Master's Thesis
석사 학위논문

Joint Optimization of Computational Accuracy and Algorithm Parameters for Energy-Efficient Recognition Algorithm

`

Heesung Lim (임 희 성 林 喜 成)

Department of Information and Communication Engineering

정보통신융합전공

DGIST

2016

Master's Thesis
석사 학위논문

Joint Optimization of Computational Accuracy and Algorithm Parameters for Energy-Efficient Recognition Algorithm

Heesung Lim (임 희 성 林 喜 成)

Department of Information and Communication Engineering

정보통신융합전공

DGIST

2016

Joint Optimization of Computational Accuracy and Algorithm Parameters for Energy-Efficient Recognition Algorithm

Advisor: Professor Jae Yoon Hwang

Advisor: Professor Minkyu Je

Co-Advisor: Professor Taejoon Park

by

Heesung Lim

Department of Information and Communication Engineering

DGIST

A thesis submitted to the faculty of DGIST in partial fulfillment of the requirements for the degree of Master of Science in the Department of Information and Communication Engineering. The study was conducted in accordance with Code of Research Ethics[1]

. .

Approved by

Professor (Advisor)	Jae Yoon Hwang	(Signature)
Professor (Advisor)	Minkyu Je	(Signature)
Professor (Co-Advisor)	Taejoon Park	(Signature)

[1] Declaration of Ethical Conduct in Research**:** I, as a graduate student of DGIST, hereby declare that I have not committed any acts that may damage the credibility of my research. These include, but are not limited to: falsification, thesis written by someone else, distortion of research findings or plagiarism. I affirm that my thesis contains honest conclusions based on my own careful research under the guidance of my thesis advisor.

Joint Optimization of Computational Accuracy and Algorithm Parameters for Energy-Efficient Recognition Algorithm

Heesung Lim

Accepted in partial fulfillment of the requirements for the degree of Master of Science.

. .

Head of Committee _____(인)

Prof. Jae Yoon Hwang

Committee Member _____(인)

Prof. Minkyu Je

Committee Member _____(인)

Prof. Taejoon Park

201422016 임 희 성. Heesung Lim. Joint Optimization of Computational Accuracy and Algorithm Parameters for Energy-Efficient Recognition Algorithm. Department of Information and Communication Engineering. 2016. 36p. Advisors Prof. Jae Yoon Hwang, Advisors Prof. Minkyu Je, Co-Advisor Prof. Taejoon Park.

ABSTRACT

The need for human-machine interaction such as speech and gesture recognition has steadily grown in wearable devices. As applications become more intelligent such as facial emotion recognition, a variety of recognition algorithms has been developed and evolving. However, as the recognition algorithms become more complex, the more computation is required to perform the application in a limited battery capacity of wearable devices, which means that energy-efficiency is critical issue. In this thesis, one of the widely used recognition algorithm, artificial neural network (ANN), is selected as a target algorithm and its characteristic, inherent algorithmic fault tolerance (AFT), is adopted to improve energy-efficiency. To compute the recognition algorithm (ANN), Significant-driven iterative approximate multiplier (SDIAM) is utilized. Motivated by the fact that both an iteration of multiplication and the number of hidden nodes play key roles for a trade-off between recognition accuracy and energy consumption, these two parameters are optimized for a minimum of energy consumption of ANN, allowing acceptable recognition accuracy. The evaluation shows that the joint optimization between the iteration of multiplication and the number of hidden nodes save 70% of the energy consumption, compared with using precise computation, at the same recognition accuracy target for both handwritten and isolated spoken digit recognition. Furthermore, adopting SDIAM in training phase, the recognition accuracy is more improved, which leads to 87% and 75% lower energy consumption for handwritten and isolated spoken digit recognition.

Keywords : artificial neural network (ANN), significant-driven iterative approximate multiplier (SDIAM), handwritten digit recognition, isolated spoken digit recognition.

Contents

List of tables

List of Figures

I. Introduction

Wearable technology is receiving significant attention worldwide. Although early version of wearable watch had mostly simple functions such as step counts and reading message from smartphone, many types of wearable devices [1] such as watch, ring, shirt and glass, has been released and their applications have become diverse. Especially, recognition applications for human-machine interaction [2] have been adopted as key functions using vison sensor, audio sensor and others. For instance, fingerprint recognition is widely used for identification. Wearable ring utilizes gesture recognition so that a user writes and sends text message and also, this ring controls devices such as smartphone, TV, lights and others with just finger movement. As recognition algorithm becomes more intelligent, facial emotion recognition on wearable glass has been developed. This application is targeted for autistic patients who hardly recognize state of emotion and have difficulty in conversation. The key part that consists of these applications are various recognition algorithms such as artificial neural network (ANN) and hidden Markov model (HMM). To solve more complex problem such as facial emotion recognition, deep learning which requires a lot of computations has emerged to meet the demand for diverse functions and intelligence of application. However, as the algorithm become more intelligent its complexity and energy consumption increasingly grow in limited battery capacity. Thus, efforts to achieve high energy-efficiency is crucial in wearable devices to enable application which become more complex and energy-hungry.

One of common characteristics of the recognition algorithms to save energy is inherent algorithmic fault tolerance (AFT) [4] that some error in each computation does not lead to critical failure, but cause slight degradation for an acceptable recognition performance. Adopting these characteristics, significance-driven iterative approximate multiplier (SDIAM) [6] which

provides different degrees of computational accuracy by each iteration with different energy consumptions, was introduced. When implementing handwritten digit recognition using ANN, SDIAM at n=1 consumes 9% of the energy that precise multiplier (PM) spends and have 71.2% of computational accuracy, which shows sufficiently high recognition accuracy (95%). Regarding AFT, the number of hidden node is another key factor for optimizing ANN. This directly affects recognition accuracy and computational cost. In order to achieve high recognition accuracy, many researchers have attempted to propose a variety of methods to determine the optimal number of hidden nodes such as trial-and-error, pruning, 70% of input size or 2/3 of the sum of input and output sizes as a rule of thumb [7, 8]. However, generally, when the size of hidden node is large in an appropriate range, a great amount of multiplications are performed and the result gives high recognition accuracy. On the other hand, with the small size of hidden node a small amount of multiplication leads to low computational cost for less accurate recognition.

In this thesis, I considerately adjusts the two factors, the number of hidden node of ANN and the iteration of SDIAM. ANN is first computed by PM in the training phase and both SDIAM and PM compute the algorithm in the evaluation phase. Furthermore, SDIAM is adopted to compute the algorithm in the training phase, evaluating ANN by all multipliers. I implements ANN-based recognition to gain recognition accuracy and energy consumption of handwritten digit and isolated spoken digit recognition [9, 10] using Matlab. Also, the relationship of two trade-offs for optimization between recognition accuracy and energy efficiency is investigated to offer acceptable accuracy degradation for greater energy saving.

The rest of this thesis is organized as follows. Section 2 presents types of application on wearable devices and recognition algorithms supporting these applications. Section 3 describes an overview of ANN as one of widely used recognition algorithms and have a close look at a

parameter for energy-efficiency. SDIAM as energy-efficient hardware accelerator for ANN is presented and a method of jointly optimizing the recognition algorithm and the accelerators is explained in Section 4. In section 5, evaluation method is presented and performance is evaluated. Regarding these thesis, related works are introduced in section 6 and lastly, this thesis is concluded in section 7.

II. Recognition Application on Wearable Devices

2.1 Trend on Recognition Application

Although smart-watch gained a lot of attention when wearable device first appeared, people faced away using the device owing to various technical barriers, such as lack of applications and battery capacity. However, wearable device has become diverse like ring, shirt and glasses. Also, there has been a lot of improvements on algorithms and energy-efficiency, which enables diverse application. Especially, not only simple function such as counting steps, alarms, listening music but also applications which exploit recognition algorithms based on various sensors has been utilized, such as face recognition, speech recognition, gesture recognition and facial emotion recognition. Also, its domain is expanding, solving more complicated problems to meet user's need.

2.1.1 Vision-Based Application

The vision-based recognition [2, 3] is probably one of the widespread area. Its research area includes face recognition, gesture recognition, body movement tracking and handwritten text recognition. These vision-based recognition application has been applied to different types of area. One of the widely used application is fingerprint identification, utilized not only on wearable devices but also smart phone. In the case of google glass, it detects a person shown on the glass and related information is displayed for user's convenience. Furthermore, as algorithms based on vision sensor develops, more complex problems could have been solved. Facial emotion recognition, using deep learning which requires large amount of computations, make autism possible to recognize emotional state of people through the google glasses. Like this, problems are increasingly getting more complex. In order to solve the problems, the algorithms

are demanding much more computations. Therefore, the need for energy-efficiency energy to implement those application in limited battery capacity on wearable devices gained a lot of attention..

2.1.2 Audio-Based Application

The audio-based recognition [2] is another important application on wearable devices. This application deals with information from different audio signals. Speech recognition is the main area and it is used to convert spoken words to text, manipulate or control a device, or communicate with a machine. The voice recognition is being developed to accurately detect every person's voice with no error even in difficult environment. Speech Recognition is regarded as today's best option for human-machine interface because physical interfaces are not necessary for users. The strong point of the audio-based recognition is that voice is hands-free and also eyes-free, making it suitable for use in a variety of environments, as well as for the visually-disabled. And voice can not only deliver the meanings but also communicate mood, gender, emphasis and even personality unlike the visual-based application. Although one of the problems on voice recognition is accuracy, thanks to the development of advanced technology such as deep learning, accuracy has been rapidly improving. Recently, Hyundai has released its Blue Link, voice recognition-enabled smartwatch companion app for wearable applications. The user on cloud-based platform can use voice commands to run remote functions, while selecting the microphone icon on the watch activates the voice function to enable commands such as "Start my car", "Lock my car" or "Find my car".

2.1.3 Other Sensor-Based Application

In addition to the above two recognition, data from a variety of sensor [2, 11] is used as input for recognition. A radar-based gesture recognition, developed by Google for wearable devices, can track small movements like waving your fingers, crossing fingers, making a fist. This could enable you to enter text on a wearable device without touching. Also, human activity recognition, using wearable sensors such as accelerometer and gyroscope sensors, is a growing area with the potential to provide valuable information on patient mobility to rehabilitation specialists. Using pattern recognition techniques a computer extracts important features from your voice to recognize a word. This is a lot more complicated than it sounds, and requires many other processing techniques to improve the signal, such as noise reduction and background separation.

2.2 Recognition Algorithms for Application

Although processes to implement recognition applications are different depending on their types, the main task of recognition application consists of 4 processes [9] as shown in Figure. 1. The first step is data acquisition. As an example of handwritten digit recognition, the data is obtained by scanning the written image and converted into a form which should be acceptable to the computing devices for further processing. Second step is preprocessing. This process is an important step because of the variation in the writing style among different users and the existence of huge amount of noise in the images after scanning. Binarization is one of preprocessing processes, which convert grayscale images to binary images in order to identify the objects of interest from the image. Also, because a large amount of noise may occur in the image obtained after scanning, noise removal process is performed. In addition to these two processes, various processes such as normalization, skew correction and segmentation are

applied during preprocessing. As third step, feature extraction is the process of extracting relevant features of the handwritten digits to form feature vectors used by classifiers for the recognition. Classification is the final step of handwritten digit recognition, which is done by assigning labels to digits based on the feature extracted. The challenging job to implement those recognition application is to find out the proper algorithms because each algorithm has its own characteristics which is more applicable to specific application. However, the most widely used recognition algorithms are ANN, SVM and HMM and recently as an advanced types of recognition algorithms, deep learning has continuously received attention because of their capability to solve more complex problems.

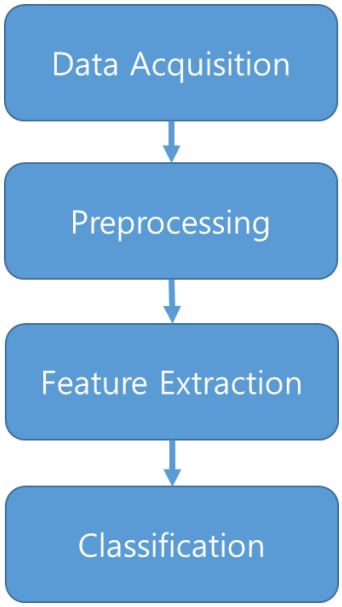

Figure. 1. Architecture of a general recognition

2.2.1 Artificial Neural Network (ANN)

ANN [12, 13] is a family of statistical learning models inspired by biological neural networks and is used to approximate functions that can depend on a large number of inputs. As systems of interconnected neurons, shown in Figure.2, ANN sends signals to each other and its connections

have weight values that can be trained based on experience. For example, ANN for handwritten digit recognition is triggered a set of input neurons which may be activated by the pixels of an input image. After being weighted and transformed by an activation function, the activations of these neurons are then delivered to other neurons. Repeatedly, this process is performed. Finally, an output neuron is activated. As results, this classifies which digit it was.

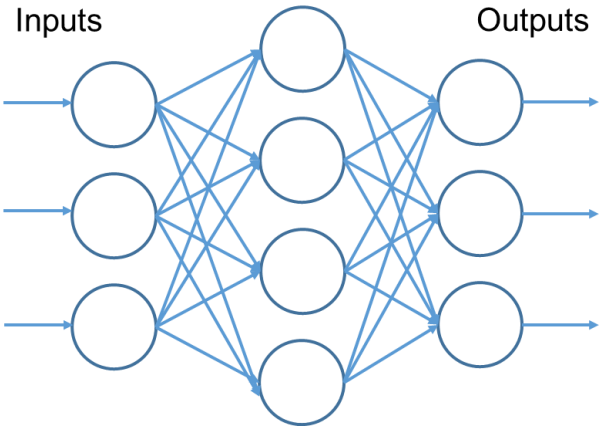

Figure. 2. Basic architecture of ANN with a single hidden layer

2.2.2 Support Vector Machine (SVM)

A SVM [14] training algorithm, a non-probabilistic binary linear classifier, is a model that divides inputs into a part or the other. A SVM model is a representation of the input as points in space in Figure. 3. A clear gap (margin) based on a separating hyper-plane, which is as wide as possible, divide the input of the separate categories. New inputs are then mapped into that same space. Based on which part of the space they are placed, the new inputs are classified to a specific category. On top of performing linear classification, SVMs can improve its performance adopting a non-linear classification using the kernel trick, mapping their inputs into high-dimensional feature spaces.

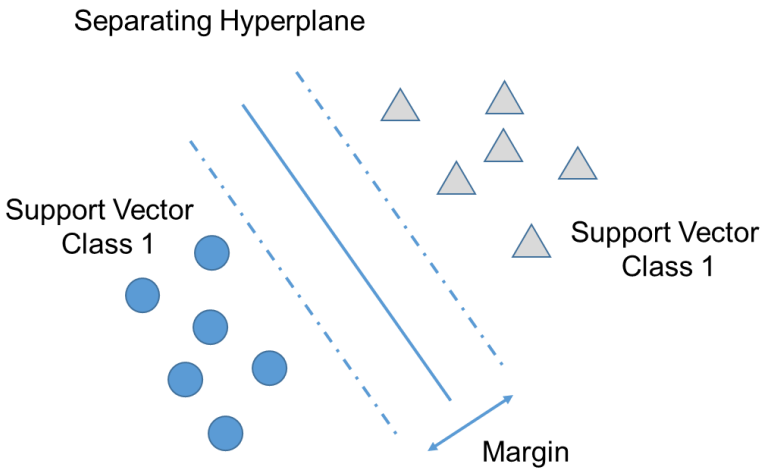

Figure. 3. The principle of SVM to classify input among 2 classes

`

2.2.3 Hidden Markov Model (HMM)

HMM [15] in Figure. 4 is a statistical Markov model in which the modeled system is assumed to be a Markov process with hidden states. In this model, the hidden states are not directly visible, but observable states which depend on the hidden states are visible. Each hidden state has a probability distribution over the possible observable states. HMM infers the most likely sequence of hidden states that produced a given observable states sequence and also infers which will be the most likely next state. Consequently, HMM calculates the probability that a given sequence of observable states originated from the system (allowing the use of hidden Markov models for sequence classification), which calculates output probability to classify the input for recognition. HMM is especially known for their applications in temporal pattern recognition such as speech, handwriting and gesture recognition.

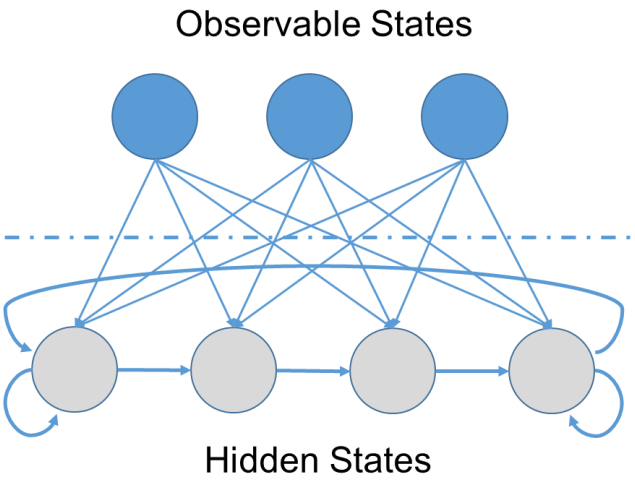

Figure. 4. The principle of HMM to select a sequence of states

2.2.4 Deep Learning

Regarding deep neural network as one of deep learning, this network is distinguished from the general neural network of a single hidden layer by its depth. The existing machine learning depends on shallow architecture, with one input and one output layer, and one hidden layer. However, deep learning [16] is composed of more than three layers. Let's take an image recognition process with deep learning as an example in Figure. 5. Computers cannot understand the meaning of a collection of pixels from an image. Thus, mappings from a collection of pixels to a complex object are complicated. However, with deep learning, the problem is broken down into a series of hierarchical mappings with each layer presenting a distinct set of features based on the previous layer's output. Each layer recombines features from the previous layer and extracts increasingly abstract features from the input, which present a hierarchy of increasing complexity and abstraction and also makes deep learning network capable of handling very large, high-dimensional data sets. The input pixels are presented at the visible layer in Figure. 5. The first hidden layer identifies the edges, which make the second layer extracts the corners and

contours. Based on the output of second layer, the parts of objects are detected. Finally, using these parts of objects, the fourth layer identifies whole objects.

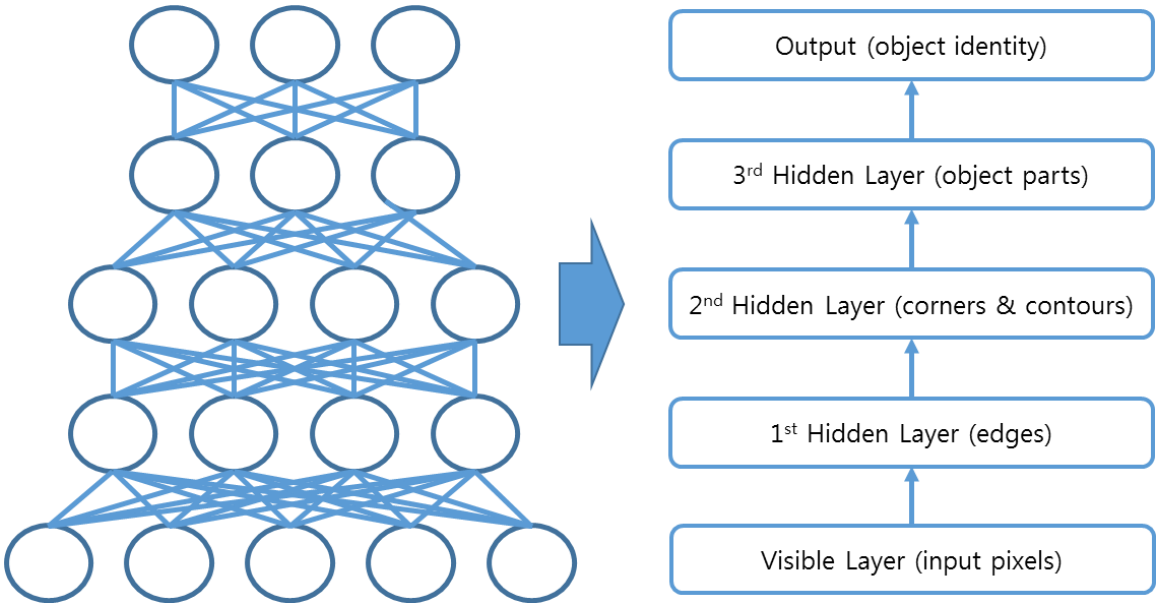

Figure. 5. A hierarchy of deep neural network and each layer's

III.Artificial Neural Network (ANN)

3.1 Introduction to ANN

The importance and use of natural HMI [2] such as handwriting, speech, and gesture recognitions, have been continuously growing. For executing such recognitions, a variety of recognitions algorithms are utilized. Although those problem can't easily be defined with a mathematical algorithm and quantified into an algorithm, these tasks are trivial to humans. One of the widely used recognition algorithms which support a variety of HMI is ANN, computing great amount of vector multiplications. The key to ANN is that their design enables them to process information in a similar way to our own biological brains, by drawing inspiration from how our own nervous system functions. This makes them useful tools for solving complex real-world problems like facial recognition, which our biological brains can do easily.

The characteristics of ANN [12, 13] is its structure massively distributed in parallel. The data processing takes place through the iteration of a great amount of computational neurons. Each neuron sends signals to other neurons in the network. Because the calculations are divided in many neurons, if any of them cause some error, it doesn't affect the behavior of the network. This leads to another important characteristics of the ANN, AFT [4]. Even when the input data exhibit variability or noise, the network classifies the data correctly. Also, when a failure takes place in any of elements of the network, it does not lead to a critical failure but to a graceful degradation in recognition accuracy.

3.2 Architecture and Feedforward Operation of ANN

A basic type of architecture is configured in the form of feedforward network adopting a single hidden layer, shown in Figure. 6, consists of M input neurons, N hidden neurons and P output neurons, having a single hidden layer. Each $\mathbf{x}$, $\mathbf{h}$ and $\mathbf{y}$ denotes the M-dimensional

external input, N-dimensional hidden layer and P dimensional output vector; and $\mathbf{W}_1$ and $\mathbf{W}_2$ the connection weight matrixes of size $M \times N$ and $N \times P$ where $w_{i,j}$ and $w_{j,k}$ are the weights between x_i and h_j and between h_j and y_k. Then, the output $\mathbf{y}$ is given by $\mathbf{y} = f\,(\,\mathbf{W_2} \times (\,f\,(\mathbf{W_1} \times \mathbf{x} + \mathbf{b^1}) + \mathbf{b^2}\,)$, where f is a nonlinear function such as a sigmoid, Heaviside, or Gaussian function, and $\boldsymbol{b_1}$ and $\boldsymbol{b_2}$ are constant biases. Consequently, the network is comprised of a number of matrix vector multiplications and instance vectors (i.e., weight values) that rarely changes in order to execute a variety of recognition.

Once trained, the neural network can be applied to classify new data. Individual nodes in an input layer take the new input data and perform feedforward operation based on its architecture, passing the results on to the next layer after performing activation functions. Finally, the pattern of activation of the network at output nodes determines the results of classification of input data. By doing this, the neural network perform a recognition such as facial recognition and speech recognition.

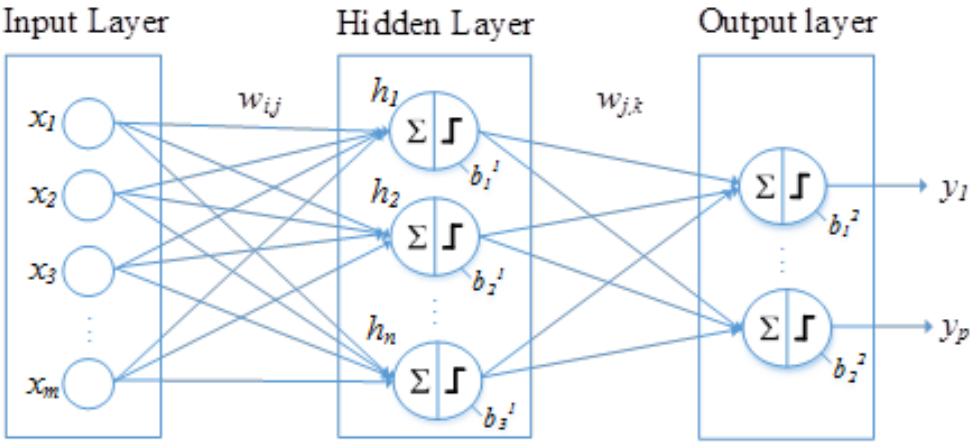

Figure. 6. Architecture of ANN with a single hidden layer

3.3 Learning Algorithm

In general, learning of recognition algorithms is performed through the process of updating the internal representation of the system. This contains modifying the network architecture, which involves adjusting the weights of the links and others. There are many different learning algorithms but the most widely used is backpropagation [12, 13]. To train the neural network for executing a recognition, Backpropagation begins with random weights. Data from inputs is fed forward through the network to optimize the weights between nodes. During training phase the weights are optimized by backward propagation of the error. Reading the input and output values in the training data set, the ANN changes the value of the weighted links to reduce the error (the difference between the predicted and target values). The error in prediction is minimized through a large number of iteration until network reaches specified level of accuracy. If a network is left to train for too long, however, it will over-train and will lose the ability to generalize.

3.4 A demand for Selecting the Optimal Number of Hidden Nodes

The performance of ANN depends strongly on the network architecture, especially the number of hidden nodes. Thus, determining the optimal number of hidden nodes is significant. Selecting too large size of hidden layer results in low training errors but still high generalization errors due to over-fitting, while too small size of hidden layer causes high training errors and high generalization errors due to under-fitting. Many researchers have proposed some rules of thumb for determining an optimal number of hidden nodes [7, 8], the size of the hidden layer are somewhere between the input layer size and the output layer size, the size of the hidden layer should never be more than twice as large as the input layer. However, raising the number of hidden nodes up to the optimal does not only guarantees the high recognition accuracy but also demand a great amount of computation, which requires much energy. Therefore, it is crucial to

select the appropriate number of hidden node minimizing energy consumption and offering just enough recognition accuracy at the same time for a given application.

IV. Energy-Efficient Hardware Accelerator for ANN

4.1 Characteristics of ANN to apply SDIAM

In this thesis, one of the widely used recognition algorithm, ANN, to support various recognition applications has been discussed and, regarding energy-efficiency due to limited battery capacity, recognition algorithms like many other DSP algorithms exhibit AFT [4, 5]. While computing the algorithm, some error is inevitably contained in the computing process, which leads to a graceful degradation in recognition accuracy. Moreover, although algorithms are generally optimized utilizing floating-point arithmetic, these algorithms are converted to fixed-point arithmetic because the delay, area, and energy consumption of fixed-point arithmetic are much less than those of floating-point arithmetic and also the loss of accuracy is negligible after the conversion from floating-point to fixed-point. By exploiting this characteristic, AFT, the previous section explained how the algorithm parameter significantly impact the energy consumption as a software approach. To be specific, ANN [7, 8] with more hidden nodes, requiring more computations, can tolerate lower computational accuracy than ANN with fewer nodes, requiring fewer computations for the same recognition accuracy. Though the number of hidden nodes gives a trade-off between recognition accuracy and the amount of multiplication, an approximate multiplier, SDIAM, adopted in this paper provides another parameter that exhibits how much reduced computational accuracy a recognition algorithm can tolerate for a target accuracy of a recognition. In other words, SDIAM gives a trade-off between computational accuracy and energy-consumption by adjusting the iteration of multiplication.

4.2 Significance-Driven Iterative Approximate Multiplier (SDIAM)

4.2.1 Architecture and Operation of SDIAM

Figure. 7 shows the block diagram of the proposed SDIAM [6]. *A*, *B*, S_i, and Z_i, denote the multiplier, multiplicand, shift amount for multiplicand *B* for i^{th} iteration, and output after i^{th} iteration. One of the operands for multiplications (trained weight values) becomes constant value after the completion of training process in recognition algorithms. Since the trained weight values are determined before an evaluation phase is performed, SDIAM processes these coefficients (*A* values) to gain the shift amounts for *B* values in advance and store them in on-chip memory, instead of the actual *A* values. *PPA*[*k*][*n:1*] in Figure. 7 denotes a preprocessed constant coefficient where each stores *N* shift amounts in the order of significance. *PPA*[*k*][*i*] represents the shift amount k^{th} coefficient for i^{th} iteration. To compute a multiplication, a coefficient is read from the memory *PPA* and delivered to the *log2*(*K*)-bit shift register where *K* denotes the bit width of the multiplier. In each iteration, the shift register will provide a necessary amount for shifting *B*. The shift amount for each iteration can be determined by *log2*(*K*) bits.

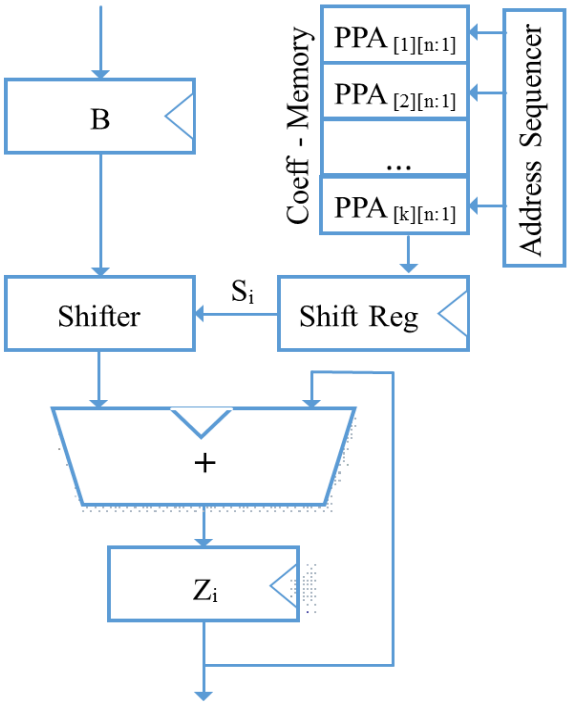

Figure. 7. The block diagram of SDIAM's architecture

Figure. 8 shows a process of an 8-bit multiplication with A = 01011010_2 (9010) and B = 01111101_2 (12510). Assume that the pre-processed A value is prepared in PPA[1] (= {S_1 = 110_2 (610), S_2 = 100_2 (410)}). B is shifted by S_1 (=610) and added to zero-initialized Z_0 during the first iteration. The result value of the first iteration is 0001011010000000_2 (567010) with 71% accuracy (i.e., 800010 / (9010×12510 = 1125010)). The result of the second iteration with S_2 (= 410) through the same process of the first iteration leads to 1000010 (800010 + 200010), achieving 89% accuracy. 98% and ~100% accuracy are obtained after the third and fourth iterations. In other words, more iterations give higher accuracy, sacrificing the cost of energy consumption. Thus, this approach doesn't gain high accuracy when the number of iterations (n) is 1 or 2. In particular, while limiting n to 1, the multiplication using the SDIAM gives 0001011010000000_2 (576010). This achieves only 51% accuracy, which means losing much accuracy with n = 1 in this particular example. This is because we cannot consider the four consecutive 1s after the first leading 1 bit. These ones represent a substantial percentage of the magnitude of the multiplier value (26% and 13% for the first and second consecutive 1s after the first leading 1, respectively).

Iteration 1:
$A = 0\underline{\mathbf{1}}011010_2$ $B = 01111101_2$
$S_1 = 110_2$
$Z_1 = 0000000000000000_2$
$+ 00\underline{\mathbf{01111101}}000000_2$

0001111101000000_2

Iteration 2:
$A = 010\underline{\mathbf{1}}1010_2$ $B = 01111101_2$
$S_2 = 100_2$
$Z_2 = 0001111101000000_2$
$+ 0000\underline{\mathbf{01111101}}0000_2$

0010011100010000_2

Figure. 8. A multiplication example of SDIAM

4.2.2 Recognition Accuracy and Energy Consumption

Table 1 shows each computational accuracy with varying n and also normalized energy consumption which means a portion of energy consumption of a 32-bit SDIAM on that of PM for each iteration (n). The computational accuracy considerably rises with increasing the iteration for 100K pairs of randomly generated operands. The normalized energy consumption also proportionally increases by the number of iteration.

Table 1.Computational accuracy and energy consumption varying *n* for SDIAM

	n=1	n=2	n=3	n=4
Computational Accuracy	71.2%	90.8%	97.1%	99.1%
Normalized Energy Consumption	9%	21%	32%	42%

4.3 Joint optimization of N and n

As discussed earlier, larger *N* generally can lead to higher recognition accuracy at the expense of more multiplications (more energy consumption) or tolerate less accurate computations. Thus, we see an opportunity of jointly optimizing *N* and *n* to minimize the energy consumption of ANN accelerators for a recognition accuracy target. Consider that the total energy consumption of ANN is dominated by multiplications. Figure. 9 shows the energy consumption per multiplication as a function of computational accuracy, *N* to achieve a certain recognition accuracy as a function of computational accuracy and also the total energy consumption of multiplications performed to evaluate ANN for a given input. This shows that judiciously choosing *n* and *N* can significantly reduce the energy consumption while achieving the same recognition accuracy. Using a PM or SDIAM with large *n* does not significantly reduce *N* (the

number of multiplications) compared to SDIAM with small *n*. In contrast, using SDIAM with very small requires notably large *N* to achieve the same recognition accuracy. This negates the benefit of low energy per multiplication since the total energy consumption is proportional to the product of *N* and energy per multiplication.

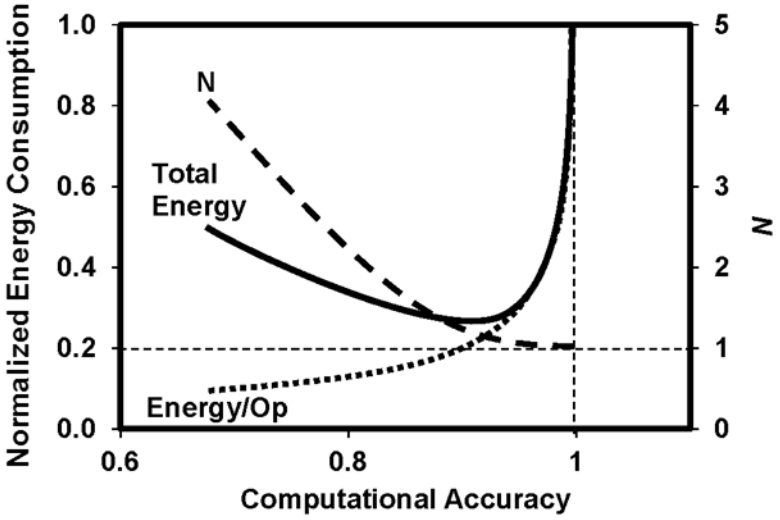

Figure. 9. Total energy consumptions as a function of computational accuracy and

4.4 Training ANN with SDIAM

In the previous studies [6], SDIAM is applied only to the evaluation phase while it is implied that PM are used for the training phase. Note that the training phase is to adjust the weight values of ANN such that it minimizes the error between the intended output and the output computed by ANN for a given input. When SDIAM is used only for the evaluation phase, the difference of values between SDIAM and PM for a given input is manifested as error at the output of ANN. This leads to a wrong recognition if the output error is too large. In this paper, we propose to utilize the same SDIAM for both the training and evaluation phases, where we hypothesize that any computational inaccuracy of SDIAM relative to PM is no longer recognized or manifested as

error during the evaluation phase. This is because the trained weight values are already determined to take the computational inaccuracy of SDIAM into account.

V. Performance Evaluation

5.1 Evaluation Methodology

ANN-based handwritten digit recognition and isolated spoken digit recognition have been implemented, using Matlab and its mex API to interface with SDIAM written in C language. As a design configuration, a single hidden layer is chosen using backpropagation as a training algorithm. PM and SDIAM with n=1, 2, 3, and 4 are applied to each algorithm. 60000 samples from the MNIST database [17] and 30000 samples from TI 46 words speech database [18] are applied to train an ANN for a random seed. Then, this training phase is repeated with 20 different random seeds to generate 20 ANNs for each N. To evaluate the recognition accuracy and energy consumption for each pair of N and n, 10000 samples from the MNIST database and 4100 samples from TI 46 words speech database have been applied to each ANN. The average recognition accuracy of 20 ANNs trained with 20 different seeds has been reported. For energy evaluation, an amount of multiplication of each multiplier is counted to evaluate a recognition using an ANN. As multiplying the amount of multiplication with the normalized energy consumption [6] in comparison to PM, energy-consumption has been evaluated.

5.2 Evaluation

The average recognition accuracy of 20 ANNs trained with 20 different seeds for handwritten digit recognition and isolated-spoken digit recognition are shown in Figure. 10a and Figure. 10c. For SDIAM with n=2,3,4 which have more than 90% of computational accuracy, each recognition accuracy nearly follows the same path of the accuracy of PM with almost no error in both recognition while the recognition accuracy of the spoken digit recognition at n=2 follows with a small gap. However, when n=1 having 71.2% of computational accuracy, the both

recognition accuracy shows notable gaps with others and fluctuates a lot for both recognitions, though the accuracy is smoothed by averaging. Figure. 1a shows that using SDIAM with n=1, 2, 3, and 4 for multiplications requires N=70, 30, 20, 20 to satisfy the minimum recognition accuracy of 95% for handwritten digit recognition demonstrating the trade-off between n and N. Regarding isolated spoken digit recognition, though SDIAM with n=1 doesn't satisfy the minimum recognition accuracy of 75%, Figure. 10c exhibits N= 45, 25, 20 to meet the minimum recognition accuracy of 75% using SDIAM with n=2, 3, and 4. When using SDIAM for both training and evaluating, the recognition accuracy using SDIAM with n=2, 3, 4 in AM for both recognitions in Figure. 10b and Figure. 10d doesn't show much difference with the averaged accuracy shown in Figure. 10a and Figure. 10c. However, the recognition accuracy, especially when n=1 having the lowest computational accuracy, becomes much higher and doesn't show as much fluctuation as the first ones do. More importantly, for the minimum recognition accuracy of 95% in handwritten digit recognition, N is reduced from 70 and 30 to 30 and 20 when n is 1 and 2, respectively, demonstrating the effectiveness of the proposed training method. In isolated spoken digit recognition when n=2 and 3, the minimum recognition accuracy 75% was satisfied by decreasing N from 45 and 25 to 20 and 17, which means reducing the amount of computations.

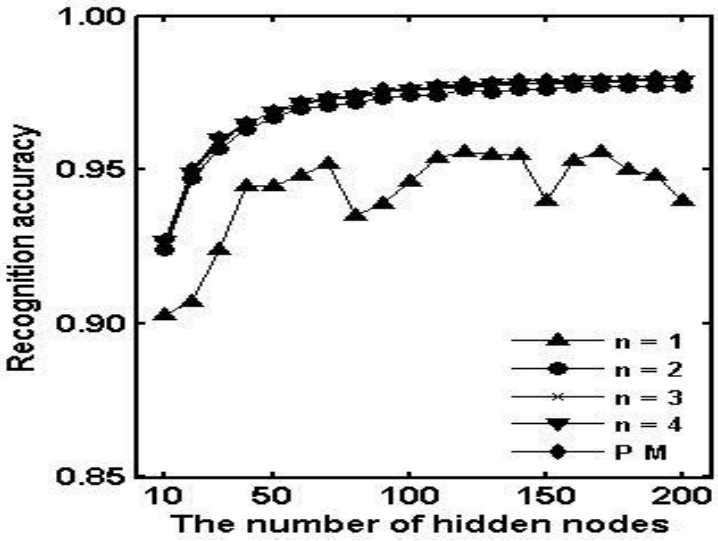

Figure. 10a. Handwritten digit recognition accuracy versus N and n; trained with PM.

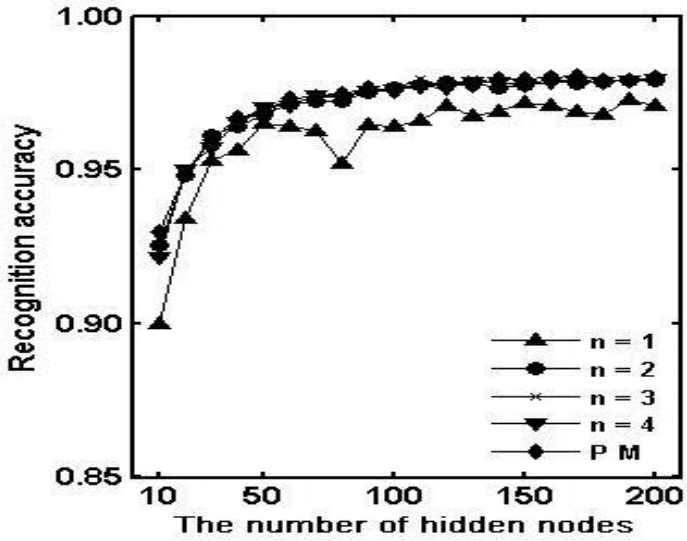

Figure. 10b. Handwritten digit recognition accuracy versus N and n; trained with SDIAM

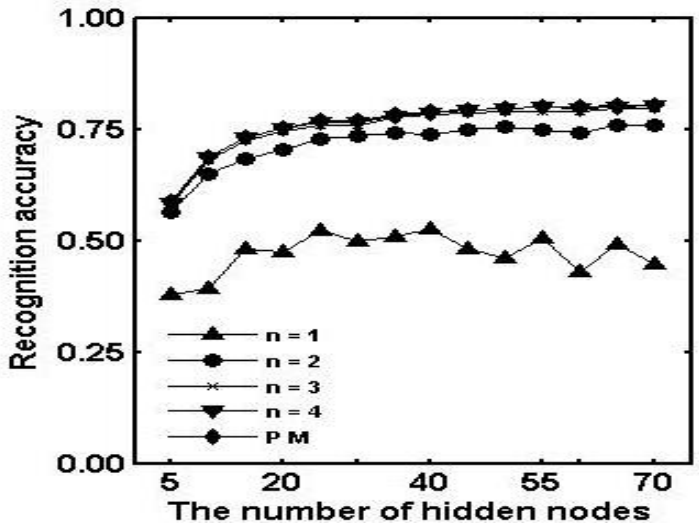

Figure. 10c. Isolated-spoken digit recognition accuracy versus N and n; trained with PM.

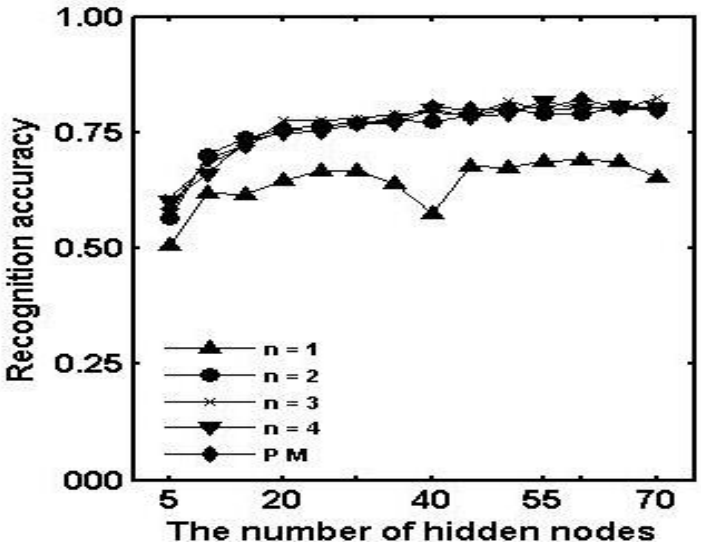

Figure. 10d. Isolated-spoken digit recognition accuracy versus N and n; trained with SDIAM.

Figure. 11a and Figure. 11b plot the normalized energy consumption of multiplications when SDIAM and PM are utilized for ANNs for handwritten digit recognition, targeting the minimum recognition accuracy of 95% and 97%. When the ANNs are trained using PM, SDIAM with n = 2 gives the minimum energy consumption for multiplications, consuming 70% and 75% less energy than PM for targeting the minimum recognition accuracy of 95% and 97%, respectively. Although the normalized energy consumption drops from PM to SDIAM with n=2, SDIAM with n = 1 consumes more energy than n = 2. This is because it requires significantly larger N to achieve the recognition accuracy target. However, SDIAM with n = 1 cannot satisfy the minimum recognition accuracy of 97% even with the maximum N (= 200) and SDIAM with n=2 exhibits the minimum energy consumption. Thus the value of SDIAM with n=1 is not displayed in Figure. 11a. When SDIAM is used for both training and evaluating the ANNs, Figure. 11b exhibits the similar trends until SDIAM with n=2 that the normalized energy consumption decrease from PM to SDIAM with n=1. However, SDIAM with n = 1 gives the minimum energy consumption for multiplications, consuming 87% and 78% lower energy than using PM for targeting the minimum recognition accuracy of 95% and 97%, respectively.

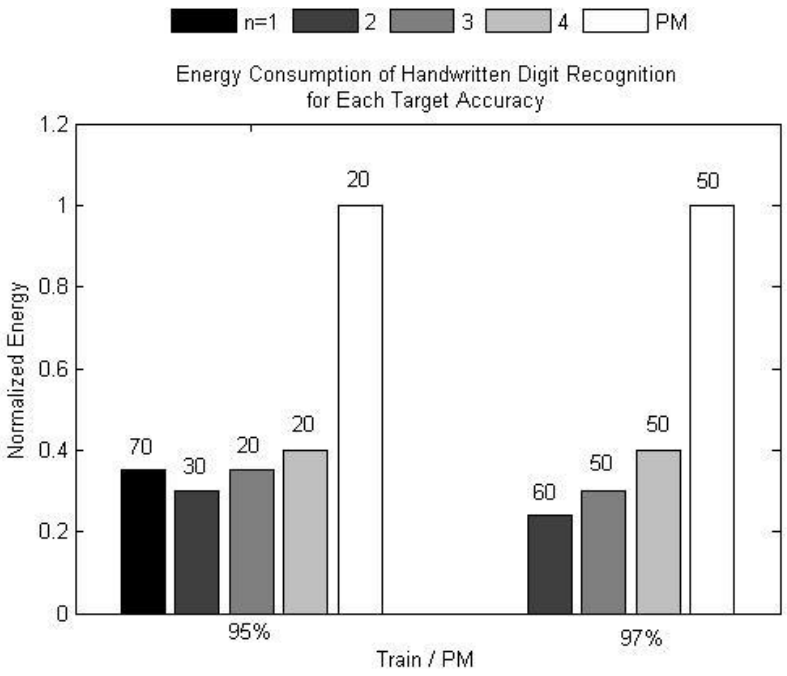

Figure. 11a. Energy consumption of multiplication for handwritten digit recognition: 95% and 97% of target accuracy, trained with PM. The number atop each bar represents required N to achieve the target recognition accuracy.

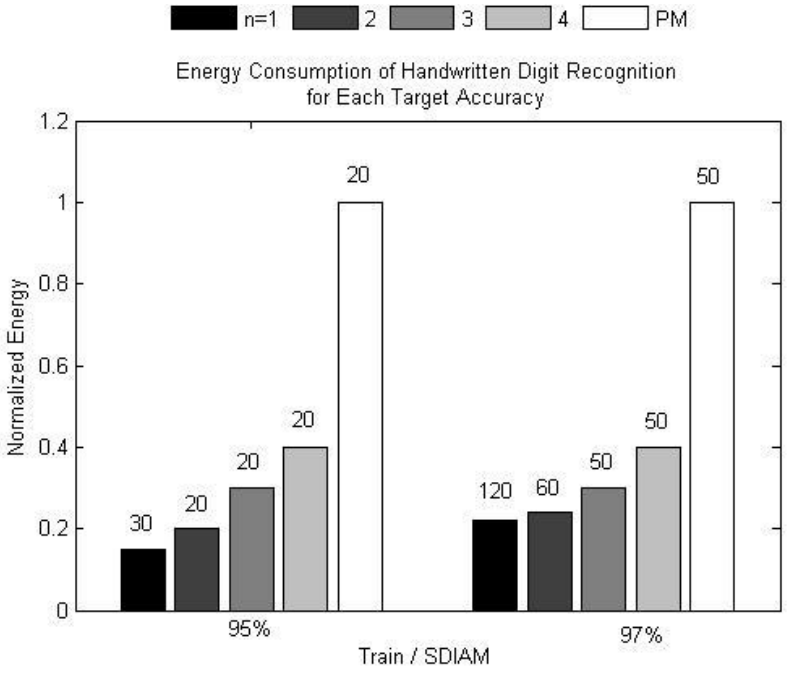

Figure. 11b. Energy consumption of multiplication for handwritten digit recognition: 95% and 97% of target accuracy, trained with SDIAM.

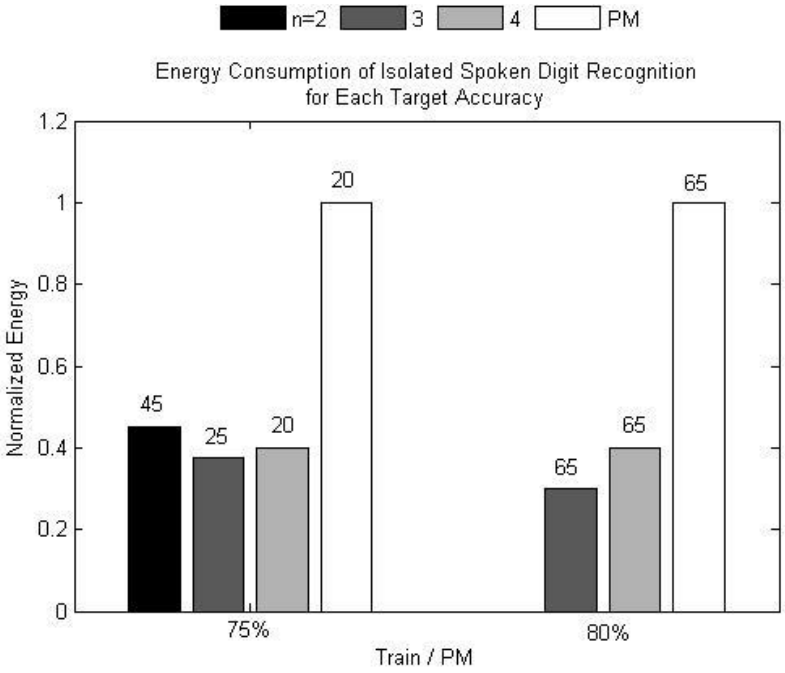

Figure. 11c. Energy consumption of multiplication for isolated-spoken digit recognition; 75% and 80% of target accuracy, trained with PM.

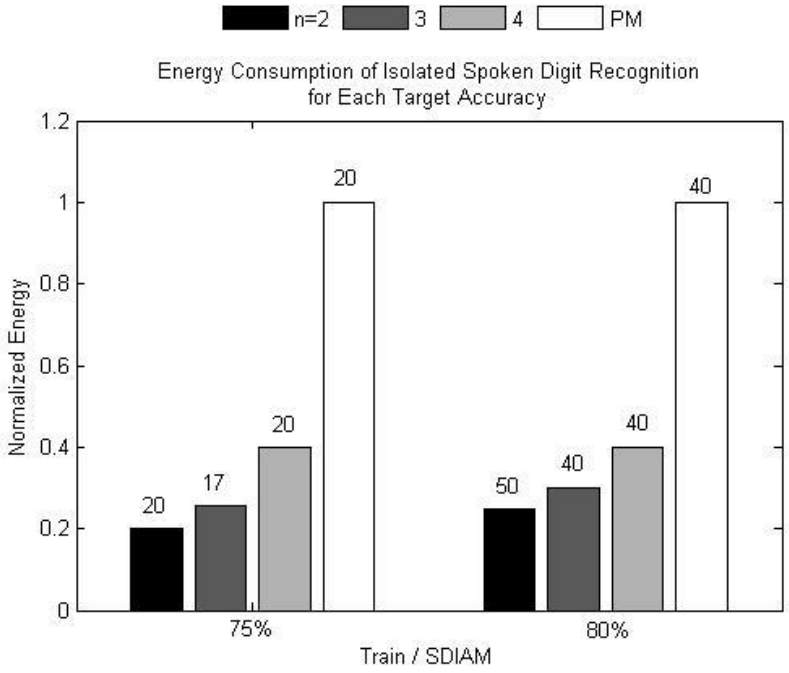

Figure. 11d. Energy consumption of multiplication for isolated-spoken digit recognition; 75% and 80% of target accuracy, trained with SDIAM.

Regarding the isolated spoken digit recognition, Figure. 11c and Figure. 11d display the normalized energy consumption of multiplications using both SDIAM and PM, targeting relatively lower recognition accuracy of 75% and 80%. Also, when SDIAM with n=1 is utilized for training, the target recognition accuracy of 75% is not satisfied even with raising the number of hidden nodes of the ANNs. Thus, I haven't displayed them in the plots. Figure. 11c shows that SDIAM with n=3 exhibits the minimum energy consumptions for each target recognition accuracy, consuming 63% and 70% less than PM, respectively. When using SDAIM for both training and evaluating the ANNs, SDIAM with n=2 gives the minimum energy consumption for multiplications, consuming 80% and 75% lower energy than using PM for the target accuracy of 75% and 80%, respectively.

VI. Related Works

As wearable and mobile devices evolve, a variety of application and function has been embedded in these devices and requires functions to solve problems with a high level of difficulty. To meet the consumer's needs, a large number of recognition algorithm has been developed and utilized to solve given problems. However, the fact that a small size device, a wearable device, does not sufficiently provide computational capacity to implement a variety of applications, which facilitated researchers to devise energy-efficient hardware technologies.

Many recognition algorithms and digital signal processing contain AFT [4] [5], which indicates that some error in each computation does not lead to a critical failure but to a graceful degradation in overall algorithmic accuracy. Exploiting this characteristic, scalable hardware with scaling to minimize energy consumption, offering acceptable classification quality for handwritten digit recognitions, was proposed by *Chippa et al* [19]. *Kim et al* proposed SDIAM which supports circuit-level scaling through iterations with different accuracy and energy consumption [6]. They exploited another characteristics of recognition algorithm, fixed coefficient values. SDIAM preprocess the values to avoid repetitive unnecessary computation.

SDIAM motivated me to do further research, analyzing the characteristic of algorithm for energy-efficiency. When designing ANN, one of the most important element to be considered is the number of hidden nodes, which directly affects recognition accuracy and the amount of computations, providing a trade-off. In order to achieve high recognition accuracy, many researchers attempted to propose a variety of methods to determine the optimal number of hidden nodes such as trial-and-error, pruning, 70% of input size as a rule of thumb [7, 8]. However, generally, when the size of hidden nodes is large, a great amount of multiplications are performed and the result gives high recognition accuracy. On the other hand, with the small size

of hidden node a small amount of multiplication leads to low computational cost for less accurate recognition. Thus, this thesis has focused on integrating and optimizing these two trade-offs for energy-efficiency, the number of hidden nodes and the iteration of SDIAM.

VII. Conclusion

A variety of recognition applications are run on wearable devices and the algorithm are becoming more complex, which consumes a lot of energy in a limited battery capacity. In this thesis, optimizing methods is proposed to minimize the total energy consumption of specialized hardware for recognition algorithm while providing sufficient recognition accuracy, where multiplications dominate the total energy consumption. Exploiting SDIAM and ANN for handwritten digit recognition and isolated spoken digit recognition, the evaluation has been performed. The number of hidden nodes (N) of ANN as an algorithm parameter is used, giving a trade-off between the amount of computations and the recognition accuracy. Also, adopting SDIAM, the number of iterations (n) is adjusted as a knob to trade computational accuracy with energy consumption per multiplication. When SDIAM is utilized to train ANNs in order to satisfy the minimum recognition accuracy of 95% in handwritten digit recognition, N is reduced from 70 and 30 to 30 and 20 when n is 1 and 2, respectively, demonstrating the effectiveness of the proposed training method, which means reducing the amount of computations. Isolated spoken digit recognition follows the similar result of handwritten digit recognition. Also, choosing 30 and 2 for N and n gives the minimum energy consumption (70% lower energy consumption than PM) to satisfy 95% recognition accuracy for handwritten digit recognition. For isolated spoken digit recognition, the minimum energy is consumed at 65 and 3 for N and n, targeting 80% recognition accuracy (70% less energy consumption than PM). However, when adopting SDIAM for the training phase, the optimal pair of N and n changes to 30 and 1 for handwritten digit recognition which consumes 87% lower energy than PM. For isolated spoken digit recognition, 75% lower energy is consumed as the minimum at 50 and 2 for N and n. Due to the fact that two elements, the number of hidden nodes and the iteration of AM are tunable,

user can select those to meet the desired recognition accuracy and also the least energy consumption.

References

1. Ye, Hanlu, et al. "Current and future mobile and wearable device use by people with visual impairments." Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, 2014.
2. Cannan, James, and Huosheng Hu. "Human-Machine Interaction (HMI): A Survey." University of (2011).
3. Mandal, Bappaditya, et al. "A wearable face recognition system on google glass for assisting social interactions." Computer Vision-ACCV 2014 Workshops. Springer International Publishing, 2014.
4. Protzel, Peter W., Daniel L. Palumbo, and Michael K. Arras. Fault tolerance of artificial neural networks with applications in critical systems. Vol. 3187. National Aeronautics and Space Administration, Office of Management, Scientific and Technical Information Program, 1992.
5. Hegde, Rajamohana, and Naresh R. Shanbhag. "Energy-efficient signal processing via algorithmic noise-tolerance." Proceedings of the 1999 international symposium on Low power electronics and design. ACM, 1999.
6. Kim, Nam Sung, et al. "Multiplier supporting accuracy and energy trade-offs for recognition applications." Electronics Letters 50.7 (2014): 512-514.
7. Gao, Pengyi, Chuanbo Chen, and Sheng Qin. "An optimization method of hidden nodes for neural network." Education Technology and Computer Science (ETCS), 2010 Second International Workshop on. Vol. 2. IEEE, 2010.
8. Xu, Shuxiang, and Ling Chen. "A novel approach for determining the optimal number of hidden layer neurons for FNN's and its application in data mining." (2008).
9. Chacko, Anitha Mary MO, and P. M. Dhanya. "Handwritten Character Recognition In Malayalam Scripts-A Review." arXiv preprint arXiv:1402.2188 (2014).
10. Chapaneri, Santosh V., and Deepak J. Jayaswal. "Efficient Speech Recognition System for Isolated Digits." Intl. Journal Computer Science and Engineering Technologies 4.3 (2013): 228-236.

11. Capela, Nicole A., Edward D. Lemaire, and Natalie Baddour. "Feature Selection for Wearable Smartphone-Based Human Activity Recognition with Able bodied, Elderly, and Stroke Patients." (2015): e0124414.
12. M Hassoun, Fundamentals of artificial Neural Networks.: A Bradford Book, 2003.
13. S. Marsland, Machine Learning: An Algorithmic Perspective.: CRC Press, 2009.
14. N. Cristianini and J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-based Learning Methods.: Cambridge University Press, 2000.
15. R. Duda, P. Hart, Pattern Classification.: Wiley-Interscience, 2000.
16. Yu, Dong, and Li Deng. "Deep learning and its applications to signal and information processing [exploratory dsp]." Signal Processing Magazine, IEEE 28.1 (2011): 145-154.
17. The MNIST database of handwritten digits. [Online]. http://yann.lecun.com/exdb/mnist/
18. TI 46 Words Speech Database. [Online]. http://www.ldc.upenn.edu/Catalog/docs/LDC93S9/ti46.readme.html
19. V.K. Chippa, D. Mohapatra, A. Raghunathan, K. Roy, and S.T. Chakradhar, "Scalable effort hardware design: Exploiting algorithmic resilience for energy efficiency," in IEEE/ACM Design Automation Conf., 2010, pp. 555-560.

요 약 문

에너지 효율적 인식알고리즘을 위한 알고리즘 파라매터와 계산의 정확도의 최적화

문자인식, 음성인식과 같은 사람-기계간의 상호작용을 위한 기술은 웨어러블 디바이스에서 매우 중요하다. 이러한 인식기술을 위해 다양한 인식알고리즘이 개발되어 왔고 많은 연구자들에 의해 진보하고 있다. 하지만, 더 복잡한 문제들을 더 정확하게 수행함으로 디바이스는 알고리즘을 수행하기 위해 많은 계산과 에너지를 필요로 하고 있다. 본 논문에서는 많이 사용되는 인식기술인 인공신경망을 사용하고 그의 특성인 알고리즘 결함 허용을 이용하고 에너지 효율적 곱셈기인 SDIAM 의 파라매터를 통해 에너지소비를 줄인다. 인식알고리즘을 연산하기 위해 SDIAM 이 사용되었다. 알고리즘에서는 은닉노드의 수, 그리고 곱셈기에서는 Iteration 의 수를 가변하며 충분한 정확도를 갖는 최소 에너지 소비 지점을 찾기 위한 실험을 진행하였다. 본 논문의 평가결과는 계산정확도와 알고리즘 파라매터의 최적화를 통해서 기존의 정확한 곱셈기와 비교하여 문자인식과 음성인식에 있어 동일한 70%의 에너지를 줄였다. 더 나아가, SDIAM 을 사용하여 학습단계에서 알고리즘을 연산했을 때 인식의 정확도는 더 향상되었고 에너지 소비를 더욱 줄일 수 있었다(87%-문자인식, 75%-음성인식).

핵심어 : 인공신경망, 곱셈기, 최적화, 문자인식, 음성인식