EE769 Introduction to Machine Learning (Jan 2022 edition)
Electrical Engineering, Indian Institute of Technology Bombay
Programming Assignment – 3 : Deep Learning and Unsupervised Learning

Instructions:

a) Submit four ipython notebooks with file names <RollNo>_<i>.pynb, where i is the question number. The notebook should be a complete code plus report with copious comments, references and URLs, outputs, critical observations, and your reasoning to choose next steps.
b) Use good coding practices such as avoiding hard-coding, using self-explanatory variable names, using functions (if applicable). This will also be graded.
c) Cite your sources if you use code from the internet. Also clarify what you have modified. Ensure that the code has a permissive license or it can be assumed that academic purposes fall under 'fair use'.

Problem statements:

1. Convolutional Neural Networks:
   a. Copy and study the starter code (until "ConvNet as fixed feature extractor") given by Sasank (CTO of Qure.ai, pytorch contributor, and alumnus of IITB) for classifying ants vs. bees: https://pytorch.org/tutorials/beginner/transfer_learning_tutorial.html. The key feature of this code is that it does not train a model from scratch, but uses transfer learning of a ResNet-18 architecture that is pre-trained on a large dataset (ImageNet) and then only fine-tunes it for the problem at hand.
   b. Modify the code to run on co-lab without any new features. [1]
   c. Modify the code to plot validation loss and accuracy after every training epoch. [2]
   d. Change the learning rate, momentum, and number of epochs at least three times to see the net effect on final validation loss and accuracy and its speed of convergence. https://pytorch.org/docs/stable/optim.html [1]
   e. Introduce weight decay (L2 penalty on weights) and find a good value for the weight decay factor. [1]
2. Clustering:
   a. Visualize and pre-process the data as appropriate from the file DataClustering.csv. You might have to use a power, an exponential, or a log transformation. [1]
   b. Train k-means, and find the appropriate number of k. [1]
   c. Using the cluster assignment as the label, visualize the t-sne embedding. [1]
3. PCA:
   a. Visualize the data from the file DataPCA.csv. [1]
   b. Train PCA. [1]
   c. Plot the variance explained versus PCA dimensions. [1]
   d. Reconstruct the data with various numbers of PCA dimensions, and compute the MSE. [1]
4. Non-linear dimension reduction:
   a. Visualize the data from the file DataKPCA.csv. [1]
   b. Train KPCA. [1]
   c. Plot the variance explained versus KPCA dimensions for up to 10 dimensions. [1]