electronics

MDPI

Article

VLSI Implementation of Restricted Coulomb Energy Neural Network with Improved Learning Scheme

Jaechan Cho [1], Yongchul Jung [1], Seongjoo Lee [2] and Yunho Jung [1,*]

[1] School of Electronics and Information Engineering, Korea Aerospace University, Goyang-si 10540, Korea; jccho@kau.kr (J.C.); ycjung@kau.kr (Y.J.)

[2] Department of Information and Communication Engineering, Sejong University, Seoul 143-747, Korea; seongjoo@sejong.ac.kr

* Correspondence: yjung@kau.ac.kr; Tel.: +82-2-300-0133

Received: 22 April 2019; Accepted: 17 May 2019; Published: 22 May 2019

Abstract: This paper proposes a restricted coulomb energy neural network (RCE-NN) with an improved learning algorithm and presents the hardware architecture design and VLSI implementation results. The learning algorithm of the existing RCE-NN applies an inefficient radius adjustment, such as learning all neurons at the same radius or reducing the radius excessively in the learning process. Moreover, since the reliability of eliminating unnecessary neurons is estimated without considering the activation region of each neuron, it is inaccurate and leaves unnecessary neurons extant. To overcome this problem, the proposed learning algorithm divides each neuron region in the learning process and measures the reliability with different factors for each region. In addition, it applies a process of gradual radius reduction by a pre-defined reduction rate. In performance evaluations using two datasets, RCE-NN with the proposed learning algorithm showed high recognition accuracy with fewer neurons compared to existing RCE-NNs. The proposed RCE-NN processor was implemented with 197.8K logic gates in 0.535 mm^2 using a 55 nm CMOS process and operated at the clock frequency of 150 MHz.

Keywords: artificial neural network (ANN); machine learning; pattern recognition; restricted coulomb energy neural network (RCE-NN); VLSI

1. Introduction

Artificial intelligence (AI) has been widely used to optimize data-driven approaches in fields such as computer vision, speech recognition, robotics and medical applications [1,2]. Deep neural networks (DNNs), also referred to as deep learning, are a part of the broad field of AI, and deliver state-of-the-art accuracy on many AI tasks [3,4]. However, to complete the tasks with higher accuracy, DNN models become deeper, i.e., the number of layers range from five to more than a thousand. Training large scale DNNs usually requires adjusting a large number of parameters, which is computationally complex. Moreover, the learning algorithms of the DNNs require complex optimization techniques, such as stochastic gradient descent (SGD) and adaptive moment estimation (ADAM) [5]. These learning algorithms take huge computing resources and can take several days depending on the size of the dataset and the number of layers in the network. In addition, the structure of the network, i.e., the number of neurons and number of layers, should be set before learning [6], and the optimum structure depends on applications and data characteristics. In other words, if a network is optimized for a specific application, it needs to be structurally changed and re-learned to use it in other applications. Therefore, DNNs cannot support real-time learning and are infeasible for embedded systems with various sensor applications because of their complexity and inflexibility.

In contrast, the restricted coulomb energy neural network (RCE-NN) can actively modify the network structure because it generates new neuron only when necessary. Therefore, it can support various sensor applications and has recently been implemented for various embedded systems [7–11]. The RCE-NN efficiently classifies feature distributions by constructing hyperspherical neurons with radii and hypersphere centers. Since the learning scheme of the RCE-NN is based on the distance between the input feature and the stored hypersphere center, it is relatively simple compared with learning algorithms of DNNs and real-time learning is possible. After completing the learning process, if the calculated distance is less than a neuron's radius, the neuron is activated and the label of the neuron with the minimum distance among the activated neurons becomes a recognition result.

In the initial RCE-NN proposed in [12,13], called traditional RCE (TR-RCE) in this paper, all neurons are learned at the same radius. This method causes confusion among some areas of the feature space, which degrades recognition accuracy. In [14], RCE-NN with a dynamic decay adjustment (DDA) algorithm was proposed, which adjusted the radius of each neuron depending on the uncertainty of activated neurons in the learning process. This technique increases the recognition accuracy in areas of conflict. However, since the radius may be excessively reduced in the learning process, unnecessary neurons are generated, which increases the network complexity.

The RCE-NN proposed in [15] estimates the reliability of neurons by counting the number of activations for each neuron. In recognition, this activation count is weighted to the output of each neuron to reflect the reliability of each neuron. Although this method improves recognition accuracy, an optimal hyperspherical classifier cannot be generated in the feature space, because all neurons are learned at the same radius, as in the TR-RCE learning method. To solve this problem, an RCE-NN with a hierarchical prototype learning (HPL) algorithm was proposed in [16,17], which reduced the learning radius for each iteration of learning. The HPL also eliminates unnecessary learned neurons based on estimated reliability in the learning process. However, since the HPL-based RCE-NN estimates reliability without considering the region of the activated neurons, some unnecessary neurons remained. In addition, various hyperspherical classifiers like DDA-RCE cannot be generated because neurons are learned with the same radius in a specific iteration period.

In this paper, an efficient learning algorithm for RCE-NN is proposed: (1) The reliability of each neuron is estimated by considering the activation region with different factors; (2) The radius is gradually reduced at a pre-defined reduction rate to prevent the generation of unnecessary neurons. The design and implementation results of the RCE-NN processor for real-time processing are also presented. The remainder of this paper is organized as follows: Section 2 briefly reviews the RCE-NN. Section 3 describes the proposed learning algorithm and its performance evaluation results. Section 4 describes the hardware architecture of the proposed RCE-NN processor. Section 5 discusses its implementation results. Finally, Section 6 concludes the paper.

2. Restricted Coulomb Energy Neural Network

The RCE-NN consists of an input layer, a prototype layer (hidden layer) and an output layer. The input layer comprises feature vectors and all feature vectors are connected to each neuron of the prototype layer. The prototype layer is the most essential part of the RCE-NN. Neurons in this layer save hypersphere centers and radii, which construct hyperspherical classifiers in the feature space. The output layer uses the neuron's response to output the label value of the neuron that best matches the input feature vector.

Each neuron $\mathbf{p}_j$ in the prototype layer contains the information as follows:

$$\mathbf{p}_j = [c_j^1, c_j^2, \ldots, c_j^k, r_j, l_j], \tag{1}$$

where $j \in \{1, 2, 3, \ldots, n\}$ is the neuron index and the total number n varies according to the learning results. That is, if n neurons are learned after learning is completed, they are defined as a set of neurons

$\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n]$. Each neuron $\mathbf{p}_j$ contains a hypersphere center $\mathbf{c}_j = [c_j^1, c_j^2, \ldots, c_j^k]$, radius r_j, and learned label l_j, where k is the number of features in the input feature vector used in the learning.

If the number of input feature vectors during the learning process is m, the input feature vector set can be represented by $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_m]$, where the feature vector $\mathbf{x}_i$ consists of k features and a label $l_{\mathbf{x}_i}$. The feature vector $\mathbf{x}_i$ is entered to each neuron and the distance between the feature vector $\mathbf{x}_i$ and the hypersphere center $\mathbf{c}_j$ is computed as follows:

$$d(\mathbf{x}_i, \mathbf{c}_j) = \sqrt{(x_i^1 - c_j^1)^2 + (x_i^2 - c_j^2)^2 + \ldots + (x_i^k - c_j^k)^2}. \quad (2)$$

Then, neuron $\mathbf{p}_j$ is activated only if $d(\mathbf{x}_i, \mathbf{c}_j) \leq r_j$. If no neurons are activated for the feature vector $\mathbf{x}_i$, a new neuron $\mathbf{p}_{n+1}$ with a hypersphere center $\mathbf{c}_{n+1} = [x_i^1, x_i^2, \ldots, x_i^k]$, label $l_{\mathbf{x}_i}$, and radius R is generated, where $[x_i^1, x_i^2, \ldots, x_i^k]$ and $l_{\mathbf{x}_i}$ are the feature values and label of the feature vector $\mathbf{x}_i$, respectively, and R is the pre-defined global radius. In addition, the total number of neurons n is increased by one. Since the TR-RCE learns with one global radius R for all data, confusion occurs in some areas of the feature space. Consider two neurons $\mathbf{p}_1$ and $\mathbf{p}_2$ learned by the two feature vectors $\mathbf{x}_1$ and $\mathbf{x}_2$ with labels l_a and l_b in the 2D feature space, as shown in Figure 1a. In other words, neurons $\mathbf{p}_1$ and $\mathbf{p}_2$ can be represented by $\mathbf{p}_1 = [\mathbf{c}_1, r_1, l_1]$ and $\mathbf{p}_2 = [\mathbf{c}_2, r_2, l_2]$, where $\mathbf{c}_1 = \mathbf{x}_1$, $\mathbf{c}_2 = \mathbf{x}_2$, $r_1 = r_2 = R$, $l_1 = l_a$ and $l_2 = l_b$. Then, in the recognition process, a new feature vector $\mathbf{x}_3$ with label l_b enters the confusion area, as in Figure 1b, which activates both neurons. In this case, the RCE-NN recognizes the feature vector $\mathbf{x}_3$ as label l_a because $d(\mathbf{x}_3, \mathbf{c}_1)$ is smaller than $d(\mathbf{x}_3, \mathbf{c}_2)$. The learning method of TR-RCE yields many such areas of confusion, thereby degrading the recognition accuracy.

The DDA-RCE was developed to solve the inherent problems associated with this method. When a neuron $\mathbf{p}_j$ with a different label than the input feature vector $\mathbf{x}_i$ is activated in the learning process as shown in Figure 1c, the r_j of the neuron is reduced as follows:

$$r_j = d(\mathbf{x}_i, \mathbf{c}_u), \quad (3)$$

where u is the index of the neuron with the smallest distance value among those activated by the current input feature vector. Then, in the recognition process, the RCE-NN correctly recognizes the feature vector $\mathbf{x}_3$ as label l_b, as shown in Figure 1d. This technique increases the recognition accuracy in areas of confusion. However, the radius of a specific neuron becomes excessively small when the radius is adjusted based on the minimum distance in the learning process, and further unnecessary neurons are learned, thereby increasing the system complexity.

In [15], the technique of measuring f_j is introduced, where f_j is the activation count of each neuron in the learning process. That is, when the specific neuron $\mathbf{p}_j$ is activated for the input feature vector $\mathbf{x}_i$, the f_j value is incremented by one to estimate the reliability of the neuron. By applying the activation count to the output of each neuron, a recognition result reflecting the reliability of each neuron is obtained. However, since all neurons are learned with a single global radius, as in TR-RCE, a confusion area is created in the feature space, thus reducing recognition accuracy. Therefore, the HPL algorithm was proposed in [16,17], which reduced the global radius from the maximum (R_{max}) to the minimum (R_{min}) according to the iteration interval. The global radius is decreased as follows:

$$R_{w+1} = \alpha \cdot R_w, \quad (4)$$

where α is a reduction rate for global radius and w is the index for iteration. In addition, to determine whether each neuron $\mathbf{p}_j$ is a suitable neuron in the learning process, the prototype density value D_j is calculated as follows:

$$D_j = \frac{f_j}{v_j}, \quad (5)$$

where v_j is the volume of the hyperspherical classifier of each neuron $\mathbf{p}_j$. If the prototype density value D_j is less than the pre-defined threshold value τ, it is determined to be an inappropriate neuron and

that neuron is removed from **P**. However, the f_j is updated to the same value regardless of whether the input feature vector is activated near or far from the hypersphere center. Therefore, the estimated reliability of each learned neuron is inaccurate and unnecessary neurons are not removed. In addition, HPL-RCE does not adjust the radius as in DDA-RCE when a neuron with a different label than the input feature vector is activated.

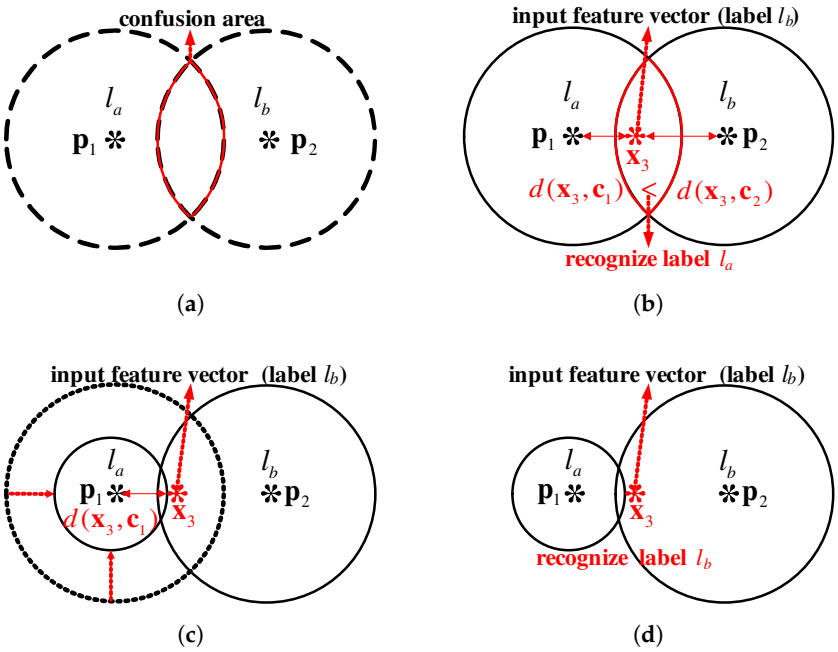


Figure 1. Learning and recognition process of traditional (TR)-restricted coulomb energy (RCE) and dynamic decay adjustment (DDA)-RCE. (**a**) Example of TR-RCE learning for two feature vectors $\mathbf{x}_1$ and $\mathbf{x}_2$, (**b**) recognition result for the feature vector $\mathbf{x}_3$ after TR-RCE learning, (**c**) example of DDA-RCE learning for three feature vectors $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$, (**d**) recognition result for the feature vector $\mathbf{x}_3$ after DDA-RCE learning.

3. Proposed Learning Algorithm for RCE-NN

3.1. Proposed Learning Algorithm

In order to overcome the problems of existing learning schemes, the proposed learning algorithm estimates reliability by dividing the activation region associated with each neuron and increasing the f_j with different factors for each region in the learning process as follows:

$$f_j^{t+1} = \begin{cases} f_j^t + 1, & \frac{r_j + R_{min}}{2} < d(\mathbf{x}_i, \mathbf{c}_j) \leq r_j \\ f_j^t + \theta, & R_{min} < d(\mathbf{x}_i, \mathbf{c}_j) \leq \frac{r_j + R_{min}}{2} \\ f_j^t + \gamma, & d(\mathbf{x}_i, \mathbf{c}_j) \leq R_{min} \end{cases}, \quad (6)$$

where θ and γ are experimentally determined according to the distribution of the feature vector ($1 < \theta < \gamma$). That is, when the current input feature vector activates a neuron in a region near the hypersphere center, the f_j is increased with a higher factor. When it activates a neuron in the boundary area, the f_j is increased with a lower factor.

An example of learning process based on the f_j of each neuron is shown in Figure 2. Consider three neurons $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$ learned by the three feature vectors with labels l_a, l_b and l_c in the 2D feature space as shown in Figure 2a. If the distribution of input feature vector to be used in the learning process is shown in Figure 2b, the activation counts of neurons $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$, which were measured during one iteration with HPL-RCE, were 9, 8 and 8, respectively. Even though $\mathbf{p}_3$ neuron covers

feature vectors better than other neurons, the reliability from activation count is estimated similarly. As a result, when the learning process of HPL-RCE was iteratively performed with a $\tau = 50$, all neurons still remained and unnecessary neurons $\mathbf{p}_4$ and $\mathbf{p}_5$ are generated as shown in Figure 2c. This leads to an increase in complexity and degradation of recognition accuracy for other test data. In contrast, if the activation region is divided as shown in Figure 2d as presented in the proposed learning algorithm, the activation counts of neurons $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$, which measured during one iteration period, are 34 (=10 + 20 + 4), 20 (=0 + 15 + 5) and 65 (=50 + 15 + 0), respectively ($\theta = 5$, $\gamma = 10$). Then, when the $\tau = 50$, $\mathbf{p}_1$ and $\mathbf{p}_2$ neurons are removed, and $\mathbf{p}_3$ neuron remains. In the additional iteration periods, new neurons $\mathbf{p}_4$ and $\mathbf{p}_5$ centered on different feature vectors are generated for the l_a and l_b labels and it is confirmed that even fewer neurons cover all the feature vectors than the HPL-RCE as shown in Figure 2e. In addition, when the label of the activated neuron is different from that of the input feature vector, the proposed algorithm gradually decreases the radius of this activated neuron according to reduction rate β as follows:

$$r_j^{t+1} = \beta \cdot r_j^t. \tag{7}$$

This not only improves the recognition accuracy but also suppresses the generation of unnecessary neurons. Algorithm 1 summarizes the proposed learning algorithm.

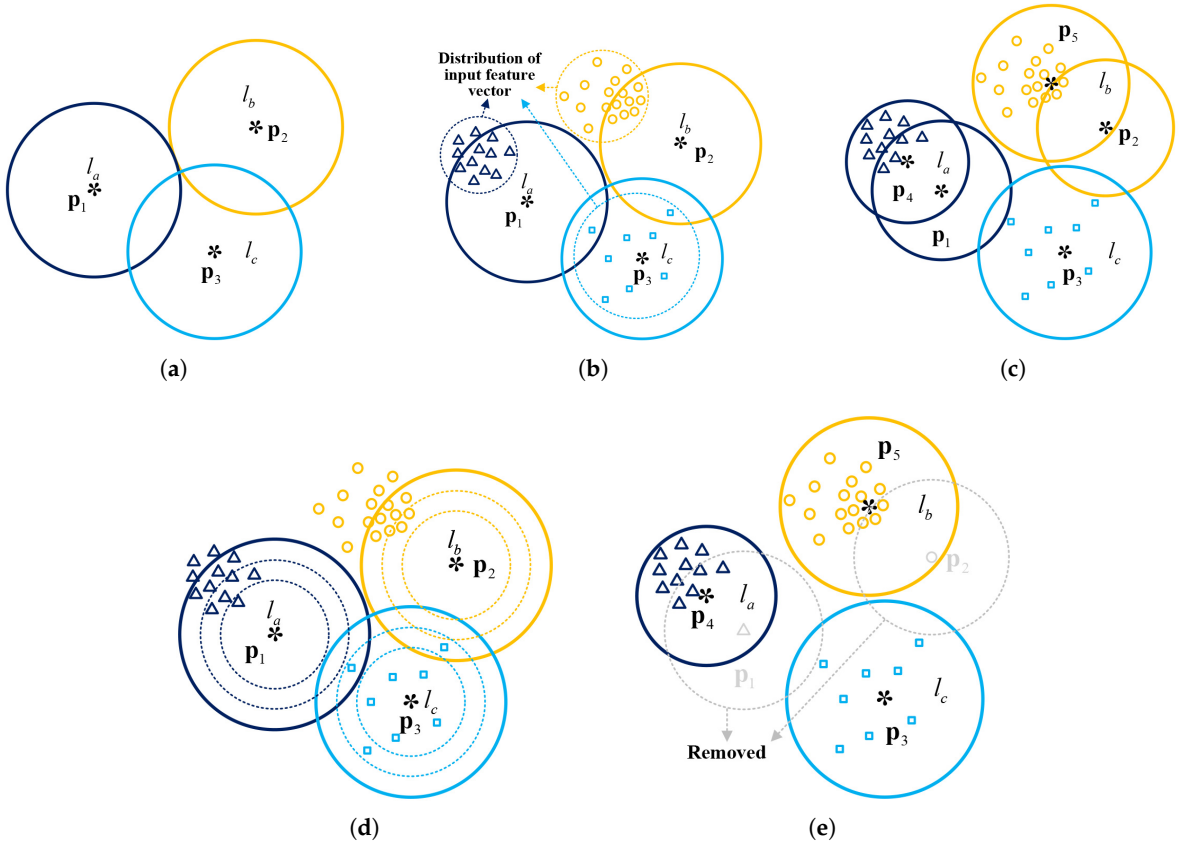


Figure 2. Learning process of hierarchical prototype learning (HPL)-RCE and the proposed RCE. (**a**) Example of 2D feature space learned with three feature vectors, (**b**) distribution of the feature vectors to be used in the learning process, (**c**) learning results of HPL-RCE, (**d**) division of the activation region as presented in the proposed learning algorithm, (**e**) learning results of the proposed algorithm.

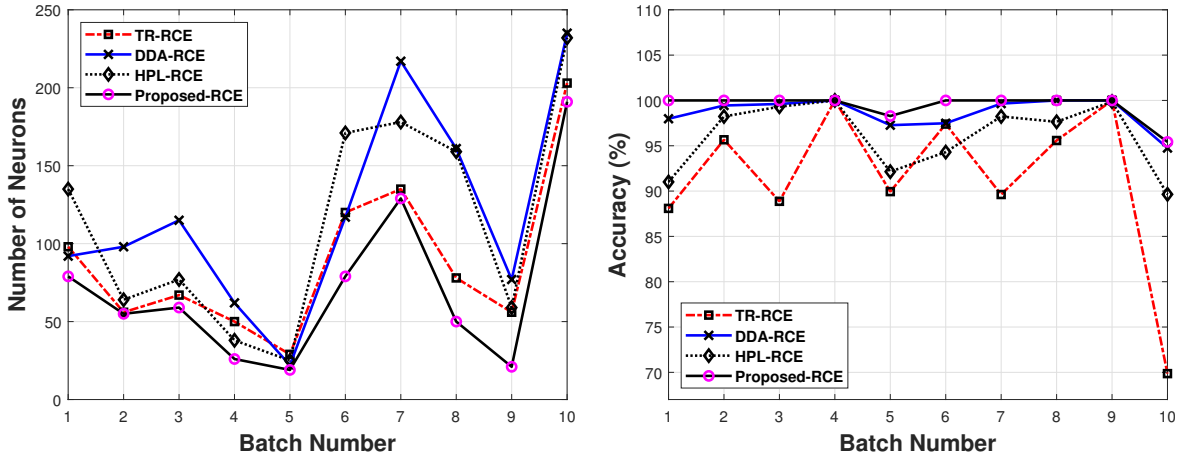
Algorithm 1: The proposed learning algorithm.

initialize global radius $R_w = R_{max}$
while $R_w > R_{min}$ **do**
  for $i = 1$ *to* m **do**
    for $j = 1$ *to* n **do**
      $\mathbf{d}_{i,j} \leftarrow d(\mathbf{x}_i, \mathbf{c}_j)$
      if $\mathbf{d}_{i,j} \leq r_j$ **then**
        if $l_j = l_{\mathbf{x}_i}$ **then**
          if $\frac{r_j + R_{min}}{2} < \mathbf{d}_{i,j} \leq r_j$ **then**
            f_j += 1
          else if $R_{min} < \mathbf{d}_{i,j} \leq \frac{r_j + R_{min}}{2}$ **then**
            f_j += θ
          else
            f_j += γ
          end
          jump out of for loop
        else
          $r_j = \beta \cdot r_j$
        end
      end
    end
    if *no neurons are activated for* $\mathbf{x}_i$ **then**
      create a new neuron $\mathbf{p}_{n+1}$ centered at $\mathbf{x}_i$
      with radius R_w, label $l_{\mathbf{x}_i}$, and $f_j = 0$
    end
  end
  for $j = 1$ *to* n **do**
    if $f_j \geq \tau$ **then**
      retain the neuron $\mathbf{p}_j$
    else
      discard $\mathbf{p}_j$
    end
  end
  $R_w = \alpha \cdot R_w$
end

3.2. Performance Evaluation Results

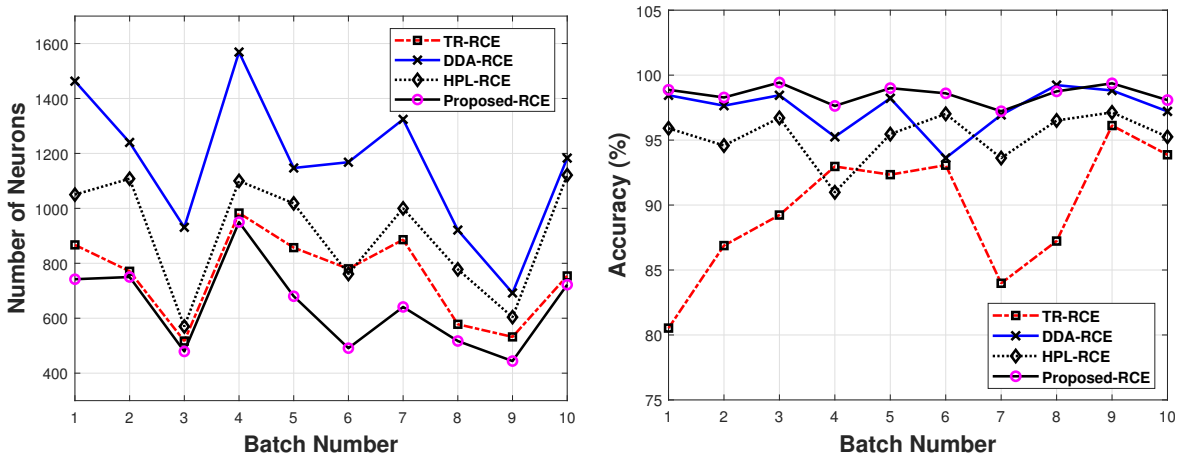
We conducted learning and recognition tasks with all methods for RCE-NN on two datasets for a gas sensor and motion-capture hand postures (MCHP) [18,19]. Two datasets have been used in gas detection and human machine interation (HMI) [20,21]. The gas sensor dataset contains 128-dimensional feature vectors taken from 16 gas sensor arrays for 36 months to detect six toxic gase. The 13,910 feature vectors were divided into 10 batches, and each batch is configured to distribute the six target gases uniformly in time order [22]. The MCHP dataset is comprised of five static gestures (hand poses) captured from 12 users using a Vicon motion-capture camera system and a glove with attached infrared markers on certain joints [23]. The five gestures captured were a fist, pointing with one finger, pointing with two fingers, stop (hand flat), and grab (fingers curled). The MCHP dataset contains 36-dimensional feature vectors, and 78,096 feature vectors are divided into 10 batches.

The learning and recognition experiments were performed via two strategies, and we analyzed the results of the number of learned neurons and recognition accuracy. At first, learning was performed separately for each batch and the recognition accuracy was measured for each learned RCE-NN (θ = 3, γ = 5). The recognition accuracy and number of learned neurons of all methods for RCE-NN are presented in Figures 3 and 4. Figure 3 shows the number of learned neurons and the recognition accuracy for the gas-sensor dataset, respectively. Figure 4 depicts the number of learned neurons and the recognition accuracy for the MCHP dataset, respectively. As shown in the Figures 3 and 4, the RCE-NN with the proposed learning algorithm shows better recognition accuracy with the average of 55.5% fewer neurons than DDA-RCE, which requires the largest number of learned neurons. In addition, it shows better recognition accuracy from 3% to 8% with fewer neurons compared to TR-RCE and HPL-RCE.



(**a**) (**b**)

Figure 3. Performance evaluation results for gas dataset (**a**) the number of learned neurons for each batch, (**b**) the recognition accuracy for each batch.



(**a**) (**b**)

Figure 4. Performance evaluation results for motion-capture hand postures (MCHP) dataset (**a**) The number of learned neurons for each batch, (**b**) The recognition accuracy for each batch.

Secondly, learning and recognition were performed by selecting 7000 learning feature vectors and 6910 test feature vectors randomly from each dataset. As a result of the performance evaluation as shown in Table 1, the RCE-NN with the proposed learning algorithm shows good recognition accuracy with half the number of neurons that DDA-RCE used. In addition, the proposed algorithm shows better recognition accuracy from 3% to 12% with fewer neurons than TR-RCE and HPL-RCE.

Table 1. Performance evaluation results ($\theta = 5, \gamma = 10$).

Algorithm	Gas Sensor [18]		MCHP [19]	
	Number of Neurons	Recognition Accuracy	Number of Neurons	Recognition Accuracy
Proposed	449	96.86%	641	98.52%
TR [13]	509	84.31%	752	89.65%
HPL [16]	559	91.29%	911	95.32%
DDA [14]	1026	95.17%	1164	97.38%

Figure 5 demonstrates the comparison results for the gas sensor dataset with existing recognition algorithms, such as support vector machine with genetic algorithm (SVM-GA), multilayer perceptron (MLP), binary decision tree (BDT) based on classification and regression tree (CART), and K-nearest neighbor (KNN) [24]. As shown in Figure 5, the proposed algorithm has better performance than other pattern recognition algorithms such as MLP, SVM, BDT, and KNN. Although SVM-GA shows slightly better performance compared to the proposed algorithm, the genetic algorithm for the learning process requires a great deal of complexity and much learning time.

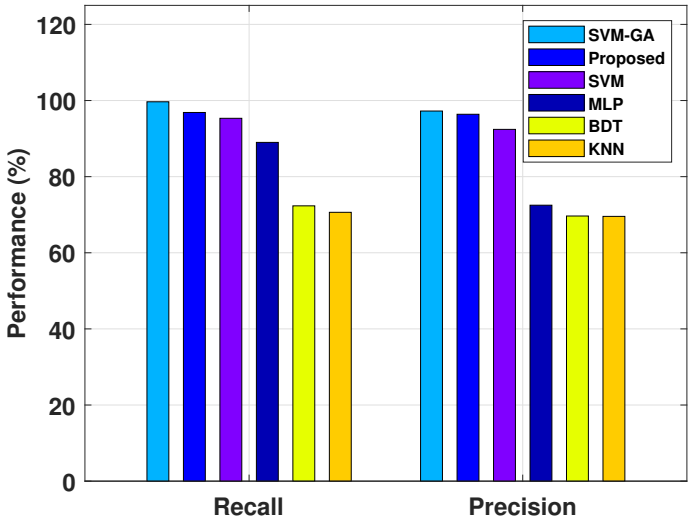


Figure 5. Performance comparison for gas recognition algorithms.

4. Hardware Architecture Design

Figure 6 shows the block diagram of the proposed RCE-NN processor, including a feature memory unit (FMU), neuron unit (NU), activated neuron detection unit (ANDU), and network control unit (NCU). In the learning process, the unlearned NUs store the input feature vector from FMU in the neuron memory, and the learned NUs calculate distance between the input feature vector and the stored vector which represents neuron center. Then, the learned NUs compare the distance to the stored radius, and the activated NUs send the distance and label to the ANDU. The ANDU analyzes the output of the activated NUs and transfers the information of the minimum distance and label to the NCU. Figure 7 depicts the architecture of the NCU, which determines whether to generate a new NU or remove an existing NU. When a new NU is generated, the value of the neuron count register is increased by 1 to monitor the number of learned NUs. In addition, if the labels of the currently activated NUs are different from each other, the NCU adjusts the radii of the conflicted NUs through the radius adjustment unit (RAU). The RAU transmits the current feature vector's label $l_{\mathbf{x}_i}$ and radius adjustment signal in parallel to each NU, and each activated NU compares $l_{\mathbf{x}_i}$ and l_j. If $l_{\mathbf{x}_i}$ and l_j are the same, f_j is increased by (6) through the activation counter as shown in Figure 8, where f_{max} is the upper limit of f_j to prevent unnecessarily increase. Conversely, the radius of an activated NU with a different label than the input feature vector is decreased by (7) through the radius decision unit in NU.

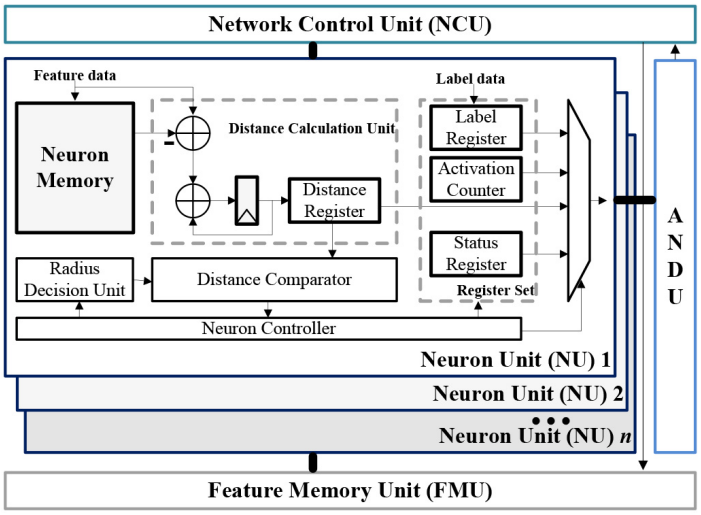


Figure 6. Block diagram of the proposed RCE-neural network (NN) processor.

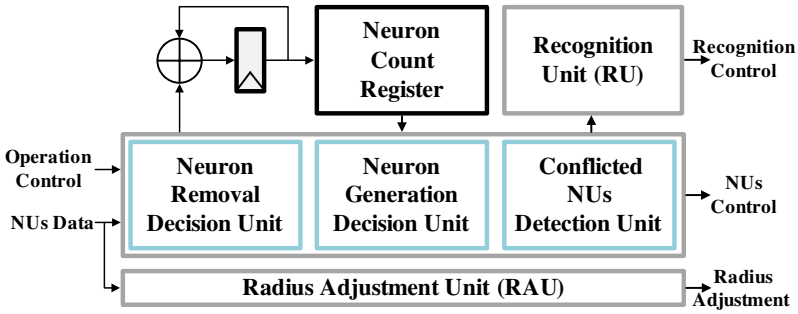


Figure 7. Block diagram of the network control unit (NCU).

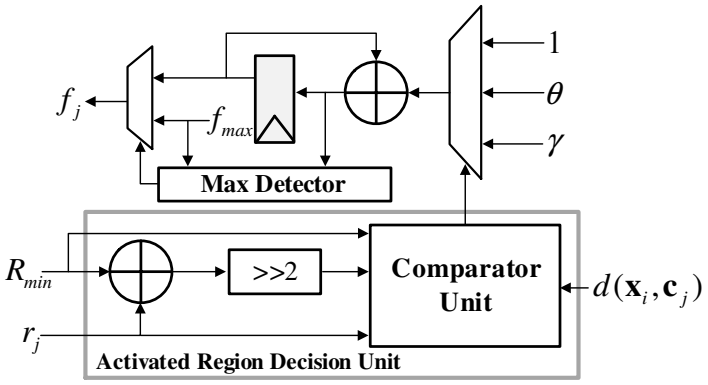


Figure 8. Block diagram of the activation counter.

In the recognition process, all learned NUs compute the distance between the feature vector from the FMU and the vector stored in the neuron memory, as in the learning process. Then, the activated NUs transmit their distances to the ANDU, and the ANDU sends the best-matched NU's distance to the NCU. The NCU transmits the best-matched NU's distance and the recognition control signal to each NU through the recognition unit (RU). Each NU compares the stored distance with the best-matched NU's distance and outputs the label if it matches. Finally, the ANDU receives the best-matched NU's label and outputs recognition results.

5. Implementation Results

The proposed RCE-NN processor was designed in Verilog hardware description language (HDL) and synthesized using the Synopsys Design Compiler to gate-level circuits with a 55 nm CMOS

standard cell library. The key features of the proposed RCE-NN processor are summarized in Table 2. It was observed that the proposed architecture required 197.8 K logic gates and 163.8 KB memory with the total die size of 0.535 mm^2. Learning time for one feature vector of 128 byte was 0.93 µs and recognition time was 0.96 µs at the operating frequency of 150 MHz. The total power consumption measured with the test platform is 6.64 mW.

Table 2. Key features of the implemented restricted coulomb energy neural network (RCE-NN) processor.

Prameter		Value
Technology		55 nm CMOS
Operating Frequency		150 MHz
Internal Memory		163.8 KB
Gate Counts	RCE-NN Processor	197.8 K
	Neuron Unit	1.5 K
RCE-NN Processor Size		0.535 mm^2
Processing time (128 Byte)		0.93 µs (learning), 0.96 µs (recognition)
Power		6.64 mW

The RCE-NN processor was integrated in a system-on-chip (SoC) designed for sensor signal processing as shown in Figure 9 and verified in real-time with test platform depicted in Figure 10. The sensor signal processing SoC consisted of an ARM Cortex-M3 micro controller unit (MCU), a sensor interface unit (SIU), a feature extraction unit (FEU), and the proposed RCE-NN processor with the total die size of 2.97 mm × 2.96 mm. The data extracted from each sensor was pre-processed in the SIU and then converted into feature data in the FEU. The feature data were transferred to the RCE-NN processor which performs the learning and recognition.

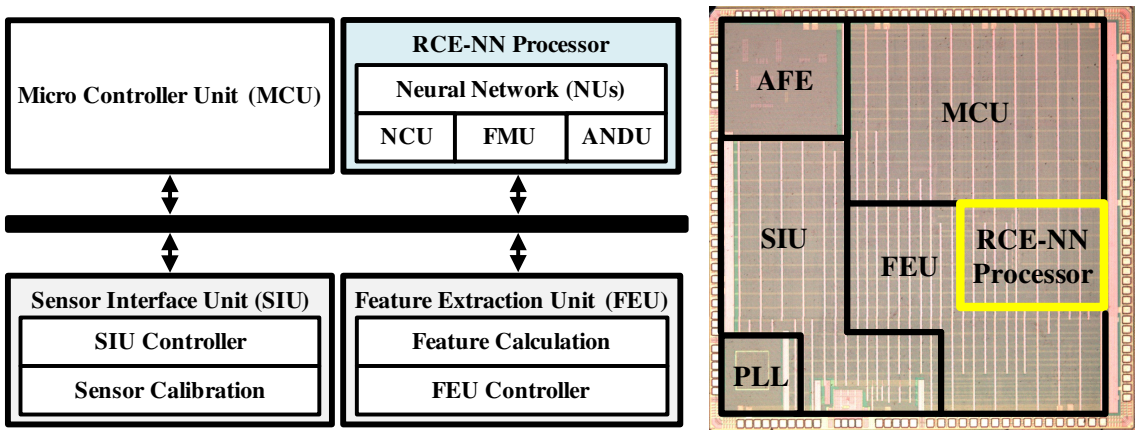


Figure 9. Sensor signal processing system-on-chip (SoC). (**a**) block diagram, (**b**) die photograph.

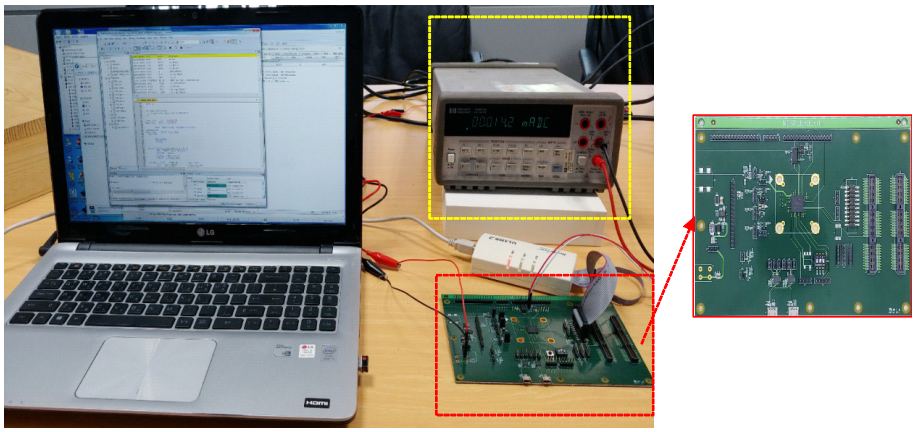


Figure 10. Verification platform of the sensor signal processing SoC.

In order to compare the complexity of the proposed RCE-NN processor and the existing RCE-NNs, we implemented each algorithm presented in Table 1 for the gas dataset. Table 3 summarizes the complexity metrics such gate counts and memory requirements for each implementation. NU of the proposed RCE-NN processor was the largest because of additional resource for the activation counter shown in Figure 8. However, since it shows better recognition accuracy with fewer neurons than TR-RCE and HPL-RCE, the total gate counts and internal memory requirements were smallest.

Table 3. Complexity comparison with existing RCE-NN implementations.

Target Algorithm	Gate Counts		Internal Memory	Number of Neurons
	Neuron Unit	RCE-NN Processor		
DDA [14]	1.21 K	1247.26 K	131,328 KB	1026
HPL [16]	1.32 K	743.68 K	71,552 KB	559
Proposed	1.5 K	679.3 K	57,472 KB	449

6. Conclusions

In this paper, we proposed an efficient RCE-NN processor with an improved learning algorithm. Learning algorithms in existing RCE-NNs show degraded recognition performance and increased complexity because of the inaccurate reliability of learned neuron and inefficient radius adjustments. To overcome such problems, the proposed algorithm divides the activation region of each neuron in the learning process and measures the reliability with different factors for each area, and gradually reduces the radius using a pre-defined rate. In performance evaluation using two datasets, RCE-NN with the proposed learning algorithm showed good recognition accuracy with fewer neurons compared with existing RCE-NNs. The hardware was also designed for real time operation. The designed RCE-NN processor has a logic gate count of 197.8 K with a die size of 0.535 mm^2 and the memory requirement of 163.8 KB, and it can support real-time learning and recognition at an operating frequency of 150 MHz.

Author Contributions: J.C. designed the algorithm, performed the simulation and experiment, and wrote the paper. Y.J. (Yongchul Jung) and S.L. implemented the evaluation platform and performed the experiment. Y.J. (Yunho Jung) conceived and led the research, analyzed the experimental results, and wrote the paper.

Funding: This research received no external funding.

Acknowledgments: This work was supported by the Technology Innovation Program, 10073122 and 10079634, funded by the Ministry of Trade, Industry and Energy (MOTIE, Korea) and CAD tools were supported by IDEC.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhou, E.; Fang, L.; Yang, B. Memristive Spiking Neural Networks Trained with Unsupervised STDP. *Electronics* **2018**, *7*, 396. [CrossRef]
2. Garner, J.; Mestres, A.; Alarcon, E.; Cabellos, A. Neural Networks for Classification: A Survey. *IEEE Trans. Syst. Man Cybern. Part C* **2000**, *30*, 451–464.
3. Alom, M.; Tha, T.; Yakopcic, C.; Westberg, S.; Sidike, P.; Nasrin, M.; Hasan, M.; Essen, B.; Awwal, A.; Asari, V. A State-of-the-Art Survey on Deep Learning Theory and Architectures. *Electronics* **2019**, *8*, 292. [CrossRef]
4. Sze, V.; Chen, Y.; Yang, T.; Ember, J. Efficient Processing of Deep Neural Networks: A Tutorial and Survey. *Proc. IEEE* **2017**, *105*, 2295–2329. [CrossRef]
5. Tang, F.; Wu, W.; Liu, J.; Wang, H.; Xian, M. Privacy-Preseving Distributed Deep Learning via Homomorphic Re-Encryption. *Electronics* **2019**, *8*, 411. [CrossRef]
6. Kouda, N.; Matsui, N. On the Function Approximation in Restricted Coulomb Energy Neural Network with Gaussian Radial Basis Function. In Proceedings of the World Automation Congress, Kobe, Japan, 19–23 September 2010; pp. 1–3.
7. Nepes Corporation. Available online: http://www.theneuromorphic.com (accessed on 21 April 2019).
8. General Vision Inc. Available online: https://www.general-vision.com/neuromem/cm1k (accessed on 21 April 2019).
9. Sardar, S.; Ananda Babu, K. Hardware Implementation of Real-time, High Performance, RCE-NN Based Face Recognition System. In Proceedings of the IEEE Conference on VLSI Design and Embedded Systems, Mumbai, India, 5–9 January 2014.
10. Labonte, G.; Deck, W. Infrared Target-Flare Discrimination using ZISC Hardware Neural Network. *J. Real-Time Image Process.* **2010**, *5*, 11–32. [CrossRef]
11. Kai, Y.; Hu, Y.; Siegel, M. Onboard Feature Indexing from Satellite Lidar Images. In Proceedings of the IEEE IWADC, Perugia, Italy, 8–10 September 2003.
12. Reilly, D.; Cooper, L.; Elbaum, C. Neural Model for Category Learning. *Biol. Cybern.* **1982**, *45*, 35–41. [CrossRef] [PubMed]
13. Hudak, M. RCE Classifiers: Theory and Practice. *Cybern. Syst.* **1992**, *23*, 483–515. [CrossRef]
14. Berthold, M.; Diamond, J. Boosting the Performance of RBF Networks with Dynamic Decay Adjustment. In Proceedings of the 7th International Conference on Neural Information Processing Systems, Denver, CO, USA, 1994; pp. 521–528.
15. Yin, X.; Don, G.; Xie, M. Hand Image Segmentation Using Color and RCE Neural Network. *Robot. Auton. Syst.* **2001**, *34*, 235–250. [CrossRef]
16. Don, G.; Xie, M. Color Clustering and Learning for Image Segmentation Based on Neural Networks. *IEEE Trans. Neural Netw.* **2005**, *16*, 925–936.
17. Sui, C.; Kwok, N.; Ren, T. A Restricted Coulomb Energy (RCE) Neural Network System for Hand Image Segmentation. In Proceedings of the IEEE Conference on Computer and Robot Vision, St. Johns, NL, Canada, 25–27 May 2011; pp. 270–277.
18. UCI Machine Learning Repository, Gas Dataset. Available online: https://archive.ics.uci.edu/ml/datasets/gas+sensor+array+drift+dataset (accessed on 21 April 2019).
19. UCI Machine Learning Repository, Hand Posture Dataset. Available online: https://archive.ics.uci.edu/ml/datasets/Motion+Capture+Hand+Postures (accessed on 21 April 2019).
20. Shirahama, K.; Grzegorzek, M. On the Generality of Codebook Approach for Sensor-Based Human Activity Recognition. *Electronics* **2017**, *6*, 44. [CrossRef]
21. Duran, C.; Benjumea, J.; Carrillo, J. Response Optimization of a Chemical Gas Sensor Array using Temperature Modulation. *Electronics* **2018**, *7*, 54. [CrossRef]
22. Vergara, A.; Vembu, S.; Ayhan, T.; Ryan, M.; Huerta, R. Chemical Gas Sensor Drift Compensation Using Classifier Ensembles. *Sens. Actuators B Chem.* **2012**, *166–167*, 320–329. [CrossRef]

23. Gardner, A.; Kanno, J. Measuring Distance Between Unordered Sets of Different Sizes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 137–143.
24. Wang, K.; Ye, W.; Zhao, X.; Pan, X. A Support Vector Machine-Based Genetic Algorithm Method for Gas Classification. In Proceedings of the 2nd International Conference on Frontiers of Sensors Technologies (ICFST), Shenzhen, China, 14–16 April 2017; pp. 363–366.