Classification of Fashion-MNIST Data Using MLP

COMP 551 - Assignment 3

Group 102
Sevag Baghdassarian (260980928)
Mike Zhu (261069547)
Rosa Wu (260901912)

Abstract

The assignment consisted of implementing the multilayer perceptron from scratch, applying them to Fashion-MNIST data sets, analyzing their performance, and evaluating against a Convolutional Neural Network (CNN) and ResNet algorithm. We found that a one-hidden-layer MLP with 128 hidden units trained with Adam optimizer performed the best among all the experiments. Nonetheless, the ResNet model we implemented on the Fashion-MNIST dataset did not outperform the other algorithms implemented on the same dataset.

Introduction

We implemented and fine-tuned the multilayer perceptron, CNN, and ResNet algorithms based on the Fashion-MNIST benchmark data sets prevalently used in myriads of research works [1, 2, 3, 4, 5, 6]. First and foremost, we preprocessed the datasets through vectorization and standardization. Further, we implemented MLP models with Xavier initialization [7], as well as the mini-batch gradient descent algorithm. We also experimented with different hidden layers, activation functions, and regularization methods. Moreover, we compared the performance of our implemented algorithms with vanilla CNN [8], as well as ResNet [9], a computer vision architecture with residual connections achieving state-of-the-art performance for a slew of computer vision problems. Since NumPy does not have GPU acceleration, it lags in running our code on Colab. Therefore, we replaced NumPy with CuPy, a NumPy/SciPy-compatible array library for GPU-accelerated computing with Python [10]. We also endeavoured to develop an MLP architecture that performs as well as possible. Based on our experiments, we found that a one-hidden-layer MLP with 128 hidden units trained with Adam optimizer outperformed the other models, including CNN and ResNet.

Data Sets

We utilised the Fashion-MNIST data sets loaded from Keras [11]. Specifically, the data sets include 70,000 small square 28×28 pixel grayscale images of items of 10 types of clothing, such as shoes, t-shirts, dresses, and more. Further, the training set has 60,000 images and the test set has 10,000 images. We first explored that the data set is balanced by analysing the class distributions demonstrated in Figure 1. Then we flattened and vectorised the data set to have the appropriate dimensions. Finally, we normalised the training and test set and created one-hot encoded labels.

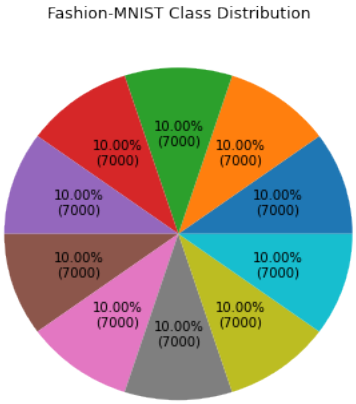

Figure 1: Class distributions of Fashion-MNIST dataset

Results

We trained several MLP, CNN, and ResNet models based on the training set of the Fashion-MNIST dataset. Further, we validated the models on the validation set. We chose the best model with the lowest validation loss in all epochs and further evaluated it on the test set. The test loss and test accuracy were demonstrated in Table 1.

We first created three different MLPs with no hidden layers using different learning rates. Since the learning rate of 0.005 yielded the highest best test accuracy (0.8435), we used this learning rate for other MLP models. We then contrasted these models with other MLPs; one having a single hidden layer (128 units/layer), and another one having two hidden layers (128 units/layer). As we expected, the models with hidden layers obtained better loss and accuracies than the model with no hidden layers. However, to our surprise, the one-layer model obtained slightly better test results than the two-layer model, possibly due to overfitting. Non-linearity in the activation functions also proved to cause better accuracy.

We also created two hidden layer MLPs with 128 hidden units, and with tanh and Leaky-ReLU activations. The best test accuracies that the tanh and Leaky-ReLU MLPs obtained are respectively 0.8728 and 0.8713. The model with tanh activation seems to be a slightly better model since it achieved a slightly better test loss and accuracy. We would expect certain activations such as Leaky-ReLU to be better than others based on the kind of data we are handling, but in this case, both models seem to be yielding weights in similar ranges.

Further, we implemented L2-regularization and tested it on two MLPs with two hidden layers and 128 hidden units; one with $\lambda = 0.1$, and the other with $\lambda = 0.5$. As apparent apparent in the figure, the MLP with $\lambda = 0.5$ has extremely poor loss and accuracy results. In fact, $\lambda = 0.1$ achieved a best test accuracy of 0.7562, whereas $\lambda = 0.5$ only achieved 0.1.

Given that the MLP with $\lambda = 0.5$ performed extremely poorly, we wanted to see if adding more layers and more hidden units, while still keeping an L2-regularization constant of 0.5, would bring any changes. We trained an MLP with three hidden layers, 256 units each, but kept $\lambda = 0.5$ and ReLU activations. Despite slight improvements in training and validation accuracy, this strategy did not make much difference on the testing loss and accuracy. The best test accuracy of the model is still very poor (0.1).

We also trained an MLP with two hidden layers, each of 128 hidden units, and ReLU activations, but trained on unnormalized images this time. The model achieved a best test accuracy of 0.8683, which is up there with some of our better models. However, a high best testing accuracy was achieved, the graphs we obtained show that there is greater variation between the training and validation losses and accuracies. Furthermore, the training and validation accuracies seem to be lower when the model is trained on unnormalized images.

We also implemented the convolutional neural networks. We compare CNN architecture with and without padding. For the first model, the first convolutional layer has kernel size 2, 1 stride; the maxpool has kernel size 2 and 2 strides. The second convolutional layer also has kernel size 2, 1 stride, with maxpool kernel size 2 and 2 strides. And we have the two fully connected layers with 128 units. For the second model, it is the same as the first one except that we added 1 padding in each of the convolutional layer. For the third model, it is the same as the first one except that we added 2 paddings in each of the convolutional layer. It turned out that model with 2

paddings performed better, with test loss 0.480, and test accuracy 0.830.

Further, we also proposed a one-hidden-layer MLP model with 128 hidden units. We trained the model with Adam optimizer with a learning rate of 0.0001, beta1 of 0.9, and beta2 of 0.999. The model achieved a test loss of 0.341 and a test accuracy of 0.880 and outperformed the other models including CNN. The accuracy based on training set and validation set was demonstrated in Figure 2.

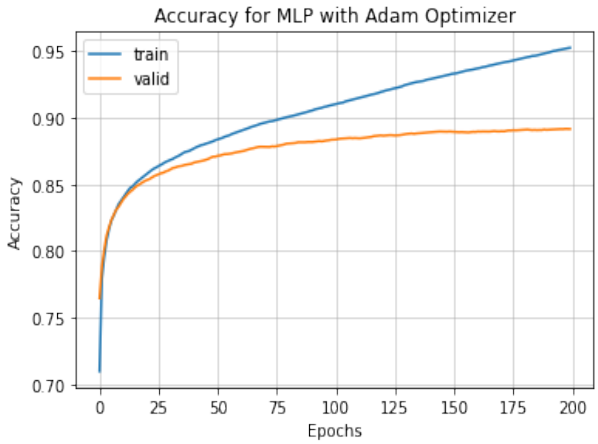

Figure 2: Accuracy for MLP with Adam Optimizer

Last but not least, we utilised ResNet architecture to evaluate the model's performance on the Fashion-MNIST dataset with state-of-the-art algorithms. Specifically, we utilised a ResNet-18 network pre-trained on more than a million images from the ImageNet database. Further, we trained it with an SGD optimizer with a learning rate of 0.0001. Our ResNet model trained on the Fashion-MNIST dataset achieves a loss of 0.477 and an accuracy of 0.834 on the test set, which did not outperform the other algorithms implemented on the same dataset. The training accuracy and validation accuracy was illustrated in Figure 3.

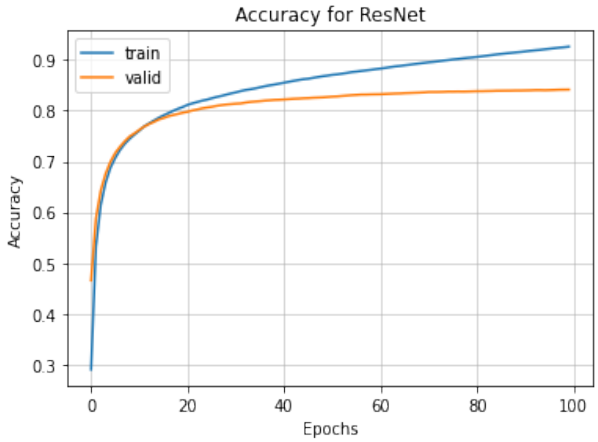

Figure 3: Accuracy for ResNet

Conclusion

In conclusion, a one-hidden-layer MLP with 128 hidden units trained with Adam optimizer performed the best among all the experiments. Further, our CNN and ResNet model did not outperform the other algorithms implemented on the Fashion-MNIST dataset.

Model	**Hidden Layers**	**Learning Rate**	**Test Loss**	**Test Accuracy**
MLP	0	0.005	0.452	0.844
MLP	0	0.1	0.560	0.810
MLP	0	0.5	1.397	0.805
MLP	1 (128 units)	0.005	0.356	0.877
MLP	2 (128 units)	0.005	0.367	0.872
MLP (Tanh)	2 (128 units)	0.005	0.358	0.873
MLP (Leaky-ReLU)	2 (128 units)	0.005	0.367	0.871
MLP (L2 λ: 0.1)	2 (128 units)	0.005	0.734	0.756
MLP (L2 λ: 0.5)	2 (128 units)	0.005	2.303	0.100
MLP (L2 λ: 0.5)	3 (256 units)	0.005	2.303	0.100
MLP (Unnormalized training set)	2 (128 units)	0.005	0.383	0.868
MLP (Adam)	1 (128 units)	0.0001	**0.341**	**0.880**
CNN (no padding)	2 Conv + 2 FC	0.0001	0.574	0.784
CNN (padding: 1)	2 Conv + 2 FC	0.0001	0.532	0.808
CNN (padding: 2)	2 Conv + 2 FC	0.0001	0.480	0.830
ResNet	18	0.0001	0.477	0.834

Table 1: Test loss and test accuracy for different models trained on Fashion-MNIST dataset.

Statement of Contributions

Sevag Baghdassarian: Task 3, Xavier Initialization (Creative Point), and report write-up
Mike Zhu: Task 1, Task 2, ResNet (Creative Point), Adam (Creative Point), CuPy GPU Accleration (Creative Point), and report write-up
Rosa Wu: Task 3, CNN, and report write-up

References

[1] H. Xiao, K. Rasul, and R. Vollgraf. (2017) Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. [Online]. Available: https://arxiv.org/abs/1708.07747

[2] L. Metz, N. Maheswaranathan, B. Cheung, and J. Sohl-Dickstein, "Meta-learning update rules for unsupervised representation learning," *The International Conference on Learning Representations 2019*, 2019.

[3] D. Lawson, G. Tucker, B. Dai, and R. Ranganath, "Energy-inspired models: Learning with sampler-induced distributions," 2019.

[4] T. Nguyen, R. Novak, L. Xiao, and J. Lee, "Dataset distillation with infinitely wide convolutional networks," *The Neural Information Processing Systems 2021*, 2021.

[5] H. Hazimeh, N. Ponomareva, P. Mol, Z. Tan, and R. Mazumder, "The tree ensemble layer: Differentiability meets conditional computation," *The International Conference on Machine Learning 2020*, 2020.

[6] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[7] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *Proceedings of Machine Learning Research*, 2010.

[8] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 1998.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[10] Cupy. [Online]. Available: https://cupy.dev/

[11] Fashion mnist dataset, an alternative to mnist. [Online]. Available: https://keras.io/api/datasets/fashion_mnist/