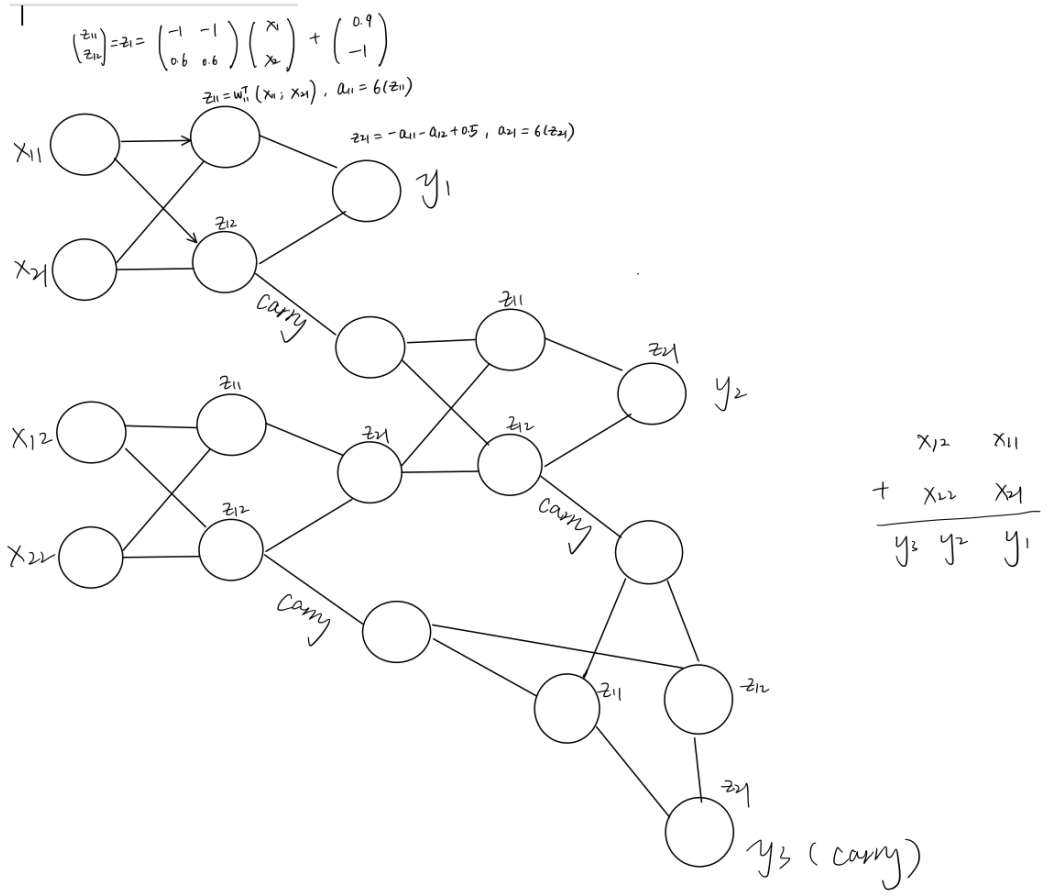
Note: This pdf file include problem 4 #5 and problem 5 #1. The rest of the answer is in the ipynb file.

Problem 4 #5



Problem 5 #1

We first use 30 hidden neurons, use SGD to learn from MNIST training_data over 30 epochs, with a mini-batch size of 10, and a learning rate of 3.0. We got a classification rate of about 84.29%. In the book,

the author got 95.42%.

```
>>> import mnist_loader
>>> training_data, validation_data, test_data = mnist_loader.load_data_wrapper()
>>> import network
>>> net = network.Network([784, 30, 10])
>>> net.SGD(training_data, 30, 10, 3.0, test_data=test_data)
Epoch 0: 8020 / 10000
Epoch 1: 8147 / 10000
Epoch 2: 8237 / 10000
Epoch 3: 8266 / 10000
Epoch 4: 8262 / 10000
Epoch 5: 8333 / 10000
Epoch 6: 8360 / 10000
Epoch 7: 8386 / 10000
Epoch 8: 8358 / 10000
Epoch 9: 8396 / 10000
Epoch 10: 8383 / 10000
Epoch 11: 8402 / 10000
Epoch 12: 8359 / 10000
Epoch 13: 8402 / 10000
Epoch 14: 8394 / 10000
Epoch 15: 8408 / 10000
Epoch 16: 8387 / 10000
Epoch 17: 8417 / 10000
Epoch 18: 8413 / 10000
Epoch 19: 8405 / 10000
Epoch 20: 8424 / 10000
Epoch 21: 8423 / 10000
Epoch 22: 8422 / 10000
Epoch 23: 8425 / 10000
Epoch 24: 8413 / 10000
Epoch 25: 8397 / 10000
Epoch 26: 8433 / 10000
Epoch 27: 8428 / 10000
Epoch 28: 8424 / 10000
Epoch 29: 8429 / 10000
```

Then we change the number of hidden neurons from 30 to 100. We got a classification rate of about 87.08%. In the book, the author got 95.42%. Both my and the author's results got improved by changing the number of hidden neurons.

```
>>> net = network.Network([784, 100, 10])
>>> net.SGD(training_data, 30, 10, 3.0, test_data=test_data)
Epoch 0: 8158 / 10000
Epoch 1: 8468 / 10000
Epoch 2: 8537 / 10000
Epoch 3: 8606 / 10000
Epoch 4: 8619 / 10000
Epoch 5: 8636 / 10000
Epoch 6: 8615 / 10000
Epoch 7: 8652 / 10000
Epoch 8: 8650 / 10000
Epoch 9: 8672 / 10000
Epoch 10: 8677 / 10000
Epoch 11: 8690 / 10000
Epoch 12: 8679 / 10000
Epoch 13: 8696 / 10000
Epoch 14: 8698 / 10000
Epoch 15: 8700 / 10000
Epoch 16: 8680 / 10000
Epoch 17: 8687 / 10000
Epoch 18: 8713 / 10000
Epoch 19: 8706 / 10000
Epoch 20: 8696 / 10000
Epoch 21: 8705 / 10000
Epoch 22: 8707 / 10000
Epoch 23: 8708 / 10000
Epoch 24: 8720 / 10000
Epoch 25: 8710 / 10000
Epoch 26: 8717 / 10000
Epoch 27: 8718 / 10000
Epoch 28: 8709 / 10000
Epoch 29: 8708 / 10000
```

Then, we chose the learning rate to be 0.001. We got 31.51% and the author got 21.42%, which is much less encouraging. We can see that the performance is getting slowly better over time. The results suggest increase the learning rate, we get better results, which suggests increasing the learning rate again.

```
>>> net = network.Network([784, 100, 10])
```

```
>>> net.SGD(training_data, 30, 10, 3.0, test_data=test_data)
```

```
>>> net.SGD(training_data, 30, 10, 0.001, test_data=test_data)
Epoch 0: 1216 / 10000
Epoch 1: 1592 / 10000
Epoch 2: 1751 / 10000
Epoch 3: 1845 / 10000
Epoch 4: 1934 / 10000
Epoch 5: 2016 / 10000
Epoch 6: 2098 / 10000
Epoch 7: 2166 / 10000
Epoch 8: 2234 / 10000
Epoch 9: 2317 / 10000
Epoch 10: 2385 / 10000
Epoch 11: 2444 / 10000
Epoch 12: 2501 / 10000
Epoch 13: 2568 / 10000
Epoch 14: 2612 / 10000
Epoch 15: 2649 / 10000
Epoch 16: 2677 / 10000
Epoch 17: 2718 / 10000
Epoch 18: 2747 / 10000
Epoch 19: 2787 / 10000
Epoch 20: 2825 / 10000
Epoch 21: 2859 / 10000
Epoch 22: 2903 / 10000
Epoch 23: 2930 / 10000
Epoch 24: 2963 / 10000
Epoch 25: 3010 / 10000
Epoch 26: 3038 / 10000
Epoch 27: 3075 / 10000
Epoch 28: 3112 / 10000
Epoch 29: 3151 / 10000
```

Finally, we change the learning rate to 100. We get 9.40% and the author got 9.82%. The learning rate is too high.

```
>>> net = network.Network([784, 100, 10])
>>> net.SGD(training_data, 30, 10, 0.001, test_data=test_data)
```

```
>>> net = network.Network([784, 30, 10])
>>> net.SGD(training_data, 30, 10, 100.0, test_data=test_data)
Epoch 0: 1044 / 10000
Epoch 1: 1311 / 10000
Epoch 2: 1315 / 10000
Epoch 3: 1309 / 10000
Epoch 4: 1291 / 10000
Epoch 5: 1278 / 10000
Epoch 6: 1283 / 10000
Epoch 7: 1281 / 10000
Epoch 8: 1279 / 10000
Epoch 9: 1283 / 10000
Epoch 10: 1397 / 10000
Epoch 11: 1441 / 10000
Epoch 12: 1484 / 10000
Epoch 13: 936 / 10000
Epoch 14: 936 / 10000
Epoch 15: 937 / 10000
Epoch 16: 937 / 10000
Epoch 17: 939 / 10000
Epoch 18: 939 / 10000
Epoch 19: 940 / 10000
Epoch 20: 940 / 10000
Epoch 21: 940 / 10000
Epoch 22: 939 / 10000
Epoch 23: 938 / 10000
Epoch 24: 939 / 10000
Epoch 25: 938 / 10000
Epoch 26: 939 / 10000
Epoch 27: 939 / 10000
Epoch 28: 939 / 10000
Epoch 29: 940 / 10000
```

Therefore, the same conclusion is obtained as given in the book.