Aula 07 – Redes Neurais

1001524 – Aprendizado de Máquina I
2023/1 - Turmas A, B e C
Prof. Dr. Murilo Naldi

naldi@ufscar.br

Agradecimentos

- Parte do material utilizado nesta aula foi cedido pelos professores André Carvalho, Ricardo Campello, Diego Silva e Alan Valejo
- Parte do material utilizado nesta aula foi disponibilizado por M. Kumar no endereço:
  - www-users.cs.umn.edu/~kumar/dmbook/index.php
- Agradecimentos a Intel Software e a

  Intel IA Academy pelo material

  disponibilizado e recursos didáticos

Redes Neurais

- Origens
  - Inspiração biológica
    - Sistema visual humano
      - Reconhece rosto familiar em ambiente estranho
    - Sonar de morcegos
      - Reconhece alvos e barreiras a distância e velocidade

“I do not see why [the computer] should not enter any one of the fields normally covered by the human intellect, and eventually compete on equal terms.”

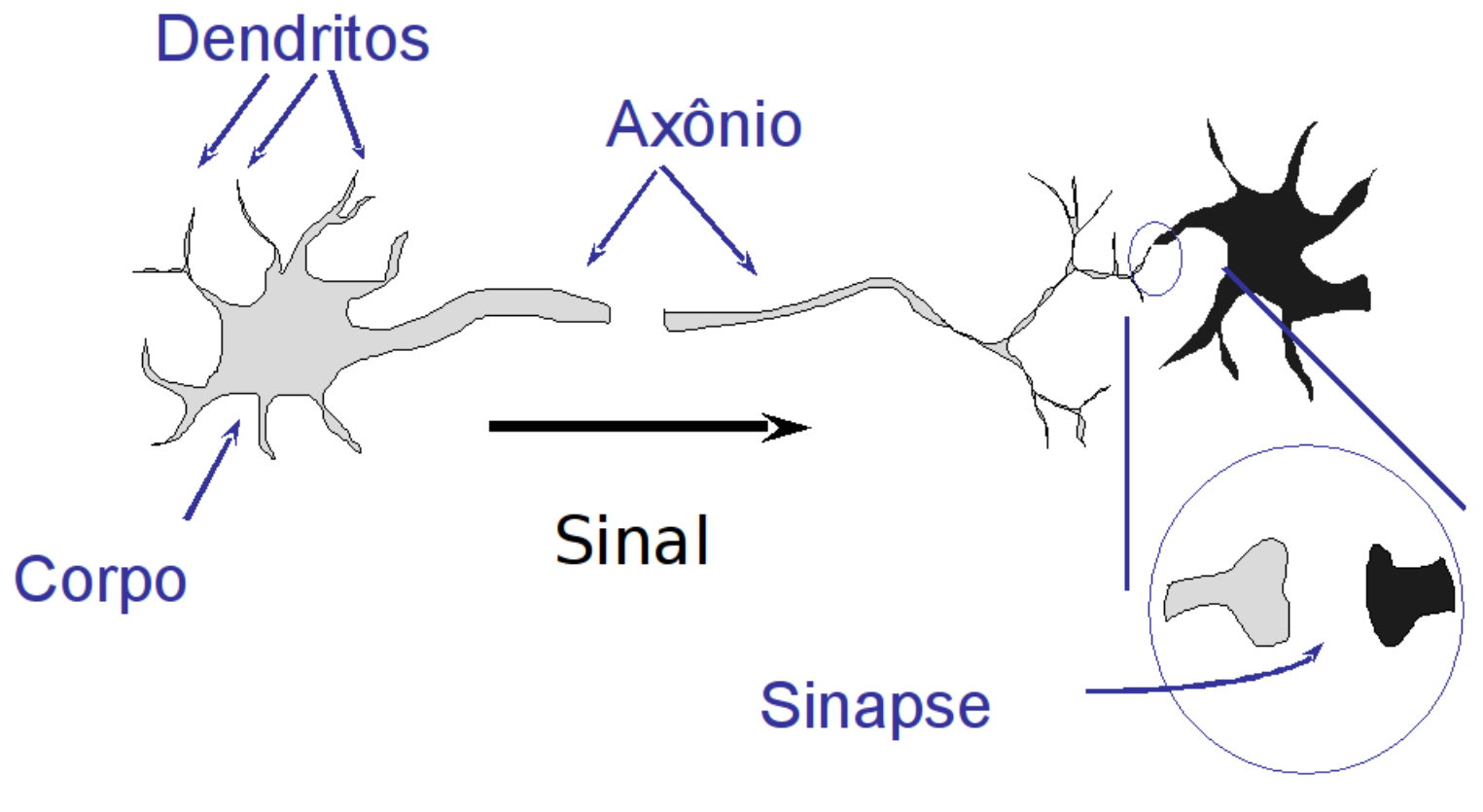
Alan Turing (1949)

Redes Neurais

- Uma rede neural é um processador massivamente distribuído e paralelo feito de unidades de processamento simples:
  - Neurônios
- Armazena conhecimento experimental
  - Utilizado para induzir um modelo

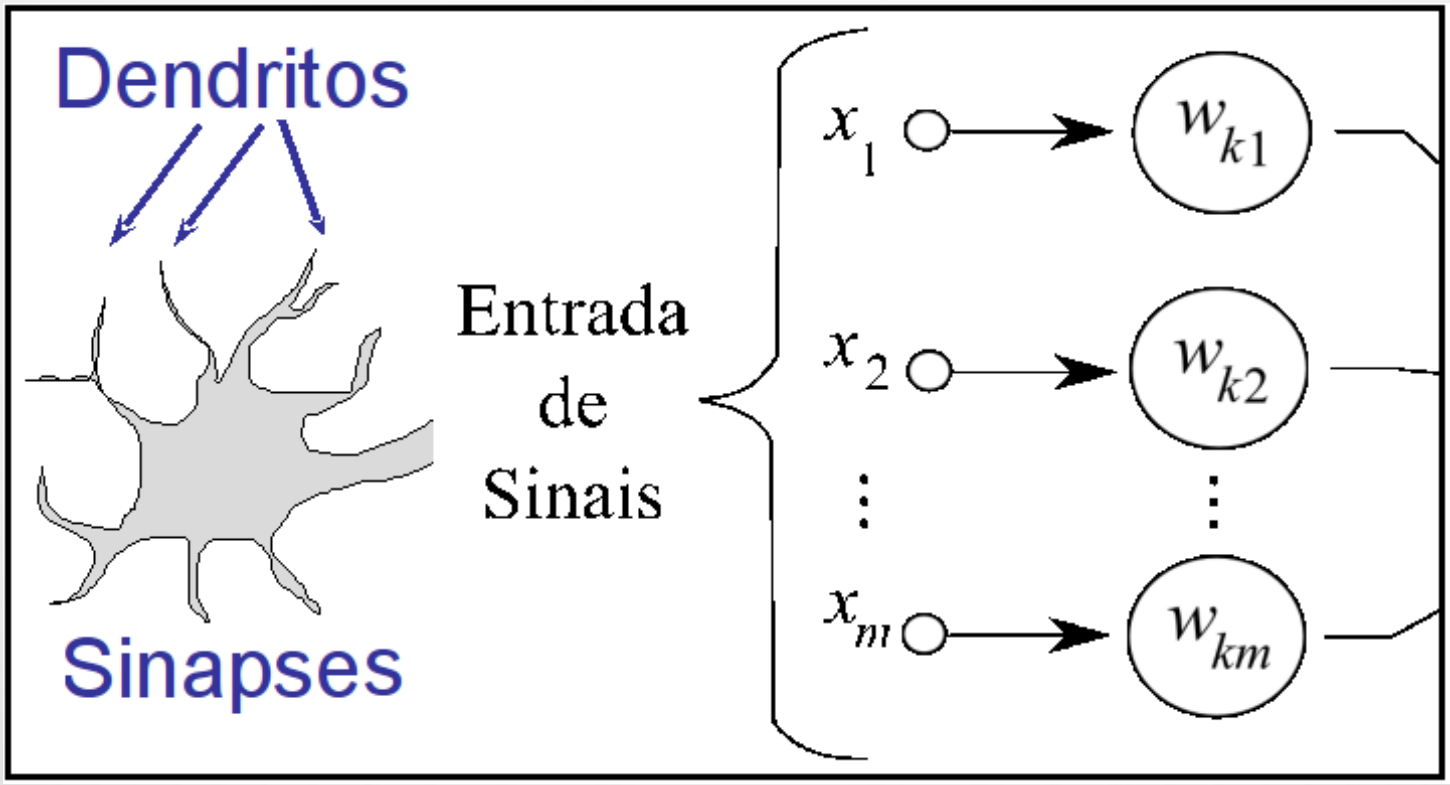
Neurônio

- Biológico



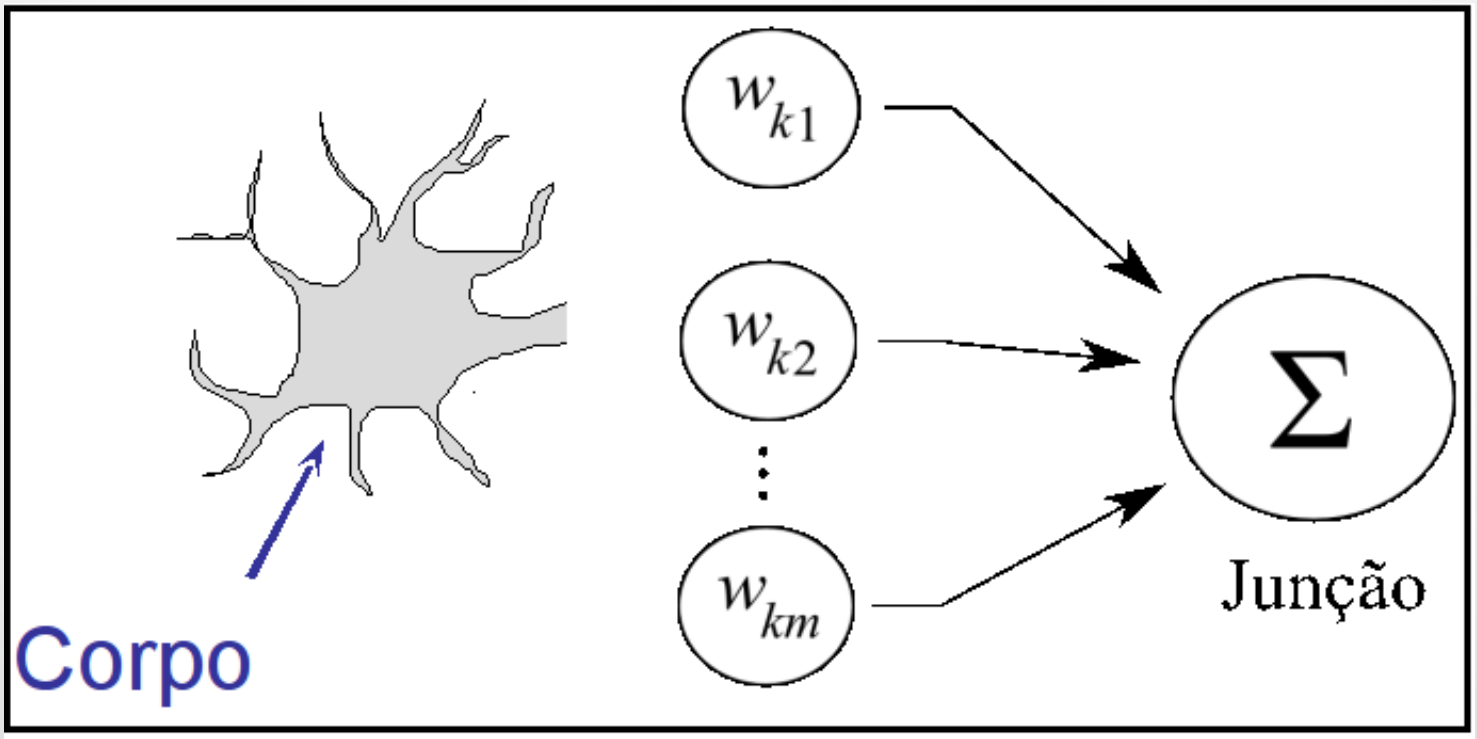
Conjunto de Sinapses

- Cada sinapse é caracterizada por um peso



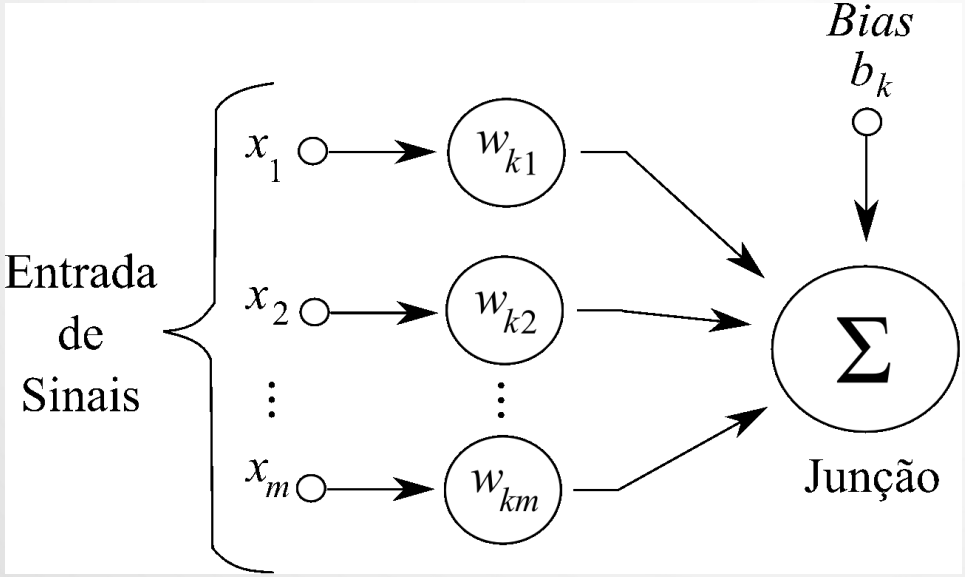
Junção de Sinais

- Somatório dos sinais multiplicados pelos pesos das Sinapses



Bias

- *Bias* é utilizados para gerar transformação afim no potencial de ativação do neurônio

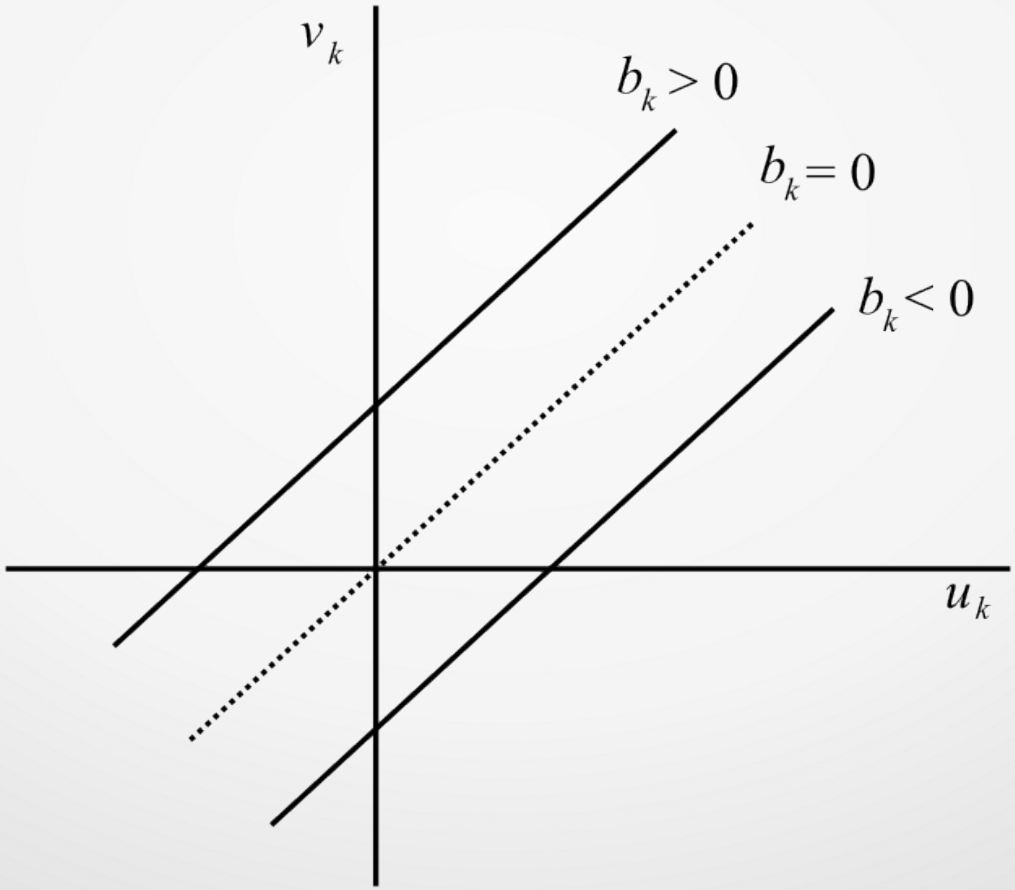


$$u_k = \sum_{j=1}^{m} w_{kj} x_j$$

$$v_k = u_k + b_k$$

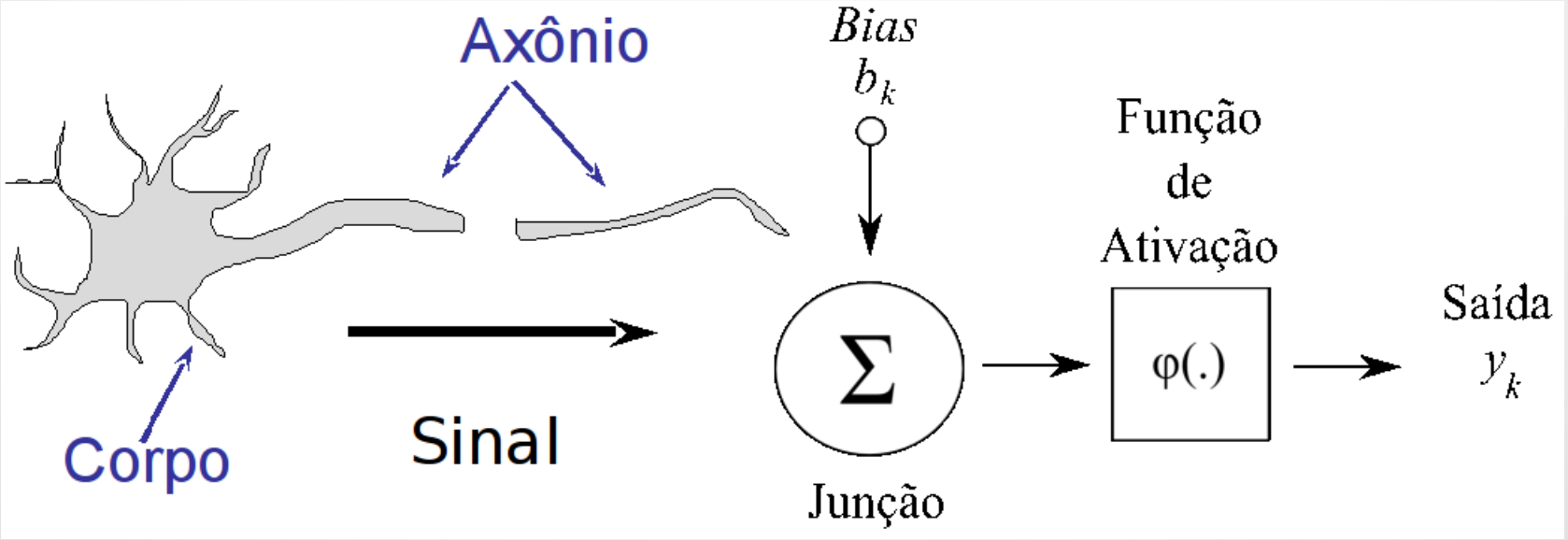
Como *bias* funciona?

- Combinação linear (u_k) + Translação (b_k)



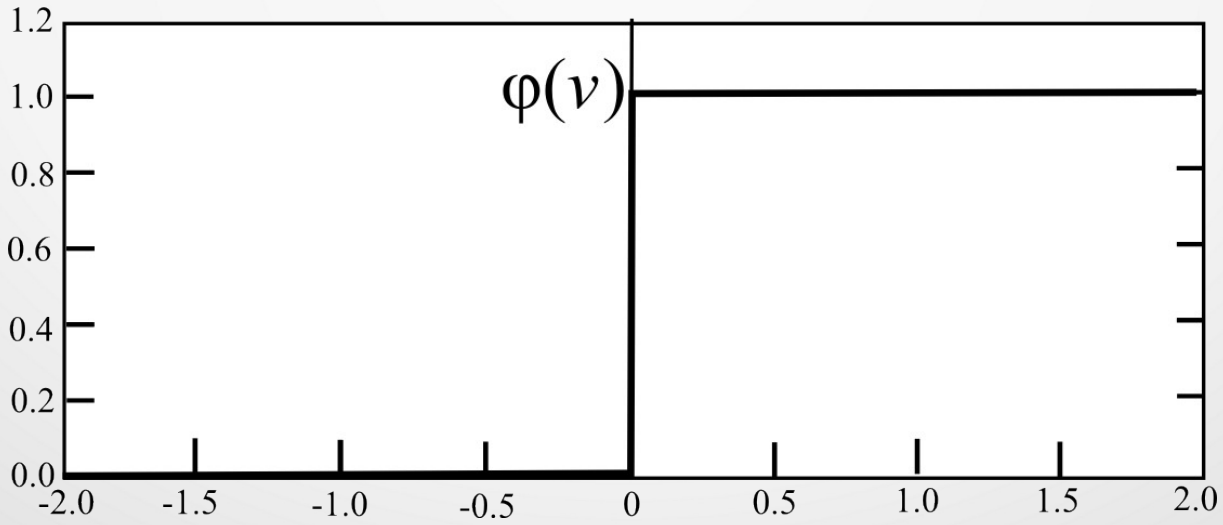
Função de ativação

- Determina se o potencial de ativação (v_k) é suficiente para ativar o neurônio



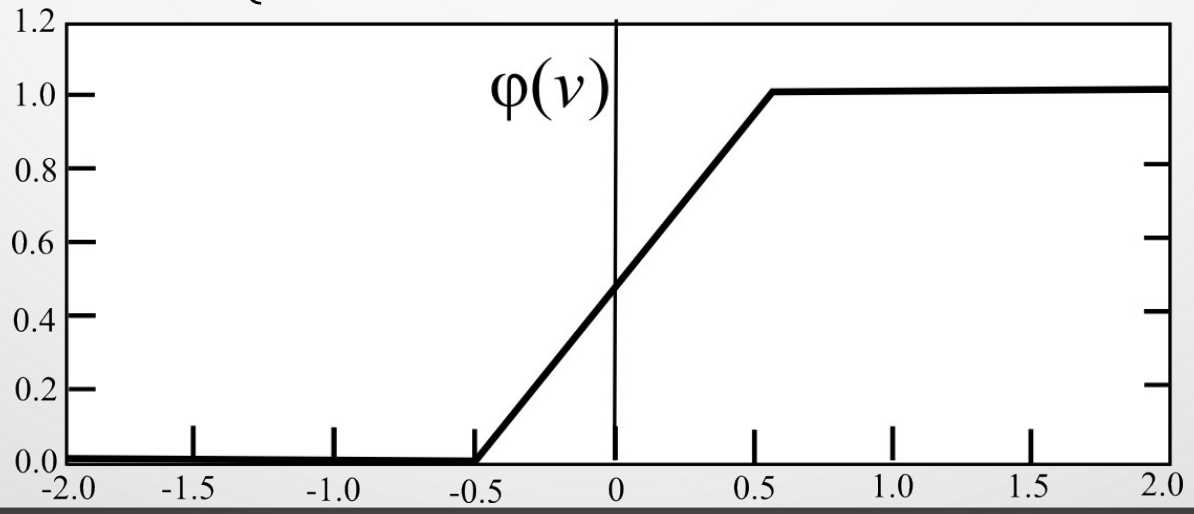
Limiar

$$\varphi(v_k) = \begin{cases} y_k = 1 & se \quad v_k \geq 0 \\ y_k = 0 & se \quad v_k < 0 \end{cases}$$



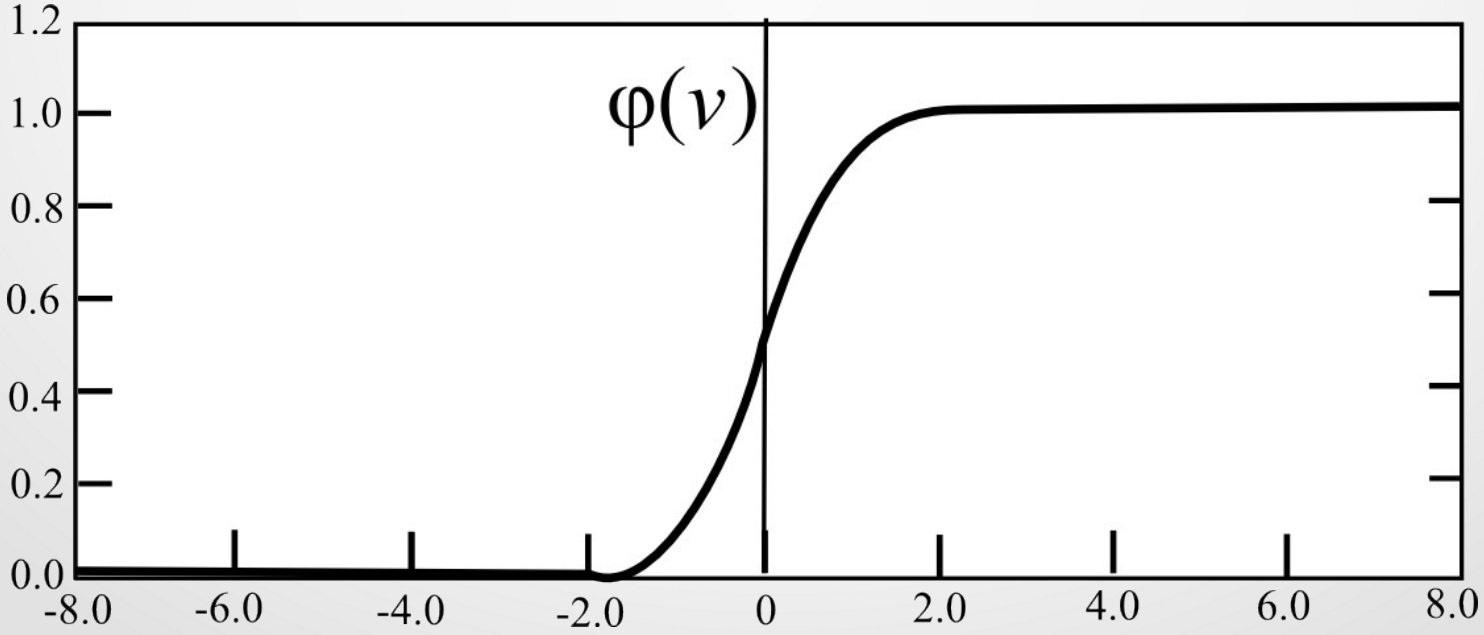
Linear

$$\varphi(v_k) = \begin{cases} y_k = 1 & se \quad v_k \geq \frac{1}{2} \\ y_k = v_k & se \quad \frac{1}{2} > v_k > -\frac{1}{2} \\ y_k = 0 & se \quad v_k \leq -\frac{1}{2} \end{cases}$$



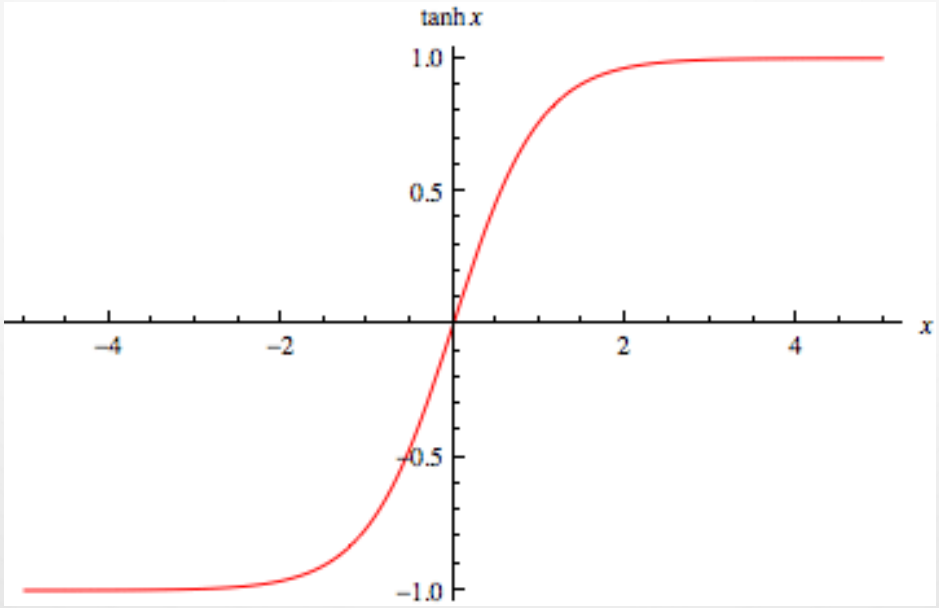
Sigmoidal

$$\varphi(v_k) = \frac{1}{1 + \exp(-av_k)}$$



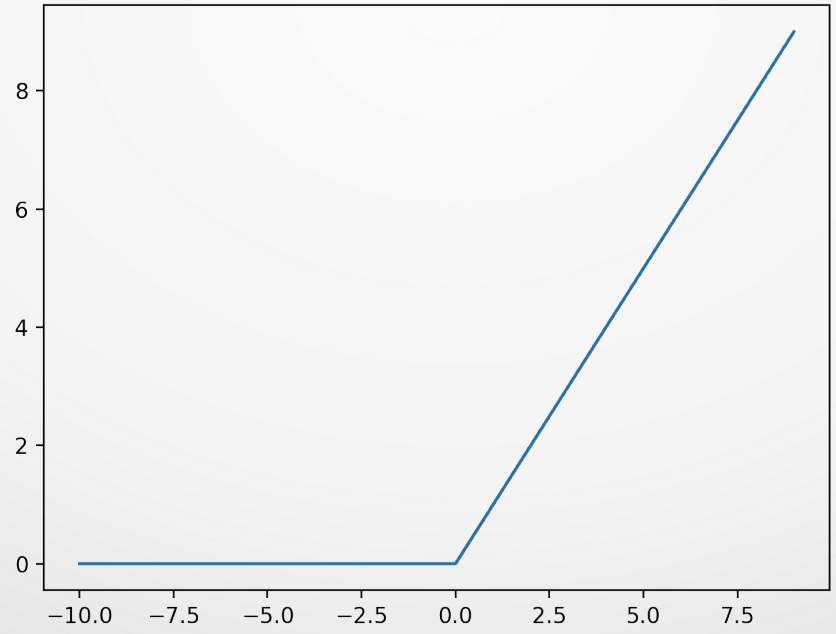
Tangente Hiperbólica

$$\tanh z = \frac{\sinh z}{\cosh z}$$



Rectified Linear Activation Function (Relu)

$$f(x) = x^+ = \max(0, x) = \frac{x + |x|}{2} = \begin{cases} x & \text{if } x > 0, \\ 0 & \text{otherwise.} \end{cases}$$

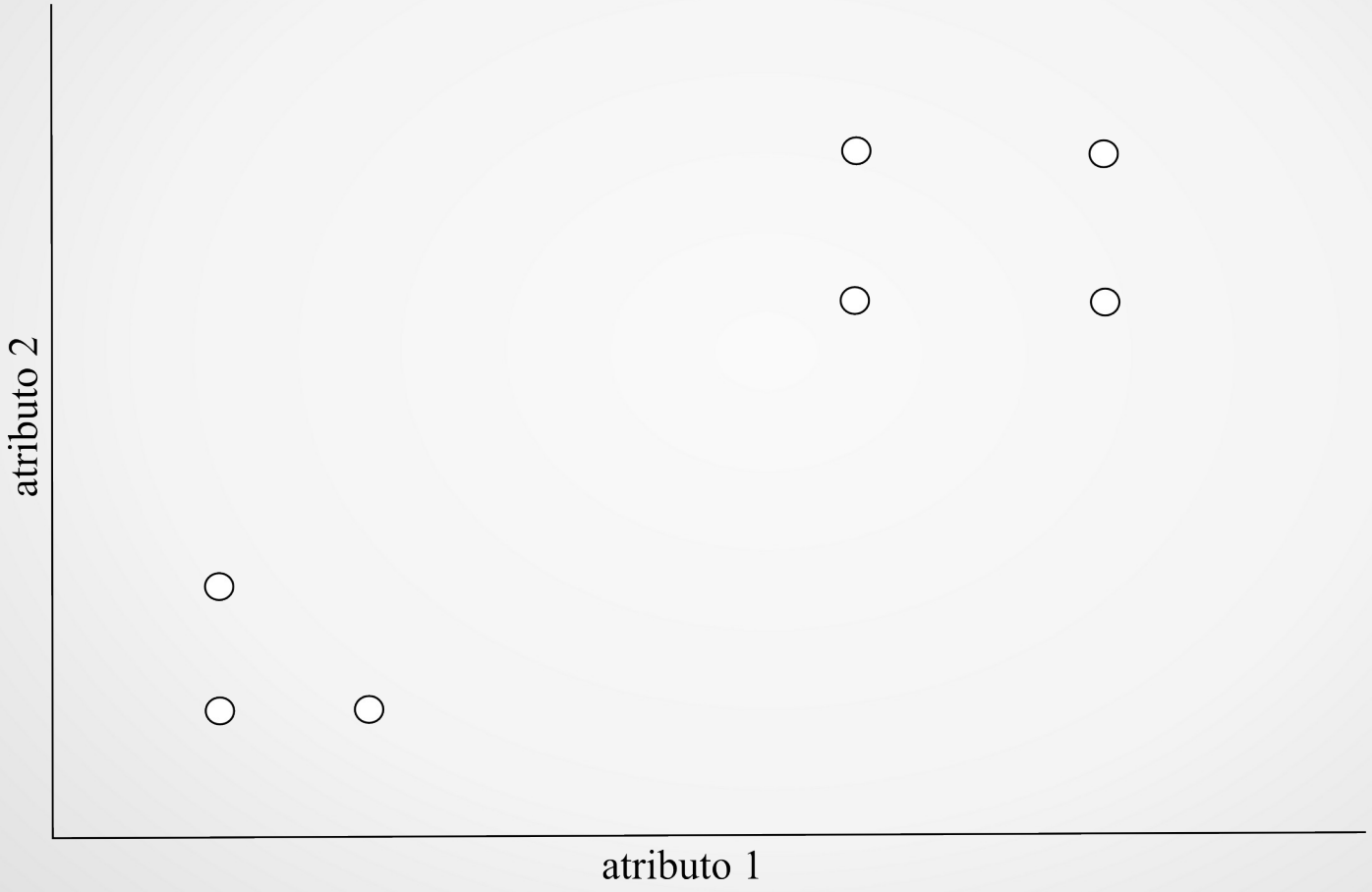


Exemplo

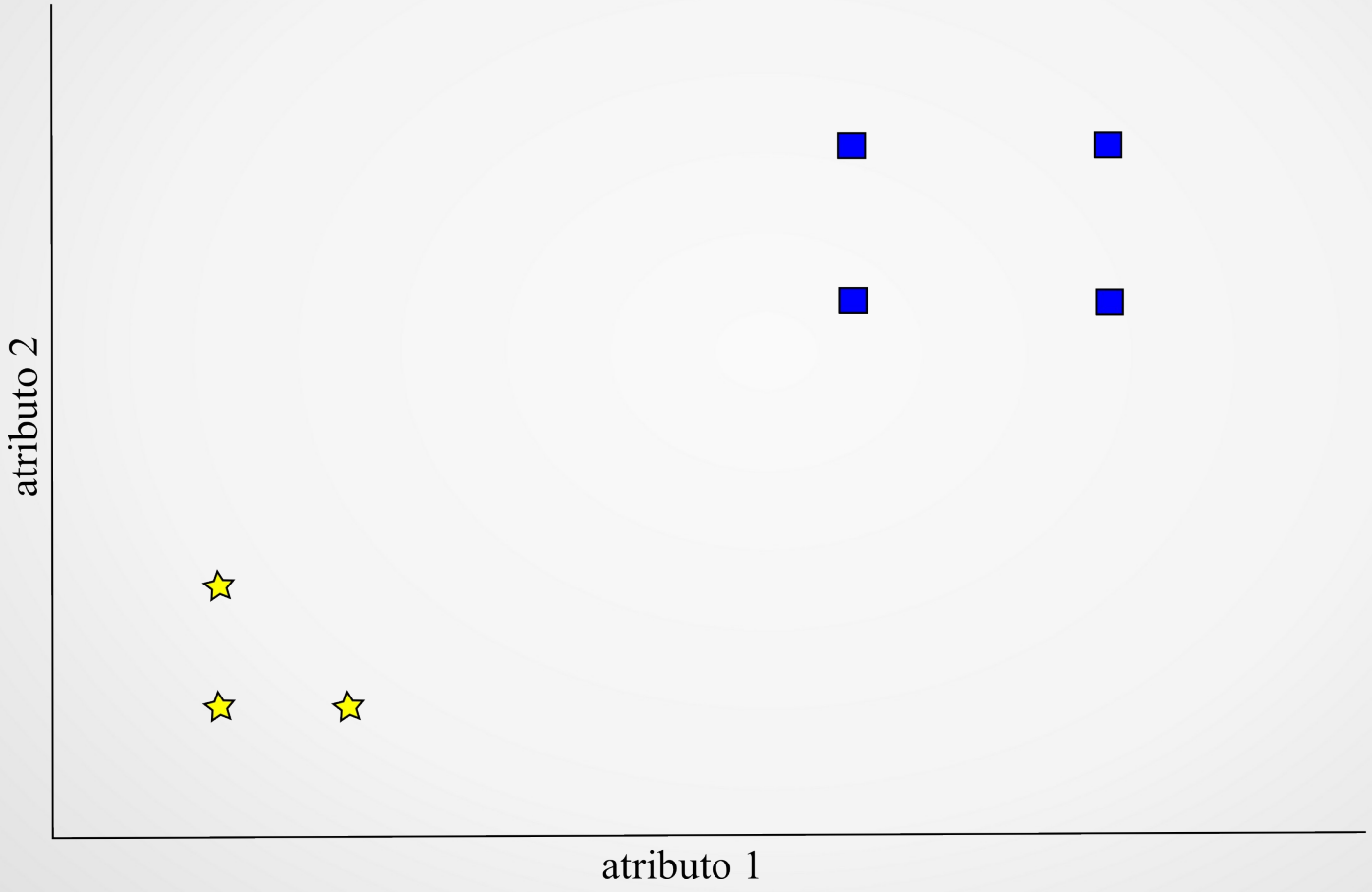
- Considere o conjunto de dados a seguir:

	Atributo 1	Atributo 2
Amostra Minério 1	0.3	0.2
Amostra Minério 2	0.2	0.3
Amostra Minério 3	0.5	0.7
Amostra Minério 4	0.6	0.6
Amostra Minério 5	0.6	0.7
Amostra Minério 6	0.2	0.2
Amostra Minério 7	0.5	0.6

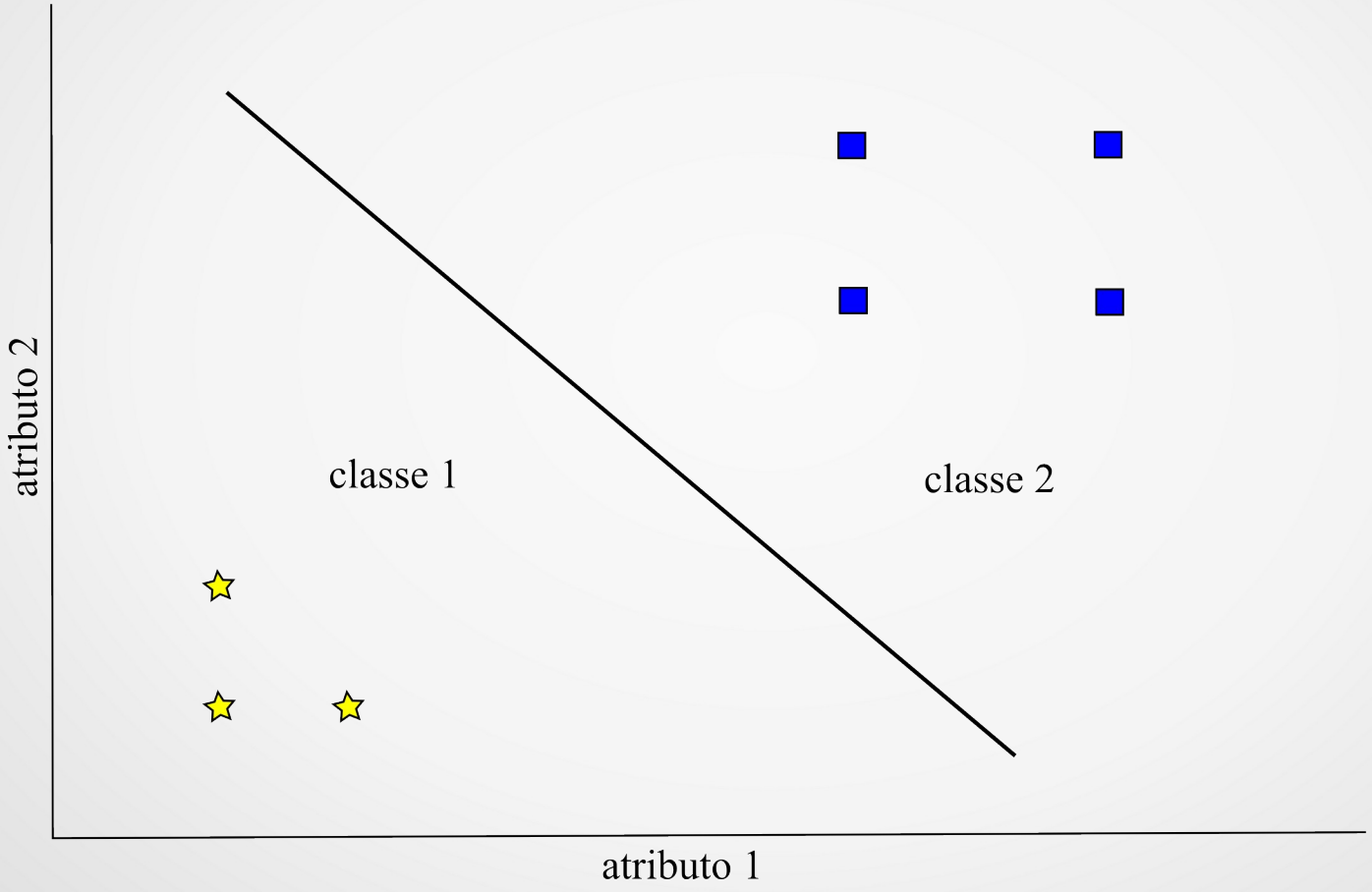
Exemplo – dados gráfico



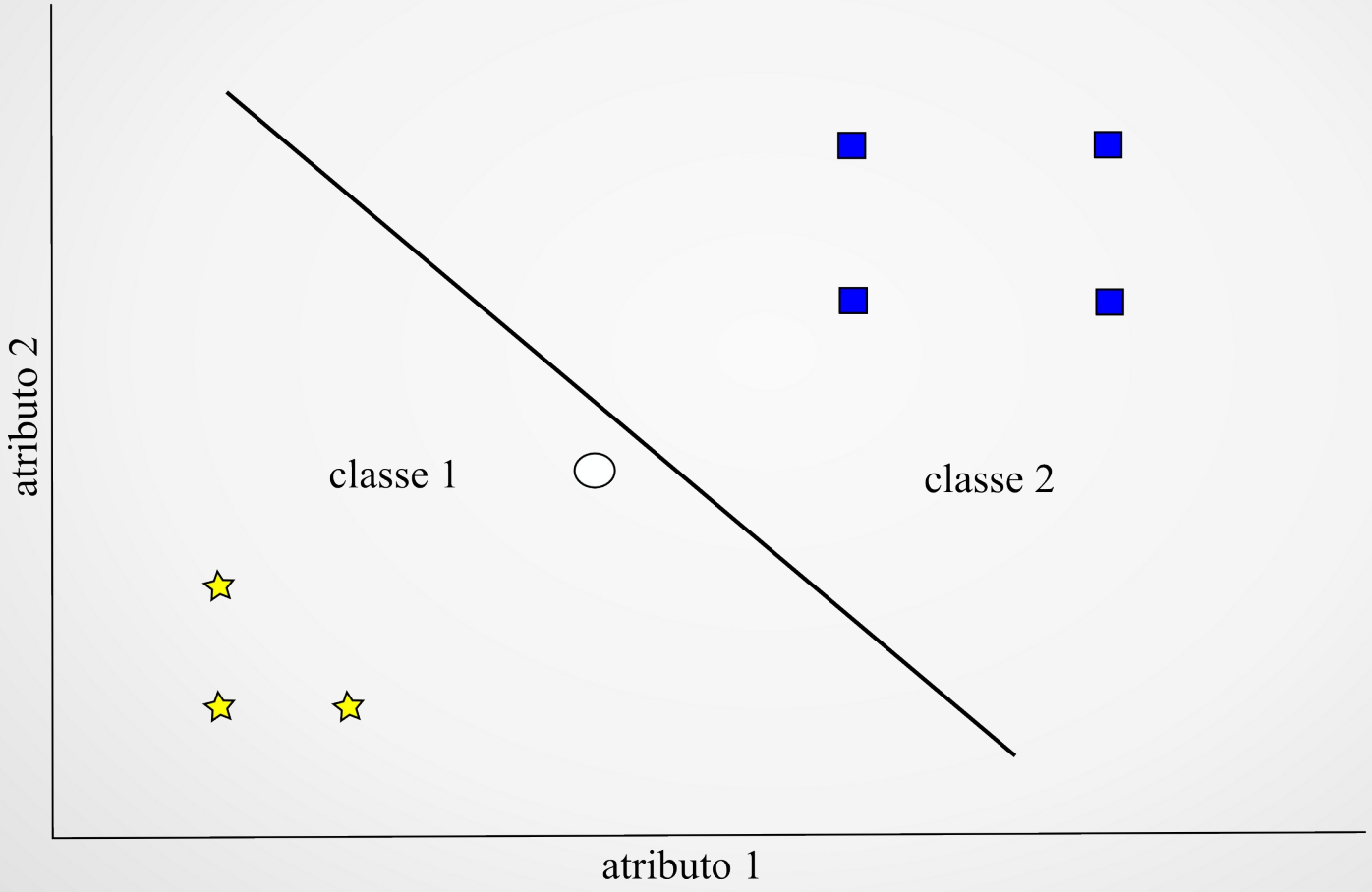
Exemplo - classes



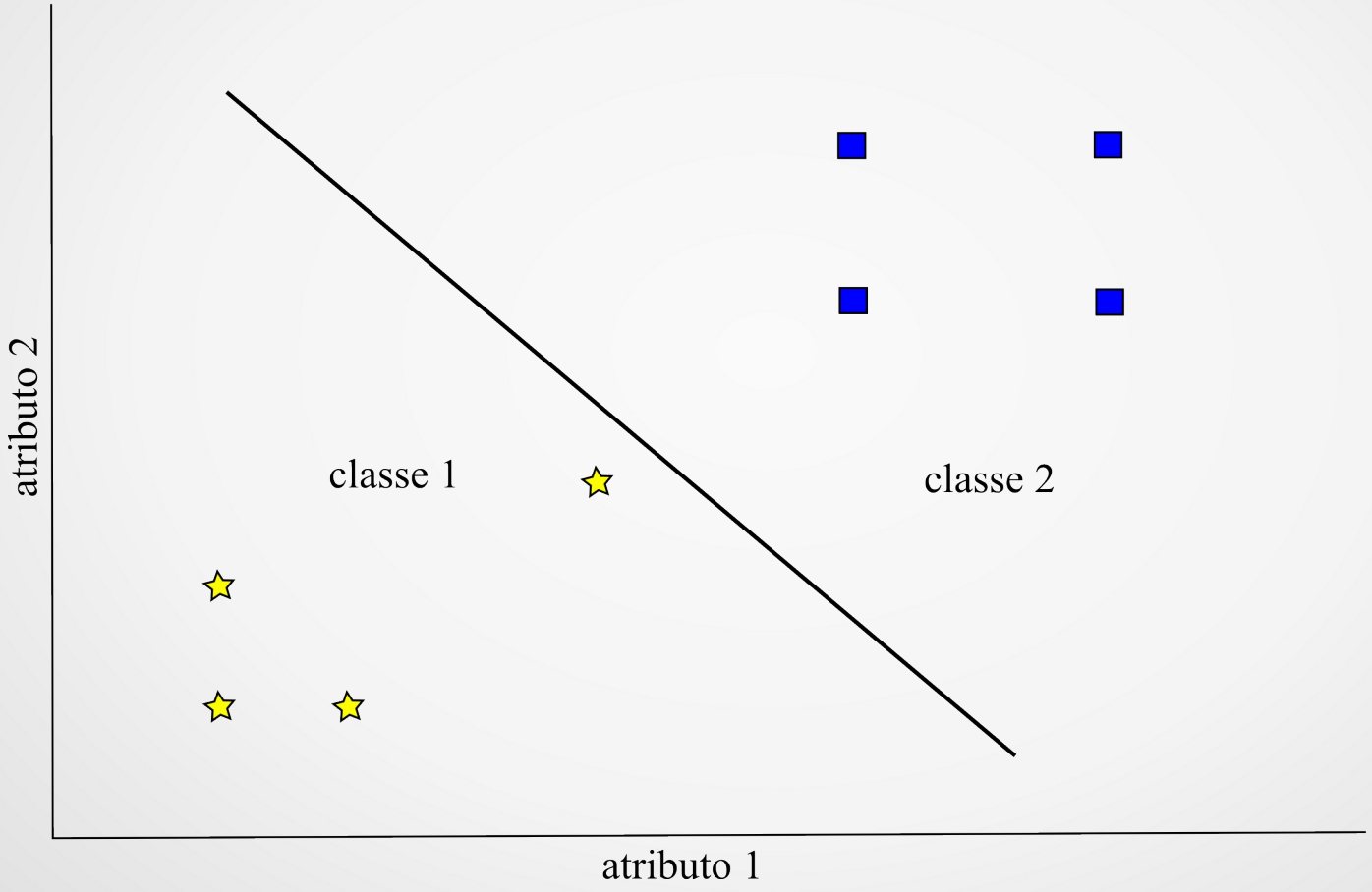
Exemplo - classificador



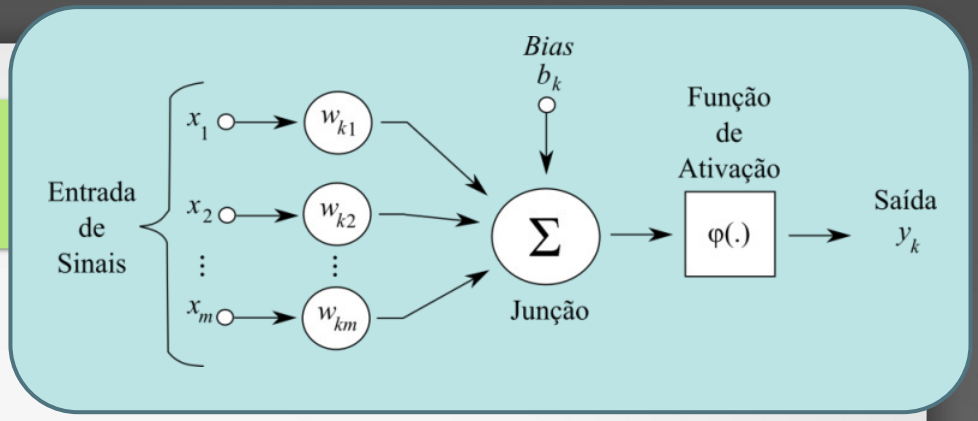
Exemplo - classificação



Exemplo - resultado



Exercício

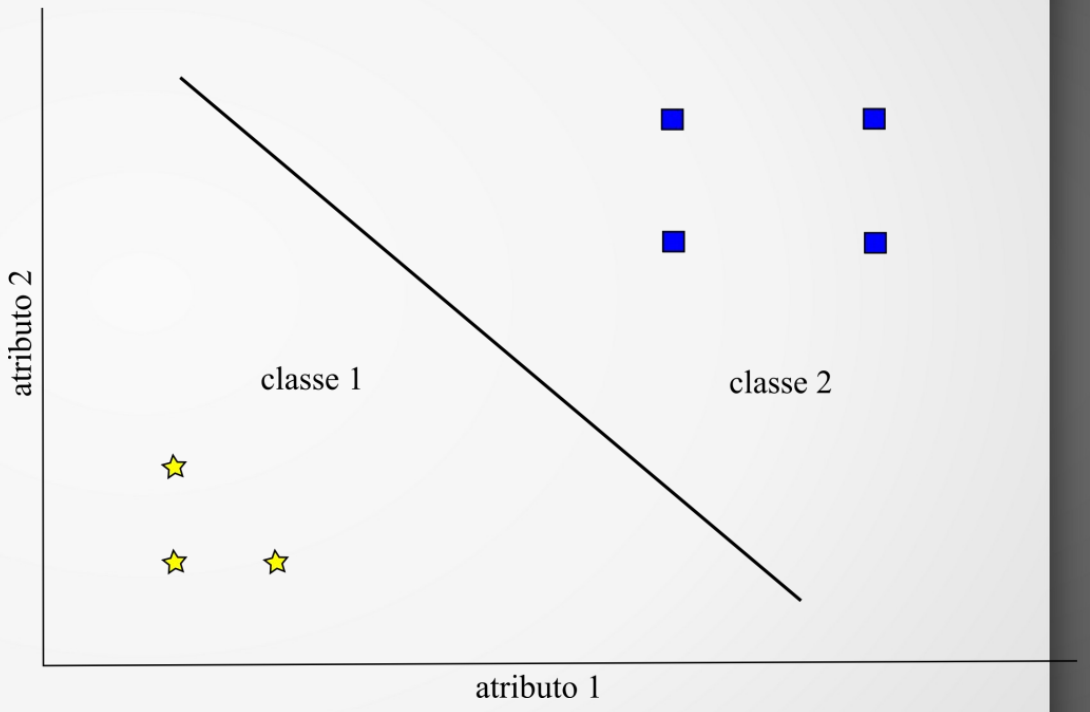


- Utilize a rede neural:
  - $w_{k1} = -1$, $w_{k2} = -1$, $b_k = 0.6$
  - ativação limiar

	Atributo 1	Atributo 2
Amostra Minério 1	0.3	0.2
Amostra Minério 2	0.2	0.3
Amostra Minério 3	0.5	0.7
Amostra Minério 4	0.6	0.6
Amostra Minério 5	0.6	0.7
Amostra Minério 6	0.2	0.2
Amostra Minério 7	0.5	0.6

Resultado

	classe
Amostra Minério 1	1
Amostra Minério 2	1
Amostra Minério 3	2
Amostra Minério 4	2
Amostra Minério 5	2
Amostra Minério 6	1
Amostra Minério 7	2



Treinamento

- Perceptron
  - Conjunto de dados é utilizado para treinar
  - Cada objeto é apresentado a rede
  - Calcula-se o erro:

$$e_k(n) = d_k(n) - y_k(n)$$

Treinamento

- Atualiza os pesos:

$$\Delta w_{kj}(n) = \eta e_k(n) x_j(n)$$

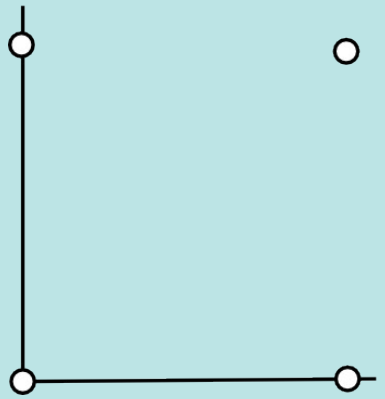
$$w_{kj}(n+1) = w_{kj}(n) + \Delta w_{kj}(n)$$

- Se os dados são linearmente separáveis, o algoritmo converge

Exemplo

- Treinar um perceptron para executar AND com função de ativação limiar

Entrada 1	Entrada 2	Saída
0	0	0
1	0	0
0	1	0
1	1	1



$$w_1 = 0.6$$

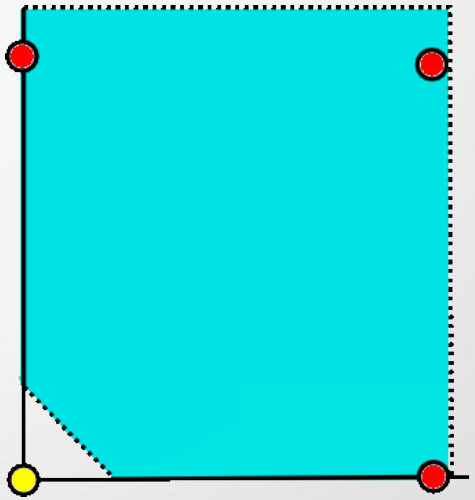
$$w_2 = 0.6$$

$$\eta = 0.2$$

$$b = -0.3$$

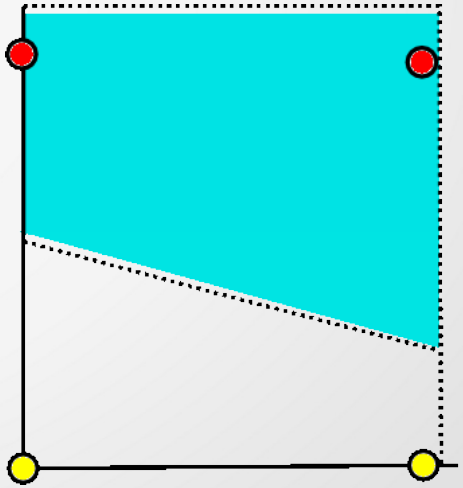
Exemplo

- Entrada: $x = [0,0]$
- Teste:
  - $0 * 0.6 + 0 * 0.6 - 0.3 = -0.3 => \varphi(-0.3) => y = 0$
  - $d = 0 => e = 0 - 0 = 0$
- Como *e*=0, não há correção



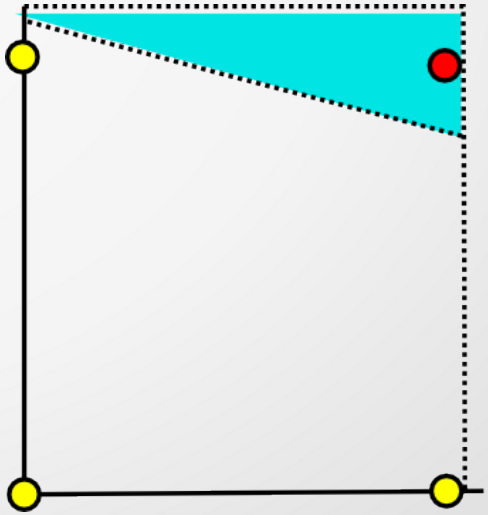
Exemplo

- Entrada: $x = [1,0]$
- Teste:
  - $1 * 0.6 + 0 * 0.6 - 0.3 = 0.3 => \varphi(0.3) => y = 1$
  - $d = 0 => e = 0 - 1 = -1$
- Atualiza pesos
  - $w_{1(n+1)} = 0.6 + 0,2 * -1 * 1 = 0,4$
  - $w_{2(n+1)} = 0.6 + 0,2 * -1 * 0 = 0,6$
  - $b_{(n+1)} = -0,3 + 0,2 * -1 * 1 = -0,5$



Exemplo

- Entrada: $x = [0,1]$
- Teste:
  - $0 * 0.4 + 1 * 0.6 - 0.5 = 0.1 => \varphi(0.1) => y = 1$
  - $d = 0 => e = 0 - 1 = -1$
- Atualiza pesos
  - $w_{1(n+1)} = 0.4 + 0,2 * -1 * 0 = 0,4$
  - $w_{2(n+1)} = 0.6 + 0,2 * -1 * 1 = 0,4$
  - $b_{(n+1)} = -0,5 + 0,2 * -1 * 1 = -0,7$

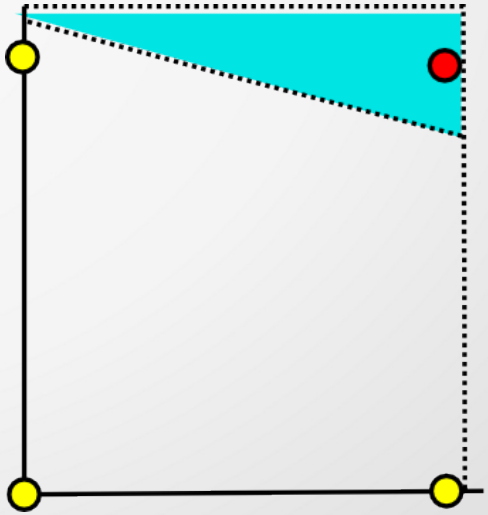


Exemplo

- Entrada: $x = [1,1]$
- Teste:
  - $1 * 0.4 + 1 * 0.4 - 0.7 = 0.1 => \varphi(0.1) => y = 1$
  - $d = 1 => e = 1 - 1 = 0$

- Como e=0, não há correção

- O perceptron convergiu



Exercícios

- Treinar perceptrons para executar:
  - as funções OR e NAND
- Induza um perceptron para classificar as classes de minério 1 e 2 abaixo

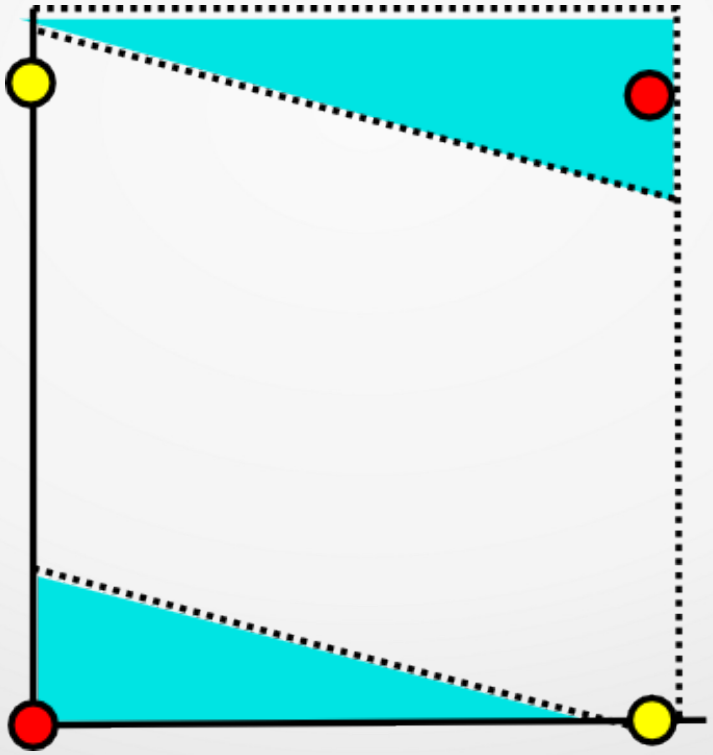
	Atributo 1	Atributo 2	Classe
Amostra 1	0,1	0,5	1
Amostra 2	0,7	0,2	2
Amostra 3	0,2	0,6	1

- A qual classe pertence a amostra 4?

	Atributo 1	Atributo 2	Classe
Amostra 4	0,2	0,4	?

Problema

- Como treinar uma rede para executar a função XOR?



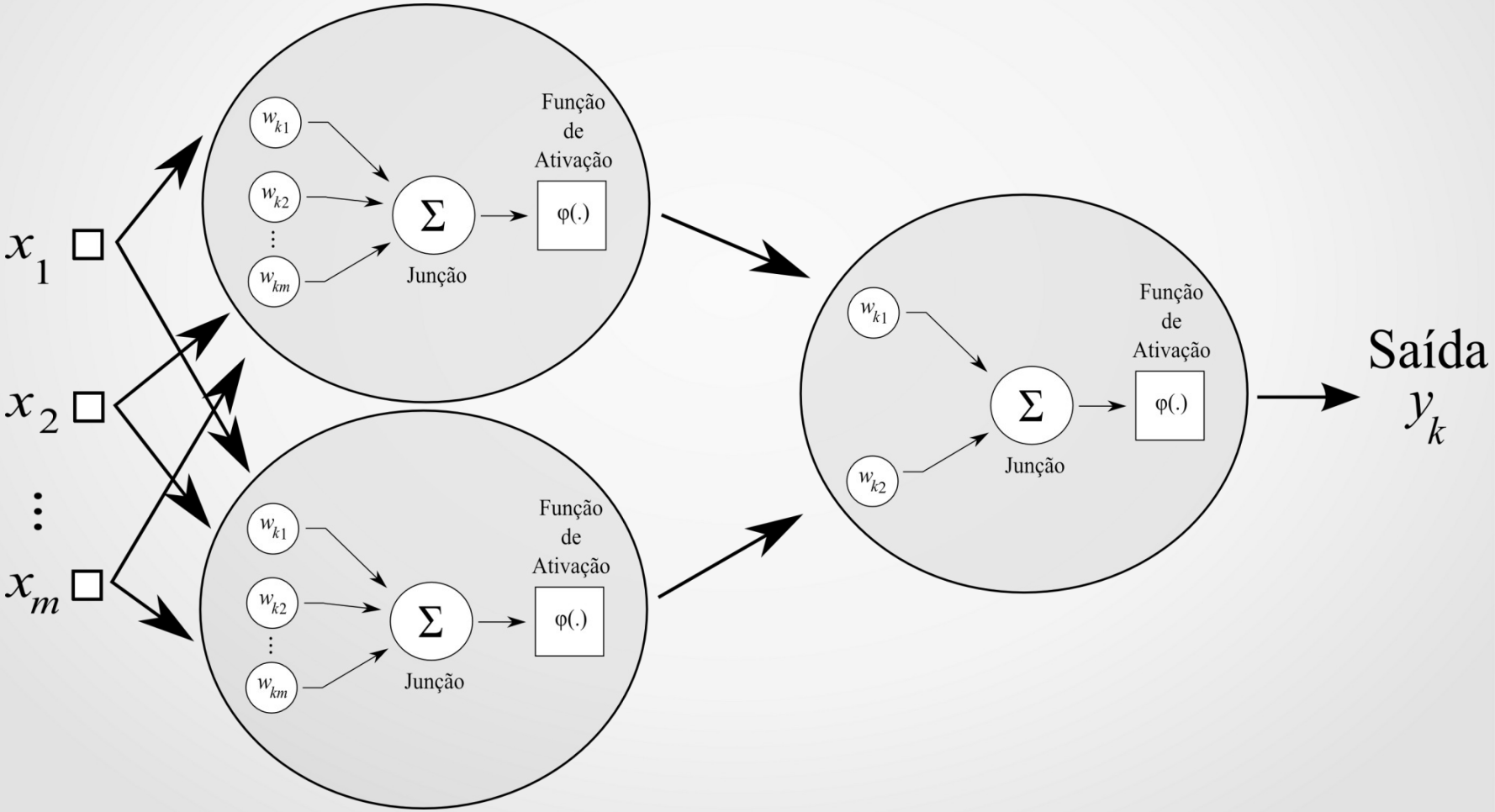
Redes neurais de múltiplas camadas

- Neurônios organizados em camadas
- Contém camadas escondidas atuam como extratores de estatísticas de mais alta ordem
- Neurônios de uma camada têm como entradas sinais provenientes apenas dos neurônios das camadas anteriores

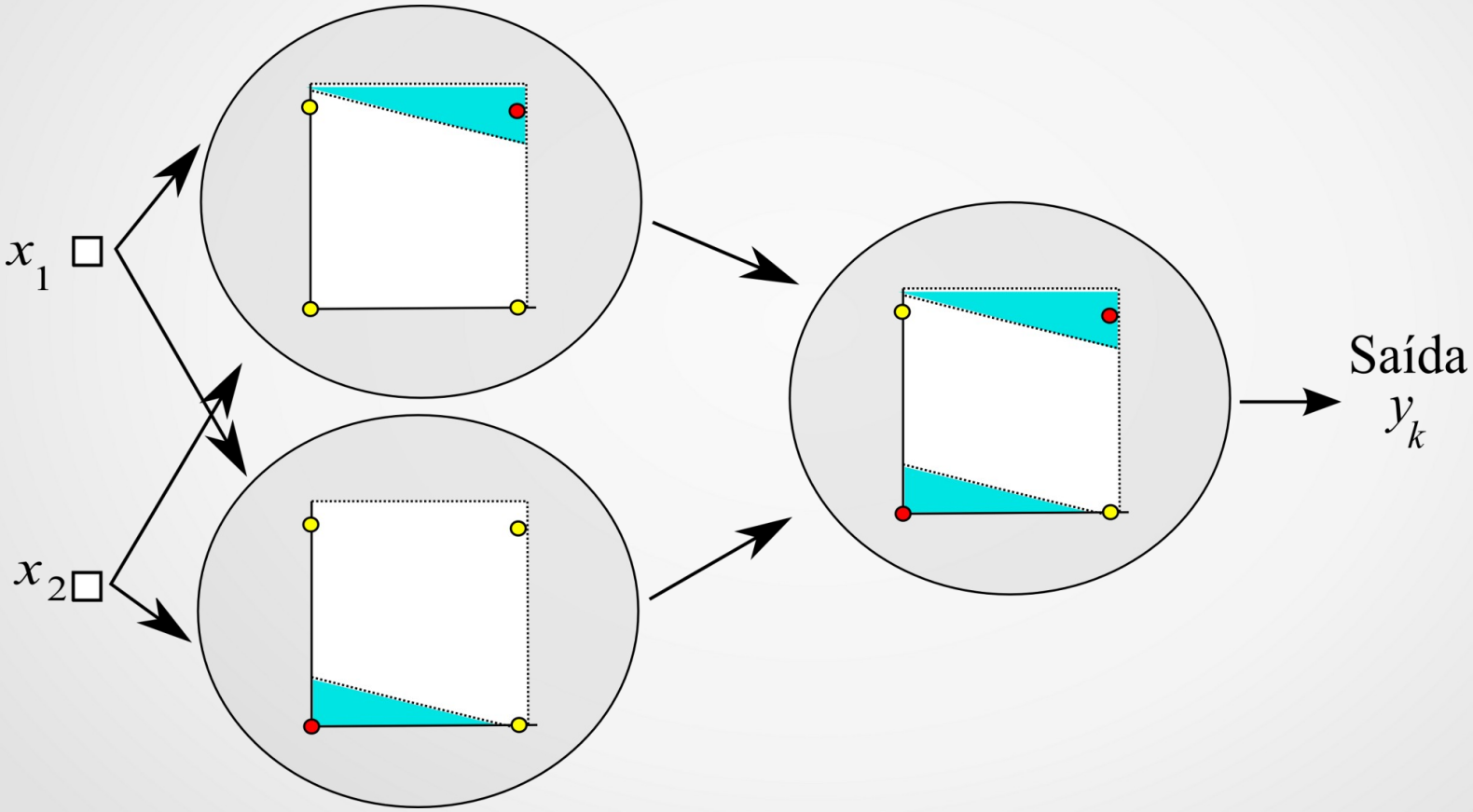
Linearidade e generalização

- A rede neural é construída por meio da interconexão de neurônios. A não linearidade é distribuída por toda a rede.
- Generalização: produção de saídas razoáveis para entradas não encontradas durante o treinamento.
- A não linearidade é importante para generalização.

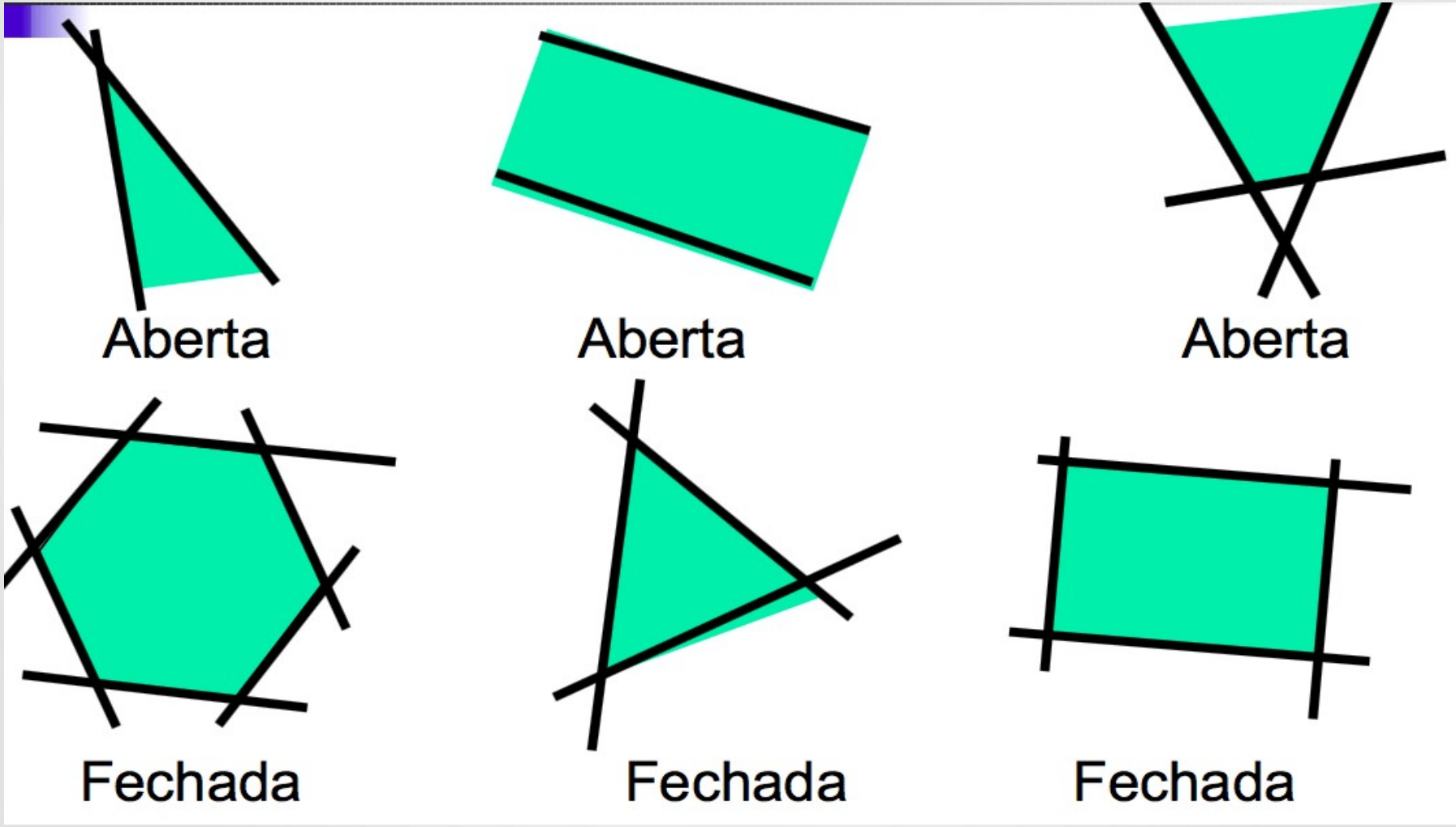
Redes Multicamadas



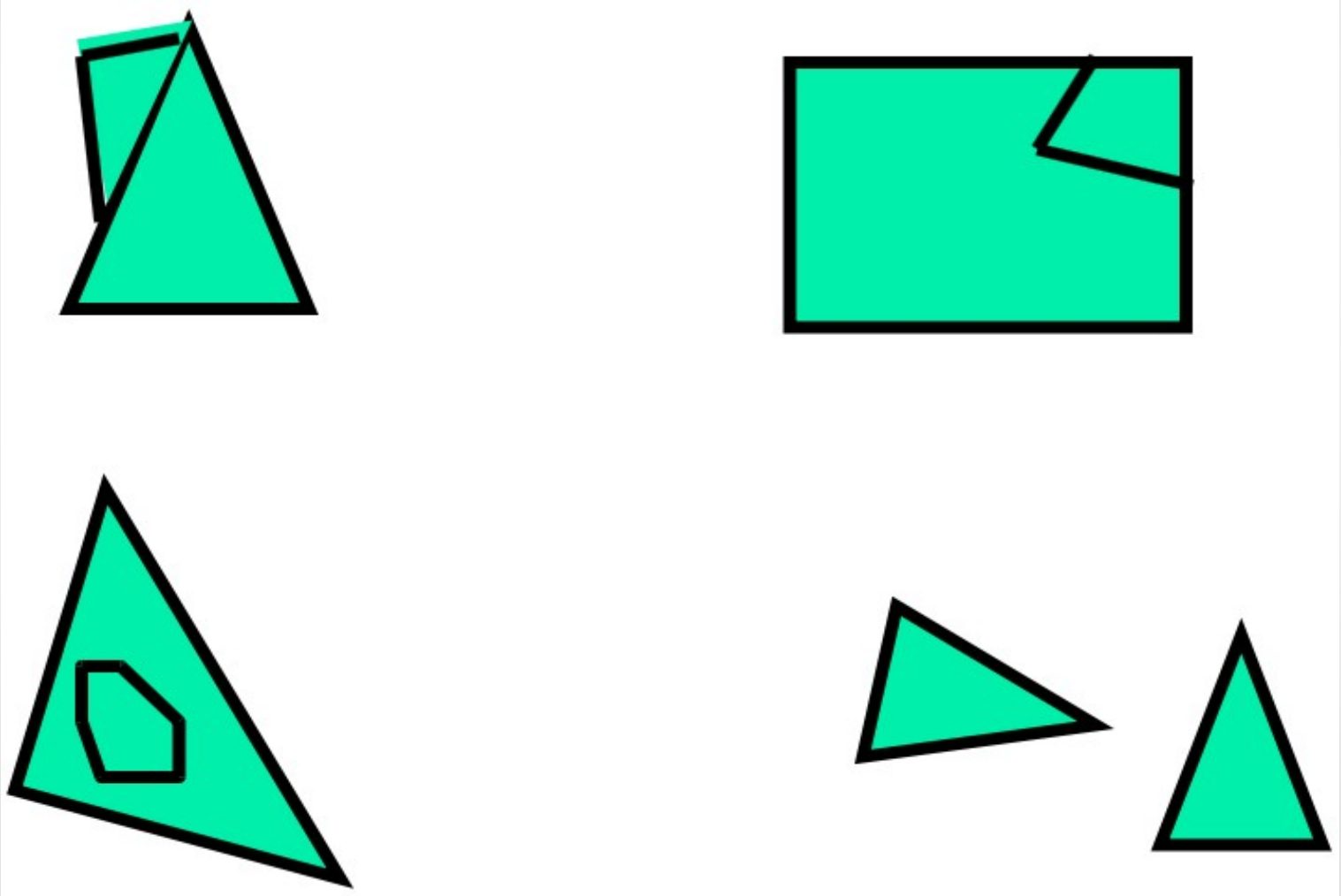
Redes Multicamadas



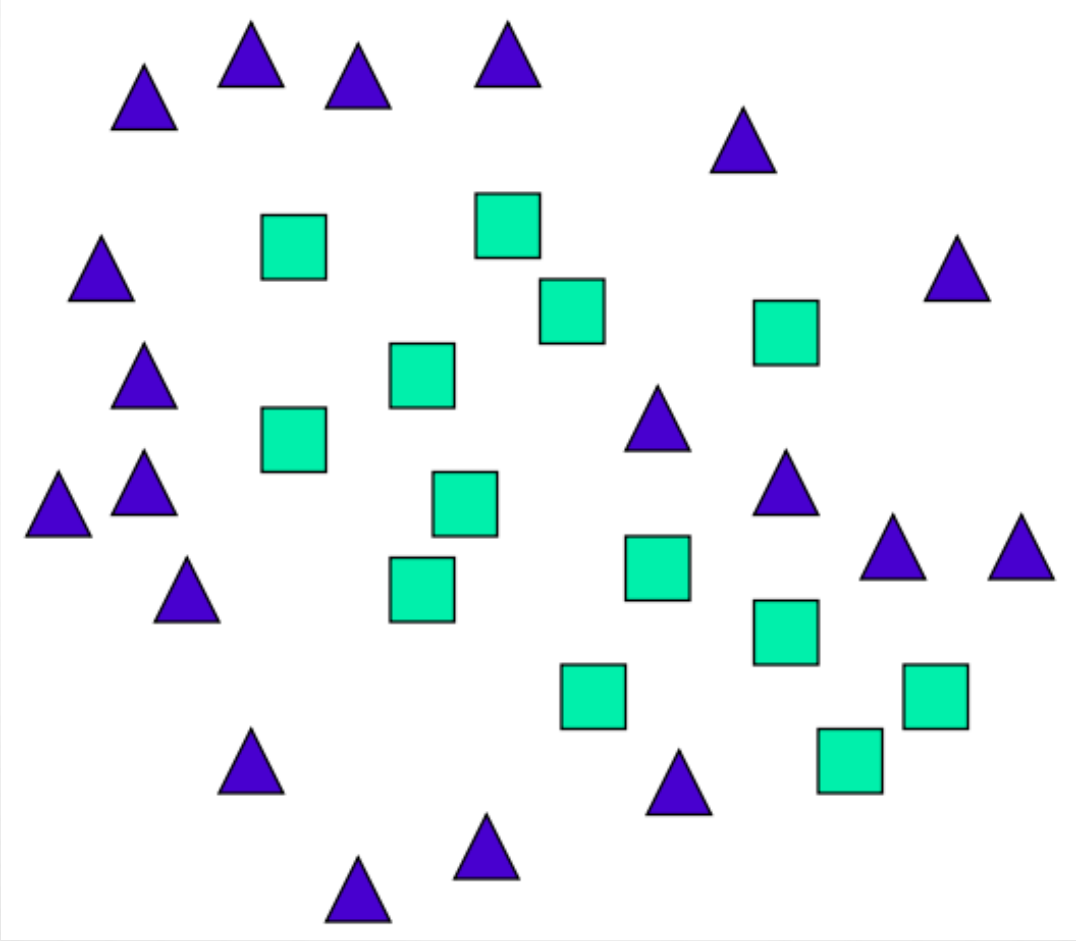
Regiões Convexas: Combinações de hiperplanos



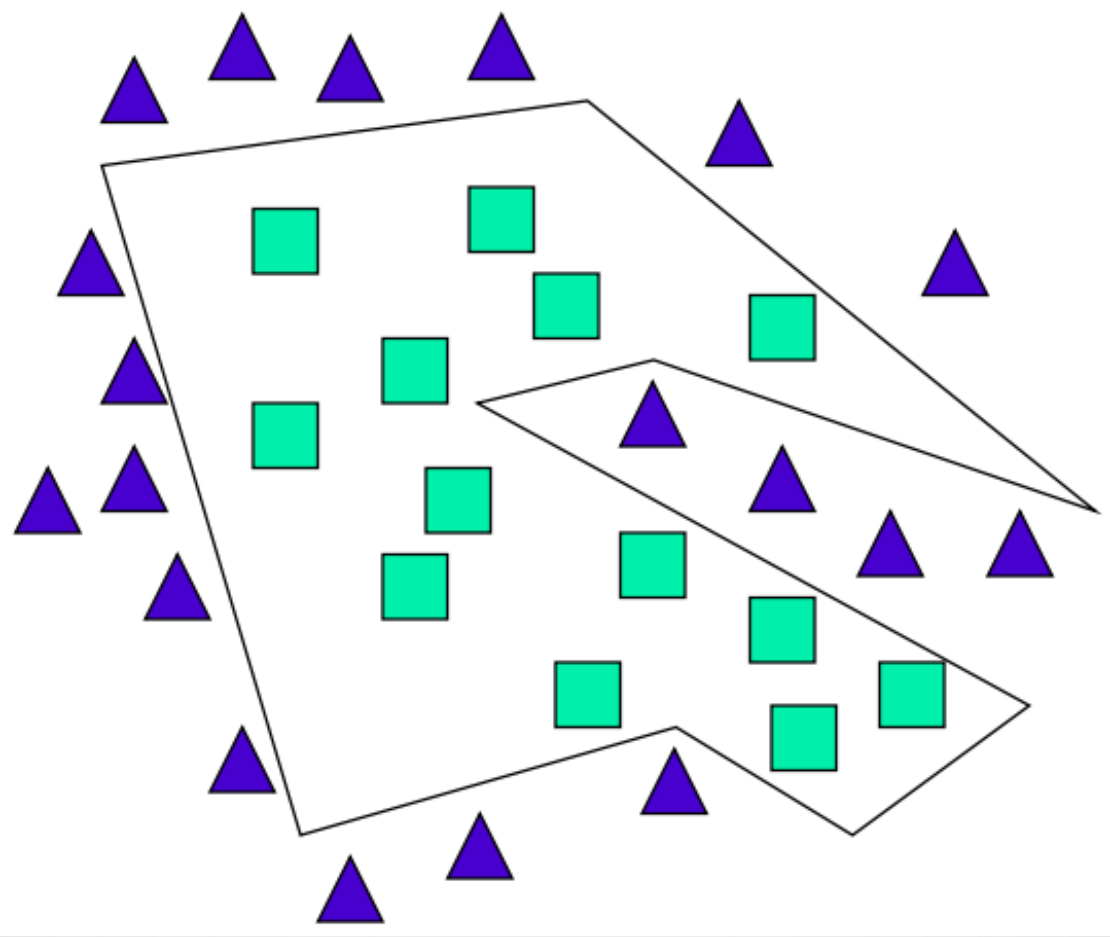
Regiões Convexas: Combinações de hiperplanos



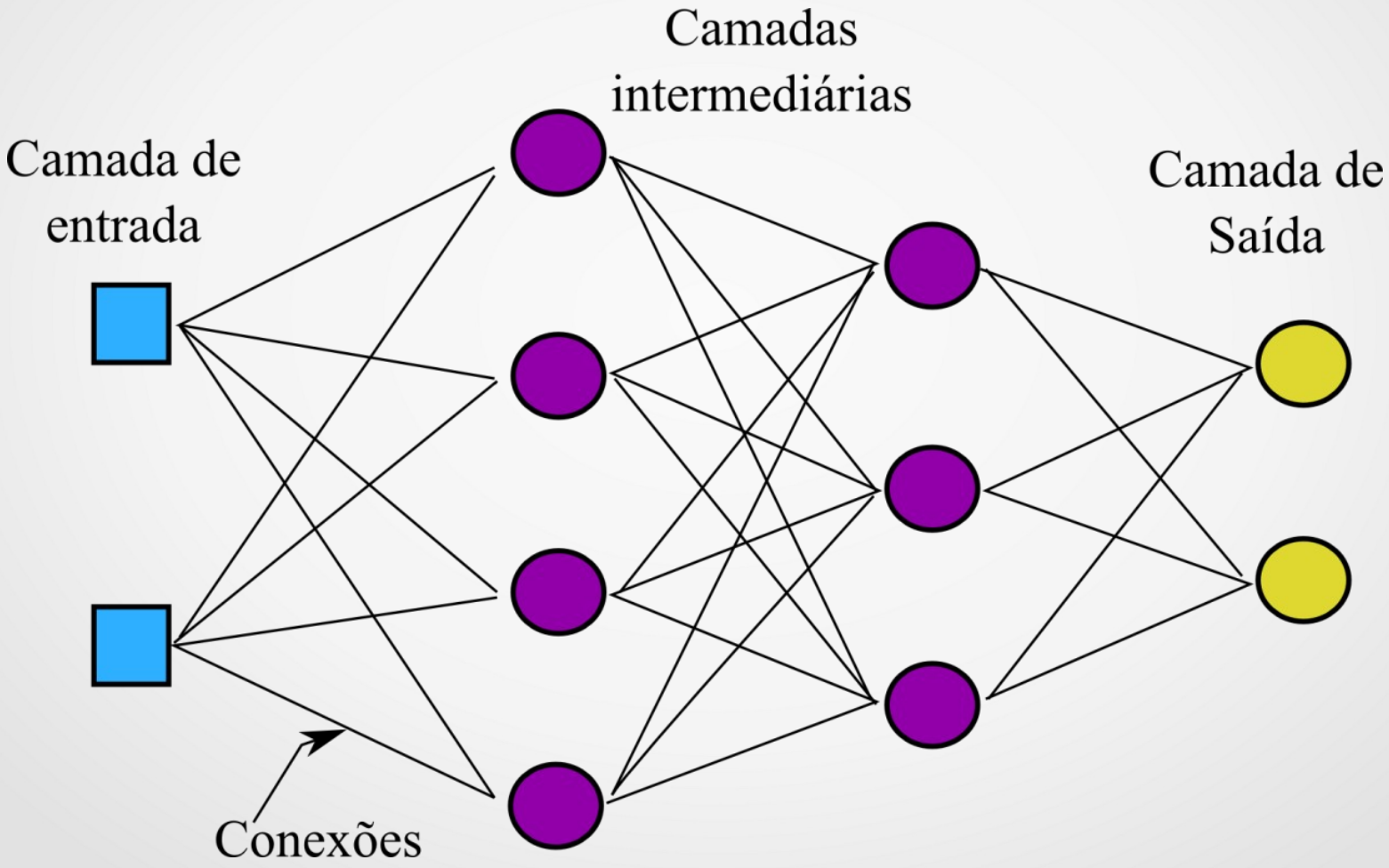
Regiões Convexas: Combinações de hiperplanos



Regiões Convexas: Combinações de hiperplanos

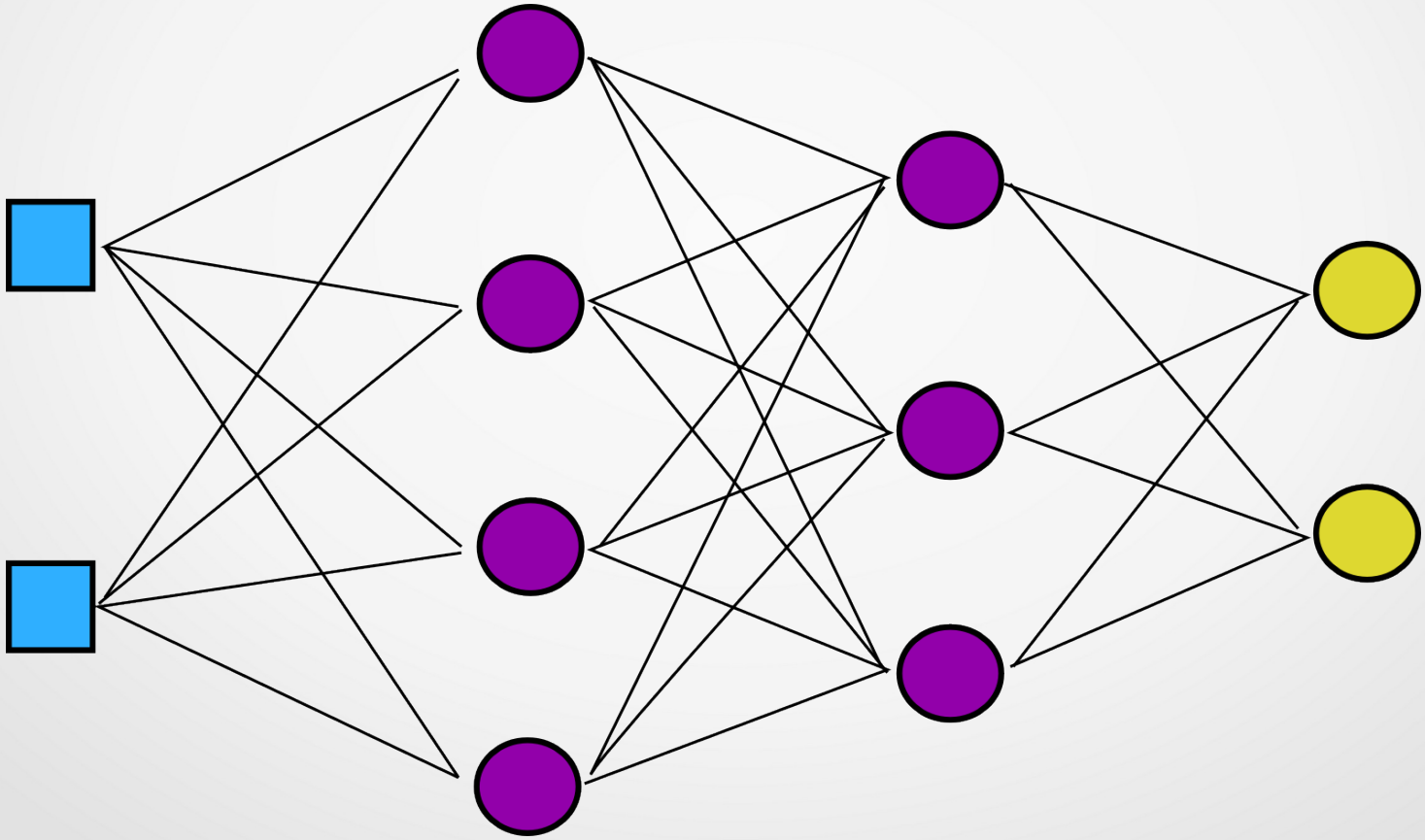


Estrutura Genérica



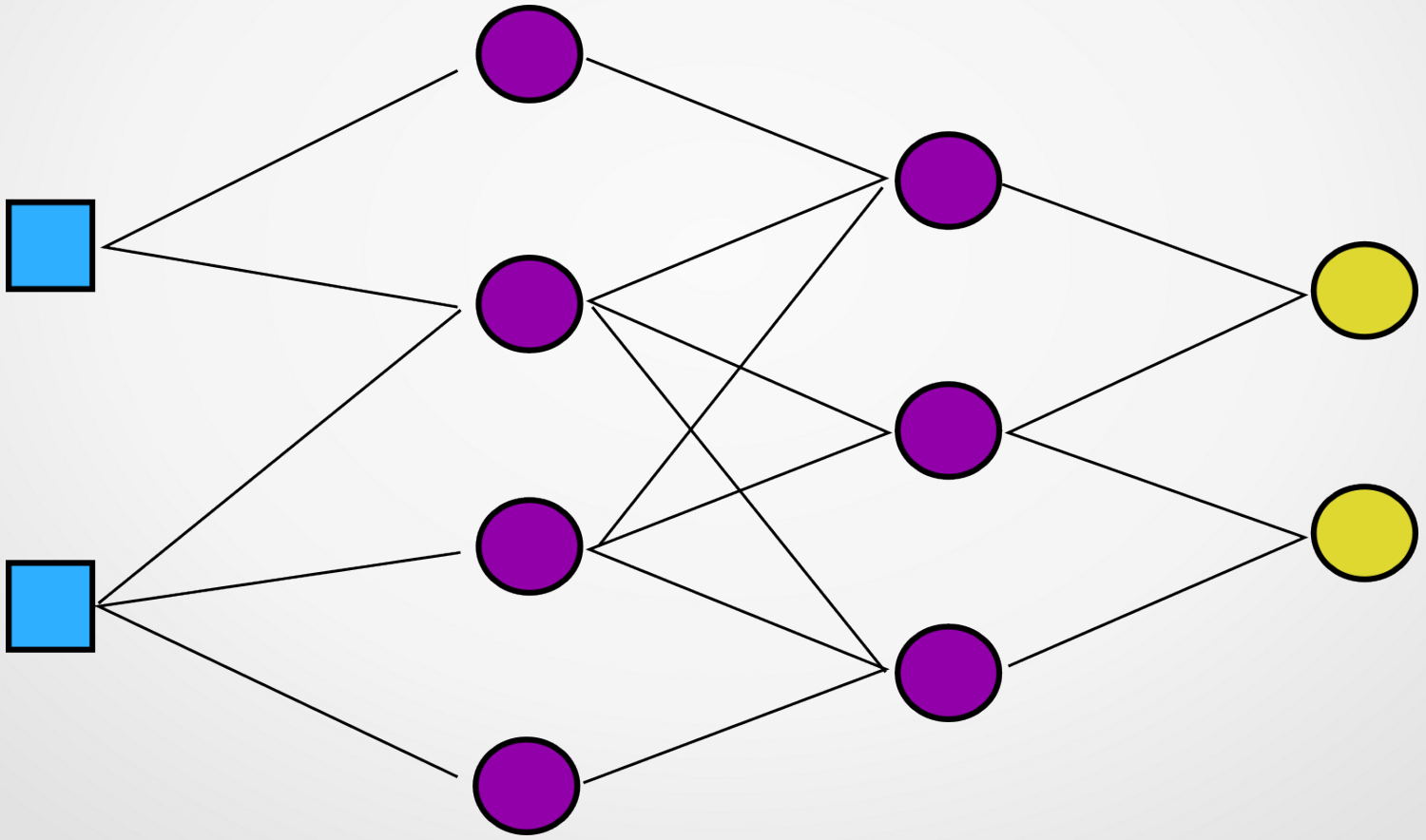
Conexões

- Completamente conectada



Conexões

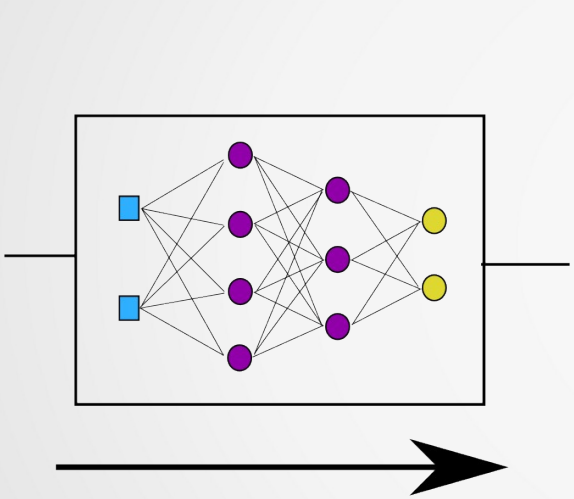
- Parcialmente conectada



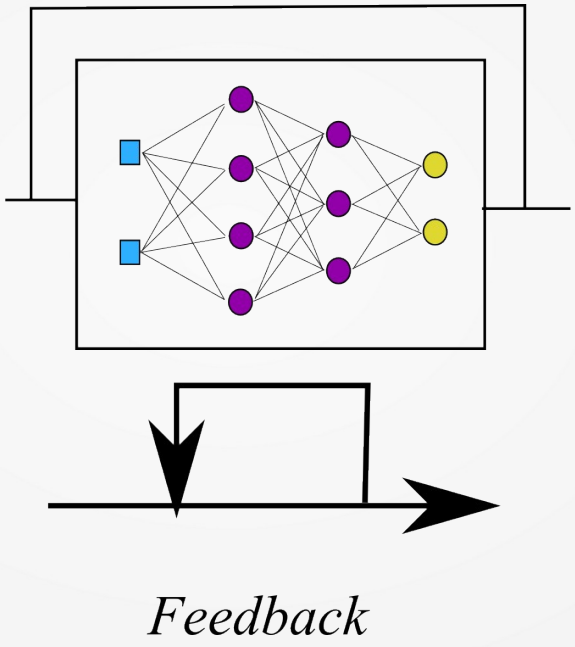
Conexões

- Excitatória
  - Quando o valor do peso é positivo
- Inibitória
  - Quando o valor do peso é negativo
- Aprendizado
  - Por meio do algoritmo retropropagação (*backpropagation)*
    - Ver nas referências bibliográficas

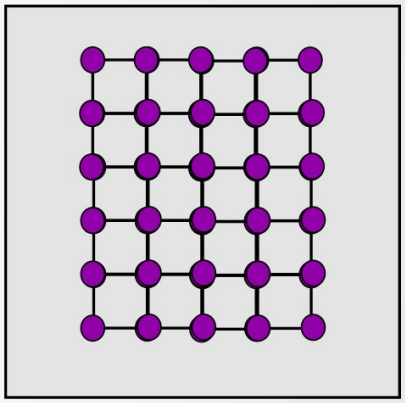
Topologia



Feedfoward



Feedback

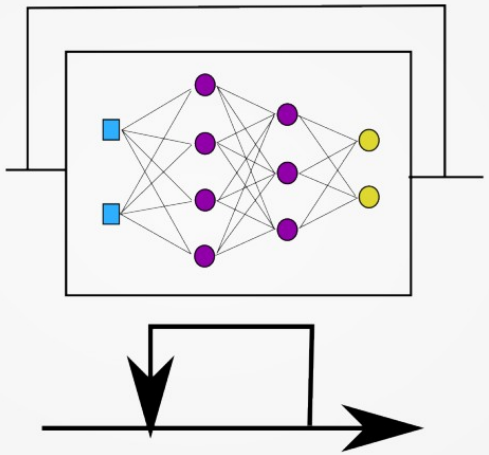


Grid

Redes Neurais Recorrentes

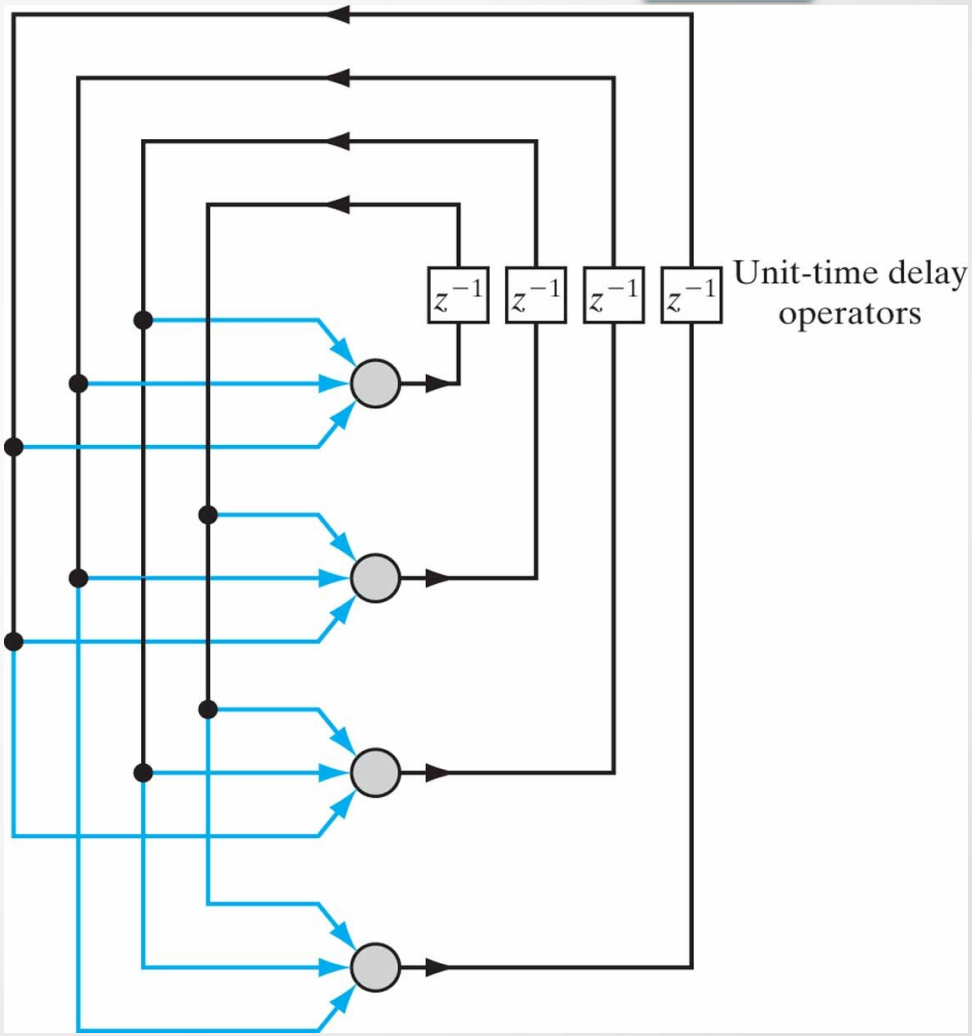
- Diferenciam-se das outras arquiteturas por possuírem retroalimentação
- Saídas de neurônios podem servir de entrada para outros neurônios e também para o próprio neurônio (auto retroalimentação)
- Podem ou não ter neurônios

  escondidos



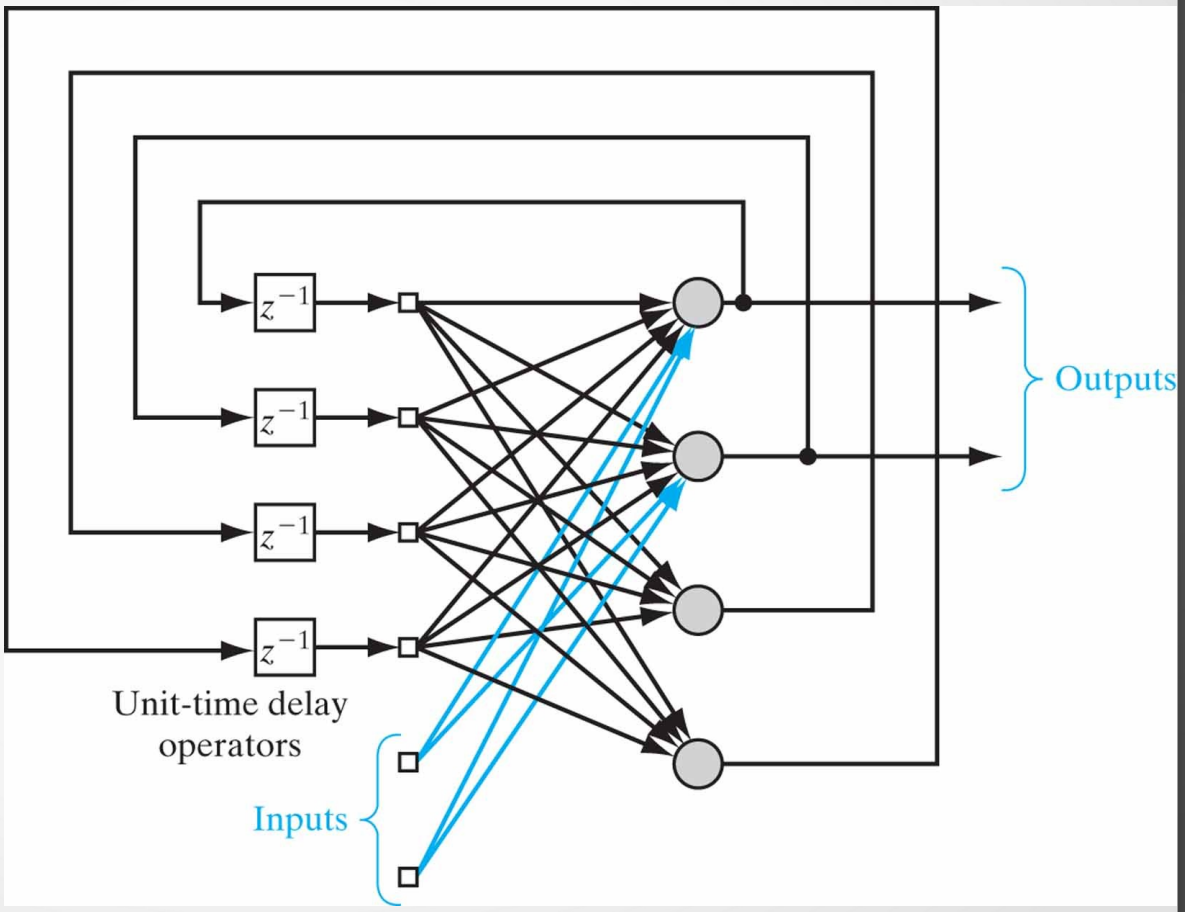
Arquiteturas Redes Neurais Recorrentes

- Redes neurais recorrentes
- Sem neurônios escondidos
- Sem auto retroalimentação



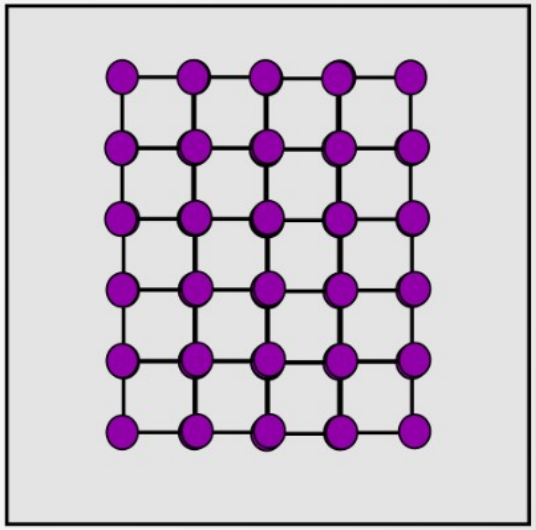
Arquiteturas Redes Neurais Recorrentes

- Redes neurais recorrentes
- Com neurônios escondidos
- Com auto retroalimentação
- z^{-1} é chamado de operador de atraso



Modelo de Kohenen

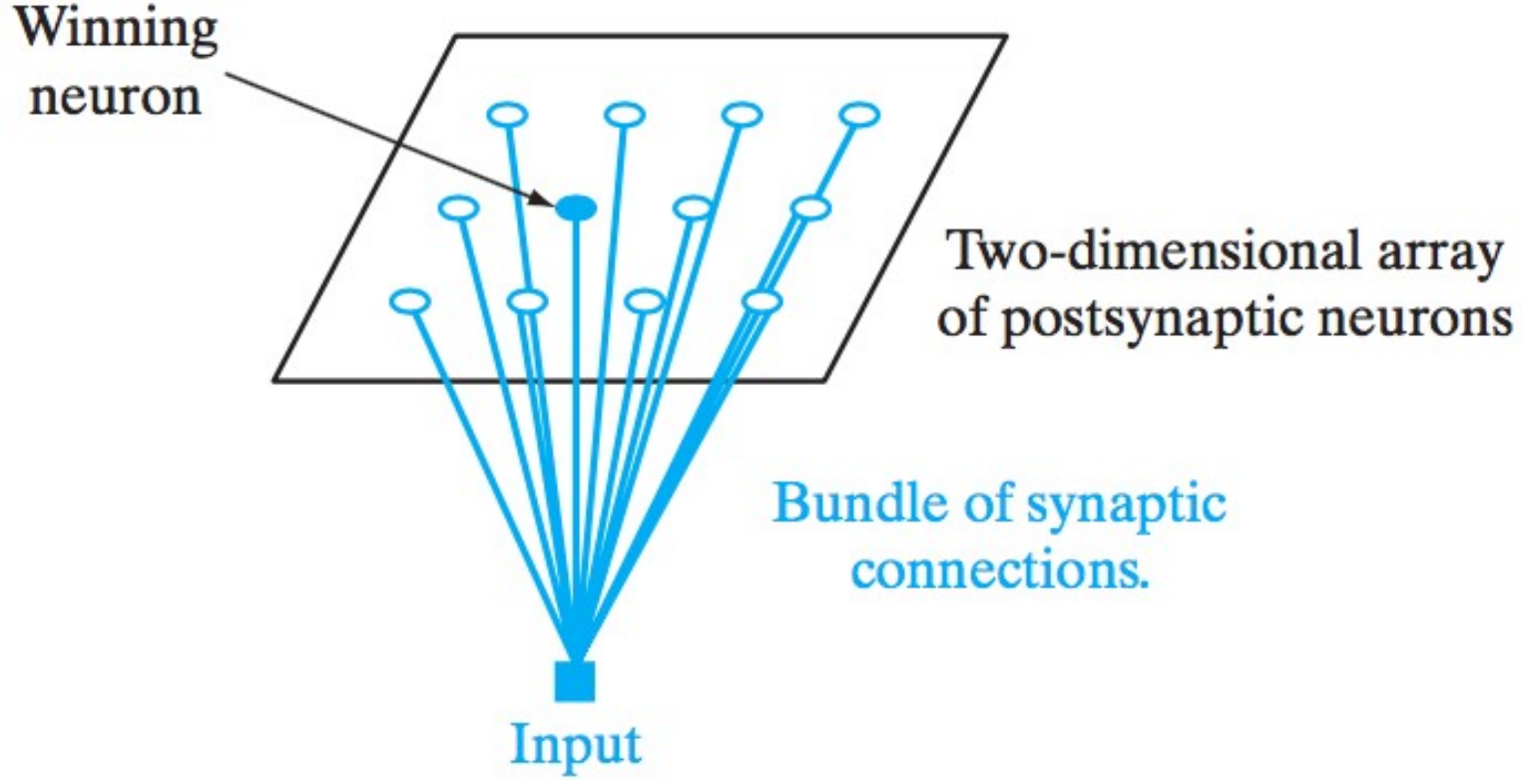
- Queremos construir mapas artificiais que aprendem por meio de auto-organização, de maneira neuro-biologicamente inspirada
- Princípio da formação do mapa topográfico:
  - A localização espacial de um neurônio de saída em um mapa topográfico corresponde a um domínio particular ou característica dos dados de entrada



Grid

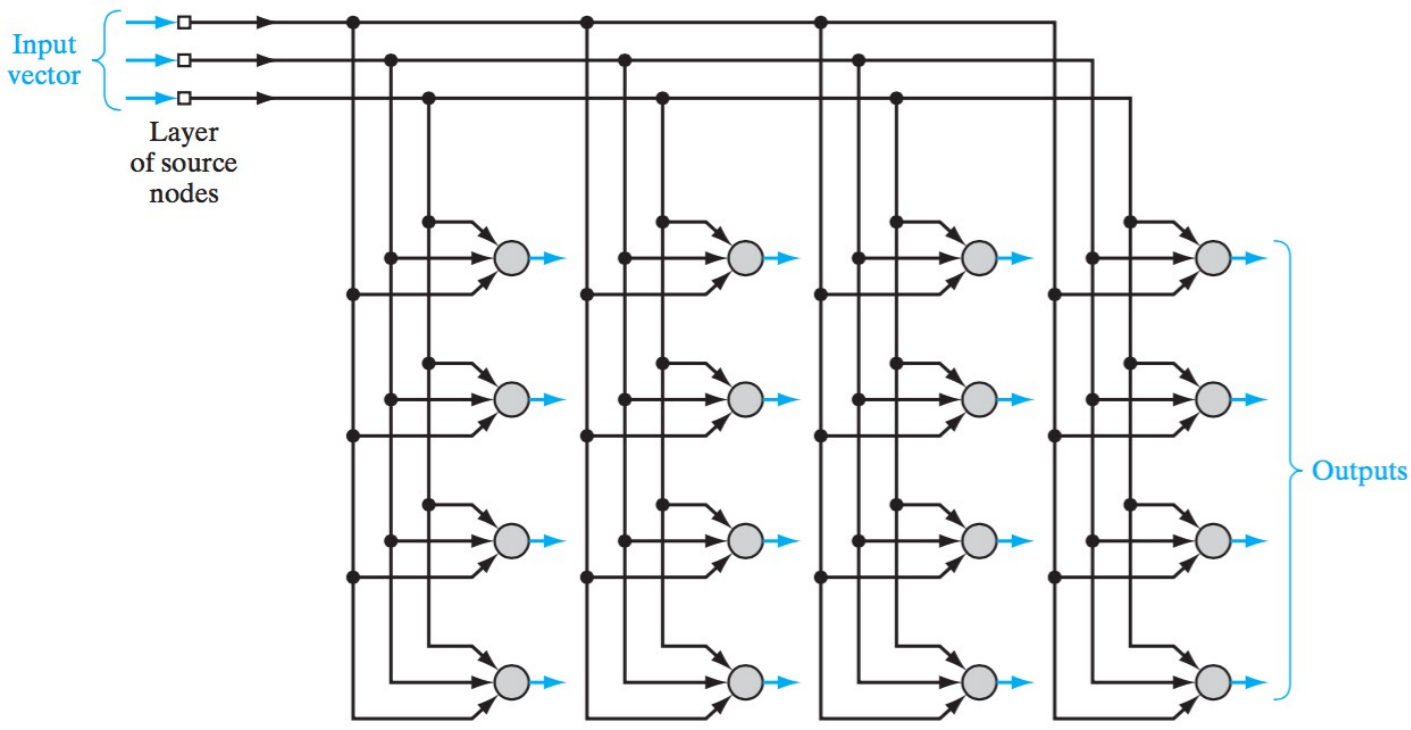
Modelo Kohonen

- Neurônio vencedor é estimulado junto aos neurônios vizinhos



Mapa Auto-Organizável (SOM)

- Transformar um padrão de entrada de dimensão arbitrária em um mapa bidimensional, de maneira adaptativa e ordenada topologicamente

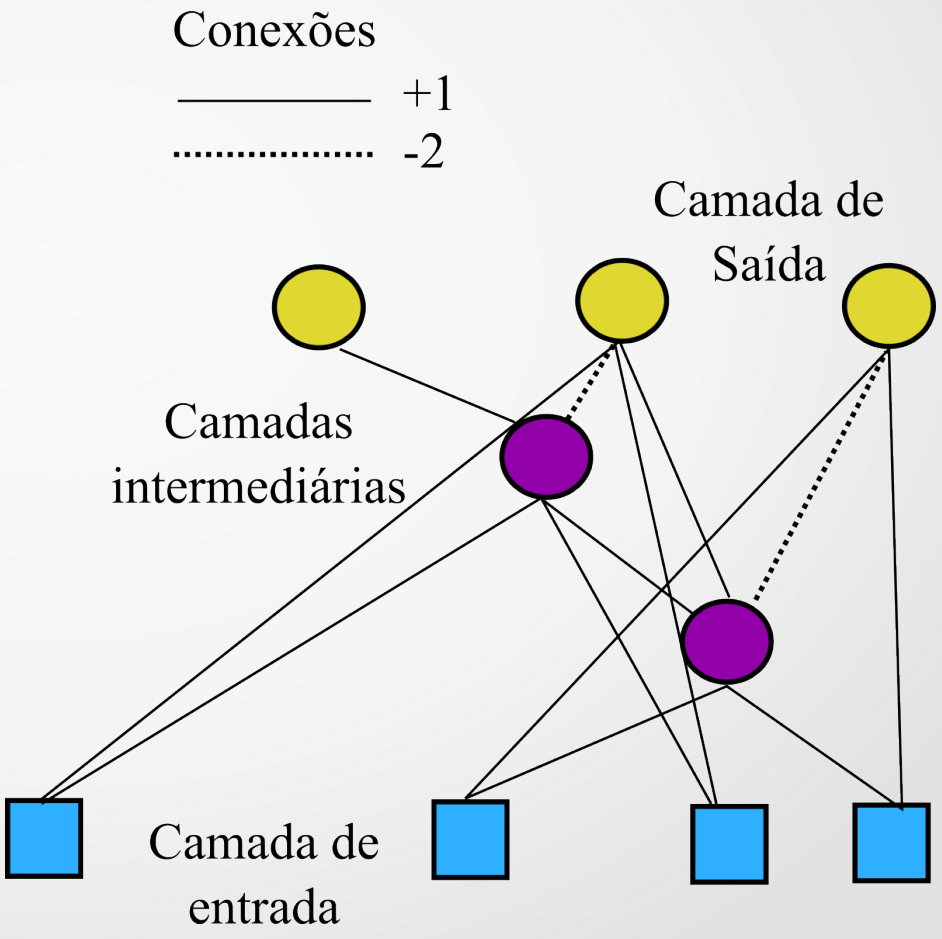


Diversos outros tipos de NN

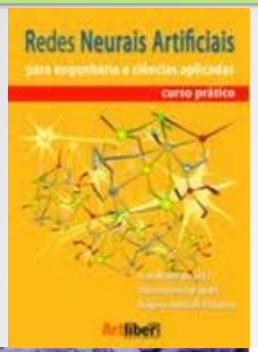
- Radial Basis Function
- Redes de Hopfield
- Redes de Boltzmann
- Aprendizado Profundo
  - Redes Convolutivas
  - Variational Autoencoder
  - etc
- Diversos tipos

Exercício

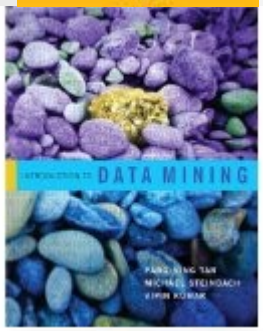
- Qual a função implementada pela rede abaixo?
- Entrada: 2 *bits*
- Saída: 3 *bits*
- Ativação limiar
  - c. saída:
    - *bias -0.5*
  - c. intermediária:
    - *bias -1.5*



Bibliografia



SILVA, I.N., SPATTI, D.H. e FLAUZINO, R. A. Redes Neurais Artificiais para engenharia e ciências aplicadas: curso prático. Editora ArtLiber 2010.



STEINBACH, M., KUMAR, V. TAN, P. Introdução ao Data Mining (Mineração de Dados). Edição 1. Ciência Moderna 2009. ISBN 9788573937619.

Inteligência Artificial - Uma Abordagem de Aprendizado de Máquina. Katti Faceli, Ana Carolina Lorena, João Gama, André C. P. L. F. de Carvalho. Grupo Gen 2011