

TensorFlow Cheat Sheet

```
In [1]: import os
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'
# os.environ['PYTHONWARNINGS'] = 'ignore'
```

```
In [2]: import tensorflow as tf
import numpy as np
```

```
In [3]: %precision 2
ipython_plain = get_ipython().display_formatter.formatters['text/plain']
ipython_plain.for_type(np.float64, ipython_plain.lookup_by_type(float));
```

tf.Tensor

```
In [4]: tf.constant(2, 'float32', [3])
```

Metal device set to: Apple M1 Max

```
Out[4]: <tf.Tensor: shape=(3,), dtype=float32, numpy=array([2., 2., 2.], dtype=float32)>
```

```
In [5]: tf.constant(np.random.uniform(size=[2,2]).astype('float32'))
```

```
Out[5]: <tf.Tensor: shape=(2, 2), dtype=float32, numpy=
array([[0.06, 0.09],
       [0.61, 0.82]], dtype=float32)>
```

```
In [6]: t3 = tf.range(3)
t3
```

```
Out[6]: <tf.Tensor: shape=(3,), dtype=int32, numpy=array([0, 1, 2], dtype=int32)>
```

```
In [7]: t23 = tf.constant(np.array([[1,2,3], [4,5,6]]))
t23
```

```
Out[7]: <tf.Tensor: shape=(2, 3), dtype=int64, numpy=
array([[1, 2, 3],
       [4, 5, 6]])>
```

tf.Variable

```
In [8]: with tf.device('/CPU:0'): # Apple TF-GPU bug
    var1 = tf.Variable(tf.constant(2, 'float32', [3]), name='var1')
var1
```

```
Out[8]: <tf.Variable 'var1:0' shape=(3,) dtype=float32, numpy=array([2., 2., 2.], dtype=float32)>
```

```
In [9]: var1.assign(var1 + 1)
var1
```

```
Out[9]: <tf.Variable 'var1:0' shape=(3,) dtype=float32, numpy=array([3., 3., 3.], dtype=float32)>
```

```
In [10]: var1[1]
```

```
Out[10]: <tf.Tensor: shape=(), dtype=float32, numpy=3.0>
```

```
In [11]: var1[1].assign(9)  
var1
```

```
Out[11]: <tf.Variable 'var1:0' shape=(3,) dtype=float32, numpy=array([3., 9., 3.], dtype=float32)>
```

tf.Operation

```
In [12]: t23
```

```
Out[12]: <tf.Tensor: shape=(2, 3), dtype=int64, numpy=  
array([[1, 2, 3],  
       [4, 5, 6]])>
```

```
In [13]: t23 + 1
```

```
Out[13]: <tf.Tensor: shape=(2, 3), dtype=int64, numpy=  
array([[2, 3, 4],  
       [5, 6, 7]])>
```

```
In [14]: t23 > 2
```

```
Out[14]: <tf.Tensor: shape=(2, 3), dtype=bool, numpy=  
array([[False, False,  True],  
       [ True,  True,  True]])>
```

```
In [15]: tf.reduce_sum(t23)
```

```
Out[15]: <tf.Tensor: shape=(), dtype=int64, numpy=21>
```

```
In [16]: tf.reduce_sum(t23, axis=0)
```

```
Out[16]: <tf.Tensor: shape=(3,), dtype=int64, numpy=array([5, 7, 9])>
```

```
In [17]: tf.reduce_sum(t23, axis=0, keepdims=True)
```

```
Out[17]: <tf.Tensor: shape=(1, 3), dtype=int64, numpy=array([[5, 7, 9]])>
```

```
In [18]: tf.reduce_max(t23)
```

```
Out[18]: <tf.Tensor: shape=(), dtype=int64, numpy=6>
```

```
In [19]: tf.reduce_mean(t23)
```

```
Out[19]: <tf.Tensor: shape=(), dtype=int64, numpy=3>
```

```
In [20]: tf.argmax(t23, axis=1)
```

```
Out[20]: <tf.Tensor: shape=(2,), dtype=int64, numpy=array([2, 2])>
```

```
In [21]: tf.cumsum(v3)
```

```
Out[21]: <tf.Tensor: shape=(3,), dtype=int64, numpy=array([1, 3, 6])>
```

```
In [22]: t32 = tf.transpose(t23)
         t32
```

```
Out[22]: <tf.Tensor: shape=(3, 2), dtype=int64, numpy=
         array([[1, 4],
                [2, 5],
                [3, 6]])>
```

```
In [23]: tf.matmul(t23, t32)
```

```
Out[23]: <tf.Tensor: shape=(2, 2), dtype=int64, numpy=
         array([[14, 32],
                [32, 77]])>
```

Reshaping

```
In [24]: tf.squeeze(tf.constant(0, 'int32', [3,2,1]))
```

```
Out[24]: <tf.Tensor: shape=(3, 2), dtype=int32, numpy=
         array([[0, 0],
                [0, 0],
                [0, 0]], dtype=int32)>
```

```
In [25]: tf.expand_dims(t23, -1).shape
```

```
Out[25]: TensorShape([2, 3, 1])
```

```
In [26]: tf.expand_dims(t23, 0).shape
```

```
Out[26]: TensorShape([1, 2, 3])
```

```
In [ ]:
```