Classification of Image Data

Yu Yun Liu; Ruoyu Deng; Pengyu Xue

April 5, 2022

Abstract

In this project, we investigated the performance of a multilayer perceptron model and a CNN for image classification. We implemented the multilayer perceptron from scratch and customized a CNN from the TensorFlow library. We explored the effect on the accuracy with different numbers of hidden layers in our MLP. We have found that generally the CNN model performs better than our MLP models and that the MLP model performs the best with the tanh function as the activation function.

1 Introduction

The goal of this project is to get familiarized with the multilayer perceptron model and its training algorithm by implementing it from scratch and to use it to do image classification on the Fashion-MNIST image dataset. We explore the different effects on accuracy on the same data with MLPs with different combinations of hidden layers, activation functions, as well as adding dropout regularization and having unnormalized training images as the input data. The number of hidden layers include 0, 1 or 2 layers, whereas the activation functions used are ReLU, tanh and leaky-ReLU. To do image classification, we apply the softmax activation function for the last layer of the MLP. Finally, we compare the performances of our MLP models with a CNN we created from the TensorFlow library with 2 convolutional and 2 fully connected layers. We have found many statements. Fewer hidden layers in MLP results in higher test accuracy with the same activation function. The model has the highest test accuracy with one hidden layer. With the same number of hidden layers, the model with the tanh function as the activation function has the best test accuracy. Moreover, the model gives the best accuracy when there is no dropout regularization . However, the model behaves poorly when trained with unnormalized images data. Lastly, the CNN model has better training and testing accuracy than our MLP models.

2 Data Sets

The dataset is an image dataset called Fashion-MNIST. It comprises 28x28 grayscale images from 10 different classes of clothing apparel ranging from T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag and Ankle boot.

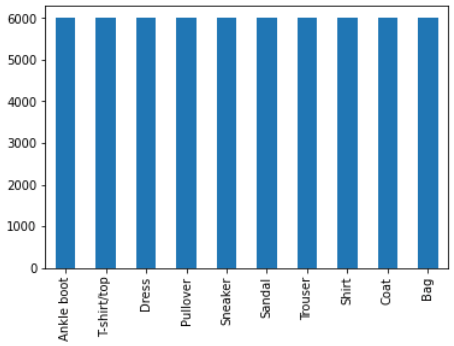

Figure 1: 10 different classes of clothing apparel ranging

The training data contains 60000 images and 6000 for each class label whereas the testing data contains a total of 10000 images. Before putting the data as input into the multilayer perceptron for training, we had to do some preprocessing on it. First, we flatten the images into 1D arrays so that they can be trained by the MLP, i.e. they go from size 28x28 to 1x784. After, we apply mean subtraction and standardization to each of the images. Mean subtraction means that we subtract the mean of the image pixels to each of the pixels of the image. This allows us to center the mean of the pixels to 0 for each image. Standardization implies dividing the image pixels by its standard deviation, which makes the pixels range between 0 and 1. Hence, we are normalizing our images so that they can be of approximately the same scale. (https://cs231n.github.io/neural-networks-2/datapre). Additionally, we have modified our labels list to be a 60000 by 10 array such that for each image, the label stores the probability of being in a certain class. For example, the first image of the dataset refers to an Ankle Boot, and the associated label is 9. Then, the first entry of the transformed labels list will become [0,0,0,0,0,0,0,0,0,1]. This transformation is necessary since the last layer's activation function of our MLP is always the softmax function. Recall that softmax produces a list of probability which sums to 1.

3 Results

To have a clear observation of how each decision affects accuracy, for each test, we have to change one parameter while keeping all other hyper parameters unchanged. In the experiments, we used gradient descent as the optimizer, and we set the learning rate to 0.0002 and the maximum iterations to 6000. First of all, we trained 3 different MLP models with respectively 0 hidden layer, 1 hidden layer and 2 hidden layers. Each model is trained with normalized training data and uses ReLU as the activation functions for all layers and each layer has 128 hidden units. In addition, the dropout ratios have been set to 1 for all three models i.e. they do not perform dropout regularization. We thus run the experiment to observe the test accuracy of each model.
The result shows that when the model contains two hidden layers, it results in relatively lower test accuracy than the models with less hidden layers. On the other hand, the other two models i.e., the one with no hidden layer and the one with one hidden layer, performed fairly close to each other and obtained similar test accuracy. However, the model with one hidden layer performed slightly better than the one with no hidden layer.

	Test Accuracy	Activate Function	Hidden Layer	Dropout Ratio	Normalized Train Data
0	0.8352	Relu	0	1	True
1	0.8334	Relu	1	1	True
2	0.7405	Relu	2	1	True

Figure 2: Comparison of Number of Hidden Layer

Second, to test which activation function performs best, we trained three models with normalized data with two hidden layers and the same dropout ratios, but different activation functions, e.g. tanh, ReLU and leaky-ReLU respectively. The result shows that the model with the tanh activation function has the highest test accuracy and the one with ReLU activation function behaves less accurately. The model trained with leaky-ReLU performs the worst among the three models.

	Test Accuracy	Activate Function	Hidden Layer	Dropout Ratio	Normalized Train Data
0	0.7137	Relu	2	1	True
1	0.7001	Leaky Relu	2	1	True
2	0.7445	Tanh	2	1	True

Figure 3: Comparison of Activation Functions

After that, we test how changing the dropout ratios can affect the test accuracy. We perform the same set-up strategy as the previous experiments. The models use ReLU as the activation function and each model has two hidden layers. They are trained with normalized image data. The result shows that under such conditions, the model obtains the best accuracy when the dropout ratio is 1.0. The model's accuracy decreases as dropout ratio decreases i.e. the model performs worse as more units are being disconnected from the neural network.

	Test Accuracy	Activate Function	Hidden Layer	Dropout Ratio	Normalized Train Data
0	0.7662	Relu	2	1.0	True
1	0.6834	Relu	2	0.9	True
2	0.6984	Relu	2	0.8	True
3	0.6391	Relu	2	0.7	True
4	0.3550	Relu	2	0.6	True

Figure 4: Comparison of Dropout Ratios

Lastly, we want to compare the test accuracy under the above same conditions when the MLP is trained with unnormalized images and normalized images. When the model is trained with unnormalized data, the test accuracy drastically decreases and the model fails to perform any meaningful prediction. The model behaves normally with normalized training data which is the originally preprocessed data.

	Test Accuracy	Activate Function	Hidden Layer	Dropout Ratio	Normalized Train Data
0	0.1000	Relu	2	1	False
1	0.7465	Relu	2	1	True

Figure 5: Comparison of Normalized Image and Unnormalized Image

In addition, we used the TensorFlow library to create a convolutional neural network (CNN) with 2 convolutional and 2 fully connected layers. The number of units in the fully connected layers is set to be 128 and ReLU is set as the activation function for all layers.
After setting up the CNN model, we trained both CNN and MLP with 5 epochs. When we trained the MLP with 5 epochs, we tried different combinations of the learning rate, maximum iterations, number of hidden layers, activation functions and dropout ratios. The combination of these hyperparameters giving the best performance is when the learning rate is 0.00035 and the maximum iterations are 5000 with one hidden layer, a dropout ratio of 0.85 and tanh as the activation function. It results in the highest test and training accuracy, and in each epoch, the model learns based on the weight matrix obtained from the previous epoch to make better predictions.
After figuring out the highest accuracy results in MLP, we compared CNN and MLP models by plotting their test accuracy and training accuracy as a function of epochs. The plot clearly demonstrates that both training accuracy and test accuracy for CNN are higher than the ones of MLP.

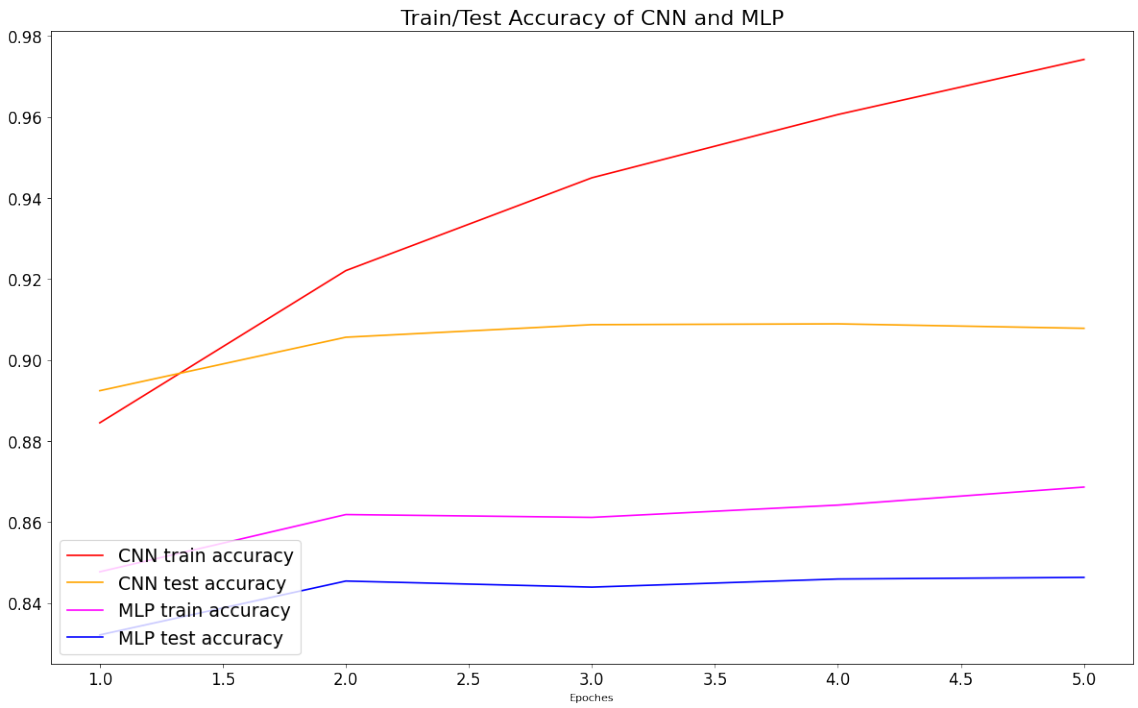

Figure 6: Comparison of CNN and MLP

4 Discussion and Conclusion

This project implements MLP and CNN models to classify images, and we are doing two parts. First, for the MLP model, we observed how each parameter will affect the accuracy of the classifications. Different numbers of hidden layers, dropout ratios, activation functions, and image normalization all affect the accuracy of the model. We found that the fewer hidden layers, the higher accuracy; Tanh as the activation function results in the best accuracy; when the dropout ratio is one, the accuracy is the highest; and the MLP performs better when the training data are normalized rather than unnormalized ones. Finally, we found that the CNN model results in higher accuracy under the same parameter conditions than the MLP model does on the Fashion-MINIST dataset. We can conclude that CNN performs better for image classification tasks.

5 Statement of Contributions

Yu Yun worked on the first task. She also contributed to writing the abstract, intro and dataset part of the report.
Ruoyu worked on implementing the models in Task 2 and the running process in Task 3 of the project.
Pengyu helped Ruoyu partially in task 3, and formatted the final version of the writeup into overleaf.