Deep Learning Project for CSC 416

Anonymous Author(s)
Affiliation
Address
email

Abstract

This paper covers the process through which a deep neural network was created for the ImageNet dataset. It covers what was hoped to have been accomplish and the setbacks that were encountered.

1 Introduction

In the process of attempting to write the program, several consecutive issues were encountered. The code was largely based off of a tutorial from the TensorFlow website which covered the estimator API[1].

1.1 What Went Wrong

The tutorial from the TensorFlow website introduced estimators using the iris dataset. This utilized data from a CSV file, which it parsed and fed into a training algorithm. The goal for this project was to modify the TensorFlow code to read and train on images from the ImageNet dataset. During this process, however, there were several issues encountered to which a solution was never found.

Through the process, many of the issues were overcome, but the problem of adapting the code to the ImageNet dataset proved more difficult than expected. The final issue encountered which no solution was found for was to use the data collected from the images and feeding it into the `DNNClassifier` function. To do this, the data first had to be used to create a `feature_column`, which was then fed into `DNNClassifier`.

It is unknown whether it was the creating a `feature_column` step which was never solved, or if it was feeding that information into the `DNNClassifier` that caused the issue. It is also possible that there was a bug elsewhere in the program which was spilling over into that section.

When the program is run, the most recently encountered error is printed to the screen. Given the error, it is possible for issue to have occurred anywhere in the program. A likely location, aside from the aforementioned areas, could be in the `train_input_fn` function, which was another parameter given to `DNNClassifier`.

1.2 What Was the Goal

As mentioned before, the goal was to modify the tutorial program given by TensorFlow to analyze and classify images from the ImageNet dataset, rather than classifying the iris dataset using CSV files.

Since the program was never finished, different optimizations could not be done. Had there been an opportunity, many different hyperparameters would have been tuned to find the most accurate result. This includes the number of hidden layers and neurons in those layers, the activation function, and the optimization function, among others.

Submitted to 31st Conference on Neural Information Processing Systems (NIPS 2017). Do not distribute.

As the program was never completed, the following the sections could not be expanded upon. However, the outline of what would have been discussed is shown below.

2 Neural Network Architecture

In this section, the program's deep neural network architecture would be explained, as well as the reasons for why it was chosen.

3 Techniques Used to Optimize Results

This section would cover what techniques would have been used to produce the best possible results. For example, an explanation behind the tests performed on different activation functions and which one was chosen to have provided the best results. How the training set was split from the test set would also be explained here.

4 The Results

In this section, the results of the deep neural network would be stated and analyzed. Including tables and visualizations of the results and the sets used to train and test.

5 Future Enhancements

In this final section, the parts of the program which could have been further improved would be discussed. Further tests could also be discussed here.

At the moment, the improvements to be made to the program includes, simply, to make the program run without error.

References

[1] The TensorFlow Authors (2016) *Premade Estimators for ML Beginners.* https://www.tensorflow.org/