**Experiment No. 3**

**Title:** Using a dataset of your choice, train a neural network model with various combinations of learning rates, batch sizes, and optimizers. Document the impact of these changes on model accuracy and training time.

Training a neural network involves tuning several hyperparameters such as learning rate, batch size, and optimizer choice, which can significantly impact both the accuracy of the model and the time it takes to train. Take an example using a standard dataset like the MNIST handwritten digits’ dataset and varying these parameters to observe their effects.

### **Dataset and Model:**

We'll use the MNIST dataset, which consists of 28x28 grayscale images of handwritten digits (0-9). Our task is to classify these digits.

### Hyperparameters to Explore:

1. **Learning Rates**: Typically ranges from 0.001 to 0.1.
2. **Batch Sizes**: Commonly used sizes are 32, 64, 128, 256.
3. **Optimizers**: Options include SGD, Adam, RMSprop, etc.

### Model Architecture

We'll use a simple feedforward neural network with two hidden layers (ReLU activation) and a softmax output layer.

### Experimentation:

Now, let's vary the hyperparameters:

1. **Learning Rates**: Try 0.001, 0.01, 0.1.
2. **Batch Sizes**: Try 32, 64, 128.

**Implementation:**

Here is simplified version of the code to implement the experiments:

A screenshot of a computer program

Description automatically generatedA screenshot of a computer code

Description automatically generatedA screenshot of a computer code

Description automatically generated

Results Analysis:

Impact on Model Accuracy

Learning Rate:

Lower learning rates (0.001) generally lead to better convergence but may take longer to train.

Higher learning rates (0.1) can cause the model to diverge or oscillate.

Batch Size:

Smaller batch sizes (32) tend to provide better generalization but increase training time.

Larger batch sizes (128) speed up training but may lead to overfitting.

Optimizer:

Adam optimizer often yields the best accuracy due to its adaptive learning rate capabilities.

SGD may perform well but requires careful tuning of the learning rate.

Impact on Training Time

Learning Rate:

Higher learning rates can reduce training time but may compromise accuracy.

Batch Size:

Larger batch sizes reduce the number of weight updates per epoch, leading to faster training times.

Optimizer:

Adam is generally slower per epoch than SGD but can converge in fewer epochs overall.

### **Conclusion**

Through these experiments, we'll gain insights into how different combinations of learning rates, batch sizes, and optimizers affect both the accuracy and training time of our neural network model. Fine-tuning these parameters is crucial for achieving optimal performance in real-world applications.