Chemometrics and Intelligent Laboratory Systems 179 (2018) 64–72

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

journal homepage: www.elsevier.com/locate/chemometrics

A novel intelligent modeling framework integrating convolutional neural network with an adaptive time-series window and its application to industrial process operational optimization

Yongjian Wang, Hongguang Li *

College of Information Science and Technology, Beijing University of Chemical Technology, China

ARTICLE INFO

Keywords:
Convolutional neural networks
Adaptive time-series window
Operational optimization
Industrial process
Furnace

ABSTRACT

With the increasing complexity of industrial processes, it becomes more and more difficult to set up process operational optimization models. Recently, the convolutional neural network (CNN) has been widely and successfully applied to extracting useful information due to its deep learning capability. Aiming at extracting useful information concerning operational optimization from complex industrial process data, this paper proposes a novel framework integrating CNN with an adaptive time-series window (ATSW-CNN). The proposed ATSW-CNN method is composed of four kinds of network layers, i.e. the proposed adaptive layer, convolutional layers, pooling layers and fully connected layers. The proposed adaptive layer provides CNN paradigms with a capability of adaptively selecting appropriate time-series windows for different steady-state operational optimization data. As a result, the proposed ATSW-CNN method can effectively extract steady-state optimal operating conditions from process time-series data. In order to validate the performance of the proposed ATSW-CNN, simulations on an industrial furnace are carried out. Simulation results verify the effectiveness of the proposed method, which demonstrates the proposed ATSW-CNN method is applicable for searching steady state operating strategies.

1. Introduction

With the development of industrial processes, efficient production is playing an increasingly important role; however, it is still very hard to realize efficient production due to complex operations and interactions in industrial processes [1–3]. In order to keep efficient production, it is necessary to carry out operational optimization. Operational optimization plays an important role in industrial processes because of its contribution to reducing unnecessary losses and improving production efficiency [4,5].

Optimal operation depends on the establishment of a good model. There are many kinds of modeling methods. Generally speaking, modeling methods can be divided into two categories: mechanism modeling methods and data-driven modeling methods. The mechanism models are established by analyzing the detailed industrial process knowledge; however, as industrial processes become more and more complex, it becomes more and more difficult to obtain accurate process knowledge and establish mechanism models. Compared with mechanism modeling methods, data-driven modeling methods require little process knowledge and mainly rely on the collected process data. Human operators' empirical decision-making knowledge is hidden in process data. Optimal operations can result from data-driven models rather than mechanism models. Many researchers have put a lot of attention on the data-driven modeling methods [6–8]. Data-driven methods have two salient features. On one hand, there is no need to understand the knowledge and information in the detailed complex processes, only the data related to the data-driven modeling method is needed. On the other hand, with the development of distributed control systems in process industries, historical operating data with rich operational experiences can be easily collected. However, these historical operating data are rarely utilized. Hence, it is necessary to extract operational optimization feature vectors from original process data. Numerous data-driven operational optimization techniques and methods for industrial processes have been developed in the literature [9–13]. Well-developed and diverse as the traditional operational optimization methods are, they still suffer from several drawbacks in real industrial applications.

Data-driven methods can avoid setting up mechanism models for the complex industries and directly train the operation variables and their associated variables. Therefore, more and more researchers replace the traditional mechanism model modeling method with data-driven

* Corresponding author.
E-mail address: lihg@mail.buct.edu.cn (H. Li).

https://doi.org/10.1016/j.chemolab.2018.06.008
Received 8 April 2018; Received in revised form 17 June 2018; Accepted 20 June 2018
Available online 21 June 2018

modeling methods. Neural networks (NNs) are one of the most widely used data-driven methods. NNs are mathematical models of distributed parallel information processing. NNs have good performance in dealing with high nonlinear data. NNs adjust the interconnected relationships among a large number of internal nodes and have the ability of self-learning and self-adaptation. Therefore, NNs have been successfully applied to the modeling of complex systems and nonlinear systems [14–17]. As operational optimization involves complex industrial processes with a high degree of non-linearity. NNs based methods are used to solve operational optimization problems [18]. Souza et al. [19] proposed a neural network based modeling and operational optimization method to deal with the biomass gasification process. Because the process operation data has time-series characteristics, and there is a relationship between the data before and after each variable. While taking account of the time-series characteristics of process data, the processing of optimal operational optimization data should concern both correlated variables and adjacent data points. Convolutional neural network (CNN) is an alternative. CNN is a feedforward neural network. The artificial neurons in CNN can respond to a part of the surrounding cells within the coverage area and have excellent performance for processing large-scale image and time-series data analysis. CNN can be adopted to achieve operational optimization of time-series industrial process data.

CNN is one of the emerging deep learning methods in recent years. Deep learning has attracted much attention in both academia researches and industry applications [20,21]. Scholars from different fields try to use deep learning methods to solve practical problems [22,23] such as a variety of modeling issues [24–27]. Deep learning can be understood as using deep neural networks to extract features from complex high-dimensional data. It is found that four basic deep neural networks are available currently: deep belief networks (DBNs) [28,29], stacked auto encoders (SAE) [30,31], recurrent neural networks (RNNs) [32,33] and convolutional neural networks (CNNs) [34–37]. CNN is good at processing data with grid topology features. Time-series data can be viewed as one-dimensional grid data sampled at fixed time intervals. So, CNN can also be used to deal with time-series data. Taking account of the advantages mentioned above, CNN is selected to deal with industrial process time-series data in this paper. In practical applications, CNN has been successful used in many fields. Chen et al. [38] proposed an algorithmic trading framework for CNN. Liu et al. [39] proposed a dislocated time-series CNN structure to improve the accuracy of fault diagnosis; in this method a fixed formula was used to extract continuous data in segments, and then the extracted data were used as new time-series data, achieving a good classification performance. Zhao et al. [40] used a novel CNN structure for time-series classifications, where convolution and pooling operations are alternatively used to generate deep features of raw data. Zheng et al. [41] proposed an effective multi-channels deep CNN model to effectively deal with weakly labeled data. Most industrial process operating data are sequential. The operating data can be regarded as one-dimensional grid data sampled at fixed time intervals.

However, all the above methods cannot well deal with the uncertain dynamic process data. In order to solve this problem, a novel intelligent modeling framework integrating convolutional neural network with an adaptive time-series window (ATSW-CNN) is proposed in this paper. The proposed ATSW-CNN model consists of the proposed adaptive layer, convolutional layers, pooling layers and fully connected layers. In the proposed adaptive layer, an adaptive time-series window selection algorithm is designed. Different from the method proposed by Liu et al. [39], the core idea of the proposed ATSW-CNN method lies in an additive adaptive layer is adopted. With the aid of the proposed adaptive layer, different time-series windows can be adaptively selected according to the different steady-state time lengths, the optimal operation features of the steady state data can be easily obtained. Then, the adaptive time-series window selection algorithm is used to pre-process the process time-series data. After several convolution and pooling processes, fully connected layers are used to obtain the final output. Optimal operational strategies for different steady states can be effectively learned and predicted based on industrial process data using the proposed ATSW-CNN model. In order to validate the performance of the proposed ATSW-CNN, a case study on an industrial furnace is carried out. Simulation results verify the effectiveness of the proposed method.

The remaining parts of this paper are organized as follows: Section 2 describes the CNN architecture; Section 3 provides the detail procedures of the proposed ATSW-CNN model and explicitly illustrates the intelligent operation framework; a case study of operational optimization modeling on an industrial furnace using the proposed ATSW-CNN model is carried out in Section 4, demonstrating the effectiveness of proposed method; Section 5 gives conclusions.

2. CNN metrics

CNN is a kind of neural networks specially used in deeply processing data. LeCun [42] is the earliest CNN structure for handwritten number recognitions. Three important ideas help CNN to improve the learning performance of neural networks: sparse interactions, parameter sharing and pooling.

2.1. Sparse interactions

Traditional neural networks use matrix multiplication to establish the connection between inputs and outputs. Therein, each individual parameter in the parameter matrix describes the interaction between an input unit and an output unit, which means that each output unit interacts with each input unit. However, the CNN with sparse interaction features can make the kernel size much smaller than the input size. Fig. 1(a) describes a normal full interaction structure between two layers. In this structure, any node is connected with all of the nodes in different layers. Fig. 1(b) describes a sparse interaction structure between two layers. In this structure, any node is not connected with all of the nodes in different layers. In Fig. 2, although the direct connection is sparse, units in deeper layers still can be indirectly connected to all or most of inputs. By sparse interactions, the network can efficiently describe the complex interactions of multiple variables by describing only sparse interactions.

2.2. Parameter sharing

Parameter sharing means that a common parameter is used in multiple functions. In traditional neural networks, as shown in Fig. 3(a), each parameter of the weight matrix represented by the yellow arrow is used when calculating the output. This parameter will be updated when an entered element is multiplied. This parameter is not shared. In Fig. 3(b), a special parameter connection represented by the red arrow is used by one of the output units. This special parameter is regarded as the sharing parameter and is simultaneously used the other output units. Parameter sharing reduces the network complexity and the number of parameters that the model needs to store when calculating.

2.3. Pooling

A typical layer in a convolutional neural network contains three levels: convolution level, detection level and pool level. The overall statistical characteristics of the adjacent position's output instead of the network's output at the current point are utilized as the outputs of pooled functions. The pooling layer has two forms: mean-pooling and max-pooling. The two kinds of sub-sampling are shown in Fig. 4. In Fig. 4(a)–(b), a, b, c, d represent the items of the input matrix $\boldsymbol{X}$; d is the maximum value among a,b,c,d. $\boldsymbol{W}$ represent the transformation matrix of