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**: 0.2 (with decay over time)

- **Epochs**: Trained for 2500 epochs.

- **Batch Processing**: Mini-batch gradient descent is implemented.

- **Best Model Selection**: Saves weights of the best-performing model (lowest validation loss).

Model Checkpointing

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

TRAINING AND VALIDATION RESULTS

Key Performance Metrics from Training Output

Epoch	Training Loss	Validation Loss	Accuracy (%)
0	2.1233	2.1250	28.85
8	1.6454	1.645	56.19
27	1.2247	1.2220	62.31
100	0.6292	0.6419	81.00
300	0.4599	0.4762	87.68
700	0.2343	0.2324	93.95
900	0.1772	0.1756	95.59
1100	0.1657	0.1642	96.14
2000	0.0206	0.0196	99.33
2492	0.0048	0.0048	99.81

Evaluation Results

- After training, the best model weights are loaded and tested on unseen test data.

- **Final Test Accuracy**: 95.38%

- **Final Test Loss**: 0.31934316092416865

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.

- Regularisation help this model to work better on new data .

- Saved best weight ensure consistent results.