

논문 2010-5-31

홉필드 네트워크에서 에너지 함수를 이용한 최적 경로 탐색에 관한 연구

Study on the Shortest Path by the energy function in Hopfield networks

고영훈*, 김윤상**

Ko Young Hoon, Kim Yoon Sang

요약 홉필드 네트워크는 패턴 매칭과 더불어 최적화 문제를 푸는 도구로 사용될 수 있다. 특히 Zhang과 Ali는 홉필드 네트워크의 노드를 2차원으로 확장하여 최적화 문제를 해결하였다. 잠재적 브랜치의 총합인 노드의 제곱만큼 뉴런이 필요한 Ali 알고리즘은 탐색 네트워크가 커지면 많은 시간이 소요되는 단점이 있다. 본 논문에서는 Ali의 방식을 개선하여 계산량을 대폭 줄이고 효과적으로 최적 경로를 탐색할 수 있는 방식을 제안한다. 효과적인 최적 경로 탐색을 위하여 2단계로 구분하여 진행된다. 1단계에는 홉필드 네트워크를 2단계에는 eSPN 알고리즘을 사용하여 최적 경로를 탐색할 수 있다. 제안된 방식은 샘플 네트워크를 통하여 최적 경로 탐색이 확인되었으며, Ali 알고리즘보다 빠르고 간단하여 실제 최적화에 적용하기가 용이하다. 특히, 네트워크의 브랜치 비용이 변화할 경우에도 홉필드 네트워크의 연결 시냅스가 아닌 입력 바이어스를 조정하므로 동적으로 변화하는 네트워크의 최적 경로 탐색에도 유용하다.

Abstract Hopfield networks have been proposed as a new computational tool for finding the shortest path of networks. Zhang and Ali studied the method of finding shortest path by expended neurons of Hopfield networks. Ali Algorithm is well known as the tool with the neurons of branch numbers. Where a network grows bigger, it needs much more time to solve the problem by Ali algorithm. This paper modifies the method to find the synapse matrix and the input bias vector. And it includes the eSPN algorithm after proper iterations of the Hopfield network. The proposed method is a tow-stage method and it is more efficient to find the shortest path. The proposed method is verified by three sample networks. And it could be more applicable than Ali algorithm because it's fast and easy. When the cost of brach is changed, the proposed method works properly. Therefore dynamic cost-varying networks could be used by the proposed method.

Key Words : Hopfield Network, Shortest path

1. 서 론

1982년에 John Hopfield가 발표한 논문은 뉴럴 네트워크에 새로운 분야를 추가하는 계기가 되었다^[1]. 홉필드 네트워크(Hopfield network)은 모든 뉴런의 출력이 입력으로

채워지는 구조이며, 초기에는 저항과 커패시터를 이용한 전자회로로 소개되었다. 홉필드 네트워크에서는 뉴런간의 연결강도를 나타내는 연결 시냅스 매트릭스가 대칭적이다. 홉필드 네트워크는 뉴런간의 연결 정보를 통하여 패턴들을 기억하고, 입력값에 가장 가까운 패턴으로 수렴하는 특징을 가지고 있어, 손상된 패턴의 복원에 주로 사용된다.

*정희원, 협성대학교 컴퓨터공학과

**정희원, 한국기술교육대학교 인터넷미디어공학부

접수일자 2010.6.11 수정일자 2010.9.13

게재확정일자 2010.10.15

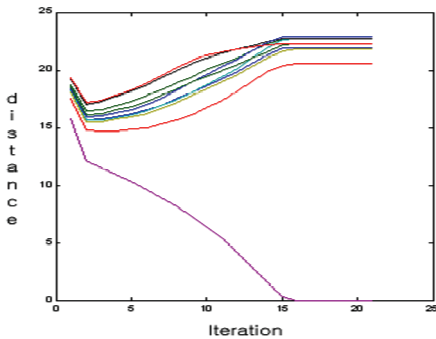


그림 1. 홉필드 네트워크의 패턴간 거리 수렴 곡선
Fig 1. Convergence curve of distance of patterns on Hopfield Network

그림 1.은 홉필드 네트워크에서 기억된 패턴과 입력 지점간의 거리를 나타낸 것이다. 반복연산이 진행됨에 따라 하나의 패턴과의 거리가 줄어들고, 다른 패턴들과의 거리는 늘어난다. 어느 정도 반복연산이 진행되면 선택된 패턴으로 수렴하게 된다.

N개의 뉴런은 N차원의 공간에서 M개의 패턴을 기억하고 있으며, 입력지점에 가까운 패턴 지점으로 수렴되며, 이 과정에서 에너지 함수는 감소하는 특성을 나타낸다. 에너지 함수는 다차원 선형 시스템의 풀이 과정에서 나오는 도함수에서 기인한다. 2차원 함수에서 기울기가 0이 되는 지점이 최소값을 나타내듯이, 다차원 선형 시스템에서 에너지 함수를 통하여 최소값을 구할 수 있다.

Hopfield는 홉필드 네트워크이 에너지 함수가 다차원 평면에서 경사로 방향으로 수렴하는 것을 증명하였다^[2]. 단 고이득 회로와 연결 시냅스 매트릭의 대칭이 경사로 방향으로 수렴하기 위한 조건이다.

Hopfield와 Tank는 홉필드 네트워크를 패턴 복원이 아닌 최적화 문제에 적용하였다^[3]. 특히 TSP(traveling sales man problem)를 적용할 수 있는 모델을 제시하였다. Rauch와 Winarske는 네트워크에서 두 개의 노드 사이의 최적 경로를 찾는데 Hopfield Network을 사용하였다^[4]. Zhang과 Thomopoulos는 n by m 매트릭스의 뉴런을 구성한 후 m을 확장하여 원치 않는 경로의 탐색을 제거하였다^[5].

특히 Ali는 그의 논문에서 다섯 개의 항으로 에너지 함수를 정의하고, 이를 연결 시냅스 매트릭스에 적용하여 최적 경로를 찾아내었다^[6]. 본 논문에서는 Ali의 알고리즘을 단순화하고 홉필드 네트워크의 선처리 과정과 제안된 eSPN(Extended Shortest Path searching by Neuron)

알고리즘을 통하여 최적 경로를 찾는다.

II. 제안된 알고리즘

홉필드 네트워크에서 리야프노프(Lyapunov) 함수로 알려진 에너지 함수는 다음과 같이 정의된다. 선형 시스템에서와 달리 홉필드 네트워크는 재귀연결을 사용하므로 음의 부호로 나타낸다.

$$E = -\frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N C_{ij} X_i X_j - \sum_{i=1}^N I_i X_i \quad (1)$$

에너지함수가 뉴런의 출력함수와 관계없이 구성되려면 고이득 회로로 구성되어야 한다. 또한 연결 시냅스 매트릭스인 C가 대칭일 경우 에너지 함수는 경사면으로 감소하는 것이 증명되었다. 경사면을 따라 감소하는 방식은 여러 가지가 있는데, 홉필드 네트워크에서는 현재 지점에서 가장 경사면이 급한 방향으로 감소한다.

Ali는 에너지 함수를 변형하여 다섯 개의 항을 제시하고 홉필드 네트워크의 연결 시냅스 매트릭스와 입력 바이어스 벡터를 구하였다. 그리고 홉필드 네트워크의 특성을 이용하여 에너지함수가 감소하면 특정 뉴런들이 1에 수렴하게 된다. 이는 곧 전체 경로의 비용을 최소화시킬 뿐만 아니라, 뉴런간의 연결정보를 사용하여 최적 경로상의 뉴런만을 발화시키게 된다.

$$E = \frac{\mu_1}{2} \sum_{i=1}^N \nu_i X_i + \frac{\mu_2}{2} \sum_{i=1}^N \rho_i X_i + \frac{\mu_3}{2} \sum_{p=1}^m \left(\sum_{q=1}^l X_{pq} - \sum_{q=1}^l X_{qp} \right)^2 + \frac{\mu_4}{2} \sum_{i=1}^N X_i (1 - X_i) + \frac{\mu_5}{2} (1 - X_{ds}) \quad (2)$$

위의 수식은 Ali의 에너지 함수이다. Ali의 방식은 뉴런이 브랜치를 의미한다. μ_1 항은 브랜치의 비용을 담고 있다. μ_2 항은 해당 브랜치가 연결되지 않는 경우에 μ_1 항의 반대되는 역할을 한다. μ_3 항은 브랜치의 연결방식에 대한 정보를 담고 있다. 브랜치는 방향성을 가지고 있어 뉴런 N_i 에서 N_j 로 연결되어 있을 때 B_{ij} 로 표기한다. B_{ij} 와 B_{jk} 는 순방향 브랜치로 하나의 경로를 형성

할 수 있다. 하지만 B_{ij} 와 B_{kj} 는 충돌하는 브랜치로 경로를 형성할 수 없는 경우이다. 이러한 연결정보를 사용하여 순방향일 때와 역방향일 때의 에너지 함수에 가중치를 반대로 한다. μ_4 항은 0.5를 기준으로 1 또는 0으로 수렴시키는 가속화기능을 한다. 이 항이 없으면 경로에 해당되는 한 개의 뉴런만 발화할 수 없다. μ_5 항은 시작점과 끝점을 연결하는 브랜치의 정보를 사용하여 최적 경로의 시작과 끝점을 나타내게 한다.

본 논문에서는 Ali의 에너지 함수를 변형하여 3개의 항으로 에너지 함수를 구성한다. μ_1 항과 μ_2 항의 값을 α_1 으로 통합하였다. 두 노드 사이에 브랜치가 존재하지 않는 경우에는 해당 브랜치의 입력 바이어스를 -1로 초기화하고 연결 매트릭스의 값을 0으로 세팅하여 변동이 없도록 처리한다. 즉, 연결되지 않은 브랜치는 최적 경로의 브랜치가 될 확률이 0이므로 반복 연산에서 제외시킨다. μ_5 항도 시작점과 끝점을 연결하는 브랜치의 비용 값을 조정하여 처리한다. 결국 3개의 항으로 정리되며 에너지 함수는 다음과 같다.

$$E = \alpha_1 \sum_{i=1}^N \gamma_i X_i + \alpha_2 \sum_{i=1}^N X_i (1 - X_i) + \alpha_3 \sum_{p=1}^m \left(\sum_{q=1}^l X_{pq} - \sum_{q=1}^l X_{qp} \right)^2 \quad (3)$$

에너지 함수는 결과적으로 홉필드 네트워크의 연결 시냅스 매트릭스와 입력 바이어스 벡터로 변환되어야 한다. 에너지 함수를 X_i 에 대하여 편미분하면 연결 시냅스 매트릭스와 입력 바이어스 벡터를 얻을 수 있다.

$$C_{ij} = \alpha_2 \delta_{ij} + \alpha_3 (\delta_{py} + \delta_{ax} - \delta_{px} - \delta_{ay}) \quad (4)$$

연결 시냅스 매트릭스의 값은 α_2 와 α_3 의 항으로 표현된다. 여기서 δ_{ij} 는 Kronecker delta로 $i=j$ 일때 1의 값을 갖고 그 외에는 0의 값을 가진다.

$$I_j = -\alpha_1 \gamma_j - \frac{\alpha_2}{2} \quad (5)$$

입력 바이어스 벡터의 값은 α_1 와 α_2 의 항으로 표현된다.

네트워크의 연결 정보는 연결 시냅스 매트릭스에 저장된다. 그리고 브랜치의 비용 정보는 입력 바이어스 매트릭스에 저장된다. 그리고 하나의 노드에 연결된 여러개의 노드 중에서 최적 경로는 하나만 선택되어야 하므로 α_2 에 의한 수렴 보조 계수와 시작점과 끝점을 잇는 브랜치의 비용처리를 통해 홉필드 네트워크이 구성된다.

특히, 브랜치의 비용이 시간에 따라서 변화하는 경우에 연결 시냅스 매트릭스의 내용과 관계없이 입력 바이어스 매트릭스만을 통하여 가능하다.

제안된 방식은 2단계 알고리즘을 사용한다. 우선 홉필드 네트워크를 통하여 뉴런에 최적 경로의 정보를 담게하고, 다음 단계로 eSPN 알고리즘을 통하여 최적 경로를 계산해낸다. 즉, 뉴런이 최종적으로 수렴하기 전에 뉴런의 데이터를 분석하여 최적 경로를 찾아낸다.

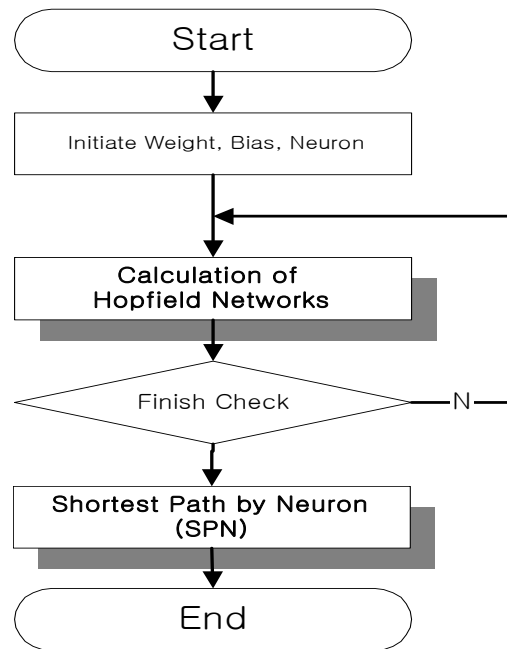


그림 2. 최적 경로 탐색을 위한 2단계 흐름도
Fig 2. Two stage flowchart for finding the shortest path

그림 2는 홉필드 네트워크의 연산과 제안된 eSPN 알고리즘의 전체적인 흐름을 나타내고 있다.

먼저 최적 경로를 찾고자 하는 네트워크의 노드와 브랜치를 분석하여 연결 시냅스와 입력 바이어스 값을 계산한다. 홉필드 네트워크의 계산을 반복하면서 다음 단계로

넘어가도 되는지 뉴런의 값을 체크한다. 다음 단계로 넘어가도 될 정도로 뉴런의 값이 수렴했으면 eSPN 알고리즘을 통하여 최적 경로를 탐색한다.

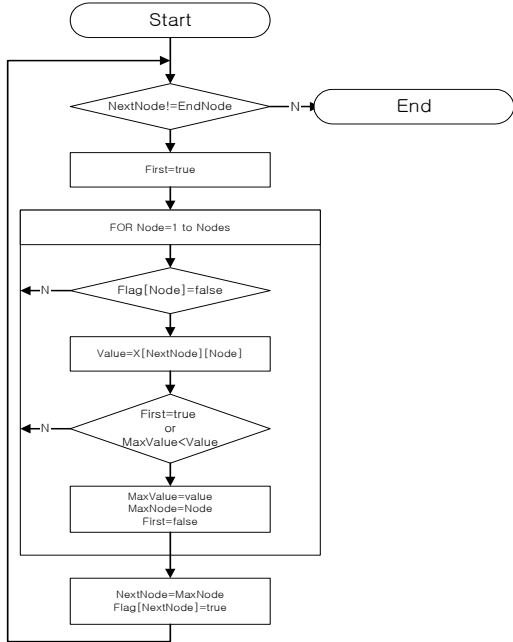


그림 3. eSPN 알고리즘의 흐름도
Fig 3. eSPN algorithm flowchart

그림 3.은 eSPN 알고리즘을 나타내고 있다. eSPN 알고리즘은 뉴런을 순회하면서 최대값과 최소값을 비교하여 각 행의 최적 뉴런을 찾아내는 알고리즘이다. 어느 정도 뉴런이 수렴하면 효과적으로 최적값의 뉴런을 찾아낼 수 있다.

III. 실험 및 결과

실험을 위해 C++를 사용하여 두개의 Class를 작성하였다. HopFieldNet 클래스는 N개의 뉴런으로 구성된 홉필드 네트워크의 기본 동작을 위한 클래스이다. 이를 계승한 HFRoute 클래스는 네트워크 정보를 분석하여 연결 시냅스 매트릭스와 입력 바이어스 벡터의 값을 계산하고 HopFieldNet 클래스를 호출하여 뉴런을 일정강도로 수렴시킨 후 eSPN 알고리즘을 통하여 최적경로를 찾아내는 클래스이다. 사용된 계수의 값은 α_1 과 α_2 가 1이며 α_3

는 0.5의 값을 사용한다.

Ali의 알고리즘이 안정적으로 수렴하는데 수천 번의 반복연산이 필요한 반면 eSPN 알고리즘을 사용할 경우 100번 정도의 반복연산으로 뉴런에 정보를 담을 수 있다. 세 개의 예제 네트워크를 사용하여 최적경로를 탐색하는지 실험한다.

그림 4.는 첫 번째 샘플 네트워크로 5개의 노드의 작은 네트워크이다. 브랜치는 비용을 담고 있으며, 브랜치는 양방향의 비용이 같다고 가정한다. 시작 노드는 0이고 끝 노드는 가장 큰 수인 4이다.

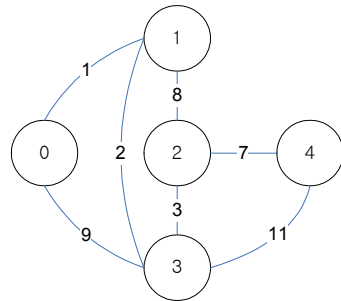


그림 4. 5-노드 샘플네트워크
Fig 4. 5-node Sample network

표 1.은 샘플 네트워크의 d행렬 값을 나타낸다. d행렬은 네트워크의 연결브랜치 정보를 나타낸 것으로, 브랜치의 양방향성이 같은 값이므로, 일단 행렬의 위쪽 삼각부분에만 브랜치값을 기입한다. 실제 프로그램이 수행되면 표 1의 d 행렬값은 표 2의 c행렬값으로 변환된다. i번 노드에서 j번 노드로의 브랜치는 j번 노드에서 i번 노드로의 브랜치와 동일하므로 아래쪽 삼각부분을 채워넣는다. 또한 i번 노드에서 I번 노드로의 브랜치인 셀프 브랜치는 -1로 지정한다. 그리고 시작노드와 끝노드의 브랜치는 +1, 끝노드와 시작노드의 브랜치는 +1의 값으로 지정된다. 또한 브랜치의 값은 노드별로 정규화되어 조정된다.

표 1. 5-노드 샘플네트워크의 d 행렬
Table 1. d matrix of 5-node sample network

	0	1	2	3	4
0	0	1	-1	9	-1
1	0	0	8	2	-1
2	0	0	0	3	7
3	0	0	0	0	11
4	0	0	0	0	0

c행렬 값을 사용하여 홉필드 네트워크를 적용하면 뉴런의 값에 최적 경로의 정보를 담을 수 있으며, 이를 eSPN 알고리즘으로 추출하면 0->1->3->2->4의 경로를 얻을 수 있다.

표 2. 5-노드 샘플 네트워크의 c 행렬값
Table 2. c matrix of 5-node sample network

	0	1	2	3	4
0	-1	0.9	-1.0	0.18	-1
1	0.9	-1	0.27	0.8	-1
2	-1	0.27	-1	0.72	0.36
3	0.18	0.81	0.72	-1	0
4	1	-1	0.36	0	-1

그림 5.는 8개의 노드를 가진 샘플 네트워크이다. 시작 노드는 0이고 끝 노드는 7이다. 8개의 노드이므로 가능한 브랜치의 개수는 8의 제곱인 64이다. 즉 64개의 뉴런이 필요하다. 뉴런간의 연결 시냅스 매트릭스의 정보에는 네트워크의 연결정보가 담겨 있고, 한 노드에 연결된 여러 브랜치 중에서 하나의 최적경로를 선택할 수 있도록 하는 수렴 가속화 계수를 포함하고 있다.

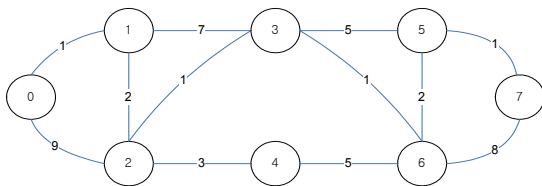


그림 5. 8-노드 샘플네트워크
Fig 5. 8-node sample network

표 3.은 8-노드 샘플 네트워크의 d행렬값을 나타낸다. d행렬값은 c행렬값으로 변환되어 홉필드 네트워크의 반복 연산 후의 뉴런의 값으로 eSPN 알고리즘을 통하여 최적 경로가 0->1->2->3->6->5->7 임을 찾을 수 있다. eSPN 알고리즘에는 빠르게 경로를 찾기 위해서 가능성이 있는 노드만을 선별하여 비교한다. 즉, 자기 자신에게로 연결된 브랜치나 브랜치 루프는 선택되지 않도록 처리한다.

표 3. 8-노드 샘플 네트워크의 d 행렬값
Table 3. d matrix of 8-node sample network

	0	1	2	3	4	5	6	7
0	0	1	5	-1	-1	-1	-1	-1
1	0	0	2	7	-1	-1	-1	-1
2	0	0	0	1	3	-1	-1	-1
3	0	0	0	0	-1	5	1	-1
4	0	0	0	0	0	-1	5	-1
5	0	0	0	0	0	0	0	2
6	0	0	0	0	0	0	0	8
7	0	0	0	0	0	0	0	0

그림 6.은 11개의 노드를 가진 샘플 네트워크이다. 네트워크의 노드 수가 증가함에 따라 브랜치의 개수는 급격하게 증가하여 홉필드 네트워크의 연산 속도를 저하시킨다. 매트랩의 nnet 툴박스에 구현된 newhop 함수를 사용하면 약 64*64=4096개의 노드까지 계산이 가능하다. 그 이상의 경우에는 파일 처리를 포함한 랭귀지로 구현하여야 하며, 클러스터링 기법을 사용하여 분산 처리하는 방법을 생각해 볼 수 있다.

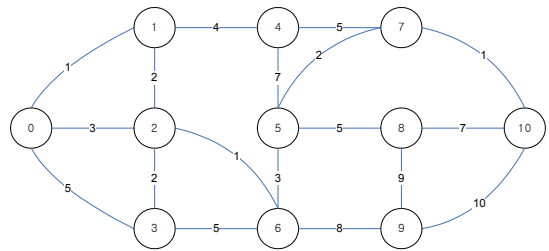


그림 6. 11-노드 샘플네트워크
Fig 6. 11-node sample network

표 4.는 11-노드 샘플 네트워크의 d 행렬값을 나타낸다. 같은 방법으로 eSPN 알고리즘을 통하여 최적 경로가 0->1->2->6->5->7->10 임을 찾을 수 있다. 뉴런이 많아지면서 하나의 최적 경로를 찾기 위한 계수의 선택이 더 민감해진다. 본 실험에서는 α_3 의 값을 감소시켜 비용의 합계가 증가하는 문제를 해소하였다.

표 4. 11-노드 샘플 네트워크의 d 행렬값
Table 4. d matrix of 11-node sample network

	0	1	2	3	4	5	6	7	8	9	10
0	0	1	3	5	-1	-1	-1	-1	-1	-1	-1
1	0	0	2	-1	4	-1	-1	-1	-1	-1	-1
2	0	0	0	2	-1	-1	-1	-1	-1	-1	-1
3	0	0	0	0	-1	-1	5	-1	-1	-1	-1
4	0	0	0	0	0	7	-1	5	-1	-1	-1
5	0	0	0	0	0	0	3	2	5	-1	-1
6	0	0	0	0	0	0	0	-1	-1	8	-1
7	0	0	0	0	0	0	0	0	-1	-1	1
8	0	0	0	0	0	0	0	0	0	9	7
9	0	0	0	0	0	0	0	0	0	0	10
10	0	0	0	0	0	0	0	0	0	0	0

IV. 결론

홉필드 네트워크를 이용한 최적 경로 탐색은 Hopfield와 Tank가 처음 시도한 이후 다양한 발전을 거듭해왔다. 특히, Ali의 알고리즘은 정확한 최적 경로를 탐색하는 뛰어난 성능을 발휘한다.

Ali 알고리즘은 다차원 시스템의 에너지 함수를 정의하고, 홉필드 네트워크의 반복연산에 따라 에너지 함수가 감소하도록 한다. 에너지 함수가 감소하면 다차원 최적 값에 도달하는 데, 이 때 1에 수렴된 뉴런이 최적 경로의 브랜치를 나타낸다.

그러나 노드수의 자승에 해당하는 브랜치 수의 뉴런이 필요하여 많은 양의 계산이 불가피하다. 그리고 에너지 함수를 5개의 항으로 정의하였는데, 정확한 동작을 위한 5개의 계수에 산출이 어렵다.

본 논문에서는 보다 효과적이고 빠른 계산을 위하여 Ali의 에너지 함수를 변형시켰다. 그리고 2단계 기법을 통하여 홉필드 네트워크의 반복 연산의 횟수를 10분의 1 이상 줄이고, 제안된 eSPN 알고리즘을 통하여 빠르게 최적 경로를 계산할 수 있다. 세 개의 예제 네트워크를 통하여 제안된 알고리즘이 정확한 최적 경로를 탐색하는지 확인하였다.

에너지 함수를 이용하여 다차원 선형 시스템의 해를 구하는 방법은 매우 다양한 응용이 가능하다. 이를 현실적으로 활용하기 위한 eSPN 알고리즘은 실시간으로 처

리되는 시스템 개발에 활용될 수 있으며, 브랜치의 비용이 시간에 따라 변화하는 동적 네트워크의 최적 경로 탐색에 활용될 수 있다.

참고 문헌

- [1] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational properties," Proceedings of the National Academy of Sciences, vol. 79, pp. 2554-2558, 1982.
- [2] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," Proc. Nat. Acad. Sci., Vol. 81, pp.3088-3092, 1984.
- [3] J. J. Hopfield and D. W. Tank, "Neural computations of decisions in optimization problems," Bol. Cybern, Vol. 52, pp. 141-152, 1986.
- [4] H. E. Rauch and T. Winarske, "Neural networks for routing communication traffic," IEEE Cont. Syst. Mag., pp. 26-30, Apr 1988.
- [5] L. Zhang and S. C. A. Thomopoulos, "Neural networks implementation of the shortest path algorithm for traffic routing in communication networks," Proc. Int. Joint Conf. Neural Networks, Jun 1989.
- [6] Mustafa K. Mehmet Ali and Faouzi Kamoun, "Neural Networks for Shortest Path Computation and Routing in Computer Networks," IEEE Trans. on Neural Networks, Vol. 4, no. 6, Nov 1993.

저자 소개

고 영 훈(정회원)



- 1999년~현재
협성대학교 컴퓨터공학과 교수
- 1997년 연세대학교 전자공학과 졸업
(공학박사)
- 1993년 연세대학교 전자공학과 졸업
(공학석사)

<주관심분야 : 데이터통신, 데이터베이스>

김 윤 상(정회원)



- 2005년 3월~현재
한국기술교육대학교 컴퓨터공학부 부
교수.
- 1999년 성균관대학교 전기공학과 졸
업 (공학박사).
- 1995년 성균관대학교 전기공학과 졸
업 (공학석사)

1993년 성균관대학교 전기공학과 졸업 (공학사)
<주관심분야 : 로봇 제어, 가상 현실>