Lidiana Silva de Oliveira – SP3046541

Thomas Wohlers Manzi – SP3050106

TensorFlow.js

TADS – DW2A2

SÃO PAULO

2021

**Sumário**

[**TensorFlow: O que é?** 2](#_Toc85686561)

[**TensorFlow e a Comunidade de Software de Código Aberto** 3](#_Toc85686562)

[**Estrutura** 4](#_Toc85686563)

[**Quem utiliza o TensorFlow?** 4](#_Toc85686564)

[**TensorFlow.js** 5](#_Toc85686565)

[**Como se utiliza o TensorFlow na prática?** 5](#_Toc85686566)

[**Aplicações:** 8](#_Toc85686567)

# **TensorFlow: O que é?**

Criado pela equipe do Google Brain (segmento do Google focado em pesquisas na área de inteligência artificial) e lançado sob a licença de código aberto em 2015, o TensorFlow é uma biblioteca de código aberto para computação numérica e aprendizado de máquina em grande escala. Agrupa uma série de modelos e algoritmos de aprendizado de máquina e aprendizado profundo (também conhecido como rede neural) que facilitam o processo de aquisição de dados, modelos de treinamento, previsões de atendimento e refinamento de resultados futuros.

É uma plataforma altamente escalável de Machine Learning que pode ser executada em smartphones, computadores ou data centers. A plataforma é open source e todos os algoritmos especializados para resolver questões específicas são disponibilizados na nuvem através de APIs. Graças a sua adaptabilidade, permite construir e treinar redes neurais e rapidamente gerar um produto ou serviço a partir do modelo preditivo treinado. Por isso, suas aplicações são inúmeras e podem ser usadas por indivíduos em busca de pesquisas ou mesmo grandes empresas que precisam implementar estratégias de Inteligência Artificial.

Atualmente está em sua versão 2.0 O TensorFlow 2.0, lançado em outubro de 2019, renovou a estrutura de várias maneiras de modo a torná-lo mais simples e proporcionar maior desempenho.

# **TensorFlow e a Comunidade de Software de Código Aberto**

O TensorFlow foi aberto para a comunidade open-source afim de permitir contribuições que aprimorassem a biblioteca. A equipe do TensorFlow configurou processos para gerenciar solicitações de extração, revisar e rotear os problemas arquivados e responder às perguntas do [StackOverflow](https://stackoverflow.com/questions/tagged/tensorflow" \t "_blank). Ou seja, o Google monitora o avanço e desenvolvimento do TensorFlow.

Até agora, já são mais de 890 contribuidores externos, contribuindo desde pequenas correções de documentação até grandes adições como o suporte a GPU do OSX ou a implementação do OpenCL. (A organização mais ampla do TensorFlow no GitHub tem quase 1.000 colaboradores únicos que não são Googlers).

O TensorFlow tem mais de 109.000 estrelas no [GitHub](https://github.com/tensorflow/tensorflow), e o número de outros repositórios que o usam cresce a cada mês. O StackOverflow é monitorado pela equipe do TensorFlow, e é uma boa maneira de obter respostas às perguntas (com mais de 15.000 respostas até agora).

A versão externa (open-source) do TensorFlow não é diferente da versão interna (exclusiva do Google), além de algumas pequenas diferenças. Isso inclui a interface com a infraestrutura interna do Google (não ajudaria ninguém), alguns caminhos e peças que ainda não estão prontas. O núcleo do TensorFlow, no entanto, é idêntico. As atualizações da versão interna aparecem externamente em cerca de um dia e meio e vice-versa.

No repositório do Github no [TensorFlow](https://github.com/tensorflow" \t "_blank), você pode encontrar não apenas o TensorFlow em si, mas também um ecossistema útil de outros repos, incluindo modelos, serviços, TensorBoard, Project Magenta e muitos outros. (Alguns destes são descritos abaixo). Você também pode encontrar APIs do TensorFlow em várias linguagens (Python, C ++, Java e Go) e a comunidade desenvolveu outras extensões, incluindo C#, Haskell, Julia, Ruby, Rust e Scala.

# **Estrutura**

As bibliotecas do TensorFlow consistem em várias APIs que podem ser divididas em duas categorias: baixo nível e alto nível.

**APIs TensorFlow de baixo nível**

A TensorFlow Core API é o mecanismo principal do TensorFlow . Os desenvolvedores podem integrar o TensorFlow Core em seu próprio código Python ou JavaScript, permitindo que criem aplicativos de aprendizado de máquina poderosos.

O TensorFlow Core é poderoso, mas tem uma curva de aprendizado íngreme. Qualquer pessoa que trabalhe com o Core deve entender não apenas a API principal, mas também os conceitos de dados que formam a base do TensorFlow.

**APIs TensorFlow de alto nível**

Trata-se de uma coleção de ferramentas e bibliotecas de alto nível executadas no TensorFlow. Alguns deles ajudam a criar modelos que podem formar a base de um gráfico. Outros fornecem uma camada modular que possibilita o desenvolvimento sem conhecer todos os prós e contras do TensorFlow.

Muitas dessas APIs são menores e mais consistentes do que a API principal e vêm com uma curva de aprendizado muito mais flexível.

# **Quem utiliza o TensorFlow?**

O TensorFlow é utilizado tanto em aplicações comerciais do dia a dia nas empresas (como cientistas de dados e engenheiros de dados) como para pesquisadores que estão desenvolvendo novos algoritmos e recursos para aprendizado de máquina.

Algumas empresas que já declararam abertamente que utilizam o TensorFlow em suas aplicações são: Coca-Cola, Intel, Airbus, General Electric (GE), Twitter, Airbnb, Paypal, e muitas outras.

O TensorFlow é multi-plataforma, podendo ser executado no Windows, MacOS ou Linux. Além disso, também pode ser executado em CPUs, GPUs (placas de vídeo, que aceleram o processamento), ou ainda TPUs (que são circuitos integrados específicos desenvolvidos pelo Google para acelerar o aprendizado de máquina).

Já existem APIs do TensorFlow em várias linguagens: Python, JavaScript, C++, Java, Go, Swift, C#, Haskell, Julia, Ruby, Rust e Scala.

# **TensorFlow.js**

TensorFlow.js é uma biblioteca JavaScript para treinar e implantar modelos de aprendizado de máquina em aplicativos da Web e em Node.js. Você pode desenvolver os modelos de aprendizado de máquina do zero usando tensorflow.js ou pode usar as APIs fornecidas para treinar seus modelos existentes no navegador ou em seu servidor Node.js.

# **Como se utiliza o TensorFlow na prática?**

Talvez você já esteja convencido de que aprender TensorFlow é importante. Chegou a hora de entender um pouco melhor o que é um código TensorFlow e o que ele faz.

Podemos pensar no TensorFlow como sendo uma espécie de linguagem de [programação](https://didatica.tech/programacao-para-quem-nao-e-programador-como-aprender-com-facilidade/), onde escrevemos um código e executamos. Na realidade, o TensorFlow é um [Framework](https://didatica.tech/frameworks-na-pratica-das-redes-neurais/), ou seja, uma união de códigos que visa a uma aplicação. Mas em vez de ficarmos detalhando uma definição rigorosa, é mais fácil enxergar na prática o que ele faz.

Programar em TensorFlow é muito semelhante a programar em Python (principalmente o [pacote Numpy](https://didatica.tech/o-pacote-numpy-python-para-machine-learning/) do Python). Se você já sabe alguma coisa de [Python](https://didatica.tech/curso-de-python-online-para-iniciantes/) (pacotes e funções), certamente não terá nenhuma dificuldade em aprender TensorFlow.

Depois que o TensorFlow está instalado, você pode utilizar a mesma interface (IDE) que utiliza para programar Python para rodar seus códigos TensorFlow. Inclusive, você pode escrever códigos Python misturados com TensorFlow e rodar tudo junto! Por isso, é mais fácil pensar no TensorFlow como se fosse uma biblioteca Python, da mesma forma que temos os pacotes/bibliotecas Numpy, [Scikit-learn](https://didatica.tech/a-biblioteca-scikit-learn-pyhton-para-machine-learning/), [Pandas](https://didatica.tech/o-pacote-pandas-python-para-machine-learning/), etc.

Mas, ao contrário das bibliotecas Python tradicionais, para executar um código TensorFlow existem alguns detalhes específicos, como a abertura de uma sessão.

Observe o código TensorFlow abaixo que imprime na tela a frase “Olá, mundo!”:

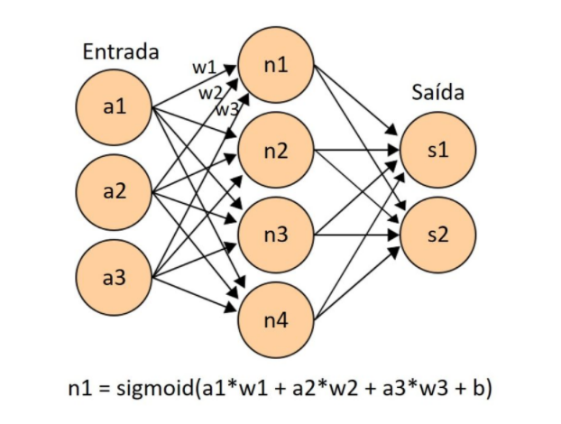
*import tensorflow as tf  
   
frase = tf.constant("Olá mundo!")  
   
with tf.Session() as sess:  
    rodar = sess.run(frase)  
   
print(rodar)*

Primeiro importamos o TensorFlow, mas para que o código seja executado, abrimos uma sessão *tf.Session()*.

Exemplo de criação de uma rede neural utilizando TensorFlow:

*# Pesos da camada 1   
 w1 = tf.Variable(tf.random\_normal([n\_input, n\_hidden\_1]))  
 # Bias da camada 1  
 b1 = tf.Variable(tf.random\_normal([n\_hidden\_1]))  
 # Aplicando a função sigmoide na camada 1  
 camada\_1 = tf.nn.sigmoid(tf.add(tf.matmul(x,w1),b1))*

Esse código acima está criando uma camada de uma rede neural. A figura abaixo mostra um exemplo do que seria uma rede neural básica:



No exemplo acima, a camada de neurônios n1, n2, n3 e n4 é resultado de uma combinação dos valores de entrada a1, a2 e a3 com pesos w que multiplicam essas entradas, e uma constante b ainda é adicionada no final. Sobre essa combinação de variáveis, a [função sigmoide](https://www.youtube.com/watch?v=DlBhJdHQElI) é aplicada para fazer com que o valor da saída fique entre 0 e 1.