Averaged-DQN: Variance Reduction and Stabilization

Yulun Wang
Department of Computer Science
National Chiao Tung University
xxx@nctu.edu.tw

1 Introduction

The recent Deep Q-Network algorithm (DQN) was the first to successfully combine Deep Neural Network (DNN), a powerful non-linear approximation technique, with the Q-learning algorithm. However, the max operation in Q-learning can lead to overestimation of state-action values in the presence of noise.

In this work, variance analysis that explains some of the DQN problems are done, and later on, a solution called Averaged-DQN is proposed to solve the overestimation problem. It extends DQN by taking Q-values learned from few iterations before and averaging those values to produce the current action-value estimate. In the end, also, showing better results in the experiments done on Arcade Lerning Enviroment (ALE).

Different from Double-DQN, which tackle the overestimation problem by replacing the positive bias into negative one, Averaged-DQN directly reduce the variance and get better results.

This work greatly increases the stability of Deep Q-learning algorithms, but only adds the computation time in linear growth. Also, only a few changes are made from DQN to Averaged-DQN, which means it's easy to implement even if one don't know the theory behind it. All of these features make the proposed method valuable.

2 Problem Formulation

Denote s as state, a as action, θ as model parameters, let's first take a look at DQN algorithm:

Algorithm 1 DQN

1: Initialize $Q(s, a, \theta)$ with random weights θ_0
2: Initialize Experience Replay buffer B
3: Initialize exploration procedure Explore($\cdot$)
4: **for** $i = 1, 2, 3..., N$ **do**
5: $\quad y^i_{s,a} = E_B[r + \gamma \max_{a'} Q(s^{'}, a^{'}, \theta_{i-1})|s, a]$
6: $\quad \theta_i \approx \arg\min_\theta E_B[(y^i_{s,a} - Q(s, a, \theta))^2]$
7: $\quad$ Explore($\cdot$), update B
8: **end for**

There are various types of errors that arise due to the combination of Q-learning and function approximation in the DQN algorithm. Let $\delta_i = Q(s, a, \theta_i) - Q^*(s, a)$, which can be rewritten as:

$$\delta_i = Q(s, a, \theta_i) - y^i_{s,a} + y^i_{s,a} - \hat{y}^i_{s,a} + \hat{y}^i_{s,a} - Q^*(s, a)$$

where $\hat{y}^i_{s,a}$ is true target:

$$\hat{y}^i_{s,a} = E_B[r + \gamma \max_{a^{'}}(y^{i-1}_{s^{'},a^{'}})|s, a]$$

Preprint. Under review.

If we can reduce the variance of δ_i, then the training progress will be more stable. To analyze the variance, let's first split the formula above into three parts:

- Target Approximation Error (TAE): $Z^i_{s,a} = Q(s, a, \theta_i) - y^i_{s,a}$
- Overestimation Error: $R^i_{s,a} = y^i_{s,a} - \hat{y}^i_{s,a}$
- Optimality Difference: $\hat{y}^i_{s,a} - Q^*(s, a)$

Optimality difference can be seen as the error of a standard tabular Q-learning. On the other hand, according to work proposed by Thrun & Schwartz (1993), assuming $Z^i_{s,a}$ is a random variable uniformly distributed in the interval $[-\epsilon, \epsilon]$, then the expected overestimation errors $E_Z[R^i_{s,a}]$ are upper bounded by $\gamma\epsilon\frac{n-1}{n+1}$ (where n is the number of applicable actions in state s).

Following from the mentioned observation, the magnitude of the bias is controlled by the variance of the TAE, and we'll focus on how to reduce the variance from TAE in the rest of this report.

In this work, two methods are mentioned: Ensemble DQN and Averaged DQN. Ensemble DQN owns K model weights and calculate the predicted action-value by averaging the output of K models:

Algorithm 2 Ensemble DQN

Initialize K Q-Networks $Q(s, a, \theta^k)$ with random weights θ^k_0 for $k \in \{1, ..., K\}$
Initialize Experience Replay buffer B
Initialize exploration procedure Explore(·)
for $i = 1, 2, 3..., N$ **do**
  $Q^E_{i-1}(s, a) = \frac{1}{K}\sum_{k=1}^{K} Q(s, a, \theta^k_{i-1})$
  $y^i_{s,a} = E_B[r + \gamma \max_{a'} Q^E_{i-1}(s^{'}, a^{'})|s, a]$
  for $k = 1, 2, 3..., K$ **do**
    $\theta^k_i \approx \arg\min_\theta E_B[(y^i_{s,a} - Q(s, a, \theta^k))^2]$
  end for
  Explore(·), update B
end for

Averaged DQN preserves weights from K iterations before and averaging the output to predict the action-value.

Algorithm 3 Averaged DQN

Initialize $Q(s, a, \theta)$ with random weights θ_0
Initialize Experience Replay buffer B
Initialize exploration procedure Explore(·)
for $i = 1, 2, 3..., N$ **do**
  $Q^A_{i-1}(s, a) = \frac{1}{K}\sum_{k=1}^{K} Q(s, a, \theta i - k)$
  $y^i_{s,a} = E_B[r + \gamma \max_{a'} Q^A_{i-1}(s^{'}, a^{'})|s, a]$
  $\theta_i \approx \arg\min_\theta E_B[(y^i_{s,a} - Q(s, a, \theta))^2]$
  Explore(·), update B
end for

In the next section, we'll go into the details of how they reduce the variance. Note that the whole analysis assumes that TAE is a random process such that in Averaged DQN:

- $E[Z^i_{s,a}] = 0$
- $Var[Z^i_{s,a}] = \sigma^2_s$
- $Cov[Z^i_{s,a}, Z^j_{s^{'},a^{'}}] = 0$ for all $i \neq j$ and $Cov[Z^i_{s,a}, Z^i_{s^{'},a}] = 0$ for all $s \neq s^{'}$

and in Ensemble DQN:

- $E[Z_{s,a}^{k,i}] = 0$
- $Var[Z_{s,a}^{k,i}] = \sigma_s^2$
- $Cov[Z_{s,a}^{k,i}, Z_{s^{'},a}^{k^{'},j}] = 0$ for all $i \neq j$ and $Cov[Z_{s,a}^{k,i}, Z_{s^{'},a^{'}}^{k^{'},j}] = 0$ for all $k \neq k^{'}$

In addition, the overestimation error is eliminated by considering a fixed policy for updating the target values.Also, since a zero reward has no effect on variance calculations, we can just assume $r = 0$ everywhere.

3 Theoretical Analysis

3.1 Ensemble-DQN

Following the Ensemble-DQN algorithm mentioned above:

$$Q_i^E(s_0, a) = \frac{1}{K}\sum_{k=1}^{K} Q(s_0, a, \theta_i^k) \tag{1}$$

$$= \frac{1}{K}\sum_{k=1}^{K}(Z_{s_0,a}^{k,i} + y_{s_0,a}^{i}) \tag{2}$$

$$= \frac{1}{K}\sum_{k=1}^{K} Z_{s_0,a}^{k,i} + y_{s_0,a}^{i} \tag{3}$$

$$= \frac{1}{K}\sum_{k=1}^{K} Z_{s_0,a}^{k,i} + \gamma Q_{i-1}^E(s_1, a) \tag{4}$$

$$= \frac{1}{K}\sum_{k=1}^{K} Z_{s_0,a}^{k,i} + \frac{\gamma}{K}\sum_{k=1}^{K} Z_{s_1,a}^{k,i-1} + \gamma y_{s_2,a}^{i-1} \tag{5}$$

We can get (4) from (3) since we assume that the reward is 0 in everywhere. In addition, $y_{s_{N-1},a}^{j} = 0$ in terminal states. By iteratively expanding $y_{s_2,a}^{i-1}$, we can get:

$$Q_i^E(s_0, a) = \sum_{n=0}^{N-1} \gamma^n \frac{1}{K}\sum_{k=1}^{K} Z_{s_n,a}^{k,i-n}$$

We also assume that TAEs are uncorrelated, meaning that $Var(\sum X_i) = \sum Var(X_i)$. In the end, given $Var[X] = E[X^2] + E[X]^2$ and $E[Z_{s,a}^{k,i}] = 0$:

$$Var[Q_i^E(s_0, a)] = \sum_{n=0}^{N-1} \frac{1}{K}\gamma^{2n}\sigma_{s_n}^2$$

3.2 Averaged-DQN

Following the Averaged-DQN algorithm mentioned above:

$$Q_i^E(s_0, a) = \frac{1}{K}\sum_{k=1}^{K} Q(s_0, a, \theta_{i+1-k}) \tag{6}$$

$$= \frac{1}{K}\sum_{k=1}^{K}(Z_{s_0,a}^{i+1-k} + y_{s_0,a}^{i+1-k}) \tag{7}$$

$$= \frac{1}{K}\sum_{k=1}^{K} Z_{s_0,a}^{i+1-k} + \frac{\gamma}{K}\sum_{k=1}^{K} Q_{i-k}^A(s_1, a) \tag{8}$$

$$= \frac{1}{K}\sum_{k=1}^{K} Z_{s_0,a}^{i+1-k} + \frac{\gamma}{K^2}\sum_{k=1}^{K}\sum_{k'=1}^{K} Q(s_1, a, \theta_{i+1-k-k'}) \tag{9}$$

Similar as before, assume reward is 0 everywhere, then we get (8) from (7). Also, $y^j_{s_{N-1},a} = 0$ in terminal states. By iteratively expanding $Q(s_1, a, \theta_{i+1-k-k'})$, we can get:

$$\begin{aligned} Q^A_i(s_0, a) = \frac{1}{K}\sum_{k_1=1}^{K} Z^{i+1-k_1}_{s_0,a} + \frac{\gamma}{K^2}\sum_{k_1=1}^{K}\sum_{k_2=1}^{K} Z^{i+1-k_1-k_2}_{s_1,a} + ... \\ + \frac{\gamma^{N-1}}{K^N}\sum_{k_1=1}^{K}\sum_{k_2=1}^{K}...\sum_{k_N=1}^{K} Z^{i+1-k_1-k_2-...-k_N}_{s_{M-1},a} \end{aligned} \tag{10}$$

To calculate $Var[Q^A_i(s_0, a)]$ easier, we'll compute the variance separately in equation (10) and sum them up. Note that assumptions like $Var(\sum X_i) = \sum Var(X_i)$ and $Var[X] = E[X^2] + E[X]^2$ will also be used in here.

For $C \in \{1, 2, ..., N\}$, denote:

$$V_C = Var[\frac{1}{K^C}\sum_{k_1=1}^{K}\sum_{k_2=1}^{K}...\sum_{k_C=1}^{K} Z_{k_1+k_2+...+k_C}] \tag{11}$$

where $Z_C, Z_{C+1}, ..., Z_{K*C}$ are independent and identically distributed TAE random variables, with $E[Z_i] = 0$ and $Var[Z_i] = \sigma^2_z$. Then:

$$\begin{aligned} V_C &= \frac{1}{K^{2C}} E[(\sum_{j=C}^{KC} n^C_j Z_j)^2] \\ &= \frac{\sigma^2_z}{K^{2C}}\sum_{j=C}^{KC}(n^C_j)^2 \end{aligned}$$

where n^C_j denotes how many times Z_j is counted in equation (11), and this can be found by converting the question to the number of solutions of the following equation:

$$k_1 + k_2 + ... + k_C = j \tag{12}$$

for $k_1, k_2, ..., k_C \in \{1, 2, ..., K\}$. After defining the new question, n^C_j can be written recursively:

$$n^C_j = \sum_{i=1}^{K} n^{C-1}_{j-i}$$

To be more precisely, consider equation (12) as distributing j same things into C same groups. Now, i items are given to a group, then the solution of the rest is $n^C - 1_{j-i}$. In the end, sum up the numbers of solution from $i = 1$ to $i = K$, and the answer will be n^C_j.

Since the goal of this calculation is to bound the variance reduction coefficient, we will calculate the solution in the frequency domain, in which the bound can be easily obtained. Denote

$$u^K_j = \begin{cases} 1, & \text{if } j \in \{1, 2, ..., K\} \\ 0, & \text{otherwise} \end{cases}$$

Trivially, n^C_1 will equal to u^K_j for any $j \in \mathbb{Z}$ when $C = 1$, and n^C_j can be written recursively as:

$$\begin{aligned} n^C_j &= \sum_{i=-\infty}^{\infty} n^{C-1}_{j-i} \cdot u^K_i \\ &\equiv (n^{C-1}_{j-i} \circledast u^K)_j \\ &= (u^K \circledast u^K ... \circledast u^K)_j \end{aligned}$$

where $\circledast$ is the discrete convolution.

Next, denote the Discrete Fourier Transform (DFT) of $u^K = (u^K_m)^{M-1}_{m=0}$ as $U = (U_m)^{M-1}_{m=0}$, and by using Parseval's theorem, we have:

$$\begin{aligned} V_C &= \frac{\sigma^2_z}{K^{2C}}\sum_{m=0}^{M-1} |u^K \circledast u^K ... \circledast u^K)_m|^2 \\ &= \frac{\sigma^2_z}{K^{2C}}\frac{1}{M}\sum_{m=0}^{M-1} |U_m|^{2C} \end{aligned}$$

where N is the length of the vectors u and U and is taken large enough so that the sum includes all nonzero elements of the convolution. Finally, we can sum up all of V_C:

$$\begin{aligned} Var[Q_i^A(s_0,a)] &= \sum_{n=1}^{N} V_n \gamma^{2(n-1)} \\ &= \sum_{n=1}^{N} \frac{1}{K^{2n}} \frac{1}{M} \sum_{m=0}^{M-1} |U_m|^{2n} \gamma^{2(n-1)} \sigma_{s_m}^2 \end{aligned}$$

3.3 Coefficient Bounding Analysis

First, let's calculate variance of original DQN, which is similar with Ensemble-DQN:

$$\begin{aligned} Q^{DQN}(s_0,a,\theta_i) &= Z_{s_0,a}^i + y_{s_0,a}^i \\ &= Z_{s_0,a}^i + \gamma Q(s_1,a,\theta_{i-1}) \\ &= Z_{s_0,a}^i + \gamma(Z_{s_1,a}^{i-1} + y_{s_1,a}^{i-1}) \end{aligned}$$

And the variance will be:

$$Var[Q^{DQN}(s_0,a,\theta_i)] = \sum_{n=0}^{N-1} \gamma^{2n} \sigma_{s_m}^2$$

According to the results from above,

$$Var[Q_i^E(s_0,a)] = \sum_{n=0}^{N-1} \frac{1}{K} \gamma^{2n} \sigma_{s_n}^2$$

$$Var[Q_i^A(s_0,a)] = \sum_{n=1}^{N} \frac{1}{K^{2n}} \frac{1}{M} \sum_{m=0}^{M-1} |U_m|^{2n} \gamma^{2(n-1)} \sigma_{s_m}^2$$

We can easily find out the difference between Ensemble-DQN and DQN:

Proposition 1 *Variance of Ensemble-DQN is K times smaller than DQN*

As for Averaged-DQN, let's first do some change to the equation:

$$\begin{aligned} Var[Q_i^A(s_0,a)] &= \sum_{n=1}^{N} \frac{1}{K^{2n}} \frac{1}{M} \sum_{m=0}^{M-1} |U_m|^{2n} \gamma^{2(n-1)} \sigma_{s_m}^2 \\ &= \sum_{n=0}^{N-1} \frac{1}{K^{2(n+1)}} \frac{1}{M} \sum_{m=0}^{M-1} |U_m|^{2(n+1)} \gamma^{2n} \sigma_{s_m}^2 \end{aligned}$$

then analyze the difference between Averaged-DQN and DQN by using Parseval's theorem, and the facts that $\frac{1}{K}|U_m| \leq 1$, $\frac{1}{K}|U_m| = 1$ only if $n = 0$:

$$\begin{aligned} \frac{1}{K^{2(n+1)}} \frac{1}{M} \sum_{m=0}^{M-1} |U_m|^{2(n+1)} &= \frac{1}{M} \sum_{m=0}^{M-1} |U_m/K|^{2(m+1)} \\ &< \frac{1}{M} \sum_{m=0}^{M-1} |U_m/K|^2 \\ &= \frac{1}{K^2} \sum_{m=0}^{M-1} |u_m^K|^2 \\ &= 1/K \end{aligned}$$

And this gives the following conclusion:

Proposition 2 *Variance of Averaged-DQN is at lease K times smaller than DQN, and even smaller than Ensemble-DQN*

4 Conclusion

In this report, firstly, discussed overestimation problem in DQN and some previous works that tries to deal with this problem. Secondly, introduce two methods proposed in the paper: Ensemble DQN and Averaged DQN, which are extensions of DQN and can be easily changed from DQN, without adding to much calculation. Thirdly, provide proofs of variance reduction from DQN to Average DQN (and Ensemble DQN), and point out that Averaged DQN can reduce more than Ensemble DQN. In addition, I cover some steps that are skipped in original paper, which may make the proof no straightforward, and also mentioned some properties that is used in the proof but not mentioned by the authors. Also, some typo is fixed simultaneously.

To further extend this work, researching over learning how many values to average for best result will be a feasible direction. In addition, maybe methods that reduces variance in overestimation error can put together with the proposed method since this work only focus on TAE.

References

[1] Oron Anschel, Nir Baram, and Nahum Shimkin. Averaged-DQN: Variance Reduction and Stabilization for Deep Reinforcement Learning, ICML 2017.

[2] Mnih, Volodymyr, Kavukcuoglu, Koray, Silver, David, Graves, Alex, Antonoglou, Ioannis, Wierstra, Daan, and Riedmiller, Martin. Playing Atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602, 2013.

[3] Thrun, Sebastian and Schwartz, Anton. Issues in using function approximation for reinforcement learning. In Proceedings of the 1993 Connectionist Models Summer School Hillsdale, NJ. Lawrence Erlbaum, 1993.