Redes Neurais Artificiais

Introdução

Prof. João Marcos Meirelles da Silva

www.professores.uff.br/jmarcos

Departamento de Eng. de Telecomunicações
Escola de Engenharia
Universidade Federal Fluminense

Objetivo da Disciplina



Oferecer uma visão geral de redes neurais artificiais, seus diferentes paradigmas, possibilidades e restrições, bem como estudar as aplicações mais recentes em nosso dia-a-dia e destacar a importância do estudo nessa área.

Avaliação



A avaliação do curso se dará da seguinte forma:

- 1 Lista de Exercícios (L)
- 1 Prova (P)
- Apresentação do Projeto (PJ)
- 1 Prova Final (Verificação Suplementar)

Avaliação

$MF = 0{,}2*L + 0{,}4*P + 0{,}4*PJ$

Condições:

Se $MF \geq 6 \rightarrow$ 

Se $4 \leq MF < 6 \rightarrow$ 

Se $MF < 4 \rightarrow$ 

Avaliação

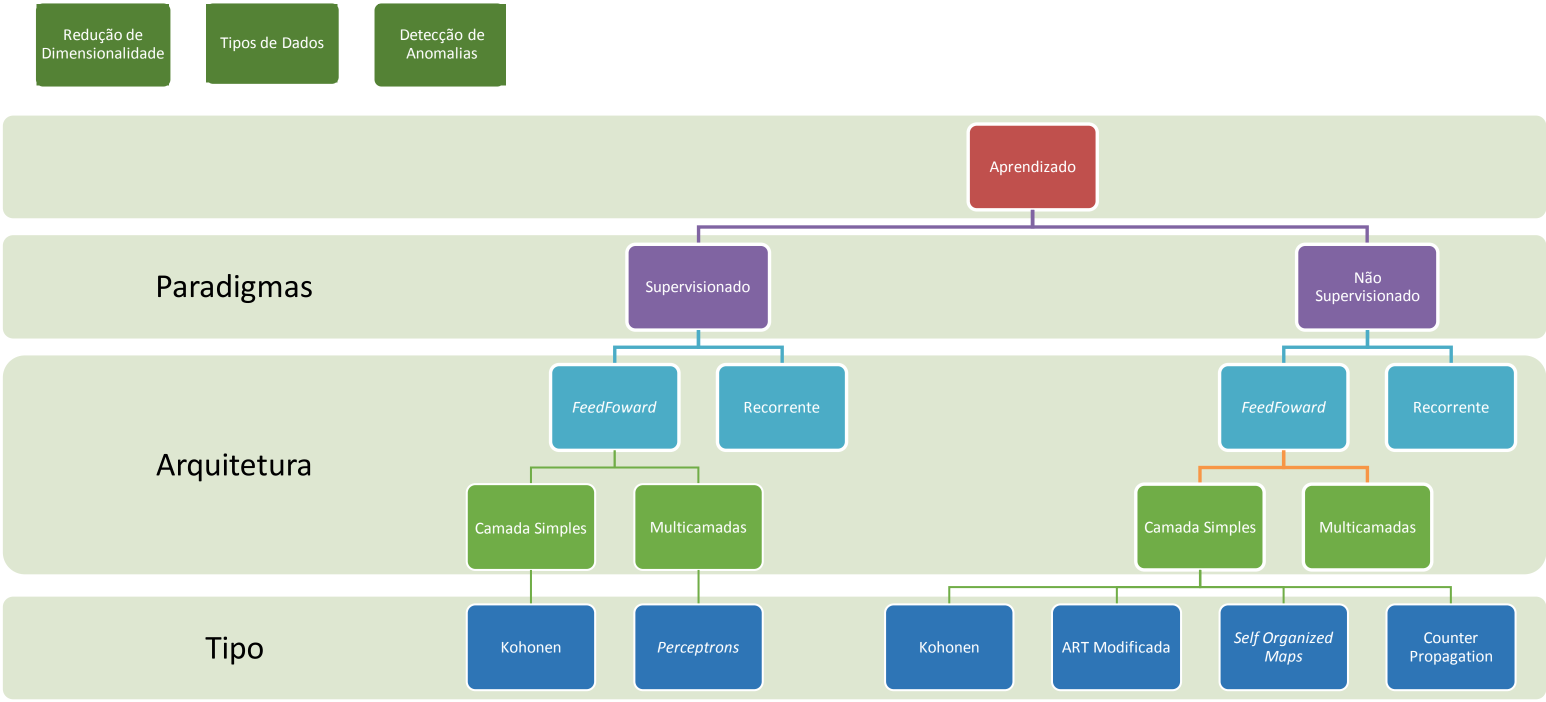


- Os alunos que fizerem a VS deverão alcançar a nota mínima de 6,0 para serem aprovados.

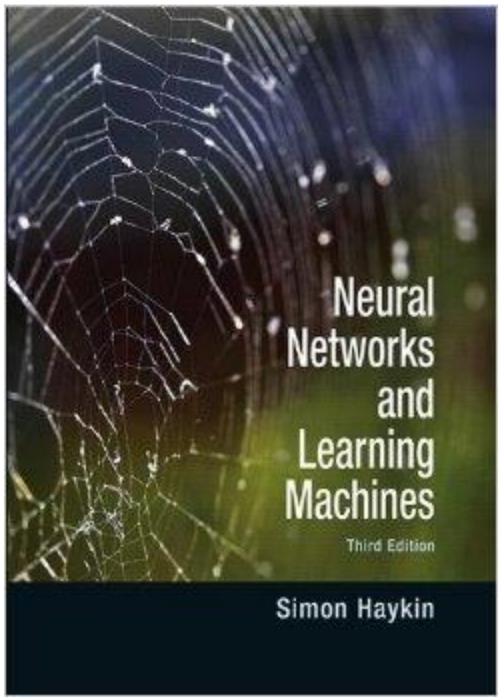
- Frequência mínima de 75% das aulas

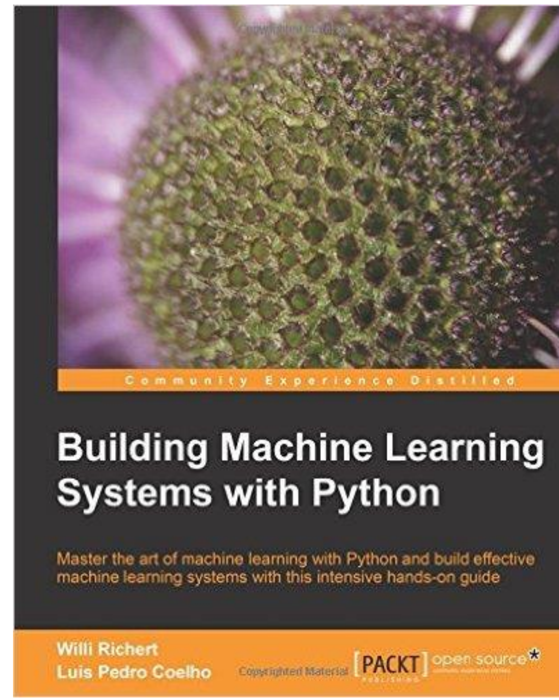
- Prova de Reposição é a matéria toda após a data da VS

Ementa


Redução de Dimensionalidade
Tipos de Dados
Detecção de Anomalias
Aprendizado
Paradigmas
Supervisionado
Não Supervisionado
Arquitetura
FeedFoward
Recorrente
FeedFoward
Recorrente
Camada Simples
Multicamadas
Camada Simples
Multicamadas
Tipo
Kohonen
Perceptrons
Kohonen
ART Modificada
Self Organized Maps
Counter Propagation

Bibliografia





Ferramentas

python

TensorFlow

scikit learn

Machine Learning with Scikit-Learn

MATLAB
The Language of Technical Computing

Introdução



Inteligência Artificial / Computacional

- Cognitiva, simbólica → IA Tradicional
- Evolucionista → Algoritmos Genéticos, vida artificial
- Conexionista → Redes Neurais Artificiais

Por quê RNA?

Desafio de utilizar máquinas para realizar tarefas "fáceis" para os seres humanos:

- Reconhecimento de escrita
- Reconhecimento de rostos
- Mapeamento de conhecimento tático
- Problemas de otimização com restrições conflitantes

→ Algoritmos "convencionais" não lidam bem com determinados problemas.

Histórico

- McCulloch e Putts (1943) → Pioneiros
- The *Organization of Behavior* de Donald Hebb (1949)
- Frank Rosemblat (1958) → Perceptron
- Widrow e Hoff (1960) → LMS ou Regra Delta
- Teuvo Kohonen (1972) → Mapas auto-organizáveis
- Rumelhart, Williams e Hinton (1986) → *Backpropagation*
- Ainda na década de 80, John Hopfield e David Tank → Redes neurais com pesos fixos.

Aplicações



- Processamento de sinais (vídeo, áudio, imagem e texto)
- Controle e automação
- Reconhecimento de padrões
- Mineração de dados
- Segmentação
- Diagnósticos em medicina
- e muitos outros...

Aplicações em Telecom.

- Segurança
  - Detecção de Intrusão e fraudes
  - Criptografia/Criptoanálise
  - Esteganografia e marcas d’água
  - Reconhecimento de voz e locutor

- Desempenho
  - Gerenciamento inteligente de redes
  - Compactação de dados
  - Predição de falhas
  - Correlação de alarmes

Vantagens



- Capacidade de aprender... e esquecer
- Generalização
- Robustez
- Não requer um modelo matemático do problema

Desvantagens



- Instabilidade nas redes com realimentações
- Problemas com diversos mínimos locais
- Problemas de aprendizagem
- Necessita de ajustes de parâmetros

RNA: Quando usar?

Existe um algoritmo (modelo matemático) que seja satisfatório para o problema em questão?

- SIM → Então use-o !
- NÃO → Então pense em usar redes neurais artificiais...

O Neurônio Biológico

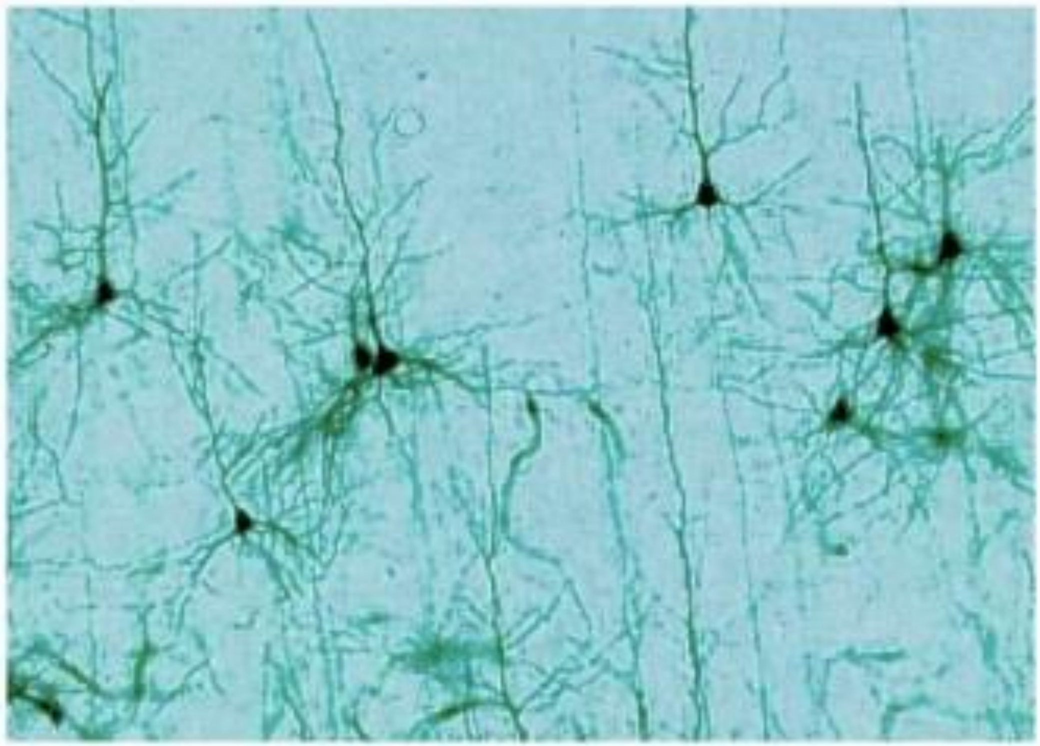


Figura 1: Neurônios do córtex cerebral

O Neurônio Biológico

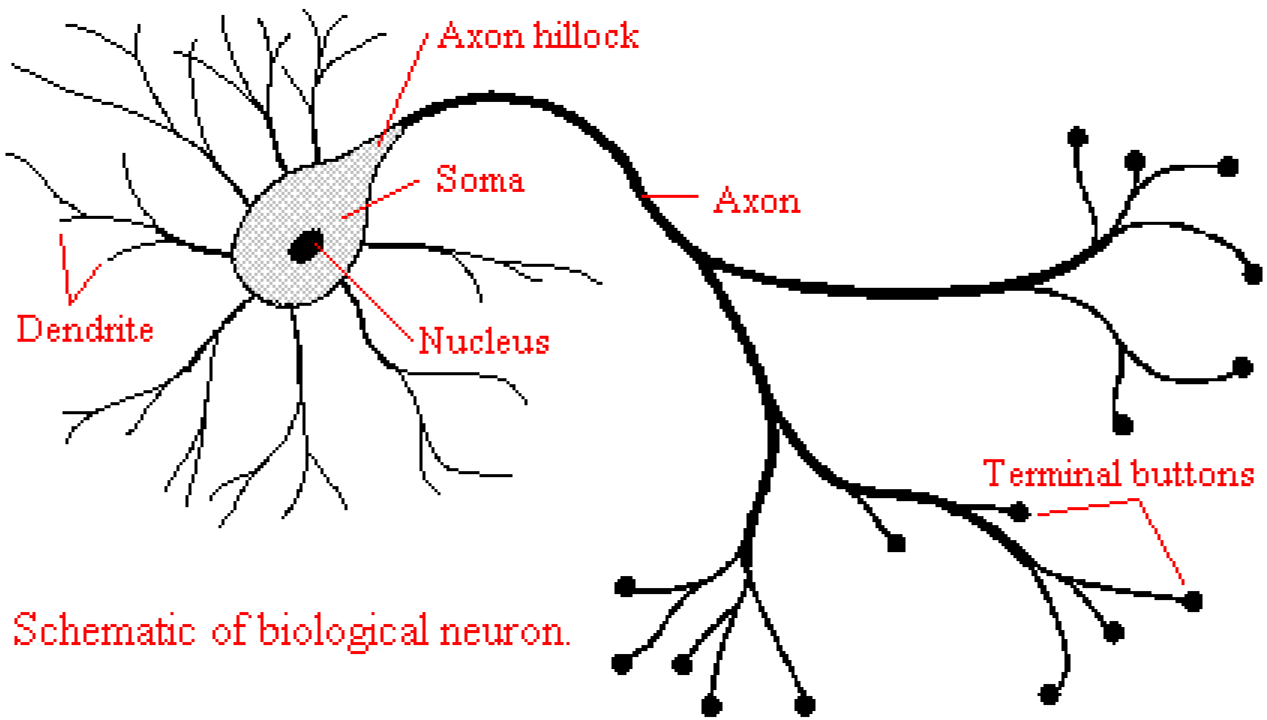


Figura 2: Representação de um neurônio

O Neurônio Biológico

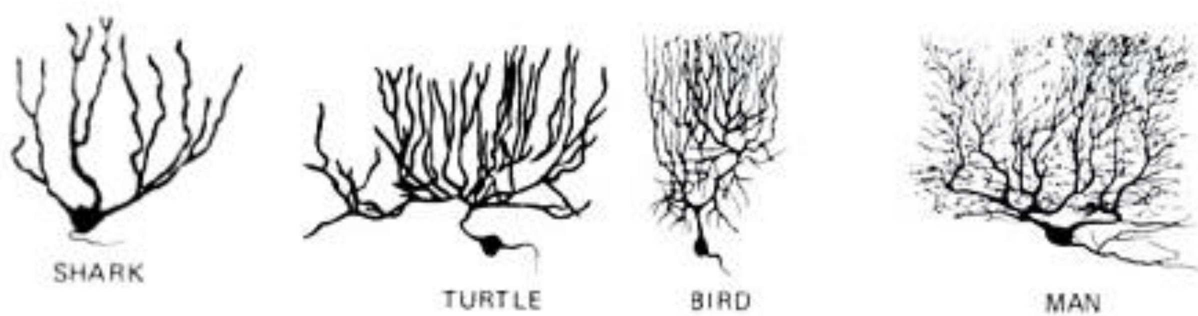


Figure 7-13. Growth of the dendritic trees and axon branches of cortical pyramidal cells in the human, from fetus to adult. (Courtesy of Sidman and Rakic 1982, and Poliakov in Sarkisov and Preobrazenskaya 1959.)

Figura 3: Neurônios em diferentes espécies e em diferentes estágios no ser humano.

O Neurônio Biológico



Em um neurônio biológico:

- Diversos sinais são recebidos através de outros neurônios ou terminações nervosas;
- Os sinais podem ser recebidos com diferentes intensidades (pesos) nas sinapses receptoras;
- As entradas são somadas;
- Sob circunstâncias apropriadas (entradas com nível suficiente), o neurônio “dispara” um sinal elétrico em direção à saída (axônio)

O Neurônio Biológico



- A saída de um neurônio pode ser a entrada de diversos outros (rede);
- A resposta a um estímulo pode alterar a "força" (peso) de uma sinapse → Aprendizado através da experiência;
- Neurotransmissores em uma sinapse podem ser *excitatórios* ou *inibitórios*.

O Neurônio Biológico

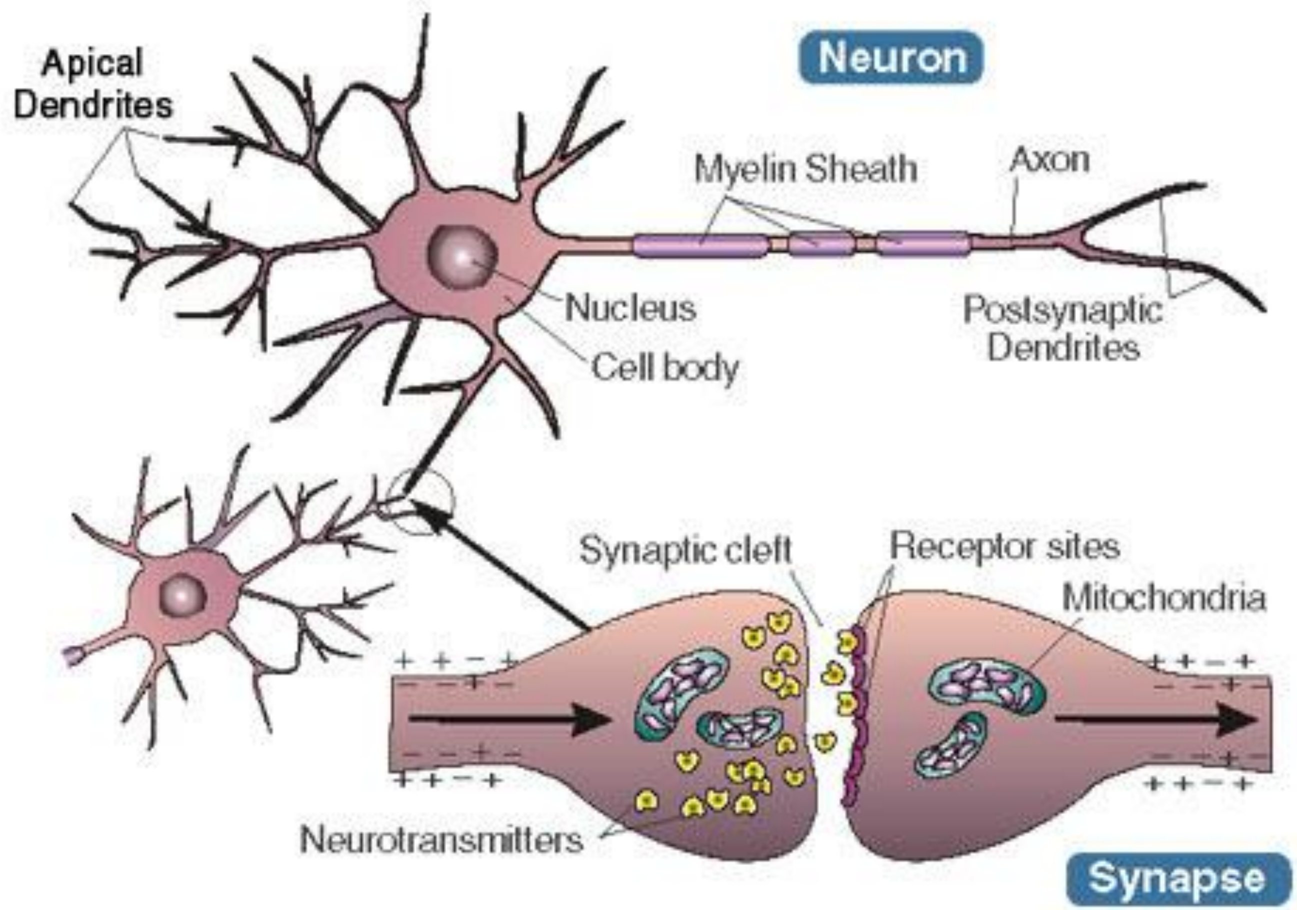


Figura 4: Sinapse entre dois neurônios biológicos.

O Neurônio Biológico

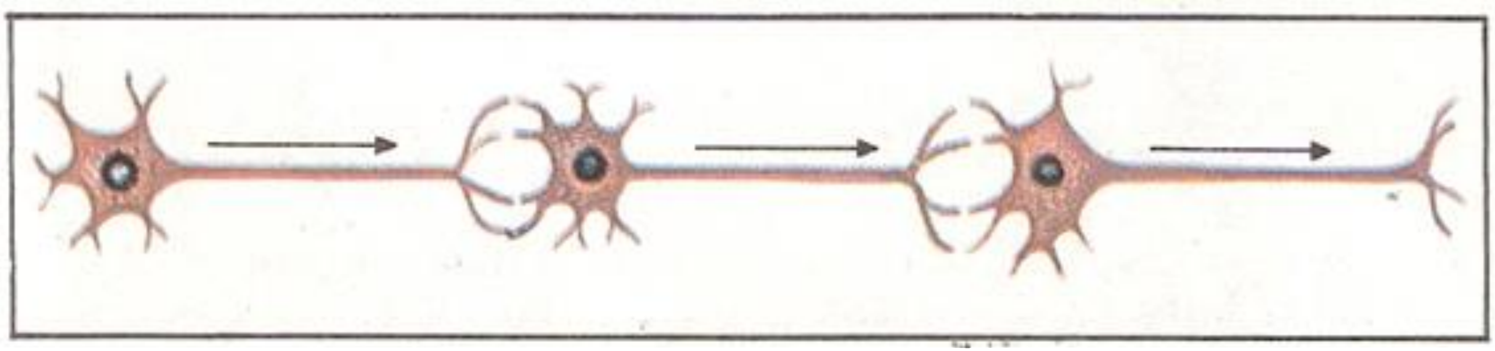
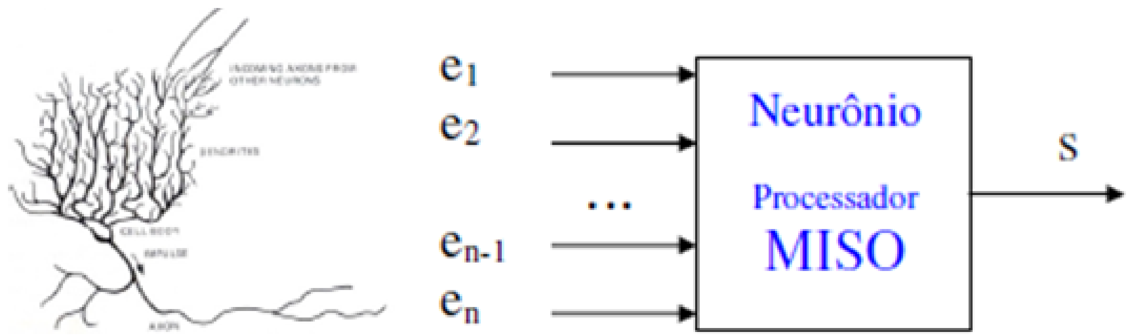


Figura 5: Sentido da propagação do sinal.

Modelando um Neurônio

- Modelo matemático de um neurônio

Um neurônio é uma unidade de processamento de informação essencial em uma rede neural biológica



$$\text{estado}\begin{cases}\text{ativo, excitado} & \text{saída} > s_0\\ \text{inativo, inativo} & \text{saída} < s_0\end{cases}$$

Figura 6: Modelagem de um neurônio biológico.

Modelando um Neurônio

O diagrama em blocos abaixo representa o modelo matemático de um neurônio...

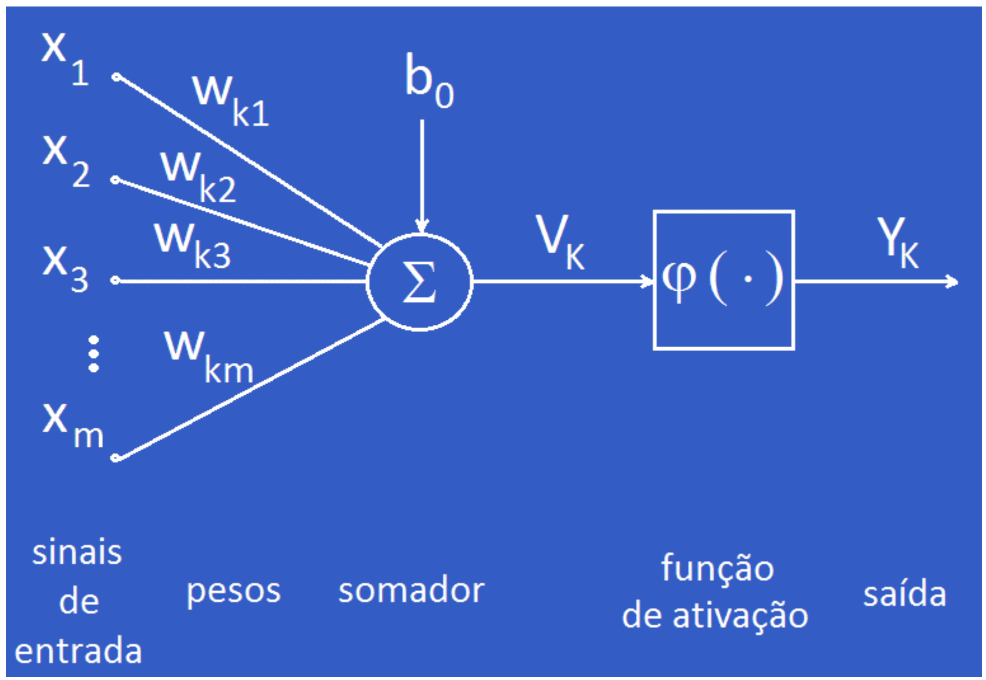


Figura 7: Modelo matemático de um neurônio artificial.

Modelando um Neurônio

... o qual forma a base para o projeto de redes neurais artificiais. Na Figura 7, temos:

- $x_1, x_2, \dots, x_m$ são os sinais de entrada
- $w_{k1}, w_{k2}, \dots, w_{km}$ são os pesos das sinapses do *k-ésimo* neurônio
- u_k é a combinação linear dos sinais de entrada ponderados por seus respectivos pesos
- b_k é o *bias* do *k-ésimo* neurônio.

Modelando um Neurônio

- $\varphi_k(\cdot)$ é a *função de ativação* do *k-ésimo* neurônio
- y_k é a saída do *k-ésimo neurônio*

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

$$v_k = u_k + b_k \qquad (2)$$

$$y_k = \varphi(u_k + b_k) \qquad (3)$$

Modelando um Neurônio

- O bias b_k é um parâmetro externo do *k-ésimo* neurônio artificial. Também pode ser interpretado como uma entrada x_0 = +1 com peso de sinapse igual a w_{k0} = b_k.

$$u_k = \sum_{j=0}^{n} w_{kj} x_j \,, \qquad x_0 = +1 \tag{4}$$

$$y_k = \varphi(u_k) \tag{5}$$

Modelando um Neurônio

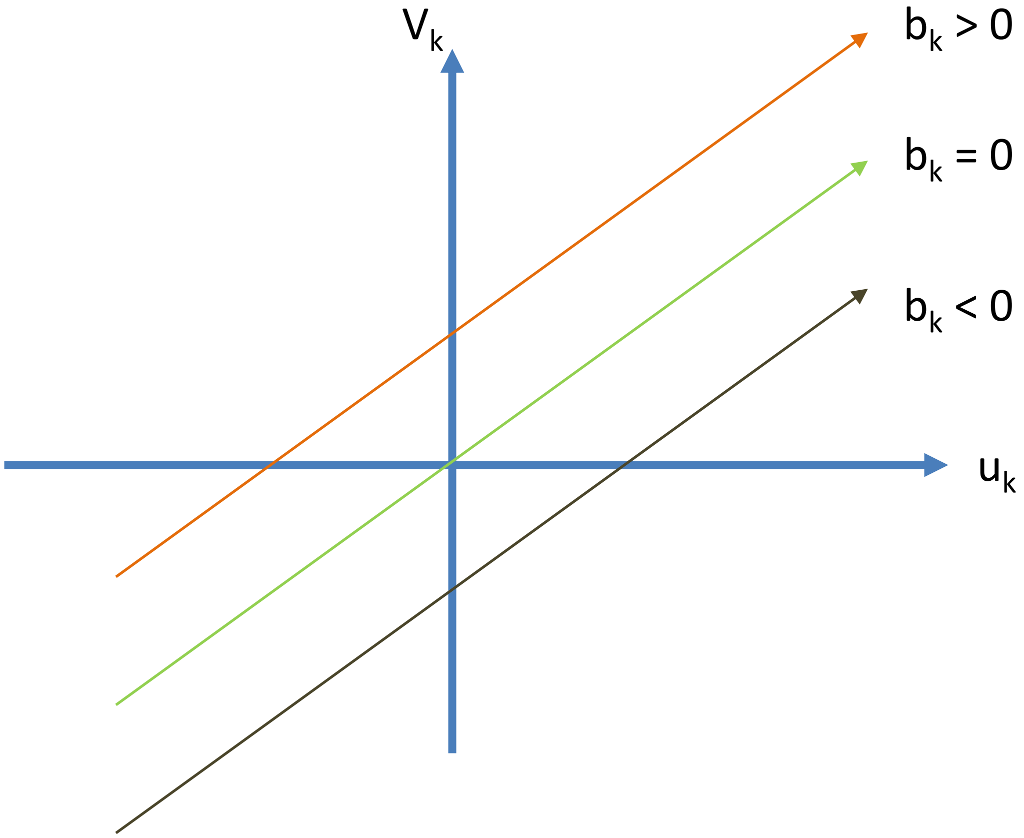


Figura 8: Efeito do bias sobre o neurônio artificial.

Modelando um Neurônio

Podemos reformular o modelo do neurônio mostrado na Figura 8 de forma a incorporar o *bias*.

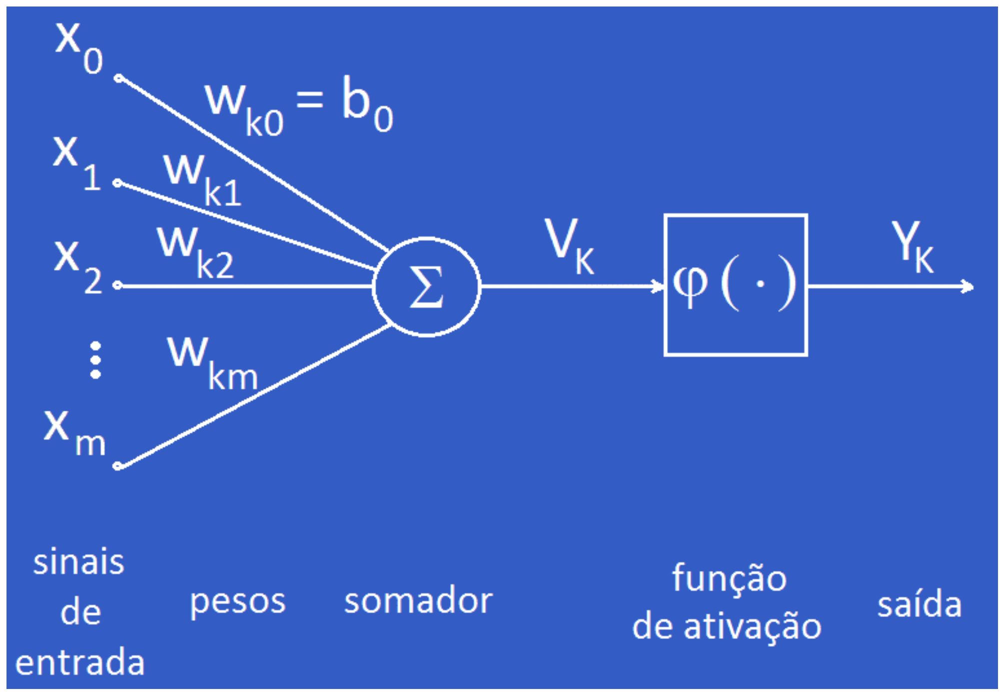


Figura 9: Segundo modelo matemático do neurônio artificial.

Modelando um Neurônio

Agora que o segundo modelo foi apresentado, podemos simplificar a sua representação gráfica apenas para maior conveniência.

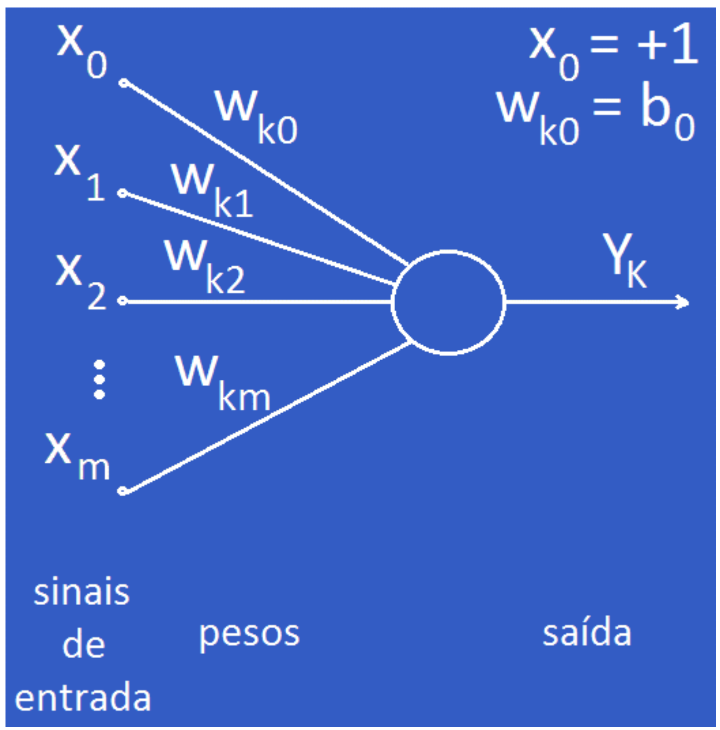


Figura 10: Modelo simplificado do neurônio artificial.

Exemplo



Seja a entrada de um neurônio dado pelo vetor $\underline{x} = [1.4, -2.5, 0.7]^T$ e seja o vetor de pesos das sinapses $\underline{w} = [0.4, 0.6, 0.3]^T$. Determine se a saída é positiva ou negativa nos casos:

1. Com *bias* unitário;

2. Sem *bias*.

Funções de Ativação



A função de ativação $\varphi(v)$ define a saída de um neurônio em termos do campo local induzido v. Basicamente, há 3 tipos de funções de ativação:

1. Função *threshold*
2. Função Linear por partes
3. Função sigmoidal

Funções de Ativação

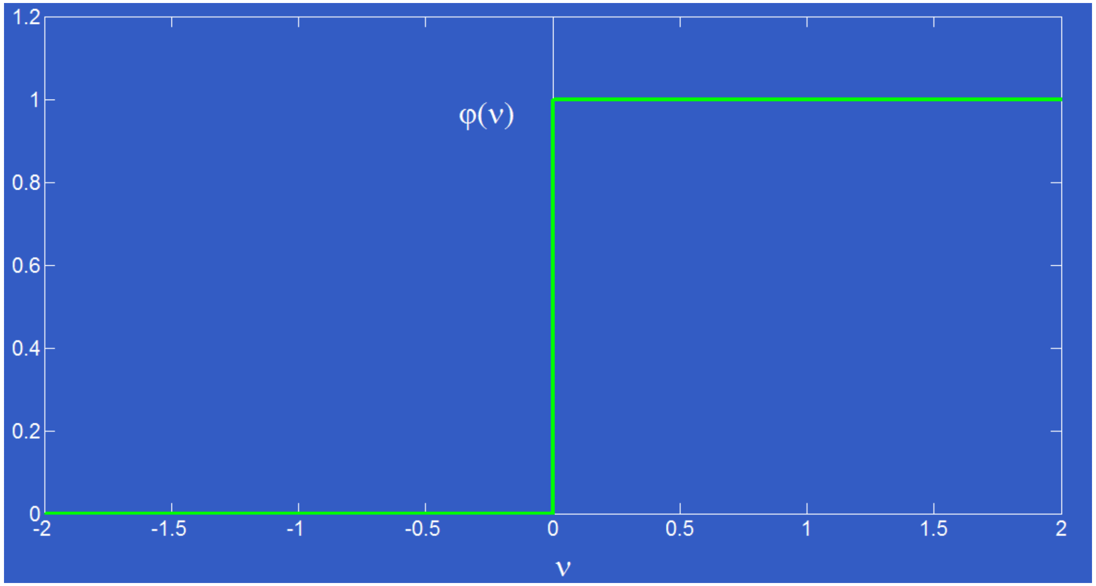


Figura 11: Função *threshold*.

$$\varphi(v) = \begin{cases} 1, se\ v \ \geq 0 \\ 0, se\ v < 0 \end{cases} \qquad (6)$$

Funções de Ativação

De forma correspondente, a saída do *k-ésimo* neurônio é dada por:

$$y_k = \begin{cases} 1, se\ v_k \geq 0 \\ 0, se\ v_k < 0 \end{cases} \qquad (7)$$

onde v_k é o campo local induzido do neurônio:

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

Funções de Ativação

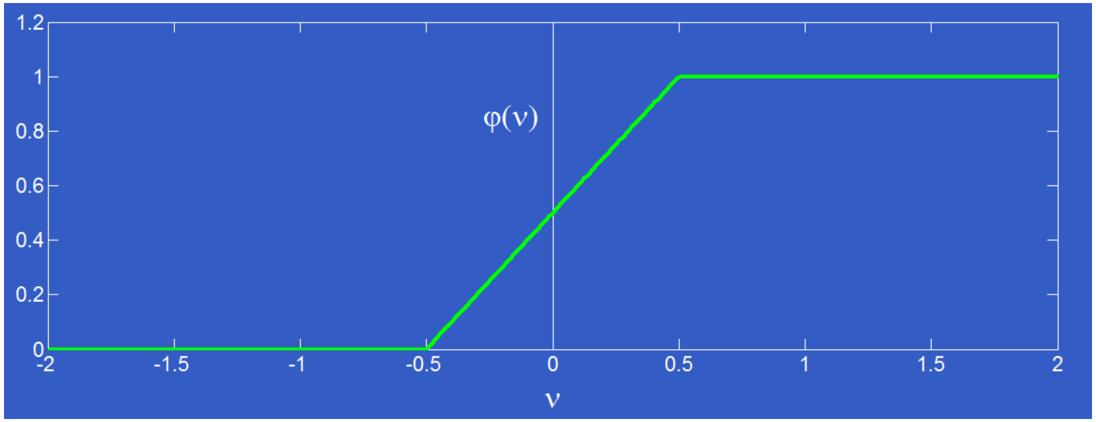


Figura 12: Função linear-por-partes.

$$\varphi(v) = \begin{cases} 1, se\ v \geq 0.5 \\ v, se\ -0.5 < v < +0.5 \\ 0, se\ v \leq -0.5 \end{cases} \quad (9)$$

Funções de Ativação

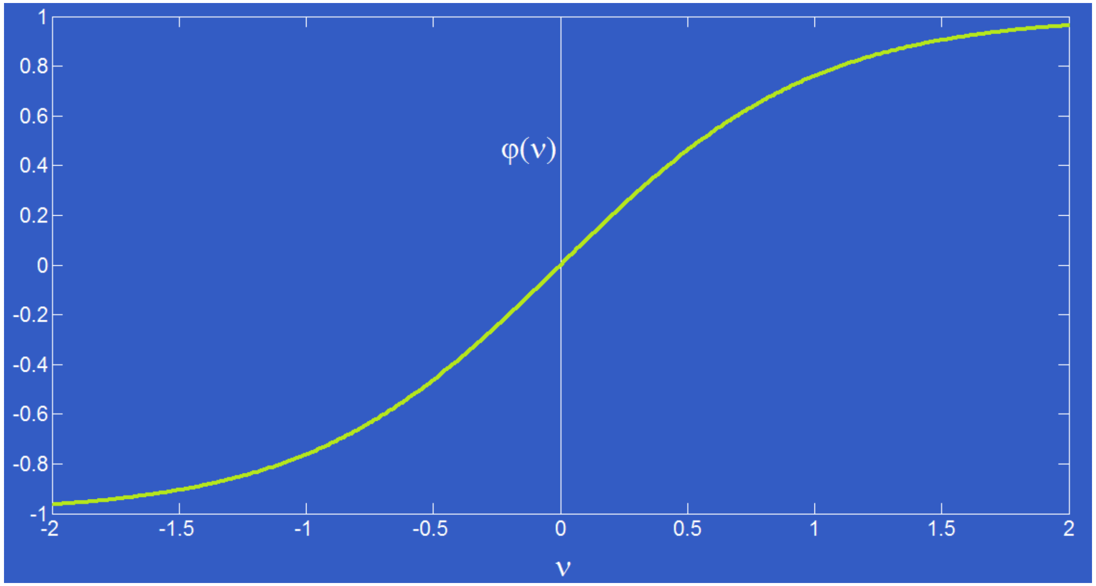


Figura 13: Função sigmoide – tangente hiperbólica.

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

Funções de Ativação



A função tangente hiperbólica:

- É a função mais comumente utilizada em redes neurais;
- É definida como uma função estritamente crescente que exibe um balanço interessante entre um comportamento linear e não linear;
- É uma função diferenciável.

Modelos Estocásticos

Em algumas aplicações, é desejável o funcionamento de um neurônio baseado em um modelo estocástico.

$$x = \begin{cases} +1.\, com\ probabilidade\ P(v) \\ -1,\, com\ probabilidade\ 1 - P(v) \end{cases} \tag{11}$$

onde x é o estado do neurônio. Uma escolha típica para P é:

$$P = \frac{1}{1 + e^{-\frac{v}{T}}} \tag{12}$$

Neurônios em Rede

Da mesma maneira que os neurônios do córtex cerebral são organizado em camadas, em geral, organizamos também as redes neurais artificiais em camadas.

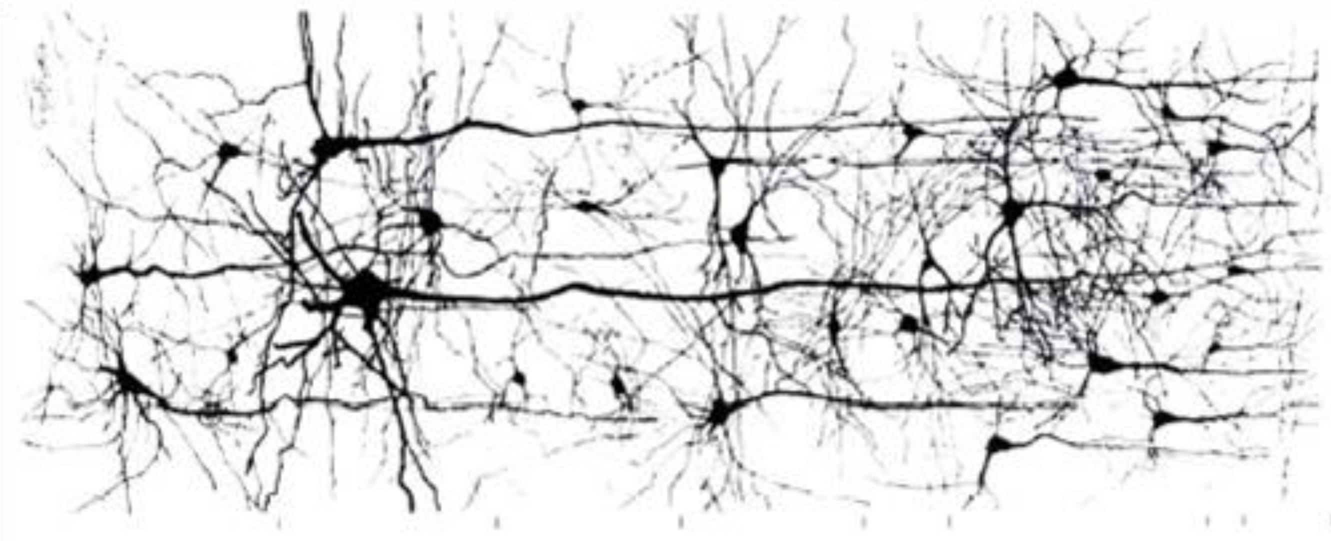


Figura 14: Organização em camadas do córtex cerebral.

Neurônios em Rede

- Arquiteturas típicas:
  - Rede de camada única
  - Redes de múltiplas camadas

- Treinamento:
  - Supervisionado
  - Não supervisionado
  - Pesos fixos

Neurônios em Rede

- Frequentemente é conveniente visualizar os neurônios em camadas;
- Neurônios da mesma camada tipicamente possuem o mesmo comportamento (função de ativação e padrão de conexões);
- O arranjo de neurônios em camadas e o padrão de interconexão é conhecido como arquitetura da rede;
- Rede neurais também podem ser classificadas como redes *feedfoward* ou recorrentes.

Redes de Camada Única



- Possuem uma única camada de conexões com pesos;
- Para classificações de padrões, cada neurônio de saída corresponde a uma classe em particular a qual um vetor de entrada pode pertencer ou não;
- Com uma pequena alteração no modelo do neurônio, elas podem ser utilizadas em problemas de segmentação.

Redes de Camada Única

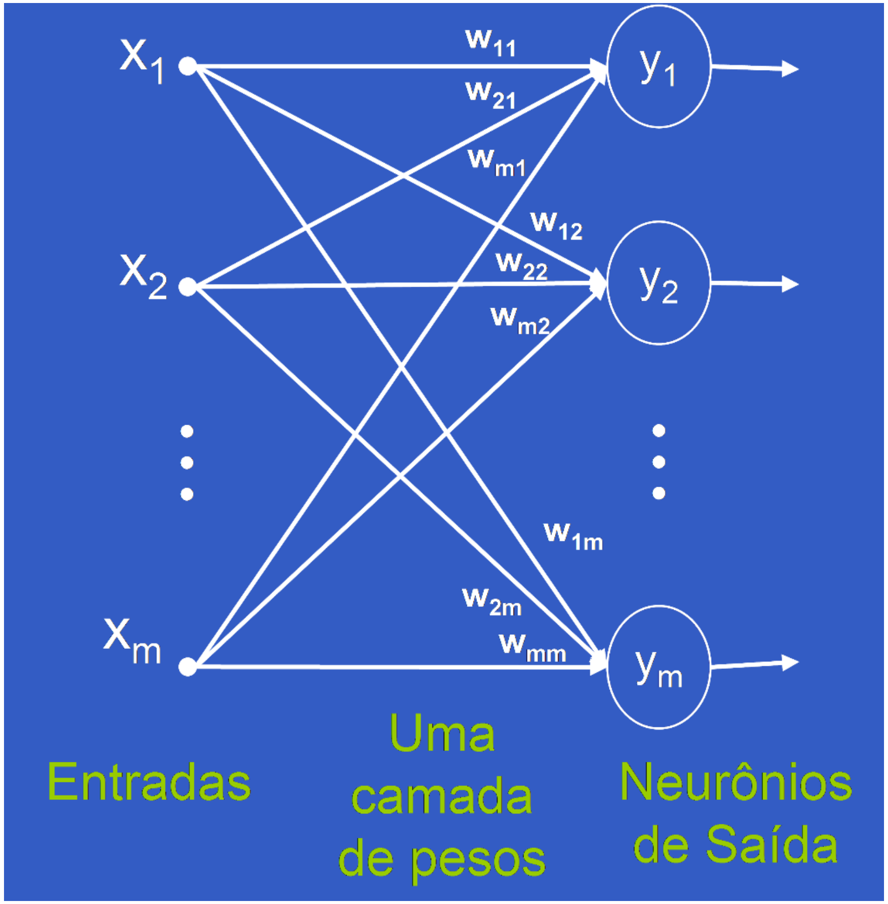


Figura 15: Rede de camada única.

Redes de Camadas Múltiplas

- Possuem uma ou mais camadas de neurônios escondidos entre as unidades de entrada e as unidades de saída;
- Em geral, redes de múltiplas camadas podem resolver problemas mais complexos que as redes de camada única, mas o treinamento pode ser bem mais demorado.

Redes de Camadas Múltiplas

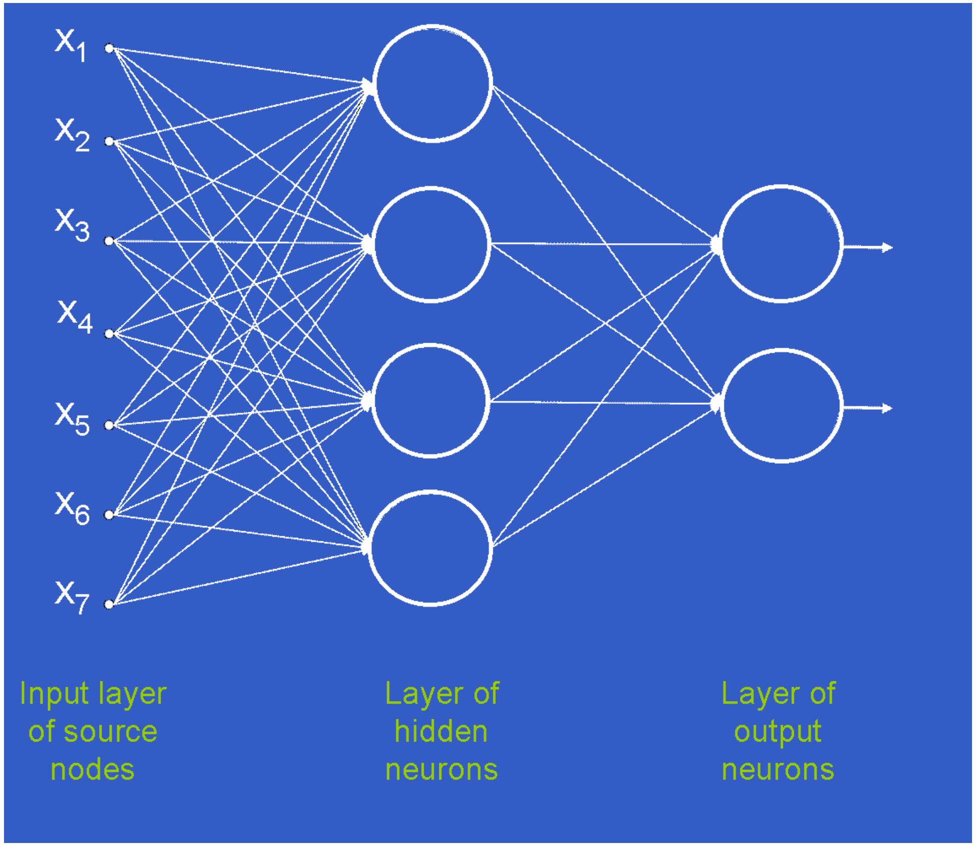


Figura 15: Rede de múltiplas camadas.

Redes *Feedfoward* x Recorrentes



- Redes feedfoward
  - Sem realimentações
  - Estáticas
  - Estruturalmente estáveis

- Redes recorrentes
  - Com realimentações
  - Dinâmica
  - Instabilidade

Redes *Feedfoward* x Recorrentes

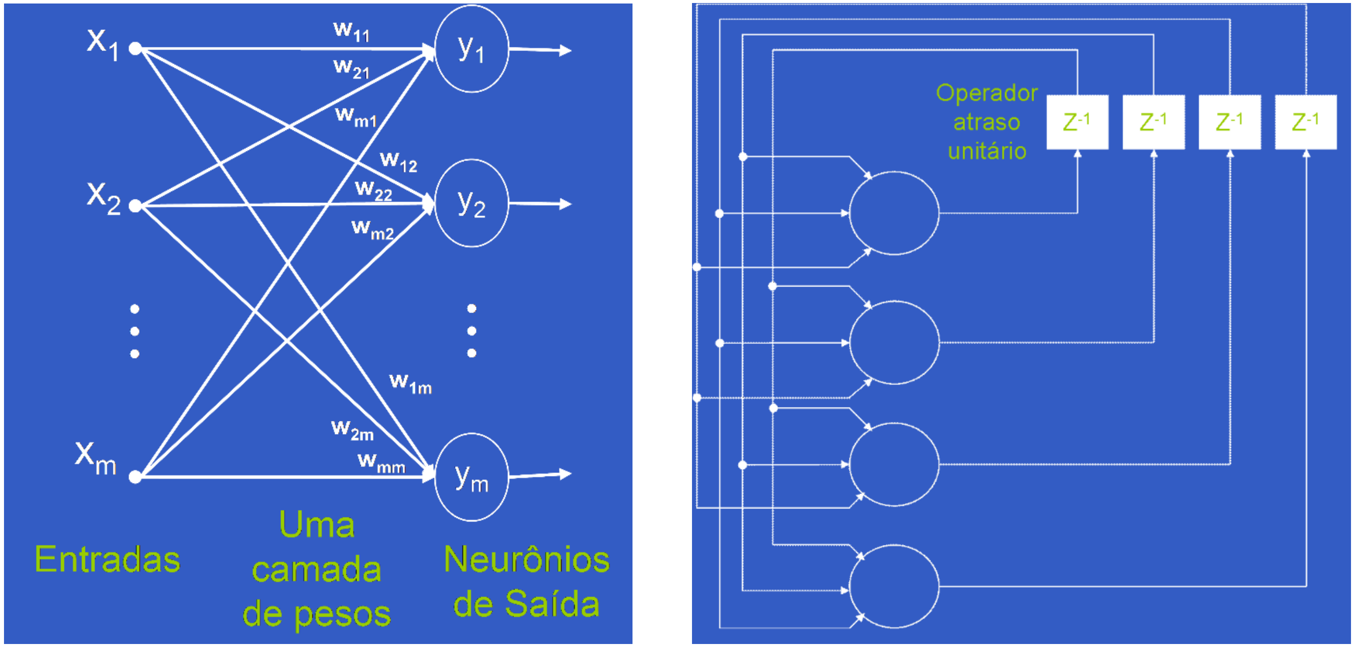


Figura 17: Redes *feedfoward* x Recorrentes.

Treinamento Supervisionado

- Geralmente a base de dados disponível é dividida em dois conjuntos: O *conjunto de treinamento* e o *conjunto de testes*;
- A base de dados é formado por tuplas, ou seja, pares de dados de “entrada” e “saída” (a base é dita rotulada);
- Apresenta-se uma sequência de vetores de treinamento, ou padrões, cada um associado a um determinado valor de saída.

Treinamento Supervisionado

- Os pesos das sinapses são ajustados de acordo com um algoritmo de aprendizado;
- Utilizado nos problemas de *classificação de padrões* e *associação de padrões* (memória auto-associativa e hetero-associativa).

Treinamento Não Supervisionado

- Uma sequência de vetores de treinamento (entrada) é apresentada à rede, mas sem apresenta a respectiva saída associada;
- Os pesos das sinapses são ajustados de forma que os vetores de entrada mais “similares” sejam agrupados sob o mesmo rótulo (*cluster* ou classe);
- A rede produzirá um vetor exemplar (representativo) para cada *cluster* formado.

Quando uma base é composta por “entradas”, mas as “saídas” associadas não existem, então a base é dita não rotulada.

Treinamento Não Supervisionado

- O número de clusters pode ser conhecido previamente ou não.
- É uma técnica bastante utilizada em problemas de *clusterização* e *data mining*.

Pesos Fixos



- Geralmente empregada em problemas de otimização com restrições;
- Tais redes podem trabalhar muito bem em problemas que causam dificuldades para técnicas tradicionais, tais como restrições conflitantes (nem todas as restrições podem ser satisfeitas simultaneamente);
- Os pesos das sinapses são ajustados de forma a representar as restrições e a quantidade a ser *maximizada* ou *minimizada*;

Pesos Fixos

- São exemplos: A Máquina de *Boltzmann* (sem aprendizado) e a Rede de *Hopfield*.

Referências



1. http://www.scikit-learn.org

2. https://www.tensorflow.org

3. Theodoris, S., Koutroumbas, K., "*Pattern Recognition*", 4th. Edition, Academic Press