基于神经网络的黑白棋

AlphaGo for Reversi

2021-12-24

刘雪琪
颜宇康
丘启圆

选题背景与简介

Topic background and introduction

第一部分：选题背景与简介

Topic background and introduction

选题的背景

2017年10月19日，AlphaGo团队在《自然》上发表文章介绍了AlphaGo Zero。AlphaGo Zero框架成功地应用在**二人有限零和博弈问题**,。通过自我对弈，AlphaGo Zero在**三天内以100比0的战绩战胜了AlphaGo Lee，花了21天达到AlphaGo Master的水平，用40天超越了所有旧版本。**AlphaGo的核心是使用**深度网络在蒙特卡罗树搜索**过程中进行棋盘局势判断和走棋选择决策。这些深度网络使用监督学习的方法从人类专家棋谱中进行训练，使用强化学习的方法从自我对弈的过程中进行训练。

研究意义——AlphaGo Zero在军事领域的应用

在2007年人机国际象棋大赛中，“深蓝”一举击败人类棋手卡斯帕罗夫，引起美国军方高度关注，提出了**“深绿”计划**。

“深绿”的任务是**预测战场上的瞬息变化**，帮助指挥员提前进行思考，判断是否需要调整计划，并协助指挥员生成新的替代方案。

在深入的研究中，发现“深绿”仅在**小型战事**中能做出合理决策，当战事变得复杂时，它的分析结果也会变得发散。

AlphaGo Zero

AlphaGo Zero除了基本规则之外，它对这些棋类游戏一无所知。初始训练时不使用人类棋谱做有监督学习，而是直接从**基于游戏规则**的随机下法开始学习。AlphaGo Zero有两个重要的部分发挥着作用，一个是**蒙特卡洛搜索树MCTS**，一个是**卷积神经网络CNN**。

蒙特卡洛搜索树MCTS

蒙特卡罗树搜索(Monte Carlo Tree Search), 一种通过随机游戏推演来逐渐建立一棵不对称搜索树的过程, 它是人工智能领域中**寻找最优决策**的一种方法。蒙特卡罗树搜索**采用树状结构表征黑白棋博弈问题。** 蒙特卡罗树搜索**减少了搜索的宽度和深度**, 并在有限的遍历过程中, 寻找到最有潜力的下一步行动, 即形成**决策**。

宽度——通过一定次数的遍历后, 部分分支会表现出更高的胜率, **将有限的遍历集中在这类更有潜力的分支上**

深度——搜索到某一中间节点时停止搜索, 用基于简单算法(如均匀随机算法)的模拟过程执行到终结点或者在停止搜索后**利用评估函数直接预测当前中间节点盘面的胜负**

卷积神经网络（Convolutional Neural Networks, CNN）

深度神经网络通过建立多个隐含层模拟人脑分析学习的机制, 吸收大量数据的经验建立规则(网络参数), 实现特征的自主学习, 主要适用于无法编制程序、需求经常改变、有大量数据且无需精确求解的一类问题。 深度神经网络组成主要包括**输入、神经元单元、神经网络、成本函数和算法**。

卷积神经网络是一类**包含卷积计算且具有深度结构的前馈神经网络**，是深度学习的代表算法之一。卷积神经网络具有表征学习能力，能够按其阶层结构对输入信息进行平移不变分类。

典型的 CNN 由3个部分构成：**卷积层、池化层、全连接层**

算法介绍

Algorithm introduction

算法组成

othello_game.py -- 关于黑白棋游戏方面

evaluate.py -- 评估函数：自我对弈时，游戏赢了+1分，游戏输了—1分，平局没变化

mcts.py -- 蒙特卡洛搜索树

neural_net.py -- 神经网络

human_play.py -- 实现人类棋手与AI下棋功能

train.py -- 训练模型

config.py -- 训练参数

main.py -- 主函数

算法组成：neural_net.py -- 神经网络

将Alpha Go中两个结构独立的**策略网络**(SL策略网络和快速走棋网络)和**价值网络**合为一体, 合并成一个深度神经网络。 在该神经网络中, 从输入层到中间层的权重是完全共享的, 最后的输出阶段分成了**策略函数输出Policy Head**和**价值函数输出Value Head**。

深度残差神经网络fθ(s)由**1个卷积块后跟19个或39个残差模块**(或残差网络)组成,。输入信息经过深度残差网络的处理, 得到盘面的深层次特征, 基于这些特征分别利用Policy Head策略输出模块和Value Head价值输出模块得到**下一步棋的动作概率分布π**和**当前棋面对应下棋方获胜的估计值z**。

算法组成：train.py -- 训练模型

训练过程主要分为三个阶段：**自我对战学习阶段，训练神经网络阶段和评估网络阶段**

在自我对战学习阶段，每一步的落子是由MCTS搜索来完成的。在MCTS搜索的过程中，遇到不在树中的状态，则使用神经网络的结果来更新MCTS树结构上保存的内容。当每一局对战结束后，我们可以得到最终的胜负奖励z,1或者-1。

在训练神经网络阶段，我们使用自我对战学习阶段得到的样本集合(s,π,z),训练我们神经网络的模型参数。训练的目的是对于每个输入s, 神经网络输出的p,v和我们训练样本中的π,z差距尽可能的少。

$$L=(z-v)2-\pi T\log(p)+c||\theta||2$$

评估阶段，自我对战的双方各自使用自己的神经网络指导MCTS搜索，并对战若干局，检验AlphaGo Zero在新神经网络参数下棋力是否得到了提高。

算法组成：config.py -- 主要训练参数

num_iterations: 迭代次数

num_games: 每次迭代中自我对弈的次数

num_mcts_sims: 每场游戏的 MCTS 模拟次数

c_put: MCTS 中的探索级别常数

l2_val: 训练期间使用的 L2 权重正则化级别

learning_rate: 动量优化器的学习率

t_policy_val: 策略预测值

epochs: 训练期间的时期数

epsilon: 用于计算Dirichlet噪声的 epsilon 值

resnet_blocks: resnet 中的残差块数

temp_init: 控制探索的初始温度参数

temp_final: 控制探索的最终温度参数

temp_thresh: 温度初始值变为最终值的阈值

算法组成： c_put: MCTS中的探索级别常数

c 是一个常数，用于权衡探索 (Exploration) 与利用 (Exploitation)。

探索是指选择一些之前没有尝试过的下法，丰富自己的知识，新的知识可能带来不错的结果。

利用是指根据现有的知识选择下法。

c 越大，就越偏向于探索；c 越小，就越偏向于利用。

$$\operatorname{argmax}(Q(s_t,a)+U(s_t,a))$$

$$U(s,a)=c\times P(s,a)\frac{\sqrt{\sum_b N(s,b)}}{1+N(s,a)}$$

算法组成： temp_init: 控制探索的初始温度参数

τ为温度参数，控制探索的程度，τ 越大，不同走法间差异变小，探索比例增大，反之，则更多选择当前最优操作。

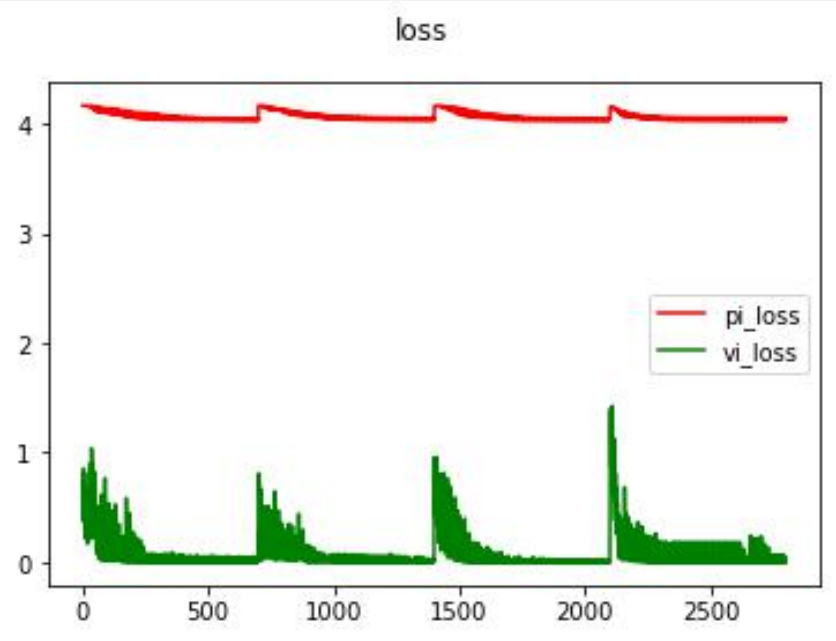
每一次完整的自我对弈的前30步，参数 τ=1 ，这是早期鼓励探索的设置。游戏剩下的步数，该参数将逐渐降低至0。如果是比赛，则直接为0。

$$\pi(a|s) = \frac{N(s,a)^{1/\tau}}{\sum_b N(s,b)^{1/\tau}}$$

成果展示与分析

Achievement display and analysis

损失函数的变化

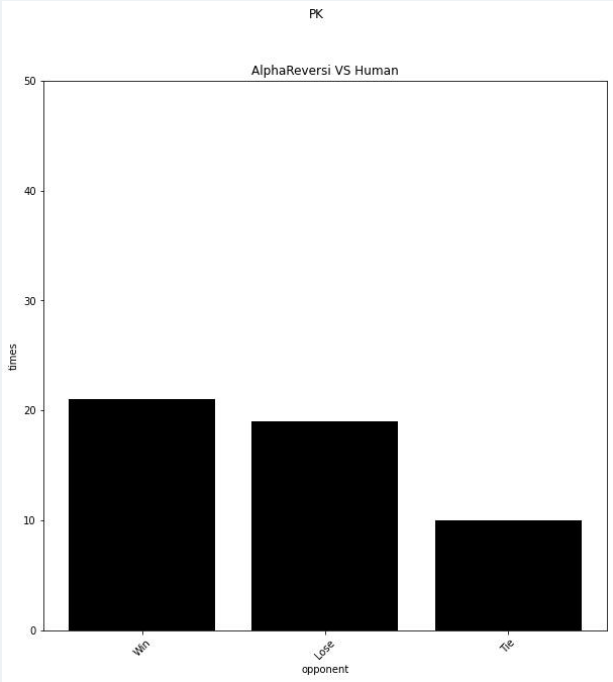


其中，青色曲线为Value Loss（Value Head的损失函数）、红色曲线为Policy Loss（Policy Head的损失函数）。

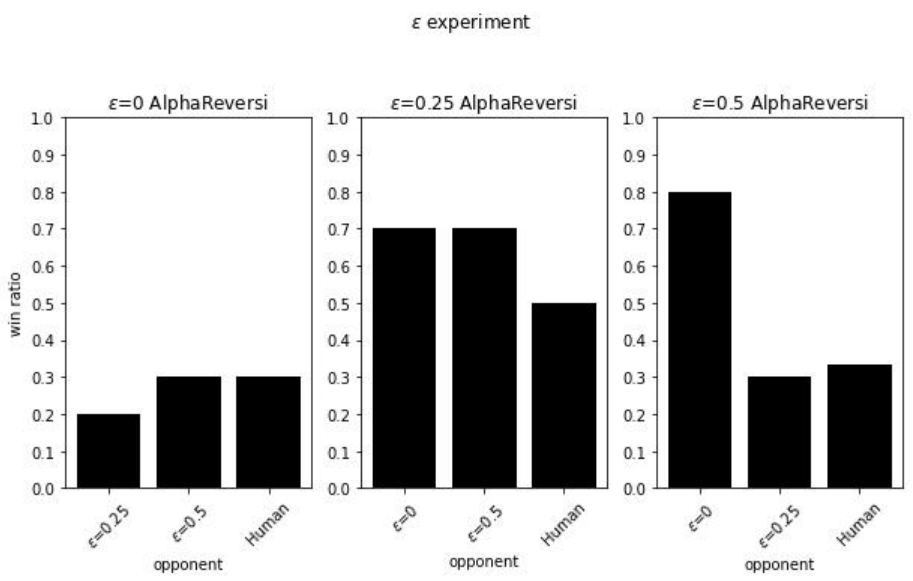
第三部分：成果展示与分析

Algorithm introduction

AlphaReversi和HumanPlayer进行30轮的游戏



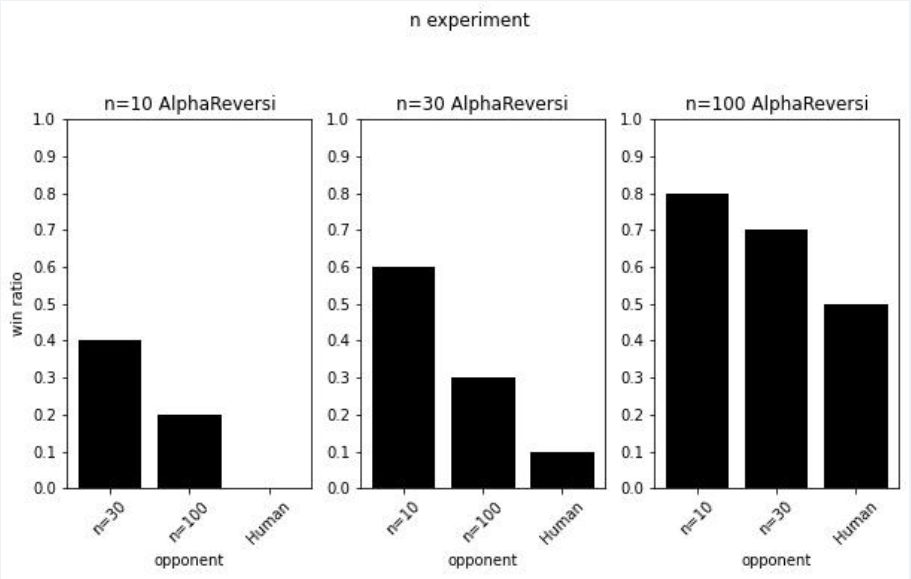
参数设置——Dirichlet噪声



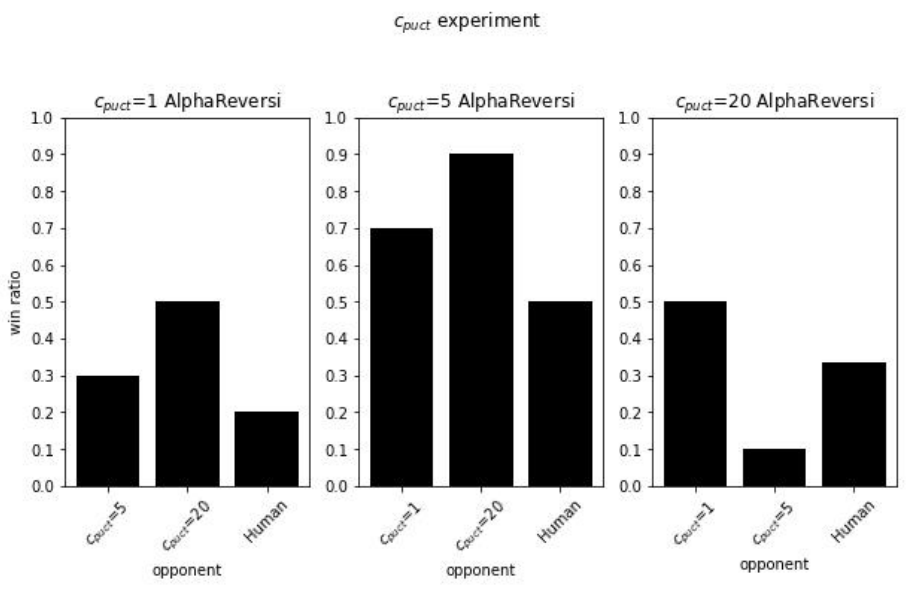
$$\epsilon = \{0，\ 0.25，\ 0.5\}$$

参数设置——MCTS的搜索次数

$$n = \{10，40，100\}$$



参数设置——用于控制Exploration程度的cput



$$c_{put} = \{1, 5, 20\}$$

总结

Summary

第四部分：总结

Summary

总结

AlphaGoZero训练得出的模型AlphaReversi搜索次数超过千次最终结果就能够打败人类棋手，证明了深度学习算法的重要性。深度学习给人工智能带来了革命性的变革，使人工智能整体水准有了质的飞跃。

AlphaGoZero摒弃了有监督学习，仅使用**强化学习**就达到了很好的效果，而且发现了以前没有被人类所有选手发现的知识。因此，AlphaGo解决了困扰机器学习的两个最重要的问题，数据的来源以及数据的质量。

小组分工

Summary

刘雪琪-汇报工作，进行结果分析
丘启圆-编程代码，进行模型训练工作
颜宇康-撰写报告