人工智能基础复习

1. 知识表达与推理

1.1 命题逻辑

- 假言推理(Modus Ponens)：$\alpha \to \beta,\ \alpha \Rightarrow \beta$
- 与消解(And-Elimination)：$\alpha_1 \wedge \alpha_2 \wedge \cdots \wedge \alpha_n \Rightarrow \alpha_1,\ \alpha_2,\ \cdots,\ \alpha_n$
- 与导入(And-Introduction)：$\alpha_1,\ \alpha_2,\ \cdots,\ \alpha_n \Rightarrow \alpha_1 \wedge \alpha_2 \wedge \cdots \wedge \alpha_n$
- 双重否定(Double-Negation Elimination)：$\neg\neg\alpha \Rightarrow \alpha$
- 单项消解或单项归结(Unit Resolution)：$\alpha \vee \beta,\ \neg\beta \Rightarrow \alpha$
- 消解或归结(Resolution)：$\alpha \vee \beta,\ \neg\beta \vee \gamma \Rightarrow \alpha \vee \gamma$

1.2 谓词逻辑

- 全称量词消去(universal instantiation, UI)：$(\forall x)A(x) \Rightarrow A(y)$
- 全称量词引入(universal generalization, UG)：$A(y) \Rightarrow (\forall x)A(x)$
- 存在量词消去(existential instantiation, EI): $(\exists x)A(x) \Rightarrow A(c)$
- 存在量词引入(existential generalization, EG): $A(c) \Rightarrow (\exists x)A(x)$

1.3 知识图谱

- 知识图谱：由有向图构成，被用来描述现实世界中实体与实体的关系
- 三元组形式<left_node,relation,right_node>
- 一阶逻辑形式 Relation(left_node,right_node)，其中Relation是一阶谓词
- 归纳：从数据到知识
- 演绎：从知识到数据
- 归纳逻辑程序设计（inductive logic programming，ILP)
- FOIL（First Order Inductive Learner）通过序贯覆盖学习推理
  - 目标谓词：需要推断规则的结论，也被称为规则头
  - 算法目标，学习得到是得目标谓词成立的前提约束谓词。

$$(\forall x)(\forall y)(\forall z)\big(Mother(z,\ y) \wedge Couple(x, z) \rightarrow Father(x,\ y)\big)$$

前提约束谓词
（学习得到）

目标谓词
（已知）

  - 训练样例的构造
    - 正例集合 E^+ ，使得目标谓词为真的一阶逻辑

        - 反例集合 E^- ，使得目标谓词为假的一阶逻辑，一般不会显式给出，需要进行构造，构造过程中的实体关系必须存在且确定与目标谓词相悖。
        - 背景知识：知识图谱中目标谓词以外的其他谓词实例化的结果
    - 添加谓词后的质量由信息增益值（information gain）进行判断，公式为
    $$FOILGain = \hat{m_+}(log_2 \frac{\hat{m_+}}{\hat{m_+} + \hat{m_-}} - log_2 \frac{m_+}{m_+ + m_-})$$
    - 如果规则中正例为0，信息增益值记为NA
    - 选择信息增益最大的规则进行添加，并将训练样例集合中与该推理规则不符（新规则下无法推断）的样例去除
    - PS：在训练过程中，规则要使用数据实例进行推导
    - 不包含反例时，学习结束
- 路径排序推理算法
    - 特征抽取：生成并选择路径特征集合，生成路径的方式：随机游走、广度优先搜索，深度优先搜索。
    - 特征计算：计算每个训练样例的特征值，表示从实体结点s出发通过关系路径 π_j 到结点t的概率；也可以表示为布尔值，是否存在路径；也可以表示为频数、频率等。
    - 分类器训练：根据训练样例的特征值，为目标关系训练分类器，当训练好分类器后，将分类器用于推理两个实体是否存在目标关系。

1.4 概率图谱推理

1. 概率图

- 概率图：用概率描述两个相连节点之间的关联
- 概率推理：基于概率图进行的推理
- 贝叶斯网络：用一个有向无环图来表示，有向边表示节点和节点之间的单向依赖。
- 马尔可夫网络：无向图的网络结构，使用无向边来表示节点和节点之间的概率依赖。

2. 贝叶斯网络

- 满足局部马尔可夫性：在给定父节点的情况下，该父节点有条件地独立于它的非后代节点。
- 主要会计算联合概率和条件概率

3. 马尔可夫逻辑网络

- 从概率统计角度：简明描述了MNs中所存在信息之间的关联；在马尔可夫网络中引入谓词逻辑，融入结构化知识。
- 从一阶谓词逻辑角度：在谓词逻辑中添加不确定性，对严格推理进行松绑，更好反映了客观世界的复杂性。

- 给定一个由若干规则构成的集合，集合中每条推理规则赋予一定权重，则可以计算某个断言成立的概率。

$$P(X=x)=\frac{1}{Z}\exp\left(\sum_i w_i n_i(x)\right)=\frac{1}{Z}\prod_i \phi_i\left(x_{\{i\}}\right)^{n_i(x)}$$

其中$n_i(x)$是在推导x中所涉及第i条规则的逻辑取值（为1或0）、w_i是该规则对应的权重，Z是一个固定的常量，可由下式计算：

$$Z=\sum_{x\in X}\exp\left(\sum_i w_i n_i(x)\right)$$

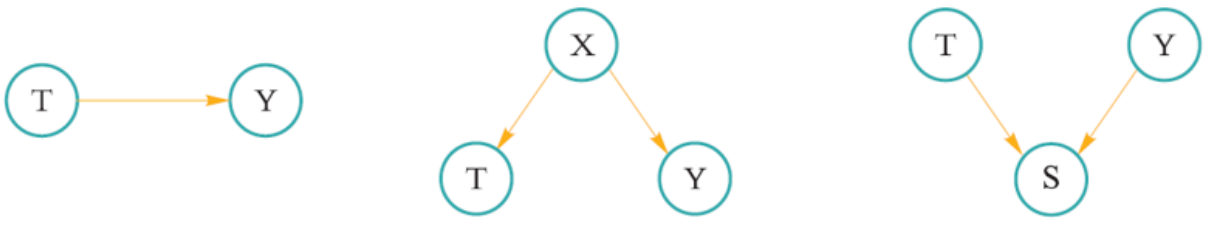
1.5 因果推理

1. 辛普森悖论

- 忽略潜在的”第三个变量“（混淆因素）可能会改变已有的结论

2. 克服辛普森悖论：厘清真/假关联

- 因果关联：数据中两个变量，一个变量是另一个变量的原因
- 混淆关联：数据中待研究的两个变量之间存在共同的原因变量；当忽略原因变量时，两个变量存在虚假的关联，即混淆关联。
- 选择关联：数据中待研究的两个变量存在共同的结果变量，当基于特定的结果变量取值进行研究，两个变量就会存在虚假的关联，即选择关联。



(a) 因果关联　　(b) 混淆偏差　　(c) 选择偏差

图 2.5　关联关系的三种来源

3. 因果分析的两种框架

- 潜在结果框架

- 结构因果模型
  - 因果图：有向无环图，刻画了图中所有节点的依赖关系。
  - DAG用于描述变量联合分布和数据生成机制的模型，被称为”贝叶斯网络“
  - 对任意的DAG，有模型中d个变量的联合概率由每个节点与其父节点之间的条件概率乘积给出

$$P(x_1, x_2, \cdots, x_d) = \prod_{j=1}^{d} P(x_j | x_{pa(j)})$$

- 因果干预：改变明确存在关联关系的某变量取值，研究变量取值改变对结果变量的影响。
- ”do“算子：计算当系统中一个变量取值发生变化，其他变量保存不变时，系统输出结果是否变化，判断变量是否是在系统中起决定作用的”原因要素“
- 因果效应：$P(Y = y|do(X = x))$
- 因果效应差：比较某一变量与否对结果的影响，也被称为平均因果效应（ACE)。
  $P(Y = y|do(X = 1)) - P(Y = y|do(X = 0))$
- 操作图模型：指定干预变量后，去除指向该变量的边得到的图
- 操纵概率
  - 边缘概率不随干预而改变
  - 条件概率不变
  - 调整公式：对于每个Z的取值，计算条件概率并取平均，也被称为Z调整或Z控制

4. 反事实推理

2. 搜索

2.1 搜索基本概念

a. 搜索的形式化描述

- 状态：对搜索算法和搜索环境当前所处情形的描述信息。
- 动作：算法从一个状态转移到另外一个状态所采取的行动。
- 状态转移：选择动作后，状态也发生改变
- 路径和代价：搜索算法通过执行一系列动作后，得到一个状态序列，被称为路径。每条路径对应一个代价。状态序列长度为2，这条路径的代价也被称为单步代价。（一般为非负）
- 目标测试：用于判断状态s是否为目标状态。

b. 搜索算法的评测标准

- 完备性：当问题存在解时，算法是否能保证找到一个解
- 最优性：搜索算法能否保证找到的第一个解是最优解
- 时间复杂度：找到一个搜索路径所需时间
- 空间复杂度：算法运行所需的内存空间

符号	含义
b	分支因子，即搜索树中每个结点最大的分支数目
d	根结点到最浅的目标结点的路径长度
m	搜索树中路径的最大可能长度
n	状态空间中状态的数量

c. 搜索框架

- 边缘集合：搜索树中可用于下一步探索的所有候选结点集合
- 初始化边缘集合，（选择并去除结点，判断目标测试，扩展边缘集合）对n进行扩展
- 对非目标结点的扩展：将非目标结点的后继结点加入边缘集合。

d. 剪枝策略

- 剪枝：放弃扩展部分结点的做法。
- 剪枝对应的搜索算法被称为剪枝搜索

2.2 贪婪最佳优先搜索与A*搜索算法

a. **有信息搜索/启发式搜索**

- 定义：利用可帮助决策的辅助信息的搜索算法。
- 评价函数：根据评价函数进行扩展结点的选择。（在整个边缘集合内）
- 启发函数：估计结点n到目标结点所形成路径的最小路径值

b. **贪婪最佳优先搜索**

- 评价函数=启发函数
- 如果无环路则具有完备性，否则不具备；不具有最优性
- 最坏情况下，时间复杂度与空间复杂度均为 $O(b^m)$

c. **A*搜索算法**

- 路径代价函数g(n)：起始结点到结点n的路径代价

- 评价函数：h(n)启发函数+g(n)
- A*的完备性和最优性取决于搜索问题和启发函数的性质

符号	含义
$h(n)$	结点 n 的启发函数取值
$g(n)$	从起始结点到结点 n 所对应路径的代价
$f(n)$	结点 n 的评价函数取值
$c(n, a, n')$	从结点 n 执行动作 a 到达结点 n' 的单步代价
$h^*(n)$	从结点 n 出发到达终止结点的最小代价

  - 可容性：$h(n) \leq h^*(n)$
  - 一致性：$h(n) \leq c(n, a, n') + h(n')$
  - 一致性必然导致可容性
- A*算法完备性条件：
  - 搜索树中分支数量有限；
  - 单步代价的下界是一个正数；
  - 启发函数有下界
- A*算法最优性条件：启发函数具有可容性

2.3 MiniMax搜索与alpha-beta剪枝

深度优先搜索算法

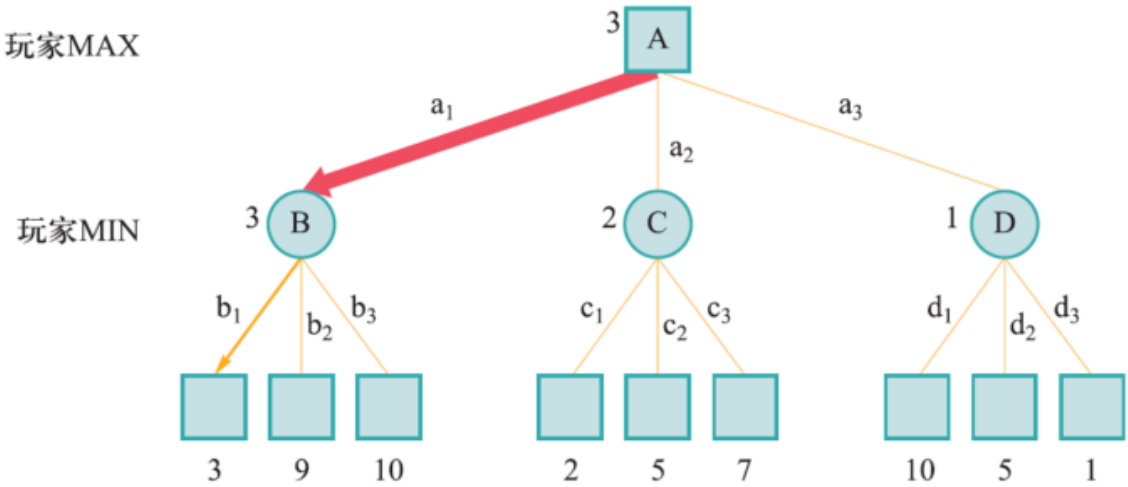
a. 对抗搜索/博弈搜索

两个智能体同处于一个竞争环境，通过竞争来实现各自相反的目标。一方最大化自身利益，一方最小化自身利益。

b. MiniMax搜索

- 状态：s包括当前的游戏局面和当前行动的智能体
- 动作：给定s，玩家(s)在当前局面下可以采取的操作a，动作集合为actions(s)
- 状态转移：result(s,a \belong actions(s))
- 终局状态检测：是否在状态s结束
- 终局得分：零和博弈下，只要记录一个玩家的分数

- Max选择让分数最大的动作，Min选择让分数最小的动作



- MiniMax对搜索树执行了完整的深度优先搜索；
  - 在每个结点进行最优策略的递归计算，并返回最优值

c. alpha-beta剪枝搜索

- 保证得到与原MiniMax算法同样的搜索结果情况下，剪去不影响最终结果的搜索分支
- 做法：设置根结点的alpha和beta值，为-无穷大和无穷大，孩子结点继承父节点的alpha beta值，对于非叶子Min结点利用子节点更新beta值，对于非叶子Max结点更新其alpha值（如果子节点大于alpha，则用子节点的值）

2.4 蒙特卡洛树搜索

a. 多臂赌博机

- 状态：每个被摇动的臂膀即为一个状态
- 动作：对应摇动一个赌博机的臂膀
- 奖励：在第t次所得的收益分数
- 悔值函数：（最优策略得分-实际得分）

$$\rho_T = T\mu^* - \sum_{t=1}^{T} \hat{r}_t$$

- 探索和利用之间的对立关系

b. $\epsilon-$ 贪心算法

- 平衡探索与利用

$$l_t = \begin{cases} \text{argmax}_i \, \bar{x}_{i,T_{(i,t-1)}}, & \text{以} 1-\epsilon \text{的概率} \\ \text{随机的} i \in \{1,2,\cdots,K\}, & \text{以} \epsilon \text{的概率} \end{cases}$$

c. 上限置信区间算法(UCB1)

- 思路：优先探索估计值不确定度高的动作，如果奖励估计值极端小则也不探索。
- 策略：计算每个动作奖励期望的估计范围，以范围上限的高低来选择动作。
- 计算范围上界：将选择动作 a_i 的 $T_{(i,T-1)}$ 个样本视为独立同分布的样本，根据霍夫丁不等式可以得到：（t为总次数，T_{(i,T-1)}为选择该动作的次数（选择多次后迅速收敛）

$$\bar{x}_{i,T_{(i,t-1)}} + C\sqrt{\frac{2\ln t}{T_{(i,t-1)}}}$$

d. 蒙特卡洛搜索树（如何高效的扩展搜索树）

- 选择：用UCB1算法实现子节点的选择，直到叶子节点/未被完全扩展的结点（并记录结点选择次数和奖励均值）
- 扩展：如果叶子节点未终止，则随机扩展一个未被扩展过的后继结点
- 模拟：从扩展的结点出发，模拟扩展搜索树，直到找到一个终止结点
- 反向传播：将模拟结果回溯更新M及M以上结点的访问均值和被访问次数。
- 以MiniMax算法为基础的蒙特卡洛搜索算法也被称为上限置信区间树搜索(UCT)。
- 其中为了能统一使用UCB1求解，对于Max节点的值未现有分数减去终局得分

3. 机器学习

3.1 机器学习基本概念

a. 从数据利用的角度，分为监督学习、无监督学习和半监督学习等。

- 监督学习：从假设空间学习得到一个最优映射函数（决策函数），将输入函数映射到语义标注空间。
- 无监督学习：从无标签数据中学习映射函数

b. 基础概念

- NFL定理：离开具体场景和问题讨论机器学习无意义。模型平均意义性能一样。

- 泛化能力：模型在训练集上所得性能与测试集上保持一致的能力
- 损失函数：估量预测值和真实值之间的差异
- 经验风险：映射函数f在训练集上产生的损失
- 期望风险：映射函数f在所有数据产生的损失，也被称为真实风险或真实误差

$$\int_{x\times y} Loss(y, f(x))P(x,y)dxdy$$

- 机器学习中模型优化目标一般为经验风险最小化，追求期望风险最小化
- 过学习：模型复杂后虽然经验风险变小但是err变大，导致期望误差反而增加。

$$\underbrace{\Re}_{\text{期望风险}} \leq \underbrace{\Re_{emp}}_{\text{经验风险}} + err$$

- 结构风险最小化：引入正则化项或惩罚项降低模型复杂度，在最小化经验风险和降低模型复杂度两方向寻求平衡。

c. 模型度量方法

- 准确率、错误率、精确率（查准率）、召回率（查全率）

d. 参数优化

- 频率学派：最大似然估计
- 贝叶斯学派：似然概率与先验概率的乘积最大，即最大后验估计

3.2 回归分析与决策树

a. 线性回归模型

- 最佳回归模型：使得残差平方和的平均值最小（mse）
- 回归方法：最小二乘法、梯度下降法
- $(XX^T)^{-1}Xy$
- 问题：对离群值非常敏感，特别是二分类问题
- 引入sigmoid函数，提出Logistic regression
- $y = \frac{1}{1+e^{-z}}$ 其中 $z = w^T x + b$
- 如果引入softmax函数可推广为多项逻辑斯蒂回归，用以解决多分类问题。

b. 决策树

- 将分类问题分解为若干基于单个信息的推理任务，采用树状结构逐步完成推理。
- 信息熵：衡量样本集合纯度的一种指标，信息熵越小，纯度越大。

$$E(D) = -\sum_{k=1}^{K} p_k \log_2 p_k$$

- 信息增益：选择属性划分样本集前后信息熵的减少量称为信息增益。

$$Gain(D, A) = Ent(D) - \sum_{i=1}^{n} \frac{|D_i|}{|D|} Ent(D_i)$$

3.3 K均值聚类

- 使得簇内方差最小化。
- 找到一个局部最优，即没有其他的聚类结果能够让簇内的方差最小，无法保证找到全局最优。
- 易受初始值影响的迭代算法
- 做法：
  - 初始化聚类中心
  - 聚类
  - 计算新的聚类中心
  - 迭代
- 另一种解释：KMeans聚类通过最小化聚簇内的数据方差来实现最大化类内相似度

3.4 监督学习与非监督学习下的特征降维

a. LDA（Fisher 线性判别分析）

- 具有标签信息的高维数据样本，LDA利用类别信息进行线性投影到一个低维空间中，使其在低维空间中同一类样本尽可能靠近，不同类样本尽可能远离。

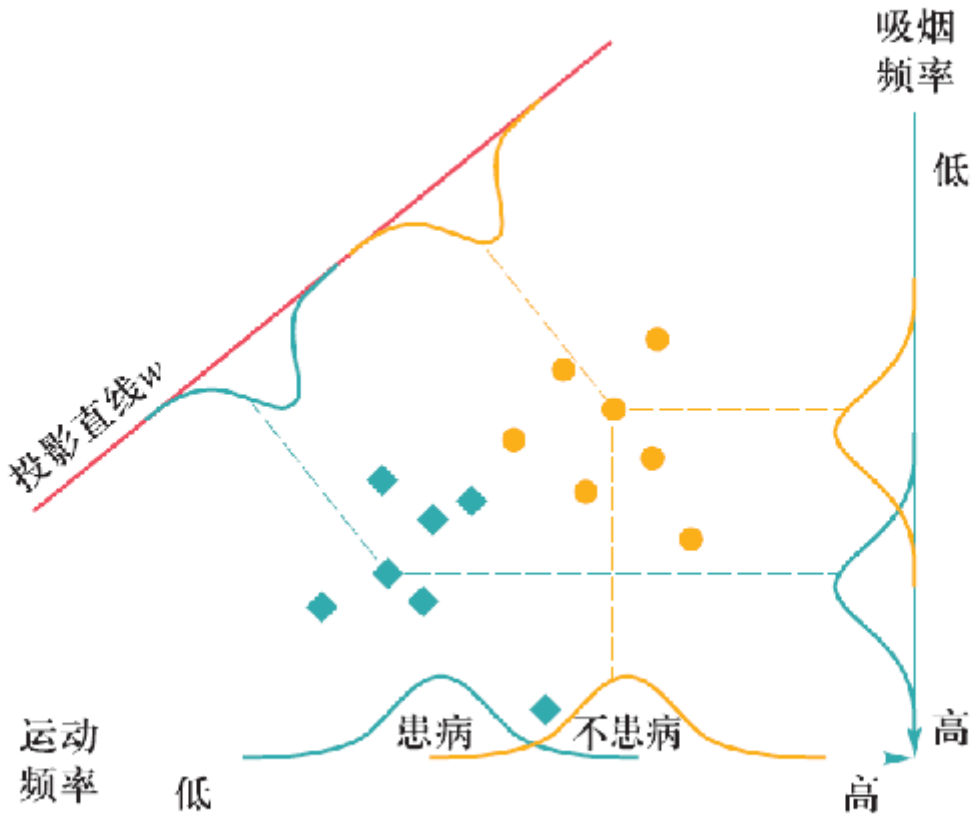


图 4.4　两个类别数据所对应的不同投影方式

- 协方差矩阵说明类内的聚集情况，其中i为类别

$$\Sigma_i = \sum_x (x - m_i)(x - m_i)^T$$

- 投影后的协方差矩阵为 $w^T \Sigma_i w$ ，记为 s_i ，类内的差异可用 $s_1 + s_2$ 度量
- 投影后类间的差异可用 $w^T m_i$ 之间的距离度量，例如 $|w^T m_2 - w^T m_1|^2$
- 为了让两个度量量均可以满足构造

$$J(w) = \frac{|m_2 - m_1|^2}{s_1 + s_2} = \frac{w^T (m2 - m1)(m2 - m1)^T w}{w^T (\Sigma_1 + \Sigma_2) w} = \frac{w^T S_b w}{w^T S_w w}$$

- 其中 S_b 被称为类间散度矩阵，衡量两个类别均值点的分离程度
- S_w 被称为类内散度矩阵，衡量每个类内数据点的分离程度
- w的解法使用拉格朗日乘子法，令 $w^T S_w w$ =1，求w的值，可以得到Fisher线性判别 $S_w^{-1} S_b w = \lambda w$ ，对于二分类问题继而可以推导得到 $w = S_w^{-1}(m2 - m1)$
- 对于多类问题，使用W=(w1,w2,……,wr)实现降维到r维，注意r=min(K-1,d)这与 S_b 的秩一致。

b. PCA

- PCA主要思想：通过分析找到数据特征的主要成分，使用数据的主要成分
- 降维后的结果保持原始数据的原有结构，最大限度地保持高维数据的总体方差结果。

- 皮尔逊相关系数 $\frac{Cov(X,Y)}{\sigma_x \sigma_y}$
- 尽可能将数据向方差最大的方向投影，使得数据所蕴含的信息丢失得尽可能少。
- 其目标是使得数据每一维的方差尽可能大，即 $tr(Y^TY)$ 尽可能大，按照拉格朗日乘子法可推出只要令 $\Sigma w_i = \lambda w_i$ 即可

- 奇异值分解（SVD）矩阵分解方法
- $A = UDV^T$ 其中 $UU^T = VV^T = I$ ， D 是对角矩阵，且每一项都是正实数。
- 可以推导得出U为 AA^T 所有特征向量构成的矩阵，矩阵V为 A^TA 所有特征向量构成的矩阵，$D = \sqrt{(AA^T\text{的特征值})} = U^TAV$

3.5 演化学习

解决最优化问题

- 受自然演化启发的启发式随机优化算法，一般考虑“突变重组”和“自然演化”两个关键因素模拟自然演化过程。
- 遗传算法：引入选择、交叉和变异等操作。
- 抽象做法（对一个待求解的最优化问题）
  - 一定数量的候选解，抽象为染色体。
  - 进化从完全随即个体的种群开始
  - 每一代评价整个种群的适应度
  - 按照适应度选择多个个体进行自然选择和突变，产生新的种群
- 基本流程
  - 初始化具有若干规模数目的群体，设置当前进化代数为0
  - 使用评估函数计算适应值，保存最大的Best值
  - 采用轮盘赌选择算法对群体染色体进行选择操作，产生规模相同的种群
  - 按照概率从种群中选择染色体进行交配，新子代染色体进入新种群，其余直接复制
  - 按照概率对新种群染色体进行变异操作，变异后的染色体取代当前的染色体
  - 迭代，直到进化代数达到指定数目或者best达到指定误差

4. 深度学习

4.1 前馈神经网络与参数优化

a. 神经元

- MCP模型，给定n个二值化的输入数据与连接参数，MCP神经元模型对数据数据进行线性加权求和，然后使用函数 $\Phi(\cdot)$ 将加权累加结果映射为0或1。

b. 前馈神经网络

- 多个隐藏层的多层感知机
- 全连接：两个相邻层之间的神经元互相成对连接（同一层内神经元没有连接
- 层层递进、逐层抽象：明暗幅度的像素点被映射为高层语义对象的概率值
- 非线性映射：
- 误差反馈调优：使用误差后向传播算法，将误差从输出端由后向前传播，逐层去更新神经元链接权重和非线性映射函数。

c. 执行非线性映射的激活函数

- sigmoid函数：概率形式输出，单调递增，非线性变化；缺点：容易出现梯度消失
- ReLU函数：缓解梯度消失问题，缓和过拟合
- Softmax函数：也被称为多项逻辑斯蒂回归模型，一般用于多分类任务中。

d. 神经网络参数优化

- 损失函数（代价函数）：计算模型预测值与真实值之间的误差
  - 均方误差损失函数：MSE
  - 交叉熵损失函数：$H(y_i, \hat{y}_i) = -y_i * log\hat{y}_i$
- 梯度下降：使损失函数最小化的方法
  - $\frac{df(x)}{dx} = lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$
  - 梯度反方向是函数值下降最快的方向
- 误差反向传播
  - 反向传播、链式求导
- 梯度下降的方法
  - 批量梯度下降算法
  - 随机梯度下降算法
  - 小批量梯度下降算法

4.2 卷积神经网络

a. 卷积过程

- 用卷积核定义的权重对图像子块区域内每个像素点进行加权求和
- 卷积核的权重可被认为是记住了领域像素点的若干特定空间模式，忽略了某些空间模式。
- 卷积滤波结果在卷积神经网络中被称为特征图。
- 图像卷积计算对图像进行了下采样操作
- 卷积利用了空间依赖度特点，即相邻像素点之间具有很强的相关性。
- 不同卷积算子具有不同效果：
  - 高斯核：图像平滑操作
- 填充：为了使得边缘位置的像素点也参与卷积滤波；在边缘像素点进行0填充；使用填充后不存在下采样
- 步长：改变卷积核在被卷积图像中移动步长的大小来跳过像素
- 卷积算子特点：
  - 选择性感受野:感受野是指特征图上像素对应的输入图像上的区域。
  - 局部感知，权重共享：减少参数，防止过拟合
  - 下采样约减抽象：$\frac{W-F+2P}{S}+1$

b. 池化

- 最大池化：区域子块中选择值最大的像素点为最大池化结果。
- 平均池化：
- k-Max池化：取前k个最大值

c. 卷积神经网络全貌

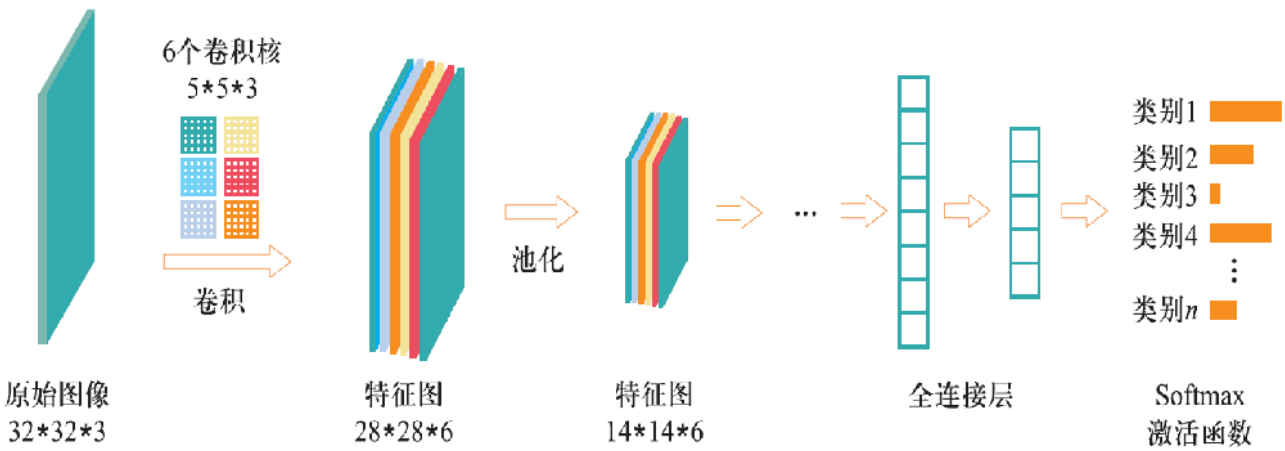


图 5.14　基于卷积神经网络的图像分类示意

4.3 循环神经网络

a. 处理序列数据时采用的网络结构

- 在时刻t，得到输入 x_t 和前一时刻的输出 h_{t-1} ，将产生输出 $h_t = \phi(Ux_t + Wh_{t-1})$

b. 应用模式

三种应用模式：按照输入序列数据和输出序列数据包含的单元数量进行分类。“多对多” “多对一” “一对多”

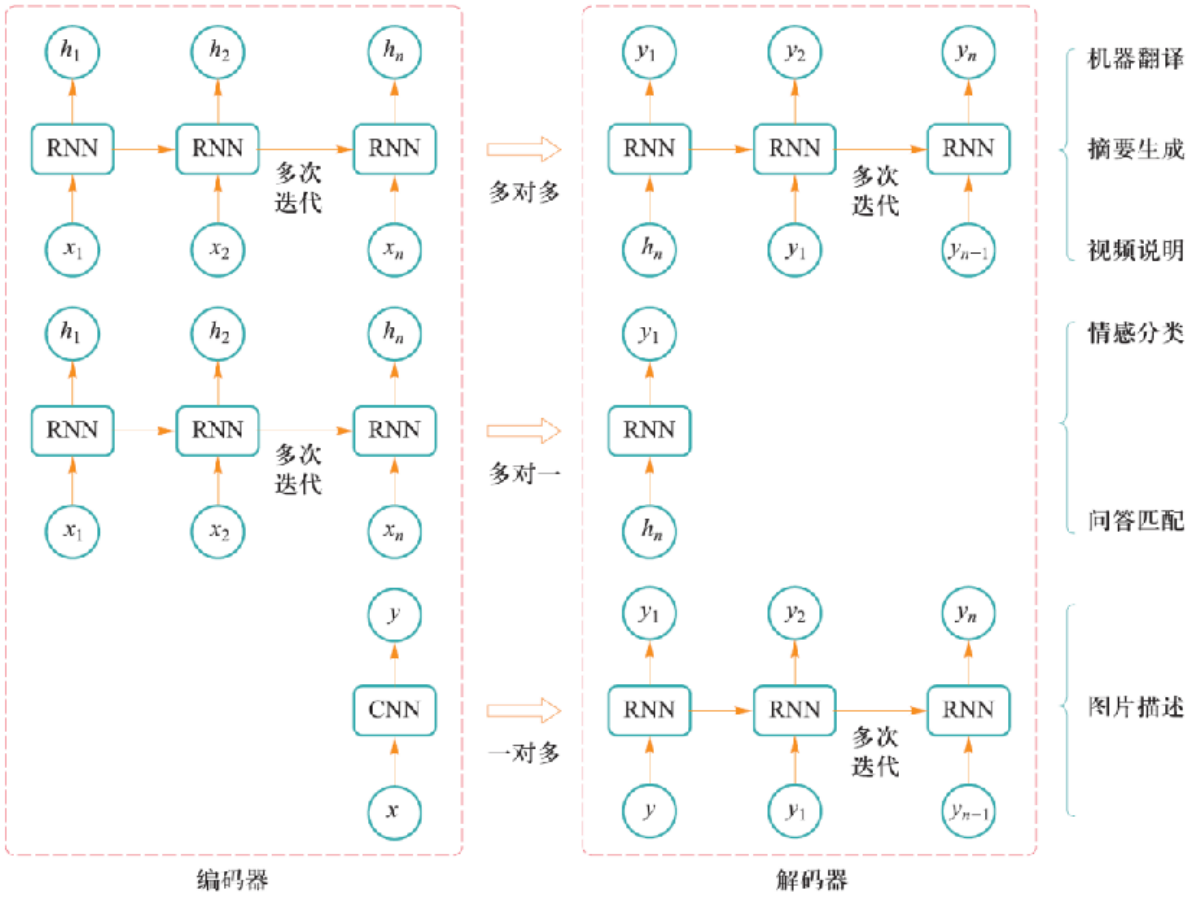


图 5.17　输入输出不同情况下循环神经网络结构示例

c. 梯度传递

- 彼此相关信息在序列中相距较远时，循环神经网络无法捕捉到数据中的时序依赖关系；当输入序列过长时容易出现梯度消失，梯度爆炸
- 假设一个RNN的表达为 $O_t = g(W_o h_t) = g(W_o \sigma(W_x x_t + W_h h_{t-1}))$ 有损失函数 $E_t = f(O_t, O_{true_t})$ ，则

$$\frac{\partial E_t}{\partial W_x} = \Sigma \frac{\partial E_t}{\partial O_t} \frac{\partial O_t}{\partial h_t} (\Pi \frac{\partial h_j}{\partial h_{j-1}}) \frac{\partial h_i}{\partial W_x}$$ (i从1到t，j从t到i+1)

由于tanh的导数取值在0-1之间，容易出现梯度消失的问题。

d. 长短时记忆模型(Long Short-Term Model)

- 引入内部记忆单元+门：对当前时刻输入信息以及前序时刻生成的信息进行整合和传递。
- 输入门、输出门、遗忘门：对于当前时刻输入的 x_t 和上一时刻的编码 h_{t-1} ，输入门、遗忘门和输出门通过各自的输出得到 i_t 、 f_t 和 O_t
- 结合前一时刻内部记忆单元 c_{t-1} 更新当前内部记忆单元，并得出隐式编码 h_t

- 遗忘门控制上一时刻内部记忆单元有多少信息可以累积到当前时刻内部记忆单元；输入门控制有多少信息流入当前时刻内部记忆单元；输出门控制内部记忆单元信息到输出编码的量
- $c_t = f_t * c_{t-1} + i_t * tanh(W_{xt}x_t + W_{ht}h_{t-1} + b_t)$ 其中*为按位乘
- $h_t = tanh(c_t) * o_t$
- 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。
- 隐式编码和内部记忆单元激活函数为tanh(-1,1)实现信息正负增益的理解
- 内部记忆单元：长时记忆；隐式编码：短时记忆

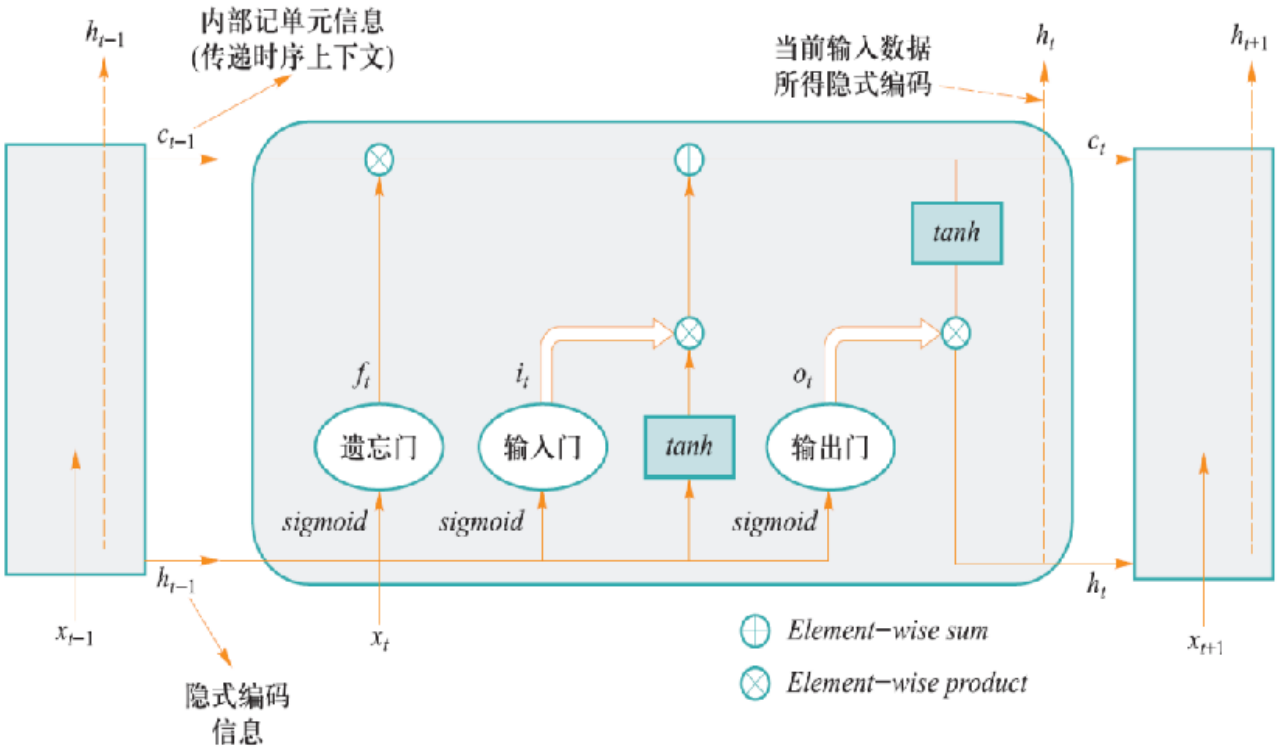


图 5.19　长短时记忆网络模型

（注：在每个时刻 t，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；只有 h_t 作为本时刻的输出用于分类等处理）

4.4 注意力机制与正则化

a. 注意：意识对一定信息或对象的指向与集中的过程

- 对于单词有编码 w_i，对于自注意力机制有共享矩阵 W_q, W_k, W_v
- 对于单词i有查询向量 $q_i = W_q \times w_i$，键向量与值向量同理
- 要计算 w_i 的自注意力，即先计算关联度 $q_i \cdot k_j$，并将计算结构通过softmax进行归一化，再乘以对应的值向量并相加。
- 使用多头注意力机制的目的是从多个角度挖掘某个单词与其他单词之间的概率关联
- 计算得到注意力关联后，通过前向神经网络进行非线性映射

b. 正则化技术（缓解过拟合现象）

- Dropout:在每次迭代训练中以一定概率随机屏蔽每一层中的若干神经元，用余下神经元构成的网络继续训练。
- 批归一化：通过规范化把神经网络每层中任意神经元的输入值分布改变成均值为0、方差为1的标准正态分布，从而使输出值被映射到非线性函数较大的区域，使得梯度变大。
- L1和L2正则化：
  - $minLoss + \lambda \times \Phi(W)$
  - $\Phi(\cdot)$ 一般用模型参数W的范数形式来表示
  - L_0 计算非零参数个数，属于NP难问题，一般不用
  - L_1 计算参数绝对值之和，也被称为“稀疏规则算子”
  - L_2 模型参数中各元素平方和的开方

4.5 深度学习的应用

a. 词向量生成(Word2Vec)

- 使用分布式向量表达对不同单词进行表达，刻画不同单词之间的语义相关性。
- word2vec是用神经网络权重作为词的向量表达
- 多对一为Cbow，一对多为skip-gram

b. 图像分类与目标定位

- 同时进行分类和定位任务：分类设计损失函数，定位设计包围盒（左上方横纵坐标+宽+高）四维输出设计损失函数，

5. 强化学习

5.1 强化学习问题定义

a. 术语概念

i. 智能体：强化学习算法的主体

ii. 智能体以外的一切统称为环境

iii. 状态：智能体对环境的理解和编码

图 6.1　强化学习的过程

iv. 动作：智能体影响环境的方式

v. 策略：智能体所处状态下去执行动作的依据

vi. 奖励：智能体序贯式采取一系列动作后从环境获得的收益

b. 强化学习的特点

基于评估；交互性；序列决策过程（决策前后关联）；

表 6.1　三种学习方式特点对比

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

c. 马尔可夫决策过程(MDP)

- 马尔可夫链（MC）：满足马尔可夫性的离散随机过程，也被称为离散马尔可夫过程。（t+1时刻状态仅与t时刻有关）

$P(X_{t+1}=x_{t+1}|X_i=x_i,\cdots)=P(X_{t+1}=x_{t+1}|X_t=x_t)$

- 引入奖励机制：用 $R(S_t,S_{t+1})$ 描述从 S_t 到 S_{t+1} 的奖励
- 回报：该时刻可得到的累加奖励，$G_t=R_{t+1}+\gamma R_{t+2}+\cdots$
- 马尔可夫奖励过程(MRP)：马尔可夫链中加入奖励函数和折扣因子，MRP=(S,P,R,r)
- 马尔可夫决策过程：$MDP=(S,A,P,R,\gamma)$
  - 状态集合S：所有可能的状态的集合
  - 动作集合A：智能体能采取的所有动作所构成的集合
  - 状态转移概率：在当前状态 S_t 下采取动作 A_t 后进入状态 S_{t+1} 的概率
  - 奖励函数 $R(S_t,A_t,S_{t+1})$：当前状态 S_t 下采取动作 A_t 后进入状态 S_{t+1} 的奖励
  - 折扣因子 γ
- 轨迹：状态序列
- 状态序列包含终止状态的为分段问题；否则为持续问题；一个初始到终止状态的完整轨迹称为片段。

d. 强化学习定义

- 策略函数 π :智能体在状态S下选择动作a的概率，简记为 $a=\pi(S)$
- 价值函数V：$V_\pi(S)=E_\pi[G_t|S_t=S]$ 在状态S时采取策略 π 得到的回报的期望
- 动作-价值函数q：$q_\pi(S,a)=E_\pi[G_t|S_t=S,A_t=a]$，智能体在t时刻处于状态S，并选择a后，基于策略 π 的回报。
- 强化学习即给定一个马尔可夫决策过程，学习最优策略，使得任意状态的价值函数最大

e. 贝尔曼方程

- $V_\pi(S)=\Sigma\pi(S,a_i)E_\pi[G_t|S_t=S,A_t=a_i]=\Sigma\,\pi(S,a_i)q_\pi(S,a_i)$
- $q_\pi(S,a)=E_{s'=P(\cdot|S,a)}[R(S,a,S')+\gamma E_\pi[G_{t+1}|S_{t+1}=s']]=E_{s'=P(\cdot|S,a)}[R(S,a,S')+\gamma\,V_\pi(s')]$
- 价值函数的贝尔曼方程为：

$$V_\pi(S)=E_{a=\pi(S)}E_{s'=P(\cdot|S,a)}[R(S,a,S')+\gamma V_\pi(s')]$$

- 动作-价值函数的贝尔曼方程为：

$$q_\pi(S,a)=E_{s'=P(\cdot|S,a)}[R(S,a,S')+\gamma E_{a'=\pi(S)}[q_\pi(S,a')]]$$

5.2 基于价值的强化学习

a. 从任意一个策略开始，首先计算该策略下的价值函数，改进策略使得价值函数最大，直到收敛。

- 通过策略计算价值函数的过程叫做策略评估

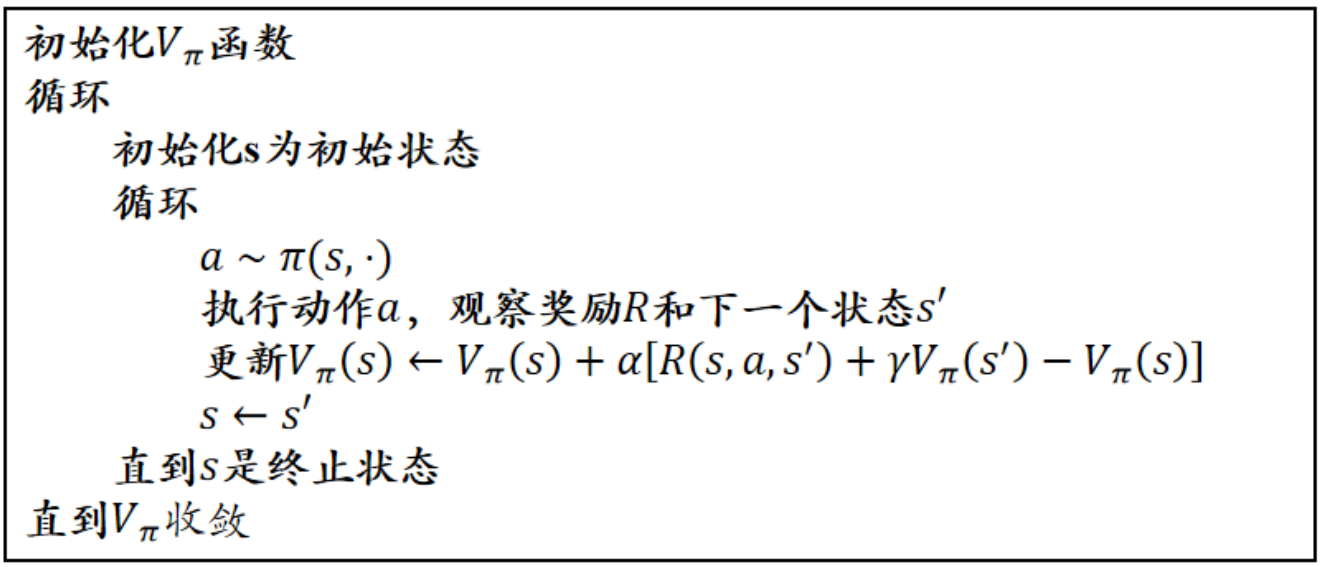
- 通过价值函数优化策略的过程叫做策略优化
- 策略评估和策略优化交替进行的强化学习求解方法：通用策略迭代（GPI）

b. 策略优化定理

- 如果两个策略 π' 和 π 满足 $q_\pi(s,\pi'(s)) \geq q_\pi(s,\pi(s))$，则有 $V_{\pi'}(s) \geq V_\pi(s)$
- 则令 $\pi'(s)$ 为 q_π 最大的动作时，则 $\pi'(s)$ 为当前策略的改进。

c. 策略评估（如何计算各状态的价值函数）【状态集合有限】

- 动态规划方法
  - 使用迭代的方法求解贝尔曼方程
  - 缺点：需要事先知道状态转移概率，无法处理集合大小无限的情况。
- 蒙特卡洛采样
  - 按照当前策略采样若干轨迹，计算轨迹集合的反馈的均值作为状态的价值函数。
- 时序差分
  - 蒙特卡洛和动态绘画方法的结合：从实际经验中获取信息，无需提前获知环境模型的全部信息；能够利用前序已知信息进行在线实施学习，无需等到片段结束；



- 在策略评估动态规划法的基础上，每次迭代只对一个状态进行策略评估和策略优化的算法被称为价值迭代算法。

d. 策略评估：Q学习

- 直接记录和更新动作-价值函数，避免不知道状态转移概率而导致不知道 q_π (用于更新策略函数)。

算法 6.6 Q学习算法

函数： QLearning

输入： 马尔可夫决策过程 MDP=(S, A, P, R, γ)

输出： 策略 π

随机初始化 q_π

repeat

  $s \leftarrow$ 初始状态

  repeat

    $a \leftarrow \text{argmax}_{a'} q_\pi(s, a')$

    执行动作 a，观察奖励 R 和下一个状态 s'

    $q_\pi(s, a) \leftarrow q_\pi(s, a) + \alpha[R + \gamma \text{max}_{a'} q_\pi(s', a') - q_\pi(s, a)]$

    $s \leftarrow s'$

  until s 是终止状态

until q_π 收敛

$\pi(s)$: $\text{argmax}_a q(s, a)$

- 探索与利用：用 ϵ 贪心策略代替动作选择操作

e. 参数化与深度学习

- 问题：
  - 动作-价值函数值过多，导致难以存储；
  - 某些状态的访问次数很少，价值估计不可靠；
- 解法：动作-价值函数参数化（用回归模型拟合q函数）
  - 如果回归模型是一个深度神经网络，即深度强化学习算法
- 神经网络真实值 $R + max q_\pi(s', a'; \theta)$ ，均方差作为误差；
- 深度强化学习的问题：
  - 采样不足：相邻的样本来自同一条轨迹，样本相关性强，容易过拟合（针对未被更新价值的状态）
  - 难以收敛：神经网络的单步优化同时影响预设真实值和估计值，算法收敛不稳定
- 深度Q网络
  - 经验重现：(s,a,R,s')
  - 两组参数: θ^- 参数保持相对稳定，以较低频率更新

5.3 基于策略的强化学习

a. 策略梯度法

- $\pi_\theta(s,a)$,函数取值表示状态s下选择动作a的概率；
- 概率随参数连续变化，具有光滑性，有利于算法收敛；
- $J(\theta)=V_{\pi\theta}(s_0)$ 最大

b. 策略梯度定理

- 其中 $\mu_{\pi_\theta}(s)$ 为访问s的期望/概率

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \sum_s \mu_{\pi_{\boldsymbol{\theta}}}(s) \sum_a q_{\pi_{\boldsymbol{\theta}}}(s,a)\pi_{\boldsymbol{\theta}}(s,a)$$

$$\propto \sum_s \mu_{\pi_{\boldsymbol{\theta}}}(s) \sum_a q_{\pi_{\boldsymbol{\theta}}}(s,a)\nabla_{\boldsymbol{\theta}}\pi_{\boldsymbol{\theta}}(s,a)$$

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \propto \mathbb{E}_{s,a\sim\pi}\left[\mathbb{E}_{\mathcal{T}(s,a)\sim\pi}[G_t|s,a]\nabla_{\boldsymbol{\theta}}\ln\pi_{\boldsymbol{\theta}}(s,a)\right]$$
$$= \mathbb{E}_{s,a,\mathcal{T}(s,a)\sim\pi}[G_t\nabla_{\boldsymbol{\theta}}\ln\pi_{\boldsymbol{\theta}}(s,a)]$$

- 计算梯度，使用梯度上升法（增加梯度）优化策略
- 该算法也被称为REINFORCE，与蒙特卡洛一样需要一个片段结束

c. Actor-Critic算法

- 参数化价值函数和策略函数

5.4 深度强化学习应用

a. 挑战

- 奖励的设置
- 样本的采集
- 局部最优解与探索
- 训练时的不稳定性与方差
- 泛化和迁移能力

6. 人工智能博弈

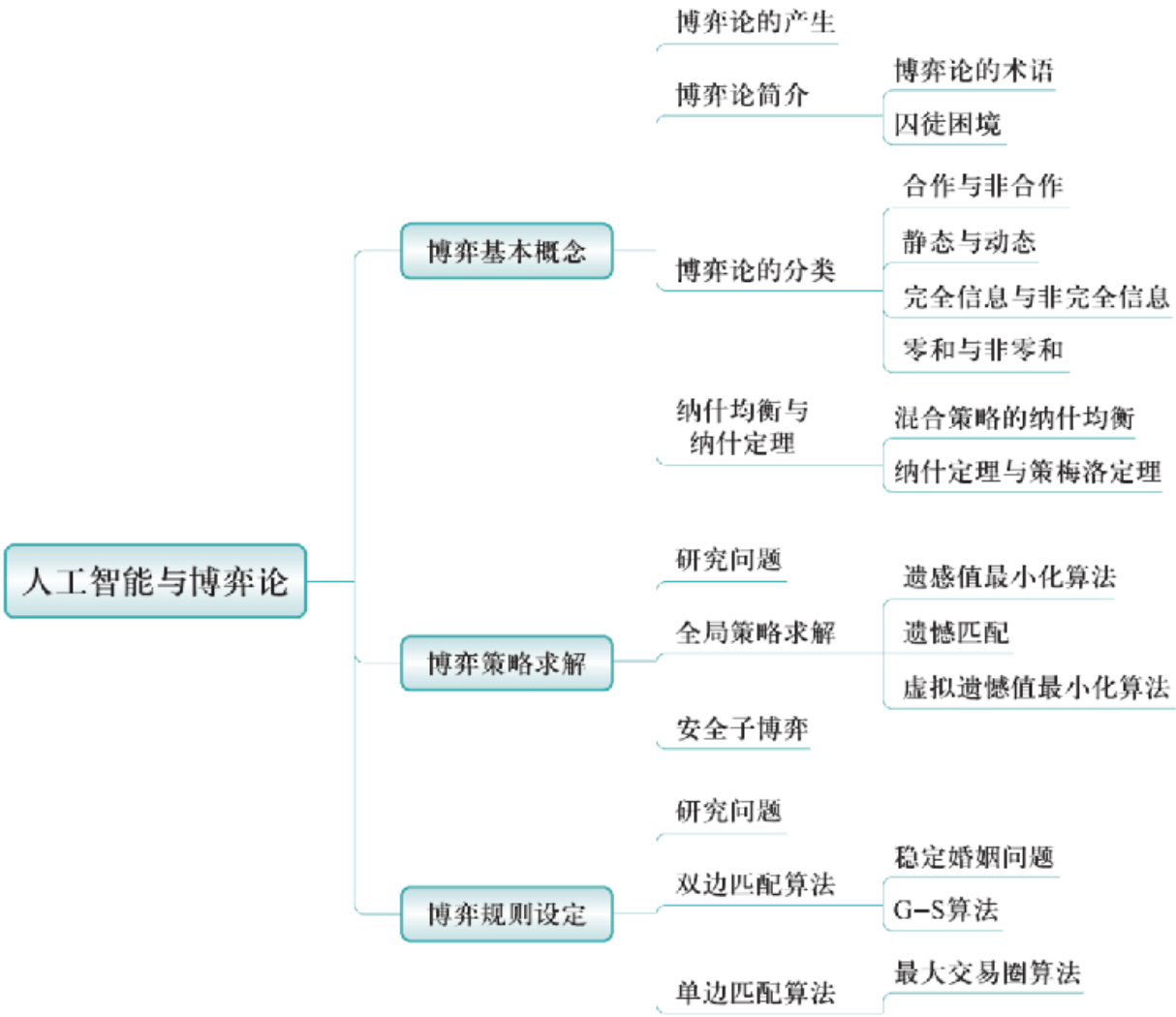


图 7.7　博弈论研究内容

6.1 博弈论相关概念

a. 博弈论

- 博弈行为：带有相互竞争性质的主体，为了达到各自的目标和利益，采取的带有对抗性质的行为。
- 博弈论：主要研究博弈行为中最优的对抗策略及其稳定局势。
- 现代系统博弈理论的初步形成：《博弈论与经济行为》；现代博弈论之父：冯·诺伊曼
- 参与者或玩家：参与博弈的决策主体
- 策略：参与者可以采取的行动方案，在采取行动前就准备好的行动方案
  - 策略集：某个参与者可采纳策略的全体组合
  - 局势：所有参与者各自采取行动后所形成的状态
  - 混合策略：参与者可通过一定概率分布选择若干不同的策略
  - 纯策略：参与者每次行动都选择确定的策略
- 收益：各参与者在不同局势下得到的利益，混合策略下为期望收益

- 规则：对参与者行动的先后顺序，参与者获得信息内容的规定
- 博弈论的研究范式：建模者对参与者规定可采取的策略集，查看当参与者采取不同的策略集以最大化收益时，会产生什么样的结果。

b. 博弈的分类

- 参与者或玩家合作博弈与非合作博弈
  - 合作博弈：参与者可以组成联盟以获取更大的收益
  - 非合作博弈：参与者在决策过程中彼此独立，不事先达成合作意向
- 静态博弈与动态博弈：
  - 静态博弈：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：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的决策
- 完全信息博弈与不完全信息博弈：
  - 完全信息博弈：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：并非所有参与者均掌握了所有信息。

c. 纳什均衡

- 纳什均衡：博弈的稳定局势，即参与者做出一种策略组合，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。
- 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。

d. 策梅洛定理

- 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者后手必胜策略或者双方保平策略

6.2 博弈策略求解

a. 虚拟遗憾最小化算法

- 玩家i在博弈中采取的策略记为 σ_i ，对于除了玩家i外其余玩家的策略记为 σ_{-i} 。
- 最优反应策略：在给定策略组合 σ 的情况下，假设玩家i在终结局势下的收益为 $\mu_i(\sigma)$ ，对玩家i的最优反应策略为 $\mu_i(\sigma_i^*) \geq \mu(\sigma', \sigma_{-i})$
- 对于策略组合 σ^* ，如果每个玩家的策略相对于其他玩家的策略都是最优反应策略，则 σ^* 就是一个纳什均衡策略，在有限对手，有限对策的情况下，纳什均衡一定存在。
- 遗憾最小化算法：根据以往博弈过程中所得到遗憾程度来选择未来反应
- 遗憾值： $\Sigma(\mu_i(\sigma_i, \sigma_{-i}^t) - \mu_i(\sigma^t))$ （按t累加）
- 遗憾匹配：根据遗憾之大小进行后续T+1轮博弈的策略
- 有效遗憾值：大于0的遗憾值

$$P(\sigma_i^{T+1}) = \begin{cases} \dfrac{Regret_i^{T,+}(\sigma_i)}{\sum_{\sigma_i' \in \Sigma_i} Regret_i^{T,+}(\sigma_i')} & if \sum_{\sigma_i' \in \Sigma_i} Regret_i^{T,+}(\sigma_i') > 0 \\ \dfrac{1}{|\Sigma_i|} & \text{otherwise} \end{cases}$$

- 为处理序贯式博弈，提出虚拟遗憾最小化算法
- 行动路径h的虚拟价值h为：

$$v_i(\sigma, h) = \sum_{z \in Z} \underbrace{\pi_{-i}^{\sigma}(h)}_{\text{不考虑玩家}i\text{的策略到达当前节点概率}} \times \underbrace{\pi^{\sigma}(h,z)}_{\text{从当前节点到叶子结点概率}} \times \underbrace{u_i(z)}_{\text{叶子结点}z\text{收益}}$$

- $r_i(a,h) = v_i(\sigma_{I \to a}, h) - v_i(\sigma, h)$ 路径h下动作a的遗憾值
- 累加所有可以到I的路径下动作a的遗憾值，得到信息集I下动作a的遗憾值
- 最后进行遗憾匹配

$$\sigma_i^{T+1}(I,a) = \begin{cases} \dfrac{Regret_i^{T,+}(I,a)}{\sum_{a \in A(I)} Regret_i^{T,+}(I,a)} & if \sum_{a \in A(I)} Regret_i^{T,+}(I,a) > 0 \\ \dfrac{1}{|A(I)|} & \text{otherwise} \end{cases}$$

算法流程如下：

1) 初始化遗憾值和累加策略表为0

2) 采用随机选择的方法来决定策略

3) 利用当前策略与对手进行博弈

4) 计算每个玩家采取每次行为后的遗憾值

5) 根据博弈结果计算每个行动的累加遗憾值大小来更新策略

6) 重复3)到5)步若干次，不断的优化策略

7) 根据重复博弈最终的策略，完成最终的动作选择

b. 安全子博弈

- 从当前已经完成的部分博弈出发，将接下来的博弈过程视为一个单独的子博弈，然后找到子博弈的最优反应策略，在接近叶节点时可以得到较准确的结果。
  - 完全信息博弈下：子博弈与其他部分无关，可以单独考虑
  - 非完全信息博弈下：子博弈与其他部分相关，不能单独考虑
- 安全子博弈：子博弈求解过程中，得到的结果一定不差于全局的近似解法

6.3 博弈规则设计

a. 研究问题

- 目标：使得整体利益最大化

b. 双边匹配问题

- 需要双向选择的问题
- 稳定婚姻问题
  - 即给定n位男士和n位女士，及其喜好偏序进行匹配
  - G-S算法：（可以得到稳定匹配的结果）
    - 单身男性向最喜欢的女孩表白
    - 所有收到表白的女性向从其表白男性中选择最喜欢的男性，暂时匹配
    - 未匹配的男性继续向没有拒绝它的女性表白：未完成匹配的女性从表白者中选择最喜欢的男性；已完成的可以拒绝之前的，重新匹配
    - 迭代循环，直到所有人都成功匹配。

c. 单边匹配问题

- 最大交易圈算法（TTC）
  - 对物品进行随机分配
  - 交易者连接一条指向其最喜欢物品的边；物品连接到其占有者或者是具有高优先权的交易者
  - 此时形成一张有向图，且必存在环，这种环被称为交易圈。交易圈中的交易者，将每人指向结点所代表的标的物赋予交易者，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市场
  - 迭代，直到无法形成交易圈

6.4 非完全信息博弈