Deep Learning Assignment 1 - MNIST Experiments

1. Introduction

This report documents the implementation and analysis of a feedforward neural network for the MNIST dataset. The goal is to train a neural network using backpropagation and evaluate its performance using different optimization techniques and hyperparameters.

2. Dataset Description

The MNIST dataset is a collection of 70,000 grayscale images of handwritten digits from 0 to 9, divided into 60,000 training images and 10,000 test images. Each image is 28x28 pixels in size.

3. Model Implementation

The feedforward neural network consists of:

Layer	***Description***
Input Layer	28x28 flattened into 784 neurons
Hidden Layers	Configurable (3, 4, or 5 layers with 32, 64, or 128 neurons)
Output Layer	10 neurons (Softmax activation for classification)

Activation functions used: ReLU, Sigmoid

4. Optimization Techniques

The following optimization methods were implemented:

Optimizer	***Description***
SGD	Standard stochastic gradient descent
Momentum	Uses momentum to accelerate convergence
Nesterov	Improves momentum by correcting overshoot
RMSprop	Adaptive learning rate for faster convergence
Adam	Combines momentum and adaptive learning rates

The primary loss function used was **Cross-Entropy Loss**. A comparison with **Squared Error Loss** was also conducted.

5. Hyperparameter Experiments

Various hyperparameters were tested to optimize model performance:

Hyperparameter	***Tested Values***	***Best Performing***
Epochs	5, 10, 30	10
Batch Size	32, 64	32
Learning Rate	0.001, 0.0001	0.001
Hidden Layers	3, 4, 5	4
Activation Function	ReLU, Sigmoid	ReLU

6. Results & Observations

Final accuracy results for the best models:

Model	***Optimizer***	***Test Accuracy***
Model 1	Adam	98
Model 2	RMSprop	97.5
Model 3	Adam	97.8

A confusion matrix was generated for the best model to analyze misclassification patterns.

Model 1 Architecture:

- Hidden Layers: [128, 64, 64, 64, 32]
- Output units: 10
- Output activation: SoftMax
- Activation: ReLU
- Optimizer: Adam
- Learning Rate: 0.001
- Kernel Initialization: Xavier
- Batch size: 32
- Accuracy: 98%

Model 2 Architecture:

- Hidden Layers: [128, 64, 64, 64, 32]
- Output units: 10
- Output activation: SoftMax
- Activation: ReLU
- Optimizer: RMSprop
- Learning Rate: 0.001
- Kernel Initialization: Xavier
- Batch size: 64
- Accuracy: 97.5%

Model 3 Architecture:

- Hidden Layers: [128, 64, 64, 32]
- Output units: 10
- Output activation: SoftMax
- Activation: ReLU
- Optimizer: Adam

- Learning Rate: 0.001
- Kernel Initialization: Xavier
- Accuracy: 97.8%

7. Conclusion

Based on the experiments, the following recommendations were made for the best performance on the MNIST dataset:

• Use ReLU activation for faster convergence.

• Adam optimizer provides stable and high accuracy.

• A learning rate of 0.001 with batch size 32 works well for training.

These findings are based on extensive experimentation and comparison of various hyperparameter settings.