

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

2 SPECIFICATIONS REQUIRE CHANGES

Well Done !!! This is a great submission. You have to make a few changes and you are good to go. Google recently released a general sequence-to-sequence learning framework based on tensorflow. You can read about it in [this blog post](#). If you want an introductory tutorial for machine translation, you can refer to the [official tensorflow tutorial](#).

All the best for your next submission. Keep learning.

NOTE: Please do not submit the link to your github project the next time. You have got quite a heavy project folder out there. It takes a few minutes to download rather than seconds. Also, after downloading, we have to go fishing around for the folder that exactly contains the project. It would be better if you zip the required files and then upload them while submitting. That would make our life easier. I see that it was requested earlier also by one of the reviewers. I would earnestly request you to please consider this. Thank You.

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.

`text_to_ids` correctly transforms text into numeric IDs. Also, it correctly adds the EOS ID to the end of target sentences.

Neural Network

The function `model_inputs` is implemented correctly.

The placeholder tensors have been correctly defined. These are the building blocks of the neural nets in Tensorflow. Well Done !!!

The function `process_decoding_input` is implemented correctly.

Well Done !!! GO ID has been appended correctly to the beginning of each batch.

The function `encoding_layer` is implemented correctly.

Well Done !!! The encoding layer RNN has been correctly setup. You have used dropout in the RNN layer too.

The function `decoding_layer_train` is implemented correctly.

Good Job !!! The decoding layer has been prepared for training correctly. You could have used dropout too in the RNN cells.

The function `decoding_layer_infer` is implemented correctly.

Good Job !!!

The function `decoding_layer` is implemented correctly.

Nice Work !!! However, you have applied dropout to the RNN cell. This decoder cell is passed later to the functions `decoding_layer_train` and `decoding_layer_infer`. Using dropout during inference leads to randomness in predictions. Although it is not a problem here as we use `keep_probability` as 1 during inference in the `train` function, it would be better to remove the dropout from the LSTM cells in this function. If you want to apply dropout to the decoder during training, please do so in the function `decoding_layer_train`. You could apply dropout to the LSTM cells or after the stacked LSTM layer. Please do not apply dropout twice there as that tends to slow down the training significantly.

The function `seq2seq_model` is implemented correctly.

Well Done !!! You have correctly pieced together all the required components fo the sequence-to-sequence model.

Neural Network Training

The parameters are set to reasonable numbers.

Although the result of your model is quite good, the hyperparameters are not optimal. The practical application of a model would be determined by taking into account many factors - accuracy, generalization capability and time to train and predict. If a model performs in terms of accuracy and takes a lot of time to train, it might not be a sustainable model in the industry setting. The current model suffers from the same kind of problem. It takes a lot of time to train the network to give reasonably good results. Below are the changes that I would like to suggest and you should be able to achieve similar results within 5- 7 epochs:

- You can increase the batch size to 256. It would speed up the training a bit without losing out too much on performance.
- You should increase the RNN size to 256 in order to give the model enough flexibility to model the intricacies of the language.
- You should try embedding sizes in the range 50 - 128. Embedding sizes of 64 and 128 perform reasonably well. Embedding sizes greater than that do not add any value to the network.
- The learning rate is fine. However, given the above changes, you might have to lower it a bit. You could try something in the range 0.001 to 0.003.
- Dropout applied is fine.

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

The validation accuracy is above 90%.

Language Translation

The function `sentence_to_seq` is implemented correctly.

Good Job !!!

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

The translation from the model is pretty good. Awesome !!!

 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)

[RETURN TO PATH](#)

[Rate this review](#)

[Student FAQ](#)