**Artificial Neural Networks**

*Exercise Session 3 – Unsupervised learning and SOM,*

*report by,*

*Marimuthu Ananthavelu (r0652832), KU Leuven.*

***Principal Components Analysis(PCA):***

Principal components analysis is the dimensionality reduction technique without reduction of the given input features. PCA is applied to the handwritten dataset of 3’s which are stored in the dataset ‘threes.mat’ and the findings are as below;

***Mean of dataset:***

The computed mean of dataset ‘threes’, Eigenvalues of the covariance matrix for the given dataset, the reconstructed image with 4 principal components are displayed as below;



When rebuilding the character set with the found principal components, there is reconstruction error which remains due to the fact the number of components is reduced. The more components we choose from the output vectors to rebuild the original character, the less reconstruction error it is to be. For the first 50 components, the reconstruction error appears as below;

For the first 50 components, the reconstruction error still appears. So, with all the output vectors from PCA (k=256), the reconstruction error ideally should be ‘0’. This can be checked by calculating/plotting the reconstruction error with k=256. The same can be seen below;

As seen above, the reconstruction error is decreasing with respect to the increase in the number of principal components which in turn makes the cumulative sum of the eigenvalues are reaching towards 1.

***Self-Organizing Maps:***

Self-organizing feature maps (SOFM) learn to classify input vectors per how they are grouped in the input space. They differ from competitive players in that neighbouring neurons in the self-organizing map learn to recognise neighbouring sections of the input space. Thus, self-organizing maps learn both the distribution (as do competitive layers) and topology of the input vectors they are trained on.

**Applying SOM on Cylinder data**



***Fig. random topology***

***before training***

***Distance function: linkdist***

***Fig. random topology***

***after training***

***Distance function: linkdist***

***Fig.random topology***

***after training***

***Fig.random topology***

***before training***

**Applying SOM on Iris Dataset**