



# Capacitação em Inteligência Artificial e Aplicações

---

## Frameworks para Deep Learning

- Prof. Gerson Vieira Albuquerque Neto
- Prof. Rodrigo Carvalho Souza Costa
- Prof. Yves Augusto Romero



UNIVERSIDADE  
ESTADUAL DO CEARÁ



MINISTÉRIO DA  
CIÊNCIA, TECNOLOGIA  
E INOVAÇÃO





# IA

## Planejamento da Disciplina

D	S	T	Q	Q	S	S
26	27 Introdução ao curso	28 Áreas e aplicações de IA	29 Tipos e definições de Inteligência artificial	30 Revisão de álgebra e probabilidade	31 Laboratório Python 1	1
2	3 Introdução aos classificadores supervisionados	4 Aula teórica Naive Bayes	5 Aula prática Naive Bayes	6 Feriado Semana Santa	7 Feriado Semana Santa	8
9	10 KNN + Métricas de Avaliação	11 Regressão Linear e Introdução à árvores de decisão	12 Prática Regressão Linear + Árvores de Decisão	13 Feriado	14 Introdução à Clusterização + KMédias	15
16	17 Falta de Energia Campus Fortaleza	18 PCA / Hiperparâmetros	19 Introdução ao Perceptron Simples – Prática	20 MLP	21 Feriado Tiradentes	28
23	24 Introdução ao DeepLearning	25 Introdução ao TensorFlow / Keras	26 Introdução ao Pytorch	27 Tensorflow for android	28	29



- Após a conclusão deste módulo, você será capaz de:
  - Descreva uma estrutura de aprendizado profundo.
  - Conhecer as principais estruturas de aprendizado profundo.
  - Conhecer as características do PyTorch.
  - Conhecer os recursos do TensorFlow.
  - Diferenciar entre TensorFlow 1.x e 2.x.

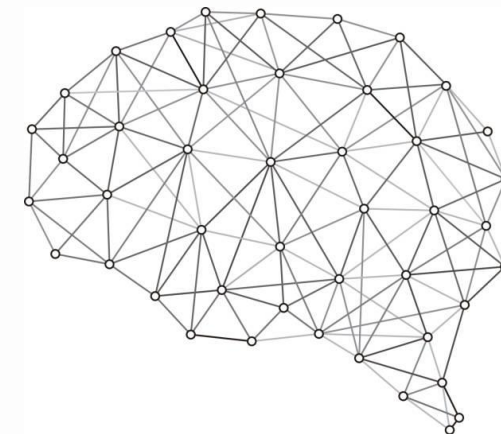






# Introdução às Redes Neurais: Deep Learning

- Frameworks
- Introdução ao TensorFlow



UNIVERSIDADE  
ESTADUAL DO CEARÁ



MINISTÉRIO DA  
CIÊNCIA, TECNOLOGIA  
E INOVAÇÃO





# IA

## Frameworks Deep Learning

- Uma estrutura de aprendizagem profunda é uma interface, biblioteca ou uma ferramenta que nos permite construir modelos de aprendizagem profunda de forma mais fácil e rápida, sem entrar nos detalhes dos algoritmos subjacentes.
- Uma estrutura de aprendizagem profunda pode ser considerada como um conjunto de blocos de construção. Cada componente nos blocos de construção é um modelo ou algoritmo. Portanto, os desenvolvedores podem usar componentes para montar modelos que atendam aos requisitos e não precisem começar do zero.

theano

Caffe

dmlc  
**mxnet**

**DL4J** Deep Learning for Java

**PYTORCH**

 torch  
A SCIENTIFIC COMPUTING FRAMEWORK FOR LUAJIT

  
TensorFlow

**[M]<sup>s</sup>**  
MindSpore



# IA

## Frameworks Deep Learning

- O surgimento de estruturas de aprendizado profundo reduz os requisitos para os desenvolvedores.
- Os desenvolvedores não precisam mais compilar código a partir de redes neurais complexas e algoritmos de retropropagação.
- Em vez disso, eles podem usar modelos existentes para configurar parâmetros conforme necessário, onde os parâmetros do modelo são treinados automaticamente.
- Além disso, eles podem adicionar camadas de rede autodefinidas aos modelos existentes ou selecionar classificadores e algoritmos de otimização necessários diretamente invocando o código existente.

theano

Caffe

dmlc  
**mxnet**

**DL4J** Deep Learning for Java

**PYTORCH**

 torch  
A SCIENTIFIC COMPUTING FRAMEWORK FOR LUAJIT

  
TensorFlow

**[M]<sup>S</sup>**  
MindSpore





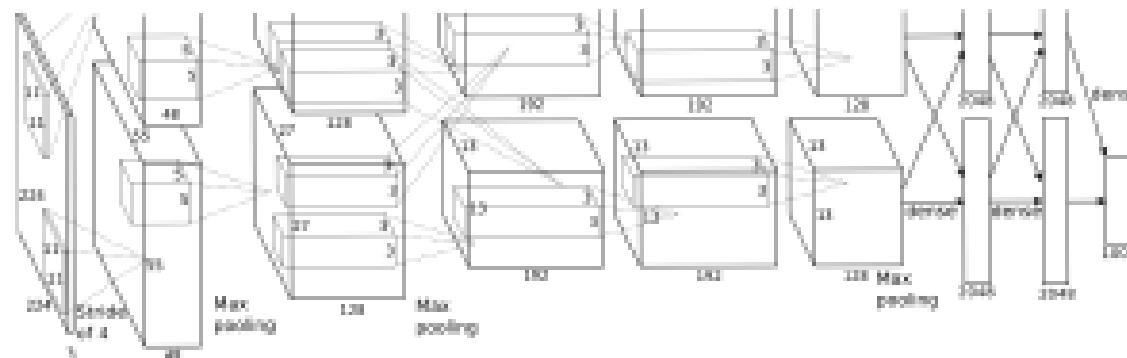
- Escrito em C++ (CUDA), tem interfaces de linha de comando, Python, MATLAB
  - Biblioteca rápida e bem testada para aprendizado profundo
  - Alternância perfeita entre CPU e GPU
- Estrutura aberta, modelos e exemplos trabalhados para aprendizado profundo
  - definições de modelo
  - configurações de otimização
  - pesos pré-treinados para que você possa começar imediatamente.
- Os modelos BVLC são licenciados para uso irrestrito.



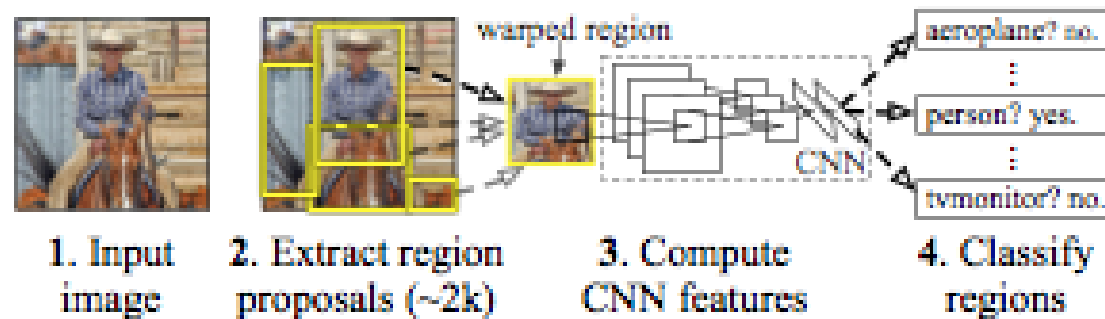
# IA

## Exemplos de Modelos CAFFE

### AlexNet: ImageNet Classification



### R-CNN: Regions with CNN features







# IA

# Pytorch

# PYTORCH

- O PyTorch é uma estrutura de computação de aprendizado de máquina baseada em Python desenvolvida pelo Facebook.
  - Ele é desenvolvido com base no Torch, uma estrutura de computação científica apoiada por um grande número de algoritmos de aprendizado de máquina.
  - Torch é uma biblioteca de operação tensorial semelhante ao NumPy, caracterizada por alta flexibilidade, mas é menos popular porque usa a linguagem de programação Lua. É por isso que o PyTorch é desenvolvido.
- Além do Facebook, institutos como o Twitter, a GMU também usam o PyTorch.



# IA

# Pytorch

# PYTORCH

- **Python primeiro:**
  - O PyTorch não simplesmente vincula o Python a uma estrutura C++.
  - O PyTorch suporta diretamente o acesso Python em um grão fino.
  - Os desenvolvedores podem usar o PyTorch tão facilmente quanto usar o NumPy ou o SciPy.
  - Isso não apenas reduz o limite para a compreensão do Python, mas também garante que o código seja basicamente consistente com a implementação nativa do Python.
- **Rede neural dinâmica:**
  - Muitas estruturas principais, como o TensorFlow 1.x, não oferecem suporte a esse recurso.
  - Para executar o TensorFlow 1.x, os desenvolvedores devem criar gráficos computacionais estáticos com antecedência e executar o feed e executar comandos para executar repetidamente os gráficos criados.
  - Por outro lado, o PyTorch com esse recurso está livre de tal complexidade, e os programas PyTorch podem construir / ajustar dinamicamente gráficos computacionais durante a execução.
- **Fácil de depurar:**
  - O PyTorch pode gerar gráficos dinâmicos durante a execução. Os desenvolvedores podem parar um interpretador em um depurador e exibir a saída de um nó específico.
- O PyTorch fornece tensores que suportam CPUs e GPUs, acelerando muito a computação



# IA

# Tensorflow



- O TensorFlow é uma biblioteca de software de código aberto de segunda geração do Google para computação digital.
- A estrutura de computação TensorFlow suporta vários algoritmos de aprendizagem profunda e várias plataformas de computação, garantindo alta estabilidade do sistema.

## Características

Escalabilidade

Multilingue

GPU

Multi-plataforma

Computação  
poderosa

Distribuído





- O TensorFlow pode ser executado em computadores diferentes:
  - De smartphones a clusters de computadores, para gerar os modelos de treinamento desejados.
- Atualmente, as estruturas de aprendizado profundo distribuídas nativas suportadas incluem apenas TensorFlow, CNTK, Deeplearning4J e MXNet.
- Quando uma única GPU é usada, a maioria das estruturas de aprendizado profundo depende do cuDNN (CUDA) e, portanto, suporta quase a mesma velocidade de treinamento, desde que os recursos de computação de hardware ou as memórias alocadas difiram um pouco. No entanto, para o aprendizado profundo em larga escala, os dados massivos dificultam que a GPU única conclua o treinamento em um tempo limitado. Para lidar com esses casos, o TensorFlow permite o treinamento distribuído.





# IA

# Keras



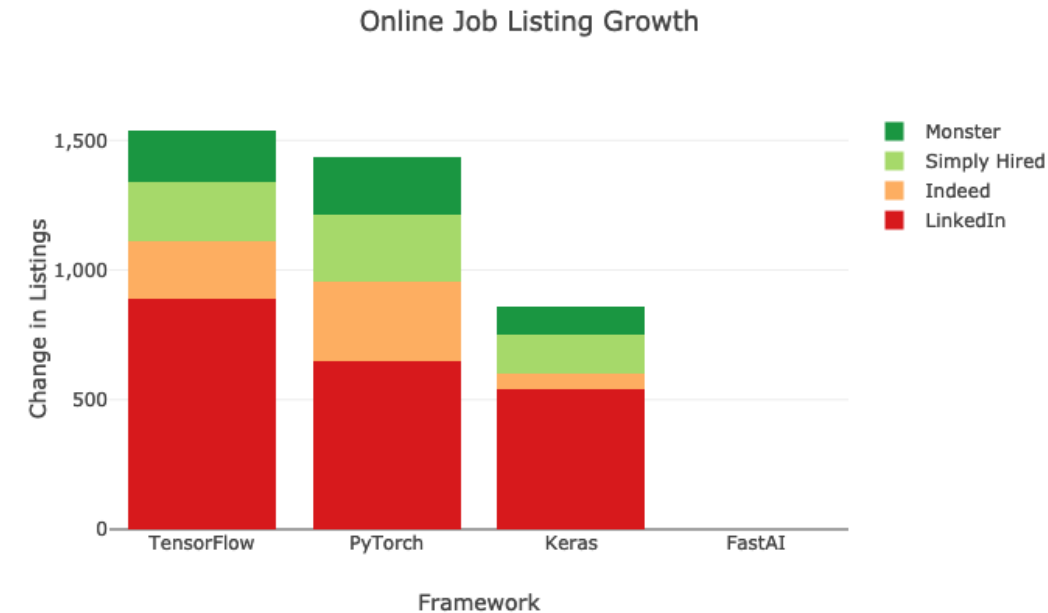
- Keras é uma biblioteca de alto nível em Python que suporta algumas ferramentas de análise de dados (por exemplo, rede neural)
- Ele é construído em uma biblioteca de baixo nível bem estabelecida (por exemplo, TensorFlow e Theano)
- Keras é a primeira biblioteca de alto nível adicionada ao núcleo do TensorFlow no Google.
- Começou a ser incorporada em 2017.
- Muitas pessoas industriais usam Keras (em cima do TensorFlow).



# IA

## Porque o tensorflow?

- O TensorFlow é considerado uma das melhores bibliotecas para redes neurais e pode reduzir a dificuldade no desenvolvimento de aprendizado profundo. Além disso, como é de código aberto, pode ser convenientemente mantido e atualizado, assim, a eficiência do desenvolvimento pode ser melhorada.
- O Keras, que ocupa o terceiro lugar no número de estrelas no GitHub, é empacotado em uma API avançada do TensorFlow 2.0, o que torna o TensorFlow 2.x mais flexível e fácil de depurar.



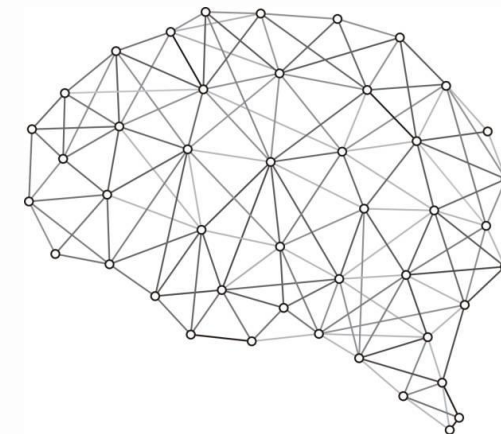
Demand on the  
recruitment market



# Introdução às Redes Neurais: Deep Learning

---

- Frameworks
- **Introdução ao tensorflow**



UNIVERSIDADE  
ESTADUAL DO CEARÁ



MINISTÉRIO DA  
CIÊNCIA, TECNOLOGIA  
E INOVAÇÃO





# IA

## Tensores

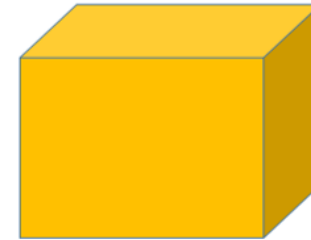
- Os tensores são as estruturas de dados mais básicas no TensorFlow.
- Todos os dados são encapsulados em tensores.
- Tensor: uma matriz multidimensional
  - Um escalar é um tensor rank-0. Um vetor é um tensor rank-1. Uma matriz é um tensor rank-2.
- No TensorFlow, os tensores são classificados em:
  - Tensores constantes
  - Tensores variáveis



Tensor unidimensional



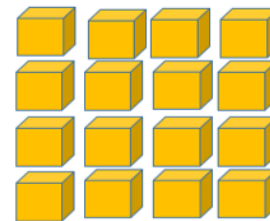
Tensor bidimensional



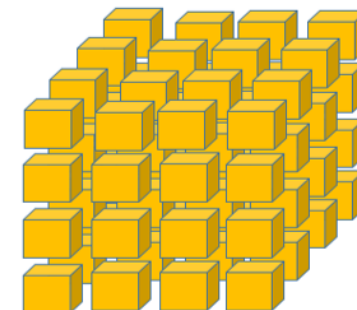
Tensor tridimensional



Tensor quadri-dimensional



Tensor de cinco dimensões



Tensor de seis dimensões





- Desvantagens do TensorFlow 1.0:
  - Depois que um tensor é criado no TensorFlow 1.0, o resultado não pode ser retornado diretamente. Para obter o resultado, o mecanismo de sessão precisa ser criado, o que inclui o conceito de gráfico, e o código não pode ser executado sem invocar o `session.run`. Este estilo é mais parecido com a linguagem de programação de hardware VHDL.
  - Em comparação com algumas estruturas simples, como o PyTorch, o TensorFlow 1.0 adiciona os conceitos de sessão e gráfico, que são inconvenientes para os usuários.
  - É complexo depurar o TensorFlow 1.0 e suas APIs são desordenadas, dificultando a vida de iniciantes. Os alunos encontrarão muitas dificuldades em usar o TensorFlow 1.0, mesmo depois de obter o conhecimento básico. Como resultado, muitos pesquisadores se voltaram para o PyTorch.



- Características do TensorFlow 2.x:
  - Keras de API avançada:
  - Fácil de usar: Os mecanismos de gráfico e sessão são removidos. O que você vê é o que você recebe, assim como Python e PyTorch.
  - Principais melhorias:
  - A função principal do TensorFlow 2.x é o mecanismo de gráfico dinâmico chamado execução ansiosa. Ele permite que os usuários compilem e depurem modelos como programas normais, tornando o TensorFlow mais fácil de aprender e usar.
  - Várias plataformas e linguagens são suportadas, e a compatibilidade entre os componentes pode ser melhorada por meio da padronização de formatos de troca e alinhamento de APIs.
  - APIs preteridas são excluídas e APIs duplicadas são reduzidas para evitar confusão.
  - Compatibilidade e continuidade: O TensorFlow 2.x fornece um módulo que permite a compatibilidade com o TensorFlow 1.x.
  - O módulo `tf.contrib` é removido. Os módulos mantidos são movidos para repositórios separados. Módulos não utilizados e não mantidos são removidos.



# IA

## Operações básicas do TensorFlow 2.x

- A seguir descreve APIs comuns no TensorFlow, concentrando-se no código. O conteúdo principal é o seguinte:
  - Métodos para criar constantes e variáveis
  - Fatiamiento e indexação de tensores
  - Mudanças de dimensão dos tensores
  - Operações aritméticas em tensores
  - Concatenação e divisão de tensores
  - Classificação tensorial



# IA

## Modo de execução ansioso (EAGER) do TensorFlow 2.x

- **Gráfico estático:** O TensorFlow 1.x usando gráficos estáticos (modo gráfico) separa a definição e a execução da computação usando gráficos computacionais. Este é um modelo de programação declarativa. No modo gráfico, os desenvolvedores precisam criar um gráfico computacional, iniciar uma sessão e, em seguida, inserir dados para obter um resultado de execução.
- Os gráficos estáticos são vantajosos no treinamento distribuído, otimização de desempenho e implantação, mas inconvenientes para depuração. A execução de um gráfico estático é semelhante a invocar um programa de linguagem C compilado, e a depuração interna não pode ser executada nesse caso. Portanto, surge uma execução ansiosa baseada em gráficos computacionais dinâmicos.
- A execução ansiosa é um método de programação baseado em comandos, que é o mesmo que o Python nativo. Um resultado é retornado imediatamente após a execução de uma operação.





- A execução ansiosa está habilitada no TensorFlow 2.x por padrão. A execução ávida é intuitiva e flexível para os usuários (mais fácil e rápida de executar uma operação única), mas pode comprometer o desempenho e a capacidade de implantação.
- Para obter o desempenho ideal e tornar um modelo implantável em qualquer lugar, você pode executar `@tf.function` para adicionar um decorador para criar um gráfico a partir de um programa, tornando o código Python mais eficiente.
- `tf.function` pode construir uma operação TensorFlow na função em um gráfico. Desta forma, esta função pode ser executada no modo gráfico. Tal prática pode ser considerada como encapsulando a função como uma operação TensorFlow de um gráfico.



- **tf:** Funções no módulo tf são usadas para executar operações aritméticas comuns, como tf.abs (calculando um valor absoluto), tf.add (adicionando elementos um por um) e tf.concat (concatenando tensores). A maioria das operações neste módulo pode ser executada pelo NumPy.
- **tf.errors:** módulo de tipo de erro do TensorFlow
- **tf.data:** implementa operações em conjuntos de dados.
  - Os fluxos de entrada criados por tf.data são usados para ler dados de treinamento. Além disso, os dados podem ser facilmente inseridos a partir de memórias (como o NumPy).
- **tf.distributions:** implementa várias distribuições estatísticas.
  - As funções neste módulo são usadas para implementar várias distribuições estatísticas, como a distribuição de Bernoulli, a distribuição uniforme e a distribuição gaussiana.



- **tf.io.gfile:** implements operations on files.
  - Functions in this module can be used to perform file I/O operations, copy files, and rename files.
- **tf.image:** implements operations on images.
  - Functions in this module include image processing functions. This module is similar to OpenCV, and provides functions related to image luminance, saturation, phase inversion, cropping, resizing, image format conversion (RGB to HSV, YUV, YIQ, or gray), rotation, and sobel edge detection. This module is equivalent to a small image processing package of OpenCV.
- **tf.keras:** a Python API for invoking Keras tools.
  - This is a large module that enables various network operations.



- O TensorFlow 2.x recomenda o Keras para a criação de redes. Redes neurais comuns estão incluídas em Keras.layers.
- O Keras é uma API de alto nível usada para criar e treinar modelos de aprendizado profundo. Ele pode ser usado para design rápido de protótipos, pesquisa avançada e produção.
- Tem as seguintes três vantagens:
  - **Fácil de usar:** O Keras fornece GUIs simples e consistentes otimizadas para casos comuns. Ele fornece feedback prático e claro sobre os erros do usuário.
  - **Modular e componível:** Você pode criar modelos Keras conectando blocos de construção configuráveis juntos, com pouca restrição.
  - **Fácil de estender:** Você pode personalizar blocos de construção para expressar novas ideias de pesquisa, criar camadas e funções de perda e desenvolver modelos avançados.





# IA

## Métodos e interfaces Keras comuns

- A seguir, descrevem-se métodos e interfaces comuns do tf.keras concentrando-se no código.
- O conteúdo principal é o seguinte:
  - Processamento de conjuntos de dados: conjuntos de dados e pré-processamento
  - Criação de modelo de rede neural: Sequencial, Modelo, Camadas...
  - Compilação de rede: compilação, perdas, métricas e otimizadores
  - Treinamento e avaliação em rede: ajuste, fit\_generator e avalie



# Introdução às Redes Neurais: Perceptron Simples

- **Prática DeepLearning**



UNIVERSIDADE  
ESTADUAL DO CEARÁ



MINISTÉRIO DA  
CIÊNCIA, TECNOLOGIA  
E INOVAÇÃO



# Dúvidas?

Módulo de Inteligência Artificial



UNIVERSIDADE  
ESTADUAL DO CEARÁ



Instituto Iracema  
PESQUISA E INOVAÇÃO



Softex

MINISTÉRIO DA  
CIÊNCIA, TECNOLOGIA  
E INOVAÇÃO

GOVERNO FEDERAL  
**BRASIL**  
UNIÃO E RECONSTRUÇÃO