

I. Scikit-learn:

- Load dataset (MNIST, FASHION MNIST) bằng cách sử dụng hàm **fetch_openml** từ thư viện scikit-learn.
- Chia dữ liệu thành training và testing bằng hàm **train_test_split**.
- Khởi tạo MLPClassifier với `hidden_layer_sizes = (100,50)` xác định mạng neural 3 layer với 100 nodes trong lớp ẩn thứ nhất và 50 nodes ở lớp ẩn thứ hai. (trong scikit-learn Python library, được sử dụng để phân loại dữ liệu. MLPClassifier là viết tắt của Multi-layer Perceptron Classifier. Nó dựa trên kiến trúc mạng neuron đa tầng (MLP) để phân loại dữ liệu, trong đó mỗi tầng có nhiều neuron kết nối với các neuron ở tầng liền kề.) `max_iter = 100` (giới hạn số lần lặp là 100 để tăng tốc quá trình training)
- Train dữ liệu bằng `fit` method và đánh giá bằng `score` method (trả về độ chính xác của mô hình trên dữ liệu test).
- Accuracy:
 - trên tập MNIST là 0.968.
 - trên tập Fashion-MNIST 0.887.

⇒ Đây là một con số khá tốt đối với một kiến trúc mạng nơ-ron 3 tầng đơn giản. Tuy nhiên, có thể tăng số lớp ẩn để cải thiện độ chính xác.

II. Tensorflow:

- Load dataset bằng hàm `.load_data()` từ TensorFlow.
- Chuẩn hóa dữ liệu, scale xuống `[0,1]` (chia pixel cho 255).
- Define model architecture sử dụng `tf.keras.Sequential` và thêm 4 layers:
 - `tf.keras.layers.Flatten`: flatten input images vào mảng 1D 784 pixels.
 - `tf.keras.layers.Dense`: 1000 nodes và ReLU activation.
 - `tf.keras.layers.Dense`: 50 nodes và ReLU activation.
 - `tf.keras.layers.Dense`: 10 nodes (1 cho mỗi class) và softmax activation.
- Sử dụng `'model.compile'` để biên dịch và train bằng `model.fit`, evaluate bằng `model.evaluate/`

- Accuracy:
 - MNIST: 0.975
 - Fashion MNIST: 0.883

III. Pytorch

- Xác định neural network architecture bằng nn.Module class và tạo một lớp NeuralNet với 3 fully connected layers, mỗi lớp có reLU activation và softmax activation trên output layer.
- Load dataset từ PyTorch dùng DataLoader.
- Xác định hàm mất mát và tối ưu hóa, đồng thời train nó bằng cách sử dụng vòng lặp trên train_loader data loader. Sau đó test neural network bằng cách lặp qua test_loader data loader, tính toán accuracy và in ra.
- Accuracy:
 - Fashion MNIST: 0.85
 - MNIST: 0.91

IV. Đánh giá:

Pytorch, tensorflow, scikit-learn đều có những điểm mạnh và điểm yếu riêng của nó. Trong thực nghiệm trên, cho thấy accuracy của Tensorflow là ổn định nhất trên cả 2 tập dataset. Nhưng accuracy không phải là mặt đánh giá duy nhất để biết được đâu là framework tốt nhất. Hiện nay PyTorch và TensorFlow được sử dụng nhiều nhất trong deep learning. Trong bài thực nghiệm, nếu đi vào thực hành em sẽ chọn tối ưu hóa PyTorch bằng nhiều cách như:

- Tăng số epochs: để mạng neural học được nhiều hơn từ data.
- Thêm nhiều hidden layers (tuy nhiên dễ dẫn tới overfitting).
- Sử dụng learning rate scheduler.
- Sử dụng data augmentation.