机器学习笔记

Leonhard Hsiao

2025 年 11 月 4 日

目录

1 线性分类

1.1 感知机

1.1.1 基本知识

感知机是神经网络发展早期的重要模型，由 Rosenblatt 于 1957 年提出。其发展历程中的关键贡献包括：

- McCulloch & Pitts (1943) 引入神经网络概念并提出 M-P 数学模型
- Hebb (1949) 提出首个自组织学习规则
- Rosenblatt (1957) 将感知机确立为有教师学习模型

定义 1.1 (线性分类决策函数). 感知机通过线性决策函数实现二分类：

$$g(\boldsymbol{x}) = \sum_{i=1}^{m} w_i x_i + w_0 = \boldsymbol{w}^T \boldsymbol{x} + w_0 \tag{1}$$

其中 $\boldsymbol{w}$ 为权重向量，w_0 为偏置项。

采用增广表示可简化表达式。令：

$$\tilde{\boldsymbol{w}} = \begin{pmatrix} \boldsymbol{w} \\ w_0 \end{pmatrix}, \quad \tilde{\boldsymbol{x}} = \begin{pmatrix} \boldsymbol{x} \\ 1 \end{pmatrix} \tag{2}$$

则决策函数可写为：

$$g(\boldsymbol{x}) = \tilde{\boldsymbol{w}}^T \tilde{\boldsymbol{x}} \tag{3}$$

定义 1.2 (分类规则). 决策超平面 $g(\boldsymbol{x}) = 0$ 将特征空间划分为两个区域：

$$若 g(\boldsymbol{x}) > 0 \Rightarrow 将 \boldsymbol{x} 划分到 \omega_1 类 \tag{4}$$

$$若 g(\boldsymbol{x}) < 0 \Rightarrow 将 \boldsymbol{x} 划分到 \omega_2 类 \tag{5}$$

决策函数几何意义的详细证明

设 $\boldsymbol{x}$ 为特征空间中的任意样本点，决策超平面为 $\boldsymbol{w}^T\boldsymbol{x} + w_0 = 0$。

步骤 *1*：超平面的法向量表示

超平面的法向量为 $\boldsymbol{w}$，因为对于超平面上任意两点 $\boldsymbol{x}_1, \boldsymbol{x}_2$，有：

$$\boldsymbol{w}^T\boldsymbol{x}_1 + w_0 = 0 \tag{6}$$

$$\boldsymbol{w}^T\boldsymbol{x}_2 + w_0 = 0 \tag{7}$$

两式相减得：$\boldsymbol{w}^T(\boldsymbol{x}_1 - \boldsymbol{x}_2) = 0$，故 $\boldsymbol{w}$ 垂直于超平面。

步骤 *2*：点到超平面的距离推导

在超平面上任取一点 $\boldsymbol{x}_0$，满足 $\boldsymbol{w}^T\boldsymbol{x}_0 + w_0 = 0$。样本点 $\boldsymbol{x}$ 到超平面的有向距离 z 可表示为向量 $\boldsymbol{x} - \boldsymbol{x}_0$ 在法向量方向上的投影：

$$z = \frac{\boldsymbol{w}^T(\boldsymbol{x} - \boldsymbol{x}_0)}{\|\boldsymbol{w}\|} \tag{8}$$

$$= \frac{\boldsymbol{w}^T\boldsymbol{x} - \boldsymbol{w}^T\boldsymbol{x}_0}{\|\boldsymbol{w}\|} \tag{9}$$

由 $\boldsymbol{w}^T\boldsymbol{x}_0 = -w_0$，代入得：

$$z = \frac{\boldsymbol{w}^T\boldsymbol{x} + w_0}{\|\boldsymbol{w}\|} = \frac{g(\boldsymbol{x})}{\|\boldsymbol{w}\|} \tag{10}$$

步骤 *3*：决策函数与距离的关系

由上式可得：

$$g(\boldsymbol{x}) = \|\boldsymbol{w}\| \cdot z \tag{11}$$

这表明决策函数值的绝对值 $|g(\boldsymbol{x})|$ 正比于样本点到超平面的欧氏距离 $|z|$，比例系数为法向量的模长 $\|\boldsymbol{w}\|$。

此几何解释说明：$g(\boldsymbol{x})$ 不仅提供分类决策，还量化了分类的置信度。

验证函数 $y_i(\boldsymbol{w}^T\boldsymbol{x}_i + w_0)$ 用于判断分类正确性。对于正确分类的样本：

$$y_i = +1 \Rightarrow \boldsymbol{w}^T\boldsymbol{x}_i + w_0 \geq 0 \tag{12}$$

$$y_i = -1 \Rightarrow \boldsymbol{w}^T\boldsymbol{x}_i + w_0 \leq 0 \tag{13}$$

统一表示为：

$$y_i(\boldsymbol{w}^T\boldsymbol{x}_i + w_0) \geq 0 \tag{14}$$

定义 1.3 (优化目标). 感知机学习的目标是最小化损失函数：

$$\min_{\boldsymbol{w}} J(\boldsymbol{w}) = \sum_i J_i(\boldsymbol{w}) \tag{15}$$

其中 $J_i(\boldsymbol{w})$ 为单个样本的损失。

例 1.4 (多层前馈神经网络的梯度下降算法). 该算法展示了标准反向传播（BP）算法的完整流程，是神经网络训练的核心方法。

Algorithm 1 多层前馈神经网络的梯度下降训练算法

输入： 训练集 $D = \{(x_k, y_k)\}_{k=1}^m$; 学习率 η

1: 在 (0,1) 范围内随机初始化网络中所有连接权和阈值
2: **repeat**
3: **for** 所有 $(x_k, y_k) \in D$ **do**
4: 根据当前参数计算当前样本的输出 $\hat{y}_k$ ▷ 前向传播
5: 计算输出层神经元的梯度项 g_j ▷ 输出层误差
6: 计算隐层神经元的梯度项 e_h ▷ 隐层误差反向传播
7: 更新连接权 w_{hj}, v_{ih} 与阈值 θ_j, γ_h ▷ 参数更新
8: **end for**
9: **until** 达到停止条件

输出： 连接权与阈值确定的多层前馈神经网络

算法详细解释：

1. 初始化策略（第 1 行）

$$w_{ij} \sim U(0,1) \times \text{小尺度因子} \tag{16}$$

$$\theta_j \sim U(0,1) \times \text{小尺度因子} \tag{17}$$

小范围随机初始化有助于避免梯度消失/爆炸问题。

2. 前向传播过程（第 4 行）对于第 h 个隐层神经元：

$$\alpha_h = \sum_{i=1}^{d} v_{ih}x_i \quad \text{（输入加权和）} \tag{18}$$

$$b_h = f(\alpha_h - \gamma_h) \quad \text{（激活函数）} \tag{19}$$

对于输出层神经元：

$$\beta_j = \sum_{h=1}^{q} w_{hj} b_h \quad （隐层输出加权和） \tag{20}$$

$$\hat{y}_j = f(\beta_j - \theta_j) \quad （最终输出） \tag{21}$$

3. 误差反向传播（第 5-6 行）输出层梯度项：

$$g_j = -\frac{\partial E_k}{\partial \hat{y}_j} \cdot \frac{\partial \hat{y}_j}{\partial \beta_j} = (\hat{y}_j - y_j) f'(\beta_j - \theta_j) \tag{22}$$

隐层梯度项：

$$e_h = -\frac{\partial E_k}{\partial b_h} \cdot \frac{\partial b_h}{\partial \alpha_h} = \left(\sum_{j=1}^{l} w_{hj} g_j \right) f'(\alpha_h - \gamma_h) \tag{23}$$

4. 参数更新规则（第 7 行）连接权更新：

$$\Delta w_{hj} = \eta g_j b_h, \quad w_{hj} \leftarrow w_{hj} + \Delta w_{hj} \tag{24}$$

$$\Delta v_{ih} = \eta e_h x_i, \quad v_{ih} \leftarrow v_{ih} + \Delta v_{ih} \tag{25}$$

阈值更新：

$$\Delta \theta_j = -\eta g_j, \quad \theta_j \leftarrow \theta_j + \Delta \theta_j \tag{26}$$

$$\Delta \gamma_h = -\eta e_h, \quad \gamma_h \leftarrow \gamma_h + \Delta \gamma_h \tag{27}$$

5. 停止条件（第 9 行）

- 达到最大训练周期数
- 验证集误差不再显著下降
- 梯度范数小于阈值：$\|\nabla J(\boldsymbol{w})\| < \epsilon$

算法特点：

- 这是标准的随机梯度下降（SGD）实现，逐个样本更新
- 通过误差反向传播高效计算梯度
- 适用于任意层数的前馈神经网络

例 1.5 (梯度下降算法中的批次与周期概念). 在梯度下降算法中，有几个关键概念需要理解：

1. 批次大小与更新策略

- **批量梯度下降（Batch GD）**：使用整个训练集计算梯度

$$\boldsymbol{w} = \boldsymbol{w} - \eta \sum_{i=1}^{N} \nabla J_i(\boldsymbol{w}) \tag{28}$$

其中 N 为训练样本总数。每次更新需要遍历所有样本，计算开销大但稳定性好。

- **随机梯度下降（SGD）**：每次随机选择一个样本更新

$$\boldsymbol{w} = \boldsymbol{w} - \eta \nabla J_i(\boldsymbol{w}), \quad i \in \{1, 2, \ldots, N\} \tag{29}$$

更新频繁，收敛快但波动较大。

- **小批量梯度下降（Mini-batch GD）**：折中方案，每次使用一个小批次

$$\boldsymbol{w} = \boldsymbol{w} - \eta \sum_{i=1}^{B} \nabla J_i(\boldsymbol{w}) \tag{30}$$

其中 B 为批次大小，通常取 32、128 等。

2. 周期与迭代的概念

- **周期（Epoch）**：整个训练集被完整使用一次的过程

$$1 \text{ 个周期} = \text{遍历所有} N \text{ 个训练样本} \tag{31}$$

- **迭代（Iteration）**：一次权重更新的过程

$$\text{迭代次数} = \frac{\text{样本总数}}{\text{批次大小}} \times \text{周期数} \tag{32}$$

- 不同方法在一个周期内的迭代次数：

$$\text{Batch GD:} \quad 1 \text{ 次迭代/周期} \tag{33}$$

$$\text{SGD:} \quad N \text{ 次迭代/周期} \tag{34}$$

$$\text{Mini-batch:} \quad \lceil N/B \rceil \text{ 次迭代/周期} \tag{35}$$

3. 学习率调度策略学习率 η 通常随时间衰减以改善收敛：

$$\eta_t = \frac{\eta_0}{1 + \alpha t} \quad \text{或} \quad \eta_t = \eta_0 \cdot \beta^t \tag{36}$$

其中 t 为周期计数，α, β 为衰减系数。

4. 收敛判断标准训练通常在满足以下条件时停止：

$$\left\| \boldsymbol{w}^{(t+1)} - \boldsymbol{w}^{(t)} \right\| < \epsilon \quad \text{或} \quad J(\boldsymbol{w}) < \delta \tag{37}$$

其中 ϵ, δ 为预设的收敛阈值。

1.1.2 感知机学习准则

感知机学习准则的核心思想是通过调整权重向量，使错分样本的代价最小化；从而将决策超平面推向能够正确划分训练集的位置。下面从结构、损失定义、优化推导与收敛性质给出更为细化的说明。

定义 1.6 (感知机基本结构). 根据第一张图片，感知机的基本结构包含以下关键组成部分：

$$\text{输入向量：}\quad \boldsymbol{x}(n) = [x_0(n), x_1(n), \ldots, x_m(n)]^T, \quad x_0(n) \equiv +1 \tag{38}$$

$$\text{权重向量：}\quad \boldsymbol{w}(n) = [w_0(n), w_1(n), \ldots, w_m(n)]^T, \quad w_0(n) \equiv b \tag{39}$$

$$\text{局部感受域：}\quad v(n) = \sum_{i=0}^{m} w_i(n) x_i(n) = \boldsymbol{w}^T(n)\boldsymbol{x}(n) \tag{40}$$

$$\text{输出：}\quad y(n) = \varphi\big(v(n)\big) \tag{41}$$

第一张图片清晰地展示了信号流：输入经线性组合器得到 v，再由硬激活函数产生输出。

其中 $x_0 \equiv 1$ 为偏置输入，$w_0 \equiv b$ 为其权重，$\varphi(\cdot)$ 为硬激活函数。

定义 1.7 (硬激活函数与符号约定). 硬激活函数（Hard Limiter）定义为

$$\varphi(v) = \begin{cases} +1, & v \geq 0 \\ -1, & v < 0 \end{cases} \tag{42}$$

等价地，$y(n) = \operatorname{sign}\big(v(n)\big)$，并采用 $\operatorname{sign}(0) = +1$ 的并列判别约定（tie-breaking）。该函数将连续的线性输出 v 映射为离散标签 $\{-1, +1\}$，完成二分类决策。

硬激活为确定性阶跃，便于将线性判别与类别标签直接对应。

定义 1.8 (感知机学习目标：错分加和形式). 根据第二张图片，经典的感知机学习准则最小化错分样本的线性代价

$$\min_{\boldsymbol{w}} \; J_1(\boldsymbol{w}) \;=\; \sum_{\boldsymbol{x}(n) \in E(\boldsymbol{w})} \big(-\, d(n)\, \boldsymbol{w}^T \boldsymbol{x}(n)\big), \tag{43}$$

其中 $E(\boldsymbol{w}) = \{\boldsymbol{x}(n) \,:\, d(n)\, \boldsymbol{w}^T \boldsymbol{x}(n) < 0\}$ 为在当前 $\boldsymbol{w}$ 下被错分的样本集合，$d(n) \in \{-1, +1\}$ 为真实标签。

$d(n)$ 是样本 $\boldsymbol{x}(n)$ 的真实类别标签，在监督学习中作为”教师信号”提供。当 $d(n) = +1$ 时表示样本属于正类，$d(n) = -1$ 时表示属于负类。

从构造动机看，若样本被错分，则 $d(n)\, \boldsymbol{w}^T \boldsymbol{x}(n) < 0$，这可以等价为另一种形式，$-d(n)\, \boldsymbol{w}^T \boldsymbol{x}(n) > 0$，推动 $\boldsymbol{w}$ 朝使其被纠正的方向更新；若样本已被正确分类，则其不进入和式，不产生代价。

定理 1.9 (等价的铰链式表示). 上述目标等价于对全体样本求和的“零间隔铰链”形式

$$J_2(\boldsymbol{w}) \;=\; \sum_{n=1}^{N} \max\big(0,\; -\, d(n)\, \boldsymbol{w}^T \boldsymbol{x}(n)\big). \tag{44}$$

证明. 当 $d(n)\, \boldsymbol{w}^T \boldsymbol{x}(n) \geq 0$（样本正确）时，$\max(0, \cdot) = 0$；当其 < 0（样本错误）时，$\max(0, \cdot) = -\, d(n)\, \boldsymbol{w}^T \boldsymbol{x}(n)$，与 J_1 中对错分样本的求和完全一致。 □

定理 1.10 (基于预测误差的等价形式). 还可写作

$$J_3(\boldsymbol{w}) \;=\; \sum_{n=1}^{N} -\,\boldsymbol{w}^T\boldsymbol{x}(n)\,\big[d(n)-y(n)\big], \tag{45}$$

其中 $y(n)=\operatorname{sign}\big(\boldsymbol{w}^T\boldsymbol{x}(n)\big)\in\{-1,+1\}$。

等价性说明. 若 $d(n)=y(n)$，则 $d(n)-y(n)=0$，该项为 0；若 $d(n)\neq y(n)$，则 $d(n)-y(n)=\pm 2$，从而 $-\,\boldsymbol{w}^T\boldsymbol{x}(n)\,[d(n)-y(n)]=-2\,d(n)\,\boldsymbol{w}^T\boldsymbol{x}(n)$，与 J_1 在错分样本上仅差常数倍，因此优化方向一致。 □

在优化实践中，J_2 更适合用于分析，因为它避免显式出现 $y(n)$ 对 $\boldsymbol{w}$ 的非光滑依赖，且可用次梯度统一处理。

命题 1.1 (次梯度与更新规则的推导). 对 $J_2(\boldsymbol{w})$ 的每个样本项 $\ell_n(\boldsymbol{w})=\max(0,-d(n)\boldsymbol{w}^T\boldsymbol{x}(n))$，其次梯度为

$$\partial\ell_n(\boldsymbol{w})=\begin{cases}\{\mathbf{0}\}, & d(n)\,\boldsymbol{w}^T\boldsymbol{x}(n)>0,\\ conv\big(\{\mathbf{0},-d(n)\boldsymbol{x}(n)\}\big), & d(n)\,\boldsymbol{w}^T\boldsymbol{x}(n)=0,\\ \{-d(n)\boldsymbol{x}(n)\}, & d(n)\,\boldsymbol{w}^T\boldsymbol{x}(n)<0.\end{cases} \tag{46}$$

采用随机次梯度下降（*SGD*）并取边界处的代表元 $-d(n)\boldsymbol{x}(n)$，得到单样本更新

$$\boldsymbol{w}\leftarrow\boldsymbol{w}-\eta\big(-d(n)\boldsymbol{x}(n)\big)\;=\;\boldsymbol{w}+\eta\,d(n)\,\boldsymbol{x}(n), \tag{47}$$

且仅在 $d(n)\,\boldsymbol{w}^T\boldsymbol{x}(n)\le 0$ 时发生更新。这与经典 *PLA* 更新完全一致。

Algorithm 2 感知机学习算法（随机次梯度/PLA 形式）

输入: 训练集 $\{(\boldsymbol{x}(n),d(n))\}_{n=1}^{N}$，学习率 $\eta>0$

随机初始化 $\boldsymbol{w}$
repeat
  随机打乱样本顺序
  for $n=1$ 到 N **do**
    $v\leftarrow\boldsymbol{w}^T\boldsymbol{x}(n)$；$y\leftarrow\varphi(v)$
    if $d(n)\,v\le 0$ **then** ▷ 仅对错分或临界样本更新
      $\boldsymbol{w}\leftarrow\boldsymbol{w}+\eta\,d(n)\,\boldsymbol{x}(n)$
    end if
  end for
until 达到停止条件：如连续一轮零错误，或迭代/时间上限

输出: 训练得到的权重向量 $\boldsymbol{w}$

权重更新规则的几何解释

当 $d(n)\,\boldsymbol{w}^T\boldsymbol{x}(n) \le 0$ 时，更新 $\boldsymbol{w} \leftarrow \boldsymbol{w} + \eta\, d(n)\boldsymbol{x}(n)$ 将 $\boldsymbol{w}$ 在 $\boldsymbol{x}(n)$ 的方向上（或其相反方向）作线性平移，使 $d(n)\,\boldsymbol{w}^T\boldsymbol{x}(n)$ 增大，从而把样本推向被正确分类的一侧。

定理 1.11 (感知机收敛定理). 若训练集线性可分，存在 $\boldsymbol{w}^\star$ 与间隔 $\gamma > 0$ 使得

$$d(n)\,\boldsymbol{w}^{\star T}\boldsymbol{x}(n) \ge \gamma, \quad \forall n, \tag{48}$$

且设 $R = \max_n \|\boldsymbol{x}(n)\|$，则感知机在有限步内收敛，错误更新次数上界满足

$$T_{\max} \;\le\; \frac{\|\boldsymbol{w}^\star\|^2\, R^2}{\gamma^2}. \tag{49}$$

该定理保证了在可分情形下以常数学习率进行的在线更新会在有限次错误后终止。

实现与训练细节（更务实的要点）

- **学习率 η：** 可取常数（如 $10^{-3}\sim 10^{-1}$），或按 $\eta_t = \eta_0/(1+t)$ 衰减；也可采用样本归一化步长 $\eta_t = \eta_0/\|\boldsymbol{x}(n)\|^2$ 提升数值稳定性。
- **特征尺度：** 对输入做标准化/归一化（如 ℓ_2 归一或零均值单位方差）有助于更快收敛并减少振荡。
- **打乱与遍历：**每轮随机打乱样本顺序可降低顺序偏置;对临界点 $v = 0$ 建议也更新一次（上式已覆盖）。
- **停止条件：** 常用为“连续一整轮零错误”或“达到最大迭代/时间预算”。线性不可分时不保证停止。
- **不可分情况下：** 可采用 Pocket（口袋）策略：维护当前最佳（在验证集或训练集上错误最少）的 $\boldsymbol{w}_{\text{best}}$，在线更新同时保留最好解。
- **与间隔的关系：**感知机损失为零间隔铰链 $\max(0, -d\,\boldsymbol{w}^T\boldsymbol{x})$;相比 SVM 的 $\max(0, 1 - d\,\boldsymbol{w}^T\boldsymbol{x})$，不强制正间隔，因此对“近边界但正确”的样本不产生梯度。
- **复杂度：** 单次更新为 $O(m)$；总开销约为 $O(m \times T)$，其中 T 为实际发生的错误更新次数。

感知机学习准则虽然形式简洁，但在优化层面可视作对分段线性凸上界（J_2）执行在线次梯度法；其“仅错分更新”的机制在可分数据上保证有限步收敛，在不可分数据上亦可通过 Pocket 等策略获得稳定的近似解。实际训练中，建议从较小 η 起步，配合特征标准化与打乱遍历，并采用明确的