



**Data Science
Academy**

www.datascienceacademy.com.br

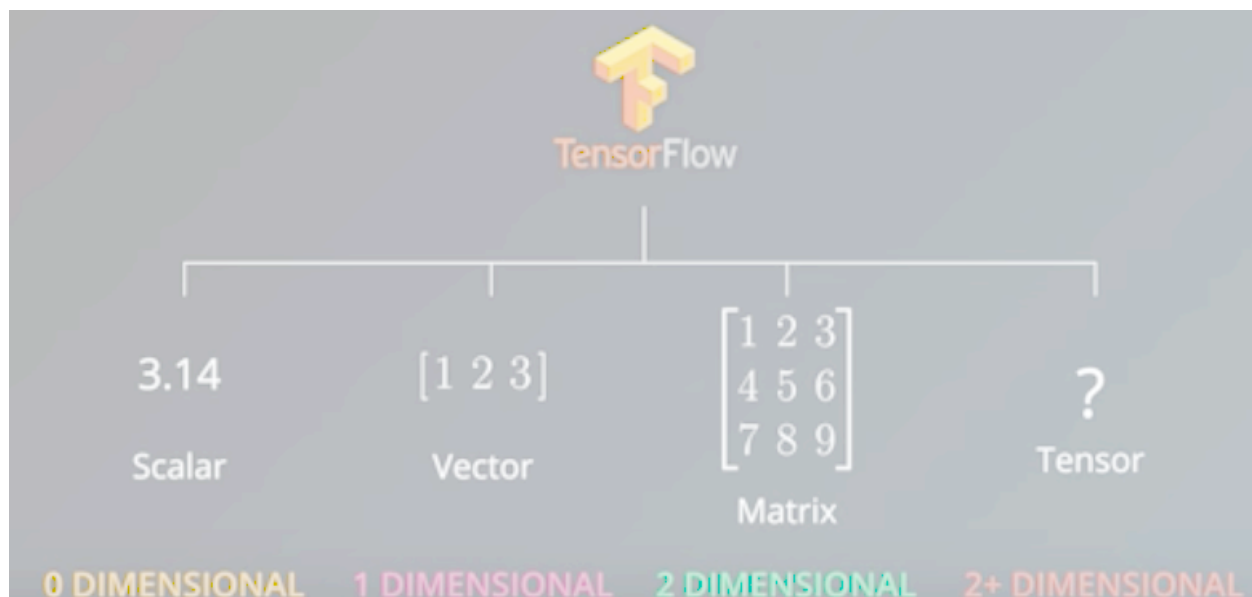
Deep Learning I

**Operações com Matrizes
Parte 1**

Deep Learning envolve muita matemática e operações com matrizes, sendo importante você entender o básico antes de mergulhar na construção de suas próprias redes neurais. Este bonus oferece um passo a passo de como realizar essas operações matemáticas e utilizar a biblioteca Python de computação científica, o Numpy, base de quase todos os frameworks de Deep Learning.

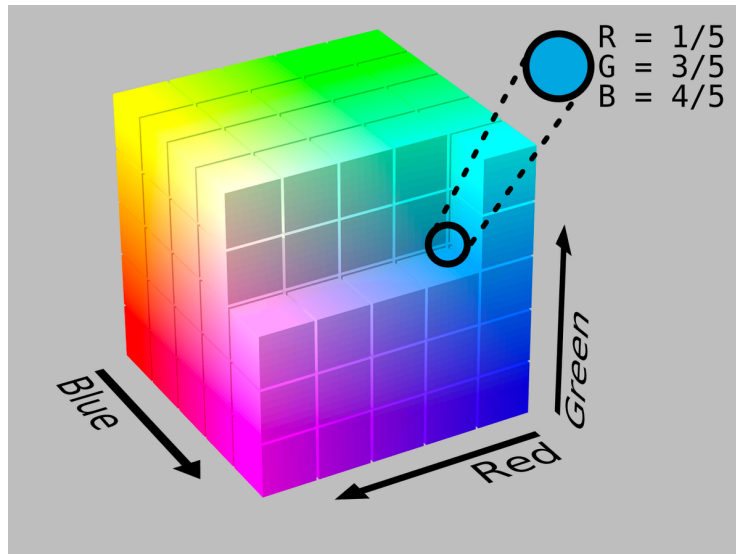
Como você aprendeu até aqui na Formação IA, as redes neurais esperam receber os dados em um formato específico e o tratamento do shape (formato) dos dados de input é sua responsabilidade. Os dados podem assumir os mais variados formatos e conhecer o que você vai processar com as redes neurais é de extrema importância.

Os dados que vão alimentar a rede neural, podem assumir basicamente 4 formatos: Scalar, Vetor, Matriz e Tensor. O número de dimensões que serão representados, definem qual desses formatos você deverá utilizar! Pode ser necessário combinar esses formatos, para representar o shape dos dados que você quer processar e isso é simplesmente uma operação matemática.



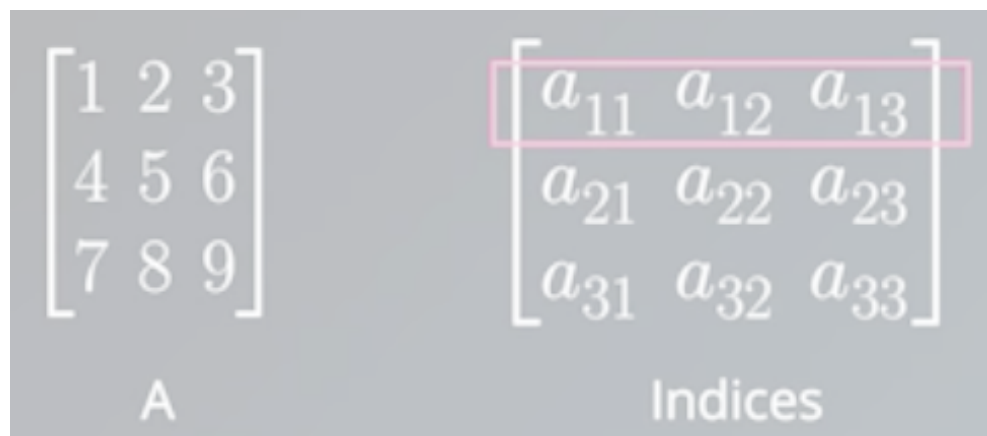
Dados com alta dimensionalidade podem ser difíceis de visualizar. Podemos ter uma matriz com 3 dimensões, ou mesmo uma matriz de vetores. Para dados com quatro dimensões, você poderia imaginar uma matriz em que cada elemento é uma matriz (algo como uma matriz de matrizes). Daí em diante você consegue imaginar a dificuldade no tratamento de dados altamente dimensionais.

Uma imagem, por exemplo, é composta de 3 canais RGB, conforme figura abaixo:



Portanto, uma imagem poderia ser armazenada no formato de um tensor de 3 dimensões, com uma dimensão para cada canal da imagem: Red, Blue e Green.

Dentro de uma matriz, cada posição é representada pelos índices, que nos ajudam a acessar cada elemento para o processamento.



Algo que costuma causar confusão, é a forma como usamos os índices para buscar os dados na matriz. Na matemática (e como você pode observar na figura acima), começamos o índice por 1. Portanto, o valor 4 na matrix está na posição a_{21} (linha 2, coluna 1). Mas as linguagens de programação possuem uma representação diferente e no caso da linguagem Python, índices começam por 0. Logo, em Python, a matriz seria representada por esta combinação de índices da figura abaixo. Fique atento para não confundir a representação



matemática com a representação da linguagem de programação. O número 4 agora seria representado pelo índice a_{10} (linha 2, coluna 1, com índices em Python).

$$\begin{array}{cc} \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} & \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix} \\ A & \text{Indices} \end{array}$$

Em anexo, você encontra um script com o passo a passo para representar esses 4 formatos (scalar, vetor, matriz e tensor) com o Numpy. O script pode ser executado em qualquer máquina que tenha Python instalado (não é necessário ter GPU).

Estude o Jupyter Notebook atentamente para aprender como representar esses objetos matemáticos em Python.