

深度学习框架与网络结构设计

荣耀学院

2022 年 07 月

第一版序言

自动微分

目 录

| | |
|------------------|-----|
| 第一版序言 | iii |
| 第一章 一个最简单的极值问题求解 | 1 |
| 第二章 向量方式的极值问题求解 | 3 |

第一章 一个最简单的极值问题求解

用 Pytorch 求解一个最简单的极值问题

$$y = \frac{1}{2}(x - 2)^2$$

```
import torch
import numpy as np

# x是一个张量Tensor，只有一个元素
x = torch.tensor(np.random.normal(0, 0.01, (1, 1)), dtype=torch.float32)
# 需要计算x的梯度，以便对x进行优化
x.requires_grad_(requires_grad=True)

# 学习速率
eta = 0.4
for i in range(15):
    print('x=', x.data.item())
    y = (x-2)**2/2
    # 反向计算梯度
    y.backward()
    # 根据梯度，对x进行优化
    x.data -= eta*x.grad
    # 对x的梯度进行清零，以便进入下一轮优化
    x.grad.data.zero_()
```

运行结果如下：

```
x= 0.007588282693177462
x= 0.8045529723167419
x= 1.2827317714691162
x= 1.5696390867233276
x= 1.7417834997177124
x= 1.8450701236724854
x= 1.9070420265197754
x= 1.9442251920700073
x= 1.9665350914001465
x= 1.9799211025238037
```

```
x= 1.987952709197998  
x= 1.9927716255187988  
x= 1.9956629276275635  
x= 1.997397780418396  
x= 1.9984387159347534
```

代码注释的非常清楚，不再解释。

第二章 向量方式的极值问题求解

假设有一个向量 $[x_0, x_1]$ ，求如下函数的极值：

$$y = x_0^2 + x_1^2 + 4x_0 + 5x_1 + 3$$

```
import torch
import numpy as np

x = torch.tensor(np.random.normal(0, 0.01, (1, 2)), dtype=torch.float32)
x.requires_grad_(requires_grad=True)

eta = 0.4
for i in range(10):
    print('x=', x)
    y = torch.mm(x, x.t())+4*x[0,0]+5*x[0,1]+3
    y.backward()
    x.data -= eta*x.grad
    x.grad.data.zero_()
```

运行结果如下：

```
x= tensor([[ -0.0040,  0.0005]], requires_grad=True)
x= tensor([[ -1.6008, -1.9999]], requires_grad=True)
x= tensor([[ -1.9202, -2.4000]], requires_grad=True)
x= tensor([[ -1.9840, -2.4800]], requires_grad=True)
x= tensor([[ -1.9968, -2.4960]], requires_grad=True)
x= tensor([[ -1.9994, -2.4992]], requires_grad=True)
x= tensor([[ -1.9999, -2.4998]], requires_grad=True)
x= tensor([[ -2.0000, -2.5000]], requires_grad=True)
x= tensor([[ -2.0000, -2.5000]], requires_grad=True)
x= tensor([[ -2.0000, -2.5000]], requires_grad=True)
```

由此可知，在 $x_0 = -2$ 和 $x_1 = -2.5$ 的时候，函数取得极小值。

其中，`torch.mm(x, x.t())` 是矩阵相乘。 x 是一个 1×2 的向量，它的传置是 `x.t()`，是个 2×1 的向量，它们相乘是一个标量。

注意：不要把向量、矩阵、张量看得很神秘，没有什么神迷的，它们只是数据的存放方式而已。计算的时候，把它们视为一堆有排列规则的标量就行了。

优化问题目标函数的值一定是标量，比如多输出模型会把所有输出的误差平方和累加起来作为目标函数。因此求解梯度必然是标量对标量、向量、矩阵、张量的求导。
