TP555 – Inteligência Artificial e Machine Learning
Josino Villela da Silva Neto – Matrícula: 854 – Mestrado em Engenharia de Telecomunicações

Lista de Exercícios #10

1. Acredito que as principais vantagens são:

- Escalabilidade possível em grandes projetos, por permitir que treinamentos gigantescos possam ser divididos entre diferentes servidores;
- Utilização de códigos otimizados para a resolução de problemas complexos;
- Suporte a visualização interativa dos modelos treinados, através do TensorBoard;
- Otimização de recursos computacionais;
- Flexibilidade por permitir a interop entre diferentes sistemas operacionais e dispositivos.

2. Sim, os comandos são equivalentes. Os resultados apresentados são os mesmos.

3. Não, os comandos não são equivalentes. Os resultados apresentados são os mesmos.

a_val, b_val = a.eval(session=sess), b.eval(session=sess): o grafo é executado duas vezes, a primeira para a_val e a segunda para b_val.

a_val, b_val = sess.run([a, b]): o grafo é executado apenas uma vez.

4. Não. Dois grafos não podem ser executados na mesma sessão. Para que ambos sejam executados, primeiramente, devem ser unificados em apenas um. Em outras palavras, apenas um grafo é executado por sessão.

5. Uma variável é inicializada quando seu inicializador é executado em uma sessão. Por fim, uma variável é destruída quando a sessão é encerrada.

6. Um nó tipo placeholder é aquele que é capaz de transferir dados ao grafo durante o processo de treinamento. Por outro lado, um nó tipo variável é aquele que armazena um valor e este é mantido por sucessivas execuções.

7. Se não for especificado nenhum valor para o placeholder, é gerada uma exception (exceptions são erros - https://www.tensorflow.org/api_docs/python/tf/errors). Se a operação não depender do placeholder, nenhuma exception é gerada.

8. O valor de uma variável, pode ser atribuído durante a fase de treinamento, ao utilizar a linha de comando abaixo:
new_var = tf.assign(var_name, var_new_value/var_expression)

Onde:
new_var – corresponde ao nome da variável que terá seus valores atualizados durante a fase de execução;
var_name – nome atribuído a variável que terá seus valores atualizados;
var_new_value – novo valor que será atribuído a variável na (i+1)-ésima execução;
var_expression – expressão que dita a atualização de valores da variável var_name, para a (i+1)-ésima execução.

```
n_epochs = 1000
learning_rate = 0.01

X = tf.constant(scaled_housing_data_plus_bias, dtype=tf.float32, name="X")
y = tf.constant(housing.target.reshape(-1, 1), dtype=tf.float32, name="y")
theta = tf.Variable(tf.random_uniform([n + 1, 1], -1.0, 1.0), name="theta")
y_pred = tf.matmul(X, theta, name="predictions")
error = y_pred - y
mse = tf.reduce_mean(tf.square(error), name="mse")
gradients = 2/m * tf.matmul(tf.transpose(X), error)
training_op = tf.assign(theta, theta - learning_rate * gradients)

init = tf.global_variables_initializer()

with tf.Session() as sess:
    sess.run(init)

    for epoch in range(n_epochs):
        if epoch % 100 == 0:
            print("Epoch", epoch, "MSE =", mse.eval())
        sess.run(training_op)

    best_theta = theta.eval()
```

O código ao lado mostra uma forma de atualizar o valor de uma variável durante o treinamento de um modelo, utilizando a função tf.assign(). Neste caso, a variável theta é atualizada.

9. Exercício implementado em python.

10. Exercício implementado em python.