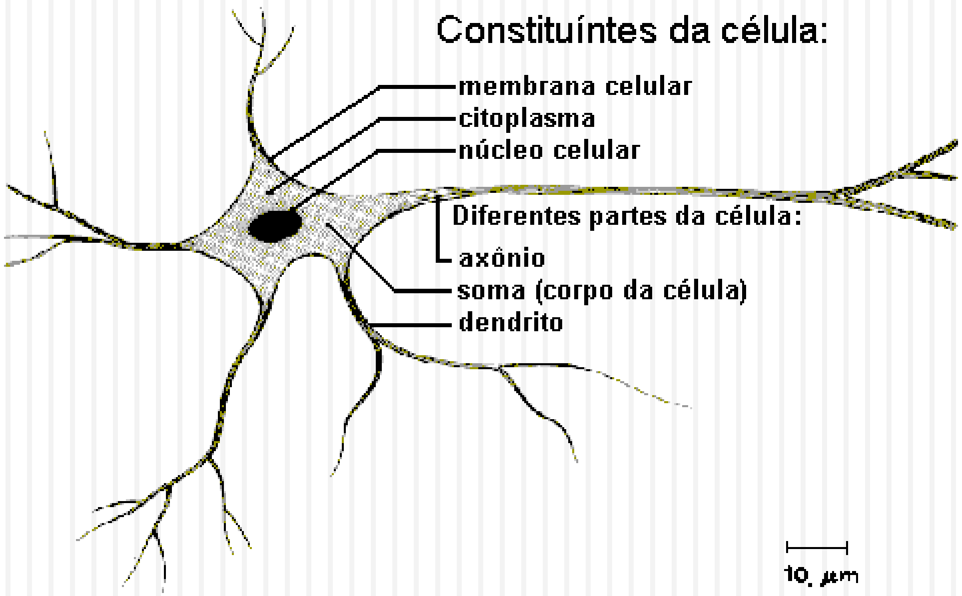
REDES NEURAIS ARTIFICIAIS

Benjamin Grando Moreira

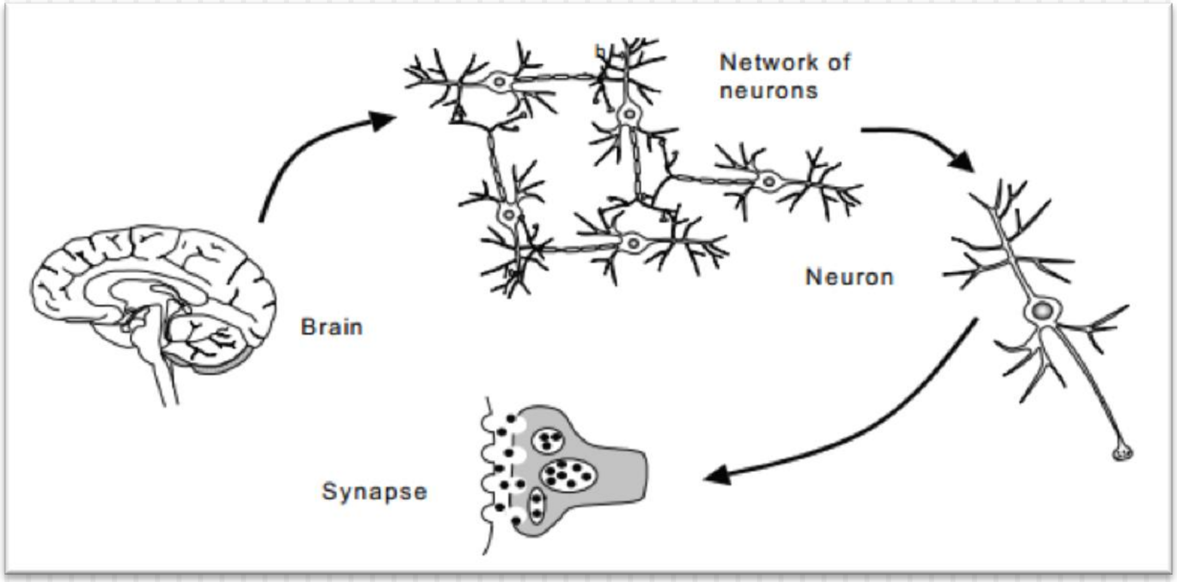
Introdução

- Técnicas computacionais que apresentam um modelo matemático inspirado na estrutura neural
- Os neurônios são formados pelos dendritos (terminais de entrada), pelo corpo central, e pelos axônios (terminais de saída)



Inspiração biológica

- Os neurônios se comunicam através de **sinapses**, através de impulsos nervosos transmitidos entre eles.
- Os impulsos recebidos por um neurônio são processados, e atingindo um dado limiar de ação, o neurônio dispara, produzindo uma substância neurotransmissora que flui do corpo celular para o axônio, que pode estar conectado a um dendrito de um outro neurônio.

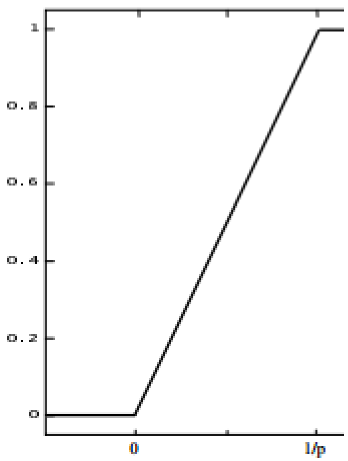


Neurônio Artificial

- Cada neurônio (ou nó) em uma rede recebe uma série de entradas. Uma função chamada de **função de ativação** é aplicada a esses valores de entrada, o que resulta no **nível de ativação** do neurônio, que é o valor de saída do neurônio.
- Há uma série de funções que podem ser utilizadas nos neurônios. Algumas das funções mais comumente usadas são: função degrau, sigmoide e linear.

Funções de ativação

$$f(\mathbf{u}_k)=\begin{cases}1 & \text{se}\\ p\mathbf{u}_k & \text{se}\\ 0 & \text{se}\end{cases}$$

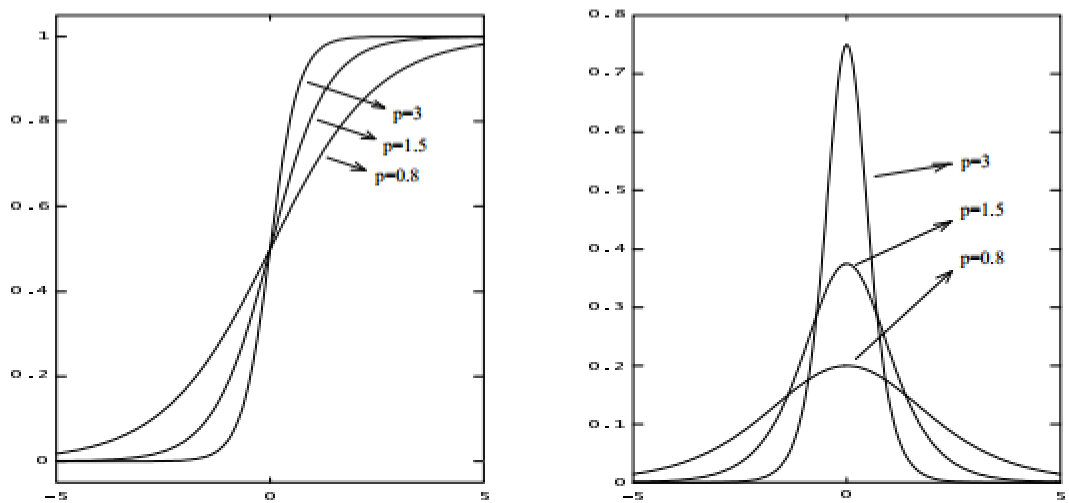


a)

Função semi-linear (a) e s

$$y=f(\mathbf{u}_k)=\frac{e^{p\mathbf{u}_k}}{e^{p\mathbf{u}_k}+1}=\frac{1}{1+e^{-p\mathbf{u}_k}}$$

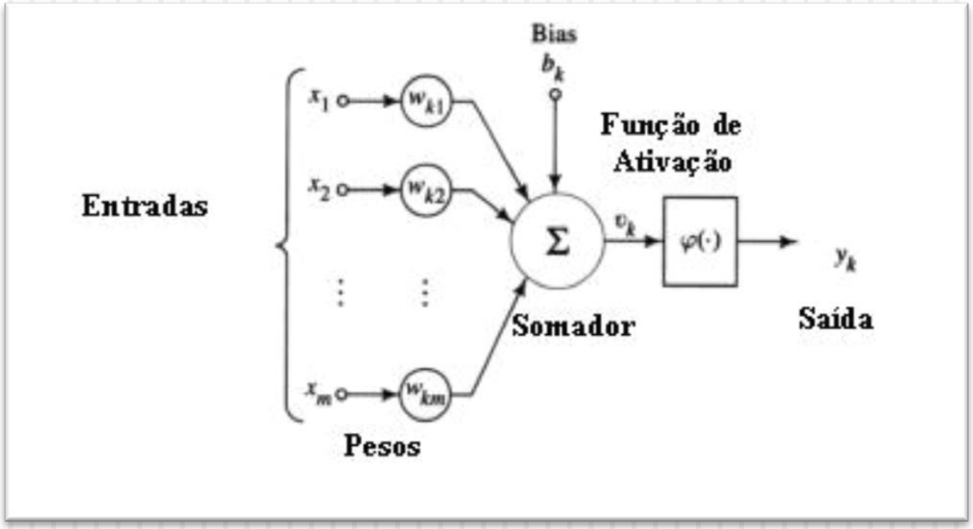
$$\frac{\partial y}{\partial \mathbf{u}_k}=p\mathbf{u}_k(1-\mathbf{u}_k)>0$$

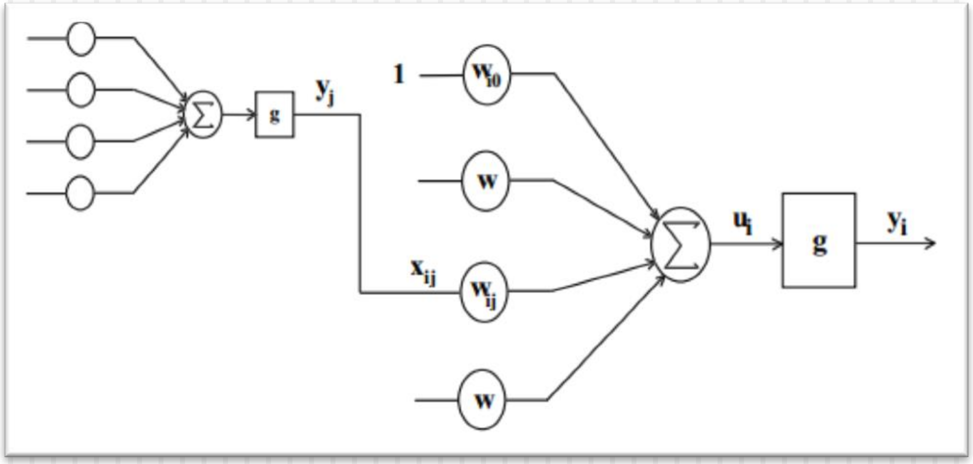


a) b)

Função logística (a) e sua derivada em relação à entrada interna (b)

Neurônio artificial





- Na função degrau as entradas são somadas (tendo cada uma sido multiplicada por um peso) e esta soma é comparada com um limiar. Se a soma for maior o neurônio ativará.

$$X = \sum_{i=1}^{n} w_i x_i$$

Características gerais do neurônio

- Sinais de entrada $X_1, X_2, ..., X_p$ (0 ou 1)
- Pesos $w_1, w_2, ..., w_p$, valores reais
- Somador: genericamente dado por:
  - $a = w_1X_1 + w_2X_2 + ... + w_pX_p$
- A função de ativação é geralmente utilizada para limitar a saída do neurônio e introduzir não-linearidade no modelo
- A saída y é dada por:
  - $y = 1$, se $a >= t$ ou
  - $y = 0$, se $a < t$.
- Bias: aumenta ou diminui a influência do valor da entrada líquida para a ativação do neurônio

Exemplo de uso da função degrau

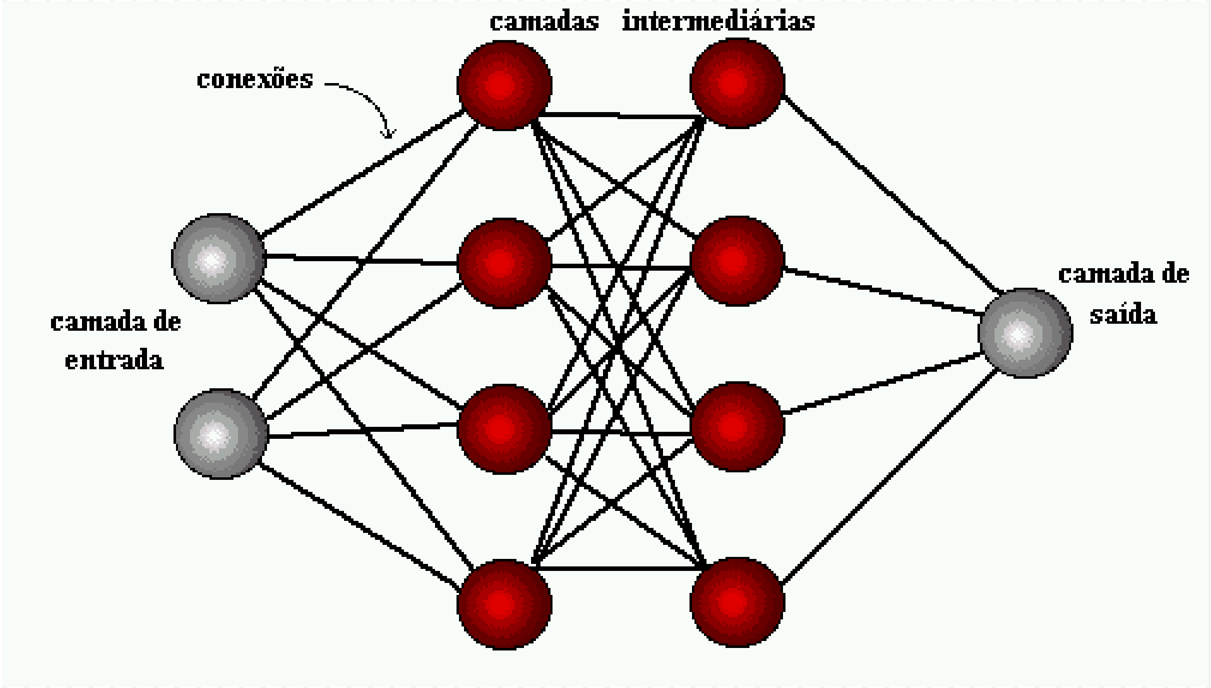
- Pesos:
  - $W_1 = 0{,}8$
  - $W_2 = 0{,}4$
- Entradas:
  - $X_1 = 0{,}7$
  - $X_2 = 0{,}9$
- Peso somado dessas entradas:
  - $(0{,}8 \times 0{,}7) + (0{,}4 \times 0{,}9) = 0{,}92$
- Nível de ativação é definido como:
  - $Y = \begin{cases} 1 \; para \; X > t \\ 0 \; para \; X \leq t \end{cases}$

As RNA aprendem alterando-se o peso associado a cada conexão

Características gerais da rede

- Uma rede neural artificial é composta por várias unidades de processamento
- Essas unidades são conectadas por canais de comunicação que estão associados a determinado peso
- As unidades fazem operações apenas sobre seus dados locais, que são entradas recebidas pelas suas conexões

Organização em camadas



Aprendizado

- Treinamento: processo iterativo de ajustes aplicado a seus pesos
- O aprendizado ocorre quando a rede neural atinge uma solução generalizada para uma classe de problemas
- Diversos tipos de algoritmos de aprendizado

Aprendizado

- **Aprendizado supervisionado:** é indicado à rede a resposta desejada para o padrão de entrada;
- **Aprendizado não supervisionado** (auto-organizado): não existe uma indicação da resposta desejada para os padrões de entrada;
- **Reforço:** um crítico externo avalia a resposta fornecida pela rede.

Perceptrons

- Neurônio simples usado para classificar suas entradas em uma de duas categorias
- Usa a função degrau
- Podem apenas aprender a modelar funções que sejam linearmente separáveis

Perceptrons – processo de aprendizado

1. Pesos aleatórios são atribuídos às entradas
   - Tipicamente valores entre -0,5 e 0,5
2. Um item de treinamento é apresentado ao *perceptron* e sua classificação é observada
3. Pesos são ajustados para classificar mais acertadamente
   - Regra de treinamento do perceptron: se a saída for muito alta, decrescer o peso
   - Fórmula para modificação: $w_i \leftarrow w_i + (a \times x_i \times e)$
     - e: é o **erro**
     - *a*: é a **taxa de aprendizado** $(0 < a < 1)$
4. Empregar o próximo fragmento de dados
   - Cada iteração é conhecida como uma época

Perceptrons - exemplo

- Classificar a função lógica OU para duas entradas
- Limiar de zero ($t = 0$)
- Taxa de aprendizado de 0,2

- **Passo 1:** pesos aleatórios para as entradas
  - $W_1 = -0,2$
  - $W_2 = 0,4$

- Inicia a primeira época

Perceptron - exemplo

- Primeiro item:
  - Dados de treinamento: $X_1 = 0$; $X_2 = 0$
  - Saída esperada: 0
  - Degrau ((0 x -0,2) + (0 x 0,4)) = 0
  - Erro = 0
- Segundo item:
  - Dados de treinamento: $X_1 = 0$; $X_2 = 1$
  - Saída esperada: 1
  - Degrau ((0 x -0,2) + (1 x 0,4)) = 1
  - Erro = 0

Perceptrons - exemplo

- Terceiro item:
  - Dados de treinamento: $X_1 = 1$; $X_2 = 0$
  - Saída esperada: 1
  - Degrau ((1 x -0,2) + (0 x 0,4)) = degrau(-0,2) = 0
  - Erro = 1
  - Ajuste de pesos:
    - Fórmula para modificação: $w_i \leftarrow w_i + (a \times x_i \times e)$
      - W1 = -0,2 + (0,2 x 1 x 1) = 0
      - W2 = 0,4 + (0,2 x 0 x 1) = 0,4

Perceptrons - exemplo

- Quarto item:
  - Dados de treinamento: $X_1 = 1$; $X_2 = 1$
  - Saída esperada: 1
  - Degrau ((1 x 0) + (1 x 0,4)) = 1
  - Erro = 0

- Terminou a primeira época

- Repete o processo até que todos os dados sejam classificados corretamente

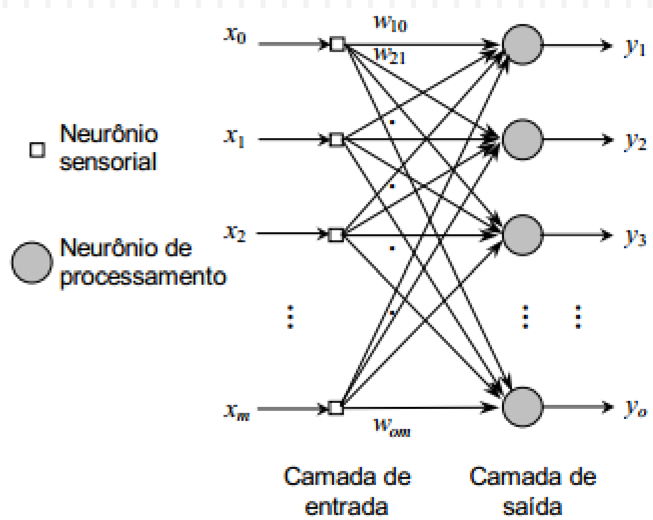
Arquitetura de RNAs

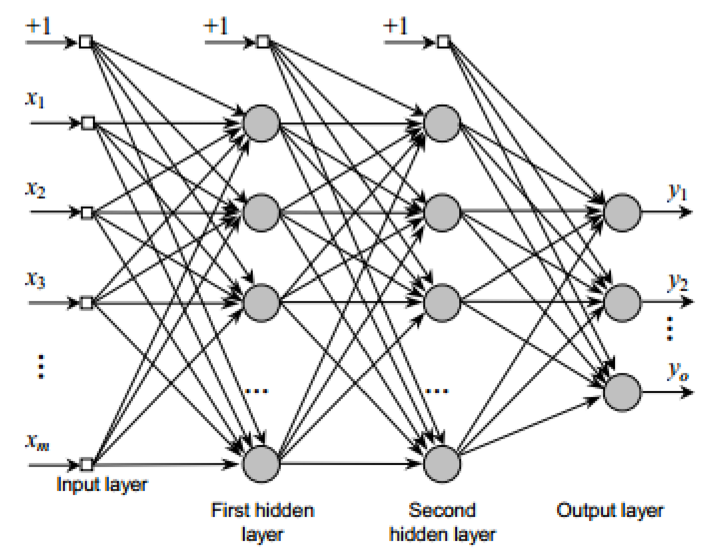
Feedforward de única camada

- Consiste em uma camada de entrada e uma camada de saída
- Propagação do sinal ocorre apenas da entrada para a saída
- Geralmente os neurônios de entrada (neurônios sensoriais) são lineares, ou seja, eles simplesmente propagam o sinal de entrada para a próxima camada

Feedforward de múltiplas camadas

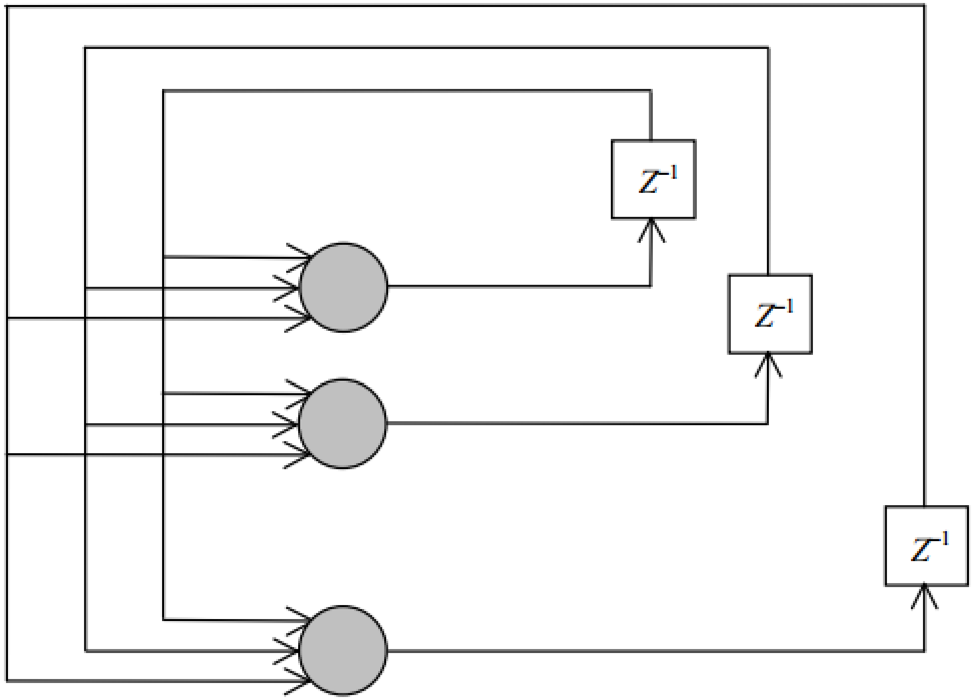
- Possuem uma ou mais camadas intermediárias ou escondidas.
- A saída de cada camada intermediária é utilizada como entrada para a próxima camada





Redes recorrentes

- Possuem pelo menos um laço realimentando a saída de neurônios para outros neurônios da rede



Rede neural recorrente de Hopfield.

Rede de Hopfield

- Forma de rede recorrente
- Considerada uma **memória autoassociativa**
- Usa função de **ativação de sinal**
  - Sinal(X) = $\begin{cases} +1\ para\ X > 0 \\ -1\ para\ X < 0 \end{cases}$
  - Ao receber 0 permanece no mesmo estado
- Pesos são representados por uma matriz W
  - W = $\sum_{i=1}^{N} X_i X^t i - N\ \mathrm{I}$
    - X_i é um vetor de entrada de tamanho m
    - $X^t i$ é a matriz transposta de X_i
    - N é o número de estados de X_i
    - I é a matriz identidade m x m

Rede de Hopfield - exemplo

- Rede com uma única camada, com cinco nós e três entradas de treinamento.
  - $X_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad X_2 = \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \\ -1 \end{bmatrix} \quad X_3 = \begin{bmatrix} 1 \\ -1 \\ 1 \\ 1 \\ -1 \end{bmatrix}$
  - A matriz de pesos será:
    - W $= \sum_{i=1}^{3} X_i X^t i - 3\ \mathrm{I}$

Rede de Hopfield - exemplo

$$= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} + \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \\ -1 \end{bmatrix} [-1 \; -1 \; -1 \; -1 \; -1]$$

$$+ \begin{bmatrix} 1 \\ -1 \\ 1 \\ 1 \\ -1 \end{bmatrix} [1 \; -1 \; 1 \; 1 \; -1] \quad - \quad 3 \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Rede de Hopfield - exemplo

$$= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$+ \begin{bmatrix} 1 \\ -1 \\ 1 \\ 1 \\ -1 \end{bmatrix} [1 \; -1 \; 1 \; 1 \; -1] \quad - \quad 3 \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Rede de Hopfield - exemplo

$$= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$+ \begin{bmatrix} 1 & -1 & 1 & 1 & -1 \\ -1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & 1 & -1 \\ -1 & 1 & -1 & -1 & 1 \end{bmatrix} - 3 \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Rede de Hopfield - exemplo

$$= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$+ \begin{bmatrix} 1 & -1 & 1 & 1 & -1 \\ -1 & 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & 1 & -1 \\ 1 & -1 & 1 & 1 & -1 \\ -1 & 1 & -1 & -1 & 1 \end{bmatrix} - \begin{bmatrix} 3 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 3 \end{bmatrix}$$

Rede de Hopfield - exemplo

$$= \begin{bmatrix} 0 & 1 & 3 & 3 & 1 \\ 1 & 0 & 1 & 1 & 3 \\ 3 & 1 & 0 & 3 & 1 \\ 3 & 1 & 3 & 0 & 1 \\ 1 & 3 & 1 & 1 & 0 \end{bmatrix}$$

Rede de Hopfield - exemplo

- Vetor de saída: $Y_i = sinal(W\ X_i$ - Θ)
  - Θ é a matriz de limiares

$$Y_1 = sinal\left(\begin{bmatrix} 0 & 1 & 3 & 3 & 1 \\ 1 & 0 & 1 & 1 & 3 \\ 3 & 1 & 0 & 3 & 1 \\ 3 & 1 & 3 & 0 & 1 \\ 1 & 3 & 1 & 1 & 0 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}\right) = sinal\left(\begin{bmatrix} 8 \\ 6 \\ 8 \\ 8 \\ 6 \end{bmatrix}\right) = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = X_1$$

Rede de Hopfield - exemplo

- $Y_2 = X_2$ e $Y_3 = X_3$
- Testar: $X_4 = \begin{bmatrix} 1 \\ 1 \\ -1 \\ 1 \\ 1 \end{bmatrix}$
- Testar: $X_5 = \begin{bmatrix} -1 \\ 1 \\ -1 \\ 1 \\ 1 \end{bmatrix}$
  - Como o resultado não é um dos atratores, necessita aplicar a regra novamente

Referências

- Ben Coppin. Inteligência Artificial. Rio de Janeiro: LTC, 2010.
- http://www.icmc.usp.br/pessoas/andre/research/neural/
- ftp://ftp.dca.fee.unicamp.br/pub/docs/vonzuben/ia006_03/topico5_03.pdf
- http://pt.slideshare.net/iaudesc/rna-redes-neurais-artificiais