Artificial Neural Networks and Deep Architectures, DD2437

Short report on lab assignment 1

Classification with a single-layer perceptron

Elias Kollberg, Ivar Karlsson, Emil Hed

January, 2025

1 Main objectives and scope of the assignment

Our major goals in the assignment were

- to implement and compare different learning algorithms for single layer perceptrons (classical and delta rule)
- to analyze the convergence properties of sequential versus batch learning approaches
- to investigate the role of bias in perceptron learning
- to study the behavior of perceptron learning on both linearly separable and non-linearly separable data

The scope of this assignment focused on binary classification problems in 2D space using single layer perceptrons. We limited this study to focus on two main learning algorithms, the perceptron learning rule, and the delta (Widrof-Hoff) rule - implemented in both sequential and batch modes.

2 Method

We used Visual Studio Code as our programming environment, and the programming-language used was Python. Jupyter notebooks was used since the lab required many different kinds of datasets and algorithms, thus making it convenient not having to for example re-generate the data every run. The only imported modules were numpy and matplotlib.

3 Results and discussion

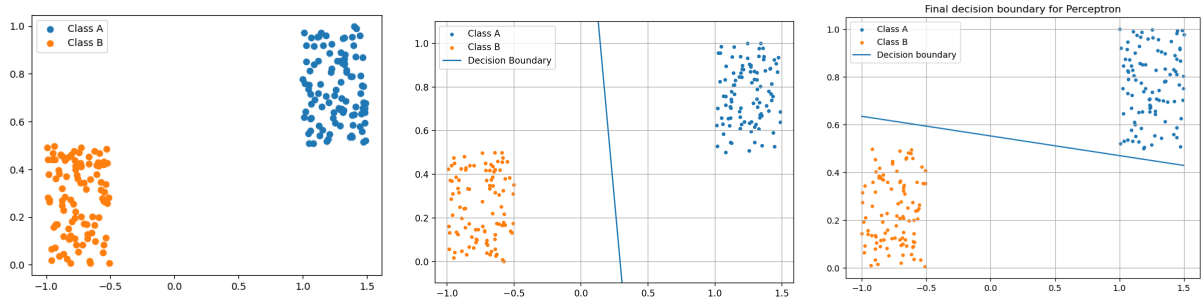
3.1 Classification with a single-layer perceptron

For the case with linearly separable data, Fig 1a, the perceptron does indeed succeed in separating the data, Fig 1c, as it always does for this type of data. However, the final boundary does not entirely match with the line a human would draw by intuition, since it stops converging as soon as no more data-points are misclassified.

We can see that the boundary generated by the delta rule is more or less perfect, considering both the amount of correctly classified points and intuition. This is assuming that we train it on enough epochs,

so that it has time to converge. When looking at the delta rule, it is more interesting to consider the difference in convergence with itself, Fig 2, the only difference being if it is run in batch mode or sequential mode. Both seem to have more or less the exact same final results (which is why we didn't consider including the graphs for the batch mode, that look identical to the sequential mode graphs). However, we can see that they converge with different speed. Batch learning is converging faster and the reason for this being that batch learning makes a better approximation of the gradient by estimating a gradient with more data points, compared to sequential learning which estimates a gradient on only one data point, leading to a much noisier gradient and thus longer convergence time. We can see that the optimal learning rate is 0.001. However, at higher learning rates, batch learning becomes notably unstable. This instability occurs because batch learning computes weight updates using the average gradient across all samples. When combined with larger step sizes, these updates can cause significant overshooting of the optimal weights, leading to divergent behavior. In contrast, sequential learning maintains stability even at higher learning rates due to its incremental update.

Finally, we look at a scenario where we remove the bias of the algorithm, which is visualized in Fig 3. Here, we generated some new data that is linearly separable, but not linearly separable from a line that passes through the origin. We get the results that are expected, the algorithm without bias fails completely to separate the data, while the algorithm with bias succeeds just like in the earlier case.



(a) Linearly separable data (b) Sequential delta algorithm (c) Perceptron algorithm

Figure 1: A comparison between linearly separable data and the results of two algorithms

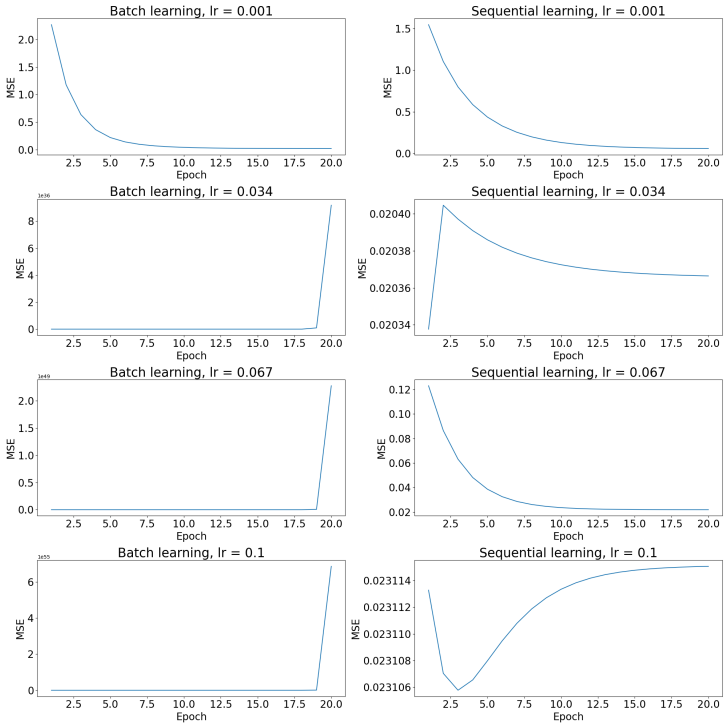
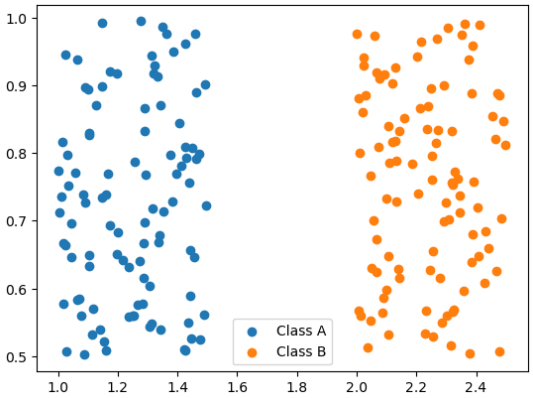
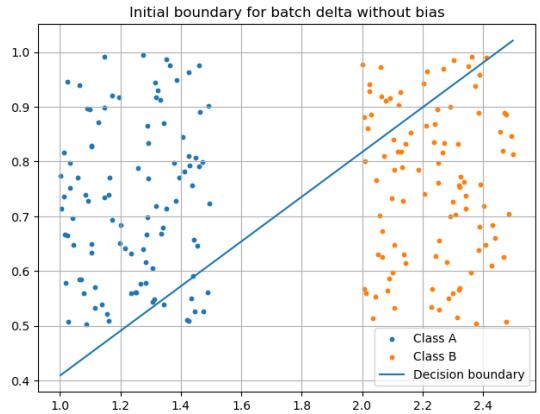


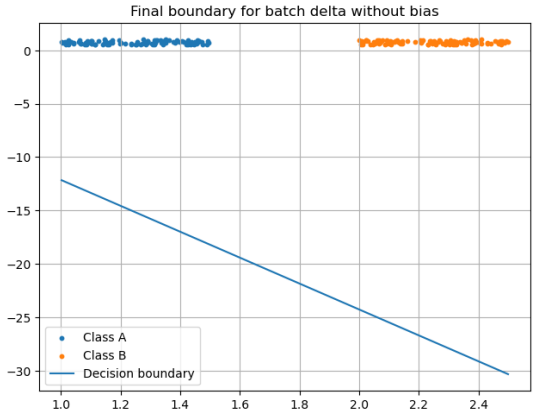
Figure 2: The MSE as a function of epochs



(a) A dataset not linearly separable with a line passing through the origin (a plane with zero bias)

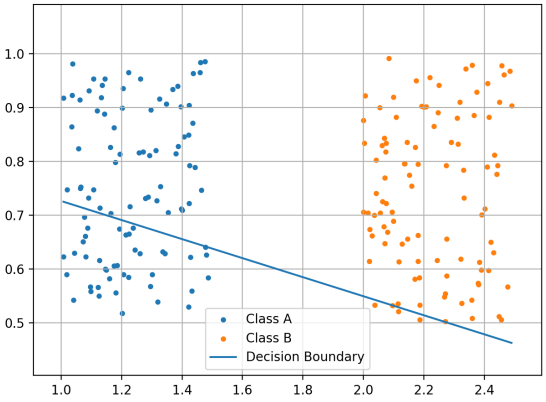


(b) The randomly initalized decision boundary of the batch delta algorithm without bias

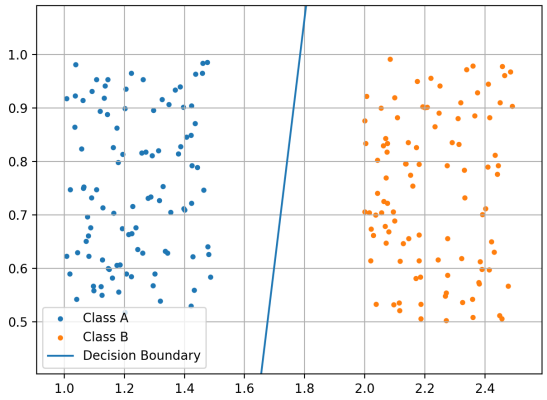


(c) The final decision boundary of the batch delta algorithm without bias after training

(



(d) The randomly initalized decision boundary of the batch delta algorithm with bias

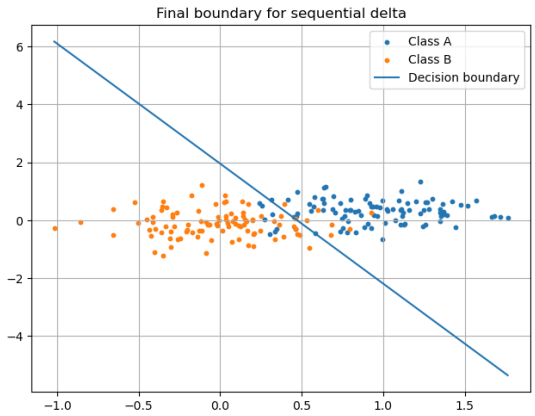


(e) The final decision boundary of the batch delta algorithm with bias

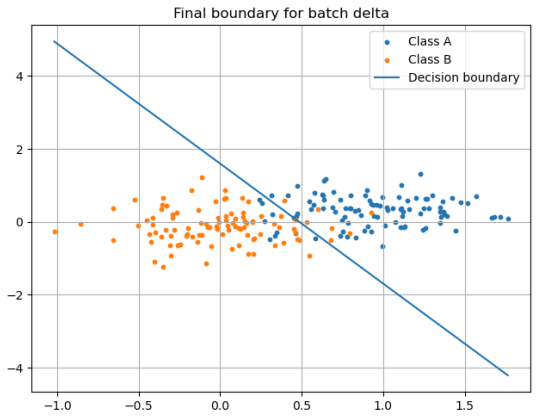
Figure 3: An example showcasing the role effect of the bias of the decision plane

3.2 Classification of data that is not linearly separable

For the case where we have data that is not linearly separable. The perceptron will not converge in this case, since it only stops running if there are no misclassified points, which there always will be because the data is not linearly separable.



(a) The final decision boundary of the sequential delta algorithm



(b) The final decision boundary of the batch delta algorithm

Figure 4: A comparison between the sequential delta algorithm and the batch delta algorithm in the case with non-linearly separable data

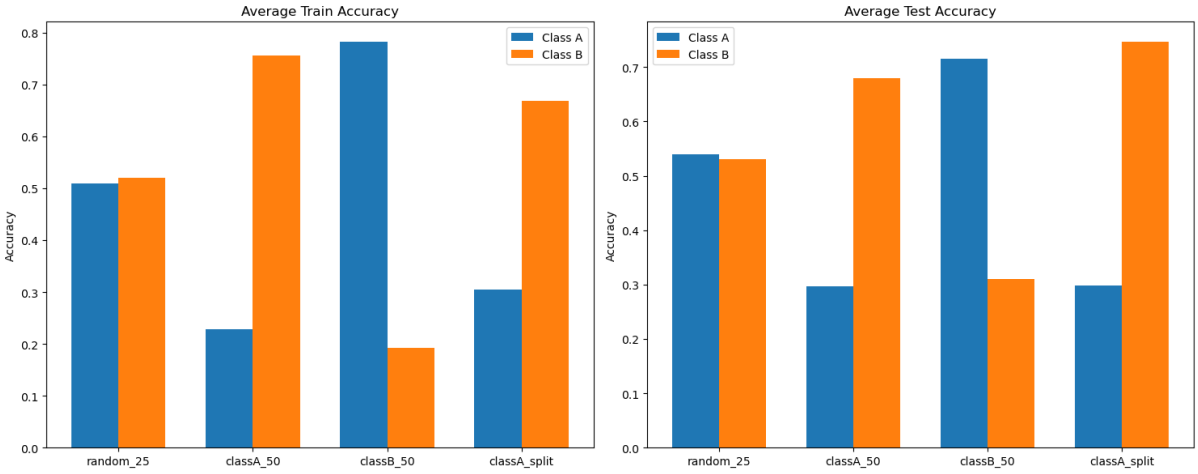
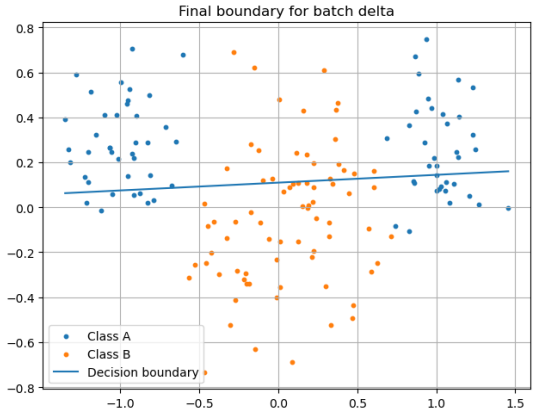
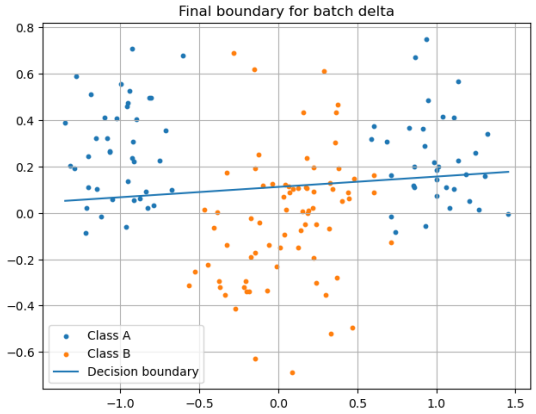


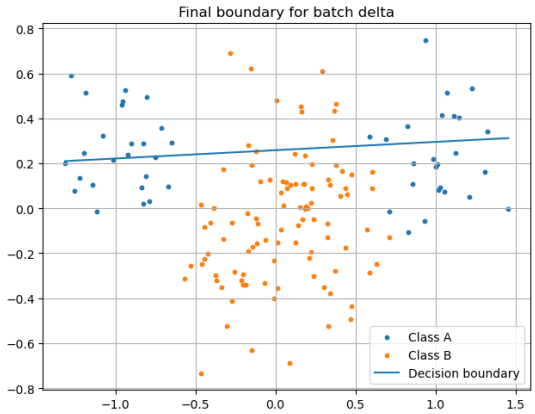
Figure 5: The accuracy compared on the training and test set for the different sub samples of the data



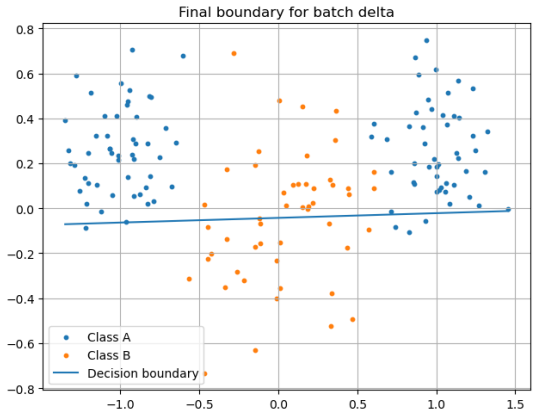
(a) Final decision boundary of delta algorithm on the entire dataset



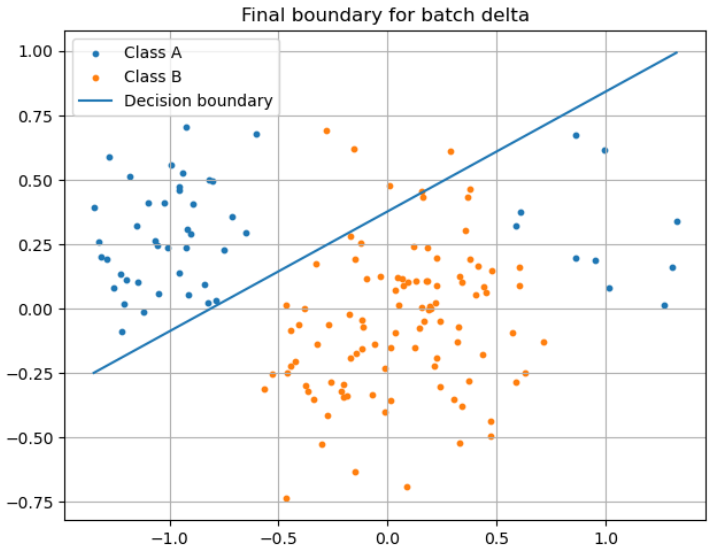
(b) Final decision boundary of delta algorithm on the subset which removed 25% of random points in both class A and B



(c) Final decision boundary of delta algorithm on the subset which removed 50 % of the points in class A



(d) Final decision boundary of delta algorithm on the subset which removed 50 % of the points in class B



(e) Final decision boundary of delta algorithm on the subset that is built from 20% of the points from the right cluster of class A and 80% of the points from the left cluster of class A, along with all class B points

Figure 6: Various configurations of the delta algorithm's final decision boundary, demonstrating different sub samples of the dataset

The generated decision boundaries of the different sub samples are different, which is to be expected since the data for them looks different. The baseline that the subsamples are compared to is figure 6a, which is a decision boundary for the whole dataset. The decision boundaries in 6a and 6b are very similar, which is reasonable since an equal proportion of both the data classes has been removed, which in most cases should lead to a similar boundary. In 6c the algorithm puts more emphasis on correctly classifying class B, since the majority of the dataset is of class B. Thus, the algorithm does not prioritize classification of class A as much, which is reflected in the accuracy plot. For plot 6d, the roles are reversed. it is of higher priority to correctly classify class A, instead of B, since most of the datapoints in the sub sample stems from class A. In the case of 6e, the cluster off class A to the right has had more points removed compared to the left cluster, which results in two clearer clusters. This leads to the algorithm finding a decision boundary between the left cluster of class A and class B and essentially ignoring the right cluster, since the price in error to do that is very low, given a large portion of the right cluster have been removed.

4 Final remarks

This lab was educational for fully understanding the perceptron and delta learning rules. Although a lot of time was also spent on plotting and not actually implementing the algorithm, it is important to practice this skill aswell. If we had to look for parts that we found confusing or not necessarily helping in understanding important concepts, we would say that the last part of the assignment was the least important part. By the time we got to part 3.2 , we had a good understanding of how the different learning rules worked, and applying them all over again did not lead to more insights in either one of the algorithms. On the other hand, one concept that was thought provoking was about what effect uneven or biased sub samples has on the final result. Different sub samples led to very different results which is reflected in the accuracy plots and the final decision boundaries for each sub samples.

Speaking more generally, since the scope of the lab was small, it lead to a deep understanding of the concepts related to this assignment.