



## 法律声明

本课件包括演示文稿、示例、代码、题库、视频和声音等内容，深度之眼和讲师拥有完全知识产权；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或者机构不得盗版、复制、仿造其中的创意和内容，我们保留一切通过法律手段追究违反者的权利。

### 课程详情请咨询

- 微信公众号：深度之眼
- 客服微信号：deepshare0920



公众号



微信

关注公众号深度之眼，后台回复论文，获取60篇AI必读经典论文





deepshare.net

深度之眼

# PyTorch的Tensor (张量)

导师：余老师

---

关注公众号深度之眼，后台回复论文，获取60篇AI必读经典论文



# 目录

1/ Tensor概念

2/ Tensor创建一：直接创建

3/ Tensor创建二：依据数值创建

4/ Tensor创建三：依据概率创建



# 张量是什么？

What is Tensor

---



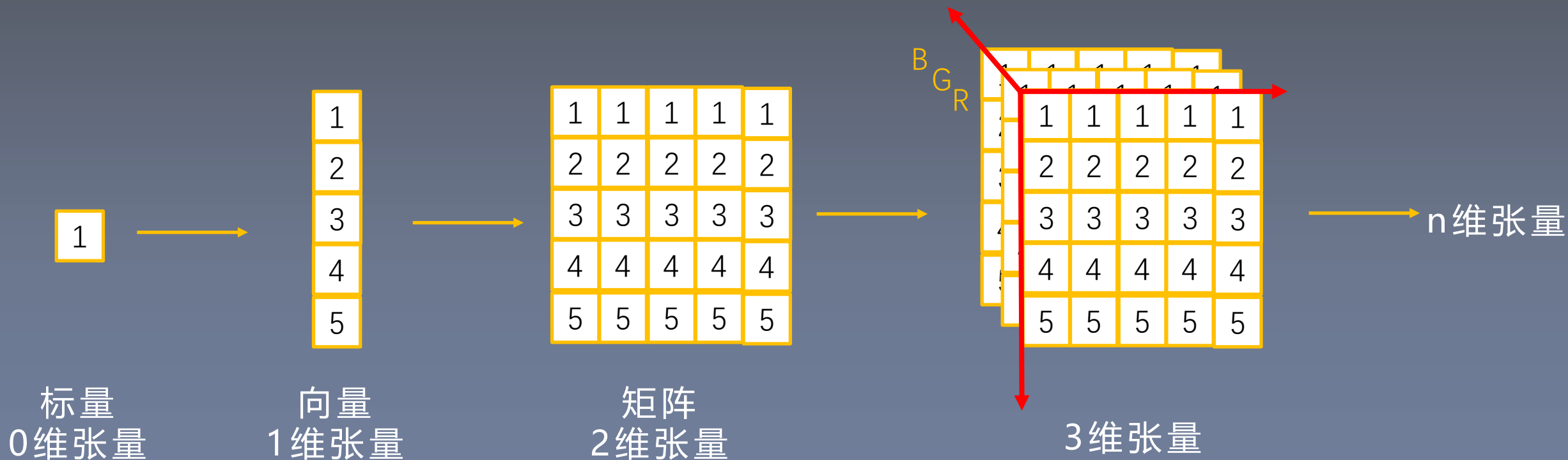
# 张量是什么？

What is Tensor?



deepshare.net

深度之眼



张量是一个**多维数组**，它是标量、向量、矩阵的高维拓展

关注公众号深度之眼，后台回复论文，获取60篇AI必读经典论文

# Tensor与Variable

Tensor and Variable



deepshare.net

深度之眼

Variable是torch.autograd中的数据类型

主要用于封装Tensor，进行自动求导

**data**: 被包装的Tensor

**grad**: data的梯度

**grad\_fn**: 创建Tensor的Function，是自动求导的关键

**requires\_grad**: 指示是否需要梯度

**is\_leaf**: 指示是否是叶子结点（张量）

torch.autograd.Variable

data

grad

grad\_fn

requires\_grad

is\_leaf

# Tensor

Tensor

PyTorch0.4.0版开始

**dtype**: 张量的数据类型

**shape**: 张量的形状,

**device**: 张量所在设备

data

require

| Data type                | dtype   | CPU tensor                      | GPU tensor                           |
|--------------------------|---|---------------------------------|--------------------------------------|
| 32-bit floating point    | <code>torch.float32</code> or <code>torch.float</code>  | <code>torch.FloatTensor</code>  | <code>torch.cuda.FloatTensor</code>  |
| 64-bit floating point    | <code>torch.float64</code> or <code>torch.double</code> | <code>torch.DoubleTensor</code> | <code>torch.cuda.DoubleTensor</code> |
| 16-bit floating point    | <code>torch.float16</code> or <code>torch.half</code>   | <code>torch.HalfTensor</code>   | <code>torch.cuda.HalfTensor</code>   |
| 8-bit integer (unsigned) | <code>torch.uint8</code>                                | <code>torch.ByteTensor</code>   | <code>torch.cuda.ByteTensor</code>   |
| 8-bit integer (signed)   | <code>torch.int8</code>                                 | <code>torch.CharTensor</code>   | <code>torch.cuda.CharTensor</code>   |
| 16-bit integer (signed)  | <code>torch.int16</code> or <code>torch.short</code>    | <code>torch.ShortTensor</code>  | <code>torch.cuda.ShortTensor</code>  |
| 32-bit integer (signed)  | <code>torch.int32</code> or <code>torch.int</code>      | <code>torch.IntTensor</code>    | <code>torch.cuda.IntTensor</code>    |
| 64-bit integer (signed)  | <code>torch.int64</code> or <code>torch.long</code>     | <code>torch.LongTensor</code>   | <code>torch.cuda.LongTensor</code>   |
| Boolean                  | <code>torch.bool</code>                                 | <code>torch.BoolTensor</code>   | <code>torch.cuda.BoolTensor</code>   |



deepshare.net

深度之眼

FloatTensor

关注公众号深度之眼，后台回复论文，获取60篇AI必读经典论文



# 张量的创建

Create Tensor

---



# 目录

1/ Tensor概念

2/ Tensor创建一：直接创建

3/ Tensor创建二：依据数值创建

4/ Tensor创建三：依据概率创建



# 张量的创建

Create Tensor

## 一、直接创建

`torch.tensor()`

功能：从data创建tensor

- **data**: 数据, 可以是list, numpy
- **dtype**: 数据类型, 默认与data的一致
- **device**: 所在设备, cuda/cpu
- **requires\_grad**: 是否需要梯度
- **pin\_memory**: 是否存于锁页内存

```
torch.tensor(  
    data,  
    dtype=None,  
    device=None,  
    requires_grad=False,  
    pin_memory=False)
```



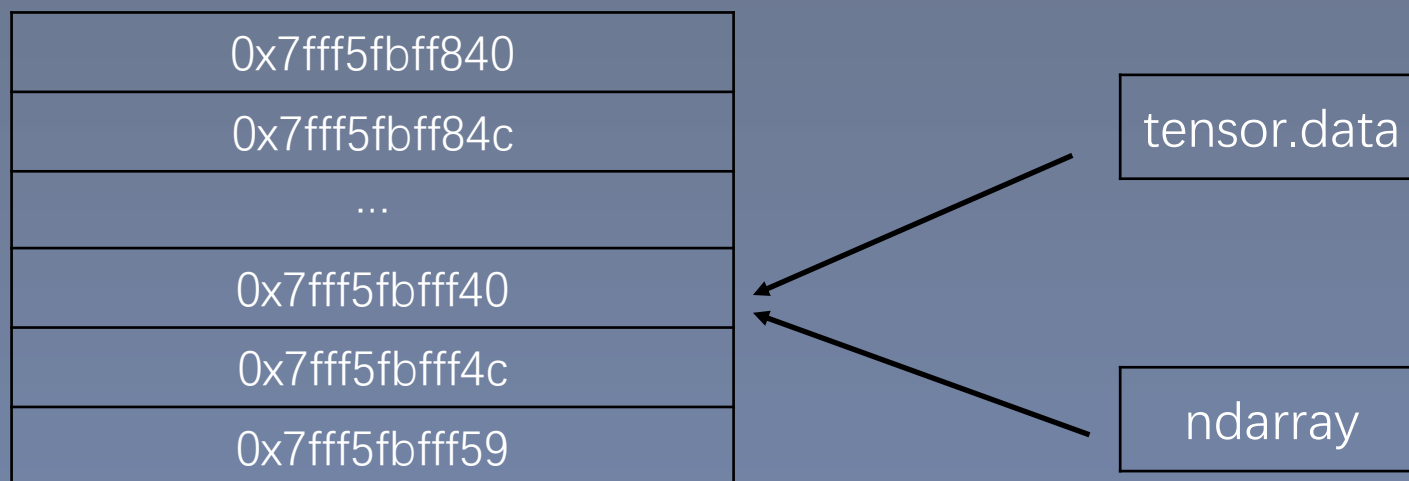
# 张量的创建

Create Tensor

```
torch.from_numpy(ndarray)
```

功能：从numpy创建tensor

**注意事项：**从torch.from\_numpy创建的tensor于原ndarray共享内存，当修改其中一个的数据，另外一个也将会被改动



关注公众号深度之眼，后台回复论文，获取60篇AI必读经典论文



# 张量的创建

Create Tensor

## 二、依据数值创建

### 2.1 torch.zeros()

功能：依size创建全0张量

- **size**: 张量的形状, 如(3, 3)、(3, 224,224)
- **out** : 输出的张量
- **layout** : 内存中布局形式, 有strided,sparse\_coo等
- **device** : 所在设备, gpu/cpu
- **requires\_grad**: 是否需要梯度

```
torch.zeros(*size,  
            out=None,  
            dtype=None,  
            layout=torch.strided,  
            device=None,  
            requires_grad=False)
```

# 张量的创建

Create Tensor



## 二、依据数值创建

### 2.2 torch.zeros\_like()

功能：依input形状创建全0张量

- **input**: 创建与input同形状的全0张量
- **dtype** : 数据类型
- **layout** : 内存中布局形式

```
torch.zeros_like(input,  
                  dtype=None,  
                  layout=None,  
                  device=None,  
                  requires_grad=False)
```



# 张量的创建

Create Tensor



## 二、依据数值创建

2.3 torch.ones()

2.4 torch.ones\_like()

功能：依input形状创建全1张量

- **size**: 张量的形状, 如(3, 3)、(3, 224,224)
- **dtype** : 数据类型
- **layout** : 内存中布局形式
- **device** : 所在设备, gpu/cpu
- **requires\_grad**: 是否需要梯度

```
torch.ones(*size,  
           out=None,  
           dtype=None,  
           layout=torch.strided,  
           device=None,  
           requires_grad=False)  
  
torch.ones_like(input,  
                dtype=None,  
                layout=None,  
                device=None,  
                requires_grad=False)
```

# 张量的创建

Create Tensor



deepshare.net

深度之眼

## 二、依据数值创建

2.5 torch.full()

2.6 torch.full\_like()

功能：依input形状创建指定数据的张量

- **size**: 张量的形状, 如(3, 3)
- **fill\_value**: 张量的值

```
torch.full(size,  
            fill_value,  
            out=None,  
            dtype=None,  
            layout=torch.strided,  
            device=None,  
            requires_grad=False)
```



# 张量的创建

Create Tensor

## 二、依据数值创建

### 2.7 torch.arange()

功能：创建等差的1维张量

注意事项：数值区间为[start, end)

- **start**: 数列起始值
- **end** : 数列 “结束值”
- **step**: 数列公差，默认为1

```
torch.arange(start=0,  
             end,  
             step=1,  
             out=None,  
             dtype=None,  
             layout=torch.strided,  
             device=None,  
             requires_grad=False)
```

# 张量的创建

Create Tensor



## 二、依据数值创建

### 2.8 torch.linspace()

功能：创建均分的1维张量

注意事项：数值区间为[start, end]

- **start**: 数列起始值
- **end** : 数列结束值
- **steps**: 数列长度

```
torch.linspace(start,  
               end,  
               steps=100,  
               out=None,  
               dtype=None,  
               layout=torch.strided,  
               device=None,  
               requires_grad=False)
```



# 张量的创建

Create Tensor

## 二、依据数值创建

### 2.9 torch.logspace()

功能：创建对数均分的1维张量

注意事项：长度为steps, 底为base

- **start**: 数列起始值
- **end** : 数列结束值
- **steps**: 数列长度
- **base** : 对数函数的底, 默认为10

```
torch.logspace(start,  
               end,  
               steps=100,  
               base=10.0,  
               out=None,  
               dtype=None,  
               layout=torch.strided,  
               device=None,  
               requires_grad=False)
```



# 张量的创建

Create Tensor



## 二、依据数值创建

### 2.10 torch.eye()

功能：创建单位对角矩阵（2维张量）

注意事项：默认为方阵

- **n**: 矩阵行数
- **m**: 矩阵列数

```
torch.eye(n,  
          m=None,  
          out=None,  
          dtype=None,  
          layout=torch.strided,  
          device=None,  
          requires_grad=False)
```

# 张量的创建

Create Tensor



deepshare.net

深度之眼

## 三、依概率分布创建张量

### 3.1 torch.normal()

功能：生成正态分布（高斯分布）

- **mean** : 均值
- **std** : 标准差

```
torch.normal(mean,  
              std,  
              out=None)
```



# 张量的创建

Create Tensor



deepshare.net

深度之眼

## 三、依概率分布创建张量

### 3.1 torch.normal()

四种模式：

mean为标量，std为标量

mean为标量，std为张量

mean为张量，std为标量

mean为张量，std为张量

```
torch.normal(mean,  
             std,  
             out=None)
```

```
torch.normal(mean,  
             std,  
             size,  
             out=None)
```

# 张量的创建

Create Tensor



## 三、依概率分布创建张量

### 3.2 torch.randn()

### 3.3 torch.randn\_like()

功能：生成**标准正态分布**

- **size**：张量的形状

```
torch.randn(*size,  
             out=None,  
             dtype=None,  
             layout=torch.strided,  
             device=None,  
             requires_grad=False)
```



# 张量的创建

Create Tensor

## 三、依概率分布创建张量

### 3.4 torch.rand()

### 3.5 torch.rand\_like()

功能：在区间 $[0, 1)$ 上，生成**均匀分布**

### 3.6 torch.randint()

### 3.7 torch.randint\_like()

功能：区间 $[low, high)$ 生成**整数均匀分布**

- **size**：张量的形状

```
torch.rand(*size,  
           out=None,  
           dtype=None,  
           layout=torch.strided,  
           device=None,  
           requires_grad=False)  
torch.randint(low=0,  
              high,  
              size,  
              out=None,  
              dtype=None,  
              layout=torch.strided,  
              device=None,  
              requires_grad=False)
```

# 张量的创建

Create Tensor

## 三、依概率分布创建张量

### 3.8 torch.randperm()

功能：生成生成从0到n-1的随机排列

- **n** : 张量的长度

### 3.9 torch.bernoulli()

功能：以input为概率，生成伯努力分布  
(0-1分布，两点分布)

- **input** : 概率值

```
torch.randperm(n,  
               out=None,  
               dtype=torch.int64,  
               layout=torch.strided,  
               device=None,  
               requires_grad=False)  
  
torch.bernoulli(input,  
                *,  
                generator=None,  
                out=None)
```



# 目录

1/ Tensor概念

2/ Tensor创建一：直接创建

3/ Tensor创建二：依据数值创建

4/ Tensor创建三：依据概率创建



# ——结 语——

在这次课程中，介绍了PyTorch的基本数据结构——  
张量，以及它的创建方法

在下次课程中，我们将会学习张量的

**索引、切片、变换**



关注公众号深度之眼，后台回复论文，获取60篇AI必读经典论文





**deepshare.net**

深度之眼

联系我们：

电话：18001992849

邮箱：[service@deepshare.net](mailto:service@deepshare.net)

QQ：2677693114



公众号



客服微信



关注公众号深度之眼，后台回复论文，获取60篇AI必读经典论文