1. a)

$$p(y=k|x,\mu,\sigma) = \frac{p(x|y=k,\mu,\sigma)p(y=k,\mu,\sigma)}{p(x,\mu,\sigma)} \quad \text{\# Apply Bayes' rule}$$

$$= \frac{p(x|y=k,\mu,\sigma)p(y=k,\mu,\sigma)}{p(x|\mu,\sigma)p(\mu,\sigma)}$$

$$= \frac{p(x|y=k,\mu,\sigma)p(y=k|\mu,\sigma)p(\mu,\sigma)}{p(x|\mu,\sigma)p(\mu,\sigma)}$$

$$= \frac{p(x|y=k,\mu,\sigma)p(y=k|\mu,\sigma)}{p(x|\mu,\sigma)}$$

$$= \frac{p(x|y=k,\mu,\sigma)p(y=k)}{p(x|\mu,\sigma)} \quad \text{\# Because y is independent of } \mu \text{ and } \sigma$$

$$= \frac{p(x|y=k,\mu,\sigma)p(y=k)}{\sum_i^k p(x|y=i,\mu,\sigma)p(y=i)} \quad \text{\# By the law of total probability}$$

$$= \frac{(\prod_{i=1}^{D} 2\pi\sigma_i^2)^{1/2} \; exp\left\{-\sum_{i=1}^{D} \frac{1}{2\sigma_i^2}(x_i-\mu_{ki})^2\right\}\alpha_k}{\sum_j^k (\prod_{i=1}^{D} 2\pi\sigma_i^2)^{1/2} \; exp\left\{-\sum_{i=1}^{D} \frac{1}{2\sigma_i^2}(x_i-\mu_{ji})^2\right\}\alpha_j} \quad \text{\# Substitute values in the function}$$

b)

$$\ell(\alpha,\mu,\sigma|D) = -\log p(y^{(1)}, x^{(1)}, \ldots, y^{(N)}, x^{(N)}|\, \alpha, \mu, \sigma)$$

$$= -\log \prod_{i=1}^{N} p(y^{(i)}, x^{(i)}|\, \alpha, \mu, \sigma) \quad \text{\# By the independence of data}$$

$$= -\log \prod_{i=1}^{N} p(x^{(i)}|y^{(i)}, \alpha, \mu, \sigma)p(y^{(i)}, \alpha, \mu, \sigma)$$

$$= -\log \prod_{i=1}^{N} p(x^{(i)}|y^{(i)}, \alpha, \mu, \sigma)p(y^{(i)}) \quad \text{\# Because y is only dependent of x}$$

$$= -\sum_{i=1}^{N} (\log p(x^{(i)}|y^{(i)}, \alpha, \mu, \sigma) + \log p(y^{(i)}))$$

$$= -\sum_{i=1}^{N} (\log ((\prod_{j=1}^{D} 2\pi\sigma_j^2)^{1/2} \; exp\left\{-\sum_{k=1}^{D} [y^{(i)}=k]\frac{1}{2\sigma_j^2}(x_{ki}-\mu_{ik})^2\right\}) +$$

$$\log \sum_{k=1}^{K} \mathbf{I}[y^{(i)}=k]\alpha^k$$

$$= -\sum_{i=1}^{N} (\tfrac{1}{2} \sum_{j=1}^{D} (\log 2\pi + \log \sigma_j^2) - \sum_{k=1}^{D} [y^{(i)}=k]\frac{1}{2\sigma_j^2}(x_{ik}-\mu_{ik})^2) - \log \sum_{k=1}^{K} \mathbf{I}[y^{(i)}=k]\alpha^k$$

c)

$$\frac{\partial \ell}{\partial \mu_{ik}} = \frac{\partial}{\partial \mu_{ik}} -\sum_{i=1}^{N} (\tfrac{1}{2} \sum_{j=1}^{D} (\log 2\pi + \log \sigma_j^2) - \sum_{k=1}^{D} [y^{(i)}=k]\frac{1}{2\sigma_j^2}(x_{ik}-\mu_{ik})^2) - \log \sum_{k=1}^{K} \mathbf{I}[y^{(i)}=k]\alpha^k$$

$$= -\sum_{i=1}^{N} (0 - \frac{\partial}{\partial \mu_{kj}} \sum_{k=1}^{D} [y^{(i)}=k]\frac{1}{2\sigma_j^2}(x_i-\mu_{ik})^2) - 0$$

$$= -\frac{1}{\sigma_i^2} \sum_{i=1}^{N} ([y^{(i)}=k](x_{ik}-\mu_{ik}))$$

$$\frac{\partial \ell}{\partial \sigma_j^2} = \frac{\partial \ell}{\partial \sigma_j^2} -\sum_{i=1}^{N} (\tfrac{1}{2} \sum_{j=1}^{D} (\log 2\pi + \log \sigma_{j\,j}^2) - \sum_{k=1}^{D} [y^{(i)}=k]\frac{1}{2\sigma_j^2}(x_{ik}-\mu_{ik})^2) - \log \sum_{k=1}^{K} \mathbf{I}[y^{(i)}=k]\alpha^k$$

$$= -\sum_{i=1}^{N} (\tfrac{1}{2} \sum_{j=1}^{D} (\frac{\partial \ell}{\partial \sigma_j} \log 2\pi + \frac{\partial \ell}{\partial \sigma_j} \log \sigma_j^2) - \frac{\partial \ell}{\partial \sigma_j} \sum_{k=1}^{D} [y^{(i)}=k]\frac{1}{2\sigma_k^2}(x_{ik}-\mu_{ik})^2) - 0$$

$$= -\sum_{i=1}^{N}(½\sum_{j=1}^{D}(\mathbf{0} + \frac{\partial \ell}{\partial \sigma_j}\log\sigma_j^2) - \frac{\partial \ell}{\partial \sigma_j}\sum_{k=1}^{D}[y^{(i)}=k]\frac{1}{2\sigma_j^2}(x_{ik}-\mu_{ik})^2) - 0$$

$$= -\frac{N}{2\sigma_j^2} + \sum_{i=1}^{N}\frac{1}{2\sigma_j^4}(x_{ik}-\mu_{ik})^2$$

d)

Setting $\frac{\partial \ell}{\partial \mu_{ik}} = 0 \Rightarrow -\frac{1}{\sigma_i^2}\sum_{i=1}^{N}([y^{(i)}=k](x_{ik}-\mu_{ik})) = 0$

$$\Rightarrow \sum_{i=1}^{N}([y^{(i)}=k](x_{ik}-\mu_{ik})) = 0$$

$$\Rightarrow \sum_{i=1}^{N}([y^{(i)}=k]x_{ik}) = \sum_{i=1}^{N}([y^{(i)}=k]\mu_{ik})$$

$$\Rightarrow \mu_{ik} = \frac{\sum_{i=1}^{N}([y^{(i)}=k]x_{ik})}{\sum_{i=1}^{N}([y^{(i)}=k])}$$

Setting $\frac{\partial \ell}{\partial \sigma_j^2} = 0 \Rightarrow -\frac{N}{2\sigma_j^2} + \sum_{i=1}^{N}\frac{1}{2\sigma_j^4}(x_{ik}-\mu_{ik})^2 = 0$

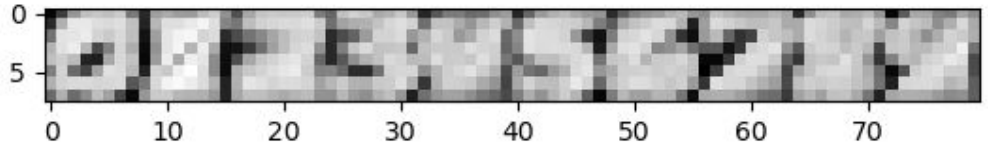
$$\Rightarrow \sum_{i=1}^{N}\frac{1}{2\sigma_j^4}(x_{ik}-\mu_{ik})^2 = \frac{N}{2\sigma_j^2}$$

$$\Rightarrow \sum_{i=1}^{N}(x_{ik}-\mu_{ik})^2 = N\sigma_j^2$$

$$\Rightarrow \sigma_j^2 = \frac{1}{N}\sum_{i=1}^{N}(x_{ik}-\mu_{ik})^2$$

2.1

a) The image of the log of the diagonal elements of each covariance matrix:



b) The conditional log-likelihood on test set:

```
The average conditional log-likelihood on test set is: -0.19667320325525822
```

On training set:

```
The average conditional log-likelihood on train set is: -0.12462443666863306
```

c) Training and test accuracy is as the following:

```
The train accuracy is: 0.9814285714285714
The test accuracy is: 0.97275
```

2.2

a)

The training set and test set is binarized using this line of code:

```
train_data, test_data = binarize_data(train_data), binarize_data(test_data)
```

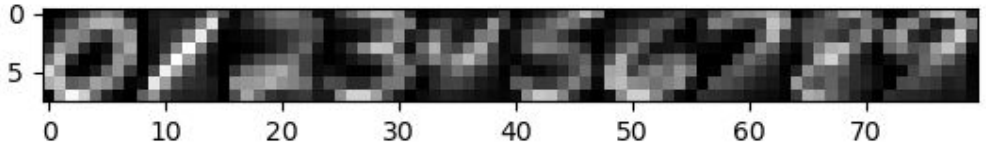
b) The parameters are fitted using the function as below:

```
def compute_parameters(train_data, train_labels):
    '''
    Compute the eta MAP estimate/MLE with augmented data

    You should return a numpy array of shape (10, 64)
    where the ith row corresponds to the ith digit class.
    '''
    eta = np.ones((10, 64))
    for i in range(train_data.shape[0]):
        corresponding_label = int(train_labels[i])
        eta[corresponding_label, :] += train_data[i, :]
    return eta / (0.1 * train_data.shape[0] + 2)
```

Since we are adding two special training cases before adding other training data, the **η** matrix is initialized to have one on every entry. After that, we count the occurrence of every pixel for each class and divide the number by the total number of samples for each class to obtain a probability for one pixel to appear in a particular class.

c) The graphs for $\boldsymbol{\eta}_k$ vectors are as the following:



d) The graphs for generated samples are as the following:

e) The average conditional log-likelihood on the test set:

```
The average conditional log-likelihood on test set is: -0.9872704337253583
```

On the training set:

```
The average conditional log-likelihood on train set is: -0.9437538618002557
```

f) Training and test accuracy is as the following:

```
The train accuracy is: 0.7741428571428571
The test accuracy is: 0.76425
```

2.3

The Conditional Gaussian Classifier outperformed the Naive Bayes Classifier by having a test accuracy of 97% while Naive Bayes Classifier only has 76% accuracy on the test set. This matches my expectation because the Naive Bayes model assumes the conditional independence relationship between all pixels. However, the pixels are somehow correlated to each other, so the Naive Bayes Classifier loses a lot of information and hence loses accuracy by making such an assumption.