COMP 551 Mini-Project 3

Yingjie Xu YJ.XU@MAIL.MCGILL.CA (260862481)
Hengxian Jiang HENGXIAN.JIANG@MAIL.MCGILL.CA (260830557)
Tianhao You TIANHAO.YOU@MAIL.MCGILL.CA (260830663)

Abstract

In this project, we investigated the performance of the Multilayer Perceptron (MLP) machine learning model. We trained the model with the MNIST database for recognizing the handwritten digits. We explored how the number of layers and different activation functions would impact the model performances. By experimenting with 0 hidden layers, 1 hidden layer and 2 hidden layers models, we found that the performance of MLP would increase as the number of layers increases. After training and testing with different activation functions, we found that hyperbolic tangent is the activation function that obtains the best performance. We use mean subtraction as our normalization method, however, by training the model with unnormalized data, we found that this normalization method does not have too much impact on the performance.

1. Introduction

Machine learning is a method used to make predictions or decisions by building mathematical models based on training data. To generate good predictions of data based on decision boundaries, different layers of Multilayer Perceptron(MLP) are investigated to implement the model.

The MNIST dataset is an acronym that stands for the Modified National Institute of Standards and Technology dataset. It is a dataset of 60,000 small square 28×28 pixel grayscale images of handwritten single digits between 0 and 9. The task is to classify a given image of a handwritten digit into one of 10 classes representing integer values from 0 to 9, respectively. This dataset is widely used in machine learning fields especially in the deep learning part.

By analyzing the performance of the model with different layers and activation functions in the MNIST dataset, we found that MLP with 2 hidden layers with Hyperbolic Tangent activation function has the best performance and is around 96.3% after around 100 epochs. MLP with 2 hidden layers with ReLU activation function achieve around 95.8% and MLP with 1 hidden layer with ReLU achieved third-best performance that is around 94.9% after around 100 epochs. And the model with 2 hidden layers and sigmoid as activation function achieved around 88.43% accuracy.

The noticeable discoveries when implementing these models to the dataset are:

Model 1. For MLP with different layers and different activation functions, we found that the more layers we added to the model, the better performance the model can achieve in around the same number of iterations. 2. MLP with ReLU as activation function achieved slightly better performance than the other models. 3. The data normalization process helped our model achieved better performance, the accuracy of

the model with ReLU with 2 hidden layers increased from 86.6% to 95.8%. 4. Applying L2 regularization does not increase our model's accuracy a lot, the accuracy sticks around 95%-96%.

Dataset MNIST is divided into two datasets: the training set has 60,000 examples of hand-written numerals, and the test set has 10,000. MNIST is a subset of a larger dataset available at the National Institute of Standards and Technology. All of its images are the same size, and within them, the digits are centered and size normalized.

The key takeaway in this project was that by adjusting and manipulating the hyper-parameters for MLP models, we have a better understanding of the impact of changing those parameters would have on the model and the interactive relationship between different parameters. And we also learned that image classification required a lot of work in choosing the different learning models and tuning the hyper-parameters. The key to obtaining high accuracy is to implement the most efficient model and to tune the best parameters. Since it takes much time to process big datasets like MINST, we should better choose our model and hyper-parameters wisely.

2. Datasets

2.1. Summary

For this project, we are only using one dataset: the MNIST dataset. The MNIST database (Modified National Institute of Standards and Technology database) contains a large number of handwritten digits and it is usually used for training various image processing systems. All of its images are the same size (28*28), and within them, the digits are centered and size normalized.

2.2. Data processing

To perform further experiments, we need to do some data processing. The first thing we did was rescale the image from [0,255] to [0.0,1.0] by dividing 255 on each element in each set. Also, we make y_train and y_test to be categorical in order to do the comparison. Then, we performed mean subtraction for x_train and x_test, by subtracting the mean of each set, which would center the cloud of data around the origin point. At last, we reshaped the sets. Hence, we got the sets that can be operated.

2.3. Training and testing sets

As mentioned before, the training set has 60,000 examples of hand-written numerals, and the testing set has 10,000 of same numerals

3. Result

3.1. Zero hidden layers

We first trained our MLP with no hidden layers which would directly map the inputs to outputs. We could see that testing and training performance is similar which is around 90%. (Figure 1)

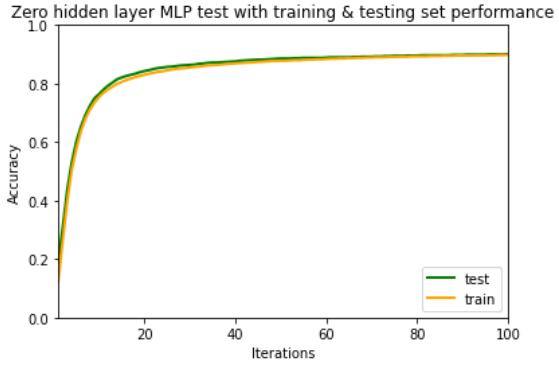

Figure 1: Zero hidden layers

3.2. One hidden layer

We then trained an MLP with a single hidden layer having 128 units and ReLU as activation functions. The performance on the training and testing set is higher than MLP with no hidden layers and reaches an accuracy of around 92%. (Figure 2)

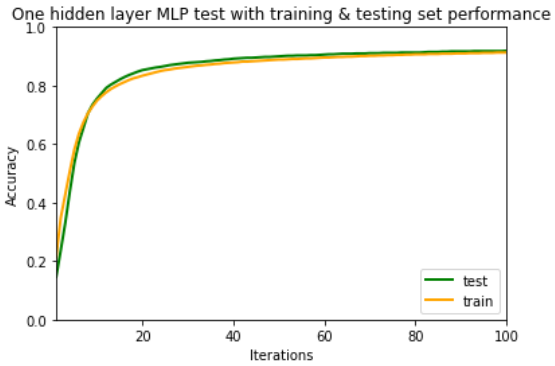

Figure 2: One hidden layer

3.3. Two hidden layers

After adding another layer, we trained an MLP with 2 hidden layers each having 128 units with ReLU activations. The performance on the training and testing set gets better than MLP with only one hidden layer and reaches an accuracy of around 95.6%. (Figure 3)

By applying hyperbolic tangent (Figure 4) and sigmoid (Figure 5) as activation functions with 2 hidden layers MLP, we get an accuracy of 96.27% on a hyperbolic tangent and 86.03% on a sigmoid. By comparing the results with ReLU, we found that hyperbolic tangent is a better activation function on this dataset than ReLU. However, using sigmoid as an activation function does not perform well. We think hyperbolic tangent is better than sigmoid because it is an asymmetric function and it is better for optimization

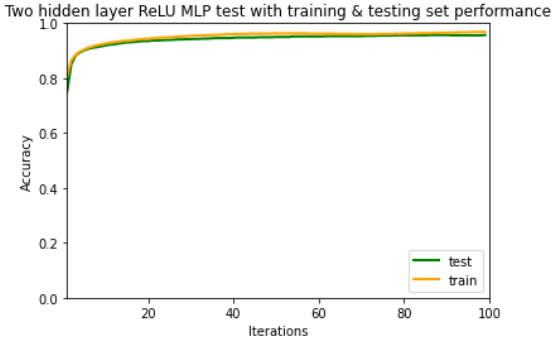

Figure 3: Two hidden layers with ReLU

because its value is close to zero when x is around zero.

Figure 4: Two hidden layers with tanh

Figure 5: Two hidden layers with sigmoid

Then we created an MLP with 2 hidden layers each having 128 units with ReLU activations as above, however, we also added L2 regularization to the cost and train the MLP in this way. We observe that the performance on the testing set reaches 96% which is a bit higher than 95% compared with the model without L2 regularization. It shows that L2 regularization could improve the performance of this dataset. (Figure 6)

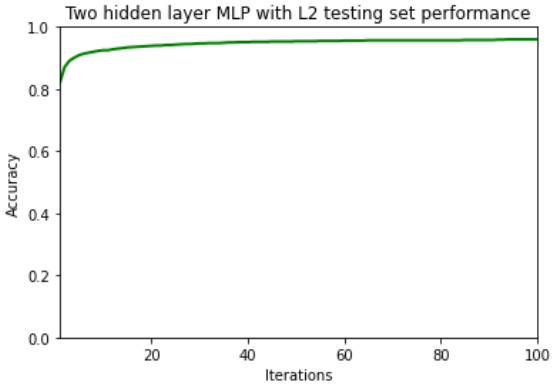

Figure 6: Two hidden layers with L2

We also created an MLP with 2 hidden layers each having 128 units with ReLU activations as above, however, we train it with unnormalized images. The resulting performance on the testing set is 86.6% which is much lower than usual. We think data processing is an important part of getting a great performance. (Figure 7)

3.4. Three hidden layers

As part of our own exploration, we also trained an MLP with 3 hidden layers with ReLU as the activation function. The testing accuracy is 95.69%. It is slightly higher than the one with only 2 hidden layers but not that much. It is also noteworthy that a model with 3 hidden layers would require much more time to train. We think 2 hidden layers are actually enough for this dataset

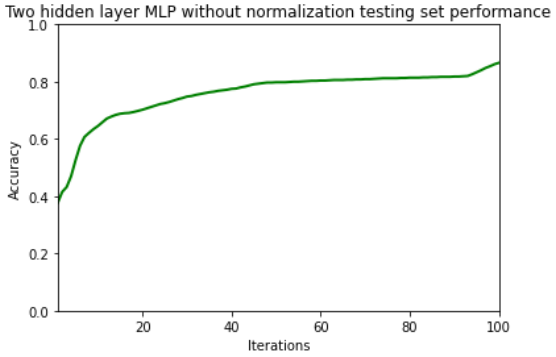

Figure 7: Two hidden layers with unnormalized data

since we do not get too much performance gain by adding another layer to it. (Figure 8)

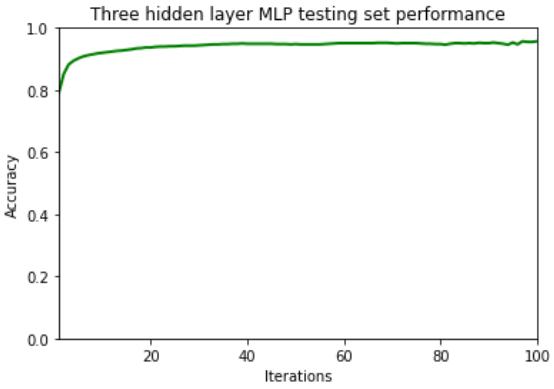

Figure 8: Three hidden layers

3.5. Resulting performance

We also created a table for comparing the resulting performance. (Table 1)

4. Discussion and conclusion

4.1. Key takeaways

1. By adjusting the manipulating the number of hidden layers for MLP, we have a better understanding of the impact of changing

Table 1: Performance Comparision

Model	Performance
0 layer	90.08%
1 layer (ReLU)	91.86%
2 layers (ReLU)	95.65%
2 layers (tanh)	96.27%
2 layers (sigmoid)	86.03%
2 layers (ReLU + L2)	96.02%
2 layers (unnormalized)	86.62%
3 layers (ReLU)	95.69%

the number of hidden layers would have on the model and the training time of the model. In general, more hidden layers lead to higher performance and more training time.

2. By applying different activation functions for our MLP models, we get a better understanding of how to choose between different activation functions and how different activation functions would impact the resulting performance.

3. Image classification requires a lot of work in data cleaning, processing and normalization. By applying those techniques thoroughly and carefully, we could obtain a better performance.

4.2. Future investigation

1. Selecting a great activation function for MLP is not easy, we need to analyze the advantages and disadvantages of different activation functions before applying them. We could try to apply other activation functions and compare the performance with what we currently have.

2. In this project, we did not change the width of our model. We could try to change the width of the MLP and see how would it impact the resulting performance.

5. Statement of Contributions

Hengxian was responsible for implementing the MLP with ReLU, Sigmoid and hyperbolic tangent with 2 hidden layers, wrote the abstract, introduction and discussion and conclusion section in the report. Yingjie was responsible for data processing, implemented MLP with ReLU with 1 hidden layer and MLP with unnormalized data, and wrote the dataset section in the report. Tianhao was responsible for data processing and implementing L2 regularization for the MLP with ReLU and 2 hidden layers models and wrote the result section in the report.