AIT-736 Applied Machine Learning

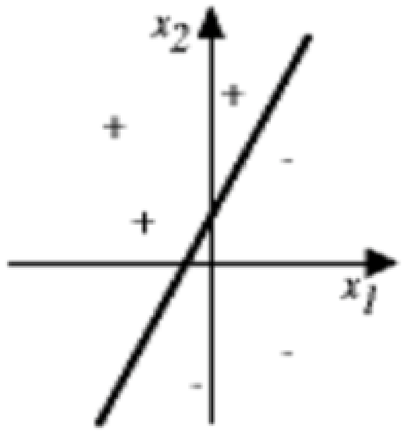
ASSIGNMENT 2 GROUP -2

GROUP -2

Indranil Pal
Yasser Parambathkandy
Brad Staton
Joshua Kloc

July 17, 2022

1. What are the values of the weights w_0, w_1, and w_2 for the perceptron whose decision surface is illustrated in the figure? Assume the surface crosses the x_1 axis at -1 and the x_2 axis at 2.



Solution:

The perceptron outputs a value *o* where $o = sgn(w_0 + w_1*x_1 + w_2*x_2)$, therefore the equation for the line separately the plus and minus items is:

$w_0 + w_1*x_1 + w_2*x_2 = 0$

Knowing two points on the line, (-1, 0) and (0, 2), we have the following equation:

$X_1 – (-1) / 0 – (-1) = X_2 – 0 / 2 – 0$

which simplifies to:

$X_1 + 1 = X_2 / 2$

which simplifies to:

$X_2/2 – X_1 – 1= 0$

Multiplying all terms by 2 to remove the fraction:

$X_2 – 2X_1 – 2 = 0$

Rearranged to the $w_0 + w_1*x_1 + w_2*x_2$ format and written in long form:

$-2 + (-2*x_1) + (1*x_2) = 0$

Therefore:

- $w_0 = -2$
- $w_1 = -2$
- $w_2 = 1$

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.

The source code for our solutions is included in the files *hw2_pla.py* and *hw2_linear_reg_pi.py*.

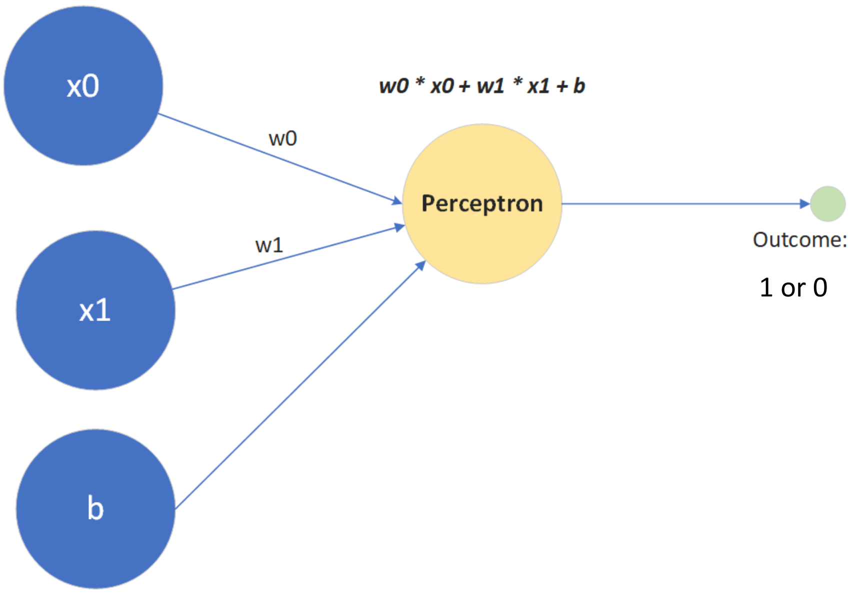
1) Generation of the data and labeling.

For PLA, we used the *make_blobs* function in the *sklearn* package to generate 1000 samples of random data. The parameters of the *make_blob* utility have been tweaked to produce a not so perfect linearly separable dataset. The intent is to show perceptron's ability to classification most this dataset and the few that it couldn't classify due to data being on the decision boundary line. The data has been split into training (80%) and test (20%) sets. The data set has two classes: 0 and 1.

For Linear regression using Pseudo Inverse, we generate random integers in the range -5 and 5.

2) Apply and describe PLA.

In machine learning, perceptron is a supervised learning algorithm used as a binary classifier. It is used to identify whether input data belongs to a specific group (class) or not. In our PLA, we have a simple perceptron as depicted in the diagram below.



The input range is specified as x_1 - x_n and x_0 is reserved as the bias value. Bias value is always one. Using 1 at the start of the array ensures if either of the other two values are zero, the next step always produces a value.

The next step in the perceptron algorithm is to calculate weighted sum with all the inputs as shown below:

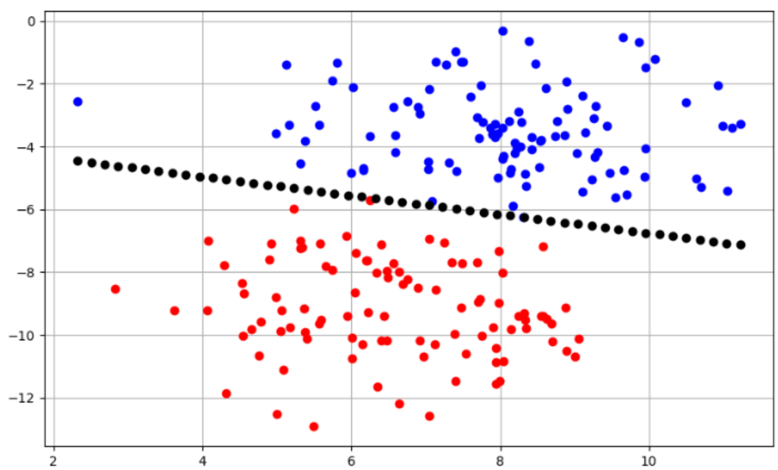
$x_1*w_1 + x_2*w_2 + x_3*w_3 + x_4*w_4 + ... + x_n*w_n + w0$

Once the weighted sum of inputs is obtained, the value is subjected to an activation function.
The activation function used shows the map between required values of 1 and 0

$$F(s)=\begin{cases} 1 \text{ if } s>0 \\ 0 \text{ if otherwise.} \end{cases}$$

The activation function performs the actual prediction. If the value from the previous step of weighted sum is greater than 0 then the function returns a 1 and if the value is negative, it returns a zero. Updating weights follows and is done by finding the difference between the actual and predicted value. This is the error value and is added to the original weights to get the updated weight. This step is required to ensure incremental adjustments are done to the weights.
This process is repeated for 1000 epochs. The training stops when there are no more errors. In the generated random sample dataset, the training stopped at 999 epochs.

The accuracy score is 98%. The decision boundary plot is shown below. Some of the datapoints are on the decision line. By tweaking the learning rate, this could be tuned for better accuracy.



3) Apply and describe Linear regression by computing the pseudo inverse

A common use of the pseudoinverse is to compute a "best fit" (least squares) solution to a system of linear equations that lacks a solution.

Underdetermined(m>n) – infinite solution

Overdetermined(n>m) – no solution

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

The data generated is a random 3x2 matrix.

We have A as 3x2, D will be 3x2, hence U should be 3x3 and Vt is 2x2
3x2 = (3x3) x (3x2) x (2x2)
Hence
- At shape should be 2x3
- Ut shape should be 3x3
- D inverse shape should be 2x3
- V shape will be 2x2

Algorithm:
a) Generate random values for x1 and x2 for A.
b) get U, d, and 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 using the *numpy pinv* function and confirm it matches the previously calculated value

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

Bias is a prejudice in favor or against a person, group, or thing that is unfair. In Machine learning, a key challenge is the presence of bias in the classifications and predictions problems. 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 should 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 should 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

Whenever we create 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

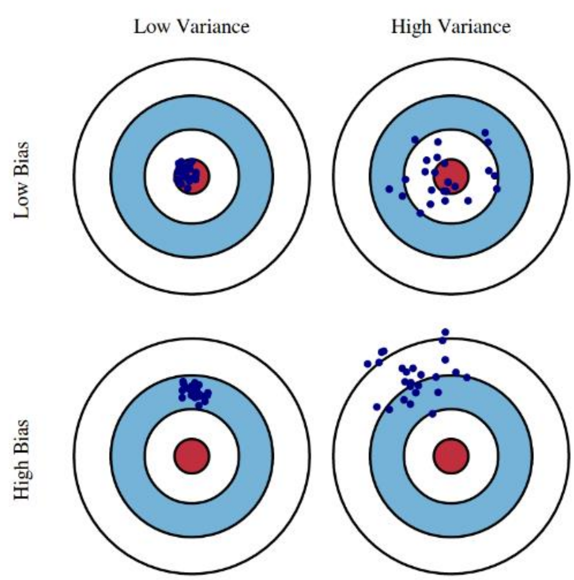
Variance: Variance is the variability of model prediction for a given data point or a value which tells us spread of our data. Model with high variance pays a lot of attention to training data and does not generalize on the data which it hasn't seen before. As a result, such models perform very well 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.

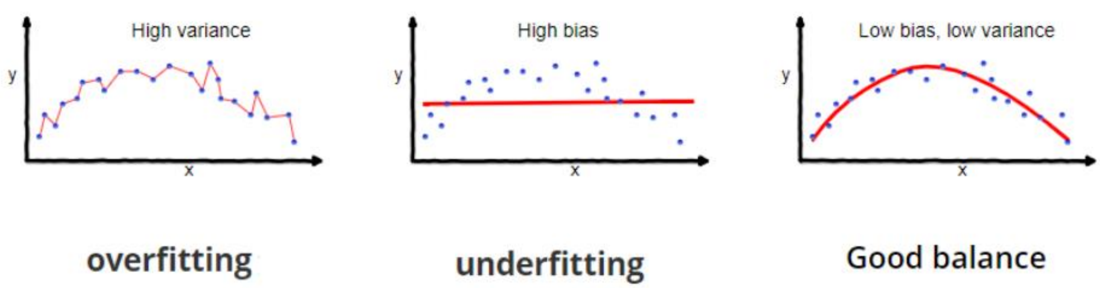
Symptoms:

- 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

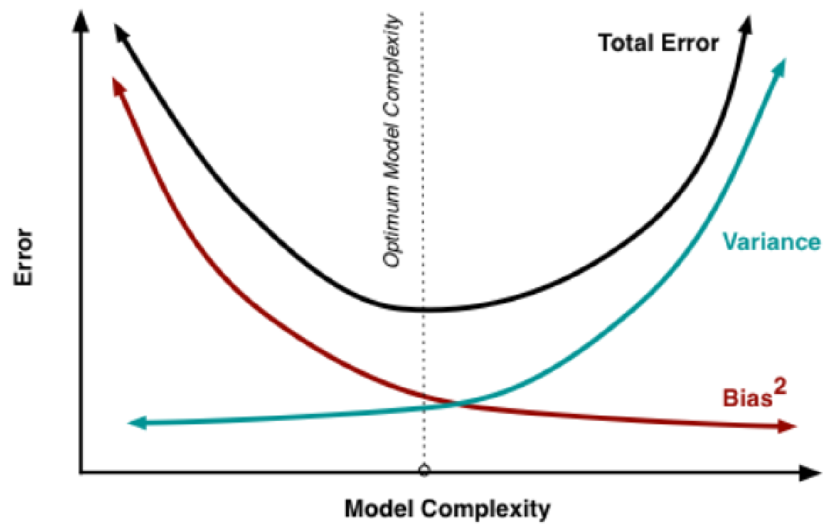




Bias – Variance Tradeoff: If the model is too simple and has very few parameters then it may have high bias and low variance. On the other hand, if the model has large number of parameters, then it's going

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.

Total Error = Bias^2 + Variance + Irreducible Error



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