RNN Report

Bryan Greener

2018-06-25

Implementation

This program uses a Long Short-Term Memory character model Recurrent Neural Network to generate sentences given a text file input. In this case, the network is trained on the book of Revelations from the King James Bible. The text input is read in 100 characters at a time and each set of 100 characters is converted into a matrix of one-hots. After training the network, the model is given an input of a sequence of characters and predicts the next character in this sequence. It then shifts the new character into the input matrix and predicts the next characters sequentially. Due to time constraints, this network uses a single LSTM layer of 256 hidden nodes, a sequence length of 100 characters, and a batch size of 32 sets of sequences of characters. This model is based on the Alice in Wonderland tutorial that many other students used as well, however it was rewritten and changed to fit an Object Oriented structure. This improves the ability for testing different model designs while instantiating the network class. The data preprocessing code also had to be changed as it did not work since it was normalizing the data instead of using one-hot encoding. Along with that, the sentence generation code needed to be rewritten as it didn't work either. The biggest issue while making and training this network was the inability to load the saved network models on different computers. The model could easily be loaded to resume training on the same computer that it was generated on however the model would not work on any other computer regardless of version matching the APIs. Finally, the network was trained using Keras with Tensorflow-CPU backend. Using Tensorflow-GPU backend or even using the Keras Multi-GPU api would have improved the training time of the model, however due to errors configuring the GPU APIs this could not be completed.

Analysis

After training the network for around 500 epochs (over 24 hours), the network achieved a loss of 0.1713 and was able to generate sentences with some level of readability however many words generated were gibberish. It is possible that changing the number of characters in each sequence would improve the results of this network however it is uncertain whether the number of characters would need to increase or decrease. Adding more LSTM layers to the network while keeping the number of hidden nodes the same or increasing them could also improve the network's performance at the cost of training time. Along with those changes, using a larger dataset would also improve the predictions however this would also require more training time.