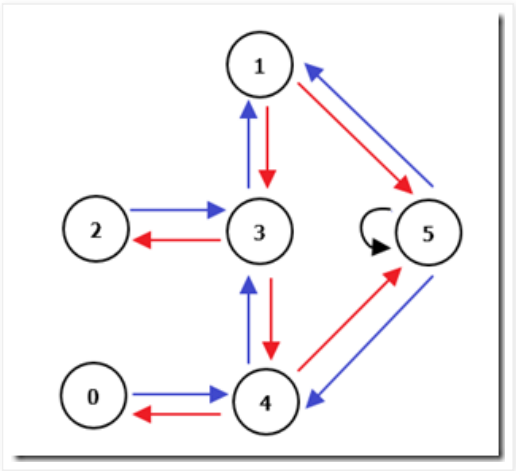
The Beginner Programmer

Shared thoughts, experiments, simulations and simple ideas with Python, R and other languages

Monday, 12 September 2016

Getting AI smarter with Q-learning: a simple first step in Python

Yesterday I found an "old" script I wrote during a morning in the last semester. I remember being a little bored and interested in the concept of **Q-learning**. That was about the time **Alpha-Go** had beaten the world champion of Go and by reading here and there I found out that a bit of Q-learning mixed with deep learning might have been involved.



Indeed Q-learning seems an interesting concept, perhaps even more fascinating than traditional supervised machine learning since in this case the machine is basically learning from scratch how to perform a certain task in oder to optimize future rewards.

If you would like to read a, quote, "Painless Q-learning tutorial", I suggest you to read the following explanation: A Painless Q-learning Tutorial. In this article the concept of Q-learning is explained through a simple example and a clear walk-through. After having read the article I decided to put into code the example shown. The example shows a maze through which the agent should go and find its way up to the goal stage (stage 5). Basically, the idea is to train an algorithm to find the optimal path, given an (often random) initial condition, in order to maximize a certain outcome. In this simple example, as you can see from the picture shown in the article, the possible choices are all known and the outcome of each choice is deterministic. A best path exists and can be found easily regardless of the initial condition. Furthemore, the maze is time invariant. These nice theoretical hypothesis are usually not true when dealing when real world problems and this makes using Q-learning hard in practice, even though the concept behind it is relatively simple.

Before taking a look at the code, I suggest to read the article mentioned above, where you will get familiar with the problem tackled below. I'm not going deep in explaining what is going on since the author of that article has already done a pretty good job doing it and my work is just a (probably horribly inefficient) translation in Python of the algorithm.

Given an initial condition of, say, state 2, the optimal sequence path is clearly 2 - 3 – 1 - 5. Let's see if the algorithm finds it!

The Beginner Programmer

Blog Archive

Labels

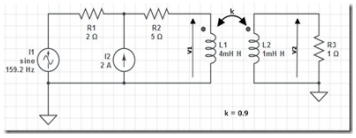
python (67) statistics (47) maths (38) physics (35) R (34) Machine Learning (19) Engineering (18) Economics and Finance (17) Electrics and electronics stuff (13) matlab (9) java (7) projects (7) sidenotes (7) Arduino (4) Hello world (3) Markov chains (3) database (2) deutsch (1)

Featured post

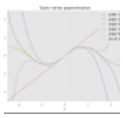
Solving a circuit with a mutual inductor using LTspice

```
import numpy as np

# R matrix
R = np.matrix([ [-1,-1,-1,-1,0,-1],
                [-1,-1,-1,0,-1,100],
                [-1,-1,-1,0,-1,-1],
                [-1,0,0,-1,0,-1],
                [-1,0,0,-1,-1,100],
                [-1,0,-1,-1,0,100] ])

# Q matrix
Q = np.matrix(np.zeros([6,6]))

# Gamma (learning parameter).
gamma = 0.8

# Initial state. (Usually to be chosen at random)
initial_state = 1

# This function returns all available actions in the state given as an argume
def available_actions(state):
    current_state_row = R[state,]
    av_act = np.where(current_state_row >= 0)[1]
    return av_act

# Get available actions in the current state
available_act = available_actions(initial_state)

# This function chooses at random which action to be performed within the ran
# of all the available actions.
def sample_next_action(available_actions_range):
    next_action = int(np.random.choice(available_act,1))
    return next_action

# Sample next action to be performed
action = sample_next_action(available_act)

# This function updates the Q matrix according to the path selected and the Q
# learning algorithm
def update(current_state, action, gamma):

    max_index = np.where(Q[action,] == np.max(Q[action,]))[1]

    if max_index.shape[0] > 1:
        max_index = int(np.random.choice(max_index, size = 1))
    else:
        max_index = int(max_index)
    max_value = Q[action, max_index]

    # Q learning formula
    Q[current_state, action] = R[current_state, action] + gamma * max_value

# Update Q matrix
update(initial_state,action,gamma)

#--------------------------------------------------------------------------
# Training

# Train over 10 000 iterations. (Re-iterate the process above).
for i in range(10000):
    current_state = np.random.randint(0, int(Q.shape[0]))
    available_act = available_actions(current_state)
```

Mutual inductors can be a lot of fun, and sometimes a bit of an headache if you mess something up or represent them in a complicated way. Ta...

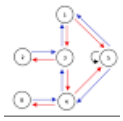


Popular Posts



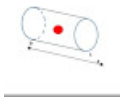
Taylor series with Python and Sympy

Here I am again using my beloved Python and doing maths stuff. Today I'd like to post a short piece of code I made after a review of Taylor...



Getting AI smarter with Q-learning: a simple first step in Python

Yesterday I found an "old" script I wrote during a morning in the last semester. I remember being a little bored and interested in the conce...



The Heat Equation: a Python implementation

By making some assumptions, I am going to simulate the flow of heat through an ideal rod. Suppose you have a cylindrical rod whose ends are ...

Follow by Email

Email address... Submit

Subscribe To

Posts

Comments

Recommended blogs

Planet Python
Codementor: Python implementation of Normalized Google Distance (Simple web scraping example)

All About Statistics
futher attacks on reproductive rights

DataScience+
Monte Carlo Simulation and Statistical Probability Distributions in Python

R-bloggers
Running UMAP for data visualisation in R

MilanoR
10th MilanoR Meeting: Information Retrieval and Price Positioning

```
    action = sample_next_action(available_act)
    update(current_state,action,gamma)

# Normalize the "trained" Q matrix
print("Trained Q matrix:")
print(Q/np.max(Q)*100)

#--------------------------------------------------------------------------
# Testing

# Goal state = 5
# Best sequence path starting from 2 -> 2, 3, 1, 5

current_state = 2
steps = [current_state]

while current_state != 5:

    next_step_index = np.where(Q[current_state,] == np.max(Q[current_state,]))

    if next_step_index.shape[0] > 1:
        next_step_index = int(np.random.choice(next_step_index, size = 1))
    else:
        next_step_index = int(next_step_index)

    steps.append(next_step_index)
    current_state = next_step_index

# Print selected sequence of steps
print("Selected path:")
print(steps)

#--------------------------------------------------------------------------
#                            OUTPUT
#--------------------------------------------------------------------------
#
# Trained Q matrix:
#[[   0.     0.     0.     0.    80.     0. ]
# [   0.     0.     0.    64.     0.   100. ]
# [   0.     0.     0.    64.     0.     0. ]
# [   0.    80.    51.2    0.    80.     0. ]
# [   0.    80.    51.2    0.     0.   100. ]
# [   0.    80.     0.     0.    80.   100. ]]
#
# Selected path:
# [2, 3, 1, 5]
#
```

q-learning.py hosted with ♡ by GitHub view raw

And sure enough there it is! Bear in mind this is a very simplified problem. At the same time though, keep in mind that Alpha Go is powered in part by a similar concept. I wonder what other application will come from Q-learning.

The concept of Q-learning is more vast than what I showed here, nevertheless I hope my post was an interesting insight.

Posted by Mic

Labels: Machine Learning, python

28 comments:

Vishwas Pai 27 June 2017 at 09:10

Hey, I am getting error in Q learning formula.The error is " too many indices for array".How to solve that?

Reply

Replies

Mic 30 June 2017 at 21:26

Hi, there seems to be an error in the indexing process, try to start debugging there.

Reply

بنت النور12 August 2017 at 16:08

In my example I have 50000 states .... and I got memory error.

Reply

Replies

Mic 13 September 2017 at 19:27

mmm.. not sure I understand. Sorry.

Reply

Manuel Amunategui 11 September 2017 at 19:34

Thanks for the nice python implementation - works great on new graphs. One minor nitpick, your graph image doesn't correspond to your reward matrix, it shouldn't go from 3 to 5 and back, instead 3 only goes to points 4 and 1. 5 loops onto itself for 100 points. Thanks for posting this!

Manuel

Reply

Replies

Mic 13 September 2017 at 19:35

Hi, thanks for reading! Oh, I didn't notice that! I didn't want to take the picture from the original article so I made one myself in a hurry. Thanks for pointing out the mistake, fixed it! :)

Reply

Nina 30 December 2017 at 10:36

Hi Mic, can you tell me why the next action is chosen at random (lines 29-33). I thought in Q learning the next action is chosen based on the highest Q value. Best, Nina

Reply

Replies

Mic 30 December 2017 at 11:52

Hi Nina, the next action is chosen at random only during the training of the agent in order to explore the environment (and build the Q matrix). In testing (lines 71->89) the next action is chosen according to the highest Q value as you expect.

Reply

Unknown 5 January 2018 at 15:34

In the update function, why are we selecting the Q max index based on the randomly selected action going into what I thought was the state field of Q? Such that Q(state, action) returns the Q-value for that specific pair, why are we inserting action into state field? (Q[action,])? Shouldn't we want to insert state there to get index from that?

Reply

emetss 28 January 2018 at 14:24

It's a fantastic example for designing MDP based algorithms for many randomly generated environments.
May I ask a question? Do you have like this example for multi agents. like team-q (friend-q) learning for cooperative mission

Reply

Replies

Mic 29 January 2018 at 12:37

Thanks! I'm sorry, as of now I haven't got any examples for multi agents.

Soumaila Fomba 9 February 2018 at 00:32

Bonjour,
comment adapter un algorithme de Q learning aux jeux de shogi ou échecs?

Reply

Louis Bazireau 25 April 2018 at 14:30

Hi ! Great articles combined with the painless Q-learning tutorial ! I'm interested to develop a reinforcement learning algorithm for a flash game, if i'm correct, you don't create any walls or rooms, your agent are just wandering around. So i just need to get data from the flash game and change the python algorithm ?

Reply

Replies

Mic 29 April 2018 at 11:16

Hi, as long as you can build the R matrix you can adapt this example by simply using your R matrix. However I think it is going to be a bit slow with larger R matrices.

Reply

Chathurangi Shyalika 17 October 2018 at 05:48

Hi,
Thanks for this great article!. Btw how many layers are in the NN you have implemented here? Can we specify the number of hidden layers in the network in reinforcement learning?

Reply

resma k 1 December 2018 at 07:15

I like your blog, I read this blog please update more content on python, further check it once at python online training

Reply

MAK 27 January 2019 at 05:31

def sample_next_action(available_actions_range):
... next_action = int(np.random.choice(available_act,1))
File "", line 2
next_action = int(np.random.choice(available_act,1))

^
IndentationError: expected an indented block

any idea why?

Reply

Shailendra 14 March 2019 at 12:32

Good Post. I like your blog. Thanks for Sharing!
Machine Learning with Python Training in Gurgaon

Reply

manishagaur 22 March 2019 at 13:01

Thanks a lot for sharing marvellous information on sap course. Thanks for sharing this valuable post with us.
Python Training in Gurgaon

Reply

Manisha singh 1 April 2019 at 13:45

Thank you for sharing such great information very useful to us.
Python Training institute in Noida

Reply

Pankaj Singh 5 April 2019 at 05:59

Thankful to you for this amazing information sharing with us. Get website designing and development services by Ogen Infosystem.
Website Designing Company in Delhi

Reply

Neha Sharma 13 April 2019 at 05:12

Rice Bags Manufacturers
Pouch Manufacturers
wall putty bag manufacturers
Lyrics with music

Reply

Neha Sharma 13 April 2019 at 05:13

we have provide the best ppc service.
ppc agency in gurgaon
website designing company in Gurgaon
PPC company in Noida
seo services in gurgaon

Reply

Neha Sharma 13 April 2019 at 05:13

we have provide the best fridge repair service.
fridge repair in faridabad
Videocon Fridge Repair in Faridabad
Whirlpool Fridge Repair in Faridabad
Washing Machine Repair in Noida
godrej washing machine repair in noida
whirlpool Washing Machine Repair in Noida
IFB washing Machine Repair in Noida
LG Washing Machine Repair in Noida

Reply

Neha Sharma 13 April 2019 at 05:13

Bali Honeymoon Packages From Delhi
Bali Honeymoon Packages From Chennai
Hong Kong Packages From Delhi
Europe Packages from Delhi
Bali Honeymoon Packages From Bangalore
Bali Honeymoon Packages From Mumbai
Maldives Honeymoon Packages From Bangalore
travel company in Delhi

Reply

Mutual Fundwala 16 April 2019 at 08:23

Your content is really awesome and understandable, thanks for the efforts in this blog. Visit Mutual Fund Wala for Mutual Fund Schemes.
Mutual Fund Companies

Reply

Kala Kutir 8 May 2019 at 08:52

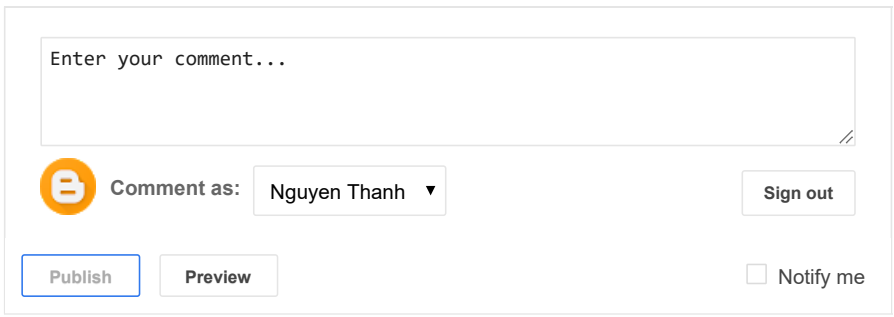
Decent, Get Service for Night out page 3 parties and this magnificent service provided by Lifestyle Magazine.
Lifestyle Magazine India

Reply

ajish 25 May 2019 at 06:44

Good Post! Thank you so much for sharing this pretty post, it was so good to read and useful to improve my knowledge as updated one, keep blogging
Python Training in electronic city

Reply



Newer Post Home Older Post

Subscribe to: Post Comments (Atom)