HOMEWORK #2: AIT 736 Summer 2022

Applied MACHINE LEARNING

DUE: July 17, 2022 (30 points)

Graphical user interface, text, application, email

Description automatically generated

Solution The output of the perceptron is o = sgn (w0 + w1 x1 + w2 x2)

The equation of the decision surface (the line) is w0 + w1 x1 + w2 x2= 0

We know the coordinates of 2 points of this line: A=(-1,0) and B=(0,2).

Therefore, the equation of the line is

A picture containing text

Description automatically generated

So,

So 2, 2, -1 are possible values for the weights w0, w1, and w2, respectively. To check if their signs are correct, consider a point on one side of the line, for instance the origin O=(0,0). The output of the perceptron for this point has to be negative, but the output of the perceptron using the candidate weights is positive. Therefore, we need to negate the previous values and conclude that

2. Implement the perceptron learning algorithm (PLA) and linear regression (pseudoinverse) discussed in class. Please separate the problem in 3 main steps, for each step indicate the command and describe what it executes.

1) Generation of the data and labeling. [10 points]

2) Apply and describe PLA. [30 points]

3) Apply and describe Linear regression by computing the pseudo inverse. [30 points]

Requirement:

1. You are required not to use the existing classifier and regression function in library.

2. Please submit your code in whatever language you prefer.

Diagram

Description automatically generated

In machine learning, the perceptron is a supervised learning algorithm used as a binary classifier, which is used to identify whether a input data belongs to a specific group (class) or not.

In our PLA, we have a simple perceptron

F(s)={1 if s>0

−1if otherwise.

X0 and X1 are inputs to the perceptron.

100 Different values of X0 and X1 are taken as train data set as random values between 100 and -100.

Weights w0 and w1 are initialized as two random values.

The training outcome has been calculated as the isPositiveTrainingY method which takes random X1 and X2 as input , add them ,if it’s a positive value, return 1 , otherwise -1. ion

So, our training dataset contains two random integers between 100 and -100 as X1 and X0 , and a Y value of either +1 or -1.

The X0 and X1 values are passed through the perceptron with the random initialized weight values.

We have defined an activation function which takes x1, x0 , and the weights and bias as input. It applies the weights to the input value and add the bias to get the guessed value.

If the guessed value is more than 0 , it returns +1 , else -1.

The main goal of a *perceptron* is to make accurate classifications. To train a model to do this, perceptron weights must be optimizing for any specific classification task at hand.The best weight values can be chosen by training a perceptron on labeled training data that assigns an appropriate label to each data sample (feature). Now the objective is the to optimize the weights so that the guessed value is closed to the actual y we already have in training data. In order to do that, for each of the training dataset, we call the activation function to get the guessed value and calculate the error as the training y and the guessed value.

Here are the possible combinations of error :

|  |  |  |
| --- | --- | --- |
| Y | Guessedvalue | Error |
| +1 | +1 | 0 |
| +1 | -1 | 2 |
| -1 | -1 | 0 |
| -1 | +1 | -2 |

So, the only 3 possible outcomes of error might be 0,-2,and 2.

Output of the program:

Random Training Data:

Chart, scatter chart

Description automatically generated

Errors after the iterations

Chart, histogram

Description automatically generated

Chart, line chart

Description automatically generated

Chart, box and whisker chart

Description automatically generated

*Iteration: 1 : [errorcount: 9 , weights:[0.81105727 1.31287152], bias:-6.0 ]*

*Iteration: 2 : [errorcount: 9 , weights:[1.78105727 1.62287152], bias:0.0 ]*

*Iteration: 3 : [errorcount: 5 , weights:[1.93105727 1.77287152], bias:-2.0 ]*

*Iteration: 4 : [errorcount: 5 , weights:[2.08105727 1.92287152], bias:-4.0 ]*

*Iteration: 5 : [errorcount: 5 , weights:[2.23105727 2.07287152], bias:-6.0 ]*

*Iteration: 6 : [errorcount: 0 , weights:[2.23105727 2.07287152], bias:-6.0 ]*

*Iteration: 7 : [errorcount: 0 , weights:[2.23105727 2.07287152], bias:-6.0 ]*

*Iteration: 8 : [errorcount: 0 , weights:[2.23105727 2.07287152], bias:-6.0 ]*

*Iteration: 9 : [errorcount: 0 , weights:[2.23105727 2.07287152], bias:-6.0 ]*

*Iteration: 10 : [errorcount: 0 , weights:[2.23105727 2.07287152], bias:-6.0 ]*

*Iteration: 11 : [errorcount: 0 , weights:[2.23105727 2.07287152], bias:-6.0 ]*

*Iteration: 12 : [errorcount: 0 , weights:[2.23105727 2.07287152], bias:-6.0 ]*

*Iteration: 13 : [errorcount: 0 , weights:[2.23105727 2.07287152], bias:-6.0 ]*

*Iteration: 14 : [errorcount: 0 , weights:[2.23105727 2.07287152], bias:-6.0 ]*

*Iteration: 15 : [errorcount: 0 , weights:[2.23105727 2.07287152], bias:-6.0 ]*

*Iteration: 16 : [errorcount: 0 , weights:[2.23105727 2.07287152], bias:-6.0 ]*

*Iteration: 17 : [errorcount: 0 , weights:[2.23105727 2.07287152], bias:-6.0 ]*

*Iteration: 18 : [errorcount: 0 , weights:[2.23105727 2.07287152], bias:-6.0 ]*

*Iteration: 19 : [errorcount: 0 , weights:[2.23105727 2.07287152], bias:-6.0 ]*

*Iteration: 20 : [errorcount: 0 , weights:[2.23105727 2.07287152], bias:-6.0 ]*

*Iteration: 21 : [errorcount: 0 , weights:[2.23105727 2.07287152], bias:-6.0 ]*

*Iteration: 22 : [errorcount: 0 , weights:[2.23105727 2.07287152], bias:-6.0 ]*

*Iteration: 23 : [errorcount: 0 , weights:[2.23105727 2.07287152], bias:-6.0 ]*

*Iteration: 24 : [errorcount: 0 , weights:[2.23105727 2.07287152], bias:-6.0 ]*

*Iteration: 25 : [errorcount: 0 , weights:[2.23105727 2.07287152], bias:-6.0 ]*

*Iteration: 26 : [errorcount: 0 , weights:[2.23105727 2.07287152], bias:-6.0 ]*

*Iteration: 27 : [errorcount: 0 , weights:[2.23105727 2.07287152], bias:-6.0 ]*

*Iteration: 28 : [errorcount: 0 , weights:[2.23105727 2.07287152], bias:-6.0 ]*

*Iteration: 29 : [errorcount: 0 , weights:[2.23105727 2.07287152], bias:-6.0 ]*

*Iteration: 30 : [errorcount: 0 , weights:[2.23105727 2.07287152], bias:-6.0 ]*

*2 has occurred 15 times*

*-2 has occurred 18 times*

*0 has occurred 2967 times*

*[0, 2.894535398006739] -6.0 [2.689307926143262, 0] [2.23105727 2.07287152]*

*(b)* **Apply and describe Linear regression by computing the pseudo inverse. [30 points]**

A common use of the pseudoinverse is to compute a "best fit" ([least squares](https://en.wikipedia.org/wiki/Ordinary_least_squares)) solution to a [system of linear equations](https://en.wikipedia.org/wiki/System_of_linear_equations) that lacks a solution.

Chart, waterfall chart

Description automatically generated

Ax = b

A = U D VT

x =AT b

AT = V (D) -1 U T

Where all the below are matrices

U – Left Singular Vector , square matrix

D - same dimension as A

V – Right Singular Vector , square matrix

We are providing two solutions – one using a random 3x2 matrix and other using a housing dataset

Solution 1 – random 3x2 matrix .

we have A as 3x2 , D will be 3x2 , hence U should be 3x 3 and Vt is 2x2

3x 2 = (3x3) x ( 3x 2 ) x ( 2x 2)

Hence

At shape should be 2x3

Ut shape should be 3x3

D inverse shape should be 2x3

V shape will be 2x2

Solution 2 – using Housing dataset

3 (a) Briefly discuss the sources of bias in supervised learning (5 points)

(b) Discuss the bias variance trade-off (5 points)