1. 연구 개요 및 목표

1.1 전체 목표

이 연구의 목표는 강화학습(Reinforcement Learning, RL) 에이전트를 이용해 스파이킹 신경망(Spiking Neural Network, SNN)의 **시냅스 가중치 업데이트 규칙(plasticity rule)** 을 학습시키는 것이다.

기존 STDP 식처럼 사람이 고정한 수식을 그대로 사용하는 대신 각 시냅스를 **로컬 상태만 관측하는 RL 에이전트**로 보고 이 에이전트가 **가중치 정책(weight policy)** 을 통해

- 로컬 상태(local state)를 기반으로 **가중치 변화량(action)** 을 제안하고
- 전역 과제 성능 혹은 BPTT 기울기 등 **전역 신호를 보상(reward)** 으로 받아 학습

하도록 설계한다.

정책이 내놓는 액션은 **새 가중치 자체가 아니라 현재 가중치에 더해질 가중치 변화량**으로 해석한다. 한 이벤트에서 정책 출력이 Δd 라면 실제 시냅스 가중치 업데이트는

$$\Delta w = \eta \cdot \Delta d$$

$$w \leftarrow w + \Delta w$$

형태가 되며 Δd 는 정규화된 가중치 변화량이다.

본 보고서는 이후 구현 보고서와 코드 구조의 기준이 되는 **고정된 실험 설계(구조와 타임스텝과 네트워크 아키텍처와 RL 정의)** 를 명시한다.

1.2 실험 개요

본 연구는 네 가지 서로 다른 설정에서 **가중치 정책이 어떤 규칙을 학습하는지 그리고 구조적 창발이나 전역 기울기 근사가 실제로 일어나는지 혹은 그렇지 않다면 어떤 방식으로 학습이 진행되는지**를 관찰하는 데 목적이 있다.

1. 완전 비지도 단일 가중치 정책 실험
2. 완전 비지도 두 가중치 정책 실험(Diehl–Cook 구조에서 입력과 억제를 구분)
3. 완전지도 전역 기울기 모방 실험(gradient mimicry)
4. 준지도 단일 가중치 정책 분류 실험

각 실험에서 에피소드 단위의 RL 궤적을 수집하고 동일한 가중치 정책 아키텍처와 가치함수 아키텍처를 사용해 학습을 진행한다.

---

2. 공통 구성 요소

2.1 데이터셋과 입력 인코딩

데이터셋은 MNIST 이며 28×28 회색조 이미지와 10 개 클래스 라벨을 사용한다. 각 픽셀 값은 $[0,1]$ 구간으로 정규화한다.

이미지는 푸아송 인코더(Poisson encoder)를 통해 스파이크 열로 변환된다. 정규화된 픽셀 값 $p \in [0,1]$ 에 대해 시뮬레이션 길이 T 동안 각 타임스텝마다 해당 픽셀에서 스파이크가 발생할 확률을 p 에 비례하도록 두고 독립적인 푸아송 샘플링을 수행한다.

완전 비지도와 완전 비지도 두 정책 실험에서는 한 입력당 T_{unsup} 타임스텝 동안 스파이크를 주입하고 완전지도와 준지도 실험에서는 T_{sup} 와 T_{semi} 타임스텝을 사용한다. 문서에서는 기본값으로 예를 들어

- $T_{\text{unsup}} = 100$
- $T_{\text{sup}} = 16$
- $T_{\text{semi}} = 16$

을 사용한다고 가정하지만 실제 구현에서는 이 값들을 모두 **커맨드라인 하이퍼파라미터(CLI 인자)** 로 노출하며 실험마다 자유롭게 조정할 수 있다.

2.2 뉴런 모델(LIF)

모든 스파이킹 층은 LIF(Leaky Integrate and Fire) 뉴런을 사용한다. 연속 시간 막전위 V 에 대해

$$\tau_m \frac{\mathrm{d}V}{\mathrm{d}t} = -(V - V_{\text{rest}}) + RI_{\text{in}}$$

을 사용한다. 이산 시간 구현에서는 시뮬레이션 타임스텝 크기를 Δt 로 두고

$$V(t+1) = V(t) + \frac{\Delta t}{\tau_m} \cdot \big(-(V(t) - V_{\text{rest}}) + RI_{\text{in}}(t)\big)$$

을 사용한다. $V(t+1)$ 이 임계치 V_{th} 를 넘으면 스파이크를 발생시키고 막전위를 V_{reset} 으로 리셋한다.

τ_m 와 V_{rest} 와 V_{th} 와 V_{reset} 와 R 와 Δt 는 실험마다 재조정해야 하는 중요한 상수이므로 **모두 CLI 하이퍼파라미터로 노출**한다. 문서에서 제시하는 값은 기본값의 예시일 뿐이며 실제 실험에서는 커맨드라인 인자를 통해 결정된다.

2.3 스파이크 히스토리 배열과 1D CNN 입력 길이

각 시냅스는 pre 뉴런과 post 뉴런의 최근 스파이크 히스토리를 길이 K 의 배열로 유지하고 이를 1D CNN 전단의 입력으로 사용한다.

시뮬레이션 타임스텝 수 T 와 CNN 에서 사용하는 히스토리 길이 K 는 서로 다른 값일 수 있으나 항상

$$K \le T$$

를 만족하도록 설정한다. 완전 비지도 실험에서는 기본적으로

- $T_{\text{unsup}} = 100$
- $K_{\text{unsup}} = 100$

을 사용하고 완전지도와 준지도 실험에서는

- $T_{\text{sup}} = 16$
- $K_{\text{sup}} = 16$
- $T_{\text{semi}} = 16$
- $K_{\text{semi}} = 16$

을 기본값으로 사용한다. 그러나 이 값들은 모두 **CLI 하이퍼파라미터** 로 두어 필요하면 K 만 짧게 두거나 T 를 늘리는 식의 변형을 쉽게 실험할 수 있도록 한다.

각 시냅스 i 의 스파이크 히스토리 배열은

- 완전 비지도 계열 실험에서 $X_i(t)^{\text{unsup}} \in \mathbb{R}^{2K_{\text{unsup}}}$

- 완전지도 실험에서 $X_i(t)^{\text{sup}} \in \mathbb{R}^{2K_{\text{sup}}}$
- 준지도 실험에서 $X_i(t)^{\text{semi}} \in \mathbb{R}^{2K_{\text{semi}}}$

형태로 정의한다. 두 채널은 pre 뉴런과 post 뉴런의 스파이크 시퀀스를 의미한다.

2.4 1D CNN 전단 구조

스파이크 시간 패턴만을 처리하기 위해 공통 1D CNN 전단을 사용한다. 입력은 항상 채널 차원 2 와 시간 축 길이 K 를 갖는 배열이다.

CNN 전단은 다음과 같이 고정한다.

1. 1차 합성곱
   입력 채널 2 와 출력 채널 16 과 커널 크기 5 와 padding 2 와 stride 1 을 사용하고 활성함수는 ReLU 를 사용한다.
2. 2차 합성곱
   입력 채널 16 과 출력 채널 32 와 커널 크기 5 와 padding 2 와 stride 1 을 사용하고 활성함수는 ReLU 를 사용한다.
3. 시간 축에 대한 global average pooling
   최종 출력 feature 벡터의 차원은 항상 32 이다.

이 CNN 전단은 **모든 가중치 정책과 가치함수가 공유**하며 이후 MLP head 에서만 정책별로 분기한다.

2.5 가중치 정책과 가치함수 입력 구성

CNN 에서 얻은 feature 벡터와 스칼라 정보를 concat 하는 late fusion 구조를 사용한다.

한 시냅스 i 와 시간 t 에 대해

- CNN feature $h_i(t) \in \mathbb{R}^{32}$
- 현재 시냅스 가중치 $w_i(t)$

를 항상 포함한다. 완전지도 실험에서는 여기에 해당 시냅스가 속한 레이어 인덱스를 정규화한 스칼라 $l_{\text{norm}i}$ 를 추가한다.

따라서 입력 벡터는

- 완전 비지도 계열과 준지도에서
  $z_i(t)^{\text{unsup}} = [h_i(t); w_i(t)] \in \mathbb{R}^{33}$
  $z_i(t)^{\text{semi}} = [h_i(t); w_i(t)] \in \mathbb{R}^{33}$
- 완전지도 실험에서
  $z_i(t)^{\text{sup}} = [h_i(t); w_i(t); l_{\text{norm}i}] \in \mathbb{R}^{34}$

로 정의한다. 여기서 $[\cdot;\cdot]$ 는 벡터 연결 연산을 의미한다.

2.6 가중치 정책 네트워크(Actor, Gaussian)

가중치 정책 네트워크는 입력 $z_i(t)$ 를 받아 **연속 스칼라 액션** $\Delta d_i(t)$ 를 출력한다. 이 액션은 정규화된 가중치 변화량이며 실제 업데이트는 학습률로 다시 스케일링된다.

MLP 구조는 모든 실험에서 공통으로

1. 입력층
   입력 차원 33 또는 34 와 hidden 크기 128 과 ReLU 활성함수를 사용한다.

2. 은닉층
   hidden 크기 128 과 ReLU 활성함수를 사용한다.
3. 출력층
   차원 1 의 선형 출력을 얻은 뒤 Tanh 를 적용하여 평균 $m_i(t) \in [-1, 1]$ 을 만든다.

이 평균 $m_i(t)$ 를 사용해 **가우시안 정책(Gaussian policy)** 을 정의한다. 정책 표준편차 σ_{policy} 는 모든 시냅스와 모든 실험에서 공유되는 스칼라 값으로 두며

$$\Delta d_i(t) \sim \mathcal{N}\big(m_i(t),,,\sigma^2_{\text{policy}}\big)$$

으로 액션을 샘플링한다. σ_{policy} 는 탐색 강도와 직접 연결되는 매우 중요한 상수이므로 **반드시 CLI 하이퍼파라미터로 노출**한다. 문서에서는 예를 들어 $\sigma_{\text{policy}} = 0.1$ 을 기본값으로 가정할 수 있다.

샘플링된 $\Delta d_i(t)$ 는 필요하면 $[-1, 1]$ 범위로 클리핑한다. 실제 가중치 업데이트는

$$\Delta w_i(t) = \eta_{\text{group}(i)} \cdot \Delta d_i(t)$$

$$w_i(t+1) = \text{clip}\big(w_i(t) + \Delta w_i(t)\ ;\ w_{\min(\text{group}(i))}\ ,\ w_{\max(\text{group}(i))}\big)$$

으로 정의한다. 여기서 $\eta_{\text{group}(i)}$ 는 시냅스 타입별 학습률이다. η_{group} 들 역시 모두 **CLI 하이퍼파라미터** 로 두며 문서의 값은 기본값 예시로만 사용한다.

이 연구에서 사용하는 RL 방법은 **연속 액션 Gaussian 정책을 갖는 on policy Advantage Actor Critic(A2C 계열)** 이며 Soft Actor Critic 과는 다르다. 즉 단일 스칼라 가치함수 V 를 baseline 으로 사용하는 표준 Actor Critic 구조를 따른다.

가중치 정책은 시냅스마다 따로 두지 않고 **정해진 소수 개의 정책 모듈을 두고 모든 시냅스와 뉴런이 이를 공유**한다. 예를 들어 완전 비지도 단일 정책 실험에서는 하나의 가중치 정책 π_{single} 만 두고 네트워크 내 모든 학습 시냅스가 이 정책을 공유한다. 완전 비지도 두 정책 실험에서는 입력층과 억제층을 담당하는 두 정책 π_{exc} 와 π_{inh} 를 두고 동일 타입 시냅스 간에는 항상 정책을 공유한다.

또한 모든 실험에서 시냅스 이벤트는 **pre 스파이크 이벤트와 post 스파이크 이벤트로 나누어 처리**하며 코드 수준에서는 각 가중치 정책이 pre 이벤트와 post 이벤트에 대해 서로 다른 출력을 내도록 구현할 수 있다. 따라서 어떤 뉴런의 입력 시냅스이든 간에

- pre 스파이크 기반 업데이트는 pre 가중치 정책에 의해
- post 스파이크 기반 업데이트는 post 가중치 정책에 의해

결정된다고 가정한다. 이때도 정책의 개수는 미리 정해진 소수 개이며 모든 시냅스가 같은 pre 와 post 가중치 정책을 공유한다.

2.7 가치함수 네트워크(Critic)

가치함수 네트워크는 동일한 입력 $z_i(t)$ 를 받아 해당 상태에서의 **기대 누적 보상 값** $V_\phi(z_i(t))$ 를 추정한다. 구조는 가중치 정책과 유사하되 출력층 활성함수는 사용하지 않는다.

1. 입력층
   입력 차원 33 또는 34 와 hidden 크기 128 과 ReLU 활성함수를 사용한다.
2. 은닉층
   hidden 크기 128 과 ReLU 활성함수를 사용한다.
3. 출력층
   차원 1 의 선형 출력을 사용한다.

정책 네트워크와 동일한 수준의 용량을 갖도록 hidden 크기를 128 로 고정하고 CNN 전단은 정책과 Critic 이 항상 공유한다. Critic 은 하나만 두고 네 가지 모든 실험에서 공통으로 사용하는 것을 기본 설정으로 한다. 즉 **보상 정의는 실험마다 다르지만 항상 단일 스칼라 return 을 최적화하는 단일 목적 문제(single objective)** 로 보고 하나의 V_ϕ 를 baseline 으로 공유한다.

2.8 RL 궤적 구조와 에피소드 정의

강화학습 표기와 에피소드 정의를 모든 실험에서 공통으로 사용한다.

- 시뮬레이션 시간 스텝은 t
- 시냅스 인덱스는 i
- 에피소드 내 이벤트 인덱스는 e 로 둔다.

각 시냅스는 pre 뉴런과 post 뉴런 중 하나라도 스파이크를 발생시키는 타임스텝마다 **이벤트** 를 생성한다. 이벤트 e 은 (i,t) 쌍에 대응한다.

각 이벤트에 대해

- 상태(state) s_e 는 $z_i(t)$
- 행동(action) a_e 는 $\Delta d_i(t)$
- 보상(reward) r_e 는 실험별 보상 정의에 따라 결정된다.

Critic 출력은

$$V_e = V_\phi(s_e)$$

로 표기한다.

에피소드 하나는 항상 **하나의 입력 이미지에 대한 전체 스파이크 시뮬레이션과 그 과정에서 발생한 모든 시냅스 이벤트의 집합**으로 정의한다. 즉

- MNIST 이미지 하나를 선택하고 푸아송 인코딩을 통해 길이 T 의 스파이크 열을 생성한 뒤
- $t=1$ 부터 $t=T$ 까지 SNN 을 시뮬레이션하면서 발생한 모든 이벤트를 모으고
- 에피소드가 끝난 시점에서 전역 보상 혹은 이벤트별 보상을 계산해 이 에피소드의 RL 업데이트를 수행한다.

누적 보상(return)은 이벤트 인덱스 e 에 대해

$$G_e = \sum_{k \geq e} \gamma^{k-e} r_k$$

로 정의한다. 완전 비지도와 준지도 실험에서는 에피소드 종료 후 전역 보상 R 을 계산한 뒤 모든 이벤트에 대해 $r_e = R$ 로 두고 discount factor 를 $\gamma_{\text{unsup}} = 1$ 과 $\gamma_{\text{semi}} = 1$ 로 두어 항상 $G_e = R$ 이 되도록 단순화한다. 완전지도 gradient mimicry 실험에서는 보상이 이미 시간과 시냅스별로 충분히 local 하므로 $\gamma_{\text{sup}} = 1$ 로 두고 $G_e = r_e$ 로 본다.

Advantage 는

$$A_e = G_e - V_e$$

로 정의한다.

각 에피소드마다 RL 버퍼에는 모든 이벤트에 대한 (s_e, a_e, V_e, r_e) 가 저장된다.

2.9 Actor Critic 업데이트 수식

가중치 정책은 Gaussian 정책이므로 로그 확률은

$$\log \pi_\theta(a_e \mid s_e) = -\frac{(a_e - m_e)^2}{2\sigma_{\text{policy}}^2} + C$$

형태가 된다. 여기서 m_e 는 해당 상태에서 정책 네트워크가 출력한 평균이고 C 는 θ 와 무관한 상수이다.

에피소드 단위 Actor 손실은

$$L_{\text{actor}} = -\mathbb{E}_e \left[A_e \log \pi_\theta(a_e \mid s_e) \right]$$

로 정의한다. Critic 손실은

$$L_{\text{critic}} = \mathbb{E}_e \left[\left(G_e - V_\phi(s_e) \right)^2 \right]$$

이다.

엔트로피 정규화를 포함한 전체 손실은

$$L_{\text{RL}} = L_{\text{actor}} + \beta_v L_{\text{critic}} - \beta_{\text{ent}} H(\pi)$$

로 정의한다. β_v 와 β_{ent} 는 각각 value 손실과 정책 엔트로피의 가중치이다. 두 값 역시 실험마다 바꾸어 볼 수 있는 하이퍼파라미터이므로 CLI 인자로 노출한다.

파라미터 업데이트는

$$\theta \leftarrow \theta - \alpha_{\text{actor}} \frac{\partial L_{\text{RL}}}{\partial \theta} \quad \phi \leftarrow \phi - \alpha_{\text{critic}} \frac{\partial L_{\text{RL}}}{\partial \phi}$$

로 수행한다. 학습률 α_{actor} 와 α_{critic} 도 모두 CLI 하이퍼파라미터로 지정한다. 최적화 알고리즘은 기본적으로 Adam 을 사용한다.

2.10 공통 CLI 하이퍼파라미터 정리

실제 구현에서는 다음과 같은 값들을 모두 커맨드라인 인자로 노출한다.

- 시뮬레이션 타임스텝 수 T_{unsup} 와 T_{sup} 와 T_{semi}
- 스파이크 히스토리 길이 K_{unsup} 와 K_{sup} 와 K_{semi}
- LIF 뉴런 상수 τ_m 와 V_{rest} 와 V_{th} 와 V_{reset} 와 R 와 Δt
- Gaussian 정책의 표준편차 σ_{policy}
- 시냅스 타입별 학습률 η_{group}
- Actor 와 Critic 학습률 α_{actor} 와 α_{critic}
- discount factor γ 와 value 손실 가중치 β_v 와 엔트로피 가중치 β_{ent}

문서에서 특정 숫자를 제시하는 경우 그 값은 **기본값의 예시** 이며 실제 실험에서는 모두 CLI 레벨에서 조정 가능한 하이퍼파라미터로 취급한다.

---

3. 실험 1: 완전 비지도 단일 가중치 정책

3.1 목표

첫 번째 실험에서는 Diehl Cook 스타일의 흥분 억제 구조 위에서 **단일 가중치 정책을 모든 학습 시냅스가 공유**하도록 두었을 때

- 입력 패턴에 따른 스파스한 활성과 뉴런별 역할이 자연스럽게 형성되는지
- 단일 정책만으로도 안정적인 흥분 억제 균형과 winner 패턴이 나타나는지

를 관찰한다. 이 시점에서는 정책을 나누지 않고 하나의 공통 가중치 정책이 모든 시냅스 업데이트를 담당한다.

3.2 SNN 구조(Diehl Cook 아키텍처)

완전 비지도 계열 실험에서 사용하는 SNN 구조는 Diehl Cook 2015 의 모델 구조를 따른다.

- 입력층은 MNIST 이미지의 784 개 픽셀을 784 개 입력 스파이크 소스로 매핑한다.
- 흥분성 LIF 층 E 는 뉴런 수 $N_E = 400$ 을 갖는다.
- 억제성 LIF 층 I 는 뉴런 수 $N_I = 400$ 을 갖는다.

연결 구조는 다음과 같다.

- 입력에서 흥분층으로 가는 Input to E 시냅스는 784×400 fully connected 구조이며 학습 대상이다. 가중치는 양수 영역 $[0, w_{\max}^{\text{exc}}]$ 로 클리핑한다.
- 흥분층에서 억제층으로 가는 E to I 시냅스는 각 흥분 뉴런과 대응하는 억제 뉴런 사이의 1 대 1 연결을 사용하고 고정 양수 가중치 w_{EI} 를 사용한다.
- 억제층에서 흥분층으로 가는 I to E 시냅스는 all to all 구조를 사용한다. 자기 자신으로의 연결은 0 으로 두는 것을 기본으로 한다. 가중치는 음수 영역 $[w_{\min}^{\text{inh}}, 0]$ 에서 클리핑하며 학습 대상이다.

완전 비지도 단일 정책 실험에서는 Input to E 와 I to E 시냅스를 모두 **단일 가중치 정책 π_{single} 이 공유**하여 업데이트한다.

3.3 순전파와 에피소드 구성

에피소드 하나는 입력 이미지 하나에 대응한다. 순전파는 다음 단계로 진행된다.

1. 입력 이미지 x 를 $[0,1]$ 로 정규화하고 푸아송 인코더를 초기화한다.
2. 모든 LIF 뉴런의 막전위와 스파이크 히스토리를 초기화한다.
3. $t = 1$ 부터 $t = T_{\text{unsup}}$ 까지
    1. 입력층에서 푸아송 샘플링으로 스파이크 벡터 $s_{\text{in}}(t)$ 를 생성한다.
    2. Input to E 와 E to I 와 I to E 가중치를 이용해 각 뉴런의 입력 전류를 계산한다.
    3. E 층과 I 층의 막전위를 LIF 식으로 업데이트하고 임계치를 넘는 뉴런에서 스파이크를 발생시킨다.
    4. 각 시냅스에 대해 pre 와 post 스파이크를 스파이크 히스토리 배열 $X_i^{\text{unsup}}(t)$ 에 기록한다.
4. T_{unsup} 스텝이 끝나면 흥분층 뉴런의 발화 패턴을 집계해 winner 패턴 등을 계산한다.

이 전체가 하나의 에피소드이며 에피소드 종료 후 전역 비지도 보상을 계산한다.

3.4 로컬 상태와 가중치 정책 적용

각 시냅스 i 에 대해 스파이크 히스토리 배열 $X_i^{\text{unsup}}(t)$ 를 CNN 전단에 통과시켜 feature $h_i(t)$ 를 얻고 현재 가중치 $w_i(t)$ 를 concat 하여

$$z_i^{\text{unsup}}(t) = [h_i(t); w_i(t)]$$

를 만든다. Input to E 와 I to E 를 포함한 모든 학습 시냅스는 공통 가중치 정책 π_{single} 을 사용한다.

이벤트는 pre 뉴런이나 post 뉴런 중 하나라도 스파이크를 낸 경우에만 생성한다. 이벤트가 발생하면

1. $z_i^{\text{unsup}}(t)$ 를 π_{single} 에 입력해 평균 m_e 를 계산한다.
2. 가우시안 정책에서 Δd_e 를 샘플링한다.
3. 시냅스 타입에 따라 대응되는 학습률 $\eta_{\text{group}(i)}$ 를 곱해 실제 변화량 Δw_e 를 계산한다.
4. 가중치 w_i 를 즉시 업데이트하고 클리핑한다.
5. Critic 에서 $V_e = V_\phi(z_i^{\text{unsup}}(t))$ 를 계산한다.
6. (s_e, a_e, V_e) 를 에피소드 버퍼에 저장한다.

pre 이벤트와 post 이벤트는 모두 같은 π_{single} 을 사용하지만 코드 수준에서는 pre 와 post 이벤트 타입을 구분해 상태에 포함시킬 수 있다.

3.5 전역 비지도 보상

에피소드가 끝난 뒤 Diehl Cook 스타일의 발화 통계를 사용해 전역 비지도 보상 R 을 계산한다. R 은

- 스파스 활성 정도
- 뉴런 간 역할 다양성
- 같은 입력에 대한 응답 안정성

을 반영하는 여러 항의 가중합으로 정의한다. 예를 들어

- 너무 많은 뉴런이 동시에 발화하면 보상을 낮추고
- 데이터셋 전체에 걸쳐 다양한 뉴런이 winner 가 되면 보상을 높이고
- 동일 이미지를 반복 제시했을 때 winner 패턴이 안정적이면 보상을 높인다.

구체적인 수식과 정규화 방식은 구현 시점에 고정한다. 완전 비지도 계열에서는 시간별 보상 대신 이 전역 보상만을 사용한다.

모든 이벤트에 대해

$$r_e = R$$

로 두고 $\gamma_{\text{unsup}} = 1$ 로 두어 항상

$$G_e = R$$

이 되도록 한다.

3.6 RL 궤적과 학습

에피소드가 끝난 뒤 에피소드 버퍼에 저장된 (s_e, a_e, V_e, r_e) 전체에 대해 2.9 의 Actor Critic 공식을 사용해 L_{actor} 와 L_{critic} 를 계산하고 파라미터를 업데이트한다.

이 실험은 단일 가중치 정책이 공유되는 상황에서 **자연스럽게 어떤 STDP 유사 커널과 sign 구조가 형성되는지** 를 관찰하기 위한 기준점 역할을 한다.

---

4. 실험 2: 완전 비지도 두 가중치 정책

4.1 목표

두 번째 실험에서는 실험 1 과 동일한 Diehl Cook 구조와 전역 비지도 보상을 유지하되 학습 시냅스를

- `Input to E`
- `I to E`

두 그룹으로 나누고 각 그룹에 서로 다른 가중치 정책 π_{exc} 와 π_{inh} 를 부여한다. 이를 통해

- 두 가중치 정책이 서로 다른 기능적 역할로 분화하는지
- 단일 정책 실험과 비교해 E I sign 구조와 스파스 패턴의 질이 어떻게 달라지는지

를 관찰한다.

4.2 구조와 로컬 상태

SNN 구조와 LIF 파라미터와 스파이크 히스토리 구조와 $z_i^{\text{unsup}}(t)$ 정의는 실험 1 과 동일하다. 차이는

- `Input to E` 시냅스는 π_{exc} 를 사용하고
- `I to E` 시냅스는 π_{inh} 를 사용한다.

는 점이다. 두 가중치 정책은 CNN 전단을 공유하고 마지막 FC head 만 다르다.

안정성을 위해 두 정책의 액션 범위를 스케일링해 사용할 수 있다. 예를 들어

$$\Delta w_e = \eta_{\text{group}(i)} \cdot \alpha_{\text{unsup}} \cdot \Delta d_e$$

와 같이 작은 상수 α_{unsup} 를 곱해 업데이트의 크기를 줄인다. α_{unsup} 도 CLI 하이퍼파라미터로 노출한다.

4.3 순전파와 RL 절차

순전파와 에피소드 정의와 전역 보상 계산과 RL 궤적 수집 절차는 실험 1 과 동일하다. 차이는 이벤트에서 어떤 가중치 정책을 호출하는지뿐이다.

- Input to E 시냅스에서 이벤트가 발생하면 π_{exc} 를 사용한다.
- I to E 시냅스에서 이벤트가 발생하면 π_{inh} 를 사용한다.

에피소드 종료 후 모든 이벤트에 동일한 전역 보상 R 을 할당하고 Actor Critic 업데이트를 수행한다.

4.4 분석 항목

실험 1 과 실험 2 를 비교해

- π_{exc} 와 π_{inh} 가 학습한 STDP 유사 커널 $\Delta w(\Delta t)$ 의 차이
- 뉴런별 outgoing weight sign 분포
- winner 뉴런 분포와 스파스 활성도

를 정량적으로 비교한다. 이를 통해 **가중치 정책을 둘로 나누는 것 자체가 역할 창발과 구조 형성에 기여하는지** 를 확인한다.

---

5. 실험 3: 완전지도 전역 기울기 모방(gradient mimicry)

5.1 목표

세 번째 실험에서는 전역 BPTT 기울기를 Teacher 신호로 사용해 로컬 가중치 정책이 **전역 손실 기울기와 얼마나 잘 정렬될 수 있는지** 를 평가한다. 기준선은 surrogate gradient 와 BPTT 로 직접 학습한 SNN 이다.

관심 대상은

- 로컬 정책이 제안한 업데이트 Δw_{agent} 와 전역 기울기 g 사이의 정렬 정도
- 분류 정확도와 손실 수렴 속도에서 기준선 대비 성능 차이

이다.

5.2 SNN 구조

입력층은 MNIST 이미지의 784 개 픽셀을 입력 스파이크 소스로 사용하고 각 입력당 T_{sup} 타임스텝 동안 스파이크를 주입한다.

히든 LIF 층은

- 히든층 1 의 뉴런 수 256
- 히든층 2 의 뉴런 수 128
- 히든층 3 의 뉴런 수 64
- 히든층 4 의 뉴런 수 32

를 사용한다. 출력층은 10 개 LIF 뉴런으로 구성된다. 입력층을 제외한 모든 층 사이의 시냅스 가중치는 학습 대상이다.

각 시냅스에는 레이어 인덱스를 정규화한 $l_{\text{norm}i}$ 를 부여한다. 예를 들어 입력 다음 히든층을 0.2 로 출력층을 1.0 으로 두는 방식이다. 이 값 역시 CLI 하이퍼파라미터로 재설정할 수 있다.

5.3 로컬 상태와 가중치 정책

각 시냅스의 스파이크 히스토리 $X_i^{\text{sup}}(t)$ 를 CNN 에 통과시켜 feature $h_i(t)$ 를 얻고 현재 가중치와 정규화된 레이어 번호를 concat 해

$$z_i^{\text{sup}}(t) = [h_i(t); w_i(t); l_{\text{norm}i}]$$

를 만든다. 모든 학습 시냅스는 단일 가중치 정책 π_{grad} 를 공유한다.

이벤트가 발생하면 π_{grad} 에서 평균 m_e 와 액션 Δd_e 를 얻어 가중치를 즉시 업데이트하고 Critic 에서 V_e 를 계산한 뒤 (s_e, a_e, V_e) 를 버퍼에 저장한다.

5.4 순전파와 BPTT

에피소드 하나는 입력 이미지 하나에 대한 전체 시뮬레이션으로 정의한다.

1. 푸아송 인코딩을 통해 T_{sup} 스텝 동안 입력 스파이크를 생성한다.
2. $t = 1$ 부터 T_{sup} 까지 SNN 을 시뮬레이션하며 모든 막전위와 스파이크를 저장한다.
3. 에피소드가 끝나면 출력층 뉴런의 스파이크 수 s_k 를 집계하고 발화율 $r_k = s_k / T_{\text{sup}}$ 를 계산한다.
4. 정답 라벨 y 에 대해 예를 들어 softmax$\big(\alpha r_k\big)$ 를 사용해 확률을 만들고 표준 cross entropy 를 사용하여 손실 L_{sup} 를 정의한다.
5. surrogate gradient 를 사용해 시간에 걸친 BPTT 를 수행하고 각 시냅스와 시간에 대한 전역 기울기 $g_{i,t} = \partial L_{\text{sup}} / \partial w_i(t)$ 를 계산한다.

5.5 gradient 정렬 기반 보상

각 이벤트 $e = (i, t)$ 에서 가중치 정책이 제안한 실제 업데이트를

$$\Delta w_{i,t}^{\text{agent}} = \eta_{\text{group}(i)} \cdot \Delta d_i(t)$$

로 두고 전역 기울기 $g_{i,t}$ 와의 정렬을 보상으로 사용한다.

기본 보상은

$$r_{i,t}^{\text{grad}} = -g_{i,t} \cdot \Delta w_{i,t}^{\text{agent}}$$

이다. 즉 기울기와 같은 방향으로 업데이트하면 보상이 커진다. 너무 큰 업데이트를 억제하기 위해

$$r_{i,t}^{\text{total}} = r_{i,t}^{\text{grad}} - \lambda \big(\Delta w_{i,t}^{\text{agent}}\big)^2$$

를 사용한다. λ 는 작은 양수이며 CLI 하이퍼파라미터로 노출한다.

각 이벤트에 대해

$$r_e = r_{i,t}^{\text{total}}$$

로 두고 $\gamma_{\text{sup}} = 1$ 이므로

$$G_e = r_e$$

로 본다.

5.6 RL 궤적과 업데이트

gradient mimicry 실험에서 에피소드 처리 흐름은 다음과 같다.

1. 순전파 중 이벤트가 발생할 때마다 (s_e, a_e, V_e) 를 버퍼에 저장한다.
2. 에피소드 종료 후 BPTT 로 모든 $g_{i,t}$ 를 계산하고 각 이벤트에 대해 r_e 를 구성한다.
3. 모든 이벤트에 대해 G_e 와 A_e 를 계산한다.
4. 2.9 의 공통 공식을 사용해 Actor 와 Critic 을 업데이트한다.

최종 분석에서는

- Δw_{agent} 와 g 의 부호 일치 비율
- 두 벡터 사이의 코사인 유사도
- 기준선 BPTT 모델과의 정확도와 loss 곡선

을 비교한다.

---

6. 실험 4: 준지도 단일 가중치 정책 분류

6.1 목표

네 번째 실험에서는 전역 기울기를 Teacher 로 사용하지 않고 **정답 라벨만을 보상으로 사용하는 순수 RL 기반 분류 문제** 를 다룬다. 단일 가중치 정책이 모든 학습 시냅스를 공유하며 이 보상만을 가지고 얼마나 좋은 분류기를 학습할 수 있는지 평가한다.

6.2 SNN 구조

입력층은 MNIST 이미지의 784 개 픽셀을 입력 스파이크 소스로 사용한다. 한 입력당 T_{semi} 타임스텝 동안 스파이크를 주입한다.

히든 LIF 층은

- 히든층 1 의 뉴런 수 256
- 히든층 2 의 뉴런 수 128

을 사용한다. 출력층은 10 개 LIF 뉴런으로 구성하며 **뉴런 인덱스와 라벨을 사전에 고정된 방식으로 매핑**한다. 예를 들어 출력층의 k 번째 뉴런이 숫자 k 를 의미하도록 정의한다.

학습되는 가중치는

- Input to Hidden1
- Hidden1 to Hidden2
- Hidden2 to Output

의 모든 시냅스이며 이들은 모두 단일 가중치 정책 π_{semi} 를 공유한다.

6.3 순전파와 출력 해석

에피소드 하나는 (x, y) 한 쌍에 대응한다.

1. 입력 x 를 푸아송 인코딩해 T_{semi} 스텝 동안 SNN 에 주입한다.
2. $t = 1$ 부터 T_{semi} 까지 SNN 을 시뮬레이션하며 이벤트가 발생할 때마다 π_{semi} 와 Critic 을 호출해 (s_e, a_e, V_e) 를 버퍼에 저장한다.
3. 에피소드 종료 후 출력층 뉴런 k 의 스파이크 수 s_k 와 발화율 $r_k = s_k / T_{\text{semi}}$ 를 계산한다.
4. 예측 라벨은

$$\hat{y} = \arg\max_k r_k$$

로 정의한다. 즉 출력층 발화율이 가장 높은 뉴런의 인덱스를 예측으로 사용한다.

6.4 보상 설계

라벨 y 와 예측 $\hat{y}$ 를 사용해 단순하지만 신뢰도 정보를 반영하는 전역 보상을 정의한다.

기본 분류 보상은

- $\hat{y} = y$ 이면 $R_{\text{cls}} = 1$
- $\hat{y} \neq y$ 이면 $R_{\text{cls}} = -1$

로 둔다.

정답 뉴런의 발화율과 가장 많이 발화한 오답 뉴런의 발화율 차이를

$$\text{margin} = r_y - \max_{k \neq y} r_k$$

로 정의하고

$$R_{\text{margin}} = \alpha_{\text{margin}} \cdot \text{margin}$$

을 추가한다. α_{margin} 은 CLI 하이퍼파라미터이다.

출력층의 전체 스파이크 수를 줄이기 위해

$$R_{\text{sparse out}} = -\alpha_{\text{spike}} \cdot \frac{\sum_k s_k}{T_{\text{semi}} \cdot 10}$$

을 추가할 수 있다. α_{spike} 역시 CLI 하이퍼파라미터이다.

최종 전역 보상은

$$R = R_{\text{cls}} + R_{\text{margin}} + R_{\text{sparse out}}$$

로 정의한다. 에피소드에 포함된 모든 이벤트에 대해

$$r_e = R$$

로 두고 $\gamma_{\text{semi}} = 1$ 이므로 항상 $G_e = R$ 이다.

6.5 RL 궤적과 업데이트

준지도 실험에서도 에피소드 단위로 (s_e, a_e, V_e, r_e) 를 모은 뒤 2.9 의 Actor Critic 업데이트를 적용한다. 이때 π_{semi} 와 V_ϕ 는 완전지도 실험과는 독립적으로 초기화할 수도 있고 완전 비지도 실험에서 학습된 가중치 정책을 초기값으로 사용할 수도 있다.

---

7. 기대 기여 및 활용

네 가지 실험을 통해 다음과 같은 점들을 관찰하고자 한다.

- 완전 비지도 단일 가중치 정책 설정에서 전역 비지도 보상만으로도 어떤 STDP 유사 규칙과 E I sign 구조가 자연스럽게 형성되는지
- 같은 구조에서 Input to E 와 I to E 를 구분한 두 가중치 정책을 사용할 때 두 정책의 역할이 어떻게 분화되는지
- 완전지도 설정에서 로컬 가중치 정책이 전역 BPTT 기울기를 어느 정도까지 근사할 수 있는지 그리고 그 근사가 실제 성능과 얼마나 연결되는지
- 준지도 설정에서 전역 기울기를 사용하지 않고 단순한 라벨 기반 보상만으로도 어느 정도 수준의 분류 성능과 구조가 학습되는지

이 문서는 이후 구현 보고서와 코드 구현에서 참조할 고정 설계를 정리한 것이며 각 실험에서 **가중치 정책과 가치함수가 무엇을 입력으로 받고 어떤 보상과 RL 궤적을 기준으로 학습되는지** 를 명확히 하기 위한 기준선 역할을 한다.