Assignment 01: Neural Architectures & Advanced Sequence Modeling

Course: Machine Learning

DATASET GUIDELINES

- **Classification:** MNIST (digits), Fashion-MNIST, or CIFAR-10.
- **Regression:** California Housing or simulated function data.
- **Sequential/Time-Series:** Stock market data, weather data.

PART 1: NEURAL NETWORKS & DEEP LEARNING

1. Multilayer Perceptron (MLP)

- **Task:** Build a dense neural network (MLP) to classify images from the `Fashion-MNIST` dataset.
- **Requirements:**
  - Implement the network using a framework like PyTorch, TensorFlow, or Keras.
  - Experiment with at least two distinct topologies (e.g., 2 hidden layers vs. 4 hidden layers).
  - Compare the performance of **ReLU** vs. **Sigmoid** activation functions in the hidden layers.
  - **Deliverable:** A plot showing *Training Loss vs. Validation Loss* over epochs.

2. Recurrent Neural Network (RNN)

- **Task:** Predict the future value of a time series.
- **Requirements:**
  - Use a Time Series dataset (e.g., daily temperature or stock prices).
  - Implement a standard **Simple RNN** or **LSTM**.
  - Train the model to predict the value at time $t + 1$ given input sequence $t - 10$ to t.
  - **Deliverable:** A plot comparing the ”Actual” time series vs. ”Predicted” values on the test set.

3. Self-Organizing Map (SOM)

- **Task:** Use a SOM to map high-dimensional data into a 2D grid.
- **Requirements:**
  - Use the **Iris dataset** or a color dataset (RGB values).
  - Train a SOM to group similar data points together.
  - **Deliverable:** A visualization of the resulting 2D map (U-Matrix or distance map) showing clusters.

PART 2: ADVANCED SEQUENCE & GENERATIVE MODELS

1. **Hidden Markov Model (HMM)**
    - **Task:** Decode hidden states from observed data.
    - **Requirements:**
        - Create or use a dataset representing a hidden state process (e.g., ”Fair” vs. ”Loaded” die, or Weather affecting Activity).
        - Use the **Viterbi Algorithm** to find the most likely sequence of hidden states.
        - **Deliverable:** Print the predicted sequence of hidden states and the transition matrix used.
2. **Large Language Model (LLM)**
    - **Task:** Leverage a pre-trained Transformer model for a specific NLP task.
    - **Requirements:**
        - Select a pre-trained model (e.g., **GPT-2**, **BERT**, or **T5**) using the Hugging Face `transformers` library.
        - Choose one option:
            - ∗ *Option A (Generation):* Tweak parameters like `temperature` and `top_k` to observe changes in creativity.
            - ∗ *Option B (Classification):* Fine-tune a small BERT model to classify sentiment.
        - **Deliverable:** Input/output logs and a brief paragraph explaining the effect of the parameters adjusted.

Submission Requirements:

1. **Source Code:** A single `.ipynb` file organized by question numbers.
2. **Visualizations:** All plots must have titles, labeled axes (x and y), and legends.
3. **Analysis:** A short summary text block after each result interpreting the findings.