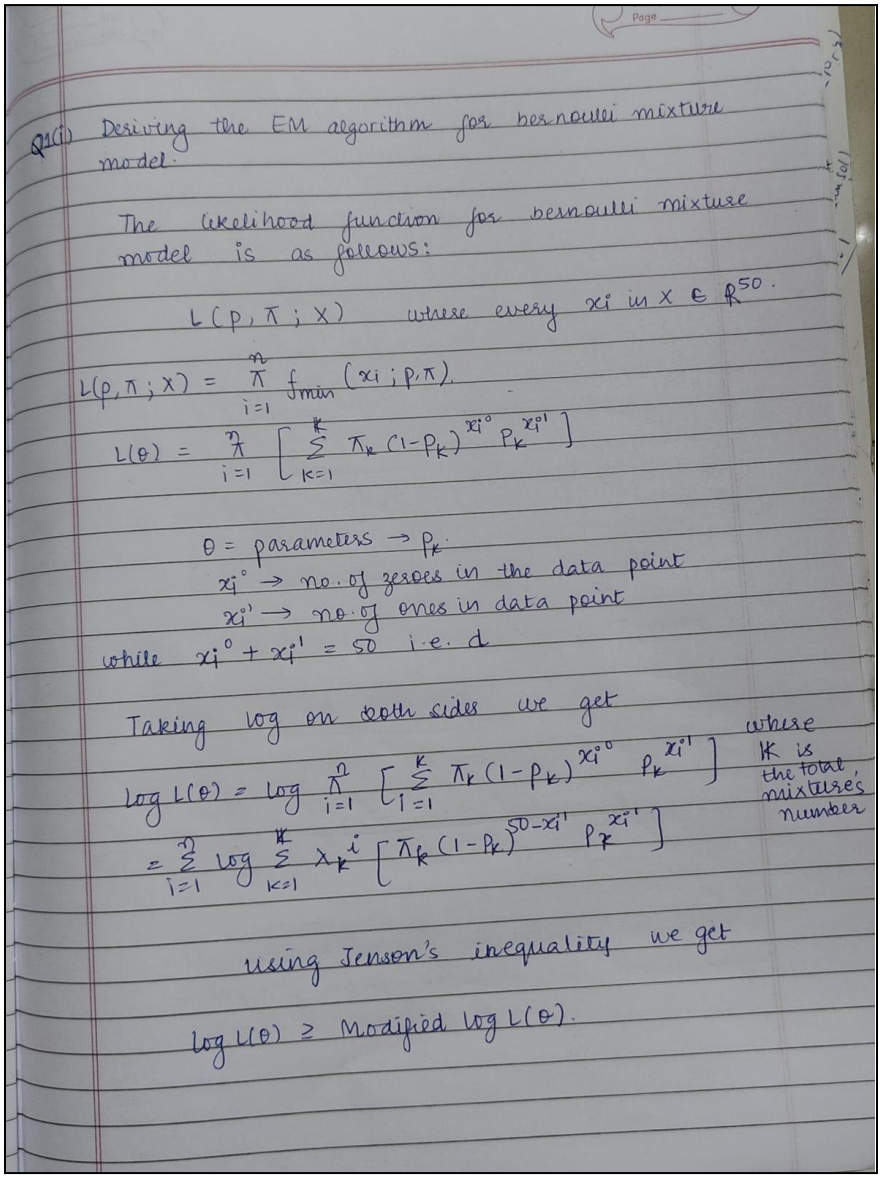
Q1 : You are given a data-set with 400 data points in $\{0, 1\}^{50}$ generated from a mixture of some distribution in the file A2Q1.csv. (Hint: Each datapoint is a flattened version of a $\{0, 1\}^{10\times5}$ matrix.)

(i) Determine which probabilistic mixture could have generated this data (It is not a Gaussian mixture). Derive the EM algorithm for your choice of mixture and show your calculations. Write a piece of code to implement the algorithm you derived by setting the number of mixtures K = 4. Plot the log-likelihood (averaged over 100 random initializations) as a function of iterations.

This data could have been generated by the Bernoulli Mixture.

The derivations of EM Algorithm for Bernoulli Mixture model are as follows:



Q1(i) Deriving the EM algorithm for bernoulli mixture model.

The likelihood function for bernoulli mixture model is as follows:

$L(p, \pi; X)$ where every x_i in $X \in R^{50}$.

$$L(p,\pi;X) = \prod_{i=1}^{n} f_{mix}(x_i; p, \pi)$$

$$L(\theta) = \prod_{i=1}^{n} \left[\sum_{k=1}^{K} \pi_k (1-p_k)^{x_i^0} p_k^{x_i^1} \right]$$

θ = parameters $\rightarrow p_k$

$x_i^0 \rightarrow$ no. of zeroes in the data point

$x_i^1 \rightarrow$ no. of ones in data point

while $x_i^0 + x_i^1 = 50$ i.e. d

Taking log on both sides we get

$$\log L(\theta) = \log \prod_{i=1}^{n} \left[\sum_{k=1}^{K} \pi_k (1-p_k)^{x_i^0} p_k^{x_i^1} \right]$$

where K is the total mixtures number

$$= \sum_{i=1}^{n} \log \sum_{k=1}^{K} \lambda_k^i \left[\pi_k (1-p_k)^{50-x_i^1} p_k^{x_i^1} \right]$$

using Jenson's inequality we get

$\log L(\theta) \geq$ Modified $\log L(\theta)$.

$$\sum_{i=1}^{n}\sum_{k=1}^{K} \lambda_k^i \log\left[\frac{\pi_k (1-p_k)^{50-x_i^j} p_k^{x_i^j}}{\lambda_k^i}\right]$$

To find λ, we maximize modified log likelihood over θ.

$$= \max_{\theta} \sum_{i=1}^{n}\sum_{k=1}^{K} \lambda_k \left[\log \pi_k + (50-x_i)\log(1-p_k) + x_i \log p_k\right]$$

$$= \max_{\theta} \sum_{i=1}^{n}\sum_{k=1}^{K} \lambda_k^i \left[\log \pi_k + (50-x_i)\log(1-p_k) + x_i^j \log p_k\right]$$

Taking Derivative w.r.t. p.

$$\sum_{i=1}^{n}\sum_{k=1}^{K} \lambda_k^i \cdot 0 + \frac{\lambda_k^i (50-x_i^j)}{(1-p_k)} + \frac{\lambda_k^i x_i^j}{-p_k} = 0,$$

for some k

$$\sum_{i=1}^{n} \frac{\lambda_k^i (50-x_i^j)}{1-p_k} - \frac{\lambda_k^i x_i^j}{p_k} = 0$$

$$\sum_{i=1}^{n} \lambda_k^i p_k 50 - \lambda_k^i x_i^j = 0$$

$$\therefore \boxed{p_k = \frac{1}{50} \frac{\sum_{i=1}^{n} \lambda_k^i x_i^j}{\sum_{i=1}^{n} \lambda_k^i}}$$

Taking differentiating w.r.t. π,

$$\sum_{i=1}^{n} \sum_{k=1}^{K} \frac{\lambda_k{}^i}{\pi_k} = 0$$

for some k.

$$\sum_{i=1}^{n} \frac{\lambda_k{}^i}{\pi_k} = 0$$

$$n\pi_k = \sum_{i=1}^{n} \lambda_k{}^i$$

$$\boxed{\pi_k = \sum_{i=1}^{n} \frac{\lambda_k{}^i}{n}}$$

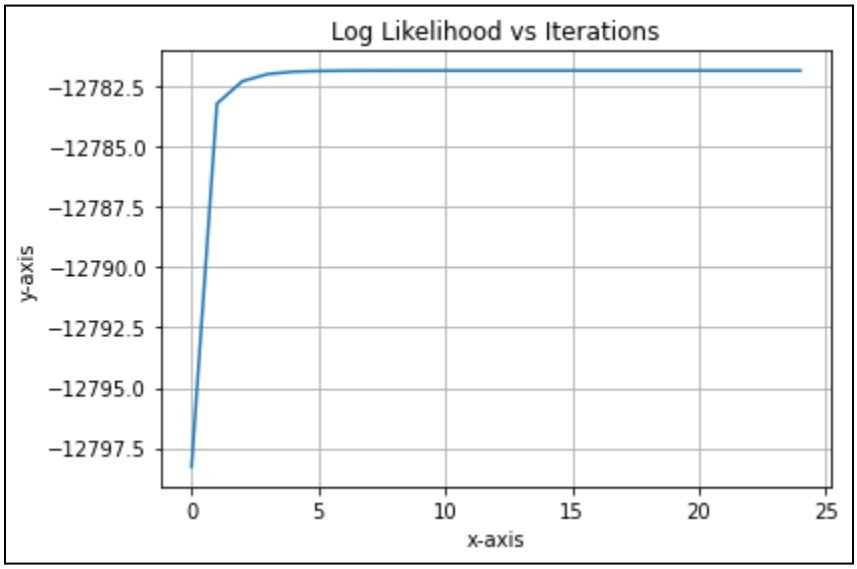
$\lambda_k{}^i$ can be desired from Baye's Theorem

$$\lambda_k{}^i = \frac{P(k)\ P(x_i/k)}{P(x_i)}$$

$$\lambda_k{}^i = \frac{\pi_k (p_k)^{x_i^1} (1-p_k)^{50-x_i^1}}{\sum_{k=1}^{K} \pi_k (p_k)^{x_i^1} (1-p_k)^{50-x_i^1}}$$

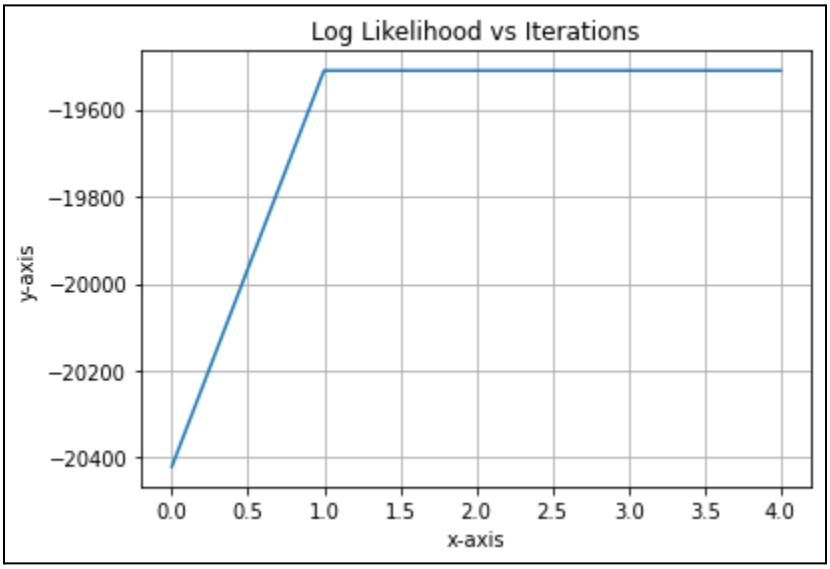
Here we can clearly see that $\lambda_k{}^i$ is the probability of x_i belonging to kth mixture.

Figure : Log-likelihood (averaged over 100 random initializations) vs Iterations Graph



(ii) Assume that the same data was in fact generated from a mixture of Gaussians with 4 mixtures. Implement the EM algorithm and plot the log-likelihood (averaged over 100 random initializations of the parameters) as a function of iterations. How does the plot compare with the plot from part (i)? Provide insights that you draw from this experiment.

Figure : Log-likelihood(averaged over 100 random initializations) vs Iterations Graph

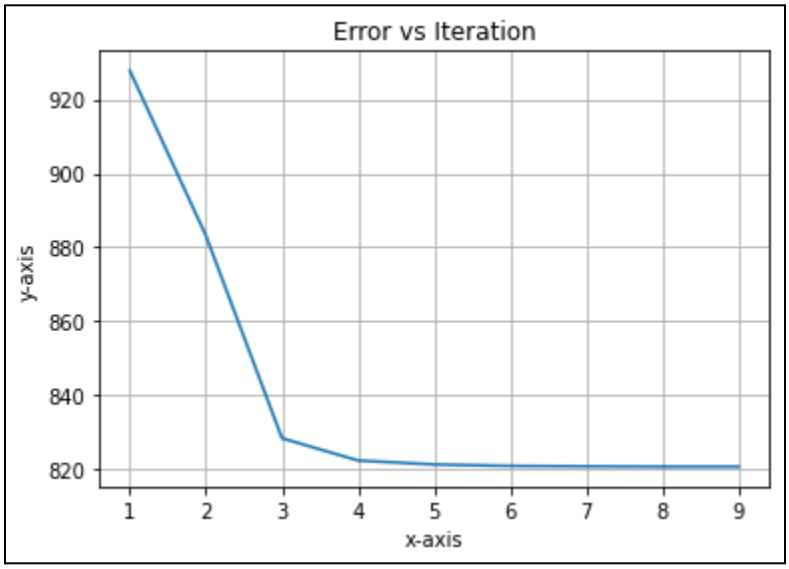


In the Log-likelihood(averaged over 100 random initializations) vs Iterations Graph for assumption of data generated from Gaussian Mixtures we see that the values of log-likelihood are lesser than the values of the Log-likelihood(averaged over 100 random initializations) vs Iterations Graph for assumption of data generated from Bernoulli Mixtures. Also the

maximization of Log-likelihood(averaged over 100 random initializations) for Bernoulli Mixture assumption is higher than that for Gaussian Mixtures assumption. Hence we can infer that the assumption that data is generated from Bernoulli Mixtures is more accurate.

(iii) Run the K-means algorithm with K = 4 on the same data. Plot the objective of K − means as a function of iterations.

Figure : Objective of K-means vs Iterations



(iv) Among the three different algorithms implemented above, which do you think you would choose for this dataset and why?

For the three different algorithms implemented above, the one that is best for this data set is the first one i.e. the EM Algorithm for Bernoulli Mixture.

In the Log-likelihood(averaged over 100 random initializations) vs Iterations Graph for assumption of data generated from Gaussian Mixtures we see that the values of log-likelihood are lesser than the values of the Log-likelihood(averaged over 100 random initializations) vs Iterations Graph for assumption of data generated from Bernoulli Mixtures. Also the maximization of Log-likelihood(averaged over 100 random initializations) for Bernoulli Mixture assumption is higher than that for Gaussian Mixtures assumption. Hence we can infer that the assumption that data is generated from Bernoulli Mixtures is more accurate.

We know that if EM is performed it performs soft clustering, that is it assigns the likelihood of that data point to be in those clusters. While K-means clustering performs hard clustering i.e. for each data point it either belongs to a cluster completely or not. So EM Algorithm is preferred over K-means clustering. Due to better likelihood of Bernoulli Mixtures EM Algorithm for it will be preferred for this data set.

Q2 : You are given a data-set in the file A2Q2 Data train.csv with 10000 points in (R^{100}, R) (Each row corresponds to a datapoint where the first 100 components are features and the last component is the associated y value).

(i) Obtain the least squares solutions w_{ML} to the regression problem using the analytical solution.

The least squares solution w_{ML} to the regression problem using the analytical solution is :

[[-7.84961009e-03]
 [-1.36715320e-02]
 [-3.61656438e-03]
 [2.64909160e-03]
 [1.88551446e-01]
 [2.65314657e-03]
 [9.46531786e-03]
 [1.79809481e-01]
 [3.73757317e-03]
 [4.99608944e-01]
 [8.35836265e-03]
 [4.29108775e-03]
 [1.42141179e-02]
 [3.94232414e-03]
 [9.36795890e-03]
 [-1.12038274e-03]
 [3.35727500e-03]
 [1.16152212e-03]
 [-9.40884707e-03]
 [-2.45575476e-03]
 [-1.17409629e-02]
 [-1.01960612e-02]
 [7.95771321e-03]
 [-1.00574854e-02]
 [6.04882939e-03]
 [-4.67345192e-03]
 [-3.09091547e-03]
 [8.14909193e-03]
 [1.20264599e-02]
 [-6.82458163e-03]
 [-8.65405539e-03]
 [9.86273479e-04]
 [4.92968011e-03]
 [5.99772461e-03]
 [-1.34667860e-02]

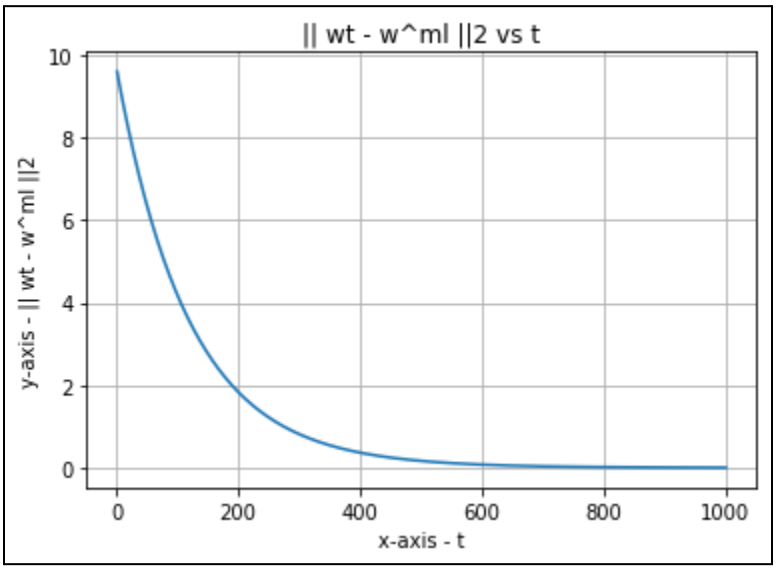
[1.07075729e-03]
[1.32745992e-02]
[-1.14148742e-02]
[-2.01056697e-02]
[5.85096240e-01]
[4.94483247e-04]
[-7.86666920e-04]
[-2.71926574e-03]
[-9.54021938e-03]
[-5.44161058e-03]
[9.80679209e-03]
[-6.72540624e-03]
[-4.45414276e-04]
[6.98516508e-03]
[3.16138907e-02]
[4.51763485e-01]
[-8.75221380e-03]
[2.55167390e-03]
[4.24921150e-03]
[2.89847927e-01]
[7.03723255e-03]
[-1.95796946e-03]
[1.41523883e-02]
[-1.06508170e-02]
[7.72743903e-01]
[-5.67126044e-03]
[-6.30026188e-04]
[6.50943015e-03]
[-4.84019165e-03]
[4.63832329e-03]
[4.54887177e-03]
[-2.99475114e-03]
[8.38781696e-03]
[-2.47558716e-03]
[9.00947922e-04]
[1.14713514e-03]
[-1.87641345e-03]
[-1.05175760e-02]
[-9.31304110e-03]
[-1.23550002e-03]
[5.97797559e-01]
[-4.78625013e-03]
[-1.13727852e-02]
[2.88477060e-03]

```
 [ 8.48999776e-01]
[-1.08924235e-02]
[ 2.26346489e-03]
[-1.38099800e-03]
[-6.35934691e-03]
[ 5.83784109e-03]
[ 5.69286755e-03]
[ 5.35566859e-03]
[-8.20616315e-03]
[ 1.29884015e-02]
[-2.30575631e-03]
[-1.22263765e-04]
[ 8.66629171e-03]
[-4.29446300e-03]
[ 5.69510898e-03]
[ 7.55483353e-03]
[-9.43540843e-03]
[ 1.82905446e-02]
[-1.16998887e-03]
[-2.61599136e-03]
[-8.58616114e-03]]
```

In this the accuracy that is obtained on test data 73.03651071046087 while the mean squared error on the test data set is 185.36365558489376

(ii) Code the gradient descent algorithm with suitable step size to solve the least squares algorithms and plot $\| w_t - w_{ML} \|_2$ as a function of t. What do you observe?

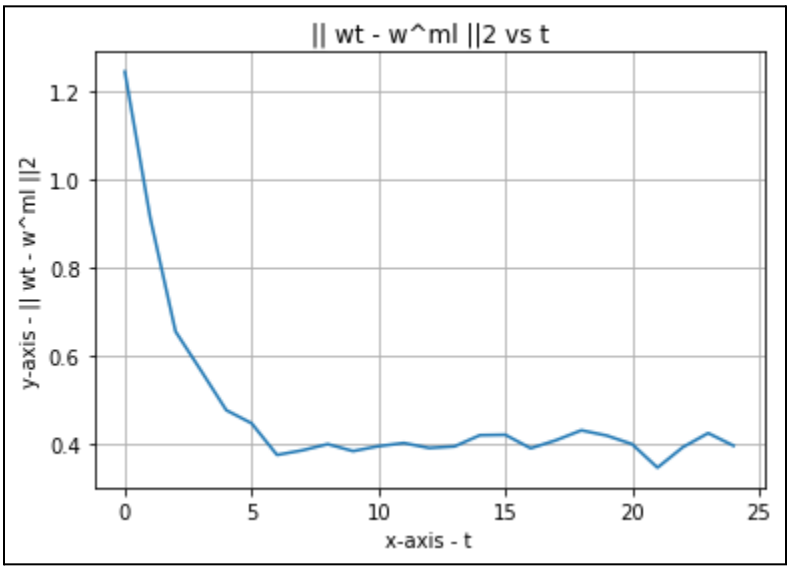
Figure : $\| w_t - w_{ML} \|_2$ vs t



As can be observed from the above plot for every w we obtain in t th iteration when we take the L2 norm of w - w^ml we see a decrease as t increases. This is because with every iteration the w nears to the w^ml and finally becomes equal to it and we can see the algorithm converge at this point. (Step size : 0.0001, Iterations : 1000)

(iii) Code the stochastic gradient descent algorithm using batch size of 100 and plot $\| w_t - w_{ML} \|_2$ as a function of t. What are your observations?

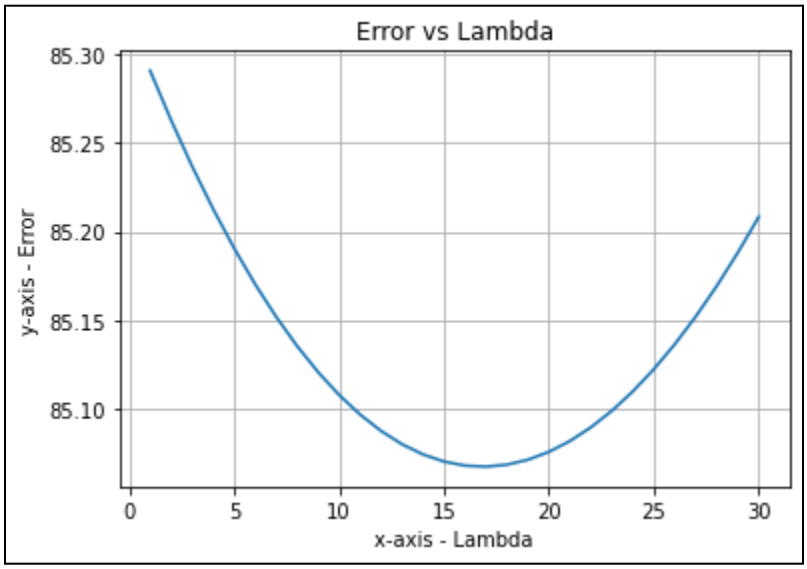
Figure : $\| w_t - w_{ML} \|_2$ vs t



As can be observed from the above plot for every w we obtain in t th iteration of Stochastic Gradient Descent for a random batch of 100 X 100 data when we take the L2 norm of w - w^ml we see a decrease as t increases. This is because with every iteration the w nears to the w^ml and finally becomes equal to it and we can see the algorithm converge at this point.

(iv) Code the gradient descent algorithm for ridge regression. Cross-validate for various choices of λ and plot the error in the validation set as a function of λ. For the best λ chosen, obtain w_R. Compare the test error (for the test data in the file A2Q2Data test.csv) of w_R with w_{ML}. Which is better and why?

Figure : Error vs lambda



Performed the cross validation through splitting data into 80:20 ratio. First picked up the 80% data, found converged wt for it through gradient descent for ridge regression (fw = 2*(X*Xt)*wt - 2*(X*y) - 2*(Lambda*wt))and then found MSE for the rest 20% data using the w obtained. This is repeated for values of lambda from 1 to 30. The minimum MSE is obtained at lambda = 17.

The test error corresponding to wr = 181.5013955124184
The test error corresponding to w^ml =185.36365558489373

wr is better because it reduces MSE to some extent than w^ml

wr is as follows :
[[-5.54420245e-03]
 [-1.38248406e-02]
 [-2.59454510e-03]
 [3.69947147e-03]
 [1.89163875e-01]
 [8.20506671e-03]

[1.03922467e-02]
[1.77567680e-01]
[6.99741346e-03]
[4.90174781e-01]
[4.83112018e-03]
[6.19982086e-03]
[1.10212195e-02]
[6.70258273e-04]
[1.69612239e-02]
[-3.64190943e-04]
[2.72316404e-03]
[5.01278107e-03]
[-1.16231911e-02]
[-5.54362867e-03]
[-1.06752442e-02]
[-1.48918907e-02]
[9.19926943e-03]
[-7.56471845e-03]
[4.37322829e-03]
[-1.16460443e-02]
[-2.00086188e-04]
[1.27201309e-02]
[1.12226432e-02]
[-4.61284195e-03]
[-4.28697323e-03]
[7.62886033e-03]
[6.35873085e-03]
[8.82953653e-03]
[-1.18814519e-02]
[-2.31175817e-03]
[1.79016564e-02]
[-1.24584828e-02]
[-1.88176782e-02]
[5.73313401e-01]
[-1.49973666e-03]
[-3.33919516e-03]
[-4.84895767e-04]
[-1.07196577e-02]
[-6.47449626e-03]
[8.83267928e-03]
[-3.91989710e-03]
[-5.78406250e-03]
[4.79513439e-03]
[3.04431118e-02]

[4.40972380e-01]
[-3.20152810e-03]
[5.76870274e-03]
[4.37440476e-03]
[2.91939830e-01]
[1.26442998e-02]
[-6.81622487e-04]
[1.66344703e-02]
[-5.08379742e-03]
[7.50257795e-01]
[-4.96408581e-03]
[1.45617279e-03]
[8.61651568e-03]
[-1.75015814e-03]
[3.64750884e-03]
[7.31182942e-03]
[-2.55332551e-03]
[1.03516790e-03]
[-4.68241899e-04]
[-2.85567114e-03]
[2.94621868e-03]
[1.05413579e-04]
[-1.04733777e-02]
[-8.39819160e-03]
[5.31366569e-04]
[5.87223583e-01]
[-3.96062676e-03]
[-1.35288044e-02]
[2.96695402e-03]
[8.32116336e-01]
[-6.26098766e-03]
[6.55882058e-03]
[-1.18107027e-03]
[-6.40277738e-03]
[9.71148882e-03]
[4.67433665e-03]
[7.98178737e-03]
[-1.10749663e-02]
[1.40685202e-02]
[8.15139223e-03]
[2.12027179e-03]
[1.24044218e-02]
[-6.81263456e-03]
[1.12369562e-02]

```
[ 4.31296439e-03]
[-1.07035807e-02]
[ 2.10393423e-02]
[-4.29343024e-03]
[-4.04894329e-04]
[-6.31101779e-03]]
```