IERG5350 Assignment5:

CarRacing-v0 Tournament

Sharing my naive car-training experience

by WU Tong

Contents

- **Two-player self-training:**
  - My implementation
  - Some experimental results
- **Default parameters: ✓**
  - Stable and promising when choosing the **right** checkpoint.
- Some other dirty tuning experiences.

Two-player self-training

My implementation

- Initialization
  - Train from scratch;
  - Start from a half-trained / well-trained agent from a single-player environment.
- Two-player self-training:
  - Close the old environment.
  - Build a new two-player environment (*cCarRacingDouble-v0* via *make_competitive_car_racing*), the opponent policy is exactly the current policy, while it is NOT updated along training.
  - Update environment after several (50, 100, 200, ...) iterations.
- Some other extensions:
  - Alternative training of single-player and two-player environments.

Two-player self-training

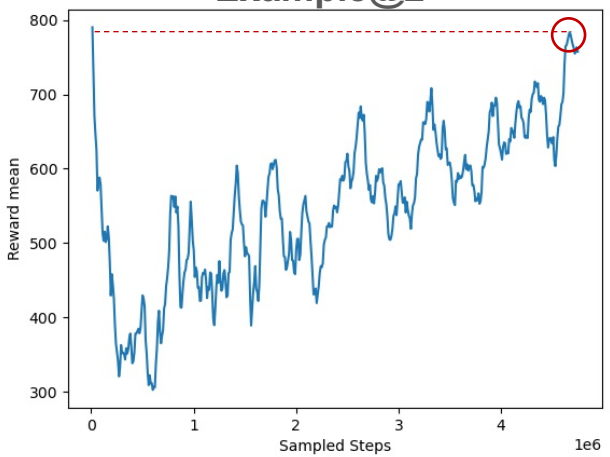
Some observations

- Train from scratch:
  - It’s hard to converge.
- Start from a half-trained / well-trained agent from a single-player environment:
  - A sharp drop of reward followed by an increasing trend.
- Alternative training of single-player and two-player environments:
  - Single-player stage converges more quickly than two-player stage;
  - No significant difference in the final results in my case.

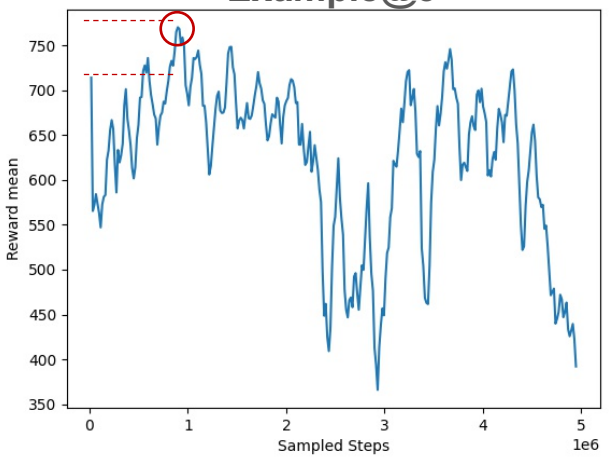
Example@1

Figure missing…
Imaging a curve with continuously low reward

Example@2



Example@3

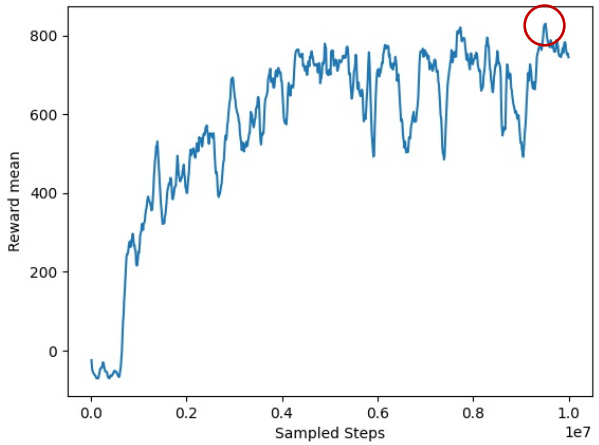


Oppo_best: 500

Using the default parameters can achieve sound results

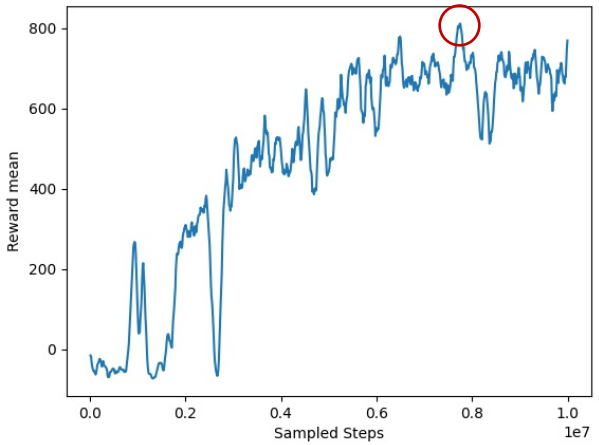
Choosing the right checkpoint is simple yet effective

Default@1



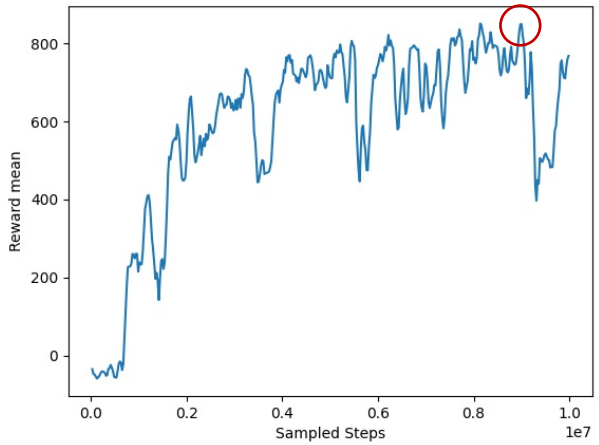
Best: 3950 Final: 4150

Default@2



Best: 3200 Final: 4150

Default@3



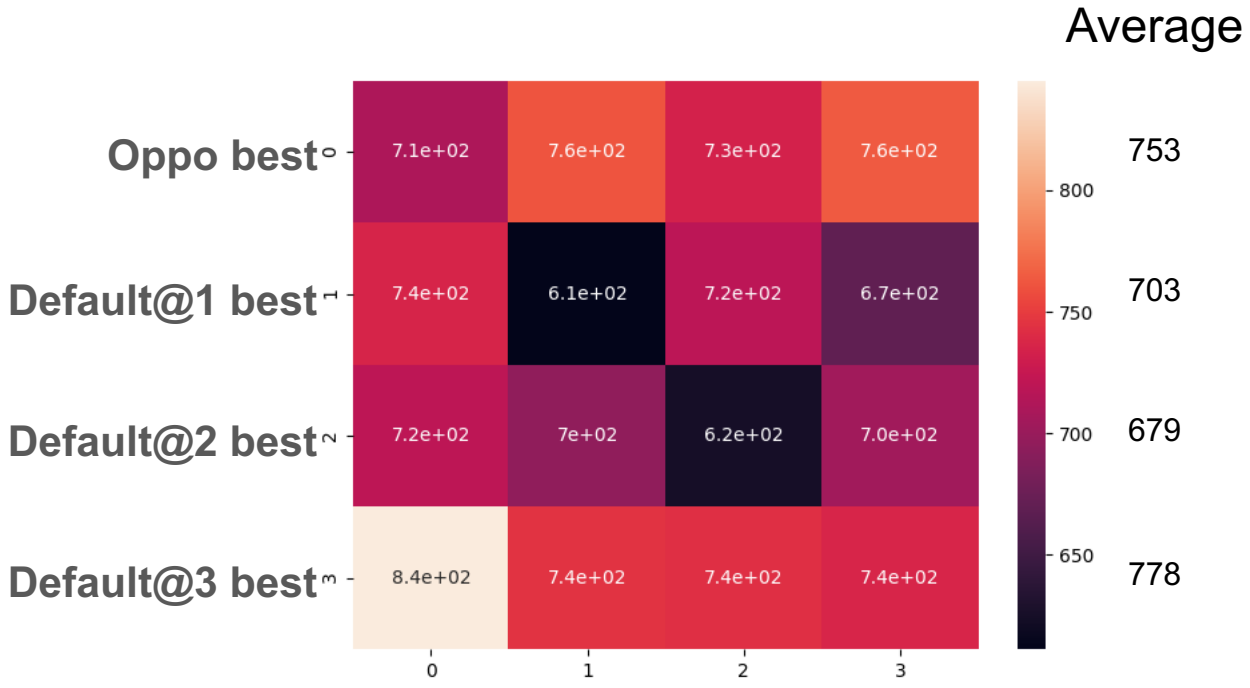
Best: 3350 Final: 4150

Iteration	Default@1	Default@2	Default@3	alphacar	zhenghao
Best	649	775	760	633	601
Final	513	647	607	586	612

One tournament

Another

Comparison and summary



Issues left unsolved

- How to make the training more stable? A promising training method that is less effected by randomization?
  - Adding entropy-weight do not make significant improvement. It would give the final checkpoint a higher performance but the optimal checkpoint is less competitive.
  - Other hyper-parameters, activation functions from github or original paper? Not work, sadly .

Thanks for listening!