Multi-Layer Perceptron and Convolutional Networks: An Implementation and Analysis

Eamonn Lye

ABSTRACT

The goal of this project was to create multi-layer perceptron models as well as a convolutional neural network in order to classify the MNIST-fashion dataset, which contains thousands of labeled images of distinct articles of clothing. Furthermore, I observed the effect of increasing the number of hidden layers, different activation functions as the effectiveness of a convolutional neural network. I found that of the multilayer perceptron models I implemented, a model with two hidden layers and tanh activation functions provided the highest accuracy with a testing accuracy of 89.5%. I also implemented a convolutional neural network to classify the images, and an architecture with two convolutional and two fully connected layers resulted in a testing accuracy of 89.0%. Therefore, by a small margin, my convolutional neural network was outperformed by my multilayer perceptron network.

INTRODUCTION

The task of this project is to classify image data using neural networks. This was completed mostly with various multi-layer perceptron models, and a convolutional neural network. I observed the effects of changing the number of hidden layers, activation functions, normalization, and regularization. Finally, I observed how the testing accuracy was affected by the implementation of a model with convolutional layers. I found that of the multilayer perceptron models I created, a model with two hidden layers and tanh activation functions achieved the highest testing accuracy of 89.5%. My convolutional neural network with two convolutional and two fully connected layers had a testing accuracy of 89.0%. Thus, the multilayer perceptron model slightly outperformed the convolutional neural network. Many others have implemented various neural net architectures in order to classify the fashion-MNIST dataset, with a particularly large number of researchers implementing convolutional neural networks. Bhatnagar, Ghosal and Kolkar (2017) implemented one such network and achieved a 92.54% testing accuracy with their best architecture. Vijayaraj et al. (2022) made an architecture involving convolutional layers and ADAM that achieved 94.52% testing accuracy. Interestingly, both of these researchers also observed differences with respect to the models' performance depending on the target label. All models tended to perform worse, according to F1-score, on the 'upper body' clothing articles, that is, 't-shirt/top', 'pullover', 'coat', and 'shirt'. It is interesting to note that this discrepancy was achieved with a wide variety of neural network architectures. Agarap (2018) made both a non-convolutional and a convolutional model to analyze the fashion-MNIST dataset. They implemented an architecture with 3 hidden, fully connected layers and dropout that achieved 92.23% accuracy with ReLU activation. Furthermore, their convolutional network achieved 83.24% testing accuracy.

DATASETS

The fashion-MNIST dataset that I analyzed contained black and white images of 10 different articles of clothing. I implemented several functions in order to understand individual data points, such as a function to display the given image, or see the distribution of pixel color within a given image. In order to use the dataset in my models, I first split the data into training and testing sets. The training dataset was further split into a training and validation set. Then I vectorized and normalized both sets. Vectorization converted 28 by 28 images into 784 length vectors, corresponding to the number of input units in my perceptron models.

RESULTS

A multilayer perceptron model trained on the dataset with no hidden layers, and only a softmax and input layer has a training accuracy of 85.6% and a testing accuracy of 83.8% (Figure 1). The learning rate that provided this accuracy was 0.01. IAn MLP with a single RELU layer has a training accuracy of 94.5% and a testing accuracy of 88.0% (Figure 2) with a learning rate of 0.1. An MLP with two RELU layers has a training accuracy of 91.3% and a testing accuracy of 87.9% (Figure 3), achieved with a learning rate of 0.01. Thus, the accuracy increases as the number of hidden layers increases. Thus, the accuracy improved significantly, increasing from zero to one hidden layer, but there was very little change in performance increasing from one to two hidden layers.

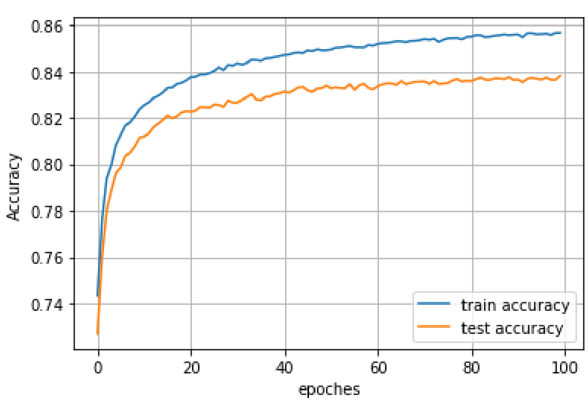


Figure 1 MLP With No Hidden Layer

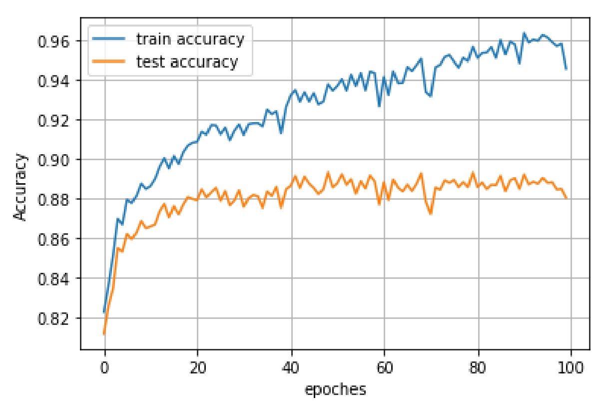


Figure 2 MLP With One Hidden Layer

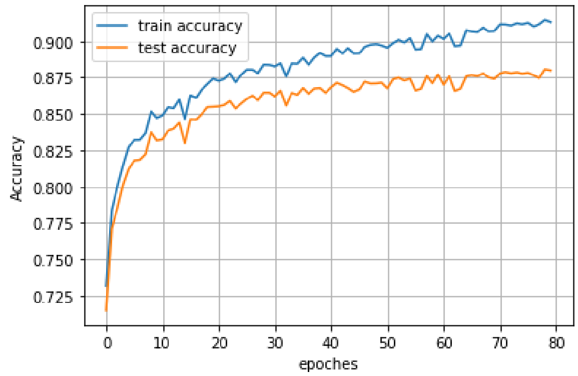


Figure 3 MLP With Two Hidden Layers

An MLP with 2 hidden layers and tanh activation functions for both layers resulted in a testing accuracy of 89.5% (Figure 4). A similar MLP except with leaky ReLU has a testing accuracy of 84.0%. Thus, the tanh activation function slightly outperforms the leaky ReLU activation function.

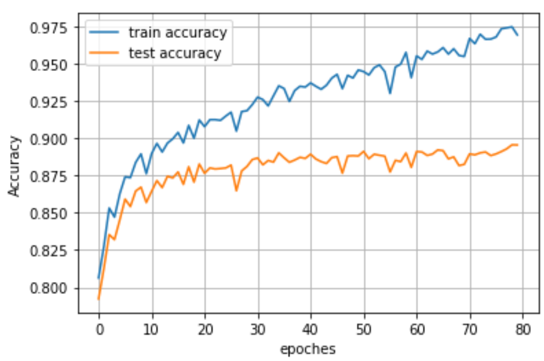


Figure 4 MLP With Two Hidden Layers (TanH, TanH Activation)

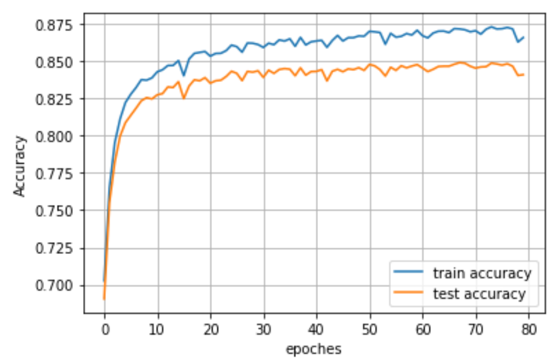


Figure 5 MLP With Two Hidden Layers (Leaky RELU, Leaky RELU Activation)

When L2 regularization is added to the network weights for a network with two hidden layers, the testing accuracy decreases as the hyperparameter lambda increases (Figure 6a), so a lambda of 0.01 results in the highest accuracy. The accuracy of a model with this lambda is 87.8% (Figure 6b).

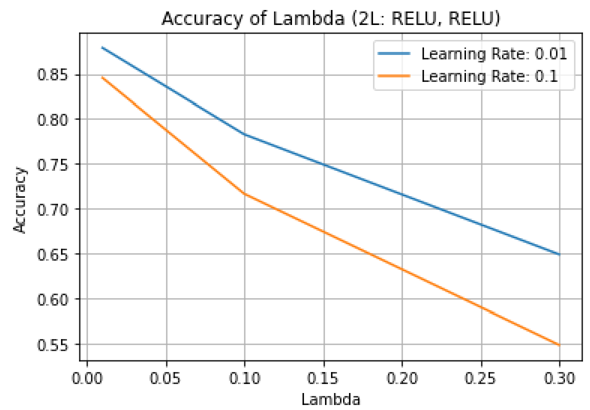
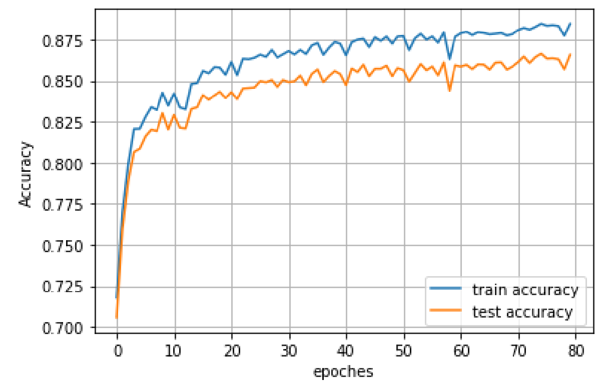
	

Figure 6a Selection of Hyperparameter Lambda	*Figure 6b* MLP With Two Hidden Layers (ReLU Activation) and L2 Regularization

Training a two-hidden layer MLP with unnormalized inputs very significantly reduces the model's performance, reducing the testing accuracy to only 10%.

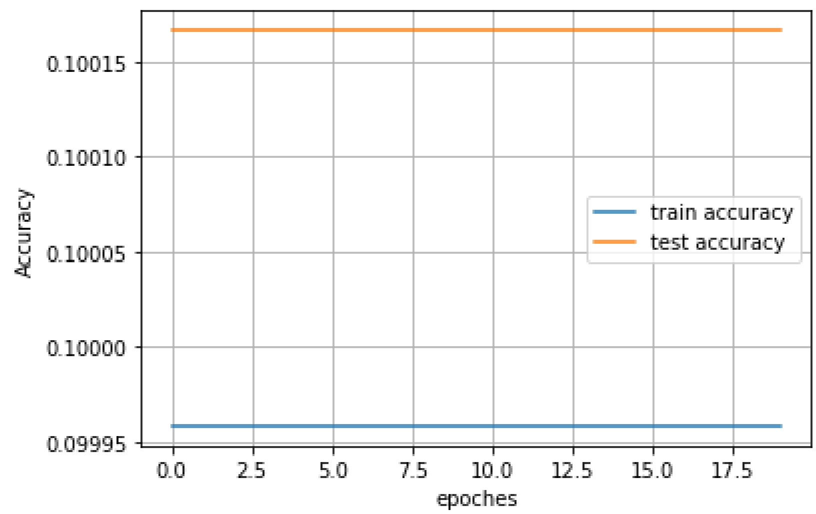


Figure 7 MLP With Two Hidden Layers (ReLU Activation) and Unnormalized Images

The convolutional neural network I implemented had two convolutional and two fully connected layers, and achieved a testing accuracy of 89.0%, which is better than most of my other models, but slightly worse than the model with two hidden layers and tanh activation functions. Furthermore, this accuracy was achieved with virtually no experimentation, changing the parameters or architecture to improve accuracy should be relatively simple.

The model that was able to achieve the highest test accuracy had two hidden layers, both with ReLU activation and L2 regularization. The learning rate and lambda were 0.1

and 0.001, respectively. This model achieved an accuracy of 87.3%. This model slightly underperformed the convolutional neural network I implemented.

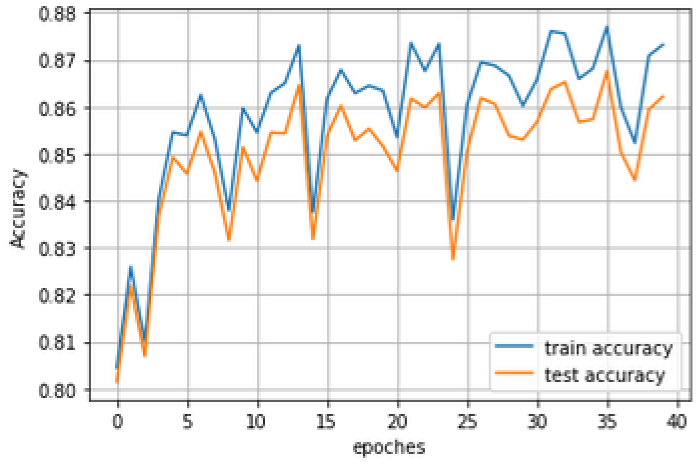


Figure 9 Highest Accuracy MLP (Two ReLU Hidden Layers)

I also observed the effect of changing the size of the hidden units and how that changed with activation function. Increasing the unit size from 64 to 256 only resulted in noticeable improvements (more than 1%) for a model with tanh activation functions, with the accuracy improving from 87.9% to 89.0% (Figure 10).

Finally, I experimented with the size of training data size. Expectedly, the model's accuracy increased as the training data size increased (Figure 11). More interestingly, this experiment highlights how important it is to train the model on a large dataset, as my model only reaches an accuracy over 80% after training on about 10000 examples.

DISCUSSION

A majority of the analysis performed on the fashion-MNIST dataset is carried out via convolutional neural networks. Furthermore, most of the efforts to improve accuracy are geared towards improving these convolutional neural network architectures. Thus, investigation of different techniques to classify the dataset might provide a more balanced understanding of the underlying trends of the dataset. Overall, my best multilayer perceptron model outperformed my convolutional network. However, I also analyzed the dataset with a wider variety of multilayer perceptron models.

APPENDIX

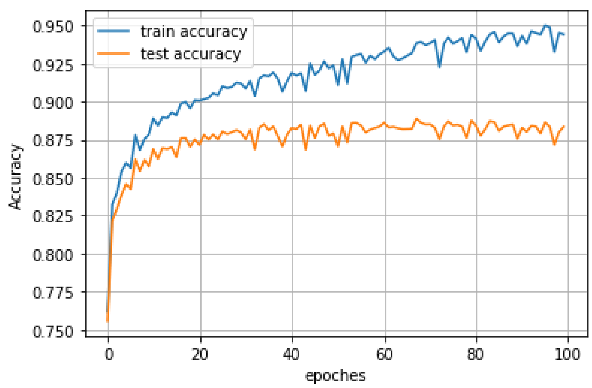
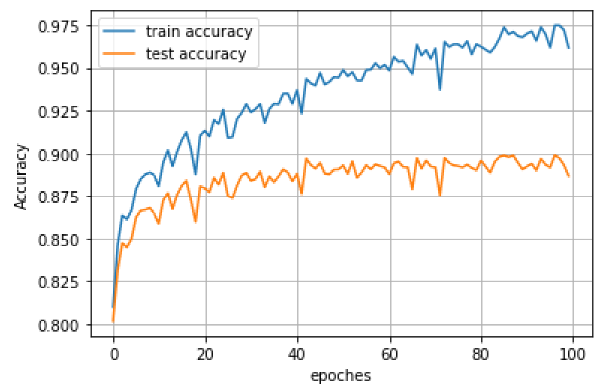
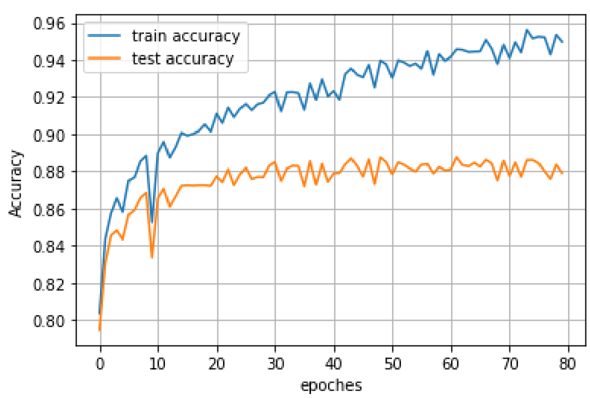
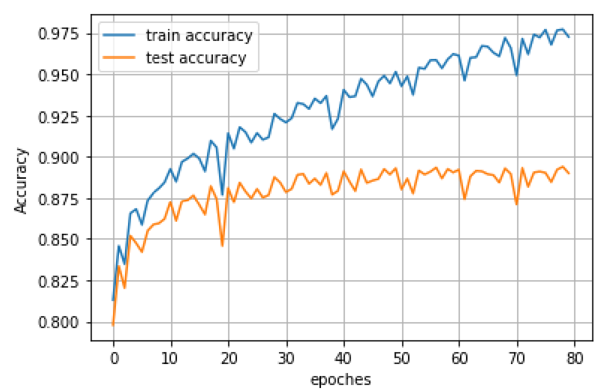
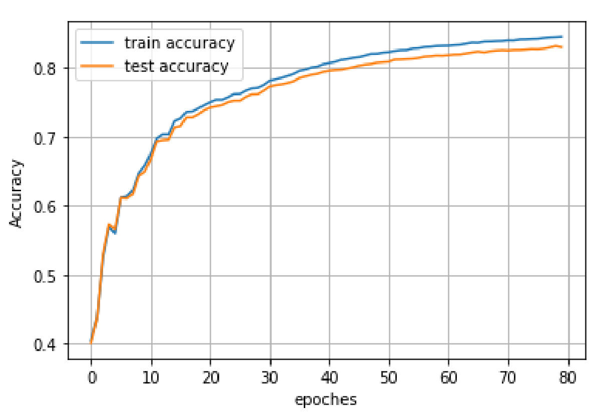
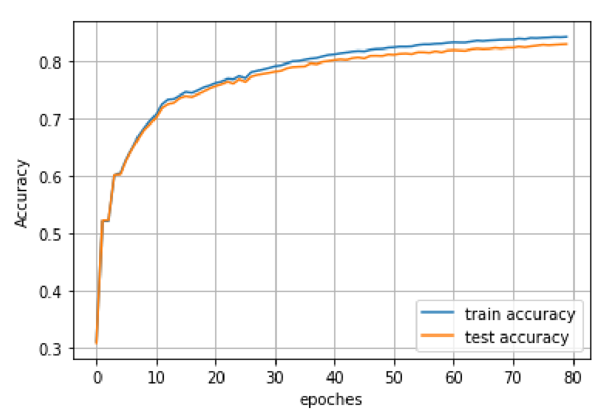
64 unit size	256 unit size
	
MLP With 1 Hidden Layer (ReLU)	MLP With 2 Hidden Layers (ReLU)
	
MLP With 2 Hidden Layer (TanH)	MLP With 2 Hidden Layer (TanH)
	
MLP With 2 Hidden Layer (Sigmoid)	MLP With 2 Hidden Layer (Sigmoid)

Figure 10 Effect of Changing Hidden Layer Unit Size

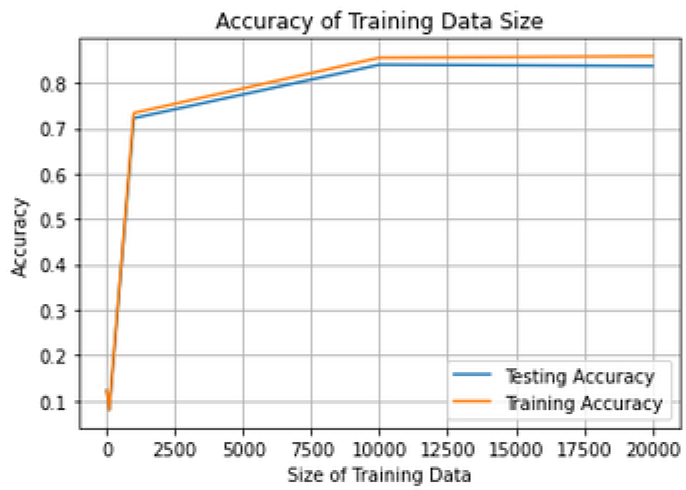


Figure 11 Effect of Changing Training Data Size

REFERENCES

A. Agarap, “Deep Learning using Rectified Linear Units (ReLU)”, arXiv:1803.08375[cs], 2018

A. Vijayaraj, P. T. Vasanth Raj, R. Jebakumar, P. Gururama Senthilvel, N. Kumar, R. Suresh Kumar, R. Dhanagopal, "Deep Learning Image Classification for Fashion Design", *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 7549397, 13 pages, 2022. https://doi.org/10.1155/2022/7549397

H. Xiao, K. Rasul, R. Vollgraf, “Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms,” arXiv:1708.07747[cs], 2017

S. Bhatnagar, D. Ghosal and M. H. Kolekar, "Classification of fashion article images using convolutional neural networks," *2017 Fourth International Conference on Image Information Processing (ICIIP)*, 2017, pp. 1-6, doi: 10.1109/ICIIP.2017.8313740.