

#1 Meetup Machine Learning – GDG Fortaleza

Perceptron: Deep Learning



Google Developer Groups

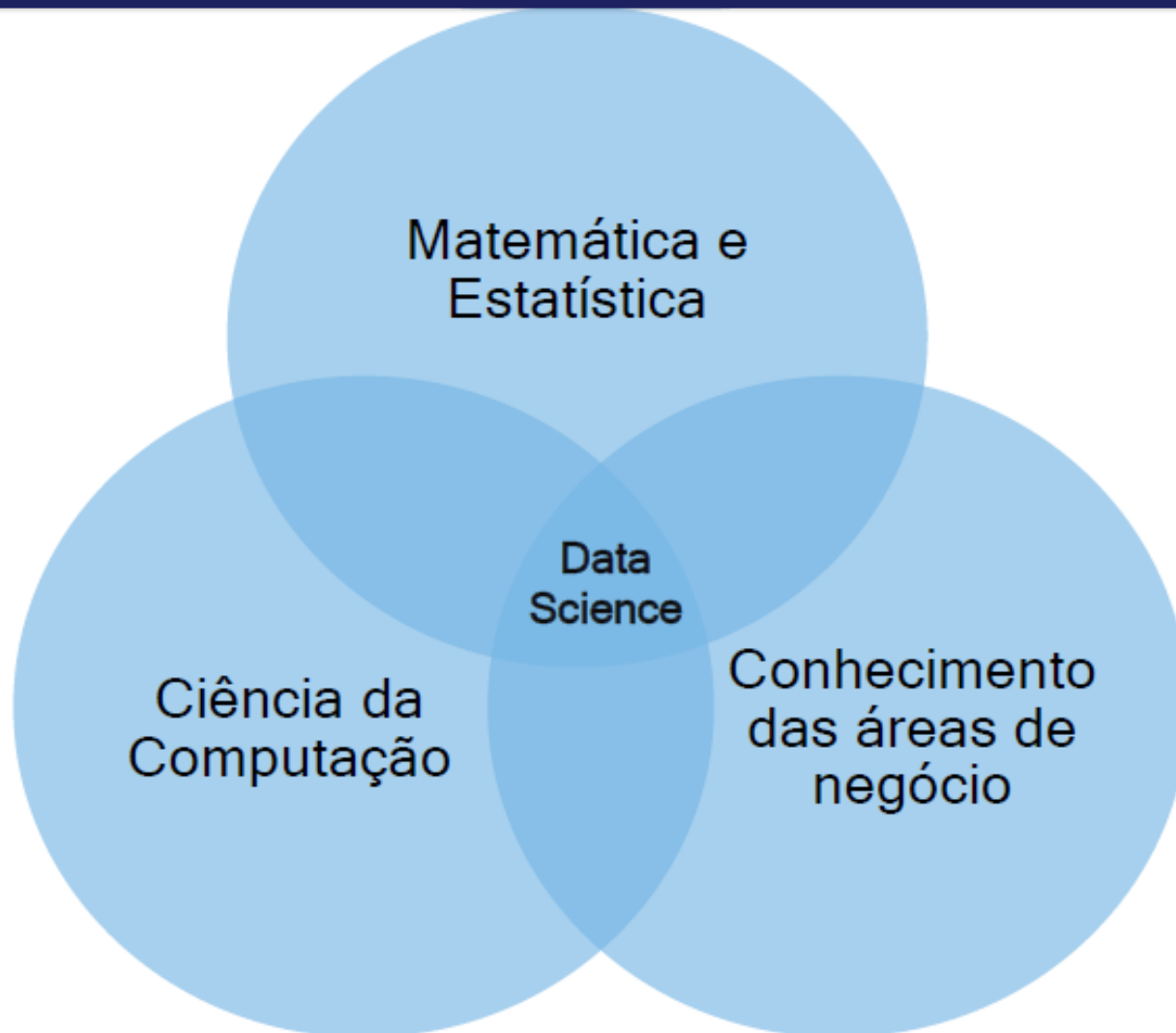
Daniela Lage
@daniela.olage

“Está relacionado com a construção de programas de computador que automaticamente aprendam com experiências, sem necessidade de serem explicitamente programados.”

(Machine Learning, Tom M. Mitchell)



ÁREAS DE CONHECIMENTO CIÊNCIA DE DADOS

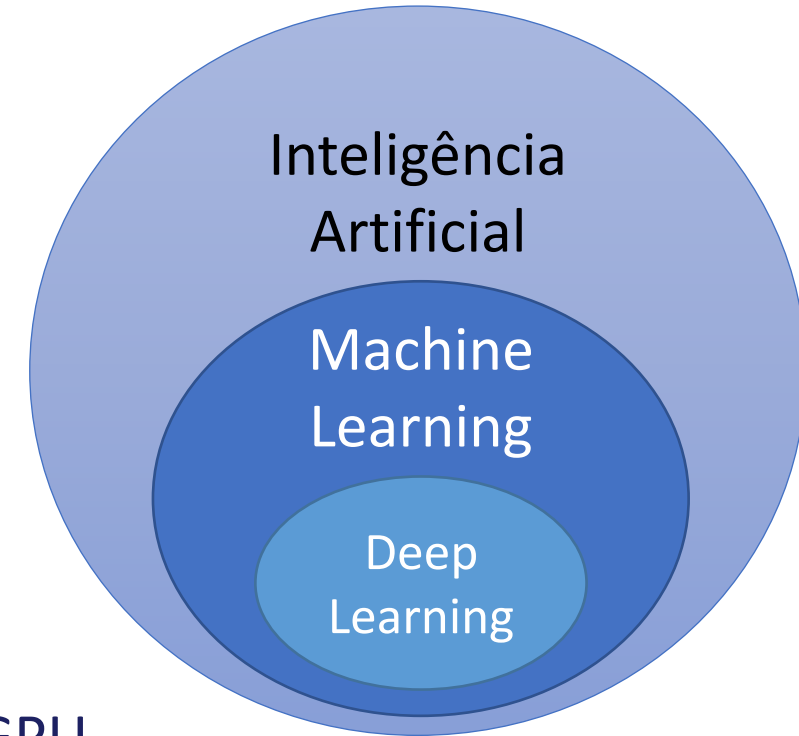


- Arquiteto de Dados (Data Architect)
- Analista de Business Intelligence (BI)
- Analista de Dados (Data Analyst)
- Analista de Machine Learning
- Cientista de Dados (Data Scientist)
- Engenheiro de Dados (Data Engineer)

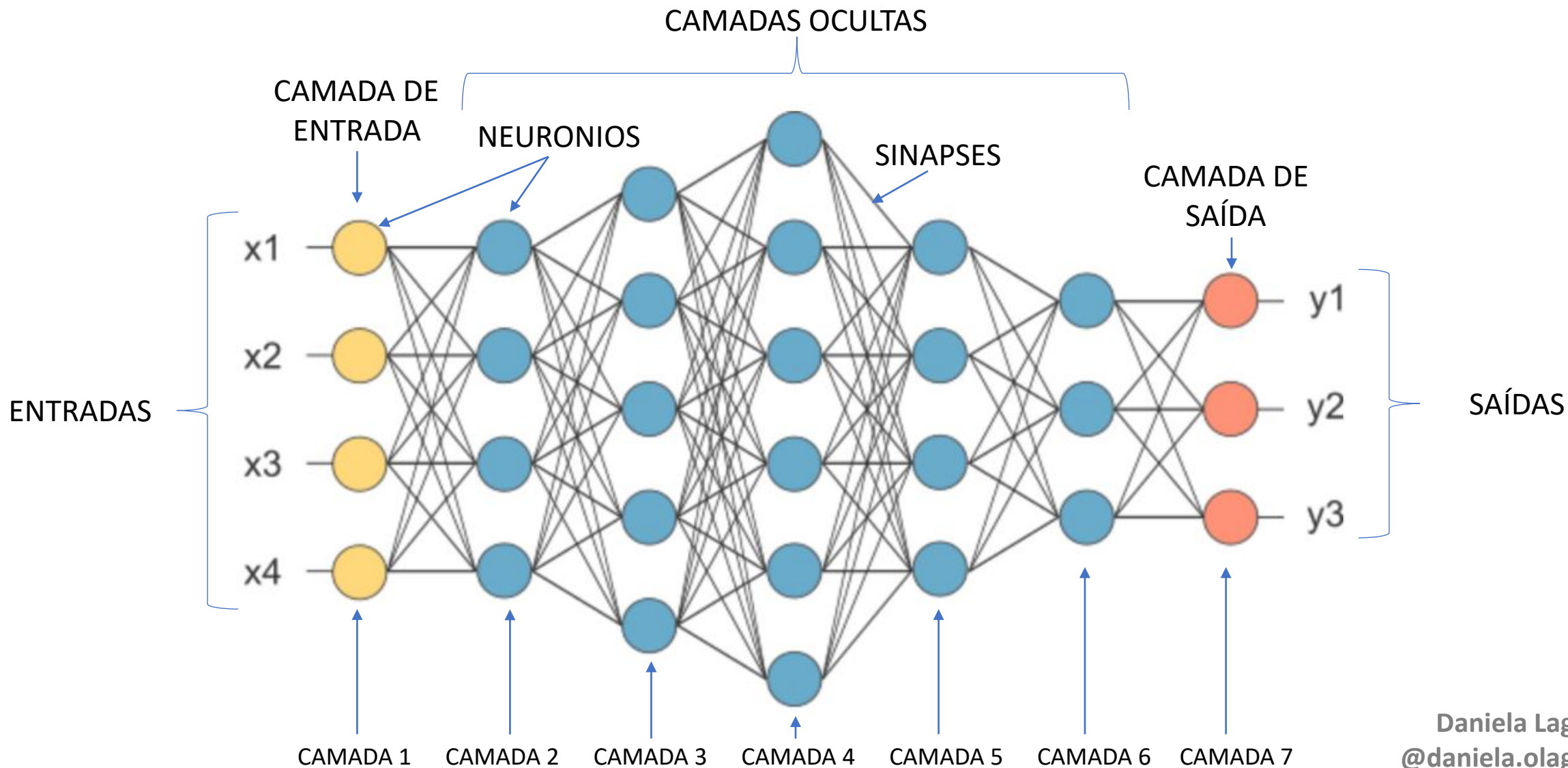


- **SUPERVISIONADO:** Conhece a entrada e a saída (rótulos ou classes)
- **NÃO SUPERVISIONADO:** Conhece a entrada mas não a saída de dados, podendo usar do recurso *clustering* para agrupamentos
- **SEMI SUPERVISIONADO ou POR REFORÇO:** Conhece a entrada e algumas saídas

- É um tipo de aprendizado de máquina.
- Ele consegue extrair as *features* ou características de entrada, de forma automática.
- Precisa de mais dados, *hardware* para processamento GPU e leva mais tempo para ser construída.
- Dispensa o especialista no domínio do problema.



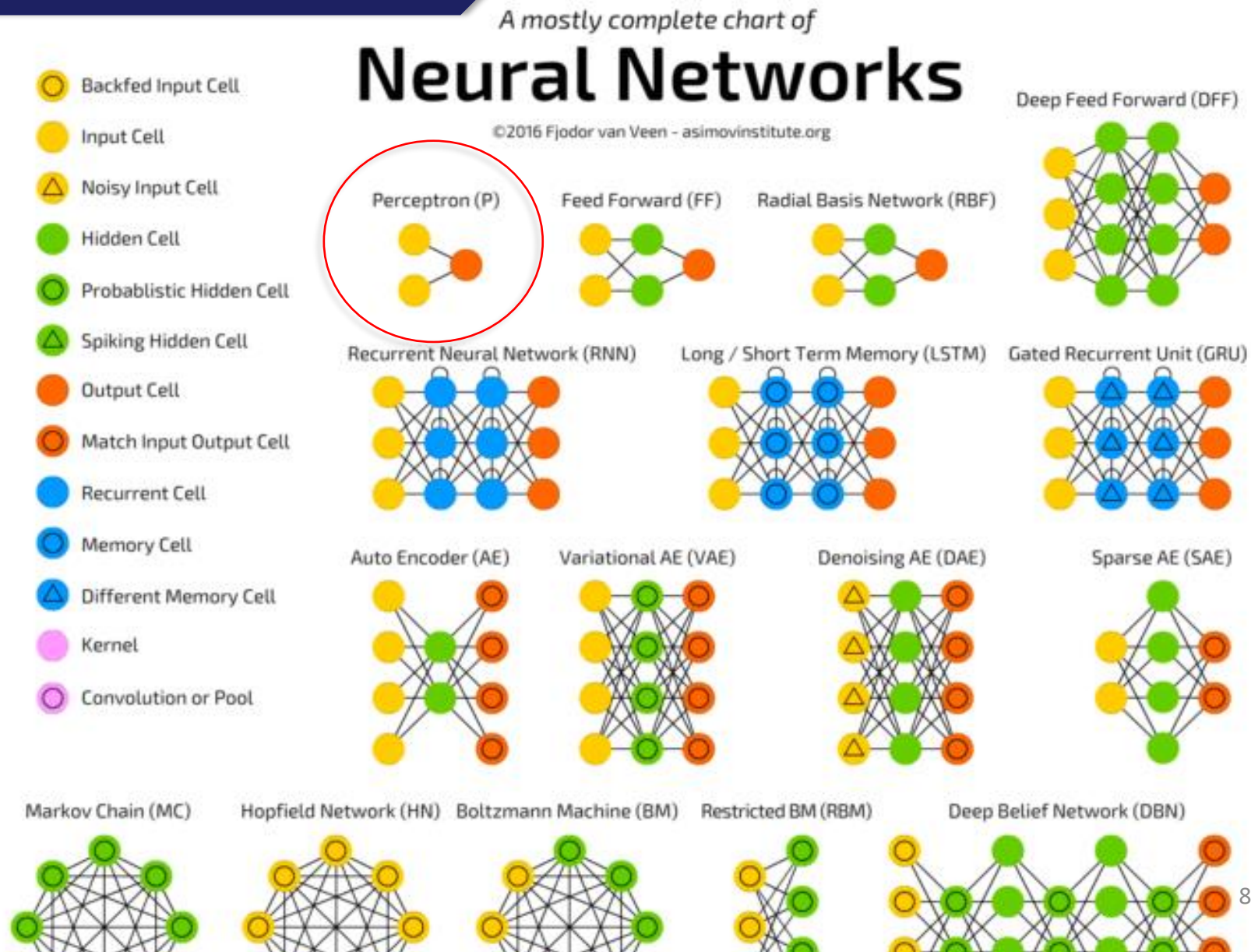
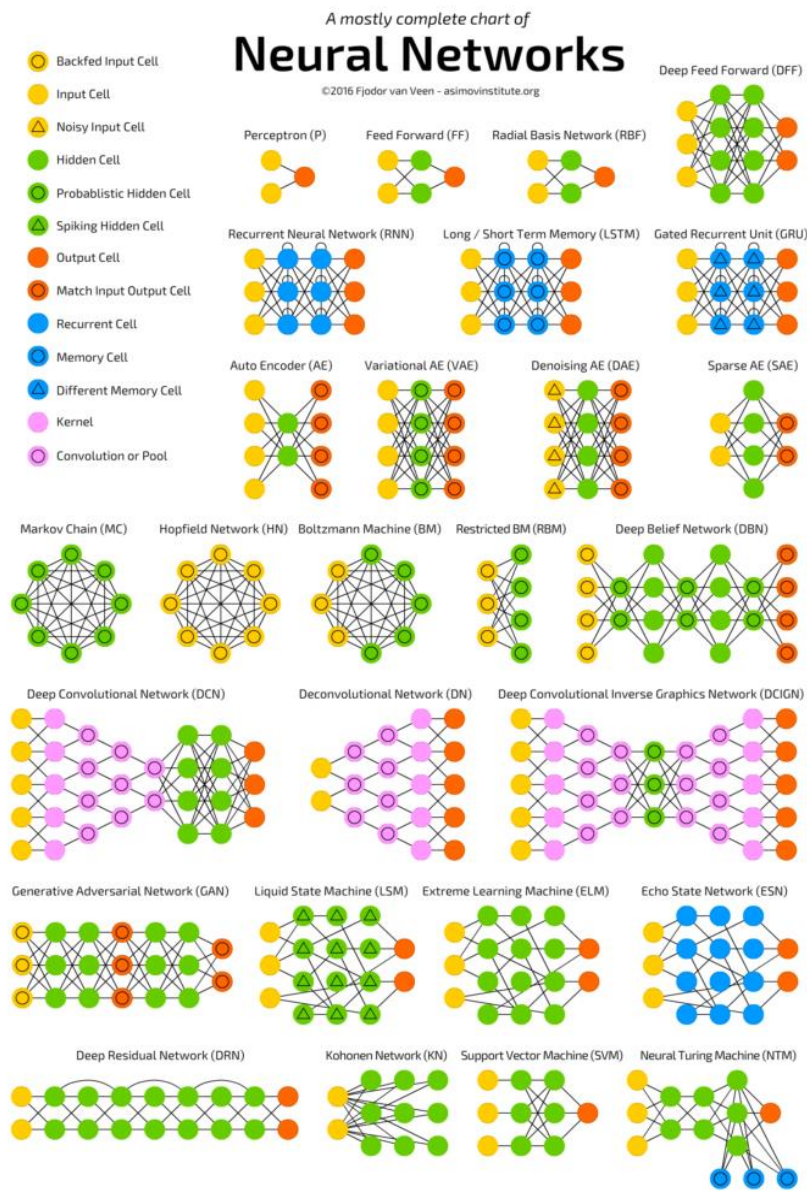
REDE NEURAL ARTIFICIAL (RNA)



TENSORFLOW PLAYGROUND

- <https://playground.tensorflow.org/>

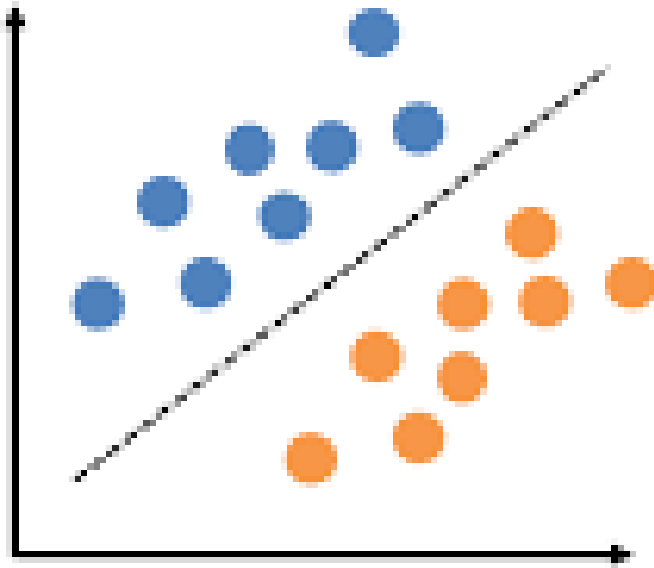
REDE NEURAL ARTIFICIAL (RNA)



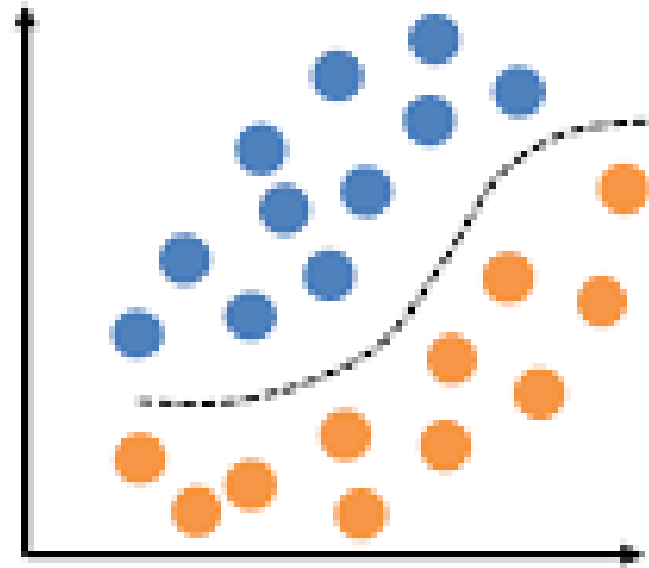
MODELO PERCEPTRON DE RNA

- É considerado como classificador linear binário

Linear



Nonlinear



MODELO PERCEPTRON DE RNA

- Primeiro modelo proposto de RNA inventado em 1957 por Frank Rosenblatt no Cornell Aeronautical Laboratory. Ele pode ser visto como o tipo mais simples de rede neural *feed-forward*: um classificador linear.
- Contem apenas uma única camada e com um único neurônio do tipo MCP
- Implementa uma forma de atualizar o peso da *feature* ou *input* além do *bias*.

MODELO PERCEPTRON DE RNA

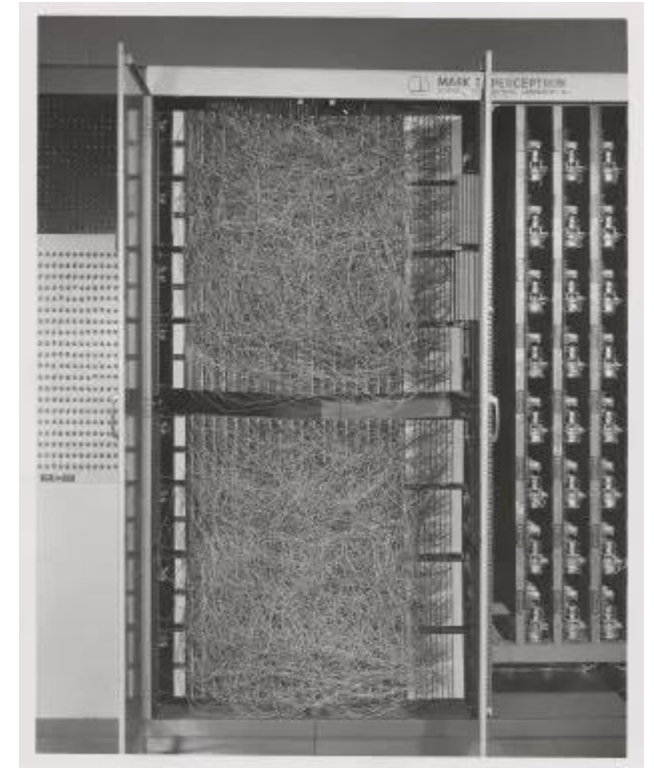
- Pode ter N entradas, mas apenas uma única saída com um valor de 0 e 1 ou de -1 e 1. (Função degrau ou Função sinal)
- Correções dos pesos sinápticos e limiar do Perceptron, através do processo de treinamento supervisionado
- Regra de aprendizado de Hebb, utilizada em cada amostra dos sinais de entrada.

MARK I PERCEPTRON

Primeira máquina projetada para **reconhecimento de imagem**, com implementação do algoritmo.

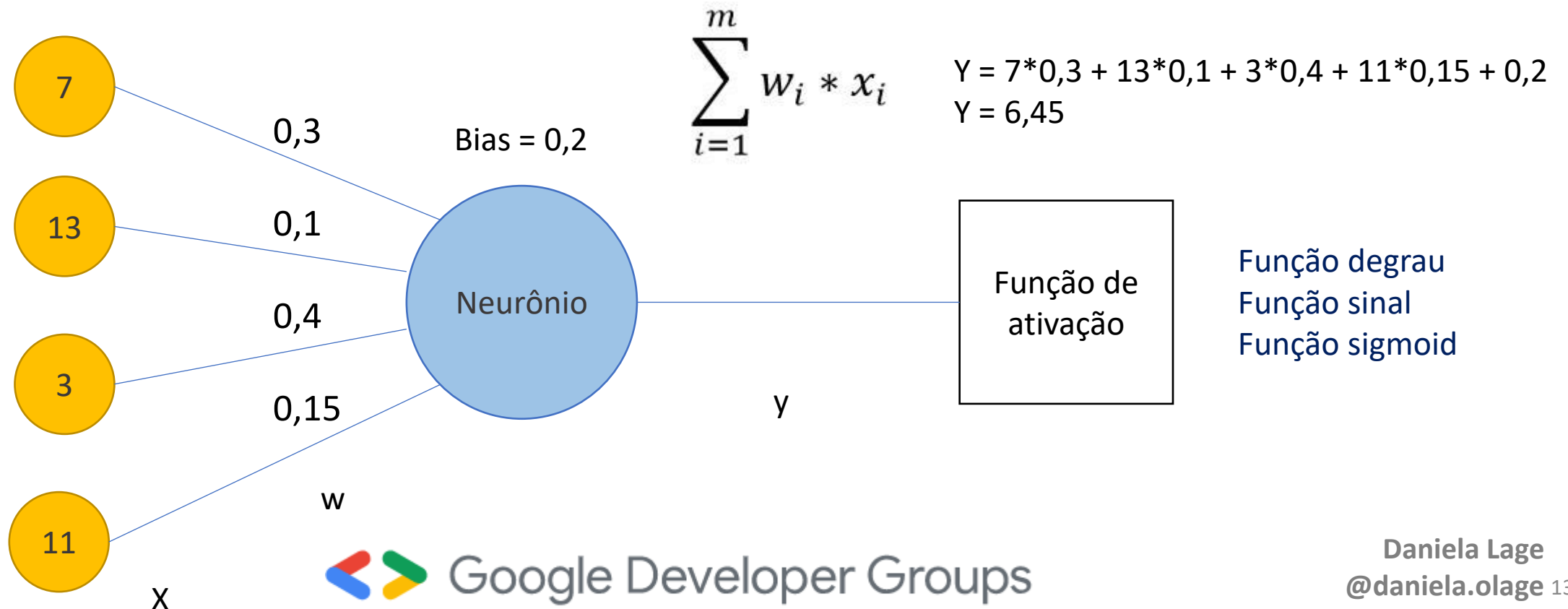
Tinha uma matriz de 400 fotocélulas, conectadas aleatoriamente aos “neurônios”.

Seus pesos eram codificados em potenciômetros, e as atualizações, durante a aprendizagem, eram realizadas por motores elétricos.



REDE NEURAL ARTIFICIAL (RNA)

- Rede Neural Artificial (RNA) 1943 primeira referência e 1975 primeira proposta.



FUNÇÃO DE ATIVAÇÃO

Função binária que testa o valor gerando resultados distintos para cada condição, sendo as condições: maior que o parâmetro de comparação (1) e menor que o parâmetro (-1).

Esse parâmetro de comparação geralmente é chamado de limiar (em inglês, *threshold*).

Por fim, o Perceptron libera como saída (usualmente chamado de output) a previsão.

tipos

- **TAXA DE APRENDIZADO (*LEARNING RATE* OU *ALPHA*):** número representativo de quanto um erro vai interferir no aprendizado. Quanto maior, mais rápido aprendizado, porém pode passar do ponto ótimo. Já uma taxa menor, diminui a velocidade do treino, mas aumenta a precisão da busca.
- **TAXA DE ITERAÇÕES:** repetições que serão feitas.
- **ÉPOCA:** cada iteração

- **ERRO DAS PREVISÕES:** resultado do seguinte cálculo =>
valor esperado ou classe esperada – previsão ou resultado

Os erros calculados anteriormente são utilizados para atualizar o peso das entradas. Suponha que eu tenho um input x e um erro r , para cada x é somado $x*r$, ou seja, $x + x*r$. O primeiro input geralmente é o bias (em português, viés).



FORMULA DE AJUSTE DOS PESOS E BIAS PERCEPTRON

$$W_i^{atual} = W_i^{anterior} + \eta * (d^{(k)} - y) * X^{(k)}$$

Onde:

$$\theta_i^{atual} = \theta_i^{anterior} + \eta * (d^{(k)} - y) * X^{(k)}$$

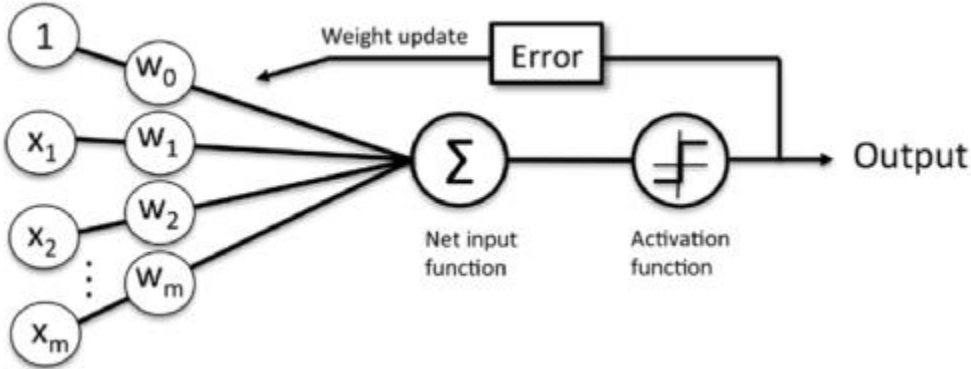
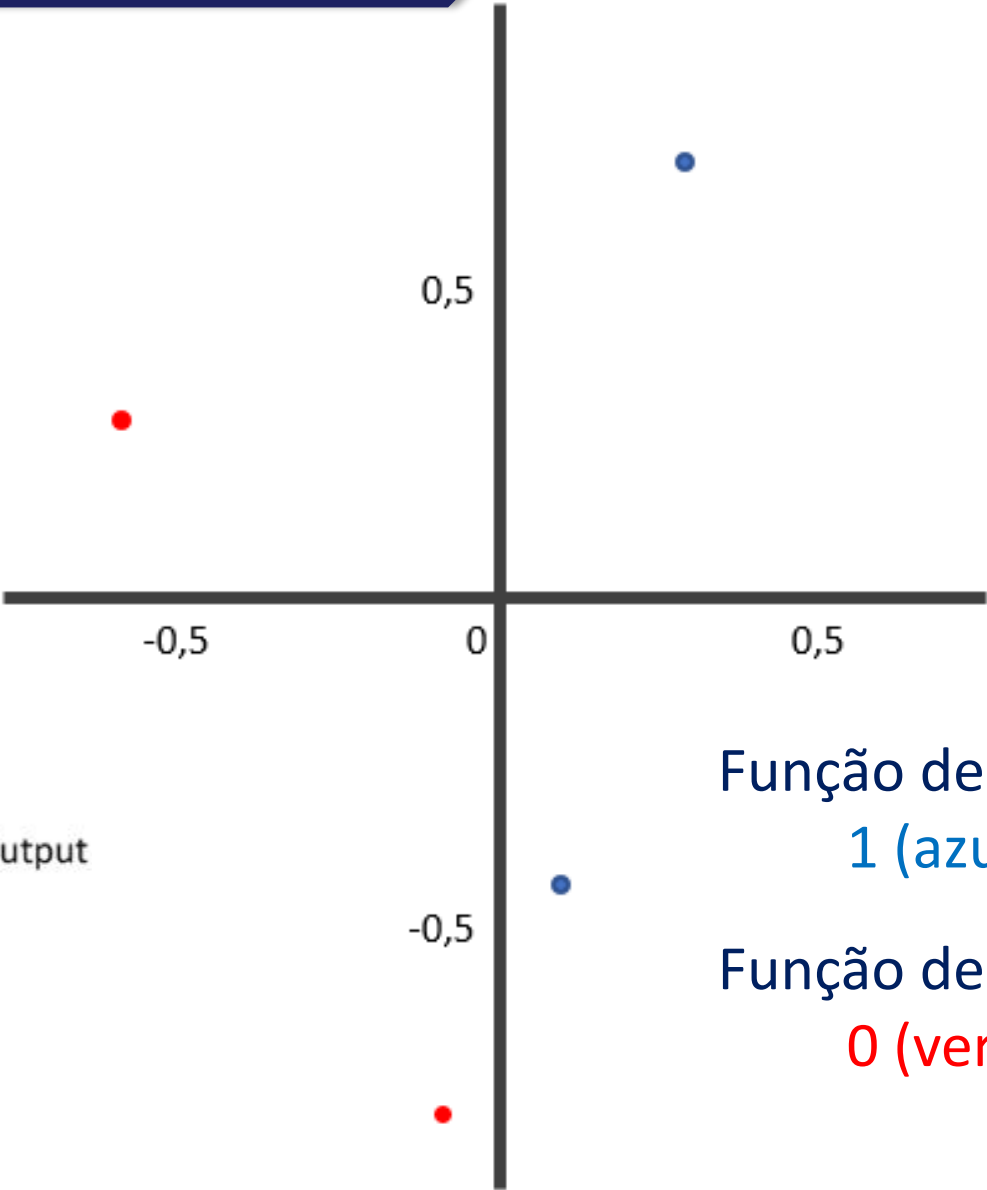
- **W** é o peso
- **Θ** é o bias
- **η** é a taxa de aprendizado, sendo que o valor de η deverá estar compreendido entre $0 < \eta < 1$;
- **K** representa a k-ésima amostra.
- **d** é a classe esperada
- **y** é o resultado ou *output* gerado pelo *perceptron*
- **X** é a entrada ou *inp*



Google Developer Groups

EXEMPLO

x	y	Classe
0,3	0,7	1
-0,6	0,3	0
-0,1	-0,8	0
0,1	-0,45	1



Função de ativação ≥ 0
1 (azul)

Função de ativação < 0
0 (vermelho)

1ª ITERAÇÃO

- Pesos aleatórios: 0,8 e -0,5
- Taxa de aprendizado: 0,5
- Usaremos um limiar de -1 ou 1 para atualizarmos os pesos
- **Primeira iteração:** $(X1 * W1 + X2 * W2)$ $0,3 * 0,8 + 0,7 * -0,5 = -0,11$
- **Erro:** classe esperada – resultado da função ativada => $1 - 0 = 1$

x	y	Classe
0,3	0,7	1
-0,6	0,3	0
-0,1	-0,8	0
0,1	-0,45	1

Função de ativação ≥ 0 1
Função de ativação < 0 0

Novos pesos:

- $W1 = 0,8 + 0,5 * 1 * 0,3 = 0,95$
- $W2 = -0,5 + 0,5 * 1 * 0,7 = -0,15$

$$W_i^{atual} = W_i^{anterior} + \eta * (d^{(k)} - y) * X^{(k)}$$
$$\theta_i^{atual} = \theta_i^{anterior} + \eta * (d^{(k)} - y) * X^{(k)}$$

2ª ITERAÇÃO

- Pesos aleatórios: 0,95 e -0,15
- Taxa de aprendizado: 0,5
- Usaremos um limiar de -1 ou 1 para atualizarmos os pesos
- **Segunda iteração:** $(X1 * W1 + X2 * W2)$ $-0,6 * 0,95 + 0,3 * (-0,15) = -0,615$
- **Erro:** classe esperada – resultado da função ativada $\Rightarrow 0 - 0 = 0$

x	y	Classe
0,3	0,7	1
-0,6	0,3	0
-0,1	-0,8	0
0,1	-0,45	1

Função de ativação ≥ 0 1
Função de ativação < 0 0

Classificou corretamente, sendo assim, não há motivo para ajustar os pesos.

3ª ITERAÇÃO

- Pesos aleatórios: 0,95 e -0,15
- Taxa de aprendizado: 0,5
- Usaremos um limiar de -1 ou 1 para atualizarmos os pesos
- **Terceira iteração:** $(X_1 * W_1 + X_2 * W_2)$ $-0,1 * 0,95 + (-0,8) * (-0,15) = 0,025$
- **Erro:** classe esperada – resultado da função ativada $\Rightarrow 0 - 1 = -1$

x	y	Classe
0,3	0,7	1
-0,6	0,3	0
-0,1	-0,8	0
0,1	-0,45	1

Função de ativação ≥ 0 1

Função de ativação < 0 0

Novos pesos:

- $W_1 = 0,95 + 0,5 * -1 * -0,1 = 1,0$
- $W_2 = -0,15 + 0,5 * -1 * -0,8 = -0,25$

$$W_i^{atual} = W_i^{anterior} + \eta * (d^{(k)} - y) * X^{(k)}$$
$$\theta_i^{atual} = \theta_i^{anterior} + \eta * (d^{(k)} - y) * X^{(k)}$$

4ª ITERAÇÃO

- Pesos aleatórios: 1,0 e 0,25
- Taxa de aprendizado: 0,5
- Usaremos um limiar de -1 ou 1 para atualizarmos os pesos
- **Quarta iteração:** $(X1 * W1 + X2 * W2)$ $-0,1 * 1,0 + -0,8 * 0,25 = -0,125$
- **Erro:** classe esperada – resultado da função ativada $\Rightarrow 1 - 0 = 1$

x	y	Classe
0,3	0,7	1
-0,6	0,3	0
-0,1	-0,8	0
0,1	-0,45	1

Função de ativação ≥ 0 1

Função de ativação < 0 0

Novos pesos:

- $W1 = 1,0 + 0,5 * 1 * 0,1 = 1,05$
- $W2 = 0,25 + 0,5 * 1 * -0,45 = -0,025$

$$W_i^{atual} = W_i^{anterior} + \eta * (d^{(k)} - y) * X^{(k)}$$
$$\theta_i^{atual} = \theta_i^{anterior} + \eta * (d^{(k)} - y) * X^{(k)}$$

TESTE

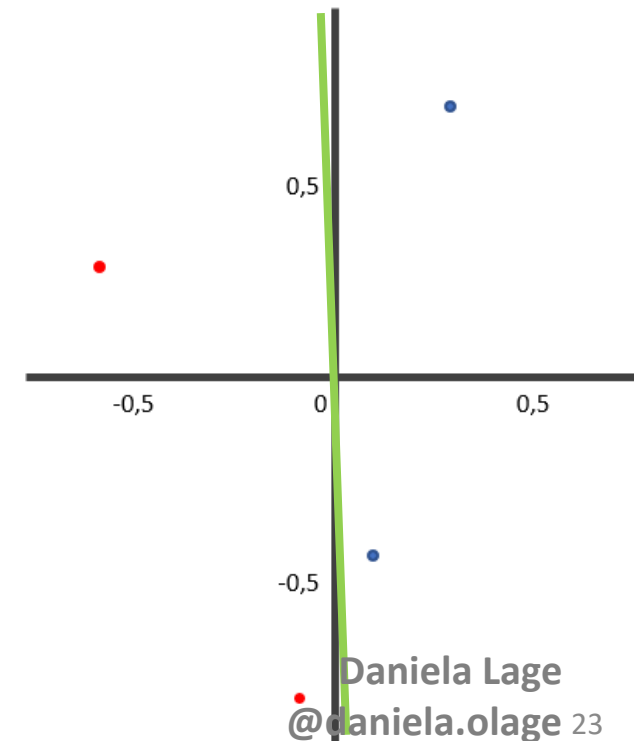
x	y	Classe
0,3	0,7	1
-0,6	0,3	0
-0,1	-0,8	0
0,1	-0,45	1

Pesos: 1,05 e 0,025

Função de ativação ≥ 0 1

Função de ativação < 0 0

- 1ª ITERAÇÃO: $0,3 * 1,05 + 0,7 * 0,025 = 0,333 \Rightarrow 1$
- 2ª ITERAÇÃO: $-0,6 * 1,05 + 0,3 * 0,025 = -0,62 \Rightarrow 0$
- 3ª ITERAÇÃO: $-0,1 * 1,05 + -0,8 * 0,025 = -0,13 \Rightarrow 0$
- 4ª ITERAÇÃO: $0,1 * 1,05 + -0,45 * 0,025 = 0,097 \Rightarrow 1$



[https://github.com/danielaolage/machineLearning/blob/master/GDGFortaleza Perceptron.ipynb](https://github.com/danielaolage/machineLearning/blob/master/GDGFortaleza%20Perceptron.ipynb)

<https://www.igti.com.br/>

- CIÊNCIA DE DADOS
- ENGENHARIA DE DADOS
 - DEEP LEARNING
- APRENDIZADO DE MÁQUINA
- VISÃO COMPUTACIONAL
 - DATA ANALYTICS



BIBLIOGRAFIAS

ALVES, Ricardo B. Redes Neurais e Perceptron IGTI 2020. Disponível em: <<https://www.igti.com.br/>>. Acesso em: 24 de jan. de 2022.

BOOK, Deep Learning. Perceptron 2021. Disponível em: <<https://www.deeplearningbook.com.br/>>. Acesso em: 20 de jan. de 2022.

MONTEIRO, André L. R. Fundamentos de Deep Learning IGTI 2021. Disponível em: <<https://www.igti.com.br/>>. Acesso em: 18 de jan. de 2022.

PALMIERE, Sérgio E. Rede Perceptron de uma única camada – Embarcados 2016. Disponível em: <<https://www.embarcados.com.br/rede-perceptron-de-uma-unica-camada/>>. Acesso em: 23 de jan. de 2022.

VINÍCIUS. Perceptron – redes neurais. Monolitonimbus 2017. Disponível em: <<https://www.monolitonimbus.com.br/perceptron-redes-neurais/>>. Acesso em: 24 de jan. de 2022.

WIKIPEDIA. Perceptron. Disponível em: <<https://pt.wikipedia.org/wiki/Perceptron/>>. Acesso em: 22 de jan. de 2022.