Московский авиационный институт (национальный исследовательский университет)

Факультет информационных технологий и прикладной математики

Кафедра вычислительной математики и программирования

Лабораторная работа №7 по курсу «Нейроинформатика»

Студент: М. А. Бронников
Преподаватель: Н. П. Аносова
Группа: М8О-407Б

Дата:
Оценка:
Подпись:

Москва, 2021

Динамические сети

Целью работы является исследование свойств некоторых динамических нейронных сетей, алгоритмов обучения, а также применение сетей в задачах аппроксимации функций и распознавания динамических образов.

Основные этапы работы:

1. Использовать сеть прямого распространения с запаздыванием для предсказания значений временного ряда и выполнения многошагового прогноза.

2. Использовать сеть сеть прямого распространения с распределенным запаздыванием для распознавания динамических образов.

3. Использовать нелинейную авторегрессионную сеть с внешними входами для аппроксимации траектории динамической системы и выполнения многошагового прогноза.

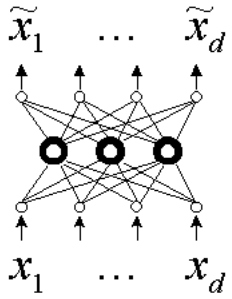
Вариант №4

1. Эллипс: $a = 0.6,\ b = 0.6,\ \alpha = 0,\ x_0 = 0,\ y_0 = 0$
2. $r = e^{\phi}$

1 Структура модели

Автоассоцитивная сеть с узким горлом

Автоассоцитивная сеть с узким горлом имеет 1 скрытый слой, выходы которого - главные компоненты входного вектора. Веса же выходного слоя настроены на то, чтобы востановить исходные данные по главной компоненте.



Такие сети позволяют сильно сжимать размерность получаемых данных и востонавливать исходные данные с минимальными потерями дисперсии(информации).

2 Ход работы

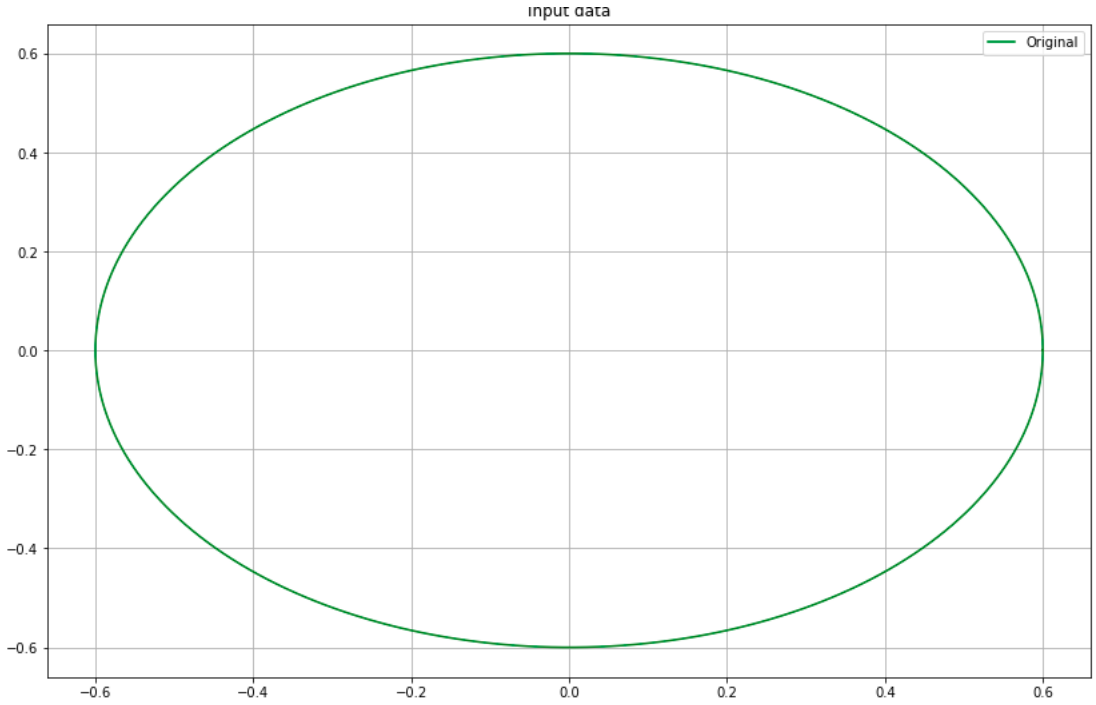
Формируем входные данные:

```
t = np.linspace(0, 2 * np.pi, int(2 * np.pi / 0.025), endpoint=True)

x = f(t)
y = g(t)

orig = np.array([x, y])
```

```
plt.figure(figsize=(14, 9))
plt.plot(orig[0], orig[1], 'green', label="Original")
plt.grid(True)
plt.title('Input data')
plt.legend()
plt.show()
```



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

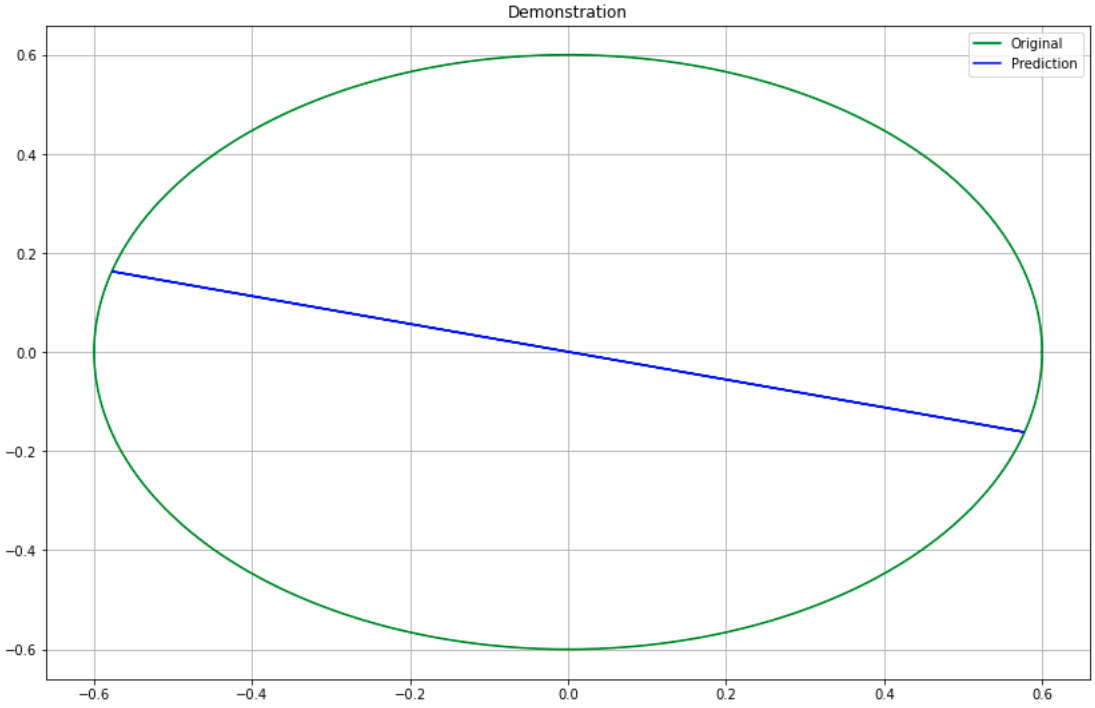
```
nn = pyrenn.CreateNN([2, 1, 2])
nn = pyrenn.train_LM(orig, orig, nn, E_stop=1e-5, k_max=200)
```

```
Maximum number of iterations reached
```

Востановленные данные:

```
pred = pyrenn.NNOut(orig, nn)
```

```
plt.figure(figsize=(14, 9))
plt.plot(orig[0], orig[1], 'green', label="Original")
plt.plot(pred[0], pred[1], 'blue', label="Prediction")
plt.grid(True)
plt.title('Demonstration')
plt.legend()
plt.show()
```



Задание №2

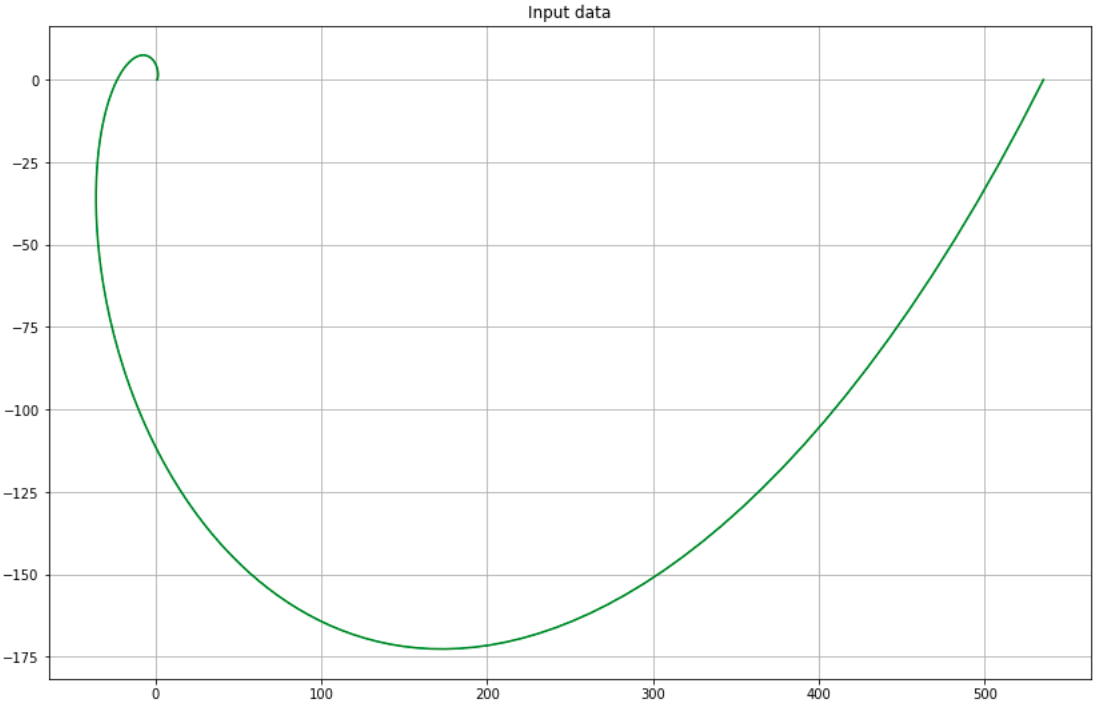
Использовать *автоассоциативную сеть с узким горлом* для аппроксимации кривой на плоскости, выделяя первую нелинейную главную компоненту данных.

Сгенерируем набор из точек:

```
: phi = np.linspace(0, 2 * np.pi, int(2 * np.pi / 0.025), endpoint=True)

x2, y2 = gen_x_y(phi)

plt.figure(figsize=(14, 9))
plt.plot(x2, y2, 'green')
plt.grid(True)
plt.title('Input data')
plt.show()
```



Обучим сеть:

```
nn2 = pyrenn.CreateNN([1, 10, 1, 10, 1])
```

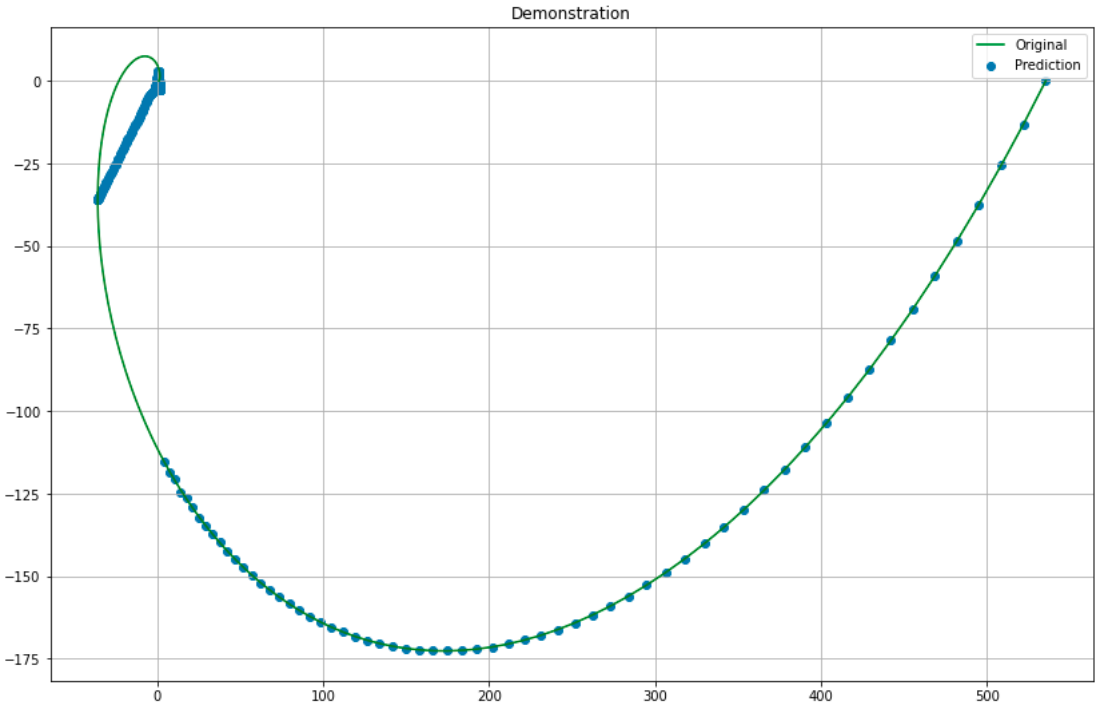
```
nn2 = pyrenn.train_LM(x2, y2, nn2, E_stop=1e-5, k_max=2000)
```

```
Maximum number of iterations reached
```

Аппроксимируем функцию:

```
a2 = pyrenn.NNOut(x2, nn2)
```

```
plt.figure(figsize=(14, 9))
plt.plot(x2, y2, 'green', label='Original')
plt.scatter(x2, a2, label='Prediction')
plt.grid(True)
plt.title('Demonstration')
plt.legend()
plt.show()
```



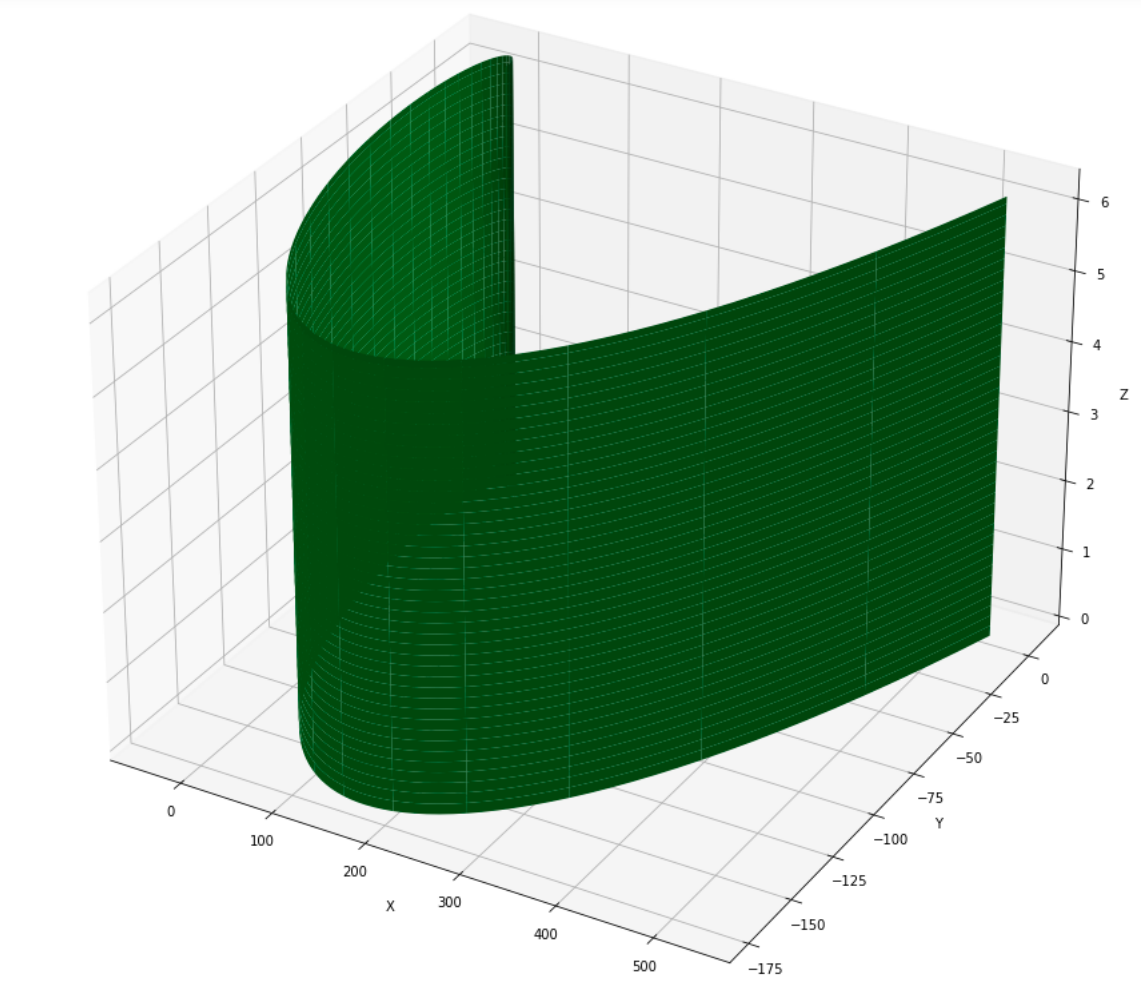
Задание №3

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 / 0.025), endpoint=True)
x3, y3 = gen_x_y(phi)
z3 = phi


fig = plt.figure(num=1, figsize=(19, 12), clear=True)
ax = fig.add_subplot(1, 1, 1, projection='3d')
ax.plot_surface(x3, y3, z3.reshape(-1, 1), color='green')
ax.set_title('Input Data')
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
fig.tight_layout()
```



Создадим и обучим сеть:

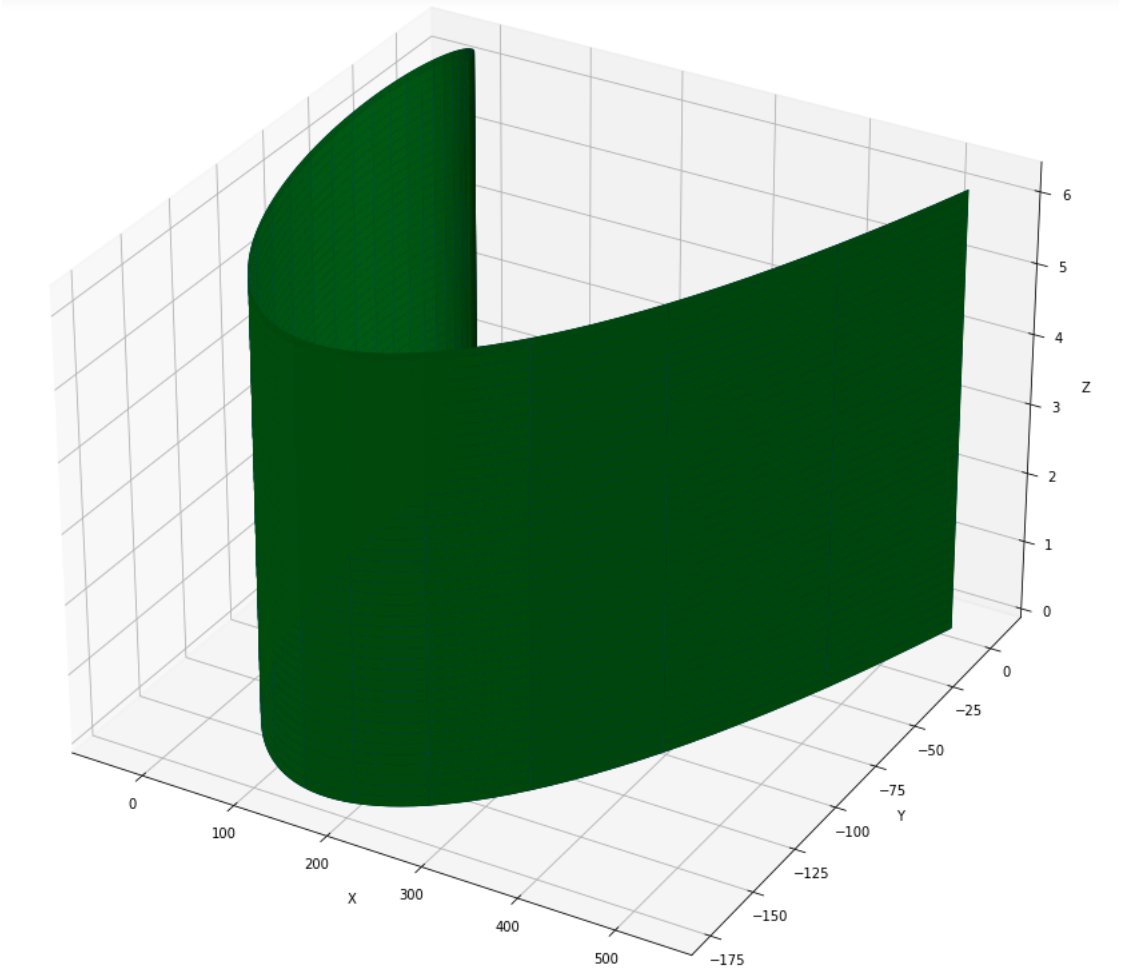
```
nn3 = pyrenn.CreateNN([2, 10, 2,  10, 1])
nn3 = pyrenn.train_LM(np.array([x3, z3]), y3, nn3, E_stop=1e-5, k_max=500)
```

```
Termination Error reached
```

Аппроксимируем функцию:

```
a3 = pyrenn.NNOut(np.array([x3, z3]), nn3)
```

```
fig = plt.figure(num=1, figsize=(19, 12), clear=True)
ax = fig.add_subplot(1, 1, 1, projection='3d')
ax.plot_surface(x3, y3, z3.reshape(-1, 1), color='green')
ax.plot_surface(x3, a3, z3.reshape(-1, 1), color='blue')
ax.set_title('Demonstration')
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
fig.tight_layout()
```



3 Выводы

Выполнив седьмую лабораторную работу по курсу «Нейроинформатика», я познакомился с автоассоциативные сети с узким горлом которые могут быть хорошей альтернативой методу главных компонент.

В основном эти сети используются только для того, чтобы наиболее компактно представить входные данные, поэтому обучаетсся сеть таким образом, чтобы кполучаемые компоненты имели как можно больше от исходной диспрсиии сети. Такое преобразование имеет не только свойство уменьшения количества признаков, что решает проблему проклятия размерности, но и позволяет удалить шумы, содержащиеся в компонентах с небольшой долей дисперсии.

Эта работа была была мне наиболее интересна, поскольку я интересуюсь методами сжатия и обработки информационных сигналов и мне было удивительно и приятно встретить представителя нейронных сетей, достойно справялющегося с этой проблемой.