人工智能与机器学习基础 2025-HW1

TA: 郑悟强

September 2025

1 模型偏差 (Model Bias)

在进行机器学习任务时，我们有时候模型最终的表现可能并不能达到非常理想的程度。除去优化方法等限制外，还有可能是我们的模型本身存在能力的上限。在这个问题中，我们会一起分析一下，从模型角度来看，为什么误差会存在，误差的上限在哪里？进一步，我们可以从结果分析一下如何尽量让这个误差变小。

(a) 对于我们的训练数据集 $\mathcal{D}_{train}\{(x_i, y_i)\}_{i=1}^N$，存在其最优拟合函数 $y = f_{\text{true}}(x) + \epsilon$，这里的误差 ϵ 服从期望为 0，方差为 σ^2 的分布。实际上，我们训练时可能会随机选取到整个训练集的一个子集，拟合出来的模型为 $f_{\hat{w}(\text{train})}$。进一步，我们假设我们一共这样选取了 n 次训练子集，得到的所有的模型的均值为 $\bar{f}$，那么，事实上，我们理论上的泛化误差，用 MSE 作为度量方式，可以由此推导：

$$\mathbb{E}_{\text{train}}[\text{Generalization Error}] = \mathbb{E}_{\text{train}}[\mathbb{E}_{x,y}[(y - f_{\hat{w}(\text{train})}(x))^2]] = \sigma^2 + \text{Bias}^2 + \text{Var}(f), \tag{1}$$

其中：

$$\text{Bias}^2 = \mathbb{E}_{\text{train}}[(\bar{f} - f_{\text{true}})^2], \quad \text{Var}(f) = \mathbb{E}_{\text{train}}[(\bar{f} - f_{\hat{w}(\text{train})})^2]$$

问题：请你补全这个推导过程

(b) 请说明，与训练数据无关，模型偏差 $Bias^2$ 非负且总是存在。并且回答，什么时候，模型偏差永远大于 0 呢？（可以举一个栗子）

(c) 请说明，$Bias^2$ 和 Var 分别与 N 相关还是与 n 相关，成什么关系？（$\mathcal{O}(?)$）。具体的，在神经网络中，模型的复杂度与这两项的关系是什么样的？

2 数据偏差 (Data Bias)

除了上题中讲述的模型能力的偏差，实际上，可能训练数据本身也限制了你的模型的表现。主要原因在于，我们的有限样本 S 无法准确泛化到整体的期望数据分布 D，我们对这种有限数据带来的泛化限制称作数据偏差。

(a) 假设我们在训练集训练得到的模型为 $h \in H$，其中 H 为整个模型代表的集合，比如，H 为所有的线性模型，h 为其中我们训练得到的一个固定参数的线性模型。我们定义这样的误差函数：

$$\text{err}_D(h) = \mathbb{E}_{(x,y)\sim D}[I(h(x) \neq y)], \quad \text{err}_S(h) = \frac{1}{n}\sum_{i=1}^{n} I(h(x_i) \neq y_i), \tag{2}$$

其中 $I(\cdot,\cdot)$ 为示性函数（相等为 1，不等为 0）。那么，我们可以这样定义这个 h 的泛化性是不好的：

$$|\text{err}_D(h) - \text{err}_S(h)| \geq \epsilon, \tag{3}$$

其中 ϵ 是我们定义的一个误差界。

请证明：对于固定的 h，$\Pr[|\mathrm{err}_D(h) - \mathrm{err}_S(h)| > \epsilon] \le 2\exp(-2n\epsilon^2)$

提示：可以使用 Hoeffding 不等式：对于独立有界随机变量 $Z_i \in [a, b]$，$\Pr[|(1/n)\sum Z_i - \mathbb{E}| > \epsilon] \le 2\exp(-2n\epsilon^2/(b-a)^2)$。

(b) 现在，我们给这个结论泛化一下，用 $|H|$ 表示整个模型类的空间大小。

请证明：$\Pr[\exists h \in H, |\mathrm{err}_D(h) - \mathrm{err}_S(h)| > \epsilon] \le 2|H|\exp(-2n\epsilon^2)$

(c) 请尝试解释一下这个结论的直观意义，我们的数据偏差与哪些因素有关？呈什么样的相关性变化趋势？我们有哪些方法来尽量减缓数据偏差带来的影响？

3 贝叶斯分类器

在课堂上，老师讲述了一种线性的分类器，逻辑回归。从数学上，其基本原理在于，直接建模 $p(y|x)$ 这个概率值，其中 x 是样本的特征，y 是类别。逻辑回归通过直接优化这个概率，通过梯度下降的更新策略，从数据中学到这个概率的建模。

但这种做法也存在一定的问题，比如训练成本的高昂，我们需要从数据集中不断采样更新模型，并且优化方式的限制导致并不一定能轻松训练得到非常好的模型。于是我们想，既然直接建模 $p(y|x)$ 很难，那能不能做一定的数学上的变化，得到更简单优美的方式呢？

考虑二分类任务，y_1, y_2 表示样本属于类别 1 或类别 2，我们对这个概率做一个贝叶斯公式的展开：

$$p(y_1|x) = \frac{p(x|y_1)p(y_1)}{p(x|y_1)p(y_1) + p(x|y_2)p(y_2)}. \tag{4}$$

我们可以做这样的假设，我们假设每个类别的数据特征服从各自的高斯分布，两个高斯分布的期望不同，但方差相同（可以想想这么简化有什么意义？）

$$f_{\mu,\Sigma}(x) = \frac{1}{\sqrt{(2\pi)^D det(\Sigma)}} exp\{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\}. \tag{5}$$

其中，D 是特征 x 的维度，类别 1 的样本期望为 μ^1，方差为 Σ，数据集中，类别 1 有 n_1 个样本。类别 2 的样本数据期望为 μ^2，方差为 Σ，数据集中有 n_2 样本。即 $x|y_1 \sim \mathcal{N}(\mu^1, \Sigma), x|y_2 \sim \mathcal{N}(\mu^2, \Sigma)$

(a) 请通过极大似然估计的方法，推导出 μ^1, μ^2, Σ 的显式解。提示：结果中应该只包含 n_1, n_2, $\{x_i^1\}_{i=1}^{n_1}$(类别 1 的所有数据特征), $\{x_i^2\}_{i=1}^{n_2}$(类别 2 的所有数据的特征)。

提示：我们希望优化 $\mathcal{L}(\mu^1, \mu^2, \Sigma) = \prod_{i=1}^{n_1} p(x_i^1|y_1)\prod_{i=1}^{n_2} p(x_i^2|y_2)$

(b) 表 1中，我们提供了一个简单的三特征的测试数据：

用贝叶斯分类器建模后，两个类别的期望分别是多少？方差是多少？现在我们有 1 个新的数据，请帮我判断它应该属于哪个类别，概率是多少？新数据：$(2.7, 2.9, 3.5)$

(c) 事实上，贝叶斯分类器也可以写成一种逻辑回归，我们可以这样处理：

$$p(y_1|x) = \frac{p(x|y_1)p(y_1)}{p(x|y_1)p(y_1) + p(x|y_2)p(y_2)} = \frac{1}{1 + \frac{p(x|y_2)p(y_2)}{p(x|y_1)p(y_1)}} = \sigma(z), \tag{6}$$

我们可以给下面那一项看作 $z = \frac{p(x|y_2)p(y_2)}{p(x|y_1)p(y_1)}$。

事实上，这个 z 相对于 x 也是线性的，即我们可以写成 $z = w \cdot x + b$。请尝试用 $\mu^1, \mu^2, \Sigma, n_1, n_2$ 推导出 w, b。

Comments：从这里我们能够看到，其实逻辑回归与贝叶斯分类都是线性分类器，不过区别在于，逻辑回归通过优化目标的梯度下降进行学习，贝叶斯分类器采用直接通过高斯分布的假设寻找 w, b 的显式解。二者

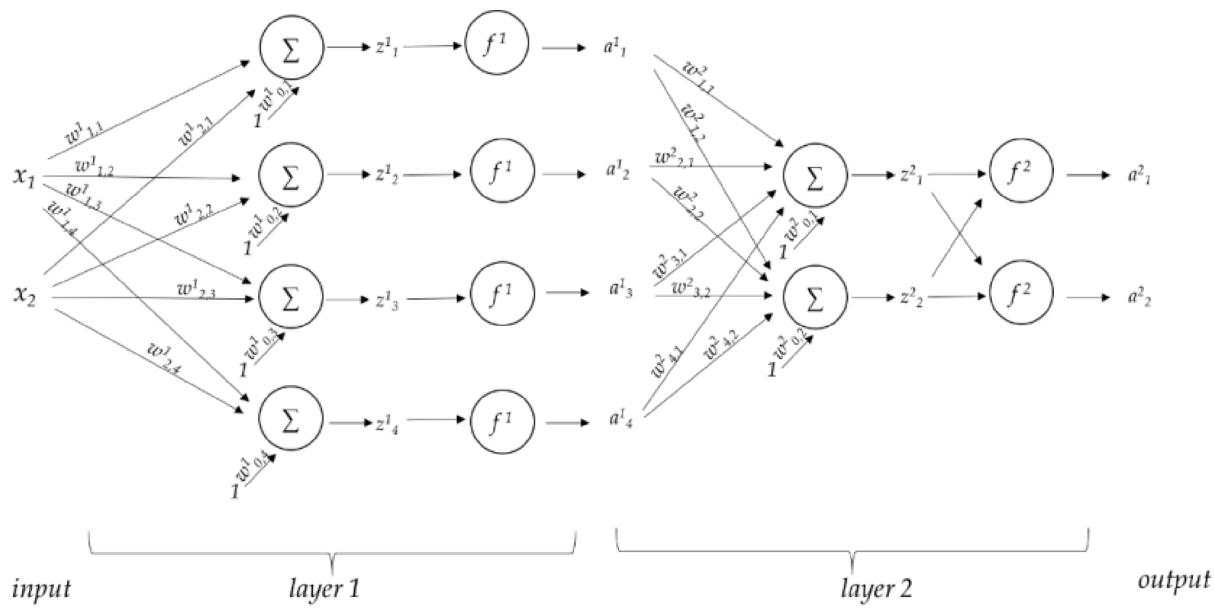
x_1	x_2	x_3	标签
2.0	2.5	2.0	0
2.5	2.8	2.2	0
3.0	2.7	2.5	0
2.2	3.0	2.3	0
2.8	2.6	2.4	0
3.5	3.8	3.2	1
3.2	4.0	3.5	1
3.8	3.5	3.7	1
3.0	3.9	3.3	1
4.0	3.6	3.9	1

表 1: 测试数据（10 条样本，3 个特征，二分类）

互有优劣。比如逻辑回归表达能力强，往往可以实现更好的泛化性，但存在优化困难等问题。贝叶斯分类器优化简单，只需要过一遍数据计算对应的参数即可，但存在表达能力弱，且理论最优解会导致容易过拟合，泛化性差。你也可以从另一种角度来理解这个问题。我们知道，线性回归是存在理论最优解的，但逻辑回归不行，直接求梯度后会发现，这个等式不存在解析解。所以我们可以对数据加上随机性（高斯分布）的假设，通过一定的数学方法求出其最优解。这样会带来更强的假设，但会得到一个更加优美的数值解。

4 神经网络

考虑下图所示的神经网络，其中所有隐藏神经元都使用 ReLU 激活函数（图中的 f 1），输出层使用 softmax 激活函数（图中的 f^2），输出为 softmax 输出（图中的 a_1^2 和 a_2^2）。



给定输入 $x = [x_1, x_2]^T$，网络的隐藏单元按以下方程分阶段激活：

$$
\begin{aligned}
z_1^1 &= x_1 w_{1,1}^1 + x_2 w_{2,1}^1 + w_{0,1}^1 \quad a_1^1 = \max\{z_1^1, 0\} \\
z_2^1 &= x_1 w_{1,2}^1 + x_2 w_{2,2}^1 + w_{0,2}^1 \quad a_2^1 = \max\{z_2^1, 0\} \\
z_3^1 &= x_1 w_{1,3}^1 + x_2 w_{2,3}^1 + w_{0,3}^1 \quad a_3^1 = \max\{z_3^1, 0\} \\
z_4^1 &= x_1 w_{1,4}^1 + x_2 w_{2,4}^1 + w_{0,4}^1 \quad a_4^1 = \max\{z_4^1, 0\}
\end{aligned}
$$

$$
\begin{aligned}
z_1^2 &= a_1^1 w_{1,1}^2 + a_2^1 w_{2,1}^2 + a_3^1 w_{3,1}^2 + a_4^1 w_{4,1}^2 + w_{0,1}^2 \\
z_2^2 &= a_1^1 w_{1,2}^2 + a_2^1 w_{2,2}^2 + a_3^1 w_{3,2}^2 + a_4^1 w_{4,2}^2 + w_{0,2}^2
\end{aligned}
$$

网络的最终输出通过对最后一层应用 softmax 函数得到：

$$
\begin{aligned}
a_1^2 &= \frac{e^{z_1^2}}{e^{z_1^2} + e^{z_2^2}} \\
a_2^2 &= \frac{e^{z_2^2}}{e^{z_1^2} + e^{z_2^2}}
\end{aligned}
$$

在这个问题中，我们将考虑以下参数设置：

$$
\begin{bmatrix} w_{1,1}^1 & w_{1,2}^1 & w_{1,3}^1 & w_{1,4}^1 \\ w_{2,1}^1 & w_{2,2}^1 & w_{2,3} & w_{2,4}^1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}, \begin{bmatrix} w_{0,1}^1 \\ w_{0,2}^1 \\ w_{0,3}^1 \\ w_{0,4}^1 \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \end{bmatrix},
$$

$$
\begin{bmatrix} w_{1,1}^2 & w_{1,2}^2 \\ w_{2,1}^2 & w_{2,2}^2 \\ w_{3,1}^2 & w_{3,2}^2 \\ w_{4,1}^2 & w_{4,2}^2 \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 1 & -1 \\ 1 & -1 \\ 1 & -1 \end{bmatrix}, \begin{bmatrix} w_{0,1}^2 \\ w_{0,2}^2 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \end{bmatrix}
$$

(a) 考虑输入 $x_1 = 3, x_2 = 14$。网络隐藏单元的输出 $(f^1(z_1^1), f^1(z_2^1), f^1(z_3^1), f^1(z_4^1))$ 和最终输出 (a_1^2, a_2^2) 是什么?

(b) 考虑以下输入向量：$x^{(1)} = [0.5, 0.5]^T, \quad x^{(2)} = [0, 2]^T, \quad x^{(3)} = [-3, 0.5]^T$。输入一个矩阵，其中每一列表示每个输入向量的隐藏单元的输出 $(f(z_1^1), \ldots, f(z_4^1))$。

(c) 使用交叉熵损失函数 (Cross-Entropy Loss)，给定输入样本 $(x_1 = 3, x_2 = 14)$ 和目标向量 $(y_1, y_2) = (0, 1)$，执行一次反向传播步骤，以学习率 $\eta = 0.1$ 更新网络中的每一个权重。