



EE228 课程大作业
2048游戏项目报告

姓名:杨子腾 学号 : 517021910683

2020年6月21日



上海交通大学

SHANGHAI JIAO TONG UNIVERSITY

项目总体情况



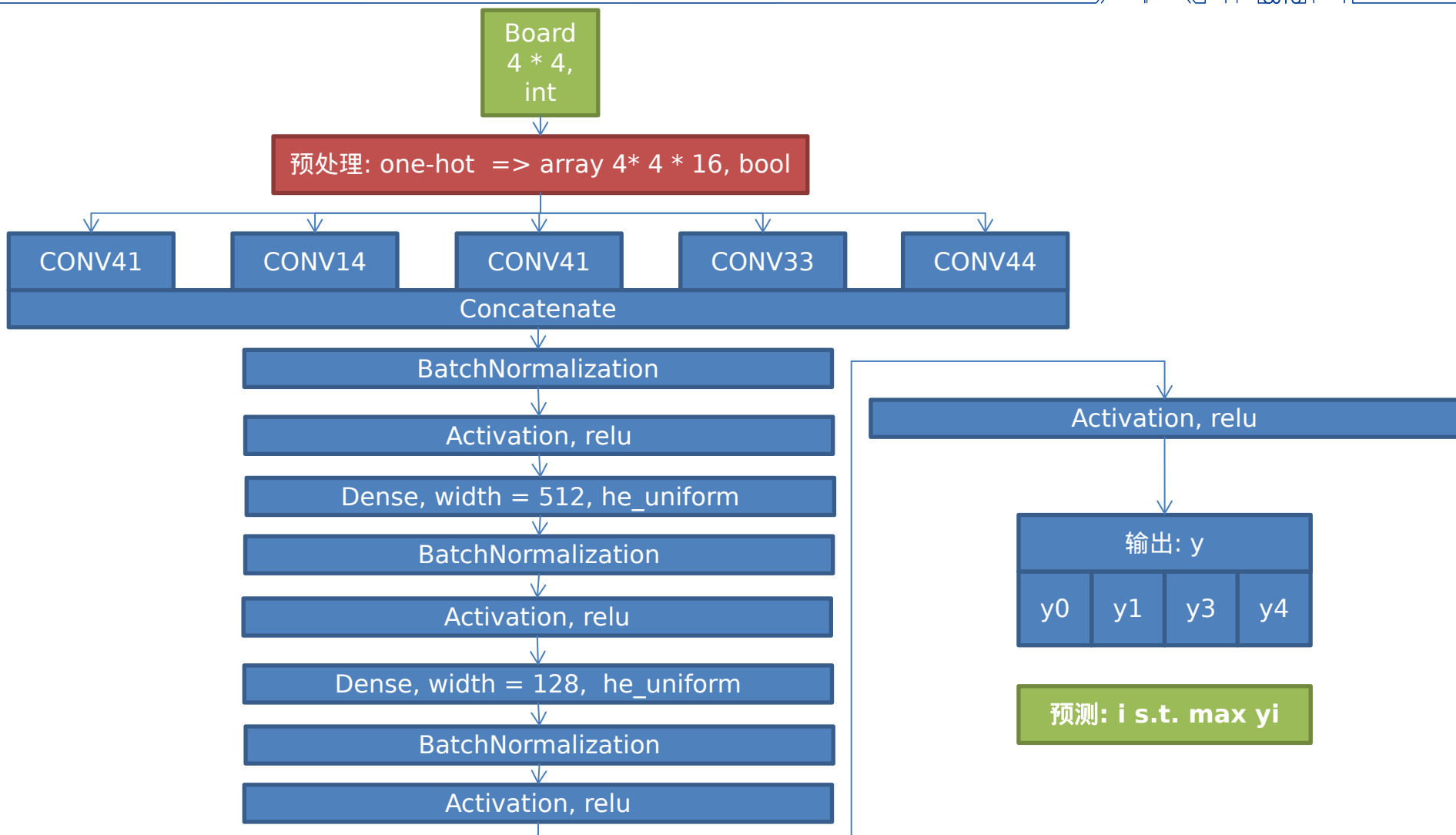
- 10轮平均分数：588.8
- 方法简述：模型为**CNN**，训练方式在线训练
- 主要使用的代码框架：Python+**Keras**神经网络框架
- 模型大小（**MB**）：19.1
- 亮点：尝试了多种不同的训练方法并对比，找到了一个较好的
- 代码链接：<https://github.com/Youngzt998/2048-api>
(github未上传不包含已经训练好的模型)

问题描述



- 给定2048游戏程序的接口，以及一个能够稳定得分2048的agent
- 训练自己新定义的agent，力求达到较高分数
- 仅使用机器学习算法进行训练

模型设计: 卷积神经网络模型



模型训练



- 设计了不同的小步训练策略，可以在训练时组合进行
- 尝试进行了不同种类的训练方式并比较各自结果
- 最终采用在线学习+模型分层的组合训练方式进行评分

```
def learn(self, itr_time=5, batch_size=128, goal=2048, dynamic_batch=False):...  
def multi_level_learn(self, batch_size=128, goal=2048):...  
def multi_level_multi_model_learn(self, itr_time, seq_=-1):...  
def improve_from_dataset(self, goal_=-1, group_=-1, go_by_self_=-1):...  
def learn_from_dataset_from_master(self, L, R, group_=-1):...
```

```
train.py  
train_batch64.py  
train_batch128.py  
train_data_in_group.py  
train_data_in_small_group.py  
train_dynamic.py  
train_improve_batch128.py  
train_multi_level.py  
train_multi_level_in_group_from_master.py  
train_multi_level_multi_model.py
```


模型训练：不同训练方式的效果

硬件信息
CPU: Intel i3
GPU: GTX 940M

- 尝试进行了不同种类的训练方式并比较各自结果
- 最终采用**在线批量学习**的组合训练方式进行评分(但直接批量在线学习效果预计更好，相同训练规模下得分预计更高)

学习方式	简述	数据量设定 (截至6.21 23:30)	大致训练时间 (截至6.21 23:30)	效果 (1000局平均分 数)	效果 (最高分)
在线学习	模型自己作决策，每 128组训练一次	$\geq 200,000$ (局游 戏)	$\geq 140h$	≥ 300 (100,000局之后，增速 极其缓慢)	1024
少量数据在线学习 + 批量数据离线学习	在线学习在稳定到 256分以后每两千局 生成一组大数据学习	\geq 200,000局游戏 + 2,000,000组批量 数据	$\geq 160h$	≥ 430 (平均分目前仍在缓慢 线 性提高[20分每15小时])	1024 仍在训练中..
批量在线学习	根据被训练的模型游 戏，每一万局游戏生 成一组数据进行学习	$\geq 40,000$ (局游戏)	$\geq 50h$ (主要耗时在数据生 成)	≥ 500 (已进行三轮，平均分目 前仍在缓慢 线性提高[增 加100分，耗时20小时])	1024 仍在训练中..
在线学习(多个模型)	按照当前分数为分成 三个模型进行在线学 习	$\geq 200,000$ (局游 戏)		≤ 200	256
数据集学习(多个模 型)	按照当前分数分成三 个模型离线批量学习	$\geq 2,000,000$ 组批 量数据		仍在运行中..	仍在运行中..

模型训练：在线学习中batch_size的影响



- 在线训练时，尝试探索了不同大小的batch_size对训练结果的影响
- 如果batch_size过小，会影响训练效果；为1时几乎没有效果
- 一个数据一个数据地训练几乎没有效果

batch_size	数据量设定	效果 (1000局平均分数)	效果 (最高分)
1	$\geq 200,000$ (局游戏)	< 100 (性能一直不超过随机Agent)	64
32	$\geq 200,000$ (局游戏)	≥ 250	1024
128	$\geq 200,000$ (局游戏)	≥ 350	1024
256	$\geq 200,000$ (局游戏)	≥ 300	1024

性能分析



最终采用在线学习+批量训练的组合训练方式进行评分

Agent成绩分布(1000局游戏测试, 统计超过 2^i 的分数, $i = 3, \dots, 10$)

1024	512	256	128	64	32	16	8
154	747	976	990	998	1000	1000	1000

单步预测时长: 4.26 ms / 步

模型大小: 19.1 MB

代码结构： LearningAgent



- 继承Agent类——LearningAgent
- 类成员除了父类规定的原有成员以外，还包括了：
 - 新模型生成函数
 - 模型，在初始化时尝试从文件导入，若失败则创建新模型
 - “教练”(即提供的稳定达到2048的Agent)
 - 进行不同训练策略的成员函数

代码结构：训练脚本



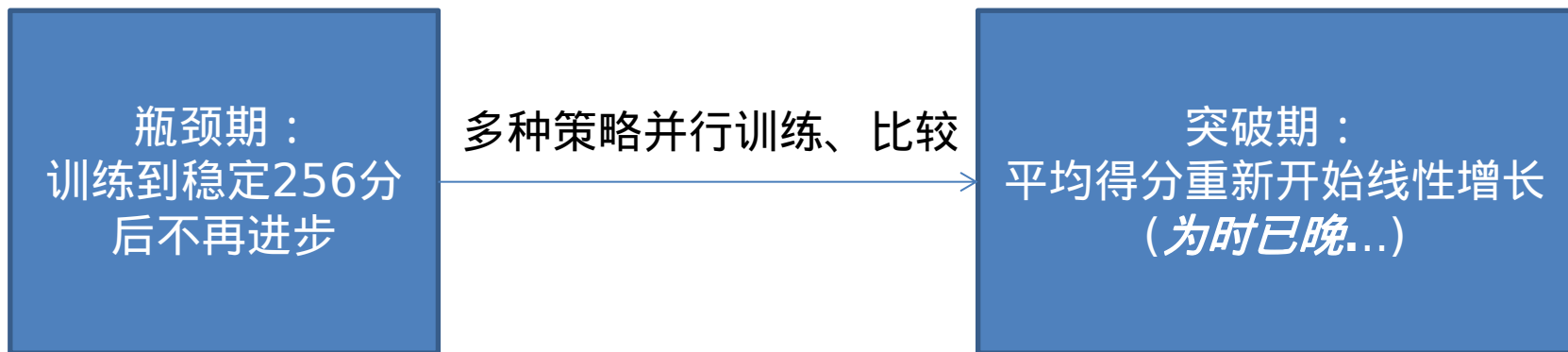
- 定义了多种不同训练策略的训练脚本，通过更改LearningAgent的训练函数以及参数来使用不同策略
- 图为达到较好效果的训练方式(以10000局游戏为一个周期进行在线训练)，也可以先进行小批量数据的在线训练，在达到较高分数后加大力度

```
if __name__ == '__main__':  
    game = Game(4, 2048)  
  
    # default: 32  
    batch_size = 128  
  
    # set to true if need to create a new model to train  
    # warning: this would overwrite existing previous model if game time reaches 500 in training  
    new_model = False  
  
    agent = LearningAgent(game=game, display=None, tch_search_fun=None,  
                           new_model=new_model,  
                           model_path="./model/model_data_in_group.h5")  
    agent.improve_from_dataset(group=10000)
```

讨论：实验过程的经验教训



- Python属于动态语言类型，一些接口的数据结构类型不显式声明，所以很容易造成参数传递出错，尤其是keras使用的**numpy的数据维度**
- 应当注意每训练一段时间就**保存**一次模型，以避免某些意外的错误浪费了训练时间
- **充分利用并行**的功能可以加快不少训练速度，即使硬件本身不优越
- 前期每一局数据量小，后期每一局数据量大，需要实时调整以避免数据过大，否则难以决定是中断程序改变数据规模还是继续训练



讨论：关于训练过程



- 在线学习的代价极高，且因为数据实时产生，GPU并行计算难以显著提高训练速度
- 在线学习时，因为数据产生缓慢，所以即使是普通的个人电脑，一个模型的训练并不会占有所有计算资源，可以同时运行多个脚本进行“并行”训练，最后对不同的模型进行评估(前期没有意识到这一点，消耗了大量不必要的时间进行单进程训练...)
- 是否始终保持同一种训练策略是值得商榷的



上海交通大学
SHANGHAI JIAO TONG UNIVERSITY

Thank You



上海交通大学
SHANGHAI JIAO TONG UNIVERSITY





附页（没有请删除）

