深度学习快速入门

10 非线性激活

POET

2024 年 2 月 16 日

1 非线性激活的作用

引入非线性关系：如果在神经网络中只使用线性操作（如线性加权和），整个网络就会变成一个大的线性函数，多个线性层的组合依然是一个线性变换。非线性激活函数（例如 sigmoid、tanh、ReLU 等）引入了非线性关系，允许网络学习和表示非线性的模式，这对于解决复杂任务非常关键.

1. **Sigmoid 函数（Logistic 函数）：**

$$\text{sigma}(x) = \frac{1}{1+e^{-x}} \tag{1}$$

作用：将输入映射到范围 (0, 1) 之间。在二元分类问题中常用作输出层的激活函数。但它在深层网络中容易引起梯度消失问题，因此在隐藏层中的使用相对较少

2. **Tanh 函数（双曲正切函数）：**

$$\tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{2}$$

3. **ReLU 函数（修正线性单元）：**

$$ReLU(x) = max(0, x) \tag{3}$$

作用：将输入映射到范围 (-1, 1) 之间，相比于 sigmoid 函数，tanh 函数的输出范围更广，可以将数据“居中”，有时用于隐藏层的激活函数，可以缓解梯度消失问题。

4. **Leaky ReLU 函数:**

$$LeakyReLU(x) = max(\text{alpha}\, x, x) \tag{4}$$

其中，α 是一个小的正数，通常取 0.01。Leaky ReLU 在负数输入时引入一个小的斜率，以防止神经元“死亡”

5. **Softmax 函数**

$$\text{Softmax}\,(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}} \tag{5}$$

作用：通常用于多类别分类问题的输出层。Softmax 函数将网络的原始输出转化为概率分布，使得每个类别的预测概率在 (0, 1) 范围内，并且所有类别的概率之和为 1。

2 Tensorboard 显示

```
import torch
import torchvision
from torch import nn
from torch.nn import ReLU
from torch.nn import Sigmoid
from torch.utils.data import DataLoader
from torch.utils.tensorboard import SummaryWriter

dataset = torchvision.datasets.CIFAR10("./dataset",train=False,transform=torchvision.transforms.ToTensor(),download=True)
dataloader = DataLoader(dataset, batch_size=64)

class Tudui(nn.Module):
    def __init__(self):
        super(Tudui, self).__init__()
        self.relu1 = ReLU()
        self.sigmoid1 = Sigmoid()

    def forward(self, input):
        output = self.sigmoid1(input)
        return output

tudui = Tudui()
writer = SummaryWriter("logs")
step = 0

for data in dataloader:
    imgs, targets = data
```

```
28            writer.add_images("input", imgs, step)
29            output = tudui(imgs)
30            writer.add_images("output", output, step)
31            step = step + 1
```