MiniProject Report 3
Classification of Image Data

Adalric Leung
John Pan
Gabrielle Thibault

March 25th 2021

0.1 Abstract

Image classification is widely used in present time through a diversity of applications such as medical imagery analysis to improve the accuracy of diagnostics. In the medical world, the importance of computer vision learning systems is growing with the application of image classification. In this report, different variations of Multi-Layer Perceptrons (MLP) were implemented to compare their accuracy on the Modified National Institute of Standards and Technology (MNIST) dataset to demonstrate the types of machine learning algorithms used for medical analysis. The optimized MLP was found following the project's guidelines. More specifically, it was concluded that the highest performance, to classify the digits of the MNIST dataset, can be obtained with 2 hidden layers, the ReLU activation function, a small regularization parameter and normalization of the input data. The MNIST dataset is often used as a benchmark dataset to insure good performance of image classification models.

0.2 Introduction

In the the medical area, the recognition, classification, and computation of disease patterns in images is of great help to practitioners to analyze and to conclude diagnosis. More specifically, recent research found that image classification of tissue structures have greatly improved histopathology [1]. There are several advantages in using MLP as a classification algorithm. It can model non-linear systems without making any assumptions on probability (e.g. Naive Bayes assumes conditional independence). Further, it does not need any feature engineering selection to train the algorithm, which can be a tedious job. However, MLP models have a lot of parameters to tune (e.g. number of layer, layer size, layer type, activation function, optimization technique, regularization technique, initialization of basis and more). Previous literature have also demonstrated great optimization methods to determine the best MLP architecture for specific applications. For example, Castro et al. implemented a logical sequence using parallel computing techniques [2]. Furthermore, the loss function for MLPs is non-convex which means that the model is not guaranteed to converge to a global minimum (might find local minimum). This has also been researched and multiple methods have been found to improve the performance of the models. An example is the research done by Dauphin et al. proposing a solution to the saddle points problem with the saddle-free Newton method [4]. The dataset used in this project is widely utilized for image classification. Variations of the MNIST dataset have also been explored in several research. More specifically, the Fashion-MNIST classifies pictures of fashion items [5] and the EMNIST classifies handwritten letters [3]. After testing several variations of MLP, we can conclude that 2 hidden layers, the ReLU activation function, a small regularization parameter and normalization of the input data is the best combination of parameters to predict the classification of the ten MINIST digits.

0.3 Datasets

The MNIST dataset is composed of 70000 digit images of size 28x28 represented by 784 values base on a gray scale intensity (black=0 and white=1) distribution. MNIST combines ten classes (0 to 9) and is the most widely used and tested dataset in deep learning due to its size and usefulness in prototyping models for researchers. The targets are represented by

an array labelling each picture with its corresponding digit. A few pre-processing steps have to be done before using the dataset. First, the images have to be reshaped from a 28x28 matrix to a 784x1 array. Second, the pixel values of those handwritten digits (gray scale) have to be normalized because they are initially found in a range of 0 to 255. Re-scaling the features normalizes them to the same weight and speeds up the process due to smaller size of numbers used in all the following calculations (such as gradient descent). Third, one hot encoding has to be applied on the labels to allow the categorical data to be represented as binary vectors. Each label is represented as a binary vector of all zeros except for the index of the specific class which will be 1. In table 1, we can observe that the data is well distributed between all classes. The classes containing the most input is 1 and the class with less input is 5.

Table 1: Class distribution
(amount of handwritten data for each digit)

Digit	Class distribution (%)	Digit	Class distribution (%)
0	9.87	5	9.04
1	11.24	6	9.86
2	9.93	7	10.44
3	10.22	8	9.75
4	9.74	9	9.92

0.4 Results

Multiple variations of MLPs have been tested in this project. In the figures below, the training and testing accuracy were plotted for the MLPs as a function of training epochs. This allows us to easily illustrate the continuous increase in performance of the network with the greater number of iterations. The first network architecture characteristic that was tested was the impact of the number of hidden layers. More specifically, the performance of the following three version of MLPs have been compared: no hidden layer, single hidden layer and two hidden layers. In figure 1, we can observe that increasing the depth increases the accuracy of the model. The model for one and two hidden layers have similar accuracy, but the two layer MLP is performing slightly better. More specifically, at epoch 8, they respectively have an accuracy of 90.69% and 91.58%. This can be explained by the increased expressiveness of the model with a greater number of layers.

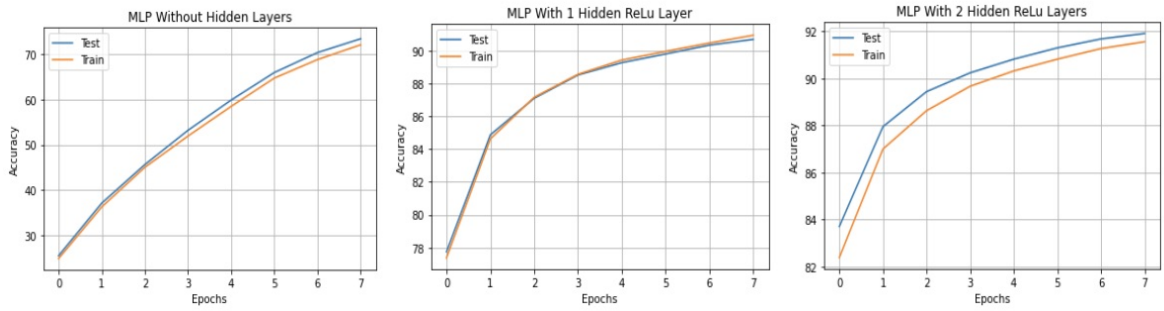

Figure 1: Accuracy of MLP for different number of hidden layers

The nonlinearity of the model is linked to the composition of the layers of nonlinear activation functions which will be further discussed in the following paragraphs. Increasing the number of layers can be more effective than increasing the number of parameters to obtain greater performance. A greater depth in an MLP can usually translate into a more expressive model for complex representation. However, more layers does not always translate into a better accuracy because it could cause instability in the network and overfitting.

Different activation functions have also been tested in this assignment. More specifically, the performance of an MLP with either sigmoid, hyperbolic (tanh) or Rectified Linear Unit (ReLU) activation functions has been compared. The advantage of the tanh activation function is that it will strongly map the negative input in the negative class and, similarly, the zero input will be mapped close to zero. The sigmoid activation function is used when we have to predict probability of an output being between 0 and 1. Most of the time, a generalization of the logistic function (i.e. softmax) is used because of its application for multiclass problem. Finally, ReLU is very popular for it's sparsity and a reduced likelihood of vanishing gradient characteristics. It is also computationally efficient. The disadvantage of ReLU is that it maps the negative values to zero immediately which affects the training result because those values might not be classified appropriately. This can be solved with the 'Leaky' ReLU. Figure 2 compares the accuracy given by all 3 activation functions for our model. The best one is ReLU followed by tanh and finally by sigmoid function. Tanh is more often used in binary classification which is why we can observe in figure 2, a lower performance with this activation function compared to ReLU. However, hyperbolic tangent often performs better than sigmoid due to it's greater symmetry (i.e. closer to zero).

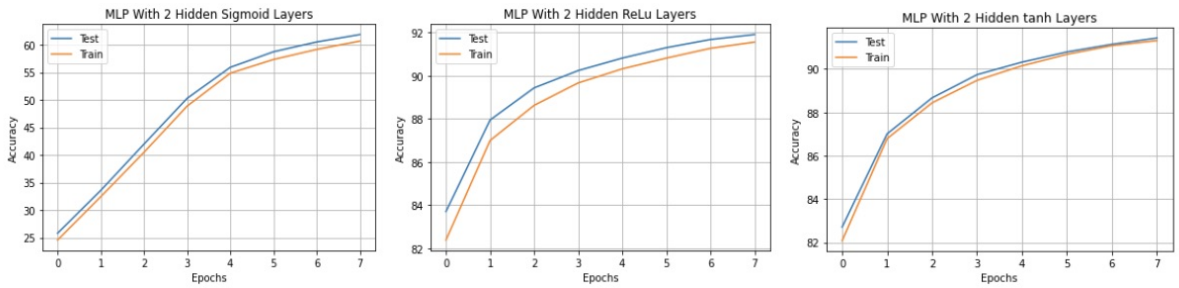

Figure 2: Accuracy of MLP with different activation functions

L2 norm regularization is used to bring the weights of the implemented function towards the origin without bringing them to zero. This decreases the chances of over-fitting the training data and obtaining poor result on the test error. After tuning the regularization parameter (lambda), we concluded that the lower the assigned value of lambda, the greater the accuracy we obtained. In figure 3, a value of 0.001 is assigned to lambda. In more complex models, regularization decreases the strength of the weights to improve the generalized error and decreases the chance of over-fitting. However, in our case, adding regularization over simplifies our model which results in under-fitting and lower accuracy.

In figure 4, we can observe that normalizing the input data affects the performance of our algorithm greatly. Indeed, normalization brings all input image pixel numbers to a value between 0 and 1. This removes numerical instabilities when used with activation functions.

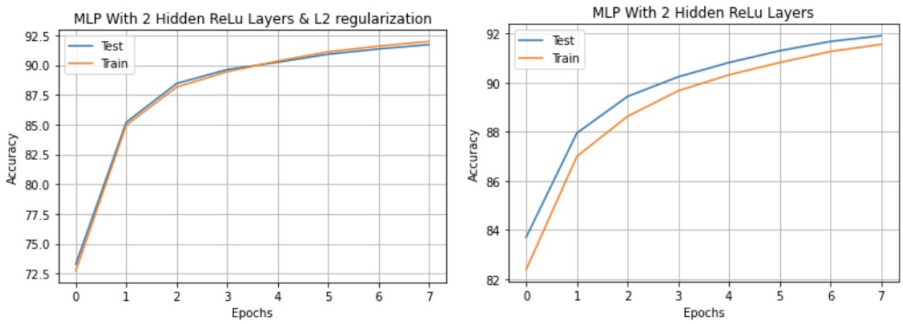

Figure 3: Accuracy of MLP with and without regularization

Without normalization, the inputs would not be at a common scale. As such, anomalies and outliers in the data can negatively impact the prediction. Furthermore, normalizing the raw input will decrease the size of the values we use in the rest of our calculations (like gradient descent) which will result in better computing speeds.

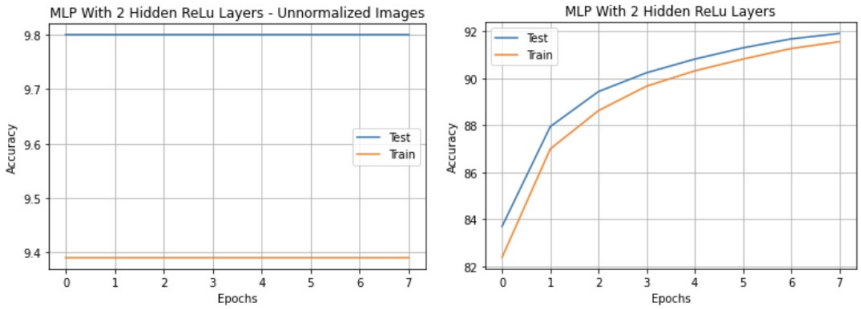

Figure 4: Accuracy of MLP with and without normalization

The neural network is shown to be better performing when trained on larger datasets. In figure 5, it can be observed that the accuracy of the network sharply increases when more images are fed into it (i.e. larger dataset). This is corroborated by the theory that neural networks perform better on large datasets.

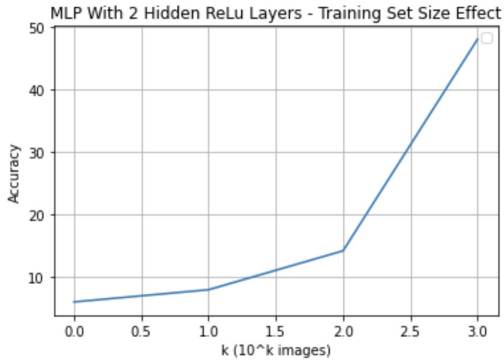

Figure 5: Accuracy of MLP with different size of subset from the dataset

The effect of width is also investigated in order to determine its effect on the accuracy of the model. It was observed that the accuracy of the model has an initial sharp increase as the width is increased, but eventually tapers off with further increase. For instance, we observe an accuracy increase of 5.6% going from a width of 32 to 128, but only an increase of 3% going from 128 to 512. This can be observed in figure 6 below.

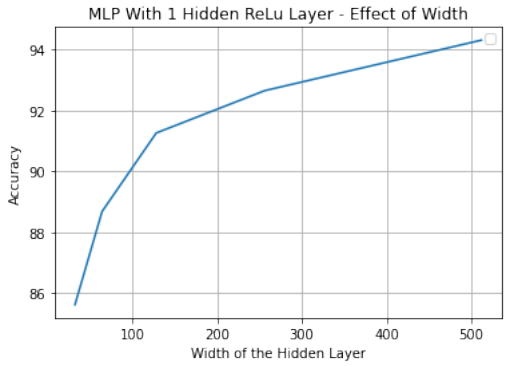

Figure 6: Accuracy of an MLP with different widths (1 ReLu hidden layer)

0.5 Discussion and Conclusion

The accuracy of neural network models, more specifically a multilayer perceptron, depends on multiple parameters. Throughout this assignment multiple MLP models were implemented to classify the ten different classes of the MNIST dataset. The number of layer, layer size, layer type, activation function, optimization technique, regularization technique, normalization of data, all impact the performance of the algorithm at different degrees. In this assignment, we explored the effects of several of these variations for our MLP model. Our best version of the MLP model that was designed had two hidden layers, ReLU activation function, a small regularization parameter and normalization of the input data. In general, it can be observed that the test and train performance of the MLPs increases with more training epochs in a similar pattern. A limitation to this project is that the fact that MLP has so many parameters it can be hard to optimize and find the very best combination of parameter for a specific situation. In summary, after tuning the number of hidden layers, the type of activation function and the value of the lambda regularization parameter we were able to obtained a maximal accuracy of 91.58 percent. Although the analysis provided is thorough, it is by no means exhaustive, and further work could be done on this project to improve the accuracy of our predictions. For example, next steps could include modifying and testing our best model with more complex datasets such has the EMNIST (26 classes for each letter of the alphabet). Momentum and other adaptive gradient methods could also be added to the model.

0.6 Statement of Contributions

All group members worked equally on this project.

Bibliography

[1] K. Balaji ME. *Chapter 5 - Medical Image Analysis With Deep Neural Networks.* ScienceDirect, 2019.

[2] Wilson Castro. *Multilayer perceptron architecture optimization using parallel computing techniques.* PLOS ONE, 2017.

[3] Gregory Cohen. *EMNIST: extending MNIST to handwritten letters.* IEEE, 2017.

[4] Yann Dauphin. *Identifying and attacking the saddle point problem in high-dimensional non-convex optimization.* Cornell University, 2014.

[5] Han Xiao. *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms.* dblp, 2017.