C.既要保证同一类别样本尽可能靠近，也要保证不同类别样本尽可能彼此远离

D.只要保证同一类别样本尽可能靠近

8.LDA(线性区别分析)与PCA（主成分分析）均是降维的方法，下面描述不正确的是（ C ）

A.LDA降维后所得到维度是与数据样本的类别个数K有关（与数据本身维度无关）

B.PCA对高维数据降维后的维数是与原始数据特征维度相关（与数据类别标签无关）

C.PCA和LDA均是基于监督学习的降维方法

D.假设原始数据一共有K个类别，那么LDA所得数据的降维维度小于或等于K−1

9.逻辑斯蒂回归和线性区别分析均可完成分类任务，下面描述正确的是（ B ）

A.逻辑斯蒂回归在降维所得空间中进行分类，线性区别分析在数据原始空间进行分类

B.逻辑斯蒂回归可直接在数据原始空间进行分类，线性区别分析需要在降维所得空间中进行分类

C.逻辑斯蒂回归和线性区别分析都是直接在数据原始空间进行分类

D.以上都不正确

10.下面对逻辑斯蒂回归（logistic regression)和多项逻辑斯蒂回归模型(multi-nominal logistic model)描述不正确的是（ D ）

A.两者都是监督学习的方法

B.两者都可被用来完成多类分类任务

C.多项逻辑斯蒂回归模型也被称为softmax函数

D.逻辑斯帝回归是监督学习，多项逻辑斯蒂回归模型是非监督学习

第五章 无监督学习

一、K均值聚类

1.K均值聚类基本概念与目标

- 若干定义：
  - n个m-维数据$\{x_1, x_2, \dots, x_n\}$，$x_i \in R^m (1 \le i \le n)$
  - 两个m维数据之间的欧氏距离为

$$d(x_i, x_j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{im} - x_{jm})^2}$$

$d(x_i, x_j)$值越小，表示x_i和x_j越相似；反之越不相似

  - 聚类集合数目k
- 问题：如何将n个数据依据其相似度大小将它们分别聚类到k个集合，使得每个数据仅属于一个聚类集合。

2.K均值聚类算法

（1）【填空题】K均值聚类算法的步骤是：初始化聚类质心->将每一个数据放入唯一一个聚类集合中->根据聚类结果，更新每个聚类集合的质心->循环迭代，直至满足条件。

（2）初始化聚类质心

第一步：初始化聚类质心

- 初始化k个聚类质心$c = \{c_1, c_2, \dots, c_k\}$，$c_j \in R^m (1 \le j \le k)$
- 每个聚类质心c_j所在集合记为G_j

（3）将每个待聚类数据放入唯一一个聚类集合中

第二步：将每个待聚类数据放入唯一一个聚类集合中

- 计算待聚类数据x_i和质心c_j之间的欧氏距离$d(x_i, c_j)$ $(1 \leq i \leq n, 1 \leq j \leq k)$
- 将每个x_i放入与之距离最近聚类质心所在聚类集合中，

  即 $\underset{c_j \in C}{\operatorname{argmin}}\, d(x_i, c_j)$

（4）根据聚类结果，更新聚类质心

第三步：根据聚类结果、更新聚类质心

- 根据每个聚类集合中所包含的数据，更新该聚类集合质心值，即：$c_j = \frac{1}{|G_j|}\sum_{x_i \in G_j} x_i$

（5）算法循环迭代，直至满足条件

第四步：算法循环迭代，直到满足条件

- 在新聚类质心基础上，根据欧氏距离大小，将每个待聚类数据放入唯一一个聚类集合中
- 根据新的聚类结果、更新聚类质心

聚类迭代满足如下任意一个条件，则聚类停止：

- 已经达到了迭代次数上限
- 前后两次迭代中，聚类质心基本保持不变

3.K均值聚类——最小化类内方差

K均值聚类算法的另一个视角：最小化每个类簇的方差

■ 方差：用来计算变量（观察值）与样本平均值之间的差异

$$\arg\min_{G} \sum_{i=1}^{k} \sum_{x \in G_i} ||x - G_i||^2 = \arg\min_{G} \sum_{i=1}^{k} |G_i| \, Var \, G_i$$

第i个类簇的方差：$\mathrm{var}(G_i) = \frac{1}{|G_i|} \sum_{x \in G_i} ||x - G_i||^2$

- 欧氏距离与方差量纲相同
- 最小化每个类簇方差将使得最终聚类结果中每个聚类集合中所包含数据呈现出来差异性最小。

4.算法的不足之处：

●需要事先确定聚类数目，很多时候我们并不知道数据应被聚类的数目

●需要初始化聚类质心，初始化聚类中心对聚类结果有较大的影响（初始化不当，有可能收敛于局部极值而非整体最值）

●算法是迭代执行，时间开销非常大

●欧氏距离假设数据每个维度之间的重要性是一样的

5.应用：图像分类

二、主成分分析

1.基本思想：对特征进行降维。

2.数据样本的方差、协方差和相关系数计算

数据样本的方差
variance

假设有n个数据，记为$X = \{x_i\}$ $(i = 1, \dots, n)$

- 方差等于各个数据与样本均值之差的平方和之平均数
- 方差描述了样本数据的波动程度

$$\mathrm{Var(X)} = \frac{1}{n}\sum_{i=1}^{n}(x_i - u)^2$$

其中u是样本均值， $u = \frac{1}{n}\sum_{i=1}^{n} x_i$

数据样本的协方差
covariance

假设有n个两维变量数据，记为$(X, Y) = \{(x_i, y_i)\}$ $(i = 1, \dots, n)$

- 衡量两个变量之间的相关度

$$\mathrm{cov(X, Y)} = \frac{1}{n}\sum_{i=1}^{n}(x_i - E(X))(y_i - E(Y))$$

其中$E(X)$和$E(Y)$分别是X和Y的样本均值, 分别定义如下

$$\mathrm{E(X)} = \frac{1}{n}\sum_{i=1}^{n} x_i, \qquad \mathrm{E(Y)} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

表5.1 方差与协方差的计算例子（无偏估计）

编号	x_i	y_i	$x_i - E(X)$	$y_i - E(Y)$	$[x_i - E(X)][y_i - E(Y)]$
1	1	7	−8.33	−16.67	138.89
2	3	11	−6.33	−12.67	80.22
3	6	17	−3.33	−6.67	22.22
4	10	25	0.67	1.33	0.89
5	15	35	5.67	11.33	64.22
6	21	47	11.67	23.33	272.22
	$E(X) = 9.33$	$E(Y) = 23.67$	$\mathrm{var}(X) = 57.87$	$\mathrm{var}(Y) = 231.47$	$\mathrm{cov}(X, Y) = 115.73$

可通过皮尔逊相关系数（Pearson Correlation coefficient ）将两组变量之间的关联度规整到一定的取值范围内。皮尔逊相关系数定义如下：

$$\mathrm{corr}(X,Y)=\frac{Cov(X,Y)}{\sqrt{Var(X)Var(Y)}}=\frac{Cov(X,Y)}{\sigma_x\sigma_y}$$

表5.1 方差与协方差的计算例子（无偏估计） $\mathrm{corr}(X,Y)=1$

表5.1 方差与协方差的计算例子（无偏估计）

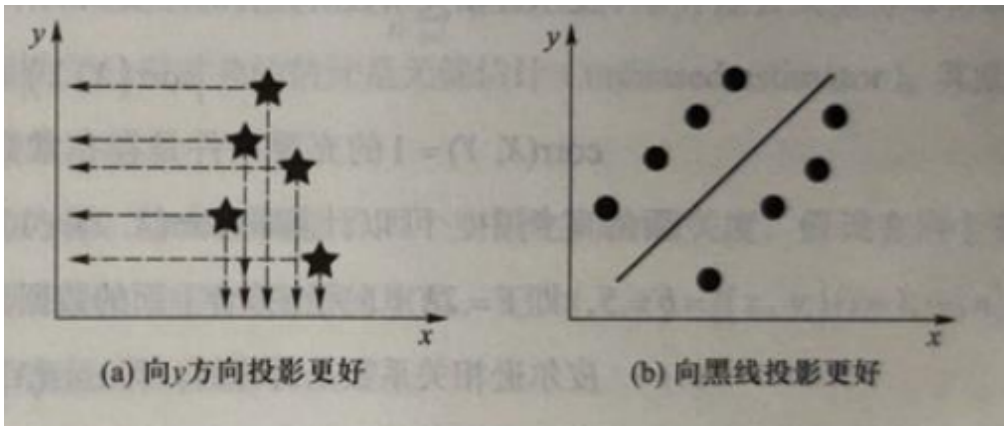
编号	x_i	y_i	$x_i-E(X)$	$y_i-E(Y)$	$[x_i-E(X)][y_i-E(Y)]$
1	1	7	−8.33	−16.67	138.89
2	3	11	−6.33	−12.67	80.22
3	6	17	−3.33	−6.67	22.22
4	10	25	0.67	1.33	0.89
5	15	35	5.67	11.33	64.22
6	21	47	11.67	23.33	272.22
	$E(X)=9.33$	$E(Y)=23.67$	$\mathrm{var}(X)=57.87$	$\mathrm{var}(Y)=231.47$	$\mathrm{cov}(X,Y)=115.73$

3.算法思想的深入理解

①在数理统计中，方差被经常用来度量数据和其数学期望（即均值）之间偏离程度，这个偏离程度反映了数据分布结构。

②在许多实际问题中，研究数据和其均值之间的偏离程度有着很重要的意义。

③在降维之中，需要尽可能将数据向方差最大方向进行投影，使得数据所蕴含信息没有丢失，彰显个性。如左下图所示，向y方向投影（使得二维数据映射为一维）就比向x方向投影结果在降维这个意义上而言要好；右下图则是向黑斜线方向投影要好。



(a) 向y方向投影更好　　(b) 向黑线投影更好

④主成分分析思想是将n维特征数据映射到l维空间（$n \gg l$），去除原始数据之间的冗余性（通过去除相关性手段达到这一目的）。

⑤将原始数据向这些数据方差最大的方向进行投影。一旦发现了方差最大的投影方向，则继续寻找保持方差第二的方向且进行投影。

⑥将每个数据从n维高维空间映射到l维低维空间，每个数据所得到最好的k维特征就是使得每一维上样本方差都尽可能大。

4.算法过程

①参数求解目标：

- 假设有n个d维样本数据所构成的集合$D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n\}$，其中$\boldsymbol{x}_i (1 \le i \le n) \in R^d$。
- 集合D可以表示成一个$n \times d$ 的矩阵$\mathbf{X}$。
- 假定每一维度的特征均值均为零（已经标准化）。
- 主成分分析的目的是求取一个且使用一个$d \times l$的映射矩阵$\mathbf{W}$。
- 给定一个样本$\boldsymbol{x}_i$，可将$\boldsymbol{x}_i$ 从d维空间如下映射到 l 维空间：$(\boldsymbol{x}_i)_{1\times d}(\mathbf{W})_{d\times l}$
- 将所有降维后数据用$\mathbf{Y}$ 表示，有 $\mathbf{Y} = \mathbf{X}\,\mathbf{W}$

降维结果　原始数据　映射矩阵

? 如何求取映射矩阵W

- $\mathbf{Y} = n \times l$
- $\mathbf{X} = n \times d$
- $\mathbf{W} = d \times l$

②在约束条件下，目的是使得转化后矩阵的迹最大。

降维后n个l维样本数据Y的方差为：

$$\mathrm{var}(\mathrm{Y}) = \frac{1}{n-1}\mathrm{trace}(\mathrm{Y}^{\mathrm{T}}\mathrm{Y})$$

$$= \frac{1}{n-1}\mathrm{trace}(\mathrm{W}^{\mathrm{T}}\mathrm{X}^{\mathrm{T}}\mathrm{X}\mathrm{W})$$

$$= trace(\mathrm{W}^{\mathrm{T}}\frac{1}{n-1}\mathrm{X}^{\mathrm{T}}\mathrm{X}\mathrm{W})$$

降维前n个d维样本数据X的协方差矩阵记为：

$$\Sigma = \frac{1}{n-1}\mathrm{X}^{\mathrm{T}}X$$

主成份分析的求解目标函数为

$$\max_{\mathrm{W}} trace(\mathrm{W}^T\Sigma\mathrm{W})$$

满足约束条件

$$\mathrm{w}_i^T\mathrm{w}_i = 1 \quad i \in \{1, 2, \dots, l\}$$

③根据约束条件，可以构造拉格朗日函数，并对各个系数wi求偏导数。令它们等于0，恰好可以得到特征方程。这说明，拉格朗日函数中的各个乘子系数，恰好等于原始协方差矩阵中的特征值。

所有带约束的最优化问题，可通过拉格朗日乘子法将其转化为无约束最优化问题

主成份分析求解目标函数为

$$\max_{\mathbf{W}} \ trace(\mathbf{W}^T \Sigma \mathbf{W})$$

满足约束条件

$$\mathbf{w}_i^T \mathbf{w}_i = 1 \quad i \in \{1, 2, \dots, l\}$$

保证降维后结果正交以去除相关性（即冗余度）

拉格朗日函数

$$\mathbf{Y} = n \times l \quad \mathbf{X} = n \times d \quad \mathbf{W} = d \times l$$

$$L(\mathbf{W}, \boldsymbol{\lambda}) = trace(\mathbf{W}^T \boldsymbol{\Sigma} \mathbf{W}) - \sum_{i=1}^{l} \lambda_i (\mathbf{w}_i^T \mathbf{w}_i - 1)$$

其中$\lambda_i (1 \le i \le l)$为拉格朗日乘子，$\mathbf{w}_i$为矩阵W第$i$列。

对上述拉格朗日函数中变量$\mathbf{w}_i$求偏导并令导数为零，得到

$$\Sigma \mathbf{w}_i = \lambda_i \mathbf{w}_i$$

上式表明：每一个$\mathbf{w}_i$都是n个d维样本数据X的协方差矩阵Σ的一个特征向量，λ_i是这个特征向量所对应的特征值。

结论：可以通过求解原始协方差矩阵的特征值->取前K大的特征值->选择这些特征值对应的特征向量->组成投影矩阵。

$$\Sigma \mathbf{w}_i = \lambda_i \mathbf{w}_i，且 trace(\mathbf{W}^T \boldsymbol{\Sigma} \mathbf{W}) = \sum_{i=1}^{l} \mathbf{w}_i^T \boldsymbol{\Sigma} \mathbf{w}_i = \sum_{i=1}^{l} \lambda_i$$

- 可见，在主成份分析中，最优化的方差等于原始样本数据X的协方差矩阵Σ的特征根之和。
- 要使方差最大，我们可以求得协方差矩阵Σ的特征向量和特征根，然后取前l个最大特征根所对应的特征向量组成映射矩阵W即可。
- 注意，每个特征向量$\mathbf{w}_i$与原始数据x_i的维数是一样的，均为d。

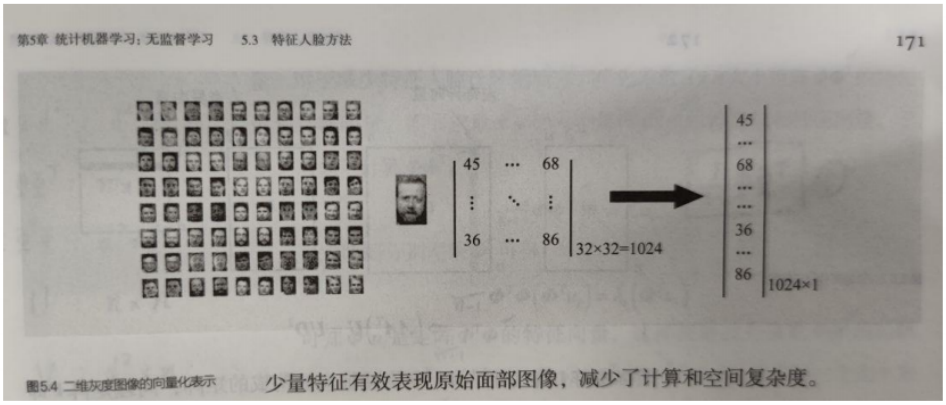
5.其他常见的降维方法：非负矩阵分解、多维尺度法、局部线性嵌入。

三、特征人脸方法

1.基本思想：**特征人脸方法是一种应用主成份分析来实现人脸图像降维的方法，其本质是用一种称为“特征人脸(eigenface)”的特征向量按照线性组合形式来**

表达每一张原始人脸图像，进而实现人脸识别。

2.人脸图像转化为列向量



- **将每幅人脸图像转换成列向量**
- **如将一幅32×32的人脸图像转成1024×1的列向量**

3.基本步骤

①对各个样本，分别进行中心化处理。

②计算原始样本数据的协方差矩阵。

③对协方差矩阵，计算特征根，取前l个最大特征值，组成映射矩阵。

④通过映射矩阵，把d维列向量（原始人脸样本矩阵转化而来的列向量，元素数目不变，只是进行了矩阵变形）压缩为l维列向量。

⑤【理解】本质上，所得l维列向量，每一维分别对应的是一个特征人脸的相应权重，那么这个l维列向量的意义是各个特征人脸的权重，就实现了利用特征人脸，进行线性组合（加权叠加），以表达原始图像的目的。

$$\mathbf{Y} = n \times l \quad \mathbf{X} = n \times d \quad \mathbf{W} = d \times l$$

- **输入：** n**个1024维人脸样本数据所构成的矩阵X，降维后的维数**l
- **输出：** **映射矩阵**$\mathbf{W} = \{w_1, w_2, \dots, w_l\}$ **(其中每个**$w_j(1 \le \mathrm{j} \le l)$**是一个特征人脸)**
- **算法步骤：**

1：对于每个人脸样本数据x_i**进行中心化处理：** $x_i = x_i - \mu,\ \mu = \frac{1}{n}\sum_{j=1}^{n} x_j$

2: 计算原始人脸样本数据的协方差矩阵： $\Sigma = \frac{1}{n-1}\mathrm{X}^{\mathrm{T}}X$

3: 对协方差矩阵Σ**进行特征值分解，对所得特征根从到小排序**$\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_d$

4: 取前l**个最大特征根所对应特征向量**$w_1, w_2, \dots, w_l$**组成映射矩阵W**

5：将每个人脸图像x_i**按照如下方法降维：** $(x_i)_{1\times d}(\mathrm{W})_{d\times l} = 1 \times l$

4.降维后，人脸匹配不再基于原始信息，而是比较两个人脸经过特征人脸方法降维处理后，对应的l维向量系数的相似程度。

5.【填空题】常见的人脸表达方法有：聚类、主成分分析、非负矩阵分解。

四、潜在语义分析

1.目的与基本思想

①潜在语义分析是一种从海量文本数据中学习单词-单词、单词-文档以及文档-文档之间隐性关系，进而得到文档和单词表达特征的方法。

②该方法的基本思想是综合考虑某些单词在哪些文档中同时出现，以此来决定该词语的含义与其他的词语的相似度。

2.算法流程：建立单词-文档矩阵->奇异值分解->重建矩阵->挖掘语义关系

（1）建立单词-文档矩阵

表5.2 单词-文档（term-document）矩阵

176

表5.2 单词－文档（term-document）矩阵

	a1	a2	a3	a4	a5	b1	b2	b3	b4
nonconvex	1	0	0	1	0	0	0	0	0
regression	1	0	1	0	0	0	0	0	0
optimization	1	1	0	0	1	0	0	0	0
network	0	1	0	0	0	0	0	1	0
analysis	0	0	1	1	0	0	0	0	0
minimization	0	0	1	0	1	0	0	0	0
gene	0	0	0	0	0	1	1	1	0
syndrome	0	0	0	0	0	0	1	0	1
editing	0	0	0	0	0	0	1	0	1
human	0	0	0	0	0	0	0	1	1

（2）奇异值分解

第5章 统计机器学习：无监督学习　5.4 潜在语义分析　177

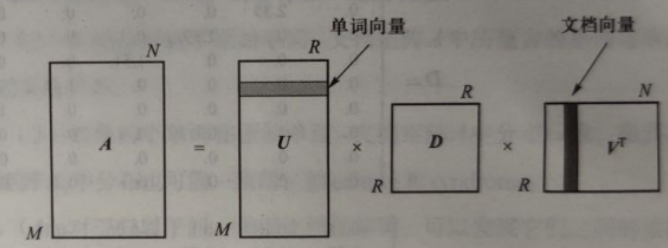


图5.8 单词－文档矩阵的奇异值分解示意

$$U = \begin{Bmatrix} -0.09 & -0.38 & -0.14 & -0.04 & 0.71 & -0.02 & 0.12 & 0.33 & 0.45 & -0. \\ -0.1 & -0.47 & -0.2 & -0.06 & 0. & -0.06 & -0.76 & -0.38 & -0.01 & 0. \\ -0.18 & -0.53 & 0.44 & 0.51 & 0. & -0.14 & 0.16 & 0.09 & -0.42 & -0. \\ -0.28 & -0.08 & 0.57 & -0.23 & -0. & 0.32 & 0.17 & -0.5 & 0.38 & 0. \\ -0.07 & -0.35 & -0.4 & -0.46 & -0. & 0.22 & 0.46 & -0.19 & -0.45 & -0. \\ -0.09 & -0.38 & -0.14 & -0.04 & -0.71 & -0.02 & 0.12 & 0.33 & 0.45 & -0. \\ -0.52 & 0.13 & 0.16 & -0.47 & 0. & -0.65 & -0.04 & 0.16 & -0.11 & -0. \\ -0.45 & 0.16 & -0.32 & 0.34 & -0. & 0.01 & 0.11 & -0.19 & 0.08 & 0.71 \\ -0.45 & 0.16 & -0.32 & 0.34 & -0. & 0.01 & 0.11 & -0.19 & 0.08 & -0.71 \\ -0.42 & 0.1 & 0.09 & -0.11 & -0. & 0.63 & -0.32 & 0.49 & -0.22 & -0. \end{Bmatrix}$$

对表5.2中的单词-文档矩阵进行奇异值分解，得到的结果展示。

$$V^T = \begin{Bmatrix} -0.15 & -0.18 & -0.1 & -0.06 & -0.11 & -0.21 & -0.58 & -0.49 & -0.54 \\ -0.59 & -0.26 & -0.51 & -0.31 & -0.39 & 0.06 & 0.19 & 0.07 & 0.18 \\ 0.05 & 0.57 & -0.42 & -0.3 & 0.17 & 0.09 & -0.26 & 0.46 & -0.3 \\ 0.27 & 0.18 & -0.37 & -0.33 & 0.32 & -0.31 & 0.13 & -0.54 & 0.38 \\ 0.5 & 0. & -0.5 & 0.5 & -0.5 & 0. & 0. & 0. & -0. \\ -0.19 & 0.15 & 0.11 & 0.16 & -0.14 & -0.53 & -0.52 & 0.25 & 0.53 \\ -0.51 & 0.36 & -0.19 & 0.62 & 0.3 & -0.04 & 0.21 & -0.2 & -0.1 \\ 0.05 & -0.56 & -0.33 & 0.19 & 0.58 & 0.21 & -0.3 & 0.2 & 0.16 \\ 0.07 & -0.27 & -0.07 & 0. & 0.14 & -0.72 & 0.38 & 0.33 & -0.36 \end{Bmatrix} \quad D = \begin{Bmatrix} 2.46 & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 2.35 & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 1.79 & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 1.51 & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 1.41 & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 1.23 & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0.93 & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0.73 & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0.15 \end{Bmatrix}$$

（3）重建矩阵

当k=2，即选取最大的前两个特征根及其对应的特征向量对矩阵A进行重建。下面给出了选取矩阵U、矩阵D和矩阵V的子部分重建所得矩阵A_2：

$$A_2 = \left\{ \begin{array}{ccccccccc} & a1 & a2 & a3 & a4 & a5 & b1 & b2 & b3 & b4 \\ nonconvex & 0.56 & 0.27 & 0.49 & 0.3 & 0.37 & -0.01 & -0.05 & 0.05 & -0.05 \\ regression & 0.68 & 0.33 & 0.59 & 0.36 & 0.45 & -0.01 & -0.06 & 0.05 & -0.06 \\ optimization & 0.79 & 0.4 & 0.68 & 0.41 & 0.53 & 0.02 & 0.02 & 0.14 & 0.02 \\ network & 0.22 & 0.18 & 0.17 & 0.1 & 0.15 & 0.13 & 0.36 & 0.32 & 0.33 \\ analysis & 0.51 & 0.25 & 0.44 & 0.27 & 0.34 & -0.01 & -0.06 & 0.03 & -0.06 \\ minimization & 0.56 & 0.27 & 0.49 & 0.3 & 0.37 & -0.01 & -0.05 & 0.05 & -0.05 \\ gene & 0.01 & 0.16 & -0.03 & -0.02 & 0.02 & 0.29 & 0.81 & 0.66 & 0.75 \\ syndrome & -0.05 & 0.11 & -0.07 & -0.05 & -0.02 & 0.26 & 0.72 & 0.58 & 0.67 \\ editing & -0.05 & 0.11 & -0.07 & -0.05 & -0.02 & 0.26 & 0.72 & 0.58 & 0.67 \\ human & 0.01 & 0.13 & -0.02 & -0.01 & 0.02 & 0.23 & 0.65 & 0.53 & 0.6 \end{array} \right\}$$

（4）挖掘语义关系

第四步：挖掘语义关系

- 基于单词-文档矩阵A，可以计算任意两个文档之间的皮尔逊相关系数，从而得到如下文档-文档相关系数矩阵：

$$\left\{ \begin{array}{ccccccccc} & a1 & a2 & a3 & a4 & a5 & b1 & b2 & b3 & b4 \\ a1 & 1. & 0.22 & 0.05 & 0.22 & 0.22 & -0.22 & -0.43 & -0.43 & -0.43 \\ a2 & 0.22 & 1. & -0.33 & -0.25 & 0.37 & -0.17 & -0.33 & 0.22 & -0.33 \\ a3 & 0.05 & -0.33 & 1. & 0.22 & 0.22 & -0.22 & -0.43 & -0.43 & -0.43 \\ a4 & 0.22 & -0.25 & 0.22 & 1. & -0.25 & -0.17 & -0.33 & -0.33 & -0.33 \\ a5 & 0.22 & 0.37 & 0.22 & -0.25 & 1. & -0.17 & -0.33 & -0.33 & -0.33 \\ b1 & -0.22 & -0.17 & -0.22 & -0.17 & -0.17 & 1. & 0.51 & 0.51 & -0.22 \\ b2 & -0.43 & -0.33 & -0.43 & -0.33 & -0.33 & 0.51 & 1. & 0.05 & 0.52 \\ b3 & -0.43 & 0.22 & -0.43 & -0.33 & -0.33 & 0.51 & 0.05 & 1. & 0.05 \\ b4 & -0.43 & -0.33 & -0.43 & -0.33 & -0.33 & -0.22 & 0.52 & 0.05 & 1. \end{array} \right\}$$

- 基于重建单词-文档矩阵A_2，可以计算任意两个文档之间的皮尔逊相关系数，从而得到如下文档-文档相关系数矩阵：

$$\left\{ \begin{array}{ccccccccc} & a1 & a2 & a3 & a4 & a5 & b1 & b2 & b3 & b4 \\ a1 & 1. & 0.97 & 1. & 1. & 1. & -0.95 & -0.95 & -0.93 & -0.95 \\ a2 & 0.97 & 1. & 0.97 & 0.97 & 0.98 & -0.85 & -0.86 & -0.83 & -0.86 \\ a3 & 1. & 0.97 & 1. & 1. & 1. & -0.95 & -0.96 & -0.94 & -0.96 \\ a4 & 1. & 0.97 & 1. & 1. & 1. & -0.95 & -0.96 & -0.94 & -0.96 \\ a5 & 1. & 0.98 & 1. & 1. & 1. & -0.94 & -0.95 & -0.93 & -0.95 \\ b1 & -0.95 & -0.85 & -0.95 & -0.95 & -0.94 & 1. & 1. & 1. & 1. \\ b2 & -0.95 & -0.86 & -0.96 & -0.96 & -0.95 & 1. & 1. & 1. & 1. \\ b3 & -0.93 & -0.83 & -0.94 & -0.94 & -0.93 & 1. & 1. & 1. & 1. \\ b4 & -0.95 & -0.86 & -0.96 & -0.96 & -0.95 & 1. & 1. & 1. & 1. \end{array} \right\}$$

3.结论：

①单词-单词之间的相关关系更加明确，在同一文档中出现过的单词之间相关系数趋近为1，没有同时出现在同一文档中的单词之间相关系数趋近

为-1。

②隐性语义分析对原始单词-文档矩阵中所蕴含关系进行了有效挖掘，

能刻画单词-单词、单词-文档以及文档-文档之间语义关系。

五、期望最大化算法

1.最大似然估计与最大后验估计

- **最大似然估计** （maximum likelihood estimation，MLE）

假设由n个数据样本构成的集合$\mathcal{D}=\{x_1,x_2,\dots,x_n\}$从参数为$\Theta$的某个模型（如高斯模型等）以一定概率独立采样得到。于是，可以通过最大似然估计算法（maximum likelihood estimation，MLE）来求取参数Θ，使得在参数为Θ的模型下数据集$\mathcal{D}$出现的可能性最大，即$\hat{\Theta}=\underset{\Theta}{\operatorname{argmax}}\,P(\mathcal{D}|\Theta)$。

- **最大后验估计** （maximum a posteriori estimation，MAP）

或者也可利用最大后验估计（maximum a posteriori estimation，MAP）从数据集$\mathcal{D}$来如下估计参数Θ：$\hat{\Theta}=\underset{\Theta}{\operatorname{argmax}}\,P(\Theta|\mathcal{D})=\underset{\Theta}{\operatorname{argmax}}\frac{P(\mathcal{D}|\Theta)P(\Theta)}{P(\mathcal{D})}$。由于$P(\mathcal{D})$与$\Theta$无关，则可得$\underset{\Theta}{\operatorname{argmax}}\frac{P(\mathcal{D}|\Theta)P(\Theta)}{P(\mathcal{D})}=\underset{\Theta}{\operatorname{argmax}}\,P(\mathcal{D}|\Theta)P(\Theta)$，对这个式子取对数，得到$\underset{\Theta}{\operatorname{argmax}}\,logP(\mathcal{D}|\Theta)+logP(\Theta)$。可见，最大后验估计与最大似然估计相比，增加了一项与Θ相关的先验概率$P(\Theta)$。

2.①无论是最大似然估计算法或者是最大后验估计算法，都是充分利用已有

数据，在参数模型确定（只是参数值未知）情况下，对所优化目标中的

参数求导，令导数为0，求取模型的参数值。

②在解决一些具体问题时，难以事先就将模型确定下来，然后利用数据来

求取模型中的参数值。在这样情况下，无法直接利用最大似然估计算法

或者最大后验估计算法来求取模型参数。

3.期望最大化算法

（1）EM算法是一种重要的用于解决含有隐变量（latent variable）问题的参数估计方法。

（2）EM算法分为求取期望（E步骤，expectation）和期望最大化（M步骤，

maximization）两个步骤。

（3）在EM算法的**E步骤**时，先假设模型参数的初始值，估计隐变量取值；在EM

算法的**M步骤**时，基于观测数据、模型参数和隐变量取值一起来最大化“拟

合”数据，更新模型参数。基于所更新的模型参数，得到新的隐变量取值

（EM算法的E步），然后继续极大化“拟合”数据，更新模型参数（EM算

法的M步)。以此类推迭代，直到算法收敛，得到合适的模型参数。

4.期望最大化算法举例——投掷2个硬币

【理解】

（1）初始化，设A硬币有0.60的概率面朝上方，B硬币有0.50的概率面朝上方。隐变量是对硬币的选择。

（2）运用贝叶斯公式，给定A、B硬币面朝上方的假设概率，以及硬币投掷结果，分别计算选择的硬币是A、选择的硬币是B的概率。

初始化每一轮中硬币A和硬币B投掷为正面的概率为 $\hat{\theta}_A^{(0)} = 0.60$ 和 $\hat{\theta}_B^{(0)} = 0.50$ 。
基于“HTTTHHTHTH”这10次投掷结果，由硬币A投掷所得概率为：

$$
\begin{aligned}
&P(\text{选择硬币A投掷}|\text{硬币投掷结果}, \theta) \\
&= \frac{P(\text{选择硬币A投掷, 硬币投掷结果}|\theta)}{P(\text{选择硬币A投掷, 硬币投掷结果}|\theta) + P(\text{选择硬币B投掷, 硬币投掷结果}|\theta)} \\
&= \frac{(0.6)^5 \times (0.4)^5}{(0.6)^5 \times (0.4)^5 + (0.5)^{10}} = 0.45
\end{aligned}
$$

这10次结果由硬币B投掷所得概率为：

$$
\begin{aligned}
&P(\text{选择硬币B投掷}|\text{硬币投掷结果}, \theta) \\
&= 1 - P(\text{选择硬币A投掷}|\text{硬币投掷结果}, \theta) \\
&= 0.55
\end{aligned}
$$

（3）已经算出了选择的是A硬币、选择的是B硬币的概率，那么用选择某个硬币的概率×投掷总次数×某个硬币正面朝上的概率，得到它正面朝上的次数的期望。负面朝上的次数的期望，可以类似计算。

（4）对于同一组(A硬币朝上概率,B硬币朝上概率)，进行5轮模拟试验，每一轮模拟试验投掷10次硬币，对于每一轮，分别计算A、B硬币正、反面朝上的数学期望。

表5.4 两个硬币在每一轮中被选择概率以及投掷正面/反面的次数

轮次	选硬币A概率	选硬币B概率	硬币A为正面期望次数	硬币A为反面期望次数	硬币B为正面期望次数	硬币B为反面期望次数
1	0.45	0.55	2.25	2.25	2.75	2.75
2	0.80	0.20	7.24	0.80	1.76	0.20
3	0.73	0.27	5.87	1.47	2.13	0.53
4	0.35	0.65	1.41	2.11	2.59	3.89
5	0.65	0.35	4.53	1.94	2.47	1.07
合计			21.30	8.57	11.70	8.43

一旦得到了在第一轮试验中选择硬币A或硬币B投掷的概率，则可以根据这个概率来计算第一轮中硬币A和硬币B投掷正面的期望次数。比如，硬币A为正面期望次数为 $N_{\text{正面总次数}} \times P_{\text{选硬币A概率}} = 5 \times 0.45 = 2.25$，其他项计算类似。

（5）这5轮模拟试验完成后，称一次迭代完成。上述步骤都是期望计算步骤。一次迭代完成时，可以得到A、B硬币正、反面朝上的总期望

次数。

（6）根据各自的总期望次数，更新各自正面朝上的概率。

期望最大化（M步骤，Maximization）：

在上面的计算中，通过初始化硬币A和硬币B投掷得到正面概率$\hat{\theta}_A^{(0)}$和$\hat{\theta}_B^{(0)}$，得到每一轮中选择硬币A和选择硬币B概率这一“隐变量”，进而可计算得到每一轮中硬币A和硬币B投掷正面次数。

在这些信息基础上，可更新得到硬币A和硬币B投掷为正面的概率，从而得到新的模型参数：

$$\hat{\theta}_A^{(1)} = \frac{21.30}{21.30+8.57} = 0.713 \quad \hat{\theta}_B^{(1)} = \frac{11.70}{11.70+8.43} = 0.581$$

接下来，可在新的概率值基础上继续计算每一轮投掷中选择硬币A或硬币B的概率，进而计算得到五轮中硬币A和硬币B投掷正面的总次数，从而得到硬币A和硬币B投掷为正面的更新概率值$\hat{\theta}_A^{(2)}$和$\hat{\theta}_B^{(2)}$。上述算法不断迭代，直至算法收敛，最终得到硬币A和硬币B投掷为正面的概率$\Theta = \{\Theta_A, \Theta_B\}$。

比如，A硬币正面朝上的更新概率 = A硬币在本轮试验中，正面朝上的总期望/(A硬币在本轮试验中，正面朝上的总期望+A硬币在本轮试验中，反面朝上的总期望)。

（7）进行多轮迭代，直到参数基本不变，模型收敛。

表5.5 两个硬币投掷结果为正面的概率迭代计算结果

表5.5 两个硬币投掷结果为正面的概率迭代计算结果

迭代次数	硬币A为正面次数	硬币A为反面次数	硬币B为正面次数	硬币B为反面次数	硬币A投掷正面概率θ_A	硬币B投掷正面概率θ_B
1	21.30	8.57	11.70	8.43	$\hat{\theta}_A^{(1)} = 0.713$	$\hat{\theta}_B^{(1)} = 0.581$
2	19.21	6.56	13.79	10.44	$\hat{\theta}_A^{(2)} = 0.745$	$\hat{\theta}_B^{(2)} = 0.569$
3	19.41	5.86	13.59	11.14	$\hat{\theta}_A^{(3)} = 0.768$	$\hat{\theta}_B^{(3)} = 0.550$
4	19.75	5.47	13.25	11.53	$\hat{\theta}_A^{(4)} = 0.783$	$\hat{\theta}_B^{(4)} = 0.535$
5	19.98	5.28	13.02	11.72	$\hat{\theta}_A^{(5)} = 0.791$	$\hat{\theta}_B^{(5)} = 0.526$
6	20.09	5.19	12.91	11.81	$\hat{\theta}_A^{(6)} = 0.795$	$\hat{\theta}_B^{(6)} = 0.522$
7	20.14	5.16	12.86	11.84	$\hat{\theta}_A^{(7)} = 0.796$	$\hat{\theta}_B^{(7)} = 0.521$
8	20.16	5.15	12.84	11.85	$\hat{\theta}_A^{(8)} = 0.796$	$\hat{\theta}_B^{(8)} = 0.520$
9	20.17	5.15	12.83	11.85	$\hat{\theta}_A^{(9)} = 0.797$	$\hat{\theta}_B^{(9)} = 0.520$
10	20.18	5.15	12.82	11.85	$\hat{\theta}_A^{(10)} = 0.797$	$\hat{\theta}_B^{(10)} = 0.520$

（8）归纳总结

①隐变量：每一轮选择硬币A还是选择硬币B来完成10次投掷是一个隐变量，硬币A和硬币B投掷。

②模型参数：结果为正面的概率$\Theta = \{\Theta A, \Theta B\}$称为模型参数。

③计算隐变量（EM算法的E步）、最大化似然函数和更新模型参数（EM算

法的M步)：EM算法使用迭代方法来求解模型参数$\Theta = \{\Theta A, \Theta B\}$：先初始化模型参数，然后计算得到隐变量（EM算法的E步），接着基于观测投掷结果和当前隐变量值一起来最大化似然函数（即使得模型参数能够更好拟合观测结果），更新模型参数（EM算法的M步)。基于当前得到的模型参数，继续更新隐变量（EM算法的E步），然后继续最大化似然函数，更新模型参数（EM算法的M步)。以此类推，不断迭代下去，直到模型参数基本无变化，算法收敛。

【理解】事实上，上述算法中，A、B硬币的出现次数期望，就是隐变量的计算结果。

5.期望最大化算法——投掷3个硬币

（1）背景：

- 假设有三枚质地材料不均匀的硬币（即每枚硬币投掷出现正反两面的概率不一定相等），这三枚硬币分别被标记为0，1，2。约定出现正面记为H（Head，头），出现反面记为T（Tail，尾），一次试验的过程如下：首先掷硬币0，如果硬币0投掷结果为H，则选择硬币1投掷三次，如果硬币0投掷结果为T，则选择硬币2投掷三次。观测结果中仅记录硬币1和硬币2的投掷结果，不出现硬币0的投掷结果。**因为硬币0的投掷结果没有被记录，所以是未观测到的数据（隐变量）。**

- 未观测数据取值集合记为 $C_Z=\{H,T\}$，观测数据取值集合记为 $C_X=\{HHH,TTT,HTT,THH,HHT,TTH,HTH,THT\}$，模型参数集合（三枚硬币分别出现正面的概率）为$\Theta=\{\lambda,p_1,p_2\}$。在n次试验中，未观测到的数据序列记为z，观测到的数据序列记为x，观测到的数据序列中有h次为正面朝上，t次为反面朝上。

$$P(x,z|\Theta)=P(z|\Theta)P(x|z,\Theta)$$

$$P(z|\Theta)=\begin{cases}\lambda & if\ z=H\\ 1-\lambda & if\ z=T\end{cases}\qquad P(x|z,\Theta)=\begin{cases}p_1^h(1-p_1)^t & if\ z=H\\ p_2^h(1-p_2)^t & if\ z=T\end{cases}$$

①假设有三枚质地材料不均匀的硬币（即每枚硬币投掷出现正反两面的概率

不一定相等），这三枚硬币分别被标记为0，1，2。约定出现正面记为H

（Head，头），出现反面记为T（Tail，尾），一次试验的过程如下：首先

掷硬币0，如果硬币0投掷结果为H，则选择硬币1投掷三次，如果硬币0投

掷结果为T，则选择硬币2投掷三次。观测结果中仅记录硬币1和硬币2的投

掷结果，不出现硬币0的投掷结果。**因为硬币0的投掷结果没有被记录，所**

以是未观测到的数据（隐变量）。

②未观测数据取值集合记为$CZ=\{H,T\}$，观测数据取值集合记为$CX=$

$\{HHH, TTT, HTT, THH, HHT, TTH, HTH, THT\}$，模型参数集合（三枚硬币

分别出现正面的概率）为Θ={λ,p1,p2}。在n次试验中，未观测到的数据序列

记为z，观测到的数据序列记为x，观测到的数据序列中有h次为正面朝上，

t次为反面朝上。

（2）参数估计与更新过程

- 在硬币0掷出正面后，选择硬币1投掷三次所得“反正反”这一结果的概率如下计算：

$$P(x = THT, z = H|\Theta) = \lambda p_1(1-p_1)^2$$

- 在硬币0掷出反面后，选择硬币2投掷三次所得“反正反”这一结果的概率如下计算：

$$P(x = THT, z = T|\Theta) = (1-\lambda)p_2(1-p_2)^2$$

如果某次观测得到“反正反”这一投掷结果，则该投掷结果发生的概率如下计算：

$$\begin{aligned} P(x = THT|\Theta) &= P(x = THT, z = H|\Theta) + P(x = THT, z = T|\Theta) \\ &= \lambda p_1(1-p_1)^2 + (1-\lambda)p_2(1-p_2)^2 \end{aligned}$$

如果某次观测得到“反正反”这一投掷结果，这一结果是由硬币0投掷为正面（未观测的数据）所促发的概率计算如下：

$$\begin{aligned} P(z = H|x = THT, \Theta) &= \frac{P(x = THT, z = H|\Theta)}{P(x = THT|\Theta)} \\ &= \frac{\lambda p_1(1-p_1)^2}{\lambda p_1(1-p_1)^2 + (1-\lambda)p_2(1-p_2)^2} \end{aligned}$$

这样，可以从观测数据来推测未观测数据的概率分布，即从硬币正面和反面观测结果来推测硬币0投掷为正面或反面这一隐变量。

上述对参数λ（硬币0正面朝上概率）的估计方法，本质上还是贝叶斯方法，给定P1,P2，以及试验结果，对参数λ进行更新。

（3）假设隐变量也可以被全部观测到，那么对模型参数（三枚硬币各自朝上的概率）的估计如下：

情况1 为了了解未观测数据对模型参数求解带来的影响，这里先假设一个简单情况，在知道全部的数据（即包括上述的观测数据和未观测数据）情况下，来计算模型参数。

假设所观测的数据如下：$(<HHH>,H),(<TTT>,T),(<HHH>,H),(<TTT>,T),(<HHH>,H)$

则可如下来估计参数Θ取值，其中

$$\hat{\lambda}=\frac{count(\text{硬币0掷出正面})}{count(\text{试验次数})}=\frac{3}{5}$$

$$\widehat{p_1}=\frac{count(\text{硬币1掷出正面})}{count(\text{硬币1掷出次数})}=\frac{9}{3*3}=1$$

$$\widehat{p_2}=\frac{count(\text{硬币2掷出正面})}{count(\text{硬币2掷出次数})}=\frac{0}{2*3}=0$$

（4）一般情况：无法得知隐变量取值时，对参数的估计。本质上还是贝叶斯方法，给定本轮假设的模型参数，以及试验结果，对参数取值进行更新。更新得到的参数将会在下一轮迭代中使用。

情况2 在硬币0所掷出结果无法观测情况下，假设得到了如下部分不完全观测数据：

$$(<HHH>,_),(<TTT,_>),(<HHH>,_),(<TTT>,_),(<HHH>,_)$$

与前面的推导类似，如下计算这一观测结果是由硬币0投掷为正面或反面所引起的概率（求取期望步骤）。这一过程也就是计算隐变量取值（硬币0所投掷的结果）。

$$P(z=H|x=HHH,\Theta)=\frac{\lambda p_1^3}{\lambda p_1^3+(1-\lambda)p_2^3}$$

$$P(z=T|x=HHH,\Theta)=1-P(z=H|x=HHH,\Theta)$$

$$P(z=H|x=TTT,\Theta)=\frac{\lambda(1-p_1)^3}{\lambda(1-p_1)^3+(1-\lambda)(1-p_2)^3}$$

$$P(z=T|x=TTT,\Theta)=1-P(z=H|x=TTT,\Theta)$$

在上述对隐变量概率估计基础上，则可估计参数Θ的更新值（期望最大化步骤）：

$$\hat{\lambda}=\frac{3*P(z=H|x=HHH,\Theta)+2*P(z=H|x=TTT,\Theta)}{5}$$

$$\widehat{p_1}=\frac{3*3*P(z=H|x=HHH,\Theta)+0*2*P(z=H|x=TTT,\Theta)}{3*3*P(z=H|x=HHH,\Theta)+3*2*P(z=H|x=TTT,\Theta)}$$

$$\widehat{p_2}=\frac{3*3*P(z=T|x=HHH,\Theta)+0*2*P(z=T|x=TTT,\Theta)}{3*3*P(z=T|x=HHH,\Theta)+3*2*P(z=T|x=TTT,\Theta)}$$

（5）参数迭代过程举例

表5.6 三个硬币投掷中参数迭代更新过程

表5.6 三个硬币投掷中参数迭代更新过程

迭代次数	λ	p_1	p_2	$P(z=\mathrm{H}\|x=\mathrm{HHH},\Theta)$	$P(z=\mathrm{H}\|x=\mathrm{TTT},\Theta)$
1	0.5000	0.4000	0.8000	0.1111	0.9643
2	0.4524	0.1474	0.9739	0.0029	1.0000
3	0.4017	0.0043	1.0000	0.0000	1.0000
4	0.4000	0.0000	1.0000	0.0000	1.0000

表5.7 三个硬币投掷中参数初始值被更改后的迭代更新过程

表5.7 三个硬币投掷中参数初始值被更改后的迭代更新过程

迭代次数	λ	p_1	p_2	$P(z=\mathrm{H}\|x=\mathrm{HHH},\Theta)$	$P(z=\mathrm{H}\|x=\mathrm{TTT},\Theta)$
1	0.5000	0.5100	0.5000	0.5148	0.4849
2	0.5028	0.6143	0.5855	0.5388	0.449
3	0.5029	0.6428	0.5567	0.609	0.346
4	0.5038	0.7253	0.4728	0.7857	0.1255
5	0.5217	0.9037	0.2688	0.9765	0.0025
6	0.5869	0.9983	0.0342	1.0000	0.0000
7	0.6000	1.0000	0.0000	1.0000	0.0000

6.期望最大化算法（EM算法）的一般形式

①首先对模型参数进行假设，对各个参数进行赋值。

②基本思想：E步骤是根据假设好的参数，计算和优化隐变量分布；M步骤是固定隐变量分布，反过来更新模型参数。两者交替进行，直到算法收敛。

③构造对数似然函数下界->下界最大化，进行反复迭代，从而近似达到对数似然函数最大化的目的。

对于n个相互独立的样本$X=\{x_1,x_2,\dots,x_n\}$及其对应的隐变量$Z=\{z_1,z_2,\dots,z_n\}$，在假设样本的模型参数为Θ前提下，观测数据x_i的概率为$\mathrm{P}(x_i|\Theta)$，完全数据(x_i,z_i)的**似然函数**为$\mathrm{P}(x_i,z_i|\Theta)$。

在上面的表示基础上，优化目标为求解合适的Θ和Z使得**对数似然函数**最大：

$$(\Theta,\mathrm{Z})=\underset{\Theta,\mathrm{Z}}{\operatorname{argmax}}\,L(\Theta,\mathrm{Z})=\underset{\Theta,\mathrm{Z}}{\operatorname{argmax}}\sum_{i=1}^{n}log\sum_{z_i}P(x_i,z_i|\Theta)$$

但是，优化求解含有未观测数据Z的对数似然函数$L(\Theta,\mathrm{Z})$十分困难，EM算法不断构造对数似然函数$L(\Theta,\mathrm{Z})$的一个下界（E步骤），然后最大化这个下界（M步骤），以迭代方式逼近模型参数所能取得极大似然值。

④下界是根据Jensen不等式得到的，计算的是原函数取对数后的数学期望。

$$\sum_{i=1}^{n} log \sum_{z_i} P(x_i, z_i|\Theta) = \sum_{i=1}^{n} log \sum_{z_i} Q_i(z_i) \frac{P(x_i, z_i|\Theta)}{Q_i(z_i)}$$

$$\geq \underbrace{\sum_{i=1}^{n} \sum_{z_i} Q_i(z_i) log \frac{P(x_i, z_i|\Theta)}{Q_i(z_i)}}_{\text{对数似然函数下界}}$$

在上式中，$Q_i(z_i)$是隐变量分布，满足：$\sum_{z_i} Q_i(z_i) = 1\ (0 \leq Q_i(z_i))$。

上述不等式中使用了Jensen 不等式 (Jensen's inequality)。对于凹函数f，Jensen 不等式使得下面不等式成立：

$$\log(\mathrm{E}(f)) \geq \mathrm{E}(\log(f)), \text{其中} \mathrm{E}(f) = \sum_i \lambda_i f_i, \lambda_i \geq 0, \quad \sum_i \lambda_i = 1$$

求解目标： $(\Theta, \mathrm{Z}) = \underset{\Theta,\mathrm{Z}}{\operatorname{argmax}}\, L(\Theta, \mathrm{Z}) = \underset{\Theta,\mathrm{Z}}{\operatorname{argmax}} \sum_{i=1}^{n} log \sum_{z_i} P(x_i, z_i|\Theta)$

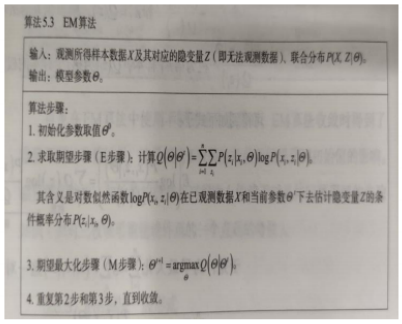
令$f_i = \frac{P(x_i, z_i|\Theta)}{Q_i(z_i)}$和$\lambda_i = Q_i(z_i)$，则根据Jensen 不等式的定义，可将$\frac{P(x_i, z_i|\Theta)}{Q_i(z_i)}$视为第$i$个样本，$Q_i(z_i)$为第$i$个样本的权重。按照这样的约定，可得到如下式子：

$$E\left(log \frac{P(x_i, z_i|\Theta)}{Q_i(z_i)}\right) = \sum_{z_i} \underbrace{Q_i(z_i)}_{\text{权重}} \underbrace{\log \frac{P(x_i, z_i|\Theta)}{Q_i(z_i)}}_{\text{样本值}}$$

于是，为了最大化$\sum_{i=1}^{n} log \sum_{z_i} P(x_i, z_i|\Theta)$这一对数似然函数，只需最大化其下界$\sum_{i=1}^{n} \sum_{z_i} Q_i(z_i) \log \frac{P(x_i, z_i|\Theta)}{Q_i(z_i)}$。这个下界实际上是$\log \frac{P(x_i, z_i|\Theta)}{Q_i(z_i)}$的加权求和。由于权重$Q_i(z_i)$累加之和为1，因此$\sum_{z_i} Q_i(z_i) \log \frac{P(x_i, z_i|\Theta)}{Q_i(z_i)}$就是$\log \frac{P(x_i, z_i|\Theta)}{Q_i(z_i)}$的加权平均，也就是所谓的期望，这就是EM算法中Expectation这一单词的来源。于是，EM算法就是不断最大化这一下界（M步骤），从而通过迭代的方式逼近模型参数的极大似然估计值。

⑤广义EM算法中，E步骤是固定参数来优化隐变量分布，M步骤是固定隐变量分布来优化参数，两者不同交替迭代。至此，证明了EM算法能够通过不断最大化下界来逼近最大似然估计值。

算法5.3 EM算法



算法5.3 EM算法

输入：观测所得样本数据X及其对应的隐变量Z（即无法观测数据），联合分布$P(X, Z|\theta)$。
输出：模型参数θ。

算法步骤：

1. 初始化参数取值θ^0。
2. 求取期望步骤（E步骤）：计算$Q(\theta|\theta^t)=\sum_{i=1}^{n}\sum_{z_i}P(z_i|x_i,\theta)\log P(x_i,z_i|\theta)$。

   其含义是对数似然函数$\log P(x_i, z_i|\theta)$在已观测数据X和当前参数θ^t下去估计隐变量Z的条件概率分布$P(z_i|x_i, \theta)$。

3. 期望最大化步骤（M步骤）：$\theta^{t+1}=\underset{\theta}{\operatorname{argmax}}\,Q(\theta|\theta^t)$。
4. 重复第2步和第3步，直到收敛。

在算法5.3中，固定上一步所得参数Θ^t情况下，通过计算Z的后验概率来得到能够逼近真实最大似然估计的下界，与算法中的求取期望步骤一致。在期望最大化步骤中，通过最大化对数似然函数，可以得到新的参数Θ^{t+1}，即：

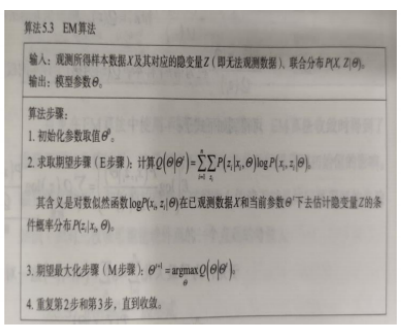
$$\Theta^{t+1}=\underset{\Theta}{\operatorname{argmax}}\sum_{i=1}^{n}\sum_{z_i}Q_i(z_i)log\frac{P(x_i,z_i|\Theta)}{Q_i(z_i)}$$

因为$Q_i(z_i)=p(z_i|x_i,\Theta)$是由上一步的$\Theta^t$估计出的，与下一步要优化的$\Theta^{t+1}$无关，所以上式等价于：

$$\begin{aligned}&\Theta^{t+1}\\&=\underset{\Theta}{\operatorname{argmax}}\sum_{i}\sum_{z_i}Q_i(z_i)\log p(x_i,z_i|\Theta)-Q_i(z_i)\log Q_i(z_i)\\&=\underset{\Theta}{\operatorname{argmax}}\sum_{i}\sum_{z_i}Q_i(z_i)\log p(x_i,z_i|\Theta)\end{aligned}$$

广义EM算法中，E步骤是固定参数来优化隐变量分布，M步骤是固定隐变量分布来优化参数，两者不同交替迭代。至此，证明了EM算法能够通过不断最大化下界来逼近最大似然估计值。

算法5.3 EM算法



算法5.3 EM算法

输入：观测所得样本数据X及其对应的隐变量Z（即无法观测数据），联合分布$P(X, Z|\theta)$。
输出：模型参数θ。

算法步骤：

1. 初始化参数取值θ^0。
2. 求取期望步骤（E步骤）：计算$Q(\theta|\theta^t)=\sum_{i=1}^{n}\sum_{z_i}P(z_i|x_i,\theta)\log P(x_i,z_i|\theta)$。

   其含义是对数似然函数$\log P(x_i, z_i|\theta)$在已观测数据X和当前参数θ^t下去估计隐变量Z的条件概率分布$P(z_i|x_i, \theta)$。

3. 期望最大化步骤（M步骤）：$\theta^{t+1}=\underset{\theta}{\operatorname{argmax}}\,Q(\theta|\theta^t)$。
4. 重复第2步和第3步，直到收敛。

在算法5.3中，固定上一步所得参数Θ^t情况下，通过计算Z的后验概率来得到能够逼近真实最大似然估计的下界，与算法中的求取期望步骤一致。在期望最大化步骤中，通过最大化对数似然函数，可以得到新的参数Θ^{t+1}，即：

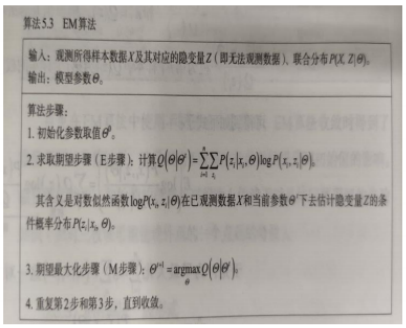
$$\Theta^{t+1}=\underset{\Theta}{\operatorname{argmax}}\sum_{i=1}^{n}\sum_{z_i}Q_i(z_i)log\frac{P(x_i,z_i|\Theta)}{Q_i(z_i)}$$

因为$Q_i(z_i)=p(z_i|x_i,\Theta)$是由上一步的$\Theta^t$估计出的，与下一步要优化的$\Theta^{t+1}$无关，所以上式等价于：

$$\begin{aligned}&\Theta^{t+1}\\&=\underset{\Theta}{\operatorname{argmax}}\sum_{i}\sum_{z_i}Q_i(z_i)\log p(x_i,z_i|\Theta)-Q_i(z_i)\log Q_i(z_i)\\&=\underset{\Theta}{\operatorname{argmax}}\sum_{i}\sum_{z_i}Q_i(z_i)\log p(x_i,z_i|\Theta)\end{aligned}$$

广义EM算法中，E步骤是固定参数来优化隐变量分布，M步骤是固定隐变量分布来优化参数，两者不同交替迭代。至此，证明了EM算法能够通过不断最大化下界来逼近最大似然估计值。

算法5.3 EM算法



算法5.3 EM算法

输入：观测所得样本数据X及其对应的隐变量Z（即无法观测数据），联合分布$P(X, Z|\theta)$。
输出：模型参数θ。

算法步骤：

1. 初始化参数取值θ^0。
2. 求取期望步骤（E步骤）：计算$Q(\theta|\theta^t)=\sum_{i=1}^{n}\sum_{z_i}P(z_i|x_i,\theta)\log P(x_i,z_i|\theta)$。

   其含义是对数似然函数$\log P(x_i,z_i|\theta)$在已观测数据X和当前参数θ^t下去估计隐变量Z的条件概率分布$P(z_i|x_i,\theta)$。

3. 期望最大化步骤（M步骤）：$\theta^{t+1}=\underset{\theta}{\operatorname{argmax}}\,Q(\theta|\theta^t)$。
4. 重复第2步和第3步，直到收敛。

假设Θ^{t+1}和Θ^t是两个相邻的迭代更新得到的参数值，如果能证明$L(\Theta^t)\le L(\Theta^{t+1})$，即可认为EM算法能够让似然单调增长。下面给出简略的证明：

$$L(\Theta^{t+1})\ge\sum_i\sum_{z_i}Q_i(z_i)\log\frac{p(x_i,z_i|\Theta^{t+1})}{Q_i(z_i)}$$

$$\Rightarrow l(\Theta^{t+1})\ge\sum_i\sum_{z_i}Q_i^t(z_i)\log\frac{p(x_i,z_i|\Theta^{t+1})}{Q_i^t(z_i)}\text{（特殊化）}$$

因为$\Theta^{t+1}=\underset{\Theta}{\operatorname{argmax}}\sum_i\sum_{z_i}Q_i(z_i)\log\frac{p(x_i,z_i|\Theta)}{Q_i(z_i)}$，即对于$\Theta^t$（事实上可以是任意的$\Theta$）都有下面的式子成立：

$$\begin{aligned}&\sum_i\sum_{z_i}Q_i^t(z_i)\log\frac{p(x_i,z_i|\Theta^{t+1})}{Q_i^t(z_i)}\\&\ge\sum_i\sum_{z_i}Q_i^t(z_i)\log\frac{p(x_i,z_i|\Theta^{t})}{Q_i^t(z_i)}\\&=L(\Theta^t)\end{aligned}$$

至此，证明了$L(\Theta^{t+1})\ge L(\Theta^t)$，即EM算法能够保证似然度取值单调增长，**即EM算法能够稳定收敛**

第五章 MOOC习题答案

1.由于K均值聚类是一个迭代过程，我们需要设置其迭代终止条件。下面哪句话正确描述了K均值聚类的迭代终止条件（ D ）

A.已经达到了迭代次数上限,或者每个待聚类样本分别归属唯一一个聚类集合

B.已经形成了K个聚类集合，或者已经达到了迭代次数上限

C.已经形成了K个聚类集合，或者每个待聚类样本分别归属唯一一个聚类集合

D.已经达到了迭代次数上限,或者前后两次迭代中聚类质心基本保持不变

2.我们可以从最小化每个类簇的方差这一视角来解释K均值聚类的结果，下面对这一视角描述不正确的是（ A ）

A.每个簇类的质心累加起来最小

B.每个簇类的方差累加起来最小

C.每个样本数据分别归属于与其距离最近的聚类质心所在聚类集合

D.最终聚类结果中每个聚类集合中所包含数据呈现出来差异性最小

3.下面哪一句话描述不属于K均值聚类算法的不足(D)

A.需要初始化聚类质心

B.算法迭代执行

C.需要事先确定聚类数目

D.K均值聚类是无监督聚类(理解：所得的聚类并不一定是我们想要的结果，可解释性、泛化能力都有可能是不够的)

4.下面对数据样本方差解释不正确的是（ C ）

A.方差（样本方差）是每个样本数据与全体样本数据平均值之差的平方和的平均数

B.方差描述了样本数据在平均值附近的波动程度

C.方差刻画了样本数据的大小

D.方差刻画了随机变量或一组数据离散程度

5.协方差可以用来计算两个变量之间的相关性，或者说计算两维样本数据中两个维度之间的相关性。基于计算所得的协方差值，可以来判断样本数据中两维变量之间是否存在关联关系。下面哪一个说法是不正确的（ A ）

A.当协方差值等于0 时，则两个变量线性正相关

B.当协方差值小于0 时，则两个变量线性负相关

C.当协方差值等于0 时，则两个变量线性不相关

D.当协方差值大于0 时，则两个变量线性正相关

6.下面对相关性(correlation)和独立性(independence)描述不正确的是（ D ）

A.如果两维变量线性不相关，则皮尔逊相关系数等于0

B.如果两维变量彼此独立，则皮尔逊相关系数等于0

C.独立指两个变量彼此之间不相互影响

D.“不相关”是一个比“独立”要强的概念，即不相关一定相互独立

7.下面对主成分分析的描述不正确的是（ D ）

A.主成份分析是一种特征降维方法

B.在主成分分析中，将数据向方差最大方向进行投影，可使得数据所蕴含信息没有丢失，以便在后续处理过程中各个数据“彰显个性”(近似认为是对的)

C.主成分分析可保证原始高维样本数据被投影映射后，其方差保持最大

D.在主成分分析中，所得低维数据中每一维度之间具有极大相关度

8.假设原始数据个数为n，原始数据维数为d，降维后的维数为l，下面对主成分分析算法描述不正确的是（ B ）

A.每个原始数据的维数大小从d变成了l

B.主成分分析学习得到了l个d维大小的向量，这l个d维向量之间彼此相关

C.主成分分析要学习一个映射矩阵，其大小是d×l

D.在主成分分析中，我们将带约束的最优化问题，通过拉格朗日乘子法将其转化为无约束最优化问题

9.在主成分分析中，我们将带约束的最优化问题，通过拉格朗日乘子法将其转化为无约束最优化问题。下面对主成分分析中优化的目标函数和约束条件描述正确的是 （ C ）

A.保证映射投影（即降维后）所得结果冗余度最小以及投影后结果方差最大

B.保证映射投影（即降维后）所得结果相关度最大以及投影后结果方差最大

C.保证映射投影（即降维后）所得结果方差最大以及投影方向正交（以去除冗余度）

D.保证映射投影方向之间的方差最大以及投影方向正交（以去除冗余度）

10.下面对特征人脸算法描述不正确的是（ B ）

A.每一个特征人脸的维数与原始人脸图像的维数一样大

B.特征人脸之间的相关度要尽可能大

C.特征人脸方法是一种应用主成分分析来实现人脸图像降维的方法

D.特征人脸方法是用一种称为“特征人脸(eigenface)”的特征向量按照线性组合形式来表达每一张原始人脸图像

第六章 深度学习

一、前馈神经网络

1.端到端学习

①深度学习所得模型可视为一个复杂函数

②非线性变换与映射的过程：像素点→语义

2.神经元结构

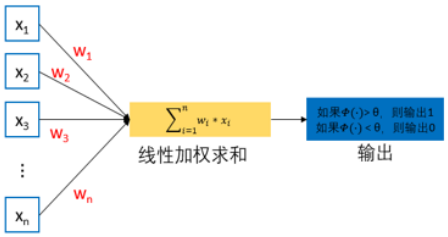


图6.1 MCP模型结构

给定n个二值化（0或1）的输入数据 $x_i(1 \le i \le n)$ 与连接参数$w_i(1 \le i \le n)$，MCP神经元模型对输入数据线性加权求和，然后使用函数$\boldsymbol{\Phi}(\cdot)$将加权累加结果映射为0或1，以完成两类分类的任务：

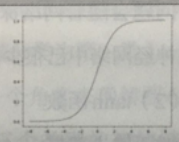
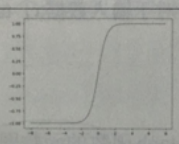
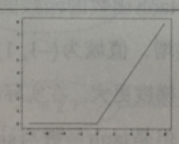
$$y = \boldsymbol{\Phi}(\sum_{i=1}^{n} w_i x_i)$$

其中w_i为预先设定的连接权重值（一般在0和1中一个值或者1和-1中取一个值），用来表示其所对应输入数据对输出结果的影响（即权重）。$\Phi(\cdot)$将输入端数据与连接权重所得线性加权累加结果与预先设定阈值θ进行比较，根据比较结果输出1或0。

3.常见的非线性激活函数

表6.1 激活函数基本性质

表6.1 激活函数基本性质

名称	函数	图像	导数	值域
sigmoid	$f(x)=\frac{1}{1+e^{-x}}$		$f'(x)=f(x)*(1-f(x))$	(0, 1)
tanh	$f(x)=\frac{1-e^{-2x}}{1+e^{-2x}}$		$f'(x)=1-f(x)^2$	(−1, 1)
ReLU	$f(x)=\begin{cases}0, \text{ for } x\leqslant 0\\ x, \text{ for } x>0\end{cases}$		$f(x)=\begin{cases}0, \text{ for } x\leqslant 0\\ 1, \text{ for } x>0\end{cases}$	$[0, +\infty)$

神经网络使用非线性函数作为激活函数（activation function），通过对多个非线性函数进行组合，来实现对输入信息的非线性变换

Softmax函数一般用于多分类问题中，其将输入数据x_i映射到第i个类别的概率y_i如下计算：

$$y_i = \text{softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{k} e^{x_j}}$$

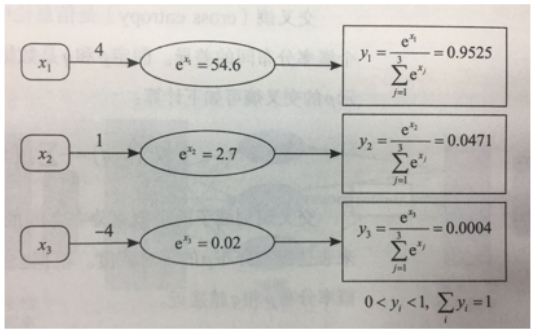
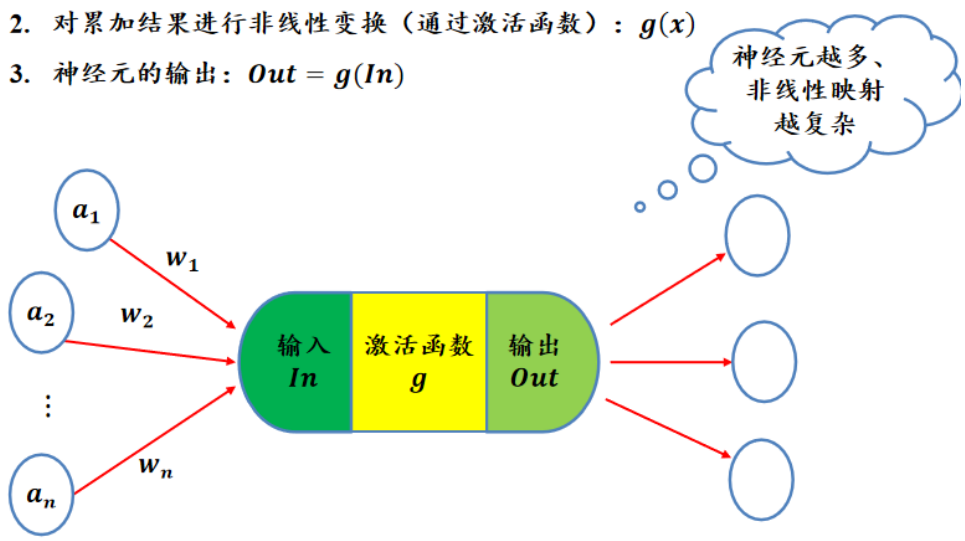


图6.2 三分类问题的 softmax 输出示意

- 对于取值为4、1和-4的x_1、x_2和x_3，通过softmax变换后，将其映射到(0,1)之间的概率值。
- 由于Softmax输出结果的值累加起来为1，因此可将输出概率最大的作为分类目标。

神经元是深度学习模型中基本单位，可以如下刻画神经元功能：

1. 对相邻前向神经元输入信息进行加权累加：$In = \sum_{i=1}^{n} w_i * a_i$
2. 对累加结果进行非线性变换（通过激活函数）：$g(x)$
3. 神经元的输出：$Out = g(In)$



4.损失函数

（1）定义：损失函数（Loss Function）又称为代价函数（Cost Function）用来计算模型预测值与真实值之间的误差。损失函数是神经网络设计中的一个重要组成部分。通过定义与任务相关的良好损失函数，在训练过程中可根据损失函数来计算神经网络的误差大小，进而优化神经网络参数。

两种最常用损失函数：均方误差损失函数、交叉熵损失函数。

（2）均方误差损失函数

均方误差损失函数通过计算预测值和实际值之间距离（即误差）的平方来衡量模型优劣。假设有n个训练数据x_i，每个训练数据x_i的真实输出为y_i，模型对x_i的预测值为$\hat{y}_i$。该模型在n个训练数据下所产生的均方误差损失可定义如下：

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

（3）交叉熵损失函数

交叉熵（cross entropy）是信息论中的重要概念，主要用来度量两个概率分布间的差异。假定p和q是数据x的两个概率分布，通过q来表示p的交叉熵可如下计算：

$$\mathrm{H}(p,\ q) = -\sum_{x} p(x) * log q(x)$$

交叉熵刻画了两个概率分布之间的距离，旨在描绘通过概率分布q来表达概率分布p的困难程度。根据公式不难理解，交叉熵越小，两个概率分布p和q越接近。

交叉熵损失函数

假设数据x属于类别1。记数据x的类别分布概率为y，显然$y=(1,0,0)$代表数据x的实际类别分布概率。记$\hat{y}$代表模型预测所得类别分布概率。

那么对于数据x而言，其实际类别分布概率y和模型预测类别分布概率$\hat{y}$的交叉熵损失函数定义为：

$$\text{cross entropy} = -\,y \times \log(\hat{y})$$

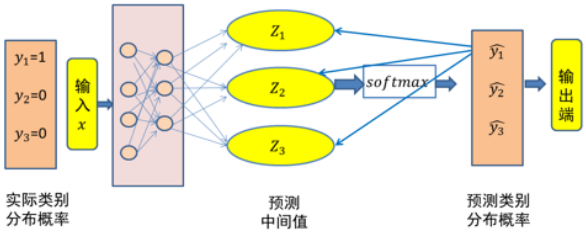


图6.3 三类分类问题中输入x的交叉熵损失示意图（x属于第一类）

交叉熵损失函数

很显然，一个良好的神经网络要尽量保证对于每一个输入数据，神经网络所预测类别分布概率与实际类别分布概率之间的差距越小越好，即交叉熵越小越好。于是，可将交叉熵作为损失函数来训练神经网络。

图6.3给出了一个三个类别分类的例子。由于输入数据x属于类别1，因此其实际类别概率分布值为$y=(y_1,y_2,y_3)=(1,0,0)$。经过神经网络的变换，得到了输入数据x相对于三个类别的预测中间值(z_1,z_2,z_3)。然后，经过Softmax函数映射，得到神经网络所预测的输入数据x的类别分布概率$\hat{y}=(\hat{y_1},\hat{y_2},\hat{y_3})$。根据前面的介绍，$\hat{y_1}$、$\hat{y_2}$和$\hat{y_3}$为(0,1)范围之间的一个概率值。由于样本$x$属于第一个类别，因此希望神经网络所预测得到的$\hat{y_1}$取值要远远大于$\hat{y_2}$和$\hat{y_3}$的取值。

交叉熵损失函数

训练中可利用如下交叉熵损失函数来对模型参数进行优化：cross entropy $=-(y_1 \times \log(\widehat{y_1}) + y_2 \times \log(\widehat{y_2}) + y_3 \times \log(\widehat{y_3}))$

在上式中，y_2和y_3均为0、y_1为1，因此交叉熵损失函数简化为：$-y_1 \times \log(\widehat{y_1}) = -\log(\widehat{y_1})$。

在神经网络训练中，要将输入数据实际的类别概率分布与模型预测的类别概率分布之间的误差（即损失）从输出端向输入端传递，以便来优化模型参数。下面简单介绍根据交叉熵计算得到的误差从$\widehat{y_1}$传递给z_1和z_2（z_3的推导与z_2相同）的情况。

$$\frac{\partial \widehat{y_1}}{\partial z_1} = \frac{\partial}{\partial z_1}\left(\frac{e^{z_1}}{\sum_k e^{z_k}}\right) = \frac{(e^{z_1})' \times \sum_k e^{z_k} - e^{z_1} \times e^{z_1}}{(\sum_k e^{z_k})^2} = \frac{e^{z_1}}{\sum_k e^{z_k}} - \frac{e^{z_1}}{\sum_k e^{z_k}} \times \frac{e^{z_1}}{\sum_k e^{z_k}}$$
$$= \widehat{y_1}(1-\widehat{y_1})$$

（注意：在推导中应用了$(e^x)' = e^x$、$(\frac{v}{u})' = \frac{v'u - vu'}{u^2}$等公式）

由于交叉熵损失函数$-\log(\widehat{y_1})$对$\widehat{y_1}$求导的结果为$-\frac{1}{\widehat{y_1}}$，$\widehat{y_1}(1-\widehat{y_1})$与$-\frac{1}{\widehat{y_1}}$相乘为$\widehat{y_1}-1$。这说明一旦得到模型预测输出$\widehat{y_1}$，将该输出减去1就是交叉损失函数相对于$z_1$的偏导结果。

$$\frac{\partial \widehat{y_1}}{\partial z_2} = \frac{\partial}{\partial z_2}\left(\frac{e^{z_1}}{\sum_k e^{z_k}}\right) = \frac{0 \times \sum_k e^{z_k} - e^{z_1} \times e^{z_2}}{(\sum_k e^{z_k})^2} = -\frac{e^{z_1}}{\sum_k e^{z_k}} \times \frac{e^{z_2}}{\sum_k e^{z_k}} = -\widehat{y_1}\widehat{y_2}$$

同理，交叉熵损失函数导数为$-\frac{1}{\widehat{y_1}}$，$-\widehat{y_1}\widehat{y_2}$与$-\frac{1}{\widehat{y_1}}$相乘结果为$\widehat{y_2}$。这意味对于除第一个输出节点以外的节点进行偏导，在得到模型预测输出后，只要将其保存，就是交叉损失函数相对于其他节点的偏导结果。在z_1、z_2和z_3得到偏导结果后，再通过链式法则（后续介绍）将损失误差继续往输入端传递即可。

交叉熵损失函数

- 在上面的例子中，假设所预测中间值(z_1, z_2, z_3)经过Softmax映射后所得结果为(0.34,0.46,0.20)。由于已知输入数据x属于第一类，显然这个输出不理想而需要对模型参数进行优化。如果选择交叉熵损失函数来优化模型，则(z_1, z_2, z_3)这一层的偏导值为 $(0.34-1, 0.46, 0.20) = (-0.66, 0.46, 0.20)$。
- 可以看出，Softmax和交叉熵损失函数相互结合，为偏导计算带来了极大便利。偏导计算使得损失误差从输出端向输入端传递，来对模型参数进行优化。在这里，交叉熵与Softmax函数结合在一起，因此也叫Softmax损失（Softmax with cross-entropy loss）。

【理解】

①实际上，样本属于第1维，所以偏导数计算：z1对应的偏导数是输出-1,z2,z3的偏导数恰好等于输出值。这是求导得到的结论。

②偏导数为负，表明参数、权重需要增大；偏导数为正，表明参数、权重需要减小。

5.单层感知机

线性加权求和->激活函数->阈值判断，只能解决线性可分问题。

6.多层感知机

（1）多层感知机由输入层、输出层和至少一层的隐藏层构成。网络中各个隐藏层中神经元可接收相邻前序隐藏层中所有神经元传递而来的信息，经过加工处理后将信息输出给相邻后续隐藏层中所有神经元。

（2）各个神经元接受前一级的输入，并输出到下一级，模型中没有反馈

（3）层与层之间通过“全连接”进行链接，即两个相邻层之间的神经元完全成对连接，但层内的神经元不相互连接。

（4）前馈神经网络（feedforward neural network）

7.梯度下降法

（1）目的：使得损失函数最小化。

（2）梯度计算

梯度下降算法是一种使得损失函数最小化的方法。一元变量所构成函数f在x处梯度为：

$$\frac{df(x)}{dx} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

- 在多元函数中，梯度是对每一变量所求导数组成的向量
- 梯度的反方向是函数值下降最快的方向，因此是损失函数求解的方向

（3）

- 假设损失函数$f(x)$是连续可微的多元变量函数，其泰勒展开如下(Δx是微小的增量)：

$$f(x+\Delta x)=f(x)+f'(x)\Delta x+\frac{1}{2}f''(x)(\Delta x)^2+\ldots+\frac{1}{n!}f^{(n)}(x)\,(\Delta x)^n$$
$$f(x+\Delta x)-f(x)\approx(\nabla f(x))^T\Delta x$$

- 因为我们的目的是最小化损失函数$f(x)$，则$f(x+\Delta x)<f(x)$，于是$(\nabla f(x))^T\Delta x<0$

- 在$(\nabla f(x))^T\Delta x=\|\nabla f(x)\|\|\Delta x\|\cos\theta$中，$\|\nabla f(x)\|$和$\|\Delta x\|$分别为损失函数梯度的模和下一轮迭代中$x$取值增量的模，两者均为正数。为了保证损失误差减少，只要保证$\cos\theta<0$。当$\theta=180^\circ$时，$\cos\theta=-1$，这时损失函数减少的幅度值$(\nabla f(x))^T\Delta x$取到最小。

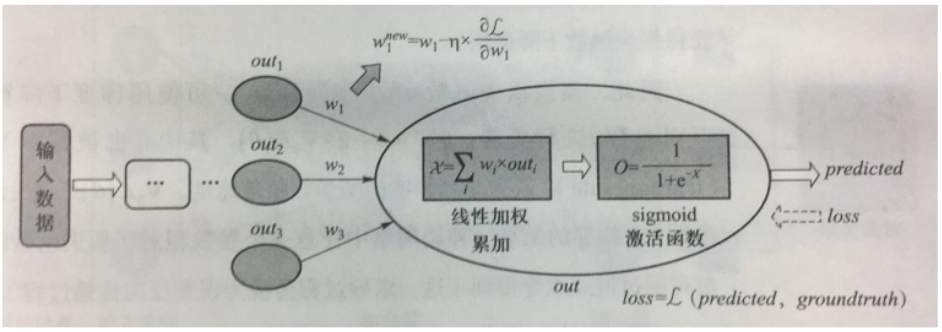
当$\theta=180^\circ$时，$f(x+\Delta x)$和$f(x)$之间的差值为：$f(x+\Delta x)-f(x)=\|\nabla f(x)\|\|\Delta x\|\cos\theta=-\|\Delta x\|\|\nabla f(x)\|$。这说明只要沿着损失函数梯度的反方向选取$x$的增量$\Delta x$，就会保证损失误差下降最多、下降最快，犹如从山峰处沿着最陡峭路径可快速走到山谷。在前面的推导中忽略了损失函数的二阶导数以及其高阶导数的取值，因此在实际中引入步长η，用$x-\eta\nabla f(x)$来更新x（在具体实现时η可取一个定值）。

从下面的公式可以计算得到变量的增量为$-\eta\nabla f(x)$

$$\underbrace{x-\eta\nabla f(x)}_{\text{变量更新后的取值}}-\underbrace{x}_{\text{变量当前取值}}=\underbrace{-\eta\nabla f(x)}_{\text{变量的增量}}$$

8.误差反向传播

- BP算法是一种将输出层误差反向传播给隐藏层进行参数更新的方法。
- 将误差从后向前传递，将误差分摊给各层所有单元，从而获得各层单元所产生的误差，进而依据这个误差来让各层单元负起各自责任、修正各单元参数。



6.7 链式求导与模型参数更新示意图

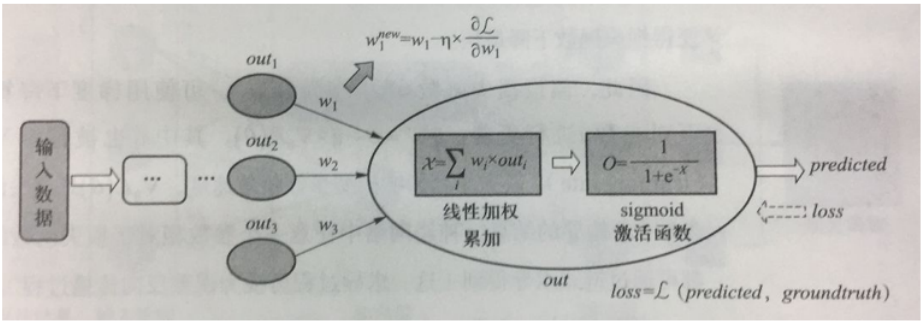


图6.7 链式求导与模型参数更新示意图

为了使损失函数$\mathcal{L}$取值减少（从而保证模型预测结果与实际结果之间的差距越来越小），需要求取损失函数$\mathcal{L}$相对于w_1的偏导，然后按照损失函数梯度的反方向选取一个微小的增量，来调整w_1的取值，就能够保证损失函数取值减少。即将w_1变为$w_1-\eta\frac{d\mathcal{L}}{dw_1}$后，能使得损失误差减少。这里$\frac{d\mathcal{L}}{dw_1}$为损失函数$\mathcal{L}$对$w_1$的偏导。于是，需要求取损失函数$\mathcal{L}$对变量参数$w_1$的偏导。

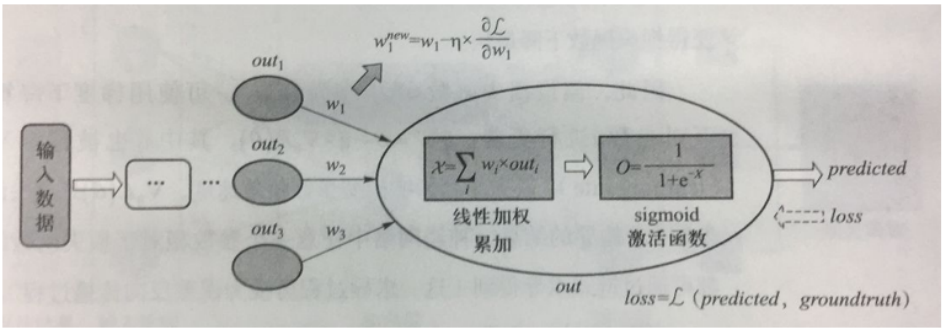


图6.7 链式求导与模型参数更新示意图

- 由于w_1与加权累加函数X和sigmoid函数均有关，因此$\frac{d\mathcal{L}}{dw_1}=\frac{d\mathcal{L}}{d\mathcal{O}}\frac{d\mathcal{O}}{dX}\frac{dX}{dw_1}$。在这个链式求导中：$\frac{d\mathcal{L}}{d\mathcal{O}}$与损失函数的定义有关；$\frac{d\mathcal{O}}{dX}$是对sigmoid函数求导，结果为$\frac{1}{1+e^{-X}}\times(1-\frac{1}{1+e^{-X}})$；$\frac{dX}{dw_1}$是加权累加函数$w_1\times out_1+w_2\times out_2+w_3\times out_3$对$w_1$求导，结果为$out_1$。
- 链式求导实现了损失函数对某个自变量求偏导，好比将损失误差从输出端向输入端逐层传播，通过这个传播过程来更新该自变量取值。梯度下降法告诉我们，只要沿着损失函数梯度的反方向来更新参数，就可使得损失函数下降最快。

- 参数w_1在下一轮迭代中的取值被调整为：$w_1^{new}=w_1-\eta\times\frac{d\mathcal{L}}{dw_1}=w_1-\eta\times\frac{d\mathcal{L}}{d\mathcal{O}}\frac{d\mathcal{O}}{dX}\frac{dX}{dw_1}=$

  $w_1-\eta\times\frac{d\mathcal{L}}{d\mathcal{O}}\times\frac{1}{1+e^{-X}}\times(1-\frac{1}{1+e^{-X}})\times out_1$。
- 按照同样的方法，可调整w_2、w_3以及其它图中未显示参数的取值。经过这样的调整，在模型参数新的取值作用下，损失函数$\mathcal{L}(\theta)$会以最快下降方式逐渐减少，直至减少到最小值（即全局最小值）。

9.举例

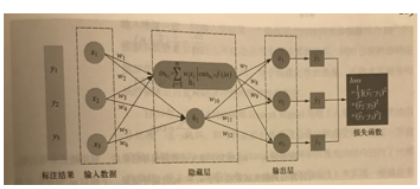


图6.8 包含一个隐藏层的多层感知机

下面以w_1为例介绍损失函数$\mathcal{L}$对w_1的偏导δ_1。

$$\delta_1=\frac{\partial\mathcal{L}}{\partial w_1}$$
$$=\frac{\partial\mathcal{L}}{\partial\widehat{y_1}}*\frac{\partial\widehat{y_1}}{\partial w_1}+\frac{\partial\mathcal{L}}{\partial\widehat{y_2}}*\frac{\partial\widehat{y_2}}{\partial w_1}+\frac{\partial\mathcal{L}}{\partial\widehat{y_3}}*\frac{\partial\widehat{y_3}}{\partial w_1}$$
$$=\frac{\partial\mathcal{L}}{\partial\widehat{y_1}}*\frac{\partial\widehat{y_1}}{\partial In_{o_1}}*\frac{\partial In_{o_1}}{\partial Out_{h_1}}*\frac{\partial Out_{h_1}}{\partial In_{h_1}}*\frac{\partial In_{h_1}}{\partial w_1}+\frac{\partial\mathcal{L}}{\partial\widehat{y_2}}*\frac{\partial\widehat{y_2}}{\partial In_{o_2}}*\frac{\partial In_{o_2}}{\partial Out_{h_1}}*\frac{\partial Out_{h_1}}{\partial In_{h_1}}*\frac{\partial In_{h_1}}{\partial w_1}+\frac{\partial\mathcal{L}}{\partial\widehat{y_3}}*\frac{\partial\widehat{y_3}}{\partial In_{o_3}}*\frac{\partial In_{o_3}}{\partial Out_{h_1}}*\frac{\partial Out_{h_1}}{\partial In_{h_1}}*\frac{\partial In_{h_1}}{\partial w_1}$$
$$=\left(\frac{\partial\mathcal{L}}{\partial\widehat{y_1}}*\frac{\partial\widehat{y_1}}{\partial In_{o_1}}*\frac{\partial In_{o_1}}{\partial Out_{h_1}}+\frac{\partial\mathcal{L}}{\partial\widehat{y_2}}*\frac{\partial\widehat{y_2}}{\partial In_{o_2}}*\frac{\partial In_{o_2}}{\partial Out_{h_1}}+\frac{\partial\mathcal{L}}{\partial\widehat{y_3}}*\frac{\partial\widehat{y_3}}{\partial In_{o_3}}*\frac{\partial In_{o_3}}{\partial Out_{h_1}}\right)*\frac{\partial Out_{h_1}}{\partial In_{h_1}}*\frac{\partial In_{h_1}}{\partial w_1}$$
$$=(\delta_7+\delta_8+\delta_9)*\frac{\partial Out_{h_1}}{\partial In_{h_1}}*\frac{\partial In_{h_1}}{\partial w_1}$$

在计算得到所有参数相对于损失函数$\mathcal{L}$的偏导结果后，利用梯度下降算法，通过$w_i^{new}=w_i-\eta*\delta_i$来更新参数取值。然后不断迭代，直至模型参数收敛，此时损失函数减少到其最小值。

二、卷积神经网络

1.基本思想

（1）图像中像素点具有很强的空间依赖性，卷积（convolution）就是针对像素点的空间依赖性来对图像进行处理的一种技术。

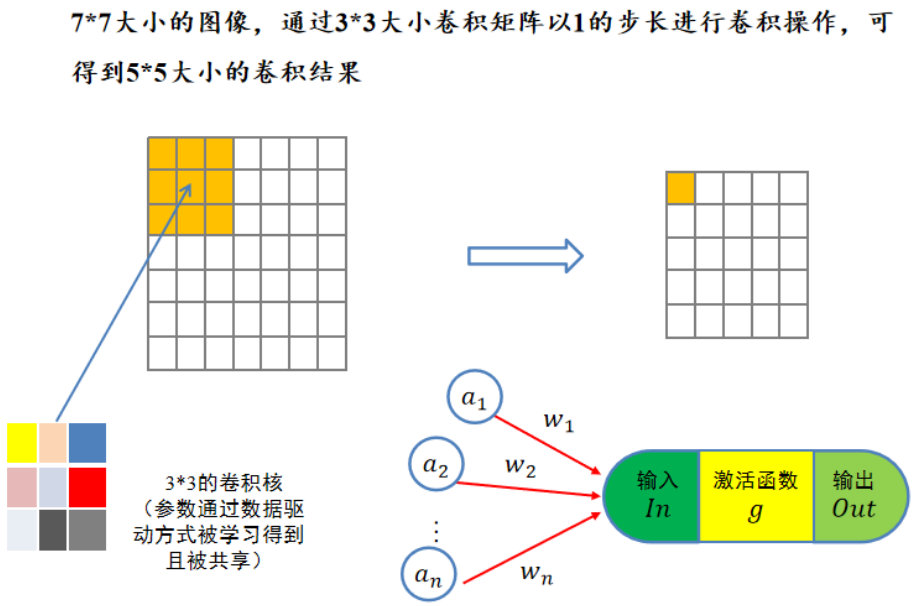
（2）在图像卷积计算中，需要定义一个卷积核（kernel）。卷积核是一个二维矩阵，矩阵中数值为对图像中与卷积核同样大小的子块像素点进行卷积计算时所采用的权重。

（3）卷积核中的权重系数wi是通过数据驱动机制学习得到，其用来捕获图像中某像素点及其邻域像素点所构成的特有空间模式。一旦从数据中学习得到权重系数，这些权重系数就刻画了图像中像素点构成的空间分布不同模式。

（4）如果卷积核中心位置的权重系数越小且与其它卷积权重系数差别越小，则卷积所得到图像滤波结果越模糊，这被称为图像平滑操作。

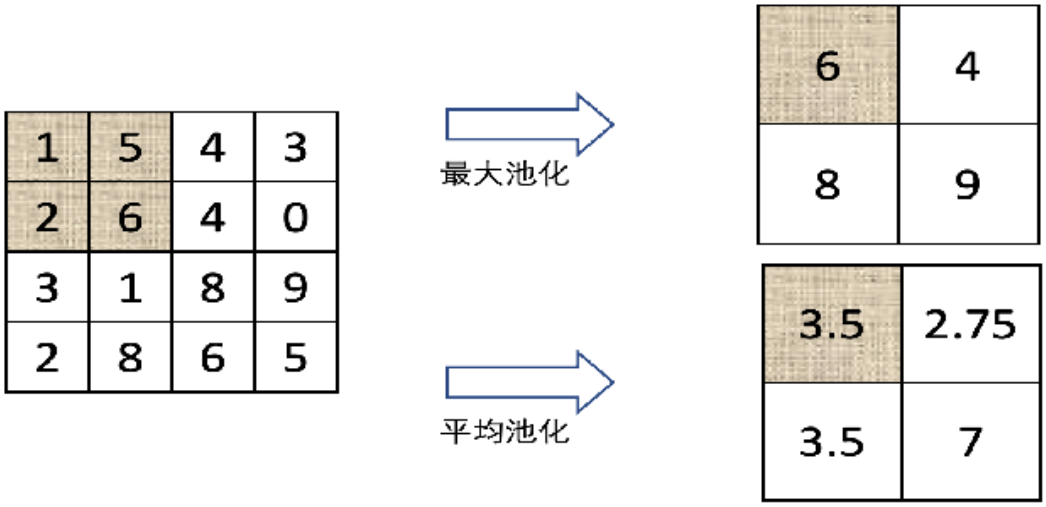
2.卷积操作

本质上就是加权叠加，通过合理大小的卷积核、合理的步长，可以实现数据的压缩。



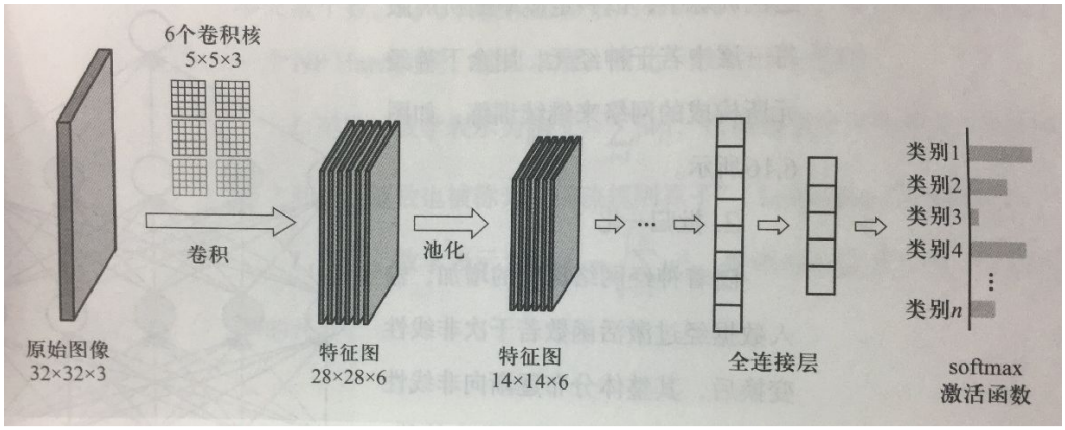
3.池化操作

由于图像中存在较多冗余，在图像处理中，可用某一区域子块的统计信息（如最大值或均值等）来刻画该区域中所有像素点呈现的空间分布模式，以替代区域子块中所有像素点取值，这就是卷积神经网络中池化(pooling)操作。池化操作对卷积结果特征图进行约减，实现了下采样，同时保留了特征图中主要信息。



4.卷积神经网络结构

卷积层->池化层->全连接层



5.对于输入的海量标注数据，通过多次迭代训练，卷积神经网络在若干次卷积操作、接着对卷积所得结果进行激活函数操作和池化操作下，最后通过全连接层来学习得到输入数据的特征表达，即分布式向量表达(distributed vector representation)。

6.【填空题】神经网络正则化的方法有：Dropout、批归一化、L1和L2正则化。

三、循环神经网络

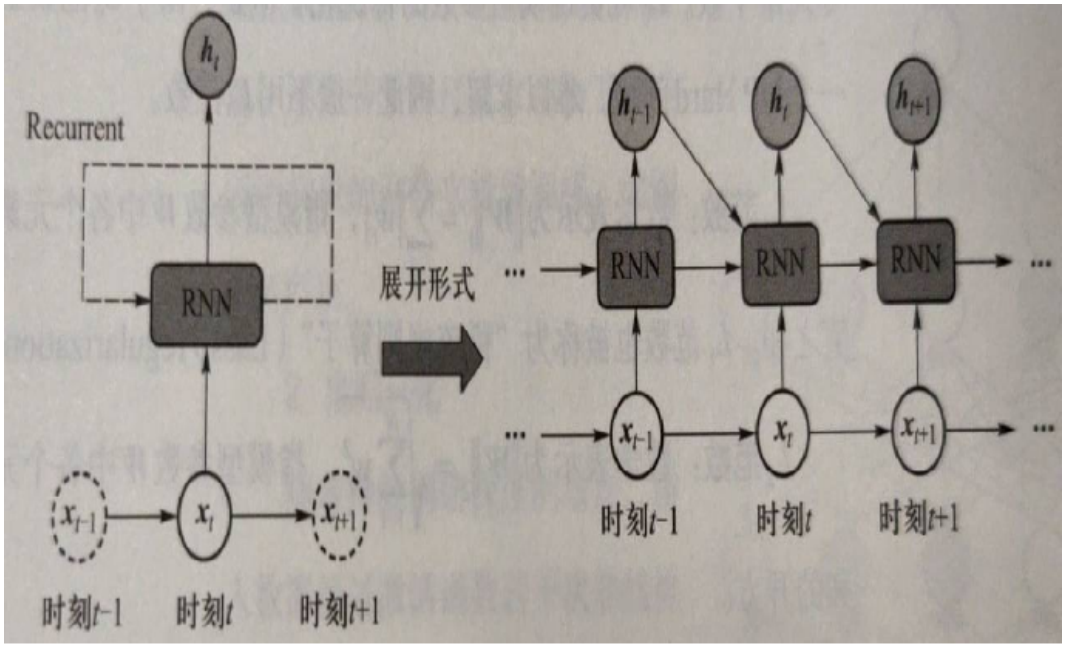
1.基本思想

（1）循环神经网络（Recurrent Neural Network, RNN）是一类处理序列数据（如文本句子、视频帧等）时所采用的网络结构。先前所介绍的前馈神经网络或卷积神经网络所需要处理的输入数据一次性给定，难以处理存在前后依赖关系的数据。

（2）循环神经网络的本质是希望模拟人所具有的记忆能力，在学习过程中记住部分已经出现的信息，并利用所记住的信息影响后续结点输出。循环神经网络在自然语言处理，例如语音识别、情感分析、机器翻译等领域有重要应用。

2.结构

（1）由“循环”两字可知，循环神经网络在处理数据过程中构成了一个循环体。对于一个序列数据，在每一时刻t，循环神经网络单元会读取当前输入数据x_t和前一时刻输入数据x_t−1所对应的隐式编码结果h_t−1，一同生成t时刻的隐式编码结果h_t。接着将h_t后传，去参与生成t+1时刻输入数据x_t+1的隐式编码h_t+1。



（2）时刻t所得到的隐式编码h_t是由上一时刻隐式编码h_t−1和当前输入x_t共同参与生成的，这可认为隐式编码h_t−1已经“记忆”了t时刻之前的时序信息，或者说前序时刻信息影响了后续时刻信息的处理。与前馈神经网络和卷积神经网络在处理时需要将所有数据一次性输入不同，这体现了循环神经网络可刻画序列数据存在时序依赖这一重要特点。

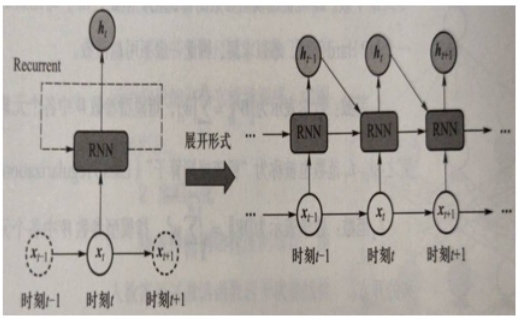


图6.17 循环神经网络结构及其展开形式示意图

在时刻t，一旦得到当前输入数据x_t，循环神经网络会结合前一时刻$t-1$得到的隐式编码h_{t-1}，如下产生当前时刻隐式编码h_t：

$$h_t = \Phi(\mathrm{U} \times x_t + \mathrm{W} \times h_{t-1})$$

这里$\Phi(\cdot)$是激活函数，一般可为Sigmoid或者Tanh激活函数，使模型能够忘掉无关的信息，同时更新记忆内容。U与W为模型参数。从这里可看出，当前时刻的隐式编码输出h_t不仅仅与当前输入数据x_t相关，与网络已有的“记忆”h_{t-1}也有着密不可分的联系。

为了更加直观展示在循环神经网络中前序时刻信息被“记住”来影响当前时刻信息编码，这里对上述公式进行如下变化：

$$h_t = \Phi(\mathrm{U} \times x_t + \mathrm{W} \times h_{t-1}) = \Phi\big(\mathrm{U} \times x_t + \mathrm{W} \times \Phi(\mathrm{U} \times x_{t-1} + \mathrm{W} \times h_{t-2})\big)$$

$$= \Phi\left(\mathrm{U} \times \underbrace{x_t}_{t\text{时刻输入}} + \mathrm{W} \times \Phi\left(\mathrm{U} \times \underbrace{x_{t-1}}_{t-1\text{时刻输入}} + \mathrm{W} \times \Phi(\mathrm{U} \times \underbrace{x_{t-2}}_{t-2\text{时刻输入}} + \cdots)\right)\right)$$

从上式可见，当前时刻编码中，的确均包含了历史信息，这也说明了循环神经网络能够记忆的原因，使得一些需要记忆历史过往信息的任务能够被循环神经网络有效处理。

（3）按照时间将循环神经网络展开后，可以得到一个和前馈神经网络相似的网络结构。这个网络结构可利用反向传播算法和梯度下降算法来训练模型参数，这种训练方法称为“沿时间反向传播算法（backpropagation through time，BPTT）”。由于循环神经网络每个时刻都有一个输出，所以在计算循环神经网络的损失时，通常需要将所有时刻（或者部分时刻）上的损失进行累加。

（4）梯度传递

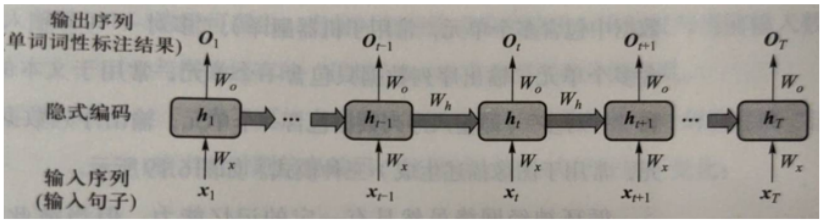


图6.20 循环神经网络中梯度传递示意图

以词性标注为例子给出了一个输入序列数据$(x_1, \cdots, x_{t-1}, x_t, x_{t+1}, \ldots, x_T)$被循环神经网络处理的示意图。在图6.20中，时刻$t$输入数据$x_t$（文本中的单词）的隐式编码为$h_t$、其真实输出为$y_t$（单词词性）、模型预测$x_t$的词性是$O_t$。参数$W_x$将$x_t$映射为隐式编码$h_t$、参数$W_o$将$h_t$映射为预测输出$O_t$、$h_{t-1}$通过参数$W_h$参与$h_t$的生成。图中$W_x$、$W_o$和$W_h$是复用参数。

（5）梯度消失问题

假设时刻t隐式编码如下得到：$h_t = \tanh(W_x x_t + W_h h_{t-1} + b)$。使用交叉熵损失函数计算时刻$t$预测输出与实际输出的误差$E_t$。显然，整个序列产生的误差为$E = \frac{1}{2}\sum_{t=1}^{T} E_t$。

下面介绍如何根据时刻t所得误差来更新参数W_x。在时刻t计算所得O_t不仅涉及到了时刻t的W_x，而且也涉及了前面所有时刻的W_x，按照链式求导法则，E_t在对W_x求导时候，也需要对前面时刻的W_x依次求导，然后再将求导结果进行累加，即：

$$\frac{\partial E_t}{\partial W_x} = \sum_{i=1}^{t} \frac{\partial E_t}{\partial O_t} \frac{\partial O_t}{\partial h_t} \left(\prod_{j=i+1}^{t} \frac{\partial h_j}{\partial h_{j-1}} \right) \frac{\partial h_i}{\partial W_x}$$

其中$\prod_{j=i+1}^{t} \frac{\partial h_j}{\partial h_{j-1}} = \prod_{j=i+1}^{t} tanh' \times W_h$

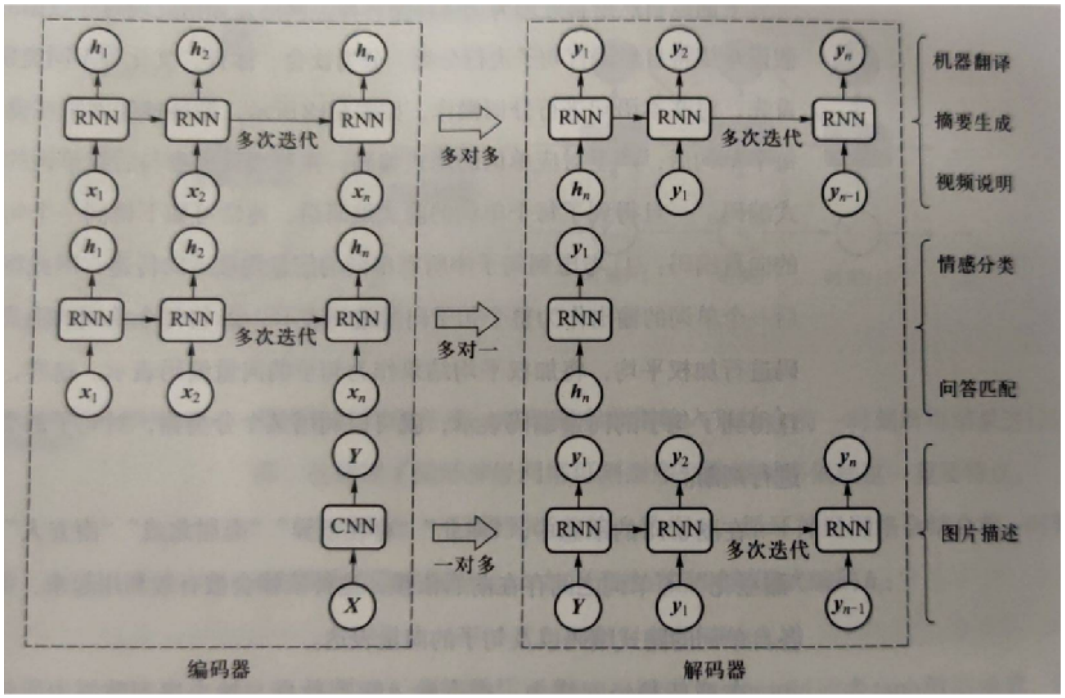
令$t = 3$，则有：

$$\frac{\partial E_3}{\partial W_x} = \frac{\partial E_3}{\partial O_3}\frac{\partial O_3}{\partial h_3}\frac{\partial h_3}{\partial W_x} + \frac{\partial E_3}{\partial O_3}\frac{\partial O_3}{\partial h_3}\frac{\partial h_3}{\partial h_2}\frac{\partial h_2}{\partial W_x} + \frac{\partial E_3}{\partial O_3}\frac{\partial O_3}{\partial h_3}\frac{\partial h_3}{\partial h_2}\frac{\partial h_2}{\partial h_1}\frac{\partial h_1}{\partial W_x}$$

由于$tanh$函数的导数取值位于0到1区间，对于长序列而言，若干多个0到1区间的小数相乘，会使得参数求导结果很小，引发**梯度消失问题**。E_t对W_h的求导类似，这里就不列出了。为了解决梯度消失问题，长短时记忆模型（Long Short-Term Memory，LSTM）被提出。

由于tan$h$函数的导数取值位于0到1区间，对于长序列而言，若干多个0到1区间的小数相乘，会使得参数求导结果很小，引发梯度消失问题。为了解决梯度消失问题，长短时记忆模型（Long Short-Term Memory，LSTM）被提出。

（6）根据输入序列数据与输出序列数据中所包含“单元”的多寡，循环神经网络可以实现“多对多”（即输入和输出序列数据中包含多个单元，常用于机器翻译）、“多对一”（即输入序列数据包含多个单元、输出序列数据只包含一个单元，常用于文本的情感分类）和“一对多”（即输入序列数据包含一个单元、输出序列数据包含多个单元，常用于图像描述生成）三种模式。



3.长短时记忆网络

（1）基本思想：

①与简单的循环神经网络结构不同，长短时记忆网络（Long Short-Term Memory，LSTM）中引入了内部记忆单元（internal memory cell）和门（gates）两种结构来对当前时刻输入信息以及前序时刻所生成信息进行整合和传递。在这里，内部记忆单元中信息可视为对“历史信息”的累积。

②常见的LSTM模型中有输入门(input gate)、遗忘门(forget gate)和输出门(output gate)三种门结构。对于给定的当前时刻输入数据x_t和前一时刻隐式编码h_t−1，输入门、遗忘门和输出门通过各自参数对其编码，分别得到三种门结构的输出i_t、f_t和o_t。

③在此基础上，再进一步结合前一时刻内部记忆单元信息c_t−1来更新当前时刻内部记忆单元信息c_t，最终得到当前时刻的隐式编码h_t。

（2）结构与计算

表6.3 长短时记忆网络中所用符号及其含义

符号	内容描述或操作
x_t	时刻t的输入数据
i_t	输入门的输出：$i_t = sigmoid(W_{xi}x_t + W_{hi}h_{t-1} + b_i)$ （W_{xi}、W_{hi}和b_i为输入门的参数）
f_t	遗忘门的输出：$f_t = sigmoid(W_{xf}x_t + W_{hf}h_{t-1} + b_f)$ （W_{xf}、W_{hf}和b_f为遗忘门的参数）
o_t	输出门的输出：$o_t = sigmoid(W_{xo}x_t + W_{ho}h_{t-1} + b_o)$ （W_{xo}、W_{ho}和b_o为输出门的参数）
c_t	内部记忆单元的输出：$c_t = f_t \odot c_{t-1} + i_t \odot tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$ （W_{xc}、W_{hc}和b_c为记忆单元的参数。输入门i_t控制有多少信息流入当前时刻内部记忆单元c_t、遗忘门控制上一时刻内部记忆单元c_{t-1}中有多少信息可累积到当前时刻内部记忆单元c_t）
h_t	时刻t输入数据的隐式编码: $h_t = o_t \odot \tanh(c_t)$ $= o_t \odot \tanh(f_t \odot c_{t-1} + i_t \odot tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c))$ (注：输入门、遗忘门和输出门的信息i_t、f_t、o_t一起参与得到h_t)
$\odot$	两个向量中对应元素按位相乘（(element-wise product）。 如: [10 6 3 7] $\odot$ [0.2 0.1 0.8 0.5] = [2 0.6 2.4 3.5]

当给定序列数据中当前时刻t的输入数据x_t和前一时刻隐式编码h_{t-1}时，可按照如下公式来计算内部记忆单元信息和隐式编码：

输入门信息输出：$i_t = sigmoid(W_{xi}x_t + W_{hi}h_{t-1} + b_i)$

遗忘门信息输出：$f_t = sigmoid(W_{xf}x_t + W_{hf}h_{t-1} + b_f)$

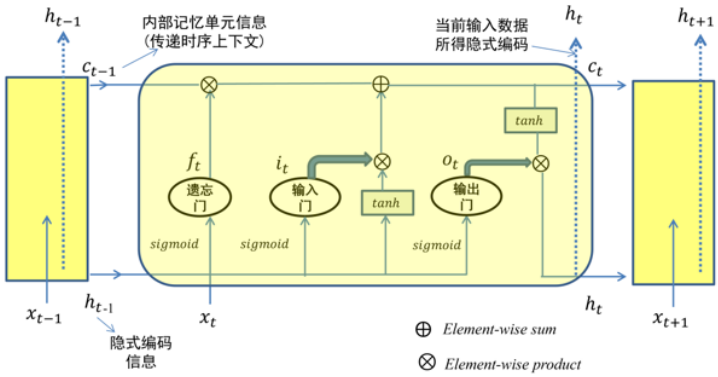
输出门信息输出：$o_t = sigmoid(W_{xo}x_t + W_{ho}h_{t-1} + b_o)$

内部记忆单元信息输出：$c_t = f_t \odot c_{t-1} + i_t \odot tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$

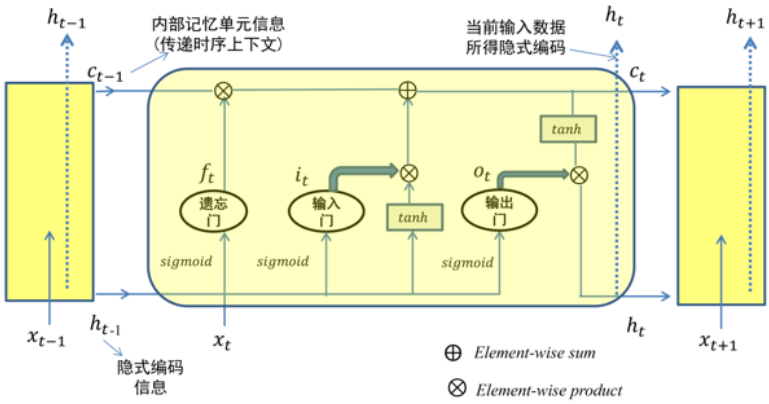
隐式编码输出：$h_t = o_t \odot \tanh(c_t)$或者$h_t = o_t \odot \tanh(f_t \odot c_{t-1} + i_t \odot tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c))$

输入门、遗忘门和输出门通过各自参数对当前时刻输入数据x_t和前一时刻隐式编码h_{t-1}处理后，利用$sigmoid$对处理结果进行非线性映射，因此三种门结构的输出i_t、f_t和o_t值域为(0,1)。正是由于三个门结构的输出值为位于0到1之间的向量，因此其在信息处理中起到了“调控开关”的“门”作用。三个门结构所输出向量的维数、内部记忆单元的维数和隐式编码的维数均相等。

长短时记忆网络



- 在每个时刻t中只有内部记忆单元信息c_t和隐式编码h_t这两种信息起到了对序列信息进行传递的作用。每个时刻t只有h_t作为本时刻的输出以用于分类等处理。
- 以输入门$i_t = sigmoid(W_{xi}x_t + W_{hi}h_{t-1} + b_i)$为例，在$sigmoid$函数作用下，假设输入门$i_t$的取值为$(0.05,\ 0.1, 0.8)$，则通过$i_t \odot tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$中按位相乘$\odot$操作，输入门会“阻断”一些信息进行后续操作（如0.05和0.1所对应按位相乘的信息）、也会“放入”一些信息进行后续操作（如0.8所对应按位相乘的信息）。当然，如果门的某一位取值为0，表示这一位所对应信息为全闭、禁止信息通过；如果门的某一位取值为1，表示这一位所对应的信息为全开、容许信息全部通过。遗忘门不容许一些信息后续传递，从另外一个侧面而言是为了关注需要关注信息。



从$c_t = f_t \odot c_{t-1} + i_t \odot tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$可以看出，对当前时刻$t$所对应内部记忆单元中信息的调整涉及到遗忘门信息、$t-1$时刻所对应内部记忆单元中信息、输入门信息和$t-1$时刻的隐式编码。

由于$sigmoid$函数的值域在$(0,1)$之间，输入门、遗忘门和输出门三种门结构采用$sigmoid$这一非线性映射函数而起到了信息“控制门”的作用。在内部记忆单元和隐式编码中，使用了$tanh$这一非线性映射函数而没有继续使用$sigmoid$非线性映射函数，其原因在于$tanh$函数的值域为$(-1,1)$，使得$tanh$函数在进行信息整合时可起到信息“增（为正）”或“减（为负）”的效果。

LSTM如何克服梯度消失

对于$c_t = f_t \odot c_{t-1} + i_t \odot tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$，有如下求导结果的存在：

$$\frac{\partial c_t}{\partial c_{t-1}} = f_t + \frac{\partial f_t}{\partial c_{t-1}} \times c_{t-1} + \cdots\cdots$$

可见，$\frac{\partial c_t}{\partial c_{t-1}}$求导的结果至少大于等于$f_t$，即遗忘门的输出结果。如果遗忘门选择保留旧状态，则这一求导结果就接近1（或者接近向量1），使得梯度是存在的，从而避免了梯度消失问题。也就是说，LSTM通过引入门结构，在从t到$t+1$过程中引入加法来进行信息更新，避免了梯度消失问题。

整体来看，内部记忆单元信息c_t和隐式编码h_t这两种信息起到了对序列信息进行传递的作用。内部记忆单元信息c_t好比人脑的长时记忆、隐式编码h_t代表了短时记忆。

【填空题】内部记忆单元信息——长时记忆，隐式编码——短时记忆。

第六章 MOOC习题答案

1.在神经网络学习中，每个神经元会完成若干功能，下面哪个功能不是神经元所能够完成的功能（ C ）

A.对前序相邻神经元所传递信息进行加权累加

B.对加权累加信息进行非线性变化（通过激活函数）

C.向前序相邻神经元反馈加权累加信息

D.将加权累加信息向后续相邻神经元传递

2.下面对前馈神经网络描述不正确的是（ D ）

A.各个神经元接受前一级神经元的输入，并输出到下一级

B.同一层内的神经元相互不连接

C.层与层之间通过“全连接”进行连接，即两个相邻层之间神经元完全成对连接

D.同一层内神经元之间存在全连接

3.下面对感知机网络（Perceptron Networks）描述不正确的是（ C ）

A.感知机网络是一种特殊的前馈神经网络

B.感知机网络不能拟合复杂数据

C.感知机网络具有一层隐藏层

D.感知机网络没有隐藏层(正确说法，注意背诵)

4.下面对梯度下降方法描述不正确的是（ B ）

A.梯度下降算法是一种使得损失函数最小化的方法

B.梯度方向是函数值下降最快方向

C.梯度反方向是函数值下降最快方向

D.梯度下降算法用来优化深度学习模型的参数

5.下面对误差反向传播（error back propagation, BP)描述不正确的是（ B ）

A.对前馈神经网络而言，BP算法可调整相邻层神经元之间的连接权重大小

B.在BP算法中，每个神经元单元可包含不可偏导的映射函数

C.BP算法将误差从后向前传递，获得各层单元所产生误差，进而依据这个误差来让各层单元修正各单元参数

D.BP算法是一种将输出层误差反向传播给隐藏层进行参数更新的方法

6.我们可以将深度学习看成一种端到端的学习方法，这里的端到端指的是（ C ）

A.输入端-中间端

B.输出端-中间端

C.输入端-输出端

D.中间端-中间端

7.在前馈神经网络中，误差后向传播（BP算法）将误差从输出端向输入端进行传输的过程中，算法会调整前馈神经网络的什么参数（ B ）

A.同一层神经元之间的连接权重

B.相邻层神经元和神经元之间的连接权重

C.输入数据大小

D.神经元和神经元之间连接有无

8.前馈神经网络通过误差后向传播（BP算法）进行参数学习，这是一种（ D ）机器学习手段.

A.半监督学习

B.无监督学习

C.无监督学习和监督学习的结合

D.监督学习

9.下面对前馈神经网络这种深度学习方法描述不正确的是（ D ）

A.是一种监督学习的方法

B.是一种端到端学习的方法

C.实现了非线性映射

D.隐藏层数目大小对学习性能影响不大

10.下面对浅层学习和深度学习描述不正确的是（ C ）

A.线性回归分析是一种浅层学习方法

B.K-means聚类是一种浅层学习方法

C.浅层学习仅能实现线性映射、深度学习可以实现非线性映射

D.包含了若干隐藏层的前馈神经网络是一种深度学习方法

1.卷积操作是卷积神经网络所具备的一个重要功能，对一幅图像进行高斯卷积操作的作用是（ B ）

A.对图像进行分类

B.对图像进行平滑（模糊化）

C.对图像进行增强

D.对图像进行裁剪

2.对完成特定任务的卷积神经网络训练采用的是监督学习方法。在这个过程中，通过误差后向传播来优化调整网络参数，请问下面哪个参数不是通过误差后向传播来优化的（ D ）

A.卷积滤波矩阵中的参数

B.激活函数中的参数

C.全连接层的链接权重

D.模型的隐藏层数目

3.下面哪个作用是池化（pooling）层所完成的（ A ）

A.下采样

B.图像裁剪

C.上采样

D.图像增强

4.

单选(2分)

假设我们需要训练一个卷积神经网络，来完成500种概念的图像分类。该卷积神经网络最后一层是分类层，则最后一层输出向量的维数大小可能是（ C ）

A.1

B.100

C.500

D.300

5.下面对Word2Vec描述不正确的是（ D ）

A.可以利用某个单词来分别预测该单词的上下文单词，从而训练得到词向量模型

B.词向量是一种分布式的单词表达形式

C.可以根据某个单词的上下文单词来预测该单词，从而训练得到词向量模型

D.词向量是一种one-hot的单词表达形式

6.前馈神经网络和卷积神经网络的模型学习均是通过误差后向传播来优化模型参数，因此是一种监督学习方法。(对)

7.一般而言，在深度学习模型中，层数越多，其完成的非线性映射就越复杂，因此模型就具有更强的学习能力。（对）

8.与one-hot的单词表达不同，在词向量模型中，单词和单词之间的共现关系被忽略了。(错)

9.卷积神经网络一般包括卷积层、池化层、全连接层，一般激活函数在卷积层后进行使用。（对）

10.在使用卷积神经网络对图像分类中，一般将最后一层全连接层（即与输出层相连的全连接层）的输出作为每幅图像的特征表达。(对)

第七章 强化学习

一、强化学习定义

1.强化学习的基本属性

•基于评估：强化学习利用环境评估当前策略，以此为依据进行优化

•交互性：强化学习的数据在与环境的交互中产生

•序列决策过程：智能主体在与环境的交互中需要作出一系列的决策，这

些决策往往是前后关联的

	有监督学习	无监督学习	强化学习
学习依据	基于监督信息	基于对数据结构的假设	基于评估
数据来源	一次给定	一次给定	在时序交互中产生
决策过程	单步决策（如分类和识别等）	无	序贯决策（如棋类博弈）
学习目标	样本到语义标签的映射	数据的分布模式	选择能够获取最大收益的状态到动作的映射

2.离散马尔可夫过程——t+1时刻的状态仅与t时刻状态有关

- 一个随机过程实际上是一列随时间变化的随机变量，其中当时间是离散量时，一个随机过程可以表示为$\{X_t\}_{t=0,1,2,\cdots}$，其中每个X_t都是一个随机变量，这被称为离散随机过程
- 马尔可夫链（Markov Chain）：满足马尔可夫性（Markov Property）的离散随机过程，也被称为离散马尔科夫过程。

$$Pr(X_{t+1}=x_{t+1}|X_0=x_0,X_1=x_1,\cdots,X_t=x_t)=Pr(X_{t+1}=x_{t+1}|X_t=x_t)$$

$t+1$时刻状态仅与t时刻状态相关

3.马尔可夫奖励过程：引入奖励、折扣系数

为了在序列决策中对目标进行优化，在马尔可夫随机过程框架中加入了奖励机制：

- **奖励函数$R: S\times S\mapsto\mathbb{R}$，其中$R(S_t,S_{t+1})$描述了从第$t$步状态转移到第$t+1$步状态所获得奖励**
- **在一个序列决策过程中，不同状态之间的转移产生了一系列的奖励$(R_1,R_2,\cdots)$，其中R_{t+1}为$R(S_t,S_{t+1})$的简便记法。**
- **引入奖励机制，这样可以衡量任意序列的优劣，即对序列决策进行评价。**

问题：给定两个因为状态转移而产生的奖励序列$(1,1,0,0)$和$(0,0,1,1)$，哪个奖励序列更好？

为了比较不同的奖励序列，定义反馈（return），用来反映累加奖励：

$$G_t=R_{t+1}+\gamma R_{t+2}+\gamma^2R_{t+3}+\cdots$$

其中折扣系数（discount factor）$\gamma\in[0,1]$

反馈值反映了某个时刻后所得到累加奖励，当衰退系数小于1时，越是遥远的未来对累加反馈的贡献越少

假设$\gamma=0.99$

(1,1,0,0)：$G_0=1+0.99\times1+0.99^2\times0+0.99^3\times0=1.99$

(0,0,1,1)：$G_0=0+0.99\times0+0.99^2\times1+0.99^3\times1=1.9504$

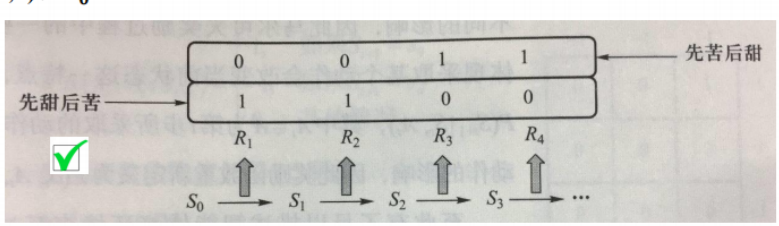


图7.3 两个奖励序列的比较

4.举例——机器人移动问题

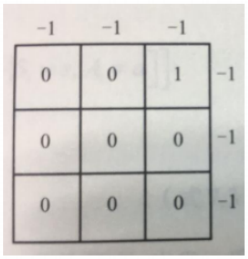
（1）若干定义

使用离散马尔可夫过程描述机器人移动问题

- 随机变量序列$\{S_t\}_{t=0,1,2,\cdots}$：S_t表示机器人第t步的位置，每个随机变量S_t的取值范围为$S=\{s_1,s_2,\cdots,s_9,s_d\}$
- 状态转移概率：$Pr(S_{t+1}|S_t)$满足马尔可夫性
- 定义奖励函数$R(S_t,S_{t+1})$：从S_t到S_{t+1}所获得奖励，其取值如图中所示
- 定义衰退系数：$\gamma\in[0,1]$

综合以上信息，可用$MRP=\{S,Pr,R,\gamma\}$来刻画马尔科夫奖励过程

图7.5 机器人寻路问题中的奖惩函数



这个模型不能体现机器人能动性，仍然缺乏与环境进行进行交互的手段

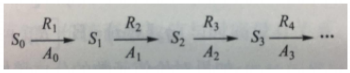
（2）动作价值的引入

在强化学习问题中，智能主体与环境交互过程中可自主决定所采取的动作，不同**动作**会对环境产生不同影响，为此：

- 定义智能主体能够采取的动作集合为A
- 由于不同的动作对环境造成的影响不同，因此状态转移概率定义为$Pr(S_{t+1}|S_t,a_t)$，其中$a_t\in A$为第t步采取的动作
- 奖励可能受动作的影响，因此修改奖励函数为$R(S_t,a_t,S_{t+1})$

- 动作集合A可以是有限的，也可以是无限的
- 状态转移可是确定（deterministic）的，也可以是随机概率性（stochastic）的。
- 确定状态转移相当于发生从S_t到S_{t+1}的转移概率为1

图7.6 智能体与环境的交互过程



（3）对动作、状态转移的一些说明

•动作集合A可以是有限的，也可以是无限的

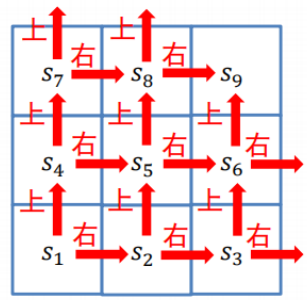
•状态转移可是确定（deterministic）的，也可以是随机概率性（stochastic）的。

•确定状态转移相当于发生从St到$St+1$的转移概率为1

(4) 较为完整的定义

使用离散马尔可夫过程描述机器人移动问题

- **随机变量序列$\{S_t\}_{t=0,1,2,\cdots}$：** S_t表示机器人第t步所在位置（即状态），每个随机变量S_t的取值范围为$S = \{s_1, s_2, \cdots, s_9, s_d\}$
- **动作集合：** $\mathbf{A} = \{上，右\}$
- **状态转移概率$Pr(S_{t+1}|S_t, a_t)$：** 满足马尔可夫性，其中$a_t \in \mathbf{A}$。状态转移如图所示。
- **奖励函数：** $\mathbf{R}(S_t, a_t, S_{t+1})$
- **衰退系数：** $\gamma \in [0, 1]$



综合以上信息，可通过$MDP = \{S, A, Pr, R, \gamma\}$来刻画马尔科夫决策过程

$S = \{s_1, s_2, \cdots, s_9, s_d\}$ **起始状态：** $S_0 = s_1$

$A = \{上，右\}$ **终止状态:** $S_T \in \{s_9, s_d\}$ $R(S_t, a_t, S_{t+1}) = \begin{cases} 1, 如果S_{t+1} = s_9 \\ -1, 如果S_{t+1} = s_d \\ 0, 其他情况 \end{cases}$

$\gamma = 0.99$

(6) 策略函数与策略学习

马尔可夫决策过程$MDP = \{S, A, Pr, R, \gamma\}$对环境进行了描述，那么

智能主体如何与环境交互而完成任务？需要进行策略学习

策略函数：

- 策略函数$\pi: S \times A \mapsto [0, 1]$，其中$\pi(s, a)$的值表示在状态$s$下采取动作$a$的概率。
- 策略函数的输出可以是确定的，即给定s情况下，只有一个动作a使得概率$\pi(s, a)$取值为1。对于确定的策略，记为$a = \pi(s)$。

(7) 价值函数与动作价值函数

如何进行策略学习：一个好的策略是在当前状态下采取了一个行动后，该行动能够在未来收到最大化的反馈：

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots$$

由马尔可夫性，未来的状态和奖励只与当前状态相关，与t无关。因此t取任意值该等式均成立，如**“逢山开路，遇水搭桥”**。

为了对策略函数π进行评估，定义

- **价值函数（Value Function）** $V: S \mapsto \mathbb{R}$，其中$V_\pi(s) = \mathbb{E}_\pi[G_t | S_t = s]$，即在第$t$步状态为$s$时，按照策略$\pi$行动后在未来所获得反馈值的期望
- **动作-价值函数(Action-Value Function)** $q: S \times A \mapsto \mathbb{R}$，其中$q_\pi(s,a) = \mathbb{E}_\pi[G_t | S_t = s, A_t = a]$表示在第$t$步状态为$s$时，按照策略$\pi$采取动作$a$后，在未来所获得反馈值的期望

这样，策略学习转换为如下优化问题：

寻找一个最优策略π^*，对任意$s \in S$使得$V_{\pi^*}(s)$值最大

智能体选择动作的模型：

策略函数$\pi: S \times A \mapsto [0,1]$，$\pi(s,a)$表示智能体在状态$s$下采取动作$a$的概率。

最大化每一时刻的回报值：

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots$$

一个好的策略函数应该能够使得智能体在采取了一系列行动后可获得最佳奖励

- **价值函数（value function）**：$V: S \mapsto \mathbb{R}$，其中$V_\pi(s) = \mathbb{E}_\pi[G_t | S_t = s]$，表示智能体在时刻$t$处于状态$s$时，按照策略$\pi$采取行动时所获得回报的期望。价值函数衡量了某个状态的好坏程度，反映了智能体从当前状态转移到该状态时能够为目标完成带来多大“好处”。
- **动作-价值函数(action-value function)**：$q: S \times A \mapsto \mathbb{R}$，其中$q_\pi(s,a) = \mathbb{E}_\pi[G_t | S_t = s, A_t = a]$，表示智能体在时刻$t$处于状态$s$时，选择了动作$a$后，在$t$时刻后根据策略$\pi$采取行动所获得回报的期望。

3.强化学习问题定义

学习最优策略，对于任何一个状态，它都能采取最佳动作，获得最大奖励。

定义： 给定一个马尔可夫决策过程$MDP=(S,A,P,R,\gamma)$，学习一个最优策略π^*，对任意$s\in S$使得$V_{\pi^*}(s)$值最大。

- 价值函数和动作-价值函数反映了智能体在某一策略下所对应状态序列获得回报的期望，它比回报本身更加准确地刻画了智能体的目标。

- 价值函数和动作-价值函数的定义之所以能够成立，离不开决策过程所具有的马尔可夫性，即当位于当前状态s时，无论当前时刻t的取值是多少，一个策略回报值的期望是一定的（当前状态只与前一状态有关，与时间无关）。

一个好的策略函数应该能够使得智能体在采取了一系列行动后可得到最佳奖励

4.贝尔曼方程

贝尔曼方程（Bellman Equation）也被称作动态规划方程（Dynamic Programming Equation），由理查德·贝尔曼（Richard Bellman）提出。

- **价值函数（Value Function）** $V_\pi(s)=\mathbb{E}_\pi\left[R_{t+1}+\gamma R_{t+2}+\gamma^2R_{t+3}+\cdots\middle|S_t=s\right]$
- **动作-价值函数(Action-Value Function)**

$$q_\pi(s,a)=\mathbb{E}_\pi\left[R_{t+1}+\gamma R_{t+2}+\gamma^2R_{t+3}+\cdots\middle|S_t=s,A_t=a\right]$$

$$\begin{aligned}
&V_\pi(s)=\mathbb{E}_\pi\left[R_{t+1}+\gamma R_{t+2}+\gamma^2R_{t+3}+\cdots\middle|S_t=s\right]\\
&=\mathbb{E}_{a\sim\pi(s,\cdot)}\left[\mathbb{E}_\pi\left[R_{t+1}+\gamma R_{t+2}+\gamma^2R_{t+3}+\cdots\middle|S_t=s,A_t=a\right]\right]\\
&=\sum_{a\in A}\underbrace{\pi(s,a)}_{\text{采取动作}a\text{的概率}}\times\underbrace{q_\pi(s,a)}_{\text{采取动作}a\text{后带来的回报期望}}\\
&=\sum_{a\in A}\pi(s,a)q_\pi(s,a)
\end{aligned}$$

- **价值函数（Value Function）** $V_\pi(s) = \mathbb{E}_\pi[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots | S_t = s]$
- **动作-价值函数(Action-Value Function)**

$$q_\pi(s,a) = \mathbb{E}_\pi[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots | S_t = s, A_t = a]$$

$$V_\pi(s) = \sum_{a\in A} \pi(s,a) \sum_{s'\in S} P(s'|s,a)\,[R(s,a,s') + \gamma V_\pi(s')]$$
$$= \mathbb{E}_{a\sim\pi(s,\cdot)} \mathbb{E}_{s'\sim P(\cdot|s,a)}[R(s,a,s') + \gamma V_\pi(s')]$$

价值函数取值与时间没有关系，只与策略π、在策略π下从某个状态转移到其后续状态所取得的回报以及在后续所得回报有关。

【理解】价值函数的数学期望，实际上有两项：一是采取动作立即带来的奖励的数学期望，另一个是折扣因此乘以下一状态的价值函数的数学期望。

价值函数的贝尔曼方程

$$V_\pi(s) = \mathbb{E}_{a\sim\pi(s,\cdot)}\, \mathbb{E}_{s'\sim P(\cdot|s,a)}[R(s,a,s') + \gamma V_\pi(s')]$$

动作-价值函数的贝尔曼方程

$$q_\pi(s,a) = \mathbb{E}_{s'\sim P(\cdot|s,a)}[R(s,a,s') + \gamma \mathbb{E}_{a'\sim\pi(s',\cdot)}[q_\pi(s',a')]]$$

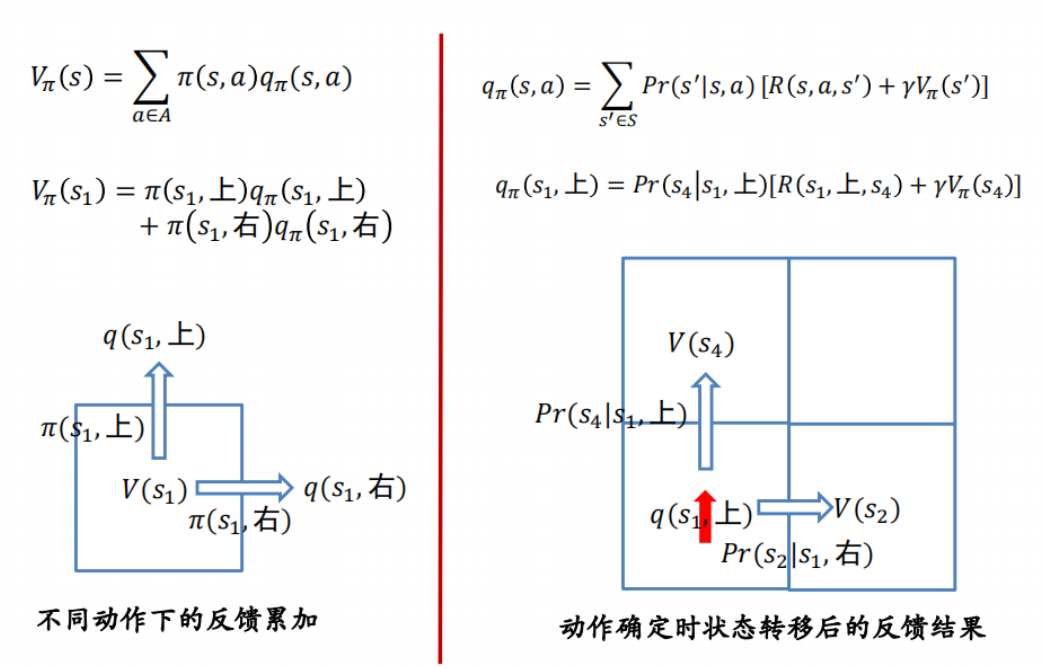
（1）贝尔曼方程描述了价值函数或动作-价值函数的递推关系，是研究强化学习问题的重要手段。其中价值函数的贝尔曼方程描述了当前状态价值函数和其后续状态价值函数之间的关系，即当前状态价值函数等于瞬时奖励的期望加上后续状态的（折扣）价值函数的期望。而动作-价值函数的贝尔曼方程描述了当前动作-价值函数和其后续动作-价值函数之间的关系，即当前状态下的动作-价值函数等于瞬时奖励的期望加上后续状态的（折扣）动作-价值函数的期望。

（2）在实际中，需要计算得到最优策略以指导智能体在当前状态如何选择一个可获得最大回报的动作。求解最优策略的一种方法就是去求解最优的价值函数或最优的动作-价值函数（即基于价值方法，value-based approach）。一旦找到了最优的价值函数或动作-价

值函数，自然而然也就是找到最优策略。当然，在强化学习中还有基于策略（policy-based）和基于模型（model-based）等不同方法。

5.【填空题】强化学习可以分为基于价值、基于策略或基于模型等不同方法。

6.价值函数、动作价值函数的计算——举例



7.【填空题】**通过策略计算价值函数的过程叫做策略评估（policy evaluation），通过价值函数优化策略的过程叫做策略优化（policy improvement），策略评估和策略优化交替进行的强化学习求解方法叫做通用策略迭代（Generalized Policy Iteration，GPI）。**

8.【填空题】强化学习中，常见的策略评估方法有：动态规划、时序差分、蒙特卡洛采样。

第七章 MOOC习题答案

1.下面对强化学习、监督学习和深度卷积神经网络学习的描述正确的是（ C ）

A.评估学习方式、端到端学习方式、端到端学习方式

B.无标注学习、有标注信息学习方式、端到端学习方式

C.评估学习方式、有标注信息学习方式、端到端学习方式

D.有标注信息学习方式、端到端学习方式、端到端学习方式

2.在强化学习中，通过哪两个步骤的迭代，来学习得到最佳策略（ B ）

A.动态规划与Q-Learning

B.策略优化与策略评估

C.Q-learning 与 贪心策略优化

D.价值函数计算与动作-价值函数计算

3.在强化学习中，哪个机制的引入使得强化学习具备了在利用与探索中寻求平衡的能力（ D ）

A.动态规划

B.Bellman方程

C.蒙特卡洛采样

D.贪心策略

4.深度强化学习是深度学习与强化学习的结合。在深度强化学习中，神经网络被用来进行哪个函数的学习？(D)

A.蒙特卡洛采样函数

B.贪心策略函数

C.Bellman方程

D.q函数

5.在本课程所讲述的范围内，“在状态s，选择一个动作，使得状态s得到最大的反馈期望”，这句话描述了状态s的（ A ）

A.策略学习与优化

B.动作-价值函数

C.采样函数

D.价值函数

6.在本课程内容范围内，“在状态s，按照某个策略行动后在未来所获得反馈值的期望”，这句话描述了状态s的（ A ）

A.价值函数

B.采样函数

C.动作-价值函数

D.策略优化

7.在本课程内容范围内，“在状态s，按照某个策略采取动作a后在未来所获得反馈值的期望”，这句话描述了状态s的（ C ）

A.策略优化

B.采样函数

C.动作-价值函数

D.价值函数

8.下面哪句话正确描述了马尔科夫链中定义的马尔可夫性（ C ）

A.t-1时刻状态取决于t+1时刻状态

B.t+2时刻状态取决于t时刻状态

C.t+1时刻状态取决于t时刻状态

D.t+1时刻状态和t时刻状态相互独立

9.下面对强化学习、有监督学习和无监督学习描述正确的是（ D ）

A.都需要标注数据

B.都不需要标注信息

C.是深度学习的学习算法

D.都是人工智能的学习算法

10.下面哪一种方法不是通过迭代计算贝尔曼方程来进行策略评估（ C ）

A.时序差分（Temporal Difference）

B.蒙特卡洛采样

C.深度学习

D.动态规划

第八章 博弈论

1.博弈的要素

- 参与者或玩家（player）：参与博弈的决策主体
- 策略（strategy）：参与者可以采取的行动方案，是一整套在采取行动之前就已经准备好的完整方案。
  - 某个参与者可采纳策略的全体组合形成了策略集（strategy set）
  - 所有参与者各自采取行动后形成的状态被称为局势（outcome）
  - 如果参与者可以通过一定概率分布来选择若干个不同的策略，这样的策略称为混合策略（mixed strategy）。若参与者每次行动都选择某个确定的策略，这样的策略称为纯策略（pure strategy）
- 收益（payoff）：各个参与者在不同局势下得到的利益
  - 混合策略意义下的收益应为期望收益（expected payoff）
- 规则（rule）：对参与者行动的先后顺序、参与者获得信息多少等内容的规定

2.**建模者对参与者（player）规定可采取的策略集(strategy sets)和取**

得的收益，观察当参与者选择若干策略以最大化其收益时会产生什么结果。两害相权取其轻，两利相权取其重。

3.【填空题】博弈分类：

①合作博弈、非合作博弈

②静态博弈、动态博弈

③完全信息博弈、非完全信息博弈

- 合作博弈与非合作博弈
  - 合作博弈（cooperative game）：部分参与者可以组成联盟以获得更大的收益
  - 非合作博弈（non-cooperative game）：参与者在决策中都彼此独立，不事先达成合作意向
- 静态博弈与动态博弈
  - 静态博弈（static game）：所有参与者同时决策，或参与者互相不知道对方的决策
  - 动态博弈（dynamic game）：参与者所采取行为的先后顺序由规则决定，且后行动者知道先行动者所采取的行为
- 完全信息博弈与不完全信息博弈
  - 完全信息（complete information）：所有参与者均了解其他参与者的策略集、收益等信息
  - 不完全信息（incomplete information）：并非所有参与者均掌握了所有信息

- 囚徒困境是一种非合作、不完全信息的静态博弈

4.纳什均衡

博弈的稳定局势即为纳什均衡（Nash equilibrium）：指的是参与者所作出的这样一种策略组合，在该策略组合上，任何参与者单独改变策略都不会得到好处。换句话说，如果在一个策略组合上，当所有其他人都不改变策略时，没

有人会改变自己的策略，则该策略组合就是一个纳什均衡。

•Nash定理：若参与者有限，每位参与者的策略集有限，收益函数为实值函数，则博弈必存在混合策略意义下的纳什均衡。

•囚徒困境中两人同时认罪就是这一问题的纳什均衡。

5.混合策略下的纳什均衡举例

- 例子：公司的雇主是否检查工作与雇员是否偷懒
- V是雇员的贡献，W是雇员的工资，H是雇员的付出，C是检查的成本，F是雇主发现雇员偷懒对雇员的惩罚（没收抵押金）。
- 假定$H < W < V$，$W > C$

- **参与者：**
  - 雇员、雇主
- **规则：**
  - 雇员与雇主两人分别决策，事先无法得知对方的选择
- **混合策略集：**
  - 雇员：偷懒、不偷懒
  - 雇主：检查、不检查
- **局势及对应收益**
  - 雇主采取检查策略时雇员工作与偷懒对应的结果
  - 雇主采取不检查策略时雇员工作偷懒对应的结果

表8.4 雇主-雇员每次采取对应行动后的收益

表8.4　雇主－雇员每次采取对应行动后的收益

		雇员	
		偷懒	不偷懒
雇主	检查	$-C+F, -F$	$V-W-C, W-H$
	不检查	$-W, W$	$V-W, W-H$

- V是雇员的贡献，W是雇员的工资，H是雇员的付出，C是检查的成本，F是雇主发现雇员偷懒而对雇员的惩罚（没收抵押金）。
- 假定$H < W < V$，$W > C$

若雇主检查的概率为α，
雇员偷懒的概率为β。

表8.4 雇主-雇员每次采取对应行动后的收益

表8.4　雇主－雇员每次采取对应行动后的收益

		雇员	
		偷懒	不偷懒
雇主	检查	$-C+F, -F$	$V-W-C, W-H$
	不检查	$-W, W$	$V-W, W-H$

表8.5 雇主-雇员之间博弈的期望收益

表8.5　雇主－雇员之间博弈的期望收益

参与者	采取策略	期望收益
雇主	检查	$T_1=\beta(-C+F)+(1-\beta)(V-W-C)$
	不检查	$T_2=-\beta W+(1-\beta)(V-W)$
雇员	偷懒	$T_3=-\alpha F+(1-\alpha)W$
	不偷懒	$T_4=\alpha(W-H)+(1-\alpha)(W-H)=W-H$

本问题中，达到纳什均衡时，雇主无论检查还是不检查，收益期望都一样；雇员无论偷懒还是不偷懒，期望收益都一样。据此，联立不同状态的期望收益，解出参数。

若雇主检查的概率为α，雇员偷懒的概率为β

表8.5 雇主-雇员之间博弈的期望收益

表8.5 雇主－雇员之间博弈的期望收益

参与者	采取策略	期望收益
雇主	检查	$T_1=\beta(-C+F)+(1-\beta)(V-W-C)$
	不检查	$T_2=-\beta W+(1-\beta)(V-W)$
雇员	偷懒	$T_3=-\alpha F+(1-\alpha)W$
	不偷懒	$T_4=\alpha(W-H)+(1-\alpha)(W-H)=W-H$

混合策略纳什均衡：博弈过程中，博弈方通过概率形式随机从可选策略中选择一个策略而达到的纳什均衡被称为混合策略纳什均衡。

- 纳什均衡：其他参与者策略不变的情况下，某个参与者单独采取其他策略都不会使得收益增加⇔无论雇主是否检查，雇主的收益都一样；无论雇员是否偷懒，雇员的收益都一样
  - 于是有$T_1=T_2$ 以及$T_3=T_4$
  - 在纳什均衡下，由于$T_3=T_4$，可知雇主采取检查策略的概率（雇主趋向于用这个概率去检查）：
    $$\alpha=\frac{H}{W+F}$$
  - 在纳什均衡下，由于$T_1=T_2$，可知雇员采取偷懒策略的概率（雇员趋向于用这个概率去偷懒）：
    $$\beta=\frac{C}{W+F}$$
- 在检查概率为α之下，雇主的收益：
  $$T_1=T_2=V-W-\frac{CV}{W+F}$$
  - 对上式中W求导，则当$W=\sqrt{CV}-F$时，雇主的收益最大，其值为$T_{max}=V-2\sqrt{CV}+F$

6.策梅洛定理：**策梅洛定理**（Zermelo's theorem）：**对于任意一个有限步的双人完全信息零和动态博弈，一定存在先手必胜策略或后手必胜策略或双方保平策略。**

证明：假设双人博弈过程结果只有胜、负、和三种结果，博弈过程最长持续N步，下面通过数学归纳法来如下证明策梅洛定理：

1)当$N = 1$时，先手玩家必选择自己收益最大的策略，若存在获胜的行动，则存在

先手必胜策略。若不存在使先手玩家获胜的行动，则先手玩家会选择平局的行动，于是存在双方保平策略。若仅存在后手玩家获胜的行动，则存在后手必胜策略。

2)假设当$N = 1,2 \ldots, k$时，均存在先手必胜策略、后手必胜策略或双方保平策略。

3)当$N = k + 1$时，将先手玩家行动一次后的局势视为开启了一个新的博弈。在这开启的新博弈中，先手玩家为原博弈的后手玩家，新博弈中的后手玩家为原博弈的先手，应用2)中的假设，新开启的博弈必存在先手必胜策略或后手必胜策略或双方保平策略，则原博弈亦必存在后手必胜策略或先手必胜策略或双方保平策略。

4)综上所述，对于任意一个有限步的双人完全信息零和动态博弈，一定存在先手必胜策略或后手必胜策略或双方保平策略。

提醒注意的是，策梅洛定理仅对两人博弈有效，如果博弈竞技者超过了2人，如对

于三人博弈，策梅洛定理无法保证三方中一定有一方获胜、其他两方必败或者三方和局的策略。

二、博弈规则设计

1.双边匹配问题

（1）背景

稳定婚姻问题（stable marriage problem）

就是一个典型的双边匹配问题，这个问题是指在给定成员偏好顺序的情况下，为两组成员寻找稳定的匹配。假设有n个单身男性构成的集合$M=\{m_1,m_2,\dots,m_n\}$，以及n个单身女性构成的集合$F=\{f_1,f_2,\dots,f_n\}$。对于任意一名单身男性m_i，都有自己爱慕的单身女性的顺序$s_{m_i}:=f_{m_{i,1}}\succ f_{m_{i,2}}\succ\cdots\succ f_{m_{i,n}}$，这里$f_{m_{i,j}}$表示第$i$名男性所喜欢单身女性中排在第$j$位的单身女性，同理对于任意一名单身女性$f_i$也有其爱慕的单身男性顺序$s_{f_i}:=m_{f_{i,1}}\succ m_{f_{i,2}}\succ\cdots\succ m_{f_{i,n}}$，$m_{f_{i,j}}$表示第$i$名女性所喜欢单身男性中排在第$j$位的单身男性。算法的最终目标是为这$2n$个男士和女士匹配得到$n$对伴侣，每一对伴侣可以表示为$(m_i,f_j)$。

（2）已知条件

表8.9 稳定婚姻匹配偏好序列

男性	偏好	女性	偏好
1	$A\succ D\succ C\succ B$	A	$3\succ 4\succ 2\succ 1$
2	$A\succ B\succ C\succ D$	B	$3\succ 2\succ 4\succ 1$
3	$A\succ C\succ D\succ B$	C	$1\succ 3\succ 4\succ 2$
4	$B\succ A\succ D\succ C$	D	$2\succ 4\succ 3\succ 1$

（3）修补策略与其漏洞

表8.10 修补策略下的稳定婚姻匹配过程

匹配1	修补	匹配2	修补	匹配3	修补	匹配4	修补	匹配5
(1, *A*)	(2, *A*)	(1, *B*)	(4, *A*)	(1, *B*)	(4, *B*)	(1, *A*)	(2, *B*)	(1, *A*)
(2, *B*)		(2, *A*)		(2, *D*)		(2, *D*)		(2, *B*)
(3, *C*)		(3, *C*)		(3, *C*)		(3, *C*)		(3, *C*)
(4, *D*)		(4, *D*)		(4, *A*)		(4, *B*)		(4, *D*)

如图所示，可以发现随机初始化匹配序列，再进行修补，可能陷入无限循环而得不到稳定的匹配。

（4）G-S算法

1962年，美国数学家大卫·盖尔和博弈论学家沙普利提出了针对双边稳定匹配问题的解决算法（也被称为Gale- Shapely算法或G-S算法），并将其应用于稳定婚姻问题的求解[Gale 1962]，算法过程如下：

1) 单身男性向最喜欢的女性表白
2) 所有收到表白的女性从向其表白男性中选择最喜欢的男性，暂时匹配
3) 未匹配的男性继续向没有拒绝过他的女性表白。收到表白的女性如果没有完成匹配，则从这一批表白者中选择最喜欢男性。即使收到表白的女性已经完成匹配，但是如果她认为有她更喜欢的男性，则可以拒绝之前的匹配者，重新匹配
4) 如此循环迭代，直到所有人都成功匹配为止

表8.11 G-S算法下的稳定婚姻匹配过程

表8.11 G-S算法下的稳定婚姻匹配过程

第一轮		第二轮		第三轮		第四轮		第五轮	
表白	选择	表白	选择	表白	选择	表白	选择	表白	选择
(1, *A*)	\	(1, *D*)	(1, *D*)	\	(1, *D*)	\	\	(1, *C*)	(1, *C*)
(2, *A*)	\	(2, *B*)	(2, *B*)	\	(2, *B*)	\	(2, *B*)	\	(2, *B*)
(3, *A*)	(3, *A*)	\	(3, *A*)	\	(3, *A*)	\	(3, *A*)	\	(3, *A*)
(4, *B*)	(4, *B*)	\	\	(4, *A*)	\	(4, *D*)	(4, *D*)	\	(4, *D*)

在第一轮中，4名男性分别向自己最喜欢的女性告白，而收到3人告白的女性*A*选择了自己最喜欢的男性3，另一个收到告白的女性*B*选择了男性4；在第二轮中，尚未匹配的男性1和男性2继续向自己第二喜欢的对象告白，收到告白的女性*B*选择了自己更喜欢的男性2而放弃了男性4，同理继续三轮告白和选择，所有人都找到了自己的伴侣，且所有匹配都是稳定的。可以看出，使用G-S算法得到了稳定匹配的结果。

2.单边匹配问题

（1）简介：在匹配问题中，除了需要双向选择的双边匹配问题，还有一种类似于以物易物方式的交换匹配问题，被称为**单边匹配问题**，例如室友的匹配或者是座位的分配。这些问题中分配的对象都是不可分的标的物，他们只能属于一个所有者，且可以属于任何一个所有者。

（2）最大交易圈算法

①初始化：随机分配物品。

②每个交易者发出一条边，指向他最喜欢的标的物；每个标的物发出一条边，指向它的当前拥有者。

③有向图中必定存在环->标的物赋予最希望拥有它的人，然后得到最想要的标的物的人放弃原先自己占有的标的物，带着新标的物离场。

④迭代，直到无法形成交易圈。

对于单边匹配问题，1974年，沙普利和斯卡夫提出了针对单边匹配问题的稳定匹配算法“最大交易圈算法（Top-trading cycle，TTC）”[Shapley 1974]，算法过程如下：

1) 首先记录每个标的物的初始占有者，或者对物品进行随机分配。
2) 每个交易者连接一条指向他最喜欢的标的物的边，并从每一个标的物连接到其占有者或是具有高优先权的交易者。
3) 此时形成一张有向图，且必存在环，这种环被称为“交易圈”，对于交易圈中的交易者，将每人指向节点所代表的标的物赋予交易者，同时交易者放弃原先占有的标的物，占有者和匹配成功的标的物离开匹配市场。
4) 接着从剩余的交易者和标的物之间重复进行交易圈匹配，直到无法形成交易圈，算法停止。

图8.6 第一轮单边匹配

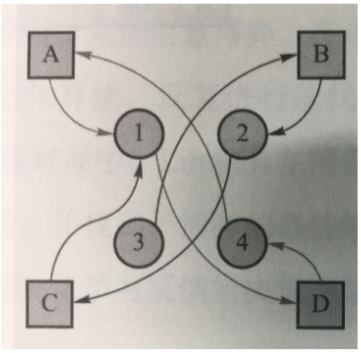
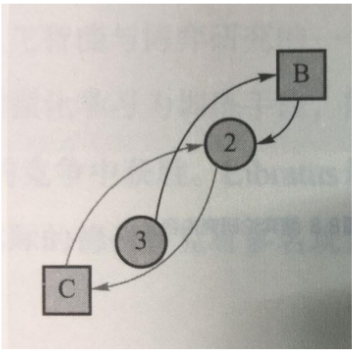


图8.7 第二轮单边匹配



- **在图8.6可以看出:，A和D之间构成一个交易圈，可达成交易，所以A得到床位1，D得到床位4，之后将A和D以及1和4从匹配图中移除。**
- **从图8.7可以看出，B和C都希望得到床位2，无法再构成交易圈，但是由于C是床位的本身拥有者，所以C仍然得到床位2，B只能选择床位3 。**

【理解】①标的物发出的箭头，指向的是它当前的持有者。随机初始化，ABCD占有的床位分别是4321.

②A最想要1床位，D最想要4床位，那么A发出一条边，指向1,D发出一条边，指向4.

③A->1->D->4形成交易圈，A与D互换床位，之后他们都得到了自己最想要的床位，就可以离场了。

④B和C都想得到床位2，C持有床位2，所以无法形成交易圈。因此算法终止。

图8.6 第一轮单边匹配

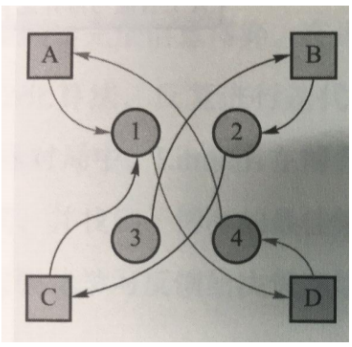
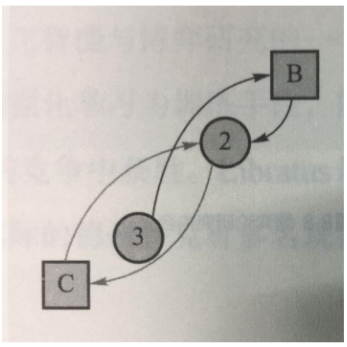


图8.7 第二轮单边匹配



- 在图8.6可以看出:，A和D之间构成一个交易圈，可达成交易，所以A得到床位1，D得到床位4，之后将A和D以及1和4从匹配图中移除。
- 从图8.7可以看出，B和C都希望得到床位2，无法再构成交易圈，但是由于C是床位的本身拥有者，所以C仍然得到床位2，B只能选择床位3 。

最后交易结果A→1，B →3，C →2，D →4

第八章 MOOC习题答案

1.标志着现代博弈理论的初步形成的事件是（ D ）

A.纳什均衡思想的提出

B.冯·诺伊曼计算机的实现

C.囚徒困境思想的提出

D.1944年冯·诺伊曼与奥斯卡·摩根斯特恩合著《博弈论与经济行为》的出版

2.下面对博弈研究分类不正确的是（ A ）

A.囚徒困境与纳什均衡

B.合作博弈与非合作博弈

C.完全信息博弈与不完全信息博弈

D.静态博弈与动态博弈

3.囚徒困境是一种（ A ）

A.纳什均衡

B.合作博弈

C.动态博弈

D.最优解

4.下面对纳什均衡描述正确的是（ C ）

A.参与者所作出的这样一种策略组合，在该策略组合上，有且只有1个参与者改变策略后，其收益会增加。

B.参与者所作出的这样一种策略组合，在该策略组合上，有且只有1个参与者改变策略后，其不会得到好处。

C.参与者所作出的这样一种策略组合，在该策略组合上，任何参与者单独改变策略都不会得到好处。

D.在一个策略组合上，当所有其他人都改变策略时，也无法破坏先前的博弈平衡，则该策略组合就是一个纳什均衡。

5.下面对混合策略纳什均衡描述正确的是（ B ）

A.博弈过程中，博弈方以概率形式随机从可选收益中选择一个收益，而达到的纳什均衡被称为混合策略纳什均衡。

B.博弈过程中，博弈方通过概率形式随机从可选策略中选择一个策略而达到的纳什均衡被称为混合策略纳什均衡。

C.博弈过程中，博弈方通过非概率形式随机从可选策略中选择一个策略而达到的纳什均衡被称为混合策略纳什均衡。

D.博弈过程中，博弈方以非概率形式随机从可选收益中选择一个收益，而达到的纳什均衡被称为混合策略纳什均衡。

6.ε-纳什均衡与纳什均衡的关系是（ D ）

A.ε-纳什均衡是纳什均衡

B.纳什均衡和ε-纳什均衡均是最优解

C.纳什均衡不是ε-纳什均衡

D.纳什均衡是ε-纳什均衡

7.在遗憾最小化算法中，玩家i按照如下方法来计算其在每一轮产生的悔恨值

（ B ）

A.至少改变1个以上玩家的策略， 所产生的收益之差。

B.其他玩家策略不变，只改变玩家i的策略后，所产生的收益之差。

C.每个玩家策略不变，只改变收益函数，所产生的收益之差。

D.所有玩家策略均改变，所产生的收益之差。

8.下面对生成对抗网络（Generative Adversarial Network）描述不正确的是

（ C ）

A.GAN包含生成网络和判别网络两个网络。

B.生成网络和判别网络分别依次迭代优化。

C.GAN是一种区别学习模型。

D.GAN是一种生成学习模型。

9.在生成对抗网络（Generative Adversarial Network）中，生成网络和判别网络的功能分别是（ C ）

A.区别真实数据与合成的虚假数据/学习真实数据的分布

B.学习真实数据的标签分布/区别真实数据与合成的虚假数据

C.学习真实数据的分布/区别真实数据与合成的虚假数据

D.区别真实数据与合成的虚假数据/学习真实数据的标签分布

10.最佳反应策略和纳什均衡之间的关系是（ C ）

A.如果每个玩家的策略相对于其他玩家的策略中至少有一个是最佳反应策略，那么策略组就是一个纳什均衡策略。

B.每个玩家的策略相对于其他玩家的策略而言不少于一个是最佳反应策略，那么策略组就是一个纳什均衡策略。

C.如果每个玩家的策略相对于其他玩家的策略而言都是最佳反应策略，那么策略组就是一个纳什均衡策略。

D.至少一个玩家的策略相对于其他玩家的策略而言都是最佳反应策略，那么策略组就是一个纳什均衡策略。

第九章 人工智能未来发展和趋势

【注】本章没有MOOC测验题。

1.【回顾】人工智能的手段与方法

①以符号主义为核心的逻辑推理

②以问题求解为核心的探寻搜索

③以数据驱动为核心的机器学习

④以行为主义为核心的强化学习

⑤以博弈对抗为核心的群体智能

手段与方法	特点
以符号主义为核心的逻辑推理	将概念（如命题等）符号化，从若干判断（前提）出发得到新判断（结论）
以问题求解为核心的探寻搜索	探寻搜索依据已有信息来寻找满足约束条件的待求解问题的答案
以数据驱动为核心的机器学习	从数据中发现数据所承载语义（如概念）的内在模式
以行为主义为核心的强化学习	根据环境所提供的奖罚反馈来学习所处状态可施加的最佳行动，在“探索（未知空间）-利用（已有经验）（exploration vs. exploitation）”之间寻找平衡，完成某个序列化任务，具备自我学习能力
以博弈对抗为核心的群体智能（两人及以上）	从“数据拟合”优化解的求取向“均衡解”的求取迈进

2.【了解】人工智能未来发展与趋势