

논문 2021-2-6 <http://dx.doi.org/10.29056/jsav.2021.12.06>

# 뉴로모픽 구조 기반 IoT 통합 개발환경에서 SNN 모델을 지원하기 위한 인코더/디코더 구현

김희남\*, 윤영선\*†

## Implementation of Encoder/Decoder to Support SNN Model in an IoT Integrated Development Environment based on Neuromorphic Architecture

Hoinam Kim\*, Young-Sun Yun\*†

### 요 약

뉴로모픽 기술은 인간의 뇌 구조와 연산과정을 하드웨어로 모방하는 기술로 기존 인공지능 기술의 단점을 보완하기 위하여 제안되었다. 뉴로모픽 하드웨어 기반의 IoT 응용을 개발하기 위해 NA-IDE가 제안되었으며, NA-IDE에서 SNN 모델을 구현하기 위하여 일반적으로 많이 사용되는 입력 데이터를 SNN모델에 사용할 수 있도록 변환이 필요하다. 본 논문에서는 이미지 데이터를 SNN 입력으로 사용하기 위하여 스파이크 시계열 패턴으로 변환하는 신경코딩 방식의 인코더 컴포넌트를 구현하였다. 디코더 컴포넌트는 SNN 모델이 스파이크 시계열 패턴을 생성하는 경우, 출력된 시계열 데이터를 다시 이미지 데이터로 변환하도록 구현하였다. 디코더 컴포넌트는 출력 데이터에 인코딩 과정과 동일한 매개변수를 사용한 경우, 원본 데이터와 유사한 정적 데이터를 얻을 수 있었다. 제안된 인코더와 디코더를 사용하면 image-to-image나 speech-to-speech와 같이 입력 데이터를 변환하여 재생성하는 분야에 사용할 수 있을 것이다.

### Abstract

Neuromorphic technology is proposed to complement the shortcomings of existing artificial intelligence technology by mimicking the human brain structure and computational process with hardware. NA-IDE has also been proposed for developing neuromorphic hardware-based IoT applications. To implement an SNN model in NA-IDE, commonly used input data must be transformed for use in the SNN model. In this paper, we implemented a neural coding method encoder component that converts image data into a spike train signal and uses it as an SNN input. The decoder component is implemented to convert the output back to image data when the SNN model generates a spike train signal. If the decoder component uses the same parameters as the encoding process, it can generate static data similar to the original data. It can be used in fields such as image-to-image and speech-to-speech to transform and regenerate input data using the proposed encoder and decoder.

**한글키워드** : 뉴로모픽, 스파이킹 뉴럴 네트워크, 컴포넌트, NA-IDE, 인코더, 디코더

**keywords** : Neuromorphic, Spiking Neural Network, Component, NA-IDE, Encoder, Decoder

\* 한남대학교 정보통신공학과

† 교신저자: 윤영선(email: ysyun@hmu.kr)

접수일자: 2021.11.12. 심사완료: 2021.12.03.

게재확정: 2021.12.20.

## 1. 서론

인공지능 기술은 전통적인 확률·통계적 방식의 패턴 인식뿐만 아니라 다양한 분야의 응용 분야에 적용하여 우수한 성능을 보인다. 하지만 핵심 기술인 딥러닝 기술은 대용량의 데이터와 많은 연산 과정이 필요하여, 저전력, 소규모 장치에서의 적용에 한계를 가져왔다. 기존 인공지능 기술의 단점을 극복하고자 뉴로모픽 구조가 제안되었다[1].

뉴로모픽 구조 방식은 인간의 신경·생물학적 구조와 특성을 물리적인 하드웨어로 모방하여 제안되었다. 뉴로모픽 구조에는 뉴런의 역할을 하는 코어가 존재하며 코어와 메모리칩의 병렬연결 구조는 대량의 연산을 동시 처리하는 데 유리하다[2].

인공지능기술의 발달과 4차산업혁명의 확산으로 사물 인터넷 환경(IoT, Internet of Things)에서 인공지능, 특히 인공신경망을 활용하는 사례가 증가하고 있다. 전력 소모 및 크기의 제한이 존재하는 IoT 환경에서, 인간의 뇌 연산 과정을 모방하여 펄스 형태로 시냅스에 인가하는 스파이킹 신경망 (Spiking Neural Network, SNN) 모델은 기존 인공신경망의 대안이 되고 있다[3].

IoT 환경과 뉴로모픽 기반의 인공신경망을 접목한 서비스를 개발하기 위한 단일 플랫폼의 개발 환경이 필요하게 되어 뉴로모픽 구조 기반 통합개발환경(Neuromorphic Architecture based Integrated Development Environment, NA-IDE)이 제안되었다[4]. NA-IDE는 미리 구현된 컴포넌트들을 배치하여 플로우(Flow)를 구성하고 데이터 흐름을 제어하여 IoT 서비스를 개발한다. 이 과정에서 기존의 IoT 컴포넌트나 전통적인 인공지능 컴포넌트 외에도 뉴로모픽 기반의 SNN 컴포넌트가 필요하다. SNN 컴포넌트는 기존 컴포넌트에서 사용하는 데이터와 달리 스파이크 시계열 패턴을 사용하기 때문에, 일반 IoT 환

경에서 수집된 데이터를 스파이크 시계열 패턴으로 변환하는 과정이 필요하다. 이를 위하여 본 논문에서는 이미지 데이터를 스파이크 시계열 패턴으로 변환하기 위하여, 신경코딩 방식의 인코더와 디코더 컴포넌트를 구현하였으며 사용자가 선택하는 변환 기술을 통해 이미지-스�파이크 데이터 변환을 지원하도록 하였다.

본 논문의 구성은 다음과 같다. 2장에서 본 연구의 기반이 되는 뉴로모픽 구조를 간략하게 소개하고, 뉴로모픽 구조를 구성하는 SNN 인공 신경망 기술과 뉴로모픽 환경에서 IoT와 SNN을 통합 개발하는 NA-IDE, 데이터 변환기법인 신경코딩기술에 대해 설명한다. 3장에서는 인코더와 디코더 컴포넌트의 구현과정을 기술하며, 4장에서 구현된 컴포넌트를 이용한 NA-IDE 동작 예시를 보이고, 마지막으로 결론을 맺는다.

## 2. 관련 연구

본 장에서는 뉴로모픽 구조 및 이를 구성하는 SNN, 기존 IoT와 뉴로모픽 환경을 지원하는 통합개발환경인 NA-IDE, NA-IDE에서 SNN 컴포넌트를 이용하는 데 필요한 데이터 변환에 관해 기술한다.

### 2.1. 뉴로모픽 구조

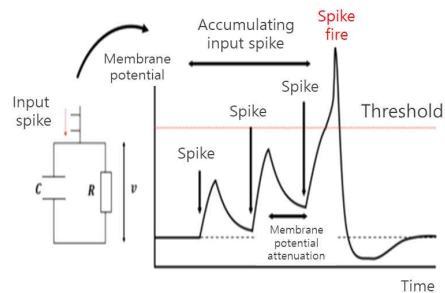


그림 1. 뉴로모픽 구조와 동작과정[5]  
Fig. 1. Neuromorphic architecture and operation process[5]

단순화된 뉴로모픽 구조는 그림 1과 같이 인간의 신경·생물학적 구조와 특성을 물리적인 하드웨어로 모방하여 제안되었다[6]. 인간의 뇌를 구성하는 뉴런은 스파이크 (spike)라고 불리는 작은 전기 신호를 생산하고, 누적된 신호가 임계값(Threshold)보다 큰 시점에 다른 뉴런으로 스파이크를 전달한다[7].

뉴로모픽 하드웨어는 전기 신호를 이용하여 스파이크를 모방하고 축전기(Capacitor)를 통해 누적된 전기 신호가 임계값을 넘어 다른 뉴런으로 스파이크 신호를 전달하도록 구현된다. 뉴로모픽 구조에는 뉴런의 역할을 하는 코어가 존재하며 코어와 메모리칩의 연결된 구조가 병렬로 이루어져 있어 다수의 연산을 동시에 처리하는데 유리하다. 따라서 기존 폰노이만 컴퓨팅 구조보다 빠른 학습·연산 속도를 가지며 전력의 소모량을 최소화할 수 있다.

## 2.2 스파이킹 신경망 (SNN)

SNN은 인간의 뉴런과 시냅스의 신호 전달과정을 전기적 신호의 전달과정으로 모방한다. SNN에서 스파이크의 발생 여부는 생물학적 뉴런에서의 정보 전달을 모방하는 방식으로 미분방정식에 의해 결정된다[8].

인간의 뇌는 약 100억개 이상의 뉴런과 각 뉴런마다 1,000-10,000개의 시냅스들이 회로망으로 이루어져 있다. 뇌의 정보처리과정은 뉴런이 환경으로부터의 감각자극이나 신경계 내에서 이전 단계의 뉴런으로부터 신호를 입력받아 자극에 포함된 정보를 전기화학적 활동으로 표현하는 과정에서 1 또는 0의 반응을 출력한다[9]. SNN의 동작을 모델링하는데, 일반적으로 사용되는 그림 2의 LIF (Leaky integrate-and-fire)[10] 뉴런 반응 모델에서와 같이 임계값을 넘어설 때만 활동전위를 발생시켜 다음 뉴런으로 스파이크 신호를 전달한다. SNN 동작의 핵심은 뉴런의 막 전위

(Membrane potential)를 이용한 스파이크 신호 전달 과정이다[11]. 뉴런에 전달되는 스파이크의 값은 시간에 따라 누적되고 특정한 전위에 도달하면 스파이크를 발생시키고 해당 뉴런의 값을 초기화한다.

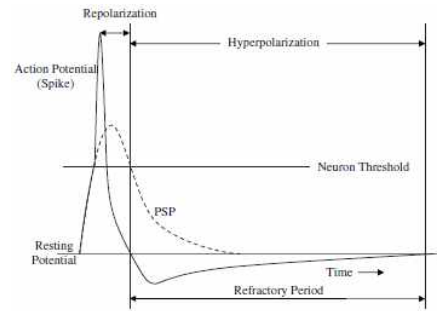


그림 2. SNN 동작 모델 (LIF)[12]  
Fig. 2. Operation model of SNN (LIF)[12]

기존 신경망은 연속적인 값을 출력하는 반면, SNN은 0 또는 1의 이진 값을 결과로 출력한다. 기존 신경망에서 출력 값 자체를 정보로 사용하였다면 스파이킹 신경망은 값이 출력되는 시계열 패턴을 학습에 사용한다[13]. 따라서 SNN을 기존 환경에서 구동하기 위해서는 기존 신경망에서 사용하던 데이터를 시계열 패턴 데이터로 변환하여야 한다.

## 2.3 NA-IDE

지능형 IoT 및 뉴로모픽 응용 개발을 위해 단일 플랫폼이 필요하며, 이를 위하여 뉴로모픽 기반 통합 개발환경인 NA-IDE가 제안되었다. NA-IDE는 사전에 구현된 컴포넌트들을 배치하여 플로우를 구성하고 데이터 흐름을 제어하여 서비스를 개발한다. NA-IDE에서 제공하는 컴포넌트를 이용하면 사용자가 세부 구현 기술을 모르더라도 목적에 맞는 서비스를 개발할 수 있다. 즉, 사용자는 관련 컴포넌트를 서비스 지원에 맞

는 데이터 흐름 (ex.입력-기능-출력)을 연결하고 매개 변수 설정만으로 원하는 서비스를 동작시킬 수 있다. 그림 3은 NA-IDE의 개발 및 작동 화면을 보여준다.

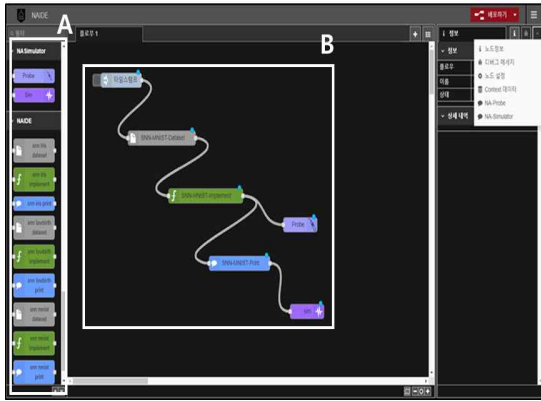


그림 3. NA-IDE 작동 예시  
Fig. 3. NA-IDE Screen shot

그림 3에서 A 상자는 사전에 구현되어 설치된 컴포넌트들을 표시한다. 사용자는 컴포넌트를 선택하여 그림 3의 B 상자과 같이 배치하고 연결하여 데이터 플로우를 구성하며 기본적으로 입력-동작-출력의 구조를 갖는다. 컴포넌트는 사용자에게 제공될 수 있도록 사전에 구현함으로써 사용자는 컴포넌트 저장소에서 선택하여 설치한 후, 배치하고 연결하는 것으로 필요한 서비스를 구축할 수 있다. NA-IDE는 SNN 모델의 구성을 제공하기 위해 지능형 컴포넌트인 SNN 컴포넌트를 제공한다. SNN 컴포넌트는 기존 컴포넌트에서 사용하는 데이터와 달리 스파이크 시계열 패턴을 사용하기 때문에, 일반 IoT 환경에서 수집된 데이터를 스파이크 시계열 패턴으로 변환하는 과정이 필요하다. 본 연구에서는 인코더 컴포넌트와 디코더 컴포넌트를 통해 기존 컴포넌트에서 사용하는 정적 데이터를 스파이크 시계열 패턴으로 변환 및 복원하는 기능을 제공한다.

## 2.4 신경 코딩

신경 코딩이란 뉴런이 자극의 정보를 활동 전위로 표현하는 규칙이라고 볼 수 있으며 전달하는 정보를 정량화하는 것을 의미한다. 신경코딩에서 인코딩은 자극에서부터 임계값을 달성한 반응까지의 과정을 나타내며, 자극의 성질과 시간에 따라 변하는 자극의 정보를 활동 전위로 구조화 한다[14]. 디코딩은 뉴런에서 전달받은 1 또는 0의 활동 전위 열 정보를 시간에 따라 또는 전달 받은 활동 전위의 발생횟수 등을 기준으로 전달 받기 전의 자극의 정보를 해석하는 과정이다[15]. 본 장에서는 일반적으로 많이 사용되는 코딩 방식을 소개한다.

### 가) Rate Coding

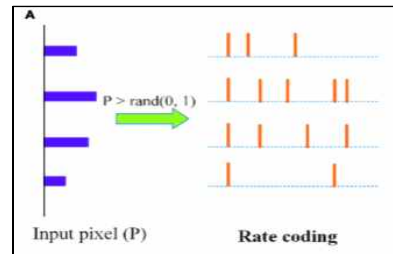


그림 4. Rate Coding방식을 이용한 스파이킹 발화[16]

Fig. 4. Spiking fire using Rate Coding

주파수 코딩이라고도 불리며 자극의 강도가 증가하면 뉴런의 활동 전위 및 스파이크의 발화 횟수가 증가한다는 특성을 이용한다. 뉴런의 발화 속도가 자극에 대한 정보를 포함한다고 가정하므로, 스파이크 발생률은 자극의 강도에 따라 비선형적으로 증가하며 정확한 스파이크 발생률에 대한 계산이 필요하다. 일반적으로 단위 시간당 평균 스파이크 발생률로 표현하며, 반복에 대한 평균으로 계산한다. Rate Coding은 상대적으로 다른 코딩 방식에 비하여 비효율적이지만 ISI (Inter Symbol Interface) 노이즈에 강력하다고

보고되었다[17]. 그림 4는 Rate Coding을 이용하여 이미지의 강도에 따라 다른 스파이크 데이터로 변환과정을 보인다.

의 다른 속성들을 동시에 표현할 수 있다는 이점이 있다.

나) Poisson Distribution

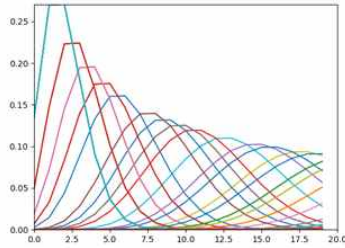


그림 5. 포아송 분포 방식을 이용한 스파이킹 발화

Fig. 5. Spiking fire using Poisson Distribution

주파수 코딩에 기반하여, 자극의 강도를 표현하는 단계에서, 단위시간 안에 스파이크가 몇 번 발화할 것인지를 그림 5와 같이 포아송 (Poisson) 분포 형태로 표현한 방식이다.

변환하고자 하는 이미지의 각 픽셀은 식(1)의 포아송 분포 기반으로 스파이크 데이터를 생성하고 발화 확률은 데이터의 각 픽셀 값에 비례한다.

$$f(k;\lambda) = \frac{\lambda^k e^{-\lambda}}{k!} \quad (\text{식1})$$

다) Population Coding

군집 코딩 (population coding)은 단일 뉴런이 아닌 군집 뉴런에 대한 코딩기법으로, 다수의 뉴런 응답을 이용하여 자극을 표현한다. 자극에 대해 뉴런별로 서로 다른 응답을 가지며 다수의 응답을 합산하여 자극에 대한 반응을 표현한다[18]. 그림 6은 자극에 대한 다수의 뉴런에 의한 응답 군집으로 표현됨을 보여준다. 군집 코딩은 뉴런의 가변성으로 인한 불확실성을 감소시키고 자극

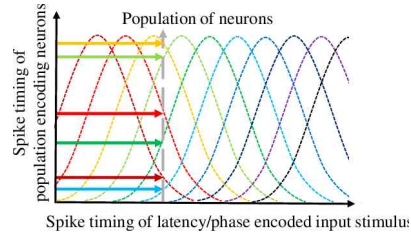


그림 6. Population Coding 방식을 이용한 스파이킹 발화[19]

Fig. 6. Spiking fire using Population Coding

### 3. 인코더 및 디코더 컴포넌트 구현

본 논문에서는 사람의 신경세포를 모방하는 SNN 모델의 데이터 전달 방식을 구현하고 스파이크 신호 입력을 지원하기 위해 인코더 컴포넌트와 디코더 컴포넌트를 구현하였다. 각 컴포넌트는 이미지 데이터를 스파이크 신호로 변환하고 변환된 스파이크 데이터를 이미지 데이터로 복원하는 기능을 제공한다. 그림 7은 인코더와 디코더를 이용한 데이터 변환 및 복원과정을 보인다.

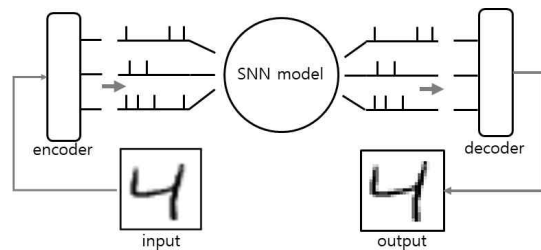


그림 7. 인코더와 디코더를 이용한 데이터 변환  
Fig. 7. Overview of data conversion using encoder / decoder

### 3.1 인코더 컴포넌트

NA-IDE 환경에서 그림 8과 같이 인코더 컴포넌트를 구현하였다. SNN 모델의 입력으로 사용하기 위해 기존의 인공신경망에서 쓰이던 이미지를 시계열 정보로 변환한다. 신경코딩 기법을 기반으로 SNN 모델에 따라 적용할 수 있는 3 가지의 코딩 방식을 제공하며, 각 변환 방식별로 매개 변수를 설정하여 데이터 변환을 조절할 수 있다. 공통 매개변수인 시간 간격은 스파이크 신호의 발화 시점을 조정하며, 시간 간격이 증가할수록  $n$  번째 스파이크와  $n+1$  번째 스파이크 사이의 지연시간을 증가시킨다.



그림 8. NA-IDE 환경에서 구현한 인코더 컴포넌트  
Fig. 8. Implemented encoder component on NA-IDE

그림 8의 왼쪽 영역은 현재 구성된 데이터 플로우를 표시하고 있으며, 오른쪽 영역은 인코더 컴포넌트의 속성 창이다. 그림 8에서 A 상자는 인코딩 방식을 선택할 수 있으며 단순 인코딩, Poisson 인코딩, Population 인코딩으로 3가지 인코딩 방식을 선택할 수 있다. 그림 8의 B 상자를 통해 사용자가 선택한 인코딩 방식에 필요한 매개변수 값을 입력할 수 있다. 인코더 컴포넌트의 입력으로 사용되는 이미지 데이터는 이미지의 크기에 따라  $N \times M$ 의 픽셀 수를 가지며 각 픽셀은 0~255 사이의 값으로 표현된다. 인코딩 과정

을 통해 변환된 시계열 데이터는 입력된 2차원 데이터에서 시간 간격으로 표현된 3차원 텐서 형태의 연속적인 데이터로 변환된다. 즉 하나의 픽셀 값은 시간 간격별 0 또는 1의 연속된 값으로 표현되며 각 시점에서 뉴런의 발화 확률을 나타낸다.

#### 3.1.1 단순 인코딩 방식

단순 인코딩 방식은 입력받은 이미지를 랜덤하게 생성한 동일 사이즈 필터 값을 통과시켜 임계값을 넘는 픽셀에 해당하는 스파이크 값을 얻는다. 그림 9와 같이 이미지와 동일한 크기의 필터를 생성하고 이미지의 각 픽셀에 1대1로 대응되는 값을 무작위로 생성한다. 입력받은 이미지는 각 필터 값과 비교하여 필터보다 큰 값만 통과시켜 스파이크 데이터로 변환을 진행한다. 사용자는 픽셀 값을 표현하는 무작위 수의 생성 분포를 설정하고 필터의 임계값을 높게 설정하면, 픽셀 값이 확률적 특성을 갖는 스파이크 시계열을 얻을 수 있다.

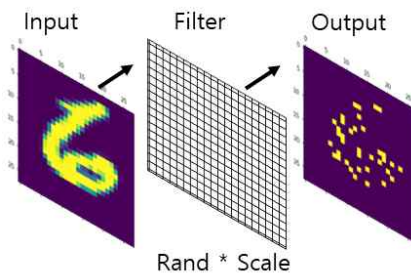


그림 9. 단순 인코딩 방식을 이용한 데이터 변환과정

Fig. 9. Data conversion using simple encoding method

#### 3.1.2 Poisson 인코딩 방식

입력 강도를 기반으로 포아송 분포가 적용된 스파이크 데이터를 생성한다. 입력된 데이터 사

이즈와 픽셀 값을 받아온 후 강도 함수를 이용하여 발화 속도를 초 단위로 계산한다. 식(2)의 포아송 분포 계산식을 이용하여 스파이크 확률 분포를 생성하고 스파이크 구간을 샘플링 한다. 스파이크 구간 사이의 간격이 0이 되지 않도록 구간은 1씩 증가시키고, 구간별 스파이크를 누적 합산하여 픽셀 값에 대응되는 스파이크 데이터를 생성한다. 그림 10은 포아송 인코딩 방식을 이용하여 이미지 데이터로부터 스파이크 시계열 데이터로 변화하는 과정을 보여준다.

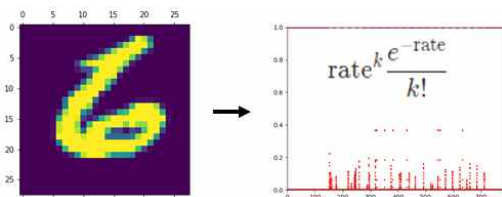


그림 10. 인코더 컴포넌트의 포아송 인코딩 방식을 이용한 데이터 변환 과정  
Fig. 10. Data conversion using Poisson encoding method

### 3.1.3 Population 인코딩 방식

뉴런의 수가 입력되면, 군집 방식으로 서로 다른 뉴런들의 자극에 대한 다른 반응 특성을 가진 신경세포들의 반응을 선형 결합 형태로 표현하는 방법이다. 다수의 응답에서 평균값은 자극 강도에 따라 선형적으로 변하는 가우스 튜닝 곡선 (Gaussian Tuning Curve)을 포함하며 평균 근처의 자극에 가장 강하게 반응한다. 그림 11은 다수의 뉴런 응답을 선형화 하는 과정을 보여준다.

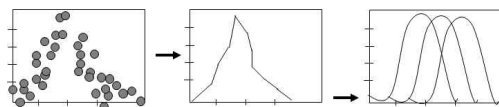


그림 11 군집 방식을 이용한 데이터 변환의 선형 결합 과정  
Fig. 11. Linear combination in Population encoding

### 3.2 디코더 컴포넌트

인코딩 과정에서 시계열 패턴 정보로 표현된 데이터는 SNN 모델의 입력으로 사용될 수 있다. SNN 모델에서 학습 및 평가의 결과로 클래스 분류가 아닌 이미지, 음성 등의 입력에서 변환된 데이터가 반환될 수 있으며 입력과 동일한 스파이크 형태의 시계열 패턴이 생성될 수 있다. SNN 모델에서 생성된 시계열 데이터는 다른 SNN 모델의 입력으로 사용할 수도 있으며, 다른 IoT나 인공지능 컴포넌트의 입력으로 사용될 수 있다. 동일한 SNN 모델이 아닌 경우에는 원래 시계열 패턴으로 변환되기 이전의 입력 데이터와 동일 형식으로 복원되어야 한다.

변환된 스파이크 데이터를 디코딩하여 얻은 복원 데이터를 인코딩에 사용된 입력 이미지와 비교하여 유사도를 확인하기 위해 인코딩 과정에서 사용된 동일한 MNIST 손글씨 이미지를 인코딩한 스파이크 데이터를 이용하여 디코딩 과정을 진행하였다. 디코딩 결과 그림 12와 같이 복원되는 결과를 확인했다.

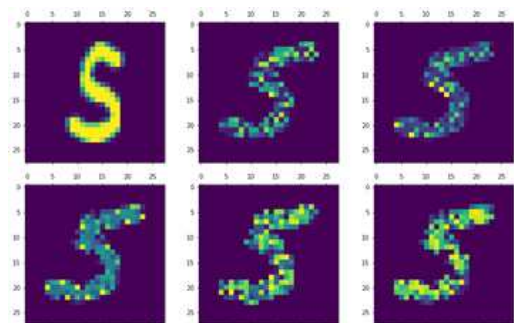


그림 12. 다양한 매개 변수에 따른 디코딩 이미지  
Fig. 12. Image decoding according to various parameters

인코딩 과정에서 사용된 매개 변수를 사용하지 않고 디코딩을 수행할 경우에는 전달받은 시계열 패턴에서 인코딩 매개 변수들을 추정해야

한다. 이를 확인하기 위하여 본 논문에서는 디코딩 과정에서 인코딩 매개 변수에 따른 입력 이미지의 복원 상태를 확인하였다. 그림 12의 첫 번째 이미지는 인코딩 과정에서 사용한 것과 동일한 입력 이미지이며, 나머지 이미지는 인코딩 과정을 거쳐 변환된 스파이크 데이터를 이용하여 인코딩 매개 변수를 변화시켜 디코딩으로 복원한 예이다.

반면 이미지를 시계열 패턴으로 변환하는 인코딩 매개 변수를 디코딩 과정에 이용하여 복원한 결과 그림 13과 같이 그림 12에 비하여 원본 데이터와 매우 유사한 이미지를 얻을 수 있었다.

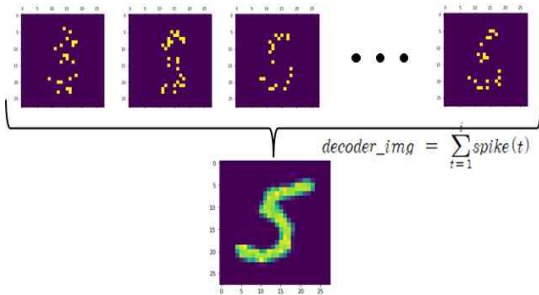


그림 13. 인코더와 동일한 매개변수 값을 사용한 디코딩 이미지

Fig. 13. Image decoding with the same encoding parameters

스파이크 시계열 패턴으로부터 인코딩 매개 변수를 추정하는 것은 추후 연구에서 진행할 예정이며, 현 단계에서는 인코더 컴포넌트에서 SNN 모듈에 인코딩 매개 변수를 전달하고, SNN 모듈은 다시 디코더에게 재전달 한다고 가정하여 동일한 인코딩 매개 변수를 사용하여 디코딩에 사용하였다.

#### 4. NA-IDE 환경에서 구현 결과 적용

NA-IDE 환경에서 3장에서 제안한 인코더 컴

포넌트를 이용하여 SNN 모델의 손 글씨 이미지 분류기를 구현하였다. 손 글씨 데이터 셋은 일반 딥러닝 기반의 이미지 분류기의 평가 지표로 널리 사용되는 MNIST 데이터를 사용하였다. MNIST 손 글씨 데이터 셋은 28x28 크기의 총 7만개의 이미지로 구성되어, 6만개는 학습으로 1만개는 평가용으로 사용된다. SNN은 아직 초기 연구 단계이기 때문에 적용 및 동작 가능 여부를 확인하기 위하여 MNIST 데이터를 선택하였다.

그림 14는 각 컴포넌트의 연결로 구현된 이미지 분류기의 전체 흐름도를 도식화한다. 그림 14에서 A 상자는 데이터 입력 컴포넌트로써 USB 카메라 컴포넌트로 MNIST 이미지 데이터셋을 촬영하여 입력 이미지를 생성한다. 그림 14의 D 창을 통하여 손 글씨 숫자 5가 정상적으로 입력되었음을 확인하였다. 수집된 이미지는 B 상자의 인코더 컴포넌트로 전달되며 스파이크 시계열 데이터로 변환된다. 인코더의 출력은 0 또는 1의 연속된 시계열로 표현되며, 변환된 스파이크 데이터는 C 상자의 SNN-MNIST 분류기로 전달된다.

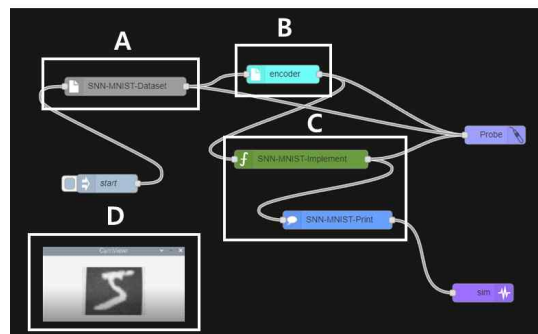


그림 14. NA-IDE 환경에서 구현된 SNN-MNIST 손 글씨 이미지 분류기

Fig. 14. Implemented SNN-MNIST hand written image classifier on NA-IDE

그림 14의 C 상자에 해당하는 SNN-MNIST 분류기에서는 스파이크 시계열 데이터를 입력받

아 0-9까지 10개의 숫자 중 가장 유사하게 분류된 숫자를 출력 컴포넌트를 통해 결과로 출력한다. 그림 15는 SNN-MNIST 손 글씨 이미지 분류기에서 출력된 동작 결과를 보인다.

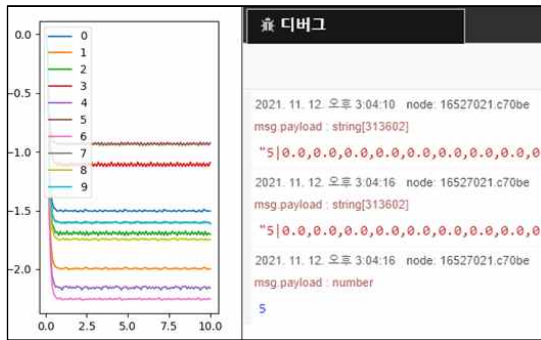


그림 15. SNN-MNIST 손 글씨 이미지 분류 결과  
Fig. 15. Result of SNN-MNIST hand written image classifier

그림 15에서 왼쪽 그래프는 결과 출력 컴포넌트를 이용하여 출력한 분류 결과로 0부터 9까지의 숫자에 대해 입력된 이미지와 동일한 정답일 확률이 시간에 따라 변화함을 보여준다. 그림 15의 오른쪽 1, 2번째 출력된 메시지는 입력받은 이미지의 데이터가 스파이크 데이터 형태로 전달되었음을 나타낸다. 최종 분류 결과 USB 카메라 컴포넌트로 입력한 이미지 숫자와 동일한 5를 가장 높은 확률로 예측함을 그림 15의 오른쪽 마지막 메시지에서 확인할 수 있다.

### 5. 결론

본 논문에서는 기존의 IoT와 뉴로모픽 구조를 지원하는 통합개발환경 NA-IDE에서 SNN 모델의 입력을 지원하기 위해 기존에 사용되는 데이터를 스파이크 시계열 데이터로 변환할 수 있도록 인코더와 디코더 컴포넌트를 구현하였다. 인

코더 컴포넌트는 다른 신경코딩 방식을 적용하여 다양한 SNN 모델에 적용할 수 있도록 이미지 데이터를 시계열 패턴 데이터로 변환하는 기능을 제공한다. 구현 결과 인코더 컴포넌트를 이용하여 이미지 데이터가 시간에 따라 형태가 다른 스파이크 시계열 데이터로 변환됨을 확인하였다. 디코더 컴포넌트는 SNN 모델의 결과로 전달되는 시계열 패턴 데이터를 기존의 컴포넌트에서 사용할 수 있도록 이미지로 복원하는 기능을 제공한다. 디코딩 과정에서 인코딩에 사용된 것과 동일한 매개변수에 따라 변환된 데이터와 원본 데이터의 유사성 차이가 있음을 확인하였다. 이는 디코딩 단계에서 이용하는 매개변수의 중요성을 알 수 있었다. 따라서 향후 연구로는 디코딩 과정에서 시계열 패턴 데이터로부터 인코딩 매개변수를 추정하는 연구와 이미지뿐만 아니라 음성과 같이 시변 정보를 포함하는 동적 데이터의 스파이크 시계열 데이터 변환에 관한 연구를 진행할 것이다. 본 논문에서 제안된 인코더와 디코더 컴포넌트를 이용한다면 NA-IDE에서 기존 IoT와 인공지능 기능뿐만 아니라 뉴로모픽 구조 기반의 다양한 SNN 모델을 지원할 수 있을 것으로 기대한다.

### 감사의 글

이 논문은 2019년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No. 2019-0-00708, 뉴로모픽 아키텍처 기반 자율형 IoT 응용통합개발환경).

### 참고 문헌

[1] Indiveri et al., “Neuromorphic silicon neuron circuits”, *Frontiers in Neuroscience*,

- vol. 5, no. 73, 2011, DOI: <https://doi.org/10.3389/fnins.2011.00073>
- [2] Davies, Mike, et al., “Loihi: A Neuromorphic Manycore Processor with on-chip Learning”, *IEEE Micro*, Vol. 38, No. 1, pp.82-99, 2018, DOI: <https://doi.org/10.1109/MM.2018.112130359>.
- [3] 최신현, “뇌과학과 뉴로모픽 기술동향”, 뉴로모픽 기술과 연구동향 2, 전기의 세계, 대한전기학회, 68(10), pp.28-31, 2019, <http://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE09216315>
- [4] Y. S. Yun, S. Kim, J. Park, H. Kim, J. Jung, S. Eun, “Development of Neuromorphic Architecture Integrated Development Environments”, *International Conference on Green and Human Information Technology (ICGHIT)*, pp.47-49, 2020, DOI: 10.1109/ICGHIT49656.2020.00019
- [5] 이성화, 김장우, “다양한 스파이크 기반 신경망 시뮬레이션을 위한 디지털 회로 구조”, 석사학위논문, 서울대학교, 서울, 2019, <https://hdl.handle.net/10371/150766>
- [6] Arbib, Michael A. “Brains, machines and buildings: towards a neuromorphic architecture”, *Intelligent Buildings International* 4.3, pp.147-168, 2012, DOI: <https://doi.org/10.1080/17508975.2012.702863>
- [7] Schliebs, Stefan, and Nikola Kasabov. “Evolving spiking neural network—a survey”, *Evolving Systems* 4.2, pp.87-98, 2013, DOI: <https://doi.org/10.1007/s12530-013-9074-9>
- [8] Spiking Neural Networks, the Next Generation of Machine Learning, towards data science, last modified Jan 11, 2018, accessed Nov 03, 2021, <https://towardsdatascience.com/spiking-neural-networks-the-next-generation-of-machine-learning-84e167f4eb2b>
- [9] Zador, Anthony. “Spikes: Exploring the neural code”, *Science* 277.5327, pp.772-773, 1997, ISBN:0-262-18174-6
- [10] Ponulak F, Kasinski A, “Introduction to spiking neural networks: Information processing, learning and applications”, *Acta Neurobiologiae Experimentalis*, 71(4): pp.409-433, 2011, PMID:22237491
- [11] 김용주, 김태호, “Spiking Neural Networks(SNN)구조에서 뉴런의 개수와 학습량에 따른 학습 성능 변화 분석”, *The Journal of the Convergence on Culture Technology (JCCT)*, Vol. 6, No. 3, pp.463-468, 2020, DOI: <https://doi.org/10.17703/JCCT.2020.6.3.463>
- [12] Ghosh-Dastidar, Samanwoy, and Hojjat Adeli, “Spiking neural networks”, *International journal of neural systems* 19.04, pp.295-308, 2009, DOI: <https://doi.org/10.1142/S0129065709002002>
- [13] Adrian ED, Zotterman Y, “The impulses produced by sensory nerve endings: Part II: The response of a single end organ”, *J Physiol.* 61 (2): pp.151 - 171, 1926, DOI: <https://doi.org/10.1113/jphysiol.1926.sp002281>
- [14] Gerstner, W., & Kistler, W. M, “Spiking neuron models : single neurons, populations, plasticity”, Kistler, Werner M., 1969-. Cambridge, U.K.: Cambridge University Press, 2002, DOI: <https://doi.org/10.1017/CBO9780511815706>
- [15] Borst, Alexander, and Frédéric E. Theunissen, “Information theory and neural coding”, *Nature neuroscience* 2.11, pp.947-957, 1999, DOI: <https://doi.org/10.1038/14731>
- [16] Guo, Wenzhe & Fouda, Mohammed E. & Eltawil, Ahmed & Salama, Khaled, “Neural Coding in Spiking Neural Networks: A Comparative Study for Robust Neuromorphic Systems”, *Frontiers in Neuroscience*, 15:638474, 2021, DOI: <https://doi.org/10.3389/fnins.2021.638474>.
- [17] Stein RB, Gossen ER, Jones KE, “Neuronal variability: noise or part of the signal?”, *Nat. Rev. Neurosci.* 6 (5): pp.389 - 397, DOI: <https://doi.org/10.1038/nrn1888>

- [18] Averbeck, Bruno B., Peter E. Latham, and Alexandre Pouget, "Neural correlations, population coding and computation", Nature reviews neuroscience 7.5, pp.358-366, 2006, DOI: <https://doi.org/10.1038/nrn1888>
- [19] Pan, Zihan et al, "Neural Population Coding for Effective Temporal Classification", 2019 International Joint Conference on Neural Networks (IJCNN), pp.1-8, 2019, DOI: <https://doi.org/10.1109/IJCNN.2019.8851858>

저 자 소 개



김회남(Hoinam Kim)

2019.8 한남대학교 정보통신공학과 졸업  
2019.9-현재 : 한남대학교 정보통신공학과  
석사과정  
<주관심분야> 인공지능



윤영선(Young-Sun Yun)

1990.2 KAIST 전산학과 졸업  
1992.2 KAIST 전산학과 석사  
2001.2 KAIST 전산학과 박사  
2006.4-2007.2 한국전자통신연구원 초빙  
연구원  
2012.8-2013.7 University of Washington  
방문학자  
2001.3-현재 : 한남대학교 교수  
<주관심분야> 음성인식, 음성변환, 화자인  
식, 인공지능, 내장형시스템, 저작권침해,  
유사도, 완성도 감정, 오픈소스 등