COMP 551 - HW2

Applied Machine Learning

Lancelot Normand
261155638
lancelot.normand@mail.mcgill.ca

Eduard Anton
261033247
eduard.anton@mail.mcgill.ca

Jessica Zhu
260957235
jessica.zhu@mail.mcgill.ca

Fall 2023

School of Computer Science

McGill

Contents

List of Figures

List of Tables

1 Abstract

Abstract This brief study investigates the performance of our Multi-Layer Perceptrons (MLP) and Convolutional Neural Network (CNN) implementations in Classification Tasks. Through systematic experimentation involving adjustments to weights, layers, activation functions, regularization, normalization, and momentum, we evaluated our models using CIFAR-10 and Fashion-MNIST datasets. Our findings align with existing literature, suggesting that even straightforward models can exhibit commendable performance in classification tasks. Impressively, our MLP model achieved 88.4% accuracy on MNIST-Fashion, while our CNN model demonstrated 73% accuracy on CIFAR-10.

2 Introduction

In the scope of this project, we implemented two distinct network architectures to address classification tasks. The initial design involves a rudimentary MLP developed from scratch, while the second design employs a PyTorch-based CNN. Both models were experimented to adjust their weights, layers, activation functions, regularization, normalization, and momentum. To test the accuracy of our implementations, we used two benchmarking datasets employed in literature: Dataset 1 comprises FashionMNIST, encompassing a diverse array of clothing articles, while Dataset 2 consists of CIFAR-10, featuring a curated collection of animals and vehicles. The popularity of these datasets make it easy to compare the quality of our results with the rest of the literature. One paper in particular [1] used a CNN model to obtain the same accuracy of 73.04% as we did on CIFAR-10 and 93.68% on FashionMNIST, while we obtained 88.4% using an MLP model. That being said it should be noted that our results are no replacement for state of the art, and that the literature has much better performance than us. Even with low memory, a paper's [2] CNN managed to beat ours, but this is in part explained by the extra layers the authors used. [3]

3 Datasets

3.1 Fashion MNIST dataset

The FashionMINIST dataset [4] consists of black and white images of 28x28 of articles of clothing. As seen in Figure 9, there are 10 classes of images equally distributed over 70,000 images, where 10,000 are used for testing. MNIST dataset is often used as a benchmark for various machine learning algorithms and is included in most ML libraries. Unless otherwise specified, our models were trained on a normalized dataset to facilitate training.

3.2 CIFAR-10 dataset

CIFAR-10 (Canadian Institute for Advanced Research) [5] is another standard dataset used for machine learning training and benchmarking. As seen in Figure 10 it consists of 60,000 32x32 color images, where 10,000 are used for testing. As the name suggests it has 10 classes of images from a pool of various vehicles and animals.

4 Results

4.1 Choosing Hyperparamters

Before investigating other aspects of our MLP, we tested multiple learning rates and batch sizes (learning rates: 0.1, 0.001, 0.0001, 1×10^{-5}, 1×10^{-6}; batch sizes: 10, 32, 64, 128, 256, 500, 1000) on a basic MLP implementation to determine the hyperparameters that would provide the best baseline model. At first, the MLP model we used had its weights initialized with a random uniform. The model was trained and tested on the Fashion-MNIST dataset. As seen in Table 7, Figure 11 and 12, a learning rate of 0.1 and a batch size of 32 provides the best performance based on accuracy (80.2% accuracy on test set) for the MLP model.

4.2 Different weights initialization

The effect of the weight initialization on the training curves and test accuracy of MLP models was investigated. Using Fashion-MNIST, five different weight initializations were tested: all zeros, random uniform (with Uniform[-1,1]), random normal, Xavier, and Kaiming. We observed that Kaiming weight initialization performed the best in both training and validation sets, after 30 epochs, with the highest test accuracy of 88.4% (Table 1). An all zero weight initialization performed the worst with an accuracy of 41.1%, as the model had a hard time being optimized as seen by the slow curve in Figure 1

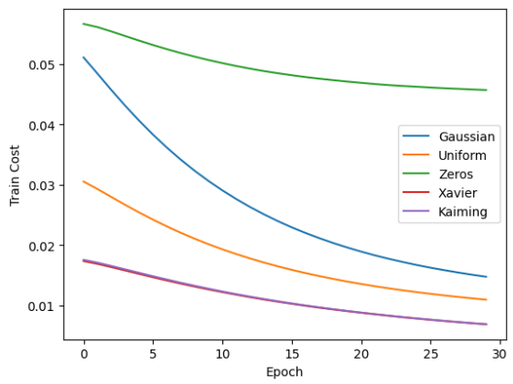


Figure 1: Training Cost Curve for MLP with Different Weight Initializations on Fashion-MNIST Training Set

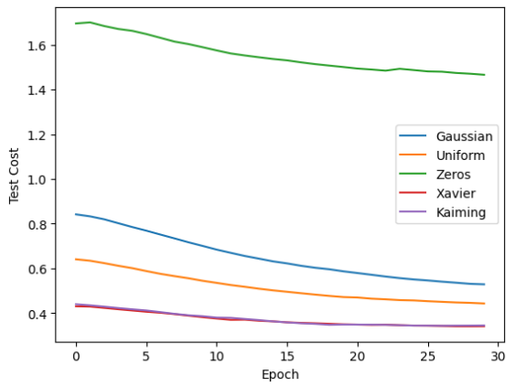


Figure 2: MLP Performance on Fashion-MNIST Test (Validation) Set for Different Weight Initializations

Weight Initialization	Accuracy
All Zeros	41.1%
Uniform[-1,1]	85.6%
Gaussian N(0,1)	82.4%
Xavier	88.0%
Kaiming	88.4%

Table 1: Accuracy of Fashion-MNIST Class Predictions using MLP with Different Weight Initializations

4.3 Changing number of hidden layers

Subsequently, we conducted experiments varying the number of hidden layers in our MLP model while maintaining our baseline model from Section 4.1 and initializing weights using Kaiming. The model was trained with 0, 1, and 2 hidden layers, employing ReLU on every hidden layer. The final output would include a softmax too. As depicted in Figure 13, our data presents non-linear features since the introducing of a hidden layer significantly improves the loss and accuracy of our model. Table 2 further illustrates that each additional

hidden layer contributes to enhancing the MLP model's accuracy. The results obtained are expected. However, as suggested by Figure 14, there is a tendency for the MLP model to start overfitting as the number of hidden layers increases, particularly evident around the 25-epoch mark.

Number of Hidden Layers	Accuracy
0	83.0%
1	86.7%
2	88.1%

Table 2: Accuracy of Class Predictions for the Fashion-MNIST Test Set using MLP with 0, 1 or 2 Hidden Layers

4.4 Different activation functions

Continuing our experiments, we employed various activation functions for the hidden layers of the MLP using the model outlined in Section 4.3 with two hidden layers. The experiment involved testing ReLU, Leaky ReLU, Sigmoid, and Tanh as activation functions. As seen in Figure 3, most activation functions had a similar behaviour and performance with the exception of the sigmoid. The higher loss observed with the sigmoid compared to others in Table 3 could be caused by the vanishing gradient of sigmoid functions that tend to squash input values between 0 and 1 which makes it slower to converge the model using SGD.

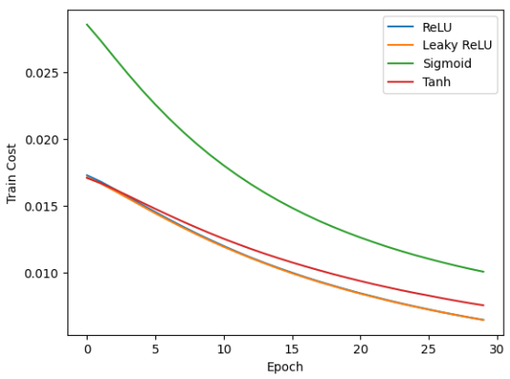


Figure 3: Training Cost Curve for MLP with Different Activation Functions on Fashion-MNIST

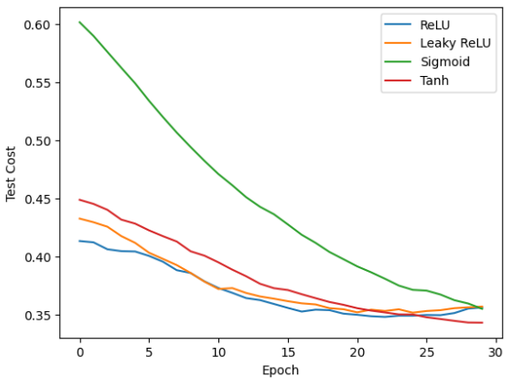


Figure 4: Test Cost for MLP with Different Activation Functions on Fashion-MNIST

Activation Function	Accuracy
ReLu	88.4%
Leaky ReLu	88.4%
Sigmoid	87.7%
Tanh	88.2%

Table 3: Accuracy of Class Predictions for the Fashion-MNIST Test Set using MLP with Different Activation Functions

4.5 Adding regularization

Continuing our refinement of the MLP model outlined in Section 4.3, we implemented L1 and L2 regularization to address observed overfitting issues around the 25-epoch mark, as identified in Section 4.3. After performing the experiment, no results improved previous performance scores as detailed in Table 8. However, the introduction of L1 (α = 0.001) or L2 (α = 0.01) regularization, as depicted in Figure 6 effectively mitigated signs of overfitting in our model.

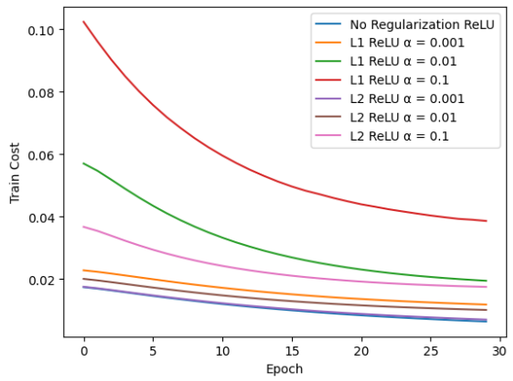


Figure 5: Training Cost Curve for MLP with Regularization on Fashion-MNIST

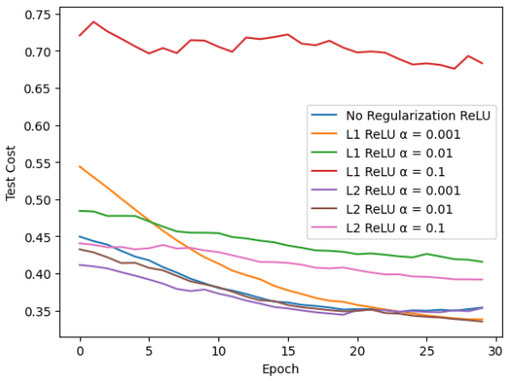


Figure 6: Test Cost for MLP with Regularization on Fashion-MNIST

Regularization	Accuracy
L1 at 0.001	87.9%
L1 at 0.01	85.0%
L1 at 0.1	77.5%
L2 at 0.001	87.7%
L2 at 0.01	88.4%
L2 at 0.1	85.0%

Table 4: Accuracy of Class Predictions for the Fashion-MNIST Test Set using MLP with Regularization

4.6 Training with unnormalized images

Throughout our experiments, MLP models were consistently trained on normalized images from the Fashion-MNIST dataset, where greyscale pixel values range from 0 to 255. Previously, normalization involved dividing pixel values by 255. In this experiment, we diverged by training an MLP model on the unnormalized Fashion-MNIST dataset, maintaining other hyperparameters. The unnormalized model, detailed in Figure 15 and 16, reached a convergence training cost of approximately 0.0156 after 50 epochs, yielding an accuracy of 77.7%. In contrast, the normalized model achieved 84.43% accuracy with a converging training cost of about 0.0105 (Appendix Figure 17 and 18). These results underscore the known impact of data normalization, with training on normalized data consistently leading to more accurate models by aligning input scales.

4.7 A CNN with PyTorch

Using a CNN on the Fashion MNIST dataset we are able to improve the accuracy up 88% overall. This could be explained in part by the fact that CNN can perform better on more complicated images since they takes tensors as input instead of vector. That distinction allows to understand the spatial differences like with nearby pixels.

4.8 CNN and MLP with CIFAR-10 Dataset

4.8.1 With MLP

Using the CIFAR-10 dataset, we trained a MLP with two hidden layers each consisting of 128 neurons, ReLu activation and Kaiming weight initialization. After 50 epochs, the training cost converged to around 0.416 and the MLP had an accuracy of 47.6% (Figure 24, 25).

4.8.2 With CNN

When constrained to use the same CNN architecture as lexperiment 6, we find that an adjustment of hyperparameters is necessary in order to be able to learn. The configuration we settled with was the following:

Figure 7: Architecture of our CNN implementation, and its hyperparameters

airplane	74.4%
automobile	78.7%
bird	64.2%
cat	57.4%
deer	69.7%
dog	60.5%
frog	82.0%
horse	79.8%
ship	83.5%
truck	80.6%

Table 5: Accuracy per classes, overall accuracy 73%

4.9 Increasing Momentum on CNN

Training our CNN for 10 epochs and slowly increasing the momentum increases the overall accuracy and clearly lowers the average loss.

We can see in Figure 8 that at a momentum of 0.9 the model converges much faster than with any lower momentum. Notice the sawtooth pattern that emerges from the plot. It becomes more apparent the higher the momentum is. We know that the major drops of this pattern occurs after the start of a new epoch. We suppose that it has something to do with the reshuffling of minibatches at the start of each epochs. Those new combinations could help the model learn new patterns.

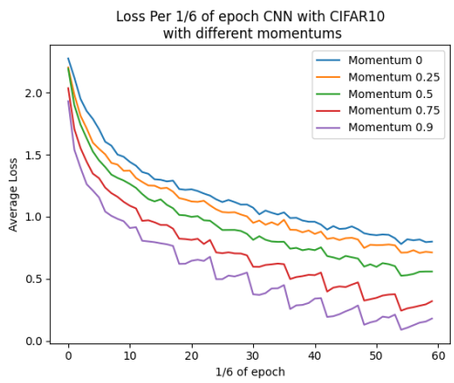


Figure 8: Training loss per epochs given different momentum

For the accuracy, we observe a trend that with higher momentum the accuracy tends to increase. Note that we failed to report the accuracy for 0.25 and 0.75 momentum so we inferred them to be the average of their two respective neighbors.

0.0 Momentum	67%
0.25 Momentum	(Inferred) 69%
0.5 Momentum	71%
0.75 Momentum	(Inferred)72%
0.9 Momentum	73%

Table 6: Accuracy per momentum after 10 epochs

Finally when using ADAM as optimizer but without any momentum its curve loss is exactly in between all our curves and with a final accuracy of 67% after 10 epochs.

4.10 Additional Experiment 1: Effect of Regularization on MLP Training with Unnormalized Images

In section 4.6, we reported that training a MLP with unnormalized images results in a significant decrease in the accuracy of the model. Naturally, we wondered if regularization (L1/L2 regularization) would improve the performance of a MLP model trained with unnormalized Fashion-MNIST dataset. We used the same hyperparameters and layer dimensions as in the experiment performed in section 4.6. It was found that with L1 regularization, the accuracy of the MLP improved from 77.7% (accuracy of MLP trained with unnormalized images & no regularization) to 82.9%. With L2 regularization, the accuracy of the MLP is 81.3% which is also a significan increase compared to without regularization. The training cost curve for both L1 and L2 regularization is much steeper than and converges faster than that without regularization (Figure 36, 37, 38, 39). Overall, the results indicate that both L1 and L2 regularization improves the train/test performance of the MLP trained with unnormalized images.

4.11 Additional Experiment 2: Effect of Training Set Size on MLP Performance

While adjusting the hyperparameters of the MLP model before conducting the experiments, we also explored using different train/validation/test splits and noticed that the size of the training set had an effect on the test set accuracy. Therefore, we decided to perform an additional experiment to quantify this change. MLP models with same hyperparameters were trained with differently sized Fashion-MNIST training sets ranging from 10000 to 60000. It was found that as the training set size increases, the training cost curves converges to a smaller error value. The accuracy of the MLP was generally higher for larger training set sizes (most around 70-75%), however the accuracy of a 20000 sized training set (75.64%) was higher than that of a 30000 sized one (51.23%). Though it may also be because the training set sizes were taken seqeuntially (1 to 10000, 1 to 200000 etc.) so the MLP model may be better by chance.

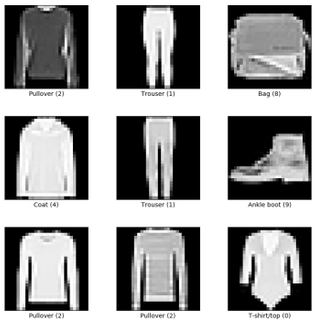
5 Discussion and conclusion

Our project delved into the comprehensive exploration of various aspects affecting the performance of MLP and CNN models on classification tasks using the Fashion-MNIST and CIFAR10 datasets. We investigated the influence of normalized data, CNNs, weight initialization, the number of hidden layers, activation functions, and regularization techniques on model accuracy. Notably, the combined usage of normalized data, CNNs, Kaiming weight initialization, multiple hidden layers, ReLu activation function, and L1 or L2 regularization provided the most performant models. Looking forward, exploring more CNNs improvements presents an avenue for future investigations. Incorporating techniques like batch normalization and examining the interplay of factors within more complex CNN architectures could offer insights into optimizing performance for image classification tasks.

5.1 Statements of contribution

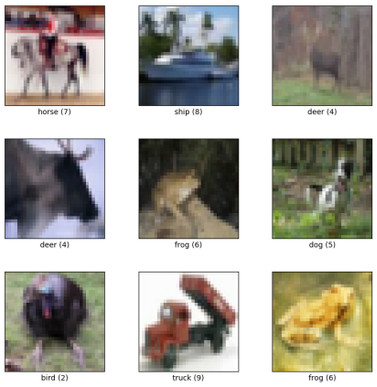
Each member of the team was tasked of doing all the experiments themselves, which include writing the code, running the tests. Then a week later we met to discuss about our findings and we selected snippets of code that were most promising. All members contributed to the code and the written report.

6 Appendix



1. T-Shirt/top
2. Trouser
3. Pullover
4. Dress
5. Coat
6. Sandal
7. Shirt
8. Sneaker
9. Bag
10. Ankle boot

Figure 9: Sample from Fashion MNIST



1. Airplane
2. Automobile
3. Bird
4. Cat
5. Deer
6. Dog
7. Frog
8. Horse
9. Ship
10. Truck

Figure 10: Sample from CIFAR-10

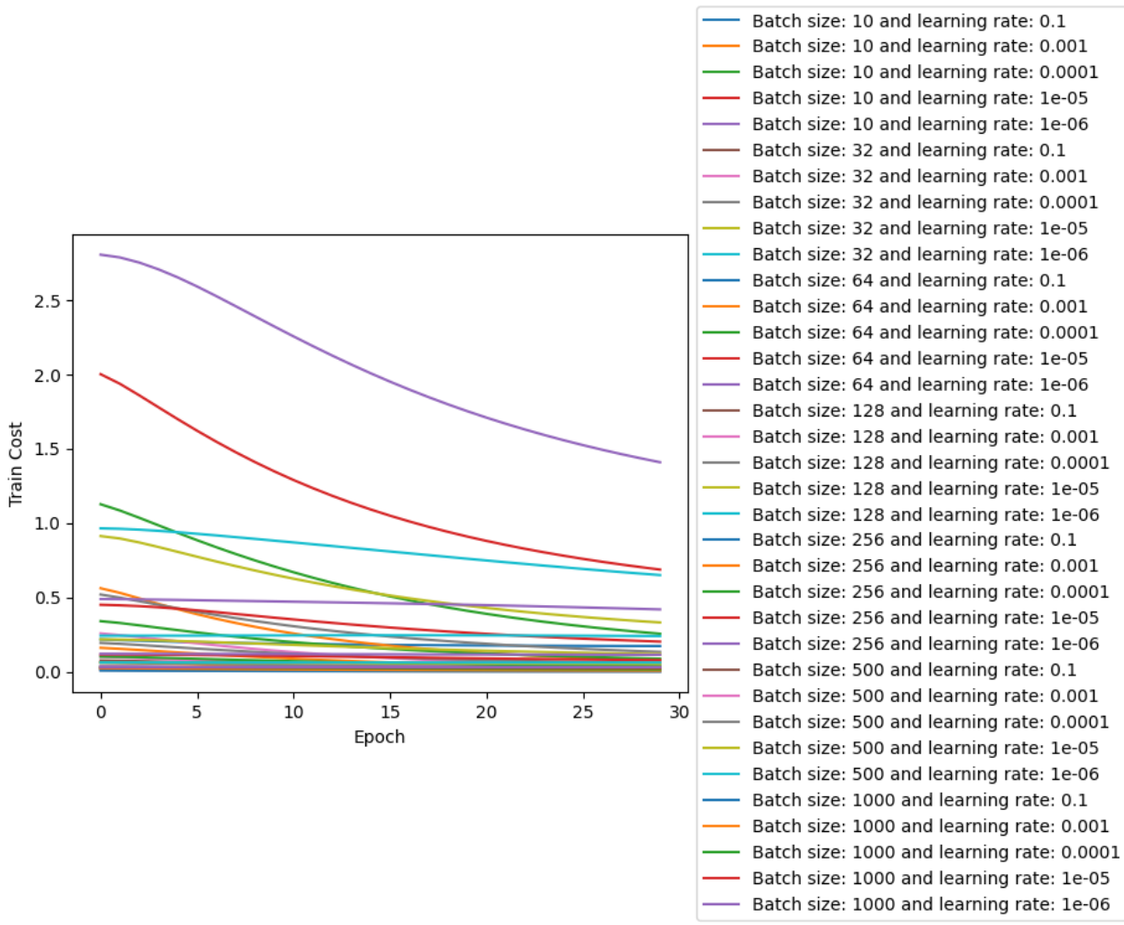


Figure 11: Training Cost Curve for MLP with Different Hyperparameters

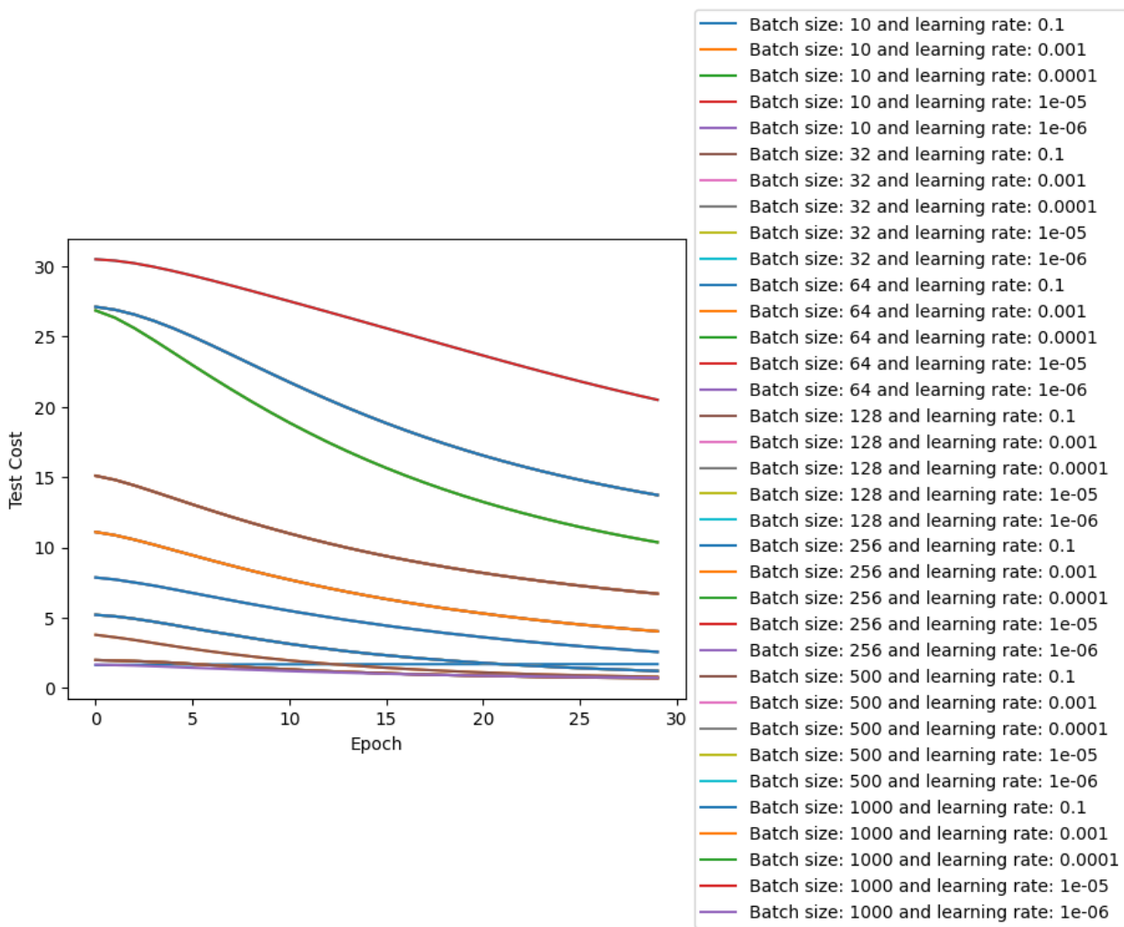


Figure 12: MLP Performance on Fashion-MNIST Test Set with Different Hyperparameters

Batch Size	Learning Rate	Accuracy
10	0.1	20.0%
10	0.001	80.2%
10	0.0001	76.0%
10	1×10^{-5}	71.0%
10	1×10^{-6}	54.1%
32	0.1	80.2%
32	0.001	76.0%
32	0.0001	71.0%
32	1×10^{-5}	54.1%
32	1×10^{-6}	79.8%
64	0.1	76.0%
64	0.001	71.0%
64	0.0001	54.1%
64	1×10^{-5}	79.8%
64	1×10^{-6}	78.9%
128	0.1	71.0%
128	0.001	54.1%
128	0.0001	79.8%
128	1×10^{-5}	78.9%
128	1×10^{-6}	74.1%
256	0.1	54.1%
256	0.001	79.8%
256	0.0001	78.9%
256	1×10^{-5}	74.1%
256	1×10^{-6}	63.4%
500	0.1	79.8%
500	0.001	78.9%
500	0.0001	74.1%
500	1×10^{-5}	63.4%
500	1×10^{-6}	36.4%
1000	0.1	78.9%
1000	0.001	74.1%
1000	0.0001	63.4%
1000	1×10^{-5}	36.4%
1000	1×10^{-6}	76.9%

Table 7: Accuracy of MLP Model Trained with Different Hyperparameters

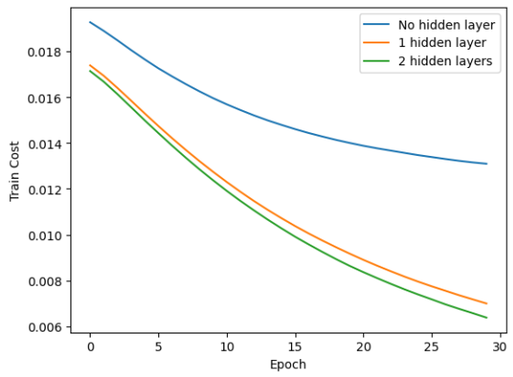


Figure 13: Training Cost Curve for MLP with Different Number of Hidden Layers on Fashion-MNIST Training Set

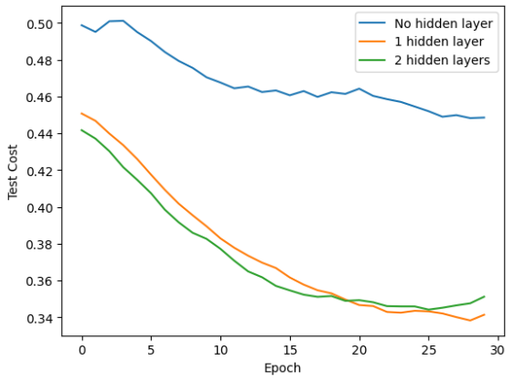


Figure 14: Test Performance for MLP with Different Number of Hidden Layers on Fashion-MNIST Dataset

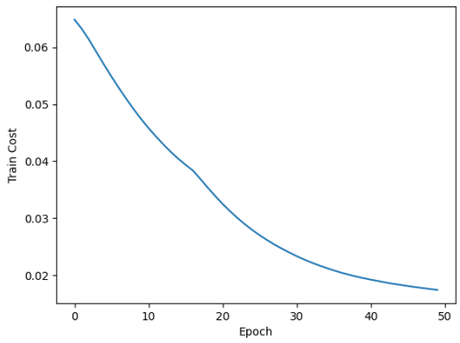


Figure 15: Training Cost Curve for MLP on Unnormalized Fashion-MNIST Dataset

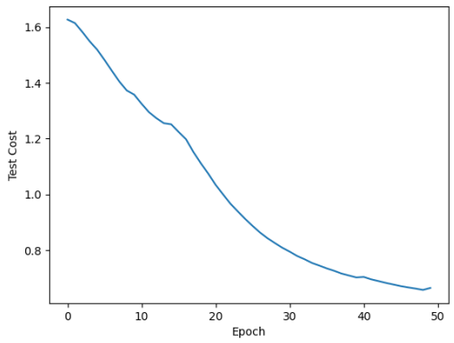


Figure 16: MLP Performance on Unnormalized Fashion-MNIST Test Set

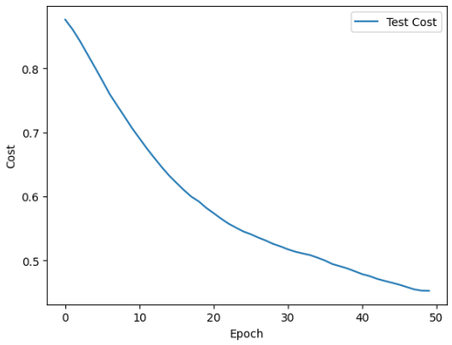


Figure 17: Training Curve for MLP on Normalized Fashion-MNIST Dataset

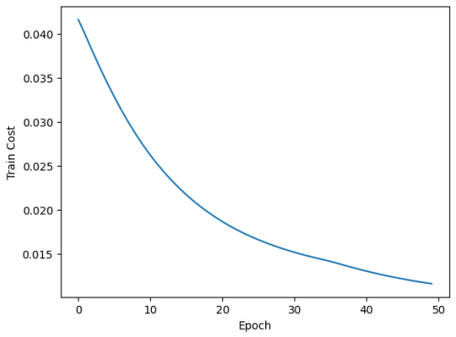


Figure 18: MLP Performance on Normalized Fashion-MNIST Test Set

The following set of five figures (Fig. 19 - 23) show images from the Fashion-MNIST that were incorrectly classified by the CNN.

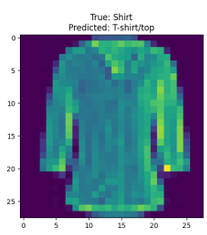


Figure 19: Shirt T-Shirt/top

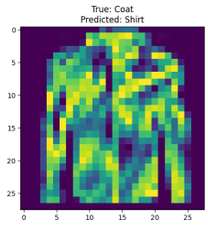


Figure 20: Coat Shirt

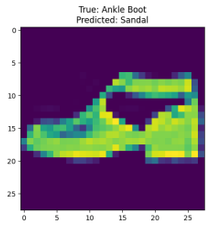


Figure 21: Ankle Boot Sandal

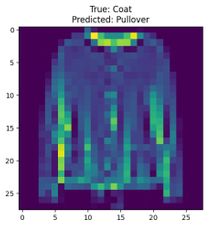


Figure 22: Coat Pullover

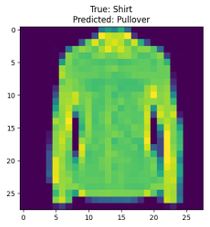


Figure 23: Shirt Pullover

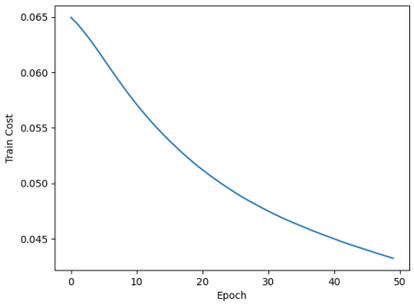


Figure 24: Training Performance for MLP on CIFAR10 dataset

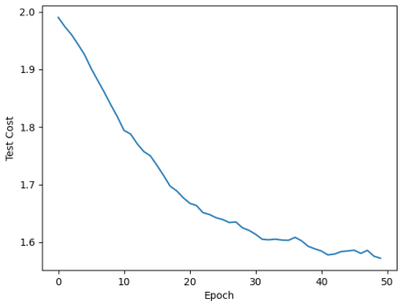


Figure 25: Test Performance for MLP on CIFAR10 dataset

The following set of five figures (Fig. 26 - 30) show example images from the CIFAR-10 that were correctly classified by the implemented CNN.

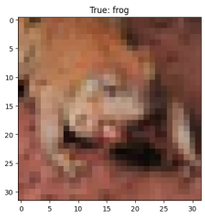


Figure 26: Frog

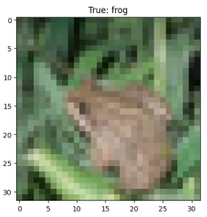


Figure 27: Frog

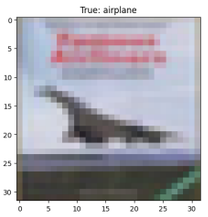


Figure 28: Plane

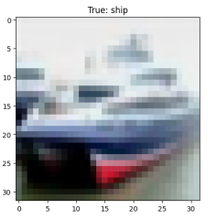


Figure 29: Ship

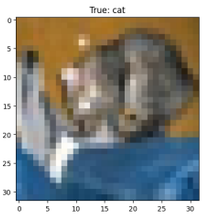


Figure 30: Cat

The following set of five figures (Fig. 31 - 35) show example images from the CIFAR-10 that were incorrectly classified by the implemented CNN.

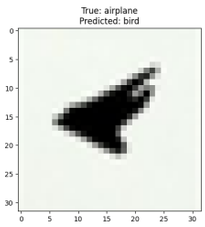


Figure 31: Bird

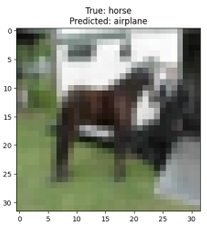


Figure 32: Plane

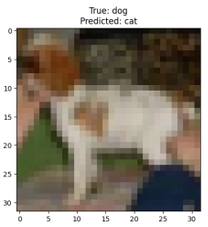


Figure 33: Cat

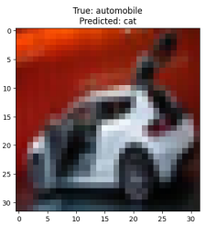


Figure 34: Cat

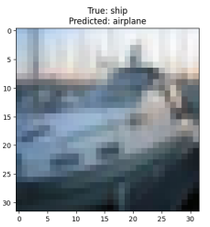


Figure 35: Plane

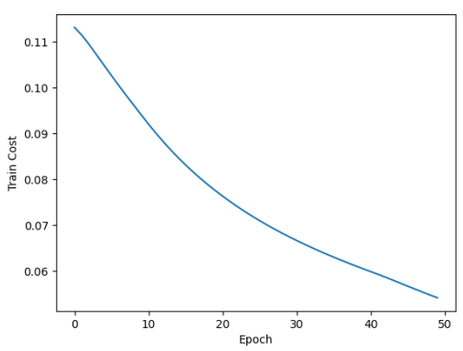


Figure 36: Training Cost Curve for L1 Regularized MLP on Unnormalized Fashion-MNIST Dataset

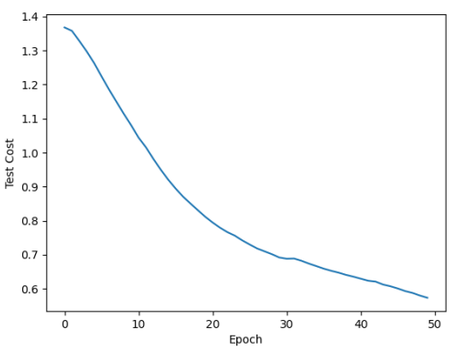


Figure 37: L1 Regularized MLP Performance on Unnormalized Fashion-MNIST Test Set

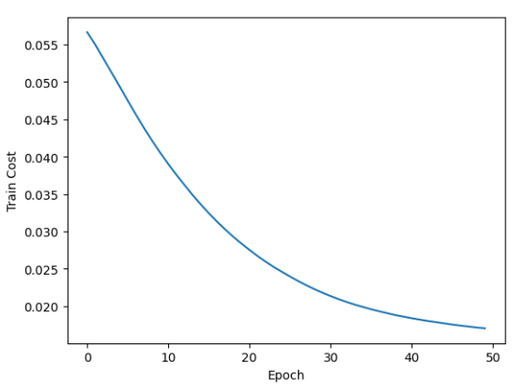


Figure 38: Training Cost Curve for L2 Regularized MLP on Unnormalized Fashion-MNIST Dataset

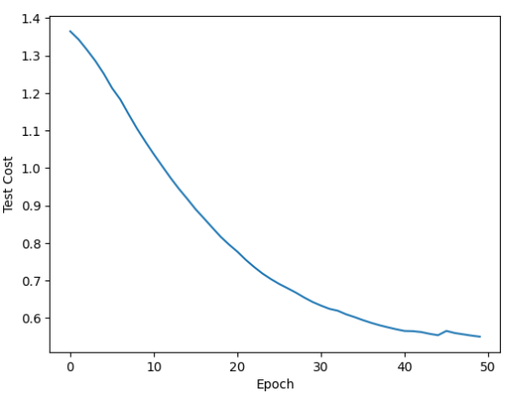


Figure 39: L2 Regularized MLP Performance on Unnormalized Fashion-MNIST Test Set

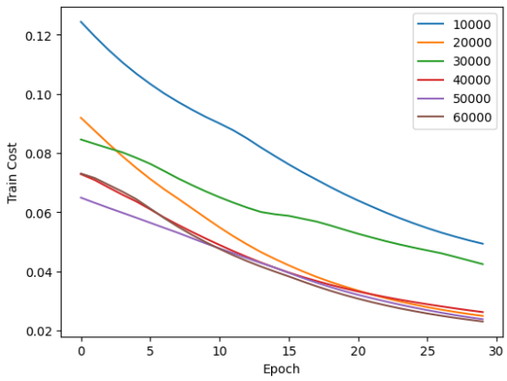


Figure 40: Training Cost Curve MLP Training with Differently Sized Fashion-MNIST Training Sets

Training Set Size	Accuracy
10000	44.93%
20000	75.64%
30000	51.23%
40000	71.63%
50000	79.34%
60000	77.65%

Table 8: Accuracy of Class Predictions for the Fashion-MNIST Test Set using MLP with Regularization

References

[1] O. M. Khanday, S. Dadvandipour, and M. A. Lone, "Effect of filter sizes on image classification in CNN: A case study on CFIR10 and Fashion-MNIST datasets," English, *IAES International Journal of Artificial Intelligence*, vol. 10, no. 4, pp. 872–878, Dec. 2021, Num Pages: 872-878 Place: Yogyakarta, Malaysia Publisher: IAES Institute of Advanced Engineering and Science. DOI: 10.11591/ijai.v10.i4.pp872-878. [Online]. Available: https://www.proquest.com/docview/2615646968/abstract/865D211F72334D1BPQ/1 (visited on 10/31/2023).

[2] R. C. Çalik and M. F. Demirci, "Cifar-10 Image Classification with Convolutional Neural Networks for Embedded Systems," in *2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)*, ISSN: 2161-5330, Oct. 2018, pp. 1–2. DOI: 10.1109/AICCSA.2018.8612873. [Online]. Available: https://ieeexplore.ieee.org/document/8612873 (visited on 10/31/2023).

[3] O. Nocentini, J. Kim, M. Z. Bashir, and F. Cavallo, "Image Classification Using Multiple Convolutional Neural Networks on the Fashion-MNIST Dataset," en, *Sensors*, vol. 22, no. 23, p. 9544, Dec. 2022, ISSN: 1424-8220. DOI: 10.3390/s22239544. [Online]. Available: https://www.mdpi.com/1424-8220/22/23/9544 (visited on 10/31/2023).

[4] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms," *CoRR*, vol. abs/1708.07747, 2017. arXiv: 1708.07747. [Online]. Available: http://arxiv.org/abs/1708.07747.

[5] A. Krizhevsky, "Learning multiple layers of features from tiny images," pp. 32–33, 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.