UDACITY

PROJECT

Translation From One Language to Another Language

A part of the Deep Learning Nanodegree Program

PROJECT REVIEW

CODE REVIEW

NOTES

SHARE YOUR ACCOMPLISHMENT!

Meets Specifications

A solid submission here, your code is good, you just need a bit of tuning and you will have a great translation model here. 👍

You have grasped the concepts of LSTM and RNNs really well. Great job!
You have successfully implemented a State of the Art model, now to expand your understanding please do go through the following links:
1.Machine Translation Tutorial
2.Attention and Augmented Recurrent Neural Networks
3.Word Embeddings

Also check out Bidirectional LSTM as they seem to have good performance. Please do make the suggested changes to start getting better results.

Its always a great practise to refer to the source research paper.
Check out this paper to get an idea about the hyper parameters used in larger corpora.

All the best in your deep learning journey! 👏

Required Files and Tests

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

All the unit tests in project have passed.

Preprocessing

The function `text_to_ids` is implemented correctly.

Neural Network

The function `model_inputs` is implemented correctly.

The function `process_decoding_input` is implemented correctly.

The function `encoding_layer` is implemented correctly.

The function `decoding_layer_train` is implemented correctly.

You can additionally try using dropouts here for better regularisation. 👍

The function `decoding_layer_infer` is implemented correctly.

The function `decoding_layer` is implemented correctly.

Suggestion

As per the project rubric you are supposed to use reuse_variables() function with the decoding_scope instead of redundantly making context managers. Instead of using

```
with tf.variable_scope("decode”, reuse=True) as decoding_scope:
        infer_logits...
```

use:

```
decoding_scope.reuse_variables()
infer_logits
```

Please do make sure that the reuse_variable() function and the consecutive code is indented to be at the same level as the previous context manager.Please do read more about Sharing Variables in Tf Also the output_fn can be created in the same context manager. :simple_smiley:

The function `seq2seq_model` is implemented correctly.

Neural Network Training

The parameters are set to reasonable numbers.

A very good set of parameters chosen here! Just a few suggestions here:

- epochs: Your number of epochs is high, usually the models tend to get good performance by 4-5 epochs.
- embedding size: Basically embedding sizes should be large enough that the model has capacity to learn. Here the size of our vocabulary is 227 words. The embedding size is what defines the amount of information that gets encoded and decoded from the sentences. So make sure this is around ~128-220 or so. When you encode information from 227 words to 256 dimensional vector you tend to add noise to the model. So please do remember this tip when setting embedding_size. 👍

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

Awesome job getting over 90% accuracy here! 👏

Language Translation

The function `sentence_to_seq` is implemented correctly.

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

Your model does get most of the word translations right, but can do a bit better here, try tuning your parameters to get perfect translations here:

```
Input
  Word Ids:      [83, 217, 100, 180, 222, 133, 40]
  English Words: ['he', 'saw', 'a', 'old', 'yellow', 'truck', '.']

Prediction
  Word Ids:      [303, 37, 343, 171, 179, 119, 230, 11, 1]
  French Words: ['il', 'a', 'vu', 'un', 'vieux', 'camion', 'jaune', '.', '<EOS>']
```

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review

Student FAQ