Reinforcement Learning Meets Balls Into Bins

ANDOR VÁRI-KAKAS

SUPERVISORS: DR THOMAS SAUERWALD, DIMITRIS LOS

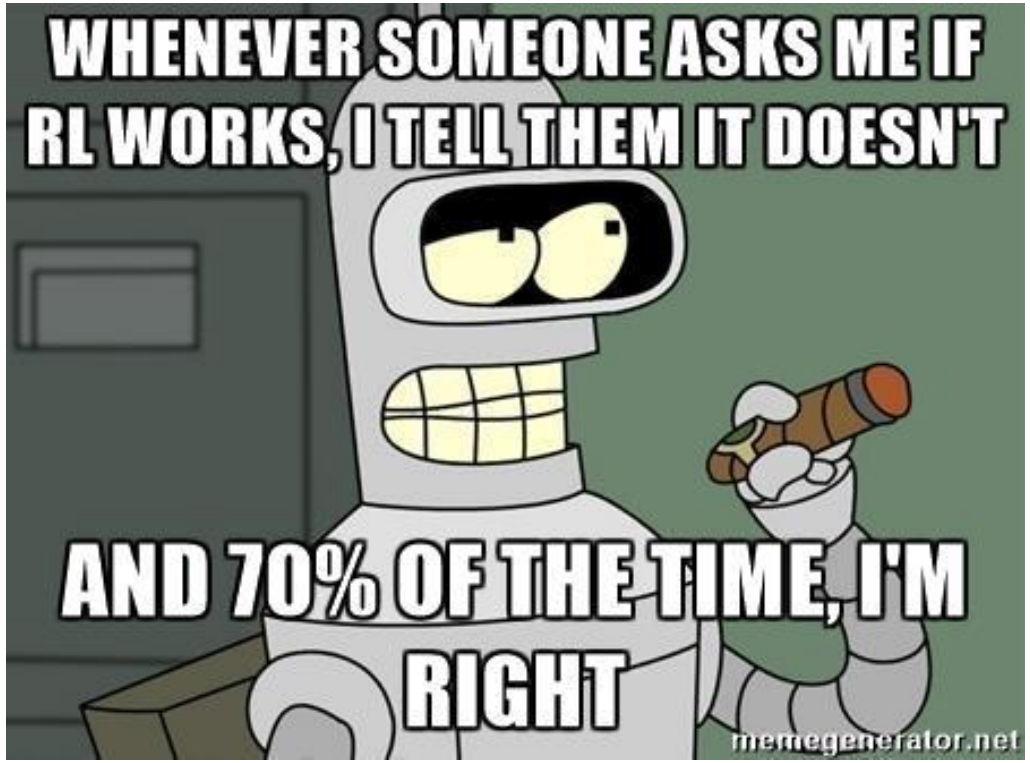
16 FEBRUARY 2022

Overview of the project

- Randomised load balancing -> balls into bins
- Many versions e.g. Two Thinning -> free parameters to optimise
- Available literature only for large values of bins, balls
- My project: use reinforcement learning for optimising the parameters
- Compare to classical algorithms

Progress

- On track apart from performance issues
- Implemented DQN for Two Thinning -> need neural network due to large state space
- Added dynamic programming and other classical approaches
- Created OOP environment for comparing strategies
- Worked on modifications of Two Thinning




```
N = 4
GRAPH = Cycle(N)
M = 11

DEVICE = torch.device("cuda" if torch.cuda.is_available() else "cpu")
BATCH_SIZE = 64
EPS_START = 0.2
EPS_END = 0.05
EPS_DECAY = 2000
CONTINUOUS_REWARD = True
TRAIN_EPISODES = 3000
TARGET_UPDATE_FREQ = 10
MEMORY_CAPACITY = 10 * BATCH_SIZE
EVAL_RUNS_TRAIN = 5
EVAL_RUNS_EVAL = 10
EVAL_PARALLEL_BATCH_SIZE = 32
PATIENCE = 1000
MAX_LOAD_INCREASE_REWARD = -1
PRINT_BEHAVIOUR = False
PRINT_PROGRESS = True
OPTIMISE_FREQ = int(sqrt(M))
NN_MODEL = FullGraphicalTwoChoiceOneHotFCNet
NN_TYPE = "fc_one_hot_cycle"
```

Difficulties

Remaining steps

- Finish some versions of Two Thinning
- Optimise hyperparameters, e.g. gridsearch
- Run thorough evaluation, create fancy plots
- Interpret RL results
- Write dissertation

Thank you for your attention!