

TDT4171 - Assignment 4

Martin Bjerke

April 9, 2018

I Gradient Descent

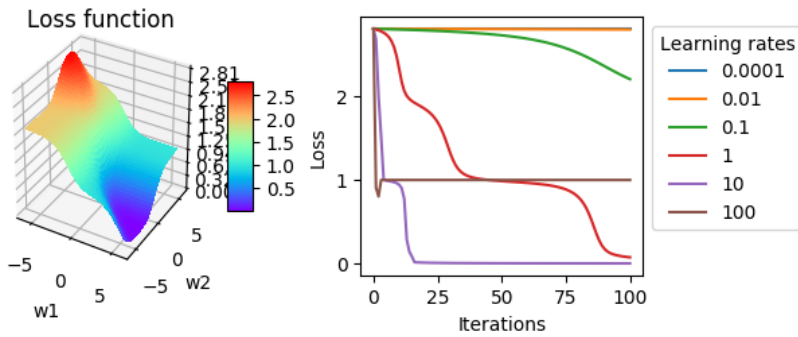


Figure 1: Loss function & learningrate

1. From figure 1 its clear that the minimum of the loss function(within the range of $w_i \in (6, -6)$) is $(w_1, w_2) = (6, -3)$.
2. Since the derivative of $\frac{\partial L_{simple}(w)}{\partial w_i}$ will be indifferent of i I'll show the calculations of w_i :

$$\begin{aligned} \frac{\partial L_{simple}(w)}{\partial w_i} &= \frac{[\sigma(w, [1, 0]) - 1]^2 + [\sigma(w, [0, 1])]^2 + [\sigma(w, [1, 1]) - 1]^2}{\partial w_i} \\ &= \frac{[\sigma(w, [1, 0]) - 1]^2}{\partial w_i} + \frac{[\sigma(w, [0, 1])]^2}{\partial w_i} + \frac{[\sigma(w, [1, 1]) - 1]^2}{\partial w_i} \\ &= \frac{1}{2} [\sigma(w, [1, 0]) - 1] \frac{[\sigma(w, [1, 0]) - 1]}{\partial w_i} + \frac{1}{2} [\sigma(w, [0, 1])] \frac{[\sigma(w, [0, 1])]}{\partial w_i} \\ &\quad + \frac{1}{2} [\sigma(w, [1, 1]) - 1] \frac{[\sigma(w, [1, 1]) - 1]}{\partial w_i} \end{aligned}$$

Next I find the derivative of $\sigma(w, x)$:

$$\begin{aligned}
\frac{\partial \sigma(w, x)}{\partial w_i} &= \frac{\partial (1 + e^{-w^T x})^{-1}}{\partial w_i} \\
&= - \left(\frac{1}{1 + e^{-w^T x}} \right)^{-2} \cdot \frac{\partial (1 + e^{-w^T x})}{\partial w_i} \\
&= - \left(\frac{1}{1 + e^{-w^T x}} \right)^{-2} \cdot (-x_i e^{-w^T x}) \\
&= x_i \frac{e^{-w^T x}}{1 + e^{-w^T x}} \cdot \frac{1}{1 + e^{-w^T x}} \\
&= x_i \frac{1 + e^{-w^T x} - 1}{1 + e^{-w^T x}} \cdot \frac{1}{1 + e^{-w^T x}} \\
&= x_i \left[\frac{1 + e^{-w^T x}}{1 + e^{-w^T x}} - \frac{1}{1 + e^{-w^T x}} \right] \cdot \frac{1}{1 + e^{-w^T x}} \\
&= x_i \left[1 - \frac{1}{1 + e^{-w^T x}} \right] \cdot \frac{1}{1 + e^{-w^T x}} = x_i \sigma(w, x) [1 - \sigma(w, x)]
\end{aligned}$$

Inserting this:

$$\begin{aligned}
\frac{\partial L_{simple}(w)}{\partial w_i} &= \frac{x_i}{2} [\sigma(w, [1, 0]) - 1] \sigma(w, [1, 0]) [1 - \sigma(w, [1, 0])] \\
&\quad + \frac{x_i}{2} [\sigma(w, [0, 1])] \sigma(w, [0, 1]) [1 - \sigma(w, [0, 1])] \\
&\quad + \frac{x_i}{2} [\sigma(w, [1, 1]) - 1] \sigma(w, [1, 1]) [1 - \sigma(w, [1, 1])]
\end{aligned}$$

This derivative results in the following derivative of the loss function:

$$\begin{aligned}
\nabla_w L_{simple}(w) &= \left[\frac{\partial L_{simple}(w)}{\partial w_1}, \frac{\partial L_{simple}(w)}{\partial w_1} \right] \\
\frac{\partial L_{simple}(w)}{\partial w_1} &= -\frac{1}{2} \left(\sigma(w, [1, 0]) [\sigma(w, [1, 0]) - 1]^2 + \sigma(w, [1, 1]) [\sigma(w, [1, 1]) - 1]^2 \right) \\
\frac{\partial L_{simple}(w)}{\partial w_1} &= -\frac{1}{2} \left(\sigma(w, [0, 1])^2 [\sigma(w, [0, 1])] + \sigma(w, [1, 1]) [\sigma(w, [1, 1]) - 1]^2 \right)
\end{aligned}$$

3. From figure 1 you can see that with a learningrate of 1 and 10 minimizes L_{simple} after 100 and 15 iterations each, while the smaller iterations barely starts to minimize after 100 iterations.

II Perceptron

1. Continuing the calculations from above:

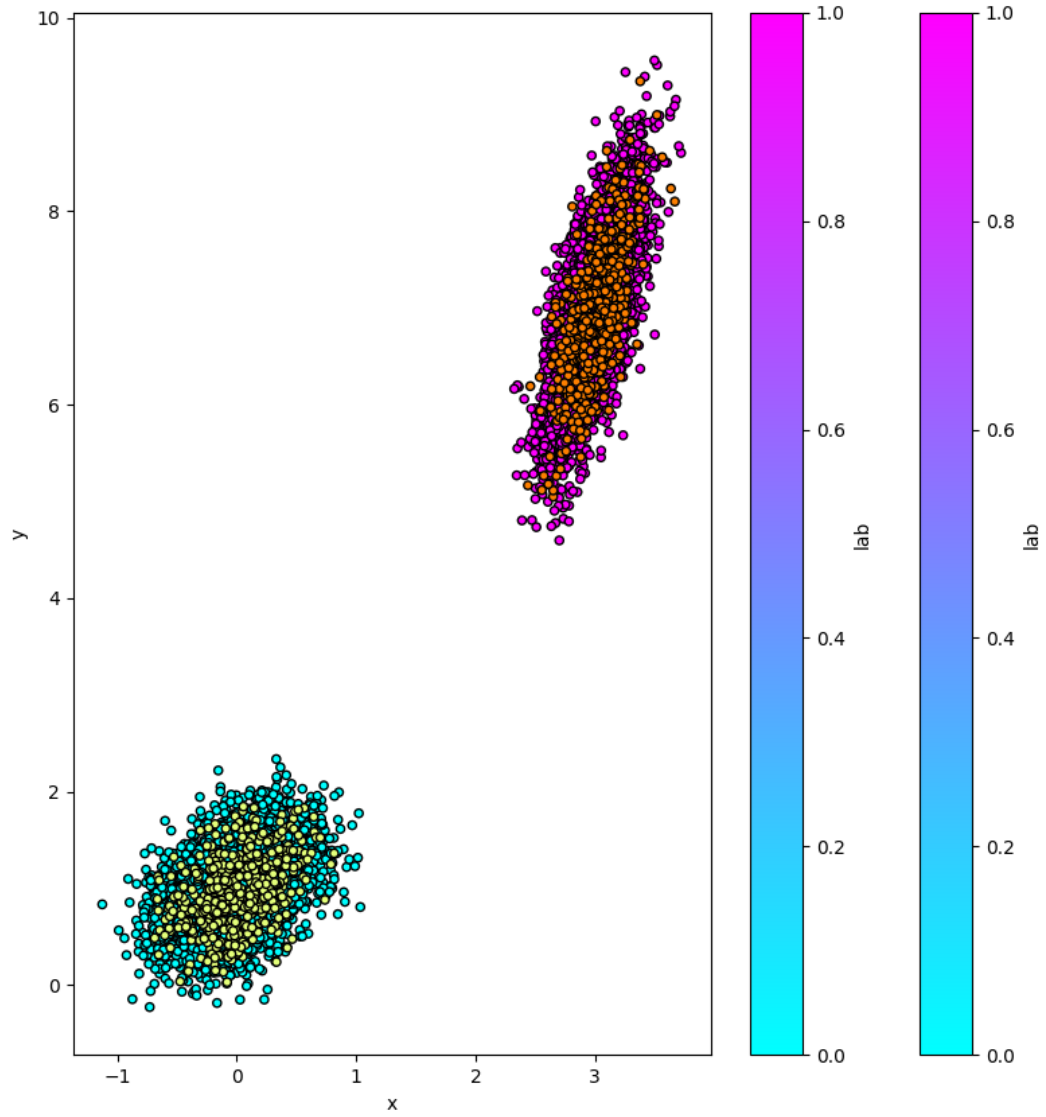
$$\frac{\partial L_n(w, x_n, y_n)}{\partial w_i} = x_i \cdot [\sigma(w, x_n) - y_n] [1 - \sigma(w, x_n)] \sigma(w, x_n)$$

2. The table below show the avgarge when the perceptron was trained 10 times with 200 iterations, time is given in seconds and errors are given in percent where 0% meaning everything was classified correctly.

Data set	Batch		Stochastic	
	Training time(s)	Avg. error(%)	Training time(s)	Avg. error(%)
Big_nonsep	57.86	19.3	0.01	21.4
Big_sep	59.31	00.9	0.02	1.5
Small_nonsep	12.07	20.6	0.01	21.3
Small_sep	11.89	00.8	0.01	2.6

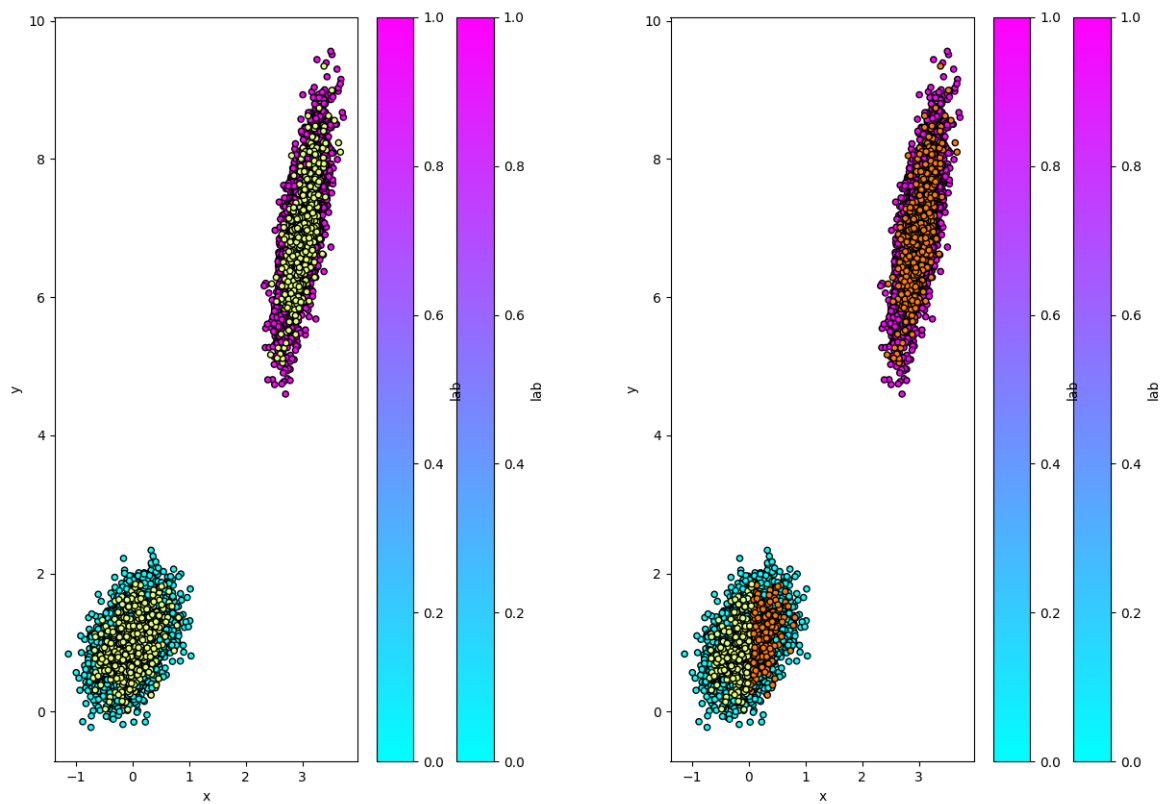
From the table above it is clear that batch gradient descent training is slower, but slightly more accurate on these data sets. This result concurs with the algorithm as batch gradient descent calculates $T \cdot (d) \cdot |D_{train}|$ gradients while stochastic only calculates $T \cdot (d)$. By looking further on the number of gradients calculated by the methods the correlation between the times also becomes apparent; Stochastic doesn't depend on the size of the data set and calculates $200 \cdot 2 = 400$ gradients, while batch calculates $400 \cdot 11000$ and $400 \cdot 2200$ gradients on 200 iterations. As the big data sets are 5 times as large as the small, the difference in training time correlates with this by being 5 times as large. As one would believe the non separable data sets are more inaccurate and the separability of the data set has no effect on the time used by the different methods since the code does not take this into consideration when calculating the gradients.

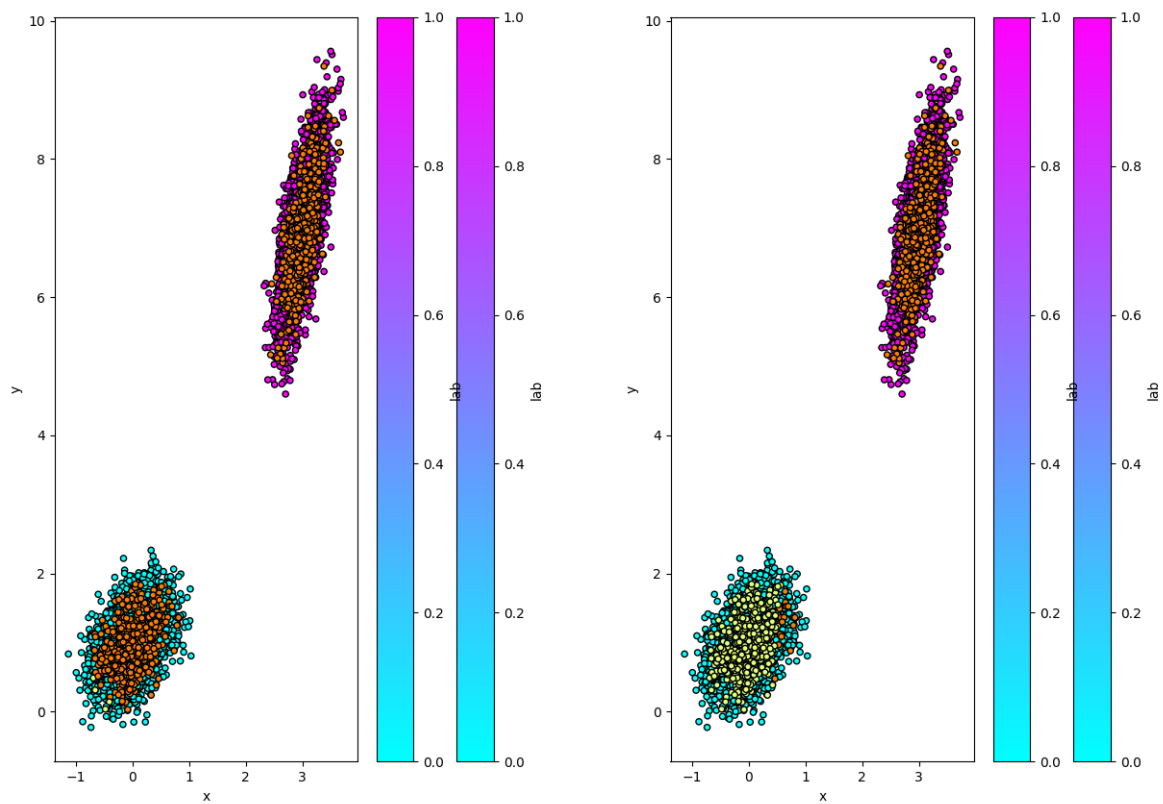
3. Training the perceptron using stochastic gradients descent on the the small separatable data set and plotting the results gives the following plot.

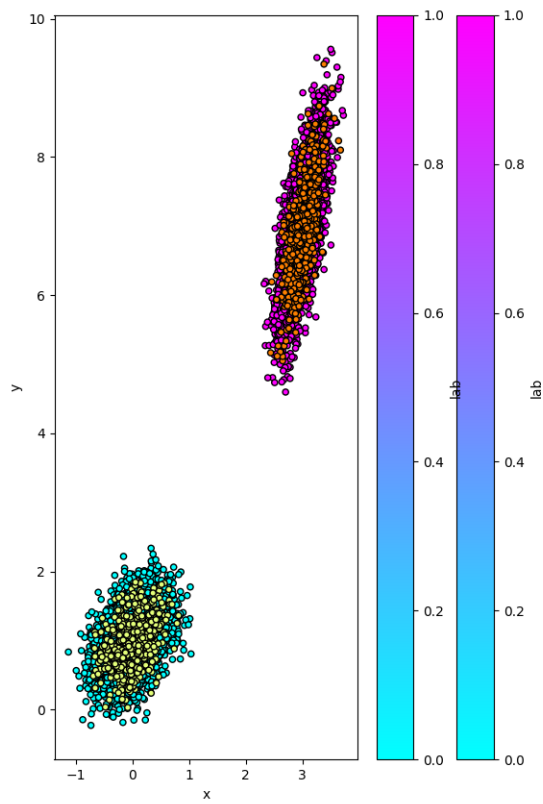


4. The same dataset and algorithm as above gives the following results(plots are placed after the discussion):

Iterations	Error(%)	Time
10	50.0	0.008
20	21.8	0.009
50	49.6	0.009
100	1.8	0.010
500	0.0	0.019







Choose one dataset and one training algorithm, run with 10, 20, 50, 100 and 500 iterations, keeping track of time and avg error, plot results and discuss your findings