

7. Hodgkin-Huxley model coding

Taehoon Kim

Contents

1. (Review) Biological backgrounds
2. (Review) Hodgkin-Huxley model
3. Hodgkin-Huxley model coding

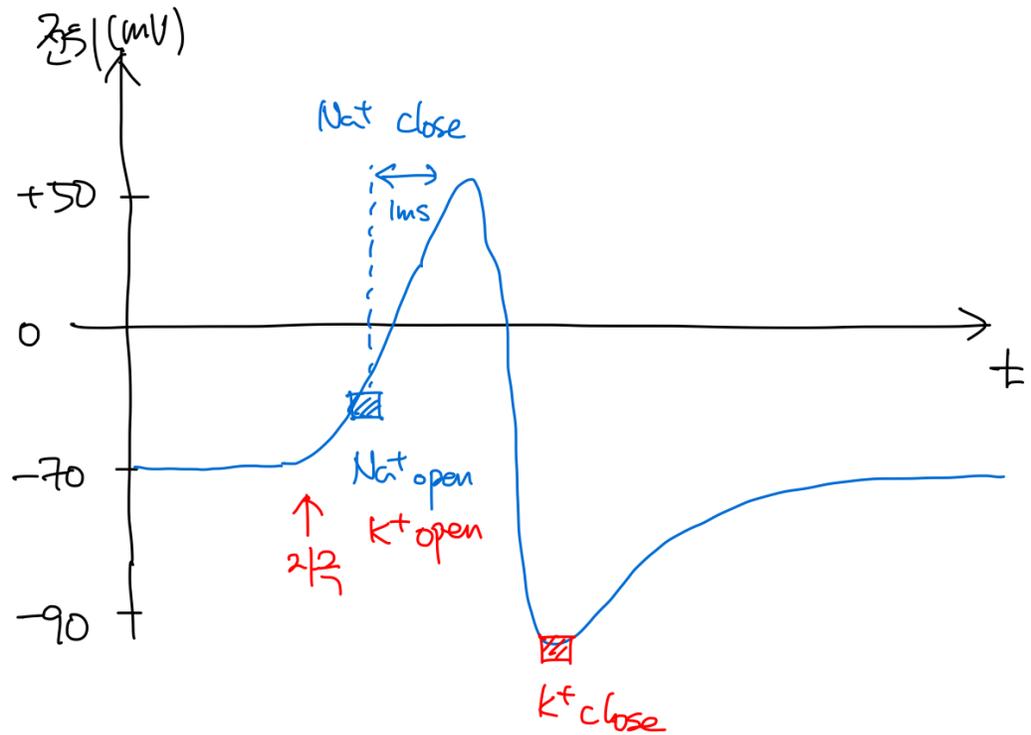
Review

Biological Backgrounds

-55mV: Voltage-gated Na⁺, K⁺ channel
open

After 1ms: Na⁺ channel closed
(Activation gate)

-90mV: Voltage-gated K⁺ channel close



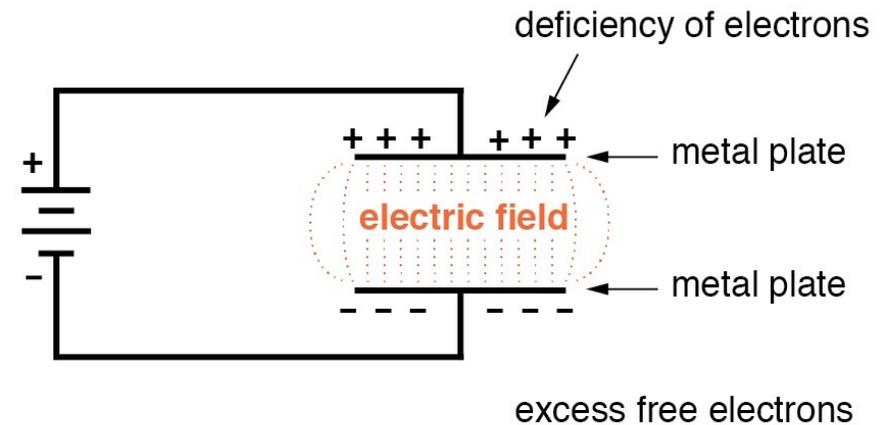
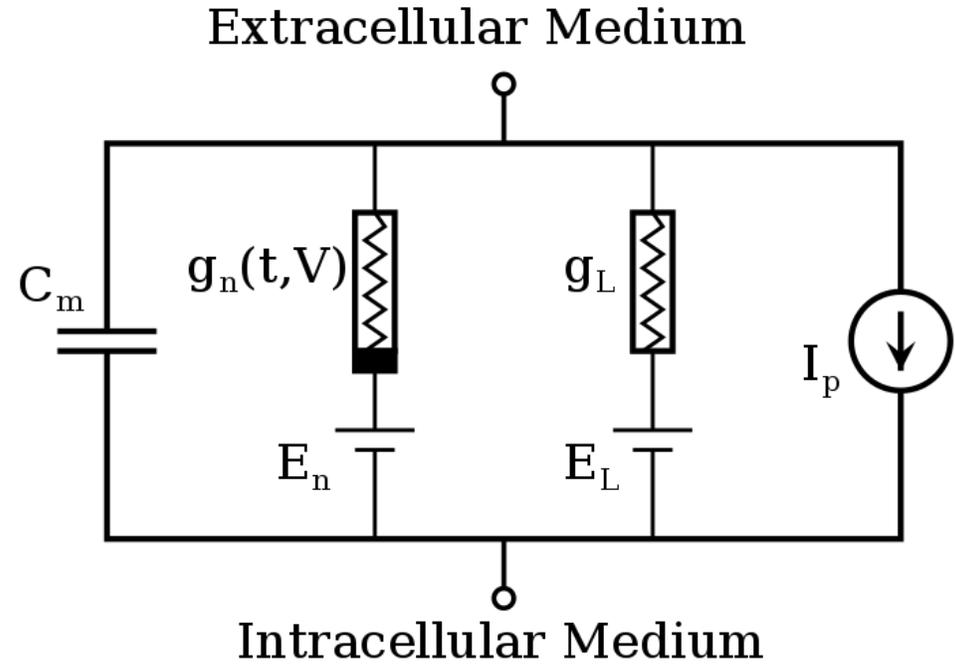
Hodgkin-Huxley model

Kirchhoff's Law

$$I_{tot} = \underbrace{I_m}_{\text{Capacitor}} + \underbrace{I_{Na} + I_K + I_{Leak}}_{\text{Conductance}}$$

Membrane Current (Capacitor)

$$Q = C_m V \Rightarrow I_m = \frac{dQ}{dt} = C_m \frac{dV}{dt}$$



Hodgkin-Huxley model

Ion flow (Conductance)

$$I \propto V \Rightarrow I = gAV \quad \begin{aligned} g_K &= \bar{g}_K n^4 \\ g_{Na} &= \bar{g}_{Na} m^3 h \end{aligned}$$

Voltage-gated Ion channel

$$\frac{dn}{dt} = \alpha_n(V)(1 - n) - \beta_n(V)n$$

$$\frac{dm}{dt} = \alpha_m(V)(1 - m) - \beta_m(V)m$$

$$\frac{dh}{dt} = \alpha_h(V)(1 - h) - \beta_H(V)h$$

$$\alpha_n = \frac{0.01(V + 55)}{1 - \exp(-0.1(V + 55))}, \quad \beta_n = 0.125 \exp(-(V + 65)/80)$$

$$\alpha_m = \frac{0.1(V + 40)}{1 - \exp(-0.1(V + 40))}, \quad \beta_m = 4 \exp(-(V + 65)/18)$$

$$\alpha_h = 0.07 \exp((V + 65)/20), \quad \beta_n = \frac{1}{1 + \exp(-0.1(V + 35))}$$

Hodgkin-Huxley model

Final solution (Hodgkin-Huxley model)

$$I_m = C_m \frac{dV}{dt} + \bar{g}_K n^4 (V - V_K) + \bar{g}_{Na} m^3 h (V - V_{Na}) + \bar{g}_L (V - V_L)$$

$$\frac{dn}{dt} = \alpha_n(V)(1 - n) - \beta_n(V)n$$

$$\frac{dm}{dt} = \alpha_m(V)(1 - m) - \beta_m(V)m$$

$$\frac{dh}{dt} = \alpha_h(V)(1 - h) - \beta_H(V)h$$

Hodgkin-Huxley model coding

Download VS code

The image shows a Google search interface for 'vscode'. The search bar contains 'vscode' and the search button is visible. Below the search bar, there are filters for '동영상', '이미지', 'Extension', '사용 하는 이유', 'Jupyter Notebook', 'C++', 'Web', and '언어 설정'. The search results show approximately 105,000,000 results in 0.31 seconds. The first result is from 'visualstudio.com' with the title 'Visual Studio Code - Code Editing. Redefined'. The snippet describes Visual Studio Code as a code editor redefined and optimized for building and debugging modern web and cloud applications. Below the snippet, there are links for 'Download', 'Docs', 'VS Code for the Web', and 'Updates'. A red box highlights the first search result. To the right of the search results, there is a knowledge panel for '비주얼 스튜디오 코드' (Visual Studio Code) with a logo and a description in Korean. The description states that Visual Studio Code is a code editor developed by Microsoft for Windows, macOS, and Linux. It features source code editing, debugging, and Git integration. The knowledge panel also lists the programming languages supported (JavaScript, C, C#, CSS, TypeScript), the developer (Microsoft), the release date (April 29, 2015), the platforms (IA-32, x86-64, AArch64), the license (MIT License), the languages supported (English, French, German, Italian, Japanese, Korean, Russian, Spanish, Chinese), and the type (source code editor, debugger).

Google

vscode

동영상 이미지 Extension 사용 하는 이유 Jupyter Notebook C++ Web 언어 설정 아이패드

모든 필터 도구 세이프서치

검색결과 약 105,000,000개 (0.31초)

visualstudio.com
https://code.visualstudio.com

Visual Studio Code - Code Editing. Redefined

Visual Studio Code is a code editor redefined and optimized for building and debugging modern web and cloud applications. Visual Studio Code is free and ...

Download
Visual Studio Code is free and available on your favorite ...

Docs
Visual Studio Code is a lightweight but powerful source code editor ...

VS Code for the Web
The VS Code Server is a private preview service you can run on ...

Updates
VS Code Profiles - Finer control with partial profiles and "Apply ...

visualstudio.com 검색결과 더보기 »

namu.wiki
https://namu.wiki > ...

Visual Studio Code - 나무위키

2023. 8. 14. — 마이크로소프트에서 개발한 텍스트 에디터로, 2015년 4월 29일에 소개되고 2016년 4월 15일에 1.0.0 정식판이 발표되었다.

비주얼 스튜디오 코드



비주얼 스튜디오 코드 또는 코드는 마이크로소프트가 마이크로소프트 윈도우, macOS, 리눅스용으로 개발한 소스 코드 편집기이다. 디버깅 지원과 Git 제어, 구문 강조 기능, SSH 접속 등이 포함되어 있으며, 사용자가 편집기의 테마와 단축키, 설정 등을 수정할 수 있다. 위키백과

프로그래밍 언어: 자바스크립트, C, C#, CSS, 타입스크립트

개발: 마이크로소프트

최초 출시일: 2015년 4월 29일

플랫폼: IA-32, x86-64, AArch64

라이선스: 소스코드: MIT 라이선스; 바이너리: 프리웨어

언어: 영어, 프랑스어, 독일어, 이탈리아어, 일본어, 한국어, 러시아어, 스페인어, 중국어 번체 및 간체

종류: 소스 코드 편집기, 디버거

Download VS code

The screenshot shows the Visual Studio Code interface with the Extensions Marketplace open. The Python extension is highlighted with a red box. The main editor shows a JavaScript file named `serviceWorker.js` with a code editor and a terminal window.

EXTENSIONS: MARKETPLACE

- Python** 2019.6.24221 54.9M ★ 4.5
Linting, Debugging (multi-threaded, ...
Microsoft **Install**
- C/C++** 0.24.0 23M ★ 3.5
C/C++ IntelliSense, debugging, and ...
Microsoft **Install**
- ESLint** 1.9.0 21.9M ★ 4.5
Integrates ESLint JavaScript into VS ...
Dirk Baeumer **Install**
- Debugger for Ch...** 4.11.6 20.6M ★ 4
Debug your JavaScript code in the C...
Microsoft **Install**
- Language Supp...** 0.47.0 18.7M ★ 4.5
Java Linting, Intellisense, formatting, ...
Red Hat **Install**
- vscode-icons** 8.8.0 17.2M ★ 5
Icons for Visual Studio Code
VSCode Icons Team **Install**
- Vetur** 0.21.1 17M ★ 4.5
Vue tooling for VS Code
Pine Wu **Install**
- C#** 1.21.0 15.6M ★ 4
C# for Visual Studio Code (powered ...
Microsoft **Install**

Code Editor:

```
src > JS serviceWorker.js > register > window.addEventListener('load') callback
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
```

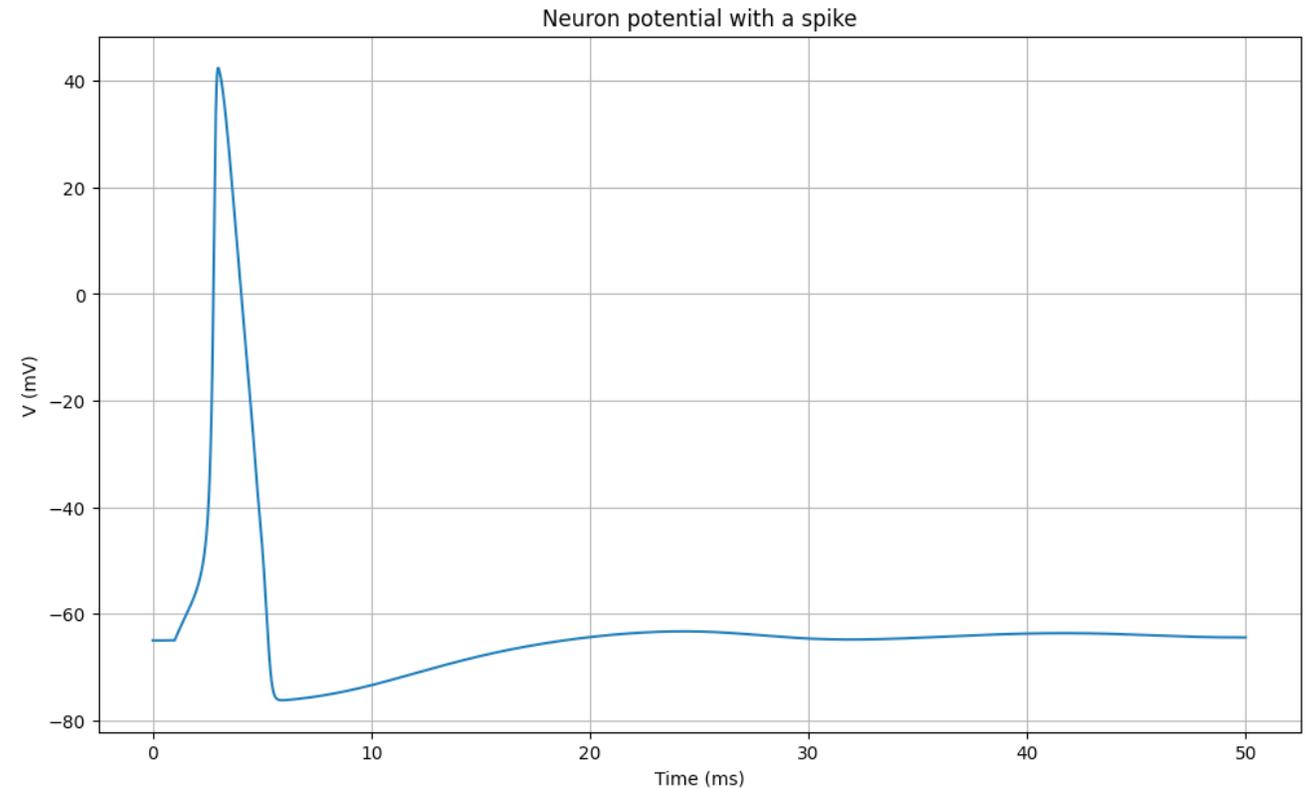
Terminal:

```
1: node
You can now view create-react-app in the browser.
Local: http://localhost:3000/
On Your Network: http://10.211.55.3:3000/
Note that the development build is not optimized.
```

Hodgkin-Huxley model coding

Our Goal:

implementing action potential with
Hodgkin-Huxley model!



Hodgkin-Huxley model coding

STEP1. Mathematical Modeling (ODE)

STEP2. Setting Parameters

STEP3. Solving ODE

STEP4. Plotting Results

Hodgkin-Huxley model coding

STEP1. Mathematical Modeling (ODE)

$$I_m = C_m \frac{dV}{dt} + \bar{g}_K n^4 (V - V_K) + \bar{g}_{Na} m^3 h (V - V_{Na}) + \bar{g}_L (V - V_L)$$

$$\frac{dn}{dt} = \alpha_n(V)(1 - n) - \beta_n(V)n$$

$$\alpha_n = \frac{0.01(V + 55)}{1 - \exp(-0.1(V + 55))}, \quad \beta_n = 0.125 \exp(-(V + 65)/80)$$

$$\frac{dm}{dt} = \alpha_m(V)(1 - m) - \beta_m(V)m$$

$$\alpha_m = \frac{0.1(V + 40)}{1 - \exp(-0.1(V + 40))}, \quad \beta_m = 4 \exp(-(V + 65)/18)$$

$$\frac{dh}{dt} = \alpha_h(V)(1 - h) - \beta_H(V)h$$

$$\alpha_h = 0.07 \exp((V + 65)/20), \quad \beta_n = \frac{1}{1 + \exp(-0.1(V + 35))}$$

Hodgkin-Huxley model coding

STEP1. Mathematical Modeling (ODE)

```
# data ref: theoretical Neuroscience
# source ref: https://www.bonaccorso.eu/2017/08/19/hodgkin-huxley-spiking-neuron-model-python/

import matplotlib.pyplot as plt
import numpy as np
from scipy.integrate import odeint
```

Hodgkin-Huxley model coding

STEP1. Mathematical Modeling (ODE)

```
# Potassium ion-channel rate functions
def alpha_n(V):
    return (0.01 * (V+55)) / (1- np.exp(-(V+55)/10))
def beta_n(V):
    return 0.125 * np.exp(-(V+65) / 80.0)

# Sodium ion-channel rate functions
def alpha_m(V):
    return (0.1 * (V + 40)) / (1-np.exp(-(V+40)/10))
def beta_m(V):
    return 4.0 * np.exp(-(V+65) / 18.0)
def alpha_h(V):
    return 0.07 * np.exp(-(V +65)/ 20.0)
def beta_h(V):
    return 1.0 / (1 + np.exp(-(V+35)))
```

Hodgkin-Huxley model coding

STEP1. Mathematical Modeling (ODE)

```
# Compute derivatives
def compute_derivatives(y, t0):
    dy = np.zeros((4,))
    V = y[0]
    n = y[1]
    m = y[2]
    h = y[3]
    # dV/dt
    GK = (gK / Cm) * np.power(n, 4.0)
    GNa = (gNa / Cm) * np.power(m, 3.0) * h
    GL = gL / Cm
    dy[0] = (Id(t0) / Cm) - (GK * (V - VK)) - (GNa * (V - VNa)) - (GL * (V - VL))
    # dn/dt
    dy[1] = (alpha_n(V) * (1.0 - n)) - (beta_n(V) * n)
    # dm/dt
    dy[2] = (alpha_m(V) * (1.0 - m)) - (beta_m(V) * m)
    # dh/dt
    dy[3] = (alpha_h(V) * (1.0 - h)) - (beta_h(V) * h)
    return dy
```

Hodgkin-Huxley model coding

STEP2. Setting Parameters

```
# Average potassium channel conductance per unit area (mS/cm^2)
gK = 36.0
# Average sodium channel conductance per unit area (mS/cm^2)
gNa = 120.0
# Average leak channel conductance per unit area (mS/cm^2)
gL = 0.3
# Membrane capacitance per unit area (uF/cm^2)
Cm = 1.0
# Potassium potential (mV)
VK = -77
# Sodium potential (mV)
VNa = 50
# Leak potential (mV)
Vl = -54.387
```

Hodgkin-Huxley model coding

STEP2. Setting Parameters

```
# Time values
# Start and end time (in milliseconds)
tmin = 0.0
tmax = 50.0
T = np.linspace(tmin, tmax, 10000)

# Input stimulus
def Id(t):
    if 1.0 < t < 3.0:
        return 10.0
    return 0.0
```

STEP3. Setting Parameters

```
# Initial Condition (V, n, m, h)
Y = np.array([-65, 0.317, 0.05, 0.6])

# Solve ODE system
Vy = odeint(compute_derivatives, Y, T)
```

Hodgkin-Huxley model coding

STEP4. Plotting Results

```
# Input stimulus
Idv = [Id(t) for t in T]
fig, ax = plt.subplots(figsize=(12, 7))
ax.plot(T, Idv)
ax.set_xlabel('Time (ms)')
ax.set_ylabel(r'Current density (uA/$cm^2$)')
ax.set_title('Stimulus (Current density)')
plt.grid()

# Neuron potential
fig, ax = plt.subplots(figsize=(12, 7))
ax.plot(T, Vy[:, 0])
ax.set_xlabel('Time (ms)')
ax.set_ylabel('V (mV)')
ax.set_title('Neuron potential with a spike')
plt.grid()
```

References

- datas & equation ref:
Theoretical Neuroscience
- source ref:
<https://www.bonaccorso.eu/2017/08/19/hodgkin-huxley-spiking-neuron-model-python/>