

## Lista de Exercícios 10 – TP555 Inteligência Artificial e Machine Learning

Aluno: Bruno Ferreira Gomes

Matrícula:842

### Ex1)

Podemos dizer que com o uso de grafos, as principais vantagens seriam a otimização e controle do uso dos recursos computacionais, a facilidade no debugging do código, a convergência de problemas bem complexos, como por exemplo redes neurais em sistemas mais lineares e bem separados, assim como uma IDE de pós processamento, com fácil interação, que conseguimos plotar os dados calculados e observar as respostas de forma dinâmica.

Se o grafo é realizado de forma coerente, tudo está certo. Mas se o grafo for feito de forma muito dispersa e sem o devido controle, seria alocado muita memória para o processamento, tornando uma vantagem em uma desvantagem.

### Ex2)

Sim, foram avaliados ambos os comandos, e podemos dizer que ambos geraram a mesma resposta. Segue imagem da avaliação:

```
In [1]: import tensorflow as tf
import tensorflow.compat.v1 as tf
tf.disable_v2_behavior()

WARNING:tensorflow:From /srv/conda/envs/notebook/lib/python3.6/site-packages/tensorflow/python/compat/v2_compat.py:96: disable_resource_variables (from tensorflow.python.ops.variable_scope) is deprecated and will be removed in a future version.
Instructions for updating:
non-resource variables are not supported in the long term

In [2]: w = tf.constant(3)
x = w + 2
a = x + 5

with tf.Session() as sess:
    a_val = a.eval(session=sess)
    a1_val = sess.run(a)

    print(a_val)
    print(a1_val)

10
10
```

In [ ]:

### Ex3)

Sim, foram avaliados ambos os comandos, e podemos dizer que ambos geraram a mesma resposta. Segue imagem da avaliação:

```
In [1]: import tensorflow as tf
import tensorflow.compat.v1 as tf
tf.disable_v2_behavior()

WARNING:tensorflow:From /srv/conda/envs/notebook/lib/python3.6/site-packages/tensorflow/python/compat/v2_compat.py:96: disable_
resource_variables (from tensorflow.python.ops.variable_scope) is deprecated and will be removed in a future version.
Instructions for updating:
non-resource variables are not supported in the long term

In [2]: w = tf.constant(3)
x = w + 2
a = x + 5
b = x + 10

with tf.Session() as sess:
    a_val, b_val = a.eval(session=sess), b.eval(session=sess)
    a1_val, b1_val = sess.run([a, b])

    print(a_val)
    print(b_val)
    print(a1_val)
    print(b1_val)

10
15
10
15

In [ ]:
```

## Ex4)

Não! Para cada grafo construído, deve ser criada uma sessão separada. Podemos criar dois grafos (grafo padrão e grafo extra), e mudar entre eles com o uso de algumas funções, porém deve ser executado somente um grafo por sessão.

## Ex5)

A variável inicia sua vida útil quando o inicializador é executado na sessão.

A variável é destruída quando a sessão é encerrada.

Podemos então notar que os valores das variáveis são atribuídos durante a execução do grafo, e quando todo o processamento é executado, estes dados são apagados.

## Ex6)

A diferença é que para o nó do tipo variável temos que atribuir um valor inicial quando este é declarado. Já para o nó placeholder, não temos que atribuir este valor inicial.

Podemos dizer:

### Variavel

- Para aprender os parâmetros;
- Valores podem ser derivados do treinamento;
- Valores iniciais são necessários (frequentemente aleatórios).

### PlaceHolders

- Aloca memória para os dados;
- Valores iniciais não são necessários, porém podem ser colocados.

### **Ex7)**

Quando depende do placeholder e não é passado nenhuma alocação de valores para este, é gerado um erro do tipo exception. Tudo isso acontece, pois, ao ser utilizado a função do placeholder, ele já espera a instancia e variáveis (valores alocados) a serem alocadas, caso esse valor seja vazio, o retorno para a função não existe mais. Deste modo acontece o erro Exception.

### **Ex8)**

Podemos utilizar o placeholder como um dos argumentos de uma função inicializada primeiramente com "0" por exemplo. Durante o período de execução/treinamento , ao atribuímos o valor do placeholder e utilizarmos a função assign, esse valor que antes era "0", pode ser substituído por outro termo, ou melhor explicando, pode ser manipulado de forma a modificar a resposta final do algoritmo. Ao modificar este termo com a função assign, podemos agora ter outra informação recebendo o valor do placeholder, que muda a saída do sistema.

