Assignment No. 3: Extending and Evaluating a Multi-layer Artificial Neural Network

- **Name:** Adir Serruya
- **Student ID:** 316472455
- **Submission Date:** 2/1/2025

1. Objective

- Extending the existing neural network implementation to include two hidden layers.
- Applying the extended model to classify handwritten digits from the MNIST dataset.
- Evaluating and comparing the performance with the original single hidden layer model and a PyTorch-based ANN implementation.

2. Methodology

1. Extending the Neural Network Architecture

- Initialization of weights and biases for the additional hidden layer.
- Forward propagation through two hidden layers.
- Backward propagation and gradient calculations for both hidden layers.
- Parameter update mechanisms.

2. Data Preparation

- Loading the dataset using fetch_openml.
- Converting labels to integers.
- Normalizing feature values using StandardScaler.
- Splitting the data into training (70%) and testing (30%) sets with stratification.

3. Implementation Details

- Sigmoid Activation Function
- One-hot encoding of labels for multi-class classification.
- Loss function: Cross Entropy Loss
- LR = 0.1

  Batchsize = 64

  Hidden L1 = 128,

  Hidden L2 = 64

  Epoch = 50

4. Training Process

- Forward pass computations.
- Loss calculation.
- Backward pass and gradient computation.
- Parameter updates using gradient descent and SGD on the pytorch version.

4. Results

1. Performance Metrics

- **Macro AUC Score:** Explain what Macro AUC represents and why it's a suitable metric for this task.

Include tables or charts to visualize the performance.

Model	Train Acc	Test Acc	Test AUC
Single Layer	0.9606	0.9401	
Two Layer	0.9646	0.9454	0.9938
Pytorch	0.9093	0.895	0.986

---

5. References

- Scikit-learn Documentation: https://scikit-learn.org
- PyTorch Documentation: https://pytorch.org
- MNIST Dataset: http://yann.lecun.com/exdb/mnist/