COMP 551 Mini Project 3

Lansu Dai, Zichuan Guan, and Lan Xi Zhu

Abstract. We implemented a multilayer perceptron (MLP) network for image classification with Fashion-MNIST dataset, and experimented with different network architectures. We investigated the impact of 0,1, and 2 hidden layers. Increasing the network depth resulted in performance increase, however, we observed diminishing returns with more than one hidden layer. The choice of activation function also showed impact on performance. tanH and ReLU yielded similar results with ReLU being slightly better and steadier, only to be topped by Leaky-ReLU with an appropriate γ. MLP's performance was not improved by applying dropout regularization, but the effect of the latter is visible with smaller training set size. Training and testing with unnormalized images decreases test accuracy. A CNN with two convolutional layers and two fully-connected layers is compared with our implementation of MLP on the same classification task. The test accuracy is 0.9159 and 0.8959, respectively.

1 Introduction

The goal of this project is to investigate the performance of multilayer perceptron (MLP) and convolutional neural network (CNN) with varying structures and hyperparameters by applying them to 2-D image classification task. The Fashion-MNIST dataset is used to train and test both models. Similar to the MNIST database with hand written digits, the Fashion-MNIST dataset consist of a set of 28x28 grayscale images displaying ten different classes of fashion products (e.g. dress, shirt, bag, etc.)[1]. While it is widely used as an alternative to the MNIST database thanks to their identical structures, it is particularly useful to benchmark models that specifically aim at identify cloths in the context of e-commerce/social media [2]. We obtain a test accuracy of 0.8959 with a two-hidden-layer MLP (256 units per layer) with ReLU activation and dropout regularization (p=0.7). Our CNN with one (2,2) max-pooling layer and 0.5 dropout following convolutional layers achieves a higher accuracy of 0.9159.

2 Datasets

The Fashion-MNIST dataset has 60,000 training instances and 10,000 test instances by default. Each instance is a 28x28 2-D image (with 784 pixels in total) of a fashion product belonging to one of the ten categories. Instances are evenly distributed amongst ten classes, with 6000/class in the training set and 1000/class in the test set.

Each pixel in the non-processed (raw) image instances is represented by a grayscale value between 0 and 255. We pre-process the images by subtracting the image-wise mean from each pixel and dividing each pixel by the associated standard deviation. Both raw and normalized images are used in model testing.

To augment the dataset in attempt to achieve better test performances, we modify a subset of the training instances by rotating them while keeping original dimensions (Figure 1).

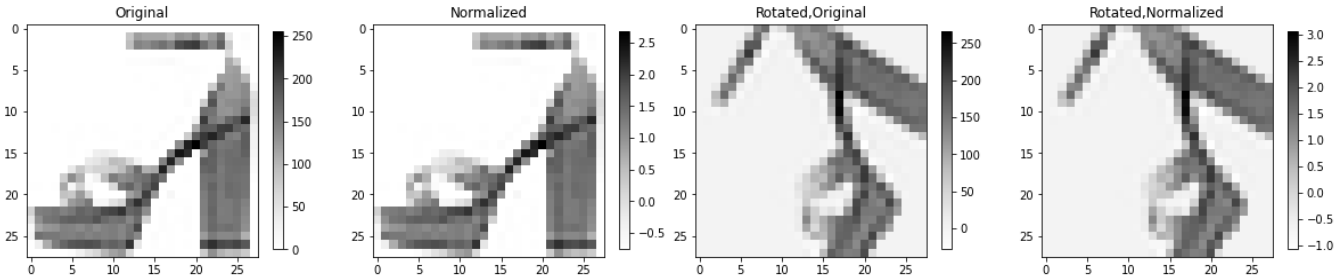

Figure 1: An example visualizing how rotation and/or normalization modifies an image instance (class 5: sandal).

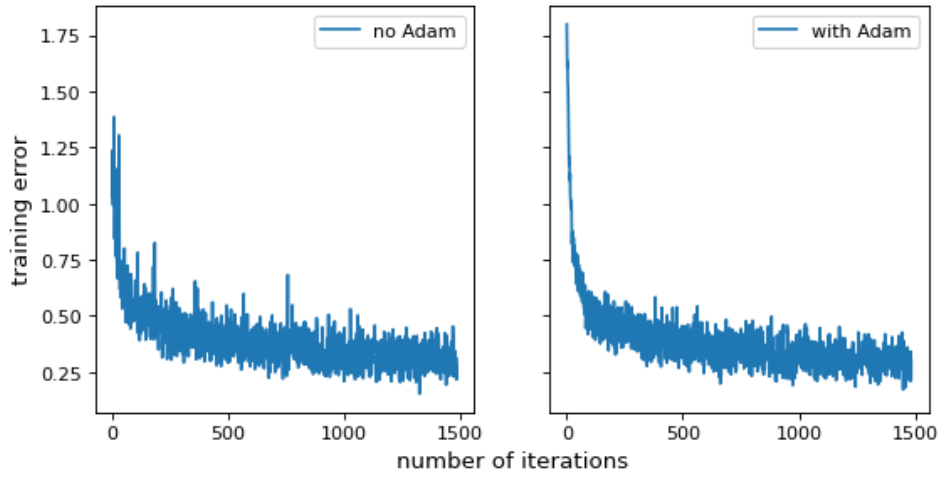

Figure 2: Training error of a MLP with one hidden layer using ReLU activation.

3 Results

3.1 Mini-batch gradient descent with Adam

We implemented mini-batch gradient descent with Adam optimizer and compared to the naive mini-batch gradient descent for training. Figure 2 shows that using Adam results in a smoother error curve that is "thinner" than with no optimizer. In general, faster convergence and higher accuracy is observed when using Adam.

We chose a learning rate of 0.001 throughout. Figure 3 shows how training costs decrease with number of iterations for different learning rate. The red curve ($\alpha = 0.001$) converges faster than those with smaller learning rate, and is smoother than the lavender curve with $\alpha = 0.01$.

We have also attempted several different values of batch size. We fix the batch size to be 200 throughout our experiments, because smaller sizes (e.g. 100, 128) have worse performance both after the same number of epochs and the same number of iterations, while larger sizes (e.g. 600) easily overfit due to the model's higher dependence on the training set.

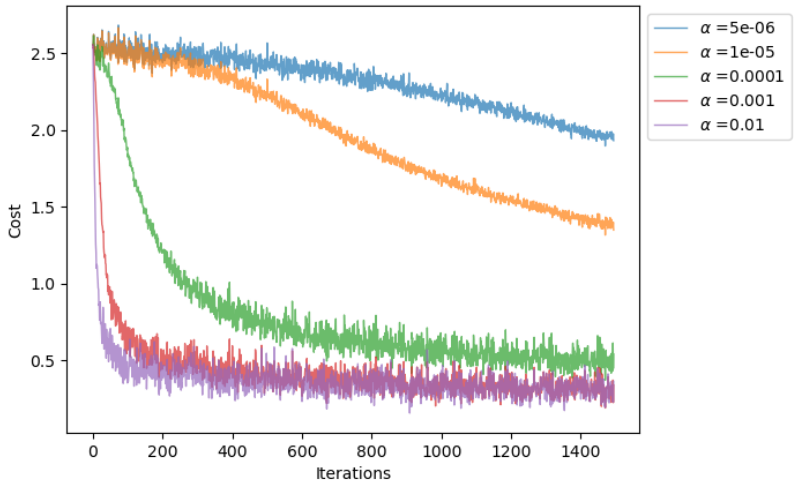

Figure 3: Training cost of MLP models with different learning rate in mini-batch gradient descent as a function of iterations (max. 5 epochs). All other model parameters are kept constant.

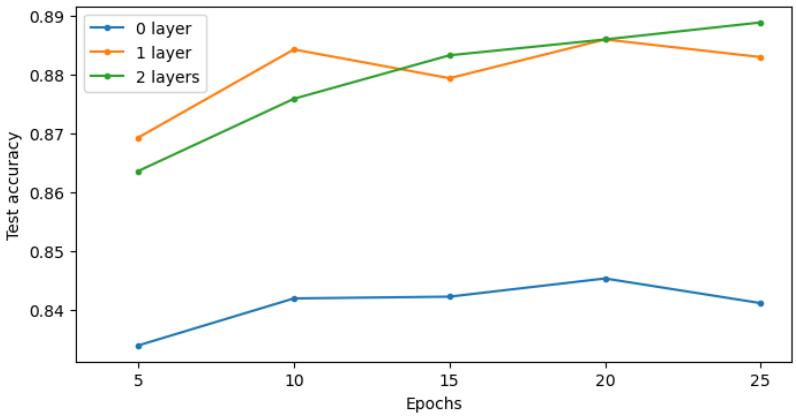

Figure 4: Test accuracy of three MLP structures as function of training epochs. The number of hidden layers in each model is indicated in the legend. Each hidden layer has 128 units.

3.2 Number of layers

We expect that adding depth to MLP will increase its performance. As observed in figure 4, the simplest model with no hidden layers performs significantly worse than the models with hidden layers. However, performance improvement for having two hidden layers instead of a single one is not significant. The model with 2 hidden layers showed consistent improvement as the number of epochs increases, while the model with only one hidden layer performed better at lower number of iterations. Having an extra layer means doubling the number of parameters, thus the model requires more iterations to train to the same accuracy. Although an extra layer adds more expressiveness, a MLP with a single hidden layer seems to be expressive enough given our test results. The observations are inline with our expectations.

3.3 Activation function

To compare the MLP's performance with different activation functions, we fix our model to take the following hyperparameters: two hidden layers with 128 units per layer, a learning rate of 0.001, a batch

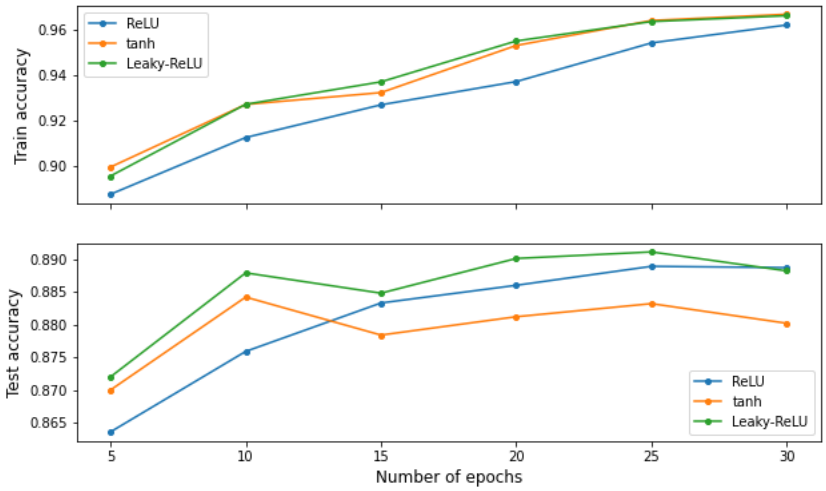

Figure 5: Train & test accuracy as a function of training epochs for a two-hidden-layer MLP with different activation functions.

size of 200 and a dropout proportion of 1 (no dropout). Under these settings, we observe that the test accuracy decreases after 25 epochs. We thus set the number of epochs to be 25 in this experiment.

With ReLU activation, the model has a training accuracy of 0.9542 and a test accuracy of 0.8889. We obtain similar performance using tanh activation, with a training accuracy of 0.9641 and a test accuracy of 0.8832. As for Leaky-ReLU activation, the accuracy largely depends on the γ parameter (the gradient at $x < 0$) which we manually set to several different values. For $\gamma = 0.01$, Leaky-ReLU performs better than ReLU activation with a training accuracy of 0.9636 and a test accuracy of 0.8911 after 25 epochs. The test accuracy decreases with larger γ.

3.4 Dropout regularization

As mentioned previously in Section 3.3, for the MLP model with 2 hidden layers each having 128 units with ReLU activation, the test accuracy starts decreasing at epochs larger than 25. Figure 6 panel (a) shows how training and test accuracy varies with increasing number of iterations for three different values of dropout proportion (p is the proportion of hidden units to be *kept* at each layer where dropout is applied). After 30 epochs, the MLP with $p = 0.7$ achieves better test accuracy than $p = 1$ (no dropout), but does not exceed the maximum accuracy obtained by the latter. Dropout regularization is therefore not improving our model performance in this case. We thus investigated how dropout regularization affects model performances with a smaller training set size of 3000 (more prone to overfitting). As shown in Figure 6 panel (b), while the training accuracy of the models with $p < 1$ stays lower than that of the $p = 1$ model, the test accuracy is comparable from one model to another. Both dropout-regularized models outperform at 40 epochs of iterations.

3.5 Normalization

With unnormalized images, our 2-hidden-layer MLP model with ReLU activations obtain a training accuracy of 0.9281 and a test accuracy of 0.8822 after 25 epochs of iterations (test accuracy decreases to 0.8817 after 30 epochs while training accuracy keeps rising). The performance, in particular the test accuracy, is

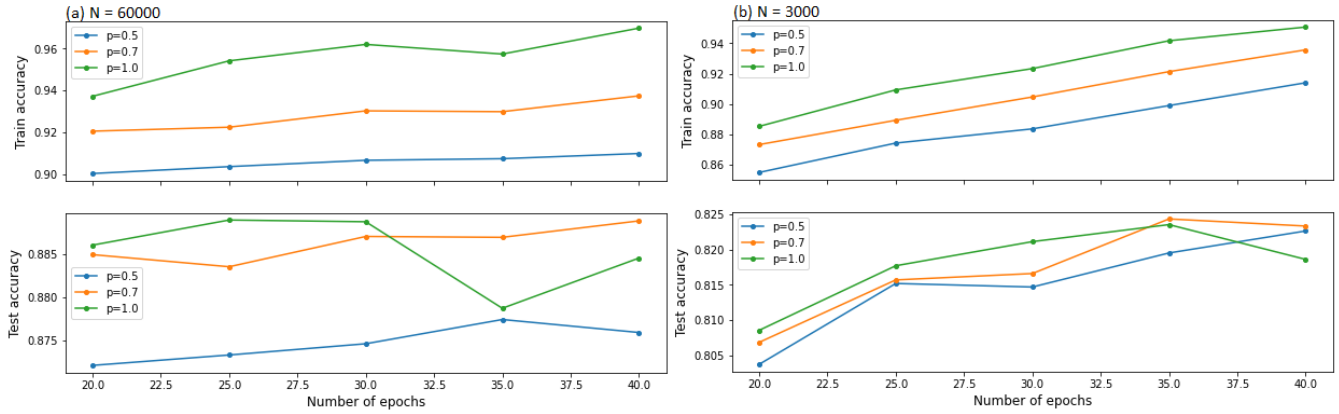

Figure 6: Train & test accuracy as a function of training epochs for a two-hidden-layer MLP with different dropout proportions. a) trained with the full 60,000 training set; b) trained with 3000 training instances.

slightly worse than but still comparable to what we previously reported using normalized images (0.8889).

3.6 CNN

We have experimented with several CNN models with 2 convolutional and 2 fully connected layers with 128 units and ReLU activations, combined with different regularization layers (e.g. max-pooling and dropout). A few of them are shown in the submitted notebook. For all models, we fixed the batch size to be 200, with `keras.optimizers.Adam()` for training. Without enough regularization layers, the CNN models easily overfit (see Figure 7 panel (a)). Using CNN generally increases the test accuracy compared to our MLP models. In particular, the model plotted in Figure 7 panel (b) gives us a test accuracy of 0.9159, while none of our MLP models was able to reach an accuracy of >0.9.

3.7 Model comparison

Starting from what we report in Section 3.3, we further improve the performance of our 2-hidden-layer MLP model with ReLU activation by increasing the width of each hidden layer to 256 and applying a dropout ratio of 0.7. We have also experimented with extending the depth of the MLP (e.g. 3 hidden layers), but are not able to improve the model performance. Our optimal MLP model's test accuracy on the Fashion-MNIST dataset (without augmentation) is 0.8959, which is slightly lower than that of the CNN model we built (0.9159).

4 Discussion and Conclusion

We observe that MLP or CNN's performance is highly dependent on their structures. With a large training set of 60,000 instances and at rather small number of iterations (epochs), regularization approaches have minimal effect on improving the performance of our MLP models with limited expressiveness (2 hidden layers, 128 units), but are quite effective while applied to CNNs. Augmenting data by appending rotated image instances to the training set decreases the test accuracy in both MLPs and CNNs, which is probably due to fact that the rotated images (Figure 1) are off-center and trimmed as compared to the originals.

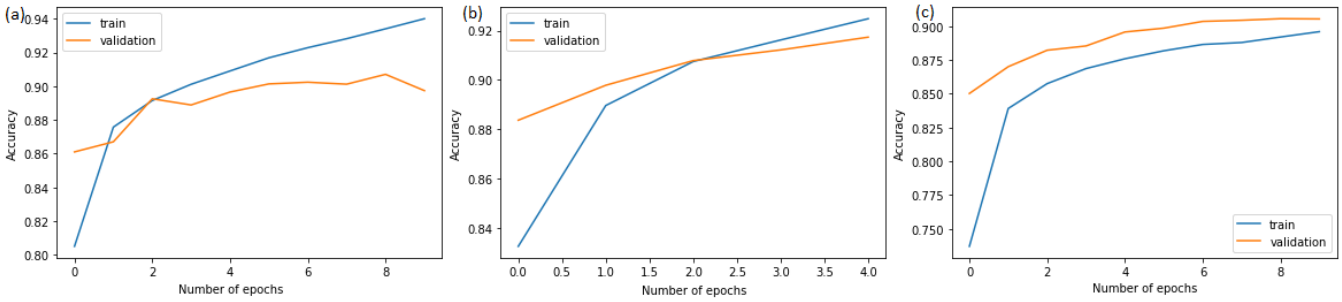

Figure 7: Training and validation (0.1 of the training set) accuracy as a function of epochs for three different CNN models. a) With one (2,2) max-pooling layer following the two convolution layers (10 epochs); b) With one (2,2) max-pooling layer and a 0.5 dropout layer following the two convolution layers (5 epochs); c) With one (2,2) max-pooling layer and a 0.5 dropout layer following the two convolution layers, and another 0.5 dropout layer following one fully connected layer(10 epochs).

CNN's performance bested our most powerful MLP architecture. We expect that this difference is due to the ability of convolution to learn locality from sequential data. Convolution is advantage in image classification since spatial information is an important feature in distinguishing objects. Even with the expressive power of MLP, we didn't observe significant over-fitting. However, CNN suffered from over-fitting with same dataset, which proves that CNN is architecturally better suited for this task.

Statement of Contributions

Everyone contributed equally to the coding component and the project write-up. We have not assigned specific part of the project to individual group member.

References

[1] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[2] K. Meshkini, J. Platos, and H. Ghassemain, "An analysis of convolutional neural network for fashion images classification (fashion-mnist)," in *International Conference on Intelligent Information Technologies for Industry*, pp. 85–95, Springer, 2019.