

深度学习

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

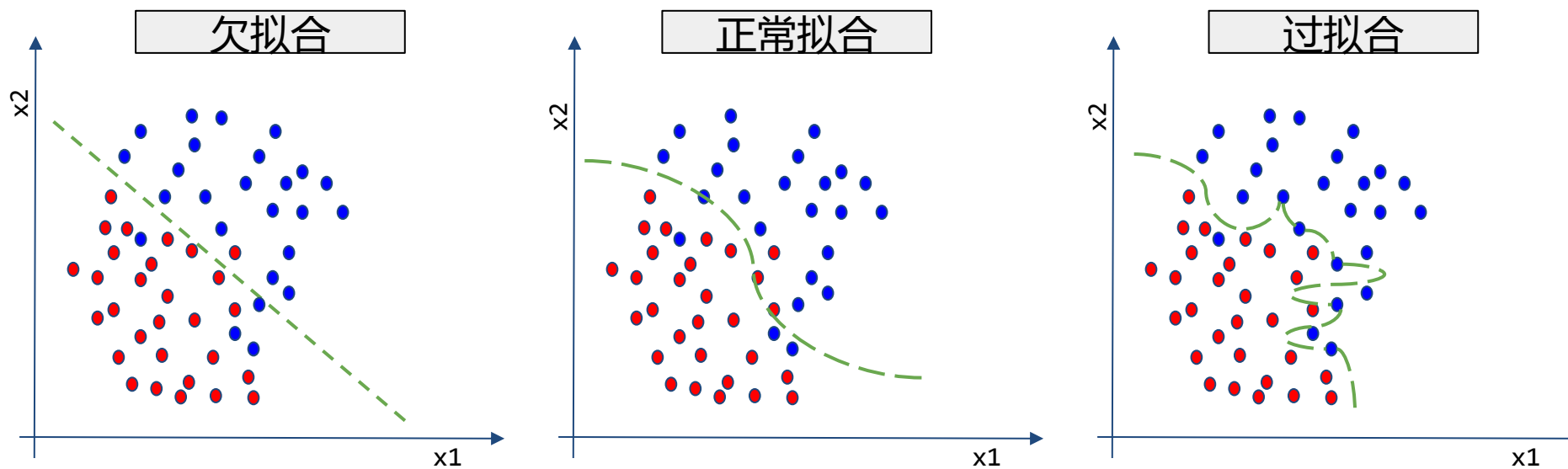
* 版权声明:

- 仅限用于个人学习
- 禁止用于任何商业用途

欠拟合与过拟合

在深度学习中，不仅要求网络模型对训练数据集有很好的拟合结果（训练误差小），同时也希望它可以对未知数据集（验证集和测试集）有很好的拟合结果（泛化能力），所产生的预测误差被称为泛化误差。

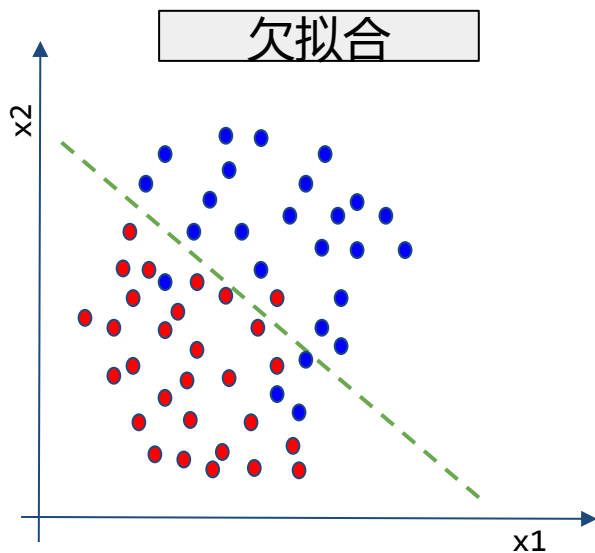
网络模型的拟合能力分为欠拟合、正常拟合和过拟合，如图所示。



欠拟合与过拟合

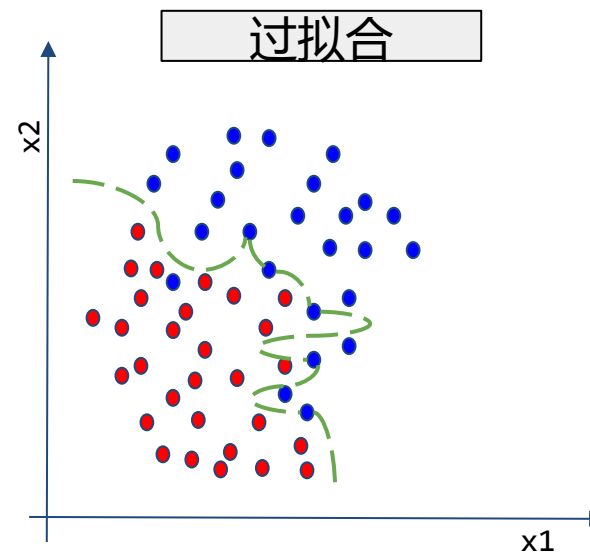
欠拟合

- 欠拟合的表现通常是训练集、验证集和测试集上的表现都较差，无法很好地捕捉数据的特征和模式。
- 解决欠拟合问题的方法包括增加模型复杂度、增加特征数量、调整超参数等。



过拟合

- 过拟合指的是模型在训练集上表现很好，但在验证集和测试集上表现较差。这是因为模型过度地学习了训练数据的噪声和细节，导致其泛化能力差。
- 解决过拟合问题的方法包括增加训练数据、使用正则化方法（如L1正则化、L2正则化）、减少模型复杂度、使用更简单的模型结构等。



过拟合处理方法 – L_1 正则化

在损失函数中加入 L_1 正则项或 L_2 正则项，可对模型的复杂度进行惩罚，以此来减小模型在训练数据集上的过拟合程度。正则项权重是控制模型复杂度的超参数。

(1) L_1 正则项是将网络模型的所有权重求绝对值再相加，公式如下：

$$\sum_{i=0}^{n-1} |w_i| = |w_0| + |w_1| + \dots + |w_{n-1}|$$

新的损失函数为原来的损失函数与 L_1 正则项的和。在相加时，需要通过指定原有损失函数与 L_1 正则项各自的权重，公式如下：

$$\begin{aligned} \text{Loss}(y, y') &= L(y, y') + \lambda \sum_{i=0}^{n-1} |w_i| \\ &= \frac{1}{n} \sum_{i=0}^{n-1} (y_i - y'_i)^2 + \lambda \sum_{i=0}^{n-1} |w_i| \end{aligned}$$

- $L(y, y')$ 为原损失函数(假设为均方误差损失函数)
- y_i 为第 i 个样本的真实值
- y'_i 为第 i 个样本的预测值
- λ 是超参数，需要人为指定

过拟合处理方法 – L₂ 正则化

L₂正则项是将网络模型的所有权重求平方和再求平方根后的结果。L₂正则项公式如下：

$$\sqrt{\sum_{i=0}^{n-1} w_i^2} = \sqrt{w_0^2 + w_1^2 + \cdots + w_{n-1}^2}$$

在应用L₂正则化防止过拟合时，需要将L₂正则项加入原有的损失函数中形成新的损失函数，公式如下：

$$\begin{aligned} \text{Loss}(y, y') &= L(y, y') + \lambda \sqrt{\sum_{i=0}^{n-1} w_i^2} \\ &= \frac{1}{n} \sum_{i=0}^{n-1} (y_i - y'_i)^2 + \lambda \sqrt{\sum_{i=0}^{n-1} w_i^2} \end{aligned}$$

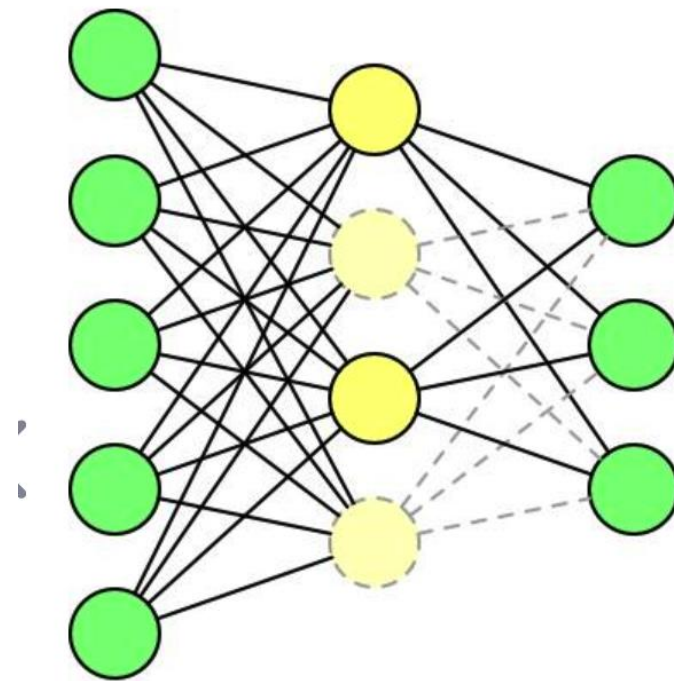
- $L(y, y')$ 为原损失函数(假设为均方误差损失函数)
- y_i 为第 i 个样本的真实值
- y'_i 为第 i 个样本的预测值
- λ 是超参数，需要人为指定

过拟合处理方法 – 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。

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。



- 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



Thank

You