

# 텐서플로의 난수 활용

# 텐서플로 난수

p36

## • 균등분포 난수

– `tf.random.uniform([1], 0, 1)`

• 배열, [시작, 끝)

```
tf.random.uniform(
    shape, minval=0, maxval=None,
    dtype=tf.dtypes.float32,
    seed=None, name=None
)
```

```
[6] 1 # 3.7 랜덤한 수 얻기 (균일 분포)
     2 rand = tf.random.uniform([1],0,1)
     3 print(rand)
```

```
↳ tf.Tensor([0.5543064], shape=(1,), dtype=float32)
```

```
[8] 1 rand = tf.random.uniform([5, 4],0,1)
     2 print(rand)
```

```
↳ tf.Tensor(
[[0.43681145 0.84187937 0.9562702 0.7846168 ]
 [0.6079582 0.95665395 0.9038415 0.19482386]
 [0.51012075 0.8609252 0.9433547 0.9636986 ]
 [0.2134043 0.9559026 0.5170028 0.4017253 ]
 [0.0141474 0.15949261 0.23697984 0.7221806 ]], shape=(5, 4), dtype=float32)
```

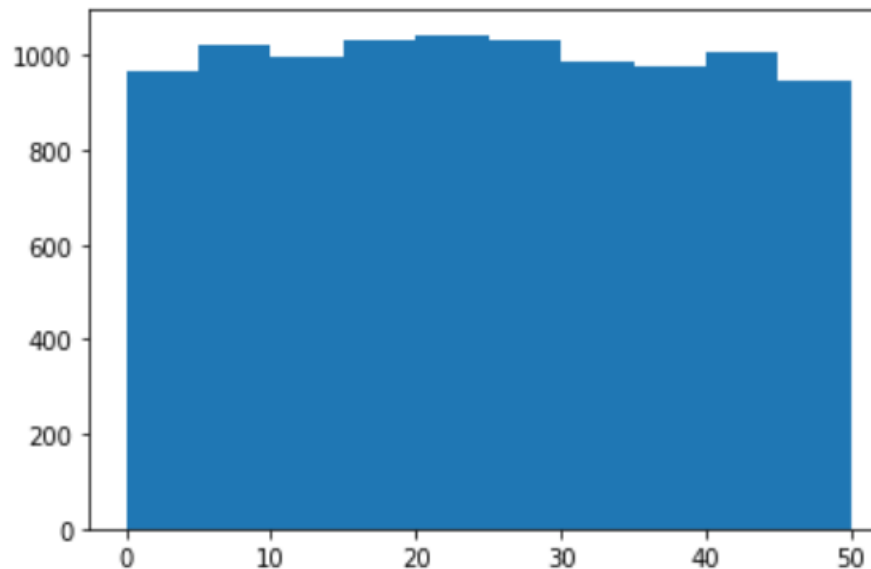
```
[11] 1 rand = tf.random.uniform([1000],0,10)
      2 print(rand[:10])
```

```
↳ tf.Tensor(
[5.1413307 1.548909 8.911686 9.880335 5.5388713 5.6710424 6.80269
 1.9444573 7.549943 6.573516 ], shape=(10,), dtype=float32)
```

# 균등 분포 1000개 그리기

```
[14] 1 import matplotlib.pyplot as plt
      2 rand = tf.random.uniform([10000],0,50)
      3 plt.hist(rand, bins=10)
```

```
↳ (array([ 965., 1020.,  994., 1032., 1043., 1030.,  987.,  976., 1008.,
          945.]),
    array([6.0796738e-04, 4.9998469e+00, 9.9990854e+00, 1.4998324e+01,
          1.9997562e+01, 2.4996801e+01, 2.9996040e+01, 3.4995281e+01,
          3.9994518e+01, 4.4993759e+01, 4.9992996e+01], dtype=float32),
    <a list of 10 Patch objects>)
```



# 정규분포 난수

- **tf.random.normal([4],0,1)**
  - 크기, 평균, 표준편차

```
[53] 1 # 3.9 랜덤한 수 여러 개 얻기 (정규 분포)
      2 rand = tf.random.normal([4],0,1)
      3 print(rand)
```

```
↳ tf.Tensor([-0.5962639  0.47093895  1.9455601 -0.42773333], shape=(4,), dtype=float32)
```

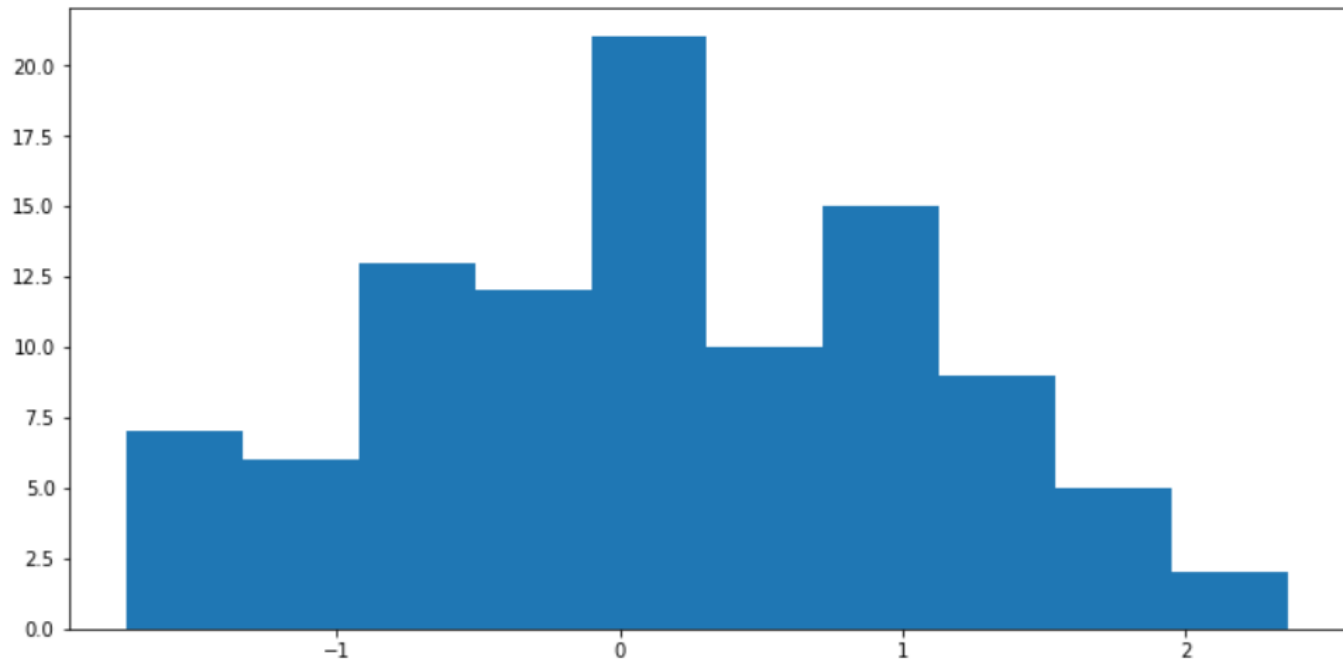
```
[54] 1 # 3.9 랜덤한 수 여러 개 얻기 (정규 분포)
      2 rand = tf.random.normal([2, 4],0,2)
      3 print(rand)
```

```
↳ tf.Tensor(
[[[-2.145662  0.64699423  2.0760484 -1.4640687 ]
 [ 1.3588632 -0.9740333  1.4347676 -1.3747462 ]], shape=(2, 4), dtype=float32)
```

# 정규 분포 100개 그리기

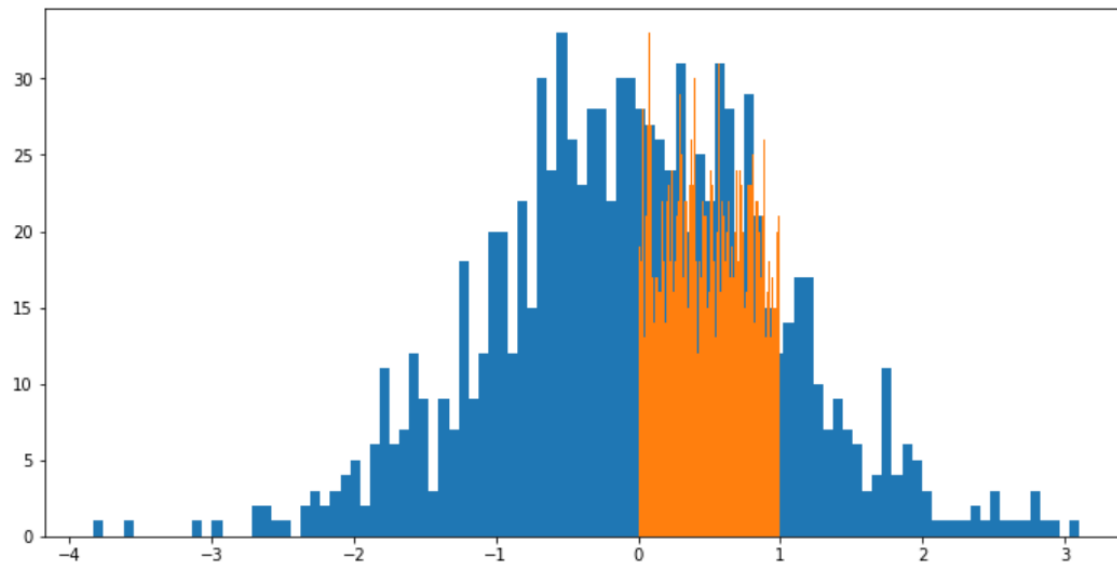
```
[52] 1 import matplotlib.pyplot as plt  
      2 rand = tf.random.normal([100], 0, 1)  
      3 plt.hist(rand, bins=10)
```

```
↳ (array([ 7.,  6., 13., 12., 21., 10., 15.,  9.,  5.,  2.]),  
    array([-1.7424849, -1.332153, -0.921821, -0.511489, -0.10115702,  
          0.30917495,  0.7195069,  1.129839,  1.5401709,  1.9505029,  
          2.3608348 ], dtype=float32),  
    <a list of 10 Patch objects>)
```



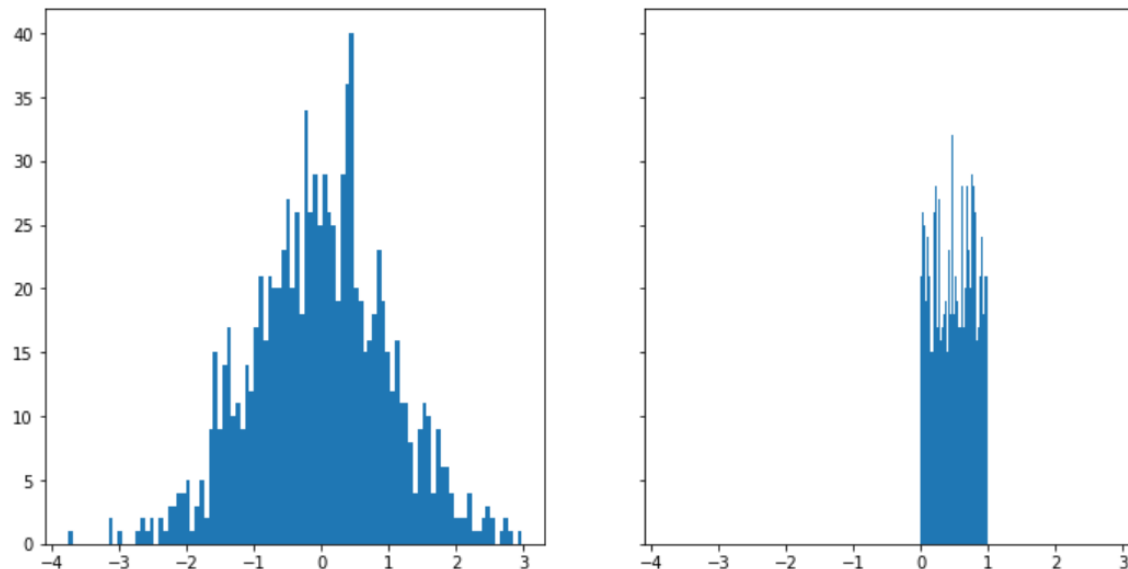
# 균등분포와 정규분포의 비교

```
[57] 1 import matplotlib.pyplot as plt  
      2 rand1 = tf.random.normal([1000], 0, 1)  
      3 rand2 = tf.random.uniform([2000], 0, 1)  
      4 plt.hist(rand1, bins=100)  
      5 plt.hist(rand2, bins=100)
```



# 균등분포와 정규분포를 부분으로 그리기

```
1 import matplotlib.pyplot as plt
2 rand1 = tf.random.normal([1000],0, 1)
3 rand2 = tf.random.uniform([2000], 0, 1)
4
5 plt.rcParams["figure.figsize"] = (12,6)
6 fig, axes = plt.subplots(1, 2, sharex=True, sharey=True)
7 axes[0].hist(rand1, bins=100)
8 axes[1].hist(rand2, bins=100)
```



# shuffle

- tf.random.shuffle(a)**

```
[29]  1 import numpy as np
      2 a = np.arange(10)
      3 print(a)
      4 tf.random.shuffle(a)
```

```
↳ [0 1 2 3 4 5 6 7 8 9]
   <tf.Tensor: shape=(10,), dtype=int64, numpy=array([7, 9, 1, 4, 3, 5, 8, 6, 2, 0])>
```

```
[26]  1 import numpy as np
      2 a = np.arange(20).reshape(4, 5)
      3 a
```

```
↳ array([[ 0,  1,  2,  3,  4],
          [ 5,  6,  7,  8,  9],
          [10, 11, 12, 13, 14],
          [15, 16, 17, 18, 19]])
```

```
[27]  1 tf.random.shuffle(a)
```

```
↳ <tf.Tensor: shape=(4, 5), dtype=int64, numpy=
   array([[ 0,  1,  2,  3,  4],
          [15, 16, 17, 18, 19],
          [10, 11, 12, 13, 14],
          [ 5,  6,  7,  8,  9]])>
```