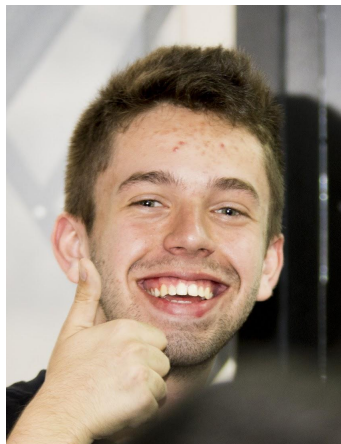

Deep Learning e Tensorflow

por Vinicius Mesel e Guilherme Uzeda

\$ whoarewe



Vinicius Mesel

Pesquisador de Machine Learning

Desenvolvedor Python



Certified Professional

Data Engineer



Guilherme Uzeda

Google Certified Data Engineer

Engenheiro de Machine Learning

Roteiro

Parte 1:

1. Machine Learning 101
2. Introdução a Redes Neurais
3. Deep Learning

Parte 2:

4. Tensorflow: Uma Introdução
 5. Criando e treinando Redes Neurais
 6. O poder do Tensorboard
-

Machine Learning 101

O que é Machine Learning?

Machine Learning = Aprendizado Estatístico

Começou a ser pesquisado por volta de 1950

Primeira rede neural de multcamadas data de 1975

Hoje é presente em quase todos os produtos modernos

O que é Machine Learning?

“ Machine Learning: um campo de estudo que dá aos computadores a habilidade de aprender sem serem explicitamente programados

[Arthur Samuel, 1959](#)

”

Exemplos Reais



Mágica + Matemática = Matemágica

O objetivo é encontrar uma função matemática que possa descrever bem dados desconhecidos

Esse processo se chama treinamento e o processo de utilizar essa função se chama predição ou classificação

Tipos de modelos

Existem inúmeros modelos e metodologias de ML, porém não existe uma bala de prata, um modelo para todos os casos.

Para cada caso de aplicação existe um modelo que irá performar com maior eficiência do que outros

Tipos de modelos

Os modelos podem ser classificados em:

- Supervisionados
- Não Supervisionados
- Semi-supervisionados
- Aprendizado por Reforço

Os modelos mais comuns são:

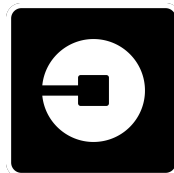
Nearest Neighbor, Naive Bayes, Decision Trees, Linear Regression, Support Vector Machines (SVM), Neural Networks

Redes Neurais: The basics

Neural Nets

Redes Neurais são o tipo de ML mais poderoso e o que está ganhando mais força devido sua ampla aplicação.

Os casos de uso mais conhecidos são:



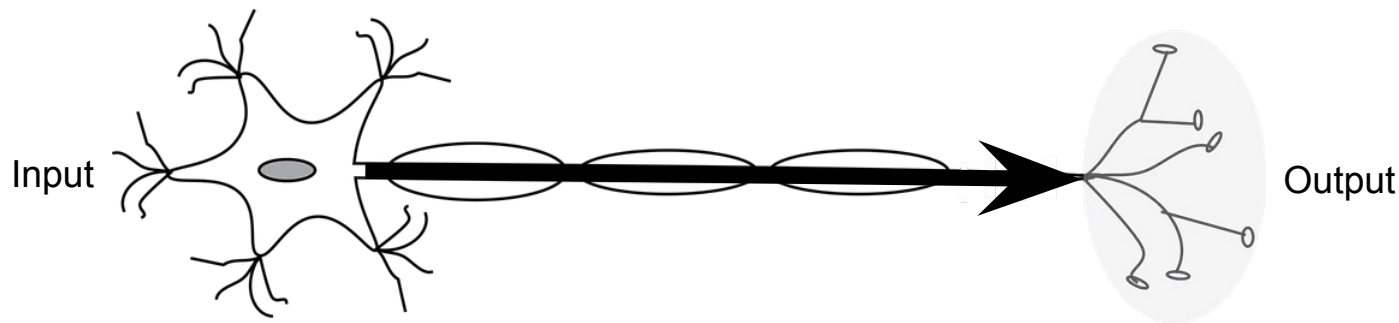
Por que Neural?

Redes Neurais são assim chamadas, pois são constituídas de uma estrutura abstrata chamada de neurônios

As redes neurais surgiram da tentativa de literalmente replicar o funcionamento de um cérebro orgânico

Neurônios Orgânicos

No cérebro humano os neurônios funcionam como moduladores de impulsos elétricos e o conjunto sequencial deles formam memórias, pensamentos, ações e aprendizado



Neurônios Digitais

Numa rede neural podemos pensar nos neurônios como se fossem “moduladores matemáticos”. Mas na verdade eles são compostos de 3 partes:

1. Pesos (weights)
2. Viés (Bias)
3. Função de ativação

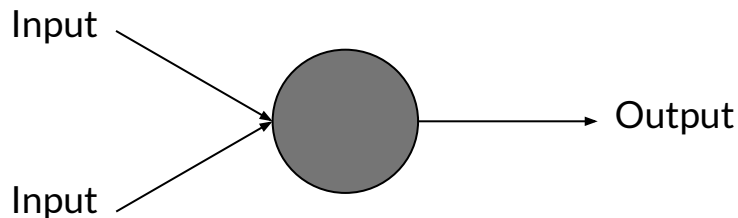
Neurônios Digitais

A estrutura matemática de um neurônio digital pode ser definida como:

$$\text{Output} = F(\text{sum}(w.\text{Input} + b))$$

Onde F = função de ativação, w = pesos, b = viés

Ou seja, podemos visualizar da seguinte forma:



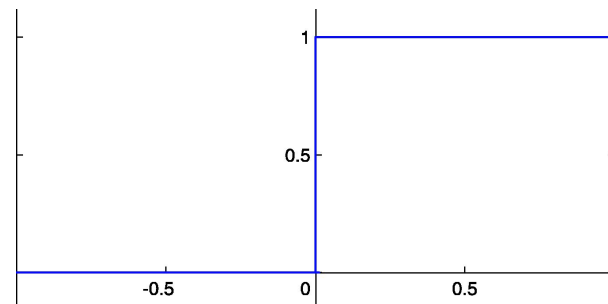
Funções de Ativação

Existem n tipos de funções de ativação, mas sua função sempre é modular a soma do resultado interno do neurônio.

Por exemplo, o neurônio mais antigo e mais simples é o **perceptron** e sua função se dá por:

$$\text{Output} = \begin{cases} 0 & \text{se input.w} \leq t \\ 1 & \text{se input.w} > t \end{cases}$$

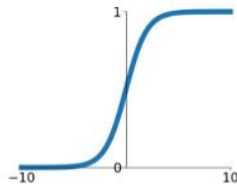
Onde w = pesos e t = limite



Outras Funções de Ativação

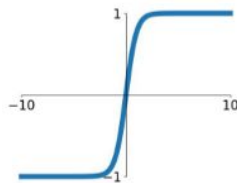
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



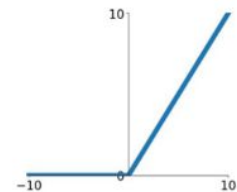
tanh

$$\tanh(x)$$



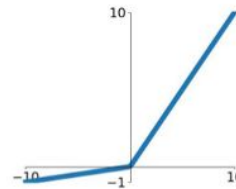
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

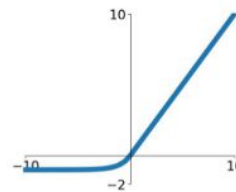


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Como o aprendizado acontece?

Em um rede neural o aprendizado acontece por meio de algo chamado Backpropagation.

Backpropagation é um algoritmo matemático introduzido 1986 por Rumelhart et al. , que permite ajustar a rede neural inteira realizando o aprendizado.

Como o aprendizado acontece?

O Backpropagation é um processo que pode ser resumido em quatro etapas:

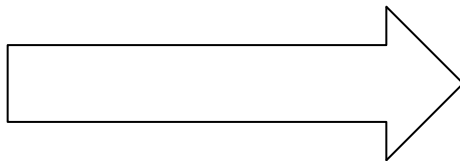
1. Predição (feed forward)
 2. Avaliação do erro total da rede
 3. Cálculo da contribuição de erro de cada camada
 4. Ajuste pequeno de cada um dos pesos (Gradient Descent)
-

Dados de treino

Vamos supor que temos de decidir se vamos ao parque com os seguintes dados:

x1 = condição climática
x2 = condição financeira

	x1	x2
-1	Chovendo	Falido
0	Nublado	Ok de grana
1	Sol	Rico!



Y = 0, não vamos

Y = 1, vamos!

x1	x2	Y
1	0	1
1	1	1
0	1	1
-1	0	0
-1	-1	0
0	-1	0

Uma rede neural simples

Viés constante e 0
Pesos iniciais 0
Perceptrons com $t=0$
Função erro: $(\text{output}-Y)^2$

Dados de treino

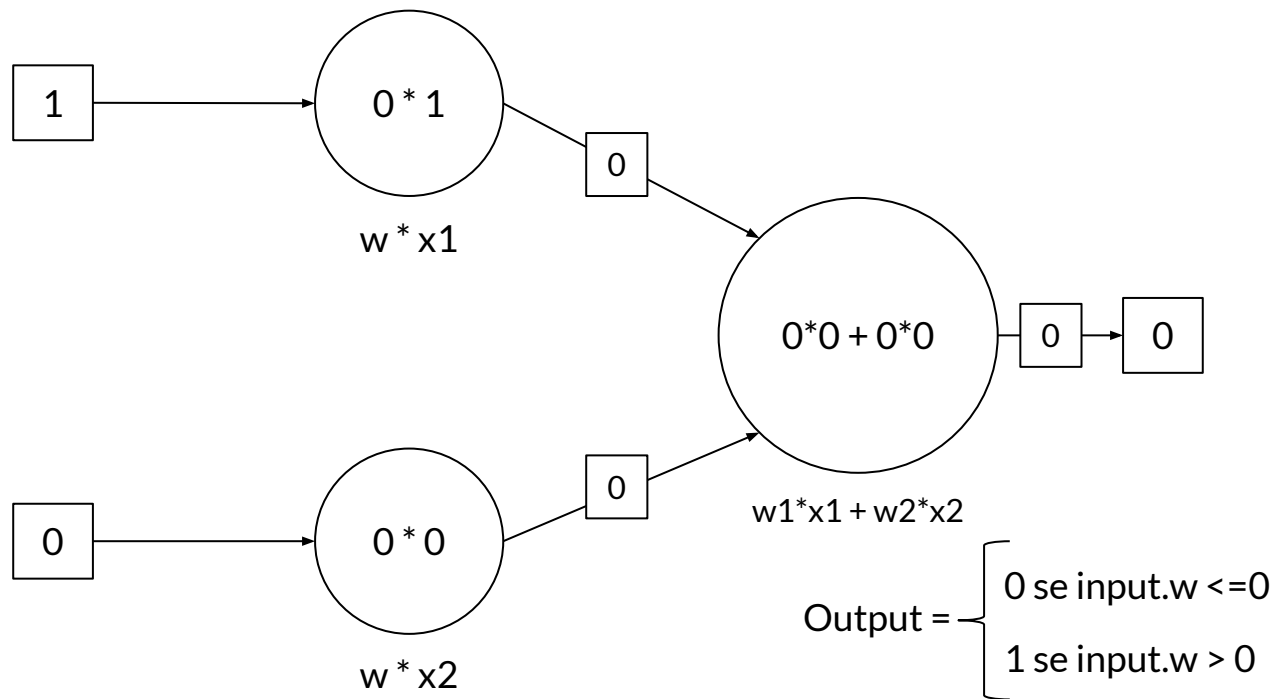
x1	x2	Y
1	0	1
1	1	1
0	1	1
-1	0	0
-1	-1	0
0	-1	0

Inputs

Camada 1

Camada 2

Output



Treinando a rede neural

Dados de treino

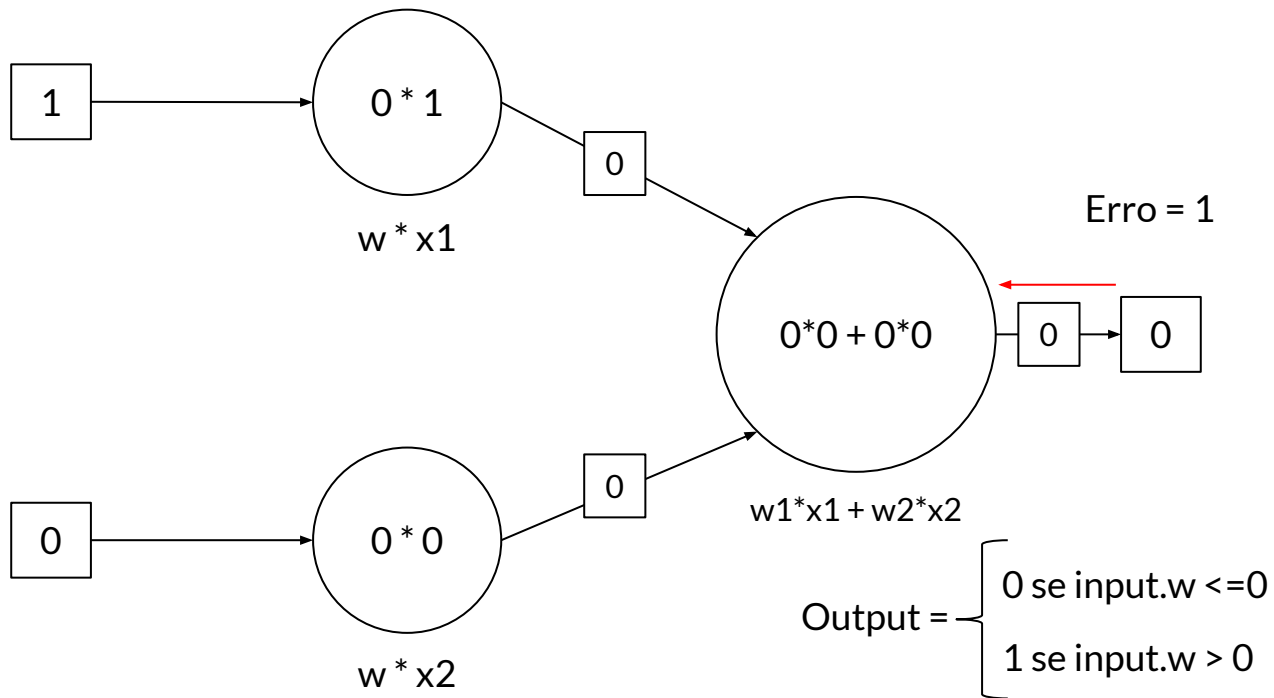
x1	x2	Y
1	0	1
1	1	1
0	1	1
-1	0	0
-1	-1	0
0	-1	0

Inputs

Camada 1

Camada 2

Output



Treinando a rede neural

Dados de treino

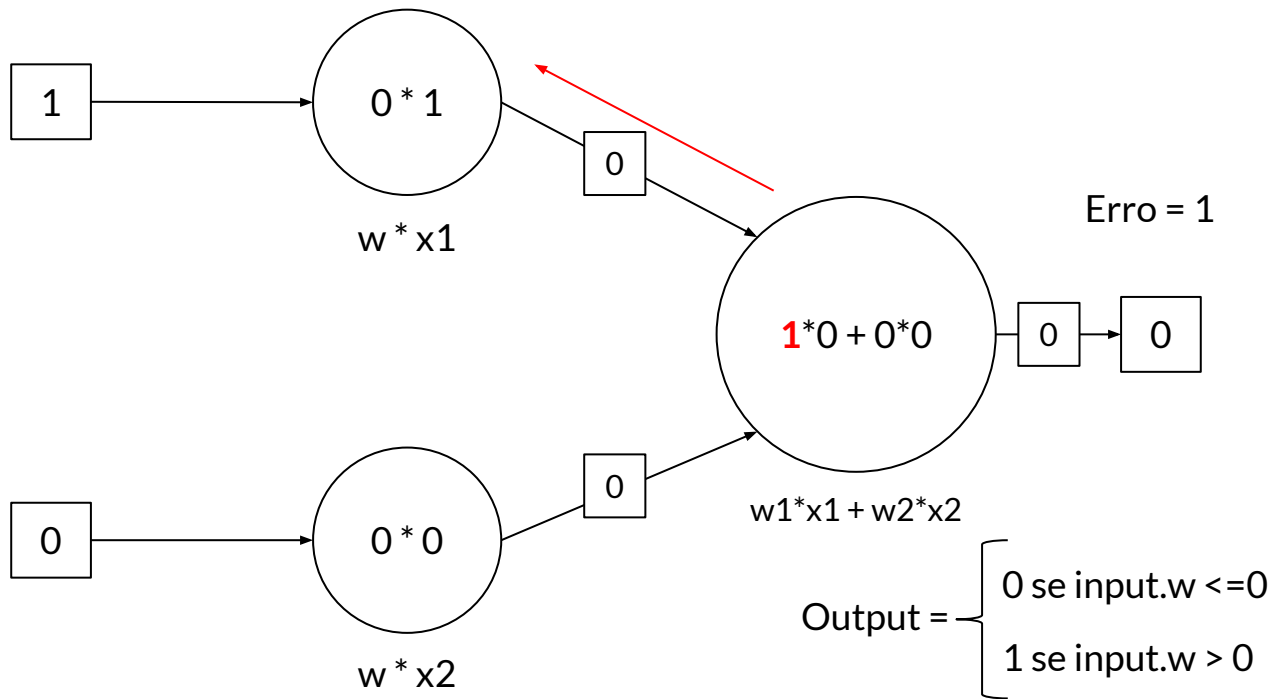
x1	x2	Y
1	0	1
1	1	1
0	1	1
-1	0	0
-1	-1	0
0	-1	0

Inputs

Camada 1

Camada 2

Output



Treinando a rede neural

Dados de treino

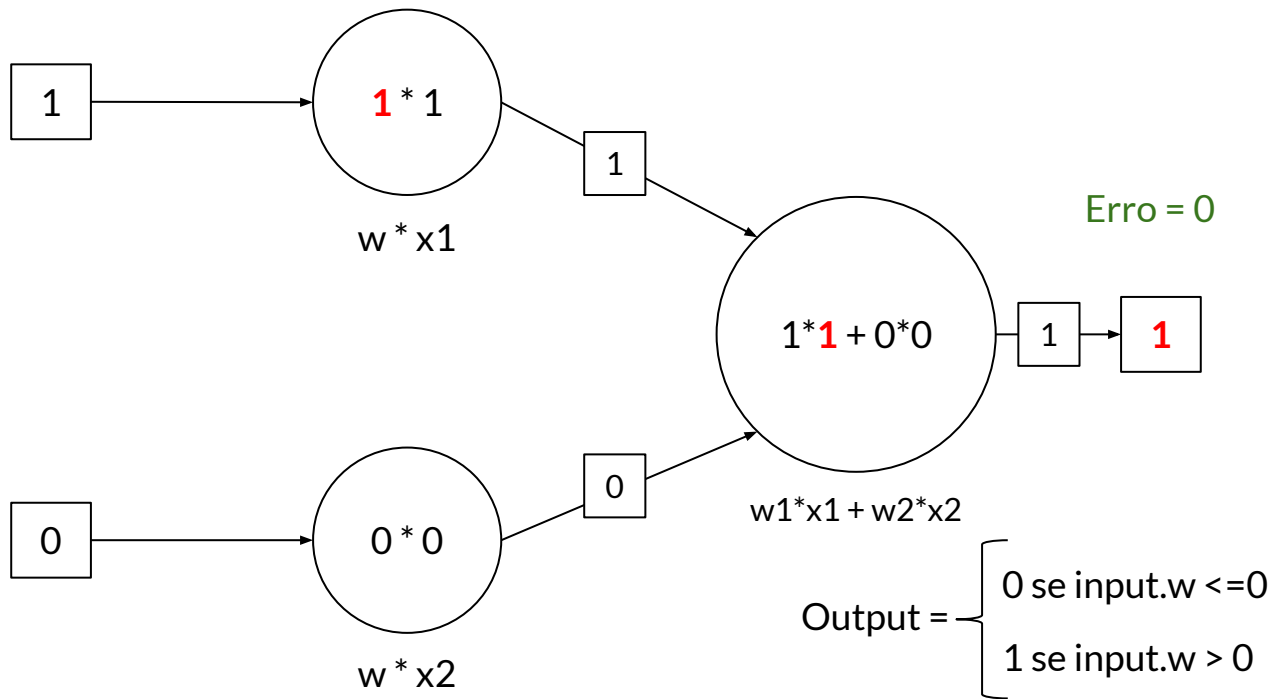
x1	x2	Y
1	0	1
1	1	1
0	1	1
-1	0	0
-1	-1	0
0	-1	0

Inputs

Camada 1

Camada 2

Output



Treinando a rede neural

Dados de treino

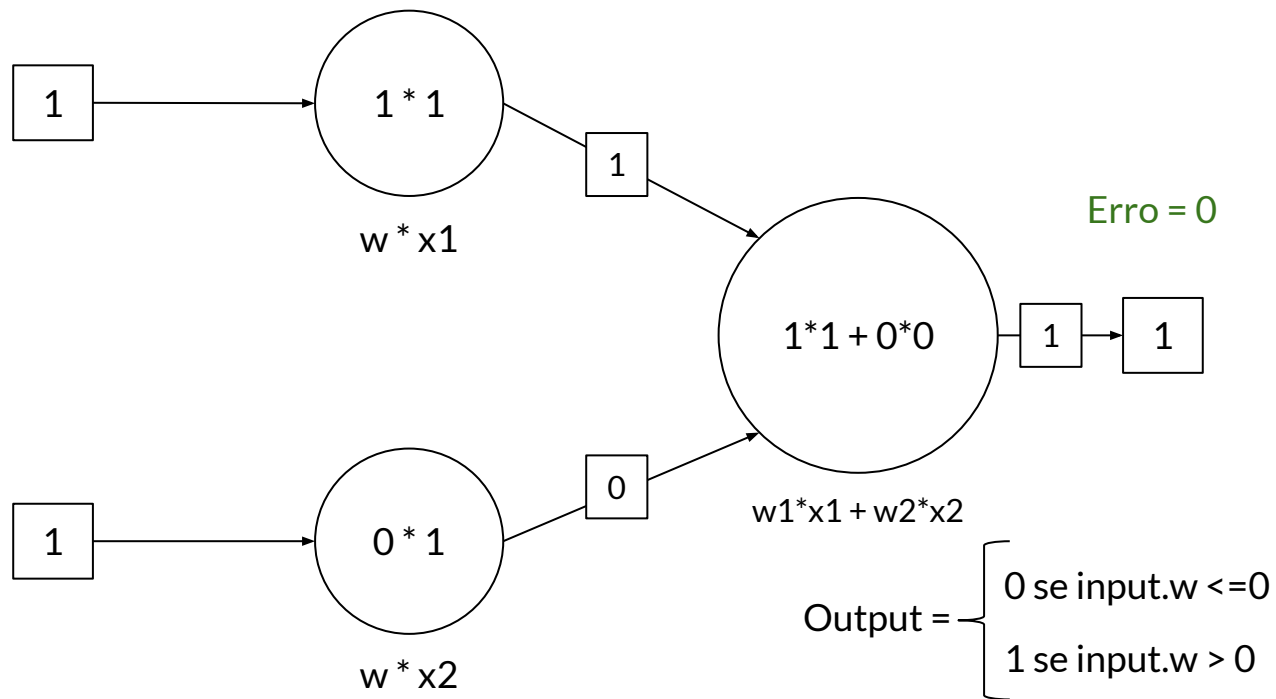
x1	x2	Y
1	0	1
1	1	1
0	1	1
-1	0	0
-1	-1	0
0	-1	0

Inputs

Camada 1

Camada 2

Output



Treinando a rede neural

Dados de treino

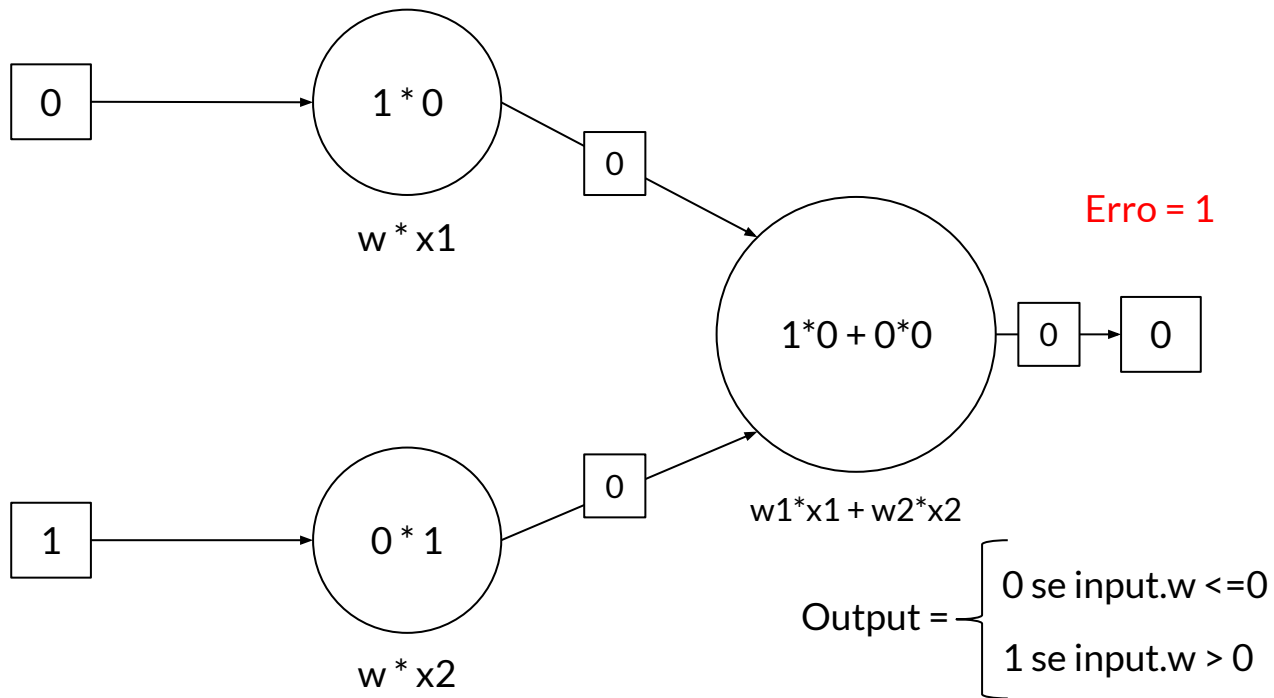
x1	x2	Y
1	0	1
1	1	1
0	1	1
-1	0	0
-1	-1	0
0	-1	0

Inputs

Camada 1

Camada 2

Output



Treinando a rede neural

Dados de treino

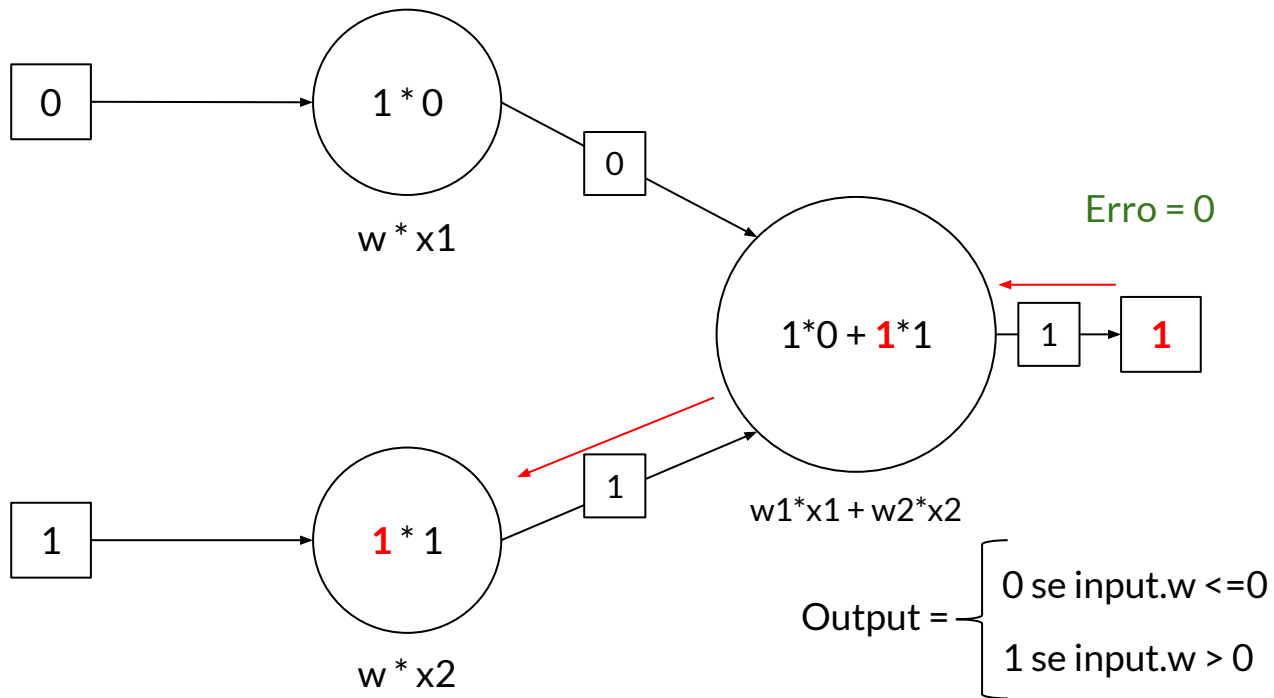
x1	x2	Y
1	0	1
1	1	1
0	1	1
-1	0	0
-1	-1	0
0	-1	0

Inputs

Camada 1

Camada 2

Output



Treinando a rede neural

Dados de treino

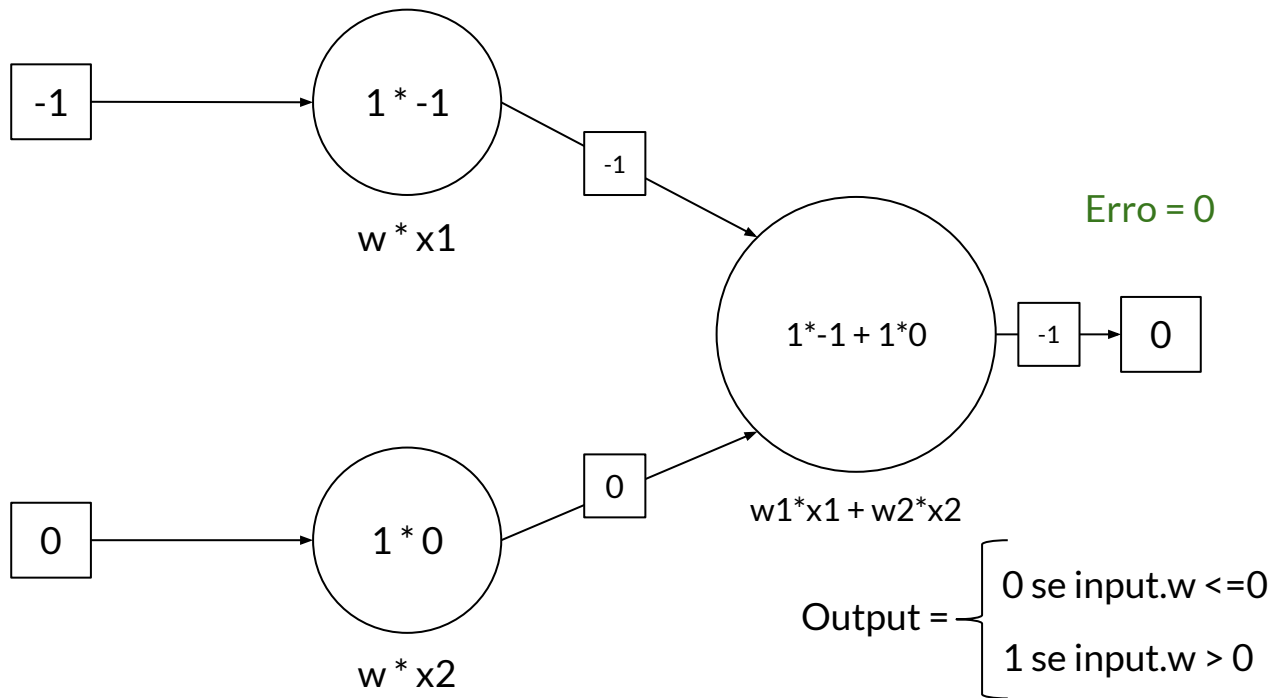
x1	x2	Y
1	0	1
1	1	1
0	1	1
-1	0	0
-1	-1	0
0	-1	0

Inputs

Camada 1

Camada 2

Output



Treinando a rede neural

Dados de treino

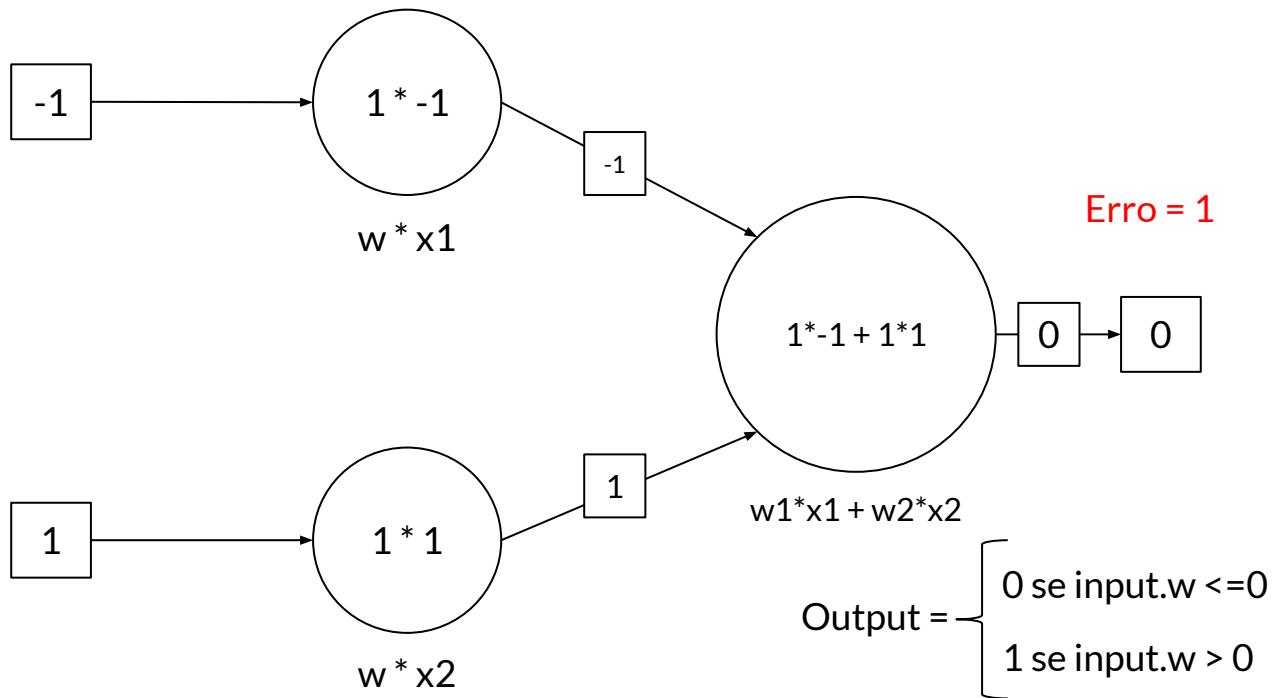
x1	x2	Y
-1	1	1

Inputs

Camada 1

Camada 2

Output



Treinando a rede neural

Dados de treino

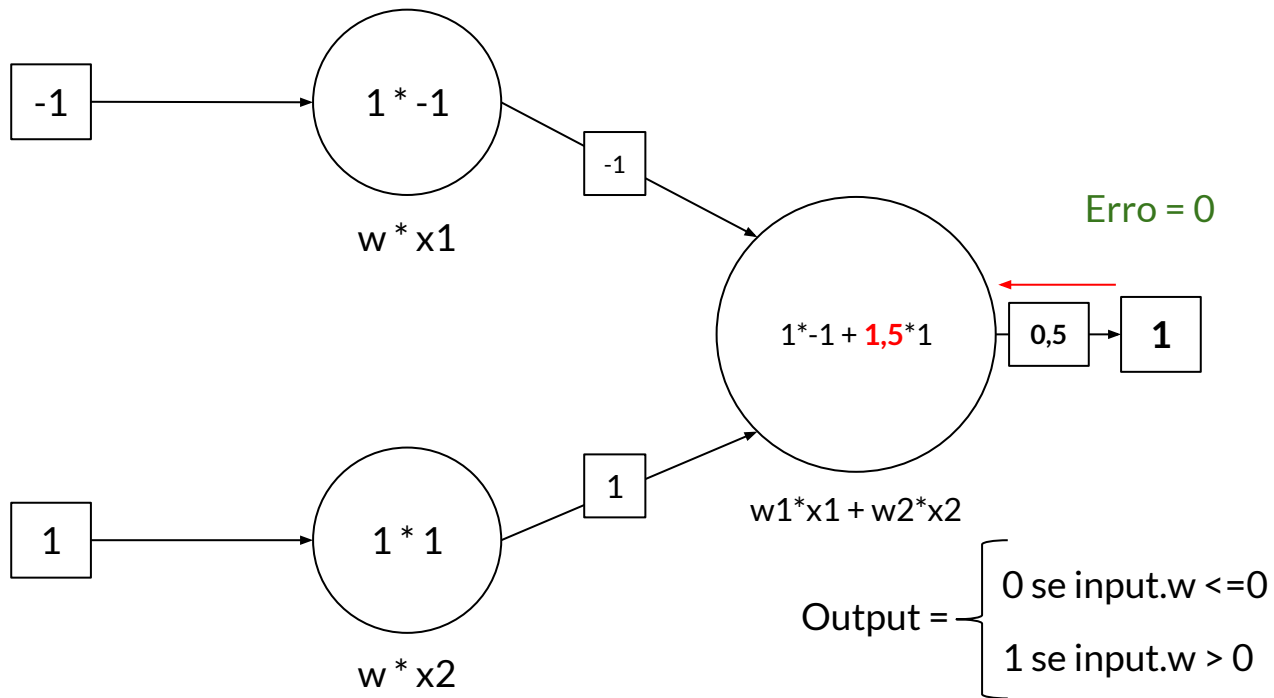
x1	x2	Y
-1	1	1

Inputs

Camada 1

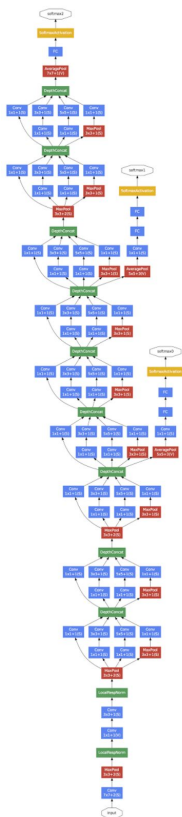
Camada 2

Output

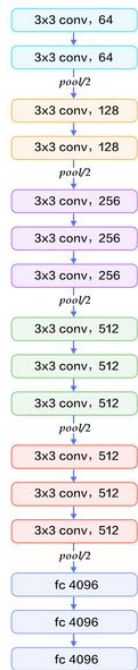


Going Deep

Inception Net



VGGNet16



Deep Learning?

Deep Learning normalmente se refere aos modelo de redes neurais com um grande quantidade de layers.

O ponto positivo desses modelos é o seu alto poder de abstração.

O ponto negativo reside no fato de que necessitam de uma grande quantidade de dados para serem treinados e alto poder computacional

Aplicações

No exemplo anterior, consideramos apenas duas features (x_1 e x_2), mas e se quiséssemos usar 50 ou 1280 features para reconhecer a raça de um cão, o modelo de um carro ou o sentimento intrínseco a uma fala.

Aí entra o Deep Learning!

Os Modelos

Por requererem muitos recursos e tempo para treinar, os modelos de Deep Learning é muito comum apenas retreinar os modelos mais conhecidos em dados diferentes.

Essa prática se chama Transfer Learning ou Retraining e pode reduzir muito o tempo de treinamento de um modelo grande

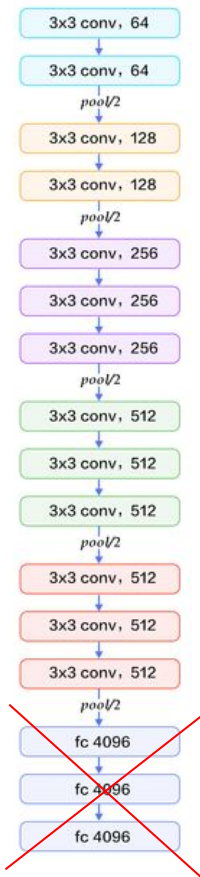
Os Modelos

Os modelos mais conhecidos são:

- RCNN
- Inception
- VGG
- Resnet
- MobileNet

Os arquivos treinados desses modelos para Tensorflow podem ser encontrados [aqui](#)

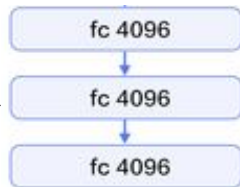
VGGNet16



Como reutilizar modelos

É possível reutilizar modelos substituindo as últimas camadas por novas camadas. Como a classificação só ocorre nas últimas camadas é possível fazer isso e reaproveitar os pesos já treinados da maioria da rede

Novas Camadas



O Playground

↺ ▶

Epoch
000,000

Learning rate
0.03

Activation
Tanh

Regularization
None

Regularization rate
0

Problem type
Classification

DATA

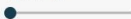
Which dataset do you want to use?



Ratio of training to test data: 50%



Noise: 0



Batch size: 10



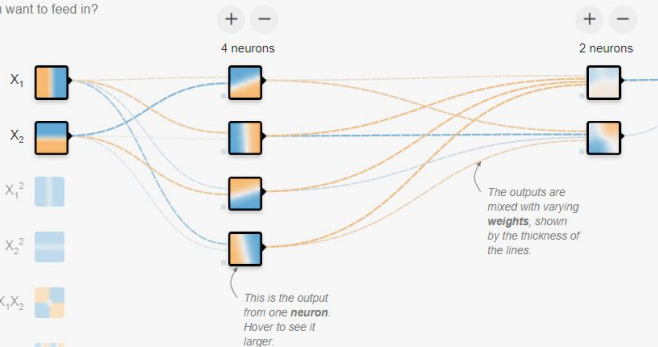
REGENERATE

FEATURES

Which properties do you want to feed in?

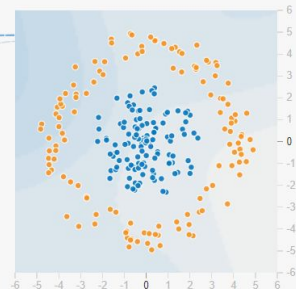


2 HIDDEN LAYERS

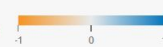


OUTPUT

Test loss 0.531
Training loss 0.524

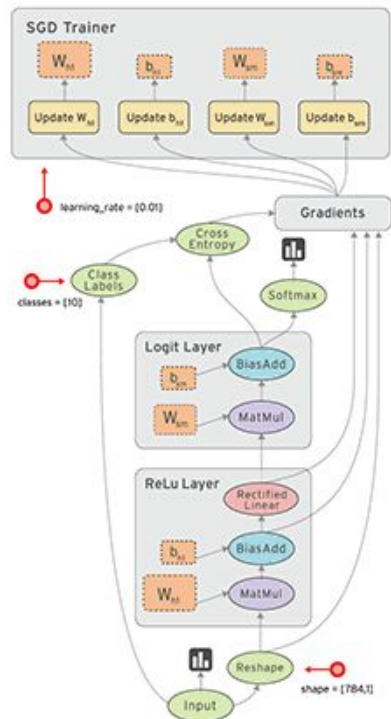


Colors shows
data, neuron and
weight values.



☐ Show test data ☐ Discretize output

Tensorflow

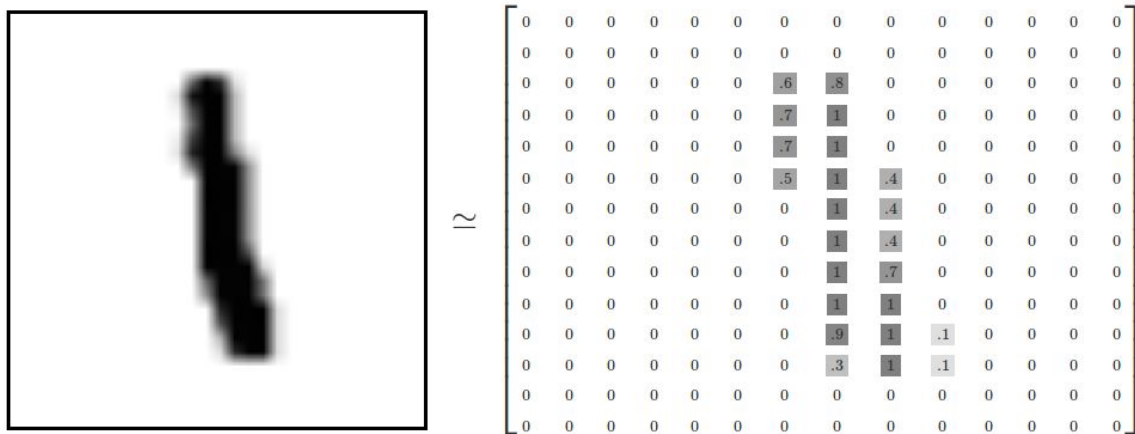


Tensorflow: Uma Introdução

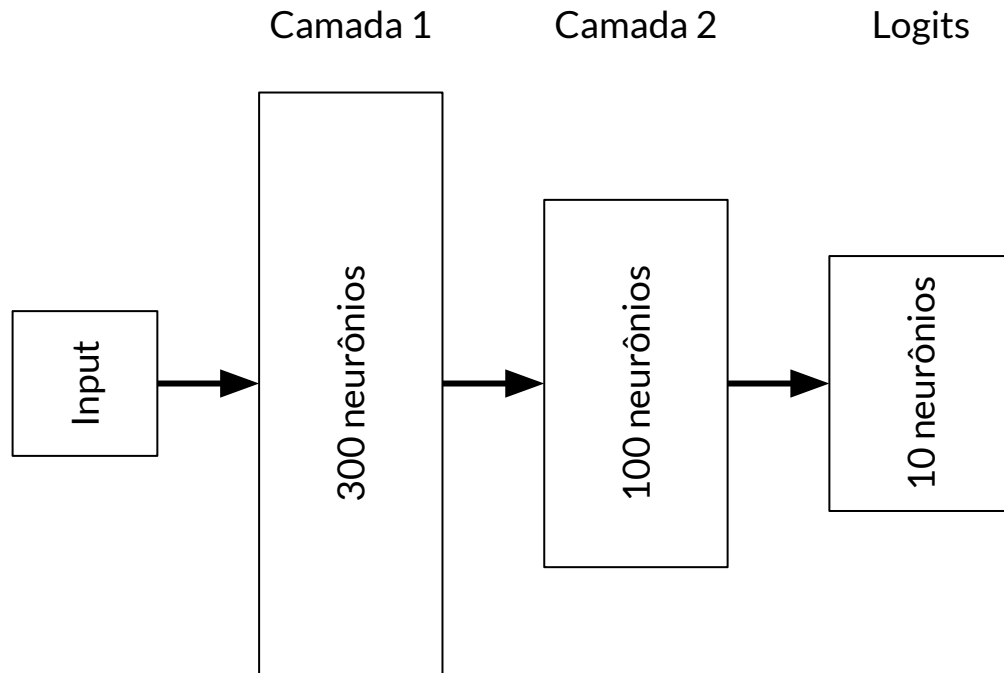
Antes de começarmos a programar no Tensorflow, precisamos entender suas estruturas de dados e processamento.

- **Grafos** - Cada nó do grafo representa uma operação computacional a ser feita.
- **Seções** - É uma classe que representa a conexão entre o cliente (Python) e o runtime (C++),
- **Tensor** - É uma estrutura de dados que contém valores primitivos e estão em um array *n-dimensional*

Criando e treinando Redes Neurais



Uma rede Densa



Logits?!

Logits são a representação de um rótulo em um array

O array abaixo está indicando que o rótulo é o número 5

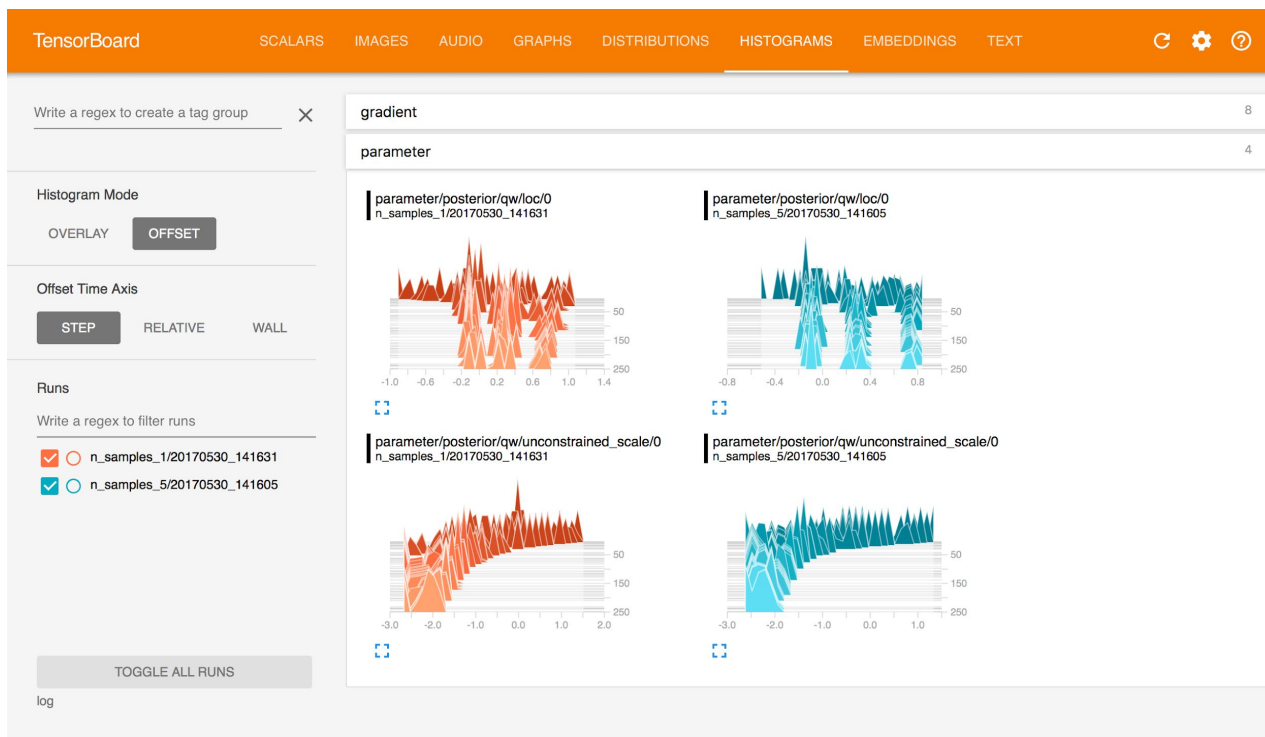
Rótulo	0	1	2	3	4	5	6	7	8	9
Array	0	0	0	0	0	1	0	0	0	0

Logits?!

No caso da última camada da rede, cada valor do array representa a ativação de um dos 10 neurônios, ou seja, nesse caso o neurônio correspondente ao rótulo 5 está ativo e indicando que o uma classificação do número 5

Rótulo	0	1	2	3	4	5	6	7	8	9
Array	0	0	0	0	0	1	0	0	0	0

O poder do Tensorboard



Fontes e material de leitura:

[What Is Machine Learning? - Jean Francois Puget](#)

[Neural Networks History: The 1940's to the 1970's - Stanford](#)

[Neural Networks and Deep Learning - Michael Nielsen](#)

[Not another MNIST tutorial with TensorFlow - Justin Francis](#)

[Hands-On Machine Learning with Scikit-Learn and TensorFlow - Aurelien Geron](#)

[Machine Learning for Humans - Vishal Maini e Samer Sabri](#)