



## 绪论

- 图灵测试：机器谎言，人说实话，判断哪个是机器  
对的  $P \leq 70\%$  则机器胜
- 七个问题：① 自动的计算机 ② 如何编程让计算机会语言  
③ 神经网络 ④ 计算规模理论（复杂度） ⑤ 自我学习提高  
⑥ 归纳演绎 ⑦ 随机性与创造力
- 1956 达特茅斯会议 麦卡锡 等人 AI 领域诞生  
↳ 会议上还提了  $\alpha$ - $\beta$  剪枝
- 神经网络历史 1943 麦卡洛克、皮茨 首次提出神经元数字模型  
~~赫伯法则~~ → 人在决策时并非完全理性，而是在认知能力、信息获取和计算资源受限的条件下做出“足够好”而非最优
- 自动定理证明：证明 → 推理 在巨大推理空间搜一条公理到结论 的决策
- 下棋 1997 IBM 深蓝击败国际冠军，机器在复杂的 ~~路径~~  $\xrightarrow{\text{期望值}}$  智力博弈中首次战胜人类
- 自然语言处理 (NLP)
- ~~遗传算法~~ → 演化学习
- 1955 乔姆斯基文法分层理论 0、1、2、3型文法
- 神经元：神经网络的构建块，输入 + 计算 + 激活
- Hebbian rule: 2 神经元若经常被同时激活，它们之间的连接  
赫伯法则 就会增强  $\Delta w = \gamma y x$  学习率  $\gamma$  前、后神经元  $x, y$   
加权求和加偏置 权重发散  $\therefore w = \frac{w}{\|w\|}$



扫描全能王 创建

## 模拟思维

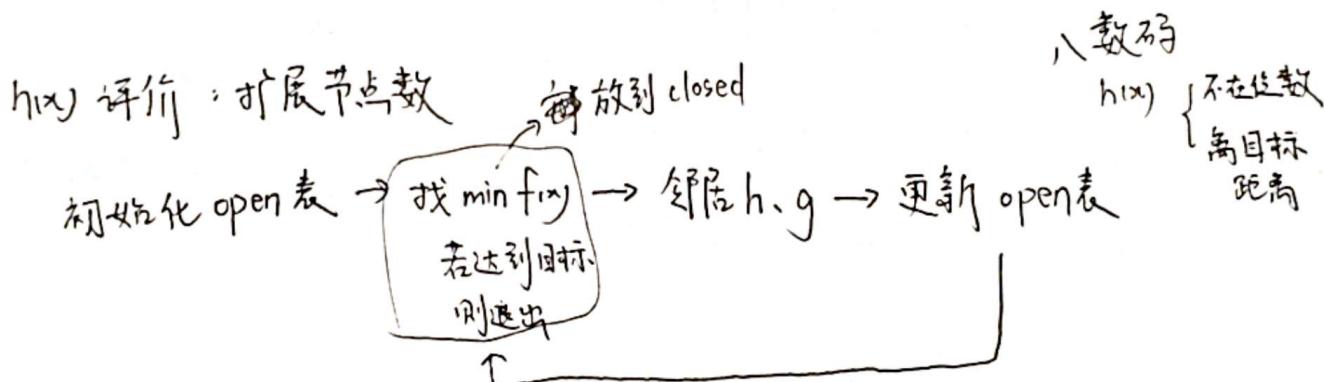
盲目搜索 {  
 dfs 八皇后：每一行安放一个，失败则回退搜下一个  
 深度问题？深度限制（要合理）死循环？记录 path  
 bfs 空间消耗大，效率低 否则找不到解

第一解 bfs 并非最优 bfs 最优

启发式搜索 ~~A\*~~ 与 A\*区别  $h(x) \leq h^*(x)$  启发函数不超过真实代价

$$A: f(x) = h(x) + g(x) \text{ 实际代价}$$

$\hat{h}(x)$  启发函数       $g(x)$  min path       $\begin{cases} \text{欧氏距离} \\ \text{曼哈顿距离} \end{cases}$

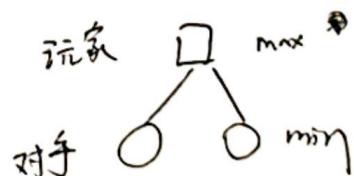


对抗搜索 2-B 剪枝  $\begin{matrix} \min = \alpha \\ \max = \beta \end{matrix}$

max 层 更新  $\alpha = \max(\alpha_p, \alpha_c, \beta_c)$   $\min$  层 更新  $\beta = \min(\beta_p, \alpha_c, \beta_c)$

剪枝 2-B

2类 {  
 $\alpha$  剪枝 MIN 层  
 $\beta$  剪枝 MAX 层



蒙特卡洛搜索：选择、拓展、模拟、反向传播

值得探索的 子节点 到结束 (回溯更新节点信息)  
 节点 加入树



扫描全能王 创建



模拟学习

无监督	无标签
有益智 SL	有标签
强化 RL	智能体 $\Leftarrow$ 环境

**强化学习：**智能体与环境交互，学习一条策略，最大化长期累积回报  
↓  
不断试错 五元组

MDP 马尔可夫决策过程  $M = (S, A, P, R, \gamma)$  折扣因子  $[0, 1]$

(动态 + 白盒环境) 状态空间 动作空间 转 即时奖励函数

$$P(s'|s, a) \quad R(s, a) \mid R(s, a, s')$$

动/静：不可转移，单步

白/黑盒：变量与目标的关系有无公式

$$s_0 \xrightarrow{a_0, R(s_0, a_0)} s_1 \xrightarrow{a_1, R(s_1, a_1)} s_2 \dots$$

(累加奖励) 总回报  $R(s_0, a_0) + \gamma R(s_1, a_1) + \gamma^2 R(s_2, a_2) \dots$

历史： $H_t$  一直到  $t$  为止所有可观测变量 策略： $\pi(a|s)$

状态： $s_t = f(H_t)$

奖励： $R(s, a)$  与  $\gamma$

贝尔曼方程

$$\begin{aligned} V_{\pi}(s) &= \sum_{a \in A} \pi(s, a) q_{\pi}(s, a) && \text{价值函数 (a未知)} \\ q_{\pi}(s, a) &= \sum_{s' \in S} P(s'|s, a) [R(s, a, s') + \gamma V_{\pi}(s')] && \text{价值-动作} \\ &= E[G_t(s)|s_t=s, A_t=a] && \text{函数} \\ &&& (a已知) \end{aligned}$$

互相代入可得递推关系

$\bar{x}$

蒙特卡洛价值估计：用  $\pi$  在  $S$  中采  $N$  个样本，用 均值 累计奖励估计  $V_{\pi}(s)$

策略提升定理： $\pi'(s) = \arg \max_a q_{\pi}(s, a)$  则  $V_{\pi'}(s) \geq V_{\pi}(s) \forall s$

策略  $\rightarrow$  价值评估  $\rightarrow$  策略提升

$$\pi \uparrow \quad V_{\pi}(s)$$

$$Q^{\pi'}(s, \pi'(s)) \geq V^{\pi}(s)$$

更新为  $\pi'$



扫描全能王 创建

无监督 K均值聚类 硬分配

- ① K簇，初始质心 ② 归类，欧氏距离  
③ 更新每个类质心 ④ 继续迭代 {  
     $(1, 1) \quad (2, 2) \quad (10, 10)$   
     $\left( \frac{1+2+10}{3}, \frac{1+2+10}{3} \right)$

EM算法 软分配

(含隐变量)，以 P 分组，最大似然估计参数



有监督 通过环境直接得标准答案 还可加激活 sigmoid  $s(x) = \frac{1}{1+e^{-x}}$

步骤：1. 给出函数  $y = w\tilde{x} + b$  表示

机器学习

2. 定义 Loss  $L(\theta)$  & 包含所有参数 评价

分类：交叉熵  
3. 优化 optimize 更新所有参数 优化

← 反向传播算法

网络不做的更深：过拟合！

Hinton

模型容量太大，会过度记住训练数据细节

s.t. 训练集表现优秀 新数据上泛化能力差，性能 test data

强化 三类方法

	基于价值	基于策略	Actor-Critic	TD 结合 MC 与 DP 优点，单步更新无需完整轨迹
几步的	TD $\times$	reinforce $\times$		
整合	MC $\times$	模型 (无需 P)		梯度下降更新 $\theta$ $G_t > 0$ 奖励，更易做此策略



扫描全能王 创建

# 模拟思维(续)

知识图谱 三元组(实体,关系,实体)

P	AN
T	TP
F	FP

TN	
FN	

$$rec = \frac{TP}{TP+FN}$$

$$pre = \frac{TP}{TP+FP}$$

$$acc = \frac{TP+TN}{TP+TN+FP+FN}$$

$$F_1 = \frac{2 \cdot pre \cdot rec}{pre + rec}$$

隐对向的观P

HMM例子 观测 发射

$$\lambda = (S, O, A, B, \pi)$$

↑  
发射  
↓  
隐→隐

隐状态集
转
初始状态分布
 $\pi = (0.6, 0.4)$

↑ 隐  $S = \begin{matrix} \text{晴} & | & \text{雨} \end{matrix}$        $O = \text{吃汉堡, 冰淇淋}$

$s$	$r$			
$s \rightarrow s$	$s \rightarrow r$	$r \rightarrow s$	$r \rightarrow r$	

A
B

$s$	$0.1$	$0.2$	$0.3$
$r$	$0.9$	$0.8$	$0.7$

~~HMM~~ HMM 对应

三类问题

	输入	输出	算法	AB 给O求S(天气)
解码	观+参	隐序列	维特比	给O求S(天气)
评估	观+参	$P_{观}$	前向	给O AB 求 $P(O A)$
学习	观	参	前向-后向	



扫描全能王 创建

模拟语言

词向量

LSTM  
1997  
→ RNN  
1987

transformer  
2017

bert

训练：自回归  
语言模型

双向 BERT：Transformer 的 Encoder + 位置

单向 GPT：生成类

理解类  
(分类、  
问答)

输入  
位置编码

三嵌入

Token 语义

Segment 句子归一化

Position 位置掩码掩上文

(让它预测)

NSP

学习句间关系

潜在语义分析 LSA：单词 - 文档矩阵

↓

神经语言模型 2003 本吉奥 首个 输入(词向量) → 隐藏 →

↓ word = vec 2013 one-hot 的词向量 → 词的频率分布  
输出 (词概率)

架构 { skip-gram 中心词 → 上下文 细致  
CBOW 上下文 → 中心词 快

作用：优化词向量，语义相近的词在向量空间更接近

通过上下文预测任务

LSTM 处理长期依赖 (防梯度消失或爆炸)  $x_t$  输入 上一隐藏  $h_{t-1}$

$$\begin{cases} \text{遗忘门} & f_t = G[W_f(h_{t-1}, x_t) + b_f] \\ \text{输入门} & i_t = G[W_i(h_{t-1}, x_t) + b_i] \quad \tilde{c}_t = \tanh(W_c(h_{t-1}, x_t) + b_c) \\ \text{输出门} & o_t = G[W_o(h_{t-1}, x_t) + b_o] \quad c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \\ & h_t = o_t \odot \tanh(c_t) \end{cases}$$

逐元素乘 更新

Transformer NLP 革新，取代 LSTM

序列长度不变，词维度分份，最后拼 cell

输入  $a^i$   $q^i = W_q a^i$   $k^i = W_k a^i$   $v^i = W_v a^i$  共  $i$  个 token 起来 状态

每层有 编码器 词嵌入 + 位置编码 → 多头注意力  $A(Q, K, V) = G(\frac{QK^T}{\sqrt{d_k}})V$   
 归一化 (Layer Norm)  
 残差

输出 预测的 token  $X-out$

mask 自注意力 (自回归) + 跨注意力 (交叉)

↓ Linear + softmax  
词概率

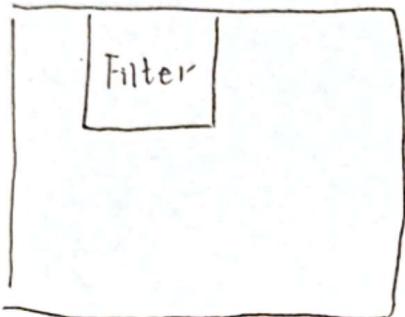


扫描全能王 创建



## 模拟视觉

### CNN 卷积神经网络



$$\text{每个维度 } W_{\text{out}} = \frac{W_{\text{in}} + 2p - f}{\text{stride}} + 1$$

VGG、Res Net  
3x3  $F(X) + X$   
maxpool

$$\text{转置卷积, } H_{\text{in}} = \frac{H_{\text{out}} + 2p - f}{\text{stride}} + 1$$

(上采样过程, 恢复分辨率)

ViT 图像  $\mapsto$  patch 加位置编码和 class token  
给 Encoder (Transformer架构)  
最后输出的是 class token 预测

GAN 生成式对抗网络 对抗训练：生成问题  $\rightarrow$  min-max博弈  
图片数据化，用神经网络去拟合  $\rightarrow$  步骤：train data; 设计网络; 优化参数; 生成新图

battle { ① 取 data ② G产假 data ③ 训 D ④ 训 G ⑤ 迭代 }

1. 真样本  $\rightarrow 1$  2. 假样本  $\rightarrow 0$

$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}} [\log D(x)] + \mathbb{E}_{z \sim p_g} [\log (1 - D(G(z)))]$

固定一个更新另一个  
交替随机梯度下降

### 通道数

out = Filter 数量

in RGB 3

灰 1

每个 Filter 的 channel

与原图一致

① 编码器将输入序列变成

“上下文信息向量”

② 解码器依据该输出生成目标序列

① 多头自注意力 Attention (Q, K, V)

$$\uparrow \text{很多个层}) = G\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

每层之间 ~~共享~~ 残差连接并归一化  
 $X_{\text{out}}$

② 抽码多头自注意力

$$\text{softmax}\left(\frac{QK^T}{\sqrt{d_k}} + \text{mask}\right) \begin{cases} \text{位于 } j \text{ 时, 只关注 } \leq j \text{ 的} \\ > j \text{ 的为 } -\infty \Rightarrow \text{softmax} = 0 \end{cases}$$

同理  
每层之间

用于已生成序列  $\rightarrow Y_{\text{out}}$

② 交叉多头自注意力

$$\begin{aligned} X_{\text{out}} \text{ 与 } Y_{\text{out}} &\rightarrow Z_{\text{out}} \\ Q &\leftarrow Y_{\text{out}} \\ K, V &\leftarrow X_{\text{out}} \end{aligned}$$

Linear + softmax

词概率

③ 输出预测的 token



扫描全能王 创建

## VAE 变分自编码器

① 编码：输入  $\rightarrow$  指概率分布（不是固定值描述特征）

连续：

② 解码：采样  $\rightarrow$  重建与原始输入相似的数据

不相关  $\leftrightarrow$  相关

~~diffusion~~ diffusion 扩散模型 加噪、去噪  
损失函数部分 1. 生成尽可能像原始 2. 分布尽可能接近  $N(0, 1)$  变化过程 ✓

~~diffusion~~ diffusion 扩散模型 加噪、去噪  
(前向) (反向)

架构 U-Net



$$x_t = \sqrt{a_t} x_0 + \sqrt{1-a_t} \varepsilon, \quad \varepsilon \sim N(0, 1)$$

## 模拟听觉

{ sound power 单位 t 产生的 E , 单位 W  
sound intensity 单位 S 的 power , 单位  $W/m^2$

分帧：语音短时间内  
“近似平稳”

$$\text{dB}(I) = 10 \log_{10} \left( \frac{I}{I_{\text{ref}}} \right)$$

$$\text{欧拉公式 } e^{i\theta} = \cos \theta + i \sin \theta$$

$$g(t) = \int c_f \cdot e^{i2\pi f t} dt$$

傅里叶变换：时域信号  $\rightarrow$  不同 f 的正弦波叠加，得到频域表示

频率 f 处正弦分量的幅度  $a_f$  和相位  $\phi_f$   $\begin{cases} \arg \max_{\varphi \in [0, 1]} \int s(t) \sin(2\pi(f t - \varphi)) dt \\ \max_{\varphi \in [0, 1]} \left| \int s(t) \sin(2\pi(f t - \varphi)) dt \right| \end{cases}$

Mel 值：人耳对声音的感知是对数型的（低 f 分辨率高，高 f 低）

$$MEL(f) = 2595 \log_{10} \left( 1 + \frac{f}{700} \right)$$

DCT 离散余弦变换

音高 pred

能量 pred

(每块多少帧)

MFCC：对 Mel 频谱做进一步压缩，提取语音谱包络特征

Embedding  $\rightarrow$  FFT E  $\rightarrow$  时长预测器  
Mel 值  $\leftarrow$  FFT D  $\leftarrow$  长度管理  
(控制语速)

Fast Speech 2 非自回归，可控的 TTS 模型

tacotron 字符序列  $\rightarrow$  Mel 值图  $\xrightarrow{\text{声码器}}$  语音波形

(TTS) 自回归 输入 输出

语速不可控，生成慢



扫描全能王 创建