

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!  

Meets Specifications

Congratulations!

You have successfully completed this project!

I can tell that you spent a considerable time on this project, and you should be proud of the great outcome!

Required Files and Tests

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

All the unit tests in project have passed.

Good job passing all the unit tests!

This is good practice for Test Driven Development, where you write your tests out before you write the code, to make sure that your code behaves as you intend once you've written it! this is especially applicable in difficult programming exercises like this one, where a small syntax or mathematical error would be hard to find.

Preprocessing

The function `text_to_ids` is implemented correctly.Good job! You correctly used the `source_vocab_to_int` and `target_vocab_to_int` dictionaries!

Neural Network

The function `model_inputs` is implemented correctly.

Awesome

Good job initializing the `input_` and `targets` to integers, and `learning_rate` and `keep_prob` to floats!

Suggestion

The word `input` is a [reserved function in python](#) so I suggest you rename this variable to something like `input_data`.

The function `process_decoding_input` is implemented correctly.

The function `encoding_layer` is implemented correctly.

Suggestion

You could also use dropout here if you would like

The function `decoding_layer_train` is implemented correctly.

Suggestion

You could also use dropout here if you would like

The function `decoding_layer_infer` is implemented correctly.

The function `decoding_layer` is implemented correctly.

Suggestion

Instead of opening a new `with` clause, you can call `decoding_scope.reuse_variables()` inside the first `with` clause:

```
decoding_scope.reuse_variables()
infer_logits = decoding_layer_infer(encoder_state, dec_cell, dec_embeddings,
                                    target_vocab_to_int['<GO>'],
                                    target_vocab_to_int['<EOS>'], sequence_length - 1, vocab_size,
                                    decoding_scope, output_fn, keep_prob)
```

The function `seq2seq_model` is implemented correctly.

Neural Network Training

The parameters are set to reasonable numbers.

- Great job overall! I just want to draw your attention to the embedding size, which you currently have as 256. Keep in mind that the vocabulary size is 227, which indicates that using anything more than that is an overkill as you are trying to represent more concepts than you have words for. Commonly, we want the opposite: embedding many words into a few concepts. A good number of embedding size for this vocabulary is 128, as it is a power of two (speeds computation) and performs well for this translation task.

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

Wonderful validation loss! It's definitely higher than 90%

Language Translation

The function `sentence_to_seq` is implemented correctly.

Good job transforming the sentences into lower case and utilizing the `get` function in dictionaries!

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

The translation gets most of the concepts correctly. You could perhaps try a more challenging sentence like "he saw an old yellow truck". It seems to be a common one 😊

[📄 DOWNLOAD PROJECT](#)

[RETURN TO PATH](#)

Rate this review

[Student FAQ](#)