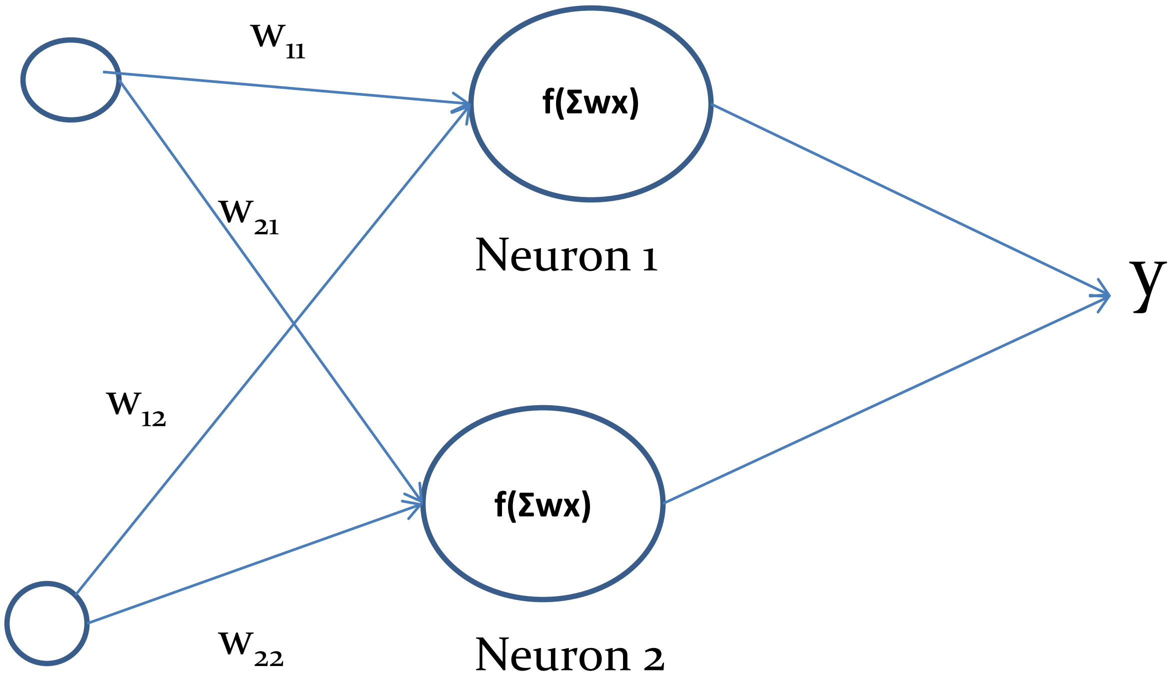
innovate achieve lead

BITS Pilani

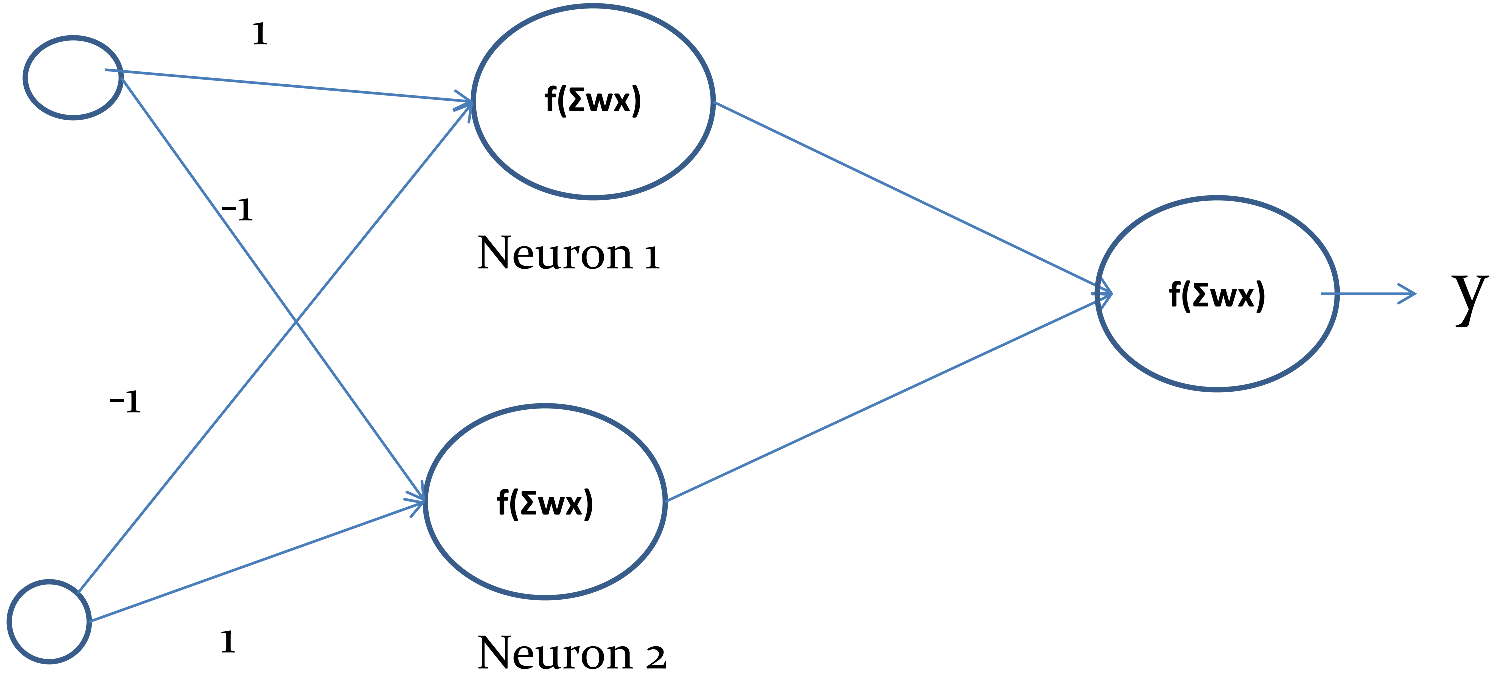
Machine Learning (IS ZC464) Session 10:

Artificial Neural Networks(ANN) – Perceptron and Linear decision Boundary, Pattern Recognition using ANN, Gradient Descent Algorithm

Neural Network for XOR



Neural Network for XOR



Perform Computations (T=1)

Activation Function

f(n) = 1 if n>=1

= 0 if n <1

x1	x2	w11	w12	w21	w22	n1	f(n1)	n2	f(n2)	f(m)
0	0	1	-1	-1	1	0	0	0	0	0
0	1	1	-1	-1	1	-1	0	1	1	1
1	0	1	-1	-1	1	1	1	-1	0	1
1	1	1	-1	-1	1	0	0	0	0	0

Activation Functions

- Activation functions trigger the received weighted sum of the input according to the expected output.
- Let g(n) be the activation function where n= $\Sigma w_i x_i$
- Based on the value of n, the g(n) is triggered.

- Different Activation Functions are
  - Step Function
  - Linear function
  - Sigmoid Function (Logistic Function)

Step Function

- G(n) = 1 if n>T

  = 0 if n<=T
- The output is always 1 or 0.
- Useful is pattern recognition where an output 1 can represent the class which the input test pattern belong to.

Example

- Let us have two discriminatory features <color, texture> for automatic fruit recognition.
- Let us code the attributes as 1, 2 and 3 for red, orange and yellow colors respectively. Also let the textures be defined as 1,2 and 3 representing the degrees of smoothness in increasing order.
- Let the training samples be

Training data

Color	Texture	Fruit (supervised learning)
1	2	Apple
3	2	Mango
2	1	Orange
1	3	Plum
1.4	2.8	plum
2	1.5	orange
2.5	2.2	mango
1	2.2	apple

Expected output

- Apple →1
- Orange→2
- Mango→3
- Plum→4

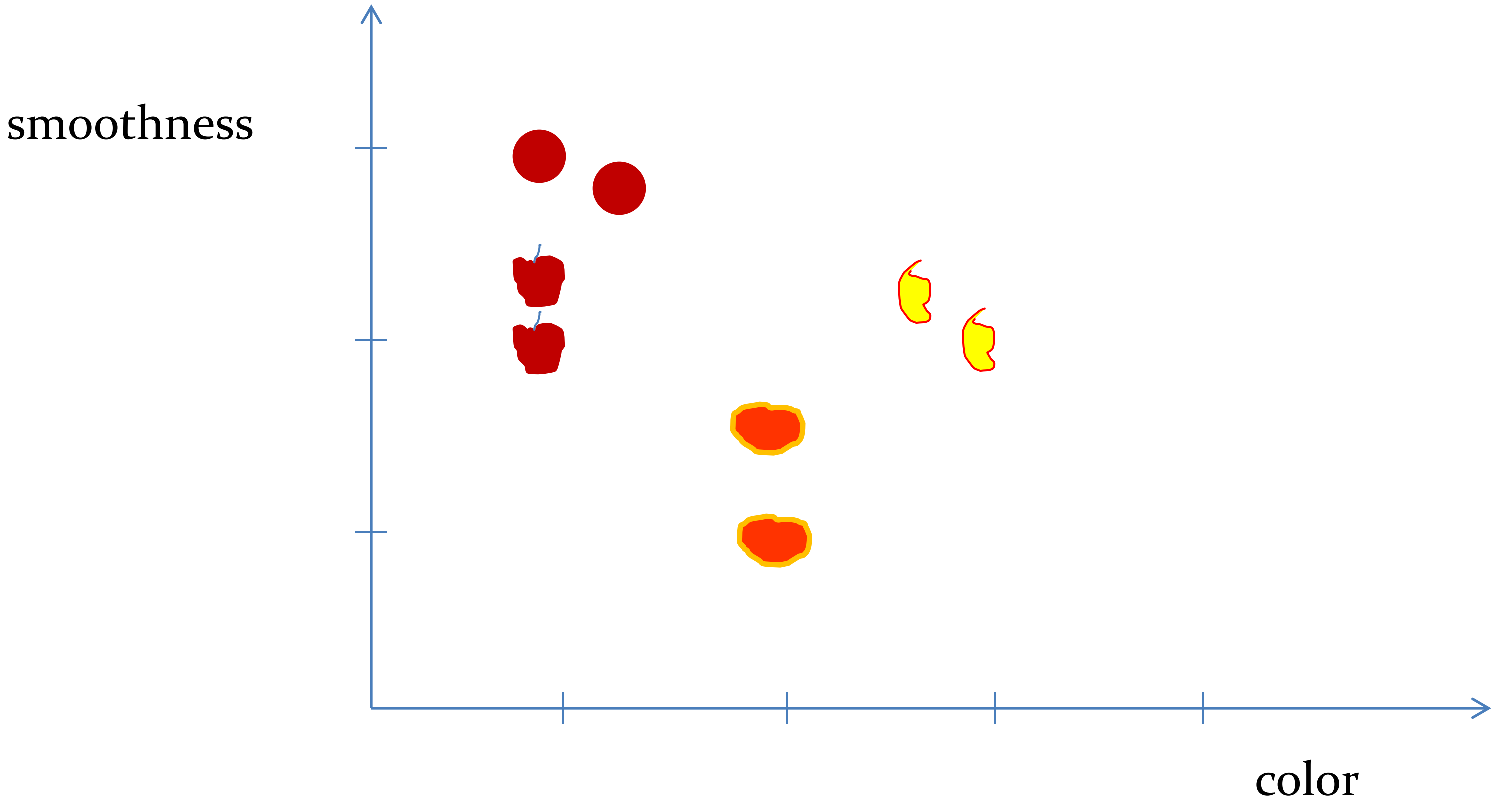
The expected output can also be modeled as a 4 tuple as <1,0,0,0> for apple

<0,1,0,0> for Orange

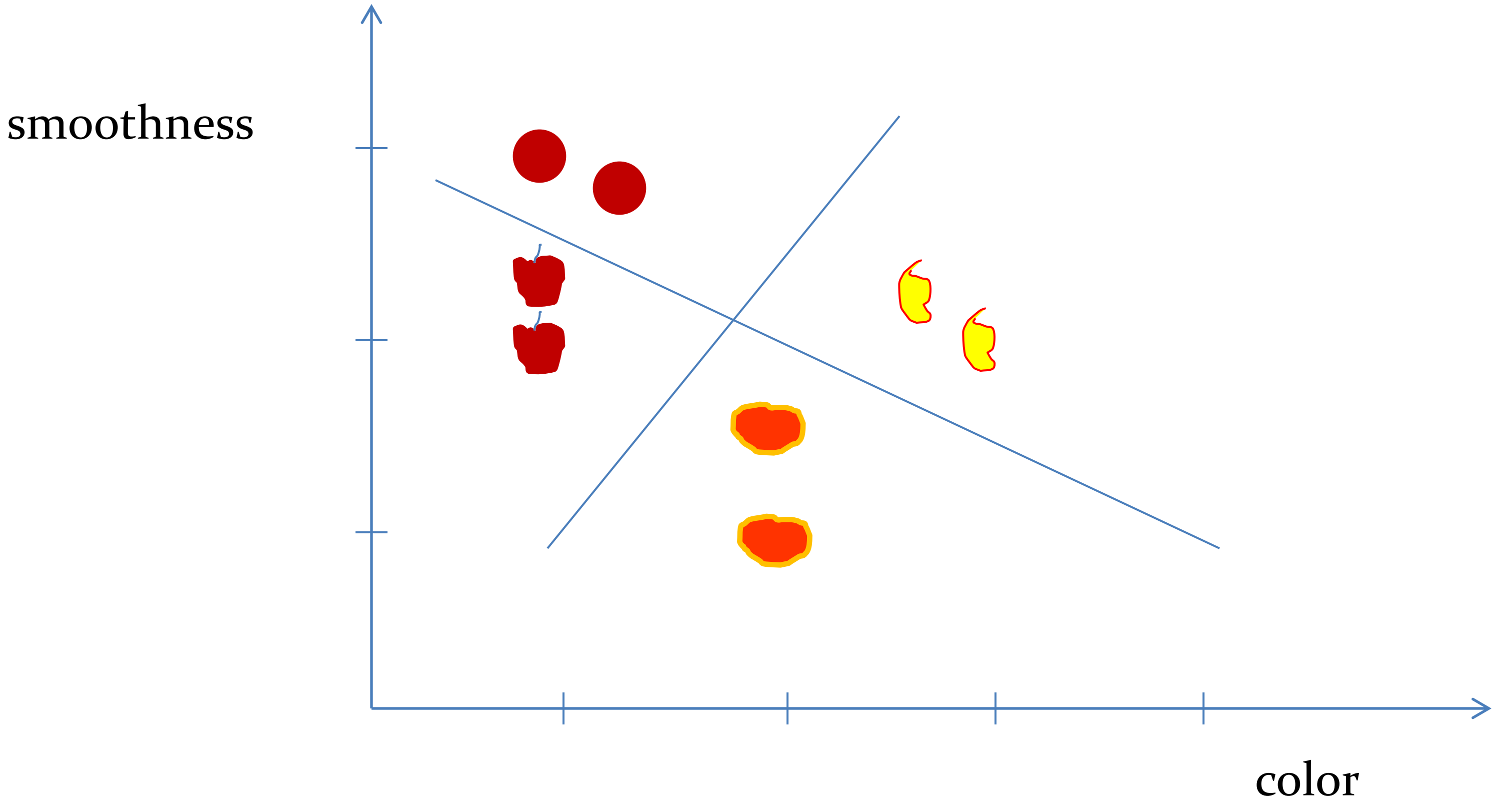
<0,0,1,0> for Mango

<0,0,0,1> for plum

Check the feasibility of using a single neuron for recognition of patterns

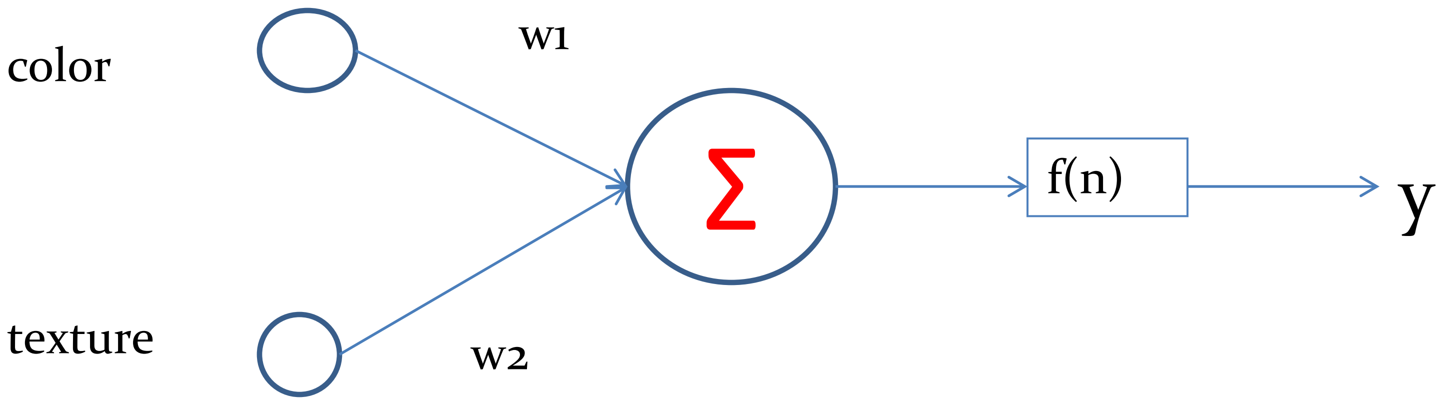


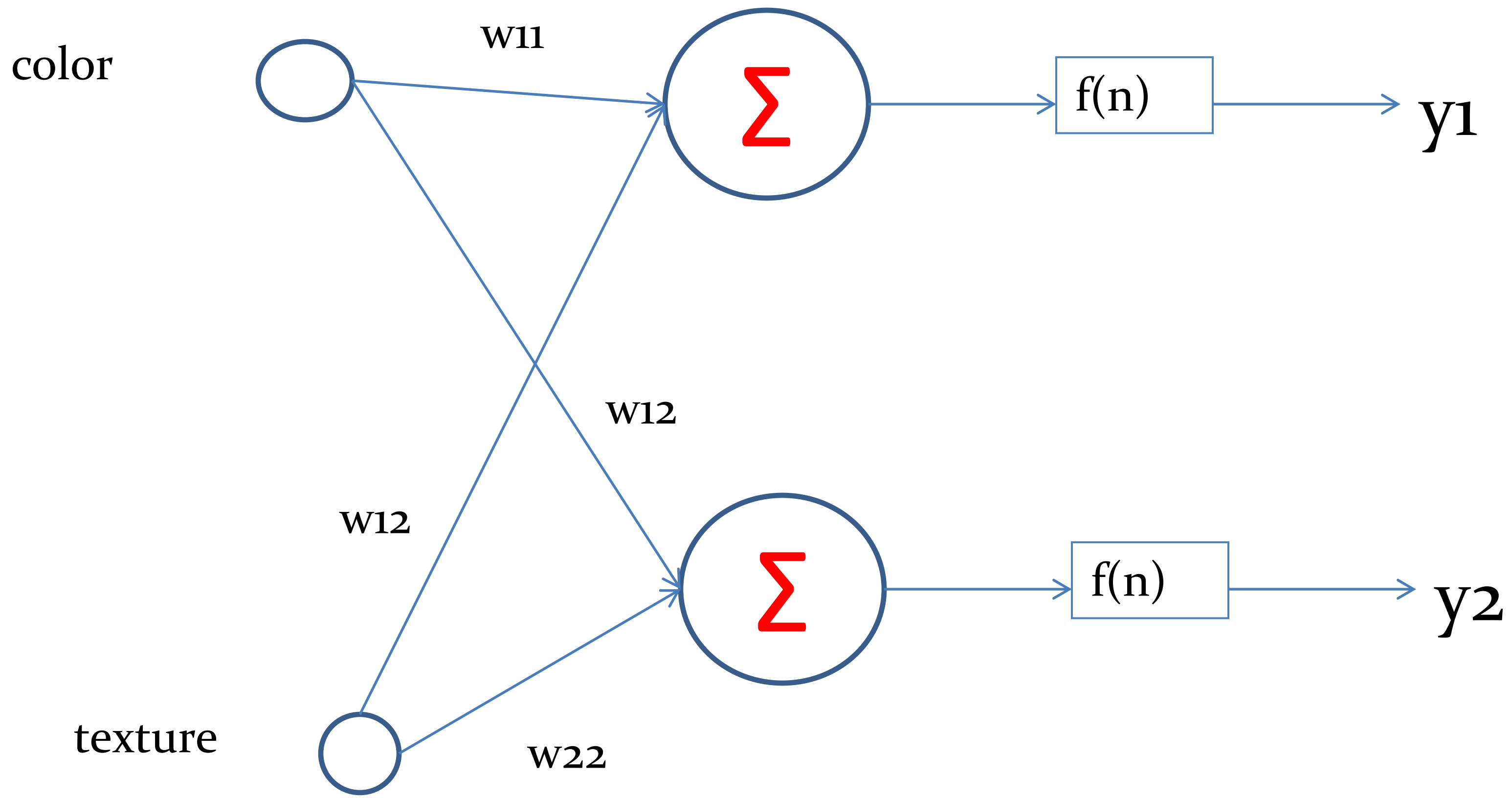
Try to draw lines that separate the classes: two decision boundaries hint the need for two neurons



Train the neuron: Select the activation function first

A simple step function simply gives 0 or 1 and is not sufficient to produce output for 4 classes




color
w11
Σ
f(n)
y1
w12
w12
Σ
f(n)
y2
texture
w22

Output modeling

- y1 = 0 , y2 = 0   represents that the test input is Apple
- y1 = 1 , y2 = 0   represents that the test input is Orange
- y1 = 1 , y2 = 0   represents that the test input is Mango
- y1 = 1 , y2 = 1   represents that the test input is Plum

Parameters for each neuron

- Neuron 1:
  - Activation Function: Step function
  - Threshold: 1
  - Weights: <1,-1>
- Neuron 2:
  - Activation Function: Step function
  - Threshold:1
  - Weights:<1,-1>

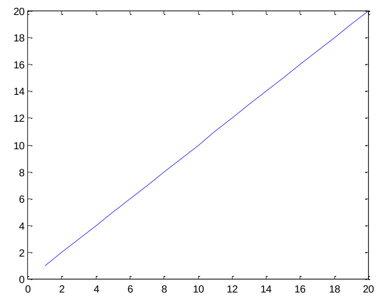
Training and Learning

- Training is supervised in nature.
- Neural network learns from training examples
- Knowledge is captured in terms of weights.
- Learning Algorithms require the initialization of weights which later adapt to other examples.
- Hit and trial methods do not work for larger data sets.
- Gradient Descent is the most popular learning algorithm.

Neural Network Architecture

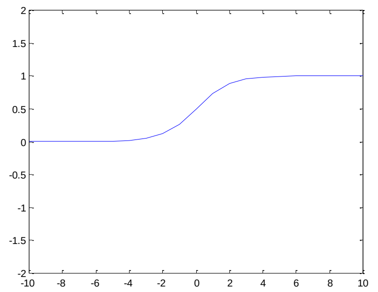
- The arrangement of neurons, along with the number of neurons, activation functions associated with the neurons and the input and output constitute the architecture of the Neural Network.
- Different Architectures include
  - Feed Forward Neural Networks-represents a function of its input
    - Single Layered
    - Multi Layered
  - Back Propagation Neural Networks- feeds its output back into its own inputs

Activation functions



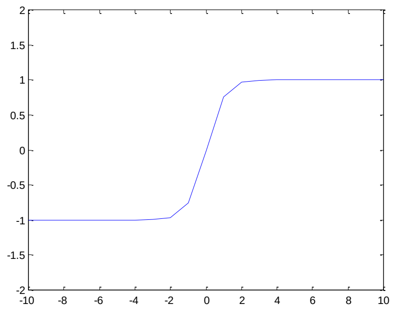
Linear

$$y = x$$



Logistic

$$y = \frac{1}{1+\exp(-x)}$$

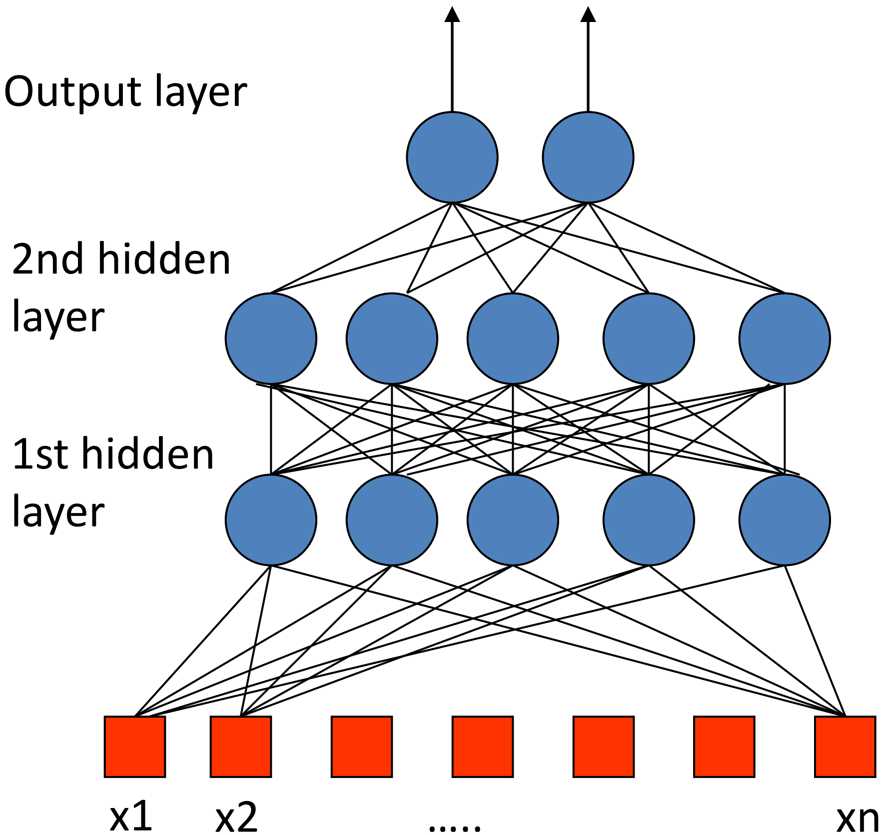


Hyperbolic tangent

$$y = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)}$$

*Slide adapted from : acat02.sinp.msu.ru/presentations/prevotet/tutorial.**ppt***

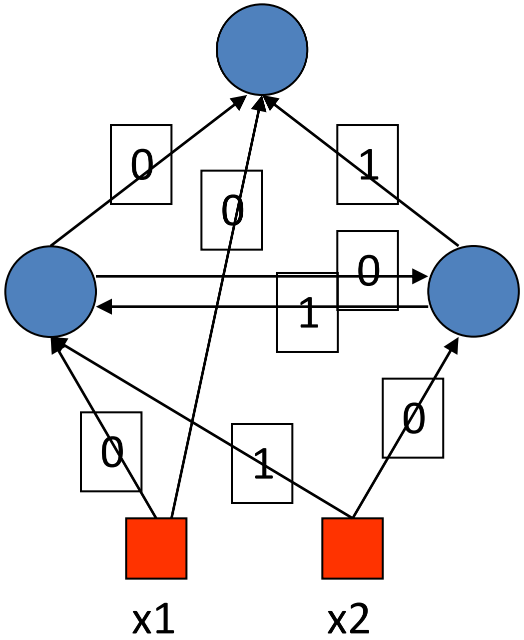
Feed Forward Neural Networks



- The information is propagated from the inputs to the outputs
- Time has no role (NO cycle between outputs and inputs)

*Slide adapted from : acat02.sinp.msu.ru/presentations/prevotet/tutorial.**ppt***

Recurrent Neural Networks



- Can have arbitrary topologies
- Can model systems with internal states (dynamic ones)
- Delays are associated to a specific weight
- Training is more difficult
- Performance may be problematic
  - Stable Outputs may be more difficult to evaluate
  - Unexpected behavior (oscillation, chaos, …)

*Slide adapted from : acat02.sinp.msu.ru/presentations/prevotet/tutorial.**ppt***

Learning

- The procedure that consists in estimating the parameters of neurons so that the whole network can perform a specific task

- 2 types of learning
  - The supervised learning
  - The unsupervised learning

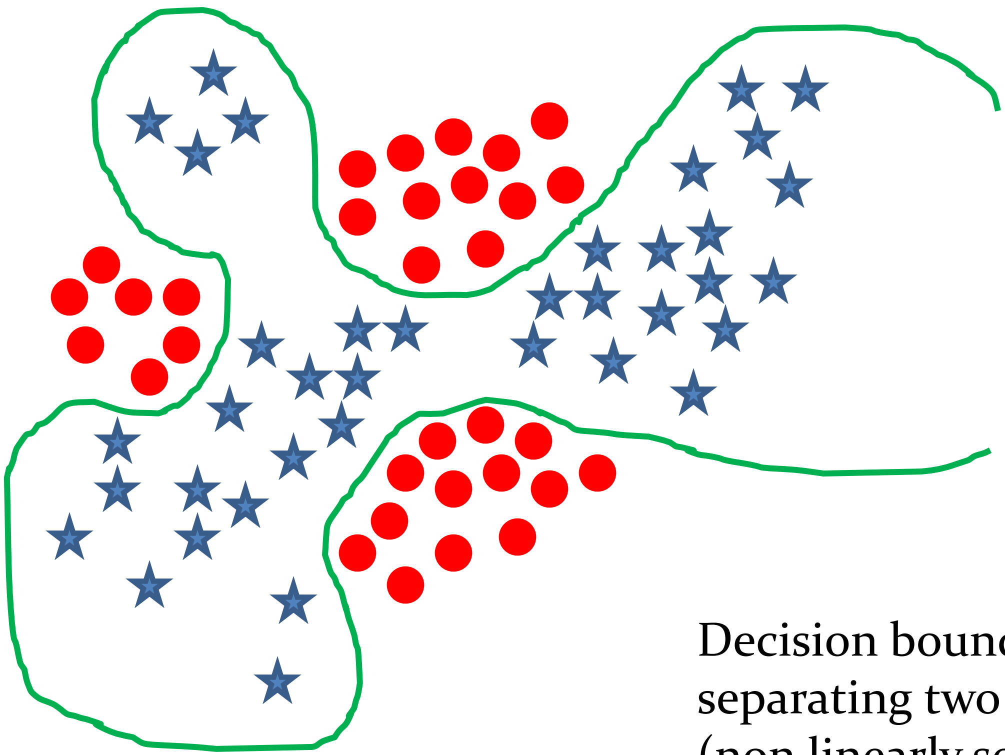
- The Learning process (supervised)
  - Present the network a number of inputs and their corresponding outputs
  - See how closely the actual outputs match the desired ones
  - Modify the parameters to better approximate the desired outputs

*Slide adapted from : acat02.sinp.msu.ru/presentations/prevotet/tutorial.**ppt***

Recall

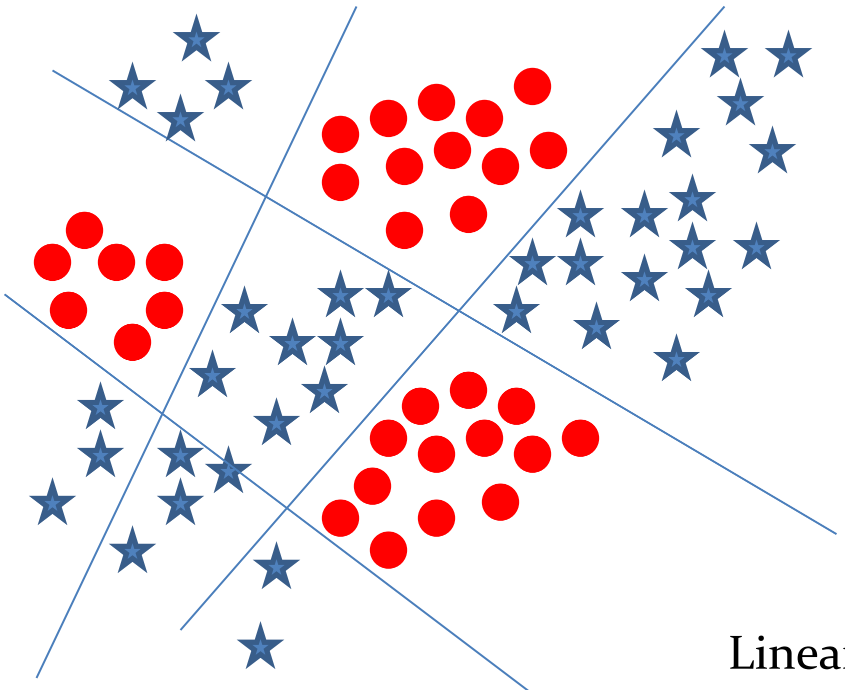
- Knowledge is acquired by the network through a learning process.
- Interconnection strengths known as synaptic weights are used to store the knowledge.

Example: Draw decision boundaries



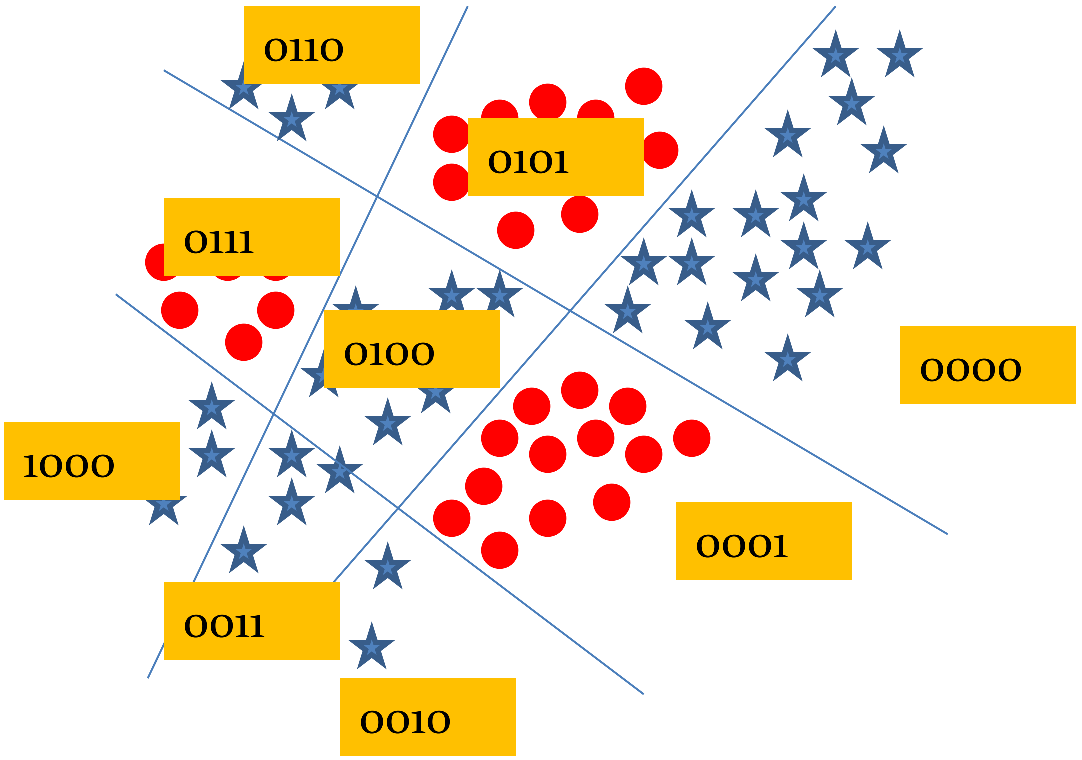
Decision boundary
separating two classes
(non linearly separable)

Example: Linear boundaries

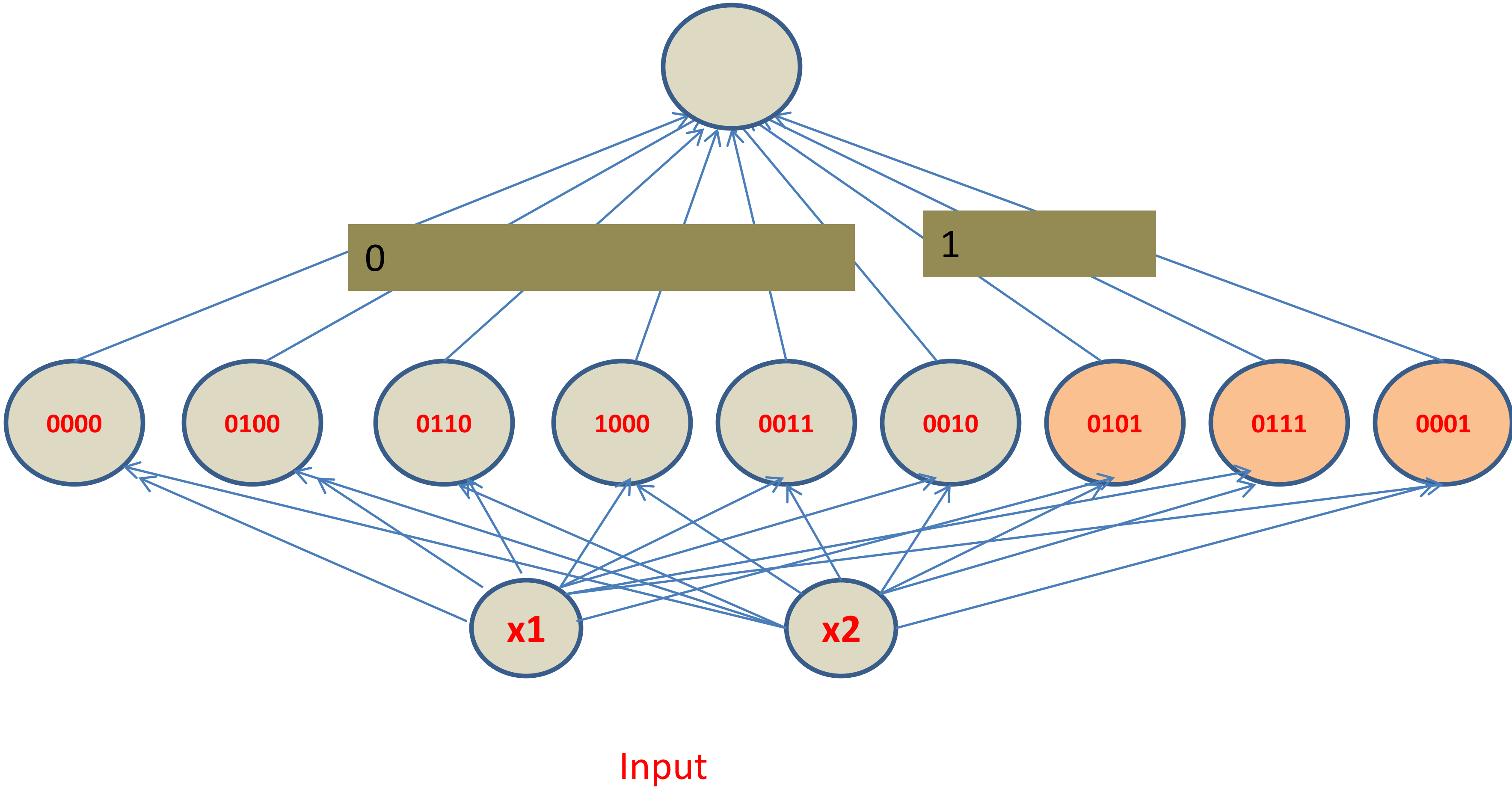


Linear Decision boundaries separating two classes

Example: give each region a label



Role of neurons in design of a neural network

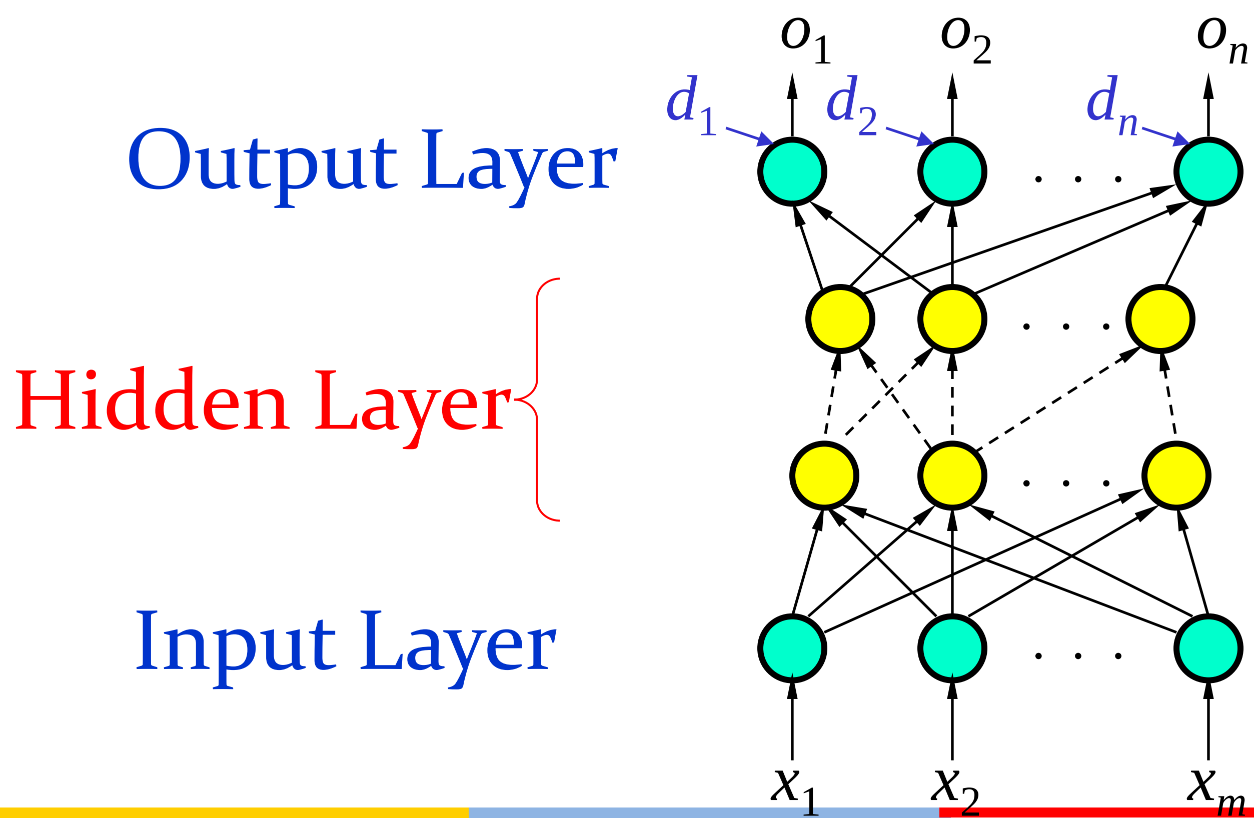


Represent Training Data

- $\{x^{(i)}, d^{(i)}\}$ for $i = 1,2,3,\ldots,m$
- Size of the training data = m
- $x^{(1)}$ is the feature vector corresponding to the first object
- $x^{(2)}$ is the feature vector corresponding to the first object
- $d^{(1)}$ is the class to which $x^{(1)}$ belongs
- $d^{(2)}$ is the class to which $x^{(2)}$ belongs
- And so on

Forward Learning

O1,o2,o3 etc are the output values produced by the NN



Goal

Sum of Squared Errors

$$E^{(l)} = \frac{1}{2}\sum_{j=1}^{n}\left[d_j^{(l)} - o_j^{(l)}\right]^2$$

Goal:

Minimize

$$E = \sum_{l=1}^{p} E^{(l)}$$

Learning Factors

- Initial Weights
- Learning Constant (η)
- Cost Functions
- Update Rules
- Training Data and Generalization
- Number of Layers
- Number of Hidden Nodes

Learning Phase

- During the learning phase the weights in the Feed Forward Neural Network are modified.
- All weights are modified in such a way that when a pattern is presented, the output unit with the correct category, hopefully, will have the largest output value.

In 2D space the line parameters are two

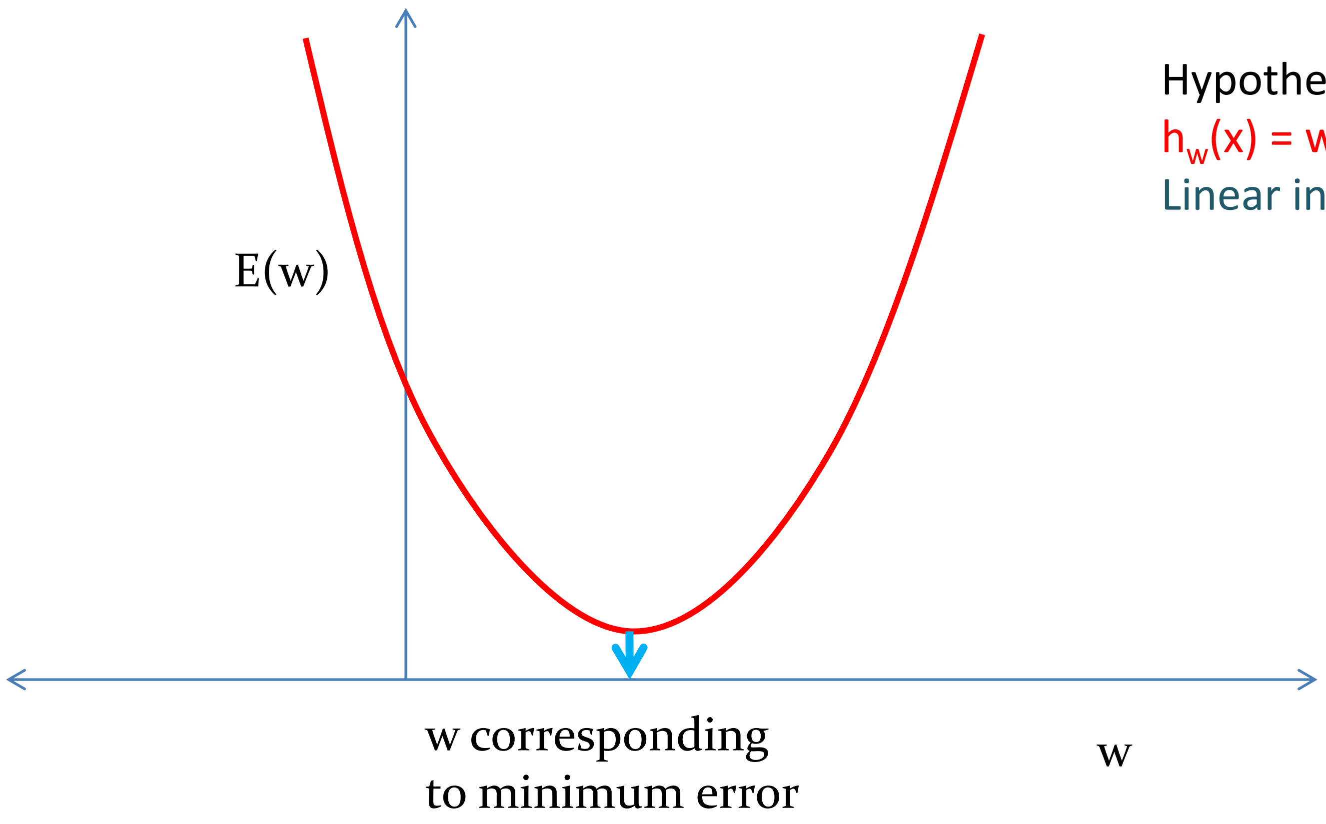
- Slope and intercept
- Can be called as w_1 and w_2
- In order to find a line that best fits the given data, we must find w1 and w2 in such a way that the sum of the squared error is minimum

Error surface for Neural Network based classification

- Consider m observations $<x^1,y^1>$, $<x^2,y^2>$, ….$<x^m,y^m>$.
- An hypothesis $h_w(x)$ that approximates the function that fits best to the given values of y
- There is likely to be some error corresponding to each observation (say i).
- The magnitude of such error is $y^i - h_w(x^i)$
- Objective is to find such w that minimizes the sum of squares of errors

$$E_{min}(w) = \text{Minimize}_w \sum_i (y^i - h_w(x^i))^2$$

Plotting error when y=f(x)

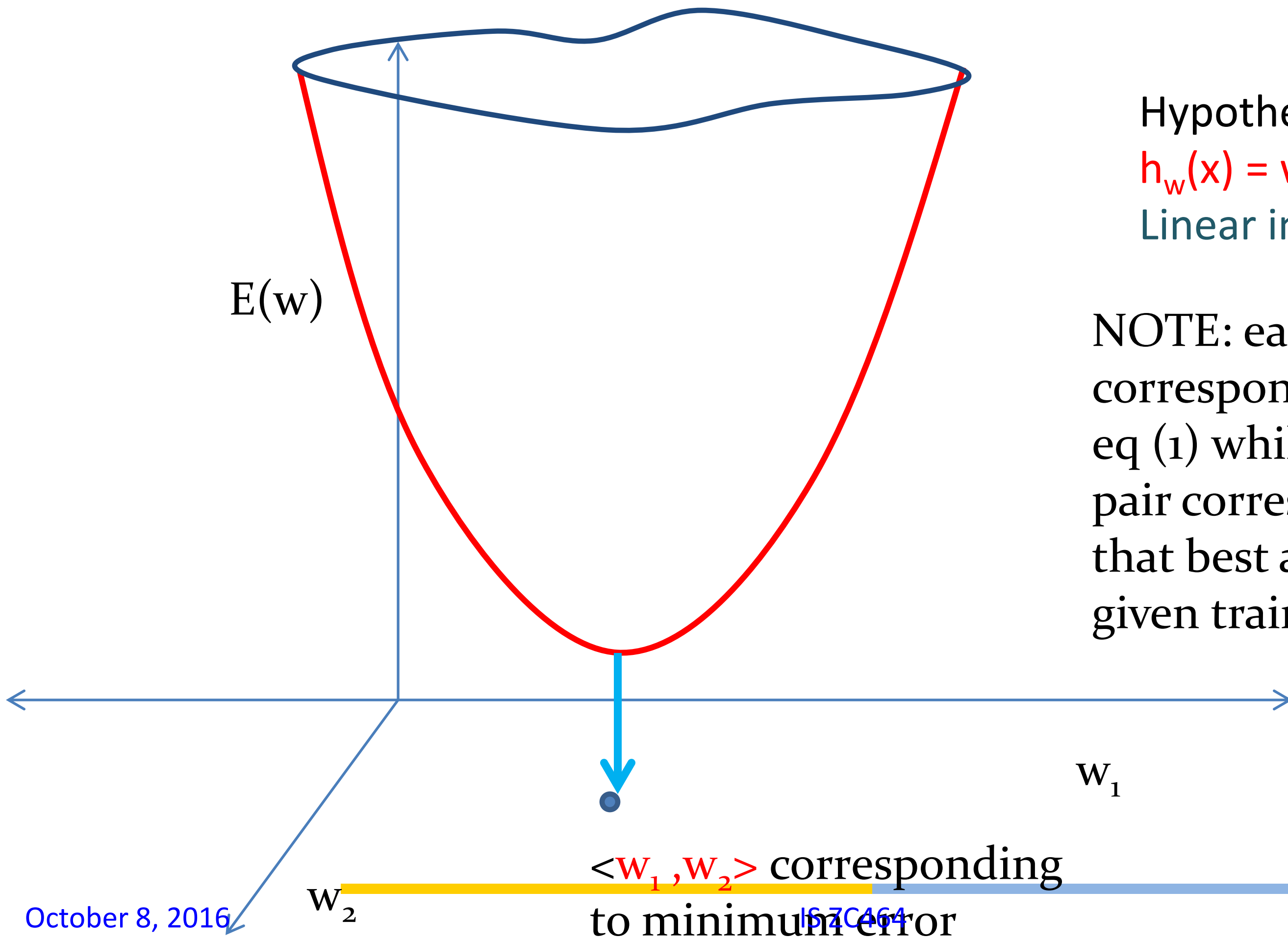


Hypothesis function

$h_w(x) = wx$

Linear in one variable

Plotting error when $y=f(x_1,x_2)$



Hypothesis function

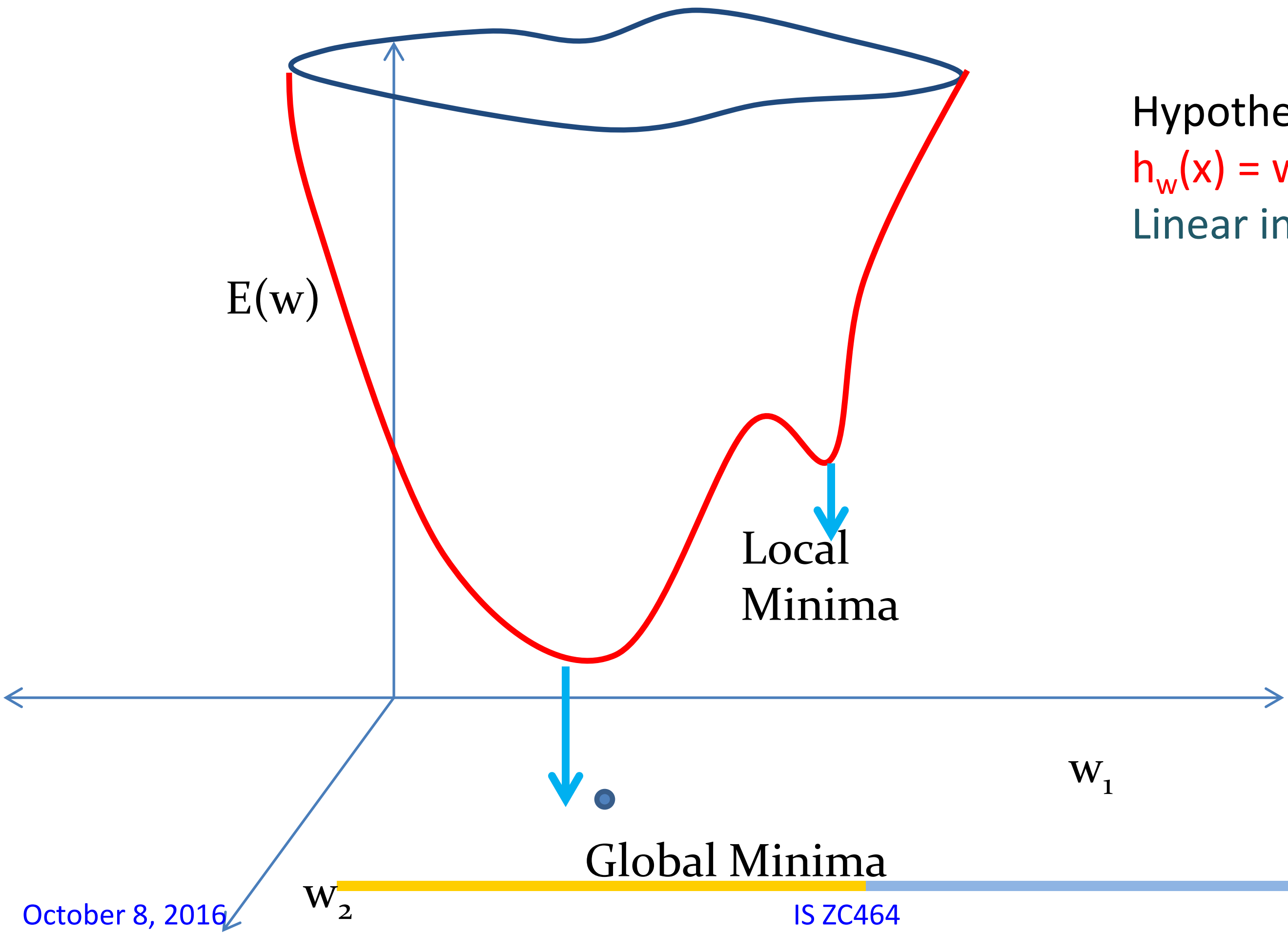
$h_w(x) = w_1x_1 + w_2x_2$ ....(1)

Linear in two variables

NOTE: each pair < w_1 ,w_2> corresponds to a line given by eq (1) while only one such pair corresponds to the line that best approximates the given training data

Another possible surface

Plotting error when $y=f(x_1,x_2)$



Hypothesis function

$h_w(x) = w_1x_1 + w_2x_2$ …….(1)

Linear in two variables