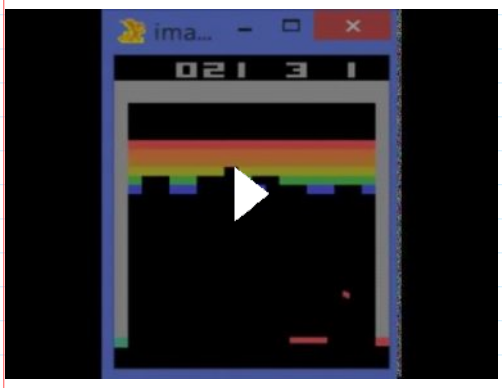
Videos

https://www.alexirpan.com/2018/02/14/rl-hard.html
Deep Q network learning to play Pong

Google DeepMind's Deep Q-learning playing Atari Breakout



input 210 × 160 → (downsample, crop) 84×84 (~7K pixels)

suppose... 3 fully connected hidden 10000 nodes / layer → here's the problem

100 M weights / layer

suppose instead

32 different 8 × 8 filters in one layer (each applied repeatedly across input)

1 M connections / layer better

but weights can be shared so 64 · 32 = 2048 weights / layer

AlphaGo 2016

Step 1: supervised learning for convolutional deep neural network

19 x 19 x 48 input

features of interest to human experts

black / white (2 features)

opp stones captured (8)

liberties

ladder captures

databases of expert play

55% match

3 weeks

+ smaller / faster network (less accurate)

policy network

output = move to make

Step 2: reinforcement learning for convolutional deep neural network

1 day

beats SL network 80% of games

Step 3: reinforcement learning for value network

output = probability / confidence in win

sample 1 pos from 30M games from step 2 RL network vs itself

Step 4: MCTS

observed rewards

tree policy uses $Q(s,a) + c \cdot P(s,a) \cdot \sqrt{\frac{\sum N(s,b)}{1 + N(s,a)}}$

from SL policy network

5 / sec per move

default policy uses fast SL network to select moves

initial reward = weighted avg of result of and output of Step 3 value network