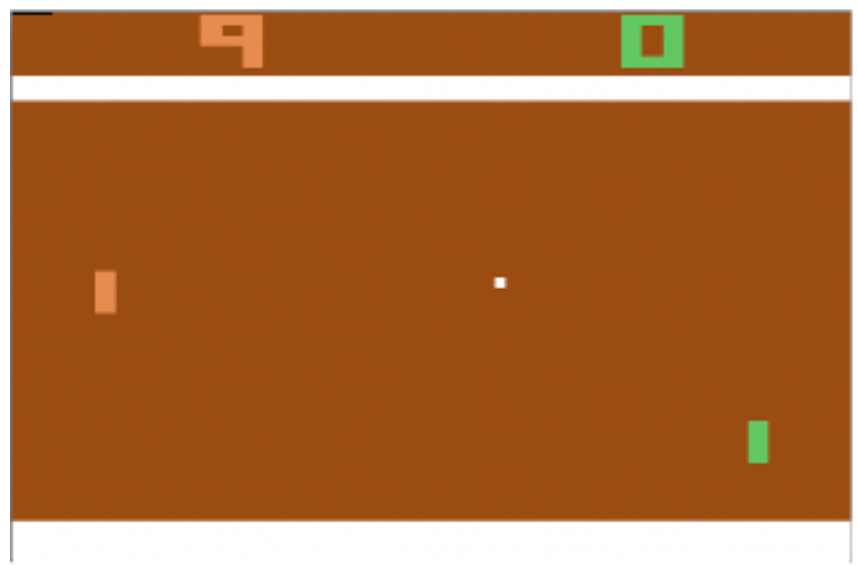
Playing atari with deep reinforcement learning

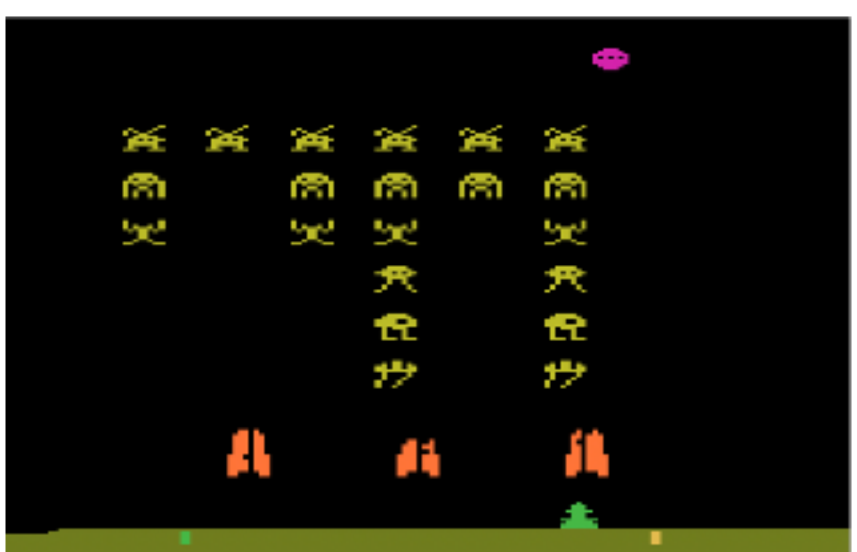
Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602. Chicago

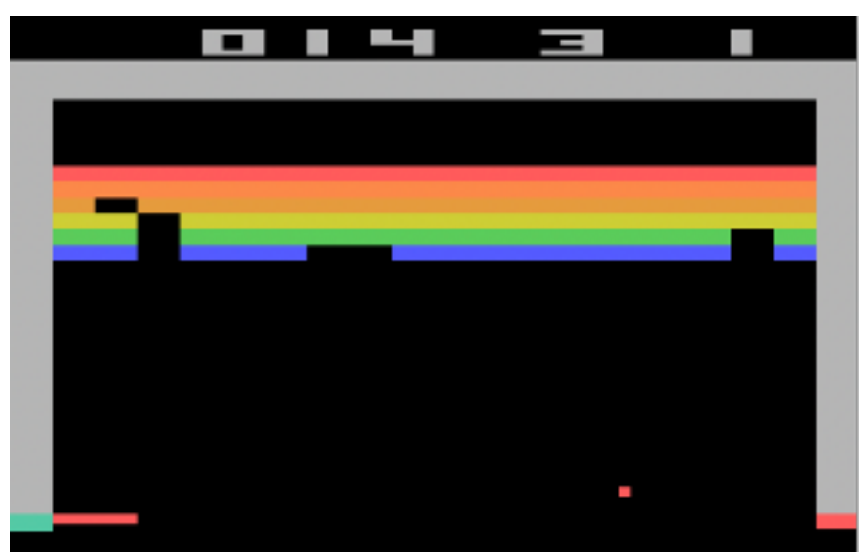
Introduction

- Create a single neural network agent that is able to successfully learn to play as many of the games as possible
- The network has outperformed all previous RL algorithms on six of the seven games we have attempted and surpassed an expert human player on three of them

Introduction







Background

Optimal action-value function $Q^*(s,a) = \mathbb{E}_{s'\sim\mathcal{E}}\left[r + \gamma \max_{a'} Q^*(s',a') \Big| s,a\right]$

Loss function

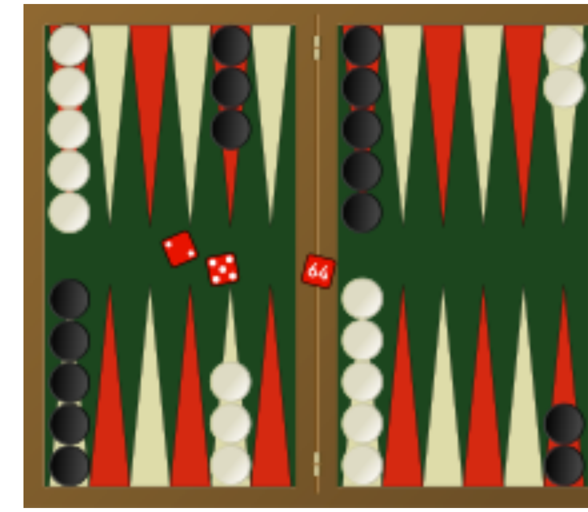
$$L_i\left(\theta_i\right) = \mathbb{E}_{s,a\sim\rho(\cdot)}\left[\left(y_i - Q\left(s,a;\theta_i\right)\right)^2\right],$$

$$y_i \ = \ \mathbb{E}_{s'\sim\mathcal{E}}\left[r + \gamma \max\nolimits_{a'} Q(s',a';\theta_{i-1}) | s,a\right]$$

Loss function (gradient)

$$\nabla_{\theta_i} L_i\left(\theta_i\right) = \mathbb{E}_{s,a\sim\rho(\cdot);s'\sim\mathcal{E}}\left[\left(r + \gamma \max_{a'} Q(s',a';\theta_{i-1}) - Q(s,a;\theta_i)\right)\nabla_{\theta_i} Q(s,a;\theta_i)\right]$$

Deep Reinforcement Learning



- TD-Gammon
  - A computer backgammon (西洋雙陸棋戲) program
  - On-policy
- Experience replay
- Store the agent’s experiences at each time-step, et = (st, at, rt, st+1) in a data-set D = e1 , ..., eN
- Off-policy (Q-learning)

Deep Reinforcement Learning

Algorithm 1 Deep Q-learning with Experience Replay

Initialize replay memory $\mathcal{D}$ to capacity N
Initialize action-value function Q with random weights
for episode $= 1, M$ **do**
  Initialise sequence $s_1 = \{x_1\}$ and preprocessed sequenced $\phi_1 = \phi(s_1)$
  for $t = 1, T$ **do**
    With probability ϵ select a random action a_t
    otherwise select $a_t = \max_a Q^*(\phi(s_t), a; \theta)$
    Execute action a_t in emulator and observe reward r_t and image x_{t+1}
    Set $s_{t+1} = s_t, a_t, x_{t+1}$ and preprocess $\phi_{t+1} = \phi(s_{t+1})$
    Store transition $(\phi_t, a_t, r_t, \phi_{t+1})$ in $\mathcal{D}$
    Sample random minibatch of transitions $(\phi_j, a_j, r_j, \phi_{j+1})$ from $\mathcal{D}$
    Set $y_j = \begin{cases} r_j & \text{for terminal } \phi_{j+1} \\ r_j + \gamma \max_{a'} Q(\phi_{j+1}, a'; \theta) & \text{for non-terminal } \phi_{j+1} \end{cases}$
    Perform a gradient descent step on $(y_j - Q(\phi_j, a_j; \theta))^2$ according to equation 3
  end for
end for

Preprocessing

- Raw: 210 × 160 pixel images with a 128 color palette
- Convert into gray scale, down-sampling to 110x84
- Crop an 84x84 region (Conv2D they selected expects square inputs)
- Stack the last 4 frames

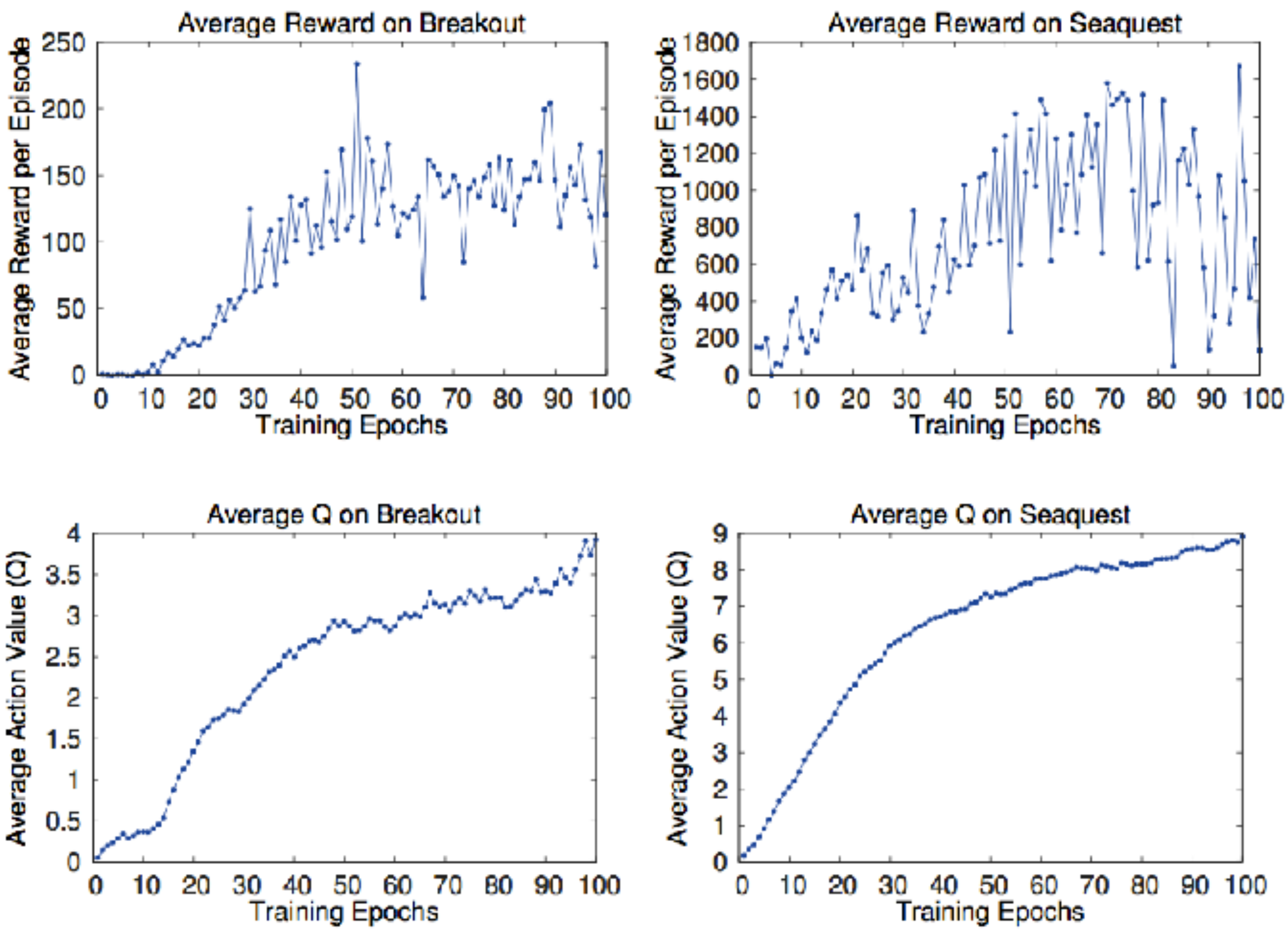
Model Architecture

- The input to the neural network is an $84 \times 84 \times 4$ image
- First layer: convolves 16 8×8 filters, stride 4, rectifier nonlinearity
- Second layer: convolves 32 4×4 filters, stride 2, rectifier nonlinearity
- Final layer: fully-connected and consists of 256 rectifier units
- The output layer is a fully-connected linear layer with a single output for each valid action.
- The number of valid actions varied between 4 and 18 on the games we considered

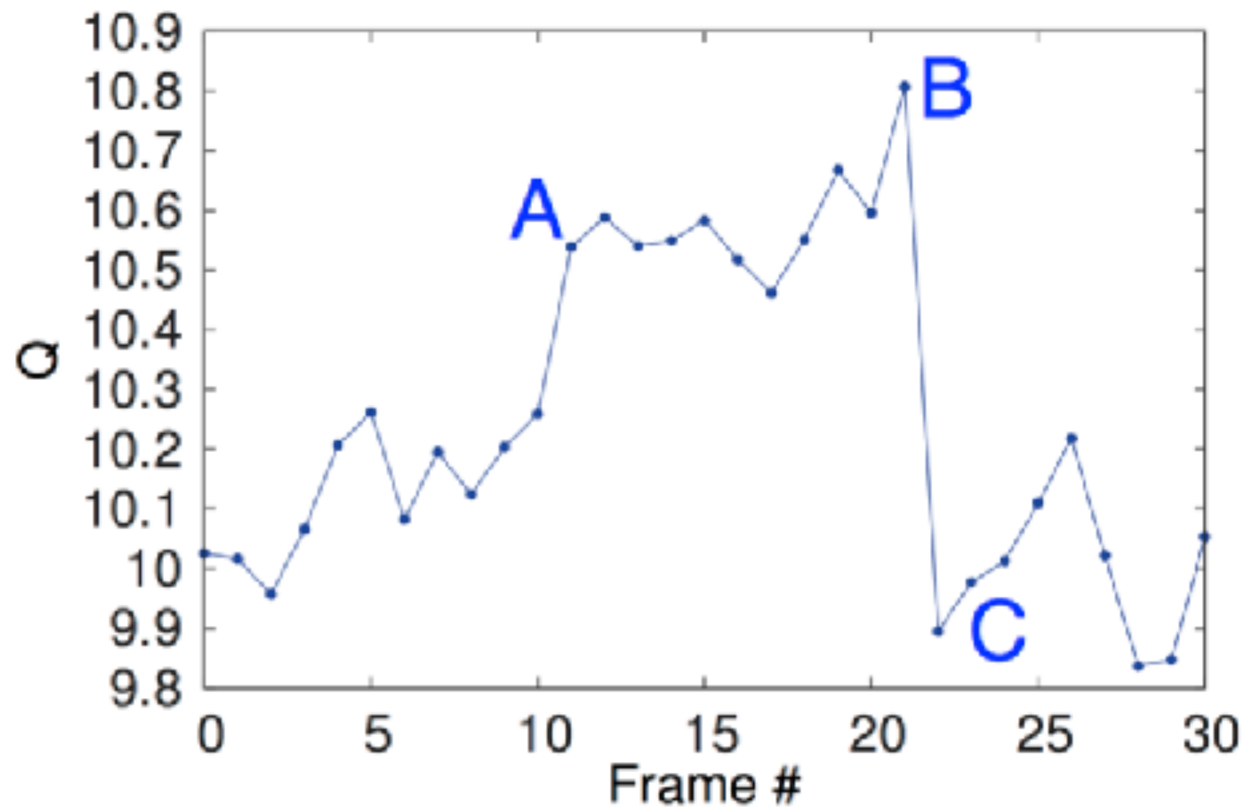
Experiments

- Set positive rewards to 1, negative rewards to -1, and unchanged rewards to 0
- Minibatch size = 32
- ε-greedy with ε annealed linearly from 1 to 0.1 over the first million frames, and fixed at 0.1 thereafter
- Frame skipping technique: the agent sees and selects actions on every kth frame instead of every frame, and its last action is repeated on skipped frames (k=3 or 4)

Experiments



Experiments



Experiments

	B. Rider	**Breakout**	**Enduro**	**Pong**	**Q*bert**	**Seaquest**	**S. Invaders**
Random	354	1.2	0	−20.4	157	110	179
Sarsa [3]	996	5.2	129	−19	614	665	271
Contingency [4]	1743	6	159	−17	960	723	268
DQN	**4092**	**168**	**470**	**20**	**1952**	**1705**	**581**
Human	7456	31	368	−3	18900	28010	3690
HNeat Best [8]	3616	52	106	19	1800	920	**1720**
HNeat Pixel [8]	1332	4	91	−16	1325	800	1145
DQN Best	**5184**	**225**	**661**	**21**	**4500**	**1740**	1075