Neural Networks and Deep Learning

Lab Exercises.

*At least 10 exercises should be completed. Make necessary assumption wherever required.

Lab 1: Python and Deep Learning Libraries Setup

- **Objective:** Set up Python, Jupyter Notebook, and install popular deep learning libraries (NumPy, TensorFlow, PyTorch).
- **Tasks:**

  a. Install Anaconda, create a virtual environment.

  b. Install TensorFlow, PyTorch, and Keras.

  c. Write a simple program to confirm installations.

Lab 2: Introduction to NumPy for Deep Learning

- **Objective:** Understand the basics of NumPy for tensor manipulation.
- **Tasks:**

  a. Create and manipulate arrays.

  b. Perform matrix multiplication, transpose, and reshaping.

Lab 3: Implementing a Perceptron

- **Objective:** Understand and implement a simple Perceptron model from scratch.
- **Tasks:**

  a. Initialize weights and bias.

  b. Implement the activation function (sign function).

  c. Train the Perceptron on a linearly separable dataset.

Lab 4: Adaline and Gradient Descent

- **Objective:** Implement Adaptive Linear Neuron (Adaline) using gradient descent.
- **Tasks:**

  a. Initialize weights and bias.

  b. Use a continuous loss function (MSE).

c. Implement gradient descent for weight updates.

Lab 5: Building a Feedforward Neural Network (from scratch)

- **Objective:** Manually build an MLP with one hidden layer.
- **Tasks:**

a. Initialize weights and biases.

b. Implement forward propagation.

c. Apply activation functions (ReLU, Sigmoid).

d. Implement backpropagation for weight updates.

Lab 6: Building an MLP using PyTorch

- **Objective:** Use PyTorch to quickly build an MLP.
- **Tasks:**

a. Use nn.Module to create a model.

b. Train the model using an optimizer (SGD).

c. Evaluate model performance.

Lab 7: Visualizing Gradient Descent

- **Objective:** Understand gradient descent with visualization.
- **Tasks:**

a. Plot a 2D loss surface.

b. Visualize gradient descent steps.

Lab 8: Exploring Optimizers

- **Objective:** Explore different optimization algorithms (SGD, Adam, RMSprop).
- **Tasks:**

a. Implement each optimizer manually.

b. Compare their convergence on a sample problem.

Lab 9: Understanding Convolution Operations

- **Objective:** Manually implement convolution and pooling.

- **Tasks:**
  a. Create a 2D image matrix.
  b. Apply convolution with a kernel.

Lab 10: Building a CNN with PyTorch

- **Objective:** Implement a simple CNN model.
- **Tasks:**
  a. Use nn.Conv2d for convolution.
  b. Train on the MNIST dataset.

Lab 11: Building a Simple RNN

- **Objective:** Manually implement an RNN.
- **Tasks:**
  a. Initialize hidden states.
  b. Implement recurrent connections.

Lab 12: LSTM with PyTorch

- **Objective:** Use PyTorch to implement an LSTM model.
- **Tasks:**
  a. Train on a text sequence.
  b. Visualize hidden state dynamics.