CZ4042 Tutorial 10 **Autoencoders**

1. Given five binary patterns:
$$\boldsymbol{x}_1 = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \boldsymbol{x}_2 = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \boldsymbol{x}_3 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \boldsymbol{x}_4 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \boldsymbol{x}_5 = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

   Design an autoencoder with four hidden neurons to reconstruct the patterns, using gradient descent learning with a learning parameter $\alpha = 0.1$.
   Find the weights, biases, hidden-layer activations and reconstructions of the input patterns at convergence.

   Repeat the above by introducing a sparsity constraint with a penalty parameter $\beta = 0.5$ and sparsity parameter $\rho = 0.1$.

2. Create 100 images of 10x10 size by randomly generating pixel values between 0.0 and 1.0 from a uniform distribution.
   Design the following autoencoders to reconstruct the input patterns, using mean square error as the cost function:
   a. An undercomplete autoencoder with 49 hidden neurons
   b. An overcomplete autoencoder with 144 hidden neurons
   c. A sparse autoencoder with 144 hidden neurons and training with sparsity parameter $\rho = 0.05$ and penalty parameter $\beta = 0.5$.

   Compare features learned by different autoencoders.

3. Design a denoising autoencoder to reconstruct MNIST images:
   http://yann.lecun.com/exdb/mnist/

   (a) Assume one hidden layer with 625 neurons, multiplicative noise, and cross-entropy cost function. Use 10% corruption level, learning factor $\alpha = 0.1$, batch size = 128, sparsity constant $\rho = 0.02$, and penalty parameter $\beta = 0.4$.
   Plot the learning curves, the weights, and the hidden layer activations for sample test images.

   (b) Add another hidden layer with 100 neurons and train the autoencoder as before. Plot the feature maps.
   Plot the learning curves, the weights, and the hidden layer activations for sample test images

   (c) Add a softmax layer on top of the second hidden layer to design a classifier. Show learning curves and find the accuracy of the classifier.
   Plot the learning curves and the weights and find the accuracy for test patterns.