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

Great job with code implementation!
My congratulation - you have passed this project!
Good luck in studying!

p.s.
Just from curiosity you can check out this super-cool visualization of Neural Net: https://www.youtube.com/watch?v=3JQ3hYko51Y

Required Files and Tests

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

All the unit tests in project have passed.

Awesome!
All the code snippets and unit tests are running perfectly.

Preprocessing

The function `text_to_ids` is implemented correctly.

Great job!

Neural Network

The function `model_inputs` is implemented correctly.

Correct!

The function `process_decoding_input` is implemented correctly.

The function `encoding_layer` is implemented correctly.

Good job with use of dropout at this step!

Here is some good explanations of what is encoder layer:

- https://www.quora.com/What-is-an-Encoder-Decoder-in-Deep-Learning
- https://www.youtube.com/watch?v=FzS3tMl4Nsc

> Autoencoders are networks, which try to reconstruct their own input. **You construct the network so that it reduces the input size** by using one or more hidden layers, until it reaches a reasonably small hidden layer in the middle. As a result **your data has been compressed (encoded) into a few variables**. From this hidden representation the network tries to reconstruct (decode) the input again.

The function `decoding_layer_train` is implemented correctly.

Great job with dropout step!

The function `decoding_layer_infer` is implemented correctly.

Well done!

Here we shouldn't use a dropout. It is a prediction/inference step. So we don't need to use dropout at the prediction step.

The function `decoding_layer` is implemented correctly.

Awesome job defining a scope to share variables between training and inference.

The function `seq2seq_model` is implemented correctly.

Neural Network Training

The parameters are set to reasonable numbers.

Good choice hyperparameters!

You are using hyperparameters as a power of 2 and such numbers handled efficiently by GPU.

- **batch_size** - Large batch sizes increase training speed, but can degrade the quality of the model. You can read this very useful discussion about this: http://stats.stackexchange.com/questions/164876/tradeoff-batch-size-vs-number-of-iterations-to-train-a-neural-network
- **rnn_size** - basically it is a the number of units in your LSTM cell. This number should be large enough that the network can generalize and avoid high bias (underfitting) and on the other hand it shouldn't be too large and shouldn't create a high variance (overfitting) model. Here is good explanation of bias-variance trade-off: http://scott.fortmann-roe.com/docs/BiasVariance.html
- **num_layers** - You've made a great choice here! Our dataset is small here so we don't have reasons to use large number of layers.
- **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. So 256 is an ideal number!

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

Language Translation

The function `sentence_to_seq` is implemented correctly.

!Important

I will mark it as passed, cause your translation and accuracy is quite good.
Note that at this step you should convert the sentence to lowercase.

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

Yuo have quite good translation!

- *he saw a old yellow truck* ->
- *il a vu l'automobile verte* ->
- *He saw the green car* (according google translation)

DOWNLOAD PROJECT

RETURN TO PATH

Student FAQ