10-703 - Homework 2: Playing Atari With Deep Reinforcement Learning

Rogerio Bonatti
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
rbonatti@andrew.cmu.edu

Ratnesh Madaan
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
ratneshm@andrew.cmu.edu

Abstract

In this assignment we implemented Q-learning using deep learning function approximators for the Space Invaders game in the OpenAI Gym environment. We implemented the following variations of Q-learning: linear network without and with experience replay and target fixing, linear double Q-network with experience replay and target fixing, and dueling deep Q-learning.

1 [5pts] Show that update 1 and update 2 are the same when the functions in Q are of the form $Q_w(s,a) = w^T\phi(s,a)$, with $w \in \mathbb{R}^{|S||A|}$ and $\phi : S \times A \rightarrow \mathbb{R}^{|S||A|}$, where the feature function ϕ is of the form $\phi(s,a)_{s',a'} = \mathbb{1}[s'=s, a'=a]$

Updates:

$$Q(s,a) := Q(s,a) + \alpha\left(r + \gamma \max_{a' \in A} Q(s',a') - Q(s,a)\right) \tag{1}$$

$$w := w + \alpha\left(r + \gamma \max_{a' \in A} Q(s',a') - Q(s,a)\right) \nabla_w Q_w(s,a) \tag{2}$$

Solution:

We begin with Eq 2, substituting the derivative with respect to w, given that $Q(s,a) = w^T\phi(s,a)$:

$$w := w + \alpha\left(r + \gamma \max_{a' \in A} Q(s',a') - Q(s,a)\right) \phi(s,a) \tag{3}$$

Now we transpose both sides of the equation, and multiply both sides by $\phi(s,a)$:

$$w^T\phi(s,a) := w^T\phi(s,a) + \alpha\left(r + \gamma \max_{a' \in A} Q(s',a') - Q(s,a)\right) \phi^T(s,a)\phi(s,a) \tag{4}$$

Now we can again use the fact that $Q(s,a) = w^T\phi(s,a)$:

$$Q(s,a) := Q(s,a) + \alpha\left(r + \gamma \max_{a' \in A} Q(s',a') - Q(s,a)\right) \phi^T(s,a)\phi(s,a) \tag{5}$$

10807 Class Project (2016).

Lastly, since $\phi(s,a)_{s',a'} = \mathbb{1}[s' = s, a' = a]$, the norm of the dot product will equal to 1, resulting in:

$$Q(s,a) := Q(s,a) + \alpha \left(r + \gamma \max_{a' \in A} Q(s',a') - Q(s,a) \right) \tag{6}$$

And this we proved that Eq 2 is the same as Eq 1.

2 [5pts] Implement a linear Q-network (no experience replay or target fixing). Use the experimental setup of [1, 2] to the extent possible

We implemented a linear Q-network, and to run the training process, one needs to run the command "python dqn.py –modes ".

We used the following hyper-parameters for this network:

- Discount factor $\gamma = 0.99$
- Learning rate $\alpha = 0.0001$
- Exploration probability $\epsilon = 0.05$, decreasing from 1 to 0.05 in a linear fashion during training process
- Number of iterations with environment: 5,000,000
- Number of frames to feed to the Q-network: 4
- Input image resizing: 84×84
- Steps between evaluations of network: 10,000
- Steps for "burn in" (random actions in the beginning of training process): 50,000
- Maximum episode length: 100,000 steps (basically we chose to allow any game size)

We plotted the performance plots of this network in Figs 2-2.

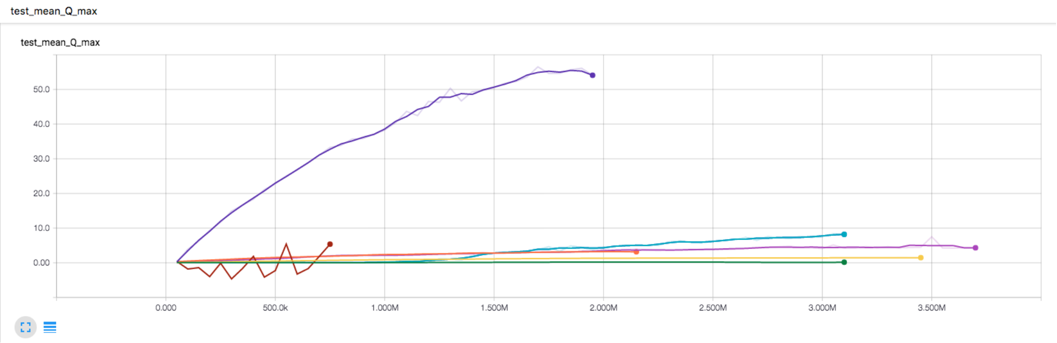

Figure 1: Mean Q per step plot for the case of linear network without target fixing and without experience replay

Using the *Monitor* wrapper of the gym environment, we generated videos of the behavior of the agent across different stages of training:

- 0/3 of training: Youtube video
- 1/3 of training: Youtube video
- 2/3 of training: Youtube video
- 3/3 of training: Youtube video

Here are also some comments about the behavior and training of this specific network:

- Bla
- Bla

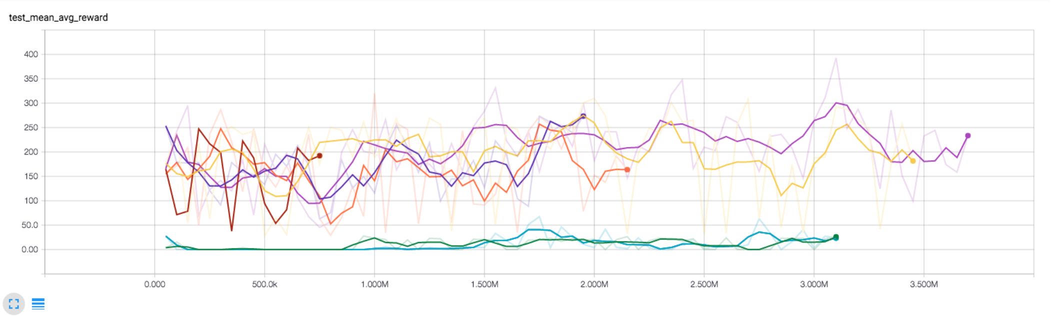

Figure 2: Mean reward per episode plot for the case of linear network without target fixing and without experience replay

3 [10pts] Implement a linear Q-network with experience replay and target fixing. Use the experimental setup of [1, 2] to the extent possible

We implemented a linear Q-network, and to run the training process, one needs to run the command “python dqn.py –modes ”.

We used the following hyper-parameters for this network:

- Discount factor $\gamma = 0.99$
- Learning rate $\alpha = 0.0001$
- Exploration probability $\epsilon = 0.05$, decreasing from 1 to 0.05 in a linear fashion during training process
- Number of iterations with environment: 5,000,000
- Number of frames to feed to the Q-network: 4
- Input image resizing: 84×84
- Replay buffer size: 1,000,000
- Target Q-network reset interval: 10,000
- Batch size: 32
- Steps between evaluations of network: 10,000
- Steps for “burn in” (random actions in the beginning of training process): 50,000
- Maximum episode length: 100,000 steps (basically we chose to allow any game size)

We plotted the performance plots of this network in Figs **??**-**??**.

Using the *Monitor* wrapper of the gym environment, we generated videos of the behavior of the agent across different stages of training:

- 0/3 of training: Youtube video
- 1/3 of training: Youtube video
- 2/3 of training: Youtube video
- 3/3 of training: Youtube video

Here are also some comments about the behavior and training of this specific network:

- Bla
- Bla

4 [5pts] Implement a linear double Q-network. Use the the experimental setup of [1, 2] to the extent possible.

We implemented a double linear Q-network, and to run the training process, one needs to run the command “python dqn.py –modes ”.

We used the following hyper-parameters for this network:

- Discount factor $\gamma = 0.99$
- Learning rate $\alpha = 0.0001$
- Exploration probability $\epsilon = 0.05$, decreasing from 1 to 0.05 in a linear fashion during training process
- Number of iterations with environment: 5,000,000
- Number of frames to feed to the Q-network: 4
- Input image resizing: 84×84
- Replay buffer size: 1,000,000
- Target Q-network reset interval: 10,000
- Batch size: 32
- Steps between evaluations of network: 10,000
- Steps for “burn in” (random actions in the beginning of training process): 50,000
- Maximum episode length: 100,000 steps (basically we chose to allow any game size)

We plotted the performance plots of this network in Figs **??**-**??**.

Using the *Monitor* wrapper of the gym environment, we generated videos of the behavior of the agent across different stages of training:

- 0/3 of training: Youtube video
- 1/3 of training: Youtube video
- 2/3 of training: Youtube video
- 3/3 of training: Youtube video

Here are also some comments about the behavior and training of this specific network:

- Bla
- Bla

5 [35pts] Implement the deep Q-network as described in [1, 2]

We implemented a deep Q-network. We tested the performance of this network with different games, and to run them, one can use the following commands:

- Space invaders: “python dqn.py –modes ”
- Enduro: “python dqn.py –modes ”
- Breakout: “python dqn.py –modes ”

We used the following hyper-parameters for this network:

- Discount factor $\gamma = 0.99$
- Learning rate $\alpha = 0.0001$
- Exploration probability $\epsilon = 0.05$, decreasing from 1 to 0.05 in a linear fashion during training process
- Number of iterations with environment: 5,000,000

- Number of frames to feed to the Q-network: 4
- Input image resizing: 84×84
- Replay buffer size: 1,000,000
- Target Q-network reset interval: 10,000
- Batch size: 32
- Steps between evaluations of network: 10,000
- Steps for “burn in” (random actions in the beginning of training process): 50,000
- Maximum episode length: 100,000 steps (basically we chose to allow any game size)

We plotted the performance plots of this network in for different games in Figs **??**-**??**.

Using the *Monitor* wrapper of the gym environment, we generated videos of the behavior of the agent across different stages of training, for the 3 games considered:

For Space invaders:

- 0/3 of training: Youtube video
- 1/3 of training: Youtube video
- 2/3 of training: Youtube video
- 3/3 of training: Youtube video

For Enduro:

- 0/3 of training: Youtube video
- 1/3 of training: Youtube video
- 2/3 of training: Youtube video
- 3/3 of training: Youtube video

For Breakout:

- 0/3 of training: Youtube video
- 1/3 of training: Youtube video
- 2/3 of training: Youtube video
- 3/3 of training: Youtube video

Here are also some comments about the behavior and training of this specific network:

- Bla
- Bla

6 [20pts] Implement the double deep Q-network as described in [3]

We implemented a double deep Q-network, and to run the training process, one needs to run the command “python dqn.py –modes ”.

We used the following hyper-parameters for this network:

- Discount factor $\gamma = 0.99$
- Learning rate $\alpha = 0.0001$
- Exploration probability $\epsilon = 0.05$, decreasing from 1 to 0.05 in a linear fashion during training process
- Number of iterations with environment: 5,000,000
- Number of frames to feed to the Q-network: 4

- Input image resizing: 84×84
- Replay buffer size: 1,000,000
- Target Q-network reset interval: 10,000
- Batch size: 32
- Steps between evaluations of network: 10,000
- Steps for "burn in" (random actions in the beginning of training process): 50,000
- Maximum episode length: 100,000 steps (basically we chose to allow any game size)

We plotted the performance plots of this network for Space Invaders in Figs **??**-**??**.

Using the *Monitor* wrapper of the gym environment, we generated videos of the behavior of the agent across different stages of training:

- 0/3 of training: Youtube video
- 1/3 of training: Youtube video
- 2/3 of training: Youtube video
- 3/3 of training: Youtube video

Here are also some comments about the behavior and training of this specific network:

- Bla
- Bla

7 [20pts] Implement the dueling deep Q-network as described in [4]

We implemented a dueling deep Q-network, and to run the training process, one needs to run the command "python dqn.py –modes ".

We used the following hyper-parameters for this network:

- Discount factor $\gamma = 0.99$
- Learning rate $\alpha = 0.0001$
- Exploration probability $\epsilon = 0.05$, decreasing from 1 to 0.05 in a linear fashion during training process
- Number of iterations with environment: 5,000,000
- Number of frames to feed to the Q-network: 4
- Input image resizing: 84×84
- Replay buffer size: 1,000,000
- Target Q-network reset interval: 10,000
- Batch size: 32
- Steps between evaluations of network: 10,000
- Steps for "burn in" (random actions in the beginning of training process): 50,000
- Maximum episode length: 100,000 steps (basically we chose to allow any game size)

We plotted the performance plots of this network in Figs **??**-**??**.

Using the *Monitor* wrapper of the gym environment, we generated videos of the behavior of the agent across different stages of training:

- 0/3 of training: Youtube video
- 1/3 of training: Youtube video
- 2/3 of training: Youtube video

- 3/3 of training: Youtube video

Here are also some comments about the behavior and training of this specific network:

- Bla
- Bla

8 Table comparing rewards for each fully trained model

We constructed a table comparing the average total reward found in 100 episodes for each fully trained model we implemented:

Table 1: Avg reward per episode for 100 episodes in implemented networks

Model	Game	Avg Reward 100 episodes
Linear, no target fix, no exp replay	Space Invaders	50 ± 5
Linear, with target fix, with exp replay	Space Invaders	50 ± 5
Double Linear	Space Invaders	50 ± 5
Deep Q	Space Invaders	50 ± 5
Deep Q	Enduro	50 ± 5
Deep Q	Breakout	50 ± 5
Double Deep Q	Space Invaders	50 ± 5
Dueling Deep Q	Space Invaders	50 ± 5

Here are some comments about the results in the table:

- Bla
- Bla

8.1 Style

Papers to be submitted to NIPS 2016 must be prepared according to the instructions presented here. Papers may only be up to eight pages long, including figures. Since 2009 an additional ninth page *containing only acknowledgments and/or cited references* is allowed. Papers that exceed nine pages will not be reviewed, or in any other way considered for presentation at the conference.

The margins in 2016 are the same as since 2007, which allow for $\sim 15\%$ more words in the paper compared to earlier years.

Authors are required to use the NIPS LaTeX style files obtainable at the NIPS website as indicated below. Please make sure you use the current files and not previous versions. Tweaking the style files may be grounds for rejection.

8.2 Retrieval of style files

The style files for NIPS and other conference information are available on the World Wide Web at

`http://www.nips.cc/`

The file `nips_2016.pdf` contains these instructions and illustrates the various formatting requirements your NIPS paper must satisfy.

The only supported style file for NIPS 2016 is `nips_2016.sty`, rewritten for LaTeX 2_{ε}. **Previous style files for LaTeX 2.09, Microsoft Word, and RTF are no longer supported!**

The new LaTeX style file contains two optional arguments: `final`, which creates a camera-ready copy, and `nonatbib`, which will not load the `natbib` package for you in case of package clash.

At submission time, please omit the `final` option. This will anonymize your submission and add line numbers to aid review. Please do *not* refer to these line numbers in your paper as they will be removed during generation of camera-ready copies.

The file `nips_2016.tex` may be used as a "shell" for writing your paper. All you have to do is replace the author, title, abstract, and text of the paper with your own.

The formatting instructions contained in these style files are summarized in Sections 9, 10, and 11 below.

9 General formatting instructions

The text must be confined within a rectangle 5.5 inches (33 picas) wide and 9 inches (54 picas) long. The left margin is 1.5 inch (9 picas). Use 10 point type with a vertical spacing (leading) of 11 points. Times New Roman is the preferred typeface throughout, and will be selected for you by default. Paragraphs are separated by ½ line space (5.5 points), with no indentation.

The paper title should be 17 point, initial caps/lower case, bold, centered between two horizontal rules. The top rule should be 4 points thick and the bottom rule should be 1 point thick. Allow ¼ inch space above and below the title to rules. All pages should start at 1 inch (6 picas) from the top of the page.

For the final version, authors' names are set in boldface, and each name is centered above the corresponding address. The lead author's name is to be listed first (left-most), and the co-authors' names (if different address) are set to follow. If there is only one co-author, list both author and co-author side by side.

Please pay special attention to the instructions in Section 11 regarding figures, tables, acknowledgments, and references.

10 Headings: first level

All headings should be lower case (except for first word and proper nouns), flush left, and bold.

First-level headings should be in 12-point type.

10.1 Headings: second level

Second-level headings should be in 10-point type.

10.1.1 Headings: third level

Third-level headings should be in 10-point type.

Paragraphs There is also a `\paragraph` command available, which sets the heading in bold, flush left, and inline with the text, with the heading followed by 1 em of space.

11 Citations, figures, tables, references

These instructions apply to everyone.

11.1 Citations within the text

The `natbib` package will be loaded for you by default. Citations may be author/year or numeric, as long as you maintain internal consistency. As to the format of the references themselves, any style is acceptable as long as it is used consistently.

The documentation for `natbib` may be found at

```
http://mirrors.ctan.org/macros/latex/contrib/natbib/natnotes.pdf
```

Of note is the command `\citet`, which produces citations appropriate for use in inline text. For example,

```
\citet{hasselmo} investigated\dots
```

produces

Hasselmo, et al. (1995) investigated. . .

If you wish to load the `natbib` package with options, you may add the following before loading the `nips_2016` package:

```
\PassOptionsToPackage{options}{natbib}
```

If `natbib` clashes with another package you load, you can add the optional argument `nonatbib` when loading the style file:

```
\usepackage[nonatbib]{nips_2016}
```

As submission is double blind, refer to your own published work in the third person. That is, use "In the previous work of Jones et al. [4]," not "In our previous work [4]." If you cite your other papers that are not widely available (e.g., a journal paper under review), use anonymous author names in the citation, e.g., an author of the form "A. Anonymous."

11.2 Footnotes

Footnotes should be used sparingly. If you do require a footnote, indicate footnotes with a number[1] in the text. Place the footnotes at the bottom of the page on which they appear. Precede the footnote with a horizontal rule of 2 inches (12 picas).

Note that footnotes are properly typeset *after* punctuation marks.[2]

11.3 Figures

All artwork must be neat, clean, and legible. Lines should be dark enough for purposes of reproduction. The figure number and caption always appear after the figure. Place one line space before the figure caption and one line space after the figure. The figure caption should be lower case (except for first word and proper nouns); figures are numbered consecutively.

You may use color figures. However, it is best for the figure captions and the paper body to be legible if the paper is printed in either black/white or in color.

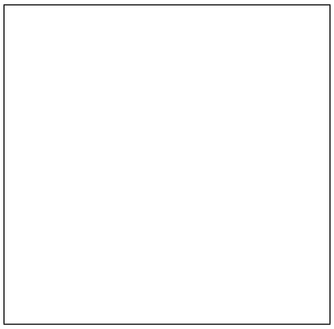

Figure 3: Sample figure caption.

11.4 Tables

All tables must be centered, neat, clean and legible. The table number and title always appear before the table. See Table 2.

Place one line space before the table title, one line space after the table title, and one line space after the table. The table title must be lower case (except for first word and proper nouns); tables are numbered consecutively.

[1] Sample of the first footnote.

[2] As in this example.

Table 2: Sample table title

Part		
Name	Description	Size (μm)
Dendrite	Input terminal	$\sim$100
Axon	Output terminal	$\sim$10
Soma	Cell body	up to 10^6

Note that publication-quality tables *do not contain vertical rules.* We strongly suggest the use of the `booktabs` package, which allows for typesetting high-quality, professional tables:

`https://www.ctan.org/pkg/booktabs`

This package was used to typeset Table 2.

12 Final instructions

Do not change any aspects of the formatting parameters in the style files. In particular, do not modify the width or length of the rectangle the text should fit into, and do not change font sizes (except perhaps in the **References** section; see below). Please note that pages should be numbered.

13 Preparing PDF files

Please prepare submission files with paper size “US Letter,” and not, for example, “A4.”

Fonts were the main cause of problems in the past years. Your PDF file must only contain Type 1 or Embedded TrueType fonts. Here are a few instructions to achieve this.

- You should directly generate PDF files using `pdflatex`.
- You can check which fonts a PDF files uses. In Acrobat Reader, select the menu Files>Document Properties>Fonts and select Show All Fonts. You can also use the program `pdffonts` which comes with `xpdf` and is available out-of-the-box on most Linux machines.
- The IEEE has recommendations for generating PDF files whose fonts are also acceptable for NIPS. Please see `http://www.emfield.org/icuwb2010/downloads/IEEE-PDF-SpecV32.pdf`
- `xfig` "patterned" shapes are implemented with bitmap fonts. Use "solid" shapes instead.
- The `\bbold` package almost always uses bitmap fonts. You should use the equivalent AMS Fonts:

  ```
  \usepackage{amsfonts}
  ```

  followed by, e.g., `\mathbb{R}`, `\mathbb{N}`, or `\mathbb{C}` for $\mathbb{R}$, $\mathbb{N}$ or $\mathbb{C}$. You can also use the following workaround for reals, natural and complex:

  ```
  \newcommand{\RR}{I\!\!R} %real numbers
  \newcommand{\Nat}{I\!\!N} %natural numbers
  \newcommand{\CC}{I\!\!\!\!C} %complex numbers
  ```

  Note that `amsfonts` is automatically loaded by the `amssymb` package.

If your file contains type 3 fonts or non embedded TrueType fonts, we will ask you to fix it.

13.1 Margins in LaTeX

Most of the margin problems come from figures positioned by hand using `\special` or other commands. We suggest using the command `\includegraphics` from the `graphicx` package. Always specify the figure width as a multiple of the line width as in the example below:

```
\usepackage[pdftex]{graphicx} ...
\includegraphics[width=0.8\linewidth]{myfile.pdf}
```

See Section 4.4 in the graphics bundle documentation (`http://mirrors.ctan.org/macros/latex/required/graphics/grfguide.pdf`)

A number of width problems arise when LaTeX cannot properly hyphenate a line. Please give LaTeX hyphenation hints using the `\-` command when necessary.

Acknowledgments

Use unnumbered third level headings for the acknowledgments. All acknowledgments go at the end of the paper. Do not include acknowledgments in the anonymized submission, only in the final paper.

References

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

[2] 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):529–533, 2015.

[3] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *AAAI*, pages 2094–2100, 2016.

[4] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado van Hasselt, Marc Lanctot, and Nando de Freitas. Dueling network architectures for deep reinforcement learning. *arXiv preprint arXiv:1511.06581*, 2015.