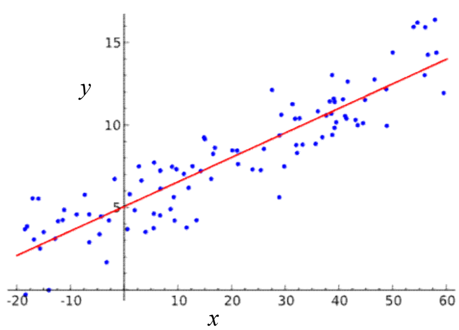
Linear Method for Classification

Linear Regression

- Linear approach to model the relationship between a scalar response, (y) (or dependent variable) and one or more predictor variables, (x or $\mathbf{x}$) (or independent variables)
- The response is going to be the linear function of input (one or more independent variables)
- Optimal coefficient vector $\mathbf{w}$ is given by

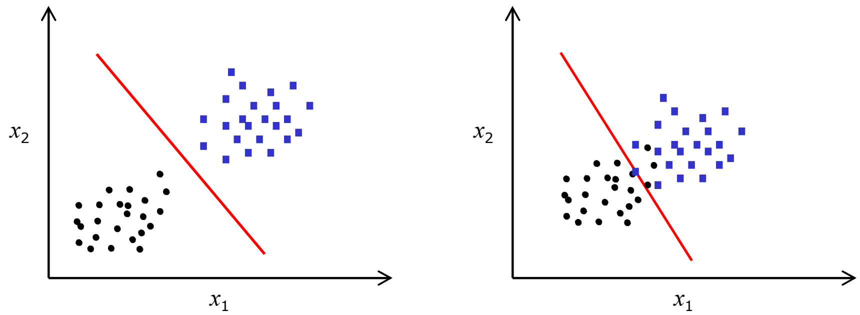
$$\hat{\mathbf{w}} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{y}$$



2

Linear Method for Classification

- The boundary that separates the region of classes is linear
- Separating surface will be a hyperplane
- A hyperplane that best fit the region of separation between the classes



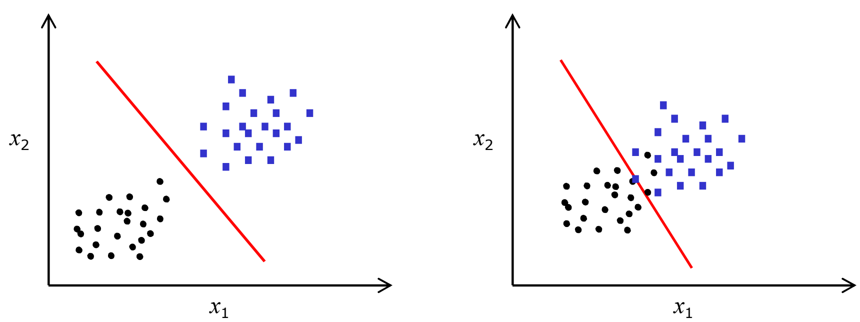
- Discriminant function in 2-dimensional space :

$\mathbf{x}_n=[x_{n1}, x_{n2}]^\mathsf{T}$ $\quad f(\mathbf{x}_n, w_1, w_2, w_0) = w_1\, x_{n1} + w_2\, x_{n2} + w_0$

3

Linear Method for Classification

- The boundary that separates the region of classes is linear
- Separating surface will be a hyperplane
- A hyperplane that best fit the region of separation between the classes



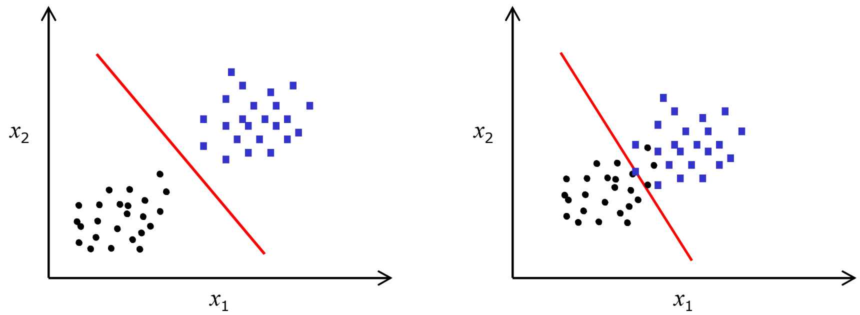
- Discriminant function in 2-dimensional space :

$\mathbf{x}_n=[x_{n1}, x_{n2}]^\mathsf{T}$ $\quad f(\mathbf{x}_n, w_1, w_2, w_0) = w_1\, x_{n1} + w_2\, x_{n2} + w_0 = 0$

4

Linear Method for Classification

- The boundary that separates the region of classes is linear
- Separating surface will be a hyperplane
- A hyperplane that best fit the region of separation between the classes



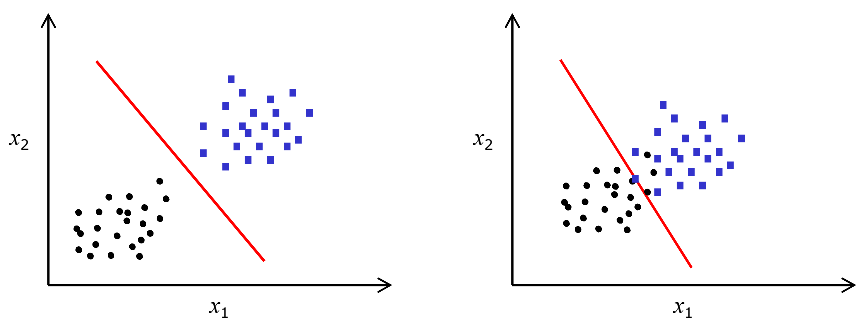
- Discriminant function in 2-dimensional space :

$\mathbf{x}_n=[x_{n1}, x_{n2}]^{\mathsf{T}}$ $$x_{n2} = -\frac{w_1}{w_2}x_{n1} - \frac{w_0}{w_2} = mx_{n1} + c$$

5

Linear Method for Classification

- The boundary that separates the region of classes is linear
- Separating surface will be a hyperplane
- A hyperplane that best fit the region of separation between the classes



- Discriminant function in d-dimensional space:

$$f(\mathbf{x}_n, \mathbf{w}) = \mathbf{w}^{\mathsf{T}}\mathbf{x}_n + w_0 = \sum_{i=0}^{d} w_i x_i$$

6

Two classes of Approaches for Linear Classification

1. Modeling a discriminating function:
   - For each class, a linear discriminant function $f_i(\mathbf{x},\mathbf{w}_i)$ is defined
   - Let C_1, C_2, …, C_i, …, C_M be the M classes
   - Let $f_i(\mathbf{x},\mathbf{w}_i)$ be the linear discriminant function for i^{th} class

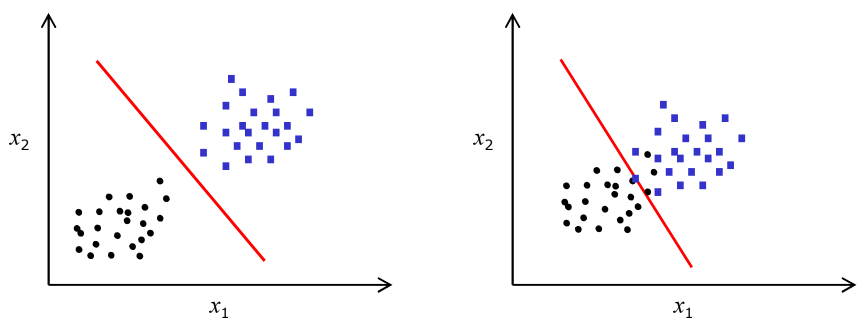
     Class label for $\mathbf{x}$ = $\arg\max_i f_i(\mathbf{x},\mathbf{w}_i) \qquad i = 1, 2, \ldots, M$

   - Discriminant function is defined independent of the classes
   - Linear regression can be treated as discriminant function
     - Do the linear regression by considering dependent variable as indicator variable
   - Logistic regression

7

Two classes of Approaches for Linear Classification

2. Directly learn a discriminant function (hyperplane):
   - Classic method: Discriminant function between the classes is learnt



   - Perceptron (linear discriminant function is learnt)
   - Support vector machine (SVM) (linear discriminant function is learnt)
   - Neural networks (when the discriminant function is nonlinear)

8

Classification Using Linear Regression

- Given:-Training data: $\mathcal{D} = \{\mathbf{x}_n, \mathbf{y}_n\}_{n=1}^{N}$, $\mathbf{x}_n \in \mathbb{R}^d$ and $\mathbf{y}_n \in \mathbb{R}^M$
  - $\mathbf{x}_n$ is input vector (d dependent variable)
  - There are M classes, represented by M indicator variables
  - $\mathbf{y}_n$ is response vector (dependent variables) which is M-dimensional binary vector i.e. one of the M values is 1
  - For N examples, $\mathbf{X}$ is data matrix of size N x (d+1) and $\mathbf{Y}$ is response matrix of size N x M
- Linear regression on response vector: $\hat{\mathbf{W}} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{Y}$
  - $\hat{\mathbf{W}}$ is of the size (d+1) x M

    $$\hat{\mathbf{W}} = \left[\hat{\mathbf{w}}_1, \hat{\mathbf{w}}_2, ..., \hat{\mathbf{w}}_M\right]$$
  - Each column of $\hat{\mathbf{W}}$ is (d+1) coefficients corresponding to a class

9

Classification Using Linear Regression

- For any test example $\mathbf{x}$, the discriminant value for class i is:

$$f_i(\mathbf{x}, \hat{\mathbf{w}}_i) = \hat{\mathbf{w}}_i^\top \mathbf{x} = \sum_{j=0}^{d} \hat{w}_{ij} x_i$$

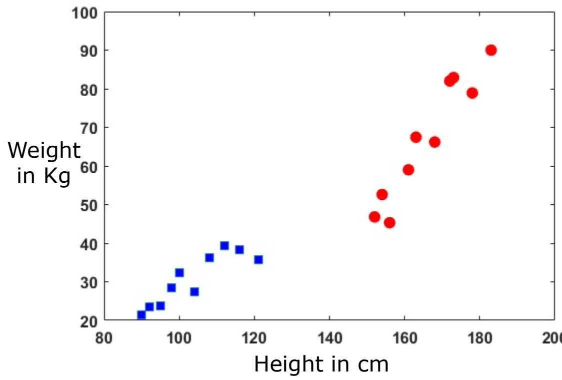
Class label for $\mathbf{x}$ = $\underset{i}{\arg\max}\, f_i(\mathbf{x}, \hat{\mathbf{w}}_i) \qquad i = 1, 2, ..., M$

10

Illustration of Classification using Linear Regression

Height	Weight	Class y1	Class y2
90	21.5	1	0
95	23.67	1	0
100	32.45	1	0
116	38.21	1	0
98	28.43	1	0
108	36.32	1	0
104	27.38	1	0
112	39.28	1	0
121	35.8	1	0
92	23.56	1	0
152	46.8	0	1
178	78.9	0	1
163	67.45	0	1
173	82.9	0	1
154	52.6	0	1
168	66.2	0	1
183	90	0	1
172	82	0	1
156	45.3	0	1
161	59	0	1

- Number of training examples (N) = 20
- Dimension of a training example = 2
- Number of classes: 2
- Each output variable is a 2-dimensional binary vector
- Class: Child (C1) Adult (C2)



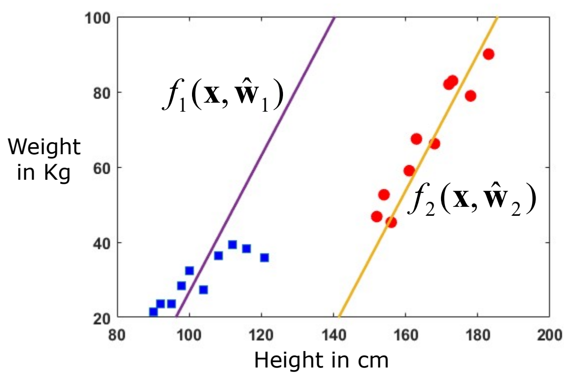
11

Illustration of Classification using Linear Regression

Height	Weight	Class y1	Class y2
90	21.5	1	0
95	23.67	1	0
100	32.45	1	0
116	38.21	1	0
98	28.43	1	0
108	36.32	1	0
104	27.38	1	0
112	39.28	1	0
121	35.8	1	0
92	23.56	1	0
152	46.8	0	1
178	78.9	0	1
163	67.45	0	1
173	82.9	0	1
154	52.6	0	1
168	66.2	0	1
183	90	0	1
172	82	0	1
156	45.3	0	1
161	59	0	1

- Training: $\hat{\mathbf{W}} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{Y}$
- $\mathbf{X}$ is data matrix of size 20×3
- $\mathbf{Y}$ is response matrix of size 20×2

$$\hat{\mathbf{W}} = [\hat{\mathbf{w}}_1 \quad \hat{\mathbf{w}}_2] = \begin{bmatrix} 2.8897 & -1.8897 \\ -0.0222 & 0.0222 \\ 0.0122 & -0.0122 \end{bmatrix}$$

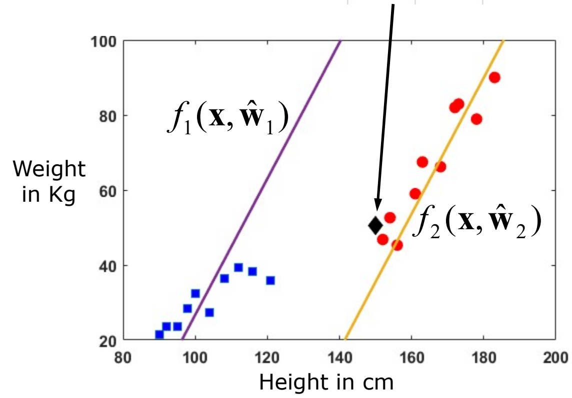


12

Illustration of Classification using Linear Regression

Height	Weight	Class y1	Class y2
90	21.5	1	0
95	23.67	1	0
100	32.45	1	0
116	38.21	1	0
98	28.43	1	0
108	36.32	1	0
104	27.38	1	0
112	39.28	1	0
121	35.8	1	0
92	23.56	1	0
152	46.8	0	1
178	78.9	0	1
163	67.45	0	1
173	82.9	0	1
154	52.6	0	1
168	66.2	0	1
183	90	0	1
172	82	0	1
156	45.3	0	1
161	59	0	1

Test Example: 150 | 50.6



$f_1(\mathbf{x}, \hat{\mathbf{w}}_1) = 0.1842$ $\quad f_2(\mathbf{x}, \hat{\mathbf{w}}_2) = 0.8158$

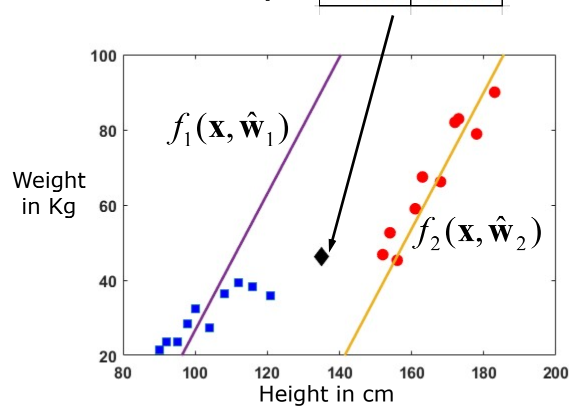
- Class: Adult (C2)

13

Illustration of Classification using Linear Regression

Height	Weight	Class y1	Class y2
90	21.5	1	0
95	23.67	1	0
100	32.45	1	0
116	38.21	1	0
98	28.43	1	0
108	36.32	1	0
104	27.38	1	0
112	39.28	1	0
121	35.8	1	0
92	23.56	1	0
152	46.8	0	1
178	78.9	0	1
163	67.45	0	1
173	82.9	0	1
154	52.6	0	1
168	66.2	0	1
183	90	0	1
172	82	0	1
156	45.3	0	1
161	59	0	1

Test Example: 135 | 46.29



$f_1(\mathbf{x}, \hat{\mathbf{w}}_1) = 0.4639$ $\quad f_2(\mathbf{x}, \hat{\mathbf{w}}_2) = 0.5361$

- Class: Adult (C2)

14

Classification Using Linear Regression

- Dependent variable is categorical (indicator variable)
- Output is multiple outputs (multiple dependent variables)
- If the input $\mathbf{x}$ belongs to C_i, then y_i is 1
- The expected output for $\mathbf{x}$ should be close to 1
- During linear regression for classification, we are trying to predict the expected output value
- In other way, we are trying to predict probability of class

$$\mathrm{E}[y_i \mid \mathbf{x}] = P(y_i = C_i \mid \mathbf{x})$$

- This is the ideal situation
- Linear regression gives the hope of getting this
- The notion of predicting probability of class is given nicely by *logistic regression*

15

Logistic Regression

Logistic Regression

- Requirement: The discriminant function $f_i(\mathbf{x},\mathbf{w}_i)$ should give the probability of class C_i

$$\boxed{\mathrm{E}[y_i \mid \mathbf{x}] = P(y_i = C_i \mid \mathbf{x})}$$

- Look for some kind of transformation of probability and fit that
- Logit transformation: $\log\left(\frac{P(\mathbf{x})}{1-P(\mathbf{x})}\right)$
- *2-class classification*:
  - Class label: 0 or 1
  - $P(\mathbf{x})$ is $P(C_i=1|\mathbf{x})$ i.e. probability that output is 1 given input (probability of success)
  - $1\text{-}P(\mathbf{x})$ is $P(C_i=0|\mathbf{x})$ i.e. probability of failure

17

Logistic Regression

- Logit function: Log of odds function
- Odds function: $\frac{P(\mathbf{x})}{1-P(\mathbf{x})}$
  - Probability of success divided by the probability of failure
- Fit a linear model to logit function:

$$\log\left(\frac{P(\mathbf{x})}{1-P(\mathbf{x})}\right) = w_0 + w_1 x_1 + \ldots + w_d x_d = \mathbf{w}^\mathsf{T}\hat{\mathbf{x}}$$

where $\mathbf{w} = [w_0, w_1, \ldots, w_d]^\mathsf{T}$ and $\hat{\mathbf{x}} = [1, x_1, \ldots, x_d]^\mathsf{T}$

  - For 1-dimensional (d=1) space, x

$$\log\left(\frac{P(x)}{1-P(x)}\right) = w_0 + w_1 x \qquad \frac{P(x)}{1-P(x)} = e^{(w_0 + w_1 x)}$$

18

Logistic Regression

- Logit function: Log of odds function
- Odds function: $\boxed{\dfrac{P(\mathbf{x})}{1-P(\mathbf{x})}}$
  - Probability of success divided by the probability of failure
- Fit a linear model to logit function:

$$\boxed{\log\left(\frac{P(\mathbf{x})}{1-P(\mathbf{x})}\right)=\mathbf{w}^{\top}\hat{\mathbf{x}}}$$

where $\mathbf{w}=[w_0, w_1, ..., w_d]^{\top}$
and $\hat{\mathbf{x}}=[1, x_1, ..., x_d]^{\top}$

  - For 1-dimensional (d=1) space, x

$$\frac{P(x)}{1-P(x)}=e^{(w_0+w_1 x)}$$

$$P(x)=\frac{e^{(w_0+w_1 x)}}{1+e^{(w_0+w_1 x)}}=\frac{1}{1+e^{-(w_0+w_1 x)}}$$

19

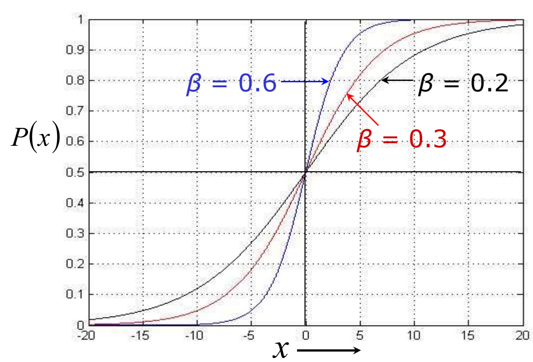
Logistic Regression

$$P(x)=\frac{1}{1+e^{-(w_0+w_1 x)}}$$

- This function is a sigmoidal function, specifically called as logistic function
- Logistic function:

$$P(x)=\frac{1}{1+e^{-(w_0+w_1 x)}}$$

$$P(x)=\frac{1}{1+e^{-(\beta x)}}$$



20

Logistic Regression

- Logit function: Log of odds function
- Odds function: $\boxed{\dfrac{P(\mathbf{x})}{1-P(\mathbf{x})}}$
  – Probability of success divided by the probability of failure
- Fit a linear model to logit function: $\boxed{\log\left(\dfrac{P(\mathbf{x})}{1-P(\mathbf{x})}\right) = \mathbf{w}^\top \hat{\mathbf{x}}}$
  – For d-dimensional space, $\mathbf{x}=[x_1, x_2, \dots, x_d]^\top$

$$\log\left(\frac{P(\mathbf{x})}{1-P(\mathbf{x})}\right) = w_0 + w_1x_1 + \dots + w_dx_d = \mathbf{w}^\top \hat{\mathbf{x}}$$

where $\mathbf{w} = [w_0, w_1, \dots, w_d]^\top$ and $\hat{\mathbf{x}} = [1, x_1, \dots, x_d]^\top$

$$\frac{P(\mathbf{x})}{1-P(\mathbf{x})} = e^{(\mathbf{w}^\top \hat{\mathbf{x}})}$$

21

Logistic Regression

- Logit function: Log of odds function
- Odds function: $\boxed{\dfrac{P(\mathbf{x})}{1-P(\mathbf{x})}}$
  – Probability of success divided by the probability of failure
- Fit a linear model to logit function: $\boxed{\log\left(\dfrac{P(\mathbf{x})}{1-P(\mathbf{x})}\right) = \mathbf{w}^\top \hat{\mathbf{x}}}$
  – For d-dimensional space, $\mathbf{x}=[x_1, x_2, \dots, x_d]^\top$

$$\frac{P(\mathbf{x})}{1-P(\mathbf{x})} = e^{(\mathbf{w}^\top \hat{\mathbf{x}})}$$

where $\mathbf{w} = [w_0, w_1, \dots, w_d]^\top$ and $\hat{\mathbf{x}} = [1, x_1, \dots, x_d]^\top$

$$P(\mathbf{x}) = \frac{e^{(\mathbf{w}^\top \hat{\mathbf{x}})}}{1+e^{(\mathbf{w}^\top \hat{\mathbf{x}})}}$$

$$P(\mathbf{x}) = \frac{1}{1+e^{-(\mathbf{w}^\top \hat{\mathbf{x}})}}$$

If $P(\mathbf{x}) \geq 0.5$ then $\mathbf{x}$ is assigned to C_2
If $P(\mathbf{x}) < 0.5$ then $\mathbf{x}$ is assigned to C_1

22

Estimation of Parameter in Logistic Regression

- Criterion considered is different than linear regression to estimate the parameter
- Optimize the likelihood of data
- As that goal is to model the probability of class, we are maximizing the likelihood of data
- **Maximum likelihood (ML) method of parameter estimation**
- Given:- Training data: $\mathcal{D} = \{\mathbf{x}_n, y_n\}_{n=1}^{N}, \ \mathbf{x}_n \in \mathbb{R}^d$ and $y_n \in \{1,0\}$
- Data of a class is represented by parameter vector: $\mathbf{w}=[w_0, w_1, \ldots, w_d]^{\mathsf{T}}$ (parameter of linear function)
- Unknown: $\mathbf{w}$
- Likelihood of $\mathbf{x}_n$: $P(\mathbf{x}_n \mid \mathbf{w}) = \underbrace{P(\mathbf{x}_n)^{y_n}}_{\text{Probability that } \mathbf{x} \text{ has label 1}} \underbrace{(1-P(\mathbf{x}_n))^{(1-y_n)}}_{\text{Probability that } \mathbf{x} \text{ has label 0}}$

23

Estimation of Parameter in Logistic Regression

- Different criterion than linear regression to estimate the parameter
- Optimize the likelihood of data
- As that goal is to model the probability of class, we are maximizing the likelihood of data
- **Maximum likelihood (ML) method of parameter estimation**
- Given:- Training data: $\mathcal{D} = \{\mathbf{x}_n, y_n\}_{n=1}^{N}, \ \mathbf{x}_n \in \mathbb{R}^d$ and $y_n \in \{1,0\}$
- Data of a class is represented by parameter vector: $\mathbf{w}=[w_0, w_1, \ldots, w_d]^{\mathsf{T}}$ (parameter of linear function)
- Unknown: $\mathbf{w}$
- Likelihood of $\mathbf{x}_n$: $P(\mathbf{x}_n \mid \mathbf{w}) = \boxed{P(\mathbf{x}_n)^{y_n} (1-P(\mathbf{x}_n))^{(1-y_n)}}$ Binomial distribution (Bernoulli Distribution)
- Total data likelihood: $P(\mathcal{D} \mid \mathbf{w}) = \prod_{n=1}^{N} P(\mathbf{x}_n \mid \mathbf{w})$

24

Estimation of Parameter in Logistic Regression

- Different criterion than linear regression to estimate the parameter
- Optimize the likelihood of data
- As that goal is to model the probability of class, we are maximizing the likelihood of data
- **Maximum likelihood (ML) method of parameter estimation**
- Given:- Training data: $\mathcal{D} = \{\mathbf{x}_n, y_n\}_{n=1}^{N}, \ \mathbf{x}_n \in \mathbb{R}^d$ and $y_n \in \{1,0\}$
- Data of a class is represented by parameter vector: $\mathbf{w}=[w_0, w_1, \ldots, w_d]^{\mathsf{T}}$ (parameter of linear function)
- Unknown: $\mathbf{w}$
- Likelihood of $\mathbf{x}_n$: $P(\mathbf{x}_n \mid \mathbf{w}) = P(\mathbf{x}_n)^{y_n}(1-P(\mathbf{x}_n))^{(1-y_n)}$
- Total data likelihood: $P(\mathcal{D} \mid \mathbf{w}) = \prod_{n=1}^{N} P(\mathbf{x}_n)^{y_n}(1-P(\mathbf{x}_n))^{(1-y_n)}$

25

Estimation of Parameter in Logistic Regression

- Total data log likelihood:

$$l(\mathbf{w}) = \ln\big(P(\mathcal{D} \mid \mathbf{w})\big)$$

$$l(\mathbf{w}) = \sum_{n=1}^{N} y_n \ln(P(\mathbf{x}_n)) + (1-y_n)\ln(1-P(\mathbf{x}_n))$$

- Choose the parameters for which the total data log likelihood is maximum:

$$\mathbf{w}_{\text{ML}} = \arg\max_{\mathbf{w}} l(\mathbf{w})$$

- Cost function for optimization:

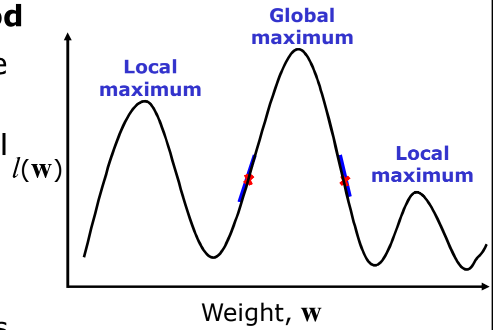
$$l(\mathbf{w}) = \sum_{n=1}^{N} y_n \ln(P(\mathbf{x}_n)) + (1-y_n)\ln(1-P(\mathbf{x}_n))$$

- Conditions for optimality: $\frac{\partial\, l(\mathbf{w})}{\partial \mathbf{w}} = \mathbf{0}$
- Unfortunately, solving this, no closed form expression for $\mathbf{w}$ is obtained
- Solution: Gradient accent method

26

Estimation of Parameter in Logistic Regression

- **Gradient accent method**
- It is an iterative procedure
- We start with an initial value for $\mathbf{w}$
- At each iteration:
  - Estimate change in $\mathbf{w}$
  - The change in $\mathbf{w}$ ($\Delta\mathbf{w}$) is proportional to the slope (gradient) of the likelihood surface
  - Then, the $\mathbf{w}$ is updated using $\Delta\mathbf{w}$
- This indicate, we move in the positive slope of the likelihood surface, likelihood is maximum in each iteration



$$\Delta\mathbf{w} = \frac{\partial\, l(\mathbf{w})}{\partial\mathbf{w}}$$

27

Estimation of Parameter in Logistic Regression – Gradient Accent Method

- Given a training dataset, the goal is to maximize the likelihood function with respect to the parameters of linear function
  1. Initialize the $\mathbf{w}$
     - Evaluate the initial value of the log likelihood, $l(\mathbf{w})$
  2. Determine the change in $\mathbf{w}$ ($\Delta\mathbf{w}$): $\Delta\mathbf{w} = \dfrac{\partial\, l(\mathbf{w})}{\partial\mathbf{w}}$
  3. Update the $\mathbf{w}$: $\mathbf{w} = \mathbf{w} + \Delta\mathbf{w}$
  4. Evaluate the log likelihood and check for convergence of the log likelihood
     - If the convergence criterion is not satisfied repeat steps 2 to 4
- Convergence criterion: Difference between log likelihoods of successive iterations fall below a threshold (E.g. threshold=10^{-3})

28

Illustration of Classification using Logistic Regression

Height	Weight	Class
90	21.5	0
95	23.67	0
100	32.45	0
116	38.21	0
98	28.43	0
108	36.32	0
104	27.38	0
112	39.28	0
121	35.8	0
92	23.56	0
152	46.8	1
178	78.9	1
163	67.45	1
173	82.9	1
154	52.6	1
168	66.2	1
183	90	1
172	82	1
156	45.3	1
161	59	1

- Number of training examples (N) = 20
- Dimension of a training example = 2
- Class label attribute is 3rd dimension
- Class:
  - Child (0)
  - Adult (1)

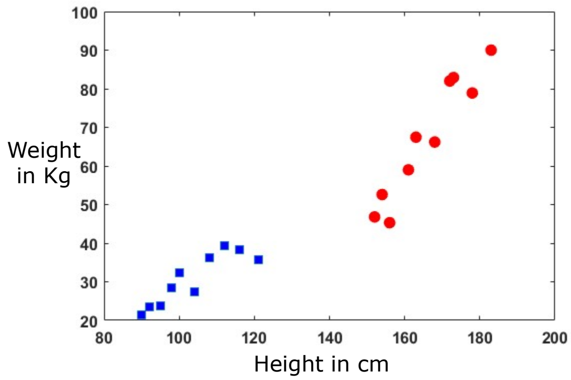


Illustration of Classification using Logistic Regression

Height	Weight	Class
90	21.5	0
95	23.67	0
100	32.45	0
116	38.21	0
98	28.43	0
108	36.32	0
104	27.38	0
112	39.28	0
121	35.8	0
92	23.56	0
152	46.8	1
178	78.9	1
163	67.45	1
173	82.9	1
154	52.6	1
168	66.2	1
183	90	1
172	82	1
156	45.3	1
161	59	1

- Training:

$$\mathbf{w} = \begin{bmatrix} -378.2085 \\ 2.2065 \\ 1.8818 \end{bmatrix}$$

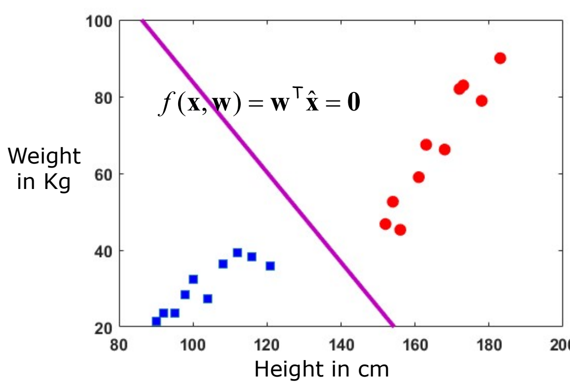
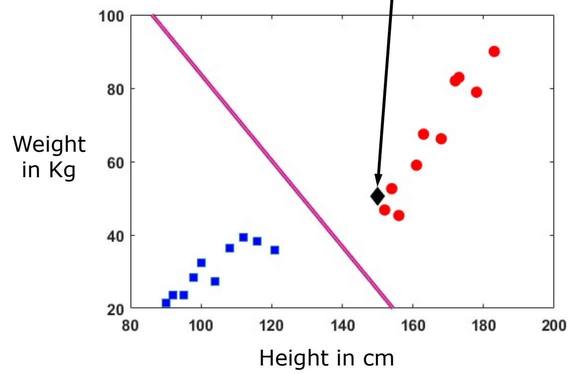


Illustration of Classification using Linear Regression

Height	Weight	Class
90	21.5	0
95	23.67	0
100	32.45	0
116	38.21	0
98	28.43	0
108	36.32	0
104	27.38	0
112	39.28	0
121	35.8	0
92	23.56	0
152	46.8	1
178	78.9	1
163	67.45	1
173	82.9	1
154	52.6	1
168	66.2	1
183	90	1
172	82	1
156	45.3	1
161	59	1

Test Example:

150	50.6



$$\mathbf{w}^{\mathsf{T}}\hat{\mathbf{x}} = 47.9851 \qquad P(\mathbf{x}) = \frac{1}{1+e^{-(\mathbf{w}^{\mathsf{T}}\hat{\mathbf{x}})}} = 1$$

- Class: Adult (C2)

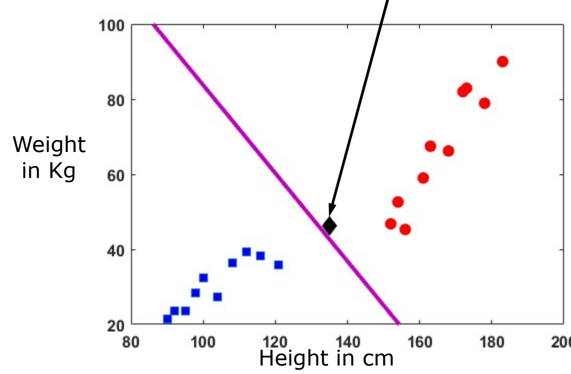
31

Illustration of Classification using Linear Regression

Height	Weight	Class
90	21.5	0
95	23.67	0
100	32.45	0
116	38.21	0
98	28.43	0
108	36.32	0
104	27.38	0
112	39.28	0
121	35.8	0
92	23.56	0
152	46.8	1
178	78.9	1
163	67.45	1
173	82.9	1
154	52.6	1
168	66.2	1
183	90	1
172	82	1
156	45.3	1
161	59	1

Test Example:

135	46.29



$$\mathbf{w}^{\mathsf{T}}\hat{\mathbf{x}} = 6.7771 \qquad P(\mathbf{x}) = \frac{1}{1+e^{-(\mathbf{w}^{\mathsf{T}}\hat{\mathbf{x}})}} = 0.9$$

- Class: Adult (C2)

32

Logistic Regression

- Logistic regression is a linear classifier
- Logistic regression looks simple, but yields a very powerful classifier
- It is used not just building classifier, but also used in sensitivity analysis
- Logistic regression is used to identify how each attribute contribute to output
  - How much each attribute is important for predicting class label
- Perform logistic regression and observe $\mathbf{w}$
- The value of each element of $\mathbf{w}$ indicate how much each attribute is contributing to the output

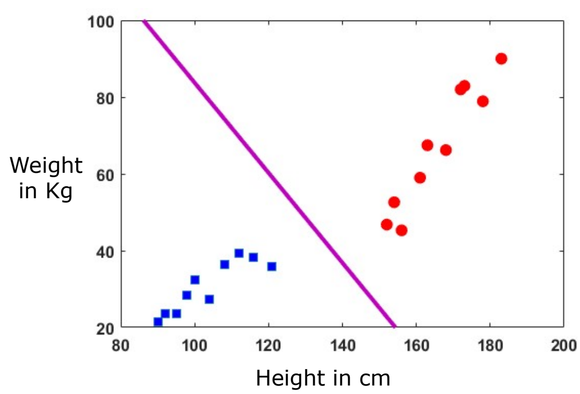
33

Illustration of Sensitivity Analysis using Logistic Regression

Height	Weight	Class
90	21.5	0
95	23.67	0
100	32.45	0
116	38.21	0
98	28.43	0
108	36.32	0
104	27.38	0
112	39.28	0
121	35.8	0
92	23.56	0
152	46.8	1
178	78.9	1
163	67.45	1
173	82.9	1
154	52.6	1
168	66.2	1
183	90	1
172	82	1
156	45.3	1
161	59	1

- Training:

$$\mathbf{w} = \begin{bmatrix} -378.2085 \\ 2.2065 \\ 1.8818 \end{bmatrix} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}$$



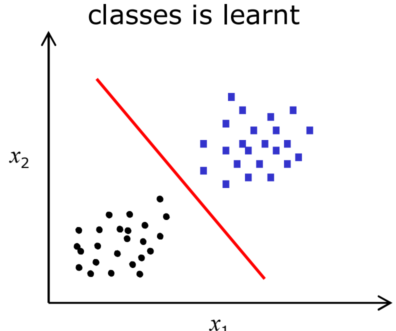
- Both the attributes are equally important

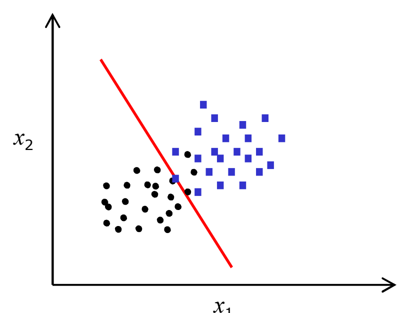
34

Discriminative Learning Methods for Classification

Two classes of Approaches for Linear Classification

1. Modeling a discriminating function:
   - Linear regression and Logistic regression
2. Directly learn a discriminant function (hyperplane):
   - Classic method: Discriminant function between the classes is learnt



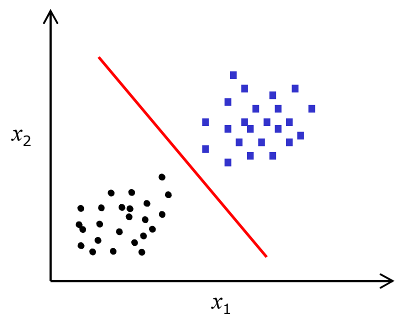


   - Perceptron (linear discriminant function is learnt)
   - Neural networks (when the discriminant function is nonlinear)

36

Discriminative Learning Methods

- Learn the surface that better separates the region of classes
- Learning discriminant function: Learns a function that maps input data to output
- Linear discriminant function:

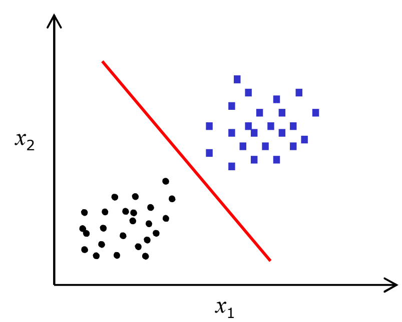


Linear Discriminant Function

- Regions of two classes are separable by a linear surface (line, plane or hyperplane)
- 2-dimensional space: The decision boundary is a **line** specified by

$$w_1 x_1 + w_2 x_2 + w_0 = 0$$

$$x_2 = -\frac{w_1}{w_2} x_1 - \frac{w_0}{w_2}$$



- d-dimensional space: The decision surface is a **hyperplane** specified by

$$w_d x_d + \ldots. + w_2 x_2 + w_1 x_1 + w_0 = \sum_{i=0}^{d} w_i x_i = \mathbf{w}^\top \hat{\mathbf{x}} = 0$$

where $\mathbf{w} = [w_0, w_1, \ldots, w_d]^\top$ and $\hat{\mathbf{x}} = [1, x_1, \ldots, x_d]^\top$

Discriminant Function of a Hyperplane

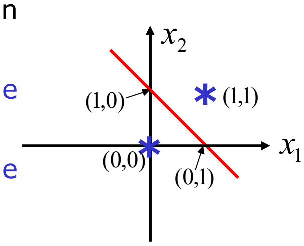
- The discriminant function of a hyperplane:

$$g(\mathbf{x}) = \sum_{i=1}^{d} w_i x_i + w_0 = \mathbf{w}^\mathsf{T}\mathbf{x} + w_0$$

- For any point the lies on the hyperplane

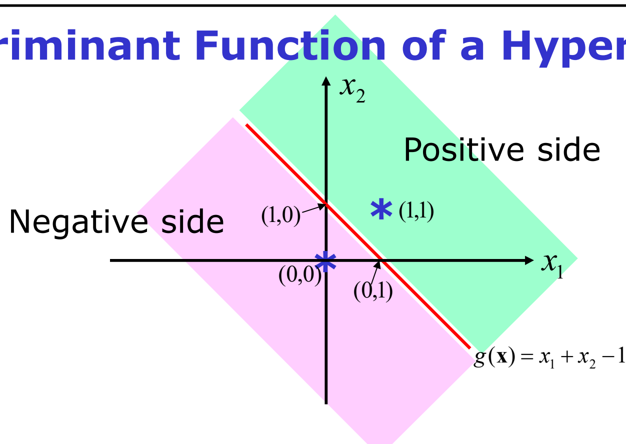
$$g(\mathbf{x}) = \sum_{i=1}^{d} w_i x_i + w_0 = \mathbf{w}^\mathsf{T}\mathbf{x} + w_0 = 0$$

- Example:
  - Consider a straight line with its equation as $x_2+x_1-1=0$
  - Discriminant function of the straight line is $g(\mathbf{x})=x_2+x_1-1$
  - For points $(1,0)$ and $(0,1)$ that lie on this straight line $g(\mathbf{x})=0$
  - For the point $(0,0)$, $g(\mathbf{x})=-1$ i.e. the value of $g(\mathbf{x})$ is negative
  - For the point $(1,1)$, $g(\mathbf{x})=+1$ i.e. the value of $g(\mathbf{x})$ is positive



39

Discriminant Function of a Hyperplane

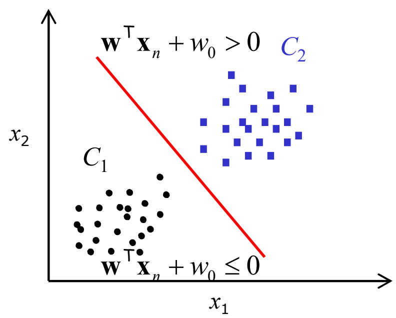


- A hyperplane has a positive side and a negative side
  - For any point on the positive side, the value of discriminant function, $g(\mathbf{x})$, is positive
  - For any point on the negative side, the value of discriminant function , $g(\mathbf{x})$, is negative

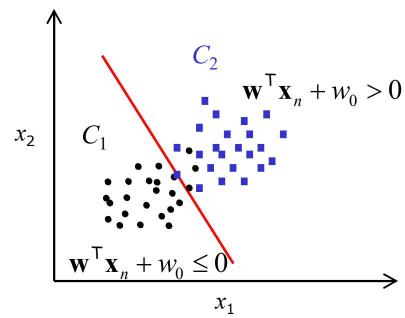
40

Perceptron Learning

- Given - training data: $\mathcal{D} = \{\mathbf{x}_n, y_n\}_{n=1}^{N},\ \mathbf{x}_n \in \mathbb{R}^d$ and $y_n \in \{+1, -1\}$
- Goal: To estimate parameter vector $\mathbf{w}=[w_0, w_1, \ldots, w_d]^\mathsf{T}$
  - such that linear function (hyperplane) is places between the training data of two classes so that training error (classification error) is minimum



OR



41

Perceptron Learning

- Given - training data: $\mathcal{D} = \{\mathbf{x}_n, y_n\}_{n=1}^{N},\ \mathbf{x}_n \in \mathbb{R}^d$ and $y_n \in \{+1, -1\}$

1. Initialize the $\mathbf{w}$
2. Choose a training example $\mathbf{x}_n$
3. Update the $\mathbf{w}$, if $\mathbf{x}_n$ is misclassified

   $\mathbf{w} = \mathbf{w} + \eta\, \mathbf{x}_n,$ for $\mathbf{w}^\mathsf{T}\mathbf{x}_n + w_0 \le 0$ and $\mathbf{x}_n \in C_2(+1)$

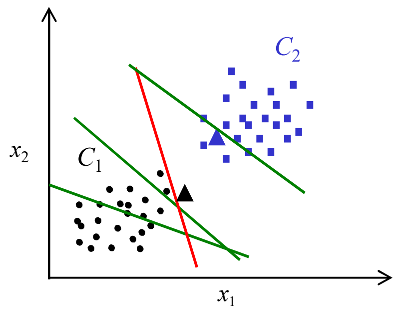
   $\mathbf{w} = \mathbf{w} - \eta\, \mathbf{x}_n,$ for $\mathbf{w}^\mathsf{T}\mathbf{x}_n + w_0 > 0$ and $\mathbf{x}_n \in C_1(-1)$
   - Here η is a positive, learning rate parameter
   - Increment the misclassification count by 1
4. Repeat steps 2 and 3 till all the training examples are presented
5. Repeat steps 2 to 4 by setting misclassification count to 0, till the convergence criterion is not satisfied

- Convergence criterion:
  - Total misclassification count is 0 **OR**
  - Total misclassification count is minimum (falls below threshold)

42

Perceptron Learning

- Training:



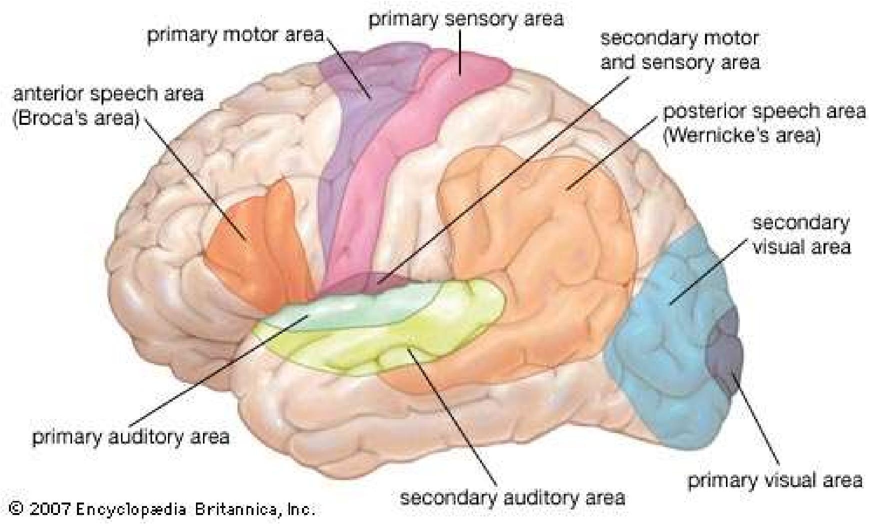
- **Test phase:**
- Classification of a test pattern $\mathbf{x}$ using the weights $\mathbf{w}$ obtained by training the model:
  - If $\mathbf{w}^{\mathsf{T}}\mathbf{x} + w_0 > 0$ then $\mathbf{x}$ is assigned to C_2
  - If $\mathbf{w}^{\mathsf{T}}\mathbf{x} + w_0 \leq 0$ then $\mathbf{x}$ is assigned to C_1

43

Discriminative Learning Methods for Classification:

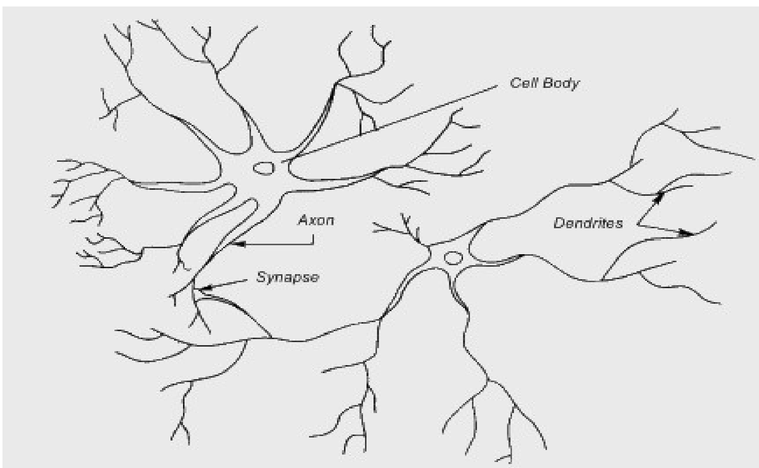
Neural Networks

Human Brain



45

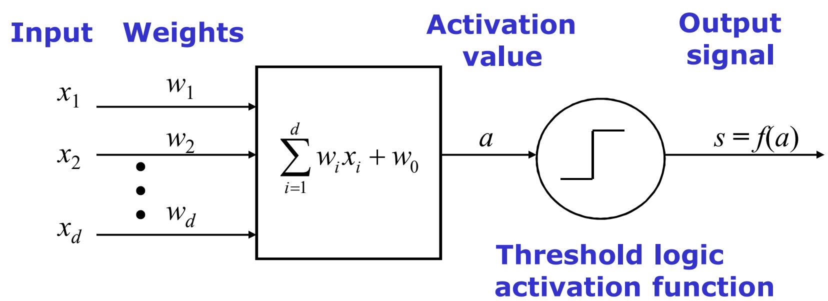
Biological Neural Networks



- Several neurons are connected to one another to form a neural network or a layer of a neural network

46

Neuron with Threshold Logic Activation Function



$$s = 1 \quad \text{if } f(a) > 0$$
$$s = 0 \quad \text{if } f(a) \leq 0$$

- McCulloch-Pitts Neuron [1]

[1] W.S.McCulloch and W.Pitts. A logival calculus of the ideas imminent in nervous activity. 1943.

47

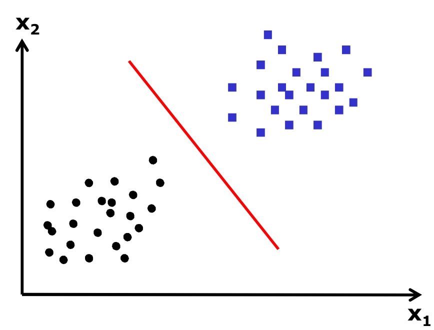
Linearly Separable Classes – Perceptron Model

- Regions of two classes are separable by a linear surface (line, plane or hyperplane)
- Perceptron model that uses a single MuCulloch-Pitts neuron can be trained using the perceptron learning algorithm [2]

Decision surface in a 2-dimensional space is a line:

$$w_1 x_1 + w_2 x_2 + w_0 = 0$$

$$x_2 = -\frac{w_1}{w_2} x_1 - \frac{w_0}{w_2}$$



Decision surface in a d-dimensional space is a hyperplane:

$$\sum_{i=0}^{d} w_i x_i = \mathbf{w}^{\mathrm{T}} \mathbf{x} + w_0 = 0$$

[2] A.G. Ivakhnenko and V.G. Lapa. Cybernetic predicting devices. 1965.

48

Perceptron Learning

- Perceptron learning rule for weight update:

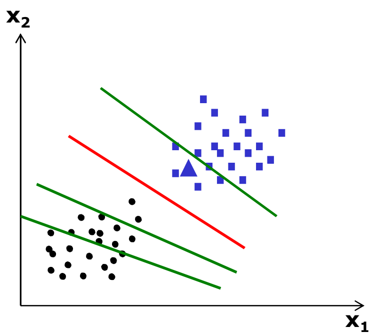
  At Step m, choose a training example $\mathbf{x}_n$

  $$\mathbf{w} = \mathbf{w} + \eta\, \mathbf{x}_n, \text{ for } \mathbf{w}^{\mathsf{T}}\mathbf{x}_n + w_0 \le 0 \text{ and } \mathbf{x}_n \in C_2(+1)$$

  $$\mathbf{w} = \mathbf{w} - \eta\, \mathbf{x}_n, \text{ for } \mathbf{w}^{\mathsf{T}}\mathbf{x}_n + w_0 > 0 \text{ and } \mathbf{x}_n \in C_1(-1)$$

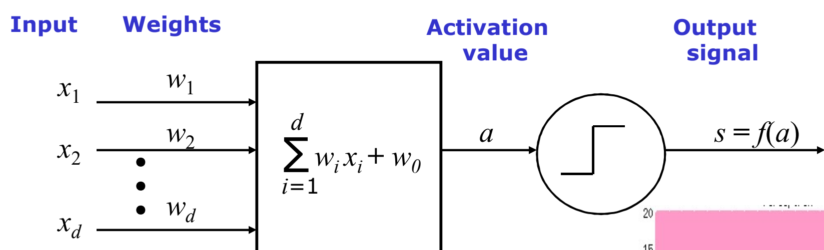
  Here η is the learning rate parameter.

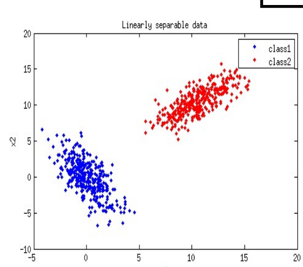
- **Test phase**:
- Classification of a test pattern $\mathbf{x}$ using the weights $\mathbf{w}$ obtained by training the model:
  - If $\mathbf{w}^{\mathsf{T}}\mathbf{x} + w_0 > 0$ then $\mathbf{x}$ is assigned to C_2
  - If $\mathbf{w}^{\mathsf{T}}\mathbf{x} + w_0 \le 0$ then $\mathbf{x}$ is assigned to C_1

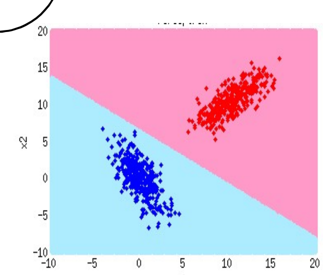


49

Neuron with Threshold Logic Activation Function and Perceptron Learning



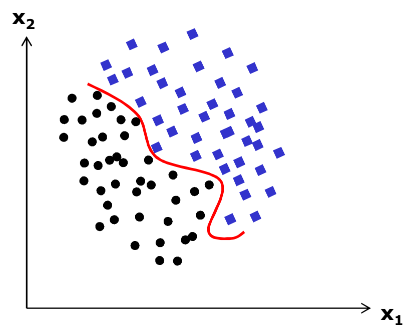




50

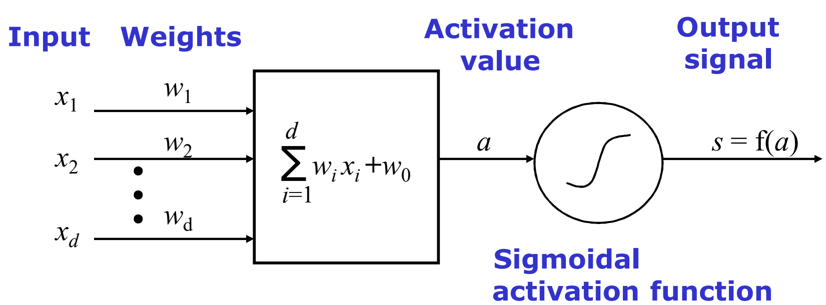
Hard Problems

- Nonlinearly separable classes



51

Neuron with Continuous Activation Function



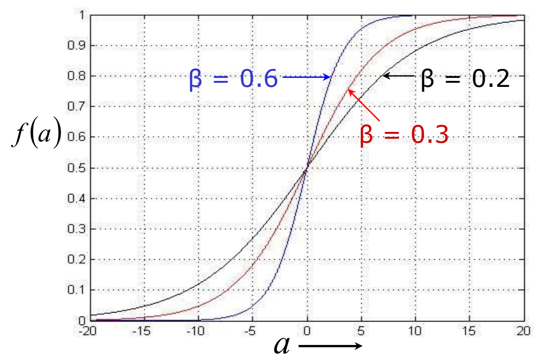
52

Sigmoidal Activation Functions

Logistic function:

$$f(a) = \frac{1}{1+e^{-\beta a}}$$

$$\frac{df(a)}{da} = \beta f(a)\left(1 - f(a)\right)$$



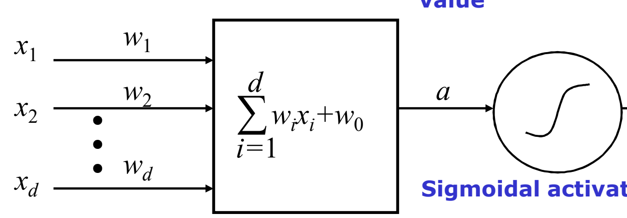
Hyperbolic tangent function:

$$f(a) = \tanh(\beta a)$$

$$\frac{df(a)}{da} = \beta\left(1 - f^2(a)\right)$$

53

Neuron with Continuous Activation Function



$$\sum_{i=1}^{d} w_i x_i + w_0 \qquad s = f(a) = \frac{1}{1+e^{-\beta a}}$$

- Given a set of N input–output pattern pairs $(\mathbf{x}_n, \mathbf{y}_n)$, n=1,2,….N. Here $y_n \in \{0, 1\}$ or $y_n \in \{+1, -1\}$
- Instantaneous error for the nth pattern is given as , $E_n = \frac{1}{2}(y_n - s)^2$
- Change in weight at m^{th} iteration: (Gradient descent technique) $\Delta w_i(m) = -\eta \frac{\partial E_n(m)}{\partial w_i}$

$$\Delta w_i(m) = \eta \delta^o s \qquad w_i(m+1) = w_i(m) + \Delta w_i(m)$$

where $\delta^o = (y_n - s)\frac{df(a_n)}{da_n}$

54

Illustration: Linearly Separable Data

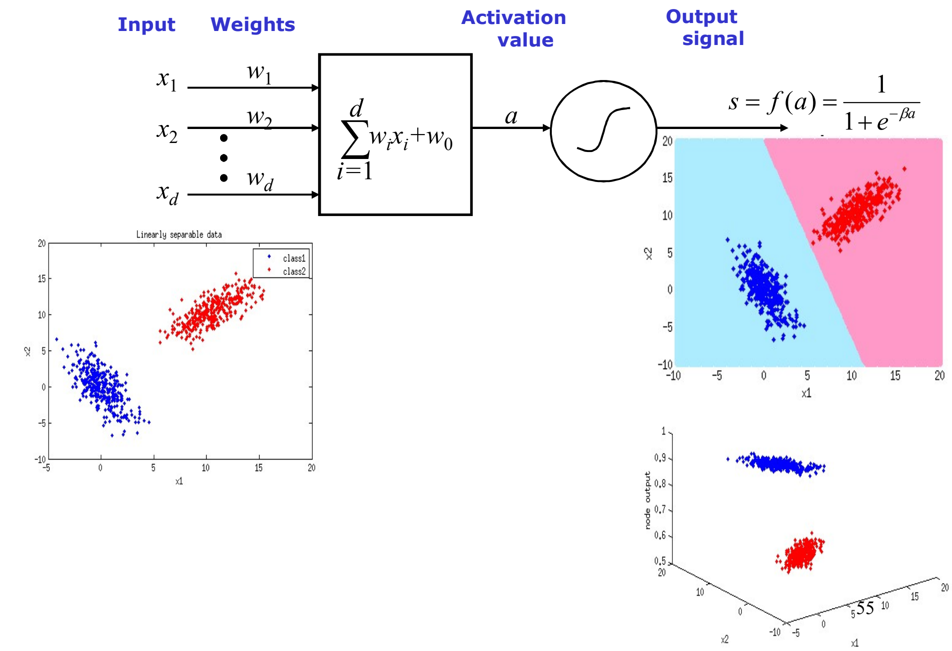


Illustration: Non-Linearly Separable Data

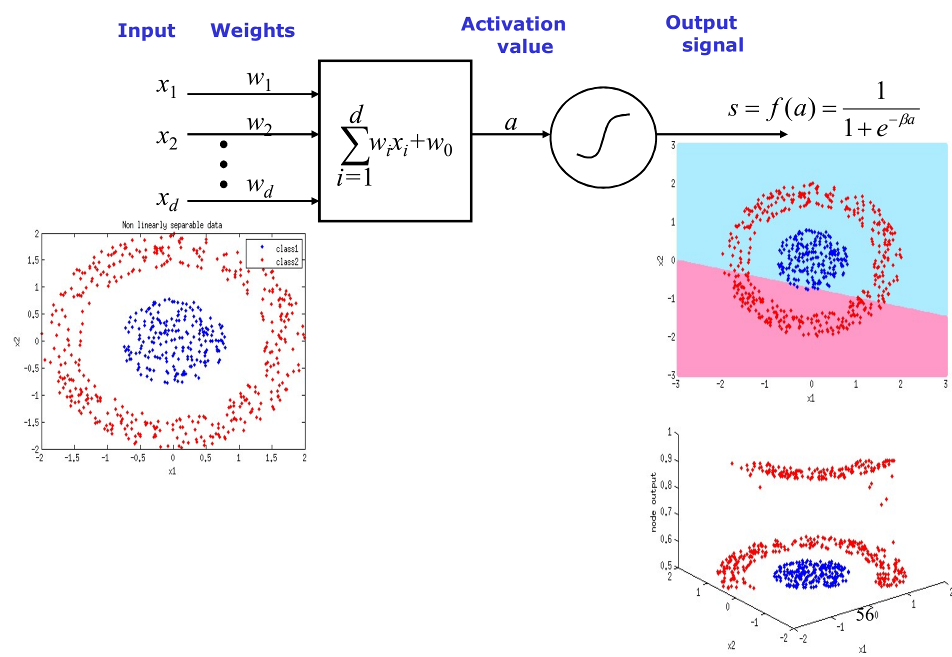
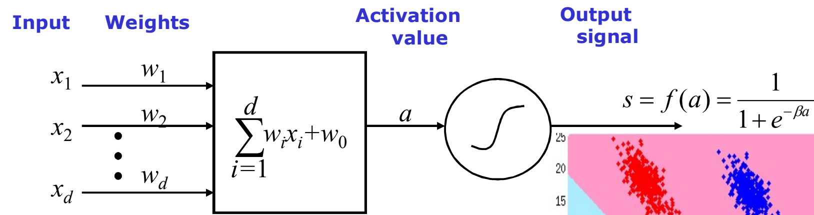
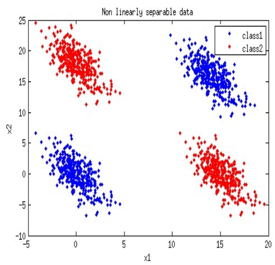
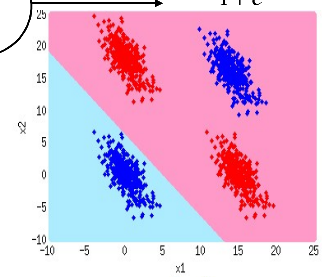
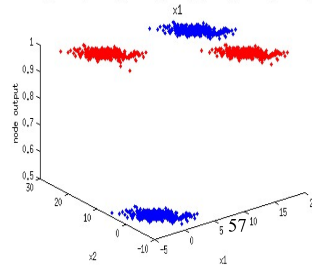


Illustration: Non-Linearly Separable Data









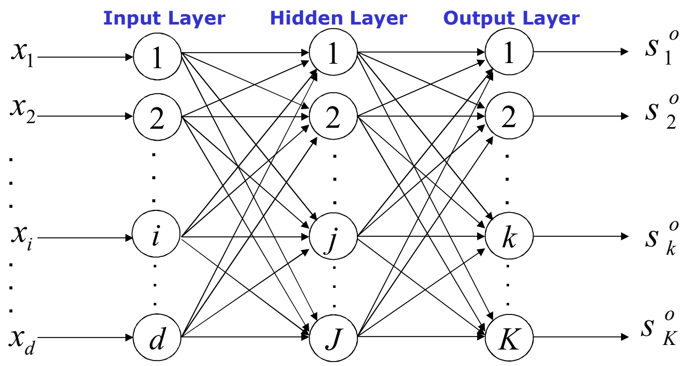
57

Feedforward Neural Networks

58

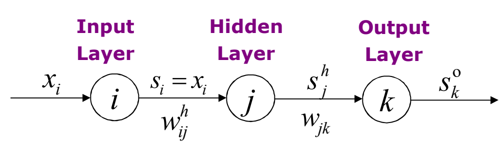
Multilayer Feedforward Neural Network

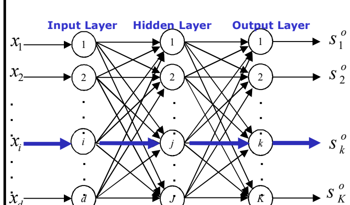
- **Architecture of an MLFFNN**
  - **Input layer: Linear neurons**
  - **Hidden layers (1 or 2 or more): Sigmoidal neurons**
  - **Output layer:**
    - **Linear neurons (for function approximation task)**
    - **Sigmoidal neurons (for pattern classification task)**



59

Multilayer Feedforward Neural Network





$$s_j^h = f_j^h(a_j^h)$$

$$a_j^h = \sum_{i=1}^{d} w_{ij}^h s_i + w_{0j}^h, \text{ where } s_i = x_i$$

$$s_k^o = f_k^o(a_k^o)$$

$$a_k^o = \sum_{j=1}^{J} w_{jk} s_j^h + w_{0k}$$

$$s_k^o = f_k^o\left[\sum_{j=1}^{J} w_{jk} f_j^h\left(\sum_{i=1}^{d} w_{ji}^h s_i + w_{0j}^h\right) + w_{0k}\right]$$

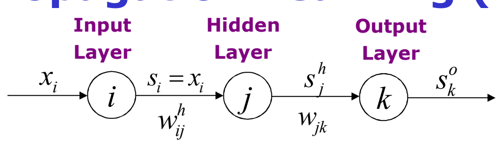
Backpropagation Learning

- Gradient descent method
- Error backpropagation algorithm
- Forward computation:
  - Innerproduct computation
  - Activation function computation
- Backward operation:
  - Error calculation and propagation
  - Modification of weights
- Given a set of N input–output pattern pairs $(\mathbf{x}_n, \mathbf{y}_n)$, n=1,2,….N.
  - $\mathbf{y}_n$ is a K-dimensional binary vector if activation function is logistic function. Only one element is 1 and rest are 0
  - $\mathbf{y}_n$ is a K-dimensional vector with only one element is 1 rest are -1 when the activation function is hyperbolic tangent function
- Instantaneous error for the nth pattern is given as ,

$$E_n = \frac{1}{2}\sum_{k=1}^{K}(y_{nk} - s_k^{o})^2$$

61

Backpropagation Learning (contd.)



- Change in weight at output layer is given by

$$\Delta w_{jk}(m) = -\eta \frac{\partial E(m)}{\partial w_{jk}}$$

$$\Delta w_{jk}(m) = \eta \delta_k^o s_j^h \qquad w_{jk}(m+1) = w_{jk}(m) + \Delta w_{jk}(m)$$

$$\text{where } \delta_k^o = (y_k - s_k^o)\frac{df_k^o(a_k^o)}{da_k^o}$$

- Change in weight at hidden layer is given by

$$\Delta w_{ij}^h(m) = -\eta \frac{\partial E(m)}{\partial w_{ij}^h}$$

$$\Delta w_{ij}^h(m) = \eta \delta_j^h s_i \qquad w_{ij}^h(m+1) = w_{ij}^h(m) + \Delta w_{ij}^h(m)$$

$$\text{where } \delta_j^h = \sum_{k=1}^{K} \delta_k^o w_{jk} \frac{df_j^h(a_j^h)}{da_j^h}$$

62

Mode of Presentation of Patterns

- Stochastic Gradient Descent Method
- **Pattern Mode:**
  - At m^{th} iteration: Weights are updated after the presentation of each pattern.

$$\Delta w_{jk}(m) = -\eta \frac{\partial E(m)}{\partial w_{jk}}$$

  - Epoch: Presentation of all the patterns once (all training examples).

- **Batch Mode:**
  - Weights are updated after the presentation of all the patterns once.

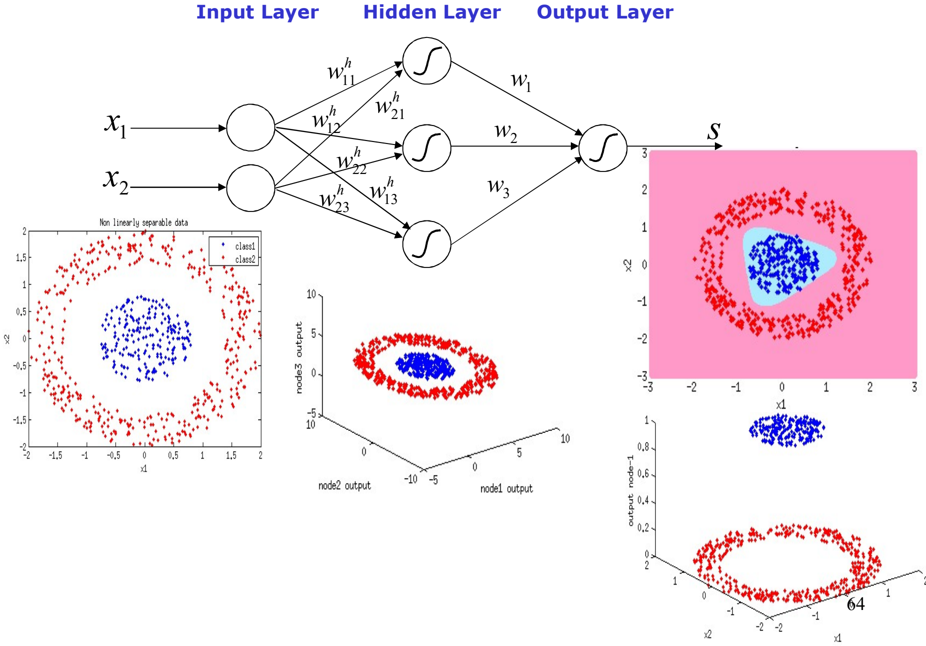
$$\Delta w_{jk}(m) = -\eta \frac{\partial E_{av}}{\partial w_{jk}}$$

$$\text{where } E_{av} = \frac{1}{2N}\sum_{l=1}^{N}\sum_{k=1}^{K}\left(y_{nk} - s_{nk}^{\text{o}}\right)^2$$

63

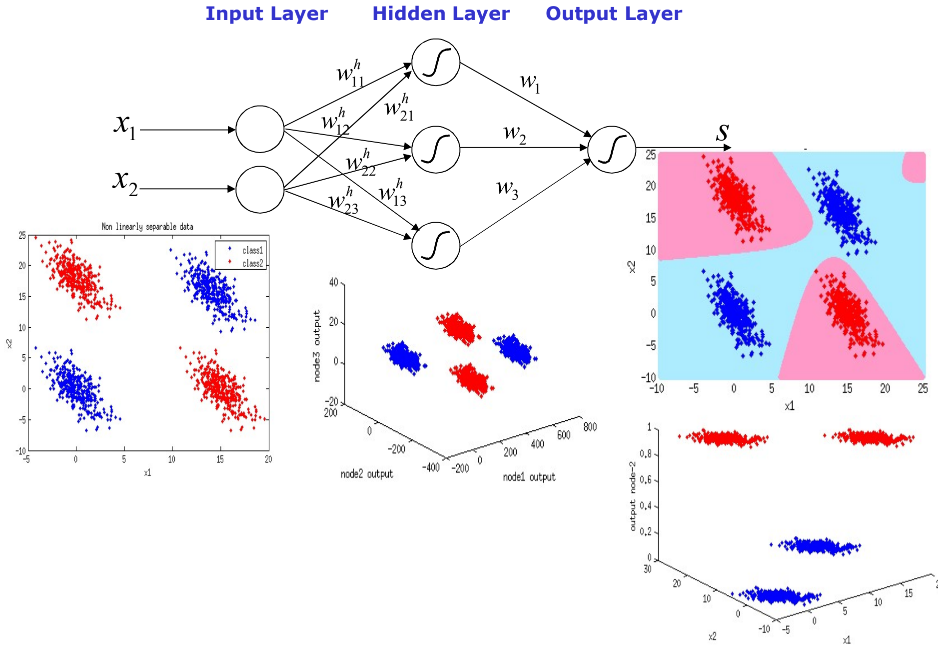
Illustration: Ring Data

Input Layer **Hidden Layer** **Output Layer**



64

Illustration: X-OR Data

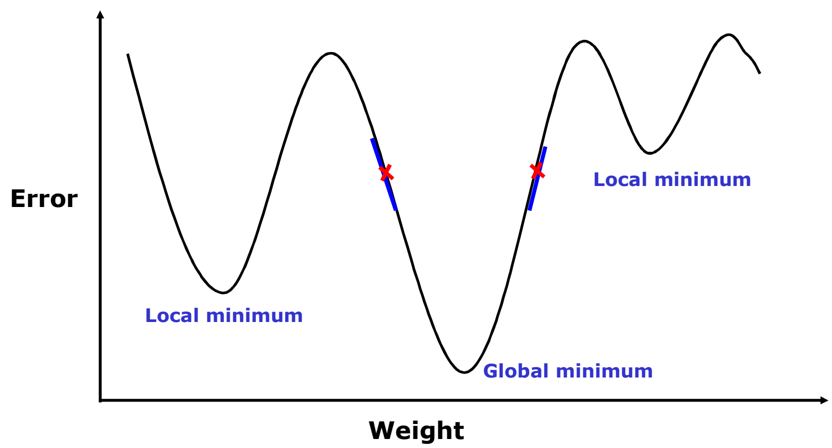


Practical Considerations

- Stopping Criterion:
  - Threshold on average error
  - Threshold on average gradient
- Number of Weights:
  - Depends on number of input nodes, output nodes, hidden nodes and hidden layers
- Number of Hidden Nodes
  - Cross-validation method
- Data Requirements
- Limitations:
  - Slow convergence
  - Local minima problem

66

Gradient Descent Method



67

Feedforward Neural Networks: Summary

- Perceptrons, with threshold logic function as activation function, are suitable for pattern classification tasks that involve linearly separable classes
- Multilayer feedforward neural networks, with sigmoidal function as activation function, are suitable for nonlinearly separable classes
  - Complexity of the model depends on
    - Dimension of the input pattern vector
    - Number of classes
    - Shapes of the decision surfaces to be formed
  - Architecture of the model is empirically determined
  - Large number of training examples are required when the complexity of the model is high
  - Local minima problem
- Multilayer feedforward neural network models are suitable for function approximation tasks also
- Multilayer feedforward neural network with one or two hidden layers is now called a shallow network

68

Text Books

1. J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Third Edition, Morgan Kaufmann Publishers, 2011.
2. S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press, 2009.
3. C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.
4. B. Yegnanarayana, Artificial Neural Networks, Prentice-Hall of India, 1999.
5. Satish Kumar, Neural Networks - A Class Room Approach, Second Edition, Tata McGraw-Hill, 2013.
6. S. Haykin, Neural Networks and Learning Machines, Prentice Hall of India, 2010.