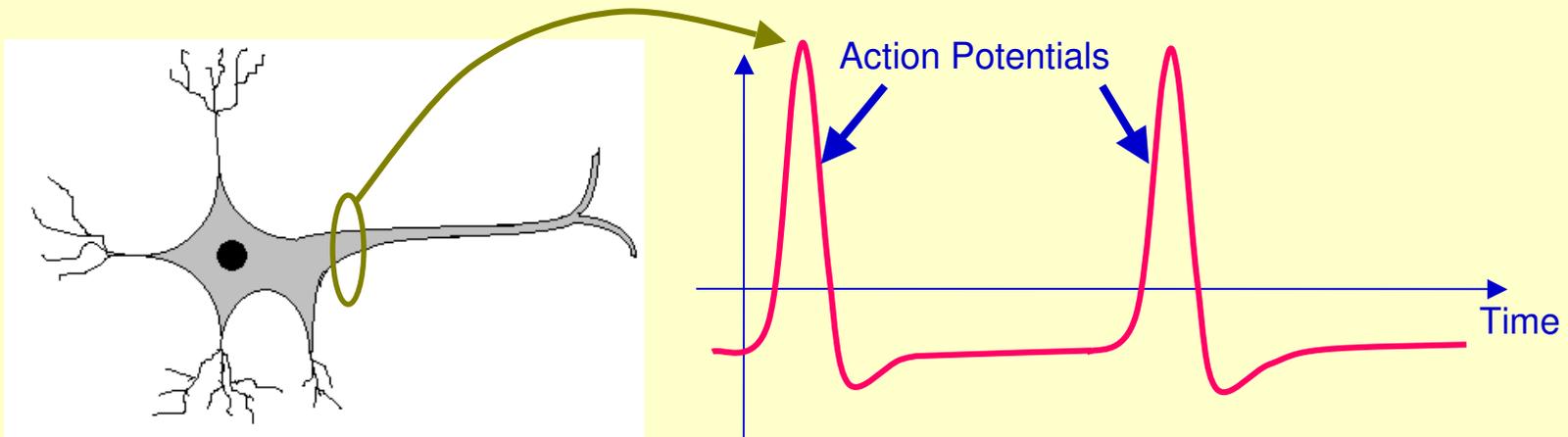


*Revisão Rápida do*  
*Fim da Aula Passada*

# The biological neuron

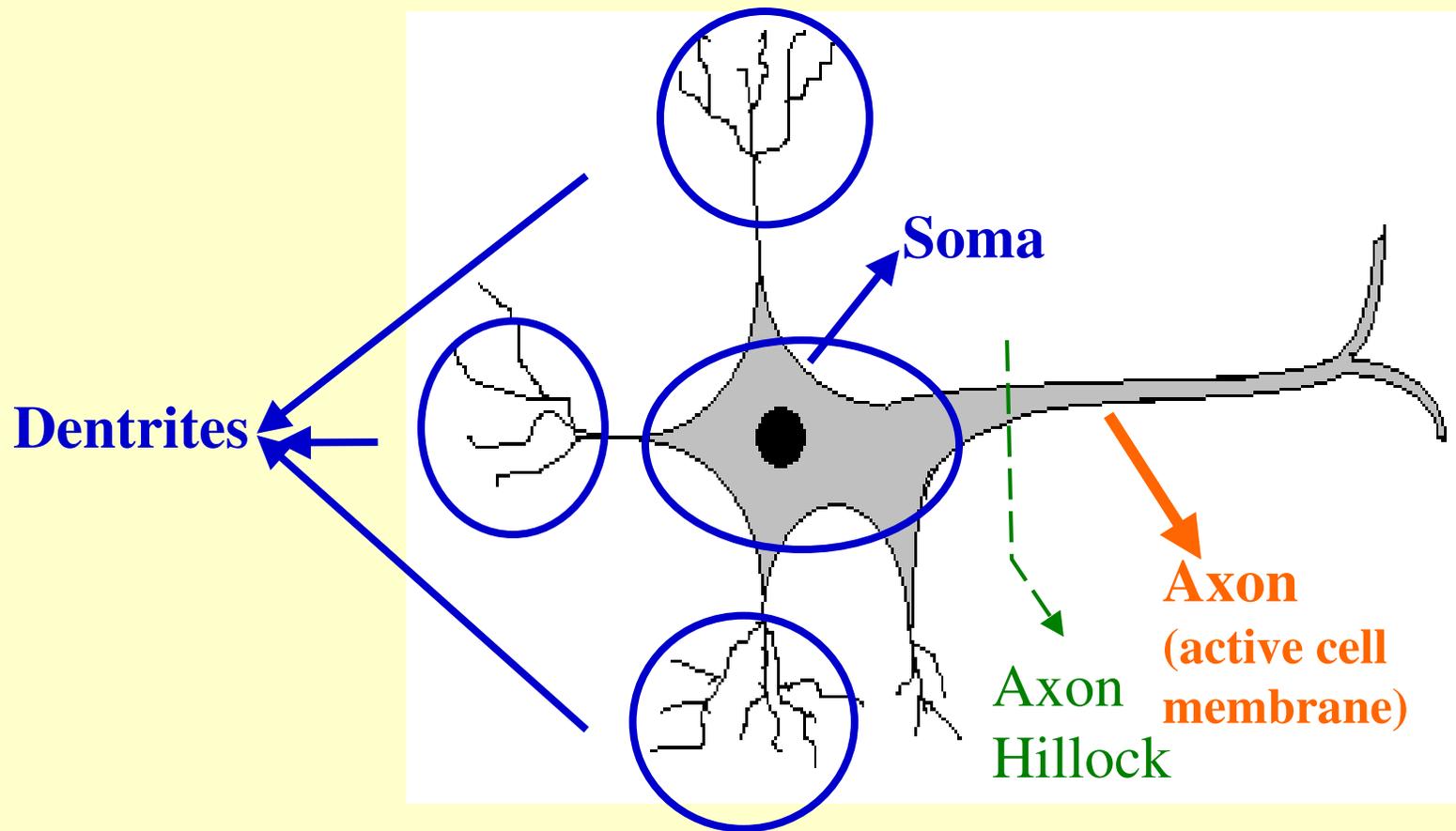
*... Information is represented in electrical pulses generated by the nervous cell, named action potentials*



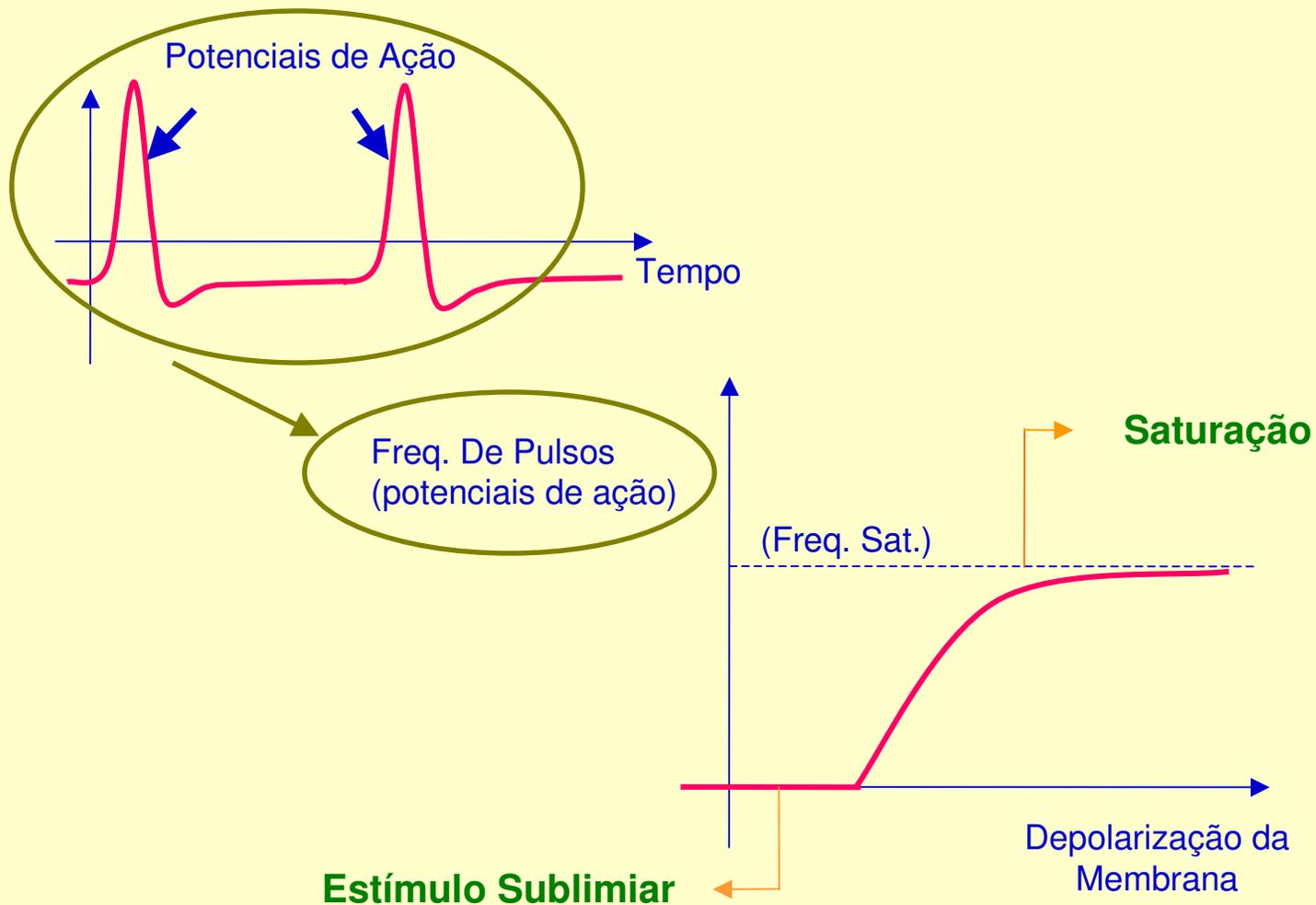
# Understanding features and role of the Action Potentials

- Membrane potential at rest: approximately – 50 mV
- Depolarization as a result of stimulation
- Once a threshold of depolarization is reached ... a pulse with characteristic dynamics is produced
- Amplitude and shape “does not code”
- Frequency of pulses “codes” (we have non repeatability of timing of spiking)

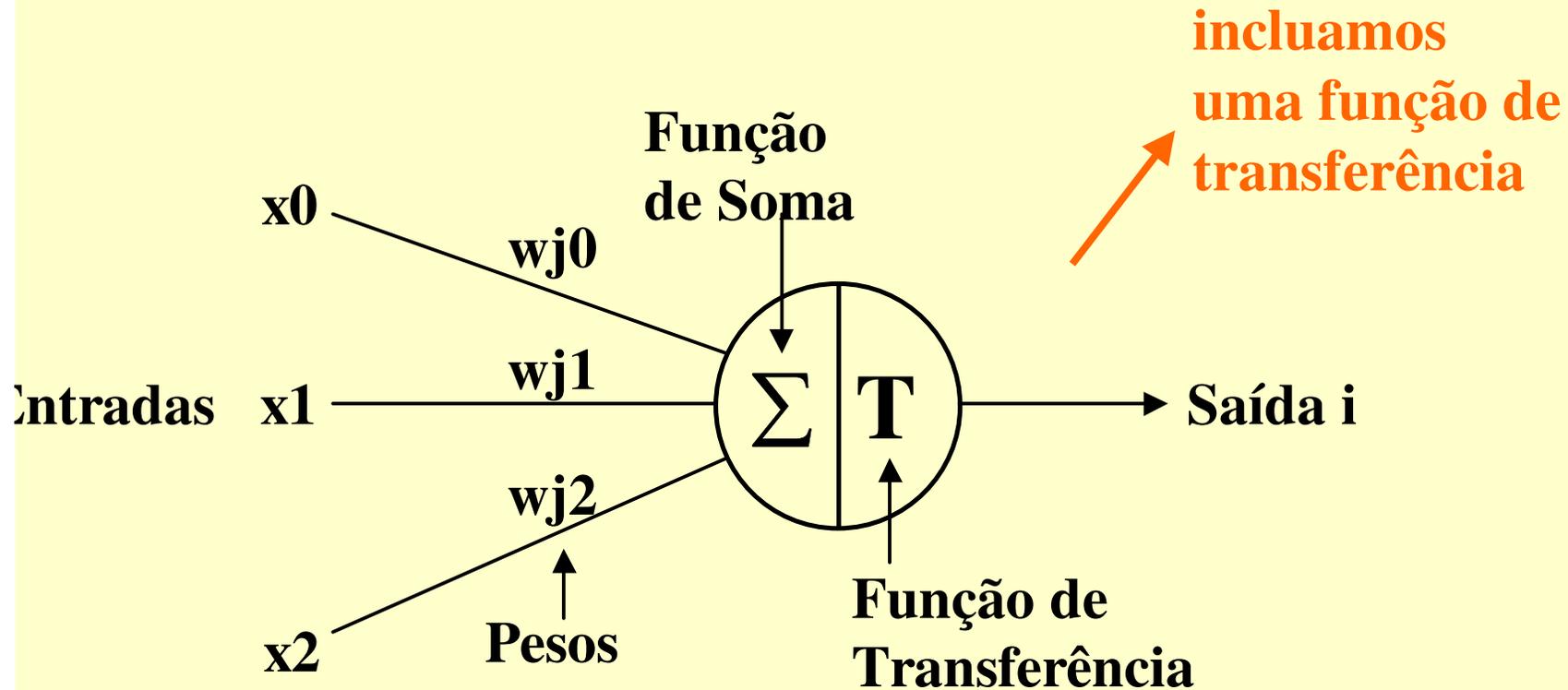
# The biological neuron



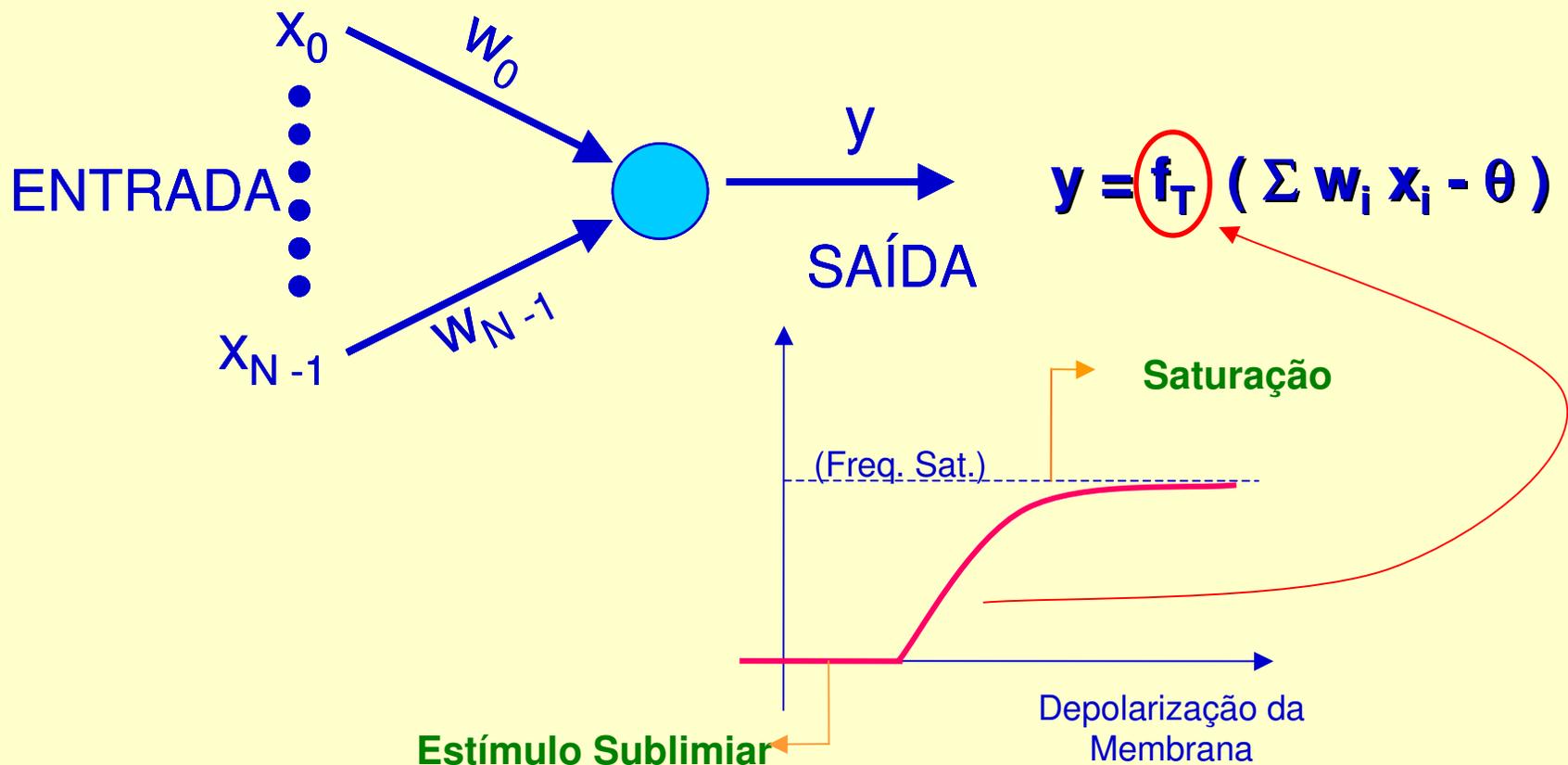
... Vendo com mais detalhe ... a relação entre estímulo e atividade na saída não é linear:



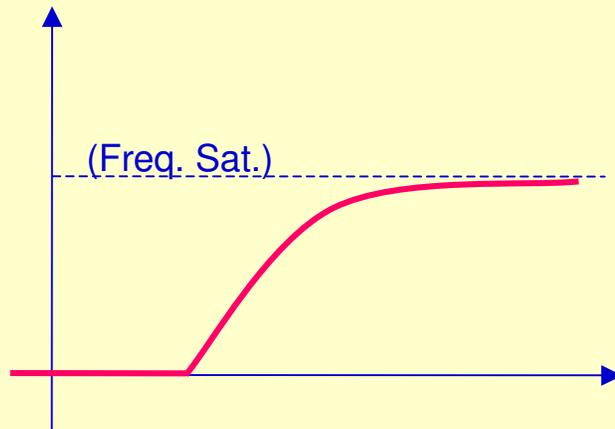
# Modelando Melhor a Relação Entrada / Saída



# Resumo ... um nó (neural) realiza a seguinte computação analógica



# The same model can now be useful for different tasks

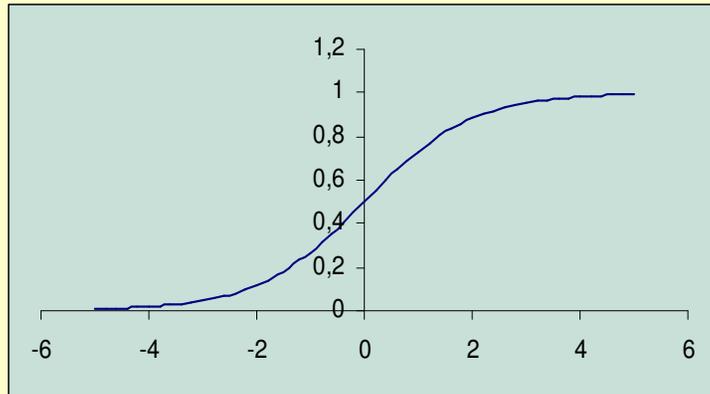


- Linear input-output mapping
- Digital output applications
- Non-linear input-output mapping

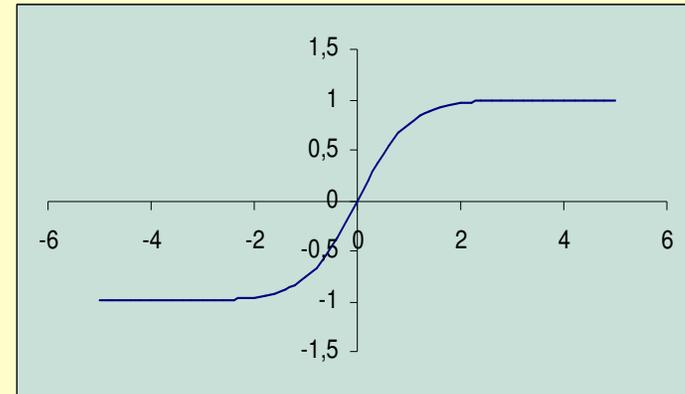
# Função de transferência não linear do tipo sigmóide ou tangente hiperbólica

- Preserva - se o fenômeno da saturação
- Preserva - se a monotonicidade na faixa dinâmica
- Temos funções matematicamente amigáveis

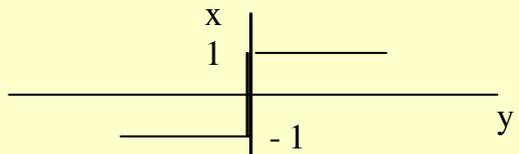
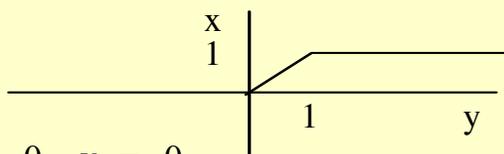
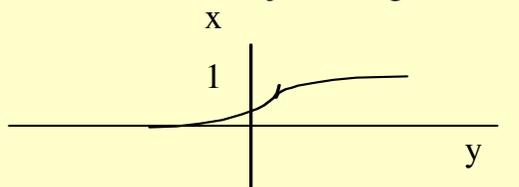
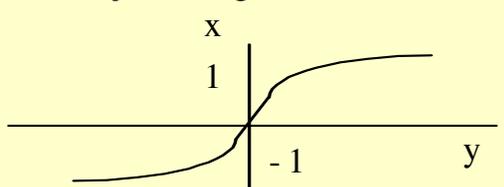
## SIGMÓIDE



## TANGENTE HIPERBÓLICA



# Outras funções de transferência não linear

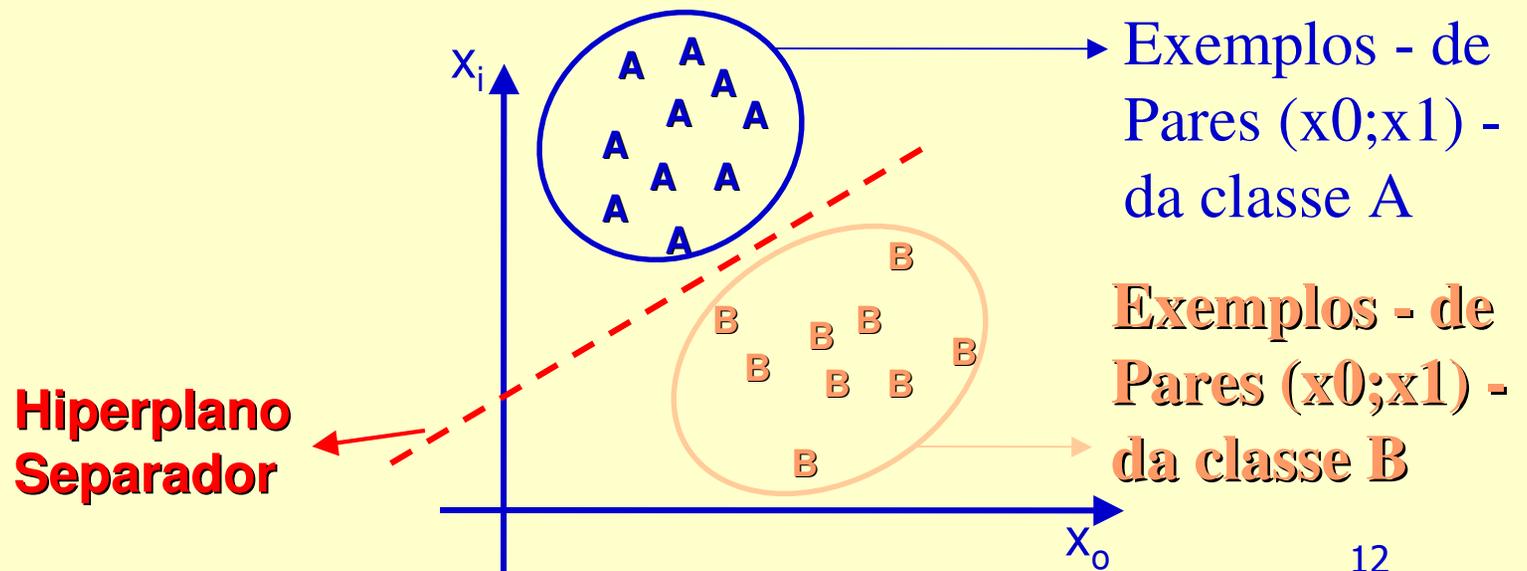
<p>Hard Limiter (limite rápido)</p>  <p> <math>s &lt; 0, y = -1</math>  <math>s &gt; 0, y = 1</math> </p>	<p>Ramping Function (função de rampa)</p>  <p> <math>s &lt; 0, y = 0</math>  <math>0 \leq s \leq 1, y = s</math>  <math>s &gt; 1, y = 1</math> </p>
<p>Sigmoid Function (função sigmóide)</p>  <p> <math>y = 1 / (1 + e^{-s})</math> </p>	<p>Sigmoid Function (função sigmóide)</p>  <p> <math>x \geq 0, y = 1 - 1 / (1 + s)</math>  <math>x &lt; 0, y = -1 + 1 / (1 - s)</math> </p>

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

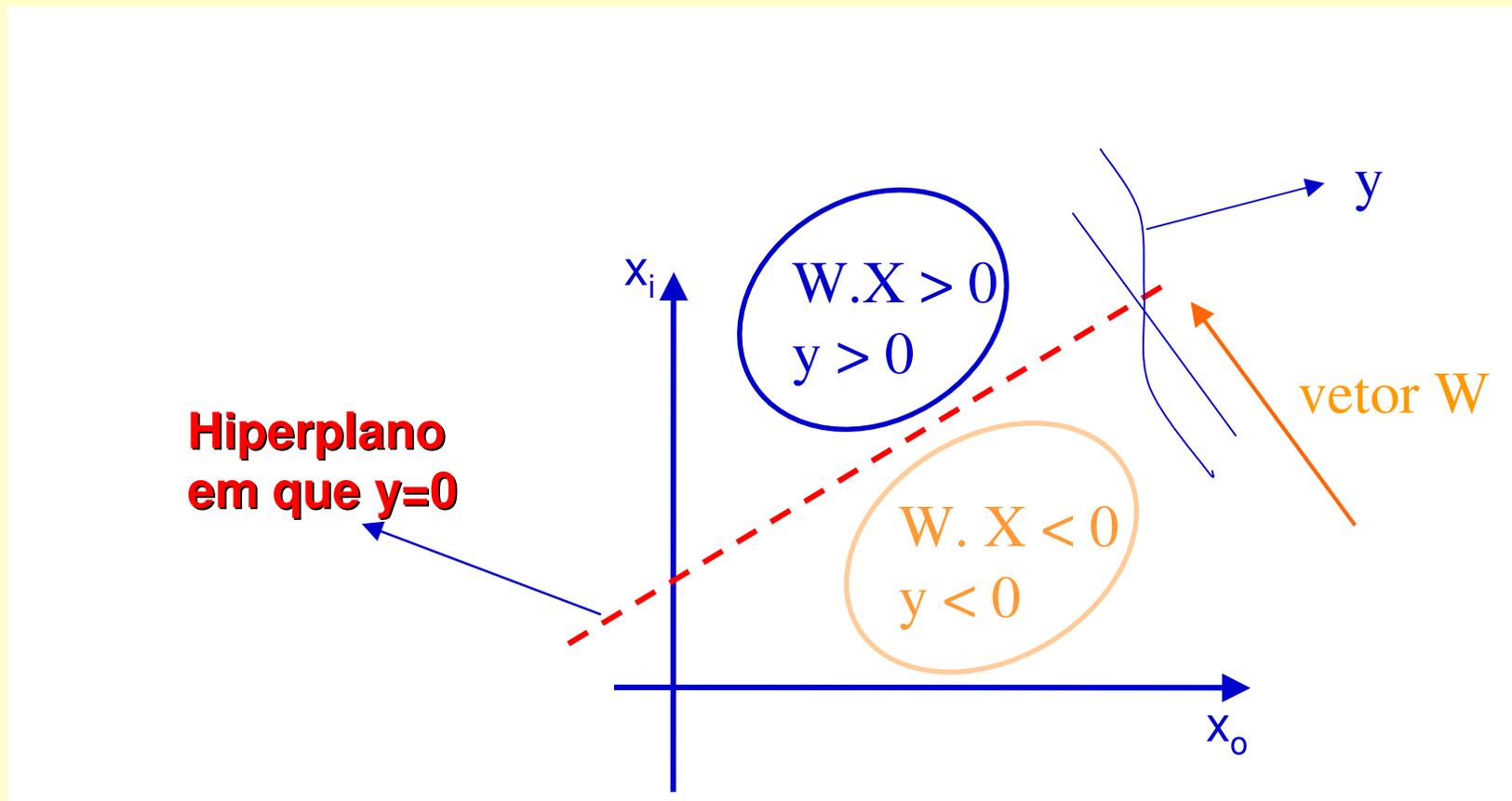
*Fim da Revisão Rápida*

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

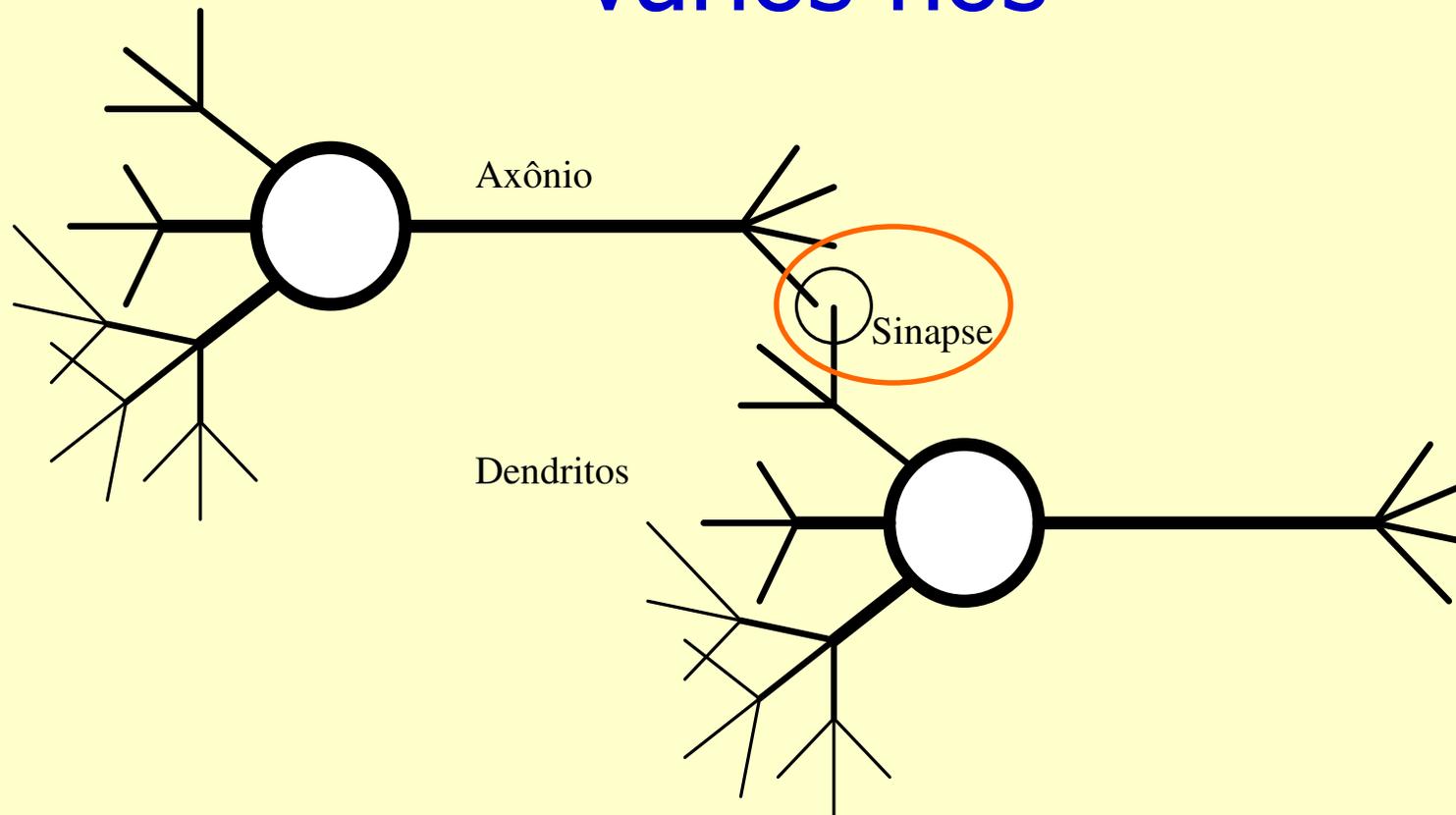
- Viabiliza a classificação de padrões com separabilidade linear
- O algoritmo de aprendizado adapta os Ws de forma a encontrar o hiperplano de separação adequado
- Aprendizado por conjunto de treinamento



# O "Perceptron Contínuo": $y = \text{tgh}(\sum w_i x_i - \theta)$



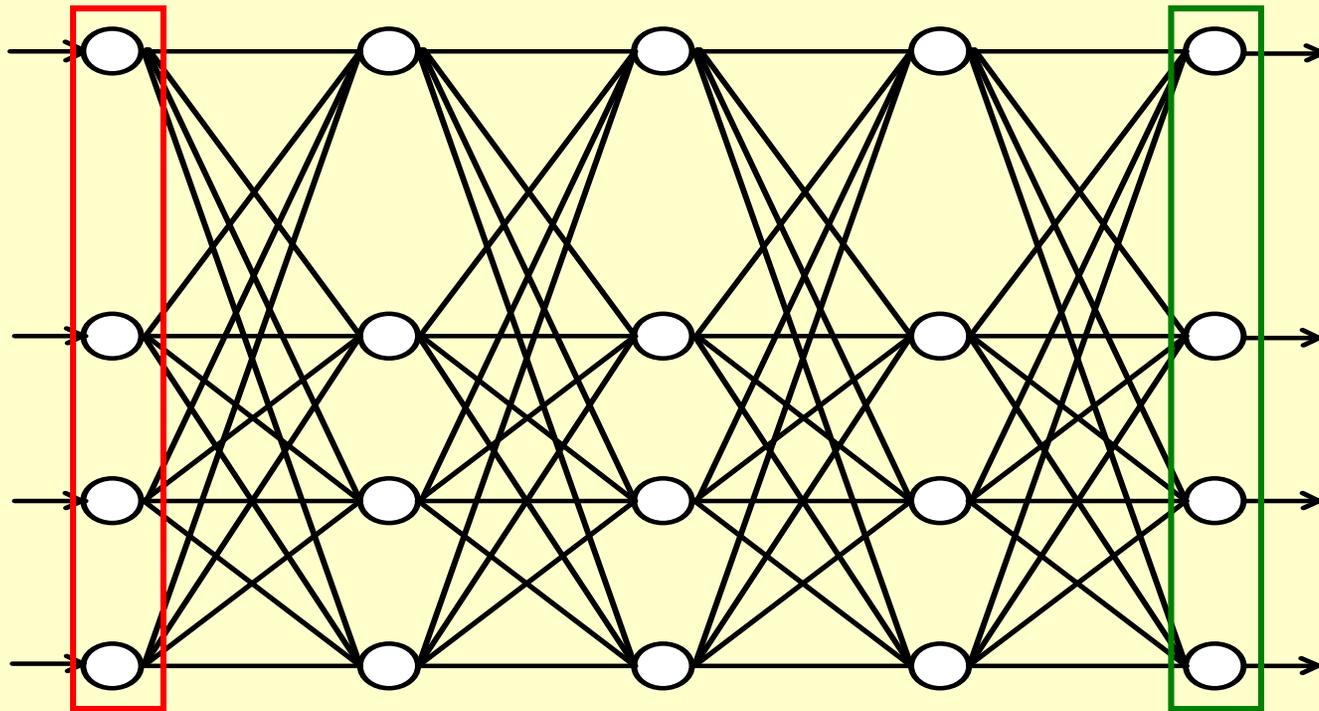
Cômputos mais complexos ... são realizados pelo encadeamento de vários nós



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

# O Multi Layer Perceptron (MLP)

- Múltiplas **entradas** / Múltiplas **saídas**
- Ambas podem ser analógicas ou digitais
- Não há mais a restrição de separabilidade linear



# Theorem of Kolmogorov & Cybenko:

- Kolmogorov:

Given for example  $F = [x_1 \cdot \sin(x_2) + \log(x_3)]/x_4 + \text{etc}$

$F(x_1, x_2, x_3, x_4, \dots) \sim a \cdot f_1(x_1) + b \cdot f_2(x_2) + f_3(x_3) \dots$   
with arbitrary precision in the approximation

- Cybenko: adapted Kolmogorov for the case where  $f_1, f_2 \dots$  are approximated by a sum of sigmoidal functions (shifted and scaled)

***Therefore, any arbitrary F CAN be implemented by an ANN with sigmoidal nodes and just 1 hidden layer!!***

# Other important applications of the MLP

In addition to generic multi dimensional function approximation / fusion ...

- Forecast of non linear time series
- Multidimensional pattern classification without linear separability

# Kolmogorov & Cybenko: impacts in Pattern Classification ...

- “Belong function” for class A =  $FA(x_1, x_2, x_3, \dots)$   
*FA is binary, with continuous argument*
- “Belong function” for class B =  $FB(x_1, x_2, x_3, \dots)$   
*FB is binary, with continuous argument*
- ... same for other classes C, D, ... etc

# **Segundo Kolmogorov & Cybenko, a estrutura MLP Permite**

***Aproximação universal ...  
Classificação universal ...***

***Basta "comprar" um MLP e o  
trabalho já está todo feito??***

***Não ... O MLP é uma  
estrutura com muitas  
constantes a determinar  
( pesos  $w_{ij}$  )***

*Temos que escolher os pesos  
adequados para atingir uma  
aproximação ou classificação  
desejada*

# "Aprendizagem" das Redes Neurais

- O aprendizado por exemplos do MLP permite que ele realize diversas funções sem a necessidade de um modelo matemático conhecido / confiável
- *"Redes Neurais possuem capacidade de aprender por exemplos"*
- *"As informações são apresentadas a rede que extrai informações a partir dos padrões apresentados"*

# O Ajuste sináptico durante o aprendizado

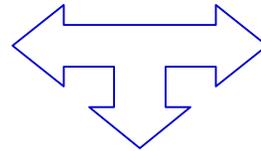
- Quando ocorre erro no reconhecimento / tratamento de uma entrada, um ajuste sináptico é necessário.
- O ajuste sináptico representa o aprendizado em do fato apresentado / O conhecimento dos neurônios reside nos pesos sinápticos.
- O ajuste sináptico procura corrigir os pesos de modo que se produza a saída desejada diante da respectiva entrada.
- Esse cálculo visa somar ao peso atual, um valor que corresponda a quantidade de erro gerada pela rede, e desta forma corrigir o valor do peso.

# Aprendizado Supervisionado

- As informações são apresentadas à rede sob forma de padrões de entrada e os resultados desejados são conhecidos previamente.
- O “supervisor” verifica a saída da rede e a compara com a saída esperada
- Minimização da diferença
- Os algoritmos mais conhecidos são :
  - Regra Delta
  - Backpropagation

# Exemplo de ajuste sináptico

Entrada X



Saída Y

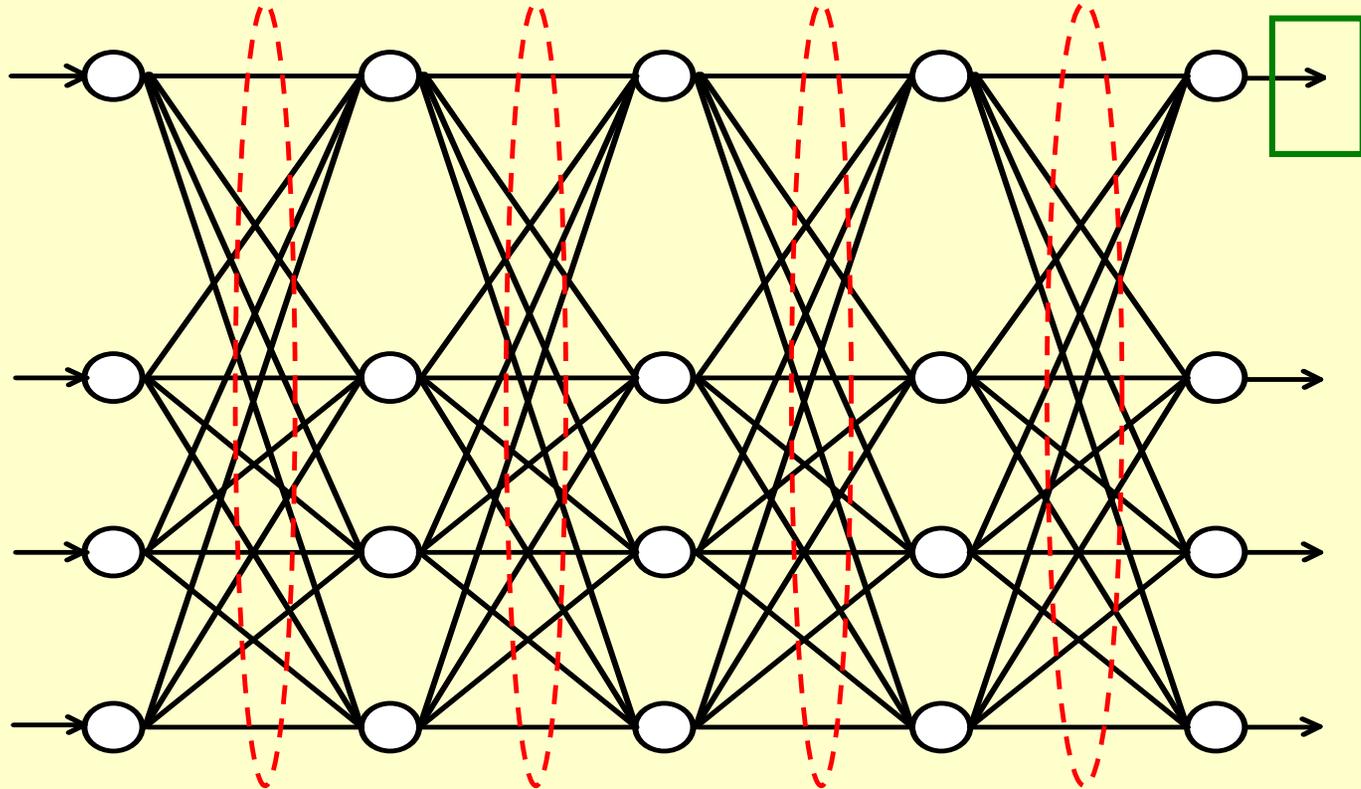
Resultado Esperado  
(= Y obtido?)

Algoritmo:

- .Aplicar entrada X
- .Comparar saída Y com a saída desejada
- .Se saída estiver errada então
  - .Calcular ajuste em pesos
- .Aplicar nova entrada e repetir

# The learning in the MLP

- The **output error** is evaluated for each sample
- The **weights  $w_{ij}$**  are changed to minimize the error – gradient descent on the error function



***Esta e as aulas seguintes***

***Aprendizado para o MLP ...***

***Error Back Propagation***

***Algorithm***

***(Propagação reversa de Erro)***

*Tema da  
Lição de Casa:  
MLP/Hopfield/SOM*

# Diversity of Neural Architectures

- Is “Perceptrons & Multi Layer Perceptrons (MPLs)” a synonym of Artificial Neural Networks?

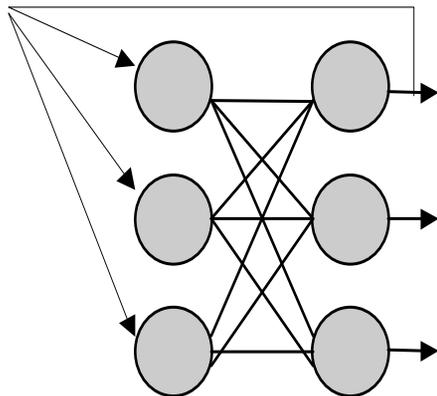
*No!!! There are many other very distinct topologies for arrangement of nodes*

- Is Error Back Propagation a synonym of learning in artificial neural networks?

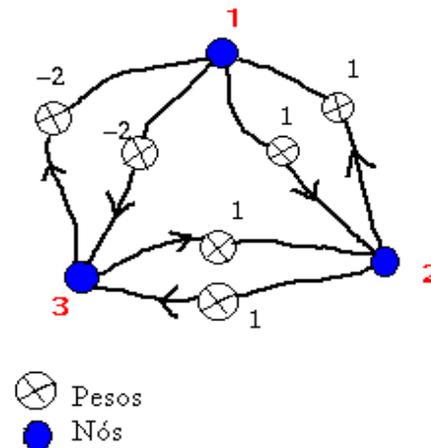
*No!!! The weights in other architectures (non MLPs) are defined through very different strategies which are nor guided by gradient descent*

# Neural Networks with Feedback

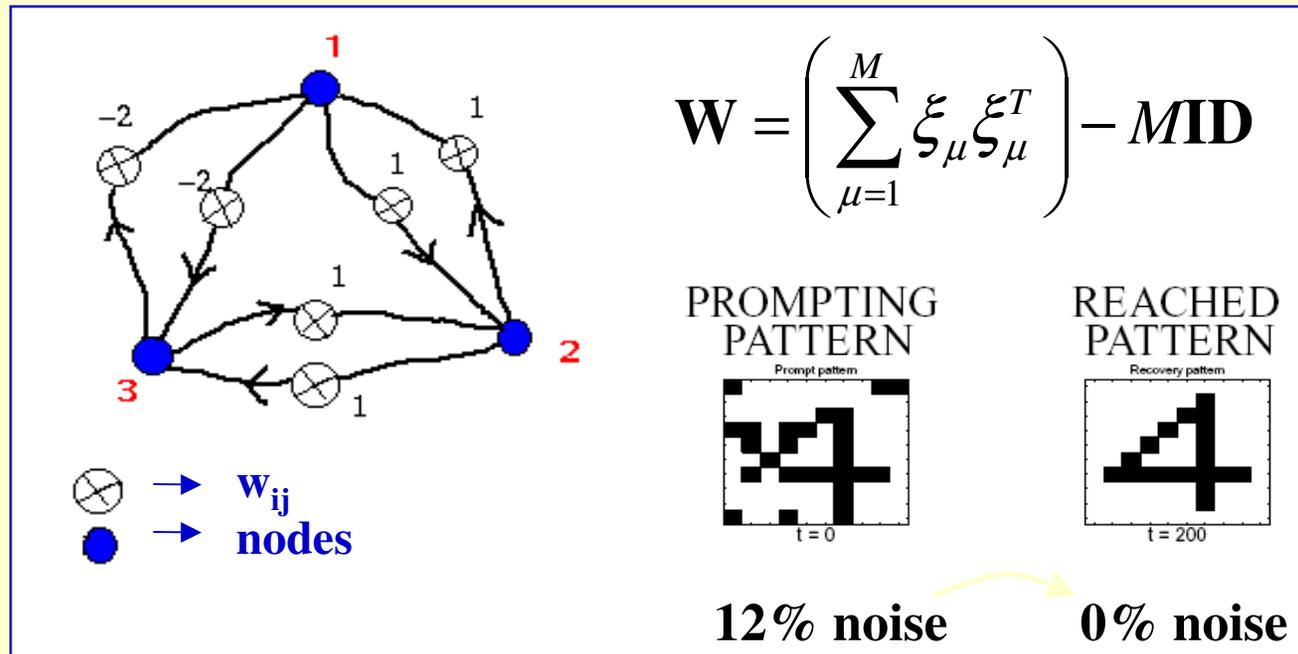
- General concept: the present state of each node affects the future states
- We have thus a dynamical system



- A particular architecture: Hopfield
- Full connection / symmetric weights

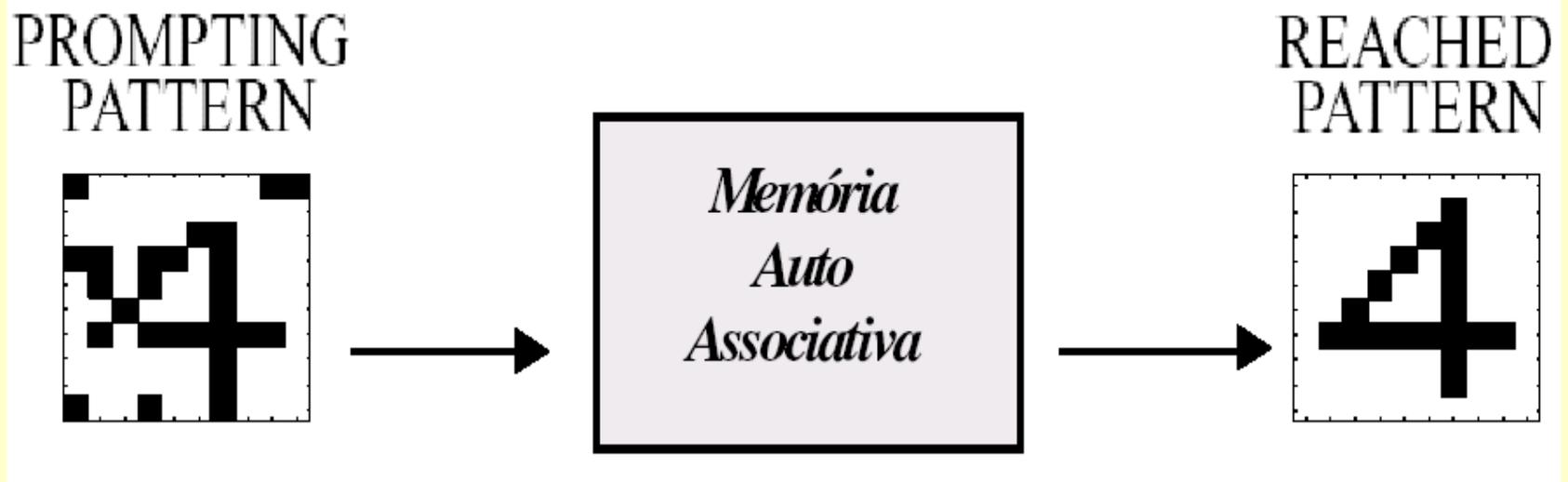


# Associative Hopfield memories



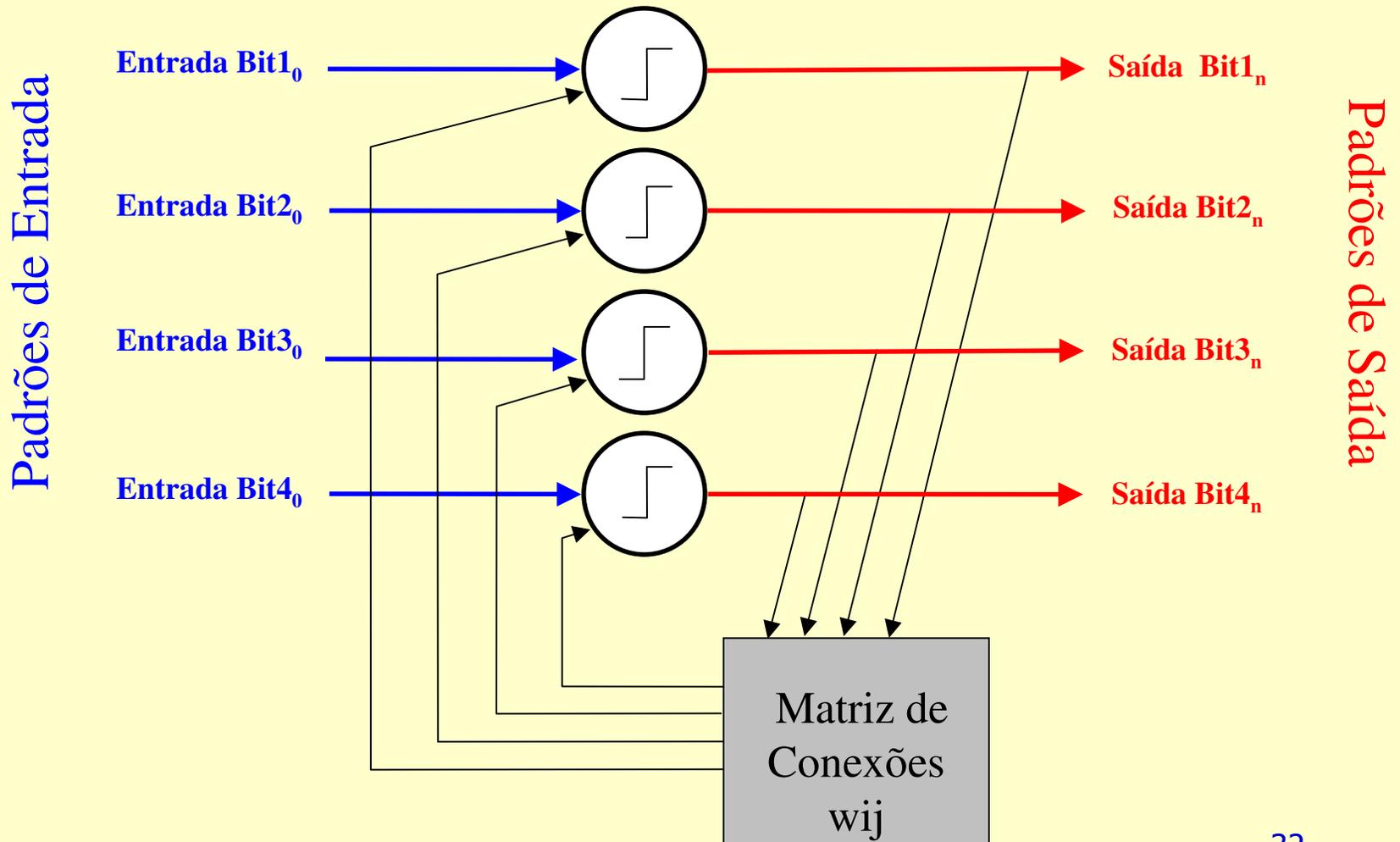
Prototypical applications of associative networks on binary strings: recovery of binary patterns from distorted or partial versions of the stored memories

# Example of auto-association in the recovery of corrupted images

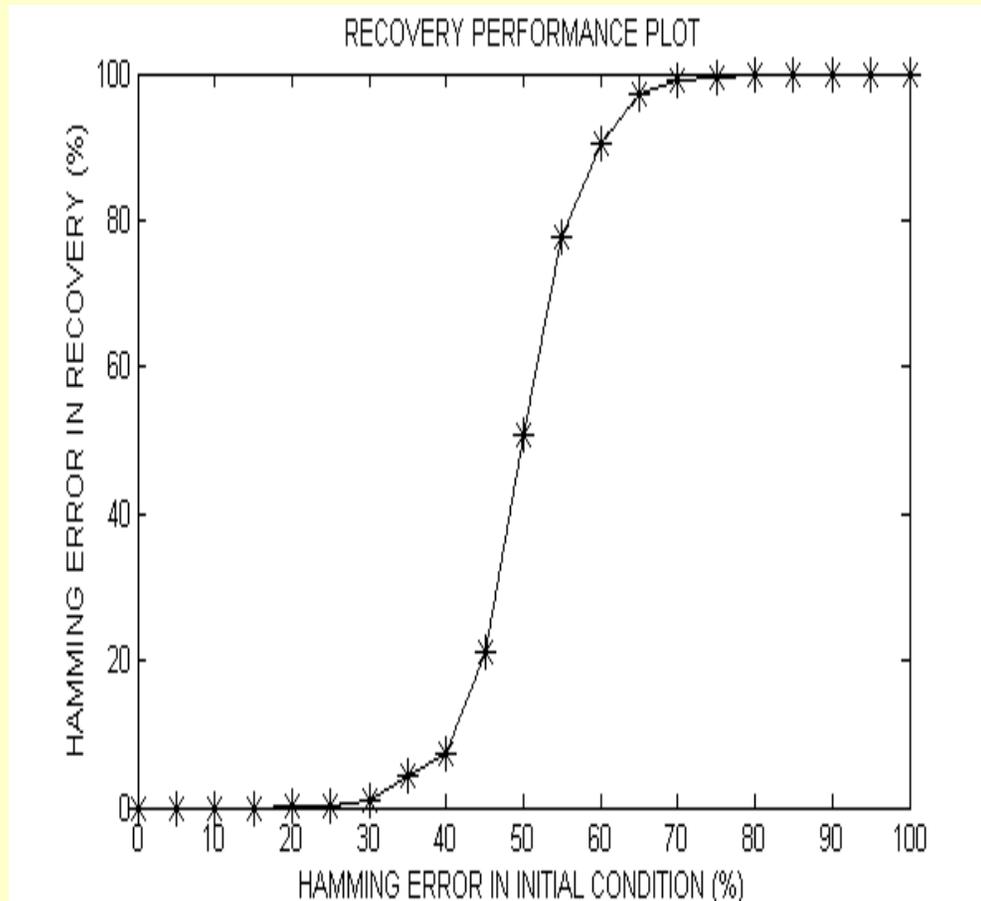


[DelMoral2004c]

# Structure of Hopfield associative architectures



# Pattern recovery in auto associative Hopfield NNs



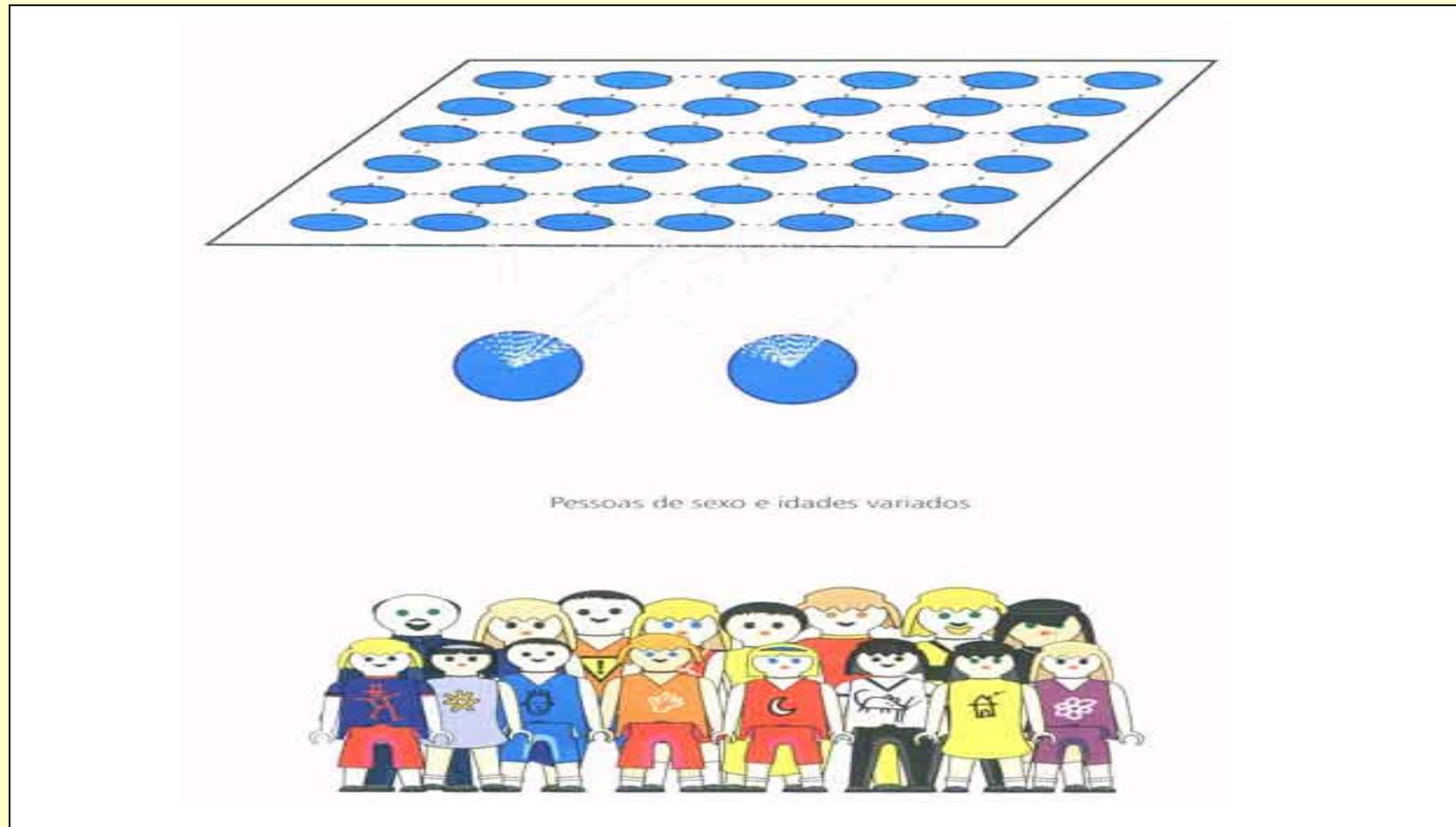
# Data Mining and Clustering of Multidimensional data

Clodis Boscarioli. PhD. Student.

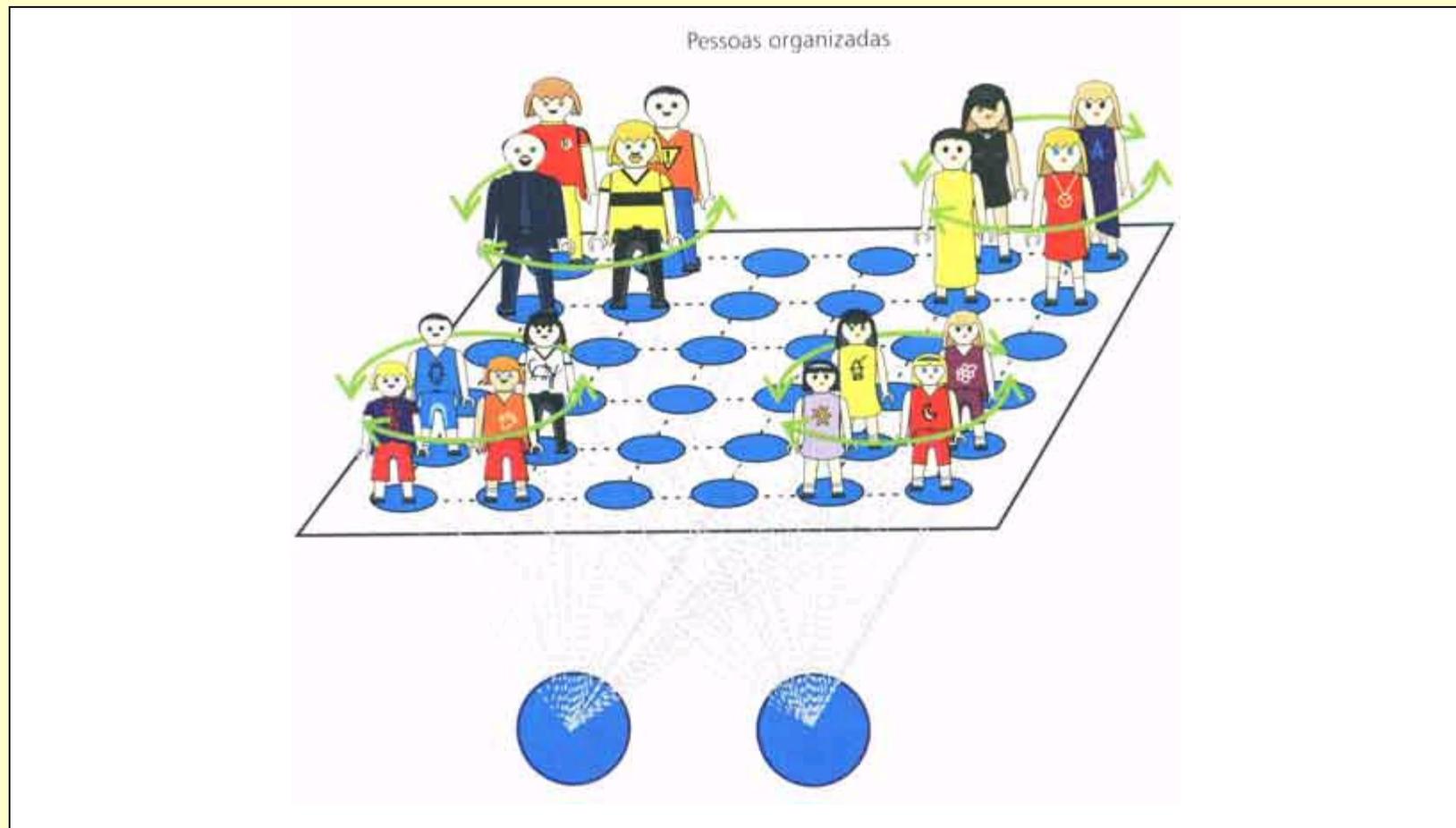
Research area: The application of clustering techniques to Data Mining. More specifically working in the development and evaluation of Self-Organizing Maps algorithms for this task.

Interests: Database systems, data mining and non-supervised neural networks.

# Kohonen Architecture (SOM) for datamining



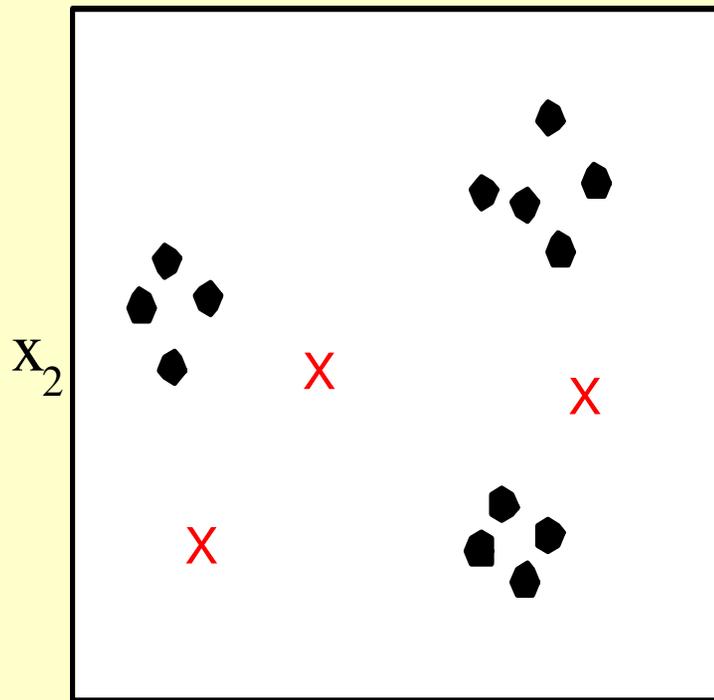
# Samples are grouped around prototypes in a 2-d grid



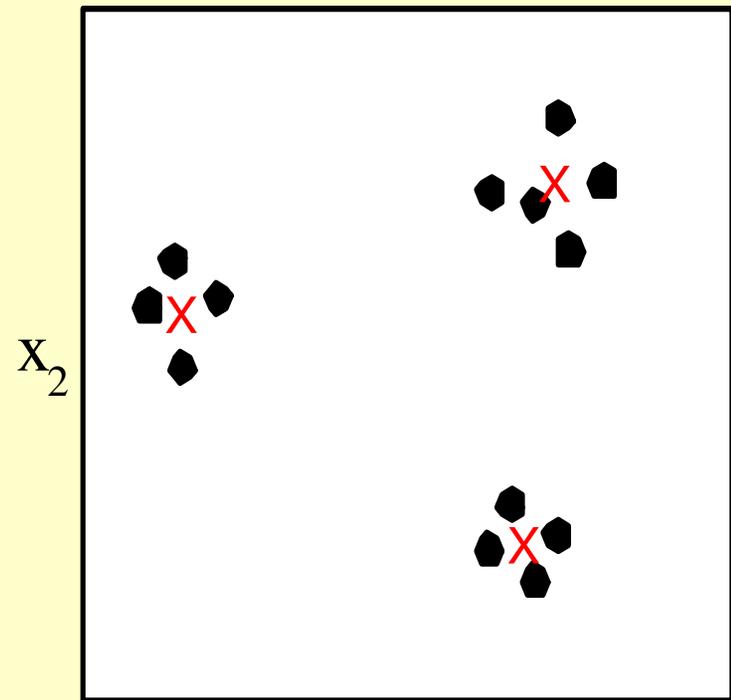
# Applications of the Self Organizing Maps (SOM) of Kohonen

- Partition of the universe of multidimensional data in sub-groups of similar elements
- Data mining (clustering + visualization)
- Multidimensional data pre processing aiming facilitated classification, within more specific sub universes

# Adaptation of the weights of prototypes in the Kohonen grid

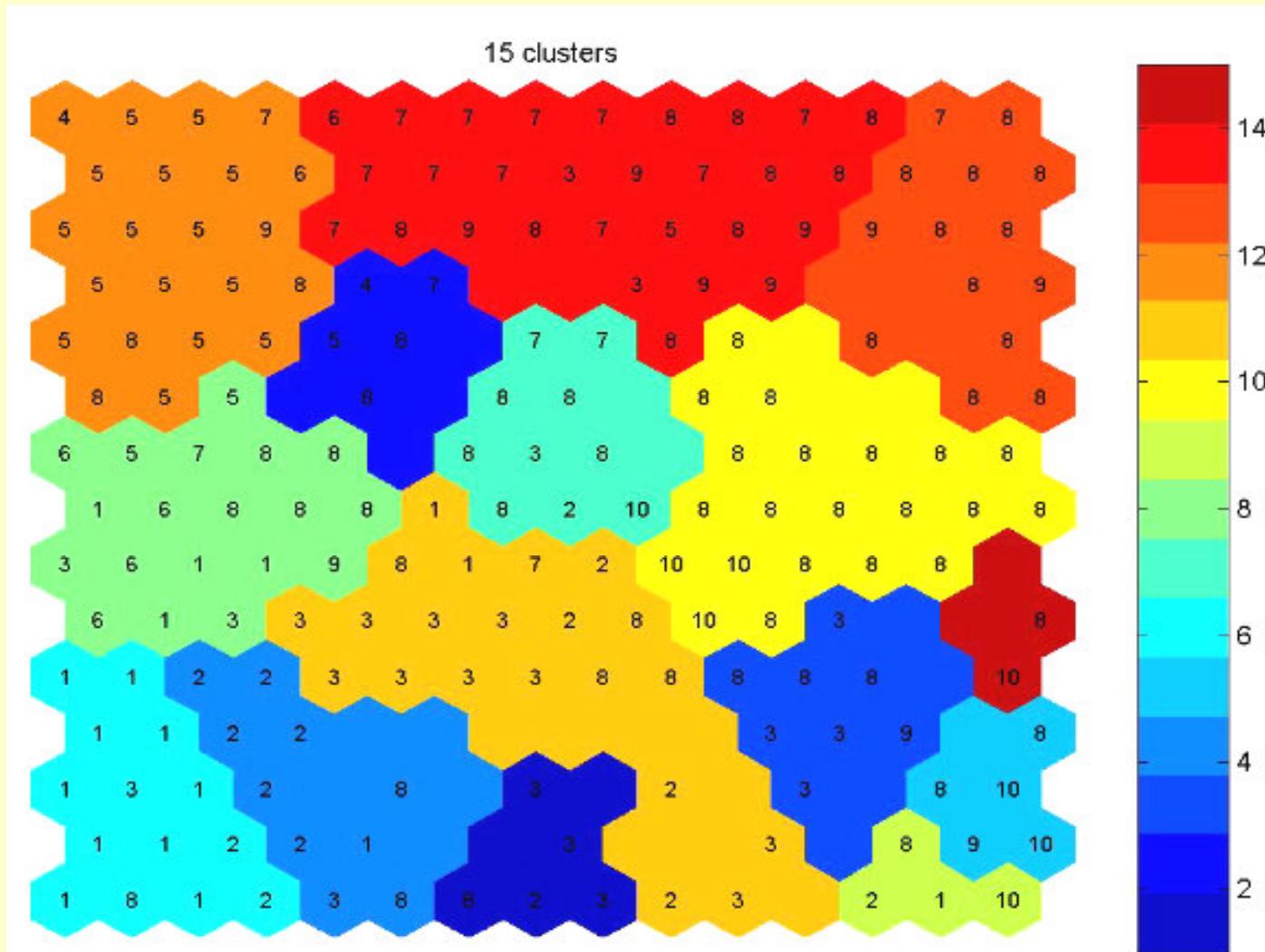


$x_1$   
Início

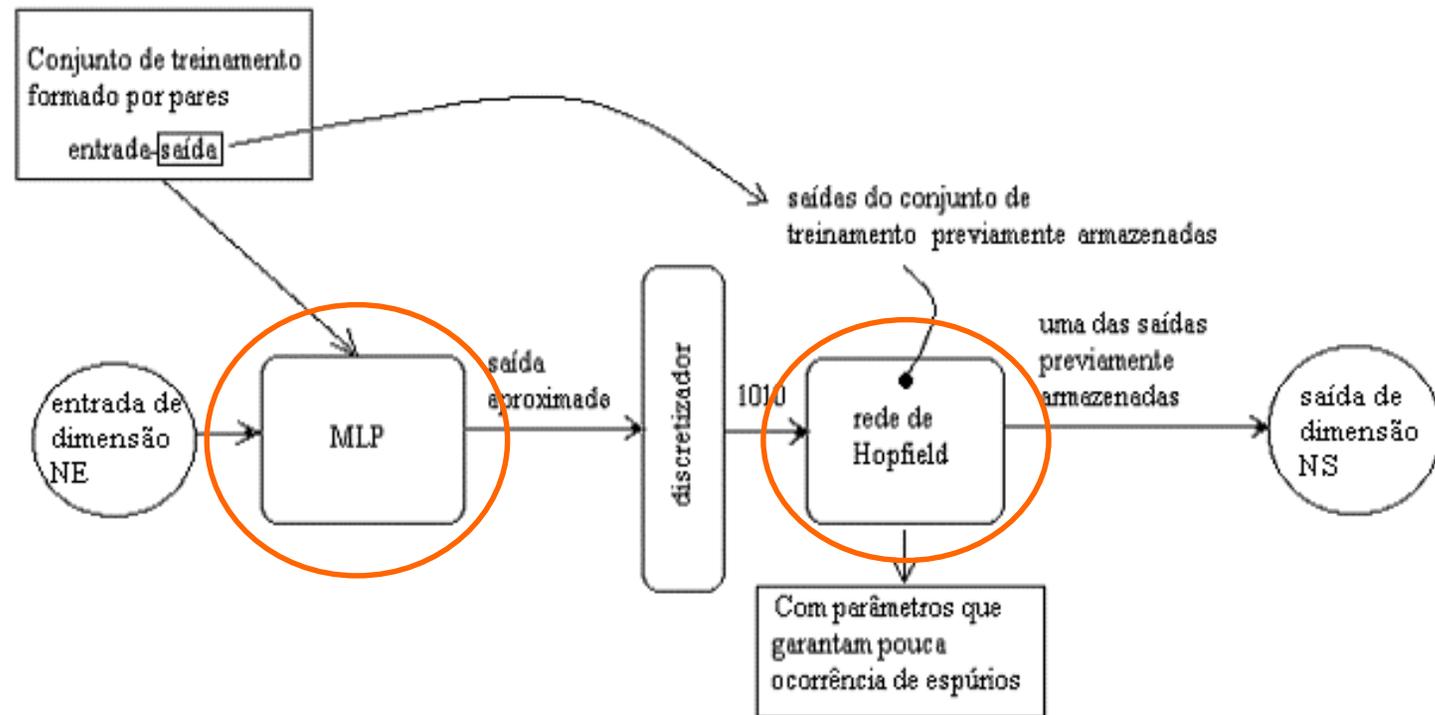


$x_1$   
Fim

# Example of clustering / visualization of clustering in a Kohonen SOM



# Hybrid Neural Networks MLP+H



Work with Clayton Silva Oliveira

*Atenção!!!  
Respeito à  
Autoria  
rima com  
Academia*

# ***Copy and Paste não é criação ... estamos no ramo de criação!***

evidenciar a distinção com as redes neurais biológicas. A ênfase do seminário será em redes neurais artificiais, mas serão abordadas de forma breve algumas outras ferramentas que também se enquadram no âmbito geral de “inteligência computacional”, a saber: inteligência artificial simbólica, lógica difusa, e apoio à decisão com ferramentas estatísticas.

Falando de uma forma bastante abrangente, o objetivo do campo “inteligência computacional” é o desenvolvimento de técnicas que permitam incorporar no ambiente computacional, mesmo que de forma limitada, algumas das funcionalidades dos sistemas biológicos (humanos ou não) no que se refere a seus aspectos de inteligência. (tipicamente, a referência é a inteligência humana, mas de uma forma geral podemos considerar a inteligência biológica em suas diversas formas e níveis) É importante notar que quando se fala em inteligência computacional como imitação ou incorporação no

---

<sup>1</sup> *Este material, deve sempre ser citado quando usado em preparação de trabalhos ou artigos de sua autoria. O mesmo procedimento se aplica a qualquer outro material que não seja de sua autoria, inclusive às apostilas da EPUSP. Adicionalmente, reproduções sem modificação de partes de textos que não sejam originalmente seus, devem vir entre aspas. Tal respeito à autoria de trabalho intelectual deve ocorrer igualmente com textos disponíveis através da Internet.*