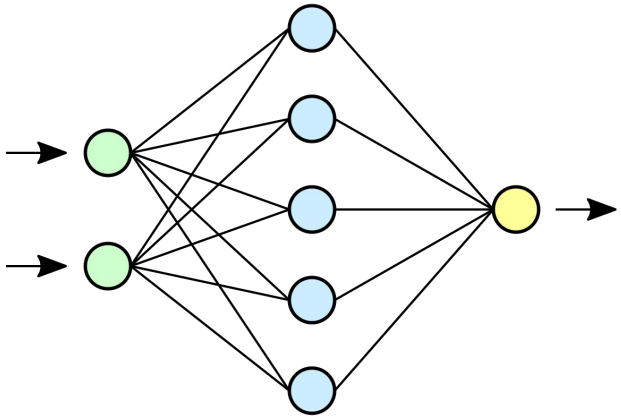
Sistemas Operacionais Embarcados

Redes Neurais

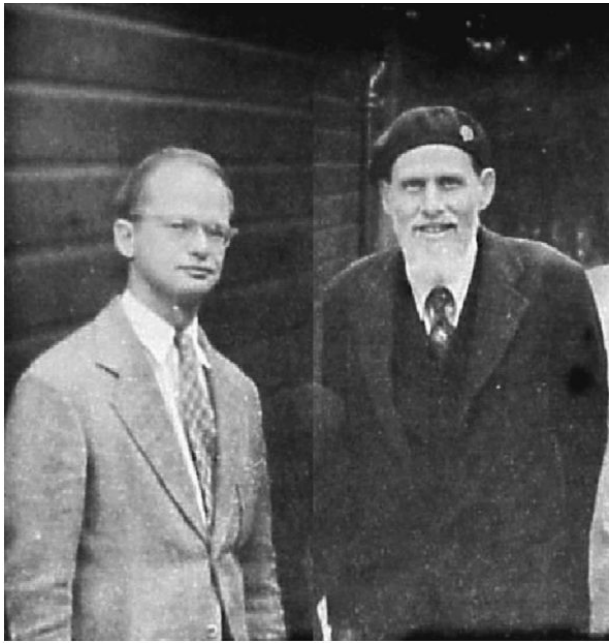
Redes neurais

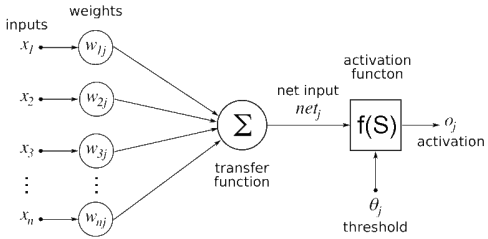
- No desenvolvimento tradicional de *software*, o programador define regras e as implementa em código (dedução)
- No desenvolvimento de *software* baseado em redes neurais, o programador as utiliza para que a máquina calcule suas regras (indução)
- Em ambos os casos, usamos as regras definidas para novos dados



Redes neurais

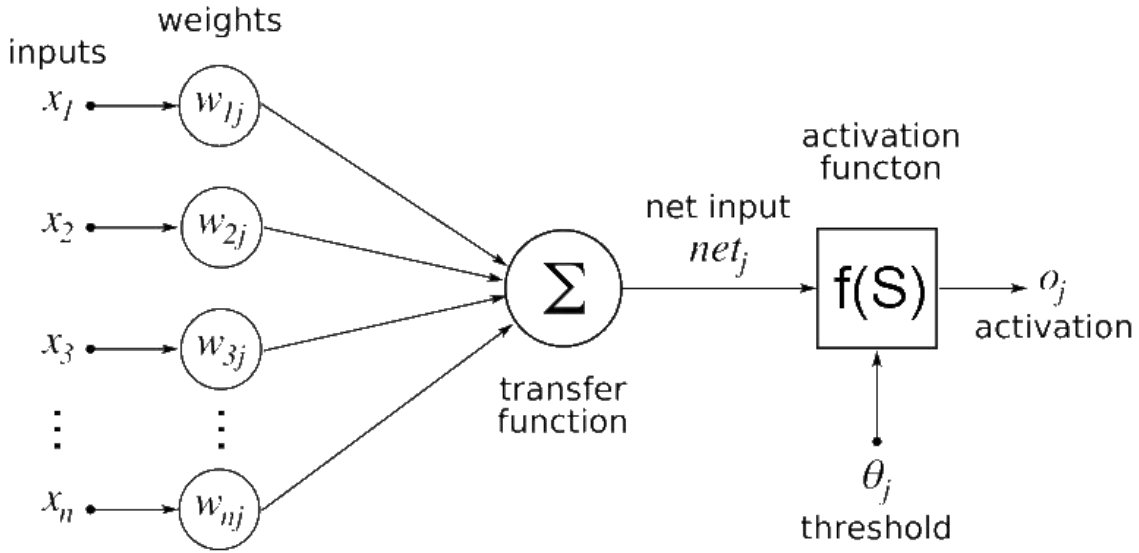
- Modelos computacionais *inspirados* no funcionamento do cérebro humano.
- Compostas por neurônios artificiais, podem “aprender” a partir de dados para realizar tarefas específicas.
- Termo criado por Warren McCulloch e Walter Pitts em 1943, em um artigo que descreveu o funcionamento de neurônios artificiais.





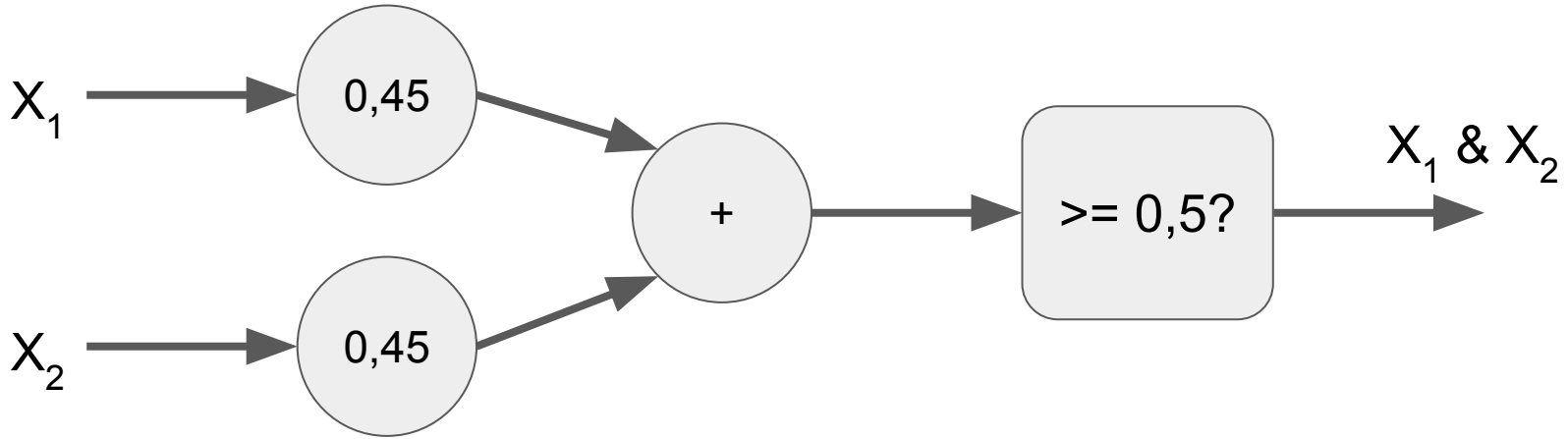
Neurônio artificial

- Recebe entradas com pesos associados
- Realiza soma ponderada
- Aplica função de ativação, introduzindo não-linearidades ao modelo.



Neurônio artificial

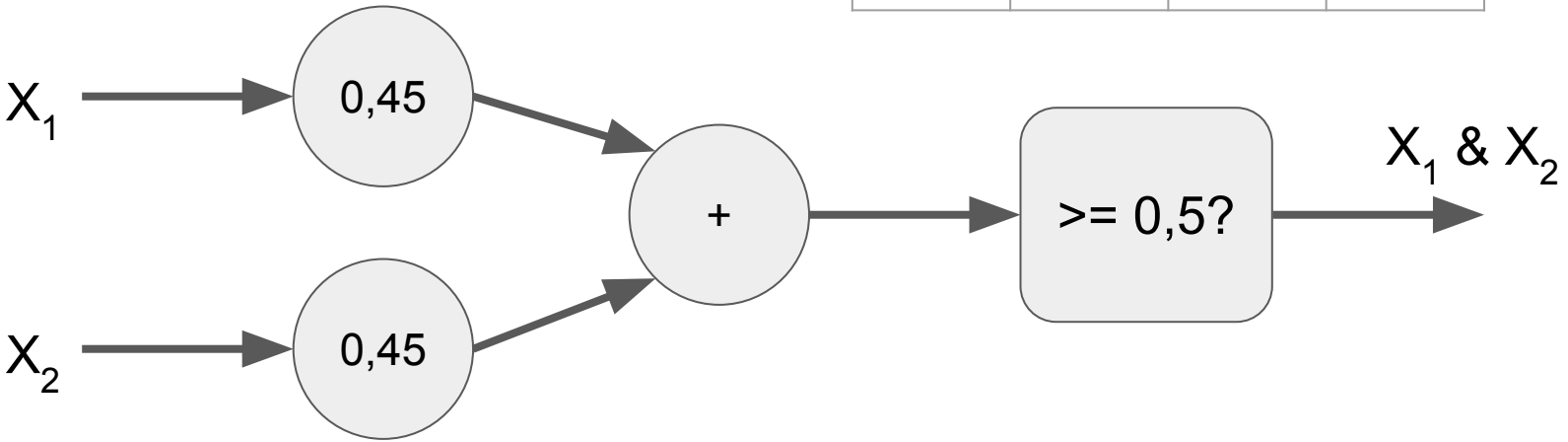
- Exemplo: porta AND



Neurônio artificial

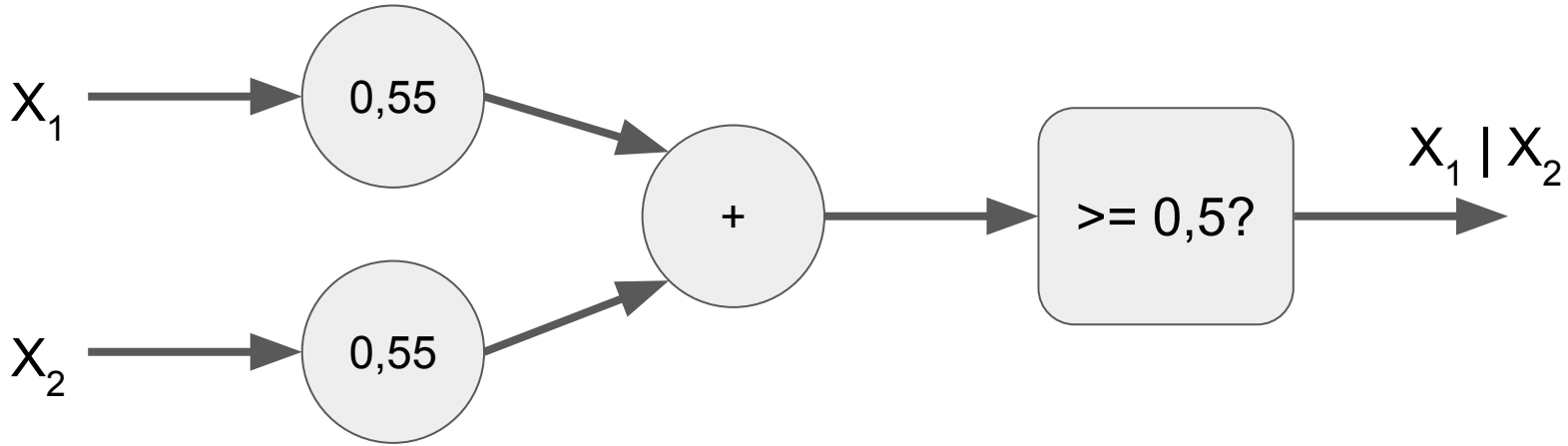
- Exemplo: porta AND

X_1	X_2	Soma	Saída
0	0	0	0
0	1	0,45	0
1	0	0,45	0
1	1	0,90	1



Neurônio artificial

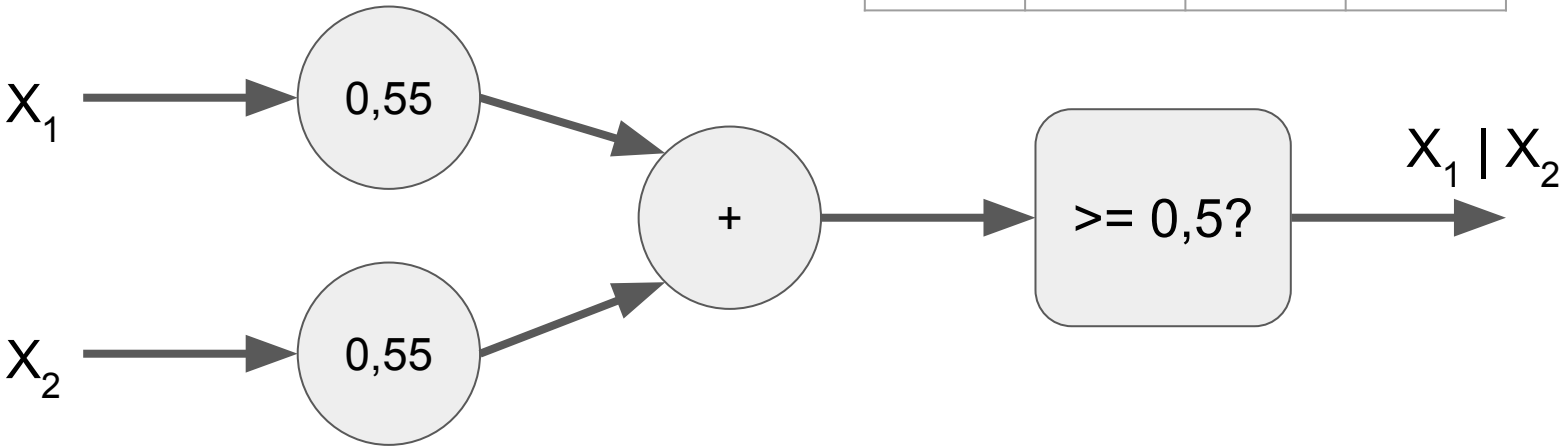
- Exemplo: porta OR



Neurônio artificial

- Exemplo: porta OR

X_1	X_2	Soma	Saída
0	0	0	0
0	1	0,55	1
1	0	0,55	1
1	1	1,10	1



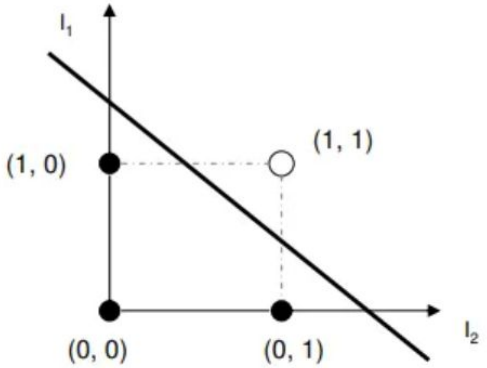
Neurônio artificial

- Exemplo: porta OR

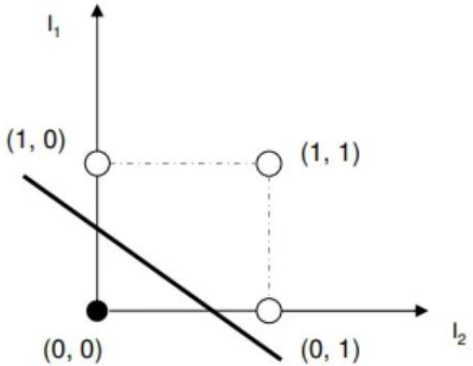
X_1	X_2	Soma	Saída
0	0	0	0
0	1	0,55	1
1	0	0,55	1
1	1	1,10	1

XOR é impossível usando somente este neurônio!

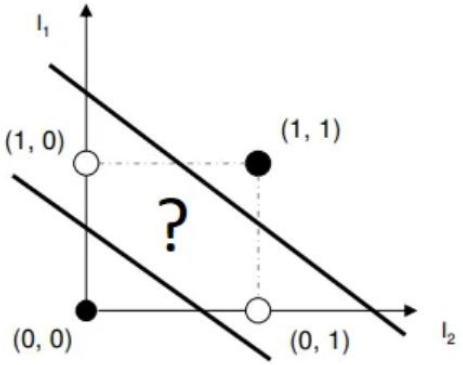
AND		
I_1	I_2	out
0	0	0
0	1	0
1	0	0
1	1	1

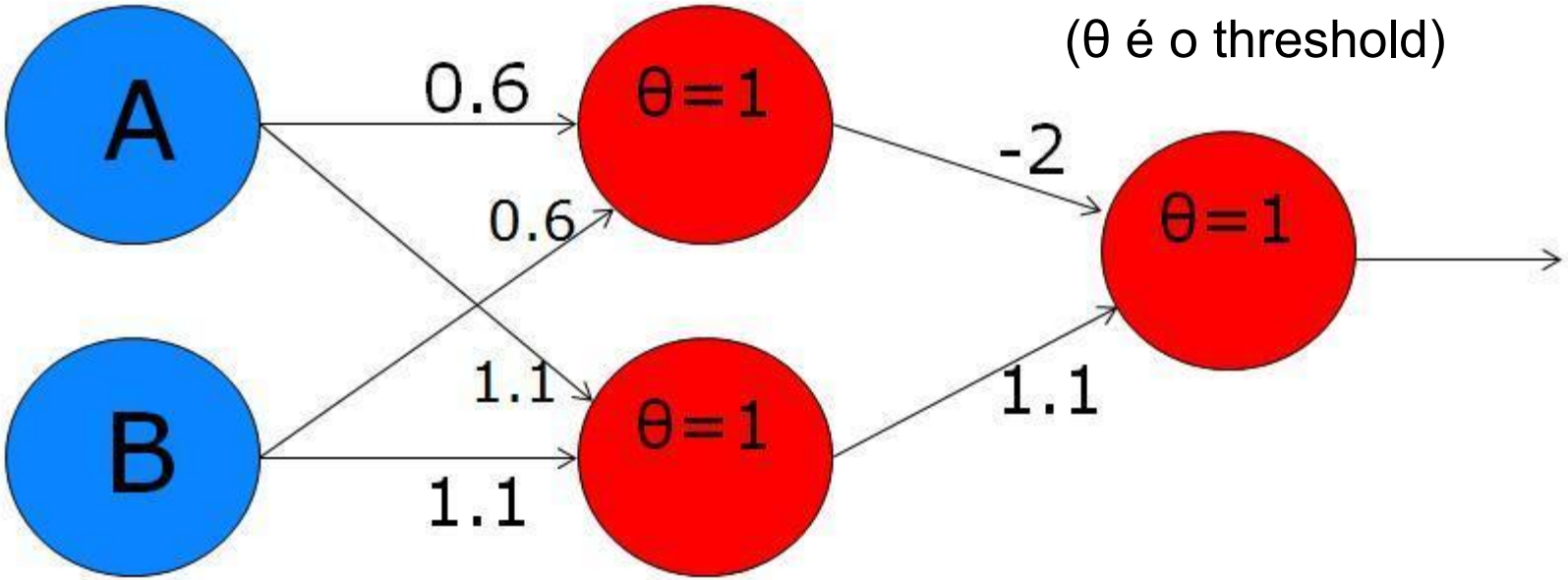


OR		
I_1	I_2	out
0	0	0
0	1	1
1	0	1
1	1	1

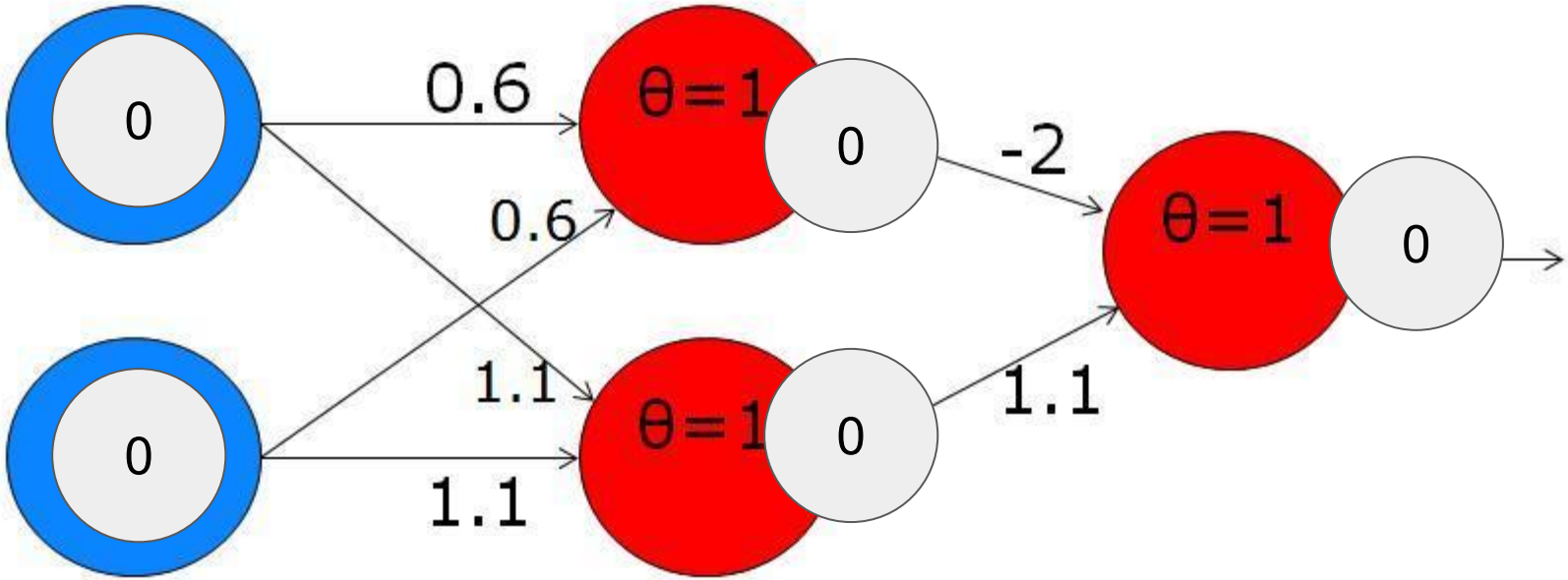


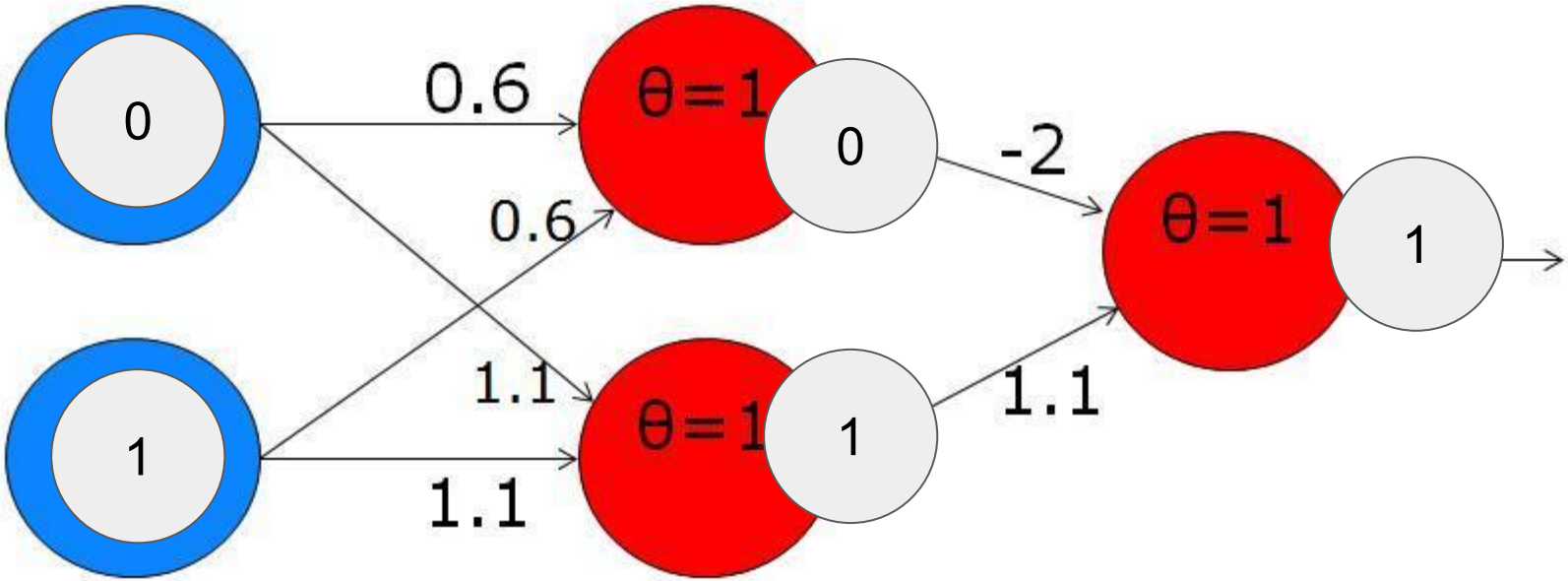
XOR		
I_1	I_2	out
0	0	0
0	1	1
1	0	1
1	1	0

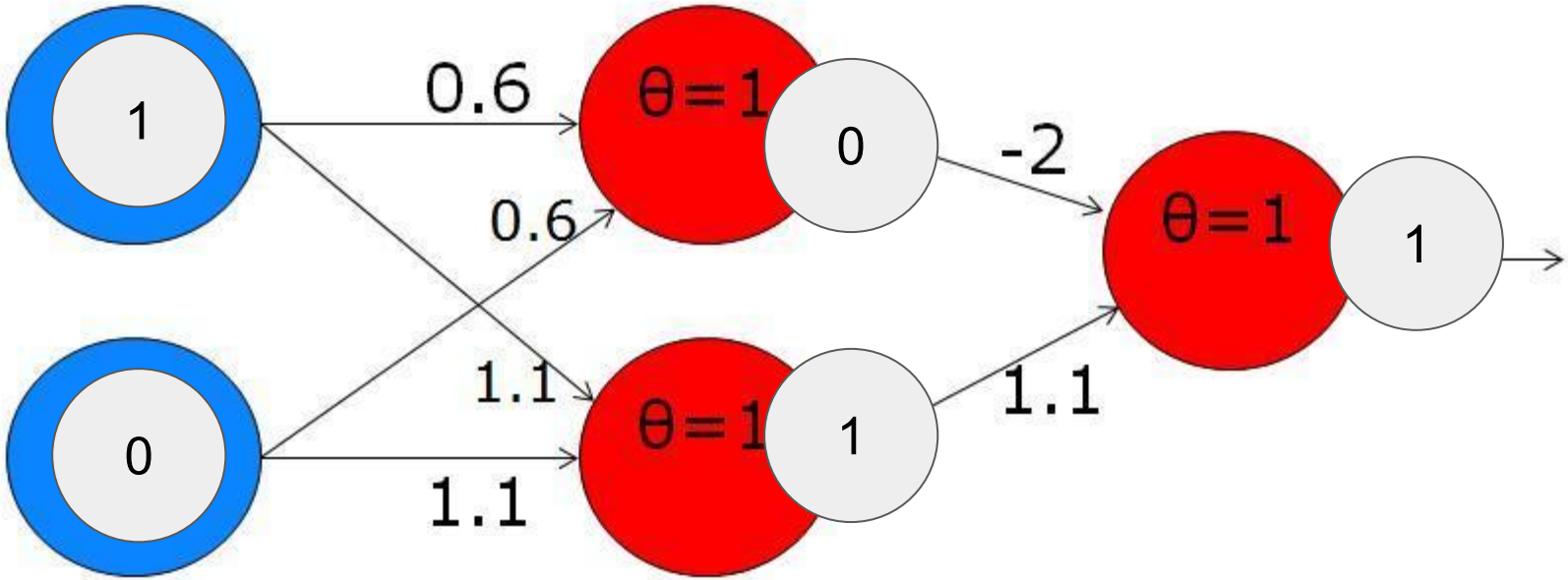


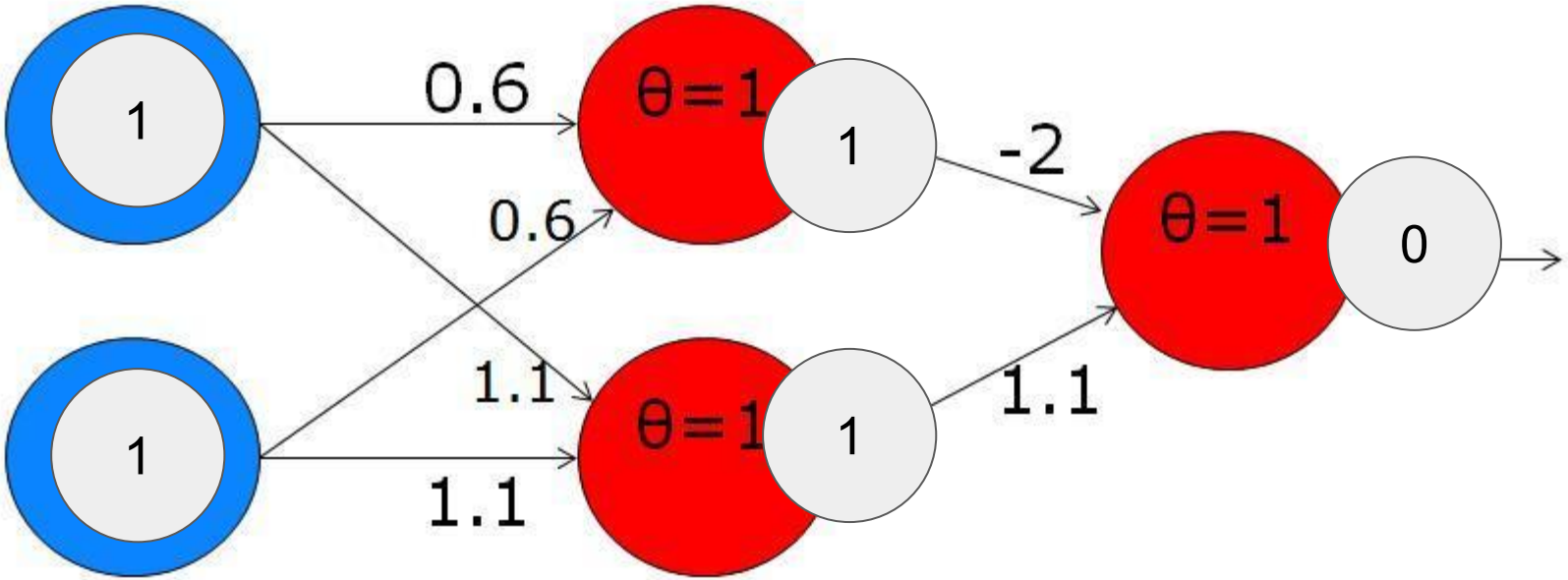


Solução para a XOR:
múltiplos neurônios


0
0
0.6
0.6
1.1
1.1
θ=1
θ=1
0
0
-2
θ=1
0
1.1

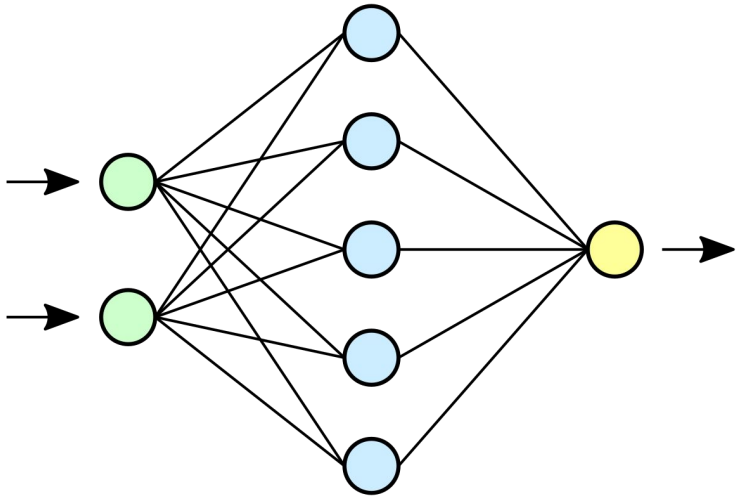

0
1
0.6
0.6
1.1
1.1
θ=1
θ=1
0
1
-2
1.1
θ=1
1


1
0
0.6
0.6
1.1
1.1
θ=1
θ=1
0
1
-2
θ=1
1
1.1


1
1
0.6
0.6
1.1
1.1
θ=1
θ=1
1
1
-2
1.1
θ=1
0

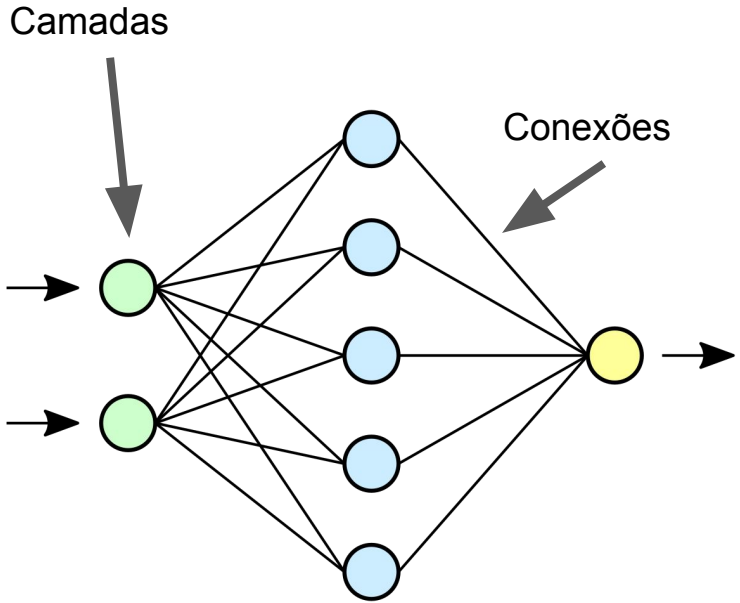
Arquitetura

- Combinando diversas camadas (***deep learning***), é possível resolver problemas muito mais complexos



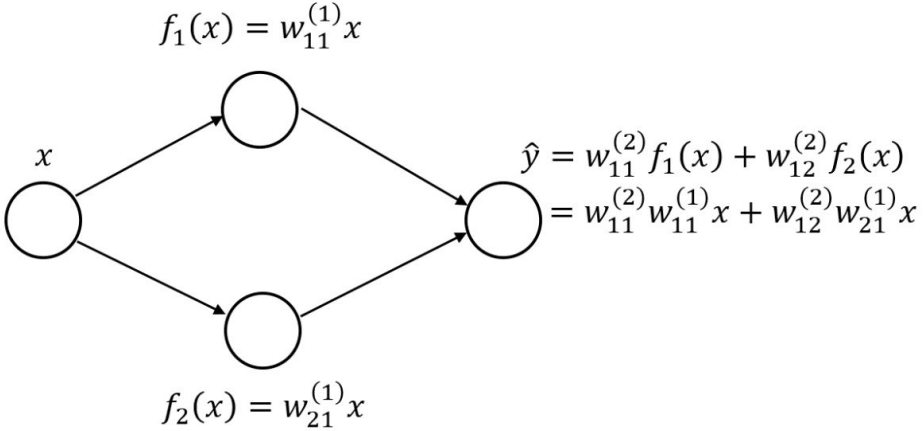
Arquitetura

- **Camadas:** As redes neurais são organizadas em camadas, sendo as principais a camada de entrada, uma ou mais camadas ocultas e a camada de saída. Cada camada contém um conjunto de neurônios.
- **Conexões:** Cada neurônio em uma camada está conectado a todos os neurônios da camada seguinte por meio de conexões ponderadas, representando os pesos sinápticos.



Funções de ativação

- Sem as funções de ativação, a combinação de neurônios resultaria em uma combinação linear das entradas (combinar mais neurônios somente mudaria os pesos)

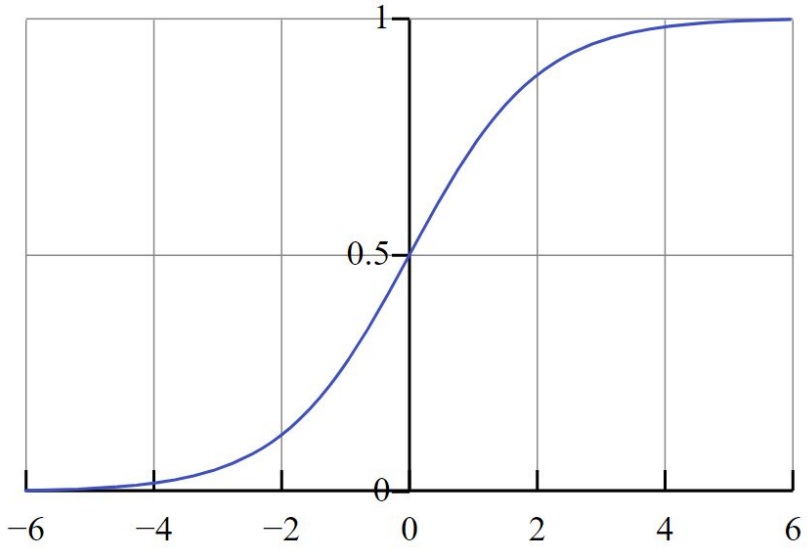


- As funções de ativação inserem não-linearidades às redes, permitindo a criação de funções mais complexas

Funções de ativação

- **Sigmóide:** mapeia valores para o intervalo (0, 1), sendo útil em problemas de classificação binária.

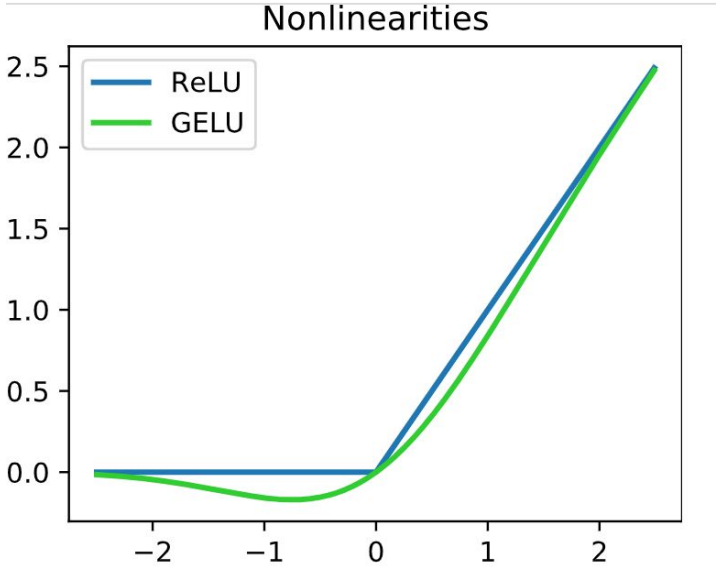
$$\sigma(x) = \frac{1}{1 + e^{-x}}$$



Funções de ativação

- **ReLU (Rectified Linear Unit):** A função ReLU retorna 0 para valores negativos e o valor original para valores positivos, ajudando a resolver problemas de classificação.

$$f(x) = x^+ = \max(0, x) = \frac{x + |x|}{2}$$



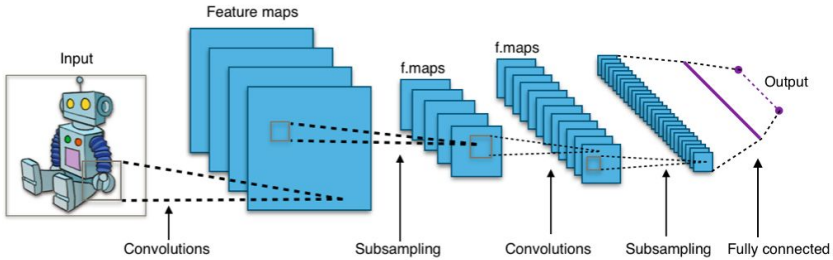
Funções de ativação

- **Softmax:** recebe um vetor *z* com *K* números, e os normaliza de forma proporcional ao exponencial destes valores.
  - Frequentemente usada em problemas de classificação multiclasse

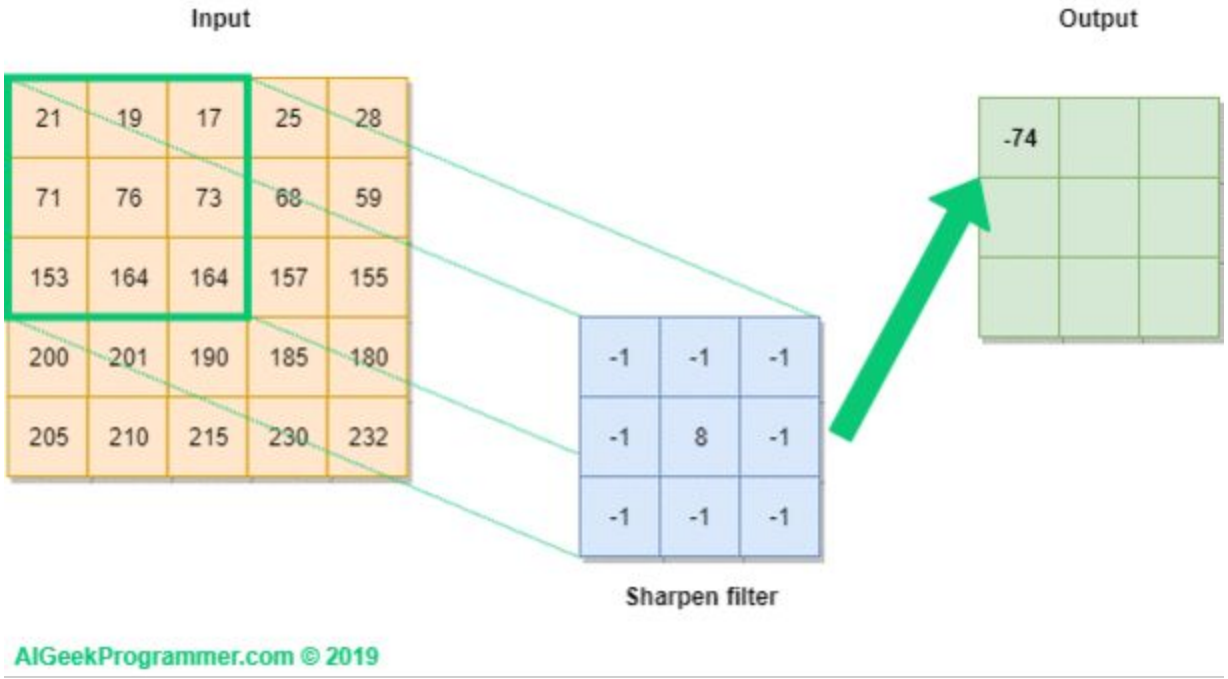
$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad \text{for } i = 1, \ldots, K \text{ and } \mathbf{z} = (z_1, \ldots, z_K) \in \mathbb{R}^K.$$

Redes neurais convolucionais

- Usam camadas convolucionais além dos neurônios tradicionais (aqui chamadas de camadas totalmente conectadas)
- Substitui o uso de todas as entradas da camada anterior multiplicadas por pesos
- Cada camada convolucional aplica filtros convolucionais em pequenas regiões (janelas) da entrada, deslizando-os sobre toda a imagem para extrair características relevantes

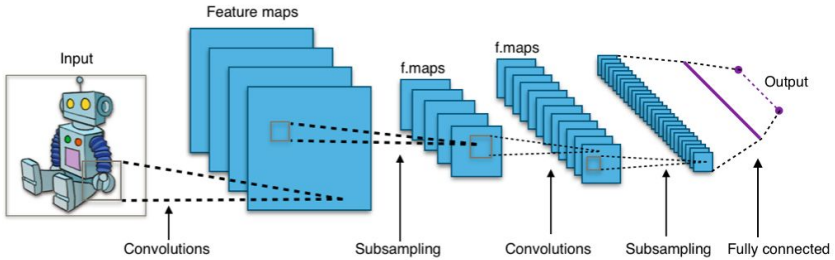


Redes neurais convolucionais

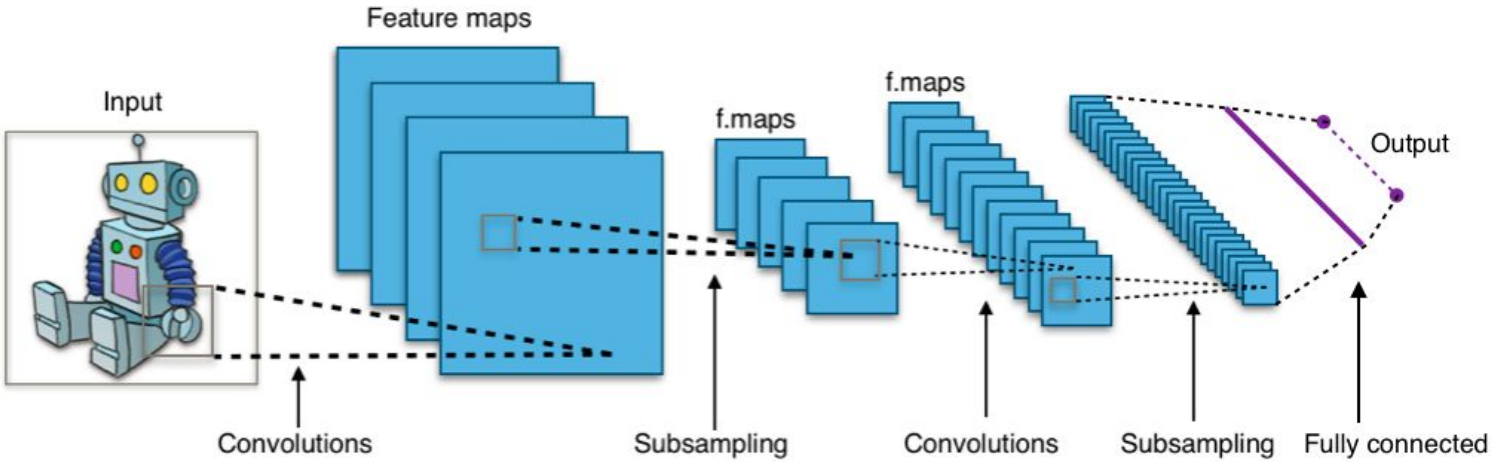


Redes neurais convolucionais

- Também usam camadas de *pooling*, que reduzem a dimensionalidade espacial das representações geradas pelas camadas convolucionais;
- O pooling realiza uma operação de resumo (como máximo ou média) em pequenas regiões da entrada, reduzindo o tamanho da representação;
- *Transfer learning:* Modelos pré-treinados em grandes conjuntos de dados podem ser usados como ponto de partida para resolver outras tarefas, permitindo uma inicialização mais rápida e melhores resultados com menos dados de treinamento.

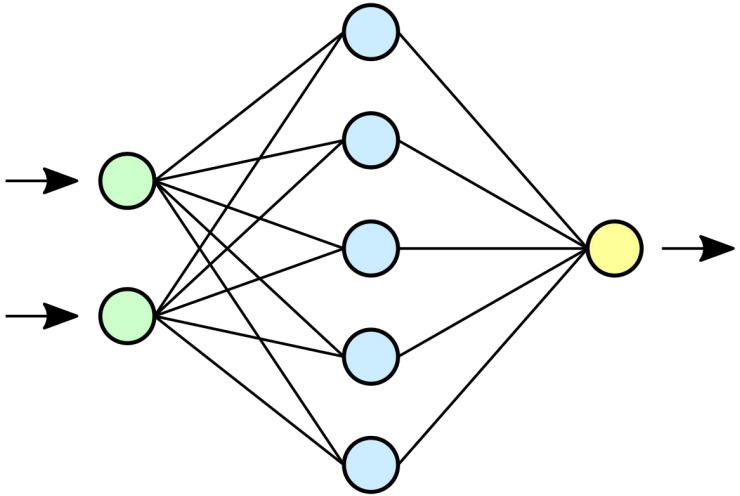


Redes neurais convolucionais



Treinamento de redes

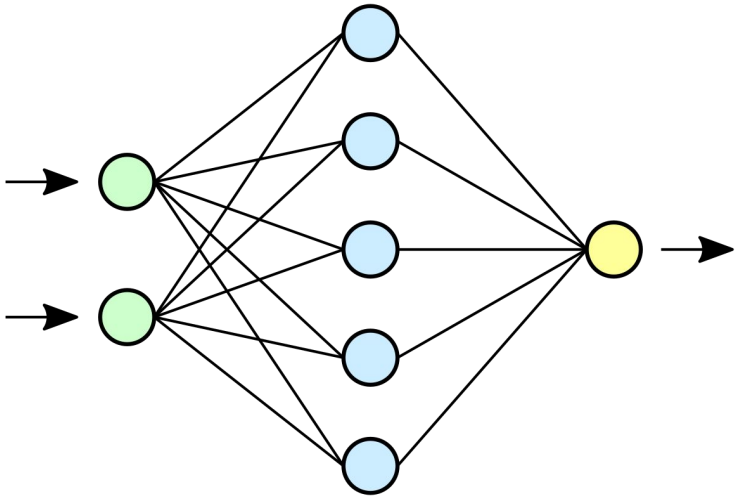
- Como calcular os pesos?
- Como escolher a quantidade de camadas e de neurônios?
- Como escolher as funções de ativação?
- Como escolher a função de custo?



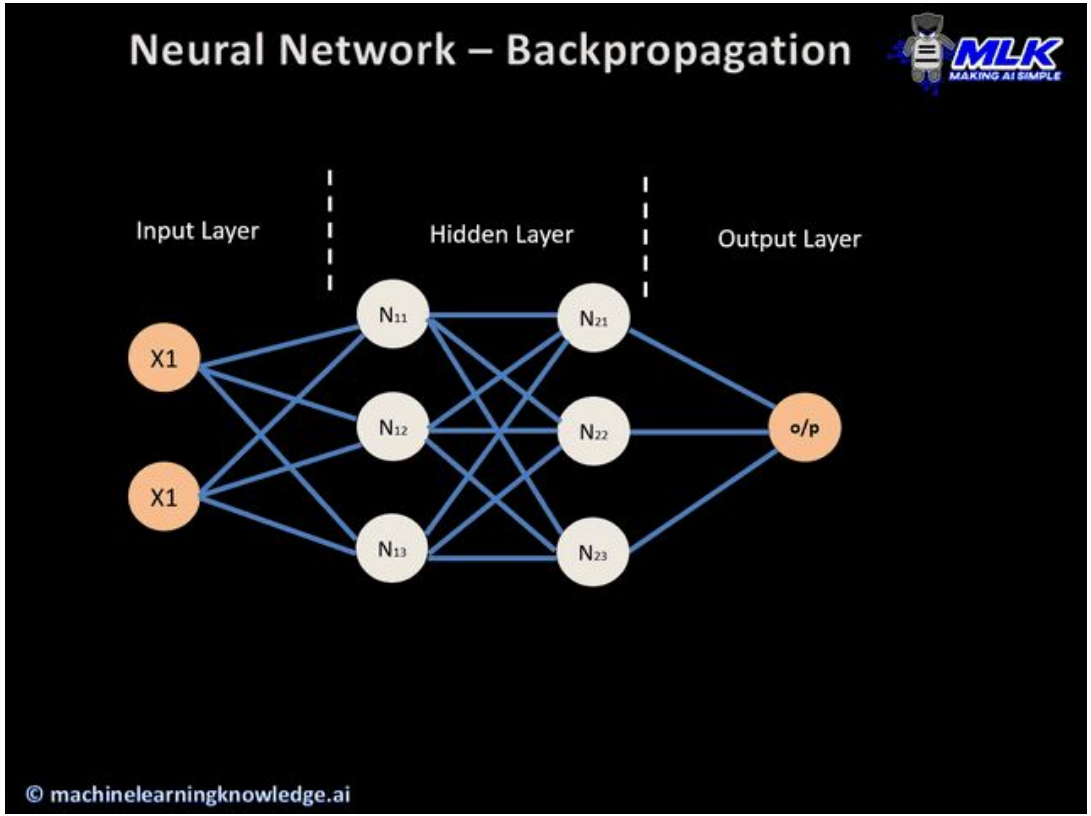
Treinamento de redes

Treinamento supervisionado:

- Usamos um conjunto de dados de entrada e suas respostas desejadas.
- Ajustamos os pesos iterativamente, de forma a minimizar o erro entre as saídas previstas pela rede e as saídas desejadas para esses dados (*backpropagation by gradient descent*).
- A função de custo é o erro entre as saídas previstas e desejadas.



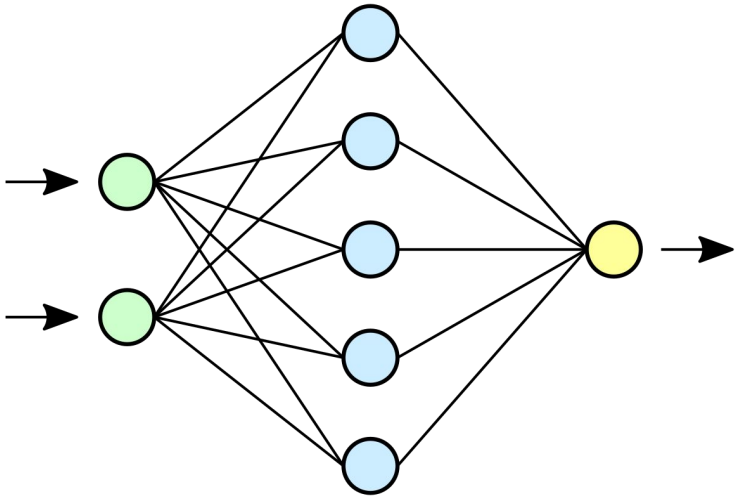
Treinamento de redes



Treinamento de redes

Taxa de aprendizado (*learning rate*):

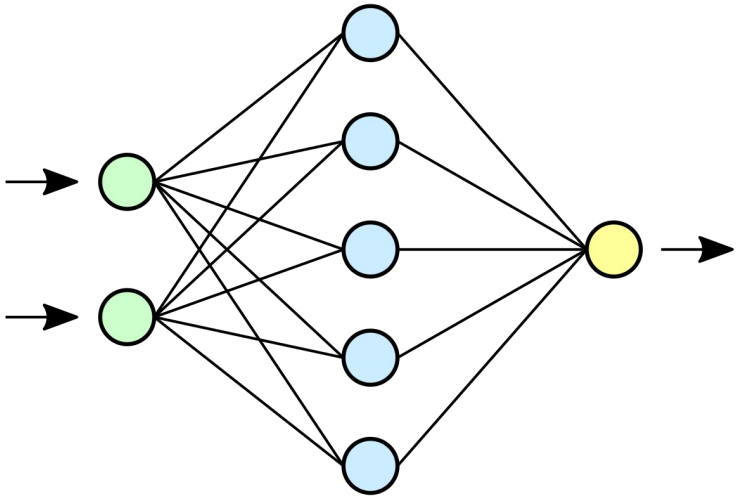
- Hiperparâmetro crítico que determina o tamanho dos passos que o algoritmo de otimização dá para atualizar os pesos.
- Uma taxa de aprendizado muito alta pode levar a oscilações e dificultar a convergência
- Uma taxa de aprendizado muito baixa pode resultar em um treinamento lento e em convergência para mínimos locais.



Treinamento de redes

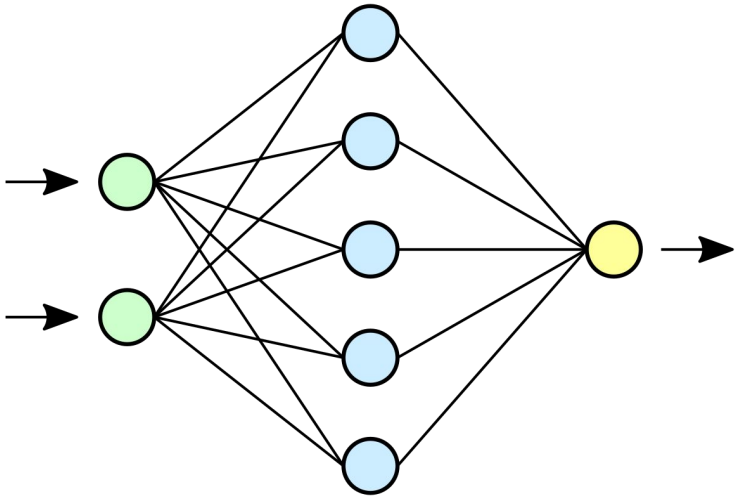
Épocas e tamanho do lote (*batch size*):

- O treinamento ocorre em etapas chamadas épocas, onde cada época consiste em uma passagem completa por todo o conjunto de treinamento.
- Durante cada época, os dados são divididos em lotes (*batches*) e a atualização dos pesos ocorre após processar cada lote.
- O tamanho do lote pode afetar a estabilidade e a velocidade do treinamento.



Treinamento de redes

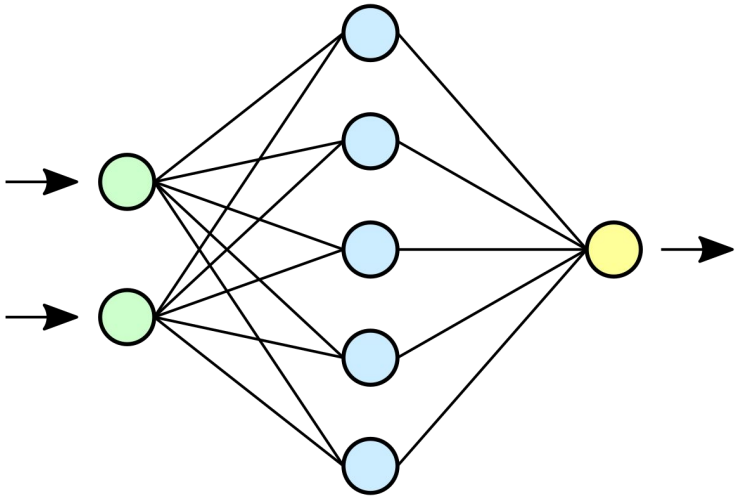
- ***Overfitting*:** ocorre quando a rede se torna muito ajustada aos dados de treinamento e tem dificuldades de generalização para novos dados.
  - Para evitar *overfitting*, são utilizadas técnicas de regularização, como a L2 *regularization* (*weight decay*) e o *dropout*, que evitam que os pesos da rede se tornem muito altos, e que reduzem a coadaptação de neurônios durante o treinamento.
- ***Underfitting*:** ocorre quando a rede não consegue capturar os padrões presentes nos dados de treinamento, resultando em um desempenho insuficiente.



Treinamento de redes

Definição da quantidade de camadas:

- Problema linear? 1 camada
- Não-linear? Experimentação + intuição
  - Aumente a complexidade gradativamente
  - Evite *overfitting*, quando a rede aprende muito bem os dados de treinamento, mas não generaliza bem para novos dados



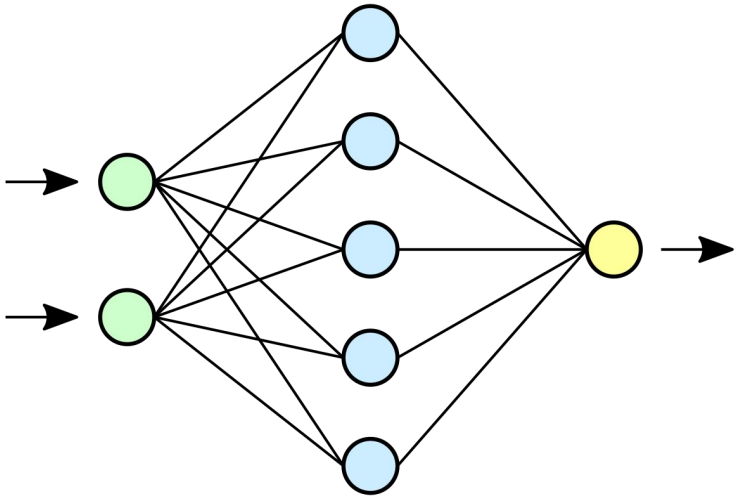
Definição da função de ativação:

- ReLU é a escolha padrão
- Sigmóide ou parecida para classificação binária
- Softmax para multiclasse

Treinamento de redes

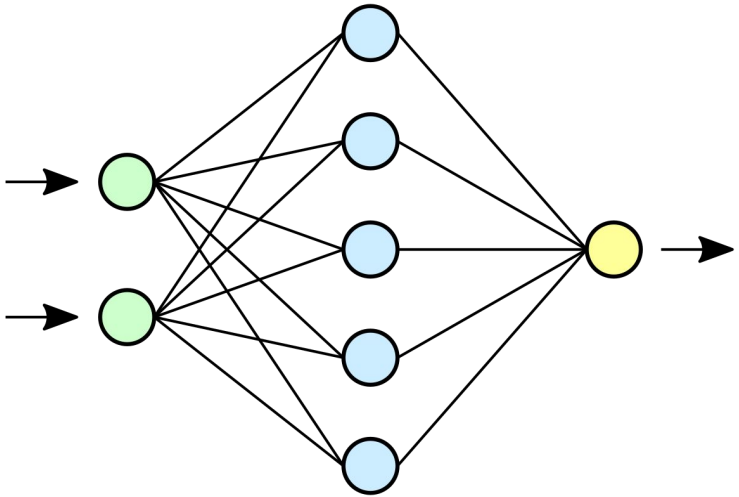
Definição da função de custo:

- **Erro Quadrático Médio (*Mean Squared Error* - MSE):** mais comum
- **Entropia Cruzada (*Cross-Entropy Loss*):** muito usada em problemas de classificação (categorização):
  - *Binary Cross-Entropy:* Classificação binária
  - *Categorical Cross-Entropy:* classificação multiclasse
- **Outras funções:** problemas especializados, como IoU (Intersection over Union) usada em tarefas de detecção de objetos para medir a sobreposição entre caixas delimitadoras.
- ***Transfer Learning:*** a função de custo pode ser escolhida de acordo com o domínio da tarefa original para a qual o modelo foi pré-treinado



Treinamento de redes

- **Experimente diferentes arquiteturas:** Não há uma regra definitiva para determinar a melhor arquitetura de antemão. Experimente diferentes números de camadas, tamanhos de camadas e arquiteturas para encontrar o melhor equilíbrio entre desempenho e eficiência.
- ***Grid Search* ou *Random Search*:** É possível utilizar técnicas de busca em grade (*Grid Search*) ou busca aleatória (*Random Search*) para explorar o espaço de hiperparâmetros e encontrar uma configuração ótima.



Avaliação de redes

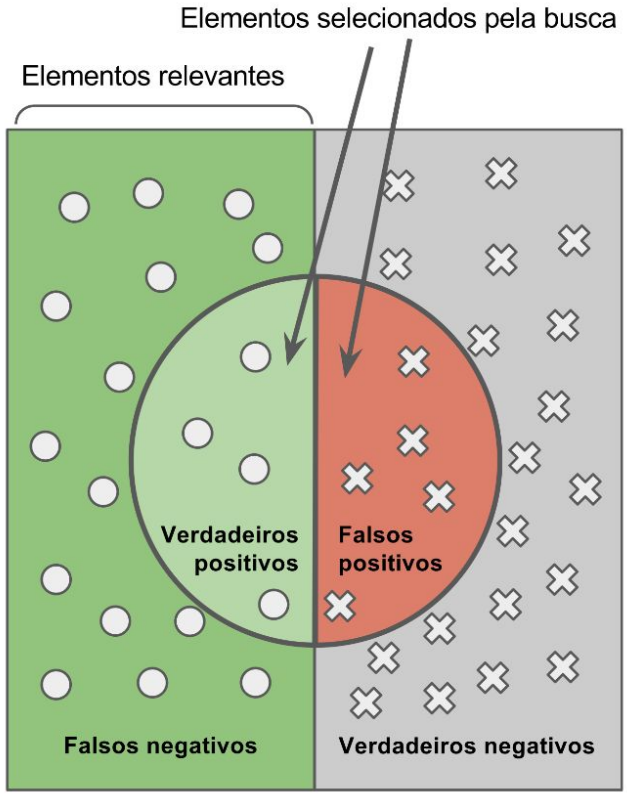
- **Separação dos dados em conjuntos de treinamento, validação e teste:**
  - **Treinamento:** para ajustar os pesos da rede;
  - **Validação:** para acompanhar o desempenho durante o treinamento e ajustar hiperparâmetros;
  - **Teste:** para avaliar o desempenho final da rede, após o treinamento ser concluído

- **Matriz de confusão:**
  - Contagem de previsões corretas e incorretas para cada classe, permitindo identificar os acertos e erros da rede

Avaliação de redes

- **Separação dos dado**
  treinamento, validaç
  - **Treinamento:** para aj
  - **Validação:** para acon
    durante o treinamento
  - **Teste:** para avaliar o
    após o treinamento se

- **Matriz de confusão:**
  - Contagem de previsõ
    para cada classe, per
    acertos e erros da rec



Avaliação de redes

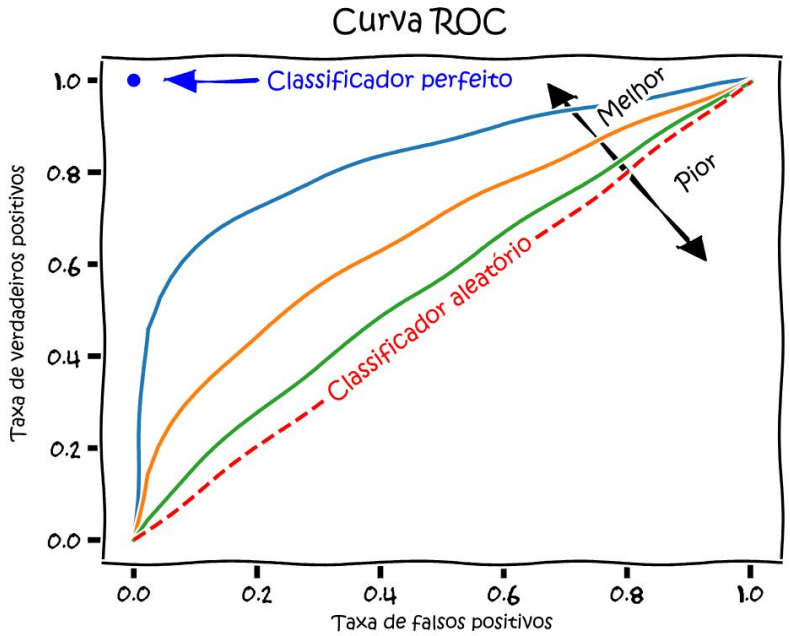
- **Separação dos c**
  treinamento, vali
  - **Treinamento:** pa
  - **Validação:** para
    durante o treinan
  - **Teste:** para avali
    após o treinamer

- **Matriz de confus**
  - Contagem de pre
    para cada classe
    acertos e erros d

Avaliação de redes

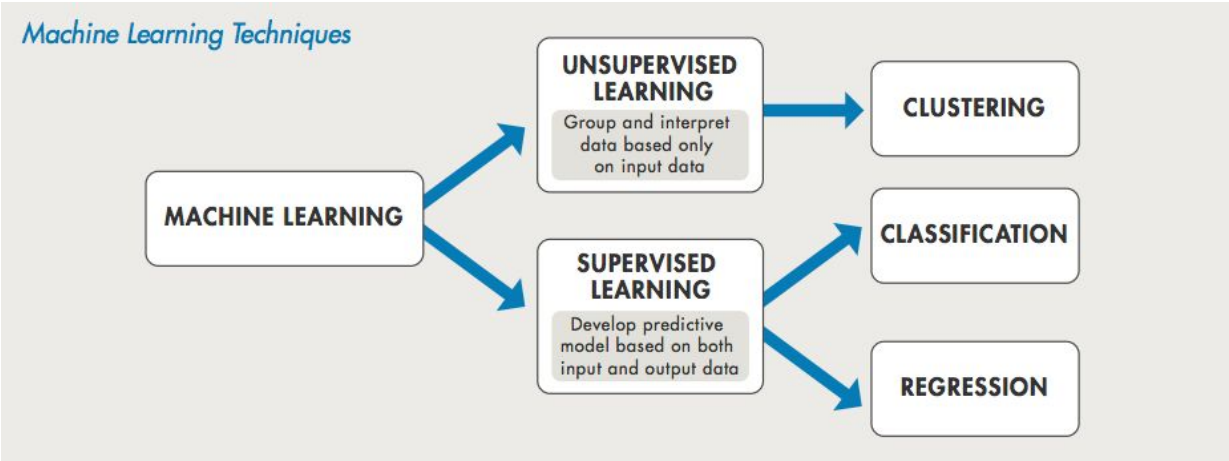
- **Curva ROC (*Receiver Operating Characteristic*):**
  - Avalia o desempenho de classificadores binários em diferentes pontos de corte

- **Área sob a curva ROC (*AUC*):**
  - *R*esume a curva ROC em um único valor, indicando a capacidade do modelo de distinguir entre as classes.



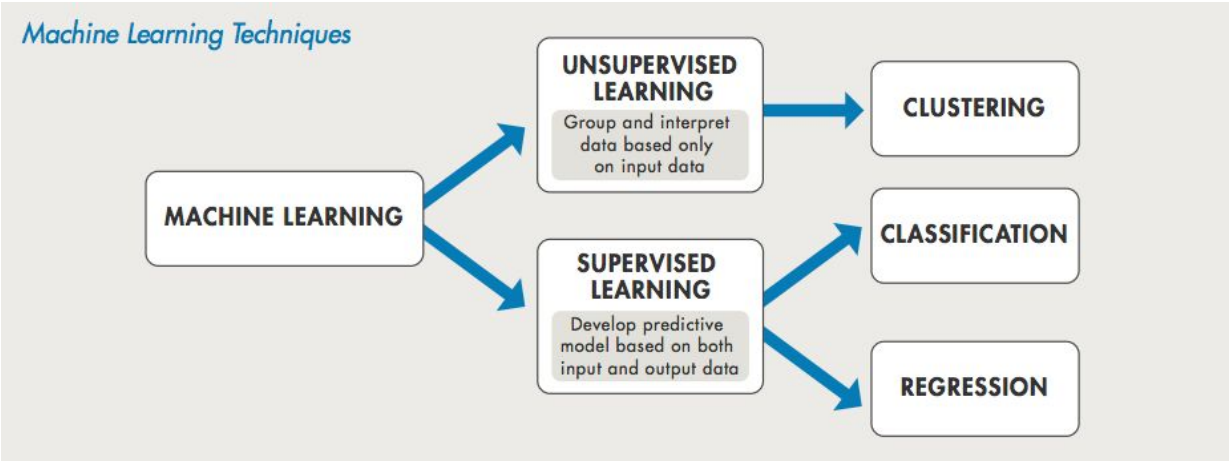
Tipos de treinamento de redes

- **Aprendizado supervisionado:**
  - Entradas ao treinamento:
    - Dados de entrada
    - Saídas respectivas desejadas
  - Saída do treinamento:
    - Pesos para minimizar o erro entre as saídas previstas e as saídas desejadas



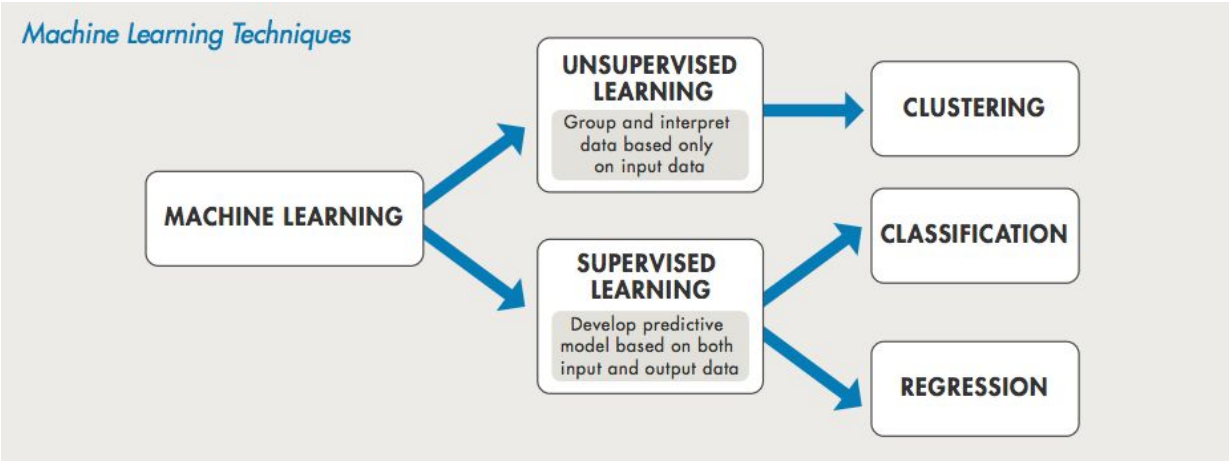
Tipos de treinamento de redes

- **Aprendizado não-supervisionado:**
  - Entradas ao treinamento:
    - Dados de entrada
  - Saída do treinamento:
    - Padrões e estruturas nos dados (agrupamentos por semelhanças)



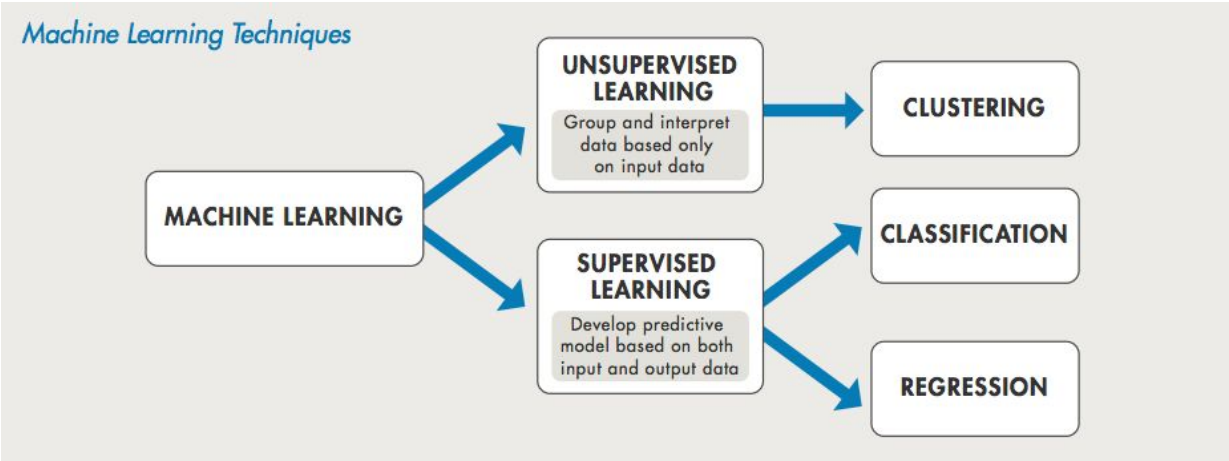
Tipos de treinamento de redes

- **Aprendizado semi-supervisionado:**
  - Combina características do treinamento supervisionado e não-supervisionado
  - O modelo é treinado com um conjunto de dados que contém tanto exemplos rotulados (supervisionado) quanto não rotulados (não-supervisionado).



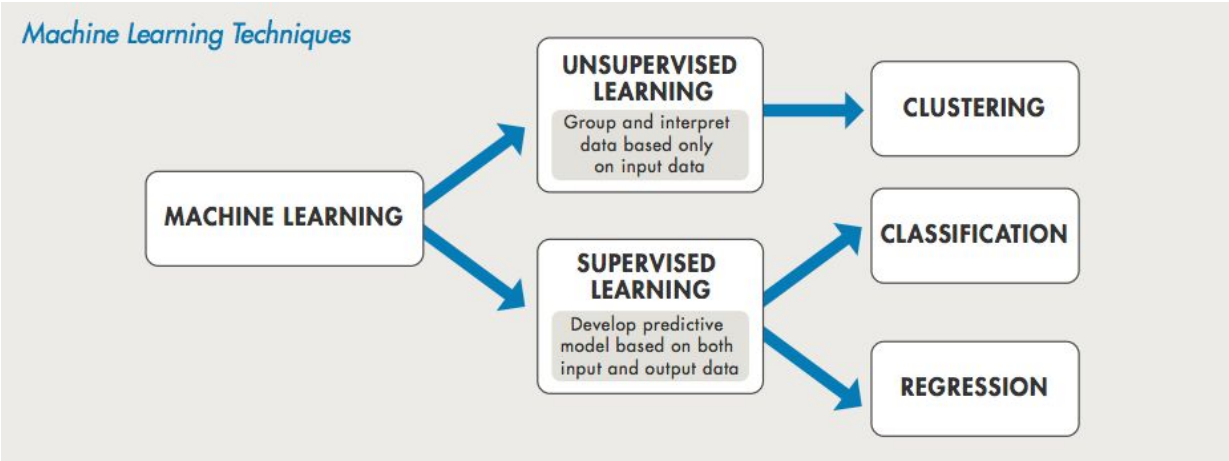
Tipos de treinamento de redes

- **Transfer learning:**
  - Aproveita o conhecimento prévio adquirido por um modelo treinado em uma tarefa;
  - O modelo pré-treinado é ajustado (*fine-tuned*) com um conjunto menor de dados específico para a nova tarefa, economizando tempo e recursos de treinamento.



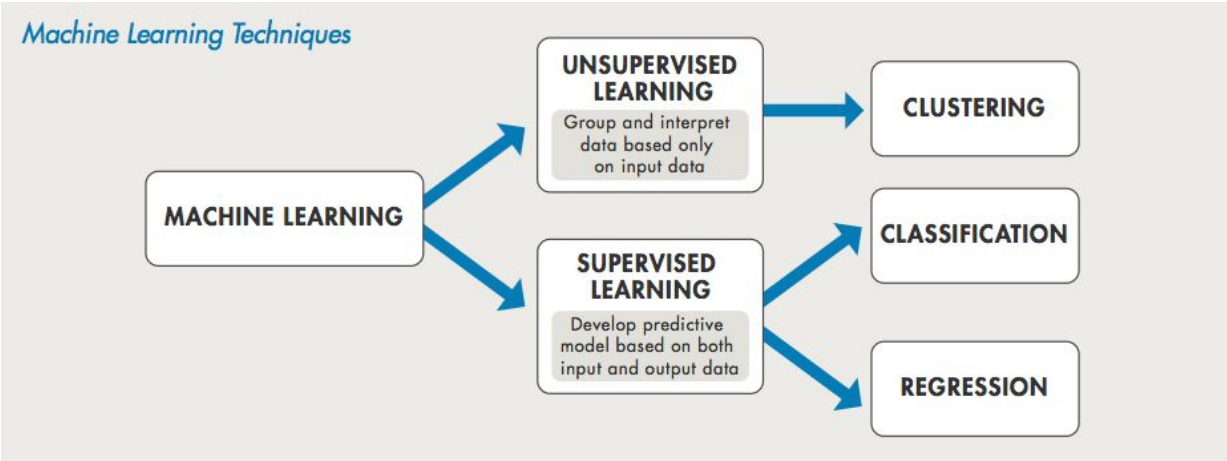
Tipos de treinamento de redes

- **Treinamento por reforço (*reinforcement learning*):**
  - O modelo (agente) interage com um ambiente e aprende a tomar decisões para maximizar uma recompensa cumulativa ao longo do tempo;
  - O agente recebe *feedback* sobre suas ações, na forma de recompensas ou penalidades, e utiliza esse feedback para melhorar seu comportamento ao longo do tempo.



Tipos de treinamento de redes

- **Treinamento por reforço (*reinforcement learning*):**
  - Exemplo: Treinamento de um agente para jogar xadrez ou dirigir um carro autônomo, onde o agente aprende com base nas recompensas recebidas após cada movimento ou ação.



Referências

- 3Blue1Brown
  - But what is a neural network? | Chapter 1, Deep learning
  - Gradient descent, how neural networks learn | Chapter 2, Deep learning
  - What is backpropagation really doing? | Chapter 3, Deep learning
  - Backpropagation calculus | Chapter 4, Deep learning

- Computerphile
  - CNN: Convolutional Neural Networks Explained - Computerphile
  - Inside a Neural Network - Computerphile
  - Deep Learning - Computerphile

Referências

- Arquiteturas
  - BerryNet
  - Yolo
  - Yolo Tiny

- Cursos e livros
  - deeplearning.ai
  - deeplearningbook.org
  - Embedded Machine Learning
  - Mastering Machine Learning: A Step-by-Step Guide with MATLAB
  - neuralnetworksanddeeplearning.com