Problem 1

My accuracy was:

Training: 85.09%

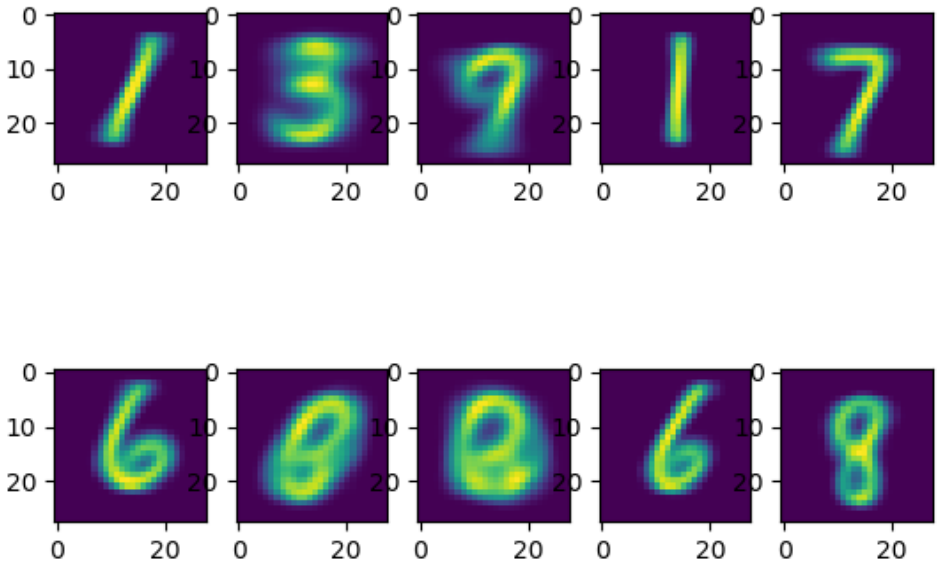
Testing: 85.06%

The result accuracy just reach the standart(85% - 90%). I guess one way to improve accuracy would be using float64 instead of float32 which is able to handle more precision but consumes more memory. Also, using a smaller eps may increase the accuracy. During the implementation my *bij* matrix was the transpose of what it supposed to be, so I changed some code in *plot_parameters()* function to make it work properly.

Furthermore, I found that in this assignment if we only use for-loops to do the matrix calculation the execution time can be unbearably long. So I tried to make everything calculated in matrix form.

Problem 2

Figure 1: EM parameters when k=5



Problem 3

	K=1	K=5	K=10	K=20
train_acc	80.80%	85.09%	88.61%	92.60%
test_acc	80.54%	85.06%	88.65%	92.82%

When K value is small, we could observe that the plot image looks like all the odd or even number blurred together just as mentioned in the handout. And when K value increases, the model has more latent parameters, which means it is capable of finding more patterns for a specific class. And the image begins to show individual even or odd numbers which means the model has made the connection between its hidden parameter to the shape of specific numbers.

One thing need to be noted is that when $k = 1$, initiating $Q = 1/k$ would cause the break instantly because the calculated Q values are 1.0 since there's only one hidden parameter.

Figure 2: EM parameters for different K values