1.**Tensors manipulation**

Element wise squaring : tf.square(_ _ tensor_ _)

Sum of all elements : tf.reduce_sum(_ _tensor _ _)

Mean of all elements : tf.reduce_mean(_ _ tensor _ _)

Element wise multiplication : tf.multiply(_ _ tensor1 _ _ , _ _ tensor2_ _)

Adding two tensors : tf.add(_ _ tensor1_ _ , _ _ tensor2_ _)

Tensor Multiplications : tf.matmul(_ _ tensor1_ _ , _ _ tensor2_ _)

Printing a Tensor (inside a session): sess.run(_ _ tensorname_ _) returns a numpy array

2.**Initializers**

tf.zeros_initiallizers or tf.initializers.zero initializes all to 0 {tf.zeroes(shape) returns a zero matrix}

tf.initializers.random_normal initializes with random values not farther away than standard dev from mean

tf.initializers.truncated_normal

initializes with random values not farther away than standard dev from mean

Variable Syntax

Declaration : _ _ variablename_ _ =tf.get_variable("_ _ variablename_ _ " , _ _ shape_ _ , dtype = _ _datatype_ _ , initializer=_ _ initialization_ _)

A variable is just a n dimensional array, the shape is the size of each dimension [1,2,3] means first dimension has 1, second has 2, third has 3 elements total = 1*2*3 elements and dtype is datatype (ex-tf.int32) of those 6 element. Example of initializer : tf.zeros_initializer

variable is trainable while placeholder is not, it doesn’t need initialization either, it is provided at runtime by

_ _ placeholdername_ _=tf.placeholder(_ _datatype_ _,shape=_ _shape_ _)

_ _ variablename_ __=tf.variable(_ _data_ _,_ _datatype_ _)

sess.run(.... , feed_dict={x:[1,2,3]....})

Sessions

Starting a Session :

_ _ objectname _ _ = tf.Session()

sess = tf.Session()

Closing a session : sess.close()