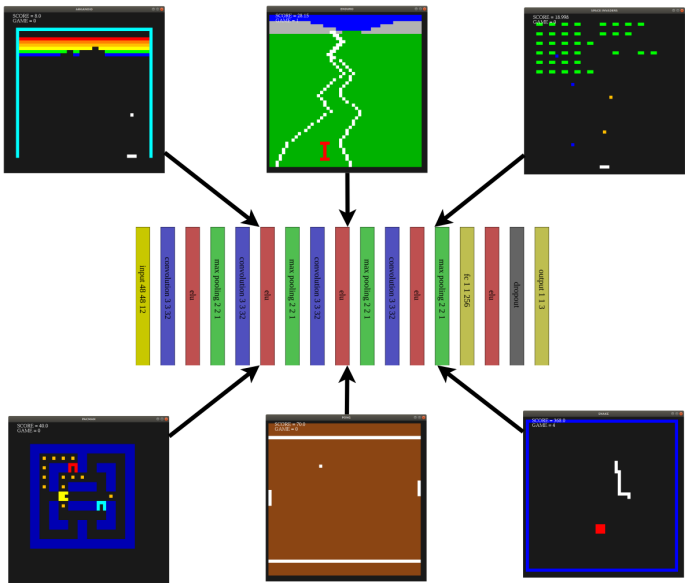
Playing multiple games using deep reinforcement learning

Michal Chovanec, Ondrej Šuch

Abstract In this paper we presents deep Q network (DQN) playing multiple games with the same weights - six ATARI games. We trained network by randomly changing games after several interations, using deep reinforcement learning. For comparison, we also trained common DQN. Instead of using 8x8 or 4x4 kernels as in (cite Atari DQN and Doom paper) we used only 3x3 kernels, and much lower of feature maps in layers. This network is much more parameter efficient. As results we present training score progress and network feature maps usability for each game.

1 Introduction



Obr. 1: Multi DQN can play six games with the same weights

2 Experiment setup

Network input: Last 4 frames, width = 48, height = 48.

Network architecture: Following state of the art (cite VGG Net, ResNet), we used only 3x3 kernels, instead of (cite Atari DQN, Doom, with 8x8 kernels). As activation function the ELU is choosen, with $\alpha = 0.1$. After four convolution and pooling layers, the full connected layer with 256 neurons is used. On the last layer the linear activation is used. We also tried network without 256 FC layer - to reduce amount of parameters, but the training was much slower.

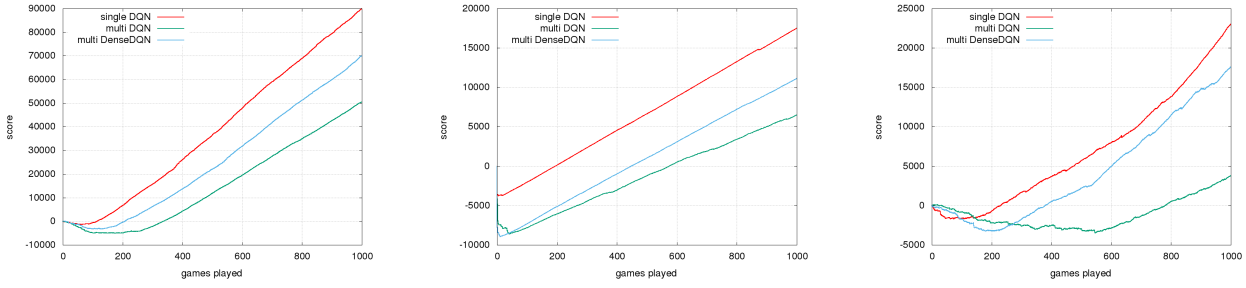
Proposed network architecture is :

$IN48\times48\times12-C3\times3\times32-P2\times2-C3\times3\times32-P2\times2-C3\times3\times32-P2\times2-FC256-FC_{actions}$

Hyperparmaters of network were choosen as :

- learning rate $\eta = 0.001$
- L1 and L2 regularization $\lambda_1 = \lambda_2 = 0.0000001$
- dropout 0.02 (only after FC256 layer)
- minibatch size 32 for single DQN, 128 for multi DQN
- gradient clip $\langle -10, 10 \rangle$

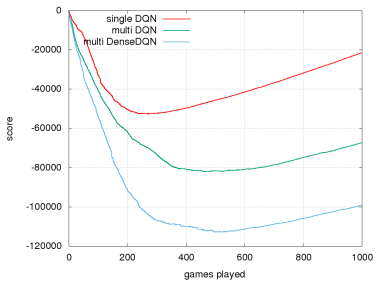
3 Results



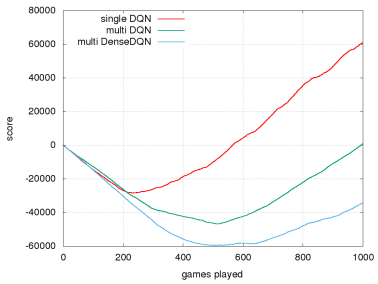
(a) Arkanoid

(b) Enduro

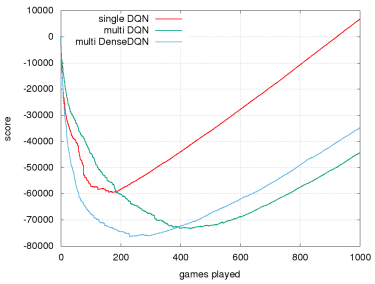
(c) Invaders



(d) Pacman

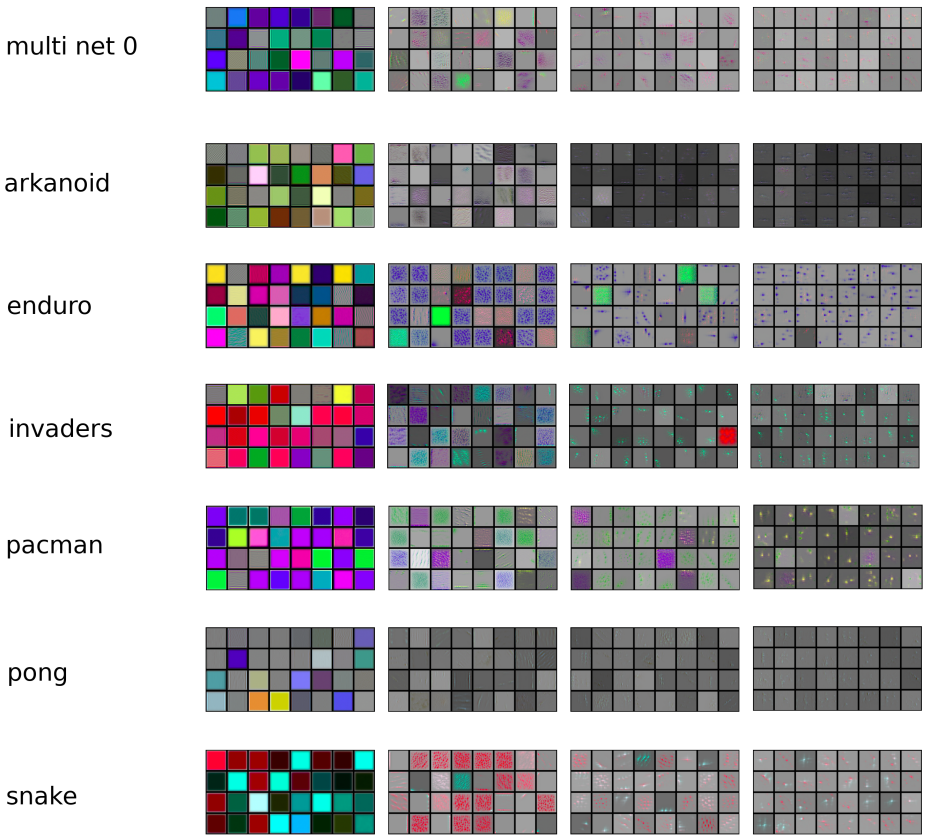


(e) Pong



(f) Snake

Obr. 2: Training score result for first 1000 games



Obr. 3: Kernel maximum response inputs visualisation for different networks