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

2 SPECIFICATIONS REQUIRE CHANGES

A good project with a coding issue in the `sentence_to_seq` function - details and suggestion are in feedback notes.

Required Files and Tests

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

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

The student has asked for assistance in resolving issues with the `sentence_to_seq` function.

Preprocessing

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

Well done, neural networks have no understanding of the "meaning" of words but need to reference words through identifiers.

Note: This concept is also found in the different ways numbers can be used:

- A cardinal number tells "how many"

- An ordinal number indicates the order of things in a set
- A nominal number names something, such as player number in team

Rate this review

Neural Network

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

Well done, placeholder are correctly instantiated.

Note: Placeholders are used to hold the input values to be used in a TensorFlow session. Placeholders can be viewed as in the same way as function parameters. Variables are used to hold values which can be updated in a TensorFlow session, in particular trainable values such as biases and weights.

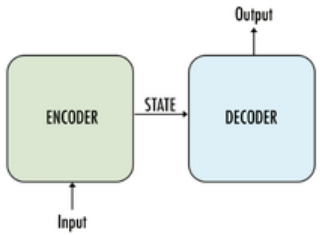
✓ **The function `process_decoding_input` is implemented correctly.**

Well done:

- taken a slice of the target data omitting the final element in the batch
- creating the target data by adding the GO id to the beginning of each batch

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

Well done. The final state of the encoding layer is used as the input of the decoding layer:



Note: Tensorflow provides 2 RNN components:

- tf.nn.rnn
- tf.nn.dynamic_rnn

`tf.nn.rnn` is used for applications where there is a fixed length unrolled RNN. This component cannot accept more than the specified number of steps and is also quite slow (RNN vs Dynamic RNN)

The `tf.nn.dynamic_rnn` uses a `tf.While` loop to dynamically construct the graph at run time. The graph creation is faster than the static rnn implementation without performance penalty.

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

Well done.

References on training seq2seq networks:

- Presentation on RNN APIs applied to seq2seq modelling
- Post introducing seq2seq, in particular the implementation of backpropogation training

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

Well done.

The difference between training and inferrence is detailed in this post.

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

`decoding_scope.reuse_variables()` can be used instead of the additional `with tf.variable_scope(...)` block (see documentation).

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

Well done.

Reference: This paper describes in detail the principles of sequence to sequence models.

Neural Network Training

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

epochs
Well done, the training loss and validation accuracy are both plateauing by the end of training without indication of overfitting. If the validation accuracy had plateaued earlier in training with the training loss continuing to decline - this would indicated overfitting to the training dataset.

rnn size
The length of the rnn_size is good, Karpathy recommends longer rnn_size (based on data and computing resources), however the benefits of excessive lengths can be limited (see post)

embedding dimension
Well done, Colah recommends between 200 to 500 dimension for complex text. This is corpus is not as complex so a lower value of embed dim is okay.

batch size
256 is a good batch size for efficient training on GPU (benefiting from parallel processing).

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

Well done. On completion of training the network validation accuracy is 93%.

Language Translation

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

You are correct the project is unable to perform translation due to issues with the `sentence_to_seq` function. On changing this function the network translation was quite accurate.

This section of code:

```
    for k in range(0,len(words)):
        if (k in vocab_to_int):
            snippet.append(vocab_to_int[words(k)])
        else:
            snippet.append(vocab_to_int['<UNK>'])
```

should change to:

```
    for word in words:
        if word in vocab_to_int:
            snippet.append(vocab_to_int[word])
        else:
            snippet.append(vocab_to_int['<UNK>'])
```

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

With the above change the network will be able to perform translations.

RESUBMIT PROJECT

DOWNLOAD PROJECT

Learn the best practices for revising and resubmitting your project.

RETURN TO PATH

Student FAQ