UDACITY

Logout

PROJECT

Translation From One Language to Another Language

A part of the Deep Learning Nanodegree Foundation Program

PROJECT REVIEW | CODE REVIEW | NOTES

Requires Changes

3 SPECIFICATIONS REQUIRE CHANGES

SHARE YOUR ACCOMPLISHMENT

You did a great job with the project.

Just some further fine-tuning of your hyperparameters, and you will be good to go! :)

I hope the review helped you. If you feel there's something more that you would have preferred from this review please leave a comment. That would immensely help me to improve feedback for any future reviews I conduct including for further projects. Would appreciate your input too. Thanks!

Good luck! :)

Required Files and Tests

✓ **The project submission contains the project notebook, called “dlnd_language_translation.ipynb”.**

✓ **All the unit tests in project have passed.**

All tests passed!

Preprocessing

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

Great work! Concise and simple implementation.

Neural Network

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

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

Good work!

Now instead of `strided_slice()` try to see if you can use something like the following -

```
ending = target_data[:,:-1]
```

Do you think the above does the same and helps?

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

Nice work!

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

Good job! Although, since you already define scope in `decoding_layer()` you need not define it here again. Plus, you are already passing the scope via the `decoding_scope` variable/parameter.

Rate this review

☆☆☆☆☆

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

Nicely done. As I pointed above, you need not specify the scope here again.

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

Excellent!

Some suggestions -

- You can define your MultiRNNCell (And related) within the `decoding_scope` itself instead of outside of it.
- You can use `decode_scope.reuse_variables()` instead of utilizing `with tf.variable_scope("decoding", reuse=True) as decoding_scope:` . Makes it slightly more convenient as per me :)
- You can consider initializing your weights and biases for your FC layer as well. Proper initialization of the weights can help with the model convergence too. If you notice your loss and accuracies during training oscillate quite a bit which indicates your model has some trouble converging properly as it's overshooting the cost function minimum. So this might help with that.

✓ **The function `seq2seq_model` is implemented correctly.**

Nicely done! 👏

Here are a couple of useful resources which can further help you build an intuitive knowledge on the matter https://indico.io/blog/sequence-modeling-neuralnets-part1/ and https://wp.wwu.edu/blogwolf/2017/02/20/seq2seq/
Do check them out! :)

Neural Network Training

↻ **The parameters are set to reasonable numbers.**

You did a good job tuning the hyperparameters. But there is some further fine-tuning required -

- You selected a good number of epochs for your current model. We should tend to select the number of epochs such that you get a low training loss which reaches kind of a steady-state (not much change in value beyond a point). With a bit more fine-tuning of the following hyperparameters you can lower the epochs.
- You selected a good batch size. Smaller batch sizes take too long to train. Larger batch sizes speed up the training but can degrade the quality of the model. Here is a useful resource to understand this better - http://stats.stackexchange.com/questions/164876/tradeoff-batch-size-vs-number-of-iterations-to-train-a-neural-network
- *Important:* I would recommend a higher RNN size since currently your model isn't generalizing well. That's why it takes much longer (more epochs) to reach a reasonable accuracy value. Also, what do you think is better - more RNN size with relatively fewer layers, or smaller RNN size with more layers?
- Currently, if you notice, your validation accuracy matches your training accuracy quite well throughout training and evaluating. At times, the validation accuracy is slightly higher than your training accuracy as well. Usually, that indicates that your model capacity is not high enough (http://cs231n.github.io/neural-networks-3/#accuracy) and you could try making the model larger by increasing some parameters (higher RNN size/layers, embedding sizes and such) Although, it's also dependent on the data (including quality) as well. But try it out.
- *Important:* Continuing the above point, I would recommend higher values for the encoding and decoding embedding sizes. The recommended values are usually based on the vocab size, which is somewhere between 200 to 300. So try a value in that range and see how it improves your model.
- Select a learning rate so that your model converges well, and there aren't many oscillations/spikes in your training loss as the training progresses. Those spikes (increase and decrease in value) happen mostly because the model overshoots the cost function minimum and can't converge properly. Currently, your training has quite a bit of such spikes. So you can try to lower your learning rate as well to see if you can minimize that.

Previous reviewer might have missed out on some of these points. Sorry for the inconvenenience.

Good luck!

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

When I ran your model you almost reached 90%, but your model is kind of underfitting currently. The above changes should improve overall training progress :)

Language Translation

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

Rate this review
☆☆☆☆☆

Well done!

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

After the above changes/fine-tuning, try to test the output on few sentences from the data directly, and on some of your own. That should help you make a better comparison of how your model is performing really.

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

Rate this review

★★★★★