2017 International Symposium on Nonlinear Theory and Its Applications,
NOLTA2017, Cancun, Mexico, December 4-7, 2017

Development and Evaluation of Neural Networks using Oxide Semiconductor Synapses for Letter Reproduction

Hiroki Yamane†, Tomoya Kameda†, Mutsumi Kimura†,‡, and Yasuhiko Nakashima†

†Graduate School of Information Science, Nara Institute of Science and Technology
Takayama, Ikoma 630-0192, Japan
‡Department of Electronics and Informatics, Ryukoku University
Seta, Otsu 520-2194, Japan
Email: mutsu@rins.ryukoku.ac.jp

Abstract– Neural networks using oxide semiconductor synapses are developed and evaluated for letter reproduction. It is assumed that amorphous metal-oxide semiconductor devices are used for the synapse elements, and the characteristic degradation is utilized for the learning rule named modified Hebbian learning. First, we explain architecture and operation of a Hopfield neural network, simulate the letter recognition by the neural network, and show a degradation map. Particularly in this presentation, we explain the simulation algorithm in detail. Next, we explain architecture and operation of a cellular neural network, and simulate them. Particularly in this presentation, we explain the simulation algorithm in detail also for the cellular neural network. In addition. we compare the Hopfield and cellular neural networks, and it is found that the former has higher performance, although the latter has a simple structure.

1. Introduction

Artificial intelligences mimic biological brains and realize thinking machines as key technologies in future societies, and neural networks are promising architectures with many advantages, such as, self-organization, self-learning, parallel distributed computing, fault tolerance, etc [1-4], whose advantages are obtained by connecting a large number of neuron elements with a much larger number of synapse elements. However, because the conventional neural networks are complicated software executed on high-spec hardware, the machine size is very bulky and power consumption is unbelievably huge. Moreover, some of the aforementioned advantages, such as, parallel distributed computing and fault tolerance, are not acquired, because they are executed on conventional Neumann-type computers, which sequentially handle processes and stop by only one breakdown.

Currently, we are investigating neural networks built by actual hardware instead of virtual software, which we name "brain-type integrated system" that can be compact, low power, robust, and integrated on everything in future life [5-11]. In order to realize such system, it is necessary to simplify the processing elements, such as, neuron elements and synapse elements, which is reported in the previous articles, and fabricate them at low cost, which is possible by using oxide semiconductor synapses, because they can be fabricate using abundant materials and printing process and stacked for 3D structure.

In this study, neural networks using oxide semiconductor synapses are developed and evaluated for letter reproduction. It is assumed that amorphous metal-oxide semiconductor devices are used for the synapse elements, and the characteristic degradation is utilized for the learning rule named modified Hebbian learning. First, we explain architecture and operation of a Hopfield neural network, simulate the letter recognition by the neural network, and show a degradation map. Next, we explain architecture and operation of a cellular neural network, simulate them, and compare the Hopfield and cellular neural networks. Particularly in this presentation, explain the simulation algorithm in detail for both the neural network.

2. Hopfield Neural Network

2.1. Architecture and Operation

An architecture of the Hopfield neural network is shown in Fig. 1 [12]. Here, all neuron elements are connected to all neuron elements through synapse elements. The neuron elements are designed in an FPGA chip, and the synapse elements are fabricated using amorphous metal-oxide semiconductor devices. Amorphous metal-oxide semiconductor films are sandwiched between multiple upper electrode lines and multiple lower electrode lines, and all cross points work as synapse elements. A neuron element has a binary states of either fire or stable, which is decided by input signals from all neuron elements weighted by electric conductance as synaptic weights and outputted to all neuron elements.

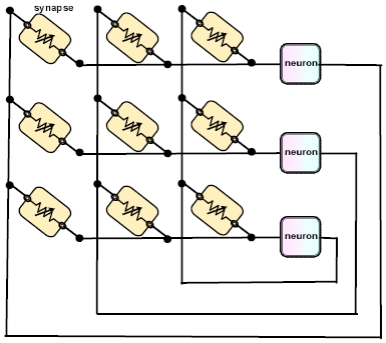

Fig. 1. Architecture of the Hopfield neural network.

An operation of the Hopfield neural network is shown in Fig. 2. For the neuron with stable states, GND=0V is applied to the corresponding upper and lower electrodes, whereas for the neuron with fire states, Vdd=3.3V is applied. At the cross points where the applied voltages are different, electric currents flow as shown by the red arrow, and the electric conductance is degraded, which realize the modified Hebbian learning, where the electric conductance is gradually degraded as the time goes,

because the amorphous metal-oxide semiconductor films are designed so that trap states are generated [13].

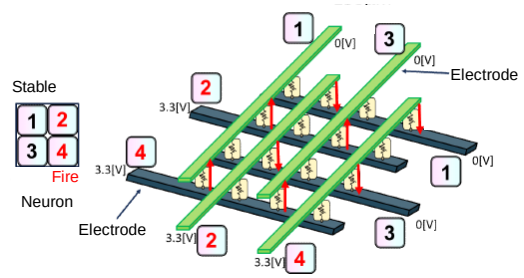

Fig. 2. Operation of the Hopfield neural network.

2.2. Simulated Results

We assume the architecture and operation of the Hopfield neural network, model the synapse elements, and simulate the letter recognition, as shown in Figs. 3 and 4. First, a pixel pattern corresponding to an input letter is inputted to neurons as firing states. Afterword, the states in all neurons are calculated using the majority rule of the neighboring neurons with consideration of the synaptic connection strengths corresponding to the conductance of the oxide semiconductor synapses. Next, after the all the states in the neurons are settled, the modified Hebbian learning are done, namely, when the neurons connected to the synapse are in different states, the synaptic connection strength decreases based on the deterioration model of the oxide semiconductor synapses. After that, a pixel pattern slightly distorted from the input pattern is inputted to neurons as firing states and immediately removed, and it is checked whether the output pattern from the neurons becomes the original inputted pattern. Finally, these procedures are repeated for multiple input patterns.

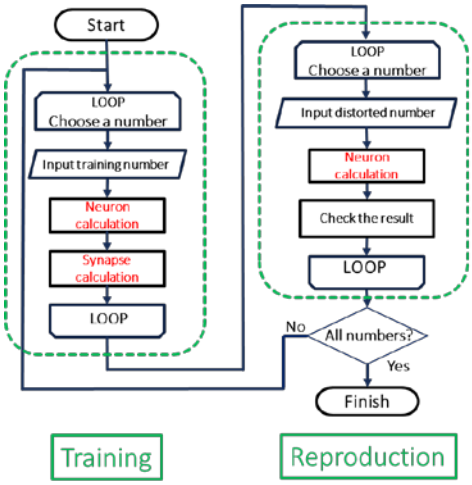

Fig. 3. Simulation algorithm of the Hopfield neural network.

```
#define DS 1 /* deterioration speed */
#define NWIDTH 9 /* pixel size */
for (y=0; y<NWIDTH*NWIDTH-1; y++) {
  for (x=0; x<NWIDTH*NWIDTH-1; x++) {
    CURR[y][x] = (oNEURON[x/NWIDTH][x%NWIDTH] -
        oNEURON[y/NWIDTH][y%NWIDTH]) / synapse[y][x]; ..... ※1
    CSUM[y][x] += abs(CURR[y][x]); ..... ※2
    synapse[y][x] = DS*CSUM[y][x] + synapseinit; ..... ※3
} }
```

```
#define MAGNI 3.3 /* max voltage */
#define NWIDTH 9 /* pixel size */
/* Current */
for (y=0; y<NWIDTH*NWIDTH-1; y++) {
  for (x=0; x<NWIDTH*NWIDTH-1; x++) {
    CURR[y][x] = (oNEURON[x/NWIDTH][x\%NWIDTH] -
          oNEURON[y/NWIDTH][y%NWIDTH])/synapse[y][x];
} } ..... ※4
/* update NEURON */
for (y=0; y<NWIDTH; y++) {
  for (x=0; x<NWIDTH; x++) {
    for (i=0; i<NWIDTH; i++) {
      for (j=0; j<NWIDTH; j++) {
        ONELINE[y][x] += (CURR[9*y+x][9*i+j]); ..... ※5
    } }
    nNEURON[y][x] = oNEURON[y][x] + ONELINE[y][x];
    if ( step ≠ STEPS-1 ) {
      if (nNEURON[y][x] < 0    ) nNEURON[y][x] = 0;
      if (nNEURON[y][x] > MAGNI ) nNEURON[y][x] = MAGNI;
    } ..... ※6
    else {
      if (nNEURON[y][x] ≤ (MAGNI/2))  nNEURON[y][x] = 0;
      if (nNEURON[y][x] > (MAGNI/2))  nNEURON[y][x] = MAGNI;
} } } ..... ※7
memcpy(oNEURON, nNEURON, sizeof(nNEURON));
```

Fig. 4. Simulation program of the Hopfield neural network.

The degradation map of the synapse elements is shown in Fig. 5. Here, 9×9=81 neuron elements corresponds to the image pixels. Number letters, 0, 1, 2, and 3, are learned. The red grayscale indicates the degradation degree of the electric conductance of the synapse elements.

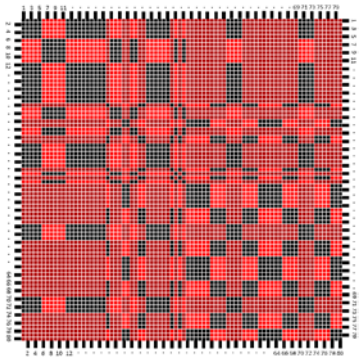

Fig. 5. Degradation map of the synapse elements

The simulation results of the letter recognition are shown in Fig. 6. After a while of the learning, even if distorted letters are inputted, they are revised, and corrected letters are outputted. This is a part of procedure of the letter recognition with pattern matching. In conclusion, this Hopfield neural network can recognize hand-written letters.

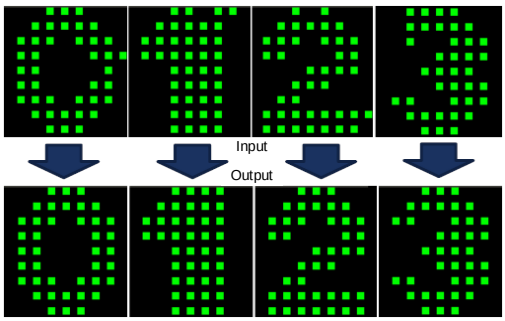

Fig. 6. Simulation results of the letter recognition.

3. Cellular Neural Network

3.1. Architecture and Operation

Architectures of the cellular neural network are shown in Fig. 7 [14]. Here, neuron elements are aligned like an array matrix. They are connected only neighboring neuron elements orthogonally as shown in the left figure or orthogonally and diagonally as shown in the right figure. The neuron elements are fabricated in an LSI chip, and the synapse elements are fabricated using amorphous metal-oxide semiconductor devices. Amorphous metal-oxide semiconductor films are deposited on the LSI chip, and the planar films work as synapse elements.

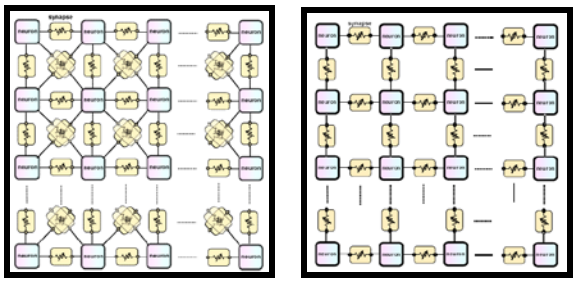

Fig. 7. Architectures of the cellular neural network.

An operation the cellular neural network is shown in Fig. 8. Here, I/O neurons are assigned to every two neuron elements, and hidden neurons are assigned to the other neuron elements. When 0 is learned, Vdd=1.8V is applied to the corresponding I/O neurons, and GND=0V is applied to the other I/O neurons. All neuron elements dynamically become either fire or stable states. Through the synapse elements that are connected to the neuron elements with the different states, electric currents flow, and the electric conductance is degraded, which realize the modified Hebbian learning. As written in the previous sentences, it is utilized that the electric conductance is gradually degraded as the time goes.

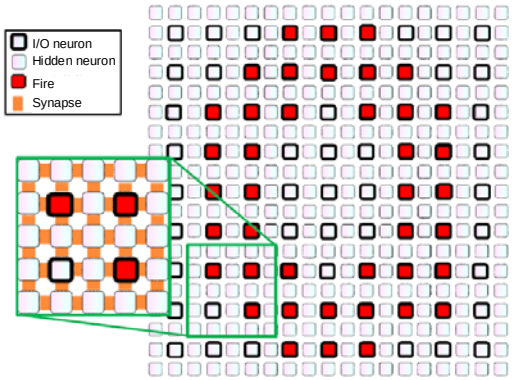

Fig. 8. Operation the cellular neural network.

3.2. Simulated Results

We assume the architecture and operation of the cellular neural network, model of the synapse elements, and simulate the letter recognition, as shown in Fig. 9. First, a pixel pattern corresponding to an input letter is inputted to I/O neurons as firing states. Afterword, the states in all neurons including the I/O and hidden neurons are calculated using the majority rule of the neighboring neurons with consideration of the synaptic connection strengths corresponding to the conductance of the oxide semiconductor synapses. Next, after the all the states in the neurons are settled, the modified Hebbian learning are done, namely, when the neurons connected to the synapse are in different states, the synaptic connection strength decreases based on the deterioration model of the oxide semiconductor synapses. After that, a pixel pattern slightly distorted from the input pattern is inputted to I/O neurons as firing states and immediately removed, and it is checked whether the output pattern from the I/O neurons becomes the original inputted pattern. Finally, these procedures are repeated for multiple input patterns.

```
#define DS 1 /* degradation speed */
/* horizontal synapse */
for (y=0; y<NWIDTH; y++) {
 for (x=0; x<NWIDTH-1; x++) {
  HCURR[y][x]=(oNEURON[y][x+1]-oNEURON[y][x])/HSYNAPSE[y][x];.. ※8
  HCSUM[y][x] += abs(HCURR[y][x]);  ..... ※9
  HSYNAPSE[y][x] = DS*HCSUM[y][x] + HSYNAPSEinit; ..... ※10
} }
/* vertical synapse */
for (y=0; y<NWIDTH-1; y++) {
 for (x=0; x<NWIDTH; x++) {
  VCURR[y][x]=(oNEURON[y+1][x]-oNEURON[y][x])/VSYNAPSE[y][x];.. ※8
  VCSUM[y][x] += abs(VCURR[y][x]);  ..... ※9
  VSYNAPSE[y][x] = DS*VCSUM[y][x] + VSYNAPSEinit; ..... ※10
} }
#define MAGNI 3.3 /* max voltage */
/* horizontal link */
for (y=0; y<NWIDTH; y++) {
 for (x=0; x<NWIDTH-1; x++) {
  HCURR[y][x] = (oNEURON[y][x+1] - oNEURON[y][x])/HSYNAPSE[y][x];
} }  ..... ※11
/* vertical link */
for (y=0; y<NWIDTH-1; y++) {
 for (x=0; x<NWIDTH; x++) {
  VCURR[y][x] = (oNEURON[y+1][x] - oNEURON[y][x])/VSYNAPSE[y][x];
} }  ..... ※11
for (y=0; y<NWIDTH; y++) {
 for (x=0; x<NWIDTH; x++) {
  int N, S, W, E;
  if(y>0)       N = -VCURR[y-1][x  ] ; else N=0;  ..... ※12
  if(y<NWIDTH-1) S =  VCURR[y  ][x  ] ; else S=0;  ..... ※13
  if(x>0)       W = -HCURR[y  ][x-1] ; else W=0;  ..... ※14
  if(x<NWIDTH-1) E =  HCURR[y  ][x  ] ; else E=0;  ..... ※15
  nNEURON[y][x] = oNEURON[y][x]+(N + S + W + E) ;  ..... ※16
  if ( step ≠ STEPS-1 ) {
   if (nNEURON[y][x] < 0    ) nNEURON[y][x] = 0;
   if (nNEURON[y][x] > MAGNI ) nNEURON[y][x] = MAGNI;
  }  ..... ※17
  else {
   if(nNEURON[y][x]≤(MAX/2)) nNEURON[y][x] = 0;
   if(nNEURON[y][x]> (MAX/2)) nNEURON[y][x] = MAX;
} } }  ..... ※18
memcpy(oNEURON, nNEURON, sizeof(nNEURON));
```

Fig. 9. Simulation program of the cellular neural network.

The degradation map of the synapse elements is shown in Fig. 10. Here, 9×9=81 I/O neuron elements corresponds to the image pixels. Number letters, 0 and 1 are learned. The red grayscale indicates the degradation degree of the electric conductance of the synapse elements.

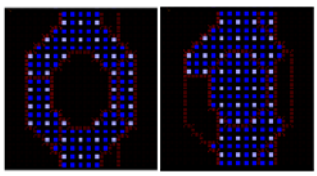

Fig. 10. Degradation map of the synapse elements.

The simulation results of the letter recognition are shown in Fig. 11. After a while of the learning, even if distorted letters are inputted, they are revised, and corrected letters are outputted. This is again a part of procedure of the letter recognition with pattern matching. In conclusion, this cellular neural network can recognize hand-written letters.

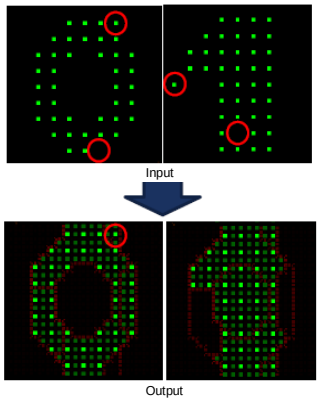

Fig. 11. Simulation results of the letter recognition.

3. Comparison and Discussion

The comparison of the Hopfield and cellular neural networks is in Fig. 12. It is found that the Hopfield neural network has higher performance for all number of the distorted pixels and number letters of 0 and 1. This is also because it has more complexity. However, the cellular neural network has a simple structure. It can be concluded that there is a trade-off between the high performance and simple structure [15].

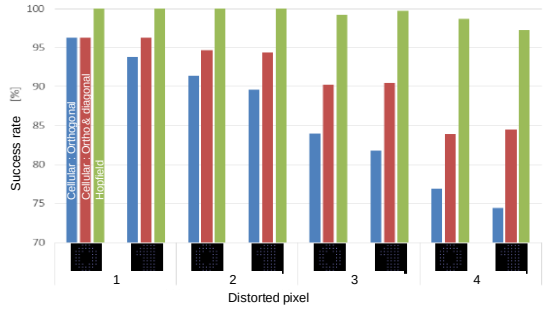

Fig. 12. Comparison of the Hopfield and cellular neural networks.

Acknowledgments

We would like to thank ROHM Semiconductor, Yazaki Memorial Foundation for Science and Technology, Support Center for Advanced Telecommunications Technology Research Foundation, and KOA Corporation.

References

[1] J. E. Dayhoff, *Neural Network Architectures, An Introduction*, Van Nostrand Reinhold, 1990.

[2] R. Hecht-Nielsen, *Neurocomputing*, Addison-Wesley Reading, 1990.

[3] S. Becker and G. E. Hinton, "Self-organizing neural network that discovers surfaces in random-dot stereograms," *Nature*, vol. 355, pp. 161-163, 1992.

[4] J. V. Stone, N. M. Hunkin, and A. Hornby, "Neural-network models: Predicting spontaneous recovery of memory," *Nature*, vol. 414, pp. 167-168, 2001.

[5] T. Kasakawa, H. Tabata, R. Onodera, H. Kojima, M. Kimura, H. Hara, and S. Inoue, "An artificial neural network at device level using simplified architecture and thin-film transistors," *IEEE Trans. Electron Devices*, vol. 57, pp. 2744-2750, 2010.

[6] M. Kimura, T. Miyatani, Y. Fujita, and T. Kasakawa, "Apoptotic self-organized electronic device using thin-film transistors for artificial neural networks with unsupervised learning functions," *Jpn. J. Appl. Phys.*, vol. 54, 03CB02, 2015.

[7] M. Kimura, Y. Fujita, T. Kasakawa, and T. Matsuda, "Novel architecture for cellular neural network suitable for high-density integration of electron devices - learning of multiple logics -," *ICONIP 2015*, pp. 12-20, 2015.

[8] M. Kimura, N. Nakamura, T. Yokoyama, T. Matsuda, T. Kameda, and Y. Nakashima, "Simplification of processing elements in cellular neural networks - working confirmation using circuit simulation -," *ICONIP 2016*, pp. 309-317, 2016.

[9] M. Kimura, R. Morita, S. Sugisaki, T. Matsuda, T. Kameda, and Y. Nakashima, "Cellular neural network formed by simplified processing elements composed of thin-film transistors," *Neurocomputing*, vol. 248, pp. 112-119, 2017.

[10] M. Kimura and T. Matsuda, "Neuromorphic application of oxide semiconductors," *ECS Trans.*, vol. 79, pp. 169-175, 2017.

[11] M. Kimura, H. Nakanishi, N. Nakamura, T. Yokoyama, T. Matsuda, T. Kameda, and Y. Nakashima, "Simplification of processing elements in cellular neural network," *J. Electrical Engineering and Electronic Technology*, to be published.

[12] T. Kameda, M. Kimura, and Y. Nakashima, "Letter reprodution simulator for hardware de-sign of cellular neural network using thin-film synapses - crosspoint-type synapses and simulation algorithm -," *ICONIP 2016*, pp. 342–350, 2016.

[13] M. Kimura, Y. Koga, T. Matsuda, and Y. Nakashima, "Characteristic analysis of IGZO thin films using planar and stacked devices - evaluation of electrical resistivity and current density," *IDW '16*, pp. 398-399, 2016.

[14] T. Kameda, M. Kimura, and Y. Nakashima, "Letter reprodution simulator for hardware design of cellular neural network using thin-film synapses," *NOLTA 2016*, pp. 40-43, 2016.

[15] T. Kameda, M. Kimura, and Y. Nakashima, "Neuromorphic hardware using simplified elements and thin-film semiconductor devices as synapse elements - simulation of Hopfield and cellular neural network -," *ICONIP 2017*, to be presented.