如何训练你自己的Leela Zero权重

(By 袁泉)

# 准备你的棋谱

## 棋谱来源

棋谱有很多来源，我找到的一些棋谱来源：

computer-go-dataset: <https://github.com/yenw/computer-go-dataset>

LeelaZero: <https://sjeng.org/zero/>

Aya-Self play: <http://www.yss-aya.com/ayaself/ayaself.html>

CGOS: <http://www.yss-aya.com/cgos/19x19/archive.html>

KGS: <https://u-go.net/gamerecords/>

新浪: <http://duiyi.sina.com.cn/gibo/new_gibo.asp>

Gokifu: <http://gokifu.com/>

野狐: <http://www.foxwq.com/qipu>

## 棋谱格式整理

以上的棋谱，有很多种格式，最后我们都要整理成sgf格式，然后将所有的sgf棋谱文件打包到一个大的sgf文件中(简单的文件合并就可以了)，命令行如下：

Windows : copy \*.sgf train.sgf

Linux: cat \*.sgf train.sgf

生成的train.sgf建议最少包含25W张围棋棋谱，50W张更好。建议用50W张棋谱进行第一轮训练，然后用25W张高段棋谱进行第二轮训练，这样生成的网络更加稳定。

# 准备你的网络

如果你只需要继续训练5\*64、6\*128或者10\*128的网络，那么你可以直接从官网下载对应结构的网络，继续训练就可以了：

<http://zero.sjeng.org/networks/>

如果你想训练其它结构不同的网络，可以用net2net工具进行扩网（注意：只能从小的网络扩成大的网络）：

1. 先clone整个leelazero项目，进行编译（过程略）
2. 用net2net工具扩网，以10\*128网络扩为20\*192网络为例：

下载10\*128网络并解压 10\_128.txt

cd leela-zero/training/tf

python net2net.py 10 64 10\_128.txt

这样就在10\*128网络上，增加了10层，每层扩展了64个filter，变成了20\*192的网络

# 生成训练数据

以下命令行比较简单，请自己理解，可以参考官网说明(https://github.com/gcp/leela-zero)：

cd <leela-zero>/training/tf

mkdir round1

cd round1

copy <sgfdir>/train.sgf .

copy <leela-zero>/src/leelaz .

copy <weightdir>/20\_192.txt .

./leelaz –w 20\_192.txt

> dump\_supervised train.sgf train.out

> exit

请耐心等待，50W盘棋谱大概需要8-12小时才能生成完整的测试数据。

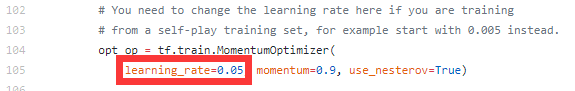
# 开始训练

1. 修改训练的网络结构

tfprocess.py   

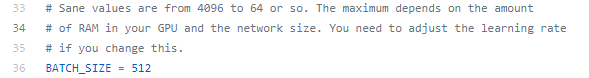

1. 修改learning\_rate的参数

tfprocess.py



1. 修改批次大小（如果你的显卡显存不够）

**parse.py**



1. 生成初始训练模型(如果你直接从刚才生成的网络进行训练，这一步可以跳过)

python3 net\_to\_model.py 20\_192.txt

1. 进行训练

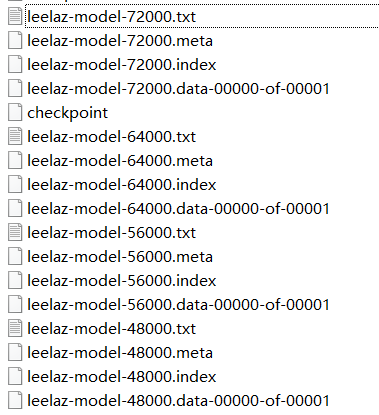
从最开始的网络开始训练

training/tf/parse.py train.out

从某个断点的批次开始训练

training/tf/parse.py train.out leelaz-model-batchnumber

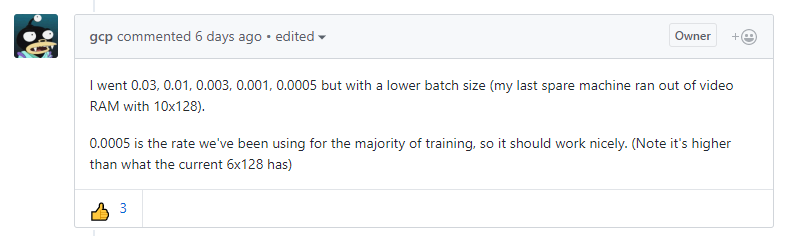
如果成功，你将会看到每隔8k步迭代后，会生成一个权重，恭喜，你自己的第一个权重训练出来了！



# 关于扩网的训练步骤

第4步中，要成功扩网并训练一个网络是一个反复迭代的过程，可以看gcp关于扩网的训练过程描述：

<https://github.com/gcp/leela-zero/issues/965>



所以要成功训练一个网络，第4步需要反复进行迭代，按照gcp给出的方案，如果扩网，建议如下进行训练：

1. learning\_rate 0.03迭代128k次
2. learning\_rate 0.01迭代128k次
3. learning\_rate 0.003迭代128k次
4. learning\_rate 0.001迭代128k次
5. learning\_rate 0.0005迭代128k次

如果扩网结束，在该网络的基础上训练一个更优化的棋谱，建议进行如下训练：

1. learning\_rate 0.005迭代128k次
2. learning\_rate 0.001迭代128k次
3. learning\_rate 0.0005迭代128k次

最后，还可以试试混血能不能更进一步提高网络的棋力：

<https://github.com/pangafu/Hybrid_LeelaZero>

混血结束后，还可以进一步训练，反复迭代，直到找到你认为最优的权重。