COMP 551 Project 2 Report: *Classification of Image Data with Multilayer Perceptrons and Convolutional Neural Networks*

Students: Albert Caupin, Max-Henri Chanut and Christine Pan

Due Date: Oct 31st, 2023

Abstract

In this project, we implemented a multilayer perceptron model and used it to classify image data from Fashion MNIST and CIFAR-10. The goal of this project is to implement a basic neural network and its training algorithm and to run different experiments by manipulating different parameters such as number of hidden layers or different type of activations. Based on the results of these experiments we were able to analyse the significance of architectural and hyperparameter choices and the importance of regularization techniques and their effects on the prediction accuracy.

1 Introduction

The goal of this project is to implement a multilayer perceptron model with which we train and test on two different datasets, FASHION MNIST and CIFAR-10, both consisting of pixel squared images that are labelled in 10 different classes. After acquiring both dataset and processed and normalizing the data, we implemented the multilayer perceptron model with the forward and backward function and with different activation functions such as ReLU, Sigmoid and Softmax. We then proceeded to experiment the effects of weight initialization, model architecture, activation functions, regularization, image preprocessing, optimizer choices, and the use of convolutional neural networks (CNNs) on both Fashion MNIST and CIFAR-10 datasets. The goal was to understand how these parameters can influence the accuracy of the predictions from the model. Some important findings were how the architectural and hyperparameter choices and the regularization techniques can affect the accuracy of the models' predictions.

2 Datasets

Fashion MNIST is a dataset designed for image classification tasks. It contains a collection of grayscale images (28x28 pixels) of clothing items and accessories, divided into ten classes. These classes include items like t-shirts, trousers, pullovers, dresses, coats, sandals, shirts, sneakers, bags, and ankle boots. Each class has an equal number of examples, making it a balanced dataset for multi-class classification. **CIFAR-10** (Canadian Institute For Advanced Research 10) is another dataset for image classification. It consists of 60,000 color images (32x32 pixels) across ten different classes. These classes represent various objects, animals, and vehicles, making it a challenging dataset for computer vision tasks. We have explored the datasets by visualizing some sample images from each classes to get an idea of the data, and determined that we needed to normalize the training and testing set. In both cases, exploratory analysis helped us understand the structure of the datasets, the distribution of classes, and these preprocessing steps were needed before we could train our neural networks.

3 Results

Learning Rate Selection

In our experiments, we tested different learning rates (lr) for training a two-hidden-layer MLP and observed their respective accuracies on the validation set. The purpose was to identify the learning rate that resulted in the best model performance.

Learning Rate (lr)	**Accuracy**
0.001	0.8741
0.01	0.8651
0.1	0.1

Table 1: Comparison of Accuracy for Different Learning Rates

As shown in the Table, we trained the MLP with three different learning rates: 0.001, 0.01, and 0.1. It is evident that a learning rate of 0.001 resulted in the highest accuracy of 0.8741 on the validation set.

3.1 Experiment 1 (Fashion MNIST)

The provided results for the three different models with different weight initialization, with one hidden layer of 128 units, at 50 epochs.

Initialization	**Accuracy**
Zeros	0.4689
Uniform	0.8541
Gaussian	0.8589
Xavier	0.8635
Kaiming	0.8607

Table 2: Accuracy Scores for Different Initializations

The above suggests that using Xavier weight initialization yields the higher accuracy when using the multilayer perceptron model with the Fashion MNIST dataset, while initializing with zeros for weights yields the lowest accuracy. Uniform, Gaussian and Kaiming distribution have a very close accuracy to Xavier, differentiating at most 1%.

3.2 Experiment 2 (Fashion MNIST)

The provided results for the three different models with varying levels of non-linearity (activation functions) and network depth are as follows:

Model	**Accuracy**
No hidden layer MLP	0.8293
MLP with one hidden layer (128 units, ReLU activation)	0.8749
MLP with two hidden layers (128 units each, ReLU activation)	0.8839

Table 3: Accuracy Scores for Different MLP Models

The no hidden layer MLP essentially works as a linear model. It directly maps the inputs to outputs, and as such, it's limited in its ability to capture complex patterns in the data. The accuracy achieved (0.8293) reflects this limitation. Adding a single hidden layer with ReLU activation introduces non-linearity to the model. ReLU is a highly effective activation function for deep learning, and it enables the model to capture more complex relationships in the data. As a result, the accuracy improves to 0.8749. Introducing two hidden layers with ReLU activation further increases the network depth. This allows the model to learn even more intricate features and representations in the data. As a result, the accuracy improves again, reaching 0.8839.

3.3 Experiment 3 (Fashion MNIST)

Here are the results of the performances of models with 2 hidden layers and different activation functions and an analysis of their relative performance:

Activation Function	**Accuracy**
ReLU Activation	0.8689
Leaky ReLU Activation	0.8605
Sigmoid Activation	0.7966

Table 4: Accuracy Scores for Different Activation Functions

The results align with expectations to some extent. ReLU outperformed Leaky ReLU and Sigmoid because it is generally more suitable for hidden layers in deep neural networks. It addresses the vanishing gradient problem better and facilitates the learning of complex features. Leaky ReLU also performed well, indicating that it can be a good alternative to standard ReLU in some cases but may not always outperform it. Sigmoid's lower performance can be attributed to its characteristics, which are less favorable for deep networks. Sigmoid activations can result in vanishing gradients, making it challenging for the model to converge effectively.

3.4 Experiment 4 (Fashion MNIST)

The provided results indicate the impact of L1 and L2 regularization on the accuracy of an MLP with two hidden layers, each having 128 units and ReLU activations:

Regularization Type	**Accuracy**
No Regularization	0.8611
L1 Regularization	0.8726
L2 Regularization	0.8698

Table 5: Accuracy Scores for Different Regularization Types

In our case, both L1 and L2 regularization have a positive impact on accuracy. L1 regularization, which encourages sparsity in the weights, achieves the highest accuracy improvement, followed by L2 regularization. The results suggest that regularization techniques are effective in improving the generalization performance of the model on unseen data. This is because regularization helps the model avoid overfitting by controlling the complexity of the model, leading to better test accuracy.

3.5 Experiment 5 (Fashion MNIST)

The provided results indicate the impact of the normalization on the dataset and its accuracy using the MLP. Since we have been using normalized data, here are results using unnormalized data (we did not divide 255 to the data, as pixels range from 0 to 255):

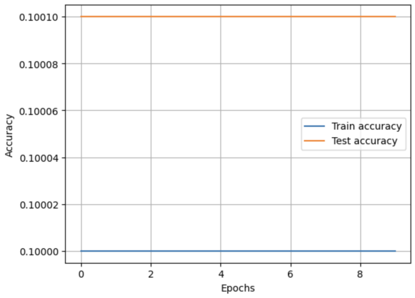

Figure 1: Accuracy vs Epochs

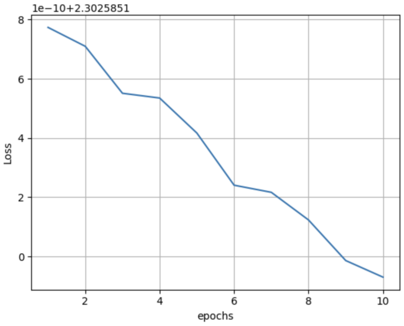

Figure 2: Loss vs Epochs

The above suggests that the model achieved a training accuracy of about 10.01% and a test accuracy of about 10% after training for 50 epochs with a learning rate of 0.001. While it is a low accuracy, this result is expected because the input vector contains values ranging up to 255 (as opposed to up to 1 when normalized). In this setup, the weighted sum of the layer significantly deviated from zero, eliminating the nonlinear component of the activation function.

Dataset	Accuracy
Unnormalized	0.8475

Table 6: Accuracy of MLP with unnormalized dataset.

Bonus Experiment (For Fashion MNIST)

The provided results indicate the impact of different units per hidden layer on accuracy when performing with MLP on the Fashion MNIST dataset:

Width (1 Layer)	**Accuracy**	**Width (2 Layers)**	**Accuracy**
Width 32	0.8496	Width 32	0.8584
Width 64	0.8552	Width 64	0.8564
Width 128	0.8592	Width 128	0.8703
Width 256	0.8673	Width 256	0.8778

Table 7: Accuracy of MLP model with different widths and layers.

The above suggests that the bigger the width, the better the accuracy of the MLP using the dataset. Furthermore, the bigger number of hidden layers shows better accuracy as well.

Bonus Experiment (For Fashion MNIST)

The next experiment is training MLP with 10^k images, $k \in \{0, 1, 2, 3, 4\}$. For all previous experiences, we have been using 60 000 examples in the training set.

Number of Images	**Accuracy**
Size 1	0.2128
Size 10	0.139
Size 100	0.1033
Size 1000	0.7108
Size 10000	0.8127

Table 8: Accuracy of MLP model with different number of images.

The above suggests that bigger number of images such as 1000 and 10000, the better the accuracy. Indeed, at size 1, 10, 1000, the accuracy is very low and show some inconsistency. Only at 1000 images in the training set is where the accuracy is reasonable. This shows that a larger dataset results in a better generalization.

3.6 Experiment 6 (CNN with MNIST fashion)

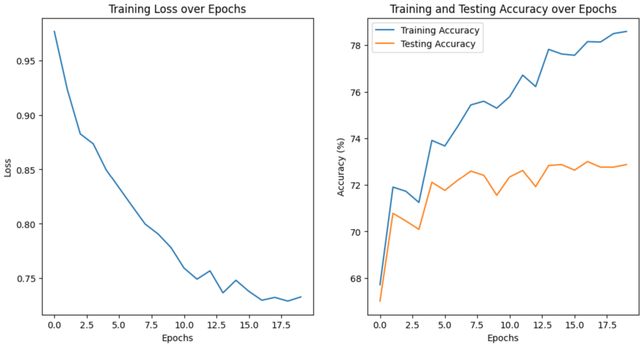

Figure 3: Training of the CNN on Fashion MNIST

During our experiment we discovered that using a CNN gave a much bigger accuracy but was also very computationally heavy. Our biggest run ran for 50 minutes and had a test accuracy of 90% (and more than 99% in training data !). Above is a more reasonable training that took only 15 minutes and took less filters per convolutional layers. In conclusion the CNN implementation is more accurate but also require much more computation power.

3.7 Experiment 7 (CIFAR-10)

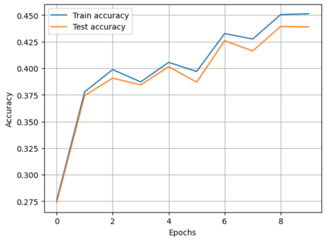

Figure 4: MLP trained on CIFAR

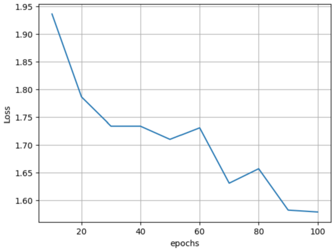

Figure 5

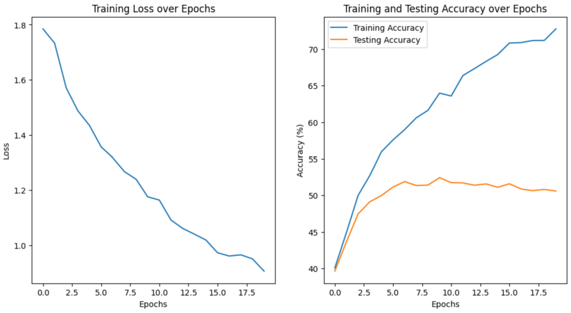

Figure 6: CNN with stride trained on CIFAR-10

As it can be seen in the graphs, the training of the CNN gave more accuracy in its results by about 7%, while also obtaining these results much faster (only 3 epochs are needed to obtain results similar to what the MLP does in 8 epochs). The CNN model could be refined to decrease the overfit we see appearing after 3 epochs but due to the much larger number of hyperparameters and computation time this would take more time and a more powerful computer. The better performances of the CNN can be explained by the fact that the convolutional layers take full advantage of the 3 dimensions of the data (x, y and RGB) by keeping the input in its original shape and preprocessing it to extract features that can then be fed to the fully connected layers.

3.8 Experiment 8 (CIFAR-10)

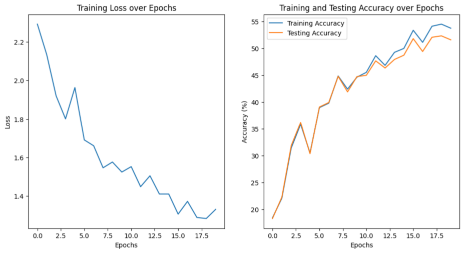

Figure 7: CNN with SGD optimization and momentum of 0

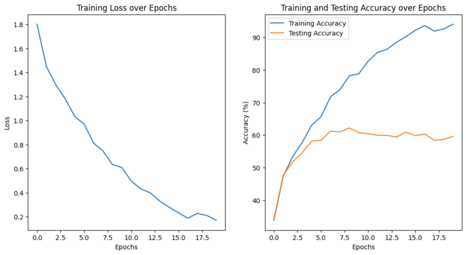

Figure 8: CNN with SGD optimization and momentum of 0.9

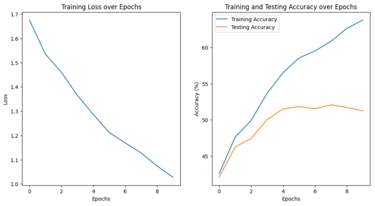

Figure 9: CNN with Adams optimization

In this experiment we can see the effect of the optimizer on evolution of the accuracy of the models. We observe that a momentum of 0 gave a slower evolution and is also less stable. When we increase the value of the momentum to 0.9 the evolution is much faster and we reach the maximum accuracy of the system after only 5 epochs, and the evolution is more stable. Finally, the Adams optimisation gives results similar to the 0.9 momentum SGD but reaches the same result even faster in 4 epochs. It is worth noticing that the Adams optimization takes more time to compute so in that particular application the SGD optimization with momentum seems to be the best option.

4 Discussion and Conclusion

In this project, we set out to explore various aspects of image classification using machine learning techniques. We covered a range of topics, from building multilayer perceptrons (MLPs) to implementing convolutional neural networks (CNNs). Furthermore, We observed that the choice of model architecture and activation function plays a critical role in determining the performance of the image classification task. Deeper networks with suitable activation functions, such as ReLU, outperformed simpler models with linear activation functions. We also found that regularization techniques, specifically L1 and L2 regularization, can significantly improve model generalization. L1 regularization, in particular, was effective in preventing overfitting and enhancing the overall accuracy of the model. We also explored how the use of more advanced models such as the CNNs could improve accuracy with more complex data by isolating meaningful features before classifying.

This project opens up avenues for future investigations. Some potential directions include: Exploring more advanced CNN architectures, such as ResNet or Inception, for image classification to further improve accuracy or investigating the impact of different optimizers (e.g., Adam, SGD) on training dynamics and model convergence. We could also consider data augmentation techniques to increase the size of the training dataset, which can lead to better model generalization.

In conclusion, this project provided valuable insights into the world of image classification and machine learning. The performance of the models was significantly influenced by architectural choices, activation functions, regularization, and hyperparameters. We have laid a foundation for future work and the development of more sophisticated image classification models.

5 Statement of Contributions

Max-Henri : Task 1, Task 2, Task 3 part 1-5, Christine : Task 2, Task 3 part 1-5, bonus, Albert : Task 2, Task 3 part 6-8