

# Self Organized Map (SOM)

Hevelyn Sthefany Lima de Carvalho 170059031

Guilherme Coelho Minervino 160123046



# Roadmap

História

1

Redes neurais no SOMs

3

Implementações e suas  
vantagens e desvantagens

5

Redes Neurais

2

Funcionamento do  
SOMs

4

Aplicações

6

# História

- ◆ Teuvo Kohonen (Mapa de Kohonen)
- ◆ Não supervisionada
- ◆ Técnica usada para reduzir dados de dimensão alta
- ◆ Rede neural artificial
- ◆ Aprendizado competitivo



# Redes Neurais Artificiais

São redes computacionais semelhante à rede neural do cérebro humano

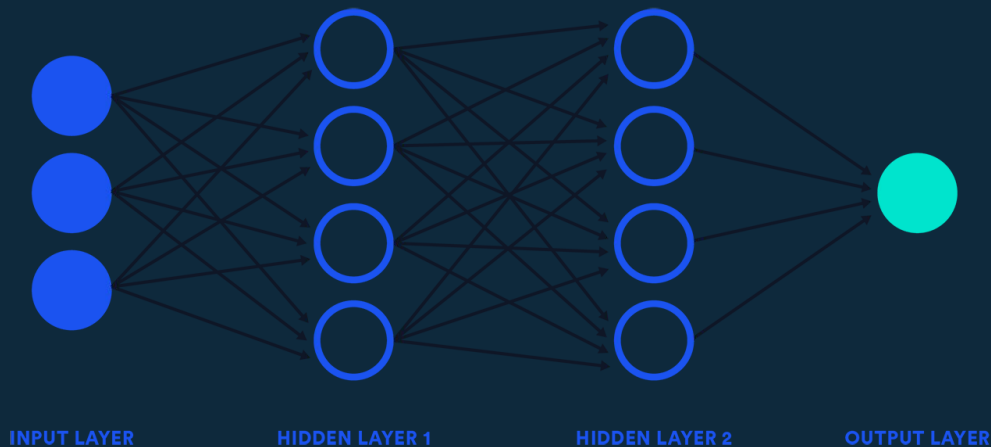
Permite que programas de computador reconheçam padrões e resolvam problemas comuns nas áreas de IA, aprendizado de máquina e aprendizado profundo

O computador aprende a realizar alguma tarefa analisando exemplos de treinamento



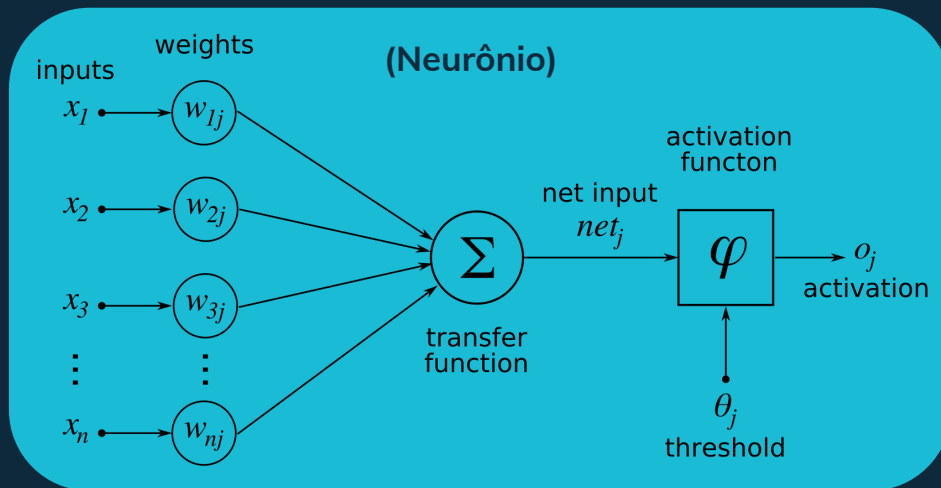
# Redes Neurais Artificiais

São compostas por camadas de nós, contendo uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída

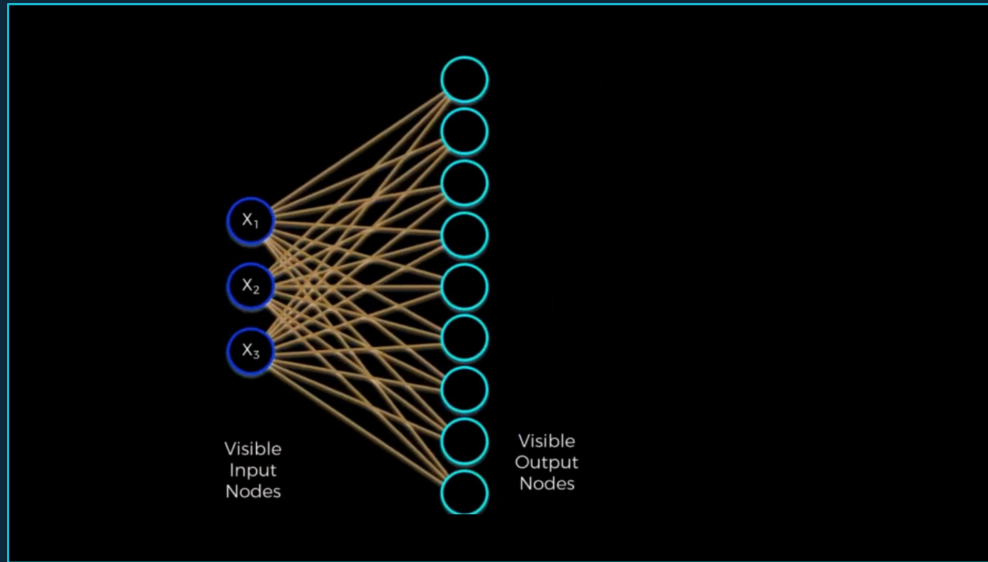


# Redes Neurais Artificiais

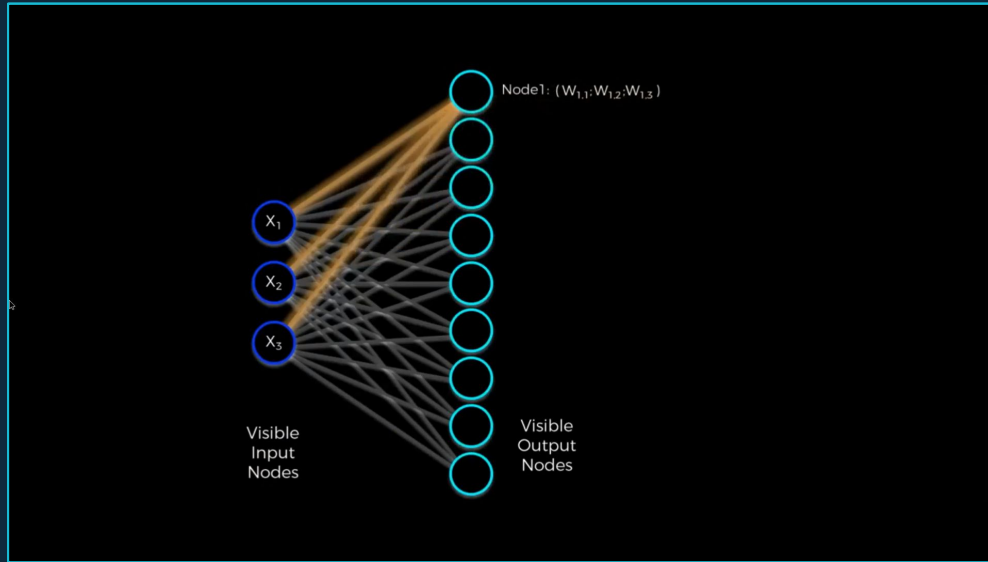
Cada neurônio é composto de dados de entrada, pesos, bias (ou limite) e uma função de ativação.



# Self Organizing Maps (SOMs)



# Self Organizing Maps (SOMs)





# Como o SOM funciona?





# Inicialização

## Aleatória

- ◇ Os pesos dos nós serão iniciados com valores aleatórios
- ◇ A dimensão dos nós será a mesma dos dados de entrada
- ◇ É como incluir novos dados no espaço original das instâncias

Recomenda-se normalizar os dados de entrada para então gerar os pesos aleatórios no intervalo adequado





# Inicialização

Com PCA

- ◇ É como um atalho para aproximar o mapa do estado final. Isso economiza muitos cálculos
- ◇ Os primeiros dois componentes principais são usados para definir os vetores de peso na direção dos componentes
- ◇ Os pesos são definidos em uma faixa que corresponde à escala dos componentes principais

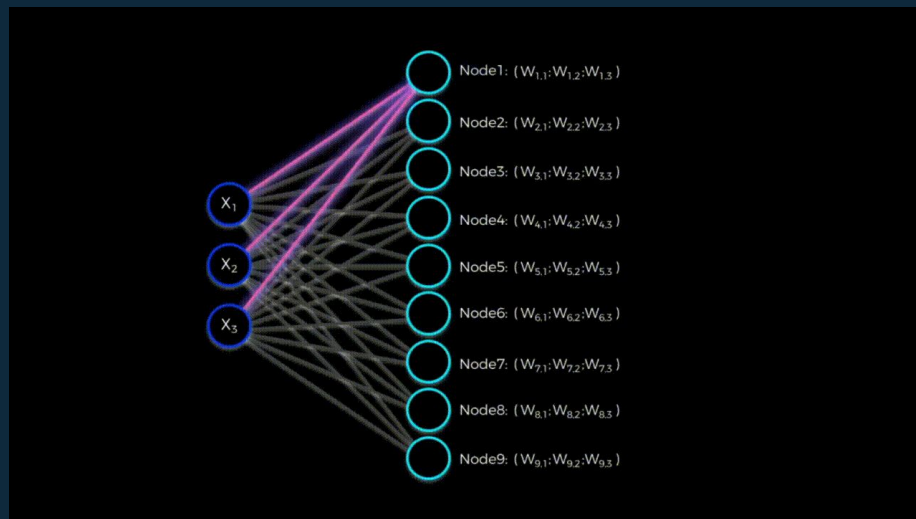
Seja o SOM um mapa  $m \times n$ ,  $a_1$  e  $a_2$  dois vetores auxiliares com pesos aleatórios,  $e_1$  e  $e_2$  os dois componentes principais. Logo:

$$\text{peso}[m, n] = a_1[m] * e_1 + a_2[n] * e_2$$



# Treinamento

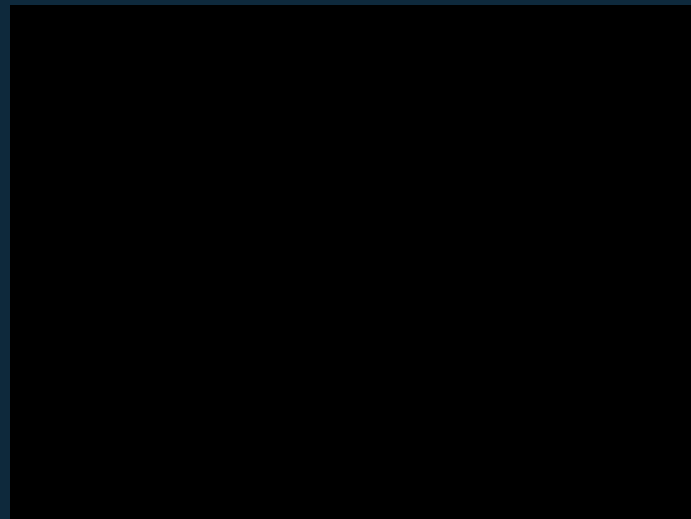
- ◇ Uma instância é escolhido aleatoriamente
- ◇ A distância euclidiana do vetor do dado de entrada para todos os vetores de peso é calculada





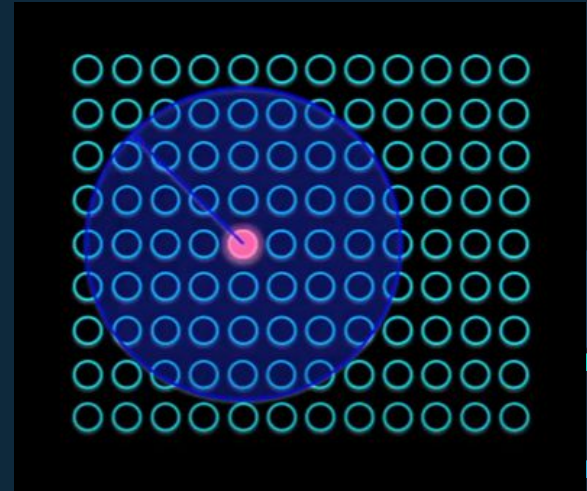
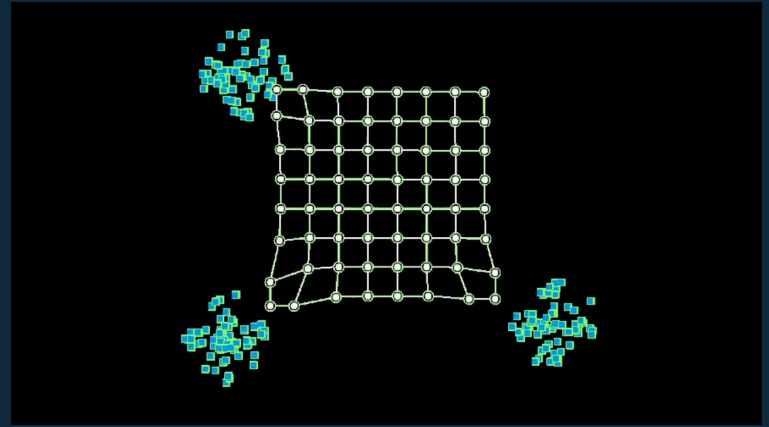
# Treinamento

- ◇ Um dado de entrada é escolhido aleatoriamente
- ◇ A distância euclidiana do vetor do dado de entrada para todos os vetores de peso é calculada
- ◇ Em seguida, será escolhido o nó com a menor distância de cada linha (BMU)
- ◇ Os vetores de peso dos nós na vizinhança do BMU (incluindo o próprio BMU) são atualizados puxando-os para mais perto do vetor de entrada



# Treinamento

- ◇ Um dado de entrada é escolhido aleatoriamente
- ◇ A distância euclidiana do vetor do dado de entrada para todos os vetores de peso é calculada
- ◇ Em seguida, será escolhido o nó com a menor distância de cada linha (BMU)
- ◇ Os vetores de peso dos nós na vizinhança do BMU (incluindo o próprio BMU) são atualizados puxando-os para mais perto do vetor de entrada



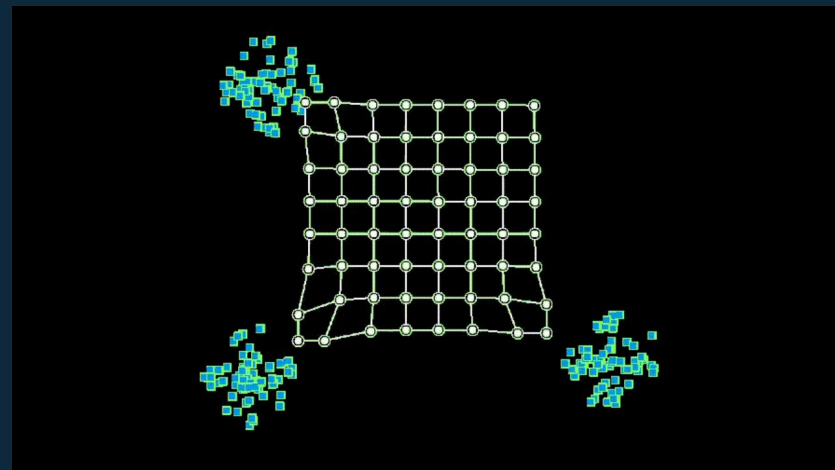
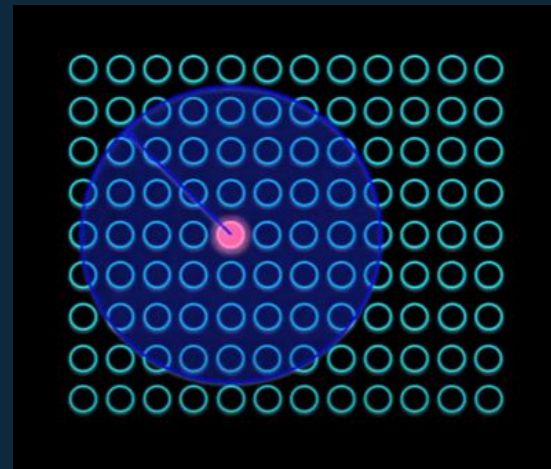
# Treinamento

- ◇ Atualização da taxa de aprendizado de cada nó

$$h_{ci}(t) = \alpha(t) \cdot \exp\left(-\frac{\|r_c - r_i\|^2}{2\sigma^2(t)}\right)$$

- ◇ Atualização dos pesos de cada nó

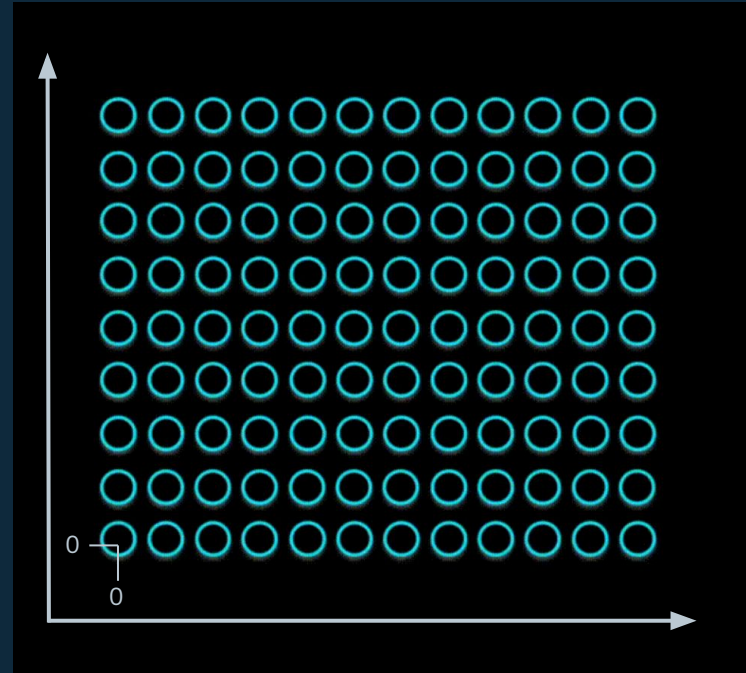
$$m_i(t+1) = m_i(t) + h_{ci}(t)[x(t) - m_i(t)]$$



# Espaço de recursos final

Suponha que  $\text{dado}_n$  pertence ao neurônio da posição  $[0, 0]$  no espaço do mapa. Portanto, os novos recursos do  $\text{dado}_n$ , ou seja, sua dimensão reduzida, será:

$$\text{Dado}_n: [a_0, a_1, a_2, a_3, a_4] \xrightarrow{\text{(SOM)}} [0, 0]$$





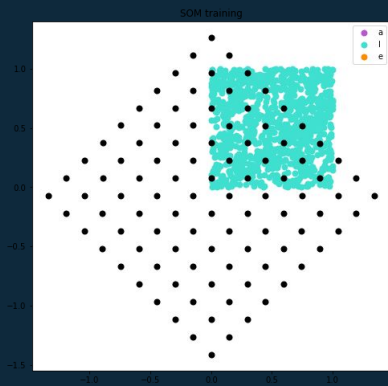


# Implementação

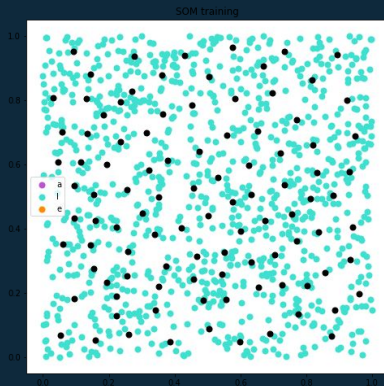


# Com PCA x Sem PCA

Com dados artificiais gerados de forma aleatória e 20 épocas

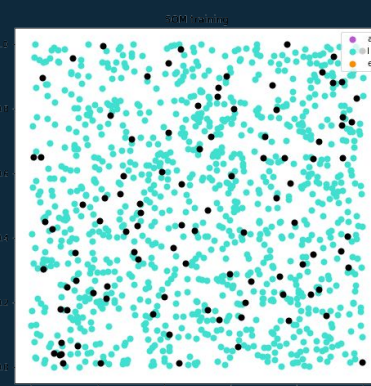


Mapa inicial

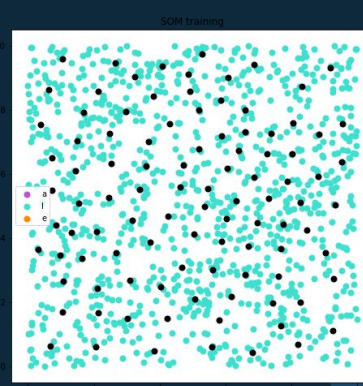


Mapa final

Com PCA



Mapa inicial



Mapa final

Sem PCA



# Aplicações

- ◇ Compressão de Dados
- ◇ Clusterização

Exemplos:

- ◇ Reconhecimento de Voz
- ◇ Compressão de imagens
- ◇ Classificação de Padrões em geral





# Referências

## ◇ Teoria

[Kohonen, Teuvo. "Self-organized formation of topologically correct feature maps." \*Biological cybernetics\* 43.1 \(1982\): 59-69](#)

<http://www.ijmo.org/vol6/504-M08.pdf>

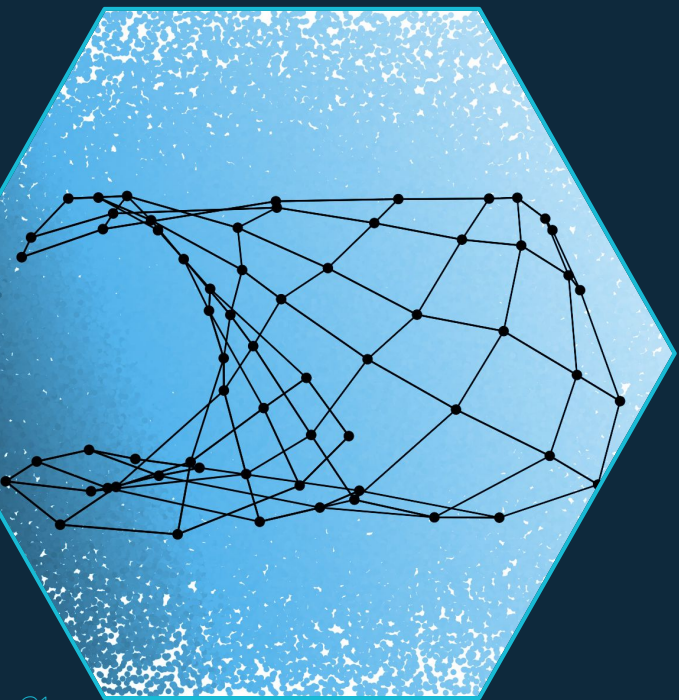
[https://www.youtube.com/watch?v=JscwdkDXtuY&ab\\_channel=hailtonjr](https://www.youtube.com/watch?v=JscwdkDXtuY&ab_channel=hailtonjr)

[https://www.youtube.com/watch?v=mShTVp6UkvQ&ab\\_channel=BeatrizFilippa](https://www.youtube.com/watch?v=mShTVp6UkvQ&ab_channel=BeatrizFilippa)

## ◇ Implementação

<https://towardsdatascience.com/how-to-implement-kohonnens-self-organizing-maps-989c4da05f19>





Obrigada

