EXPLORING THE OPTIMAL LEARNING TECHNIQUE FOR IBM NEUROSYNAPTIC SYSTEM TO OVERCOME QUANTIZATION LOSS

by

Hsin-Pai Cheng

Bachelor of Science, National Sun Yat-sen University, 2014

Submitted to the Graduate Faculty of

Swanson School of Engineering in partial fulfillment

of the requirements for the degree of

Master of Science

University of Pittsburgh

2017

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This thesis was presented

by

Hsin-Pai Cheng

It was defended on

November 29th, 2016

and approved by

Hai Li, Ph.D., Adjunct Associate Professor,
Department of Electrical and Computer Engineering

Yiran Chen, Ph.D., Adjunct Associate Professor,
Department of Electrical and Computer Engineering

Zhi-Hong Mao, Ph.D., Associate Professor,
Department of Electrical and Computer Engineering and Bioengineering

Thesis Advisor: Yiran Chen, Ph.D., Adjunct Associate Professor,
Department of Electrical and Computer Engineering

EXPLORING THE OPTIMAL LEARNING TECHNIQUE FOR IBM NEUROSYNAPTIC SYSTEM TO OVERCOME QUANTIZATION LOSS

Hsin-Pai Cheng, M.S.

University of Pittsburgh, 2017

Inspired by the fact that human brain is much more efficient than any nowadays computers, neuromorphic computing is aim at performing near human brain ability of processing huge amount of data in an extreme short time. For the hardware part, neuromorphic computing is also extended to systems facilitating the computation of neural network and machine learning algorithms. Recently, IBM Neurosynaptic system is one of the well-known project that dedicated on energy-efficient neural network applications. However, However, one of the known issues in TrueNorth design is the limited precision of synaptic weights, each of which can be selected from only four integers. To improve the computation accuracy and reduce the incurred hardware cost, in this work, we investigate seven different regularization functions in the cost function of the learning process on TrueNorth platform. Our experimental results proved that the proposed techniques considerably improve the computation accuracy of TrueNorth platform and reduce the incurred hardware and performance overheads.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

Ever since I stepped into the laboratory, I was determined to be the one who can inspire those who come after me, just like my mentors. To materialize this aspiration, beyond putting countless efforts in the courses of my major, I spent much time on machine learning research and publishing papers. The knowledge I have acquired and the hands-on experiments I have been through all equip me with what I shall need in the coming days.

I want to thank Dr. Yiran Chen and Dr. Hai (Helen) Li. They are the driving force behind all my contribution and they always excite my potential. I also thank Kent Nixon, Xiang Chen, Chupeng Wu, Wei Wen, Elizabeth Koo for the selfless help and advice. I thank my parents always supporting me.

1.0 INTRODUCTION

The modern computing industry is primarily constructed atop von Neumann architecture, which separates computation and data processing from main storage [1]. Following technology scaling, the increasing gap between limited memory bandwidth and large computing power offered by multicore microprocessors becomes prominent and starts to hinder the performance upscaling of computer systems [2]. Moreover, it has been shown that von Neumann architecture is not very efficient in processing cognitive applications that often require qualitative rather than quantitative results [3]. Neuromorphic computing, which denotes the VLSI systems that mimic neuro-biological architecture, was recently attracted study as an alternative solution of von Neumann architecture for future computing system development. The concept of neuromorphic computing is also extended to systems facilitating the computation of neural network and machine learning algorithms [4]. Various platforms including CPUs [5], GPUs [6], FPGAs [7], and ASIC chips [8]are developed to satisfy different tradeoffs among performance, power, adaptability, programmability, and scalability. Inspired by the spiking-event driven structure of human brains, IBM released a special ASIC neuromorphic computing chip named TrueNorth in 2014. A TrueNorth chip contains 5.4 billion transistors with 4,096 neurosynaptic cores, 1 million digital neurons, and 256 million synapses. It is structured with 100 billion parallel computational units (neurons) [9]. The TrueNorth is able to achieve a peak computation of 58 Giga synaptic operations

per second with only 65mW in power consumption, making it extremely efficient and promising in neuromorphic computing applications [4].

The TrueNorth achieves the high computing efficiency and the large supported computing model scale by sacrificing the precision of the signals and data computed and propagated on the system. For example, all communications between the neurosynaptic cores are represented by binary spikes. And the synaptic weights are stored as integers with only four values [10]. The low resolution of the data representation induces quantization loss when mapping a trained neural network in floating-point data format onto a TrueNorth chip and therefore, results in inference accuracy degradation. To compensate the quantization loss, stochastic computing methods are adopted when implementing a design on TrueNorth: data is statistically represented in both temporal and spatial domains, i.e., using multiple spikes or hardware copies of the neural networks [11]. The final outcome is obtained by averaging the results of the redundant computations. Such a workaround prolongs the execution time and increases the required hardware resources.

In our initial analysis on the model learning and deploying process of the TrueNorth, we found that except for using redundant copies to represent high-precision data, another way to reduce the quantization loss is to minimize the discrepancy between the trained weights and the quantized weights of the network. In this work, we systematically study the effect on system quantization when applying seven different regularization methods in the cost function of the training process of the TrueNorth. Our experiments show that the L1TEA regularization gives the best inference accuracy on the TrueNorth after quantization among all seven tested regularizations – the trained weights are effectively clustered around the available integer representations, significantly reducing the quantization loss.

2.0 TRUENORTH PRELIMINARY

Figure 1 depicts the architecture of the TrueNorth chip. The chip is composed of a scalable network of configurable neurosynaptic cores, each of which contains memory (“synapses”), processors (“neurons”), and communication (“axons”) in close proximity. The inter-core communication is carried by all-or-none spike events over a message-passing network. The numbers of the neurons and the axons in a neurosynaptic core are limited at 256 and one axon in a core can connect to only one neurosynaptic core. The TrueNorth is physically designed to efficiently implement regular neural networks, which must be able to be mapped to densely connected 256×256 local proximities.

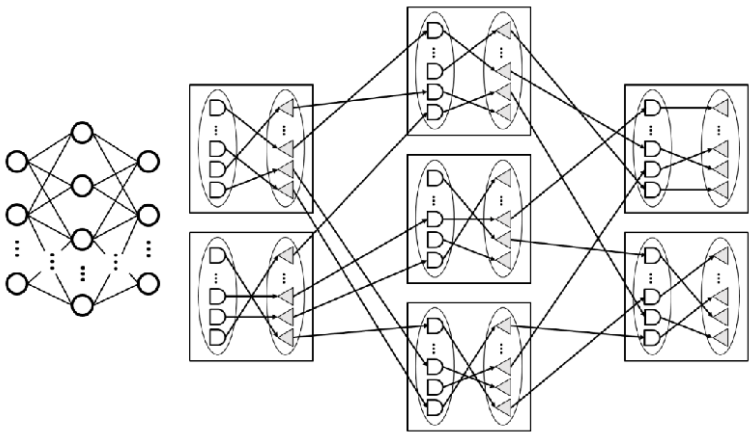

Figure 1. Architecture of the TrueNorth chip.

Traditional neural networks often require utilization of high precision floating point computations, and therefore is unsuitable to low precision integer computations on the True North. Additionally, the state-of-the-art training algorithms also must be performed in the continuous and differentiable space. However, the activation information in the TrueNorth is represented as binary spikes; the synaptic weight at each cross-point of crossbar structure in a neurosynpatic core is quantized to a low-resolution integer, which is selected from only four candidates. Accordingly, the quantization effect is the major cause of inference accuracy degradation of the model after being deployed onto the TrueNorth chip.

The current workaround [10], as illustrated in Figure 2, utilizes stochastic computing concept to mitigate the adverse impact of quantization in both temporal and spatial domains:

- *Temporal copying*: multiple ticks of spikes are randomly sampled in a frame so that the math expectation of the spikes can approximate the floating-point inputs or activations;
- *Spatial copying*: multiple copies of the trained neural networks are randomly instantiated so that the math expectation of the weights can approximate the trained floating-point values.

In original mathematic representation, for example, for a neuron with input vector **x**, weight vector **w**, activation function $f(\cdot)$, the output of the neuron a can be calculated by:

$$y = \mathbf{x}^T\mathbf{w}, \text{ and} \tag{1a}$$

$$a = f(y). \tag{1b}$$

Here y is sum of the products of respective input and weight. The McCulloch-Pitts neuron model that is adopted by the TrueNorth can be expressed by:

$$y' = (\mathbf{x}')^T \cdot \mathbf{w}' \text{ , and} \tag{2a}$$

$$a' = \begin{cases} 1, y' \geq 0 \\ 0, y' < 0 \end{cases}. \tag{2b}$$

Where $\mathbf{x}'$ and a' are the binary spike approximations of inputs and activations, respectively. $\mathbf{w}'$ is the vector of the synaptic weights selected from four integer values, which can be either positive or negative (for simplicity, we only depict two integers in Figure 2).

It is known that the accuracy of stochastic computing can be improved by raising the number of the temporal and/or spatial copies and averaging the result of these copies. Although the same tricks can be applied to the TrueNorth, say, increasing the number of ticks and network copies, they prolong execution time or occupy more hardware cores significantly. This intrinsic drawback of stochastic computing greatly hinders the scalability of the TrueNorth platform.

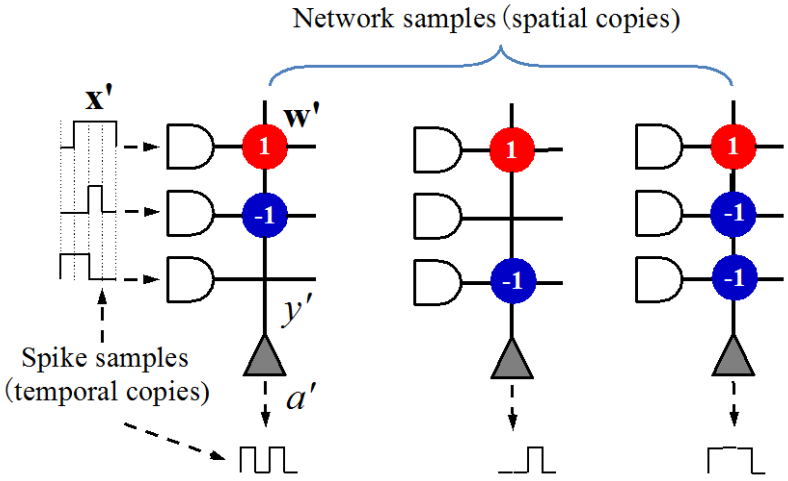

Figure 2. Temporal and special copying's of the TrueNorth.

3.0 METHODOLOGY

3.1 MODEL DEPLOYMENT ON THE TRUENORTH

As aforementioned in Section 2, the discrepancy between the trained weights and the quantized weights is the major cause of quantization loss of the TrueNorth. In other words, if we can minimize such a discrepancy, the quantization loss should be reduced accordingly.

To better understand the generation of quantization loss, we mathematically formulate the quantization process on the TrueNorth as follows: Given a trained weight w_i in floating-point format, the TrueNorth first selects one (c_i) of four integer levels (c_0, c_1, c_2, c_3) on the synapse at the corresponding cross-point of the neurosynaptic core; The connectivity (ON/OFF) on the cross-point is then randomly sampled with a sampling probability $p_i = w_i/c_i$. Mathematically, the expectation of the synaptic weight (w_i') on TrueNorth equals the trained floating-point weights as:

$$E\{w_i'\} = p_i \cdot c_i = \frac{w_i}{c_i} \cdot c_i = w_i \ . \tag{3}$$

Based on Equation (3), the variance of w_i' can be calculated by:

$$var\{w_i'\} = c_i^2 p_i(1 - p_i) = w_i(c_i - w_i) \ . \tag{4}$$

Equations (3) and (4) show that there are two parameters affecting the deployment accuracy of the trained neural networks on the TrueNorth: the quantized weights c_i and the sampling probability p_i. Since p_i is a value within [0, 1], the selection of c_i basically defines the bounds of the trained weights.

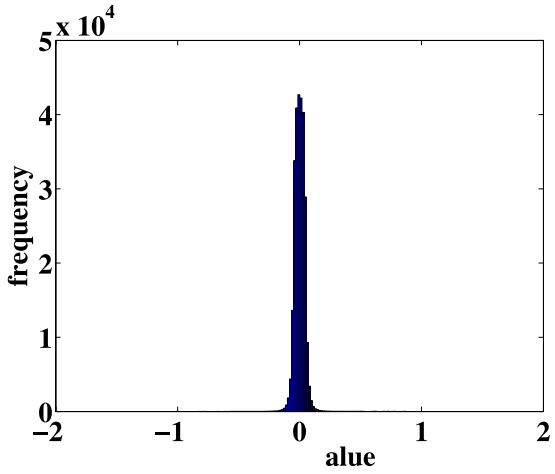

Figure 3. Weight distribution example of a neural network.

In practice, the trained weights of the neural network are often normalized or carefully bounded within a range, e.g., between [–2, +2]. Figure 3 plots the weight distribution of a trained neural network, which is composed of 784 input neurons, one hidden layers with 300 neurons, and 10 output classifier neurons. The weight distribution is well bounded between [–2, +2]. Four integers: –2, –1, +1, and +2 can be selected to represent the trained weights on the TrueNorth. For simplicity, in this work, we use this integer configuration in our experiments. If the values of the trained weights is beyond the range between [–2, +2], they can be easily normalized to the range.

Our experiment result shows that the trained network in Figure 3 achieves a test accuracy of 95.27%. However, after deploying the learned neural network to the TrueNorth, the test accuracy drops down to 90.04%.

3.2 ANALYSIS ON VARIANCE OF THE WEIGHTS

Figure 4 shows the relationship between the trained floating-point weight w_i and its variance on the TrueNorth. When the weight is between [–1, 1], c_i will be selected as either –1 or 1 depending on the sign of the weight. The largest variance, hence, happens at $w_i = 0.5$, leading to a variance of 0.5. The smallest variance, which is zero, happen at –1/1 and 0. It implies that the trained weights of –1, 0, and 1 has no quantization loss after deployment. The corresponding $p_i = 1 \; or \; 0$, respectively, at –1/1 and 0.

When the weight w_i is between [–2, –1) or (1, 2], the largest variance happens when the $|w_i|$ just slightly larger than 1. In such a case, w_i has to be quantized to –2 or 2 since value of p_i is bounded between [0, 1]. Similarly, the smallest variance, which is zero, happens at –2 and 2.

The above analysis inspires the thinking that if we can intentionally train the weight distributions around the local minimums of the variance (the quantized levels in this case), we shall be able to reduce the accumulated variance of the weights and consequently minimize the quantization loss.

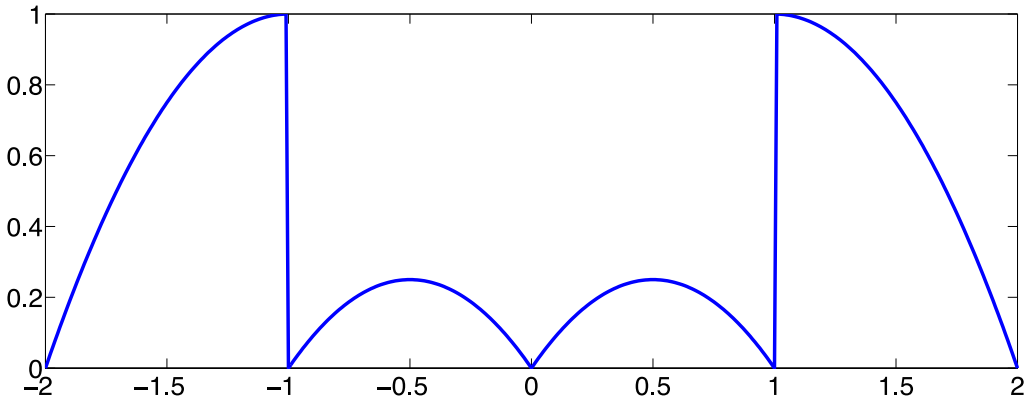

Figure 4. Weight distribution example of a neural network.

3.3 ANALYSIS ON QUANTIZATION LOSS

The above discussion shows that in TrueNorth, the quantization of weights $\boldsymbol{W}$ is performed through the quantization of sampling probability at the synapse. Here $\boldsymbol{W}$ is a $m \times n$ matrix. Without loss of generality, the discrepancy (D) between $\boldsymbol{W}$ and the quantized values $\boldsymbol{W_q}$ can be defined as:

$$D = \|\boldsymbol{W} - \boldsymbol{W_q}\| = \sum_{i=1}^{m} \sum_{j=1}^{n} |w_{ij} - w_{q,ij}| \,, \tag{5}$$

where w_{ij} and $w_{q,ij}$ are the i-th row, j-th column element of $\boldsymbol{W}$ and $\boldsymbol{W_q}$, respectively.

Considering an extreme case that the trained weight distribution follows uniform distribution $U(0, 1)$, the weights in [0, 0.5) will be quantized to 0 and the weights in [0.5 , 1] will be quantized to 1. When $\boldsymbol{W}$ is large enough, the sum in D can be represented as definite integration form. Thus, the discrepancy depicted in Equation (5) can be written as:

$$D \approx \int_0^{0.5} w \, dw + \int_{0.5}^{1} (1 - w) \, dw = 2 \cdot \int_0^{0.5} w \, dw = \frac{1}{4} \,. \tag{6}$$

If we can train the neural network so that the trained weights cluster around the quantized levels, e.g., reshape the weight distribution to the shape shown in Figure 5, the accumulated quantization discrepancy D' of this new distribution can be calculated as:

$$D' \approx 2 \cdot \int_0^{t} \left(-\frac{1}{t^2} w + \frac{1}{t} \right) \cdot w \, dw = \frac{t}{6} \leq \frac{1}{12} < D \,. \tag{7}$$

Here $t \leq 0.5$. The results of Equations (6) and (7) proved our hypothesis discussed in Section 3.1, which motivated us to optimize the learning process for weight distribution reshaping.

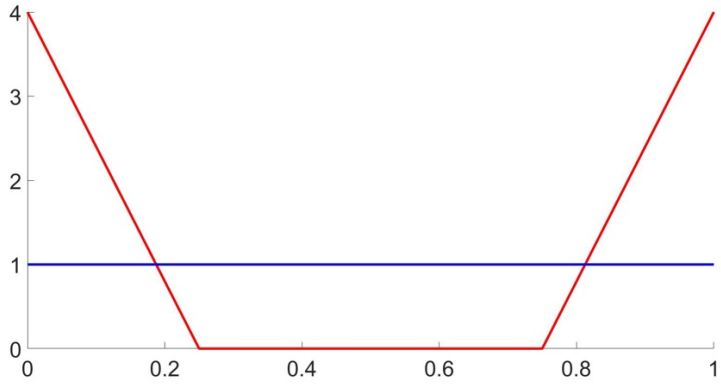

Figure 5. Original weight distribution (blue) and theoretical reshaped weight distribution (red) with $t = 0.25$.

3.4 THE PROPOSED METHODS

Overfitting describes the considerable accuracy drop between the training and testing processes of a trained neural network. It is because during training process, too many details of the training set are captured so that the generality of the trained model is lost. To eliminate overfitting effect and increase testing accuracy, regularization is introduced to the training process. In general, regularization can be expressed as a penalty term in the cost function of learning process as:

$$J' = J + \lambda R(J)\ J' = J + \lambda \cdot R(J). \tag{8}$$

Here coefficient λ defines the significance of the regularization term and $R(J)$ is the added penalty on top of the original cost function J.

Two commonly used regularizations are L1-norm and L2-norm regularizations as:

$$R(J) = \|w\| R_{L1}(J) = ||w||, \text{ and} \tag{9a}$$

$$R_{L2}(J) = ||w||^2, \tag{9b}$$

respectively, as shown in Figure 6(a) and (b). The hypothesis behind these two regularization methods is that a bigger weight is likely to produce a larger impact on the system result. Since the object of training process is minimizing the cost function, the existence of L1-norm or L2-norm regularizations will bias the trained weight distribution toward zero, as we shall show in Section 4. In other words, regularizations may be leveraged to reshape the trained weight distribution toward the quantized levels on the TrueNorth.

Besides L1-norm and L-2 norm, we also try the following regularizations in the training process of the TrueNorth:

1) ***L1TEA***: a penalty curve consists of two linear curves with the highest penalty at the probability of 0.5. L1TEA bias the probability (p_i) of the weight (w_i) on the TrueNorth toward both 0 and 1.

2) ***Sine***: a penalty curve defined by sine function. Different from L1TEA, the derivative of Sine changes with different probability values.

3) ***Circle_11***: a penalty curve that is defined as a circle centered at (1, 1) with radius of 1. Circle_11 offers a penalty curve that is almost symmetric to L2-norm, which generally pushes the weight distribution to high value side;

4) ***Circle_00***: a penalty curve that is defined as a circle centered at (0, 0) with radius of 1. Similar to Circle_ 11, Circle_00 also pushes the weight distribution low value side. We note that Circle_00 has the highest derivative when the weight probability is 1 while Circle_11 has the lowest derivative when the weight probability is 1.

5) ***Circle_0111***: a penalty curve consists of two circle curves jointed at (0.5, 1).

Note that in all the sub-figures in Figure 6, the x-axis is the probability p_i instead of the actual weight value (w_i). The y-axis is the penalty value.

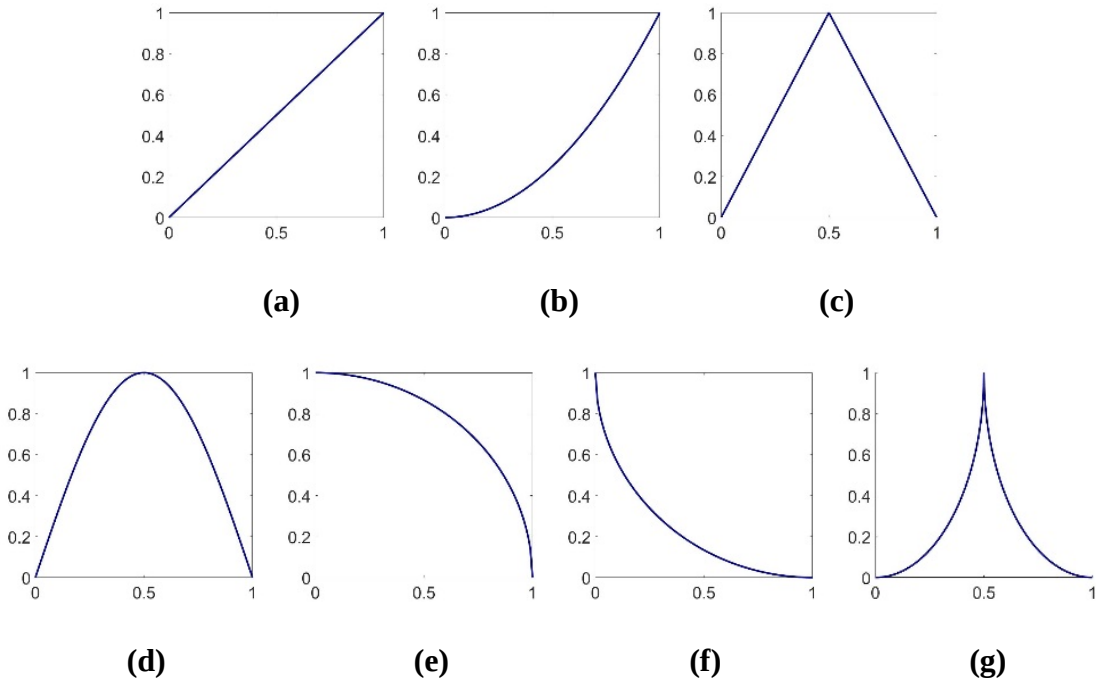

Figure 6. Four regularizations investigated in this thesis.

4.0 EXPERIMENTAL RESULTS

4.1 EXPERIMENTAL SETUP

We conducted our experiment on both the IBM Neuro Synaptic Chip Simulator and the NS1e development platform hardware, which contains one IBM TrueNorth chip. The handwritten digit classification on MINIST database is used as testbench in the work. This network utilizes 4 neuro-synaptic cores, each of which receives a 16×16 (256 pixels) block from the image via its input neurons, with output axons from all neurosynaptic cores being merged to 10 output classes for digit classification. For use of TrueNorth, pixel values are scaled to [0, 1] and converted to spikes. The network is trained in Caffe (a CPU and GPU framework for deep learning [6]) and deployed to TrueNorth by randomly sampling synaptic connections.

The neural network is trained by following Tea learning method integrated with the regularization methods. The probability distributions of synaptic weights under different training conditions are collected in Figure 7. And the digit recognition accuracy of these networks before and after deploying on the TrueNorth chip are summarized in Table 1. In the following subsections, we will evaluate efficiencies of these regularization functions in TrueNorth implementation.

4.2 BASELINE WITHOUT REGULARIZATION FUNCTION

The baseline utilizes the original TrueNorth Tea learning method without applying any regularization function. As can be seen in Figure 7(a), majority of the probability values are spreading from 0 to 1, thought 0 and 1 has relatively higher ratio. Such a distribution indeed is corresponding to the hypothesis in Section 3.2.

Our experiment shows that the network obtained in Caffe with floating-point precision has a test accuracy of 95.27%. After deploying the same learned model to TrueNorth by sampling the connectivity of synapses with the learned connection probability $\boldsymbol{p}$, however, the test accuracy quickly drops to 90.04%. Such a big accuracy loss is mainly resulted by the date quantization and therefore is denoted as quantization induced accuracy loss.

The recognition accuracy can be partially compensated if instantiating more copies of the networks and averaging their results. However, the total neurosynaptic cores required by this workaround sharply increases as the design scale and complexity grow up. It is not a scalable solution.

Table 1. The Accuracy on Caffe and TrueNorth

	Caffe	**TrueNorth**	**Difference**	**λ**
No Penalty	95.27	90.04	5.23	0
L1-norm	95.36	89.83	5.53	10^{-5}
L2-norm	96.04	88.36	7.68	10^{-4}
L1TEA	95.03	92.78	2.25	10^{-4}
Sine	95.16	91.58	3.58	10^{-5}
Circle_0111	95.06	90.7	4.36	10^{-3}
Circle_11	95.06	90.2	4.86	10^{-3}
Circle_00	95.09	90.59	4.5	10^{-4}

4.3 CONVENTIONAL REGULARIZATIONS

L1-norm and L2-norm are widely adopted in training neural networks to eliminate the overfitting effect. As the two regularization methods increase the penalty of higher weight values, both probability distributions in Figure 7(b, c) demonstrate the trend of clustering the weights toward 0. The weight probability distribution obtained by applying L2-norm gradually drops as probability increases from 0 to 1, while applying L1-norm results in more abrupt change in probability distribution. This is because the penalty of L2-norm changes gradually at small weight values, while L1-norm follows a fixed gradient corresponding to high penalty and restriction, as shown in Figure 6(b, c).

For the given example, applying L1-norm to the learning method shows similar performance as the baseline: the test accuracy obtained in Caffe or the TrueNorth is 95.36% or 89.83%, respectively. The baseline produces a slightly better result on the TrueNorth than the L1-norm function.

Very interestingly, our experiment shows that when deploying the L2-norm regularization, the network obtained in Caffe with floating-point precision achieves the highest test accuracy of 96.04% among all the learning methods investigated in the work. Unfortunately, the deployment on TrueNorth shows a test accuracy of merely 88.36%, which is the lowest result among all the tests. Such a big discrepancy is because a large number of weights fall into the range between 0 and 0.5 as shown in Figure 7(c). This results in a big accuracy loss during data quantization.

4.4 L1TEA AND SINE REGULARIZATION

L1TEA combines two linear curves and scores the highest penalty to the weight of 0.5. Accordingly, the probability of weights is pushed and converged close to 0 and 1. The scenario has been clearly demonstrated in Figure 7(d).

Due to the high penalty and strong constrain of L1TEA regularization, the network obtained in training phase has a relatively lower test accuracy of 95.03%. However, since most of the weights during the training process are clamped to 0 and 1, the design suffers less on quantization induced accuracy loss. Its deployment on TrueNorth obtains the highest test accuracy of 92.78%. The accuracy difference between Caffe and TrueNorth is only 2.25%.

The sine regularization follows the similar attempt as L1TEA and therefore demonstrates similar weights distribution, as shown in Figure 7(e). The major difference is the hill shape concentrated in the range from 0.4 to 0.6 in probability distribution. This comes from the intrinsic characteristic of the sine function – the zero gradient at the peak. Thus, it is difficult to push the probability at (or close to) the peak to another specific value.

The sine function has a positive effect to TrueNorth. It is the second best among all the functions being used. It raises the accuracy of TrueNorth implementation to 91.58%, which is 1.54% higher than that of the baseline.

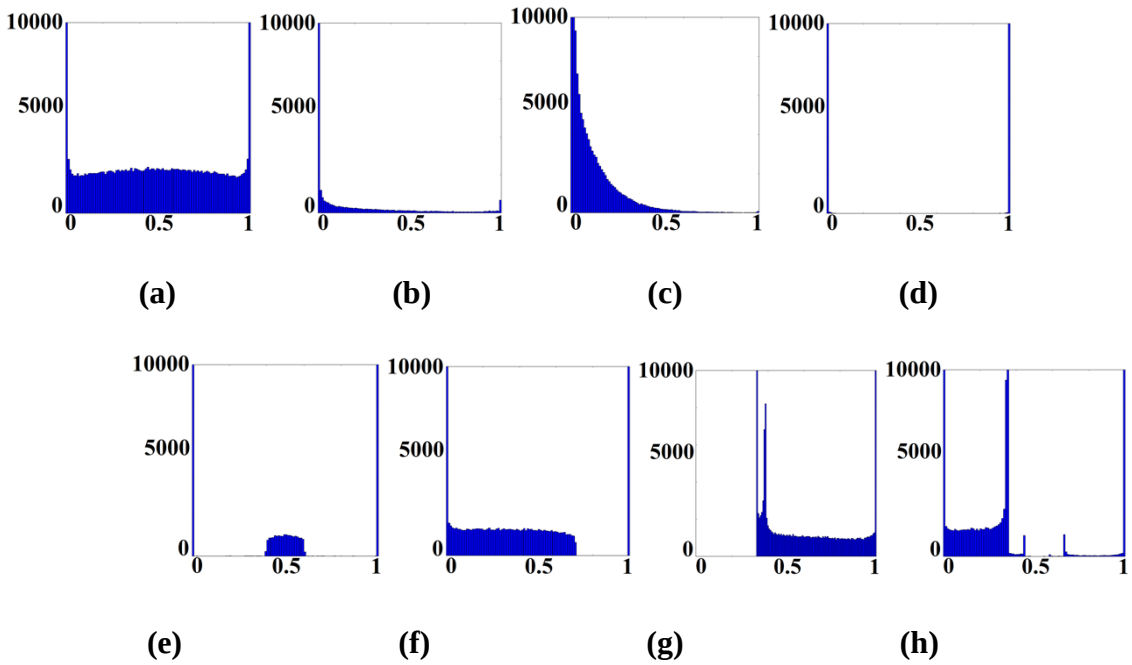

Figure 7. Probability distribution. (a) Without penalty (b) L1-norm (c) L2-norm (d) L1TEA (e) Sine function (f) Circle_00 (g) Circle_11 (h) Circle_0111.

4.5 CIRCLE REGULARIZATIONS

The circle_00 function attempts to push the weight probability toward the point of 1. The experimental result in Figure 7(f), however, shows that it is not able to diverge the small probability value (less than 0.72) to 1. The explanation is that the gradient of penalty function from 0 to 0.72 is too small to penalize the weight. The gradient of large value is big enough to realize the penalization.

The circle_11 function utilizes the similar concept as the circuit_00, except it tends to squeezes the weight probability to 0. Accordingly, Figure 7(g) shows a similar trend as the results in Figure

7(f). The regularization functions works fine for smaller probability but is not successful as the probability is larger than 0.36.

The circle_0111 function tends to restrain the probability to 0, 0.37, 0.65 and 1. There is a high peak at 0.37, which means that many probability was lower than 0.5 before regularization.

Based on the weight probability distributions, none of the three circle functions investigated in the work can obtain (even close to) binary data regularization. The performance of the corresponding TrueNorth implementations is almost same as the accuracy obtained by the baseline design without applying any penalty function.

5.0 CONCLUSION AND FUTURE WORK

The release of IBM Neurosynaptic System announces a new age of commercial semiconductor neuromorphic computing chip. Although very comprehensive software and simulators are provided to ease the adoption of the TrueNorth, there still exist many rooms for further improvement. As one example presented in this work, quantization loss of the TrueNorth can be reduced by reshaping the distribution of the trained weights to cluster around the quantized levels. We systematically study the effects of using different regularization terms in the learning process of the TrueNorth. Among the seven tested options, L1TEA function offers the best post-deployment inference accuracy on the TrueNorth. Experiment results also show that such an accuracy improvement can be translated to performance enhancement or hardware cost reduction.

BIBLIOGRAPHY

[1] G. Myers, “Advances in computer architecture,” John Wiley & Sons, 1982.

[2] S. A. McKee, “Reflections on the memory wall,” in the 1st ACM International Conference on Computing Frontiers, p. 162, 2004.

[3] A. S. Cassidy, P. Merolla, J. V. Arthur, S. K. Esser, B. Jackson, R. Alvarez-Icaza, P. Datta, J. Sawada, T. M. Wong, V. Feldman, A. Amir, D. B. D. Rubin, F. Akopyan, E. McQuinn, W. P. Risk, and D. S. Modha, “Cognitive computing building block: A versatile and efficient digital neuron model for neurosynaptic cores,” in the 2013 International Joint Conference on Neural Networks, pp. 1-10, 2013.

[4] F. Akopyan, J. Sawada, A. Cassidy, R. Alvarez-Icaza, J. Arthur, P. Merolla, N. Imam, Y. Nakamura, P. Datta, N. Gi-Joon, B. Taba, M. Beakes, B. Brezzo, J. B. Kuang, R. Manohar, W. P. Risk, B. Jackson, and D. S. Modha, “TrueNorth: Design and Tool Flow of a 65 mW 1 Million Neuron Programmable Neurosynaptic Chip,” in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 34, pp. 1537-1557, 2015.

[5] V. Vanhoucke, A. Senior, and M. Z. Mao, “Improving the speed of neural networks on CPUs,” Deep Learning and Unsupervised Feature Learning NIPS Workshop, 2011.

[6] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” in the ACM International Conference on Multimedia, pp. 675-678, 2014.

[7] M. Naylor, P. J. Fox, and S. W. Moore, “Managing the FPGA memory wall: Custom computing or vector prcessing,” International Conference on Field Programmable Logic and Applications, pp. 1-6, 2013.

[8] S. K. Esser, R. Appuswamy, P. A. Merolla, J. V. Arthur, D. S. Modha, “Backpropagation for energy-efficient neuromorphic computing,” Advances in Neural Information Processing Systems, pp. 1117-1125, 2015.

[9] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura and B. Brezzo, “A million spiking-neuron integrated circuit with a scalable communication network and interface,” Science, 345(6197), pp. 668-673, 2014.

[10] S. K. Esser, A. Andreopoulos, R. Appuswamy, et al., “Cognitive computing systems: Algorithms and applications for networks of neurosynaptic cores,” in the 2013 International Joint Conference on Neural Networks, pp. 1-10, 2013.

[11] A. S. Cassidy, R. Alvarez-Icaza, F. Akopyan, et al., “Real-time scalable cortical computing at 46 giga-synaptic OPS/watt with ~100× speedup in Time-to-Solution and ~ 100,000× reduction in Energy-to-Solution,” in the International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 27-38, 2014.