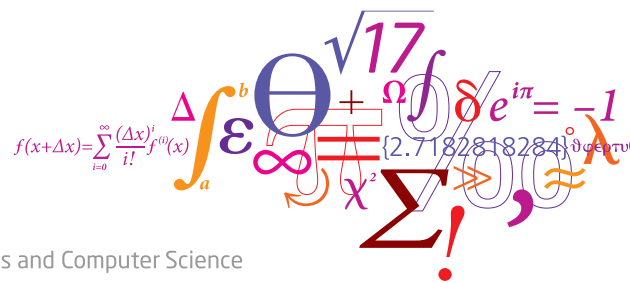
Deep Q learning

Ellen Vanhove, Emil Tyge, Carl Kjærgaard



DTU Compute
Department of Applied Mathematics and Computer Science

Outline

- Introduction
  - Our Problem
- Q-Learning
  - Reinforcement Learning
  - Different R-Learning Methods
  - Applying Deep Learning Methods
- Our Implementation
  - Breakout
  - Modelling
  - Difficulties
  - Plots
  - Comparison to other solutions

Our Problem

Why study these problems?

- Solutions to complex problems - State explosions
- Model-free AI planners
- Other real-value solutions

Reinforcement Learning

Problem Modelling

- Markovian Decision Problems
- On- vs Off-policy learning
  1. Policies in Reinforcement Learning
  2. Problems with on-policy learning

Reinforcement Learning

On Policy methods

- Policy Gradients

Off Policy Methods

- Q-Learning
- n-step Q-Learning

Mixed Policy Methods

- Actor-Critic methods
- A3C

Applying Deep Learning

Q Function Estimators

- Estimating complex linear functions
- Using deep networks as q-function estimator

Q-Learning

Q-Learning

- Use ε-greedy policy

$$\pi(a|s,\varepsilon) = \begin{cases} \text{argmax}_a \ Q(s,a,\theta) & \varepsilon > rand \\ a_{rand} & otherwise \end{cases}$$

- Store s_n, a_n, r_n in replay memory
- Optimize

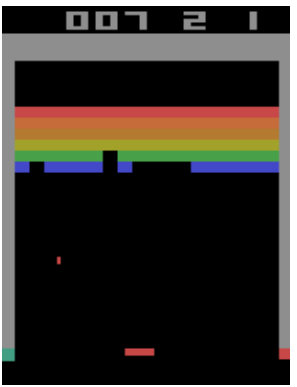
$$R_n = r_n + \gamma Q(s_{n+1}, \pi(s_{n+1}, 0), \theta^*)$$

$$\min_\theta \sum_n (R_n - Q(s_n, a_n, \theta))^2 + \alpha(R_n - Q(s_n, a_n, \theta))$$

Breakout

Breakout

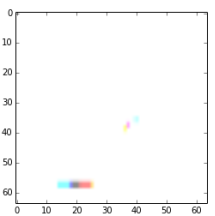
- Hidden markov model - fully observable with multiple frames
- Large observation space
- 4 Actions
- Reward when scoring



Modelling

Generalized input

- Generalized input
  64x64 images, 3 temporal channels
- Convolutional Neural net
  - Providing location invariance and pattern matching
  - Reduces parameters to be learned
  - Learning element relationships
- Dense neural net
  - Collecting filter information
- Quadratic cost - with L2 regularization



Difficulties

Problems

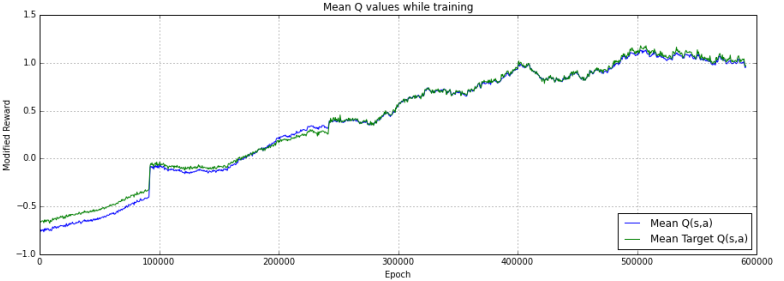
- Complicated policies to learn, very general problem description
- Robustness and convergence of solutions
- Connecting actions to rewards
- Hyper parameter - long training time
- Large observation space - smaller state space

Our solutions

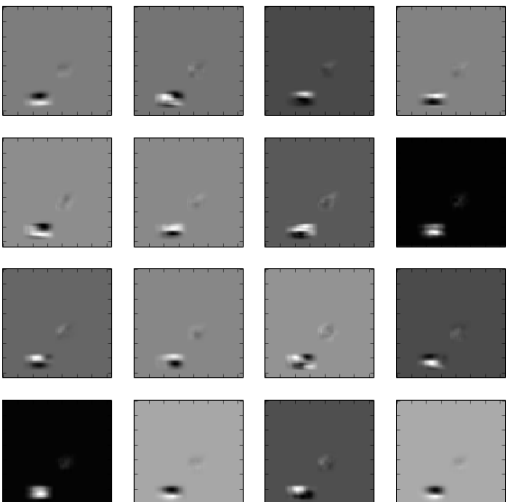
Our solutions

- Large memory of state/action/reward samples at least 250k
- Lookahead - calculating the discounted rewards
- Appropriate rewards - punish loosing the ball as well
- Cropping - simplifying the observational space
- L2 parameter costs - restrain parameters

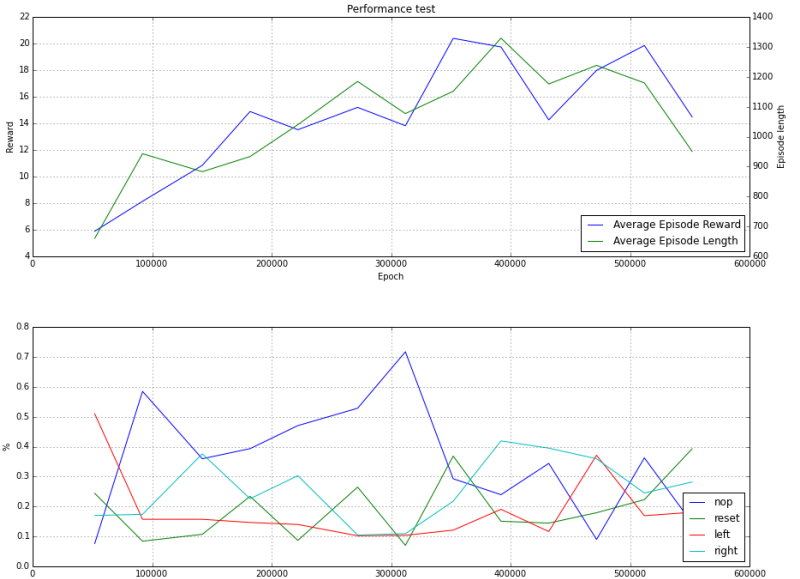
Evolution of Q values when training



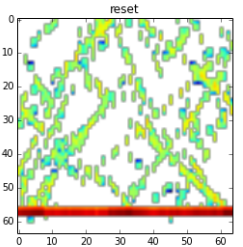
Filtered frame

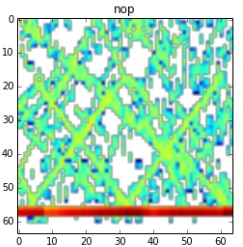


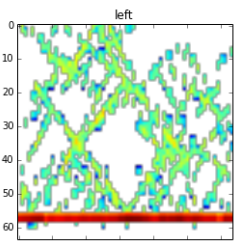
Performance testing

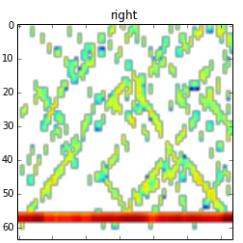


Analysing policies









Comparison to other solutions

Solutions

- Deepmind
- A3C