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

3 SPECIFICATIONS REQUIRE CHANGES

A solid submission here, your code is good, you just need a bit of tuning and you will have a great translation model here. 👍

You have grasped the concepts of LSTM and RNNs really well. Great job!
You have successfully implemented a State of the Art model, now to expand your understanding please do go through the following links:
1.Machine Translation Tutorial
2.Attention and Augmented Recurrent Neural Networks
3.Word Embeddings

Also check out Bidirectional LSTM as they seem to have good performance. Please do make the suggested changes to start getting better results.

Its always a great practise to refer to the source research paper.
Check out this paper to get an idea about the hyper parameters used in larger corpora.

All the best in your deep learning journey! 👋

### 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.**

### Neural Network

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

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

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

Good job using dropouts here! 👍

The function `decoding_layer_train` is implemented correctly.

The function `decoding_layer_infer` is implemented correctly.

The function `decoding_layer` is implemented correctly.

## Suggestion

As per the project rubric you are supposed to use reuse_variables() function with the decoding_scope instead of redundantly making context managers. Instead of using

```
with tf.variable_scope("decode", reuse=True) as decoding_scope: #Naming the context manager is essential
        inference_decoder_output...
```

use:

```
    decoding_scope.reuse_variables() #You are using the name here
    inference_decoder_output
```

Please do make sure that the reuse_variable() function and the consecutive code is indented to be at the same level as the previous context manager.Please do read more about Sharing Variables in Tf Also the output layer can be created in the same context manager.

The function `seq2seq_model` is implemented correctly.

### Neural Network Training

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

You need to tune your hyper parameters as your model accuracy is fluctuating most probably due to underfitting.
Here are a few suggestions:

- epochs: Your number of epochs is high, usually the models tend to get good performance by 4-10 epochs.
- batch_size: The batch size is good here, larger batch size tends to make the training process a bit fast but may deteriorate performance. So a good batch_size is required, also the batch_size should be compatible to the GPU memory in hand.
- rnn_size :This is the number of units in the LSTM cell.The value of it is it up to you, too high a value may lead to overfitting or a very low value may yield extremely poor results. Your selection here is a bit too small. Try 128,256
- hidden_layers: This is again in your hands to choose, more rnn_size and small hidden_layers or low rnn_size and more hidden_layers. More number of layers tend to increase the training time. 2 is good
- embedding size: Basically embedding sizes should be large enough that the model has capacity to learn. Here the size of our vocabulary is 227 words. The embedding size is what defines the amount of information that gets encoded and decoded from the sentences. So make sure this is around ~128-220 or so.

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

```
Epoch 59 Batch 1040/1077 - Train Accuracy: 0.6953, Validation Accuracy: 0.6864, Loss: 0.4185 Epoch 59 Batch 1060/1077 - Train Accuracy: 0.6672, Validation Ac
```

You need to get both train and validation accuracy over 90% to pass the rubric.
This project is one of the toughest in the course, the model is state of the art and a version of it is being used everyday in google translate. Also please note these projects will be part of your portfolio projects from this course, so make sure you get the best of out of them. You are on the right track here, hyper parameter tuning is crucial part, with a bit of tuning you will get an awesome model! All the best! 👋

### Language Translation

The function `sentence_to_seq` is implemented correctly.

A nice implementation.

However here is a simpler pythonic way to do:

```
[vocab_to_int.get(word, vocab_to_int['<UNK>']) for word in sentence.lower().split()]
```

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

Your model translations are a bit off, you need to tune your hyper parameters a bit to make sure that your model translations are better, also please do go through the suggestions to get better understanding.

Try getting translations like:

```
Input
  Word Ids:      [83, 217, 100, 180, 222, 133, 40]
  English Words: ['he', 'saw', 'a', 'old', 'yellow', 'truck', '.']

Prediction
  Word Ids:      [303, 37, 343, 171, 179, 119, 230, 11, 1]
  French Words: ['il', 'a', 'vu', 'un', 'vieux', 'camion', 'jaune', '.', '<EOS>']
# Your word ids may be different
```

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

Student FAQ