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!

Requires Changes

6 SPECIFICATIONS REQUIRE 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.

All placeholders are correctly defined except for the placeholders `target_seq_length` and `source_seq_length`. These must be Rank 1 tensors . However, you have defined these as a scalar. This is the reason you are getting error later while training. Please define these as something like below:

```
target_seq_len = tf.placeholder(tf.int32, [None], name='target_sequence_length')
```

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. However, you should have used dropout in the encoder cells.

The function `decoding_layer_train` is implemented correctly.

Good Job !!! The decoding layer has been prepared for training correctly. **I would strongly recommend adding dropout to the decoder cells during training.**

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. However, you should be using the same encoder state passed in this function to the function, `tf.contrib.seq2seq.BasicDecoder` rather than re-initalizing the state and also use the output layer. Please refer to the API for details.

The function `decoding_layer` is implemented correctly.

Well Done !!! However, instead of defining multiple variable contexts, you could have used `reuse_variables()` on the first context defined.

The function `seq2seq_model` is implemented correctly.

Well Done !!! You have correctly pieced together all the required components for 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 128 is too small. It takes a lot of time to train the network. In order to speed up the training significantly, you could use a batch size of 256 or 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. Please try RNN size in the range 128 - 256.
- We do not need a deeper network here as we are working with a small corpus. 2 layers are fine.
- The embedding size is too low. We have a vocabulary of 227 words. 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 to the higher side of this range.
- Training for 60 epochs is not ideal. You should get a better trained model within 10 epochs with a good set of hyperparameters.
- You are using no dropout. However, as I have suggested earlier, you should use dropout in the encoder cells and decoder cells (during training only). Once, you are using dropout in the model, you can try with 30% - 50% dropout.

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

The validation accuracy must be at least 90%.

Language Translation

The function `sentence_to_seq` is implemented correctly.

Please convert to lowercase before converting to numeric IDs.

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

The translation has to be at least contextual. Please try to achieve something like below:

```
INFO:tensorflow:Restoring parameters from checkpoints/dev
Input
  Word Ids:      [163, 190, 129, 122, 113, 74, 6]
  English Words: ['he', 'saw', 'a', 'old', 'yellow', 'truck', '.']
```

```
Prediction
  Word Ids:      [178, 47, 222, 85, 175, 12, 223, 92, 1]
  French Words: il a vu le vieux camion jaune . <EOS>
```

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