Comp551 - Assignment 3

Members:
Adam Dufour(261193949)
Gabriel Caballero (261108565)

November 23, 2025

Abstract

In this work, multiple Multi-layer perceptrons (MLP) were constructed with varying depths, activation functions, and regularization parameters. The findings of this work showed that increasing depth leads to better models, however there is a plateau as $h_n \to \infty$. It seems that ReLu and LeakyRelu result in better accuracy than tanh as activation functions. As for the regularization, both L1 and L2 regularization achieve very similar results, but with the L1 model narrowly being the best MLP overall achieving 88.7% accuracy. In addition, a convolutional neural network (CNN) was used to explore a different architecture, the CNN showed better results as the accuracy increased to 93.47% compared to 88.13% on the default 2 hidden layers MLP. Finally, we experiment with transfer learning by freezing convolutional layers of a pretrained ResNet18 model. We compare several architectures and were able to achieve an 85.41% accuracy with this method.

Introduction

MLPs are a very powerful and useful architecture used to many different ends. To assess the construction and reasoning behind many key concepts of MLP, the Fashion-MNIST dataset was studied. This dataset contains tens of thousands of images each labeled of a clothing item. The model's depth, activation functions and regularization approach. This work reports the findings and results obtained across various different MLP structures. MLPs are great but will also be compared with a CNN model and ResNet to assess performance across different models.

Datasets

The dataset that has been used in this work has been retrieved from [5] and consists of : article images. Fashion-MNIST, the dataset, has collected various items form the Zalando catalog, a clothing and accessory brand. This dataset is based on another dataset, the MNIST dataset composed of hand-written digits. The use of MNIST has been to validate their approach as it is recognized as a great reference for ML models [3], Fashion-MNIST recreates the idea of MNIST for validating models. Looking closer into Fashion-MNIST, it is split in two sets, training set and testing set each containing 60 000 and 10 000 examples respectively. These grayscale images of 28 by 28 pixels are spanning over 10 different labels of different clothing items.

Results

Task 1 - Retrieving and cleaning the Dataset

Basing our work on [1], the Fashion-MNIST dataset was cleaned and the images were processed in order to prepare the construction of the MLP model. First, the images were reshaped into vectors of 784 pixels, compiling each pixels of the 28 by 28 raw image. The images were then scaled to normalize the inputs, zero centered to achieve a $\mu = 0$ distribution, PCA was applied to reduce the dimension of the input and decorrelate the data, finally the data was whitened to normalize the scale of the PCA dimensions. To visualize the pre-processing, the figure 1 shows the impact of zero-centering and reconstruction of the image following PCA processing.

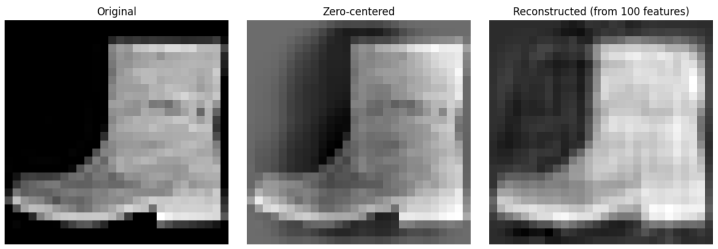

Figure 1: Input image following pre-processing

Hyperparameter Choice

Figure 2 shows the tuning that was done in order to choose the hyperparameter values. Learning rate, batch size and epoch count are crucial in the construction of the different MLP. For regularization, λ_{L1} and λ_{L2} were tuned in order to validate the choice of hyperparameter for the rest of the experiments.

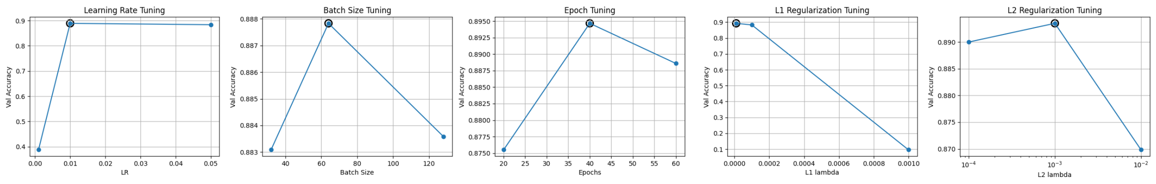

Figure 2: Model performance across hyperparameter tuning

Task 2 and 3 - Implement MLP and run experiments and obtain statistics

The implementation of the MLP followed the architecture shown on the lecture slides.

Varying MLP Depth

First off, the depth of the MLP was studied in order to grasp the importance of hidden layers in the model. The weights were optimized using min-batch gradient descent and used the cross-entropy loss function shown below :

$$L_{CE} = -\sum_{i=1}^{C} y_i \log(p_i) \tag{1}$$

This loss function is suited for a multi-class regression problem like the one in this case. Table 1 shows the performance of each model at varying depth. The key takeaway from this table is that increasing depth seems to have shown better results on the datasets, however the 2 layers MLP did not increase significantly the performance of the model over the 1 layer MLP. This result is aligned with the theory that states that increasing depth leads to better models (might however plateau as $h_n \to \infty$). Also, by increasing depth, the time of training increased significantly showing that, while performance increases, the time taken to train can be a valid reason to limit the depth of the model. Implementing non-linear activation functions allowed to capture relation between the data that are not linearly separable, hence making the model more expressive and gave better results.

Depth of the model	**Accuracy (%)**	**Training time (s)**
0 hidden layers	83.91	3.52
1 hidden layer	87.94	28.71
2 hidden layers	88.13	55.78

Table 1: Results reported for different depth MLPs.

Varying MLP Activation Function

Secondly, different activation function were used to monitor the performance of the model. Table 2 shows the performance of the different activation functions. It seems that ReLU and Leaky ReLU outperform tanh, which suffers from vanishing gradient issues. Leaky ReLU provides a small but consistent improvement over ReLU.

Activation Function	Accuracy (%)
ReLU	88.13
Leaky ReLU	88.43
tanh	85.49

Table 2: Results reported for different activation functions (2 hidden layers).

Varying Regularization

Thirdly, the implementation of regularization lead to the results shown in table 3. L1 regularization achieves the best performance (88.74%), slightly outperforming both L2 and the unregularized model. But in general they all performed at roughly the same accuracy. As for the combination of L1 and L2, it creates what is known as an elastic net. This regularization method has been proven to be efficient and yields good results with better tuning [2].

Regularization method	Accuracy (%)	Training time (s)
None	88.13	55.78
L1 regularization	**88.74**	87.79
L2 regularization	88.40	63.58
L1 and L2	88.37	94.66

Table 3: Results reported for different regularization methods.

Figure 3 shows the results obtained from the previous MLP models. Out of all of these, MLP Q6 gave the best result. It also shows that without normalization, the model can still perform accurately. Although we did encounter runs where the un-normalized model was unstable (NaN gradients), so its always safer to at least do input scaling. Regarding time to train, it is possible to observe that L1 regularization increases the training time much more than L2, this can be explained by the core difference between L2 and L1, L2 is simply smoother than L1 and that changes decreases time to train.

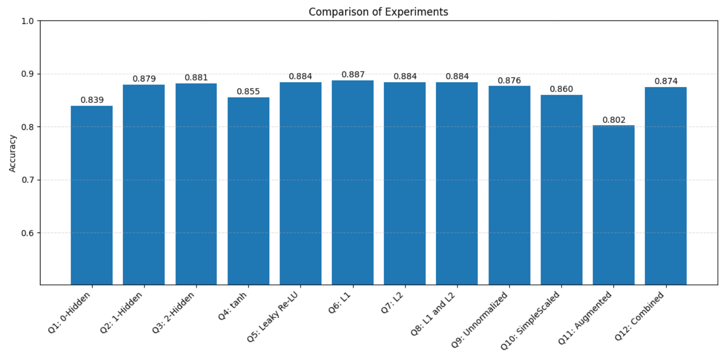

Figure 3: Model performance for the different MLP

CNN and ResNet implementation

Implementing the CNN gave better results than using the previously studied MLP. In fact, table 4 shows that gain in performance. This result is aligned with the theory since a CNN has, in its definition a better notion of of the image's content and context, that is relevant in a cases where images are trying to be understood. Giving this information to the model will certainly lead to a better performance and that is what is observed. Training this model with data-augmented input alone wasn't enough to improve performance, but its performance improved significantly when combining the original dataset with the augmented one, yielding our best overall model at 93.47% accuracy. Finally, using a pre-built model ResNet [4] yielded 85.41%, this result indicates that transfer learning is a more or less efficient technique at predicting new data when trained on a similar but different dataset, but with by far worse accuracy / training time ratio. Figure 4 shows that a more connected CNN (ResNet) did not lead to a more accurate model, so a more scarce model is well equipped to deliver good performances.

Model	**Accuracy (%)**	**Training time (s)**
Best MLP (2 hidden, L1)	88.74	87.79
MLP (augmented only)	80.22	84.90
MLP (augmented + original)	87.44	169.92
CNN (no augmentation)	91.94	103.5
CNN (augmented dataset)	91.92	240.5
CNN (augmented + original)	**93.47**	356.5
ResNet18 (frozen conv + 2 FC layers)	85.41	1012.2

Table 4: Results reported for different models (accuracy and training time).

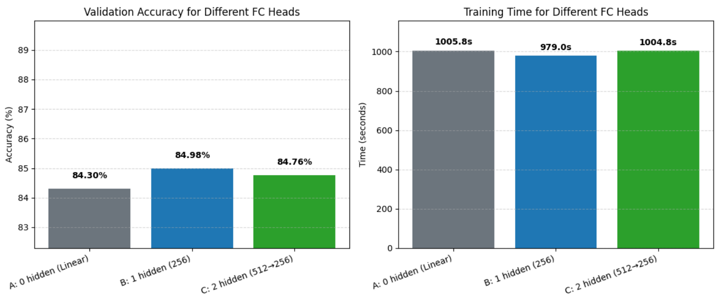

Figure 4: Performance of CNN at different connectivity levels

Discussions and Conclusions

As observed in this work, MLPs are complex and contain many key components that require optimization. This report showed the impact of these different aspects on the results of the MLP. Our best-performing MLP used two hidden layers with ReLU activation and L1 regularization, achieving 88.74% accuracy. But it also showed that, in some cases, increasing the model's complexity did not contribute significantly to the model's success rate, as can be seen in Figure 3. For example, the difference in performance between the unnormalized model and our best performing MLP is only 1.1%, although this is only when training is stable. Overall, the most significant change to the MLP made during this exercise was the increase in depth from 0 hidden layers to at least 1. This being because a model with no hidden layers is essentially a logistic regression classifier. Finally, implementing a CNN on the same dataset showed how their ability to capture spatial structure makes them that much more effective for image data, implementing the pre-built model ResNet allowed to visualize how generalizable are CNN on unseen data. These models allowed to observe different architecture and explained once again the tradeoff between performance and real world limitations as by using more sophisticated models. CNN. the training time fot longer by more than 10 times.

Additional work

An additional point that was observed during the process is in the PCA reconstruction of the input images. Figure 5 shows the different components extracted during the PCA processing, showing clearly that, in this case, the shirt is broken apart and different features of the image are highlighted. For example, PC 6 seems to show relevance to the shoulders and PC 2 seems to point out the arms besides the shirt. With these components, reconstructing the image with a varying amount of PC yielded figure 6 that shows that even at a low resolution, the image still retains an impressive amount of information. Figure 7 corroborate this finding as it is possible to visualize the fact that the variance quickly rises as the components increases. This figure shows that in the dataset, it is possible to achieve a high diversity at a low amount of components. This refers again at the amount of information available even at smaller dimensionality.

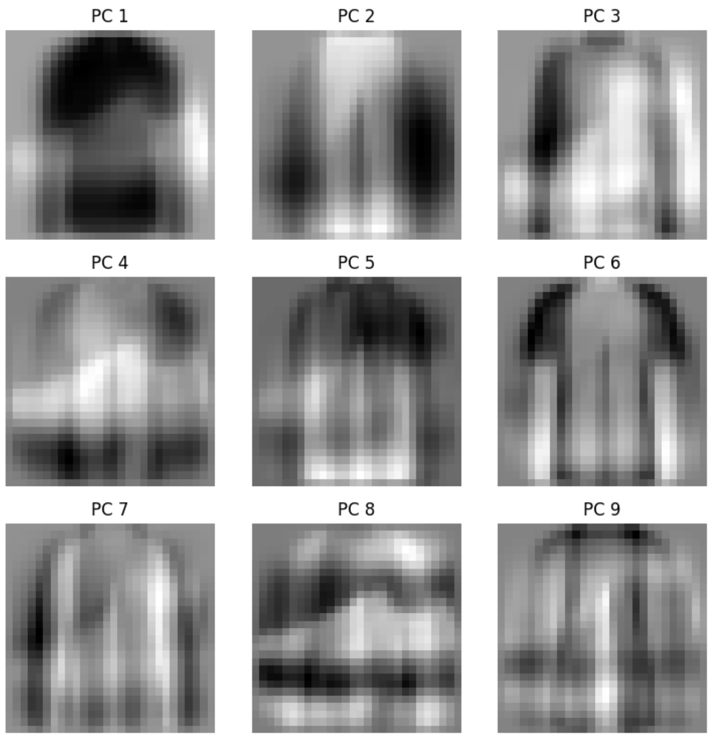

Figure 5: PCA components of the image

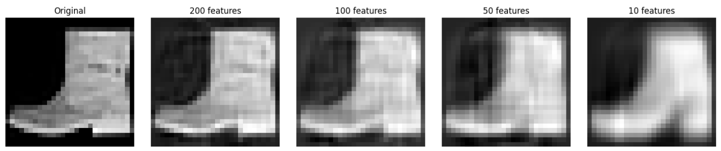

Figure 6: Reconstruction of the image with decreasing features

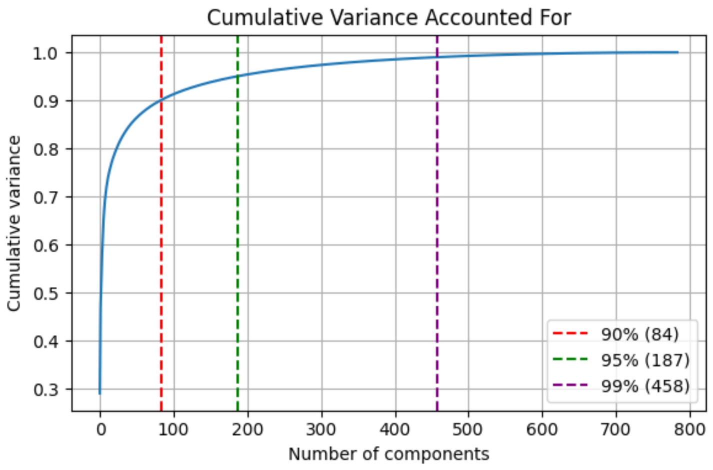

Figure 7: Variance of dataset plotted across component number

The implementation of L1 and L2 regularization at the same time gave the results observed in table 3, performing very similar to just choosing L1 or L2, indicate that with a proper grid search over more combinations of L1 and L2 could result in a better model.

Statement of Contribution

Gabriel Caballero: Data cleanup, Implementation of MLP in task 2, testing of the MLP in task 3 and implementation of the CNN for task 3, writeup section Hyperparameter tuning and additional work

Adam Dufour: Data cleanup, Implementation of MLP in task 2, writeup sections Introduction, Results, Discussion and additional work

References

[1] Data preprocessing — cs231n neural networks part 2. https://cs231n.github.io/neural-networks-2/datapre. Accessed: 23 November 2025.

[2] Elastic net regularization. https://en.wikipedia.org/wiki/Elastic_net_regularization. Accessed: 23 November 2025.

[3] Yann LeCun, Corinna Cortes, and Christopher J.C. Burges. The mnist database of handwritten digits. http://yann.lecun.com/exdb/mnist/. Accessed: 2025-11-23.

[4] Great Learning Editorial Team. What is resnet? https://www.mygreatlearning.com/blog/resnet/, 2025. Accessed: 2025-11-23.

[5] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. https://www.kaggle.com/datasets/zalando-research/fashionmnist, 2017.