Computational Microelectronics L9

Sung-Min Hong

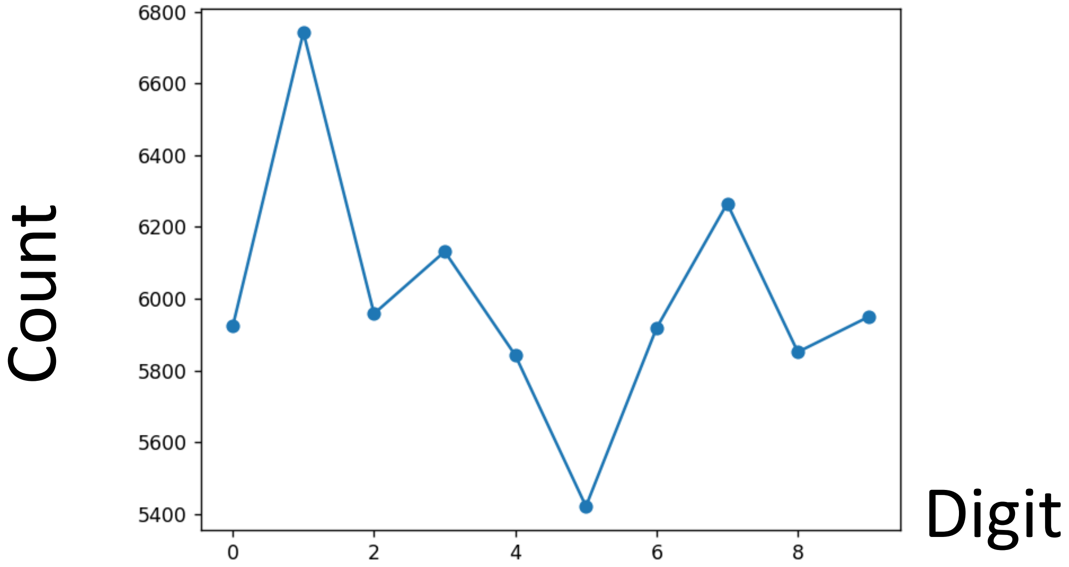
smhong@gist.ac.kr

Semiconductor Device Simulation Laboratory, GIST

Neural network

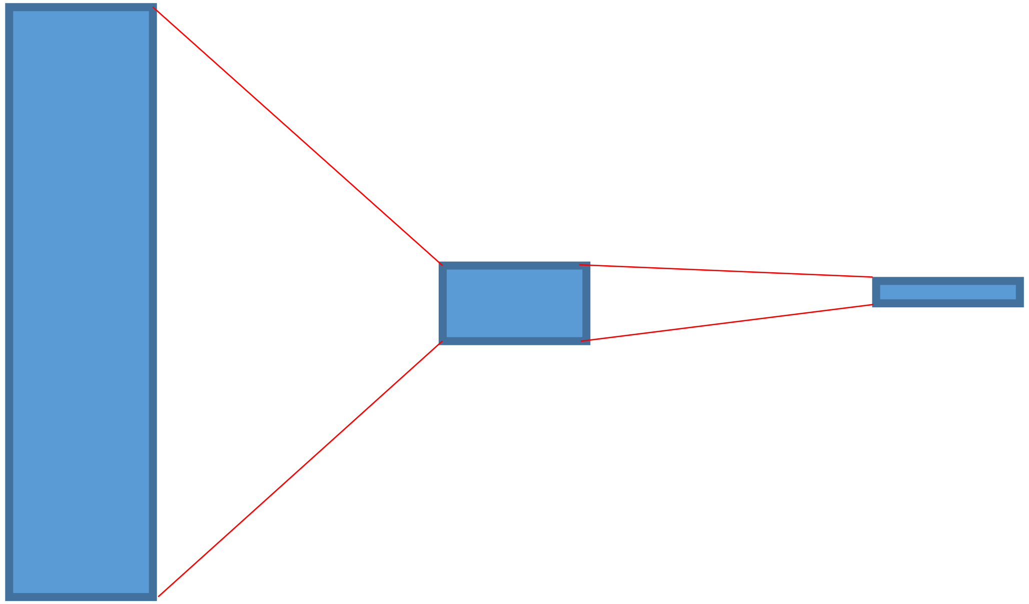
Prepare dataset.

- Reference: https://numpy.org/numpy-tutorials/content/tutorial-deep-learning-on-mnist.html
- Load mnist_train.csv and mnist_test.csv.
  - From these two files, prepare images and labels. For example, they can be (A matrix, 60,000X784), (A vector, 60,000X1), (A matrix, 10,000X784), and (A vector, 10,000X1), respectively.



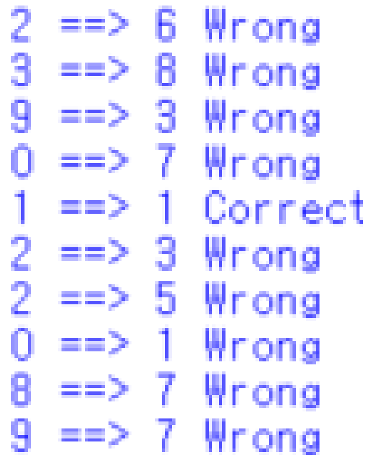
Build a small neural network.

- Input layer, hidden layer, output layer
  - An image is loaded in the input layer. (A vector, 784X1)
  - Matrix-vector multiplication in the hidden layer. (A vector, 100X1) Then, ReLU activation. Then, dropout
  - Matrix-vector multiplication in the output layer. (A vector, 10X1)
  - (No bias term in this example)



First, random weights

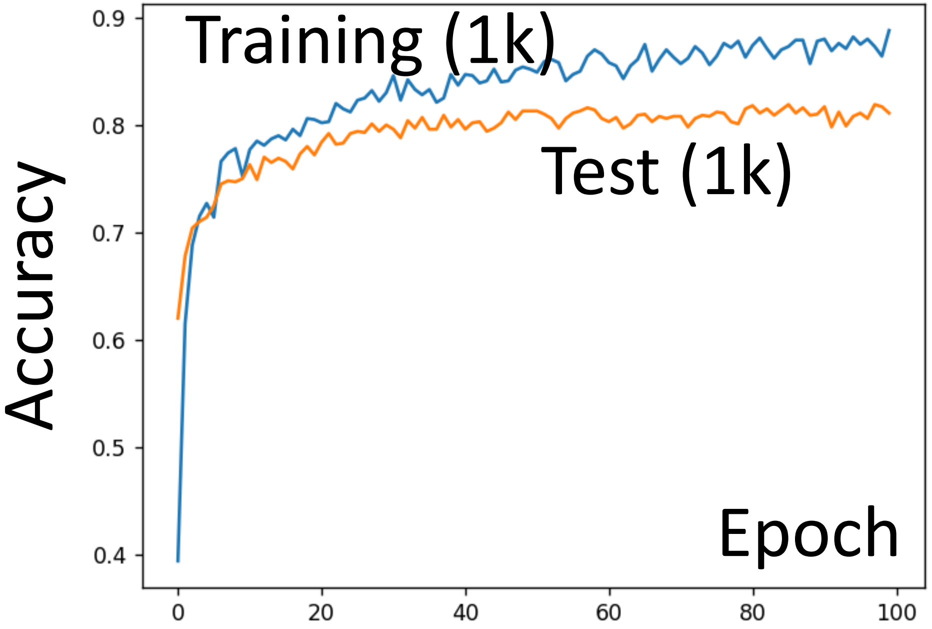
- Let’s start with random weights.
  - 1) weights_1 (A matrix, 100X784)
  - 2) weights_2 (A matrix, 10X100)
  - Every entry in weights_1 and weights_2 is within a range of [-0.1,0.1].
  - Dropout: Half of the hidden layer outputs are dropped out. Instead, surviving entries are doubled.
- Accuracy?
  - For 10k training set, the accuracy is 11.31 %. (As expected)

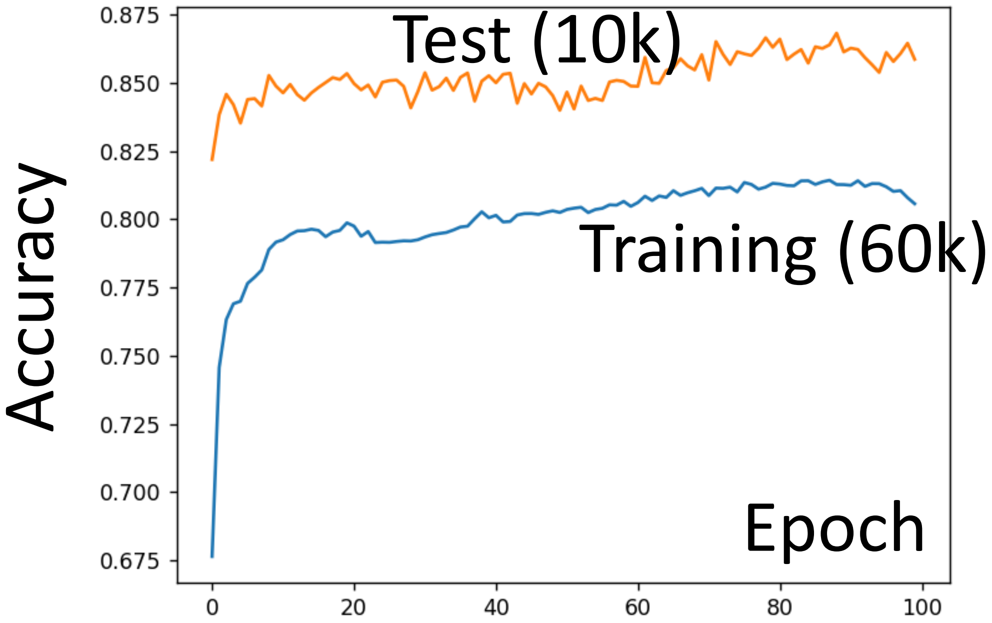


```
2 ==> 6 Wrong
3 ==> 8 Wrong
9 ==> 3 Wrong
0 ==> 7 Wrong
1 ==> 1 Correct
2 ==> 3 Wrong
2 ==> 5 Wrong
0 ==> 1 Wrong
8 ==> 7 Wrong
9 ==> 7 Wrong
```

Skip the training phase.

- Pre-trained neural network
  - Two cases: 1k training samples (learning rate of 0.005) & 60k training samples (learning rate of 0.001)





- You can find them in our GitHub repository. (They're transposed.)

Homework#9

- Due: AM08:00, October 15
- Problem#1
  - Implement a neural network for detecting the MNIST dataset. Calculate the accuracy of your neural network.

Thank you for your attention!