Deep Learning I

4. 过拟合

WU Xiaokun 吴晓堃

xkun.wu [at] gmail

2021/03/14

模型选择

一致逼近原理

Universal approximation theorem（非常有用的废话）

In approximation theory, both *shallow* and *deep* networks are known to **approximate any continuous functions** at an **exponential cost**.

一致逼近原理

Universal approximation theorem（非常有用的废话）

In approximation theory, both *shallow* and *deep* networks are known to **approximate any continuous functions** at an **exponential cost**.

就像C语言：能够表达**任何可计算的程序**

- 问题是：如何把这个程序符合规范地写出来?
- 指数级复杂度：NP问题

回顾：模型参数

测试集：实际应用时才能拿到的数据

- 对于训练过程来说是**未知**的
- 测试集划分：二八原则

回顾：模型参数

测试集：实际应用时才能拿到的数据

- 对于训练过程来说是**未知**的
- 测试集划分：二八原则

类比：训练集=日常习题；测试集=考试题

- 训练参数的过程：通过习题掌握解题思路

超参数

超参数：“参数的参数”；模型的**种类**

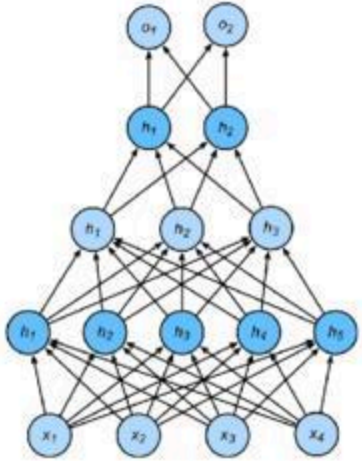
- 模型、黑盒函数、超参数：三者等价

超参数

超参数：“参数的参数”；模型的**种类**

- 模型、黑盒函数、超参数：三者等价

$$
\begin{aligned}
\mathbf{h}_1 &= \sigma(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1) \\
\mathbf{h}_2 &= \sigma(\mathbf{W}_2\mathbf{h}_1 + \mathbf{b}_2) \\
\mathbf{h}_3 &= \sigma(\mathbf{W}_3\mathbf{h}_2 + \mathbf{b}_3) \\
\mathbf{o} &= \mathbf{W}_o\mathbf{h}_3 + \mathbf{b}_o
\end{aligned}
$$



超参数：取决于模型的**设计**

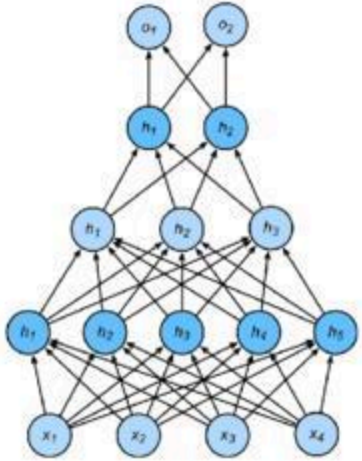
- 隐藏层数
- 每个隐藏层的大小

超参数

超参数：“参数的参数”；模型的**种类**

- 模型、黑盒函数、超参数：三者等价

$$
\begin{aligned}
\mathbf{h}_1 &= \sigma(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1) \\
\mathbf{h}_2 &= \sigma(\mathbf{W}_2\mathbf{h}_1 + \mathbf{b}_2) \\
\mathbf{h}_3 &= \sigma(\mathbf{W}_3\mathbf{h}_2 + \mathbf{b}_3) \\
\mathbf{o} &= \mathbf{W}_o\mathbf{h}_3 + \mathbf{b}_o
\end{aligned}
$$



超参数：取决于模型的**设计**

- 隐藏层数
- 每个隐藏层的大小

- 注意：模型本身的参数是**学习**得到的

模型选择、参数估计

超参数一旦确定，模型也就确定

- 确定超参数的过程称为**模型选择**
- 确定超参数的方法称为**参数估计**

模型选择、参数估计

超参数一旦确定，模型也就确定

- 确定超参数的过程称为**模型选择**
- 确定超参数的方法称为**参数估计**

参数估计的方法

- 统计学习：最小二乘、最大似然、最大后验

模型选择、参数估计

超参数一旦确定，模型也就确定

- 确定超参数的过程称为**模型选择**
- 确定超参数的方法称为**参数估计**

参数估计的方法

- 统计学习：最小二乘、最大似然、最大后验
- 手工方法：凭**经验**选择与试错
  - 所以也称“手气法”、“炼丹”
  - 目前深度学习最常用的方法

计算与应用

机器学习计算与应用的**矛盾**

- 计算过程：已知数据，即训练集
- 应用场景：未知数据，即测试集

计算与应用

机器学习计算与应用的**矛盾**

- 计算过程：已知数据，即训练集
- 应用场景：未知数据，即测试集

例如：训练集中只有动物的图片，应用却是人脸识别

- 《戴珍珠耳环的少女》：什么叫做惊喜？长得像
- 案例：种族歧视

独立同分布假设

理论上：训练集和测试集的数据分布必须一致

- **独立同分布假设 i.i.d.**
- 对数据进行采样的过程没有进行“操控”
  - 无论如何采样，数据分布都是一样的

独立同分布假设

理论上：训练集和测试集的数据分布必须一致

- **独立同分布假设 i.i.d.**
- 对数据进行采样的过程没有进行“操控”
  - 无论如何采样，数据分布都是一样的

实际应用：不可能

- 尽可能地收集与测试相近的数据
  - 习题要尽量像考试题

独立同分布假设

理论上：训练集和测试集的数据分布必须一致

- **独立同分布假设 i.i.d.**
- 对数据进行采样的过程没有进行“操控”
  - 无论如何采样，数据分布都是一样的

实际应用：不可能

- 尽可能地收集与测试相近的数据
  - 习题要尽量像考试题
- 机器学习的**bug之一**
  - 早期机器学习实践：基本等于数据工程
  - 万万没想到：深度学习横空出世

训练误差、泛化误差

评测机器学习算法有两个维度

- **训练误差**：训练阶段，已知数据上的优化误差
- **泛化误差**：测试阶段，未知数据上的泛化误差

例如：

- 日常习题成绩：训练误差
- 真实考试成绩：泛化误差

训练误差越低越好?

考虑预测抛硬币的结果，假设硬币是公平的

- 无论采用什么算法，泛化误差一定是0.5

训练误差越低越好?

考虑预测抛硬币的结果，假设硬币是公平的

- 无论采用什么算法，泛化误差一定是0.5

假设训练集是$\{1，1，1，0，1\}$

- 预测器：始终输出固定值1，训练误差只有0.2

泛化与逻辑

机器学习的**根本问题**：从数据中发现可以泛化的模式

- 从而减小泛化误差
- 学习必须能够迫使机器**理解完成任务的逻辑**

泛化与逻辑

机器学习的**根本问题**：从数据中发现可以泛化的模式

- 从而减小泛化误差
- 学习必须能够迫使机器**理解完成任务的逻辑**

例如：28×28的8位单通道灰度图片

- $2^8 = 256$个灰度值；$28 \times 28 = 784$个像素
  - 总共256^{784}种可能性，不可能被完全记住

泛化与逻辑

机器学习的**根本问题**：从数据中发现可以泛化的模式

- 从而减小泛化误差
- 学习必须能够迫使机器**理解完成任务的逻辑**

例如：28×28的8位单通道灰度图片

- $2^8 = 256$个灰度值；$28 \times 28 = 784$个像素
  - 总共256^{784}种可能性，不可能被完全记住

例如：19×19的围棋棋盘

验证集、测试集

验证集：调超参数时用的数据集

- 对于训练过程来说是**已知**的

类比考试：

- 训练集=日常习题；测试集=考试题
- 验证集=模拟考试题

为什么模拟考试题不能重复?

模拟考试的目的：**预测未来**真实考试的分数

- 注意讨论的问题：**考试、学习的有效性**

为什么模拟考试题不能重复?

模拟考试的目的：**预测未来**真实考试的分数

- 注意讨论的问题：**考试、学习的有效性**

模拟考试分数高，真实考试分数不一定高

- 模拟试题与真实试题差异巨大
- 某次模拟考试发挥超常

为什么模拟考试题不能重复?

模拟考试的目的：**预测未来**真实考试的分数

- 注意讨论的问题：**考试、学习的有效性**

模拟考试分数高，真实考试分数不一定高

- 模拟试题与真实试题差异巨大
- 某次模拟考试发挥超常

模拟考试需要进行多次；每次的试题不能一样

- 减小与真实试题之间的差距
- 减小猜题、背题等运气成分
  - 必须**理解解答问题的逻辑**

验证集划分

最简单的方法：固定比例

- 例如：“二八原则”
- 相当于**模拟考试题雷同**

验证集划分

最简单的方法：固定比例

- 例如：“二八原则”
- 相当于**模拟考试题雷同**

问题：模型**间接推测**出数据分布怎么办？

- 多次重复提交评测：根据评测结果修改模型
- 考试能多次提交吗？根据打分修改解题方法

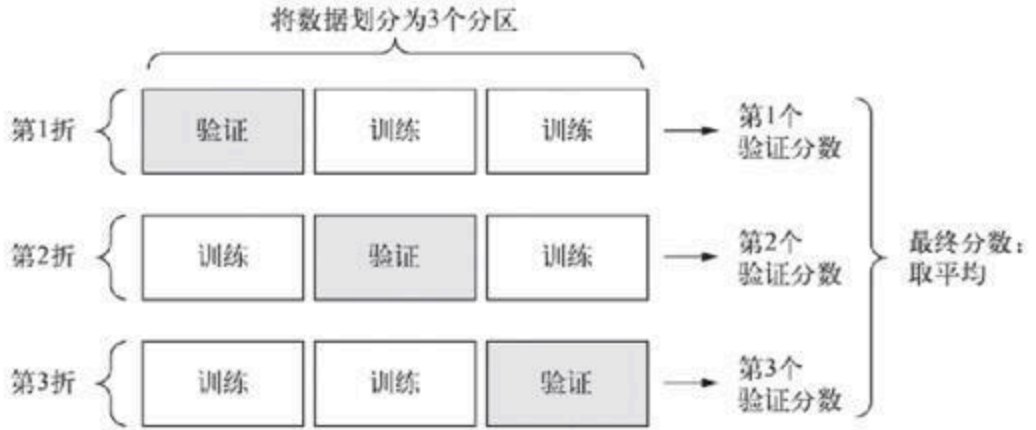
信息泄露

多次重复验证：**间接推测**出验证集分布

- “卷竞赛”：暴力刷分=作弊
  - （负面）案例：禁赛事件
  - 道德、法律风险

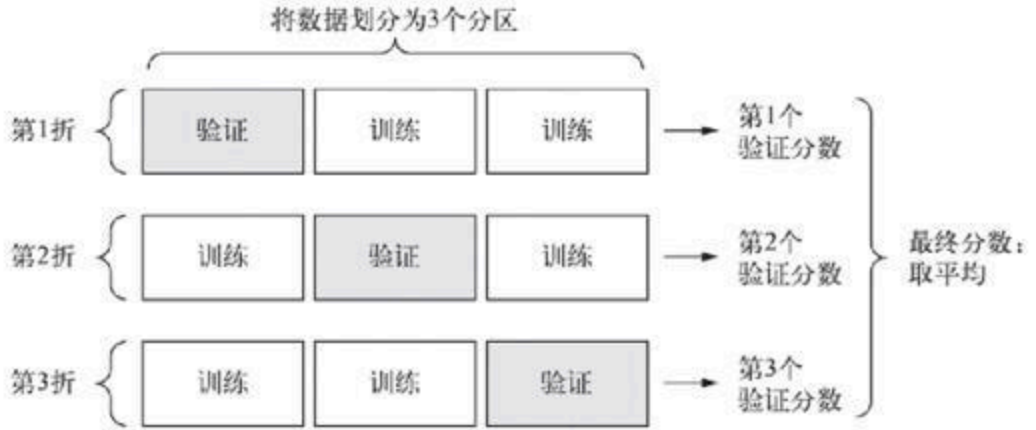
K 折交叉验证

重复使用训练数据，但模拟出每次都不一样的效果



K 折交叉验证

重复使用训练数据，但模拟出每次都不一样的效果

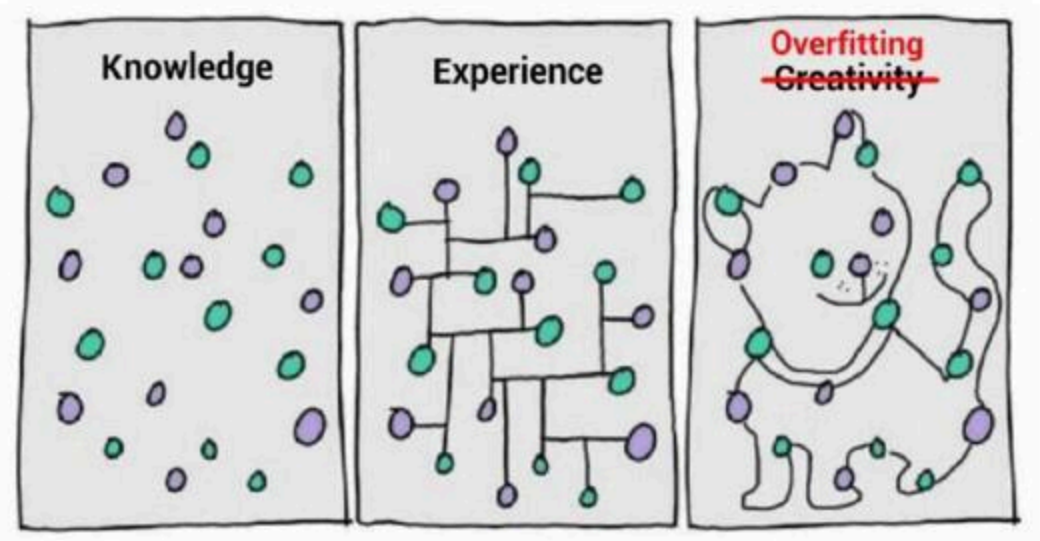


也在**数据量比较少**的时候应用

欠拟合和过拟合

创意?

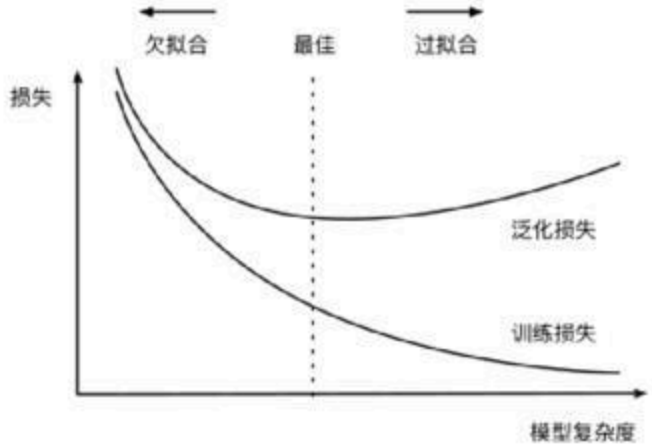
就像观星一样**过度解读**数据



是创意吗？不，只是过拟合

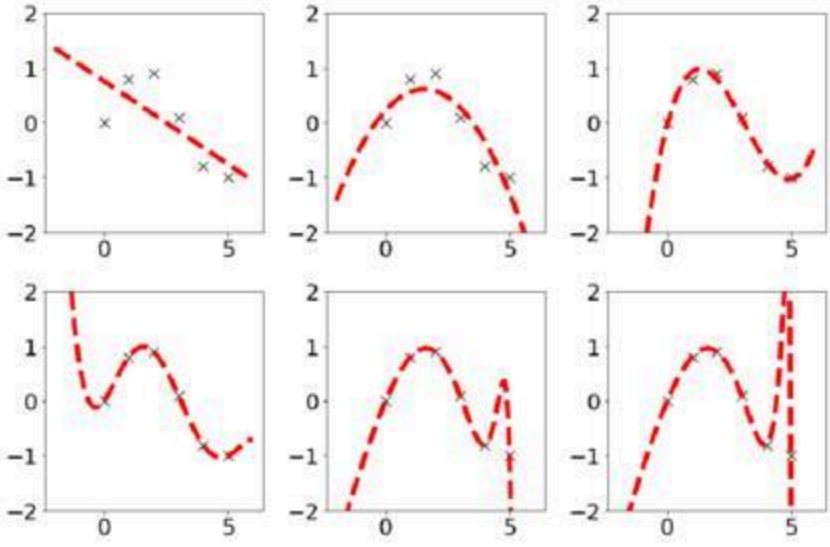
欠拟合、过拟合

- **欠拟合**：训练误差不下降；两种误差接近平行
  - 习题没做够，没有掌握规律
- **过拟合**：训练误差很小，泛化误差达到低点后开始增大
  - 开始死记硬背，但模拟题与考试题不一样



多项式拟合

$$y = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + \ldots$$



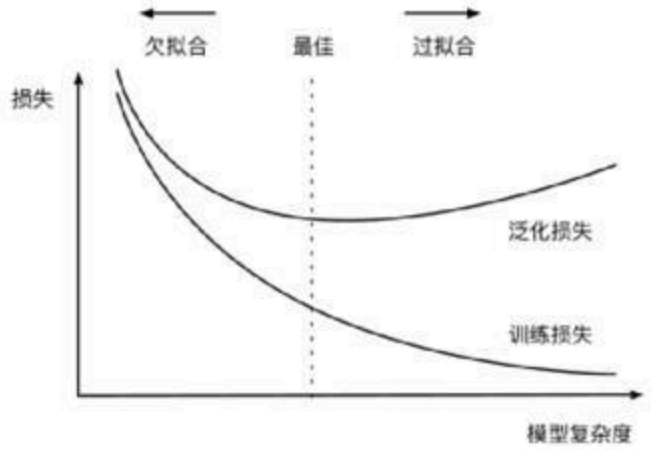
多项式次数：$1, 2, 3, 5, 30, 100$，代表模型、计算复杂度

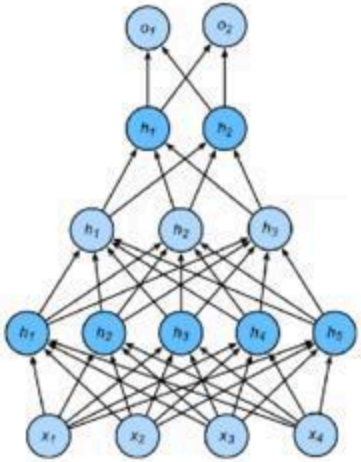
- 剧烈波动：异常规律；如果数据没错，模型必定出错

模型容量

拟合函数的能力：可以简单认为是模型**参数的数量**

- 低容量：欠拟合，不足以发现复杂规律
- 高容量：过拟合，记住所有训练数据

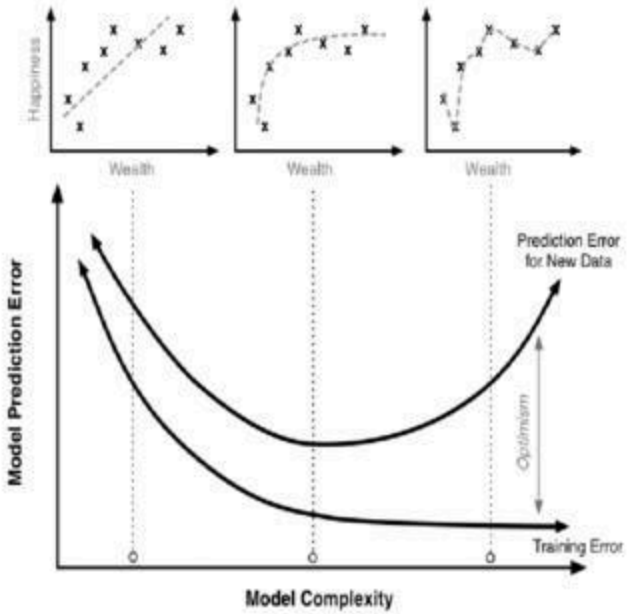




财富和幸福

幸福指数对财富增长不是固定比率关系

- 更真实的情况应该是财富按比例增长
  - 例如1000块工资涨幅对于1亿财富
  - 在**对数空间**应该是线性关系
    - 无法直接观测的
- 第二个数据点应该是其他方面的问题



数据复杂度

模型容量需要匹配数据复杂度

- 实际应用主要是~~（钱）~~**数据量不足**的问题
  - 数据量太少导致模型相对复杂：容易过拟合

数据复杂度

模型容量需要匹配数据复杂度

- 实际应用主要是~~**(钱)**~~**数据量不足**的问题
  - 数据量太少导致模型相对复杂：容易过拟合

几个重要因素

- 样本个数
- 每个样本的元素数量：通道、像素数
- 时间、空间结构：天在上、海在下
- 多样性：不同姿态，类别平衡

估计模型容量

一般难以在不同种类的算法之间比较

- 例如：树模型、感知机之间没有可比性

估计模型容量

一般难以在不同种类的算法之间比较

- 例如：树模型、感知机之间没有可比性

给定模型种类，可以简单判断区别

- 参数个数
- 参数值的取值范围

VC维

统计学习理论，人名缩写，不是“维C”

> *给定分类模型，确定一个最大的数据量：可以被完美分类。*

VC维

统计学习理论，人名缩写，不是“维C”

> *给定分类模型，确定一个最大的数据量：可以被完美分类。*

例如二维输入的感知机，不能解决“XOR问题”，VC维=3

VC维的作用

从数学上提供模型好坏的理论依据

- 确定数据量大小，估计模型容量

VC维的作用

从数学上提供模型好坏的理论依据

- 确定数据量大小，估计模型容量

深度学习中很少使用

- 度量不太准确
- 计算非常困难

数学理论的特点：一般无法在工程实践上应用

实验：多项式拟合

权重衰减

减轻过拟合

无限~~(钱)~~数据量的情况

- 无论模型多复杂，都不会出现过拟合

减轻过拟合

无限~~(钱)~~数据量的情况

- 无论模型多复杂，都不会出现过拟合

限制参数：技巧往往是**无奈之选**

- 参数个数：降低模型复杂度
- 参数值大小

减轻过拟合：Occam剃刀原则

无限~~（钱）~~数据量的情况

- 无论模型多复杂，都不会出现过拟合

限制参数：技巧往往是**无奈之选**

- 参数个数：降低模型复杂度
- 参数值大小

Occam剃刀原则：更简单的往往也是更好的

为什么要限制参数值大小?

参数值大小代表自变量的“**影响力**”

$$y = w_1 x_1 + w_2 x_2 + .. + w_d x_d + b$$

- 参数值过大：输入微小变动，输出巨幅变化
  - 例如分类：导致类别发生变化

为什么要限制参数值大小?

参数值大小代表自变量的“**影响力**”

$$y = w_1 x_1 + w_2 x_2 + .. + w_d x_d + b$$

- 参数值过大：输入微小变动，输出巨幅变化
  - 例如分类：导致类别发生变化
- 等价于过拟合：模型**“适应”了（异常）训练数据**
  - 数据拟合理论：异常值的重要性更高，进而拉升对应参数数值

为什么要限制参数值大小?

参数值大小代表自变量的“**影响力**”

$$y = w_1 x_1 + w_2 x_2 + .. + w_d x_d + b$$

- 参数值过大：输入微小变动，输出巨幅变化
  - 例如分类：导致类别发生变化
- 等价于过拟合：模型**“适应”了（异常）训练数据**
  - 数据拟合理论：异常值的重要性更高，进而拉升对应参数数值

但测试数据中，输入的取值范围可能不一样

- 模型**无法适应测试数据**：导致分类错误

硬性约束

优化问题：添加**约束条件**

$$\arg\min \mathcal{L}(\mathbf{w}, b), \text{s.t.} \|\mathbf{w}\|^2 \leq \theta$$

硬性约束

优化问题：添加**约束条件**

$$\arg\min \mathcal{L}(\mathbf{w}, b), \text{s.t.} \|\mathbf{w}\|^2 \leq \theta$$

- 通常不限制偏移量b
  - 常数项的影响力不大；加法运算数值上比较稳定

硬性约束

优化问题：添加**约束条件**

$$\arg\min \mathcal{L}(\mathbf{w}, b), \text{s.t.} \|\mathbf{w}\|^2 \leq \theta$$

- 通常不限制偏移量b
  - 常数项的影响力不大；加法运算数值上比较稳定
- θ越小：过拟合控制得越好

柔性约束

优化问题的求解：**Language乘数法**

$$\arg\min \mathcal{L}(\mathbf{w}, b), \text{s.t.} \|\mathbf{w}\|^2 \leq \theta$$

$$\arg\min \mathcal{L}(\mathbf{w}, b) + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

- 也称**正则化方法 regularization**

柔性约束

优化问题的求解：**Language乘数法**

$$\arg\min \mathcal{L}(\mathbf{w}, b), \text{s.t.} \|\mathbf{w}\|^2 \leq \theta$$

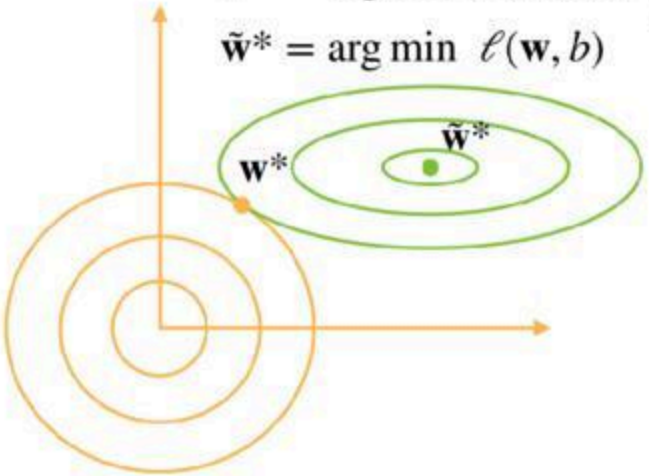
$$\arg\min \mathcal{L}(\mathbf{w}, b) + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

- 也称**正则化方法 regularization**
- 超参数λ控制正则项（即约束条件）的强度
  - $\lambda \to \infty$时：$\mathbf{w} \to 0$

正则化对最优解的影响

$$\mathbf{w}^* = \arg\min \ \ell(\mathbf{w}, b) + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

$$\tilde{\mathbf{w}}^* = \arg\min \ \ell(\mathbf{w}, b)$$



- 把参数值“**拉向**”**原点**：原点附近数值更小

更新法则

梯度

$$\frac{\partial}{\partial \mathbf{w}} \left(\mathcal{L}(\mathbf{w}, b) + \frac{\lambda}{2} \|\mathbf{w}\|^2 \right) = \frac{\partial}{\partial \mathbf{w}} \mathcal{L}(\mathbf{w}, b) + \lambda \mathbf{w}$$

更新法则

梯度

$$\frac{\partial}{\partial \mathbf{w}} \left(\mathcal{L}(\mathbf{w}, b) + \frac{\lambda}{2} \|\mathbf{w}\|^2 \right) = \frac{\partial}{\partial \mathbf{w}} \mathcal{L}(\mathbf{w}, b) + \lambda \mathbf{w}$$

更新参数

$$\mathbf{w}_{t+1} = (1 - \eta\lambda) \mathbf{w}_t - \eta \frac{\partial}{\partial \mathbf{w}_t} \mathcal{L}(\mathbf{w}_t, b_t)$$

- 通常$\eta\lambda < 1$，因此通常称为**权重衰退 weight decay**

其他正则化

L_2正则化，也称**脊回归 ridge regression**

$$\arg\min \mathcal{L}(\mathbf{w}, b) + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

L_1正则化，也称**套索回归 lasso regression**

$$\arg\min \mathcal{L}(\mathbf{w}, b) + \lambda|\mathbf{w}|$$

其他正则化

L_2正则化，也称**脊回归 ridge regression**

$$\arg\min \mathcal{L}(\mathbf{w}, b) + \frac{\lambda}{2}\|\mathbf{w}\|^2$$

L_1正则化，也称**套索回归 lasso regression**

$$\arg\min \mathcal{L}(\mathbf{w}, b) + \lambda|\mathbf{w}|$$

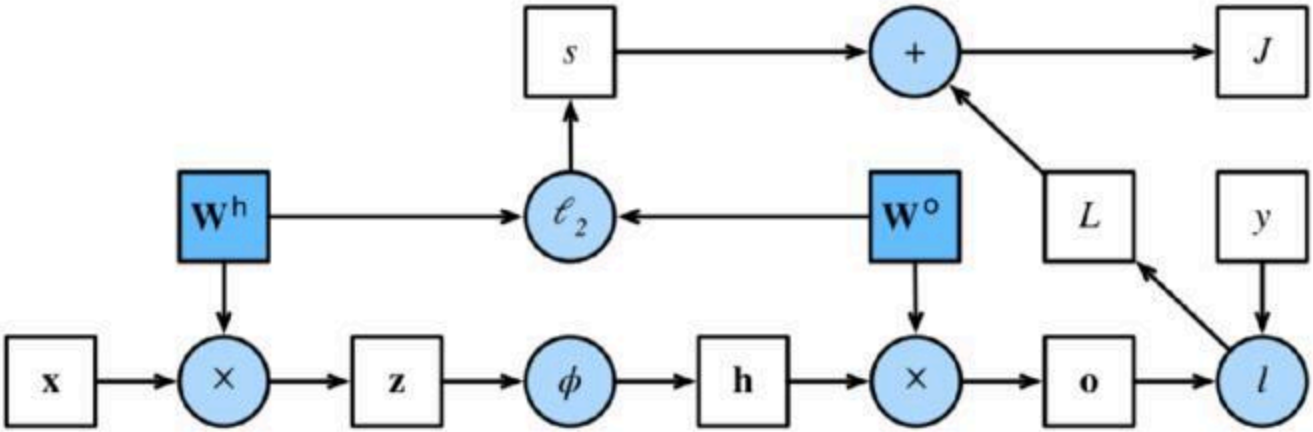
一般情况：L_n正则化

$$\arg\min \mathcal{L}(\mathbf{w}, b) + \lambda\|\mathbf{w}\|_n$$

自动微分计算图

$$\mathbf{h} = \sigma(\mathbf{W}_h\mathbf{x} + \mathbf{b}_h)$$
$$\mathbf{o} = \mathbf{W}_o\mathbf{h} + \mathbf{b}_o$$
$$s = \frac{\lambda}{2}\left(\|\mathbf{W}_h\|_F^2 + \|\mathbf{W}_o\|_F^2\right)$$

$$\mathbf{z} = \mathbf{W}_h\mathbf{x}$$
$$\mathbf{h} = \phi(\mathbf{z})$$
$$\mathbf{o} = \mathbf{W}_o\mathbf{h}$$
$$\mathcal{L} = l(O, y)$$



实验：权重衰退

丢弃法

Dropout

Dropout：劝退

Dropout

Dropout：劝退

注意：标准高不等于内卷！

> *普通人有上上智。– 李克强*

什么是好模型?

Occam剃刀原则：更简单的往往也是更好的

- 限制参数的个数、取值

什么是好模型?

Occam剃刀原则：更简单的往往也是更好的

- 限制参数的个数、取值

鲁棒性原则：对输入数据的噪音不敏感

- [Bishop 1995] 具有输入噪声的训练等价于Tikhonov正则化

什么是好模型?

Occam剃刀原则：更简单的往往也是更好的

- 限制参数的个数、取值

鲁棒性原则：对输入数据的噪音不敏感

- [Bishop 1995] 具有输入噪声的训练等价于Tikhonov正则化
- 训练时**给模型加入噪音**
  - 相当于模拟有噪音的输入："制造困难也要上"

无偏噪音

假设对$\mathbf{x}$加入噪音得到$\mathbf{x}'$：相当于**直接修改数据**

- 因此应该保证**数据分布不发生偏移**：$\mathbb{E}[\mathbf{x}'] = \mathbf{x}$

无偏噪音

假设对$\mathbf{x}$加入噪音得到$\mathbf{x}'$：相当于**直接修改数据**

- 因此应该保证**数据分布不发生偏移**：$\mathbb{E}[\mathbf{x}'] = \mathbf{x}$

丢弃法 Dropout：对每个元素做无偏随机扰动

- 按照概率缩小（清0）一部分数据：剩下的部分必须要**放大相应的量**

$$x_i' = \begin{cases} 0 & \text{概率 } p \\ \frac{x_i}{1-p} & \text{otherwise} \end{cases}$$

- **丢弃率**p是超参数

无偏噪音

假设对$\mathbf{x}$加入噪音得到$\mathbf{x}'$：相当于**直接修改数据**

- 因此应该保证**数据分布不发生偏移**：$\mathbb{E}[\mathbf{x}'] = \mathbf{x}$

丢弃法 Dropout：对每个元素做无偏随机扰动

- 按照概率缩小（清0）一部分数据：剩下的部分必须要**放大相应的量**

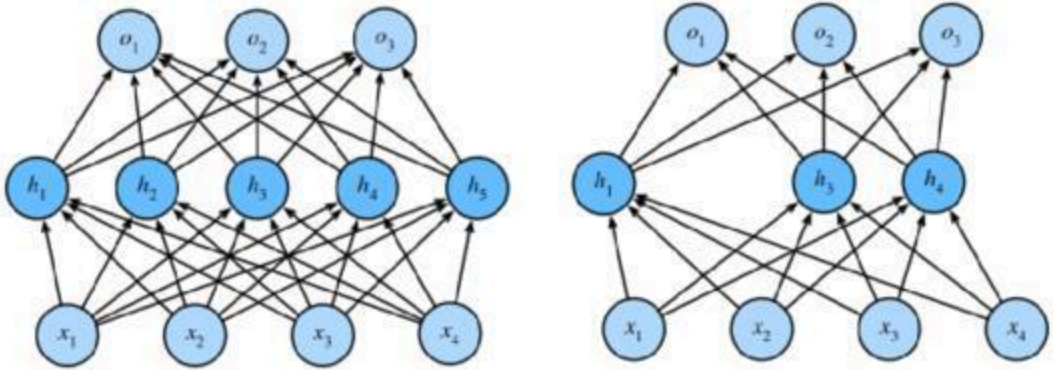
$$x_i' = \begin{cases} 0 & \text{概率 } p \\ \frac{x_i}{1-p} & \text{otherwise} \end{cases}$$

- **丢弃率**p是超参数
- 验证**无偏**：计算期望，$p * 0 + (1-p) * \frac{x_i}{1-p}$

使用丢弃法

丢弃法通常作用在**隐藏层的输出**上

$$\mathbf{h} = \sigma(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1)$$
$$\mathbf{h}' = \text{dropout}(\mathbf{h})$$
$$\mathbf{o} = \mathbf{W}_o\mathbf{h}' + \mathbf{b}_o$$



评测过程

正则化方法都**只在训练中使用**

- 训练更新部分参数；其余参数不参与优化，保持随机化状态

评测过程

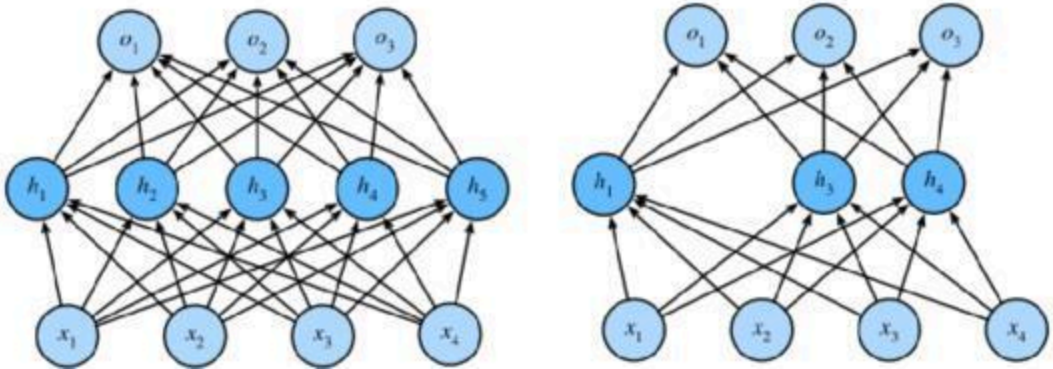
正则化方法都**只在训练中使用**

- 训练更新部分参数；其余参数不参与优化，保持随机化状态

评测过程中，丢弃法直接返回原输入

$$\mathbf{h} = \text{dropout}(\mathbf{h})$$

- 保证**输出的确定性**：架构、参数保持不变



实验：丢弃法

实战Kaggle比赛：预测房价