大作业报告

FreezeOut 等相关技术

Xiao

2026 年 6 月 8 日

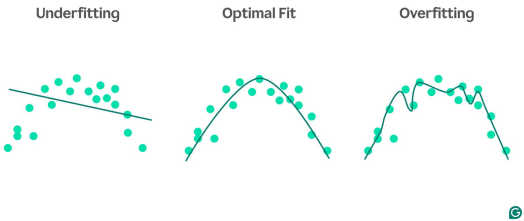
目录

过拟合问题

核心定义

过拟合是指模型在训练数据上表现极佳（甚至记住了噪声），但在测试数据上表现很差，泛化能力弱。

- **欠拟合**：模型太简单，没学到数据的真实规律（高偏置）
- **过拟合**：模型太复杂，连训练数据的噪声都学进去了（高方差）
- **刚刚好**：学到了数据的真实规律，能很好泛化



偏置（Bias）：预测值与真实值的差距。高偏置 = 欠拟合。
方差（Variance）：模型对训练数据微小变化的敏感度。高方差 = 过拟合。**典型表现**：训练损失持续下降，验证损失先降后升。

正则化技术

核心思想：奥卡姆剃刀

在能够正确解释现象的前提下，最简单的模型就是最好的模型。

从数学上看正则化：

$$L_{\text{总}} = \underbrace{\text{Loss}(y_{\text{true}}, y_{\text{pred}})}_{\text{拟合能力}} + \underbrace{\lambda R(W)}_{\text{惩罚项}}$$

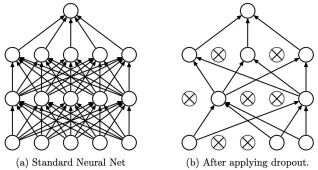
类别	代表方法	原理
参数惩罚	L1/L2 正则化	限制权重大小
结构惩罚	Dropout	随机删除神经元
数据增强	旋转/裁剪	制造更多训练数据
早停	Early Stopping	验证损失回升前停止

Dropout 概述

核心思想

训练时以概率 p 随机将神经元输出置为 0，每次前向传播都对应一个不同的子网络。

- **提出**：Hinton 等，2012 年（AlexNet 夺冠关键）
- **本质**：一种强大的正则化技术，用于防止过拟合
- **训练时**：输出 $= \begin{cases} x/(1-p) & \text{以概率} 1-p \\ 0 & \text{以概率} p \end{cases}$
- **测试时**：使用完整网络，不再丢弃



Dropout 示意图

Dropout 与 L2 正则化及量化效果

Dropout：破坏共适应（核心洞察）

- Hinton 观察到：神经元会相互依赖、协同工作
- 当部分神经元失效时，其余神经元会被动“补救”
- 易导致模型记忆训练集特有模式，泛化能力下降

L2 正则化（权重衰减）

$$R(W) = \|W\|_2^2 = \sum w_i^2$$

- 效果：权重整体缩小，不会归零
- 几何理解：在圆形约束域内求解最优值
- 作用：模型更平滑，降低输入扰动影响

Dropout 与 L2 正则化的联系

- Dropout 等价于自适应 L2 正则化
- 对和输入相关性高的参数施加更强约束

Dropout 实验效果

数据集	无 Dropout	有 Dropout	提升
MNIST	1.60% 错误	1.35% 错误	15.6%
CIFAR-10	19% 错误	16% 错误	15.8%
ImageNet	18.5% Top-5	15.8% Top-5	14.6%

数据来源：Srivastava et al.
Dropout: A Simple Way to Prevent Overfitting, JMLR 2014

BN 技术简介

内部协变量偏移（Internal Covariate Shift）

训练深度网络时，每一层的输入分布随前一层参数更新而变化。
层 1 参数更新 $\rightarrow$ 层 1 输出分布改变 $\rightarrow$ 层 2 输入分布改变 $\rightarrow$ 层 2 需要重新适应 $\rightarrow$ ...

分布的变化（可能很剧烈），导致下面三种问题：

- 需要极低的学习率
- 梯度消失/爆炸
- 对参数初始化敏感

Batch Normalization 的解决方案

标准化每一层的输入，使其保持均值 0、方差 1 的稳定分布：

$$\hat{x} = \frac{x - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}}, \quad y = \gamma \hat{x} + \beta$$

Dropout 与 BN 的冲突

为什么现代 CNN 中 Dropout 用得少?

Dropout 与 BatchNorm 存在**冲突**:

- 训练时，Dropout 随机关掉一半神经元，导致 BN 计算的批次均值和方差严重失准
- 测试时，Dropout 关闭，但 BN 使用全局统计量，导致训练/测试不一致
- **实验结果**：在已使用 BN 的 ResNet-50 上添加 Dropout，Top-1 准确率反而下降 1-2%

Transformer 中 Dropout 仍大量使用

- Transformer 使用 **LayerNorm**（对每个样本独立归一化），与 Dropout 无冲突
- 参数量极大（如 BERT-large 有 3.4 亿参数），需要 Dropout 防止过拟合
- Dropout 可应用于 Attention 矩阵、FFN 内部、残差连接前等多个位置

深度学习训练的过程

训练过程的五个步骤

1. **前向传播**：不断给数据，进行预测
2. **计算损失**：计算预测与真实值的差距，进行评分
3. **反向传播**：反过来，逐级算梯度
4. **更新参数**：按照梯度，统一调每一层的参数
5. **迭代**：重复 1-4 步，直到损失足够小

前向：x → 层 0 → 层 1 → 层 2 → ... → 层 L → loss
反向：loss → 层 L 梯度 → ... → 层 2 梯度 → 层 1 梯度 → 层 0 梯度 → 更新所有层

图: 一次完整的前向 + 反向传播

FreezeOut 概述：核心思想与动机

核心思想

训练过程中，按照预定策略**逐层冻结**网络浅层参数，停止其梯度更新，只做前向传播。

- **提出**：Brock 等，2017 年（OPT 研讨会）
- **目标**：**加速训练**（节省计算时间）
- **本质**：一种训练加速技术，而非正则化

核心洞察：早期层学得快（参数快速收敛，梯度趋于 0）→ 学完后不再需要大量更新 → 冻结它们以节省计算。

维度	早期层（底层）	后期层（高层）
位置	靠近输入	靠近输出
学习内容	边缘、纹理、短期模式	形状、语义、长期趋势
收敛速度	快（几轮后稳定）	慢（需要更多轮次）
计算量	大（特征图分辨率高）	小（特征图已压缩）

FreezeOut 的动机：早期层计算量大但收敛快，后期层计算量小但收敛慢。

FreezeOut 工作流程

1. **训练初期**：所有层正常参与前向 + 反向传播
2. **训练中期**：按调度策略，逐层将学习率退火到 0
3. **学习率 = 0 后**：将该层切换为 `requires_grad = False`
4. **继续训练**：已冻结层只做前向，不再反向传播；未冻结层正常更新

前向：x → 层 0（冻结）→ 层 1（冻结）→ 层 2 → ... → 层 L → loss
反向：loss → 层 L 梯度 → ... → 层 3 梯度 → （层 2，1、层 0 无梯度更新，参数不变）

图: FreezeOut 训练阶段的前向与反向传播

为什么能加速？——反向传播链变短

正常反向传播（L 层网络）：

$$\frac{\partial L}{\partial \theta_0} = \frac{\partial L}{\partial h_L} \cdot \frac{\partial h_L}{\partial h_{L-1}} \cdots \frac{\partial h_1}{\partial \theta_0}$$

FreezeOut 冻结前 k 层后：

$$\frac{\partial L}{\partial \theta_{k+1}} = \frac{\partial L}{\partial h_L} \cdot \frac{\partial h_L}{\partial h_{L-1}} \cdots \frac{\partial h_{k+1}}{\partial \theta_{k+1}}$$

- 第 0 到 k 层：`requires_grad = False`，不参与反向传播
- 第 $k+1$ 到 L 层：正常参与

加速来源

反向传播的**计算链长度**从 $L+1$ 减少到 $L-k+1$，省去了早期层的大量矩阵乘法。

两个指标

FreezeOut 需要指定：

1. **首次冻结时间** t_0（通常以 iteration 为单位）
2. **调度函数** $s(i)$（控制层之间的时间间隔）

Cubic Scaling 调度（论文推荐）

$$T_l = T_{\text{total}} \left[t_0 + (1 - t_0) \left(\frac{l}{L} \right)^3 \right]$$

- 早期层之间的冻结时间间隔较**小**（快速冻结）
- 后期层之间的间隔较**大**（给更多训练时间）
- 论文中使用 $t_0 = 0.9$，$T_{\text{total}} = 100$ epochs

FreezeOut vs. Dropout

技术对比

维度	Dropout	FreezeOut
主要目标	防止过拟合	加速训练
操作对象	单个神经元	整个层
作用时间	每次前向随机	训练中永久冻结
与 BN 兼容性	易冲突	完全兼容
本质	正则化技术	计算优化技术

FreezeOut 实验效果

- DenseNet：最高 20% 加速，错误率增加 $< 3\%$
- Wide ResNet：20% 加速，精度无损失
- VGG（无残差）：无明显加速（依赖残差结构）

TDA 概述

一句话定义

TDA 通过分析数据点云的**拓扑特征**（连通分支、环、空洞等），揭示数据底层结构。

- **核心思想**：忽略几何细节（距离、角度、曲率），只关注**同伦信息**（洞的数量、连通性）
- **与 Dropout/FreezeOut 的区别**：TDA 是**分析方法/监测工具**，而非训练策略
- **作用方式**：被动观测特征空间形状，不主动干预训练

几何 vs. 拓扑：

- 几何：关心形状的精确形态（胖瘦、高低）
- 拓扑：关心有几个头、几只手、几个洞

TDA 与前两种技术的本质区别

维度	Dropout	FreezeOut	TDA
技术类别	正则化	训练加速	分析/监测
核心目标	防止过拟合	减少计算时间	检测过拟合
操作对象	神经元	网络层	特征空间
是否修改模型	是	是	否（只读）
输出	更好的参数	更快的训练	信号 + 调控指令

在深度学习的工具箱中：

Dropout 是“让模型更强壮”，

FreezeOut 是“让训练更快”，

TDA 是“让监控更聪明”。

结束

谢谢！