[brief introduction]

1. Collections is a dictionary-like object that the tensorflow keep track of its graph variables

For example, when tf.Variable() is called, this variable tensor is stored in the collections of defualt graph with the key of “variables” and “trainable\_variables”.

1. Collection enable us to get access to these variables tensor even though we didn’t catch them with python variable.
2. Each graph has its own collections, so the many collection related functions are graph method.

[function to get access of variables]

<variable\_tensor\_list> = <graph>.get\_collection(<key>, <scope>=None)

<key>: string, the key of the collection. The default graph has keys like ‘variables’, ‘trainable\_variables’

<scope>: string, the further variable searching keyword. It will return the list of variable tensor whose name   
matches the keyword

( 1. the tensor need to be named to be searched with <scope>

2. the keyword should be the substring that contain the first character.

e.g. a = tf.Variable(0, name = ‘w’)  
b = tf.Variable(0, name = ‘aw’)  
c = tf.Variable(0, name = ‘wa’)  
  
<graph>.get\_collections(‘variables’, ‘w’)

>> [a, c] # Even though b has name ‘w’ in it, it is not a match )

[automatically adding rule]

* rule:

1. If the variable doesn’t specify collection, it will be automatically added to key ‘variables’

2. If the variable is trainable, it will be automatically added to key ‘trainable\_variables’

* example:

1. trainable default variable

>> tf.Variable(0) # can access by <graph>.get\_collection(‘variables’)

<graph>.get\_collection(‘trainable\_variables)

1. not trainable default variable

>> tf.Variable(0, trainalbe=False) # can access by <graph>.get\_collection(‘variables’)

1. trainalbe variable with specific collection

>> tf.Varaible(0, collection=[‘hello, ‘test’]) # can be aceess by <graph>.get\_collection(‘hello’)

<graph>.get\_collection(‘test’)

<graph>.get\_collection(‘trainable\_variables)