenized` variable.

Well done! You successfully preprocess the message to get the required tokens with `preprocess` function💯

Create a bag of words using the tokenized data.

Great work creating a bag of words from the tokens with Counter!👍

Remove most common and rare words by defining the following variables: `freqs`, `low_cotoff`, `high_cutoff`, `K_most_common`.

Perfect! You have successfully removed the most common and rare words with the specified variables `freqs`, `low_cotoff`, `high_cutoff`, and `K_most_common`
Tip: There is not an exact number for low and high-frequency cut-offs, however, there is a correct optimal range. The low-frequency cut-off should be ranged from 0.000002 to 0.000007 (inclusive) and high-frequency from 5 to 20 (inclusive). If the number is too big, we lose lots of important words that we can use in our data.

Defining the variables : 'vacab', 'id2vocab' and 'filtered' correctly.

Well done, you have correctly defined the variables `vacab`, `id2vocab` and `filtered` 👏

Neural Network

The init function correctly initializes the following parameters: `self.vocab_size`, `self.embed_size`, `self.lstm_size`, `self.lstm_layers`, `self.dropout`, `self.embedding`, `self.lstm`, and `self.fc`.

Perfect! You have correctly initialized the required parameters. 🎉

The 'init_hidden' function generates a hidden state

Good job! The `init_hidden` function correctly initializes the hidden state of the LSTM layer. 💯

The 'forward' function performs a forward pass of the model the parameter input using the hidden state.

Awesome! You have successfully implemented the forward function. 👏

Training

Correctly split the data into `train_features`, `valid_features`, `train_labels`, and `valid_labels`.

Fantastic, you correctly split the data into 80% training and 20% validation datasets. 🙌

Tip: Usually the training set is something between 0.8 or 0.9 percentage of data

Tip: Usually the training set is something between 0.8 or 0.9 percentage of data.

Train your model with dropout and clip the gradient. Print out the training progress with the loss and accuracy.

Awesome! The model is successfully trained and the required accuracy is displayed.

Making Predictions

The `predict` function correctly prints out the prediction vector from the trained model.

Excellent! You have correctly calculated the sentiment score by using the trained model.👏

```
tensor([[ 0.0007,  0.0364,  0.0120,  0.6930,  0.2579]])
```

Answer what the prediction of the model is and the uncertainty of the prediction.

Great answer💯Your model is giving a positive predicted sentiment for this tweet

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review

START