Performance vs. Feature Compression in Atari Games

Holt Spalding and Oliver Newland

Background

1. DQN and Atari Games
    - “Playing Atari with Deep Reinforcement Learning” by Volodymyr Mnih *et. al.*
2. Frame Skipping
    - “Dynamic Frame skip Deep Q Network” By Aravind S. Lakshminarayanan *et. al.*
3. Resolution Reduction
4. Cropping

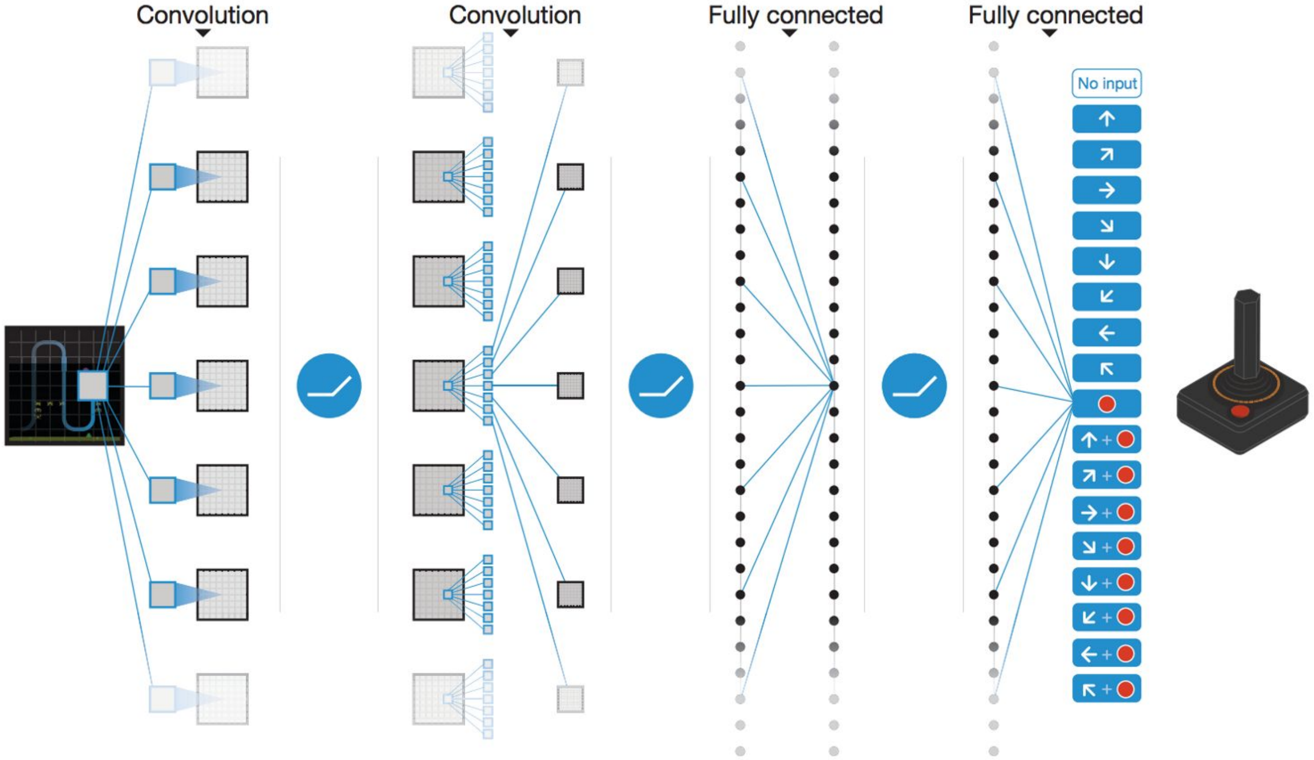
Motivation

- DQN agents are powerful but computationally expensive
- Can we discover an optimal amount of information?
- What does this optimum tell us about the problem itself?
- Discover patterns?

DQN

- Convoluted Neural Network with Q-Learning
- Built using inspiration from:
    - https://github.com/keon/deep-q-learning/blob/master/dqn.py
    - https://github.com/rockeyben/MsPacman

DQN



Convolution
Convolution
Fully connected
Fully connected
No input

DQN

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

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

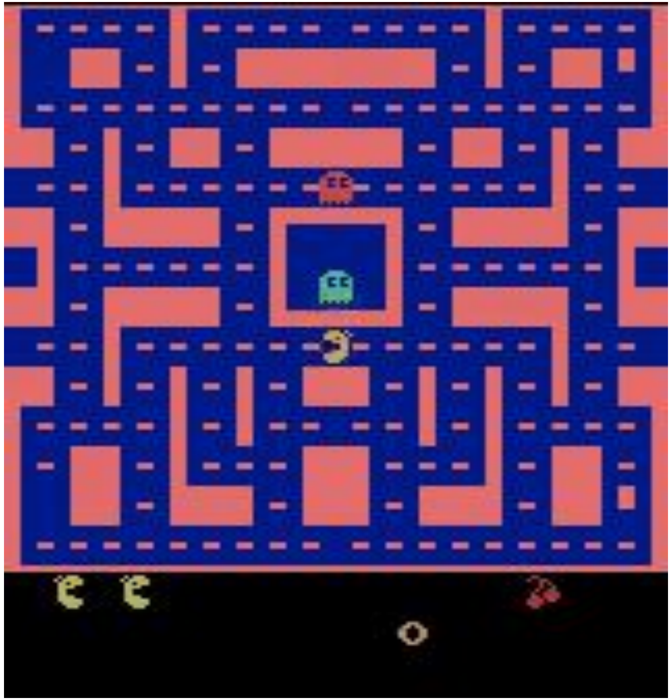
$$\nabla_{\theta_i}L_i\left(\theta_i\right)=\mathbb{E}_{s,a\sim\rho(\cdot);s'\sim\mathcal{E}}\left[\left(r+\gamma\max_{a'}Q(s',a';\theta_{i-1})-Q(s,a;\theta_i)\right)\nabla_{\theta_i}Q(s,a;\theta_i)\right].$$

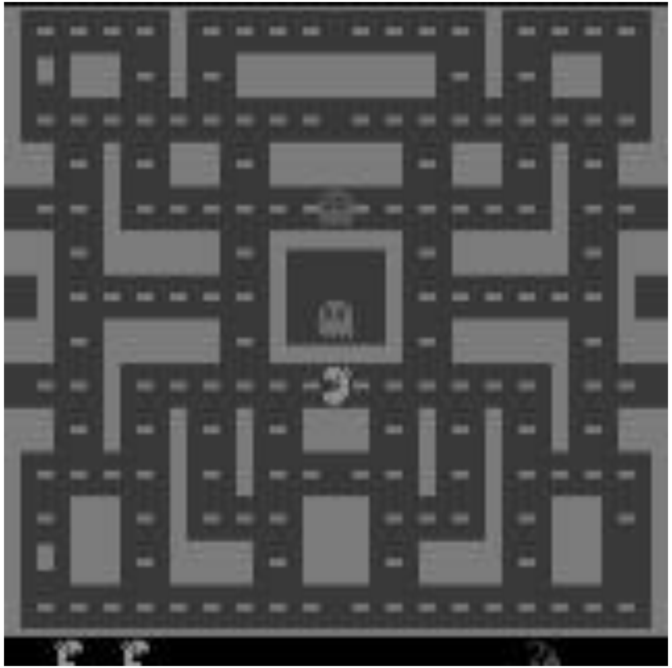
DQN

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

OpenAI Gym





Experimental Details

- Experiment over three separate feature compressions:
    a. Frame skipping
    b. Resolution Reduction
    c. Cropping
- Compare performance vs. computational demand (clock time and time steps)
- Replicate over different Atari games

Progress

1. Created working DQN with experience replay in tensorflow
2. Learned to manipulate pixels and frames with OpenCV, still working on cropping (different for different games)

Next Steps

1. Complete preprocessing for all experimental conditions
2. Glue preprocessing to DQN
3. Run experiments and produce learning curves for all experimental conditions
4. Determine optimums and analyze

Expected Outcomes

- Feature reduction will perform better than baseline inputs
- Exact optimum will depend on game
- Predictions:
  - Frame skipping and feature resolution will be similar for all games
  - Cropping will vary wildly
- Variation in GPU?

References

1. 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
2. Van Hasselt, H., Guez, A., & Silver, D. (2016, February). Deep Reinforcement Learning with Double Q-Learning. In *AAAI* (Vol. 2, p. 5).
3. Wang, Z., Schaul, T., Hessel, M., Van Hasselt, H., Lanctot, M., & De Freitas, N. (2015). Dueling network architectures for deep reinforcement learning. *arXiv preprint arXiv:1511.06581*.
4. Lakshminarayanan, A. S., Sharma, S., & Ravindran, B. (2016). Dynamic frame skip deep q network. *arXiv preprint arXiv:1605.05365*.

Thanks!

Questions?