[Using the numpy]

* Load the numpy array to the tensor Variable  
  >> w = tf.get\_variable(<name>, initializer = <numpy array>)

But the model should be designed to have two modes, one is for very first variable initialization, the other is for loading numpy vairable

* how to load numpy from txt

>> array = np.loadtxt(<filename>)

* how to dump out numpy as txt

>> np.savetxt(<filename>)

[Using tensorflow: tf.train.Saver()]

* Step1: declare the saver

>> saver = tf.train.Saver()

The declaration of saver should be after the whole variable declaration, because it will only save the variable before it.

* Step2: save the whole session.  
  >> saver.save(<session\_variable>, <save\_file\_path/filename> )

1. It doesn’t actually save the file as <filename>. Instead it saves three files.
2. The filename usally use ‘.ckpt’ extension for check point.
3. The saved file names are…

Ex: >> saver.save(sess, ‘tmp.ckpt’)

The saved file: ‘tmp.ckpt.data-xxxx-of-xxxx, ‘tmp.ckpt.index’, ‘tmp.ckpt.meta’

* Step3: restore the session

>> saver.restore(<session\_variable>, <save\_file\_path/filename>)

The session does not need to initialize variable with initializer before restore operation.