

基于分类 DQN 的建筑能耗预测^①



李可^{1,2}, 傅启明^{1,2}, 陈建平^{2,3,4}, 陆悠^{1,2}, 王蕴哲^{1,2}, 吴宏杰^{1,2}

¹(苏州科技大学 电子与信息工程学院, 苏州 215009)

²(苏州科技大学 江苏省建筑智慧节能重点实验室, 苏州 215009)

³(苏州科技大学 建筑与城市规划学院, 苏州 215009)

⁴(重庆工业大数据创新中心有限公司, 重庆 400707)

通信作者: 傅启明, E-mail: fqm_1@126.com; 陈建平, E-mail: alanjpchen@aliyun.com

摘要: 本文提出一种可用于建筑能耗预测的基于 KNN 分类器的 DQN 算法——K-DQN. 其在利用马尔科夫决策过程对建筑能耗进行建模时, 针对大规模动作空间问题, 将原始动作空间缩减进而提高算法的预测精度及收敛速率. 首先, K-DQN 将原始动作空间平均划分为多个子动作空间, 并将每个子动作空间对应的状态分为一类, 以此构建 KNN 分类器. 其次, 利用 KNN 分类器, 将不同类别相同次序动作进行统一表示, 以实现动作空间的缩减. 最后, K-DQN 将状态类别概率与原始状态相结合, 在构建新状态的同时, 帮助确定缩减动作空间内每一动作的具体含义, 从而确保算法的收敛性. 实验结果表明, 文章提出的 K-DQN 算法可以获得优于 DDPG、DQN 算法的能耗预测精度, 且降低了网络训练时间.

关键词: 分类; 能耗预测; 动作空间; 深度强化学习

引用格式: 李可, 傅启明, 陈建平, 陆悠, 王蕴哲, 吴宏杰. 基于分类 DQN 的建筑能耗预测. 计算机系统应用, 2022, 31(10): 156-165. <http://www.c-s-a.org.cn/1003-3254/8692.html>

DQN Based on Classifier for Building Energy Consumption Prediction

LI Ke^{1,2}, FU Qi-Ming^{1,2}, CHEN Jian-Ping^{2,3,4}, LU You^{1,2}, WANG Yun-Zhe^{1,2}, WU Hong-Jie^{1,2}

¹(School of Electronic and Information Engineering, Suzhou University of Science and Technology, Suzhou 215009, China)

²(Jiangsu Province Key Laboratory of Intelligent Building Energy Efficiency, Suzhou University of Science and Technology, Suzhou 215009, China)

³(School of Architecture and Urban Planning, Suzhou University of Science and Technology, Suzhou 215009, China)

⁴(Chongqing Industrial Big Data Innovation Center Co. Ltd., Chongqing 400707, China)

Abstract: This study proposes a deep Q-network (DQN) algorithm based on the K-nearest neighbor (KNN) algorithm (K-DQN) for the energy consumption prediction of buildings. When using the Markov decision process to model the energy consumption of buildings, the K-DQN algorithm shrinks the original action space to improve the prediction accuracy and convergence rate considering large-scale action space problems. Firstly, the original action space is evenly divided into multiple sub-action spaces, and the corresponding state of each sub-action space is regarded as a class to construct the KNN algorithm. Secondly, actions of the same sequence in different classes are denoted by the KNN algorithm to shrink the original action space. Finally, state class probabilities and original states are combined by K-DQN to construct new states and help determine the meaning of each action in the shrunken action space, which can ensure the convergence of the K-DQN algorithm. The experimental results indicate that the proposed K-DQN algorithm can achieve higher prediction accuracy than deep deterministic policy gradient (DDPG) and DQN algorithms and take less network training time.

Key words: classification; energy consumption prediction; action space; deep reinforcement learning

① 基金项目: 国家重点研发计划 (2020YFC2006602); 国家自然科学基金 (61876121, 61876217, 62072324); 江苏省重点研发计划 (BE2020026); 江苏省高校自然科学基金 (21KJA520005)

收稿时间: 2021-12-17; 修改时间: 2022-01-18; 采用时间: 2022-01-30; csa 在线出版时间: 2022-07-15

近年来,建筑已逐渐成为全球能耗占比最高的领域^[1],建筑能耗的增加不仅提升了二氧化碳的排放量,还加剧了能源危机及温室效应。因此,提升建筑能效、实现节能减排,进一步实现建筑碳中和是社会发展的重要任务。建筑能耗预测作为提升建筑能效的重要组成部分,在提高电网效率^[2],改善需求响应^[3],评估建筑设计方案及建筑运营策略等方面发挥着重要作用^[4]。

建筑能耗预测的解决方法可大致分为两类:一类是工程方法,另一类是数据驱动方法。其中,工程方法也称白盒方法,其利用建筑传热原理和物理学方程对建筑系统进行分析建模,从而计算建筑的能源消耗^[5]。然而,这种方法需要详细的建筑结构信息和环境参数。一旦仿真设备参数不够准确,将会导致较高的预测误差^[6]。相比之下,数据驱动方法无需了解建筑内部信息,可在使用历史数据的情况下进行能耗预测。不仅如此,先前的研究已经表明,数据驱动方法可借助其解决非线性问题的能力获得优于工程方法的预测精度^[7]。

一般而言,数据驱动方法可分为机器学习、深度学习两类。在机器学习领域, Li 等人^[8]采用支持向量机(support vector machine, SVM)构建了办公建筑冷负荷能耗预测模型,其预测结果的均方根误差和平均相对误差约为BP神经网络(back propagation neural network, BPNN)的50%,验证了SVM算法在能耗预测领域的优越性。Wang 等人^[9]使用集成模型随机森林(random forest, RF)进行每小时的建筑能耗预测,并将其与传统的机器学习算法回归树(regression tree, RT)及支持向量回归(support vector regression, SVR)进行比较。实验结果表明,RF可以取得更高的预测精度,具有更强的预测能力。文献[10]提出了一种基于决策树(decision tree, DT)的能耗预测模型,该方法可以准确分类并预测建筑能耗水平,同时自动排序影响建筑能耗的关键因素。与其他机器学习方法相比,该方法的竞争优势在于预测结果具备较强的可解释性。

然而,与深度学习相比,传统机器学习的特征提取能力稍显欠缺,其仅能挖掘出输入数据的浅层特征。而深度学习具有多个处理层,可将输入数据进行多次转换,获取数据内部潜藏的深层特征^[11]。因此,深度学习在建筑能耗预测领域受到了广泛关注。Fan 等人^[12]将无监督学习提取的特征作为网络输入,而后利用深度神经网络(deep neural networks, DNN)预测建筑冷负荷,发现可以显著提高预测精度。文献[13]则提出了一

种深度递归神经网络(recursive neural network, RNN)预测模型,以小时为单位预测办公建筑的能耗,与传统的多层感知器(multilayer perceptron, MLP)相比,其预测的相对误差较低。

深度强化学习作为深度学习的一个重要分支,其将深度学习的感知能力与强化学习的决策能力相结合,近年来在机器人,自动驾驶等方向都获得了快速发展^[14,15]。而在建筑领域,深度强化学习常被用作优化控制,例如Zhang 等人^[16]将深度强化学习算法用于HVAC系统控制,实现15%的供热节能。Kazmi 等人^[17]则利用深度强化学习控制建筑的空间加热,并通过实验结果证明其性能优于基于规则的控制方法。但深度强化学习的能力远不止于此,它可将能耗预测建模为高维连续状态空间中的优化问题进行求解,这吸引了众多研究者的关注。Liu 等人^[18]探究了深度确定性策略梯度(deep deterministic policy gradient, DDPG)算法与异步优势行动者评论家算法(asynchronous advantage actor-critic, A3C),并将其与BPNN, RF等算法进行比较,发现DDPG算法建筑能耗预测精度最高。Zhang 等人^[19]则在DDPG的基础上,添加了自适应早期预测方法及奖赏激励机制,进一步提高预测精度并确保算法的收敛性。但现有的研究多关注于DDPG算法,而忽视了经典的DQN。一个可能的原因是DQN算法只能处理离散动作空间问题。但和DDPG相比,其优势在于可以花费更少的训练时间。

为在少于DQN算法训练时间的情况下,取得优于DDPG算法的预测精度,本文提出一种K-DQN算法框架。其将深度Q网络(deep Q-network, DQN)算法与K近邻(K-nearest neighbor, KNN)分类器相结合,构建缩减动作空间进行能耗预测。具体而言, K-DQN算法将预测下一时刻的能耗转换为动作选择的形式,并针对动作空间大而导致的算法收敛慢及精度低的问题,将原始动作空间平分为多个子动作空间,并将每个子动作空间对应的状态全部划分为一类,以此训练KNN分类器。而后在借助KNN分类器的基础上,对不同类别相同次序的动作进行统一表示,从而将原始动作空间映射至单一子动作空间,实现对原始动作空间的缩减,使算法更易寻找最优值。此外,考虑到缩减的动作空间内每个动作具有多重含义,故将原始状态及其对应的类别概率相结合,构建出新状态作为网络输入,帮助确定动作含义并提高算法的收敛速率及预测

精度. 实验结果表明, K-DQN 算法在建筑能耗预测方面优于 DQN、DDPG 算法, 且花费更少的训练时间.

1 理论基础

1.1 深度强化学习

1.1.1 强化学习

强化学习是机器学习领域的一个重要分支, 它与监督学习, 无监督学习不同, 其目标是最大化 Agent 获得的累积折扣奖励的期望^[20]. 通常, 强化学习问题可以被建模为马尔科夫决策过程进行求解. 一个马尔科夫决策过程可以表示为一个五元组 (S, A, P, R, γ) , 其中 S 表示状态集合, A 表示动作集合, P 为转移函数, R 为奖励函数, γ 为折扣因子. 五元组可看作是 Agent 与环境进行交互产生的结果. 在一次交互过程中, Agent 观察到状态 s_t , 根据策略 π 选择动作 a_t , 并通过转移函数 $P(s_{t+1}|s_t, a_t)$ 转移到状态 s_{t+1} , 获得立即奖励 r_{t+1} . 此处, 策略 π 的具体含义为状态空间 S 到动作空间 A 的映射, $P(s_{t+1}|s_t, a_t)$ 则表示 Agent 在状态 s_t 下执行动作 a_t , 转移到下一状态 s_{t+1} 的概率.

Agent 的最终目标是最大化累积折扣奖励的期望, 而这一目标可通过找寻最优策略实现. 策略 π 可利用动作价值函数与最优动作价值函数进行评估与改进, 从而产生最优策略 π_* , 实现最终目标. 其中, 动作价值函数 Q 表示在状态 s 下执行动作 a 的累计折扣奖励的期望:

$$Q_{\pi}(s, a) = E_{\pi} \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | s_t = s, a_t = a \right] \quad (1)$$

最优动作价值函数则表示在状态 s 下执行最优动作时的累计折扣奖励的期望为:

$$\begin{aligned} Q_*(s, a) &= \max_{\pi} Q_{\pi}(s, a) \\ &= E \left[R_{t+1} + \gamma \max_{a'} Q_*(s_{t+1}, a') | s_t = s, a_t = a \right] \quad (2) \end{aligned}$$

1.1.2 深度 Q 网络

传统的强化学习算法, 例如 Q-learning, SARSA^[21,22], 只能解决状态空间较小或离散的相关问题. 虽然采用函数逼近等方式可以打破这一限制, 但这些方法通常不够稳定, 不易收敛^[23]. 而深度 Q 网络利用经验回放机制和目标网络解决了这一问题.

经验回放机制, 即将 Agent 与环境交互的数据样本存入经验池中, 而后从经验池随机采样训练神经网络. 这种方式破坏了输入数据之间的相关性, 使神经网络更易收敛. 其次, DQN 新构建目标 Q 网络用于 Q 网络的更新. 具体而言, 在网络第 i 次迭代时, Q 网络的 loss 函数为:

其中, (s, a, r, s') 为一个经验池样本, 目标 Q 网络的输出值为 $\hat{Q}(s', a' | \theta_i^-)$, Q 网络输出值为 $Q(s, a | \theta_i)$. 目标 Q 网络的引进一定程度上降低了 Q 值和目标 Q 值的相关性, 提升了 DQN 算法的稳定性. 此外, 目标 Q 网络的更新依赖于 Q 网络参数. 每过 J 步, Q 网络的所有参数将复制到目标 Q 网络中, 完成目标 Q 网络的更新.

$$L(\theta_i) = E \left[(r + \gamma \max_{a'} \hat{Q}(s', a' | \theta_i^-) - Q(s, a | \theta_i))^2 \right] \quad (3)$$

1.2 K 近邻算法

KNN 是一种基本的机器学习方法, 该算法用作分类时, 以样本的特征向量为输入, 输出样本的类别. 其基本原理如下: 对于给定的新样本, 计算其与所有训练样本的距离, 而后找出距离最近的 K 个点投票得出分类结果. K 值不同, 其分类的最终结果也不同.

1.2 K 近邻算法

K 值选择的问题本质是一个超参数问题. 选择较小 K 值时, 算法的近似误差会降低, 但估计误差会增大. 同时, 算法结果对近邻的样本点会非常敏感^[24]. 一旦近邻样本为噪声, 分类结果就会出错. 而选择较大 K 值时, 算法的估计误差会减少, 但近似误差会增大. 与样本点相关性较弱的其他样本可能也会参与投票, 降低分类结果的正确性. 故在实际应用中, 常采用交叉验证法选择最优 K 值.

2 基于 K-DQN 算法的能耗预测模型

2 基于 K-DQN 算法的能耗预测模型

2.1 K-DQN 算法的能耗预测模型

图 1 展示了 K-DQN 算法进行建筑能耗预测的模型框架. 在数据预处理阶段, 能耗数据按日期划分为训练集和测试集, 并采用异常值检测方法对训练集数据进行异常值检测. 同时, 在考虑能耗数据日期属性及所属时刻两种因素的前提下, 采用均值法对异常能耗数据进行替换. 特征提取是选取一定数量的历史能耗数据作为特征的过程. 可将待预测时刻的能耗当作标签, 待预测时刻前的一定数量的历史能耗当作特征, 对替换异常值后训练集数据进行重构, 生成新的样本及标签.

在训练过程中, 归一化后的新样本首先传入 K-DQN 模块中的 KNN 分类器对其进行训练. 待 KNN 分类器训练完成后, 该样本重新输入至分类器获取分类

概率. 此时, 再将分类概率与归一化后的新样本进行合并, 构建出 t 时刻的状态 s_t . Q 网络将 s_t 看作输入, 计算动作空间下所有动作的 Q 值. 同理, 利用分类概率和归一化样本构建状态 s_{t+1} 传入目标 Q 网络, 计算所有动作的目标 Q 值. 利用目标 Q 值与 Q 值构建损失函数进行 Q 网络的更新.

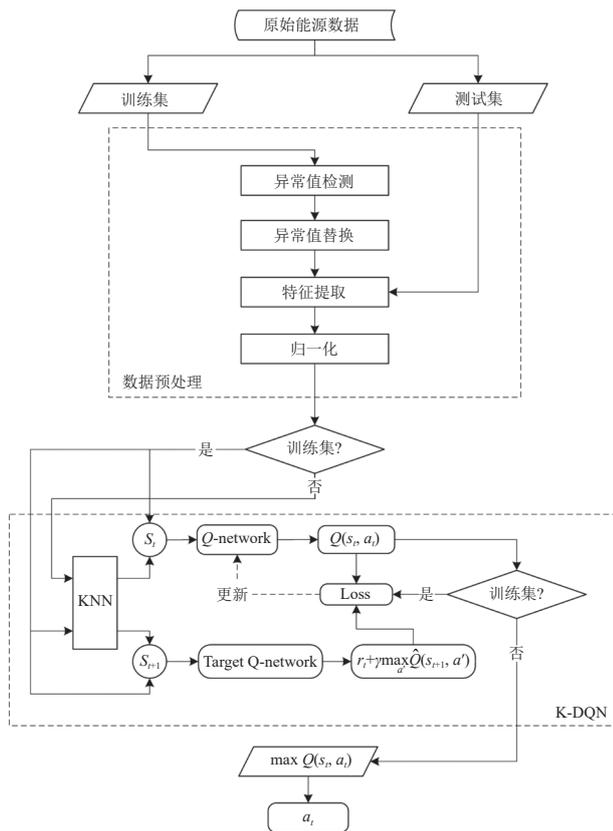


图1 K-DQN 算法能耗预测模型框架

在测试过程中, 归一化后的新样本首先传入训练完成的 KNN 分类器获取分类概率, 而后与分类概率组成新状态传入 Q 网络. Q 网络计算出动作空间下的所有动作的 Q 值, 选取最优动作作为预测的能耗.

2.2 数据预处理

研究采用上海某一办公建筑能耗数据, 数据采集范围为 2015.1.1 至 2016.12.31, 采集频率为 1 h/次. 此外, 由于 2016.2.29 全天数据缺失, 故总数据个数为 17520.

由于建筑内部可能存在电表混用或空调末端接入插座的现象, 故需采用局部异常因子 (local outlier factor, LOF) 算法和均值法对能耗数据分别进行异常值的检测和替换, 以提高能耗预测的准确性. LOF 是基于

密度的高精度离群点检测算法^[25], 其计算每个样本对其邻域的局部密度偏差 (即 LOF 值), 以发现可能的异常值. LOF 值计算公式如式 (4):

$$LOF_k(P) = \frac{\sum_{O \in N_k(P)} \frac{\rho_k(O)}{\rho_k(P)}}{|N_k(P)|} \quad (4)$$

其中, $\rho_k(P)$ 表示点 P 的局部可达密度, 即点 P 第 k 邻域内的所有点到 P 点的平均可达距离. $N_k(P)$ 表示点 P 的第 k 距离邻域.

直接丢弃异常数据会破坏能耗数据的时序周期性, 故采用均值法进行异常值的替换. 异常值的替换考虑日期和时刻两种因素. 具体而言, 若工作日的能耗数据存在异常, 则需将前一个工作日相同时刻的正常能耗和后一个工作日相同时刻的正常能耗求和取平均, 替换异常数据. 若节假日的能耗数据存在异常, 则利用节假日相同时刻的正常能耗进行替换. 此外, 研究设定训练集数据的日期范围为 2015.1.1 至 2016.10.31, 即异常值的检测和替换操作的数据均在此日期范围, 不使用测试集的任何信息.

替换异常值后的训练集数据需进行特征提取构建新样本和标签, 方能传入 K-DQN 模块. 例如, 若预测 t 时刻能耗, 则选择 $t-h$ 至 $t-1$ 时刻 h 个历史能耗数据作为特征. 此时 $(\tilde{E}_{t-h}, \tilde{E}_{t-h-1}, \dots, \tilde{E}_{t-1})$ 看作样本, E_t 看作其对应的标签. 在训练集数据总数为 M 时, 可将训练集重构为 $M-h$ 个新样本和标签. 同时, 为了降低不同特征间的影响, 提高能耗预测精度, 对新样本按特征进行归一化操作:

$$\tilde{X}_i^{(j)} = \frac{X_i^{(j)} - \mu^{(j)}}{\sigma^{(j)}} \quad (5)$$

其中, $X_i^{(j)}$ 和 $\tilde{X}_i^{(j)}$ 分别表示第 i 样本 j 特征的原始值和归一化值, $\mu^{(j)}$ 和 $\sigma^{(j)}$ 分别表示 j 特征的均值和方差.

2.3 MDP 建模

使用 K-DQN 进行能耗预测时, 需将能耗预测问题转化为 MDP 过程进行求解, 并对状态, 动作, 奖赏等进行设置.

K-DQN 算法的 MDP 建模本质上是 DQN 算法建模的一种改进. 在使用 DQN 进行能耗预测时, 状态由一定数量的历史能耗数据组成, 动作则根据训练集能耗数据范围进行设置. 最低的能耗值可看作第一个动作, 而最高的能耗值为最后一个动作. 具体而言, 假设

历史能耗数据范围为 $[x, z]$, 在步长为 g 的情况下, 其动作空间大小为 $\{x, x+g, x+2g, \dots, z\}$, 动作总个数为 $(z-x)/g+1$.

而 DQN 算法的 MDP 建模缺陷在于, 一旦动作空间增大, 动作总个数则随之增长. Q 网络需计算更多动作的 Q 值, 从而影响该算法进行能耗预测的精度和收敛速度. 相比之下, K-DQN 算法采用缩减动作空间的方式进行 MDP 建模, 通过减少原有动作空间大小及动作总个数, 提升能耗预测的精度及收敛速度. 现以能耗数据范围 $[10, 59]$, 步长为 1 为例进行说明. 如图 2 所示, 原始动作空间 X 共包含 50 个动作, 且其内部动作值等同于 Agent 预测的能耗值. 实际上, 原始动作空间 X 的动作值均可写作在某一动作值基础上进行增加的形式. 转换结果如图 2 中间部分, 此时动作空间 X 转换为动作空间 Y , 且动作空间大小未发生改变.

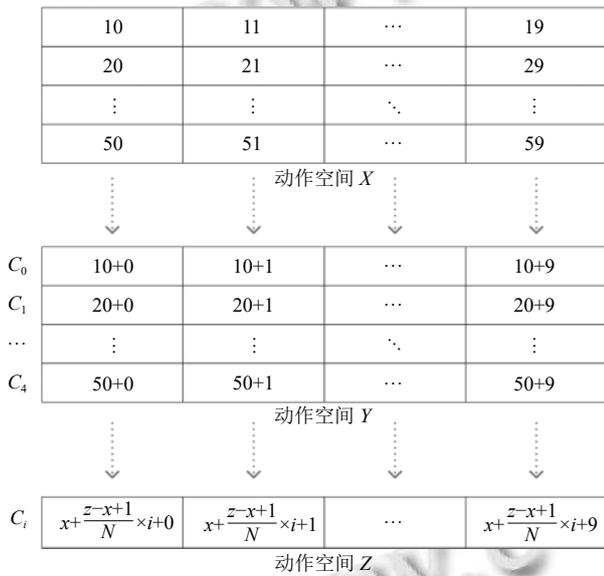


图 2 缩减动作空间示例

对动作空间 Y 而言, 每行可看作一个子动作空间. 若将每一子动作空间对应的状态划分为一类, 则动作空间 Y 对应的所有状态被划分为 5 类, 即 C_0, C_1, C_2, C_3, C_4 . 此时, 不同类别相同次序的动作可表示为:

$$x + \frac{z-x+1}{N} \times i + j, \quad i = 0, 1, 2, 3, 4, \quad j = 1, \dots, 9 \quad (6)$$

其中, N 表示状态类别总个数, i 表示第 i 类状态, j 表示第 j 个动作, x, z 为能耗数据范围的最低值 10 和最高值 59. 由此动作空间 Y 可被动作空间 Z 替代, 总动作个数由 50 减少至 10. 更进一步考虑步长为 g 时的情况, 此时

不同类别相同次序的动作可表示为:

$$x + \frac{z-x}{N} \times i + j, \quad i = 0, 1, \dots, N-1, \quad j = 0, g, 2g, \dots, n \quad (7)$$

最终, $N \times n$ 个总动作可减少至 n , 实现原始动作空间的缩减.

此外, 由于缩减后的动作空间是由不同类别相同次序的动作组合而成, 因此每个动作具有多重含义. 例如, 缩减后的动作空间第一个动作可表示每一个子动作空间的第一个动作. 映射至神经网络中, 则意味着一个神经元具备多种含义, 这影响了 Q 网络的收敛速率及算法的预测精度.

为解决上述问题, 我们将类别概率与归一化后的新样本进行组合, 构造出状态 s_t 传入至 Q 网络中. 在训练过程中, 状态类别概率可以确定当前状态类别, 从而将缩减后的动作空间映射至唯一确定的子动作空间. 如图 3 所示, 当状态类别为 C_0 时, 缩减后的动作空间可看作是第一个子动作空间, 从而确定每一动作的具体含义, 提高 Q 网络的收敛速率及算法预测精度.

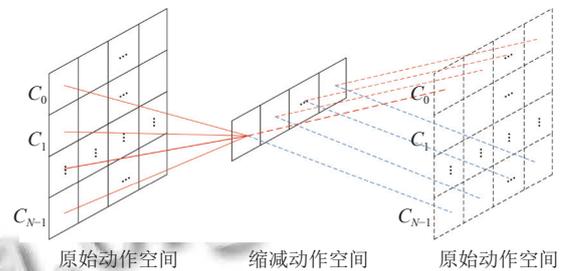


图 3 缩减动作空间与原始动作空间的映射

K-DQN 算法利用缩减的动作空间进行能耗预测时, 状态 s_t 由归一化样本及其对应的各类别的概率组成, 表示为 $(P_0, P_1, \dots, P_{N-1}, \tilde{E}_{t-h}, \tilde{E}_{t-h-1}, \dots, \tilde{E}_{t-1})$. 其中, $(\tilde{E}_{t-h}, \tilde{E}_{t-h-1}, \dots, \tilde{E}_{t-1})$ 表示归一化处理后的历史能耗数据样本, P_i 表示该样本被判断为 i 类的概率. 动作 a_t 表示 Agent 在 t 时刻的预测能耗值, r_t 为 t 时刻 Agent 获得的立即奖赏, 具体表示为:

$$r_t = -|E_t - a_t| \quad (8)$$

其中, E_t 表示 t 时刻的真实能耗. 由式 (8) 可知, 奖赏越接近 0, Agent 预测能耗的误差越低.

2.4 K-DQN 算法流程

将能耗预测问题建模为 MDP 模型后, 便可利用

K-DQN 算法进行建筑能耗的预测. 具体来说, 首先按照日期将数据集划分为训练集和测试集, 并对其进行异常值的检测和替换. 而后根据训练集能耗范围及设定的状态类别总数 N , 将原始动作空间的动作总个数缩减 N 倍, 构建缩减的动作空间. 其次, 对训练集数据进行特征提取及归一化, 生成新样本及标签训练 KNN 分类器.

K-DQN 算法在训练时, 每回合从经验池中随机选择样本作为原始状态, 同时利用 KNN 分类器获取该样本的各类别概率, 并将二者结合构建新状态 s_t 传入 Q 网络. 同理, 构建新状态 s_{t+1} 传入目标 Q 网络. 而后利用两个网络各自输出的结果, 构建损失函数 $(y_j - Q(s_j, a_j; \theta))^2$ 更新 Q 网络, 直至其收敛. 算法 1 详细描述了此训练过程.

算法 1. K-DQN 算法进行能耗预测

- 1) 初始化状态类别总数 N
- 2) 初始化经验池 D
- 3) 初始化 Q 网络和目标网络 Q 参数 $\theta, \theta^- = \theta$
- 4) 按日期划分数据集
- 5) 对训练集数据进行异常值检测和替换
- 6) 构建缩减的动作空间
- 7) 特征提取及归一化
- 8) 训练 KNN 分类器
- 9) 每一回合:
 - 10) 随机选择样本
 - 11) 使用 KNN 分类器获取当前样本的各类别概率
 - 12) 构建新状态 s_t
 - 13) 每一步:
 - 14) 在缩减的动作空间中, ε 概率选择随机动作, $1-\varepsilon$ 概率选择 $\max(s_t, a; \theta)$
 - 15) 执行所选动作获取立即奖赏 r_t
 - 16) 构建新状态 s_{t+1}
 - 17) 将 (s_t, a_t, r_t, s_{t+1}) 存入经验池 D
 - 18) 从经验池中采样 (s_j, a_j, r_j, s_{j+1})
 - 19) 计算 $y_j = \begin{cases} r_j, & j+1 \text{ 步结束} \\ r_j + \gamma \max_{a'} \bar{Q}(s_{j+1}, a'; \theta^-), & \text{否则} \end{cases}$
 - 20) 利用 $(y_j - Q(s_j, a_j; \theta))^2$ 更新 Q 网络
 - 21) 每 J 步设置 $\hat{Q} = Q$
 - 22) $s_t \leftarrow s_{t+1}$
 - 23) Until 最终状态 or 最大步数
 - 24) Until 最大回合数

3 实验仿真

在本节中, 我们将采用 DQN, K-DQN, DDPG 算法进行建筑能耗预测的分析与比较. 第 4.1 节描述了 3 种算法的基本实验设置. 第 4.2 节选取了 3 种评价指标比较不同算法的预测结果. 第 4.3 节从预测精度, 收敛速率, 训练时间 3 个角度分析 3 种算法的优劣.

3.1 基本实验设置

在特征提取阶段, 实验采用前 24 个时刻历史能耗数据作为特征, 预测下一时刻的能耗. 由此 3 种算法的超参数设置如下: DQN, DDPG 算法输入层神经元个数为 24, K-DQN 输入层神经元个数为 $24+N$ (N 为状态类别总数). 此外, 由于训练集能耗数据范围为 $[24, 2703]$, 故 DQN, K-DQN 算法采用步长为 1 的方式, 将连续动作空间转换为离散动作空间进行能耗的预测. 同时, 为保证算法比较的公平性, 3 种算法均使用相同学习率及相同网络架构. 对输出层而言, DQN, K-DQN 算法的输出层神经元个数取决于动作总个数, 而 DDPG 算法由于自身处理连续问题的性质, 需将其设置为 1. 表 1 详细列举了 3 种算法所使用的超参数.

表 1 超参数设置

算法	参数	取值
DQN	神经元个数	24, 32, 32, 2680
	激活函数	ReLU
	学习率	0.001
K-DQN	神经元个数	$24+N$, 32, 32, 2680/ N
	激活函数	ReLU
	学习率	0.001
DDPG	神经元个数 (Actor)	24, 32, 32, 1
	激活函数 (Actor)	ReLU
	学习率 (Actor)	0.001
	神经元个数 (Critic)	24, 32, 32, 1
	激活函数 (Critic)	ReLU
	学习率 (Critic)	0.001

此外, 由于 K-DQN 算法内部使用 KNN 分类器, 且 K 值选择问题是超参数问题, 故实验采用 6 折交叉验证选取最优 K 值.

3.2 评价指标

为比较 3 种算法的预测精度, 选取平均绝对误差 (mean absolute error, MAE), 均方根误差 (root mean square error, RMSE), 决定系数 (coefficient of determination), 作为评价指标. 其计算公式分别为:

$$MAE = \frac{1}{m} \sum_{i=1}^m |y_i - y'_i| \quad (9)$$

$$RMSE = \sqrt{\frac{1}{m} \sum_{i=1}^m (y_i - y'_i)^2} \quad (10)$$

$$R^2 = 1 - \frac{\sum_{i=1}^m (y_i - y'_i)^2}{\sum_{i=1}^m (y_i - \bar{y})^2} \quad (11)$$

其中, m 表示样本总个数, y_i 和 y_i' 表示第 i 样本的真实值及预测值, \bar{y} 为整体样本的均值。

3.3 结果与分析

研究使用的实验结果均建立在表1超参数设置的基础上,且为降低训练过程中其他随机因素的影响,每一实验结果都取自10次实验结果的平均值。

图4展示了DQN, DDPG以及不同类别总数情况下K-DQN算法的整体预测情况。其中,横轴表示算法的预测能耗值,纵轴表示真实能耗。蓝色实线为经过原点的斜率为1的直线,代表预测值和真实能耗值相同。故图像中的散点越靠近该直线,则说明算法的预测精度越高。此外,蓝色虚线是20%的误差线,靠近纵轴的蓝色虚线意味着预测能耗值是真实能耗值80%,靠近横轴的蓝色虚线则表示预测能耗值是真实能耗的120%。二者组成的阴影部分为预测值和真实值相差20%的误差带。从这个角度看,DQN算法在真实能耗值低于500时有较多的预测点处于误差带之外,说明预测精度较差。相比之下,DDPG算法的预测结果整体偏向于左上方,说明该算法的整体预测值均偏小。对K-DQN算法而言,由于其算法内部使用KNN分类器对状态进行了划分,故预测出的能耗点呈现出一定的分类趋势。例如状态类别总数为6和8的K-DQN中,有较为明显的簇。

表1和表2描述了3种算法的具体预测精度。从MAE角度看,DQN算法预测精度最低,为51.566,而状态类别总数为7的K-DQN算法预测精度最高,其MAE值为39.852。DDPG与状态类别总数为14的K-DQN算法预测精度相近,其预测结果的MAE值均为44以上。在比较RMSE指标的情况下,不同类别总数的K-DQN预测精度均高于DQN, DDPG算法,其整体RMSE值处于70以下。类别总数为3时的K-DQN是个特例,其原因可能是测试过程中KNN分类器对新样本分类错误导致。 R^2 指标反映了算法模型的拟合精度,越靠近1则说明算法的预测精度越高。由表1、表2可知,K-DQN算法的预测精度均高于DQN、DDPG算法,且类别数为6、7的K-DQN算法预测精度最高。

此外,对K-DQN算法而言,类别数越多,动作空间总动作个数越低,算法的预测精度越高。但由于其内部分类器的分类准确率会随类别数的增加而逐渐降低,

从而影响算法预测精度,故表3中K-DQN的预测精度整体呈先上升后下降的趋势。综合来看,类别总数为7的K-DQN模型效果最优,预测精度最高。

图5展现了3种算法训练过程中的MAE变化趋势。其中,DQN算法的收敛速度最慢,且收敛后的MAE值高于DDPG及K-DQN算法。此外,由于DQN及 $N=2$ 时的K-DQN算法起始MAE值较高,以致图5不易比较分析DDPG及其他分类总数下的K-DQN算法。故将二者去除,生成图6。图6由第3回合开始进行展示,可以看出DDPG算法与类别总数为8的K-DQN算法收敛趋势相近,但收敛后的MAE值略高。对K-DQN算法而言,KNN分类使得K-DQN具有较低的MAE初始值,且状态类别数越多,K-DQN算法收敛速率越快,收敛后的MAE值越低。故该实验中,类别总数为14的K-DQN算法收敛速率最快。

表2和表3同时展示了3种算法的训练时间。其中,DDPG算法的训练时间最长,为1353.058 s。DQN算法训练时间其次,为1190.733 s。而K-DQN算法的训练时间则取决于状态类别总数。一般而言,状态类别总数越多,动作空间越小,网络训练所需的参数越少,训练时间越低,故K-DQN算法的训练时间随类别数增加整体呈下降趋势。但由于实验过程中随机因素无法完全消除,以致于出现个别偏差值,例如类别数为10时的K-DQN算法情况。

4 结论与展望

K-DQN是一种将KNN分类器与DQN相融合的算法框架,可在减少网络训练时间的情况下,获得优于DQN、DDPG算法的建筑能耗预测精度。该算法首先对原始动作空间进行平均划分,产生多个子动作空间,然后将每个子动作空间视作一类训练KNN分类器。并在此基础上,对不同类别相同次序的动作进行统一表示,从而将原始动作空间映射至单一子动作空间中,实现原始动作空间的缩减。缩减后的动作空间与原始动作空间相比,动作个数更少,算法更易寻找最优值,进而提升建筑能耗预测的精度。此外,K-DQN算法将历史能耗数据及其对应的分类概率组合构建为Q网络的输入,确定了单一动作在网络中的具体含义,提高了算法的收敛速率及预测精度。最终的实验结果证明,与DQN, DDPG算法相比,类别为7的K-DQN算法在预测精度,收敛速率,训练时间方面均取得更优结果,验

证了 K-DQN 算法的有效性. 此外, 由于 K-DQN 算法的本质是缩减动作空间, 故在建筑能耗范围较大时, 算

法特性会发挥得更加明显. 在后续工作中将对这一特性进行讨论说明, 提升 K-DQN 算法框架的通用性.

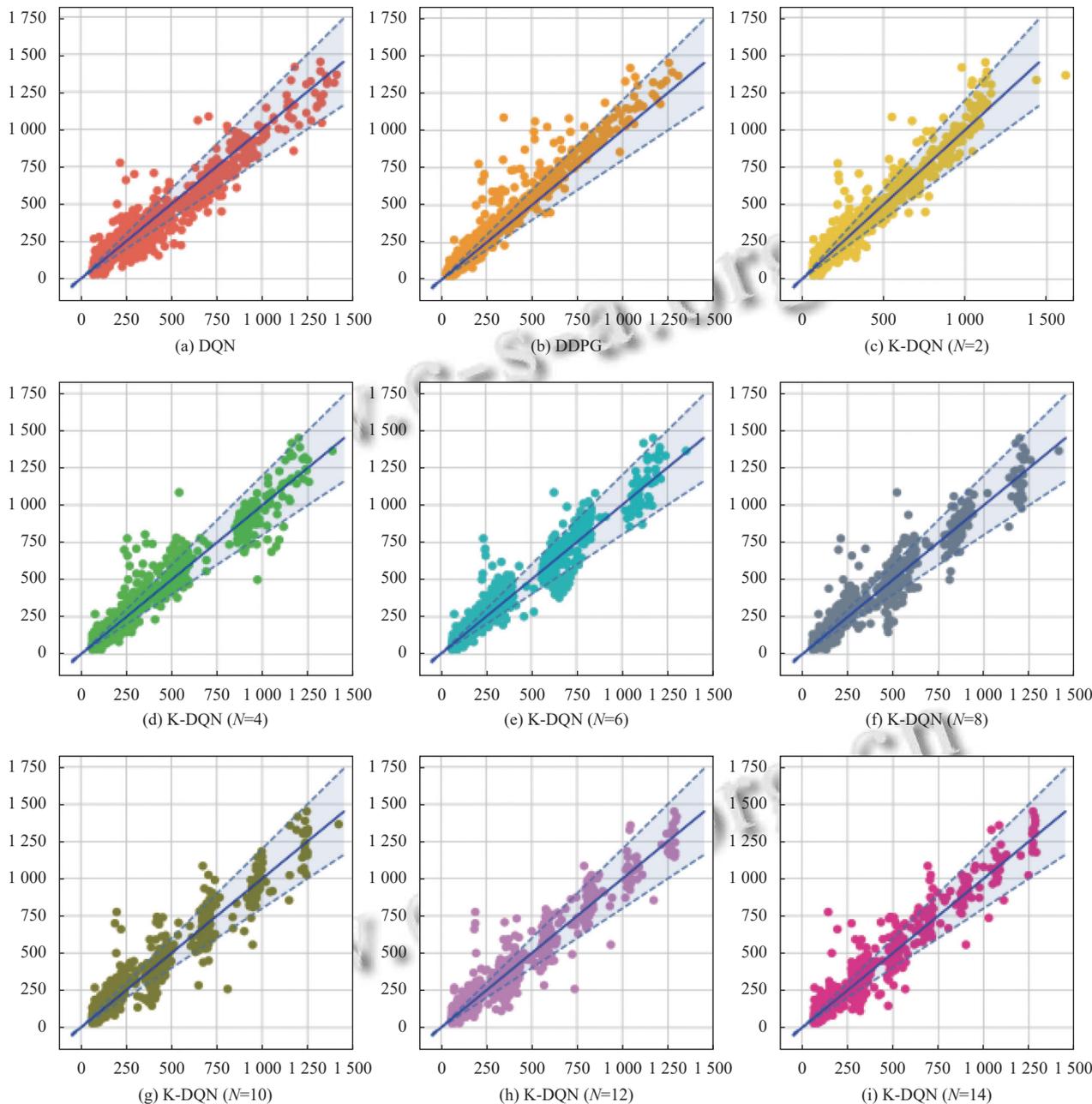


图 4 3 种算法预测结果 (其中, 横坐标表示算法的预测能耗值 (kWh), 纵坐标表示真实能耗 (kWh))

表 2 DQN 和 DDPG 算法的预测精度及训练时间

算法	MAE (kWh)	RMSE (kWh)	R^2	训练时间 (s)
DQN	51.566	74.346	0.921	1 190.733
DDPG	44.478	75.760	0.922	1 353.058

表3 K-DQN 算法的预测精度及训练时间

类别总数	动作总个数	K值	分类准确率 (%)	MAE (kWh)	RMSE (kWh)	R ²	训练时间 (s)
2	1340	6	99.727	43.779	66.757	0.924	1103.653
3	894	4	98.156	43.773	71.104	0.923	1025.076
4	670	11	97.268	41.228	67.581	0.925	985.108
5	536	15	95.970	42.783	69.185	0.924	957.143
6	447	7	94.809	41.663	66.992	0.926	847.062
7	383	8	93.716	39.852	67.326	0.926	830.333
8	335	9	89.891	43.821	69.644	0.923	819.153
9	298	7	88.115	43.838	69.329	0.923	832.182
10	268	6	85.109	43.622	69.867	0.923	878.649
11	244	7	82.719	42.309	66.571	0.925	875.988
12	224	8	78.552	43.597	68.532	0.924	774.784
13	207	11	76.708	43.639	66.801	0.924	797.845
14	192	8	74.112	44.385	67.146	0.923	784.872

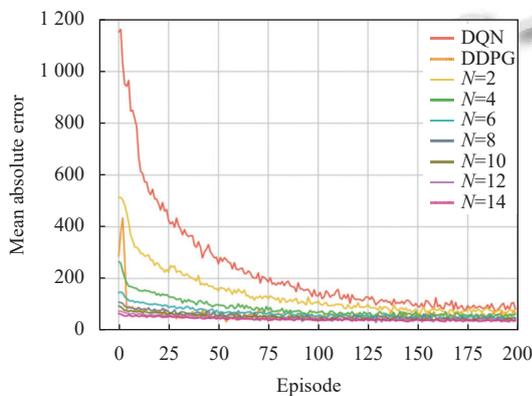


图5 3种算法的MAE变化趋势

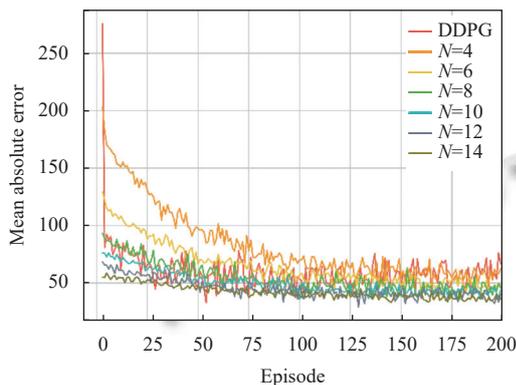


图6 3种算法的MAE变化趋势(从第3回合开始)

参考文献

1 Cao XD, Dai XL, Liu JJ. Building energy-consumption status worldwide and the state-of-the-art technologies for zero-energy buildings during the past decade. *Energy and Buildings*, 2016, 128: 198–213. [doi: 10.1016/j.enbuild.2016.06.089]

2 Massana J, Pous C, Burgas L, et al. Short-term load forecasting for non-residential buildings contrasting artificial occupancy attributes. *Energy and Buildings*, 2016, 130: 519–531. [doi: 10.1016/j.enbuild.2016.08.081]

3 Pedersen TH, Hedegaard RE, Petersen S. Space heating demand response potential of retrofitted residential apartment blocks. *Energy and Buildings*, 2017, 141: 158–166. [doi: 10.1016/j.enbuild.2017.02.035]

4 Amasyali K, El-Gohary NM. A review of data-driven building energy consumption prediction studies. *Renewable and Sustainable Energy Reviews*, 2018, 81: 1192–1205. [doi: 10.1016/j.rser.2017.04.095]

5 Wang ZY, Srinivasan R, Wang YR. Homogeneous ensemble model for building energy prediction: A case study using ensemble regression tree. *Proceedings of 2016 ACEEE Summer Study on Energy Efficiency in Buildings*. Pacific Grove, 2016. 21–26.

6 Cuerda E, Guerra-Santin O, Sendra JJ, et al. Understanding the performance gap in energy retrofitting: Measured input data for adjusting building simulation models. *Energy and Buildings*, 2020, 209: 109688. [doi: 10.1016/j.enbuild.2019.109688]

7 Ekici BB, Aksoy UT. Prediction of building energy consumption by using artificial neural networks. *Advances in Engineering Software*, 2009, 40(5): 356–362. [doi: 10.1016/j.advengsoft.2008.05.003]

8 Li Q, Meng QL, Cai JJ, et al. Applying support vector machine to predict hourly cooling load in the building. *Applied Energy*, 2009, 86(10): 2249–2256. [doi: 10.1016/j.apenergy.2008.11.035]

9 Wang ZY, Wang YR, Zeng RC, et al. Random forest based hourly building energy prediction. *Energy and Buildings*, 2018, 171: 11–25. [doi: 10.1016/j.enbuild.2018.04.008]

10 Yu Z, Haghghat F, Fung BCM, et al. A decision tree method for building energy demand modeling. *Energy and Buildings*, 2010, 42(10): 1637–1646. [doi: 10.1016/j.enbuild.2010.04.006]

11 LeCun Y, Bengio Y, Hinton G. Deep learning. *Nature*, 2015, 521(7553): 436–444. [doi: 10.1038/nature14539]

12 Fan C, Xiao F, Zhao Y. A short-term building cooling load prediction method using deep learning algorithms. *Applied Energy*, 2017, 195: 222–233. [doi: 10.1016/j.apenergy.2017.03.064]

13 Rahman A, Srikumar V, Smith AD. Predicting electricity consumption for commercial and residential buildings using deep recurrent neural networks. *Applied Energy*, 2018, 212:

- 372–385. [doi: [10.1016/j.apenergy.2017.12.051](https://doi.org/10.1016/j.apenergy.2017.12.051)]
- 14 Levine S, Finn C, Darrell T, *et al.* End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 2016, 17(1): 1334–1373. [doi: [10.5555/2946645.2946684](https://doi.org/10.5555/2946645.2946684)]
- 15 El Sallab A, Abdou M, Perot E, *et al.* Deep reinforcement learning framework for autonomous driving. *Electronic Imaging*, 2017, 29: art00012. [doi: [10.2352/ISSN.2470-1173.2017.19.AVM-023](https://doi.org/10.2352/ISSN.2470-1173.2017.19.AVM-023).]
- 16 Zhang ZA, Chong A, Pan YQ, *et al.* A deep reinforcement learning approach to using whole building energy model for HVAC optimal control. *Proceedings of 2018 Building Performance Analysis Conference and SimBuild*. Chicago: ASHRAE, 2018. 22–23.
- 17 Nagy A, Kazmi HS, Cheaib F, *et al.* Deep reinforcement learning for optimal control of space heating. arXiv: 1805.03777, 2018.
- 18 Liu T, Tan ZH, Xu CL, *et al.* Study on deep reinforcement learning techniques for building energy consumption forecasting. *Energy and Buildings*, 2020, 208: 109675. [doi: [10.1016/j.enbuild.2019.109675](https://doi.org/10.1016/j.enbuild.2019.109675)]
- 19 Zhang WY, Chen Q, Yan JY, *et al.* A novel asynchronous deep reinforcement learning model with adaptive early forecasting method and reward incentive mechanism for short-term load forecasting. *Energy*, 2021, 236: 121492. [doi: [10.1016/j.energy.2021.121492](https://doi.org/10.1016/j.energy.2021.121492)]
- 20 Sutton RS, Barto AG. *Reinforcement Learning: An Introduction*. Cambridge: MIT Press, 2018. 75–76.
- 21 Watkins CJCH, Dayan P. Q-learning. *Machine Learning*, 1992, 8(3–4): 279–292. [doi: [10.1007/BF00992698](https://doi.org/10.1007/BF00992698)]
- 22 Rummery GA, Niranjan M. *On-line Q-learning Using Connectionist Systems*. Cambridge: University of Cambridge, 1994. 5–6
- 23 Tsitsiklis JN, Van Roy B. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 1997, 42(5): 674–690. [doi: [10.1109/9.580874](https://doi.org/10.1109/9.580874)]
- 24 Hastie T, Tibshirani R, Friedman J. *The elements of statistical learning: Data mining, inference, and prediction*. New York: Springer, 2017. 493–497.
- 25 Breunig MM, Kriegel HP, Ng RT, *et al.* LOF: Identifying density-based local outliers. *ACM SIGMOD Record*. 2000, 29(2): 93–104.

(校对责编: 牛欣悦)