GATED RECURRENT UNIT FOR LOW POWER WAKE-WORD DETECTION

CHIN JIAN QEE

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

FEBRUARY 2021

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, Professor Muhammad Mun'im bin Ahmad Zabidi, for encouragement, guidance, critics, friendship, advices and motivation. Without his continued support and interest, this thesis would not have been the same as presented here.

I am also indebted to Universiti Teknologi Malaysia (UTM) for funding my Master study. Librarians at UTM, Cardiff University of Wales and the National University of Singapore also deserve special thanks for their assistance in supplying the relevant literatures.

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. I am grateful to all my family member.

ABSTRACT

Neural networks made some of the latest state of the art technologies such as speech recognition, language translation and stock prediction possible. Among them, speech recognition is a very popular application which is growing rapidly. It is widely used in applications such as mobile phones and Amazon smart speakers in order to enhance user experience. However, neural networks used for speech recognition require a large amount of computations, especially if it is in always-on state. This made it infeasible to be implemented in battery-powered edge devices such as wearables, sensors, and internet-of-things devices, as the battery life will not last long enough to provide a good user experience. To address this issue, this work enhances the recurrent neural network (RNN), or specifically, Gated Recurrent Unit (GRU) for the task of wake-word detection. A wake-word detector is always powered-on, listening to a specific phrase, the wake-word. Therefore, the power consumption must be low enough to enable long battery usage – a feature that is sought by many end-consumers. This work proposes four modifications to the existing GRU architecture. First, the reset gate is removed as there are researches which implies that it is not needed in application such as speech recognition. Second, the activation function is changed from the conventional sigmoid/hyperbolic tangent function to softsign function. Third, weight quantization is carried out to reduce the memory footprint and speed up calculations. Fourth, fixed point arithmetic is used instead of floating point format. With the above enhancements in architecture, memory and power consumption is reduced while keeping the impact to the accuracy minimal. Furthermore, it is possible to embed this new neural network model to battery-powered edge devices such as wearables. In summary, this work explores the possibility of implementing an improved GRU architecture in battery-powered edge devices to enable low-power usage for speech recognition purpose.

ABSTRAK

Rangkaian neural memungkinkan beberapa teknologi terkini seperti pengecaman pertuturan, penterjemahan bahasa dan ramalan saham. Antara teknologi ini, pengecaman pertuturan adalah aplikasi yang sangat popular dan berkembang pesat. Ia digunakan secara meluas dalam aplikasi seperti telefon bimbit dan pembesar suara pintar Amazon untuk meningkatkan pengalaman pengguna. Walau bagaimanapun, rangkaian neural yang digunakan untuk pengecaman pertuturan memerlukan pengiraan yang banyak, terutamanya jika sentiasa berada dalam keadaan aktif. Ini menjadikan rangkain neural tidak dapat dilaksanakan menggunakan tenaga bateri seperti peranti yang dipakai pada badan, sensor, dan peranti internet, kerana jangka hayat bateri tidak dapat bertahan untuk memberikan pengalaman pengguna yang baik. Untuk mengatasi masalah ini, tesis ini meningkatkan rangkaian neural RNN, atau secara khusus, Gated Recurrent Unit (GRU) untuk tugas pengesanan kata bangun. Pengesan kata bangun sentiasa hidup untuk mendengar frasa tertentu yakni kata bangun. Oleh itu, penggunaan kuasa mestilah cukup rendah untuk membolehkan penggunaan bateri yang lama - ciri yang dimahukan ramai pengguna. Tesis ini mencadangkan empat modifikasi kepada seni bina GRU yang ada. Pertama, pintu reset dikeluarkan kerana terdapat penyelidikan yang menunjukkan bahawa ia tidak diperlukan dalam aplikasi seperti pengecaman pertuturan. Kedua, fungsi pengaktifan diubah dari fungsi sigmoid/tangen hiperbolik konvensional kepada fungsi softsign. Ketiga, pengkuantuman pemberat dilakukan untuk mengurangkan jejak memori dan mempercepat pengiraan. Keempat, aritmetik titik tetap digunakan dan bukannya format titik terapung. Dengan naik taraf seni bina tersebut, ingatan dan penggunaan kuasa dapat dikurangkan sambil mengurangkan impak terhadap ketepatan. Selanjutnya, model rangkaian neural baru ini mampu diterap ke peralatan bertenaga bateri seperti peranti yang dipakai di badan. Ringkasnya, tesis ini meneroka kemungkinan penerapan senibina GRU yang lebih baik dalam peranti berkuasa bateri untuk menghasilkan pengecaman pertuturan menggunakan kuasa rendah.

TABLE OF CONTENTS

	TITLE			PAGE
	DECLARATION			**iii**
	DEDICATION			**iv**
	ACKNOWLEDGEMENT			**v**
	ABSTRACT			**vi**
	ABSTRAK			**vii**
	TABLE OF CONTENTS			**viii**
	LIST OF TABLES			**xi**
	LIST OF FIGURES			**xii**
	LIST OF ABBREVIATIONS			**xiii**
	LIST OF SYMBOLS			**xiv**
CHAPTER 1	**INTRODUCTION**			**1**
	1.1	Problem Background		1
	1.2	Problem Statement		3
	1.3	Research Questions		3
	1.4	Research Goal		3
	1.5	Objective		4
	1.6	Scope and Limitation		4
	1.7	Structure of the Thesis		5
CHAPTER 2	**LITERATURE REVIEW**			**7**
	2.1	Overview		7
	2.2	Machine Learning		7
		2.2.1	RNN	8
		2.2.2	LSTM	11
		2.2.3	GRU	16
	2.3	Neural Network Optimizations		17
		2.3.1	Single Gate Mechanism	18
		2.3.2	Changes in Activation Function	19
		2.3.3	Weight Quantization	20

2.3.4 Fixed-Point Arithmetic 21

2.4 Summary of Literature Review 22

CHAPTER 3 RESEARCH METHODOLOGY 25

3.1 Flow of Methodology 25

3.1.1 Single Gate Mechanism 26

3.1.2 Changes in Activation Functions 27

3.1.3 Weight Quantization 29

3.1.4 Fixed Point Arithmetic 30

3.1.5 Training Setup 30

3.1.6 Training Data Preparation 32

3.1.7 Implementation of Enhanced GRU on lower power MCU 34

3.2 Analysis of the Performance of the Enhanced GRU 35

3.2.1 Experiment 1 35

3.2.2 Experiment 2 36

3.2.3 Experiment 3 36

3.2.4 Experiment 4 37

3.3 Summary of Methodology 38

CHAPTER 4 RESULT AND DISCUSSION 39

4.1 Experimental Setup 39

4.1.1 Experiment 1 42

4.1.1.1 Full Precision Original GRU Cell 42

4.1.1.2 Single Gated Recurrent Cell 45

4.1.1.3 Softsign Activation Function 47

4.1.1.4 Quantized Weights GRU 48

4.1.2 Experiment 2 51

4.1.3 Experiment 3 52

4.1.4 Experiment 4 54

4.2 Summary of Performance Evaluation 55

CHAPTER 5 CONCLUSION **57**

5.1 Research Outcomes 57

5.2 Contributions to Knowledge 58

5.3 Future Works 58

REFERENCES **59**

LIST OF TABLES

TABLE NO.	TITLE	PAGE
Table 1.1	Price-Power Comparison for Existing Technologies	3
Table 2.1	Summary of Enhancements	22
Table 3.1	GRU Architecture	26
Table 3.2	Enhanced GRU Architecture (Removal of Reset Gate)	27
Table 3.3	Activation Function Equations.	28
Table 3.4	Enhanced GRU Architecture (Changes in Activation Function)	28
Table 3.5	Quantization Example	29
Table 3.6	Summary of Performance Evaluation	35
Table 4.1	Hyperparameter List	40
Table 4.2	Original GRU Performance	43
Table 4.3	Original GRU Performance (early stopping)	43
Table 4.4	Single Gated Recurrent Cell	45
Table 4.5	Single Gated Recurrent Cell (early stopping)	45
Table 4.6	Softsign Activation Function	47
Table 4.7	Softsign Activation Function (early stopping)	47
Table 4.8	TFLite model (with float16 quantization)	49
Table 4.9	TFLite model (with uint8 quantization)	50
Table 4.10	Enhanced GRU	51
Table 4.11	Enhanced GRU (early stopping)	51
Table 4.12	Original Model	53
Table 4.13	Enhanced model (on Desktop)	53
Table 4.14	Original GRU model (on Raspberry Pi)	54
Table 4.15	Enhanced GRU model (on Raspberry Pi)	54
Table 4.16	Enhanced GRU model (on Raspberry Pi)	55

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE
Figure 1.1	The Future of Edge-based Computing	2
Figure 2.1	Machine Learning	8
Figure 2.2	Feedforward Neural Network.	9
Figure 2.3	Recurrent Neural Network.	9
Figure 2.4	Sequence of Words.	10
Figure 2.5	Sequence of RNN with Input Vectors	12
Figure 2.6	Relationship of RNN Inputs and Output	13
Figure 2.7	LSTM Cell Architecture	13
Figure 2.8	Operation of LSTM	15
Figure 2.9	GRU Cell Architecture	16
Figure 3.1	GRU Cell Architecture	26
Figure 3.2	Softsign Activation Function	28
Figure 3.3	Training Setup Flow	32
Figure 3.4	Training Data Preparation	33
Figure 3.5	Amplitude vs Time Domain	33
Figure 3.6	Magnitude vs Frequency Domain (After FT)	34
Figure 3.7	Spectrogram	34
Figure 4.1	GRU model configuration	41
Figure 4.2	GRU model configuration visualization	41
Figure 4.3	Original GRU tensorboard	43
Figure 4.4	Single Gated Recurrent Cell tensorboard	45
Figure 4.5	Softsign Activation Function tensorboard	47
Figure 4.6	Enhanced GRU tensorboard	52

LIST OF ABBREVIATIONS

ANN	-	Artificial Neural Network
DNN	-	Deep Neural Network
DT	-	Decision Tree
RNN	-	Recurrent Neural Network
LSTM	-	Long Short Term Memory
GRU	-	Gated Recurrent Unit
MCU	-	Microcontroller Unit
KWS	-	Key-Word Spotting
SVM	-	Support Vector Machine
HVM	-	Hidden Markov Model
ReLU	-	Rectified Linear Unit
DSP	-	Digital Signal Processing
FT	-	Fourier Transform
FLOPS	-	Floating Point Operations
UTM	-	Universiti Teknologi Malaysia

LIST OF SYMBOLS

σ	-	Sigmoid Function
ζ	-	Softsign Function
$\cdot$	-	Dot Product

CHAPTER 1

INTRODUCTION

1.1 Problem Background

Deep neural networks (DNN) perform well in extracting key information from unstructured data which typically comes from the real-world environment. DNNs had been utilized in applications such as speech recognition [1], embedded vision [2], and health monitoring purpose [3]. DNNs are naturally computationally and memory intensive, therefore, they are normally implemented on advanced cloud compute servers. For IoT applications, this introduces several disadvantages. First, data transmission between edge sensors and the cloud compute servers consumes a lot of energy [4]. Besides, the latency of data transfer impacts the real time responsiveness of the edge devices [5]. Third, data privacy is also a concern since data on the cloud are much more vulnerable as compared to private storage [6]. Furthermore, in order to perform computation on cloud servers, a stable network connection is required, which could pose a problem when the DNNs is required to be implemented in secluded areas where network connectivity or congestion is a problem. All of the above problems can be addressed by implementing the DNN directly into the edge devices itself, aligning with the future trend of mobile edge computing [7].

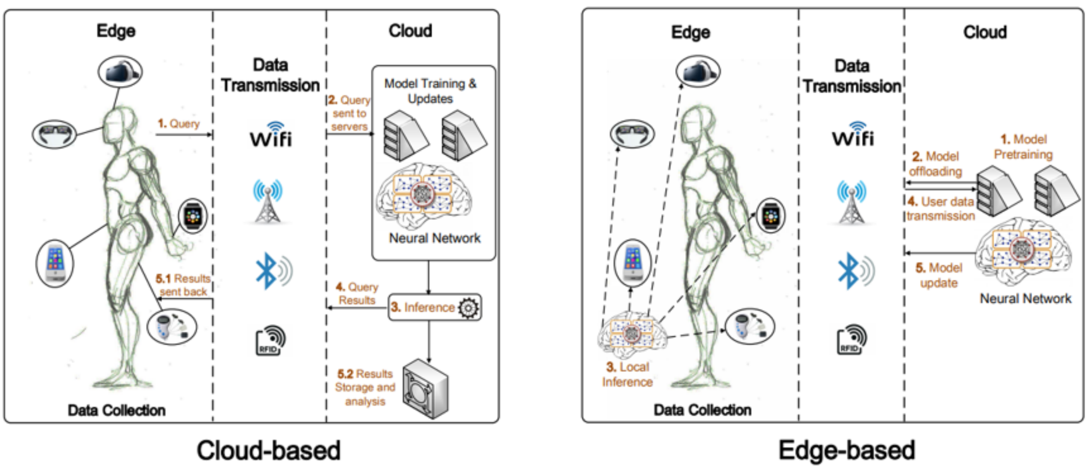

Figure 1.1 The Future of Edge-based Computing

There are two approaches to implement DNN into the edge devices. The first approach is to use customized hardware processors such as Apple A11 Bionic Chip and Nvidia Drive Px2. However, this approach is not suitable for low-powered applications. DNN has to use as little power as possible if it is to be implemented directly into edge devices. Edge devices are often restricted by their small size requirements, which directly influences the size of the battery. Since A11 and Px2 are very powerful, they consume a lot of energy if they are used in heavy tasks such as DNN. The battery will be depleted very fast and make it infeasible to implement DNN in them. The second approach is to use microcontrollers running lightweight software libraries such as TensorFlow Lite and CMSIS-NN [8]. By using a low power MCU and software libraries, the problem of power consumption can be solved, but it only provides a limited amount of memory and computational resources. The low power MCU might not be able to meet the requirements in order to execute properly. A solution that balances power consumption and computation resources is by implementing a simplified DNN structure on a slightly larger processor. In this report, we propose to run the gated recurrent unit (GRU) on a Cortex A53 on a Raspberry Pi. The small board can be used virtually anywhere with added benefits such as faster computation, low memory requirement and low power consumption. It can be observed from Table 1.1 that Cortex A53 is the cheapest and uses the least power among other existing competitors.

Table 1.1 Price-Power Comparison for Existing Technologies

	Google Coral	Intel Neural Compute Stick 2	Cortex A53 (RP3)
Price	RM 250	RM 350	RM 155
Power	0.5W per TOP	0.375W per TOP	159mW/MHz

TOP - Tera (10^{12}) Operations

1.2 Problem Statement

RNN are computationally and memory intensive which make it impossible to be implemented in edge devices. The conventional RNN model used are LSTM and GRU, with GRU consuming lower computation resource. This work aims to improve GRU to further reduce computation resources so that it can be implemented in edge device.

1.3 Research Questions

By referring to the problem statement above, several research questions arise:

- Does existing GRU architecture have some redundant features that can be removed to reduce computational power?
- Is it possible to enhance GRU architecture up to the point that it is able to operate on low power and low cost MCU (edge device)?
- How does this enhanced GRU perform as compared to conventional GRU model?

1.4 Research Goal

The aim of this project is to address the above challenges and develop a novel RNN architecture which is able to carry out and low memory computation on low power ARM MCU – which combines high efficiency signal processing functionality together with its low power and low-cost features so that it can be embedded directly onto the edge device without sacrificing too much performance.

1.5 Objective

The objectives are as follows:

- To review existing GRU architecture and identify field of improvements
- To enhance GRU architecture in order to produce a low computation and low memory consumption architecture.
- To compare accuracy, memory consumption and power consumption for the proposed architecture against the conventional architecture.

1.6 Scope and Limitation

In this project, the scopes are to:

- Improve the existing GRU network to result in a low memory and higher energy efficiency network.

REFERENCES

1. Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A. and Ng, A. Y. Deep Speech: Scaling up end-to-end speech recognition. *arXiv e-prints*, 2014: arXiv:1412.5567.

2. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. and Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv e-prints*, 2017: arXiv:1704.04861.

3. Hyun-Jung Kwak, G. and Hui, P. DeepHealth: Deep Learning for Health Informatics. *arXiv e-prints*, 2019: arXiv:1909.00384.

4. Miettinen, A. P. and Nurminen, J. K. Energy efficiency of mobile clients in cloud computing. *HotCloud*, 2010. 10(4): 19.

5. Dillon, T., Wu, C. and Chang, E. Cloud Computing: Issues and Challenges. *2010 24th IEEE International Conference on Advanced Information Networking and Applications*. 2010. 27–33.

6. Takabi, H., Joshi, J. B. D. and Ahn, G. Security and Privacy Challenges in Cloud Computing Environments. *IEEE Security Privacy*, 2010. 8(6): 24–31.

7. Zhang, C., Patras, P. and Haddadi, H. Deep Learning in Mobile and Wireless Networking: A Survey. *arXiv e-prints*, 2018: arXiv:1803.04311.

8. Lai, L., Suda, N. and Chandra, V. CMSIS-NN: Efficient Neural Network Kernels for Arm Cortex-M CPUs. *arXiv e-prints*, 2018: arXiv:1801.06601.

9. Sak, H., Senior, A. and Beaufays, F. Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition. *arXiv e-prints*, 2014: arXiv:1402.1128.

10. Staudemeyer, R. C. and Rothstein Morris, E. Understanding LSTM – a tutorial into Long Short-Term Memory Recurrent Neural Networks. *arXiv e-prints*, 2019: arXiv:1909.09586.

11. Bengio, Y., Simard, P. and Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 1994. 5(2): 157–166.

12. Dey, R. and Salem, F. M. Gate-variants of Gated Recurrent Unit (GRU) neural networks. *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*. 2017. 1597–1600.

13. Hochreiter, S. and Schmidhuber, J. Long Short-Term Memory. *Neural Comput.*, 1997. 9(8): 1735–1780. ISSN 0899-7667. doi:10.1162/neco.1997.9.8.1735. URL `https://doi.org/10.1162/neco.1997.9.8.1735`.

14. Cho, K., van Merrienboer, B., Bahdanau, D. and Bengio, Y. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. *arXiv e-prints*, 2014: arXiv:1409.1259.

15. Chung, J., Gulcehre, C., Cho, K. and Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv e-prints*, 2014: arXiv:1412.3555.

16. Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv e-prints*, 2014: arXiv:1406.1078.

17. Amoh, J. and Odame, K. Deep Neural Networks for Identifying Cough Sounds. *IEEE Transactions on Biomedical Circuits and Systems*, 2016. 10(5): 1003–1011.

18. Zhou, G.-B., Wu, J., Zhang, C.-L. and Zhou, Z.-H. Minimal Gated Unit for Recurrent Neural Networks. *arXiv e-prints*, 2016: arXiv:1603.09420.

19. Ravanelli, M., Brakel, P., Omologo, M. and Bengio, Y. Improving speech recognition by revising gated recurrent units. *arXiv e-prints*, 2017: arXiv:1710.00641.

20. Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. Teh, Y. W. and Titterington, M., eds. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Chia Laguna Resort, Sardinia, Italy: PMLR. 2010, *Proceedings of*

Machine Learning Research, vol. 9. 249–256. URL http://proceedings.mlr.press/v9/glorot10a.html.

21. Nwankpa, C., Ijomah, W., Gachagan, A. and Marshall, S. Activation Functions: Comparison of trends in Practice and Research for Deep Learning. *arXiv e-prints*, 2018: arXiv:1811.03378.

22. Lu, L., Shin, Y., Su, Y. and Karniadakis, G. E. Dying ReLU and Initialization: Theory and Numerical Examples. *arXiv e-prints*, 2019: arXiv:1903.06733.

23. Elliott, D. and Elliott, D. L. A better Activation Function for Artificial Neural Networks, 1993.

24. Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H. and Kalenichenko, D. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. *arXiv e-prints*, 2017: arXiv:1712.05877.

25. Chen, W., Wilson, J. T., Tyree, S., Weinberger, K. Q. and Chen, Y. Compressing Neural Networks with the Hashing Trick. *arXiv e-prints*, 2015: arXiv:1504.04788.

26. Denil, M., Shakibi, B., Dinh, L., Ranzato, M. and de Freitas, N. Predicting Parameters in Deep Learning. *arXiv e-prints*, 2013: arXiv:1306.0543.

27. Ba, L. J. and Caruana, R. Do Deep Nets Really Need to Be Deep? *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*. Cambridge, MA, USA: MIT Press. 2014, NIPS'14. 2654–2662.

28. Han, S., Mao, H. and Dally, W. J. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. *arXiv e-prints*, 2015: arXiv:1510.00149.

29. Wu, J., Leng, C., Wang, Y., Hu, Q. and Cheng, J. Quantized Convolutional Neural Networks for Mobile Devices. *arXiv e-prints*, 2015: arXiv:1512.06473.

30. Ott, J., Lin, Z., Zhang, Y., Liu, S.-C. and Bengio, Y. Recurrent Neural Networks With Limited Numerical Precision. *arXiv e-prints*, 2016: arXiv:1608.06902.

31. Gupta, S., Agrawal, A., Gopalakrishnan, K. and Narayanan, P. Deep Learning with Limited Numerical Precision. *arXiv e-prints*, 2015: arXiv:1502.02551.

32. Chen, Y., Luo, T., Liu, S., Zhang, S., He, L., Wang, J., Li, L., Chen, T., Xu, Z., Sun, N. and Temam, O. DaDianNao: A Machine-Learning Supercomputer. *2014 47th Annual IEEE/ACM International Symposium on Microarchitecture*. 2014. 609–622.

33. Vanhoucke, V., Senior, A. W. and Mao, M. Z. Improving the speed of neural networks on CPUs, 2011.

34. Hwang, K. and Sung, W. Fixed-point feedforward deep neural network design using weights +1, 0, and -1. *2014 IEEE Workshop on Signal Processing Systems (SiPS)*. 2014. 1–6.

35. Shin, S., Boo, Y. and Sung, W. Fixed-point optimization of deep neural networks with adaptive step size retraining. *arXiv e-prints*, 2017: arXiv:1702.08171.

36. Courbariaux, M., Bengio, Y. and David, J.-P. Training deep neural networks with low precision multiplications. *arXiv e-prints*, 2014: arXiv:1412.7024.

37. Pascanu, R., Mikolov, T. and Bengio, Y. On the difficulty of training Recurrent Neural Networks. *arXiv e-prints*, 2012: arXiv:1211.5063.

38. TensorFlow. *Speech Commands Dataset Version 1*, 2018 (accessed May 25, 2020). URL `http://download.tensorflow.org/data/speech_commands_v0.02.tar.gz`.

39. Kingma, D. P. and Ba, J. Adam: A Method for Stochastic Optimization. *arXiv e-prints*, 2014: arXiv:1412.6980.

40. Chen, H., Lundberg, S. and Lee, S.-I. Checkpoint Ensembles: Ensemble Methods from a Single Training Process. *arXiv e-prints*, 2017: arXiv:1710.03282.