

# 이동 에이전트에 의한 학습세션 병렬 인공지능망 시뮬레이터

강태원 조용만<sup>0</sup> 김미숙  
강릉대학교 컴퓨터공학과 대학원  
twkang@kangnung.ac.kr, cs9235<sup>0</sup>@lycos.co.kr, mikiss@hanmir.com

## Training Session Parallel ANN Simulator using Mobile Agent

Taewon Kang Yongman Cho<sup>0</sup> Misuk Kim  
Dept. of Computer Science & Engineering, Kangnung National University

### 요 약

이 연구는 이동 에이전트 시스템에 기반한 가상의 병렬분산 컴퓨팅 환경에서 병렬로 수행되는 인공지능망 시뮬레이터를 구현하는 것을 목적으로 하며, 학습세션 수준에서 병렬로 학습하는 병렬 인공지능망 시뮬레이터의 성능을 대표적인 벤치마크 문제인 NetTalk을 대상으로 평가한 결과, 개발한 시뮬레이터가 상당히 효과적임을 알 수 있다.

### 1. 연구배경 및 목적

신경망은 컴퓨터가 사람의 학습 기능을 갖게 하기 위해서 고안된 것으로, 문자, 영상, 음성인식뿐 아니라 컴퓨터 게임, 로봇제어 등과 같은 다양한 분야로 응용 범위가 확대되어 가는 시점에 있다. 그러나 신경망 응용시스템의 개발에는 많은 시간과 비용이 필요하다. 따라서 이들 시스템을 개발하는데 필요한 시간과 비용을 줄이기 위한 연구가 필요하게 되었다.

신경망 모델은 그 자체에 병렬성이 내포되어 있으므로 신경망을 병렬로 구현하려는 것은 매우 자연스러운 일이다. 따라서 최소의 비용으로 최대의 효과를 얻기 위해서, 인터넷을 돌아다니며 실행되는 이동 에이전트를 이용하여 병렬·분산 컴퓨팅 환경을 실현하는 것이 가능하다. 국내의 경우 기술적인 여건 역시 충분히 갖추어져 있고, 인터넷 인프라가 세계적 수준인 국내 환경을 고려한다면 고가의 병렬컴퓨터가 아닌 인터넷 환경에 기반 하여 신경망을 병렬로 구현한다는 것은 의의가 크다고 할 수 있다.

신경망을 병렬로 구현하는 경우 병렬화의 대상에 따라 학습세션(training session) 병렬화, 학습데이터(training example) 병렬화, 계층(layer) 병렬화, 노드(뉴런) 병렬화 및 가중치(weight) 병렬화 5가지로 나뉜다[1,2,3]. 이 논문에서는 대표적 이동에이전트 플랫폼인 aglets과 java언

어에 기반으로 하여, 인터넷 상에서 학습세션을 병렬로 처리하는 학습세션 병렬 인공지능망 시뮬레이터를 개발하고 평가한다.

신경망 학습을 위해서는 여러 가지 매개변수를 정해주어야 하며 이러한 매개변수 값들은 신경망의 학습에 영향을 준다. 하지만 특정 문제에 가장 적절한 매개변수를 정하는 규칙은 없다. 따라서 신경망의 학습과 성능을 최적으로 하기 위해서는 여러 번의 실험을 통해서 최적의 매개변수의 값을 찾을 수밖에 없으며 여기에는 많은 시간과 노력이 필요하게 된다. 이러한 단점을 극복하기 위한 하나의 선택이 바로 병렬 인공지능망 시뮬레이터를 이용해서 다양한 매개변수 하에서 신경망을 동시에 학습하여 신경망 응용 시스템의 개발을 위한 시간과 노력을 최소화하는 것이다.

### 2. 이동 에이전트에 의한 학습세션 병렬 인공지능망 시뮬레이터

학습세션 병렬화는 상이한 학습계수나 신경망 구조를 갖는 다수의 신경망을 다수의 프로세서가 실행하도록 하는 것이다. 이 수준의 병렬화는 에이전트간 통신이 필요하지 않으므로 이동 에이전트를 기반으로 하는 경우가 가장 효과적으로 실행될 수 있는 경우에 해당한다. 구현을 위한 이동 에이전트 플랫폼은 aglets[4]을 사용하였으며 학습세션 병렬 인공지능망 시뮬레이터의 구성은 다음 그림 1과 같다. 즉, 표준 인터페이스를 제공하는 제어 에이전트가 있고, 사용자는 이를 이용하여 신경망의 구조 및 여러 학습 매개변수와 실행 호스트를 지정한다.

본 논문은 정보통신부의 정보통신기초기술연구지원사업(정보통신연구진흥원)으로 수행한 연구결과입니다.

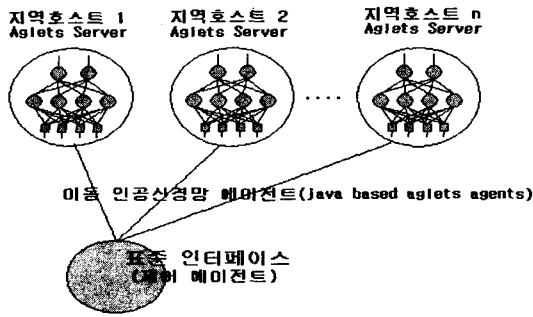


그림 1 신경망 시뮬레이터의 구조

제어 에이전트는 이를 기반으로 이동 에이전트를 생성하여 정해진 호스트로 전송하면 각 호스트는 자신의 신경망을 학습한다. 학습이 종료되면 결과는 제어 에이전트로 전송되고 개발자는 결과를 분석한다. 다음 그림 2는 개발한 병렬 인공신경망 시뮬레이터의 소프트웨어 구성도를 나타낸다.

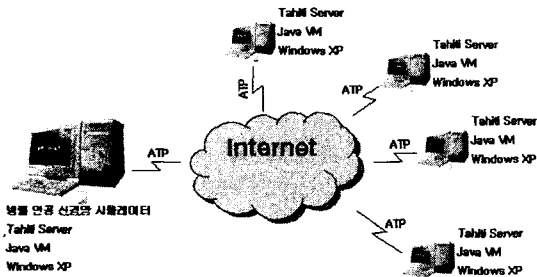


그림 2 병렬 인공신경망 시뮬레이터 소프트웨어 구성도

병렬 인공신경망 시뮬레이터는 15개의 java 클래스들로 구성되어 있으며, 이 클래스들의 상속관계를 그림 3에서 나타내고 있다.

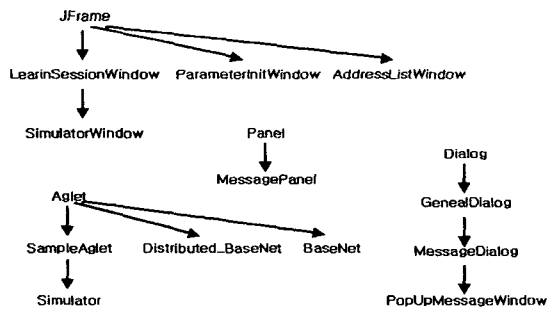


그림 3 병렬 인공신경망 시뮬레이터 클래스들의 상속관계

위 그림에서 Aglet을 상속받은 클래스들은 모두 에이전트들이고, 이들의 생성관계를 그림 4에서 나타내고 있

다. 다음의 그림 4에서처럼 Simulator클래스는 분산작업을 수행하는 Distributed\_BaseNet클래스를 생성하고 그리고 이 클래스는 BaseNet클래스를 생성한다. 이때 BaseNet클래스가 바로 이동에이전트이며, 이것이 인터넷에 연결되어 있는 지역호스트로 전송되어 신경망 학습을 하게 된다.

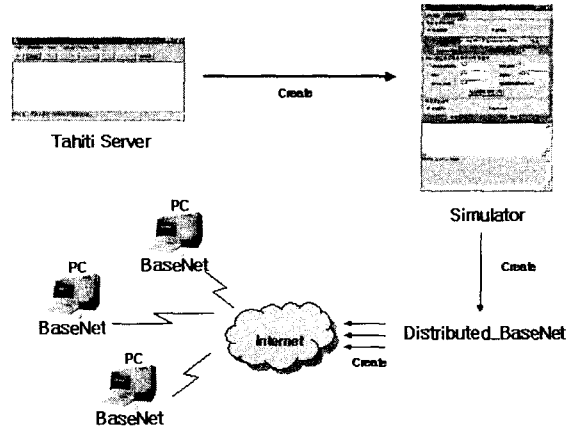


그림 4 병렬 인공신경망 시뮬레이터에 사용된 에이전트들의 생성관계

다음의 그림 5는 이 연구의 결과물인 시뮬레이터의 실행 모습을 나타내고 있다.

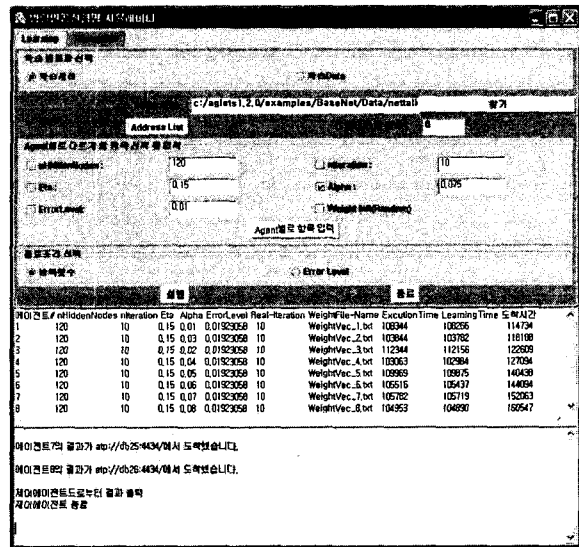


그림 5 병렬 인공신경망 시뮬레이터의 실행 모습

### 3. 실험에 의한 평가

평가를 위해서는 병렬 신경망 연구분야의 대표적 벤치마크인 NetTalk을 사용하였다. 평가에 사용한 신경망은

입력 뉴런 182(구두점 제외), 출력 뉴런 52(발음요소), 중간뉴런 10~120개를 사용하였다. NetTalk의 구조를 그림 6에서 나타내고 있다. 이 실험에서 평가를 위한 척도로는 CUPS(connection update per second)를 사용하였다. 즉 1초당 신경망이 학습을 위하여 가중치 수정량을 몇 개 계산하느냐를 기준으로 하였다. 결과는 다음 표 1과 같다(학습데이터 756개 기준, 네트워크 시간포함).

과거의 연구로 알려진 평가결과는 다음 표 2와 같다 [5]. 이들 연구와 비교함에 있어서 다수의 프로세서를 사용하는 병렬 컴퓨터와 직접적인 비교는 큰 의미는 없다고 판단된다. 그러한 컴퓨터들은 상대적으로 CPU 속도가 Pentium 보다 현저히 느린 대신 통신 오버헤드는 아주 작을 것이다. 따라서 수백 개의 프로세서를 사용하는 정도의 성능을 얻을 수 있다는 것은 개발한 시뮬레이터가 효과적임을 보인다. 또한 Sun SparcStation 10 (1.1MCUPS)에서의 결과와도 잘 들어맞는다. 실험에 사용한 P-IV는 Sun보다 약 9배 빠르며, 실험에서와 같이 3대에서 병렬로 실행한 경우 27.0MCUPS가 나온 것은 개발 시뮬레이터가 잘 수행되는 것을 보여준다.

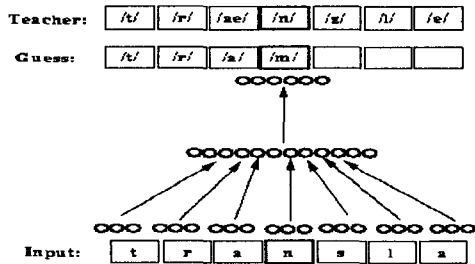


그림 6 NetTalk의 구조(그림에서 입력뉴런은 각 문자당 26개(총 182개)이며 출력뉴런은 52개)

표 2 평가결과

computer	호스트 수	신경망 크기	학습방법	MCUPS
P-IV 1.8GHz	4	NetTalk 182×120×52	lbe	5.9~6.5
"	8	NetTalk 182×120×52	lbe	7.9~11.1
"	3	NetTalk 182×10~120×52	lbp	9.8~27.0

lbe: learning by epoch  
lbp: learning by pattern  
MCUPS: Mega CUPS

다만, 표 1에서와 같이 개발한 시뮬레이터의 경우 어떤 경우에는 3대에서 병렬로 수행하여도 9.8MCUPS가 나오는 경우가 있는데, 그 경우는 실험에서 중간노드의 개수가 적고 반복 횟수가 적어서 상대적으로 네트워크 오버헤드가 크기 때문에 발생한 수치로 파악되었다. 실제로 신경망의 규모가 크고 학습횟수가 많을수록 성능이 안정적임을 확인할 수 있었다.

표 3 과거의 연구로 알려진 평가 결과

Computer	Number of PEs	Network Size	$\mu$	FP	MCUPS
Sun-3[19]	1	Nettalk	1	32	0.034
NCube/4 [5]	16	Optimal network	1	32	0.19
Sun SparcStation 10[84]	1		1	32	1.1
Alpha Station[84]	1		1	32	3.2
CM-2[85]	16K	Nettalk(60)	1	32	2.8
Cray2[22]	4	257×256×131,072	1	32	10 <sup>1</sup>
iPSC/860[19]	32	Nettalk(80)	2,000	32	11
MP-1[14]	4,096	128×64×16	1,536	32	12
IMB RISC/6000 550 [86]	1	1000×1000×1	1,000	32	17.6
Cray X-MP [22]	4	257×256×131,072	1	32	18 <sup>1</sup>
CM-5[23]	32	Nettalk(80)	lbb	32	18.33
CM-2[21]	65,536	Nettalk(80)	4,096	32	40
Cray Y-MP[22]	2	257×256×131,072	1	32	40 <sup>1</sup>
MP-1216 [15]	16,384	Nettalk(120)	lbb	32	41
Q-x system [28]	64	257×128×256	8,196	32	44 <sup>1</sup>
CM-5 [11]	256	1024×256×64	lbe	32	50
Fujitsu VP-2400/10 [25]	1	Nettalk(60)	1	32	60
CM-5[22]	512	257×256×131,072	1	32	76
AP1000 [19(in Chapter 7)]	512	Nettalk(120)	906	32	86
CM-2 [87]	64K	128×128×128	65,536	32	350
TLA[10]	16	256×64×256	1	32	0.6
Transputers[54]	53	Nettalk(60)	1,500	32	1.54
MEIKO [51]	120	512×256×64	30	32	8
Sandy [62]	32	Nettalk(60)	1	32	42
Cellular arch. [88]	4,175	Nettalk(60)	1	16	51.5 <sup>1</sup>
SNAP [79]	64	64×64×64	1	32	76.6
RAP [59]	40	640×640×640	1	32	102
MUSIC [6]	45		1	32	203
SNAP[79]	64	5124×512×512	1	32	302
Sandy [62]	256	Optimal network	1	32	567
MY-NEUPOWER [74]	512	Optimal network	1	16	1,260
CNAPS [72,73]	512	1,900×500×12	1	16	2,379 <sup>1</sup>

<sup>1</sup>The performance has been estimated or simulated (not measured).

#### 4. 결론

이 연구는 인터넷 환경에서 이동 에이전트에 의하여 규모가 큰 신경망을 병렬로 학습할 수 있는 학습 세션 병렬 인공신경망 시뮬레이터에 관한 것이다. 대표적인 벤치마크 문제인 NetTalk을 대상으로 실험을 통하여 개발한 시뮬레이터가 상당히 효과적임을 알 수 있다. 하지만 보다 정교한 실험과 부동 소수점 연산의 개수를 이용한 이론적 분석과 실험 결과와의 비교와, 이질적인 호스트로 구성된 환경에서의 평가가 요구된다.

#### 5. 참고문헌

[1] Nikola B. Serbedzija, "Simulating Artificial Neural Networks on Parallel Architectures", Computer, Vol. 29, No. 3, 56-63, 1996.  
 [2] Manavendra Misra, "Parallel Environments for Implementing Neural Networks", Neural Computing Surveys Vol. 1, 48-60, 1997. <http://www.icsi.berkeley.edu/~jagota/NCS>  
 [3] Gerd Kock and Nikola B. Serbedzija, "Simulation of Artificial Neural Networks", SAMS, Vol. 27, 15-59, 1996.  
 [4] D. B.Lange, M. Oshima, "Programming and Deploying java Mobile Agents with Aglets". Addison Wesley, 1998.  
 [5] N. Sundararajan, "Parallel Architectures for Artificial Neural Networks", pp. 49.