



## 码农的专栏

[目录视图](#)[摘要视图](#)[RSS 订阅](#)

### 个人资料



代码学习者



访问：2085056次

积分：28701

等级：[BLOG > 7](#)

排名：第183名

原创：80篇 转载：463篇

译文：0篇 评论：23条

### 文章搜索

[赠书 | AI专栏 \(AI圣经！《深度学习》中文版\)](#) [评论送书 | 机器学习、Java虚拟机、微信开发](#)

## DQN从入门到放弃5 深度解读DQN算法

2017-03-30 23:57

7283人阅读

[评论\(0\)](#)[分类：](#) [RL \(33\)](#)[目录\(?\)](#)[\[+\]](#)

### 0 前言

如果说DQN从入门到放弃的前四篇是开胃菜的话，那么本篇文章就是主菜了。所以，等吃完主菜再**放弃**吧！

[关闭](#)

### 1 详解Q-Learning

在上一篇文章[DQN从入门到放弃 第四篇](#)中，我们分析了动态规划Dynamic Programming **算法**。可能一些知友不是特别理解。那么这里我们再用简单的语言描

为了得到最优策略Policy，我们考虑估算每一个状态下每一种选择的价值Value。一个时间片的 $Q(s,a)$ 和当前得到的Reward以及下一个时间片的 $Q(s,a)$ 有关。有只能知道当前的Q值，怎么知道下一个时刻的Q值呢？大家要记住这一点，



实验。这意味着可以把上一次实验计算得到的Q值拿来使用呀。这样，不就可以根据当前的Reward及上一次实验中下一个时间片的Q值更新当前的Q值了吗？说起来真是很拗口。下面用比较形象的方法再具体分析一下Q-Learning。

Q-Learning的算法如下：



对于Q-Learning，首先就是要确定如何存储Q值，最简单的想法就是用矩阵，一个s一个a对应一个Q值，所以可以把Q值想象为一个很大的表格，横列代表s，纵列代表a，里面的数字代表Q值，如下表示：



这样大家就很清楚Q值是怎样的了。接下来就是看如何反复实验更新。

Step 1：初始化Q矩阵，比如都设置为0

Step 2：开始实验。根据当前Q矩阵及  $\epsilon - greedy$  方法获取动作。比如当前处在状态s1，那么在s1一列4  
都是0，那么这个时候随便选择都可以。



假设我们选择a2动作，然后得到的reward是1，并且进入到s3状态，接下来我

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(R_{t+1} + \lambda \max_a Q(S_{t+1}, a) - Q(S_t, A_t))$$

来更新Q值，这里我们假设  $\alpha$  是1， $\lambda$  也等于1，也就是每一次都把目标Q值赋

$$Q(S_t, A_t) = R_{t+1} + \max_a Q(S_{t+1}, a)$$

所以在这里，就是

## 文章分类

TX1 (6)

ELK (5)

ubuntu (43)

ffmpeg (1)

hbase (1)

torch (8)

windows (7)

DL (106)

opencv (8)

ML (15)

NLP (39)

image (2)

centos (1)

face (10)

GAN (9)

RL (34)

iOS (5)

python (15)

lua (2)

caffe (4)

C++ (2)

ROS (3)

algorithm (2)

pytorch (34)

paper reading (76)

tensorflow (2)

automatic driving (3)

关闭



FPGA (1)  
RNN (5)  
leetcode\_python (9)  
android (2)  
hadoop (2)  
mahout (2)  
openstack (1)  
oracle (1)  
security (4)  
SLAM (17)  
cocos2D-x (1)  
caption (14)  
tracking (5)  
vSLAM (10)  
IOT (2)  
CV (3)  
speech (20)  
object detection (3)  
VAE (2)  
Adreno (2)  
OpenCL (1)  
Snapdragon (6)  
Raspberry Pi (10)  
robot (1)  
system (0)  
segmentation (1)  
linux (2)  
Financial (1)  
VQA (4)  
cuda (1)  
Blockchain (13)  
video (6)

$$Q(s_1, a_2) = 1 + \max_a Q(s_3, a)$$

那么对应的s3状态，最大值是0，所以  $Q(s_1, a_2) = 1 + 0 = 1$ , Q表格就变成：



Step 3：接下来就是进入下一次动作，这次的状态是s3，假设选择动作a3，然后得到1的reward，状态变成s1，那么我们同样进行更新：

$$Q(s_3, a_3) = 2 + \max_a Q(s_1, a) = 2 + 1 = 3$$

所以Q的表格就变成：



Step 4：反复上面的方法。

就是这样，Q值在试验的同时反复更新。直到收敛。

相信这次知友们可以很清楚Q-Learning的方法了。接下来，我们将Q-Learning

关闭

## 2 维度灾难

在上面的简单分析中，我们使用表格来表示Q(s,a)，但是这个在现实的很多问题是太多。使用表格的方式根本存不下。

举Atari为例子。



[wordpress](#) (0)[latex](#) (1)[boost](#) (1)

### 文章存档

[2017年07月](#) (11)[2017年06月](#) (73)[2017年05月](#) (54)[2017年04月](#) (237)[2017年03月](#) (77)

展开

### 阅读排行

[linux pytorch 安装](#) (9473)[SLAM算法解析：抓住视](#) (8354)[PyTorch中文文档](#) (8229)[GAN学习指南：从原理入](#) (8104)[CNN浅析和历年ImageNet](#) (8073)[MTCNN训练整理](#) (8047)[带你搞懂朴素贝叶斯分类](#) (8043)[Adversarial Nets Papers](#) (7965)[elasticsearch 5.0 版本安](#) (7933)[CentOS安装nvidia显卡驱](#) (7814)

### 评论排行

[MTCNN训练数据整理](#) (4)[linux pytorch 安装](#) (2)

计算机玩Atari游戏的要求是输入原始图像数据，也就是210x160像素的图片，然后输出几个按键动作。总之就是和人类的要求一样，纯视觉输入，然后让计算机自己玩游戏。那么这种情况下，到底有多少种状态呢？有可能每一秒钟的状态都不一样。因为，从理论上讲，如果每一个像素都有256种选择，那么就有：

$$256^{210 \times 160}$$

这简直是天文数字。所以，我们是不可能通过表格来存储状态的。我们有必要对状态的维度进行压缩，解决办法就是 价值函数近似Value Function Approximation

## 3 价值函数近似Value Function Approximation

什么是价值函数近似呢？说起来很简单，就是用一个函数来表示 $Q(s,a)$ 。即

$$Q(s, a) = f(s, a)$$

$f$ 可以是任意类型的函数，比如线性函数：

$$Q(s, a) = w_1 s + w_2 a + b \quad \text{其中 } w_1, w_2, b \text{ 是函数 } f \text{ 的参数。}$$

大家看到了没有，通过函数表示，我们就可以无所谓 $s$ 到底是多大的维度，反正 $Q$ 。

这就是价值函数近似的基本思路。

如果我们就用  $w$  来统一表示函数 $f$ 的参数，那么就有

$$Q(s, a) = f(s, a, w)$$

为什么叫近似，因为我们并不知道 $Q$ 值的实际分布情况，本质上就是用一个函数来近似。

关闭





- TX1 安装 ROS Indigo (2)
- pytorch学习笔记 (八) : (2)
- Jetson TX1 开发教程 (4 (2)
- opencv 仿射变换 根据眼 (2)
- [iOS]iOS结合OpenCV做 (2)
- MTCNN训练整理 (2)
- 深度学习中的激活函数导 (1)
- MAT: A Multimodal Atten (1)

### 推荐文章

- \* CSDN日报20170725——《新的开始，从研究生到入职亚马逊》
- \* 深入剖析基于并发AQS的重入锁(ReentrantLock)及其Condition实现原理
- \* Android版本的"Wannacry"文件加密病毒样本分析(附带锁机)
- \* 工作与生活真的可以平衡吗？
- \* 《Real-Time Rendering 3rd》提炼总结——高级着色：BRDF及相关技术
- \* 《三体》读后思考-泰勒展开/维度打击/黑暗森林

### 最新评论

- MAT: A Multimodal Attentive Traqq\_39582061: 可以请问一下这篇论文发表的期刊吗？我这里搜不到。。
- [iOS]iOS结合OpenCV做视频流代码学习者: 哪3个C++问题
- [iOS]iOS结合OpenCV做视频流

$$Q(s, a) \approx f(s, a, w)$$

## 4 高维状态输入，低维动作输出的表示问题

对于Atari游戏而言，这是一个高维状态输入（原始图像），低维动作输出（只有几个离散的动作，比如上下左右）。那么怎么来表示这个函数f呢？

难道把高维s和低维a加在一起作为输入吗？

必须承认这样也是可以的。但总感觉有点别扭。特别是，其实我们只需要对高维状态进行降维，而不需要进行降维处理。

那么，有什么更好的表示方法吗？

当然有，怎么做呢？

其实就是  $Q(s) \approx f(s, w)$ ，只把状态s作为输入，但是输出的时候输出每一个动作的Q值，也就是输出一组值  $[Q(s, a_1), Q(s, a_2), Q(s, a_3), \dots, Q(s, a_n)]$ ，记住这里输出是一个值，只不过是包含了所有动作的Q值的向量而已。这样我们就只要输入状态s，而且还同时可以得到所有的动作Q值，也将更方便的进行下一步动作的选择与Q值更新（这一点后面大家会理解）。

## 5 Q值神经网络化！

终于到了和深度学习相结合的一步了！

意思很清楚，就是我们用一个深度神经网络来表示这个函数f。

这里假设大家对深度学习特别是卷积神经网络已经有基本的理解。如果不是很懂，可以先看我的译系列文章。

关闭



Await\_Xpf: xcode 中都弄好之后报3个 c++ 的问题 , 遇到过吗? 794778062希望能与您沟通交流...

#### MTCNN训练整理

huangbo1221: 您好, 能参考一下修改后的softmax\_loss\_layer和euclidean\_loss\_layer...

Jetson TX1 开发教程 (4) --TensorFlow 代码学习者: @u013768935:是的

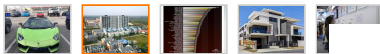
Jetson TX1 开发教程 (4) --TensorFlow 代码学习者: 博主, 在tx2上运行上述例程的话, 过程是先安装jetpack, 然后配置caffe吗?

pytorch学习笔记 (八): PyTorch 代码学习者: 可以啊

pytorch学习笔记 (八): PyTorch 代码学习者: 博主您好! 请教您: 我不用pyTorch在Torch中实现可视化每一层的输出吗?

机器学习&深度学习资料汇总 (含PDF, PPT, 视频) 代码学习者: 不要以为蓝色的就是链接, 有可能就是字体是蓝色的

## 美国房价



以DQN为例, 输入是经过处理的4个连续的84x84图像, 然后经过两个卷积层, 两个全连接层, 最后输出包含每一个动作Q值的向量。

对于这个网络的结构, 针对不同的问题可以有不同的设置。如果大家熟悉Tensorflow, 那么肯定知道创建一个网络是多么简单的一件事。这里我们就不具体介绍了。我们将在之后的DQN tensorflow实战篇进行讲解。

总之, 用神经网络来表示Q值非常简单, Q值也就是变成用Q网络 (Q-Network) 来表示。接下来就到了很多人都会困惑的问题, 那就是

怎么训练Q网络???

## 6 DQN算法

我们知道, 神经网络的训练是一个最优化问题, 最优化一个损失函数loss function, 也就是标签和网络输出之间的差异, 目标是让损失函数最小化。为此, 我们需要有样本, 巨量的有标签数据, 然后通过反向传播使用梯度下降更新神经网络的参数。

所以, 要训练Q网络, 我们要能够为Q网络提供有标签的样本。

所以, 问题变成:

如何为Q网络提供有标签的样本?

答案就是利用Q-Learning算法。

大家回想一下Q-Learning算法, Q值的更新依靠什么? 依靠的是利用Reward和当前状态S<sub>t</sub>以及动作a来更新Q值。

$$R_{t+1} + \lambda \max_a Q(S_{t+1}, a)$$

关闭



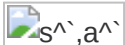
## 呼伦贝尔自驾



因此，我们把目标Q值作为标签不就完了？我们的目标不就是让Q值趋近于目标Q值吗？

因此，Q网络训练的损失函数就是



上面公式是   $s^a, a^a$  即下一个状态和动作。这里用了David Silver的表示方式，看起来比较清晰。

既然确定了损失函数，也就是cost，确定了获取样本的方式。那么DQN的整个算法也就成型了！

接下来就是具体如何训练的问题了！

## 7 DQN训练

我们这里分析第一个版本的DQN，也就是NIPS 2013提出的DQN。



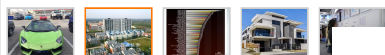
我们分析了这么久终于到现在放上了DQN算法，真是不容易。如果没有一定基

具体的算法主要涉及到Experience Replay，也就是经验池的技巧，就是如何在

由于玩Atari采集的样本是一个时间序列，样本之间具有连续性，如果每次得到效果会不好。因此，一个很直接的想法就是把样本先存起来，然后随机采样如思。按照脑科学的观点，人的大脑也具有这样的机制，就是在回忆中学习。

那么上面的算法看起来那么长，其实就是反复试验，然后存储数据。接下来数据，进行梯度下降！

## 美国房价



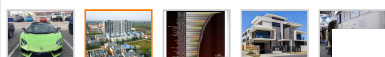
关闭



## 呼伦贝尔自驾



## 美国房价



也就是

在DQN中增强学习Q-Learning算法和深度学习的SGD训练是同步进行的！

通过Q-Learning获取无限量的训练样本，然后对神经网络进行训练。

样本的获取关键是计算y，也就是标签。

## 8 小结

好了，说到这，DQN的基本思路就介绍完了，不知道大家理解得怎么样？在下一篇文章中，我们将分析年来的发展变化！感谢知友们的关注！

文中图片引用自

[1] Mnih, Volodymyr, et al. "Playing atari with deep reinforcement learning." arXiv preprint arXiv:1312.5602 (2013).

[2] Mnih, Volodymyr, et al. "Human-level control through deep reinforcement learning." Nature 518.7540 (2015): 529-533.

**版权声明：本文为原创文章，未经允许不得转载**

顶  
0

踩  
0





## 呼伦贝尔自驾



上一篇 [caffe中Istm的实现以及Istm1ayer的理解](#)

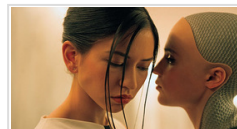
下一篇 [Reinforcement Learning \(DQN\) tutorial](#)

### 相关文章推荐

- [google deepMind DQN 源码解读\(1\)](#)
- [Deep Reinforcement Learning 基础知识 \(DQN方...](#)
- [漫谈DQN之Q-Learning](#)
- [DQN 原理 \(一\) : 环境, 行为, 观测](#)
- [DQN 从入门到放弃1 DQN与增强学习](#)
- [用Tensorflow基于Deep Q Learning DQN 玩Flappy...](#)
- [DQN 原理 \(二\) : 理解 DQN 中的“Q”](#)
- [DQN](#)
- [double dqn report](#)
- [强化学习系列<4>DQN](#)



卖车



人工智能机器人



寇驰折扣店



奥特莱斯



婚庆租车

### 猜你在找

【直播】机器学习&深度学习系统实战 (唐宇迪)

【直播回放】深度学习基础与TensorFlow实践 (王琛)

【直播】机器学习之凸优化 (马博士)

【直播】机器学习之概率与统计推断 (冒教授)

【直播】TensorFlow实战进阶 (智亮)

【直播】Kaggle 神器 : 2017年Kaggle竞赛

【直播】计算机视觉原理

【直播】机器学习之矩阵

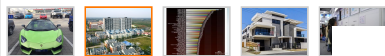
【直播】机器学习之数学

【直播】深度学习30天

### 查看评论

暂无评论

## 美国房价



关闭



## 呼伦贝尔自驾



## 发表评论

用户名： haijunz

评论内容：



提交

\* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

公司介

网

京 ICP

联系方式 | 版权声明 | 法律顾问 | 问题报告 | 合作伙伴 | 论坛反馈

webmaster@csdn.net

400-660-0108 | 北京创新乐知信息技术有限公司 版权所有 | 江苏知之为计算机有限公司 | 江苏乐知网络技术有限公司

17, CSDN.NET, All Rights Reserved



关闭

## 美国房价

