

# Neural Networks

## MLP ANNs



André C P L F de Carvalho  
Instituto de Ciências Matemáticas e de Computação  
Universidade de São Paulo, Brazil

1

## Topics

- Multi-Layer Perceptron
- Training
- Backpropagation network
- Input space partition
- Classification

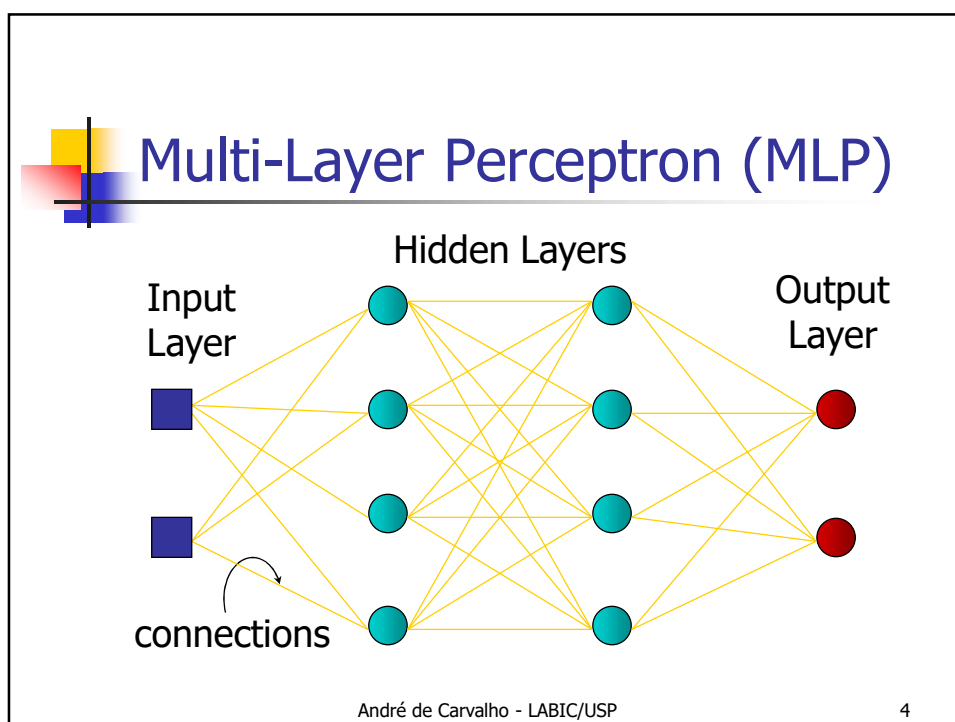
2

## A bit of history


- (1982) Hopfield showed that ANNs could be seen as dynamic systems
- (1986) Hinton, Rumelhart e Williams, proposed an algorithm to train multi-layer perceptron (MLP) networks
  - *Parallel Distributed Processing*
  - Bryson e Ho (1959), Werbos (1974), Parker (1985) and Le Cun (1985)

3

3



4




## Multi-Layer Perceptron (MLP)

- Most used ANN model
  - One or more hidden layers
- Increased functionality
  - One hidden layer: any Boolean or continue function
  - Two hidden layers: any function
- Trained by the Backpropagation algorithm

18/12/2019 André de Carvalho 5

5



## Backpropagation

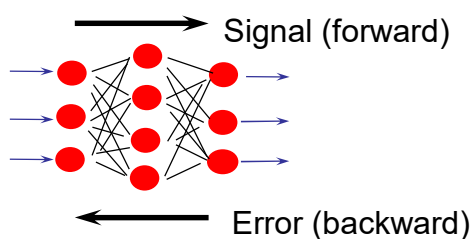
- Train by reducing errors made by the MLP network
  - Supervised
  - Error correction
    - Output layer
    - Hidden layers
      - Proportional to the error made by the nodes in the next layer

18/12/2019 André de Carvalho 6

6

## Backpropagation

- Training follows two directions
  - Forward
  - Backward



18/12/2019

André de Carvalho

7

7

## Backpropagation

- Training
  - Supervised
  - Adjust two weights:  $\Delta w_{ij} = \eta x_i \delta_j$

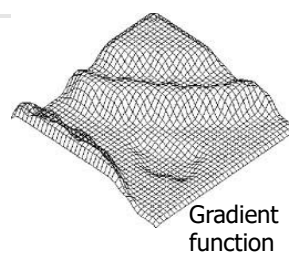
$$\delta_j = \begin{cases} f'(net) error_j & \text{if } j \text{ is the output layer} \\ f'(net) \sum w_{jk} \delta_k & \text{if } j \text{ is a hidden layer} \end{cases}$$

$$error_j = \frac{1}{2} \sum_{q=1}^c (y_q - f(net_q))$$

$$net = \sum_{i=0}^m x_i w_i$$

If  $f(net)$  is a sigmoidal function,  $f'(net) = f(net)(1 - f(net))$

Training is not guaranteed to converge

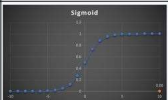
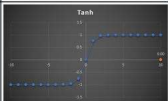
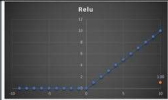
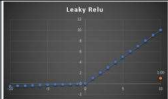


© André de Carvalho - ICMC/USP

8

8

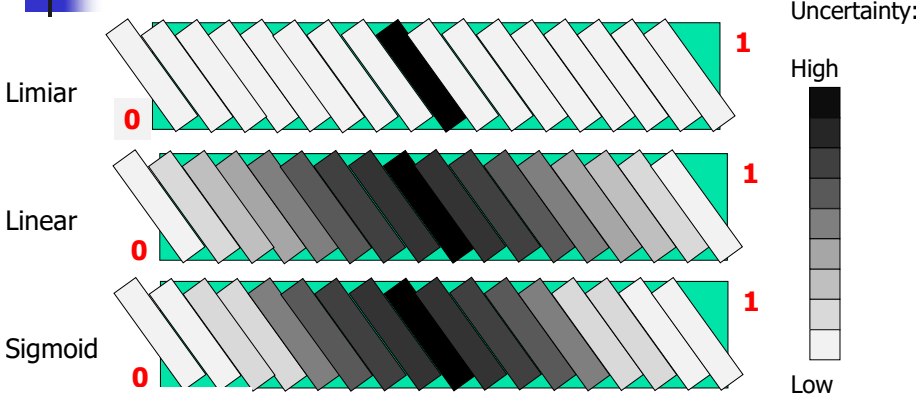
## Activation functions

Name	Plot	Equation	Derivative
Sigmoid		$f(x) = \sigma(x) = \frac{1}{1 + e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
Tanh		$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	$f'(x) = 1 - f(x)^2$
Rectified Linear Unit (relu)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Leaky Rectified Linear Unit (Leaky relu)		$f(x) = \begin{cases} 0.01x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0.01 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$

© André de Carvalho - ICMC/USP


9

## Activation functions and borders




Uncertainty:  
High  
Low


Step function




Slope function



Sigmoid function



Linear function



© André de Carvalho - ICMC/USP

10

## Training

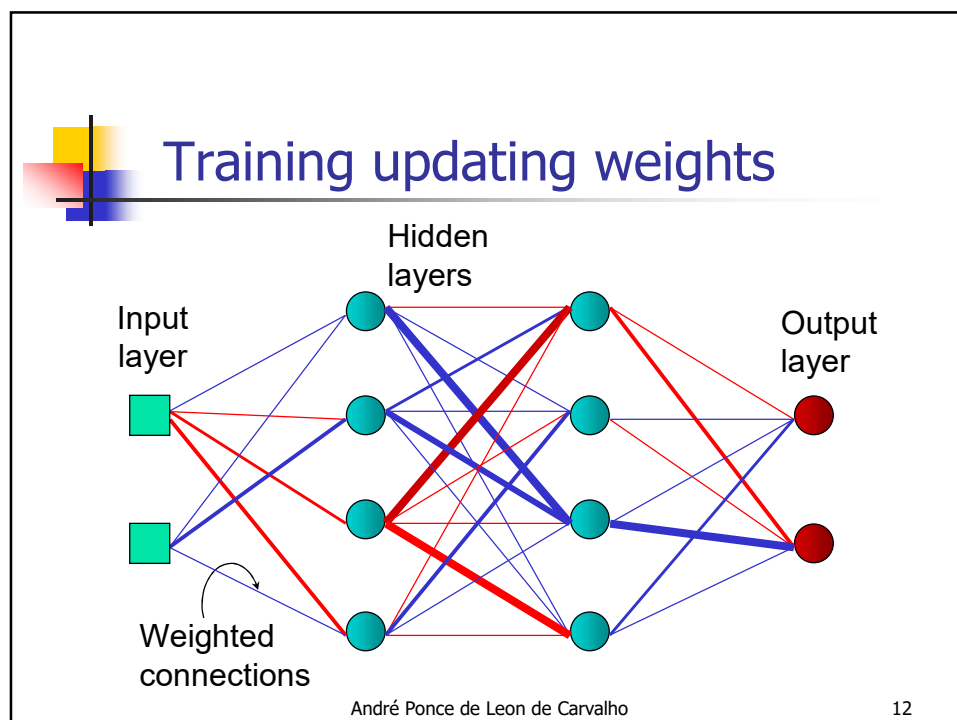
```

start all the weights with random values
repeat
  for each training pattern (X, d) do
    for each layer k from 1 to n do
      for each node j from 1 to mk do
        calculate the output yjk
        if layer = n
          then calculate error
        if error > ε
          then for each layer k from n to 1 do
            for each node j from 1 to mk do
              update weight values
      until error ≤ ε

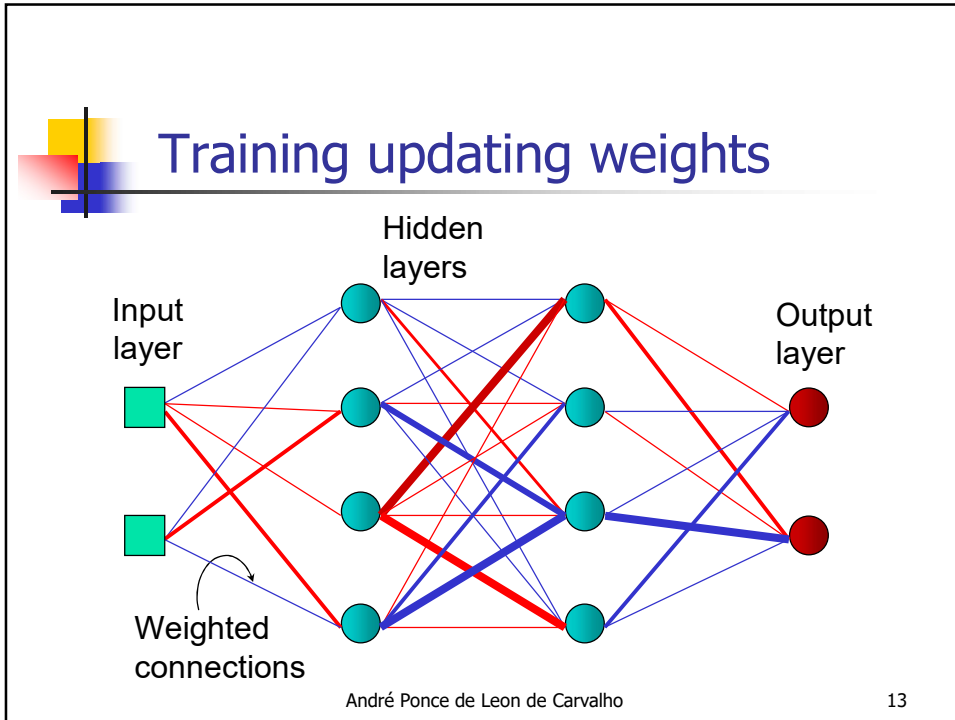
```

18/12/2019 André de Carvalho 11

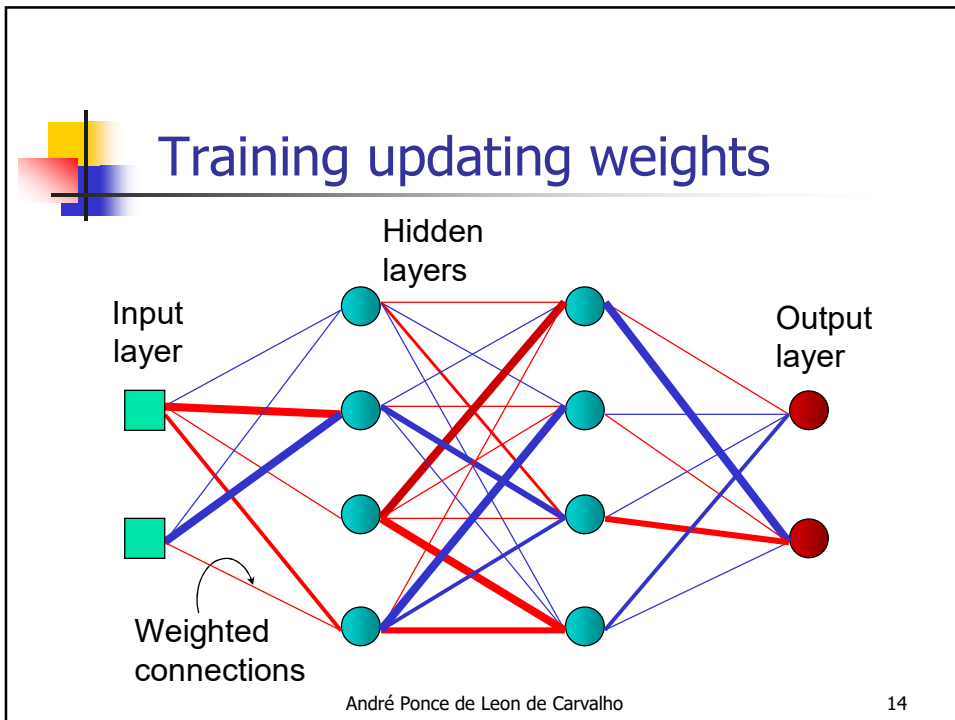
11



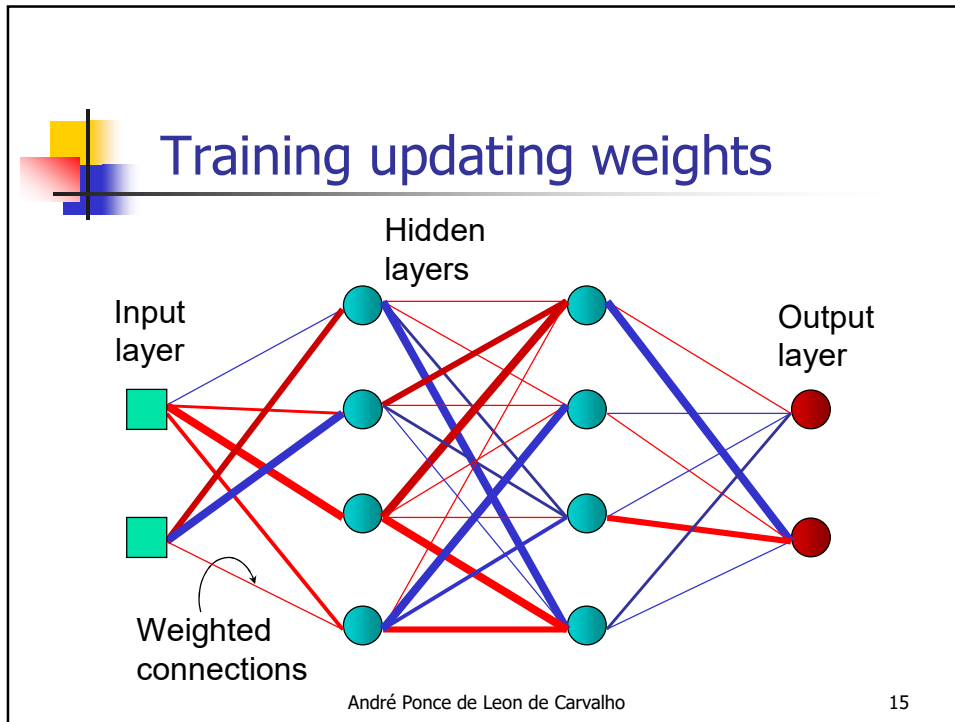
12



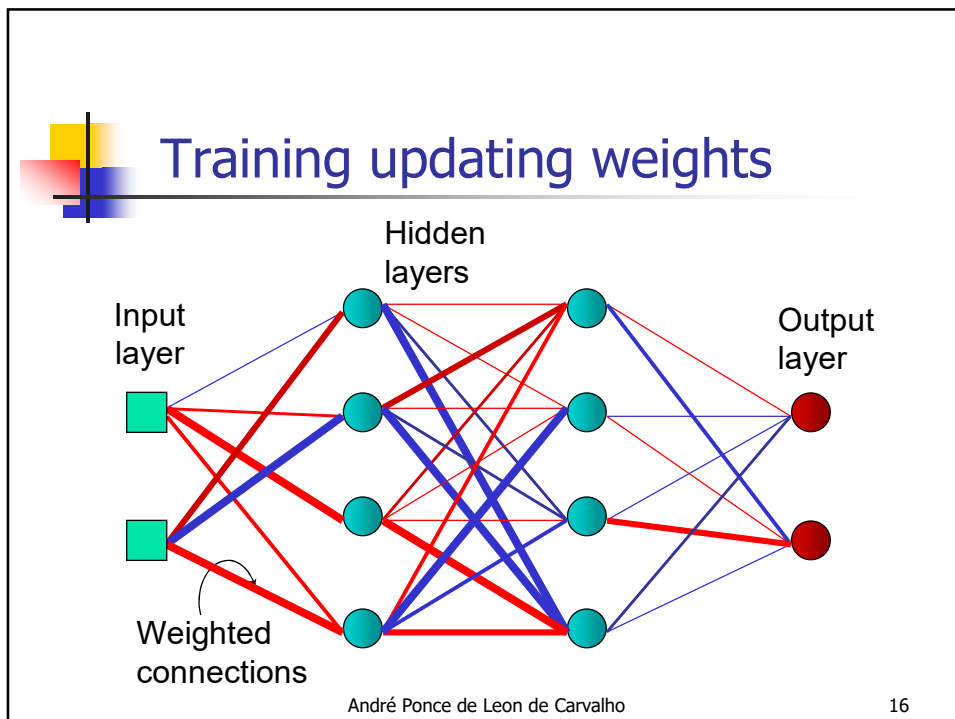
13



14

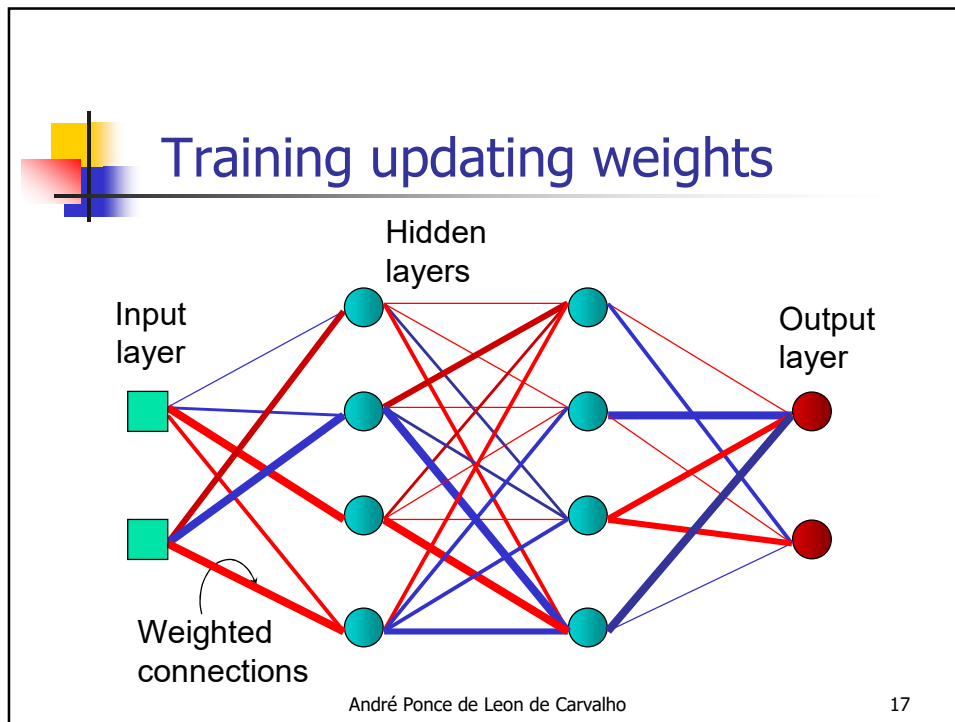


15

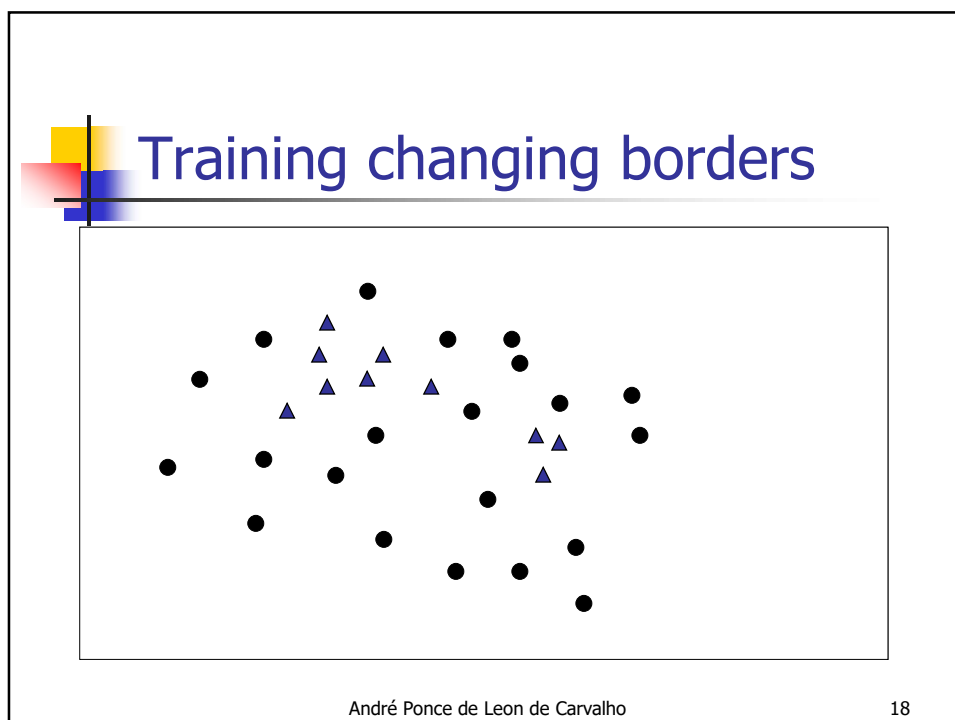


16



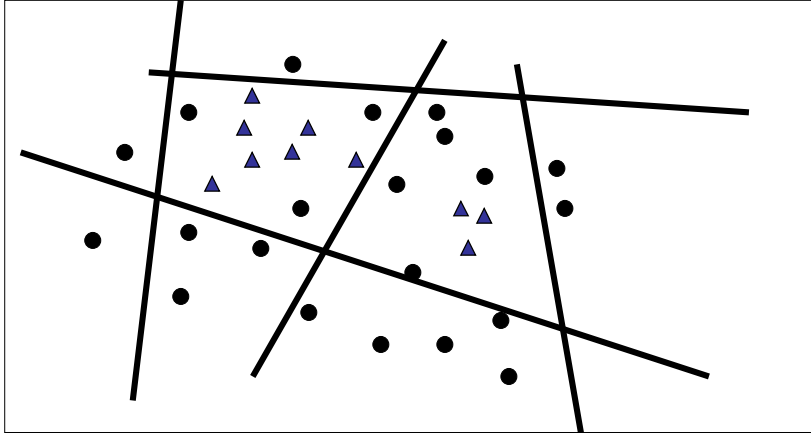


17



18

## Training changing borders

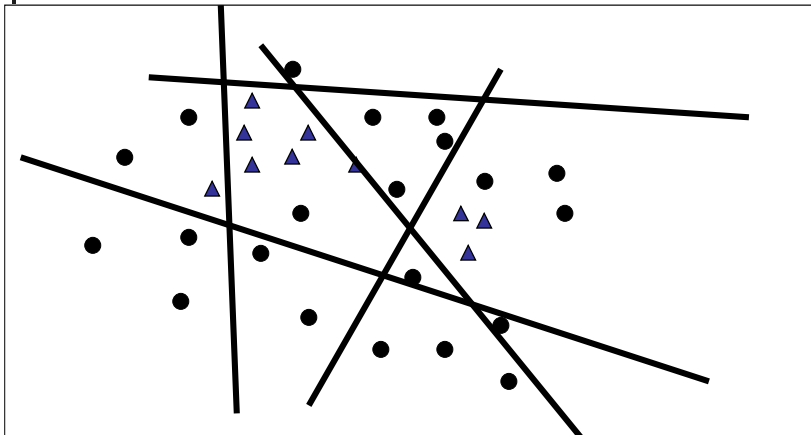


André Ponce de Leon de Carvalho

19

19

## Training changing borders

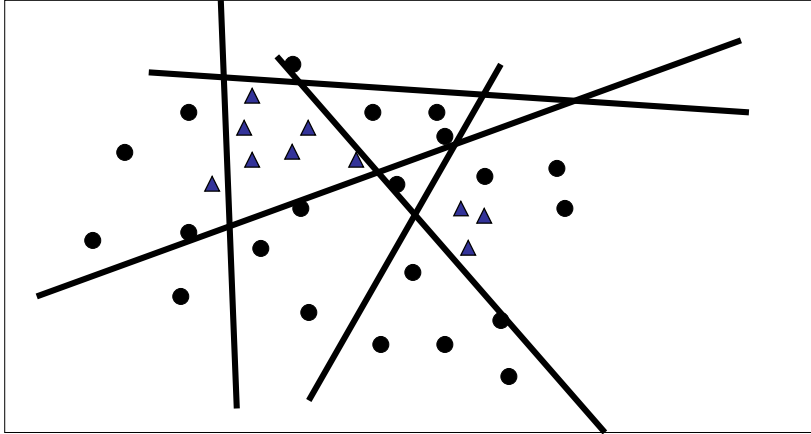


André Ponce de Leon de Carvalho

20

20

## Training changing borders



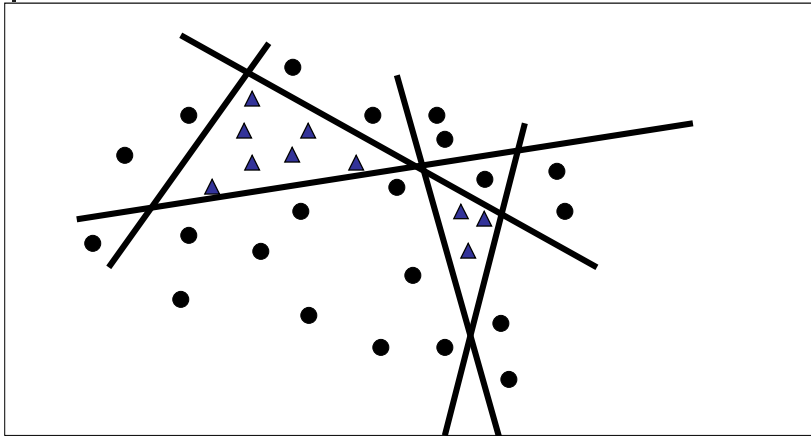
André Ponce de Leon de Carvalho

21

The figure shows a 2D scatter plot with two classes of data points: black circles and blue triangles. There are approximately 15 black circles and 10 blue triangles. The plot is divided into several regions by five black lines representing decision boundaries. The boundaries are somewhat complex and non-linear, suggesting a model that has learned to separate the classes based on a set of linear features. The title 'Training changing borders' is displayed in blue text at the top left of the plot area.

21

## Training changing borders



André Ponce de Leon de Carvalho

22

This figure is similar to the one above, showing a scatter plot of black circles and blue triangles. However, the decision boundaries (black lines) are different, indicating a different model or a different set of training data. The boundaries are more complex and appear to be more tailored to the specific distribution of the data points in this plot. The title 'Training changing borders' is displayed in blue text at the top left of the plot area.

22

## Test

*for each test pattern do*  
*for each layer  $k$  from 1 to  $n$  do*  
*for each node  $j$  from 1 to  $m_k$  do*  
*calculate the output  $y_{jk}$*   
*compare  $Y$  and  $D$*   
*classify the test pattern  $X$  as belonging to*  
*the class whose desired output is*  
*closer to the produced output*

18/12/2019
André de Carvalho
23

23

## MLPs as classifiers

© André de Carvalho - ICMC/USP

24

## Convex regions

Open      Open      Open

Closed      Closed      Closed

© André de Carvalho - ICMC/USP      25

25

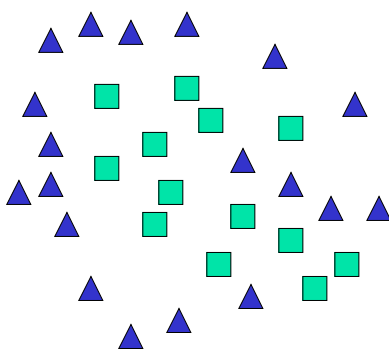
## Combinations of convex regions

© André de Carvalho - ICMC/USP      26

26

## Combinations of convex regions

- Find decision boundaries that separate the data below:

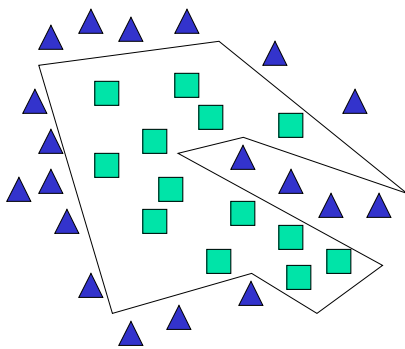


© André de Carvalho - ICMC/USP

27

## Combinations of convex regions

- Find decision boundaries that separate the data below:

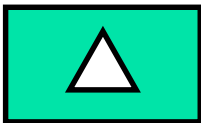


© André de Carvalho - ICMC/USP

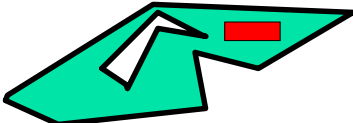
28

## Quiz 1

- How many layers and at least how many neurons in each layer have the networks that divide the input space of the shapes below:



■ class 1  
□ class 2



□ class 1  
■ class 2  
■ class 3

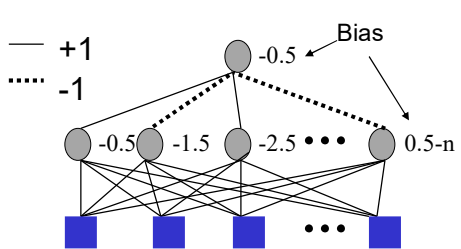
© André de Carvalho - ICMC/USP 29

29

## Exercise

- Given the ANN below whose input is a n-bit binary vector and output is a binary value:
  - Indicate the function implemented:
  - Explain what each neuron does

The activation function uses step function with threshold



© André de Carvalho - ICMC/USP 30

30



## Exercise

- Parity
  - One of the limitations of the Perceptron raised by Minsky and Papert
- Difficult problem
  - More similar patterns require different responses
  - Uses  $n$  intermediate units to detect parity in binary vectors with  $n$  elements

André Ponce de Leon de Carvalho

31

31



## Example

9	0	8	8	1	3	2	5	7	4	8	0	3	8	0	2	4	9	0	9	2	0	5
8	8	8	6	5	8	3	3	0	6	4	4	8	1	2	6	9	4	3	3	0	6	3
4	6	2	9	3	6	7	7	5	2	7	8	0	1	2	4	4	8	4	3	9	2	
2	1	7	8	9	5	8	7	4	4	9	4	8	9	5	6	6	6	7	4	4	2	1
9	2	0	3	6	4	2	1	8	6	7	4	3	8	1	9	1	6	3	4	5	6	5
1	0	1	2	2	0	6	5	4	6	7	5	2	0	8	1	2	8	1	7	8	8	1

© André de Carvalho - ICMC/USP

32

32



## Example

© André de Carvalho - ICMC/USP

33

## Other training algorithms

- Backpropagation momentum
- Resilient propagation (Rprop)
- Quickprop
- Newton
- Levenberg Marquardt
- Super Self-Adjusting Backpropagation (superSAB)
- Conjugate gradient algorithms
- ...

André Ponce de Leon de Carvalho

34



## Backpropagation momentum

- Adds a momentum term to the weight update equation
  - If the last and current weight update go in the same direction, current update is larger
    - Direction: increase or decrease weight
  - Specifies the amount of the old weight change to be added to the current change
  - Increase chance to escape from local minima

© André de Carvalho - ICMC/USP

35

35




## Quiz 2

- What are limitations of MLP?
  - A) Only works with up to 2 hidden layers
  - B) Can learn any function using 2 hidden layers
  - C) Can use any nonlinear activation function
  - D) Can be used only for classification tasks

© André de Carvalho - ICMC/USP

36

36



---

**Next:**  
**Other Neural  
Networks**

© André de Carvalho - ICMC/USP 37