Weekly Lab Assignment – II

Course Code: CS16201/CS18223/CS22203

1. Design a Multi-layer perceptron to make a binary decision on the following datasets. Use 70% of the dataset as training set and 30% as test set.
    a. Adult Census Income dataset: https://www.kaggle.com/datasets/uciml/adult-census-income/ - Predict whether the annual income of the person is >=50K or <50K
    b. Indian Liver Patient Records dataset: https://www.kaggle.com/datasets/uciml/indian-liver-patient-records - Predict whether person needs to be diagnosed or not?
    c. Titanic: Machine Learning from Disaster dataset: https://www.kaggle.com/c/titanic/data - Predict survival on the Titanic

2. Implement a Multi-Layer Perceptron (MLP) to classify handwritten digits from the MNIST dataset. However, restrict the implementation to not use any deep learning library such as TensorFlow or PyTorch, except for data loading. Specifically, you can use libraries or utilities to load and preprocess the MNIST dataset, but all aspects of building and training the neural network should be implemented without relying on external deep learning libraries. Experiment with different architectures by varying the number of hidden layers and neurons to observe their effects on classification performance.
3. Implement an MLP using a neural network library (such as TensorFlow or PyTorch) to classify handwritten digits from the MNIST dataset. Experiment with different numbers of hidden layers and neurons.
4. Modify the MLP from the previous question to use different activation functions (ReLU, Sigmoid, Tanh) in the hidden layers. Evaluate and compare the performance of each model on the same dataset (MNIST).
5. Explore the effect of different learning rates and optimization algorithms (SGD, Adam, RMSprop) on the training performance of an MLP. Use a dataset like CIFAR-10 for image classification. Comment on the generalization gap between the training and test accuracy.
6. Design a set of experiments to perform hyperparameter tuning on an MLP model. Focus on parameters such as the number of hidden layers, number of neurons per layer, learning rate, and batch size. Use tools like GridSearchCV or a library-specific solution for hyperparameter optimization.