



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE**  
**PROJETO DE INICIAÇÃO CIENTÍFICA**  
**UNIDADE ACADÊMICA DE ENGENHARIA ELÉTRICA**

# Classificação de distúrbios pulmonares em radiografias de tórax usando Redes Convolucionais

Reunião 02/10/2020

# Introdução às Redes Neurais Artificiais

- Inspiração na arquitetura do cérebro para construir um sistema inteligente.
- Cerne do Aprendizado Profundo.



**Classificação de  
Bilhões de Imagens**



**Reconhecimento  
de Fala**



**Sistemas de  
Recomendação**



**Jogos Digitais**

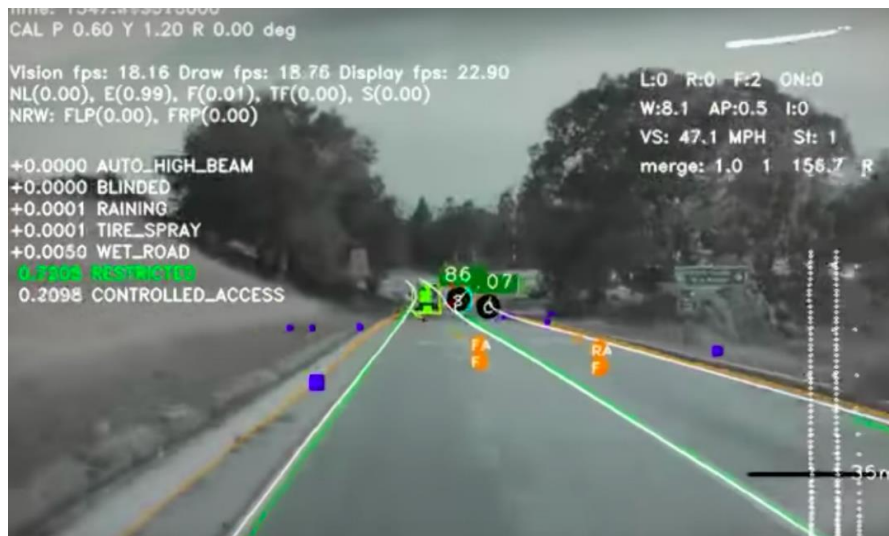
- Lidar de forma simples com atividades complexas.

# De Neurônios Biológicos a Neurônios Artificiais

- **1943:** primeira arquitetura de rede neural feito pelo neurofisiologista Warren McCulloch e o matemático Walter Pitts.
- **1960:** “Idade Média das RNA” – as crenças foram maiores do que os resultados.
- **1980:** “Renascimento das RNA” - novas arquiteturas de rede e melhores técnicas de treinamento.
- **1990:** A promessa das Máquinas de Vetores de Suporte – por muito tempo ofereceu resultados melhores que as RNA.

# Nova onda para as Redes Neurais Artificiais

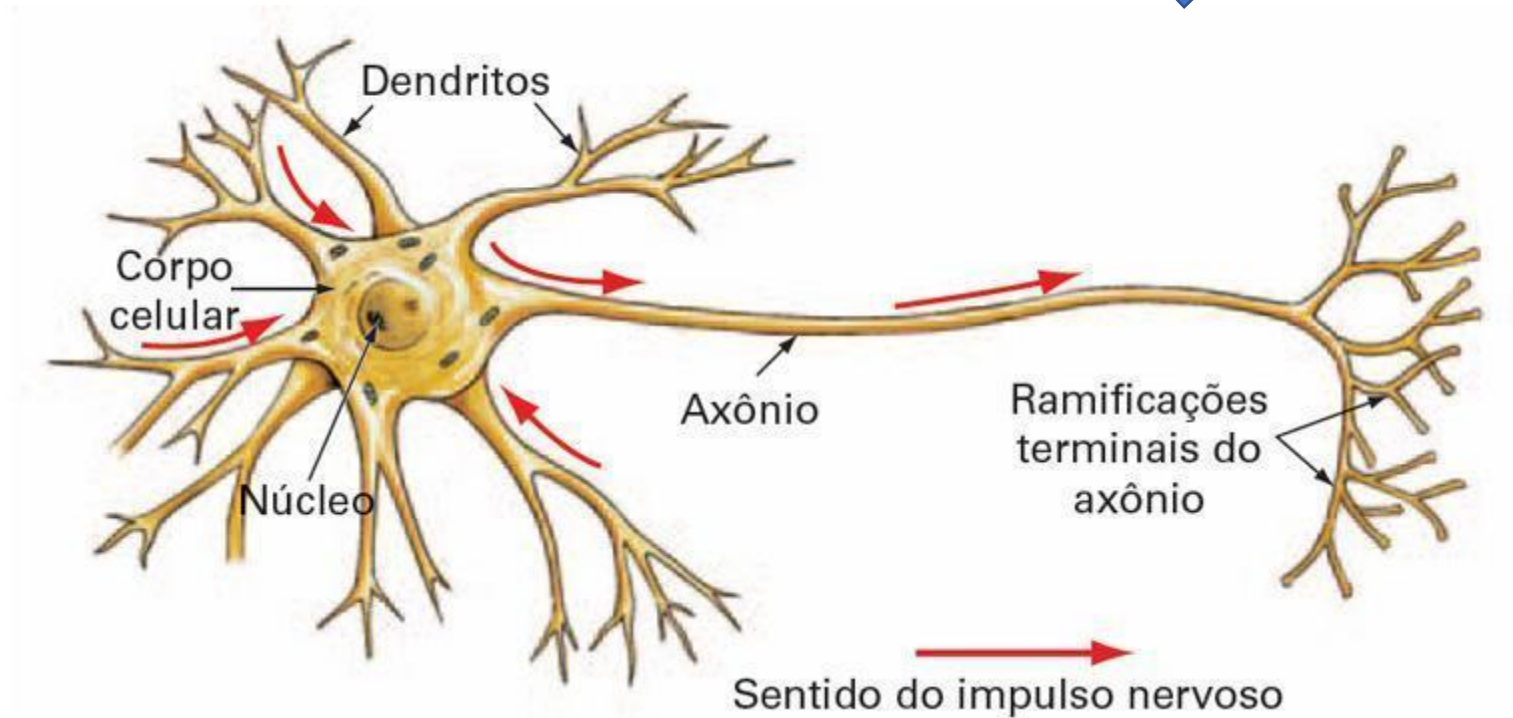
- Grande quantidade de dados disponíveis.
- Produção de máquinas mais potentes.
- Algoritmos de treinamento melhorados.
- Circulo virtuoso de financiamento e progresso.



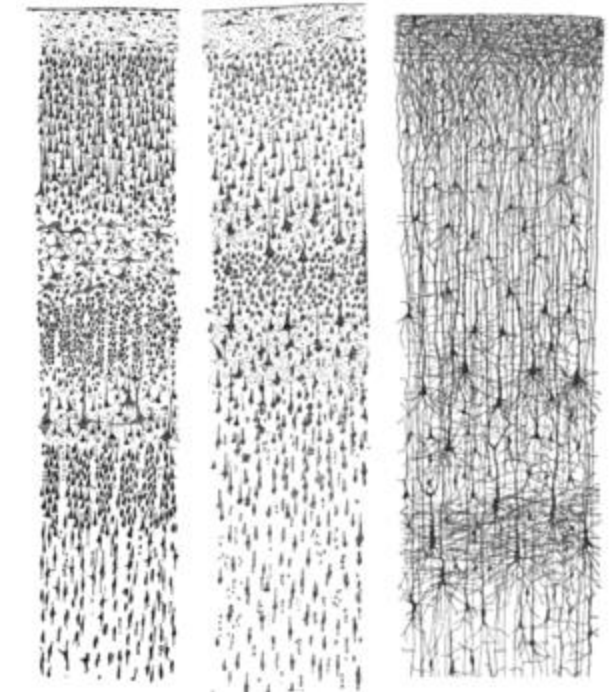
**Piloto Automático  
de um Tesla**

# Neurônios Biológicos

Comunicação Através de  
Sinais Sinápticos



Neurônio Biológico

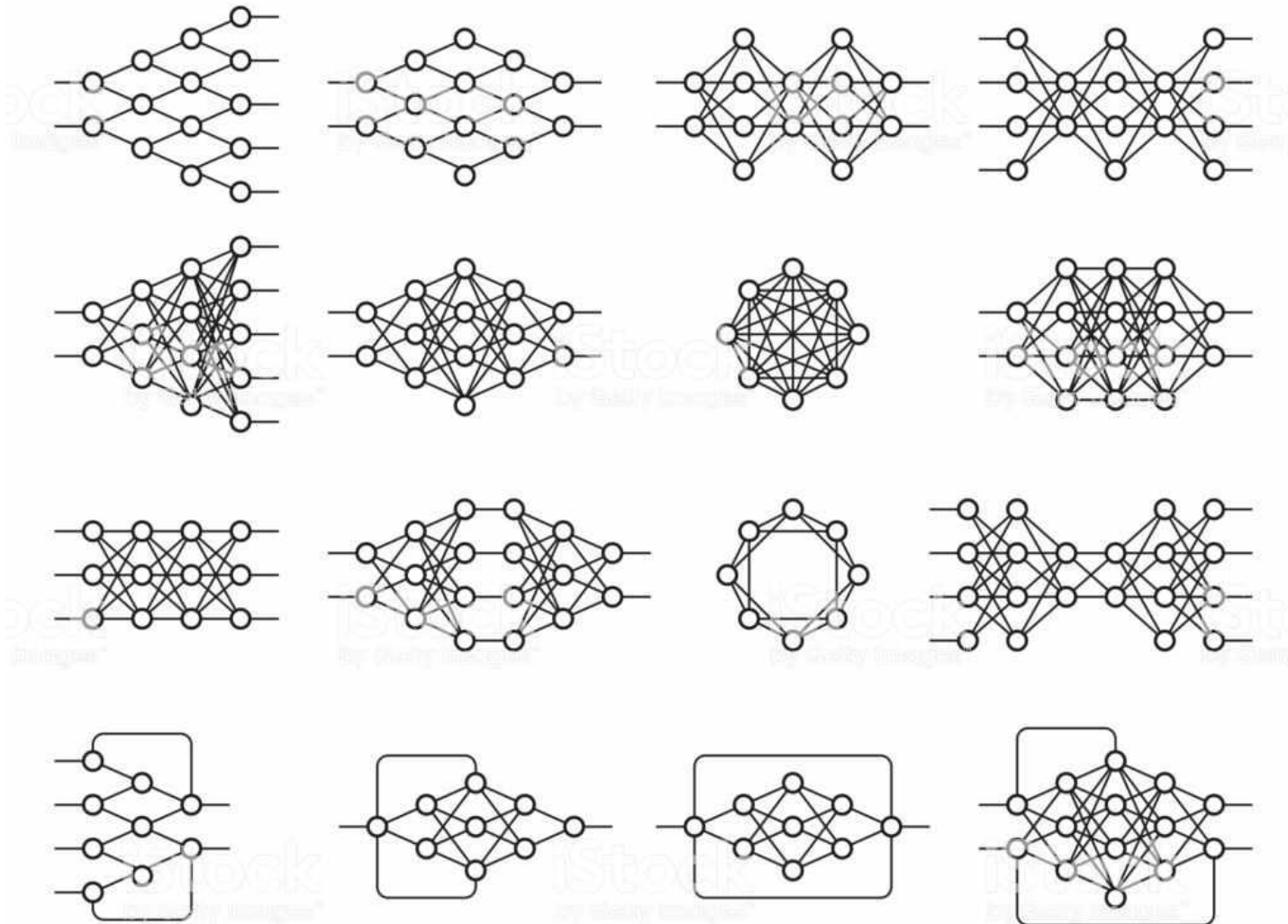


Laminação  
Cortical

Mapeamentos do Cérebro mostram que os neurônios se organizam em camadas consecutivas.



# Redes Neurais Artificiais



Você pode ter o melhor dos dois mundos encarando abertamente inspirações biológicas sem ter medo de criar modelos biologicamente não realistas, desde que funcionem bem.

Os neurônios biológicos parecem implementar uma função de ativação aproximadamente sigmoide (em forma de S), de modo que os pesquisadores a mantiveram por muito tempo. Mas acontece que a função ReLU geralmente funciona melhor nas RNA. Este é um dos casos em que a analogia biológica foi enganosa.

# Cálculos Lógicos com Neurônios

- McCulloch e Pitts propuseram um modelo simples de neurônio biológico com uma ou mais entradas binárias e uma saída binária.

NOT

<i>in</i>	<i>out</i>
0	1
1	0

AND

<i>in</i> <sub>1</sub>	<i>in</i> <sub>2</sub>	<i>out</i>
0	0	0
0	1	0
1	0	0
1	1	1

OR

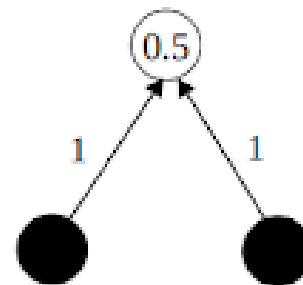
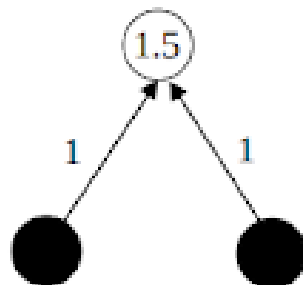
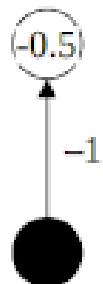
<i>in</i> <sub>1</sub>	<i>in</i> <sub>2</sub>	<i>out</i>
0	0	0
0	1	1
1	0	1
1	1	1



Exemplo simples para calcular qualquer proposta lógica desejada.

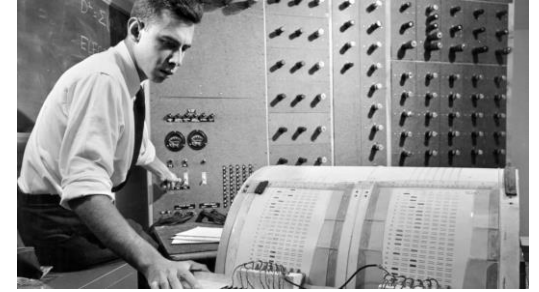


Você pode imaginar facilmente como essas redes podem ser combinadas para calcular expressões lógicas complexas.

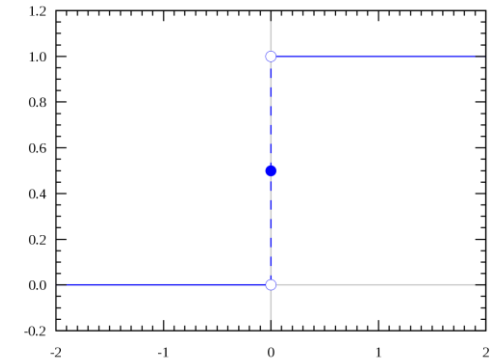
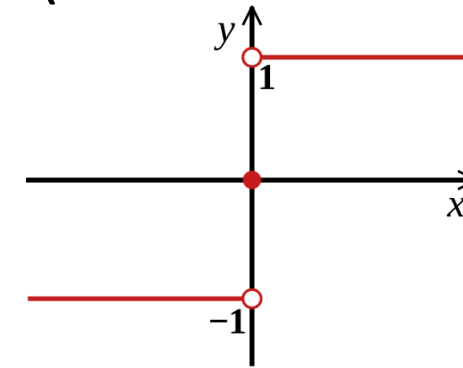
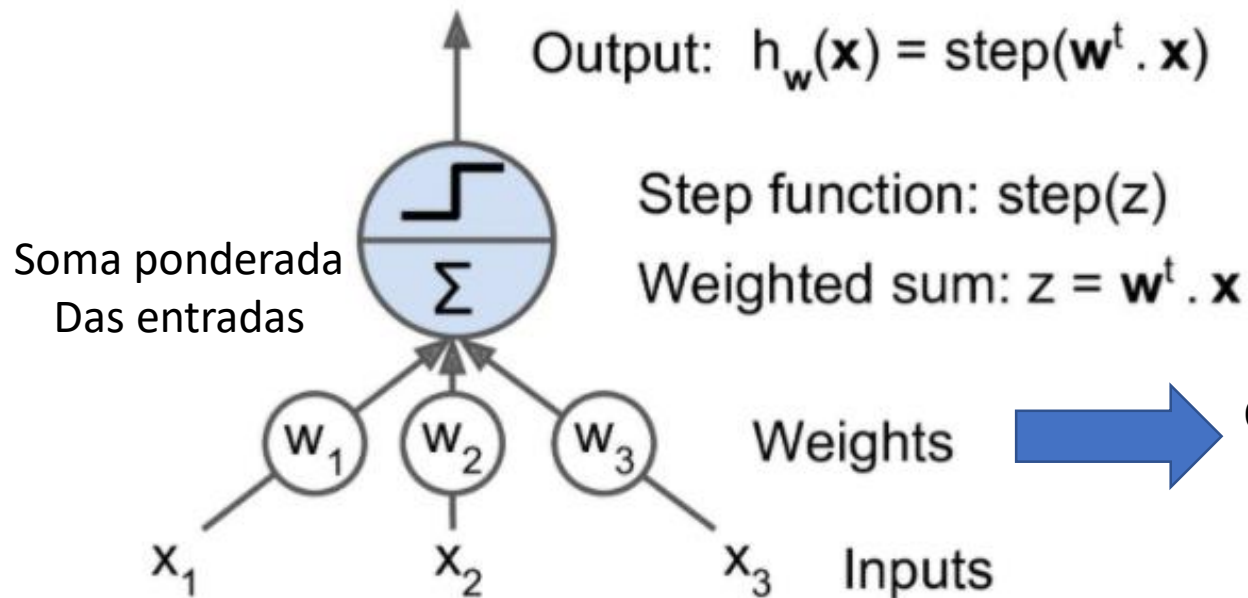




# Perceptron



- Inventada em 1957 por Frank Rosenblatt (uma das mais simples arquiteturas de RNA).



As funções de ativação basicamente decidem se um neurônio deve ser **ativado** ou não.

O treinamento da minha rede neural consistem em encontrar os pesos ideais (quanto mais generalista, melhor).

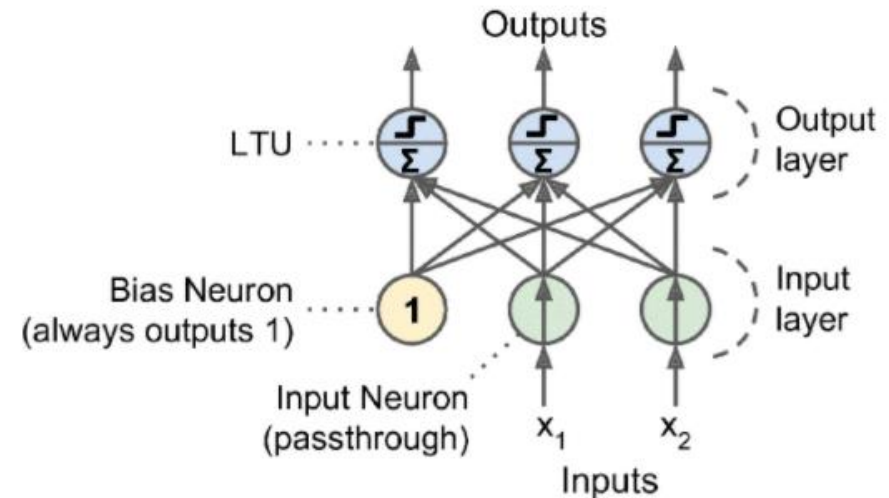
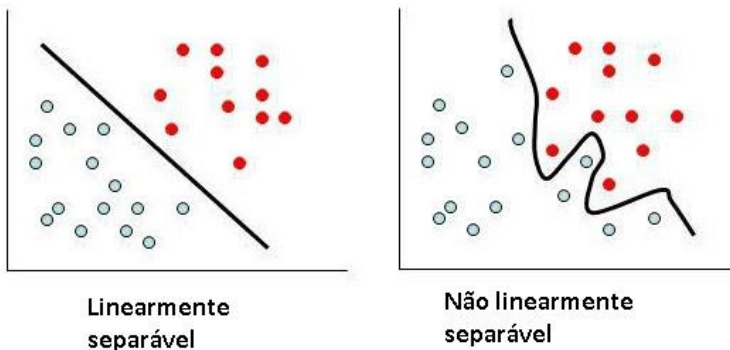


# Perceptron

Problemas linearmente separáveis  
oferecem uma infinidade de hiperplanos!

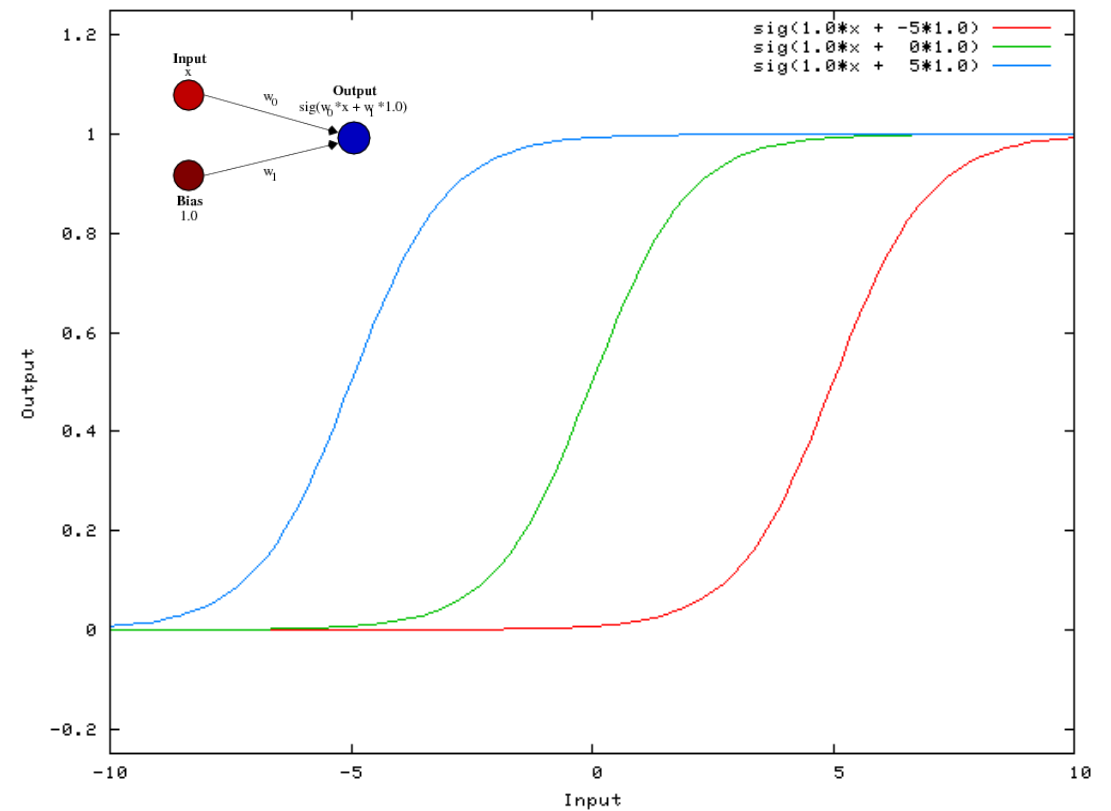
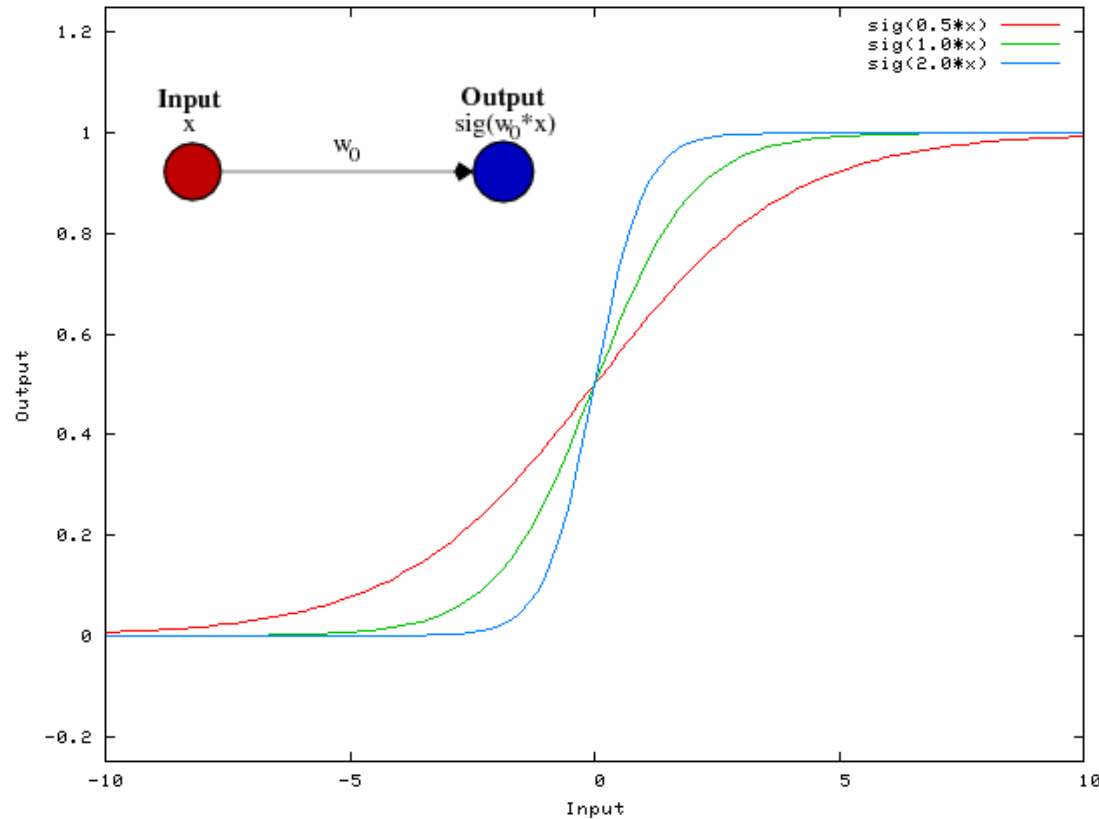


- Classificações Binárias simples (problemas lineares e linearmente separáveis).
- Uma perceptron é composta por uma camada de entrada e uma camada de saída.
- Adição de um neurônio de viés (bias) – permite mudar a função de ativação para a esquerda ou direita.
- Classificador multioutput.



# Neurônio de Viés

Ajuda na convergência do resultado desejado



# Atualização dos pesos

$$w_{i,j}^{(\text{next step})} = w_{i,j} + \eta(\hat{y}_j - y_j)x_i$$

- $w_{\{i,j\}}$  é o peso da conexão entre o i-ésimo neurônio de entrada e o j-ésimo neurônio de saída.
- $x_{\{i\}}$  é o valor da i-ésima entrada da instância de treinamento atual.
- $\hat{y}_{\{j\}}$  é a j-ésima saída do neurônio de saída para a instância de treinamento atual.
- $y_{\{j\}}$  é a j-ésima saída alvo do neurônio de saída para a instância de treinamento atual.
- $\eta$  é a taxa de aprendizado (velocidade da descida estocástica do gradiente).

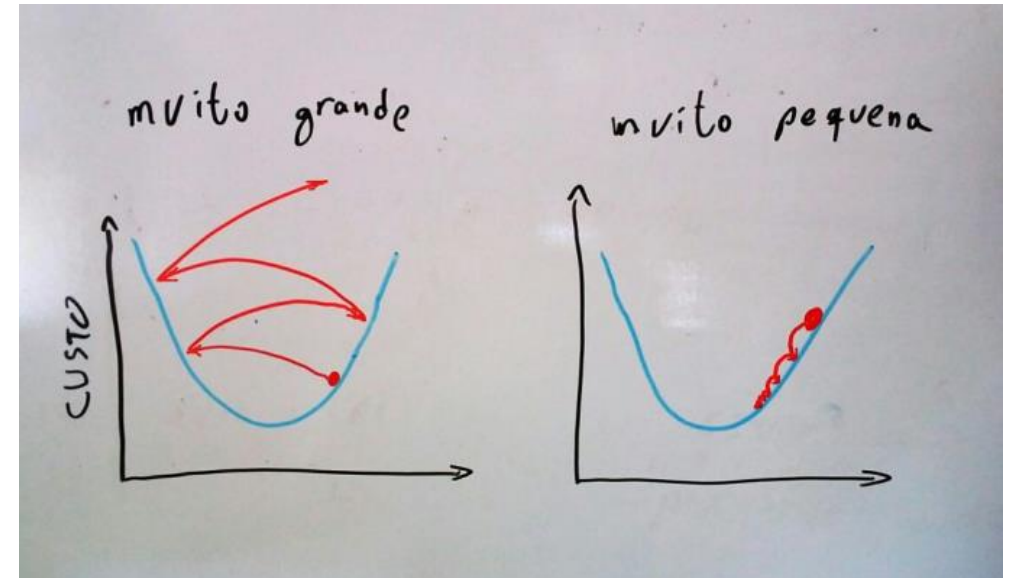
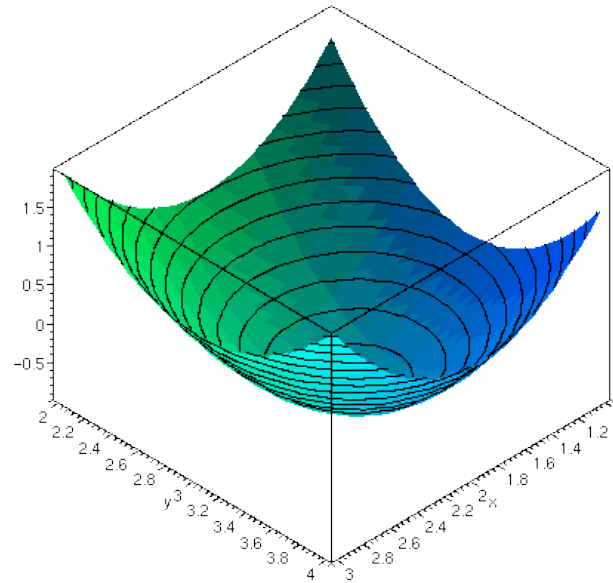
# Gradiente Descendente

- O método do gradiente é um método numérico usado em otimização. Para encontrar um mínimo de uma função usa-se um esquema iterativo, onde em cada passo se toma a direção do gradiente, que corresponde à direção de declive máximo.

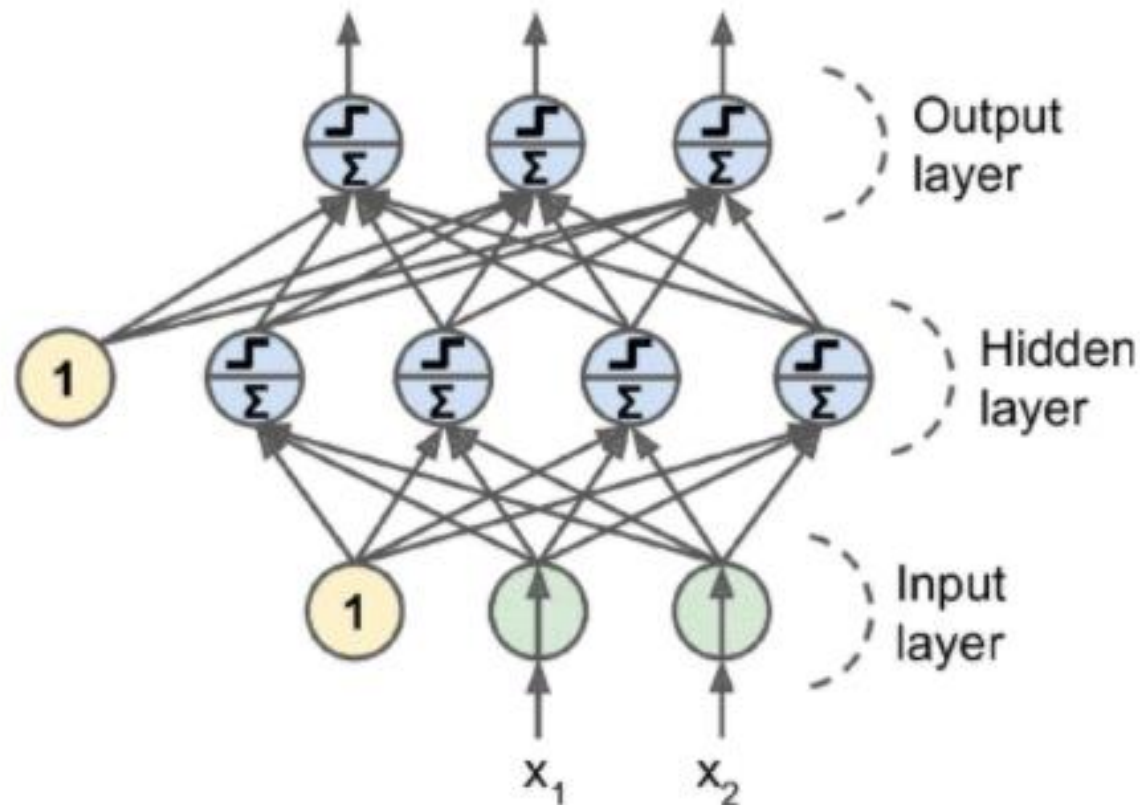
A função de custo se baseia em: o que saiu da rede é parecido com o que eu esperava?

**Ex: erro quadrático médio**

Minimizar o custo alterando os pesos da rede (**Retropropagação**).



# Perceptron Multicamada e Retropropagação

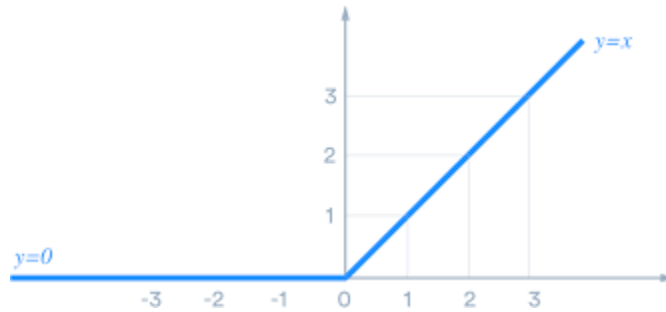


Quando uma rede neural possui duas ou mais camadas, é chamada de rede neural profunda.

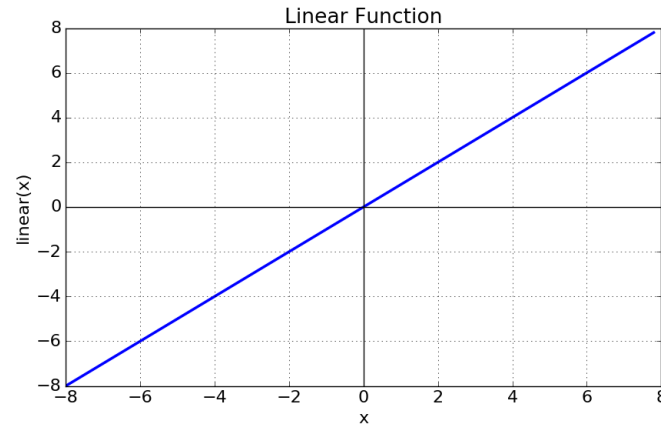
Para cada instância de treinamento, o algoritmo de retropropagação primeiro faz uma previsão, mede o erro, então passa por cada camada no reverso para medir a contribuição do erro em cada conexão e, finalmente, ajusta os pesos para reduzir o erro.

Para cada neurônio que produziu uma previsão incorreta, ele reforçará os pesos da conexão das entradas que contribuíram para a previsão correta.

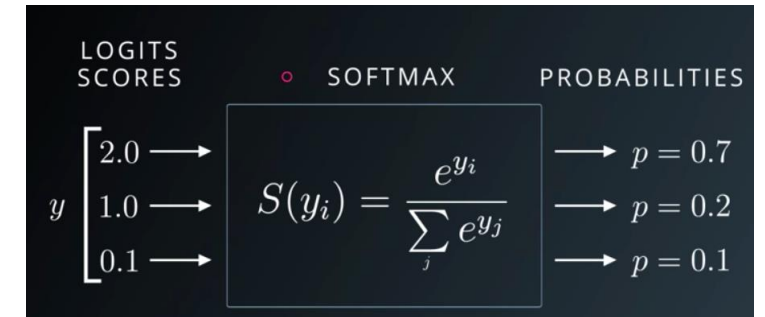
# Funções de Ativação



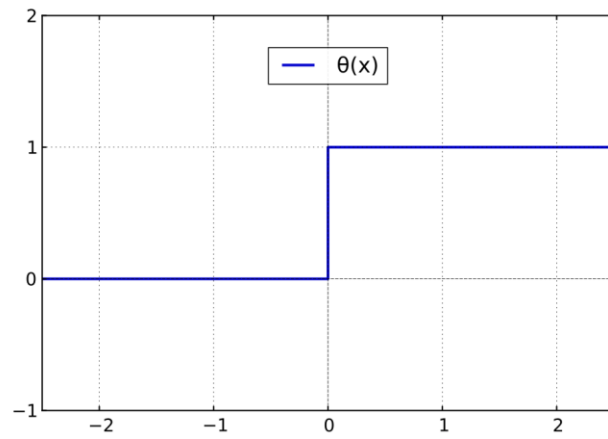
ReLU



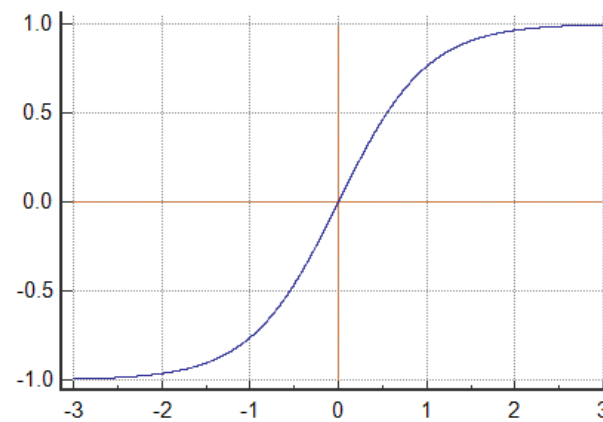
Função Linear



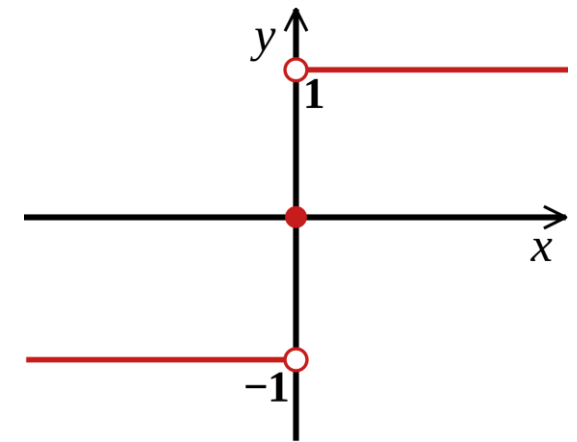
Softmax



Função Degrau



Função Tangente



Função de Sinal

# Próximos passos...

- **Gradiente Descendente + Funções de Ativação**
- **Hiperparâmetros**
- Curso relâmpago de Python e seus pacotes
- **Treinamento TensorFlow**
- Redes Neurais Convolucionais
- Aplicação em Python de algumas redes neurais convolucionais



# Referências

- Mãos a obra: Aprendizagem de Máquina com Scikit-Learn & TensorFlow por Aurélien Géron (AltaBooks). Copyright 2019 de Aurélien Géron, 978-85-508-0381-4.
- NG, Andrew. Neural networks and deep learning. Disponível em: <https://www.coursera.org/learn/neural-networks-deeplearning?specialization=deep-learning>. Acesso em 24 set. 2020.