

# Bibliotecas para implementação de IA

Gustavo Pellozo de Oliveira  
Ciência da Computação  
Centro Educacional Guaxupé - Unifeg  
Guaxupé, Brasil  
gpellozo@gmail.com

Gustavo Poli  
Ciência da Computação  
Centro Educacional Guaxupé - Unifeg  
Guaxupé, Brasil  
gustavo.poli@unifeg.edu.br

## I. INTRODUÇÃO À INTELIGENCIA ARTIFICIAL

Inteligência artificial começa como, praticamente, todo projeto de programação: ideia, objetivos e linhas de código.

A inteligência artificial (IA) possibilita que máquinas aprendam com experiências, se ajustem a novas entradas de dados e performem tarefas como seres humanos. A maioria dos exemplos de IA sobre os quais você ouve falar, computadores mestres em xadrez a carros autônomos dependem de deep learning e processamento de linguagem natural. Com essas tecnologias, os computadores podem ser treinados para cumprir tarefas específicas ao processar grandes quantidades de dados e reconhecer padrões nesses dados.

O termo inteligência artificial foi criado em 1956, mas só se popularizou hoje graças aos crescentes volumes de dados disponíveis, algoritmos avançados, e melhorias no poder e no armazenamento computacionais.

As primeiras pesquisas de IA nos anos 1950 exploraram temas como a resolução de problemas e métodos simbólicos. Na década de 1960, o Departamento de Defesa dos EUA se interessou por este tipo de tecnologia e começou a treinar computadores para imitar o raciocínio humano básico. Por exemplo, a Defense Advanced Research Projects Agency (DARPA) completou um projeto de mapeamento de ruas nos anos 1970. E a DARPA criou assistentes pessoais inteligentes em 2003, muito tempo antes de Siri, Alexa ou Cortana serem nomes comuns do nosso cotidiano.

Esses primeiros trabalhos prepararam o caminho para a automação e o raciocínio formal que vemos nos computadores de hoje, incluindo sistemas de apoio à decisão e sistemas inteligentes de pesquisa que podem ser projetados para complementar e expandir as capacidades humanas.

Enquanto os filmes de Hollywood e os romances de ficção científica retratam a inteligência artificial como robôs humanoides que dominam o mundo, a evolução atual das tecnologias de IA não é tão assustadora – ou tão inteligente assim. Em vez disso, a IA evoluiu para fornecer muitos benefícios específicos para todas as indústrias.

A aplicação trás vantagens como de automatizar a aprendizagem repetitiva e a descoberta a partir dos dados, adiciona inteligência a produtos existentes, se adapta através de algoritmos de aprendizagem progressiva para deixar que os dados façam a programação, analisa e mais dados e em maior profundidade usando redes neurais que possuem muitas camadas escondidas, atinge uma precisão incrível através de redes neurais profundas obtém o máximo dos dados.

## II. PYTHON

Uma linguagem que está em ascensão, onde diversas empresas já a adotaram como por exemplo o Google, a NASA, a IBM, a Embratel e o Serpro. Disponibiliza recursos

para IA, com várias bibliotecas para aprendizagem de máquina e análise de dados. Pelo fato da usabilidade dentro das empresas e instituições que disponibilizam tal conhecimento, mantem-se uma comunidade ativa.

Antes de começar, você pode descobrir quais IDEs e editores de texto são adaptados para facilitar a edição do Python, navegar na lista de livros introdutórios ou ver exemplos de códigos que podem ser úteis.

Python é uma linguagem de programação poderosa e fácil de aprender. Possui estruturas de dados de alto nível eficientes e uma abordagem simples, mas eficaz, da programação orientada a objetos. A sintaxe elegante do Python e a digitação dinâmica, juntamente com sua natureza interpretada, o tornam uma linguagem ideal para scripts e desenvolvimento rápido de aplicativos em muitas áreas da maioria das plataformas.

O interpretador Python e a extensa biblioteca padrão estão disponíveis gratuitamente na forma de código-fonte ou binário para todas as principais plataformas do site Python, <https://www.python.org/>, e podem ser distribuídos gratuitamente. O mesmo site também contém distribuições e ponteiros para muitos módulos, programas e ferramentas Python de terceiros gratuitos e documentação adicional.

O interpretador Python é facilmente estendido com novas funções e tipos de dados implementados em C ou C++ (ou outras linguagens que podem ser chamadas de C). O Python também é adequado como uma linguagem de extensão para aplicativos personalizáveis.

## III. BIBLIOTECAS PYTHON PARA IA

### A. NLTK

Trabalha com dados da linguagem humana. Ele fornece interfaces fáceis de usar com recursos lexicais como o WordNet, juntamente com um conjunto de bibliotecas de processamento de texto para classificação, tokenização, stemming, marcação, análise, análise e raciocínio semântico, invólucros para bibliotecas NLP de nível industrial, e um fórum de discussão ativo.

Possui um guia prático que introduz os fundamentos da programação, além de tópicos em linguística computacional, além de documentação abrangente da API. O NLTK está disponível para Windows, Mac OS X e Linux. Projeto gratuito, de código aberto.

O Processamento de linguagem natural com Python fornece uma introdução prática à programação para processamento de linguagem. Escrito pelos criadores do NLTK, ele orienta o leitor sobre os fundamentos da criação de programas em Python, trabalhando com corpora, categorizando texto, analisando a estrutura linguística e muito mais

A versão NLTK 1.4.3 foi lançada em fevereiro de 2005; Esta é a primeira versão compatível com o Python 2.4. NLTK 3.4.5 é a última versão atual, com Lançamento do agosto de 2019.

#### B. Theano

É uma biblioteca Python que permite definir, otimizar e avaliar expressões matemáticas, especialmente aquelas com matrizes multidimensionais (numpy.ndarray). Usando o Theano, é possível atingir velocidades que rivalizam com implementações C criadas manualmente para problemas que envolvem grandes quantidades de dados. Ele também pode superar o C em uma CPU em muitas ordens de magnitude, aproveitando as GPUs recentes.

- forte integração com o NumPy - Use numpy.ndarray nas funções compiladas pelo Theano .
- uso transparente de uma GPU - Realize cálculos com muitos dados muito mais rápido do que em uma CPU.
- diferenciação simbólica eficiente - o Theano faz suas derivadas para funções com uma ou várias entradas.
- otimizações de velocidade e estabilidade - obtenha a resposta certa  $\log(1+x)$  mesmo quando  $x$  é realmente pequeno.
- geração dinâmica de código C - Avalie expressões mais rapidamente.
- testes de unidade extensivos e auto-verificação - Detecte e diagnostique muitos tipos de erros.

#### C. PyTorch

É um pacote de computação científica baseado em Python direcionado a dois conjuntos de públicos:

Um substituto para o NumPy usar o poder das GPU soma plataforma de pesquisa de aprendizado profundo que fornece flexibilidade e velocidade máximas

permite experimentação rápida, flexível e produção eficiente por meio de um front-end fácil de usar, treinamento distribuído e ecossistema de ferramentas e bibliotecas.

#### PRINCIPAIS RECURSOS E CAPACIDADES

TorchScript facilidade de uso e flexibilidade no modo ansioso, enquanto faz a transição perfeita para o modo gráfico para velocidade, otimização e funcionalidade em ambientes de tempo de execução.

Treinamento Distribuído desempenho em pesquisa e produção, aproveitando o suporte nativo para execução assíncrona de operações coletivas e comunicação ponto a ponto

Ferramentas e Bibliotecas Um rico ecossistema de ferramentas e bibliotecas estende o PyTorch e suporta o desenvolvimento em visão computacional, PNL.

#### D. Scikit-Learn

O Scikit-learn é um módulo Python que integra uma ampla gama de algoritmos de aprendizado de máquina de ponta para problemas supervisionados e não supervisionados de média escala. Este pacote se concentra em levar o aprendizado de máquina para não especialistas, usando uma

linguagem de alto nível para uso geral. A ênfase é colocada na facilidade de uso, desempenho, documentação e consistência da API. Possui dependências mínimas e é distribuído sob a licença BSD simplificada, incentivando seu uso em ambientes acadêmicos e comerciais.

é uma biblioteca de código aberto. inclui algoritmos de classificação, regressão e agrupamento incluindo máquinas de vetores de suporte, florestas aleatórias, gradient boosting, k-means e DBSCAN, e é projetada para interagir com as bibliotecas Python numéricas e científicas NumPy e SciPy.

Este projeto foi iniciado em 2007 como um projeto do Google Summer of Code por David Cournapeau. Em 2015, scikit-learn está sob desenvolvimento ativo e é patrocinado por INRIA, Telecom ParisTech e ocasionalmente pelo Google (através do Google Summer of Code). The word “data” is plural, not singular.

#### E. Keras

é uma API de alto nível redes neurais, escrito em Python e capaz de correr em cima de TensorFlow , CNTK , ou Theano . Foi desenvolvido com o objetivo de permitir experimentação rápida. Ser capaz de passar da ideia para o resultado com o menor atraso possível é essencial para fazer uma boa pesquisa.

Permite prototipagem fácil e rápida (através da facilidade de uso, modularidade e extensibilidade).

Suporta redes convolucionais e redes recorrentes, bem como combinações das duas.

Funciona perfeitamente na CPU e GPU.

Facilidade de uso. Keras é uma API projetada para seres humanos, não máquinas. Ele coloca a experiência do usuário na frente e no centro. Segue as práticas recomendadas para reduzir a carga cognitiva: oferece APIs simples e consistentes, minimiza o número de ações do usuário necessárias para casos de uso comuns e fornece feedback claro e acionável mediante erro do usuário.

Modularidade. Um modelo é entendido como uma sequência ou um gráfico de módulos autônomos totalmente configuráveis que podem ser conectados com o mínimo de restrições possível. Em particular, camadas neurais, funções de custo, otimizadores, esquemas de inicialização, funções de ativação e esquemas de regularização são todos módulos independentes que você pode combinar para criar novos modelos.

Fácil extensibilidade. Novos módulos são simples de adicionar (como novas classes e funções), e os módulos existentes fornecem amplos exemplos. Ser capaz de criar facilmente novos módulos permite expressividade total, tornando o Keras adequado para pesquisas avançadas.

Trabalhe com Python . Não há arquivos de configuração de modelos separados em um formato declarativo. Os modelos são descritos no código Python, que é compacto, mais fácil de depurar e permite facilidade de extensibilidade.

Instale o Keras a partir do PyPI (recomendado):

Essas etapas de instalação pressupõem que você esteja em um ambiente Linux ou Mac. Se você estiver no Windows, será necessário remover sudo para executar os comandos abaixo.

```
sudo pip install keras
```

A estrutura de dados principal do Keras é o modelo de organizar camadas. O tipo mais simples de modelo é o Sequential, uma pilha linear de camadas. Para arquiteturas mais complexas, você deve usar a API funcional Keras, que permite criar gráficos arbitrários de camadas.

```
from keras.models import Sequential

model = Sequential()
```

Empilhar camadas é tão fácil quanto .add():

```
from keras.layers import Dense

model.add(Dense(units=64, activation='relu', input_dim=100))
model.add(Dense(units=10, activation='softmax'))
```

Quando seu modelo estiver bom, configure seu processo de aprendizado com .compile():

```
model.compile(loss='categorical_crossentropy',
              optimizer='sgd',
              metrics=['accuracy'])
```

#### F. TensorFlow

O TensorFlow concentra-se na simplicidade e facilidade de uso, com atualizações como execução rápida, APIs intuitivas de nível superior e construção de modelo flexível em qualquer plataforma. Biblioteca de código aberto para computação numérica de forma ágil, criada e lançada pelo Google. Foi programado para ser utilizado em pesquisa e desenvolvimento e em sistemas de produção.

facilita para iniciantes e especialistas a criação de modelos de aprendizado de máquina para desktop, celular, web e nuvem. O TensorFlow fornece uma coleção de fluxos de trabalho para desenvolver e treinar modelos usando Python, JavaScript ou Swift.

Como instalar o TensorFlow

A instalação do TensorFlow é simples se você já possui um ambiente Python. Instale o TensorFlow com o gerenciador de pacotes pip do Python.

```
$ pip install --upgrade pip

$ pip install tensorflow

$ pip install tf-nightly
```

Seus primeiros exemplos no TensorFlow

A computação é descrita em termos de fluxo de dados e operações na estrutura de um gráfico direcionado.

Nós : os nós realizam o cálculo e possuem zero ou mais entradas e saídas. Os dados que se movem entre os nós são conhecidos como tensores, que são matrizes multidimensionais de valores reais.

Arestas : o gráfico define o fluxo de dados, ramificação, loop e atualizações para o estado. Arestas especiais podem ser usadas para sincronizar o comportamento dentro do gráfico, por exemplo, aguardando o cálculo de várias entradas para serem concluídas.

Operação : Uma operação é um cálculo abstrato nomeado que pode receber atributos de entrada e produzir atributos de saída. Por exemplo, você pode definir uma operação de adição ou multiplicação.

Computação com TensorFlow

Este primeiro exemplo de uma versão TensorFlow . Ele mostra como você pode criar uma sessão, definir constantes e executar o cálculo com essas constantes usando a sessão.

```
1 import tensorflow as tf
2 sess = tf.Session()
3 a = tf.constant(10)
4 b = tf.constant(32)
5 print(sess.run(a+b))
```

A execução deste exemplo exibe 42

#### IV. USANDO O TEMPLATE

Esta breve introdução usa o TensorFlow e Keras para: Construir uma rede neural que classifique imagens. Treine essa rede neural. E, finalmente, avalie a precisão do modelo.

A. Baixe e instale o pacote TensorFlow 2. Importe o TensorFlow para o seu programa:

```
from __future__ import absolute_import, division, print_function, unicode_literals

# Install TensorFlow

import tensorflow as tf
```

B. Carregue e prepare o conjunto de dados MNIST . Converta as amostras de números inteiros em números de ponto flutuante:

```
mnist = tf.keras.datasets.mnist

(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0
```

C. Crie o `tf.keras.Sequential` modelo empilhando camadas. Escolha uma função otimizadora e de perda para treinamento:

```
model = tf.keras.models.Sequential([
    tf.keras.layers.Flatten(input_shape=(28, 28)),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(10, activation='softmax')
])

model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])
```

D. Treine e avalie o modelo:

```
model.fit(x_train, y_train, epochs=5)

model.evaluate(x_test, y_test, verbose=2)
```

#### REFERENCIAS

- [1] <https://www.nltk.org/index.html> - © Copyright 2019, NLTK Project. Last updated on Aug 20, 2019. Created using Sphinx 2.1.2
- [2] <https://pytorch.org/>.
- [3] <https://scikit-learn.org/stable/> - © 2007-2019, scikit-learn developers (Licença BSD).
- [4] Scikit-learn: Aprendizado de Máquina em Python <<http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html>>.
- [5] <https://keras.io>.
- [6] <http://deeplearning.net/software/theano/index.html> - © Copyright 2008--2017, laboratório LISA. Última atualização em 21 de novembro de 2017.
- [7] [https://www.sas.com/pt\\_br/insights/analytics/inteligencia-artificial.html](https://www.sas.com/pt_br/insights/analytics/inteligencia-artificial.html) - Declaração de Privacidade | Termos de Uso | © 2019 SAS Institute Inc. All Rights Reserved.
- [8] <https://www.python.org/> - Copyright ©2001-2019. Python Software Foundation Legal Statements Privacy Policy Powered by Heroku.