

# 템포럴 코딩과 레이트 코딩을 이용한 Spiking Neural Network의 RF 신호 탐지 및 분류 성능 분석

이 현 중\*, 임 재 한<sup>o</sup>

## Performance Analysis of RF Signal Detection and Classification with Spiking Neural Network Using Rate Coding and Temporal Coding

Hyun-Jong Lee\*, Jae-Han Lim<sup>o</sup>

요 약

Spiking Neural Networks (SNNs)는 3세대 인공지능망으로 많은 연구자들의 주목을 받고 있다. SNN은 정보를 처리하기 위해 이산적인 스파이크와 막 전위가 임계치를 넘어야 정보를 전달하는 스파이킹 뉴런을 사용한다. 이로 인해 SNN은 다른 인공지능망들보다 에너지 효율성이 더 좋으며 순차적 데이터에 처리에 좋은 성능을 보인다고 알려져 있다. 하지만 현재 SNN의 응용 분야는 기존 인공지능망들이 자주 응용되는 분야인 이미지 분석에 국한되어 있으며 다른 응용 분야에서는 널리 쓰이지 못하고 있다. 따라서 본 논문에서는 SNN을 Radio Frequency (RF) 신호 탐지 및 분류에 사용하여 그 성능을 측정했다. 또한, RF 신호 스파이크 변환 인코딩에 레이트 코딩과 템포럴 코딩을 적용해서 각 코딩 방식의 성능과 계산 복잡도를 측정하여 Signal-to-Noise Ratio (SNR)에 따른 효율적인 신호 인코딩 방식을 제안한다.

**Key Words** : Spiking Neural Network, RF signal, Rate coding, Temporal coding, STDP, Memristive synapse

### ABSTRACT

Spiking Neural Networks (SNNs) are the third generation of neural networks and attract many researchers' attention currently. SNN uses discrete spikes and spiking neurons that deliver information when their membrane potentials reach threshold to process data. Due to this, SNN is more energy efficient than other neural networks and is known to have good performance with processing sequential data. However, SNN has been confined to image analysis like other neural networks and it is not being used in other applications. Therefore, in this paper, we propose to use SNN for detecting and classifying Radio Frequency (RF) signals and measure its performance. Furthermore, we apply rate encoding and temporal encoding for translating RF signals into spikes. We measure performance and computational complexity of each encoding scheme and propose efficient encoding scheme according to Signal-to-Noise Ratio (SNR).

\* 본 연구는 광운대학교 우수연구자 지원 사업 2021과 한국연구재단 신소자원첨기술개발사업 (grant no. NRF-2021M3F3A2A0103796 2)의 지원을 받아 수행된 연구입니다.

• First Author : Kwangwoon University Department of Software, hetzer44@kw.ac.kr, 학생회원

<sup>o</sup> Corresponding Author : Kwangwoon University Department of Sofatware, ljhar@kw.ac.kr, 정회원

논문번호 : 202211-263-A-RN, Received November 1, 2022; Revised November 28, 2022; Accepted November 29, 2022

## I. 서 론

인공신경망은 머신 러닝을 이용한 인공지능 전반에서 널리 사용되고 있다. 그 중 Deep Neural Networks (DNNs)는 현재 여러 분야에서 강력한 성능으로 주목을 받고 있다<sup>[1]</sup>. 하지만 DNN은 여러 계층의 신경망을 사용하여 연산을 수행하며 이로 인해 많은 에너지를 소모한다. 이는 DNN이 충분한 자원이 갖춰진 환경에서만 사용될 수 있도록 제한한다. 이러한 DNN의 한계로 인해 최근 많은 연구자들이 Spiking Neural Networks (SNNs)에 주목하고 있다. SNN은 퍼셉트론을 사용하는 다른 인공신경망들과는 다르게 스파이크 특성을 주로 사용하여 학습을 진행한다. 이는 SNN이 각 계층에 존재하는 퍼셉트론 간의 통신을 위해 실수를 사용하는 것이 아니라 어떠한 단위 시간 내에서 발생하는 스파이크들을 사용한다는 것을 의미한다. SNN에서 자주 사용하는 스파이킹 뉴런은 스파이크 특성을 통해 서로 정보를 주고받으며, 스파이크는 스파이킹 뉴런의 막 전위를 올린다. 이때, 막 전위가 임계치에 도달하면 뉴런들 사이의 가중치 값이 갱신되고 동시에 다음 계층의 뉴런으로 스파이크를 방출한다. 이러한 특성으로 인해 SNN의 각 계층은 입력을 받을 때마다 가중치를 갱신하는 다른 신경망들과는 다르게 입력을 매번 받을 때마다 가중치를 갱신할 필요가 없어 에너지 효율성이 다른 인공신경망에 비해 높다<sup>[2]</sup>.

이러한 장점에도 불구하고 현재 SNN은 DNN을 비롯한 다른 인공신경망만큼 널리 쓰이지 못하고 있는데 이는 크게 두 가지 이유 때문으로 볼 수 있다. 첫 번째로는 SNN의 계층 수를 늘리는 연구가 부족하여 DNN을 비롯한 다른 인공신경망만큼 높은 성능을 보이지 못하고 있으며 두 번째로는 SNN의 응용 및 연구 분야가 아직 기존 인공신경망이 자주 사용되는 분야인 이미지 분석에만 국한되어 있기 때문이다. SNN은 인공신경망의 에너지 효율성을 높이고 이를 하드웨어로 구현한 뉴로모픽 칩 개발에 핵심적인 기술이므로 현재의 문제점을 해결하고 개선할 필요가 있다.

이에 본 논문에서는 SNN이 계층 수를 늘리지 않고 기존의 저전력 기반 특성을 유지하면서 높은 성능을 보일 수 있음과 동시에 이미지 분석뿐만 아니라 다른 분야에서도 성공적으로 사용될 수 있음을 보이고자 SNN을 통해 Radio Frequency (RF) 신호를 탐지 및 구분하는 모델을 구현하여 Universal Software Radio Peripheral (USR)로 만든 RF 신호를 Signal-to-Noise Ratio (SNR)을 바꿔가며 탐지 및 구

분하였고 높은 탐지 및 구분 성능을 얻었다. 이에 더해 두 가지 뉴런 인코딩 방식으로 신호를 인코딩하고 각 인코딩 방식별 성능과 연산 요구량을 비교하여 신호의 SNR에 따른 적절한 뉴런 인코딩 방식을 알아내었다. 또한, 모델 구현 과정에 뉴로모픽 칩의 하드웨어적 특성과 핵심 가중치 업데이트 규칙인 Spike Timing Dependent Plasticity (STDP)를 반영하여 실제 뉴로모픽 칩이 RF 신호를 받았을 때 어떻게 작동하는지 예측하였다<sup>[3,4]</sup>. 따라서 본 논문의 의의는 다음과 같이 정리할 수 있다.

- SNN의 계층 수를 늘리지 않고 SNN의 에너지 효율성은 유지하면서 높은 성능을 보일 수 있는 응용 분야로 RF 신호 분석을 제시하였다.
- RF 신호 분석에서 인코딩 방식에 따른 SNN의 성능을 분석하고 상황에 따라 적절한 인코딩 방식을 제시하였다.
- 실제 뉴로모픽 하드웨어의 특징을 반영하여 시뮬레이션을 진행했으며, SNN이 실제 하드웨어 환경에서 어떻게 동작하는지 분석하였다.

본 논문의 구성은 다음과 같다. 2장에서는 SNN의 작동 방식과 원리를 설명한다. 3장에서는 SNN을 이용한 RF 신호 탐지 및 구분 모델의 구조와 작동 절차를 설명한다. 4장에서는 해당 모델을 이용한 RF 신호 탐지 및 구분 성능을 인코딩 방식을 바꿔가며 도출하고 이를 비교 및 분석한다. 5장에서는 실험 결과를 바탕으로 얻은 결론을 서술한다.

## II. Spiking Neural Network

### 2.1 스파이크와 스파이킹 뉴런

본 논문에서는 스파이킹 뉴런에 기반한 SNN을 사용하며 스파이킹 뉴런은 다른 인공신경망에서 널리 쓰이는 퍼셉트론과는 다르게 입력에 따라 자신의 막 전위 값을 조절하며 이 막 전위의 상태에 따라 스파이크를 출력 여부를 결정한다<sup>[5]</sup>. 스파이킹 뉴런에는 여러 가지 종류가 있으며 본 논문에서는 스파이킹 뉴런 중 가장 널리 쓰이는 방식인 Leaky Integrate and Fire (LIF) 뉴런을 사용하였으며<sup>[6]</sup>, 그림 1에서 LIF 뉴런의 작동 방식을 살펴볼 수 있다.

LIF 뉴런으로 입력이 들어오면 막 전위는 입력 스파이크가 들어올 때마다 올라가고 입력이 들어오지 않을 때는 막 전위의 누수로 인해 막 전위가 점점 감소한다. 입력 스파이크가 자주 발생하여 막 전위가 계

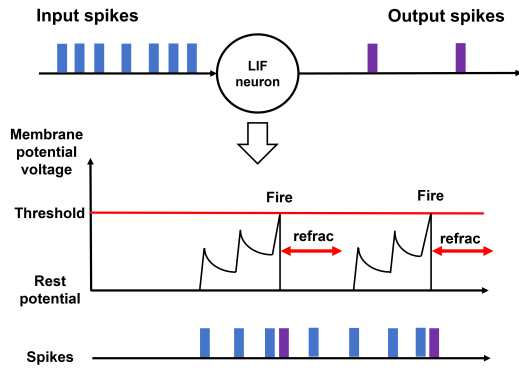


그림 1. LIF 뉴런의 작동 방식 및 절차  
Fig. 1. Operation method and sequence of LIF neurons

속 오르다가 특정 임계치에 막 전위가 도달하는 순간 해당 뉴런은 스파이크를 출력하고 원래의 막 전위 값으로 초기화된다. 이 과정을 발화라고 하며 출력된 스파이크는 다음 계층의 뉴런으로 전달되어 해당 뉴런의 막 전위를 올려주는 역할을 한다. 또한, 발화한 뉴런은 휴지기를 가지며 휴지기 동안은 입력 스파이크를 받더라도 막 전위를 올리지 않는다.

스파이킹 뉴런들 사이에서 스파이크가 전해질 수 있도록 뉴런들을 이어주는 신경 연결을 시냅스라 하며 스파이킹 뉴런들의 활동성은 시냅스의 연결 강도에 따라 조절된다. 시냅스의 연결 강도가 강해지면 해당 시냅스의 후 뉴런은 높은 활동성을 가지게 되어 이전보다 자주 발화하게 되며, 연결 강도가 약해지면 해당 시냅스의 후 뉴런은 낮은 활동성을 가지게 되어 이전보다 덜 발화하게 된다.

## 2.2 STDP 학습 규칙

SNN에서 시냅스 가중치를 입력 데이터를 통해 적절하게 조절하여 최적화하는 것을 학습이라 한다. 또한, 입력 데이터를 학습하는 여러 가지 방법들을 학습 규칙이라 하며 이러한 학습 규칙으로는 대표적으로 Spike-Timing-Dependent Plasticity (STDP), SpikeProp 그리고 Spatio-Temporal Backpropagation (STBP)가 있다<sup>7,8)</sup>. 본 논문에서는 STDP를 SNN 모델에 적용하여 RF 신호들을 탐지 및 구분했다. STDP는 생물의 뇌에서 기억이 생성되고 제거되는 과정을 모방한 학습 규칙으로 시냅스 전 뉴런과 후 뉴런 사이의 스파이크 발생 시간의 시간적 전후 관계에 따라 시냅스 가중치를 조절한다. 그림 2에서 볼 수 있듯이, 시냅스 전 뉴런이 발화하여 스파이크를 방출한 후 시냅스 후 뉴런이 발화하여 스파이크를 방출한 경우, 이는 Long Term Potentiation (LTP)라 하며 시냅스 가중치를 증

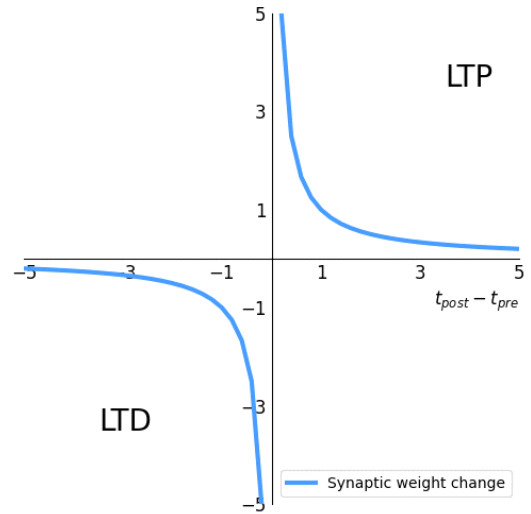


그림 2. STDP의 가중치 조절 방식.  
Fig. 2. Synaptic weight control method of STDP.

가시킨다. 이때 가중치가 증가하는 정도는 시냅스 전 뉴런과 후 뉴런의 스파이크 발생 시점 사이의 시간적 간격에 반비례한다. 시냅스 후 뉴런이 발화하여 스파이크를 방출하고 시냅스 전 뉴런이 발화하여 스파이크를 방출한 경우, 이는 Long Term Depression (LTD)라 하며 시냅스 가중치를 감소시킨다. 이때 가중치가 감소하는 정도는 LTP와 마찬가지로 시냅스 전 뉴런과 후 뉴런의 스파이크 발생 시점 사이의 시간적 간격에 반비례한다. 이러한 STDP의 학습 절차는 STDP가 다른 학습 방식에 비해 적은 입력 데이터로 학습을 빨리 진행할 수 있게 해준다<sup>9)</sup>.

SNN은 데이터를 전부 스파이크의 형식으로 나타내기 때문에 입력 데이터를 스파이크로 변환해주는 인코더가 필요하다. 인코더는 입력 데이터의 특징을 반영하여 스파이크를 생성하며 이를 다음 계층의 스파이킹 뉴런들에게 전달한다. 인코더가 데이터를 스파이크의 형태로 변환하는 방식은 크게 두 가지를 들 수 있다. 하나는 데이터의 크기를 스파이크의 발생 빈도로 나타내주는 레이트 인코딩이고 다른 하나는 데이터의 크기를 스파이크의 발생 시각으로 나타내주는 템포럴 인코딩이다. 레이트 인코딩 방식에서는 데이터의 크기가 크면 클수록 인코더에서 발생하는 스파이크의 빈도가 커진다. 반면에 템포럴 인코딩 방식에서는 데이터의 크기가 크면 클수록 인코더에서 발생하는 스파이크의 최초 발생 시각이 빨라진다. 본 논문에서는 레이트 인코딩 방식 중 푸이송 확률 분포를 이용해 데이터 크기에 따른 스파이크 발생 확률을 계산하

여 스파이크를 발생시키는 푸아송 인코딩<sup>[10]</sup> 방식과 템포럴 인코딩 중 하나의 데이터 샘플에 존재하는 입력값들 사이의 상대적 크기를 비교해 그 입력값의 크기가 클수록 스파이크를 먼저 발생시키는 Time-to-First Spiking (TTFS)<sup>[11]</sup> 방식을 적용하여 RF 신호를 탐지 및 분류했으며 두 인코딩 방식 사이의 성능과 에너지 효율성을 비교하였다.

### III. SNN 모델의 구조 및 원리와 RF 신호 탐지 및 분류 절차

#### 3.1 SNN 모델의 구조

본 논문에서 소개하는 SNN 기반 RF 신호 탐지 및 분류 모델은 오픈소스 SNN 시뮬레이터인 BindsNET을 기반으로 구현되었으며 구조는 그림 3과 같다<sup>[12]</sup>.

SNN 모델에 들어가는 입력 데이터는 입력 신호와 입력 신호 샘플의 종류를 특정해주는 마커로 이뤄져 있다. 입력 신호는 인코더로 전달되며 마커는 인코딩이 끝난 신호 샘플의 뒤에 붙어서 샘플의 종류를 특정해준다. 인코더와 10개의 입력 계층 뉴런은 입력 신호의 크기에 따라 정해진 인코딩 방식을 통해 입력 스파이크를 생성하여 흥분성 뉴런 계층으로 전달한다. 이때, 입력 뉴런의 개수는 샘플링된 샘플을 모두 입력받기 위해 10개로 설정되었다. 흥분성 뉴런 계층은 4개의 흥분성 뉴런으로 구성되어 있으며 흥분성 뉴런은 adaptive threshold가 적용된 LIF 뉴런이다. 흥분성 뉴런의 개수는 각 데이터 사이의 편차를 반영하여 성능을 높일 수 있으면서 복잡도는 낮게 유지할 수 있는 뉴런 개수인 4개로 설정되었다. 흥분성 뉴런에는 적응적 임계치 적용을 위한 변수로  $\theta_+$ 가 존재하는데, 이  $\theta_+$  값은 흥분성 뉴런이 발화할 때마다 임계치에 더해져서 흥분성 뉴런의 임계치를 서서히 올려주며 이는

흥분성 뉴런의 과도한 발화를 막는다.

흥분성 뉴런 계층에서 입력 데이터에 따라 흥분성 뉴런 각각의 발생 스파이크 개수를 측정한다. 이때 가장 많은 스파이크를 내보낸 뉴런은 해당 입력 데이터를 가장 잘 학습한 뉴런으로 간주하고 해당 데이터의 마커를 할당받는다. 마커를 할당받은 흥분성 뉴런은 마커가 특정하는 데이터를 입력받을 때마다 발화하게 되고 SNN 모델은 발화한 뉴런의 마커를 결과로 삼아 입력 데이터의 실제 마커와 비교한다. 이때 발화한 뉴런의 마커와 해당 입력 데이터의 마커가 서로 같으면 탐지 및 분류에 성공한 것으로 간주하며 반대의 경우 실패한 것으로 간주한다.

흥분성 뉴런에서 나온 스파이크들은 억제성 뉴런 계층으로 전달된다. 억제성 뉴런 계층은 흥분성 뉴런과 같은 4개의 뉴런으로 이뤄져 있으며 이는 억제성 뉴런이 흥분성 뉴런들을 균등하게 억제하기 위함이다. 억제성 뉴런 계층은 adaptive threshold가 없는 단순 LIF 뉴런으로 이뤄져 있으며 Winner Takes All (WTA) 메커니즘에 의해 가장 스파이크를 많이 방출하는 흥분성 뉴런으로부터만 스파이크를 전달받는다. 스파이크를 전달받은 억제성 뉴런의 막 전위는 증가하게 되고 임계치에 도달하는 순간 억제성 뉴런은 발화하게 된다. 억제성 뉴런은 발화할 때마다 자신과 연결된 흥분성 뉴런을 제외한 나머지 흥분성 뉴런들의 막 전위를 강제로 낮춰서 흥분성 뉴런들의 발화를 억제한다. 이를 lateral inhibition이라 하며 SNN 모델에서 입력 데이터에 대한 학습이 가장 잘 이뤄진 뉴런만 발화하여 해당 입력 데이터에 대응하는 마커만 결과로 출력될 수 있도록 해준다.

#### 3.2 SNN 모델의 STDP 학습 원리

본 논문의 SNN 모델에서 사용된 STDP는 멤리스터의 특성이 반영된 STDP를 사용했다. 해당 STDP의 작동 방식은 그림 4와 같다. SNN 모델에서 시냅스 전 뉴런은 입력 뉴런을, 시냅스 후 뉴런은 흥분성 뉴런을 의미한다. 시냅스 후 뉴런에서 스파이크가 발생한 경우, SNN 모델은  $\Delta t$  시간 이전 혹은 이후에 시냅스 전 뉴런에서 스파이크 발생 여부를 확인한다. 시냅스 전 뉴런에서 스파이크가 발생했을 경우 SNN 모델은 LTP가 발생한 것으로 판단하여 수식 1과 2에 따라 입력 스파이크가 발생한 시냅스 전 뉴런과 출력 스파이크가 발생한 시냅스 후 뉴런 사이의 시냅스 가중치를 증가시킨다. 이때, 시냅스 전 뉴런에서 입력 스파이크가 여러 번 발생했을 경우 해당 시냅스 가중치를 입력 스파이크가 발생한 횟수만큼 증가시킨다<sup>[12,13]</sup>.

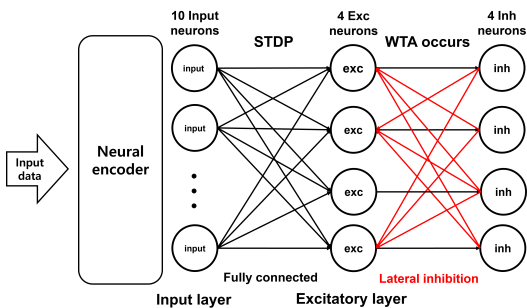


그림 3. SNN 기반 RF 신호 탐지 및 분류 모델의 구조.  
Fig. 3. Structure of RF signal detecting and classifying model based on SNN.

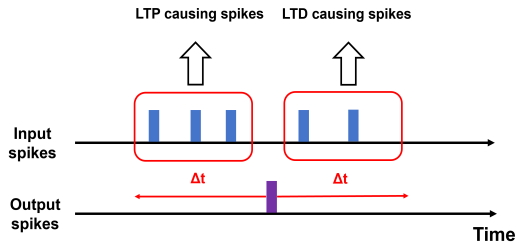


그림 4. SNN 모델에 적용된 STDP의 작동 방식.  
Fig. 4. Operation sequence of STDP in the SNN model.

$$\alpha = \frac{w_{max} - w_{min}}{1 - e^{-v}} \quad (1)$$

$$\Delta w = \frac{\Delta t}{t_{post} - t_{pre} + 1} (\alpha + w_{min} - w) (1 - e^{-\beta \frac{v_{LTP}}{256}}) \quad (2)$$

w는 현재 시냅스 가중치 값을 의미하고  $w_{max}$ 와  $w_{min}$ 은 각각 시냅스 가중치의 최댓값과 최솟값을 의미한다. 해당 논문의 SNN 모델에선  $w_{max}$ 를 0.5와 1 사이의 값 중 무작위로 선택하게 하였고  $w_{min}$ 은 0과 0.5 사이의 값 중 무작위로 선택하게 하였다.  $\Delta w$ 는 시냅스 가중치의 변화량을 의미하며 멤리스터의 특성을 반영한다.  $\Delta t$ 는 SNN 모델이 LTP와 LTD를 판단하는 기준 시간으로 50ms로 설정하였다.  $t_{post}$ 와  $t_{pre}$ 는 시냅스 후 뉴런의 스파이크 발생 시간과 시냅스 전 뉴런의 스파이크 발생 시간을 의미한다. v는 가중치 업데이트의 비선형성을 조절하는 변수로 LTP 발생 경우와 LTD 발생 경우 적용되는 v 값을  $v_{LTP}$ 와  $v_{LTD}$ 로 나누었다.  $\beta$ 는 LTP와 LTD 사이의 비선형성의 비율을 조절하는 변수이다.

$\Delta t$  시간 이후에 시냅스 전 뉴런에서 스파이크가 발생했을 경우 SNN 모델은 LTD가 발생한 것으로 판단하여 수식 1과 3에 따라 입력 스파이크가 발생한 시냅스 전 뉴런과 출력 스파이크가 발생한 시냅스 후 뉴런 사이의 시냅스 가중치를 감소시킨다. 이때 LTP의 경우와 마찬가지로 시냅스 전 뉴런에서 입력 스파이크가 여러 번 발생했을 경우 해당 시냅스 가중치를 입력 스파이크가 발생한 횟수만큼 감소시킨다<sup>13,14</sup>.

$$\Delta w = - \frac{\Delta t}{t_{pre} - t_{post} + 1} (\alpha - w_{max} + w) (1 - e^{-\frac{v_{LTD}}{256}}) \quad (3)$$

### 3.3 RF 신호 탐지 및 분류 절차

본 논문에서 사용된 SNN 모델의 RF 신호 탐지 및 분류 절차는 다음과 같다. 먼저, RF 신호 및 노이즈를 학습 데이터로 SNN 모델에 입력하면 입력된 RF 신호들은 인코더에 의해 스파이크의 형태로 변환된다. 스파이크 형태로 변환된 학습 데이터들은 입력 뉴런으로부터 방출되어 흥분성 뉴런들의 막 전위를 올린다. 막 전위가 올라가 발화한 흥분성 뉴런들은 STDP 학습 규칙에 따라 가중치를 업데이트하고 스파이크를 방출한다. 방출된 흥분성 뉴런들의 스파이크는 억제성 뉴런의 막 전위를 올리고 발화한 억제성 뉴런은 스파이크를 방출하며 가중치가 가장 올바르게 수렴한 흥분성 뉴런을 제외한 나머지 뉴런들의 활동을 억제하며 이로 인해 흥분성 뉴런들의 가중치가 입력 데이터의 특징 벡터를 잘 반영할 수 있도록 수렴된다.

위 과정을 통해 학습을 마친 SNN 모델은 실제 응용 단계에서 어떠한 신호를 입력받았을 때, 이를 스파이크 형태로 변환하고 저장된 가중치 벡터와 변환된 입력 데이터의 특징 벡터를 비교하여 입력받은 신호에 대한 올바른 탐지 및 분류 결과를 도출한다.

## IV. 실험 설정 및 결과

### 4.1 실험 설정

본 연구에서 진행한 RF 신호 탐지 및 분류 실험에 사용된 데이터셋들은 USRP를 통해 만들어졌다<sup>15</sup>. USRP는 송신과 수신 USRP로 나뉘며 송신과 수신 USRP 제어는 GNU radio를 통해 이뤄졌다. GNU radio의 블록들을 이용해 송신 및 수신 USRP의 중심 주파수 대역, 송신 및 수신 샘플링 레이트 그리고 게인 값을 선택하였다. 수신 USRP에서 수신된 신호는 MATLAB을 통해 데이터셋의 형태로 변환되어 저장된다. 위 과정에서 사인파, 사각파, 톱니파를 송신 USRP를 통해 RF 대역에서 만들고 이 신호들을 수신 USRP에서 수신한 값을 1kHz로 샘플링하여 10개의 진폭 값으로 기록했다.

SNR은 0dB, 6dB, 12dB, 18dB로 조절하였으며 데이터셋들은 탐지용 데이터셋과 분류용 데이터셋으로 나누어져 있다. 탐지용 데이터셋들은 사인파, 사각파, 톱니파 샘플들이 마찬가지로 10개의 진폭 값을 가지는 노이즈 샘플들과 1:1 비율로 섞여 있는 데이터셋이며 총 20000개의 노이즈와 신호 샘플들로 이뤄져 있다. 분류용 데이터셋들은 사인파, 사각파, 톱니파 샘플들이 두 종류씩 1:1 비율로 섞여 있는 데이터셋이며 탐지용 데이터셋과 똑같이 총 20000개의 샘플들로 이

표 1. 푸아송 인코딩 및 TTFS 인코딩을 적용했을 때 SNN 모델 변수들의 설정값 및 범위.  
Table 1. Range of parameters for Poisson and TTFS encoding.

	detection	classification
vLTP	0	0
vLTD	0	0
$\beta$	1	1
$\Theta+$	0.003	0.003
intensity	80~260	350
$\Delta s$	10ms	5ms
$\Delta t$	50ms	50ms

뤄져 있다. 해당 데이터셋들에서 샘플 부류별 비율을 1:1로 설정한 뒤 500개의 샘플들을 파이썬의 사이킷런 라이브러리를 통해 무작위로 균등하게 뽑아서 SNN 모델을 학습시켰으며 나머지 19500개의 샘플을 통해 SNN 모델의 성능 테스트를 진행하였다. 훈련 데이터의 샘플 개수와 테스트 데이터 샘플 개수는 SNN 모델이 실험 데이터에 과적합 되는 것을 방지하고, 동시에 적은 개수의 데이터로도 빠르게 학습이 진행되는 STDP의 특성을 반영하여 정해졌다<sup>9)</sup>.

SNN 모델을 통한 신호 탐지 및 분류 실험에서 사용한 변수들의 설정값 및 범위들은 표 1에 나타나 있다. intensity는 푸아송 인코더에서 데이터에 곱해지는 값으로 입력 뉴런에서 나온 스파이크가 흥분성 뉴런들을 발화시킬 수 있을 만큼 충분한 양을 만들어 낼 수 있도록 한다.  $\Delta s$ 는 TTFS 인코더에서 스파이크를 만들어낼 때의 간격이며 간격이 짧을수록 더 많은 스파이크를 내보낸다.

#### 4.2 실험 결과

SNN 모델의 사인파, 사각파, 톱니파의 인코딩 방식에 따른 탐지 성능은 정확도, Probability of Detection (PD), False Negative Probability (FNP), 그리고 False Positive Probability (FPP)를 통해 측정하였으며 SNR의 변화에 따른 탐지 및 분류 정확도는 그림 5~8에 나타나 있으며 SNR의 변화에 따른 PD, FNP, FPP 변화는 표 2, 3에 나타나 있다.

그림과 표를 통해 확인할 수 있듯이 푸아송 인코딩을 통해 신호를 인코딩한 경우, 세 가지 데이터셋 모두 0~18dB에서 80% 이상의 탐지 정확도와 75% 이상의 PD를 보였으며 12dB 이상에서는 100%에 근접한 탐지 정확도와 PD를, 0%에 근접한 FNP와 FPP를 보였다. 하지만 TTFS 인코딩을 통해 신호를 인코딩한 경우, 세 가지 데이터셋 모두 0~18dB에서 90% 이하

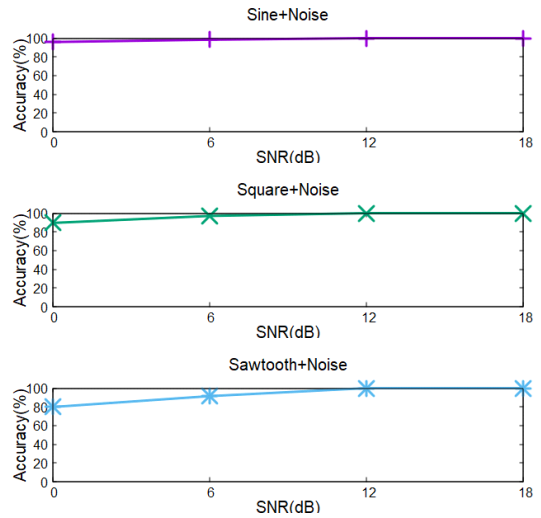


그림 5. 푸아송 인코딩 적용 시 SNN 모델의 신호 탐지 정확도.  
Fig. 5. Detection accuracy of SNN model with Poisson encoding.

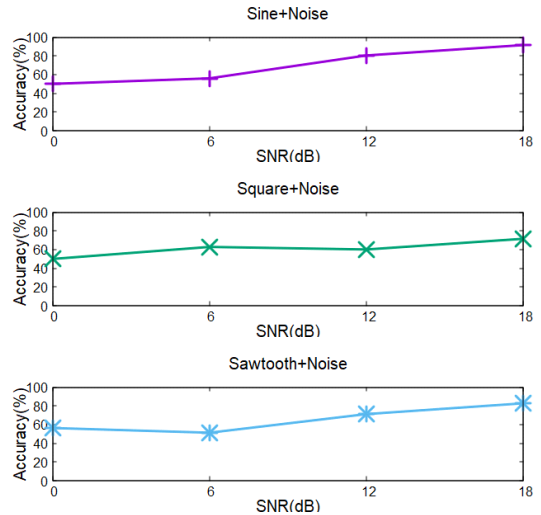


그림 6. TTFS 인코딩 적용 시 SNN 모델의 신호 탐지 정확도.  
Fig. 6. Detection accuracy of SNN model with TTFS encoding.

의 정확도를 보였으며 푸아송 인코딩과 비교했을 때, 전반적으로 더 낮은 PD와 높은 FNP 및 FPP를 보여 주고 있다.

이는 TTFS의 인코딩 방식에서 기인하는 문제이다. TTFS 인코딩은 샘플 내에서 10개의 각 진폭 값의 크기 차이에 따라 상대적으로 스파이크 발생 시작 시점을 정해준다. 이로 인해 노이즈와 신호가 절대적인 에너지 값은 노이즈에 비해 크더라도 각 진폭 값들의 크

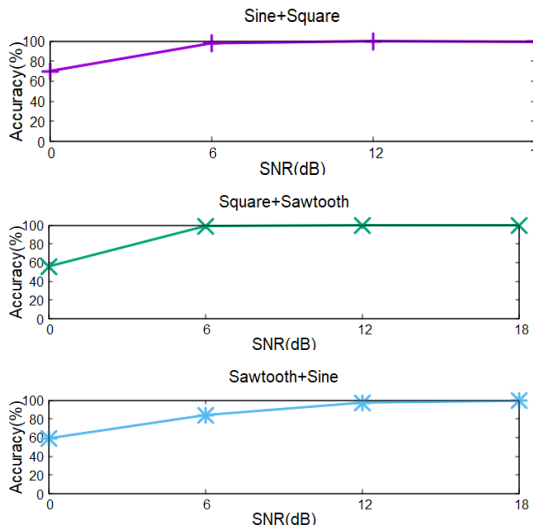


그림 7. 푸아송 인코딩 적용 시 SNN 모델의 신호 분류 정확도.  
Fig. 7. Classification accuracy of SNN model with Poisson encoding.

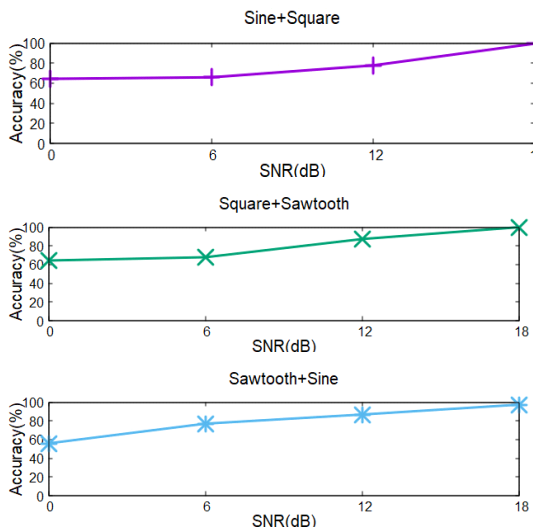


그림 8. TTFS 인코딩 적용 시 SNN 모델의 신호 분류 정확도.  
Fig. 8. Classification accuracy of SNN model with TTFS encoding.

기 차이가 비슷하면 유사한 패턴의 스파이크들로 인코딩될 수 있다. 하지만 푸아송 인코딩에서는 샘플들의 진폭 값이 가지는 절대적인 크기를 스파이크의 빈도로 인코딩하기 때문에 노이즈와 신호 사이의 절대적인 에너지 값 차이가 크면 상당히 다른 패턴의 스파이크들로 인코딩된다. 이 때문에, 푸아송 인코딩에서는 TTFS 인코딩에 비해 높은 탐지 성능을 보이며 RF

표 2. 푸아송 인코딩 적용 시 SNN 모델의 PD, FNP, FPP 변화.  
Table 2. PD, FNP and FPP Change of SNN model with Poisson encoding.

	0dB (PD, FNP, FPP)(%)	6dB (PD, FNP, FPP)(%)	12dB (PD, FNP, FPP)(%)	18dB (PD, FNP, FPP)(%)
sine	92.03, 7.97, 0	97.07, 2.93, 0	100, 0, 0	100, 0, 0
square	80.87, 19.13, 2.79	92.49, 7.51, 0	100, 0, 0	100, 0, 0
sawtooth	75.12, 24.88, 16.51	82.84, 17.16, 0	100, 0, 0	100, 0, 0

표 3. TTFS 인코딩 적용 시 SNN 모델의 PD, FNP, FPP 변화.  
Table 3. PD, FNP and FPP Change of SNN model with TTFS encoding.

	0dB (PD, FNP, FPP)(%)	6dB (PD, FNP, FPP)(%)	12dB (PD, FNP, FPP)(%)	18dB (PD, FNP, FPP)(%)
sine	99.99, 0.01, 99.97	55.88, 44.12, 44.06	78.73, 21.27, 19.87	90.39, 9.61, 7.27
square	99.94, 0.06, 100	64.16, 36.84, 37.46	57.64, 42.36, 37.44	69.43, 35.07, 26.29
sawtooth	59.99, 40.01, 47.43	99.48, 0.52, 96.92	100, 0, 57.56	68.55, 31.45, 3.15

신호 탐지에는 TTFS 인코딩 대비 푸아송 인코딩을 사용하는 것이 더 효과적인 것을 알 수 있다.

SNN 모델의 사인파, 사각파, 톱니파의 인코딩 방식에 따른 분류 성능은 정확도만을 통해 측정하였으며 SNR의 변화에 따른 분류 정확도는 그림 7과 8에서 확인할 수 있다. 그림들을 통해 확인할 수 있듯이 푸아송 인코딩을 통해 신호를 인코딩한 경우, 세 가지 데이터셋 모두 6dB 이상에서 80% 이상의 분류 정확도를 보였으며 12dB 이상에서는 100%에 근접한 분류 정확도를 보였다. 하지만 TTFS 인코딩을 통해 신호를 인코딩한 경우, 세 가지 데이터셋 모두 0~12dB에서 푸아송 인코딩에 비해 낮은 분류 정확도를 보여주며 18dB에서는 푸아송 인코더와 비슷한 분류 정확

도를 보여준다.

이는 신호 탐지의 경우와 마찬가지로 신호 분류에서도 TTFS 인코딩은 신호 샘플 내의 진폭 값들 사이의 상대적인 차이로 신호를 나타내기 때문에 샘플들 사이의 절대적인 차이를 나타내기 힘들어서 나타나는 현상이다. 또한, 푸아송 인코딩에서는 TTFS 인코딩과 비교했을 때 2~2.5배에 가까운 양의 스파이크를 이용해 입력 신호를 나타내기 때문에 TTFS 인코딩보다 데이터의 손실 없이 입력 신호를 더 자세히 나타낼 수 있다. 두 인코딩 방식에 따른 신호 분류 실험 시 스파이크 발생 개수의 차이는 표 4에 나타나 있다.

표 4에서 볼 수 있듯이 푸아송 인코딩은 평균적으로 TTFS 인코딩보다 더 많은 양의 스파이크를 사용하여 데이터를 인코딩하며 이는 푸아송 인코딩이 TTFS 인코딩보다 더 좋은 성능을 가질 수 있도록 해준다. 하지만 인코딩 과정에서 더 많은 입력 스파이크를 발생시킨다는 것은 더 많은 에너지를 인코딩 과정에서 이용한다는 것을 의미한다<sup>16)</sup>. 이는 푸아송 인코딩이 TTFS 인코딩보다 성능은 더 좋지만 그만큼 더 많은 에너지를 소모함을 나타낸다. 따라서 두 인코딩 방식을 적용하여 성능을 측정했을 때 서로 유사한 성능이 나온다면 TTFS 인코딩을 사용하는 것이 더 효율적이라고 할 수 있다. 본 연구에서 진행한 실험의 결과를 분석했을 때 신호를 탐지하는 상황과 18dB 미만에서 신호를 분류하는 상황에서는 푸아송 인코딩이 월등한 성능을 보이므로 푸아송 인코딩을 사용하는 것이 좋지만 18dB 이상에서 신호를 분류하는 경우, TTFS 인코딩을 사용하는 것이 더 효율적이라고 볼 수 있다. 표 5와 6은 18dB에서 푸아송 인코딩과 TTFS 인코딩의 신호 분류 성능과 입력 스파이크 개수를 보여준다.

표 5와 6에서 볼 수 있듯이 18dB에서 신호를 분류하는 경우 푸아송 인코딩과 TTFS 인코딩을 사용했을 때 모두 97%의 높은 분류 정확도를 보인다. 하지만 인코딩을 통해 만들어낸 입력 스파이크의 경우 푸아송 인코딩보다 TTFS 인코딩이 최대 3.5배 정도 적은

표 4. 모든 SNR에 대한 신호 분류 실험에서 인코딩 방식에 따른 입력 스파이크 개수의 평균값.  
Table 4. Number of Spikes according to encoding schemes in all SNR ranges.

	Poisson	TTFS
Sine + Square	1107	432
Square + Sawtooth	1003	551
Sawtooth + Sine	915	450

표 5. SNR이 18dB인 경우의 신호 분류 실험에서 인코딩 방식에 따른 SNN 모델의 분류 정확도.  
Table 5. Accuracy of SNN model according to encoding schemes when SNR is 18dB.

	Poisson	TTFS
Sine + Square	99.22%	99.45%
Square + Sawtooth	100%	99.99%
Sawtooth + Sine	99.38%	97.37%

표 6. SNR이 18dB인 경우의 신호 분류 실험에서 인코딩 방식에 따른 입력 스파이크 개수의 평균값.  
Table 6. Average spike numbers of SNN model according to encoding schemes when SNR is 18dB.

	Poisson	TTFS
Sine + Square	932	268
Square + Sawtooth	896	573
Sawtooth + Sine	718	348

것을 볼 수 있다. 이는 SNR이 18dB 이상인 상황에서는 TTFS 인코딩을 사용하는 것이 푸아송 인코딩보다 적은 에너지로 푸아송 인코딩과 유사한 높은 성능을 보여줌을 의미한다. 따라서 SNR이 18dB보다 높은 경우, 푸아송 인코딩보다는 TTFS 인코딩을 사용하여 신호를 인코딩하는 것이 더 효율적이라고 할 수 있다.

## V. 결론

본 연구에서는 SNN을 사용하여 RF 신호를 SNR과 인코딩 방식을 바꿔가며 탐지 및 분류해보면서 성능을 측정해 SNN이 기존의 응용 분야가 아닌 RF 신호 분석에도 응용될 수 있는지 확인해보고 SNR에 따라 알맞은 인코딩 방식을 보여주었다. SNN은 신호 탐지에서는 6dB부터 굉장히 높은 성능을 보였으며 분류에서는 12dB부터 굉장히 높은 성능을 보였다. 이는 SNN이 RF 신호를 적절하게 전처리해주면 낮은 SNR에서도 RF 신호를 좋은 성능으로 분석할 수 있음을 의미한다. 또한, RF 신호 탐지와 낮은 SNR에서의 RF 신호 분류에는 푸아송 인코딩이 유리하고 높은 SNR에서의 RF 신호 분류에는 TTFS 인코딩이 유리함을 보여서 SNR에 따라 SNN 기반 RF 신호 분석 기법을 더 효율적으로 사용할 수 있음을 증명하였다.

## References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp.



- 436-444, May 2015.
- [2] Z. Jingren, W. Jingjing, Y. Jingwei, W. Chunmao, and P. Shiliang, "Deep spiking neural network for high-accuracy and energy-efficient face action unit recognition," *2021 JCNN*, pp. 1-7, 2021. (<https://doi.org/10.1109/IJCNN52387.2021.9533451>)
- [3] S. Park and S. Yoon, "Analysis of the training performance and time of SNN by STDP algorithms and spike temporal interactions," *KIISE Trans. Comput. Practices*, vol. 24, no. 9, pp. 482-486, 2018.
- [4] C. I. Nwakanma, J.-W. Kim, J.-M. Lee, and D.-S. Kim, "Edge AI prospect using the NeuroEdge computing system: Introducing a novel neuromorphic technology," *ICT Express*, vol. 7, no. 2, pp. 152-157, 2021, ISSN 2405-9595.
- [5] J. Pérez, J. A. Cabrera, J. J. Castillo, and J. M. Velasco, "Bio-inspired spiking neural network for nonlinear systems control," *Neural Netw.*, vol. 104, pp. 15-25, 2018, ISSN 0893-6080.
- [6] J.-W. Han and M. Meyyappan, "Leaky integrate-and-fire biristor neuron," in *IEEE Electron Device Lett.*, vol. 39, no. 9, pp. 1457-1460, Sep. 2018. (<https://doi.org/10.1109/LED.2018.2856092>)
- [7] J. J. Wade, L. J. McDaid, J. A. Santos, and H. M. Sayers, "SWAT: An unsupervised SNN training algorithm for classification problems," *2008 IEEE Int. Joint Conf. Neural Netw. (IEEE World Congress on Computational Intell.)*, pp. 2648-2655, 2008. (<https://doi.org/10.1109/IJCNN.2008.4634169>)
- [8] P.-Y. Tan, C.-W. Wu, and J.-M. Lu, "An improved STBP for training high-accuracy and low-spike-count spiking neural networks," *2021 Design, Automat. & Test in Eur. Conf. & Exhibition (DATE)*, pp. 575-580, 2021. (<https://doi.org/10.23919/DATE51398.2021.9474151>)
- [9] H. Kim, H. Tang, W. Choi, and J. Park, "An energy-quality scalable STDP based sparse coding processor with on-chip learning capability," in *IEEE Trans. Biomed. Cir. and Syst.*, vol. 14, no. 1, pp. 125-137, Feb. 2020. (<https://doi.org/10.1109/TBCAS.2019.2963676>)
- [10] C. Johnson, S. Roychowdhury, and G. K. Venayagamoorthy, "A reversibility analysis of encoding methods for spiking neural networks," *The 2011 Int. Joint Conf. Neural Netw.*, pp. 1802-1809, 2011. (<https://doi.org/10.1109/IJCNN.2011.6033443>)
- [11] S. Oh, et al., "Spiking neural networks with time-to-first-spike coding using TFT-type synaptic device model," in *IEEE Access*, vol. 9, pp. 78098-78107, 2021. (<https://doi.org/10.1109/ACCESS.2021.3083056>)
- [12] H. Hazan, D. J. Saunders, H. Khan, D. Patel, D. T. Sanghavi, H. T. Siegelmann, and R. Kozma, "BindsNET: A machine learning-oriented spiking neural networks library in python," *Frontiers in Neuroinformatics*, vol. 2, 2018. <https://www.frontiersin.org/articles/10.3389/fninf.2018.00089>, (<https://doi.org/10.3389/fninf.2018.00089>, ISSN 1662-5196)
- [13] T. Kim, S. Hu, J. Kim, J. Y. Kwak, J. Park, S. Lee, I. Kim, J.-K. Park, and Y. Jeong, "Spiking neural network (snn) with memristor synapses having non-linear weight update," *Frontiers in Computational Neuroscience*, vol. 15, Mar. 2021, ISSN 1662-5188, (<https://doi.org/10.3389/fncom.2021.646125>)
- [14] S. Agarwal, et al., "Resistive memory device requirements for a neural algorithm accelerator," *2016 IJCNN*, pp. 929-938, 2016. (<https://doi.org/10.1109/IJCNN.2016.7727298>)
- [15] J. Muslimin, A. L. Asnawi, A. F. Ismail, and A. Z. Jusoh, "SDR based transceiver of digital communication system Using USRP and GNU radio," *2016 ICCCE*, pp. 449-453, 2016. (<https://doi.org/10.1109/ICCCE.2016.100>)
- [16] N. Abderrahmane and B. Miramond, "Neural coding: adapting spike generation for embedded hardware classification," *2020 IJCNN*, pp. 1-8, 2020.

(<https://doi.org/10.1109/IJCNN48605.2020.9207702>)

**이 현 중 (Hyun-Jong Lee)**



2024년 2월 : 광운대학교 소프트웨어학부 졸업 예정  
<관심분야> 무선 신호 처리, 머신러닝, 뉴럴 네트워크

**임 재 한 (Jae-Han Lim)**



2004년 2월 : 서울대학교 전기공학부 공학사

2006년 2월 : 서울대학교 전기컴퓨터공학부 공학석사

2014년 12월 : UCLA Electrical Engineering 공학박사 한국 전자통신연구원 선임연구원

광운대학교 소프트웨어학부 부교수 (현)

<관심분야> 스파이킹 뉴럴 네트워크, 차량용 네트워크, 자율주행, 모바일 컴퓨팅, 기계학습, 백스캐터 통신