

# 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
2.2	Модель Adaptive Exponential Integrate-and-fire . . . . .	4
2.3	Кодирование спайковым кодом . . . . .	5
2.3.1	Линейная кривая настройки . . . . .	5
2.3.2	Сигмоидная кривая настройки . . . . .	6
2.3.3	Критерий оценки . . . . .	6
2.4	Задание . . . . .	7
2.5	Материалы . . . . .	7

## 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).

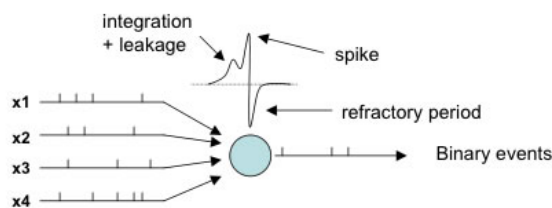


Рис. 1: Спайковый нейрон

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.

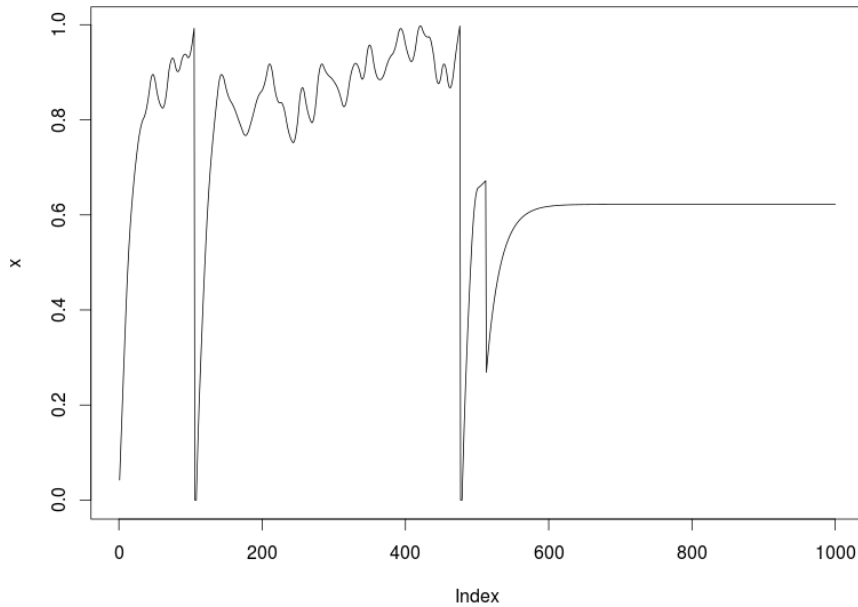


Рис. 2: Напряжение на мембране IaF, при  $u_r = 0, \vartheta = 1, \tau_{ref} = 2$  мс,  $\tau_m = 20$  мс

## 2.2 Модель Adaptive Exponential Integrate-and-fire

Более сложная, но более приближенная к биологии модель спайкового нейрона, описывается системой дифференциальных уравнений с двумя переменными. Основная особенность поведения системы, в том, что нейрон адаптируется, т.е. со временем показывает меньшую активность при одинаковой характеристике входа

$$\begin{aligned} C \frac{du}{dt} &= -g_L(V - E_L) + g_L \Delta_T \exp\left(\frac{V - V_T}{\Delta_T}\right) - w + I(t) \\ \tau_w \frac{dw}{dt} &= a(V - E_L) - w \end{aligned} \quad (3)$$

Генерация спайка происходит аналогично обычной модели IaF:

$$u \leftarrow u_r \text{ в течении } \tau_{ref}, \quad (4)$$

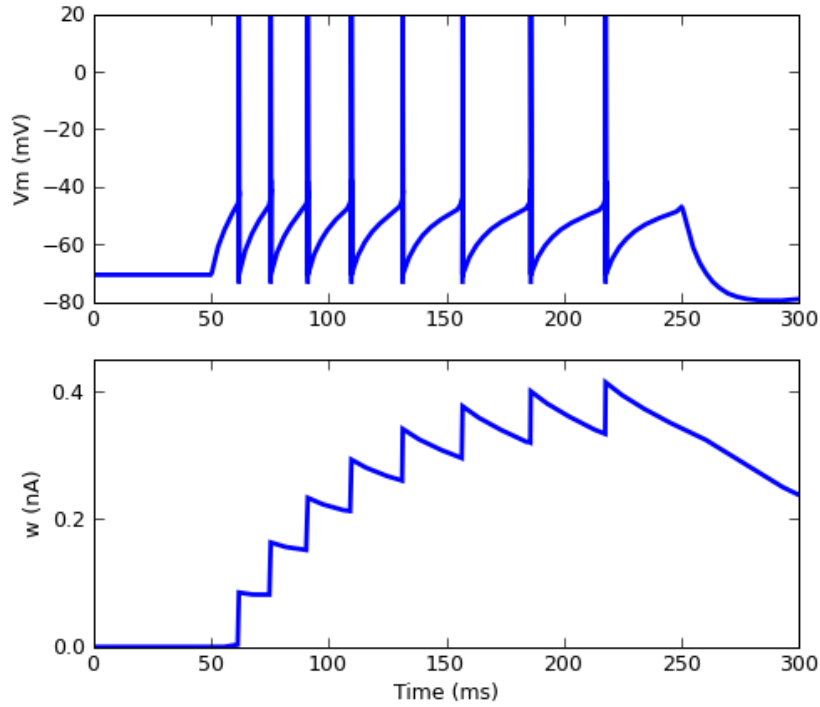


Рис. 3: Поведение переменных AdEx IaF

## 2.3 Кодирование спайковым кодом

Задача кодирования спайковым кодом временного ряда  $X(t)$  популяцией  $N$  нейронов, заключается в том, чтобы найти такое преобразование  $X(t) \rightarrow I_i(t)$ , для  $i \in \{1, \dots, N\}$ , которое является наиболее оптимальным по данному критерию оценки.

В качестве критерия оценки возьмём качество линейного восстановления сигнала  $X(t)$  из спайкового кода.

### 2.3.1 Линейная кривая настройки

Под линейной кривой настройки понимается, что нейрон чувствителен к определённому значению временного ряда, причём частота его спайков, линейно возрастает с возрастанием или убыванием значения

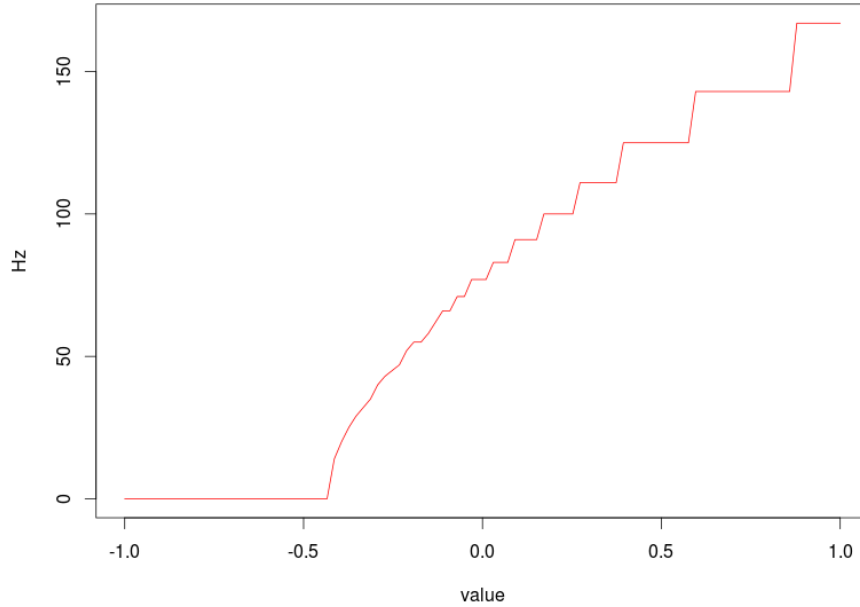


Рис. 4: Линейная кривая настройки

Как для IaF, так и для AdEx IaF, такая кривая настройки задаётся преобразованием

$$I_i(t) = g_i X(t) + b_i \quad (5)$$

где  $g_i, b_i \in \mathbb{R}$  – усиление и смещение преобразования, соответственно.

### 2.3.2 Сигмоидная кривая настройки

Сигмоидная кривая настройки получается преобразованием вида

$$I_i(t) = g_i^s \exp\left(-\frac{(X(t) - C_i)^2}{2\sigma_i^2}\right) \quad (6)$$

где  $g_i^s, C_i, \sigma_i \in \mathbb{R}$  – усиление, центр и разброс сигмоидной кривой, соответственно.

На рисунке показан типичный профиль этой кривой настройки

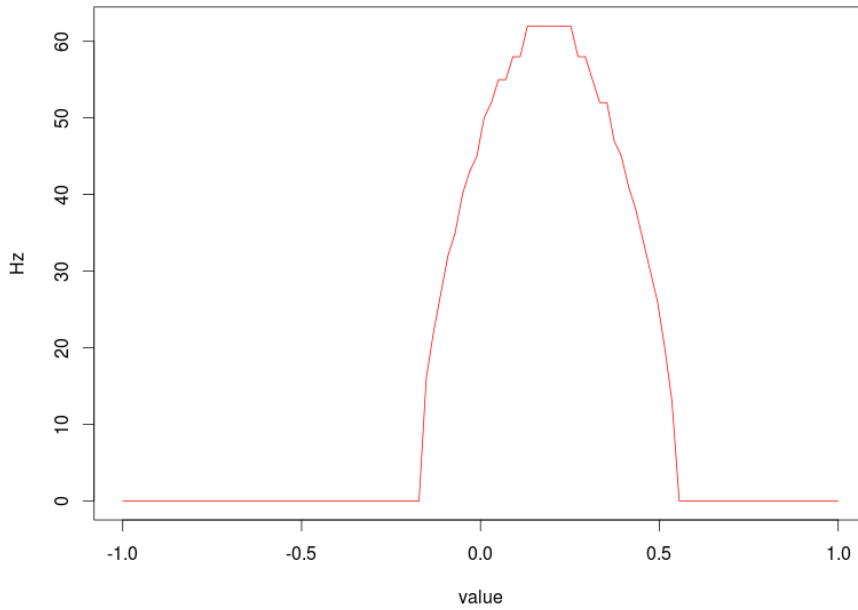


Рис. 5: Сигмоидная кривая настройки

### 2.3.3 Критерий оценки

В качестве критерия оценки возьмём качество линейного восстановления исходного сигнала  $X(t)$ . Фильтр Винера – один из самых известных и простых линейных фильтров. После обучения фильтра, фильтр с тем или иным качеством восстанавливает сигнал, и это качество можно выразить отношением сигнал-шум:

$$SNR = \frac{X(w)}{X(w) - X'(w)}, \quad (7)$$

где  $X'(w), X(w)$  – восстановленный сигнал и исходный сигнал в частотном домене.

В качестве базовой метрики качества спайкового кодирования возьмем информацию на каждый частотный канал:

$$Info = \frac{1}{2} \log_2(1 + SNR) \quad (8)$$

## 2.4 Задание

Основная задача состоит в исследовании влияния параметров кривых настроек и нейрона на критерий оценки (8). Провести мета-оптимизацию по параметрам для максимизации критерия.

**Вариант 1.** Исследование линейных кривых настроек для определенных данных. Мета-оптимизация методом *stas*.

**Вариант 2.** Исследование сигмоидных кривых настроек и настроек нейрона для определенных данных. Мета-оптимизация методом *GP*.

## 2.5 Материалы

Нейронные модели, кривые настройки, оптимизация фильтров Винера находятся в репозитории:

[https://github.com/alexeyche/snn\\_sim.git](https://github.com/alexeyche/snn_sim.git)

Репозиторий содержит научный проект в виде библиотеки на C, который имеет, как обычную точку входа в виде программ, так и пакет для языка **R**.

Для того чтобы собрать пакет для **R**, необходимо собрать и установить библиотеку:

```
1 $ git clone https://github.com/alexeyche/snn_sim.git
2 $ mkdir snn_sim/build
3 $ cd snn_sim/build
4 $ cmake ../sources
5 $ make -j8
6 $ sudo make install
```

далее, для сборки пакета **R** необходимо выполнить скрипт:

```
1 $ cd ../r_package
2 $ ./build.sh
```

Примеры построения кривых настроек можно посмотреть в скриптах, для линейных кривых:

```
1 snn_sim/r_package/r_scripts/linear_curves_test.R
```

для сигмоидных кривых:

```
1 snn_sim/r_package/r_scripts/sigma_curves_test.R
```