Front. Mech. Eng. 2022, 17(2): 17
https://doi.org/10.1007/s11465-022-0673-7

RESEARCH ARTICLE

A new automatic convolutional neural network based on deep reinforcement learning for fault diagnosis

Long WEN[a], **You WANG**[a], **Xinyu LI** (✉)[b]

[a] *School of Mechanical Engineering and Electronic Information, China University of Geosciences, Wuhan 430074, China*
[b] *State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan 430074, China*

✉ *Corresponding author. E-mail: lixinyu@mail.hust.edu.cn (Xinyu LI)*

ABSTRACT Convolutional neural network (CNN) has achieved remarkable applications in fault diagnosis. However, the tuning aiming at obtaining the well-trained CNN model is mainly manual search. Tuning requires considerable experiences on the knowledge on CNN training and fault diagnosis, and is always time consuming and labor intensive, making the automatic hyper parameter optimization (HPO) of CNN models essential. To solve this problem, this paper proposes a novel automatic CNN (ACNN) for fault diagnosis, which can automatically tune its three key hyper parameters, namely, learning rate, batch size, and L2-regulation. First, a new deep reinforcement learning (DRL) is developed, and it constructs an agent aiming at controlling these three hyper parameters along with the training of CNN models online. Second, a new structure of DRL is designed by combining deep deterministic policy gradient and long short-term memory, which takes the training loss of CNN models as its input and can output the adjustment on these three hyper parameters. Third, a new training method for ACNN is designed to enhance its stability. Two famous bearing datasets are selected to evaluate the performance of ACNN. It is compared with four commonly used HPO methods, namely, random search, Bayesian optimization, tree Parzen estimator, and sequential model-based algorithm configuration. ACNN is also compared with other published machine learning (ML) and deep learning (DL) methods. The results show that ACNN outperforms these HPO and ML/DL methods, validating its potential in fault diagnosis.

KEYWORDS deep reinforcement learning, hyper parameter optimization, convolutional neural network, fault diagnosis

1 Introduction

Fault diagnosis has attracted increasing attention in the academia and industrial fields [1]. With the rapid development of smart manufacturing, data-driven fault diagnosis (also referenced as knowledge-based fault diagnosis), which always establishes the machine learning (ML) or deep learning (DL) model to map the nonlinear relationship among the collected signals to the health condition of the machine, has been widely investigated and obtained remarkable achievement results. It has attracted more attention in the industrial and academic fields [2,3].

DL is one of the most widely used methods in data-driven fault diagnosis [4,5]. DL-based fault diagnosis mainly contains feature extraction realized by signal processing techniques and fault recognition implemented by DL classification algorithms [6]. DL can automatically handle signals after processing to predict the health condition of machines. However, obtaining excellent performance on all situations is impossible for every supervised algorithm due to "no free lunch" theorems [7,8]. Therefore, finding its optimal configurations to ensure its performance is essential for the supervised algorithm. DL is a kind of supervised algorithm, so it suffers from this phenomenon as well. In various applications, the tuning to search the optimal configurations for DL models is necessary and has been investigated widely in the field [9].

Tuning aims to find an appropriate set of hyper parameters (e.g., batch size, regularization coefficient, and learning rate of neural networks) for DL models. Although several default hyper parameters have been recommended, researchers show that they cannot

Received September 3, 2021; accepted January 10, 2022

guarantee the performances of DL methods on real-world applications [10]. As a result, tuning would be conducted on every dataset, making it very time consuming and labor intensive. Typically, the most usual tuning method is manual search, but it often requires expert experience to guide the search, and this is a considerable barrier for new users, especially on the new dataset [9,10].

Hyper parameter optimization (HPO), regarding tuning as the optimization problem, has attracted the attention of many researchers [9]. It can automatically search the optimal hyper parameters for DL models. The most used HPO methods include grid search (GS), random search (RS), Bayesian optimization (BO), sequential model-based algorithm configuration (SMAC), and tree Parzen estimator (TPE) [11]. GS and RS are simple HPO methods, whereas BO, SMAC, and TPE establish a surrogate model that maps the hyper parameter space to the final generation error space for the DL modes and then optimizes the hyper parameters online. As HPO can automatically tune the DL models, it is easier to use and obtain the state-of-the-art results in various applications [12]. However, most DL models developed for fault diagnosis are tuned manually, and the applications of HPO to the DL model for fault diagnosis are few. Thus, improving the search efficiency for the DL models for fault diagnosis and developing a systemic search strategy for its tuning are vital to the process.

In this paper, a new automatic convolutional neural network (ACNN) is studied for fault diagnosis. Convolutional neural network (CNN) is applied for fault diagnosis. As some theoretical and empirical evidence show that three hyper parameters, namely, batch size, learning rate, and L2-regulation, have great effect on the performance of CNN models [13,14], they are tuned simultaneously online by a deep reinforcement learning (DRL) agent. First, a new DRL paradigm is developed for controlling the hyper parameters for the CNN model. Second, a new DRL structure is developed, which combines long short-term memory (LSTM) and deep deterministic policy gradient (DDPG), named LSTM–DDPG. LSTM–DDPG can take the historical training loss of CNN models as input and control the batch size, learning rate, and L2-regulation during the training of the CNN model. DDPG is a powerful kind of DRL, and it can release the continuous control on these three hyper parameters. Third, a new training method for ACNN with random walk is designed to avoid falling into the local minimum. Finally, the proposed ACNN is conducted on two famous bearing datasets, and the results show that it achieves state-of-the-art performance and is easy to use.

The remainder of this paper is organized as follows. The related work is presented in Section 2. The structure of the proposed ACNN is shown in Section 3. The ACNN method for fault diagnosis is illustrated in Section 4. The experiment study is carried out to validate ACNN in Section 5. Finally, the conclusions and future research are introduced in Section 6.

2 Related work

2.1 CNN-based fault diagnosis

As one of the powerful kinds of DL, CNN has been studied by many researchers in fault diagnosis [15,16]. Xu et al. [17] studied online fault diagnosis by using deep transfer CNN, and it achieved remarkable results on the bearing and pump dataset. Li et al. [18] proposed a Wasserstein generative adversarial network for imbalance fault diagnosis. It generated several high-quality samples of the minority class, and the results showed that it has better performance. Chen et al. [19] investigated multiscale CNN with feature alignment for rolling bearing fault diagnosis, which outperformed other CNN-based methods in terms of accuracy and feature robustness. Jiao et al. [20] provided an extensive review of the CNN models. Yao et al. [21] studied an attention assist representation method with multiscale CNN for fault diagnosis of gear under non-linear and non-stationary working conditions. Li et al. [22] investigated an adaptive fusion CNN for multisensor fault diagnosis with adaptive sized convolution kernels and achieved good performance. Kolar et al. [23] developed a new deep CNN model for fault diagnosis of rotary machinery using the raw three axes' accelerometer signal as input.

During the applications of CNN in fault diagnosis, several decisions should be pre-defined, including determining the structure of the CNN as well as the hyper parameters for training the CNN models [9]. Searching for the optimal structure of CNN is the field of neural architecture search (NAS). Wang et al. [24] studied reinforcement NAS for rolling bearing fault diagnosis, and the results confirmed that the proposed method can realize the automatic design of the CNN model. Zhang et al. [25] investigated differentiable NAS for fault diagnosis of bearing dataset and escalator dataset. The experimental results showed the effectiveness of NAS in achieving competitive performance.

In this research, famous CNN structures, such as ResNet and DenseNet, are adopted for fault diagnosis, and the hyper parameters for training the CNN models are mainly focused on. As some theoretical and empirical evidence show that batch size, learning rate, and L2-regulation have great effect on the performance of the CNN models [13,14], a new DRL-based automatic CNN that can control the three key hyper parameters of CNN-based fault diagnosis is developed.

2.2 Hyper parameter optimization

The typical HPO methods include GS, RS, BO, SMAC, and TPE. Among these methods, BO, SMAC, and TPE use surrogate models to map the hyper parameter space to the performance space and can often find good hyper

parameter configuration for the ML/DL models.

(1) Grid search and random search

GS is a famous hyper parameter tuning method, and it tries every combination of the hyper parameter candidate to find their optimal values. GS is reliable and widely applied in low-dimensional space. It suffers from "the curse of dimension," so computational time would increase heavily in high-dimensional hyper parameter space.

The procedure of RS is similar to that of GS, except that it randomly selects the trial hyper parameter candidate. The efficiency of RS is better than that of GS in high-dimensional hyper parameter space. However, RS is prone to fall into the local minimum because it lacks guidance for the search on hyper parameter configuration.

(2) Bayesian optimization

BO is a sequencing optimization method, which establishes a Gaussian model as the surrogate model and selects the next potential candidate hyper parameter to be tested. With the certain steps, the hyper parameter can be optimized. BO has been useful in many applications [26].

(3) Sequential model-based algorithm configuration

SMAC applies random forest as a surrogate model to estimate the mean and variance for each hyper parameter. It is suitable for all types of hyper parameters [12], including continuous, discrete, categorical, and conditional.

(4) Tree Parzen estimators

TPE applies a graphic structure to deal with conditional search space for the surrogate model [27]. It creates two density functions for each hyper parameter, good ones and bad ones [11].

The applications of HPO methods for CNN-based fault diagnosis are few. Li et al. [28] combined CNN with automatic HPO for fault diagnosis of roller bearing. Cabrera et al. [26] studied the hyper parameter search guided by the Bayesian approach for LSTM-based fault diagnosis. Long et al. [29] investigated a competitive swarm optimizer combined with a local search for the optimization of deep echo state network architecture for intelligent fault diagnosis. Han et al. [30] proposed a genetic algorithm method for the HPO of a simple CNN model on the MNIST dataset and the motor fault diagnosis dataset. Wei et al. [31] presented a new imbalance fault diagnosis method using a cluster–majority weighted minority oversampling technique and support vector machine (SVM). In this method, genetic algorithm and particle swarm optimization are applied to optimize the hyper parameters of the SVM.

The hyper parameters in these HPO methods are assumed constant, but several hyper parameters in DL models, such as batch size, learning rate, and L2-regulation, can be time varying. In this paper, the proposed ACNN is developed to control these hyper parameters online for CNN-based fault diagnosis.

3 Proposed ACNN using LSTM–DDPG

The proposed ACNN is defined in this section. First, the DRL preliminaries are presented. Second, the ACNN based on DRL is defined. Third, the LSTM–DDPG structure for the ACNN is given.

3.1 DRL preliminaries for ACNN

Reinforcement learning (RL) has been widely used for online control. The procedure of RL can be modelled as a Markov decision process. It trains an RL agent to make the continuous decisions, which decides to take action a according to its state s while receiving reward r. Thus, RL can be defined as the tuple $\langle s,a,p,r,\gamma\rangle$. p denotes the transition probability function and is $p: s\times a = [0, 1]$. γ is discount factor. In most cases, policy $\Pi: s \to a$ is used for the agent to choose the action according to its current state. The ultimate goal of the RL agent is to find the proper policy that can maximize the cumulative reward via the Q-value function, as shown in Eq. (1):

$$Q^{\Pi}(s,a) = E_{\Pi}\left\{\sum_{k=t}^{T}\gamma^{k} r_k | s,a\right\}, \tag{1}$$

where E_{Π} denotes the expected value under policy Π, r_k is the reward at the kth step, and $Q^{\Pi}(s,a)$ is the Q-value function under policy Π.

Equation (1) shows that when the states and actions are finite, the Q-value function can be viewed as a look-up Q table; otherwise, it is modelled using a surrogate model. When the artificial neural network (ANN) is used as the surrogate model, this type of RL is denoted as DRL. However, the hyper parameters of CNN are the continuous controlling type, so the action should be continuous. DDPG, which is an improved version of DRL, is familiar with continuous control and is investigated in this research.

3.2 RL definition of ACNN

The definitions of RL for the concepts of ACNN-based fault diagnosis include state space, action space, and reward function.

3.2.1 State definition

State definition should reflect the current state of the ACNN training state to guide the further trend of the training. Some previous works have been done to design the state for ANN, but most of them are carefully hand constructed, making them task specific, such as the state by Hansen [32] that has been used for tuning the learning rate only.

In this research, the state for ACNN is defined as the sequencing training loss of ACNN on mini-batch data,

and the number of the sequencing training loss is defined as M. Training loss is meaningful because it can reflect the convergence and performance of the ACNN model. Hence, the state of ACNN is defined as Eq. (2):

$$s_t = (loss_{t-M}, loss_{t-M+1}, ..., loss_{t-1}), \quad (2)$$

where s_t is the state at time step t, and $loss_t$ denotes the loss value of CNN model at time step t. The current state is the past sequencing M loss of ACNN, and it is time series data. Thus, LSTM is applied with DDPG to handle it, as shown in Section 3.3.

3.2.2 Action space for controlling the hyper parameter of ACNN

In this research, the controlled hyper parameters include batch size, learning rate, and L2-regulation value, so the action space is 3D. To simplify the action space, the spaces of the three elements in action space are all pre-defined as (0, 1), and it can be easily achieved by the "sigmoid" activation function. Suppose that $a_t = (a_{t,1}, a_{t,2}, a_{t,3})$ defines the current action, then the current batch size b_t, current learning rate lr_t, and current L2-regulation value l_t can be updated using Eqs. (3)–(5), respectively. The L2-regulation value is updated in the exponential scale because this value is also commonly tuned in the exponential term in manual search. For example, the change from 0.001 to 0.01 is expected to have the same effect as that from 0.01 to 0.1.

$$b_t = \text{int}((b_{\max} - b_{\min}) a_{t,2} + b_{\min}), \quad (3)$$

$$lr_t = (lr_{\max} - lr_{\min}) a_{t,1} + lr_{\min}, \quad (4)$$

$$l_t = pow(10, (\lg l_{\max} - \lg l_{\min}) a_{t,3} + \lg l_{\min}), \quad (5)$$

where $lr_{\max}$ and $lr_{\min}$ are the upper and lower boundaries for the learning rate, respectively, $b_{\max}$ and $b_{\min}$ are the upper and lower boundaries for the batch size, respectively, and $l_{\max}$ and $l_{\min}$ are the upper and lower boundaries for the regulation, respectively. To make the controlling smooth, the final batch size, learning rate, and L2-regulation are soft updated, as shown in Eqs. (6)–(8), respectively.

$$b_t = \text{int}(b_{t-1}(1-\alpha) + b_t\alpha), \quad (6)$$

$$lr_t = lr_{t-1}(1-\alpha) + lr_t\alpha, \quad (7)$$

$$l_t = pow(10, ((1-\alpha)\lg l_{t-1} + \alpha \lg l_t)), \quad (8)$$

where α is the factor to control the degree of soft updating. Eq. (8) is smoothed in the log scale as Eq. (5).

3.2.3 Reward function

Reward function is also a vital component in the RL paradigm for ACNN, as the RL agent tries to maximize the cumulative reward during training. The reward function is determined as the improvement on the training loss between two time steps on the ACNN model. This definition is to promote the convergence speed of the ACNN and ensure the final training loss. The final reward function can be expressed in Eq. (9):

$$r_{t+1} = loss_{t-1} - loss_t, \quad (9)$$

where r_{t+1} is the reward at time step t+1.

3.3 LSTM–DDPG for ACNN

DDPG is an improved version in RL paradigm, which is familiar with the continuous state space, and it has achieved many successful applications on the task in Gym and Atari environment [33]. The structure of DDPG contains two main networks, namely, actor network and critic network. The function of the actor network is to choose the action according to the state. Suppose θ^μ denotes the actor network, and the process of actor network can be presented as $\theta^\mu(s_t) = a_t$. At the same time, the critic network is used as the surrogate model for Q-value function. ω^Q denotes the critic network, and the process of critic network is $Q(s_t, \theta^\mu(s_t)|\omega^Q)$.

In DDPG, the actor and critic networks have two subnets. These two subnets share the same structure, denoted as online subnet and target subnet. The online subnets operate the normal workflow of DDPG, whereas the target subnets are used for the experience replay, that is, the training of the subnet. The actor and critic target networks are denoted as $\theta^{\mu'}$ and $\omega^{Q'}$, respectively. The LSTM network is used to handle with the sequence loss information in DDPG because the state is the defined as the time sequence of loss. This structure is denoted as LSTM–DDPG, and its details are shown in Fig. 1. The actor network applies an LSTM with two fully connected (FC) layers. The activation function of the actor network is "sigmoid." The input of the actor network is the state s_t. The output is the action with the range of (0, 1), which would be used to adjust the batch size, learning rate, and L2-regulation using Eqs. (6)–(8), respectively. The critic network has two inputs, namely, state (Input 1) and action (Input 2). LSTM is used to copy the state, and then the result of the LSTM is concatenated with action a_t to estimate the Q-value function. No activate function exists in the end of the FC layers in the critic network, and it can predict any value of Q-value function to fit the range of reward.

The training of DDPG needs to train the actor and critic networks. The training loss L of critic network is defined as the squared error of the prediction on the Q-value function $Q(s_t, a_t|\omega^Q)$ and the actual Q-value y_t, as shown in Eq. (10), where n is the number of samples in the experience storage D. The actual Q-value y_t is calculated using the reward r_{t+1} and the prediction reward $Q(s_{t+1}, \mu(s_{t+1})|\omega^{Q'}|\theta^{\mu'})$ on the state s_{t+1}, as shown in Eq. (11). The training of the actor network is updated using

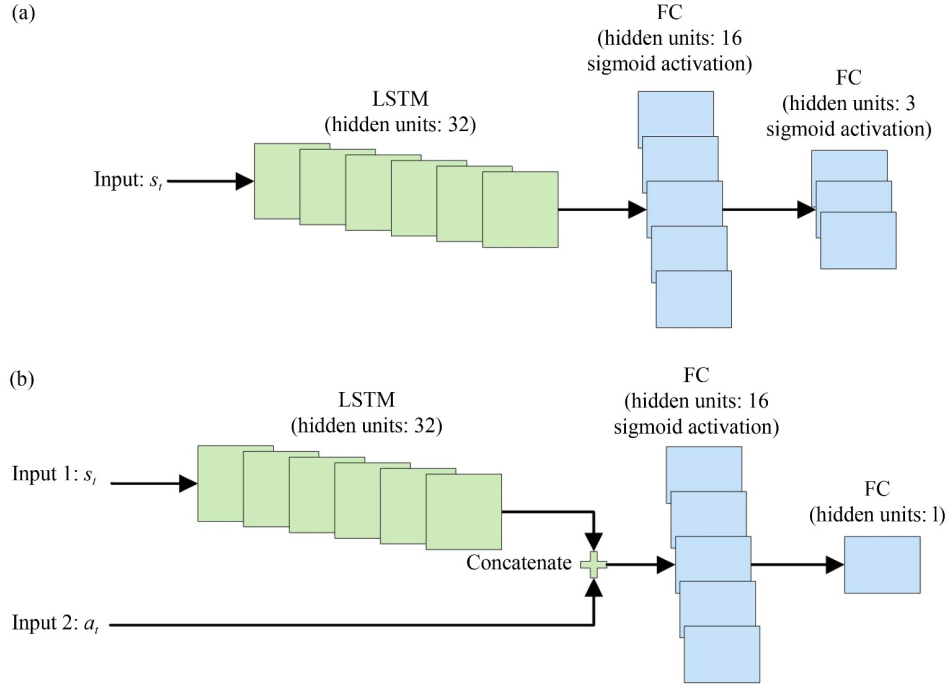

Fig. 1 Detail structure of LSTM–DDPG: (a) actor network, (b) critic network.

the sample policy gradient $\nabla_{\theta^\mu} J(\theta^\mu)$ shown in Eq. (12).

$$L = \frac{1}{n}\sum_t \left(y_t - Q(s_t, a_t|\omega^Q)\right)^2, \tag{10}$$

$$y_t = r_{t+1} + \gamma Q\left(s_{t+1}, \mu(s_{t+1})|\omega^{Q'}|\theta^{\mu'}\right), \tag{11}$$

$$\nabla_{\theta^\mu} J(\theta) = \frac{1}{n}\sum_{j=1}^{n} \nabla_a Q(s_i, a_i|\omega^Q)\nabla_{\theta^\mu}\mu(s|\theta^\mu). \tag{12}$$

4 Proposed ACNN-based fault diagnosis

This section presents ACNN-based fault diagnosis, which includes data preprocessing, ACNN structure, and ACNN training.

4.1 Data preprocessing based on S-transform

S-transform is a kind of time–frequency technique, and it has a good ability to handle non-stationary signals. The application of S-transform on fault diagnosis has been studied and shown a good performance. S-transform is based on the famous short-time Fourier transform. Suppose that signal $x(t)$ is defined, its short-time Fourier transform formulation result denoted by *STFT* is as Eq. (13):

$$STFT(\tau, f) = \int_{-\infty}^{+\infty} x(t)w(t-\tau)e^{-\mathrm{j}2\pi ft}\mathrm{d}t, \tag{13}$$

where τ is the time of spectral localization, f is the Fourier frequency, and $w(t)$ is the window function. S-Transform defines $w(t)$ as Gaussian function, as shown in Eq. (14). π is the circumference ratio. Then, S-transform can be calculated as Eq. (15):

$$w(t) = \frac{|f|}{\sqrt{2\pi}}\exp(-t^2f^2/2), \tag{14}$$

$$ST(\tau, f) = \int_{-\infty}^{+\infty} x(t)\frac{|f|}{\sqrt{2\pi}}\exp\left(-(t-\tau)^2f^2/2\right) \cdot \exp(-\mathrm{j}2\pi ft)\mathrm{d}t. \tag{15}$$

Equation (15) shows that the results of S-transform contain time and frequency elements, and then these data are fed to the ACNN-based fault diagnosis method.

4.2 ACNN structure

The structure of ACNN applies the famous off-the-shelf CNN models as the backbones. These backbones have been validated to be powerful. At the end of these backbones, the new FC layer and a SoftMax classifier are added for the fault classification. ResNet and DenseNet are used as the backbones in this research.

1) ResNet: ResNet uses shortcut connections to skipping blocks of convolutional layers, which can ease the training, because deeper neural networks are more difficult to train. With the deeper network structure, ResNet can substantially improve the final performance.

2) DenseNet: DenseNet is an improved version of ResNet, and it is based on the empirical finding that the CNN network can be more efficient to train when the shorter connections are used between the layers close to the input and those close to the output. Thus, DenseNet

connects each layer to every other layer, and it can alleviate the vanishing-gradient problem, encourage feature reuse, and reduce the number of parameters.

In this research, three ResNet variants and three DenseNet variants are used as the structure of ACNN for fault diagnosis. They are ResNet-50V2, ResNet-101V2, ResNet-152V2, DenseNet-121, DenseNet-169, and DenseNet-201. Their well-trained models are on the Keras application. The hidden neuron of the followed FC layer is 512, and that in the SoftMax classifier depends on the fault cases.

4.3 ACNN training

The training of the proposed ACNN consists of the alternative training of CNN models and DDPG in turn. The training of ACNN is shown in Algorithm 1. During the training, a random walk procedure is designed to improve the stability of ACNN, as shown in Algorithm 2.

totalstep is the total training step, and *step4game* is used to control the frequency of random walks. Algorithm 1 shows that ACNN first generates its state s_t, and the DDPG predicts the learning rate, batch size, and regulation value of ACNN for this training step. Then, this setting is conducted by ACNN and receives its reward r_{t+1}, which is used to indicate the goodness of this setting.

Algorithm 1: ACNN Training

Data preprocessing the fault samples using Eq. (15)
Initialize ACNN model and DDPG model
Initialize the samples using mini-batch method
for t=1, *totalstep* **do**
  if t+1% *step4game* == 0
    Implement the random walks as Algorithm 2
    Train critic online subnets using Eq. (10) and D'
    Train actor online subnets using Eq. (12) and D'
    Update target actor and critic subnets
  end if
  Generate the CNN state as s_t using Eq. (2)
  Predict the action $a_t = \mu_{\theta^\mu}(s_t)$
  Obtain lr_t, b_t and l_t using Eqs. (6)–(8)
  Obtain the mini-batch fault samples using b_t
  Train one step of CNN model using lr_t and l_t
  Obtain the $loss_t$ and get the reward r_{t+1} using Eq. (9)
  Store $(s_t, a_t, r_{t+1}, s_{t+1})$ in experience storage D
  t=t+1
end for

Algorithm 2: Random walks for ACNN

Clone the ACNN model to a temporary CNN model
Clone the mini-batch indicator
Clone the experience storage D to D'
for k = 1, *step4game* **do**
  randomly generate the action a_k
  Obtain b_k, lr_k and l_k using Eqs. (6)–(8)
  Generate the next mini-batch fault samples
  Train the temporary CNN model and obtain its reward r_{k+1}
  Store $(s_k, a_k, r_{k+1}, s_{k+1})$ in D'
end for

The random walk procedure is almost the same with training the CNN model, as shown in Algorithm 2. However, it applies random action to replace the prediction action of DDPG. This approach is helpful for exploring the training space. The CNN model, mini-batch indicator, and experience storage are cloned to a temporary version and used only in the random walk procedure. The experience storage used for training the critic and actor networks is D' in the random walk procedure.

Algorithms 1 and 2 present that ACNN can estimate its batch size, learning rate, and L2-regulation online by the LSTM–DDPG. These three factors are key hyper parameters that should be tuned during CNN training. Thus, the proposed ACNN can automatically self-control these three values.

5 Case studies and results

The proposed ACNN method is released using TensorFlow 1.15, and the DDPG part is implemented using TRFL 1.1.0 package, which is compatible with TensorFlow. The proposed ACNN runs on Ubuntu Linux 18.04 with RTX 2080Ti GPU.

5.1 Experiment setup

ACNN is conducted on the motor bearing dataset from Case Western Reserve University, USA (CWRU dataset) and KAT data center in Paderborn University, Germany (KAT dataset). These two datasets are widely used to verify the potential in the fault diagnosis field.

First, the proposed ACNN is compared with four HPO methods, namely, RS, BO, TPE, and SMAC. RS and TPE are implemented based on Hyperopt software. BO is based on Scikit-Optimize. SMAC is implemented by

SMAC3, which is developed by the AutoML group. Hyperopt, Scikit-Optimize, and SMAC3 are all famous software for HPO.

Second, the results of the proposed ACNN are validated by comparing them with other reported ML and DL methods in the literature to show its potential on fault diagnosis further.

The ranges for the hyper parameters for the tuning are set, as shown in Table 1. As the learning rate of the ACNN is controlled by the LSTM–DDPG, the decay of learning rate is eliminated in the ACNN. The hyper parameter setting of ACNN, Scikit-Optimize, and Hyperopt are the same. The training method is stochastic gradient descent with moment. *step4game* is 5.

5.2 Case study 1: CWRU motor bearing dataset

The bearing dataset provided by CWRU [34] is selected to validate the proposed ACNN. In this dataset, the bearing type is 6205-2RSJEM SKF. The fault type contains roller fault (RF), outer race fault (OF), inner race fault (IF), and the normal condition. Each fault type has three different damage sizes, namely, 0.18, 0.36, and 0.54 mm, for a total of 10 health conditions, which include nine fault conditions and the normal condition. During the experiments, the vibration signals are used to analyze the fault classification. The sampling frequency is 12 kHz. The working conditions have four different loads, namely, 0, 1, 2, and 3 hp (1 hp = 735 W).

Table 1 Hyper parameter range for ACNN

Hyper parameter	Range
Learning rate	$(10^{-4}, 10^{-2})$
Decay	(0.99, 0.995)
Batch size	(10, 100)
L2-regulation value	$(10^{-11}, 10^{-3})$

The task configuration comes from the Zhu et al. [34]. The notation of '1→2' means that ACNN is trained on load 1 and tested on load 2, and the accuracy on 1→2 is denoted as Acc(1→2). The six configurations are 1→2, 1→3, 2→1, 2→3, 3→1, and 3→2. The learning task is the compare the average prediction accuracy (AVG) on the cross working load, and AVG values of these six configurations are used for this case study.

5.2.1 Comparisons between ACNN and HPO methods

ACNN is compared with four famous HPO methods. DenseNet-201 is used as the default backbone for the comparison, and its results are shown in Table 2. Table 2 presents the detail comparisons of the ACNN and HPO methods on all six configurations as well as the AVG value. The comparison results with other backbones are presented in Table 3, which presents the AVG results of the ACNN and HPO methods on different backbones.

Table 2 shows that ACNN achieves four best results out of six configurations. The final AVG of the ACNN also outperforms that of the other HPO methods. Table 3 shows that the AVG results of ACNN outperform those of RS, TPE, BO, and SMAC on the five other backbones. These results can validate that ACNN is competitive than these HPO methods in fault diagnosis.

To investigate the convergence of ACNN, the curves of learning rate, batch size, L2-regulation values, and reward of the ACNN are presented in Fig. 2. All the curves initially fluctuate greatly, showing that the DRL agent (the LSTM–DDPG described in Section 3.3) tries to find the optimal values for these three hyper parameters for the ACNN. Then, all the curves gradually return to a stable state. ACNN controls the learning rate, batch size, and L2-regulation value according to its historical training information. Thus, these results mean that ACNN learns a good DRL agent to control them, and all three hyper parameters converge to stable values.

Table 2 Comparison results of ACNN with HPO methods using DenseNet-201 in Case 1

Method	Acc(1→2)/%	Acc(1→3)/%	Acc(2→1)/%	Acc(2→3)/%	Acc(3→1)/%	Acc(3→2)/%	AVG/%
RS	97.47	95.24	96.79	98.81	92.86	98.67	96.639
TPE	97.48	95.19	96.81	98.81	92.91	98.72	96.653
BO	97.69	95.33	96.91	98.93	92.67	98.60	96.687
SMAC	97.60	95.21	96.82	98.90	92.92	98.67	96.685
ACNN	98.10	95.76	97.20	99.11	91.96	98.64	96.793

Table 3 Comparison results of ACNN with HPO methods using other backbones in Case 1

CNN backbone	Acc(RS)/%	Acc(TPE)/%	Acc(BO)/%	Acc(SMAC)/%	Acc(ACNN)/%
ResNet-50V2	94.746	94.788	94.871	94.896	94.910
ResNet-101V2	95.592	95.632	95.787	95.796	96.226
ResNet-152V2	94.095	94.075	94.206	94.184	94.396
DenseNet-121	95.065	95.039	95.41	95.288	96.314
DenseNet-169	93.006	93.002	93.253	93.096	93.903

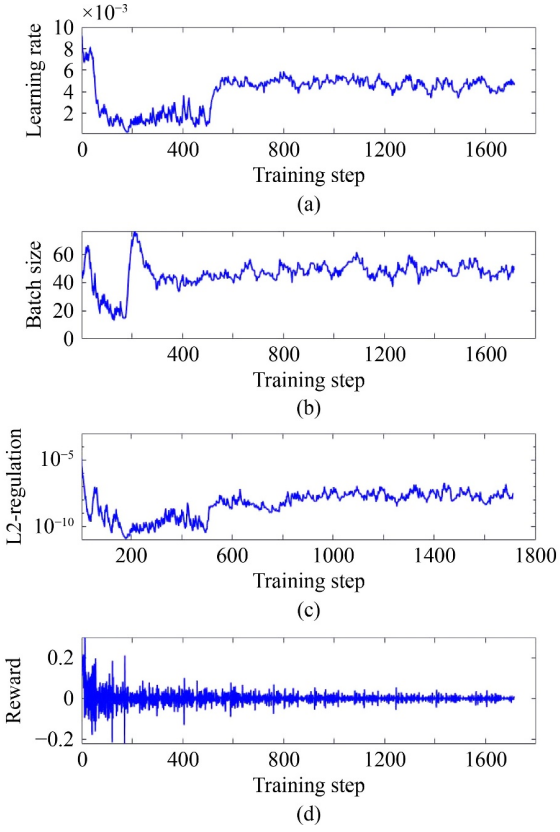

Fig. 2 Online curves of ACNN in Case 1: (a) learning rate curve, (b) batch size curve, (c) L2-regularization curve, and (d) reward curve.

In this research, reward is defined as the improvement of training loss of the ACNN model, so reward is the good indicator to show the convergence of the ACNN and the DRL agent. Figure 2 shows that the reward also converges to near 0, showing that the DRL agent and ACNN converge well.

5.2.2 Comparison between ACNN and ML and DL

The proposed ACNN is compared with published famous ML and DL methods in the literature. The compared methods are capsule network based on wide convolution and multiscale convolution (WMSCCN) [35], hierarchical CNN (HCNN) [36], deep convolutional neural networks with wide first-layer kernels (WDCNN) [37], CNN based on vibration image (VI-CNN) [38], multiscale CNN (MSCNN) [39], and adaptive weighted multiscale CNN (AWMS-CNN) [40]. The ML methods are also adopted as the baseline methods, including SVM and ANN. These two ML methods using fast Fourier transform as the data pre-processing method. The comparison results are shown in Table 4.

The results in Table 4 show that ACNN is superior to WMSCCN, HCNN, WDCNN, AWMS-CNN, and VI-CNN. The AVG values of these methods show that ACNN achieves good prediction results. ACNN is also compared with SVM and ANN. The results show that the great improvement is achieved by ACNN compared with SVM and ANN. These results can validate the potential of ACNN in this case study.

5.3 Case study 2: Paderborn University bearing dataset

ACNN is conducted on the bearing dataset from KAT data center in Paderborn University, Germany (KAT dataset) [34]. The KAT dataset contains three health conditions, namely, normal condition, outer ring with damage, and inner ring with damage. The type of bearing is ball bearing of type 6203. The vibration signals are used, and the sampling frequency is 64 kHz. The operation conditions can be found in Ref. [34], and the three conditions are denoted as 5, 6, and 7. The task in Case 2 is from Zhu et al. [34], and the AVG of six configurations, namely, 5→6, 5→7, 6→5, 6→7, 7→5, and 7→6, is used to evaluate the final performance of ACNN. The denotation of "5→6" is the same with Case 1.

5.3.1 Comparisons between ACNN and HPO methods

The ACNN and HPO methods are compared in Tables 5 and 6. In this section, DenseNet-201 is used as the default backbone similar to Case 1. The comparison between

Table 4 Comparison results of ACNN with ML and DL in Case 1

Method	Acc(1→2)/%	Acc(1→3)/%	Acc(2→1)/%	Acc(2→3)/%	Acc(3→1)/%	Acc(3→2)/%	AVG/%
ACNN	98.10	95.76	97.20	99.11	91.96	98.64	**96.793**
WMSCCN	95.17	97.24	98.79	96.55	93.10	97.76	96.435
HCNN	99.93	98.79	95.15	99.45	89.33	93.69	96.100
WDCNN	98.10	92.59	91.90	96.73	86.21	91.03	92.760
MSCNN	94.14	93.28	90.86	95.17	84.66	90.69	91.467
AWMS-CNN	93.82	86.73	93.10	91.03	85.52	93.62	90.637
VI-CNN	89.60	80.27	88.43	88.03	79.83	83.80	84.993
SVM	61.03	60.17	74.14	60.69	61.90	62.24	63.362
ANN	88.10	85.00	77.93	88.45	87.24	86.03	85.458

Table 5 Comparison results of ACNN with HPO methods using DenseNet-201 in Case 2

Method	Acc(5→6)/%	Acc(5→7)/%	Acc(6→5)/%	Acc(6→7)/%	Acc(7→5)/%	Acc(7→6)/%	AVG/%
RS	88.36	96.75	90.42	90.23	98.23	88.54	92.090
TPE	88.51	96.78	90.39	90.20	98.22	88.72	92.135
BO	88.14	97.17	90.81	90.32	98.41	88.33	92.195
SMAC	88.49	96.76	90.77	90.30	98.37	88.84	92.253
ACNN	85.78	97.88	93.14	92.33	98.75	88.16	92.671

Table 6 Comparison results of ACNN with HPO methods using other backbones in Case 2

CNN backbone	Acc(RS)/%	Acc(TPE)/%	Acc(BO)/%	Acc(SMAC)/%	Acc(ACNN)/%
ResNet-50V2	92.241	92.220	92.433	92.428	92.940
ResNet-101V2	93.036	92.987	93.294	93.194	94.088
ResNet-152V2	87.368	87.305	88.146	87.737	90.026
DenseNet-121	89.407	89.376	89.650	89.639	90.993
DenseNet-169	90.286	90.139	90.461	90.385	91.255

ACNN and HPO methods is presented in Table 5, whereas the comparison on five other backbones is presented on Table 6.

Table 5 shows that ACNN obtains the best results on 5→7, 6→5, 6→7, and 7→5 configurations. Table 6 compares ACNN and HPO methods on other CNN models. The results show that ACNN obtains better results than RS, TPE, BO, and SMAC on all CNN models, indicating that ACNN achieves good results in this case study.

Figure 3 presents the curves of learning rate, batch size, L2-regulation, and reward. The values of learning rate, batch size, and L2-regulation constantly change in the early stage but eventually converge to a stable state. The reward curve is almost the same. It initially fluctuates but finally converges near 0. These curves validate the convergence of the DRL agent and the CNN models in the proposed ACNN in this case study.

5.3.2 Comparison between ACNN and ML and DL

In this subsection, ACNN is compared with ML and DL to show its effectiveness. The selected ML and DL methods are WDCNN [34], CNN based on a capsule network with an inception block (ICN) [34], domain adaption network (DAN) [41], ResNet [34], AlexNet [34], SVM [41], and extreme learning machine (ELM) [41]. The comparison results are shown in Table 7.

The results show that ACNN achieves the best AVG among these eight famous DL and ML methods. The results show that the performance of ACNN improves much, which provides solid support to validate potential of ACNN on fault diagnosis in this case study.

5.4 Discussion

In this research, a new DRL-based ACNN for fault diagnosis is proposed. In this method, an agent-based DDPG is developed, and it can automatically control three key hyper parameters during the training of CNN models, namely, batch size, learning rate, and L2-regulation.

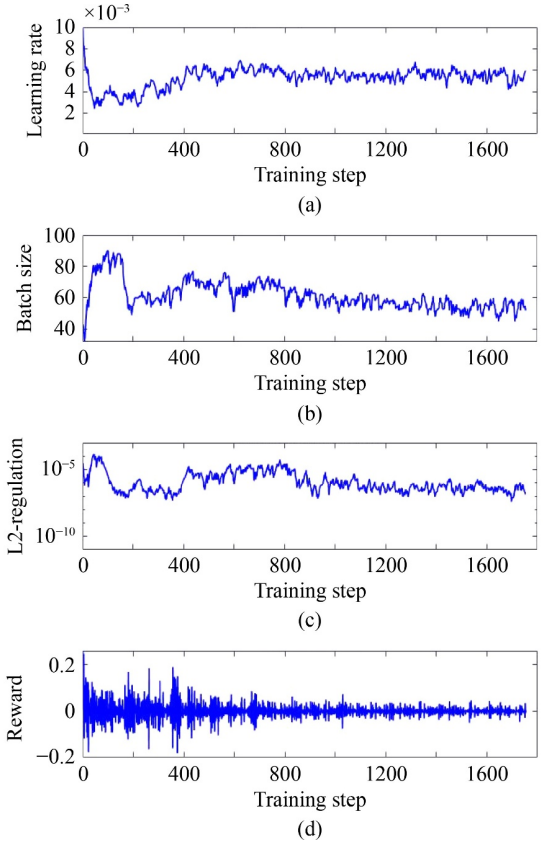

Fig. 3 Online curves of ACNN in Case 2: (a) learning rate curve, (b) batch size curve, (c) L2-regularization curve, and (d) reward curve.

First, a new structure of DRL is proposed and named LSTM–DDPG. This structure can take the training loss of CNN as its input to control the three hyper parameters on

Table 7 Comparison results of ACNN with ML and DL in Case 2

Method	Acc(5→6)/%	Acc(5→7)/%	Acc(6→5)/%	Acc(6→7)/%	Acc(7→5)/%	Acc(7→6)/%	AVG/%
ACNN	**85.78**	**97.88**	**93.14**	**92.33**	**98.75**	**88.16**	**92.671**
WDCNN	72.33	94.70	69.33	69.77	93.67	70.27	78.35
ICN	80.67	96.97	70.23	70.67	94.27	79.50	82.05
DAN	85.70	98.40	81.58	89.29	98.00	90.50	90.58
ResNet	71.33	96.67	64.53	67.23	92.73	72.60	77.52
AlexNet	78.87	98.47	65.93	66.20	96.03	74.07	79.92
SVM	56.25	68.63	54.45	53.32	68.65	56.10	59.56
ELM	39.28	39.07	39.18	38.92	39.00	38.43	38.98

each training step. The results show that the reward of the proposed LSTM–DDPG can converge well, and the three parameters all converge to the stable state, indicating the proposed method is feasible and efficient.

Second, the proposed method is tested on two famous datasets, namely, the CWRU motor bearing dataset and the Paderborn University bearing dataset. The results of ACNN are compared with other DL methods published in recent years. The comparison results show that ACNN achieves a good performance on fault diagnosis, showing its potential.

6 Conclusions

This paper presents a new ACNN for fault diagnosis. The main contributions can be summarized as follows:

1) A new DRL paradigm is developed to control the three hyper parameters (batch size, learning rate, and L2-regulation) of the ACNN model online.

2) A new LSTM–DDPG structure is designed, which can take the training loss of the ACNN models as input and automatically adjust these three hyper parameters.

3) A new training method for ACNN with random walk is developed to enhance the search ability of ACNN. The proposed ACNN is conducted on the CWRU and KAT datasets, and the results validate its performance.

The proposed ACNN is compared with four famous HPO methods and the published ML and DL. The results indicate that ACNN achieves good results in the fault diagnosis field.

The limitations of the ACNN method are as follows:

1) It can only control the hyper parameters of the CNN models, and it ignores the optimization on the CNN structure.

2) The DRL agent is trained along with the CNN model. Thus, it should be re-trained for each run, and it cannot be shared and re-used.

Based on the limitations, future research can be organized in the following ways. First, NAS can be studied to find the optimal CNN structure. Second, the spirit of transfer learning can be applied in the DRL agent in ACNN to enable its reuse, which can promote the performance of ACNN further.

Nomenclature

Abbreviations

ACNN	Automatic convolutional neural network
ANN	Artificial neural network
AVG	Average prediction accuracy
AWMS-CNN	Adaptive weighted multiscale convolutional neural network
BO	Bayesian optimization
CNN	Convolutional neural network
DAN	Domain adaption network
DDPG	Deep deterministic policy gradient
DL	Deep learning
DRL	Deep reinforcement learning
ELM	Extreme learning machine
FC	Fully connected
GS	Grid search
HCNN	Hierarchical convolutional neural network
HPO	Hyper parameter optimization
ICN	CNN based on a capsule network with an inception block
IF	Inner race fault
LSTM	Long short-term memory
ML	Machine learning
MSCNN	Multiscale convolutional neural network
NAS	Neural architecture search
OF	Outer race fault
RF	Roller fault
RL	Reinforcement learning
RS	Random search
SMAC	Sequential model-based algorithm configuration
SVM	Support vector machine
TPE	Tree Parzen estimator
VI-CNN	CNN based on vibration image
WDCNN	Deep convolutional neural networks with wide first-layer kernels
WMSCCN	Wide convolution and multiscale convolution

Variables

a	Action of DDPG algorithm
a_t	Current action
b_t	Current batch size
$b_{\max}$, $b_{\min}$	The upper and lower boundaries for batch size
E_{Π}	Expected value under policy Π
f	Fourier frequency in short-time Fourier transform
l_t	Current L2-regulation value
$l_{\max}$, $l_{\min}$	The upper and lower boundaries for L2-regulation value
$lr_{\max}$, $lr_{\min}$	The upper and lower boundaries for learning rate, respectivley
lr_t	Current learning rate
$loss_t$	Loss value of the CNN model at time step t
L	Training loss of critic network
M	Number of the sequencing training loss
n	Number of samples in the experience storage D
p	Transition probability function
$Q^{\Pi}(s, a)$	Q-value function under policy Π
r	Reward of DDPG algorithm
s	State of DDPG algorithm
s_t	State at time step t
$STFT$	Short-time Fourier transform formulation
t	Time step
$w(t)$	Window function
y_t	Actual Q-value
α	Factor to control the degree of soft updating
γ	Discount factor
Π	Policy of the agent to choose the action
θ^{μ}	Online actor network
$\theta^{\mu'}$	Target actor network
ω^{Q}	Online critic network
$\omega^{Q'}$	Target critic network

Acknowledgements This work was supported in part by the National Natural Science Foundation of China (Grant Nos. 51805192 and U21B2029), the Major Special Science and Technology Project of Hubei Province, China (Grant No. 2020A EA009), and the State Key Laboratory of Digital Manufacturing Equipment and Technology of Huazhong University of Science and Technology, China (Grant No. DMETKF2020029).

References

1. Zhang X, Huang T, Wu B, Hu Y M, Huang S, Zhou Q, Zhang X. Multi-model ensemble deep learning method for intelligent fault diagnosis with high-dimensional samples. Frontiers of Mechanical Engineering, 2021, 16(2): 340–352
2. Chen X F, Wang S B, Qiao B J, Chen Q. Basic research on machinery fault diagnostics: past, present, and future trends. Frontiers of Mechanical Engineering, 2018, 13(2): 264–291
3. Lei Y G, Yang B, Jiang X W, Jia F, Li N P, Nandi A K. Applications of machine learning to machine fault diagnosis: a review and roadmap. Mechanical Systems and Signal Processing, 2020, 138: 106587
4. Nath A G, Udmale S S, Singh S K. Role of artificial intelligence in rotor fault diagnosis: a comprehensive review. Artificial Intelligence Review, 2021, 54: 2609–2668
5. Wang J L, Xu C Q, Dai L, Zhang J, Zhong R Y. An unequal deep learning approach for 3-D point cloud segmentation. IEEE Transactions on Industrial Informatics, 2021, 17(12): 7913–7922
6. Chen Z Y, Mauricio A, Li W H, Gryllias K. A deep learning method for bearing fault diagnosis based on cyclic spectral coherence and convolutional neural networks. Mechanical Systems and Signal Processing, 2020, 140: 106683
7. Wolpert D H, Macready W G. No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 67–82
8. Wolpert D H. The supervised learning no-free-lunch theorems. In: Roy R, Köppen M, Ovaska S, Furuhashi T, Hoffmann F, eds. Soft Computing and Industry. London: Springer, 2002, 25–42
9. Hutter F, Kotthoff L, Vanschoren J. Automated Machine Learning: Methods, Systems, Challenges. Cham: Springer, 2019
10. Wen L, Li X Y, Gao L. A new reinforcement learning based learning rate scheduler for convolutional neural network in fault classification. IEEE Transactions on Industrial Electronics, 2021, 68(12): 12890–12900
11. Wen L, Ye X C, Gao L. A new automatic machine learning based hyperparameter optimization for workpiece quality prediction. Measurement and Control, 2020, 53(7−8): 1088–1098
12. Feurer M, Eggensperger K, Falkner S, Lindauer M, Hutter F. Practical automated machine learning for the AutoML challenge 2018. In: Proceedings of International Workshop on Automatic Machine Learning at ICML. 2018, 1189–1232
13. He F X, Liu T L, Tao D C. Control batch size and learning rate to generalize well: theoretical and empirical evidence. In: Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS). Vancouver, 2019, 1143–1152
14. Li Y Z, Wei C, Ma T Y. Towards explaining the regularization effect of initial large learning rate in training neural networks. In: Proceedings of the 33rd International Conference on Neural Information Processing Systems. Vancouver, 2019, 11674–11685
15. Zhou P, Peng Z K, Chen S Q, Yang Y, Zhang W M. Non-stationary signal analysis based on general parameterized time-frequency transform and its application in the feature extraction of a rotary machine. Frontiers of Mechanical Engineering, 2018, 13(2): 292–300
16. Wang J L, Xu C Q, Yang Z L, Zhang J, Li X O. Deformable convolutional networks for efficient mixed-type wafer defect pattern recognition. IEEE Transactions on Semiconductor Manufacturing, 2020, 33(4): 587–596
17. Xu G W, Liu M, Jiang Z F, Shen W M, Huang C X. Online fault diagnosis method based on transfer convolutional neural networks. IEEE Transactions on Instrumentation and Measurement, 2020, 69(2): 509–520
18. Li Z X, Zheng T S, Wang Y, Cao Z, Guo Z Q, Fu H Y. A novel

method for imbalanced fault diagnosis of rotating machinery based on generative adversarial networks. IEEE Transactions on Instrumentation and Measurement, 2021, 70: 3500417

19. Chen J B, Huang R Y, Zhao K, Wang W, Liu L C, Li W H. Multiscale convolutional neural network with feature alignment for bearing fault diagnosis. IEEE Transactions on Instrumentation and Measurement, 2021, 70: 3517010
20. Jiao J Y, Zhao M, Lin J, Liang K X. A comprehensive review on convolutional neural network in machine fault diagnosis. Neurocomputing, 2020, 417: 36–63
21. Yao Y, Zhang S, Yang S X, Gui G. Learning attention representation with a multi-scale CNN for gear fault diagnosis under different working conditions. Sensors, 2020, 20(4): 1233
22. Li S, Wang H Q, Song L Y, Wang P X, Cui L L, Lin T J. An adaptive data fusion strategy for fault diagnosis based on the convolutional neural network. Measurement, 2020, 165: 108122
23. Kolar D, Lisjak D, Pająk M, Pavković D. Fault diagnosis of rotary machines using deep convolutional neural network with wide three axis vibration signal input. Sensors, 2020, 20(14): 4017
24. Wang R X, Jiang H K, Li X Q, Liu S W. A reinforcement neural architecture search method for rolling bearing fault diagnosis. Measurement, 2020, 154: 107417
25. Zhang K Y, Chen J L, He S L, Xu E Y, Li F D, Zhou Z T. Differentiable neural architecture search augmented with pruning and multi-objective optimization for time-efficient intelligent fault diagnosis of machinery. Mechanical Systems and Signal Processing, 2021, 158: 107773
26. Cabrera D, Guamán A, Zhang S H, Cerrada M, Sánchez R V, Cevallos J, Long J Y, Li C. Bayesian approach and time series dimensionality reduction to LSTM-based model-building for fault diagnosis of a reciprocating compressor. Neurocomputing, 2020, 380: 51–66
27. Li L, Jamieson K, DeSalvo G, Rostamizadeh A, Talwalkar A. Hyperband: a novel bandit-based approach to hyperparameter optimization. The Journal of Machine Learning Research, 2018, 18(1): 6765–6816
28. Li H, Zhang Q, Qin X R, Sun Y T. Raw vibration signal pattern recognition with automatic hyper-parameter-optimized convolutional neural network for bearing fault diagnosis. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 2020, 234(1): 343–360
29. Long J Y, Zhang S H, Li C. Evolving deep echo state networks for intelligent fault diagnosis. IEEE Transactions on Industrial Informatics, 2020, 16(7): 4928–4937
30. Han J H, Choi D J, Park S U, Hong S K. Hyperparameter optimization using a genetic algorithm considering verification time in a convolutional neural network. Journal of Electrical Engineering & Technology, 2020, 15(2): 721–726
31. Wei J A, Huang H S, Yao L G, Hu Y, Fan Q S, Huang D. New imbalanced fault diagnosis framework based on cluster-MWMOTE and MFO-optimized LS-SVM using limited and complex bearing data. Engineering Applications of Artificial Intelligence, 2020, 96: 103966
32. Hansen S. Using deep Q-learning to control optimization hyperparameters. 2016, arXiv:1602.04062
33. Zhang Z Z, Chen J L, Chen Z B, Li W P. Asynchronous episodic deep deterministic policy gradient: toward continuous control in computationally complex environments. IEEE Transactions on Cybernetics, 2021, 51(2): 604–613
34. Zhu Z Y, Peng G L, Chen Y H, Gao H J. A convolutional neural network based on a capsule network with strong generalization for bearing fault diagnosis. Neurocomputing, 2019, 323: 62–75
35. Wang Y, Ning D J, Feng S L. A novel capsule network based on wide convolution and multi-scale convolution for fault diagnosis. Applied Sciences, 2020, 10(10): 3659
36. Wen L, Li X Y, Gao L. A new two-level hierarchical diagnosis network based on convolutional neural network. IEEE Transactions on Instrumentation and Measurement, 2020, 69(2): 330–338
37. Zhang W, Peng G L, Li C H, Chen Y H, Zhang Z J. A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals. Sensors, 2017, 17(2): 425
38. Hoang D T, Kang H J. Rolling element bearing fault diagnosis using convolutional neural network and vibration image. Cognitive Systems Research, 2019, 53: 42–50
39. Jiang G Q, He H B, Yan J, Xie P. Multiscale convolutional neural networks for fault diagnosis of wind turbine gearbox. IEEE Transactions on Industrial Electronics, 2019, 66(4): 3196–3207
40. Qiao H H, Wang T Y, Wang P, Zhang L, Xu M D. An adaptive weighted multiscale convolutional neural network for rotating machinery fault diagnosis under variable operating conditions. IEEE Access: Practical Innovations, Open Solutions, 2019, 7: 118954–118964
41. Song Y, Li Y B, Jia L, Qiu M K. Retraining strategy-based domain adaption network for intelligent fault diagnosis. IEEE Transactions on Industrial Informatics, 2020, 16(9): 6163–6171