



AIA计划：让AI帮人做算法?!

卜 东 波

前瞻研究实验室

2019年2月28日

算法课上的容易事和难事

- 容易回答的问题：

Q: 你设计的这个算法是**怎样工作**的？

- 难回答，甚至无法回答的问题：

Q: 你是**怎样想出来**这个算法的？

A: 聪明？顿悟？灵感？

理解算法易，理解设计过程难！

Human algorithmist如何设计算法？

- 观察**少数简单例子**，寻找**灵感**！



- 为证明一个问题是否是NP完全的，我们首先观察这个问题的**简单实例**，直到我们找到一些实例，能够展现出有趣的行为（问题结构）

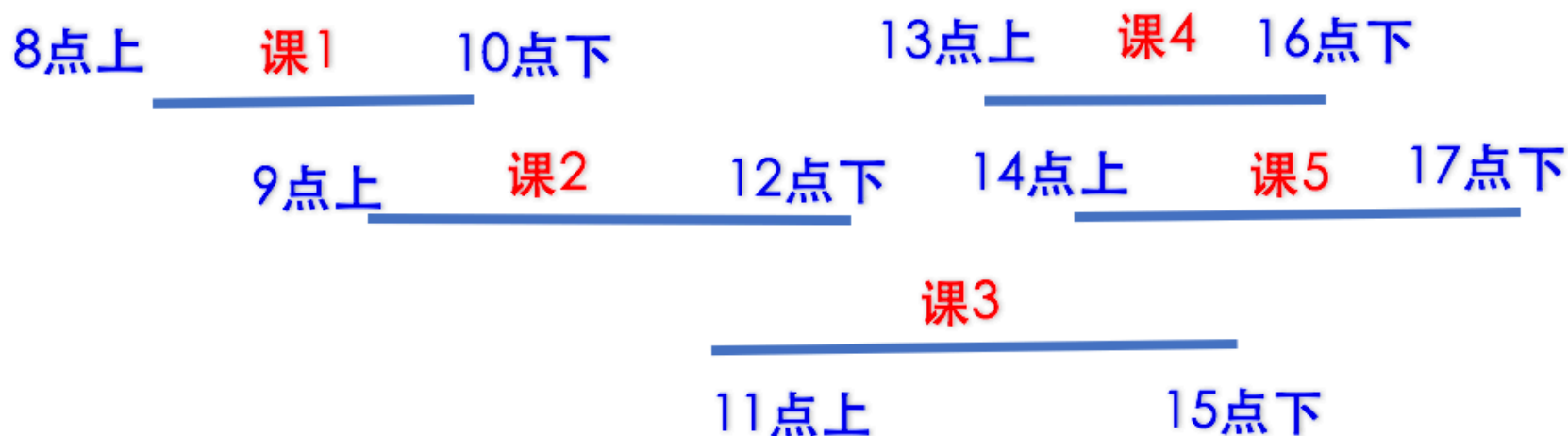
----- C. Papadimitriou

- R. Karp随身带一个小本子，碰到问题，手工**画几个最简单的例子**，观察规律，寻找灵感。。。

----- S. C. Li

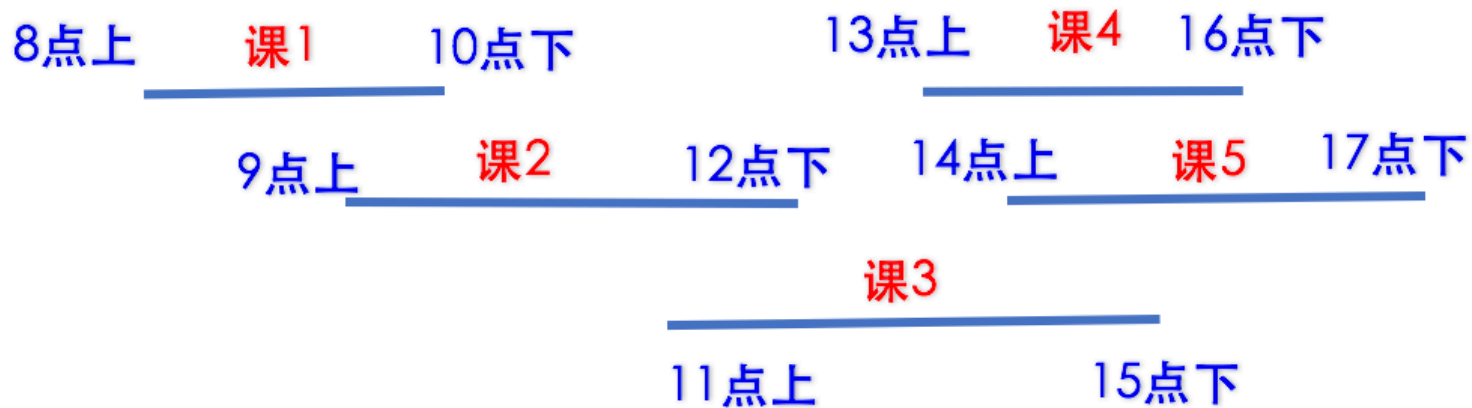
- 李明，姜涛，张凯宗，刘晓文，王鲁生，.....

以排课问题为例



- 有5门课申请使用报告厅，已知上下课时间
- 问：最多能安排几门课？

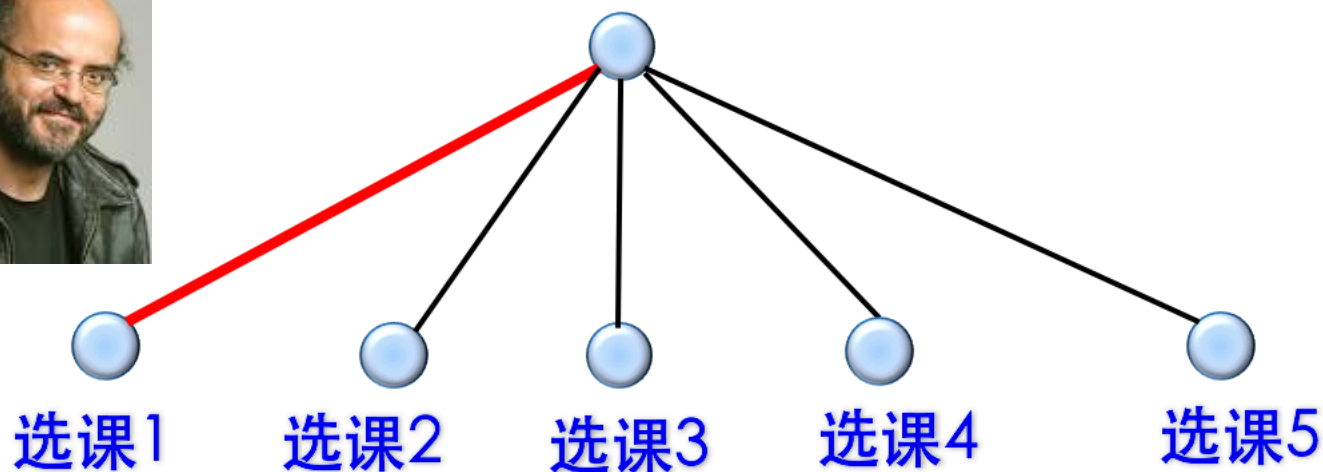
经典的贪心算法: 选最早下课的!



GREEDYINTERVALSCHEDULING(*CourseSet*)

- 1: **while** *CourseSet* $\neq \emptyset$ **do**
- 2: Select the course *C* with **earliest finishing time**;
- 3: Remove *C* and related courses from *CourseSet*;
- 4: **end while**

规则简单，想出来不容易！



- 规则1：优先安排最早上课的；
- 规则2：优先安排时长最短的；
- 规则3：优先安排冲突最少的；
- 规则4：优先安排最早下课的；
- 规则5：优先安排最晚上课的；

×

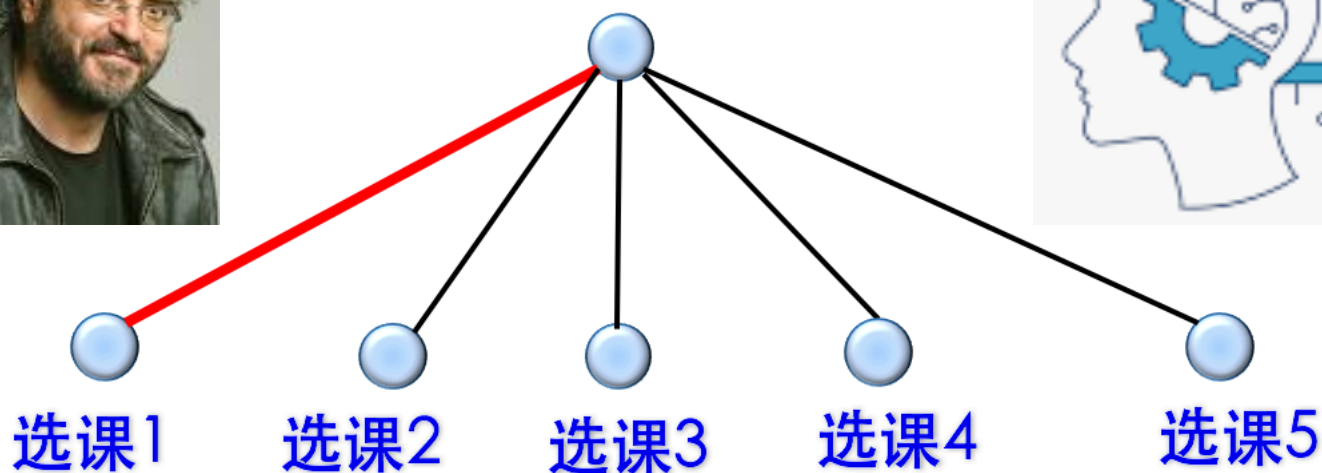
×

×

✓

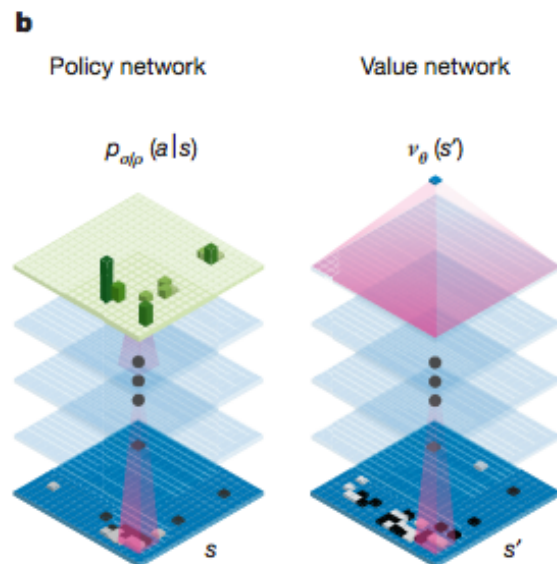
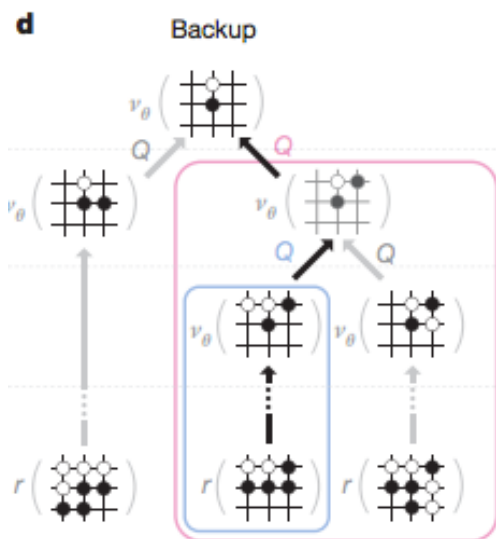
✓

能否用AI学出算法设计背后的“灵感”？



Human Algorithmist = AI Algorithmist?

AlphaGO的启示



- 关键技术：
 - 蒙特卡洛树搜索；
 - 采用深度学习技术学习已有的棋谱；
 - 采用增强学习技术，将学习目标从“预测”变成“对局”

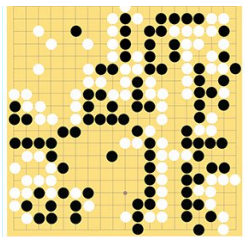
AlphaGo纠正我的观念

- Dirty的事情让dirty tool干！
- 说不清楚的事情就是dirty:

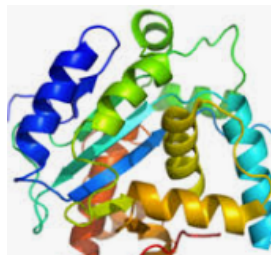
– 视觉机理



– 下棋



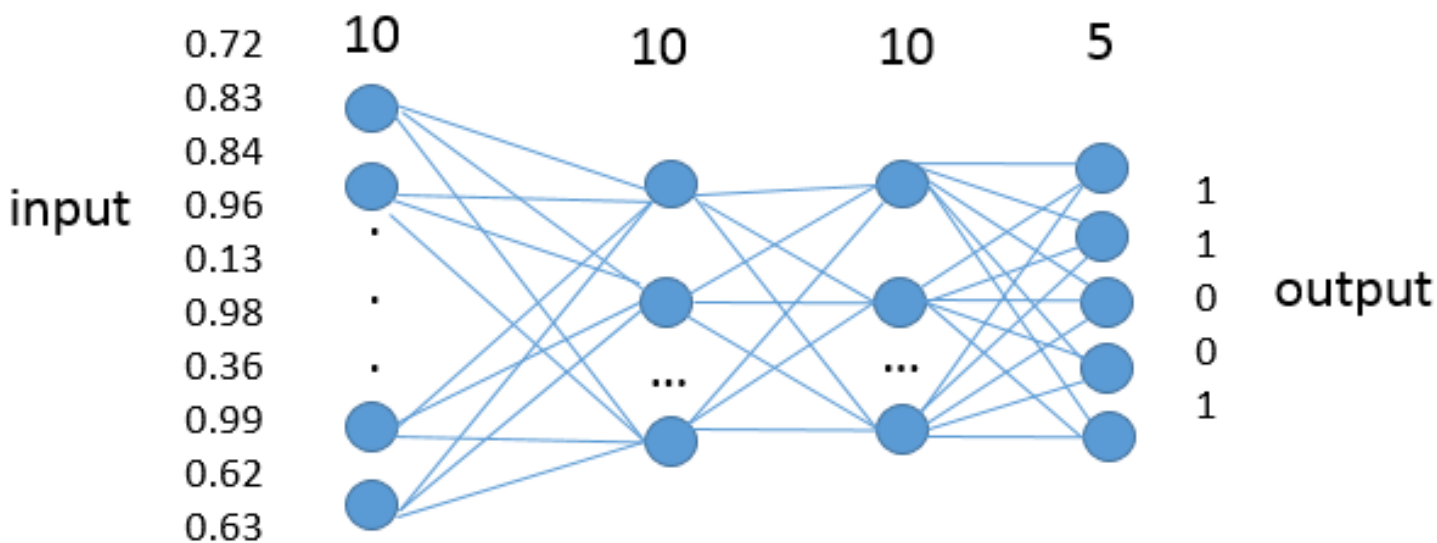
– 蛋白质折叠



– 灵感

AIA计划：让NN学“灵感”！

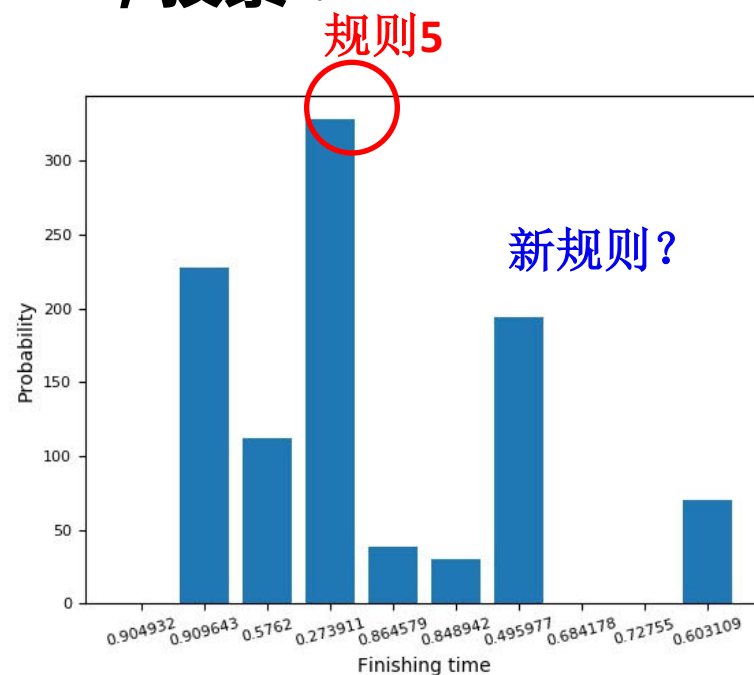
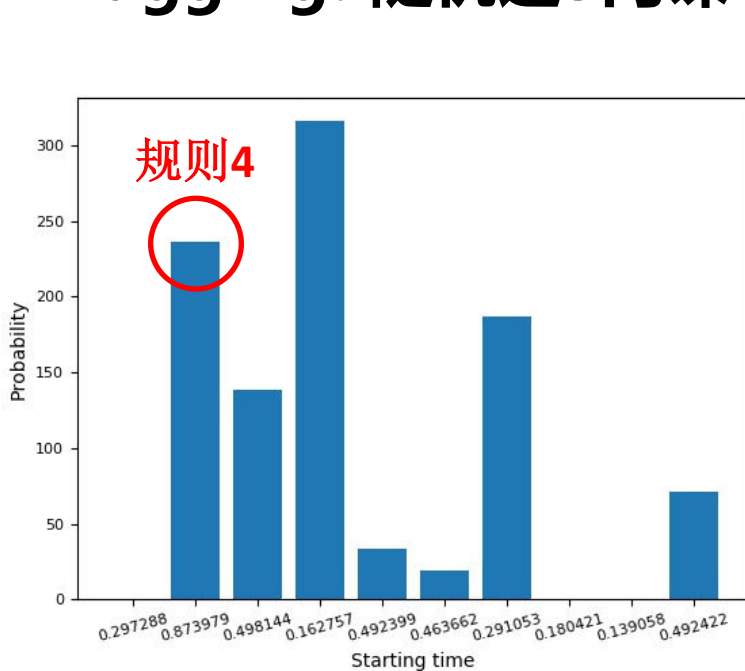
- 实验设计：
 - 随机产生10万个5门课的例子，
 - 先用笨拙的动态规划算法跑出标准答案
 - 然后用NN来学



- 正确率：98%!

AIA计划：以小NN指导解大例子

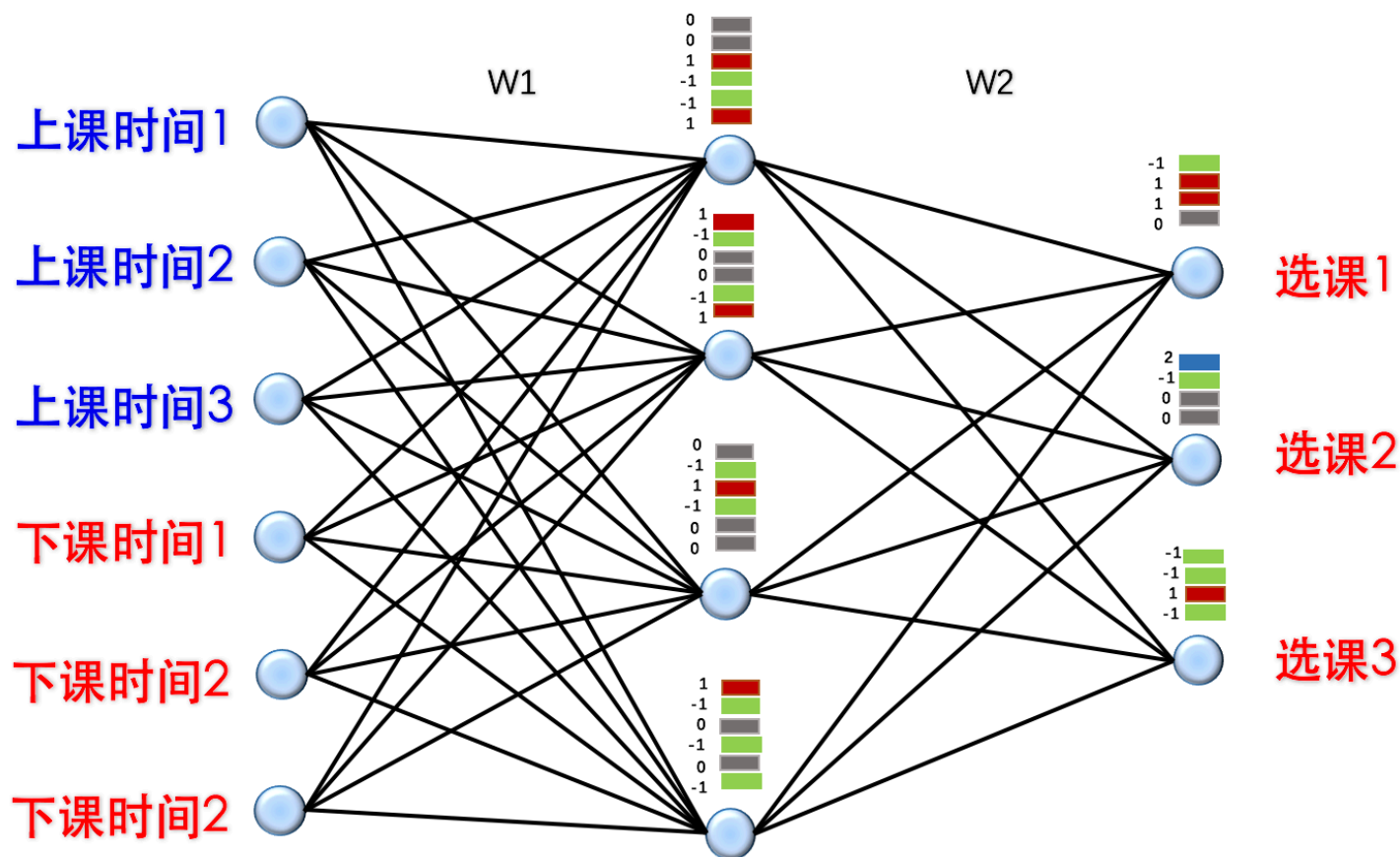
- 如何解10门课的例子？
- Bagging! 随机选5门课，跑小NN，投票！



- 正确率：95%!

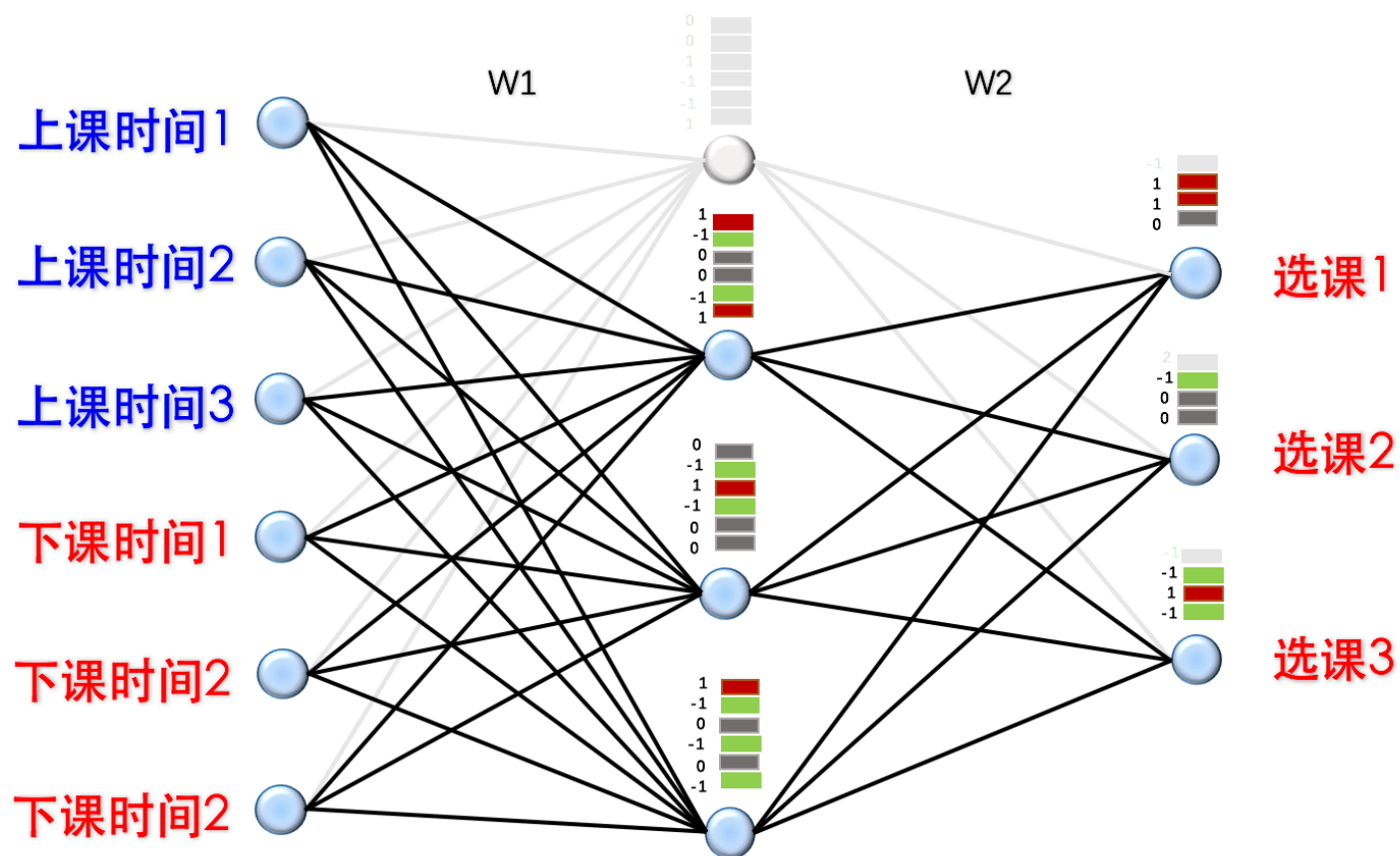
NN学出了规则4和规则5!

NN怎么工作的？打开网络！解释网络！



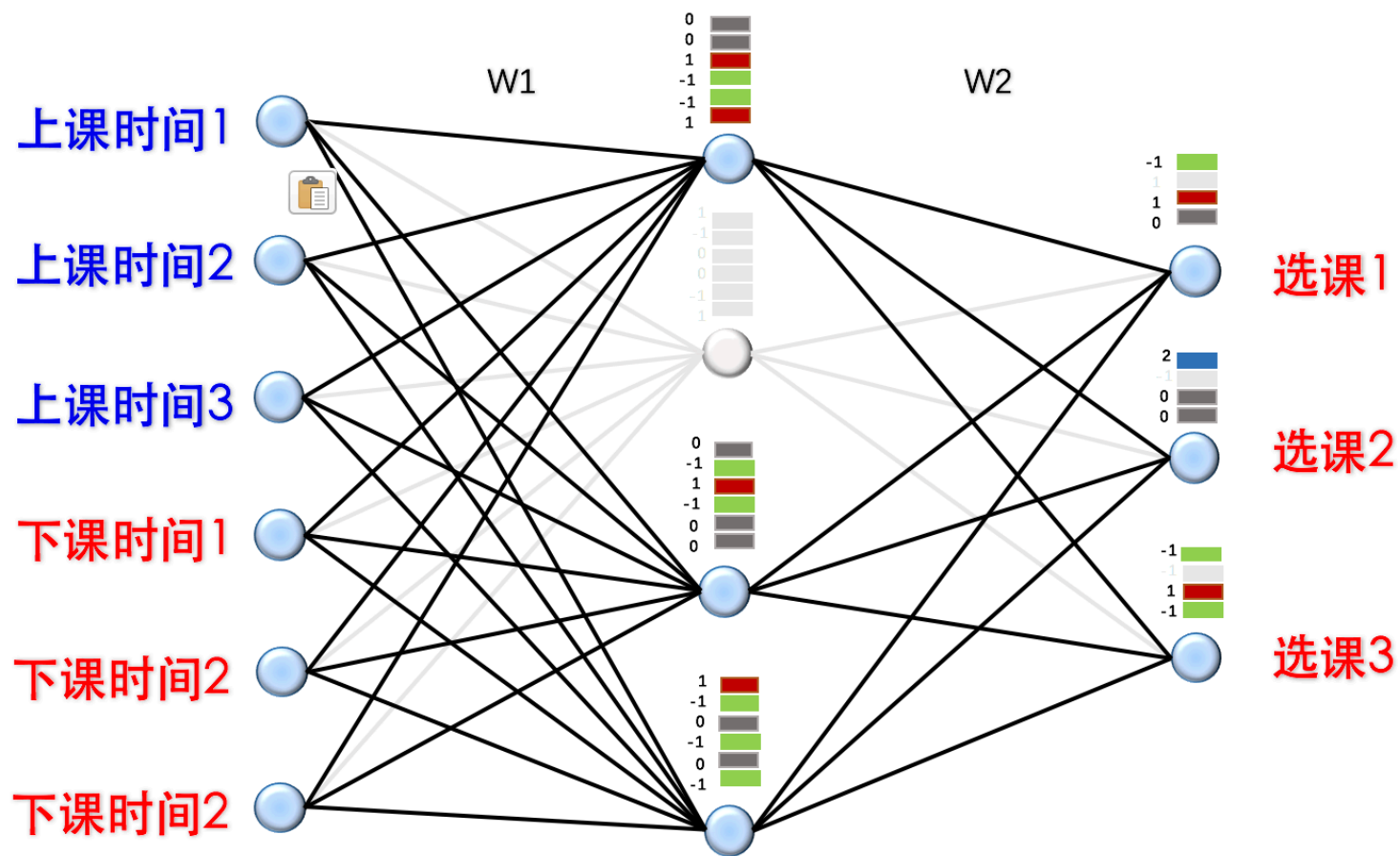
NN怎样同时掌握贪心规则4和规则5？
选最早下课的和最晚上课的

隐层结点2, 3, 4管规则4



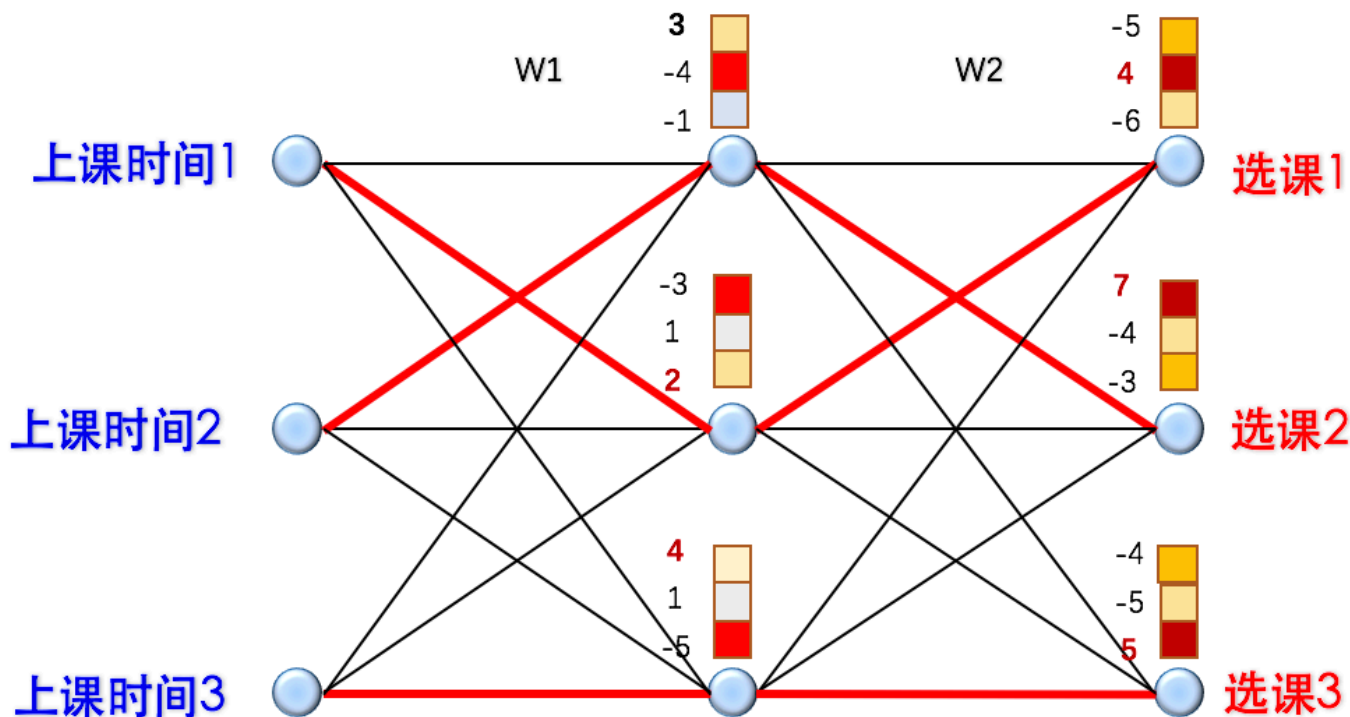
一个只会贪心规则4的网络
选最早下课的

隐层结点1, 3, 4管规则5



一个只会贪心规则5的网络
选最晚上课的

再深入解释：“最早下课”咋实现？



NN：中间层每个结点都“强调”一门课，把上课时间的差异逐层放大！

Human设计的算法 vs AI设计的算法

Human algorithmist:

GREEDYINTERVALSCHEDULING(*CourseSet*)

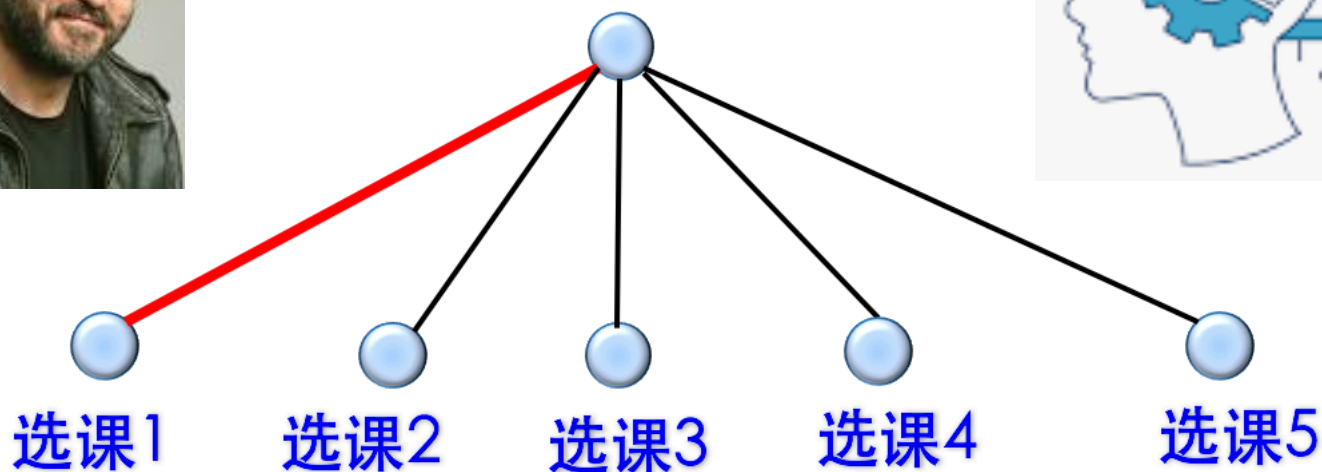
- 1: **while** *CourseSet* $\neq \emptyset$ **do**
- 2: Select the course *C* with **earliest finishing time**;
- 3: Remove *C* and related courses from *CourseSet*;
- 4: **end while**

AI algorithmist:

NNINTERVALSCHEDULING(*CourseSet*)

- 1: **while** *CourseSet* $\neq \emptyset$ **do**
- 2: Select the course *C* with **highest score by NN**(*CourseSet*);
- 3: Remove *C* and related courses from *CourseSet*;
- 4: **end while**

Human Algorithmist = AI Algorithmist?



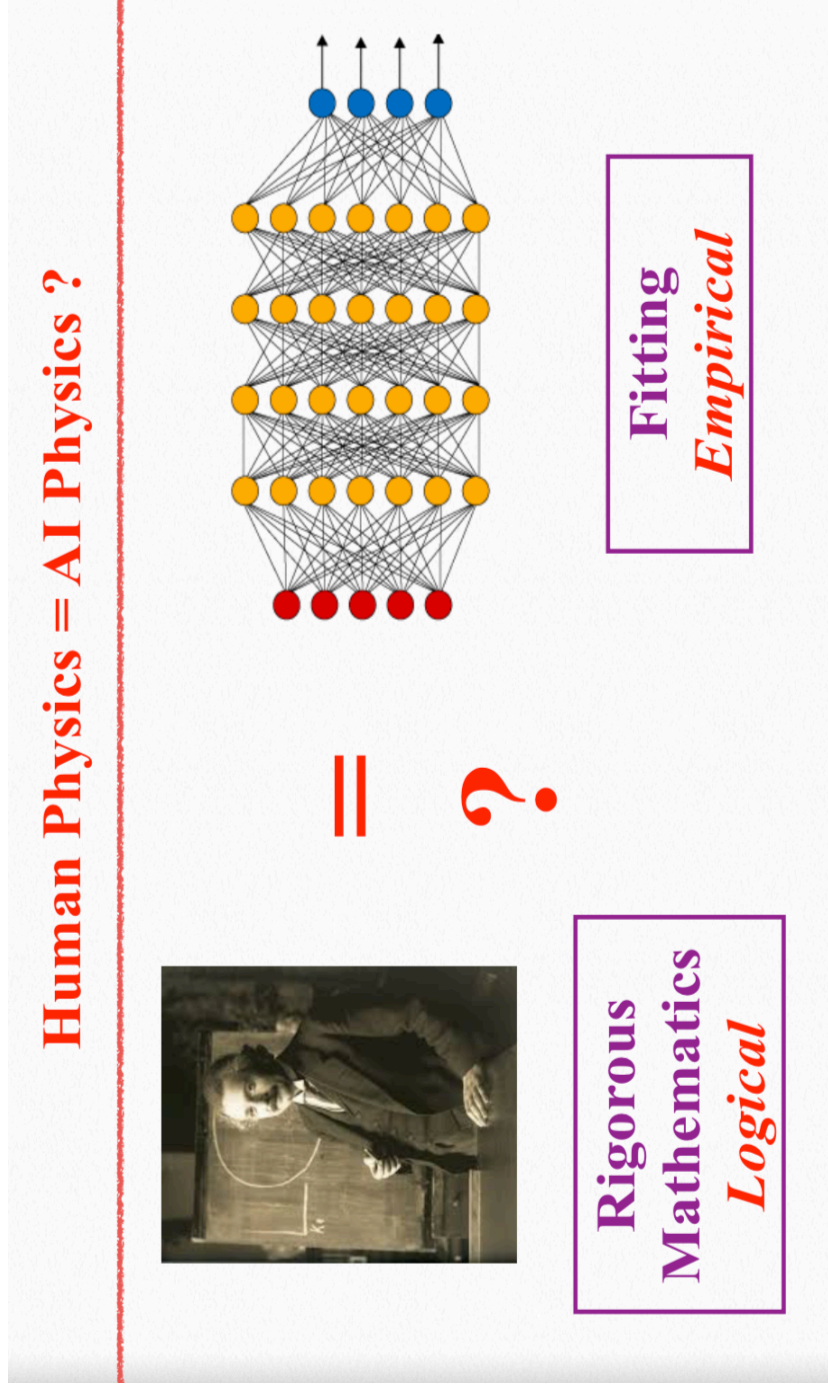
我的回答：Yes !

AIA计划：让AI帮忙设计算法

- 优势：
 - 构造训练集**无需人工标注**，只要**机器算**；
 - 使用曙光机的**强大传统算力**
- 下一步计划：
 - NP-完全问题：**AIA-SAT, AIA-TSP**
 - 整数线性规划求解器：**AIA-ILP**

致谢

- SIGMA(Special Interest Group on Mathematics and Algorithms)讨论班 (孙晓明、张家琳)
- 清华翟辉报告：

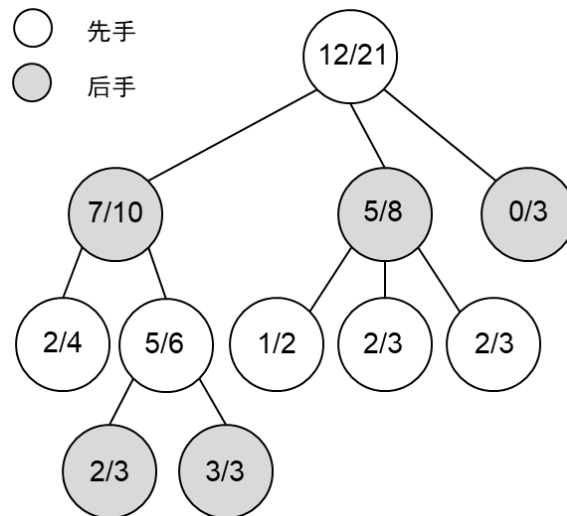


Thanks !

AlphaGo中的蒙特卡洛树搜索

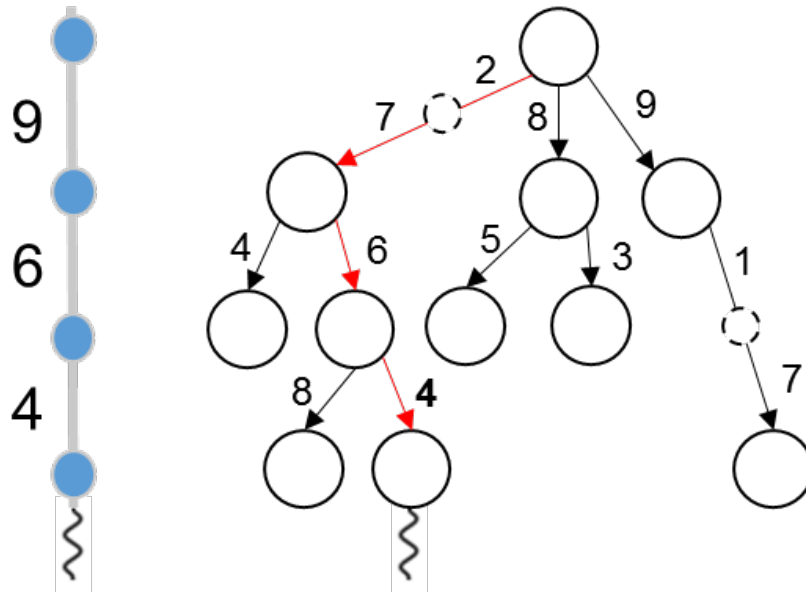
Algorithm 8 蒙特卡洛树搜索算法

- 1: **while** 当前局面无法区分胜负 **do**
- 2: **for** $i = 0$ 到 N **do**
- 3: **for** $j = 0$ 到 L **do**
- 4: 按一定规则**选择**一个走子策略
- 5: **end for**
- 6: **模拟**快速走子直到可以分出胜负
- 7: 将胜负信息**反向传播**到经过路径的所有节点
- 8: 更新每个节点的权重
- 9: **end for**
- 10: 以一定规则选择当前局面的子节点为该步走子策略
- 11: 更新当前局面
- 12: **end while**



采用蒙特卡洛树搜索的拼接

- 基本思想：将一定长度的一个或几个位点的联配作为路径搜索的一步，然后使用蒙特卡洛树搜索求解



- 关键点：
 - 1.评价值：模拟完成后返回什么值？
 - 2.选择规则：每次选哪个点尝试？
 - 3.“走子”：选哪个节点作为该步搜索结果？

我们的解决方案

解决方法：

- 评价值： S_{leaf} 即：完整路径和部分路径打分加权和
- 选择规则：选Exploitation+Exploration最大节点
- “走子”： n （选中次数）最大的子节点

$$S_{\text{leaf}} = (1 - \lambda)S_{\text{global}} + \lambda S_{\text{local}}$$

$$\text{Exploitation} = \text{Exploitation} + \frac{S_{\text{leaf}} - \text{Exploitation}}{n}$$

$$\text{Exploration} = \frac{\eta S_{\text{init}} \sqrt{n_p}}{1 + n}$$

• 核心思想：

- 平衡对已有搜索结果的利用和对未知领域的探索

