МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики.

Лабораторная работа №1

# Моделирование биологических систем зрительного восприятия

по дисциплине
Компьютерное зрение

Выполнил студент группы М3403:
Давлетов Артем Эдуардович

Преподаватель:
Титаренко Михаил Алексеевич

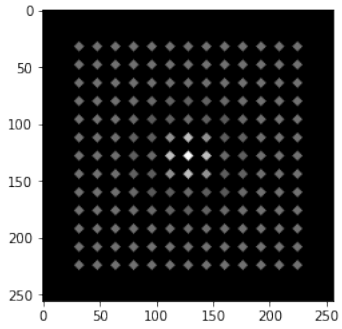
2020

Цель работы: исследовать возможность компьютерного моделирования поведения ганглиозных клеток сетчатки и нейронов стриарной коры либо в форме искусственных нейронных сетей с повторением структуры рецептивных полей, либо в форме последовательности фильтров, осуществляющих свертку изображения с различными ядрами.

Реакцию ганглиозных клеток можно представить как результат применения фильтра, ядро которого определяется как разность гауссиан с разной дисперсией. С помощью функций библиотеки OpenCV такая фильтрация может быть вычислена путем следующей серии вызовов:

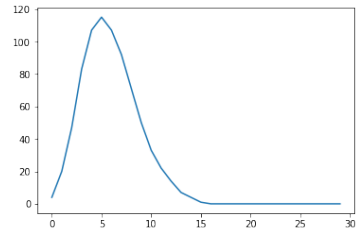
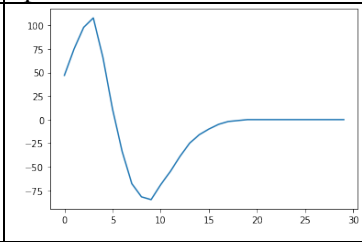
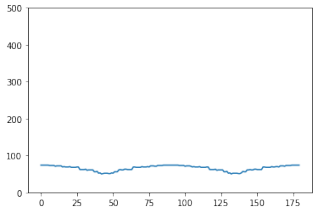
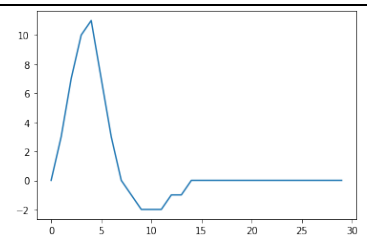
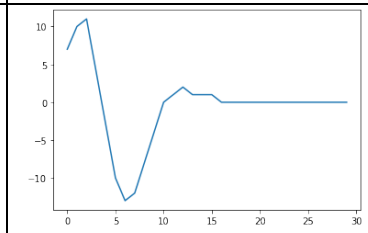
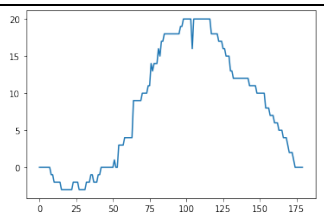
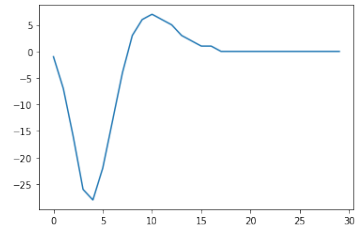
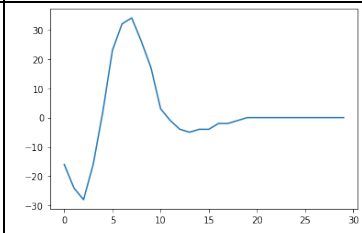
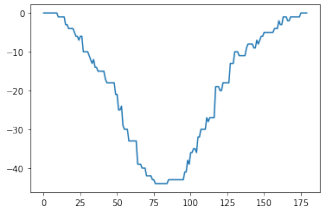
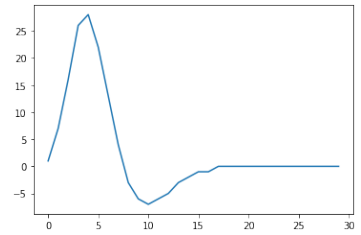
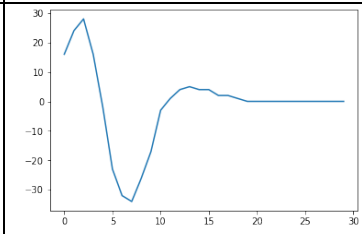
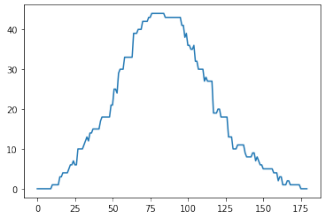
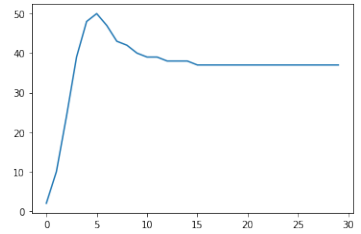
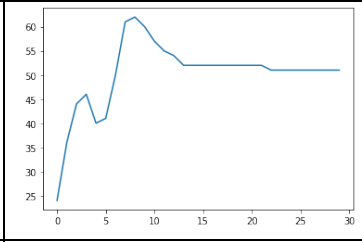
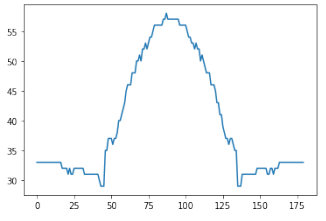
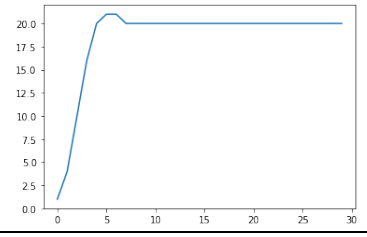
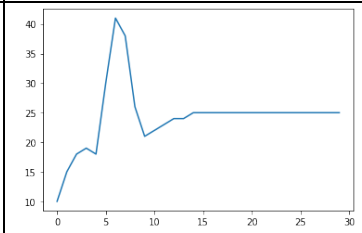
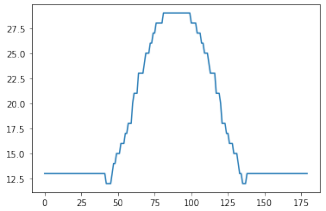
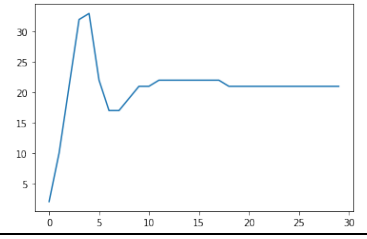
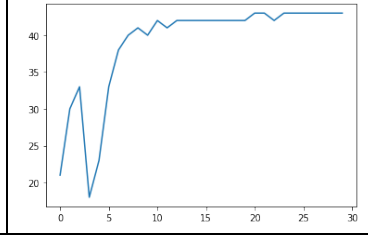
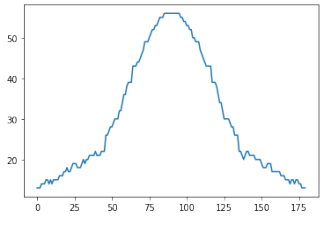
```
gauss_d1 = cv2.GaussianBlur(image, (3, 3), 0)
gauss_d2 = cv2.GaussianBlur(image, (7, 7), 0)
```

Для проведения требуемого в работе эксперимента требуется исследование реакции одной ганглиозной клетки на разные стимулы. Сначала установим реакцию клетки на световое пятно малого размера.




Здесь фиксируются отклики клетки на «точечные» стимулы, находящиеся в разных местах рецептивного поля. Центр рецептивного поля клетки совпадает с центром изображения.

Построим графики зависимости отклика ганглиозных, простых и сложных клеток на светлое пятно на темном фоне, светлое кольцо на темном фоне, на лини, в зависимости от разных типов стимулов: граница, тёмная полоса на светлом фоне и светлая полоса на темном фоне.

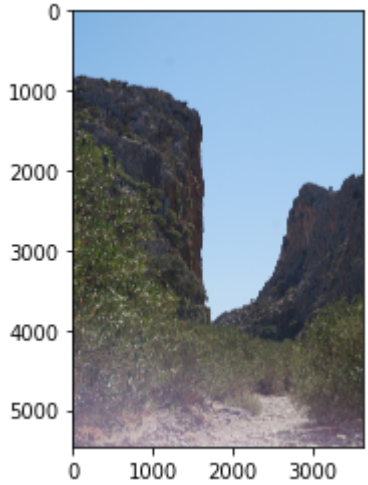
| | Фильтр | Светлое пятно на темном фоне | Светлое кольцо на темном фоне | Линия |
|---|---|---|---|---|
| Реакция ганглиозных клеток | - |  |  |  |
| Реакция простых клеток | 1  |  |  |  |
| | 2  |  |  |  |
| | 3  |  |  |  |
| Реакция сложных клеток | 1  |  |  |  |
| | 2  |  |  |  |
| | 3  |  |  |  |

Модель, эмулирующая работу ганглиозных клеток
сетчатки, имеет центрально-симметричное рецептивное поле и не обладает дирекционной избирательностью. Максимальный отклик обнаруживается на светлое пятно некоторого размера на тёмном поле. При дальнейшем увеличении размера пятна отклик уменьшается. Подавление активности клетки наблюдается, когда в качестве стимула задается светлое кольцо на тёмном фоне. Данные результаты согласуются с основными свойствами ганглиозных клеток сетчатки.

Реализация прикреплена ниже в виде jupyter notebook:

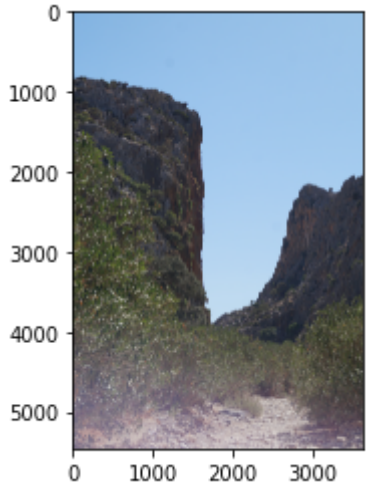
```
import cv2
from matplotlib import pyplot as plt
import numpy as np
from scipy.signal import convolve2d


img = cv2.imread('img.JPG')
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
plt.imshow(img)
```

```
<matplotlib.image.AxesImage at 0x7f71af8ce358>
```




```
gauss_d1 = cv2.GaussianBlur(img, (3, 3), 0)
gauss_d2 = cv2.GaussianBlur(img, (7, 7), 0)
plt.imshow(gauss_d1)
plt.imshow(gauss_d2)
```

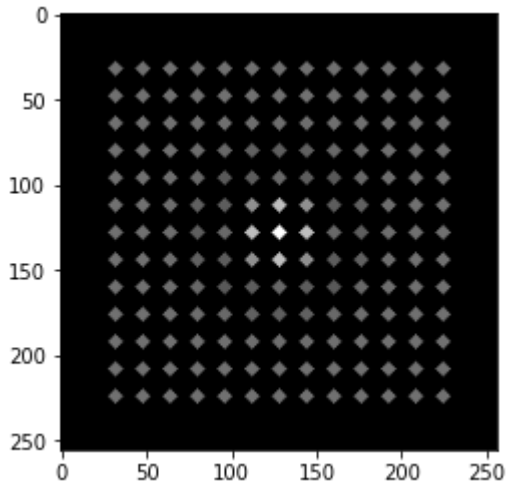
```
<matplotlib.image.AxesImage at 0x7f71af9034a8>
```




```
image = np.zeros((256, 256), np.float32)
gauss_d1 = np.zeros((256, 256), np.float32)
gauss_d2 = np.zeros((256, 256), np.float32)
ganglionic = np.zeros((256, 256), np.float32)
res = np.zeros((256, 256), np.float32)


for i in range(13):
```

```
    for j in range(13):
        image = np.zeros((256, 256), np.float32)
        cv2.circle(image, (j + 122, i + 122), 1, (127, 127, 127), -1, 8, 0)
        gauss_d1 = cv2.GaussianBlur(image, (3, 3), 0)
        gauss_d2 = cv2.GaussianBlur(image, (7, 7), 0)
        ganglionic = cv2.subtract(gauss_d1, gauss_d2)
        v = ganglionic[128][128] * 3 + 128
        cv2.circle(res, (j * 16 + 32, i * 16 + 32), 2, (v, v, v), 3, 8, 0)


plt.imshow(res, cmap='gray')
```

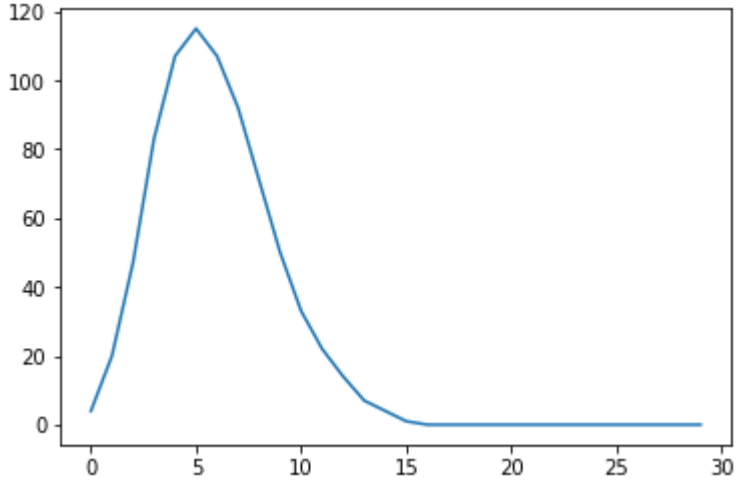
```
<matplotlib.image.AxesImage at 0x7f71af880978>
```




```
def circle(filter=None, is_complex=False):
    chart = []
    res = np.zeros((256, 256), np.float32)
    v_prev = 0
    for i in range(30):
        image = np.zeros((256, 256), np.float32)
        cv2.circle(image, (128, 128), i, (255, 255, 255), -1, 8, 0)
        gauss_d1 = cv2.GaussianBlur(image, (15, 15), 0)
        gauss_d2 = cv2.GaussianBlur(image, (31, 31), 0)
        ganglionic = cv2.subtract(gauss_d1, gauss_d2)

        if filter is not None:
            ganglionic = convolve2d(ganglionic, filter)

        if is_complex:
            v = int(ganglionic.max())
        else:
            v = int(ganglionic[128][128])

        chart.append(v)
        cv2.line(res, (i * 8, 128 - v_prev // 5), ((i + 1) * 8, 128 - v // 5), (25
        v_prev = v

    return chart


def ring(filter=None, is_complex=False):
    chart = []
```

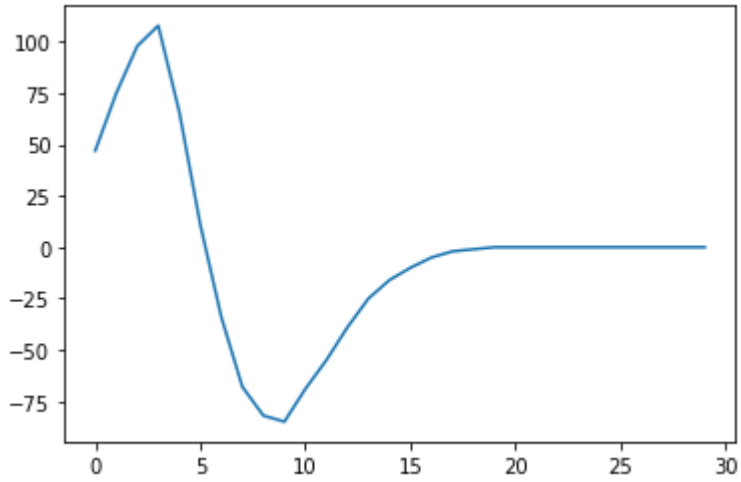
```
        res = np.zeros((256, 256), np.float32)
        v_prev = 0
        for i in range(30):
            image = np.zeros((256, 256), np.float32)
            cv2.circle(image, (128, 128), i, (255, 255, 255), 3, 8, 0)
            gauss_d1 = cv2.GaussianBlur(image, (15, 15), 0)
            gauss_d2 = cv2.GaussianBlur(image, (31, 31), 0)
            ganglionic = cv2.subtract(gauss_d1, gauss_d2)


            if filter is not None:
                ganglionic = convolve2d(ganglionic, filter)

            if is_complex:
                v = int(ganglionic.max())
            else:
                v = int(ganglionic[128][128])

            chart.append(v)
            cv2.line(res, (i * 8, 128 - v_prev // 5), ((i + 1) * 8, 128 - v // 5), (255
            v_prev = v

        return chart


    def line(filter=None, is_complex=False):
      chart = []
      res = np.zeros((256, 256), np.float32)
      v_prev = 0
      for i in range(180):
        image = np.zeros((256, 256), np.float32)
        cv2.line(image, (128 + int(100 * np.cos(i / 57.2)), 128 + int(100 * np.sin(i /
                        (128 - int(100 * np.cos(i / 57.2)), 128 - int(100 * np.sin(i /
        gauss_d1 = cv2.GaussianBlur(image, (15, 15), 0)
        gauss_d2 = cv2.GaussianBlur(image, (31, 31), 0)
        ganglionic = cv2.subtract(gauss_d1, gauss_d2)


        if filter is not None:
          ganglionic = convolve2d(ganglionic, filter)

        if is_complex:
            v = int(ganglionic.max())
        else:
            v = int(ganglionic[128][128])

        chart.append(v)
        cv2.line(res, (i * 8, 128 - v_prev // 5), ((i + 1) * 8, 128 - v // 5), (255, 25
        v_prev = v

      return chart


    chart = circle()
    plt.plot(chart)
```

```
[<matplotlib.lines.Line2D at 0x7f71af7e3b38>]
```



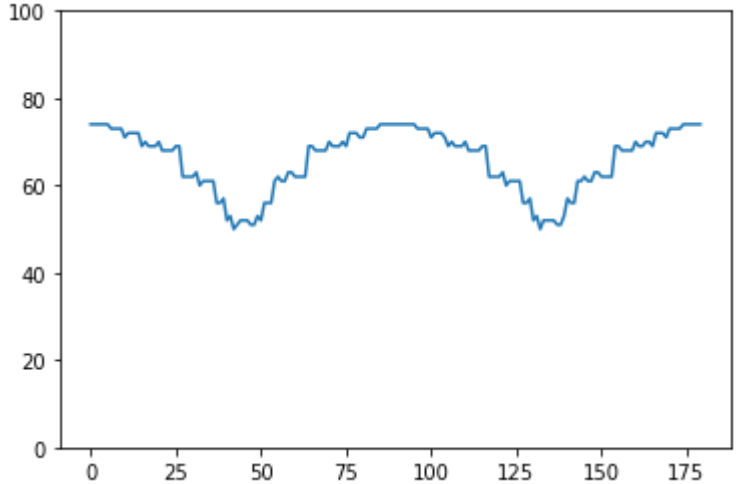

```
chart = ring()
plt.plot(chart)
```

```
[<matplotlib.lines.Line2D at 0x7f71af74b320>]
```




```
chart = line()
plt.plot(chart)
plt.ylim(0, 100)
```

```
(0.0, 100.0)
```



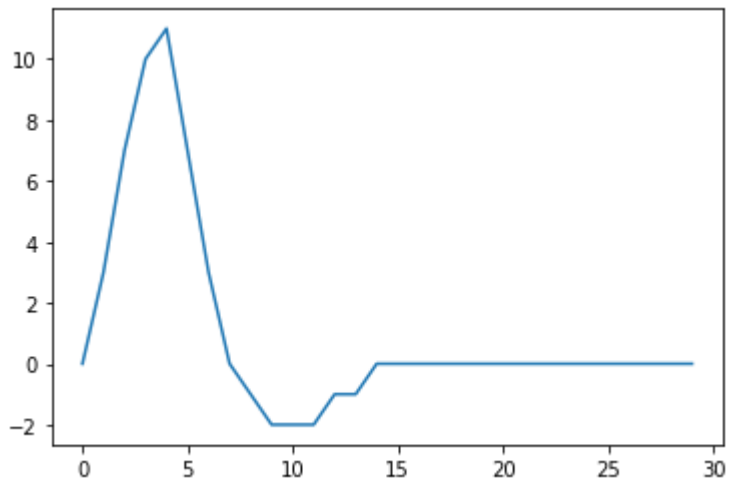

```
filter1 = [[1, -1], [1, -1]]  # фильтр границ
```

```
filter1 = [[1, -1], [1, -1]]  # фильтр границ
filter2 = [[1, -2, 1],[1, -2, 1],[1, -2, 1]]  # фильтр черной линии на белом фоне
filter3 = [[-1, 2, -1],[-1, 2, -1],[-1, 2, -1]]  # фильтр белой линии на черном фоне
```

Реакция простых клеток:

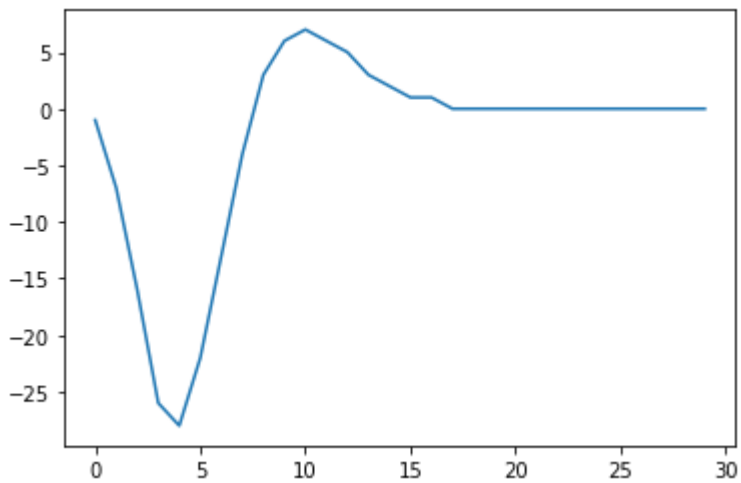
```
chart = circle(filter1)
plt.plot(chart)
```

```
[<matplotlib.lines.Line2D at 0x7f71af687fd0>]
```




```
chart = circle(filter2)
plt.plot(chart)
```

```
[<matplotlib.lines.Line2D at 0x7f71af6706a0>]
```

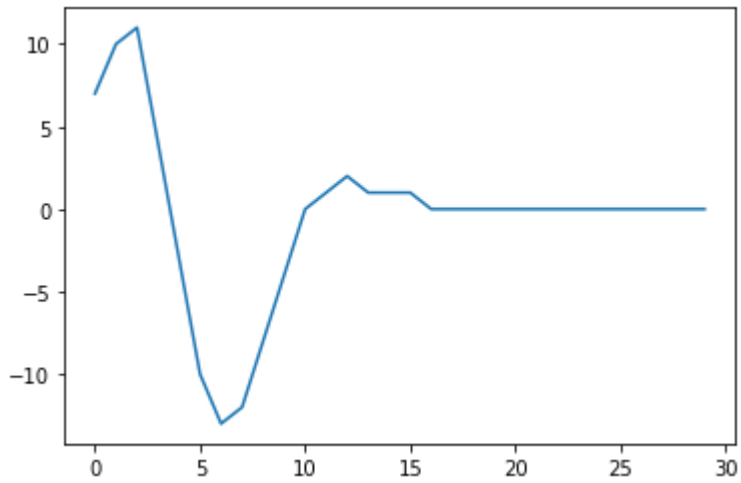



```
chart = circle(filter3)
plt.plot(chart)
```

```
[<matplotlib.lines.Line2D at 0x7f71b07095f8>]
```

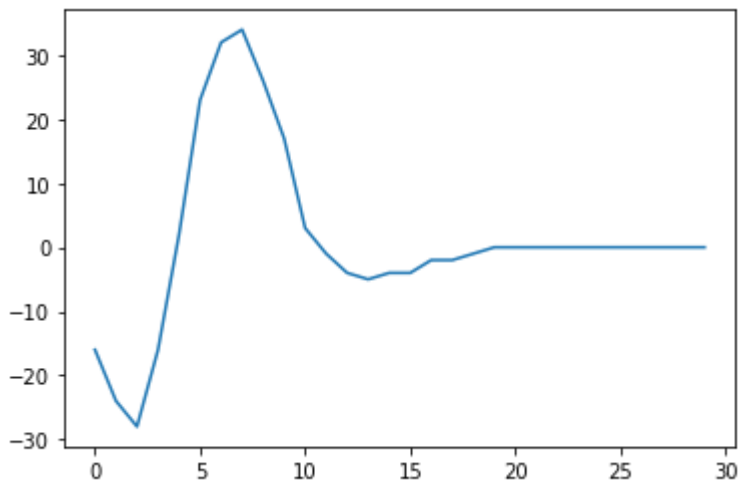

```
chart = ring(filter1)
plt.plot(chart)
```

```
[<matplotlib.lines.Line2D at 0x7f71aff1e390>]
```



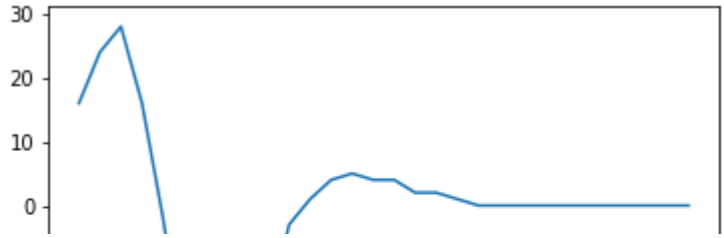

```
chart = ring(filter2)
plt.plot(chart)
```

```
[<matplotlib.lines.Line2D at 0x7f71af70d470>]
```



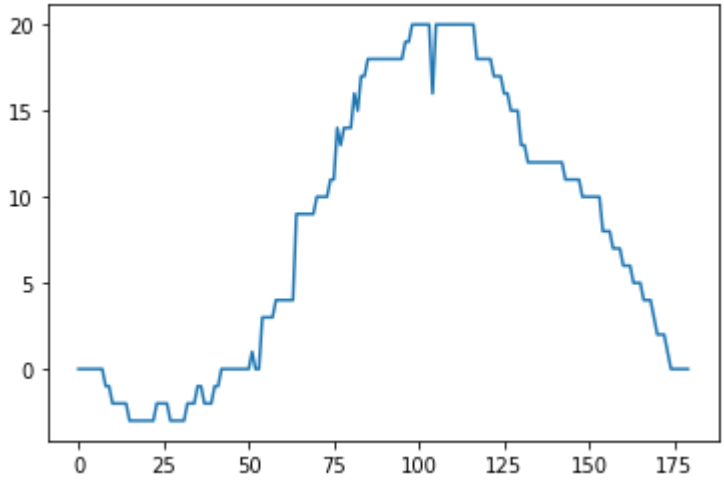

```
chart = ring(filter3)
plt.plot(chart)
```

```
[<matplotlib.lines.Line2D at 0x7f71af98e7f0>]
```



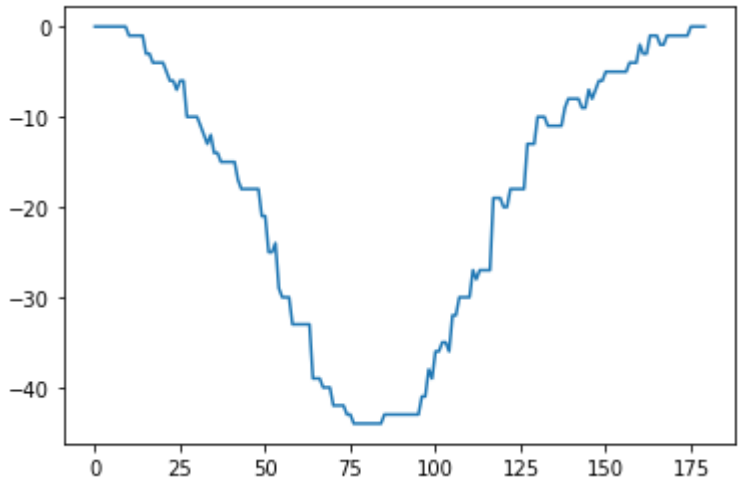

```
chart = line(filter1)
plt.plot(chart)
```

```
[<matplotlib.lines.Line2D at 0x7f71af07fbe0>]
```



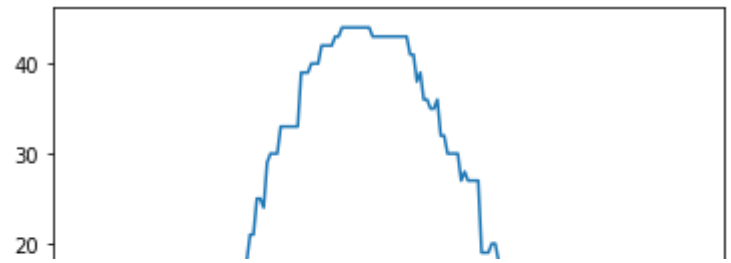

```
chart = line(filter2)
plt.plot(chart)
```

```
[<matplotlib.lines.Line2D at 0x7f71af511320>]
```




```
chart = line(filter3)
plt.plot(chart)
```

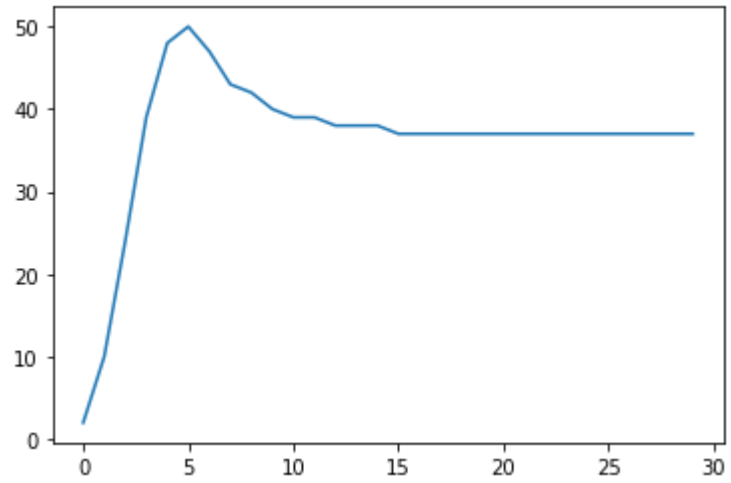
```
[<matplotlib.lines.Line2D at 0x7f71af4eea90>]
```




Реакция сложных клеток

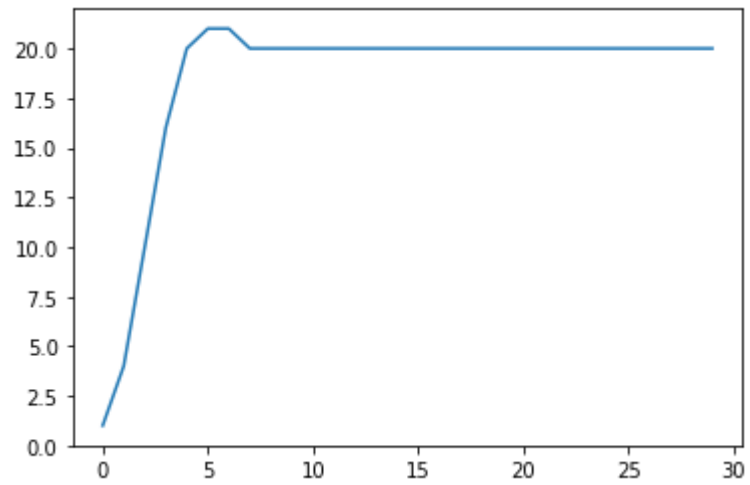
```
chart = circle(filter1, is_complex=True)
plt.plot(chart)
```

```
[<matplotlib.lines.Line2D at 0x7f71af454390>]
```



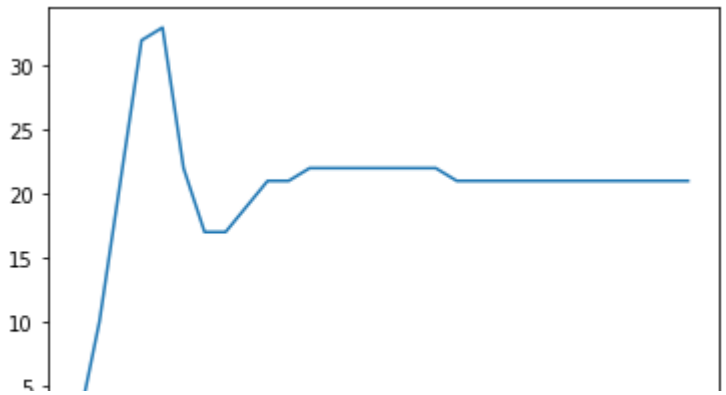

```
chart = circle(filter2, is_complex=True)
plt.plot(chart)
```

```
[<matplotlib.lines.Line2D at 0x7f71af432ba8>]
```



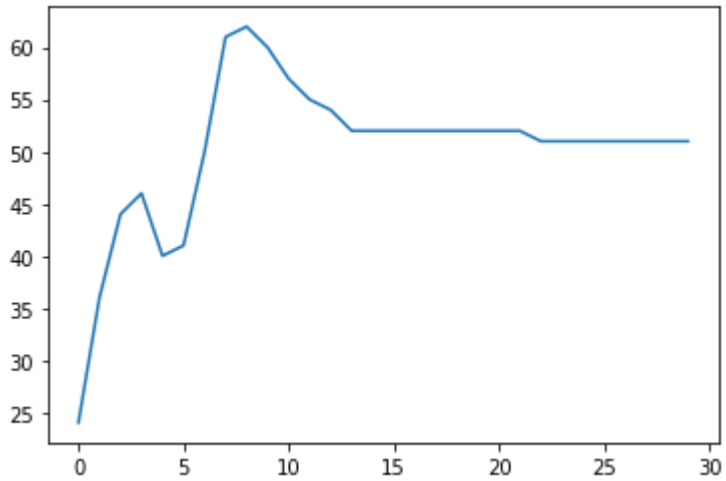

```
chart = circle(filter3, is_complex=True)
plt.plot(chart)
```

```
[<matplotlib.lines.Line2D at 0x7f71af39f438>]
```



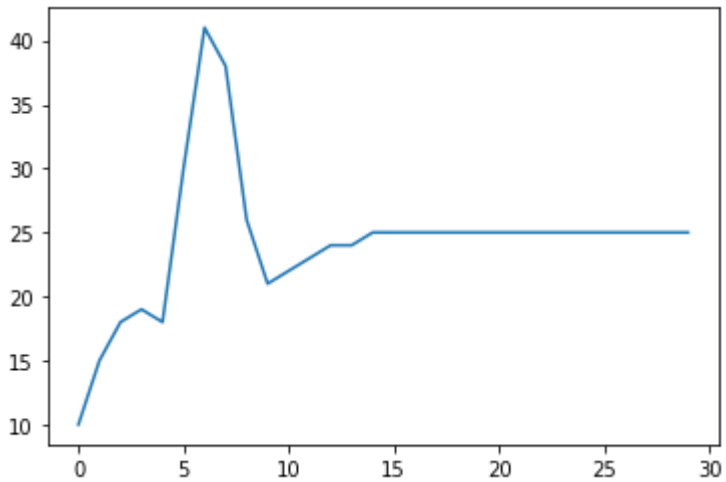

```
chart = ring(filter1, is_complex=True)
plt.plot(chart)
```

```
[<matplotlib.lines.Line2D at 0x7f71af379c50>]
```



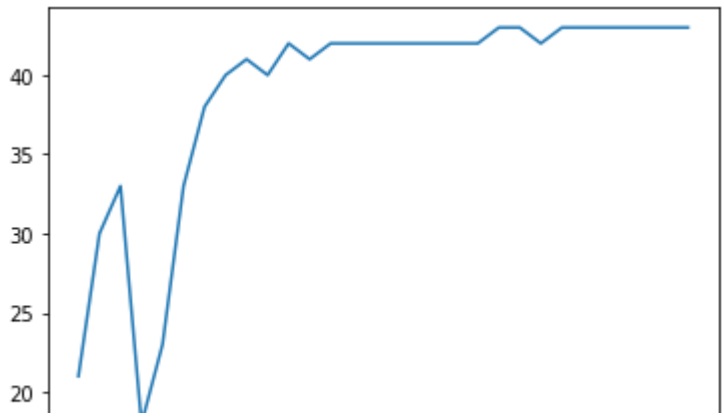

```
chart = ring(filter2, is_complex=True)
plt.plot(chart)
```

```
[<matplotlib.lines.Line2D at 0x7f71af2eb208>]
```



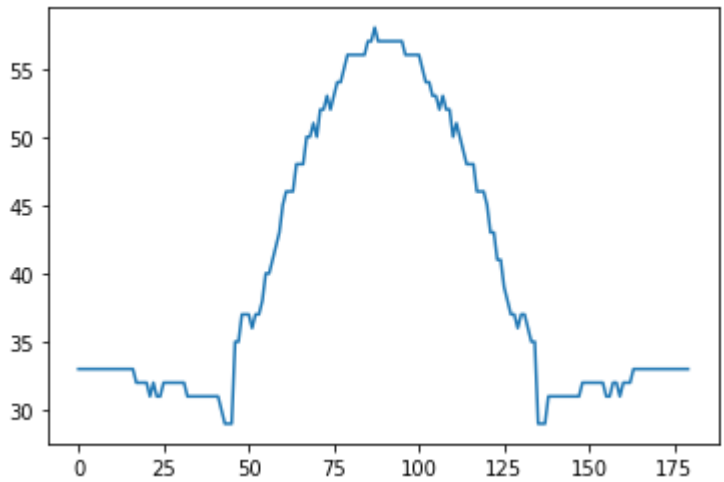

```
chart = ring(filter3, is_complex=True)
plt.plot(chart)
```

```
[<matplotlib.lines.Line2D at 0x7f71af249908>]
```



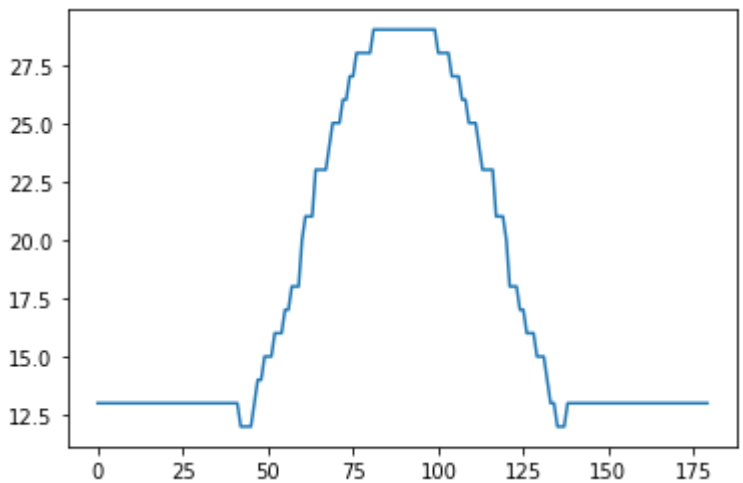

```
chart = line(filter1, is_complex=True)
plt.plot(chart)
```

```
[<matplotlib.lines.Line2D at 0x7f71af229710>]
```




```
chart = line(filter2, is_complex=True)
plt.plot(chart)
```

```
[<matplotlib.lines.Line2D at 0x7f71af187f60>]
```




```
chart = line(filter3, is_complex=True)
plt.plot(chart)
```

```
[<matplotlib.lines.Line2D at 0x7f71af16ecf8>]
```