Applied Machine Learning: Mini-Project 2

JAN TIEGGES, RISHABH THANEY, JONATHAN COLAÇO CARR

ABSTRACT: This report provides an in-depth analysis of Multi-Layer Perceptrons (MLPs) and Convolution Neural Networks (CNNs) for image classification. Using the Fashion MNIST and CIFAR-10 datasets, we examine how the performance of different MLPs varies according to several variables and design decisions, including depth, activation function and initialization method. Compared to a CNN with two convolutional layers and two fully connected layers, the MLP consistently underperforms. On the FashionMNIST dataset, the best CNN achieved an accuracy of 91.4% while the best MLP only achieved an accuracy of 88.61%. The discrepancy between the MLP and the CNN was even larger on the CIFAR-10 dataset, where the MLP achieved a test accuracy of only 50.11% compared to the CNN's 71.7%. Our findings are consistent with the theory of image classification discussed in class and suggest that the CNN is fundamentally better suited for image data.

1. Introduction

Learning from visual data is a fundamental problem in deep learning. The aim of this project was to gain hands-on experience in implementing a deep learning model, the Multi-Layer Perceptron (MLP) and to see how it compares to Convolutional Neural Networks (CNNs) on image classification tasks. We compared these models on the Fashion MNIST [XRV17] and CIFAR-10 [KNH08] datasets, both of which have been widely used to compare image classification models [Res, wC].

For both models, we performed a suite of experiments to investigate the effect of various design choices on overall performance and speed of convergence. For the MLP we considered how the model's performance changed with respect to the effect of the weight initialization method, number of hidden layers, choice of activation function, regularization method and data preprocessing steps. We then compared the MLP to a CNN with two convolutional layers and two fully connected layers. Our best MLP achieved accuracy scores of 88.61% on the FashionMNIST dataset and 50.11% on CIFAR-10 dataset, using the Xavier weight initializer, two hidden layers and ReLU activation functions. The CNN was significantly better than the MLP on both datasets, achieving test accuracies of 91.4% and 71.7% on the FashionMNIST and CIFAR-10 datasets, respectively. These results indicate that the convolution operation significantly improves the performance of a model for image classification tasks, as is consistent with the theory presented in class [PSa].

2. Datasets

This section highlights a few key findings for both datasets. The rest of the data analysis is available in our code.

Fashion MNIST Dataset. The Fashion MNIST Dataset [XRV17] consists of 70 000 images of fashion products sourced from Zalando, a European fashion website. Each image is a 28x28 pixel grayscale image falling into one of 10 image categories. There are 7 000 images for each category. This dataset was intended as a more challenging image classification task than the MNIST handwritten digits dataset [Den12]. As in the original paper [XRV17] we split the dataset into 60 000 training samples and 10 000 testing samples, ensuring an equal distribution among classes in both the training and testing set. The top two plots in Figure 1 show sample data points and the PCA analysis we conducted on this dataset. As highlighted by the PCA analysis (shown in the top right plot), the magnitude of the eigenvectors drops sharply after only a handful of dimensions. Thus, we anticipate that low rank approximations of the FashionMNIST images will lead to fair results, as confirmed by Experiment 5+.

Ethical Concerns. This dataset was collected from a European clothing website and is not representative of all clothing styles. Thus, clothing which does not adhere to typical European trends may be easily mis-classified by models that are trained on the Fashion MNIST dataset [OAI+21].

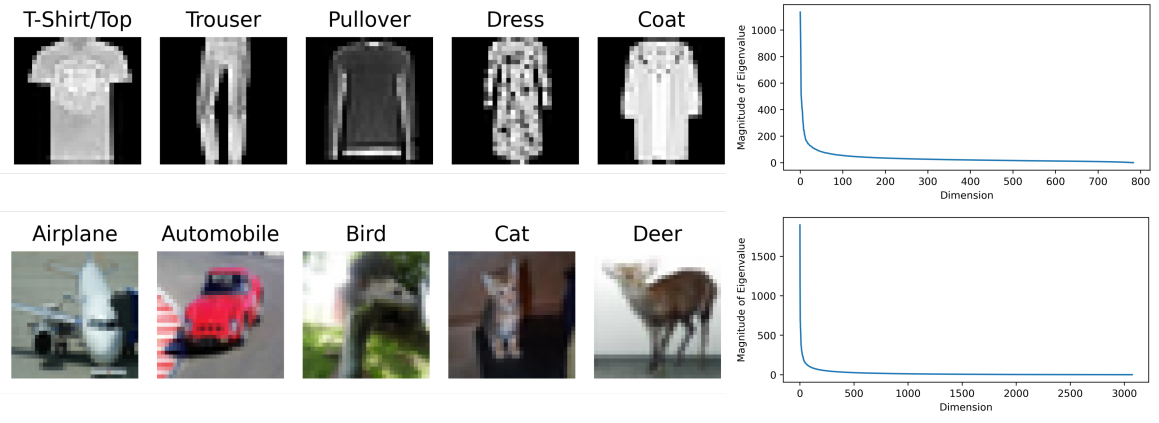


Figure 1. Top row: Five sample data points (left) and the magnitude of each eigenvalue in the training data matrix (right) for the Fashion MNIST dataset. Bottom row: Five sample data points (left) and the magnitude of each eigenvalue in the training data matrix (right) for the CIFAR-10 dataset.

CIFAR-10 Dataset. The CIFAR-10 Dataset [KNH08] is a subset of the Tiny Images Dataset [TFF08] and consists of 60 000 images. Each image is a 32x32 colored image labelled from one of 10 categories. There are 6 000 images for each category. We adopted the convention [KNH08] of splitting the dataset into 50 000 training samples and 10 000 testing samples, ensuring that each class had an equal distribution of samples in both the training and testing set. The bottom row of Figure 1 shows sample data points and the PCA analysis for the CIFAR-10 dataset. As with the FashionMNIST dataset, the magnitude of the eigenvalues of the data matrix drops sharply after only a handful of eigenvectors.

Ethical Concerns. The Tiny Image Dataset contained offensive judgements, derogatory terminology and systematic biases targeting marginalized communities [PB20] and has been redacted. While the CIFAR-10 dataset has been reviewed by humans, we are not aware of any audit of the CIFAR-10 dataset that confirms whether the subset of CIFAR-10 images contain the harms of the Tiny Images dataset.

Data Preprocessing. We applied min-max scaling to both datasets, ensuring that the pixel values lay in $[-1, 1]$. For the MLP we encoded each image as a one-dimensional vector.

3. Results

In this section we present the results of our experiments. The first five experiments are specific to the MLP and were carried out on the FashionMNIST dataset. In the remaining experiments, we compared the MLP to the CNN on both the FashionMNIST and CIFAR-10 datasets.

MLP Hyperparameter Search. We tested various hyperparameters for an MLP with 1 hidden layer, ReLU activation and Kaiming initialization by splitting the training data into a training and development set. We selected baseline parameters by choosing the parameters which had the best validation accuracy and used them subsequently in all our experiments. The chosen parameters were: a learning rate of 0.01, a momentum of 0.95, a learning rate decay of 1×10^{-7}, a batch size of 128 and 25 training epochs (except for Experiment 1).

Experiment 1: Weight Initializers. In the first experiment, we explored five different weight initialization methods for an MLP with a single hidden layer. The weights were initialized as (1) all zeros, (2) Uniform [-1, 1], (3) Gaussian N(0,1), (4) Xavier [GB10] and (5) Kaiming [HZRS15]. Looking at the training curve and the final accuracy of (1) in Figure 2, we can see that the model does not learn anything at all. This is to be expected, because when all weights are initialized with zeros, all neurons in the hidden layer are updated with the zero values during backpropagation. All other models succeed in learning from the data, but there are large differences in convergence speed and final accuracy, as can be seen in Table 1. The model with Gaussian weight initalization takes the longest time to converge, with the uniform model also achieving better final accuracy. The models initialized with Xavier and Kaiming show very similar results and largely outperform all other methods, with the Xavier model performing slightly better than Kaiming in this experiment. Both of these methods are specifically designed to preserve the variance of activations across layers, which is critical for training deep networks [NBS22]. The Xavier and Kaiming weight initializers were intended for the neural network architecture, which is a possible explanation for why the Gaussian N(0,1) method may not be effective, since the standard deviation is not appropriately set. It should also be noted that in all subsequent experiments we used the Kaiming initialization for the MLPs and that we used 25 training epochs instead of 50, as the models with Kaiming weight initialization stabilized after 25 epochs.

All zeros	**Uniform [-1, 1]**	**Gaussian N(0,1)**	**Xavier**	**Kaiming**
0.1	0.8535	0.8295	0.8861	0.8673

TABLE 1 *Test accuracies of MLP with various weight initialization methods on the Fashion MNIST dataset.*

Experiment 2: Number of Hidden Layers. In the next experiment, we looked at the effect of network depth, considering MLPs with none, one, and two hidden layers. Observing the training curves of the different models in Figure A.1, the effect of hidden layer becomes clear. As discussed in the lecture, the MLP without hidden layers can only correctly classify linear separable data, which is why it clearly performs worst (Test Accuracy: 82.91%) [PSc]. The advantage of the model with two hidden layers is only marginal in this case, but still present (Test Accuracy 87.62% vs 86.46%). In general, these results are to be expected, as the non-linearity introduces the capacity for the model to capture intricate patterns in the dataset that a linear model might not catch [PSc]. However, after a certain depth, the improvements might plateau or even decrease due to overfitting.

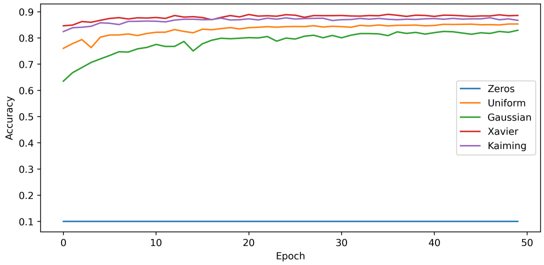


Figure 2 *Test accuracy over training time of MLPs with different weight initialization methods on the Fashion MNIST dataset.*

Experiment 3: Activation Functions. Subsequently, we investigated the effect of the activation function on MLP performance. For this experiment, we compared the accuracy of MLPs trained with ReLU, Logistic and hyperbolic tangent (TanH) activations. The model with ReLU activations performed the best, recording a test accuracy of 87.77%. This model was closely followed by the model with TanH activation which had a test accuracy of 87.11%. The model with logistic activation had the lowest performance with a test accuracy of 83.91%. The ReLU activation function is known for its computational efficiency and its ability to deal with the vanishing gradient problem [PSc]. The TanH function is similar to the logistic function, but outputs values between -1 and 1. It often performs better in practice than the logistic function because its output is zero-centered [PSc]. The logistic function is susceptible to the vanishing gradient problem, especially for deep networks, which slows down training [PSc]. Our results are consistent with material presented in class regarding these activation functions and show that the ReLU function is the best choice for our experiments.

Experiment 4: L1 and L2 Regularization. In the next experiment, we looked at the effect of regularization on performance. After performing a hyperparameter search we selected $\lambda = 0.001$ for the regularization proportion parameter. The model with L1 regularization negatively impacts the performance, with a test accuracy of 82.48%. The models without and with L2 regularization had similar test accuracy (None: 88.03%, L2: 87.75%). As shown in Figure A.2, adding L1 regularization also caused the training and testing accuracy to be unstable during training. This indicates that it might be too strong or not suitable for this particular problem. It can also be seen from the training curves in Figure A.2 that the model without regularization shows more signs of overfitting than the model with L2 regularization. Without L2 regularization, the difference between the training and testing accuracy increases at a higher rate during training. One other possible explanation for the fact that the final performance of the L2 regularized model is comparable to the model without regularization is that we train our models with learning rate decay. This already has a regularizing effect, making the effect of L2 regularization perhaps less relevant.

Experiment 5+: Unnormalized Images and PCA Reductions. In our next experiment, we considered the effect of normalization and PCA reduction on the MLP's performance. The MLP trained on unnormalized images performed significantly worse on the FashionMNIST data, achieving only a 10% test accuracy. This indicates that model is not able to learn on unnormalized data and highlights to importance of data normalization, as discussed in class [PSd]. Next, we considered the effect of dimensionality reduction on the MLP's performance. For the FashionMNIST dataset, we compared the performance of the MLP when trained on the original images with its performance when trained on PCA-reduced images. For the PCA-reduced Fashion MNIST images, we considered rank 10, rank 100, and rank 200 approximations to the original data. For the CIFAR-10 images, we considered rank 50, rank 500 and rank 1 000 approximations to the original data. As shown in Figure A.3, for both datasets we found that the model trained on low rank approximations of the data were comparable to the model trained on the un-reduced data. In fact, the MLP trained on the rank 200 approximation of the Fashion MNIST data slightly outperformed the MLP trained on the full rank data, achieving a final accuracy score of 87.84% instead of 87.66%. These results are consistent with our original data analysis, which showed that only a small fraction of the principal components contain a large majority of the information in the training datasets.

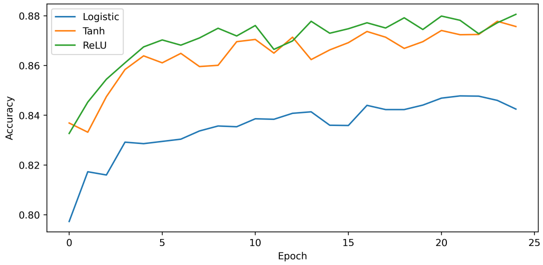


Figure 3 *Test accuracy over training time of MLP with different activation functions*

Experiment 6: FashionMNIST - CNN vs MLP. We then trained a CNN on the FashionMNIST dataset, for which we first performed a grid search with 288 combinations of different architecture and optimization parameters, including kernel size, output channels and learning rate. We selected the our CNN parameters based on the top 10 results on a validation set. The two convolutional layers were configured with 32 and 64 output channels respectively, a kernel size of 5, a stride of 1, and padding of 2. We observed that momentum and learning rate had a big influence on model performance, with a learning rate of 0.01 consistently achieving better results than 0.001. The effect of momentum will be investigated further in Experiment 8. In addition, the largest batch size (64) performed the worst on average. The impact of the kernel size and padding, on the other hand, were not as significant, with fair results being achieved using different values. A batch size of 32, learning rate of 0.01, and momentum of 0.9 were used for subsequent training over 5 epochs. The performance trajectory over the training period in Figure A.4 shows that our model does not overfit and that the final score on the test set of 91.4% outperforms all previous MLP models. This was to be expected as CNNs are specifically designed to extract features from images [PSa]. An important side note, however, is that the CNN takes more time to train due to the architecture's complexity.

Experiment 7: CIFAR-10 - CNN vs MLP. We then applied the CNN and our best performing MLP to the CIFAR-10 dataset. Both models perform significantly worse on the CIFAR-10 dataset, with the CNN having a test accuracy of 71.7% and the MLP a terrible 50.11%. Figure A.5 demonstrates that the MLP model failed to generalize to testing data and overfit strongly, which can be seen from the steadily increasing training accuracy at constant test accuracy. Perhaps stronger regularization could have yielded some performance gains here. The CNN, on the other hand, seems to generalize reasonably well, even if it shows small signs of overfitting at the end of the training. The spikes in the training curves in Figure A.5 can be attributed to our method of reporting the training error. At each iteration, the training error is reported for the batch that is used to update the gradient, rather than for the full training set. Therefore, the reported training loss has a higher variance between timesteps. Overall, a worse performance on the CIFAR-10 dataset was to be expected, since it is much more complex due to slightly larger images (32x32 instead of 28x28) and the fact that images are colored.

Experiment 8: CNN - SGD with momentum vs Adam. For the CNN, we then investigated the effect of the Stochastic Gradient Descent (SGD) momentum parameter and compared it to the Adam [KB14] optimization method. As Table 2 shows, the best momentum value was 0.9, which is a typical value for this parameter [PSb]. The momentum value in SGD plays a crucial role in the stability of learning, helping the optimizer avoid local minima in the loss landscape [PSb]. We can see from the training curves in Figure A.6 that a very high momentum (e.g. 0.99) can lead to oscillations and instability. Adam is known for its fast convergence and stable training [KB14], which is reflected in our experiments, where it initially learns the fastest and is relatively stable. In the end, both the SGD and Adam optimization methods lead to similar results.

SGD ($\beta = 0.0$)	**SGD** ($\beta = 0.5$)	**SGD** ($\beta = 0.75$)	**SGD** ($\beta = 0.9$)	**SGD** ($\beta = 0.99$)	**Adam**
0.5669	0.7081	0.7015	0.7159	0.2477	0.7142

TABLE 2 *Final test accuracies of CNN with various optimizers, trained on the CIFAR-10 dataset.*

4. Discussion and Conclusion

We have provided a thorough analysis of MLP and CNN design choices for image classification tasks. Our findings are consistent with the theory deep learning and image classification discussed in class. For instance, we found that deeper MLPs outperformed shallow ones, as they can learn more complex features from the training data. It was also interesting to note that MLPs trained on low-rank approximations of the training data were able to perform well, and even lead to improved performance in some cases. In addition, our experiments conclusively show that the CNN out-performs MLPs for image classification tasks, both in terms of overall performance as well as the number of epochs required to train. Our findings suggest that the CNN architecture is inherently better suited for image classification tasks and we conclude that the MLP is not well-adapted to classify colored images.

From our analysis there are many interesting avenues for future work. Since the MLP performance significantly deteriorates for colored images, it would be interesting to investigate whether the MLP's performance improves after applying a grayscale transformation to the CIFAR-10 images. For the CNN, it would be interesting to see if the performance improves when augmenting the image dataset with cropped and rotated images. Another area of future exploration would be to obtain comparable scores to the latest benchmarks [Res, wC] by using pre-trained model weights. Lastly, it would be worthwhile to visualize which features are encoded in each convolutional layer in order to provide a better understanding of the inner workings for the CNN.

Statement of Contribution. All members contributed to experiments and report writing. RT and JCC performed the data analysis. JT and JCC implemented the models and optimizers.

References

Den12. Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

GB10. Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

HZRS15. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

KB14. Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

KNH08. Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). 2008.

NBS22. Meenal V Narkhede, Prashant P Bartakke, and Mukul S Sutaone. A review on weight initialization strategies for neural networks. *Artificial intelligence review*, 55(1):291–322, 2022.

OAI^{+}21. Wuraola Fisayo Oyewusi, Olubayo Adekanmbi, Sharon Ibejih, Opeyemi Osakuade, Ifeoma Okoh, and Mary Salami. Afrifashion1600: A contemporary african fashion dataset for computer vision. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, volume 1, pages 3963–3967, 2021.

PB20. Vinay Uday Prabhu and Abeba Birhane. Large image datasets: A pyrrhic win for computer vision?, 2020.

PSa. Isabeau Prémont-Schwarz. Convolutional neural networks.

PSb. Isabeau Prémont-Schwarz. Gradient descent.

PSc. Isabeau Prémont-Schwarz. Multilayer perceptron.

PSd. Isabeau Prémont-Schwarz. Regularization.

Res. Zalando Research. A database for fashionmnist benchmarks.

TFF08. Antonio Torralba, Rob Fergus, and William T. Freeman. 80 million tiny images: A large data set for non-parametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):1958–1970, 2008.

wC. Papers with Code. State of the art for cifar10 datasets.

XRV17. Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

A. Appendix

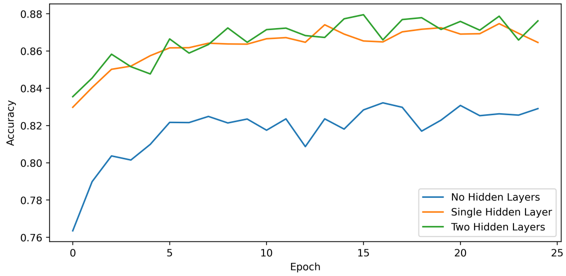


Figure A.1. Test accuracy over training time of MLP with different depths on the Fashion MNIST dataset.

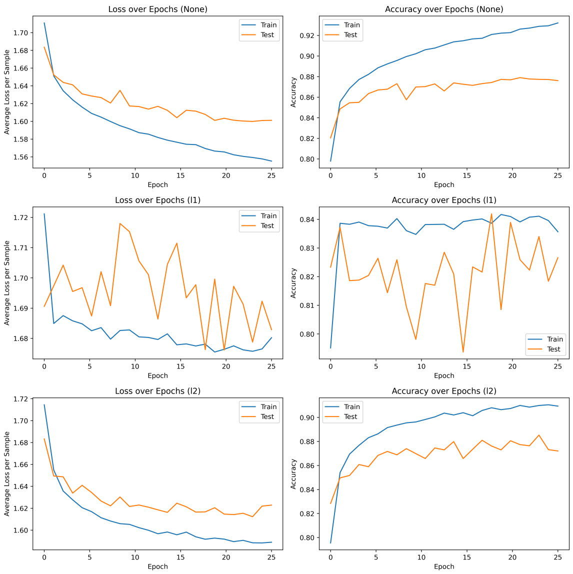


Figure A.2. Average loss per sample (left) and accuracy (right) over training time of MLP with different initialization methods on the Fashion MNIST dataset.

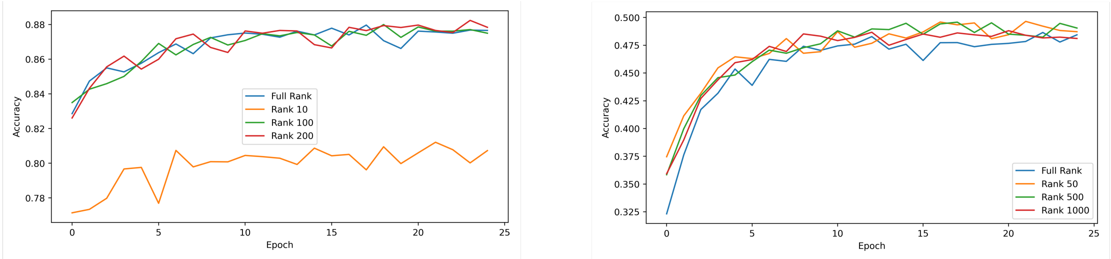


Figure A.3. Test accuracy for the MLPs trained on PCA-reduced versions of the FashionMNIST data (left) and CIFAR10 data (right).

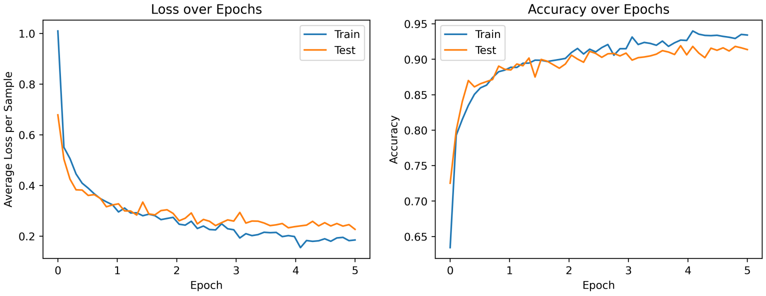
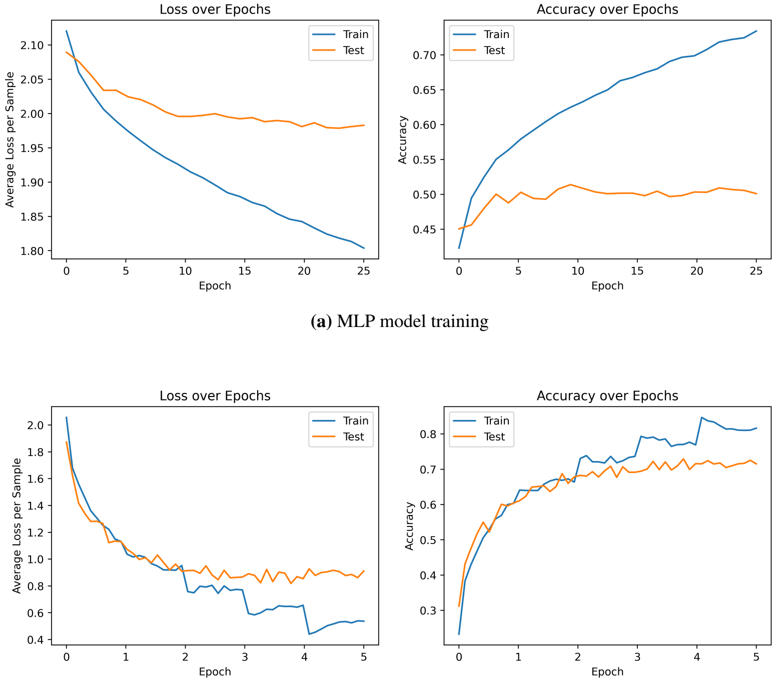


Figure A.4. Average loss per sample (left) and accuracy (right) over training time of CNN on FashionMNIST dataset.



(a) MLP model training

(b) CNN model training

Figure A.5. Training trajectories for models trained on CIFAR10 dataset.

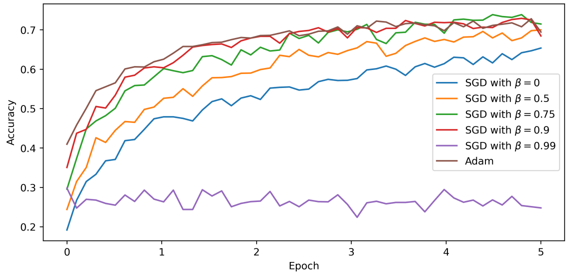


Figure A.6. Test accuracy versus training time of CNNs with different optimizers and momentum values for the CIFAR-10 dataset.