1. 执行policy gradient分别去run CartPole-v1，InvertedPendulum-v2的环境

执行如下代码：

python "F:\专业\reinforcement learning\CS285\_Fa19\_Deep\_Reinforcement\_Learning-master\hw2\cs285\scripts\run\_hw2\_policy\_gradient.py" --env\_name CartPole-v1 --exp\_name test\_pg\_cartpole

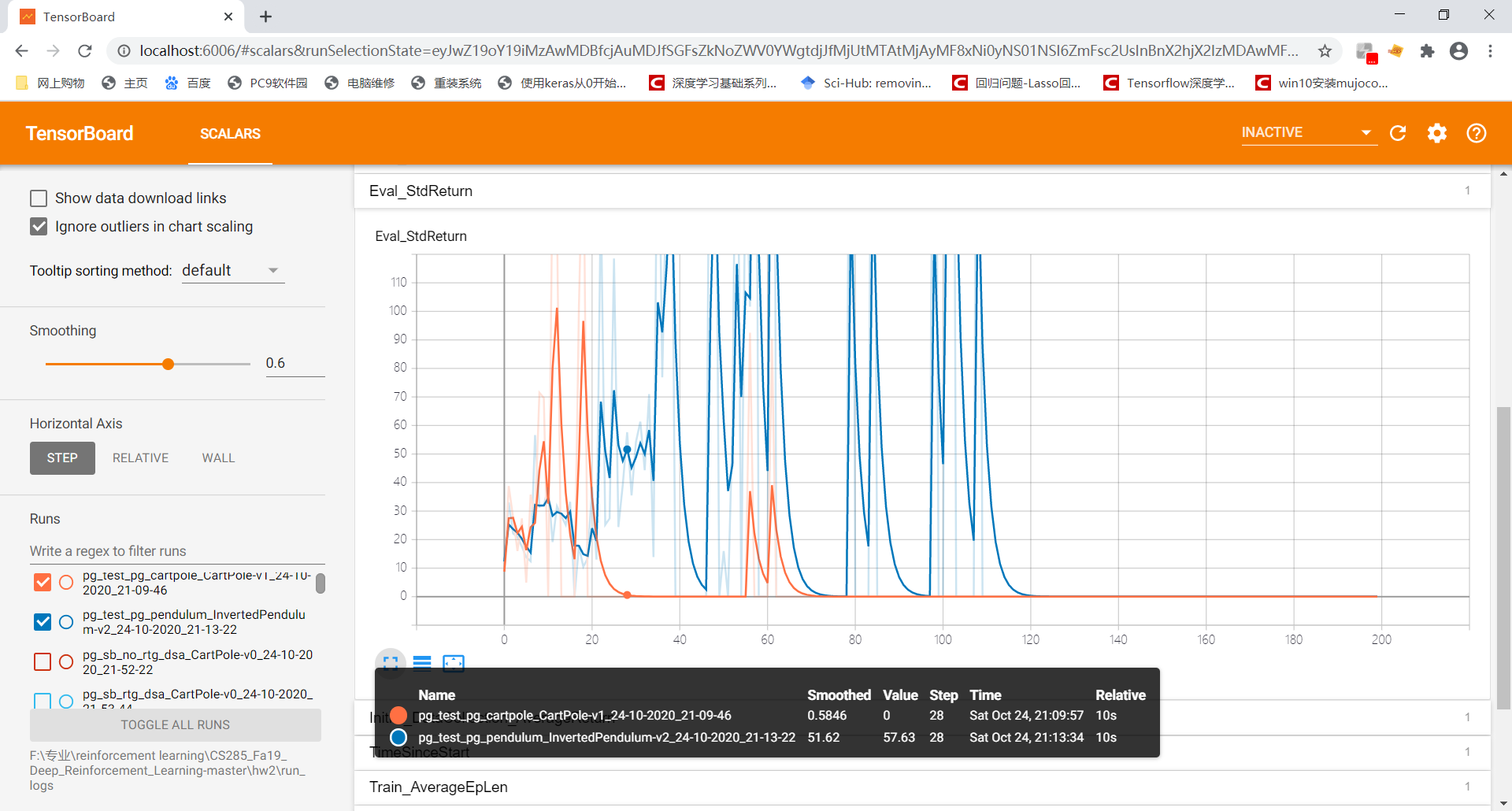
python "F:\专业\reinforcement learning\CS285\_Fa19\_Deep\_Reinforcement\_Learning-master\hw2\cs285\scripts\run\_hw2\_policy\_gradient.py" --env\_name InvertedPendulum-v2 --exp\_name test\_pg\_pendulum

可视化结果：

平均回报如图所示，其中黄线是CartPole-v1环境，蓝线是InvertedPendulum-v2环境：



方差如图所示, 其中黄线是CartPole-v1环境，蓝线是InvertedPendulum-v2环境：



1. 用PG algorithm对CartPole-v0环境进行多次试验

执行如下代码

python "F:\专业\reinforcement learning\CS285\_Fa19\_Deep\_Reinforcement\_Learning-master\hw2\cs285\scripts\run\_hw2\_policy\_gradient.py" --env\_name CartPole-v0 -n 100 -b 1000 -dsa --exp\_name sb\_no\_rtg\_dsa

python "F:\专业\reinforcement learning\CS285\_Fa19\_Deep\_Reinforcement\_Learning-master\hw2\cs285\scripts\run\_hw2\_policy\_gradient.py" --env\_name CartPole-v0 -n 100 -b 1000 -rtg -dsa --exp\_name sb\_rtg\_dsa

python "F:\专业\reinforcement learning\CS285\_Fa19\_Deep\_Reinforcement\_Learning-master\hw2\cs285\scripts\run\_hw2\_policy\_gradient.py" --env\_name CartPole-v0 -n 100 -b 1000 -rtg --exp\_name sb\_rtg\_na

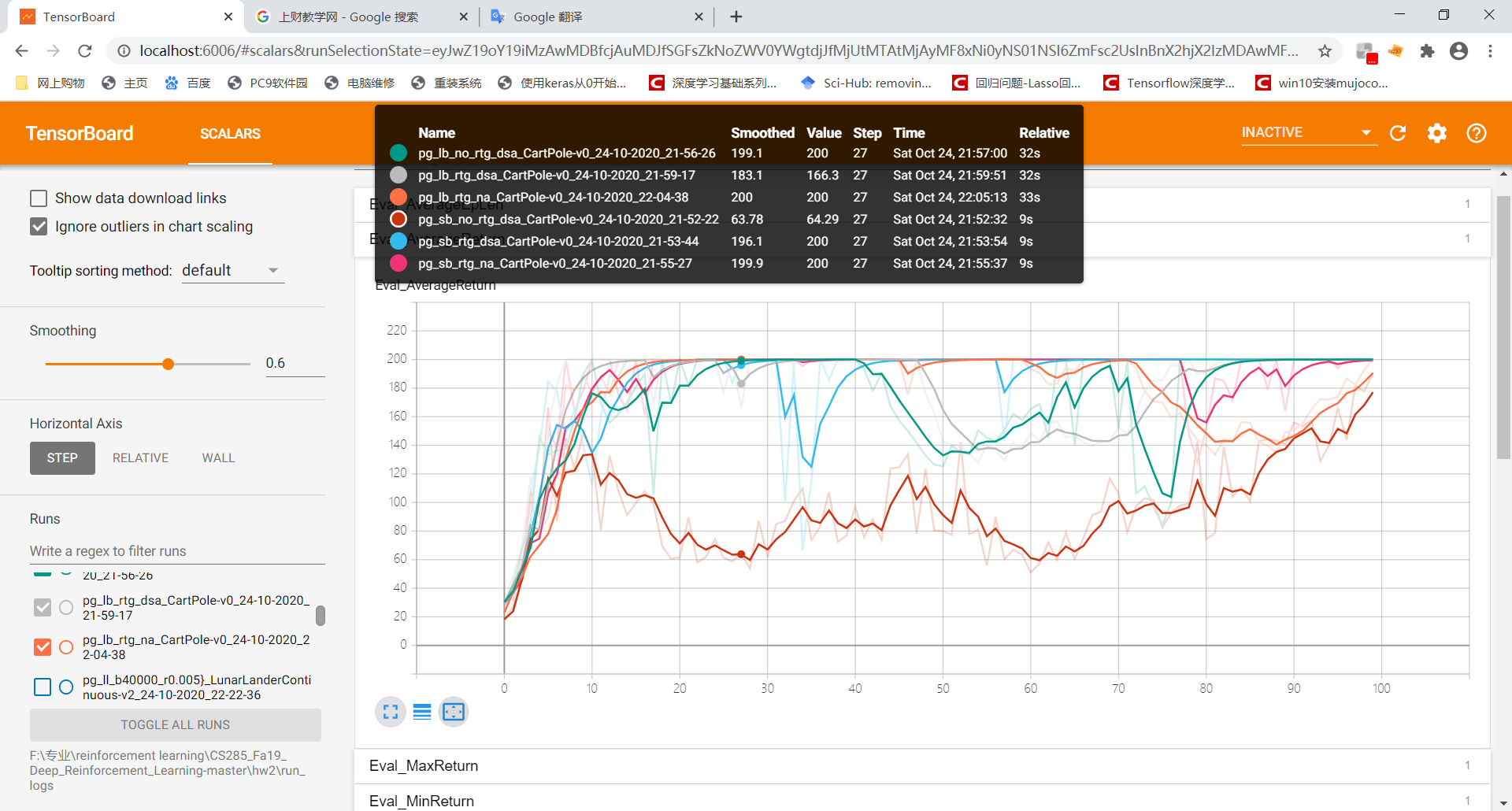
python "F:\专业\reinforcement learning\CS285\_Fa19\_Deep\_Reinforcement\_Learning-master\hw2\cs285\scripts\run\_hw2\_policy\_gradient.py" --env\_name CartPole-v0 -n 100 -b 5000 -dsa --exp\_name lb\_no\_rtg\_dsa

python "F:\专业\reinforcement learning\CS285\_Fa19\_Deep\_Reinforcement\_Learning-master\hw2\cs285\scripts\run\_hw2\_policy\_gradient.py" --env\_name CartPole-v0 -n 100 -b 5000 -rtg -dsa --exp\_name lb\_rtg\_dsa

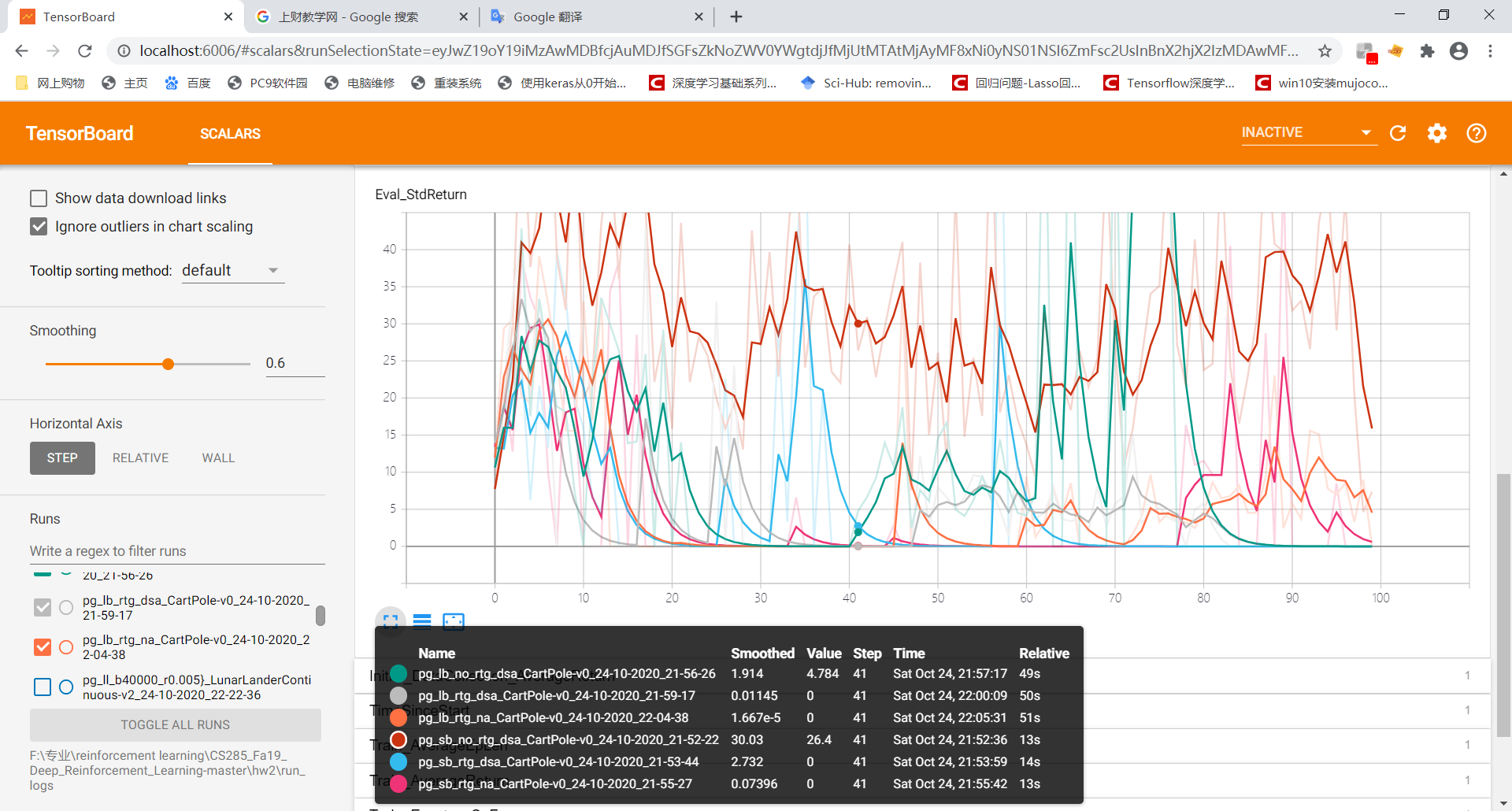
python "F:\专业\reinforcement learning\CS285\_Fa19\_Deep\_Reinforcement\_Learning-master\hw2\cs285\scripts\run\_hw2\_policy\_gradient.py" --env\_name CartPole-v0 -n 100 -b 5000 -rtg --exp\_name lb\_rtg\_na

可视化结果

六次试验的平均回报如图所示：



六次试验的方差如图所示：



1. 对InvertedPendulum-v2环境进行试验选取最优超参数

执行代码

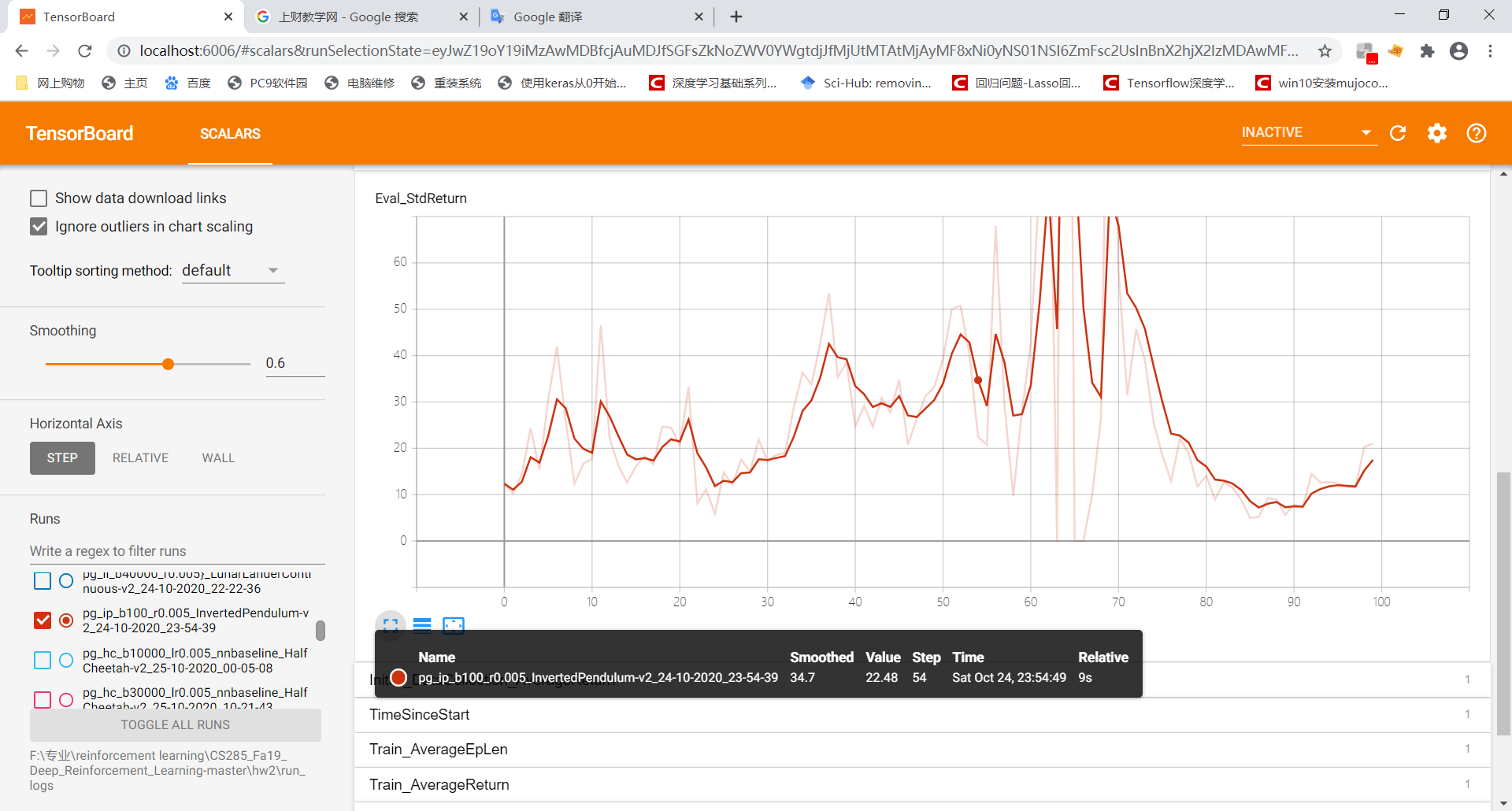
python "F:\专业\reinforcement learning\CS285\_Fa19\_Deep\_Reinforcement\_Learning-master\hw2\cs285\scripts\run\_hw2\_policy\_gradient.py" --env\_name InvertedPendulum-v2 --ep\_len 1000 --discount 0.9 -n 100 -l 2 -s 64 -b 100 -lr 0.005 -rtg --exp\_name ip\_b100\_r0.005

最优超参数为 batch\_size选取100，learning rate选取0.005，可视化结果如下：

平均回报：



方差：



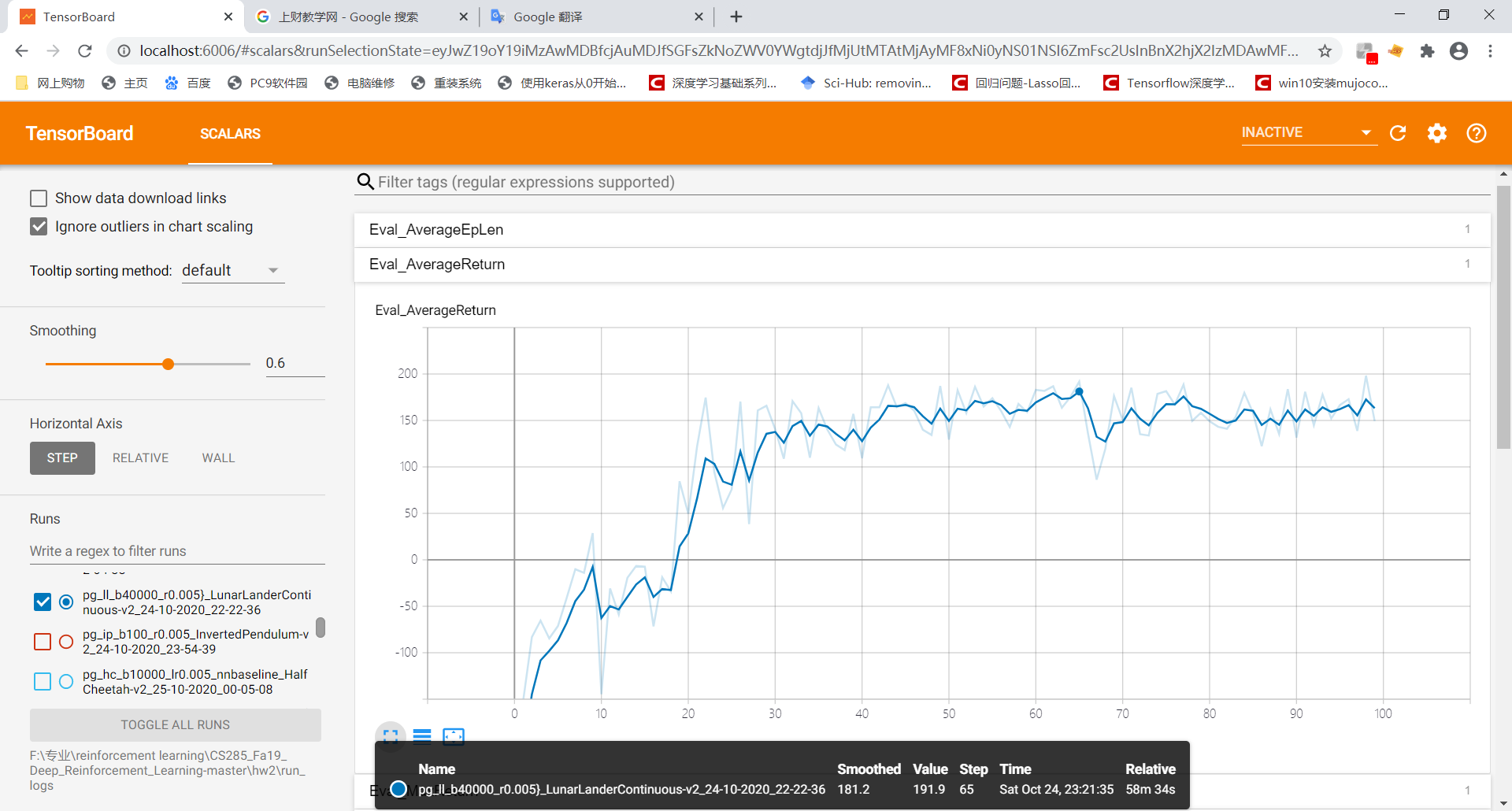
1. 用policy gradient去解LunarLanderContinuous-v2环境

执行代码：

python "F:\专业\reinforcement learning\CS285\_Fa19\_Deep\_Reinforcement\_Learning-master\hw2\cs285\scripts\run\_hw2\_policy\_gradient.py" --env\_name LunarLanderContinuous-v2 --ep\_len 1000 --discount 0.99 -n 100 -l 2 -s 64 -b 40000 -lr 0.005 -rtg --nn\_baseline --exp\_name ll\_b40000\_r0.005}

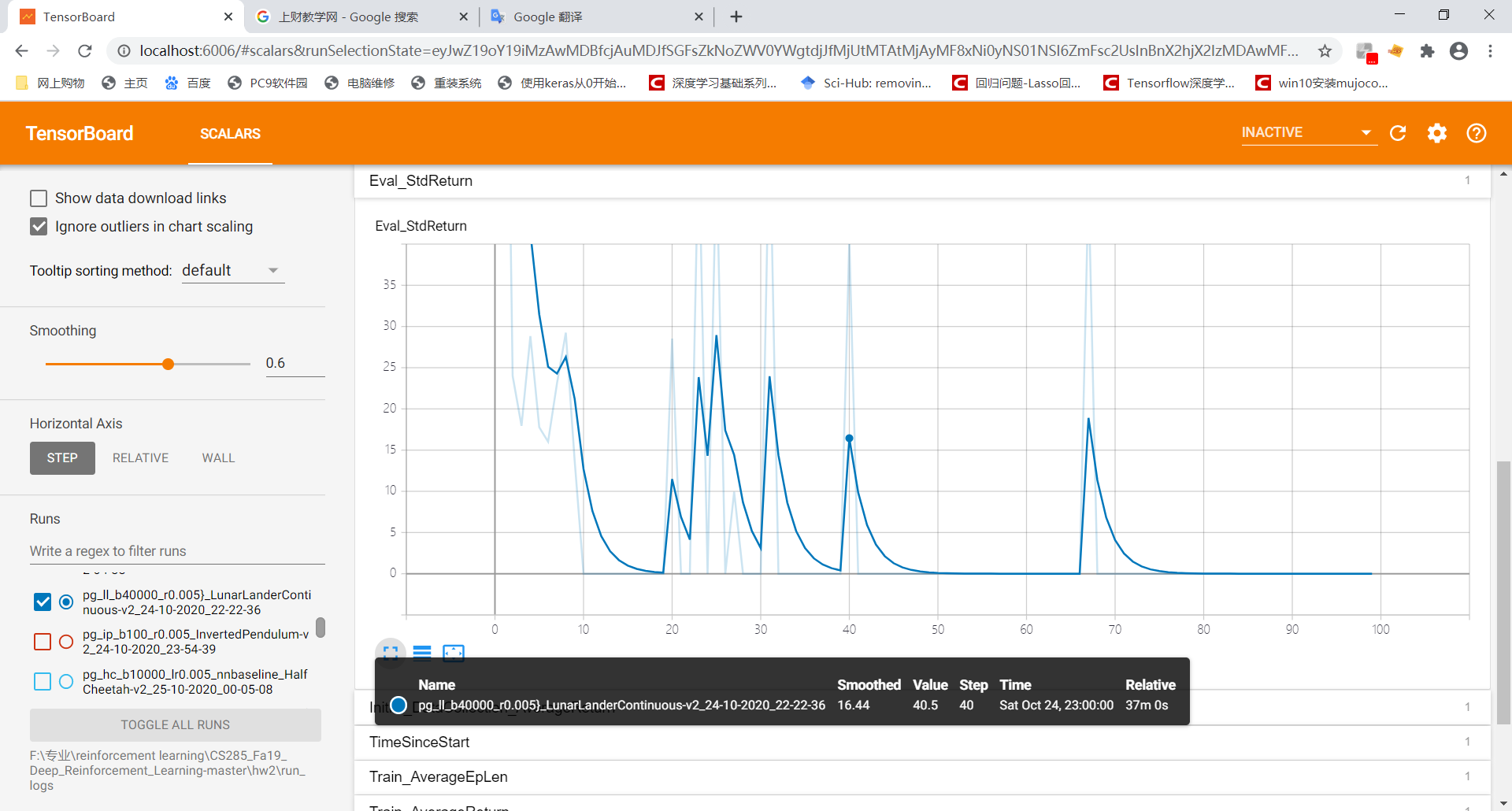
可视化结果：

平均回报：



You should expect to achieve an average return of around 180，从图上看好像差不多

方差：



1. 用policy gradient解HalfCheetah-v2环境，寻找最优的超参数组合，其中batch sizes∈ [10000,30000,50000]，learning rates r ∈ [0.005,0.01,0.02]

执行代码：

python "F:\专业\reinforcement learning\CS285\_Fa19\_Deep\_Reinforcement\_Learning-master\hw2\cs285\scripts\run\_hw2\_policy\_gradient.py" --env\_name HalfCheetah-v2 --ep\_len 150 --discount 0.95 -n 100 -l 2 -s 32 -b 10000 -lr 0.005 --video\_log\_freq -1 --reward\_to\_go --nn\_baseline --exp\_name hc\_b10000\_lr0.005\_nnbaseline

python "F:\专业\reinforcement learning\CS285\_Fa19\_Deep\_Reinforcement\_Learning-master\hw2\cs285\scripts\run\_hw2\_policy\_gradient.py" --env\_name HalfCheetah-v2 --ep\_len 150 --discount 0.95 -n 100 -l 2 -s 32 -b 30000 -lr 0.005 --video\_log\_freq -1 --reward\_to\_go --nn\_baseline --exp\_name hc\_b30000\_lr0.005\_nnbaseline

python "F:\专业\reinforcement learning\CS285\_Fa19\_Deep\_Reinforcement\_Learning-master\hw2\cs285\scripts\run\_hw2\_policy\_gradient.py" --env\_name HalfCheetah-v2 --ep\_len 150 --discount 0.95 -n 100 -l 2 -s 32 -b 50000 -lr 0.005 --video\_log\_freq -1 --reward\_to\_go --nn\_baseline --exp\_name hc\_b50000\_lr0.005\_nnbaseline

python "F:\专业\reinforcement learning\CS285\_Fa19\_Deep\_Reinforcement\_Learning-master\hw2\cs285\scripts\run\_hw2\_policy\_gradient.py" --env\_name HalfCheetah-v2 --ep\_len 150 --discount 0.95 -n 100 -l 2 -s 32 -b 10000 -lr 0.01 --video\_log\_freq -1 --reward\_to\_go --nn\_baseline --exp\_name hc\_b10000\_lr0.01\_nnbaseline

python "F:\专业\reinforcement learning\CS285\_Fa19\_Deep\_Reinforcement\_Learning-master\hw2\cs285\scripts\run\_hw2\_policy\_gradient.py" --env\_name HalfCheetah-v2 --ep\_len 150 --discount 0.95 -n 100 -l 2 -s 32 -b 30000 -lr 0.01 --video\_log\_freq -1 --reward\_to\_go --nn\_baseline --exp\_name hc\_b30000\_lr0.01\_nnbaseline

python "F:\专业\reinforcement learning\CS285\_Fa19\_Deep\_Reinforcement\_Learning-master\hw2\cs285\scripts\run\_hw2\_policy\_gradient.py" --env\_name HalfCheetah-v2 --ep\_len 150 --discount 0.95 -n 100 -l 2 -s 32 -b 50000 -lr 0.01 --video\_log\_freq -1 --reward\_to\_go --nn\_baseline --exp\_name hc\_b50000\_lr0.01\_nnbaseline

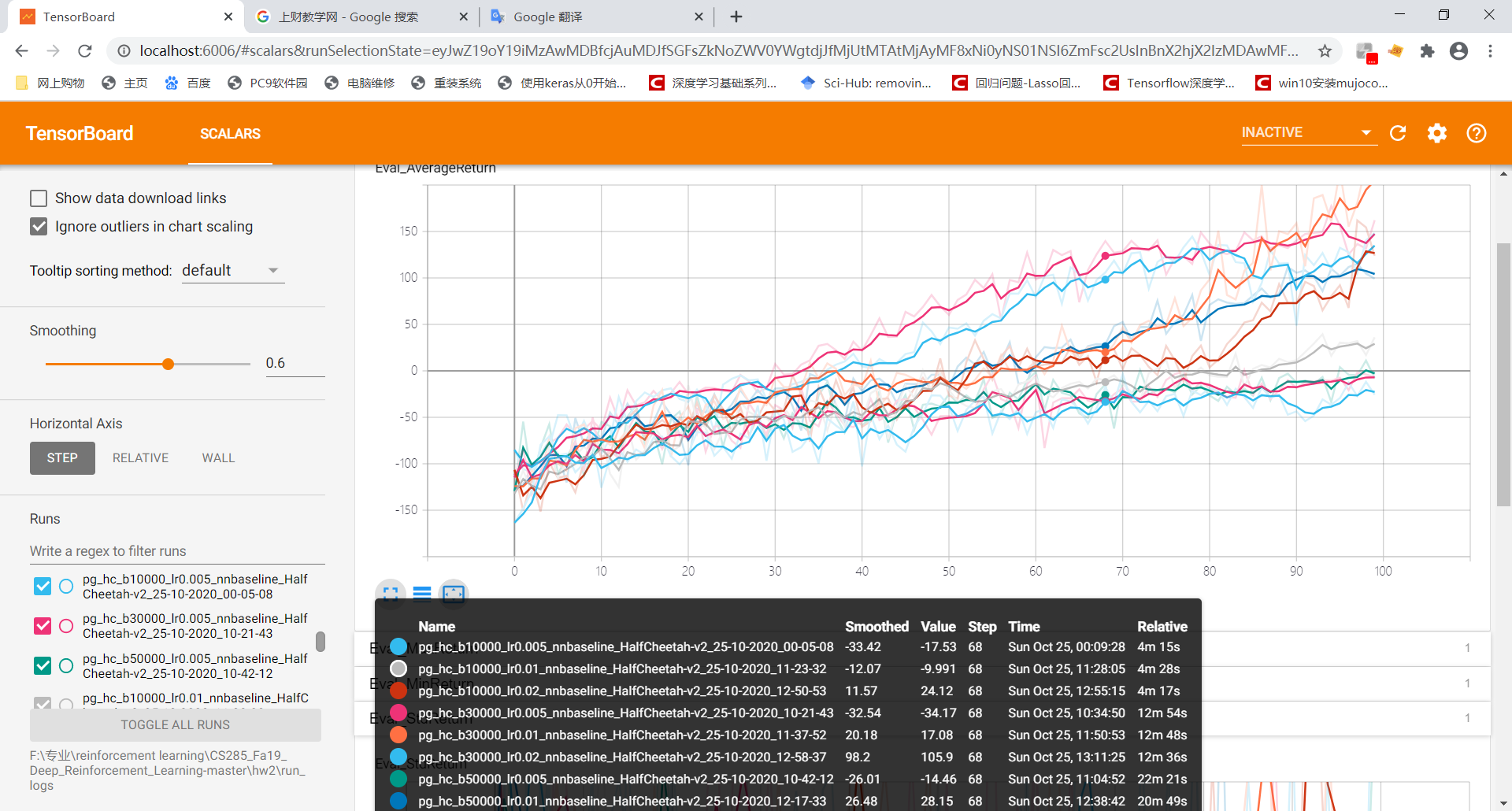
python "F:\专业\reinforcement learning\CS285\_Fa19\_Deep\_Reinforcement\_Learning-master\hw2\cs285\scripts\run\_hw2\_policy\_gradient.py" --env\_name HalfCheetah-v2 --ep\_len 150 --discount 0.95 -n 100 -l 2 -s 32 -b 10000 -lr 0.02 --video\_log\_freq -1 --reward\_to\_go --nn\_baseline --exp\_name hc\_b10000\_lr0.02\_nnbaseline

python "F:\专业\reinforcement learning\CS285\_Fa19\_Deep\_Reinforcement\_Learning-master\hw2\cs285\scripts\run\_hw2\_policy\_gradient.py" --env\_name HalfCheetah-v2 --ep\_len 150 --discount 0.95 -n 100 -l 2 -s 32 -b 30000 -lr 0.02 --video\_log\_freq -1 --reward\_to\_go --nn\_baseline --exp\_name hc\_b30000\_lr0.02\_nnbaseline

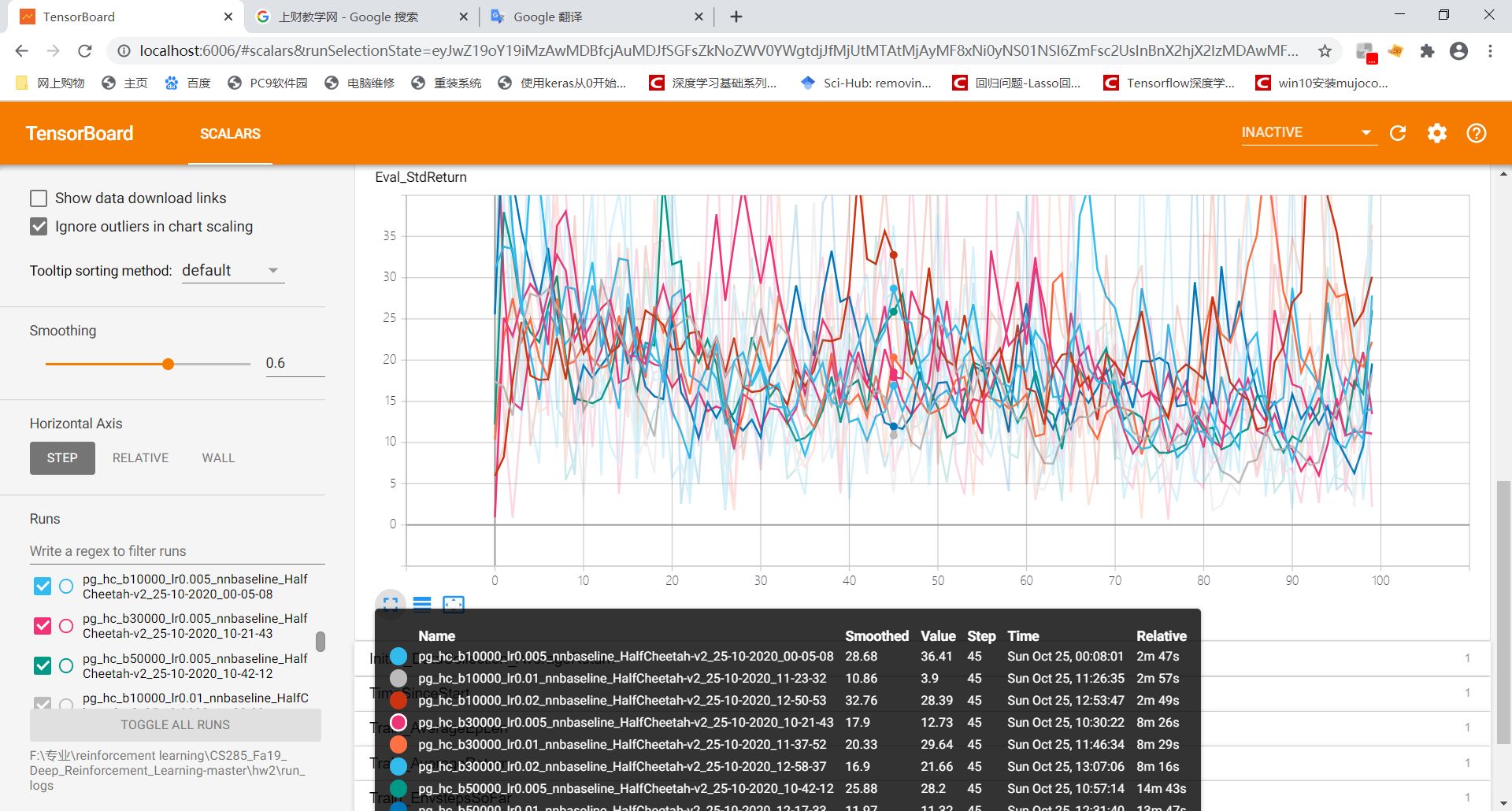
python "F:\专业\reinforcement learning\CS285\_Fa19\_Deep\_Reinforcement\_Learning-master\hw2\cs285\scripts\run\_hw2\_policy\_gradient.py" --env\_name HalfCheetah-v2 --ep\_len 150 --discount 0.95 -n 100 -l 2 -s 32 -b 50000 -lr 0.02 --video\_log\_freq -1 --reward\_to\_go --nn\_baseline --exp\_name hc\_b50000\_lr0.02\_nnbaseline

我们分别选取所有的超参数组合run一次我们的policy gradient，将结果可视化出来，选取其中最优的超参数组合。可视化结果如下：

平均回报:



方差：



根据我们的可视化结果，综合考虑方差和平均回报得出最优的超参数组合为（30000，0.01），（30000，0.02），（50000,0.02），这三个超参数组合的效果是差不多的，考虑到batch size越小越好，learning rate越大越好，我们最终选取的最优超参数组合是（30000,0.02）

1. 将我们找到的最优超参数组合（30000,0.02）run下面的几个commands

执行代码：

python "F:\专业\reinforcement learning\CS285\_Fa19\_Deep\_Reinforcement\_Learning-master\hw2\cs285\scripts\run\_hw2\_policy\_gradient.py" --env\_name HalfCheetah-v2 --ep\_len 150 --discount 0.95 -n 100 -l 2 -s 32 -b 30000 -lr 0.02 --exp\_name hc\_b30000\_r0.02

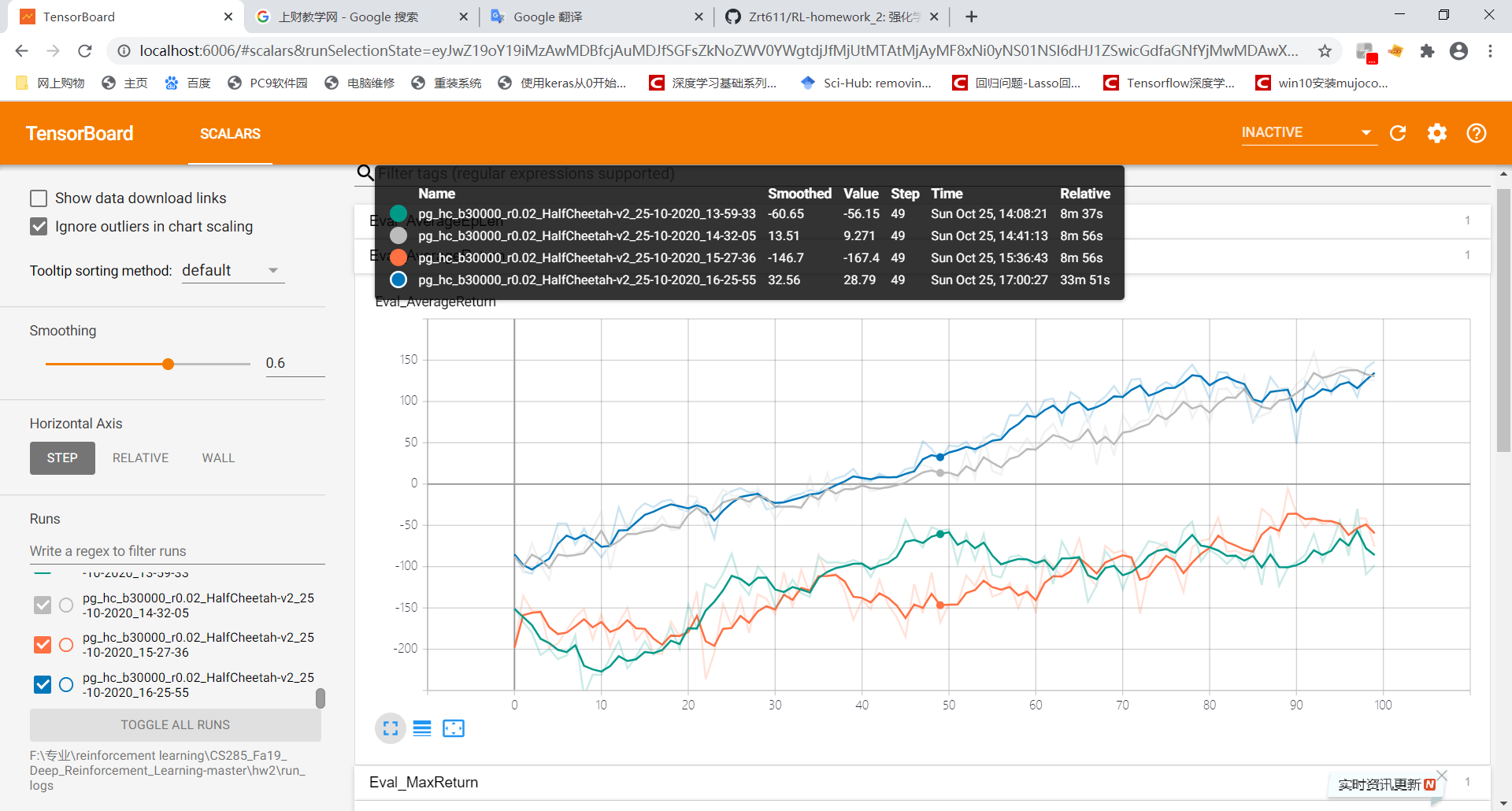
python "F:\专业\reinforcement learning\CS285\_Fa19\_Deep\_Reinforcement\_Learning-master\hw2\cs285\scripts\run\_hw2\_policy\_gradient.py" --env\_name HalfCheetah-v2 --ep\_len 150 --discount 0.95 -n 100 -l 2 -s 32 -b 30000 -lr 0.02 -rtg --exp\_name hc\_b30000\_r0.02

python "F:\专业\reinforcement learning\CS285\_Fa19\_Deep\_Reinforcement\_Learning-master\hw2\cs285\scripts\run\_hw2\_policy\_gradient.py" --env\_name HalfCheetah-v2 --ep\_len 150 --discount 0.95 -n 100 -l 2 -s 32 -b 30000 -lr 0.02 --nn\_baseline --exp\_name hc\_b30000\_r0.02

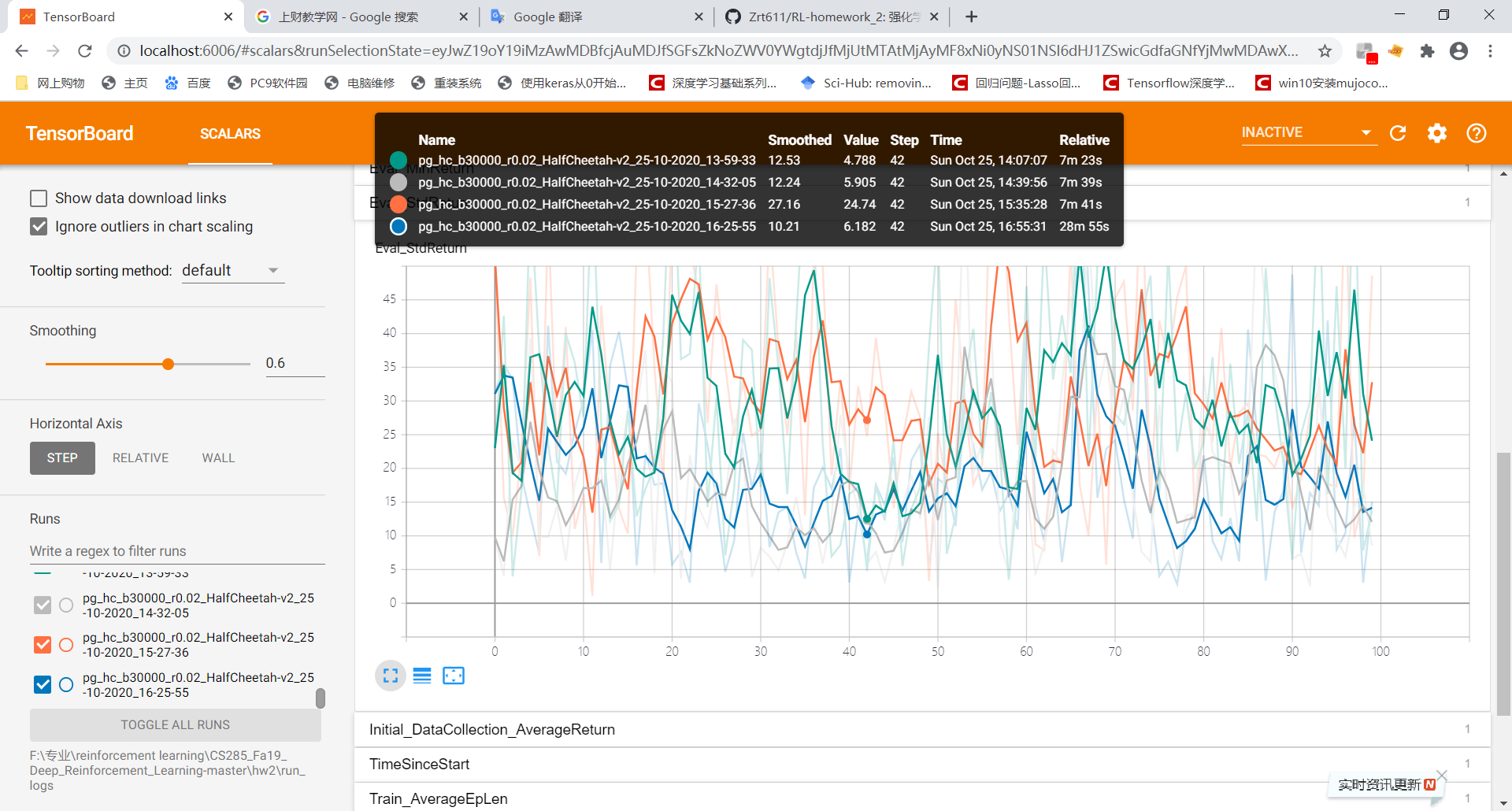
python "F:\专业\reinforcement learning\CS285\_Fa19\_Deep\_Reinforcement\_Learning-master\hw2\cs285\scripts\run\_hw2\_policy\_gradient.py" --env\_name HalfCheetah-v2 --ep\_len 150 --discount 0.95 -n 100 -l 2 -s 32 -b 30000 -lr 0.02 -rtg --nn\_baseline --exp\_name hc\_b30000\_r0.02

可视化结果如下：

平均回报：



方差：



Tensorboard可视化执行代码:

tensorboard --logdir="F:\专业\reinforcement learning\CS285\_Fa19\_Deep\_Reinforcement\_Learning-master\hw2\run\_logs" --host localhost