Analysis and Comparison of MLP and CNN Classification Models

Rafe Kruse
Herbert Wertheim College of Engineering
University of Florida
Gainesville, USA
rafekruse@ufl.edu

Bo Chen
Herbert Wertheim College of Engineering
University of Florida
Gainesville, USA
bo.chen@ufl.edu

Rafe Kruse: CNN implementation, testing, results, and CNN relevant report content.
Bo Chen: MLP implementation, testing, results, and MLP relevant report content.
Both: Abstract, Introduction, Discussion, Conclusion

Abstract**—In this project report, we implement the Multilayer Perceptron (MLP) and the convolutional neural network (CNN) for classification purposes. The two models are constructed on the Fashion-MNIST dataset and a comparison study has been conducted to investigate which classifier hits better performance on the testing set. As a result, the CNN model is able to achieve test accuracy of around 91.8% using the Fashion-MNIST dataset, On the other hand, the MLP model is able to achieve test accuracy of around 85.2% using the same dataset. The CNN demonstrated a better accuracy on the given dataset and also proved to be a better candidate on image classification application. On the other hand, CNN architectures are deep and require additional computational operations. Hence, it takes CNN longer time and higher computation power resources to train properly.**

I. INTRODUCTION

A number of studies have applied machine learning models, especially Multilayer Perceptron (MLP) and Convolutional Neural Networks (CNN), in a considerate number of tasks. These include but not limited to image classification[1][2], natural language processing[3], speech recognition[4][5], and text classification[6]. A MLP is a perceptron that teams up with additional perceptrons, stacked in several layers, to solve complex problems. CNN is a natural extension to MLP with few modifications. Mainly, the MLP algebraic dot product as a similarity function was replaced with two-dimension convolution; in addition to a pooling layer which reduces parameter dimensions making the model equi-variant to translations, distortions, and transformations. The sparse connectivity nature of CNN is also a variation to the MLP[7].

In this project report, we build a MLP model and a CNN model for image classification purposes. The objective of this work is to investigate which classifier gives the best performance on Fashion-MNIST[8] dataset. The rest of this report is organized as follows. Section 2 describes the approaches we use to construct the MLP and CNN model and pick the set of hyperparameters with best performance. Section 3 presents the testing result including the training loss with respect to each epoch as well as the testing set accuracy. The performance discussion and concluding remarks are included in section 4 and 5 respectively.

II. METHODS

2.1 Multilayer Perceptron

A multilayer perceptron (MLP) is a class of feedforward artificial neural networks. In this project, we construct a MLP with four layers of units: one input layer, two hidden layers and one output layer. Each unit represents a distinguished hyperplane of its input space. There are a few hyperparameters which affect the final performance of the MLP: learning rate, batch size, and number of units in each layer. The rest of this section explains how we tune these hyperparameters to achieve the best performance on the testing set.

Note that at this point we are not setting the number of training epochs as a hyperparameter. This is because it requires multiple models to be trained and discarded which is computationally inefficient and time-consuming. Alternatively, during training, the model is evaluated on a fixed cross validation set after each epoch. Once the loss on the validation set degrades for five times, we assure that the model starts memorizing the training data and occurs overfitting. Then the training process is stopped.

For faster convergence purposes, the optimizer of MLP is set to ADAM[10] whose step size is adaptive and is bigger in the few epochs and keeps decreasing when epoch getting larger. The loss function is chosen to cross entropy. The activation functions of the hidden layer are set to rectified linear activation functions (ReLU). SoftMax function is used for units of output layer.

Before sending the data through MLP, we conduct a preprocessing to rescale these data. Note that instead of implementing a widely used min-max scaler which maps the data to range [0,1], we rescale these data to [-1,1]. This is because image backgrounds are black so most of coefficients are equal to 0 when they're represented using range [0,1]. By transforming to [-1,1], the number of elements equal to 0 in the input samples is dramatically reduced, which helps in training the models.

In order to find the best set of hyperparameters, we use grid search methods to build a model for each possible combination of all of the hyperparameter values provided. By evaluating each model, we could select the architecture which produces the best results. We define two learning rate values 0.01 and 0.001, two batch size values 128 and 1024, two possible units number of the first layer 128 and 256, and two units number of the second layer 256 and 512. We build a model for each combination of these hyperparameters candidates. Each model would be fit to the training data and evaluate on the cross validation set. In addition, for each model we use 4 random sets of initial conditions and choose the best of them to compare with other candidates. The following table shows the validation loss of each model. Note that lr denotes learning rate, bs denotes batch size, nf denotes the number of neurons in the first hidden layer and ns denotes the number of units in the second hidden layer.

Combinations	The Best Cross Validation Loss
lr=0.001, bs=1024, nf=256, ns=512	0.39
lr =0.001, bs=1024, nf=256, ns =256	0.3985
lr =0.001, bs=1024, nf =128, ns =512	0.39
lr =0.001, bs =1024, nf =128, ns =256	0.401
lr =0.001, bs =128, nf =256, ns =512	0.3946
lr =0.001, bs =128, nf =256, ns=256	0.3977
lr =0.001, bs =128, nf =128, ns =512	0.4174
lr =0.001, bs =128, nf =128, ns =256	0.3944
lr =0.005, bs =1024, nf =256, ns =512	0.414
lr =0.005, bs =1024, nf =256, ns =256	0.4162
lr =0.005, bs =1024, nf =128, ns =512	0.4286
lr =0.005, bs =1024, nf =128, ns =256	0.4285
lr =0.005, bs =128, nf =256, ns =512	0.4526
lr =0.005, bs =128, nf =256, ns =256	0.4484
lr =0.005, bs =128, nf =128, ns =512	0.4543
lr =0.005, bs =128, nf =128, ns =256	0.4529

Table 1. MLP hyperparameter grid search results

As a result, we choose to set the learning rate to 0.001, batch size equals to 1024, the number of neurons in the first layer to 128 and the number of units in the second layer to 512. And the final MLP model is trained by all training data without cross validation split.

2.2 Convolutional Neural Network

A Convolutional Neural Network(CNN) relies on layers of convolutional and feed-forward perceptrons for the purpose of function approximation, in the case of our project the classification of articles of clothing. While applying a CNN for this project both the architecture and the hyperparameters: learning rate, batch size, and epochs all have to be tuned to maximize accuracy.

While the hyperparameter values were maximized through the comparison of a spread of values, the architecture of a CNN can be far more varied. Given the different types of layers, number of layers, and layer parameters we built the architecture based on common practice and limitations on our project. The first four layers of the network consists of two convolution layers and two max pool layers. In CNNs often there is a trade off of more convolutional layers being able to better generalize at the cost of computation time. We chose two layers because further layers show greatly diminishing returns and two allows for testing given our available hardware. The max pooling operations have been shown to maintain network accuracy while greatly reducing the complexity of the model [9]. The final three layers within the network are fully connected layers. The first serves the purpose of flattening the max pool output, the second as a classification layer, and the final as the 10 classification confidence values.

Within CNN the activation functions for all but the last layer were RELU with the last layer being a log softmax. Additionally the pytorch max pool 2D function was chosen for pooling layers and dropout was applied following the convolutional layers to reduce model complexity and reduce training time. Similarly to the MLP

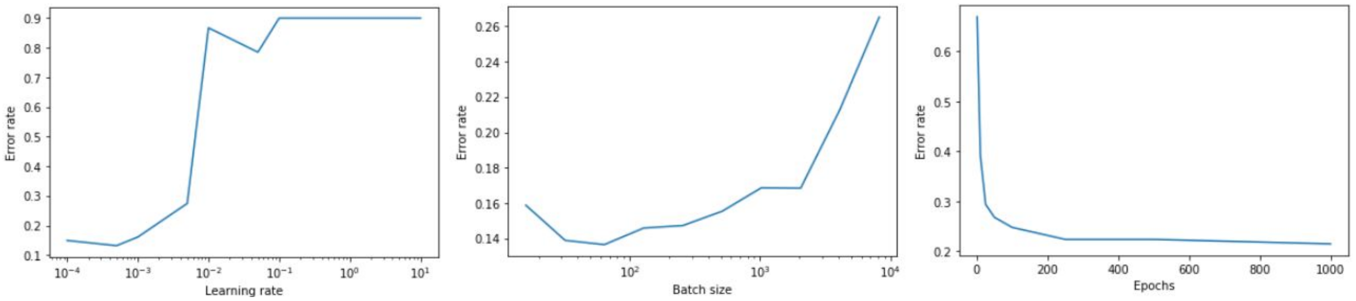


Figure 1. Hyperparameter value comparison

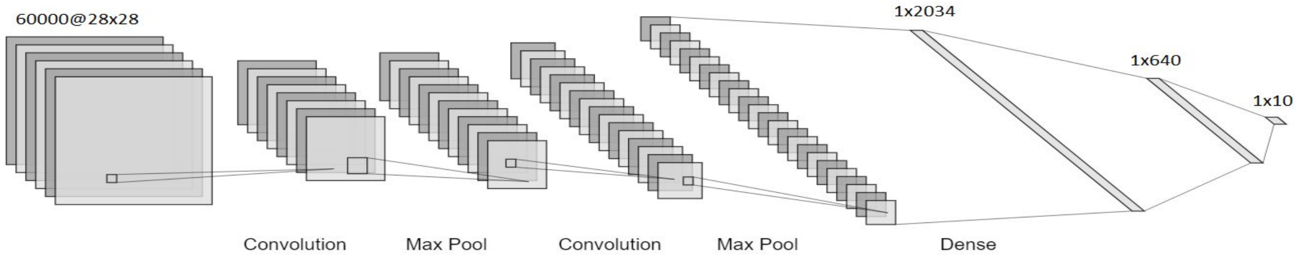


Figure 2. CNN architecture

ADAM was chosen as the optimizer and cross entropy loss as the model loss function.

As previously mentioned the ideal learning rate, batch size, and number of epochs were chosen from a spread of values based upon resulting accuracy. Figure 1 demonstrates that the ideal learning rate, batch size, and epochs is 0.0005, 64, and 250 respectively. Clarifying the number of epochs, after 250 epochs there was no measurable improvement in accuracy thus 250 epochs was chosen to minimize the required time to train

III. RESULTS

3.1 Multilayer Perceptron

The training and testing experiment is conducted on a laptop computer with Intel(R) Core(TM) i5-7300HQ CPU @ 2.50GHz and 8GM of RAM. The following plots show the training accuracy and the training loss of MLP on image classification using Fashion-MNIST[8].

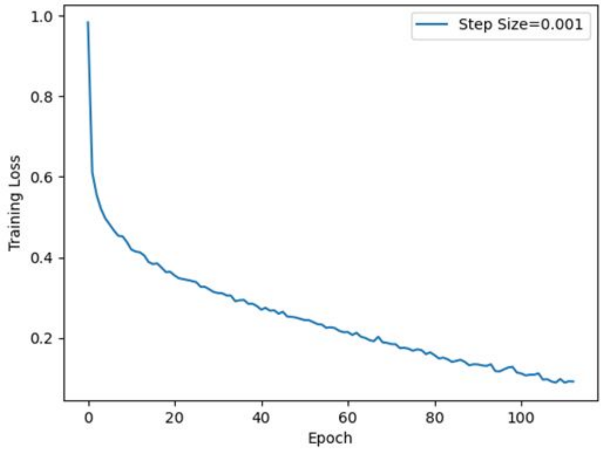


Figure 2: Training loss of MLP on image classification using Fashion-MNIST.

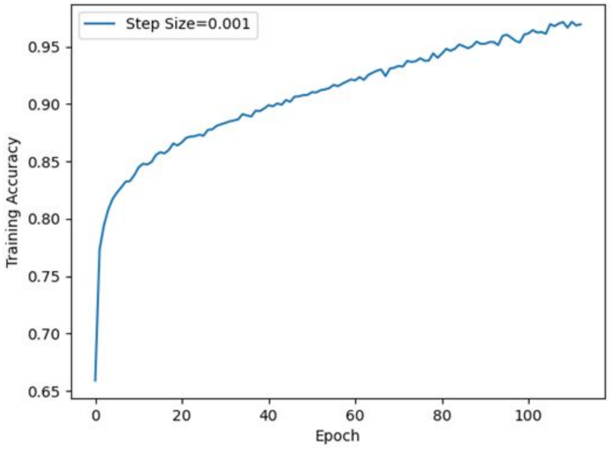
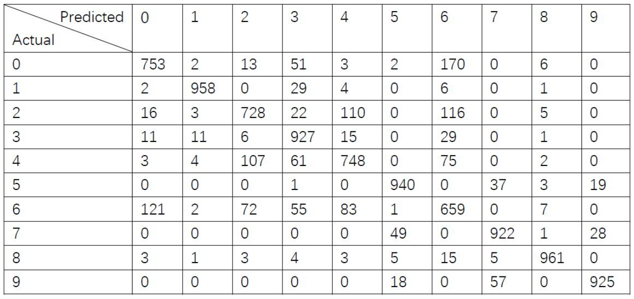


Figure 3. Training accuracy of MLP on image classification using Fashion-MNIST.

After finishing training, the model is tested on the testing set which has 10,000 data cases. The testing accuracy could achieve 85.21%. The confusion matrix for the testing data set is shown below:



Predicted Actual	0	1	2	3	4	5	6	7	8	9
0	753	2	13	51	3	2	170	0	6	0
1	2	958	0	29	4	0	6	0	1	0
2	16	3	728	22	110	0	116	0	5	0
3	11	11	6	927	15	0	29	0	1	0
4	3	4	107	61	748	0	75	0	2	0
5	0	0	0	1	0	940	0	37	3	19
6	121	2	72	55	83	1	659	0	7	0
7	0	0	0	0	0	49	0	922	1	28
8	3	1	3	4	3	5	15	5	961	0
9	0	0	0	0	0	18	0	57	0	925

Table 2. Confusion matrix of MLP on image classification using Fashion-MNIST.

3.2 Convolutional Neural Network

The training and testing of the CNN was done using the CUDA cores of a dedicated NVIDIA GeForce 1080 Ti graphics card and 16GM of RAM. Figure 4 shows the total loss on the training dataset throughout the 250 epochs of training. The model was trained on the 60000 labeled training samples of the Fashion-MNIST dataset.

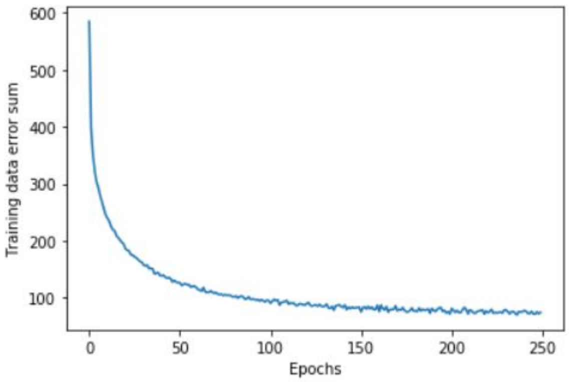
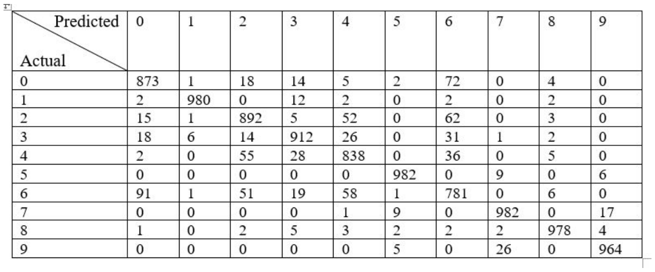


Figure 4. Training data error during each epoch of training

At the conclusion of training the model was then applied to the 10000 test samples within the same Fashion-MNIST dataset. The CNN model was able to achieve an accuracy of 91.8% on the test set. Table 3 displays the resulting confusion matrix for the classification of the test samples.



Predicted Actual	0	1	2	3	4	5	6	7	8	9
0	873	1	18	14	5	2	72	0	4	0
1	2	980	0	12	2	0	2	0	2	0
2	15	1	892	5	52	0	62	0	3	0
3	18	6	14	912	26	0	31	1	2	0
4	2	0	55	28	838	0	36	0	5	0
5	0	0	0	0	0	982	0	9	0	6
6	91	1	51	19	58	1	781	0	6	0
7	0	0	0	0	1	9	0	982	0	17
8	1	0	2	5	3	2	2	2	978	4
9	0	0	0	0	0	5	0	26	0	964

Table 3. CNN confusion matrix for classification of the Fashion-MNIST dataset.

IV. DISCUSSION

A. *Findings,comparison, Meaning,*

Our results have shown that our CNN model far outperformed the MLP model. However, this came with the caveat of training time. Our MLP was far quicker to train even using vastly inferior processing power. This variance in training time can be attributed to the large difference in size between the two models. Our MLP only consisted of 640 perceptrons while the CNN had in excess of 5000 nodes. Additionally convolutional operations take more computation to perform even in a sparsely connected network.

The resulting greater accuracy by our CNN was to be expected as CNNs have grown to be the prevalent model for the task of image processing and classification. CNNs because of their convolutional filters have the distinct benefit of being able to make use of structural information within images. CNNs are able to detect spatial patterns more efficiently than MLPs. This is because convolution inherently relates groups of inputs (pixels in this case) which is important for image tasks where neighbor pixels relate to each other when trying to classify a sample.

V. CONCLUSION

The Fashion-MNIST is considered a far more difficult problem then both the digit and character MNIST dataset which have been classified at rates exceeding 99%. Traditionally MLP or CNN implementations are able to get >95% accuracy on simpler datasets, however, that is not the case with more complicated classification problems. While we were able to achieve an accuracy of 85.2% with an MLP and 91.8% with a CNN implementation we still were short of current state-of-the-art techniques.

Within this project we were limited by our compute abilities and time. It is likely we could achieve novel accuracy results by increasing the size of the CNN model as well. With a greater size network and more time for computation a resulting model would have a greater ability to better generalize. Additionally, further data preprocessing could be applied specifically for the Fashion-MNIST dataset. Processing the dataset for the model to be able better extract relevant features would allow the model to better differentiate classes, improving accuracy.

VI. REFERENCES

[1] Y. V. Venkatesh, S. Kumar Raja, On the classification of multispectral satellite images using the multilayer perceptron, Pattern Recognition 36 (9) (2003) 2161-2175, https://doi.org/10.1016/S0031-3203(03)00013-X

[2] D. Han, Q. Liu, W. Fan, A new image classification method using CNN transfer learning and web data augmentation, Expert Systems with Applications, 95 (1) (2018) 43-56, https://doi.org/10.1016/j.eswa.2017.11.028

[3] W. Yin, K. Kann, M. Yu, H. Schütze, Comparative Study of CNN and RNN for Natural Language Processing, arXiv:1702.01923 [cs.CL]

[4] J. Park; F. Diehl; M.J.F. Gales; M. Tomalin; P.C. Woodland, Training and adapting MLP features for Arabic speech recognition, 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, 2009.

[5] T. Hori, S. Watanabe, Y. Zhang, W. Chan, Advances in Joint CTC-Attention based End-to-End Speech Recognition with a Deep CNN Encoder and RNN-LM, arXiv:1706.02737 [cs.CL]

[6] H. Peng, J. Li, Y. He, Y. Liu, M. Bao, L. Wang, Y. Song, and Q. Yang. 2018. Large-Scale Hierarchical Text Classification with Recursively Regularized Deep Graph-CNN. In Proceedings of the 2018 World Wide Web Conference (WWW '18). International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 1063–1072. DOI:https://doi.org/10.1145/3178876.3186005

[7] A. Botalb, M. Moinuddin, U. M. Al-Saggaf and S. S. A. Ali, "Contrasting Convolutional Neural Network (CNN) with Multi-Layer Perceptron (MLP) for Big Data Analysis," 2018 International Conference on Intelligent and Advanced System (ICIAS), Kuala Lumpur, 2018, pp. 1-5, doi: 10.1109/ICIAS.2018.8540626.

[8] Xiao, Han, Kashif Rasul, and Roland Vollgraf. "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms." arXiv preprint arXiv:1708.07747 (2017).

[9] Scherer, Dominik & Müller, Andreas & Behnke, Sven. (2010). Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition. 92-101. 10.1007/978-3-642-15825-4_10.

[10] Kingma, Diederik P. and Jimmy Ba. “Adam: A Method for Stochastic Optimization.” *CoRR* abs/1412.6980 (2015): n. pag.