

Importar os pacotes necessários

Importar os pacotes necessários e renomeá-los para serem acessados no código abaixo

In [1]:

```
import numpy as np
import tensorflow as tf
```

Iniciar uma sessão do TensorFlow

In [2]:

```
sess = tf.Session()
```

Definir a função print_tf(x)

Esta função imprime o tipo de x e o seu valor

tf.constant

É uma função para definir uma constante tensor. O retorno desta função é um tensor constante.

In [3]:

```
def print_tf(x):
    print("TIPO: \n %s" % (type(x)))
    print("Valor: \n %s" % (x))
hello = tf.constant("www.deeplearningbrasil.com.br")
print_tf(hello)
```

```
TIPO:
<class 'tensorflow.python.framework.ops.Tensor'>
Valor:
Tensor("Const:0", shape=(), dtype=string)
```

sess.run(weight)

É uma classe que executa operações TensorFlow Uma sessão encapsula o ambiente na qual as operações são executadas e os objetos Tensor são calculados.

In [4]:

```
hello_out = sess.run(hello)
print_tf(hello_out)
```

```
TIPO:
<class 'bytes'>
Valor:
b'www.deeplearningbrasil.com.br'
```

Define novas constantes a e b

Define constantes a e b e as imprime

In [5]:

```
a = tf.constant(1.5)
b = tf.constant(2.5)
print_tf(a)
print_tf(b)
```

TIPO:

```
<class 'tensorflow.python.framework.ops.Tensor'>
```

Valor:

```
Tensor("Const_1:0", shape=(), dtype=float32)
```

TIPO:

```
<class 'tensorflow.python.framework.ops.Tensor'>
```

Valor:

```
Tensor("Const_2:0", shape=(), dtype=float32)
```

sess.run(a) | sess.run(b)

Inicia uma sessão para cada constante e exibe resultado da execução

In [6]:

```
a_out = sess.run(a)
b_out = sess.run(b)
print_tf(a_out)
print_tf(b_out)
```

TIPO:

```
<class 'numpy.float32'>
```

Valor:

```
1.5
```

TIPO:

```
<class 'numpy.float32'>
```

Valor:

```
2.5
```

tf.add(a, b)

Esta função faz a adição de dois parametros. Neste caso os valores não foram somados, pois ainda não foi iniciada a sessão de execução.

In [7]:

```
a_plus_b = tf.add(a, b)
print_tf(a_plus_b)
```

TIPO:

```
<class 'tensorflow.python.framework.ops.Tensor'>
```

Valor:

```
Tensor("Add:0", shape=(), dtype=float32)
```

sess.run(a_plus_b)

Inicializa uma sessão de execução que tem duas constantes a e b, com a operação de adição entre elas.

In [8]:

```
a_plus_b_out = sess.run(a_plus_b)
print_tf(a_plus_b_out)
```

TIPO:

```
<class 'numpy.float32'>
```

Valor:

```
4.0
```

Definir operação mul entre a e b e iniciar uma sessão de execução

Ocorreu erro module 'tensorflow' has no attribute 'mul', pois mul não é conhecido pelo tensorflow

In [9]:

```
a_mul_b = tf.mul(a, b)
a_mul_b_out = sess.run(a_mul_b)
print_tf(a_mul_b_out)
```

```
-----
AttributeError                                Traceback (most recent call last)
<ipython-input-9-72fa51afc8b3> in <module>()
----> 1 a_mul_b = tf.mul(a, b)
      2 a_mul_b_out = sess.run(a_mul_b)
      3 print_tf(a_mul_b_out)
```

AttributeError: module 'tensorflow' has no attribute 'mul'

tf.Variable

Adicionar uma variável com dados aleatórios ao grafo

tf.random_normal([5, 2], stddev=0.1)

Cria uma matriz 5 linhas e 2 colunas, com valores aleatórios e desvio padrão de 0.1

In [10]:

```
weight = tf.Variable(tf.random_normal([5, 2], stddev=0.1))
print_tf(weight)
```

TIPO:

```
<class 'tensorflow.python.ops.variables.Variable'>
```

Valor:

```
<tf.Variable 'Variable:0' shape=(5, 2) dtype=float32_ref>
```

sess.run(weight)

Inicia sessão de execução com a variável weight, porém ocorre um erro pois a variável não foi iniciada antes da sessão de execução.

In [11]:

```
weight_out = sess.run(weight)
print_tf(weight_out)
```

```
-----
FailedPreconditionError                                Traceback (most recent call last)
D:\Programas\Anaconda3\envs\rnp_ufg\lib\site-packages\tensorflow\python\client\session.py in _do_call(self, fn, *args)
    1326         try:
-> 1327             return fn(*args)
    1328         except errors.OpError as e:

D:\Programas\Anaconda3\envs\rnp_ufg\lib\site-packages\tensorflow\python\client\session.py in _run_fn(session, feed_dict, fetch_list, target_list, options, run_metadata)
    1305         feed_dict, fetch_list, target_list,
t,
-> 1306         status, run_metadata)
    1307

D:\Programas\Anaconda3\envs\rnp_ufg\lib\contextlib.py in __exit__(self, type, value, traceback)
     65         try:
--> 66             next(self.gen)
     67         except StopIteration:

D:\Programas\Anaconda3\envs\rnp_ufg\lib\site-packages\tensorflow\python\framework\errors_impl.py in raise_exception_on_not_ok_status()
    465         compat.as_text(pywrap_tensorflow.TF_Message(status)),
--> 466         pywrap_tensorflow.TF_GetCode(status))
    467     finally:

FailedPreconditionError: Attempting to use uninitialized value Variable
[[Node: _retval_Variable_0_0 = _Retval[T=DT_FLOAT, index=0, _device=
="/job:localhost/replica:0/task:0/cpu:0"]](Variable)]]
```

During handling of the above exception, another exception occurred:

```
FailedPreconditionError                                Traceback (most recent call last)
<ipython-input-11-e453db2b7ada> in <module>()
----> 1 weight_out = sess.run(weight)
      2 print_tf(weight_out)

D:\Programas\Anaconda3\envs\rnp_ufg\lib\site-packages\tensorflow\python\client\session.py in run(self, fetches, feed_dict, options, run_metadata)
    893         try:
-> 895             run_metadata_ptr)
    896         if run_metadata:
    897             proto_data = tf_session.TF_GetBuffer(run_metadata_ptr)

D:\Programas\Anaconda3\envs\rnp_ufg\lib\site-packages\tensorflow\python\client\session.py in _run(self, handle, fetches, feed_dict, options, run_metadata)
    1122         if final_fetches or final_targets or (handle and feed_dict_tensor):
-> 1124             feed_dict_tensor, options, run_metadata)
    1125         else:
```

```

1126         results = []

D:\Programas\Anaconda3\envs\rnp_ufg\lib\site-packages\tensorflow\python\client\session.py in _do_run(self, handle, target_list, fetch_list, feed_dict, options, run_metadata)
    1319         if handle is None:
    1320             return self._do_call(_run_fn, self._session, feeds, fetches, targets,
-> 1321                                 options, run_metadata)
    1322         else:
    1323             return self._do_call(_prun_fn, self._session, handle, feeds, fetches)

D:\Programas\Anaconda3\envs\rnp_ufg\lib\site-packages\tensorflow\python\client\session.py in _do_call(self, fn, *args)
    1338         except KeyError:
    1339             pass
-> 1340         raise type(e)(node_def, op, message)
    1341
    1342     def _extend_graph(self):

```

```

FailedPreconditionError: Attempting to use uninitialized value Variable
[[Node: _retval_Variable_0_0 = _Retval[T=DT_FLOAT, index=0, _device=
="/job:localhost/replica:0/task:0/cpu:0"] (Variable)]]

```

tf.initialize_all_variables()

Função para inicializar as variáveis.

THIS FUNCTION IS DEPRECATED. It will be removed after 2017-03-02. Instructions for updating: Use tf.global_variables_initializer instead.

In [12]:

```

init = tf.initialize_all_variables()
sess.run(init)

```

```

WARNING:tensorflow:From D:\Programas\Anaconda3\envs\rnp_ufg\lib\site-packages\tensorflow\python\util\tf_should_use.py:175: initialize_all_variables (from tensorflow.python.ops.variables) is deprecated and will be removed after 2017-03-02.

```

Instructions for updating:
Use `tf.global_variables_initializer` instead.

sess.run(weight)

Atribui valores à variável com matriz aleatória

In [13]:

```
weight_out = sess.run(weight)
print_tf(weight_out)
print ("INITIALIZING ALL VARIALBES")
```

TIPO:

```
<class 'numpy.ndarray'>
```

Valor:

```
[[ 0.01781844  0.05108102]
 [-0.0785652  -0.09789164]
 [-0.03827545 -0.00432287]
 [ 0.07305237  0.05169081]
 [-0.11997268 -0.13522641]]
```

INITIALIZING ALL VARIALBES

tf.placeholder(tf.float32, [None, 5])

Cria um espaço reservado do tipo float32, formando uma matrix de ? por 5

In [14]:

```
x = tf.placeholder(tf.float32, [None, 5])
print_tf(x)
```

TIPO:

```
<class 'tensorflow.python.framework.ops.Tensor'>
```

Valor:

```
Tensor("Placeholder:0", shape=(?, 5), dtype=float32)
```

oper = tf.matmul(x, weight)

Define operação de multiplicação entre x (?, 5) e weigth (5 x 2)

In [15]:

```
oper = tf.matmul(x, weight)
print_tf(oper)
```

TIPO:

```
<class 'tensorflow.python.framework.ops.Tensor'>
```

Valor:

```
Tensor("MatMul:0", shape=(?, 2), dtype=float32)
```

np.random.rand(1, 5)

Inicializa uma matriz Numpy (1 x 5) com números aleatórios entre 0 e 1

sess.run(oper, feed_dict={x: data})

Inicia sessão de execução com os valores aleatórios data e com a operação de multiplicação de matrizes. Resultando numa matriz de (1 x 2)

In [16]:

```
data = np.random.rand(1, 5)
oper_out = sess.run(oper, feed_dict={x: data})
print_tf(oper_out)
```

TIPO:

```
<class 'numpy.ndarray'>
```

Valor:

```
[[-0.05984762 -0.05425179]]
```

np.random.rand(2, 5)

Inicializa uma matriz Numpy (2 x 5) com números aleatórios entre 0 e 1

sess.run(oper, feed_dict={x: data})

Inicia sessão de execução com os valores aleatórios data e com a operação de multiplicação de matrizes. Resultando numa matriz de (2 x 2)

In [17]:

```
data = np.random.rand(2, 5)
oper_out = sess.run(oper, feed_dict={x: data})
print_tf(oper_out)
```

TIPO:

```
<class 'numpy.ndarray'>
```

Valor:

```
[[-0.08749945 -0.0736561 ]
 [-0.14074717 -0.14042133]]
```

In []: