AI Lesson

04.02.2021

Vincent Manier, Jonathan Laurent

Main Take-aways from last week:

- Start simple!
- Diversity of experiences should be obtained, not by weakening MCTS, but by augmenting exploration.
  - *There was a bug in the code impacting the temperature parameter*
- Optimization of the deduplication function
  - What is the relative time / computation cost compared to the overall algorithm? Is it worth optimizing? We should look at the overall impact on the program.

    Donald Knuth : "The real problem is that programmers have spent far too much time worrying about efficiency in the wrong places and at the wrong times; **premature optimization is the root of all evil (or at least most of it) in programming**."

  - Use of a profiler to analyze computation costs
  - Study big-O notation
  - Use of a hashing table
- Random permutation before selecting a batch and run an epoch
- Masking after the loss function gives the network the opportunity to learn

A. Objectives

- Use native Pytorch loss function vs. own calculation
- Remove masking in the forward pass and in the loss calculation
- Use a hashing table for the buffer - total time divided by 4-5
- Dirichlet distribution
- Ensure diversification of experiences is not done at the expense of weakening MCTS
- Improve the deduplication functions (averages)
- Stats about the network, buffer, performance
- Find a better way to update stats
- Organize competition against baseline
- Compare networks and replace the network used for self-play by the trained network if the latter is significantly better. (not fully tested)
- Better parameterization
- Better modularization
- Look at the use of a profiler
- Adjust the v value to reward faster wins v = win/loss * f_penalty (number of turns played - minimum)
- Review symmetries functions (2 are not operational)
- Run experiences

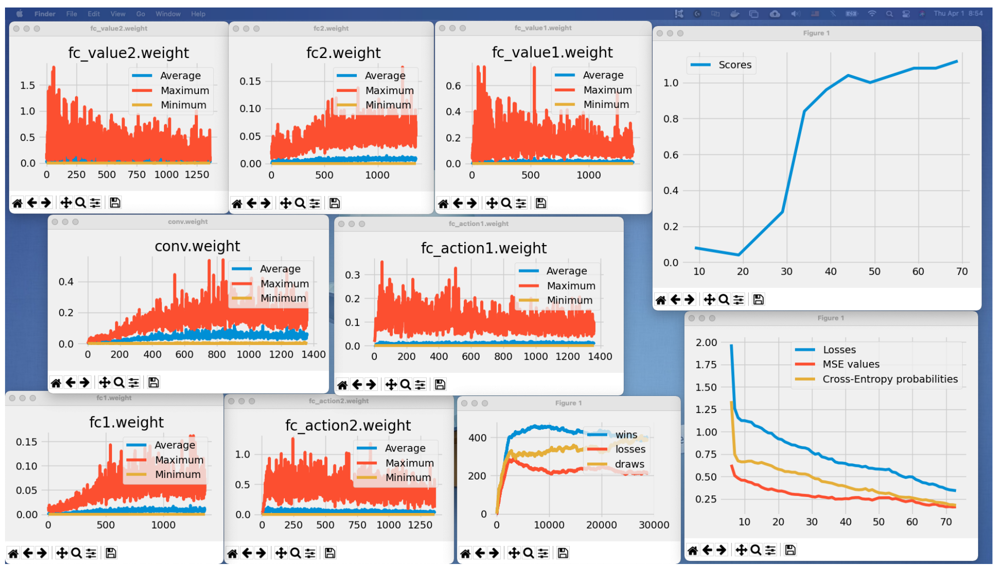
B. Experiences

a. Approach:

i. Start with some sets of parameters

ii. Run experiences by changing one indicator at a time

iii. Visually observe impacts on the neural network gradients, loss curve (MSE values and cross-entropy probabilities) and scores against a baseline (Network only against MCTS 1000 iterations)



b. Main parameters tested:

i. Buffer size target (500, 1000, 1500 unique positions)

ii. Puct for encouraging exploration (1, 2 and 4)

iii. Dirichlet Alpha (2, 1, 0.5, 0.25, Disabled)

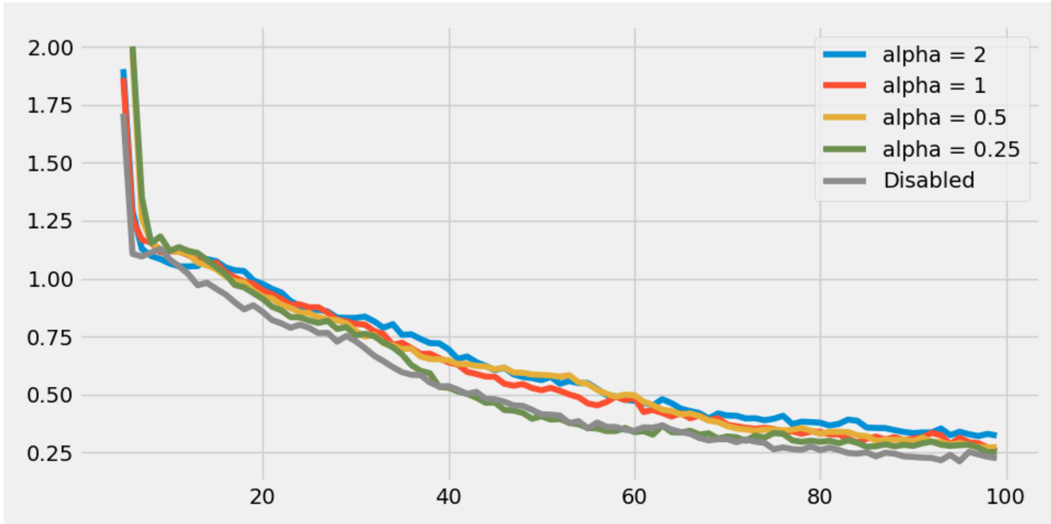
iv. MCTS iterations during search phase of self-play

c. Results

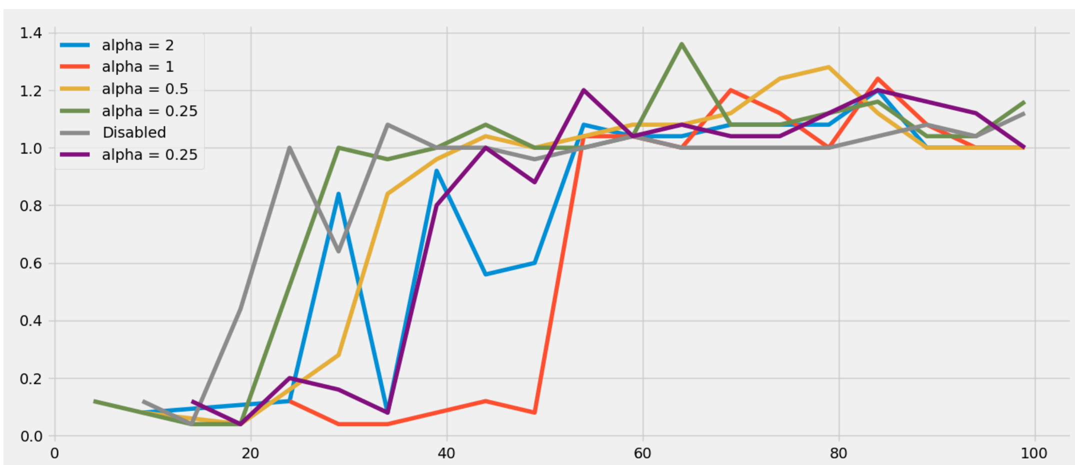
	Experiences	Value	Time (s)	Observations
Buffer Size Target	1	500	1200	High fluctuations in scores against baseline MCTS 1000
	3	1000	1283	Still a dip in scores but better shape
	2	1500	1319	fc_action2 gradients max btw 0 and 1.5 - too high?
Puct	3	1	1283	Base case
	5.1	2	1301	No real improvement
	4	4	1314	No real improvement
Dirichlet Alpha	5.1	2	1301	Base case
	6	1	1261	
	7	0.5	1234	Much better scores evolution
	8	0.25	1306	Still better
	9	DISABLED	1241	not much difference vs. exp8
	10	0.25	1191	Similar to experience 8 - as expected
explore_steps (MCTS search in self-play)	9	50	1241	Base case
	11	500	4106	not significantly better but nice score evolution

- See /docs/assets/charts for the charts of each experience

- Dirichlet Alpha comparison - loss curve



- Dirichlet Alpha comparison - score against baseline



- **Experience 11**