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

Steps :   
a) Generate random values for x1 and x2 for A.

b) get U, d, VT by applying SVD of A values

c) Calculate D inverse

d) Since its of dimension of 2x2 , add a column to make it 2x3

e) Get A plus by dot product of V, D plus and U transpose

f) For verification, multiply A and A Plus and confirm if it is an identity matrix

g) for verification, get the A plus by numpy pinv function and confirm it matches the previously calculated

**Solution 2 – using Housing dataset**

Use the same approach with the difference that the housing data set has (21574, 8) dimension.

we have A as 21574x8 , D will be 21574x8 , hence U should be 21574x21574 and Vt is 8x8

21574x 8 = (21574x21574) x ( 21574x8 ) x ( 8x 8)

Hence

At shape should be 8x21574

Ut shape should be 21574x21574

D inverse shape should be 8x21574

V shape will be 8x 8

Steps :   
a) read housing dataset for A.

b) get U, d, VT by applying SVD of A values

c) Calculate D inverse

d) Since its of dimension of 8x8 , add 21566 columns to make it 8x21574

e) Get A plus by dot product of V, D plus and U transpose

f) For verification, multiply A and A Plus and confirm if it is an identity matrix

g) for verification, get the A plus by numpy pinv function and confirm it matches the previously calculated

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

Bias is a prejudice in favor or against a person, group, or thing that is considered to be unfair. Machine learning has shown great promise in several industries like medical, financial, retail etc to predict the future outcome based on the existing data. One key challenge is the presence of bias in the classifications and predictions of machine learning. If there is bias in model, it can have severe impact on the effectiveness of the model.

Though it is considered unfair to have a bias in anything in life, in machine learning, biases can be inherent in the data itself, It may be subtle and related to the source of data, the contents of the data (does it include elements that the model should be ignorant of?), and the training of the model itself.

As an example, if a model was trained solely on video of daytime driving, it would have tragic results if the model were permitted to drive at night. Similarly, if a financial model is trained only on the data of affluent San Francisco suburbs, the same model may not perform well for rural mid west population.

### In a machine learning models, the following can be sources of bias : 1. Sample Bias

Machine learning models are predictive engines that train on a large mass of data based on the past. They are made to predict based on what they have been trained to predict. These predictions are only as reliable as the human collecting and analyzing the data. The decision makers have to remember that if humans are involved at any part of the process, there is a greater chance of bias in the model.

The sample data used for training has to be as close a representation of the real scenario as possible. There are many factors that can bias a sample from the beginning and those reasons differ from each domain (i.e. geography, age, gender, time , education etc.)

**2. Prejudice Bias**

 This again is a cause of human input. Prejudice occurs as a result of cultural stereotypes in the people involved in the process. Social class, race, nationality, gender can creep into a model that can completely and unjustly skew the results of the model.

**3. Confirmation Bias**

 This is a bias that has been studied in the field of psychology and directly applicable to how it can affect a machine learning process. If the people of intended use have a pre-existing hypothesis that they would like to confirm with machine learning. The people involved in the modelling process might be inclined to intentionally manipulate the process towards finding that answer

**4. Association bias:** This bias occurs when the data for a machine learning model reinforces and/or multiplies a cultural bias. The dataset may have a collection of jobs in which all men are doctors and all women are nurses. This does not mean that women cannot be doctors, and men cannot be nurses. However, as far as our machine learning model is concerned, female doctors and male nurses do not exist.

**5. Measurement bias:** This type of bias occurs when the data collected for training differs from that collected in the real world, or when faulty measurements result in data distortion. A good example of this bias occurs in image recognition datasets, where the training data is collected with one type of camera, but the production data is collected with a different camera. Measurement bias can also occur due to inconsistent annotation during the data labeling stage of a project.

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

Whenever we make a model, there are some errors and inaccuracies built in the model. The prediction errors in a model can be broken down into 2 parts:  
1. Reducible Error  
2. Irreducible Error

Irreducible errors are errors that cannot be reduced even if we use any other machine learning model. Reducible errors, on the other hand, is further broken down into square of bias and variance. Due to this bias-variance, it causes the model to either overfit or underfit the given data.

There is a tradeoff between a model’s ability to minimize bias and variance. Gaining a proper understanding of these errors would help us not only to build accurate models but also to avoid the mistake of overfitting and underfitting. Bias Variance Tradeoff is a design consideration when training the model. Certain algorithms inherently have a high bias and low variance and vice-versa.

**Bias** : Bias is the difference between the average prediction of our model and the correct value which we are trying to predict. Model with high bias pays very little attention to the training data and oversimplifies the model. It always leads to high error on training and test data. Bias occurs when the model has limited flexibility to learn the true signal of dataset.

Symptoms

* Training error is higher than ϵϵ

Remedies:

* Use more complex model (e.g. kernelize, use non-linear models)
* Add features
* Boosting

**Variance**: Variance is the variability of model prediction for a given data point or a value which tells us spread of our data. Model with high variance pays a lot of attention to training data and does not generalize on the data which it hasn’t seen before. As a result, such models perform very well on training data but has high error rates on test data. Variance occurs when the model has high sensitivity to specific sets of training data.

Symptoms:

* Training error is much lower than test error
* Training error is lower than ϵϵ
* Test error is above ϵϵ

Remedies:

* Add more training data
* Reduce model complexity -- complex models are prone to high variance
* Bagging

Shape, circle

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

**Bias – Variance Tradeoff :** If the model is too simple and has very few parameters then it may have high bias and low variance. On the other hand if the model has large number of parameters then it’s going to have high variance and low bias. So we need to find the right/good balance without overfitting and underfitting the data.

This tradeoff in complexity is why there is a tradeoff between bias and variance. An algorithm can’t be more complex and less complex at the same time.

Total Error:To build a good model, we need to find a good balance between bias and variance such that it minimizes the total error.



Diagram

Description automatically generated

An optimal balance of bias and variance would avoid overfit or underfit the model.