

Minicurso

Descomplicando o conceito de Redes Neurais

Quem sou

Me chamo Ronald Lopes, sou bacharelando em Engenharia de Computação pelo IFF, atualmente cursando o 10º período. Com mais de 2 anos de experiência na área de Engenharia de Software e desenvolvimento de sistemas embarcados.

GitHub: github.com/RonaldLopes

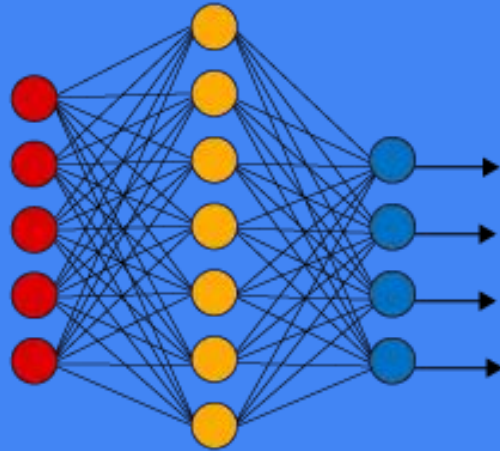
LinkedIn: www.linkedin.com/in/ronaldlopes/



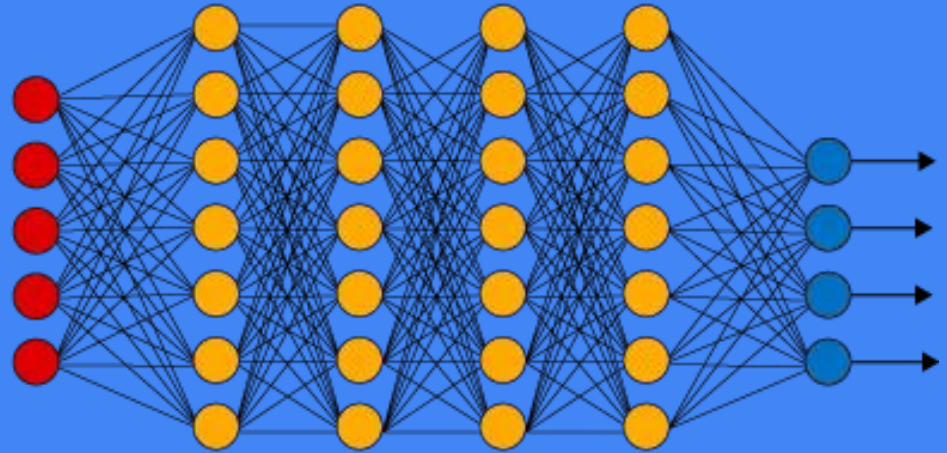
Objetivos

- O que é uma rede neural, seus tipos e como começar
- Entender o que é o TensorFlow e como ele trabalha
- Laboratório 1: Entendendo a estrutura do TensorFlow
- Estudo de caso: Classificando imagens de roupas com o Keras
- Laboratório 2: Classificando imagens de roupas com o Keras
- Estudo de caso: Classificando textos com a base de dados IMDB
- Laboratório 3: Classificando textos com a base de dados IMDB

Simple Neural Network



Deep Learning Neural Network



● Input Layer

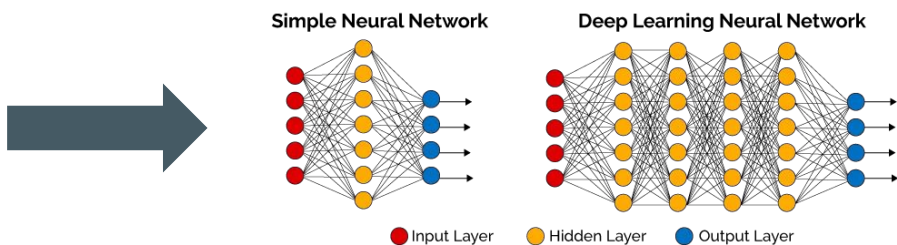
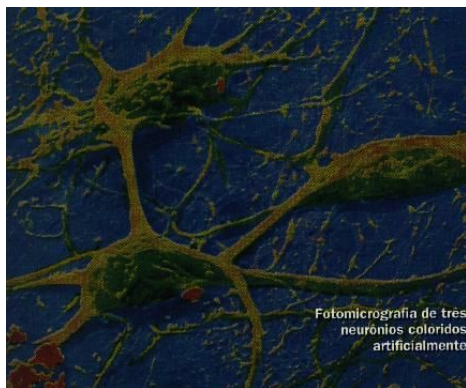
● Hidden Layer

● Output Layer

Redes Neurais

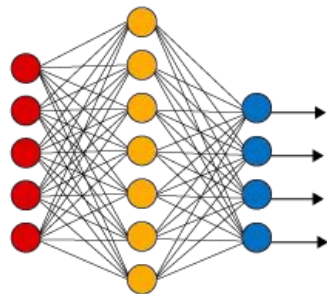
O que é?

Redes Neurais Artificiais consistem no emprego de algoritmos computacionais que apresentam um modelo matemático a fim de imitar a estrutura neural de organismos inteligentes e que adquirem conhecimento através da experiência.

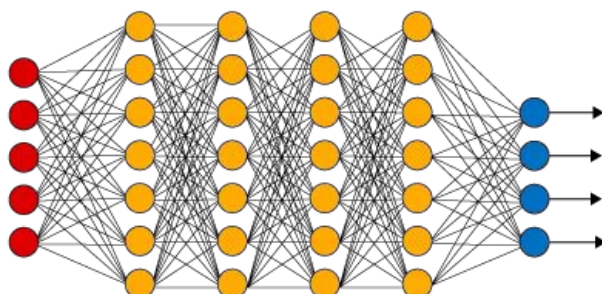


Estrutura de uma rede neural

Simple Neural Network



Deep Learning Neural Network



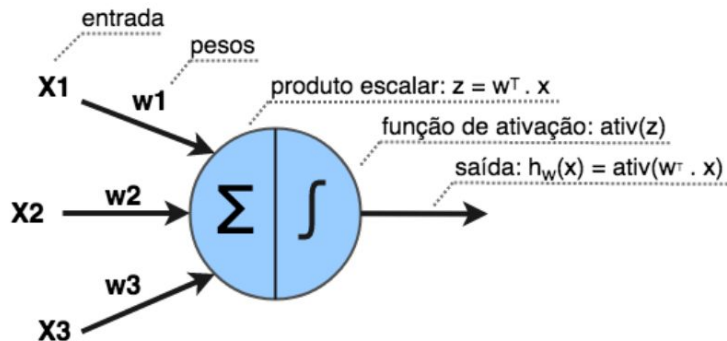
● Input Layer

● Hidden Layer

● Output Layer

- **Camada de Entrada:** aqui temos a entrada dos dados/padrões;
- **Camadas Escondidas:** a maior parte do processamento ocorre aqui
- **Camada de Saída:** onde o resultado final é concluído e apresentado.

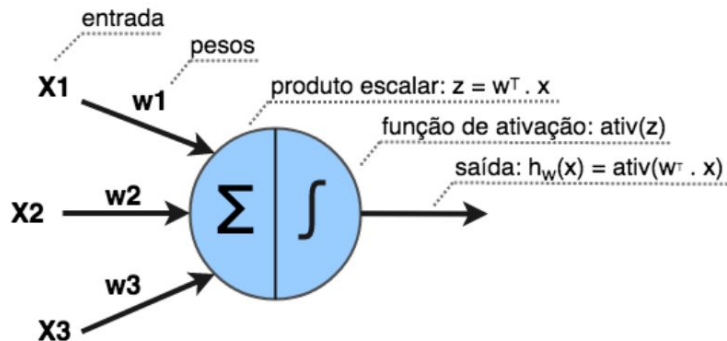
Como funciona um neurônio?



- Cada entrada x possui um peso w associado.
- É calculado o produto escalar entre os dados de entrada e seus pesos

$$(z = w_1 x_1 + w_2 x_2 + \dots + w_n x_n = w^t \cdot x)$$

Como funciona um neurônio?



- Então uma função de ativação é aplicada sobre o produto escalar, resultando na saída do neurônio:

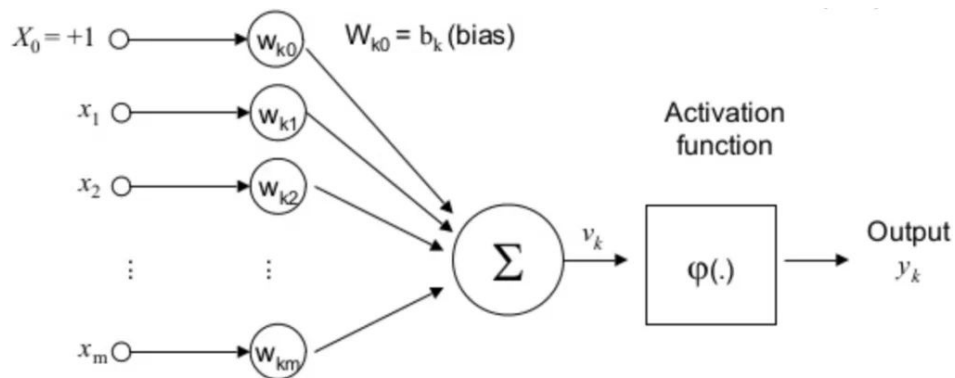
$$a_w(x) = \text{ativ}(z) = \text{ativ}(w^t \cdot x)$$

- O **b**ias é considerado, por padrão, um dado interno do neurônio, resultando na equação:

$$z = w^t \cdot x + b$$

Função de ativação

Basicamente decide se um neurônio deve ser ativado ou não. Permitindo saber se a informação que o neurônio está recebendo é relevante para a informação fornecida ou deve ser ignorada.



Tipos Populares de Funções de Ativação

Função de Etapa Binária

$$f(x) = 1, x \geq 0$$

$$f(x) = 0, x < 0$$

Função Linear

$$f(x) = ax$$

Sigmoid

$$f(x) = 1 / (1 + e^{-x})$$

ReLU

$$f(x) = \max(0, x)$$

Softmax

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad \text{for } j = 1, \dots, K.$$

Tipos de aprendizados

As tarefas de aprendizado de máquina geralmente são categorizados em três tipos:

- **Aprendizado supervisionado:** é fornecido amostras pré-classificadas para o aprendizado.
- **Aprendizado não-supervisionado:** é fornecido apenas os dados, sem qualquer classificação.
- **Aprendizado por reforço:** A máquina recebe sinais na forma de premiações ou punições a fim de gerar um feedback para o aprendizado.

Ciclos/Época de aprendizado

Um ciclo consiste na apresentação de todos os elementos do conjunto de treinamento no processo de aprendizado.

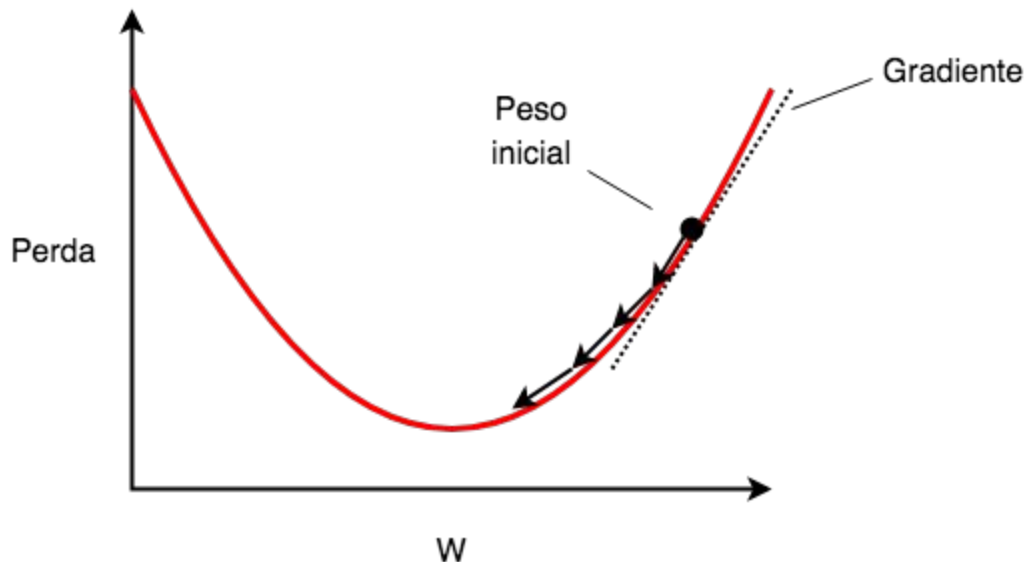
- **Modo Padrão:** Cada correção de pesos baseia-se somente no erro do exemplo apresentado naquela iteração. Assim, em cada ciclo ocorrem N correções.
- **Modo Batch:** Todos os exemplos do conjunto de treinamento são apresentados à rede, seu erro médio é calculado e a partir deste erro fazem-se as correções dos pesos.

Função de perda

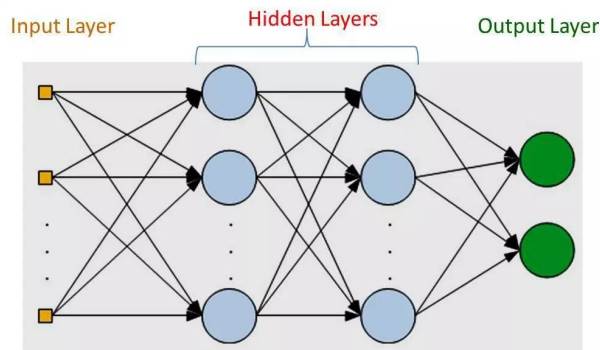
Tem a função de medir o nível próximo o treinamento está do resultado ideal esperado, ou seja, se o algoritmo estiver muito ruim, seu valor será alto. Logo esta função fica responsável pelo cálculo do valor de erro. Permitindo saber o quão longe o nosso treinamento está do resultado ideal.

Função de otimização

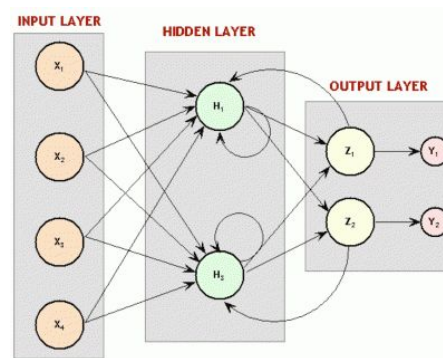
O principal objetivo da função de otimização é encontrar os parâmetros que minimizam a nossa função de perda.



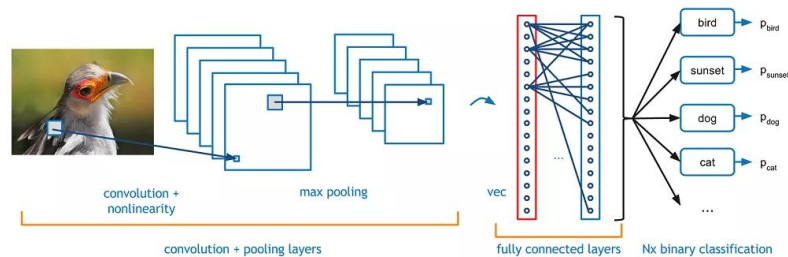
Principais tipos de Redes Neurais



Redes Multilayer Perceptrons (MLP)



Redes Neurais Recorrentes (RNN)



Redes Neurais Convolucionais (CNN)

Como começar a desenvolver minha aplicação?

5 passos essenciais para montar sua aplicação

1º Passo: Coleta de dados e separação em conjuntos

1. Análise cuidadosa sobre o problema para minimizar ambiguidades e erros nos dados.
2. Os dados devem ser significativos e cobrir amplamente o domínio do problema
3. Os dados são separados em dados de treinamento e dados de teste

2º Passo: Configuração da rede

- Selecionar o paradigma neural apropriado para a aplicação.
 - Determinar a topologia da rede a ser utilizada: o número de camadas, o número de neurônios em cada camada, etc.
 - Determinar os parâmetros do algoritmo de treinamento e as funções de ativação
-

3º Passo: Treinamento

- Definir a inicialização da rede (escolher os valores iniciais dos pesos da rede)
- Definir o modo de treinamento
- Definir o tempo de treinamento.

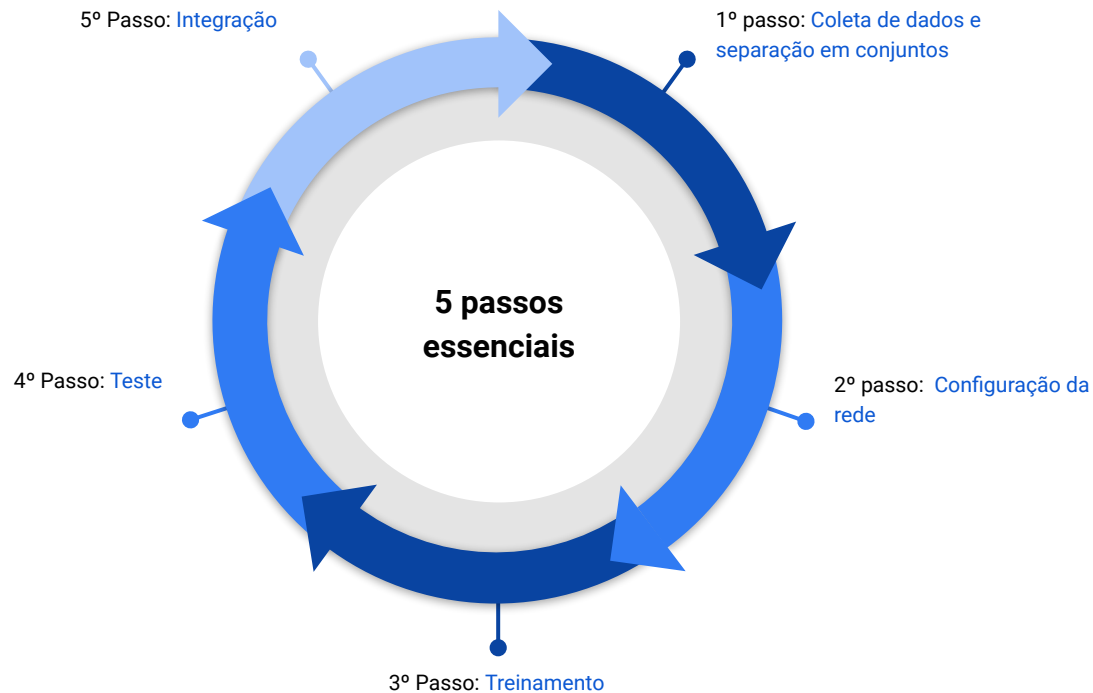
4º Passo: Teste

- Fase crucial para determinar a performance da rede
 - Deve-se utilizar dados que não foram previamente utilizados no treinamento
 - O resultado é uma boa indicação de sua performance real
-

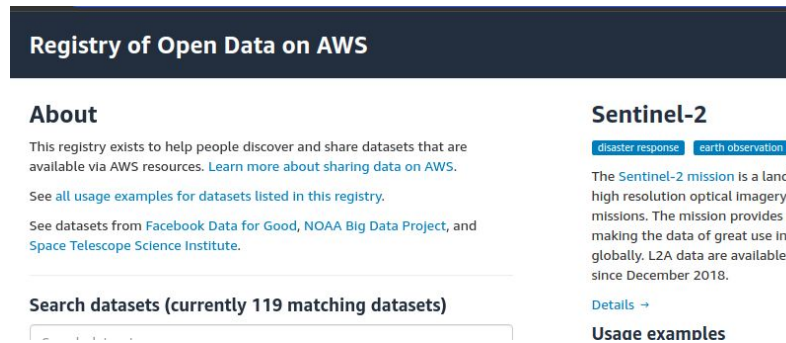
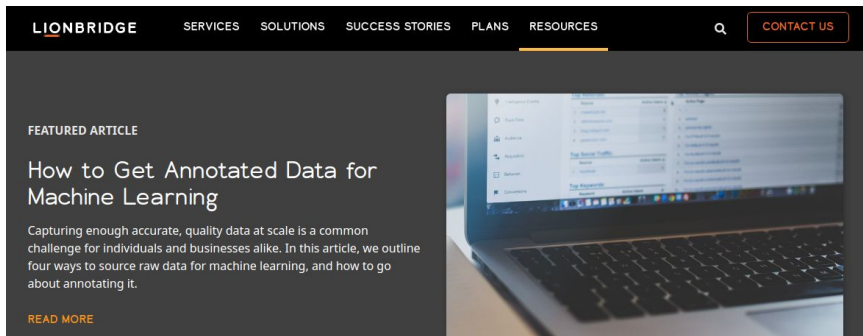
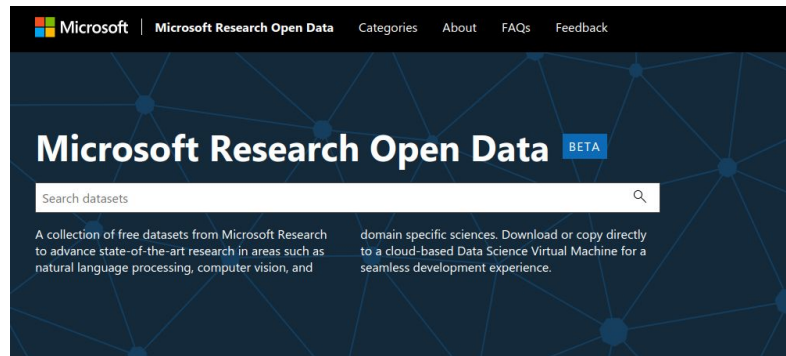
5º Passo: Integração

- Com a rede treinada e avaliada, ela pode ser integrada com uma aplicação que atenda as necessidades do seu problema.

Desenvolvendo uma aplicação



Como encontrar um dataset para treinar sua rede?

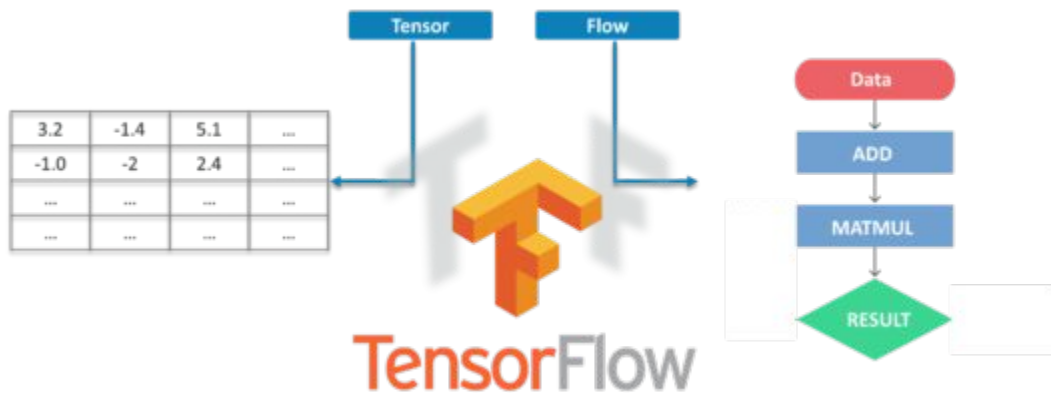




TensorFlow

O que é?

- É uma biblioteca de software de código aberto desenvolvida para a computação numérica usando grafos computacionais.
- Todas as operações do TensorFlow são representadas na forma de grafos (dataflow graph)
- Suporta linguagens como Python, Javascript, C/C++ e Go



Quem utiliza?



Estrutura

High-Level
TensorFlow APIs

Estimators

Mid-Level
TensorFlow APIs

Layers

Datasets

Metrics

Low-level
TensorFlow APIs

Python

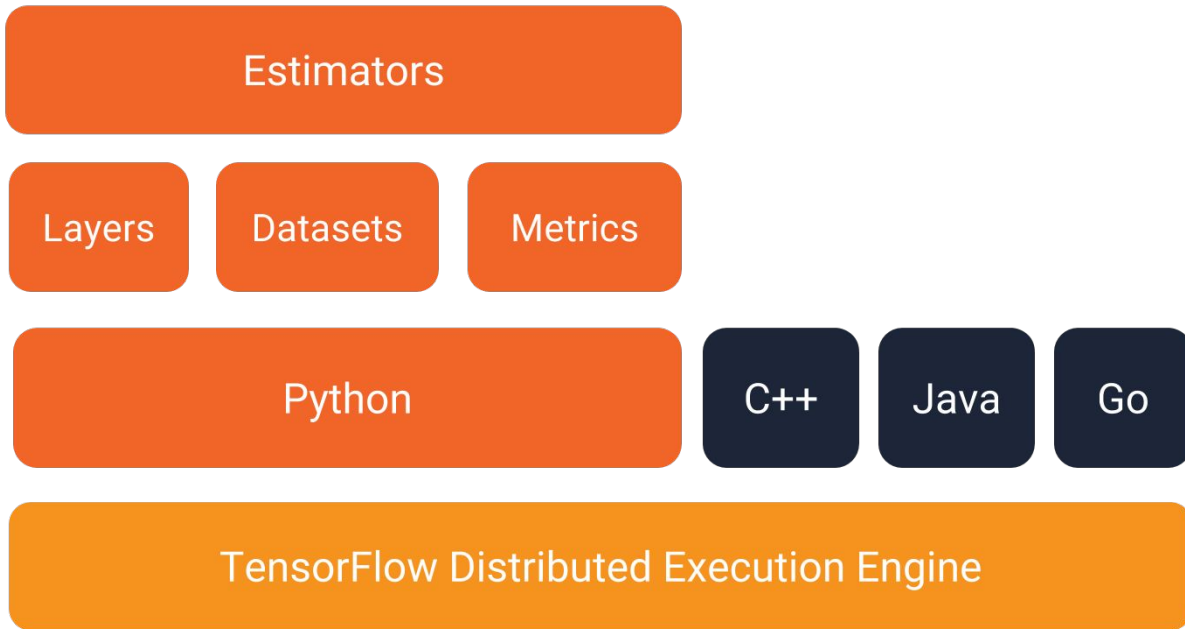
C++

Java

Go

TensorFlow
Kernel

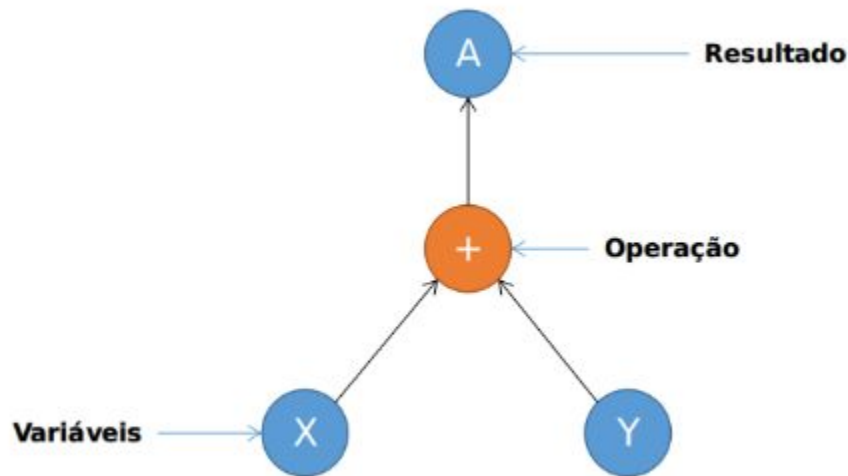
TensorFlow Distributed Execution Engine



Componentes básicos

Nós do grafo: representam as operações matemáticas.

Borda do grafo: representa os Tensores que seguem um fluxo entre as operações.







Laboratório 1: Conhecendo o TensorFlow

Laboratório 1




 lab1.ipynb

File Edit View Insert Runtime Tools Help [Last edited on October 28 by RonaldLopes](#)

+ Code + Text  Copy to Drive

Share   

Connect  Editing 

>

▾ Laboratório 1: Entendendo o funcionamento do Tensorflow

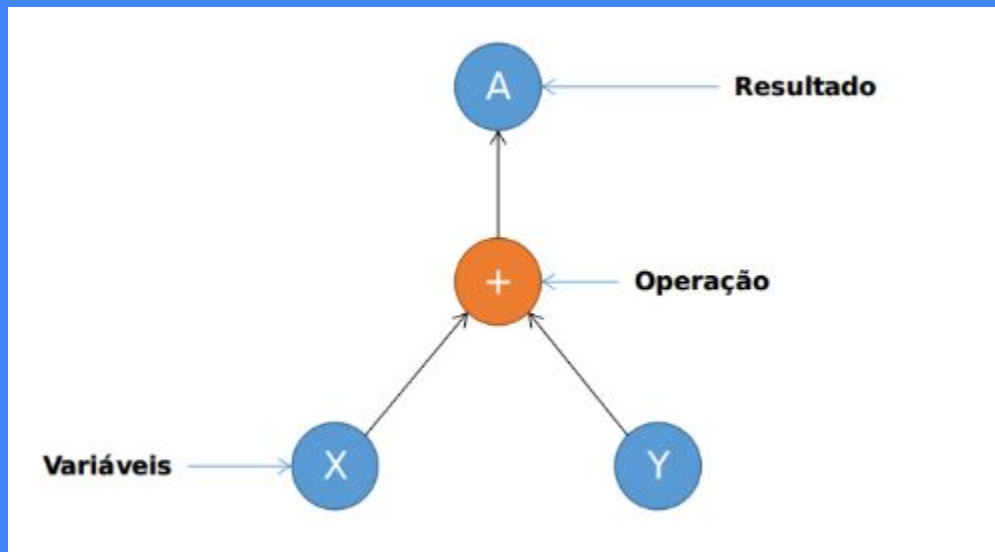
Bem vindo ao laboratório 1 do Mini curso de TensorFlow, neste tutorial vamos aprender a estrutura base de funcionamento do TensorFlow.



TensorFlow

O TensorFlow é uma biblioteca de software de código aberto desenvolvida para a computação numérica usando grafos computacionais, originalmente desenvolvido pela Google Brain Team na organização de pesquisa Machine Intelligence do Google para aprendizado de máquina e pesquisa de redes neurais profundas. O nome da biblioteca ajuda a compreender a forma de se trabalhar com ela:

Laboratório 1



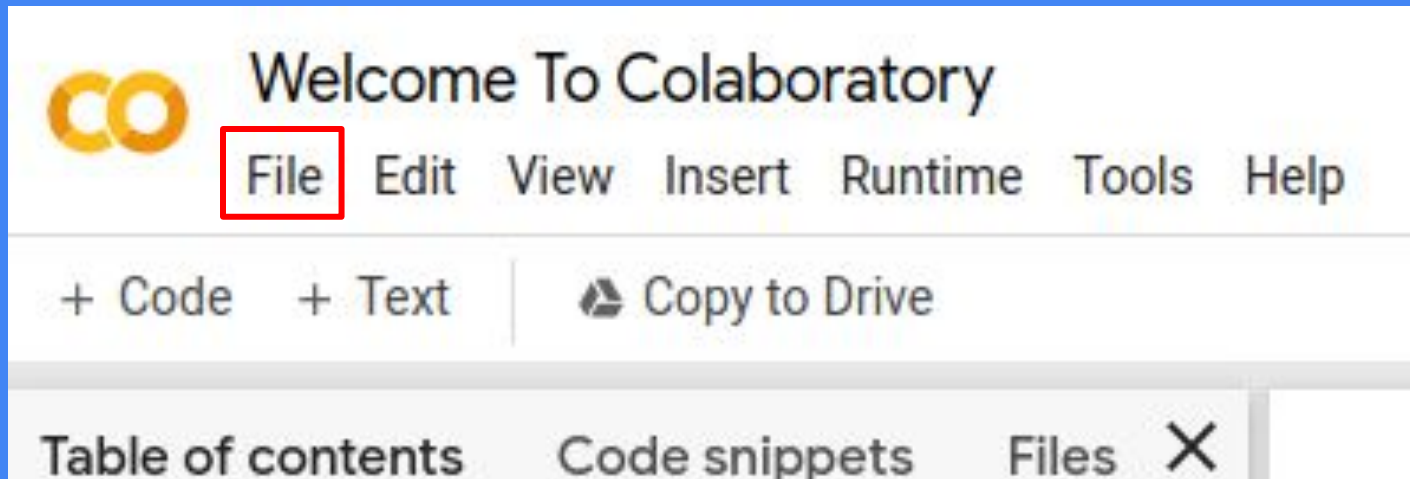
Acessando o material dos laboratórios



The screenshot displays the Google Colaboratory web interface. At the top, a white header bar contains the 'colab' logo, the text 'Welcome To Colaboratory', and a menu with 'File', 'Edit', 'View', 'Insert', 'Runtime', 'Tools', and 'Help'. Below the header, a toolbar includes '+ Code', '+ Text', 'Copy to Drive', 'Share', 'Connect', 'Editing', and a refresh icon. The main content area is divided into two panels. The left panel, titled 'Table of contents', lists: 'Introducing Colaboratory', 'Getting Started', 'More Resources', 'Machine Learning Examples: Seedbank', and a 'Section' with a plus icon. The right panel displays the 'Welcome to Colaboratory!' message, followed by a description of the environment as a free Jupyter notebook that runs in the cloud. Below this, a video player is embedded, showing a thumbnail for 'Intro to Google Colab' with a play button and a man's face. The video player has controls for 'Assistir mais tarde' and 'Compartilhar'.

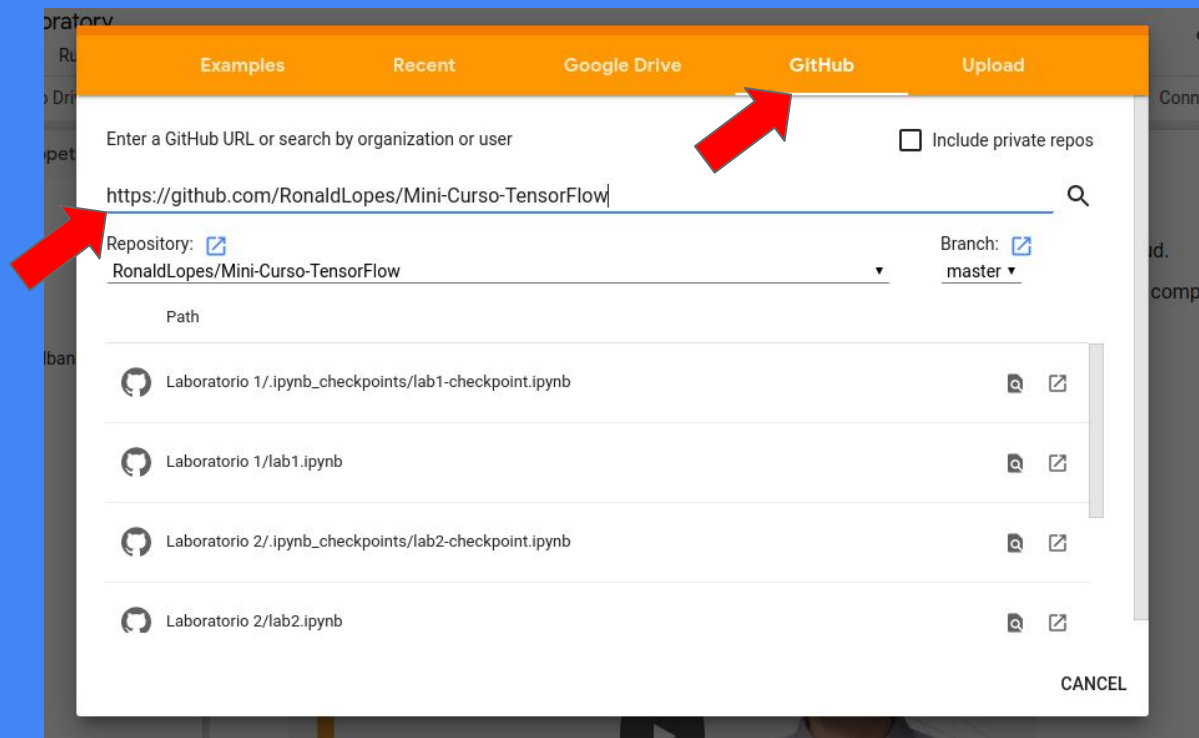
<https://colab.research.google.com>

Acessando o material dos laboratórios



File > Open Notebook

Acessando o material dos laboratórios

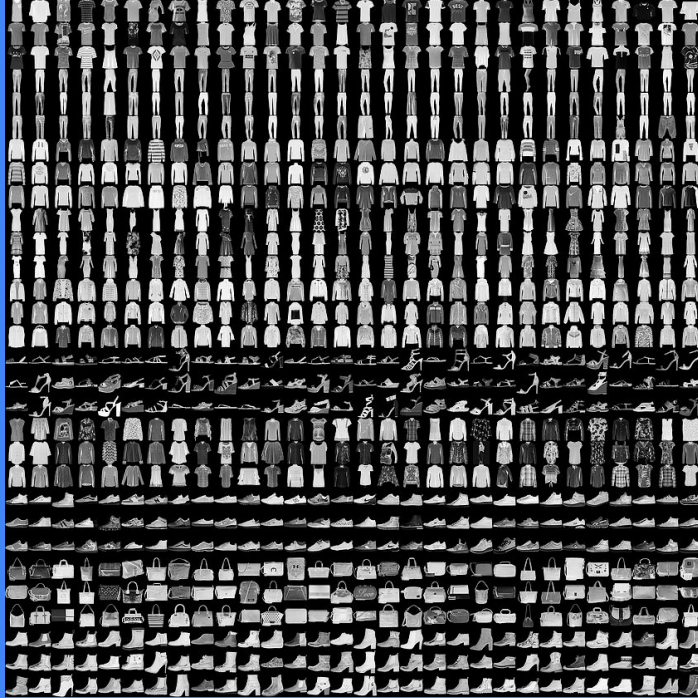


<https://github.com/RonaldLopes/Mini-Curso-TensorFlow>

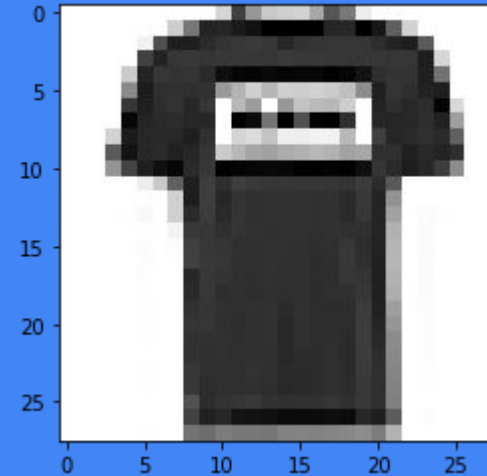


Laboratório 2: Classificando imagens de roupas com o Keras

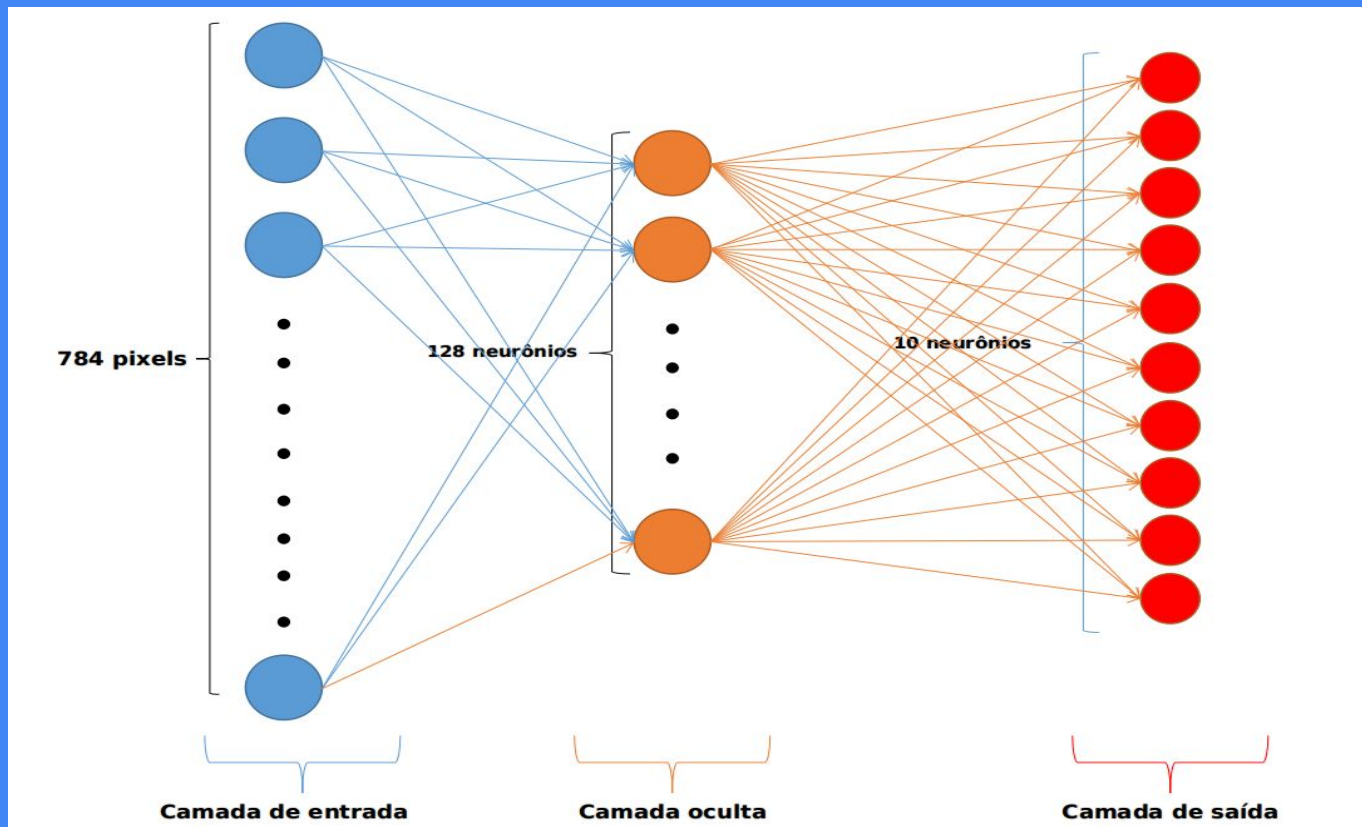
Laboratório 2 - Dataset



70.000 imagens



Laboratório 2 - Rede Neural



Laboratório 2 - Objetivo Final



Classe reconhecida: Botas



Figura utilizada: Botas

Classe reconhecida: Suéter



Figura utilizada: Suéter

Classe reconhecida: Calça

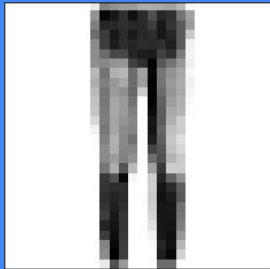


Figura utilizada: Calça

Classe reconhecida: Camisa



Figura utilizada: Camisa

Referencias

<http://deeplearningbook.com.br/as-10-principais-arquiteturas-de-redes-neurais/>

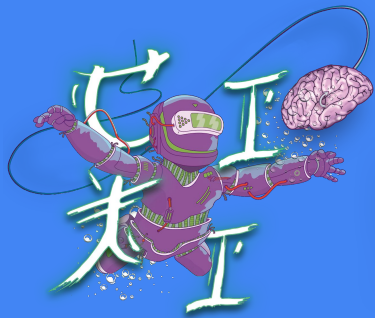
<http://conteudo.icmc.usp.br/pessoas/andre/research/neural/desenv.html> (
Laboratory of Computational Intelligence (LABIC))

<http://conteudo.icmc.usp.br/pessoas/andre/research/neural/>

<http://deeplearningbook.com.br/funcao-de-ativacao/>

[https://towardsdatascience.com/top-sources-for-machine-learning-datasets-bb6d0d
c3378b](https://towardsdatascience.com/top-sources-for-machine-learning-datasets-bb6d0dc3378b)

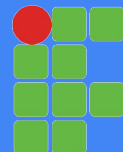
<https://www.tensorflow.org/>



X_CONGRESSO_
INTEGRADO_DA_
TECNOLOGIA_DA_
_INFORMACAO.pyo

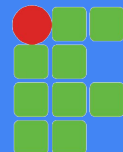


Obrigado!



INSTITUTO FEDERAL
FLUMINENSE
Campus Campos Centro

FIM



INSTITUTO FEDERAL
FLUMINENSE
Campus Campos Centro