

Deep Learning in Practice - TP1 Report

Hyperparameters and Training Basics with PyTorch

Jean-Vincent Martini - Adonis Jamal
January 23, 2026

1 Multi-Classification Problem

For this problem, we want to perform multi-class classification on the USPS dataset using a neural network implemented in PyTorch. This dataset consists of grayscale images of handwritten digits (0-9) with a resolution of 16x16 pixels. Examples of these images are shown in Figure 1.

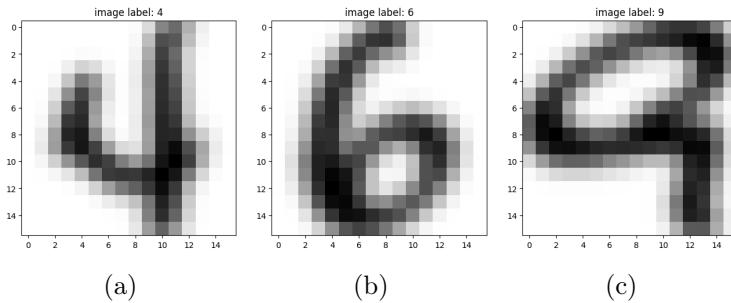


Figure 1: Samples from the USPS dataset.

Our neural network model consists of identical 2d convolutional layers with a kernel size of 3x3, followed by a non-linear activation function and max-pooling layers. The final layer is a fully connected layer that maps the extracted features to the 10 output classes corresponding to the digits 0-9. We chose this architecture because convolutional layers are well-suited for image data, as they can effectively capture spatial hierarchies and patterns in the images. We devided the dataset into training, validation, and test sets that we used respectively for training the model, tuning hyperparameters, and evaluating the final performance.

To tune the hyperparameters of the model, we will proceed by ablation studies, meaning that we will vary one hyperparameter at a time while keeping the others fixed to their default values. This approach allows us to isolate the effect of each hyperparameter on the model's performance and to gain time compared to a full grid search, as we will discuss later in section 3. The hyperparameters we will explore include the number of layers, the number of neurons in each hidden layer, the activation function, the batch size, the learning rate, the number of epochs, the optimizer, and the loss function.

The default configuration of the model is as follows: 2 hidden layers with 16 neurons each, ReLU activation function, batch size of 64, learning rate of 0.001, 30 epochs, Adam optimizer, and Mean Squared Error (MSE) loss function. The results of the ablation studies for each hyperparameter are presented in the Appendix, section A. For each configuration, we report the training accuracy, validation accuracy, and training time to provide a comprehensive view of the model's performance.

From the results, we decide to choose the final configuration of the model based on the best validation accuracy achieved for a reasonable training time. For the final configuration, we choose the same architecture as the default one, but with a learning rate of 0.01 and the loss function set to Cross-Entropy. This configuration achieves a training accuracy of 99.98%,

a validation accuracy of 98.30%, and crucially a test accuracy of 95.81%. The test accuracy is obtained by evaluating the final model on the test set, which was not used during training or validation in section A. This metric therefore provides an unbiased estimate of the model’s generalization performance on unseen data.

2 Regression Problem

3 Observations and Trade-offs to Consider When Training Deep Learning Models

A Problem 1: Results for Different Hyperparameters

The symbol * indicates the value used in the default configuration.

Impact of the number of layers:

Number of Layers	Train Accuracy (%)	Validation Accuracy (%)	Train Time (s)
1	98.57	96.82	7.91
2*	99.08	97.44	14.52
3	99.22	97.37	18.26
5	97.83	96.90	24.69

Impact of the number of neurons in hidden layers:

Hidden Layer Size	Train Accuracy (%)	Validation Accuracy (%)	Train Time (s)
8	97.72	96.13	11.62
16*	99.08	97.44	14.52
32	99.68	98.14	18.94
64	99.90	98.92	29.29

Impact of the activation function:

Activation Function	Train Accuracy (%)	Validation Accuracy (%)	Train Time (s)
ReLU*	99.08	97.44	14.52
Sigmoid	96.83	94.58	14.28
Tanh	99.72	97.60	14.79
Leaky ReLU	98.77	97.37	14.39

Impact of the batch size:

Batch Size	Train Accuracy (%)	Validation Accuracy (%)	Train Time (s)
16	99.62	98.76	18.74
32	99.05	96.82	15.78
64*	99.08	97.44	14.52
128	98.18	96.59	13.57

Impact of the learning rate:

Learning Rate	Train Accuracy (%)	Validation Accuracy (%)	Train Time (s)
0.001*	99.08	97.44	14.52
0.01	99.43	98.14	14.36
0.1	16.23	17.04	14.21
1	16.23	17.04	14.34

Impact of the number of epochs:

Number of Epochs	Train Accuracy (%)	Validation Accuracy (%)	Train Time (s)
10	97.27	95.82	4.87
30*	99.08	97.44	14.52
50	99.50	97.68	24.05
100	99.50	97.13	47.41

Impact of the optimizer:

Optimizer	Train Accuracy (%)	Validation Accuracy (%)	Train Time (s)
Adam*	99.08	97.44	14.52
RMSprop	99.17	97.29	14.27

Impact of the loss function:

Loss Function	Train Accuracy (%)	Validation Accuracy (%)	Train Time (s)
MSE*	99.08	97.44	14.52
Cross-Entropy	99.05	97.68	14.35

B Problem 2: Results for Different Hyperparameters