

EE 228 HW#2 - Neural Network for MNIST

Name: Shaorong Zhang
SID: 862394226

May 4, 2023

1. (2 pts) Apply the normalization on the training and test data.

Solution: see HW2 - Code.ipynb.

2. (2 pts) As a baseline, train a linear classifier $\hat{y} = \mathbf{v}^T \mathbf{x}$ and quadratic loss. Report its test accuracy.

Solution: The final test accuracy is 83.36%.

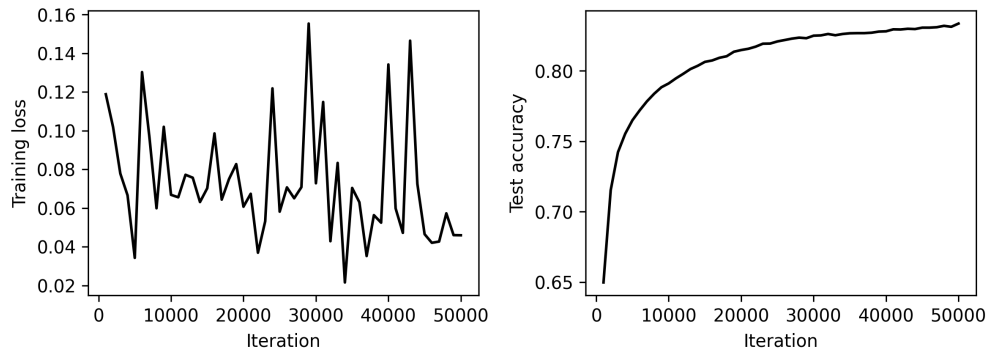


Figure 1: Training loss and test accuracy for linear classifier

3. (7 pts) Train a neural network classifier with quadratic loss $l(y, f(x)) = (y - f(x))^2$. Plot the progress of the test and training accuracy (y-axis) as a function of the iteration counter t (x-axis) 2. Report the final test accuracy for the following choices

- $k=5$
- $k=40$
- $k=200$
- Comment on the role of hidden units k on the ease of optimization and accuracy.

Solution:

Training and test accuracy for network classifier with quadratic loss are shown in the following figures.

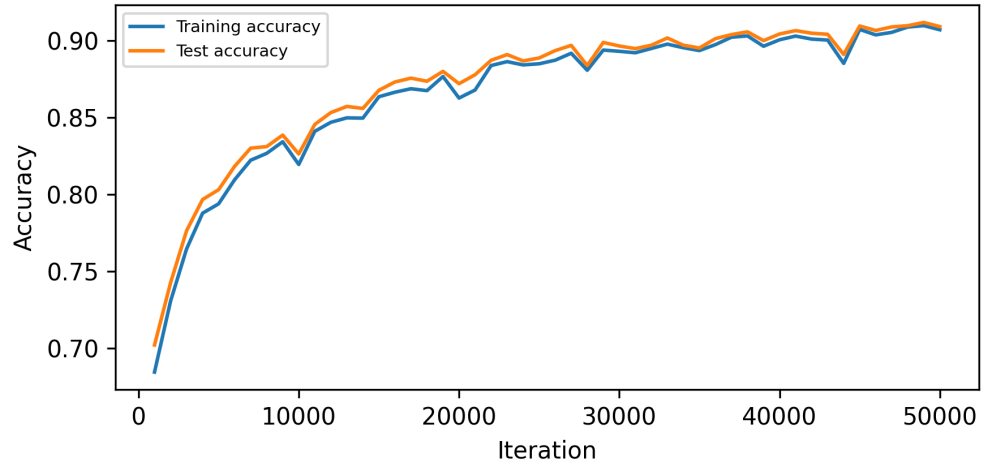


Figure 2: Training and test accuracy for network classifier with quadratic loss as $k = 5$

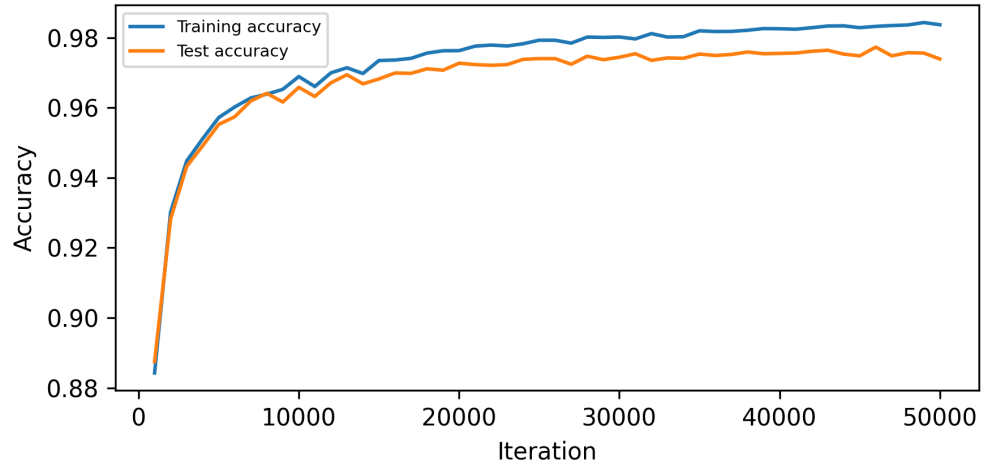


Figure 3: Training and test accuracy for network classifier with quadratic loss as $k = 40$

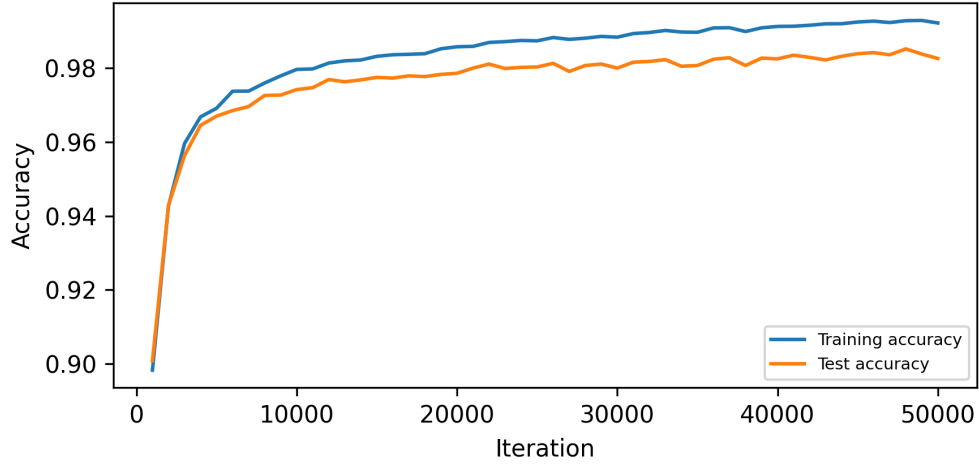


Figure 4: Training and test accuracy for network classifier with quadratic loss as $k = 200$

The parameter configuration, final test accuracy are total training time are shown in the Table 1. With the increase of hidden unit k , the final accuracy is also growing. In the mean time, The training time for one iteration is increasing.

Table 1: Results for neural network classifier with quadratic loss

Hidden units	Learning rate	Training time (sec)	Final test accuracy
5	10^{-3}	5.96	90.92%
40	10^{-3}	24.25	97.39%
200	10^{-3}	75.31	98.26%

4. (7 pts) Train a neural network classifier with logistic loss, namely $l(y, f(\mathbf{x})) = -y \log(\sigma(f(\mathbf{x}))) - (1 - y) \log(1 - \sigma(f(\mathbf{x})))$ where $\sigma(x) = 1/(1 + e^{-x})$ is the sigmoid function. Repeat step 3.

Training and test accuracy for network classifier with quadratic loss are shown in the folowing figures.

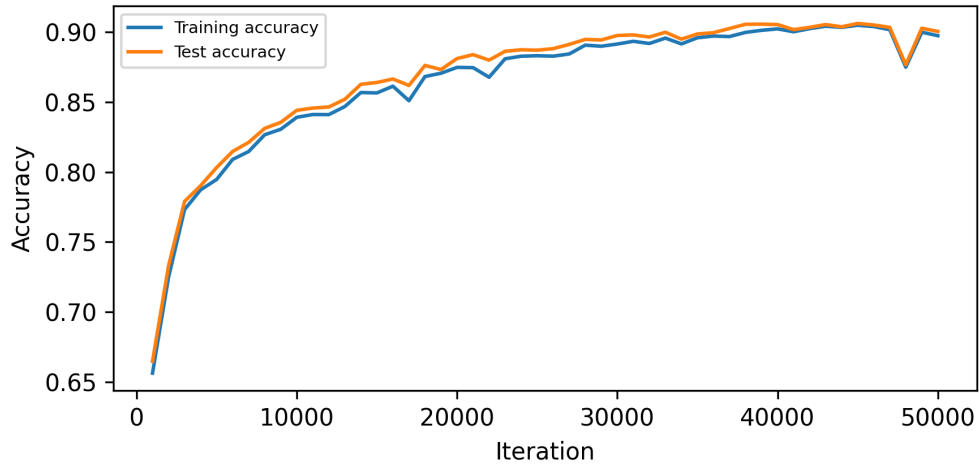


Figure 5: Training and test accuracy for network classifier with logistic loss as $k = 5$

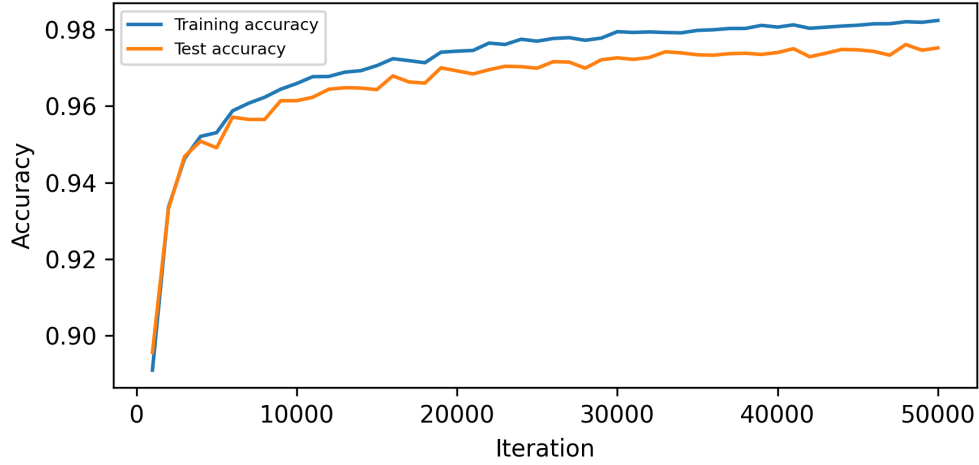


Figure 6: Training and test accuracy for network classifier with logistic loss as $k = 40$

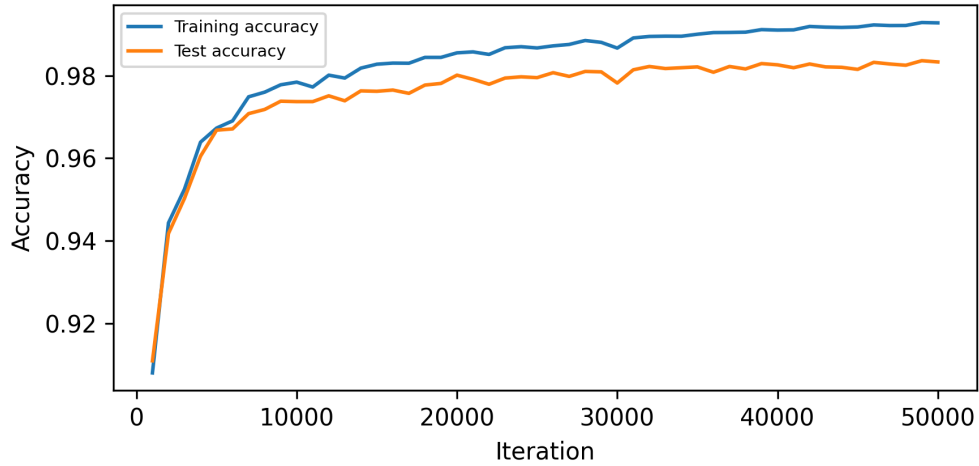


Figure 7: Training and test accuracy for network classifier with logistic loss as $k = 200$

The parameter configuration, final test accuracy are total training time are shown in the Table 2. Same as neural network classifier with quadratic loss, With the increase of hidden unit k , the final accuracy is also growing. In the mean time, The training time for one iteration is increasing.

Hidden units	Learning rate	Training time (sec)	Final test accuracy
1	10^{-3}	5.67	90.05%
40	10^{-3}	23.90	97.52%
200	10^{-3}	71.81	98.33%

5. (2 pts) Comment on the difference between linear model and neural net. Comment on the differences between logistic and quadratic loss in terms of optimization and test/train accuracy.

Solution

Difference between linear model and neural net in terms of optimization and test/train accuracy. It seems that neural net have a much better performance than linear model, especially for dataset which is not linear seperatable. Nonlinear activation function played a very important role to fit complicated function mapping, which greatly increased the capacity of the model. Meanwhile, neural network model usually have a larger number of parameters, which require very high computing resources.

Difference between logistic and quadratic loss in terms of optimization and test/train accuracy. In this task, neural network model with logistic and quadratic loss have similar final test accuracy and training time.