

“Uma inteligência artificial geral pode significar o fim da raça humana ... ela se refinaria a um ritmo cada vez maior. Os humanos não poderiam competir e seriam ultrapassados” (Stephen Hawking).

# Redes Neurais

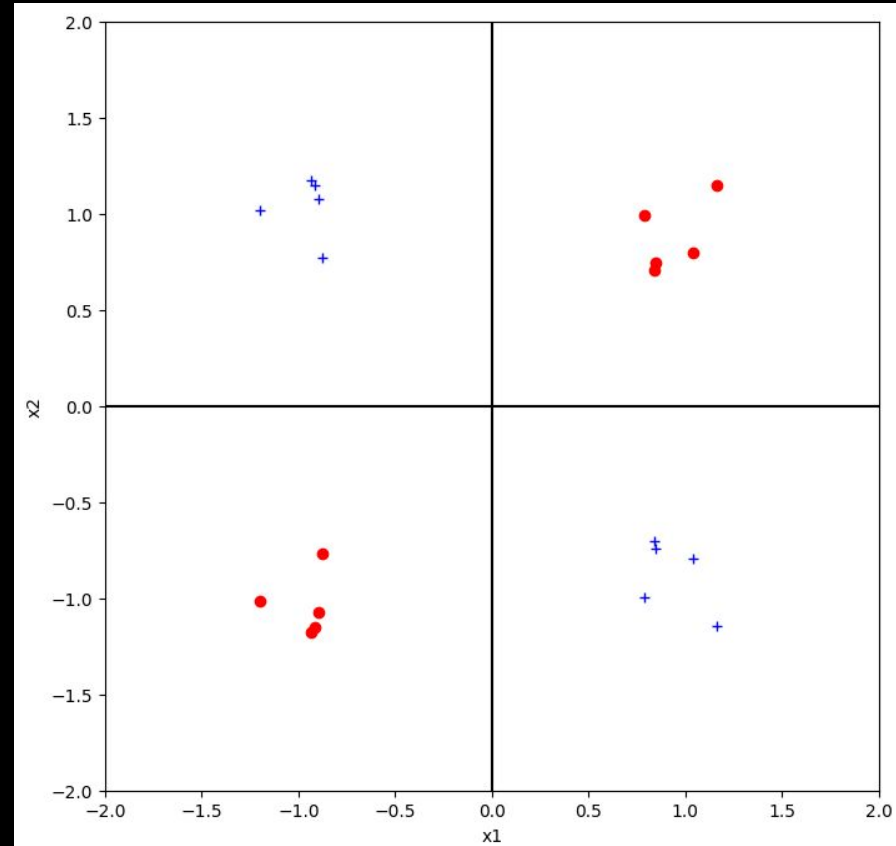
Paulo Ricardo Lisboa de Almeida



# Pergunta

Considere os dados do problema binário XOR.

É possível criar um perceptron que separa os dados corretamente?

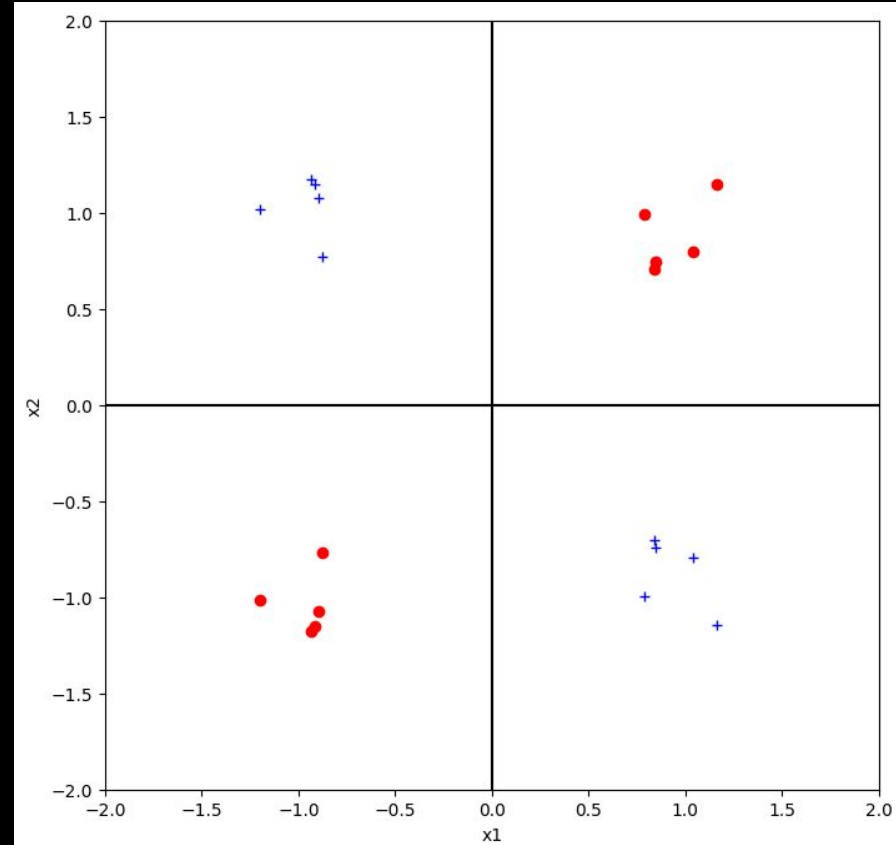


# Pergunta

Considere os dados do problema binário XOR.

É possível criar um perceptron que separa os dados corretamente?

**Não.** Um perceptron é capaz de criar apenas uma fronteira linear.



# Função degrau

Lembrando que a função de ativação de um perceptron é a Função Degrau.

$$f(a) = \begin{cases} +1, & \text{se } a \geq 0 \\ -1, & \text{se } a < 0 \end{cases}$$

A função de ativação injeta uma **não linearidade**, também chamada de **função de transferência**.

# Função degrau

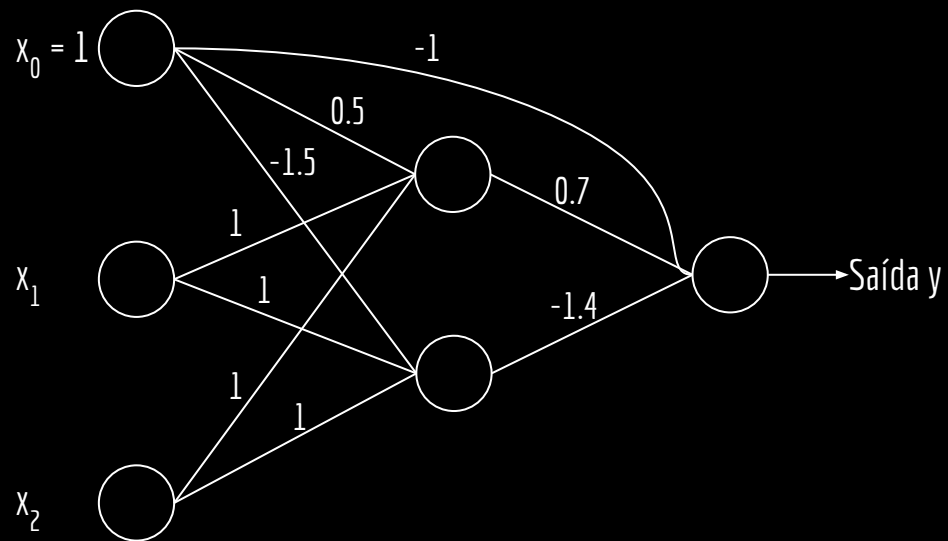
Lembrando que a função de ativação de um perceptron é a Função Degrau.

$$f(a) = \begin{cases} +1, & \text{se } a \geq 0 \\ -1, & \text{se } a < 0 \end{cases}$$

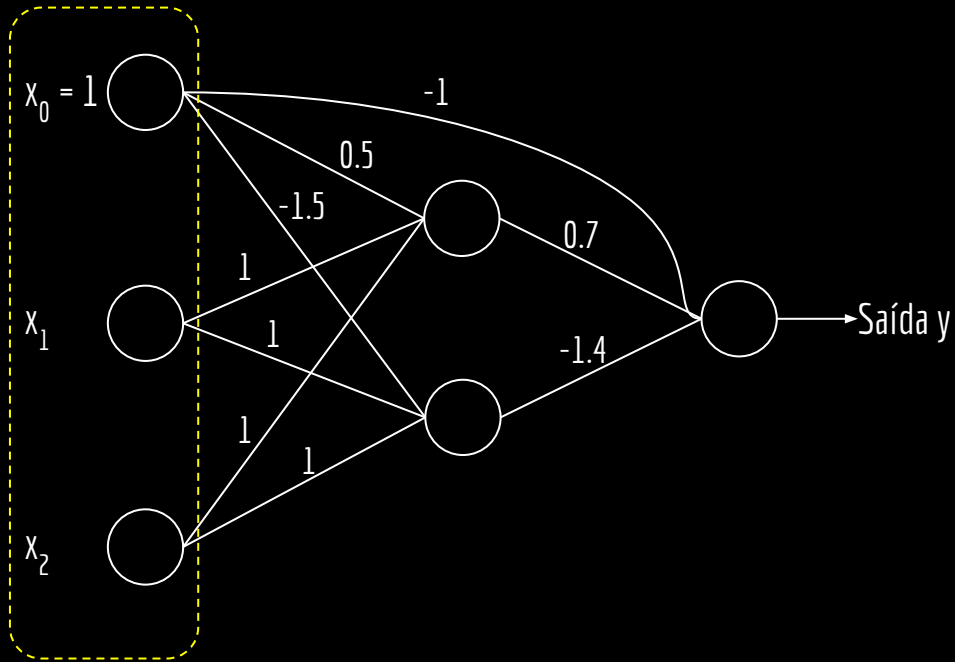
A função de ativação injeta uma **não linearidade**, também chamada de **função de transferência**.

**Podemos combinar múltiplas não linearidades para gerar um espaço onde os dados são separáveis.**

# Exemplo

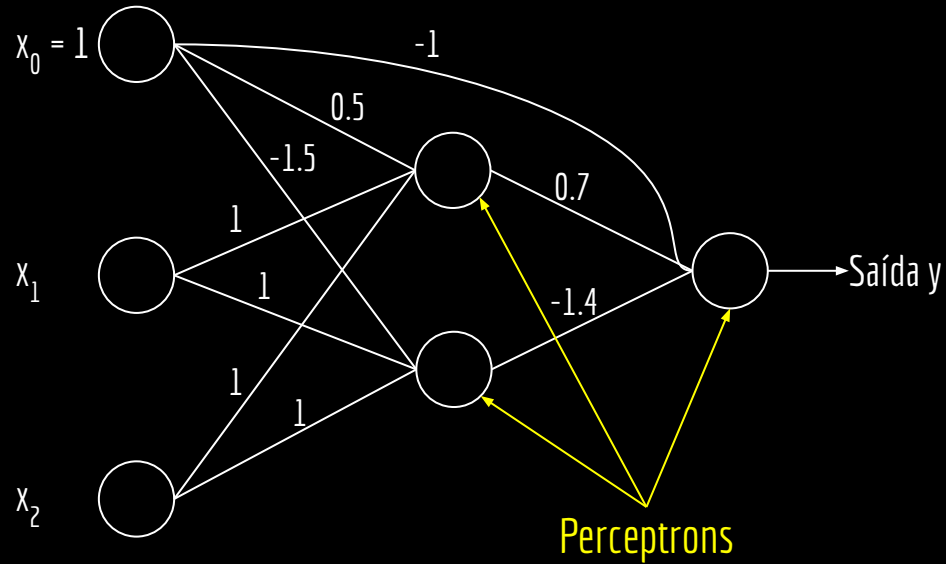


# Terminologia



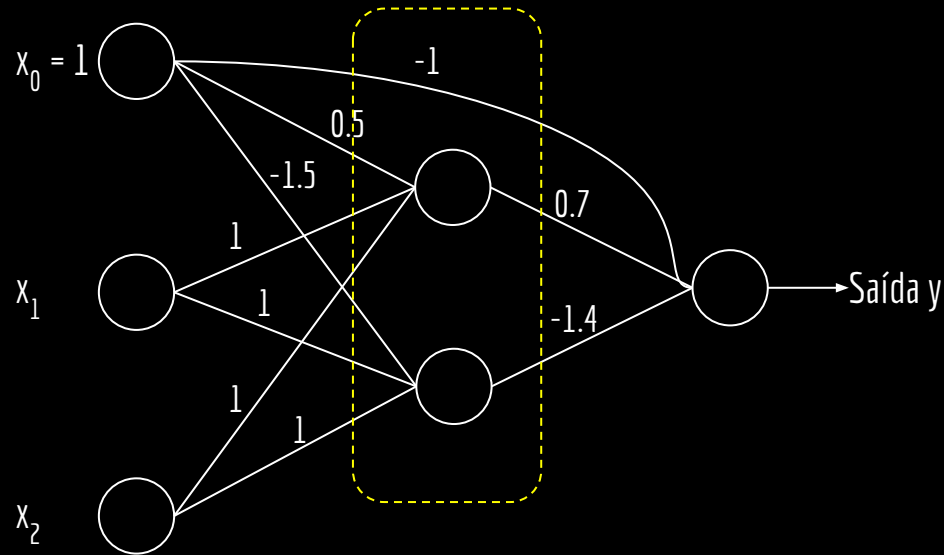
Entrada

# Terminologia



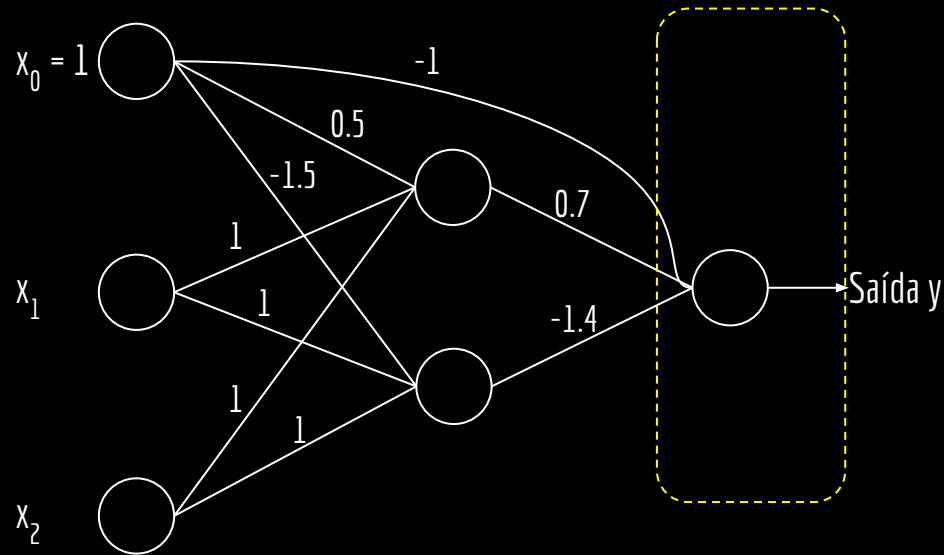


# Terminologia



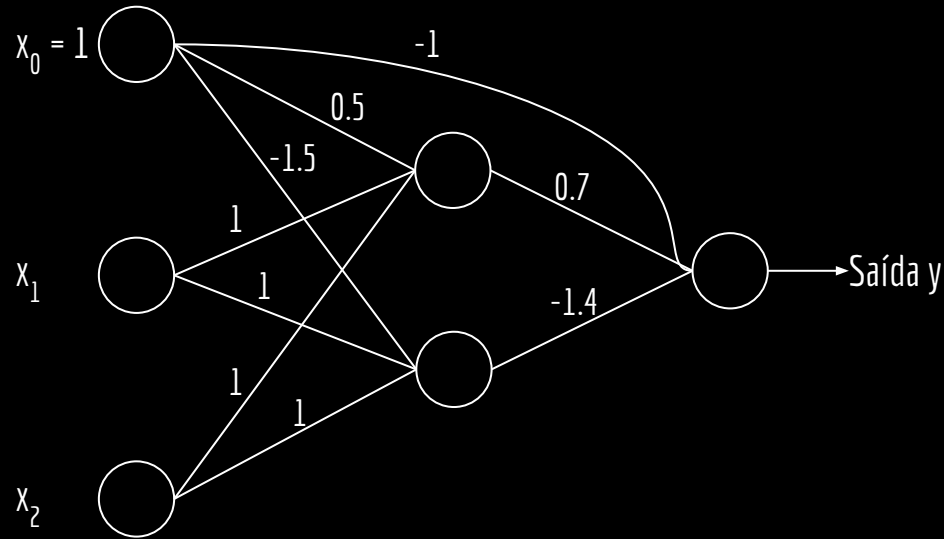
Perceptrons (neurônios) conectados à entrada formam o primeiro hidden layer (camada oculta). Uma rede pode ter múltiplas camadas ocultas.

# Terminologia



A última camada é formada de neurônios que geram a resposta.  
Geralmente chamada de output layer (camada de saída).

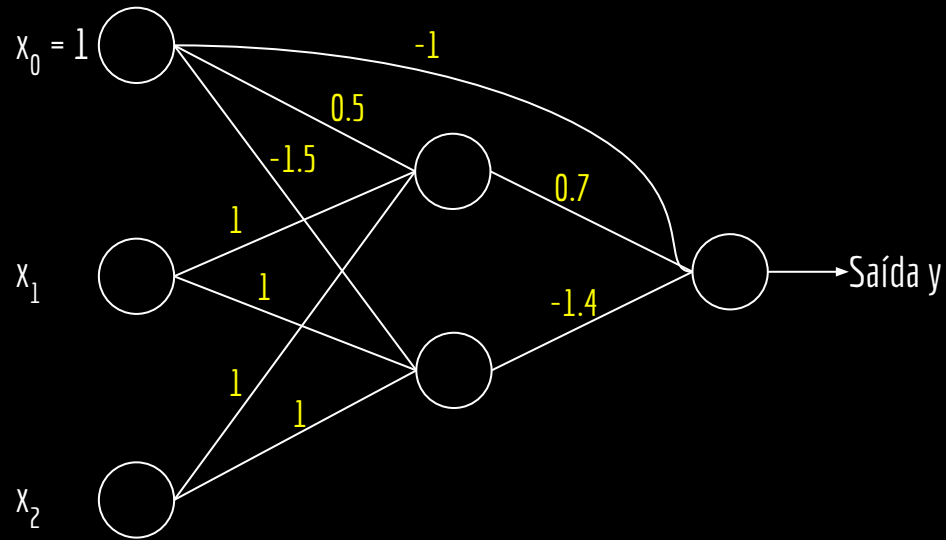
# MLP



Essa é uma **rede neural de duas camadas**.

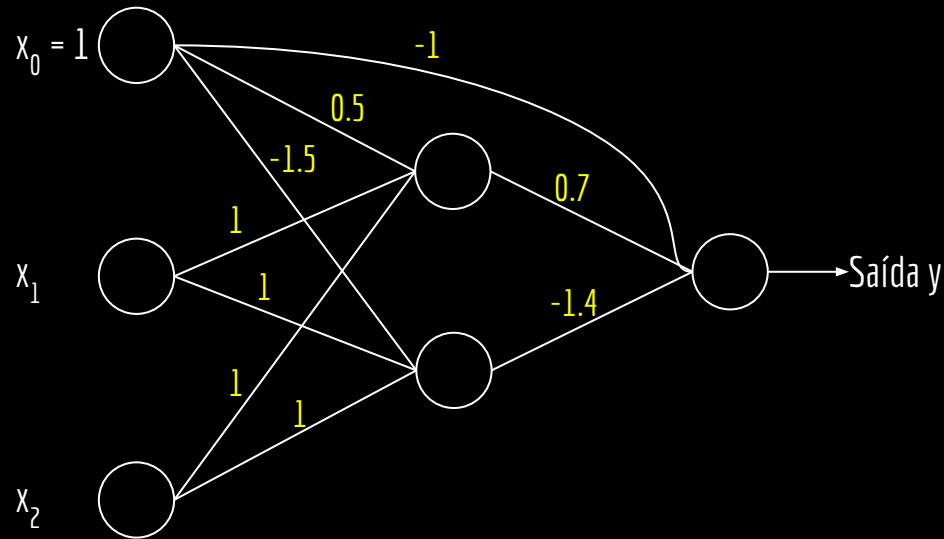
Por se tratar da conexão de múltiplos perceptrons (neurônios) em camadas, também pode ser chamada de **Multilayer perceptron**, ou **MLP**.

# Terminologia



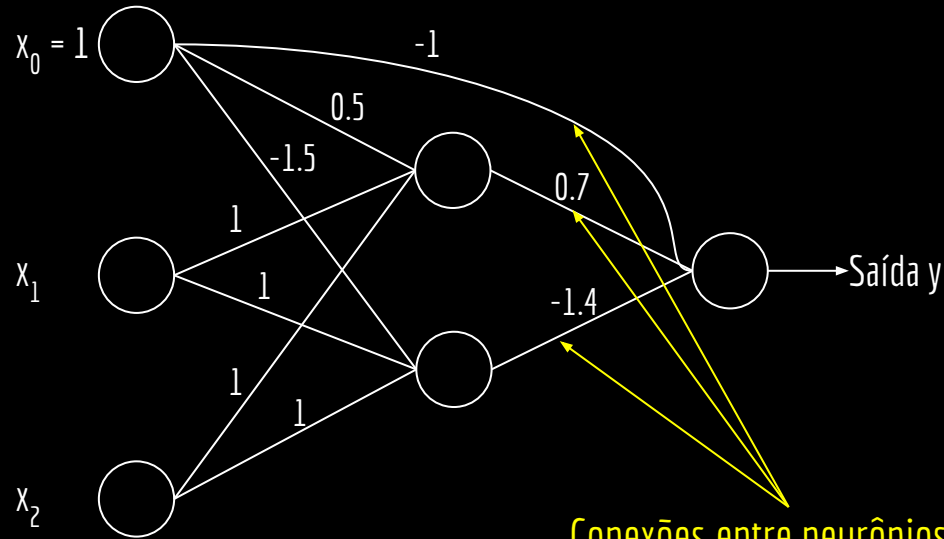
Os pesos são os parâmetros da rede. Temos uma rede de 9 parâmetros.

# Terminologia



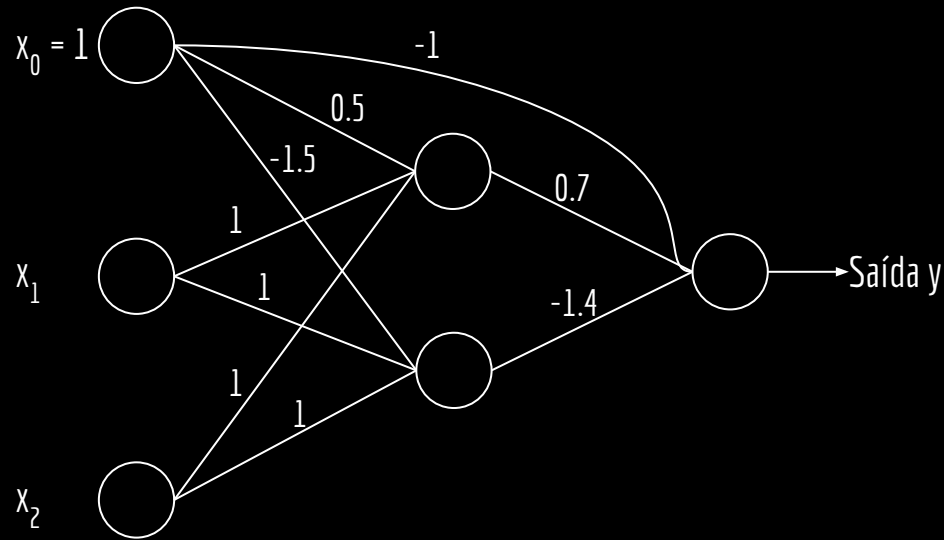
Os pesos são os parâmetros da rede. Temos uma rede de 9 parâmetros.  
A rede GPT-3, por exemplo, possui cerca de 175 bilhões de parâmetros!!!

# Terminologia



Conexões entre neurônios são chamadas de sinapses.

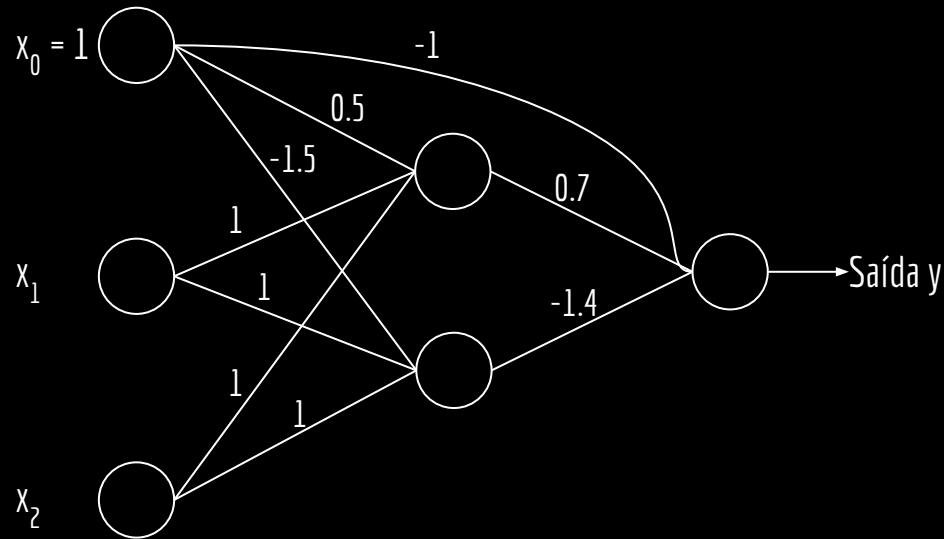
# Terminologia



Esse é um dos tipos mais comuns de rede, gerada por um grafo direcionado e acíclico.

Por não conter ciclos, essa rede é do tipo *Feed-forward*.

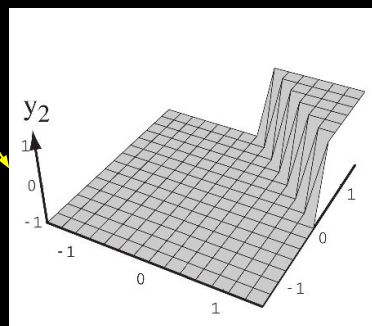
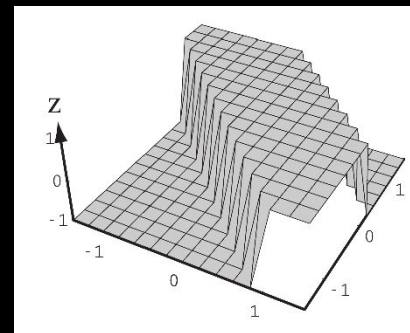
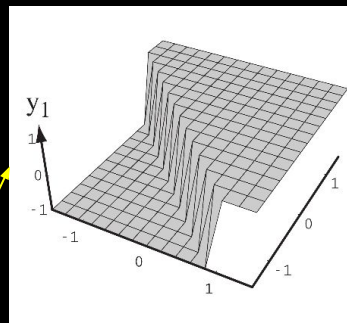
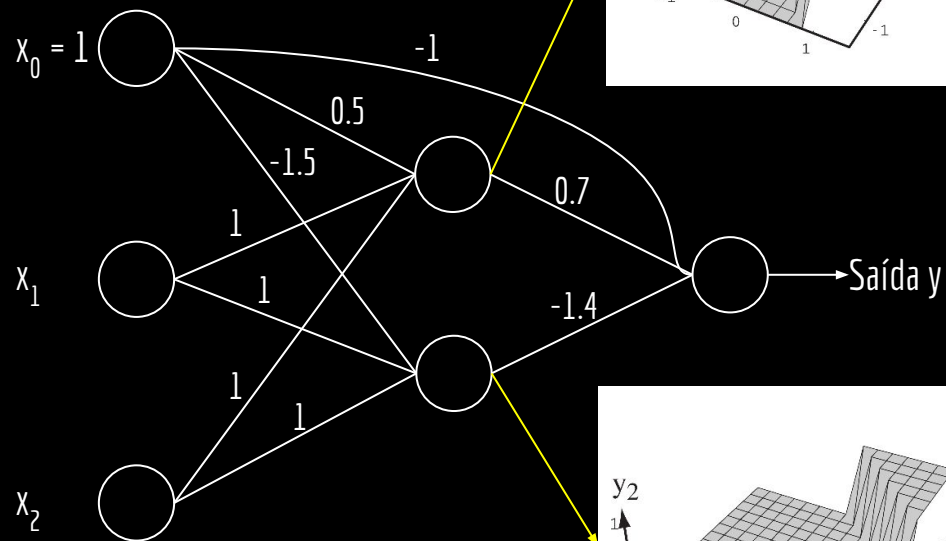
# Terminologia



O processo de entrar com um dado e gerar sua saída é chamado de *forward propagation*.

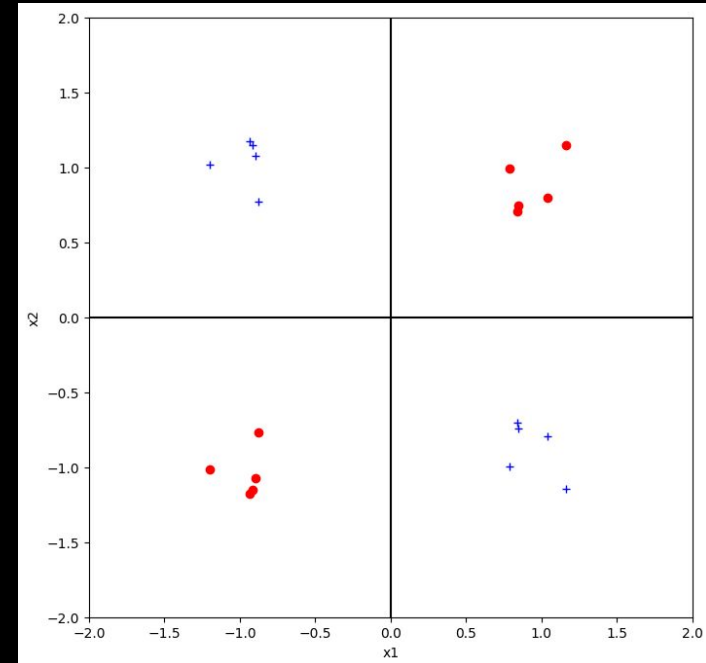
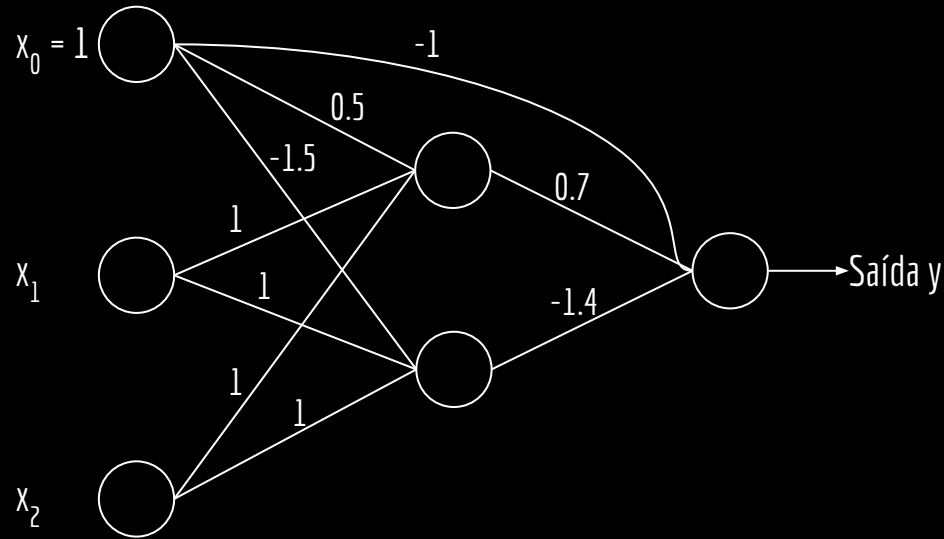


# Não linearidades



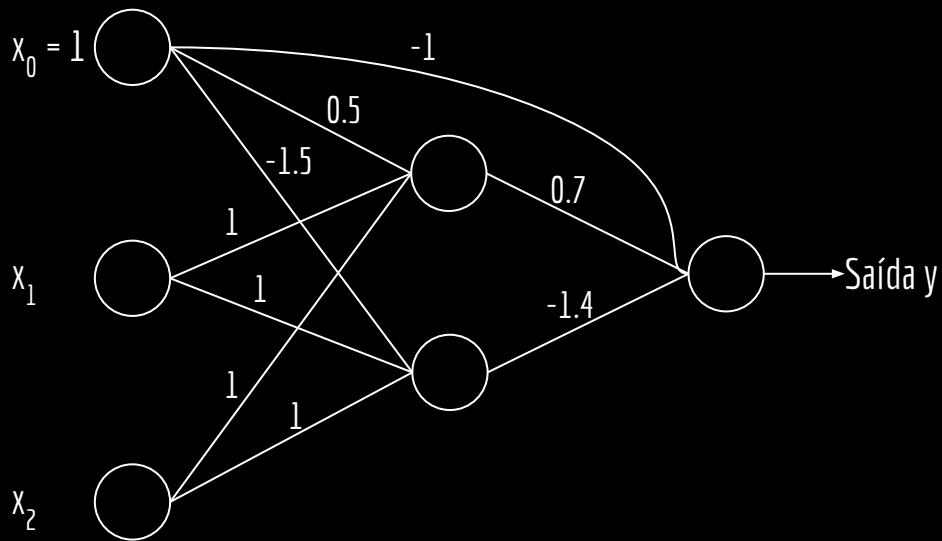
# XOR

Essa rede é capaz de resolver o problema XOR.



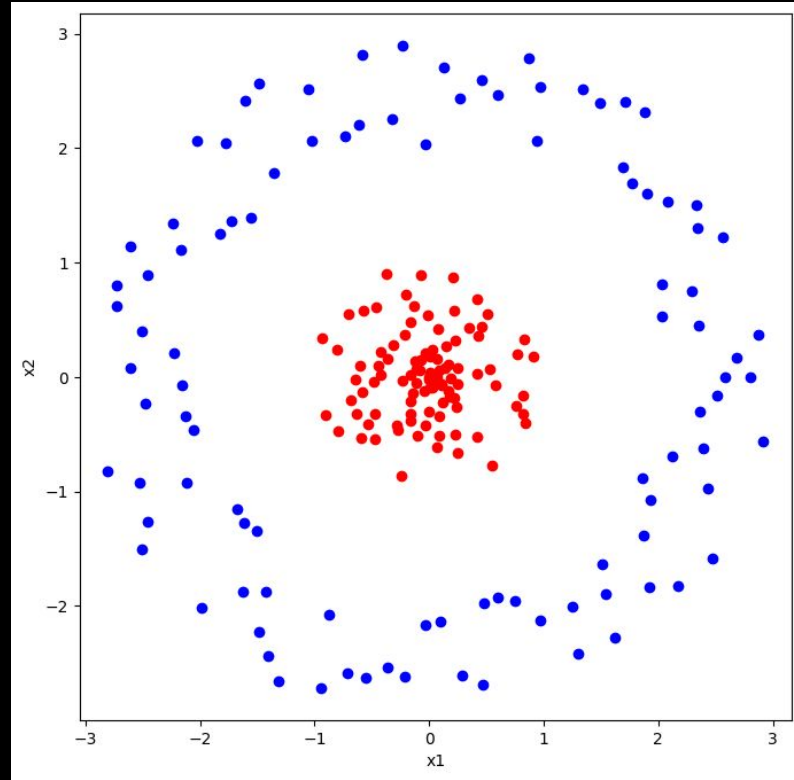
# Faça você mesmo #1

Use o exemplo disponibilizado no Google Colab para implementar a rede da imagem usando o scikit-learn e perceptrons simples.



# A teoria ...

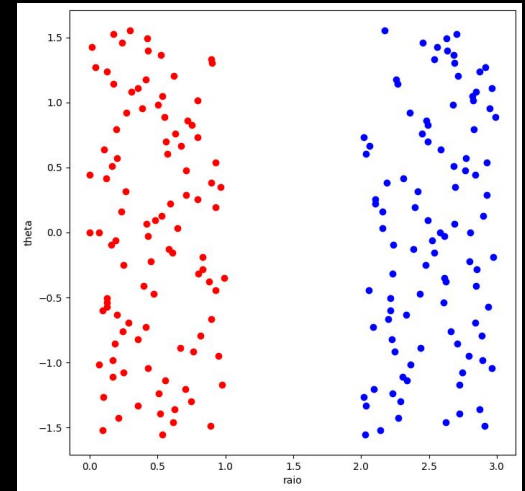
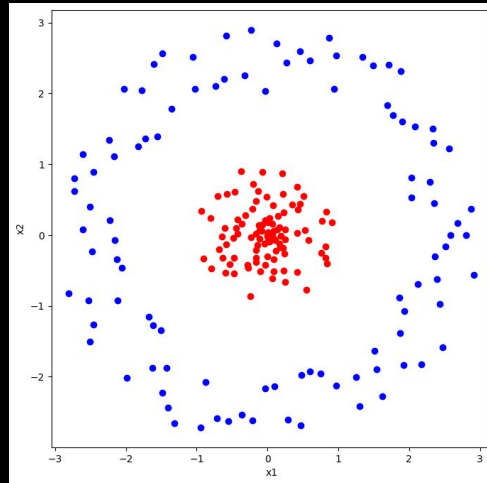
Se você precisasse criar uma fronteira linear para o problema a seguir, o que você faria?



# A teoria ...

Se você precisasse criar uma fronteira linear para o problema a seguir, o que você faria?

No exemplo, poderíamos transformar as coordenadas cartesianas em coordenadas polares ...



# Voltando ao Perceptron

A equação geral de um perceptron pode ser modificada de forma a permitir uma função de base não linear.

$$y(\mathbf{x}) = f(\mathbf{w}^T \phi(\mathbf{x}))$$

onde  $\phi(\cdot)$  é a função de transformação não linear.

# Voltando ao Perceptron

No exemplo, basta usar uma função de conversão de coordenadas polares como  $\phi(\cdot)$  para obter um problema linearmente separável, e aplicar diretamente um perceptron.

# O problema $y(\mathbf{x}) = f(\mathbf{w}^T \phi(\mathbf{x}))$

Temos uma infinidade de funções possíveis para  $\phi(\cdot)$ , e geralmente não é óbvio nem viável definir uma função  $\phi(\cdot)$  fixa.

O truque de uma rede neural é inserir neurônios intermediários que “aprendem” os pesos necessários na função não linear  $\phi(\cdot)$ , que transforma os dados e os passa para a próxima camada.

Podem ser necessárias múltiplas camadas de neurônios intermediários (hidden layers) para gerar as não-linearidades necessárias.



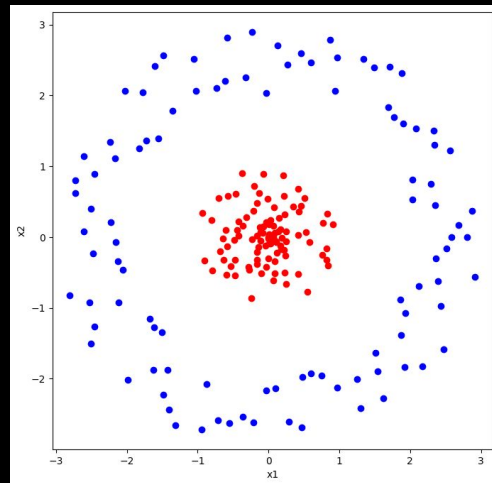
# Faça você mesmo #2

Modifique o código disponibilizado no Google Colab para transformar os dados disponibilizados no gráfico em coordenadas polares.

As equações são:

$$r = \sqrt{(x^2 + y^2)}$$

$$\theta = \arctan \frac{y}{x}$$



# SVMs e Mark 1



O Mark I Perceptron de Rosenblatt possuía pesos ajustáveis para realizar transformações não lineares nos dados de entrada.

Um Máquina de Vetores de Suporte (SVM) é um tipo de classificador que usa modificações não lineares para aumentar a dimensionalidade dos dados. Nessas dimensões, é possível separar os dados linearmente.

O classificador usa uma função de kernel.

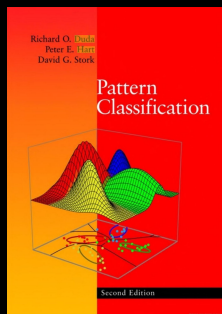
Técnica chamada de *kernel trick*.

# Exercícios

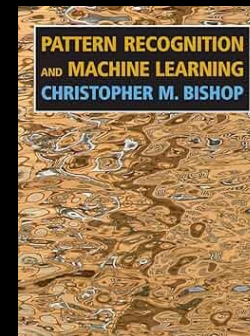
1. Treine um perceptron para classificar os dados convertidos em coordenadas polares dos slides anteriores.

# Referências

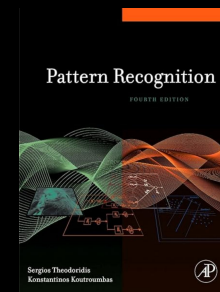
Duda, R. O., Hart, P. E., Stork, D. G. Pattern Classification. 2012.



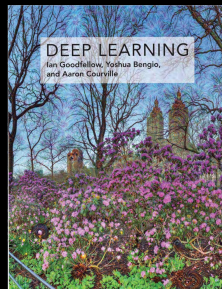
Bishop, C. M. Pattern Recognition and Machine Learning. 2006.



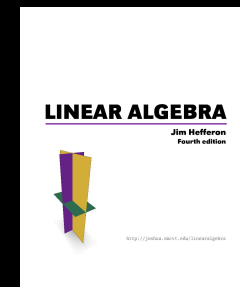
Theodoridis, S., Koutroumbas, K. Pattern Recognition & Matlab Intro. 2010.



Goodfellow, I., Bengio, Y., Courville, A. Deep Learning. 2016.



Hefferon, J. Linear Algebra. 2015.



# Licença

Esta obra está licenciada com uma Licença [Creative Commons Atribuição 4.0 Internacional](https://creativecommons.org/licenses/by/4.0/).