Character and Object Recognition using MLP and CNN

Original Work by Michael Shostrand

Abstract

This paper explores the feasibility of applying machine learning models to a given marketing problem. Using security camera footage, the client wants to suggest clothing items to customers. Convolutional Neural Networks and Multi-Layer Perceptrons are tested in this experiment. Through these experiments, it is clear that machine learning is well suited for this task

Introduction

To determine whether the use of machine learning is feasible for this task, several datasets were used. MNIST is a set of images of number. EMNIST is a larger dataset of images of letters. Fashion MNIST is a set of images of clothing. If we can create models that understand these datasets, the real world application is possible. Machine learning is a powerful tool. It’s a framework that allows machines find relationships in data that wouldn’t be possible otherwise. For this application specifically, it can allow a computer to gain knowledge from an image. This is very intuitive to a human, but it’s not something we can easily explain or write as a conventional algorithm. There are limitations to these various MNIST datasets. The main one being that the images are very small compared to real world image data. These are either 28x28 or 32x32. MLPs had very good performance in this study. They may be infeasible in a real world application due to the input size.

Methods

Data Preparation

We were provided binary files containing the fashion MNIST and eMNIST datasets. Those were read and flattened such that they can be passed into the network. Another key step is normalizing the input data. This ensures that the inputs are on the same scale which increases the chance that the model arrive at a usable solution. The data is split into three categories, train, test, and validation.

Multi-Layer Perceptron
MLPs were created and trained on both the EMNIST and fashion MNIST dataset. Cross-entropy loss was the loss function. In both cases, two hidden layers were used. They were trained until the validation check failed six times in a row.

EMNIST Best Net

Hidden Layers	Hidden 1 size	Hidden 2 size	Data division (train/val/test)	Epoch	Avg ROC AUC
2	64	24	70/15/15	638 (validation stop	0.9835

Table 1

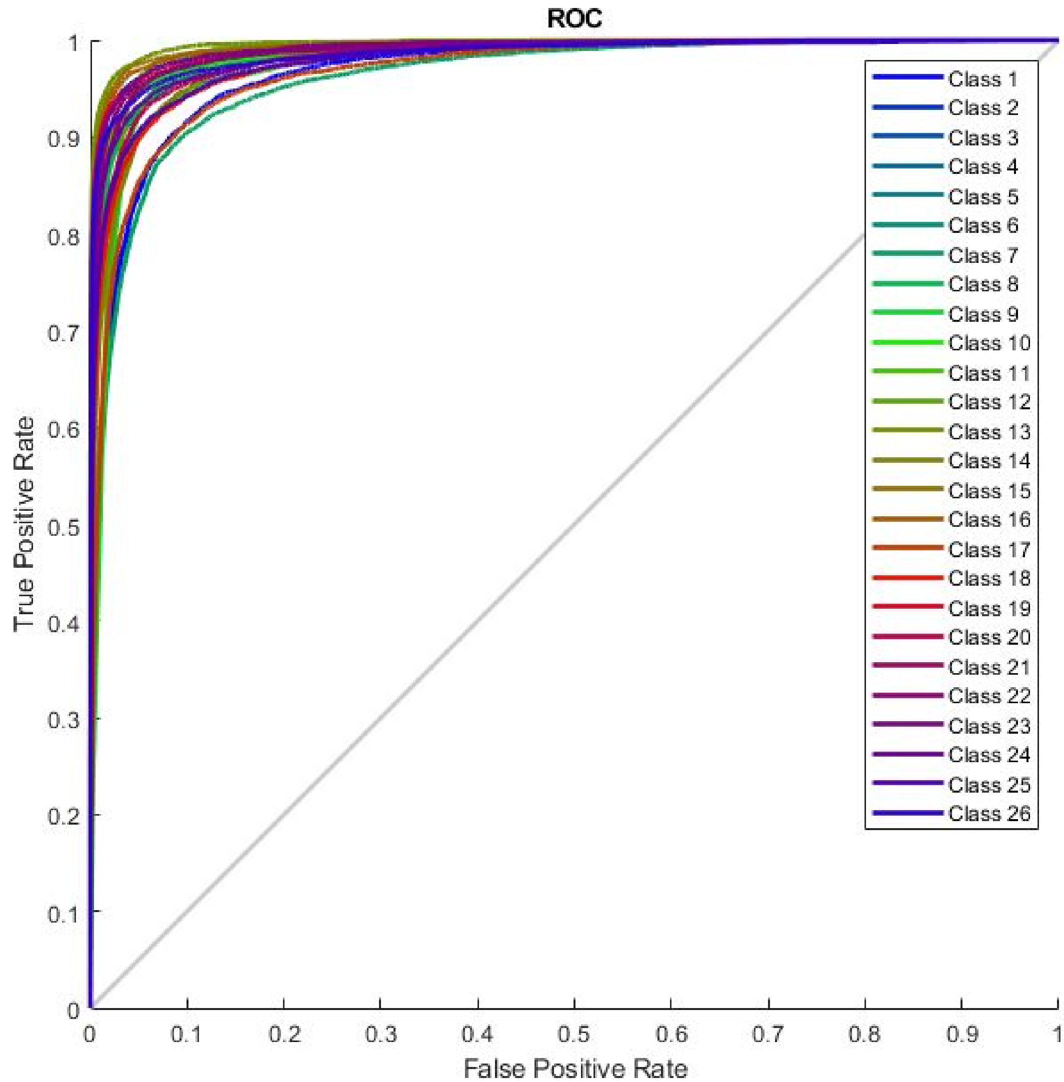

Figure 1

Fashion MNIST Best Net

Hidden Layers	Hidden 1 size	Hidden 2 size	Data division (train/val/test)	Epoch	Avg ROC AUC
2	64	32	60/25/15	100	0.9859

Table 2

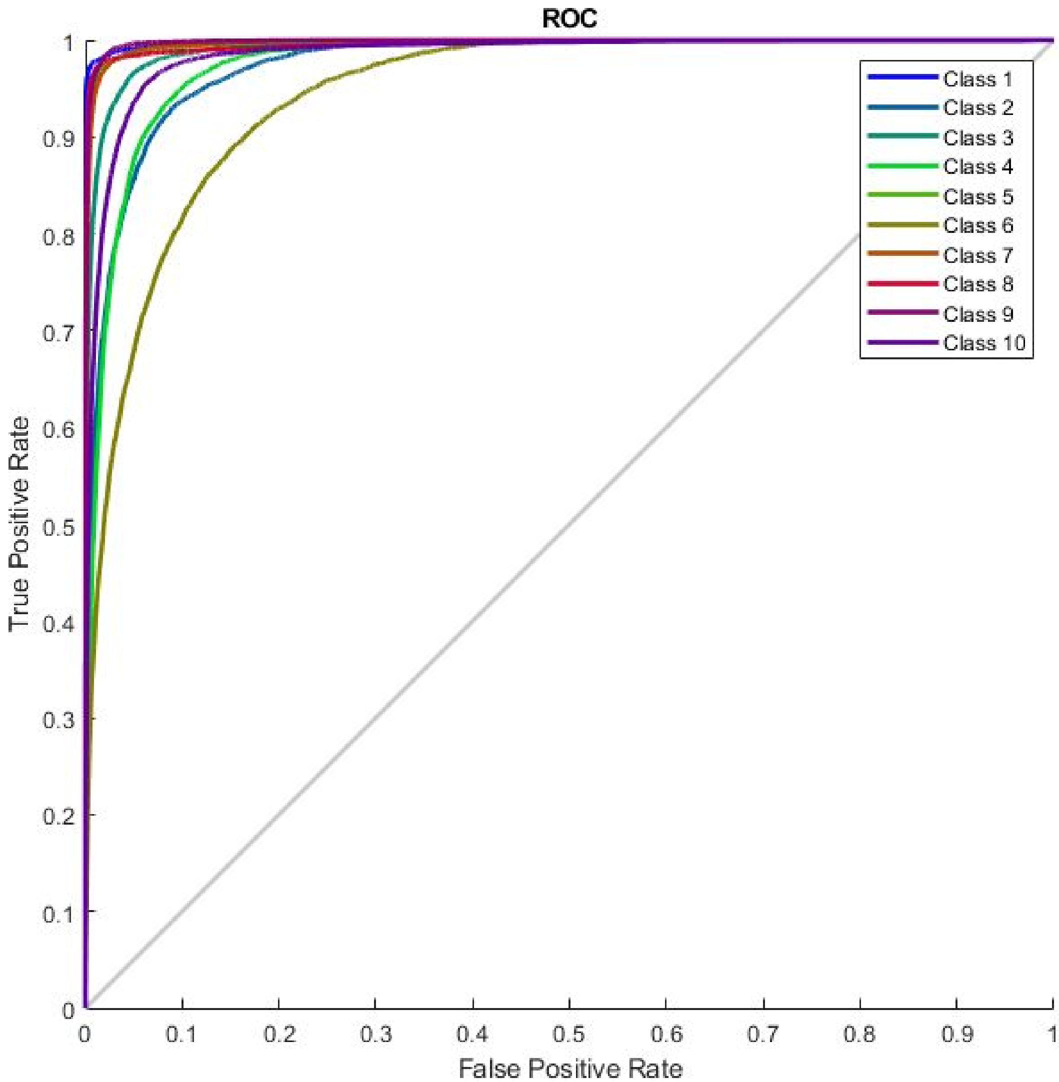

Figure 2

CNN

The CNNs in this experiment consist of three blocks. Each one has a convolutional layer, normalization, non-linearity (ReLU), and a pooling layer.

EMNIST Best Net

Block 1 filter size	Block 1 num filters	Block 2 filter size	Block 2 num filters	Block 3 filter size	Block 3 num filters	Test Accuracy	Train Accuracy
7	16	5	24	3	48	0.9163	0.9149

Table 3

Fashion MNIST Best Net

Block 1 filter size	Block 1 num filters	Block 2 filter size	Block 2 num filters	Block 3 filter size	Block 3 num filters	Test Accuracy	Train Accuracy
7	12	5	20	3	48	0.8827	89.22

Table 4

Results and Discussion

The MLPs had very good performance on both datasets. The average of the ROC AUC values for each class were 0.9835 and 0.9859 for EMNIST and fashion MNIST respectively. The input size when working with MNIST images is very manageable. For the 28x28 grayscale images, the input layer size is 784. I found that increasing the hidden layer sizes increased performance to a certain point. After which it just led to more overfitting. One problem is that the MLPs may not be able to scale with larger, real world images. A most realistic image would be 480x480 with 3 color channels. The input layer size for that network would be 691200. CNNs are usually favored instead of MLPs for image processing[1].

The performance of the CNNs lagged behind the MLPs in these experiments. Modifying the filter size yielded small returns in accuracy. The idea behind that was that increasing the filter size allows the network look a larger curves. Also, increasing the number of filters at each layer was attempted. Similarly, this did result in a large performance increase. From these results, I would suggest that MLPs were better suited for this task. However when comparing to others' results, this is most likely not the case [1].

Transfer learning was also used with the CNNs. This is where a pre-trained network is used on a new dataset. With CNNS, typically the convolution blocks of the pre-trained network are retained. The fully connected head is swapped out with one compatible with the target dataset. This did result in faster training. However, the accuracy was not as good as the model trained on the target dataset from the beginning.

Conclusion

Machine learning is perfect for this task. All the models performed well. More tuning is required for the real world application, but it is feasible.

References

[1] Uniqtech,
https://medium.com/data-science-bootcamp/multilayer-perceptron-mlp-vs-convolutional-neural-network-in-deep-learning-c890f487a8f1
[2] I. Goodfellow, et al. Deep Learning. Cambridge: MIT Press, 2016.