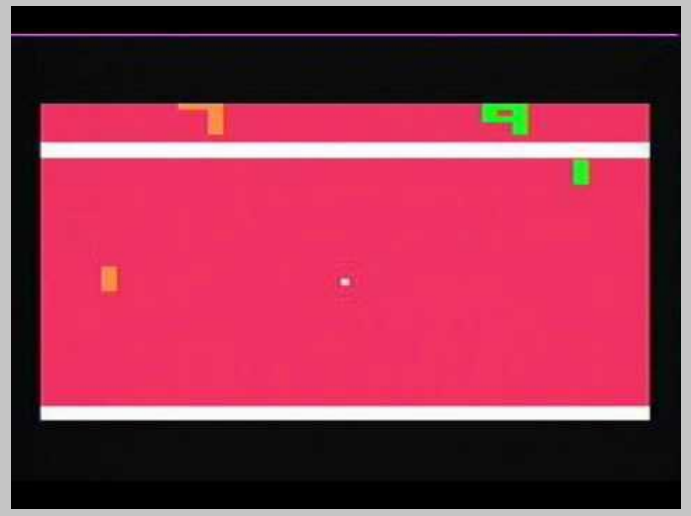
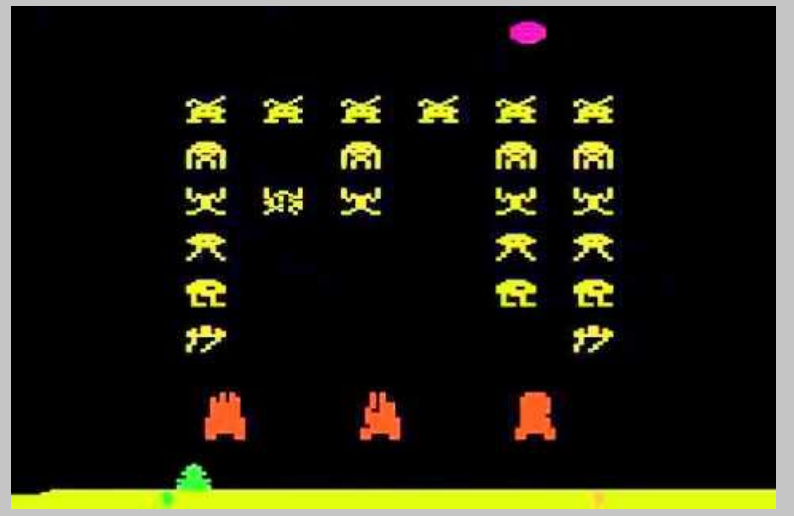
Remember these?





Playing Atari Games using RL

VARSHA LALWANI
AKSHAY MASARE

Motivation

May be we can design game players for each one of them!

But, how about an AI agent who can **learn** to play them **all**!

This is where the concept of a *general game player* come into the picture.

In this project we are trying to implement a deep reinforced learning based agent to play multiple video games.

Problem Statement

Learning to play *Breakout* using a convolutional neural network model trained with a variant of Q-learning, whose input would be raw pixels and whose output would be a value function estimating future rewards.

Concepts Involved

Reinforcement Learning

Q-Learning

Convolutional Neural Network

Reinforcement Learning and Q-Learning

In a reinforcement learning model, an agent takes actions in an environment with the goal of maximising a cumulative reward.

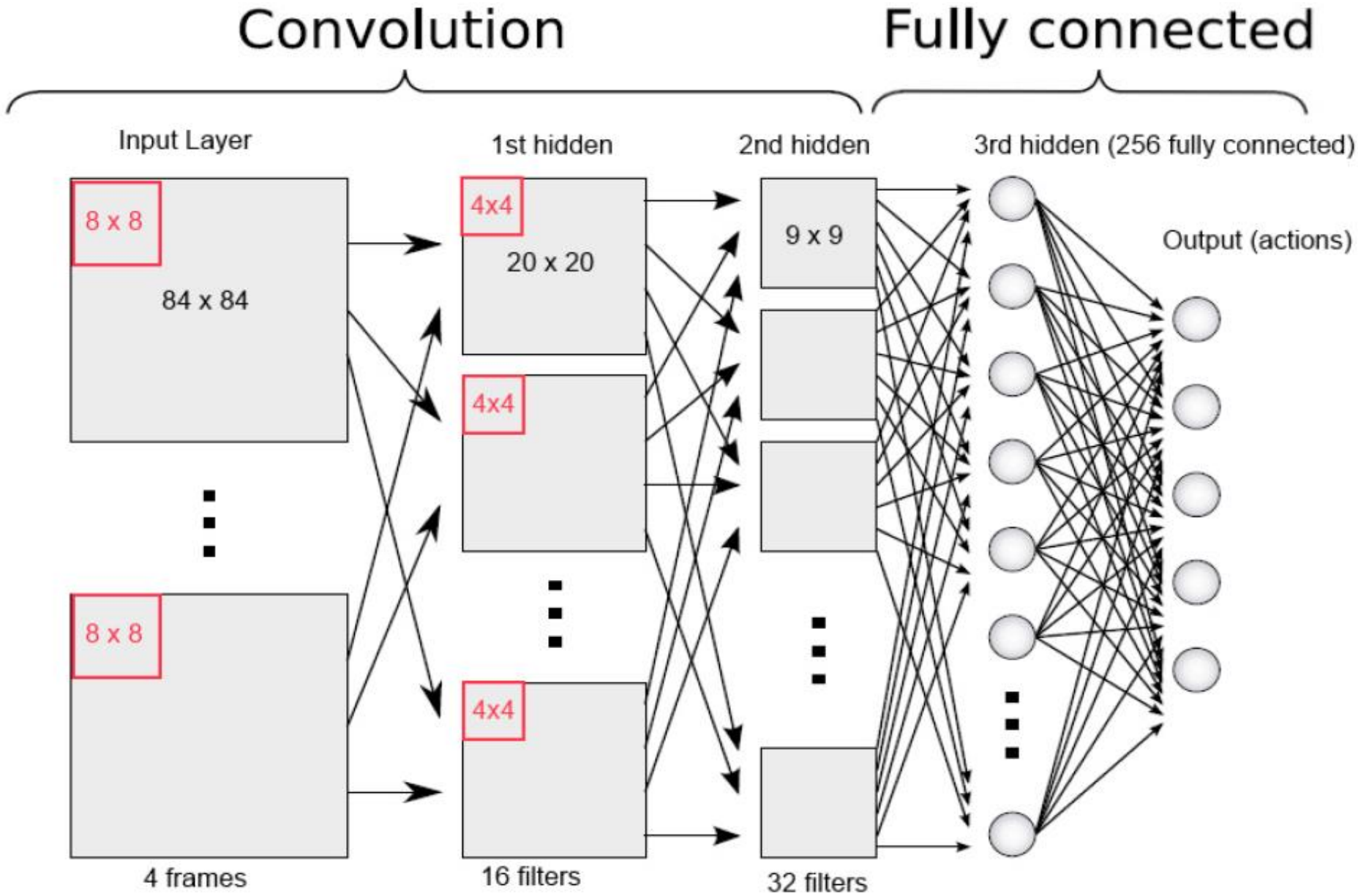
Q-learning is a model free form of RL

Algorithm:

$$
\begin{aligned}
&Initialize\ \boldsymbol{Q}(\boldsymbol{s},\boldsymbol{a})\ arbitrarily\\
&Repeat\ (for\ each\ episode):\\
&Initialize\ \boldsymbol{S}\\
&Repeat\ (for\ each\ step\ of\ episode):\\
&Choose\ \boldsymbol{a}\ from\ \boldsymbol{s}\ using\ policy\ derived\ from\ \boldsymbol{Q}\\
&(e.g.\ \in -greedy)\\
&Take\ action\ \boldsymbol{a}, observe\ \boldsymbol{r}, \boldsymbol{s'}\\
&\boldsymbol{Q(s,a)} <-- \boldsymbol{Q(s,a)} + \boldsymbol{\alpha}[\boldsymbol{r} + \boldsymbol{\gamma}.\boldsymbol{max}\,\boldsymbol{Q}(\boldsymbol{s'},\boldsymbol{a'}) - \boldsymbol{Q(s,a)}]\\
&\boldsymbol{s} <-- \boldsymbol{s'}\\
&until\ \boldsymbol{s}\ is\ terminal
\end{aligned}
$$

Convolutional Neural Networks

- Suited for extracting features from images
- We take 4 images at a time, downscaled to 84x84 pixels
- Images taken as 2D matrices
- 2D matrices convolved with linear filters
- Weight matrices for multiple image


Convolution
Fully connected
Input Layer
1st hidden
2nd hidden
3rd hidden (256 fully connected)
8 x 8
84 x 84
8 x 8
4x4
20 x 20
4x4
4x4
9 x 9
Output (actions)
4 frames
16 filters
32 filters
Nodes: 84 x 84 x 4 20 x 20 x 16 9 x 9 x 32 256 4
Weights: 8 x 8 x 4 x 16 4 x 4 x 16 x 32 9 x 9 x 32 x 256 256 x 4

Arcade Learning Environment

- It is built on top of Stella, open-source Atari 2600 emulator
- Built in C++, Support for over 50 games
- Can programmatically input player commands
- Outputs Image of the game screen, score and the state of the game

References

[1] The Arcade Learning Environment: An Evaluation Platform for General Agents by Marc G. Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling Journal of Artificial Intelligence Research 47, pp. 253-279, 2013.

[2] Stella Emulator: http://stella.sourceforge.net/

[3] Playing Atari with Deep Reinforcement Learning by Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, Martin Riedmiller NIPS Deep Learning Workshop, 2013.

Any Questions ??