

PROJECT

Translation From One Language to Another Language
A part of the Deep Learning Nanodegree Foundation Program

PROJECT REVIEW

CODE REVIEW

NOTES

Meets Specifications

SHARE YOUR ACCOMPLISHMENT



Excellent work with the project! Congratulations! :)

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 with the final project!

Required Files and Tests

✓ The project submission contains the project notebook, called "d1nd_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! Very concise!

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 -

```
push_back = target_data[:,t+1:]
```

Do you think the above does the same and helps?

✓ The function `encoding_layer` is implemented correctly.

Nicely done! Especially for adding dropout!

Here's a question for you -

Do you think dropout should be added to the Basic LSTM cell, like you have, or to the stack of LSTM layers instead?

✓ The function `decoding_layer_train` is implemented correctly.

Well done!

✓ The function `decoding_layer_infer` is implemented correctly.

Great job!

✓ 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 consider initializing your weights and biases for your FC layer as well. Proper initialization of the weights can help with the model convergence too.

✓ 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.wvu.edu/blogwoolf/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. Some suggestions -

- You selected a good number of epochs. 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).
- 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>
- You selected a good value for RNN size. What do you think is better - more RNN size with relatively fewer layers, or smaller RNN size with more layers? Do you think you need such a relatively large model for this dataset?
- You can consider lowering your embedding sizes, since for this dataset and vocab size you don't require relatively large embedding sizes.
- 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 some spikes. So you can try to fine-tune your learning rate as well to see if you can minimize that, although it's not required.

Good work!

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

Excellent work! You got a ~96% validation accuracy when I run your model. Really awesome!

Language Translation

✓ The function `sentence_to_seq` is implemented correctly.

Nice work!

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

Well done! 🍷

I get the following results when I run your model for a sample sentence. Quite correct

```
Input
Word Ids:      [15, 62, 29, 28, 34, 72, 219, 140, 149, 62, 222, 121, 125, 97, 227]
English Words:  ['california', 'is', 'never', 'cold', 'during', 'february', ',', 'but', 'it', 'is', 'sometimes', 'freezing', 'in', 'june', ',']

Prediction
Word Ids:      [327, 163, 11, 158, 158, 47, 237, 118, 352, 159, 191, 294, 187, 25, 273, 237, 170]
French Words:  ['californie', 'ne', 'fait', 'jamais', 'jamais', 'froid', 'en', 'fevrier', ',', 'mais', 'il', 'est', 'parfois', 'le', 'gel', 'en', 'juin']
```

Try it out on the larger dataset now if you can!

Also, one very simple question I would like you to think about - if you wanted to translate from French To English, would you have to retrain the entire model with french dataset? Or you think you could somehow reverse the prediction only?

📄 DOWNLOAD PROJECT

RETURN TO PATH

Rate this review



Rate this review



Rate this review



Rate this review



Rate this review

