Introduction To AI and ML
PLA : Perceptron Learning Algorithm
Presentation

P.Srijith Reddy,
EE19BTECH11041,
Dept. of Electrical Engg.,
IIT Hyderabad.

February 4, 2020

Perceptron

In Machine learning ,the perceptron is an algorithm for supervised learning of binary classifiers.

Binary classifiers : A binary classifier is a function which can decide whether or not an input, represented by a vector of numbers, belongs to some specific class

•It is a type of linear classifier, i.e. a classification algorithm that makes its predictions based on a linear predictor function combining a set of weights with the feature vector.

Threshold Function

The perceptron is an algorithm for learning a binary classifier called a threshold function.(f)

$$f : \mathbb{R}^n \to \mathbb{B} \tag{3.1}$$

Where B is a single binary value.

$$f(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{w} \cdot \mathbf{x} + b > 0, \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

where $\mathbf{w}$ is a vector of real-valued weights, $\mathbf{w}.\mathbf{x}=\sum_{i=1}^{m} w_i x_i$,
where m is the number of inputs to the perceptron, and b is the bias.

Steps of algorithm

Let's see an example for the algorithm (single-layer).

some variables :

• r is the learning rate of the perceptron. Learning rate is between 0 and 1.

• In machine learning and statistics, the learning rate is a tuning parameter in an optimization algorithm that determines the step size at each iteration while moving toward a minimum of a loss function.

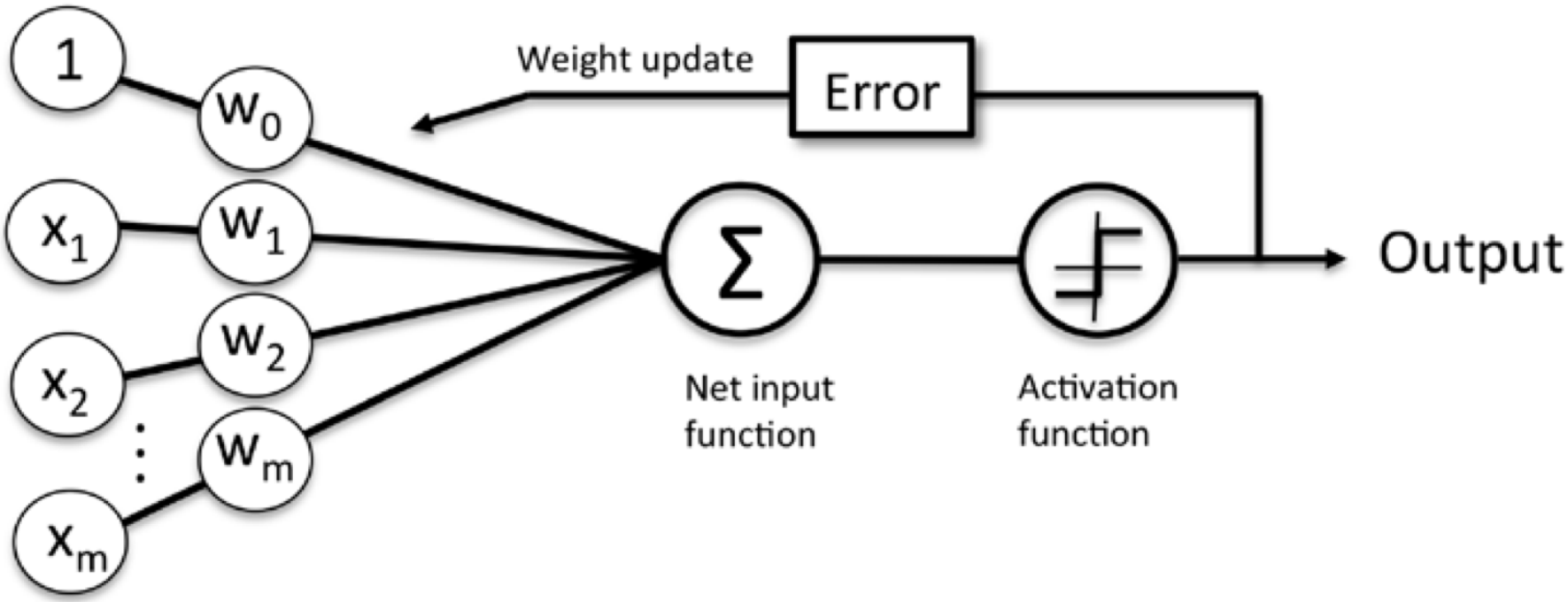
• $D = \{(\mathbf{x}_1, d_1), ..., (\mathbf{x}_s, d_s)\}$ is the training set of s samples.

- $\mathbf{x}_j$ *is the* $n-$*dimensional input vector*.
- d_j *is the desired output value for that input*.
- $\mathbf{w}_i(t)$ *is the weight* i *at time* t.

$$y_j(t) = f[\mathbf{w}(t).\mathbf{x}_j] \tag{4.1}$$

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + r.(d_j - y_i(t))x_{j,i} \tag{4.2}$$

Here $x_{j,i}$ *is the* i *th feature of* j *th training input vector*


1
w_0
x_1
w_1
w_2
x_2
w_m
x_m
Weight update
Error
Σ
Net input function
Activation function
Output

Activation function

The activation function applies a step rule (convert the numerical output into +1 or -1) to check if the output of the weighting function is greater than zero or not.

ex: sigmoid function , hyperbolic functions,softmax function.

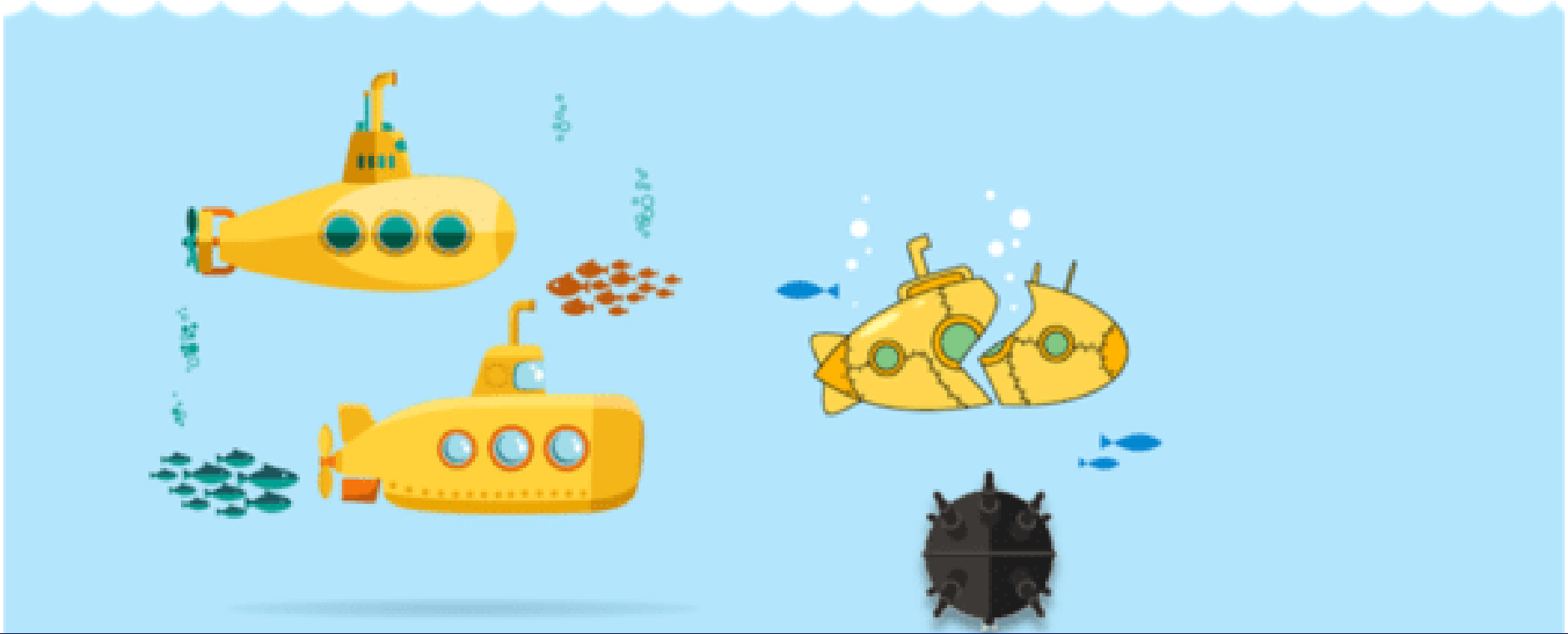
Error in perceptron

In the Perceptron Learning Rule, the predicted output is compared with the known output. If it does not match, the error is propagated backward to allow weight adjustment to happen.

SONAR Data Classification Using Single Layer Perceptrons

Here we have the data of a SONAR data set which contains the data about 208 patterns obtained by bouncing sonar signals off a metal cylinder (naval mine) and a rock at various angles and under various conditions.

- A naval mine is a self-contained explosive device placed in water to damage or destroy surface ships or submarines.
- we need to build a model that can predict whether the object is a naval mine or rock based on our data set.



Let's have a look at the information .

60 columns representing different features obtained by bouncing sonar signals off a metal cylinder and a rock

0.0453, 0.0523, 0.0843, 0.0689, 0.1183, 0.2583, 0.2156, 0.3481, 0.3337, 0.2872, 0.4918, 0.6552, 0.6919, 0.7797, 0.7464, 0.9444, 1, 0.8874, 0.8024, 0.7818, 0.5212, 0.4052, 0.3957, 0.3914, 0.325, 0.32, 0.3271, 0.2767, 0.4423, 0.2028, 0.3788, 0.2947, 0.1984, 0.2341, 0.1306, 0.4182, 0.3835, 0.1057, 0.184, 0.197, 0.1674, 0.0583, 0.1401, 0.1628, 0.0621, 0.0203, 0.053, 0.0742, 0.0409, 0.0061, 0.0125, 0.0084, 0.0089, 0.0048, 0.0094, 0.0191, 0.014, 0.0049, 0.0052, 0.0044, R

Label associated with each record contains the letter **"R"** if the object is a rock and **"M"** if it is a mine

IMPLEMENTATION

1.Read and Pre-process the data set:

• At first we will read the CSV file (input data set) using readcsv() function.

• Then, we will segregate the feature columns (independent variables) and the output column (dependent variable) as X and y respectively.

• The output column consists of string categorical values as M and R, signifying Rock and Mine respectively. So, I will label them them as 0 and 1 w.r.t. M and R.

2.Function for One Hot Encoder:

• One Hot Encoder adds extra columns based on number of labels present in the column. In this case, I have two labels 0 and 1 (for Rock and Mine). Therefore, two extra columns will be added corresponding to each categorical value

3.Dividing data set into Training and Test Subset

• While working on any deep learning project, we need to divide your data set into two parts where one of the parts is used for training your deep learning model and the other is used for validating the model once it has been trained.

• Training Subset: It is used for training the model

• Test Subset: It is used for validating our trained model.

4.Define Variables and Placeholders

• Learning Rate: The amount by which the weight will be adjusted.

• Training Epochs: No. of iterations

• Cost History: An array that stores the cost values in successive epochs.

• Weight: Tensor variable for storing weight values

• Bias: Tensor variable for storing bias values

5.Calculate the Cost or Error

• we will calculate the cost or error produced by our model. Instead of Mean Squared Error, we will use cross entropy to calculate the error in this case.

6.Training the Perceptron Model in Successive Epochs

• Now, we will train my model in successive epochs. In each of the epochs, the cost is calculated and then, based on this cost the optimizer modifies the weight and bias variables in order to minimize the error.

7.Validation of the Model based on Test Subset

• The accuracy of a trained model is calculated based on Test Subset. Therefore, at first, we will feed the test subset to my model and get the output (labels).

• Then, we will compare the output obtained from the model with that of the actual or desired output and finally, will calculate the accuracy as percentage of correct predictions out of total predictions made on test subset.

Output:

```
epoch :  986  -  cost:  0.423809
epoch :  987  -  cost:  0.423757
epoch :  988  -  cost:  0.423704
epoch :  989  -  cost:  0.423652
epoch :  990  -  cost:  0.4236
epoch :  991  -  cost:  0.423548
epoch :  992  -  cost:  0.423496
epoch :  993  -  cost:  0.423444
epoch :  994  -  cost:  0.423392
epoch :  995  -  cost:  0.42334
epoch :  996  -  cost:  0.423288
epoch :  997  -  cost:  0.423236
epoch :  998  -  cost:  0.423185
epoch :  999  -  cost:  0.423133
Accuracy:  0.833333
```

Code for SONAR Data Classification Using Single Layer Perceptron

code