BMC Medical Imaging

RESEARCH ARTICLE

Open Access

Reinforcement learning using Deep *Q* networks and *Q* learning accurately localizes brain tumors on MRI with very small training sets

J. N. Stember[1*] and H. Shalu[2]

Abstract

Background: Supervised deep learning in radiology suffers from notorious inherent limitations: 1) It requires large, hand-annotated data sets; (2) It is non-generalizable; and (3) It lacks explainability and intuition. It has recently been proposed that reinforcement learning addresses all three of these limitations. Notable prior work applied deep reinforcement learning to localize brain tumors with radiologist eye tracking points, which limits the state-action space. Here, we generalize Deep Q Learning to a gridworld-based environment so that only the images and image masks are required.

Methods: We trained a Deep *Q* network on 30 two-dimensional image slices from the BraTS brain tumor database. Each image contained one lesion. We then tested the trained Deep Q network on a separate set of 30 testing set images. For comparison, we also trained and tested a keypoint detection supervised deep learning network on the same set of training/testing images.

Results: Whereas the supervised approach quickly overfit the training data and predictably performed poorly on the testing set (11% accuracy), the Deep *Q* learning approach showed progressive improved generalizability to the testing set over training time, reaching 70% accuracy.

Conclusion: We have successfully applied reinforcement learning to localize brain tumors on 2D contrast-enhanced MRI brain images. This represents a generalization of recent work to a gridworld setting naturally suitable for analyzing medical images. We have shown that reinforcement learning does not over-fit small training sets, and can generalize to a separate testing set.

Keywords: Deep reinforcement learning, Reinforcement learning, Gridworld, Localization, Regression, Brain tumors

Introduction

Recently, reinforcement learning (RL, used interchangeably with the term deep reinforcement learning) has shown tremendous promise for landmark localization. Researchers have recently applied RL successfully to landmark or lesion localization in various image types and modalities [1–4]. Examples of applications are localization of breast lesions [5], lung nodules [6], anatomic landmarks on cardiac MRI [7] and vessel centerline tracing [8]. However, not much work has been done in the field of brain lesion localization with RL.

In recent work [9], Stember and Shalu applied RL to localize brain tumors on MRI. They sought to address

*Correspondence: joestember@gmail.com

[1] Department of Radiology, Memorial Sloan Kettering Cancer Center, 1275 York Avenue, Box 29, New York, NY 10065, USA
Full list of author information is available at the end of the article

three key shortcomings in current supervised deep learning approaches:

1. Requirement of large amounts of expert-annotated data.
2. Lack of generalizability, making it "brittle" and subject to grossly incorrect predictions when even a small amount of variation is introduced. This can occur when applying a trained network to images from a new scanner, institution, and/or patient population [10, 11].
3. Lack of insight or intuition into the algorithm, thus limiting confidence needed for clinical implementation and curtailing potential contributions from non-AI experts with advanced domain knowledge (e.g., radiologists or pathologists) [12, 13].

Their initial proof-of-principle application of RL to medical images used 2D slices of image volumes from the publicly available 2014 BraTS primary brain tumor database [14]. These T1-post-contrast images included one tumor per image. In addition, their images included an overlay of eye tracking gaze points obtained during a previously performed simulated image interpretation. The state-action space was limited to the gaze plots, consisting of the gaze points for that image. The gaze plots were essentially one-dimensional, and the various possible agent states were defined by location along the gazeplot. Actions were defined by the agent moving anterograde versus retrograde along the gaze plot, or by staying still. As a localization task, the goal was for the agent to reach the lesion. Using the manually traced tumor mask images, a reward system was introduced that incentivized finding and staying within the lesion, and discouraged staying still while the agent was still outside the tumor [9].

The results from this study showed that RL has the potential to make meaningful brain lesion localization predictions based on very small data sets (in this case, 70 training set images). Supervised deep learning woefully overfit the training set, with unsurprisingly low accuracy on the testing set (around 10%). In contrast, RL improved steadily with more training, ultimately predicting testing set image lesion location with over 80% accuracy [9].

However, the system studied was not generalized, as it included eye tracking points, which are usually not available with radiological images. Additionally, the eye tracking points confined the state-action space to one dimension. In order to apply RL more generally to medical images, we must be able to analyze raw images along with accompanying image masks without the need for eye tracking gaze plots.

In this study, we generalize the approach to show that RL can effectively localize lesions using a very small training set using the gridworld framework, which requires only raw images and the accompanying lesion masks. This represents an important early step in establishing that RL can effectively train and make predictions about medical images. This can ultimately be extended to 3D image volumes and more sophisticated implementations of RL. Gridworld is a classic, paradigmatic environment in RL [15]. Given their pixelated character, medical images tiled with a gridworld framework provide a natural, readily suitable environment for our implementation.

Methods

Basic terms

Following the basic approach of recent work [9], we analyzed 2D image slices from the BraTS 2014 public brain tumor database [14]. Since these are publicly available images with no patient identifying information, this study did not require IRB approval. These slices were randomly selected from among T1-weighted contrast-enhanced image slices that included brain tumor. Images from around the level of the lateral ventricles were selected. We used the BraTS 2014 data set specifically because it had used in an earlier, less generalized study using eye tracking points [9]. We wished to minimize the variables/confounders between these two studies.

As in the recent work, we implementated a combination of standard $TD(0)$ Q-learning with Deep Q learning (DQN). The key difference was how we defined the environment, states, and actions.

We divided the image space of the 240×240 pixel images by grids spaced 60 pixels, so that our agent occupied the position of a 60×60 pixel block, shown in Figs. 1 and 2. The initial state for training and testing images was chosen to be the top-left block (Fig. 1a). The action space consisted of: (1) staying at the same position, (2) moving down by one block, or (3) moving to the right by one block. In other words, introducing some notation, the action space $\mathcal{A} \in \mathbb{N}_0^3$, consisting of three non-negative integers, is defined by:

$$\mathcal{A} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = \begin{pmatrix} \text{stay still} \\ \text{move down} \\ \text{move right} \end{pmatrix}. \tag{1}$$

To each of the possible actions, $a \in A$, given a policy π, there is a corresponding action value depending on the state, $Q^\pi(s,a)$, defined by:

$$\begin{aligned} Q^\pi(s,a) &= E_\pi\{R_t | s_t = s, a_t = a\} \\ &= E_\pi\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a\right\} \end{aligned} \tag{2}$$

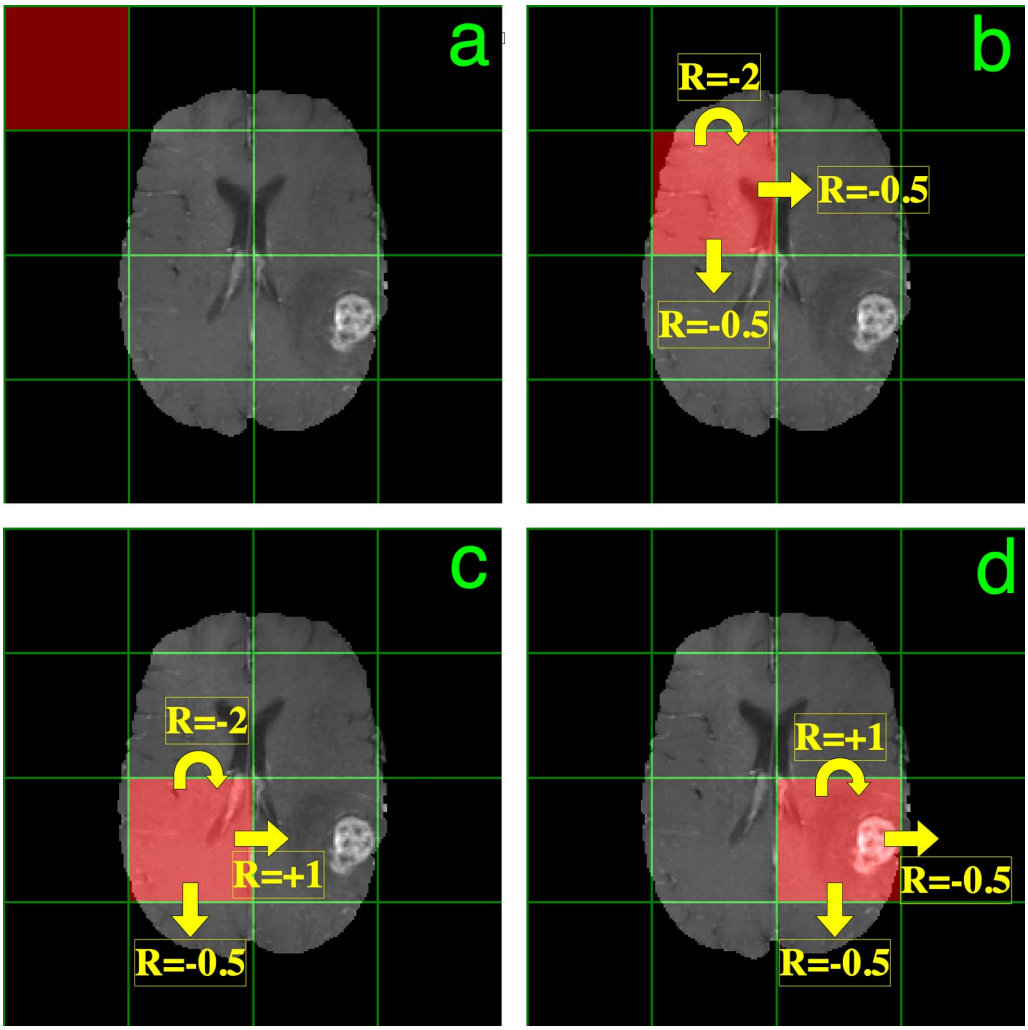

Fig. 1 Environment and reward scheme for training. **a** Shows the initial state (s_1) for all episodes, with the agent in the upper left corner. **b**–**d** Display the rewards in different states for the three possible actions. When the agent is not in a position overlapping or next to the lesion (b), staying in place gets the biggest penalty (reward of − 2), with a lesser penalty if the agent moves (reward of − 0.5). **c** Shows the rewards for the possible actions in the state just to the left of the mass. Moving toward the lesion so that the agent will coincide with it receives the largest possible and only positive reward (+ 1). **d** Shows the state with the agent coinciding with the lesion. Here we want the agent to stay in place, and thus reward this action with a + 1 reward

where R_t is the total cumulative reward starting at time t and $E_\pi\left\{\sum_{k=0}^{\infty}\gamma^k r_{t+k+1}|s_t = s, a_t = a\right\}$ is the expectation for R_t upon selecting action a in state s and subsequently picking actions according to π [15].

Training: sampling and replay memory buffer

In each of the $N_{\text{episodes}} = 90$ episodes of training, we sampled randomly from the 30 training set images. For each such image we subdivided into grids, and initialized such that the first state s_1 was in the upper left block (Fig. 1a). We selected each action a_t at time step t according to the off-policy epsilon-greedy algorithm, which seeks to balance exploration of various states with exploiting the known best policy, according to

$$a_t = \begin{cases} max_{a\in A}\{Q_t(a)\} & \text{with probability } \epsilon \\ \text{random action in } A & \text{with probability } 1-\epsilon. \end{cases} \quad (3)$$

for the parameter $\epsilon < 1$. We used an initial ϵ of 0.7 to allow for adequate exploration. As Q learning proceeds, and we wished to increasingly favor exploitation of a better known and more optimal policy, we set ϵ to decrease by a rate of 1×10^{-4} per episode. The decrease continued

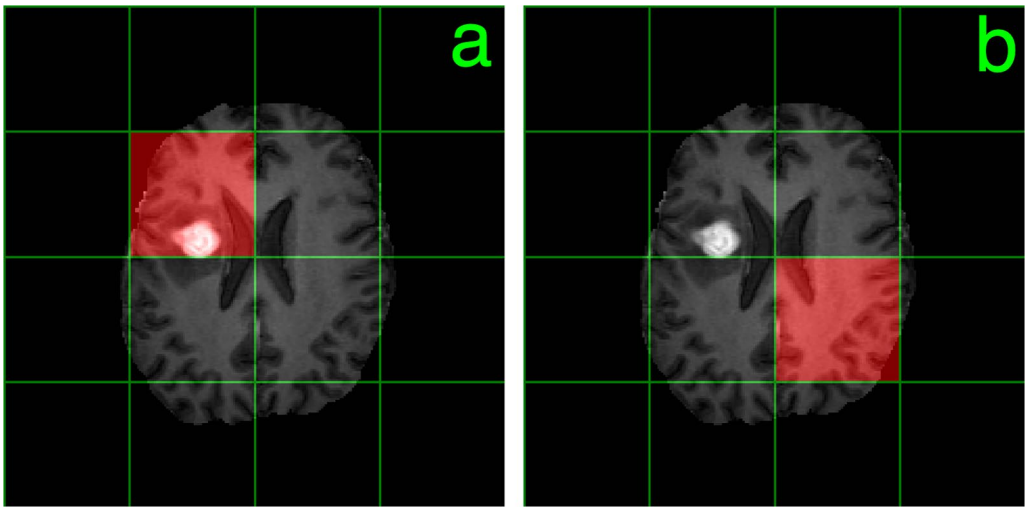

Fig. 2 Two possible testing/deployment results. **a** shows a case of an accurate prediction, a true positive. After the 20 steps of forward inference on a presumed testing set image, the agent overlies the lesion. **b** shows a testing set miss, a false positive, where the agent does not overlap the lesion. In this particular case, there is no way for the lesion to get back to the lesion, since only the three actions of stay in place, move down and move to the right are defined in our formulation, although a more general formulation with 5 directions is possible in future work

down to a minimum value $\epsilon_{\text{min}} = 1 \times 10^{-4}$, so that some amount of exploring could always take place.

Our reward scheme is illustrated for sample states in Figs. 1b–d. The reward r_t is given by

$$r_t = \begin{cases} -2, \text{if agent is outside the lesion and staying still} \\ +1, \text{if agent overlaps the lesion and is staying still} \\ -0.5, \text{if agent moves to a position outside the lesion} \\ +1, \text{if agent moves to a position overlapping the lesion.} \end{cases} \quad (4)$$

Replay memory buffer

In general, for each time t we thus have state s_t, the action we have taken a_t, for which we have received a reward r_t and which brings our agent to the new state s_{t+1}. We store these values in a tuple, called a transition, as $\mathcal{T}_t = (s_t, a_t, r_t, s_{t+1})$. For each successive time step, we can stack successive transitions as rows in a transition matrix $\mathbb{T}$. We do so up to a maximum size of $N_{\text{memory}} = 15,000$ rows. These represent the replay memory buffer, which allows the CNN that predicts Q values to sample and learn from past experience sampling from the environments of the various training images. Then, we use $\mathbb{T}$ to train the CNN and perform Q learning [15]. The value of $N_{\text{memory}} = 15,000$ was chosen to be as large as possible without overwhelming the available RAM.

Training: Deep *Q* network and *Q* learning

Using a CNN to approximate the function $Q_t(a)$, we give the CNN the name of Deep Q network (DQN). The architecture of the DQN, shown in Fig. 3, is very similar to that of recent work [9]. It takes the state as input, using 3×3 kernels with stride of 2 and padding such that the resulting intermediate output layer sizes are unchanged. We produced 32 intermediate output channels at each convolution block. The network consisted of four such convolutional layers in sequence, using exponential linear unit (elu) activation. The last output volume was flattened and followed by a 512-node layer with elu activation, followed by a few more fully connected layers and ultimately to a 3-node output layer representing the 3 actions and corresponding Q values.

Our DQN loss is the difference between the Q values resulting from a forward pass of the DQN, which at time step t we shall denote as Q_{DQN}, and the "target" Q value, Q_{target}, computed by the Bellman equation [15]. The latter updates by sampling from the environment and experiencing rewards. Denoting the forward pass by F_{DQN}, we can obtain state-action values by

$$Q^{(t)}_{\text{DQN}} = F_{\text{DQN}}(s_t). \quad (5)$$

But the function approximation we wish to learn is for the *optimal* state-action values, which maximize expected total cumulative reward. We do so by Q learning, in which the model learns from sampled experience, namely individual state-action pair-associated rewards. The method used in recent work [9], and that we employ here, is temporal difference in its simplest form: $TD(0)$. With $TD(0)$ the state-action value function is updated in each step of sampling to compute the $TD(0)$ target, denoted by $Q^{(t)}_{\text{target}}$ [15]:

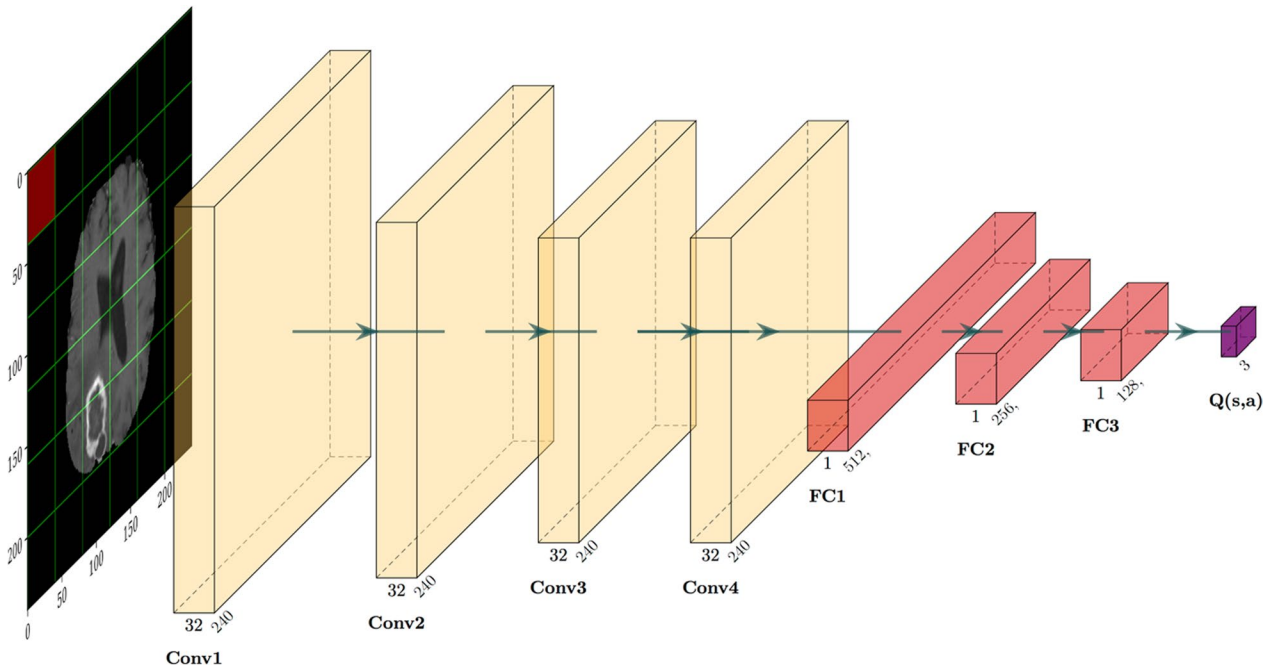

Fig. 3 Deep Q network architecture. Of note, the output consists of the three Q values corresponding to the three possible actions. A sample input image representing the initial state s_1 is also noted

$$Q^{(t)}_{\text{target}} = r_t + \gamma max_a Q(s_{t+1}, a), \tag{6}$$

where γ is the discount factor and $max_a Q(s_{t+1}, a)$ is another way of writing the state value function $V(s_{t+1})$. The key part of the environment sampled is the reward value r_t. Over time, with this sampling, $Q^{(t)}_{\text{target}}$ converges toward the optimal Q function, $Q^\star$. In our implementation, for each episode, the agent was allowed to sample the image for 20 steps. We set $\gamma = 0.99$, a frequently used value that, being close to 1, emphasizes current and next states but also includes those further in the future.

Training: Backpropagation of the Deep *Q* network

In each step of DQN backpropagation, we randomly selected a batch size of $N_{\text{batch}} = 128$ transitions from the rows of $\mathbb{T}$ and computed corresponding $Q^{(t)}_{\text{target}}$ and $Q^{(t)}_{\text{DQN}}$ values, yielding the vectors $\overrightarrow{Q}_{\text{target}} = \{Q^{(t)}_{\text{target}}\}_{t=1}^{N_{\text{batch}}}$ and $\overrightarrow{Q}_{\text{DQN}} = \{Q^{(t)}_{\text{DQN}}\}_{t=1}^{N_{\text{batch}}}$. We backpropagated to minimize the loss L_{batch} of said batch,

$$L_{\text{batch}} = \frac{1}{N_{\text{batch}}} \sum_{i=1}^{N_{\text{batch}}} \left| Q^{(i)}_{\text{target}} - Q^{(i)}_{\text{DQN}} \right|. \tag{7}$$

Fortunately, as we proceed in training $Q^{(t)}_{\text{DQN}}$ to successively approximate $Q^{(t)}_{\text{target}}$, our CNN function approximation should converge toward that reflecting the optimal policy, so that

$$\lim_{t\to\infty} \left(Q^{(t)}_{\text{DQN}} \right) = \lim_{t\to\infty} \left(Q^{(t)}_{\text{target}} \right) = Q^\star. \tag{8}$$

To train the DQN, we employed the Adam optimizer with learning rate of 1×10^{-4}. We implemented DQN training in the Pytorch package in Python 3.7 executed in Google Colab.

Results

We trained the DQN on 30 two-dimensional image slices from the BraTS database at the level of the lateral ventricles. We did not employ any data augmentation. Training was performed for 90 episodes. For each of the separate 30 testing set images, the trained *DQN* was applied to the initial state with agent in the top left corner and successively to each subsequent state for a total of 20 steps. Even if the agent overlaps the lesion before the 20$^{\text{th}}$ step, we would expect that, with adequate prior training, the agent would stay on the lesion, given the training incentive to do so, as shown in Fig. 1d. Upon testing, the agent does not have prior knowledge about where the lesion is, so we felt that taking 20 steps was adequate to reach any lesion given our 4×4 (16 patches) grid.

Figure 2 shows the two possible testing/deployment outcomes. Figure 2a displays a true positive (TP) outcome, in which the agent overlies a patch that has nonzero overlap with the lesion. Figure 2b shows a false positive (FP) missed testing set case, in which the agent has zero overlap with the lesion after 20 steps of deployment.

Because this is a localization task, we take both true and false negative to be zero. Hence our accuracy is defined as $\frac{TP}{TP+FP}$.

In order to compare the performance of RL / Deep Q learning with that of standard supervised deep learning, we trained a localization supervised deep learning network as well. More specifically, we trained a keypoint detection CNN with architecture essentially identical to that of our DQN. Again, to make the comparison as fair as possible, we trained the keypoint detection CNN on the same 30 training images for 90 epochs. TP is defined when the keypoint lies within the lesion, FP when it lies outside of the lesion. Not surprisingly for such a small training set, the supervised keypoint detection CNN quickly overfit the training set, with training and testing set losses diverging before the 10 th epoch.

Figure shows a head-to-head comparison of the two techniques. It plots accuracy of the trained networks on the separate testing sets of 30 images as a function of training time, measured as episodes for deep reinforcement learning and as epochs for supervised deep learning. Supervised deep learning does not learn in a way that generalizes to the testing set, as evidenced by the essentially zero slope of the best fit line during training. The DQN learns in a more generalized manner during training, as manifested by the positive slope of the best fit line. Ultimately, RL/Deep Q learning achieves an average accuracy of 70% over the last 20 episodes, whereas supervised deep learning has a corresponding mean accuracy of 11%, a difference that is statistically significant by standard t-test, with p-value of 5.9×10^{-43}.

We also note that if the training:testing split were more weighted toward training images, for example 55:5, we would expect the following: a testing set accuracy of 3/5 or 4/5 could very well be due to chance. In this case, for better statistics, one would perform 12-fold cross validation and take the average testing set accuracy, computing a confidence interval.

In our case, the training:testing split was 30:30. As such, it stands to reason that $21/30 = 0.7$ or 70% testing set accuracy is very unlikely due to chance. Nevertheless, we ran twofold cross validation, reversing training and testing set, and obtained a testing set accuracy again of 70%.

Discussion

We have successfully applied deep reinforcement learning, here implemented as Deep Q learning, in tandem with temporal difference learning. Specifically, in this demonstration, we have applied the approach to identifying and locating glioblastoma multiforme brain lesions on contrast-enhanced MRI. By locating, we mean more specifically locating at least one point within the lesion, noting that the lesion can have nonzero overlap with more than one patch in our gridworld image tiling.

We have shown that the approach can produce reasonably accurate results with a training set size of merely 30 images. This number is at least an order of magnitude below what is generally considered necessary for radiology AI. This follows from the fact that current radiology AI has been dominated by supervised deep learning, an approach that depends on large amounts of annotated data. Supervised deep learning typically requiring hundreds (or, preferably, thousands) of annotated images to achieve high performance.

To restate the three key limitations of the currently prevalent supervised deep learning approach in radiology, they are: (1) Requirement of large amounts of expert-annotated data; (2) Susceptibility to grossly incorrect predictions when applied to new data sets; and (3) Lack of insight or intuition into the algorithm.

This proof-of-principle work provides evidence that reinforcement learning can address limitation #1. It can also address limitation #3, as evidenced by the reward structure illustrated in Fig. 1b–d.

We note that deep reinforcement learning becomes very time-consuming to train for large training sets, making this less practical for our proof-of-principle work. We note further that the purpose of this work was to show the data efficiency of deep reinforcement learning, which would not be furthered by training it on a large database. Prior work has shown [16] that for 700 training set images, supervised deep learning can achieve high accuracy, of around 92%. Hence, the transition to accurate supervised deep learning regression on the BraTS database can be presumed to lie somewhere between 30 and 700 training set images.

Future work will address limitation #2 by comparing deep reinforcement learning and supervised deep learning trained on data from one institution and tested on separate images from another institution, ideally constituting a wide range of tumor characteristics.

An important limitation of the present work is that it has been performed on two-dimensional image slices. Future work will extend to fully three-dimensional image volumes. As can be seen in Fig. 4, the Deep Q learning training process is somewhat noisy. Future work will

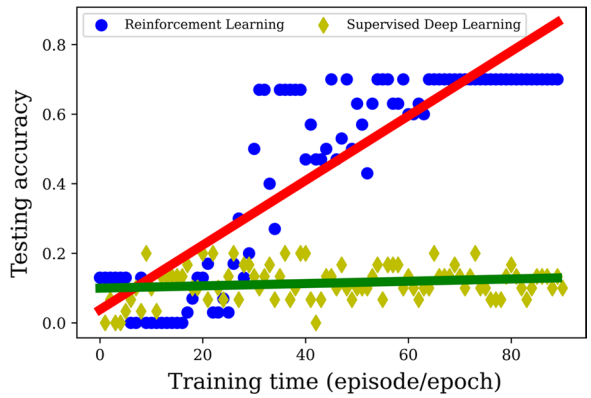

Fig. 4 Comparison between (deep) reinforcement learning/Deep Q learning and supervised deep learning. The mean accuracy of the testing set of 30 images is plotted as a function of training time, which is measured for reinforcement learning in episodes, and for supervised deep learning in epochs. Both methods were trained on the same 30 training set images. Blue circles represent the results of reinforcement/Deep Q learning, with best fit line in red. Accuracy for supervised deep learning is shown as yellow diamonds, with best fit line in green. Whereas supervised deep learning quickly overfits the small training set and thus cannot learn from it in any generalized manner, the Deep Q network is able to learn over time in a way that can generalize to the separate testing set

utilize different techniques in reinforcement learning to learn in a smoother fashion. It should be noted that we tried employing policy-gradient learning to achieve this less noisy learning. We did so with the actor-critic approach in both its single-agent version, A2C, and its multi-agent form, A3C. Both approaches failed to learn as Deep *Q* learning could. We suspect that this is caused by the sequential nature of sampling in A2C/A3C, which could not make use of the varied sampling across environments (i.e., different training set images) and states. We anticipate that incorporating a replay memory buffer with policy gradient may ultimately work best, and this will be a focus of future work.

Conlusions

We have shown as proof-of-principle that deep reinforcement learning can accurately localize brain lesions on MRI using a gridworld framework. High testing set accuracy is achieved despite a very small training set. Hence, deep reinforcement learning can provide a data-efficient method to localize lesions when limited image data is available.

Abbreviations

RL: Reinforcement learning; DRL: Deep reinforcement learning (used interchangeably with reinforcement learning); DQN: Deep Q Network; SDL: Supervised deep learning.

Acknowledgements

A conference proceeding presentation of this work has appeared in: https://www.asnr.org/wp-content/uploads/2021/07/ASNR21-Proceedings.pdf. A preprint including most of the work presented here has appeared in: https://arxiv.org/pdf/2010.10763.pdf

Author contributions

Both authors (JNS and HS) contributed extensively to concept generation, planning, data analysis, scripting/coding, writing and editing of the manuscript—every phase of the work. Both authors read and approved the final version of the manuscript.

Funding

We gratefully acknowledge support from the: 2020 Canon Medical Systems USA, Inc./ Radiology Society of North America (RSNA) Research Seed Grant #RSD2027. 2021 American Society of Neuroradiology (ASNR) Foundation Artificial Intelligence Grant. MSKCC Internal Radiology Artificial Intelligence Grant. The funding bodies played no role in the design of the study, collection, analysis, or interpretation of data or in writing the manuscript.

Availability of data and materials

The BraTS database can be accessed at: http://braintumorsegmentation.org/

Declarations

Ethics approval and consent to participate

Not applicable given no protected health information used.

Consent for publication

Not applicable.

Competing interest

We are seeking a patent based in part on the methods detailed here. The patent does not restrict research applications of the method or work presented in this paper. We have recently launched a radiology AI startup, Authera Inc, that will probably apply some of the methods described here.

Author details

[1]Department of Radiology, Memorial Sloan Kettering Cancer Center, 1275 York Avenue, Box 29, New York, NY 10065, USA. [2]Department of Aerospace Engineering, Indian Institute of Technology Madras, Chennai 600 036, India.

Received: 12 December 2021 Accepted: 22 October 2022
Published online: 23 December 2022

References

1. Alansary A, et al. Evaluating reinforcement learning agents for anatomical landmark detection. Med Image Anal. 2019;53:156–64.
2. Ghesu F-C, et al. Multi-scale deep reinforcement learning for real-time 3D-landmark detection in CT scans. IEEE Trans Pattern Anal Mach Intell. 2017;41:176–89.
3. Zhou SK, Le HN, Luu K, Nguyen HV, Ayache N. Deep reinforcement learning in medical imaging: a literature review. 2021; arXiv preprint, arXiv: 2103.05115.
4. Al WA, Yun ID. Partial policy-based reinforcement learning for anatomical landmark localization in 3d medical images. IEEE Trans Med Imaging. 2019;39:1245–55.
5. Maicas G, Carneiro G, Bradley AP, Nascimento JC, Reid I. Deep reinforcement learning for active breast lesion detection from DCE-MRI. In: International Conference on Medical Image Computing and Computer Assisted Intervention. 2017; 665–673.
6. Ali I, et al. Lung nodule detection via deep reinforcement learning. Front Oncol. 2018;8:108.
7. Jang Y, Jeon B. Deep reinforcement learning with explicit spatiosequential encoding network for coronary ostia identification in CT images. Sensors. 2021;21:6187.

8. Zhang P, Wang F, Zheng Y. Deep reinforcement learning for vessel centerline tracing in multi-modality 3D volumes. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. 2018, 755–763.
9. Stember J, Shalu H. Deep reinforcement learning to detect brain lesions on MRI: a proof-of-concept application of reinforcement learning to medical images. 2020; arXiv preprint arXiv:2008.02708.
10. Wang X et al. Inconsistent Performance of deep learning models on mammogram classification. J Am Coll Radiol; 2020.
11. Goodfellow IJ, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. 2014; arXiv preprint arXiv:1412.6572.
12. Buhrmester V, Münch D, Arens M. Analysis of explainers of black box deep neural networks for computer vision: a survey; 2019. arXiv preprint arXiv:1911.12116
13. Liu X, et al. A comparison of deep learning performance against healthcare professionals in detecting diseases from medical imaging: a systematic review and meta-analysis. Lancet Digit Health. 2019;1:e271–97.
14. Menze BH, et al. The multimodal brain tumor image segmentation benchmark (BRATS). IEEE Trans Med Imaging. 2014;34:1993–2024.
15. Sutton RS, Barto AG. Reinforcement learning: an introduction, MIT press; 2018.
16. Stember JN, et al. Integrating eye tracking and speech recognition accurately annotates MR brain images for deep learning: proof of principle. Radiol Artif Intell. 2021. https://doi.org/10.1148/ryai.2020200047.

Publisher's Note