INTRODUCTION TO MACHINE LEARNING
Assignment 4
Deep Learning, Unsupervised Learning

Deadline: December 1st, 11:59 p.m. IST **Max points: 100**

1. **(Programming - Healthcare Resource Distribution Using K-means Clustering)** The World Health Organization is developing a strategic plan for distributing medical resources and establishing healthcare infrastructure. Using the provided dataset that contains healthcare indicators, demographic data, and infrastructure metrics, implement k-means clustering to categorize countries based on their healthcare needs and existing capabilities.

   Dataset features:

   - `child_mort`: Death of children under 5 years of age per 1000 live births
   - `health`: Total health spending as % of GDP
   - `income`: Net income per person
   - `inflation`: Annual growth rate of the Total GDP
   - `life_expec`: Average life expectancy at birth
   - `total_fer`: Fertility rate (children per woman)
   - `gdpp`: GDP per capita

   Tasks:

   (a) **Data Preparation:** [3 points]
      i. Load and examine the dataset structure
      ii. Plot histograms for each individual feature
      iii. Perform feature scaling/standardization (normalize mean to 0 and standard deviation to 1 for each feature)

   (b) **K-means Implementation:** [8 points]
      i. Implement k-means algorithm from scratch with k=4:
         - Initialize centroids randomly
         - Assign countries to nearest centroid using Euclidean distance
         - Update centroids by computing mean of assigned countries
         - Repeat until convergence or maximum iterations (100) reached
      ii. Run the algorithm with five random initializations
      iii. Implement and explain your convergence criteria

   (c) **Analysis:** [5 points]
      i. Create 2D plots for child_mort vs. health, income vs.life_expec, total_fer vs. gdpp, where each cluster is colour coded.
      ii. For each cluster, analyze and report:
         - Number of countries
         - Average values of key indicators
         - Characteristic features of the cluster

2. **(Development Aid Analysis Using PCA and Clustering)** The United Nations Development Programme (UNDP) is analyzing global development indicators to identify patterns and groupings among countries for more targeted development assistance. The provided dataset contains key socio-economic and health indicators for different countries:

   Dataset features:

   - `child_mort`: Death of children under 5 years of age per 1000 live births
   - `health`: Total health spending as % of GDP
   - `income`: Net income per person
   - `inflation`: Annual growth rate of the Total GDP
   - `life_expec`: Average life expectancy at birth
   - `total_fer`: Fertility rate (children per woman)
   - `gdpp`: GDP per capita

   The UNDP wants to explore whether dimensionality reduction techniques combined with clustering can reveal more intuitive patterns in the data.

   Tasks:

   (a) **Principal Component Analysis** [6 points]

      i. Implement PCA on the standardized dataset
         - Calculate the covariance matrix
         - Compute all the eigenvalues and eigenvectors
         - Sort the principal components by eigenvalue, in descending order

      ii. Compute the explained variance ratio for each component (recall that explained variance ratio for a component is the ratio between the corresponding eigenvalue and the sum of all eigenvalues)

      iii. What is the minimum number of principal components needed to explain at least 80% of the variability?

   (b) **2D Analysis and Visualization** [6 points]

      i. Create scatter plots of the first two principal components

      ii. Apply k-means (k=4) on the 2D reduced data

      iii. Show a 2D plot for the dimension-reduced data, colour coded by cluster assignment.

   (c) **3D Analysis and Comparison** [6 points]

      i. Create pairwise scatter plots of the first three principal components

      ii. Apply k-means (k=4) on the 3D reduced data

      iii. Show pairwise 2D plots for the dimension-reduced data, colour coded by cluster assignment.

      iv. Compare clustering results across:
         - Original high-dimensional clustering
         - 2D PCA clustering
         - 3D PCA clustering

   Requirements:

   - Use Python with numpy, sklearn, and appropriate visualization libraries

- Include clear documentation and comments in your code
- Provide visualizations that effectively communicate the results
- Include error handling in your implementation
- Discuss the practical implications of your findings for development aid allocation

3. **(Classification with Neural Networks on MNIST)** Implement a neural network classifier using the MNIST dataset provided in IDX format. The goal is to build a classifier to distinguish between digits.

   Dataset Files:

   - Training images: `train-images.idx3-ubyte`
   - Training labels: `train-labels.idx1-ubyte`
   - Test images: `t10k-images.idx3-ubyte`
   - Test labels: `t10k-labels.idx1-ubyte`

   Tasks:

   (a) **Data Loading and Preprocessing** [6 points]

      i. Implement functions to read IDX format files:
         - Parse the IDX file header (magic number and dimensions)
         - Load image data as numpy arrays (784 dimensions per image)
         - Load corresponding labels

      ii. Create a binary classification task by:
         - Selecting images of two digits (e.g., 0 and 1)
         - Converting labels to binary format

      iii. Split training data into training (80%) and validation (20%) sets

   (b) **Neural Network Implementation** [10 points]

      i. Implement a one-hidden-layer neural network with:
         - Input layer: 784 units + bias
         - Hidden layer: Choose between {100, 200, 300} units + bias
         - Output layer: 2 units (softmax activation)

      ii. Initialize weights using uniform distribution $U(-\sqrt{\frac{6}{V_{l-1}+V_l}}, \sqrt{\frac{6}{V_{l-1}+V_l}})$

      iii. Implement forward propagation and backpropagation

      iv. Use cross-entropy loss with L_2 regularization:

$$L = -\frac{1}{N}\sum(y_i \log(\hat{y}_i)) + \frac{\lambda}{2}\sum(w^2)$$

   (c) **Training and Optimization** [5 points]

      i. Implement batch gradient descent

      ii. Train with following hyperparameters:
         - Learning rates $\alpha \in \{0.01, 0.05, 0.1\}$
         - Regularization parameter $\lambda = 0.4$
         - Maximum iterations = 500

      iii. Plot learning curves (training and validation loss vs. iterations)

      iv. Implement early stopping using validation loss

(d) **Evaluation and Analysis** [3 points]

i. Report final accuracy on:
   - Training set
   - Validation set
   - Test set (t10k files)

ii. Provide five examples each of correctly and incorrectly classified digits

iii. Analyze the impact of learning rate and number of hidden layers on model performance

4. **(Neural Network Regression for Airfoil Self-Noise Prediction)** Using the provided Airfoil Self-Noise dataset, develop a neural network regression model to predict the Scaled Sound Pressure Level (SSPL) based on the input features. Your solution should address the following requirements:

   (a) **Data Preprocessing** [4 points]
   - Load the dataset
   - Perform feature scaling/normalization
   - Split the data into training, validation, and test sets (60-20-20 split)

   (b) **Neural Network Architecture** [8 points]
   - Design a neural network architecture with 2 hidden layers of 300 units each
   - Implement sigmoid activation functions
   - Include regularization (L_2) in the model to prevent overfitting

   (c) **Model Training** [7 points]
   - Implement the training loop as per your selected training set
   - Apply early stopping based on validation loss
   - Plot training and validation loss curves

   (d) **Model Evaluation** [3 points]
   - Evaluate the MSE of your model
   - Create a 2D plot of predicted values vs. actual values. What should the plot look like if your predictions have high accuracy?

5. **(Programming - Expectation-Maximization for mixture of Gaussians)** Consider the following data points from a mixture of two univariate Gaussian distributions.

$$
\begin{array}{cccccccccc}
-0.39 & 0.12 & 0.94 & 1.67 & 1.76 & 2.44 & 3.72 & 4.28 & 4.92 & 5.53 \\
0.06 & 0.48 & 1.01 & 1.68 & 1.80 & 3.25 & 4.12 & 4.60 & 5.28 & 6.22
\end{array}
$$

   Implement the Expectation-Maximization Algorithm to find the maximum likelihood estimates of the means and variances of the two Gaussians. [20 points]