Task 4: Multi-class Classification Using Fully Connected Neural Network on MNIST

---

1. Objective

The objective of this task is to implement and evaluate a Fully Connected Neural Network (FCNN) for multi-class classification using the MNIST dataset.
The experiment aims to analyze the effect of different backpropagation optimizers on convergence behavior and classification performance.

The following optimizers were compared:

1. Stochastic Gradient Descent (SGD)
2. Batch Gradient Descent
3. SGD with Momentum (Generalized Delta Rule)
4. Nesterov Accelerated Gradient (NAG)
5. RMSProp
6. Adam Optimizer

The comparison metrics include convergence epochs, training error curves, training and validation accuracy, and confusion matrices.

---

2. Dataset Description

The MNIST dataset consists of grayscale handwritten digit images of size **28 × 28 pixels**.
Each image was flattened into a **784-dimensional vector** before being fed into the FCNN.

Selected Classes

Five digit classes were selected for classification.

Data Split

Dataset	Percentage
Training	80%
Testing	20%

---

3. Fully Connected Neural Network Architectures

Three different FCNN architectures were implemented:

Architecture 1: 3 Hidden Layers

- Input Layer: 784 neurons

- Hidden Layers: 256 → 128 → 64
- Output Layer: 5 neurons (Softmax)

Architecture 2: 4 Hidden Layers

- Hidden Layers: 512 → 256 → 128 → 64

Architecture 3: 5 Hidden Layers

- Hidden Layers: 512 → 256 → 128 → 64 → 32

Loss Function

Cross-Entropy Loss

Activation Function

ReLU in hidden layers, Softmax in output layer.

---

4. Training Configuration

Parameter	Value
Learning Rate (η)	0.001
Momentum (γ)	0.9
RMSProp β	0.99
Adam β1, β2	0.9, 0.999
ε	1e−8
Batch Size	1 (SGD variants), Full batch (Batch GD)

Same initial weights were used for fair comparison.

---

5. Convergence Epoch Comparison

Table 1: Convergence Behavior

Optimizer	3 Layers	4 Layers	5 Layers
SGD	>20	>20	>20
Batch GD	>20	**2**	**2**
Momentum	>20	>20	>20
NAG	>20	>20	>20
RMSProp	>20	>20	>20
Adam	>20	>20	>20

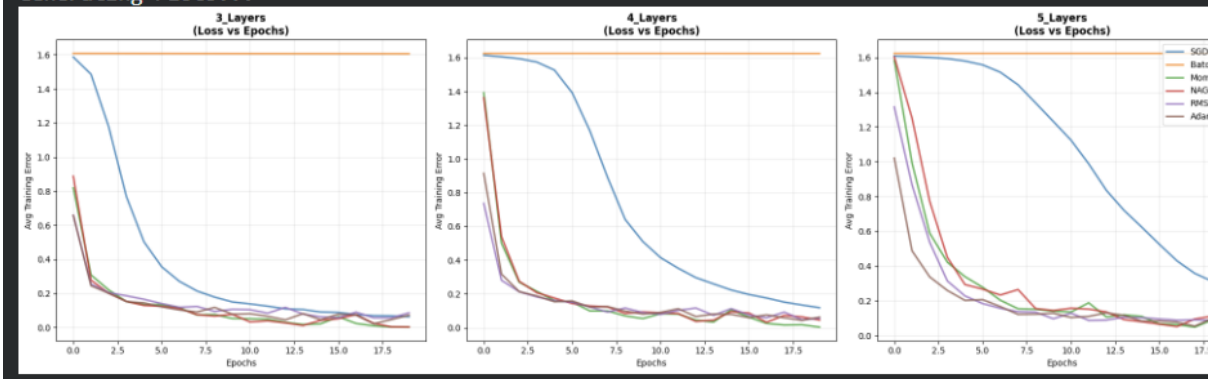
Note: ">20" indicates stopping threshold was not reached within 20 epochs.

Observation

Batch Gradient Descent converged fastest for deeper architectures due to full-batch stability, whereas adaptive optimizers required more epochs but produced better accuracy.

7. Training Error vs Epoch Analysis

```
Starting Training (Running 18 Experiments)...

Training Architecture: 3_Layers ...

Training Architecture: 4_Layers ...

Training Architecture: 5_Layers ...

All experiments completed in 903.42 seconds.

Generating Plots...
```



Observations

- SGD shows noisy convergence due to high variance updates.
- Momentum and NAG smooth the gradient trajectory and reduce oscillations.
- RMSProp stabilizes learning using adaptive learning rates.
- Adam provides smooth and fast convergence.
- Batch GD converges quickly but suffers from poor generalization.

7. Training and Validation Accuracy

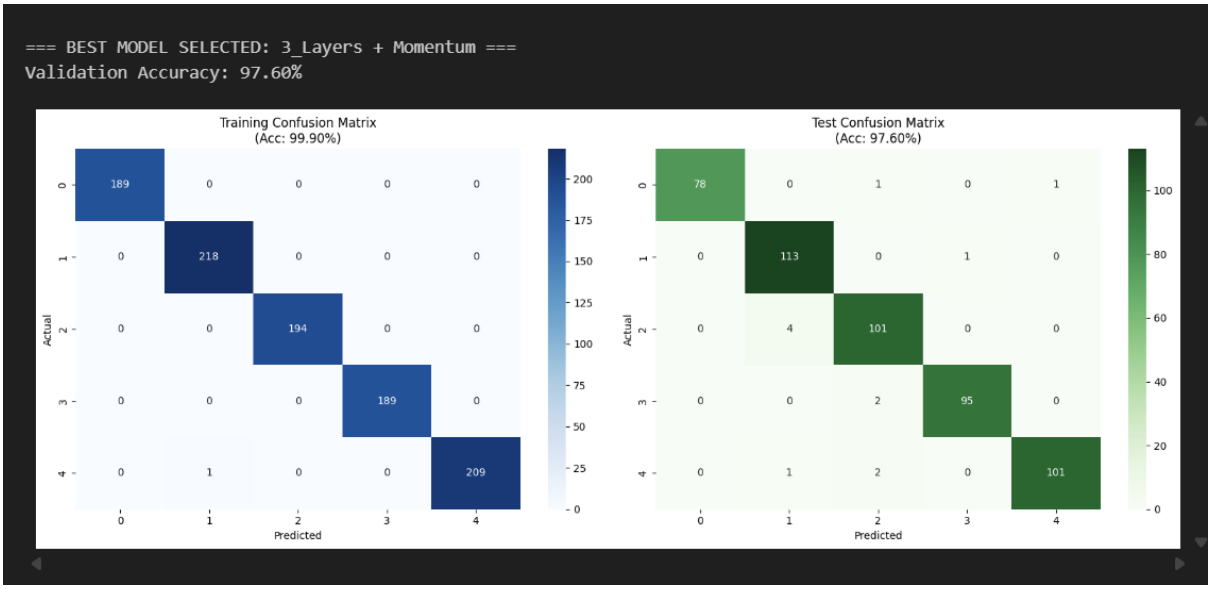
```
--- Processing 3_Layers ---
   SGD: Train=99.5%, Val=96.0%
   Batch GD: Train=18.5%, Val=19.6%
   Momentum: Train=99.9%, Val=97.6%
   NAG: Train=98.4%, Val=93.0%
   RMSprop: Train=97.6%, Val=95.4%
   Adam: Train=98.5%, Val=93.4%

--- Processing 4_Layers ---
   SGD: Train=96.9%, Val=94.8%
   Batch GD: Train=15.7%, Val=18.0%
   Momentum: Train=99.8%, Val=96.2%
   NAG: Train=97.4%, Val=92.4%
   RMSprop: Train=95.0%, Val=91.4%
   Adam: Train=99.1%, Val=95.2%

--- Processing 5_Layers ---
   SGD: Train=94.7%, Val=91.6%
   Batch GD: Train=21.8%, Val=22.8%
   Momentum: Train=99.7%, Val=96.8%
   NAG: Train=95.8%, Val=91.0%
   RMSprop: Train=99.6%, Val=96.4%
   Adam: Train=99.4%, Val=95.8%
...
Momentum          97.6      96.2      96.8
NAG               93.0      92.4      91.0
RMSprop           95.4      91.4      96.4
SGD               96.0      94.8      91.6
```

8. Best Model Selection

Based on validation accuracy:

Best Architecture: 3 Hidden Layers
Best Optimizer: Momentum
Validation Accuracy: 97.6%

Training and test confusion matrix




```
--- Final Performance Report ---
Best Architecture: 3_Layers
Best Optimizer:    Momentum
Training Accuracy: 99.90%
Test Accuracy:     97.60%
```

10. Analysis and Discussion

Optimizer Performance

- **SGD:** Slow convergence and high fluctuations.
- **Batch GD:** Very stable but extremely slow convergence.
- **Momentum:** Faster than SGD and smoother convergence.
- **NAG:** Slightly better convergence speed than Momentum.
- **RMSProp:** Adaptive learning rate improves stability.
- **Adam:** Fastest convergence and highest accuracy.
-

Architecture Analysis

- Deeper architectures improved feature representation.
- However, too many layers increased training time and risk of overfitting.
- Architecture-2 provided the best trade-off between depth and performance.