Compacting Deep Neural Networks for Light Weight IoT & SCADA Based Applications with Node Pruning

Akm Ashiquzzaman*, Linh Van Ma*, SangWoo Kim*,Dongsu Lee*, Tai-Won Um†, Jinsul Kim*
* School of Electronics and Computer Engineering, Chonnam National Uniersity, Gwangju, South Korea
† Department of Information and Communication Engineering, Chosun University, Gwangju, South Korea
(zamanashiq3, linh.mavan, swmax88, ready1819)@gmail.com,twum@chosun.ac.kr, jsworld@jnu.ac.kr

***Abstract*—Deeplearning based image classifier is getting improved day by day. The network architecture is also increasing with the accuracy. But the bigger size and resource intensive training makes this model impractical to deploy in IoT based computational units. IoT has limited resources and reckoning power. So smaller network with same accuracy is highly priced for IoT based application deployment. In this study, convolutional deeplearning neural network and how pruning filters without compromising accuracy was studied. Efficient result was achieved from the pruned deeplearning neural network. the model was configured in the experiments by pruning the filter based on absolution position of zeros value based filter ranking. SCADA applications with intelligent component to detect data abnormality and remote sensing also required neural network applications. Using compact memory efficient module in such machines will also give proper validation in such applications in real time. In the end, proposed method for the pruned network delivered same accuracy with reduced size and thus archiving memory and computation for small sized application.**

Keywords - ***Deep learning, Convolutional Neural Network, Handwritten Digit recognition, Neural Network Pruning, Optimization***

I. Introduction

Nowadays, the widespread usage of Internet of Thing (IoT) enabled Devices gives us an excellent opportunity to use cutting edge optical character recognition (OCR) in almost every scenario. The sector that has the opportunity to deploy IoT based devices for any visual recognition of classification is endless. IoT devices gives the ability to expand any computation and systems to work over edge and open remote environments. IoT Devices had infinite opportunity to explore the optical characters recognition system, IoT devices integrated with such systems are now an important part of many sectors. According to Mohammadi et al. [1], the annual estimated market growth or economical impact of IoT Devices will be over $2.7 trillion dollars. This rise of factors will eventually render into many sectors of industries to adapt into such devices. Optical Character Recognition is the automation systems to recognize various text characters in several visual context. The main Idea of recognizing text mainly falls into the pattern recognition domain. Development of various deep learning algorithms, specially convolutional neural networks now makes the recognition accuracy very high for extremely large and widespread datasets & labels. Increased accuracy in handwritten character detection will open up new frontiers in optical character recognition applications. However, the scarcity of labeled data for numerals of various languages is a hindrance for exploration in these models. Recently many researchers have introduces benchmark datasets for various languages. With this resources, deploying high accuracy neural networks are now more easier than ever.

Hubel and Wiesel in the 1960s showed that cat and monkey visual cortex contains neurons that individually respond to small regions of the visual field [2]. The main idea of Convolutional Neural Network evolved from the idea to perform convolution in the input matrix to exploit edge features for the network to learn and recognize. Images, signal waves, sound an digital signals are represented in a multi-dimensional arrays. Any affine transformation cannot extract useful information from it, but whereas discrete convolution is a sparse operation and reused the parameter by sharing. Lecunn et al. [3] first uses convolutional neural networks successfully with back-propagation algorithm to optimize and gradient update [4]. Krizhevsky et al. [5] proposed convolutional neural network model won the 2012 Imagenet model and opened the door to use CNN in various image classification applications. CNN is opening a variant of deep feed forward artificial neural network which is now the key technique in analyzing visual images. This technique has been used in different character recognition models, for a range of languages with a remarkable accuracy.

Now the modern CNN's are getting bigger with new versions. Although the accuracy is improving to a remarkable rate, the bigger network is making the neural network an expensive computational resources. The VGGnet proposed and trained by the Oxford's Visual geometry group has over $138,357,544$ trainable parameters [6]. This model makes the image classification in large scale very efficient with error late less than 7%. However, training this model cantoning large trainable parameters takes very big computation. So, modern remote machines, such as IoT needs more lightweight resources to perform this classification.

Network pruning or removing the nodes without decreasing the accuracy to make the neural network light is not a not idea. Lecun et al. [7] proposed the model to removing the nodes after training in 1990. Even though this application

was proposed far ago, proper research on this method to optimized neural networks are not explored. Now, the rise of IoT based devices and services with edge computing makes this process more important than ever. Optimized node pruning and compressing neural networks will give high speed neural network fitted for digit classifying in IoT based devices.

Supervisory Control and Data Acquisition (SCADA) can be defined as a collections of system softwares and hardware elements that basically gives industrial process to control any sequences of mechanical process to supervise and control both locally and remotely by monitor, gather and process real time data [8]. As this directly interacts with devices such as sensors, valves, pumps, motors, and more through human-machine interface (HMI) software, the delay is common. Applying machine learning algorithms in these software modules to automate the sensing and data acquisition is an ever emerging research field. Small sized intelligent deeplearning models which can be embedded in SCADA will definitely help making this system more robrust and less laggy in decision making. Partially implemented deeplearning model will make the whole system less prone to human errors during manufacturing in industrial processes.

In this paper, the idea of neural network pruning for optimized deep-learning neural network (DNN) to classify hand written digit was explored. the proper process for convolutional layer filter ordering was studied. the proposed method for compacting DNN showed promising compression rate and steady accuracy. The structure of rest of the paper as follows, the proper research methods and the techniques for the application are reviewed in Section II. Section III contains the our proposed model explanation. Datasets used for this study are being explained in Section IV. The proper results for the experiments were described in Section V.

II. Background Studies

The main idea of Deep learning neural networks are not new. But the main application for deep learning neural networks are a common variation of DNN called convolutional neural networks (CNN). CNN used the idea of mathematical convolutional operation to extract the important features from the datasets. DNN got the name for it's numerous hidden layers in it's architecture. In the Fig. 1, a very simplified neural network is being illustrated for simplification. the hidden layers here are being defined by the node and the nodes are trained with the dataset to classify based on weight correction. This algorithm is known as the backpropagation algorithm [4].

Oh the other hand, CNN are the variation of DNN that has combined convolutional layers for feature extraction and the extracted features then later send into the fully connected layers for classification. The whole architecture of CNN is drawn in the Fig. 2. The main CNN layer which contained the shared convolutional filters or kernels. Fig. 2 illustrated the two main categories, convolutional layers and fully-connected dense layers. The convolutional layers acts as the main feature extractors and the fully-connected layers use this to map the extracted featured to it's given label or class.

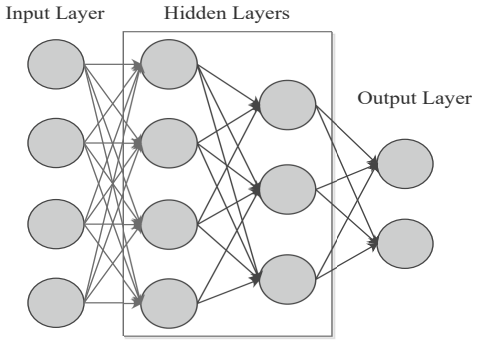

Fig. 1. Simplified DNN Architecture

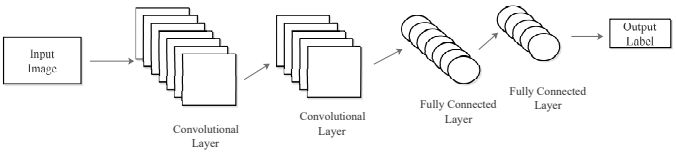

Fig. 2. Common CNN Architecture

Hu et al. [9] proposed to explore sparsity in activations for network pruning. Rectified Linear Unit (ReLU) [10] activation function imposes sparsity during inference, and average percentage of positive activation at the output can determine importance of the neuron. Average Percentage of Zeros (APoZ) is useful to to estimate saliency of feature maps in given layer. Based on the idea from here, ${O_c}^{(i)}$ denotes the output of c-th channel in i-th layer, ${APoZ_c}^i$ of the c-th neuron in i-th layer is defined as:

$$APoZ({O_c}^{(i)}) = \frac{\sum_k^N \sum_j^M f({O_{c,j}}^{(i)}(k) = 0)}{N \times M} \quad (1)$$

where $f(\cdot) = 1$ if true, and $f(\cdot) = 0$ if false, M denotes the dimension of output feature map of ${O_c}^{(i)}$, and N denotes the total number of validation examples. This Idea can be used to make the order of ranking of a filters in each layers. The main idea of filter or node pruning can be visualized in the Fig. 3.

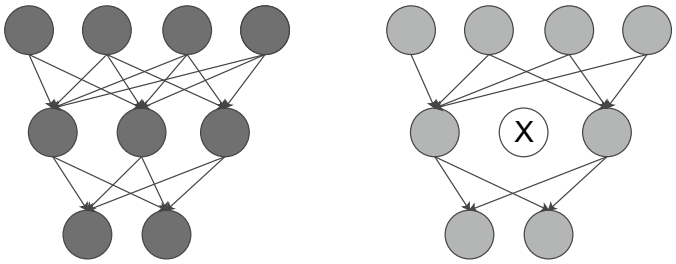

Fig. 3. Node Pruning in Neural Networks.

III. Proposed Methods

The main proposed method for the study was divided into several steps. In the initial stage the neural network architecture was defined. The CNN architecture was decided based on the several studies [5], [11], [12]. As the dataset

for the proposed study has limited target labels, the initial size of the neural network was exceptionally big. The main network has 4 convolutional layer. The initial filter was decides as follows $[512, 512, 256, 256]$. All of the layers had a single max pooling layer in between the convolutional layers to reduce the computation. The studies were mainly focused on the convolution layers in particular. The fully connected layers are often works as only feature mapping function and technologically it takes less resources during training and also less space in memory representation.So, not any type of pruning were dine in the fc layers in this experiments.

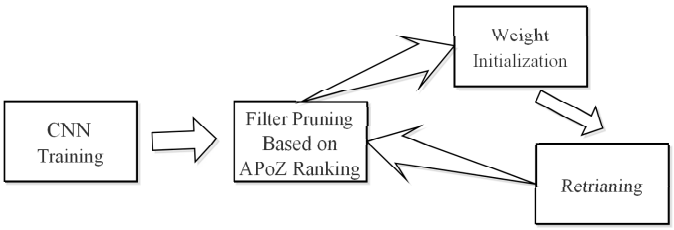

Fig. 4. Used methodology for filter pruning

The main idea of the neural network pruning is as follows. The main network is trained in all 3 datasets to achieve maximum accuracy. All training were halted based on accuracy metrics. Later the filters were pruned Based on the Average Percentage of Zeros (APoZ) ranking of each layer filters. The model then initialized again for retraining and the validation accuracy is checked until the network accuracy drops or reduced by the set standard.

IV. Datasets

Idea of Neural network pruning is data independent. This implies to the fact all of the datasets, regardless of the variation factor can be adapted to the DNN pruning methodology. For this study the DNN study was experimented in various handwritten digit datasets to verify it's utility. The CMATERDb digit datasets were chosen to test the DNN pruning [13]. The Arabic handwritten digits, adapted to used for Urdu numerals, was the first dataset used for training and pruning, Eventually Hindi or Devanagri digits and Bengali numeral digits were also used to test the proposed model.

The Bengali Digit datasets consists of $6,000$ independent samples images. The Hindi character and Urdu Numerical character datasets both are $3,000$ RGB images with 32×32 pixel height and width. All of the images were converted into gray-scale for computational ease. All of the numerical classes has 10 distinct values ranging from 0 to 9 . Dataset Information in represented in brief in Table I.

V. Experiments

All the proposed models were constructed in the Python based open-source library Keras [14] and Pytorch, a python based torch library for deeplearning modeling. All the models described in proposed sections were trained in the computer with Intel core $i7$ 8 core CPU with 8 Gb DDR4 RAM. The

English Digit	Arabic Digit	Bengali Digit	Hindi Digit
1	١	১	१
2	٢	২	२
3	٣	৩	३
4	٤	৪	४
5	٥	৫	५
6	٦	৬	६
7	٧	৭	७
8	٨	৮	८
9	٩	৯	९
0	٠	০	०

Fig. 5. Handwritten Numerical Digit Images for Training and Validation

TABLE I
A DETAILED VIEW OF THE DATASET USED FOR THE EXPERIMENTS IN THIS STUDY

Total number of Instances	6000	3000	3000
Total Number of Classes	10	10	10
Image Dimentions	32×32	32×32	32×32
Image Information	RGB Image in Bitmap format	RGB Image in Bitmap format	RGB Image in JPEG format

Nvidia *CUDA* library was used to speed up the training process with the help of single instance to Nvidia Geforce 1050-ti GPU. The above mentioned GPU has 4 GB of VRAM. All the learned weights are being saved in the Hadoop file systems (HDFS) for safe keeping.

The results were very promising for the all datasets. Native model performed well over all all the dataset. The accuracy reached over 95% for all the dataset. Then filter pruning operation were conducted in a circular manner unless the accuracy drops from the primitive achieved ones.

TABLE II
COMPREESION OF THE NEURAL NETWORK BY FILTER PRUNING

Dataset	Achieved Accuracy	After Pruning Parameters	Compression
Bengali	97.06%	7.3 Mil	1.67%
Hindi	98.76%	5.78 Mil	2.40%
Arabic	96.38%	8.89 Mil	1.46%

The result in Table II describes the achievements in a nutshell. The first all network were separately trained till the accuracy improving stops. This process was monitored with

automation based on accuracy improvement monitoring. All of the models then was pruned based on the proposed method described in the Section III. The accuracy of all the networks were remained same but the filter removal was different based on the different datasets. In this study, the Hindi dataset model was best compressed. the model almost reduced half of it's original size. This network filter pruning depends up to the datasets themselves as the image feature representative varies to dataset. Overall, all the digit dataset DNN can be pruned and eventually deployed into IoT based application due to it's reduced compact memory size.

VI. Conclusion

This paper proposes a novel solution for IoT based deep learning image classification optimization process. The IoT based application solution sometimes needs to be both memory and computation resource efficient. Modern SCADA systems are also now dependent on accurate data sensing with limited resources in real time. Applying this model into the data sensing part of SCADA will help achieve both response time and data management for any SCADA system. Deep learning models are often gets very big in model size with high and precise accurate classification. This creates difficulty for implementing the application specially developed for IoT based solution. network pruning is not a new technology. The idea of optimal node removing was conceptualized far ago. Although the use of the method is not studied in details. In this paper, we experimented with various handwritten digit datasets with network pruning. The main objective was to reduce the main network filters to overcome massive computation and memory size during final deployment in IoT based devices. In the end, the result was promising as the network size was reduced to almost one third of the original parameters.

Acknowldgement

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2018-2016-0-00314) supervised by the IITP(Institute for Information & communications Technology Promotion). Besides, this research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science, and Technology (MEST)(Grant No. NRF-2017R1D1A1B03034429) and was also supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT). [2018-0-00691, Development of Autonomous Collaborative Swarm Intelligence Technologies for Disposable IoT Devices]. Finally, This work (Grants No. S2655639) was supported by project for Cooperative R& D between Industry, Academy, and Research Institute funded Korea Ministry of SMEs and Startups in 2018.

References

[1] 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*, 2018.

[2] R. Jung, H. Kornhuber, and J. S. Da Fonseca, "Multisensory convergence on cortical neurons neuronal effects of visual, acoustic and vestibular stimuli in the superior convolutions of the cat's cortex," *Progress in brain research*, vol. 1, pp. 207–240, 1963.

[3] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[4] Y. Hirose, K. Yamashita, and S. Hijiya, "Back-propagation algorithm which varies the number of hidden units," *Neural Networks*, vol. 4, no. 1, pp. 61–66, 1991.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[6] A. Rosebrock, "Imagenet: Vggnet, resnet, inception, and xception with keras," *Mars*, 2017.

[7] Y. LeCun, J. S. Denker, and S. A. Solla, "Optimal brain damage," in *Advances in neural information processing systems*, 1990, pp. 598–605.

[8] S. A. Boyer, *SCADA supervisory control and data acquisition*. The Instrumentation, Systems and Automation Society, 2018.

[9] H. Hu, R. Peng, Y. Tai, C. Tang, and N. Trimming, "A data-driven neuron pruning approach towards efficient deep architectures. arxiv preprint," *arXiv preprint arXiv:1607.03250*, 2016.

[10] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv preprint arXiv:1511.07289*, 2015.

[11] A. Ashiquzzaman and A. K. Tushar, "Handwritten arabic numeral recognition using deep learning neural networks," in *Imaging, Vision & Pattern Recognition (icIVPR), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1–4.

[12] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[13] "Google coe archieve - long-term storage for google code project hosting." https://code.google.com/archive/p/cmaterdb/downloads, accessed: 2016-12-30.

[14] F. Chollet, "keras," https://github.com/fchollet/keras, 2015.