

# Исследование спайкового кода

## Содержание

<b>1</b>	<b>Спайковый код</b>	<b>2</b>
<b>2</b>	<b>Спайковый нейрон</b>	<b>2</b>
2.1	Модель Integrate-and-fire . . . . .	3
<b>3</b>	<b>Пример: кодирование временного ряда</b>	<b>4</b>

## 1 Спайковый код

Спайк - элементарная единица информации, которой обмениваются биологические нейроны. Соединённые синаптическими связями, нейроны, сложно и нелинейно перерабатывая вход с синапсов, вырабатывают на своём выходе (аксон нейрона), короткие электрохимические импульсы. В анализе удобно рассматривать спайки, как бинарные события (рис. 1).

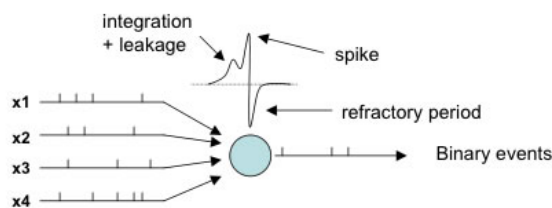


Рис. 1: Спайковый нейрон

То как нейроны кодируют информацию в спайках, очень живой и трепещущий вопрос для современного научного сообщества. Важный нюанс спайкового кода в том, что он не надёжен. Исследования показали, что на один и тот же стимул популяция нейронов может дать разный ответ, что даёт огромное пространство для интерпретаций.

На протяжении 20-ого века большинство нейрофизиологов было убеждено, что ненадёжность разрешается, если усреднять спайки на некотором промежутке времени (около 100 мс), и вся информация хранится в средних активностях нейронов. Однако, в конце 20-ого века было проведено множество исследований, показавших, что достаточно много информации хранится в самих временах спайков, и что усреднение, только ухудшает декодирование информации полученной от нейронов.

В данном задании будет возможность закодировать сигнал в виде спайкового кода, через симуляцию популяции нейронов. Получив ответ в виде спайков, мы его декодируем, и необходимо будет сделать анализ полученных результатов, при помощи инструментов данных в этом руководстве.

## 2 Спайковый нейрон

На рис. 1, также, показан типичный профиль активности нейрона, основные свойство которого:

- интеграция входного сигнала (integration);
- угасание этого сигнала на нейроне со временем, или иначе говоря “утечка” (leakage);
- рефракторный период, нейрон переживает его после выработки спайка, как следствие сложных химических реакций, некоторое время (от 2-10 мс) неработоспособен.

Моделирование спайковых нейронов в виде наиболее приближенном к биологии, насколько позволяет современная нейронаука, возможно, но очень

трудозатратно с точки зрения ресурсов компьютера. Существует модели, более менее, приближенные к биологическому аналогу и которые не так сложно моделировать. Две из них будут рассмотрены в задании.

## 2.1 Модель Integrate-and-fire

Самая простая спайковая модель, основанная на RC цепи, записывается в виде дифференциального уравнения

$$\tau_m \frac{du}{dt} = -u + RI(t), \quad (1)$$

при  $u \geq \vartheta$  потенциал мембраны сбрасывается и на  $\tau_{ref}$  держится в сброшенном состоянии

$$u \leftarrow u_r \text{ в течении } \tau_{ref}, \quad (2)$$

где  $\vartheta$  - порог напряжения, временная константа мембраны  $\tau_m = RC$ ,  $R$  и  $C$  - сопротивление и ёмкость RC-цепи соответственно,  $\tau_{ref}$  - рефракторное время,  $I(t)$  - приложенный ток извне,  $u_r$  - константа описывающая потенциал мембраны покоя нейрона.

Пример работы такого нейрона можно посмотреть на рис.2.

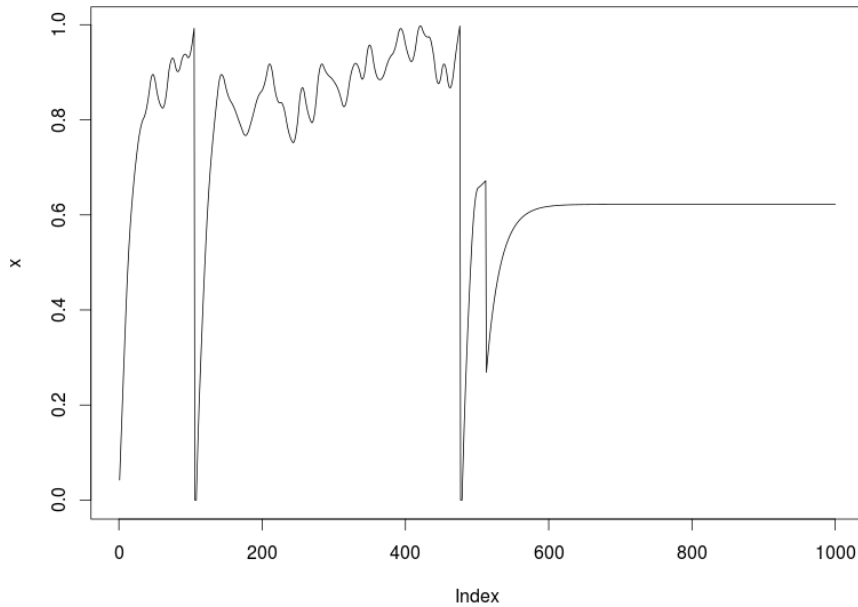


Рис. 2: Напряжение на мембране IaF, при  $u_r = 0, \vartheta = 1, \tau_{ref} = 2$  мс,  $\tau_m = 20$  мс

### 3 Пример: кодирование временного ряда

В качестве среды используется пакет **R**. Также используется набор скриптов для симуляции нейронов, его можно взять из репозитория: [https://github.com/alexeyche/spike\\_code\\_project.git](https://github.com/alexeyche/spike_code_project.git)

Возьмём временной ряд из набора *synthetic control* и нормируем его в пределах  $[-1, 1]$ :

```
1 > X = loadMatrix("~/ts/synthetic_control/synthetic_control_
  TRAIN_120", 1)[1,]
2 > X = 2*(X-min(X))/(max(X)-min(X)) - 1
3 > plot(X, type="l")
```

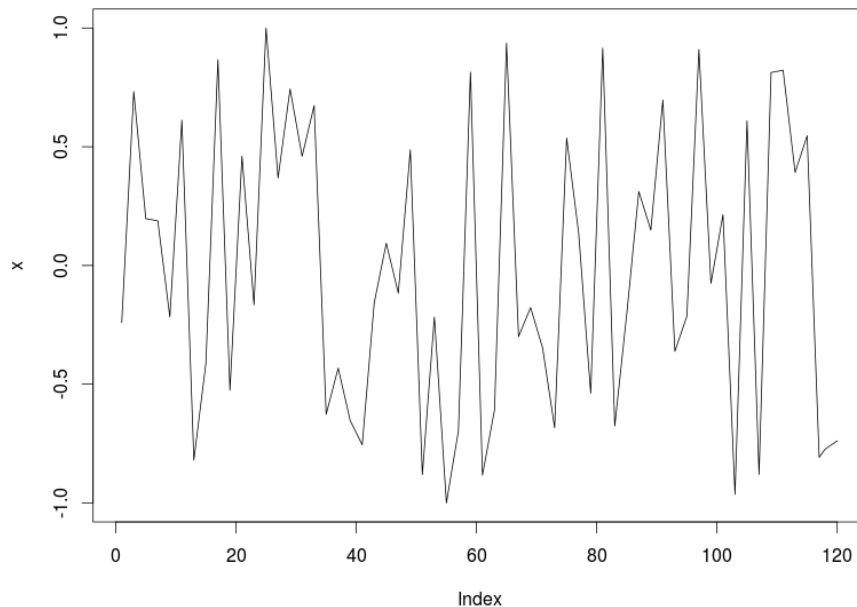


Рис. 3: Временной ряд

Приложение  $I(t)$  к каждому нейрону будет происходить в виде

$$I_i(t) = g_i X + b_i \quad (3)$$

где  $g_i$  и  $b_i$  усиление и смещение, которое для каждого нейрона уникально.

Используя данный набор скриптов можно сгенерировать усиление и смещение для каждого нейрона, таким образом чтобы обеспечить равномерный охват сигнала  $X$

```

1 > rate_low = 0 # Herz
2 > rate_high = 50 # Herz
3 > M = 25 # Size of neuron population
4 > c(gain, bias) := generate_gain_and_bias(M, rate_low, rate_
    high)
5 > gain
6 [1] 0.6758357 1.7004524 0.2071122 0.2756547 0.4887354
    1.6412549 0.3116990 0.7759411 0.6372584 2.2066023
7 [11] 0.3249702 1.4093830 0.4640132 0.8744656 0.3372742
    0.6675960 1.9251885 4.0983182 0.9741277 0.1989673
8 [21] 0.7413383 0.2470770 2.1391720 0.1389673 0.4076056
9 > bias
10 [1] 0.76748365 -0.08399640 1.15624594 1.19207789
    0.93562484 0.02672873 1.16662126 0.47109941
11 [9] 0.76147893 -0.76017726 1.02079654 0.09442703
    0.83244672 0.75302014 0.94207371 0.87191443
12 [17] -0.46682617 -2.50856272 0.54474359 1.11589300
    0.91945595 0.94598459 -0.54479961 1.09119150
13 [25] 1.21586165

```

Также распределим ответственность для каждого нейрона, за отрицательную и положительную часть графика, сгенерировав равномерное распределение из набора  $\{-1, 1\}$

```

1 > encoder = sample(c(1, -1), M, replace=TRUE)
2 > encoder
3 [1] -1 1 -1 1 -1 -1 1 1 -1 1 -1 -1 1 1 1 1 1 1
    -1 1 1 -1 1 1 1

```

После всех подготовок, можно создать список с переменными для каждого нейрона, взять первую точку временного ряда, и прогнать его через популяцию *IaF* нейронов, с характеристиками, как на рис.2

```

1 > n = list(v=rep(0, M), ref=rep(0, M))
2 > input = X[1] * encoder * gain + bias
3 > c(n, current_spikes) := run_neurons(input, n)
4 > current_spikes
5 [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
    TRUE FALSE FALSE FALSE FALSE TRUE FALSE FALSE
6 [18] FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE
    FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
7 [35] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
    FALSE FALSE FALSE FALSE FALSE FALSE FALSE
8 > which(current_spikes)
9 [1] 10 15 20

```

*current\_spikes* содержит вектор булевских значений, каждое значение, отвечает за то издал спайк этот нейрон или нет

Прогнав цикл по всем точкам

```
1 spikes = NULL
2 for(i in 1:length(X)) {
3   x = X[i]
4   input = x * encoder * gain + bias
5
6   c(n, current_spikes) := run_neurons(input, n)
7   spikes = cbind(spikes, as.integer(current_spikes))
8 }
```

мы получим матрицу 25x120, значения которой будут говорить о том, издал ли данный нейрон в данный момент спайк или нет

```
1 > gr_pl(t(spikes))
```

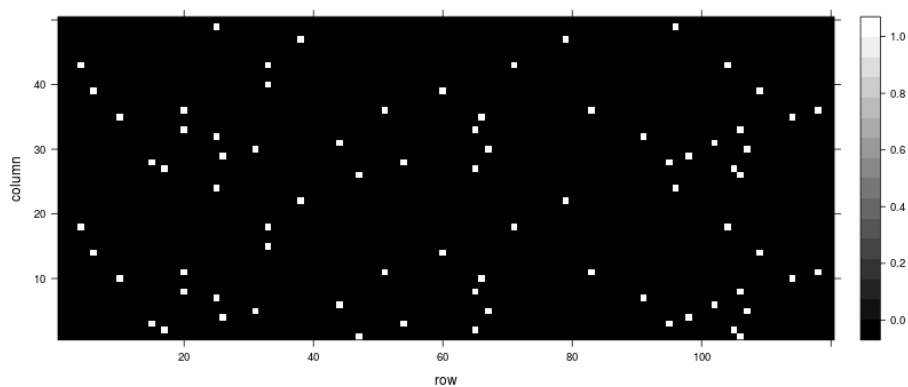


Рис. 4: Контурный график спайков