

Bearing Fault Detection with a Deep Light Weight CNN

Jin Woo Oh and Jongpil Jeong(✉)

Department of Smart Factory Convergence, Sungkyunkwan University, Jangan-gu, Suwon, Gyeonggi-do 16419, Republic of Korea
{itdojust,jpjeong}@skku.edu

Abstract. Bearings are vital part of rotary machines. A failure of bearing has a negative impact on schedules, production operation and even human casualties. Therefore, in prior achieving fault detection and diagnosis (FDD) of bearing is ensuring the safety and reliable operation of rotating machinery systems. However, there are some challenges of the industrial FDD problems. Since according to a literature review, more than half of the broken machines are caused by bearing fault. Therefore, one of the important thing is time delay should be reduced for FDD. However, due to many learnable parameters in model and data of long sequence, both lead to time delay for FDD. Therefore, this paper proposes a deep Light Convolutional Neural Network (LCNN) using one dimensional convolution neural network for FDD.

Keywords: Data augmentation · CNN · Light · Fault diagnosis · Bearing

1 1 Introduction

Failures in rotating machinery such as helicopters and wind turbines is the problem of rolling element bearings (REBs). According to a literature review, more than half of the broken machines are caused by bearing faults [1]. Therefore, in prior achieving fault detection and diagnosis (FDD) of bearings is significant tasks as well as it is needed to collect and preprocessing the vibration signals for monitoring bearing condition. Many researchers have studied FDD for collecting, preprocessing and predicting the life of mechanical systems and so these field has become an important research area in industry [2,3]. Machines run almost always reliably, and few defects occur during stable operation of the control process, such as mechanical failures in manufacturing [4,5]. However, the existing traditional data-driven algorithms have a problem in that they are not good at learning the huge number of various and nonlinear data. In addition, due to the nature of the process, if the deformation of the process is slightly different, the data also can be different. This has to be revised each time by experienced experts.

Supported by organization x.

© Springer Nature Switzerland AG 2020
O. Gervasi et al. (Eds.): ICCSA 2020, LNCS 12250, pp. 604–612, 2020.
https://doi.org/10.1007/978-3-030-58802-1_43

Instead of traditional machine learning algorithms, data-driven deep learning algorithms have ability of automatically learning the discriminative feature representation from input data effectively and accurately. As one of the popular models of deep neural network (DNN) is CNN, it has been applied in many fields, such as recognizing image and speech analysis [6–8]. However, many parameters of the model require a lot of time to process complex data generated in the industry and to detect and diagnose faults. Therefore, as mentioned above, this paper proposes a deep light weight CNN model.

The remainder of this paper is structured as follows. In Sect. 2, Overall description of the CNN is introduced briefly. In Sect. 3, the proposed fault diagnosis method based on deep light weight CNN is explained. In Sect. 4, The effectiveness of the proposed method is described. Finally, the conclusions are drawn in Sect. 5.

2 Convolutional Neural Network

Convolutional neural network (CNN) mainly consist of convolutional layer (CL), pooling layer (PL) and fully-connected layer (FL) as well as activation and loss functions. In the CL, a number of filters are used to perform filtering of input time series data and the obtained feature maps are overlapped to form an output feature map of the CL. Convolution operations in CL have translation invariant that means translation invariant makes the CNN invariant to translation. Invariance to translation means that if we translate the inputs the CNN will still be able to detect the class to which the input belongs.

PL extracts the fixed length feature vector from the feature maps for feature dimension reduction. Consequently, CL can extract important features and simplify the complexity of the network computation. PL provide an approach to down sampling feature maps by summarizing the presence of features in patches of the feature map. Two common pooling methods are average pooling and max pooling that summarize the average presence of a feature and the most activated presence of a feature respectively.

FL in a neural networks are those layers where all the inputs from one layer are connected to every activation unit of the next layer. In most popular machine learning models, the last few layers are full connected layers which compiles the data extracted by previous layers to form the final output. It is the second most time consuming layer second to CL.

As per the published literature [9,10], a neural network is referred to as shallow if it has single FL. Whereas, a deep CNN consists of CL, PL, and FC layers. In this paper, we assume a CNN model N1 as deep/shallow compared to another CNN model N2, if the number of trainable layers in N1 is more/less than N2, respectively.

2.1 One Dimensional and Two Dimensional CNN

One dimensional neural network (1D CNN) can be applied to time series analysis of sensor data [11,12]. Compared to one-dimensional processing, two-dimensional

processing has a relatively complex structure and also requires more time and calculation resources. For example, the computational load of 3×1 convolution is only one-third compared to 3×3 convolution.

CNNs share the same characteristics regardless of 1D, 2D or 3D and follow the same approach. The main difference is in the dimensions of the input data and the way the feature detector (or filter) slices across the data. The difference can be seen in Fig. 1, while the 1D CNN filter learns the features of stride 1 all the way down, while the 2D CNN slicing horizontally and vertically filters the data with RGB. It can be seen that the difference remains in terms of the amount of computation brought by the difference in learning operations.

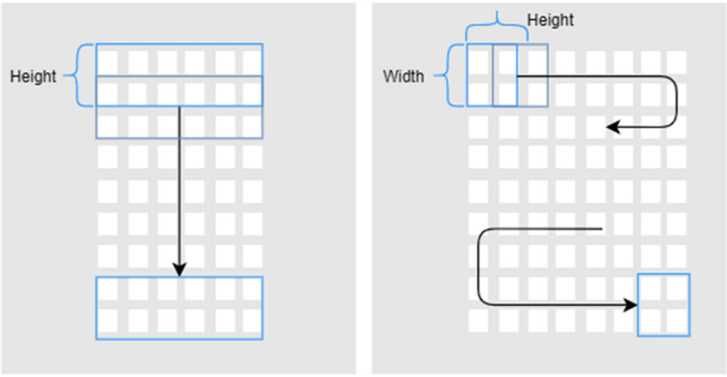

Fig. 1. Difference between 1D CNN and 2D CNN

3 The Proposed Methods

In this paper, a deep Light Convolutional Neural Network (CNN) model for rotating machinery fault detection is designed based on 1D CNN.

3.1 Model Structure

The parameters and computational time in a proposed CNN are affected by the number of channels, filters, data length, stride size. As shown in Fig. 2, the proposed model structure have conv-1 to conv-4 convolution layers of the network include convolution operations, rectified linear unit (ReLU) activation functions. The use of a ReLU activation function increase the rate of convergence and prevent a gradient explosion and vanishing problems. Maximum pooling is used in the pool-1 to pool-4 to make model have translation invariant and reduce the dimensionality of the data.

In FL, in dense layer, we use only use 10 nodes. In general, it is demonstrated in [13] that the FL needs a lot of nodes to get better performance, but a model with a large number of CLs and PLs needs fewer neurons in the FC.

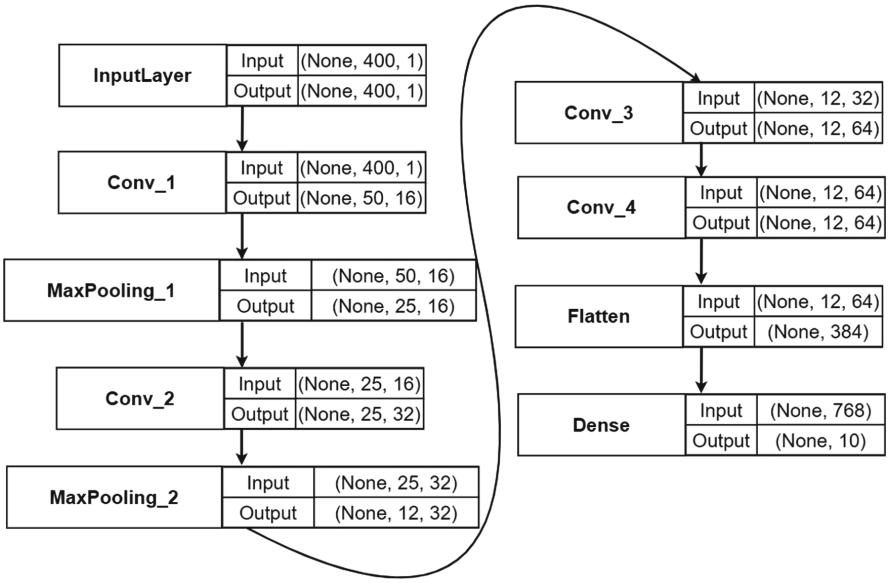

Fig. 2. The propose model overall structure

3.2 Experimental Model and Its Comparison

As mentioned in Sect. 3.1, depend on the number of filters, filter size, data length, data shape and connection mode, the performance of the model and the amount of space occupied can be vary. To demonstrate the effectiveness of proposed deep light CNN using the 1D-CNN as shown in Fig. 2, we use basic model of 1D CNN, 2D CNN and reference [14] for the performance comparison. The layers of the models were set identically and possibly. The name of models and its layers can be seen in Table 1.

The proposed model is designed to have the lightness and high performance of the model as much as possible. Therefore, in order to numerically confirm that the proposed model has achieved its purpose, the computation of the floating numbers and contain parameters of the convolutional layers and fully connected layers can be calculated by the following equations:

$$Parameters_{conv} = K_h \times K_w \times C_{in} \times C_{out} + C_{out} \tag{1}$$

$$Params_{sfc} = I \times O + O \tag{2}$$

$$FLOP_{conv} = 2 \times H \times W \times K_h \times K_w \times C_{in} \times C_{out} \tag{3}$$

$$FLOP_{sfc} = 2 \times I \times O \tag{4}$$

Table 1. Comparison between the proposed model and other models

	Proposed Model	1D-CNN	2D-CNN	Reference [14]
Input length	400 × 1	400 × 1	400 × 1	400 × 1
Layer1	Conv1d-16	Conv1d-32	Conv2d-32	Conv1d-32
Layer2	maxPool-16	maxPool-32	maxPool-32	maxPool-32
Layer3	Conv1d-32	Conv1d-64	Conv2d-64	Conv1d-5, Conv1d-5
				Conv1d-5, maxPool-5
Layer4	maxPool-32	maxPool-64	maxPool-64	maxPool-5, Conv1d-5
				Conv1d-5, Conv1d-5
Layer5	Conv1d-64	Conv1d-128	Conv2d-128	maxPool-5, maxPool-5
Layer6	Conv1d-64	maxPool-128	maxPool-128	Dense-100
Layer7	Dense-10	Dense-64	Dense-64	Dense-10
Layer8		Dense-10	Dense-10	
Layer9				

Where $Parameters_{conv}$ and $FLOP_{conv}$ represent the number of parameters and the computation of floating numbers in the convolutional layers, respectively. $Params_{sfc}$ and $FLOP_{sfc}$ represent the number of parameters and the computation of floating numbers in the fully connected layer respectively. H, W and C_{in} are the height, width and number of channels of the input feature map respectively. K_h and K_w represent the size of the convolution filter and C_{out} is the number of convolution filters. I is the dimensionality of the input and O is the dimensionality of the output.

4 Experiments and Analysis

The hardware and software environments are described in Table 2. To evaluate the calculation time, we used CPU for model training.

Table 2. The hardware and software environment.

Hardware environment	Software environment
CPU Intel Core i7-8700K, 3.7 GHz, Six-core twelve threads Programming language Six-core twelve threads Memory 16 GB	Windows TensorFlow 2.0 framework and Python and Python programming language

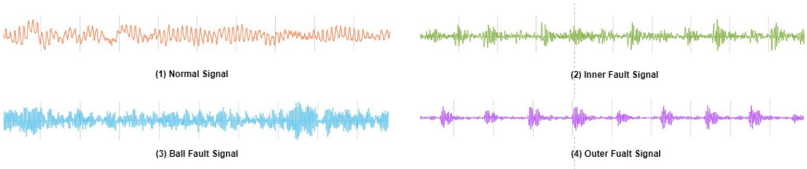

Fig. 3. Vibration signals.

4.1 Data Description

In order to evaluate the performance of the proposed model, data from the Case Western Reserve University Bearing Fault Data is used. Signals of four types can be seen in Fig. 4, including normal, inner race fault, ball fault, outer race fault. After the vibration signals are augmented and then datasets are composed of various signals. The dataset A, B, C and D contain four types of bearing health and the datasets are operated under 0, 1, 2 and 3 hp respectively. For each type of fault, there are three type of diameters, that is, 0.007, 0.014, and 0.021. Segment size of signal for 1D CNN is 400, so each segment contains 400 points. The details of vibration datasets are described in Table 3.

Table 3. Vibration datasets

Data sets	Load condition	Training samples	Validation samples	Test samples	Fault types	Fault diameters	Classification
		1650	192	240	Normal	0	0
		1650	192	240	Ball	0.007	1
		1650	192	240	Ball	0.014	2
A	0 hp	1650	192	240	Ball	0.021	3
B	1 hp	1650	192	240	Inner-race	0.007	4
C	2 hp	1650	192	240	Inner-race	0.014	5
D	3 hp	1650	192	240	Inner-race	0.021	6
		1650	192	240	Outer-race	0.007	7
		1650	192	240	Outer-race	0.014	8
		1650	192	240	Outer-race	0.021	9

4.2 Experimental Result and Analysis

As mentioned in Sect. 3.2, in order to demonstrate the effectiveness of our proposed method, we used the comparison models. There are x training data used in the experiment, x valid data, and x test data. First, the results of models of training results and valid data are shown in Fig. 3. As you can see in Fig. 3, you can see that models are resistant to overfitting and the performance is similar. Since Case Western Reserve University Bearing dataset is simple and well refined, all of models can have good performance. Also it can be seen in Table 4. all models same Precision, Recall, F1-Score but notice that the proposed model have most low Parameters, FLOPs, Training Time(s).

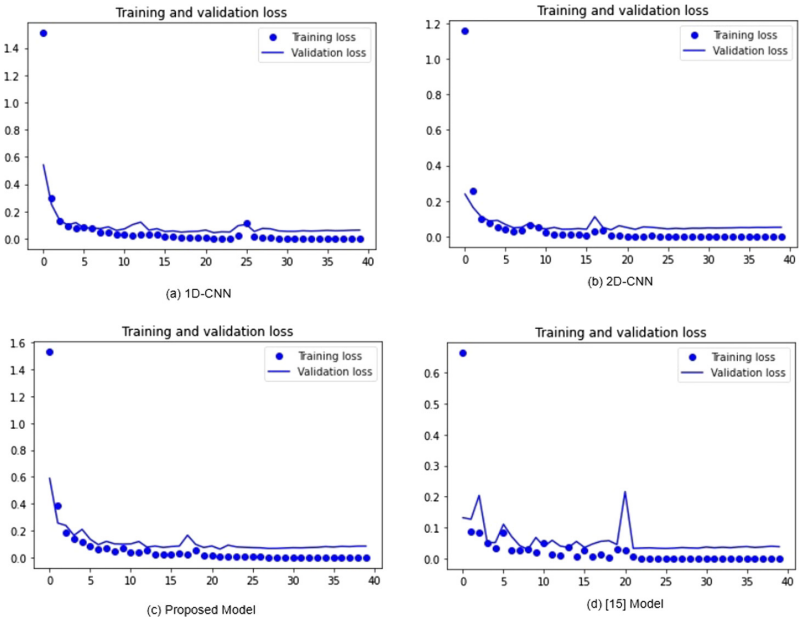

Fig. 4. Comparison training loss and validation loss of models

Table 4. Vibration datasets

Model	Parameters (KB)	FLOPs	Training time (s)	Precision	Recall	F1-score
Proposed model	49	9.08×10^5	124	0.99	0.99	0.99
1D-CNN	190	6.88×10^6	200	0.99	0.99	0.99
2D-CNN	198	1.35×10^7	332	0.99	0.99	0.99
Reference [14]	203	6.81×10^7	433	0.99	0.99	0.99

The classification result of the 10 class of samples is shown in the following confusion matrix in Fig. 5. The identification of each faulty label in each method is represented in the form of a confusion matrix. According to the results from Fig. 5, This represents the accumulated difference between all predictions and actual values in the course of learning. In general, the accuracy of the labels and actual labels predicted by all models has similar performance.

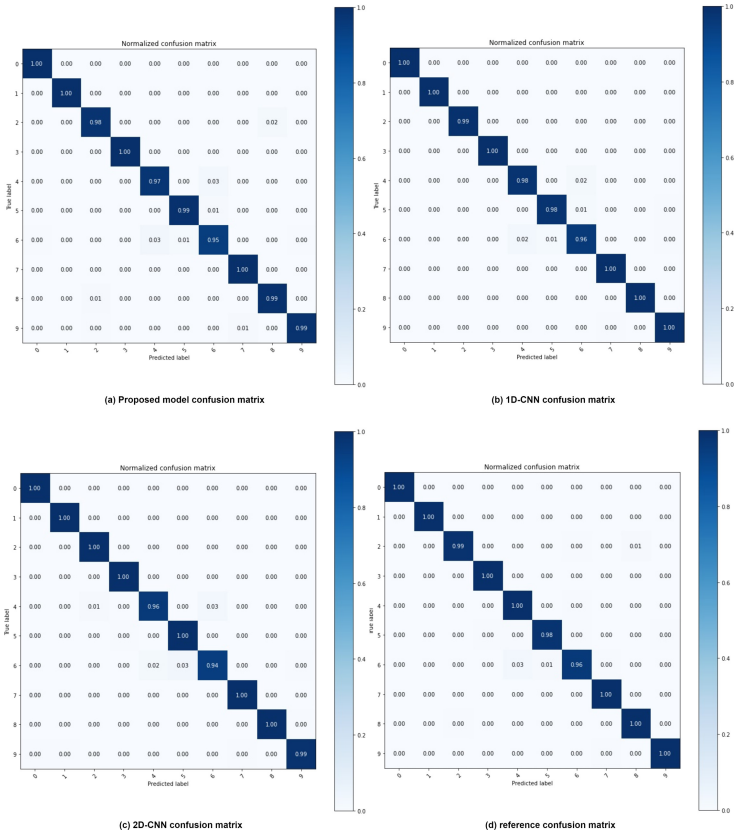

Fig. 5. The comparison results of fault identification using the proposed methods without data augmentation and with augmentation.

5 Conclusion

The proposed model is expected to be able to suggest a direction to solve the delay time problem in the actual process. In particular, the proposed model can significantly reduce the parameters and FLOPs compared to other models. As mentioned 3.1, we can be able to reduce FL nodes and then reduce a lot of parameters in the proposed model.

As a future study, However, there are a few more problems with FDD. Since in industry, process control always run anomaly, the available labels are limited and at the same time there is a challenge to create a model that is resistant to noise in the process. All models were able to achieve good performance in terms of performance, because the data was simple and well-purified, so additional experiments is to prove the performance of the model when complex data and noise-caught data.

Acknowledgments. This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2016R1D1A1B03933828). This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2020-2018-0-01417) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation).

References

1. Nandi, S., Toliyat, H.A., Li, X.: Condition monitoring and fault diagnosis of electrical motors–a review. IEEE Trans. Energy Convers. **20**(4), 719–729 (2005)
2. Liu, X., Choo, K.K.R., Deng, R.H., Lu, R., Weng, J.: Efficient and privacy-preserving outsourced calculation of rational numbers. IEEE Trans. Dependable Secure Comput. **15**(1), 27–39 (2016)
3. Qiao, W., Lu, D.: A survey on wind turbine condition monitoring and fault diagnosis-Part I: components and subsystems. IEEE Trans. Ind. Electron. **62**(10), 6536–6545 (2015). https://doi.org/10.1109/TIE.2015.2422112
4. Kim, J., Caire, G., Molisch, A.F.: Quality-aware streaming and scheduling for device-to-device video delivery. IEEE/ACM Trans. Netw. **24**(4), 2319–2331 (2015)
5. Ranjan, C., Reddy, M., Mustonen, M., Paynabar, K., Pourak, K.: Dataset: rare event classification in multivariate time series. arXiv preprint arXiv:1809.10717 (2018)
6. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)
7. Tompson, J., Goroshin, R., Jain, A., LeCun, Y., Bregler, C.: Efficient object localization using convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 648–656 (2015)
8. Gao, L., Guo, Z., Zhang, H., Xu, X., Shen, H.T.: Video captioning with attention-based LSTM and semantic consistency. IEEE Trans. Multimedia **19**(9), 2045–2055 (2017)
9. Takahashi, N., et al.: Deep convolutional neural networks and data augmentation for acoustic event detection. arXiv preprint arXiv:1604.07160 (2016)
10. Zhang, H., et al.: mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412 (2017)
11. Zhao, R., Yan, R., Wang, J., Mao, K.: Learning to monitor machine health with convolutional bi-directional LSTM networks. Sensors **17**(2), 273 (2017)
12. Zhang, W., Peng, G., Li, C.: Bearings fault diagnosis based on convolutional neural networks with 2-D representation of vibration signals as input. In MATEC web of conferences, vol. 95, p. 13001. EDP Sciences (2017)
13. Basha, S.H.S., et al.: Impact of fully connected layers on performance of convolutional neural networks for image classification. Neurocomputing **378**, 112–119 (2020)
14. Zilong, Z., Wei, Q.: Intelligent fault diagnosis of rolling bearing using one-dimensional multi-scale deep convolutional neural network based health state classification. In: 2018 IEEE 15th International Conference on Networking, Sensing and Control (ICNSC). IEEE (2018)