Experiment 1- Documentation

Name - Aditya Luthra

Roll no - 2K22/CO/30

---

1. Introduction

This Experiment demonstrates the implementation of a fully connected neural network (FCNN) to classify handwritten digits from the MNIST dataset. The notebook processes image and label data, reshapes the images, normalizes them, and implements a feedforward neural network with hidden layers . The model is trained using backpropagation and evaluated based on training and validation loss.

---

2. Libraries Used

The following Python libraries are used in this project:

- **numpy**: For numerical operations such as matrix manipulations and mathematical computations.
- **matplotlib.pyplot**: For visualizations, specifically displaying sample images from the MNIST dataset.
- **os**: For managing file paths.

---

3. Dataset Preparation

The MNIST dataset is used in this project, which contains 60,000 training images and 10,000 testing images of handwritten digits (28x28 pixels). The dataset is read from binary files using numpy and processed as follows:

- **Image files** are read as a byte stream, skipping header bytes, and reshaped into 28x28 images.
- **Labels** are processed into one-hot encoded vectors for classification.

The data is then split into training, validation, and test datasets, where the validation set is 20% of the training data.

---

4. Neural Network Model

A simple fully connected neural network is implemented. The architecture consists of:

- **Input Layer**: 784 neurons (for the 28x28 pixel values of each image).
- **Hidden Layer**: 128 neurons with ReLU activation.
- **Output Layer**: 10 neurons (for the 10 digit classes), with softmax activation for multi-class classification.
- **Loss Function**: Cross-entropy loss is used to compute the difference between predicted and true labels.
- **Optimizer**: Stochastic gradient descent (SGD) with a learning rate of 0.005 is used for training.

---

5. Model Training and Evaluation

The model is trained for 400 epochs. The performance is evaluated on both the training and validation datasets using loss and accuracy. The best model (lowest validation loss) is saved during training and loaded for final evaluation on the test dataset.

Training and validation losses are computed at each epoch, and the model's accuracy on the test set is reported after training is complete.

---

6. Results

The model achieved the following performance on the test set:

- **Test Accuracy** : 92.81%
- **Test Loss** : 0.5028302131428092

This experiment shows how a Neural Network is Implemented from scratch using Numpy .

7. Conclusion

This experiment demonstrates the implementation of a fully connected neural network for handwritten digit classification using Numpy . By preprocessing the MNIST dataset and implementing the training loop, the experiment shows the essential steps in neural network training, evaluation, and model saving/loading. Future steps could involve exploring more complex architectures or other optimization techniques to improve performance.

GitHub Link

https://github.com/Adiiii02/DL-LAB/tree/main/EXP-1