12/27/2020

A PROJECT OF ARTIFICIAL INTELLIGENCE

LEARNING OF PERCEPTRON MODEL

BY: MEHMOOD YOUSAF, SIDRA AYESHA, SANA TARIQ, USAMA ALI

ACKNOWLEDGEMENT

It is saying that "A LEARNING CURVE IS ESSENTIAL TO GROWTH ", learning new things and tackle new problems are very interesting and adventurous as always.

We are highly Thankful to our respected instructor **Dr. Junaid Akhtar** who provided us a golden and useful opportunity to learn scientific approach through different phenomena's. It's above all the programming techniques, we are towards to map the real line mathematics concepts along with programming problems. We are on our way to learn and to be able to solve real life problems.

Problem statement

In this project our required outcome was to learn about calculate the weights. Straight line is basic need to apply this weight concept. We are given some Data points which have to be used in this techniques. We also have to clear the concept of straight line translation and its rotation as well. As we have given actual outcomes and with help of those points we have to predict the new weights value or points. We have to able to rectify this concept with coding in core python as well.

Predicted outcomes, desired outcomes and actual outcomes regarding weights are being used to clarify the line translation and rotation as well.

- ⇨ **Language constraint**

We have to map mathematics straight line concept with calculating the weights first by conceptual prototypes than our requirement and constraint about language was to use core python programming language.

Phenomena and Requirements

The concept was to learn about single line. Basically this concept is match of mathematics science as well, so line equation is also being involved in it. Equation of line depends on weights like W1, W2, and W0 as well. This is our pictorially representation of weights which could differentiate it clearly. These weights are regarding and along with the values of X, Y and training output or desired outputs. Basically we are calculating the weights along given ten data points, and then we are updating the weights according to new coming outcomes. There are basically two to three different situations on which our next updating of weights are dependent.

SCANARIOS:

- ➢ If there is a situation when delta w means change is same as desired outcome so this value of given data point might be skipped because this will be clearly classified in a specific category. So there is no need to further classify it.
- ➢ UN specified category is to be classify with this weights of line. So change in predicted and desired outcome is requirement for updating of weights towards checking the next data point.
- ❖ (Data point contains = value of x, value of y, desired output)
- ❖ If desired and predicted outputs are different and there rate of change is also different so there might be two paths:

- ❖ If calculated value is greater than "bias factor/ value" which is "one 1" so this will classify as good category.
- ❖ If calculated value is less than bias factor (one/1) so it will be considered as bad category.
- ➢ We will update the weights so there will be a clear point whether it would be classify as rotation of line and translation of line.
- ➢ Calculated weights are multiplied by value of given data points regarding x and y and then added up together so this factor will indicate the rotation and translation.
  - o This is basically technique and learning initial phase of machine learning.it means how actually a dumb machine will classify a real problem.

DESIRED GOAL

We have to learn basic machine learning so this is very initial and basic step towards it.

We have to calculate and learn the straight line concept and calculate the weights along with given data set or data points as well. The updating of weights will clear the classification concept so how things and real life huge data problems are being classified in a while.

➢ Conceptual Model:

The concept of the model is to learn a linear line in 2-D space. Linear line specifies the good or bad points. It refers to the model called "Perceptron Model".

- **Perceptron:**

  Perceptron is a model identical to the biological neuron in an artificial neural networks. The algorithm was designed to classify visual inputs, categorizing the subjects in two types and separate the groups by a line.

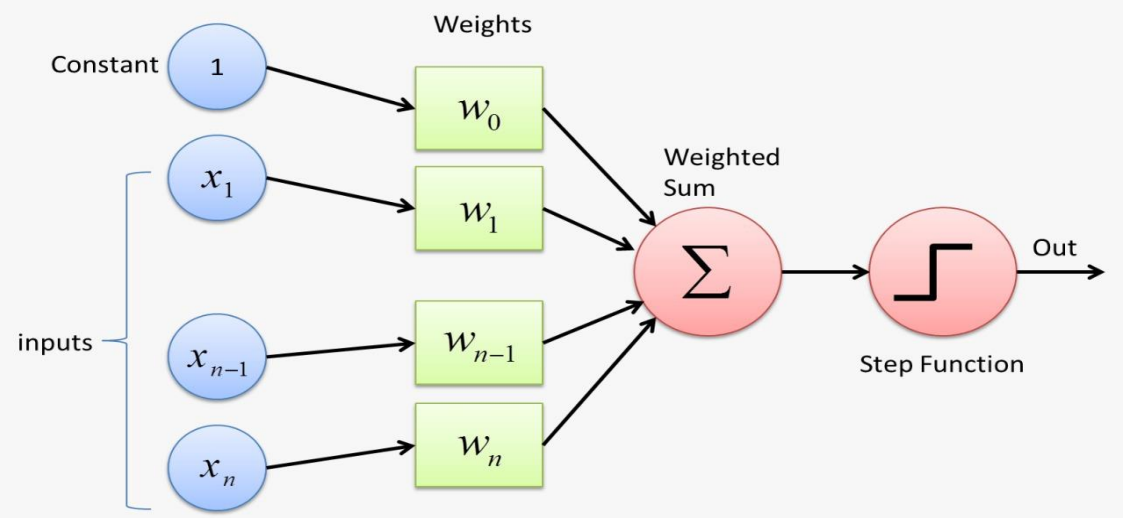
- **Perceptron Model in Neural Networks:**

  A perceptron is a neural network unit, called as an artificial neuron, does computations to detect features in the input data. It was introduced by **Frank Rosenblatt** in **1957**.

- **Perceptron Algorithm:**

  The algorithm is a **Linear Classifier** which makes predictions through a linear predictor function. The function has given some inputs as variables with some weights and a constant. They combine and give the output and algorithm predicts that output is good or bad.

  Below is the diagram of perceptron model/algorithm:



➢ Methodology:

Methods used in perceptron model are:

- **Learning**
- **Activation**

○ Learning():

The basic main motive of the project is "Learning" by testing the program. The combination of the inputs is done through the equation:

$$\text{Sum} = \omega_1 x + \omega 2y + \omega_o c$$

ω_1, ω_2, ω_o are old weights or initial weights given. X, y are variables with some value and c is constant. W1 and w2 are the weights that makes rotation while wo is for translation. Learning rate is 0.02. Job of learning is to update weights. It is done by the desired output – actual output. If output is zero then learning is not done. It means it made no mistake. And if there is another value rather than zero it means learning is done because it is making mistakes and weights has to update. As weights are depends on

i*): *learning rate

ii): n (D-A) x1

n is constant. D is Model's prediction and A is actual output. X1 is new variable/index on which mistake takes place.

○ Activation():

In this method activation_fucntion () is used. It is a prediction part. Which shows whether there is desired output or not. When weights are calculated then this function gives predictions, if sum of the weights is greater than and equal to zero then function returns 1 which means function classified it as good or it is desired output.
If sum is less than zero function returns -1 which is classified as bad.
activated_function ():

➔ **sum >= 0**
return 1
➔ **sum < 0**
return -1

So,

⇨ w.x >= 0 ⟶ Good
⇨ w.x < 0 ⟶ Bad

so, the main point is "when it made mistakes then there will be learning".

Application:

The perceptron learning model are being used in various fields. The application of Perceptron learning model is described following.

- **APPLICATION OF CLASSIFICATION:**

  Main application of this model is to classify between two things like good or bad. For example this model is used for the recommendation of movies or to classification of movies whether this is a good movie or bad. A simple perceptron is a single layer neural unit which is a linear classifier. It is a neuron capable of producing only two output patterns, which can be synthesized in *active* or *inactive*. Its decision rule is implemented by a *threshold* behavior: if the sum of the activation patterns of the individual neurons that make up the input layer, weighted for their weights, exceeds a certain threshold, then the output neuron will adopt the output pattern *active*. Conversely, the output neuron will remain in the *inactive* state.
- **APPLICATION OF BUILDING OF MINIG PROCESS:**

  Four models of a multilayer perceptron neural network were built for this purpose. Travel times and the condition of transport roads were adopted as input parameters. The output of the network is the cycle time of the analyzed process. On the basis of an analysis of learning errors, a model with two hidden layers was selected. A series of experiments was conducted on the selected model. An assessment was also performed to determine at which values of input parameters the stability of the analyzed process could be ensured.
- **APPLICATION FOR FAULT DETECTION:**

  The investigation was done based on the real data acquired from first stage of evaporation station of sugar factory. The non-linear perceptron MA-

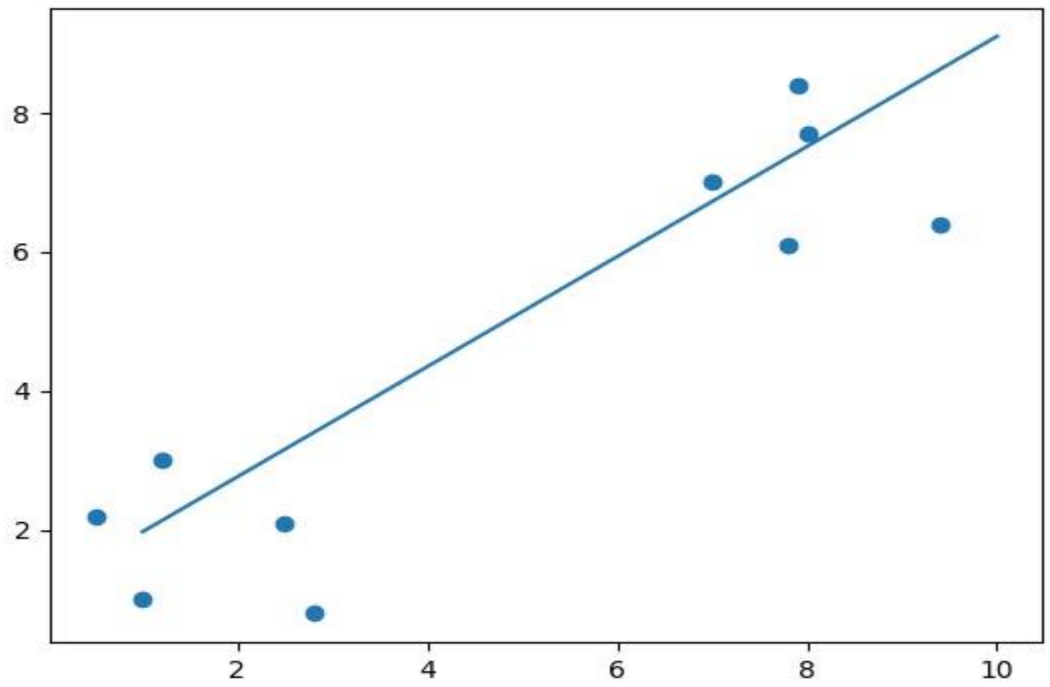
type models and perceptron ARMA-type models were examined. The only three-layer perceptron networks were investigated.

Exception:

There will be learning when model make mistake otherwise without mistake there will be no learning. During the implementation the problems solving and exceptions are following.

- If our system is learning and if there will be no learning in the 30 epochs then the flow our program will be stop and the learning process also stop.
- During the running of program if four or five epochs run and our learning is done then loop will break because there will be no need of learning. How we check this. When the value of activated function and the desired output value is same its mean there will be no need of learning because no mistake made by our model and the learning is complete so our model do not go to the next epoch for learning and the learning should be stop.

GRAPHICAL REPRESENTATION:



Given graph shows the final output of our model. The straight line was learned by the model by training and according to the given datasets. Below the lines of blue dots are good and up of the line the bad dots is present. This line is used as a classifier by our model and the classification of good or bad is done through the output of the activation function. If the activation function return value 1 the training data will be plot below the line on the other hand if the output of activation function is -1 the training set is bad and will plot above the line.

So, we did our best to implement best logic and we are on our way to become scientist ;)