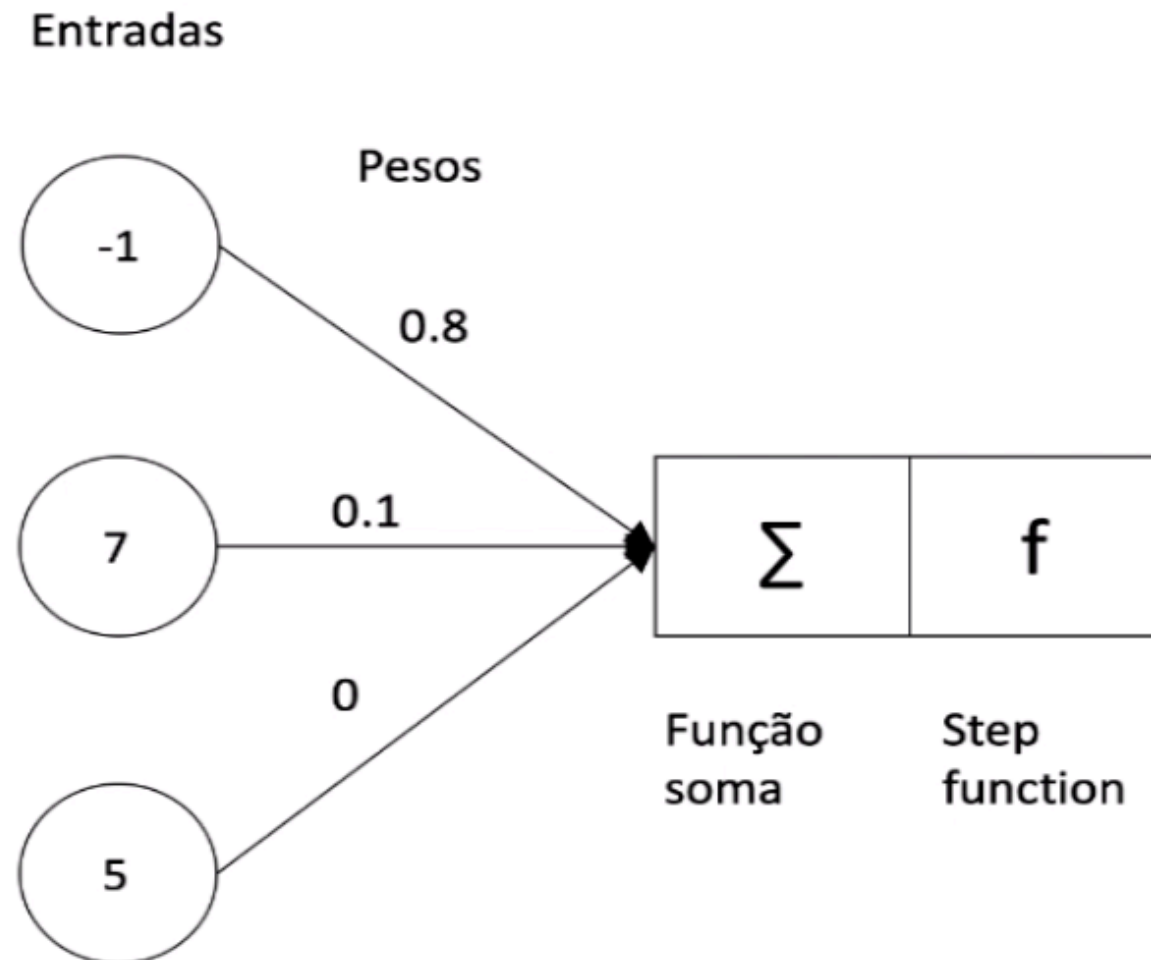


# Produto escalar

Prof. Luiz Alberto Bordignon

# Calculo Peso Neurônio Artificial



# Calculo Peso Neurônio Artificial

- Para o calculo do peso de um neurônio em uma rede, devemos seguir a equação:

$$soma = \sum_{i=1}^n x_i * w_i$$

# Calculo Peso Neurônio Artificial

- Onde podemos ter um vetor representando as entradas (x) e um vetor representando os pesos (w), o somatório desses valores será utilizado na função de ativação da rede neural.

$$soma = \sum_{i=1}^n x_i * w_i$$

# Calculo Peso Neurônio Artificial - TensorFlow

- Para o TensorFlow, temos a multiplicação de dois vetores.
- E o somatório desses valores.

# Calculo Peso Neurônio Artificial - TensorFlow

```
import tensorflow as tf
```

```
a = tf.constant([-1.0, 7.0, 5.0], name = 'entradas')  
b = tf.constant([0.8, 0.1, 0.0], name = 'pesos')  
multiplicacao = tf.multiply(a,b)
```

```
with tf.Session() as sess:  
    print(sess.run(a))  
    print('\n')  
    print(sess.run(b))  
    print('\n')  
    print(sess.run(multiplicacao))
```

```
[-1.  7.  5.]
```

```
[0.8 0.1 0. ]
```

```
[-0.8  0.7  0. ]
```

# Calculo Peso Neurônio Artificial - TensorFlow

- Calculando o somatório dos pesos calculados no vetor soma.

# Calculo Peso Neurônio Artificial - TensorFlow

```
import tensorflow as tf
```

```
a = tf.constant([-1.0, 7.0, 5.0], name = 'entradas')  
b = tf.constant([0.8, 0.1, 0.0], name = 'pesos')  
multiplicacao = tf.multiply(a,b)  
soma = tf.reduce_sum(multiplicacao)
```

```
with tf.Session() as sess:  
    print(sess.run(a))  
    print('\n')  
    print(sess.run(b))  
    print('\n')  
    print(sess.run(multiplicacao))  
    print(sess.run(soma))
```

```
[-1.  7.  5.]
```

```
[0.8 0.1 0. ]
```

```
[-0.8  0.7  0. ]  
-0.100000024
```



# PlaceHolders

- Placeholders são variáveis que definimos que tipo de dados serão utilizados.
- Não é passado dados e nem inicializada.
- Muito usado para configurações de matrizes e vetores de informações do problema que se está trabalhando.
- Permite criar as operações sem termos os dados (valores).
- Cria um espaço reservado para inserir dados nas operações.

# PlaceHolders

- Sintaxe:

```
import tensorflow as tf
```

```
ph = tf.placeholder('float', None)
```

**Float** determina que tipo de dados queremos trabalhar.

**None** determina a dimensão da estrutura (escalar, vetor, matriz).

# PlaceHolders

- Criando uma operação:

```
import tensorflow as tf
```

```
ph = tf.placeholder('float', None)
```

```
op = ph + 5
```

```
with tf.Session() as sess:  
    result = sess.run(op, feed_dict = {ph: [2, 3, 4]})  
    print(result)
```

```
[7. 8. 9.]
```

**Feed\_dict** insere valores ao placeholder.

# PlaceHolders

- Criando placeholder, no qual não sabemos a quantidade de linhas que podemos usar no problema.

```
import tensorflow as tf
```

```
ph = tf.placeholder('float', [None, 4])
```

```
op = ph * 3  
dados = [[2, 3, 4, 5], [3, 4, 5, 9]]
```

```
with tf.Session() as sess:  
    result = sess.run(op, feed_dict = {ph: dados})  
    print(result)
```

```
[[ 6.  9. 12. 15.]  
 [ 9. 12. 15. 27.]]
```

Atividades

# Calculo Peso Neurônio Artificial - TensorFlow

- Calcule o produto escalar dos seguintes neurônios:
  - N1 = entradas: 13, -5, 2, 6. Pesos: 0.3, 0.1, 0.8, 0.5
  - N2 = entradas: 3, -1, 9. Pesos: 0.1, 0.7, 0.5
  - N3 = entradas: 11, -5, 1. Pesos: 0.1, 0.0, 0.8, 0.1
  - N4 = entradas: -2, -5, 3, 9. Pesos: 0.4, 0.2, 0.7, 0.4

# PlaceHolders

- Crie um ***placeholder*** que receba dados do tipo ***float***, e que tenha o formato de matriz 2x5.
- Crie uma matriz 2x5 utilizando os métodos ***linspace*** e ***reshape***.
- Com esses dados faça uma operação de divisão por 4 dos valores contidos no ***placeholder***.
- Altere a operação de divisão por subtração, e subtraia do ***placeholder*** o valor 20.

# PlaceHolders

- Crie um ***placeholder*** que receba dados do tipo ***float***, e que tenha o formato de matriz 3 X None.
- Crie uma matriz 3 X None utilizando os métodos ***linspace*** e ***reshape***.
- Com esses dados faça uma operação de multiplicação por 2 dos valores contidos no ***placeholder***.