Assignment 05 : Neural Regression (Iris Dataset)

UTA027 : Artificial Intelligence

Raghav B. Venkataramaiyer

Mar '25

Assignment 05: Multi-Layer Neural Network on the Iris Dataset Using PyTorch
(07-Apr to 18-Apr)

1 Objective

The goal of this assignment is to implement a multi-layer neural network (MLP) using PyTorch to classify the Iris dataset.

2 Question

Design and implement a multi-layer neural network using PyTorch to classify the Iris dataset. Your implementation should follow these steps:

2.1 Dataset Preparation

- Load the Iris dataset using sklearn.datasets.load$_{\text{iris}}$.
- Convert the dataset into PyTorch tensors.
- Split the dataset into training and test sets (e.g., 80% training, 20% testing).
- Normalize the feature values.

2.2 Build the Neural Network Model

- Implement an MLP with PyTorch using `torch.nn.Module`.

- The model should have:
  - An input layer with 4 neurons (one for each feature).
  - At least one hidden layer with ReLU activation.
  - An output layer with 3 neurons (one for each class) and softmax activation.

2.3 Train the Model

- Define the loss function (CrossEntropyLoss).
- Choose an optimizer (e.g., Adam or SGD).
- Train the model for a fixed number of epochs (e.g., 100 epochs).
- Track the loss during training.

2.4 Evaluate the Model

Compute accuracy on the test set. Generate a confusion matrix to visualize performance.

3 Theory

- Neuron and it application in Regression/ Classification For Latex

4 Boilerplate Code

- How to create a neural network module: + Python Notebook [Click here].
- How to create a pytorch dataset from tensors: + Python Notebook [Click here].

5 Evaluation Criterion

This assignment shall be implemented by students and evaluated by the instructor in lab.