

# Лабораторная работа N°7.

## Автоассоциативные сети с узким горлом.

Выполнила: Михеева Кристина Олеговна

Группа М8О-407Б-20

Вариант: 15

## Цель работы

Целью работы является исследование свойств автоассоциативных сетей с узким горлом, алгоритмов обучения, а также применение сетей для выполнения линейного и нелинейного анализа главных компонент набора данных.

## Ход работы

### Задание 1

Использовать автоассоциативную сеть с узким горлом для отображения набора данных, выделяя первую главную компоненту данных.

```
!pip install pyrenn

Requirement already satisfied: pyrenn in
/usr/local/lib/python3.10/dist-packages (0.1)
Requirement already satisfied: numpy in
/usr/local/lib/python3.10/dist-packages (from pyrenn) (1.23.5)

import matplotlib.pyplot as plt
import pyrenn
from mpl_toolkits.mplot3d import Axes3D
import numpy as np
```

Формируем входные данные.

```
a = 0.5
b = 0.4
alpha = 0
x0 = 0.3
y0 = -0.1
h = 0.025
```

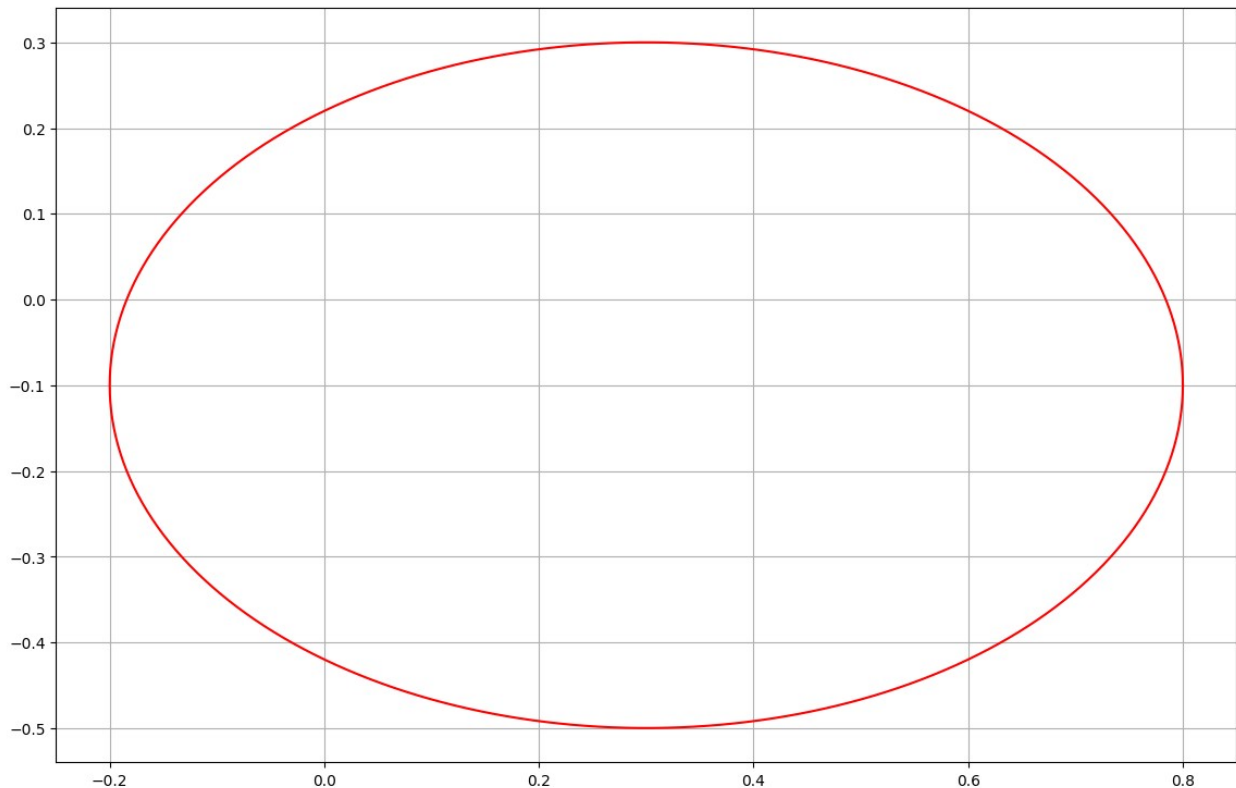
```

t = np.linspace(0, 2 * np.pi, int(2 * np.pi / h))
x1 = a * np.cos(t + alpha) + x0
y1 = b * np.sin(t) + y0

arr1 = np.array([x1, y1])

plt.figure(figsize=(14, 9))
plt.plot(arr1[0], arr1[1], c='r')
plt.grid()

```



Зададим нейронную сеть с 1 скрытым слоем и используем метод Левенберга-Марквардта в качестве алгоритма обучения.

```

nn = pyrenn.CreateNN([2, 1, 2])
nn = pyrenn.train_LM(arr1, arr1, nn, E_stop=1e-5, k_max=100)

```

Maximum number of iterations reached

Аппроксимируем функцию.

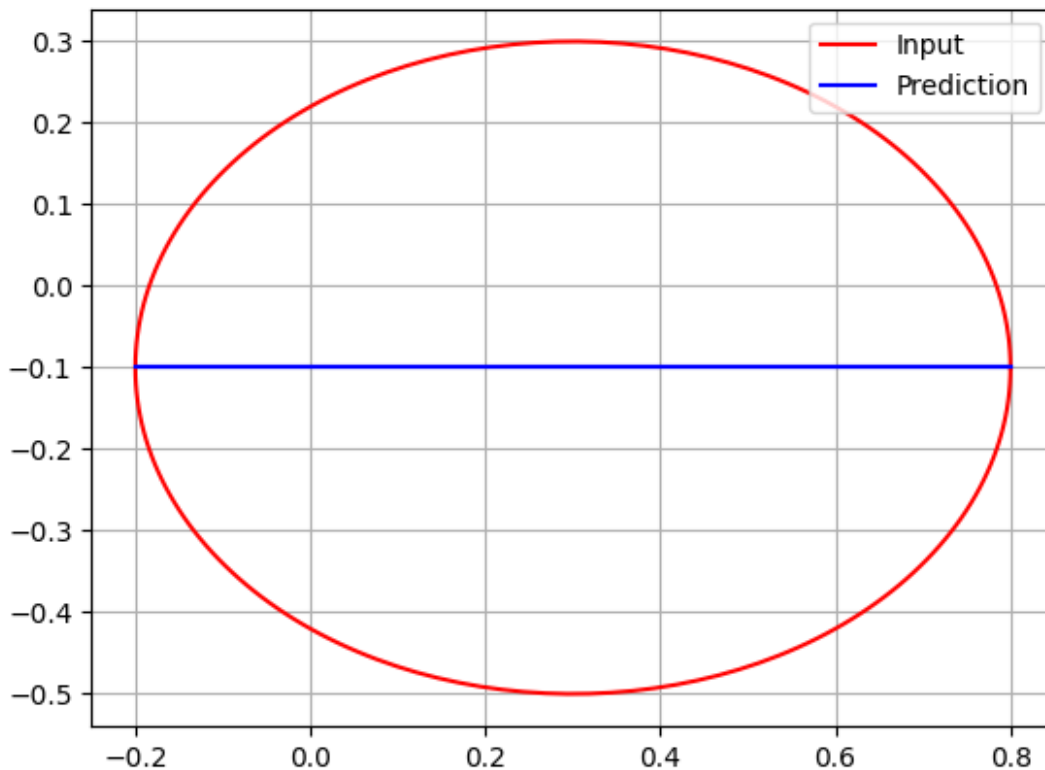
```

pred = pyrenn.NNOut(arr1, nn)

plt.plot(arr1[0], arr1[1], c='r', label = "Input")
plt.plot(pred[0], pred[1], c='b', label = "Prediction")
plt.grid()

```

```
plt.legend()  
plt.show()
```

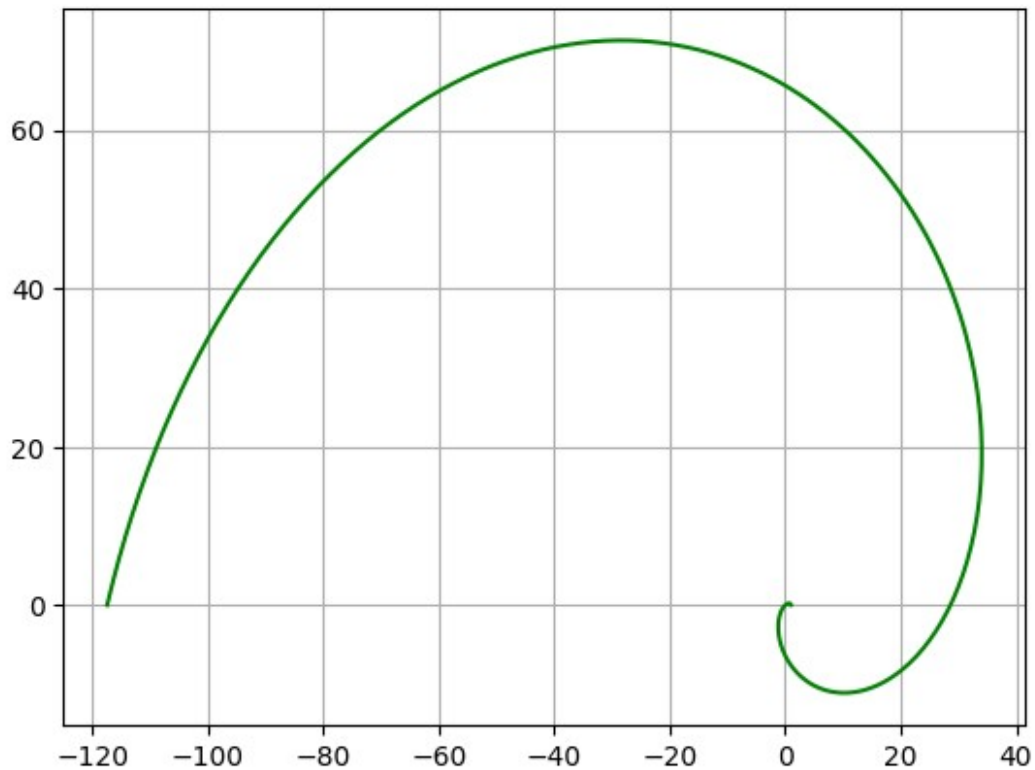


## Задание 2

Использовать автоассоциативную сеть с узким горлом для аппроксимации кривой на плоскости, выделяя первую нелинейную главную компоненту данных.

Сгенерируем набор из точек.

```
def get_func(t):  
    r = (-3 * (t**2)) + 1  
    return r * np.cos(t), r * np.sin(t)  
  
x2, y2 = get_func(t)  
arr2 = np.array([x2, y2])  
  
plt.plot(x2, y2, c='green')  
plt.grid()  
plt.show()
```

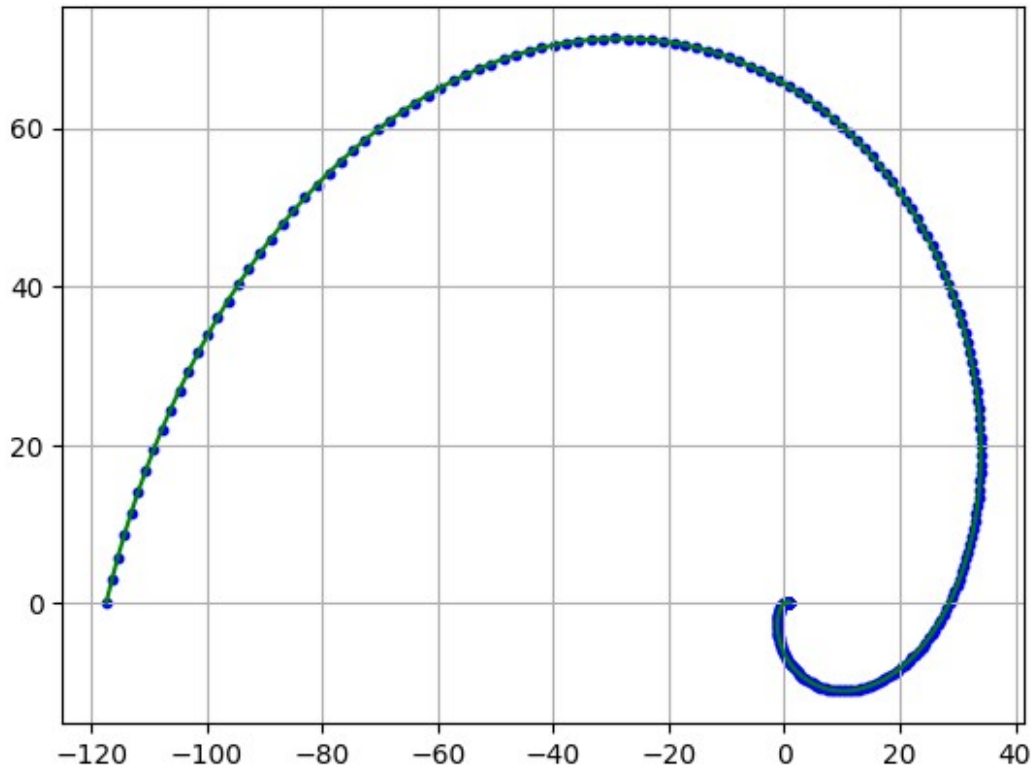


Обучим сеть.

```
nn2 = pyrenn.CreateNN([1, 10, 1, 10, 1])
nn2 = pyrenn.train_LM(arr2, arr2, nn2, E_stop=1e-5, k_max=2000)
Maximum number of iterations reached
```

Аппроксимируем функцию.

```
out2 = pyrenn.NNOut(arr2, nn2)
plt.plot(x2, y2, c='green')
plt.scatter(out2[0], out2[1], c='b', s=10)
plt.grid()
```



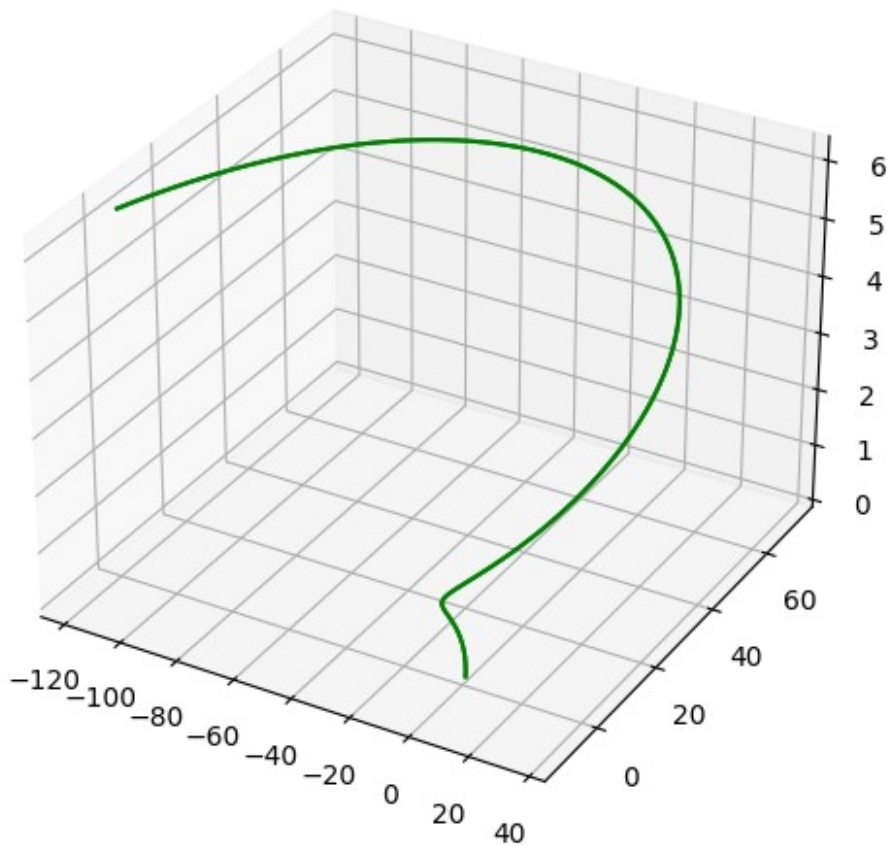
## Задание 3

Применить автоассоциативную сеть с узким горлом для аппроксимации пространственной кривой, выделяя старшие нелинейные главные компоненты данных.

Сгенерируем набор из точек.

```
phi = np.linspace(0, 2 * np.pi, int(2 * np.pi / h))
z = phi
arr3 = np.array([x2, y2, z])

fig = plt.figure(figsize=(8, 6))
ax = fig.add_subplot(111, projection='3d')
ax.plot_wireframe(x2, y2, np.array([z, z]), color='g')
plt.show()
```



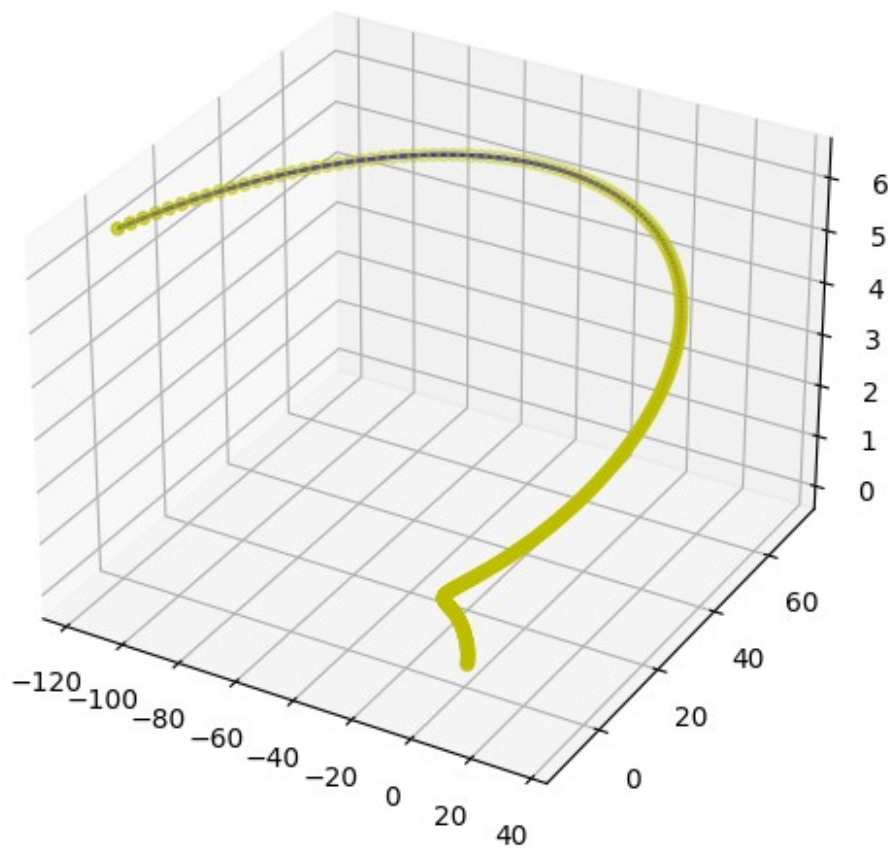
Создадим и обучим сеть.

```
nn3 = pyrenn.CreateNN([2, 10, 2, 10, 2])
nn3 = pyrenn.train_LM(arr3, arr3, nn3, E_stop=1e-5, k_max=1000)
```

Аппроксимируем функцию.

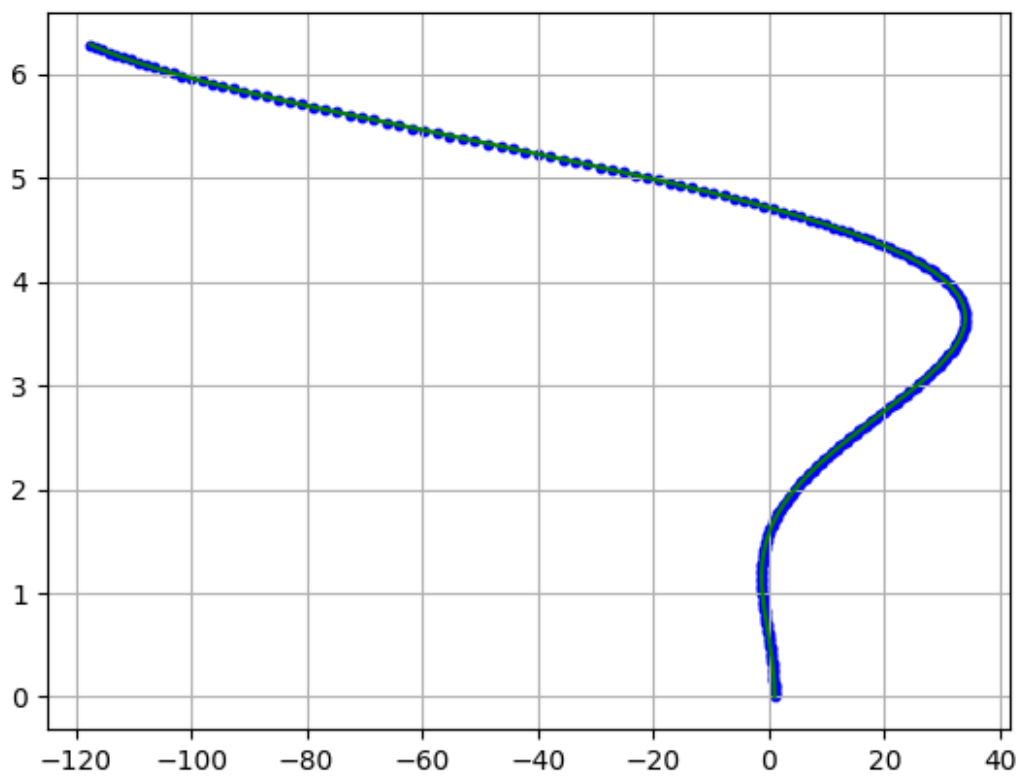
```
out3 = pyrenn.NNOut(arr3, nn3)

fig = plt.figure(figsize=(8,6))
ax = fig.add_subplot(111, projection='3d')
ax.scatter(out3[0], out3[1], out3[2], color='y')
ax.plot_wireframe(x2, y2, np.array([z, z]), color='b')
plt.show()
```



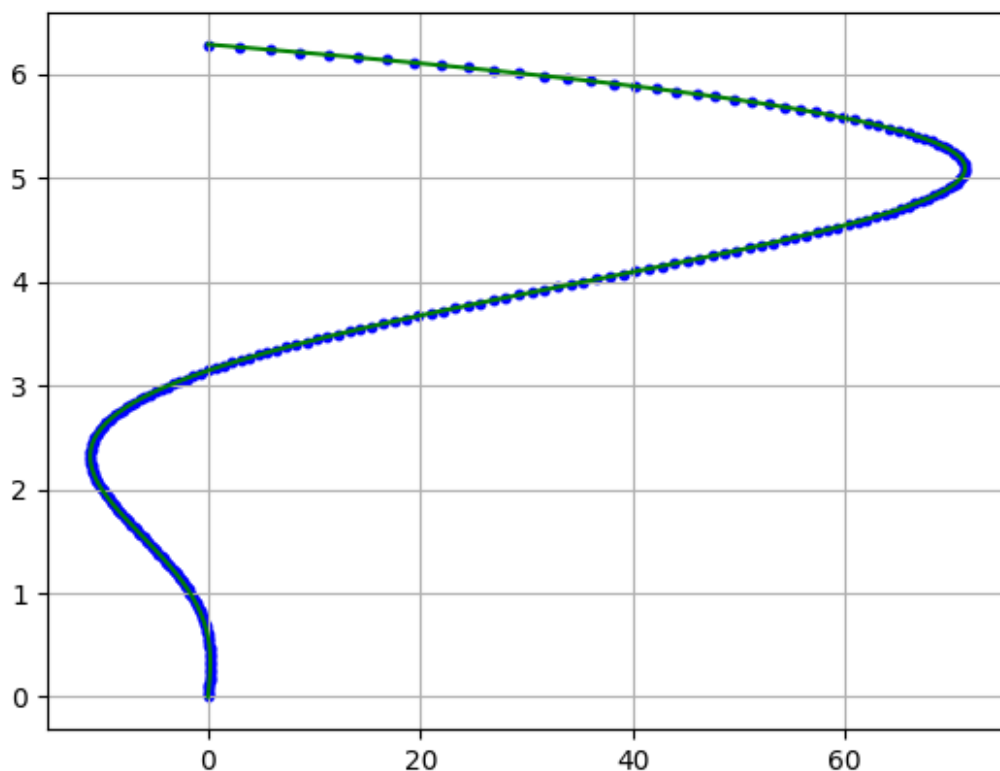
Отобразим обучающее множество и выход сети.

```
plt.plot(x2,z, c='g')  
plt.scatter(x2,out3[2], c='b', s = 10)  
plt.grid()
```



```
plt.plot(y2, z, c='g')  
plt.scatter(y2, out3[2], c='b', s=10)  
plt.grid()
```





## Вывод

В данной лабораторной работе я познакомилась с исследованием свойств автоассоциативных сетей с узким горлом.

По сути это такие нейронные сети, которые используются для решения задачи восстановления входных данных, что может быть полезным в различных областях, например как: колоризация, шумоподавление, восстановление изображений, повышение качества и многое другое.