HYPERPARAMETER TUNING FOR MNIST CLASSIFICATION WITH NEURAL NETWORKS

COVER PAGE

Faculty of Computers and Artificial Intelligence

Course: Supervised Learning (Spring 2025)

Team Members:

- Mohamed Hisham (Team Leader) – ID: 20220310
- Mohamed Islam – ID: 20220282
- Ali Mohamed Bahey – ID: 20220211
- Amr Mostafa – ID: 20220238
- Nour Shaaban – ID: 20220361

INTRODUCTION

This project focuses on the systematic exploration of hyperparameters in Artificial Neural Networks (ANN) and Convolutional Neural Networks (CNN) applied to the MNIST dataset, a benchmark dataset of handwritten digit images. The main objective is to investigate how varying key hyperparameters influences model performance, aiming to identify the optimal configuration that maximizes accuracy while maintaining computational efficiency.

The hyperparameters studied include the number of layers (both convolutional and fully connected), activation functions, optimizers, batch sizes, dropout rates, and learning rates. By methodically tuning these parameters, we seek to understand their impact on training dynamics and generalization ability.

Through a comparative analysis between ANN and CNN architectures, the project aims to demonstrate the effectiveness of deep learning models in image classification tasks and to highlight the most effective strategies for model optimization. This report presents experimental results that validate

the tuning methodology and provide insights into best practices for neural network hyperparameter selection on the MNIST dataset.

DATASET AND PREPROCESSING

The MNIST dataset is a widely-used benchmark in image classification consisting of 70,000 grayscale images of handwritten digits (0–9). Each image has a fixed input shape of **28x28 pixels**, representing a single channel (grayscale). The dataset is divided into a training set of 60,000 images and a test set of 10,000 images, supporting the supervised learning framework for evaluating model generalization.

Preprocessing the MNIST data is essential to enhance model performance and ensure compatibility with different architectures. First, pixel intensities are normalized to the range [0, 1] by dividing by 255, which accelerates convergence and stabilizes training. Labels are one-hot encoded to enable multi-class classification using categorical loss functions.

Data reshaping is performed according to model requirements: for Artificial Neural Networks (ANN), images are flattened into vectors of shape 784 × 1, while for Convolutional Neural Networks (CNN), the 2D spatial structure is preserved by reshaping inputs to 28 × 28 × 1, maintaining the channel dimension. This distinction allows CNNs to exploit spatial locality, while ANNs operate on vectorized pixel values.

These preprocessing steps ensure the data is standardized and formatted correctly for systematic hyperparameter tuning experiments across ANN and CNN models.

MODEL EXPERIMENTS AND ANALYSIS

SVM BASELINE MODEL

As a preliminary benchmark, a Support Vector Machine (SVM) model with a radial basis function (RBF) kernel was trained on a subset of 15,000 images from the MNIST training set. The hyperparameter C was set to 5, balancing margin maximization and empirical risk minimization.

The SVM achieved an accuracy of **94.8%** on the test set, demonstrating reasonable performance despite the reduced training data size. However, this approach exhibited significant computational cost when scaling to the full training dataset, limiting its practicality for high-volume training scenarios.

The result validated the necessity of exploring neural network architectures for improved scalability and performance.

ARTIFICIAL NEURAL NETWORK (ANN) MODELS

Multiple ANN models were constructed to assess the impact of hidden unit quantity, optimizer choice, and learning rate on classification accuracy. The architecture consisted of fully connected layers, using flattened MNIST data inputs.

Model	Hidden Units	Optimizer	Learning Rate	Accuracy	Loss
ANN-1	64	SGD	0.01	96.12%	0.132
ANN-2	128	SGD	0.01	96.45%	0.125
ANN-3	128	Adam	0.001	**97.66%**	0.104
ANN-4	256	RMSprop	0.001	97.21%	0.112
ANN-5	128	SGD	0.1	95.89%	0.148

Key observations from these experiments include:

- The Adam optimizer outperformed SGD and RMSprop, achieving the highest accuracy in the ANN group at 97.66% with a learning rate of 0.001.
- Increasing the number of hidden units from 64 to 128 provided a noticeable performance improvement, while further increasing to 256 units did not yield a proportional accuracy gain.
- High learning rates, such as 0.1 for SGD, caused unstable training and lower accuracy, indicating the importance of careful tuning of this hyperparameter.

CONVOLUTIONAL NEURAL NETWORK (CNN) MODELS

The bulk of experimentation focused on CNN architectures, given their superior ability to capture spatial features in image data. Over 18 CNN models were evaluated, systematically varying key hyperparameters and architectural components.

1. Epochs Variation

Model	Epochs	Accuracy	Avg Time per Epoch
CNN-1	10	98.12%	12.4s
CNN-2	15	**98.45%**	12.6s
CNN-3	5	97.67%	12.1s

Training for 15 epochs gave the best balance between accuracy and training duration, with gains over 10 epochs evident but plateauing thereafter.

2. Learning Rate Exploration

Model	Learning Rate	Accuracy
CNN-4	0.001	97.89%
CNN-5	0.1	**98.73%**

A notably higher learning rate of 0.1 with SGD optimizer led to the best CNN accuracy, emphasizing different optimization dynamics compared to ANN models.

3. Layer Configuration

Model	Conv2D Layers	Fully Connected Layers	Accuracy
CNN-6	2	1	98.21%
CNN-7	2	2	98.56%
CNN-8	2	3	**98.73%**

Increasing the number of fully connected layers up to three while maintaining two convolutional layers led to peak performance, balancing expressive power and overfitting risk.

4. Batch Size Tuning

Model	Batch Size	Accuracy
CNN-9	128	98.52%
CNN-10	192	98.34%

Smaller batch sizes of 64 (used in other experiments) generally outperformed larger sizes, suggesting finer gradient updates enhance generalization.

5. Activation Functions

Model	Activation	Accuracy
CNN-11	Sigmoid	97.12%
CNN-12	Tanh	98.01%
CNN-13	ReLU6	98.23%
Best	**ReLU (default)**	**98.73%**

The default ReLU activation function yielded the highest accuracy, confirming its widespread suitability for CNNs in digit classification.

6. Optimizer Comparison

Model	Optimizer	Accuracy
CNN-14	Adam	97.89%
CNN-15	RMSprop	97.67%
Best	**SGD**	**98.73%**

Although Adam and RMSprop often excel in various tasks, here SGD optimizer with a learning rate of 0.1 consistently produced the best CNN performance.

7. Dropout Rate Application

Model	Dropout Rate	Dropout Location	Accuracy
CNN-16	0.3	After 1st Fully Connected Layer	98.61%
CNN-17	0.5	After 1st Fully Connected Layer	98.45%
CNN-18	0.3	After Last Conv2D Layer	98.52%

Applying dropout with a rate of 0.3 directly after the first fully connected layer slightly improved generalization by mitigating overfitting without severely impacting accuracy.

BEST MODEL CONFIGURATION

The best-performing model was a Convolutional Neural Network (CNN) characterized by a carefully optimized combination of hyperparameters that together maximized classification accuracy on the MNIST dataset, reaching **98.73%**. This model utilized two convolutional layers followed by three fully connected (FC) layers. The choice of two Conv2D layers allowed effective hierarchical feature extraction, capturing spatial dependencies in the image data without overcomplicating the architecture. The three FC layers provided sufficient representational capacity to model complex relationships while avoiding overfitting.

A learning rate of **0.1** paired with the **Stochastic Gradient Descent (SGD)** optimizer proved to be optimal for convergence speed and training stability. Unlike lower learning rates used in ANN models, this higher rate facilitated rapid descent toward an accurate solution without causing instability in weight updates, demonstrating the distinctive dynamics of CNN training.

The model was trained for **15 epochs**, balancing sufficient exposure to the data for learning while preventing overfitting, as longer training showed diminishing returns. A **batch size of 64** was selected since it yielded better gradient estimation and generalization compared to larger batch sizes.

The activation function chosen was the standard **ReLU**, which effectively introduces non-linearity while avoiding issues like vanishing gradients, thus maintaining stable and efficient training. To further improve generalization, a **dropout rate of 0.3** was applied immediately after the first fully connected layer. This dropout rate successfully reduced overfitting with minimal impact on accuracy.

Collectively, these hyperparameter choices represent a harmonized model configuration that balances feature extraction depth, training efficiency, and generalization capacity to achieve superior MNIST classification performance.

CONCLUSION

This project systematically explored the influence of various hyperparameters on the performance of neural networks applied to the MNIST handwritten digit classification task. The key insight is that Convolutional Neural Networks (CNNs) significantly outperform Artificial Neural Networks (ANNs), achieving a top accuracy of **98.73%** compared to **97.66%** for the best ANN model.

Through comprehensive tuning, the optimal CNN model was identified with the following characteristics:

- **Architecture:** Two convolutional layers followed by three fully connected layers that balance model complexity and the risk of overfitting.
- **Optimizer and Learning Rate:** Stochastic Gradient Descent (SGD) with a relatively high learning rate of 0.1 enabled fast and stable convergence.
- **Batch Size:** Smaller batch sizes (64) were found to improve generalization performance, likely due to more frequent parameter updates.
- **Activation Function:** The ReLU activation consistently led to higher accuracy than alternative activations like sigmoid or tanh.
- **Regularization:** Applying dropout at 0.3 after the first fully connected layer helped reduce overfitting and enhanced model robustness.