3-2 正则化

王中雷

厦门大学王亚南经济研究院和经济学院, 2025

内容摘要

1. 参数惩罚

2. Dropout

3. 早停

直观

1．全连接神经网络的参数规模往往较大

- 复杂模型往往会对训练集进行过拟合
- 降低模型复杂度

2．当然，我们可以选择更为简单的神经网络模型

- 然而，简单的模型往往对数据进行欠拟合

3．我们希望在不改变模型框架的基础上，降低模型复杂度

直观

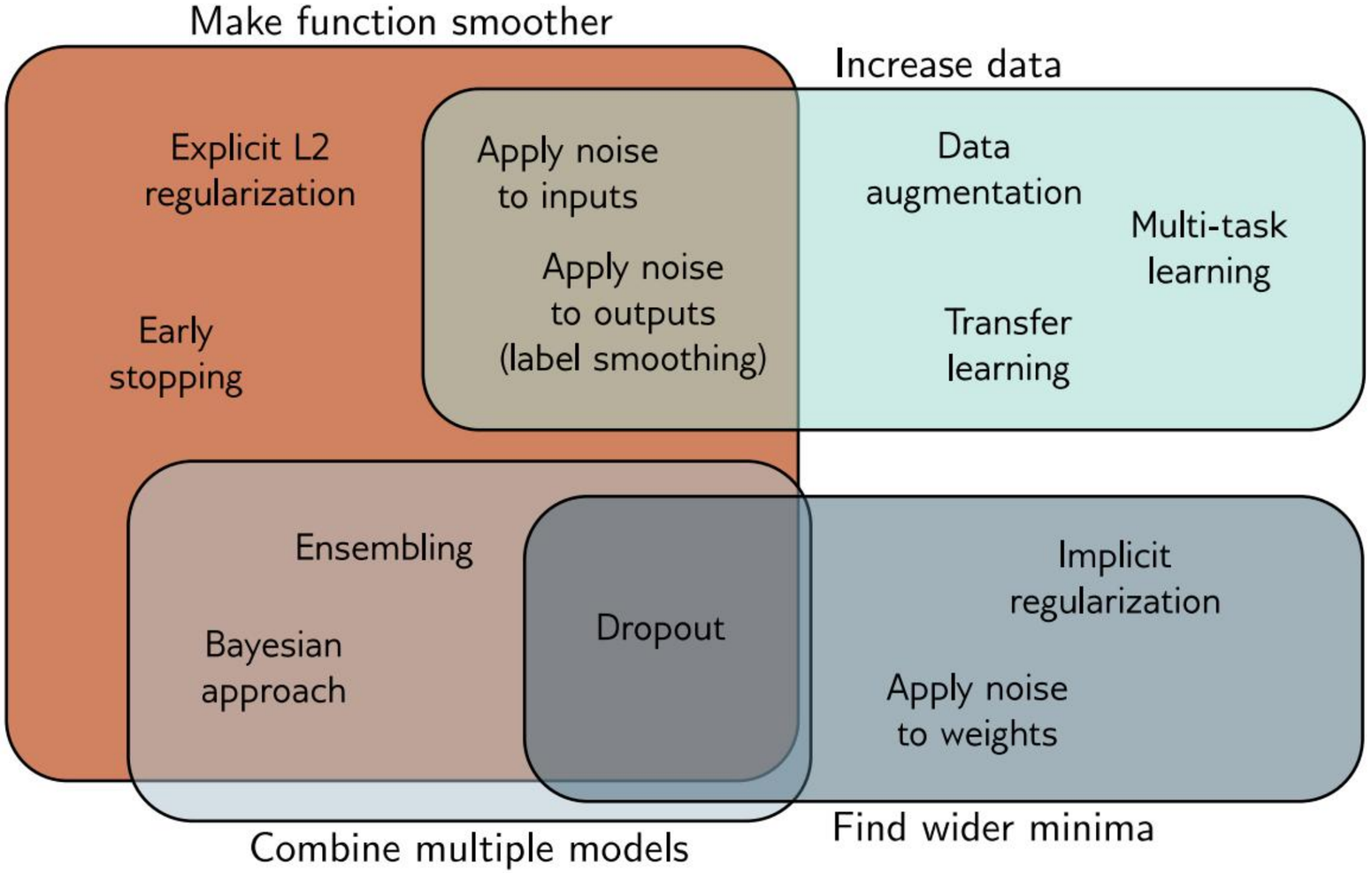
1．在机器学习领域，正则化

- 通常通过在代价函数上增加惩罚项来实现
- 能够提高深度学习模型的泛化性

2．正则化也包括 dropout、早停以及模型集成等

直观

1. 下图来自于 Prince (2024) 的图 9.14



对模型参数的惩罚

1. l_2-惩罚

$$\mathcal{J}_2 = \frac{\lambda}{2n} \sum_{l=1}^{L} \|\boldsymbol{W}^{[l]}\|_F^2$$

- λ : 超参
- $\|\boldsymbol{A}\|_F = (\sum_i \sum_j a_{ij}^2)^{1/2}$: 矩阵$\boldsymbol{A} = (a_{ij})$ 的 Frobenius 范数
- 导数结果

$$\frac{\partial \mathcal{J}_2}{\partial \boldsymbol{W}^{[l]}} = \frac{\lambda}{n} \boldsymbol{W}^{[l]}$$

- 类似于岭回归，用来控制模型复杂程度

对模型参数的惩罚

1. l_1-惩罚

$$\mathcal{J}_1 = \frac{\lambda}{n}\sum_{l=1}^{L}\|\boldsymbol{W}^{[l]}\|_1$$

- $\|\boldsymbol{A}\|_1 = \sum_i \sum_j |a_{ij}|$
- 导数结果

$$\frac{\partial \mathcal{J}_1}{\partial \boldsymbol{W}^{[l]}} = \frac{\lambda}{n}\text{sign}(\boldsymbol{W}^{[l]})$$

- 该惩罚常被用于
  - 控制模型复杂度
  - 类似于 LASSO，可实现模型参数的稀疏化

注意

1. 我们仅将l_2- 或者l_1-惩罚施加于权重项$\{\boldsymbol{W}^{[l]}: l=1,\ldots,L\}$
   - l_2- 或者l_1-惩罚直接添加在原始代价函数上
   - 我们不对偏置项 $\{\boldsymbol{b}^{[l]}: l=1,\ldots,L\}$ 施加任何惩罚
2. 我们 “错误地使用”$\|\boldsymbol{A}\|_1$表达矩阵$\boldsymbol{A}$ 元素绝对值的加和
   - 一般来讲，$\|\boldsymbol{A}\|_1=\max_j \sum_i |a_{ij}|$ 表示每列元素绝对值加和的最大值

Dropout

1．模型训练阶段常用的正则化方法

- 随机在隐藏层中删除神经元
- 如果某神经元被删除，那么它对应的激活值被设置为 0
- 删除对每个训练样本是独立随机进行的

2．提高模型的稳健性和泛化能力

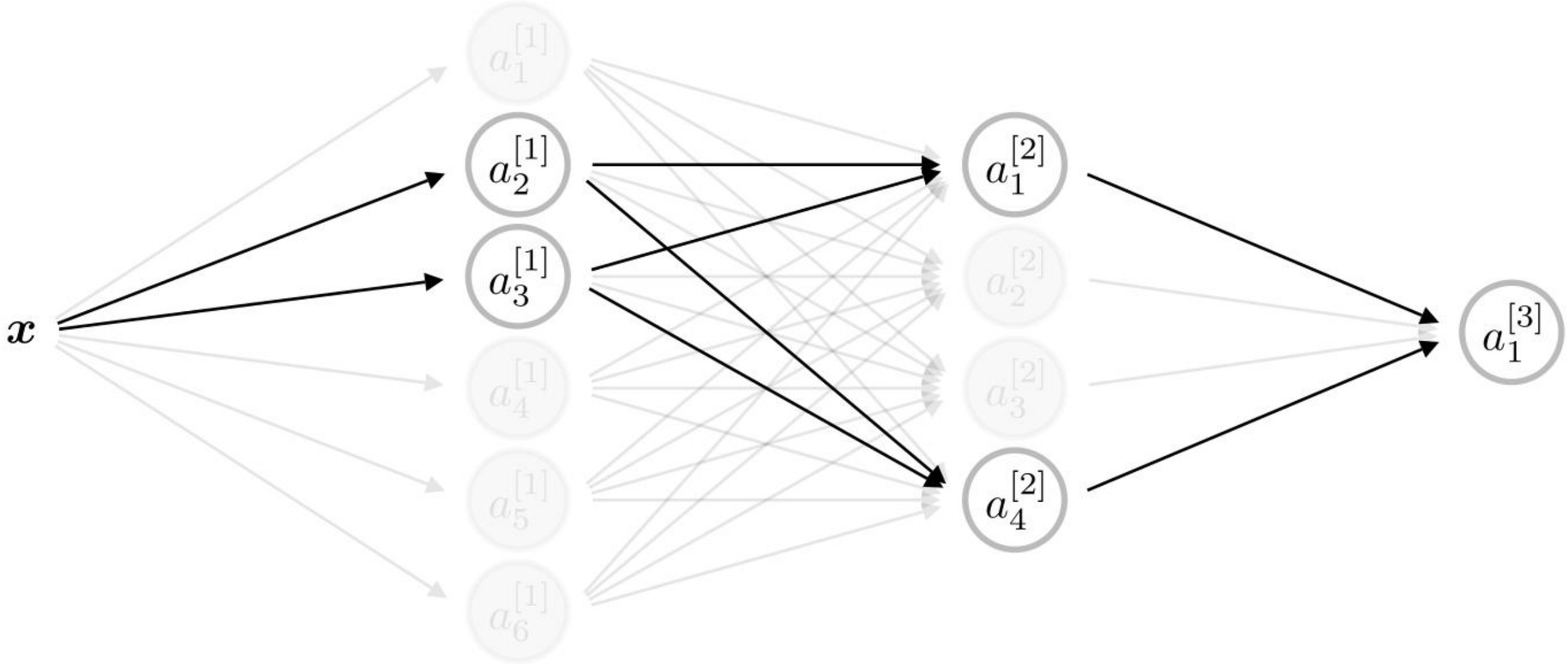
Dropout

1. 对于第 l 层中的每个神经元$(l = 1, \ldots, L-1)$

   - 令 dropout 概率为$p^{[l]}$
   - 以$p^{[l]}$ 为概率，将对应神经元的激活值设置为 0
   - 基于因子$(1 - p^{[l]})^{-1}$ 对激活值进行放缩，弥补 dropout 带来的信息损失

2. 我们通过掩码矩阵实现 dropout 正则化

Dropout


$\boldsymbol{x}$
$a_1^{[1]}$
$a_2^{[1]}$
$a_3^{[1]}$
$a_4^{[1]}$
$a_5^{[1]}$
$a_6^{[1]}$
$a_1^{[2]}$
$a_2^{[2]}$
$a_3^{[2]}$
$a_4^{[2]}$
$a_1^{[3]}$
输入层
第一隐藏层
第二隐藏层
输出层

Dropout--实施过程

1. 令$\boldsymbol{M}^{[l]} \in \{0,1\}^{n \times d^{[l]}}$ 表示一个掩码矩阵

   - $\boldsymbol{M}^{[l]}$中的每个元素来自于成功概率$1-p^{[l]}$的二项分布

2. 前向传播

$$\boldsymbol{A}_d^{[l]} = \frac{\boldsymbol{A}^{[l]} \circ \boldsymbol{M}^{[l]}}{1-p^{[l]}}.$$

3. 后向传播（容易得到）

   - Dropout 不涉及额外模型参数

Dropout--实施过程

1．Dropout 是将一个二项随机变量乘在激活值上

2．更一般地，我们可将其他随机噪声加在或者乘在激活值上

- 在训练集上加随机噪声
- 在模型权重项上加噪声
- （适当）随机打乱训练集标签

早停

1．模型在训练集上的表现会逐渐变好

2．然而，模型可能对训练集进行了过拟合

- 模型在验证集或者测试集上的表现并不会随着训练次数的增加而一直变好

3．早停技术监控模型在验证集上的表现

- 当模型的表现在验证集上不再提高时，模型训练停止

模型集成

1．为了提高模型的泛化能力，我们可以

- 构建多个模型，并将他们输出的均值作为最终估计结果
  - 例如，依据不同的模型初始值得到的不同模型等
- 利用重抽样得到不同的训练集，进而得到不同的训练模型
  - 该方法通常用于衡量深度学习模型的不确定性