产生式系统的搜索

状态空间法(S,F,G)(初始状态 操作符 目标状态)
如食人生番:动作 op(b1,b2,b3)(人数,野人数,哪岸到哪岸)
状态(a1,a2,a3)(左岸人 num,左岸野人 num,船在左 or 右)
状态转换: $A \to B$, $OPERATOR$(操作对象,初值,终值)
搜索:宽/深:后代放进 open-后代是否目标-取点放进 closed
一致代价搜索: Dijkstra 算法。
贪婪最优搜索:仅利用启发函数作为代价。
A*算法: $f(n) = h(n) + g(n)$
最优性:可采纳:树搜索的 A*,一致:图搜索的 A*
可采纳性: $h(n) \le h^*(n)$;一致性: $h(n) \le c(n,n') + h(n')$
/深度优先: $h(n)=0, g(n)=d(n)$
/一致代价 $h(n)=0, f(n)=g(n)+0$
/宽度优先: $h(n)$ 足够大, $g(n)=0$, $f(n)=h(n)$
计算复杂性 $f(x) = O(g(x)) \Leftrightarrow \exists A,B>0, A \le |f/g| \le B$
P 类问题:有多项式时间算法解决的**判定问题**(有 $P \subseteq NP$)
NP 类问题:对猜想存在多项式时间算法来验证的判定问题,非确定性多项式问题
NP 完全问题:(NP 中最难的问题)判定问题 $P_1 \in NP$,对所有其他判定问题 $P_m \in NP$,有 $P_m \propto P_1$,则 P_1 是 NP 完全的。
NP 难题: $P_1 \propto P_2, P_1 \in NP$完全$\Rightarrow P_2$是 NP 难的。$P_2$ 至少和 P_1 一样难。如 TSP 问题不是判定问题,但它是 NP 难题。
约束满足问题和搜索:
约束满足问题:变量 X、值域 D、约束 C。
回溯搜索:一次赋值一个变量,合法时从该状态继续为下一变量赋值,没有合法值则返回父节点尝试新赋值/选择顺序:最少剩余值、最多约束项、最少约束值
推理:节点相容(变量在其值域中的所有取值均满足其一元约束)、边相容(变量在其值域中的所有取值均满足与该变量相关的二元约束)、k-相容、路径相容。
搜索后推理:在变量赋值后对它进行边相容检查,删除与该变量相关的未赋值变量不相容的值,冲突时回溯。
局部搜索:爬山法:从初始状态开始,不断向 h(n)函数值增加的状态移动/随机梯度下降:一次用一个样例计算梯度
博弈树搜索:画与或图,我方或、MAX,对方与、MIN
与或图:注意弧线!有弧线是与,没弧线是或。
叶节点一定可解;非叶节点:若含有或(与)后继点,那么至少一个(全部)后继可解时,该非叶节点可解。
非叶节点无后继不可解/或(与)后继全不(有一个)可解。
极大极小搜索:我方取后继的 MAX,对方取后继的 MIN。
$\alpha-\beta$ **剪枝**:用深搜。α 是 MAX 的下界;β 是 MIN 的上界。
α: β(后继层)$\le \alpha$(先辈层),中止该 MIN 层后续搜索;
β: α(后继层)$\ge \beta$(先辈层),中止该 MAX 层后续搜索。
效率:设深度为 P,每节点有 B 个后继,生成端节点最小值 N_P。p 为偶数: $N_p = 2B^{p/2} - 1$;p 为奇数: $N_p = B^{(p+1)/2} + B^{(p-1)/2} - 1$
蒙特卡洛树搜索:权重 $I_j = \bar{v}_j + c\sqrt{\ln N / n_j}$ (N 是总访问次数)
n_j、$\bar{v}_j$ 算法开始到现在,节点 j 被访问的次数和平均收益
流程:选择-拓展-模拟-反向传播

谓词逻辑与归结原理:命题是具有唯一真值的陈述句。

2.∧:合取(与) ∨:析取(或) →:蕴含 ↔:等价
3. $p \to q$ 为假当且仅当 $p=T, q=F$
4.永真/重言式:真;永假/矛盾式:假;可满足式:至少一个成真赋值;**非重言式的可满足式**:至少一个成真一个成假
5.吸收律: $A \vee (A \wedge B) \Leftrightarrow A$; $A \wedge (A \vee B) \Leftrightarrow A$
$$A \vee (B \wedge C) \Leftrightarrow (A \vee B) \wedge (A \vee C)$$
$$A \wedge (B \vee C) \Leftrightarrow (A \wedge B) \vee (A \wedge C)$$
$$\sim(A \vee B) = \sim B \wedge \sim A, \sim(A \wedge B) = \sim B \vee \sim A$$
重要: $A \to B \Leftrightarrow \sim A \vee B, (A \to B) \wedge (A \to \sim B) \Leftrightarrow \sim A$
$$A \leftrightarrow B \Leftrightarrow (A \to B) \wedge (B \to A)$$
$$A \to B \Leftrightarrow \sim B \to \sim A, A \leftrightarrow B \Leftrightarrow \sim A \leftrightarrow \sim B$$
6.原子公式:不含任何联结词的公式//子句:任何文字的析取式;文字:原子或~原子//合取**范式**:简单析取式构成的合取式//析取范式:简单合取式构成的析取式
命题逻辑的推理规则
7.附加: $A \Rightarrow (A \vee B)$;假言推理: $((A \to B) \wedge A) \Rightarrow B$
简化: $(A \wedge B) \Rightarrow A$;拒取式: $((A \to B) \wedge \sim B) \Rightarrow \sim A$
析取三段论: $((A \vee B) \wedge \sim A) \Rightarrow B$
假言三段论: $((A \to B) \wedge (B \to C)) \Rightarrow (A \to C)$
等价三段论: $((A \leftrightarrow B) \wedge (B \leftrightarrow C)) \Rightarrow (A \leftrightarrow C)$
构造性二难: $(A \to B) \vee (C \to D) \wedge (A \vee C) \Rightarrow (B \vee D)$
可做:前提引入/结论引入/置换规则比如 $\sim p \vee q$ 换 $p \to q$
命题逻辑的归结方法
8.子句:简单析取式,项是一个变量或者其否定。
子句集:合取范式中所有子句的集合,∧换成“,”。
9.归结原理:如 $p \vee q$ 和 $\sim p \vee r$ 都为真,那么 $q \vee r$ 为真。
10.归结式: $C_1 = P \vee C_1', C_2 = \sim P \vee C_2'$ 归结 $C_{12} = C_1' \vee C_2'$
11.**归结(消解)法**:为了证明 $A \to B$ 真,先转为 $A \wedge \sim B$,证明该命题公式永假。提取其子句集,归结为空即可。
谓词逻辑 12.个体词/谓词 任意/存在量词 约束出现/自由出现//函数是个体域到个体域的映射,不同于谓词
13.换名:辖域中约束出现的变量名换掉/替代:自由出现的个体变量名字可换掉
14.谓词公式永真称为逻辑有效/永真的,谓词公式永假成为不一致/不可满足。
15.**谓词演算公式** $\sim(\forall x)P(x) \Leftrightarrow (\exists y)\sim P(y)$
$$\sim(\exists x)P(x) \Leftrightarrow (\forall y)\sim P(y)$$
$$(\forall x)(P(x) \wedge Q(x)) \Leftrightarrow (\forall x)P(x) \wedge (\forall x)Q(x)$$
$$(\exists x)(P(x) \vee Q(x)) \Leftrightarrow (\exists x)P(x) \vee (\exists x)Q(x)$$
$$(\forall x)(P(x) \to Q) \Leftrightarrow (\exists x)P(x) \to Q$$
$$(\forall x)(Q \to P(x)) \Leftrightarrow Q \to (\forall x)P(x)$$
$$(\exists x)(P(x) \to Q) \Leftrightarrow (\forall x)P(x) \to Q$$
$$(\exists x)(Q \to P(x)) \Leftrightarrow Q \to (\exists x)P(x)$$
注意: $(\exists x)(P(x) \wedge Q(x)) \ne (\exists x)P(x) \wedge (\exists x)Q(x)$
$(\forall x)(P(x) \vee Q(x)) \ne (\forall x)P(x) \wedge (\exists x)Q(x)$
16.**前束范式**:将量词均提到最左边
17.谓词推理:存在用常量替代,任意用变量 or 常量替代
谓词逻辑归结原理
18.skolem 标准型:变前束范式;消去存在量词;略去全称量词例子: $\forall x \exists y P(x,y) \to \forall x P(x, f(x))$ | $\exists x\, f(x) \to f(c)$
19.**置换**: $\theta = \{t_1/a_1, \ldots\}$,用 t_1 代替 a_1
结合律 $(\theta \cdot \lambda_1)\lambda_2 = \theta(\lambda_1 \cdot \lambda_2)$,不满足交换律。
20.合一: $F = \{F_1, F_2, \ldots\}$,存在置换θ,使 $F_1\theta = F_2\theta \cdots$。
21.**最一般合一**:α是 F 的最一般合一置换,那 F 的任一合一θ,都有置换λ,有 $\theta = \alpha\lambda$ (从左比较对应项不同就换)
22.**归结过程**:归谬-化合取范式-子句集-归结-空子句。

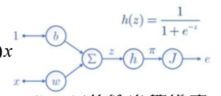
监督学习:线性回归

线性回归: $y = \omega x + b$ 找函数族/找优化准则/找最优函数
$S_{xx} = \sum (x_i - \bar{x})^2, S_{xy} = \sum (x_i - \bar{x})(y_i - \bar{y}), S_{yy} = \sum (y_i - \bar{y})^2$
评价: $MAE = \frac{1}{n} = \sum |y_i - (w_i + b)|$, MSE(改平方)
最小二乘: $\hat{\omega} = \frac{S_{xy}}{S_{xx}} = \frac{\sum x_i y_i - n\bar{x}\bar{y}}{\sum x_i^2 - n\bar{x}^2} = \frac{S_{xY}}{S_{xx}}, \hat{b} = \bar{y} - \hat{\omega}\bar{x} = \bar{Y} - \hat{\omega}\bar{x}$

最大似然估计: $Y_i = \omega x_i + b + \varepsilon_i, Y_i | x_i \sim N(\omega x_i + b, \sigma^2)$
概率密度函数: $f(y_i | \omega, b, x_i) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{[y_i - (\omega x_i + b)]^2}{2\sigma^2}\right\}$
$$l = \log L = \log \prod_{i=1}^{n} f = -\frac{n\log(2\pi)}{2} - n\log(\sigma) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}[y_i - (\omega x_i + b)]^2$$
去找使似然函数值 l 最大的参数值,结果同最小二乘法。
参数 $\hat{\omega} == \frac{S_{xY}}{S_{xx}} = \sum \frac{x_i - \bar{x}}{S_{xx}} Y_i$ $\hat{\omega} \sim N\left(\omega, \frac{\sigma^2}{S_{xx}}\right), \frac{\hat{\omega} - \omega}{\sqrt{\sigma^2 / S_{xx}}} \sim N(0,1)$
$$\hat{b} = \bar{Y} - \hat{\omega}\bar{x} \sim N\left(b, \sigma^2/nS_{xx}\sum x_i^2\right), \quad \varepsilon_i \sim N(0, \sigma^2)$$
假设检验: $H_0: \omega = 0 \text{ vs } H_1: \omega \ne 0$, $p = 2\Phi\left(|\hat{\omega}| / \sqrt{\sigma^2 / S_{xx}}\right)$
P 小 H1 真,检验越合理。$\sum (y_i - \bar{y})^2 = \sum (\hat{y}_i - \bar{y})^2 + \sum (y_i - \hat{y}_i)^2$
确定系数: $r^2 = S_{xy}^2/(S_{xx}S_{yy})$,越接近 1 拟合效果越好
多项式回归: $y = \omega_0 + \omega_1 x + \omega_2 x^2 + \cdots + \omega_m x^m, y = Xw$
多元回归: $y = \omega_0 + \omega_1 x_1 + \omega_2 x_2 + \cdots + \omega_m x_m$, $y = Xw$
$$y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}, X\begin{pmatrix} 1 & & \cdots & x_{1m} \\ 1 & x_{21} & & x_{2m} \\ & & \ddots & \\ 1 & x_{n1} & & x_{mm} \end{pmatrix}, W = \begin{pmatrix} \omega_0 \\ \vdots \\ \omega_n \end{pmatrix},$$
$$\min (y - Xw)^T (y - Xw) \Rightarrow w = (X^T X)^{-1} X^T y$$

监督学习:Logistic 回归

L 回归: $y = \pi(x) = \frac{1}{1 + \exp\{-(wx+b)\}}, \pi'(x) = w\pi(x)(1 - \pi(x))$
$P(Y=1) = 1/(1 + \exp\{-(wx+b)\}) = \pi(x), P(Y=0) = 1 - \pi(x)$
$\log \frac{P(Y=1)}{1 - P(Y=1)} = wx + b$, $e^w = \frac{\pi(x+1)/(1-\pi(x+1))}{\pi(x)/(1-\pi(x))}$
对数似然: $L(w,b | x, y) = f(y | w,b,x) = \prod_{i=1}^{n} \pi^{y_i}(x_i)(1 - \pi(x_i))^{1-y_i}$
$l(w,b | x,y) = \sum_{i=1}^{n} -H(p_i, q_i)$,需要 $\max \sum_{i=1}^{n} -H(p_i, q_i)$
经验损失: $J(w,b) = \frac{1}{n}\sum_{i=1}^{n} H(p_i, q_i)$,需要 $\min \frac{1}{n}\sum_{i=1}^{n} H(p_i, q_i)$
交叉熵 $H(p,q) = -y_i \log \pi(x_i) - (1 - y_i)\log(1 - \pi(x_i))$
梯度下降找最小: $x^{new} \leftarrow x - \alpha f'(x)$
$\frac{de}{db} = \frac{de}{d\pi}\frac{d\pi}{dz}\frac{dz}{db} = \pi - y, \frac{de}{dw} = (\pi - y)x$



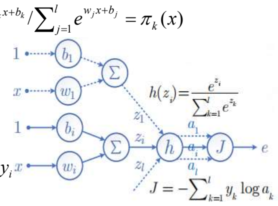
$\frac{de}{d\pi} = -\frac{y}{\pi} + \frac{1-y}{1-\pi}, \frac{d\pi}{dz} = \frac{1}{1+e^{-z}}\frac{e^{-z}}{1+e^{-z}} = \pi(1-\pi)$ |从输出算梯度

二分类的评价
T/F:True / False ; P / N : Positive / Negative; R : Rate
TPR= TP / (TP + FN);FPR= FP / (TN + FP);TNR= TN / (TN + FP)
FNR= FN / (TP + FN);敏感Sensitivity = TPR; 1-Sensitivity = FNR
特异Specificity = TNR; 1-Specificity = FPR;Fall-out=FP/(TP+FP)
精度 Precision = TP / (TP + FP);召回率Recall = TP / (TP + FN);
F1-score = 2TP / (2TP+FP+FN) = 2*P* R/ (P+R)||P/R:精度/召回
Accuracy= (TP+TN) / (TP+TN+FN+TN);
BER=1/2(FPR+FNR);BCR=1/2(TPR+TNR)
$MCC: (TP \cdot TN - FP \cdot FN) / \sqrt{(TP+FP)(FP+TN)(TN+FN)(FN+TP)}$

混淆矩阵		真实结果	
		Positive	Negative
预测结果	Claim Posi	TP	FP
	Claim Nega	FN	TN

ROC: 横(假阳性率)1 - specifi, 纵(真阳性率)Sensitity
AUC: $\int$ROC,近1好,近0.5差

softmax 回归: $P(Y=k) = e^{w_k x + b_k} / \sum_{j=1}^{l} e^{w_j x + b_j} = \pi_k(x)$
$\max \sum_{i=1}^{n} \sum_{k=1}^{l} y_{ik} \log \pi_k(x_i)$
$\min -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{l} y_{ik} \log \pi_k(x_i)$
$\frac{de}{db_i} = -\sum_{k=1}^{l} \frac{\partial e}{\partial \pi_k}\frac{\partial \pi_k}{\partial z_i}\frac{\partial z_i}{\partial b_i} = \pi_i - y_i$



$\frac{de}{dw_i} = (\pi_i - y_i)x, \frac{d\pi_k}{dz_i} = \begin{cases} e^{z_i}, k = i \\ 0, k \ne i \end{cases}$ $\frac{\partial e}{\partial \pi_k} = -\frac{y_k}{\pi_k}, \frac{\partial z_i}{\partial b_i} = 1, \frac{\partial z_i}{\partial w_i} = x$
一元变多元: $wx + b \leftarrow \sum_{j=1}^{m} w_j x_j, w_k x + b_k \leftarrow \sum_{j=1}^{m} w_{jk} x_j$

前馈神经网络:输入单元—隐层单元—输出单元

线性单元: $x = (x_1, x_2, \cdots x_m)^T, w = (w_1, w_2, \cdots w_m)^T$
$z = w^T x + b$, $a = z$, $\frac{da}{db} = \frac{da}{dz}\frac{\partial z}{\partial b} = 1 \cdot 1, \frac{da}{dw} = \frac{da}{dz}\frac{\partial z}{\partial w} = 1 \cdot x^T$
Logistic 单元: $z = w^T x + b$, $a = 1/(1 + e^{-z})$
$\frac{\partial a}{\partial b} = \frac{da}{dz}\frac{\partial z}{\partial b} = a(1-a), \frac{\partial a}{\partial w} = \frac{da}{dz}\frac{\partial z}{\partial w} = a(1-a)x^T$
Softmax: $w_i = (w_{i1}, \cdots w_{im})^T, W = (w_1, \cdots w_m)$, $b = (b_1, \cdots, b_m)^T$
$z = Wx + b$, $z_i = W_{[i\cdot]}x + b_i$, $a = \frac{\exp(z)}{\|\exp(z)\|_1}, a_i = \frac{e^{z_i}}{\sum_{k=1}^{l} e^{z_k}}, e = -y^T \log a$
$$J_{ew} = \left(\frac{\partial e}{\partial a_i}\right)_{1*l}\left(\frac{\partial a_i}{\partial z_i}\right)_{l*l}\left(\frac{\partial z_i}{\partial w_{jk}}\right)_{l*(l*m)} = \left(\frac{\partial e}{\partial w_{jk}}\right)_{l*(l*m)}, J_{eb} = \left(\frac{\partial e}{\partial b_j}\right)_{1*l}$$

双曲正切单元： $a=(e^{z}-e^{-z})/(e^{z}+e^{-z})$， $da/db=1-a^2$

整流线性单元(ReLU)： $a=\max(0,z), da/dz=1, z>0; 0, \text{other}$

梯度消失问题略优。其他隐层单元：softplus $a=\log(1+e^{z})$

输入单元—隐层单元(一般 ReLU)—输出单元(按需选)

网络结构设计:可表示区域的数量是深度的指数函数(d 深度,每个隐层 k 单元 m 输入)：$O\left(k^{m}\left((k \quad m)^{T}\right)^{m(d-1)}\right)$

随机梯度下降:一次用一个样例更新梯度(SGD)

动量梯度下降位移导是速度$v(t)=\frac{\partial}{\partial t}d(t)$，速度导是受力$f(t)=\frac{\partial}{\partial t}v(t)$，将位移类比参数$\Delta\theta\propto \mathrm{v}$,受力类比负梯度$\Delta \mathrm{v}\propto-\nabla_{\theta}$。参数更新公式:$\mathrm{v}\leftarrow\alpha\mathrm{v}-\varepsilon\nabla_{\theta}$，$\theta\leftarrow\theta+\mathrm{v}$

Nesterov 动量：$\tilde{\theta}\leftarrow\theta+\alpha\mathrm{v}, \mathrm{v}\leftarrow\alpha\mathrm{v}-\varepsilon\nabla_{\theta}, \theta\leftarrow\theta+\mathrm{v}$

参数初始化策略

破坏对称性:与同一个输入相连的具有相同激活函数的隐层单元参数不同；随机初始化:权重随机产生,偏置设为零

学习速率调整方法:学习速率反比于梯度累计平方和的根

线性衰减:$\alpha_k=\begin{cases}\lambda\alpha_{\tau}+(1-\lambda)\alpha_0, k\le\tau\\ \alpha_{\tau}, k>\tau\end{cases}$，$\lambda=\frac{k}{\tau}$ //自适应衰减:

AdaGrad (Duchi et al., 2011)

$\mathbf{r}\leftarrow\mathbf{r}+\mathbf{g}\odot\mathbf{g}$

$\boldsymbol{\theta}\leftarrow\boldsymbol{\theta}-\frac{\epsilon}{\delta+\sqrt{\mathbf{r}}}\odot\mathbf{g}$

RMSProp (Hinton, 2012)

$\mathbf{r}\leftarrow\rho\mathbf{r}+(1-\rho)\mathbf{g}\odot\mathbf{g}$

$\boldsymbol{\theta}\leftarrow\boldsymbol{\theta}-\frac{\epsilon}{\delta+\sqrt{\mathbf{r}}}\odot\mathbf{g}$

Adam (Kingma and Ba, 20

$\mathbf{s}\leftarrow\rho_1\mathbf{s}+(1-\rho_1)\mathbf{g}$

$\mathbf{r}\leftarrow\rho_2\mathbf{r}+(1-\rho_2)\mathbf{g}\odot\mathbf{g}$

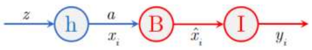
$\hat{\mathbf{s}}\leftarrow\frac{\mathbf{s}}{1-\rho_1^k}$

$\hat{\mathbf{r}}\leftarrow\frac{\mathbf{r}}{1-\rho_1^k}$

$\boldsymbol{\theta}\leftarrow\boldsymbol{\theta}-\frac{\epsilon}{\delta+\sqrt{\hat{\mathbf{r}}}}\odot\hat{\mathbf{s}}$

正则化:增加数据/提前终止/修改目标函数加入惩罚项
简化网络结构:随机扔掉一些节点/连接

批次标准化:SGD 过程中对隐层输出标准化,可以用较大学习率/提升训练速度/收敛快/性能好(类似扔节点的正则化方法)/初始参数要求低 $\mu_B\leftarrow 1/m\sum x_i$，$\sigma_B^2\leftarrow 1/m\sum(x_i-\mu_B)^2$



$\hat{x}_i\leftarrow(x_i-\mu_B)/\sqrt{\sigma_B^2+\varepsilon}$

$y_i\leftarrow\gamma\hat{x}_i+\beta\equiv \mathrm{BN}_{\gamma,\beta}(x_i)$

卷积神经网络

卷积:填充:允许卷积核超出边界(Samepadding)如边缘补 0。

稀疏连接:可以用全连接的前馈网络实现,不参与卷积的点权重为 0。节点 N，卷积核 m*m，连接$O(m^2N)$

参数共享:在不同位置卷积时像素点一样则结果一样(等变表示)，局部特征为制不重要，$O(m^2)$。

非共享卷积核/平铺卷积核:不共享参数/不完全共享参数

1*1 卷积核可以降维/多个卷积核升维/减少计算量

池化:用某一位置相邻输出的统计特征代替该位置的输出。

池化函数:最大池化/平均池化/随机池化

步长大于 1 的池化能够**降低输入规模**。

卷积神经网络典型结构:卷积层、池化层叠起来，再由全连接网络完成相关学习。卷积功能:特征提取—端到端学习。

训练过程:前向/后向,关键是池化和卷积单元的梯度计算。

数据的扩增:原图抽取小图、原图进行几何变换等。

图卷积模型:$x_v^{(con)}\leftarrow w_v x_v+\sum_{u\in u}w_u x_u$,u是邻居

强化学习

马尔可夫过程：二元组(S,P)　状态空间$S=\{s_1,\cdots,s_n\}$

状态转移矩阵$P=(p_{ss'})_{n\times n}$，$p_{ss'}=P(S_{t+1}=s'|S_t=s)$

马尔可夫回报过程：四元组(S,P,r,γ)

状态转移$S_t\to S_{t+1}$产生回报R_{t+1}。折现因子$\gamma\in[0,1]$;期望状态回报$\mathrm{r}=\{r_1,r_2,\cdots,r_n\}$,$\mathrm{r_s}=E[R_{t+1}|S_t=s]=\sum_{r\in R}rp(r|S_t=s)$

累计回报:$G_t=R_{t+1}+\gamma R_{t+2}+\gamma^2R_{t+3}+\cdots=\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}$

状态价值函数:$v(s)=E[G_t|S_t=s]=r_s+\gamma\sum_{s'\in S}p_{ss'}v(s')$

贝尔曼期望方程:$\mathrm{v}=\mathrm{r}+\gamma\mathrm{P}\mathrm{v}$，$\mathrm{v}=(\mathrm{I}-\gamma\mathrm{P})^{-1}\mathrm{r}$

马尔可夫决策过程:五元组(S,A,P,R,γ)

行动A_t导致状态转移$S_t\to S_{t+1}$产生回报R_{t+1}行动空间:$A=\{a_1,a_2,\cdots,a_m\}$；状态转移矩阵:$P=(p_{ss'}^a)_{n\times n\times m}$其中$p_{ss'}^a=P(S_{t+1}=s'|S_t=s,A_t=a)=\sum_{r\in R}p(s',r|s,a)$；行动期望回报$\mathrm{R}=(r_s^a)_{n\times m}$，$\mathrm{r_s^a}=E[R_{t+1}|S_t=s,A_t=a]=\sum_{r\in R}r\sum_{s'\in S}p(s',r|s,a)$

策略:$\pi(a|s)=P(A_t=a|S_t=s)$,s 下选策略 a 的概率.

$p^{\pi}(S_{t+1}=s'|S_t=s)=\sum_{a\in A}\pi(a|s)p_{ss'}^a$，$r_s^{\pi}=\sum_{a\in A}\pi(a|s)r_s^a$

状态价值:$v_{\pi}(s)=E[G_t|S_t=s]=\sum_{a\in A}\pi(a|s)q_{\pi}(s,a)$

行动价值:由后续状态价值的加权和计算。

$q_{\pi}(s,a)=E[G_t|S_t=s,A_t=a]=r_s^a+\gamma\sum_{s'\in S}p_{ss'}^a v_{\pi}(s')$

贝..期望方程:$v_{\pi}(s)=\sum_{a\in A}\pi(a|s)\left(r_s^a+\gamma\sum_{s'\in S}p_{ss'}^a v_{\pi}(s')\right)$

$q_{\pi}(s,a)=r_s^a+\gamma\sum_{s'\in S}p_{ss'}^a\sum_{a'\in A}\pi(a'|s')q_{\pi}(s',a')$

矩阵形式:$\mathrm{v}=\mathrm{r}^{\pi}+\gamma\mathrm{P}^{\pi}\mathrm{v}$，$\mathrm{v}=(\mathrm{I}-\gamma\mathrm{P}^{\pi})^{-1}\mathrm{r}^{\pi}$

动态规划—策略评价(状态价值计算)、策略改进

$\mathrm{v}^{(k+1)}=\mathrm{r}^{\pi}+\gamma\mathrm{P}^{\pi}\mathrm{v}^{(k)}$，$v_{k+1}(s)=r_s^{\pi}+\gamma\sum_{s'\in S}p_{ss'}^{\pi}v_k(s')$

最优状态价值:$v_*(s)=\max_{\pi}v_{\pi}(s)=v_{\pi_*}(s)$

最优行动价值:$q_*(s,a)=\max_{\pi}q_{\pi}(s,a)=q_{\pi_*}(s,a)$

最优 a 价值是同时刻最优 q 价值$v_*(s)=\max_{a\in A}q_*(s,a)$

最优策略(**贪心策略**):$\pi_*(a|s)=1, \text{if} a=\arg\max q_*(s,a)$

策略迭代：已知策略 pi，用动态规划计算出各个行动下的状态价值 v(直至收敛，或者算一步改进一次)，在某状态可行的所有策略中选择使得状态价值最大的那个（贪心策略），然后重复以上过程，得到一列状态、策略。$v_{k+1}(s)=\max_{a\in A}\left(r_s^{\pi}+\gamma\sum_{s'\in S}p_{ss'}^{\pi}v_k(s')\right)$

同步迭代:算完所有 s 更新一次策略。(不用算 v 至收敛)

异步迭代:算完一个 s 就更新一次策略。

蒙特卡洛—无 P 矩阵时状态价值预测、策略改进

已知观测片段，对某个状态s，以其出现作为开始计算出这些幕各自的G_i，对这些G_i取平均值来估计 v(s)。$V(S_t)=1/n\sum G_i$,其中$G_i=R_{t+1}+\gamma R_{t+2}+\cdots+\gamma^{T-t}R_{T+1}$

首次访问/每次访问:作为起始的 s 是否第一次出现。

增量式蒙特卡洛:$V(S_t)\leftarrow V(S_t)+1/k(G_t-V(S_t)), k\leftarrow k+1$

定步长蒙特卡洛:$V(S_t)\leftarrow V(S_t)+\alpha(G_t-V(S_t))$

无模型时预测行动价值，$q_{\pi}(s,a)\approx 1/n\sum G_i$

增量式:$Q(S_t,A_t)\leftarrow Q(S_t,A_t)+1/k(G_t-Q(S_t,A_t)), k\leftarrow k+1$

定步长:$Q(S_t,A_t)\leftarrow Q(S_t,A_t)+\alpha(G_t-Q(S_t,A_t))$

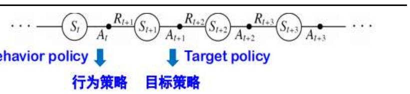
策略改进:$\pi'(S_t)=\arg\max_{a\in A}Q(S_t,a)$

贪心改进:$\pi_*(a|s)=\begin{cases}1-\varepsilon+\varepsilon/m, \text{if} a=\arg\max q_*(s,a)\\ \varepsilon/m, \text{otherwise}\end{cases}$

时序差分—从部分序列学习

递推状态价值:$V(S_t)\leftarrow V(S_t)+\alpha(R_{t+1}+\gamma V(S_{t+1})-V(S_t))$

蒙从完整序列学习,适用带终止的决策过程/离线学习(必须获得完整片段来估计累计汇报)/非马更有效；时序从部分序列学习,可用于无终止状态的决策过程/在线学习,只需获取下一状态的及时回报/马环境更有效。



序列产生:

在线策略:目标策略和行为策略相同(SARSA)

离线策略:不同(Q-learning,E-SARSA)

当前状态S_t,根据行为策略选择A_t获得回报R_{t+1}、下一状态S_{t+1},根据目标策略从S_{t+1}产生A_{t+1},计算$Q(S_{t+1},A_{t+1})$,迭代更新$Q(S_t,A_t)$。不断循环以更新 Q 表。

算法 SARSA:行为策略和目标策略都是ε-greedy

$Q(S_t,A_t)\leftarrow Q(S_t,A_t)+\alpha(R_{t+1}+\gamma Q(S_{t+1},A_{t+1})-Q(S_t,A_t))$

$\pi'(S_{t+1})=\arg\max_{a\in A}Q(S_{t+1},a)$, $Q(S_{t+1},A_{t+1})=\max_{a\in A}Q(S_{t+1},a)$

算法 Q-learning:行为策略ε-greedy 目标策略 greedy

$Q(S_t,A_t)\leftarrow Q(S_t,A_t)+\alpha(R_{t+1}+\gamma\max_{a\notin A}Q(S_{t+1},a)-Q(S_t,A_t))$

算法 E-SARSA:行为策略ε-greedy 目标策略行动期望

$Q(S_t,A_t)\leftarrow Q(S_t,A_t)+\alpha(R_{t+1}+\gamma\sum_{a\in A}\pi(a|S_{t+1})Q(S_{t+1},a)-Q(S_t,A_t))$

增量式价值近似

状态价值近似:特征提取$\mathrm{x}(s)=(x_1(s),x_2(s),\cdots,x_m(s))^T$,函数族$\hat{v}(\mathrm{s}|\mathrm{x},\mathrm{w})=\mathrm{w}^T\mathrm{x}$(线性函数),

优化准则$\min\sum_{i=1}^{n}(v_{\pi}(s_i)-\hat{v}(s_{\mathrm{i}}|\mathrm{x},\mathrm{w}))^2$,

优化方法 SGD:$\mathrm{w}^{(\mathrm{new})}\leftarrow\mathrm{w}+\alpha(v_{\pi}(s)-\hat{v}(s|\mathrm{x},\mathrm{w}))\mathrm{x}$

行动价值近似:特征提取$\mathrm{x}(s,a)=(x_1(s,a),\cdots,x_m(s,a))^T$,函数族$\hat{q}(\mathrm{s},\mathrm{a}|\mathrm{x},\mathrm{w})=\mathrm{w}^T\mathrm{x}$(线性函数),

优化准则$\min\sum_{i=1}^{n}(q_{\pi}(s_i,a_i)-\hat{q}(s_{\mathrm{i}},a_{\mathrm{i}}|\mathrm{x}_{\mathrm{i}},\mathrm{w}))^2$,

优化方法 SGD:$\mathrm{w}^{(\mathrm{new})}\leftarrow\mathrm{w}+\alpha(q_{\pi}(s,a)-\hat{q}(s,a|\mathrm{x},\mathrm{w}))\mathrm{x}$

批量式价值近似(经验回放)

经验回放的 SGD 方法:$\mathrm{w}^{(\mathrm{new})}\leftarrow\mathrm{w}+\alpha(v_{\pi}(s)-\hat{v}(s|\mathrm{x},\mathrm{w}))\mathrm{x}$

交替进行采样和梯度下降至收敛$\min\sum(v_{\pi}(s)-\hat{v}(s|\mathrm{x},\mathrm{w}))^2$

状态价值的蒙特卡洛经验回放:$v_{\pi}(s)\approx G_t$

积累经验:$\mathrm{D}=\{(s_1,G_t(s_1)),\cdots,(s_n,G_t(s_n))\}$

从经验采样:$(s,G_t(s))\sim\mathrm{D}$，计算系数:$\mathrm{w}=(\mathrm{X}^{\mathrm{T}}\mathrm{X})^{-1}\mathrm{X}^{\mathrm{T}}g$

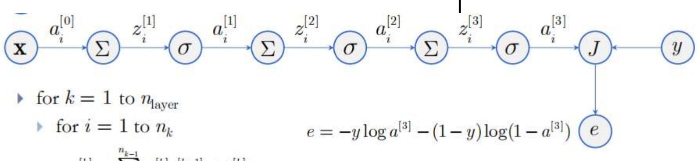
状态价值的时序差分经验回放:$v_{\pi}(s)\approx R_{t+1}+\hat{v}(S_{t+1}|\mathrm{x},\mathrm{w})$

$\mathrm{D}=\{(s_1,r_1',s_1'),\cdots,(s_n,r_n',s_n')\}$, $(s_t,r_t',s_t')\sim\mathrm{D}$, $\mathrm{w}=(\mathrm{X}^{\mathrm{T}}(\mathrm{X}-\gamma\mathrm{X}))^{-1}\mathrm{X}^{\mathrm{T}}\mathrm{r}$

行动价值的时序差分经验回放:

$\mathrm{D}=\{(s_1,a_1,r_1',s_1',a_1'),\cdots,(s_n,a_n,r_n',s_n',a_n')\}$, $(s_t,a_t,r_t',s_t',a_t')\sim\mathrm{D}$

深度强化学习:用神经网络做价值近似,Deep Q-Network



for $k=1$ to n_{layer}

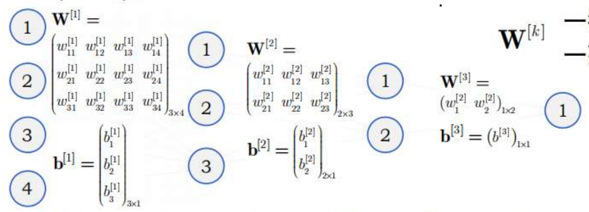
for $i=1$ to n_k

$z_i^{[k]}=\sum_{j=1}^{n_{k-1}}w_{ij}^{[k]}a_j^{[k-1]}+b_i^{[k]}$

$a_i^{[k]}=\sigma(z_i^{[k]})$

$e=-y\log a^{[3]}-(1-y)\log(1-a^{[3]})$

非常慢

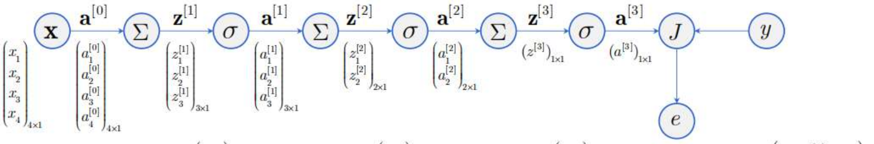


$\mathbf{W}^{[k]}$ 一行：本层的一个神经元
一列：前层的一个神经元

$\mathbf{W}^{[1]}=(w_{ij}^{[1]})_{3\times4}$　$\mathbf{b}^{[1]}=(b_i^{[1]})_{3\times1}$

$\mathbf{W}^{[2]}=(w_{ij}^{[2]})_{2\times3}$　$\mathbf{b}^{[2]}=(b_i^{[2]})_{2\times1}$

$\mathbf{W}^{[3]}=(w_{ij}^{[3]})_{1\times2}$　$\mathbf{b}^{[3]}=(b_i^{[3]})_{1\times1}$



$\mathbf{z}^{[1]}=(\mathbf{b}^{[1]}\ \mathbf{W}^{[1]})\begin{pmatrix}\mathbf{1}\\ \mathbf{a}^{[0]}\end{pmatrix}$　$\mathbf{z}^{[2]}=(\mathbf{b}^{[2]}\ \mathbf{W}^{[2]})\begin{pmatrix}\mathbf{1}\\ \mathbf{a}^{[1]}\end{pmatrix}$　$\mathbf{z}^{[3]}=(\mathbf{b}^{[3]}\ \mathbf{W}^{[3]})\begin{pmatrix}\mathbf{1}\\ \mathbf{a}^{[2]}\end{pmatrix}$　$e=-(y\ \ 1-y)\begin{pmatrix}\log a^{[3]}\\ \log(1-a^{[3]})\end{pmatrix}$

$\mathbf{a}^{[1]}=\sigma(\mathbf{z}^{[1]})$　$\mathbf{a}^{[2]}=\sigma(\mathbf{z}^{[2]})$　$\mathbf{a}^{[3]}=\sigma(\mathbf{z}^{[3]})$

$$\frac{\partial e}{\partial \mathbf{b}^{[3]}}=\frac{de}{da^{[3]}}\frac{da^{[3]}}{dz^{[3]}}\frac{\partial z^{[3]}}{\partial \mathbf{b}^{[3]}}=\left(-\frac{y}{a^{[3]}}+\frac{1-y}{1-a^{[3]}}\right)a^{[3]}(1-a^{[3]})=a^{[3]}-y \qquad \frac{de}{da^{[3]}}=-\frac{y}{a^{[3]}}+\frac{1-y}{1-a^{[3]}}$$

$$\frac{\partial e}{\partial \mathbf{W}^{[3]}}=\frac{de}{da^{[3]}}\frac{da^{[3]}}{dz^{[3]}}\frac{\partial z^{[3]}}{\partial \mathbf{W}^{[3]}}=(a^{[3]}-y)(\mathbf{a}^{[2]})^T \qquad \frac{da^{[3]}}{dz^{[3]}}=a^{[3]}(1-a^{[3]})$$

$$\frac{\partial z^{[3]}}{\partial \mathbf{b}^{[3]}}=1 \qquad \frac{\partial z^{[3]}}{\partial \mathbf{W}^{[3]}}=(\mathbf{a}^{[2]})^T$$

$$\frac{\partial e}{\partial \mathbf{W}^{[2]}}=\frac{de}{da^{[3]}}\frac{da^{[3]}}{dz^{[3]}}\frac{\partial z^{[3]}}{\partial \mathbf{a}^{[2]}}\frac{\partial \mathbf{a}^{[2]}}{\partial \mathbf{z}^{[2]}}\frac{\partial \mathbf{z}^{[2]}}{\partial \mathbf{W}^{[2]}}=\left(-\frac{y}{a^{[3]}}+\frac{1-y}{1-a^{[3]}}\right)a^{[3]}(1-a^{[3]})\mathbf{W}^{[3]}\begin{pmatrix}a_1^{[2]}(1-a_1^{[2]}) & \\ & a_2^{[2]}(1-a_2^{[2]})\end{pmatrix}\begin{pmatrix}(\mathbf{a}^{[1]})^T & \\ & (\mathbf{a}^{[1]})^T\end{pmatrix}$$

1×6	根据最终需要的维度解释	1×1	1×1	1×2	2×2	2×6

$\frac{de}{da^{[3]}}=-\frac{y}{a^{[3]}}+\frac{1-y}{1-a^{[3]}}$　$\frac{\partial z^{[3]}}{\partial \mathbf{a}^{[2]}}=\mathbf{W}^{[3]}$　$\frac{\partial \mathbf{z}^{[2]}}{\partial \mathbf{W}^{[2]}}=\begin{pmatrix}(\mathbf{a}^{[1]})^T & \\ & (\mathbf{a}^{[1]})^T\end{pmatrix}$　$\mathbf{W}^{[1]}=(w_{ij}^{[1]})_{3\times4}$

$\mathbf{W}^{[2]}=(w_{ij}^{[2]})_{2\times3}$

$\frac{da^{[3]}}{dz^{[3]}}=a^{[3]}(1-a^{[3]})$　$\frac{\partial \mathbf{a}^{[2]}}{\partial \mathbf{z}^{[2]}}=\begin{pmatrix}a_1^{[2]}(1-a_1^{[2]}) & \\ & a_2^{[2]}(1-a_2^{[2]})\end{pmatrix}$　$\mathbf{W}^{[3]}=(w_{ij}^{[3]})_{1\times2}$