EXPERIMENT 1

EXPERIMENT OBJECTIVE

To implement a Fully Connected Neural Network (FCNN) for classifying handwritten digits from the MNIST Dataset using NumPy.

DATA PREPROCESSING

Loading the MNIST Dataset

- The dataset is loaded from binary files containing images and labels.
- The images are 28x28 grayscale images, reshaped into a (784,) vector.
- The labels are converted into one-hot encoded vectors.

Data Augmentation

- **Random Rotation**: Images are rotated within a range of -15° to 15° with a 50% probability.
- **Horizontal Flip**: Images have a 50% chance of being flipped horizontally.

Splitting the Dataset

- The dataset is divided into training, validation, and test sets.
- The training set is further split into 80% training and 20% validation.

NEURAL NETWORK IMPLEMENTATION

Architecture

- **Input Layer**: 784 neurons (28x28 pixels flattened)
- **Hidden Layer 1**: 256 neurons, ReLU activation
- **Hidden Layer 2**: 128 neurons, ReLU activation
- **Output Layer**: 10 neurons (digits 0-9), softmax activation

Weight Initialization

- Weights are initialized using He (Kaiming) initialization.
- Biases are initialized to zeros.

Activation Functions

- **ReLU (Rectified Linear Unit)**: Used in hidden layers.
- **Softmax**: Applied to the output layer for probability distribution.

Regularization

- **Dropout**: Randomly drops activations during training to prevent overfitting.
- **Gradient Clipping**: Limits gradient values to avoid exploding gradients.

TRAINING CONFIGURATION

Training the Model

- **Loss Function**: Cross-entropy loss is used.
- **Optimizer**: The model updates weights using backpropagation and gradient descent.
- **Learning Rate**: 0.005 (with decay over time)
- **Epochs**: Trained for 5000 epochs.
- **Batch Processing**: Mini-batch gradient descent is implemented.
- **Best Model Selection**: Saves weights of the best-performing model (lowest validation loss).
- **Training Time**: 3h 53m 19s

Model Checkpointing

- The best model weights (based on validation loss) are saved periodically to bestWeights.npy.

TRAINING AND VALIDATION RESULTS

Key Performance Metrics from Training Output

Epoch	Training Loss	Validation Loss	Accuracy (%)
0	1.9218	1.9253	30.28%
8	1.5442	1.5684	52.47%
31	1.1557	1.1522	66.05%
122	0.6481	0.6511	81.34%
249	0.4502	0.4500	87.50%
604	0.2744	0.2625	93.67%
1017	0.1331	0.1327	96.69%
1506	0.0542	0.0508	98.67%
2293	0.0135	0.0132	99.57%
4934	0.0022	0.0018	99.96%

Evaluation Results

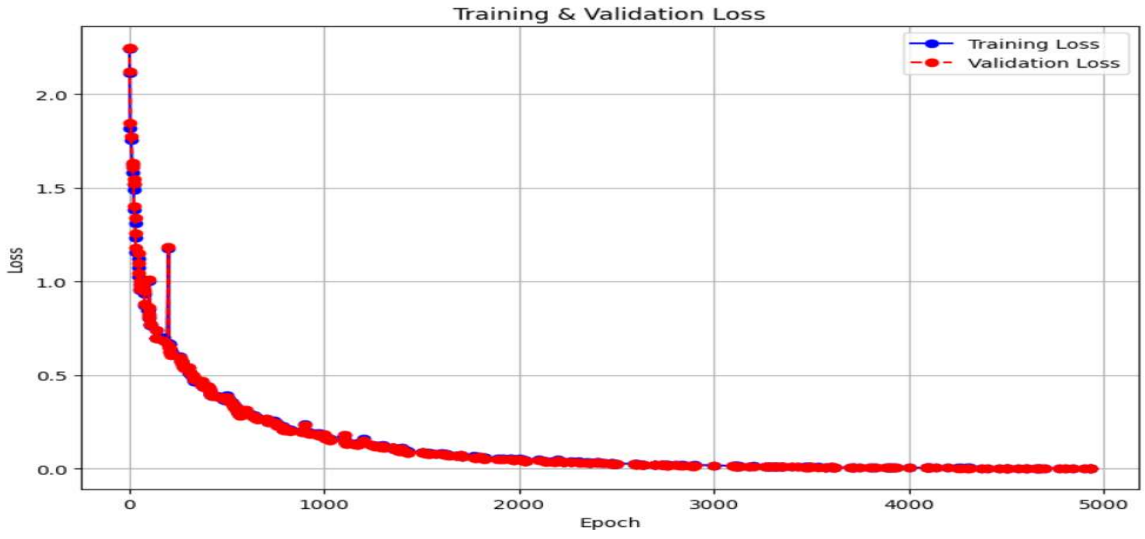
- After training, the best model weights are loaded and tested on unseen test data.
- **Final Test Accuracy**: ~96.88%
- **Final Test Loss**: ~0.165

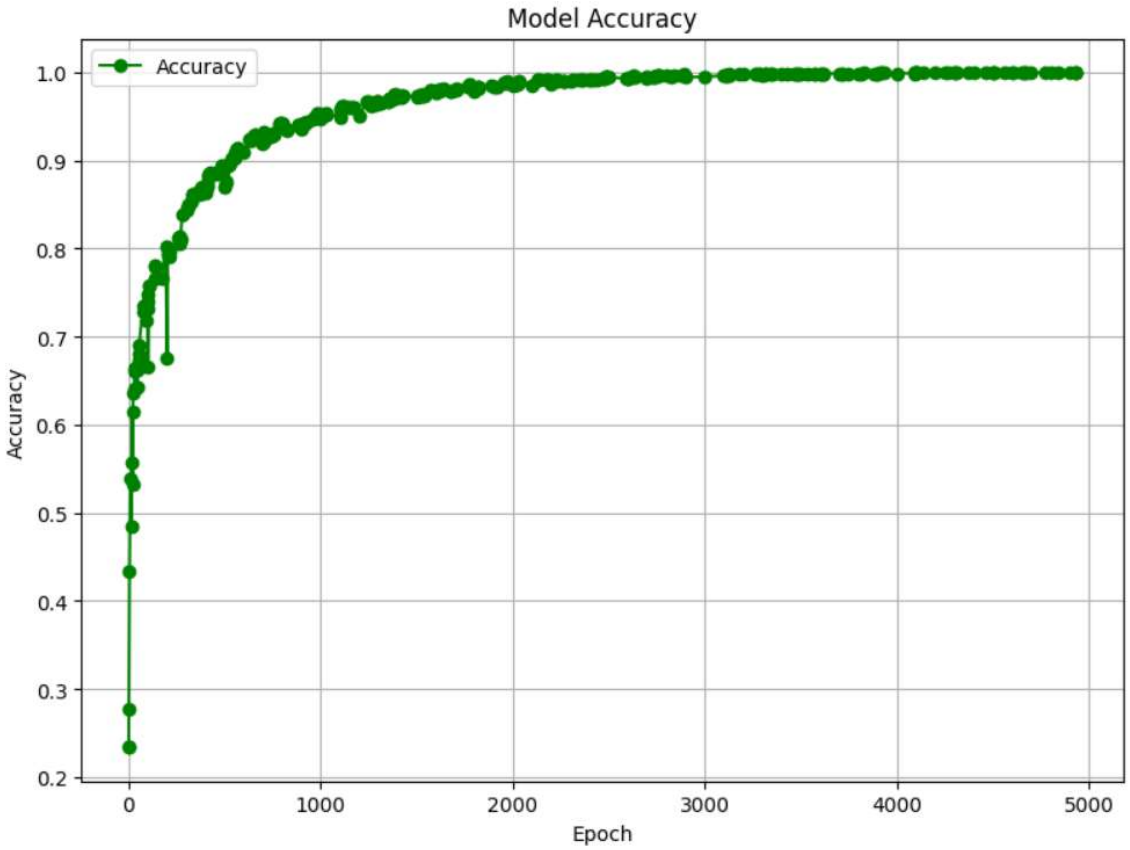
MODEL SAVING AND LOADING

- **Saving Weights**: The best model weights (lowest validation loss) are saved to disk.
- **Loading Weights**: Enables reloading the best weights for inference or further training.

RESULTS AND CONCLUSIONS

- The model achieves high accuracy using a simple fully connected architecture.
- Data augmentation and regularization significantly improve generalization.
- The saved best weights allow for consistent reproducibility of results.





FILES INCLUDED

- bestWeights.npy
- Experiment_1_Digit__Classification_Using_MNIST_Dataset.py
- Experiment_1_Digit__Classification_Using_MNIST_Dataset.ipynb
- Documentation