

Spiking Neural Networks 에서의 단계적 구간연산 학습

이경희*

*평택대학교 정보통신학과

khlee@ptu.ac.kr

A stepwise interval arithmetic learning for spiking neural networks

Kyunghee Lee*

*Dept. of Info. & Comm., PyeongTaek University

요 약

본 논문은 스파이킹 신경회로망(Spiking Neural Networks)에서 BP(Back Propagation) 학습알고리즘에 대한 일반화 형태인 기존의 구간연산 학습알고리즘을 개선하여 스파이킹 뉴런(Spiking Neuron)들의 반응영역 범위를 학습 진행에 따라 조절이 가능한 단계적 구간학습 방안을 제안한다. 또한, 간단한 컴퓨터 시뮬레이션을 통하여 쿨롱에너지 포텐셜(Coulomb Energy Potential)을 가지는 스파이킹 신경회로망에서 범위(Lower bound & Upper bound)의 값을 가지는 구간데이터(Interval data)와 하나의 값을 가지는 포인트데이터(Point data)가 혼재된 학습데이터에서의 제안한 학습방안의 타당성과 전문가시스템(expert system)에서의 “don't care attributes” 적용가능성을 보인다.

1. 서론

생물학적인 뇌의 동작을 기반으로 하는 스파이킹 신경회로망(Spiking Neural Networks) 모델은 3세대 모델로서 뉴런(Neuron)에서 발생하는 스파이크 (Spike)의 정확한 발생시각을 중심으로 반응하는 모델로서 시간적 정보에 대한 실시간 처리를 필요로 하는 분야에서 좋은 결과를 내고 있다[1-2]. 스파이킹 신경회로망을 학습시키기 위한 알고리즘은 BP(Back Propagation) 기반의 SpikeProp 알고리즘이 일반적이며 이를 일반화한 구간연산 학습알고리즘이 제안되었다[3-4] 본 논문에서 스파이킹 신경회로망에서 BP 알고리즘에 대한 일반화 형태인 기존의 구간연산 학습알고리즘을 개선하여 스파이킹 뉴런들의 반응영역 범위를 학습 진행에 따라 조절이 가능한 단계적 구간학습 방안을 제안하고, 컴퓨터 시뮬레이션을 통하여 쿨롱에너지 포텐셜을 가지는 스파이킹 신경회로망에서 구간데이터와 포인트데이터가 효과적으로 학습될 수 있으며 전문가시스템에서의 “don't care attributes”에 대한 적용 가능성도 보인다.

2. 스파이킹 신경회로망과 구간연산 학습

2.1 구간연산

신경회로망에서의 구간연산은 다음과 같다[5].

구간들에 대한 합:

$$A + B = [a^l, a^u] + [b^l, b^u] = [a^l + b^l, a^u + b^u] \quad (2-1)$$

구간에 대한 실수의 곱:

$$m \cdot A = m \cdot [a^l, a^u] = \begin{cases} [m \cdot a^l, m \cdot a^u] & \text{if } m \geq 0 \\ [m \cdot a^u, m \cdot a^l] & \text{if } m < 0 \end{cases} \quad (2-2)$$

여기서 l 과 u 는 최하치와 최상치를 각각 나타낸다.

2.2 스파이킹 신경회로망과 구간연산

시간 t 에서 스파이킹 뉴런 j 의 post-synaptic, $x_j(t)$, 과 활성화함수는 다음과 같다[3-4].

$$x_j(t) = \sum_{i \in \Gamma_j} \sum_{k=1}^m w_{ij}^k y_i^k(t) \quad (2-3)$$

$$y_j^k(t) = \varepsilon(t - t_j - d^k) \quad (2-4)$$

$$\varepsilon(t) = 1/\tau \exp(1 - 1/\tau) \quad (2-5)$$

M 개의 메모리를 갖는 스파이킹 신경회로망의 경우 쿨롱포텐셜에 의한 에너지와 학습을 위한 가중치의 조정은 다음과 같다[3].

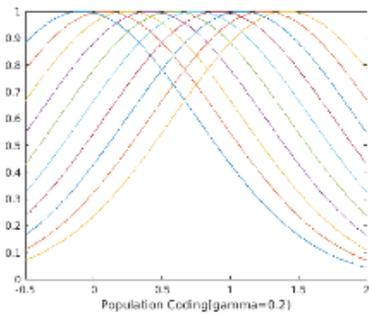
$$\Psi = \frac{1}{(2L)} \sum_{a,b} \sum_{j \in J} \text{sign}(a,b) \|t_j^a - t_j^b\|^{-L} \quad (2-6)$$

$$\Delta w_{ij}^k = \frac{1}{2} \eta \cdot \{ y_i^k(t_j^{a'}) \delta_j^{a'} - y_i^k(t_j^{b'}) \delta_j^{b'} + y_i^k(t_j^{a''}) \delta_j^{a''} - y_i^k(t_j^{b''}) \delta_j^{b''} \}$$

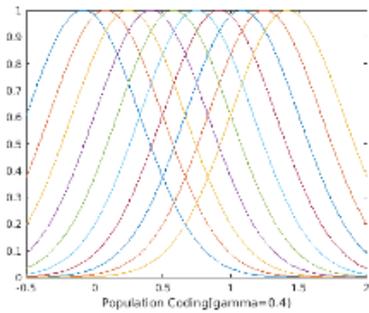
where $w_{ij} \geq 0$ (2-7)

3. 컴퓨터 시뮬레이션

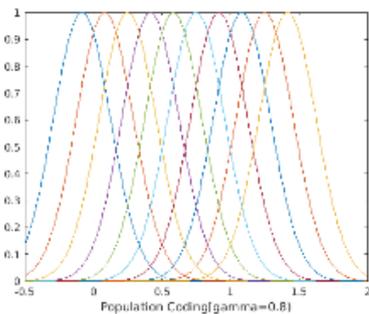
제안한 단계적 구간연산 학습의 성능을 확인하기 위한 간단한 컴퓨터 시뮬레이션 결과를 보인다. 실험에서는 뉴런에서의 정보부호화로서 주변의 관련 뉴런들의 영향정도를 포함하여 부호화하는 주변적 부호화 (Population Coding) 등을 적용한다[6]. 각 뉴런의 주변 6 개 뉴런 들에 적용한 가우시안 활성화 함수의 예를 그림 1 에 표현한다. Gamma(γ)의 값이 증가될수록 뉴런 반응영역의 범위가 감소하여 보다 정확한 스파이크에만 반응하게 됨을 알 수 있다.



(a)



(b)



(c)

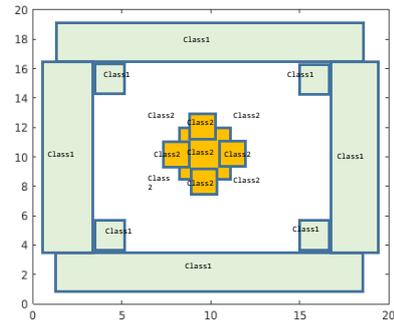
(그림 1) 가우시안 활성화함수의 예 ((a) $\gamma=0.2$, (b) $\gamma=0.4$, (c) $\gamma=0.8$)

실험에 사용한 주요 파라미터로는 학습율=0.0000002, 가중치의 초기값=0.0~1.0, 학습횟수(epoch)=20 이며 학습횟수당 300 번의 학습데이터 선택하여 진행하였다.

표 1 과 그림 2 에 학습에 사용한 13 개(Class1 8 개, Class2 5 개)의 구간데이터와 4 개의 포인트 데이터를 좌표와 모양으로 각각 나타낸다. 시험용 데이터는 임의의 233 개 포인트 데이터를 선택하였다

(표 1) 학습을 위한 구간 및 포인트 데이터(좌표)

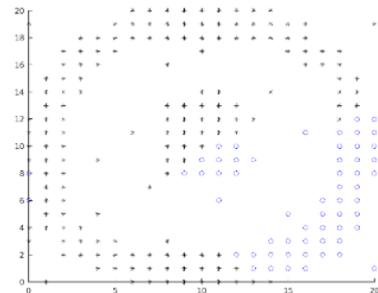
	학습데이터 (X_min, X_max, Y_min, Y_max)
Class 1	[1.0, 19.0, 17.0, 20.0] [0.0, 3.0, 3.0, 17.0]
	[1.0, 19.0, 0.0, 3.0] [17.0, 20.0, 3.0, 17.0]
	[3.0, 5.0, 3.0, 5.0] [15.0, 17.0, 3.0, 5.0]
	[15.0, 17.0, 15.0, 17.0] [3.0, 5.0, 15.0, 17.0]
Class 2	[8.0, 12.0, 8.0, 12.0] [7.0, 9.0, 9.0, 11.0]
	[11.0, 13.0, 9.0, 11.0] [9.0, 11.0, 11.0, 13.0]
	[9.0, 11.0, 7.0, 9.0]
	[12.0, 12.0, 12.0, 12.0] [8.0, 8.0, 8.0, 8.0]
	[8.0, 8.0, 12.0, 12.0] [12.0, 12.0, 8.0, 8.0]



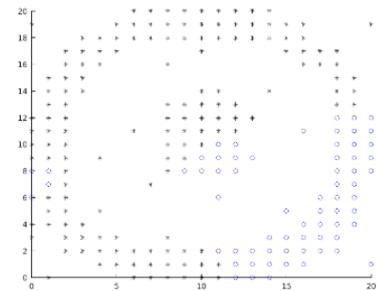
(그림 2) 학습을 위한 구간 및 포인트 데이터 (2 차원)

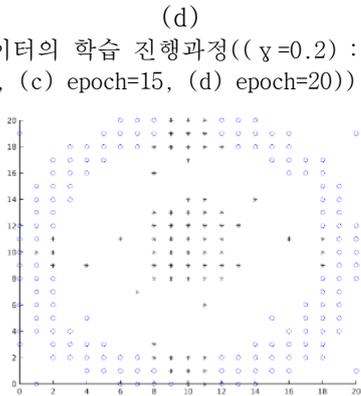
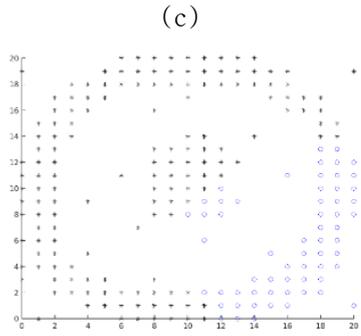
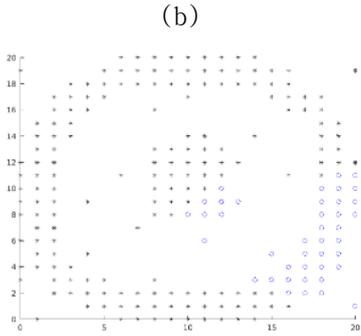
3.1 반응영역의 고정적 학습실험

그림 3 과 4 그리고 5 에 $\gamma=0.2$, $\gamma=0.4$, $\gamma=0.8$ 로 고정된 반응영역을 설정하고 학습을 진행시킨 과정을 각각 보인다. 학습실험의 결과 class 2 데이터와 class 1 데이터에 대한 구간학습이 일부 제대로 이루어지지 않은 것을 알 수 있다(그림 3(d), 4(d), 5(d) 참조).

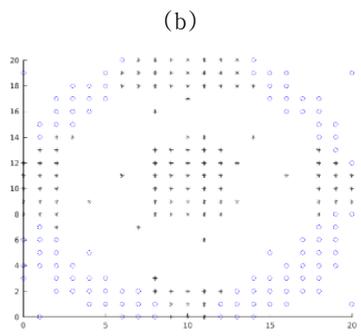
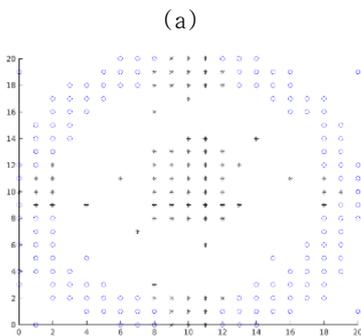


(a)

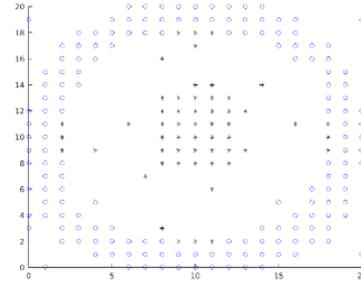




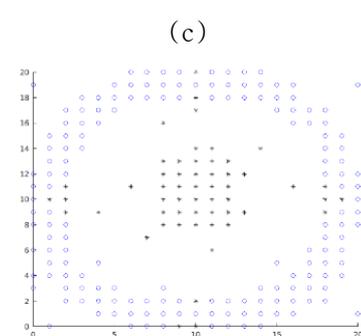
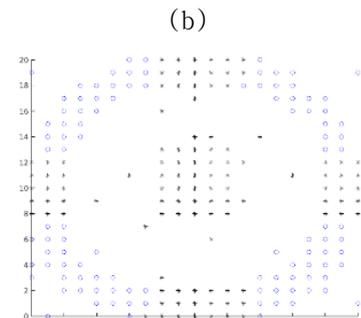
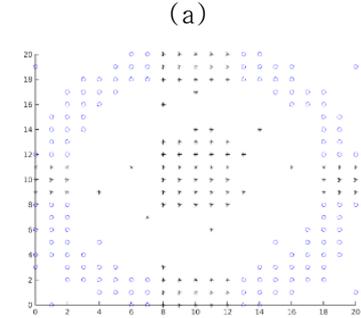
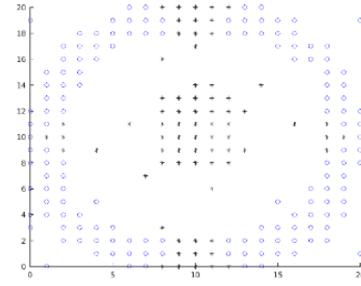
(그림 3) 데이터의 학습 진행과정($\gamma=0.2$) : ((a) epoch=5, (b) epoch=10, (c) epoch=15, (d) epoch=20)



(c)



(그림 4) 데이터의 학습 진행과정($\gamma=0.4$) : ((a) epoch=5, (b) epoch=10, (c) epoch=15, (d) epoch=20)



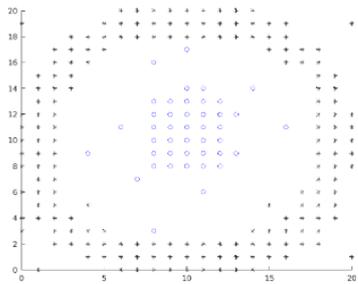
(그림 5) 데이터의 학습 진행과정($\gamma=0.8$) : ((a) epoch=5, (b) epoch=10, (c) epoch=15, (d) epoch=20)

3.2 반응영역의 단계적 학습실험

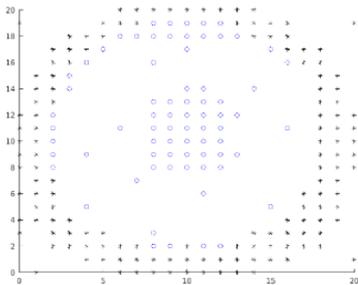
학습이 진행되는 과정에 대하여 표 2 와 같이 뉴런 반응영역을 단계별 설정($\gamma=0.2, 0.4, 0.8$)하여 학습을 진행시킨 과정을 그림 6 에 각각 보인다. 초기단계에는 주변의 반응영역을 보다 넓게 설정하고($\gamma=0.2$) 학습단계가 진행될수록 반응의 영역을 조금씩 줄임으로써($\gamma=0.4, \gamma=0.8$) 보다 세밀한 영역의 학습이 마무리되도록 조정하여 학습한다. 이러한 반응영역의 조정을 통한 단계적 학습은 고정적 반응영역의 학습과는 다르게 class1 과 class2 의 영역 데이터에 대한 학습이 적절하게 이루어지는 것을 볼 수 있다(그림 6(d)).

(표 2) 학습진행에 따른 반응영역 조절값(γ)

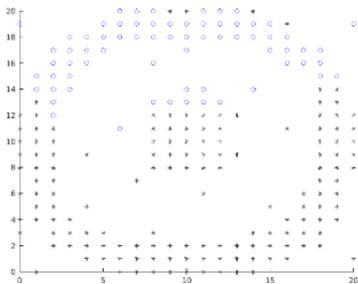
학습 진행	학습횟수 (비율)	γ
1단계	60 (20%)	0.2
2단계	160 (60%)	0.4
3단계	60 (20%)	0.8



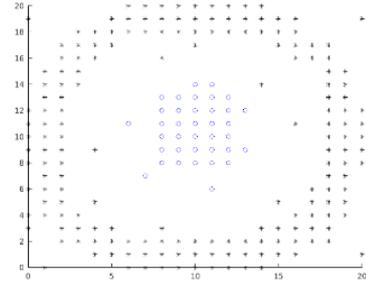
(a)



(b)



(c)



(d)

(그림 6) 데이터의 단계적 학습 진행과정($\gamma=0.2, 0.4, 0.8$) : ((a) epoch=5, (b) epoch=10, (c) epoch=15, (d) epoch=20))

4. 결론

제안한 학습 알고리즘은 기존의 구간연산 학습알고리즘과 동일하게 영역내의 많은 개수의 포인트 데이터를 작은 개수의 구간데이터로 변환 입력이 가능함으로써 학습의 시간을 줄일 수 있고 다양한 형태의 특정 영역학습에 쉽게 적용이 가능하다. 그러나, 기존의 구간연산 학습알고리즘을 개선하여 스파이킹 뉴런들의 반응영역 범위를 학습의 진행과정에 따라 조절이 가능한 단계적 구간연산 학습알고리즘이다.

이러한 성능의 개선 및 가능성을 확인하기 위하여 구간데이터와 포인트 데이터로 구성된 링형의 학습데이터에 대한 컴퓨터시뮬레이션 결과를 통하여 그 가능성을 보였다.

참고문헌

- [1] H.Fang, Y.Wang, J.He, and S.Liu, "Temporal pattern recognition using spiking neural networks for cortical neuronal spike train decoding", Proceedings of the 17th World Congress The international Federation of Automatic Control, 2008, 5203-5208.
- [2] S.Shrestha and Q.Song, "Adaptive Delay Learning in SpikeProp Based on Delay Convergence Analysis," 2016 International Joint Conference on Neural Networks (IJCNN 2016), 2016, 277-284.
- [3] S. Bothe, J. Kok, and J. Poutre, "Error-backpropagation in temporally encoded networks of spiking neurons", Neurocomputing, 48(1-4), 2002, 17-37.
- [4] K. Lee and H. Shi, "A Modular Approach to Construction of Spiking Neural Networks," 2019 International Joint Conference on Neural Networks (IJCNN 2019), Budapest, Hungary, 2019, 1-8.
- [5] G.Alefeld and J.Herzberger, Introduction to Interval Computations. Academic Press. NewYork.
- [6] B. Meftah, O. Lezoray, and A. Benyettou, "Segmentation and Edge Detection Based on Spiking Neural Network Model," Neural Processing Letters, vol. 32(2), 2010, 131-146.