PROJECT

Translation From One Language to Another Language

A part of the Deep Learning Nanodegree Foundation Program

PROJECT REVIEW | CODE REVIEW | NOTES

Requires Changes

SHARE YOUR ACCOMPLISHMENT

1 SPECIFICATION REQUIRES CHANGES

Well Done !!! This is a great first submission. It is clear that you have invested a lot of time and effort into this project and have clearly understood the foundations of a sequence to sequence model.

Google recently open-sourced a general sequence-to-sequence framework based on tensorflow. You can read about it in this blog post. You can refer to this paper for gaining an intuition about the hyperparameters by reading the Experiments section.

All the best for your next submission. Keep learning !!!

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.**

Go ID has been correctly added to the beginning of each batch. Nice Work !!!

✓ **The function `encoding_layer` is implemented correctly.**

Well Done !!! The encoding layer RNN has been correctly setup. Also, you have used dropout in the RNN layer.

✓ **The function `decoding_layer_train` is implemented correctly.**

Good Job !!! The decoding layer has been prepared for training correctly. Since you have already applied dropout to the cells in the function, `decoding_layer`, you do not have to use `keep_prob` here. However, I have explained why you should not apply dropout in the function, `decoding_layer`. Based on that, if you wish to apply dropout to the decoder during training, you will have to apply `DropoutWrapper` to the passed.

✓ **The function `decoding_layer_infer` is implemented correctly.**

Good Job. It is correct that you have not applied dropout to the inference layer. Applying dropout during inference leads to random predictions. So, dropout should never be applied to the inference layer.

✓ **The function `decoding_layer` is implemented correctly.**

Good Job !!! The decoding layer has been setup correctly. However, I would like to make a few suggestions:

- You have applied dropout to the RNN cells. These cells are used both during decoder training and inference. However, applying dropout during inference leads to random predictions as stated in the rubric above. But, this is not a problem here as we pass `keep_prob` as 1 during inference later. However, it will be still a good idea to segregate the things. If you want to apply dropout to the decoder RNN cells or to the outputs of the decoder, please do so in the function `decoding_layer_train` and remove the dropout from cells in this function.
- Instead of defining two context scopes, you can define a single context scope and use `reuse_variables()` method on the context scope before using for inference.

✓ **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.**

The hyperparameters have not been chosen properly. Please see below:

- Batch size of 256 is fine. However, you could use a batch size of 512 without any noticeable performance drop.
- RNN Size is also small. You should use a larger RNN size in order to give the network enough power to learn the intricacies of the language. RNN size of 256 would have been a better fit.
- The number of layers is optimal as we are working with a small corpus.
- The embedding size is quite high. We only have a small vocabulary. The vocabulary size is only 227 words. You are using an embedding size of 300 which is higher than the actual vocabulary. It does not sound logical to use an embedding size that is higher than the actual vocabulary as the purpose of embedding is to reduce dimensions while capturing most of the information. We want to choose an embedding size such that it will be able to capture the semantics of the words. An embedding size in the range 50 - 128 works quite well for this problem. I would recommend going for the higher side of this range.
- Both learning rate and dropout applied are fine.

With the above settings, you should be able to get to a fairly good accuracy within 5 - 6 epochs.

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

The validation accuracy is well 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 is pretty good. Well Done !!!

RESUBMIT PROJECT

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

Student FAQ