Paper Reading No.8

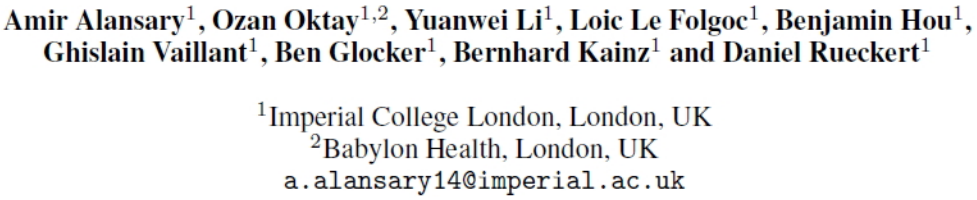
Evaluating Reinforcement Learning Agents for Anatomical Landmark Detection

Sheng Lian

July 2019

1 Brief Paper Intro

- ***Paper ref:*** Accepted by 1st Conference on Medical Imaging with Deep Learning (MIDL 2018), Amsterdam, The Netherlands. And recommended to Medical Image Analysis , https://doi.org/10.1016/j.media.2019.02.007
- ***Authors:*** See Fig. 1.



Amir Alansary[1]**, Ozan Oktay**[1,2]**, Yuanwei Li**[1]**, Loic Le Folgoc**[1]**, Benjamin Hou**[1]**,**
Ghislain Vaillant[1]**, Ben Glocker**[1]**, Bernhard Kainz**[1] **and Daniel Rueckert**[1]

[1]Imperial College London, London, UK
[2]Babylon Health, London, UK
a.alansary14@imperial.ac.uk

Figure 1: authors' brief intro.

- ***Paper summary:***A DQN-based method is proposed for automatic anatomical landmark detection. The authors applies it to 3D fetal head ultrasound scans, and get promising performance.

- ***Reading motivation:*** This paper is among the reading list provided by Dr. Li. At first this paper is accepted to MIDL as a oral paper, then recommended to MedIA journal. MIDL uses an open-review process, and all the review comments can be viewed on openreview, and this is funny. Applying reinforce learning (RL) to landmark localization is an interesting try for medical imaging community. Let's see how this paper do.

2 Backgrounds

Accurate detection of anatomical landmarks from medical images is an essential step for many image analysis and acquisition methods. Actually, this paper reminds me of the previous paper ***(Cardiac MRI Segmentation with Strong Anatomical Guarantees, in MICCAI 2019)*** I have just read, and the reading note can be seen here: DIG paper reading No.7.

In that paper, the 16 anatomical metrics in Sec 3.1 confused me. Because they are not differentiable, I don't know if they can be implemented automatically and how they work. It seems that this issue can be solved with the way proposed in this paper by locating key landmarks.

3 Reinforcement Learning and DQN Recap

Reinforcement Learning and DQN's recaps are well summarized in Skymind.AI blog and CSDN blog.

Here, I will briefly introduce the key points of RL from Q-learning to DQN.

3.1 Q-Learning

The optimal action selection policy can be identified by learning a state-action value function Q(s; a). The Q-function can be regarded as the expected future rewards, which goes as $E\left[r_{t+1}+\gamma r_{t+2}+\cdots+\gamma r_{t+n} \mid s, a\right] . \gamma \in [0,1]$. Using Bellman Equation, this function can be unrolled recursively:

$$Q_{i+1}(s, a)=E\left[r+\gamma \max _{a^{\prime}} Q_i\left(s^{\prime}, a^{\prime}\right)\right] \tag{1}$$

, where s' and a' are the next state and action.

3.2 Deep Q-Learning ((Nature-version DQN)

DQN (Nature-version DQN, referring 刘建平's blog, Nature DQN) is introduced in [1], where a target $Q\left(\omega^{-}\right)$ network that is periodically updated with the current Q(ω) every n iterations. In Nature DQN, it first find the maximum Q value in each action in the target Q network. DQN's loss function is

$$L_{DQN}(\omega)=E_{s, r, a, s^{\prime} \sim D}\left[\left(r+\gamma \max _{a^{\prime}} Q\left(s^{\prime}, a^{\prime} ; \omega^{-}\right)-Q(s, a ; \omega)\right)^2\right] \tag{2}$$

Here, D denotes an experience replay memory, which can be used to avoid problem of successive data sampling.

3.3 Double DQN (DDQN)

Unlike Nature DQN, DDQN [2] (referring 刘建平's blog, DDQN) first find the action corresponding to the maximum Q value in the current Q network.

The loss goes as:

$$L_{DDQN}(\omega) = E_{s,r,a,s' \sim D}\left[\left(r + \gamma \max_{a'} Q\left(s', Q\left(s', a; \omega\right); \omega^{-}\right) - Q(s, a; \omega)\right)^2\right] \tag{3}$$

With such modification, DDQN improves the stability of learning and may translate to the ability to learn more complicated tasks.

3.4 Duel DQN

In [3], action-state value function is decomposed into two more fundamental notions of value. The architecture of Duel DQN is indicated in Fig 2, where the fully connected (FC) layers are split into two paths: the state value V (s) and action advantage A(s; a) functions. Detailed introduction can be referred in 刘建平's blog, Duel DQN.

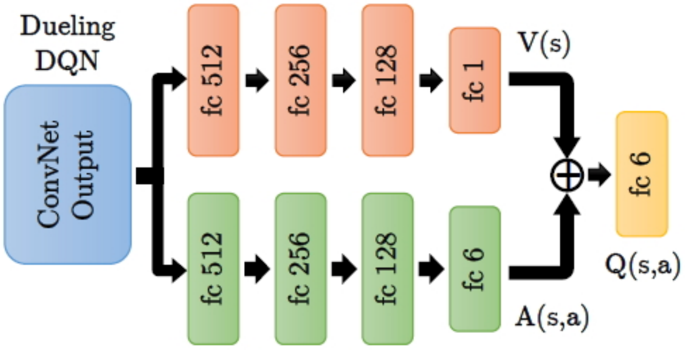


Figure 2: Duel DQN architecture.

4 Methodology

In this work, authors formulate the problem of landmark detection as an MDP. In this setup, the input 3D image defines the environment E. Also, following the main elements of MDP, we have set of actions A, set of states S, and reward function R.

The overall pipeline of the proposed method follows the definition of reinforcement learning, which can be summarized as follows:

Navigation actions The navigation actions set compose of six actions, $\{\pm a_x, \pm a_y, \pm a_z\}$, as is shown in Figure 3.

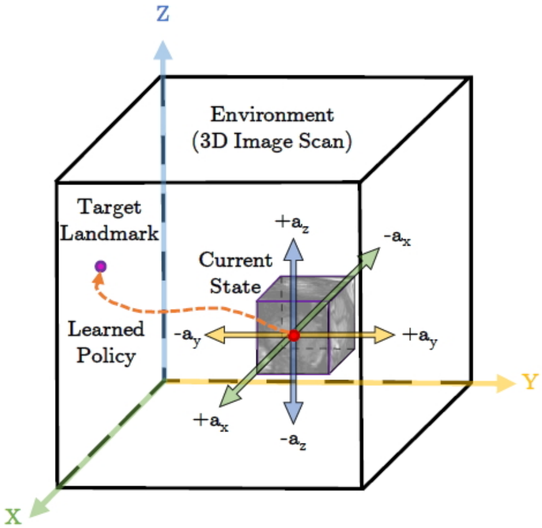


Figure 3: schematic visualization of the navigation actions in a 3D scan.

States Each state s defines a 3D region of interest (ROI) centered around the target landmark. A history buffer is used to capture the last 4 action steps (ROIs) for preventing the agent from getting stuck in repeated cycles.

Reward function The reward function goes as $R = D\left(P_{i-1}, P_t\right) - D\left(P_i, P_t\right)$, where D represents the Euclidean distance between two points. This formula measures whether the agent is moving closer to or further away from the desired target location.

Terminal state Authors define the terminal state during training when the distance between the current point of interest and the target landmark are less than or equal to 1mm. The training stage are a bit like supervised learning. While in testing stage, it is more challenging due to the absence of the landmark's true location. In this work, authors adopt the oscillation property to terminate the search process during testing, and choose the terminating state based on the corresponding lower Q-value. Because they find that Q-values are lower when the agent is closer to the target point and higher when it is far.

Multi-scale agent In this work, authors adopt a multi-scale agent strategy in a coarse-to-fine fashion with hierarchical action steps. Visual animation results can be seen HERE. This procedure can be summarized as follows:

1) the environment E Initial a fixed size $((S_x, S_y, S_z))$ image-grid around the current point of interest;

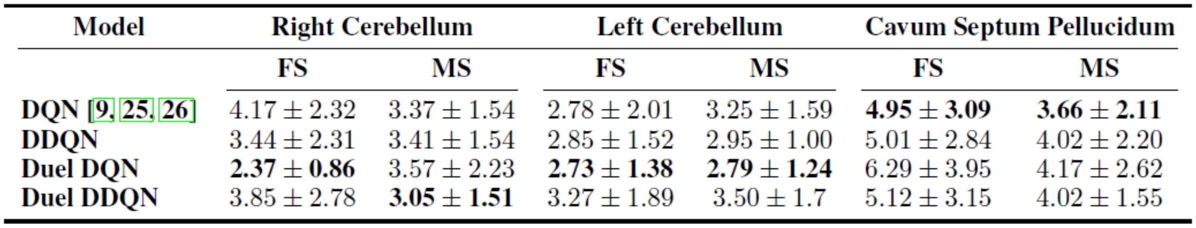
2) the agent searches for the target landmark, with relatively big action steps;

3) E samples the new image-grid with smaller spacing, as well as smaller action steps.

5 Experiment

In this paper, DQN-based methods is validated on 72 fetal head ultrasound scans. Authors measure the accuracy based on the distance error and implemented DQN, DDQN, Duel DQN, and Duel DDQN for comparison.

Here, fixed-scale (FS) and multi-scale (MS) search strategies on different DQN-based agents are compared. We can find that MS seems to act as a fine-tune step and works well in many situation. For different tasks, different DQN-based methods have different advantages.



Model	**Right Cerebellum**		**Left Cerebellum**		**Cavum Septum Pellucidum**	
	FS	**MS**	**FS**	**MS**	**FS**	**MS**
DQN [9, 25, 26]	4.17 ± 2.32	3.37 ± 1.54	2.78 ± 2.01	3.25 ± 1.59	$\mathbf{4.95 \pm 3.09}$	$\mathbf{3.66 \pm 2.11}$
DDQN	3.44 ± 2.31	3.41 ± 1.54	2.85 ± 1.52	2.95 ± 1.00	5.01 ± 2.84	4.02 ± 2.20
Duel DQN	$\mathbf{2.37 \pm 0.86}$	3.57 ± 2.23	$\mathbf{2.73 \pm 1.38}$	$\mathbf{2.79 \pm 1.24}$	6.29 ± 3.95	4.17 ± 2.62
Duel DDQN	3.85 ± 2.78	$\mathbf{3.05 \pm 1.51}$	3.27 ± 1.89	3.50 ± 1.7	5.12 ± 3.15	4.02 ± 1.55

Figure 4: Comparison of different DQN-based agents using fixed-scale (FS) and multi-scale (MS) search strategies.

6 My thoughts

- Different DQN-based methods are evaluated for automatic landmark detection on challenging fetal head ultrasound images.

- Experiments shows the effectiveness of multi-scale search strategy.

- The learning route of Reinforcement learning, Q-Learning, DQN, etc., covers lots of complex contents. I haven't implemented reinforcement learning related models, and feel confused about the terminal state 4. The Q-values is inference-based, how can it correctly guide the agent's movement during the testing phase?

- This paper does not list results compared to other non-reinforcing learning models.

References

[1] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

[2] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Thirtieth AAAI conference on artificial intelligence*, 2016.

[3] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, and Nando Freitas. Dueling network architectures for deep reinforcement learning. In *International Conference on Machine Learning*, pages 1995–2003, 2016.