



UNIVERSIDADE DE SÃO PAULO  
INSTITUTO DE CIÊNCIAS MATEMÁTICAS E DE COMPUTAÇÃO  
Departamento de Ciências de Computação

# SCC-5809

## Capítulo 2 - Topologia e Representação

João Luís Garcia Rosa<sup>1</sup>

<sup>1</sup>SCC-ICMC-USP - joaoluis@icmc.usp.br

2012

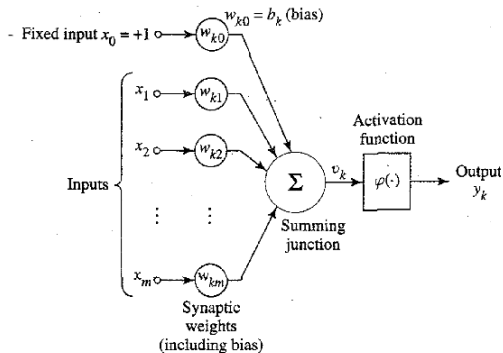
# Sumário

- 1 Modelos
  - Modelo de um neurônio
  - McCulloch-Pitts
  - Funções de ativação
- 2 Arquiteturas
  - Colunas e camadas
  - Redes *Feedforward*
  - Redes Recorrentes
- 3 Representação do Conhecimento
  - Representação Conexionista
  - Regras para Representação

# Sumário

- 1 Modelos
  - Modelo de um neurônio
    - McCulloch-Pitts
    - Funções de ativação
- 2 Arquiteturas
  - Colunas e camadas
  - Redes *Feedforward*
  - Redes Recorrentes
- 3 Representação do Conhecimento
  - Representação Conexionista
  - Regras para Representação

# Modelo Não-linear de um neurônio [2]



# Fórmulas [2]

$$u_k = \sum_{j=1}^m w_{kj} x_j \quad (1)$$

$$y_k = \varphi(u_k + b_k) \quad (2)$$

$$v_k = u_k + b_k \quad (3)$$

$$v_k = \sum_{j=0}^m w_{kj} x_j \quad (4)$$

$$y_k = \varphi(v_k) \quad (5)$$

$$x_0 = +1 \quad (6)$$

$$w_{k0} = b_k \quad (7)$$

# Sumário

## 1 Modelos

- Modelo de um neurônio
- **McCulloch-Pitts**
- Funções de ativação

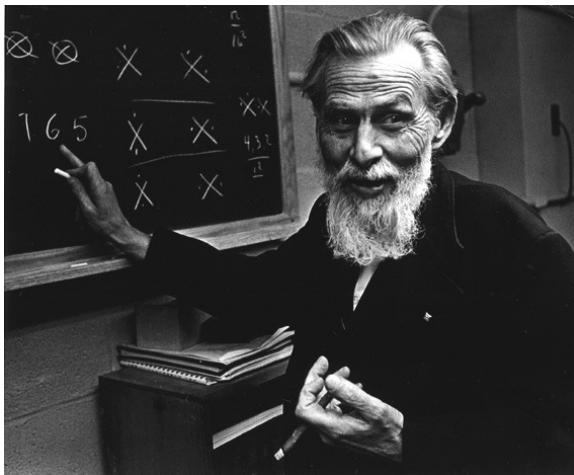
## 2 Arquiteturas

- Colunas e camadas
- Redes *Feedforward*
- Redes Recorrentes

## 3 Representação do Conhecimento

- Representação Conexionista
- Regras para Representação

# Warren McCulloch

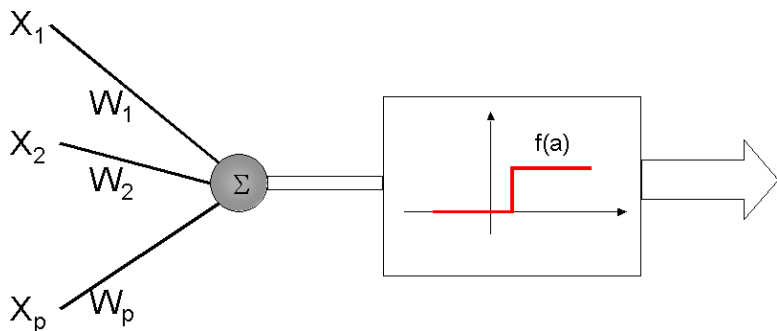


# Neurônio de McCulloch-Pitts (1943) [4]

- 1 A atividade do neurônio é um processo “tudo-ou-nada”.
- 2 Um certo número fixo de sinapses deve ser excitado dentro do período de adição latente para excitar um neurônio, e este número é independente da atividade anterior e da posição do neurônio.
- 3 O único atraso significativo no sistema nervoso é o atraso sináptico.
- 4 A atividade de qualquer sinapse inibitória previne a excitação do neurônio.
- 5 A estrutura da rede não muda com o tempo.



# Neurônio de McCulloch-Pitts (1943)



# Neurônio de McCulloch-Pitts (1943)

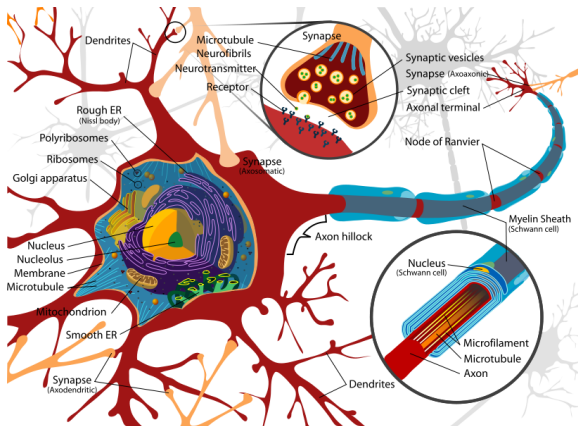
- $x_i$ : entradas (binário);
- $w_i$ : pesos sinápticos (real, porque as sinapses podem inibir (-) ou excitar (+) e têm intensidades diferentes);
- A computação ocorre no soma:

$$\sum_{i=0}^n x_i \cdot w_i$$

$$x_0 = 1 \text{ e } w_0 = \beta = -\theta$$

$\beta$  = bias e  $\theta$  = limiar de ativação (*threshold*).

# As RNA se abstram de detalhes dos neurônios reais



# As RNA se abstraem de detalhes dos neurônios reais

- Atrasos de condução são ignorados,
- Um sinal de saída ou é discreto (p. ex., 0 ou 1) ou é um número real (p. ex., entre 0 e 1),
- A entrada da rede é calculada como a soma ponderada dos sinais de entrada,
- A entrada da rede é transformada em um sinal de saída via uma função simples (p. ex., uma função de limiar).

# Sumário

1

## Modelos

- Modelo de um neurônio
- McCulloch-Pitts
- **Funções de ativação**

2

## Arquiteturas

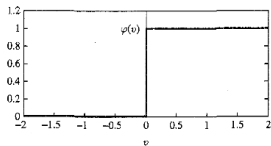
- Colunas e camadas
- Redes *Feedforward*
- Redes Recorrentes

3

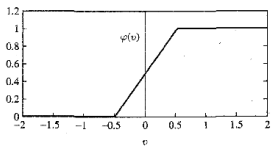
## Representação do Conhecimento

- Representação Conexionista
- Regras para Representação

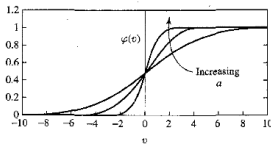
# Funções de ativação [2]



(a)



(b)



(c)

# Funções de ativação

- Função degrau (*heaviside*) [2]:

$$\varphi(v) = \begin{cases} 1, & \text{se } v \geq 0 \\ 0, & \text{se } v < 0 \end{cases}$$

$$y_k = \begin{cases} 1, & \text{se } v_k \geq 0 \\ 0, & \text{se } v_k < 0 \end{cases}$$

$$v_k = \sum_{j=1}^m w_{kj} x_j + b_k \quad (8)$$

# Funções de ativação

- Função linear (*piecewise-linear*) [2]:

$$\varphi(v) = \begin{cases} 1, & \text{se } v \geq +\frac{1}{2} \\ v, & \text{se } +\frac{1}{2} > v > -\frac{1}{2} \\ 0, & \text{se } v \leq -\frac{1}{2} \end{cases}$$



# Funções de ativação

- Função sigmoide [2]:

$$\varphi(v) = \frac{1}{1 + \exp(-av)} \quad (9)$$

onde  $a$  é o *parâmetro de inclinação* da função sigmoide.

$$\varphi(v) = \begin{cases} 1, & \text{se } v > 0 \\ 0, & \text{se } v = 0 \\ -1, & \text{se } v < 0 \end{cases}$$

$$\varphi(v) = \tanh(v) \quad (10)$$

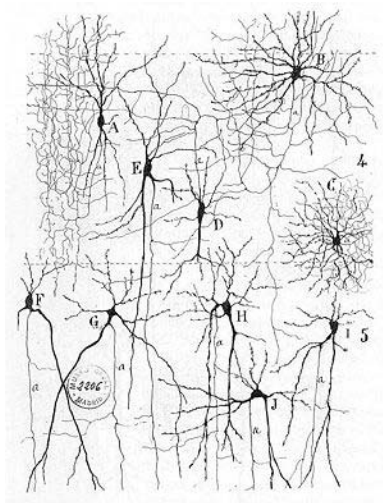
# Sumário

- 1 Modelos
  - Modelo de um neurônio
  - McCulloch-Pitts
  - Funções de ativação
- 2 **Arquiteturas**
  - **Colunas e camadas**
  - Redes *Feedforward*
  - Redes Recorrentes
- 3 Representação do Conhecimento
  - Representação Conexionista
  - Regras para Representação

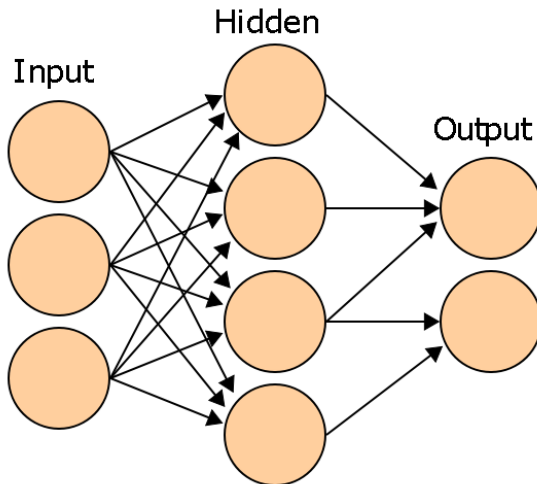
# Colunas e camadas

- Os neurônios no córtex cerebral estão dispostos em colunas e a maior parte das sinapses ocorre entre colunas diferentes.
- No modelo matemático extremamente simplificado, os neurônios estão dispostos em camadas (para representar as colunas) e há comunicação entre neurônios de camadas diferentes.

# Colunas no córtex



# Camadas da RNA



# Sumário

- 1 Modelos
  - Modelo de um neurônio
  - McCulloch-Pitts
  - Funções de ativação
- 2 **Arquiteturas**
  - Colunas e camadas
  - **Redes Feedforward**
  - Redes Recorrentes
- 3 Representação do Conhecimento
  - Representação Conexionista
  - Regras para Representação

# Redes Progressivas

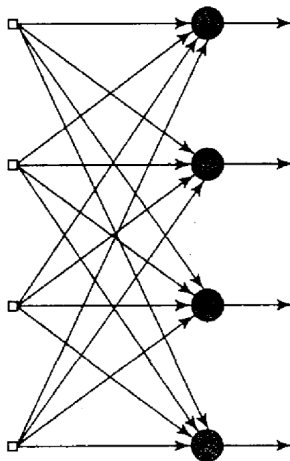
- A forma como os neurônios de uma rede neural estão estruturados depende do algoritmo de aprendizado usado para treinar a rede.
- Três classes [2]:
  - Redes Progressivas de Única Camada,
  - Redes Progressivas de Camadas Múltiplas,
  - Redes Recorrentes.

## Redes Progressivas de Única Camada [2]

- Em uma rede com camadas, os neurônios são organizados em camadas.
- A camada de entrada projeta uma camada de saída de neurônios (nós de computação), mas não vice-versa.
- Portanto, esta rede é apenas progressiva (*feedforward*) ou acíclica.
- A figura do próximo slide ilustra uma rede progressiva de única camada.
- A camada única designa a camada de saída de nós de computação.
- A camada de entrada não é contada, pois nela não há computação.



# Redes Progressivas de Única Camada [2]



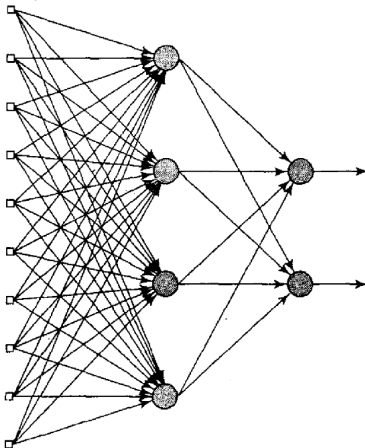
## Redes Progressivas de Camadas Múltiplas [2]

- Presença de uma ou mais *camadas escondidas*, cujos nós de computação são chamados de *neurônios escondidos* ou *unidades escondidas*.
- A função dos neurônios escondidos é intervir entre a entrada externa e a saída da rede de alguma maneira útil.
- Adicionando uma ou mais camadas escondidas, a rede torna-se capaz de extrair estatísticas de ordem mais alta.
- A habilidade dos neurônios escondidos de extrair estatísticas de ordem mais alta é mais valiosa quando o tamanho da camada de entrada é grande.

## Redes Progressivas de Camadas Múltiplas [2]

- O próximo slide apresenta um grafo de arquitetura de uma rede neural progressiva de múltiplas camadas, com uma única camada escondida.
- Trata-se de uma rede 10-4-2, pois possui 10 nós fonte (de entrada), 4 neurônios escondidos e 2 neurônios de saída.
- A rede é chamada de *totalmente conectada*, já que cada nó em cada camada da rede é conectado a todo nó da camada posterior adjacente.
- No caso de algum *link* de comunicação (conexão sináptica) estiver ausente, a rede é chamada de parcialmente conectada.

# Redes Progressivas de Camadas Múltiplas [2]



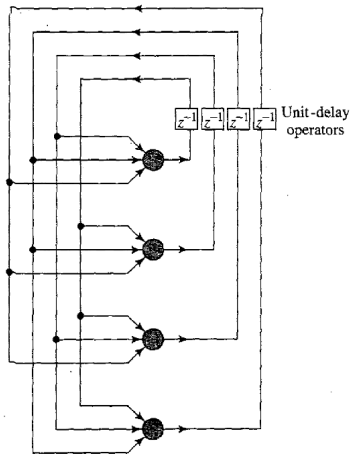
# Sumário

- 1 Modelos
  - Modelo de um neurônio
  - McCulloch-Pitts
  - Funções de ativação
- 2 **Arquiteturas**
  - Colunas e camadas
  - Redes *Feedforward*
  - **Redes Recorrentes**
- 3 Representação do Conhecimento
  - Representação Conexionista
  - Regras para Representação

## Redes Recorrentes [2]

- Uma *Rede Neural Recorrente* possui pelo menos um *loop* de retro-alimentação.
- Por exemplo, uma rede recorrente pode consistir de uma única camada de neurônios com cada neurônio retornando seu sinal de saída de volta às entradas de todos os neurônios, conforme figura do próximo slide.
- Nesta figura, não há *loops* de auto-retro-propagação e nem neurônios escondidos.

# Redes Recorrentes [2]

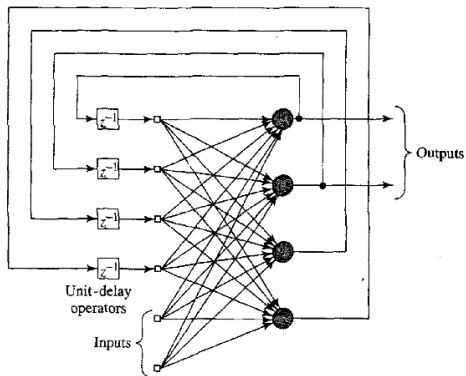


## Redes Recorrentes [2]

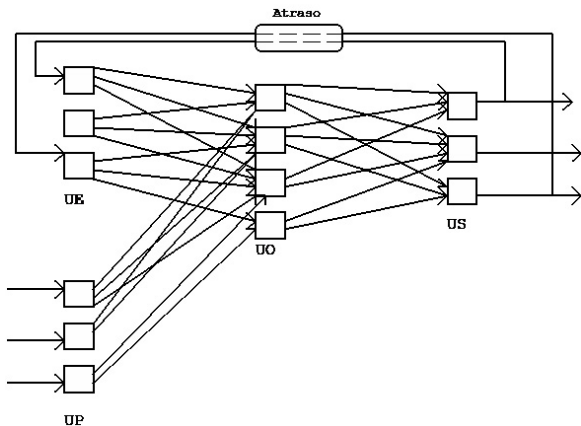
- Na figura que se segue, há uma rede recorrente com neurônios escondidos.
- Neste caso, as conexões de retro-alimentação originam tanto dos neurônios escondidos quanto dos neurônios de saída.
- A presença de *loops* de retro-alimentação tem um profundo impacto na capacidade de aprendizado e na performance da rede.
- Os *loops* envolvem os *elementos de atraso* representados por  $z^{-1}$ , que resultam num comportamento dinâmico não-linear.



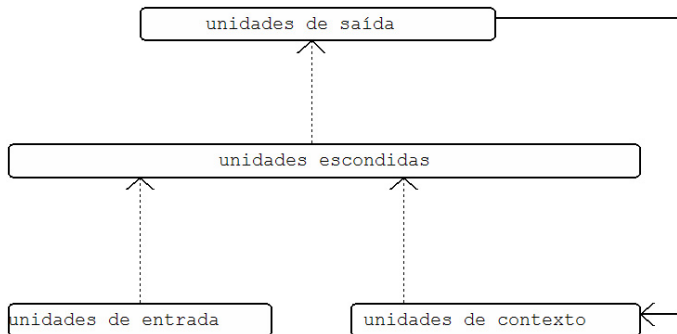
# Redes Recorrentes [2]



# Rede Recorrente de Jordan [3]



# Rede Recorrente de Elman [1]



# Sumário

- 1 Modelos
  - Modelo de um neurônio
  - McCulloch-Pitts
  - Funções de ativação
- 2 Arquiteturas
  - Colunas e camadas
  - Redes *Feedforward*
  - Redes Recorrentes
- 3 Representação do Conhecimento
  - Representação Conexionista
  - Regras para Representação

# Definição de “conhecimento”

- “Conhecimento se refere à informação armazenada ou a modelos usados por uma pessoa ou máquina para interpretar, predizer e responder apropriadamente ao mundo externo” [2].
- Representação do conhecimento:
  - Que informação é realmente tornada explícita,
  - Como a informação é fisicamente codificada para uso posterior.
- Na rede neural, a representação é um desafio de projeto.

## Representação e Redes Neurais [2]

- A principal tarefa de uma RNA é aprender um modelo do mundo (ambiente) no qual está inserida e manter o modelo suficientemente consistente com o mundo real, de tal forma que alcance as metas especificadas da aplicação de interesse.
- Conhecimento de mundo consiste de dois tipos de informação:
  - O estado do mundo conhecido, representado por fatos sobre o que é conhecido.
  - Observações (medições) do mundo, obtidas por meio de sensores projetados para “sentir” o ambiente no qual a RNA deve operar.
- As observações obtidas provêm informação da qual são obtidos *exemplos* usados para treinar a RNA.

## Representação e Redes Neurais [2]

- Os exemplos podem ser *rotulados* ou *não-rotulados*.
- Nos exemplos rotulados, cada exemplo que representa um *senal de entrada* é emparelhado com uma *resposta desejada* (isto é, saída alvo).
- Os exemplos não-rotulados consistem de realizações diferentes do sinal de entrada.
- Em cada evento, um conjunto de exemplos, rotulados ou não, representa conhecimento sobre o ambiente de interesse que uma RNA pode aprender através do treinamento.

## Representação e Redes Neurais [2]

- Um conjunto de pares entrada-saída, com cada par consistindo de um sinal de entrada e da resposta desejada correspondente, é conhecido como um *conjunto de dados de treinamento*.
- Como ilustração, considere o problema do *reconhecimento de dígito manuscrito*.
- Neste problema, o sinal de entrada consiste de uma imagem com pixels branco ou preto, representando um dos 10 dígitos.
- A resposta desejada é definida pela *identidade* do dígito em particular, cuja imagem é apresentada à rede como sinal de entrada.



# Representação e Redes Neurais

- Dado tal conjunto de exemplos, o projeto de uma RNA consiste em selecionar uma arquitetura apropriada. Por exemplo, para uma rede de três camadas:
  - quantos neurônios deve-se ter na entrada da rede (que corresponde, normalmente, ao número de bits que representa o padrão),
  - quantos deve-se ter na saída (que corresponde, normalmente, a quantidade de bits do padrão de saída) e,
  - o número de neurônios na camada escondida (o mais difícil).

# Representação e Redes Neurais

- Os neurônios da camada escondida normalmente não são “calculados” e devem ser tentados empiricamente.
- Depois de construída a RNA, deve-se escolher um algoritmo conexionista para “treinar” a rede (**fase de aprendizado**).
- O treinamento da rede normalmente é demorado, pois requer muitos “ciclos”, ou seja, deve-se mostrar à rede várias vezes, tudo que se deseja que ela aprenda.
- Depois do treinamento, a RNA deve ser capaz de, numa única propagação (único ciclo) reconhecer o padrão no qual ela foi ensinada (**fase de reconhecimento**).
- Este processo é muito mais rápido do que a execução de qualquer instrução simbólica.

# Sumário

- 1 Modelos
  - Modelo de um neurônio
  - McCulloch-Pitts
  - Funções de ativação
- 2 Arquiteturas
  - Colunas e camadas
  - Redes *Feedforward*
  - Redes Recorrentes
- 3 Representação do Conhecimento
  - Representação Conexionista
  - **Regras para Representação**

# Representação e Redes Neurais

- Numa RNA de uma determinada arquitetura, a representação de conhecimento do ambiente na qual está inserida é definida pelos valores dos parâmetros livres (pesos sinápticos e biases) da rede.
- A forma desta representação de conhecimento em uma RNA é, entretanto, muito complicada.
- Quatro regras para a representação de conhecimento [2]:
  - 1 Entradas similares de classes similares devem produzir representações similares dentro da RNA.
  - 2 Itens categorizados como classes separadas devem ter representações bem diferentes na rede.
  - 3 Se um determinado traço (ou característica) é importante, então deve haver um grande número de neurônios envolvidos na representação deste item na rede.
  - 4 Informação prévia e invariâncias devem ser incluídas no projeto da RNA, para a rede não precisar aprendê-las.

# Representação e Redes Neurais

- REGRA 1 [2]: Uma forma comum de medir a similaridade é baseada na *distância Euclidiana*. Seja  $x_i$  um vetor  $m$ -por-1:

$$x_i = [x_{i1}, x_{i2}, \dots, x_{im}]^T \quad (11)$$

O vetor  $x_i$  define um ponto no espaço  $m$ -dimensional (espaço Euclidiano) denotado por  $\mathbb{R}^m$ . A distância Euclidiana entre vetores  $x_i$  e  $x_j$  é dada por

$$d(x_i, x_j) = \|x_i - x_j\| = \left[ \sum_{k=1}^m (x_{ik} - x_{jk})^2 \right]^{\frac{1}{2}}. \quad (12)$$

onde  $x_{ik}$  e  $x_{jk}$  são os  $k$ -ésimos elementos dos vetores de entrada  $x_i$  e  $x_j$

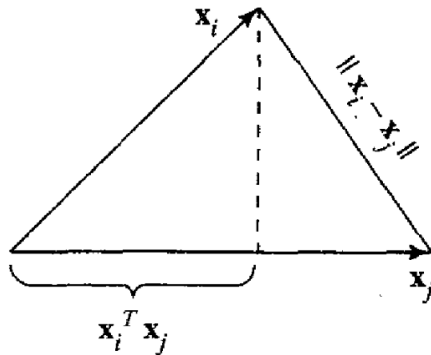
# Representação e Redes Neurais

- REGRA 1 [2]: Uma outra forma de medir a similaridade é baseada na ideia do *produto interno*.
- Dado um par de vetores  $x_i$  e  $x_j$  de mesmas dimensões, seu produto interno é  $x_i^T x_j$  dado por:

$$x_i^T x_j = \sum_{k=1}^m x_{ik} x_{jk} \quad (13)$$

- O produto interno  $(x_i, x_j)$  dividido por  $\|x_i\| \|x_j\|$  é o cosseno do ângulo entre os vetores  $x_i$  e  $x_j$ .
- Ou seja, as duas medidas de similaridades estão relacionadas.
- A figura do próximo slide mostra a relação entre o produto interno e a distância Euclidiana como medidas de similaridade entre padrões.

# Medidas de Similaridade [2]



# Representação em RNA

- Para que a rede conexionista possa receber e enviar informação de/para seu meio-ambiente, é essencial que essa informação possa ser representada.
- Dois tipos:
  - representação externa:
    - usada nas unidades de entrada e saída.
  - representação interna:
    - a representação que se pode associar às unidades escondidas.
- Duas técnicas para a representação **externa** em redes conexionistas:
  - Representação localista,
  - Representação distribuída.



# Representação em RNA

- Representação localista:
  - cada entidade ou conceito ou característica é representada por uma única unidade processadora,
  - a semântica de uma unidade independe das interpretações associadas às outras unidades,
  - pouco plausível biologicamente,
  - pouco econômica em recursos e não permite representar semelhanças entre conceitos.

# Representação em RNA

- Representação distribuída:
  - cada entidade é representada por um padrão de ativação distribuído entre mais de uma unidade,
  - cada unidade participa da representação de mais de uma entidade,
  - a interpretação de uma dada representação só pode ser obtida a partir da análise do padrão global de ativação de várias unidades,
  - a interpretação da ativação de uma unidade isolada pode não ter sentido,
  - a semântica associada a uma representação conexionista distribuída não é composicional,
  - mais plausível biologicamente,

# Representação em RNA

- Representação distribuída (continuação):
  - mais econômica,
  - poder representacional muito maior - permite representar semelhanças entre conceitos,
  - são radicalmente diferentes das representações simbólicas.
- A escolha da representação da informação externa é um ponto chave na modelagem:
  - redes conexionistas capturam regularidades estatísticas presentes no seu meio-ambiente (conjunto de treinamento),
  - a rede só capta a informação que a representação externa possa codificar,
  - a representação externa pode ser a diferença entre a rede aprender ou não.

# Bibliografia I

- [1] J. L. Elman  
*Finding Structure in Time.*  
Technical Report, Center for Research in Language, UCSD,  
Number CRL-8801, April 1988.
- [2] S. Haykin  
*Neural networks - a comprehensive foundation.*  
2nd. edition. Prentice Hall, 1999.
- [3] M. I. Jordan  
*Attractor Dynamics and Parallelism in a Connectionist  
Sequential Machine.*  
*Proceedings of the Eighth Annual Cognitive Science  
Society Conference*, pp. 531–546, Amherst, MA. Erlbaum,  
1986.

# Bibliografia II

[4] W. S. McCulloch and W. Pitts

*A logical calculus of the ideas immanent in nervous activity*  
*Bulletin of Mathematical Biophysics*, 5, pp. 115-133, 1943.