ACOUSTIC EVENT DETECTION WITH BINARIZED NEURAL NETWORK

WONG KAH LIANG

A project report submitted in partial fulfilment of the
requirements for the award of the degree of
Master of Engineering (Computer & Microelectronic Systems)

School of Electrical Engineering
Faculty of Engineering
Universiti Teknologi Malaysia

JULY 2020

DEDICATION

This thesis is dedicated to my father, who taught me that the best kind of knowledge to have is that which is learned for its own sake. It is also dedicated to my mother, who taught me that even the largest task can be accomplished if it is done one step at a time.

ACKNOWLEDGEMENT

In preparing this thesis, I was in contact with many people, researchers, academicians, and practitioners. They have contributed towards my understanding and thoughts. In particular, I wish to express my sincere appreciation to my main thesis supervisor, Associate Professor Muhammad Mun'im bin Ahmad Zabidi, for encouragement, guidance, critics and friendship. Without his continued support and interest, this thesis would not have been the same as presented here. I would like to thank my panel of examiner on FYP 1, Dr Usman Ullah Sheikh and Dr Musa Mokji, panel of examiner on FYP 2, Dr Mohd Afzan Othman and PM Ir Dr Muhammad Nadzir Marsono who have gave me feedbacks and comment of improvement in my research during the presentation session.

My fellow postgraduate student should also be recognised for their support. My sincere appreciation also extends to all my colleagues and others who have provided assistance at various occasions. Their views and tips are useful indeed. Unfortunately, it is not possible to list all of them in this limited space.

Last but not least, I would like to thank all my family member who have gave me encouragement, moral and mental support throughout this project.

ABSTRACT

Implementation of deep learning for Acoustic Event Detection (AED) on embedded systems is challenging due to constraints on memory, computational resources and, power dissipation. Various solutions to overcome this limitation have been proposed. One of the latest methods to overcome this limitation is by using Binarized Neural Network (BNN) which has been proven to achieve approximately 32x memory savings and 58x lower computational resources. XNOR-Net is a type of BNN which uses the XNOR gate to perform a logical function on the input data and give all outputs in binary form. In this project, the XNOR-Net model is constructed and trained for the AED task using urban sound (UrbanSound8K) and bird sound (Xeno-Canto) datasets. Prior to performing the training, the datasets were pre-processed through audio segmentation to produce 1-second sound files. Each audio file is converted from the time domain to Mel-Spectrogram in the frequency domain and thresholding was implemented to convert each spectrogram into a binary image. The images are then reshaped to 32×32 pixels before being used for the training procedure. A performance comparison between BinaryNet and XNOR-Net in terms of the number of hidden layers used was performed and one binary convolutional layer structure XNOR-Net was determined and constructed. The block structure and hyperparameters of the XNOR-Net were analyzed and optimized to achieve a training accuracy of 96.06% and validation accuracy of 94.08%.

ABSTRAK

Pelaksanaan pembelajaran mendalam untuk Pengesanan Acara Akustik (AED) pada sistem tertanam sangat mencabar kerana kekangan pada memori, sumber komputasi dan pelesapan daya. Pelbagai penyelesaian untuk mengatasi batasan ini telah dicadangkan. Salah satu kaedah terkini untuk mengatasi batasan ini adalah dengan menggunakan Binarized Neural Network (BNN) yang telah terbukti mencapai kira-kira 32x penjimatan memori dan 58x sumber pengiraan yang lebih rendah. XNOR-Net adalah jenis BNN yang menggunakan gerbang XNOR untuk melakukan fungsi logik pada data input dan memberikan semua output dalam bentuk binari. Dalam projek ini, model XNOR-Net dibina dan dilatih untuk tugas AED menggunakan set data bunyi bandar (UrbanSound8K) dan suara burung (Xeno-Canto). Sebelum melakukan latihan, set data telah diproses sebelumnya melalui segmentasi audio untuk menghasilkan fail suara 1 saat. Setiap fail audio ditukarkan dari domain waktu ke Mel-Spectrogram dalam domain frekuensi dan ambang dilaksanakan untuk mengubah setiap spektrogram menjadi gambar biner. Gambar kemudian dibentuk semula menjadi 32×32 piksel sebelum digunakan untuk prosedur latihan. Perbandingan prestasi antara BinaryNet dan XNOR-Net dari segi jumlah lapisan tersembunyi yang digunakan telah dilakukan dan satu struktur lapisan konvolusional binari XNOR-Net telah ditentukan dan dibina. Struktur blok dan hiperparameter XNOR-Net dianalisis dan dioptimumkan untuk mencapai ketepatan latihan 96.06% dan ketepatan pengesahan 94.08%

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE

LIST OF ABBREVIATIONS

AED	-	Acoustic Event Detection
ASR	-	Autonomous Speech Recognition
AUC	-	Area Under the ROC curve
BatchNormal	-	Batch Normalizing
BAD	-	Bird Acoustic Detection
BCNN	-	Binarized Convolutional Neural Networks
BNN	-	Binarized Neural Network
BSD	-	Bird Sound Detection
CA-NN	-	Context-Adaptive Neural Network
CNN	-	Convolutional Neural Network
DNN	-	Deep Neural Network
GMM	-	Gaussian Mixture Model
GPU	-	Graphical Processing Unit
HMM	-	Hidden Markov Model
MAC	-	Multiply-Accumulate
MFCCs	-	Mel-Frequency Cepstral Coefficients
PCEN	-	Per-Channel Energy Normalization
RNN	-	Recurring Neural Network
ROC	-	Receiver Operating Characteristic
SIMD	-	Single Instruction, Multiple Data
SWAR	-	SIMD within a register

LIST OF SYMBOLS

σ	-	Sigma

CHAPTER 1

INTRODUCTION

1.1 Background of Research

The sound comes with a wide variety of frequency content and temporal structure in unstructured environments. The wide range of frequency variations gives different kinds of information especially in urban areas such as car horns, dog bark, birds chirp, and others [1]. For example, the center frequency of baby crying is at 2 kHz and glass shattering at 4 kHz as shown in Table 1.1.

Table 1.1 Types of sounds and their center frequencies [2].

Sound Number	**Sound Name**	**Center Frequency (Hz)**
1	Airplane passing	250
2	Baby crying	2000
3	Bird singing	2000
4	Cow mooing	500
5	Cuckoo clock sounding	1000
6	Dog barking	1000
7	Coyote howling	500
8	Glass shattering	4000
9	Baby rattle shaking	4000
10	Train chugging along	250
11	Thunder cracking	250
12	Drum beating	500

For home surveillance, Amazon has introduced a smart sensor that can help users to keep their home safe, known as the Alexa Guard [3]. The Alexa Guard can be activated by the sound of smoke alarms, carbon monoxide alarms, or glass breaking that happens when the user is out of the home. This system uses a Convolutional Neural Network (CNN) to recognize the type of sounds detected. If it is an alarm sound, the Alexa Guard will send the sound recording to notify the user remotely.

The Convolutional Neural Network (CNN) is one of the fundamental network architectures of Deep Neural Networks (DNNs). The CNN performs very well on object recognition and detection in real-world applications. In common with other classes of intelligent systems, CNN must be trained to obtain the model of the desired behaviour. Training CNN-based recognition systems require large amounts of computational power and memory resources. Today very fast and power-hungry Graphics Processing Units (GPUs) are used to train the neural network [4].

For the embedded system such as the Alexa Guard, the training can be done by high-performance computers. The embedded system only requires the model produced by the training process for run-time inference [5, 6]. The main issue with embedded systems is the limited resources available on the devices. The CNN architecture must run with sufficient performance at low power with the available memory and compute capabilities.

1.2 Problem Statement

- The Deep Neural Networks (DNNs) are becoming more powerful and hence the power and resource constraints have become the challenge as they require more storage and computational power.
- This causes the DNN is not capable to be implemented into low power devices such as smartphones, drones, mobile devices, and embedded systems that are able to provide low memory storage and low computational power.

1.3 Objective

- To implement Binarized Neural Networks (BNNs) in performing training and validation using binary input images.
- To explore the architecture of Binarized Neural Networks (BNNs) so that to optimize the training and accuracy of the model.
- To analyze and optimize the hyperparameter of the neural network improve the testing and validation accuracy.

1.4 Scope of work

- All work was performed on a personal laptop with the Intel Core i5 7th-generation processor and NVIDIA GeForce MX150 graphics card in the Windows environment.
- Process the input sound datasets to create a binarized spectrogram for BNN training samples.
- Preparation of positive datasets from Xeno-Canto website [7] and negative datasets from UrbanSound8K [8].
- BinaryNet was used to structure the neural network for bird sound presence/absence detection and recognition using the Python Keras framework.

REFERENCES

1. Cakır, E., Parascandolo, G., Heittola, T., Huttunen, H. and Virtanen, T. Convolutional recurrent neural networks for polyphonic sound event detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2017. 25(6): 1291–1303.

2. Letowskit, T. Comparison of air-conduction and bone-conduction hearing thresholds for pure tones and octave-band filtered sound effects. *J Am Acad Audiol*, 1999. 10: 422–428.

3. Chen, E. and Wang, C. (A06) Acoustic Event Detection with Alexa Guard. [Online]. Available at `https://www.youtube.com/watch?v=-nKelNVVblM`. Accessed on: December 16, 2019.

4. Courbariaux, M., Hubara, I., Soudry, D., El-Yaniv, R. and Bengio, Y. Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. *arXiv preprint arXiv:1602.02830*, 2016.

5. Liu, B., Wang, Z., Fan, H., Yang, J., Zhu, W., Huang, L., Gong, Y., Ge, W. and Shi, L. EERA-KWS: A 163 TOPS/W always-on keyword spotting accelerator in 28nm CMOS using binary weight network and precision self-adaptive approximate computing. *IEEE Access*, 2019. 7: 82453–82465.

6. Yin, S., Ouyang, P., Zheng, S., Song, D., Li, X., Liu, L. and Wei, S. A 141 uw, 2.46 pj/neuron binarized convolutional neural network based self-learning speech recognition processor in 28nm cmos. *2018 IEEE Symposium on VLSI Circuits*. IEEE. 2018. 139–140.

7. xeno-canto: Sharing bird sounds from around the world. Xeno-canto Foundation and Naturalis Biodiversity Center, [Online]. Available at `https://www.xenocanto.org/`. Accessed on: October 11, 2019.

8. Salamon, J., Bello, J. and Jacoby, C. Urban Sound Datasets. [Online]. Available at `https://urbansounddataset.weebly.com/`. Accessed on: October 13, 2019.

9. Furnas, B. J. and Callas, R. L. Using automated recorders and occupancy models to monitor common forest birds across a large geographic region. *The Journal of Wildlife Management*, 2015. 79(2): 325–337.

10. Stowell, D., Wood, M., Stylianou, Y. and Glotin, H. Bird detection in audio: a survey and a challenge. *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE. 2016. 1–6.

11. Lostanlen, V., Salamon, J., Farnsworth, A., Kelling, S. and Bello, J. P. Birdvox-full-night: A dataset and benchmark for avian flight call detection. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2018. 266–270.

12. Lostanlen, V., Salamon, J., Farnsworth, A., Kelling, S. and Bello, J. P. Robust sound event detection in bioacoustic sensor networks. *PloS one*, 2019. 14(10): e0214168.

13. de Oliveira, A. G., Ventura, T. M., Ganchev, T. D., de Figueiredo, J. M., Jahn, O., Marques, M. I. and Schuchmann, K.-L. Bird acoustic activity detection based on morphological filtering of the spectrogram. *Applied Acoustics*, 2015. 98: 34–42.

14. Takahashi, N., Gygli, M., Pfister, B. and Van Gool, L. Deep convolutional neural networks and data augmentation for acoustic event detection. *arXiv preprint arXiv:1604.07160*, 2016.

15. Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

16. Salamon, J., Bello, J. P., Farnsworth, A. and Kelling, S. Fusing shallow and deep learning for bioacoustic bird species classification. *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2017. 141–145.

17. Salamon, J., Bello, J. P., Farnsworth, A., Robbins, M., Keen, S., Klinck, H. and Kelling, S. Towards the automatic classification of avian flight calls for bioacoustic monitoring. *PloS one*, 2016. 11(11): e0166866.

18. F.Chollet. Keras v2.0.0. 2018. [Online]. Available at `https://github.com/fchollet/keras`.

19. B.McFee, C.Jacoby and E.Humphrey. pescador. 2017. [Online]. Available at https://doi.org/10.5281/zenodo.400700.

20. Ahirwar, K. Everything you need to know about Neural Networks. Mate Labs, November 1, 2017. [Online]. Available at https://hackernoon.com/everything-youneed-to-know-about-neural-networks-8988c3ee4491. Accessed on: December 3, 2019.

21. Understanding Binary Neural Networks. All Things Geeky, 2 October 2017. [Online]. Available at https://sushscience.wordpress.com/2017/10/01/understandingbinary-neural-networks/. Accessed on: December 2, 2019.

22. Courbariaux, M., Bengio, Y. and David, J.-P. Binaryconnect: Training deep neural networks with binary weights during propagations. *Advances in neural information processing systems*. 2015. 3123–3131.

23. Kim, S., Kim, S.-H., Hwang, Y. and Jeong, J. Comparison of training methods for the binarized neural object detection network. *2019 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*. IEEE. 2019. 1–3.

24. Rastegari, M., Ordonez, V., Redmon, J. and Farhadi, A. Xnor-net: Imagenet classification using binary convolutional neural networks. *European conference on computer vision*. Springer. 2016. 525–542.

25. Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

26. Song, J. and Li, S. Bird Sound Detection Based on Binarized Convolutional Neural Networks. *Proceedings of the 6th Conference on Sound and Music Technology (CSMT)*. Springer. 2019. 63–71.

27. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 2014. 15(1): 1929–1958.

28. Keras: The Python Deep Learning library. [Online]. Available at https://keras.io/.

29. TensorFlow. [Online]. Available at https://www.tensorflow.org/.

30. Pytorch. [Online]. Available at https://pytorch.org/.

31. Araya-Salas, M. and Smith-Vidaurre, G. warbleR: an R package to streamline analysis of animal acoustic signals. *Methods in Ecology and Evolution*, 2017. 8(2): 184–191.

32. D.Gartzman. Getting to Know the Mel Spectrogram. Medium, Aug 12 2019. [Online]. Available at https://towardsdatascience.com/getting-to-know-the-melspectrogram-31bca3e2d9d0.

33. Marini, A., Turatti, A. J., Britto, A. and Koerich, A. L. Visual and acoustic identification of bird species. *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2015. 2309–2313.

34. Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., Chen, Z., Zhang, Y., Wang, Y., Skerrv-Ryan, R. *et al.* Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2018. 4779–4783.

35. Image Thresholding. [Online]. Available at https://docs.opencv.org/master/d7/d4d/tutorial_py_thresholding.html.

36. BatchNormalization layer. [Online]. Available at https://keras.io/api/layers/normalization_layers/batch_normalization/.

37. XNOR Net PyTorch. [Online]. Available at https://github.com/jiecaoyu/XNOR-Net-PyTorch.

38. BinaryNet and XNORNet. [Online]. Available at https://github.com/yaysummeriscoming/BinaryNet_and_XNORNet.

39. 2D Convolution layer. [Online]. Available at https://keras.io/api/layers/convolution_layers/convolution2d/.

40. Sign Activation. [Online]. Available at http://tensorflow.biotecan.com/python/Python_1.8/tensorflow.google.cn/api_docs/python/tf/sign.html.

41. Max pooling operation for 2D spatial data. [Online]. Available at https://keras.io/api/layers/pooling_layers/max_pooling2d/.

42. Flatten layer. [Online]. Available at https://keras.io/api/layers/reshaping_layers/flatten/.

43. Using the Keras Flatten Operation in CNN Models with Code Examples. [Online]. Available at https://missinglink.ai/guides/keras/using-keras-flatten-operation-cnn-models-code-examples/.

44. Dropout layer. [Online]. Available at https://keras.io/api/layers/regularization_layers/dropout/.

45. Dense layer. [Online]. Available at https://keras.io/api/layers/core_layers/dense/.

46. Park, D. S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E. D. and Le, Q. V. SpecAugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*, 2019.