Studies ML

Jacob Xie

2023-03-11

2 模型评估与选择

2.1 误差与拟合

误差与拟合		
名称	英文	描述
错误率	error rate	如果在 m 个样本中有 a 个样本分类错误，则错误率 $E = a/m$
精度	accuracy	$1 - a/m$
误差	error	学习器的实际预输出与样本的真实输出之间的差异
训练误差	training error	学习器在训练集上的误差
经验误差	empirical error	
泛化误差	generalization error	在新样本上的误差
过拟合	over fitting	学习器把训练样本自身的一些特点当做了所有潜在样本都会具有的一般性质，导致泛化性能下降
欠拟合	under fitting	与过拟合相对应
模型选择	model selection	

2.2 评估方法

评估方法		
测试集	testing set	
测试误差	testing error	
留出法	hold-out	直接将数据集 D 划分为两个互斥的集合，其中一个集合作为训练集 S，另一个作为测试集 T，即 $D = S \cup T, S \cap T = \varnothing$。在 S 上训练出模型后，用 T 来评估其测试误差，作为对泛化误差的估计。
采样	sampling	
分层采样	stratified sampling	保留类别比例的采样方式
保真性	fidelity	
交叉验证法	cross validation	将数据集 D 划分为 k 个大小相似的互斥子集，即 $D = D_1 \cup D_2 \cup \cdots \cup D_k, D_i \cap D_j = \varnothing$ $(i \neq j)$。每个子集 D_i 都尽可能保持数据分布的一致性，即从 D 中通过分层采样的到。然后每次用 $k-1$ 个子集的并集作为训练集，余下的那个子集作为测试集；这样就可获得 k 组训练/测试集，从而可进行 k 次训练和测试，最终返回的是这 k 个测试结果的均值。
k 折交叉验证	k-fold cross validation	
留一法	leave-one-out	
自助法	bootstrapping	
自助采样法	bootstrap sampling	
包外估计	out-of-bag estimate	
参数	parameter	
调参	parameter tuning	
验证集	validation set	

2.3 性能度量

2.3.1 错误率与精度

性能度量（performance measure）：衡量模型泛化能力的评价标准。
均方误差（mean squared error）：

$$E(f;D)=\frac{1}{m}\sum_{i=1}^{m}(f(\boldsymbol{x}_i)-y_i)^2 \tag{2.2}$$

对于数据分布 $\mathcal{D}$ 和概率密度函数 $p(\cdot)$，均方误差可描述为：

$$E(f;\mathcal{D})=\int_{\boldsymbol{x}\ \mathcal{D}}(f(\boldsymbol{x}-y)^2)p(\boldsymbol{x})\,d\boldsymbol{x} \tag{2.3}$$

错误率是分类错误的样本数占样本总数的比例：

$$E(f;D)=\frac{1}{m}\sum_{i=1}^{m}\mathbb{I}(f(\boldsymbol{x}_i)\neq y_i) \tag{2.4}$$

精度则是分类正确的样本数占样本总数的比例：

$$\begin{aligned} acc(f;D)&=\frac{1}{m}\sum_{i=1}^{m}\mathbb{I}(f(\boldsymbol{x}_i)=y_i)\\ &=1-E(f;D)\end{aligned} \tag{2.5}$$

对于数据分布 $\mathcal{D}$ 和概率密度函数 $p(\cdot)$，错误率与精度可分别描述为

$$E(f;\mathcal{D})=\int_{\boldsymbol{x}\ \mathcal{D}}\mathbb{I}(f(\boldsymbol{x})\neq y)p(\boldsymbol{x})\,d\boldsymbol{x} \tag{2.6}$$

$$\begin{aligned} acc(f;\mathcal{D})&=\int_{\boldsymbol{x}\ \mathcal{D}}\mathbb{I}(f(\boldsymbol{x})=y)p(\boldsymbol{x})\,d\boldsymbol{x}\\ &=1-E(f;\mathcal{D})\end{aligned} \tag{2.7}$$

2.5 偏差与方差

偏差-方差分解（bias-variance decomposition）：对学习算法的期望泛化错误率进行拆解。
偏差-方差窘境（bias-variance dilemma）

3 线性模型

4 决策树

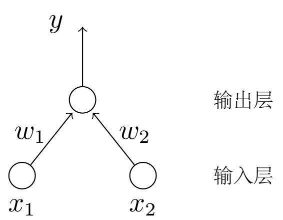
5 神经网络

5.1 神经元模型

神经元模型		
名称	英文	描述
神经网络	neural networks	由具有适应性的简单单元组成的广泛并行互联的网络，它的组织能够模拟生物神经系统对真实世界物体所作出的交互反应
神经元	neuron	神经网络中最基本的成分
阈值	threshold	公式中记作 θ
连接	connection	
激活函数	activation function	处理以产生神经元的输出
挤压函数	squashing function	

5.2 感知机与多层网络

感知机（Perceptron）由两层神经元组成。如图示，输入层接受外界输入信号后传递给输出层，输出层是 M-P 神经元，也称阈值逻辑单元（threshold logic unit），其中 $y = f(\sum_i w_i x_i - \theta)$，而 f 为激活函数。



一般而言，给定训练数据集，权重 $w_i\,(i = 1, 2, \ldots, n)$ 以及阈值 θ 可通过学习得到。而阈值 θ 可视为一个固定输入为 -1.0 的哑结点（dummy node）所对应的链接权重 w_{n+1}，因此权重和阈值的学习就可以统一为权重的学习。感知机的学习规则非常简单，对训练样例 $(\boldsymbol{x}, y)$ 而言，如果当前感知机的输出位 $\hat{y}$，则感知机权重将这样调整：

$$w_i \leftarrow w_i + \Delta w_i \tag{5.1}$$

$$\Delta w_i = \eta(y - \hat{y})x_i \tag{5.2}$$

其中 $\eta \in (0, 1)$ 称为学习率（learning rate）。公式 5.2 为感知机学习算法中的参数更新式。

I 感知机模型

感知机模型的式可表示为：

$$\begin{aligned} y &= f\left(\sum_{i=1}^{n} w_i x_i - \theta\right) \\ &= f(\boldsymbol{w}^T\boldsymbol{x} - \theta) \end{aligned}$$

其中，$x \in \mathbb{R}^n$ 即样本的特征向量，是感知机模型的输入；$\boldsymbol{w},\theta$ 是感知机模型的参数，权重 $w \in \mathbb{R}^n$，θ 为阈值。假定 f 为阶跃函数，那么感知机模型的式可以表示为：

$$y = \varepsilon(\boldsymbol{w}^T\boldsymbol{x} - \theta) = \begin{cases} 1, & \boldsymbol{w}^T\boldsymbol{x} - \theta \geq 0; \\ 0, & \boldsymbol{w}^T\boldsymbol{x} - \theta < 0. \end{cases}$$

由于 n 维空间中的超平面方程为

$$w_1x_1 + w_2x_2 + \cdots + w_nx_n + b = \boldsymbol{w}^T\boldsymbol{x} + b = 0$$

因此感知机模型式中的 $\boldsymbol{w}^T\boldsymbol{x} - \theta$ 可视为 n 维空间中的一个超平面，将 n 维空间划分为 $\boldsymbol{w}^T\boldsymbol{x} - \theta \geq 0$ 与 $\boldsymbol{w}^T\boldsymbol{x} - \theta < 0$ 的两个子空间（试想一下三维空间下的一个平面将空间切分为两部分）。那么落在前一个子空间的样本对应的模型输出值为 1，而后者为 0，如此实现了分类功能。

II 学习策略

给定一个线性可分的数据集 T，感知机的学习目标是求得能对数据集 T 中的正负样本完全正确划分的分离超平面

$$\boldsymbol{w}^T\boldsymbol{x} - \theta = 0$$

假设此时误分类样本集合为 $M \subset T$，对任意一个误分类样本 $(\boldsymbol{x}, y) \in M$ 而言，当 $\boldsymbol{w}^T\boldsymbol{x} - \theta \geq 0$ 时，模型输出值为 $\hat{y} = 1$，样本真实标记为 $y = 0$；反之亦然。综上，以下式恒成立：

$$(\hat{y} - y)(\boldsymbol{w}^T\boldsymbol{x} - \theta) \geq 0$$

因此对于给定数据集 T，其损失函数可以定义为

$$L(\boldsymbol{w}, \theta) = \sum_{\boldsymbol{x} \in M} (\hat{y} - y)(\boldsymbol{w}^T\boldsymbol{x} - \theta)$$

非负之和显然非负，因此损失函数为非负。当没有误分类点时，损失函数的值为 0；误分类点越少，误分类点离超平面越近，损失函数值就越小。因此对于给定的数据集 T，损失函

数 $L(\boldsymbol{w},\theta)$ 是关于 $\boldsymbol{w},\theta$ 的连续可导函数（注意是关于 $\boldsymbol{w},\theta$ 的可导，意味着之后将要对其进行梯度下降算法，即对其使用导数计算）。

连续：

$$\lim_{x\to x_0} f(x)=f(x_0)$$

可导：

$$f'(x_0)=\lim_{\Delta x\to 0}\frac{\Delta y}{\Delta x}=\lim_{\Delta x\to 0}\frac{f(x_0+\Delta x)-f(x_0)}{\Delta x}$$

III 学习算法

感知机模型的学习问题可以转化为求解损失函数的最优化问题。给定数据集

$$T=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_N,y_N),\}$$

其中 $\boldsymbol{x}_i\in\mathbb{R},y_i\in\{0,1\}$，求参数 $\boldsymbol{w},\theta$ 使得损失函数最小化：

$$\min_{\boldsymbol{w},\theta} L(\boldsymbol{w},\theta)=\min_{\boldsymbol{w},\theta}\sum_{\boldsymbol{x}_i\in M}(\hat{y}_i-y_i)(\boldsymbol{w}^T\boldsymbol{x}-\theta)$$

其中 $M\subset T$ 为误分类样本集合。若将阈值 θ 视为一个固定输入为 -1 的“哑结点”（前文有提到），即：

$$-\theta=-1\cdot w_{n+1}=x_{n+1}\cdot w_{n+1}$$

那么 $\boldsymbol{w}^T\boldsymbol{x}-\theta$ 可简化为

$$\begin{aligned}\boldsymbol{w}^T\boldsymbol{x}-\theta&=\sum_{j=1}^{n}w_jx_j+x_{n+1}\cdot w_{n+1}\\&=\sum_{j=1}^{n+1}w_jx_j\\&=\boldsymbol{w}^T\boldsymbol{x_i}\end{aligned}$$

其中 $\boldsymbol{x_i}\in\mathbb{R}^{n+1},\boldsymbol{w}\in\mathbb{R}^{n+1}$，有此可将最小化问题进一步简化：

$$\min_{\boldsymbol{w}} L(\boldsymbol{w})=\min_{\boldsymbol{w}}\sum_{\boldsymbol{x_i}\in M}(\hat{y}_i-y_i)\boldsymbol{w}^T\boldsymbol{x_i}$$

假设误分类样本集合 M 固定，那么可以求得损失函数 $L(\boldsymbol{w})$ 的梯度

$$\nabla_{\boldsymbol{w}}L(\boldsymbol{w})=\sum_{\boldsymbol{x_i}\in M}(\hat{y}_i-y_i)\boldsymbol{x_i}$$

感知机的学习算法具体采用的是**随机梯度下降法**，即在最小化的过程中，不是一次使 M 中所有误分类点的梯度下降，而是一次随机选取一个误分类点，并使其梯度下降。所以权重 $\boldsymbol{w}$ 的更新式为

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \Delta\boldsymbol{w},$$
$$\Delta\boldsymbol{w} = -\eta(\hat{y}_i - y_i)\boldsymbol{w} = \eta(y_i - \hat{y}_i)\boldsymbol{w}$$

即 $\boldsymbol{w}$ 中的某个分量 w_i 的更新式即式 5.2。

备注：这里随机的意义在于每次调整的权重 w 不会对某个/些样本点产生依赖，即常说的“过拟合”。

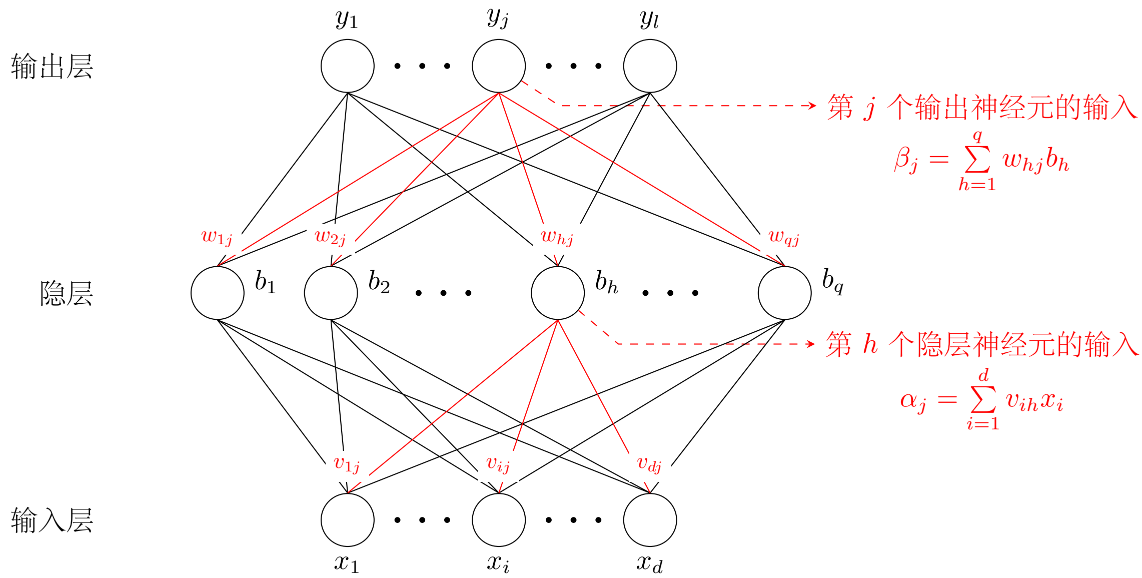
本章其余名词：

神经元模型	
名称	英文
功能神经元	functional neuron
线性可分	linearly separable
收敛	converge
震荡	fluctuation
隐层或隐含层	hidden layer
多层前馈神经网络	multi-layer feedforward neural networks
连接权	connection weight

5.3 误差逆传播算法

误差逆传播算法（error BackPropagation，简称 BP）。

给定训练集 $D = \{(\boldsymbol{x}_1, \boldsymbol{y}_1), (\boldsymbol{x}_2, \boldsymbol{y}_2), \ldots, (\boldsymbol{x}_m, \boldsymbol{y}_m)\}$, $\boldsymbol{x}_i \in \mathbb{R}^d$, $\boldsymbol{y}_i \in \mathbb{R}^l$，即 input 为 d 维，output l 维。如图：



如图所示，该神经网络是一个拥有 d 个输入神经元，l 个输出神经元，q 个隐层神经元的多层前馈网络结构。

其中输出层第 j 个神经元的阈值用 θ_j 表示，隐层第 h 个神经元的阈值用 γ_h 表示；输入层第 i 个神经元与隐层第 h 个神经元之间的连接权位 v_{ih}，隐层第 h 个神经元与输出层第 j 个神经元之间的连接权位 w_{hj}。

其中隐层的第 h 个神经元接收到的输入（图中输入层与隐层之间的三条红线所示）为 $\alpha_h = \sum_{i=1}^{d} v_{ih} x_i$；输出层的第 j 个神经元接收到的输入（图中隐层与输出层之间的四条红线所示）为 $\beta_j = \sum_{h=1}^{q} w_{hj} b_h$，其中 b_h 为隐层第 h 个神经元的输出。

对训练例 $(\boldsymbol{x}_k, \boldsymbol{y}_k)$ 假设神经网络的输出为 $\hat{\boldsymbol{y}}_k = (\hat{y}_1^k, \hat{y}_2^k, \ldots, \hat{y}_l^k,)$，即

$$\hat{y}_j^k = f(\beta_j - \theta_j) \tag{5.3}$$

这里的真实值 $y_j^k = (y_1^k, y_2^k, \ldots, y_l^k)^2$，那么对于某第 i 个输出层的神经元而言，其均方误差即

$$\frac{(\hat{y}_i^k - y_i^k)}{2}$$

那么将所有神经元加总，就有了网络在 $(\boldsymbol{x}_k, \boldsymbol{y}_k)$ 上的均方误差为：

$$E_k = \frac{1}{2} \sum_{j=1}^{l} (\hat{y}_j^k - y_j^k)^2 \tag{5.4}$$

另外上图的网络中有 $(d+l+l)q+l$ 个参数需确定：输入层到隐层的 $d\times q$ 个权值、隐层到输出层的 $q\times l$ 个权值、q 个隐层神经元的阈值、l 个输出层神经元的阈值。BP 是一个迭代学习算法，在迭代的每一轮中采用广义的感知机学习规则对参数进行更新估计，即与式 5.1 类似，任意参数 v 的更新估计式为

$$v \leftarrow v + \Delta v. \tag{5.5}$$

BP 算法基于 **梯度下降（gradient descent）** 策略，以目标的负梯度方向对参数进行调整。对式 5.4 的误差 E_k，给定学习率 η，有

$$\Delta w_{hj} = -\eta \frac{\partial E_k}{\partial w_{hj}} \tag{5.6}$$

这里提到的负梯度，是利用了导数趋近于零时可以得到最小值的特性，去求得均方误差 E_k 的最小值。回顾上文中的感知机学习算法：

> ... 随机选取一个误分类点并使其梯度下降，所以权重 $\boldsymbol{w}$ 的更新式为
>
> $$\boldsymbol{w} \leftarrow \boldsymbol{w} + \Delta \boldsymbol{w},$$
> $$\Delta \boldsymbol{w} = -\eta(\hat{y}_i - y_i)\boldsymbol{w} = \eta(y_i - \hat{y}_i)\boldsymbol{w}$$

式 5.6 与感知机的最小化损失函数的不同之处在于：感知机是有两层神经元构成的，且输出层只有一个神经元，因此只需要考虑一层的权重变化，即 $(\hat{y}_i - y_i)\boldsymbol{w}$；而对于由若干个感知机所构成的神经网络而言，所有权重的变化则变为了 $\frac{\partial E_k}{\partial w_{hj}}$。
那么对于多层的神经网络，式 5.4 仅表现出了最后一层隐层与输出层的权重变化，那么就有了疑问：其它层之间的权重好像并不会被式 5.6 所改变？接下来继续，注意到 $w_h j$ 先影响到第 j 个输出层神经元的输入值 β_j。根据图示其为隐层所有神经元与其本身的表达式，即

$$\beta_j = \sum_{h=1}^{q} w_{hj} b_h$$

接着 w_{hj} 再影响到其他输出值 $\hat{y}_j^k$，最后再影响到 E_k，那么根据求导的链式法则有：

$$\frac{\partial E_k}{\partial w_{hj}} = \frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j} \cdot \frac{\partial \beta_j}{\partial w_{hj}} \tag{5.7}$$

根据 β_j 的定义，有：

$$\begin{aligned}
\frac{\partial \beta_j}{\partial w_{hj}} &= \frac{\partial(\sum_{h=1}^{q} w_{hj} b_h)}{\partial w_{hj}} \\
&= \frac{\partial(w_{1j}b_1 + w_{2j}b_2 + \cdots + w_{hj}b_h + \cdots + w_{qj}b_q)}{\partial w_{hj}} \\
&= 0 + 0 + \cdots + b_h + \cdots + 0 \\
&= b_h
\end{aligned} \tag{5.8}$$

而又根据 Sigmoid 函数的一个很好的性质：

$$f'(x) = f(x)(1 - f(x)) \tag{5.9}$$

根据式 5.4 与式 5.3 有：

$$\begin{aligned} g_j &= -\frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j} \\ &= -(\hat{y}_j^k - y_j^k) f'(\beta_j - \theta_j) \\ &= \hat{y}_j^k (1 - \hat{y}_j^k)(y_j^k - \hat{y}_j^k) \end{aligned} \tag{5.10}$$

将式 5.10 与式 5.8 代入式 5.7，再代入式 5.6，就得到了 BP 算法中关于 w_{hj} 的更新公式：

$$\begin{aligned} \Delta w_{hj} &= -\eta \frac{\partial E_k}{\partial w_{hj}} \\ &= -\eta \frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j} \cdot \frac{\partial \beta_j}{\partial w_{hj}} \\ &= \eta \hat{y}_j^k (1 - \hat{y}_j^k)(y_j^k - \hat{y}_j^k) \cdot b_h \\ &= \eta g_j b_h \end{aligned} \tag{5.11}$$

类似可得：

$$\Delta \theta_j = -\eta g_j \tag{5.12}$$

$$\Delta v_{ih} = \eta e_h x_i \tag{5.13}$$

$$\Delta \gamma_h = -\eta e_h \tag{5.14}$$

5.4 全局最小与局部最小

WIP

6 支持向量机

7 贝叶斯分类器

8 集成学习

9 聚类

10 降维与度量学习

11 特征选择与稀疏学习

12 计算学习理论

13 半监督学习

14 概率图模型

15 规则学习

16 强化学习