第 2 单元　机器学习基础知识

支撑的课程目标

1. 能够基于智能信息处理的基本理论和技术，识别和理解数据处理与分析等问题的相关特性.
2. 能够运用智能信息处理的相关原理和专业知识，设计实验方案，为解决数据处理与分析等问题提供支持.

基本要求

1. 能够详述监督学习与非监督学习的区别，理解两种模型的特点，根据处理数据的特性选择合适的机器学习方法.
2. 能够应用评估参数评价模型优劣，运用交叉验证和损失函数正则化来确定模型参数.

教学重点与难点

重点：　模型参数选择；监督学习.

难点：　无监督学习.

教学过程设计

新课导入、知识讲授、教学目标达成考核、总结.

教学过程设计

本单元教学通过“互动、开放”的课堂形式，采用探究式学习、问题导入的教学方法，激发学生的学习兴趣，促成课程目标的达成.

教学学时

4 学时.

一、导入新课（5 分钟）

深度学习是作为机器学习的一个分支发展而来的，因此有必要介绍机器学习的基础知识。本章首先介绍模型评估与模型参数选择，这些知识在深度学习中具有相当重要的地位。而后，本单元简要介绍了监督学习与非监督学习。大多数基础的深度学习模型都是基于监督学习的，但是随着模型复杂度的提高，模型对数据的需求量也日益增加。因此，许多研究者都在尝试将非监督学习应用到深度学习中，以获得更佳廉价的训练数据。

二、新课讲授 (170 分钟)

本单元要点

* 模型评估与模型参数选择
* 验证
* 正则化
* 监督学习
* 非监督学习
* 梯度下降算法

1. 模型评估与模型参数选择

若对于给定的输入 x，若某个模型的输出 $\widehat{y} = f(x)$ 偏离真实目标值 y，那么就说明模型存在误差；$\widehat{y}$ 偏离 y 的程度可以用关于 $\widehat{y}$ 和 y 某个函数 $L(y, \widehat{y})$ 来表示，作为误差的度量标准：这样的函数 $L(y, \widehat{y})$ 称为损失函数。

在某种损失函数度量下，训练集上的平均误差被称为训练误差，测试集上的误差称为泛化误差。由于我们训练得到一个模型最终的目的是为了在未知的数据上得到尽可能准确的结果，因此泛化误差是衡量一个模型泛化能力的重要标准。

训练集可能存在以下问题：(1) 训练集样本太少，缺乏代表性；(2) 训练集中本身存在错误的样本，即噪声。如果片面地追求训练误差的最小化，就会导致模型参数复杂度增加，使得模型过拟合。

为了选择效果最佳的模型，防止过拟合的问题，通常可以采取的方法有：(1) 使用验证集调参 (2) 对损失函数进行正则化

2. 验证

模型不能过拟合于训练集，否则将不能在测试集上得到最优结果；但是否

能直接以测试集上的表现来选择模型参数呢？答案是否定的。因为这样的模型参数将会是针对某个特定测试集的，得出来的评价标准将会失去其公平性，失去了与其他同类或不同类模型相比较的意义。

因此参数的选择（即调参）必须在一个独立于训练集和测试集的数据集上进行，这样的用于模型调参的数据集被称为开发集或验证集。

然而很多时候我们能得到的数据量非常有限。这个时候我们可以不显式地使用验证集，而是重复使用训练集和测试集，这种方法称为交叉验证。常用的交叉验证方法有：(1) 简单交叉验证（2）K-重交叉验证（3）留一法

2.1 简单的交叉验证：

将原始数据随机分为两组，一组做为训练集，一组做为验证集，利用训练集训练分类器，然后利用验证集验证模型，记录最后的分类准确率为此分类器的性能指标。

好处：处理简单，只需随机把原始数据分为两组即可

坏处：但没有达到交叉的思想，由于是随机的将原始数据分组，所以最后验证集分类准确率的高低与原始数据的分组有很大的关系，得到的结果并不具有说服性。

2.2 2-折交叉验证（2-fold Cross Validation，记为 2-CV）

做法是将数据集分成两个相等大小的子集，进行两次的分类器训练。在第一次中，一个子集作为训练集，另一个便作为测试集；在第二次训练中，则将训练集与测试集对换。

其并不常用：主要原因是训练集样本数太少，不足以代表母体样本的分布，导致 te 测试阶段辨识率容易出现明显落差。

2.3 K-折交叉验证（K-fold Cross Validation，记为 K-CV）

将原始数据分成 K 组（一般是均分），将每个子集数据轮换做一次验证集，其余的 K-1 组子集数据作为训练集，这样会得到 K 个模型，用这 K 个模型最终的验证集的分类准确率的平均数作为此 K-CV 下分类器的性能指标。K 一般大于等于 2，实际操作时一般从 3 开始取，只有在原始数据集合数据量小的时候才会尝试取 2。

应用最多，K-CV 可以有效的避免过拟合与欠拟合的发生，最后得到的结果

也比较具有说服性。

Eg: 十折交叉验证

（1）将数据集分成十份，轮流将其中 9 份作为训练数据，1 份作为测试数据，进行实验。每次试验都会得出相应的正确率。

（2）10 次的结果的正确率的平均值作为对算法精度的估计，一般还需要进行多次 10 折交叉验证（例如 10 次 10 折交叉验证），再求其均值，作为对算法准确性的估计。

此外：

（1）多次 k 折交叉验证再求均值，例如：10 次 10 折交叉验证，以求更精确一点。

（2）划分时有多种方法，例如对非平衡数据可以用分层采样，就是在每一份子集中都保持和原始数据集相同的类别比例。

（3）模型训练过程的所有步骤，包括模型选择，特征选择等都是在单个折叠 fold 中独立执行的。

2.4 留一交叉验证（Leave-One-Out Cross Validation 记为 LOO-CV）

在数据缺乏的情况下使用，如果设原始数据有 N 个样本，那么 LOO-CV 就是 N-CV，即每个样本单独作为验证集，其余的 N-1 个样本作为训练集，故 LOO-CV 会得到 N 个模型，用这 N 个模型最终的验证集的分类准确率的平均数作为此下 LOO-CV 分类器的性能指标。

优点：每一回合中几乎所有的样本皆用于训练模型，因此最接近原始样本的分布，这样评估所得的结果比较可靠；实验过程中没有随机因素会影响实验数据，确保实验过程是可以被复制的。

缺点：计算成本高，需要建立的模型数量与原始数据样本数量相同。当数据集较大时几乎不能使用。

3. 正则化

为了避免过拟合，需要选择参数复杂度最小的模型。这是因为如果有两个效果相同的模型，而它们的参数复杂度不相同，那么冗余的复杂度一定是由于过拟合导致的。为了选择复杂度较小的模型，一种策略是在优化目标中加入正

则化项，以惩罚冗余的复杂度：

$$\min_{\theta} L(y, \widehat{y}; \theta) + \lambda \cdot J(\theta)$$

其中 θ 为模型参数，$L(y, \widehat{y})$ 为原来的损失函数，$J(\theta)$ 是正则化项，λ 用于调整正则化项的权重。正则化项通常为 θ 的某阶向量范数。

4. 监督学习与非监督学习

模型与最优化算法的选择，很大程度上取决于能得到什么样的数据。如果数据集中样本点只包含了模型的输入 x，那么就需要采用非监督学习的算法；如果这些样本点以 (x, y) 这样的输入-输出二元组的形式出现，那么就可以采用监督学习的算法。

4.1 监督学习

在监督学习中，我们根据训练集 $\{(x_i, y_i\}_{i=1}^{N}$ 中的观测样本点来优化模型 $f(\cdot)$，使得给定测试样例 x' 作为模型输入，其输出 $\widehat{y}$ 尽可能接近正确输出 y'。

监督学习算法主要适用于两大类问题：回归和分类。这两类问题的区别在于：回归问题的输出是连续值，而分类问题的输出是离散值。

4.1.1 回归

回归问题在生活中非常常见，其最简单的形式是一个连续函数的拟合。如果一个购物网站想要计算出其在某个时期的预期收益，研究人员会将相关因素如广告投放量、网站流量、优惠力度等纳入自变量，根据现有数据拟合函数，得到在未来某一时刻的预测值。回归问题中通常使用均方损失函数来作为度量模型效果的指标，最简单的求解例子是最小二乘法。

4.1.2 分类

分类问题也是生活中非常常见的一类问题，例如我们需要从金融市场的交易记录中分类出正常的交易记录以及潜在的恶意交易。度量分类问题的指标通常为准确率（Accuracy）：对于测试集中 D 个样本，有 k 个被正确分类，D-k 个被错误分类，则准确率为：

$$Accuracy = \frac{k}{D}$$

然而在一些特殊的分类问题中，属于各类的样本的并不是均一分布，甚至其出现概率相差很多个数量级，这种分类问题称为不平衡类问题。在不平衡类问题中，准确率并没有多大意义，我们需要一些别的指标。通常在不平衡类问题中，我们使用 F-度量来作为评价模型的指标。以二元不平衡分类问题为例，这种分类问题往往是异常检测，模型的好坏往往取决于能否很好地检出异常，同时尽可能不误报异常。定义占样本少数的类为正类（Positive class），占样本多数的为负类（Negative class），那么预测只可能出现 4 种状况：

将正类样本预测为正类（True Positive, TP）、将负类样本预测为正类（False Positive, FP）、将正类样本预测为负类（False Negative, FN）、将负类样本预测为负类（True Negative, TN）

定义召回率（recall）：

$$R = \frac{|TP|}{|TP| + |FN|}$$

召回率度量了在所有的正类样本中，模型正确检出的比率，因此也称为查全率；

定义精确率（precision）：

$$P = \frac{|TP|}{|TP| + |FP|}$$

精确率度量了在所有被模型预测为正类的样本中，正确预测的比率，因此也称查准率。

F-度量则是在召回率与精确率之间去调和平均数；有时候在实际问题上，若我们更加看重其中某一个度量，还可以给它加上一个权值 α，称为 F_α-度量：

$$F_\alpha = \frac{(1+\alpha^2)RP}{R + \alpha^2 P}$$

特殊地，当 $\alpha = 1$ 时：

$$F_1 = \frac{2RP}{R+P}$$

可以看到，如果模型“不够警觉”，没有检测出一些正类样本，那么召回率就会受损；而如果模型倾向于“滥杀无辜”，那么精确率就会下降。因此较高的

F-度量意味着模型倾向于“不冤枉一个好人，也不放过一个坏人”，是一个较为适合不平衡类问题的指标。

4.2 非监督学习

在非监督学习中，数据集 $\{x_i\}_{i=1}^N$ 中只有模型的输入，而并不提供正确的输出 y_i 作为监督信号。非监督学习通常用于这样的分类问题：给定一些样本的特征值，而不给出它们正确的分类，也不给出所有可能的类别；而是通过学习确定这些样本可以分为哪些类别、它们各自都属于哪一类。这一类问题称为聚类。非监督学习得到的模型的效果应该使用何种指标来衡量呢？由于通常没有正确的输出 y，我们采取一些其他办法来度量其模型效果：

直观检测，这是一种非量化的方法。例如对文本的主题进行聚类，我们可以在直观上判断属于同一个类的文本是否具有某个共同的主题，这样的分类是否有明显的语义上的共同点。

基于任务的评价。如果聚类得到的模型被用于某个特定的任务，我们可以维持该任务中其他的设定不变，使用不同的聚类模型，通过某种指标度量该任务的最终结果来间接判断聚类模型的优劣；

人工标注测试集。有时候采用非监督学习的原因是人工标注成本过高，导致标注数据缺乏，只能使用无标注数据来训练。在这种情况下，可以人工标注少量的数据作为测试集，用于建立量化的评价指标。

5. 梯度下降算法

梯度下降算法作为机器学习中模型参数更新的算法，从形式上看十分简单，但又十分好用，广受好评，其表达式为：

$$w_{t+1} = w_t - \eta \cdot \nabla L(w_t)$$

其中 w_t 表示损失函数在时刻 t 的参数，η 表示学习因子，即梯度下降的步长，$\nabla L(w_t) = \frac{dL(w_t)}{dw_t}$ 表示损失函数 $L(w_t)$ 在位置 w_t 的梯度。

梯度下降算法的目的在于，通过不断更新权重参数 w，从而使得损失函数 L 的值不断变小。

那么背后的数学推导是什么样子的呢？首先我们谈一谈什么是梯度。

什么是梯度？

函数在某点的梯度是这样一个向量，它的方向与方向导数取得最大值的方向一致，表示函数在该点处沿着该方向（此梯度的方向）变化最快，变化率最大（为梯度的模）。如果用山来比喻的话，就是沿着梯度方向的坡度最陡峭。沿着梯度方向，函数增加的最快；沿着负梯度方向，函数减少的最快。

为什么沿着负梯度方向时函数值下降最快呢？

此时需要引入泰勒展开式，具体来说是一阶泰勒展开式。

$$f(x)=\frac{f(x_0)}{0!}+\frac{f'(x_0)}{1!}(x-x_0)+\frac{f''(x_0)}{2!}(x-x_0)^2+\cdots+\frac{f^{(n)}(x_0)}{n!}(x-x_0)^n+R_n(x)$$

其中 $f^{(n)}(x_0)$ 表示 $f(x)$ 的 n 阶导数，等号右边的多项式称为函数 $f(x)$ 在 x_0 的泰勒展开式，$R_n(x)$ 是泰勒展开式的余项，是 $(x-x_0)$ 的高阶无穷小（趋向于0）。

取一阶泰勒展开式

$$f(x)=\frac{f(x_0)}{0!}+\frac{f'(x_0)}{1!}(x-x_0)+R_1(x)$$

$R_1(x)$ 是一个无穷小量，因此得到一阶泰勒展开式的近似表达

$$f(x)\approx\frac{f(x_0)}{0!}+\frac{f'(x_0)}{1!}(x-x_0)$$

梯度下降算法推导

根据一阶泰勒展开式，可以将损失函数展开为

$$L(w_{t+1})\approx L(w_t)+(w_{t+1}-w_t)\cdot\nabla L(w_t)$$

其中 $(w_{t+1}-w_t)$ 表示前进的步长，引入单位向量 $\vec{v}$ 表示方向，$(w_{t+1}$、$w_t)$ 和 $\vec{v}$ 之间的关系如图1所示，即

$$w_{t+1}-w_t=\eta\vec{v}$$

进而有

$$L(w_{t+1})\approx L(w_t)+\eta\vec{v}\cdot\nabla L(w_t)$$

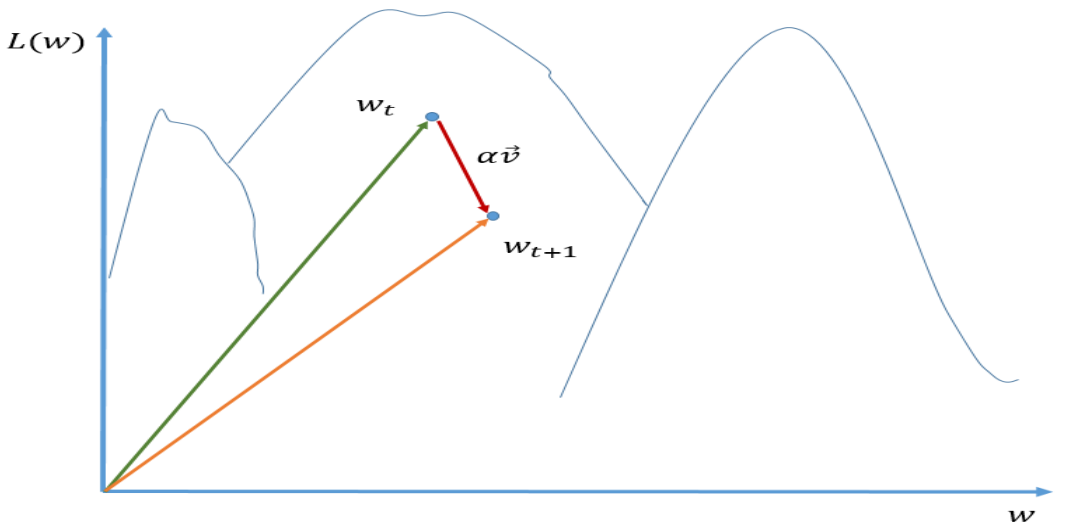

图 1: 梯度下降的示意图

局部下降的目的是希望每次 w_t 更新，都能让损失函数值 $L(w_{t+1})$ 变小。我们的目标希望每移动一步，损失函数的值都能减小到最大程度，即 $\eta\vec{v}\cdot\nabla L(w_t)$ 的绝对值取得最大值。那么如何能做点这一点呢？

首先我们看损失函数值变小，即有 $L(w_{t+1}) < L(w_t)$，则

$$L(w_{t+1}) - L(w_t) \approx \eta\vec{v}\cdot\nabla L(w_t) < 0$$

η 为标量，且一般设定为正值，所以可以忽略，不等式变成了：

$$\vec{v}\cdot\nabla L(w_t) < 0$$

因为

$$\vec{v}\cdot\nabla L(w_t) = \|\vec{v}\|\cdot\|\nabla L(w_t)\|\cdot cos(\alpha)$$

当夹角大于 90 度而小于 270 度时，$cos(\alpha) < 0$，上述不等式成立。那么什么时候损失函数的值都能减小到最大程度呢？只有当 $\alpha = -\pi$，$cos(\alpha) = -1$ 时，即 $\|\eta\vec{v}\cdot\nabla L(w_t)\| = \|\vec{v}\|\cdot\|\nabla L(w_t)\|$ 取得最大值（向量的乘积最小），$L(w_{t+1}) \approx$

$L(w_t)-\|\vec{v}\|\cdot\|\nabla L(w_t)\|$。因此，此时损失函数的值下降的最快，且向量 $\vec{v}$ 与 $\nabla L(w_t)$ 方向完全相反。

$\vec{v}$ 是单位向量向量，所以

$$\vec{v}=-\frac{\nabla L(w_t)}{\|\nabla L(w_t)\|}$$

$\frac{\nabla L(w_t)}{\|\nabla L(w_t)\|}$ 是 $\nabla L(w_t)$ 的单位向量，$\|\nabla L(w_t)\|$ 是 $\nabla L(w_t)$ 的模。因此

$$w_{t+1}-w_t=\eta\vec{v}=-\eta\frac{\nabla L(w_t)}{\|\nabla L(w_t)\|}=-\frac{\eta}{\|\nabla L(w_t)\|}\nabla L(w_t)$$

设 $\eta=\frac{\eta}{\|\nabla L(w_t)\|}$（合并参数 η 和梯度的模），得到

$$w_{t+1}=w_t-\eta\nabla L(w_t)$$

三、教学目标考核（20 分钟）

讨论：

1. 在机器学习中如何做模型评估
2. 在机器学习如何调整模型参数
3. 什么是监督学习？什么是非监督学习？

四、总结（5 分钟）

本部分对机器学习基础知识进行了介绍，这部分是理解后续高级操作的基础，需要读者认真消化。监督学习与非监督学习主要针对数据集定义。有监督数据集需要人工标注，成本较为昂贵，但是在训练模型时往往能够保障效果。无监督数据集一般不需要过多人工操作，可以通过爬虫等方式自动大量获得。由于没有监督信息的约束，需要设计巧妙的学习算法才能有效利用无监督数据集训练模型，不过大量廉价数据可以从另一个方面提高模型性能。模型评估需要根据模型的训练历史判断模型是否处于欠拟合或过拟合状态。尽管有一定的规律作为指导，而且有一些工具可以辅助分析，但是模型的评估过程一般需要较为丰富的经验。可以在深度学习实验中有意识地训练自己的模型评估能力。