Evaluating Basic Convolutional Neural Network Structures

Coursework for Data Science 458: Research Programming Assignment II

Researcher: Michael Rocchio

Graduate Student at Northwestern University

Researcher Contact Email: MichaelRocchio2023@u.northwestern.edu

Abstract

In this research assignment we will be conducting Language Modeling with an RNN. Natural language processing (NLP) is the process of automatically analyzing and understanding human language in order to make sense of it. NLP systems work by analyzing the structure of natural languages, using programs that can achieve a variety of tasks, including machine translation, information retrieval, text classification, speech recognition and generation. They do this by using word embedding which refers to algorithms that aim at representing words (or word senses) as vectors in a continuous semantic space. These vectors can then be used for tasks such as automatic inference or dimensionality reduction for machine learning models such as neural nets.

For instance, if we think about words in terms of their two-dimensional representations in the form of vectors, we can see that similar words will lie close together in the semantic space. This allows us to easily perform tasks like clustering or classification on word data. In more practical terms, word embedding algorithms may also be used for mapping natural language sentences into vector representations for various NLP applications such as

information retrieval and question answering. In addition, some studies have shown that word embeddings may aid in building chatbots with capabilities approaching human-level intelligence.

Research Purpose, Dataset Introduction, & Literature Review

The purpose of the research below will be to, in 6 separate experiments, build and evaluate several different network architectures for the purpose of text classification on the ag_news_subset dataset. Each of these news snippets are then categorized into genres.

For my literature review I read the article cited in the bibliography below. My summary is as follows. The passage begins by stating that convolutional neural networks (CNNs) have surpassed recurrent neural networks (RNNs) in the area of speech and text recognition. It goes on to state that CNNs excel at recognizing features in static images, while RNNs are better at recognizing features in time-dependent problems such as speech and text recognition. The passage states that RNNs are good at all types of sequence problems, including speech/text recognition, language-to-language translation, handwriting recognition, sequence data analysis (forecasting), and even automatic code generation in many different configurations. However, the passage goes on to state that RNNs have an inherent design problem. The problem is that because they read and interpret input text one word (or character) at a time, they cannot take advantage of massive parallel processing (MPP) in the same way CNNs can. This means that RNNs are very compute intensive since all the intermediate results must be stored until the processing of the entire phrase is complete. The passage then discusses how Google and Facebook came up with solutions to solve the problem of speed of translation by using CNNs combined with the attention function. The study is a series of head-to-head benchmark competitions of TCNs versus RNNs, LSTMs, and GRUs across 11 different industry standard RNN problems well beyond language-to-language translation. The conclusion is that TCNs are not only faster but also produced greater accuracy in 9 cases and tied in one. I found this passage to be interesting because it discusses how advances in technology have changed how companies like Google and Facebook operate.

Experiments

One thing to note about the experiments is that they take extremely long to vectorize and train. There were a number of items that took multiple hours to train. Additional on the final iteration I placed my early stopping value too high and it resulted in very overtrained models with train and validation accuracy rates venturing as high as 0.15; as this is the case I would not recommend utilizing these within a production environment. This is also true as the vectorization process included extensive data leakage between the test and train set. This means that some vectorized words were more likely to appear in both sets than ones that would be fed into the model in a production environment.

In each experiment a different vocabulary size, one of 3, and neural network, one of two, will be utilized. In the first experiment we tested a text vectorization with 1000 characters on two different but similar network architectures. It is clear based on **Figure 1** that the deeper architecture is beneficial but at the lower vocab text vectorization length it is negligible. As the vectorization increases the increase in validation accuracy caused by the deeper network should begin to show, however during my experiments it was negligible.

Test	Name	Train_Time	Train_Accuracy	Validation_Accuracy
1	1000_shallow	118.24	99.53	82.55
2	1000_deep	125.15	99.48	84.03
3	2000_shallow	119.38	99.52	83.38
4	2000_deep	200.43	99.34	84.23
5	3000_shallow	116.14	99.33	82.38
6	3000_deep	203.13	99.19	83.83

Conclusion and Final Recommendations

I definitely enjoyed doing this assignment, but it proved to be quite long due to the intense training times caused by the vectorization of the text prior to the training of the model. Going forward I will also be trying to be careful about tuning my parameters to prevent overtraining.

Bibliography

Posted by William Vorhies on May 1, 2018 at 7:29amView Blog. "Temporal Convolutional Nets (Tcns) Take over from Rnns for Nlp Predictions." Data Science Central, www.datasciencecentral.com/profiles/blogs/temporal-convolutional-nets-tcns-take-over-from-rnns-for-nlp-pred.