Mini Project 2: Classification of Image Data with Multilayer Perceptrons and Convolutional Neural Networks

Jonathan Halimi, Thien Pham, Rafid Saif

October 31, 2023

1 Abstract

The significance of machine learning algorithms in improving the accuracy of trend predictions and classifications is garnering increasing attention. This project focuses on a widely used method for image data classification. We will implement various Multilayer Perceptrons (MLPs) and Convolutional Neural Networks (CNNs), assessing their accuracy under different hyperparameter configurations. These models will be employed to classify data from the Fashion MNIST and CIFAR-10 datasets. Our findings emphasize the critical role of hyperparameter selection.

Contents

2 Introduction

The field of machine learning has witnessed remarkable advancements, particularly in the domain of image classification. Multilayer perceptrons (MLPs) and Convolutional neural networks (CNNs) have set new standards in achieving state-of-the-art performance on a wide range of image datasets. However, understanding the intricacies of neural network design and training remains a crucial step for aspiring machine learning practitioners.

In this mini-project, we embark on a journey to demystify the art of image classification using neural networks. We focus on two diverse datasets: Fashion MNIST, a collection of fashion items, and CIFAR-10, a dataset comprising ten distinct object classes. Our goal is to not only apply existing machine learning libraries but to also gain a deeper understanding of the underlying mechanisms by building our solutions from scratch when necessary.

3 Datasets

3.1 Fashion-MNIST

Fashion-MNIST is a dataset of Zalando's article images—consisting of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image, associated with a label from 10 classes [2]. We intend Fashion-MNIST to serve as a direct drop-in replacement for the original MNIST dataset for benchmarking machine learning algorithms. It shares the same image size and structure of training and testing splits.

3.2 CIFAR-10

The CIFAR-10 dataset was developed by researchers at the Canadian Institute For Advanced Research (CIFAR) Institute. It consists of 60,000 32x32 pixel colour images in 10 classes. There are 50,000 training images and 10,000 test images. The dataset is divided into five training batches and one test batch, each with 10,000 images. The test batch contains exactly 1000 randomly selected images from each class [1].

4 Result

4.1 Effects of weight initialization on MLPs

Five distinct MLPs were built to understand the effect of different weight initialization methods. Their findings and accuracy on the Fashion MNIST dataset are summarized in Table 1. The corresponding results are also visually presented in Figure 1a) and 2a), providing valuable insights into the influence of weight initialization techniques on model accuracy. Notably, the Xavier and Kaiming initialization methods demonstrated superior performance, underscoring the critical role of appropriate weight initialization in MLPs.

Table 1: Accuracy results with different weight initialization

Weight initialization	Zeros	Uniform	Gaussian	Xavier	Kaiming
Test accuracy	0.1	0.7828	0.8033	0.8754	0.8725

4.2 Effects of hidden layers on MLPs

We conducted an experiment involving three distinct MLP models, varying the number of hidden layers to assess their impact on accuracy. The data presented in Table 2 indicates that incorporating additional hidden layers enhances the model's capacity to capture intricate dataset relationships. Furthermore, the introduction of non-linearity parallels our previous use of Gaussian Basis Functions in A1, providing a more realistic fit to the data. By enabling the model to adapt to more complex relationships, we achieve heightened accuracy, as evident in Figure 1b) and 2b). Of course, it must be noted that given the simple dataset, one layer achieves results that are close to that of two layers, with the benefit of taking much less time to train.

Table 2: Accuracy results with numbers of hidden layers

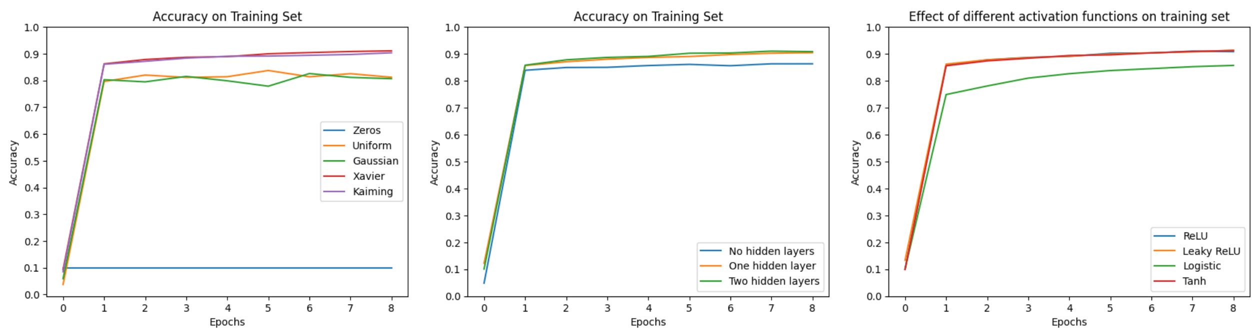
# of hidden layers	Test accuracy
Zero	0.8386
One	0.8719
Two	0.8755

Table 3: Accuracy results of different regularizations

Regularization	Test accuracy
None	0.8830
L1	0.8039
L2	0.8808

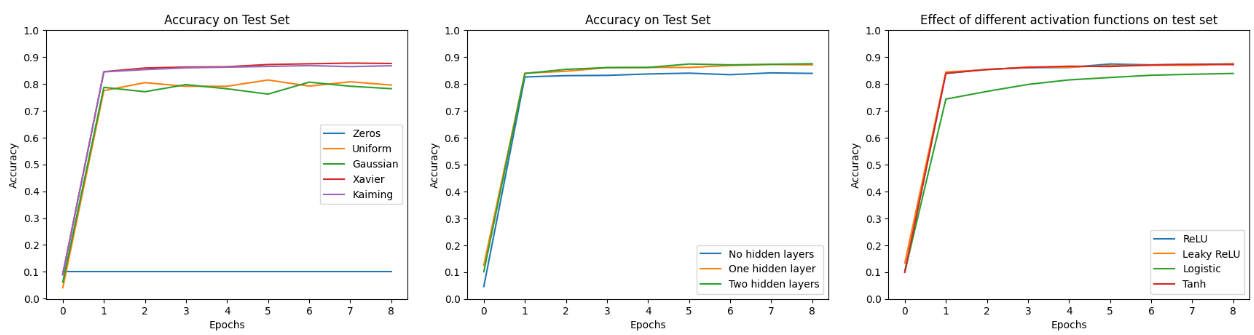
4.3 Activation functions and MLPs

We study the effects of the activation function on the quality of our results on the Fashion MNIST dataset. In this experiment, we use MLPs consisting of two hidden layers, each containing 128 units. The activations we explore include ReLU, Leaky ReLU, Logistic, and TanH. Figure 1c) and 2c) illustrate our findings and accuracy.



(a) weight initialization (b) hidden layers (c) activation functions

Figure 1: MLPs accuracy results of different factors on train set



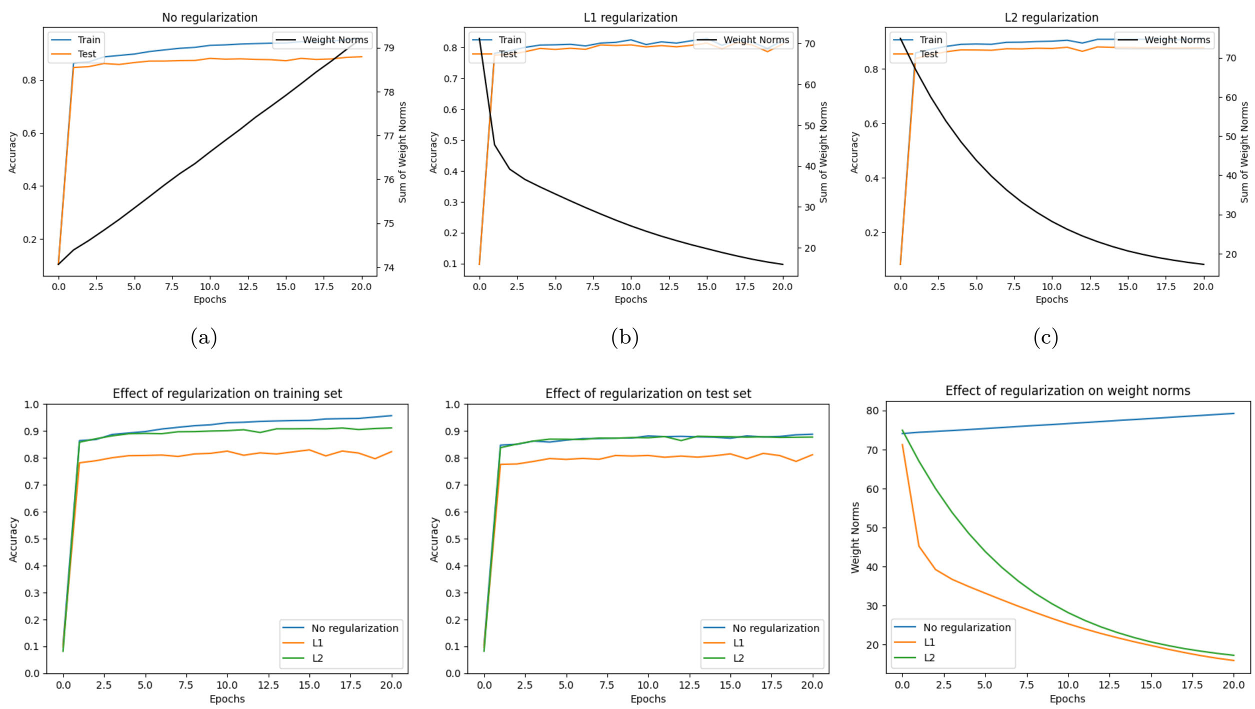
(a) weight initialization (b) hidden layers (c) activation functions

Figure 2: MLPs accuracy results of different factors on test set

The results show that ReLU, Leaky ReLU (with γ=0.05) and TanH have similar performances. In contrast, the Logistic activation function performs less effectively, possibly due to issues like the vanishing gradient problem and a lack of zero-centred output. ReLU, for example, does not have the vanishing gradient problem for positive values. TanH is zero-centred and performs better than the Logistic activation as well.

4.4 L1 and L2 regularization on MLPs

We study the effects of different regularization techniques on the quality of our results, as shown in Figure 3a), b), and c). In this experiment, we use L1, L2 and a combination of the two in our MLP which has 2 hidden layers each with 128 units. To better understand the effect of regularization on overfitting, we intentionally iterate enough times to ensure the model with no regularization starts to overfit. The accuracy results are found in Table 3. We see that using the same regularization constant, L1 is more aggressive in reducing the weights, and thus has less accuracy. L2 norm is less aggressive and achieves nearly the same accuracy as the unregularized model. Both regularized models do not exhibit overfitting over the training interval, while the unregularized model does.



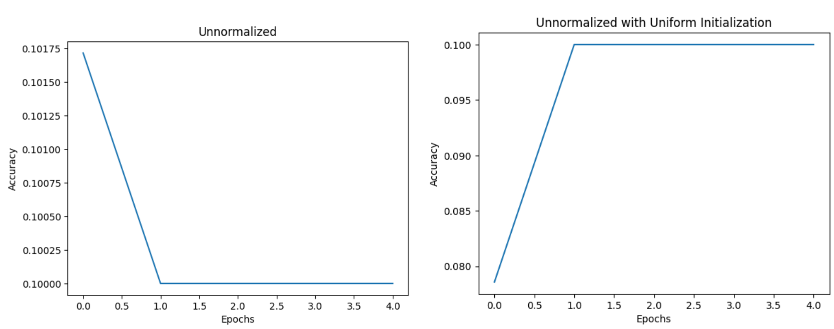
(a) (b) (c)

(d) (e) (f)

Figure 3: Accuracy results of (a) None regularization, (b) L1 regularization, (c) L2 regularization; and Effects of regularization on (d) training set, (e) test set, (f) weight norms

4.5 Normalization and MLPs

The MLP fails to converge with unnormalized data due to its lack of zero-centeredness and different feature scales. This, combined with positive weights, renders ReLU activations ineffective, they simply pass the input to the output, making the MLP less expressive, and reducing its power. Unnormalized images lead to slower convergence, lower accuracy, and instability, causing the model to perform significantly worse compared to using properly normalized data. See Figure 4 for details on both Kaiming and Uniformed initialization experiments.

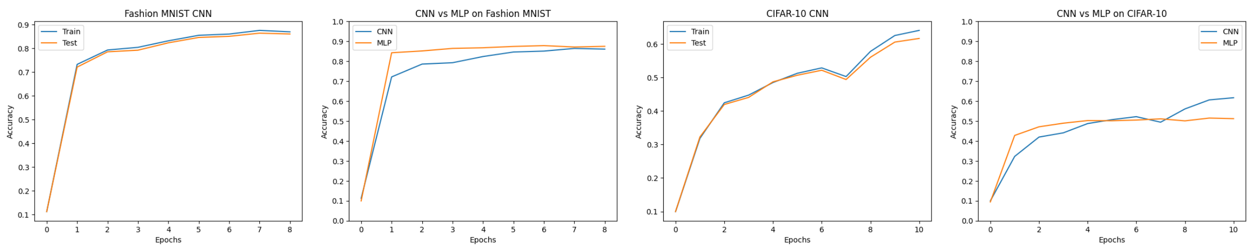


(a) Kaiming initialization (b) uniformed initialization

Figure 4: Accuracy results of unnormalized images

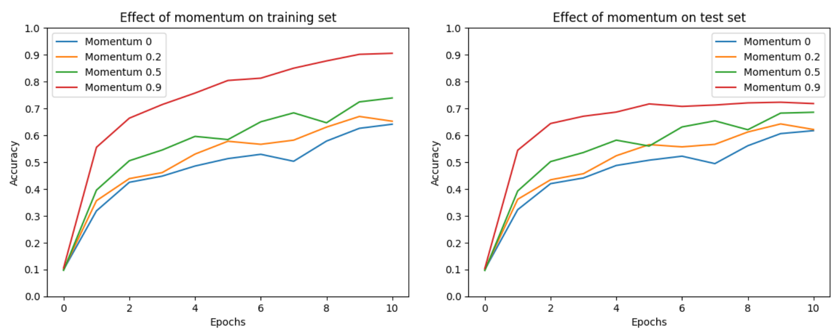
4.6 Convolutional neural networks (CNNs)

Constructing a CNN with two convolutional and two fully connected layers, each comprising 128 units and utilizing ReLU activation, yielded accuracy results displayed in Figure 5b). Interestingly, CNNs performance did not exhibit a substantial improvement over the MLP model for the Fashion dataset. This outcome was unexpected, as CNNs are specifically designed to leverage spatial relationships in images for more informative feature extraction compared to MLPs. For the CIFAR dataset, the plots in Figure 5d) were generated using 20,000 iterations, as utilizing 50,000 iterations resulted in overfitting. This adjustment led to a slight accuracy increase, as reflected in the results presented in Table 4.



(a) CNN - Fashion set (b) CNN vs MLP - Fashion set (c) CNN - CIFAR set (d) CNN vs MLP - CIFAR set

Figure 5: Accuracy results of CNN vs MLPs models on the two datasets



(a) train set (b) test set

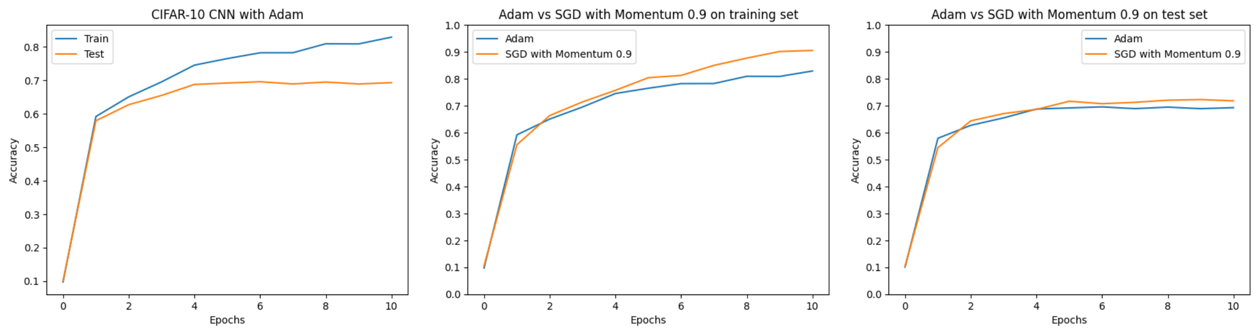
Figure 6: Effect of momentum on model accuracy

Table 4: Accuracy results of CNN model

		Epoch #	1	2	3	4	.	.	.	8
(a)	Fashion dataset	Train	0.7326	0.7931	0.8044	0.8318	.	.	.	0.8697
		Test	0.7212	0.7856	0.7924	0.8234	.	.	.	0.8606
(b)	CIFAR dataset	Train	0.3182	0.4246	0.4476	0.4852	.	.	.	0.5781
		Test	0.3226	0.4196	0.4408	0.4873	.	.	.	0.5607

Using an SGD optimizer on our CNN model. While working on the CIFAR-10 dataset we studied how the momentum factor affected our convergence speed, accuracy, and stability. The plots, illustrated in Figure 6, as well as the results recorded in Table 5, demonstrate higher momentum allows faster convergence, however, it may lead to overfitting the model. Additionally, we compared this approach with the utilization of the Adam optimizer. As shown in Table 6, SGD with momentum 0.9 and the Adam optimizer have fairly similar performances.

.



(a) Adam Optimizer accuracy (b) SGD vs. Adam Optimizer (trainset) (c) SGD vs. Adam Optimizer (testset)

Figure 7: SGD vs. Adam Optimizer effects on model accuracy

Table 5: Effects of momentum variables on accuracy

Momentum	0	0.2	0.5	0.9
Test accuracy	0.6169	0.6213	0.6855	0.7179

Table 6: Accuracy results of different optimizers

	SGD with Momentum 0.9	Adam
Test accuracy	0.7179	0.6928

5 Discussion and Conclusion

In this mini-project, we embarked on a journey to explore the world of image classification using multilayer perceptrons (MLPs) and convolutional neural networks (CNNs). Our investigation began with the acquisition of two image datasets, Fashion MNIST and CIFAR-10, setting the stage for our experiments. We delved into the world of MLPs, implementing them from scratch with ReLU activations and two hidden layers, each containing 128 units. Then we explored CNNs and assessed the effects of different configurations and techniques as compared to MLPs. Through a series of experiments and analyses, we gained valuable insights into the impact of various factors on model performance. Key findings emerged from our experiments:

- Effect of Weight Initialization: Weight initialization impacts training dynamics and accuracy, with Xavier and Kaiming outperforming others
- Impact of Network Depth: Deeper MLPs enhance accuracy by capturing complex features
- Activation Functions Matter: Comparing different activation functions, including ReLU, Leaky ReLu, and Logictis, and Tanh, highlighted the significance of this choice.
- Regularization Strategies: L1 and L2 regularization control overfitting and improve generalization.
- Unnormalized Data: Unnormalized data led to slower convergence and reduced accuracy, reaffirming the value of normalization.
- CNNs vs. MLPs: CNNs slightly exhibited superior performance, reaffirming their effectiveness in image-related tasks due to their ability to capture spatial relationships.
- Optimization Algorithms: Optimizer choice affects convergence speed and stability. In our case, it is Momentum in SGD and the use of the Adam optimizer

In conclusion, this project highlights the significant influence of design decisions on image classification model effectiveness. Our observations emphasize the importance of making deliberate choices in neural network design and training.

6 Statement of Contributions

- Jonathan Halimi - ran experiments on the models and wrote this report.
- Thien Pham - implemented CNN model, computed dataset statistics and wrote this report.
- Rafid Saif - implemented the MLP in Python, and ran experiments on MLP and CNN.

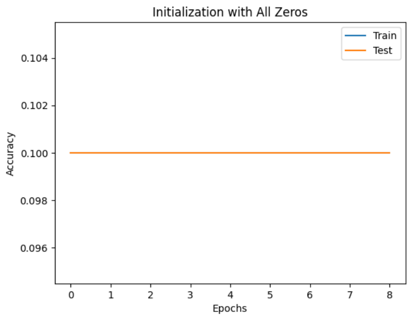
7 Bibliography

References

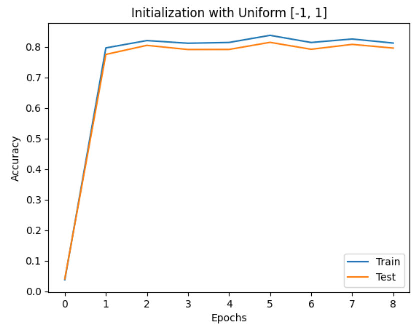
[1] *CIFAR10 Dataset*. URL: https://pytorch.org/vision/main/generated/torchvision.datasets.CIFAR10.html.

[2] *Fashion MNIST Dataset*. URL: https://pytorch.org/vision/stable/generated/torchvision.datasets.FashionMNIST.html.

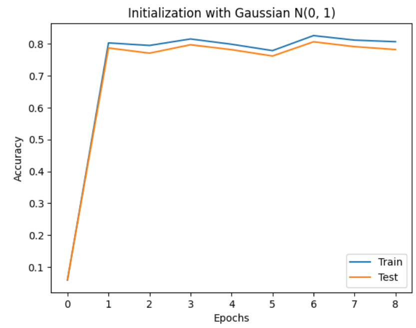
8 Appendix A - List of Figures



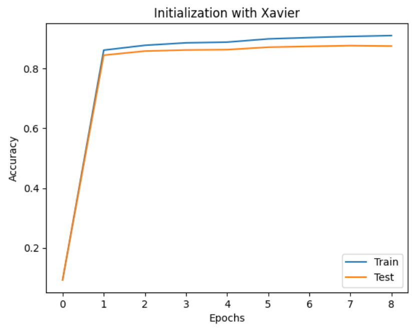
(a)



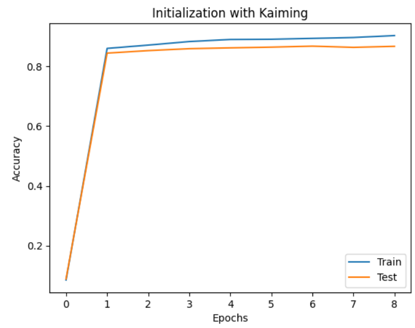
(b)



(c)

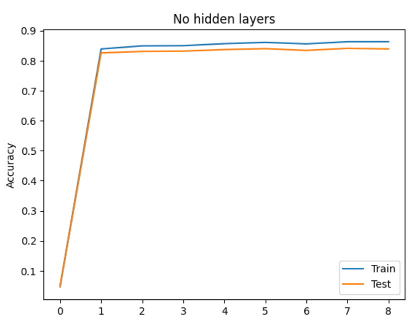


(d)

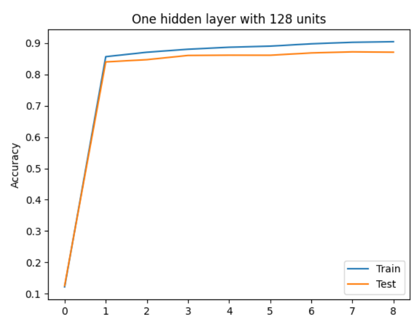


(e)

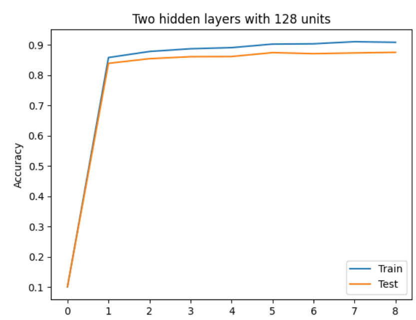
Figure 8: Effects of different weight initialization on MLPs: (a) no initialization, (b) uniformed, (c) gausian, (d) xavier, and (e) kaiming



(a)

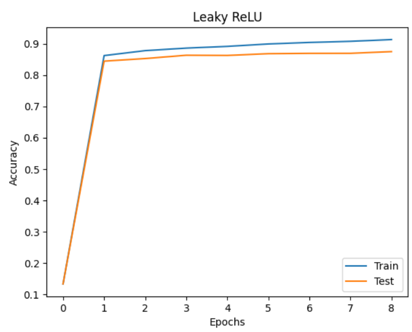


(b)

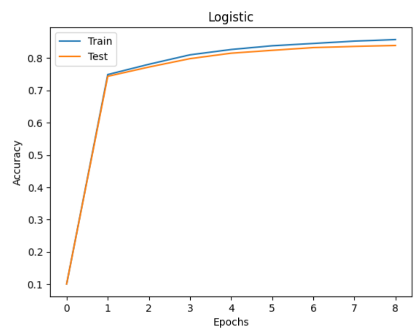


(c)

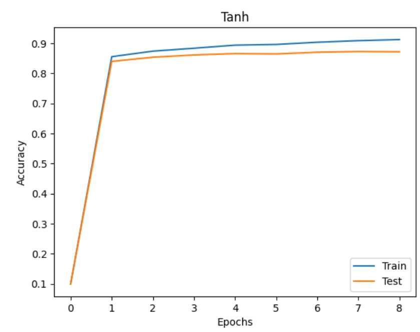
Figure 9: Effects of numbers of hidden layers on MLPs (a) zero hidden layers, (b) one hidden layer, (c) two hidden layer



(a)

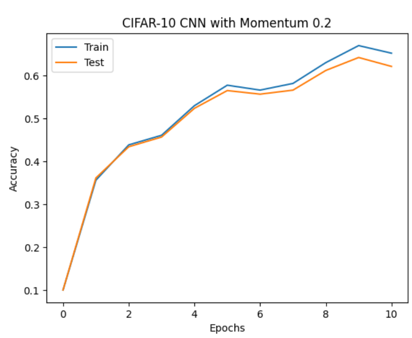


(b)

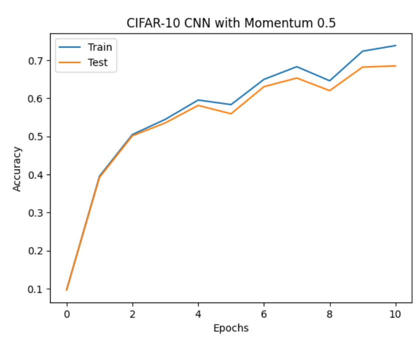


(c)

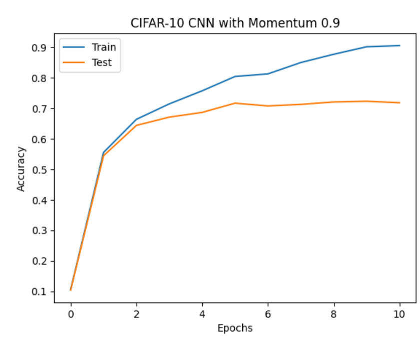
Figure 10: Effects of different activation fuction layers on MLPs (a) Leaky ReLU, (b) Logistics, (c) Tanh



(a) Momentum 0.2



(b) Momentum 0.5



(c) Momentum 0.9

Figure 11: Effect of momentum on model accuracy