LTPS(LTP Simulator)

생명과학

보고서 작성자: 2203 김석준
팀원: 2215 조혜준

Ⅰ. 프로그램 개발 동기 및 목적

LTP(Long-term potentiation, 장기강화)는 신경학, 특히 뇌 신경생리학에서 중요하게 다루어지는 신경현상으로, 신경세포의 지속적인 자극을 통해 두 신경세포 사이의 신호 전달이 장기적으로 강화되는 현상이다. 이러한 LTP 현상은 신경세포 사이 공간인 시냅스에서 이루어지는 일종의 '기억' 메커니즘으로, 지속적인 자극을 가해주면 이후로는 조금의 자극으로도 잘 반응하게 된다는 점에서 동물과 사람에까지 해당되는 다양한 종류의 학습 효과를 나타나게 하는 원리일 가능성이 높은 매우 중요한 현상이다.

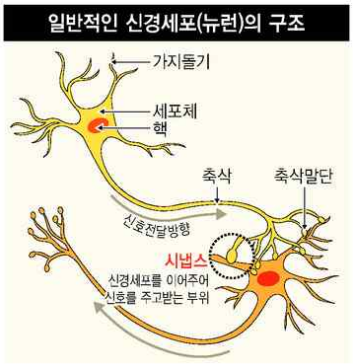

figure1. 신경세포와 시냅스의 이해

LTP는 종류에 따라 다양성이 높기 때문에 그 메커니즘이 아주 정확히 규명되지는 않았으나, 이것이 뇌에 있어 학습과 기억에 중대한 영향을 미치는 점은 간과할 수 없다. 그러나, 이러한 중요성에도 불구하고 일반 교양 서적에 짧게 소개되거나 접하기 어렵다는 점에서 일반 대중들과 생명과학 입문자들에게 그리 친숙한 개념은 아니라는 점에서 문제가 존재한다. 이에 본 연구자들은 LTP 현상을 가시화하여 사용자들로 하여금 이를 그래프로써 이해하고, 보다 향상된 이해력을 함양시키고자 본 프로그램 'LTPS'를 개발하게 되었다. 본 LTPS는 단순한 신경 전달 메커니즘까지 본떠 개발하였으며, 기타 세부사항을 사용자 스스로 제어할 수 있는 사용자 기반 커스터마이징(customizing) 기능도 포함하고 있어 보다 간단한 사실부터 차근차근 이해를 높이는 데에 도움이 될 것으로 예상된다. 본 프로그램을 통해 목표 사용자인 일반 대중 또는 생명과학 입문자에게 신경 전달의 원리와 LTP의 효과를 더 잘 이해시키고자 한다.

Ⅱ. 프로그램 소개

본 프로그램은 앞서 소개한 바와 같이 LTP 현상과 신경 전달 및 전도 과정을 시각화한 생명과학 연계 프로그램이다. 우선 회원가입을 통해 메인 창에 들어오면 프로그램에서 사용할 수 있는 기능들을 수행하게 된다.

먼저, 뉴런 커스터마이징(customizing) 기능으로 각 뉴런을 사용자 임의로 특징을 바꿀 수 있다. 이는 각 뉴런별로 존재하며, 뉴런의 특징을 사용자 임의로 바꿀 수 있어 사용자가 확인하고자 하는 현상을 맞춤별로 제공할 수 있다. 또한, 그래프 버튼을 누르면 matplotlib를 통한 그래프를 확인할 수 있으며, 이는 메인창과 실시간으로 연동되어 메인 창에서의 '자극'을 바로 감지해 그래프로 나타내어 뉴런의 전달과 전도 과정을 시각화한다. 사용자는 자신이 디자인한 뉴런들에 대해 서로 다른 빈도와 패턴으로 자극을 가하며 이를 확인할 수 있고, 생명과학적 소양을 크게 늘릴 수 있는 기회가 될 것으로 기대된다.

Ⅲ. 메뉴 구성도

1. 로그인/회원가입 창

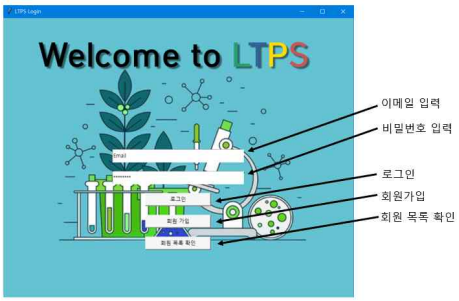

figure2. 로그인 창

figure3. 회원가입 창

2. 메인 창

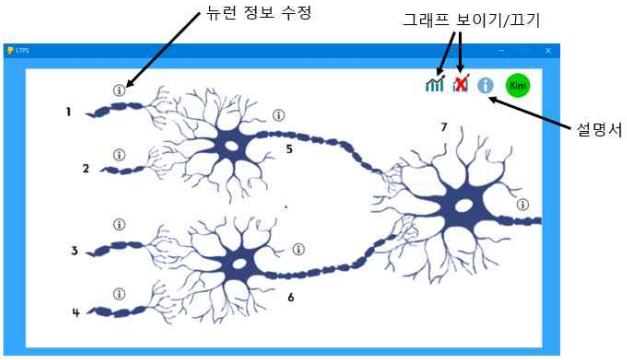

figure4. 메인 창

3. 기능 창

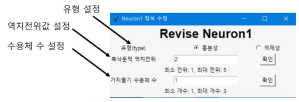

figure5. 뉴런 정보 수정 창

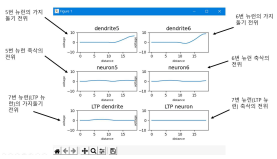

figure6. 그래프 창

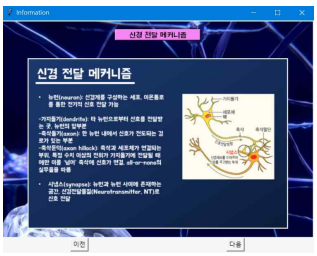

figure7. 설명서 창

본 프로그램에서는 다음과 같이 위계적으로 창들을 설정함으로 순서에 맞게 프로그램을 사용할 수 있도록 유도하였다. 먼저 로그인 및 회원가입 창을 통해 사용자를 확인한 후, 메인 창을 제시하여 여러 기능들을 수행할 수 있도록 함으로 자연스러운 프로그램 진행을 할 수 있도록 하였다.

Ⅳ. 프로그램 사용법 및 주요 실행 결과

앞서 소개한 바와 같이, 본 'LTPS' 프로그램은 사용자 기반 커스터마이징(customizing) 기능을 기반으로 사용자가 직접 조작함으로 신경에서 발생하는 전위를 그래프를 통해 확인한다. 우선, 본 프로그램을 실행하기 위해서는 로그인 창을 거쳐야 하며, 회원가입 정보는 database를 통해 저장된다. 회원가입에는 이름과 생년월일, 이메일 주소와 비밀번호를 입력하도록 하며, 입력된

정보는 database에 저장된다. 로그인을 거친 후에는 pygame 기반 메인 창이 뜨게 되며, 메인 창의 버튼들을 통해 기능들이 수행된다. 기능들에는 뉴런 정보 수정, 그래프, 설명서가 있다.

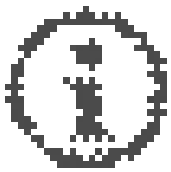

figure8. 정보 수정 기능 버튼

첫 번째 기능은 뉴런의 정보 수정 기능으로, 메인 창에 제시된 7개의 뉴런 각각을 사용자가 스스로 수정할 수 있도록 한 기능이다. 이는 이후 소개할 그래프 기능을 위한 보조기능으로, 사용자가 원하는 현상을 만들어내기 위한 자유도를 부여하였다. 각 뉴런별로 제시된 info 그림은 뉴런의 특징을 바꿀 수 있는 GUI 창을 띄운다. 각 뉴런별로 바꿀 수 있는 값은 유형(type), 측삭둔덕 역치전위, 가지돌기 수용체 수이며, 디폴트(default)값은 '흥분성', 2, 1이다. 특히, 축삭둔덕 역치전위와 가지돌기 수용체 수는 사용자가 직접 입력하도록 하였으며, 그 범위는 미리 하단에 공지하여 알맞은 범위로 입력하도록 유도하였다. 또한, 그럼에도 불구하고 잘못된 값을 입력하는 경우를 방지하여 확인 버튼을 눌러 적절한 값인지 확인하는 과정을 거치도록 하였다. 끝으로, 수정이 완료되면 바로 종료를 하면 되고, 수정이 되었을 경우 '변경하시겠습니까?'라는 문구가, 수정을 하지 않았을 경우 '변경사항 없이 끝내겠습니까?' 문구가 나오도록 설정하였다.

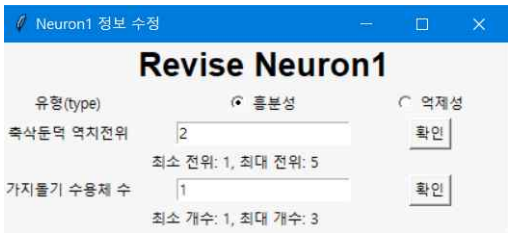

figure9. 뉴런 정보 수정 창 및 디폴트 값

두 번째 기능은 그래프 기능으로, 본 프로그램에서 핵심적인 기능이다. 본 프로그램의 이론인 신경 전달 이론과 LTP 현상에 대한 이해가 부족하면 뉴런의 정보 수정부터 그래프의 개형까지 이해가 어렵기 때문에 추후에 소개할 설명서 기능을 통해 충분히 학습한 후 프로그램을 진행할 수 있도록 한다. 그래프 창에 수록된 그래프는 총 6개로, 각각 5번 뉴런의 가지돌기, 6번 뉴런의 가지돌기, 5번 뉴런의 축삭, 6번 뉴런의 축삭, 7번 뉴런(LTP 뉴런)의 가지돌기, 7번 뉴런의 축삭에서의 전위 값을 실시간으로 표시한다. 가로축은 원점으로부터의 거리로, 모든 그래프에서 시간에 따라 그래프가 우측으로 이동함을 볼 수 있다. 또한, 메인 창에서 키보드로 1, 2, 3, 4를 누르면 뉴런 1, 2, 3, 4가 각각 자극되고, 이는 그래프를 통해 확인할 수 있다. 각각의 모든 자극은 사인(sine)형 펄스로 근사하였으며, 따라서 모든 자극은 파동의 성질인 중첩의 원리에 근거해 단순 합으로써 그래프에 표시된다.

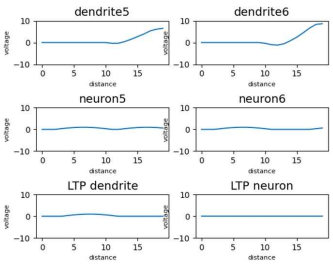

figure10. 그래프 창

본 그래프에 대한 상세한 설명은 다음과 같다. 뉴런 5와 6은 각각 뉴런 1, 2와 뉴런 3, 4 말단에 시냅스를 통해 연결되있다. 뉴런 정보 수정 기능을 통해 뉴런 1, 2, 3, 4의 유형을 흥분성과 억제성 중 하나로 바꿀 수 있으며, 이는 각각의 뉴런이 신경전달물질로 어떠한 종류의 물질을 채택하는지를 결정한다. 따라서, 이러한 변경은 그래프상에서 뉴런 5와 6의 가지돌기 전위가 음의 펄스가 나타나는 것으로 확인할 수 있다. 또한, 뉴런 5와 6의 정보중 축삭둔덕 역치전위값을 변경하면 뉴런 5와 6 각각의 축삭에 신호가 전달되는지의 여부가 변경되며, 이는 실무율(all-or-none)에 근거한다. 만약 뉴런 5 또는 6의 축삭둔덕 역치전위값을 내리면, 가지돌기에서의 전위가 기존에 비해 낮아도 충분히 축삭으로 신호가 연결됨을 확인할 수 있으며, 축삭둔덕 역치전위값을 올리면 가지돌기에서의 전위가 기존에 비해 더 높아야만 축삭으로 신호가 연결됨을 확인할 수 있다. 뉴런 5와 6과 7의 축삭에서는 축삭에서의 전기신호의 전도과정에 근거하기 때문에 항상 양의 펄스가 이동하게 된다. 뉴런 5와 6의 말단에서의 신경전달물질은 역시 각각의 정보 수정에서 유형을 통해 변경 가능하며, 축삭을 통해 연결된 신호가 7번 뉴런(LTP 뉴런)의 가지돌기로 이어지고, 이는 전과 동일하게 7번 뉴런의 축삭둔덕 역치전위를 넘을 때 축삭으로 펄스가 이어지는 것을 확인할 수 있다. 추가적으로, 뉴런에 지속적인 자극을 가했을 때,

각 뉴런이 축삭에서 전위 그래프를 나타낼 때, 펄스가 중첩되지 않고 서로 떨어져 이동하는 것을 관찰할 수 있는데, 이는 뉴런의 불응기(refractory period)를 표현한 것이다.

세 번째 기능은 설명서 기능으로, 본 프로그램의 기반 이론과 조작법을 소개한 창이다. 버튼을 눌러 여러 설명서들을 쉽게 확인할 수 있다.

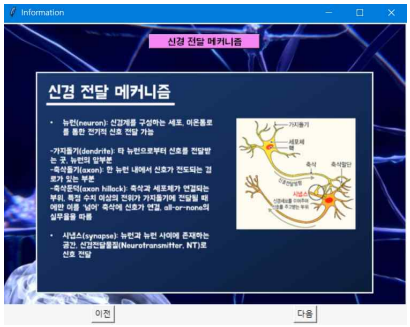

figure11. 설명서 창

Ⅴ. 프로그램에 적용한 아이디어

본 프로그램은 신경 전도 및 전달 이론과 LTP 현상을 융합시킨 것으로, 기존과 비슷한 아이디어 또는 프로그램이 없는 것으로 알고 있다. 본 프로그램은 신경 전달 및 전도 과정을 단순한 지식으로만 받아들이는 것이 아닌, 파이썬이 제공하는 다양한 모듈을 활용하여 시각화하고 체험할 수 있도록 함으로 생명과학의 공부에 있어 새로운 방향성을 제시하는 아이디어라고 생각한다. 이러한 시뮬레이션은 생명과학에 있어 부담이 덜한 학습을 제공할 뿐 아니라, 생물학적 실험에 있어서 이를 간편화할 수 있다는 점에서도 큰 의미가 있다. 기존 생물학적 실험은 안전교육에서 시작하여 각종 기기와 철저한 실험이 원칙이나, 사이버환경상에서의 실험을 통해 보다 안전하고 편리한 실험을 유도할 수 있다. 또한, 물론 본 프로그램의 주요 융합 학문은 생물이나, 물리학의 측면에서 사인(sine)형 파를 구현하고 이를 중첩의 원리를 적용하여 계산했다는 점에서 물리학과의 연계도 빼놓을 수 없으며, 간편한 확인 및 저장 시스템까지 갖추어져 있는 프로그램으로 아이디어를 구상하였다.

Ⅵ. 기술적 요소

1. 큐(Queue) 자료구조

본 프로그램에서는 큐(Queue) 자료구조를 적용하여 뉴런에서의 펄스형 전기 신호 전달을 구현하였다. 다음은 큐를 구현한 클래스이다.

```
class Queue:
    def __init__(self):
        self.rear = -1
        self.item = []
    def isEmpty(self):
        if self.rear == -1:
            return True
        else:
            return False
    def enqueue(self, newVal):
        self.item.insert(0, newVal)
        self.rear += 1
    def dequeue(self):
        if self.isEmpty():
            return -1
        else:
```

```
            self.rear -= 1
            return self.item.pop()
```

또한, 다음은 큐를 사용한 Neuron 클래스이다.

```
class Neuron:
    def __init__(self, number, type, sensitivity, receptors):
        self.number = number
        self.type = type
        self.axonHillock_sensitivity = sensitivity
        self.receptors = receptors
        self.LTP_receptors = 1
        self.dendrite_item = Queue()
        self.neuron_item = Queue()
        for i in range(totalTime):
            self.dendrite_item.enqueue(0)
            self.neuron_item.enqueue(0)
    def dendrite_stimulate(self, neuron):
        dendrite_stimulated_voltage = np.sin(np.array([v*np.pi/timeRange
for v in range(timeRange)])).tolist()
        if neuron.type == "excitatory":
            for i in range(timeRange):
                self.dendrite_item.item[i] +=
self.receptors*self.LTP_receptors*dendrite_stimulated_voltage[i]
        elif neuron.type == "inhibitory":
            for i in range(timeRange):
                self.dendrite_item.item[i] -=
self.receptors*self.LTP_receptors*dendrite_stimulated_voltage[i]
    def neuron_stimulate(self):
        neuron_stimulated_voltage = np.sin(np.array([v*np.pi/timeRange
for v in range(timeRange)])).tolist()
        for i in range(timeRange):
            self.neuron_item.item[i] += neuron_stimulated_voltage[i]
    def conduct(self):
        self.dendrite_item.dequeue()
        self.dendrite_item.enqueue(0)
        self.neuron_item.dequeue()
        self.neuron_item.enqueue(0)
```

```
    def get_dendrite_voltage(self):
        return self.dendrite_item.item
    def get_last_dendrite_voltage(self):
        return self.dendrite_item.item[totalTime-1]
    def get_neuron_voltage(self):
        return self.neuron_item.item
    def get_last_neuron_voltage(self):
        return self.neuron_item.item[totalTime-1]
```

이처럼, 큐를 이용해 각 뉴런의 가지돌기와 축삭돌기의 전위값을 연속적으로 표현하고, enqueue와 dequeue를 통해 간단하게 신호 펄스가 이동하도록 하였다.

2. 실시간으로 업데이트되는 그래프 그리기

본 프로그램에서 가장 중요한 것은 matplotlib 모듈에서 실시간으로 정보를 받아와 실행하는 것이었다. 기존 matplotlib에서는 animation이라는 기능이 있으나, 이는 본 프로그램에 적합하지 않아 사용할 수 없었다. 이를 해결하기 위해, 본 연구자는 pygame의 loop적 성질을 사용하였는데, graph_setting 함수를 통해 pygame 내에서 특정 위치를 클릭시 figure가 나타나게 하며, 이후 draw_graph 함수를 통해 pygame의 loop를 계속해서 지나며 그릴 수 있도록 하였다. 비록 그래프의 개수가 비교적 많아 시간 지연이 조금 있으나, pygame과 matplotlib를 엮어 프로젝트에 적합한 형태로 코드를 구성했다는 점에서 의미가 있는 어려운 작업이었다고 생각한다.

Ⅶ. 개발 과정 및 소감

나는 본 프로젝트를 진행하면서 회원가입 및 로그인을 제외한 대체적인 아이디어 구성 및 프로그래밍을 담당하였다. 특히, 생물학적 개념을 파이썬의 코드로 구현하며, 이를 자료구조를 통해 해결하는 학문 융합과 알고리즘 방향의 부분을 맡았다. 특히, 뉴런(Neuron) 클래스에 필요한 다양한 메소드들을 필요에 따라 구현하면서 코드 진행을 병행하는 점에서 어려움이 있었으나, 계획과 구현을 반복하는 과정을 통해 해결하면서 문제해결력을 기를 수 있었다. 앞으로 관련 프로젝트를 진행할 때, 구체적인 사항과 구현 계획을 미리 설정할 필요가 있다고 생각하였으며, 특히 Neuron 객체와 같이 프로그램에 적합한 객체를 만들기 위해서 더더욱 필요하다고 생각하였다.

Ⅷ. 참고 문헌 및 웹 사이트

-https://076923.github.io/posts/Python-tkinter-7/
-https://qastack.kr/programming/111155/how-do-i-handle-the-window-close-event-in-tkinter
-프로그래밍실습 학습지