

# Python小练习：从正态分布中采样

作者：凯鲁嘎吉 - 博客园 <http://www.cnblogs.com/kailugaji/>

本文用Python实现三种从正态(高斯)分布中的采样方式：确定性采样、重参数化技巧(推荐)、直接采样。

## 1. normal\_test.py

```
1 # -*- coding: utf-8 -*-
2 # Author: 凯鲁嘎吉 Coral Gajic
3 # https://www.cnblogs.com/kailugaji/
4 # Python小练习：从正态分布中采样
5 import torch
6 from torch.distributions import Normal
7 def my_normal(means, std, str):
8     dist = Normal(means, std)
9     # Normal(loc: torch.Size([10]), scale: torch.Size([10]))
10    if str == 'deterministic':
11        out_pred = means
12        # 如果是确定性采样，均值就是输出
13    elif str == 'reparameterize':
14        # 重参数化技巧
15        out_pred = dist.rsample()
16        # rsample() 不是在定义的正态分布上采样
17        # 而是mean+std×采样值eps，其中eps~N(0, I)
18    else:
19        out_pred = dist.sample()
20        # 直接从定义的正态分布（均值为mean，标准差std）上采样
21    return out_pred
22
23 str = ['deterministic', 'reparameterize', 'sample']
24 means = torch.arange(1, 11) * 1.0
25 # tensor([ 1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10.])
26 std = torch.arange(1, 0, -0.1)
27 # tensor([1.0000, 0.9000, 0.8000, 0.7000, 0.6000, 0.5000, 0.4000, 0.3000, 0.2000, 0.1000])
28 print('均值: \n', means)
29 print('方差: \n', std)
30 for i in str:
31     out_pred = my_normal(means, std, i)
32     print(i, '下的采样结果: \n', out_pred)
```

## 2. 结果

```
D:\ProgramData\Anaconda3\python.exe "D:/Python code/2023.3 exercise/Other/normal_test.py"
```

均值:

```
tensor([ 1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10.])
```

方差:

```
tensor([1.0000, 0.9000, 0.8000, 0.7000, 0.6000, 0.5000, 0.4000, 0.3000, 0.2000,
        0.1000])
```

deterministic 下的采样结果:

```
tensor([ 1.,  2.,  3.,  4.,  5.,  6.,  7.,  8.,  9., 10.])
```

reparameterize 下的采样结果:

```
tensor([1.7890, 1.6882, 2.9405, 3.5366, 4.5679, 6.0867, 6.7917, 8.0406, 9.0113,
        9.9845])
```

sample 下的采样结果:

```
tensor([0.4161, 1.5916, 3.0199, 3.5160, 4.6002, 5.7837, 7.1503, 8.2893, 8.7639,
        9.9661])
```

```
Process finished with exit code 0
```

注意：每次采样结果是不一样的。