

Inteligência Artificial

Instrutores

Ph.D. Professor Aluisio Igor Rego
Fontes



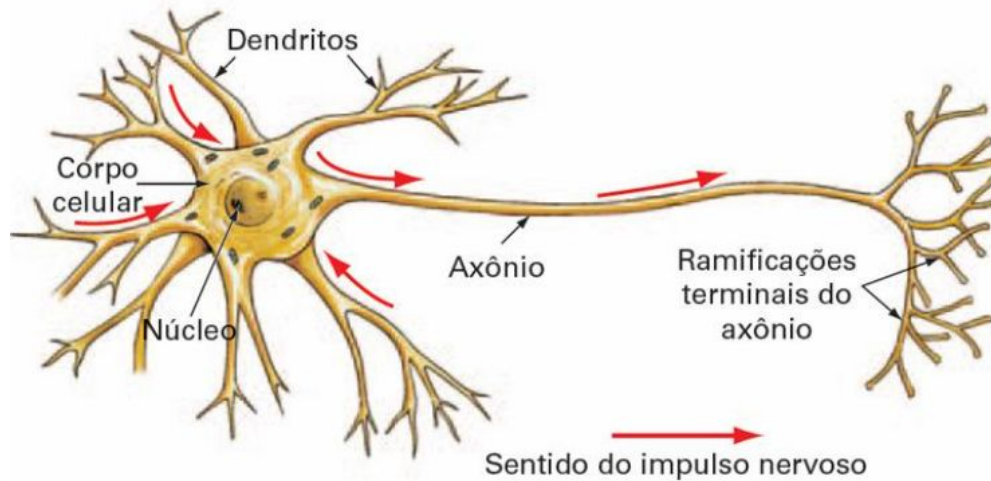
Capacitação Tecnológica
em Indústria 4.0 e Cidades
Inteligentes



Redes Neurais e Deep Learning

Cérebro humano e seus componentes

- Figura simplificada de um neurônio

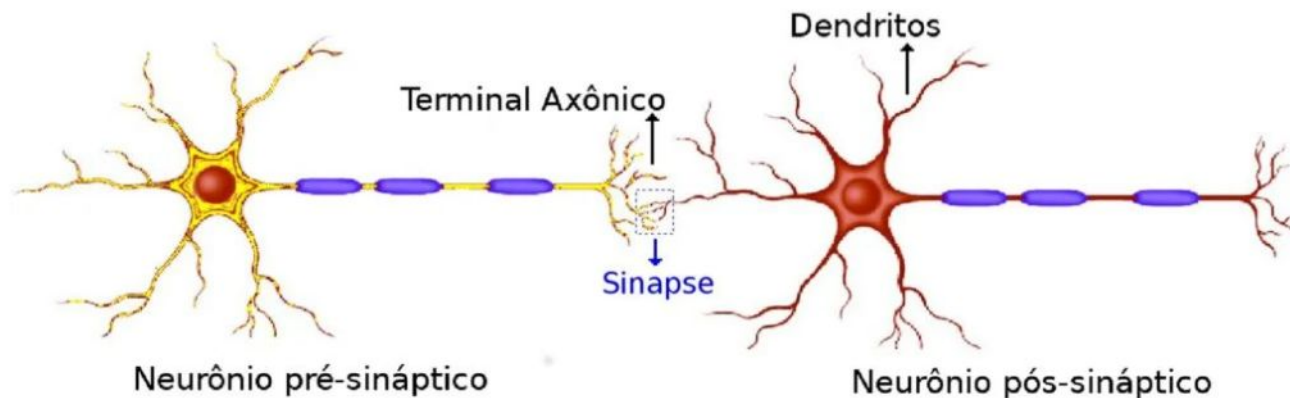


<https://www.deeplearningbook.com.br/o-neuronio-biologico-e-matematico/>



Cérebro humano e seus componentes

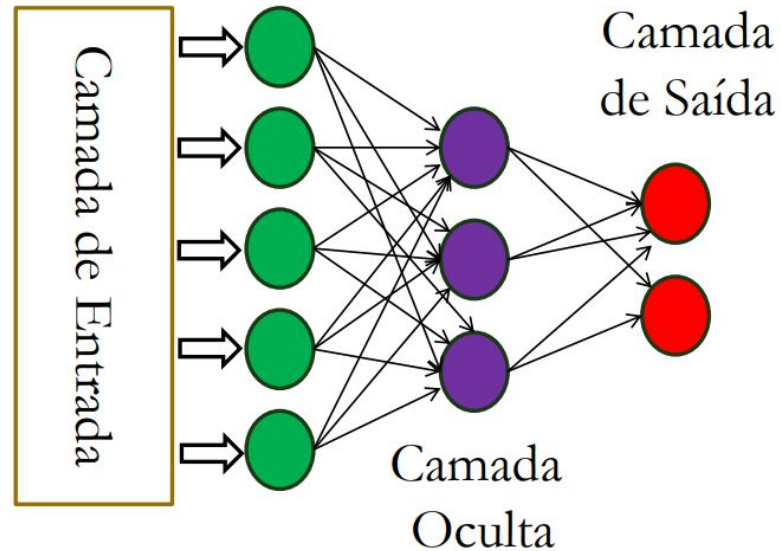
- Sinapses



<https://escolaeducacao.com.br/sinapses/>



Estrutura de uma rede neural



Neurônio de McCulloch e Pitts

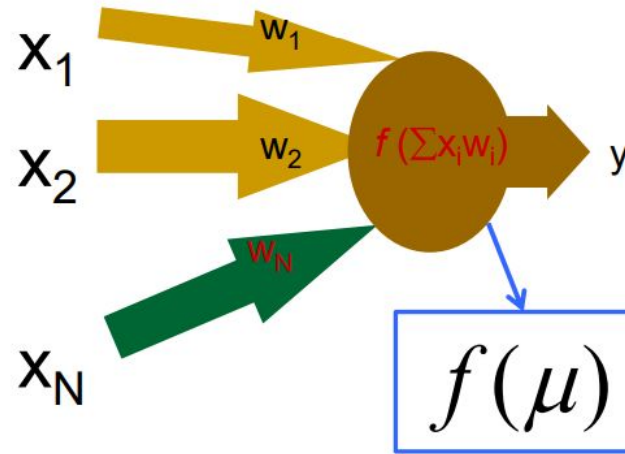
- Modelo era uma simplificação do que se sabia na época sobre o neurônio biológico.
- Sua representação matemática resultou em um modelo com:
 - N valores de entradas (referenciados com $X.N$);
 - Valores em suas arestas, representando sinapses (pesos, referenciados com $W.N$);
 - Apenas 1 (um) valor de saída (referenciado com Y).
- Os pesos são multiplicados às entradas.



Neurônio de McCulloch e Pitts

$$u = \sum_{j=1}^N x_j w_j$$

Soma ponderada



Função de ativação



Função de ativação

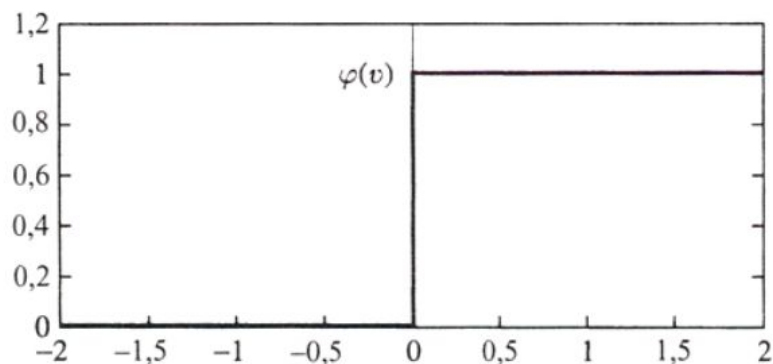
- Tipos mais comuns de função de ativação:
 - Linear
 - Limiar
 - Sigmóide
 - Tangente Hiperbólica



Função de ativação

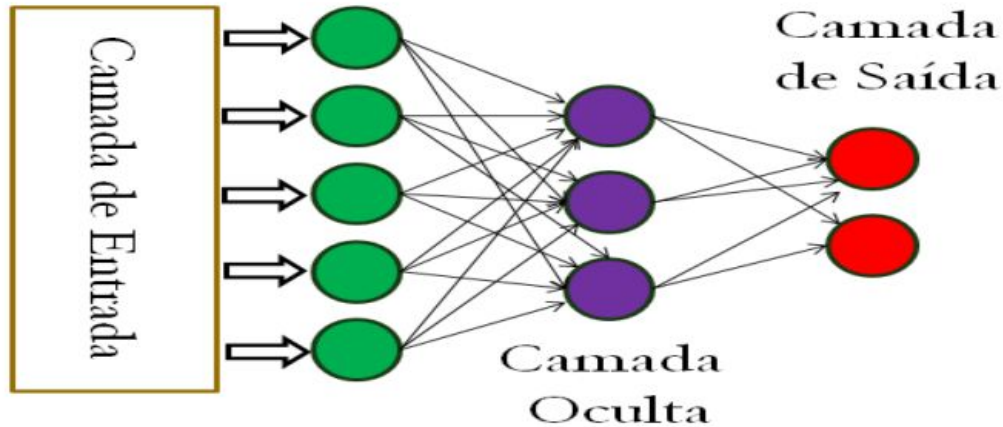
- Função Limiar

$$\varphi(y) = \begin{cases} 1 & \text{se } x > 0 \\ 0 & \text{se } x \leq 0 \end{cases}$$



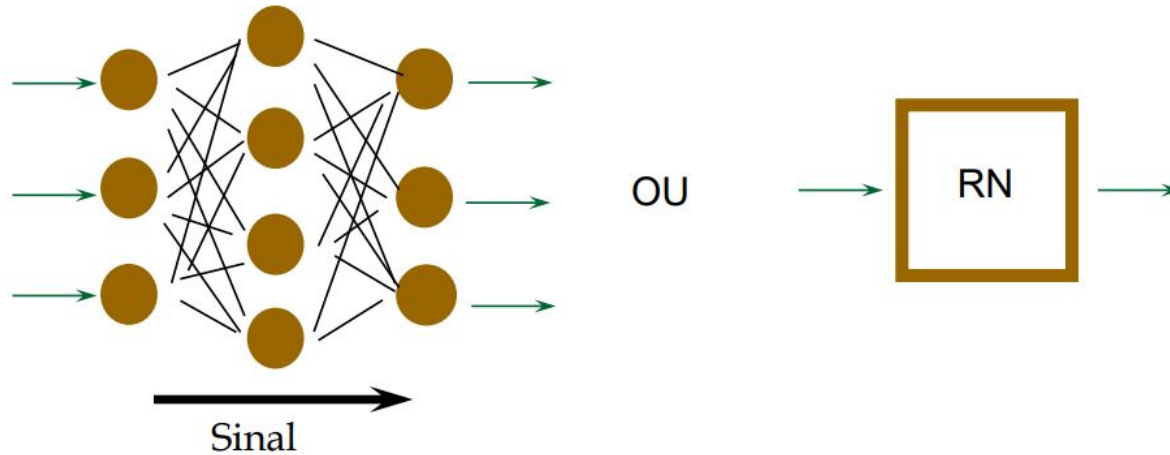
Topologia

- Múltiplas camadas
 - Exemplos: MLP
 - Completamente Conectadas
 - Parcialmente Conectada



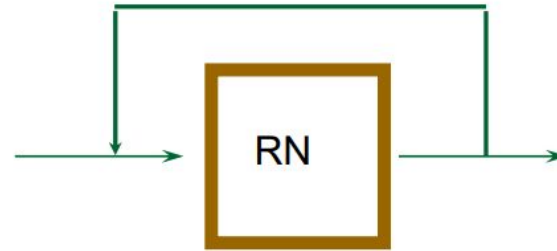
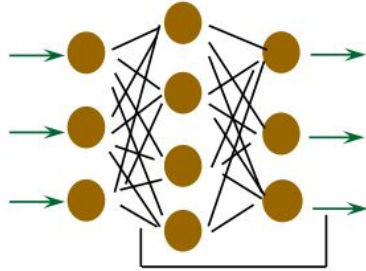
Redes *Feedforward*

- Sinais seguem uma única direção



Redes Recorrentes

- Possuem conexões ligando saída da rede a sua entrada

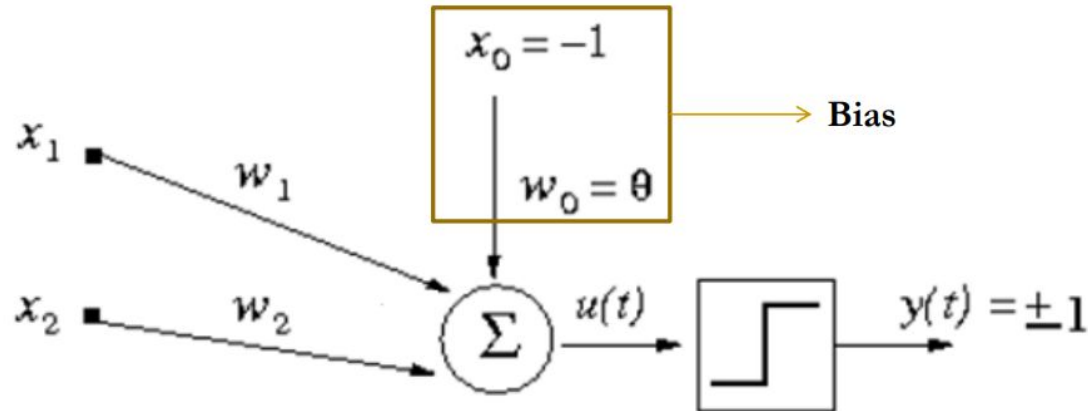


- Podem lembrar entradas passadas e, conseqüentemente, processar seqüência de informações (no tempo ou espaço).



Perceptrons

- Desenvolvido por Rosemblat, 1958.
- Utiliza modelo de McCulloch-Pitts como neurônio.



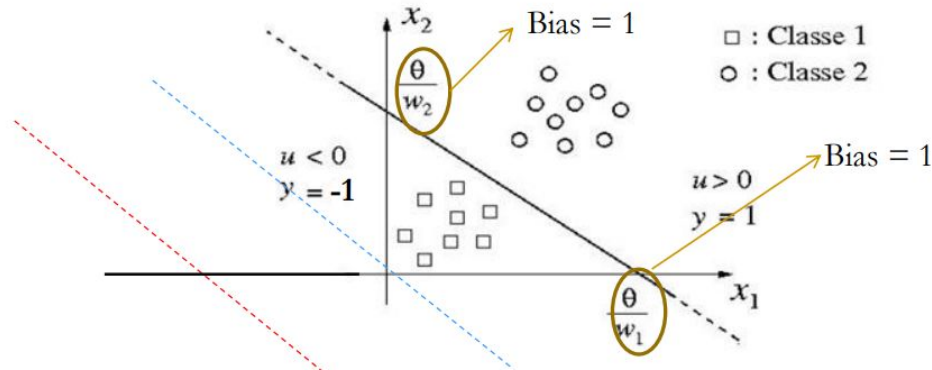
Perceptrons - *Bias*

- O que seria *Bias*?
 - Um parâmetro a mais na função de ativação;
 - Valor fixo (-1, 0, ou 1);
 - O uso do bias permite que definamos o valor de *threshold* adotado em nossa função de ativação, sendo necessário então atualizar somente os pesos e o bias na rede.
 - A mesma regra para atualização dos pesos é válida também para a atualização do *bias*.



Perceptrons

- O neurônio MP, do ponto de vista geométrico, pode ser interpretado como uma reta (2D), ou um plano (3D) ou ainda um hiperplano ($> 3D$), que é usado para separar duas categorias de dados distintas.



Perceptrons e Adaline (1 camada)

- Problema linearmente separável

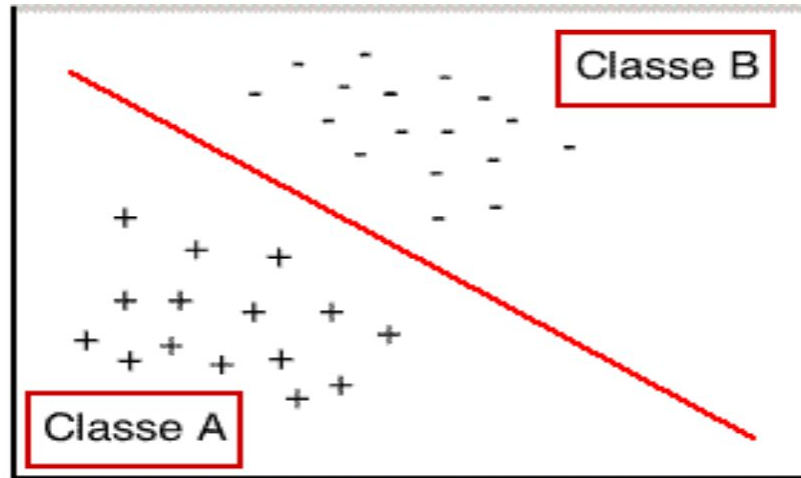


Fig. 1: Classes linearmente separáveis



Perceptrons e Adaline (1 camada)

- Problema não linearmente separável

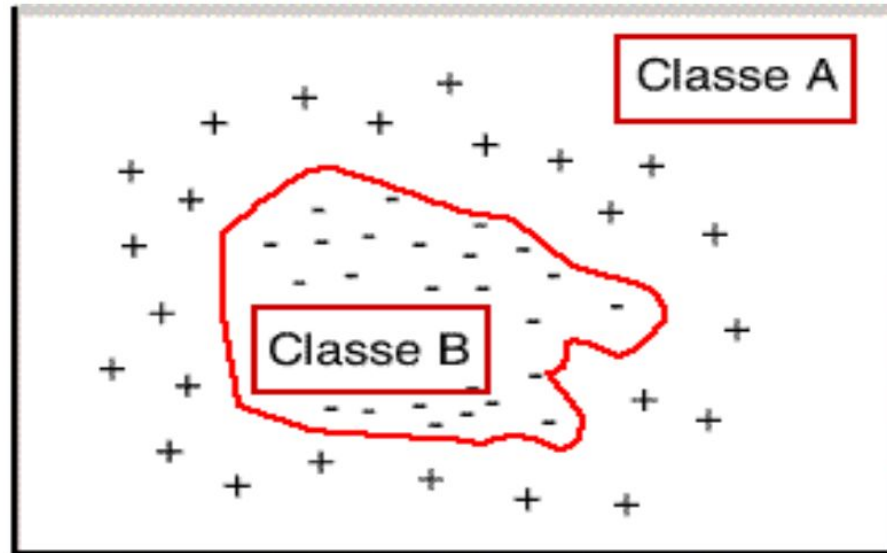


Fig. 2: Classes não linearmente separáveis



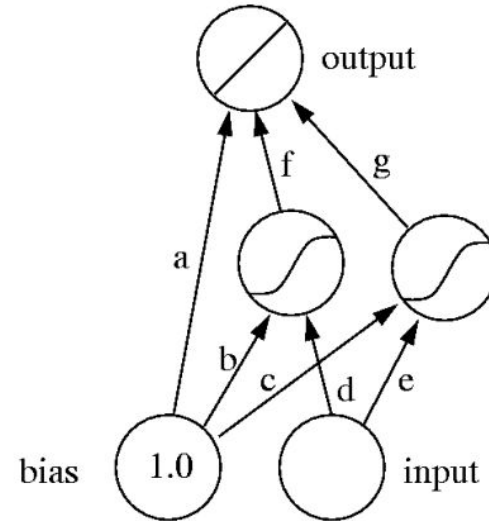
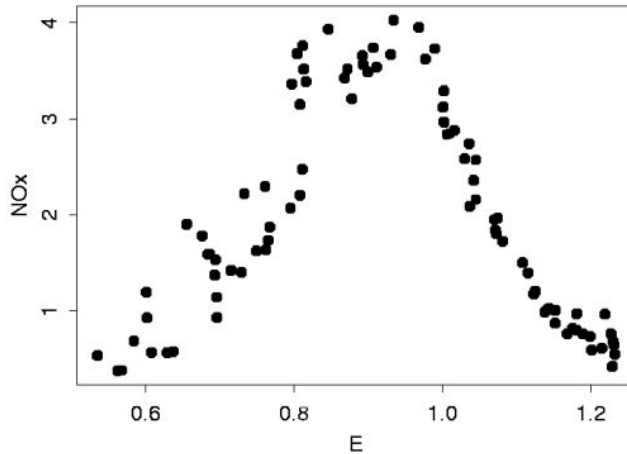
Backpropagation

O *backpropagation*, abreviação de "*backward propagation of errors*" (propagação reversa de erros), é um algoritmo fundamental utilizado no treinamento de redes neurais artificiais. Ele é empregado para ajustar os pesos das conexões entre os neurônios, de modo a minimizar o erro na saída da rede em relação às saídas desejadas.

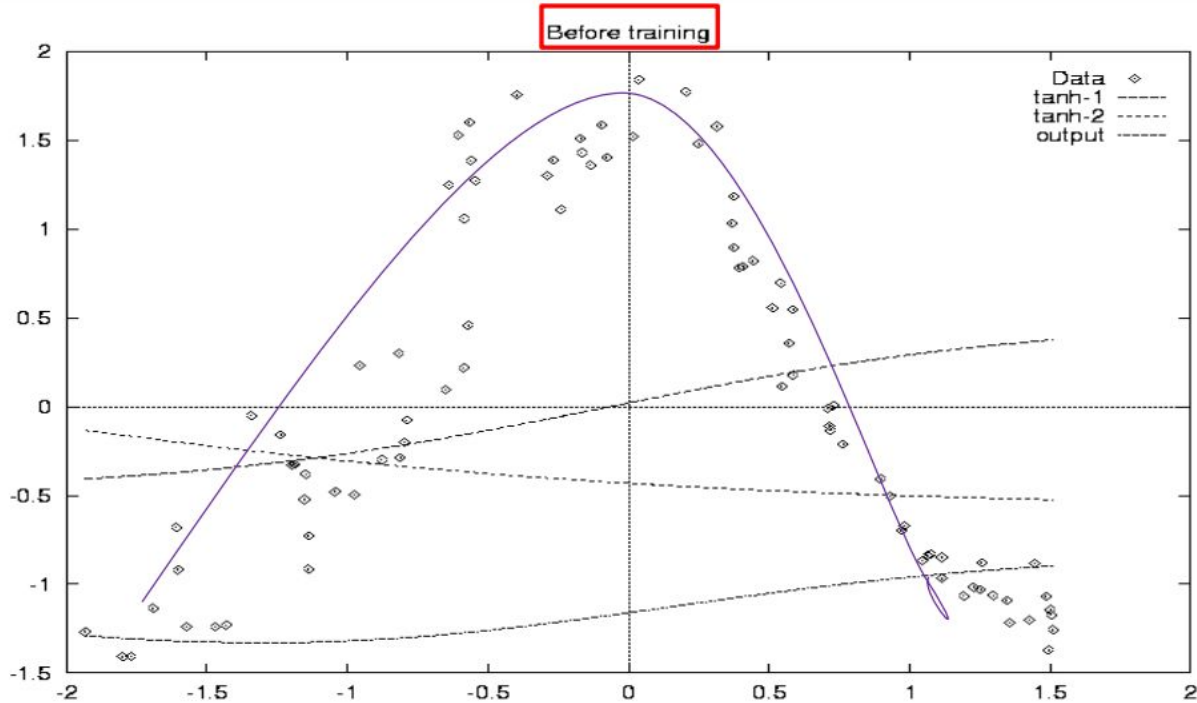


Backpropagation - Exemplo (1/7)

- Demonstração de RNA como aproximador de função



Backpropagation - Exemplo (3/7)

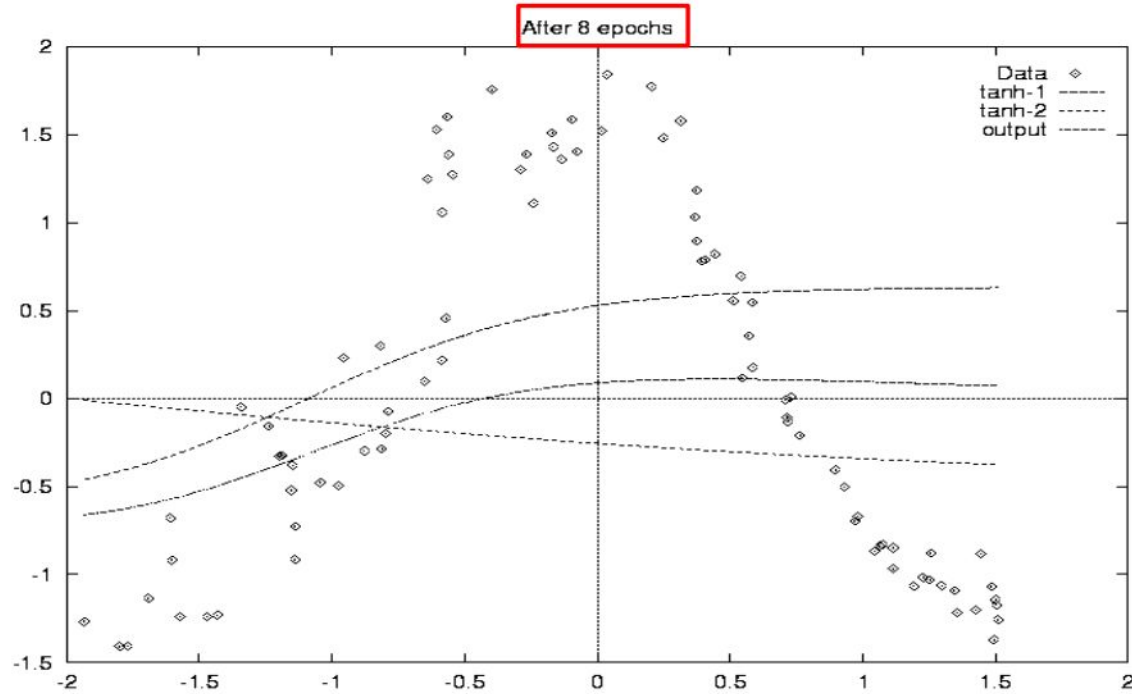


Backpropagation - Exemplo (4/7)

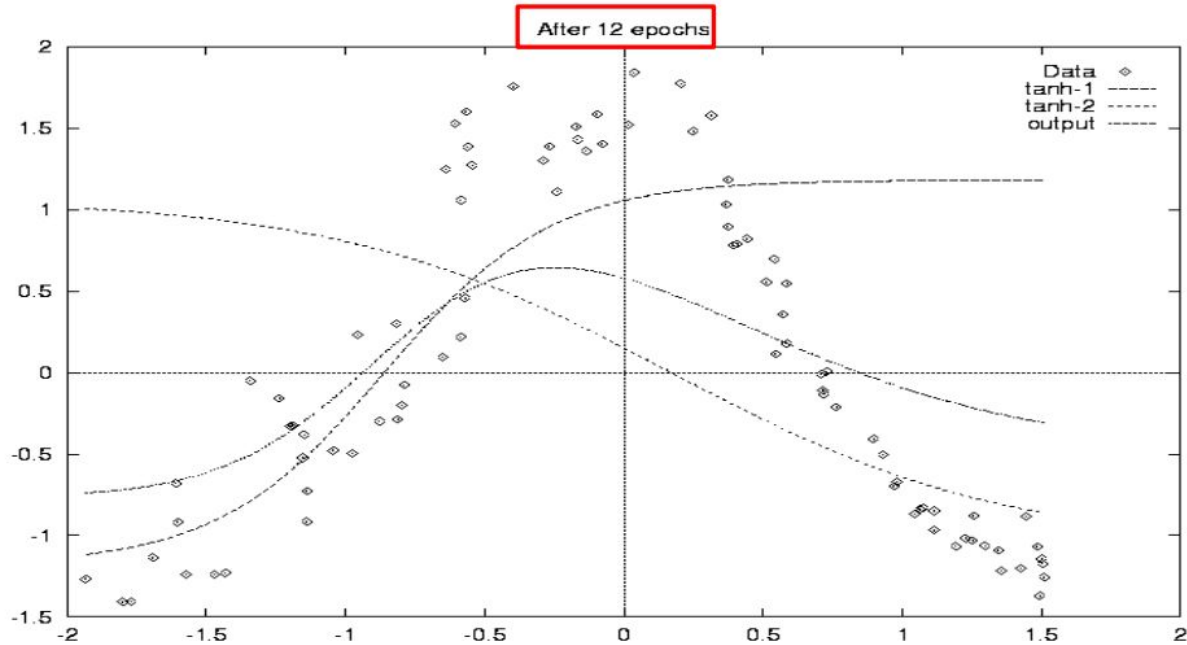
- Treinamento da rede
- Taxa de aprendizado 0.3
- Atualização dos pesos após cada padrão
- Aprendizagem incremental
- Depois de passarmos por todo o conjunto de treinamento 20 vezes (20 épocas) as funções computadas pela rede têm a seguinte forma:



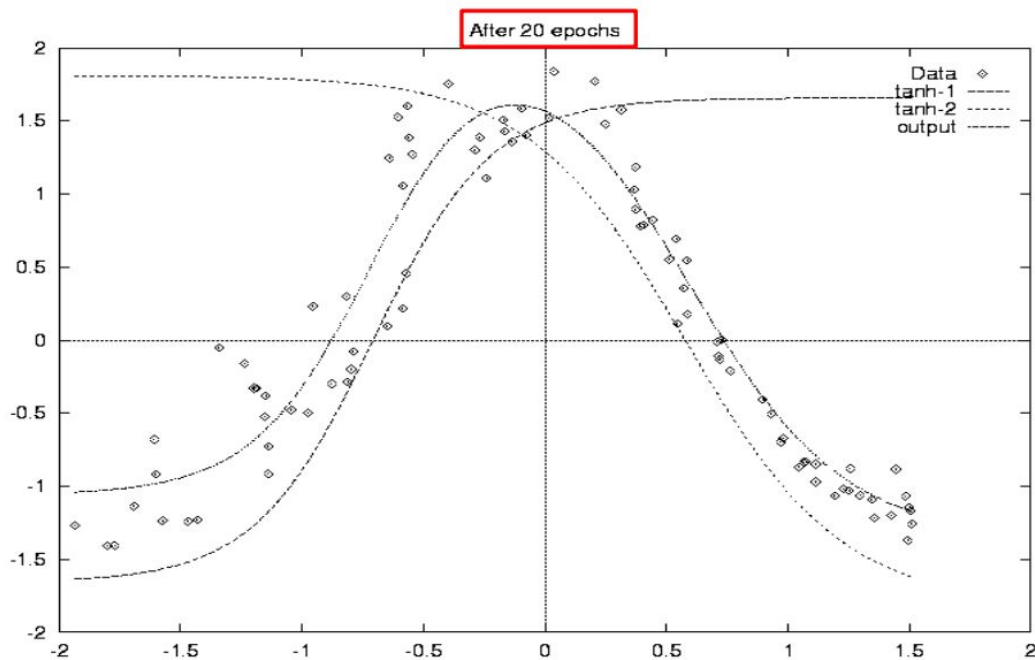
Backpropagation - Exemplo (5/7)



Backpropagation - Exemplo (6/7)

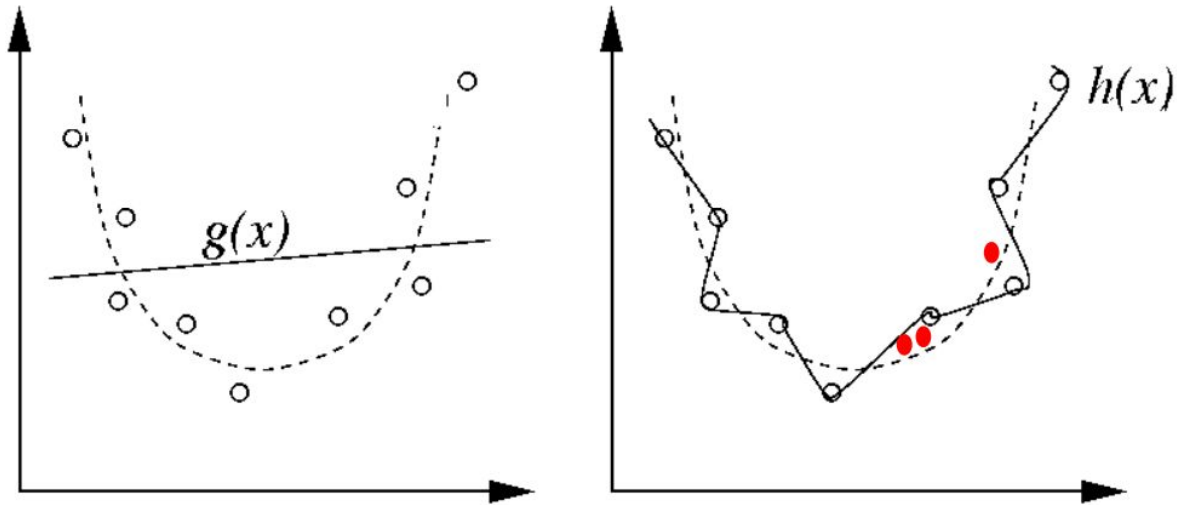


Backpropagation - Exemplo (7/7)



O problema de *underfitting* e *overfitting*

- Na estrutura de rede neural
 - Número de neurônios



Introdução ao *Deep Learning*

Introdução

Deep Learning é uma área da inteligência artificial (IA) que se concentra em algoritmos capazes de aprender a representar dados de maneira hierárquica e complexa, através do uso de redes neurais artificiais com múltiplas camadas.



Importância

- Algoritmos de *Deep Learning* têm alcançado resultados impressionantes em uma variedade de tarefas desafiadoras. Um exemplo notável é o AlphaGo, desenvolvido pela DeepMind, que derrotou os melhores jogadores de Go do mundo, demonstrando a capacidade do Deep Learning de lidar com problemas complexos e de alto nível.



Importância

- O *Deep Learning* tem sido aplicado com sucesso em diversos setores, incluindo saúde (diagnóstico médico por imagem, descoberta de medicamentos), finanças (previsão de mercado, detecção de fraudes) e tecnologia (reconhecimento de fala, visão computacional, assistentes virtuais).

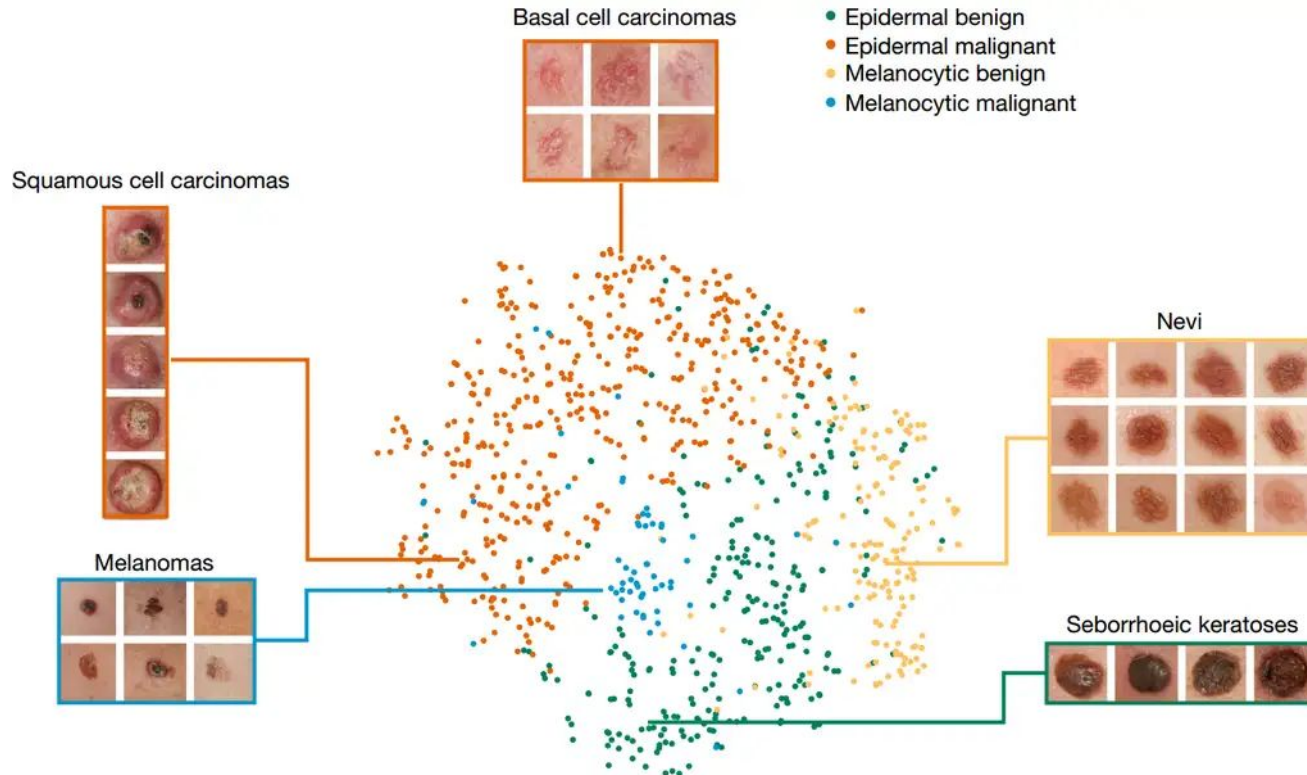


Importância

- Em 2017, cientistas da computação da Universidade de Stanford criaram um modelo de CNN que foi treinado com mais de 130 mil imagens médicas sobre patologias da pele para detecção de câncer.
- Classificando as imagens desse jeito:



Importância



Importância

- Um ano depois, em 2018, a Sociedade Europeia de Oncologia Médica, mostrou resultados ainda melhores: A CNN classificou corretamente 95% dos casos de câncer de pele, enquanto os dermatologistas que participaram da pesquisa, 86%.





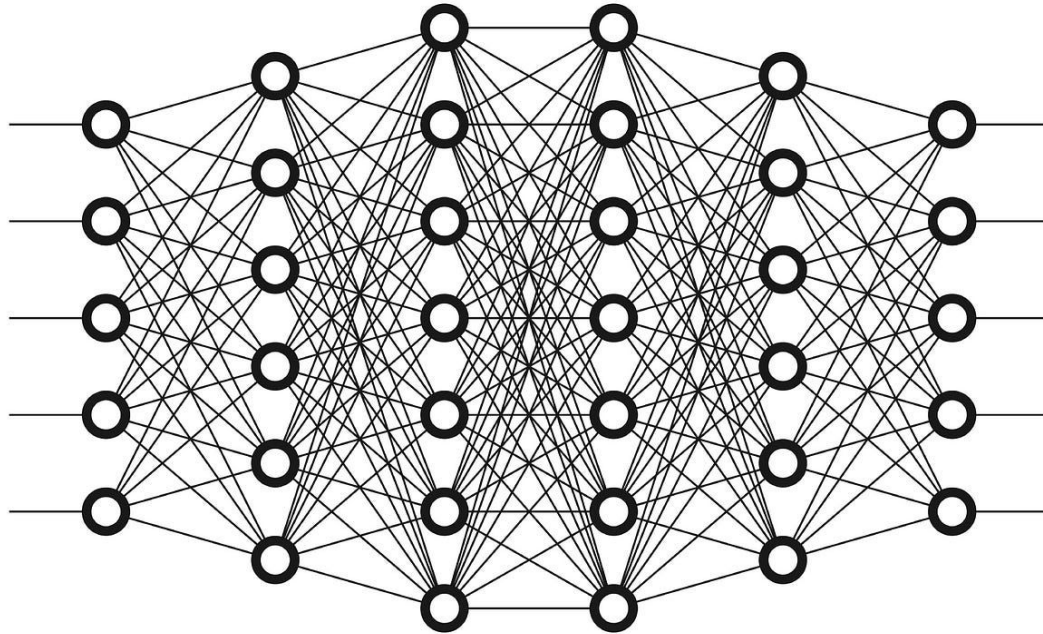
O que é *Deep Learning*?

O que é *Deep Learning*?

O *deep learning* é composto por redes neurais artificiais com várias camadas (daí o termo "profundo"), o que permite aprender representações complexas dos dados. As redes neurais profundas são compostas por várias camadas de neurônios interconectados, onde cada camada processa informações de forma progressivamente mais abstrata, permitindo a extração de características complexas dos dados.



O que é *Deep Learning*?



O que é *Deep Learning*?

É possível adicionar várias camadas no mlp e torná-la deep?



Problema do *Vanish Gradient*

- NO PROCESSAMENTO DE UMA MLP:
 - O gradiente do erro é retropropagado, se tornando cada vez menor
 - Os gradientes vai diminuindo cada vez mais.
 - Com gradientes pequenos, não é possível atualizar efetivamente os pesos e desvios das camadas iniciais a cada sessão de treinamento.
 - As camadas iniciais são vitais para reconhecer os elementos centrais dos dados de entrada
 - Se os seus pesos e tendências não forem devidamente atualizados, é possível que toda a rede seja imprecisa.



Problema do *Vanish Gradient*

- Então, quais são os principais modelos de *deep learning*?



Principais arquiteturas

● Aplicações

- Remoção de ruídos
- Geração de texto/imagem/vídeo
- Processamento de Imagens
- Detecção de Objetos em um vídeo
- Processamento de Linguagem Natural
- Modelos de tradução de Línguas
- Modelos de Línguas

● Modelos

- *Autoencoders*
- GAN's
- Arquiterutas CNNs
- YOLO (You Only Look Once)
- RNNs, LSTM
- LSTM e GRUs
- *Transformers*





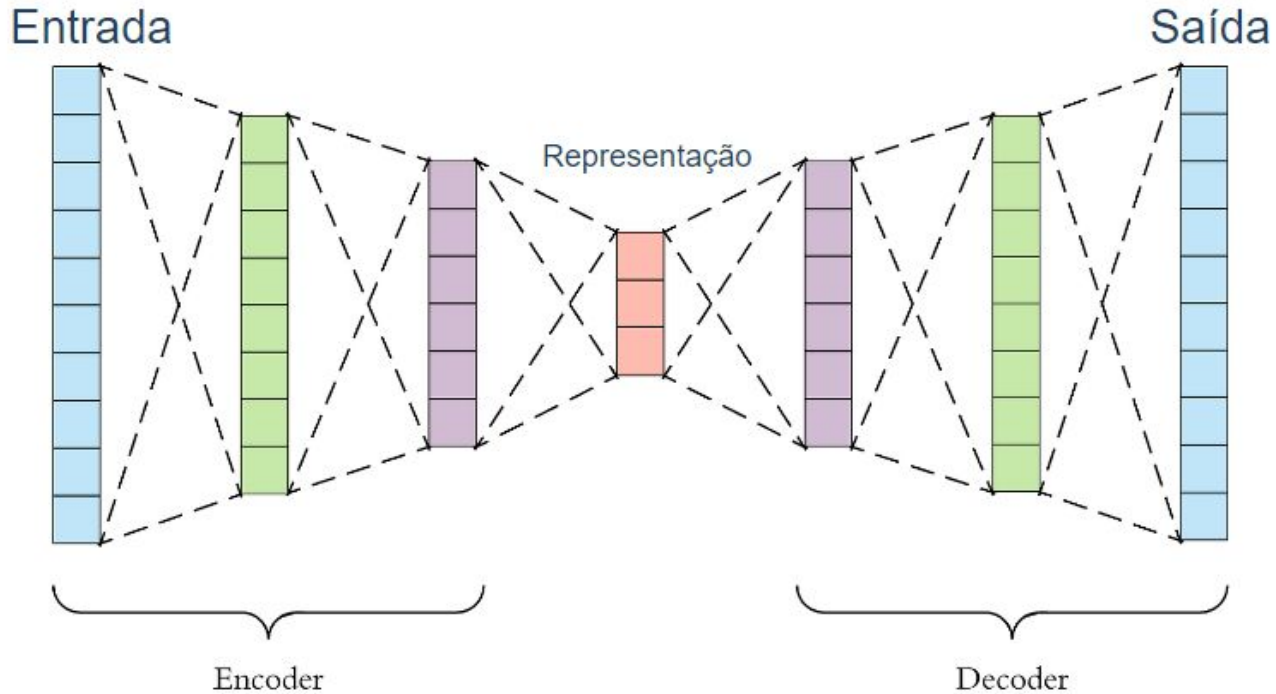
Iniciando as arquiteturas

Autoencoder

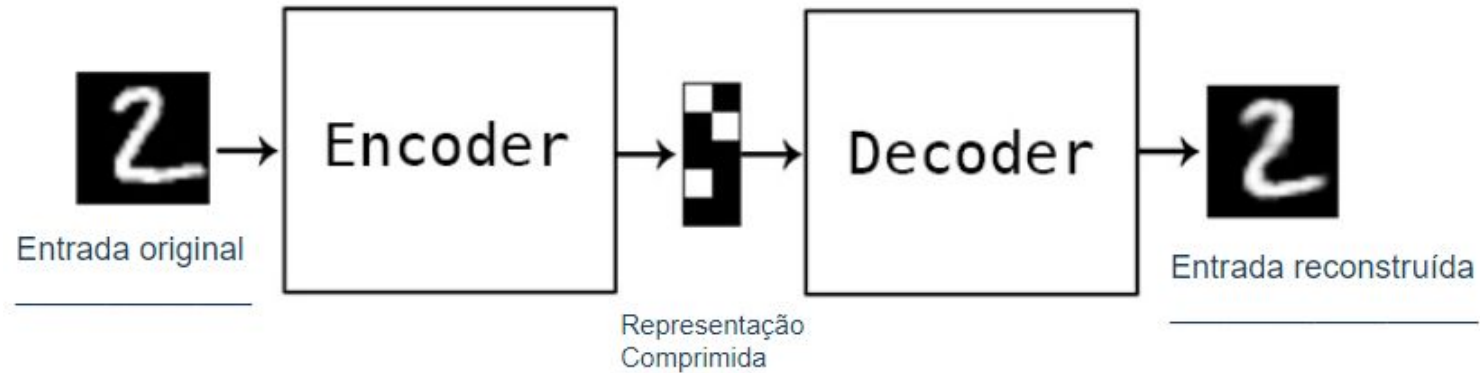
- A rede de *autoencoder* é uma rede que tenta recriar a entrada em sua saída usando *backpropagation*. A rede irá comprimir a entrada, transformar ela uma representação e usar essa representação para reconstruir a entrada.
 - Algo parecido com o PCA.
- É uma Arquitetura não supervisionada



Exemplo - Autoencoder



Exemplo - Autoencoder



Gan - *Generative adversarial network*

As Redes Generativas Adversariais (GANs) são uma arquitetura de rede neural composta por duas redes principais: o gerador e o discriminador.

- Gerador:
 - O gerador cria novas instâncias de dados sintéticos a partir de um espaço de entrada aleatório, geralmente vetores de ruído, com o objetivo de gerar dados que se assemelham aos dados reais.
- Discriminador:
 - O discriminador é treinado para distinguir entre dados reais e dados gerados pelo gerador.
 - Ele recebe amostras de dados (reais ou sintéticas) e tenta classificá-las como "real" ou "falsa" (gerada pelo gerador).

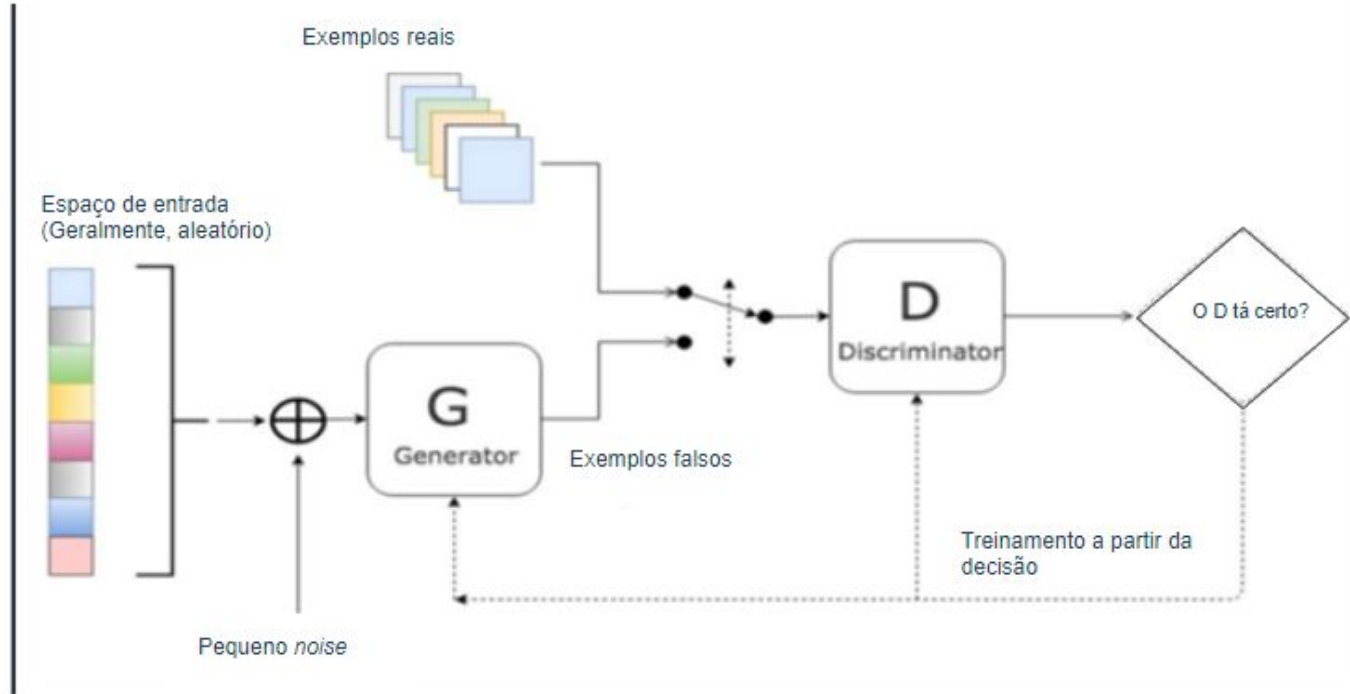


Gan - *Generative adversarial network*

- **Processo de Treinamento:**
 - Durante o treinamento, o gerador e o discriminador são treinados de forma adversarial.
 - O gerador tenta gerar dados que enganem o discriminador, enquanto o discriminador tenta melhorar sua capacidade de distinguir entre dados reais e falsos.
 - Esse jogo entre gerador e discriminador leva a uma melhoria contínua em ambos os modelos.
- **Objetivo das GANs:**
 - O objetivo das GANs é aprender a distribuição dos dados reais para que o gerador possa criar amostras que se pareçam o máximo possível com os dados reais.
- **Aplicações das GANs:**
 - As GANs têm uma ampla gama de aplicações, incluindo geração de imagens realistas, super-resolução de imagens, geração de texto, geração de música, entre outras.



Exemplo – Gan



Exemplo – Gan - Rostos de pessoas

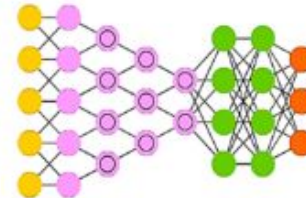
Exemplos reais



Rosto Real



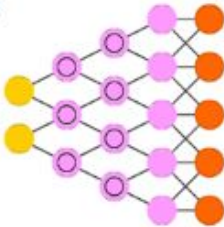
Discriminator



Verdadeiro ou falso?

Generator

Entradas aleatórias



Rosto gerado artificialmente





OBRIGADO!