

1

**Aula 5  
do Prof. Emilio  
de PSI3471 >>>>  
Se iniciou aqui**

© Prof. Emilio Del Moral Hernandez

1

2

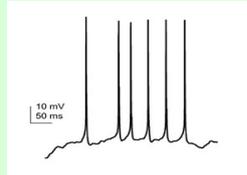
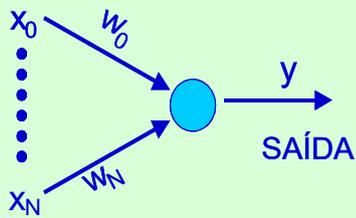
*PSI3471 – Prof. Emilio:*

*Recordando (pós carnaval)  
alguns dos pontos principais  
das aulas #1 a #4 e seguindo  
com as aulas #5, #6 ...*

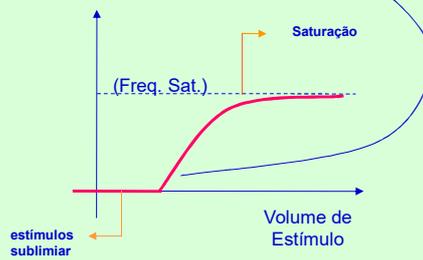
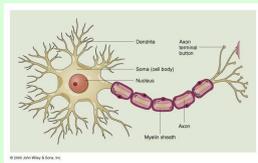
© Prof. Emilio Del Moral Hernandez

2

## Computação linear e não linear, com codificação frequencial



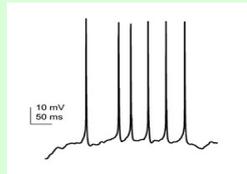
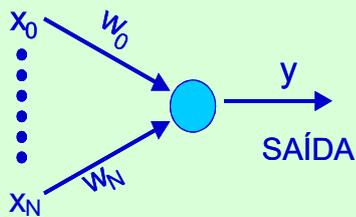
$$y = f_T(\sum w_i x_i)$$



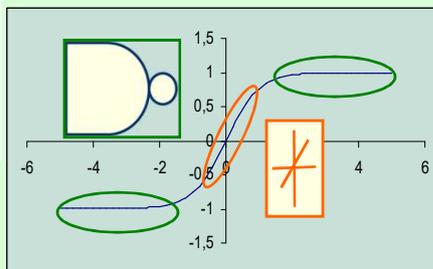
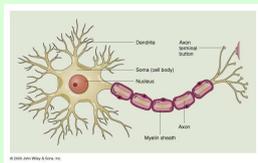
© Prof. Emilio Del Moral Hernandez

7

## Computação linear e não linear, com codificação frequencial



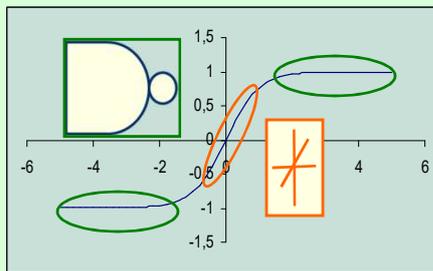
$$y = f_T(\sum w_i x_i)$$



© Prof. Emilio Del Moral Hernandez

8

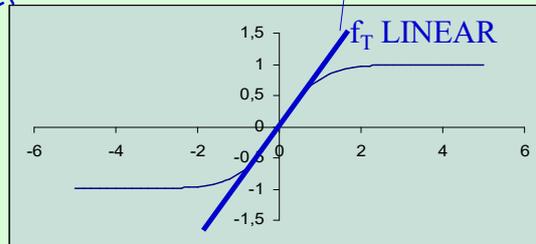
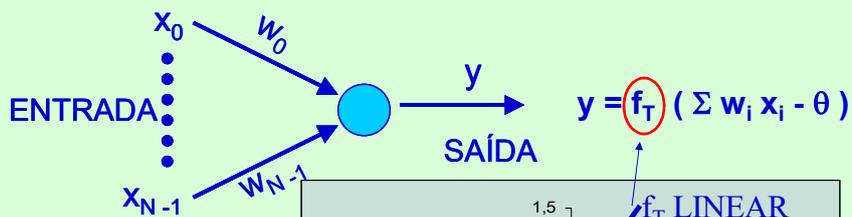
## Computação linear e não linear, com codificação frequencial



© Prof. Emilio Del Moral Hernandez

9

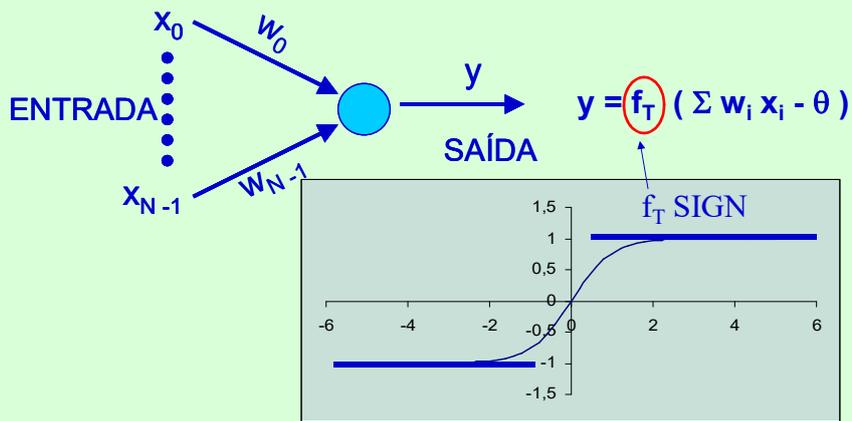
## Caso particular do nó com comportamento linear ... $f_T(\arg) = k \cdot \arg$



© Prof. Emilio Del Moral Hernandez

10

Caso particular do nó com comportamento binário  
 ...  $f_T(\text{arg}) = \text{sign}(\text{arg})$

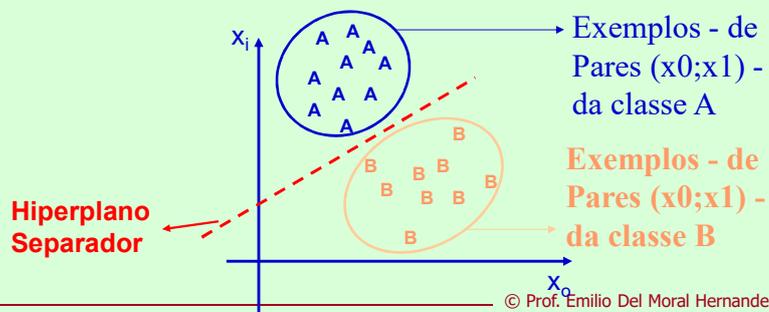


© Prof. Emilio Del Moral Hernandez

11

O Perceptron Digital:  $y = \text{signal}(\sum w_i x_i - \theta)$   
 (função de transferência tipo “degrau”)

- Viabiliza a classificação de padrões com separabilidade linear
- O algoritmo de aprendizado adapta os Ws de forma a encontrar o hiperplano de separação adequado
- Aprendizado por conjunto de treinamento

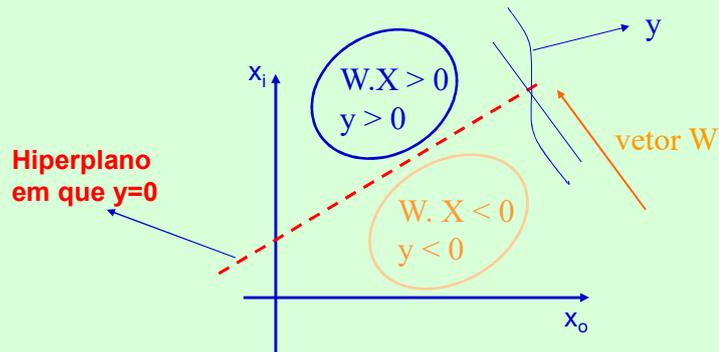


© Prof. Emilio Del Moral Hernandez

12

E se a saída do nosso problema não for digital?  
O "Perceptron Contínuo":  $y = \text{tgh}(\sum w_i x_i - \theta)$

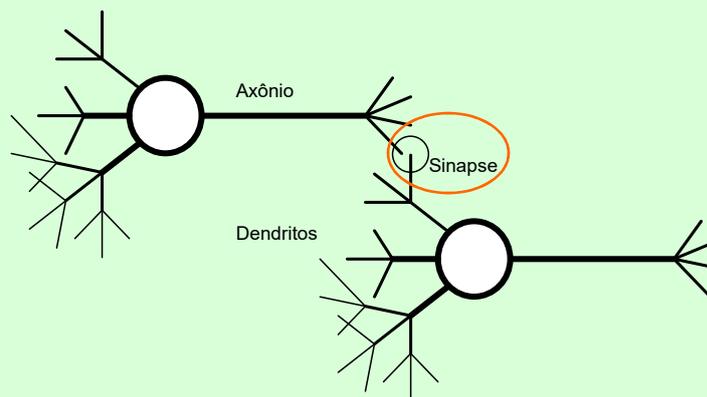
- Que problemas de entradas contínuas conseguimos atacar usando uma função de transferência tangente hiperbólica)



© Prof. Emilio Del Moral Hernandez

13

Cômputos mais complexos ... são realizados  
pelo encadeamento de vários neurônios



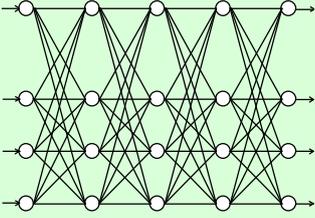
A conexão entre um axônio de um neurônio e um dendrito de outro é denominada **Sinapse**

© Prof. Emilio Del Moral Hernandez

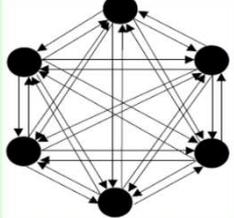
14

### Três arquiteturas neurais importantes (abordadas em pósgrad – PSI 5886)

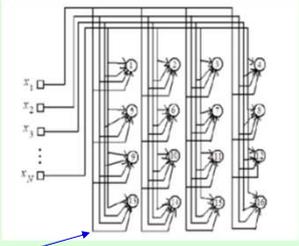
1) MLP  
- Multi Layer  
Perceptron

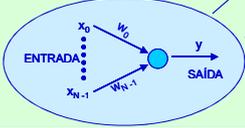


2) Memória  
Associativa  
de Hopfield



3) Mapas Auto-  
Organizáveis  
de Kohonen





ENTRADA  $x_0$   $w_0$   $y$  SAÍDA  
 $x_{n-1}$   $w_{n-1}$

© Prof. Emilio Del Moral Hernandez

15

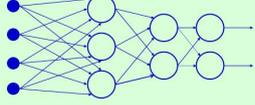
### Três arquiteturas neurais importantes (abordadas em pósgrad – PSI 5886)

**Nosso Foco aqui**

1) MLP  
- Multi Layer  
Perceptron

Foco deste Curso:  
o Multi Layer Perceptron (MLP)

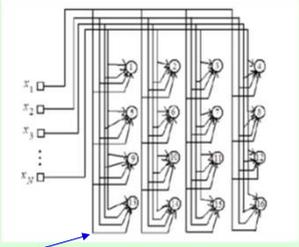
- Múltiplas entradas / Múltiplas saídas / Múltiplas camadas
- Variáveis (internas e externas) analógicas ou digitais
- Relações lineares ou não lineares entre elas

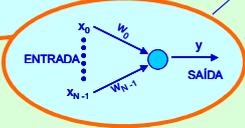


2) Memória  
Associativa  
de Hopfield



3) Mapas Auto-  
Organizáveis  
de Kohonen





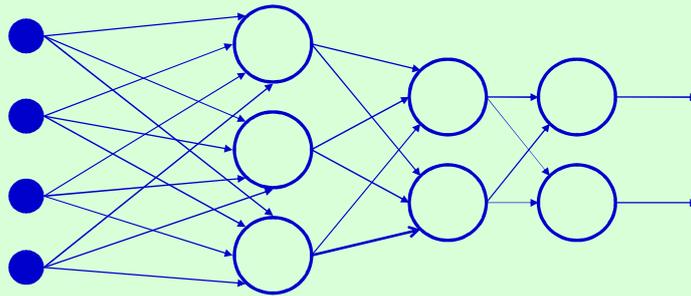
ENTRADA  $x_0$   $w_0$   $y$  SAÍDA  
 $x_{n-1}$   $w_{n-1}$

© Prof. Emilio Del Moral Hernandez

16

## Foco deste Curso: o Multi Layer Perceptron (MLP)

- Múltiplas entradas / Múltiplas saídas / Múltiplas camadas
- Variáveis (internas e externas) analógicas ou digitais
- Relações lineares ou não lineares entre elas



© Prof. Emilio Del Moral Hernandez

19

Aparte ...

*Vocês já ouviram falar de ...*

- Deep Learning?
- Redes Neurais Profundas?
- Redes Neurais Convolucionais?

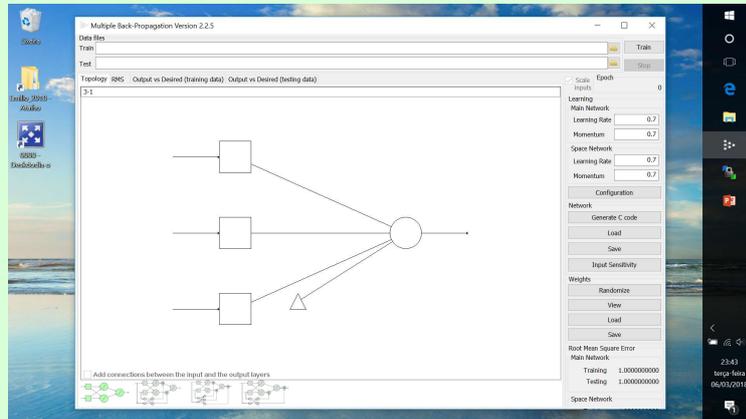
*Pois é ... São temas bem quentes do momento e que têm muito a ver com a arquitetura MLP que vocês estão aprendendo aqui em PSI3471!!*

© Prof. Emilio Del Moral Hernandez

20

## Instale o MBP e digite "3-1" no campo Topology

21

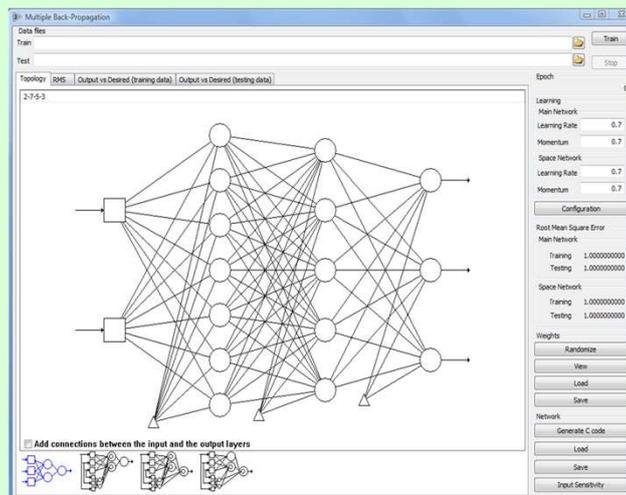


© Prof. Emilio Del Moral Hernandez

21

## Exemplo de tela do ambiente MBP definindo uma Rede Neural do tipo MLP – Topology "2-7-5-3"

22



© Prof. Emilio Del Moral Hernandez

22

## Exemplo de tela do ambiente MBP definindo uma Rede Neural do tipo MLP – Topology “2-7-5-3”

23

Multiple Back-Propagation

Data Files

Train [Train]

Test [Stop]

Topology: RMS | Output vs Desired (training data) | Output vs Desired (testing data)

Epoch: 0

Learning Rate: 0.7

Momentum: 0.7

Space Network Learning Rate: 0.7

Momentum: 0.7

Configuration

Root Mean Square Error

Main Network Training: 1.0000000000

Testing: 1.0000000000

Space Network Training: 1.0000000000

Testing: 1.0000000000

Weights

Randomize

View

Load

Save

Network

Generate C code

Load

Save

Input Sensitivity

Escolha MLPs

© Prof. Emilio Del Moral Hernandez

23

23

## Algumas Telas do MBP Mudando a função do nó neural

24

Multiple Back-Propagation

Data Files

Train [Train]

Test [Stop]

Topology: RMS | Output vs Desired (training data) | Output vs Desired (testing data)

Epoch: 0

Learning Rate: 0.7

Momentum: 0.7

Space Network Learning Rate: 0.7

Momentum: 0.7

Configuration

Root Mean Square Error

Main Network Training: 1.0000000000

Testing: 1.0000000000

Space Network Training: 1.0000000000

Testing: 1.0000000000

Weights

Randomize

View

Load

Save

Network

Generate C code

Load

Save

Input Sensitivity

Output layer activation function: Sigmoid

$f(x) = \frac{1}{1 + e^{-kx}}$

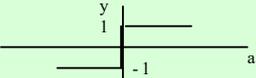
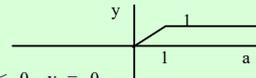
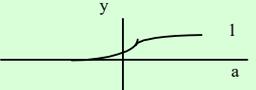
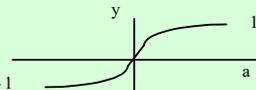
© Prof. Emilio Del Moral Hernandez

24

24

## Outras funções de transferência não linear (nem todas disponíveis no ambiente MBP)

25

<p><b>Hard Limiter</b> (limite rápido)</p>  <p> <math>a &lt; 0, y = -1</math>  <math>a &gt; 0, y = 1</math> </p>	<p><b>Ramping Function</b> (função de rampa)</p>  <p> <math>s &lt; 0, y = 0</math>  <math>0 \leq a \leq 1, y = a</math>  <math>a &gt; 1, y = 1</math> </p>
<p><b>Sigmoide Function "type I"</b> (função sigmoide I)</p>  <p> <math>y = 1 / (1 + e^{-a})</math> </p>	<p><b>Sigmoide Function "type II"</b> (função sigmoide II)</p>  <p> <math>a \geq 0, y = 1 - 1 / (1 + a)</math>  <math>a &lt; 0, y = -1 + 1 / (1 - a)</math> </p>

- Com escalamento do argumento, pode-se abarcar os universos digital e analógico / linear e não linear simultaneamente

© Prof. Emilio Del Moral Hernandez

25

25

## MBP – uma plataforma didática para redes neurais gratuita, de fácil uso e com 12 excelentes tutoriais

26

site <http://mbp.sourceforge.net/>



**TUTORIAL**

1. Introduction (includes the MBP Algorithm)
2. Creating the training and the test datasets
3. Defining the topology of the neural networks
4. Configuring the activation functions of the neurons
5. Defining the neural network learning configuration
6. Training a neural network - Part I (regression)
7. Training a neural network - Part II (classification)
8. Copying data and graphics
9. Initialize, view, save and load the neural network weights
10. Load and save a neural network
11. Generate C code from a trained neural network
12. Analyzing the input sensitivity of a neural network

Ambiente desenvolvido pelo Prof. Noel Lopes e colaboradores  
– Instituto Politécnico da Guarda – Portugal

© Prof. Emilio Del Moral Hernandez

26

26

## Tutorial 2 – criando 2 conjuntos empíricos, de treino e de teste

27

Multiple Back-Propagation

About Screenshots Download Tutorial News Papers Develop/Contact

Multiple Back-Propagation

File Edit View Help

Train

Test

Epoch

Learning Rate: 0.7

Momentum: 0.7

Spoke Network Learning Rate: 0.7

Momentum: 0.7

Configuration

Root Mean Square Error

Main Network Training: 1.000000000

Testing: 1.000000000

Spoke Network Training: 1.000000000

Testing: 1.000000000

Weights

Randomize

Yes

Load

Save

Network

Generate Code

Load

Save

Input Sensitivity

Tutorial 2  
Creating the training and the test datasets.

If you use this program for any scientific work, or related, please cite one of the references below and if possible send me an e-mail with the citation to acad@igp.pt

Lopes, N. and Ribeiro, B. (2003). An Efficient Gradient-Based Learning Algorithm Applied to Neural Networks with Selective Activation Neurons. In *Neural, Parallel & Scientific Computation*, volume 11, pages 253-272. Dynamic Publishers.

Lopes, N. and Ribeiro, B. (2001). Hybrid learning in a multi neural network architecture. In *INNS-IEEE International Joint Conference on Neural Networks, IJCNN01*, volume 1, pages 101-106. IEEE Press.

In the Articles folder accompanying this program you will find links and the BibTex, ACM Ref and EndNote references for the articles.

Thank you.

This program can be freely obtained on the site <http://igp.pt/MBP>. You should not pay or donate any money for this program. Please read the license accompanying the program.

Add connections between the input and the output layers

© Prof. Emilio Del Moral Hernandez

27

27

Multiple Back-Propagation

About Screenshots Download Tutorial News Papers Develop/Contact

Multiple Back-Propagation

File Edit View Help

Train

Test

Epoch

Learning Rate: 0.7

Momentum: 0.7

Spoke Network Learning Rate: 0.7

Momentum: 0.7

Configuration

Root Mean Square Error

Main Network Training: 1.000000000

Testing: 1.000000000

Spoke Network Training: 1.000000000

Testing: 1.000000000

Weights

Randomize

Yes

Load

Save

Network

Generate Code

Load

Save

Input Sensitivity

Tutorial 2  
Creating the training and the test datasets.

If you use this program for any scientific work, or related, please cite one of the references below and if possible send me an e-mail with the citation to acad@igp.pt

Lopes, N. and Ribeiro, B. (2003). An Efficient Gradient-Based Learning Algorithm Applied to Neural Networks with Selective Activation Neurons. In *Neural, Parallel & Scientific Computation*, volume 11, pages 253-272. Dynamic Publishers.

Lopes, N. and Ribeiro, B. (2001). Hybrid learning in a multi neural network architecture. In *INNS-IEEE International Joint Conference on Neural Networks, IJCNN01*, volume 4, pages 2788-2793. Washington D.C., USA.

In the Articles folder accompanying this program you will find links and the BibTex, ACM Ref and EndNote references for the articles.

For MBP to understand the data files, they must obey to the following conditions:

1. The columns containing the data must be separated by white spaces and its number must remain fixed.
2. The first line of the data file may optionally be a title line, where you may have a description of the columns. Just remember that the title columns cannot have white spaces because they are used as column separators. You may however use any other characters, such as the underscore, to separate words in the title columns.
3. MBP recognizes only numbers as data. Moreover only the dot (.) is recognized as the decimal separator and no thousands separator is recognized.

X	Y
0.218959	0.196178
0.047045	0.162958
0.678865	0.835997
0.679296	0.602405
0.934693	0.623243
0.383502	0.838679
0.519416	0.671648

Add connections between the input and the output layers

© Prof. Emilio Del Moral Hernandez

28

28

29

Para STOA - Alguns : Multiple Back-Propa #bottom Multiple Back-Propa Multiple Back-Propa (342) Laboratorio de +

mbp.sourceforge.net/tutorial/tutorial2.html#bottom

## Multiple Back-Propagation

About Screenshots Download Tutorial News Papers Develop/Contact

Multiple Back-Propagation

Date file: Train: Test: Epoch: Learning Rate: 0.7 Momentum: 0.7

Test: [BMS] [Output vs Desired (training data)] [Output vs Desired (testing data)] Epoch: Learning Rate: 0.7 Momentum: 0.7

Tetrahydrocortisone	Pregnanetriol	Type
3.0	1.3	A
1.9	1.4	A
9.1	0.6	B
9.2	7.9	C
2.6	0.1	A
3.9	0.6	B
3.8	0.2	A

Still you can code characteristics that have non-numeric values transforming them into numeric characteristics.

Local Learning Algorithm Applied to Neural Networks with In Computers, volume 11, pages 253-272. Dynamic.

is neural network architecture. In INNS-IEEE International Joint pages 2768-2793, Washington D.C., USA.

and links and the BbTex, ACM Ref and EndNote references for

Thank you.

This program can be freely obtained on the site <http://ds.fgg.pt/MBP>. You should not pay or donate any money for this program. Please read the license accompanying the program.

Add connections between the input and the output layers

Weights: Randomize, View, Load, Save, Network, Generate C code, Load, Save, Input Sensitivity

© Prof. Emilio Del Moral Hernandez

29

29

30

Para STOA - Alguns : Multiple Back-Propa #bottom Multiple Back-Propa Multiple Back-Propa (342) Laboratorio de +

mbp.sourceforge.net/tutorial/tutorial2.html#bottom

## Multiple Back-Propagation

About Screenshots Download Tutorial News Papers Develop/Contact

Multiple Back-Propagation

Date file: Train: Test: Epoch: Learning Rate: 0.7 Momentum: 0.7

Test: [BMS] [Output vs Desired (training data)] [Output vs Desired (testing data)] Epoch: Learning Rate: 0.7 Momentum: 0.7

Tetrahydrocortisone	Pregnanetriol	Type	Type-A	Type-B	Type-C
3.0	1.3	A	1	0	0
1.9	1.4	A	1	0	0
9.1	0.6	B	0	1	0
9.2	7.9	C	0	0	1
2.6	0.1	A	1	0	0
3.9	0.6	B	0	1	0
3.8	0.2	A	1	0	0

For example, if a variable has three possible values (A, B and C) you can replace it with three columns of data, so that each column would have a value of one for all the lines where the characteristic presents the corresponding values and a value of zero for the remaining lines.

we and if possible send me an

Neural Networks with 253-272. Dynamic.

NS-IEEE International Joint USA.

and EndNote references for

Thank you.

This program can be freely obtained on the site <http://ds.fgg.pt/MBP>. You should not pay or donate any money for this program. Please read the license accompanying the program.

Add connections between the input and the output layers

Weights: Randomize, View, Load, Save, Network, Generate C code, Load, Save, Input Sensitivity

© Prof. Emilio Del Moral Hernandez

30

30

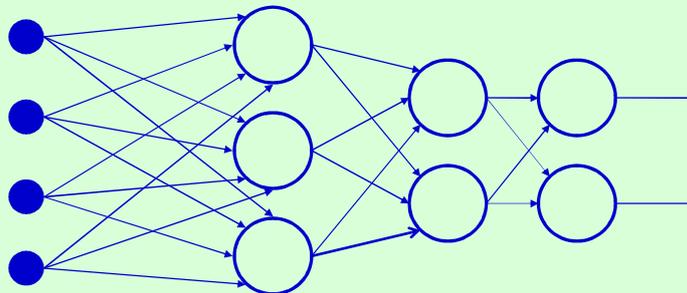
## Tutorial 3 – definindo entradas saídas e topologia da rede neural

© Prof. Emilio Del Moral Hernandez

31

## Foco deste Curso: o Multi Layer Perceptron (MLP)

- Múltiplas entradas / Múltiplas saídas / Múltiplas camadas
- Variáveis (internas e externas) analógicas ou digitais
- Relações lineares ou não lineares entre elas



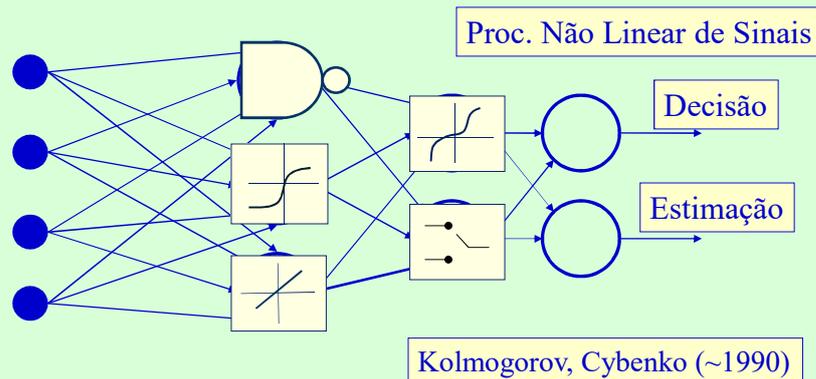
© Prof. Emilio Del Moral Hernandez

34

## O Multi Layer Perceptron (MLP)

35

- Múltiplas entradas / Múltiplas saídas / Múltiplas camadas
- Variáveis (internas e externas) analógicas ou digitais
- Relações lineares ou não lineares entre elas



© Prof. Emilio Del Moral Hernandez

35

35

## Resumindo os aspectos conceituais principais

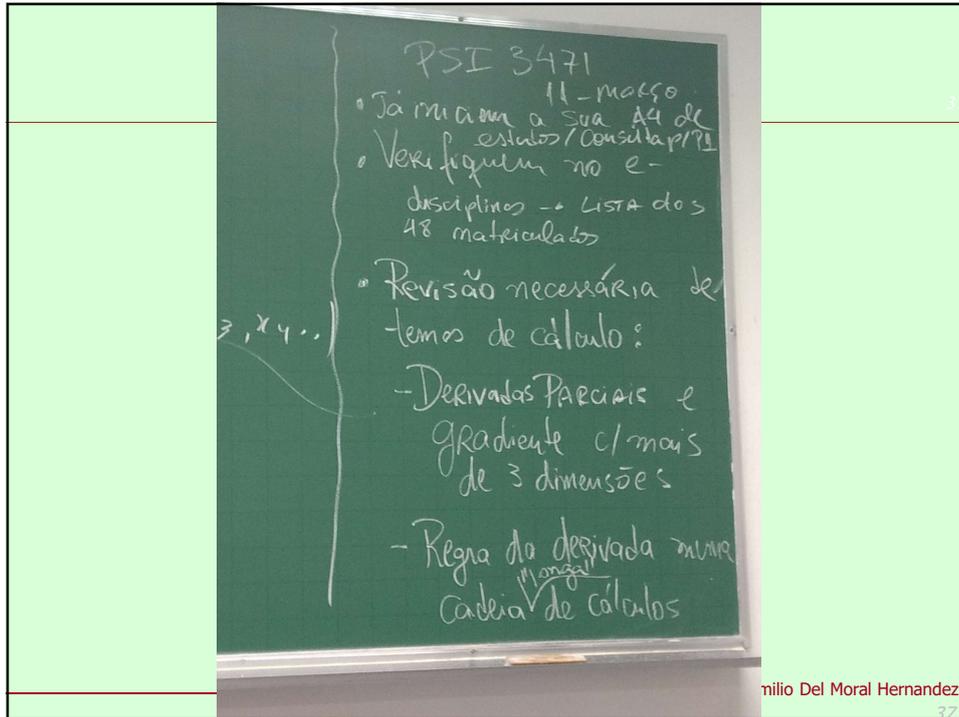
36

- 1) Não linearidade com a função neural sigmoideal
  - 2) Possibilidade de conjugar na mesma estrutura ...
    - Cálculos digitais
    - Cálculos lineares multivariáveis
    - Funções genéricas não lineares multivariáveis
  - 3) Comportamento adaptativo com aprendizado através de exemplos
- 
- Problemas complexos, multidimensionais, não lineares e mesmo aqueles sem teoria conhecida
  - Decisão automática, estimação, reconhecimento de padrões, classificação, processamento não linear de sinais, *clustering* multidimensional ...

© Prof. Emilio Del Moral Hernandez

36

36



37



39

## *De onde vem o grande poder do MLP?*

## Theorem of [Kolmogorov &] Cybenko:

- Kolmogorov:  
Given any  $F$  of many variables  $x_1, x_2, x_3, x_4 \dots$  for example, the complicated  $F = [x_1 \cdot \sin(x_2) + \log(x_3)] / x_4 + \text{etc} \dots$  or any other  $F$ , the following approximation can always be obtained ...

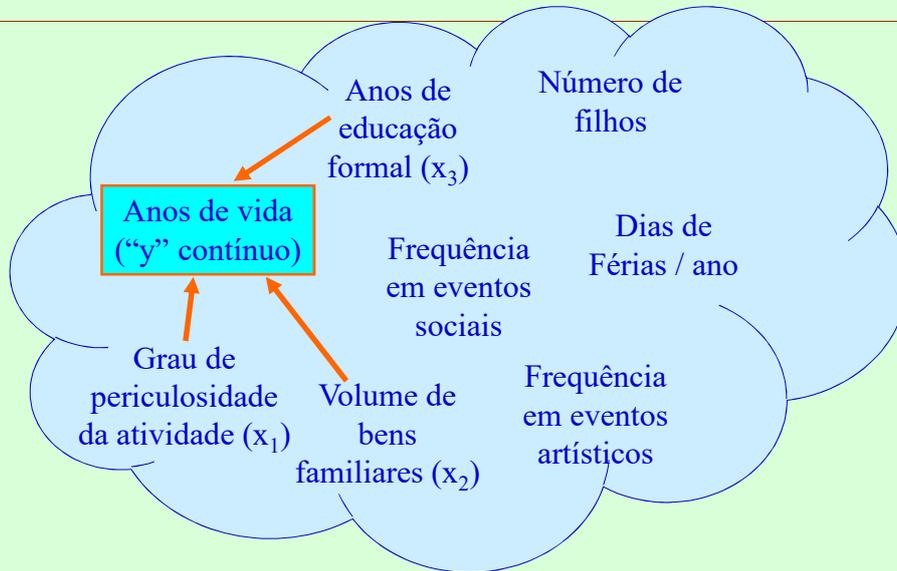
$F(x_1, x_2, x_3, x_4 \dots) \sim$  linear combination and composition of a finite (limited) number of functions  $g_k(v)$  of just one variable  $v$ , and we can have arbitrary precision in the approximation of  $F$

- Cybenko: adapted Kolmogorov for the particular case in which the single argument functions  $g_k$  are approximated by a sum of sigmoidal functions ... he noticed that several sigmoids shifted and scaled properly can approximate any  $g_k$ (scalar argument)

**Cybenko concluded that any arbitrary  $F$  CAN be "implemented" by an ANN with sigmoidal nodes and just 1 hidden layer!!**

## Um hipotético universo de variáveis interdependentes, passível de modelagem/ens

42



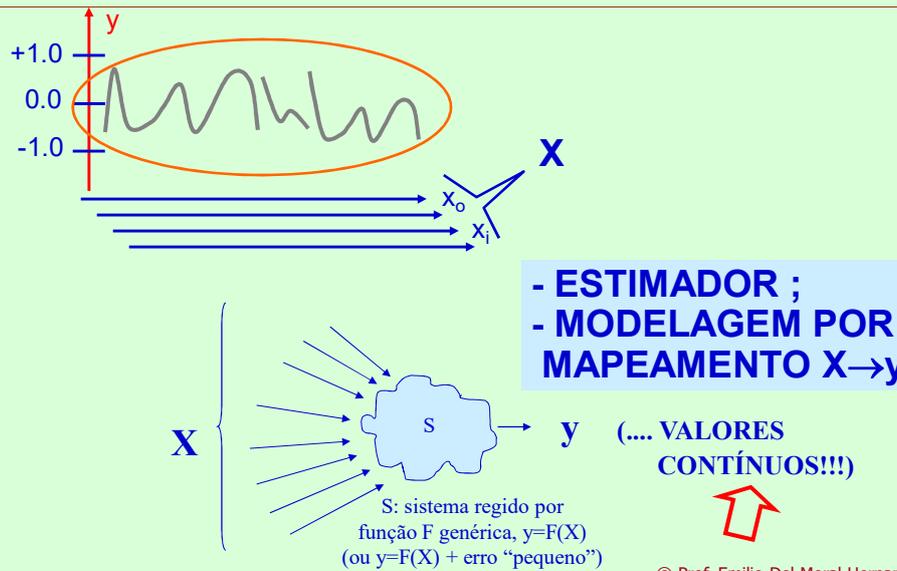
© Prof. Emilio Del Moral Hernandez

42

42

## A função $y(X)$ "a descobrir", num caso geral de função contínua $y(X)$ ....

43



© Prof. Emilio Del Moral Hernandez

43

43

## Cybenko – Enunciado da Prova ... (premissas + resultado)

The screenshot shows the Wikipedia article for the 'Universal approximation theorem'. The title is 'Cybenko – Enunciado da Prova ... (premissas + resultado)'. The article text includes the following:

In the mathematical theory of artificial neural networks, the **universal approximation theorem** states<sup>[1]</sup> that a feed-forward network with a single hidden layer containing a finite number of neurons (i.e., a multilayer perceptron), can approximate continuous functions on compact subsets of  $\mathbb{R}^n$ , under mild assumptions on the activation function. The theorem thus states that simple neural networks can represent a wide variety of interesting functions when given appropriate parameters; it does not touch upon the algorithmic learnability of those parameters.

One of the first versions of the theorem was proved by George Cybenko in 1989 for sigmoid activation functions<sup>[2]</sup>. Kurt Hornik showed in 1991<sup>[3]</sup> that it is not the specific choice of the activation function, but rather the multilayer feedforward architecture itself which gives neural networks the potential of being universal approximators. The output units are always assumed to be linear. For notational convenience, only the single output case will be shown. The general case can easily be deduced from the single output case.

**Formal statement** [edit]

The theorem<sup>[3][4][5]</sup> in mathematical terms:

Let  $\varphi(\cdot)$  be a nonconstant, bounded, and monotonically-increasing continuous function. Let  $I_m$  denote the  $m$ -dimensional unit hypercube  $[0, 1]^m$ . The space of continuous functions on  $I_m$  is denoted by  $C(I_m)$ . Then, given any function  $f \in C(I_m)$  and  $\epsilon > 0$ , there exist an integer  $N$  and real constants  $a_i, b_i \in \mathbb{R}, w_i \in \mathbb{R}^m$ , where  $i = 1, \dots, N$  such that we may define:

$$F(x) = \sum_{i=1}^N \alpha_i \varphi(w_i^T x + b_i)$$

as an approximate realization of the function  $f$  where  $f$  is independent of  $\varphi$ , that is,

$$\|F(x) - f(x)\| < \epsilon$$

for all  $x \in I_m$ . In other words, functions of the form  $F(x)$  are dense in  $C(I_m)$ .

**References** [edit]

- ↑ Balazs Csordás Csáji: Approximation with Artificial Neural Networks; Faculty of Sciences, Eötvös Loránd University, Hungary
- ↑ ↑ Cybenko, G. (1989) "Approximations by superpositions of sigmoidal functions". *Mathematics of Control, Signals, and Systems*, 2 (4), 303-314
- ↑ ↑ Kurt Hornik (1991) "Approximation Capabilities of Multilayer Feedforward Networks". *Neural Networks*, 4(2), 251-257
- ↑ Haykin, Simon (1998): *Neural Networks: A Comprehensive Foundation*, Volume 2, Prentice Hall (ISBN 0-13-27350-1)
- ↑ Hassoun, M. (1995) *Fundamentals of Artificial Neural Networks* MIT Press, p. 48

↑ This applied mathematics-related article is a stub. You can help Wikipedia by expanding it.

Categories: Theorems in discrete mathematics | Artificial neural networks | Neural networks | Network architecture | Networks | Information, knowledge, and uncertainty | Applied mathematics stubs

The page was last modified on 1 June 2014, at 20:06.

Fwd: Proposta ...eml | Alterações vag...doc | [Mostrar todos os downloads...](#)

© Prof. Emilio Del Moral Hernandez

44

45

## Entendamos ...

- as premissas da demonstração de Cybenko
- a notação não muito familiar a nós que ele usou
- o quão poderoso é o resultado que ele obteve
- como com passos simples podemos estender a sua aplicação (ou ... *relaxando algumas das (apenas aparentes) limitações impostas nas premissas*)

© Prof. Emilio Del Moral Hernandez

45

Kurt Hornik showed in 1991<sup>[2]</sup> that it is not the specific choice of the  $\varphi$  assumed to be linear. For notational convenience, only the single out

## Formal statement [\[edit\]](#)

The theorem<sup>[2][3][4][5]</sup> in mathematical terms:

Let  $\varphi(\cdot)$  be a nonconstant, bounded, and monotonically-increasing function in  $C(I_m)$  and  $\epsilon > 0$ , there exist an integer  $N$  and real constants  $\alpha_i, b_i$

$$F(x) = \sum_{i=1}^N \alpha_i \varphi(w_i^T x + b_i)$$

as an approximate realization of the function  $f$  where  $f$  is independent of  $N$

$$|F(x) - f(x)| < \epsilon$$

for all  $x \in I_m$ . In other words, functions of the form  $F(x)$  are dense in  $C(I_m)$ .

46

Kurt Hornik showed in 1991<sup>[2]</sup> that it is not the specific choice of the  $\varphi$  assumed to be linear. For notational convenience, only the single out

## Formal statement [\[edit\]](#)

The theorem<sup>[2][3][4][5]</sup> in mathematical terms:

Let  $\varphi(\cdot)$  be a nonconstant, bounded, and monotonically-increasing function in  $C(I_m)$  and  $\epsilon > 0$ , there exist an integer  $N$  and real constants  $\alpha_i, b_i$

$$F(x) = \sum_{i=1}^N \alpha_i \varphi(w_i^T x + b_i)$$

as an approximate realization of the function  $f$  where  $f$  is independent of  $N$

$$|F(x) - f(x)| < \epsilon$$

for all  $x \in I_m$ . In other words, functions of the form  $F(x)$  are dense in  $C(I_m)$ .

47

Kurt Hornik showed in 1991<sup>[2]</sup> that it is not the specific choice of the  $\varphi$  assumed to be linear. For notational convenience, only the single out

## Formal statement [edit]

The theorem<sup>[2][3][4][5]</sup> in mathematical terms:

$Y_{rede}(X)$

$X$

Let  $\varphi(\cdot)$  be a nonconstant, bounded, and monotonically-increasing function in  $C(I_m)$  and  $\epsilon > 0$ , there exist an integer  $N$  and real constants  $\alpha_i$  and  $b_i$

$$F(x) = \sum_{i=1}^N \alpha_i \varphi(w_i^T x + b_i)$$

número de nós escondidos

sigmoidal

viés; : viés do nó escondido  $i$

$W_i$ : vetor de pesos do nó escondido  $i$

elementos do vetor de pesos do nó linear de saída  $W_s$

$$|F(x) - f(x)| < \epsilon$$

for all  $x \in I_m$ . In other words, functions of the form  $F(x)$  are dense in  $C(I_m)$ .

48

Kurt Hornik showed in 1991<sup>[2]</sup> that it is not the specific choice of the  $\varphi$  assumed to be linear. For notational convenience, only the single out

## Formal statement [edit]

The theorem<sup>[2][3][4][5]</sup> in mathematical terms:

Let  $\varphi(\cdot)$  be a nonconstant, bounded, and monotonically-increasing function in  $C(I_m)$  and  $\epsilon > 0$ , there exist an integer  $N$  and real constants  $\alpha_i$  and  $b_i$

$$F(x) = \sum_{i=1}^N \alpha_i \varphi(w_i^T x + b_i)$$

$Y_{rede}(X)$

Fescondida\_sistema(X)

as an approximate realization of the function  $f$  where  $f$  is independent of  $x$

Limite de erro

$$|F(x) - f(x)| < \epsilon$$

for all  $x \in I_m$ . In other words, functions of the form  $F(x)$  are dense in  $C(I_m)$ .

49

# Cybenko – a prova matemática, disponível para download na internet, é bastante complexa

Math. Control Signals Systems (1989) 2: 303-314

Mathematics of Control, Signals, and Systems  
© 1989 Springer-Verlag New York Inc.

**Approximation by Superpositions of a Sigmoidal Function\***

G. Cybenko†

**Abstract.** In this paper we demonstrate that finite linear combinations of superpositions of a fixed, univariate function and a set of affine functions can uniformly approximate any continuous function of  $n$  real variables with support in the unit hypercube, only mild conditions are imposed on the univariate function. Our results settle an open question about representability in the class of single hidden layer neural networks. In particular, we show that arbitrary decision regions can be arbitrarily well approximated by continuous feedforward neural networks with only a single interval, hidden layer and any continuous sigmoidal nonlinearity. The paper discusses approximation properties of other possible types of nonlinearities that might be implemented by artificial neural networks.

**Key words.** Neural networks, Approximation, Completeness.

**1. Introduction**

A number of diverse application areas are concerned with the representation of general functions of an  $n$ -dimensional real variable,  $x \in \mathbb{R}^n$ , by finite linear combinations of the form

$$\sum_{j=1}^m \alpha_j \sigma(y_j^T x + \theta_j), \quad (1)$$

where  $y_j \in \mathbb{R}^n$  and  $\alpha_j, \theta_j \in \mathbb{R}$  are fixed. ( $y^T$  is the transpose of  $y$  so that  $y^T x$  is the inner product of  $y$  and  $x$ .) Here the univariate function  $\sigma$  depends heavily on the context of the application. Our major concern is with so-called sigmoidal  $\sigma$ 's.

$$\sigma(t) \rightarrow \begin{cases} 1 & \text{as } t \rightarrow +\infty, \\ 0 & \text{as } t \rightarrow -\infty. \end{cases}$$

Such functions arise naturally in neural network theory as the activation function of a neural node (or neur as it is becoming the preferred term [L1], [RHM]). The main result of this paper is a demonstration of the fact that sums of the form (1) are dense in the space of continuous functions on the unit cube if  $\sigma$  is any continuous sigmoidal

\* Date received: October 21, 1988. Date revised: February 17, 1989. This research was supported in part by NSF Grant DCR-8619103, ONR Contract N00010-86-G-0202 and DOE Grant DE-FG02-83ER22001.  
† Center for Supercomputing Research and Development and Department of Electrical and Computer Engineering, University of Illinois, Urbana, Illinois 61801, U.S.A.

310

G. Cybenko

313

**4. Results for Other Activation Functions**

In this section we discuss other classes of activation functions that have approximation properties similar to the ones enjoyed by continuous sigmoidals. Since these other examples are of somewhat less practical interest, we only sketch the corresponding proofs.

There is considerable interest in discontinuous sigmoidal functions such as hard limiters ( $\sigma(x) = 1$  for  $x \geq 0$  and  $\sigma(x) = 0$  for  $x < 0$ ). Discontinuous sigmoidal functions are not used as often as continuous ones (because of the lack of good training algorithms) but they are of theoretical interest because of their close relationship to classical perceptrons and Gamba networks [MP].

Assume that  $\sigma$  is a bounded, measurable sigmoidal function. We have an analog of Theorem 2 that goes as follows:

**Theorem 4.** Let  $\sigma$  be bounded measurable sigmoidal function. Then finite sums of the form

$$G(x) = \sum_{j=1}^m \alpha_j \sigma(y_j^T x + \theta_j)$$

are dense in  $L^1(I_n)$ . In other words, given any  $f \in L^1(I_n)$  and  $\epsilon > 0$ , there is a sum,  $G(x)$ , of the above form for which

$$\|G - f\|_{L^1} = \int_{I_n} |G(x) - f(x)| dx < \epsilon.$$

The proof follows the proof of Theorems 1 and 2 with obvious changes such as replacing continuous functions by integrable functions and using the fact that  $L^1(I_n)$  is the dual of  $L^\infty(I_n)$ . The notion of being discriminatory accordingly changes to the following: for  $h \in L^\infty(I_n)$  the condition that

$$\int_{I_n} \sigma(y^T x + \theta) h(x) dx = 0$$

for all  $y$  and  $\theta$  implies that  $h(x) = 0$  almost everywhere. General sigmoidal functions are discriminatory in this sense as already seen in Lemma 1 because measures of the form  $h(x) dx$  belong to  $M(I_n)$ .

Since convergence in  $L^1$  implies convergence in measure [A], we have an analog of Theorem 3 that goes as follows:

**Theorem 5.** Let  $\sigma$  be a general sigmoidal function. Let  $f$  be the decision function for any finite measurable partition of  $I_n$ . For any  $\epsilon > 0$ , there is a finite sum of the form

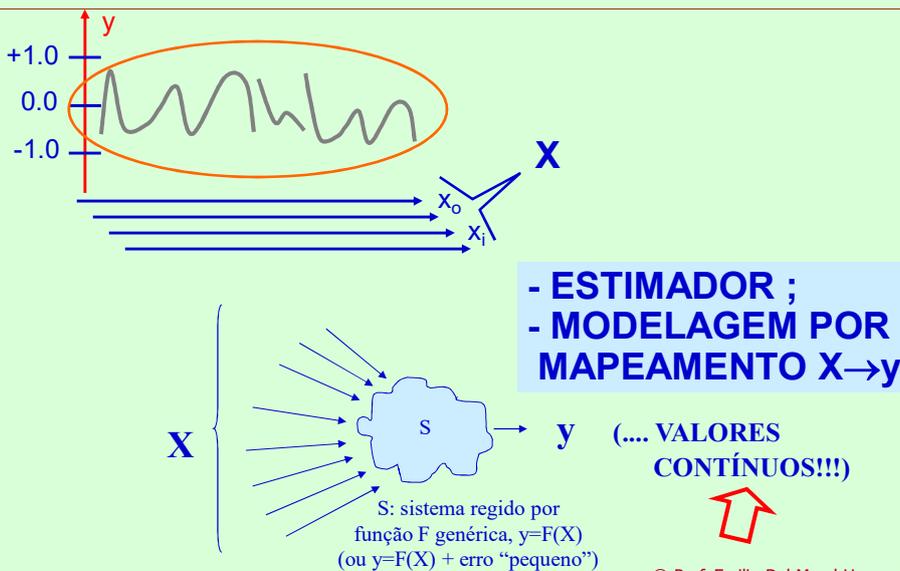
$$G(x) = \sum_{j=1}^m \alpha_j \sigma(y_j^T x + \theta_j)$$

and a set  $D \subset I_n$ , so that  $m(D) \geq 1 - \epsilon$  and

$$|G(x) - f(x)| < \epsilon \quad \text{for } x \in D.$$

50

# A função $y(X)$ "a descobrir", num caso geral de função contínua $y(X)$ ....



© Prof. Emilio Del Moral Hernandez

51

**Um hipotético universo de variáveis interdependentes, passível de modelagem/ens**

Vamos pensar em nossos próprios desafios de regressão multivariada (com, digamos ao menos 3 variáveis de "entrada" no modelo) para a previsão / estimação de alguma grandeza relevante?  
Façamos isto no contexto de Eletrônica / Sistemas / Sua ênfase.

**ATIVIDADE EM SALA, COM DEBATE COM COLEGAS E ENTREGA ESCRITA**

© Prof. Emilio Del Moral Hernandez

**Um hipotético universo de variáveis interdependentes, passível de modelagem/ens**

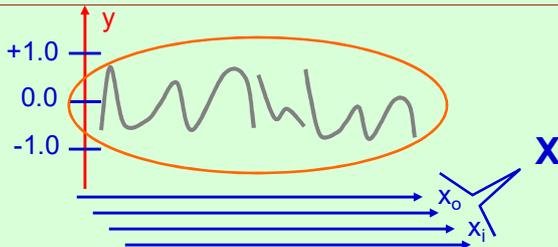
Vamos pensar em nossos próprios desafios de reconhecimento multivariado (com, digamos ao menos 4 variáveis de entrada no modelo) para uma classificação / reconhecimento / categorização automática relevante?  
Façamos isto no contexto de Eletrônica & Sistemas / Sua ênfase.

**ATIVIDADE EM SALA, COM DEBATE COM COLEGAS E ENTREGA ESCRITA**

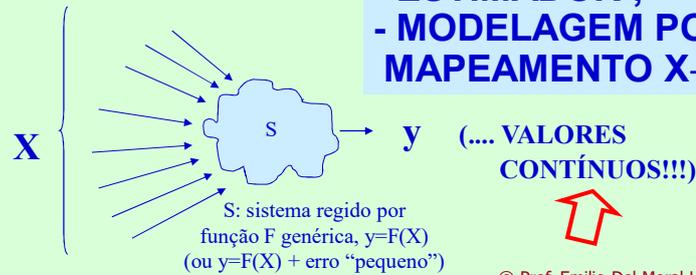
© Prof. Emilio Del Moral Hernandez

52

**A função  $y(X)$  "a descobrir", num caso geral de função contínua  $y(X)$  ....**



**- ESTIMADOR ;  
- MODELAGEM POR  
MAPEAMENTO  $X \rightarrow y$**



53

Curso: PSI3471 - Funda

https://edisciplinas.usp.br/course/view.php?id=66581#section-2

### Materiais do prof. Emilio (15 aulas de 18-fev a 22-abr + P1)

Slides apresentados e outros materiais de apoio às atividades em sala serão disponibilizados aqui.

Atenção: Os materiais de apoio disponibilizados via STOA **NÃO Substituem as discussões e o aprendizado que ocorrem com a sua presença nas aulas desta disciplina**. Se tiver alguma dúvida sobre a necessidade da sua presença em sala de aula, estou disponível para ajudá-lo a entender; me procure após encerrarmos a nossa próxima aula e conversamos sobre isso.

- 📄 Ementa no sistema Jupiter
- 📄 Preliminar - Spoiler de Slides 2019-PSI3471 - I - Introducao; Aulas 1 a 4; e mais - ProfEmilioDelMoralHernandez
- 📄 Instale o ambiente MBP no seu computador Windows (instale / use máquina virtual se seu computador for Apple; ou use os computadores da sala C1-10 ou similar)
- 📄 Enunciado de atividade em sala (com entrega), realizada na aula de 20-fev (se perdeu, converse pessoalmente comigo (Prof) ao fim da próxima aula)
- 📄 Enunciado de atividade em sala (com entrega), realizada na aula de 27-fev (se perdeu, converse pessoalmente comigo (Prof) ao fim da próxima aula)
- 📄 Escaneinho de entrega para atividade #3 com data (pós-carnaval) ainda por definir - enunciado preliminar no PDF de entrada deste escaneinho.
- 📄 Complemento ao enunciado preliminar da tarefa #3 - orientações adicionais / dicas
- 📄 Caro/a aluno/a, confira aqui se está matriculado/a ou apenas inscrito/Pendente (neste caso, você precisa contatar sua secretaria de ênfase e informar-se de providências a tomar; se não reverter o "pendente", pode ser excluído) - atualizado em 12/03

🔒 Restrito Disponível se: Você faz parte de T-PSI3471-2019101

Exemplos de sistemas inteligentes usando técnicas de regressão multivariada e/ou reconhecimento multivariado (materiais usados como exemplos nas aulas do Prof Emilio)

© Prof. Emilio Del Moral Hernandez

55

55

Concluindo com um 3º item e entregando a atividade criativa ...  
Registrar adequadamente em A4 as duas propostas (regressor e reconhecedor) e em 3º item formalizar uma ideia possível para seu TCC envolvendo regressão multivariada e/ou reconhecimento de padrões multivariado

**PRELIMINAR, para você já ir adiantando**

© Prof. Emilio Del Moral Hernandez

56

56

PSI3471 Sistemas Inteli

https://edisciplinas.usp.br/mod/page/view.php?id=2531708

DISCIPLINAS USP Disciplinas > Suporte > Idioma > Emilio Del Moral Hernandez

Início > Meus Ambientes > 2019 > EP > PSI > PSI3471 Sistemas Inteligentes 1 (2019) > Materiais do prof. Emilio (15 aulas de 18-fev a 22... > Complemento ao enunciado preliminar da tarefa #3 - ...

### Complemento ao enunciado preliminar da tarefa #3 - orientações adicionais / dicas

Como esclarecido em sala de aula, o enunciado preliminar é mantido em sua essência de focar numa possível proposta de projeto de formatura que envolva regressor/es multivariado/s e/ou reconhecedor/es multivariado/s.

Esclarecimentos / contornos adicionais a esse enunciado de base:

- 1- Organize adequadamente o documento com capa contendo autor, disciplina, professor, data da submissão de sua tarefa, etc etc.
- 2- O(s) regressor(es) e/ou reconhecedor(es) podem ser uma parte do seu projeto total, não precisam ser o TCC todo. Esclareça em detalhes que parte/s do seu projeto total corresponde ao/s regressor/es / reconhecedor/es
- 3- Liste claramente quais e quantas são as variáveis de entrada e de saída de cada regressor / reconhecedor.

**Extras bastante desejáveis**

- Inclua ao fim do documento alguma/s referência/s de sua busca bibliográfica mínima no tema do seu projeto, com um parágrafo simples de destaques seus da importância dessa referência bibliográfica para você.
- Explícite o nome completo de ao menos um colega da turma com o qual tenha trocado ideias sobre a viabilidade e a utilidade de seu/s regressor/es / reconhecedor/es ; faça um relato dos resultados dessa discussão.

**Extra extra:**

- Faça um ensaio computacional básico, mesmo que com dados empíricos fictícios (use ao menos 10 pares empíricos X & y), do/s regressor/es / reconhecedor/es envolvido/s, usando o ambiente MBP e documentando com capturas de telas adequadas o seu pré-ensaio, acompanhadas de comentários seus simples que expliquem cada figura.

Última atualização: quarta, 13 mar 2019, 07:27

© Prof. Emilio Del Moral Hernandez

57

Curso: PSI3471 - Funda

https://edisciplinas.usp.br/course/view.php?id=66581#section-2

Preliminar - Spoiler de Slides 2019-PSI3471 - I - Introducao; Aulas 1 a 4; e mais - ProfEmilioDelMoralHernandez

Instale o ambiente MBP no seu computador Windows (instale / use máquina virtual se seu computador for Apple; ou use os computadores da sala C1-10 ou similar)

Enunciado de atividade em sala (com entrega), realizada na aula de 20-fev (se perdeu, converse pessoalmente comigo (Prof) ao fim da próxima aula)

Enunciado de atividade em sala (com entrega), realizada na aula de 27-fev (se perdeu, converse pessoalmente comigo (Prof) ao fim da próxima aula)

Escaneinho de entrega para atividade #3 com data (pós-carnaval) ainda por definir - enunciado preliminar no PDF de entrada deste escaneinho.

Complemento ao enunciado preliminar da tarefa #3 - orientações adicionais / dicas

Caro/a aluno/a, confira aqui se está matriculado/a ou apenas Inscrito/Pendente (neste caso, você precisa contatar sua secretaria de ênfase e informar-se de providências a tomar; se não reverter o "pendente", pode ser excluído) - atualizado em 12/03

Restrito Disponível se: Você faz parte de T-PSI3471-2019101

**Exemplos de sistemas inteligentes usando técnicas de regressão multivariada e/ou reconhecimento multivariado (materiais usados como exemplos nas aulas do Prof Emilio)**

- Lista / Inventário com Projetos PSI 2672 (disciplina prática de 5o ano) de 2011 ate 2016
- Repositório (36Mb) de apresentações finais de PSI2672 (disciplina de 5o ano) nos anos 2011 a 2016 - perto de 50 projetos de alunos de 5o ano / Prof Emilio Del Moral Hernandez
- RESUMO DAS APRESENTAÇÕES FINAIS DOS PROJETOS DE PSI3571 EM 2018 (Disciplina de 5o ano)
- RESUMO DAS APRESENTAÇÕES FINAIS DOS PROJETOS DE PSI2672 EM 2017 (Disciplina de 5o ano)
- Vitrine com 10 apresentações finais de PSIS886-2018 (Princípios de Neurocomputação - Pós Grady; apresentações finais dos 10 grupos de alunos que cursaram a disciplina em 2018

https://edisciplinas.usp.br/mod/resource/view.php?id=2503337

© Prof. Emilio Del Moral Hernandez

58

**P/ aula que vem: estudar / reestudar os conceitos e a parte operacional de derivadas parciais, do vetor Gradiente, e da regra da cadeia ...**

- **Derivadas parciais (que são as componentes do gradiente):**

$$\frac{\partial f(a,b,c)}{\partial a} \quad \frac{\partial f(a,b,c)}{\partial b} \quad \frac{\partial f(a,b,c)}{\partial c}$$

- **Vetor Gradiente, útil ao método do máximo declive:**

$$\Delta \vec{W} = -\eta \cdot \vec{\nabla} E_{qm}$$

- **Regra da cadeia, necessária ao cálculo de derivadas quando há encadeamento de funções:**

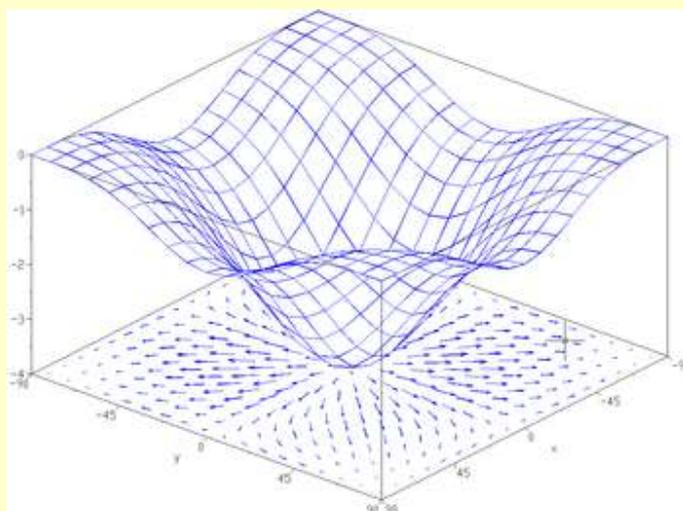
$$\frac{\partial f(g(h(a)))}{\partial a} = \frac{\partial f}{\partial g} \cdot \frac{\partial g}{\partial h} \cdot \frac{\partial h}{\partial a}$$

*EGF-006 – Métodos Numéricos e Reconhecimento de Padrões – Prof. Emilio Del Moral – EPUSP*

59

**Método de otimização por máximo declive / máximo active ...**

*(image from <http://en.wikipedia.org/wiki/Gradient>)*



60

*EGF-006 – Métodos Numéricos e Reconhecimento de Padrões – Prof. Emilio Del Moral – EPUSP*

60

**P/ aula que vem: estudar / reestudar os conceitos e a parte operacional de derivadas parciais, do vetor Gradiente, e da regra da cadeia ...**

- **Derivadas parciais (que são as componentes do gradiente):**

ex: [http://en.wikipedia.org/wiki/Partial\\_derivative](http://en.wikipedia.org/wiki/Partial_derivative)

- **Vetor Gradiente, útil ao método do máximo declive:**

ex: <http://en.wikipedia.org/wiki/Gradient>

- **Regra da cadeia, necessária ao cálculo de derivadas quando há encadeamento de funções:**

ex: [http://en.wikipedia.org/wiki/Chain\\_rule](http://en.wikipedia.org/wiki/Chain_rule)

61

*EGF-006 – Métodos Numéricos e Reconhecimento de Padrões – Prof. Emilio Del Moral – EPUSP*

61

## **Mas ATENÇÃO: Os links anteriores são apenas amostras ...**

Use **Google**, **Wikipedia** e **Youtube** para encontrar materiais adequados às suas necessidades >>>

Youtube, por exemplo, tem uma série vídeos com aulas que podem ajudar no estudo de ....

Derivadas Parciais (palavras de busca sugeridas: “derivada/s parcial/is” ; “partial derivative/s”),

Vetor Gradiente (palavras de busca sugeridas: “vetor gradiente” ; “gradient vector”) e

Regra da Cadeia (palavras de busca sugeridas: “regra da cadeia” ; “chain rule”)

62

*EGF-006 – Métodos Numéricos e Reconhecimento de Padrões – Prof. Emilio Del Moral – EPUSP*

62

**Conexão de RNAs com elementos de “Aprendizado de Máquina” / “Machine Learning”**

**O “aprendizado do modelo” é feito de maneira automática a partir de casos / de exemplos concretos: a definição matemática do modelo é feita a partir de um conjunto rico de exemplos numéricos empíricos de pares  $(X,y)$**

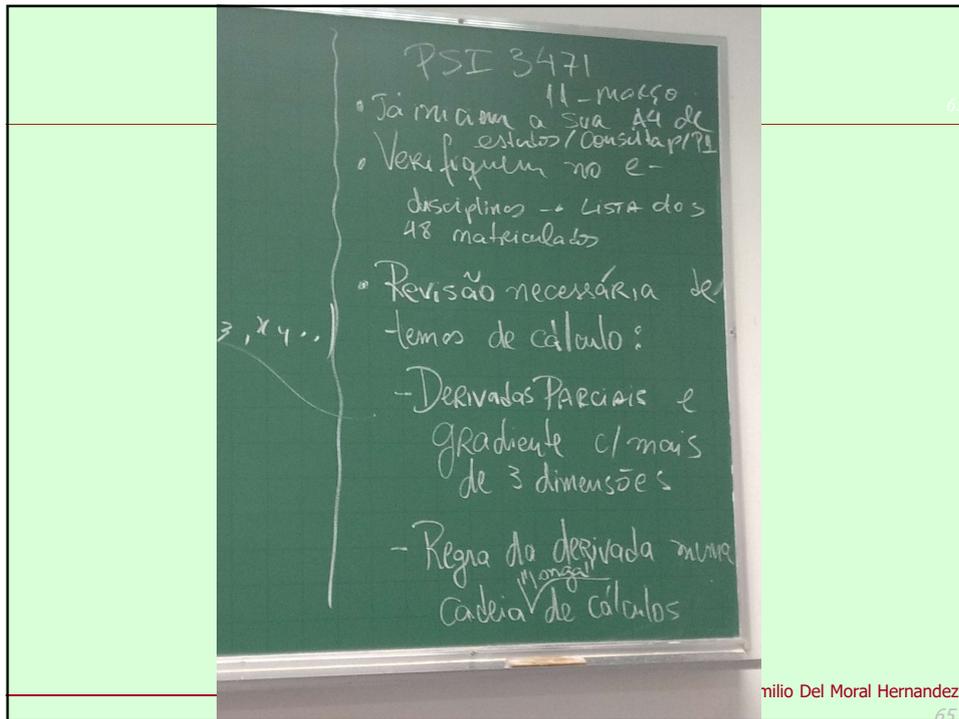
**Conhecimento rico de exemplares / casos  $(X^\mu ; y^\mu)$ :  
Temos M observações empíricas  $(X^\mu ; y^\mu)$ , onde  $\mu$  identifica cada observação, e varia entre 1 a M**



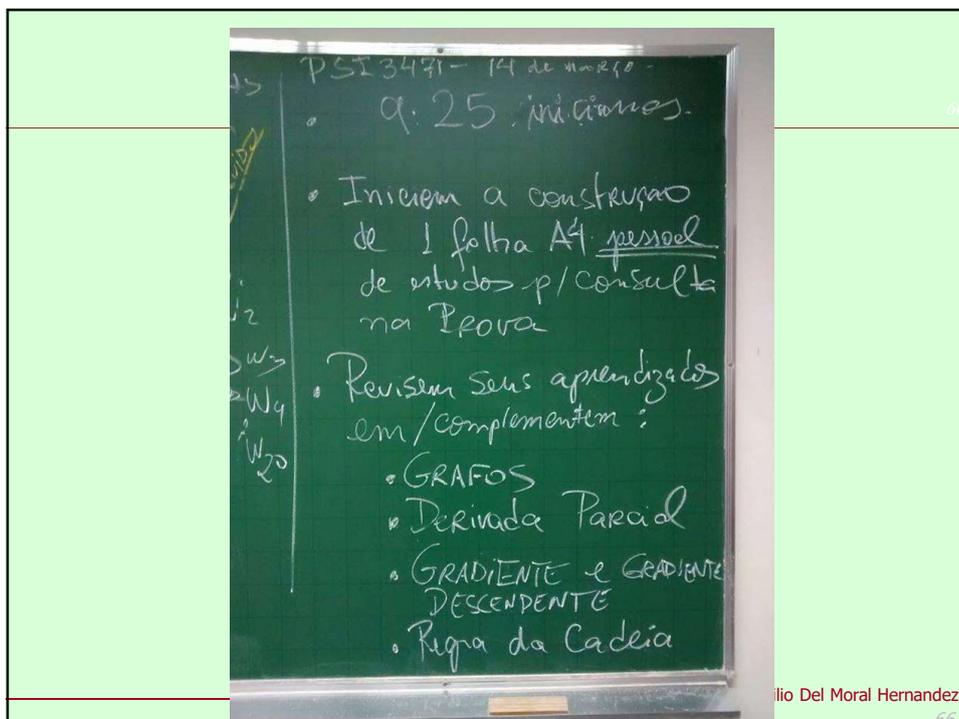
$y_{\text{modelo}}$  = cálculo neural (ou seja, somas ponderadas e com tgh's) que opera sobre as componentes do vetor X  $(x_1, x_2, x_3 \dots)$ ; Esse cálculo neural é calibrado (via escolha dos valores dos ponderadores w's) a partir de M pares empíricos  $(X^\mu ; y^\mu)$

*EGF-006 – Métodos Numéricos e Reconhecimento de Padrões – Prof. Emilio Del Moral – EPUSP*

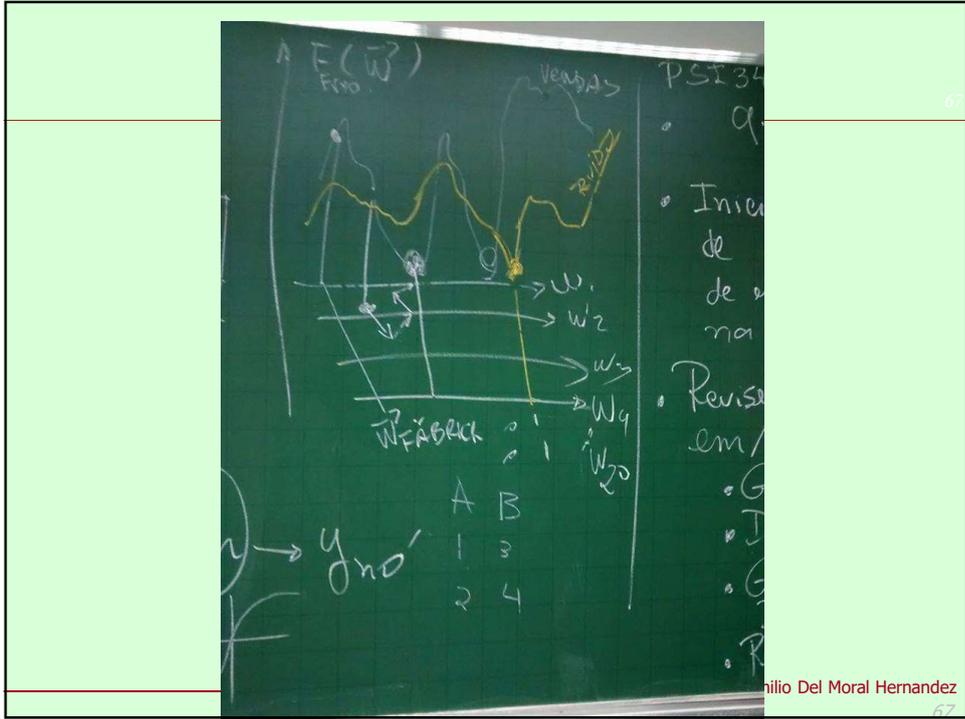
*Como escolhemos os valores dos diversos w's ?*



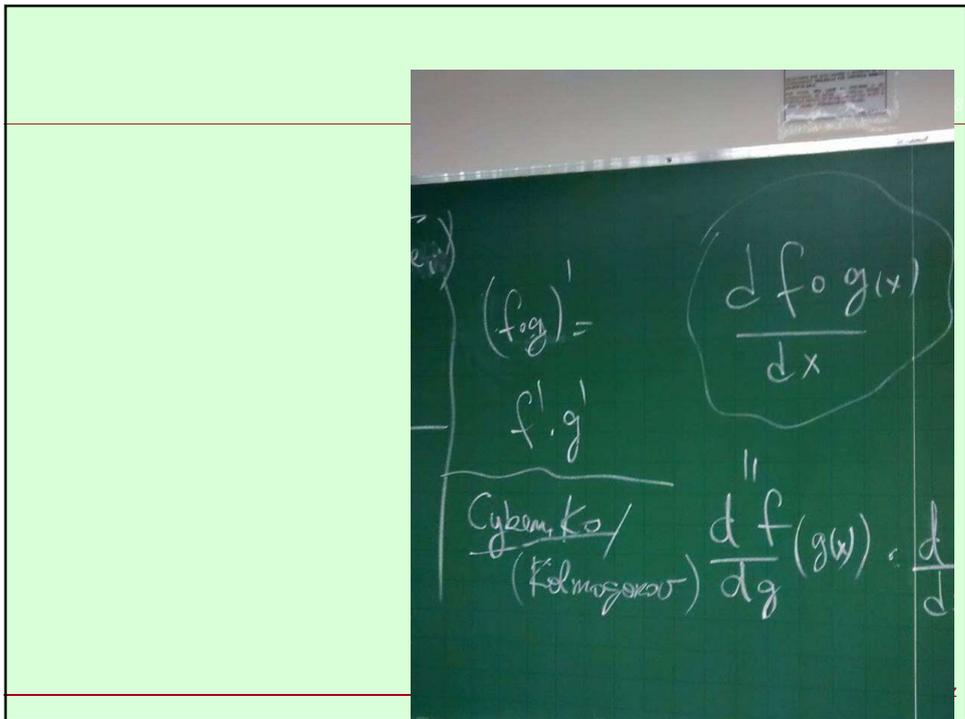
65



66



67

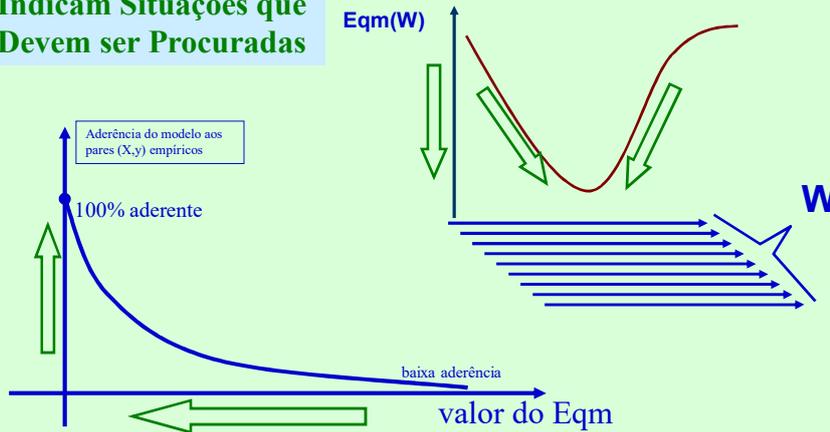


68

O que devemos buscar quando exploramos o espaço de pesos  $W$  buscando que a RNA seja um bom modelo?

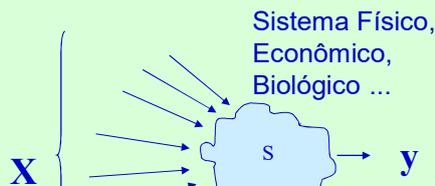
Devemos buscar Maximização da aderência = Mínimo  $Eqm$  possível

As Setas Verdes Indicam Situações que Devem ser Procuradas



69

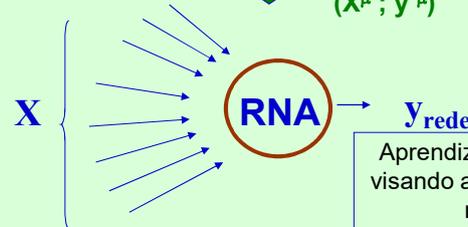
Conjunto de treino em arquiteturas supervisionadas (ex. clássico: MLP com Error Back Propagation)



A computação desejada da rede pode ser definida simplesmente através de observações / exemplares do comportamento requerido

Conjunto de  $M$  observações  $(X^\mu ; y^\mu)$

$$Eqm = \frac{1}{M} \sum_{\mu=1}^M (y_{rede}(\vec{X}^\mu, \vec{W}) - y^\mu)^2$$



$$\Delta \vec{W} = -\eta \cdot \vec{\nabla} Eqm$$

... em loop ...

Aprendizado: Espaço de pesos  $W$  é explorado visando aproximar ao máximo a computação da rede da computação desejada

© Prof. Emilio Del Moral Hernandez

70

71

**Aula 6**  
 (de 15)  
 do Prof. Emilio  
 de PSI3471 de  
**13-mar-2019**  
 se encerrou aqui

© Prof. Emilio Del Moral Hernandez  
71

71

Exemplo de tela do ambiente MBP  
 definindo uma Rede Neural do tipo MLP – Topology “2-7-5-3”

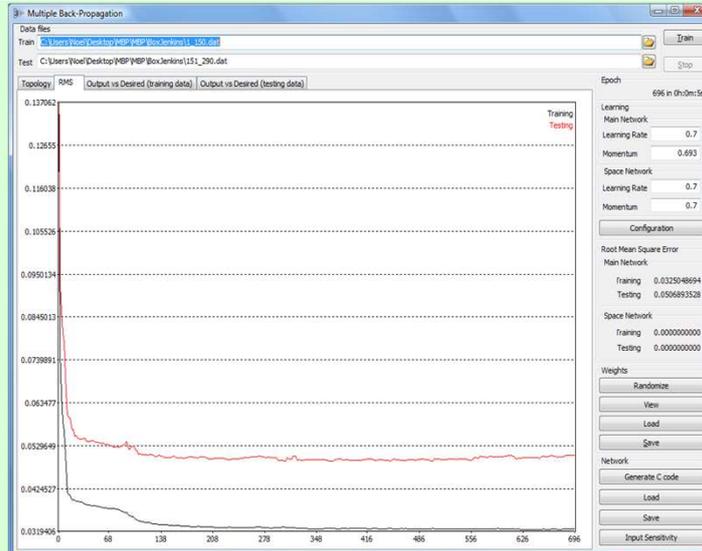
72

© Prof. Emilio Del Moral Hernandez  
72

72

Algumas Telas do MBP: acompanhando a queda do erro (RMS) da rede neural enquanto está aprendendo / adaptando seus pesos  
 (nota: O Eqm é um primo direto do RMS ...  $RMS = Eqm^{1/2}$ )

73



© Prof. Emilio Del Moral Hernandez

73

73

## Aprendizado do MLP por Error Back Propagation ...

$$\vec{\Delta W} = -\eta \cdot \vec{\nabla} Eqm$$

Gradiente de Eqm no espaço de pesos =  $(\partial Eqm(W)/\partial w_1, \partial Eqm(W)/\partial w_2, \partial Eqm(W)/\partial w_3, \dots)$

**Chegando às fórmulas das derivadas parciais, necessárias à Bússola do Gradiente**

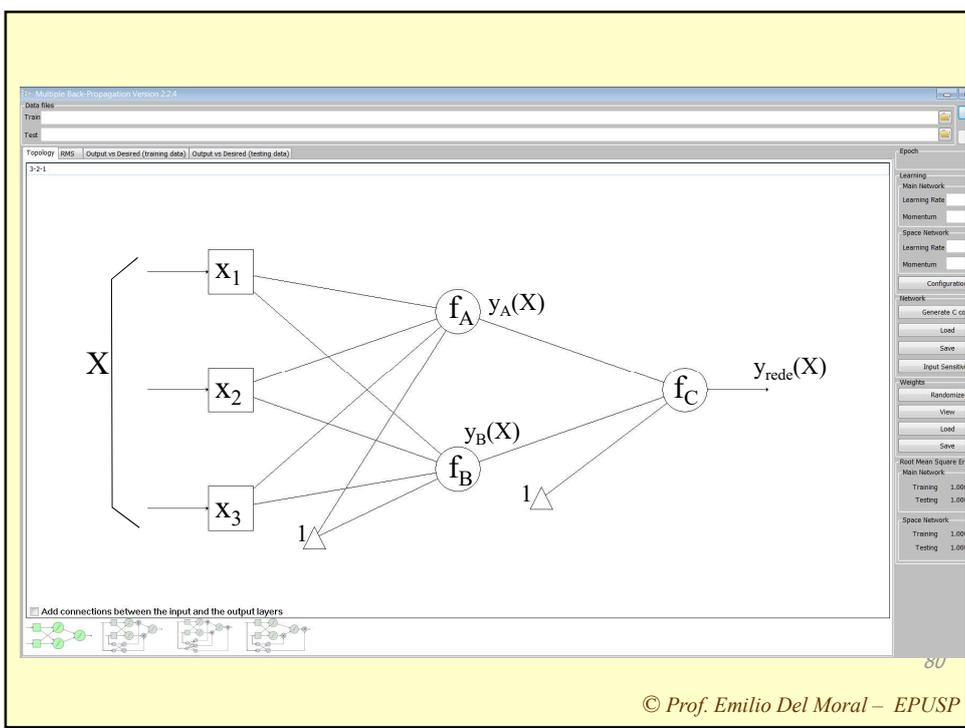
76

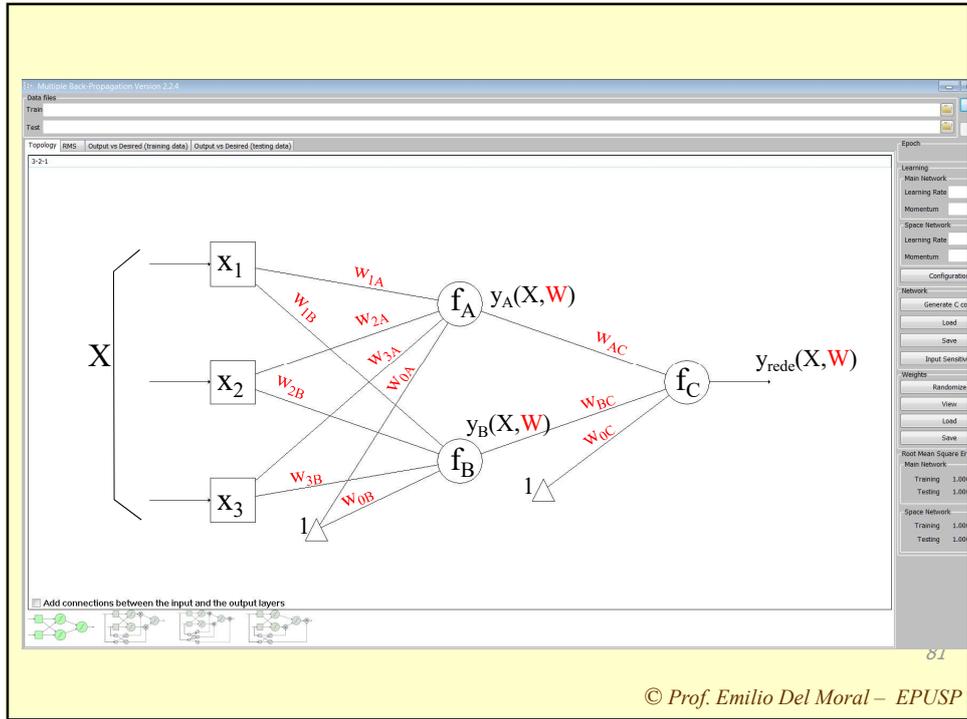
© Prof. Emilio Del Moral – EPUSP

76

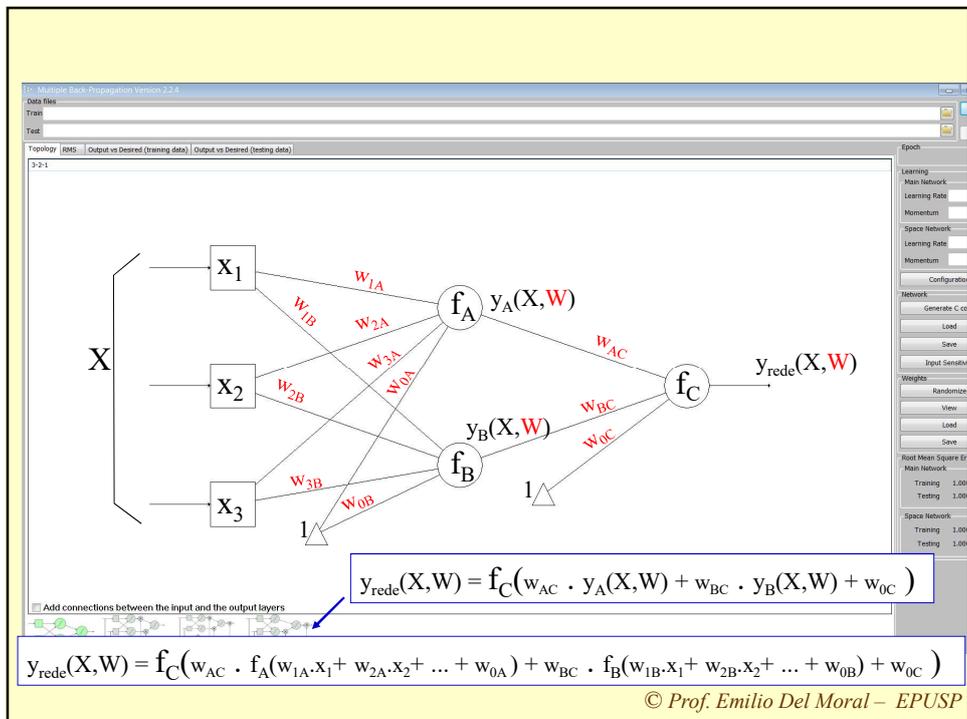
Aplainando o terreno ... Para o gradiente descendente ...

*Vamos falar um pouco sobre GRAFOS?*





81



82

... erro da rede com relação ao conjunto de treinamento como um todo;  
 simbologia ( $X^\mu$ ;  $y^\mu$ ); Erro quadrático de exemplar ( $Eq^\mu$ ); Erro quadrático  
 médio ( $Eqm$ )

83

*Resumo de principais resultados em lousa ...*

$\mu$  identifica um de  $M$  exemplos de treinamento

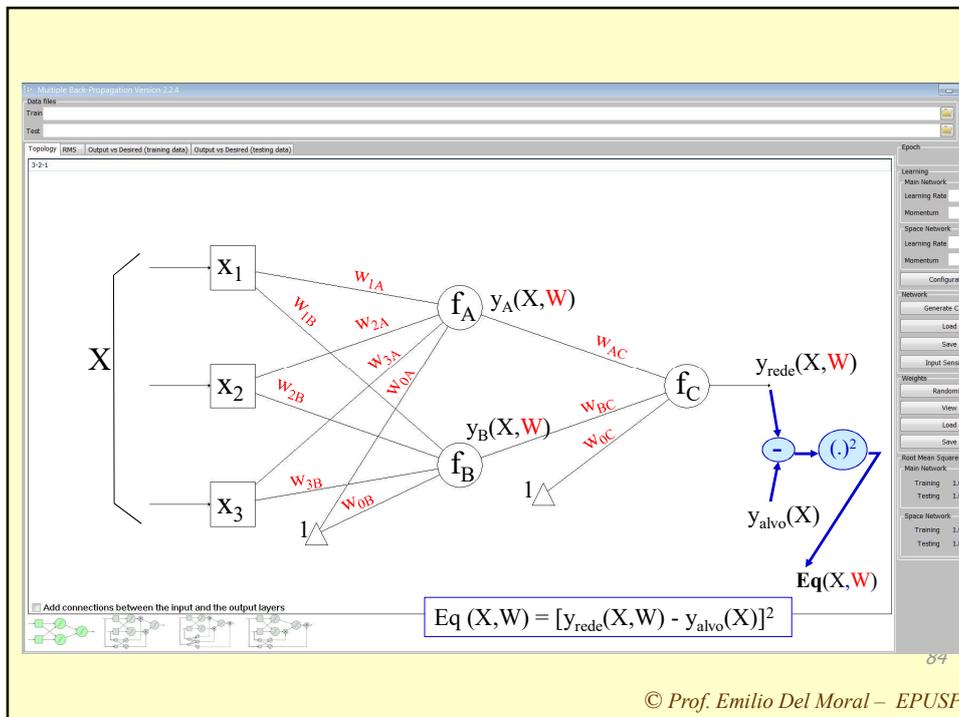
$$Eqm = \frac{1}{M} \sum_{\mu=1}^M (y_{rede}(\vec{X}^\mu) - y^\mu)^2$$

$$Eqm = \frac{1}{M} \sum_{\mu=1}^M (y_{rede}(\vec{X}^\mu, \vec{W}) - y^\mu)^2$$

© Prof. Emilio Del Moral Hernandez

83

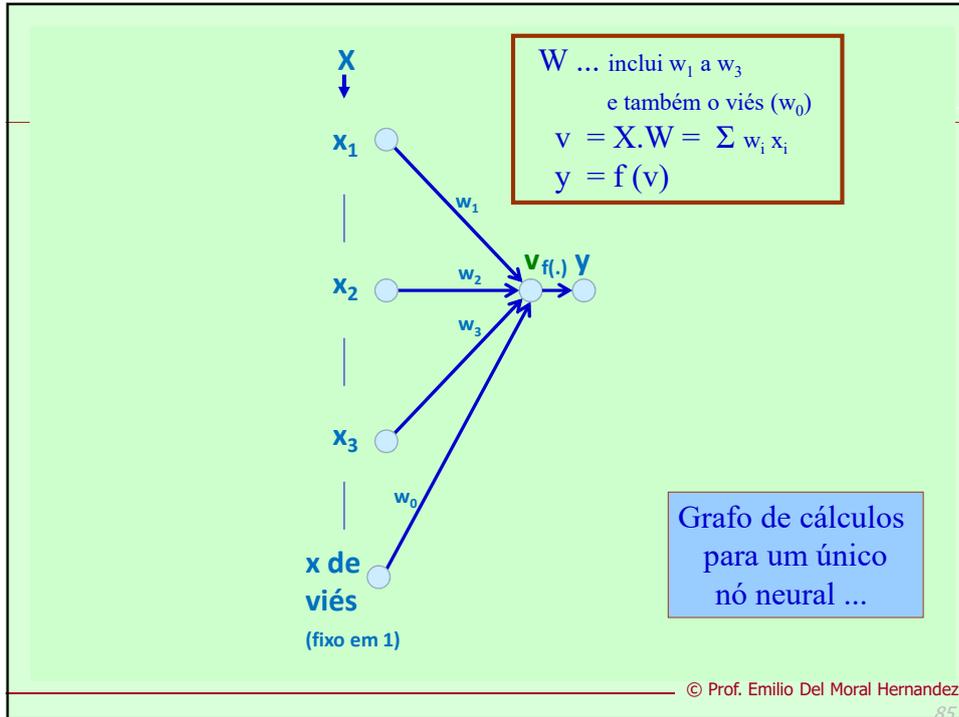
83



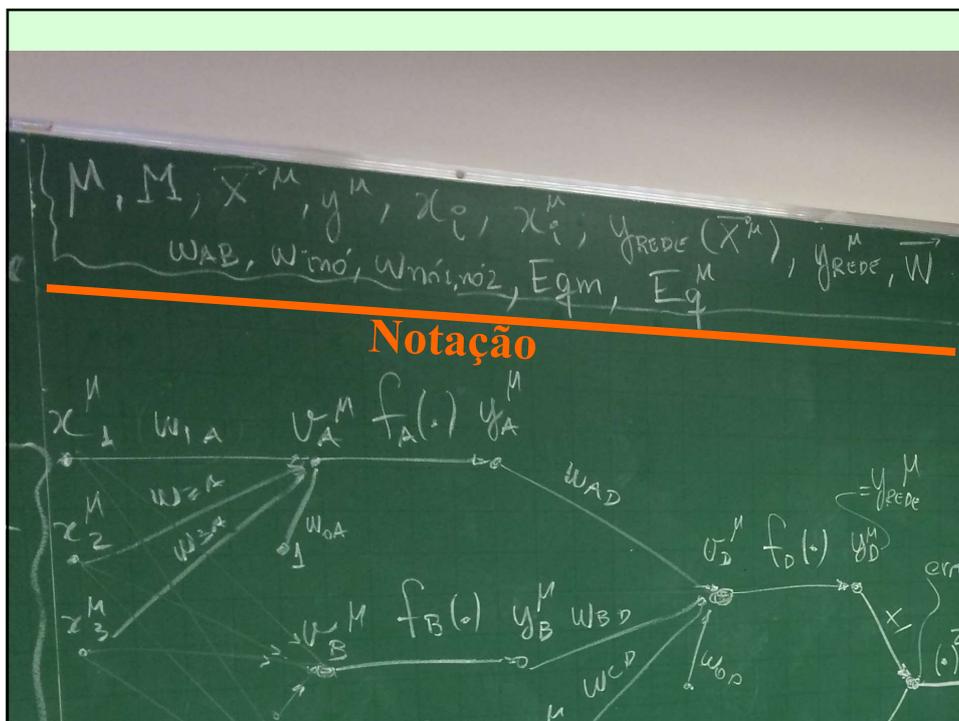
84

© Prof. Emilio Del Moral – EPUSP

84



85



86

## Exemplo de regressão multivariada para estimação contínua usando MLP

87

- O valor do  $y$  contínuo ... neste exemplo corresponde ao volume de consumo futuro num dado tipo de produto "A" a ser ofertado pela empresa a um cliente corrente já consumidor de outros produtos da empresa ("B" e "C"), volume esse previsto com base em várias medidas quantitativas que caracterizam tal indivíduo. ... Assim,  $y = \text{Consumo do Produto A} = F(x_1, x_2, x_3, x_4, x_5)$ .
- Consideremos 4 variáveis de entrada no modelo preditivo neural, ou seja, temos 5 medidas em  $X$ :
  - $x_1$ : Idade do indivíduo
  - $x_2$ : Renda mensal do indivíduo
  - $x_3$ : Volume de clicks do indivíduo no website de exibição de produtos oferecidos pela empresa
  - $x_4$ : Volume de consumo desse cliente observado para outro Produto B da mesma empresa
  - $x_5$ : Volume de consumo desse cliente Produto C da mesma empresa
- Problema: desenvolver uma MLP para regressão contínua multivariada que permita estimar esse volume de consumo futuro  $y$  com base no conhecimento dos  $X$  e numa base de dados de aprendizado com esses dados  $X$  e  $y$  para 350 já clientes de universo populacional similar ao do novo consumidor potencial.

87

87

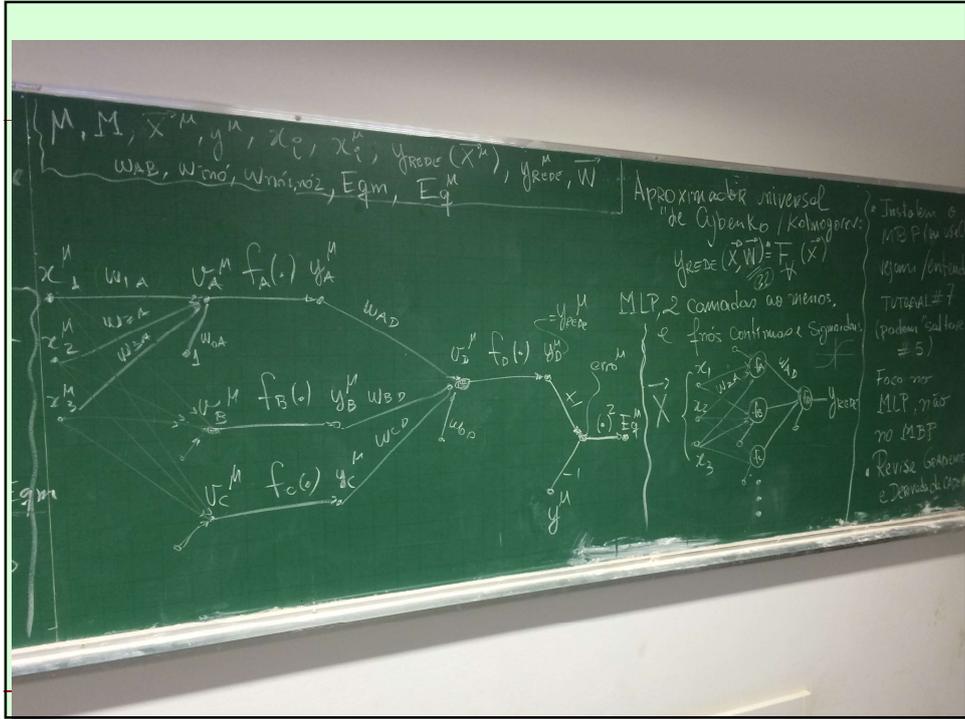
## Exemplo de dados empíricos tabulados em Excel ...

88

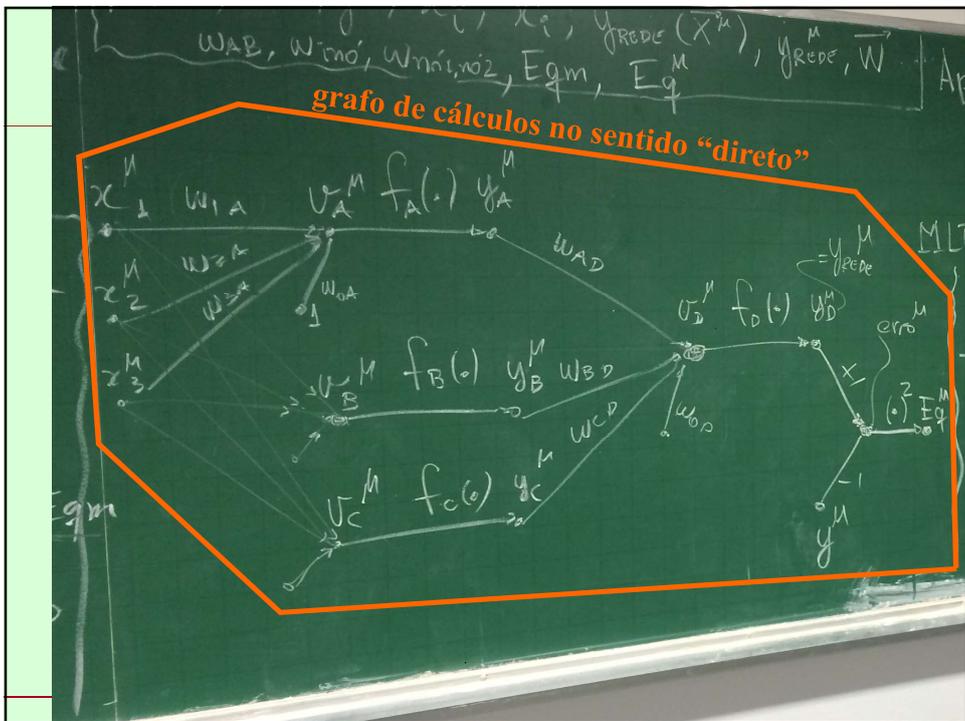
Cliente ( $\mu$ )	Idade ( $x_1$ )	Renda ( $x_2$ )	Clics ( $x_3$ )	Consumo do Produto B ( $x_4$ )	Consumo do Produto C ( $x_5$ )	Consumo do Produto A ( $y$ )
1	50	78	302	958	136	9800
2	65	128	186	985	196	8760
3	57	150	221	1093	35	520
.....	.....	.....	.....	.....	.....	.....
M-2	16	19	51	707	131	11640
M-1	30	75	7	29	78	9640
M	19	47	116	285	124	5320

88

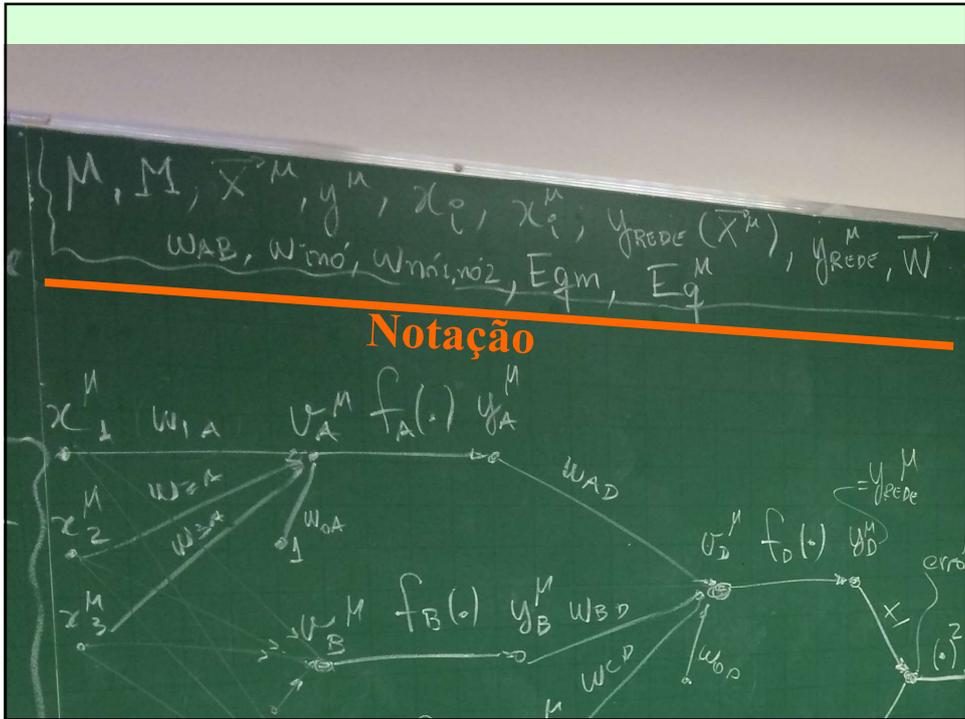
88



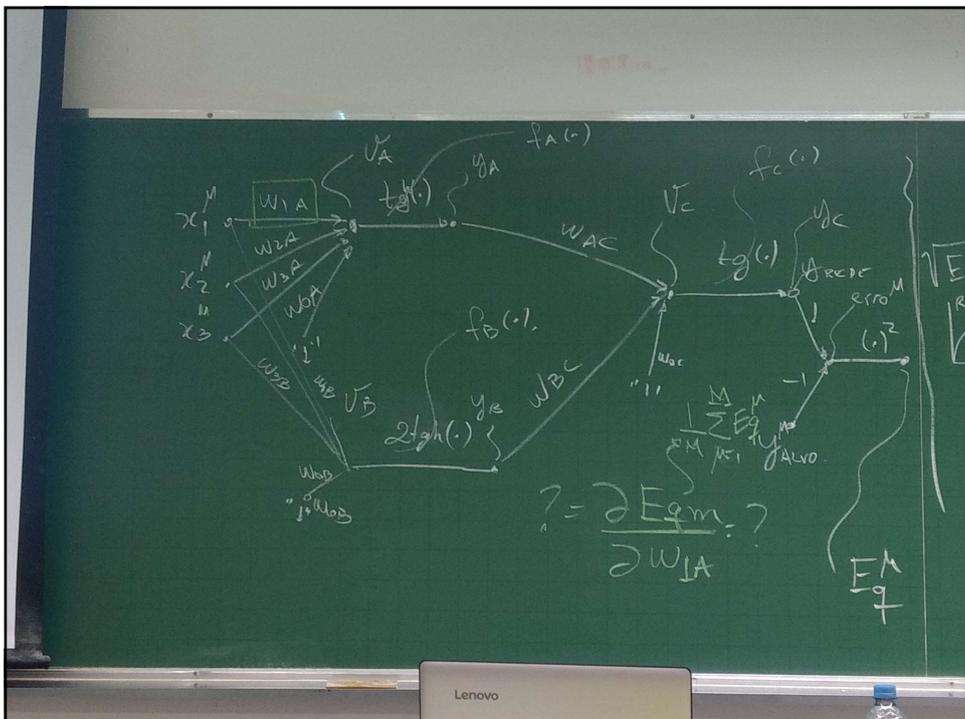
98



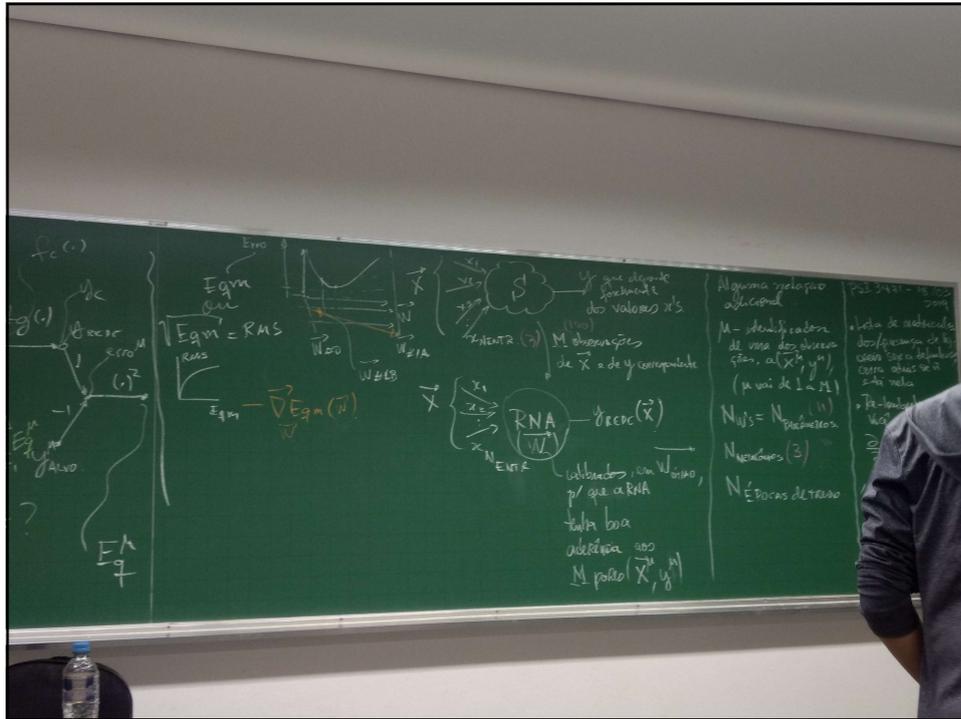
99



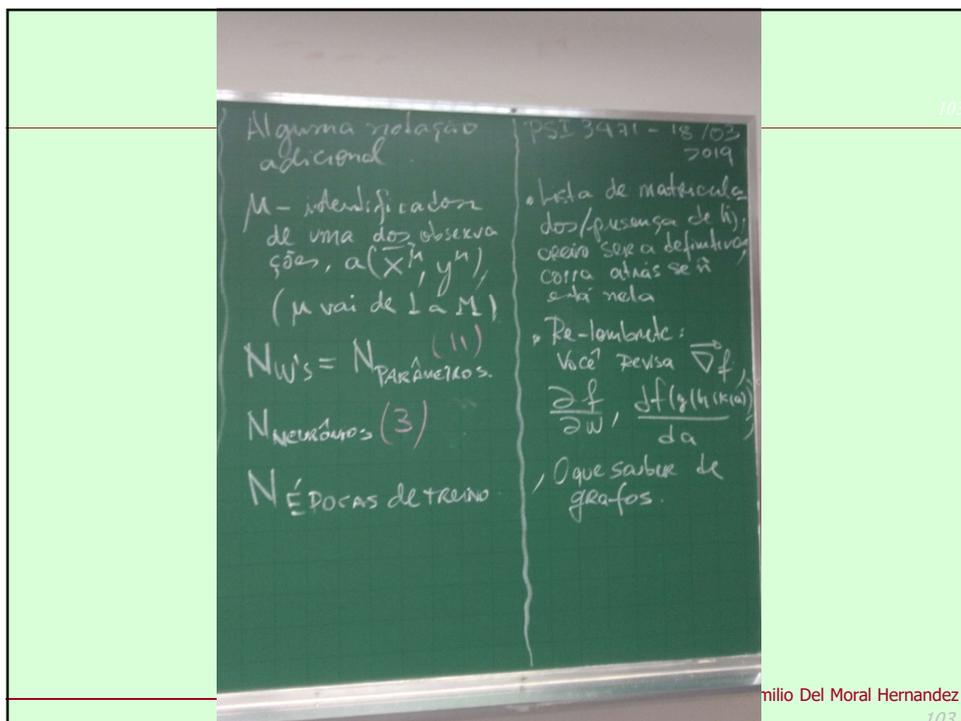
100



101



102



103

104

**Aula 7**  
(de 15)  
**do Prof. Emilio**  
**de PSI3471 de**  
**18-mar-2019**  
**se encerrou aqui**

© Prof. Emilio Del Moral Hernandez

104