A Comparative Study of Reinforcement Learning Algorithms Applied to Medical Image Registration

Ramiro F. Isa-Jara[1(✉)], Gustavo J. Meschino[2], and Virginia L. Ballarin[1]

[1] Image Processing Laboratory, ICYTE-CONICET,
Universidad Nacional de Mar del Plata, Mar del Plata, Argentina
ramfer.isaj@gmail.com
[2] Bioengineering Laboratory, ICYTE-CONICET,
Universidad Nacional de Mar del Plata, Mar del Plata, Argentina

Abstract. Nowadays, several medical procedures depend on the comparison and combination of images obtained in different modalities (magnetic resonance, computed tomography, PET, among others). Image registration is a geometric transformation process to align two or more images. It is necessary to have robust algorithms to find the best parameters of transformation in order to achieve accurate registrations. Reinforcement learning allows to train an agent through direct environment interaction, to achieve a goal. In this work, a comparison of the performance of Q-learning and Deep-Q with its variants is presented. Brain magnetic resonance images are used in 2D domain considering rigid deformations. The comparison is based on the reward values, computing the Pearson correlation factor in monomodal registration and Mutual information in multimodal registration, obtained during the learning process. It is also considered an error measure between the target parameters and the achieved ones. Finally, a backup memory criterion is proposed to train the Q-Network methods. Experimental results show a successfully behavior in all cases, but performance is improved when the proposed criterion is applied.

Keywords: Q-learning · Deep-Q Networks · Reinforcement learning · Image registration · Machine learning

1 Introduction

Reinforcement Learning (RL) is one of the three main paradigms of Machine Learning. RL is composed by: an agent, an environment and its model, a policy, a reward signal and a value function [1]. The main idea is that an agent can be trained through direct interaction with its environment, therefore it is a learning process to take actions to maximize a numerical reward signal [1, 2].

Registration is a geometrical aligning process between two or more images, taken in different times, views or modalities, for comparison and combination of information purposes [3]. In the medical field, this process has an important role, especially in

C. A. González Díaz et al. (Eds.): CLAIB 2019, IFMBE Proceedings 75, pp. 281–289, 2020.
https://doi.org/10.1007/978-3-030-30648-9_36

patient monitoring and treatment [4]. This requires robust algorithms that allow to obtain better results using reasonable time and computing resources.

The aim of this work is to apply RL algorithms, *Q-learning, Deep-Q-Network (DQN), Double DQN and Dueling DQN,* to image registration tasks in order to compare their performances. Moreover, a backup memory criterion, as a novel way to train a neural network, is introduced in order to improve the results of DQN algorithms. It is applied in Double DQN to analyze their performances.

This work is organized as follows: Sect. 2 summarizes the algorithms used; Sect. 3 describes their application in an image registration process; Sect. 4 shows the obtained results and Sect. 5 presents the conclusions and future work.

2 Reinforcement Learning Algorithms

2.1 General Description

Reinforcement learning is based on dynamic system theory for optimal control. Markov decision processes (MDPs) are used due its mathematical background. MPDs are the formalization of sequential decision making where actions can influence not just immediate rewards but also in future states and rewards [1].

A finite *Markov Decision process* is defined by a tuple $\langle S, A, T, R, E \rangle$, where S is the agent's state-space, A is the action-space, T is the dynamic transition in reference to $T(s, a, s')$ i.e. the probability to select an action a in current state s to obtain the next state s', R is the numeric reward value obtained when an agent change the state s to s' selecting an action, and E is the terminal states set to avoid future transitions where $E \subset S$ [5]. Thus, the probability of those values occurring at time t, given particular values of the state and actions, is defined as:

$$p(s', r|s, a) = \Pr\{S_t = s', R_t = r|S_{t-1} = s, A_{t-1} = a\}. \tag{1}$$

2.2 Q-Learning Algorithm

Q-learning is a RL technique derived from Temporal Difference (TD) methods. TD methods combine properties of Dynamic programming algorithms and Monte-Carlo methods. The goal is to learn a policy, which tells an agent what action to take in different states [6].

In order to adjust the polices, Watkins [7] introduced an *action-value function* $Q^{\pi}(s, a)$ which is the expected value in the initial state s, selecting an action a through the policy π. According to the Bellman equation, the Q function [8] for an optimal policy, is defined as:

$$Q^*(s, a) = \sum\nolimits_{s' \epsilon S} T(s, a, s')[R(s, a, s') + \gamma \max\nolimits_{a'} Q^*(s', a')], \tag{2}$$

where, $0 \leq \gamma \leq 1$, is the discounting factor to evaluate how valuable is the immediate reward in terms of the expected final reward.

Function Q updating is simply performed with instantaneously available information, which is a function of the optimal actions (normally a greedy policy is used) [6]. The update of function Q at time t can be written as:

$$\begin{gathered}\delta_t = \left[r_{t+1} + \gamma \max_{a'} Q_t(s_{t+1}, a') - Q_t(s_t, a_t)\right] \\ Q_{t+1}(s_t, a_t) \leftarrow Q_t(s_t, a_t) + \alpha_t(s_t, a_t)\delta_t,\end{gathered} \tag{3}$$

where α is the learning rate, $\alpha \epsilon (0, 1]$.

2.3 Deep-Q Network Algorithm

The DQN algorithm, proposed by Mnih [9], uses a multilayer neural network that has an array of states S as input and an array of actions A as output. The *action-value function* will be $Q(s, .; \theta)$, where θ are the weights of the network called *Deep*. The function Q is calculated by updating θ based on experience [10, 11].

In this case, an agent learns the parameterized function $Q(s, a; \theta_t)$:

$$\theta_{t+1} = \theta_t + \alpha\big(Y_t^Q - Q(s_t, a_t; \theta_t)\big)\nabla_{\theta_t} Q(s_t, a_t; \theta_t), \tag{4}$$

where α is the learning rate and Y_t^Q are the target values computed by: $Y_t^Q = r_{t+1} + \gamma \max_a Q(s_{t+1}, a; \theta_t)$.

2.4 Double DQN Algorithm

If the same *action-value function* is used to evaluate and select an action a, it may lead to over-optimization. In the Double DQN algorithm, an agent learns two *action-value functions* through assignment of random experiences to obtain two sets of parameters or weights θ and θ' [10]. To update the function, one set of parameters is used to determine *the policy* and the other one is used to compute *the value*. The target values in Double DQN algorithms are computed as:

$$Y_t^{DoubleQ} = r_{t+1} + \gamma Q\Big(\max_a Q(s_{t+1}, a; \theta_t); \theta_t^{'}\Big). \tag{5}$$

2.5 Dueling DQN Algorithm

It is usually assumed that the choice of an action is always a priority, but there are some cases, especially in a free-model problem, where selecting an action or another does not affect the overall performance of the agent [12]. Therefore, *the action-value function* $Q(s, a; \theta_t)$ can be separated into its two main subfunctions:

$$Q(s, a; \theta, \alpha, \beta) = V(s; \theta, \beta) + A(s, a; \theta, \alpha), \tag{6}$$

where $V(s)$ is the *value function*, $A(s,a)$ is the *advantage function*, θ is the set of parameters of the neural network and α, β are the set of parameters corresponding to $V(s)$ and $A(s,a)$. The subfunctions are combined in the output layer in order to obtain the function Q [13].

3 Description of Problem and Applications of Algorithms

3.1 Medical Image Registration

Two images are considered during the registration process: the *target image* and the *moving image*. The moving image is deformed to align to the coordinate system of the target image [14].

A geometric transformation is estimated considering 3 parameters: *scale, orientation and translation.* Monomodal and multimodal registration are performed in the 2D domain with the translation parameter over the X and Y axes.

3.2 Image Description

The image dataset is composed of brain magnetic resonance images (MRI) in T1 and T2 modalities and SPECT images (Fig. 1). The images were provided by the Radiologic Institute of Mar del Plata, Argentina. In these images, only rigid deformations can occur during acquisition.

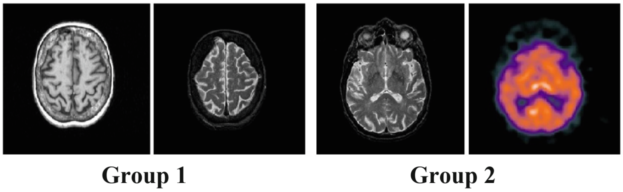

Fig. 1. Brain magnetic resonance images used in tests. **G1** - MRI of the same slide in T1 and T2 weighted modalities. **G2** - MRI of same slide in T2 weighted and SPECT modalities

T1 and T2 images are sized 256 × 256, 8 bits, grayscale. SPECT images have the same size, RGB color space, 8 bits per channel.

3.3 Software and Hardware Description

RL algorithms were implemented in *Python 3.6*, using *TensorFlow 1.10*. The tests were performed using a Core-i5 computer with 8 GB RAM and Debian operating system.

3.4 RL Algorithms in Image Registration

RL algorithms have been adapted for image registration taking the next considerations:

1. *Environment:* composed by the target image and the moving image. The moving images are created using the parameters of Table 1 as targets, in order to measure the error achieved by each algorithm.

Table 1. Parameters used to generate the testbench to test RL algorithms.

Registration modality	Test 1 $[\boldsymbol{s}, \boldsymbol{\theta}, \boldsymbol{t_x}, \boldsymbol{t_y}]$	Test 2 $[\boldsymbol{s}, \boldsymbol{\theta}, \boldsymbol{t_x}, \boldsymbol{t_y}]$
G1 T1 – T1	$[1.2, 29°, -10, 1]$	$[0.8, -29°, 10, -5]$
G1 T1 – T2	$[1.0, -15°, 0, -7]$	$[1.0, 15°, 8, 0]$
G2 T2 – T2	$[1.15, -10°, -5, -15]$	$[0.85, 10°, 7, 12]$
G2 T2-SPECT	$[1.1, -45°, -9, 6]$	$[0.9, 45°, -17, 12]$

2. *States:* described as the set $[s, \theta, t_x, t_y]$, where s is the scale value, θ is the orientation angle, and t_x and t_y are the values to move pixels over the X and Y axes. The initial state is set as $[1.0, 0°, 0, 0]$.
3. *Actions:* the total number of actions is $2^4 = 16$ because there are 4 transformation parameters and 2 possibilities for each one (increment or decrement). The parameter step is defined by $[\pm 0.05, \pm 0.25°, \pm 1, \pm 1]$.
4. *Reward:* Pearson correlation coefficient in monomodal registration and Mutual information in multimodal registration are considered. These similarity measures are maximized during the registration process. When an action produces a decrease in similarity, the agent receives a penalty of -0.1 and the algorithm restarts the learning process.

3.5 Backup Memory Criterion

In DQN algorithms, the agent experiences are stored in order to train the neural networks. A batch technique is used for this purpose [15]. In the backup memory proposed in this work, best experiences are saved, and they are only replaced by others with better reward values, which prevents these good solutions from being lost.

In a random position of the batch, current information is replaced by experiences of the backup memory. Therefore, the batch contains information from the database of experiences (good and bad ones) and from the backup memory (best experiences). In this work, backup memory is used with the Double DQN algorithm and the results show a good performance compared to the other algorithms.

4 Results

The initial parameters are the same for all algorithms. The number of epochs is 200 and the learning process runs 100 iterations, though it can end before if the agent receives a penalty. The tests are evaluated along 5 independent runs.

In Figs. 2 and 3, the mean values of the Pearson coefficient for monomodal registration are presented. In Fig. 3, the double-back_ algorithm, which uses the backup memory criterion, achieved a better correlation factor than the other algorithms.

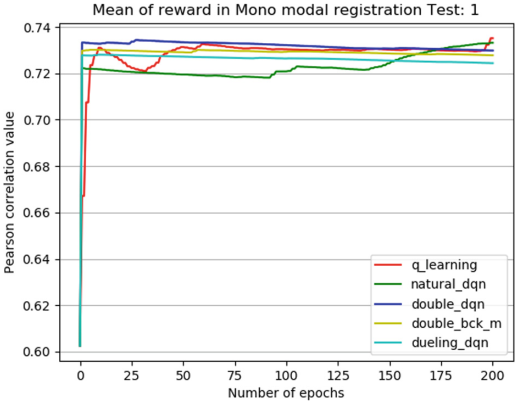

Fig. 2. Mean of monomodal registration results obtained according to parameters of Test 1.

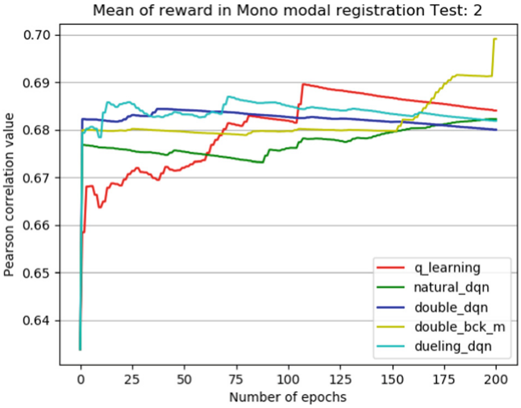

Fig. 3. Mean of monomodal registration results obtained according to parameters of Test 2.

In Figs. 4 and 5, the mean values of mutual information for multimodal registration are presented. The double-back_ algorithm shows a similar performance to other Deep Networks algorithms.

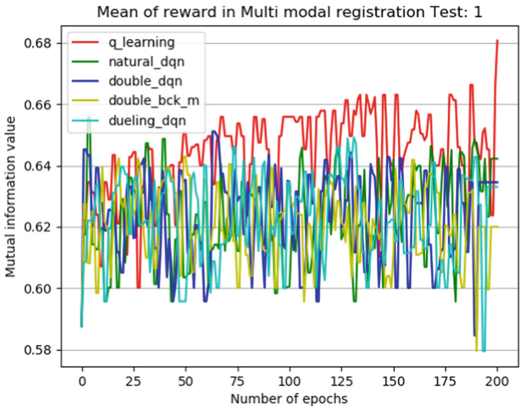

Fig. 4. Mean of multimodal registration results obtained according to the parameters of Test 1.

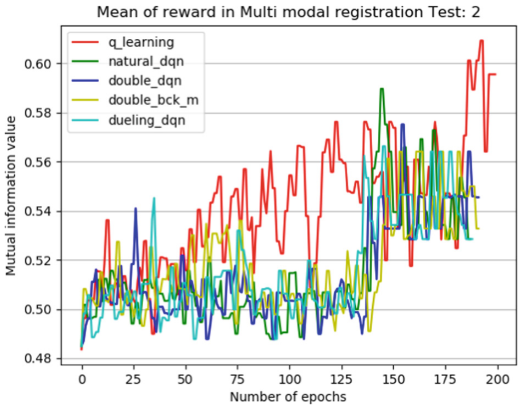

Fig. 5. Mean of multimodal registration results obtained according to the parameters of Test 2.

In Tables 2 and 3, the mean values of similarity measures and errors computed for monomodal and multimodal registration tests are presented. The tables show the performance of each algorithm through mean values of similarity measures and error between target and obtained parameters.

Table 2. Performance of RL algorithms achieved during Monomodal registration.

RL algorithms	Test 1		Test 2	
	Pearson value	% Error	Pearson value	% Error
Q-learning	0.54	0.27	0.63	0.36
Natural-DQN	0.51	0.48	0.49	0.25
Double-DQN	0.62	0.23	0.55	0.25
Double-back_	0.62	0.24	0.55	0.28
Dueling-DQN	0.61	0.37	0.52	0.72

Table 3. Performance of RL algorithms achieved during Multimodal registration.

RL algorithms	Test 1		Test 2	
	Mutual info	% Error	Mutual info	% Error
Q-learning	0.65	0.23	0.58	0.39
Natural-DQN	0.67	0.15	0.40	0.43
Double-DQN	0.67	0.16	0.42	0.76
Double-back_	0.66	0.16	0.58	0.37
Dueling-DQN	0.66	0.15	0.59	0.20

Finally, average of times spent during the tests are presented in Table 4.

Table 4. Average time spent by each algorithm during tests.

Algorithms	Test 1	Test 2
Q-learning	14.40 min	19.22 min.
Natural-DQN	20.60 min	22.46 min
Double-DQN	30.55 min	15.67 min
Double-back_	15.12 min	29.71 min
Dueling-DQN	61.27 min	55.46 min

5 Conclusions

In this work, the Reinforcement Learning paradigm is applied to medical image registration. A performance comparison study between four RL algorithms is presented. The registration process is an optimization problem where a similarity measure is maximized. RL allows that an agent could be trained in order to find the best parameters in medical images considering rigid deformations.

In all cases, the algorithms gave successfully results which allow a significant improvement in similarity between the target images and registered images. According to the results, Q-learning, Double-DQN and Double-DQN with backup memory show better performances than the others.

According to the results, the mean error obtained by Double-back is reduced around of 37% in Test1 and 35% in Test2 compared with the highest error achieved by each algorithm. Q-learning and Double-DQN reduce the mean error around of 24% in Test1 and 19% in Test2. Moreover, these algorithms also use a reasonable time for the training process which is important in reference to computational resources available.

Future work focuses on analyzing the optimum size of batch and backup memories to optimize the training of the deep neural networks which could improve their performance and to maximize the reward values. The RL paradigm could also be applied in nonlinear registration.

Conflict of Interest. The authors declare that they have no conflict of interest.

References

1. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction, 2nd edn. (2017)
2. Kaelbling, L.P., Littman, M.L., Moore, A.W.: Reinforcement learning: a survey. J. Artif. Intell. Res. **4**, 237–285 (1996)
3. Zitová, B., Flusser, J.: Image registration methods: a survey. Image Vis. Comput. **21**, 977–1000 (2003)
4. Maintz, J.B.A., Viergever, M.A.: A survey of medical image registration. Med. Image Anal. **2**(1), 1–37 (1998)
5. Roderick, M., MacGlashan, J., Tellex, S.: Implementing the deep Q-network, November 2017
6. Sigaud, O., Buffet, O.: Markov decision processes in artificial intelligence (2010)
7. Watkins, C.J.C.H., Dayan, P.: Q-learning. Mach. Learn. **8**(3–4), 279–292 (1992)
8. Bellman, R.: A Markovian decision process (1957)
9. Mnih, V., et al.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529–533 (2015)
10. van Hasselt, H., Guez, A., Silver, D.: Deep reinforcement learning with double Q-learning, September 2015
11. Lin, L.-J.: Self-improving reactive agents based on reinforcement learning, planning and teaching. Mach. Learn. **8**(3–4), 293–321 (1992)
12. Wang, Z., Schaul, T., Hessel, M., Com, M., Van Hasselt, H., Lanctot, M.: Dueling network architectures for deep reinforcement learning (2015)
13. Harmon, M.E., Baird, L.C., Klopr, A.H.: Advantage updating applied to a differential game. In: NIPS (1995)
14. Fitzpatrick, J.M., Hill, D.L.G., Maurer, C.R.: Image registration. In: Handbook of Medical Imaging. Medical Image Processing and Analysis, vol. 2, pp. 447–514 (2000)
15. Hoffer, E., Hubara, I., Soudry, D.: Train longer, generalize better: closing the generalization gap in large batch training of neural networks (2017)