

### Parshveneth Charteble Trucks

# COCCOCATION TO CHUMBERT HATE & A

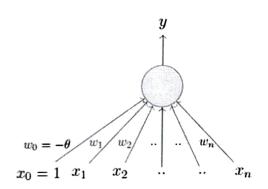


(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)
(Religious Jain Minority)

# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)

### Perceptron Learning Rule

We are going to use a perceptron to estimate if I will be watching a movie based on historical data with the above-mentioned inputs. The data has positive and negative examples, positive being the movies I watched i.e., 1.



 $x_1 = isActorDamon$ 

 $x_2 = isGenreThriller$ 

 $x_3 = isDirectorNolan$ 

 $x_4 = imdbRating(scaled to 0 to 1)$ 

.. ...

 $x_n = criticsRating(scaled to 0 to 1)$ 

### Perceptron Learning Algorithm:

Our goal is to find the  $\mathbf{w}$  vector that can perfectly classify positive inputs and negative inputs in our data. Initialize  $\mathbf{w}$  with some random vector.

We then iterate over all the examples in the data, (P U N) both positive and negative examples.

Now if an input x belongs to P, ideally what should the dot product w.x be? I'd say greater than or equal to 0 because that's the only thing what our perceptron wants at the end of the day so let's give it that. And if x belongs to N, the dot product must be less than 0.



### Parahyanath Charleadic Truck's

# A B SINH MERHAND OF THESINGLOCKY



(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai) (Religious Jain Minority)

# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)

```
Algorithm: Perceptron Learning Algorithm
P \leftarrow inputs \quad with \quad label \quad 1;
N \leftarrow inputs \quad with \quad label \quad 0;
Initialize w randomly;
while !convergence do
| \quad \text{Pick random } \mathbf{x} \in P \cup N ;
if \mathbf{x} \in P \quad and \quad \mathbf{w}.\mathbf{x} < 0 \text{ then}
| \quad \mathbf{w} = \mathbf{w} + \mathbf{x} ;
end
if \mathbf{x} \in N \quad and \quad \mathbf{w}.\mathbf{x} \geq 0 \text{ then}
| \quad \mathbf{w} = \mathbf{w} - \mathbf{x} ;
end
end
//the algorithm converges when all the inputs are classified correctly
```

Case 1: When x belongs to P and its dot product  $\mathbf{w}.\mathbf{x} < 0$ 

Case 2: When x belongs to N and its dot product  $\mathbf{w}.\mathbf{x} \ge 0$ 

Only for these cases, we are updating our randomly initialized  $\mathbf{w}$ . Otherwise, we don't touch  $\mathbf{w}$  at all because Case 1 and Case 2 are violating the very rule of a perceptron.

When x belongs to P, we want w.x > 0, basic perceptron rule. What we also mean by that is that when x belongs to P, the angle between w and x should be less than 90 degrees.



#### Parahyanetth Charlettha Trackla

## A. P. SIMI INSTRUMENT OF THEORIES



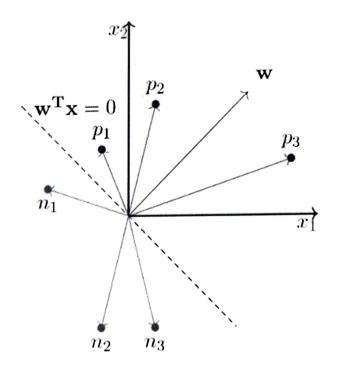
(Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai) (Religious Jain Minority)

# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)

$$\cos\alpha = \frac{\mathbf{w}^T \mathbf{x}}{||\mathbf{w}||||\mathbf{x}||} \qquad \cos\alpha \propto \mathbf{w}^T \mathbf{x}$$
So if  $\mathbf{w}^T \mathbf{x} > 0 \Rightarrow \cos\alpha > 0 \Rightarrow \alpha < 90$ 
Similarly, if  $\mathbf{w}^T \mathbf{x} < 0 \Rightarrow \cos\alpha < 0 \Rightarrow \alpha > 90$ 

So whatever the w vector may be, as long as it makes an angle less than 90 degrees with the positive example data vectors (x E P) and an angle more than 90 degrees with the negative example data vectors (x E N), we are cool.

So ideally, it should look something like this:





#### Parshvanath Charitable Trust's

## AL PA SHANI INSTRUMENTA OF TREE INTO LOCAY (Approved by AICTE New Delhi & Govt. of Maharashtra, Affiliated to University of Mumbai)

(Religious Jain Minority)

## DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)

So we now strongly believe that the angle between w and x should be less than 90 when x belongs to P class and the angle between them should be more than 90 when x belongs to N class.

Weight Updation:
For positive examples-

Knew= W+2

:. Knew = Klold +2

B new = Bold +2

For Negative examples -

KINEW = W-2

When = Wold-2

Bnew = Bold-x