Data Science e Machine Learning na Prática: Introdução e Aplicações na Indústria de Processos

Apresentação 3 - Noções de Redes Neurais Artificiais

Afrânio Melo

afraeq@gmail.com
afrjr.weebly.com

Escola Piloto Prof. Giulio Massarani
PEQ-COPPE-UFRJ

2019

COPPE UFRJ

Redes neurais

Do que se tratam?

- Redes neurais artificiais são uma classe de técnicas de aprendizado inspiradas vagamente no funcionamento do cérebro humano.
- Quando usadas em conjunto com *hardwares* de grande poder de processamento, as redes neurais constituem as técnicas de aprendizado mais poderosas da atualidade.

Redes neurais

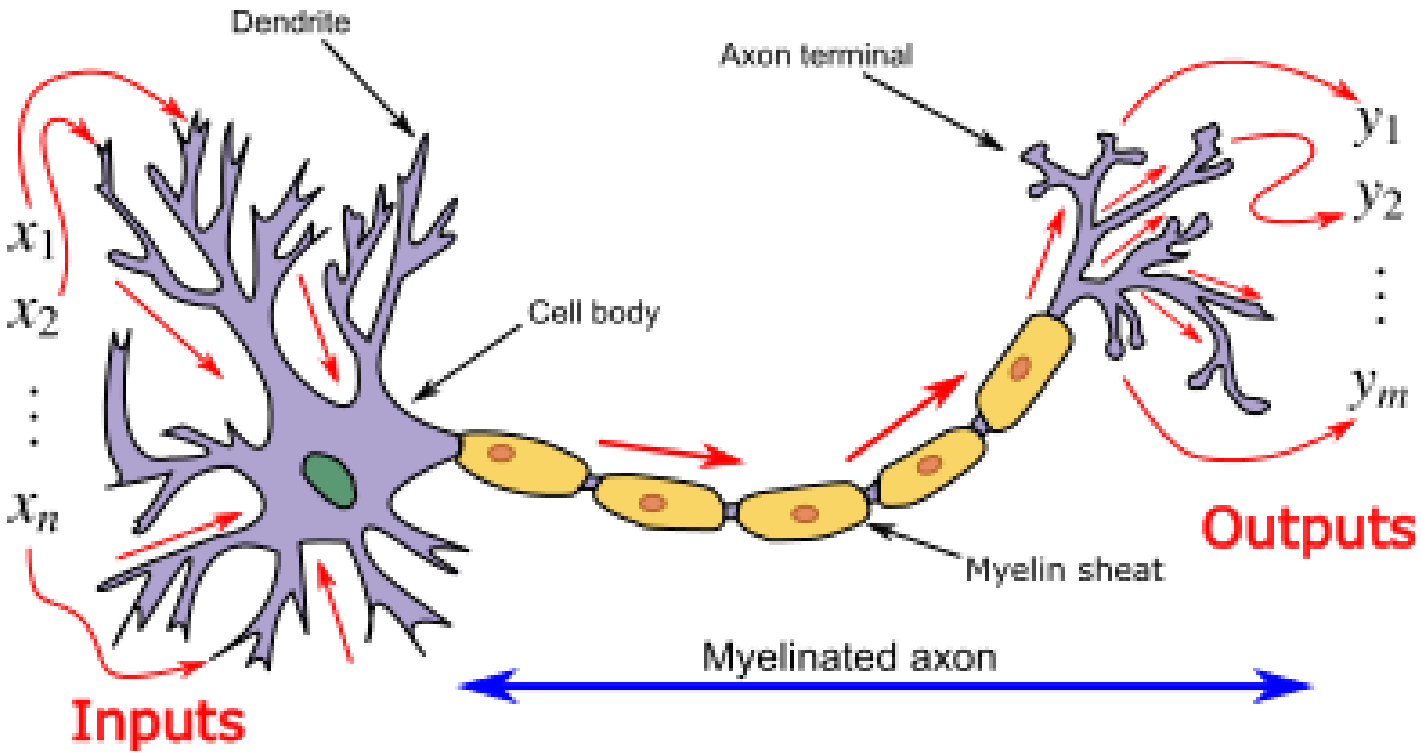


Figura 1: Modelo de um neurônio biológico.

Redes neurais

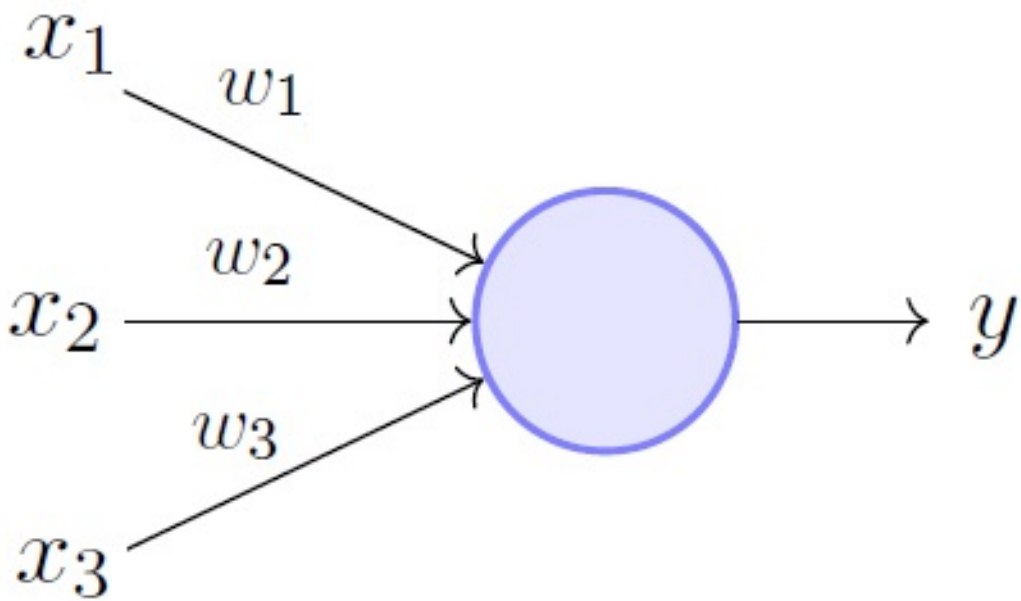


Figura 2: Modelo de um neurônio artificial.

Redes neurais

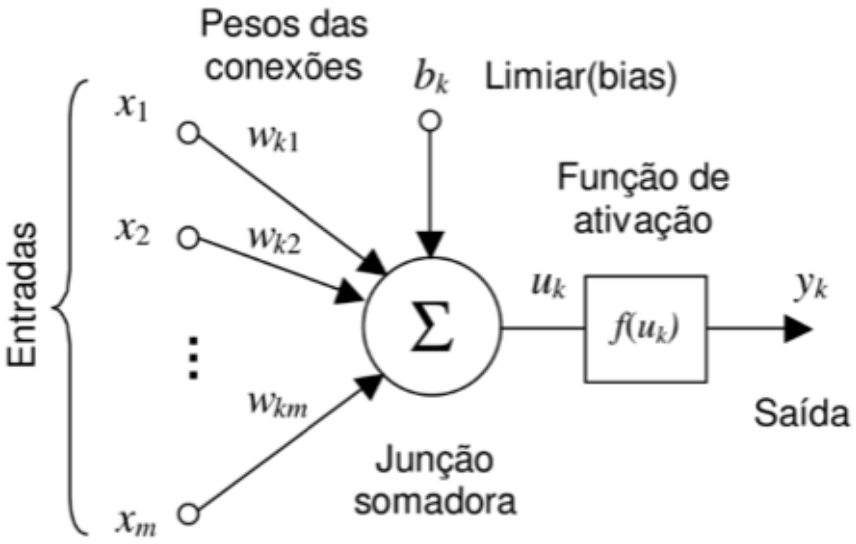


Figura 3: Esquema de um neurônio.

Funcionamento de um neurônio na rede

- Cada neurônio k na rede é uma unidade de processamento local, que aceita m conexões por meio das entradas x_1, x_2, ..., x_m e fornece uma saída y_k.
- Cada conexão i tem um peso associado, denotado por w_{ki}.
- A saída y_k do neurônio é expressa por:

$$y_k = f(u_k) = f\left(\sum_{j=1}^{m} w_{kj}x_k + b_k\right)$$

Redes neurais

Do que se tratam?

- A *rede neural* é o resultado da conexão entre vários neurônios.
- Ela pode ser caracterizada quanto a vários aspectos, destacando-se:
  - **arquitetura da rede**: a forma como os vários neurônios se conectam. Dividem-se em dois grandes grupos: *feedforward* (a informação se move apenas em um sentido, da entrada para a saída da rede) e *recurrent* (a informação se propaga nos dois sentidos, permitindo que a rede tenha memória);
  - **método de treinamento**: algoritmo usado para determinar os pesos w das conexões; em outras palavras, o algoritmo usado para fazer a rede aprender;
  - **função de ativação**: função f que transforma a soma $\sum_{j=1}^{m} w_{kj} x_k + b_k$ na saída da rede y_k.
- Nos próximos slides, estudaremos o tipo mais simples de neurônio, o *percéptron*.

Percéptrons

Do que se tratam?

- O *percéptron* é um neurônio cuja função de ativação é a *função degrau binária*:

$$f(u_k) = \begin{cases} 0, & \text{se } u_k \leq 0 \\ 1, & \text{se } u_k > 0 \end{cases}$$

- É o tipo mais simples de neurônio capaz de tomar decisões (p. ex., classificação) com base em pesos w aprendidos na etapa de treinamento.

Redes neurais

Redes de percéptrons

- É possível tornar cada vez mais sutil a tomada de decisão realizada pelos percéptrons por meio da associação deles em rede, como na figura abaixo.
- Na verdade, pode-se mostrar que em teoria é possível calcular *qualquer função lógica* com redes de percéptrons, por mais complicadas que sejam!

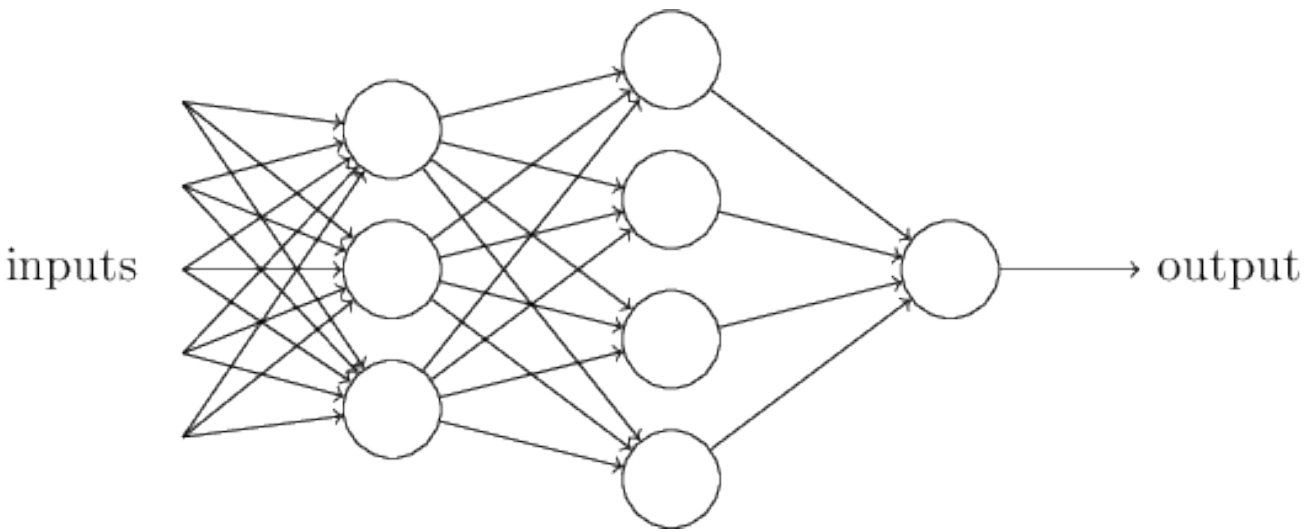


Figura 4: Rede de percéptrons com camadas ocultas. Cada uma das várias conexões acima tem um peso w associado, que é calculado no momento do treino para produzir a saída desejada.

Redes neurais

Redes de percéptrons

- Já que as redes de percéptrons, em teoria, são suficientes para o cálculo de qualquer função lógica, por que não as usamos na prática?
- Resposta: ausência de suavidade na relação entre os pesos w e a saída y_k, por conta da forma da função de ativação degrau binária!
- Treinar a rede significa ajustar os pesos w de todas as conexões de modo a produzir a saída y_k desejada (fornecida como *target* ao algoritmo de treino). Se não há uma relação suave entre w e y_k, fazer esse ajuste torna-se muito difícil!

Redes de percéptrons

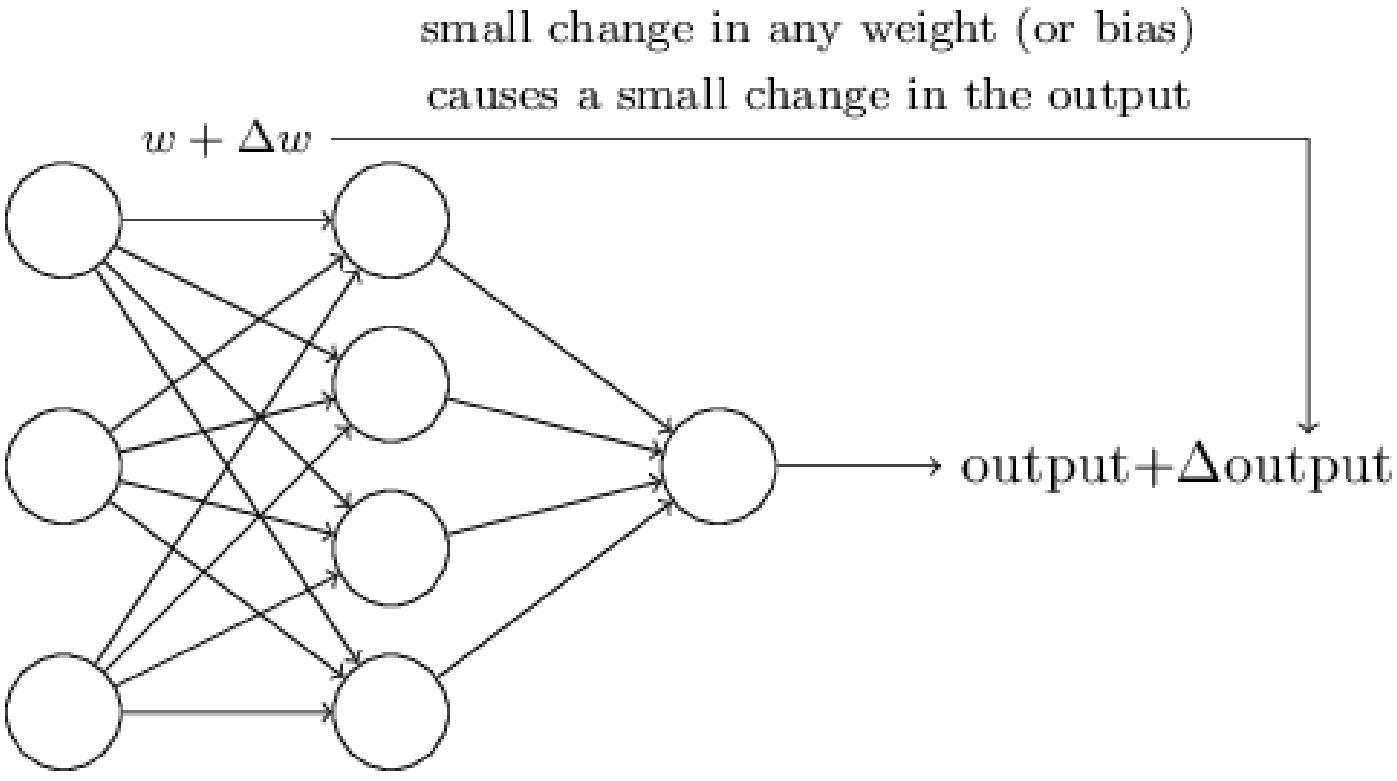


Figura 5: Comportamento desejado em uma rede neural. A função de ativação do percéptron não satisfaz esse requerimento.

Neurônios sigmóide

Introduzindo o neurônio sigmóide

- Essa é a motivação para introdução dos *neurônios sigmóide*, um tipo de neurônio cuja função de ativação $f(u_k)$ é a função sigmóide:

$$f(u_k) = \frac{1}{1 + \exp(-u_k)}$$

- Com esse tipo de neurônio, o treinamento do aprendizado de padrões não-lineares torna-se factível.

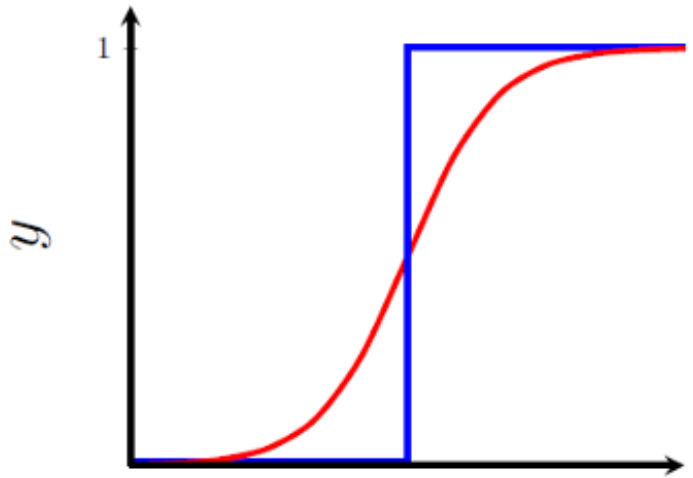
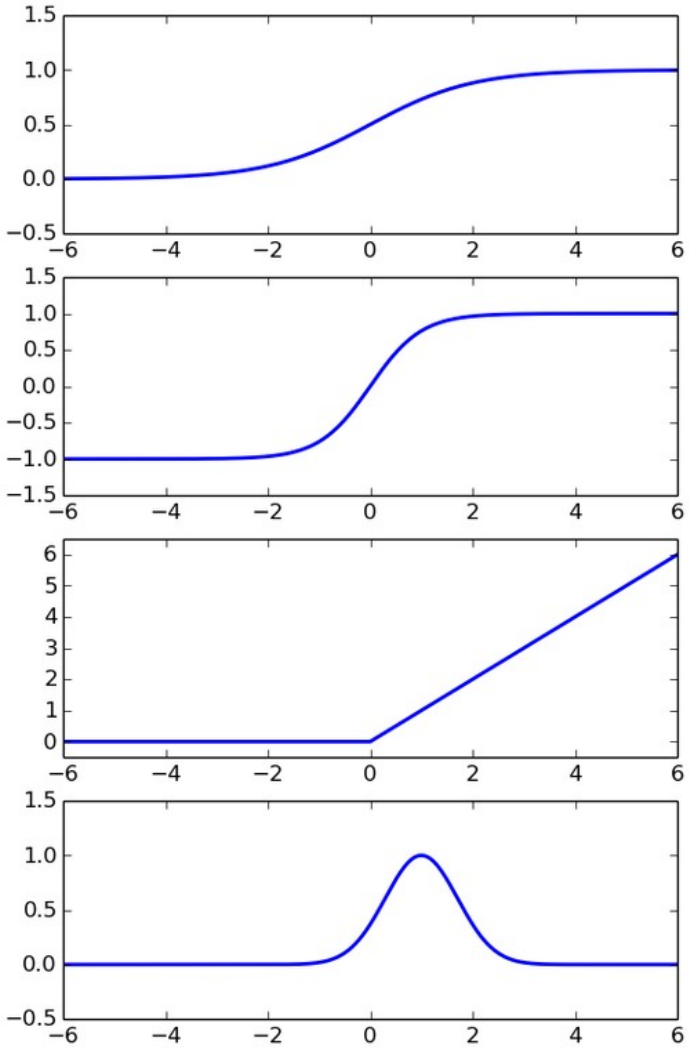


Figura 6: Relação entre y e w em neurônios utilizando a função de ativação degrau (azul) e a função sigmóide (vermelho).

Outras funções de ativação



Sigmoid

$$\phi(z) = \frac{1}{1 + e^{-z}}$$

Hyperbolic Tangent

$$\phi(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$

Rectified Linear

$$\phi(z) = \begin{cases} 0 & \text{if } z < 0 \\ z & \text{if } z \geq 0 \end{cases}$$

Radial Basis Function

$$\phi(z, c) = e^{-(\epsilon \|z - c\|)^2}$$

Figura 7: Funções de ativação comumente encontradas em redes neurais

Deep Learning

Redes neurais: ascensão e queda (~1980)

- O *boom* do interesse em redes neurais surgiu com a proposta de um algoritmo de retropropagação para o treinamento de redes *feedforward* em 1986 por Rumelhart, Hinton e Williams.
- Logo depois dessa animação inicial, as coisas esfriaram e pararam de avançar.
- Problemas mais complexos surgiam e as redes não eram capazes de resolver.
- A intuição dizia que, para aumentar o poder de processamento da rede, necessitava-se de múltiplas camadas ocultas!
- Só que, na época, parecia simplesmente impossível treinar esse tipo de rede...

Deep Learning

Redes neurais: ascensão novamente (~2000)

- Até que, em 2006, veio o *breakthrough*: Hinton GE, Osindero S e Teh Y-W. publicaram uma metodologia para treinar redes neurais com várias camadas em um intervalo de tempo factível!
- Esse tipo de abordagem hoje é conhecido como *deep learning*.
- A proposta revolucionou a inteligência artificial, aumentando incrivelmente a capacidade de treinamento de algoritmos de aprendizado e tornando possível todos os avanços que testemunhamos hoje.
- Em geral, aplicações de *deep learning* fazem uso extensivo de computação de alto desempenho (paralelismo, GPU's, etc.) e se destinam a estudos de larga escala (big data).

Deep Learning

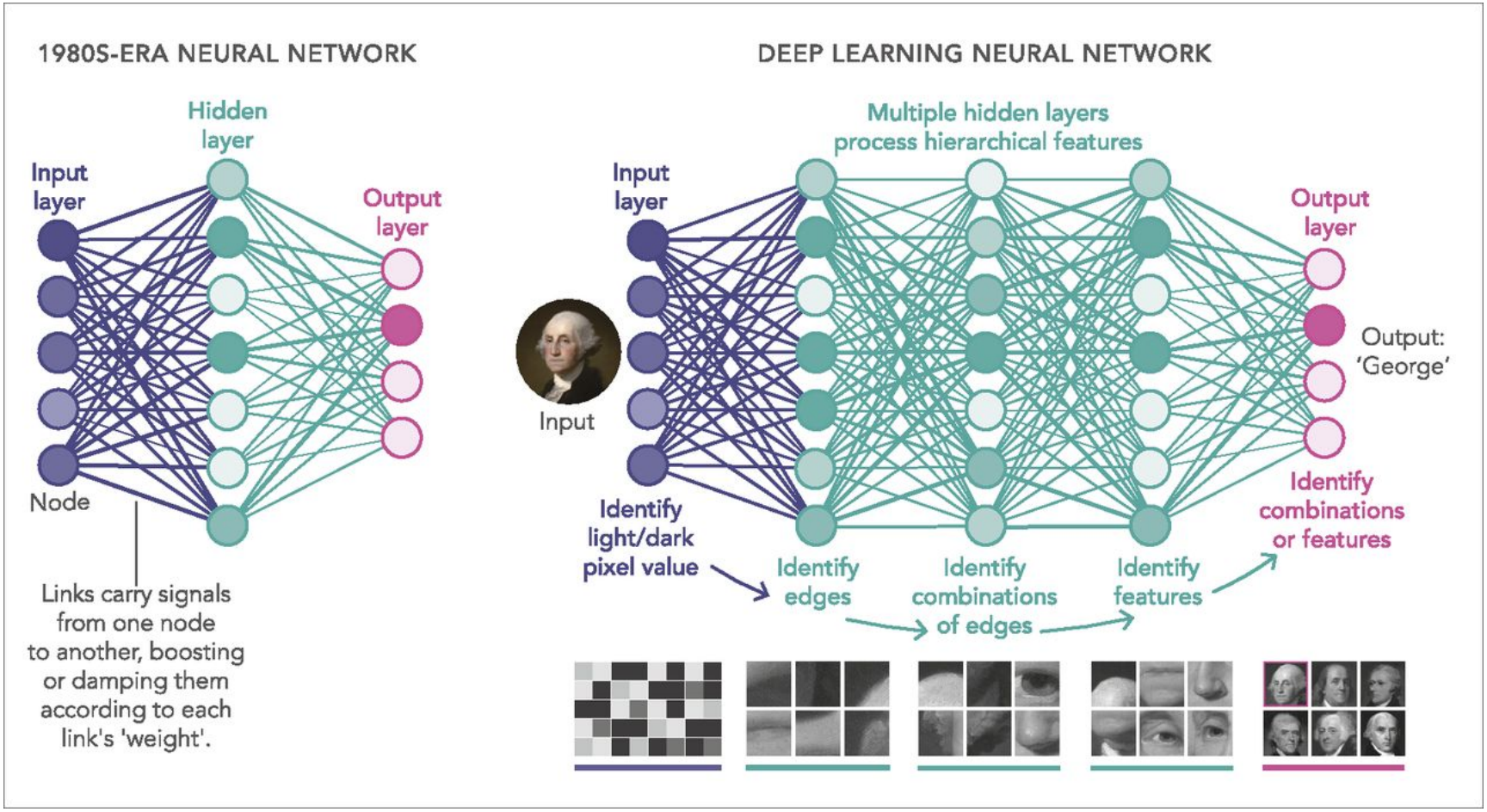


Figura 8: Comparação entre as arquitetura "tradicional" e "profunda" das redes neurais.

Deep Learning

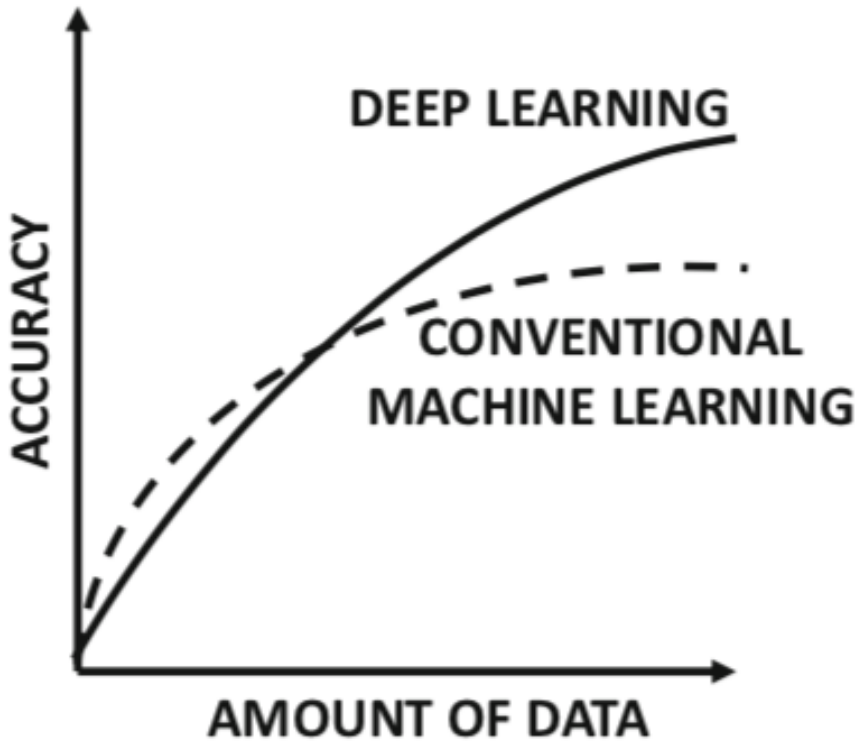


Figura 9: Relação entre acurácia, quantidade de dados e técnica adequada de aprendizado a ser aplicada.

Mão na massa!

Pra brincar

- Acesse o site playground.tensorflow.org. Lá tem uma rede neural de brinquedo pra você se divertir!

Mão na massa!

Uso no scikit-learn

- A técnica de rede neural disponível no scikit-learn é chamada de Multi-layer Perceptron.
- Leia o guia do scikit-learn e aplique a técnica a alguns dos problemas de classificação e regressão que estudamos ao longo do curso.
- Obs: tome cuidado, pois há um quê de inconsistência na nomenclatura, já que os neurônios a serem usados na rede não devem necessariamente ser percéptrons (há uma gama de funções de ativação não-lineares para escolha).

Mão na massa!

Indo além

- Esse é só para quem tem interesse de saber mais sobre *deep learning*. Pesquise sobre a biblioteca TensorFlow. A que ela se destina? Quais são as diferenças para o scikit-learn? Reflita: vale a pena, no seu caso profissional, ir além e aprender a usar essa biblioteca?
- Se a resposta é sim... mão na massa!!!

Obrigado pela atenção!