Pós Graduação em Engenharia Elétrica e de Computação
Universidade Federal do Ceará – Campus Sobral

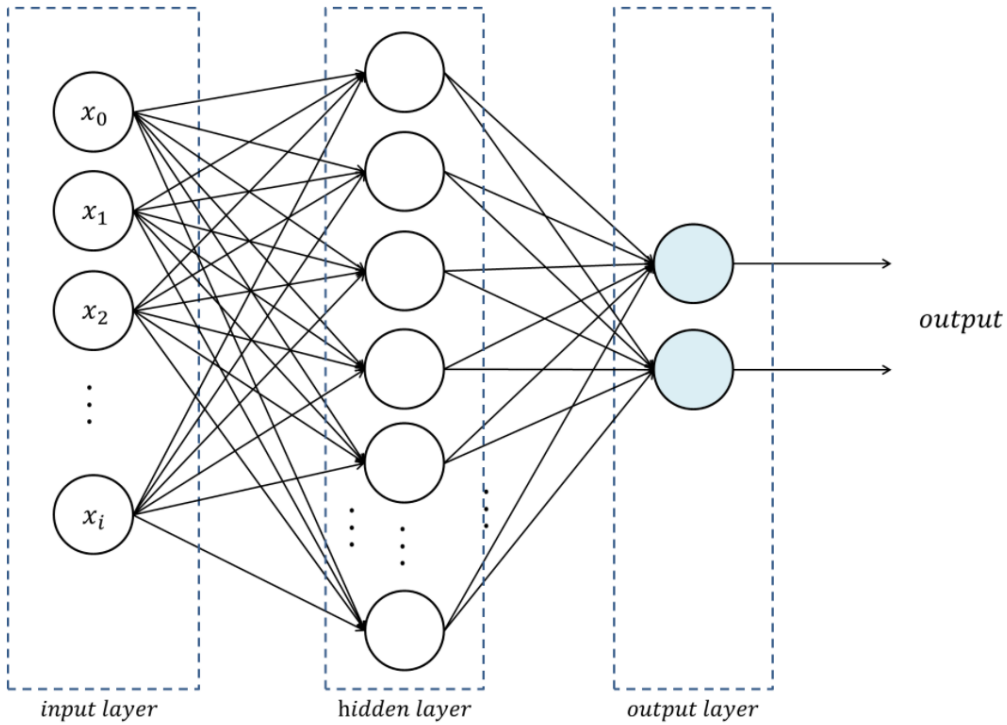
MLP e Aplicações

David Borges
davidborges@protonmail.com

10 de Maio, 2019

MLP (MultiLayer Perceptron)

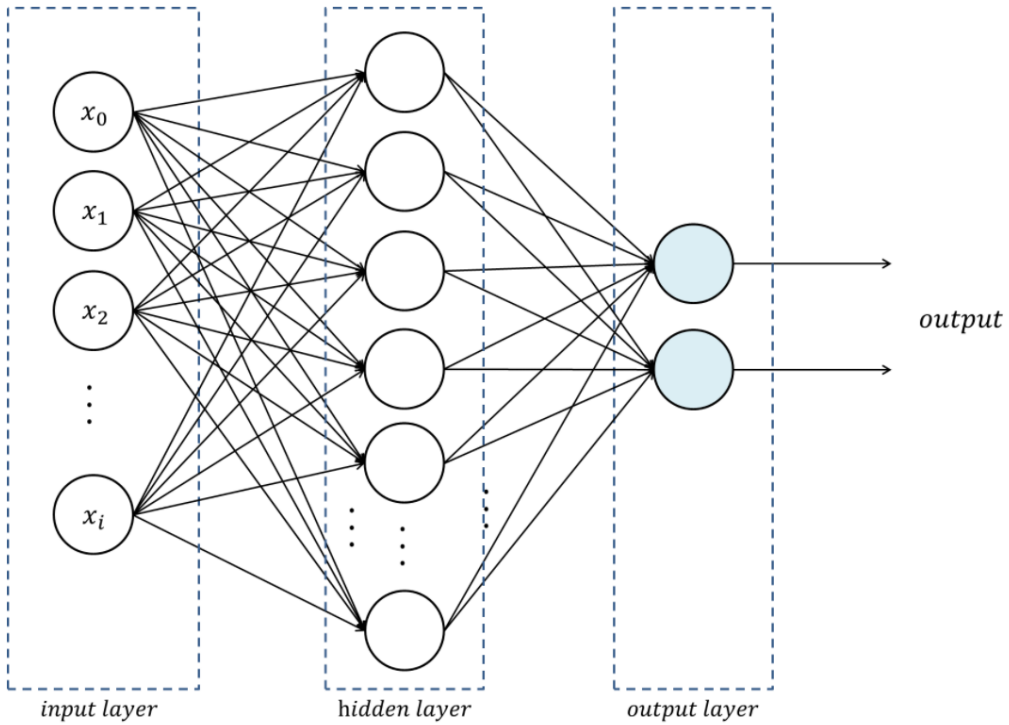
MLP Uma rede de alimentação direta com camadas totalmente conectadas compostas de perceptrons



Fonte: https://www.cc.gatech.edu/~san37/post/dlhc-fnn/

Alimentação direta

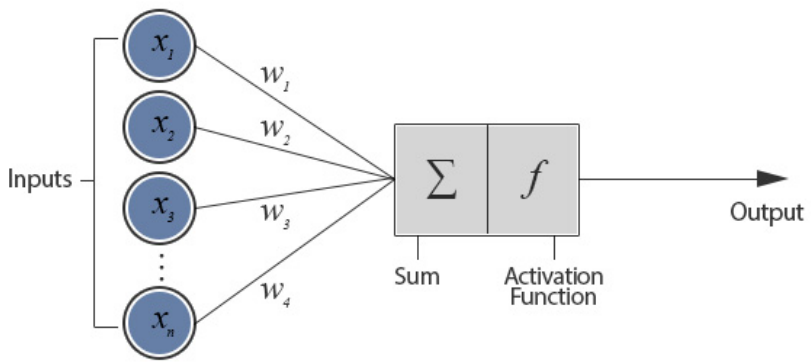
Forward pass Os dados são fornecidos à rede pela **camada de entrada**, processados sequencialmente e unidirecionalmente pelas suas **camadas ocultas** até que seja atingida a **camada de saída**



Fonte: https://www.cc.gatech.edu/~san37/post/dlhc-fnn/

Unidade de processamento

Neurônio O processamento realizado em cada neurônio é o produto escalar entre suas entradas e os **pesos de suas conexões**, seguido da aplicação de uma **função de ativação**, que determina sua saída



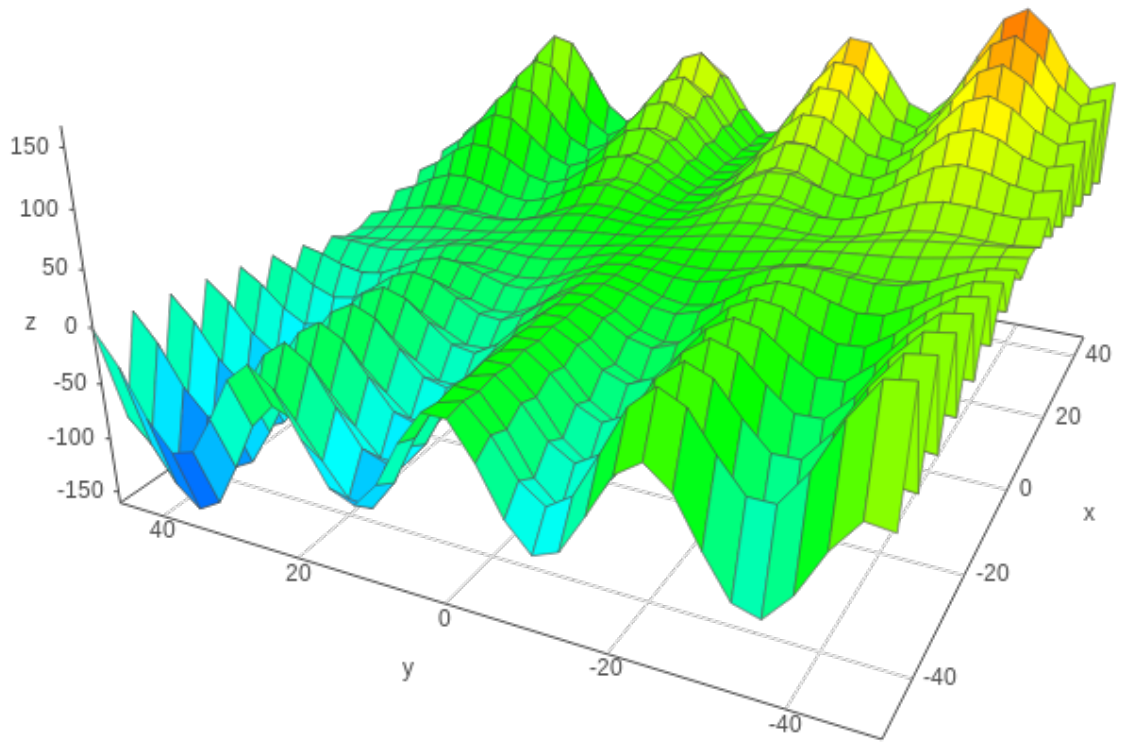
Fonte: https://www.cc.gatech.edu/~san37/post/dlhc-fnn/

Treinamento

Otimização Busca pelo conjunto de pesos W que minimiza uma função de perda

Perda Estimativa do erro do modelo

Variedade Existem muitos algoritmos de otimização e funções de perda. Qual a melhor escolha?



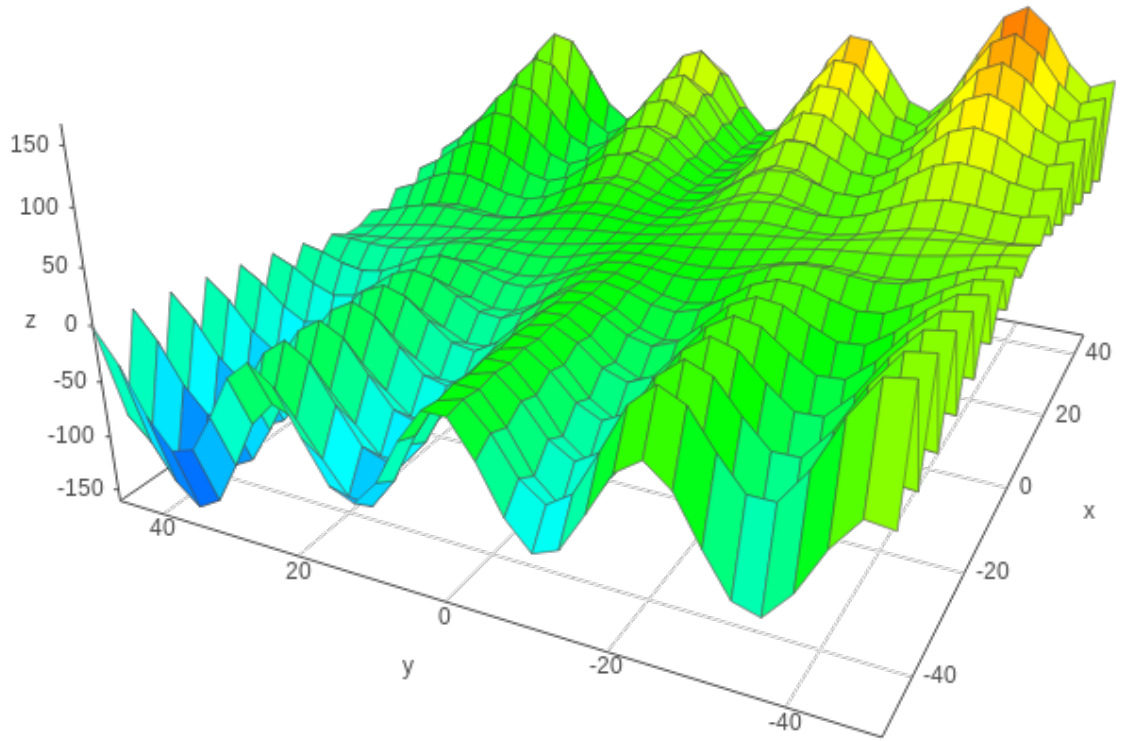
Fonte: https://academo.org/demos/3d-surface-plotter

Treinamento

Época Uma passagem completa pelo conjunto de treinamento

Convergência O momento em que o algoritmo de otimização estabiliza em um mínimo local ou global

Parada Quantas épocas realizar até atingir a convergência?



Fonte: https://academo.org/demos/3d-surface-plotter

Parâmetros da MLP

Parâmetros MLP possui um número elevado de parâmetros:

- Número de camadas ocultas
- Número de neurônios em cada camada
- Função de ativação
- Pesos
- Algoritmo de otimização dos pesos
- Função de perda
- Critério de parada
- Número máximo de épocas de treinamento
- Etc, etc, etc...

Como escolher? Experimentando!

Vantagens da MLP

Adaptabilidade Aprendizagem com base nos dados apresentados

Não linearidade Capacidade de aprender relações não lineares entre entrada e saída;

Generalização Capacidade de generalização

Uso geral Não impõe restrições sobre a distribuição dos dados de entrada

Disponibilidade Muito utilizada em pesquisa e na indústria, portanto existem diversas implementações disponíveis, prontas para uso

Desvantagens da MLP

Parâmetros	Possui uma grande quantidade de parâmetros a definir
Recursos	Treinamento e uso podem requerer alto poder de processamento e memória, a depender da arquitetura escolhida
Datasets	A depender do problema, pode ser necessário um grande conjunto de treinamento para que a rede possa aprender
Mínimos locais	O treinamento pode convergir para mínimos locais
Opacidade	Seu mecanismo de funcionamento é opaco, difícil de compreender

"Uma rede neural é a segunda melhor maneira de resolver qualquer problema. A melhor maneira é realmente entender o problema." (Autor desconhecido)

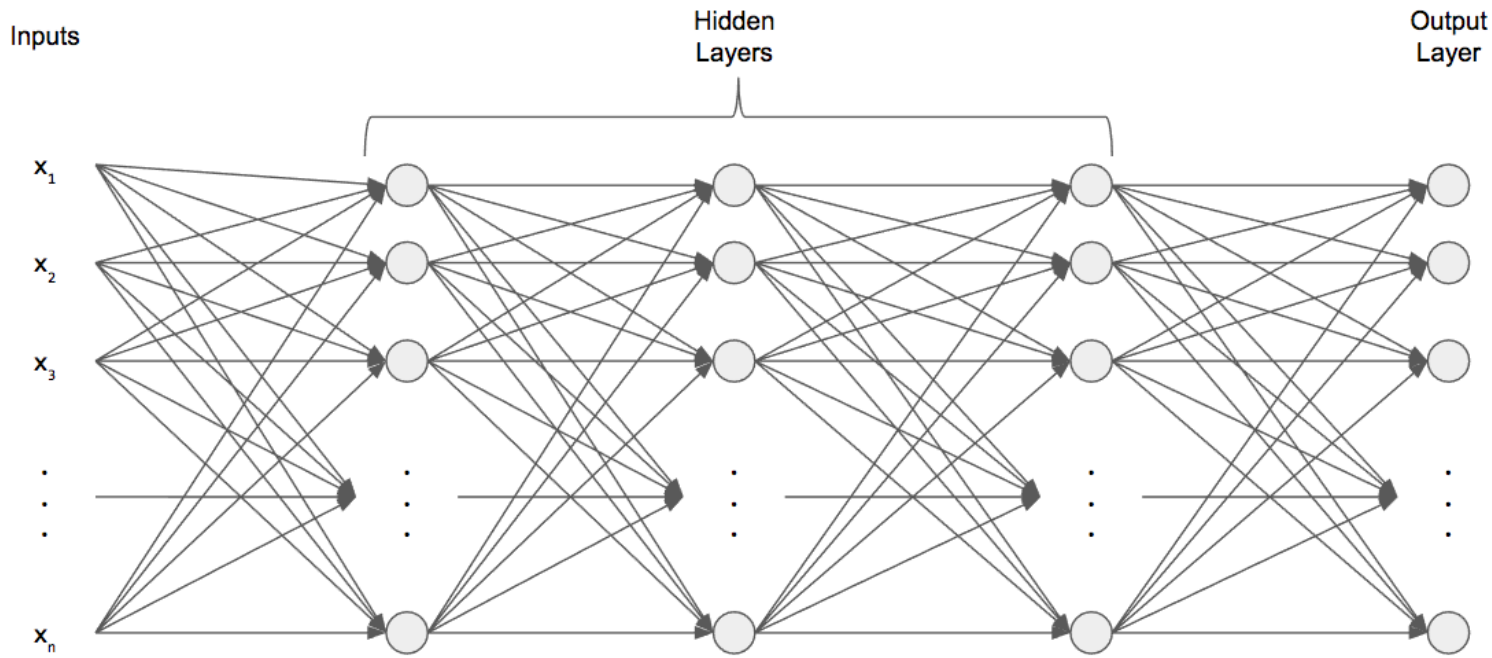
Número de camadas ocultas

Que tal duas? MLP com duas camadas ocultas é capaz de traçar regiões de classificação de qualquer forma desejada

An introduction to computing with neural nets (Lippman, 1982)

Apenas uma? Teorema da aproximação universal

Approximation capabilities of multilayer feedforward networks (Hornik, 1991)



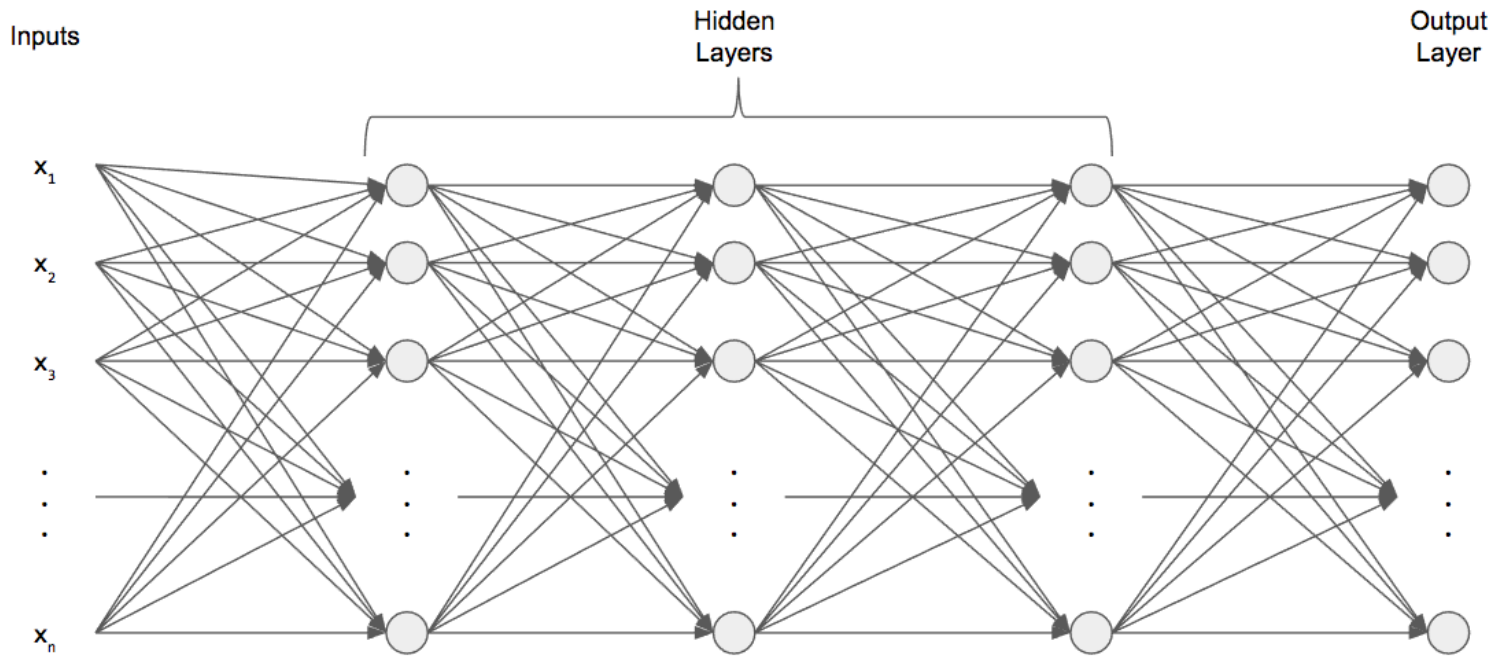
Fonte: https://pythonmachinelearning.pro/complete-guide-to-deep-neural-networks-part-1/

Número de neurônios em cada camada

Entrada Número de atributos + 1 (bias)

Ocultas Existem heurísticas e regras, mas não garantias

Saída Se o problema for de classificação: usar o número de classes; se o problema for regressão: basta um neurônio



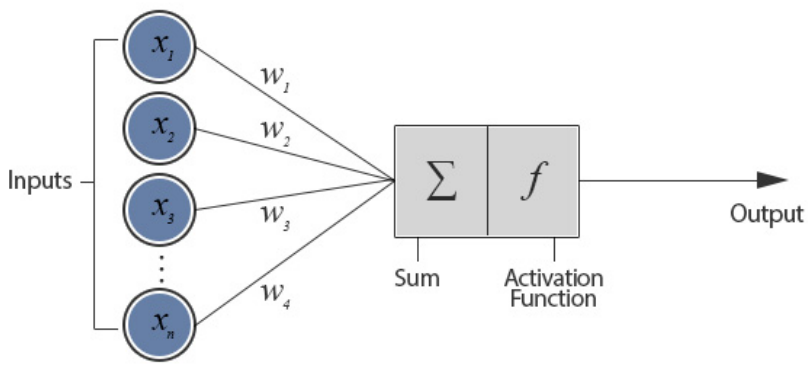
Fonte: https://pythonmachinelearning.pro/complete-guide-to-deep-neural-networks-part-1/

Função de ativação

Ativação Determina se o neurônio é ativado ou não com base em sua entrada

Normalização A saída do neurônio é normalizada para um intervalo conhecido

Não-linearidade Torna a rede capaz de modelar relações não-lineares



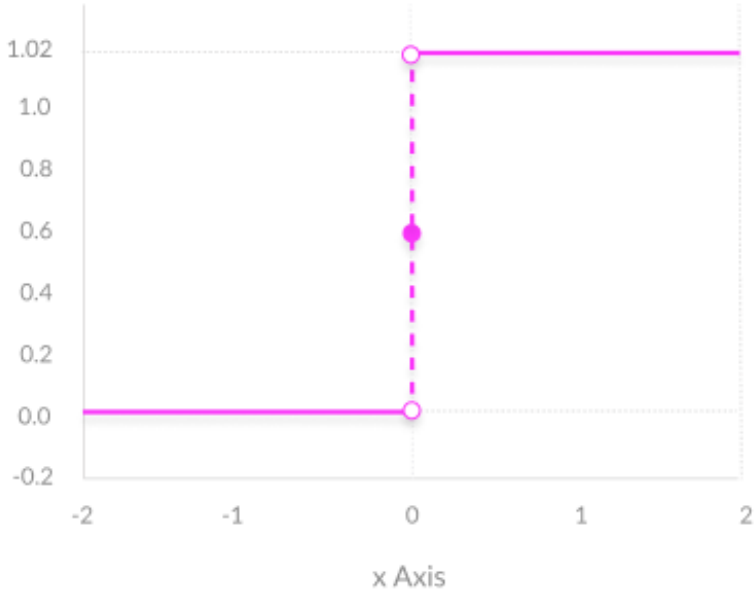
Fonte: https://www.cc.gatech.edu/~san37/post/dlhc-fnn/

Funções de ativação

Função degrau binário

Vantagens Simples, Rápida

Desvantagens Apenas dois valores de saída

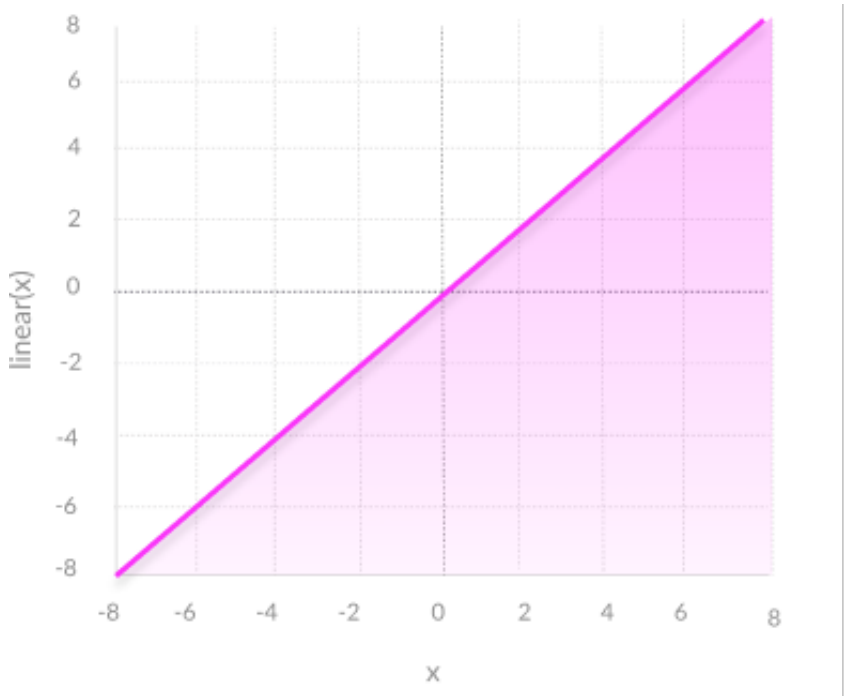


Fonte: https://missinglink.ai/guides/neural-network-concepts/7-types-neural-network-activation-functions-right/

Função linear

Vantagens Permite múltiplas saídas

Desvantagens É puramente linear; possui derivada constante



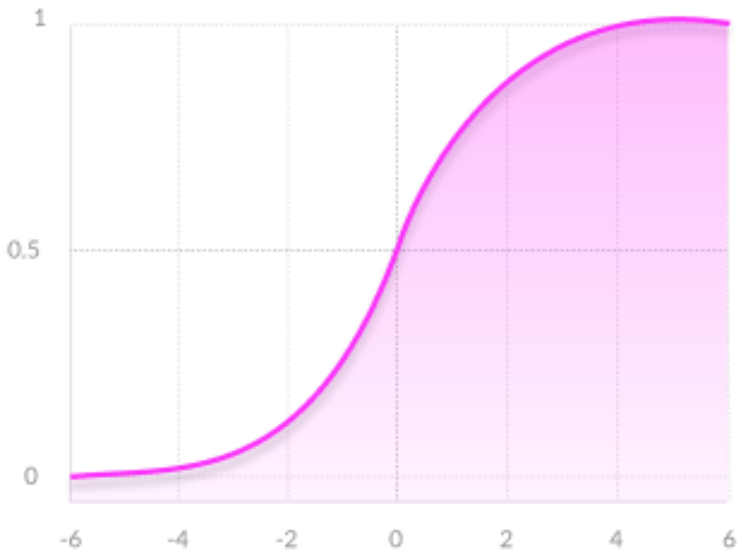
Fonte: https://missinglink.ai/guides/neural-network-concepts/7-types-neural-network-activation-functions-right/

Funções de ativação

Função sigmoide

Vantagens Não-linear; derivada suave; saída normalizada entre 0 e 1

Desvantagens *Vanishing gradients*; gradientes relativamente fracos

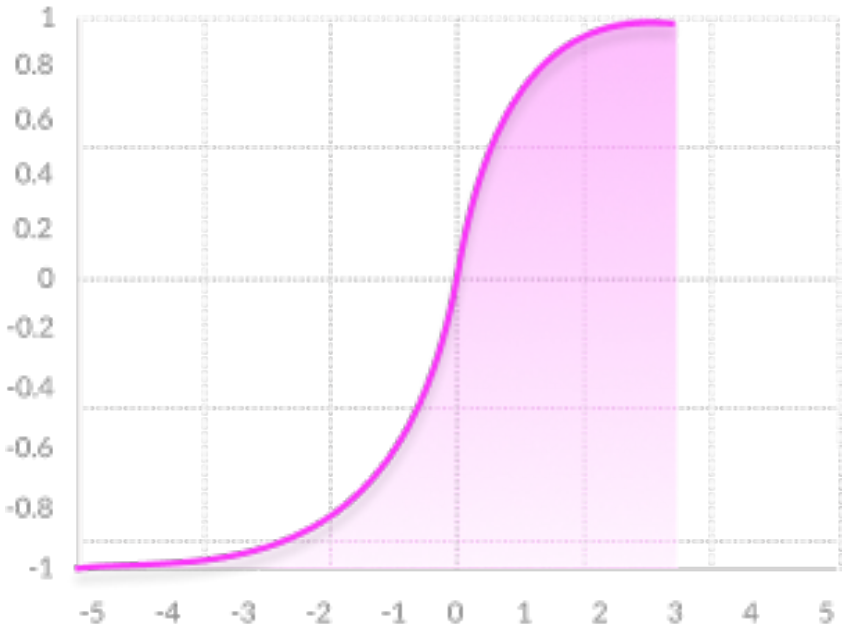


Fonte: https://missinglink.ai/guides/neural-network-concepts/7-types-neural-network-activation-functions-right/

Função tangente hiperbólica

Vantagens Similar à sigmoide; saída normalizada entre -1 e 1; centralizada em zero; gradientes mais fortes

Desvantagens *Vanishing gradients*



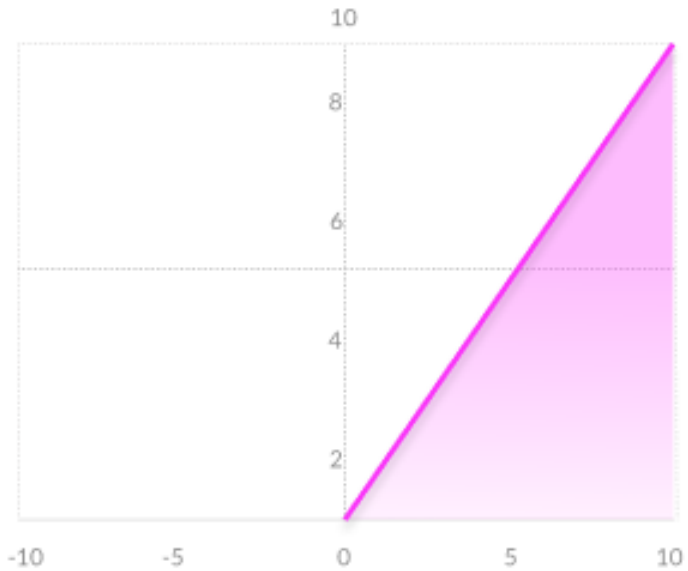
Fonte: https://missinglink.ai/guides/neural-network-concepts/7-types-neural-network-activation-functions-right/

Funções de ativação

RELU (Rectified Linear Unit)

Vantagens Não-linear; computação eficiente; convergência rápida

Desvantagens *Dying RELU problem*

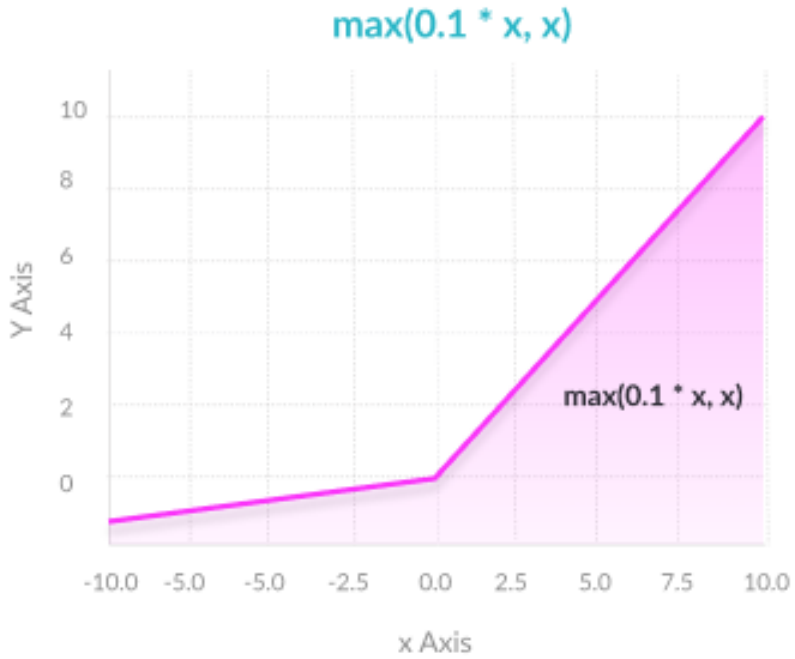


Fonte: https://missinglink.ai/guides/neural-network-concepts/7-types-neural-network-activation-functions-right/

Leaky RELU (Leaky Rectified Linear Unit)

Vantagens Evita *dying RELU problem*

Desvantagens Resultados inconsistentes para valores negativos



Fonte: https://missinglink.ai/guides/neural-network-concepts/7-types-neural-network-activation-functions-right/

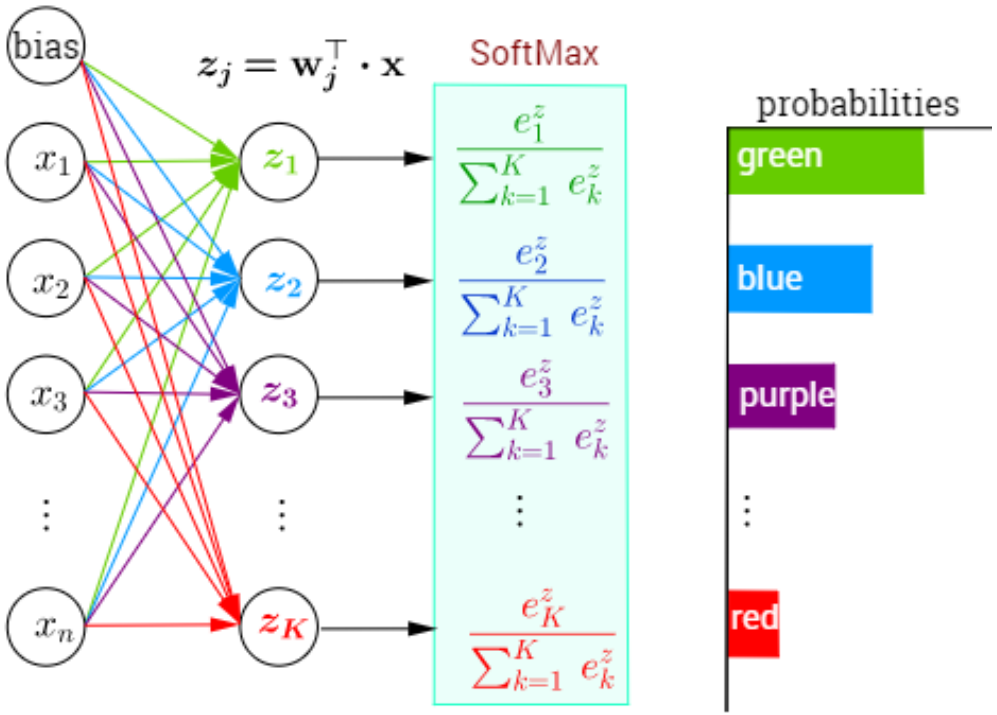
Funções de ativação

Softmax

Classificação Geralmente utilizada na camada de saída para classifição multiclasse

Probabilidade Pode ser usada para representar a probabilidade das classes

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$



Fonte: https://deepnotes.io/

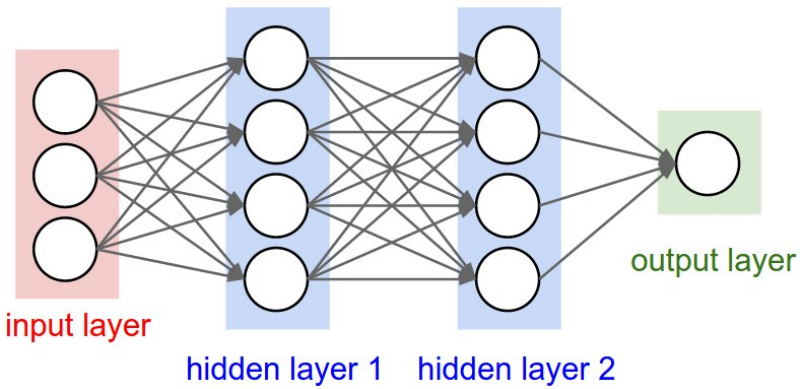
Função de perda

Variedade Novamente, existe uma grande variedade de funções de perda, no entanto, algumas delas são mais comuns

Mean Squared Error

Regressão

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2.$$

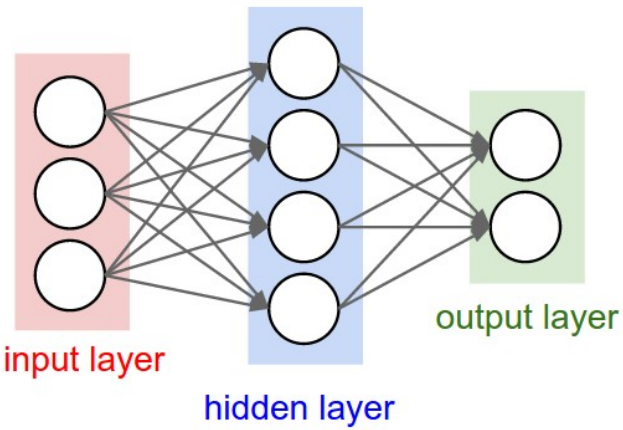


Fonte: http://cs231n.github.io/neural-networks-1/

Cross-Entropy

Classificação

$$CE = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(\epsilon + \hat{y}_i)$$

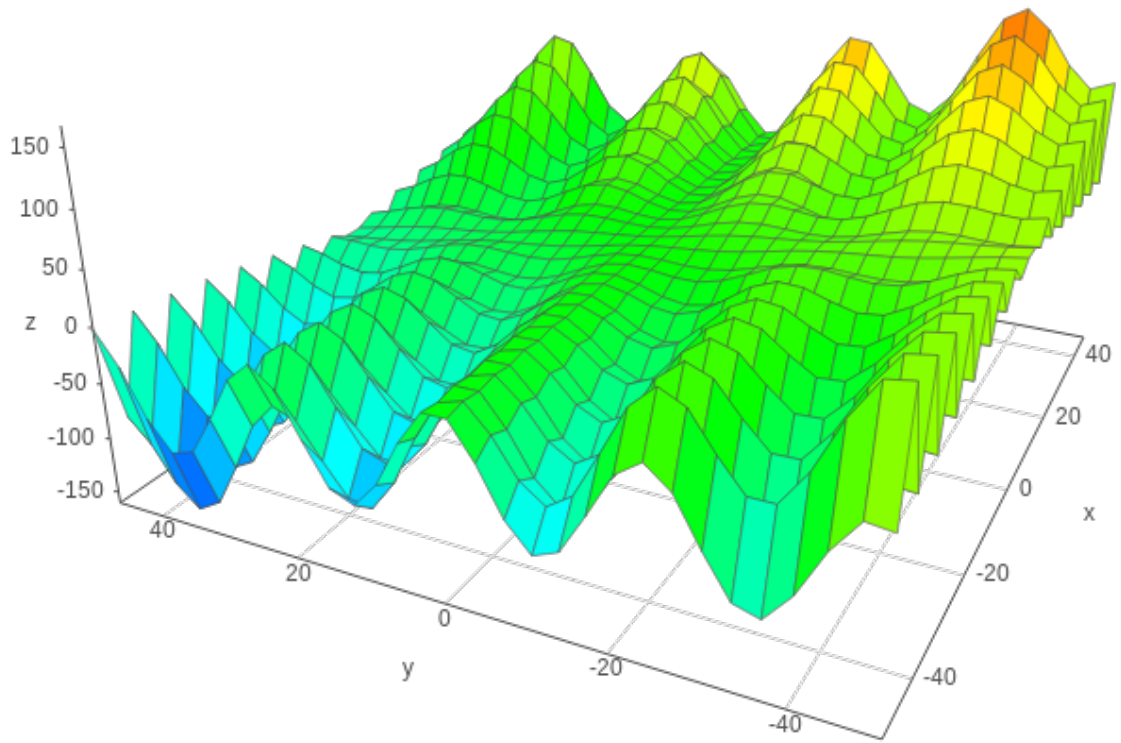


Fonte: http://cs231n.github.io/neural-networks-1/

Algoritmo de otimização

Erro A função de perda $\mathcal{L}(W)$ produz uma superfície de erro

Otimização O algoritmo de otimização busca o conjunto de parâmetros treináveis da rede que minimiza o erro do modelo



Fonte: https://academo.org/demos/3d-surface-plotter

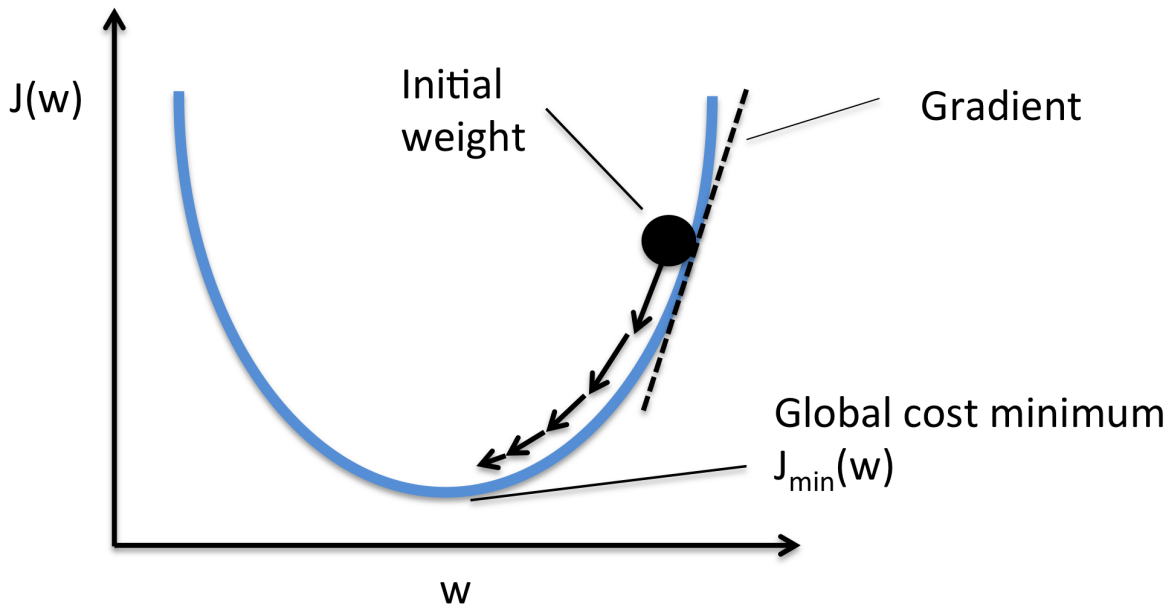
Algoritmo de otimização

Gradient Descent

Clássico Base para diversos algoritmos de otimização

Método Mover-se na direção que proporciona o declínio mais íngreme no valor da função de perda: a direção oposta ao gradiente

Problemas Mínimos locais; valor ideal da taxa de aprendizado



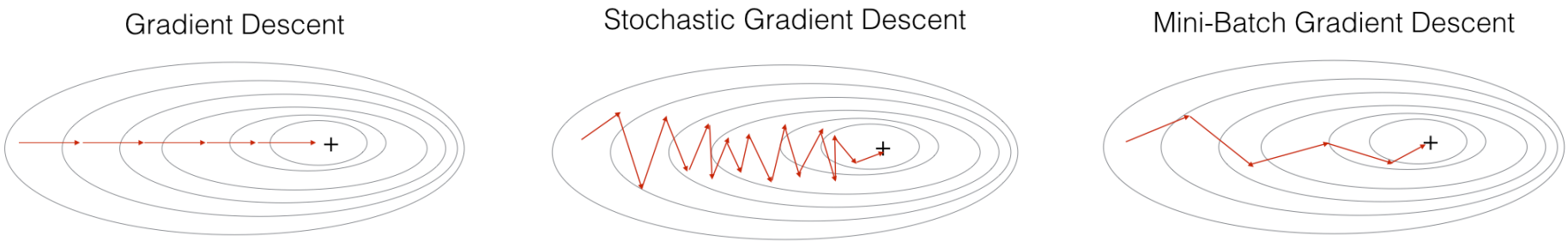
Fonte: https://sebastianraschka.com/faq/docs/closed-form-vs-gd.html

Algoritmo de otimização

Vanilla Algoritmo clássico, atualização dos pesos ocorre a cada época

Stochastic Atualização dos pesos ocorre a cada nova amostra

Mini-Batch Atualização dos pesos ocorre a cada grupo de K amostras



Fonte: https://lovesnowbest.site/2018/02/16/Improving-Deep-Neural-Networks-Assignment-2/

Algoritmo de otimização

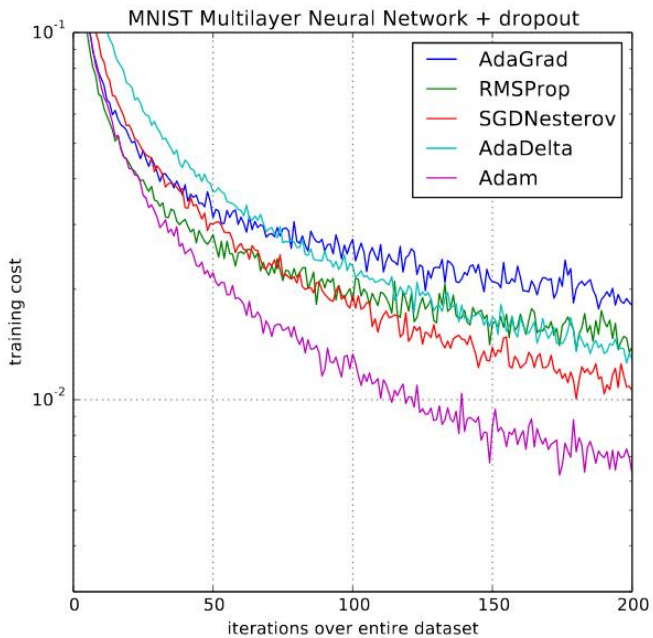
$\alpha \rightarrow$ taxa de aprendizado: influencia a escala das atualizações dos pesos

$\mu \rightarrow$ momentum: influencia a direção das atualizações dos pesos

Adagrad Atualiza α a cada iteração com base no acúmulo dos gradientes passados $\rightarrow$ decaimento de α

Adadelta Atualiza α a cada iteração com base no acúmulo de um determinado número de gradientes passados

Adam Atualiza α e μ durante o treinamento

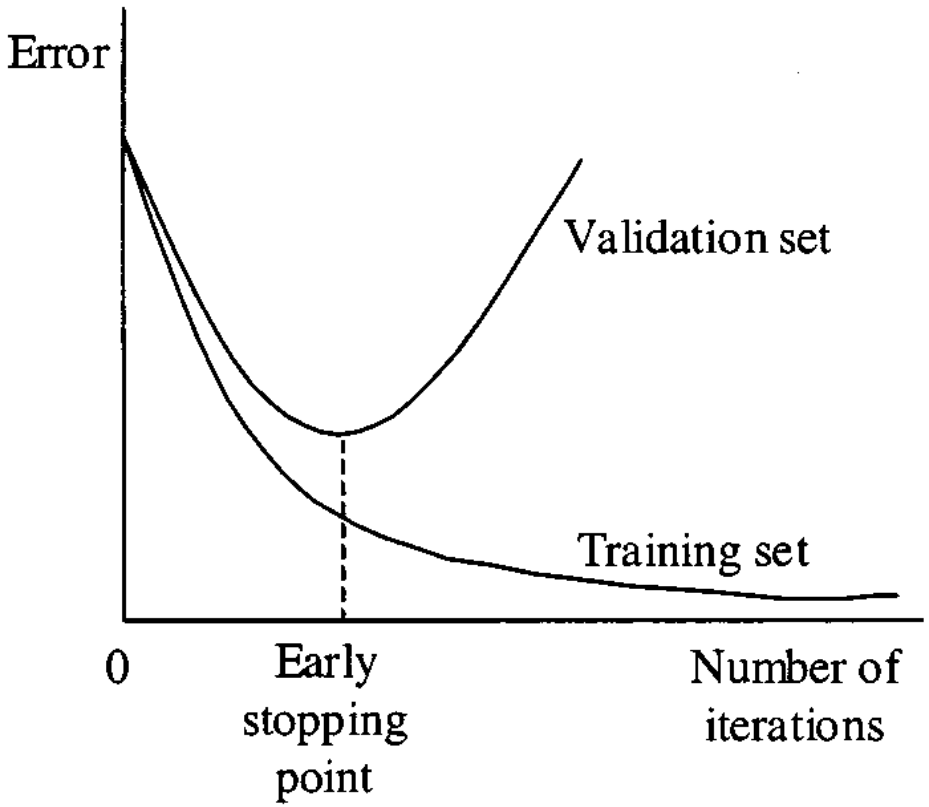


Fonte: https://arxiv.org/pdf/1412.6980.pdf

Critério de parada

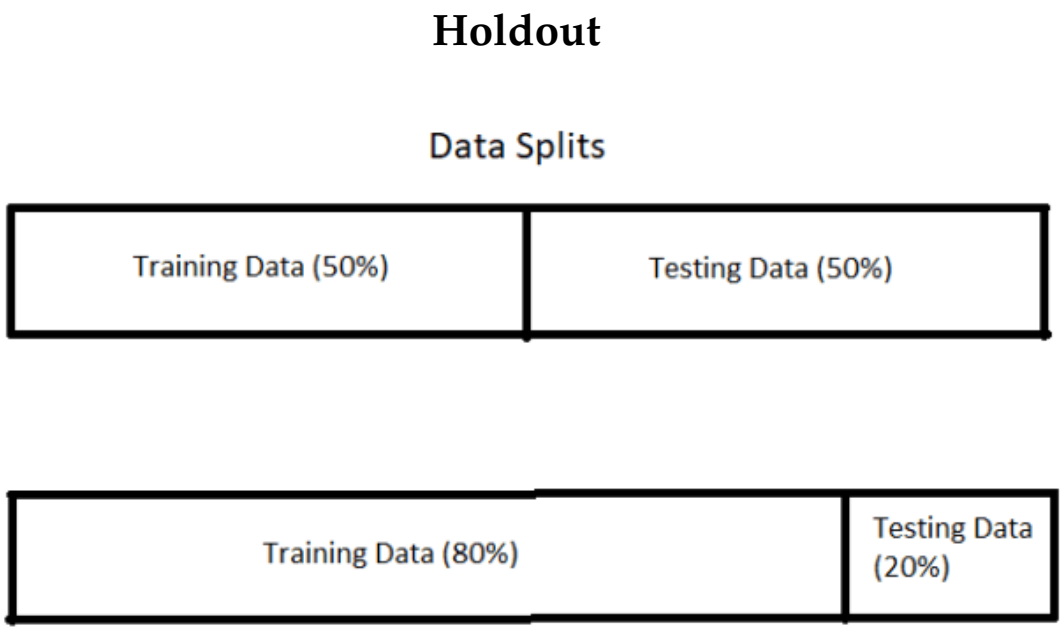
Número fixo de épocas Pode ocorrer *overfitting* ou *underfitting*

Parada antecipada Utiliza um conjunto de validação durante treinamento para avaliar quando a variação do erro começa a decrescer



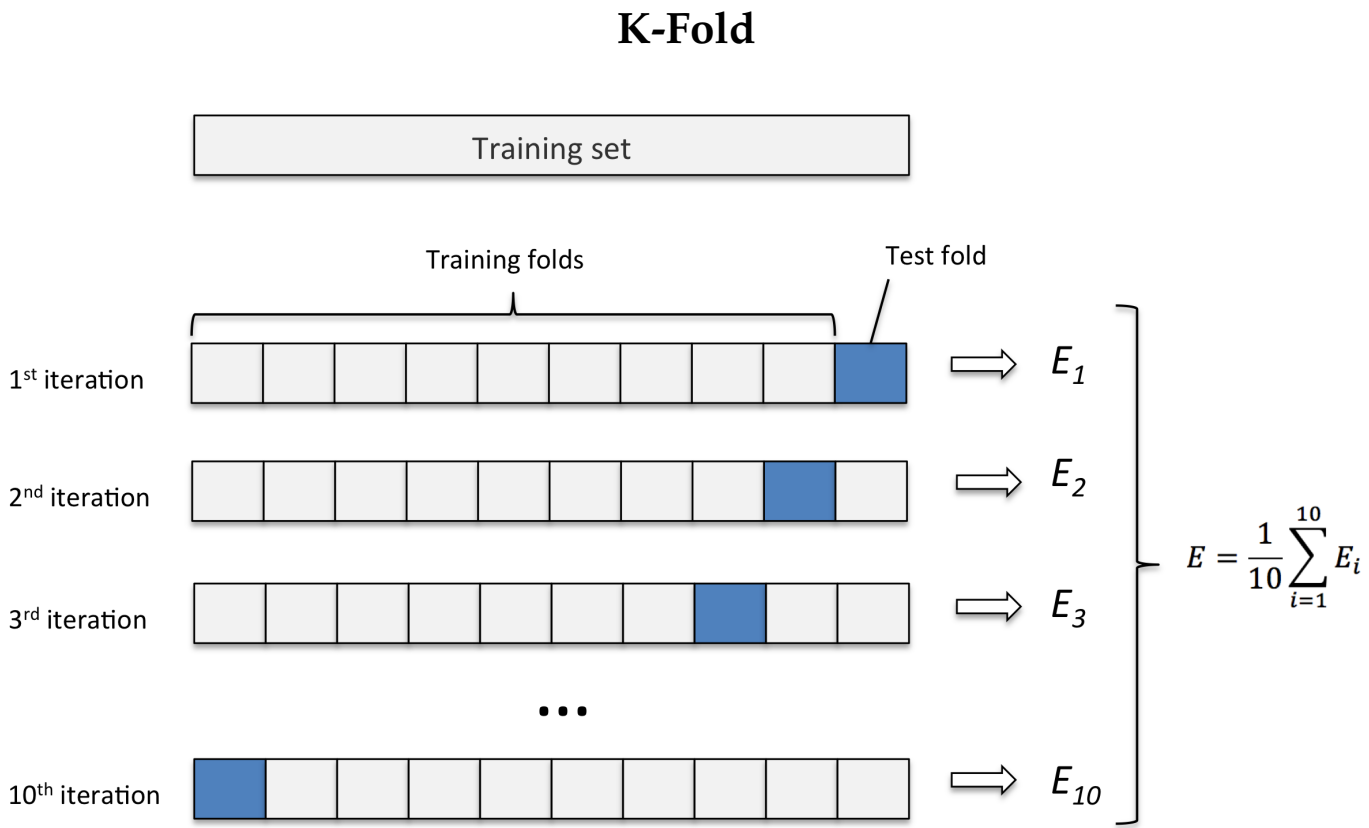
Fonte: https://www.researchgate.net/figure/Early-stopping-based-on-cross-validation_fig1_3302948

Validação



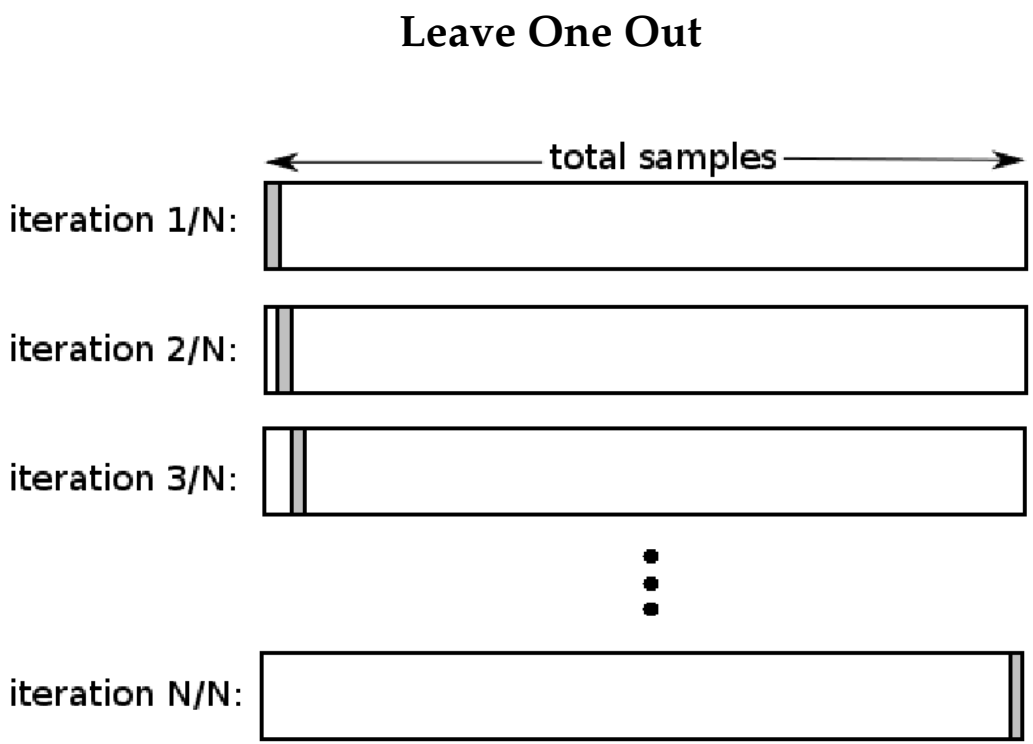
Fonte: http://thatdatatho.com/2018/10/11/detailed-introduction-cross-validation-machine-learning/

Validação



Fonte: http://karlrosaen.com/ml/learning-log/2016-06-20/

Validação



Fonte: https://www.researchgate.net/figure/Leave-One-Out-Cross-Validation_fig11_266617511