CH2

达特茅斯会议(1956年6月18日至8月17日)

自动计算机

计算机如何使用语言来编程

神经网络

计算复杂性

智能算法自我学习与提高

智能算法归纳与演绎能力

智能算法随机性和创造力

以符号主义为核心的逻辑推理

符号主义人工智能

原子命题：不包含其他命题作为其组成部分的命题，又称简单命题。

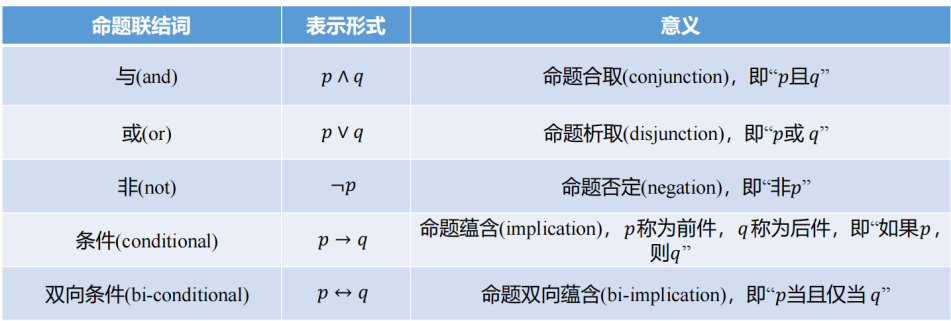
悖论不是一个命题（仅有真与假）

如果“我正在说谎”是真的，那么说话者确实在撒谎。

撒谎的定义是说不真实的话。

因此，如果说话者在撒谎，那么命题“我正在说谎”应该是假的。

复合命题：包含其他命题作为其组成部分的命题



p→q 与 ’p or q等价

p←→q 与 p，q均为真/假等价

逻辑等价：具有相同的真假结果，一般用≡来表示



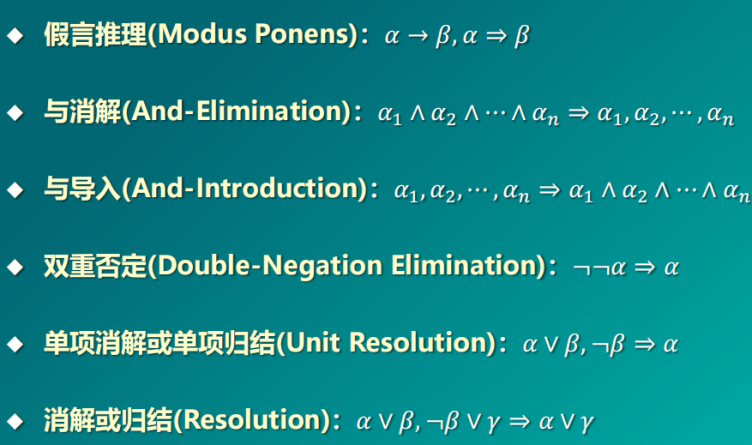
逻辑推理，从前提推出结论

谓词逻辑：刻画主体（个体和群体）之间逻辑关系的方法

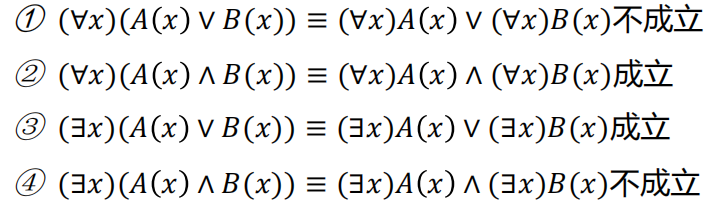
原子命题无法包含个体、整体及之间的关系

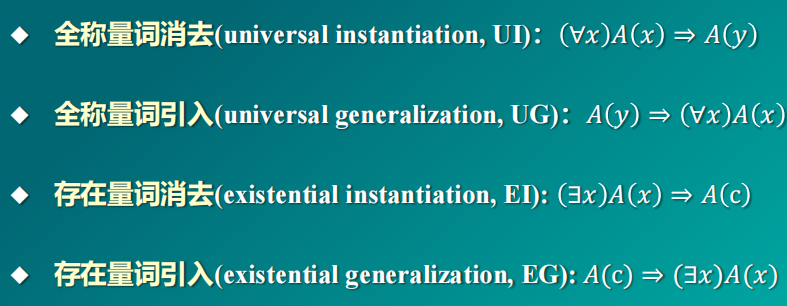
个体：个体是指所研究领域中可以独立存在的具体或抽象的概念。

谓词：谓词是用来刻画个体属性或者描述个体之间关系存在性的元素，其值为真或为假（可有一元谓词和多元谓词之分）



量词（存在量词，全称量词）



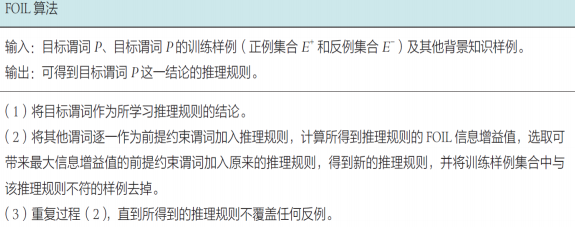


CH2 PPT26证明

知识图谱(knowledge graph)由有向图(directed graph)构成，被用来描述现实世界中实体及实体之间的关系

first order logic（一阶逻辑）

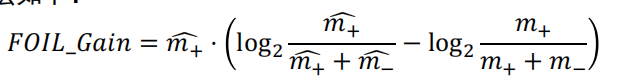
序贯覆盖学习推理（FOIL）



目标谓词，前提约束谓词（需要学习的知识关系）

正例集合/反例集合：根据已有关系和目标谓词进行匹配，和目标谓词一直的关系记录为正例集合；不一致的记录为反例集合

信息增益值计算（负无穷记作NA）：



根据不同的前提约束谓词，计算信息增益值，选取最大项（选择后，需要根据加入的前提约束谓词对正反例进行调整）

一般使用x，y，z来表示（相同/不同的）人

路径排序推理算法（PRA）

通过学习路径表示不同个体间的关系 [正反例路径]（使用向量表示，不同个体间可能有多个关系路径符合，并添加标签）

概率图

贝叶斯网络：有向无环图（满足局部马尔可夫性）

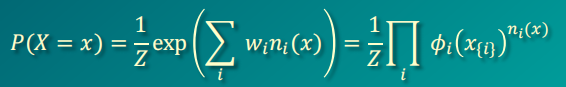
父亲节点有条件地独立于它的非后代节点

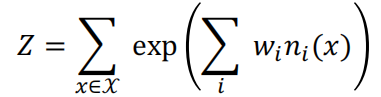


P(A|B) = P(AB) / P(B)

马尔科夫链：无向图网络结构

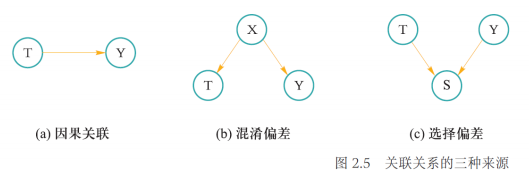
概率图推理：





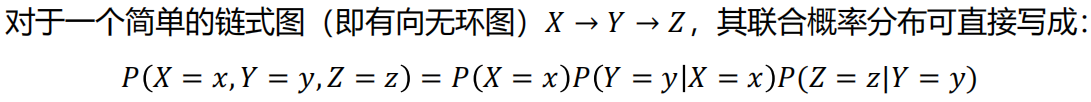
根据已有信息的表述，计算当前情况可能发生的概率

辛普森悖论（ Simpson’s Paradox ）



混淆偏差（关联）： X是T的原因，同时也是Y的原因。如果忽略掉X，那么T和Y在数据中会存在虚假关联。

选择（偏差）关联：T是S的原因，Y也是S的原因。如果我们基于特定S的取值来选择一部分变量进行研究，那么T和Y在数据中就会存在虚假关联。



Do算子

P(Y=y | X=x) 反映的是在取值为x的个体X上，Y的总体分布；而P(Y=y | do( X=x ))反映的是如果将每一个X取值都固定为x时，Y的总体分布。

因果效应差（平均因果效应，ACE）

Do算子往往与因果图相联系，可以在因果图中进行do算子的操作来表明后续概率的计算累乘（称为操纵概率）

可用正常(无干预)条件下的条件概率来计算干预后的条件概率（调整公式）

一般因果效应差大于5%认为有效

反事实模型，通过已有结果（系数）计算环境变量，通过已得到的环境变量与反事实假设，对反事实结果进行预测。

CH3

以问题求解为核心的探寻搜索

搜索

搜索的形式化表达：<状态、动作、状态转移、路径/代价、目标测试〉

状态：初始状态与终止状态

动作：算法从一个状态转移到另外一个状态所采取的行为（动作）

状态转移：算法选择了一个动作之后，其所处状态也会发生相应变化（结果）

在搜索树中，同一个标号一定表示相同的状态

搜索算法的评判标椎

完备性：当问题存在解时，算法是否能找到一个解，虽然这个解可能不是最优解。

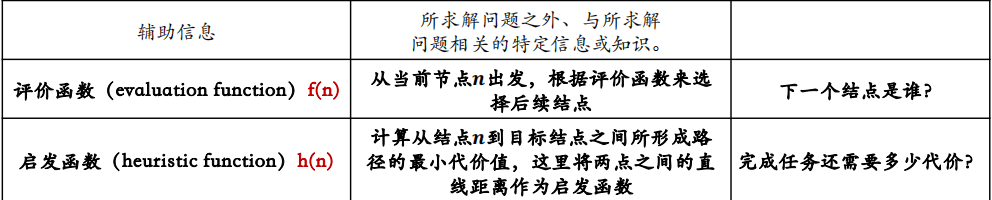
最优性：搜索算法是否能保证找到的第一个解是最优解。

时间复杂度：找到一个搜索路径所需时间。

空间复杂度：算法运行时所需的内存空间。

函数pick\_from：决定了扩展结点的顺序

函数successor\_nodes：决定了哪些结点可被放入边缘集合供后续被扩展

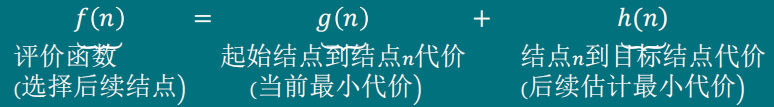


贪婪最佳优先搜索（有信息/启发式搜索）

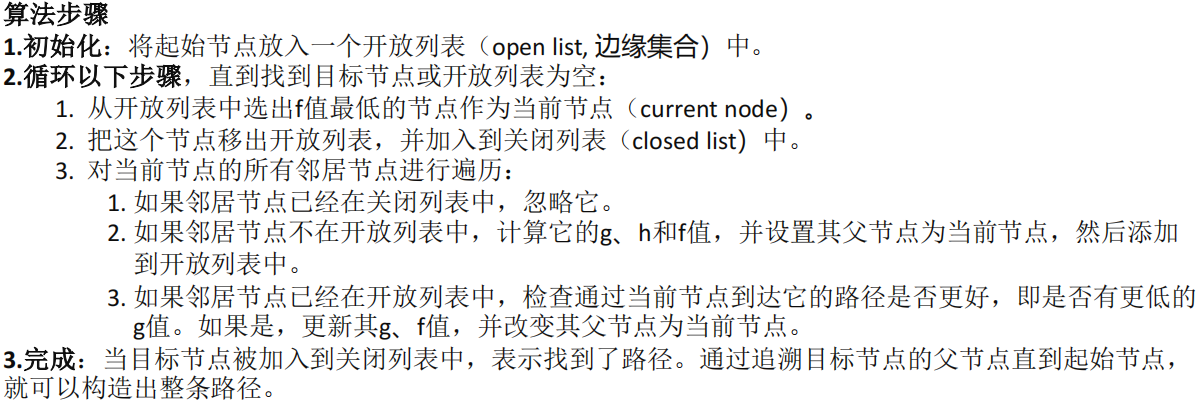
贪婪最佳优先搜索算法时间和空间复杂度均为O(bm)

贪婪最佳选取的评价函数最小项是所有出现过状态（评价函数）的最小项

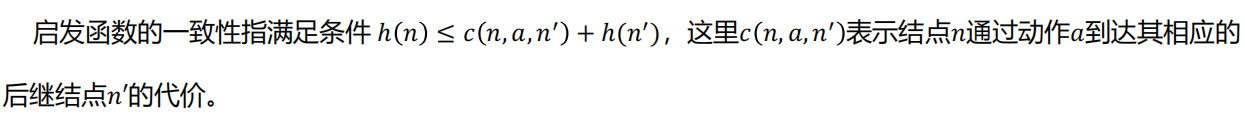
A\*算法



通过评价函数选择下一步（step）结点



可容性（可采纳性Admissibility）：对于任意结点n，有h (n) ≤ h \*(n) 。如果n是目标结点，则有h (n) = 0。



满足一致性条件的启发函数一定满足可容性条件

如果满足以下条件，则A\*算法是完备的

搜索树中分支数量是有限的，即每个结点的后继结点数量是有限的

单步代价的下界是一个正数

启发函数有下界

如果启发函数是可容的，那么A\*算法满足最优性

Minimax搜索

状态：状态s包括当前的游戏局面和当前行动的智能体

Minimax算法的时间复杂度为O(bm) ，空间复杂度为O(bm)

αβ减枝：

对于MAX结点，如果其孩子结点（MIN结点）的收益大于当前的α值，则将α值更新为该收益；对于MIN结点，如果其孩子结点（MAX结点）的收益小于当前的β值，则将β值更新为该收益。根结点（MAX结点）的α值和β值分别被初始化为 −∞ 和 +∞。

Min层修改β，Max层修改α

CH3 PPT35

蒙特卡洛树搜索

探索（exploration）与利用（exploitation）的平衡

（探索-利用）贪心算法，在利用的过程中引入一定的探索

上限置信区间算法：

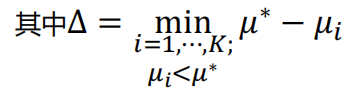
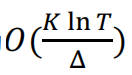
UCB1算法的策略是：为每个动作的奖励期望计算一个估计范围，优先采用估计范围上限较高的动作

综合考虑探索与利用，选取如下公式计算得到的最大项



T(i,t-1)表示访问i结点动作尝试次数；x表示总尝试次数中，（多次）动作i获得的奖励均值，t为同父节点下选取尝试总次数

已证明，UCB1算法在经过T次测试后悔值函数的期望上界为K为（不同）动作总数目，T为尝试总数目



选择：根据UCB1的计算，向下递归选择结点

扩展：若结点不是叶子结点，则随机扩展一个未扩展的结点

模拟：从当前节点出发，一般使用随机策略直到找到一个终止节点为止

反向传播：回溯更新路径上的UCB1计算所需信息（随机模拟部分不需要更新）

在更新时，会将MIN层结点现有总分加上终局得分分数，MAX层结点现有总分减 去终局得分分数（选择时都选取最大的项）

CH4

以数据驱动为核心的机器学习

监督学习：学习得到一个最优映射函数f（又称决策函数）

训练集：用于训练的数据

验证集：训练集中的一部分数据（评估模型以调整参数）

测试集：最终测评模型性能

任何⼀个机器学习模型如果在⼀些训练集样本误差小（off-training set error），那么必然在另外⼀些训练集样本上表现欠佳，任何模型在平均意义上而言其性能都是一样的，即没有放之四海而皆准的最好算法

损失函数

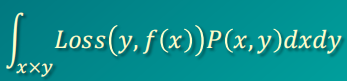


在机器学习中，需要保证模型在**训练集**上所取得性能与在**测试集**上所取得性能保持一致，即模型具有泛化能力（generalization）

映射函数f在训练集上所产⽣损失⼀般被称为经验风险；经验风险越小说明模型对训练集数据拟合程度越好

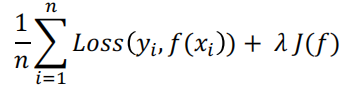


期望风险（真实风险，真实误差）；从所有数据中计算模型产生的损失，记该任务中所有数据的联合分布为P( X, Y)





若使用一批数据反复训练会使得模型越来越复杂，虽然经验风险下降，但err增加，导致过学习的现象；为了防止过学习，结构风险最小化(structural risk minimization)引入正则化 (regularizer) 或惩罚项(penalty term)，来降低模型复杂度



模型衡量方法：

真正例（True Positive, TP）/假正例（False Positive, FP)/真反例（True Negative, TN）/假反例（False Negative, FN）

准确率ACC：（TP + TN）/(P + N) 所有预测正确项；

错误率： errorRate = 1 - ACC

精确率：预测为正中实际为正准确率 precision = TP / (TP + FP)

召回率：实际为正中预测为正 recall = TP / (TP + FN)

综合分类率：F1-score = 2\*precision \* recall /（precision + recall）

频率会收敛到概率

频率学派：最大似然估计，使观测数据发生概率最大的模型参数

贝叶斯学派：似然概率（模型参数产生数据的概率）与先验概率（没有任何实验数据时对模型参数的经验判断）乘积最大，又称为最大后验估计（maximum a posteriori estimation, MAP）。[数据已经出现的条件下求取参数的可能值]

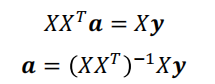
回归分析（监督学习）

分析不同变量之间存在关系的研究叫作回归分析，刻画不同变量之间关系的模型称为回归模型

最佳回归模型将使得残差平方和（预测值和实际值差的平方和）的平均值最小

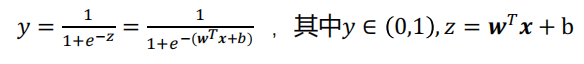
使用最小二乘法求取最优解

经矩阵化简后，回归模型的参数可由下式得到，其中X矩阵为回归模型中自变量的矩阵形式（每一行为一个自变量，每一列为一组数据，最后一行全为1 --- 对映于偏置量），y为预期因变量矩阵形式



线性回归对离群点十分敏感，导致模型不稳定

逻辑斯蒂回归(logistic regression) --- 只能解决二分类的问题（也对异常值敏感）



本质是sigmoid函数

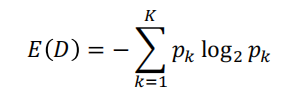
决策树（监督学习）

采用树状结构来逐步完成决策判断（根据不同条件不停向下进行判断，直到目标标签已经完全分类）

通过信息熵大小来选择进行比较的属性，信息熵越大，说明该集合的不确定性越大，“纯度”越低

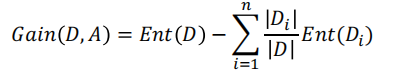
选择属性划分样本集前后信息熵的减少量被称为**信息增益**（information gain）

信息熵计算表达式为，其中k为目标分类类别，但信息熵的计算过程中本身有属性的选取，属性可能将空间划分为多个区域（每个区域对映一个Ent(D) 计算量，需要对其进行（样本占比）累加而得到 E(D) ）



E(D) 值越小，表示D包含的信息越确定，也称D的纯度越高

信息熵起始值基于分类目标类别，属性区域默认为 1



无监督学习：无标签

K-means聚类（无监督学习）

目的是分为K个类别，使簇内方差最小（不能保证全局最优）

是易受初始值影响的迭代算法

算法：

先初始化聚类质心

计算不同node所属的聚类类别

更新聚类中心（均值计算即可，易受到极端值的影响）

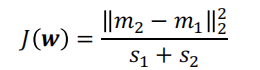
迭代至到达次数上限或收敛

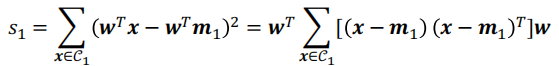
半监督学习：一些有标签，一些无标签

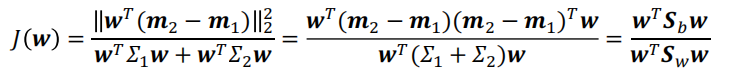
线性判别分析（LDA）--- 基于监督学习的降维方法（标签用在类别的确定上）

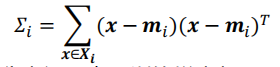
利用其类别信息，将其线性投影到一个低维空间上，在低维空间中同一类别样本尽可能靠近，不同类别样本尽可能彼此远离

考虑二元时，目标为最大化下式；分子表示不同类间距离，分母表示相同类内距离，其中 m 为当前（经投影过）聚类均值向量，s 为经过线性投影至一维空间后类别的协方差矩阵









Sb称为类间散度矩阵，Sw称为类内散度矩阵

因为都是w的二项式，所以最后仅与w的方向有关，而与其大小无关，可以使用拉格朗日乘子法进行求解

可以解得投影变量w为Sw为类内散度矩阵，m 为（不同类）聚类均值向量 --- 未经过投影



降维步骤如下:

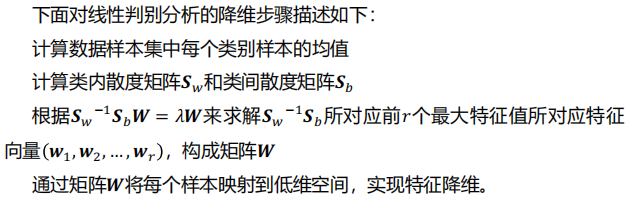
其中最低维度为min(K - 1 , d )，K为类别数（下标），d是向量有多长

特征值计算，计算矩阵 |A -α\* E| = 0，α就是所求的特征值，E为单位举证

Aw = αw ，其中 w 就是特征向量，α就是特征值，A就是目标矩阵；

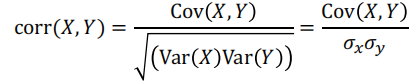
（A - α\* E）w = 0 可以解得特征向量，求 w 的非零解

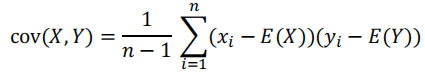
→ 也可以直接计算Sw-1Sb矩阵的最大 r 个特征值对映的特征向量，组合为投影矩阵

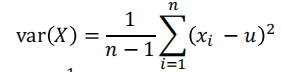


主成分分析（KT变换，KLT）--- 无监督学习

皮尔逊系数，系数等于1表示正线性相关，-1表示负线性相关，0表示线性无关；数值越大（正）表示则两者在线性相关的意义下相关程度越大

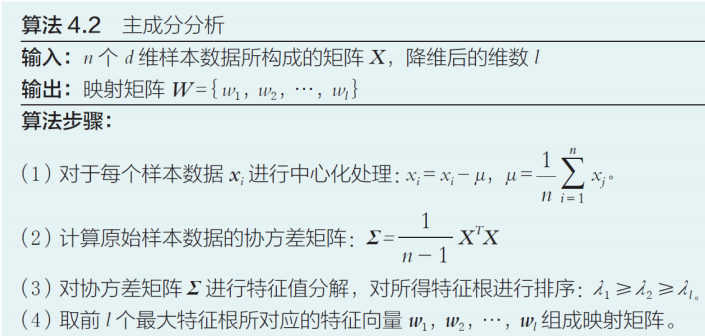




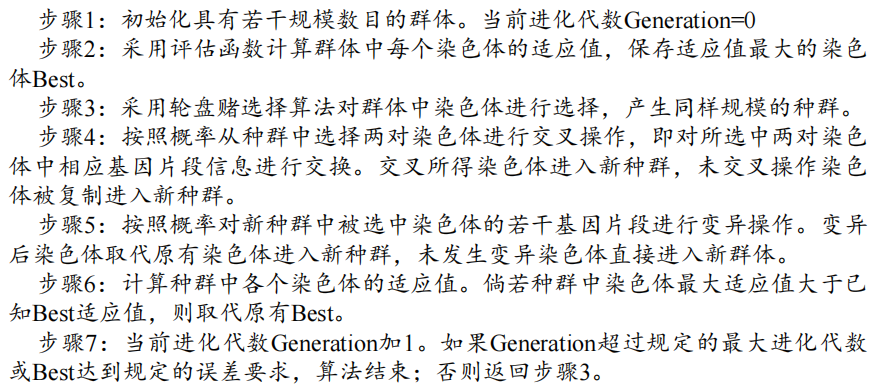


独立一定线性无关，但线性无关不一定独立 --- 可能是其他类型的相关

主成分分析希望将原始数据向数据方差最大的方向进行投影



演化算法 --- 遗传算法



特定的评估函数模仿选择过程（评估函数也是最终目标函数）；轮盘赌选择算法：适应度与被选择的概率成正比；Best只是记录适应值最大的染色体

CH5

MCP神经元（权重预先设定）--- 输出 1 / 0

二值输出

线性加权和（权重无法学习）

阈值函数（阈值无法学习）

感知机模型：只有输入层和输出层（含有激活函数） --- 共两层，输出1 / -1

已实现自动更新系数

误差后向传播

前向传播：网络输入输出一层层向后

计算误差并反向传播：链式求导，每一神经元对误差的贡献

多层感知机（前馈神经网络）

相邻层使用全连接

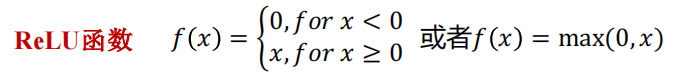
可以模拟复杂非线性函数功能，所模拟函数的复杂性取决于网络隐藏层数目和各层中神经元数目

非线性函数

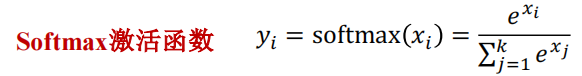


值域为( 0 , 1 )，输出可被视为概率值

导数恒小于1，易导致在使用反向传播算法更新参数过程中出现导数接近于0的情况（梯度消失，网络深度越深越严重）



X > 0，函数导数恒为1，有效防止梯度消失；x < 0，导数恒为 0，一定程度克服过拟合现象；若希望防止神经死亡（导数恒为0）的情况，可以使用leaky ReLU函数（x < 0 时也有恒定斜率但小于 1 ）

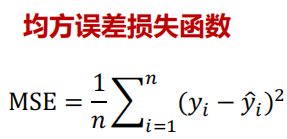


Softmax函数也被称为多项逻辑斯蒂回归模型(multi-nominal logistic mode)

神经网络参数优化是一个监督学习的过程

损失函数（代价函数）

均方误差损失函数



n表示训练数据个数

交叉熵损失函数



两个概率分布越接近，交叉熵越小，即预测类别分布概率与实际类别分布概率之间的差距越小，交叉熵就越小

不同模型性能比较时，一般也取交叉熵的平均值

梯度下降是一种使损失函数最小化的方法

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

每一步优化为“学习率 \* 偏导值”（使用链式求导法则，使用当前轮次计算得到的中间结果进行带入）

梯度下降法的分类：

批量梯度下降算法（batch gradient descent），在整个数据集上计算误差

随机梯度下降算法（stochastic gradient descent），使用每个数据更新参数（容易导致梯度方向有很大的波动，随机性可能跳出局部最优找到全局最优）

小批量梯度下降算法（mini-batch gradient descent），使用一小批量数据计算误差对参数进行优化（稳定而常用），往往是取误差均值

卷积神经网络

级联方式（逐层滤波）

卷积 <==> 滤波，卷积对图像进行了下采样（减少图像像素个数）操作，很好的运用了图片的“空间依赖度”

（一次）卷积滤波结果在卷积神经网络中被称为特征图（对于每一层/每一卷积核而言）

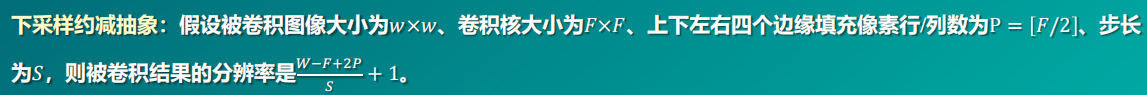
在高斯卷积核中，所有权重系数累加之值为1；卷积核中心位置权重系数的取值最大，从中心位置向四周边缘扩散过程中，权重系数的取值逐渐减少

若卷积核中心位置的权重越小（与其他权重系数越接近），则卷积后图像越模糊（图像平滑操作）；

Padding 边缘填充 0 像素点，防止下采样的发生

Striding 单步滑动是标准的卷积模式

卷积核大小常被称为“感受野”



Pooling（池化）

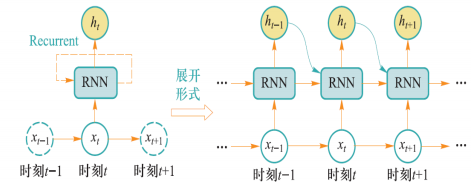
最大池化：从特征图的某个区域子块中选择值最大的像素点作为结果

平均池化：均值作为池化结果

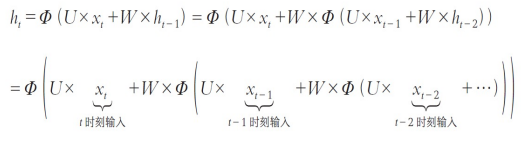
1. max池化：取前K个最大值作为池化结果

循环神经网络

处理序列数据



一个系数矩阵处理当前输入x，一个系数矩阵处理上一时刻输出h，具体算术式形式如下（每一重复结构参数复用）：



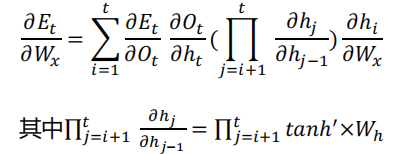
在处理自然语言时，有两种计算句子向量编码的方式（每个单词都有对映的隐式编码）

1.将最后一个单词的输出作为整个句子的向量编码

2.将每个单词的隐式编码进行加权平均，将加权平均结果作为句子的向量编码表示

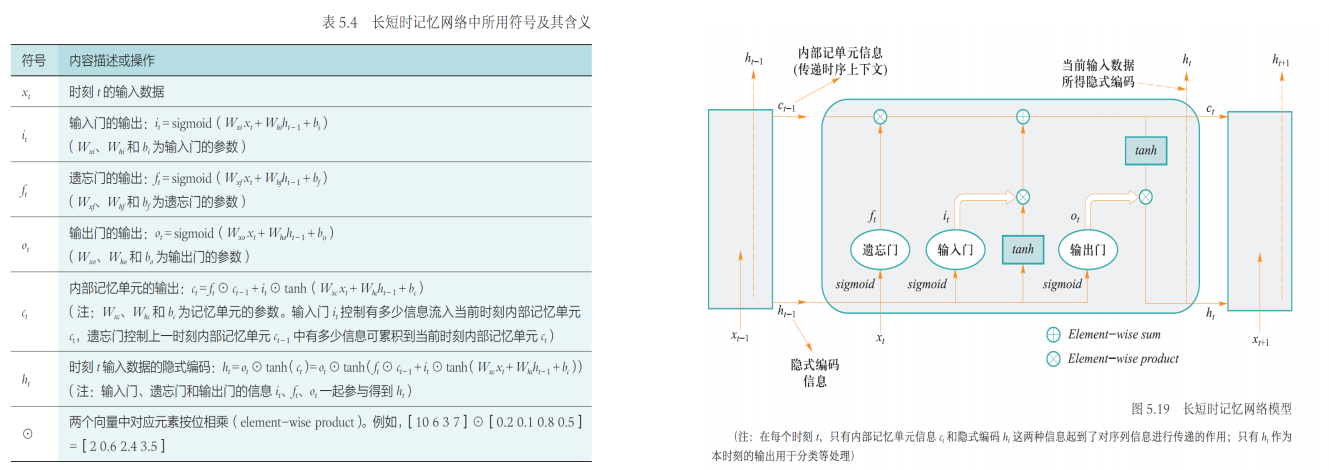
当输入序列过长时，循环神经网络也容易出现梯度消失（gradient vanishing） 或者梯度爆炸（gradient exploding）的问题

误差梯度回传的过程，（对Wx进行更新）考虑不同时刻时对Wx的利用，下列求导过程中，tanh’ 的导数值恒小于 1，连乘 tanh’ 容易导致梯度消失的问题。



长短记忆模型（LSTM）

输入门、遗忘门、输出门三种门结构



圆内一点（同圆内\*表示一致）表示“按位乘法”操作，

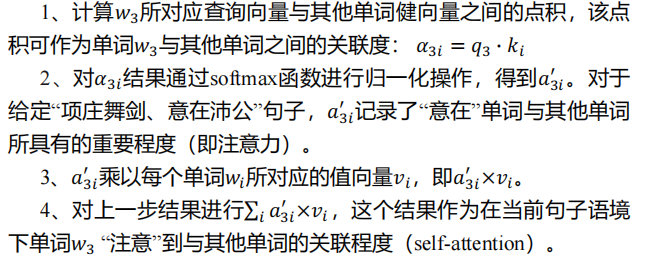
Tanh的值域为 ( -1 , 1 )，从而对映可正可负的操作

可以避免梯度消失的问题（是Ct对Ct-1求导，感觉很怪），梯度至少为遗忘门的输出，Ct表示一种长时记忆，ht表示一种短时记忆

Transformer（自注意力机制）

（在处理自然语言时）首先生成每个单词的内嵌向量（包含单词在句子中位置的编码向量信息）

多头注意力机制（增加厚度）



神经网络正则化

缓解神经网络在训练过程中可能产生的过拟合现象，正则化技术可以一定程度提升神经网络的泛化能力

Dropout：训练过程中随机丢掉一部分神经元来降低神经网络的复杂度（每次迭代训练中，一定概率屏蔽每一层某些神经元）

批归一化：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（可以使激活函数的映射保持较大的梯度水平）

损失函数中加上模型复杂度的惩罚项

L0范数：w中非0参数个数（NP hard问题，一般不使用）

L1范数：w中各元素绝对值之和（稀疏规则算子）

L2范数：w中各元素平方和开方

词向量生成

One-hot vector（词袋的形式，记录单词是否出现，出现把向量对映位置置1即可）

Word2vec词向量表现形式

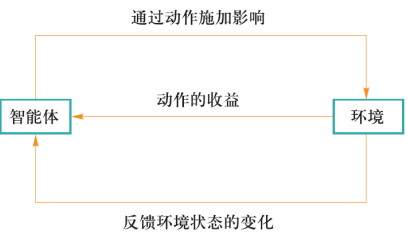
Continuous Bag-ofWords (CBoW)，根据某个单词所处上下文单词来预测该单词。

Skip-gram，利用某个单词来分别预测该单词的上下文单词。

在输入层 - 隐藏层 - 输出层架构中，输入层 - 隐藏层参数矩阵的第 k 列即为one-hot编码中第k个单词对映的词向量

CH6

以行为主义为核心的强化学习

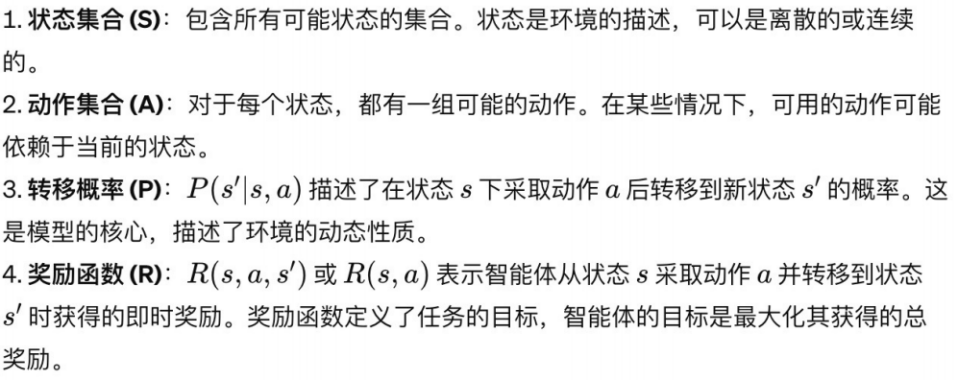


强化学习：智能体、环境、状态（智能体对环境的解读）、动作、策略、奖励



马尔可夫性质——下一个状态的概率分布仅依赖于当前状态和所采取的行动，与之前的历史状态或行动无关

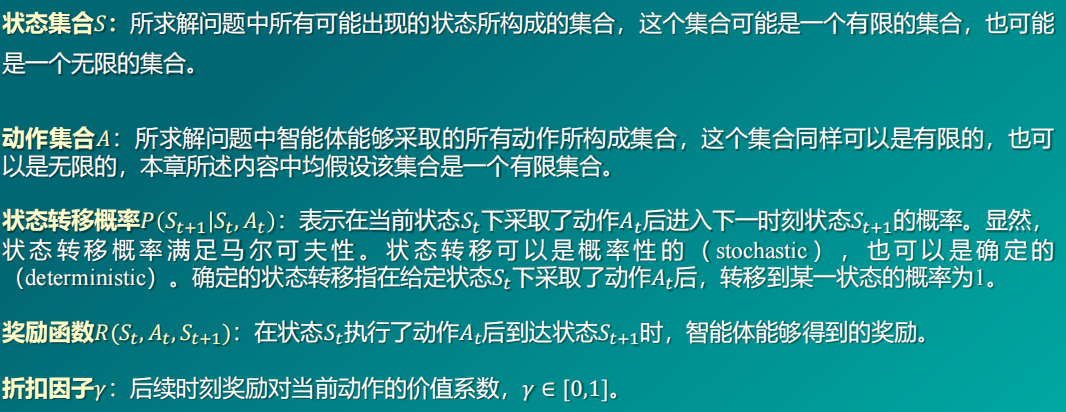
马尔可夫决策过程（MDP）



马尔可夫链（MP），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（当前状态概率计算只与前一时刻状态/当前时刻动作有关）--- 有状态和转移概率的概念

奖励是每一步动作可能获得的效益，回报反映了该时刻可得到的累加奖励加权和 --- 引入奖励函数和折扣因子的概念

马尔可夫奖励模型（MRP），在此基础之上，还可以引入动作的概念，从而得到MDP

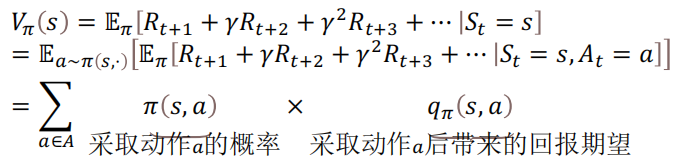


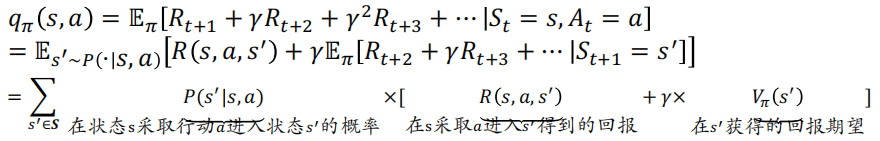
状态转移序列可以称为“轨迹”，轨迹长度可以无限也可以有终止状态；若包含终止状态，则可以称为分段问题（在分段问题中，一个从初始状态到终止状态的完整轨迹称为一个片段）；若不包含终止状态，则称为持续问题

价值函数：表示智能体在时刻t处于状态s时，按照策略π采取行动时所获得回报的期望（各类行动的回报期望）

动作-价值函数：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π采取行动所获得回报的期望（特定行动后的回报期望）。

贝尔曼方程（动态规划方程），价值函数与动作 - 价值函数间可以（递推）





基于价值的强化学习

通过策略计算价值函数的过程叫做策略评估（policy evaluation）

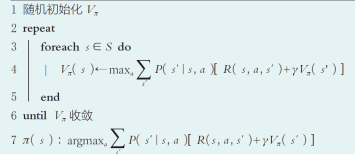
动态规划法：使用迭代的方法求解，但状态转移概率一般难以获得，无法处理状态集合大小无限的情况

蒙特卡洛采样法：根据当前策略采样若干轨迹，（当前数值计算）取均值更新价值函数（耗时过大，效率太低）

时序差分法：每一动作执行价值函数更新，且考虑保留部分原有价值函数，更新部分新价值函数（仅新价值函数项为时序差分目标，下式中α乘积项为时序差分偏差）

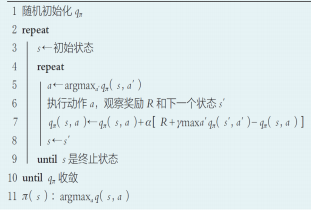


基于价值的强化学习算法（价值迭代算法）



策略收敛可能会早于价值收敛

Q学习算法

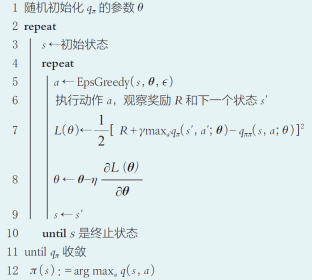


因为动作概率一般不可知，所以价值函数较难计算，从而Q学习算法（不同动作对映不同价值）较时序差分法使用了动作 - 价值函数（按照策略获取动作）

策略收敛可能早于价值收敛，学习过程受初始值影响

在选取动作时，考虑探索与利用的综合，一定的概率随机选择动作

当状态数量无限时，动作 - 价值函数无法详尽的保存，可以使用神经网络来进行参数的预测



采样不足，样本的相关性可能太强（来自于同一条轨迹），回归模型模拟时泛化能力变弱

难以收敛，因为当前状态和下一状态的估计值都使用了相同参数，参数的更新会影响两者的数值，导致优化失败

深度Q网络

经验重现方法，在算法运行的过程中将算法探索的片段以四元组的形式存储，当参数需要更新时取出，可以显著减弱样本之间的相关性；

使用不同的参数来进行动作 - 价值函数的预测且更新频率较低，可以使参数更新过程更稳定

通过价值函数优化策略的过程叫做策略优化（policy improvement）

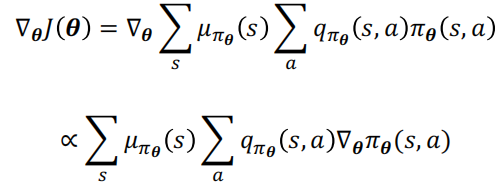
通过价值计算选择更优的路径

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）

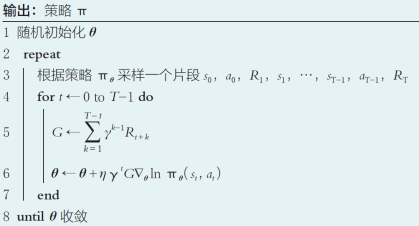
基于策略的强化学习

策略梯度法

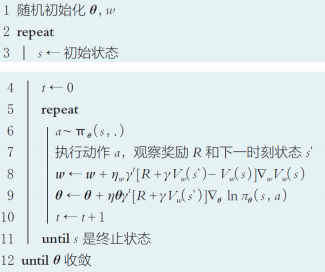
需要求取s0状态处最大的价值函数收益，可以使用梯度递增进行求解



REINFORCE策略梯度算法（类似于蒙特卡洛，只有在一条轨迹采样结束后，才能更新参数）



Actor-Critic算法（actor执行者，负责挑选动作；critic评论家，负责计算动作价值）



其中V() 函数为价值函数，π（s, a）函数为特定状态s下动作a发生的概率函数

CH7

以博弈对抗为核心的决策智能

竞争博弈中的博弈行为是多个带有相互竞争性质的主体，为了达到各自目标和利益，采取的带有对抗性质的行为

从“求取最优解”到“求取均衡解”的转变

博弈部分概念：

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）

如果参与者每次行动前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）

混合策略下的收益称为期望收益

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（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（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

纳什均衡

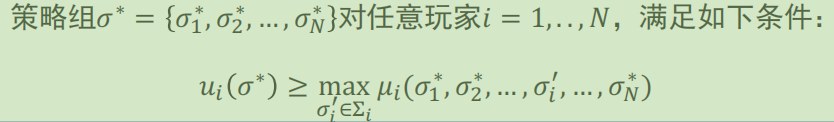
参与者所作出的这样⼀种策略组合，在该策略组合上，任何参与者单独改变策略都不会得到好处

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。

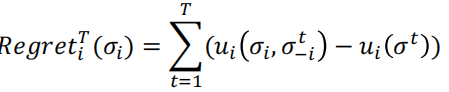
策梅洛定理（Zermelo's theorem）：对于任意⼀个**有限步**的**双⼈完全信息零和动态博弈**，⼀定存在先手必胜策略或后手必胜策略或双方保平策略

最优反应策略，当其他参与者策略不变时，自身可获得最大收益的反映策略

若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，那么该策略组合是一个纳什均衡（Nash equilibrium）策略。在有限对手、有限策略情况下，纳什均衡一定存在



遗憾值最小化算法（对于每次独立的决策）

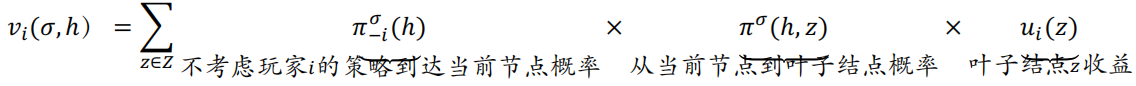


T表示轮次次数，减法前项表示特定的一个（未采取的）策略可能带来收益值，后项表示当前选择策略带来的收益值

根据不同策略的遗憾值，我们可以使用遗憾匹配来选择下一轮策略（遗憾值若为负则取数值0）（策略选择概率可与遗憾值成正比，可取最大遗憾值，可随机），若不同动作遗憾值的总和不大于0，则直接使用随机策略。

虚拟遗憾值最小化算法（对于序贯决策而言）

在特定策略组合下，对玩家i而言，如下计算从根节点到当前节点的行动序列路径h的虚拟价值：



不考虑玩家i的策略是指玩家i每次都作出概率为 1 的动作，使得行动序列路径可以顺利的到达当前节点

遗憾值计算的是玩家i通过行动序列路径h到达当前节点后，采取不同动作带来的预期收益之差；需要对所有能到达同一个信息集I的所有行动序列遗憾值进行累加；其余如，遗憾值之和、遗憾匹配过程等同“每次独立决策”计算

遗憾值计算如下（注意减法方向，减法后项是期望收益，减法前项是特定动作收益）：



安全子博弈：从当前已经完成的部分博弈出发，将接下来博弈过程视为是一个单独子博弈（只对完全信息博弈有效）；且在子博弈的求解过程中，得到的结果⼀定不差于全局的近似解法

双边匹配问题（稳定婚姻问题）

男性先（向未被拒绝过的女性对象）选择，女性被选择后再反选（可以更换）男性，直到全部匹配完成

单边匹配问题（选寝室床位）

分配的对象都是不可分的标志物，他们只能属于一个所有者，且可以属于任何一个所有者

最大交易圈算法

