Artificial Neural Networks and Deep Architectures, DD2437

Short report on lab assignment 1

Classification with a single-layer perceptron

Anas Fadil, Chaimae Menouar and Kevin Armbruster

September 8, 2022

1 Main objectives and scope of the assignment

Our major goals in the assignment were :

- to implement Perceptron Learning Algorithm and Delta rule.
- to compare Delta rule in sequential and batch mode.
- to investigate Delta rule without Bias.
- to investigate Delta rule with not linearly separable data.
- to investigate the effect of skewed data sets on the final accuracy.

2 Methods

We used the following environment to solve the assignment:

- Code Management in github.com
- Programming environment Google Colab and PyCharm
- Programming language Python
- Used packages were Numpy, Pandas and Matplotlib
- Perceptron learning algorithm was taken from the lecture
- Delta rule was taken from the assignment sheet

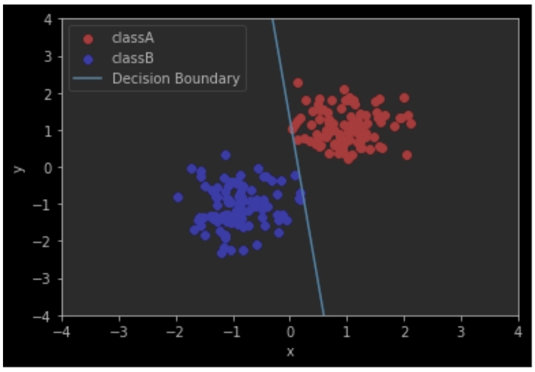
3 Results and discussion

3.1 Classification with a single-layer perceptron

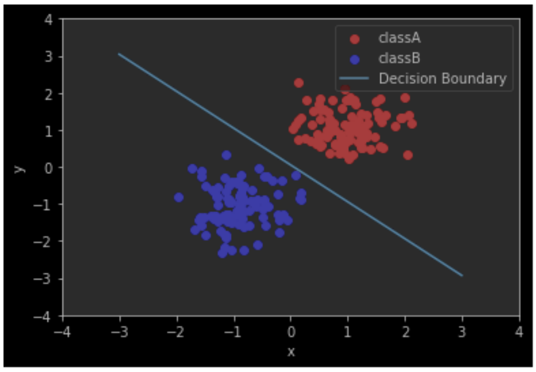
3.1.1 Perceptron learning and Delta rule

We can see the perceptron learning algorithm in figure 1. It terminates prematurely after finding a valid seperating hyperplane. Even through the perceptron learning hyperplane separates the data well, from the figure we can imagine that new data might easily be generated on the other side of the line, therefore generalizes not well on future data.

In contrast the delta rule in figure 2 continued to learn to find a better separating hyperplane, as it minimized not the misclassification but the error, which leads to a bigger margin between the class averages. From the figure we see that due to this margin it is more unlikly that new data is generated on the wrong side, therefore the learned separating hyperplane generalizes better. Sometimes it happens that the Delta rule will prioritize minimizing the error over minimizing misclassifications, which leads to small amounts of misclassifications on the training data.



Figur 1: Perceptron Learning Algorithm



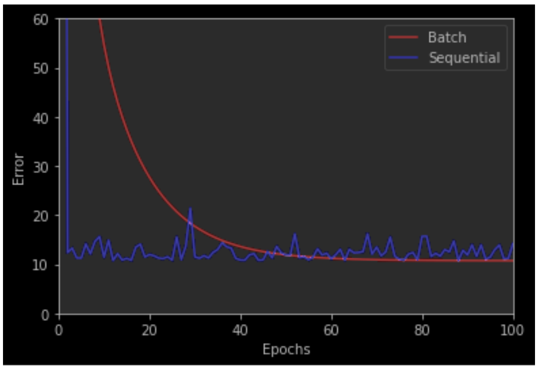
Figur 2: Delta Rule

3.1.2 Delta rule - Sequential and Batch

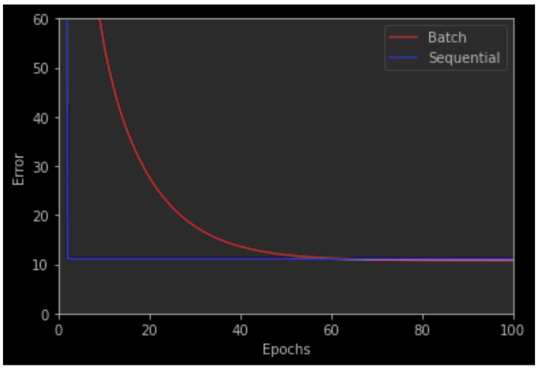
Both figures 4 and 3 illustrate the error over the epochs for both batch and sequential learning. While we keep the data in the same order for figure 4, we shuffle the data between epochs for figure 3. The batch learning curve is the same in both and displays a slower and smoother curvature than the sequential. Noticeably, in both figures the error drops much faster for the sequential learning. This is due to many more updates during each epoch and the noise of the individual gradients helps to navigate the error space.

While the sequential learning without shuffeling the data converges after 1-2 epochs, it stays worse and does no longer improve. To counter this behavior we introduced the shuffeling of the data, which introduces a noise pattern over the error across epochs. While it sometimes worsens the results, we can also see dips below the error of the batch curve.

The learning rate was 0.1 and 0.001 for sequential and batch respectively. We noticed that sequential learning allowed for much higher learning rates than batch, without overshooting. This is due to summing up the individual gradient steps in batch learning leading to bigger, overshooting steps, if not properly scaled.



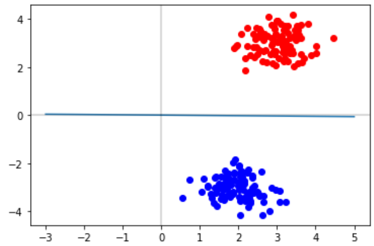
Figur 3: Convergence while shuffeling data each epoch



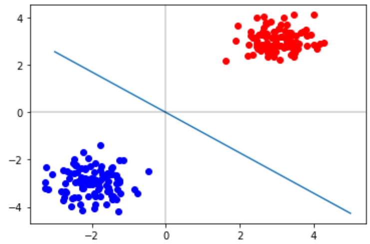
Figur 4: Convergence without shuffeling data each epoch

3.1.3 Delta rule - without the bias

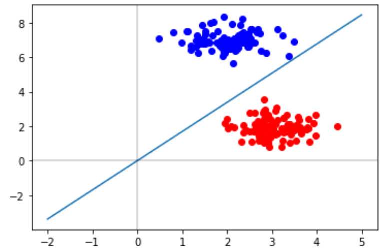
By carefully choosing the learning rate and the number of epochs, we can see that if we are in front of some simple cases (the classes are in different quarters), the delta rule in batch mode converges and classifies correctly the data samples (figure 4 : a and b).



(a)

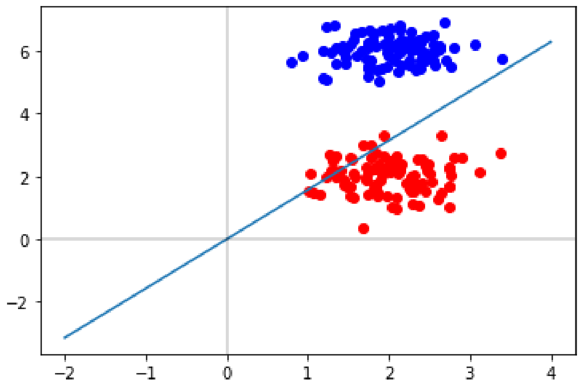


(b)



(c)

Figur 5: Examples of the convergence and the correct classification of data samples using delta rule without bias



Figur 6: Example of the misclassification of the data samples using using the delta rule without bias

However, we can also observe some convergence cases with correct classification when the classes are in the same area/quarter (figure 5 : c).
In general, we can conclude that if there is a line passing through the origin that can separate the data, the delta rule in batch mode will converge and correctly classify the data (we just need to choose a good learning rate).For instance in figure 6, this line doesn't exist, so we cannot have a correctly classified data using delta rule.

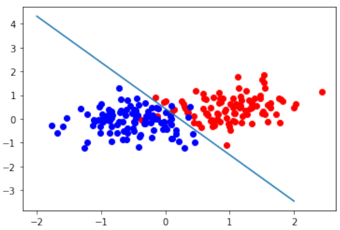
3.2 Classification of data that are not linearly separable

3.2.1 Part One

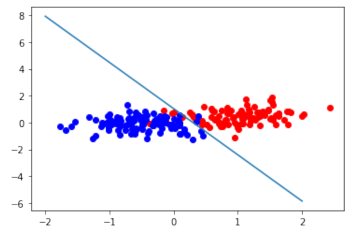
In this part we focused on creating a non-linearly separable dataset through ovelapping both class A and class B. In order to do that, we chose the starting normal distribution with the following parameters :

$$mA = [1, 0.5]\ ;\ mB = [-0.5, 0]\ ;\ sigmaA = 0.5\ ;\ sigmaB = 0.5$$

We then applied both perceptron learning and the delta rule with sequential learning mode on this dataset with a learning rate $learning\_rate = 0.005$ and a number of epochs $n = 40$. The final classification is shown in the following figures :



Figur 7: Perceptron learning classification

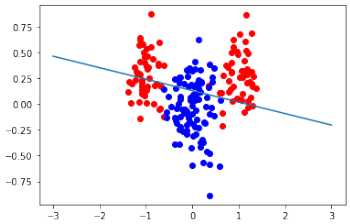


Figur 8: Delta Online learning classification

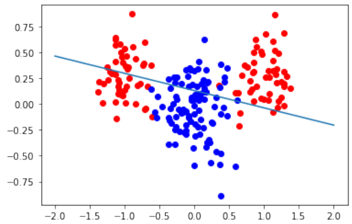
Both learning rules did fairly good classification of the two datasets with the Perceptron learning rule reaching 97.5% accuracy and the Delta Online learning rule reaching 95.5% accuracy. As expected, the accuracy did not reach 100% as the data is not linearly separable. The resulting accuracies were also in par with our intuition as both classes are only overlapping by a little which explains the very high accuracies. We could expect the accuracy to drop a lot with more complicated non linearly separable datasets.

3.2.2 Part Two

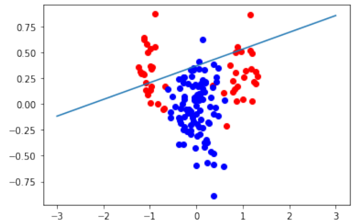
Using the given normal distribution for the classes we obtain the following decision boundaries :



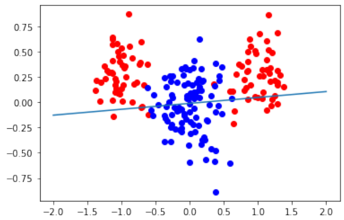
Figur 9: No sampling



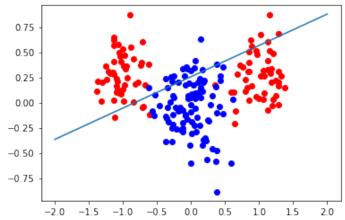
Figur 10: 25% sampling for each class



Figur 11: 50% A sampling



Figur 12: 50 % B sampling



Figur 13: Sampling right side A data

After applying the perceptron learning rule for each subsampling scenario for a few epochs we collected the accuracy data in a table as shown :

	Global Accuracy	Class A Accuracy	Class B Accuracy
No subsampling	0.70	0.6	0.8
25% global subsampling	0.69	0.68	0.69
50% A subsampling	0.67	0.37	0.98
50% B subsampling	0.7	0.89	0.51
A conditionned subsampling	0.67	0.51	0.83

Figur 14: Accuracies for subsampling scenarios

We can observe that reducing the data samples from one class or another shifts the decision boundary to classify the most present class, hence why we see a jump in class A accuracy when removing 50% of class B data samples or the opposite when removing 50% of class A data samples. However, reducing an equal amount of data on both sides does not chance the accuracy as the data size balance is not shifted towards any of the classes.
As noted in the lab presentation, sub-sampling offers unseen test set for the model, it can help measure how well our model can generalise. However in the case of uneven class representation we could sub-sample too much from a class and remove information which would hinder the

model's generalisation. When sampling , we should then make sure that both the sampled and original data follow the same pattern so that our trained model isn't wrongly biased.

4 Final remarks

The effect of randomness during interpretation and reporting lead sometimes to weird behavior that could not be replicated. Especially for the delta rule sequential learning with shuffling of the data between epochs. As shortly discussed in the lecture this seems to be common practice. Since the error was never consistent over multiple epochs traditional stopping rules such as the same error over multiple epochs did not work. Would a reasonable approach be to keep a copy of the best weight matrix while learning?

The exercises were helpful for further understanding the topics and actually practicing what we learned during the lectures. It was also beneficial when comparing our intuition with the actual lab results like the delta rule classification without bias that we predicted or also the shifting of the decision boundary when sub-sampling from one class or another.