基于深度 Q 值网络的自动小车控制方法*

王立群 朱 舜 韩 笑 何 军
(南京信息工程大学 电子与信息工程学院 南京 210044)

摘 要：随着计算机技术和人工智能的飞速发展，无人驾驶车辆成为了一个新的热点。提出了一种自动小车的验证模型来模拟无人车，并验证了深度 Q 值网络(deep Q network，DQN)算法对自动小车的控制。该算法使用了强化学习和神经网络技术，能够在缺乏先验知识的情况下，根据获取的传感器信息训练神经网络，然后做出正确的决策，实现对车辆的控制，达到躲避障碍物的效果。此外，通过在模拟环境下的实验验证了 DQN 算法对自动小车的控制效果。实验结果表明，经过一定时间的训练，DQN 算法可以有效的控制自动小车。

关键词：自动小车控制；强化学习；神经网络
中图分类号：TP242.6 **文献标识码**：A **国家标准学科分类代码**：510.8050

Control method of autonomous mini-car based on deep Q-network

Wang Liqun Zhu Shun Han Xiao He Jun
(College of Electronic and Information Engineering, Nanjing University of Information Science and Technogy, Nanjing 210044, China)

Abstract: With the rapid development of computer technology and artificial intelligence, unmanned vehicles have become a new hot spot. In this paper, a verification model of the automatic car is proposed to simulate the unmanned vehicle, and the deep Q network (DQN) algorithm is used to control the automatic car. The algorithm uses reinforcement learning and neural network technology, in the case of less prior knowledge, it can train the neural network according to the obtained sensor information ,then make the right decision to achieve the control of the vehicle and the effect of avoiding obstacles. In addition, this paper verifies the control effect of DQN algorithm on automatic trolley by experimenting in simulated environment. Experimental results show that, after a certain period of training, DQN algorithm can effectively control the automatic car.

Keywords: autonomous mini-car control; reinforcement learning; neural network

0 引 言

无人驾驶车辆是由微机控制的智能车辆的简称，本质是移动机器人[1]。它涉及众多学科的前沿研究领域，包括：计算机技术、环境感知、模式识别、导航定位、自主决策等[1-2]。研究无人车的主要目的是希望车辆在没有认为干预的情况下，能够根据传感器和路况等信息做出正确的决策，达到预定的目的地，并且提高行车安全和效率。将人工智能与传统汽车工业相结合，开发出无人驾驶车辆，被认为是人工智能最具发展前景的行业之一。

自动小车作为一个机器人系统，主要涉及到机器人技术中的控制、感知和路径规划这三大块内容。控制作为机器人系统的核心技术之一，是自动小车能否根据传感器信息和任务信息，做出正确决策，躲避障碍物，做出合理路径规划到达目的地的关键。传统的自动小车控制算法基于构建车辆的模型，控制的精确度取决于模型的精确性[3]。如基于 PID 控制的自动小车控制算法，通过传感器判断车与行驶路径的相对位置，当出现偏差时，就调整车轮的转角，控制车辆回到行驶路径[4]。还有部分学者尝试采用模糊控制[5-6]方法，提高自动小车控制的鲁棒性。大连理工大学的研究人员就提出了基于遗传优化的自动小车模糊控制算法，通过摄像机建立基于视觉的自动小车模糊控制系统模型，使用遗传算法优化模糊控制器，达到鲁棒控制的效果[7]。

随着计算机技术的进步，人工智能也得到了飞速的发展，诸如神经网络[8]、强化学习等前沿领域也被应用于自动小车控制中。这类算法通过训练采集的数据信息，不断优化控制器，达到理想的结果。如 Levine 等人[9]在 2015 年提出的基于卷积神经网络和强化学习的端到端(end-to-end)的训练深层视觉的方法，通过收集数据(图像和机器人关节角度)，使用神经网络和强化学习进行训练，直接输出控制

收稿日期：2017-04
* 基金项目：上海市北斗导航与位置服务重点实验室开放基金、江苏省大学生创新训练项目重点项目(201610300033)资助

动作。DeepMind 实验室在 2015 年提出的深度确定性策略梯度(DDPG)算法,采用两个神经网络,一个网络输出控制动作,另一个网络评估该动作,通过不断的优化,达到理想的控制效果[10]。该算法已经可以在模拟器上实现对车辆的控制。

本文采用深度 Q 网络(deep Q network)[11]算法作为自动小车的控制算法,该算法不需要对车辆进行建模,通过处理以时间为序列的传感器数据,对控制器进行训练,即可达到对自动小车的控制。将 DQN 算法应用于自动小车模拟环境上,实验结果表明,DQN 算法对车辆的控制可以达到理想的效果。

1 自动小车模型设计

工业界或者大型公司研制的无人车是通过在传统汽车上安装高精度的激光雷达、摄像头、超声波探测器等传感器和车载计算机来完成无人驾驶的功能,如 Google Car,特斯拉 Autopilot 等。该方法成本非常高,比如高精度的激光雷达就高达数十万元,因此不适合高校实验室。为此,本文提出了自动小车的验证模型,用自动小车模拟无人车,小车的硬件由双目摄像头(ZED)、超声波传感器、陀螺仪、NVIDIA Jetson TK1 开发板和电机驱动的小车组成。此外,小车需要一个操作系统,将小车的各个部分连接起来,形成一个完整的系统。因此,小车的采用机器人操作系统(robot operating system)。

ROS 操作系统将自动小车控制系统分为一个个的节点(Node)[12],如一个节点控制双目摄像头,一个节点控制超声波传感器,一个节点控制马达和舵机等,ROS 操作系统通过此方法将各个模块连接成一个完整的自动小车系统。首先,双目摄像头将左右并排的图像通过 USB 3.0 发送给 Jetson TK1 开发板,开发板使用 GPU 构建深度图像,同时超声波传感器将超声波信号转换成电信号,通过串口发送给 NVIDIA TK1 开发板。开发板根据接收的传感器信号,使用 DQN 算法做出合理的决策,控制小车的马达和舵机,达到控制小车避开障碍物的目的

自动小车可以很好的模拟无人车的工作原理,且成本低廉,适合高校实验室进行无人车相关技术的研究。自动小车系统框图如图 1 所示。

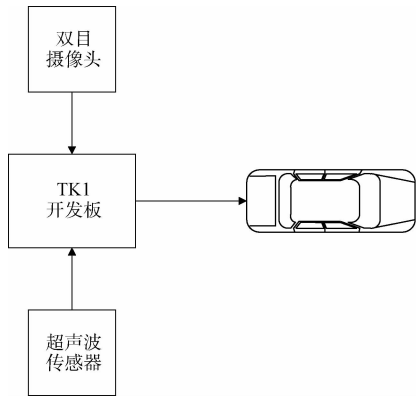

图 1 自动小车系统框图

2 强化学习与 Q 值神经网络

强化学习是系统与环境产生交互,自主探索环境,采取动作影响环境并且从环境中得到奖赏的过程。强化学习的目标是在这个交互过程中不断学习策略,最大化长期奖励总和,作为系统的最优策略[13-14]。

一般而言,强化学习问题是使用马尔科夫决策过程(markov decision processes,MDP)来建立模型的,因此,强化学习满足马尔科夫属性[13]。

一个基本的 MDP 可以用 $<S,A,T,R>$ 表示,其中 S 是状态空间,A 是动作空间,T 是状态转移函数,R 是奖赏函数。

强化学习框架如图 2 所示。Agent 从给定状态 s 开始,采取动作 a,以 T 的概率转移到一个新状态 s',同时环境给 Agent 反馈奖励 r,直到结束状态为止。Agent 的目的就是通过不停的学习,最大化长期奖励总和 R。长期奖励总和 R 定义为:

$$R = \sum_{t=0}^{\infty} \gamma^t r_t \tag{1}$$

式中:γ 是折扣因子,r_t 是在 t 步的奖励。

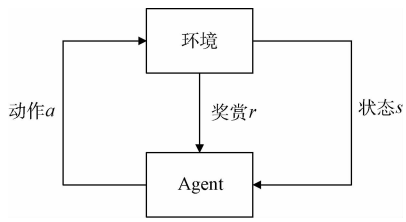

图 2 强化学习框架

由强化学习的框图可以得知,奖励 r 与状态 s 和动作 a 有关,价值函数可以很好的描述奖励 r 与状态 s 和动作 a 的这种关系。本文使用了最优动作价值函数 $Q*(s,a)$,定义如下:

$$Q*(s,a) = \max_{\pi} E[R \mid s_t = s, a_t = a, \pi] \tag{2}$$

式中:π 是 Agent 学习到的策略。

最优动作价值函数满足贝尔曼方程的特性,即当 $t+1$ 时刻,状态 s' 对应的所有可能的动作 a' 已知,那么最优策略就是选择一个动作使得 $Q*(s',a')$ 能够最大化奖励的期望 $r+\gamma Q*(s',a')$。因此可以将价值函数 Q 转换为迭代形式:

$$Q*(s,a) = E[r+\gamma Q*(s',a') \mid s,a] \tag{3}$$

在 RL 算法中,通常使用神经网络构造的函数逼近器来近似动作状态函数,Q 值神经网络定义如下:

$$Q(s,a) \approx f(s,a,w) \tag{4}$$

式中:w 是神经网络的网络参数。

3　自动小车控制算法

自动小车将传感器信息作为输入，是以时间为序列的连续状态信息。本文采用 DQN 算法作为自动小车的控制算法，该算法将 Q-学习算法与神经网络相结合，可以有效解决输入状态为连续状态空间的问题。

Q-学习算法是强化学习里的经典算法之一，它的思想就是基于迭代形式的价值函数，算法的基本形式如下：

$$Q(s,a)=Q(s,a)+\alpha(r'+\lambda Q*(s',a)-Q(s,a)) \quad (5)$$

Q-学习使用 ε-贪心策略选取动作执行，根据公式更新(4)更新 Q 值，直到 Q 值收敛。

由于传感器信息是一个时间序列，样本之间具有连续性，如果直接根据样本更新 Q 值，效果不理想。因此，DQN 采用了经验回放(experience replay)的技巧，把样本先存起来，达到一定程度后根据 minibatch 的大小，再随机采样若干个样本，这样做就消除了样本的连续性。

DNQ 算法中使用了神经网络作为函数逼近器来逼近 Q 函数，而训练神经网络的基本思想是通过最小化代价函数来训练神经网络的参数，以此获得最优的神经网络参数。因此，在 Q 网络中，代价函数定义如下：

$$L_i(w_i)=E[(y_i-Q(s_i,a_i;w_i))^2] \quad (6)$$

其中 $y_i=E[r+\gamma Q*(s',a';w_{i-1})\mid s,a]$

求出代价函数 L 关于参数 w 的梯度，就可以用随机梯度下降等方法训练神经网络，获得最优的参数 w。

本文采用了一个全连接神经网络作为 Q 网络，网络有 3 层，一个输入层，一个隐含层和一个输出层，隐含层的大小为 164×150。首先将传感器数据转换成神经网络可识别的数据结构张量(tensor)，接着将张量数据输入到输入层，输出层输出的是所有可能动作的 Q 值，选择最大 Q 值对应的动作作文输出动作 a 输出。神经网络结构如图 3 所示。

此外，奖励函数定义如下：当小车与障碍物发生碰撞时，奖励函数 $r=-500$，并且立即结束当前 episode，开始新的 episode。自动小车控制算法如表 1 所示。

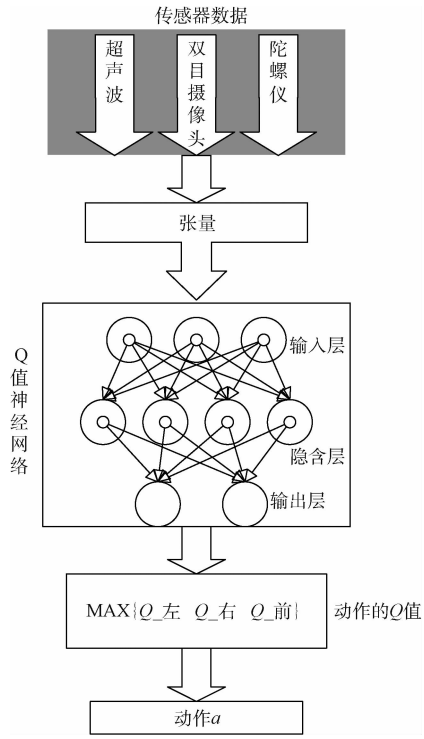

图 3　神经网络结构

表 1　自动小车控制算法

小车控制算法：
初始化缓存 D
初始化 Q 函数
for episode=1，M do
　初始化车辆状态
　for t=1，T do
　　读取传感器数据，得到状态 st
　　根据概率 ε 选择随机动作 at
　　否则选择动作 $a_t=\mathrm{argmax}_a Q(s_t,a)$
　　执行动作 at，得到奖励 rt 和传感器的下一个状态 s_{t+1}
　　将 $(st,at,rt,st+1)$ 存储到缓存 D
　　根据 minibatch 的大小，从 D 中随机抽取若干个 $(sj,aj,rj,sj+1)$，作为标签值

$$\text{令 } yj=\begin{cases} r_j(s_{t+1}\text{ 是最终状态}) \\ r_j+\lambda\max a_j+Q(s_{j+1},a_{j+1}) \\ (s_{t+1}\text{ 不是最终状态}) \end{cases}$$

　根据 $L=E[(yj-Q(sj,aj))^2]$ 代价函数，基于随机梯度下降方法更新 Q 网络
　end for
end for

4　实验与结果分析

本节首先介绍了实验平台和试验中所需设置的参数，随后评估了训练结果的好坏。

4.1　实验平台描述

本文在模拟环境下实现了自动小车的控制算法。使用了 Pygame[15] 和 Pymunk[16] 所提供的 Carmunk 小游戏作为自动小车的模拟环境，模拟环境与小车所处的真实环境相类似，小车前装有距离传感器，有静态障碍物、动态障碍物以及活动边界。小车的目标是通过接收距离传感器的信息在活动边界内移动，并且避开障碍物。

4.2　实验参数设置

本试验基于 Python 语言进行编程，程序以深度学习框架 Keras 为基础，采用了一个全连接神经网络作为 Q 网络，隐含层的大小为 164×150，为了防止神经网络过拟合，将 dropout 设置为 0.2。训练算法将 minibatch 大小设置为 400，缓存 D 大小为 50 000。

此外，在编写算法时，对表 1 提出的算法做出了部分修

改,包括贪心算法的 ε 在训练过程中会从 1 减小到 0.1,这样做的目的是随着训练的进行,减小随机动作的出现。

4.3 实验结果分析

程序还记录了小车每一个 episode 的行驶距离和训练时代价函数 L 的大小,并且绘制了相应的图表。从图表上可以很直观的看到,小车的行驶距离随着训练的进行不断增长,同时代价函数也不断下降,证明控制算法可以很好的控制小车躲避障碍物。图 4 是小车行驶距离和 episode 的关系,横轴是 episode,纵轴是小车在模拟环境下行驶的距离,距离的基本单位是帧,从图上可以看出,在 episode 达到 1 800 左右时,小车的行驶距离有了明显的增长,当 episode 达到 2 000 时,小车已经能够行驶超过 3 000 帧的距离。

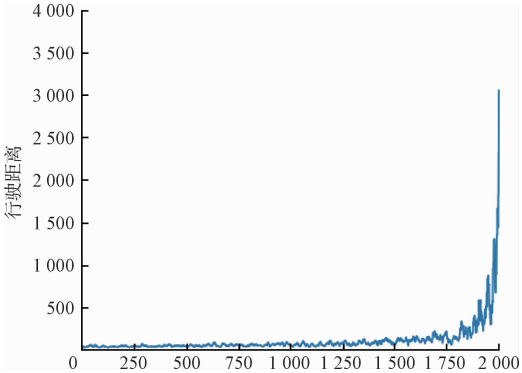

图 4 自动小车行驶距离

图 5 是训练时小车做的每一个动作与代价函数值的关系,横轴为小车的每一次动作,纵轴为代价函数的大小,即训练误差。由图可知,随着训练进行,误差有明显的下降,最终会趋于一个稳定值。

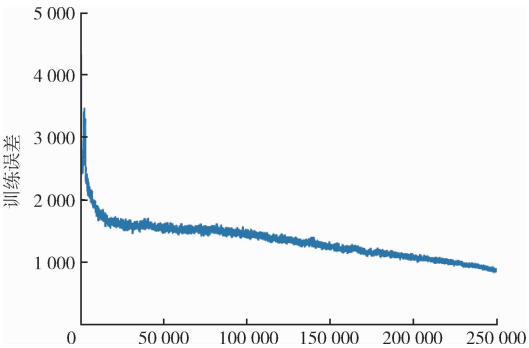

图 5 代价函数

5 结 论

本文提出了一种自动小车验证模型,并且使用 DQN 控制算法在模拟环境下实现了控制自动小车躲避障碍物的功能。实验表明 DQN 控制算法可以有效的实现对自动小车的控制,为今后实现自动小车的控制算法有指导作用。

本文在模拟环境下实现了自动小车的控制算法,下一步的计划是将该算法运行在 Jetson TK1 开发板上,实现用 Jetson TK1 开发板控制小车的目的。

参考文献

[1] 叶伟铨. 无人车的自主导航与控制研究[D]. 广州:华南理工大学,2016.

[2] 赵盼. 城市环境下无人驾驶车辆运动控制方法的研究[D]. 合肥:中国科学技术大学,2012.

[3] 邓伟,刘平,李贻斌,等. 基于模型预测控制的排爆机器人轨迹跟踪算法研究[J]. 仪器仪表学报,2016,37(S1):1-6.

[4] 时巧,李财,邓渊. 智能巡线小车的设计[J]. 微型机与应用,2015(9):78-80.

[5] 徐娟,陈时桢,何烊剑,等. 基于模糊 PID 的平衡头自适应控制策略研究[J]. 电子测量与仪器学报,2016,30(6):895-902.

[6] 冯超,陈双叶. 基于模糊控制的 PAC 控制器的设计[J]. 国外电子测量技术,2016,35(7):47-51.

[7] 郭景华, 胡平, 李琳辉, 等. 基于遗传优化的无人车横向模糊控制[J]. 机械工程学报, 2012, 48(6):76-82.

[8] 刘浩然,赵翠香,李轩,等. 一种基于改进遗传算法的神经网络优化算法研究[J]. 仪器仪表学报,2016,37(7):1573-1580.

[9] LEVINE S, FINN C, DARRELL T, et al. End-to-end training of deep visuomotor policies[J]. 2015, arXiv:1504.00702.

[10] LILLICRAP T P, HUNT J J, PRITZEL A, et al. Continuous control with deep reinforcement learning[J]. 2015, arXiv:1509.02971.

[11] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Playing atari with deep reinforcement learning[J]. 2013, arXiv:1312.5602.

[12] SANTOS J M, PORTUGAL D, ROCHA R P. An evaluation of 2D SLAM techniques available in robot operating system[C]. IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), 2013: 1-6.

[13] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning[J]. Nature, 2015, 518(7540): 529-533.

[14] LILLICRAP T P, HUNT J J, PRITZEL A, et al. Continuous control with deep reinforcement learning[J]. 2015, arXiv:1509.02971.

[15] SHINNERS P. PyGame-Python game development[R]. 2011.

[16] RUÍZ J M. Física: pygame y pymunk[J]. Linux Magazine, 2011(68): 39-42.

作者简介

王立群,硕士研究生,研究方向为强化学习。
E-mail:576979604@qq.com