Additive Noise Effect on Ramp Function Neuron

Shinsuke Kuroda and Daisuke Maruyama and Hiroomi Hikawa

Oita University
700 dannoharu, Oita, Japan
Phone:+81-97-554-7875, Fax:+81-97-554-7886
Email: kuroda@csis.oita-u.ac.jp

Abstract

In this paper, a new type of pulse mode neuron is proposed. In the proposed neuron the ramp function activation function is made to be differentiable by adding random noise as a form of random number to the neuron input. The additive random number to the input of the neuron makes the activation function smooth thus differentiable, which is suitable for back propagation (BP) learning. The feasibility of the neuron is tested by computer simulation. The simulation results show that the proposed neuron has smooth differentiable activation function, and multilayer neural network (MNN) with the proposed neuron has better learning capability than MNN with ramp function neuron.

1. Introduction

Although most applications of neural networks are carried out using software simulators, many potential applications require high-speed network implemented in custom hardware, which can fully use the inherent parallelism embedded in neural network dynamics. Back propagation (BP) algorithm is the most widely used training algorithm for multilayer neural network (MNN) architecture. The learning performance of neural network is heavily affected by the characteristic of the activation function. As the derivative of the activation function plays an important role in the BP learning, the activation function should be differentiable.

One of the effective approaches for the hardware implementation of neural networks is pulse mode architecture. This paper proposes a new pulse mode neuron with differentiable activation function. By adding random noise to a ramp function neuron, the ramp function is made to be differentiable. The noise is generated by linear feedback shift register (LFSR) as a form of random number. The additive random number to the input of the neuron makes the activation function smooth thus differentiable. The feasibility of the neuron is tested by computer simulation.

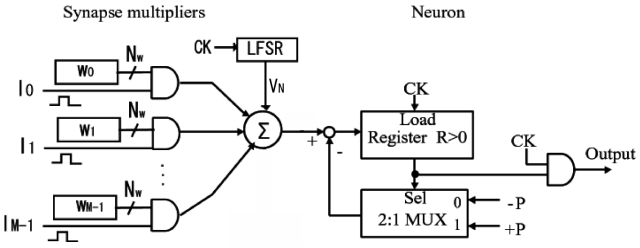


Figure 1: Proposed neuron with synapse weights

2. Proposed neuron with additive noise

2.1. Synapse multiplier[1]

The block diagram of the proposed neuron with synapse multipliers is depicted in Figure 1. Input signals $I_0 \sim I_{M-1}$ are multiplied with synapse weights $w_0 \sim w_{M-1}$. The input signal is given in pulse form whose pulse density or frequency is proportional to signal level. In the proposed system, multiplication is realized by simple AND gates. Each weight is given in N_w -bit fixed point format and each synapse multiplication is carried out by N_w AND gates.

Obviously the proposed synapse multiplier has an advantage in the circuit size. Additionally the proposed multiplier accepts any weight value that can be represented in fixed point binary format and the precision of the weights can be extended easily by increasing the number of bit-length of the weights.

2.2. Pulse mode neuron

Proposed pulse mode neuron consists of adders, a register, a 2:1 multiplexer, and the LFSR.

First, the sum of weight values from AND gates are calculated and added to the register. While the sum is accumulated in the register, $+P$ is subtracted from the register in case that the content of the register is positive, and $-P$ is subtracted, if it is not so. Therefore, if

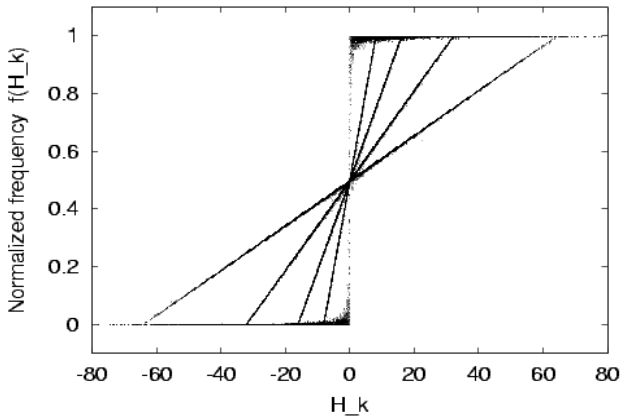


Figure 2: Nonlinear function of the proposed neuron (non-additive noise), P=0, P=8, P=16, P=32, P=64

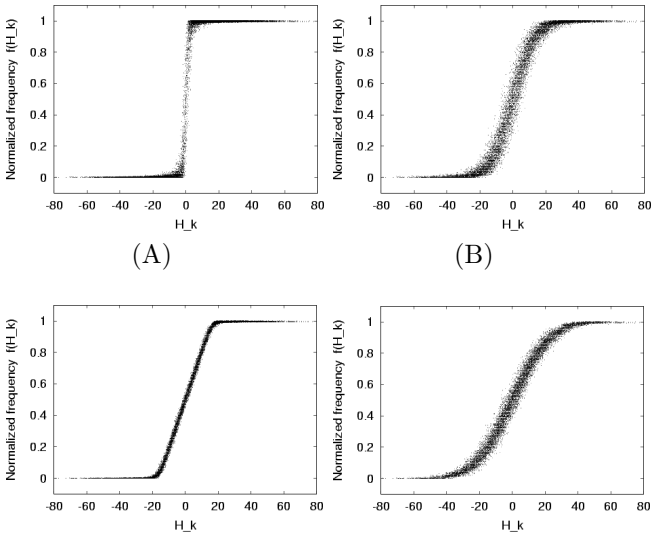


Figure 3: Additive noise effect on neuron's characteristic, (A)P=0(V_{MAX}=20), (B)P=0(V_{MAX}=60), (C)P=16(V_{MAX}=20), (D)P=16(V_{MAX}=60)

the input level is very small, the content of the register converges to zero. The output pulse is generated when the content of the register is not negative. When average weighted sum of the input signals is less than $-P$, neuron output is 0.0 because the content of the register is always negative. The average weighted sum greater than $+P$ makes the register always positive and the output is 1.0. When $-P \leq H_k^{(s)} \leq +P$, the content of the register becomes positive or negative alternately. Hence, the activation function of the neuron is ramp function, which is written as[2]

$$f(x) = \begin{cases} 1.0 & x > P \\ x/(2 \times P) + 0.5 & -P \leq H_k^{(s)} \leq +P \\ 0.0 & x < -P \end{cases} \quad (1)$$

The content of the register R is bounded between $-R_{MAX}$ and $+R_{MAX}$ to avoid overflow.

Although the derivative of the activation function is used in the BP algorithm, the ramp function is not differentiable. In the proposed neuron, ramp function is made to be differentiable by additive random noise to the neuron input. The noise is generated by LFSR as a form of random number V_N and is added to the input of the ramp function neuron. The range of V_N is defined as follows.

$$-V_{MAX} \leq V_N \leq +V_{MAX} \quad (2)$$

The additive random number to the input of the neuron makes the activation function smooth thus differentiable.

2.3. Neuron characteristic

To test the proposed neuron and synapse multiplier, computer simulations are carried out.

The relation between H_k and $f(H_k)$ of the proposed neuron is depicted in Fig. 2. H_k is weighted sum of input signals and $f(H_k)$ is normalized frequency of the neuron output pulse signal. The number of output pulse are counted during observation period T_p. T_p is defined by the number of reference clock pulses during the observation period. If the neuron pulse count equals T_p, then the normalized frequency is $f(H_k) = 1.0$. As additive noise is not added, the measured activation functions in Fig. 2 are ramp functions. Also the simulation results show good agreements with the theoretical values calculated from (1).

Next simulation measures the additive noise effect on ramp function neuron. Fig. 3 shows the activation functions with additive noise. As shown in the figure, additive noise makes the neuron's activation function more like Sigmoid function. As the maximum random number V_{MAX} increases, activation function becomes smoother and thicker. Even though the $+P/-P$ operation is absent $(P = 0)$, neuron's activation function becomes Sigmoid-like function. However, as Fig. 4 shows, the neuron's activation function is also governed by R_{MAX} and observation period T_p. The smoothness of activation function is changed by the R_{MAX} and the thickness is decided by T_p.

3. Pulse mode MNN with on-chip learning

3.1. System configuration

To provide the on-chip learning, the BP algorithm is

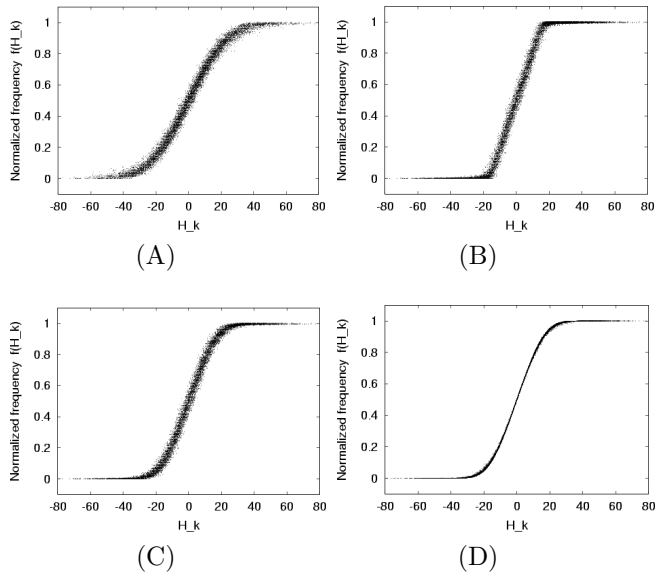


Figure 4: Effect of maximum value of register and the number of input-output pulse on neuron's characteristic, (A)R_{MAX}=40, (B)R_{MAX}=400, (C)T_p=256, (D)T_p=65536

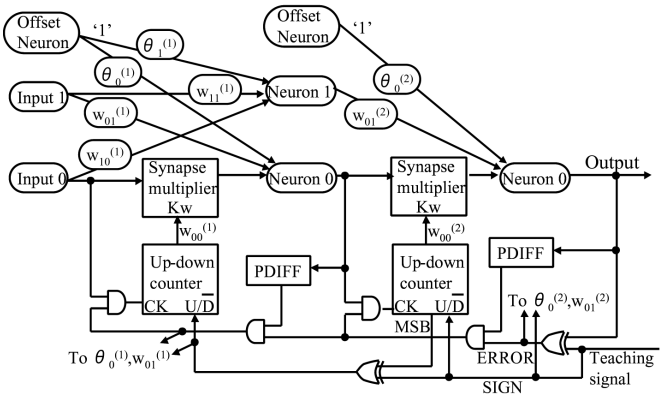


Figure 5: MNN configuration

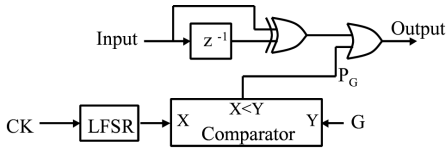


Figure 6: Pulse differentiator

modified to have pulse mode operation for the effective hardware implementation.

System configuration of the MNN architecture with on-chip learning[2] is depicted in Fig. 5. The lower half of Fig. 5 is the on-chip learning circuit, i.e., the hardware implementation of the BP algorithm. A pulse differentiator (PDIFF) is employed to generate the derivative $f'(H_k)$. The block diagram of PDIFF is shown in Fig. 6. The differentiator gives output pulse every time it finds head and tail of pulse stream. It is reported that the learning performance of the MNN with BP algorithm can be improved by adding an offset to $f'(H_k)$. Pseudo random sequence P_G is added to $f'(H_k)$ by using OR gate.

3.2. Simulation

Using the MNN with the proposed neuron and synapse multipliers, the performance of the proposed neuron is verified by computer simulation.

3.2.1. Forward operation

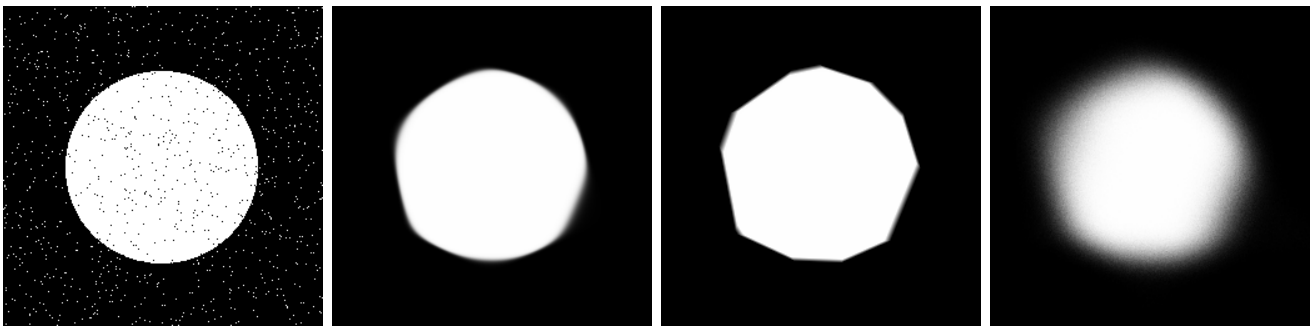
This simulation tests the forward operation of the MNN. To obtain the synapse weights, BP learning is performed on host computer. Sigmoid function that is similar to the measured activation function is used for the learning. After the training, trained weights are used to configure the MNN.

The MNN is tested with 2-D binary classifying problem. Given x,y coordinate of a point within the square region bounded by $0 \leq x < 256$ and $0 \leq y < 256$, the MNN is trained to recognize if the given point lies inside or outside of circular region. The network output is zero if the point lies outside, one if it lies inside.

The network has two inputs with an offset neuron, a single hidden layer that contains twelve neurons with an offset neuron, and single output neuron. Training data set is composed of 1,024 coordinates randomly selected from the complete data set (65,536 coordinates). After the training, all 65,536 coordinates are fed to the MNN to measure the response. Fig. 7 shows original pattern, training data set and network's output patterns. Small dots in Fig. 7(A) are the coordinates used for the training. Output level is represented by gray level. While the target shape is smooth circular shape, the output response of MNN with ramp function looks like a polygonal shape. However, the response of MNN with the proposed neuron seems to be smoother because the noise makes the activation function smoother.

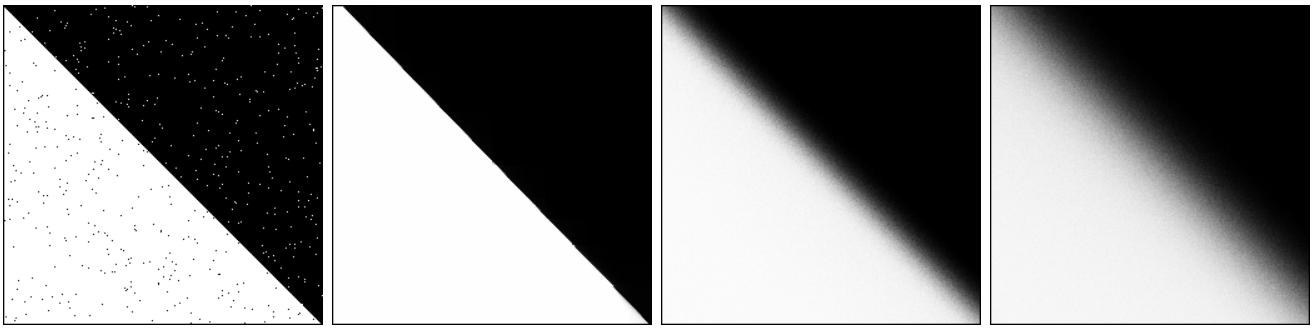
3.2.2. On-chip learning

In this section, simulation results of the on-chip learning are presented. The on-chip learning capability is tested by 2-D binary classifying problem. The proposed MNN is trained with the teaching data that consists of



(A) (B) (C) (D)

Figure 7: 2-D binary classifying problem, (A)Training data, (B)MNN output(Sigmoid), (C)MNN output(ramp function), (D)Output of MNN with proposed neuron(V_{MAX}=40)



(A) (B) (C) (D)

Figure 8: Learning of 2-D binary classifying problem, (A)Training data, (B)MNN output(ramp function), (C)Output of MNN with proposed neuron(V_{MAX}=80), (D)Output of MNN with proposed neuron(V_{MAX}=160)

512 coordinates. The teaching data and the response of the MNN are shown in Fig. 8. Figures show that the output plots resemble very much the target shape even though the training patterns are incomplete. Obviously the boundary of the output plots of Fig. 8(B) shows error in the upper left portion of the figure. On the other hand, although the boundary of the output plots of Fig. 8(C) and (D) are vague, the MNN is successfully trained to classify the region. As noise becomes large, the network response becomes vague.

4. Conclusions

In this paper, a new pulse mode neuron with differential activation function is proposed. By adding random number noise to ramp function neuron, differentiable activation function is realized. The feasibility of the proposed architecture was verified by computer simulations. The results show that activation function becomes smoother by additive random noise to the neuron input. Also the MNN with on-chip learning which uses the proposed neuron has better learning capability compared to the ramp function neuron.

References

[1] H.Hikawa, "Digital Pulse Mode Neural Network with Simple Synapse Multiplier," Proceedings of 2001 IEEE International Symposium on Circuits and Systems (IEEE ISCAS 2001), pp. III-596–III572, 2001.5.

[2] H.Hikawa, "Pulse Mode Multilayer Neural Network with Floating Point Operation and On-chip Learning," Proceedings of 2000 IEEE-INNS-ENNS International Joint Conference on Neural Networks (IEEE-INNS-ENNS IJCNN 2000), pp. II-71–II-76, 2000.7.