Machine Learning Project Report: Neural Network Implementation on MNIST Dataset

Noé Bourgeois

Academic Year 2023-2024

Abstract

1 Introduction

2 Experimentation Framework

2.1 Neural Network Architecture

2.2 Forward Propagation

2.3 Error Calculation

2.4 Backpropagation

2.5 Batching

2.5.1 Lower Memory Requirements

Smaller batches require less memory, making it feasible to train models on machines with limited GPU memory. This allows for using more complex models or higher-resolution data.

When we tried to train the network with the whole dataset on 21000+ neurons, we ran into memory issues. We then decided to split the dataset into batches of 100 samples each.

2.5.2 Better Generalization

We realised that the accuracy was higher than expected with smaller batches. Smaller batches often provide a regularizing effect and higher generalization performance. This is because each update is noisier, which can help the network to escape local minima in the loss landscape and explore a broader range of solutions. We finally decided to use batches of 32 samples each.

2.5.3 Frequent Updates

With smaller batches, the model weights are updated more frequently, which can lead to faster learning initially. This is especially useful in the early stages of training, where large updates can help to make significant progress.

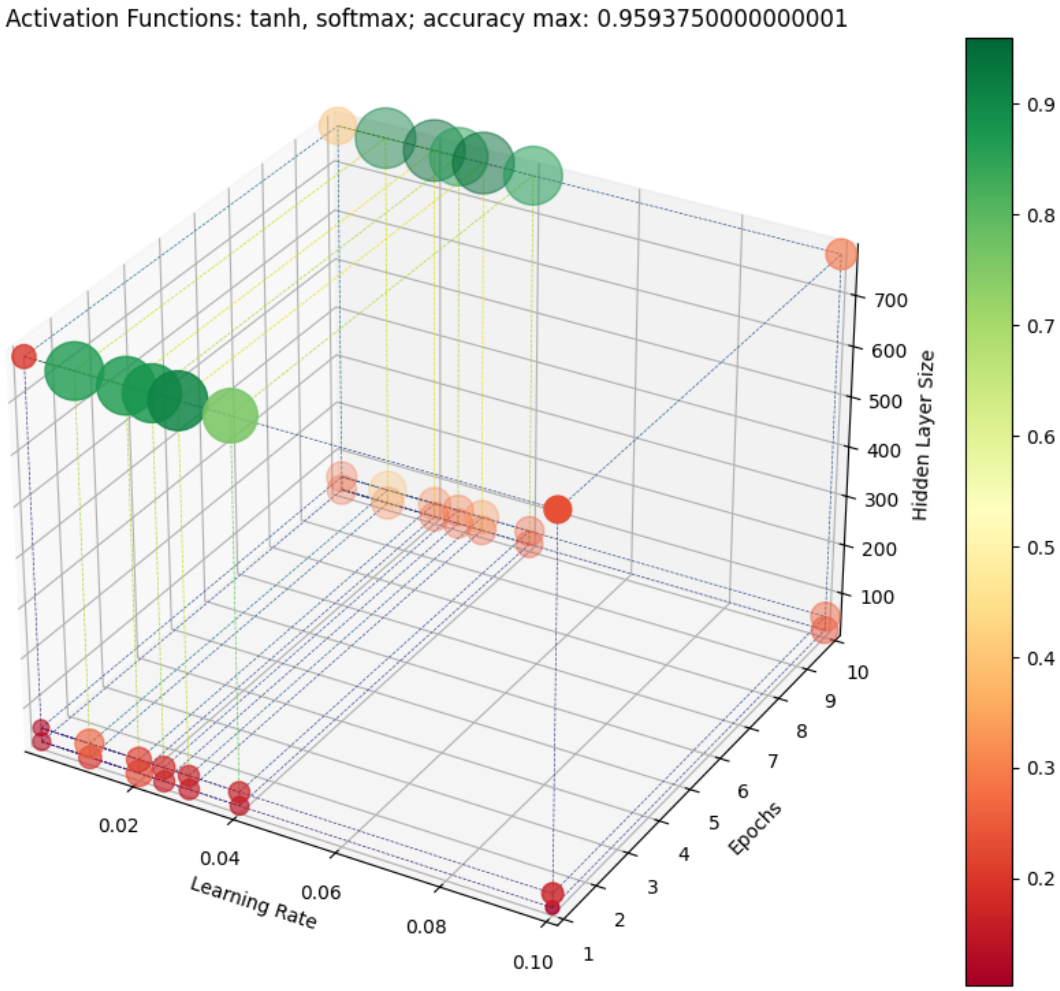

Figure 1: Training comparison results for tanh activation function, softmax output function

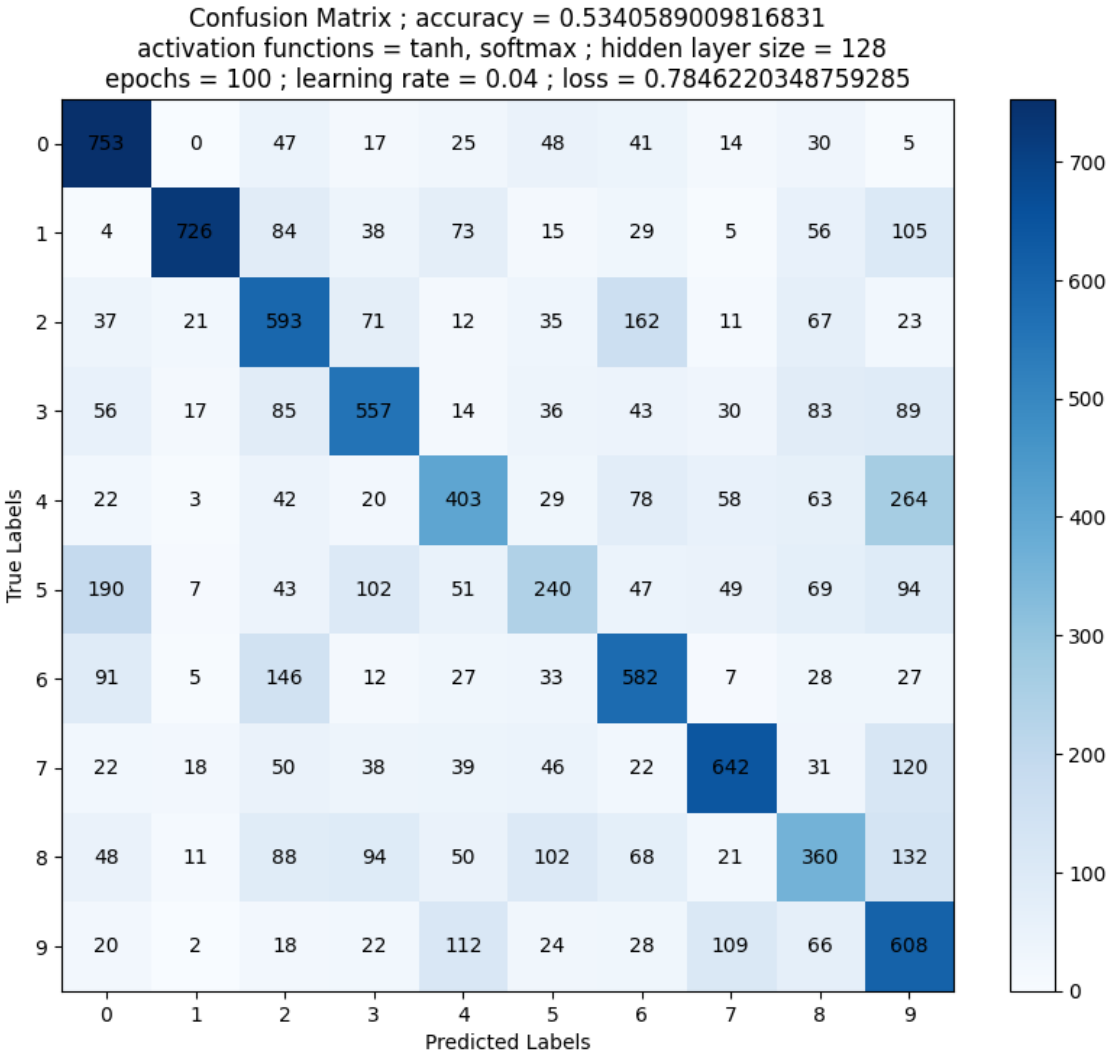

Figure 2: Confusion Matrix for tanh activation function, softmax output function, 1 hidden layer of 128 neurons, 100 epochs, 0.04 learning rate, and 0.7846220348759285 loss.

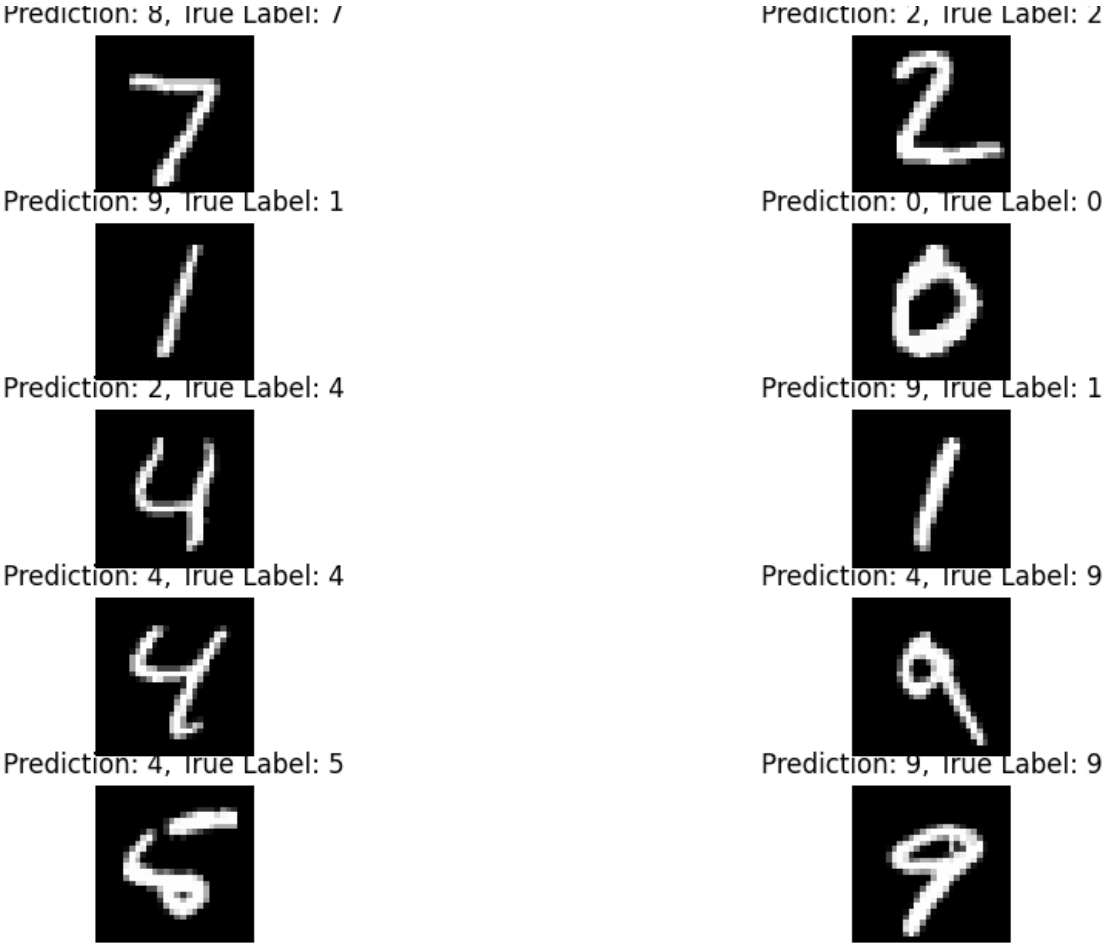

Figure 3: Prediction

3 Results

3.1 Training

3.2 Testing

3.2.1 Confusion Matrix

3.2.2 Accuracy

3.2.3 Prediction

3.3 Convolution

98.8% accuracy

3.3.1 Feature Learning

Unlike traditional machine learning algorithms, CNNs automatically detect important features without any human intervention. This is achieved through the use of filters to the input, enabling the model to learn features directly from the images.

3.3.2 Spatial Hierarchy of Features

CNNs are adept at understanding the spatial hierarchy in images. Early layers might detect simple features like edges or corners, while deeper layers can recognize more complex features like shapes or specific objects. This hierarchical approach is particularly effective for image classification.

3.3.3 Parameter Sharing and Local Connectivity

In CNNs, the same weights (filters) are shared across different parts of an image. This not only reduces the number of parameters the network needs to learn, making the model more efficient, but also allows the network to recognize features regardless of their position in the input image.

3.3.4 Robustness to Image Variations

CNNs are less sensitive to variations and distortions in images, such as slight shifts, rotations, and changes in scale. This robustness is due to the pooling layers commonly used in CNNs, which help in making the detection of features invariant to small changes in the position of the feature in the image.

3.3.5 Efficient Handling of High-Dimensional Data

Images are high-dimensional data (considering each pixel as a dimension). CNNs can manage this high dimensionality better than many traditional algorithms, which often struggle with the curse of dimensionality.

3.3.6 Specialization for Image Data

The convolutional layers are particularly well-suited for processing images pixels that are spatially close to each other, capturing local dependencies effectively. This specialization perfectly matches the structure of characters, in our case digits, images.

3.3.7 Confusion Matrix

4 Analysis

The results indicate a strong dependency of neural network performance on its architecture and hyperparameters.

4.1 Hidden Layer Size impact

biggest impact

4.2 Learning Rate impact

reducing learning rate increases accuracy

4.3 Epochs impact

augmenting epochs is good only if neurons quantity and learning rate decay are augmented exponentially. Otherwise, overfitting occurs.

4.4 Convolutional Model

The convolutional model particularly excelled in image classification tasks.

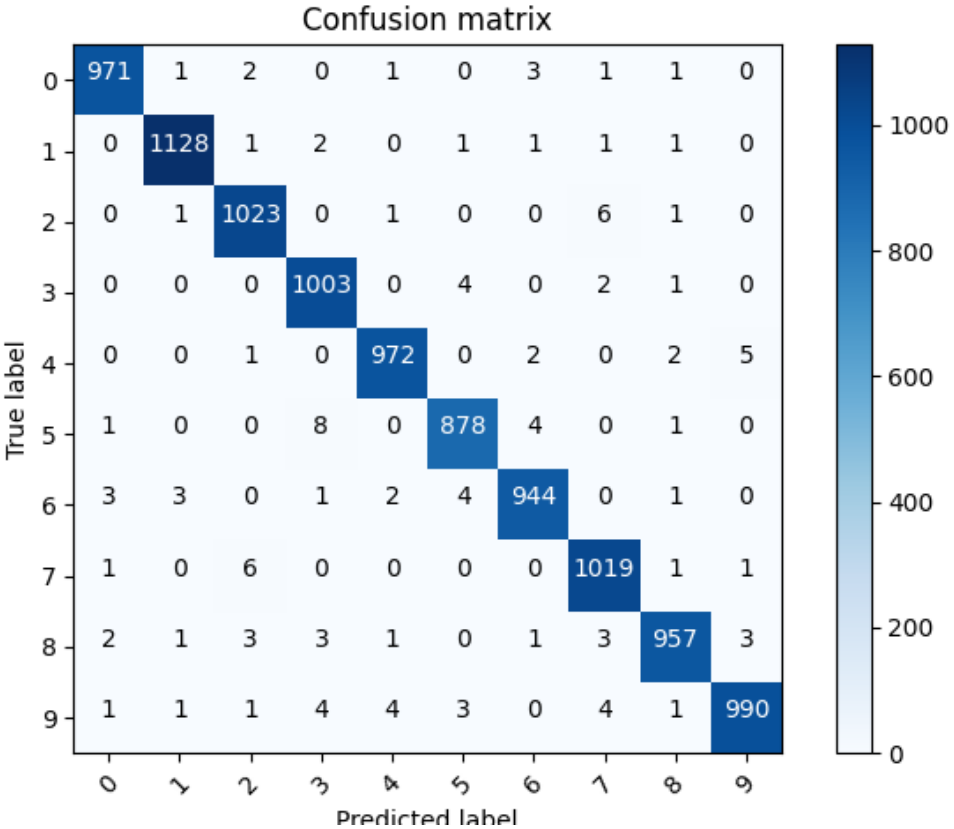

Figure 4: Confusion Matrix for convolutional model.

5 Conclusion

5.1 Main Findings

The study demonstrates the effectiveness of neural networks in digit classification.

5.2 Tools

5.2.1 Assistance

- ChatGPT

5.3 Future Work

We will continue exploring deeper and evolving architectures and implementing advanced techniques like dropout and batch normalization at different intensities at particular network states.

6 References

References