# Foundations and Frontiers of Machine Learning Group Assignment 2 Group 1

Contents

[Foundations and Frontiers of Machine Learning Group Assignment 2 Group 1 1](#_Toc179642033)

[1. Table of Contributions 1](#_Toc179642034)

[2. Data visualization (Task 1) 1](#_Toc179642035)

[3. Perceptrons (Task 2) 2](#_Toc179642036)

[4. Multilayer Perceptrons (Task 3) 2](#_Toc179642037)

[5. CNN (Task 4) 2](#_Toc179642038)

[6. Visualising CNN Outcomes (Task 5) 2](#_Toc179642039)

[7. Multitask learning – Fashion MNIST (Task 6) 2](#_Toc179642040)

## Table of Contributions

1. Data visualization - Emmanouil
2. Perceptrons – Emmanouil
3. Multilayer Perceptrons – Emmanouil
4. CNN - Emmanouil
5. Vicualising CNN Outcomes – Emmanouil
6. - Emmanouil

## Data visualization (Task 1)

Machine learning models use data to finetune the values of their internal parameters in a process called training. The power of the classification model is closely related to the amount of relevant information contained in the dataset, thus bigger datasets are usually preferred. However, not all information within the dataset is useful. Oftentimes, a feature contains information that has already been made available by another feature or a combination thereof, and as number of features within a dataset grows larger, the chances that a feature consists of the linear combination of other features of the dataset increase. At the same time, the mediums of visualising these datasets are usually books or electronic devices that are best suited for up to two features, as they physically have two dimensions. Using techniques such as video or colour/shape encoding can increase that by some amount, but datasets can have dimensions in the order of millions, orders of magnitude higher than what humans can visualise or comprehend (Sarkar, 2018).

A diagram of a colorful circle

Description automatically generated with medium confidenceThe issue of repeated information within the dataset can be addressed by constructing new features that encompass the dataset’s information without repetition, using with the Principal Components Analysis (PCA) technique. The first new feature is engineered to explain as much variance as possible from all features of the dataset (Maćkiewicz, 1993). Each next one is engineered to contain as much of the information that is left, without incorporating any information of the previously engineered features. This often means that the last features contain little to no additional information (depending on how linearly independent the dataset was) and can be discarded, allowing for easier visualization and reducing the overall noise. What is more, the features are now in descending order in terms of variance explained so we can select the first n ones, knowing that we have retain maximum information per feature. We can now select a number that is suitable for visualization. The new features have no actual meaning but can be used to show the distribution of the samples in the space, informing us about the separability of the classes.

Figure 1. Plot of 2 first components of PCA for all digits of the MNIST dataset

If we plot the first two components with different colour for each class, we can see which classes overlap, and which ones are linearly separable. Thus we have that, for example, the pairs 0 and 1, 0 and 2, 2 and 9 are some of the most easily to differentiate since the occupy opposing parts on the plane. Still, due to the significant overlap caused by 10 different classes, the distribution of some of the classes is obscured.

A screenshot of a computer game

Description automatically generatedA screenshot of a computer game

Description automatically generated

Figure 2a. Plot of the first 2 PCA Components for all pairs of digits

Figure 2b. Pair-wise PCA plot

Figure 2 provides a pairwise visualisation of the distributions. It Is clear that some pairs of classes, e.g. (0, 1), (0, 7), (3, 7) are relatively distinct, while others (5, 8), (7, 9) are virtually impossible to separate. However it must be kept in mind that this is the result of considering just 2 components and not all of the information contained in the dataset. That means distributions that do not seem differentiable now, may actually be when the model is trained on the original dataset. What is more, the above results are obtained by performing PCA on the whole dataset. However, when it comes to distinguishing classes in pairs, the datapoints of other classes introduce variance and noise that may be irrelevant. Thus, it is worth producing the same plot with the PCA calculated for each pair. This is shown in figure 3 and it becomes apparent that the results are better in terms of separability. To objectively measure this, we use the silhouette score which takes into account the average distance of a clusters samples between each-other and between the samples of the other class. Higher scores mean better clustering and for the generic PCA we get 0.337 where as the same score for the pairwise PCA is 0.407. This proves that we can achieve better clustering (and hence prediction performance) by performing pairwise PCA.

### Math of PCA

Before applying and transformations on the dataset, it is important to normalize it so the features with larger ranges do not affect the calculations disproportionately.

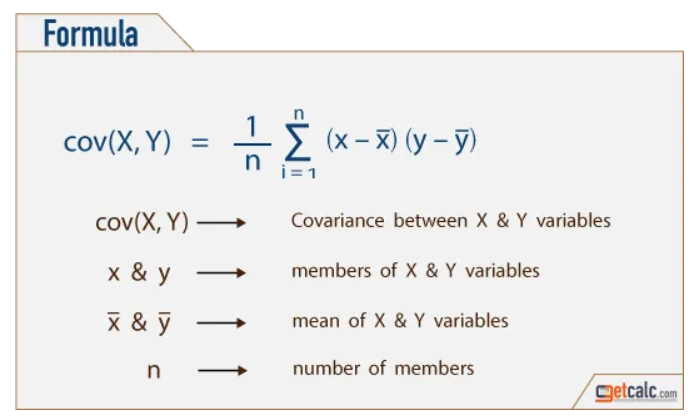
The next step is to identify how much each feature correlates with the other ones, as the more they correlate, the less useful information they add. Figure 4 shows the steps of calculation, which essentially shows that the more two variables follow the same trend, then larger the product of their difference to the mean will be.

Figure 3. Calculation of Covariance Matrix (Dubey, 2018)

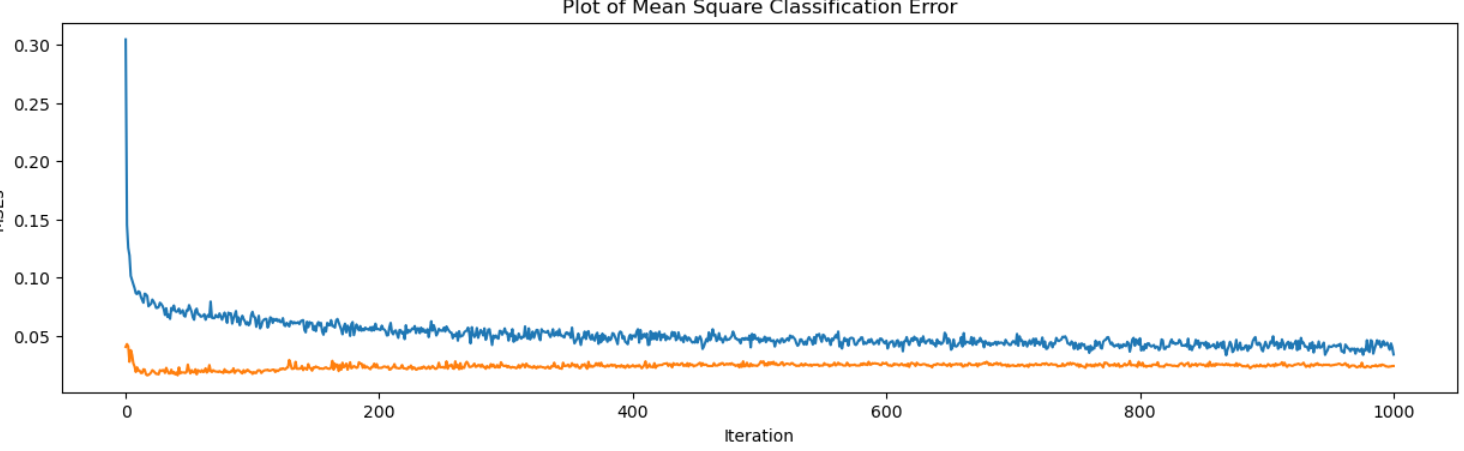
From Linear Algebra we know that a matrix can represent a transformation and its eigenvectors represent the directions in which the data has the most variance. Thus these eigenvectors will form the basis of the new vector space which is now built so that each new dimensions contains the most variance

The actual amount of variance explained by each dimension is given away by the eigenvalues of the eigenvectors. We can create a new matrix by concatenating the desired number of eigenvectors which, when multiplied with our data, produces the PCA-Data, which is our data transformed to the new base.

## Perceptrons (Task 2)

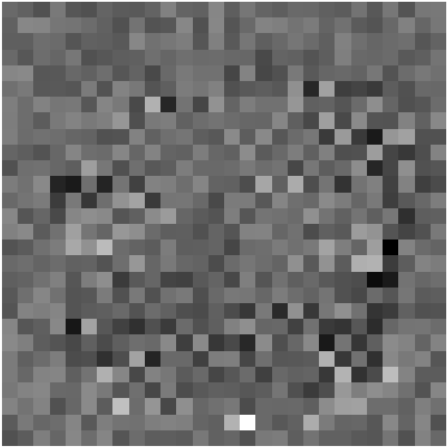
The human neurons can receive multiple stimulations that are then upregulated or downregulated by special chemicals at the neural synapses. The result then is not a linear stimulation but instead a binary state of excitation which closely resembles what we humans would call “making a decision” (Glasgow University). Rosenblatt’s perceptron machine is a software analogous which receives a multidimensional input that it scales individually, and then compares its sum to a bias in order to select one of two kinds of responses (Cornell University) . In that case, the input can be the coordinates of some samples that we know belong in different classes, and that we would like the machine to be able to distinguish. The weights and the bias are independent of the input and can be set up to do exactly that. Since the operation of scaling with a parameter and then subtracting another is the equation of the line (in two dimensions) or a hyperplane, we can set up an algorithm that hypotheses which plane would separate these classes based on the values of their coordinates, and if this hypothesis is wrong, it changes it so that it would work well for these specific samples. If the samples occupy distinct areas of the feature space and we do that for all samples, the resulting hyperplane would end up satisfying the requirement for all samples (since we know the have similar feature values). From Task 2 we know that this applies more of less for the digit pairs

The algorithmic approach consists of two functions. The first implements the matrix multiplication/scaling and the comparison with the threshold to return a result/decision. The second iterates through the samples of the dataset, producing a prediction and then correcting the weights so that if the prediction was wrong, next time it will be closer to right. The algorithms stops upon achieving an MSE goal or reaching the iterations threshold. With each iteration the number of misclassifications reduces on the training dataset, but the same is not always true for the test dataset, as the two contain different samples and too accurate a modeling of the training samples can end up imprinting noise on the model (overfitting). This is demonstrated by the training vs test accuracy, over number of iterations in figure 4.



On one hand it is clear that peak accuracy is achieved at 10-20 iterations, after which the reduction in training MSE seems to indicate overfitting rather than actual improvement. Since the weights are randomly initialized at each run, multiple runs equal to simulated annealing and can serve to expose entrapment to local minimum. As the MSE is repeatedly similar, we can infer that the algorithm does not converge to a poor local minimum.

Still, given the baseline MSE calculated in the notepad as 3.6, it is evident that the model has actually learned to distinguish between the two classes (8 and 9). The weights matrix has the same properties as the input matrix and thus, be reversing the process of transforming the digit images to input arrays, we can produce an image of the learned weights. Figure 5a provides this visualization, and makes clear that this particular simple model does not learn any high level features a human would use. Instead it may opportunistically utilize whichever pixels help it correctly split groups of samples that the training so far has not been able to assist it with, similar to nodes that split high impurity leaves in decision trees. Figure 5b shows the difference between the average image of 8 and 9. Areas with high or low brightness indicate consistent differences between the classes that could be used for a rule based system, yet we see that the model has focused elsewhere.

A close-up of a number

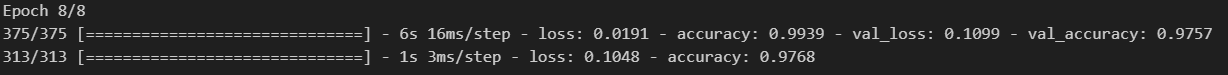
Description automatically generated A screenshot of a graph

Description automatically generated

Figure 6 is a compilation of other pairs of digits with the respective achieved accuracy after training. We see variations that are aligned with how visually similar the two digits are. It must be kept in mind that handwritten digits have different shapes that the digitally displayed ones, hence the digit ‘5’ is often drawn as an ‘S’, which can more easily be interpreted as an ‘8’.

## Multilayer Perceptrons (Task 3)

To solve the problem of the perceptron’s simplicity, we can organize large groups of them into Neural Networks (NN). The motivation stems from a property called “emergence” that we observe when multiple biological neurons work together, where the resulting structure achieves capabilities that exceed the sum of its parts (Pedder, 2023). In the software domain, the model can learn complex curves to separate the classes more effectively than hyperplanes can (Choi, 2020). Using the Keras Library we can initialize and train a network as per the assignment brief’s requirements. Even with the architecture fixed there are various hyperparameters that define how the model works. Number of epochs is similar to the iterations in task 2, and batch size allows the model to work with groups of samples speeding up training. The amount of time needed to find the best combination increases exponentially with the number of parameters so using guidelines (Bengio, 2012) and some tests, we find one that is sufficient. The resulting training and test accuracy is as follows:



At first sight it might seem as if the accuracy is lower that with the perceptron. The reason why this comparison would be misguided is that the model now performs multiclass classification, not binary classification, which is significantly more demanding.

A graph of a graph

Description automatically generatedLeaving the model to train for 30 epochs while logging train and test accuracy we get the following curves. We see that train accuracy increases along with test accuracy for the first 25 epochs after which it appears to decline. It is also interesting to compare it to the loss plot. We see that the test loss has started to increase since epoch ~10. It appears that while the predicted probability start to get worse early but the model keeps improving in terms of picking the right class for another 20 epochs, which is interesting.

## CNN (Task 4)

asdfasdfasdfasdfasd

## Visualising CNN Outcomes (Task 5)

asdfasdfasdfasdfasdfasdfasdf

## Multitask learning – Fashion MNIST (Task 6)

Asdfasdfasdfasdfasdfqwerqwerqwer

References

<https://towardsdatascience.com/the-art-of-effective-visualization-of-multi-dimensional-data-6c7202990c57#:~:text=Visualizing%20data%20in%20Three%20Dimensions,values%20in%20a%20categorical%20dimension>.

https://www.sciencedirect.com/science/article/pii/009830049390090R

<https://towardsdatascience.com/the-mathematics-behind-principal-component-analysis-fff2d7f4b643>

<https://edshare.gcu.ac.uk/4381/2/index.html>

<https://news.cornell.edu/stories/2019/09/professors-perceptron-paved-way-ai-60-years-too-soon>

<https://www.linkedin.com/pulse/what-emergence-neural-networks-chris-pedder-phd/>

<https://pmc.ncbi.nlm.nih.gov/articles/PMC7347027/>

https://link.springer.com/chapter/10.1007/978-3-642-35289-8\_26