# COMP-551 Mini-Project 3

COMP-551 Winter 2021

Arneet Singh Kalra, Behnaz Ghefati Feizabadi, Hussein Lakkis

April 1, 2021

**Abstract**

Neural networks are powerful tools to model complex patterns in large databases. In this project, we implemented a neural network, the Multilayer Perceptron, to classify image data from the MNIST (Modified National Institute of Standards and Technology) Dataset containing handwritten digits. A variety of experiments tested the performance of the network with varying hidden layers and respective units, activation functions and the use of regularization. Our results showed that the model performed best with 2 hidden layers of 256 units using ReLU activations, achieving a testing accuracy of 98.19%. Increasing the number of hidden layers with ReLU activations up to 2 layers resulted in an overall increase in testing accuracy. Furthermore, increasing the width of the hidden layers also increased testing accuracy for MLPs with both ReLU and Tanh activations, but not for Sigmoid functions which capped at 128 units. Lastly, using L2 regularization on the MNIST dataset decreased testing accuracy, likely a result of overregularization.

## 1 Introduction

### 1.1 Project Background and Goals

The goal of this project is to implement a Multilayer Perceptron, and train and test it on the MNIST Dataset of handwritten digits. The original dataset comes from the National Institute of Standards and Technology, a branch of the United States Department of Commerce. The dataset was normalized to fit into a 28 by 28 pixel bounding box and anti-aliased, resulting in a gray-scale. This modified dataset is commonly used in training machine learning algorithms. In our experiments, we tested a variety of different activation functions, learning rates, number of hidden layers, varying layer widths, and other key metrics such as L2 regularization. Our best model resulted in an accuracy of 98.19%. The model used a learning rate of 0.01, 2 hidden layers, 512 units in each layer, and ReLU activations. The model performed the worst on unnormalized images, with a testing accuracy of 11.46% (slightly better than random chance). Using a ReLU activation, increasing the number of hidden layers up to 2 and the number of units in each hidden layer, increased the testing accuracy. However, adding L2 regularization to the cost decreased the accuracy of the network.

### 1.2 Discussion of Related Work

The MNIST dataset is a popular set used in academics to test different models. In this project, we developed a Multilayer Perceptron, however there exist a variety of neural networks which can be trained on this dataset. At the time of this project, one of the most efficient set of models are Convolutional Neural Networks (CNNs). A study by Kumar et al. showcases their CNN model which is composed of multiple convolutions and pooling layers alongisde ReLU activations [2]. The CNN models performs exceptionally well, receiving a test accuracy greater than 98% [2]. In the original paper where MNIST dataset was used, a Support Vector Machine was able to achieve an error rate of 0.8% [4].While examining other articles, we came across a variety of other real-life applications for MLPs. A study published in the Journal of Food Engineering uses MLPs to model the soaking characteristics of wheat kernel at varying temperatures. The soaking temperature and time were used as input parameters and the moisture ratio was used as output parameter [3]. The resulting network performed relatively well when compared to the actual experimental data. Another paper from Boughara et al. describes the use of a single hidden layer MLP in facial recognition applications to classify facial expressions from three large datasets [1]. The results of the study clearly demonstrate the efficiency of their model.

## 2 Dataset

The MNIST dataset contains 60 000 training images and 10 000 testing images of handwritten digits between 0 and 9. Each image is a 28 x 28 pixel bounded box, encompassed as a 2D array. Each value in the array represents the gray-scale factor of the pixel, a value between 0 and 255, as visualized in Figure 2. Figure 3 shows the distribution of the values between 0 and 255. As expected, a majority of the pixels contain a value of 0, representing no part of the digit present at the respective pixel. Therefore, during the pre-processing of the data, the digits were first normalized to have values between 0.0 and 1.0. This step of normalization reduces the complexity of the model, improves the accuracy, speeds up the training time, and reduces variance. Then, the images were flattened to convert the 2D array into a one dimensional array, the input vector, of size 784. Finally, for a majority of the experiments, 10 000 data samples from the training set were isolated as the validation set, leaving 50 000 for training.

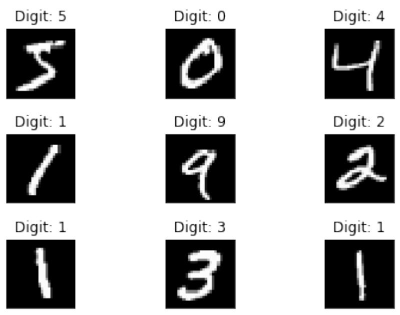

Figure 1: Sample Handwritten Digits

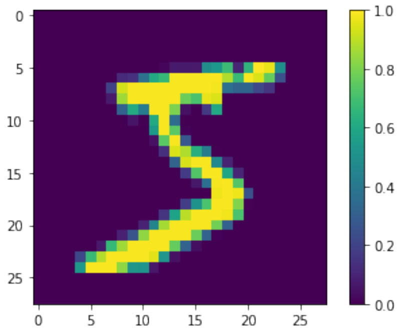

Figure 2: Handwritten Digit from Dataset Colourized

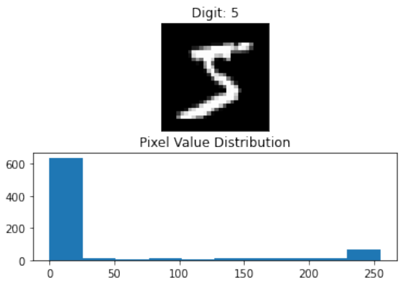

Figure 3: Pixel Distribution of a Digit in MNIST

## 3 Results

### 3.1 Varying Number of Hidden Layers with ReLU Activation

In our first experiment, we ran three iterations of our model: 0 hidden layers, 1 hidden layer, and 2 hidden layers. The MLPs with hidden layers used ReLU activations, and since we are performing classification, had a Softmax layer at the end. Each hidden layer had 128 units. By monitoring the training accuracy against the number of iterations, we concluded that the training accuracy started to stay constant after 60 epochs for 0 and 1 hidden layers. An epoch of 40 was used for 2 hidden layers. Then, hyperparameter tuning was performed on each model to find the optimal learning rate, as demonstrated by figures 7 to 9. All 3 models performed best using a learning rate of 0.1. As indicated by Table 1, the best model out of the three used 2 hidden layers and received a test accuracy of 98.01%. It also received a perfect training accuracy. On the other hand, the model with no hidden layers performed the worst among the three, receiving a testing and training accuracy lower than or equal to 93%. Therefore, we see that with more hidden layers, our results are more accurate. The reason for this is perhaps using softmax regression (no hidden layers) does not capture the complexity of the data. Therefore, when hidden layers are added, and non-linearities are applied, the complex feature (vertical edges, or patterns) are resolved, thus getting a better-fit model. It is worth mentioning that adding additional hidden layers might lead to an overfit model.

| Hidden Layers | Train Accuracy | Test Accuracy(%) |
|---|---|---|
| 0 | 93.00 | 92.54 |
| 1 | 99.93 | 97.79 |
| 2 | 100.00 | 98.01 |

Table 1: Varying Number of Hidden Layers Effect on Test Accuracy

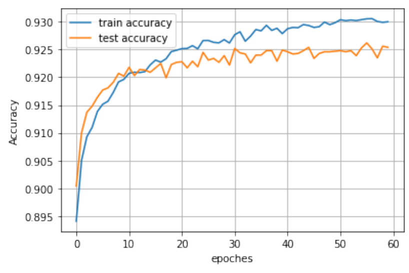

Figure 4: No Hidden Layer Accuracies

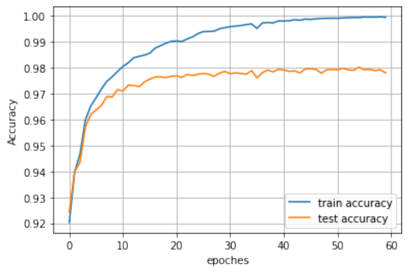

Figure 5: 128 unit Hidden Layer ReLU Accuracies

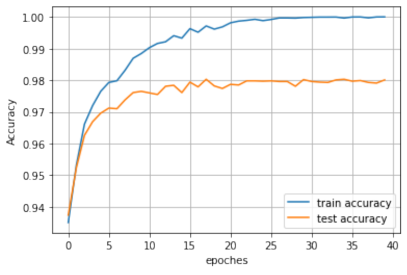

Figure 6: 128 unit 2 Hidden Layers ReLU Accuracies

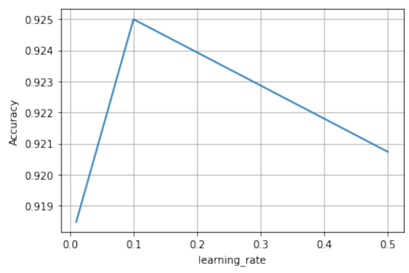

Figure 7: No Hidden Layer Tuning

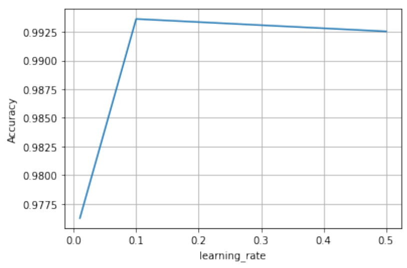

Figure 8: 128 unit Hidden Layer ReLU Tuning

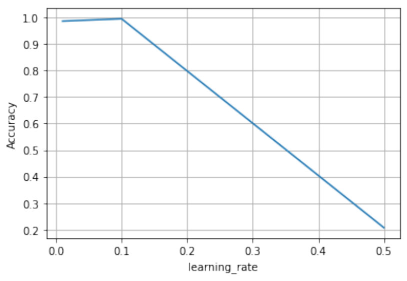

Figure 9: 128 unit 2 Hidden Layers ReLU Tuning

### 3.2 Varying Activation Functions

Using the same model as above with 2 hidden layers and 128 units, we ran experiments with 2 other types of activation functions: Sigmoid and Hyperbolic tangent. Hyperparameter tuning was performed on the two models to find the optimal learning rate. The Sigmoid activation function model performed best with a 0.5 learning rate, whereas the Tanh model was best with 0.1 just as the ReLU model. Figure 10 and 11 plot the accuracy of the Sigmoid and Hyperbolic tangent MLPs respectively, relative to epochs. Our results showed that the Sigmoid model performed the best, achieving a test accuracy of 98.09%. The ReLU model performed a tad lower with a test accuracy of 98.01%. However, all results are approximately the same and therefore no difference was observed. This might be due to random chance.

| Activation Function | Learning Rate | Train Accuracy | Test Accuracy(%) |
|---|---|---|---|
| ReLU | 0.1 | 100.00 | 98.01 |
| Sigmoid | 0.5 | 99.98 | 98.09 |
| Tanh | 0.1 | 100.00 | 98.04 |

Table 2: Varying Activation Function Effect on Test Accuracy

### 3.3 L2 Regularization

In this experiment, we added L2 regularization to the cost of the MLP with 2 hidden layers using ReLU activations. During the hyperparameter tuning, we tested learning rates of 0.01 and 0.1, as well as lambda values of 0.3, 0.1 and 0.01. Figure 12 plots the parameters with their respective accuracies. Our optimal values are 0.1 for the learning rate and 0.01 for lambda. With these parameters, our model received a

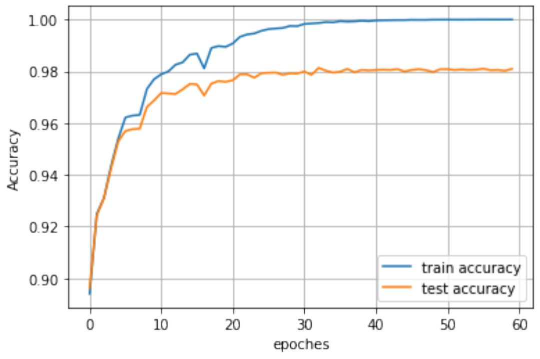

Figure 10: Accuracy of Sigmoid Activation

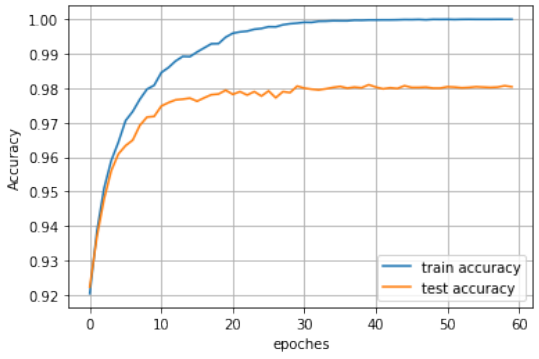

Figure 11: Accuracy of Tanh Activation

training accuracy of 97.33% and a testing accuracy of 97.05%. Comparing this to the model without L2 regularization, this modification to the cost resulted in a decrease of testing accuracy by 0.96%. One reason might be that regularization with the hyper-parameters and activation functions we have might be leading to over-fitting. Moreover, neural networks have more than one minimum and with the learning rate and activation functions used, we might be converging to a local optimum. ReLU might not be the best function here. More experiments should be done to test the effect of regularization.

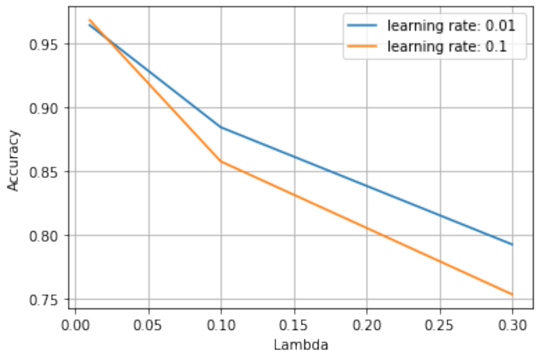

Figure 12: Hyperparameter Tuning of L2 Regularization Model

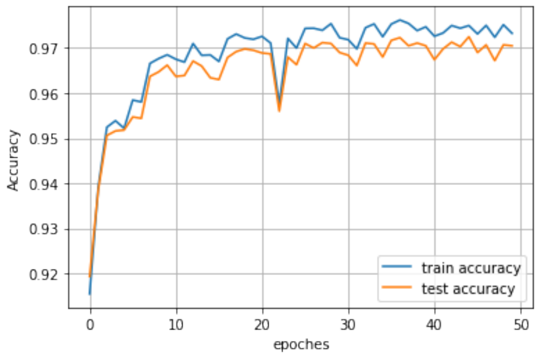

Figure 13: Accuracy with L2 Regularization

## 3.4 Unnormalized Images

This model used 2 hidden layers with 128 units and ReLU activations, but was trained using unnormalized images. Therefore the range of each value in the unit vector is between 0 and 255. Using 20 epoches and a learning rate of 0.1, the model resulted in a training accuracy of 9.87% and a test accuracy of 9.80%. This is to be expected since we now have values ranging up to 255 in the input vector. These values result in the weight sum of the layer to be far from 0 taking away the non-linear aspect of the activation function.

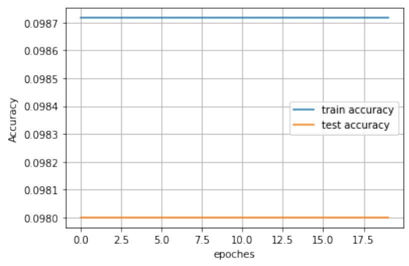

Figure 14: Accuracy with Unnormalized Data

## 3.5 Additional Experiments

### 3.5.1 Varying Widths of Hidden Layers

We investigated the effect that the width of the hidden layers has on the testing accuracy. During the experiments, we tested 64, 128, 256 and 512 unit hidden layers with ReLU activations. For the models with Sigmoid and Tanh activations, we test 2 hidden layers and tested 64, 128, and 526 units. Table 3 shows the results of our various experiments. We achieve the best test accuracy of 98.19% using 2 hidden layers with 512 units each and a ReLU activation function. Using 256 units instead on the same model results a 0.02% lower accuracy. With the models using Tanh, the most accurate result is using a hidden layer width of 256 units. However, for the model with the Sigmoid function, 256 units is the least accurate: the most accurate uses a hidden layer width of 128 units.

| Hidden Layers | Units | Activations | Test Accuracy(%) |
|---|---|---|---|
| 0 | - | - | 92.54 |
| 1 | 64 | ReLU | 97.47 |
| 1 | 128 | ReLU | 97.79 |
| 1 | 256 | ReLU | 97.99 |
| 1 | 512 | ReLU | 98.09 |
| 2 | 64 | ReLU | 97.79 |
| 2 | 128 | ReLU | 98.01 |
| 2 | 256 | ReLU | 98.17 |
| 2 | 512 | ReLU | 98.19 |
| 2 | 64 | Tanh | 97.64 |
| 2 | 128 | Tanh | 97.99 |
| 2 | 256 | Tanh | 98.16 |
| 2 | 64 | Sigmoid | 97.27 |
| 2 | 128 | Sigmoid | 98.09 |
| 2 | 256 | Sigmoid | 97.17 |

Table 3: Accuracies of Varying Number of Units in Hidden Layer(s)

### 3.5.2 Varying Number of Training Images

Lastly, we observed the test accuracy based on the number of training images used. Tested values are $10^k$ where k ∈ 0, 1, 2, 3, 4. Table 4 shows the accuracies based off the number of images. The test accuracies with 1, 10 and 100 training images are all extremely low at 11.67%. This is expected as the model hasn't seen enough data to learn the parameters, converge, and make the correct decisions. As the number of samples in the training data increases, the more accurate the model becomes. When 10 000 training images are used, a testing accuracy greater than 95% is achieved. The training accuracies are approximately the same as the testing accuracies.

| Num of Training Images | Train Accuracy(%) | Test Accuracy(%) |
|---|---|---|
| 1 | 12.28 | 11.67 |
| 10 | 12.28 | 11.67 |
| 100 | 12.28 | 11.67 |
| 1000 | 82.58 | 82.58 |
| 10000 | 95.30 | 95.37 |

Table 4: Accuracies of Varying Number of Training Images

## 4 Discussion and Conclusion

Based on our models and analyses, we conclude that for MNIST Dataset of handwritten digits, the best MLP model has a test accuracy of 98.19%. This is achieved by first normalizing the data and reshaping the inputs to a unit vector of length 784. The most accurate MLP has 2 hidden layers of 512 units each and use ReLU activations without L2 regularization. The neural network has a learning rate of 0.1 and uses 40 training epochs. Comparing the accuracies of 0 hidden layers to 2 hidden layers of an MLP with ReLU activation, 2 hidden layers is much more accurate. However, in this case, using a Sigmoid activation is more or less the same as using ReLU. Furthermore, for the MNIST dataset, adding L2 regularization to the cost causes a decrease in test accuracy. Lastly, increasing the width of the hidden layers increases the testing accuracy with ReLU and Tanh activations, however when using the Sigmoid activation, the optimal test accuracy is achieved with 128 units. Further tests can be done on the MNIST dataset to explore at which width the hidden layers must be to see diminishing returns on test accuracy with ReLU and Tanh activations.

## 5 Statement of Contributions

Arneet Singh Kalra completed the additional experiments and developed the report which was then proofread by the team members. Behnaz Ghefati Feizabadi completed the mandatory experiments. Hussein Lakkis developed the Multilayer Perceptron with regularization.

## References

[1] Hayet Boughrara et al. "Facial expression recognition based on a mlp neural network using constructive training algorithm." In: *Multimedia Tools and Applications* 75.2 (2016), pp. 709–731.

[2] Adhesh Garg et al. "Validation of random dataset using an efficient CNN model trained on MNIST handwritten dataset." In: *2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)*. IEEE. 2019, pp. 602–606.

[3] M Kashaninejad, AA Dehghani, and M Kashiri. "Modeling of wheat soaking using two artificial neural networks (MLP and RBF)." In: *Journal of food engineering* 91.4 (2009), pp. 602–607.

[4] Yann LeCun et al. "Gradient-based learning applied to document recognition." In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.