

在使用PyTorch做实验时经常会用到生成随机数Tensor的方法，比如：

- `torch.rand()`
- `torch.randn()`
- `torch.normal()`
- `torch.linspace()`

在很长一段时间里我都没有区分这些方法生成的随机数究竟有什么不同，由此在做实验的时候经常会引起一些莫名其妙的麻烦。

所以在此做一个总结，以供大家阅读区分，不要重蹈我的覆辙。

### 均匀分布

`torch.rand(*sizes, out=None)` → Tensor

返回一个张量，包含了从区间[0, 1)的均匀分布中抽取的一组随机数。张量的形状由参数sizes定义。

参数：

- `sizes (int...)` - 整数序列，定义了输出张量的形状
- `out (Tensor, optional)` - 结果张量

例子：

```
torch.rand(2, 3)
0.0836 0.6151 0.6958
0.6998 0.2560 0.0139
[torch.FloatTensor of size 2x3]
```

### 标准正态分布

`torch.randn(*sizes, out=None)` → Tensor

返回一个张量，包含了从标准正态分布（均值为0，方差为1，即高斯白噪声）中抽取的一组随机数。张量的形状由参数sizes定义。

参数：

- `sizes (int...)` - 整数序列，定义了输出张量的形状
- `out (Tensor, optional)` - 结果张量

例子：

```
torch.randn(2, 3)
0.5419 0.1594 -0.0413
```

```
-2.7937 0.9534 0.4561  
[torch.FloatTensor of size 2x3]
```

离散正态分布

```
torch.normal(means, std, out=None) → Tensor
```

返回一个张量，包含了从指定均值means和标准差std的离散正态分布中抽取的一组随机数。

标准差std是一个张量，包含每个输出元素相关的正态分布标准差。

参数：

- means (float, optional) - 均值
- std (Tensor) - 标准差
- out (Tensor) - 输出张量

例子：

```
torch.normal(mean=0.5, std=torch.arange(1, 6))  
-0.1505  
-1.2949  
-4.4880  
-0.5697  
-0.8996  
[torch.FloatTensor of size 5]
```

线性间距向量

```
torch.linspace(start, end, steps=100, out=None) → Tensor
```

返回一个1维张量，包含在区间start和end上均匀间隔的step个点。

输出张量的长度由steps决定。

参数：

- start (float) - 区间的起始点
- end (float) - 区间的终点
- steps (int) - 在start和end间生成的样本数
- out (Tensor, optional) - 结果张量

例子：

```
torch.linspace(3, 10, steps=5)  
3.0000  
4.7500  
6.5000  
8.2500
```

10.0000

[torch.FloatTensor of size 5]