基于深度强化学习的桥梁经济跨径选择

张乐业[1]，田翔翔[1]，张成利[2]，张洪俊[3]

（1. 江苏财会职业学院，连云港 222061；

2. 连云港开放大学，连云港 222006；

3. 万世先行数智交通科技有限公司，南京 210016）

摘要：利用深度 Q 网络算法实现桥梁经济跨径的选择。桥梁跨径的选择对桥梁的总造价有显著影响，合理的跨径选择可以降低工程成本。对桥梁经济跨径进行理论分析，推导出经济跨径的理论求解公式。详细描述了桥梁模拟环境的构建过程，包括环境的观测空间、动作空间和回报函数等。构建智能体，利用卷积神经网络来逼近 Q 函数，采用贪婪策略进行动作选择，通过经验回放机制进行训练。测试验证了智能体能够成功学习到最优策略，实现桥梁经济跨径的选择。本研究为桥梁设计领域提供了一种潜在的决策工具。

关键词：强化学习；桥梁经济跨径；深度 Q 网络；桥梁模拟环境；智能体

中图分类号：TP181；U442.5　　**文献标志码：**A

Economic span selection of bridge based on deep reinforcement learning

Leye Zhang[1]，Xiangxiang Tian[1]，Chengli Zhang[2]，Hongjun Zhang[3]

（1. Jiangsu College of Finance & Accounting, Lianyungang, 222061, China;

2. Lianyungang Open University, Lianyungang, 222006, China;

3. Wanshi Antecedence Digital Intelligence Traffic Technology Co., Ltd, Nanjing, 210016, China）

Abstract: Deep Q-network algorithm is used to select economic span of bridge. Selection of bridge span has a significant impact on the total cost of bridge, and a reasonable selection of span can reduce engineering cost. Economic span of bridge is theoretically analyzed, and the theoretical solution formula of economic span is deduced. Construction process of bridge simulation environment is described in detail, including observation space, action space and reward function of the environment. Agent is constructed, convolutional neural network is used to approximate Q function, ε greedy policy is used for action selection, and experience replay is used for training. The test verifies that the agent can successfully learn optimal policy and realize economic span selection of bridge. This study provides a potential decision-making tool for bridge design.

Keywords: reinforcement learning; economic span of bridge; deep Q-network; bridge simulation environmen; agent

0　引言

对于一座较长的桥梁，如何选择合理的桥梁分孔方案，是桥梁设计师的重要任务。桥梁跨径的选择对总造价有巨大的影响。当没有通航等硬性条件制约时，桥梁跨径应该选择经济跨径，即上、下部结构的总造价最低。就造价而言，跨径越大则孔数越少，这可以降低下部结构费用，却使上部结构费用增高；反之，则上部结构费用降低，但下部结构费用增高。桥位处的地形、地质、水文等因素，影响经济跨径的取值，比如跨海桥梁的经济跨径数值通常会大于平原河流浅滩处的。桥梁分孔是复杂的工程问题，需要通过技术、经济等方面的分析比较，才能得到比较理想的设计方案[1-2]。

强化学习（Reinforcement Learning，RL）起源于 20 世纪 50~60 年代心理学中的动物学习和最优控制的优化理论[3-4]。2016 年 AlphaGo 击败了人类世界围棋冠军，将默默发展半个世纪之久的强化学习正式带入了大众的视野。程国忠基于深度强化学习（Deep Reinforcement Learning，DRL），提出高层剪力墙结构智能设计方法[5]；何佳琛基于强化学习，探究大跨度桥梁风致振动的主动控制[6]；袁泉基于深度强化学习，提出铁路智能选线方法[7]；罗睿锋基于强化学习，提出桁架结构生成式设计算法[8]；Cheng M 采用深度强化学习，提出老龄桥梁荷载等级规划决策框架[9]；Yang DY 基于深度强化学习，提出考虑资产和网络风险的桥梁管理方法[10]。而强化学习应用于桥梁经济跨径选择的研究，尚未见报道。

本文自建桥梁模拟环境，运用深度强化学习的深度 Q 网络（Deep Q-Network, DQN）算法，探究桥梁经济跨径

作者简介：张乐业（2003—），男，2023 级人工智能技术应用学生

通信作者：田翔翔（1991—），男，讲师，主要研究方向：视觉传达设计，E-mail：3514956127@qq.com

的选择（本文源代码开源地址 https://github.com/zhangleye/BridgeSpan-DQN）。

1　桥梁经济跨径理论分析和深度强化学习简介

1.1 桥梁经济跨径理论分析

这里借鉴了参考文献[2]的思路，并且为了便于理论分析作了适当简化（如将两个桥台造价合并成一个桥墩造价，假设造价是跨径的可导函数而不考虑结构形式变化引起的突变）。

桥梁总造价是上、下部结构造价之和。

$$W_{total}=W_{upper}+W_{under} \tag{1}$$

式中：W_{total}为桥梁总造价；W_{upper}为上部结构造价；W_{under}为下部结构造价。

上部结构由桥面系（铺装、防撞墙等）、承重结构组成。桥面系造价与跨径无关。假设承重结构造价是跨径的幂函数。

$$W_{upper}=(gS_1+C_1x^mS_2)L \tag{2}$$

式中：gS_1为纵向单位长度的桥面系造价，g为纵向单位长度桥面系材料用量，S_1为桥面系材料单价；$C_1x^mS_2$为纵向单位长度的承重结构造价，C_1x^m为纵向单位长度承重结构材料用量，C_1为幂函数的系数，x为跨径（幂函数的自变量），m为幂函数的指数（不小于 1），S_2为承重结构材料单价；L为多孔跨径总长。

假设下部结构造价也是跨径的幂函数。

$$W_{under}=\frac{L}{x}C_2x^{1/n}S_3 \tag{3}$$

式中：$\frac{L}{x}$为桥墩数量；$C_2x^{1/n}S_3$为单个桥墩造价，$C_2x^{1/n}$为单个桥墩材料用量，C_2为幂函数的系数，$\frac{1}{n}$为幂函数的指数（n大于 1），S_3为桥墩材料单价。

桥梁总造价对跨径求导：

$$\frac{d(W_{total})}{dx}=C_1mx^{m-1}S_2L+(\frac{1}{n}-1)C_2x^{1/n-2}S_3L \tag{4}$$

桥梁总造价函数图形是凹的，当导数为 0 时，桥梁总造价最低，即：

$$C_1mx^{m-1}S_2L=(1-\frac{1}{n})C_2x^{1/n-2}S_3L \tag{5}$$

$$C_1x^{m+1}S_2=\frac{n-1}{mn}C_2x^{1/n}S_3 \tag{6}$$

式中：$C_1x^{m+1}S_2$为单跨承重结构造价。

即单跨承重结构造价=$\frac{n-1}{mn}$单个桥墩造价时，为经济跨径。

1.2 深度强化学习简介

强化学习是机器学习的一个分支，它解决的是智能体（agent）在环境（environment）中应该采取什么样的行动(action)，从而使回报(reward)最大化。环境内置了状态更新和奖惩的规则，但这些规则对智能体而言是一个黑箱，智能体只能通过试错学习(Trail-and-error)方法与环境交互，通过环境的反馈来找到最优行动策略(policy)。例如智能体在桥梁模拟环境中，先随机选取承重结构材料和跨径，然后尝试调整材料和跨径，一些调整会让造价增加，另一些调整会让造价降低，通过不断地尝试和总结，最终找到桥梁造价最低的材料和跨径。

有许多算法可以帮助智能体找到最优行动策略，其中 Q-learning 算法影响深远，为后来很多算法奠定了基础。Q-learning 算法的具体步骤是：首先建立一个表格（Q 值表，行数为状态数目、列数为动作数目、单元格数值为该状态下执行该动作的价值），一开始表格的单元格数值（Q 值）是随机值；然后智能体与环境交互，根据环境的反馈逐步更新表格中的各个 Q 值，通过多轮迭代，直至所有 Q 值几无变化为止；有了正确的 Q 值表，智能体就知道什么状态下该执行什么样的动作，从而实现目标。

深度强化学习将深度学习（Deep Learning，DL）的感知能力和强化学习的决策能力相结合，根据输入的状态图像进行动作控制，这种端到端的学习方式更接近于人类思维。DQN 是目前主流且广泛应用的深度强化学习算法。DQN 实质上是用卷积神经网络来代替 Q 值表，具体步骤是：首先建立卷积神经网络（输入为状态图像、输出为该状态下执行各种动作的 Q 值），一开始神经网络的权重参数是随机值；然后智能体与环境交互，根据环境的反馈逐步更新神经网络权重参数，通过多轮迭代，直至神经网络的输出（Q 值）几无变化为止；有了正确的神经网络权重参数，就可以为每个状态输出正确的 Q 值，智能体就知道什么状态下该执行什么样的动作，从而实现目标。

2　基于深度强化学习的桥梁经济跨径选择

采用强化学习解决工程实际问题，首先要建立环境，其次是建立智能体，最后是训练、测试。

本任务基于 Python3.10 编程语言、OpenAI Gym0.26.2 强化学习工具包、TensorFlow2.10 及 Keras2.10 深度学习平台框架。

2.1 自建桥梁模拟环境

因市面上强化学习论文、书籍中几乎均未涉及自建环境的内容，故这里给出详细的过程。

1.该环境是一个二维方格网，共 3 行、80 列，坐标原点位于左上角。行代表承重结构材料类别，从上至下第 0、1、2 行依次为混凝土结构、钢混组合结构、钢结构。列代表跨径，从左到右第 0、1、2、……、77、78、79 列依次为 10、20、30、……、780、790、800 米。每个单元格交替采用黑色、灰色加以区别，智能体所在单元格用红色表示，观测空间数目为 3×80=240 个。详见图 1。

图 1 桥梁模拟环境示意图

Fig.1 Schematic diagram of bridge simulation environmen

环境的动作空间数目为 5 个，分别为 0（原地不动）、1（向上一格）、2（向下一格）、3（向左一格）、4（向右一格）。

智能体可以在环境中任何一个单元格，做任何动作（如果跑出环境则返回原地）。图 1 中智能体位于第 1 行、第 58 列，如果执行“向上一格”动作，动作后位置是第 0 行、第 58 列。智能体不断地在环境中运动，通过回报最大化来找到经济跨径的单元格。

环境属性代码见图 2。

```python
class BridgeSpanEnv(gym.Env):
    def __init__(self, Min_length=10,Max_length=800,step_length=10,max_steps=200):
        self.Min_length=Min_length
        self.Max_length=Max_length
        self.step_length=step_length
        self.max_steps=max_steps
        self.Material_Type=3
        self.state_pos_dict = np.arange(self.Min_length,self.Max_length+self.step_length,self.step_length)
        self.observation_space = gym.spaces.Discrete( self.Material_Type*len(self.state_pos_dict) )
        self.states=np.arange(self.observation_space.n)
        self.state=random.choice(self.states)
        self.action_space = gym.spaces.Discrete(5)
        self.actions = [NOOP, UP, DOWN, LEFT, RIGHT]
        self.action_pos_dict = {NOOP:[0,0],UP:[-1,0],DOWN:[1,0],LEFT:[0,-self.step_length],RIGHT:[0,self.step_length]}
        self.step_num = 0
        self.done = False
        self.truncated= False
        self.info = {}
        self.viewer = None
        self.img_shape = [16*self.Material_Type, 16*len(self.state_pos_dict), 3]
```

图 2 环境属性代码

Fig.2 Code for environment properties

2. step()方法是环境的核心，它接收智能体的动作作为参数，返回状态、回报和其它信息。

它首先计算动作之后的行列坐标和状态，然后计算动作的回报。

求解最优策略是强化学习的唯一任务，智能体通过改进策略使回报最大化，因此回报函数的设计是关键。本次任务是寻找到桥梁经济跨径，最低的总造价就是目标，因此能够降低造价的动作产生高回报、能够增加造价的动作产生低回报，即造价与回报负相关。故以桥梁总造价的负值作为回报。

基于公式（1）、（2）、（3）和工程经验，假设桥梁单位面积上、下部结构造价计算公式为（单位：元每平方米）：①混凝土结构：上部结构单价为 $250+40x^{1.2}$，下部结构单价为 $50000x^{-0.5}$。②钢混组合结构：上部结构单价为 $500+90x^{1.07}$，下部结构单价为 $45000x^{-0.5}$。③钢结构：上部结构单价为 $2000+140x^{1.0}$，下部结构单价为 $40000x^{-0.5}$。（注：手工依据公式（6）计算，混凝土结构、钢混组合结构、钢结构的经济跨径与对应桥梁单位面积总造价分别为 39.6 米/11501 元每平方米、32.3 米/12125 元每平方米、27.3 米/13478 元每平方米。取三者造价最低值，本环境的经济跨径是 40 米/混凝土结构，对应于环境第 0 行、第 3 列单元格，这就是智能体的目的地。如果智能体在环境中游荡，总是能够花费较少的步骤到达和停留在这个目的地，那么就代表学习成功。）

相关代码见图 3。

```python
def step(self, action):
    number_of_rows, span=self.state_to_grid(self.state)
    next_number_of_rows = number_of_rows + self.action_pos_dict[action][0]
    next_span = span + self.action_pos_dict[action][1]
    if next_number_of_rows < 0 or next_number_of_rows > 2:
        next_number_of_rows =number_of_rows
    if next_span < self.Min_length or next_span > self.Max_length:
        next_span =span
    self.state=self.grid_to_state(next_number_of_rows, next_span)
    self.info = {"number_of_rows(0=concrete、1=composite、2=steel)": next_number_of_rows,"span(m)":next_span}
    if next_number_of_rows==0:
        reward= (40*next_span**1.2+250)+(next_span**(1/2-1)*50000)
    elif next_number_of_rows==1:
        reward= (90*next_span**1.07+500)+(next_span**(1/2-1)*45000)
    elif next_number_of_rows==2:
        reward= (140*next_span**1+2000)+(next_span**(1/2-1)*40000)
    self.step_num += 1
    if self.step_num >= self.max_steps:
        self.done = True
    return self.state, -reward, self.done, self.truncated, self.info
```

图 3 环境 step()方法代码

Fig. 3 Code for step() method of Environment

3. reset()方法用于重置环境，给智能体随机分配一个初始位置（单元格）。state_to_grid()、grid_to_state()方法用于状态索引与行列坐标的转换。相关代码见图 4。

```python
def reset(self):
    self.state=random.choice(self.states)
    number_of_rows, span=self.state_to_grid(self.state)
    self.info = {"number_of_rows(0=concrete、1=composite、2=steel)": number_of_rows,"span(m)":span}
    self.step_num = 0
    self.done = False
    self.truncated= False
    return self.state, self.info

def state_to_grid(self,the_state):
    number_of_rows=the_state//len(self.state_pos_dict)
    span=self.state_pos_dict[the_state%len(self.state_pos_dict)]
    return number_of_rows, span

def grid_to_state(self,number_of_rows, span):
    the_state=np.where(self.state_pos_dict == span)[0][0]+number_of_rows*80
    return the_state
```

图 4 环境 reset()、state_to_grid()、grid_to_state()方法代码

Fig. 4 Code for reset(), state_to_grid(), grid_to_state() method of Environment

4. render()、gridarray_to_image()方法用于渲染状态图像，详见源代码。

2.2 建立智能体和训练

建立和训练 DQN 智能体是常规操作，市面上强化学习书籍均有详细阐述[3-4]，故这里仅简要介绍，相关细节详见源代码。

1. 建立卷积神经网络

输入为状态图像，输出为该状态下执行各种动作的 Q 值。

卷积基由 2 个 Conv2D 层堆叠而成，过滤器数量依次为 16、32。采样窗口尺寸为 4×4、步幅为 4、激活函数为 relu。

分类器由 2 个 Dense 层堆叠而成，神经元数目依次为 128、5，激活函数依次为 relu、linear。

表 1 卷积神经网络的模型摘要

Tab.1 Model summary of convolutional neural network

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 12, 320, 16)	784
conv2d_1 (Conv2D)	(None, 3, 80, 32)	8224

flatten (Flatten)	(None, 7680)	0
dense (Dense)	(None, 128)	983168
dense_1 (Dense)	(None, 5)	645
Total params: 992,821		
Trainable params: 992,821		
Non-trainable params: 0		

2. 采用贪婪策略来选择动作：以概率 ε 选择随机动作，以概率（1- ε ）选择具有最大 Q 值的动作。

3. 用队列方式建立经验回放缓存，用于保存当前状态、动作、回报、下一个状态的数据。

4. 用经验回放缓存中的数据，训练卷积神经网络。标签是环境反馈的总回报，由即时回报和下一个状态价值（自举）组成。训练的目的就是让模型预测值吻合标签。损失函数采用均方误差（Mean Square Error，MSE），优化器采用“Adam”。最终卷积神经网络能够为每个状态输出正确的 Q 值。训练损失曲线见下图（前 100 轮）：

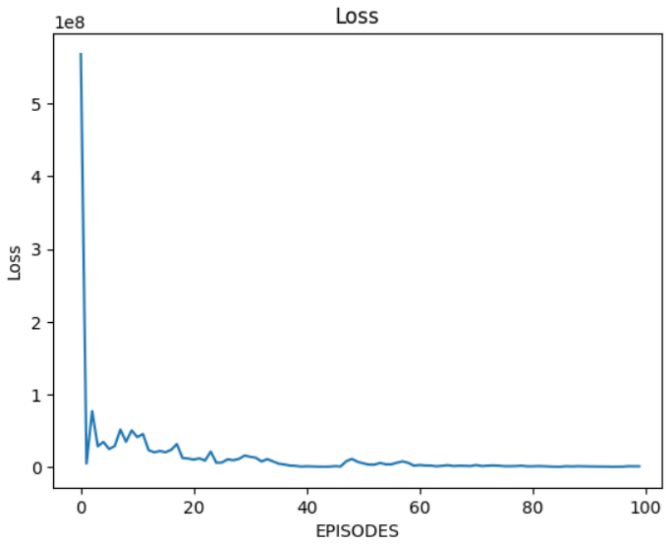

图 5 训练损失曲线

Fig.5 Training loss curve

2.3 测试

Q 值是动作的总回报。一个训练成功的智能体，在每个状态下，执行具有最大 Q 值的动作，就能够成功实现目标。对于本文环境，无论智能体被初始分配到哪个单元格，此时都应该能够较快运动到第 0 行、第 3 列单元格（经济跨径）。测试结果如下图所示（图中红色单元格为智能体初始位置，蓝色单元格为智能体运动轨迹，绿色单元格为智能体的运动终点）：

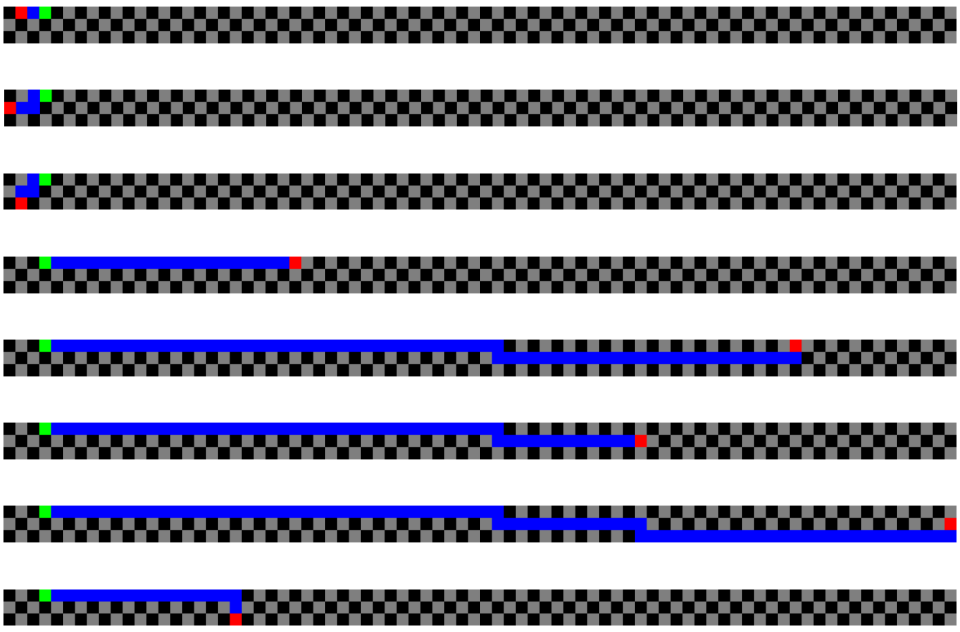

图 6 智能体策略测试

Fig.6 Policy testing of agent

上图显示，智能体的终点均是第 0 行、第 3 列单元格（经济跨径）。可见经过训练的智能体，能够成功进行桥梁经济跨径选择。

3 结语

本文采用导数求函数极值方法，推导出经济跨径的理论求解公式。

通过构建桥梁模拟环境并应用深度 Q 网络算法，成功实现了深度强化学习在桥梁经济跨径选择中的应用。研究结果表明，智能体通过学习，能够快速有效地识别出成本最低的桥梁跨径，为桥梁设计领域提供了新的视角，也为强化学习在工程问题中的应用提供了有益的探索。

本次尝试存在一些局限性，如造价函数、桥梁模拟环境未考虑桥梁结构形式的变化，以及智能体训练耗时过长，未来可进一步深入优化。

参考文献

[1] 邵旭东.桥梁工程[M].第五版.北京:人民交通出版社股份有限公司,2019 年.

[2] 黄秀全,李军.浅谈桥梁经济跨径的选择[J].辽宁交通科技,1996,(04):33-34.

[3] Richard S.Sutton, Andrew G.Barto.强化学习[M].第二版.北京:电子工业出版社,2019 年.

[4] 肖智清.强化学习原理与 Python 实现[M].北京:机械工业出版社,2019 年.

[5] 程国忠,周绪红,刘界鹏,王禄锋.基于深度强化学习的高层剪力墙结构智能设计方法[J].建筑结构学报,2022,43(9):84-91.

[6] 何佳琛.基于强化学习的大跨度桥梁风致振动主动控制研究[J].交通科技,2023,(06):18-24.

[7] 袁泉,曾文驱,李子涵,等.基于改进型 D3QN 深度强化学习的铁路智能选线方法[J].铁道科学与工程学报,2022,19(02):344-350.DOI:10.19713/j.cnki.43-1423/u.t20210179.

[8] 罗睿锋.基于强化学习的桁架结构生成式设计算法研究[D].同济大学,2022.DOI:10.27372/d.cnki.gtjsu.2022.000615.

[9] Cheng M, Frangopol DM. A decision-making framework for load rating planning of aging bridges using deep reinforcement learning[J]. Journal of Computing in Civil Engineering. 2021 Nov 1;35(6):04021024.

[10] Yang DY. Deep Reinforcement Learning‐Enabled Bridge Management Considering Asset and Network Risks[J]. Journal of Infrastructure Systems. 2022 Sep 1;28(3):04022023.