Evaluating Reinforcement Learning Agents for Anatomical Landmark Detection

Amir Alansary, Ozan Oktay, Yuanwei Li, Loic Le Folgoc, Benjamin Hou, Ghislain Vaillant, Ben Glocker, Bernhard Kainz and Daniel Ruckert

Imperial College London, UK
a.alansary14@imperial.ac.uk
Reinforcement learning - Motivation

Mnih et al. 2015

Our agent for landmark detection
Unsupervised Learning

Explores data and draws inferences from datasets to describe hidden structures from unlabeled data.
Supervised Learning

Learning from a training set of labeled examples provided by a knowledgeable external supervisor.
Reinforcement Learning

Computational approach to learn by **interacting** with an environment

- Single decision must be made
  - Multiple actions
  - Each action has a reward associated with it

- Goal is to maximize reward
  - Pick an action with the highest reward
Reinforcement Learning

Sequential decision making

Agent

06/06/2018

Bernhard Kainz
Reinforcement Learning

Markov Decision Process (MDP)

- Set of states $S$
- Set of actions $A$
- Reward signal $R: s_t \times a_t \times s_{t+1} \rightarrow R$
- Transition function $T: s_t \times a_t \rightarrow s_{t+1} \equiv P(s_{t+1} \mid s_t, a_t)$

Markov assumption

- $s_t$ and $a_t$ are conditionally independent of all previous states and actions

Diagram:
- **Agent**
- **Environment**
- **Action $a_t$**
- **State $s_t$**
- **Reward $R_t$**
- **New State $s_{t+1}$**

Date: 06/06/2018

Bernhard Kainz
RL Main Elements

**Policy π**
- The agent’s strategy to choose an action at each state
- **Optimal Policy π** is the theoretical policy that maximizes the expectation of cumulative rewards

**Reward signal**
- Specifies what’s good and what’s bad in an immediate sense

**Value function**
- The total amount of reward an agent can expect to accumulate over the future
RL Solution

• Approximates iteratively the optimal value function when the whole MDP is unknown by sampling states and actions from the MDP, and learning from experience
  • Certainty equivalence
  • Temporal difference (TD)
  • State-action-reward-state-action (SARSA)
  • Q-learning
  • ...

Reinforcement learning
Learning what to do (how to map situations to action) -> so as to maximize sum of numerical rewards seen over the learner’s lifetime (Policy $\pi$: $S \rightarrow A$)
Value Functions

• A value function is defined as a prediction of the expected, accumulated, discounted, future reward in order to measure how good each state or state-action is

• **State-action value function**: Estimates a value of each action $a$ in each state $s$ under policy $\pi$

\[ Q^\pi(s, a) = E[R|s, a, \pi] \]

• Optimal policy $^*$ achieves the best expected return from any initial state

\[ Q^*(s, a) = \max_\pi Q^\pi(s, a) \]
Deep Q-Networks (DQN) \textsuperscript{Mnih 2013}

- DQN is an implementation of a standard Q-learning algorithm with function approximation using a ConvNet

\[ Q^\pi(s, a) \approx Q^\pi(s, a; \theta) \]

- Objective function: MSE in Q-values

\[ L(\theta) = E \left[ (r + \gamma \max_{a'} Q(s', a'; \theta) - Q(s, a; \theta))^2 \right] \]

- Optimize \textbf{end-to-end} by SGD, using \( \frac{\delta L(\theta)}{\delta \theta} \)
RL in Medical Imaging Analysis

**Image Segmentation**
- RL for image thresholding and segmentation
  - Sahba, F. et al. (2006)

**Image Localization**
- Deep RL for Active Breast Lesion Detection from DCE-MRI
  - Maicas, G. et al. (2017)

**Landmark Detection**
- Artificial agent for anatomical landmark detection in medical images
  - Ghesu, FC. et al. (2016, 2017)

**Image Registration**
- Artificial Agent for Robust Image Registration (rigid, non-rigid, 2D/3D)
  - Liao, R. et al. (2017)
  - Krebs J. et al. (2017)
  - Miao, S. et al. (2017)

**View Planning**
- Automatic view planning using deep RL agents
RL in Medical Imaging Analysis

Landmark Detection

- Artificial agent for anatomical landmark detection in medical images

Ghesu, FC. et al. (2016, 2017)
RL Agents for Landmark Detection

- Sequential decision process, where our RL-agent learns to navigate in an environment towards the target landmark using discrete action-steps

**States:**
3D region of interest (ROI) centered around the target landmark and current position

**Navigation actions:**
[left, right, up, down, forward, backward]
Terminal State

Training:
• Distance to the target landmark is ≤ 1mm

Testing:
1. Extra trigger action that terminates
   + Modifies the environment by marking the region centered around the correct target location
   - Increases the complexity of the task to be learned by increasing the action space size.
2. Oscillation property [1]
   + No added complexity to the action space
   - The correct target location is unmarked in the environment

• Here, we choose the terminating state based on the corresponding lower Q-value, when the agent oscillates
• Q-values are lower when the agent is closer to the target point and higher when it is far
• Intuitively, it encourages awarding higher Q-values to actions for far states from target

**Multi-scale Agent**

**Motivation**
Capture spatial relations within a global neighborhood

**Challenge**
Increasing the network’s field of view requires larger memory and higher computational complexity

**Solution**
+ **Multi-scale agent strategy** (coarse-to-fine fashion) [Ghesu et al 2017]
  - **Coarser levels** enables the agent to see more structural information
  - **Finer scales** provides more precise adjustments for the final estimation
+ **Hierarchical action steps**
  - **Larger steps** speed convergence towards the target plane
  - **Smaller steps** fine tune the final estimation of plane parameters
Proposed ConvNet Architecture

- Navigation actions are based on the estimated Q-values from the output of DQN.
Reward Function

• Designing good empirical reward functions $R$ is often difficult as RL agents can easily overfit the specified reward and thereby produce undesirable or unexpected results.

• $R$ should be proportional to the improvement that the agent makes to detect a landmark after selecting a particular action.

• We define the reward function,

\[ R = D(P_{i-1}, P_t) - D(P_i, P_t) \]

• $D$: Euclidean distance between two points.
• $P_i$: current position at step $i$
• $P_t$: target ground truth landmark’s location
We experimentally evaluate two recent state-of-the-art variants of the standard DQN

• **Double DQN (DDQN)** H. Van Hasselt 2015
  Removes upward bias caused by maximum approximated action value
  • Current Q-net $\theta$ is used to select actions
  • Older target Q-net $\theta^-$ is used to evaluate actions
  \[
  L(\theta) = E_{s,r,a,s' \sim D} \left[ (r + \gamma \max_{a'} Q(s', Q(s', a'; \theta), \theta^-) - Q(s, a; \theta))^2 \right]
  \]

• **Dueling DQN** Z. Wang 2015
  Split Q-net into two channels:
  • Action-independent value function $V(s)$
  • Action-dependent advantage function $A(s, a)$
  \[
  Q^\pi(s, a) = A^\pi(s, a) + V^\pi(s)
  \]
Experiment I - Fetal Head Ultrasound Landmarks

• Finding the target landmarks in fetal ultrasound images is a challenging task because of the shadowing, mirror images, refraction, and fetal motion

Dataset

• 72 fetal head ultrasound scans\(^1\) - 21 testing and 51 training
• Three landmarks:
  1. Right cerebellum (RC)
  2. Left cerebellum (LC)
  3. Cavum septum pellucidum (CSP)

\(^{[1]}\) http://www.ifindproject.com/
Comparison with state-of-the-art methods

- Comparison between different DQN–based agents and recent state-of-the-art methods for detecting the Cavum Septum Pellucidum (CSP) point from fetal ultrasound head scans.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Distance Error (mm)</td>
<td>7.37 ± 5.86</td>
<td>6.51 ± 5.41</td>
<td>5.47 ± 4.23</td>
<td>5.50 ± 2.79</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ours Fixed-scale</th>
<th>DQN</th>
<th>DDQN</th>
<th>Duel DQN</th>
<th>Duel DDQN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distance Error (mm)</td>
<td>4.95 ± 3.09</td>
<td>5.01 ± 2.84</td>
<td>6.29 ± 3.95</td>
<td>5.12 ± 3.15</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ours Multi-scale</th>
<th>DQN</th>
<th>DDQN</th>
<th>Duel DQN</th>
<th>Duel DDQN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distance Error (mm)</td>
<td><strong>3.66 ± 2.11</strong></td>
<td>4.02 ± 2.20</td>
<td>4.17 ± 2.62</td>
<td>4.02 ± 1.55</td>
</tr>
</tbody>
</table>

- Our agents outperforms state-of-the-art methods
The best performing agent varies for each landmark
Choosing the best DQN architecture is environment-dependent
Multi-scale agents do not improve significantly the performance upon fixed-scale in images with smaller field of view
Visualizations

Fixed-Scale

Left Cerebellar Duel DQN

Multi-Scale

Right Cerebellar DuelDoubleDQN

Multi-Scale

CSP DQN
Experiment II - Brain MRI

- **Anterior** and **posterior commissure** (AC and PC) commonly used by the neuroimaging community to define the axial plane during image acquisition

- **Dataset**
  - 832 isotropic 1mm MR scans from the ADNI database\(^1\)
    - 728 and 104 images for training and testing

\begin{table}
\begin{center}
\begin{tabular}{|c|c|c|c|c|}
\hline
Model      & Anterior Commissure & & Posterior Commissure &  \\
& FS & MS & FS & MS \\
\hline
DQN        & 3.04 ± 1.70 & 2.46 ± 1.44 & 2.03 ± 0.97 & 2.05 ± 1.14 \\
DDQN       & 2.62 ± 1.24 & 2.61 ± 1.64 & 3.31 ± 1.2 & 1.86 ± 1.07 \\
Duel DQN   & 3.04 ± 1.28 & 2.4 ± 1.42 & 3.6 ± 1.46 & 2.15 ± 1.24 \\
Duel DDQN  & 2.97 ± 1.23 & 2.01 ± 1.29 & 2.04 ± 1.04 & 2.27 ± 1.22 \\
\hline
\end{tabular}
\end{center}
\end{table}

Visualizations

Fixed-Scale

Multi-Scale

AC - DuelDoubleDQN

PC – Double DQN
Experiment III – Cardiac MRI

- **Apex** and **center of mitral valve**, commonly used for defining the short axis view during image acquisitions.

- Also used to assist automatic segmentation methods by defining starting and ending slices in the acquired cardiac stack of 2D image sequence.

- **Dataset**
  - 455 short-axis cardiac MR of resolution 1.25x1.25x2mm obtained from the UK Digital Heart Project [1]
    - 364 training and 91 testing

Results

- Duel DQN performs the best for detecting the apex
- Multi-scale agents significantly improve upon the fixed-scale agents, as the field of the view of cardiac scans is wider
- The performance of the agent improves with larger contextual information
Visualizations

Multi-Scale

Mitral DoubleDQN

Apex DuelDQN
Runtime

• The agent finds the target location using sequential steps

• Total runtime depends on the starting point – the further it is, the longer it will take to find the target landmark

• In our implementation, each step takes around 0.0005-0.001 seconds. For example, if the agent is far 1000 steps from the target, it will take 0.5-1 second to find the target… Very fast!
Current Challenges

• Background noise may confuse the agent for finding the accurate location of the target landmark

• No terminal state by following a long circular path around the target. This can be alleviated by using bigger memory to trace agent’s recent path and detect oscillations frequencies
Limitations

• Reinforcement learning is a difficult problem that needs a careful formulation of its elements

• For example, RL tends to overfit to the rewards, which may cause unexpected behaviors

• Our results show that the optimal algorithm for achieving the best performance depends on the target landmark (environment-dependent) – similarly on different Atari games
Conclusion

• Fast automatic RL-agents can achieve the state-of-the-art performance for detecting anatomical landmarks from ultrasound and MRI scans.

• Our extensive evaluations using several DQN based strategies show similar performance of all agents. However, multi-scale agents improve the performance in images with larger field of view such as cardiac MRI.

• Hierarchical action steps speeds up the performance with larger steps, and yet smaller steps fine tune the fine location precisely.
Future Work

• Investigate using intrinsic geometry instead of intensity patterns for the RL-environment to improve the performance using collaborative or competitive agents.

• Explore the use of either competitive or collaborative multi-agents to detect a single or multi-landmarks.

• Inspired by AlphaGo RL agents could mimic the moves of a human expert and accumulate this experience, thus learning from experienced operators during real-time observation.

• Another future direction, investigate involving human experts for learning the artificial agents actively, inspired by AlphaGo [D. Silver et al. 2016], where the agents can learn from experienced operators by interaction and accumulate this experience.
Code is publicly available

https://github.com/amiralansary/tensorpack-medical