Microsoft Azure

As Loucas Aventuras no Mundo da Inteligência Artificial

George Luiz Bittencourt

Sobre mim

SCAN ME

George Luiz Bittencourt

Arquiteto de soluções cloud com mais de 20 anos de experiência em infraestrutura e desenvolvimento de software.

/glzbcrt

/glzbcrt

george.bittencourt@microsoft.com

Porque essa apresentação?

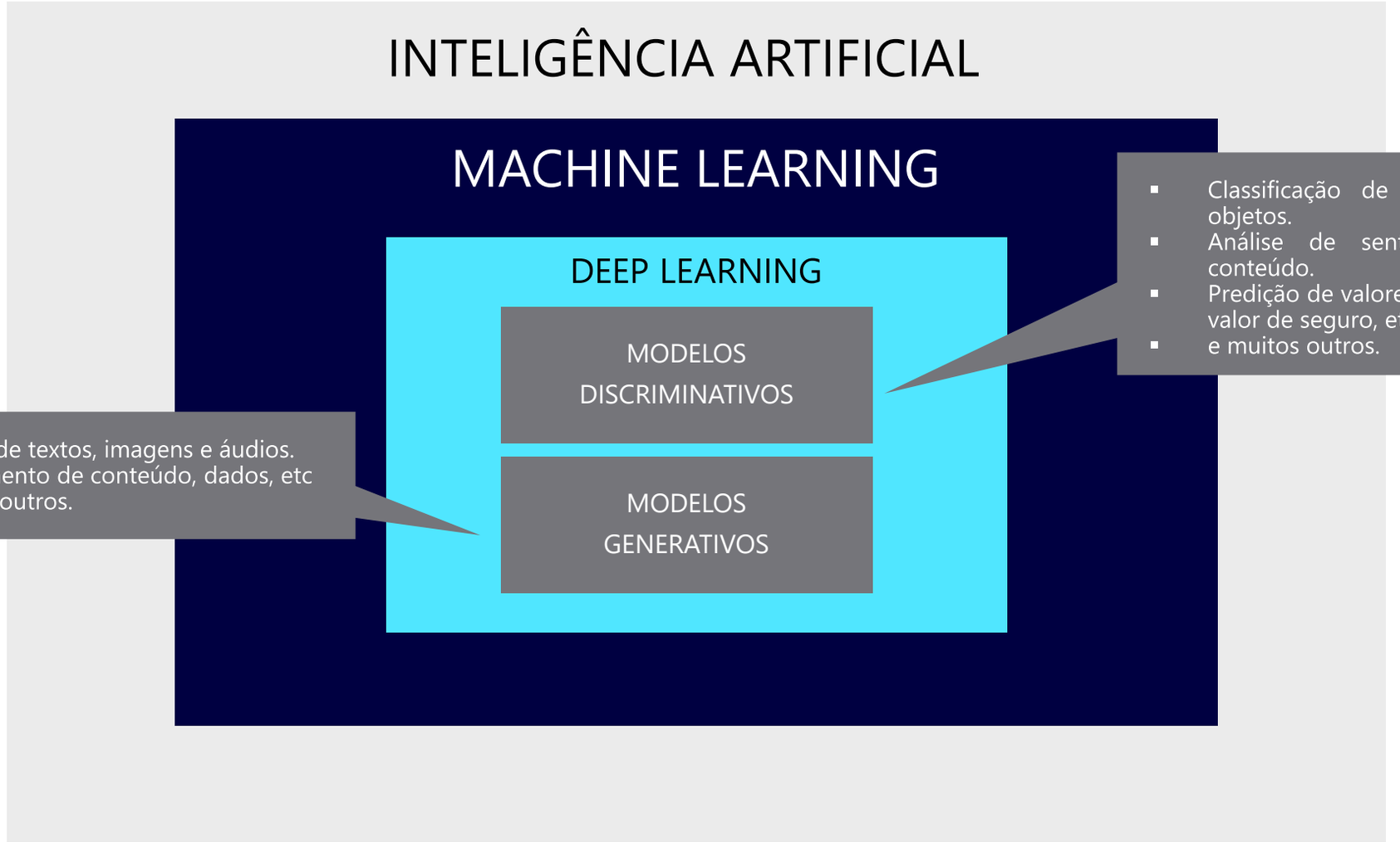
Estamos na era da AI

AI parece complexa, mas não é

Uma boa base sempre nos ajuda

Contar um pouco da minha história

O que é Inteligência Artificial?



O que é um modelo?

- É uma **fórmula matemática** com **múltiplas variáveis** de entrada e saída **otimizada** a partir de **exemplos anotados**.

$$Y = in^0 w^0 + in^1 w^1 + in^2 w^2 + in^n w^n + \beta$$

- Um modelo é composto de **parâmetros**, que são **números** encontrados durante o processo de **treinamento** que tem por objetivo **reduzir o erro** entre o **valor calculado** e o **valor anotado**.
- **Os valores de entrada e saída precisam ser numéricos**, já que são utilizados em milhares de cálculos matemáticos. Etapas de pré-processamento e pós-processamento fazem as conversões necessárias.
- Durante a execução milhões de cálculos são executados. Esses cálculos podem ser executados tanto na CPU quanto utilizando aceleradores como as GPUs da NVIDIA ou ainda TPUs. A unidade FLOPS é muito importante, pois mede quantos cálculos é possível fazer por segundo. Quanto maior, menor será a latência da inferência.

Do que é feito um modelo?

Camada de entrada

```
return tf.keras.Sequential([
    tf.keras.layers.Rescaling(
        1./255, input_shape=(img_height, img_width, img_channels)),
    tf.keras.layers.Conv2D(16, 3, padding='same', activation='relu'),
    tf.keras.layers.MaxPooling2D(),
    tf.keras.layers.Conv2D(64, 3, padding='same', activation='relu'),
    tf.keras.layers.MaxPooling2D(),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dense(len(output_classes), name="output")
])
```

Camada de saída

Layers

- *Cada layer parâmetros, aprendidos treinamento.*
- *Algumas layers não possuem parâmetros, com a Dropout e Flatten.*

$$Y = in^0w^0 + in^1w^1 + in^2w^2 + in^nw^n + \beta$$

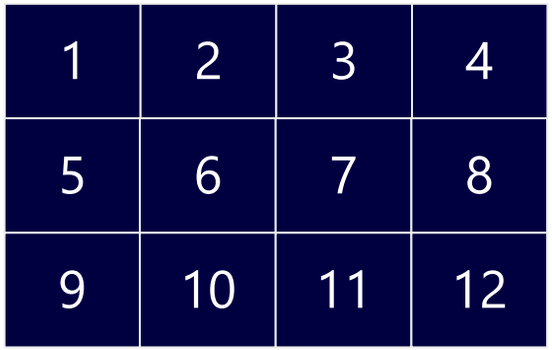
Outra forma de escrever o modelo:

```
output = Dense(Dense(Dropout(Flatten(MaxPooling2D(Conv2D(MaxPooling2D(Conv2D(MaxPooling2D(Conv2D(Rescaling(input)))))))))))
```

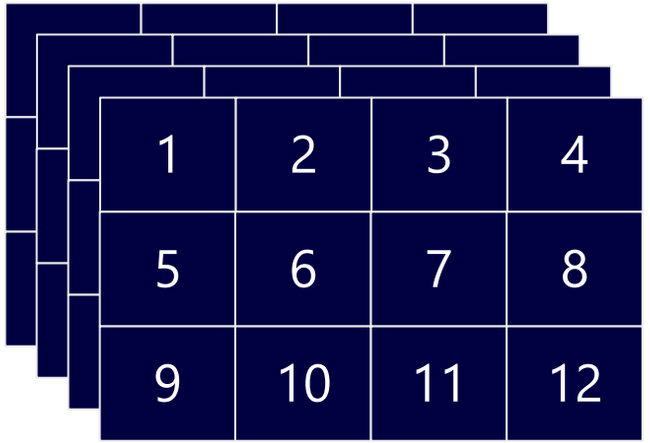
Como o dado é representado?

escalar

vetor



matriz

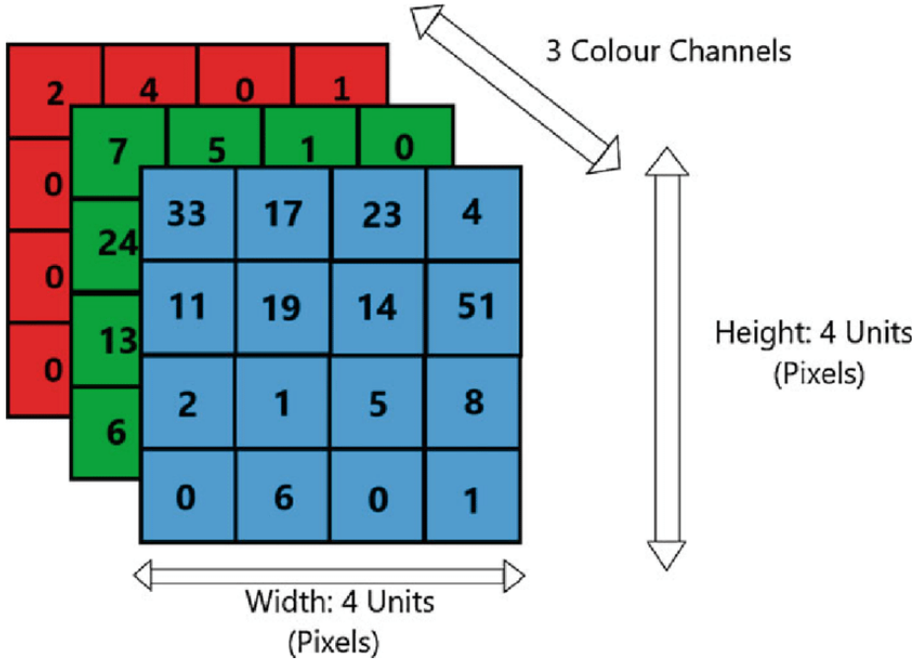


cubo

TENSOR

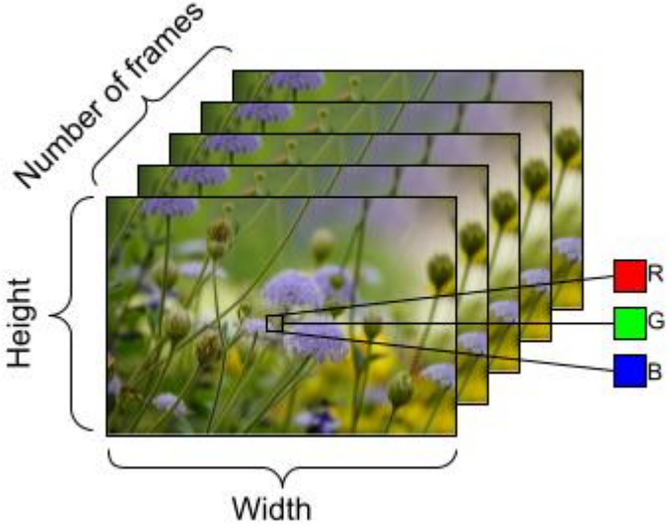
- Não existe limite na quantidade de dimensões, porém o cérebro humano lida com até 3 dimensões facilmente.
- Um tensor tem um formato, que é a quantidade de dimensões e elementos por dimensão.
- Um local. Ele pode estar armazenado na memória RAM ou ainda na GPU/TPU.
- + um tipo de dado que define a faixa de valores e quantidade de memória necessária. Os mais comuns:
  - FP32
  - FP16
  - BF16
  - INT8

Algumas representações



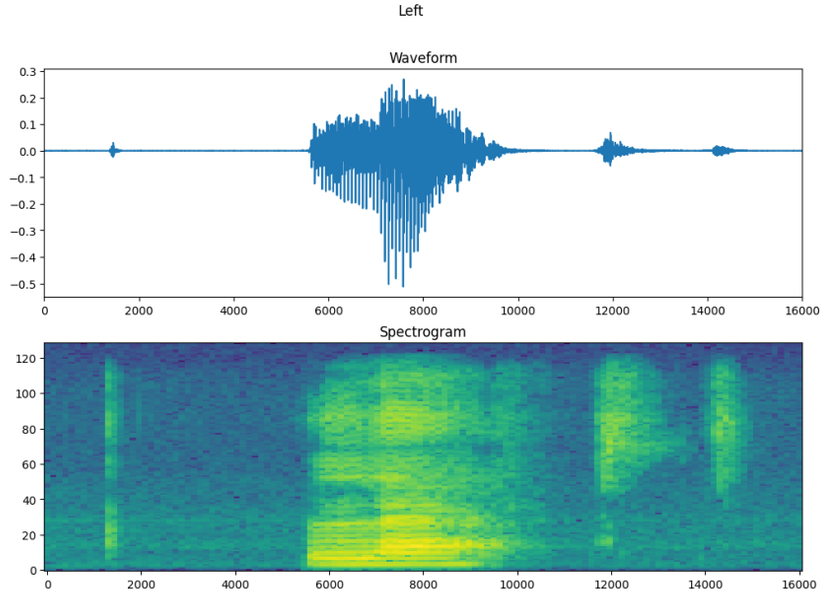
Fonte: [2 A 3D tensor of a Red-Green-Blue (RGB) image of a dimension of 4 Â 4 Â 3 | Download Scientific Diagram (researchgate.net)]

Acesso em: 11/03/2024



Fonte: [Using TensorFlow for Deep Learning on Video Data — The TensorFlow Blog]

Acesso em: 11/03/2024



Fonte: [Simple audio recognition: Recognizing keywords | TensorFlow Core]

Acesso em: 11/03/2024.

Como o modelo aprende?

- Da mesma maneira que nós! Ao sermos expostos a vários exemplos corretos e incorretos conseguimos criar regras para entender.
- No caso do computador ele calcula para cada exemplo o quão errado ele estava e ajusta seus parâmetros para na próxima iteração reduzir o erro.
- Existem várias formas de calcular o erro, sendo o MSE uma das mais comuns.

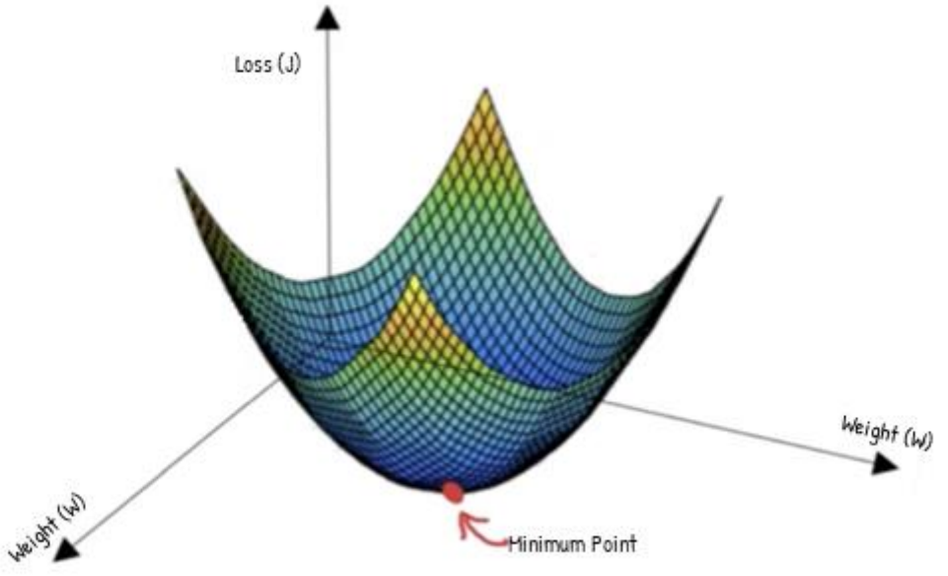
$$Y = in^0w^0 + in^1w^1 + in^2w^2 + in^nw^n + \beta$$

Exemplo	Valor Esperado	Valor Calculado #1	Valor Calculado #2	Valor Calculado #N
1	5	2	4	5.3
2	6	15	8	6.7
3	2	2.3	2.7	2.1
4	1	7	3	0.6
5	7	2	9	6.6

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

Como o modelo aprende?

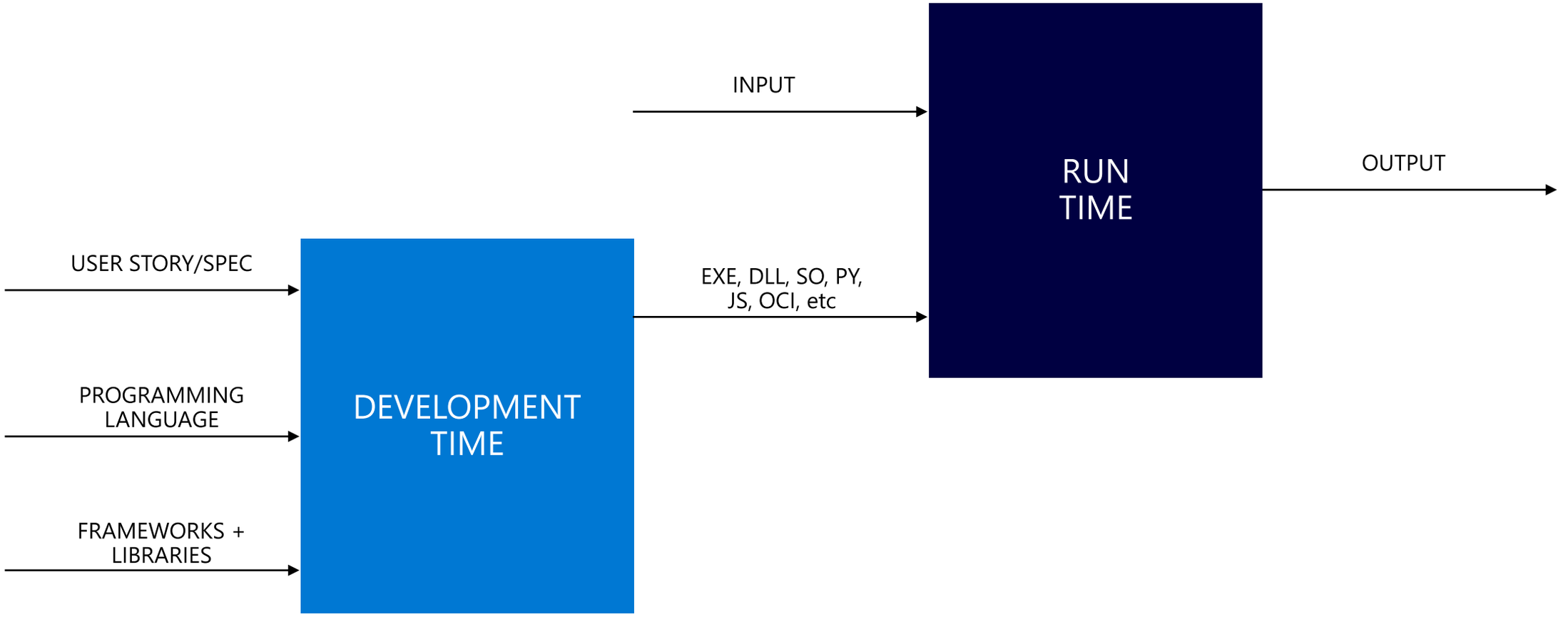
- Durante o treinamento do modelo o **otimizador** em conjunto com o **algoritmo de erro** busca os **melhores valores**. Somente os valores iniciais são aleatórios.
- Depois da primeira iteração utilizando matemática diferencial e calculando os gradientes eles são aplicados aos parâmetros gerando novos valores e tudo recomeça. Esse passo é conhecido como *backpropagation* e é um dos princípios mais importantes em redes neurais.
- O objetivo é buscar o conjunto de parâmetros que reduz ao menor valor possível o erro **dentro do tempo alocado** para o treinamento.
- Existem **várias** respostas para o mesmo *dataset*.



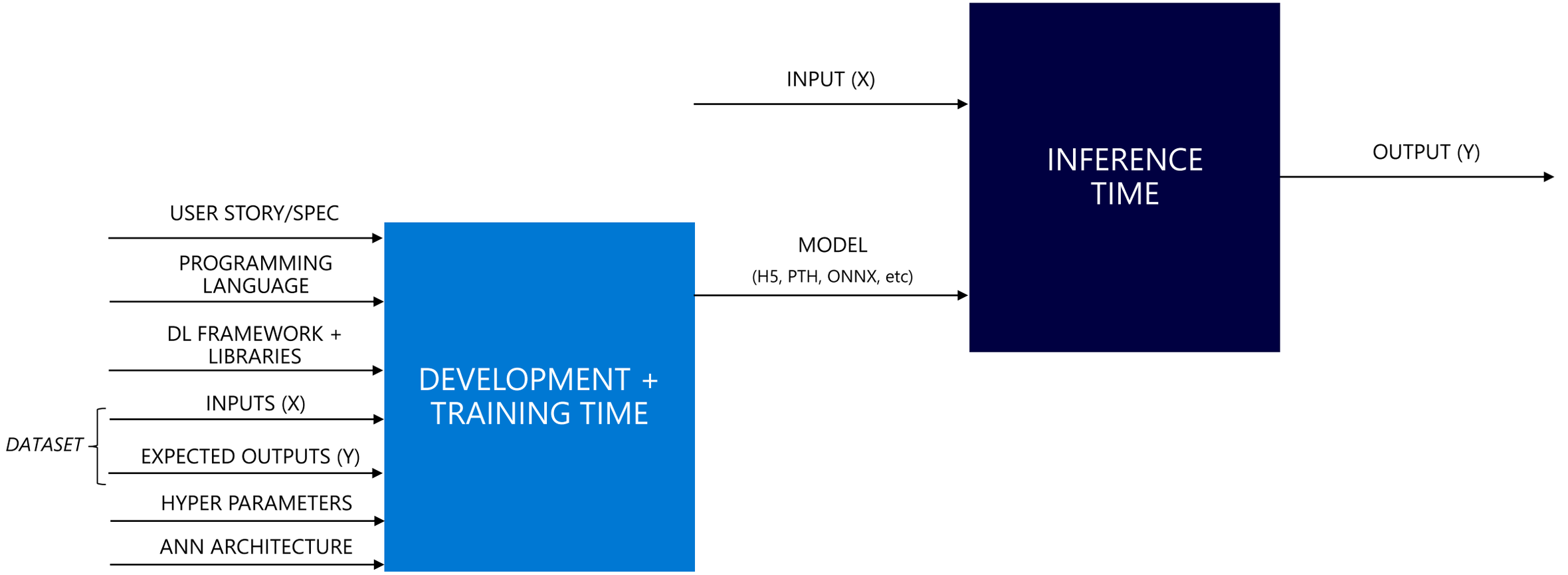
Fonte: Gradient Descent vs. Backpropagation: What's the Difference? (analyticsvidhya.com)

Acesso em: 11/03/2024

Como criamos um programa?



Como criamos um modelo?

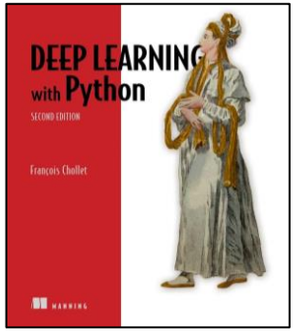


O que queremos com um modelo?

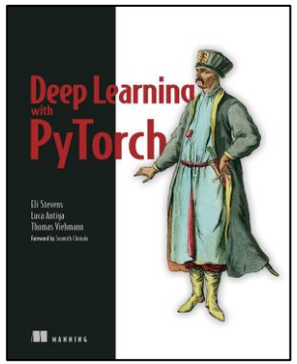
- Que ele atenda o caso de uso para o qual foi criado.
- Que o ROI seja positivo, ou seja, o custo de desenvolvimento, hospedagem e manutenção deve ser menor que o valor gerado pelo uso do modelo.
- e principalmente:

Que o modelo generalize o conhecimento!

Referências



Deep Learning with Python, Second Edition 2nd Edition
by Francois Chollet (Author)



Deep Learning with PyTorch: Build, train, and tune neural networks using Python tools First Edition
by Eli Stevens (Author), Luca Antiga (Author), Thomas Viehmann (Author)

TensorFlow Developer Professional Certificate (DeepLearning.AI) | Coursera

Referências

Neural networks by 3Blue1Brown

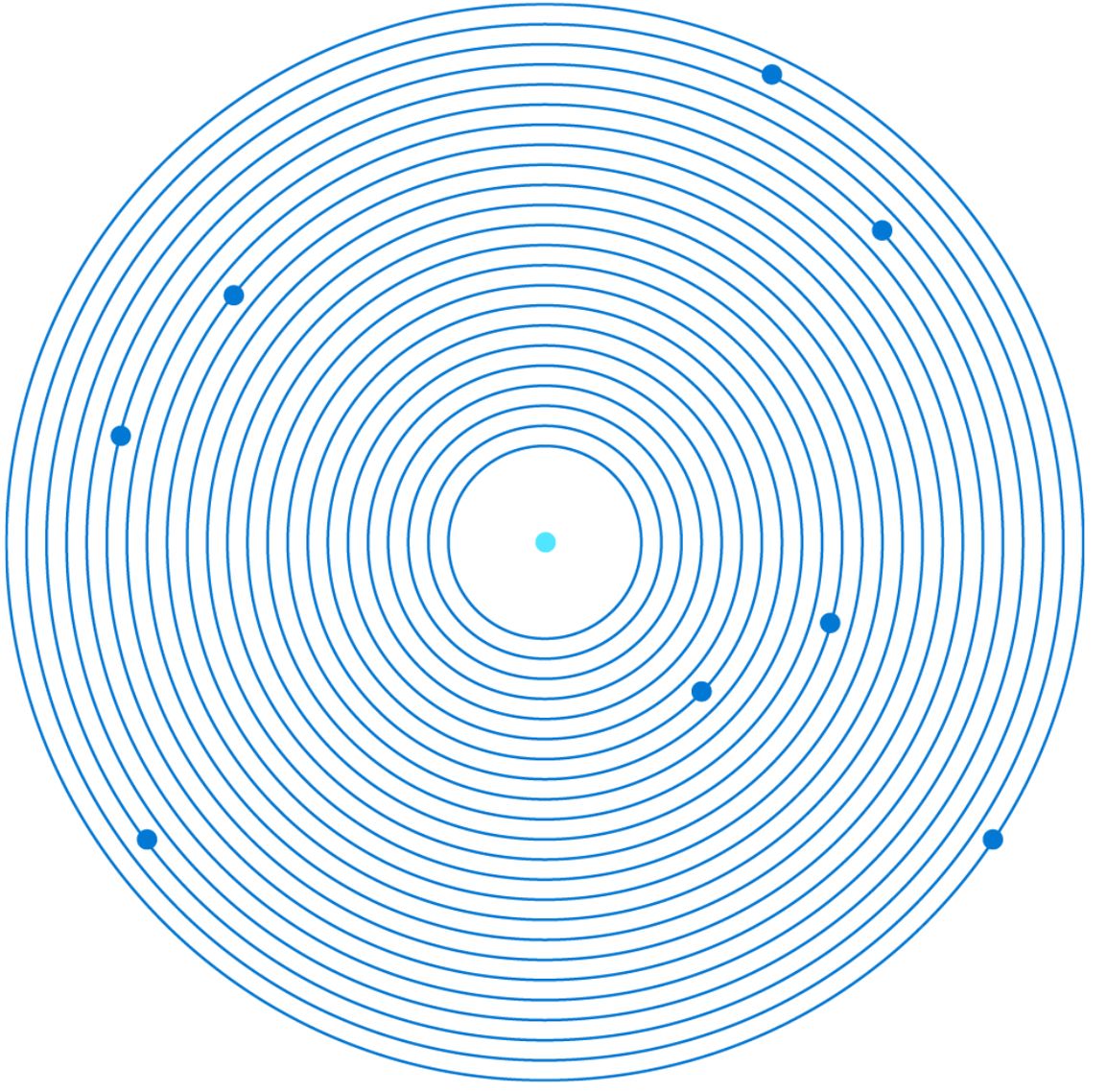
Valerio Velardo - The Sound of AI - YouTube

glzbcrt/tf-onnx-sample: Sample Tensorflow model to classify images and infer using ONNX. (github.com)

glzbcrt/cuda-python: Simple Python script that using the ctypes library will call a function inside a CUDA DLL file. (github.com)

Microsoft Azure

DEMO



Microsoft Azure

EOP

/glzbcrt

/glzbcrt

george.bittencourt@microsoft.com

SCAN ME