Huy Doan

In this assignment, I'm using SVD to compress the model. I'm creating a model with the first hidden layer has 100 neurons, the next hidden layer has 50 and the final output has 10 neurons.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
flatten (Flatten)            (None, 784)               0
_________________________________________________________________
dense (Dense)                (None, 100)               78500
_________________________________________________________________
dense_1 (Dense)              (None, 50)                5050
_________________________________________________________________
dense_2 (Dense)              (None, 10)                510
=================================================================
Total params: 84,060
Trainable params: 84,060
Non-trainable params: 0
```

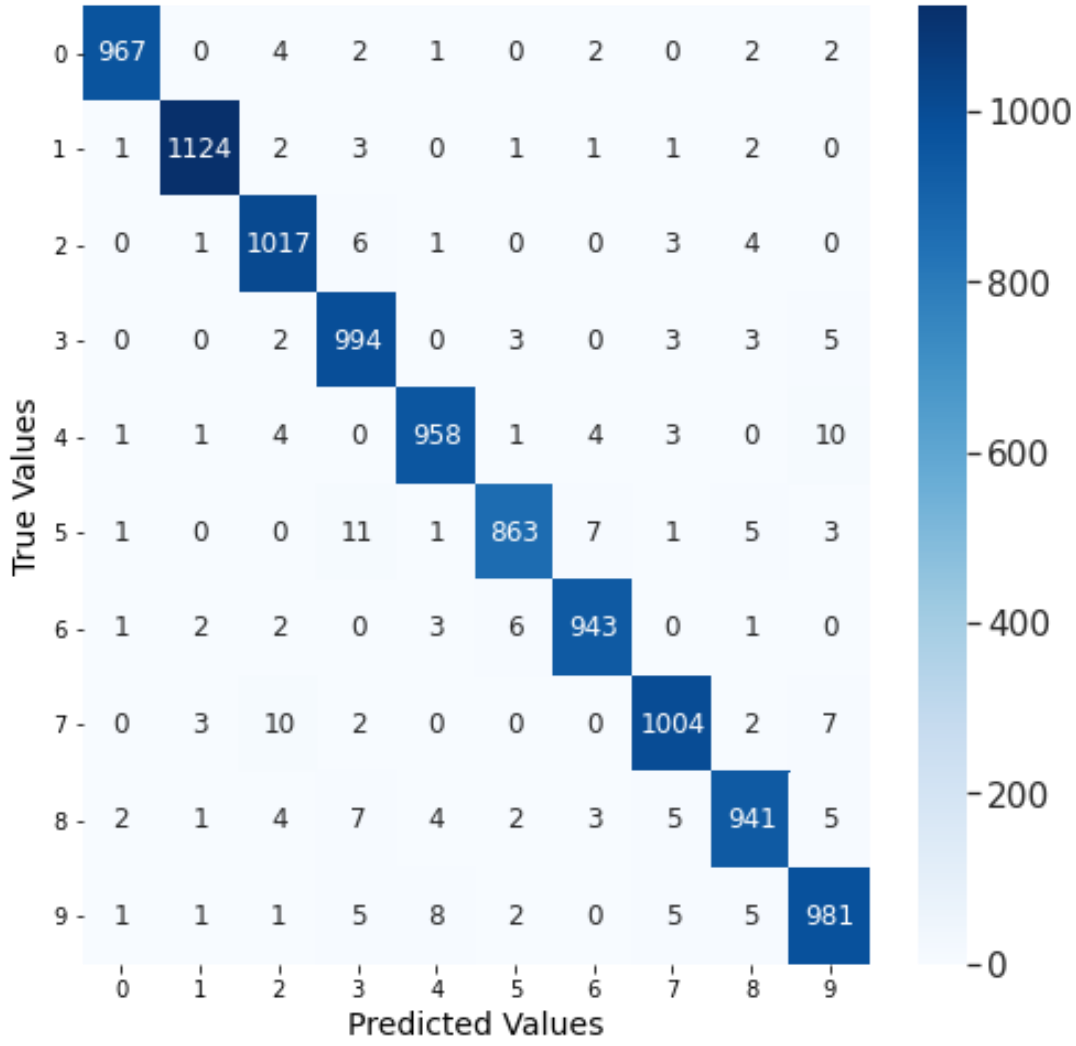
Then I train the model on MNIST dataset for 100 epochs. MNIST data set is normalized by dividing 255.

```
Epoch 1/100
1875/1875 [==============================] - 19s 3ms/step - loss: 0.4528 - accuracy: 0.8697 - val_loss: 0.1409 - val_accuracy: 0.9554
Epoch 2/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.1139 - accuracy: 0.9649 - val_loss: 0.1087 - val_accuracy: 0.9659
Epoch 3/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.0763 - accuracy: 0.9771 - val_loss: 0.0962 - val_accuracy: 0.9710
Epoch 4/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.0597 - accuracy: 0.9816 - val_loss: 0.0881 - val_accuracy: 0.9750
Epoch 5/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.0454 - accuracy: 0.9851 - val_loss: 0.0826 - val_accuracy: 0.9748
Epoch 6/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.0367 - accuracy: 0.9879 - val_loss: 0.0909 - val_accuracy: 0.9742
Epoch 7/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.0293 - accuracy: 0.9906 - val_loss: 0.0954 - val_accuracy: 0.9733
Epoch 8/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.0256 - accuracy: 0.9915 - val_loss: 0.0826 - val_accuracy: 0.9770
Epoch 9/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.0225 - accuracy: 0.9929 - val_loss: 0.0856 - val_accuracy: 0.9757
Epoch 10/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.0170 - accuracy: 0.9946 - val_loss: 0.0959 - val_accuracy: 0.9747
Epoch 11/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.0175 - accuracy: 0.9939 - val_loss: 0.0916 - val_accuracy: 0.9778
Epoch 12/100
1875/1875 [==============================] - 5s 2ms/step - loss: 0.0125 - accuracy: 0.9960 - val_loss: 0.1006 - val_accuracy: 0.9782
Epoch 13/100
```

```
Epoch 91/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.0034 - accuracy: 0.9990 - val_loss: 0.2887 - val_accuracy: 0.9779
Epoch 92/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.0039 - accuracy: 0.9988 - val_loss: 0.2998 - val_accuracy: 0.9777
Epoch 93/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.0069 - accuracy: 0.9986 - val_loss: 0.2731 - val_accuracy: 0.9793
Epoch 94/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.0026 - accuracy: 0.9992 - val_loss: 0.3140 - val_accuracy: 0.9770
Epoch 95/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.0053 - accuracy: 0.9988 - val_loss: 0.3275 - val_accuracy: 0.9757
Epoch 96/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.0032 - accuracy: 0.9991 - val_loss: 0.2883 - val_accuracy: 0.9785
Epoch 97/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.0035 - accuracy: 0.9991 - val_loss: 0.3001 - val_accuracy: 0.9773
Epoch 98/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.0053 - accuracy: 0.9985 - val_loss: 0.2598 - val_accuracy: 0.9792
Epoch 99/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.0023 - accuracy: 0.9994 - val_loss: 0.3479 - val_accuracy: 0.9753
Epoch 100/100
1875/1875 [==============================] - 5s 3ms/step - loss: 0.0042 - accuracy: 0.9991 - val_loss: 0.2907 - val_accuracy: 0.9792
<keras.callbacks.History at 0x7f5c6dbde8d0>
```

The accuracy after training is about 0.99.

This is confusion matrix of the original model after training.



Then I compressed the model by using SVD:

$$W^{(i)} \approx U'^{(i)}\left(V^{(i)}\right)^{T}, \text{where } U'^{(i)} = U^{(i)}\Sigma^{(i)}$$

and replace every layer with 2 Dense layers of U' and V.

2X Compression:

Firstly, I compressed model to a half of the original one.

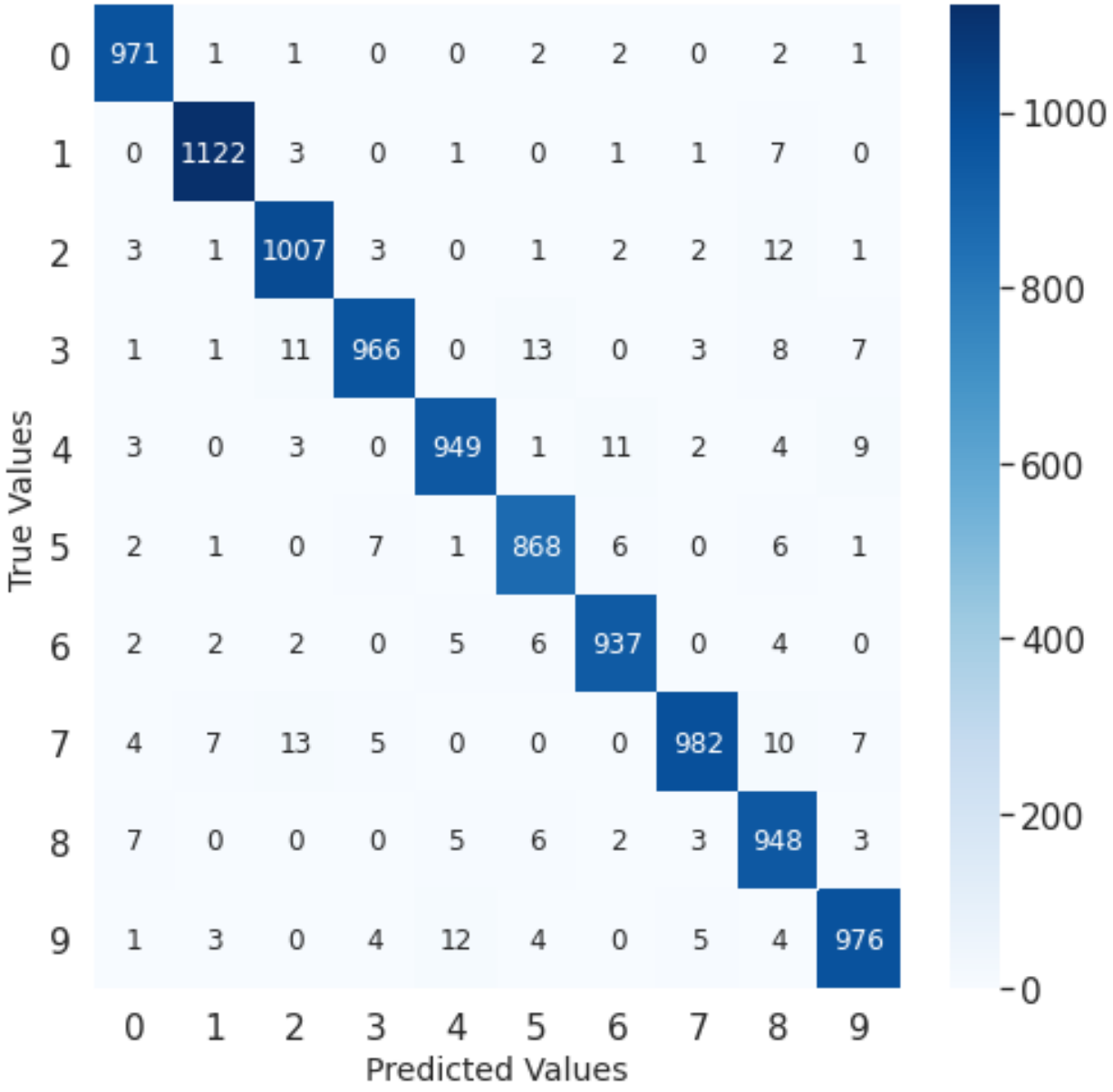
```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
flatten_1 (Flatten)          (None, 784)               0
_________________________________________________________________
dense_3 (Dense)              (None, 50)                39200
_________________________________________________________________
dense_4 (Dense)              (None, 100)               5100
_________________________________________________________________
dense_5 (Dense)              (None, 25)                2500
_________________________________________________________________
dense_6 (Dense)              (None, 50)                1300
_________________________________________________________________
dense_7 (Dense)              (None, 5)                 250
_________________________________________________________________
dense_8 (Dense)              (None, 10)                60
=================================================================
Total params: 48,410
Trainable params: 48,410
Non-trainable params: 0
```

We can see the number of parameters is decreased to a half of the original. Then I set the weights of the original after training to this model and I retrain the model for the refinement.

```
Epoch 1/10
1875/1875 [==============================] - 6s 3ms/step - loss: 0.2778 - accuracy: 0.9494 - val_loss: 0.2043 - val_accuracy: 0.9637
Epoch 2/10
1875/1875 [==============================] - 5s 2ms/step - loss: 0.0659 - accuracy: 0.9825 - val_loss: 0.1644 - val_accuracy: 0.9730
Epoch 3/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.0469 - accuracy: 0.9865 - val_loss: 0.1952 - val_accuracy: 0.9640
Epoch 4/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.0498 - accuracy: 0.9854 - val_loss: 0.1463 - val_accuracy: 0.9746
Epoch 5/10
1875/1875 [==============================] - 5s 2ms/step - loss: 0.0377 - accuracy: 0.9896 - val_loss: 0.1452 - val_accuracy: 0.9726
Epoch 6/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.0351 - accuracy: 0.9898 - val_loss: 0.1512 - val_accuracy: 0.9713
Epoch 7/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.0349 - accuracy: 0.9889 - val_loss: 0.1312 - val_accuracy: 0.9759
Epoch 8/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.0287 - accuracy: 0.9912 - val_loss: 0.1345 - val_accuracy: 0.9754
Epoch 9/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.0242 - accuracy: 0.9928 - val_loss: 0.1413 - val_accuracy: 0.9759
Epoch 10/10
1875/1875 [==============================] - 5s 3ms/step - loss: 0.0223 - accuracy: 0.9930 - val_loss: 0.1442 - val_accuracy: 0.9726
<keras.callbacks.History at 0x7f5c6ca6a090>
```

The accuracy of the compressed model is still 0.99.

This is confusion matrix after the training.



4X Compression:

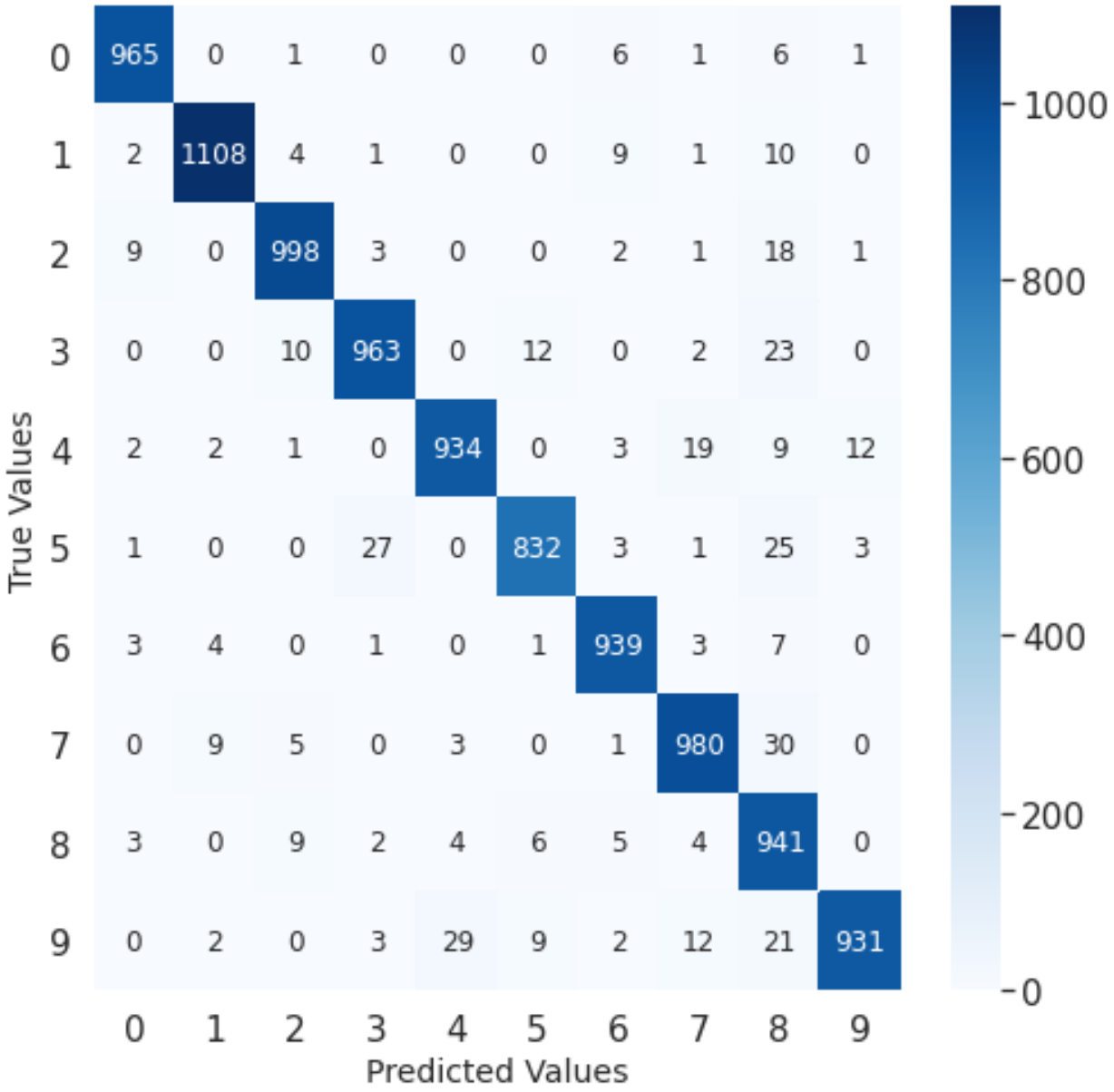
```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
flatten_2 (Flatten)          (None, 784)               0
_________________________________________________________________
dense_9 (Dense)              (None, 25)                19600
_________________________________________________________________
dense_10 (Dense)             (None, 100)               2600
_________________________________________________________________
dense_11 (Dense)             (None, 12)                1200
_________________________________________________________________
dense_12 (Dense)             (None, 50)                650
_________________________________________________________________
dense_13 (Dense)             (None, 2)                 100
_________________________________________________________________
dense_14 (Dense)             (None, 10)                30
=================================================================
Total params: 24,180
Trainable params: 24,180
Non-trainable params: 0
```

We can see the number of parameters is decreased to a quarter of the original. Then I set the weights of the original after training to this model and I retrain the model for the refinement.

```
Epoch 1/10
1875/1875 [==============================] - 5s 2ms/step - loss: 1.3722 - accuracy: 0.7526 - val_loss: 0.3158 - val_accuracy: 0.9396
Epoch 2/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.2246 - accuracy: 0.9475 - val_loss: 0.2806 - val_accuracy: 0.9438
Epoch 3/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.1721 - accuracy: 0.9594 - val_loss: 0.2761 - val_accuracy: 0.9514
Epoch 4/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.1430 - accuracy: 0.9661 - val_loss: 0.2389 - val_accuracy: 0.9605
Epoch 5/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.1206 - accuracy: 0.9698 - val_loss: 0.2357 - val_accuracy: 0.9607
Epoch 6/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.1126 - accuracy: 0.9720 - val_loss: 0.2443 - val_accuracy: 0.9607
Epoch 7/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.1027 - accuracy: 0.9729 - val_loss: 0.2387 - val_accuracy: 0.9610
Epoch 8/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.0998 - accuracy: 0.9758 - val_loss: 0.2448 - val_accuracy: 0.9622
Epoch 9/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.0810 - accuracy: 0.9780 - val_loss: 0.2629 - val_accuracy: 0.9672
Epoch 10/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.0802 - accuracy: 0.9791 - val_loss: 0.2328 - val_accuracy: 0.9591
<keras.callbacks.History at 0x7f5c6db27710>
```

The accuracy of the 4X compressed model is decreased a little bit to 0.97.

This is confusion matrix after the training.


True Values
Predicted Values
0 965 0 1 0 0 0 6 1 6 1
1 2 1108 4 1 0 0 9 1 10 0
2 9 0 998 3 0 0 2 1 18 1
3 0 0 10 963 0 12 0 2 23 0
4 2 2 1 0 934 0 3 19 9 12
5 1 0 0 27 0 832 3 1 25 3
6 3 4 0 1 0 1 939 3 7 0
7 0 9 5 0 3 0 1 980 30 0
8 3 0 9 2 4 6 5 4 941 0
9 0 2 0 3 29 9 2 12 21 931
0 1 2 3 4 5 6 7 8 9
1000
800
600
400
200
0

8X Compression:

```
Model: "sequential_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
flatten_4 (Flatten)          (None, 784)               0
_________________________________________________________________
dense_21 (Dense)             (None, 12)                9408
_________________________________________________________________
dense_22 (Dense)             (None, 100)               1300
_________________________________________________________________
dense_23 (Dense)             (None, 6)                 600
_________________________________________________________________
dense_24 (Dense)             (None, 50)                350
_________________________________________________________________
dense_25 (Dense)             (None, 1)                 50
_________________________________________________________________
dense_26 (Dense)             (None, 10)                20
=================================================================
Total params: 11,728
Trainable params: 11,728
Non-trainable params: 0
```

We can see the number of parameters is decreased to 1/8 of the original. Then I set the weights of the original after training to this model and I retrain the model for the refinement.

```
Epoch 1/10
1875/1875 [==============================] - 5s 2ms/step - loss: 2.8032 - accuracy: 0.2736 - val_loss: 1.2960 - val_accuracy: 0.5321
Epoch 2/10
1875/1875 [==============================] - 4s 2ms/step - loss: 1.2019 - accuracy: 0.5773 - val_loss: 1.1417 - val_accuracy: 0.7023
Epoch 3/10
1875/1875 [==============================] - 4s 2ms/step - loss: 1.0319 - accuracy: 0.7152 - val_loss: 1.0605 - val_accuracy: 0.7252
Epoch 4/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.9063 - accuracy: 0.7851 - val_loss: 1.0372 - val_accuracy: 0.8121
Epoch 5/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.8300 - accuracy: 0.8192 - val_loss: 0.9582 - val_accuracy: 0.7850
Epoch 6/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.7979 - accuracy: 0.8259 - val_loss: 0.8817 - val_accuracy: 0.8323
Epoch 7/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.7578 - accuracy: 0.8394 - val_loss: 0.8750 - val_accuracy: 0.8309
Epoch 8/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.7166 - accuracy: 0.8528 - val_loss: 0.8727 - val_accuracy: 0.8418
Epoch 9/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.6758 - accuracy: 0.8589 - val_loss: 0.9551 - val_accuracy: 0.8617
Epoch 10/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.6514 - accuracy: 0.8662 - val_loss: 0.8891 - val_accuracy: 0.8510
<keras.callbacks.History at 0x7f5c39931ad0>
```

The accuracy of the 8X compressed model is decreased to 0.88. So if we compressed intensely to the model. The accuracy is also decrease.

This is confusion matrix after the training: