UDACITY

PROJECT

Translation From One Language to Another Language

A part of the Deep Learning Nanodegree Foundation Program

PROJECT REVIEW

CODE REVIEW

NOTES

SHARE YOUR ACCOMPLISHMENT!

Requires Changes

1 SPECIFICATION REQUIRES CHANGES

Congratulations! This is an outstanding submission. There is a minor issue in *Sentence to Sequence* method. Everything else is correct. I have added few points in hyperparameter tuning part. Try experimenting them and resubmit your work. Your work is very close to an acceptable submission.

Keep up the great work!
Keep Learning! Deep Learning!

Required Files and Tests

The project submission contains the project notebook, called "dlnd_language_translation.ipynb".

You have added some of the folders that are not required for reviewing the project. It's recommended to only send the `language-translation` folder.

All the unit tests in project have passed.

Great Job! All the unit tests are running flawlessly. Unit tests help in validating our code. We should always use unit tests to validate the code before deployment. Although it's possible that there can be still some issue even after passing the unit tests. You can read this article-> https://www.toptal.com/qa/how-to-write-testable-code-and-why-it-matters

Preprocessing

The function `text_to_ids` is implemented correctly.

The `text_to_ids()` method is correctly preprocessing the text to numbers.

Neural Network

The function `model_inputs` is implemented correctly.

The placeholders are correctly declared in model_inputs() method. Their datatype, rank and name are also correct.

The function `process_decoding_input` is implemented correctly.

The `process_decoding_input()` is correctly using TensorFlow to remove the last word id from each batch in target_data and concat the GO ID to the beginning of each batch.

The function `encoding_layer` is implemented correctly.

The encoder network helps to map the input sequence to an encoded representation of the sequence. Thumbs up👍 for using the Dropout. Dropout helps in reducing the overfitting. To further read about Encoder-Decoder models in RNN, this quora discussion can be helpful. https://goo.gl/XvrybG

The function `decoding_layer_train` is implemented correctly.

The training logits are correctly created. It would be better if you could have used Dropout in the code.

The function `decoding_layer_infer` is implemented correctly.

The `decoding_layer_infer()` is accurately creating the inference logits. 👌

The function `decoding_layer` is implemented correctly.

This is correctly creating the RNN layers.

Suggestion:

There is no need of using 2 variable scopes. It can be simply implemented like this:

```
with tf.variable_scope("decoding") as decoding_scope:
       output_fn = lambda x: tf.contrib.layers.fully_connected(x,
                                                               vocab_size,
                                                               None,
                                                               scope=decoding_scope)

   # Train Logits
         train_logits = decoding_layer_train(encoder_state,
                                             dec_cell,
                                             dec_embed_input,
                                             sequence_length,
                                             decoding_scope,
                                             output_fn,
                                             keep_prob)
```

The function `seq2seq_model` is implemented correctly.

All the formulas in `seq2seq_model()` method are precisely coded. 👏

This discussion on quora can help you further: https://www.quora.com/What-is-conditioning-in-seq2seq-learning

Neural Network Training

The parameters are set to reasonable numbers.

The hyperparameters are acceptable although you can get similar results with a lower number of epochs. I encourage experimenting with epochs and increasing the embeddings size. Overall, it's good. 👏

The project should end with a validation and test accuracy that is at least 90.00%

Awesome! The accuracy is over 90%, this is really appreciable. ⭐

Language Translation

The function `sentence_to_seq` is implemented correctly.

There is a minor issue. You have missed converting the sentences to lower case. It can be done using lower() method.

> ### Sentence to Sequence
>
> To feed a sentence into the model for translation, you first need to preprocess it. Implement the function sentence_to_seq() to preprocess new sentences.
>
> - Convert the sentence to lowercase
> - Convert words into ids using vocab_to_int
> - Convert words not in the vocabulary, to the <UNK> word id

The project gets majority of the translation correctly. The translation doesn't have to be perfect.

The majority of translation is correct. ⭐

RESUBMIT

DOWNLOAD PROJECT

Best practices for your project resubmission

Ben shares 5 helpful tips to get you through revising and resubmitting your project.

Watch Video (3:01)