Edge Computing Opportunities for Vibration Diagnostics of Rotary Machinery Using Neural Network Approach

Valentin Perminov, Vladislav Ermakov, Dmitry Korzun
Petrozavodsk State University (PetrSU)
Petrozavodsk, Russia
{perminov, vlaermak, dkorzun}@cs.petrsu.ru

***Abstract*—Industrial rotary machinery needs real-time diagnostics based on measured data from multiple sensors. Vibration diagnostics can be implemented using the neural network approach, which achieves high accuracy in fault detection. Transferring a neural network model to edge devices leads to performance issues and platform limitations. In this work, we discuss edge computing opportunities for vibration diagnostics of rotary machinery using the neural network approach.**

I. Problem

In order to keep industrial machinery in appropriate condition, the methods of technical condition diagnostics and predictive maintenance are applied. The diagnostics is based on the current machinery state. The predictive maintenance aims at forecasting mechanism behavior at a certain point in time with the current state. Both of these methods found application in IIoT diagnostic systems (Industrial Internet of Things), such as [1]. The utilization of condition diagnostics and predictive maintenance services establishes an effective equipment machinery operation mode and personnel timetable.

The methods need to analyze data from multiple sensors. The data are flow-based, and continuous sensing and processing of different data types are performed:

- mechanical parts vibration, position, speed;
- electric motor current;
- temperature;
- acoustic signals.

Therefore, continuous data fusing is needed, which is now typical in many intelligent system, where Ambient Intelligence is implemented [2].

Data flow from each sensor might be processed with different techniques:

- Spectral Fourier analysis [3];
- Statistic methods based on time series models [4];
- Neural Network models [5].

The first two methods do not analyze relations between various frames in the obtained data [6]. They do not analyze relations between multiple sensors data flow. Methods based on Neural Networks (NN) allow identifying events in the sensor data flow [7], such as bearing treadmill issue. Besides these methods can recognize bursts and non-standard combinations of heterogeneous data, even within normal operating ranges.

Rotary machines have important components, such as bearings and rotors. These components must be monitored online with real-time condition diagnostics systems. The major parameter of those components is vibration. Real-time vibration monitoring aims at quickly responding to problems and provides recommendations to the personnel.

Neural Network-based approach consists of making model, training it on an obtained data set, and deploying in real-time diagnostic systems. There are some types of Neural Network models, such as deep feedforward neural network and deep convolutional neural network. Vibration analysis with that Neural Network models proved themselves in fault detection with a high precision rate [8]. Besides, to increase the prediction rate, we can apply a data fusion-approach that combines heterogeneous data. For data fusion-approach, we need to train our network with an appropriate data set. Here are some tips for data preparation:

1) Sample rate. As described in [8], the data from various sensors must be sampled at the same rate. Otherwise, we should interpolate them to make their number of counts the same.
2) Normalization. For optimal compute performance and precision rate on the training set its recommended to scale values from zero to one.
3) Data augmentation. If there is not enough data for making a training set, then we can use the augmentation method by combining random data frames.
4) Data submission. Work [9] provides two ways of data submission. The first consider making a dataset from raw sensor data with a fixed sample rate and fixed window size. The second one offers to make a dataset spectrogram with FFT as a two-dimensional image. The pixel intensity on this image shows power spectral density, x-axis shows frequency in the spectrum, the y-axis shows spectral sequences in time.
5) Data fuse. To extract more features from heterogeneous data we should concatenate a one-dimension data vector from various sensors to a two-dimensional tensor, where the first dimension corresponds to samples from one sensor and the second one corresponds to the single measurement from different sensors.

II. Neural Network Accelerators in Edge Computing

Deployment of neural networks in real-time diagnostic systems requires high-performance hardware. The conventional approach is to use server-class computers with high-performance GPU. However, increasing interest in NN has motivated many manufacturers to develop application-specific hardware for NN computing, i.e., neural network accelerators (NNA) [10], [11], [12].

To understand the benefits of such hardware we need to consider the structure of NN and corresponding computational operations. The DNN consists of multiple layers. Each layer takes an input data tensor, processes it according to the layer's type, and produces an output data tensor. Layers could be connected sequentially or with recurrent connections. The basic layers types are fully-connected and convolution layers. The fully-connected layer performs matrix multiplication of weights matrix with input data vector and applies activation function, such as tanh, sigmoid, or ReLU. The convolution layer applies multiple convolution kernels (filters) to the input data tensor, then applies activation function similar to a fully-connected layer. The convolution and matrix multiplication are high computational consuming operations. Neural network accelerators aim to improve the efficiency of these operations.

Usually, on the hardware level, neural network accelerators implement only one type of operation mentioned above. The reason is that matrix multiplication could be expressed as multiple convolution operations, and convolution could be expressed as multiple matrix multiplications [13]. Of course, these approaches are less efficient in terms of power efficiency and performance. However, they allow reducing the chip area and cost.

To deploy Neural Networks in embedded systems and edge devices the hardware with neural network accelerators should be used. But, as such hardware designed to be low power and mobile, it has the next limitations.

A. Performance

While data-center NNA could reach performance from dozens to hundreds of TOPS (Tera Operations Per Second) [11], the performance of NNA for edge devices is restricted down to tenth and units of TOPS, as shown in Table I. This circumstance limits the neural network model size in terms of the volume of computational operations and the amount of processed data per second. To cope with this, more thoroughly model architecture selection is required. While compact DC-NNs for image classification are widely investigated [14], [15], developing small NN without a significant precision loss for technical condition diagnostics and predictive maintenance is the subject of future research.

B. Memory

As shown in [11], [18] memory bandwidth often occurs to be a bottleneck of NNA performance. For high-performance data-center NNA, the multi-channel DDR RAM or HBM seems to be the best choice since a very large memory size is required. However, in the case of edge devices, the power consumption and device dimensions are restricted, and high-bandwidth off-chip memory is not available, so build-in on-chip memory comes in the first place. But, on-chip memory of typical embedded NNA is strongly limited. Taking into consideration the fact that on-chip memory stores not only NN model parameter data but also executable code, the available on-chip memory size for the NN model varies from 1 to 7 MB for present-day NNA as seen in Table I. Although it is possible to store a part of the NN model parameter data in off-chip memory, the bandwidth of such memory in edge devices bottlenecks their performance in this case. Summarizing, deployed NN models have to be compact not only in terms of computation amount but also in terms of memory footprint.

TABLE I. Neural network accelerators specifications

NNA model	TOPS	Type	Power consumption, mW	On-chip memory, MB
Kendryte K210 [10]	0.46	System-on-Chip (2 core CPU, NNA, I/O interfaces)	300	8
Lightspeeur 5801 [16]	2.8	NNA (host processor is required)	224	-
Google Edge TPU [17]	4	NNA (host processor is required)	2000	8
Bitmain Sophon BM1880 [12]	1	System-on-Chip (2+1 core CPU, NNA, I/O interfaces)	2500	2

C. Number format

The neural network quantization is a widely used technique to reduce the already trained NN model memory footprint [19]. The quantization is a conversion of the NN model parameters from 32- or 64-bit width floating-point number format to a fixed-point format of eight-bit width or less. Although this leads to minor NN accuracy decreasing, the use of this method is reasonable in mobile devices due to significant memory footprint reduction. Moreover, the fixed-point multiplies and addition are 6-38 times more efficient in energy and area [20]. So, to minimize chip area and power consumption inference-proposed NNAs operate with fixed-point numbers, typically eight-bit integers [10], [16], [17], [12].

III. Conclusion

This work-in-progress paper considered the opportunities of edge computing for NN-based vibration diagnostics. We overview the current state in the field of NN, vibration diagnostics, and hardware computing systems. We identified and analyzed the basic research problems for selecting NN-model architecture, proper data preparation, and hardware-specified optimization.

Acknowledgment

This research is financially supported by the Ministry of Science and Higher Education of Russia within project no. 075-11-2019-088 in part of pilot implementation of IIoT services for smart manufacturing. This research is supported by RFBR (research project # 19-07-01027) in part of methods for event-driven interaction for edge-centric computing. The work is implemented within the Government Program of Flagship

University Development for Petrozavodsk State University (PetrSU) in 2017-2021.

REFERENCES

[1] R. Burdzik, Ł. Konieczny, and T. Figlus, "Concept of on-board comfort vibration monitoring system for vehicles," in *International Conference on Transport Systems Telematics*. Springer, 2013, pp. 418–425.

[2] D. Korzun, E. Balandina, A. Kashevnik, S. Balandin, and F. Viola, *Ambient Intelligence Services in IoT Environments: Emerging Research and Opportunities*. IGI Global, 2019.

[3] D. Goyal and B. Pabla, "The vibration monitoring methods and signal processing techniques for structural health monitoring: a review," *Archives of Computational Methods in Engineering*, vol. 23, no. 4, pp. 585–594, 2016.

[4] A. Sánchez-Fernández, F. Baldán, G. Sainz-Palmero, J. Benítez, and M. Fuente, "Fault detection based on time series modeling and multivariate statistical process control," *Chemometrics and Intelligent Laboratory Systems*, vol. 182, pp. 57–69, 2018.

[5] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep learning for iot big data and streaming analytics: A survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2923–2960, 2018.

[6] M. V. Nural, M. E. Cotterell, and J. A. Miller, "Using semantics in predictive big data analytics," in *2015 IEEE International Congress on Big Data*, 2015, pp. 254–261.

[7] K. Zhong, M. Han, and B. Han, "Data-driven based fault prognosis for industrial systems: a concise overview," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 2, pp. 330–345, 2020.

[8] L. Jing, T. Wang, M. Zhao, and P. Wang, "An adaptive multi-sensor data fusion method based on deep convolutional neural networks for fault diagnosis of planetary gearbox," *Sensors*, vol. 17, no. 2, p. 414, 2017.

[9] Z. Zhu, G. Peng, Y. Chen, and H. Gao, "A convolutional neural network based on a capsule network with strong generalization for bearing fault diagnosis," *Neurocomputing*, vol. 323, pp. 62–75, 2019.

[10] "K210 datasheet," Canaan Inc., Beijing, Datasheet, 2017.

[11] N. P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers *et al.*, "In-datacenter performance analysis of a tensor processing unit," in *Proceedings of the 44th Annual International Symposium on Computer Architecture*, 2017, pp. 1–12.

[12] "Bm1680 datasheet," Beijing, Datasheet, 2017.

[13] (2016) CS231n convolutional neural networks for visual recognition. [Online]. Available: https://cs231n.github.io/convolutional-networks/

[14] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[15] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[16] (2019) Lightspeeur 5801 — gyrfalcon technology inc. Gyrfalcon Technology Inc. [Online]. Available: https://www.gyrfalcontech.ai/solutions/lightspeeur-5801/

[17] (2020) Technology — coral. Google LLC. [Online]. Available: https://coral.ai/technology/

[18] K. Siu, D. M. Stuart, M. Mahmoud, and A. Moshovos, "Memory requirements for convolutional neural network hardware accelerators," in *2018 IEEE International Symposium on Workload Characterization (IISWC)*. IEEE, 2018, pp. 111–121.

[19] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng, "Quantized convolutional neural networks for mobile devices," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4820–4828.

[20] W. Dally, "High-performance hardware for machine learning," in *Conference on Neural Information Processing Systems*, 2015.