Machine Learning Project Report: Neural Network Implementation on MNIST Dataset

Noé Bourgeois

Academic Year 2023-2024

1 Introduction

This report details a neural network's development and analysis for digit recognition using the MNIST dataset, highlighting key findings and future research directions.

2 Experimentation Framework

For the framework base, please see the instructions.

2.1 Batching

We used batching from 0.5 to 1e-5 to reduce the training process memory need and increase noisy data able accuracy.

2.2 Learning rate

Reducing the batch size requires to lower the learning rate proportionally. The learning rate can also be augmented with the neurons quantity or the loss and reduced with the epochs or the fitness

2.3 Weights

2.3.1 Weight Restoration

The network can memorize the training lowest loss weights and revert to the memorized weights.

3 Results

3.1 Training

3.2 Testing

3.2.1 Confusion Matrix

3.3 Convolution

With a convolutional model, we achieved 98.8% accuracy in 1 minute and 40 seconds. to compare, the best accuracy we achieved with a non-convolutional model was 91% in 1 minute and 40 seconds.

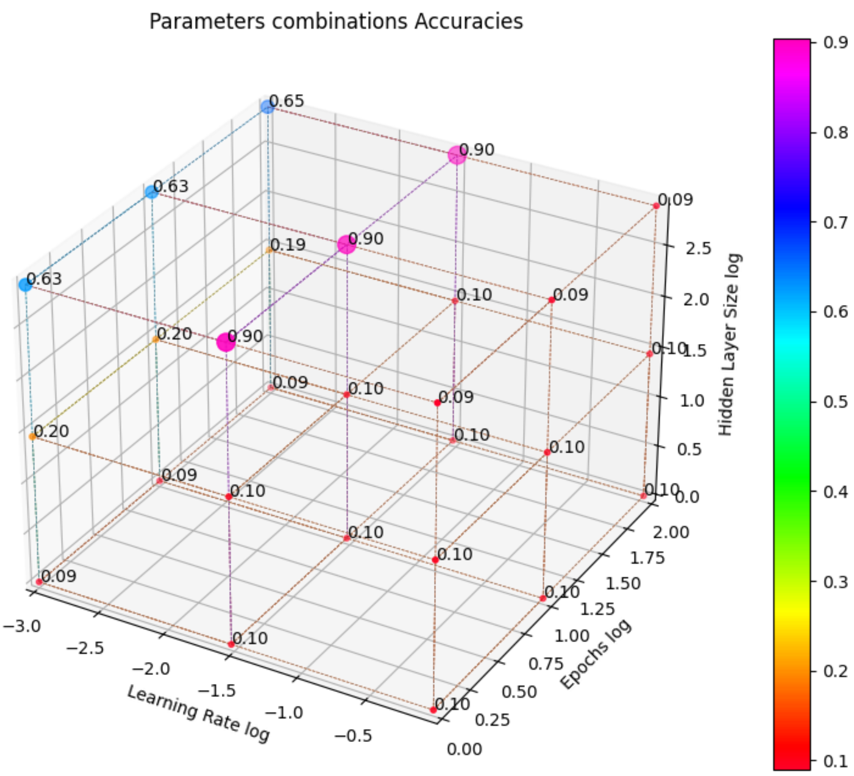
4 Analysis

The "parameters combinations accuracies" results plots indicate a strong dependency of our neural network performance on its size, we can clearly see that color shadind on the vertical axis. The learning rate, whose calculation was optimized for 784 neurons and becomes even more fragile with batch size reduction also has a strong impact on the accuracy. Even at this same size, a milli scale variation separates near- maximum and maximum accuracy. During training, micro scale variations determines the accuracy evolution. The epochs quantity allows the accuracy to evolve, but only positively if the learning rate is adapted. Otherwise, stagnation or overfitting occurs.

5 Conclusion

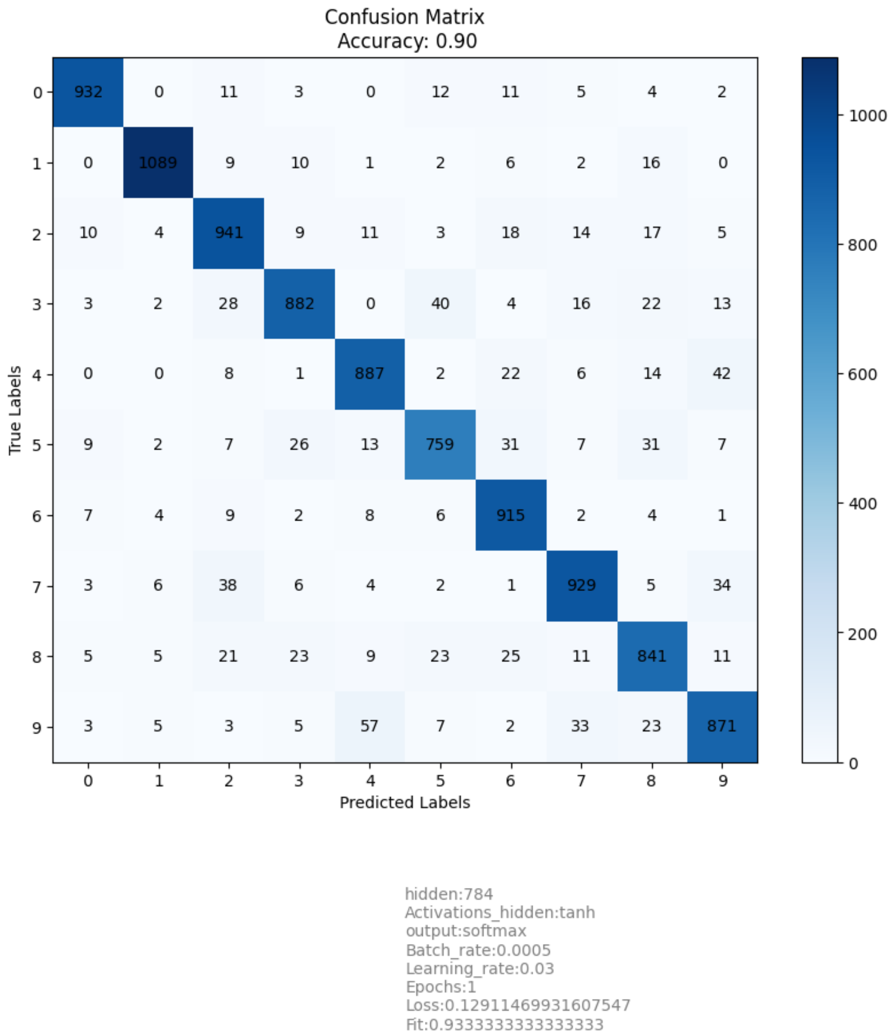
5.1 Main Findings

To be concise, the biggest challenge being to constantly adpapt the learning rate, we conclude that a biggest network allowing high and fast performance (to 90% accuracy after 1 epoch), smallest batching catalyzing that behavior on noise and a learning rate adapted to network and batch size,



accuracy max: 0.9036
hidden layer sizes : [1, 28, 784]
learning rates: [0.001, 0.03, 0.9]
epochs : [1, 10, 100]

Activations:
hidden:tanh
output:softmax
Batch_rate:0.0005


Confusion Matrix
Accuracy: 0.90
True Labels
Predicted Labels
932 0 11 3 0 12 11 5 4 2
0 1089 9 10 1 2 6 2 16 0
10 4 941 9 11 3 18 14 17 5
3 2 28 882 0 40 4 16 22 13
0 0 8 1 887 2 22 6 14 42
9 2 7 26 13 759 31 7 31 7
7 4 9 2 8 6 915 2 4 1
3 6 38 6 4 2 1 929 5 34
5 5 21 23 9 23 25 11 841 11
3 5 3 5 57 7 2 33 23 871
hidden:784
Activations_hidden:tanh
output:softmax
Batch_rate:0.0005
Learning_rate:0.03
Epochs:1
Loss:0.12911469931607547
Fit:0.9333333333333333

and dynamically to loss. should be preffered. That having been said, many other techiques can be added to improve the model's time complexity.

5.2 Tools

- ChatGPT

5.3 Future Work

We will continue Implementing advanced techniques like filtering images on sparse matrices, biases, batch normalization- and dropout regularization- at different intensities at particular network states and exploring convolutional models and deeper pattern finding architectures.

The project is available on https://github.com/nobourge/artificial-intelligence-machine-learning.