CSC411 Assignment 2

Tianrui Xiao, 999018049

November 16, 2015

1 EM for Mixture of Gaussians

For the case of a Gaussian mixture model with shared covariance matrix across the Gaussians, we have the same likelihood function as with the general MoG model.

$\mathcal{L}_{log\_loss} = \ln p(\mathbf{X}|\pi, \mu, \mathbf{\Sigma})$

X is an N by D matrix with the input vectors in each row.
Then the derivatives of the log likelihood function at the maximum would be 0 w.r.t. model parameters. We know the following assumptions and matrix identities:

$\mathbf{\Sigma}^{\mathsf{T}} = \mathbf{\Sigma}$
$\sum_{k=1}^{K} \pi_k = 1$
$\frac{d\mathbf{a}^{\mathsf{T}}\mathbf{X}\mathbf{b}}{d\mathbf{X}} = \mathbf{a}\mathbf{b}^{\mathsf{T}}$
$\frac{d|\mathbf{X}|}{d\mathbf{X}} = |\mathbf{X}|(\mathbf{X}^{-1})^{\mathsf{T}}$

We can observe that it is easier to take derivative with the inverse of the covariance matrix.

$$\frac{\partial \mathcal{L}}{\partial \mu_k} = -\sum_{n=1}^{N} \frac{\pi_k \mathcal{N}(\mathbf{x_n}|\mu_\mathbf{k}, \mathbf{\Sigma})}{\sum_{j}^{K} \pi_j \mathcal{N}(\mathbf{x_n}|\mu_\mathbf{k}, \mathbf{\Sigma})}$$

$$\frac{\partial \mathcal{L}}{\partial \Sigma^{-1}} = \sum_{n=1}^{N} \frac{\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x_n}|\mu_\mathbf{k}, \mathbf{\Sigma}) \frac{1}{2}(\Sigma^{\mathsf{T}} - (x_n - \mu_k)(x_n - \mu_k)^{\mathsf{T}})}{\sum_{j}^{K} \pi_j \mathcal{N}(\mathbf{x_n}|\mu_\mathbf{k}, \mathbf{\Sigma})}$$

Optimizing w.r.t. $\Sigma^{-}1$ is equivalent to optimizing directly with Σ^{-1} due to the fact that in general

$\frac{\partial F}{\partial \Sigma^{-1}} = \frac{\partial F}{\partial \Sigma}\frac{\partial \Sigma^{-}1}{\partial \Sigma} = \frac{\partial F}{\partial \Sigma}(-\Sigma^{\mathsf{T}}\Sigma^{\mathsf{T}})$

Setting derivatives w.r.t. parameters to zero, we can obtain the following:

$$\boxed{\mu_{kML} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n}$$

$$\boxed{\Sigma_{ML} = \frac{\sum_{n=1}^{N} \sum_{k=1}^{K} (x_n - \mu_k)(x_n -_k)^\mathsf{T}}{\sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{nk})} = \frac{\sum_{n=1}^{N} \sum_{k=1}^{K} (x_n - \mu_k)(x_n -_k)^\mathsf{T}}{\sum_{n=1}^{N} N_k}}$$

$N_k = \sum_{k=1}^{K} \gamma(z_{nk})$

Furthermore, we utilize the following Lagrange multiplier and differentiate w.r.t. mixing coefficients:

$L = \mathcal{L} + \lambda(\sum_{k=1}^{K} \pi_k - 1)$

$\frac{dL}{d\pi_k} = \frac{\mathcal{L}}{d\pi_k} + \lambda \frac{d}{d\pi_k}(\sum_{j=1}^{K} \pi_j - 1)$

$\frac{1}{\pi_k} \sum_{n=1}^{N} \gamma(z_{nk}) + \lambda = 0$

$\pi_k = -\frac{1}{\lambda} \sum_{n=1}^{N} \gamma(z_{nk})$

Substitute back to constraint:

$-\sum_{k=1}^{K} \frac{1}{\lambda} \sum_{n=1}^{N} \gamma(z_{nk}) = 1$

$-\frac{1}{\lambda} \sum_{n=1}^{N} \sum_{k=1}^{K} \gamma(z_{nk}) = 1$

$-\frac{1}{\lambda} \sum_{n=1}^{N} 1 = 1$

$\gamma = -N$

$$\boxed{\pi_k = \frac{1}{N} \sum_{n=1}^{N} \gamma(z_{nk})}$$

2 Neural Networks

2.1 Basic Generalization

The neural network performs very similarly in the training and validation sets and cross entropies for both steadily decrease. However, the discrepancy between the training and validation error increases as number of epochs increase, with the training CE lower than validation CE, indicating that overfitting might have occurred. Refer to plot figures 1-6 for plots of cross entropy.

2.2 Classification error

Refer to plot figures 7-8 for plots of classification error rates and cross entropy over a run of 100 epochs and 10 hidden units.

2.3 Learning rate

Convergence is usually faster when the learning rate is higher, although it is more likely to have oscillations, as can be seen in plot figures 9-16. Momentum generally increases the convergence rate of the cross entropy across training and validation, as can be seen in a comparison of plot figures 17-22. The optimal value of such hyperparameters can be taken empirically or approximated by cross-validation.

2.4 Number of hiddens

Refer to plot figures 23-32 for plots of the cross entropy and classification error rate over a range of hidden units. Generally the performance of the network increases as the number of hidden units increases when trained for the same number of epochs, since it is able to model more complex non-linear relationships between the input and output.

2.5 Compare with KNN

When the training and validation sets are big and the data is clustered in clear clusters, KNN performs quite well, surpassing the performance of the previous runs of the neural network, as can be seen in plot figures 33-34. The best performances achieved more than 98% classification rate in both validation and test sets.

3 Mixture of Gaussians

3.1 Training

Refer to figures 35-38 for images of the mean and variance images of The following hyper-parameter settings were used for each model:
K = 2
Iterations = 20

Min Variance = 0.01
The results are shown below:

Table 1: Mixing coefficients and log probability after iter=20

Model.Cluster	Mixing Coef	logP (for entire model)
2.1	0.49335	
2.2	0.50665	-3868.40
3.1	0.51146	
3.2	0.48854	2272.26

3.2 Initialize MoG with K-means

The hyperparameters used for this comparison are as follows:
K = 20
Iterations = 20
Min Variance = 0.01
The K-means initialization yielded faster convergence rates on the plots, and the mean/variance visualizations are shown in the appendix. The final log probability is significantly higher for a kmeans initialization and converges slightly faster in the first few iterations, as shown in Table 2, because in essence the initialization already contains 5 iterations of kmeans.

Table 2: Final logP of models after iter=10

Model Initialization	Final logP
random	25362.15
kmeans	27724.57

3.3 Classification using MoG

Refer to plot figure 45 for the plot of classification error rates over a range of components for the entire data set. The following observations can be made:
1. The error rates generally decreases for all data sets as cluster components increase. This is obvious due to the fact that we can model more groups that share a particular quality (e.g. slight shift or tilt).
2. The error rate curve for the test set is consistently higher than the training set which is to be expected, and exhibits the same trends as the training curve, indicating that the model is still optimized w.r.t. data.
3. As can be seen in the figure, the training error rate becomes 0 at 25 components, while the validation error sharply increases; this is an indication of the data being overfitted to the training set. In this situation, the model with 15 components for each class (30 in total) is more competitive, because it yields the lowest average error rate across the three data sets, and we have not overfitted yet.

3.4 Comparison with neural networks

For comparison purposes, the neural network has 30 hidden units, corresponding to the total 30 components used in the MoG model. Over training, we can see a sample of the hidden unit input layer visualizations in the figures in appendix. The figures 46-55 show the visualization for one hidden unit sampled at stride 20 in training, indicating that the hidden unit has gradually been trained to classify one variation of the digit as training progresses. Complete visualizations can be seen in the attached folders E1-E10.

4 Appendix

Figure 1: 2.1 Cross entropy over epochs=100, 10 hiddens, run 1

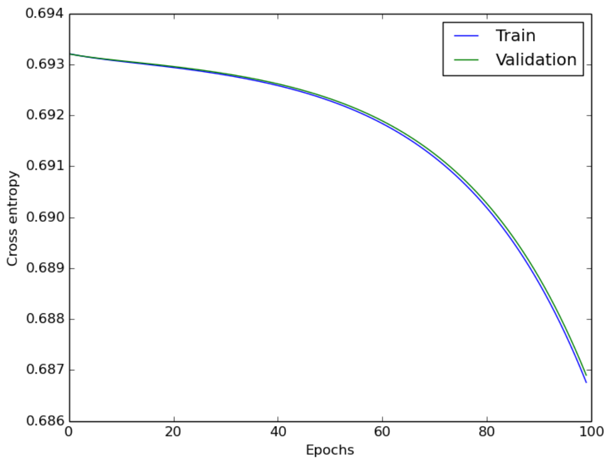

Figure 2: 2.1 Cross entropy over epochs=100, 10 hiddens, run 2

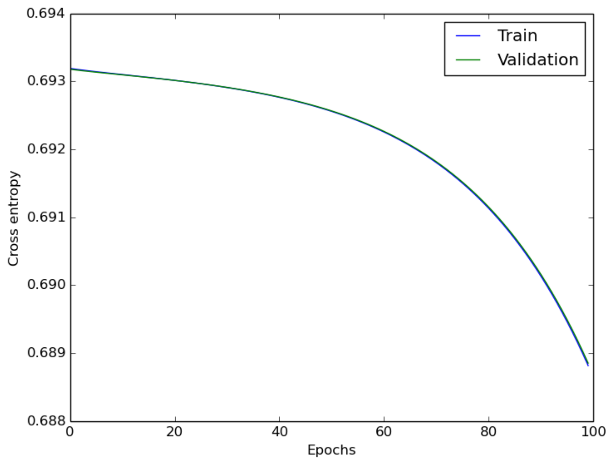

Figure 3: 2.1 Cross entropy over epochs=100, 10 hiddens, run 3

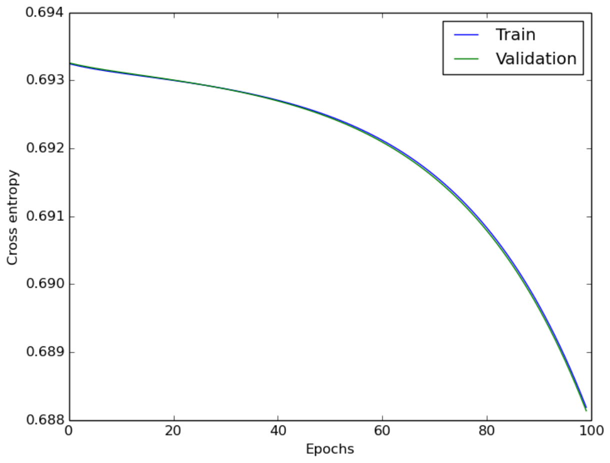

Figure 4: 2.1 Cross entropy over epochs=100, 10 hiddens, run 4

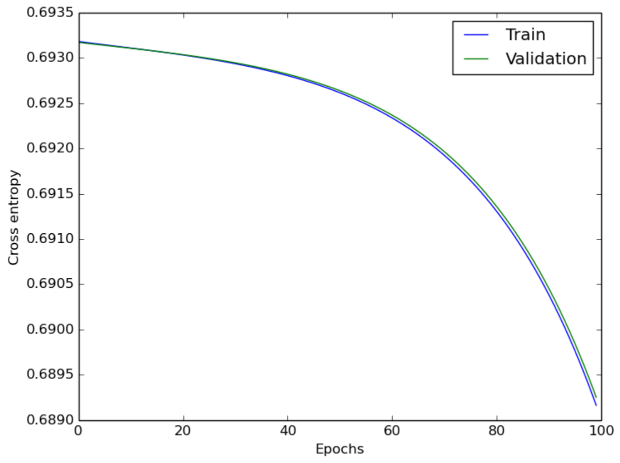

Figure 5: 2.1 Cross entropy over epochs=100, 10 hiddens, run 5

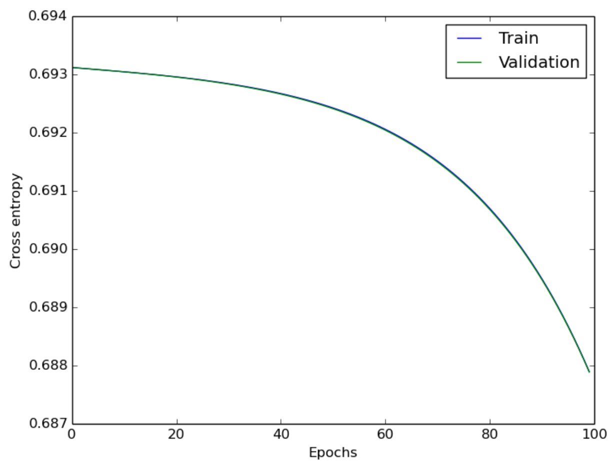

Figure 6: 2.1 Cross entropy over epochs=500, 10 hiddens

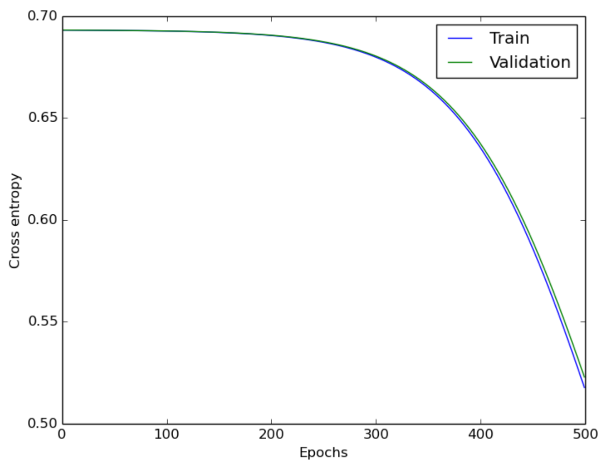

Figure 7: 2.2 Cross entropy over epochs=100, 10 hiddens

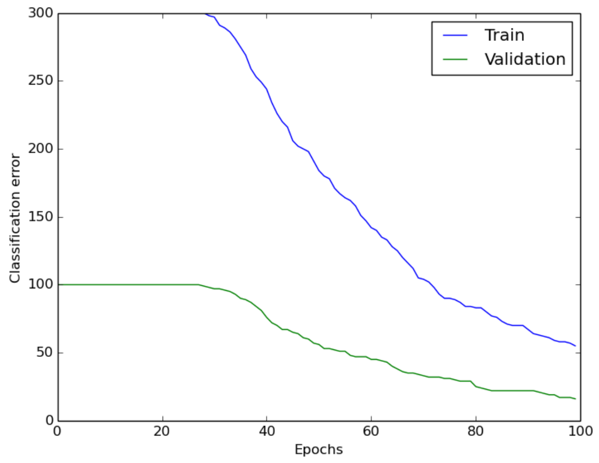

Figure 8: 2.2 Class error rate over epochs=100, 10 hiddens

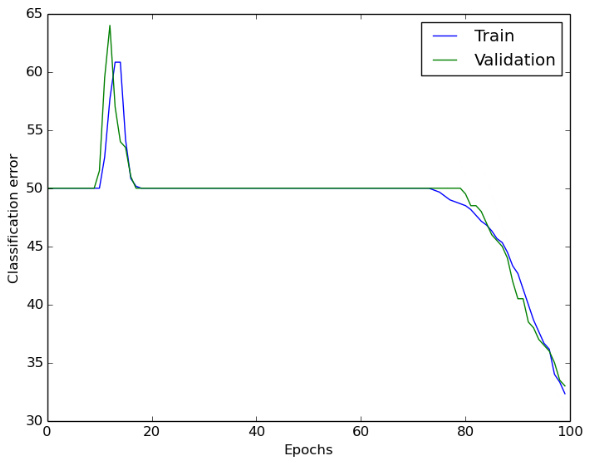

Figure 9: 2.3 Cross entropy over epochs=100, 10 hiddens, learning rate=0.01

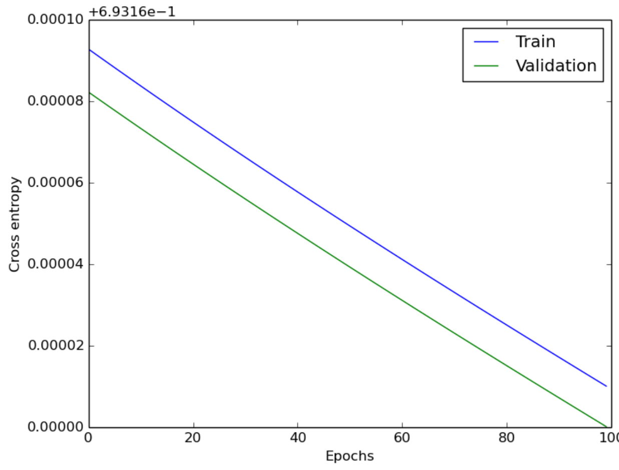

Figure 10: 2.3 Class error rate over epochs=100, 10 hiddens, learning rate=0.01

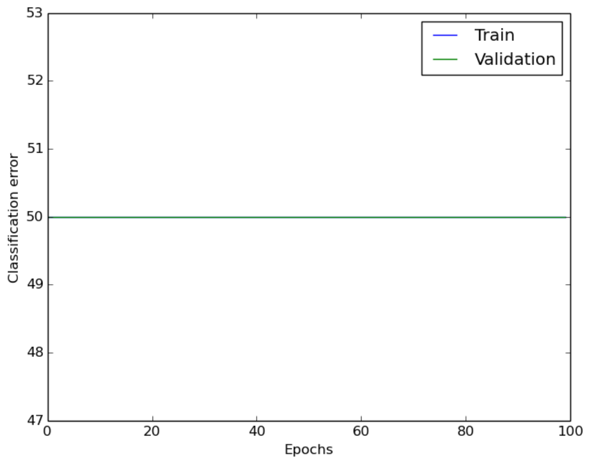

Figure 11: 2.3 Cross entropy over epochs=100, 10 hiddens, learning rate=0.1

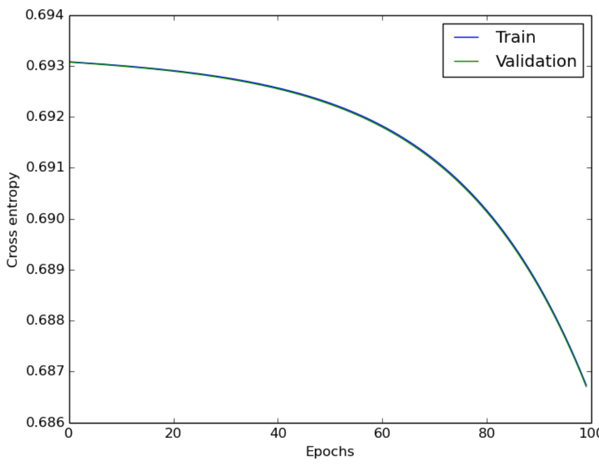

Figure 12: 2.3 Class error rate over epochs=100, 10 hiddens, learning rate=0.1

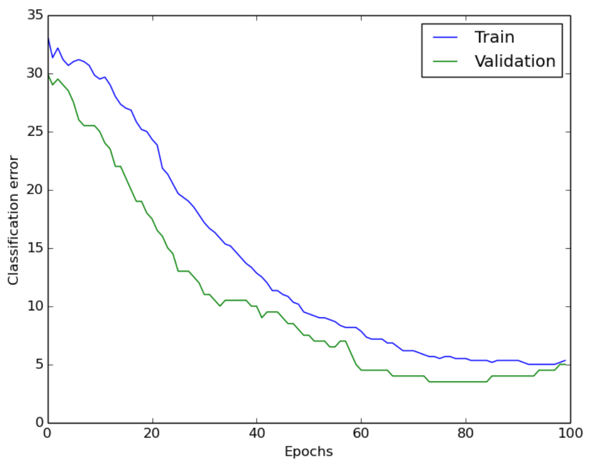

Figure 13: 2.3 Cross entropy over epochs=100, 10 hiddens, learning rate=0.2

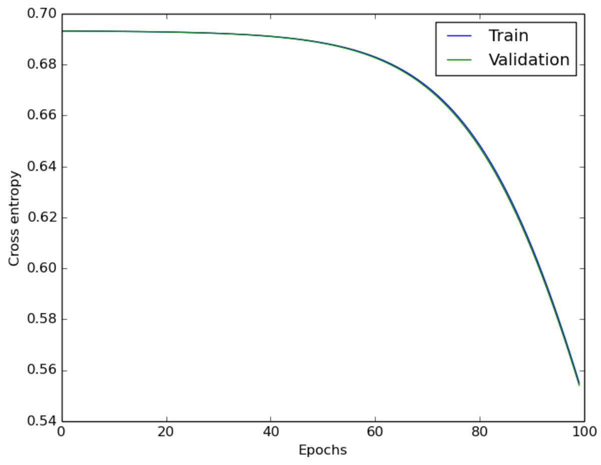

Figure 14: 2.3 Class error rate over epochs=100, 10 hiddens, learning rate=0.2

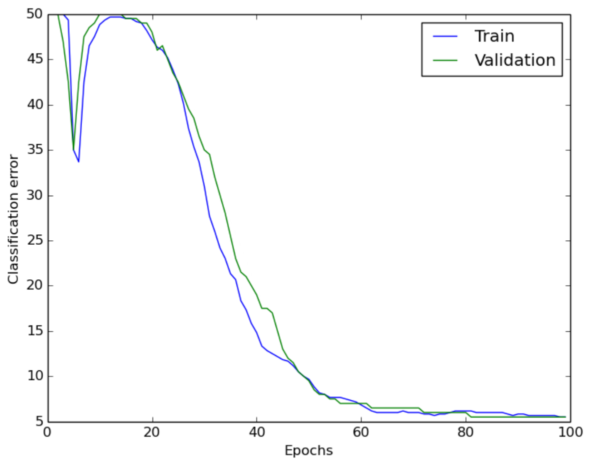

Figure 15: 2.3 Cross entropy over epochs=100, 10 hiddens, learning rate=0.5

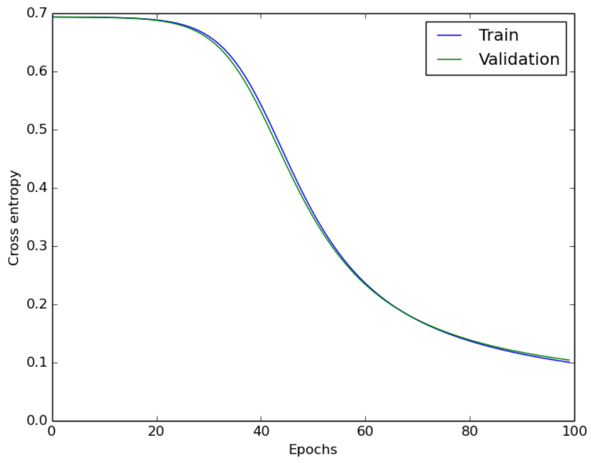

Figure 16: 2.3 Class error rate over epochs=100, 10 hiddens, learning rate=0.5

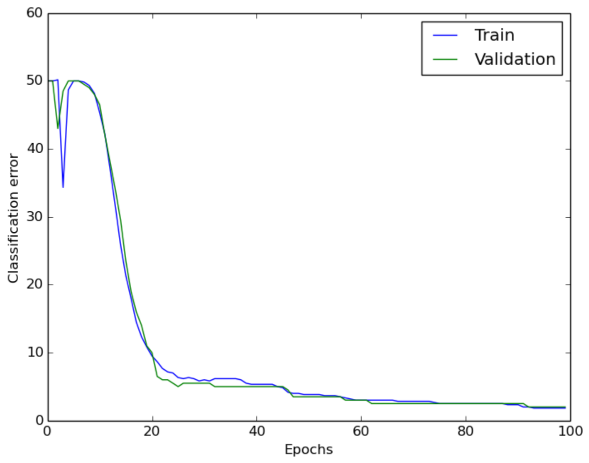

Figure 17: 2.3 Cross entropy over epochs=100, 10 hiddens, momentum=0

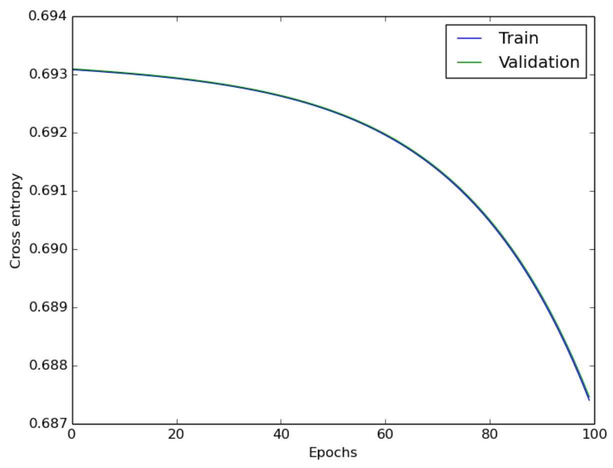

Figure 18: 2.3 Class error rate over epochs=100, 10 hiddens, momentum=0

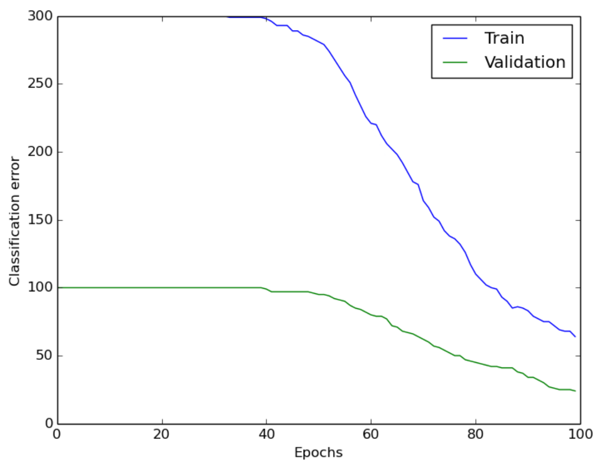

Figure 19: 2.3 Cross entropy over epochs=100, 10 hiddens, momentum=0.5

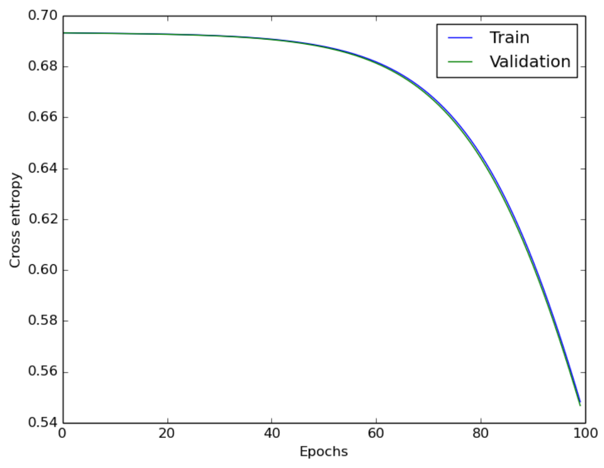

Figure 20: 2.3 Class error rate over epochs=100, 10 hiddens, momentum=0.5

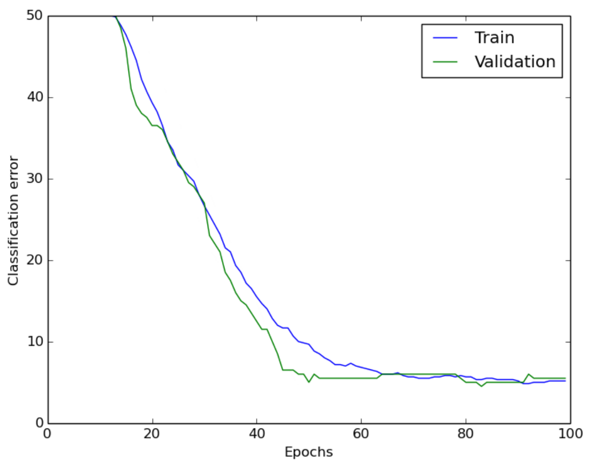

Figure 21: 2.3 Cross entropy over epochs=100, 10 hiddens, momentum=0.9

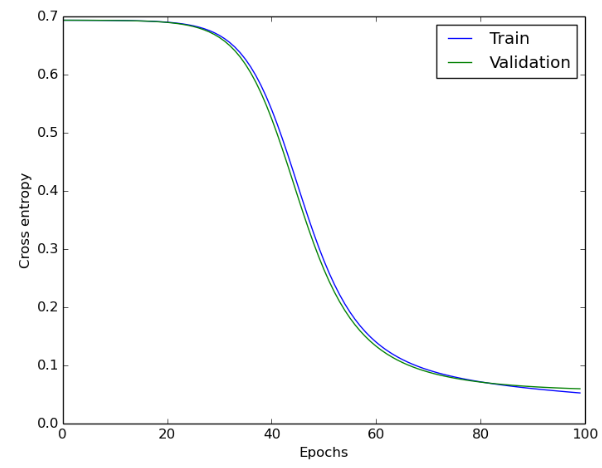

Figure 22: 2.3 Class error rate over epochs=100, 10 hiddens, momentum=0.9

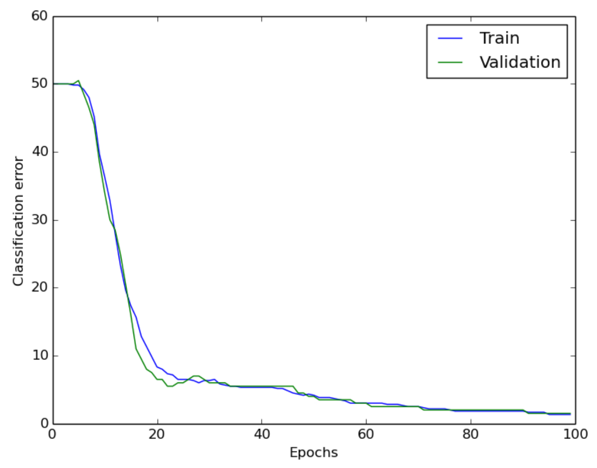

Figure 23: 2.4 Cross entropy over epochs=500, 2 hiddens, momentum=0.5, learning rate=0.02

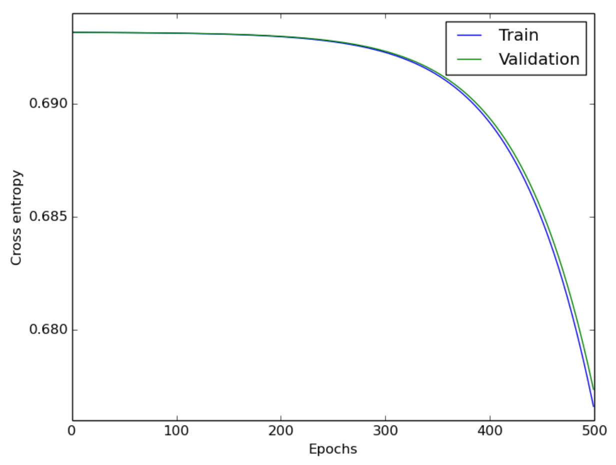

Figure 24: 2.4 Class error rate over epochs=100, 2 hiddens, momentum=0.5, learning rate=0.02

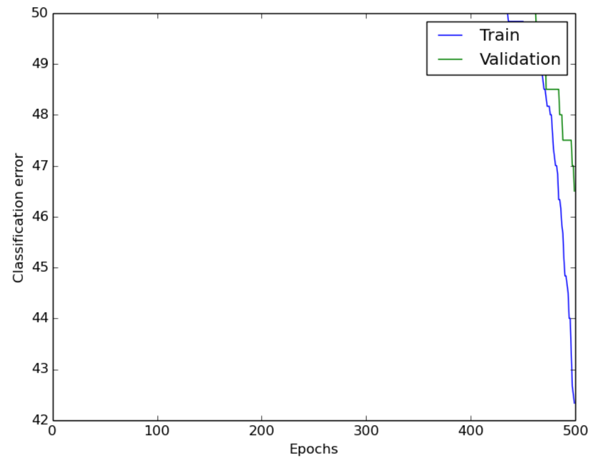

Figure 25: 2.4 Cross entropy over epochs=500, 5 hiddens, momentum=0.5, learning rate=0.02

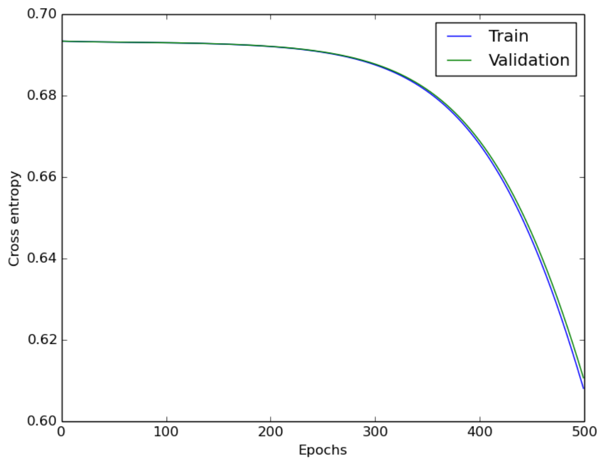

Figure 26: 2.4 Class error rate over epochs=100, 5 hiddens, momentum=0.5, learning rate=0.02

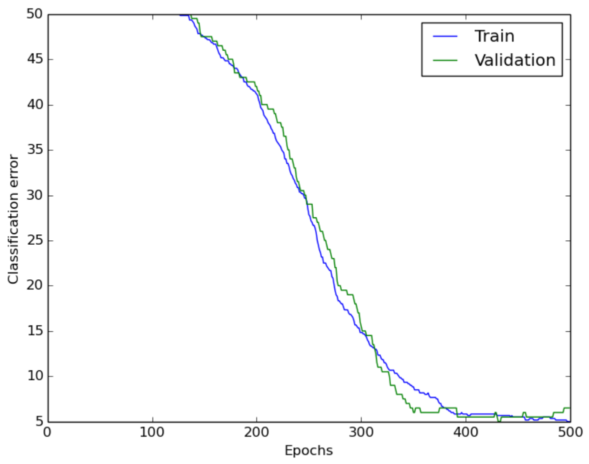

Figure 27: 2.4 Cross entropy over epochs=500, 10 hiddens, momentum=0.5, learning rate=0.02

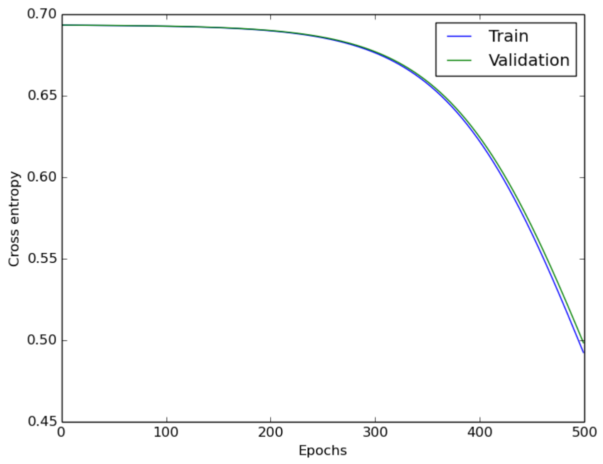

Figure 28: 2.4 Class error rate over epochs=100, 10 hiddens, momentum=0.5, learning rate=0.02

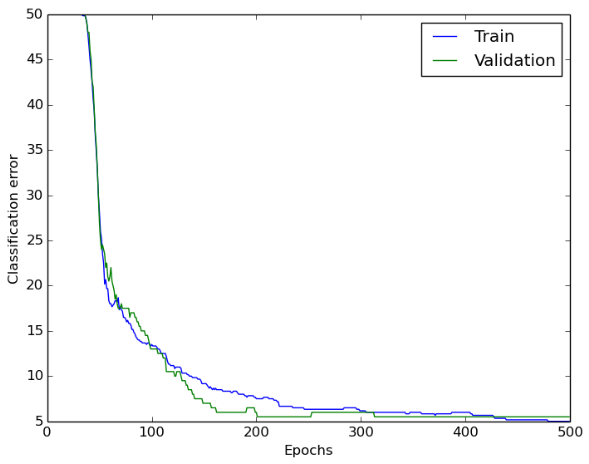

Figure 29: 2.4 Cross entropy over epochs=500, 30 hiddens, momentum=0.5, learning rate=0.02

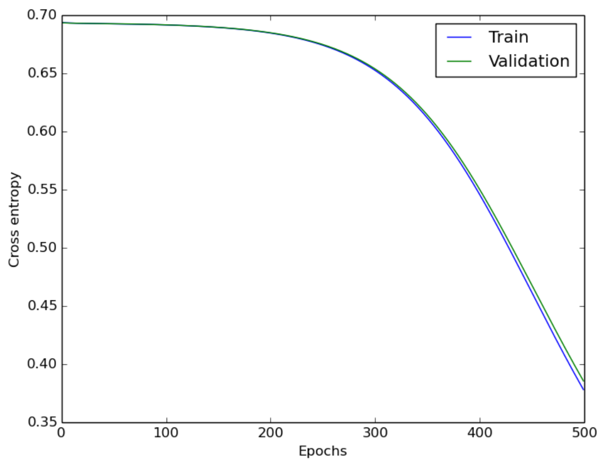

Figure 30: 2.4 Class error rate over epochs=100, 30 hiddens, momentum=0.5, learning rate=0.02

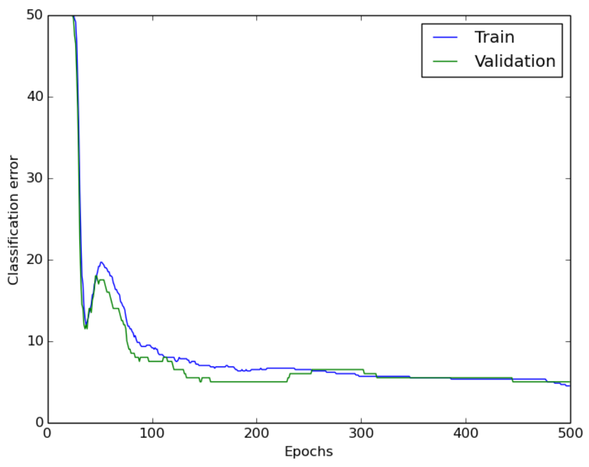

Figure 31: 2.4 Cross entropy over epochs=500, 100 hiddens, momentum=0.5, learning rate=0.02

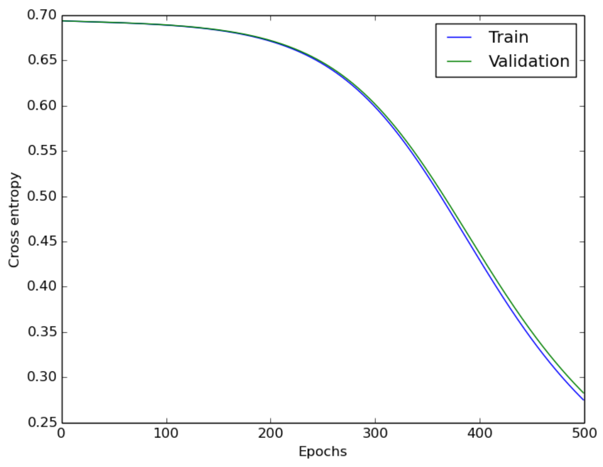

Figure 32: 2.4 Class error rate over epochs=100, 100 hiddens, momentum=0.5, learning rate=0.02

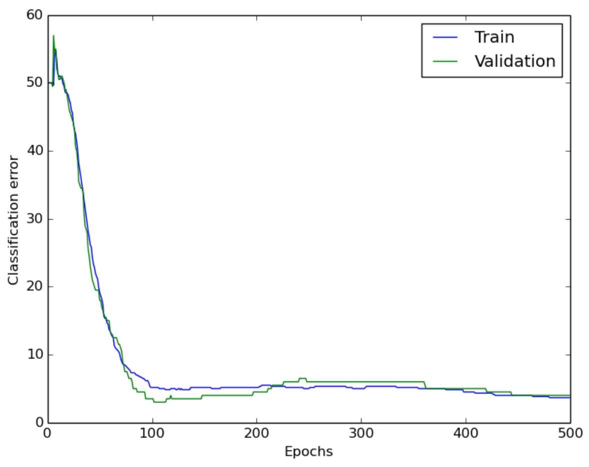

Figure 33: Classification rate of KNN algorithm in validation set

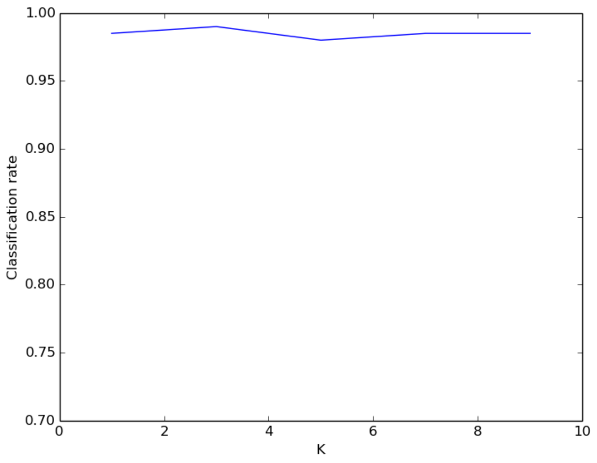

Figure 34: Classification rate of KNN algorithm in test set

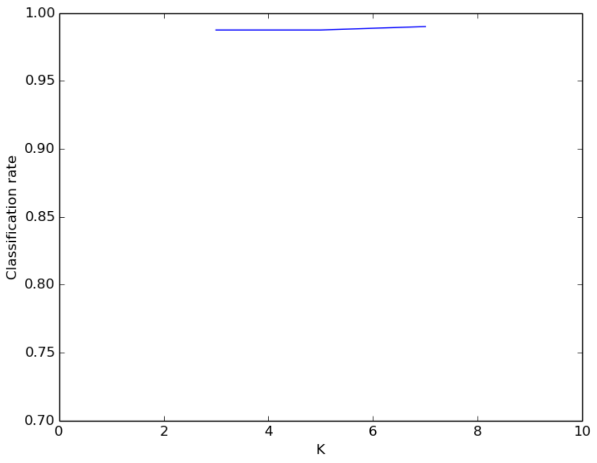

Figure 35: 3.2 Mixture of Gaussians mean image for '2', iter=20, K=2

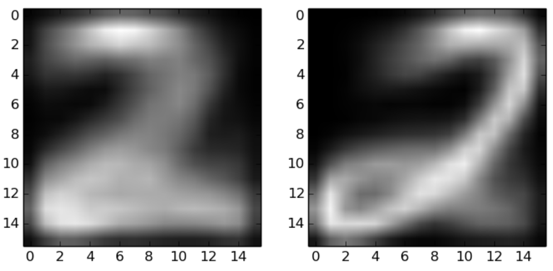

Figure 36: 3.2 Mixture of Gaussians variance image for '2', iter=20, K=2

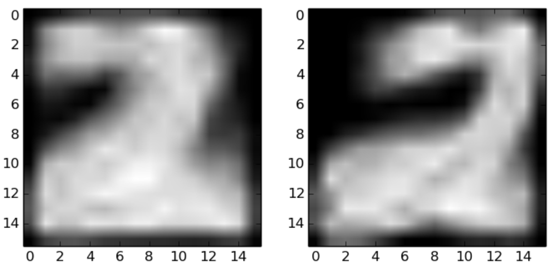

Figure 37: 3.2 Mixture of Gaussians mean image for '3', iter=20, K=2

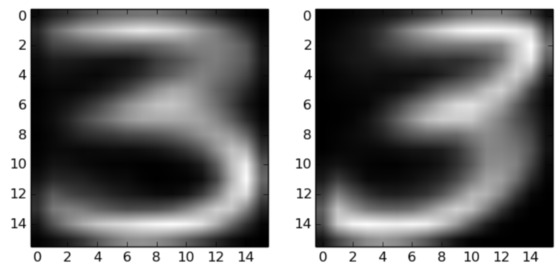

Figure 38: 3.2 Mixture of Gaussians variance image for '3', iter=20, K=2

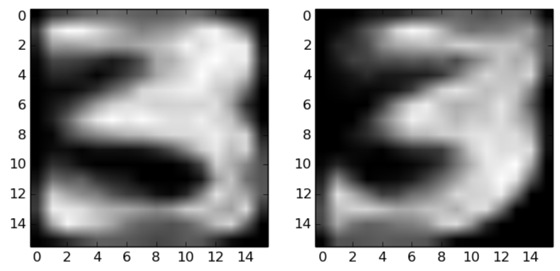

Figure 39: 3.3 Mixture of Gaussians log probability with K-means, iter=10, K=20

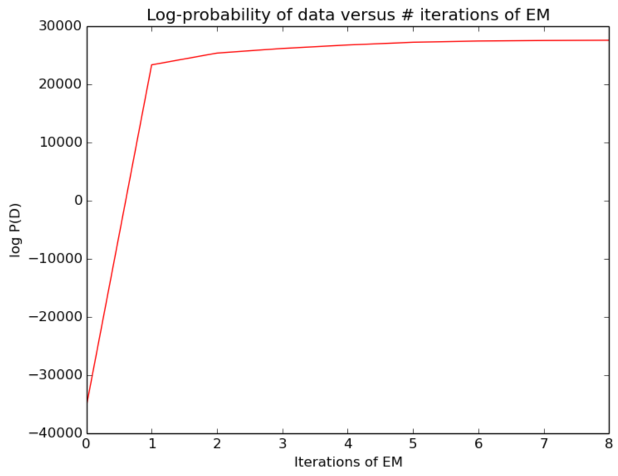

Figure 40: 3.3 Mixture of Gaussians mean image with K-means, iter=10, K=20

Figure 41: 3.3 Mixture of Gaussians variance image with K-means, iter=20, K=20

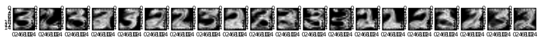

Figure 42: 3.3 Mixture of Gaussians log probability with random initialization, iter=20, K=20

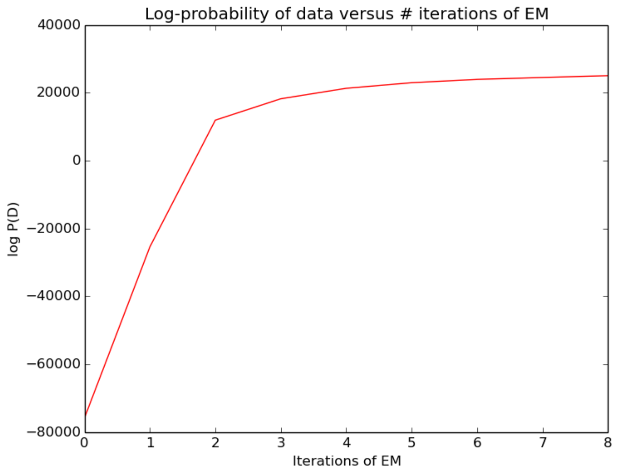

Figure 43: 3.3 Mixture of Gaussians mean image with random initialization, iter=20, K=20

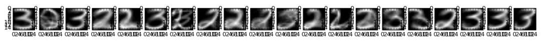

Figure 44: 3.3 Mixture of Gaussians variance image with random initialization, iter=20, K=20

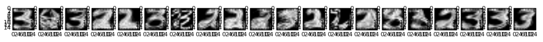

Figure 45: 3.4 MoG classification error rate with varying components, iter=10

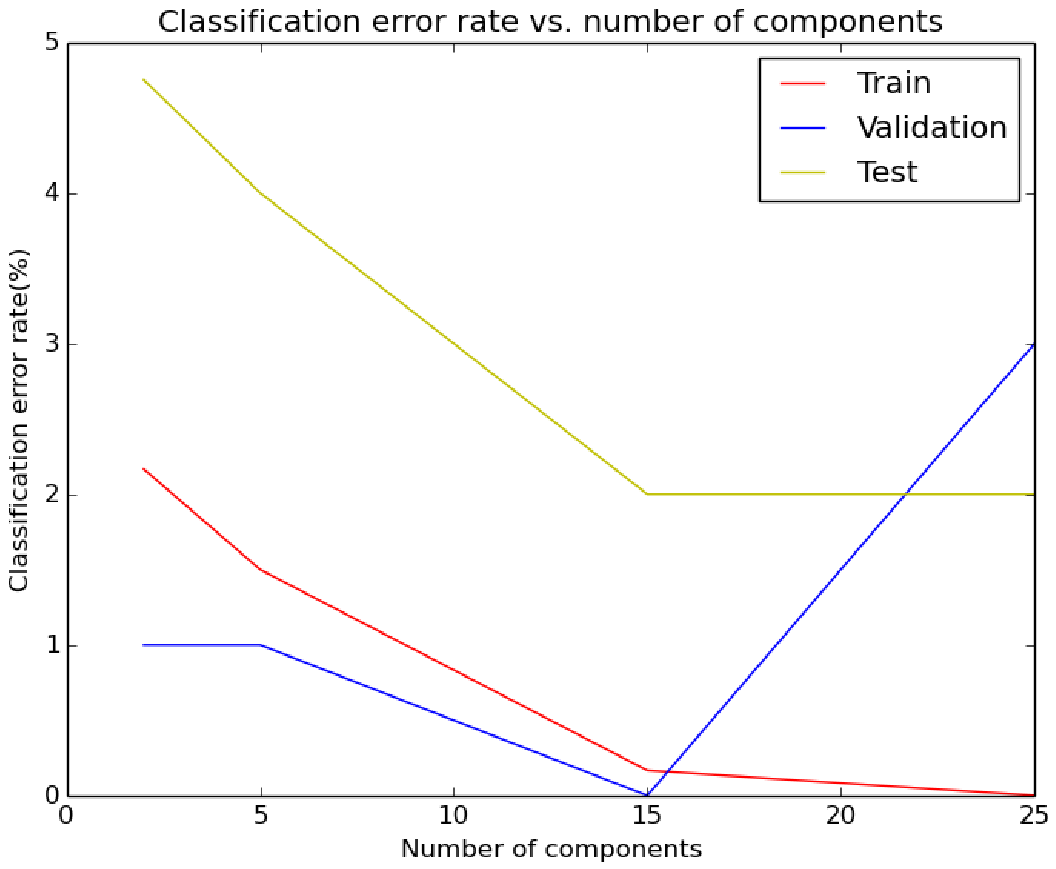

Figure 46: 3.5 Visualization of one hidden unit at epoch 0

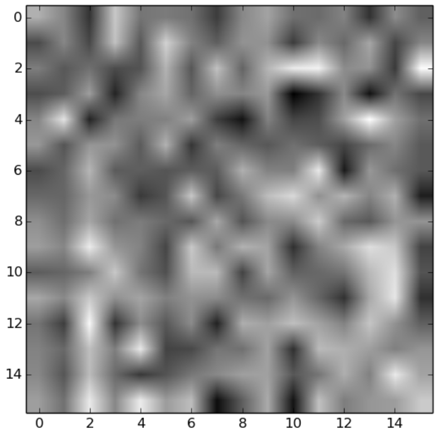

Figure 47: 3.5 Visualization of one hidden unit at epoch 20

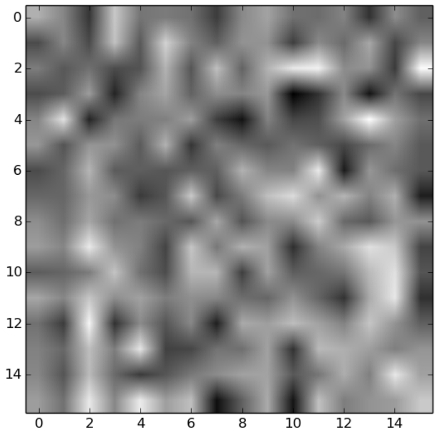

Figure 48: 3.5 Visualization of one hidden unit at epoch 40

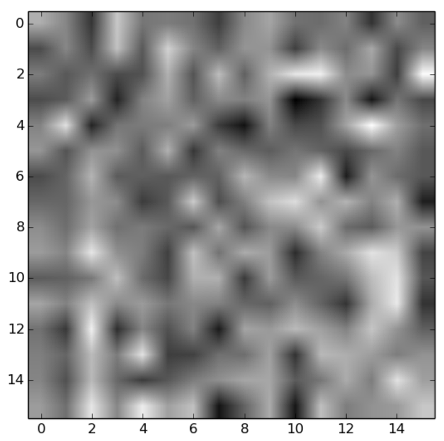

Figure 49: 3.5 Visualization of one hidden unit at epoch 60

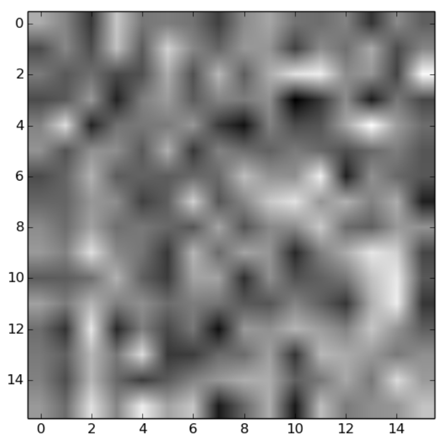

Figure 50: 3.5 Visualization of one hidden unit at epoch 80

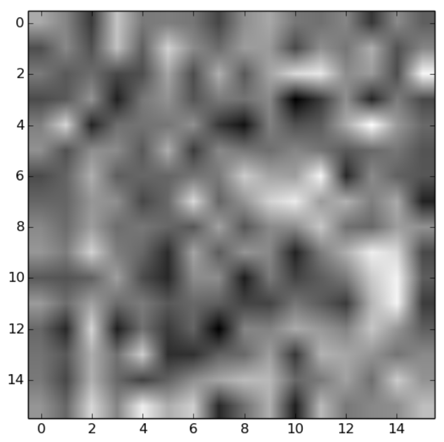

Figure 51: 3.5 Visualization of one hidden unit at epoch 100

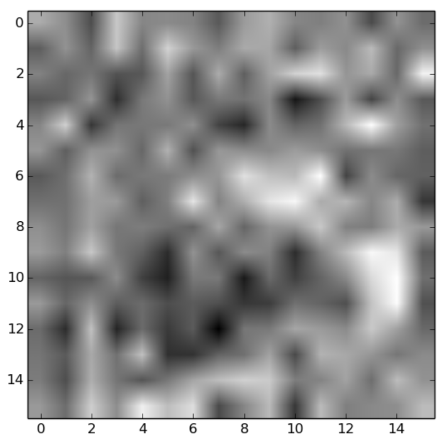

Figure 52: 3.5 Visualization of one hidden unit at epoch 120

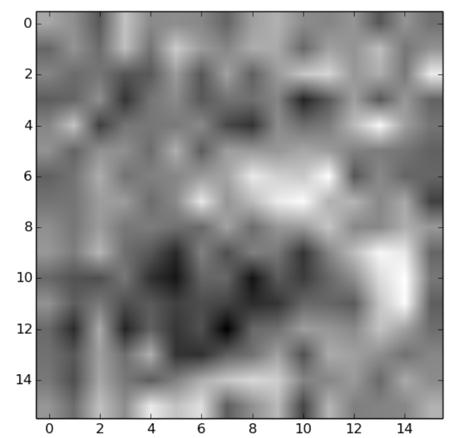

Figure 53: 3.5 Visualization of one hidden unit at epoch 140

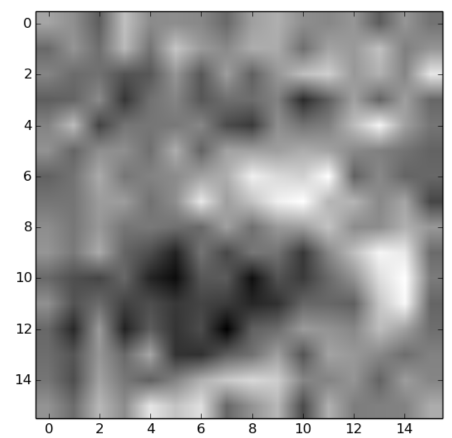

Figure 54: 3.5 Visualization of one hidden unit at epoch 160

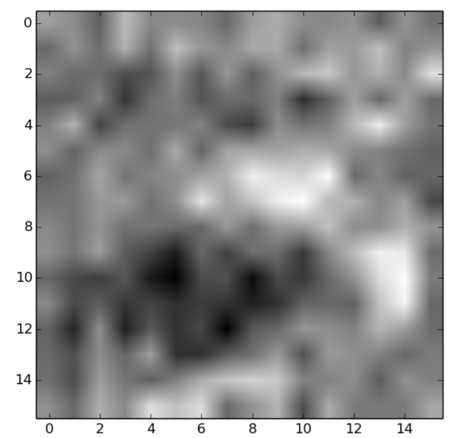

Figure 55: 3.5 Visualization of one hidden unit at epoch 180