Problem Solving with AI Techniques
Deep RL

Paul Weng

UM-SJTU Joint Institute

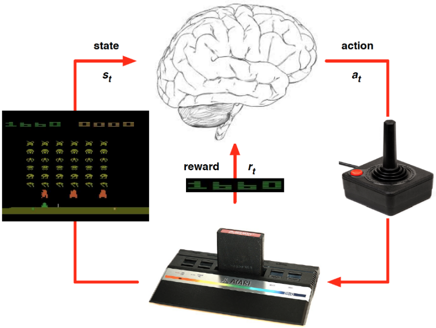
VE593, Fall 2018

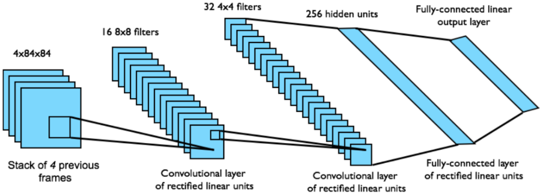
JOINT INSTITUTE
交大密西根学院

What is Deep Reinforcement Learning (DRL)?

- Idea: Combine deep learning and RL
- Use deep ANN as function approximator for value functions, Q-functions or policy
- End-to-end approach: Learn controller from raw inputs (e.g., direct from sensors) directly
- DRL can tackle large complex problems
- Issues:
  - states are not observable in practice (env. not MDP, but POMDP)
  - DRL sample inefficient

Example: Atari Games





from Gašić

Detour: Batch RL and Fitted Q-Iteration

- Goal: learn a good policy with training data $\mathcal{D} = \{(s^i, a^i, r^i, s'^i \mid i = 1, \ldots, N\}$ and no possible other interaction
- Idea: Approximate Q-iteration
  Q-iteration:

$$Q^*_0(s, a) = 0$$
$$Q^*_t(s, a) = R(s, a) + \gamma \sum_{s'} T(s, a, s') \max_{a'} Q^*_{t-1}(s', a')$$

- Fitted Q-iteration: each update is a regression problem given $\mathcal{D}$:
  At iteration t, for $\boldsymbol{x} = \langle s, a \rangle$, learn to predict $\boldsymbol{y} = Q(s, a)$ from $\{(\langle s^i, a^i \rangle, r^i + \gamma \max_a Q_{t-1}(s'^i, a)) \mid i = 1, \ldots, N\}$
- Possibly, use linear or non-linear model for regression

DQN: Principle

- DQN $\sim$ online fitted Q-iteration
- DQN uses CNN to approximate Q^*

- Memory replay
  - Issue: training point generated online not i.i.d
  - Solution: store training points in memory and sample mini-batch from it for training
- Target Q-function
  - Issue: ever changing target
  - Solution: freeze ANN at regular interval and use it as target

DQN: Algorithm

initialize $\hat{Q}_{\boldsymbol{w}}$ with random small weights $\boldsymbol{w}$
for $t = 1, \ldots$ **do**
    choose action a in state s with ε-greedy from $\hat{Q}_{\boldsymbol{w}}$
    observe r, s' after applying a in s
    add (s, a, r, s') to replay memory
    sample minibatch (s^i, a^i, r^i, s') from replay memory
    do mini-batch gradient descent on $\hat{Q}_{\boldsymbol{w}}$

- ε annealed from 1 to 0.1
- $\hat{Q}_{\boldsymbol{w}}$ is a CNN
- loss function is defined by $\left(r + \gamma \max_{a'} \hat{Q}_{\boldsymbol{w}^-}(s', a') - \hat{Q}_{\boldsymbol{w}}(s, a)\right)^2$
- $\hat{Q}_{\boldsymbol{w}^-}$ is target function, provided by frozen CNN
- Stochastic gradient update:
  $\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha \nabla_{\boldsymbol{w}} \hat{Q}_{\boldsymbol{w}}(s, a)\left(r + \gamma \max_{a'} \hat{Q}_{\boldsymbol{w}^-}(s', a') - \hat{Q}_{\boldsymbol{w}}(s, a)\right)$

Actor-Critic Methods

- Issue: Policy gradient with Monte Carlo sampling has large variance and is sample-inefficient
- Idea: learn V^{π_θ}, called **critic** while learning policy π_θ, called **actor**
- V^{π_θ} is appoximated by $V_{\boldsymbol{w}}$
- In deep RL, both actor and critic are represented by ANNs
- Example of updates for a sample (s, a, r, s'):

$$\boldsymbol{w} \leftarrow \boldsymbol{w} + \alpha_1 \nabla_{\boldsymbol{w}} V_{\boldsymbol{w}}(s) \big(r + \gamma V_{\boldsymbol{w}}(s') - V_{\boldsymbol{w}}(s) \big)$$
$$\theta \leftarrow \theta + \alpha_2 \nabla_\theta \log \pi_\theta(s, a)(r + \gamma V_{\boldsymbol{w}}(s') - V_{\boldsymbol{w}}(s))$$

- Why do we subtract $V^{\boldsymbol{w}}(s)$ (called baseline)?

Variance Reduction

- We want to estimate $\mathbb{E}[X]$ with $\bar{X} = \frac{1}{N}\sum_i X_i$ where X_i's i.i.d.

Variance Reduction

- We want to estimate $\mathbb{E}[X]$ with $\bar{X} = \frac{1}{N}\sum_i X_i$ where X_i's i.i.d.
- The variance of this estimator is $\mathbb{V}[\bar{X}] = \frac{1}{N}\mathbb{V}[X]$

Variance Reduction

- We want to estimate $\mathbb{E}[X]$ with $\bar{X} = \frac{1}{N}\sum_i X_i$ where X_i's i.i.d.
- The variance of this estimator is $\mathbb{V}[\bar{X}] = \frac{1}{N}\mathbb{V}[X]$
- Can we do better if N is fixed?

Variance Reduction

- We want to estimate $\mathbb{E}[X]$ with $\bar{X} = \frac{1}{N}\sum_i X_i$ where X_i's i.i.d.
- The variance of this estimator is $\mathbb{V}[\bar{X}] = \frac{1}{N}\mathbb{V}[X]$
- Can we do better if N is fixed?

- Choose Y such that it can be sampled, $\mathbb{E}[Y] = 0$, and X and Y correlated

Variance Reduction

- We want to estimate $\mathbb{E}[X]$ with $\bar{X} = \frac{1}{N}\sum_i X_i$ where X_i's i.i.d.
- The variance of this estimator is $\mathbb{V}[\bar{X}] = \frac{1}{N}\mathbb{V}[X]$
- Can we do better if N is fixed?

- Choose Y such that it can be sampled, $\mathbb{E}[Y] = 0$, and X and Y correlated
- Consider new variable $Z = X - \eta Y$

Variance Reduction

- We want to estimate $\mathbb{E}[X]$ with $\bar{X} = \frac{1}{N}\sum_i X_i$ where X_i's i.i.d.
- The variance of this estimator is $\mathbb{V}[\bar{X}] = \frac{1}{N}\mathbb{V}[X]$
- Can we do better if N is fixed?

- Choose Y such that it can be sampled, $\mathbb{E}[Y] = 0$, and X and Y correlated
- Consider new variable $Z = X - \eta Y$
- $\mathbb{V}[Z] = \mathbb{V}[X] - 2\eta\text{cov}(X, Y) + \eta^2\mathbb{V}[Y]$

Variance Reduction

- We want to estimate $\mathbb{E}[X]$ with $\bar{X} = \frac{1}{N}\sum_i X_i$ where X_i's i.i.d.
- The variance of this estimator is $\mathbb{V}[\bar{X}] = \frac{1}{N}\mathbb{V}[X]$
- Can we do better if N is fixed?

- Choose Y such that it can be sampled, $\mathbb{E}[Y] = 0$, and X and Y correlated
- Consider new variable $Z = X - \eta Y$
- $\mathbb{V}[Z] = \mathbb{V}[X] - 2\eta \mathrm{cov}(X, Y) + \eta^2 \mathbb{V}[Y]$
- Using $\eta^* = \frac{\mathrm{cov}(X,Y)}{\mathbb{V}[Y]}$, $\mathbb{V}[Z] = (1 - \rho(X, Y)^2)\mathbb{V}[X]$

Variance Reduction

- We want to estimate $\mathbb{E}[X]$ with $\bar{X} = \frac{1}{N}\sum_i X_i$ where X_i's i.i.d.
- The variance of this estimator is $\mathbb{V}[\bar{X}] = \frac{1}{N}\mathbb{V}[X]$
- Can we do better if N is fixed?

- Choose Y such that it can be sampled, $\mathbb{E}[Y] = 0$, and X and Y correlated
- Consider new variable $Z = X - \eta Y$
- $\mathbb{V}[Z] = \mathbb{V}[X] - 2\eta \text{cov}(X, Y) + \eta^2 \mathbb{V}[Y]$
- Using $\eta^* = \frac{\text{cov}(X,Y)}{\mathbb{V}[Y]}$, $\mathbb{V}[Z] = (1 - \rho(X, Y)^2)\mathbb{V}[X]$

- Estimate $\mathbb{E}[X] = \mathbb{E}[Z]$ with $\bar{Z} = \frac{1}{N}\sum_i X_i - \eta Y_i$

Conclusion

- Many algorithms
  - Extension of DQN: Double DQN, Dueling DQN, prioritized replay...
  - Actor-critic algorithms: A3C, ACER, PPO, Rainbow...
  - On-going research work
- Current issues
  - Sample efficiency
  - Computational efficiency
  - Still difficult to apply
- Related research problems
  - Learning by demonstration
  - Transfer learning
  - Meta learning (e.g., autoML)
  - Learning + reasoning