Richard J. Taylor, PMP
December 22, 2018

Text Generation with Recurrent Neural Networks

Domain Background

Science fiction is the oracle of technology. Isaac Asimov alone created visions of a future with tales of sentient machines and space travel is it any wonder so many children have grown to create thriving businesses as SpaceX and Boston Dynamics. Even the one that brought us the iPhone. Although I think they stole that naming idea from Asimov, Isaac, A. (2018). Wikipedia.

Artificial Intelligence (AI) has been a part of the genre since its inception. There is no better way to give tribute to the gods of technology than have their creation foretell of a universe of its making.

This project will create an application that the user provides a quick blurb with a main character name and scenario that will generate a 1,000 word-essay.

Problem Statement

Using Natural Language Processing (NLP) the application will create fictional work based on what its learned from the dataset, in this case stories of fiction. The purpose of this is not to create full works of fiction, but to illicit interest and motivation and with a little luck act as a sort of muse for both writers and scientists alike.

This project will use a type of Recurrent Neural Network (RNN), a Bidirectional Recurrent Neural Network. Utilizing the Long Short-Term Memory (LSTM) mechanism or more specifically the Gated Recurrent Unit (GRU), which is a simplified LSTM to help with the Vanishing Gradient Problem that most RNN's suffer.

Datasets and Inputs

The initial dataset is provided by Jannes Klaas via Kaggle and was originally created by Robin Sloan who collected it largely from the Pulp Magazine Archive, Klaas, J. (2018). SciFi Stories Text Corpus. It is a 150MB .txt file containing Science Fiction stories from Galaxy and IF Magazine, but also contains a lot of noise.

There are errors and advertisements mixed within the stories. The file has been processed so it is in the form of a massive text file without line breaks. The punctuation has been normalized and most of the numbers replaced with a "#". There is a lot of text, *137,246-words*, which works well with RNNs.

I would like to have better datasets. At some point I would like one for each genre of fiction.

Solution Statement

We will use a type of neural network referred to as a Recurrent Neural Network (RNN) like what is used in Robin Sloan's, Writing with the Machine, Sloan, R. (2016). Using memory, or a hidden state, a RNN models sequential interactions of inputs; such as a

sentence. The outputs may be a classification of how often words are used, or even the correlation of words used in conjunction with one-another. For each step the same parameters are used over for the different inputs, Natural Language Processing (NLP), Lin, C., Manning, C., Ng, A., and Socher, R. (2011).
As described in the paper Parsing Natural Scenes and Natural Language with Recursive Neural Networks, Lin, C. Manning, C. Ng, A. and Socher, R. (2011), a NLP will be used to parse sentences in a recursive manner. The sentences will be cut into segmented features and word indices using TensorFlow and Keras APIs. The RNN will map these as semantic structures, where higher scoring structures are merged into larger ones and assigned class labels.

Benchmark Model

Support Vector Machines (SVM) are regularly used as baselines for text categorization, Manning, C. and Wang, S (n.d.). A TensorFLow Support Vector Machine (SVM) model will be used as a benchmark for this project, TensorFlow, (2018).

Evaluation Metrics

While intuition will be the main evaluation metric; readable stories are easy to identify by human evaluation. I will also use the Bilingual Evaluation Understudy (BLUE) score as an evaluation metric.
The BLEU Score evaluates a generated sentence to a reference sentence. The range of matching is from 1.0, which is a perfect match, to 0.0 for complete mismatch. Some good reasons to use this as an evaluation metrics as it is quick and inexpensive to calculate, easy to understand, language independent and correlates highly with human evaluations, Brownlee, J. (2017).

Project Design

The project work-flow will be as follows:

- Data Pre-Processing
- Model Design
- Model Evaluation

Data Pre-Processing

First the data is compiled into a single string of characters. This will help when tokenizing the words and phrases. Then we will create lookup tables for the vocabulary which creates a tuple. After this the punctuation is tokenized by mapping. Finally, the data is processed and saved for further use.

Model Design

The data is batched into input and target data. The words are replaced with IDs and the batch sizes are defined and combined as a numbpy array.
Hyperparameters such as the number of epochs, batch size, layers, dimensions, learning rate, etc. are defined and set.

Using TensorFlow the data is graphed. TensorFlow provides a context manager graph (tf.Graph.as_default). This step is where the text attributes are calculated and the RNN is called into action and built. The RNN will calculate the probability of word use and then have a loss function set. It will further have a learning optimizer and any gradient issues addressed and resolved via optimizers.
The training is then set by defining the semantic structures. The structures are batched and graphed, then trained over a set number of epochs.

Model Evaluation

The BLEU Score is computed by counting matching n-grams in the candidate text compared to the reference text. A N-gram represents a sequence of words, where ‘n’ is the number of words in the ‘gram’. For instance:

1. Snickerdoodles are delicious | is a 3-gram
2. Cthulhu rises | is a 2-gram
3. He walked with intent | is a 4-gram

Probabilities are assigned to the n-grams, which helps the model predict which N-grams may be placed together. This also helps to make next word predictions and spelling error corrections.
Using the predictive properties of the N-gram model predicts the next word based on the value of N. If ‘N = 2’ the model will use the previous word to predict the next word. If the instance of ‘N = 3’ the model will use the previous two words to predict the next word, Brownlee, J. (2017).

References

Brownlee, J. (2017). A Gentle Introduction to Calculating the BLEU Score for Text in Python. Retrieved from https://machinelearningmastery.com/calculate-bleu-score-for-text-python

Britz, D. (2018). Artificial Intelligence, Deep Learning, and NLP. Retrieved from http://www.wildml.com

Isaac, A. (2018). Wikipedia: Wikipedia.com. Retrieved from https://en.wikipedia.org/wiki/Isaac_Asimov

Karpathy, A. (2015). The Unreasonable Effectiveness of Recurrent Neural Networks. Retrieved from http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Keras, (2018). Retrieved from https://keras.io/preprocessing/text/

Klaas, J. (2018). SciFi Stories Text Corpus. Retrieved from https://www.kaggle.com/jannesklaas/scifi-stories-text-corpus

Lin, C., Manning, Ng, A., and Socher, R. (2011). Parsing Natural Scenes and Natural Language with Recursive Neural Networks. Retrieved from https://nlp.stanford.edu/pubs/SocherLinNgManning_ICML2011.pdf

Manning, C. and Wang, S. (n.d.). Baselines and Bigrams: Simple, Good Sentiment and Topic Classification. Retrieved from https://nlp.stanford.edu/pubs/sidaw12_simple_sentiment.pdf

Sloan, R. (2016). Writing the machine. Retrieved from https://www.robinsloan.com/notes/writing-with-the-machine/

TensorFlow, (2018). Retrieved from www.tensorflow.org

TensorFLow, (2018). Class SVM. Retrieved from https://www.tensorflow.org/api_docs/python/tf/contrib/learn/SVM