POWER SCALABLE IMPLEMENTATION OF ARTIFICIAL NEURAL NETWORKS

Sankalp S. Modi, Peter R. Wilson & Andrew D. Brown

School of Electronics and Computer Science, University of Southampton

ABSTRACT

As the use of Artificial Neural Network(ANN) in mobile embedded devices gets more pervasive, power consumption of ANN hardware is becoming a major limiting factor. Although considerable research efforts are now directed towards low-power implementations of ANN, the issue of dynamic power scalability of the implemented design has been largely overlooked. In this paper, we discuss the motivation and basic principles for implementing power scaling in ANN Hardware. With the help of a simple example, we demonstrate how power scaling can be achieved with dynamic pruning techniques.

1. INTRODUCTION

Traditionally, Artificial Neural Networks(ANN)[1] involves simulations running on a conventional Von Neuman serial processor. Despite the tremendous growth in the computing power of general-purpose processors, it is insufficient for many ANN applications [2]. Neural networks are inherently parallel architectures and hence parallel processing using hardware techniques can provide significant speed improvements[2,3]. The increasing use of ANN in embedded devices has motivated the development of specialized hardware neural networks implementation [4,5].

The increasing demand for computational power of neural networks is leading to achieve low power VLSI circuits [6]. Furthermore, as the ANN is getting more pervasive in mobile embedded devices[7,8], the power requirement of ANN hardware is proving to be a major limitation [9]. As a result, there has been increasing emphasis on the low-power implementation of the ANN [6,10,11].

Despite these efforts, the issue of dynamic power scalability of the implemented design has been largely overlooked. In this paper, we discuss the motivation and basic principles for implementing power scaling in ANN Hardware. With the help of a simple example, we demonstrate how such scaling can be achieved with dynamic pruning techniques.

2. MOTIVATION

In order to understand the motivation behind implementing power scalable ANN, we first consider an example ANN application. Suppose that a typical multi-layer Feedforward ANN is used for noise reduction. Its input is a noisy signal and expected output is a noise free signal.(Several such Noise reduction and cancellation ANN applications have been reported in literature. [12-14]).

With the current ANN hardware approach, once the ANN hardware is designed and trained, its energy consumption during one forward pass remains almost constant throughout its operational period. This is because during the forward pass, the signal passes through the same number of neurons/connections and hence the number of arithmetic operations (addition/multiplication etc.) performed during one forward pass remains constant. With a constant supply voltage and clock speed, this will lead to almost constant power consumption. However, in a battery operated mobile applications, ability to trade-off power with other performance parameter (I.e. Mean Squared Error (MSE) in INN) is highly desirable. Current ANN hardware approaches do not support such dynamic Error-power trade-offs. Furthermore, we can also generally state that with an increase in input SNR, the complexity of the noise reduction task should decrease. With the reduction in the task complexity, it should be possible to reduce the required amount of processing and hence reduction in its power consumption. Current ANN hardware designs lack the ability to transform reduction in the task complexity into power saving by scaling power accordingly.

In this paper, we will demonstrate through simulation results that it is possible to obtain power reduction by scaling power according to the input noise level without any increase in MSE using a simple network pruning technique. It is interesting to note that amongst all well-explored pruning theories, there is no systematic study available that links dynamic pruning techniques to power scaling in ANN.

3. POWER SCALABLE IMPLEMENTATION: BASIC PRINCIPLES

3.1. Network Pruning and Power Consumption

Pruning methods gradually remove connections/nodes of a trained ANN. In general, removal of one connection saves power by decreasing one MAC (Multiply and Accumulate) instruction. The removal of a connection

also saves power during the learning and weight update phase. Thus, the power consumption is reduced with pruning of each connection.

The exact relationship between numbers of pruned connections and amount of power reduction can vary according to the implementation. The basic principles discussed in this paper just assume general positive correlation between them and do not depend on the exact mathematical relationship. Hence, for simplicity, we will assume for the rest of the discussion that the power reduction in ANN hardware implementation due to the pruning is approximately proportional to the number of pruned connections.

3.2. Pruning: Beyond the Improvement in Generalization

Generally, as we start pruning the trained ANN, initially the MSE slightly decreases due to improved generalization[15] or stays constant. But the capability of the ANN is ultimately limited by its size. Hence beyond certain pruning, the MSE starts increasing again. Our various experiments indicate the following approximate trend between increase in MSE and pruning. (Fig.1 - Region 1: the MSE is slightly decreased because of the improved generalization. Region 2: MSE increases with pruning in somewhat linear fashion. Region 3: MSE start saturating at very high MSE level)

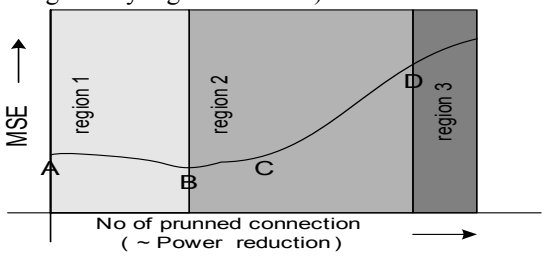

Figure 1. Increase in MSE with pruning of ANN.

3.3. Pruning in presence of variable SNR

Reconsider the noise reduction example discussed in section 2. Our experiments show that for the same ANN, MSE in the output is decreased with the increase in SNR (Fig. 2).

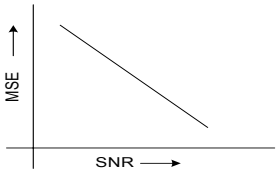

Figure 2. Decreasing trend of MSE with increase in SNR

It was observed during our experiments that this trend is also maintained when pruning is applied to ANN(Fig. 3). Generally, ANNs are designed to handle the worst-case scenario (i.e. `SNR_min`). With `SNR_min` in the input, ANN is pruned to obtained minimum error (point X). Network X is implemented in hardware with power reduction is P. However if during the operation of ANN, if the SNR is increased from the `SNR_min` to `SNR2`, we can prune the network further to the point Y within the same error margin and reduce the power further to point Q. Thus, characteristics in Fig. 3 present an attractive opportunity to transform increase in SNR (i.e. decrease in task complexity) into power reduction without any increase in Error through pruning (Depicted in Fig.4).

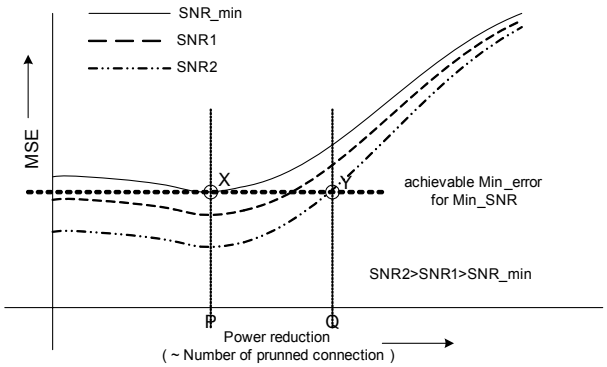

Figure 3. MSE vs Pruning with SNR variation

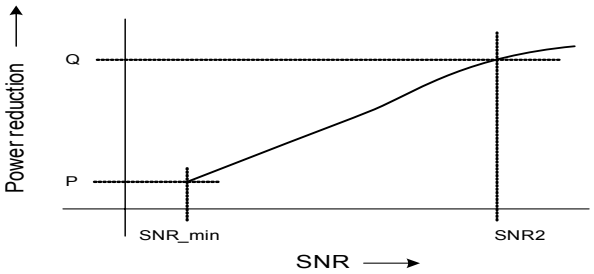

Figure 4. Power reduction with increase in SNR (i.e. decrease in task complexity) with constant MSE

To determine the appropriate level of pruning we need some kind of feedback mechanism. In many cases, the feedback can be obtained from the higher-level module that is utilizing filtered output from the ANN. For instance, if the noise cancellation ANN is preceding the speech recognition unit (as in [13]), the recognition unit will have to inform the ANN whether the current level of noise reduction is adequate for unambiguous speech recognition. The ANN will simply keep pruning itself as long as the recognition unit allows for it.

If pruning is used without any re-training (as done in our experiments described in the next section), then during the pruning process the connections are simply disabled, but their weight values are still stored in the memory. Hence, if the SNR drops again and MSE increases beyond maximum tolerable MSE, then we can simply 'grow' the network back by enabling the connections in the reverse order. If the pruning is used with the re-training, then we need to use appropriate growth methods with the training dataset to re-grow the network in case of a drop in SNR.

4. SIMULATION RESULTS

In [16], a 4-layer ANN was used for harmonic retrieval from noisy tone (SNR range: 0 db to –3 db). Each layer contains 60 neurons. For demonstration purposes, we have selected the same ANN and similar test dataset as used in [16]. The same 60x4 ANN architecture was also used for background noise reduction from speech signal in [13] and a very similar one was used in [17] for multi-tone detection. The ANN has 4 layers with 60 neurons in each and a total number of connections of 10800. The training signal is a 0.2 Hz sine wave sampled at 5 Hz. The network was trained using a combination of standard Back-propagation and Resilient back-propagation [18]. Simulations were performed using SNNS (Stuttgart Neural Network Simulator).

Once the network was trained, its performance in terms of MSE was measured for inputs with various SNR. The results are presented in the graph shown in Fig.5, which is in general agreement with the graph in Fig.2.

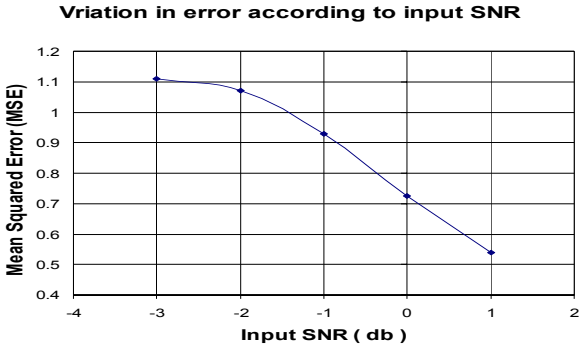

Figure 5. Variation in MSE according to input SNR

The trained ANN was then pruned using simple Magnitude Pruning method without any further training. In the Magnitude Pruning method, importance (saliency) of each connection is equal to its weight. This pruning method simply disables the connections with the least weight. The pruned network was then tested with signals with –3 db SNR to 0 db SNR. Fig.6 represents the simulation results showing the effect of pruning on MSE for different input SNR and they are consistent with the trends displayed in Fig.3.

Here we can see that an increase in MSE is not significant until the number of pruned connections is 2600 (point X). Hence, an ANN with 10800 – 2600 = 8200 connections should be implemented in hardware (ANN-H). However, after the ANN is implemented in hardware, we should dynamically scale its connectivity according to the input SNR in order to save power. The horizontal dashed line represents the minimum MSE possible to obtain with minimum SNR. With the SNR increase of 4 db (–3 db to 1 db), we can obtain about a 28% power reduction without any increase in MSE. (ANN-H with 8200 connections is considered as the network operating with 100% power.)

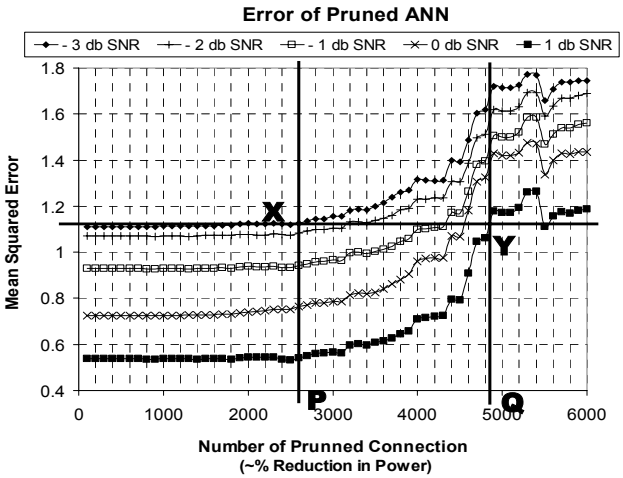

Figure 6. Error of pruned ANN for different input SNR

The graph of Fig.7 represents achievable Power Reduction as a function of input SNR for various tolerable MSE. The results corroborate our projections presented in Fig. 4.

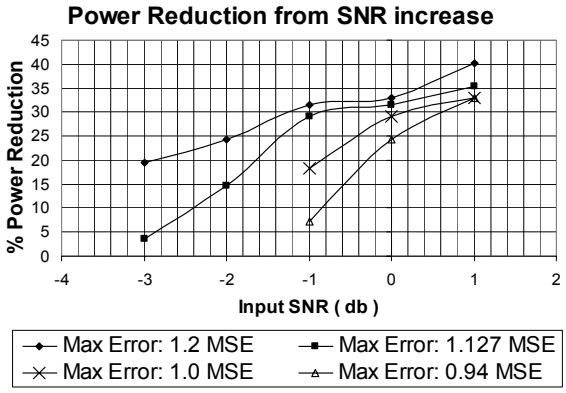

Figure 7. Achievable power reduction with increase in SNR (for different MSE value)

5. CONCLUSION AND FUTURE WORK

In this paper, we discussed the motivation for investigating power scalable ANN implementation and illustrated the basic principles with the help of an example noise reduction ANN application. The simulation results shows that using simple Magnitude Pruning, 4 db increase in SNR can be translated into about 28 % reduction in number of connections (and thus a significant power reduction) in ANN without any increase in MSE. These results demonstrate that it is possible to translate reduction in task complexity into power saving using dynamic pruning of ANN.

The Magnitude pruning method used in the experiment is a very simple pruning method and it was chosen for its simplicity and minimum overheads. Other sophisticated pruning method like Optimal brain damage[19] and

Optimal Brain Surgeon[20] are likely to produce superior results. Effects of different pruning and growth methods will be carried out in the next phase of our research.

The example discussed here is a simple noise reduction application with a typical feedforward network. However, the application of power scaling is not limited to this type of network/application. It is possible to apply it to other types of ANN architectures (K-map, auto-associative memories, Radial basis function etc.) for a variety of application with variable task complexity and further research in this direction is warranted. Pruning theories are generally well explored only for feedforward networks with an aim to reduce the generalization error. This work provided a strong motivation for further exploration of pruning and growth theories in the light of resulting power scalability for various types of ANN architectures.

6. REFERENCES

[1] S. Hyakin, *Neural Networks: A Comprehensive Foundation*, 2nd Edition, ed Prentice Hall, Upper Saddle River, New Jersey 07458, 1999.

[2] C. S. Lindsey and T. Lindblad, "Survey of neural network hardware," *Proceedings of the SPIE - The International Society for Optical Engineering*, vol. 2492, no. pt.2, pp. 1194-1205, 1995.

[3] R. Schuffny, A. Graupner, and J. Schreiter, "Hardware for neural networks," in 4th Int.Workshop Neural Networks Applications, Magdeburg, 1999.

[4] A. G. Andreou, R. C. Meitzler, K. Strohbehn, and K. A. Boahen, "Analog VLSI neuromorphic image acquisition and pre-processing systems," *Neural Networks*, vol. 8, no. 7-8, pp. 1323-1347, 1995.

[5] K. Chang-Min and Y. L. Soo, "A digital chip for robust speech recognition in noisy environment,"*2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221),* vol.2 ed Salt Lake City, UT, USA: IEEE, 2001, pp. 1089-1092.

[6] A. Koenig, A. Guenther, J. Doege, and M. Eberhardt, "Cell library of scalable neural network classifiers for rapid low-power vision and cognition systems design," *International Conference on Knowledge-Based Intelligent Electronic Systems, Proceedings, KES*, vol. 1, pp. 275-282, 2000.

[7] K. Majumdar and N. Das, "Neural networks for location management in mobile cellular communication networks,"*IEEE TENCON 2003. Conference on Convergent Technologies for the Asia-Pacific Region (IEEE Cat. No.03CH37503),* Vol.2 ed Bangalore, India: Allied Publishers Pvt. Ltd, 2003, pp. 647-651.

[8] L. F. Ni and B. Y. Zheng, "Application investigation of neural networks for uplink power control in CDMA mobile communications," *Nanjing Youdian Xueyuan Xuebao/Journal of Nanjing Institute of Posts and Telecommunications*, vol. 25, no. 1, pp. 1-8, 2005.

[9] K. Wawryn and A. Mazurek, "Low power, current mode circuits for programmable neural network,"*ISCAS 2001. The 2001 IEEE International Symposium on Circuits and Systems (Cat. No.01CH37196),* vol. 2 ed Sydney, NSW, Australia: IEEE, 2001, pp. 628-631.

[10] Y. Dumonteix, Y. Bajot, and H. Mehrez, "A fast and low-power distance computation unit dedicated to neural networks, based on redundant arithmetic,"*Materials Research Society Symposium - Proceedings,* 626 ed San Francisco, CA: Materials Research Society, 2001, pp. 878-881.

[11] L. Ravezzi, G. F. la Betta, and G. Setti, "Compact CMOS implementation of a low-power, current-mode programmable cellular neural network," *International Journal of Circuit Theory and Applications*, vol. 29, no. 3, pp. 299-310, May2001.

[12] K. Kasper, H. Reininger, and D. Wolf, "A neural network based adaptive noise reduction filter for speech recognition,"*Signal Processing VII, Theories and Applications. Proceedings of EUSIPCO-94. Seventh European Signal Processing Conference,* vol.3 ed Edinburgh, UK: Eur. Assoc. Signal Process, 1994, pp. 1701-1704.

[13] S. Tamura and A. Waibel, "Noise reduction using connectionist models,"*ICASSP 88: 1988 International Conference on Acoustics, Speech, and Signal Processing (Cat. No.88CH2561-9)* New York, NY, USA: IEEE, 1988, pp. 553-556.

[14] M. Trompf, "Neural network development for noise reduction in robust speech recognition,"*IJCNN International Joint Conference on Neural Networks (Cat. No.92CH3114-6)* Baltimore, MD, USA: IEEE, 1992, pp. 722-727.

[15] R. Reed, "Pruning algorithms - a survey," *IEEE Transactions on Neural Networks*, vol. 4, no. 5, pp. 740-747, 1993.

[16] S. S. Rao and P. M. Pisharam, "A noise-reduction neural network as a preprocessing stage in the SVD based method of harmonic retrieval,"*1990 IEEE International Symposium on Circuits and Systems (Cat. No.90CH2868-8)* New Orleans, LA, USA: IEEE, 1990, pp. 491-494.

[17] S. S. Rao and S. Sethuraman, "A neural network pre-processor for multi-tone detection and estimation,"*Neural Networks for Signal Processing. Proceedings of the 1991 IEEE Workshop (Cat. No.91TH0385-5)* Princeton, NJ, USA: IEEE, 1991, pp. 580-588.

[18] C. Hubert, "Pattern completion with the random neural network using the RPROP learning algorithm,"*Proceedings of the IEEE International Conference on Systems, Man and Cybernetics,* 2 ed Le Touquet, Fr: Publ by IEEE, Piscataway, NJ, USA, 1993, pp. 613-617.

[19] L. C. Yann, John S.Denker, and A. S. Sara, *Optimal brain damage* Morgan Kaufmann Publishers Inc., 1990, pp. 598-605.

[20] B. Hassibi, D. G. Stork, and G. J. Wolff, "Optimal brain surgeon and general network pruning," San Francisco, CA, USA: Publ by IEEE, Piscataway, NJ, USA, 1993, pp. 293-299.