A Comparative Study in Image Classification through Multilayer Perceptron and Convolutional Neural Networks

Shreenika Aldur Krishnegowda

Shreenika.Aldur-Krishnegowda@city.ac.uk

Abstract

This paper aims to present a crucial evaluation of two algorithm models performed in a supervised learning approach on image classification tasks on the Fashion MNIST dataset. The two algorithms used in the study are Feedforward Multilayer Perceptron (MLP) and Convolutional Neural Networks (CNN). Multiple models are tried and tested with varying hyperparameters in grid search where the CNN architecture includes two convolutional layers and two fully connected layers while the MLP consists of three hidden layers with ReLU activation and dropout. Test results are compared on the Confusion Matrix and classification report. Test Results show that CNN achieves higher accuracy with both training and test datasets compared to MLP. Hence, for such a classification problem CNN is preferable to MLP.

1. Introduction

The MNIST dataset was first introduced by LeCun et al.[1998] and has become one of the most widely used datasets in deep learning even surpassing the CIFAR-10 [Krizhevsky and Hinton, 2009] and ImageNet [1]. The fashion MNIST dataset introduced by Xiao et al. In 2017 is based on the MNIST dataset emerged as a benchmark in the field of Computer Vision and deep learning. It offers a diverse range of fashion items in a grayscale format offering great flexibility and efficiency in developing models for classifying clothing and accessories [1].

The purpose of this study is to compare and evaluate the performance of two models designed on Convolutional Neural Networks (CNNs) and Multilayer Perceptron (MLP) on the Fashion MNIST dataset. The Study investigates the respective strengths and weaknesses of both these models in handling image classification tasks. It aims to evaluate the accuracy of different configurations of these two models and analyse the suitability of CNNs and MLPs for image classification thus contributing towards a deep learning approach in Computer Vision.

1.1. Multilayer Perceptron (MLP)

Multilayer Perceptron is a type of Artificial Neural network architecture is adopted for various machine learning purposes including Image Classification. MLP consists of fully connected layers where each neuron in a layer receives input from all the neurons in the previous layer and sends the output to all the neurons in the next layer. These fully connected layers also known as Dense Layers are the building blocks of MLP. By applying Rectified Linear Unit (ReLU) to the output of each neuron it becomes capable of modelling complex and non-linear relationships present in the data [2]. MLPs are relatively easy to implement compared to more complex models like CNNs, while still offering competitive performance, making them suitable for to be used as more base-line models to compare with complex architectures like CNNs [3].

1.2. Convolutional Neural Networks (CNN)

Convolutional Neural Networks belong to a class of deep learning architectures specializing in processing grid-like data such as Images. They leverage specialised layers like convolutional layers which we have used in this study, to effectively and efficiently extract spatial hierarchies of features from the images [3]. Learning filters under its convolutional layers capture the patterns at different spatial locations detecting features like edges, textures, and shapes. CNNs down sample the feature maps which reduces the spatial dimension and preserves important information. This feature extraction process allows the models to learn abstract and complex images in the network [2]. CNNs

are tailored for tasks like Image classification which has helped in revolutionizing the Computer Vision field and implementation of machine learning tasks and applications.

2. Dataset

The Fashion MNIST dataset consists of a collection of 70,000 unique grayscale images each depicting various fashion items such as T-shirts, dresses, trousers, etc. Images are further categorised into 10 distinct classes representing different fashion items. It is similar in structure to the classic MNIST dataset, making it a most suitable and perfect alternative to test and develop image classification algorithms [1].

Each grayscale image present in the Fashion MNIST dataset has 28x28 pixels which results in 784 features per image. The dataset is evenly balanced with an equal number of images for each class offering a fair evaluation of classification algorithms.

Table 1 – Statistical Summary of Fashion MNIST database

Dataset contents	Number of Images
Training	60,000
Testing	10,000
Classes	10
Image Size	28x28 pixels
Color Channels	Grayscale
Total Features	784

3. Methodology

The Dataset is split into 60,000 images for training and 10,000 images for testing in an 85% and 15% ratio and the pixel values are normalized to the range of [0,1] to ensure optimal performance [1].

Both Models were processed using GridSearchCV to employ and explore various hyperparameters, such as learning rate, momentum, dropout rate, weight decay and number of filters for convolutional layers. GridSearchCV searches through a grid of hyperparameters and finds optimal combinations that yield the best performance [4].

In both models, GridSearchCV performs cross-validation to find the best-performing combination of hyperparameters which is determined by its accuracy. Stochastic gradient descent optimizer minimizes the error of the model and updates the hyperparameters in the opposite direction of the gradient of the loss function[6]. According to [4] GridSearchCV is best suited to help identify the hyperparameters which generalize well to the unseen data. Both MLP and CNN were evaluated on their accuracies obtained on the test data as well as during their training accuracies along with the total time taken for training was evaluated to select the best model.

3.1. Architecture and Parameters of MLP

MLP model is implemented with three fully connected layers and it pays no consideration to the spatial relationship of the input images. As [5] mentions, each neuron inside the fully connected layer treats each pixel in the image as an independent feature which in turn takes away its ability to capture

spatial patterns and structures. The input layer has 784 neurons and the subsequent layers progressively reduce the number of neurons with 512 in the first layer, 256 in the second layer and 128 in the third layer. The output layer of the architecture consists of 10 units which corresponds to the 10 classes of the Fashion MNIST dataset [1] [2].

During training, epochs were set to [10,20] to enable converging the model to a stable solution alongside learning rates of [0.01, 0.05, 0.1]. One slower and one faster learning rate was chosen to provide more stability. Momentum is set at [0.7,0.9] to offer faster convergence in case of noisy gradients. To avoid overfitting of the model dropout values of [0.5,0.6] were implemented to avoid the model's overdependency on specific neurons during training.

3.2. Architecture and Parameters of CNN

CNN model implemented in the study begins with two convolutional layers followed by and ReLU activation function. Each of these convolutional layers uses 5x5 kernel sizes to extract features from the input images. The first layer will take a single channel input and outputs 32 feature maps however the second layer will take 32 feature maps as input and outputs 64 feature maps. Further max pooling layer of kernel size 2x2 is applied to reduce the spatial dimensions of the feature maps while extracting the most salient features. The first fully connected layer consists of 256 neurons while the second fully connected layer produces the final output of 10 units corresponding to the 10 classes of the Fashion MNIST dataset [1][2].

Over the course of [10, 20] epochs, filters were increased from 32 to 64 allowing the model to learn different levels of abstract features from the input images, as a higher number of filters offers more diverse and complex patterns. Dropout rate of 0.5 is applied to the first fully connected layer giving a 50% probability to the deactivation of each neuron during the training iteration to prevent overfitting. A learning rate of [0.1,0.01] is provided for the model to be more stable though it takes more time for all the training iterations to converge. Both MLP and CNN models use a Stochastic Gradient Descent optimizer to update the parameters from gradients of the loss function while minimizing the loss from model performance.

4. Results, Findings and Evaluation

4.1. Model Evaluation

Table 2 – Parameters obtained during Grid Search

CNN

Learning rate	max_epochs	module__dropout_rate	Mean Score
0.1	20	0.5	0.893
0.1	20	0.5	0.892
0.1	20	0.5	0.892
0.1	20	0.5	0.891
0.1	20	0.25	0.890
0.1	20	0.25	0.887
0.1	20	0.25	0.887
0.1	20	0.25	0.887
0.1	10	0.5	0.877
0.1	10	0.25	0.876
0.1	10	0.5	0.872
0.1	10	0.5	0.872
0.1	10	0.25	0.871
0.01	20	0.25	0.836
0.01	20	0.25	0.836
0.01	20	0.25	0.827
0.01	10	0.25	0.793
0.01	10	0.25	0.793
0.01	10	0.5	0.792
0.01	10	0.5	0.791

MLP

Learning rate	max_epochs	module__dropout_rate	Mean Score
0.05	20	0.5	0.877
0.01	20	0.5	0.875
0.01	20	0.5	0.857
0.01	20	0.5	0.855
0.1	20	0.5	0.862
0.1	20	0.5	0.862
0.05	20	0.5	0.861
0.01	20	0.5	0.854
0.01	20	0.5	0.847
0.01	20	0.6	0.845
0.1	10	0.5	0.808
0.1	10	0.5	0.787
0.1	10	0.5	0.850
0.1	10	0.5	0.844
0.1	10	0.6	0.713
0.1	10	0.6	0.735
0.1	20	0.5	0.775
0.1	20	0.6	0.639

Table 2 indicates that CNN performs relatively while during training compared to MLP. Both models were applied to grid search which explored various combinations of hyperparameters shown in Table 2. The best combination of hyperparameters observed for CNN is learningrate=0.1 with dropout=0.5 and running 64 filters for each convolutional layer. It managed to achieve a mean accuracy score ranging from 78% – 89% with varying hyperparameters.

MLP on the other hand observed learningrate=0.1 with dropout=0.5 and optimizer momentum=0.7 as best hyperparameters. The model managed to get in 63% - 87.70% of the training score. Both models achieve an improved performance with an increased number of epochs as observed by a higher mean score for 20 epochs compared to 10 epochs.

4.2. Performance Evaluation

CNN achieves a test accuracy of 91.84% compared to MLP which scored 87.97% in test accuracy. While comparing training accuracy results both were extremely close as observed by Table 2. MLP with its 87.70% doesn't fall far behind CNN with its 89% training score. However, CNN leverages its convolutional layers which allows it to capture spatial hierarchies in the input image data which results in better performance on image classification tasks. Its training and test scores are indications that the CNN model performs relatively well on unseen data compared to MLP.

Figure 1:Performance matrix of CNN

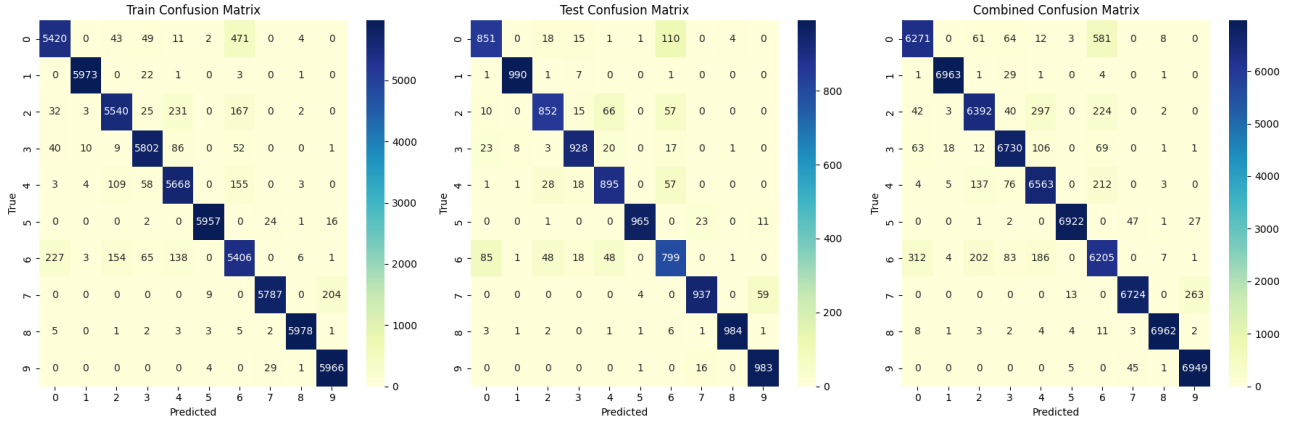

Figure 2:Performance matrix of CNN

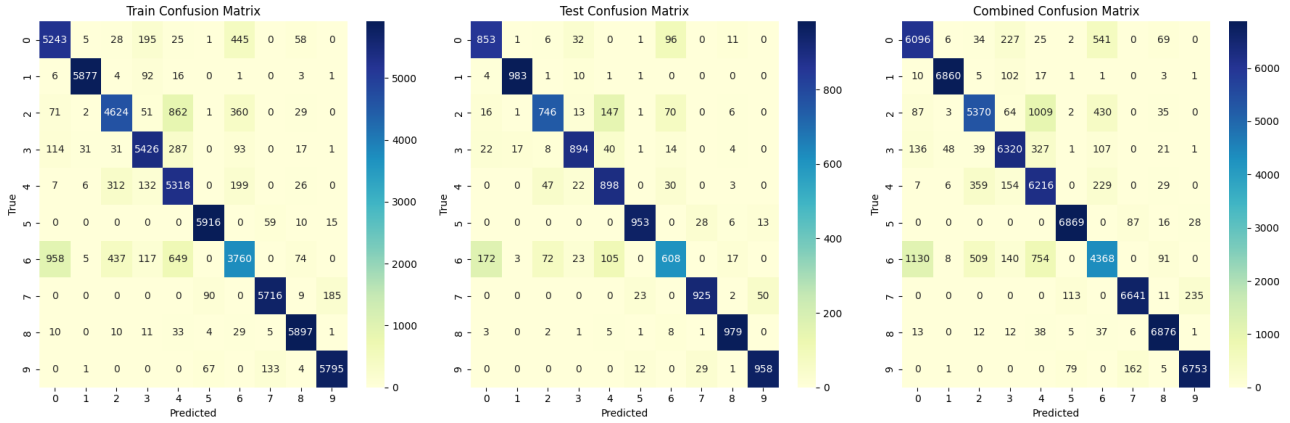

Figure 1 and Figure 2 clearly suggest that the MLP classifier achieves higher precision while predicting 'Sandel' and 'Ankle Boots' classes whereas the instance of classes like 'Sandel',' Trouser' and 'Bag' achieve excellent performance under CNN. both CNN and MLP achieve moderate results while classifying 'Tshirts/Tops', 'Dress' and 'Coat'. MLP comparatively struggles In the prediction of the 'Shirt' and 'Pullover' categories with 0.74 and 0.75 precision rates for each respectively whereas CNN achieves 0.84 precision in the 'Shirt' category but has a lower Recall value. this suggests that both models comparatively struggle while predicting the instances of 'Shirt' and 'Pullover' classes.

CNN achieves higher macro and micro average metrics compared to MLP indicating a strong and balanced performance.

5. Conclusion

In this study, two powerful neural models were trained and tested to classify 70000 grayscale images of clothing items from 10 categories, consisting of 7000 images per category [1][2]. Both CNN and MLP models demonstrate effective performance and achieve competitive accuracies with hyperparameter tuning. CNN takes leverage of the spatial relationship within the input images using its convolutional layers capturing relevant hierarchical patterns and features from the data.

MLP on the other hand makes use of its fully connected layers (Dense Layers) and effectively learns data representation leveraging on its various hyperparameters such as optimizer momentum, weight decay, and dropout rate. while MLP did exhibit good performance across most classes overall CNN achieves higher accuracy and slightly better scores throughout the various evaluation metrics. However, both of the models in this study face challenges when identifying the instances of shirt and pullover classes.

Both CNN and MLP models trained in this study are heavily influenced by the choice of optimizer and its parameters. Optimizers like SGD, Adam and RMSprop implemented in this study effectively update model parameters during training and minimize the loss function. For the sake of convenience base models without hyperparameters were employed for both MLP and CNN, where a steep rise in test accuracy was witnessed in CNN. MLP barely displayed any improvement.
However, there were several other techniques that could further improve the convergence speed and final performance of the model which was not successfully achieved in this study.

One such approach is incorporating learning rate decay or cyclic learning rates with methods like AdamW to dynamically adjust the learning rate during training [6]. Another improvement for future work is going beyond dropout by using regularization methods such as L1 and L2 regularization or data augmentation, which helps to prevent overfitting [7].

To conclude, the CNN model exhibits superior performance on image classification for the Fashion MNIST dataset compared to MLP. However, by this study, we learn the importance of careful hyperparameter tuning indicating that depending on the nature of a given problem and specific requirements either of the models can be suitable for deployment.

6. References:

1. Xiao, H., Rasul, K., & Vollgraf, R. (2017). Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms. arXiv preprint arXiv:1708.07747.
2. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning (Vol. 1). MIT press.
3. LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.
4. Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. Journal of Machine Learning Research, 13(Feb), 281-305.
5. Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.
6. Ruder, S. (2016). An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747
7. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(1), 1929-1958.