



卷积神经网络

作者: Calvin

QQ: 179209347

Mail: 179209347@qq.com

介绍

笔记简介:

- 面向对象: 深度学习初学者
- 依赖课程: **线性代数, 统计概率**, 优化理论, 图论, 离散数学, 微积分, 信息论

知乎专栏:

<https://zhuanlan.zhihu.com/p/693738275>

Github & Gitee 地址:

https://github.com/mymagicpower/AIAS/tree/main/deep_learning

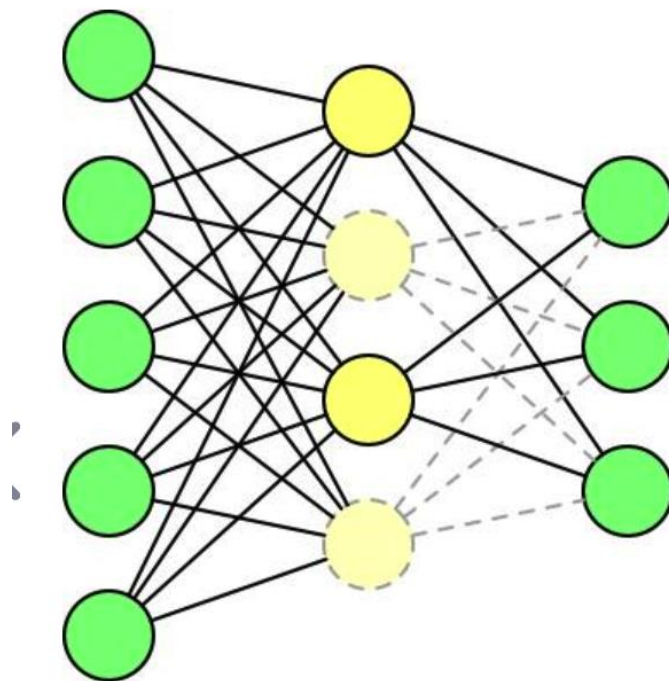
https://gitee.com/mymagicpower/AIAS/tree/main/deep_learning

* 版权声明:

- 仅限用于个人学习
- 禁止用于任何商业用途

Dropout方法

Dropout方法：在训练过程中，Dropout会随机地将神经网络中的一部分神经元的输出设置为0，即“丢弃”这些神经元，这样可以使得网络不依赖于任何一个特定的神经元，从而增强了网络的泛化能力。

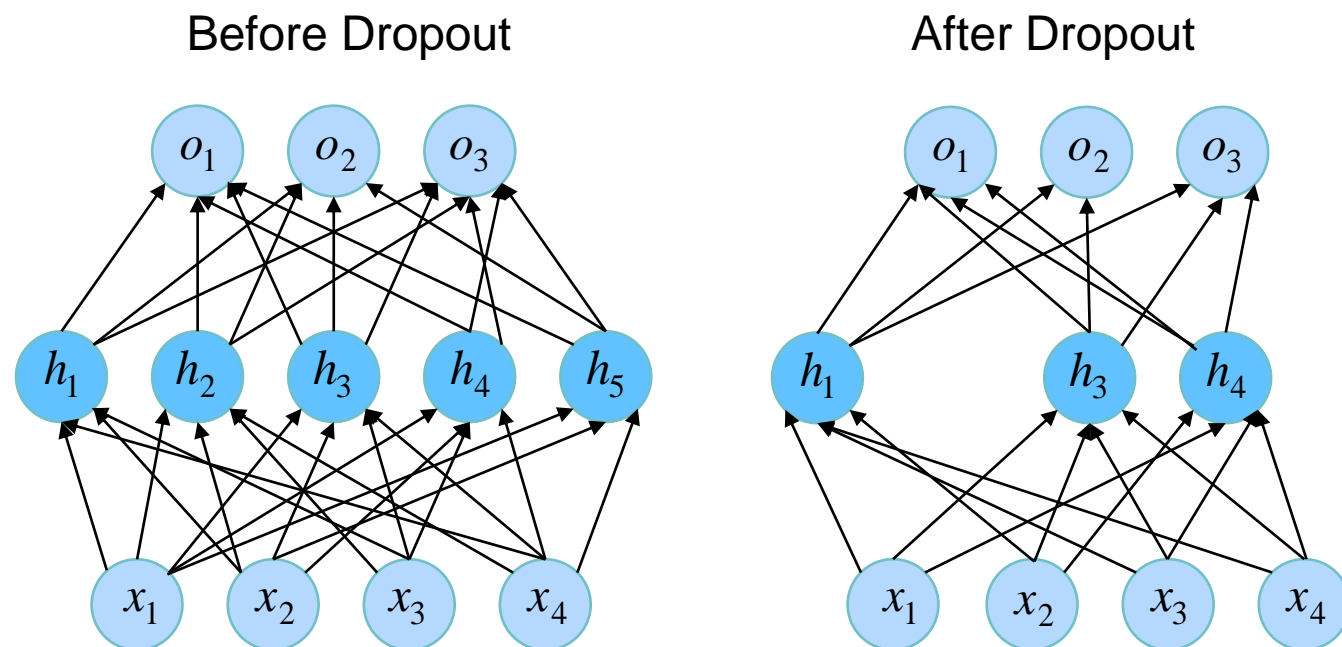


- 通常将丢弃法作用在隐藏全连接层的输出上
- 丢弃概率是控制模型复杂度的超参数

Dropout方法

以下是关于Dropout的一些关键特点和工作原理：

- **随机性**：在每次训练迭代中，Dropout会随机选择一些神经元丢弃。
- **训练和推理时的不同**：在训练过程中，Dropout会被应用，而在推理过程中通常不会使用Dropout，保证确定性的输出。
- **减少过拟合**：Dropout通过减少神经元之间的协作，迫使网络学习更加鲁棒和泛化的特征。
- **适用范围**：Dropout通常适用于全连接层和卷积层，但不适用于循环神经网络（RNN）等具有记忆性的结构。



- 在训练过程中随机丢弃单元（以及它们的连接）
- 每个单元以固定的丢弃率 p 保留，与其他单元独立
- 需要选择（调整）超参数 p ，通常会丢弃20%到50%的单元

Dropout方法

以下是一个简单的示例，展示了如何在 PyTorch 中实现一个自定义的 Dropout 类：

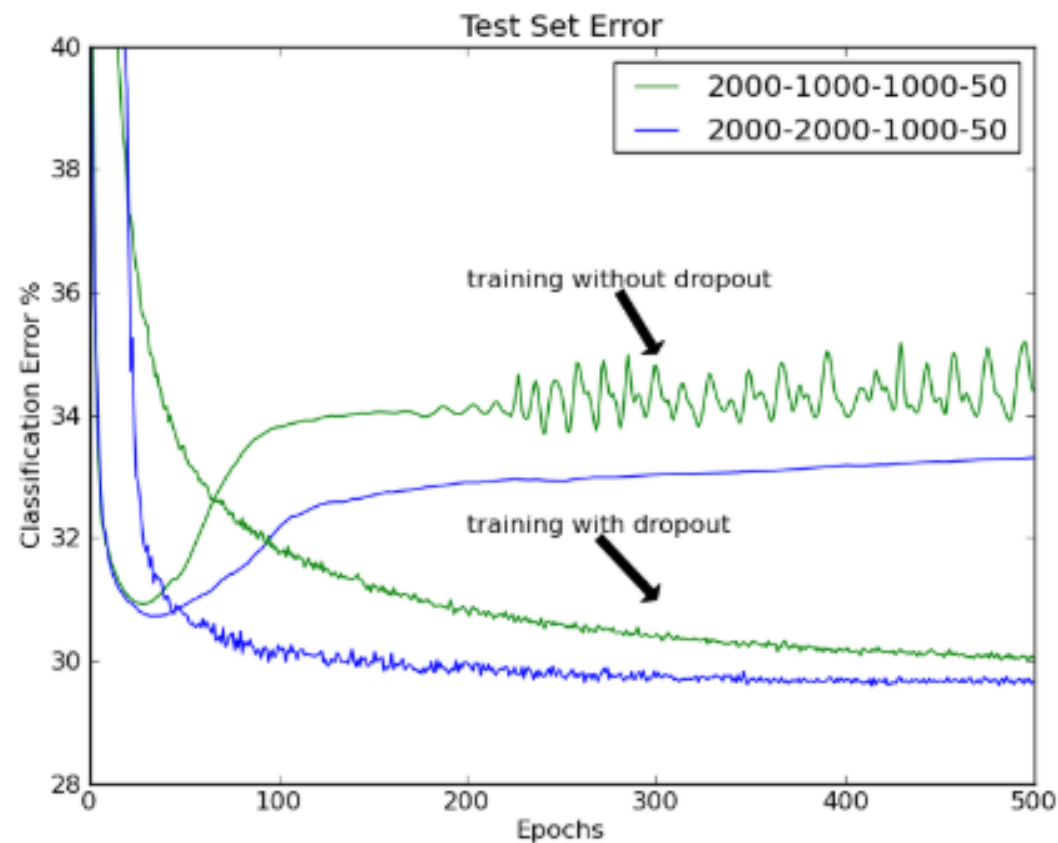
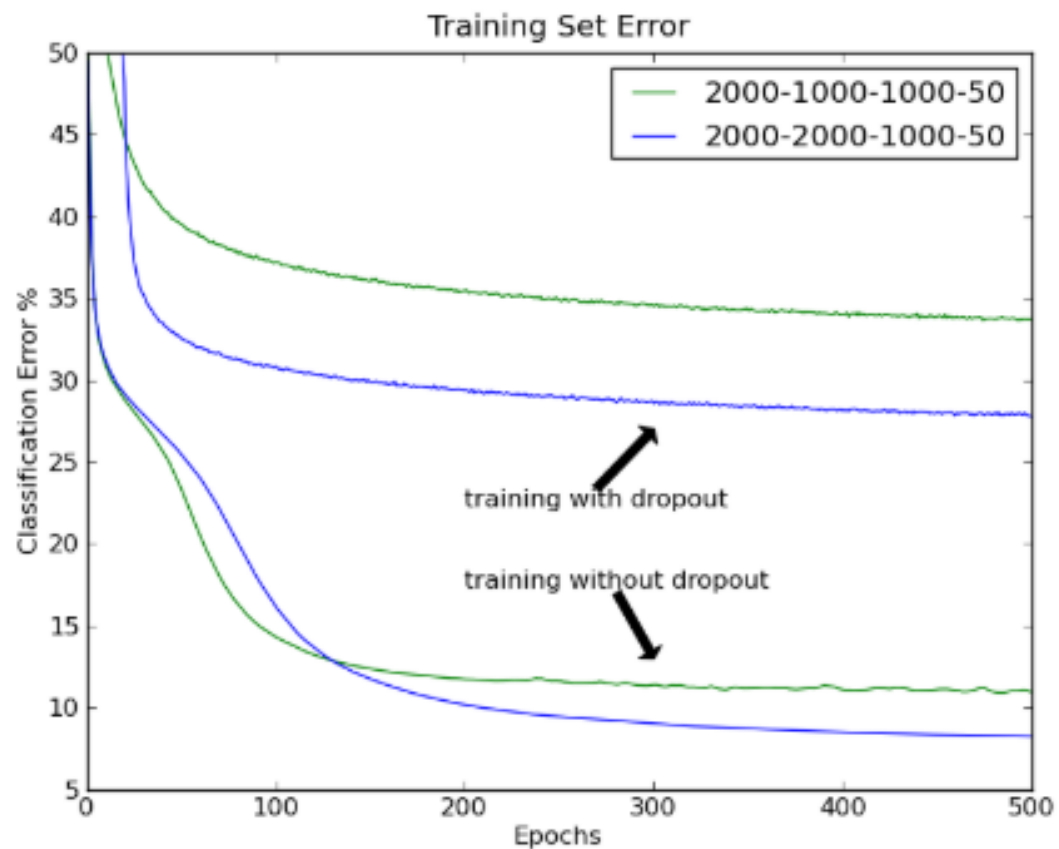
```
class CustomDropout(nn.Module):
    def __init__(self, p=0.5):
        super(CustomDropout, self).__init__()
        self.p = p

    def forward(self, x):
        if not self.training:
            return x

        mask = torch.bernoulli(torch.full_like(x, 1 - self.p))
        mask = mask / (1 - self.p)
        return x * mask

# 使用自定义的 Dropout 类
dropout = CustomDropout(p=0.5)
```

Dropout方法





Thank

You