Fashion MNIST

Leon Petrinos, Andrea Trugenberger, Youssef Chelaifa

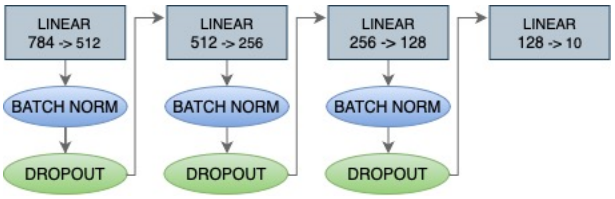
1 Introduction

In this project, we explore the application of three distinct neural network architectures to the Fashion MNIST dataset: Multilayer Perceptrons (MLPs), Convolutional Neural Networks (CNNs), and Vision Transformers (ViTs). The Fashion MNIST dataset is a collection of grayscale images of clothing items, each sized 28 by 28 pixels.

2 Methods

Multilayer Perceptron (MLP)

MLPs are a class of artificial neural networks that consist of multiple layers of neurons. Each neuron in a layer is fully connected to the neurons in the previous and following layers. Our best validation accuracy was achieved with an architecture comprising three hidden layers, each with ReLU activation functions. To address overfitting, we incorporated dropout layers, which randomly set a fraction of the input units to zero at each update during the training process. This encourages the network to develop more robust features. Additionally, we integrated batch normalization to accelerate the training process.

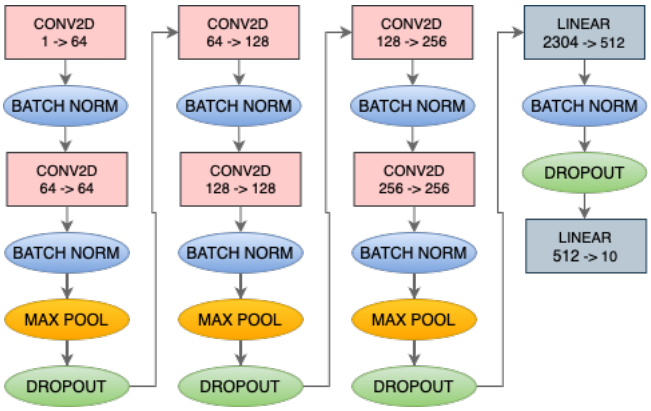


Vision Transformer (ViT)

ViTs treat images as sequences of patches (smaller dimension images) and apply self-attention mechanisms to model global relationships within the image. Despite its promise, the computational time for ViTs was significant, making it challenging to finely tune and identify the optimal parameters within a reasonable timeframe.

Convolutional Neural Network (CNN)

CNNs are specifically designed for processing grid-like data, such as images. They use convolutional layers, which apply filters to the input data to capture complex patterns. After extensive training and evaluation of all three networks, we found that the CNN provided the best results and was relatively efficient to train slightly slower than the MLP but much faster than the ViT. Consequently, we focused on optimizing our CNN architecture. We began with two convolutional layers and progressively increased the number of layers, adding batch normalization and dropout layers. We observed continual improvement in validation accuracy until it plateaued and sometimes even decreased with further additions. Thus, we finalized our CNN architecture as follows:



Data Preprocessing

For all methods, we began by normalizing the data to ensure consistency across inputs. For the MLP, we also implemented Principal Component Analysis (PCA) as a dimensionality reduction technique (unsupervised learning). PCA enhances computational efficiency and can potentially improve model performance by focusing on the most significant features of the data.

3 Training with Validation Set

For each of our network architectures, we tuned the hyperparameters to obtain the best accuracy and macro F1 score by splitting the training data into 80% train set and 20% validation set.

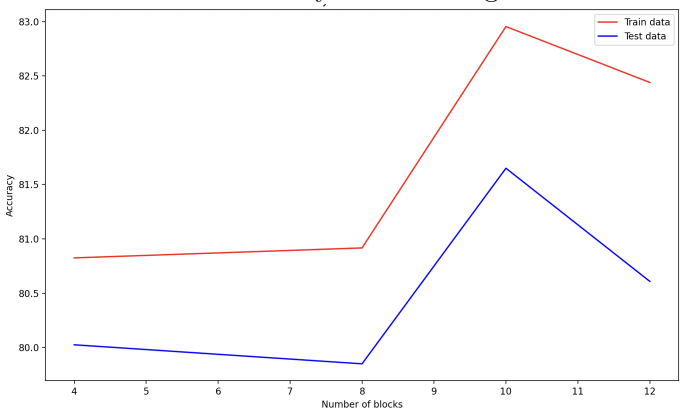
MLP Performance

To train the MLP, we fixed the number of epochs to 30 as this generally yielded good accuracies. We thus focused on comparing the accuracy with respect to the learning rate when using PCA versus not using PCA.

The optimal learning rate yielding the highest validation accuracy of 87.2% is $2 \cdot 10^{-4}$. We obtained a macro F1 score of 0.867. We can see that there is about a 10% difference between training and validation accuracy, indicating that the model is not overfitting. The results clearly show that applying PCA enhances accuracy across all learning rates (the yellow line is always above the blue one). This consistency of PCA demonstrates its effectiveness in feature reduction and overall model performance.

ViT Performance

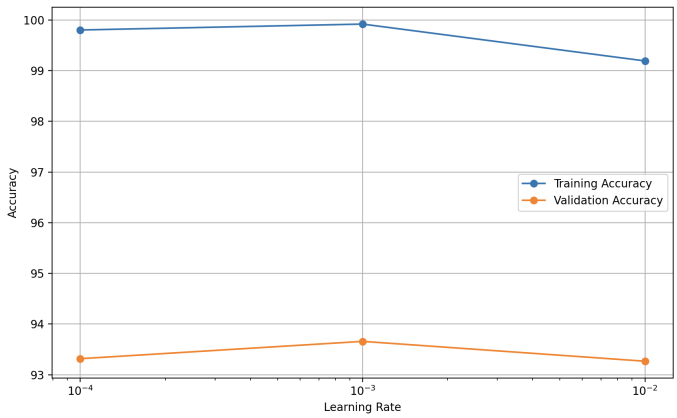
For the transformer we decided to run on 5 epochs at most to avoid rendering the computation times too long. We decided to vary the number of blocks and see how the accuracy would change.



According to this graph, it seems that the optimal number of blocks would be 10 yielding a validation set accuracy of 81.6%.

CNN Performance

The following graph shows how the accuracy varies when varying the learning rate. We fixed the number of epochs to 100 as we found that this yielded a good accuracy. Running for more epochs would simply take more time without a great improvement.



It seems that there is potential overfitting due to the very high train accuracies. Nevertheless, the validation accuracies are also very high suggesting that this is a good model. To verify this we later test it on the test set. The learning rate giving the highest validation accuracy of 93.7% is 10^{-3}. We got a macro F1 score of 0.936.

4 Performance on Test Set

We submitted our 2 best models on the AIcrowd contest to see how they performed on unseen data (we omitted the transformer as the computation time was too long). Here are our results:

Model	Accuracy	Macro F1 Score
MLP	86.0%	0.873
CNN	93.6%	0.942

5 Conclusion

Among the tested models, the CNN consistently outperformed the others in terms of accuracy and efficiency thanks to its superior ability to handle image data which made it the most suitable model. Also the MLP was showing a significantly fast computation time unlike ViT whose limitations in computational efficiency were notable. Furthermore, we were not able to test many different architectures as this would take too much running time especially for the transformer. On the other hand, we could only include the GPU functionality for CNN and MLP opposed to ViT for which we used the cpu.