Neural Networks – Tutorial

(Course: Introduction to Data Science)

Tirtharaj Dash

BITS Pilani, K.K. Birla Goa Campus

tirtharaj@goa.bits-pilani.ac.in

November 7, 2020

NN Tutorial I

In the lecture, we defined a single-layered network with logistic function to motivate the idea of a neural network based on logistic regression. We also called this single layered network as perceptron.

For this tutorial, we will stick to the basic Rosenblatt's perceptron model (1958) that is defined as a thresholded model for binary classification:

$$f(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \boldsymbol{w} \cdot \boldsymbol{x} > 0 \\ 0 & \text{otherwise} \end{cases}$$

NN Tutorial II

1. Show that a perceptron cannot solve the 2-bit Exclusive-OR (XOR) problem.

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

Answer. A perceptron is required to satisfy the following inequalities:

$$0 \times w_1 + 0 \times w_2 + w_0 \leq 0 \therefore w_0 \leq 0$$
$$0 \times w_1 + 1 \times w_2 + w_0 > 0 \therefore w_0 > -w_2$$
$$1 \times w_1 + 0 \times w_2 + w_0 > 0 \therefore w_0 > -w_1$$
$$1 \times w_1 + 1 \times w_2 + w_0 \leq 0 \therefore w_0 \leq -(w_1 + w_2)$$

The above sets of inequalities are self-contradictory for any given set of weights $\{w_0, w_1, w_2\}$.

NN Tutorial III

2. Why is the convergence of a feed-forward neural network dependent on the initial configuration (weight set) of the net? Justify with regard to the property of the objective function that is minimized during the learning process.

   Answer. The loss or cost function of a neural network is not convex. For a non-convex function the convergence depends on the initial configuration of the input parameters. The initial configuration refers to the initial values of the function paramters, in this case weights.

NN Tutorial IV

3. Consider a deep fully connected neural network with 10 hidden layers, with architecture $m : n : 2n : \ldots : 10n : p$. What is the number of synaptic connections present in this net? Note that the hidden layers and the output layer involves additional bias.

   Answer. Total number of connections $=$
   $(m+1)n+(n+1)2n+(2n+1)3n+\ldots+(9n+1)10n+(10n+1)p$
   $= 330n^2 + 55n + mn + 10np + p.$

   Note that the '+1' in the intermediate terms is for the additional bias.

NN Tutorial V

4. Consider the 2-bit digital gate operation as shown in the following table.

x_1	x_2	y
–1	–1	1
–1	1	–1
1	–1	–1
1	1	–1

   (a) Denote the synaptic weights as w_1, w_2, and bias as w_0. Model a perceptron using the above training data.

   (b) Set the initial parameters to 0 and the step size (α) to 1. Draw the decision line of the perceptron after one epoch [1].
   Test your perceptron (after 1 epoch) for the first pattern $(-1, -1)$. Is it correctly classified?

NN Tutorial VI

(c) Now, retrain your perceptron for one more epoch and obtain the new parameters. Draw the decision line of the perceptron now.
Test your perceptron again for the first pattern $(-1, -1)$. Is it correctly classified?

(d) If you saw two different answers for the same test pattern $(-1, -1)$, provide suitable reason behind it.

Answer.

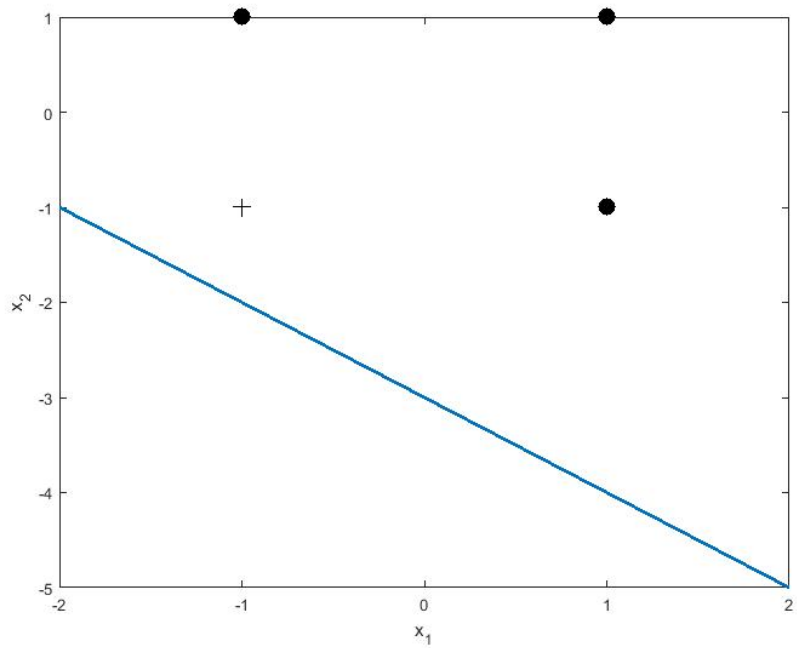
(a) The perceptron output is $\hat{y} = f(\boldsymbol{x}; \boldsymbol{w}) = \text{sign}(w_1 x_1 + w_2 x_2 + w_0)$. The perceptron learning rule is

$$w(t) = w(t-1) + \alpha \{y(t) - \hat{y}(t)\} x(t)$$

(b) If we apply the perceptron learning rule, for the first epoch, we get

$(\boldsymbol{x}; y)$	$(w_1, w_2, w_0);\ \boldsymbol{w}(0) = 0$
$((-1,-1);1)$	$(-1,-1,1)$
$((-1,1);-1)$	$(1,-3,-1)$
$((1,-1);-1)$	$(-1,-1,-3)$
$((1,1);-1)$	$(-1,-1,-3)$ (No change)

The decision line equation can be obtained by setting $\boldsymbol{w}^\top \boldsymbol{x} = 0$. So, we get $x_2 = -\frac{w_1}{w_2} x_1 - \frac{w_0}{w_2}$. Here is the decision line for the learned perceptron after one epoch.



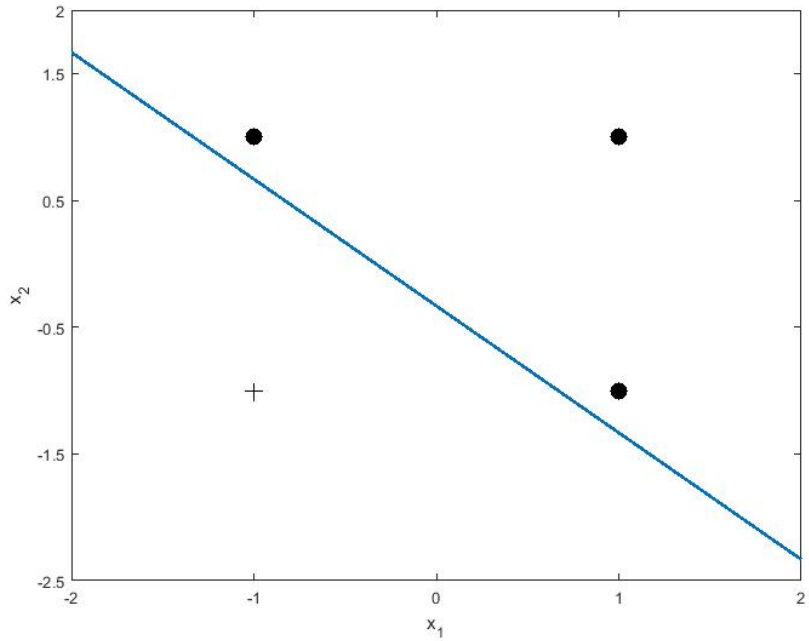
The output for the test pattern $(-1, -1)$, $\hat{y} = f(\boldsymbol{x}; \boldsymbol{w}) = -1$. The pattern is misclassified.

- Training the perceptron for one more epoch, we get

$(\boldsymbol{x}; y)$	(w_1, w_2, w_0); $\boldsymbol{w}(0) = 0$
$((-1, -1); 1)$	$(-3, -3, -1)$
$((-1, 1); -1)$	$(-3, -3, -1)$ (No change)
$((1, -1); -1)$	$(-3, -3, -1)$ (No change)
$((1, 1); -1)$	$(-3, -3, -1)$ (No change)

NN Tutorial IX

The new decision boundary is given as



The output for the test pattern $(-1, -1)$,
$\hat{y} = f(\boldsymbol{x}; \boldsymbol{w}) = 1$. The pattern is now correctly classified.

- Number of epochs were not sufficient to learn all the representation of the four patterns. After the first epoch, the perceptron had forgotten the representation for the pattern $(-1, -1)$, which it could recover after the second epoch (see the first instance in the epoch2 table).

NN Tutorial XI

5. The following data points represented as $(x_1,\ x_2)$ are provided to you. Data points are: Positive: (1, 3) (2, 2) (3, 1) and Negative: (2, 4) (3, 3) (4, 1). Data points are classified as either $+1$ or -1. An unknown point is located at (1, 4). The goal is to learn a strict-threshold perceptron $(f(\mathrm{x}, \mathrm{w}) > 0?\ +1\ :\ -1)$ and classify the unknown data point by the learned perceptron. For this, answer the following questions.

NN Tutorial XII

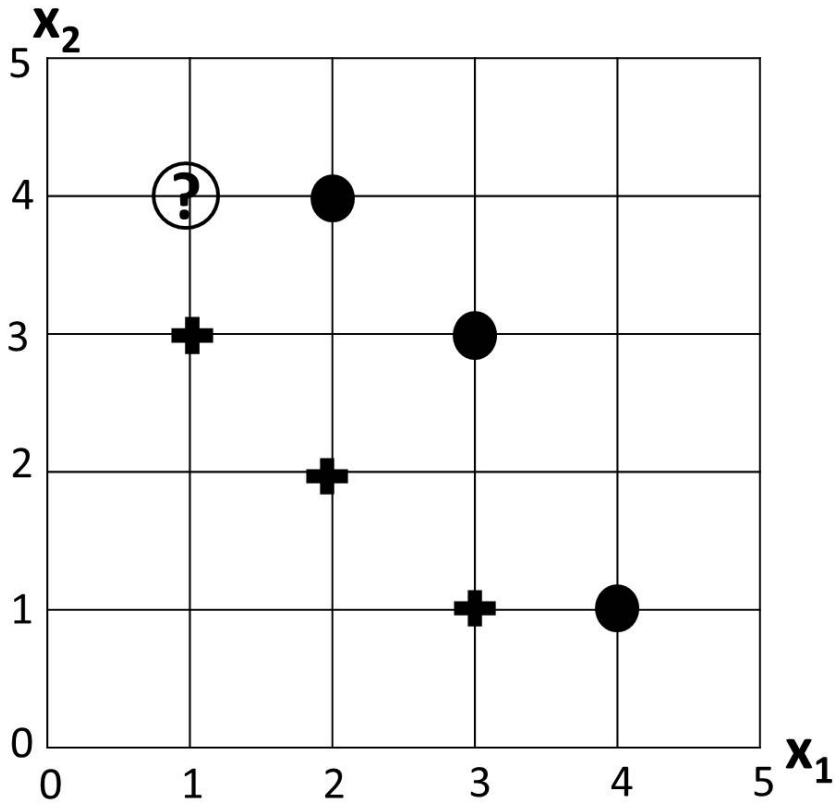


Figure: Scatter plot for the given data points

NN Tutorial XIII

(a) Assume that the points are introduced to perceptron in the order given above, that is, first all positive in the sequence followed by all negative in the sequence. Simulate one iteration of the perceptron algorithm with a learning rate (η) of 0.5 and an initial weight vector of $(w_1, w_2, w_0) = (3, 3, 30)$. Fill the following table for your weight vector in the same format (Finally, all 1+6 rows to be filled. Show the calculations for your answers):

Data point (x_1, x_2)	w_1	w_2	w_0 (bias)
	3	3	30
(1, 3), +1	?	?	?
(2, 2), +1	?	?	?
(3, 1), +1	?	?	?
(2, 4), –1	?	?	?
(3, 3), –1	?	?	?
(4, 1), –1	?	?	?

NN Tutorial XIV

(b) What is the equation of your perceptron obtained in part (a)?

(c) Is your perceptron a correct linear separator of the given data?

(d) What is the class label returned by your learned perceptron for the unknown data point (1, 4)? What is the distance of this point from the learned perceptron?

Answer. The answers are as follows:

(a) It is a strict-threshold perceptron. The loss function is:

$$L(y, \hat{y}) = \frac{1}{2}(y - \hat{y})^2$$

where, $\hat{y} = f(\mathrm{w}, \mathrm{x}) = w_1 x_1 + w_2 x_2 + w0$.
The update equation is:

$$\begin{aligned} \mathrm{w} &\leftarrow \mathrm{w} - \eta \times \frac{\partial L}{\partial \mathrm{w}} \\ &\leftarrow \mathrm{w} + \eta \times (y - \hat{y}) \cdot [x_1, x_2, +1] \end{aligned}$$

The weight vector after introduction of each example is:

w_1	w_2	w_0
3	3	30
3	3	30
3	3	30
3	3	30
1	–1	29
–2	–4	28
–6	–5	27

(b) Equation of perceptron is $w_1x_1 + w_2x_2 + w0 = 0$ (Put the values of these parameters)

(c) Yes. See the plot below

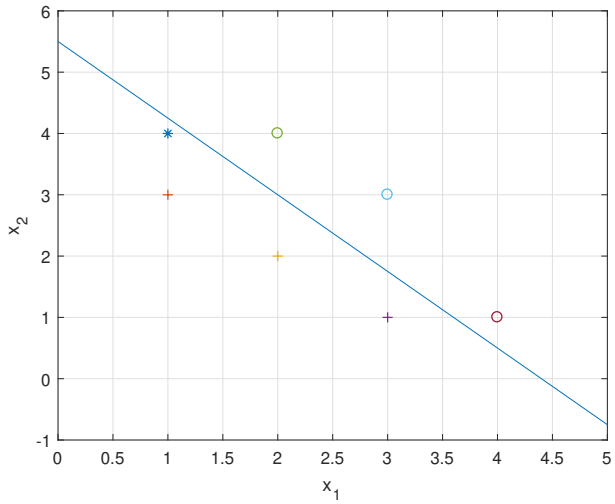


Figure: Perceptron for the given data

- $\hat{y} = +1$. This is correct from the visuals. Thd distance is $d(perceptron, (1, 4)) = \frac{|w_1 \times 1 + w_2 \times 4 + w_0|}{\sqrt{(-6)^2 + (-5)^2}} = 0.1280.$

NN Tutorial XVIII

6. Consider a neural network with three layers including an input layer. The first (input) layer has four inputs x_1, x_2, x_3, and x_4. The second layer has six hidden units corresponding to all pairwise multiplications. The output unit $\hat{y}$ simply adds the values in the six hidden units. Let L be the loss at the output node. Suppose that you know that $\frac{\partial L}{\partial \hat{y}} = 2$, and $x_1 = 1$, $x_2 = 2$, $x_3 = 3$, and $x_4 = 4$.
   a. Compute $\frac{\partial L}{\partial x_i}$ for all i.
   b. Now change the $\hat{y}$ to $\max(x_1x_2, x_1x_3, x_1x_4, x_2x_3, x_2x_4, x_3x_4)$. Compute $\frac{\partial L}{\partial x_i}$ for all i in this new setting of $\hat{y}$.

NN Tutorial XIX

Answer.

(a) $\hat{y} = x_1x_2 + x_1x_3 + x_1x_4 + x_2x_3 + x_2x_4 + x_3x_4$. Now, $\frac{\partial L}{\partial x_1} = \frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial x_1}$. So, $\frac{\partial L}{\partial x_1} = 2(x_2 + x_3 + x_4)$. Similarly, other gradients can also be computed.

(b) In such a case, one just back propagates along the maximum of the nodes in the last hidden layer. Here, the maximum product is $x_3x_4 = 12$. Therefore, the partial derivatives of the loss with respect to x_1 and x_2 are 0. The partial derivative with respect to x_3 is $2x_4 = 8$, and the partial derivative with respect to x_4 is $2x_3 = 6$.

[1] An epoch is defined as an iteration over all the training instances.