UDACITY

PROJECT SPECIFICATION

Generate TV Scripts

All Required Files and Tests

CRITERIA	MEETS SPECIFICATIONS
All project files are included in the submission	The project submission contains the project notebook, called “dlnd_tv_script_generation.ipynb”.
All unit tests in the project have passed	All the unit tests in project have passed.

Pre-processing Data

CRITERIA	MEETS SPECIFICATIONS

CRITERIA	MEETS SPECIFICATIONS
The function `create_lookup_tables` is implemented	The function `create_lookup_tables` create two dictionaries: • Dictionary to go from the words to an id, we'll call vocab_to_int • Dictionary to go from the id to word, we'll call int_to_vocab The function `create_lookup_tables` return these dictionaries as a tuple (vocab_to_int, int_to_vocab).
A special token dictionary is created	The function `token_lookup` returns a dict that can correctly tokenizes the provided symbols.

Batching Data

CRITERIA	MEETS SPECIFICATIONS
Data is broken into sequences	The function `batch_data` breaks up word id's into the appropriate sequence lengths, such that only complete sequence lengths are constructed.
Data is created using	In the function `batch_data`, data is converted into Tensors and formatted with TensorDataset.

TensorDataset CRITERIA	MEETS SPECIFICATIONS
Data is batched correctly.	Finally, `batch_data` returns a DataLoader for the batched training data.

Build the RNN

CRITERIA	MEETS SPECIFICATIONS
An RNN class has been defined	The RNN class has complete `__init__`, `forward`, and `init_hidden` functions.
The RNN includes at least one LSTM (or GRU) and fully-connected layer.	The RNN must include an LSTM or GRU and at least one fully-connected layer. The LSTM/GRU should be correctly initialized, where relevant.

RNN Training

CRITERIA	MEETS SPECIFICATIONS

CRITERIA	MEETS SPECIFICATIONS
Reasonable hyperparameters are selected for training	• Enough epochs to get near a minimum in the training loss, no real upper limit on this. Just need to make sure the training loss is low and not improving much with more training. • Batch size is large enough to train efficiently, but small enough to fit the data in memory. No real "best" value here, depends on GPU memory usually. • Embedding dimension, significantly smaller than the size of the vocabulary, if you choose to use word embeddings • Hidden dimension (number of units in the hidden layers of the RNN) is large enough to fit the data well. Again, no real "best" value. • n_layers (number of layers in a GRU/LSTM) is between 1-3. • The sequence length (seq_length) here should be about the size of the length of sentences you want to look at before you generate the next word. • The learning rate shouldn't be too large because the training algorithm won't converge. But needs to be large enough that training doesn't take forever.
The model shows improvement during training	The printed loss should decrease during training. The loss should reach a value lower than 3.5.
Question about hyperparameter choices is answered.	There is a provided answer that justifies choices about model size, sequence length, and other parameters.

CRITERIA	MEETS SPECIFICATIONS

Generate TV Script

CRITERIA	MEETS SPECIFICATIONS
The generator code generates a script	The generated script can vary in length, and should look structurally similar to the TV script in the dataset. It doesn't have to be grammatically correct or make sense.

Suggestions to Make Your Project Stand Out!

- Use validation data to choose the best model
- Initialize your model weights, especially the weights of the embedded layer to encourage model convergence
- Use topk sampling to generate new words
- Check out the "Advanced projects" section in the project overview to take this work even further!