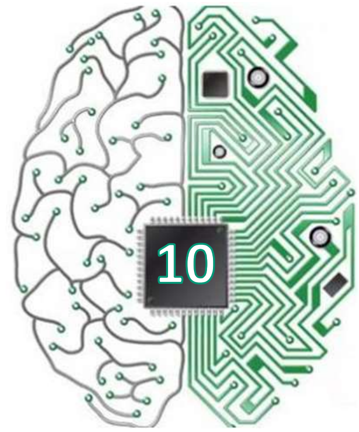
Open Elective Course [OE]
Course Code: CSO507
Winter 2023-24

Lecture#



Deep Learning

Unit-2: Linear and Logistic Regression (Part-III)

Unit-3: Artificial Neural Network (Part-I)

Course Instructor:
Dr. Monidipa Das
Assistant Professor
Department of Computer Science and Engineering
Indian Institute of Technology (Indian School of Mines) Dhanbad, Jharkhand 826004, India

Multi-Class Classification

Given:

– Data $\boldsymbol{X} = \left\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}\right\}$ where $\boldsymbol{x}^{(i)} \in \mathbb{R}^d$

– Corresponding labels $\boldsymbol{y} = \left\{y^{(1)}, \ldots, y^{(n)}\right\}$ where $y^{(i)} \in \{1, \ldots, K\}$

– Examples of multi-class classification:
  - classify e-mails as spam, travel, work, personal

- Targets form a discrete set $\{1, \ldots, K\}$. It is often more convenient to represent them as one-hot vectors, or a one-of-K encoding:

$$y^{(i)} = \underbrace{(0, \ldots, 0, 1, 0, \ldots, 0)}_{\text{entry } k \text{ is } 1} \in \mathbb{R}^K$$

Multi-Class Classification

3

- Now there are d input dimensions (plus adding a dummy variable $x_0 = 1$) and K output dimensions, so we need $K \times (d+1)$ weights, which we arranged as a weight matrix Θ.

$$\theta = \begin{bmatrix} | & | & & | \\ \theta^{(1)} & \theta^{(2)} & \cdots & \theta^{(K)} \\ | & | & & | \end{bmatrix}^{\mathrm{T}} \qquad \boldsymbol{z} = \Theta \boldsymbol{x} = \begin{bmatrix} z_1 \\ z_2 \\ \ldots \\ z_k \end{bmatrix} = \begin{bmatrix} \theta^{(1)T}\boldsymbol{x} \\ \theta^{(2)T}\boldsymbol{x} \\ \ldots \\ \theta^{(K)T}\boldsymbol{x} \end{bmatrix} \qquad \boldsymbol{z} \in \mathbb{R}^K$$

- We want soft predictions that are like probabilities, i.e., $0 \le \hat{y}_k = h_{\theta^{(k)}}(x) \le 1$ and $\sum_k \hat{y}_k = 1$.
- Use softmax function, a multivariable generalization of the logistic function such that:

$$\hat{y}_k = softmax(z_1, \ldots, z_K)_k = softmax(\mathbf{z})_k = \frac{\exp^{\theta^{(k)T}\boldsymbol{x}}}{\sum_{j=1}^{K} \exp^{\theta^{(j)T}\boldsymbol{x}}}$$

The inputs z_k are called the logits.

Multi-Class Classification

4

- Overall Softmax regression (with dummy $x_0 = 1$):

$$h_\theta(\boldsymbol{x}) = \hat{\boldsymbol{y}} = \mathrm{softmax}(\mathbf{z})$$

$$h_\theta(x) = \begin{bmatrix} P(y=1|x;\theta) \\ P(y=2|x;\theta) \\ \vdots \\ P(y=K|x;\theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^{K} \exp(\theta^{(j)\top} x)} \begin{bmatrix} \exp(\theta^{(1)\top} x) \\ \exp(\theta^{(2)\top} x) \\ \vdots \\ \exp(\theta^{(K)\top} x) \end{bmatrix}$$

- Loss/Cost Function: just like with logistic regression, we typically combine the softmax and cross-entropy into a ***softmax-cross-entropy function***.

$$J(\theta) = -\sum_{i=1}^{n}\sum_{k=1}^{K} y_k^{(i)} log\ h_{\theta^{(k)}} x^{(i)} = -\sum_{i=1}^{n}\sum_{k=1}^{K} y_k^{(i)} log\ P\left(y^{(i)} = k \middle| x^{(i)}; \theta\right) = -\sum_{i=1}^{n}\sum_{k=1}^{K} y_k^{(i)} log \frac{exp^{\theta^{(k)T} x^{(i)}}}{\sum_{j=1}^{K} exp^{\theta^{(j)T} x^{(i)}}}$$

Multi-Class Classification: Example

5



						$x^{(i)}$
$\theta^{(1)\mathrm{T}} \rightarrow$	1.1	0.2	-0.5	0.1	2.0	1
$\theta^{(2)\mathrm{T}} \rightarrow$	3.2	1.5	1.3	2.1	0.0	56
$\theta^{(3)\mathrm{T}} \rightarrow$	-1.2	0	0.25	0.2	-0.3	231
						24
						2

Θ $x^{(i)}$

$$\boldsymbol{z} = \Theta x^{(i)} = \begin{bmatrix} -96.8 \\ 437.9 \\ 60.75 \end{bmatrix} \longrightarrow \begin{bmatrix} 6.06\text{E-}233 \\ 1 \\ 1.61\text{E-}164 \end{bmatrix} \begin{matrix} \leftarrow h_{\theta^{(1)}} x^{(i)} \\ \leftarrow h_{\theta^{(2)}} x^{(i)} \\ \leftarrow h_{\theta^{(3)}} x^{(i)} \end{matrix}$$

$$h_\theta\left(x^{(i)}\right) = \hat{y}^{(i)} = softmax(\boldsymbol{z})$$

Predicted Output

$$y^{(i)} \; \boxed{2} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \begin{matrix} \leftarrow y_1^{(i)} \\ \leftarrow y_2^{(i)} \\ \leftarrow y_3^{(i)} \end{matrix}$$

← One-hot encoding of the actual output

softmax-cross-entropy function Loss

$$\sum_{k=1}^{K} y_k^{(i)} log\ h_{\theta^{(k)}} x^{(i)} = 0$$

Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad

Performance Metrics: Classification

6

- **Binary Classification**
  - **Confusion matrix:** The confusion matrix is used to have a more complete picture when assessing the performance of a model.

Predicted Class

Actual Class	+	-
+	**TP** True Positives	**FN** False Negatives Type II error
-	**FP** False Positives Type I error	**TN** True Negatives

Main metrics

Metric	Formula
Accuracy	$\frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}}$
Precision	$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$
Recall Sensitivity	$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$
Specificity	$\frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}}$
F1 score	$\frac{2\mathrm{TP}}{2\mathrm{TP}+\mathrm{FP}+\mathrm{FN}}$

The ideas are extendible to Multiclass-classification

Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad

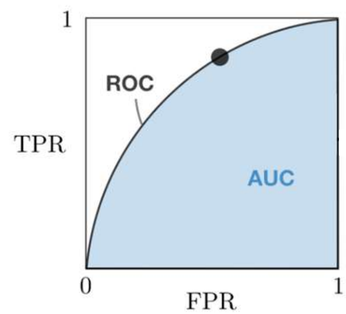
Performance Metrics: Classification

7

- **ROC:** The receiver operating curve
  - the plot of TPR versus FPR by varying the threshold

- **AUC:** The area under the receiving operating curve

Metric	Formula	Equivalent
True Positive Rate TPR	$\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}$	Recall, sensitivity
False Positive Rate FPR	$\frac{\mathrm{FP}}{\mathrm{TN}+\mathrm{FP}}$	1-specificity



Performance Metrics: Regression

8

- **Basic metrics:** Given a regression model f, the following metrics are commonly used to assess the performance of the model:

Total sum of squares	Explained sum of squares	Residual sum of squares
$\mathrm{SS}_{\text{tot}} = \sum_{i=1}^{n}(y_i - \overline{y})^2$	$\mathrm{SS}_{\text{reg}} = \sum_{i=1}^{n}(f(x_i) - \overline{y})^2$	$\mathrm{SS}_{\text{res}} = \sum_{i=1}^{n}(y_i - f(x_i))^2$

- **Coefficient of determination:** The coefficient of determination, often noted R^2 or r^2, provides a measure of how well the observed outcomes are replicated by the model and is defined as follows:

$$\boxed{R^2 = 1 - \frac{\mathrm{SS}_{\text{res}}}{\mathrm{SS}_{\text{tot}}}}$$

Sample Questions

9

Q1 Let $\theta^* \in \mathbb{R}^d$, and let $f(\theta) = \frac{1}{2}\|\theta - \theta^*\|^2$. Show that the Hessian of f is the identity matrix. **[This question is related to Unit-1 of the course as well]**

Q2 Consider the following training data.

x_1	x_2	y
0	0	0
0	1	1.5
1	0	2
1	1	2.5

Suppose the data comes from a model $y = \theta_0 + \theta_1 x_1 + \theta_2 x_2$ for unknown constants $\theta_0, \theta_1, \theta_2$. Use linear regression to find an estimate of $\theta_0, \theta_1, \theta_2$.

Q3 Consider a binary classification problem whose features are in $\mathbb{R}^2$. Suppose the predictor learned by logistic regression is $\sigma(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$, where $\theta_0 = 4, \theta_1 = -1, \theta_2 = 0$. Find and plot curve along which P(class 1) = 1/2 and the curve along which P(class 1) = 0.95.

Sample Questions

10

Q4 Consider the following training dataset corresponding to spam email recognition task. Apply logistic regression to determine whether a new email containing the word "bank" and "burkina" is spam or not. Assume $\boldsymbol{\alpha}$ = 0.5 and initial $\boldsymbol{\theta}$ = [0, 0, 0, 0, 0, 0]

	and	bank	the	of	burkina	y
Email **a**	1	1	0	1	1	1
Email **b**	0	0	1	1	0	0
Email **c**	0	1	1	0	0	1
Email **d**	1	0	0	1	0	0
Email **e**	1	0	1	0	1	1
Email **f**	1	0	1	1	0	0

A Training Dataset

Q5 Consider a 3-class classification problem. You have trained a predictor whose input is $x \in \mathbb{R}^2$ and whose output is softmax$(x_1 + x_2 - 1, 2x_1 + 3, x_2)$. Find and sketch the three regions in $\mathbb{R}^2$ that gets classified as class 1, 2, and 3.

Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad

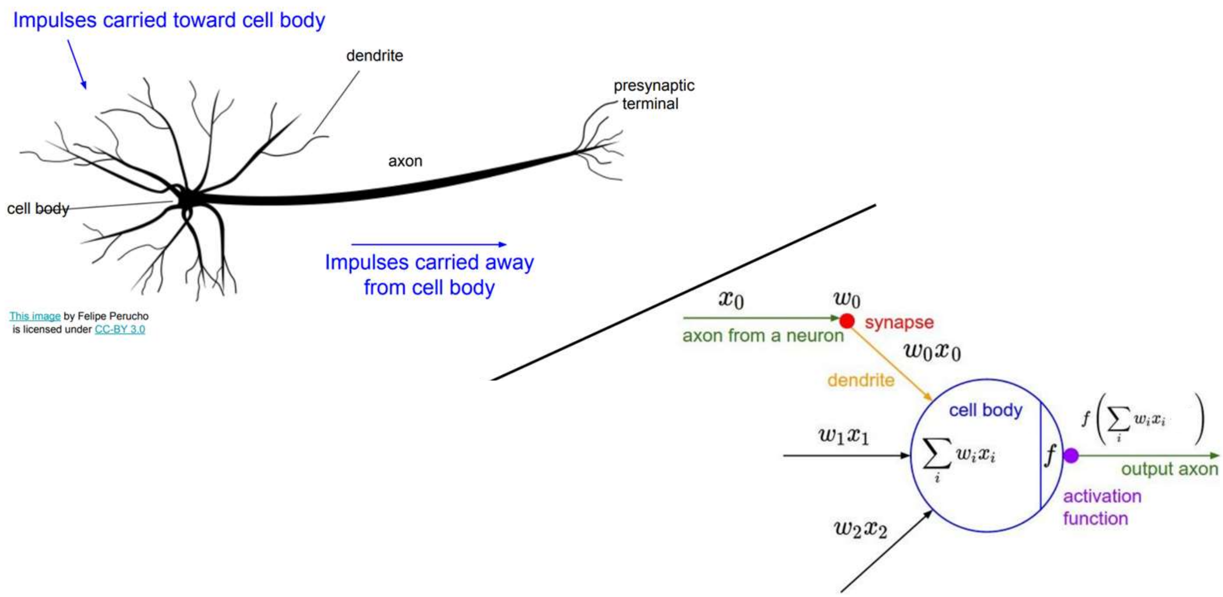
Artificial Neural Network

(Unit-3)

Neural Function

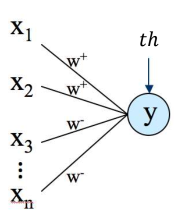
- Brain function (thought) occurs as the result of the firing of neurons
- Neurons connect to each other through synapses, which propagate action potential (electrical impulses) by releasing neurotransmitters
  - Synapses can be excitatory (potential-increasing) or inhibitory (potential-decreasing), and have varying activation thresholds
  - Learning occurs as a result of the synapses' plasticity: They exhibit long-term changes in connection strength
- "One Learning Algorithm" Hypothesis
- There are about 10^{11} neurons and about 10^{14} synapses in the human brain!

Neural Networks



Brief History of Neural Network

McCulloch-Pitts Neuron



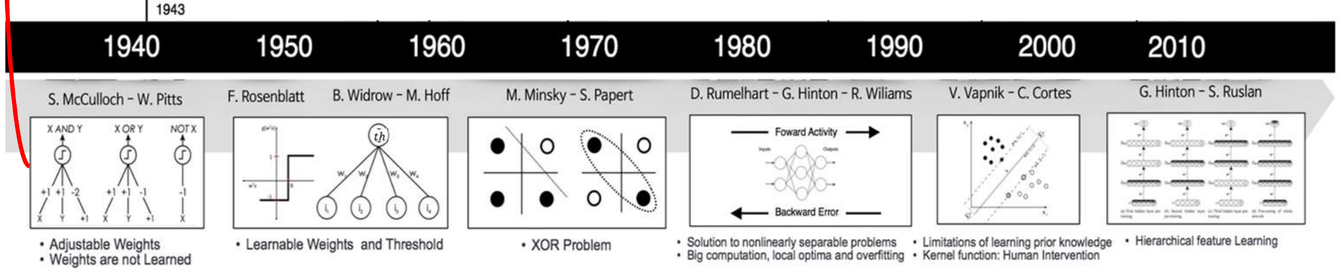
x_i - input
w^+ - excitatory input ($w > 0$)
w^- - inhibitory input ($w < 0$)
th - firing threshold
y - output

- Only models binary input
- Structure doesn't change
- Weights are set by hand
  - No learning!!

Electronic Brain

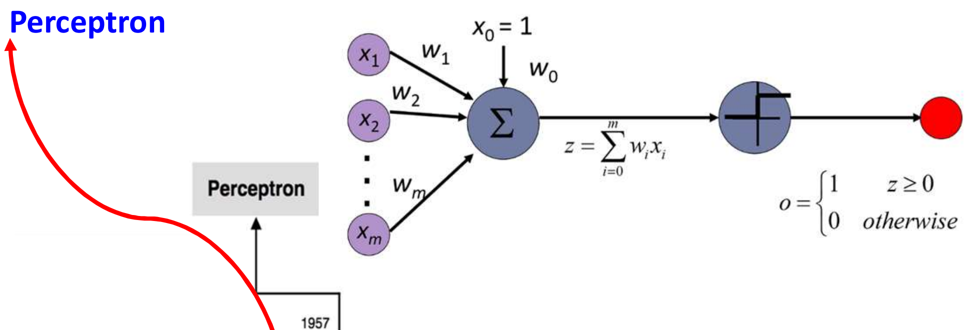
1943

$y = 1$ if sum of excitatory inputs $>= th$ and no inhibitory input
$y = 0$ if sum of excitatory inputs $< th$ or inhibitory input

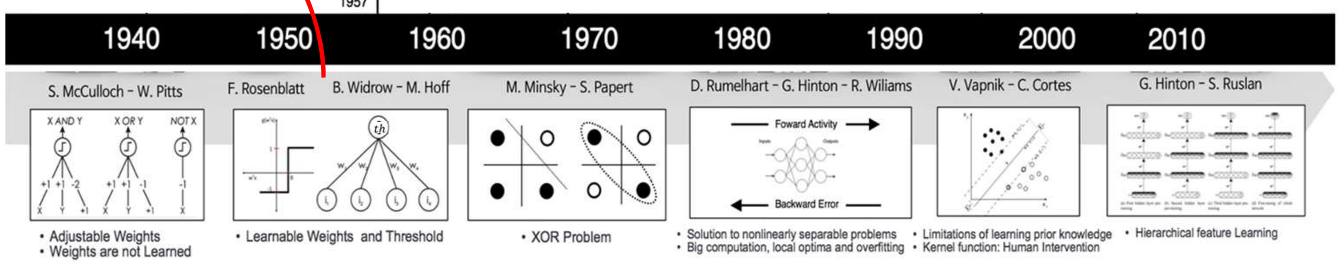


Brief History of Neural Network

15

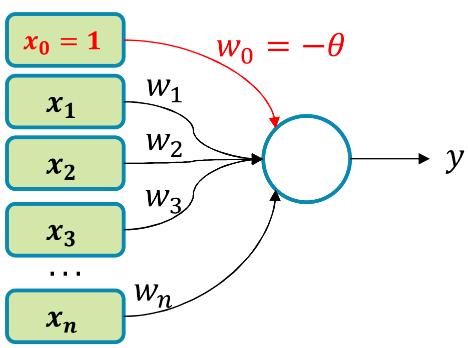


- Learnable weights and threshold
- Guarantee to converge within a finite number of iterations – i.e., weight vector is able to classify all examples correctly. Learning rate α needs to be sufficiently small.
- Training examples should be linearly separable



Perceptron: Introduction

16



$$y = \begin{cases} 1 \;\; if \sum_{i=0}^{n} w_i x_i \geq 0 \\ 0 \;\; if \sum_{i=0}^{n} w_i x_i < 0 \end{cases}$$

$$x_0 = 1, w_0 = -\theta$$

Perceptron: Learning Algorithm

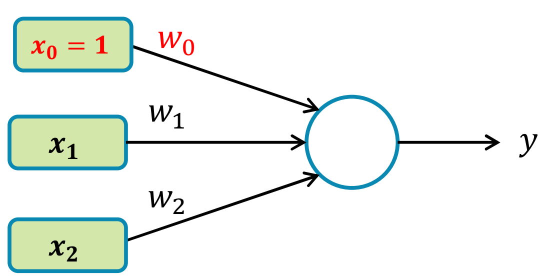
Algorithm: Perceptron Learning Algorithm

$P \leftarrow inputs \quad with \quad label \quad 1;$
$N \leftarrow inputs \quad with \quad label \quad 0;$
Initialize $\mathbf{w}$ randomly;
while $!convergence$ **do**
    Pick random $\mathbf{x} \in P \cup N$;
    if $\mathbf{x} \in P \quad and \quad \sum_{i=0}^{n} w_i * x_i < 0$ **then**
        $\mathbf{w} = \mathbf{w} + \mathbf{x}$;
    end
    if $\mathbf{x} \in N \quad and \quad \sum_{i=0}^{n} w_i * x_i \geq 0$ **then**
        $\mathbf{w} = \mathbf{w} - \mathbf{x}$;
    end
end
//the algorithm converges when all the inputs are classified correctly

Perceptron Learning Example

- Implement OR function with binary inputs and binary targets using perceptron training algorithm

x_1	x_2	t
0	0	0
0	1	1
1	0	1
1	1	1



Perceptron: Learning

19

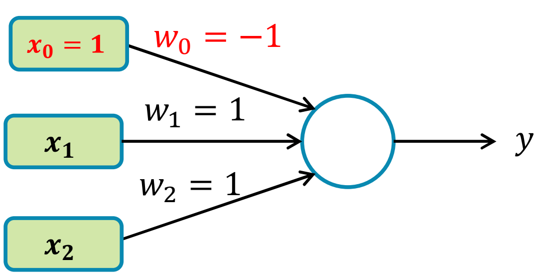
	Input			Target	Net Input	Calculated output	Weights		
	x_1	x_2	x_0	t	y_{in}	y	w_1	w_2	w_0
							0	0	0
EPOCH-1	0	0	1	0	0	1	0	0	-1
	0	1	1	1	-1	0	0	1	0
	1	0	1	1	0	1	0	1	0
	1	1	1	1	1	1	0	1	0
EPOCH-2	0	0	1	0	0	1	0	1	-1
	0	1	1	1	0	1	0	1	-1
	1	0	1	1	-1	0	1	1	0
	1	1	1	1	2	1	1	1	0
EPOCH-3	0	0	1	0	0	1	1	1	-1
	0	1	1	1	0	1	1	1	-1
	1	0	1	1	0	1	1	1	-1
	1	1	1	1	1	1	1	1	-1

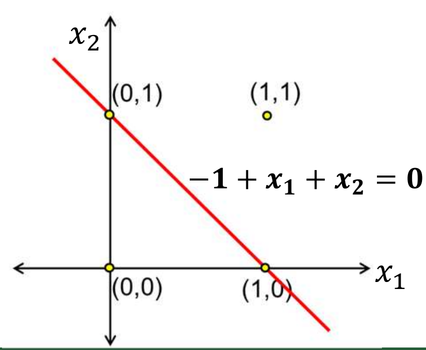
Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad

Perceptron: Learning

20

	Input			Target	Net Input	Calculated output	Weights		
	x_1	x_2	x_0	t	y_{in}	y	w_1	w_2	w_0
							1	1	-1
EPOCH-4	0	0	1	0	-1	0	1	1	-1
	0	1	1	1	0	1	1	1	-1
	1	0	1	1	0	1	1	1	-1
	1	1	1	1	1	1	1	1	-1





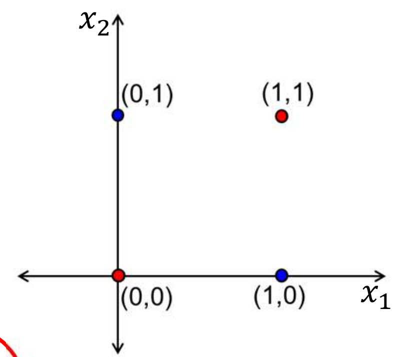
Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad

Points to Remember

- *Real valued inputs are allowed* in perceptron
- A ***single perceptron* cannot learn a function** that is **not linearly separable**

x_1	x_2	Target (XOR)	Objective	
0	0	0	$w_0 + w_1x_1 + w_2x_2 < 0$	$\Rightarrow w_0 < 0$
0	1	1	$w_0 + w_1x_1 + w_2x_2 \geq 0$	$\Rightarrow w_2 \geq -w_0$
1	0	1	$w_0 + w_1x_1 + w_2x_2 \geq 0$	$\Rightarrow w_1 \geq -w_0$
1	1	0	$w_0 + w_1x_1 + w_2x_2 < 0$	$\Rightarrow w_1 + w_2 < -w_0$

Contradictory!



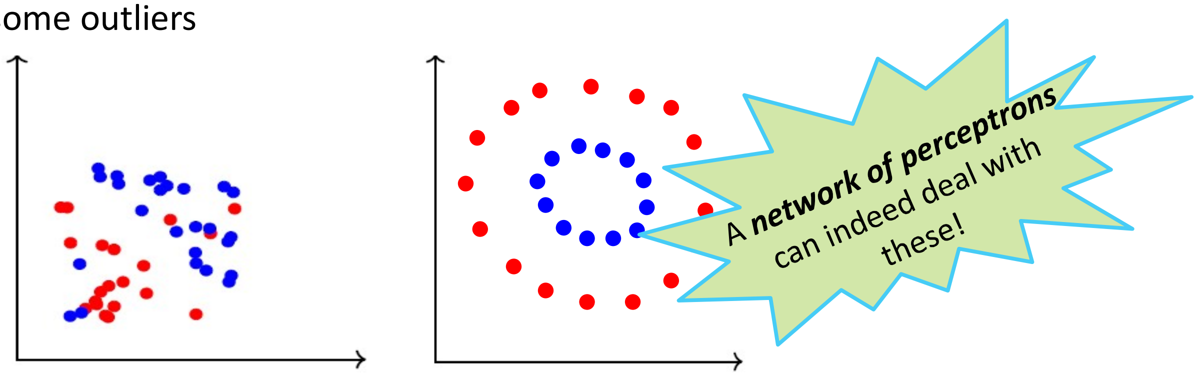
Perceptron Convergence and Cycling Theorems

- **Perceptron convergence theorem**: If the data is linearly separable and therefore a set of weights exist that are consistent with the data, then the Perceptron algorithm will eventually converge to a consistent set of weights.
- **Perceptron cycling theorem**: If the data is not linearly separable, the Perceptron algorithm will eventually repeat a set of weights and threshold at the end of some epoch and therefore enter an infinite loop.

Points to Remember

23

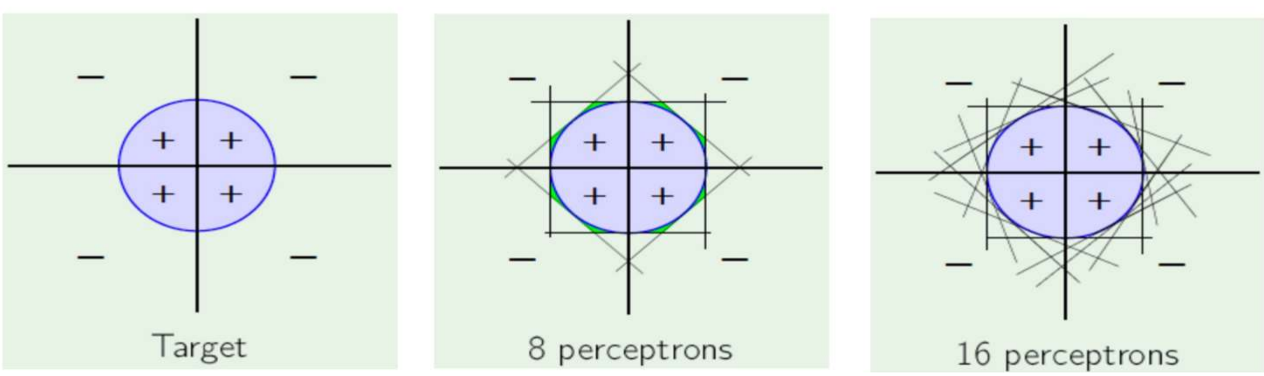
- Most real-world data is **not linearly separable** and will always contain some outliers



How do we implement functions that are not linearly separable ?

Combining Many Linear Classifiers

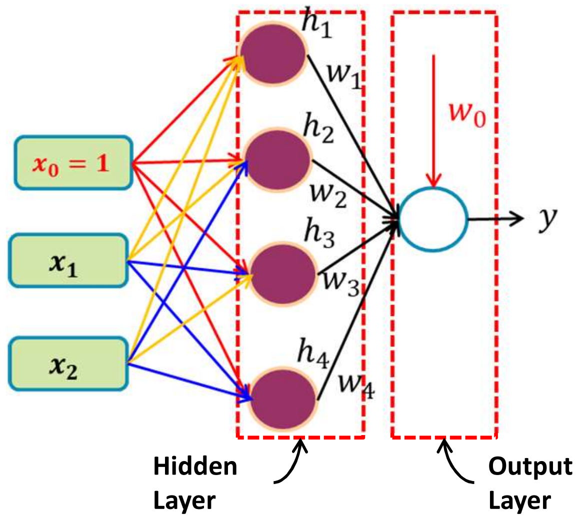
24



Perceptron Network

25

- Any boolean function of n inputs can be represented by a network of perceptrons containing ***1 hidden layer with 2^n perceptrons*** and ***one output layer containing 1 perceptron***
- Perceptron networks of these forms are called Multilayer Perceptrons (MLP)

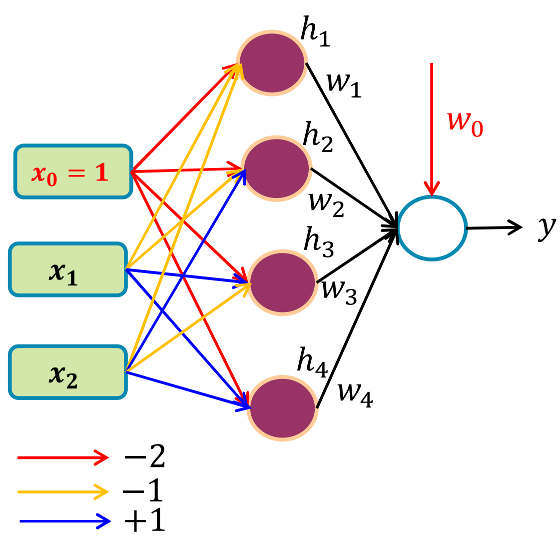


Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad

Perceptron Network

26

- This network can be used to implement any boolean function (linearly separable or not) [we assume the inputs are bipolar here]
- Each perceptron in the middle layer fires only for a specific input (and no two perceptrons fire for the same input)
- We need to find appropriate w_1, w_2, w_3, and w_4

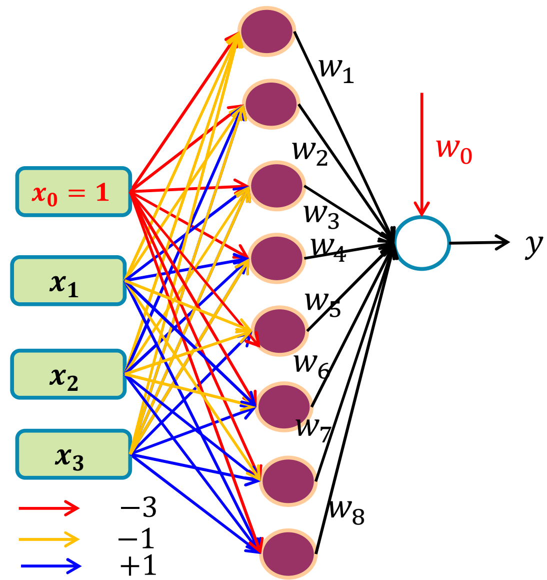


Prof. Monidipa Das, Department of CSE, IIT (ISM) Dhanbad

Perceptron Network

27

- If we have more than 2 inputs
  - Each of these 8 perceptrons will fire only for one of the 8 inputs
  - Each of the 8 weights in the second layer is responsible for one of the 8 inputs and can be adjusted to produce the desired output for that input



Example: XOR function

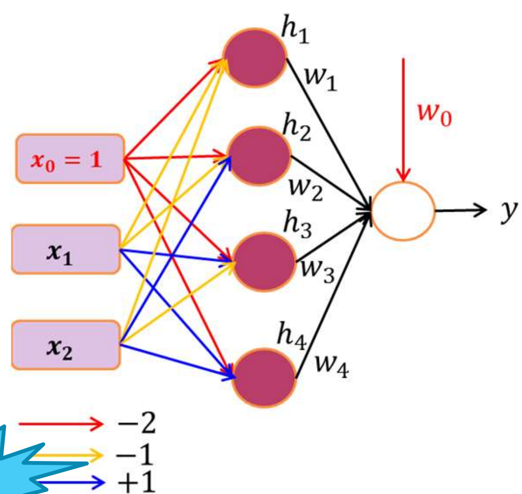
28

x_1	x_2	XOR	h_1	h_2	h_3	h_4	$\sum_{i=1}^{4} w_i * h_i$
-1	-1	-1	1	0	0	0	w_1
-1	1	1	0	1	0	0	w_2
1	-1	1	0	0	1	0	w_3
1	1	-1	0	0	0	1	w_4

$w_1 + w_0 < 0 \Rightarrow w_1 < 2$
$w_2 + w_0 \geq 0 \Rightarrow w_2 \geq 2$
$w_3 + w_0 \geq 0 \Rightarrow w_3 \geq 2$
$w_4 + w_0 < 0 \Rightarrow w_4 < 2$

e.g.
[1.5, 2, 2.5, 1.2]
[1, 2.5, 2.1, 1.8]
[1, 2, 2, 1]

XOR!



Perceptron: Drawback

29

- Output of perceptron: $\sum w_i x_i$
- For both inputs and output, -ve means logical 0, +ve means logical 1
- Basically, a hard threshold decides the output (logical 0 or 1)
- Optimization becomes difficult with many perceptrons

- We would like to change the input a little and see how the output changes (iterative methods)

- **Desirable Property:**
  - Instead of a hard threshold, a smooth function that is efficient to differentiate
  - So that we can change the inputs a little, observe the corresponding small change in the output, hence compute gradient, etc.

- ***A perceptron with a smooth non-linear function*** is equivalent to a ***neuron*** *in NN*

30

Questions?