

Курсова работа по "Математически основи на невробиологията"

Иван Христов Христов, фак № 62130

17 Февруари 2022

Тема: Модел на първичната зрителна кора

Съдържание

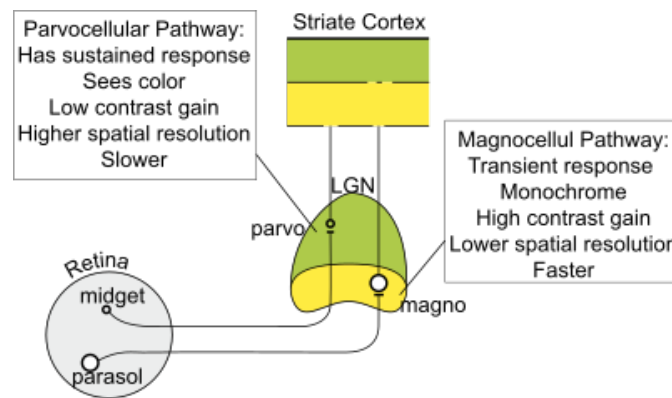
| | | |
|----------|--|----------|
| 1 | Роля и структура на първичната зрителна кора | 2 |
| 2 | Структура на нервите от окото до първичната зрителна кора | 2 |
| 3 | Програмен код - описание | 3 |
| 3.1 | Model | 4 |
| 3.2 | Layer | 5 |
| 3.3 | Conn | 6 |
| 3.4 | Net | 6 |
| 4 | Симулационни резултати | 7 |

1 Роля и структура на първичната зрителна кора

Зрителната кора е част от главния мозък, намираща се в тилния дял, която обработва визуална информация. Откъм структура, можем да кажем, че визуалната кора е разположена и в двете хемисфери.

Такъв вид информация се получава, когато тя постъпи в ретината. Там има неврони, наречени ганглионови клетки. От своя страна те предават информацията на таламуса, която играе ролята на реле за предаването ѝ. По него тя достига до първичната зрителна кора наречена още V1 (от visual area 1). V1 е една от най-изучаваните области на мозъка, но все още има много необяснени неща. След като премине през първичната зрителна кора, то информацията се предава на следващите нива, като V2, V3, V4 и други. Тя се разпространява до тях по пътища като коремното (ventral) и гръбначното (dorsal).

От своя страна първичната визуална кора е разделена на няколко слоя. Първа по трасето е Магно частта, която съдържа първия 2 слоя. След е Парво частта, която съдържа останали 4 слоя.



Фигура 1: Структура на първичната зрителна кора

2 Структура на нервите от окото до първичната зрителна кора

За да моделирам зрителната кора започвам от ретината, минавам през таламуса и стигам до нея. В тази част от мозъка са включени следните видове невронни клетки:

- Ретинна ганглионова клетка (Retinal ganglion cell) - клетките разположени в ретината, които получават първоначалния сигнал.
- LGN (Lateral geniculate nucleus) реле клетка (LGN relay cell) - клетките от таламуса, които отговарят за предаването на информацията. Често имаме съотношение 1 към 1 между броя на ганглионовите и на реле клетките. Ако ганглионова клетка има знак ON или OFF, то тя предава сигнала си на някое от реле клетките.
- LGN интерневрон (LGN Interneuron) - клетки разположени както името подсказва измежду другите неврони. Те играят ролята на инхибитори, т.е. клетки, които изсмукват електрическия сигнал от други клетки. Обичайно ганглионовите клетки предават своя сигнал на интерневрони с обратния знак. Всъщност интерневроните сами неутрализират самите себе си, както и служат за инхибитори на LGN реле клетките.

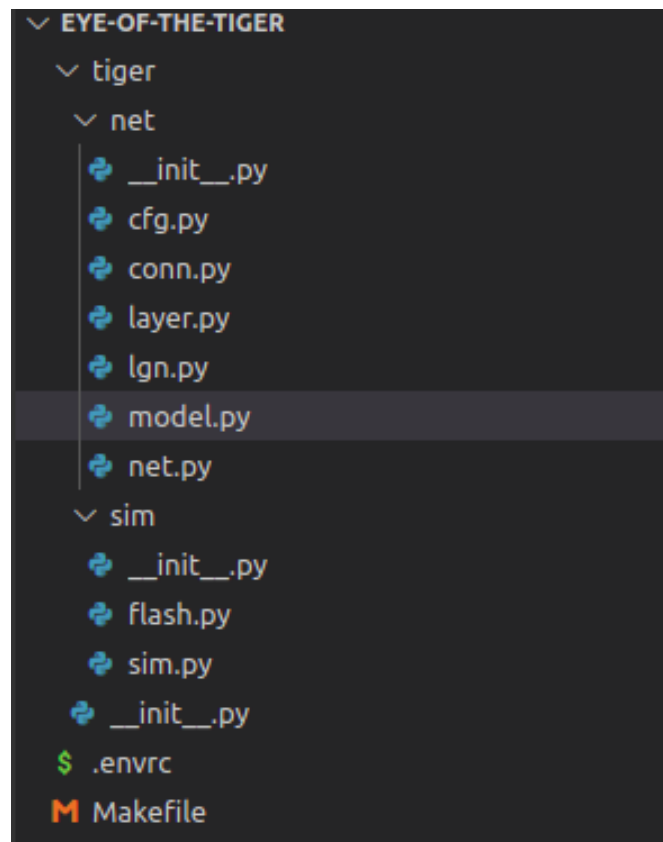
- Възбуждащи клетки от кората (Cortex excitatory cell) - те са разположени в първичната зрителна кора и получават сигнали от реле клетките. Има 3 основни вида - реагиращи еднакво добре на комбинация от светлина и цвят, реагиращи на цвят и реагиращи на светлина. Съотношението между тях не е 1 към 1.
- Инхибиторни клетки от кората (Cortex inhibitory cell) - те отново са разположени в първичната зрителна кора, като обаче са инхибитори и изсмукват заряда от другите клетки. Получават акционен потенциал от синапсите си с реле клетките. Отново има 3 вида, които са аналогични на предните - реагиращи на комбинация от светлина и цвят, реагиращи на цвят и реагиращи на светлина.

За да е по-реалистичен модела съм добавил и гаусов шум върху всяка от клетките. Това се осъществява чрез добавянето на нов модел, който предава шум под формата на сигнали към други клетки чрез връзки с тях. Не бива да забравя да кажа, че като се навържат инхибиторите с възбудителите, се получава механизъм наречен push-pull.

Сигналите се предават чрез spike генератори, които са свързани към ретините.

3 Програмен код - описание

Проекта е направен на питон, като са използвани NEST и Numpy.



Фигура 2: Структура на проекта

Проекта е имплементиран в пакета tiger. В него има съответно два подпакета. Към момента на писане само пакета net е имплементиран, докато пакета sim не е напълно. Заради това фокуса ще бъде над net.

Очевидно net е разбит на много подмодули. Първия по-важен е csg, където е дефиниран модел, който съдържа конфигурацията, която е обща за много от функциите при генериране на мрежата.

3.1 Model

Следва model модула. Той съдържа дефиниции на моделите на невроните, които влизат в състава на първичната зрителна кора.

```
1 def _retinal_ganglion_cell() -> Model:
2     params = {"origin": 0.0, "start": 0.0}
3     return Model(SPIKE_GENERATOR, params)
4
5
6 def _lgn_relay_cell() -> Model:
7     params = {
8         "C_m": 100.0,
9         "g_L": 10.0,
10        "E_L": -60.0,
11        "V_th": -55.0,
12        "V_reset": -60.0,
13        "t_ref": 2.0,
14        "E_ex": 0.0, # AMPA, from Hill-Tononi 2005
15        "E_in": -80.0, # GABA-A of thalamocortical cells, from Hill-Tononi 2005
16        "tau_syn_ex": 1.0, # it approximates Hill-Tononi's diff. of exp. response,
17        also Casti 2008
18        "tau_syn_in": 3.0 # it approximates Hill-Tononi's diff. of exp. response
19    }
20    return Model(IAF_COND_ALPHA, params)
```

Listing 1: Дефиниция на ганглионова клетка

Това, което се забелязва, че поставяме стойности на параметрите спрямо тези, които желаем да получим в крайното уравнение за потенциала. Не бива да забравяме, че всеки неврон го моделираме по специфичен начин спрямо някакъв модел като например Зинкевич или Ходжкин-Хъксли. Тези стойности, които са поставени там са иззети от научни статии (както посочват коментарите отстрани), т.е. не съм си играл аз да ги мисля.

```
1 def get_models() -> List[Tuple[str, str, Dict]]:
2     retinal_ganglion_cell = _retinal_ganglion_cell()
3     lgn_relay_cell = _lgn_relay_cell()
4     lgn_interneuron_cell = _lgn_interneuron_cell()
5     cortex_exc_cell = _cortex_excitatory_cell()
6     cortex_inh_cell = _cortex_inhibitory_cell()
7     thalamo_noise = _thalamo_noise()
8
9     return [
10         (retinal_ganglion_cell.name, RETINAL_GANGLION_CELL, retinal_ganglion_cell.
11          params),
12         (lgn_relay_cell.name, LGN_RELAY_CELL, lgn_relay_cell.params),
13         (lgn_interneuron_cell.name, LGN_INTERNEURON, lgn_interneuron_cell.params),
14         (cortex_exc_cell.name, CORTEX_EXC_CELL, cortex_exc_cell.params),
15         (cortex_inh_cell.name, CORTEX_INH_CELL, cortex_inh_cell.params),
16         (thalamo_noise.name, THALAMO_NOISE, thalamo_noise.params),
17     ]
```

Listing 2: Извличане на всички модели

Модулът model разполага с функция, чрез която директно можем да извлечем всичките модели, които ни интересуват, т.е. тези, които описав в предната секция. Веднъж щом са извлечени могат

да бъдат приложени чрез Сору функцията на NEST.

3.2 Layer

Дефиницията на слоевете, техните размери и брой елементи се случва в този модул. Основно слоевете могат да бъдат разделени на. Такива за LGN, които включва невроните на ретината, интерневроните и таламуса. Освен тях има група слоеве за първичната мозъчна гора, като тя е разделена на две групи, съответно за възбудители и за инхибитори в зависимост от функцията на невроните в тях. Не на последно място ще спомена, че има и допълнителен слой, който е за невроните причиняващи шум.

```
1 def layers(cfg:Config) -> Tuple[str, Dict]:
2     ls = [_lgn_layers(cfg)]
3     ls += [_cortex_color_luminance_exc_layers(cfg),
4           _cortex_luminance_preferring_exc_layers(cfg),
5           _cortex_color_preferring_exc_layers(cfg)]
6     ls += [_cortex_color_luminance_inh_layers(cfg),
7           _cortex_luminance_preferring_inh_layers(cfg),
8           _cortex_color_preferring_inh_layers(cfg)]
9     ls += [_noise_gen_layers(cfg)]
10
11     return ls
```

Listing 3: Извличане на всички слоеве

```
1 def _lgn_layers(cfg: Config) -> Tuple[str, Dict]:
2     base_props = _base_lgn_layer_props(cfg)
3
4     midget_ganglion_cell_layers = [
5         (MIDGET_GANGLION_CELLS_L_ON, _merged_dicts(base_props, {'elements': mdl.
6             RETINAL_GANGLION_CELL})),
7         (MIDGET_GANGLION_CELLS_L_OFF, _merged_dicts(base_props, {'elements': mdl.
8             RETINAL_GANGLION_CELL})),
9         (MIDGET_GANGLION_CELLS_M_ON, _merged_dicts(base_props, {'elements': mdl.
10             RETINAL_GANGLION_CELL})),
11         (MIDGET_GANGLION_CELLS_M_OFF, _merged_dicts(base_props, {'elements': mdl.
12             RETINAL_GANGLION_CELL})),
13     ]
14     parvo_lgn_relay_cell_layers = [
15         (PARVO_LGN_RELAY_CELL_L_ON, _merged_dicts(base_props, {'elements': mdl.
16             LGN_RELAY_CELL})),
17         (PARVO_LGN_RELAY_CELL_L_OFF, _merged_dicts(base_props, {'elements': mdl.
18             LGN_RELAY_CELL})),
19         (PARVO_LGN_RELAY_CELL_M_ON, _merged_dicts(base_props, {'elements': mdl.
20             LGN_RELAY_CELL})),
21         (PARVO_LGN_RELAY_CELL_M_OFF, _merged_dicts(base_props, {'elements': mdl.
22             LGN_RELAY_CELL})),
23     ]
24     parvo_lgn_interneuron_layers = [
25         (PARVO_LGN_INTERNEURON_ON, _merged_dicts(base_props, {'elements': mdl.
26             LGN_INTERNEURON})),
27         (PARVO_LGN_INTERNEURON_OFF, _merged_dicts(base_props, {'elements': mdl.
28             LGN_INTERNEURON})),
29     ]
30
31     return midget_ganglion_cell_layers + parvo_lgn_relay_cell_layers +
32           parvo_lgn_interneuron_layers
```

Listing 4: Дефиниция на слоевете за LGN като те са разделени по ON и OFF в зависимост от

3.3 Conn

Този модул е може би най-сложния и досаден за имплементация. В общи линии той съдържа това, как навръзваме различните слоеве от нашата мрежа. За всяка една връзка се дефинира каква да бъде свързаността, като за свързаността между невроните от LGN се използва кръгловидна, а между невроните от релето и тези от първичната кора се използва правоъгълна форма.

```

1 # Returns connections between layers.
2 def get_connections(cfg: Config) -> List:
3     pop_size = lyr.pop_size_from_cfg()
4     _copy_synapse_model()
5
6     # LGN connections
7     conns = _retinal_ganglion_cells_to_relay_cells(cfg)
8     conns += _retinal_ganglion_cells_to_interneurons(cfg)
9     conns += _interneurons_to_relay_cells(cfg)
10    conns += _interneurons_to_interneurons(cfg)
11
12    # Thalamocortical connections
13    # Excitatory
14    conns += _relay_cells_to_color_luminance_cells(cfg, pop_size)
15    conns += _relay_cells_to_luminance_preferring_cells(cfg, pop_size)
16    conns += _relay_cells_to_color_preferring_cells(cfg, pop_size)
17
18    # Inhibitory
19    conns += _relay_cells_to_color_luminance_inh_cells(cfg, pop_size)
20    conns += _relay_cells_to_luminance_preferring_inh_cells(cfg, pop_size)
21    conns += _relay_cells_to_color_preferring_inh_cells(cfg, pop_size)
22
23    # Gaussian noise connections
24    conns += _noise_to_relay_cells(cfg)
25    conns += _noise_to_interneuron(cfg)
26    conns += _noise_to_excitatory_cortex_cells(cfg)
27    conns += _noise_to_inhibitory_cortex_cells(cfg)
28
29    return conns

```

Listing 5: Дефиниция на всички конекции

3.4 Net

Накрая за да се конструира мрежата се извличат всичките модели, слоеве и конекциите между тях чрез една проста функция дефинирана в net модула.

```

1 def get_network(cfg: cfg.Config) -> Tuple:
2     models = mdl.get_models()
3     layers = lyr.layers(cfg)
4     conns = conn.get_connections(cfg)
5
6     return models, layers, conns

```

Listing 6: Генериране на цялата мрежа

4 Симулационни резултати

В момента на писане на този документ е почти 4 сутринта и за една нощ реализирах само мрежата в NEST. Това се забелязва и от предадения код. Предстои ми да направя входна точка на програмата, където да се вика симулация. Най-вероятно ще кача това, което съм направил малко по-късно през деня преди крайния срок и ще доразширя леко документацията конкретно в тази секция. Предавам го почти завършено за да може да прочетете поне основната част, а именно архитектурата на мрежата.