Task 1, Hopfield-Boltzmann Machine

Hopfield Network

Instead of Correlation Memory which achieves partial Perfection for heavily distorted digits even after many loops Pseudo Inverse Memory is achieving best for 2 Loops as shown in Figure 1. When we use only 1 Loop then it comes distorted as shown in Figure 2

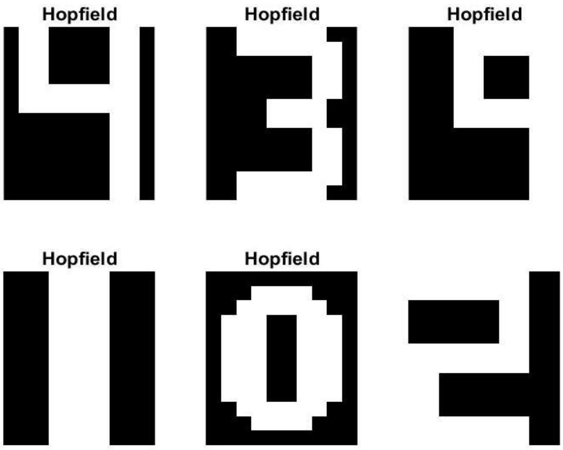


Figure 1 Pinv after 2 Loops

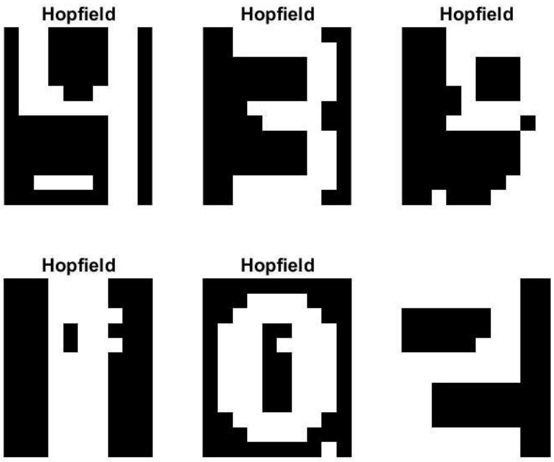


Figure 2 Hopfield Pinv 1 Loop

Boltzmann Machine

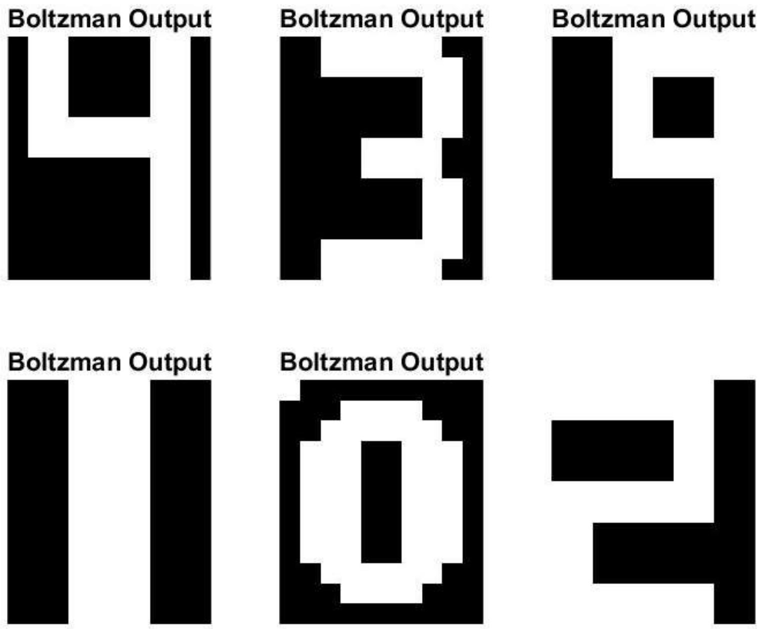
By giving Input from Hopfield of 1 Loop, results are improved. After 2 Loops Hopfield is achieving perfection so Boltzman we cant find any change since its already achieving perfection using Hopfield.

When keeping T equals to zero we observed that it is performing same as Hopfield and no change is observed. Initially we keep large number of T and decrease gradually to zero.

If we keep very large T then its becoming too distorted due to probability and stabilizing at some other points giving error. So T should be taken not too small since it will be same as Hopfield with no change also not too large T since it will behave as random network due to probability's assigning different states. Around T equals 1 to 4 is giving optimum good results but as T increases large error due to probabilitys.

By T equals to 1 performs good after some trying. For other values large error as T increases.

Boltzmann was able to improve results of Hopfield Network



Task 2, ICA Image Separation

It requires 1 iteration for seeing without any dominant artifacts. By choosing large step size (0.2) its converging faster as a result large error for next values resulting in black image and if we choose small step size(0.002) it is taking too long to converge and we observe the dominant artifacts as a result no separation since it takes too long to converge for every value.

Task 3, Principal Component Analysis

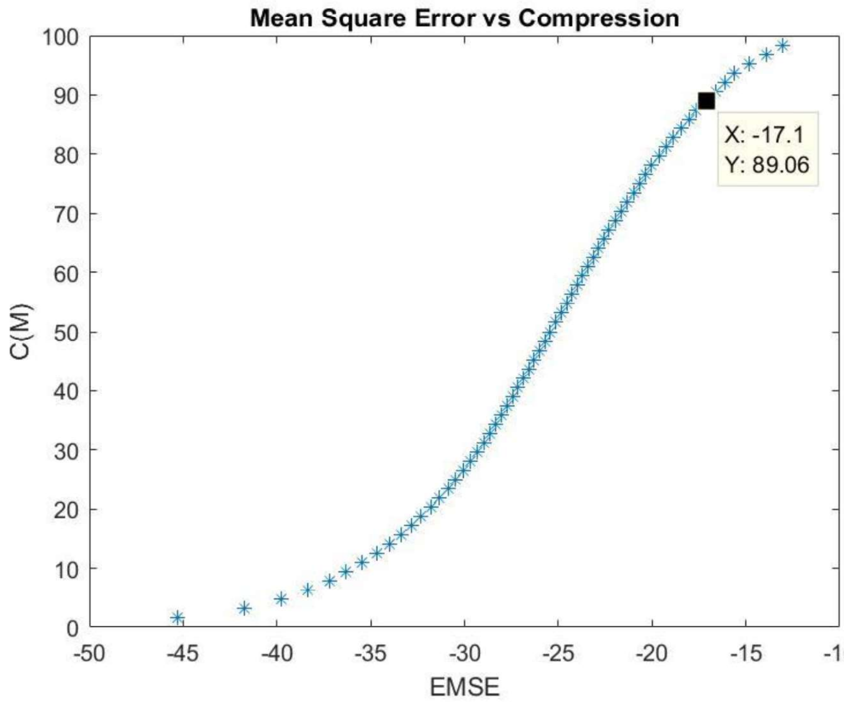
Task 3-A Eigen Vectors Compression and MSE

Highest Possible Compression Ratio before image becomes unreadable: -

5,4 principal Components if 3 unreadable has highest variance too if we consider eigen values when compared to others which has less variance so we can compress it. Compression is around 92 Percent

Components of M required for less than 17 dB error: -

7 Principal Components around 89 Percent Compression.



Task 3-B

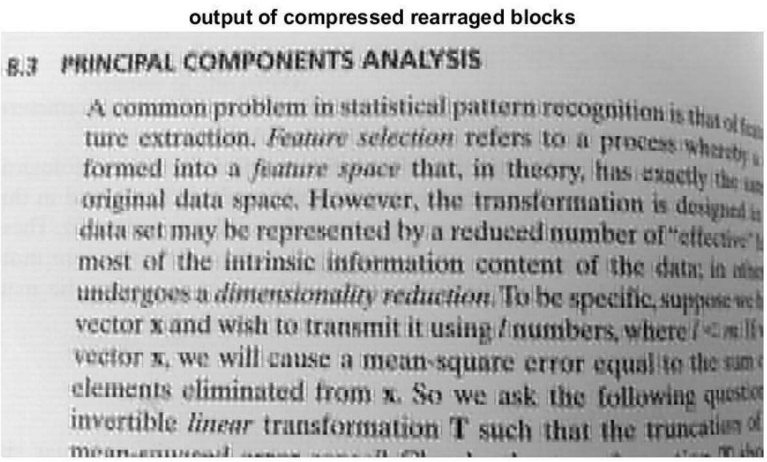
Iterations Required to get EMSE same as in previous Task: -

250 iterations giving 17.01 with M=7

600 iteratins17.10 same as last task with M=7

M Principal Components that has less than -17db error: -

11 Principal Components need for getting less than -17 db error with 10 iterations. If using more number of iterations like 250 with M=7 only giving less than -17db error.

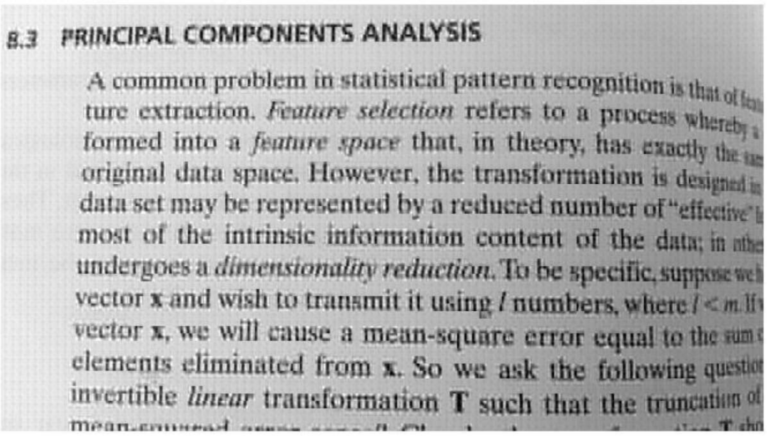


8.3 PRINCIPAL COMPONENTS ANALYSIS

A common problem in statistical pattern recognition is that of fea-
ture extraction. *Feature selection* refers to a process whereby a
formed into a *feature space* that, in theory, has exactly the sa
original data space. However, the transformation is designed i
data set may be represented by a reduced number of "effective"
most of the intrinsic information content of the data; in othe
undergoes a *dimensionality reduction*. To be specific, suppose we h
vector x and wish to transmit it using *l* numbers, where *l* < m. If
vector x, we will cause a mean-square error equal to the sum
elements eliminated from x. So we ask the following questio
invertible *linear* transformation T such that the truncation of

Figure 3 Eigen Values 7 Principal Components

GCA in Figure 4 gives better clarity in image and be able to easy to read for same MSE even though at cost of high number of iterations when compared with calculating directly in Figure 3.



8.3 PRINCIPAL COMPONENTS ANALYSIS

A common problem in statistical pattern recognition is that of fea-
ture extraction. *Feature selection* refers to a process whereby a
formed into a *feature space* that, in theory, has exactly the sa
original data space. However, the transformation is designed i
data set may be represented by a reduced number of "effective"
most of the intrinsic information content of the data; in othe
undergoes a *dimensionality reduction*. To be specific, suppose we
vector **x** and wish to transmit it using *l* numbers, where $l < m$. If
vector **x**, we will cause a mean-square error equal to the sum
elements eliminated from **x**. So we ask the following questio
invertible *linear* transformation **T** such that the truncation of

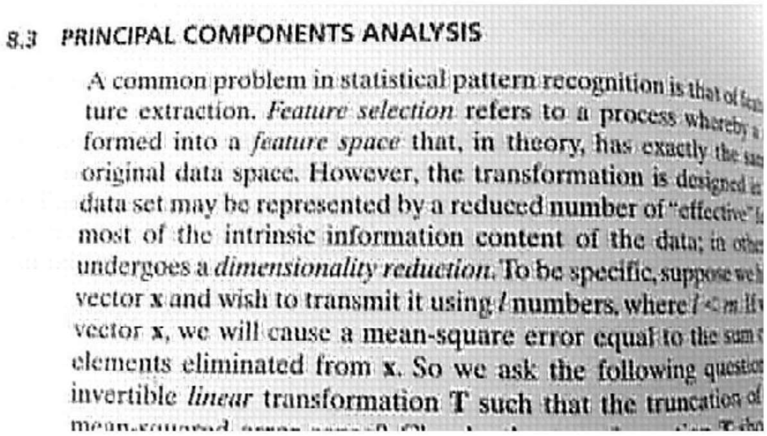
Figure 4 250 iterations 7 Principal Components

Task 3-C Feed Forward 2 Layer Back Propagation

More principal Components like 30 etc giving better clarity but at cost of compression.

Same Principal components as previous task Emse:-

With M=7 Principal Componenets getting -10 dB around. Iterations :-20



8.3 PRINCIPAL COMPONENTS ANALYSIS

A common problem in statistical pattern recognition is that of fea-
ture extraction. *Feature selection* refers to a process whereby a
formed into a *feature space* that, in theory, has exactly the sa
original data space. However, the transformation is designed i
data set may be represented by a reduced number of "effective"
most of the intrinsic information content of the data; in othe
undergoes a *dimensionality reduction*. To be specific, suppose we
vector **x** and wish to transmit it using *l* numbers, where $l < m$. If
vector **x**, we will cause a mean-square error equal to the sum
elements eliminated from **x**. So we ask the following questio
invertible *linear* transformation **T** such that the truncation of

Figure 5 Only 7 Prinicipal Componnets image

Task 4, Prediction Using Multilayer Neural Network

Optimal Network Structure number of neurons per layer

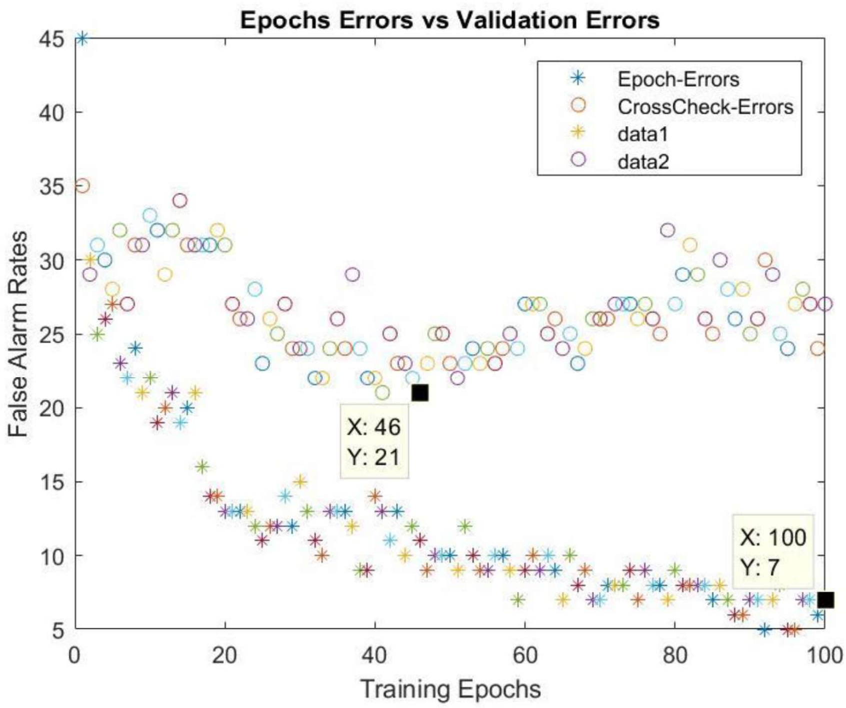
For 2 Layer Neural Network :16-3-1

Other Neurons cost computation and errors increasing

Iterations required before stopping: -

Circles are Validated Samples which are we used to test for given training weights obtained using epochs. As epochs increases we see that false alarm rates increases greater than 46 as shown in Figure is making Validated Samples increase in error though Training Samples error is decreasing. So its better to stop at around 40 to 46 epochs for good network.

Probability Rate is shown in Figure 6



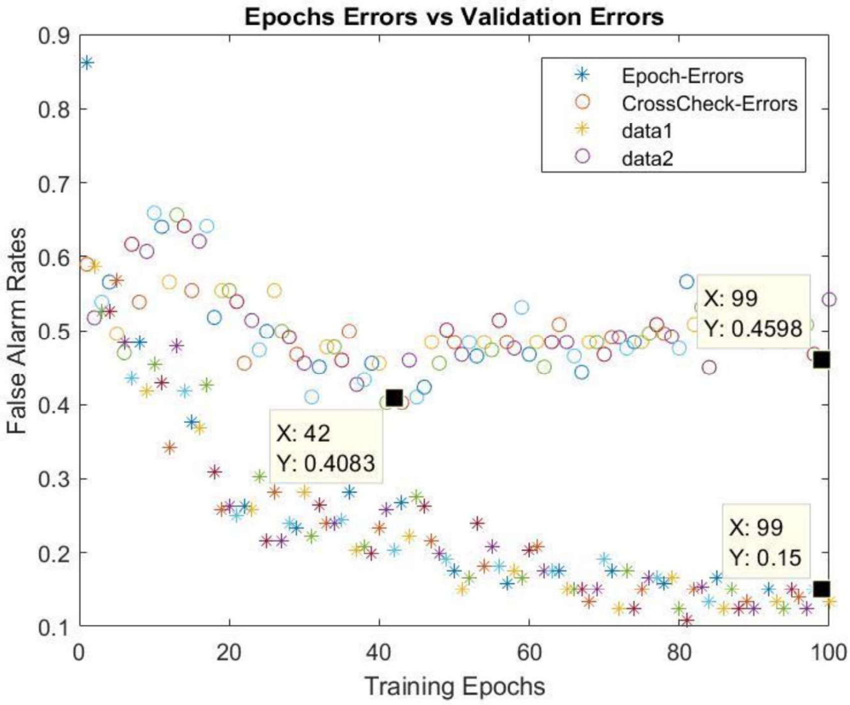
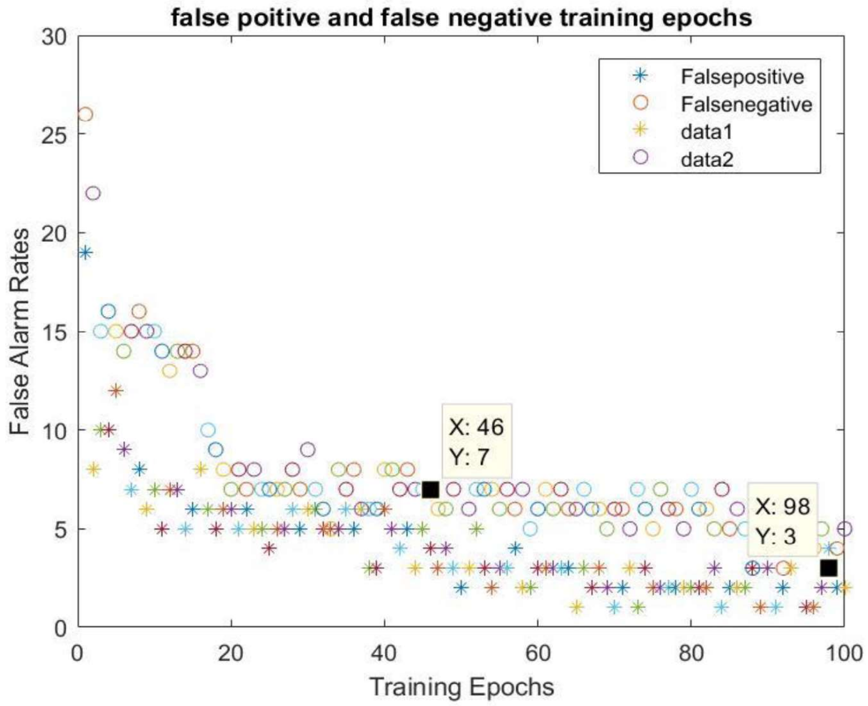


Figure 6

Plot False Alarm Rates during Training epochs.

Since we took 46 Epochs based on validation samples we get around 7 False Alarms out of 100 took. More epochs gives less errors but no generalization as seen in Validation.

Probability Rate is shown in Figure 7


false poitive and false negative training epochs
Falsepositive
Falsenegative
data1
data2
X: 46
Y: 7
X: 98
Y: 3
False Alarm Rates
Training Epochs

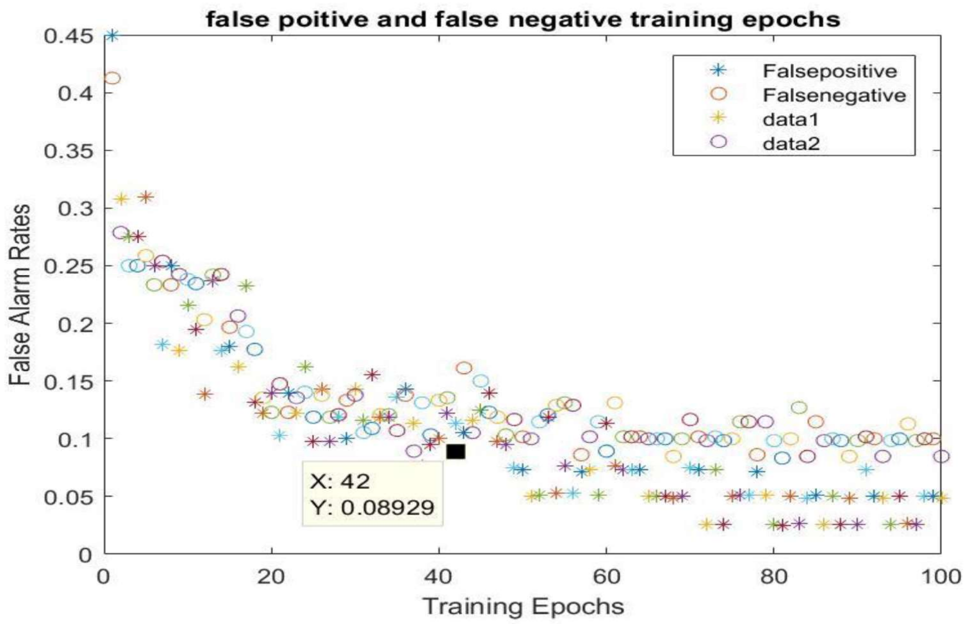

false poitive and false negative training epochs
Falsepositive
Falsenegative
data1
data2
X: 42
Y: 0.08929
False Alarm Rates
Training Epochs

Figure 7

Final False Alarm Rates:-

After Taking 42 Epochs for good testing we test our network on different samples from 200:371 around Samples. We Trained Network using first 100 Samples and Validated using Next 100 Samples for best epochs.

We get Finally

136 True ,36 False Alarms out of 172

132 True, 40 False Alarms out of 172

Probability

True Alarm Rates :-136/172=0.7907;   132/172=0.764

Around 0.8 Percent Correct

False Alarm Rates :-36/72=0.2093;   1-0.764=0.2326

Around 0.2 Percent Wrong

Task 5, Radial Basis Function Network

Optimum Network Structure:-16-3-1

3 Neurons in Neural Network

Iterations Required before Stopping

Around 3 Iterations

Convergence Faster than Previous Task taking very less Time.

Final False Alarm Rates

135 to 141 True out of 172 took around 0.8 to 0.82 probability

Very less Improvement but one improvement is very fast outputs when compared to previous Task.