CS-F415: Data Mining

12 Perceptron Introduction

Dr. Kamlesh Tiwari
Associate Professor, Department of CSIS,
BITS Pilani, Pilani Campus, Rajasthan-333031 INDIA

Feb 28, 2024 M/W/F 4:00pm 6101 @ BITS-Pilani [Jan-May 2024]

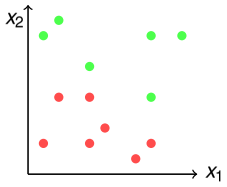
http://ktiwari.in/dm

Linear Classification

Consider Following data

x_1	x_2	y
1	9	green
10	9	green
4	7	green
4	5	red
5	3	red
8	9	green
4	2	red
2	5	red
7	1	red
2	10	green
8	5	green
1	2	red
8	2	red

Data is in 2D, so let us visualize



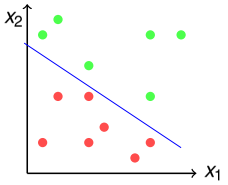
- Data looks linearly separable
- What is the decision boundary?

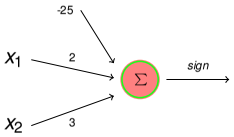
Many Possibilities, such as

if $(2x_1 + 3x_2 - 25 > 0)$ it is green
otherwise red

What about this arrangement?

With chosen *decision boundary* $2x_1 + 3x_2 - 25 = 0$





- This illustration is called as **perceptron**
- Provides a graphical way to represent the linear boundary
- Values 3, 2, -25 are its parameters or weights

Given a data

"How to find appropriate parameters?" is an important issue

Perceptron Training Rule

Different algorithms may converge to different acceptable hypotheses

Algorithm 1: Perceptron training rule

```
Begin with random weights w
repeat
    for each misclassified example do
        w_i = w_i + η(t − o)x_i
until all training examples are correctly classified;
return w
```

- **Why would this strategy converge?**
  1. Weight does not change when classification is correct
  2. If perceptron outputs -1 when target is +1: weight increases ↑
  3. If perceptron outputs +1 when target is -1: weight decreases ↓

Conversion with perceptron training rule is subject to linear separability of training example and appropriate η

Example

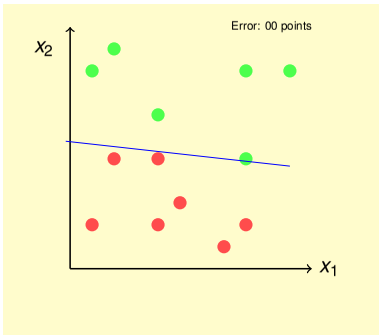
Consider the same data

x_1	x_2	y
1	9	green
10	9	green
4	7	green
4	5	red
5	3	red
8	9	green
4	2	red
2	5	red
7	1	red
2	10	green
8	5	green
1	2	red
8	2	red

$\eta = 0.01$

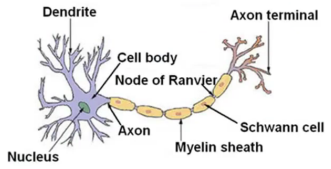
w0=0.500, w1=0.500, w2=0.500	err=7
w0=0.360, w1=-0.120, w2=0.100	err=6
w0=0.300, w1=-0.180, w2=0.060	err=5
w0=0.240, w1=-0.140, w2=0.140	err=4
w0=0.180, w1=-0.200, w2=0.100	err=5
w0=0.120, w1=-0.160, w2=0.180	err=4
w0=0.080, w1=-0.060, w2=0.180	err=5
w0=0.020, w1=-0.120, w2=0.140	err=4
w0=-0.040, w1=-0.180, w2=0.100	err=5
w0=-0.100, w1=-0.140, w2=0.180	err=4
w0=-0.140, w1=-0.040, w2=0.180	err=5
w0=-0.200, w1=-0.100, w2=0.140	err=3
w0=-0.260, w1=-0.160, w2=0.100	err=4
w0=-0.320, w1=-0.120, w2=0.180	err=3
w0=-0.360, w1=-0.020, w2=0.180	err=3
w0=-0.420, w1=-0.080, w2=0.140	err=2
w0=-0.420, w1=-0.080, w2=0.240	err=2
Fourteen more iterations	
w0=-0.900, w1=-0.020, w2=0.180	err=1
w0=-0.900, w1=-0.020, w2=0.240	err=2
w0=-0.920, w1=0.020, w2=0.220	err=2
w0=-0.960, w1=-0.020, w2=0.220	err=3
w0=-0.980, w1=0.020, w2=0.200	err=2
w0=-1.000, w1=0.060, w2=0.180	err=2
w0=-1.040, w1=0.020, w2=0.180	err=0

Visual Interpretation



- Conversion is not gradual. (Error is NOT reducing monotonically)
- It is difficult to decide when to stop if data is not linearly separable

Neural Network (NN)



- Cell, Axon, Synopses, Molicules, and Dendrites
- Humans have 10^{11} neurons, each connected to 10^4 others, switches in 10^{-3} sec

NN is biologically motivated learning model that mimic human brain

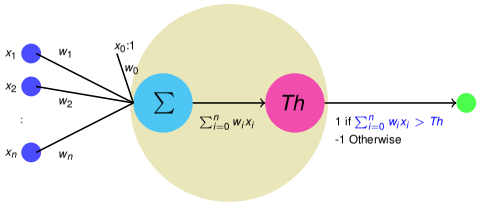
- Started by *W. McCulloch* study on working of neurons in 1943
- MADALINE (1959), an adaptive filter that eliminates echoes on phone lines was the first neural network
- Popularity of Neural Network diminished in 90's but, due to advances in **processing power** and availability of **large data** it again became state-of-the-art

Brief History

- 1872 Staining/Reticular Theory of Nervous Tissue
- 1943 McCulloch & Pitt (Neuron Model)
- 1947 Donald Hebb (Hebbian Learning)
- 1948 Nerbert (Cybernetics, optimal filter, feedback)
- 1954 Frank Rosenblatt (Perceptron)
- 1959 (MADALINE)
- 1962 Hubel & Wiesel (Visual Cortex Model)
- 1965 Frank Rosenblatt (MLP: Multi Layer Perceptron)
- 1969 Minsky (Limitations of Perceptron)
- 1986 (Backpropagation)
- 1989 Universal Approximation Theorem
- 1997 LSTM
- 1998 LeCun (ConvNet for MNIST digit)
- 2006 Unsupervised Pre Training
- 2010 Dahl. (Speech Recognition)
- 2012 AlexNet (ImageNet: Computer Vision) 26→16 … →12% zfNet
- 2013 VGG →7.3
- 2014 Google LeNet →6.7
- 2015 ResNet →3.6
- 2017 DenseNet
- 2018 Transformers, U-Net segmentation

A Single Perceptron

Perceptron representation

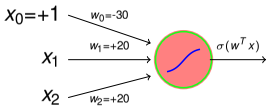


- A single perceptron can represent many boolean functions
- Any *m*-of-*n* function (at least *m* of the *n* inputs must be true) can be represented by perceptron. OR (*m*=1) and AND (*m*=*n*)

Two layer NN can represent any boolean function (Consider SOP)

An Example

Consider a perceptron with output 0/1 as below



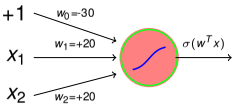
x_1	x_2	Output
0	0	0
0	1	0
1	0	0
1	1	1

This perceptron computes logical AND

- w_0=-10 gives logical OR
- w_0=10, w_1=-20 with single input gives logical NOT
- XOR is not possible (MLP[1] can do it)

[1] MLP (multi layer perceptron) with one-hidden-layer is *universal boolean function*, capable of expressing any truth table

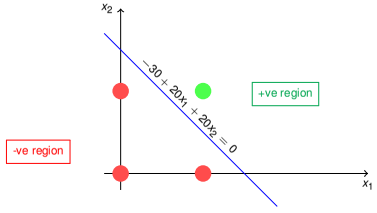
Essentially it Represents A Decision Boundary



Provides positive classification if

$$-30 + 20x_1 + 20x_2 \geq 0$$

Represents a linear decision boundary

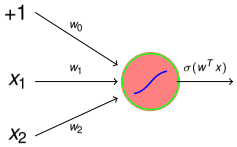


An Example

Design a **perceptron** for

x_1	x_2	Classification
0	0	0
0	1	0
1	0	1
1	1	0

Let us assume following



We have following four equations

$$w_0 + w_1 \times (0) + w_2 \times (0) < 0 \quad (1)$$
$$w_0 + w_1 \times (0) + w_2 \times (1) < 0 \quad (2)$$
$$w_0 + w_1 \times (1) + w_2 \times (0) \geq 0 \quad (3)$$
$$w_0 + w_1 \times (1) + w_2 \times (1) < 0 \quad (4)$$

By (1) $w_0 < 0$ so let $w_0 = -1$
By (2) $w_0 + w_2 < 0$ so let $w_2 = -1$
By (3) $w_0 + w_1 \geq 0$ so let $w_1 = 1.5$
By (4) $w_0 + w_1 + w_2 < 0$ that is valid

So $(w_0, w_1, w_2) = (-1, -1, 1.5)$

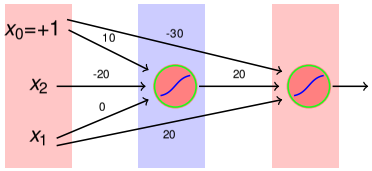
Other possibilities are also there

An Example

Design a **neural network** for

x_1	x_2	Classification
0	0	0
0	1	0
1	0	1
1	1	0

x_1	x_2	$\bar{x_2}$	$(x_1)AND(\bar{x_2})$
0	0	1	0
0	1	0	0
1	0	1	1
1	1	0	0



This arrangement is mostly avoided, as training is very challenging

Thank You!

Thank you very much for your attention!

Queries ?