Homework3: Neural Network (Alternative)

Yuan-Hung Lo

I. Introduction

In this report I will demonstrate the process of optimizing a neural network model. With the solution codes provided by the instruction, I focused on tweaking variables in the program and seeking to find the best configuration that either reaches the best accuracy or has the highest evaluation score, which takes other aspect of the training result into account and is defined later in this report.

II. Basis of Model Evaluation

A. *Optimizing Variables*

Neural Network Model uses multiple layers of neurons to classify input numeric image value with possible output number. To potentially increase the model's performance, changing the number of neurons or layers could lead to successful results. Another variable that has an impact on the performance in the epochs number. It represents the number of times a dataset is fed through an algorithm. Intuitively, more epochs can lead to more accurate results, but may also cause over fitting. The variable learning rate also plays a major role. It controls the pace at which an algorithm updates its parameter estimation. Smaller learning rate means that the algorithm makes smaller changes each time, which means that it's possible for it to get closer to the optimal result, but sacrifices run speed and can settle on local max/min values. On the other hand, a high learning rate can cause algorithms to skip pass critical solutions and affect results. Lastly, I would also evaluate the effect of batch size, which governs the number of data samples that is fed into the model before its updated.

B. *Creating Evaluation Reference*

There are a lot of different parameters that can be used to describe the goodness of a machine learning model. The ones I use are test accuracy, run time, and train loss. Accuracy calculates the percentage of testing data that the model can classify correctly. With our non-binary and uniform distributed correct answer to training and testing data, accuracy alone can adequately depict the goodness of our prediction and it is not necessary to introduce other means of evaluation such as precision, recall, or f-1 score. Run time serves as a good indicator of how our training environment is set up. A well-designed model training environment should run in a reasonable amount of time while maintaining certain levels of accuracy. We also use logarithmic loss to show the wellness of each guess, and priorities training on inaccurate predictions.

III. Creating Evaluation Score

As stated, I evaluate the performance of our model with test accuracy, run time, and train loss. I aim to create an evaluation score that takes these parameters into account with different weights assigned. This would serve as a optimizing reference and help me determine my changes in variables.

A. *Gathering Data for Parameter Normalization*

For an unbiased evaluation score, each variable is normalized using their maximum and minimum value found by training with large changes in optimizing variables: number of neurons, learning rate, batch size, and epochs. We use two training results per variation and collected a total of 20 training results. Using the extreme value of run time and train loss, we can normalize them and create an unbiases evaluation of the wellness of each parameter value between 0 to 1.

$$value_N = \frac{value - min}{max - min} \quad (1)$$

Weight is then assigned to accuracy, normalized run time and train loss. The purpose of this training process is to find a model that can recognize numbers from images, therefore we assign the largest weight on accuracy so that we value it more than other variables. In practice, although high run time and high train loss can lead to inconvenience in training, they do not pose a large effect on the application of the model. The evaluation score then can be written in equation (2).

$$\begin{gathered} Evaluation\ score = 0.8 \times accuracy \\ +0.1 \times (run\ time_N + train\ loss_N) \quad (2) \end{gathered}$$

Evaluation score is then used to represent goodness of our model. In this way we can focus on optimizing the accuracy of our model while also taking run time and train loss into account.

IV. Training Model

A. *Comprehensive Optimization*

To get an optimization result that takes a comprehensive evaluation of all result parameters, the evaluation score is utilized. With the original configuration, the model is trained with results of 0.921 accuracy, 0.488 normalized run time, and 0.370 normalized training loss, reaching an evaluation score of 0.651. We then optimized the model with each variable and analyzed their effect on the result. We found that although increasing epochs number can increase accuracy, the difference is often minor and was at cost of much longer run time. Therefore, decreasing epochs to a certain level can drastically reduce run time while maintaining accuracy, leading to better scores. We also found that increasing the learning rate in this case can decrease run time with not a significant downside. The effect of neurons in each layer is also looked at, and it is found that increasing the value leads to significant decrease in training loss. However, the accuracy isn't improved, and the run time is greatly increased, causing this method to be inefficient. Changing the batch size is shown to be negatively affecting the training result, with decreasing score at both higher and lower size.

The optimal configuration with the best evaluation score is in the configuration shown in Table. 1. Reaching an evaluation score of 0.66. Proving a balance between performance and efficiency.

neurons	learning rate	batch size	epochs
64	0.02	64	100

test accuracy	Run time	Train loss	score
0.9223	99.0723	63.6837	0.660215

Table 1. Optimal training result with evaluation score

B. Accuracy Analysis

In practice, models are only trained once, and the time and loss of training doesn't affect the application of such models. Therefore, the model is also optimized to achieve the highest level of accuracy to account for such a use case.

From the results of the training process, we see that the biggest factor affecting the accuracy of our model is the number of epochs. The model is then trained using a wide range of epochs' value; the result is shown in Fig. 1 below. We can see that accuracy increases as epochs increase. With limited run time we obtained the best resulting accuracy of 93.6% with 550 epochs.

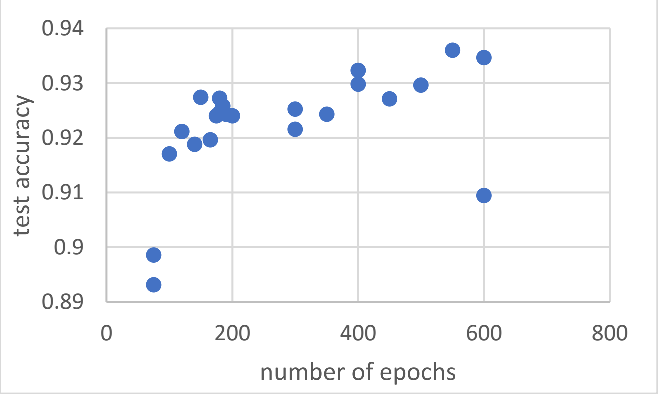

Figure 1. Final shape of rod.

With this trend, our training accuracy could continue to rise as we input larger value of epochs. However due to a significant increase in run time and CPU workload, further testing results are unavailable at this moment.

V. Conclusion

Two optimization processes are carried out and trained with different evaluation methods. To ensure an efficient training process an evaluation scored defined according to our specification is utilized. With this we reached a model that has high accuracy and is easy to train. A model that was trained solely for the purpose of high accuracy is also created by focusing our optimization with higher epochs. Hence, we were able to achieve much better accuracy while sacrificing training efficiency. The training time of such an approach greatly exceeds that of previous optimization result.