

VLSI Implementation of Neural Network with Stochastic Architecture

鄭 德 鎮* · 韓 相 旭**

(Duck-Jin Chung · Sang-Wook Han)

Abstract - Using random pulse stream, a number can be transformed to the pulse stream with the probability value. So the digital value are computed by simple digital gates. Thus it will be possible to build a small and strong noise immunity processing element. We propose a faster convergence algorithm using a new method for better performance of Random Number Generator(RNG) and the nonlinear transfer function(Sigmoid function)in this paper. And a feedback circuit were fitted for pulse stream in this paper. We proposed method is simulated with C program language and conformed by circuit implementation. Finally a system for hand written number recognition is constructed by FPGA and its performance verified.

Key Words : NEURAL, STOCHASTIC, RANDOMPULSE

1. 서 론

기존의 신경회로망의 하드웨어 구현 방법에는 디지털, 아나로그 그리고 혼합형의 방법을 세 가지 방법을 이용하고 있다.^[1] 그러나 디지털 방법의 경우는 노이즈에 강한 장점이 있으나, 시냅스에 요구되는 multiplier와 비선형 전달함수의 구현에 많은 수의 Gate를 요구하게 되는 단점이 있다. 아나 로그 방법의 경우에는 수행 속도가 빠르고 단위소자의 크기를 작은 장점을 가지나, 노이즈에 민감하여 분해능이 큰 시스템을 구현할 수 없는 단점이 있다. 또 혼합형의 경우도 디지털과 아나 로그를 한 chip에 구현해야 하는 단점이 있다. 따라서 본 논문에서는 확률 값을 이용하여 간단한 디지털 회로로 pseudoanalog 계산을 수행함으로써,^[5] 노이즈에 강하고 단위소자의 크기를 작게 할 수 있는 stochastic 구조를 이용하여 많은 수의 뉴런과 시냅스를 요구하는 필기체 숫자 인식을 위한 하드웨어 구조를 설계하고 그 동작을 확인한다.

2. pulse stream을 이용한 신경망의 구현

pulse stream을 사용하는 구조의 연산 방법은 수치가 일련의 line을 통과하는 연속된 randompulse의 평균율에 의해 주어진다 가정 하에서 출발한다.^[3] 이러한 가정 하에서는 수치를 확률 값인 pulse stream으로 전환할 경우에는 수학적 연산이 간단한 논리 회로를 이용해 수행될 수 있게 된다는 원리를 이용한 것이다. 따라서 신경회로망의 설계시 일반 디지털 회로를 이용할 때 많은 Gate를 요구하는 비선형 전달함수(sigmoid)와 시냅스의 구현이 간단한 Gate로 대처가 가능하게 된다. 따라서 같은 chip 영역에 더 많은 수의 단위소자의 구현이 가능하게

됨으로 많은 수의 곱셈기와 비선형 전달함수를 요구하는 신경 회로망에 적합한 구조라 할 수 있다. 이러한 stochastic 구조를 사용할 경우에는 다음과 같은 장점이 있다.^[4]

- (1) pseudoanalog 계산이 random pulse하에서 간단한 디지털 논리를 이용하여 쉽게 수행된다.
- (2) 간단하고 작은 크기의 stochastic 회로는 massive parallelism로 수행이 가능하다.
- (3) 디지털 stochastic 신호는 아나 로그 회로보다 chip간에 연결에 있어 기생 capacitor와 Noise에 강함으로 여러 chip을 이용한 방대한 시스템의 개발이 가능하다.

3. 시스템의 구현

3.1 난수 발생기의 문제점

기존의 pulse stream의 방법에서 random pulse를 만들기 위해 세 가지 방법^[3]을 이용하고 있다. 첫째는 아나 로그의 방법을 이용하는 것이다. 이 방법은 회로의 크기가 매우 작고 양질의 random pulse를 얻을 수 있으나, 아나 로그를 이용하기 때문에 일반적인 디지털 회로에서는 구현할 수가 없게 된다. 두 번째 방법은 look Up table을 이용하는 것이다. 이 방법도 양질의 난수를 얻을 수 있으나, look Up table이 차지하는 chip의 영역이 매우 커져 구현이 어렵게 된다. 세 번째 방법은 현재 stochastic 구조에서 주로 사용하는 방법으로 linear feedback shift register (LFSR)를 사용하는 것이다.^[5] 이 방법은 LFSR이 표시할 수 있는 최대의 수가 2^n 임으로, 따라서 난수 발생기가 나타낼 수 있는 수의 종류도 2^n 이 된다.

그러나 기존의 RNG의 경우는 shift register의 최종타과 그 전단의 출력을 EX-NOR한 값을 첫 번째 register로 feedback 함으로써, 각 flip-flop에서 임의의 pulse를 발생하도록 하고, 이를 다시 EX-OR하여 난수를 발생하게 한다.^[5] 따라서 feedback의 특성에 의해 shift register의 초기값이 모두 '1'로 초기화되었을 경우에는 feedback되는 값이 항상 '1'이 됨으로, 발생하는 숫자를 변화시킬 수 없게 된다. 이로 인하여 RNG가

*正 會 員 : 仁荷大 工大 電子材料工學科 副教授 · 工博

**正 會 員 : LG半導體(株)

接受日字 : 1994年 9月 6日

1次 修正 : 1995年 11月 29日

2次 修正 : 1995年 12月 28日

발생할 수 있는 숫자의 최대 주기는 $2^n - 1$ 이 된다. 그러나 위와 같은 feedback 방법은 shift register가 "0 0 0 0 0"의 초기화 상태에서 random-pulse를 발생하기 시작하였을 경우에 각 shift register의 값이 "0 1 0 1..."의 값을 갖게 되면 "0"만을 feedback이 하게 됨으로 그 주기가 곧 끝나게 된다. 따라서 주기가 flip-flop의 수에 따라 일정한 규칙성을 갖지 못하게 되고, 또 LFSR의 주기가 짧아지게 됨에 따라 LFSR을 이용한 난수 발생기가 나타낼 수 있는 수치들 중에 반복주기 내에 있지 못한 수에 의하여, 발생되는 random pulse가 오차를 갖게 되는 원인이 되며, 또 correlation이 커지게 되는 원인이 되었다.

3.2 문제점의 해결방안 및 성능향상을 위한 제안

3.2.1 LFSR을 이용한 난수 발생기의 개선

표 1은 shift register의 flip flop 수에 따른 반복주기를 C - 언어 모의실험한 결과를 나타낸 것이다. 여기서 실제 반복주기가 두개 이상인 것은 난수 발생기의 feedback특성에 의해 최대 주기보다 짧은 주기를 갖는 여러 개의 난수 주기의 크기를 나타난 것으로, 각 주기에 나타나는 난수는 서로소인 값을 갖게 된다. 이에 따라서 실제 발생되는 난수는 LFSR이 나타낼 수 있는 모든 수를 포함하지 못하게 된다.

따라서 본 논문에서는 다음과 같은 방법을 제안한다. 여러 주기가 존재하는 shift register의 출력에서 각 주기의 끝을 감지하여 강제로 다음 주기로 전환하도록 하여 발생되는 난수를 shift register가 나타낼 수 있는 최대의 주기로 확장시키는 것이다. 이것을 예를 들어 설명하면 다음과 같다. 먼저 shift register의 내용을 "0 0 0 0 0"로 초기화시킨 후 feedback을 수행한다. 이후에 shift register의 각 flip-flop의 출력을 관찰하여 그 출력이 "0 0 0 0 1"의 출력을 내게 되면, shift register의 각 flip-flop의 값을 다음 주기의 초기 값인 "0 0 0 1 1"으로 변환한 후 feedback을 수행하게 한다. 같은 방법으로 이 주기의 끝에서 다시 초기화를 하여 반복수행 하게 함으로써 난수 발생가능한 전구간에서 난수를 발생시키도록 할 수 있다.

그림 1은 앞의 예를 그림으로 나타낸 것이다.

표 1 F/F의 수와 난수의 주기와의 관계

Table 1 The relationship between number of F/F and cycle of random number

# of F/F	최대 반복 주기	실제 반복 주기 (%)
4	$2^4 - 1$	100
5	$2^5 - 1$	67, 33
6	$2^6 - 1$	100
7	$2^7 - 1$	100
8	$2^8 - 1$	24, 24, 24, 24
9	$2^9 - 1$	14
10	$2^{10} - 1$	86, 12
11	$2^{11} - 1$	74, 24
12	$2^{12} - 1$	79, 11
13	$2^{13} - 1$	96
14	$2^{14} - 1$	71, 23
15	$2^{15} - 1$	100

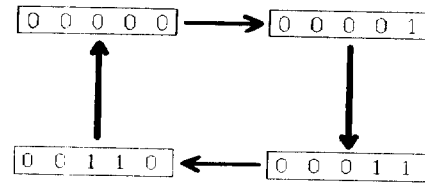


그림 1 flip-flop의 출력값 변화 과정
Fig. 1 Output sequence of flip-flop

3.2.2 학습 속도의 향상을 위한 제안

오류역전파 학습의 알고리즘은 feed forward를 수행한 후 지도출력과 실제출력간의 차를 식(3)을 사용하여 feedback 시키고 이 feedback된 값을 이용하여 가중치의 변화를 하여, 실제출력이 지도출력과 차가 기준 이하가 될 경우에 학습을 종결시킨다[4].

$$N\text{-th hidden Neuron Feedback } \Sigma = \sum_{k=1}^K \delta_{ok} w_{kj} \quad (1)$$

$$\delta_{ok} = (d_k - o_k)(1 - o_k)o_k u[(\sigma_{th} - (d_k - o_k))^2] \quad (2)$$

$$u[x] = \text{unit step function } u[x] = 1 \quad x \geq 0 \\ u[x] = 0 \quad x < 0$$

$$\delta_{yj} = y_j(1 - y_j) \sum_{k=1}^K \delta_{ok} w_{kj} \quad (3)$$

위와 같은 학습 방법은 변화시킬 필요가 없는 가중치의 값까지 변화를 시키게 된다. 또 대부분의 학습 알고리즘에서 사용하는 학습의 종결방법인 출력층 뉴런의 오차들의 합을 이용하는 방법은, 출력층의 어느 한 뉴런에 오차를 집중시키는 경우를 발생시킬 수 있다.

따라서 본 논문에서는 feedback의 수행을 할 때 출력층 뉴런에 feedback threshold를 도입하여, 지도출력과 실제출력간의 오차가 이 threshold 이하의 값을 가질 경우에는 그 feedback을 제한하는 방법을 제안한다. 이 방법을 사용함으로써 불필요한 feedback을 줄이고, 최종적으로 학습이 완료됐을 때 출력층의 오차가 각 뉴런에 고르게 분포할 수 있는 효과를 얻을 수 있다. 또 학습의 종결에 사용되는 출력층 오차의 합의 계산에 있어 출력층 뉴런에서 threshold이하의 오차를 출력층 뉴런 오차의 합에서 제외시키는 방법을 이용하여 고른 오차의 분포를 갖으면서 학습에 걸리는 시간을 단축할 수 있는 방법을 제안한다.

3.3 인식을 위한 학습 프로그램의 개발

3.3.1 학습 알고리즘

학습을 위한 컴퓨터 프로그램은 C-언어(C-programming language)를 이용하여 작성하였다. 이 프로그램의 학습 알고리즘은 앞에서 제시한 feedback threshold를 갖는 역전파 알고리즘(Back-propagation Algorithm)을 사용했다. 또 feedback threshold이하의 출력층 뉴런의 오차를 오차의 합에서 제거시키는 방법을 이용하여 학습에 걸리는 시간을 줄이면서 출력층의

각 뉴런에 오차가 고르게 분포할 수 있는 방법을 사용하였다.

3.3.2 학습에 사용된 패턴의 구성방법

학습에 사용되는 패턴은 '0' .. '9'의 숫자를 사용하였다. 각 수치는 컴퓨터의 graphic 프로그램 상에서 마우스를 이용하여 입력받고 이를 C - 언어로 제작된 프로그램을 이용하여 정규화 과정을 거친 후 사용하였다.

◆ 정규화 방법

그림 (2-a)와 같이 입력된 패턴의 네 끝점을 찾아 글자의 크기를 알아낸다. 이후에 얻어진 글자의 크기를 이용하여, 비례식을 이용하여 각 pixel의 위치를 6*8의 크기로 정규화 하도록 하였다. 학습과 재현을 위한 입력 패턴은 마우스를 통해 입력을 받아 위와 같은 방법으로 정규화를 시킨 후에 사용하였다.

◆ 재현을 위한 입력 패턴

재현을 위한 패턴은 7 명의 사람으로부터 각 숫자마다 5개씩의 패턴을 받은 후 이를 정규화 하여 사용하였다. 그림 2는 필기자로부터의 입력과 정규화된 입력을 비교하여 나타낸 것이다. 그림 2는 입력에 사용될 필기자로부터 입력 패턴의 예를 보인 것이다.

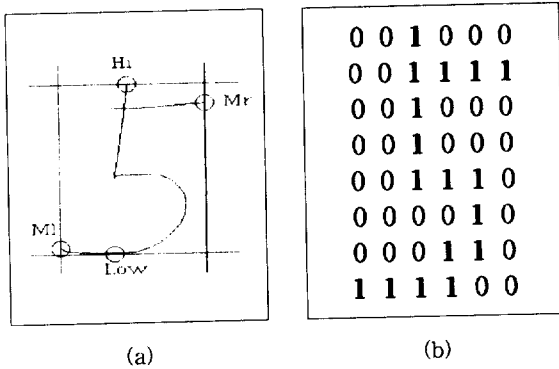


그림 2 입력과 정규화된 데이터의 비교
Fig 2 Input vs. Normalized Data

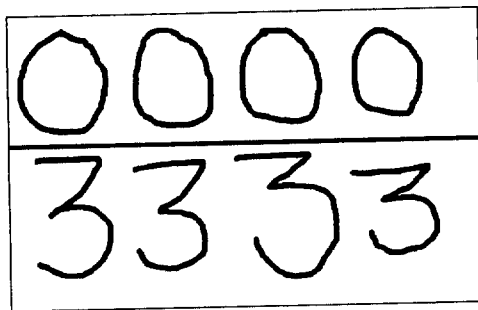


그림 3 입력 데이터의 예
Fig 3 Example of Input Data

3.3.3 필기체 숫자 인식을 위한 시스템의 구현

3.3.3.1 시스템의 구성 형태

48개의 입력과 20개의 은닉층 뉴런 4개의 출력층 뉴런으로 구성되어 있다. C-언어 프로그램에 의한 모의실험에서 재현(recall)동작시에 mapping이 가능한 최소의 Bit은 5 Bit 이었다. 즉 학습은 12bit으로 수행하고, 재현은 하드웨어 설계시에는 오

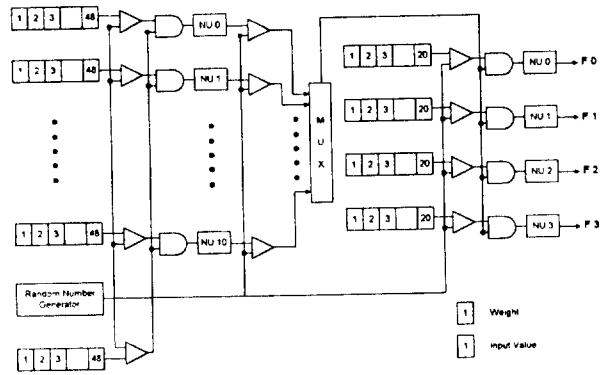


그림 4 재현 시스템의 구성도
Fig 4 Block Diagram of Recall System

차와 시스템의 크기를 감안하여 6 bit의 시스템을 구현하였다. 이 회로에 사용된 module수는 15,000 module정도로 구현이 가능하였다.

4. 모의실험

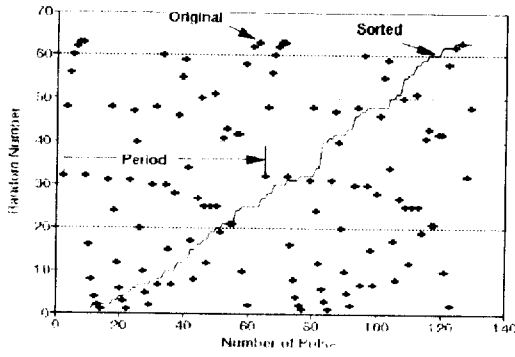
4.1 LFSR의 모의실험

다음은 기존의 방법을 이용한 RNG의 프로그램에 의한 모의 실험의 결과와 그 주기를 확장시켰을 때의 모의실험을 비교한 것으로, 이때 발생하는 난수를 십진수로 환산하여 나타낸 것이다. 선으로 표시된 것은 발생된 난수를 크기 순으로 sort하여 발생되지 않는 수를 알기 위한 것이고, 점으로 나타낸 것은 발생되는 난수를 발생 순서대로 표시한 것이다. 여기서 사용한 shift register는 8 bit shift register를 사용하였고, feedback을 고려하여 6 bit 출력을 표시했다.(시스템의 구현 시에는 학습 시에는 14 bit shift register를 사용하여 12 bit출력을 내도록 하였고, 재현 시에는 7 bit shift register를 이용하여 6 Bit출력을 내게 하였다.) 먼저 그림 5의 a)의 경우는 기존의 RNG를 프로그램으로 모의실험한 것으로 34에서 40사이의 값이 발생하지 않았고, 주기 또한 8 bit의 shift register의 주기를 모두 나타내지 못했다. b)의 그래프는 제안된 방법을 적용한 프로그램으로 모의실험한 것으로, 그 주기가 2 배로 확장된 것을 관찰할 수 있다.(이때 주기확장은 8 Bit shift register의 4개의 주기 중 2개의 주기만을 확장에 이용하였다.) 따라서 제안된 방법을 이용할 경우에는 난수의 주기가 확장됨으로 correlation에 관한 문제가 줄어들었고, 또 RNG가 고른 출력을 나타냄으로 발생하는 random pulse의 정확도도 향상될 수 있다.

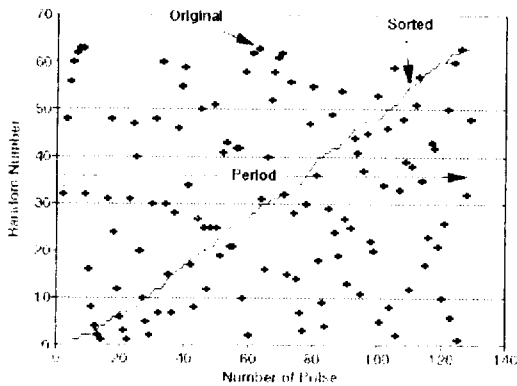
4.2 Feedback Threshold의 도입

그림 6은 출력 뉴런의 feedback을 기존의 방법을 사용할 때와 제안된 방법을 썼을 때의 수렴 속도를 비교하여 나타낸 것이다.

그림 6에서 modified 1의 경우는 제안된 방법을 사용하였으나 feedback threshold 이하의 오차도 출력층 뉴런의 오차 합에 포함시킨 경우로 곡선의 기울기가 완만해 지는 것이 수렴되어 가는 것을 표시한다. 이때 feedback threshold를 사용한 방법의 그래프의 경우에 그 기울기가 더 빠르게 완만하게 됨을 알 수 있다.



(a) 기존 모델^[5]



(b) 제안된 모델

그림 5 LFSR에서 발생하는 난수들
Fig 5 Random numbers generated by LFSR

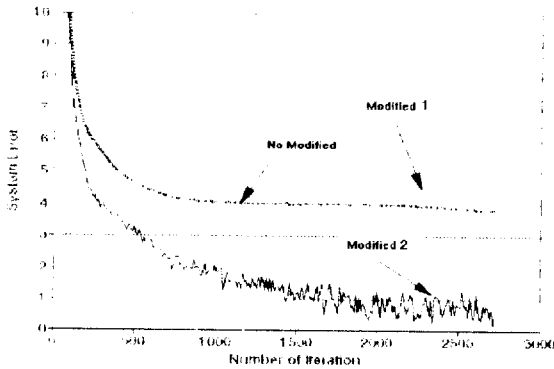


그림 6 두 알고리즘간의 수렴 속도의 차
Fig 6 Difference of convergence speed between to algorithm

Modified 2의 그래프는 제안된 feedback threshold방법을 모두 적용한 경우로써 feedback threshold 이하의 오차를 갖는 뉴런의 오차를 오차의 합에서 제외하여 수렴의 속도를 향상시킨 것이다. 이 경우 feedback threshold이하의 오차를 제외시킴으로써 출력층 뉴런의 오차를 기준치 이하로 고정시키면서 출력층 뉴런들의 오차의 합이 원하는 값을 가질 때, 그 수행을 종료시킨다. 따라서 기존의 방법처럼 출력층의 오차를 모두 더해 그 값을 학습의 종결에 사용할 때보다, 시스템

오차의 합은 더 큰 값을 갖으나 출력층의 각 뉴런에 분포하는 오차의 크기는, 기존의 방법보다 더 고르게 분포하게 됨으로 더 유리한 면을 갖는다. 따라서 오차의 합이 기존보다 더 큰 값에서 수행을 종결시킬 수 있음으로 학습의 속도 면에서 더 유리할 수 있다. 이때 최종 학습후의 출력층 뉴런의 오차의 크기를 보면 기존의 것이 0.00에서 0.17정도의 오차를 갖는 반면에, 제안된 방법은 0.00에서 0.07정도의 오차를 갖음을 확인할 수 있었다.

4.3 학습에 사용한 data

학습에 사용한 패턴은 컴퓨터에서 그림을 그리는데 사용하는 tool인 dirux paint 프로그램의 실행 상에서 마우스를 이용하여 입력을 받았다. 이 입력 패턴을 C - 언어로 제작된 프로그램으로 각 숫자의 윤곽만을 찾아 일정한 크기로 정규화만을 하였다. 학습시 입력된 패턴은 필기자와 무관한 학습을 하는 경우에는 재현에 사용한 패턴중 첫 번째 패턴을 골라 각 숫자마다 4개씩을 사용하였다. 또 필기자와 연관된 학습을 하는 경우에는 각 필기자의 필기 중 3번째 또는 4번째의 패턴을 사용하여 학습시켰다.

4.4 컴퓨터에 의한 모의 실험 결과

4.4.1 프로그램에 의한 모의실험 결과

프로그램은 두 가지 방법으로 제작하였다. 첫째는 학습과 재현을 할 수 있는 프로그램으로 일반적인 디지털 계산을 하는 방법을 이용한 것이다. 둘째는 pulse stream의 방법을 프로그램의 형태로 전환하여 구현한 것으로 재현만을 위한 시스템을 구현하였다. 따라서 학습은 첫 번째 프로그램으로 수행하였고 인식은 두 번째 프로그램을 이용하여 pulse stream을 이용한 재현 시스템의 동작을 완전하게 모의실험 할 수 있었다.

위의 표는 첫 번째 프로그램에 의해 학습을 수행하고 두 번째 프로그램에 의해 재현이 실행된 것으로 7명에게 각각의 숫자마다 5개씩의 숫자를 받아서 수행한 것이다. 표2-a)경우는 모든 필기자에 대해서 같은 학습 패턴을 사용한 경우로 각 필기자의 특성이 매우 다양하게 됨으로 좋은 결과를 얻지 못했다. 이는 각 layer의 node수가 제한 되어 있게 됨으로 많은 수의 입력 pattern에 대한 특성을 저장하는 각 node의 수가 부족하기 때문으로 생각 된다. 이는 더 많은 수의 node를 갖는 시경회로망의 모의실험 이러한 결과를 얻을 수 있었다. 표2-b)경우는 각 필기자에 따라 다른 학습 패턴을 사용한 경우로 이 경우에는 각 필기자에 대한 특성을 찾아내어 이를 학습을 통해 기억할 수 있는 특성을 나타내었다. 따라서 본 시스템은 가중치를 고정된 상태에서 사용하는 범용의 시스템보다는 구체적인 대상을 위한 필기체 숫자 인식에 적용하면 매우 효과적인 결과를 얻을 수 있을 것으로 생각된다.

4.4.2 FPGA로 구현한 회로의 모의실험 결과

위의 학습 패턴을 이용하여 제안된 알고리즘에 따라 제작된 학습 프로그램에 의하여 학습을 한 후에 학습된 가중치를 출력 받아 이를 인식 시스템의 가중치의 입력으로 사용하였다. 재현을 위한 패턴은 프로그램에서와 같이 필기자로부터의 필기를 정규화한 패턴을 입력으로 사용하였다.

[8] J. M. Zurada, Artificial Neural System, West publishing Company

[9] J. W. Suh and S. I. Chae, 'A Back- propagation Algorithm for Neural Networks Using Random Pulse Streams' Artificial Neural Network 2, pp.1035-1038, 1992

[10] B. R. Gaines, 'Uncertainty as a Foundation of Computational Power in Neural Networks' IEEE IJCNN Vol 3, pp.51-57, 1987

[11] E. W. Lee, S. I. Chae, 'An Alternative RPT Neuron Model :Introducion a Nonlinearity into RPT Neuron Model', JCEANF'92 pp.530-535, 1992

[12] Krister Valtonen et al. 'Pulse-Stream Neural Networks and Rainforcement Learning', IEEE IJCNN Vol2,pp.267-272, 1990

[13] Dziem Nguyen and Fred Holt, 'Stochastic Processing in a Neural Network Application', IJCNN Vol 3, pp.281-291, 1987

[14] Gamze Erten, R. M. Goodman, 'A Digital Neural Network Architecture Using Random Pulse Trains', IEEE IJCNN Vol 1 pp.190-195, 1992

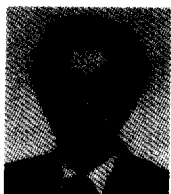
[15] Jouni Tomberg and Kimmo Kaski, 'VLSI Archicture of the Self-organizing Neural Network Using Synchronous Pulse Density Modulation Technique' Artificial Neural Network 2, pp.1431-434, 1992

[16] Jouni Tomberg and Kimmo Kaski, 'An Effective Traoninf Method for Fully Pulse-Density Modulated Neural Network Architecture', Proc. IEEE ISCAS'91 Singapore, June pp.11-14, 1991

[17] J. R. Beerhold, M. Jansen, R. Eckmiller, 'Pulse-Processing Neural Net Hardware with Selectible Topology and Adaptive Weights and Delays', IEEE IJCNN, Vol 2, pp.569-574, 1990

[18] Mark Bagula, 'High-Speed Pulse mode Neural Network with EE Weight and On-chip Learning(PMNN)', IEEE IJCNN Vol 2, pp.A-927, 1992 (July)

자 작 소 개



정 덕 진 (鄭 德 鎭)

1948년 2월 8일생. 1970년 서울대 공대 전기 공학과 졸업. 1984년 미국 Utah. State University 전기공학과 졸업(석사). 1988년 미국 University of Utah 전기공학과 졸업(공학박). 1978년~1980년 오토론 상품 개발부 차장.

1980년~1989년 전자통신연구소 책임연구원. 현재 인하대 공대 전자재료공학과 부교수



한 상 옥 (韓 相 旭)

1970년 4월 19일생. 1992년 인하대 공대 응용물리학과 졸업. 1994년 인하대 대학원 전자재료공학과 졸업(석사). 1994년~현재 LG반도체 PCMCIA ATA Controller 설계팀 근무중