Inteligência Artificial RNA’s não Supervisionadas

Prof. Gizelle / Marcelo Dib / Livia Ruback / Raimundo Macario

Visão Geral

Tipos de Aprendizagem

- **Supervisionada:** um ”professor“ diz quanto a resposta dada pelo sistema se aproxima da resposta desejada.
- **Não-Supervisionada:** o sistema tenta se auto-organizar baseado nas similaridades entre os exemplos a ele apresentados.

Visão Geral

Tipos de Aprendizagem

- **Supervisionada:**
  - Conjunto de treinamento s = $\{(x_1, f(x_1)), (x_2, f(x_2)), \ldots, (x_n, f(x_n))\}$
- **Não-Supervisionada:**
  - Conjunto de treinamento s = $\{(x_1, \ldots), (x_2, \ldots), \ldots, (x_n, \ldots)\}$

Visão Geral

- **Problema de classificação/Regressão**
  - Regressão Linear
  - Regressão Logistica
  - K-NN Classified (Vizinho mais próximo)
  - Rede Neural Artificial
  - Support Vector Machine(SVM)
  - Árvore de decisão
  - Random Forests
  - Ensemble Methods
  - Gradiente Tree Boosting

Visão Geral

Problema de Agrupamento ou Clusterização

- K conhecido (numero de grupos)
- K-means

Visão Geral

Problema de Agrupamento ou Clusterização Automática

- **K não conhecido (numero de grupos)**
  - from sklearn.cluster import DBSCAN
  - from sklearn.cluster import AgglomerativeClustering
  - from sklearn.cluster import OPTICS
  - from sklearn.cluster import MiniBatchKMeans
  - from sklearn.cluster import AffinityPropagationfrom sklearn.cluster import MeanShift
  - from sklearn.cluster import Birch

Redes auto organizadas

- **Motivação :**
  - Quando não existe um especialista que possa classificar as amostras de treinamento, precisamos usar uma rede neural que seja capaz de mapear de forma autônomas as entradas e saída.

Redes auto organizadas

- Este processo de aprendizado é chamado de auto-organizado (ou Competitivo) e existem muitos tipos de redes que o utilizam:
  - Os Mapas Auto Organizáveis de Kohonen (1984),
  - A rede de *aprendizagem competitiva* proposta por Rumelhart e Zipser (1985),
  - As redes ART propostas por Carpenter and Grossberg (1987).

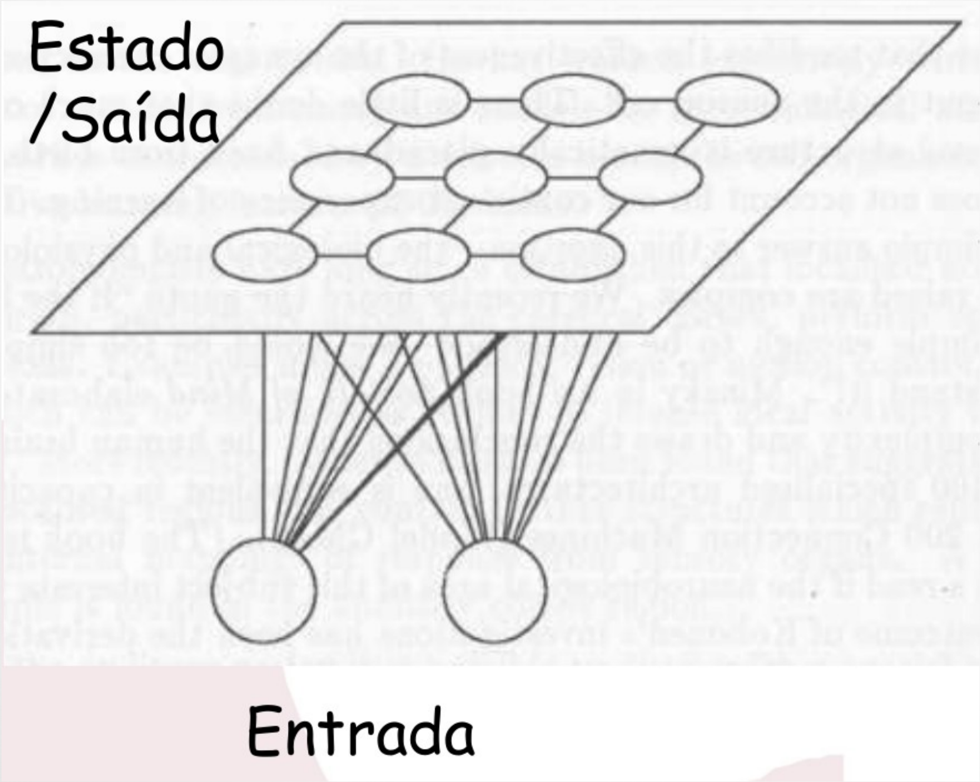
Aprendizado Competitivo

- O aprendizado competitivo é um algoritmo que divide uma série de dados de entradas em grupos (*clusters*).
- A informação é extraída sem que exista um par entrada/saída alvo.

Aprendizado Competitivo

- As redes para este tipo de problema possuem uma única camada de nós de saída ligados à camada de entrada, de tal forma que exista um nó de entrada para cada característica dos padrões e tantos nós de saída quantos sejam os *clusters* em que iremos classificar as entradas.

Aprendizado Competitivo



Aprendizado Competitivo

- O treinamento promove uma competição entre os nós da rede, onde somente um dos nós da saída é ativado para cada padrão de entrada apresentado.
- Este nó é justamente o nó que representa o *cluster* ao qual pertence aquele padrão de entrada.

Aprendizado Competitivo

- Durante o treinamento, a cada dado de entrada apresentado, o algoritmo calcula algum tipo de distância, como o produto escalar ou a distância euclidiana, dentre outros, entre os vetores de entrada e os nós de saída.

Aprendizado Competitivo

- Desta forma, descobre-se qual dos nós da saída melhor representa a entrada atual.
- A cada entrada apresentada à rede, os pesos que ligam a entrada ao nó de saída são ajustados, diminuindo a distância que separa a entrada do nó de saída.

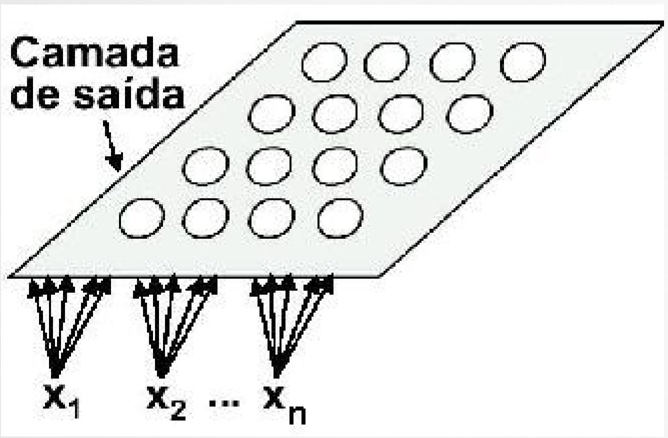
Aprendizado Competitivo

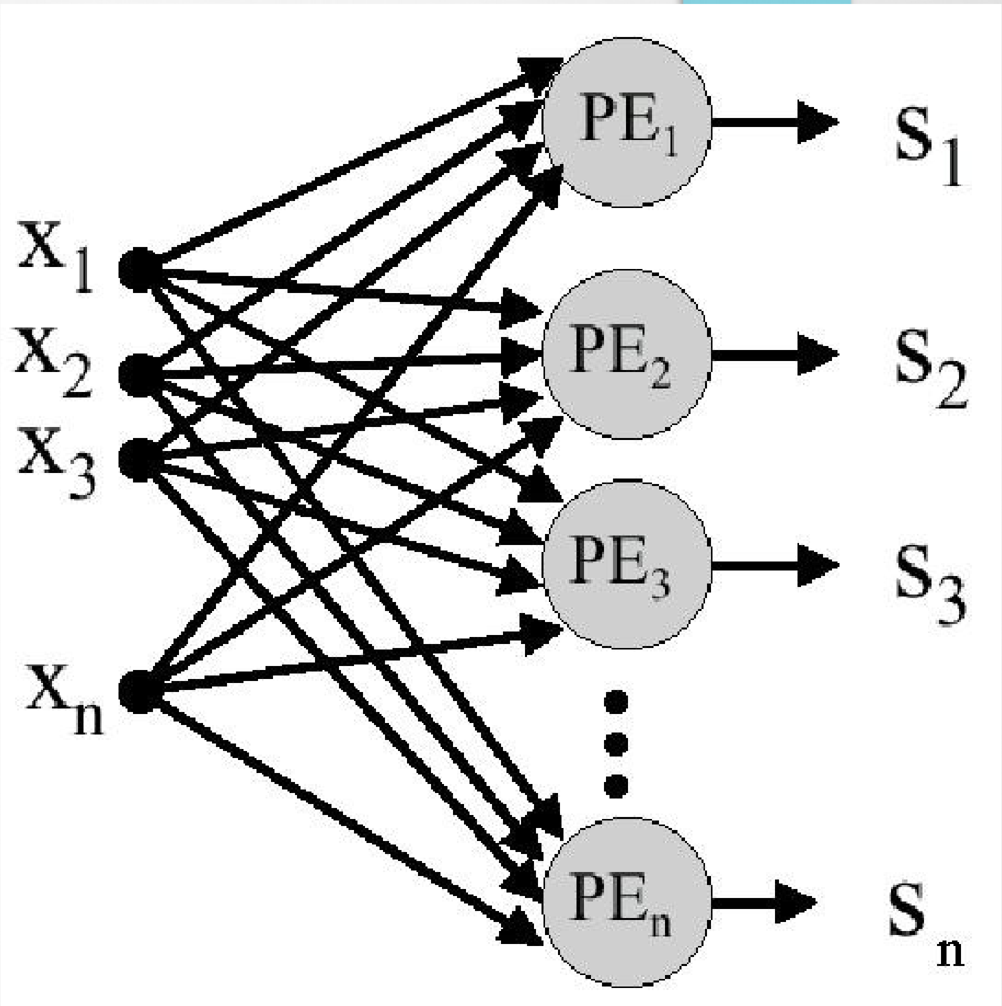
- Somente os pesos que ligam a entrada a este nó serão atualizados.
- Em uma rede já treinada, todos os vetores de entrada que pertencerem a um mesmo *cluster* acionarão o mesmo nó de saída.

Redes de Kohonen

- As redes de Kohonen, também conhecidas como Mapas Auto Organizáveis de Kohonen, ou redes SOM, foram propostas por Teuto Kohonen em 1984;
- A arquitetura é constituída de uma só camada de nós de saída, organizados como uma malha, totalmente ligada ao vetor de nós da entrada.

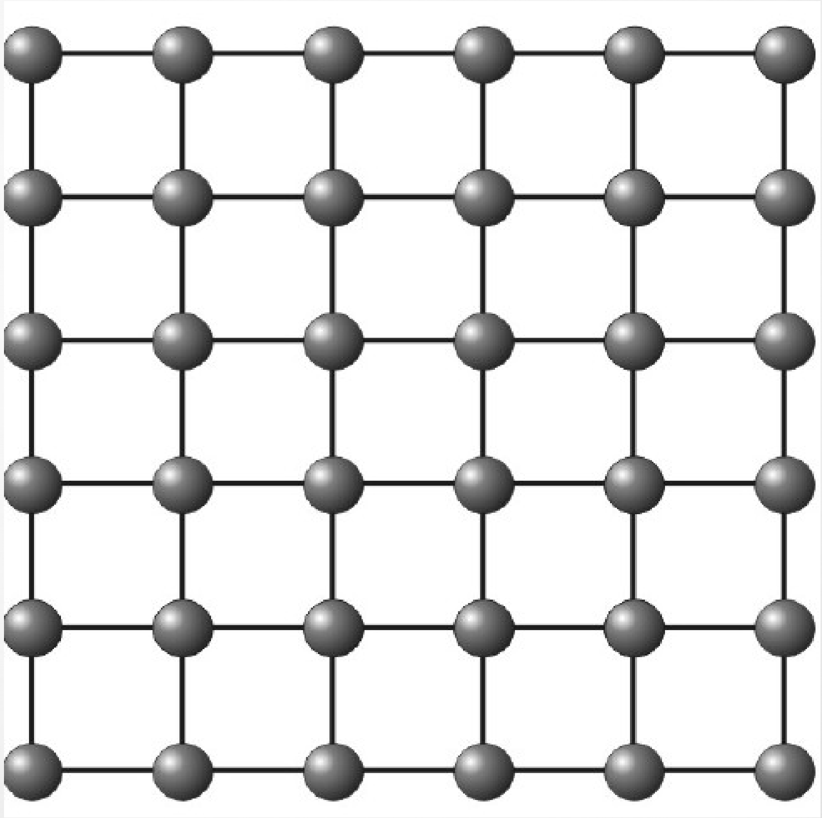
Redes de Kohonen





Redes de Kohonen

- **Malha de Kohonen**



Redes de Kohonen

Vizinhança

- Nesta rede, o comportamento de um determinado nó é diretamente afetado pelo comportamento dos nós vizinhos, em uma vizinhança local.
- Cada neurônio de saída está conectado a todos os componentes de entrada.

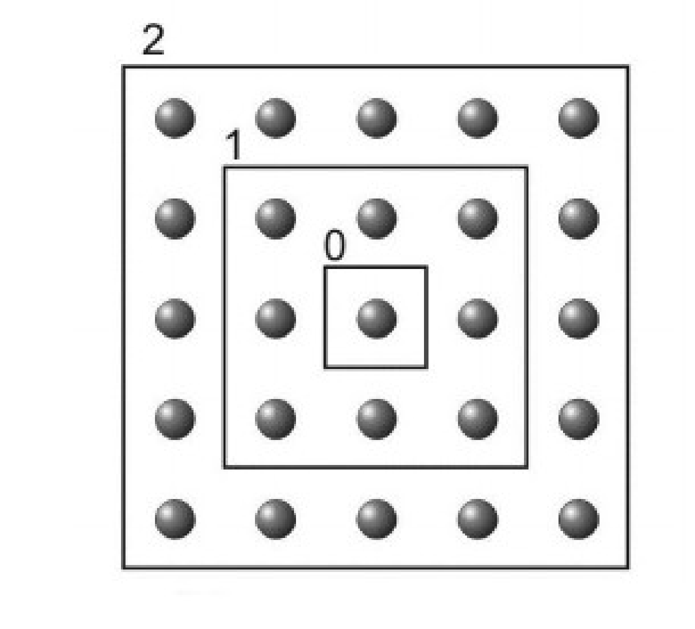
Redes de Kohonen

Vizinhança

- Durante o treinamento são alterados não apenas os pesos que chegam ao neurônio vencedor, mas também os pesos dos neurônios de sua vizinhança local.

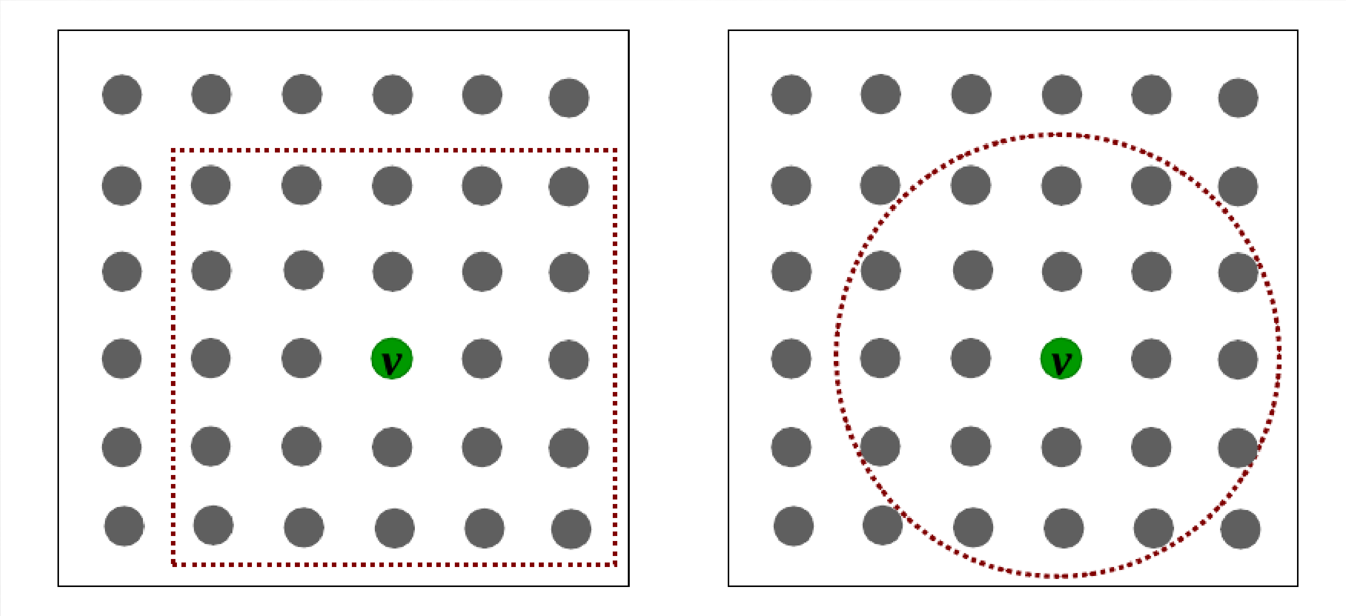
Redes de Kohonen

Vizinhança



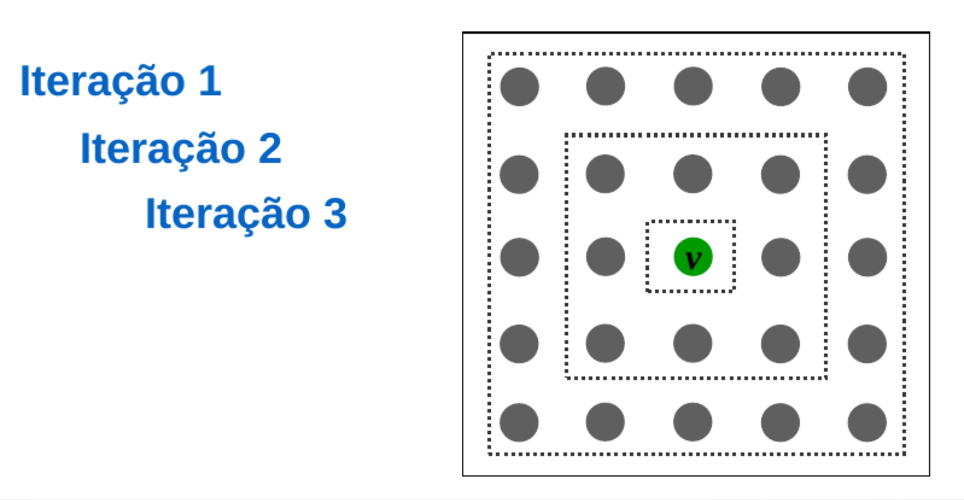
Redes de Kohonen

- Exemplos de função de vizinhança
  - Base Quadrada / Base Circular



Redes de Kohonen

- Inicia-se o algoritmo com alto nível de vizinhança e esta é reduzida conforme o tempo avança
- É necessário diminuir a região de vizinhança para obter a auto-organização do mapa



Redes de Kohonen

- Uma função muito interessante a ser usada como função de vizinhança é a Gaussiana, dada por:

$$h_{vj}(t) = \exp\left(-\frac{\left\| \mathbf{r}_v - \mathbf{r}_j \right\|}{2\sigma^2(t)} \right)$$

onde o termo $||r_v - r_j||$ é a distância entre o neurônio v vencedor e o neurônio j que está sendo atualizado

Redes de Kohonen

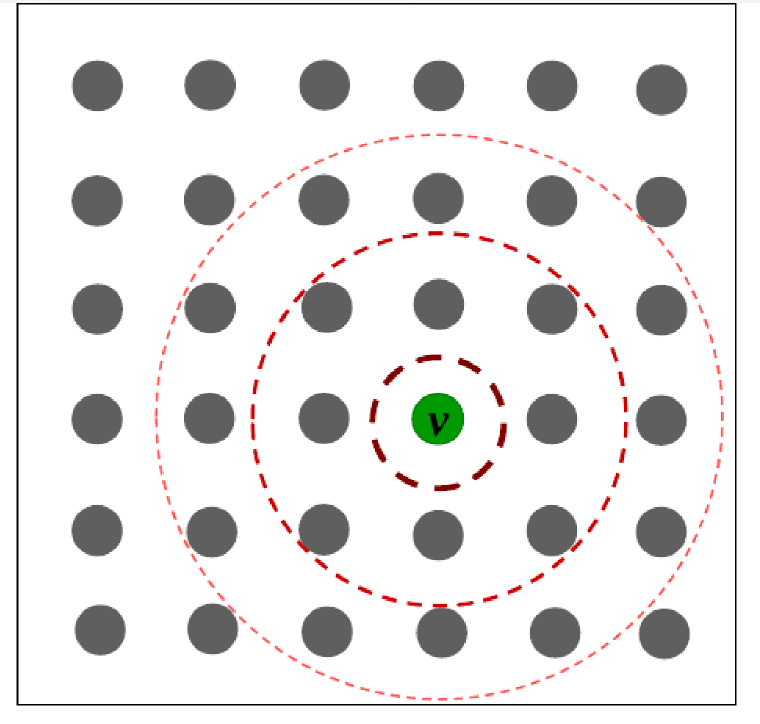
- O parâmetro σ define a largura da função e deve ser decrescente no tempo. Pode ser usada uma função linear, mas em geral é usada a exponencial:

$$\sigma(t) = \sigma(0)\exp\left(-\frac{t}{\tau_1}\right)$$

- Onde σ(0) é um valor inicial para σ
- onde τ_1 é uma constante de tempo do SOM definida para que a taxa de aprendizagem nunca decaia para zero

Redes de Kohonen

Exemplo de função de vizinhança Gaussiana



Redes de Kohonen - Algoritmo

1) Escolha, aleatoriamente, q vetores de pesos iniciais.
2) Apresente a entrada x(t) (t = 0 na primeira iteração).
3) Calcule a distância de x(t) a cada um dos q vetores de pesos w_j, j=1. . .q, e nomeie como neurônio vencedor aquele para o qual $||w_j(t) - x(t)||$ é minimizado.
4) Escolha uma vizinhança $h_{vj}(t)$ para o neurônio vencedor i.

Redes de Kohonen - Algoritmo

5) Atualize os pesos

$$\underbrace{\mathbf{w}_j(t+1)}_{\text{Vetor peso atualizado}} = \underbrace{\mathbf{w}_j(t)}_{\text{Vetor peso anterior}} + \underbrace{\alpha(t)}_{\text{Taxa de aprendizagem}} \underbrace{h_{vj}(t)}_{\text{Vizinhança}} \underbrace{[\mathbf{x} - \mathbf{w}_j(t)]}_{\text{Adaptação}}$$

O parâmetro de aprendizagem α , assim como a função de vizinhança deve decrescer com o tempo, para que as adaptações sejam cada vez mais “finas”

6) Retorne ao passo 2 até a convergëncia

Exemplos

- Dado 10 mil padrões de clientes bancários, cada um formado por um vetor de tamanho 5, contendo as informações de: idade; imóvel (próprio ou não); CEP; profissão; renda.
- Se escolhermos uma malha bidimensional 4x4 como saída, teremos 16 neurônios na malha, cada um deles ligado ao vetor de entradas através de um vetor de pesos de tamanho 5. (Inicialmente, são gerados 16 vetores de pesos de 5 posições)
- Ao final da fase de treinamento, poderemos atribuir 1 das 16 classes possíveis a cada um dos 10 mil padrões de entrada.

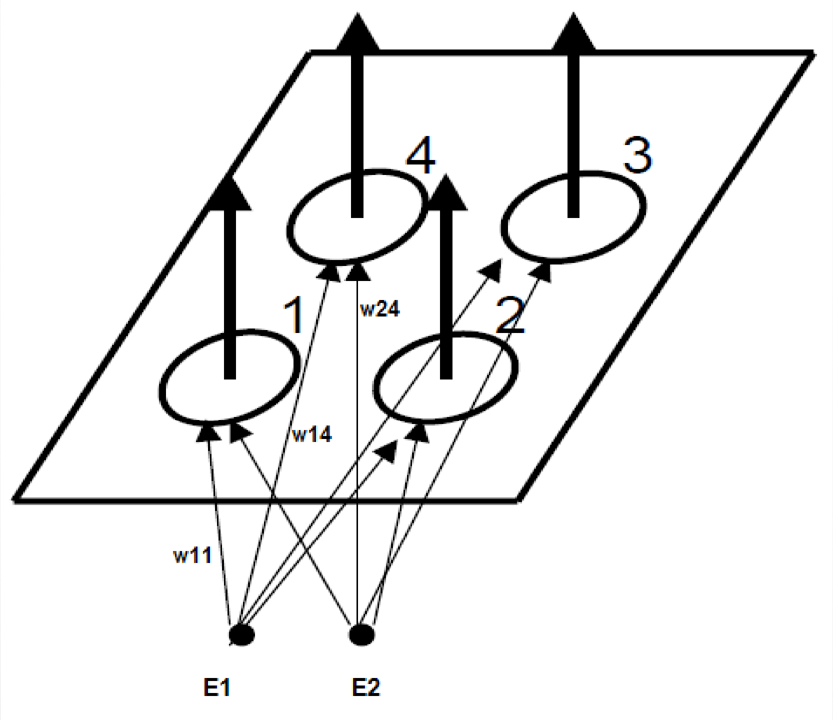
Exemplos

- Suponha uma rede simples para classificar veículos, com a vizinhança sendo o próprio neurônio, duas iterações, distância euclidiana e $\eta = 0.8$ (fixo).
- Como características serão usadas número de rodas e a existência ou não de motor. O valor 1 indica a existência de motor enquanto 0 indica a ausência.
- Usaremos os dados de dois veículos para o treinamento:

	Rodas	**Motor**
bicicleta	2	0
carro	4	1

Exemplos

- Como há duas características, a camada de entrada da rede consistirá de dois neurônios. Na camada de saída serão utilizados quatro neurônios.



Exemplos

- Aleatoriamente os pesos são inicializados para:
  - neurônio 1 = { w11, w21} = {1, 2}
  - neurônio 2 = { w12, w22} = {2, 2}
  - neurônio 3 = { w13, w23} = {1, 3}
  - neurônio 4 = { w14, w24} = {3, 2}
- PRIMEIRA ITERAÇÃO:
- Apresentado as característica da bicicleta {2,0} e calculando as distâncias:
  - $d1 = (2-1)^2 + (0-2)^2 = 5$
  - $d2 = (2-2)^2 + (0-2)^2 = 4$
  - $d3 = (2-1)^2 + (0-3)^2 = 10$
  - $d4 = (2-3)^2 + (0-2)^2 = 5$

Exemplos

- O neurônio vencedor é o segundo (d2). Ajustando seu peso:
  - $w12 = w12 + 0.8(x1(t) - w12(t)) = 2 + 0.8.(2-2) = 2$
  - $w22 = w22 + 0.8(x2(t) - w22(t)) = 2 + 0.8.(0-2) = 0.4$
- Apresentado as características do automóvel {4,1} e calculando as distâncias:
  - $d1 = (4-1)^2 + (1-2)^2 = 10$
  - $d2 = (4-2)^2 + (1-0.4)^2 = 4.36$
  - $d3 = (4-1)^2 + (1-3)^2 = 13$
  - $d4 = (4-3)^2 + (1-2)^2 = 2$
- O neurônio vencedor é o quarto (d4). Ajustando seu peso:
  - $w14 = w14 + 0.8(x1(t) - w14(t)) = 3 + 0.8.(4-3) = 3.8$
  - $w24 = w24 + 0.8(x2(t) - w24(t)) = 2 + 0.8.(1-2) = 1.2$

Exemplos

- SEGUNDA ITERAÇÃO:
- Apresentado as característica da bicicleta {2,0} e calculando as distâncias:
  - $d1 = (2-1)^2 + (0-2)^2 = 5$
  - $d2 = (2-2)^2 + (0-0.4)^2 = 0.16$
  - $d3 = (2-1)^2 + (0-3)^2 = 10$
  - $d4 = (2-3.8)^2 + (0-1.2)^2 = 4.68$
- Novamente o neurônio vencedor é o segundo (d2). Ajustando seu peso:
  - w12 = w12 + 0.8(x1(t) - w12(t)) = 2 + 0.8.(2-2) = 2
  - w22 = w22 + 0.8(x2(t) - w22(t)) = 0.4 + 0.8.(0-0.4) = 0.08

Exemplos

- Apresentado as características do automóvel {4,1} e calculando as distâncias:
  - $d1 = (4-1)^2 + (1-2)^2 = 10$
  - $d2 = (4-2)^2 + (1-0.08)^2 = 4.8464$
  - $d3 = (4-1)^2 + (1-3)^2 = 13$
  - $d4 = (4-3.8)^2 + (1-1.2)^2 = 0.08$
- Novamente o neurônio vencedor é o quarto (d4). Ajustando seu peso:
  - $w14 = w14 + 0.8(x1(t) - w14(t)) = 3.8 + 0.8.(4-3.8) = 3.96$
  - $w24 = w24 + 0.8(x2(t) - w24(t)) = 1.2 + 0.8.(1-1.2) = 1.04$

Exemplos

- Vamos testar a rede, apresentando uma moto que possui 2 rodas e motor. Como as entradas são mais parecidas à da bicicleta, a rede deve ir para o mesmo neurönio da bicicleta.
- Verificando a saída da rede para a entrada da moto:

x = {2,1}

  - d1 = (2-1)2 + (1-2)2 = 2
  - d2 = (2-2)2 + (1-0.08)2 = 0.8464
  - d3 = (2-1)2 + (1-3)2 = 5
  - d4 = (2-3.8)2 + (1-1.2)2 = 3.8432

Exemplos

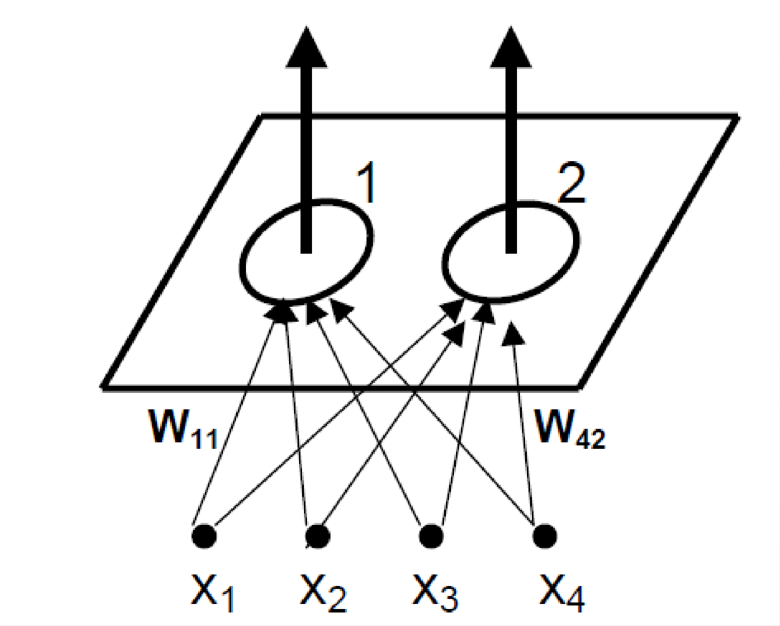
- Classificar 4 vetores, com quatro características cada, em dois *clusters* diferentes. Os vetores são:
  - a =(1,1,0,0) b=(0,0,0,1) c=(1,0,0,0) d=(0,0,1,1)
- A taxa de aprendizado η começará com 0,6 e será cortada pela metade a cada iteração, isto é,

$$\eta(t+1) = \eta(t)/2$$

- O treinamento se encerrará quando a taxa de treinamento cair abaixo de 0,01 ou o erro médio para os quatro vetores cair também abaixo de 0,01 (estes critérios são arbitrários).

Exemplos

- Iremos classificar em dois *clusters* e usaremos uma arquitetura bem simples com apenas dois nós de saída (1 e 2) e uma vizinhança de raio zero, isto é, somente o nó vencedor terá seus pesos alterados para cada padrão apresentado.



Exemplos

- Iniciamos os pesos, aleatoriamente, com os valores:
  - W1 = (0,2 0,6 0,5 0,9) e W2 = (0,8 0,4 0,7 0,3)
- Calculando as distâncias dos vetores de entrada aos vetores de pesos de cada nó, teremos, para o vetor a=(1,1,0,0):
  - $D1 = (0{,}2 - 1)^2 + (0{,}6 - 1)^2 + (0{,}5 - 0)^2 + (0{,}9 - 0)^2 = 1{,}86$
  - $D2 = (0{,}8 - 1)^2 + (0{,}4 - 1)^2 + (0{,}7 - 0)^2 + (0{,}3 - 0)^2 = 0{,}98$
- Assim, o vetor *a* está mais próximo do nó 2, que é, portanto, o nó vencedor.

Exemplos

- Atualizando os pesos do nó vencedor, temos:
  - $W12(t) = W12(t-1) + 0,6(1-0,8) = 0,8 + 0,12 = 0,92$
  - $W22(t) = W22(t-1) + 0,6(1-0,4) = 0,4 + 0,36 = 0,76$
  - $W32(t) = W32(t-1) + 0,6(0-0,7) = 0,7 - 0,42 = 0,28$
  - $W42(t) = W32(t-1) + 0,6(0-0,3) = 0,3 - 0,18 = 0,12$
- Os novos valores dos pesos, são, portanto:
  - $W1 = (0,2\ 0,6\ 0,5\ 0,9)$ e $W2 = (0,92\ 0,74\ 0,28\ 0,12)$
- Para o vetor b=(0,0,0,1), temos:
  - $D1 = (0,2 – 0)^2 + (0,6 – 0)^2 + (0,5 – 0)^2 + (0,9 – 1)^2 = 0,66$
  - $D2 = (0,92 – 0)^2 + (0,76 – 0)^2 + (0,28 – 0)^2 + (0,12 – 1)^2 = 2,27$

Exemplos

- Para este caso, o vetor *b* está mais próximo do nó 1, que é, portanto, o nó vencedor.
- Atualizando os pesos do nó vencedor:
  - W11(t) = W11(t-1) + 0,6(0-0,2) = 0,2 - 0,12 = 0,08
  - W21(t) = W21(t-1) + 0,6(0-0,6) = 0,6 - 0,36 = 0,24
  - W31(t) = W31(t-1) + 0,6(0-0,5) = 0,5 - 0,30 = 0,20
  - W41(t) = W31(t-1) + 0,6(1-0,9) = 0,9 + 0,06 = 0,96
- Os novos valores dos pesos, são, portanto:
  - W1 = (0,08 0,24 0,20 0,96) e W2 = (0,92 0,74 0,28 0,12)

Exemplos

- Para o vetor c=(1,0,0,0), temos:
  - D1 = 1,86
  - D2 = 0,67
- O nó 2 será será o vencedor e os novos pesos serão:
  - W1=(0,08 0,24 0,20 0,96) e W2=(0,968 0,304 0,112 0,048)
- Para o vetor d=(0,0,1,1), temos:
  - D1 = 0,70
  - D2 = 2,72
- O nó 1 será o vencedor e os novos pesos serão:
  - W1=(0,032 0,096 0,680 0,984) e
  - W2=(0,968 0,304 0,112 0,048)