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!

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 “dlnd_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.

Suggestion

You can simplify this a bit by using list comprehension:

```
source_id_text = [
    [source_vocab_to_int[word] for word in sentence.split()] for sentence in source_text.split('\n')]
target_id_text = [
    [target_vocab_to_int[word] for word in sentence.split()]] for sentence + ' <EOS>' in target_text.split('\n')]
```

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!

The function `process_decoding_input` is implemented correctly.

The function `encoding_layer` is implemented correctly.

Awesome

Using dropout here will enable your network to learn the importance of each input and not give too much attention to a specific one! This might mean the network will need more time to learn, but eventually will perform better.

The function `decoding_layer_train` is implemented correctly.

The function `decoding_layer_infer` is implemented correctly.

The function `decoding_layer` is implemented correctly.

Dropout should be used only in training layers, not in inference. In the training layers, you don't want to be too dependent on any set of inputs, so you use dropout. However, in inference, when you have nodes representing concepts in language, then it can be quite detrimental if you drop any specific concept. Furthermore, as you pointed out, the project is setup to use 1.0 keep_prob, so there is no dropout during validation anyway.

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.

Suggestion

The network for this project doesn't actually need that many epochs to reach a good accuracy level. Good accuracies can be reached at 5-10 epochs.

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

Awesome

Great job! This is one of the highest I've seen for this project yet!

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.

Perfect! Good job inspecting various other sentences as well!

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review

Student FAQ