УДК 004.02

Сенотова Светлана Анатольевна,
к.т.н., доцент, Ангарский государственный технический университет,
e-mail: sveta-senotova@mail.ru

СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ АППРОКСИМАЦИИ С ПОМОЩЬЮ РЕГРЕССИОННЫХ ЗАВИСИМОСТЕЙ И НЕЙРОННЫХ СЕТЕЙ ДЛЯ ЛИНЕЙНЫХ МОДЕЛЕЙ

Senotova S.A.

COMPARATIVE ANALYSIS OF APPROXIMATION METHODS USING REGRESSION RELATIONSHIPS AND NEURAL NETWORKS FOR LINEAR MODELS

Аннотация. В работе рассматривается сравнительный анализ методов аппроксимации с помощью регрессионных зависимостей и нейронных сетей для линейных моделей.

Ключевые слова: регрессионный анализ, метод наименьших квадратов, нейронные сети, Python, Keras.

Abstract. The paper examines comparative analysis of approximation methods using regression dependencies and neural networks for linear models.

Keywords: regression analysis, least squares method, neural networks, Python, Keras.

Регрессионный анализ – это группа методов, направленных на выявление и математическое выражение тех изменений и зависимостей, которые имеют место в системе случайных величин.

Регрессия – функция, позволяющая по величине одного признака определить величину другого признака.

Выделим основные этапы регрессионного анализа:

1) выбор вида модели (построение гипотезы о виде функции);
2) численная оценка параметров модели;
3) проверка адекватности полученной модели.

Построение гипотезы о виде функции выполняется исследователем исходя из его представлений о взаимосвязи между входными и выходными данными. Функция, кроме переменных, зависит от некоторого числа параметров. Необходимо наилучшим образом определить значения этих параметров. Метод наименьших квадратов основан на предположении, что наилучшими будут те значения параметров, при которых сумма квадратов отклонений расчетных величин от экспериментальных окажется наименьшей.

Нейронные сети – это самообучающиеся системы, имитирующие работу человеческого мозга. Нейронные сети являются мощным инструментом анализа и аппроксимации зависимостей и широко используются в различных областях науки и техники. Широкое использование нейросетевого подхода обусловлено следующими причинами:

1) нейронная сеть способна обучаться на примерах в случаях, когда неизвестен вид и структура взаимосвязей между входными и выходными данными;

2) с помощью нейронных сетей можно получать приемлемое решение даже в случае неполной, искаженной и зашумленной информации;

3) нейронная сеть позволяет быстро обработать огромный объем информации и выявить в наблюдаемых данных скрытые закономерности;

4) с помощью нейронной сети можно аппроксимировать любую нелинейную зависимость.

Нейрон с одним входом – это простой элемент, который умножает скалярный вход x на скалярный вес w_1, формируя произведение $w_1 x$. Затем полученная величина складывается с некоторой пороговой величиной b_1, которая называется смещением (bias). Скалярный выход нейрона формируется как некоторая функция его состояния:

$$y = f(w_1 x + b_1).$$

Функция f называется функцией активации. Вид функции активации выбирается разработчиком сети, в зависимости от особенностей конкретной задачи и назначения сети, а параметры нейрона w_1 и b_1 определяются в процессе обучения сети в соответствии с поставленной целью.

Рассмотрим пример аппроксимации функции с помощью регрессионной зависимости и нейронной сети.

Определим методом наименьших квадратов параметры a и b линейной зависимости. Вычислим параметры уравнения градуировочной прямой по следующим данным оптической плотности стандартных растворов бензола в этаноле (таблица 1).

Таблица 1

Оптическая плотность стандартных растворов бензола в этаноле

Концентрация бензола, г/л	0,50	1,00	1,50	2,00	2,50
Оптическая плотность	0,37	0,64	0,93	1,22	1,55

Сначала вычисляем величины, входящие в следующие уравнения:

$$a = \frac{N\sum_{i=1}^{N} x_i y_i - \sum_{i=1}^{N} x_i \sum_{i=1}^{N} y_i}{N\sum_{i=1}^{N} x_i^2 - \left(\sum_{i=1}^{N} x_i\right)^2}; \qquad b = \frac{\sum_{i=1}^{N} y_i - a\sum_{i=1}^{N} x_i}{N},$$

$$\sum_{i=1}^{5} x_i = 7.5; \quad \sum_{i=1}^{5} x_i^2 = 13.75; \quad \sum_{i=1}^{5} y_i = 4.71; \quad \sum_{i=1}^{5} x_i y_i = 8.535,$$

затем рассчитываем параметры a и b:

$$\mathrm{a} = \frac{5 \cdot 8.535 - 7.5 \cdot 4.71}{5 \cdot 13.75 - (7.5)^2} = 0.588; \qquad b = \frac{4.71 - 0.588 \cdot 7.5}{5} = 0.06.$$

Градуировочная прямая выражается уравнением:

$$y = 0.588 \cdot x + 0.06 . \quad (1)$$

Графики реальной зависимости (рисунок 1) и зависимости, полученной по методу наименьших квадратов (рисунок 2), совпадают.

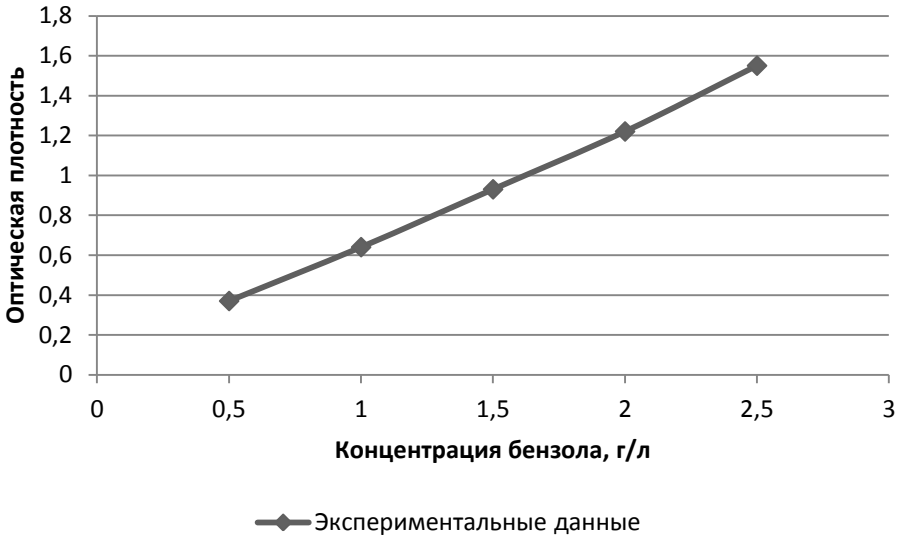

Рисунок 1 – Реальная зависимость

Рисунок 2 – Зависимость, полученная по методу наименьших квадратов

Для проверки модели на адекватность реальные значения оптической плотности сравниваются со значениями, полученными по формуле (1).

Среднее значение ошибки рассчитывается по формуле

$$e = \frac{1}{N} \cdot \sum_{t=1}^{N} e_t .$$

Дисперсия рассчитывается по формуле

$$\sigma^2 = \frac{1}{N} \cdot \sum_{t=1}^{N} (e_t - e)^2.$$

Результаты расчетов, полученные по формуле (1), представлены в таблице 2.

Таблица 2

Результаты расчетов, полученные по формуле (1), и реальные значения

Исходные данные		Результат	Ошибка
Концентрация бензола, г/л	Оптическая плотность	Оптическая плотность, полученная по формуле (1)	
0,50	0,37	0,354	0,016
1,00	0,64	0,648	-0,08
1,50	0,93	0,942	-0,012
2,00	1,22	1,236	-0,016
2,50	1,55	1,53	0,02

Затем вычисляются: математическое ожидание ошибки: m = 4,44 · 10^{-17}; дисперсия: σ^2 = 0,00112; среднее квадратическое отклонение: σ = 0,0334664.

Проверка показывает, что ошибка во всех 5 случаях не отклоняется от математического ожидания по абсолютной величине больше, чем на 3σ. Это дает основание считать модель адекватной реальным данным.

Применение нейросетевого подхода с использованием библиотеки Keras дает следующие результаты:

[array([[0.5885246]], dtype=float32), array([0.06005999], dtype=float32)],

т.е.

$$a = 0.5885246, \quad b = 0.06005999.$$

Графики реальной и найденной зависимостей при числе эпох, равном 100, также практически совпадают (рисунок 2). График ошибки имеет вид, представленный на рисунке 3. Программа написана на языке программирования Python.

```
import tensorflow as tf
from tensorflow import keras
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.layers import Dense
c = np.array([0.5,1,1.5,2,2.5])
f = np.array([0.37,0.64,0.93,1.22,1.55])
model = keras.Sequential()
model.add(Dense(units=1, input_shape=(1,), activation='linear'))
```

```
model.compile(loss='mean_squared_error', optimizer=keras.optimizers.Adam(0.1))
log = model.fit(c, f, epochs=100, verbose=False)
print(model.get_weights())
plt.plot(c, f)
plt.plot(c, model.predict(c))
plt.show()
```

В результате исследования можно сделать вывод, что на линейных моделях библиотека Keras дает результаты, которые совпадают с результатами метода наименьших квадратов.

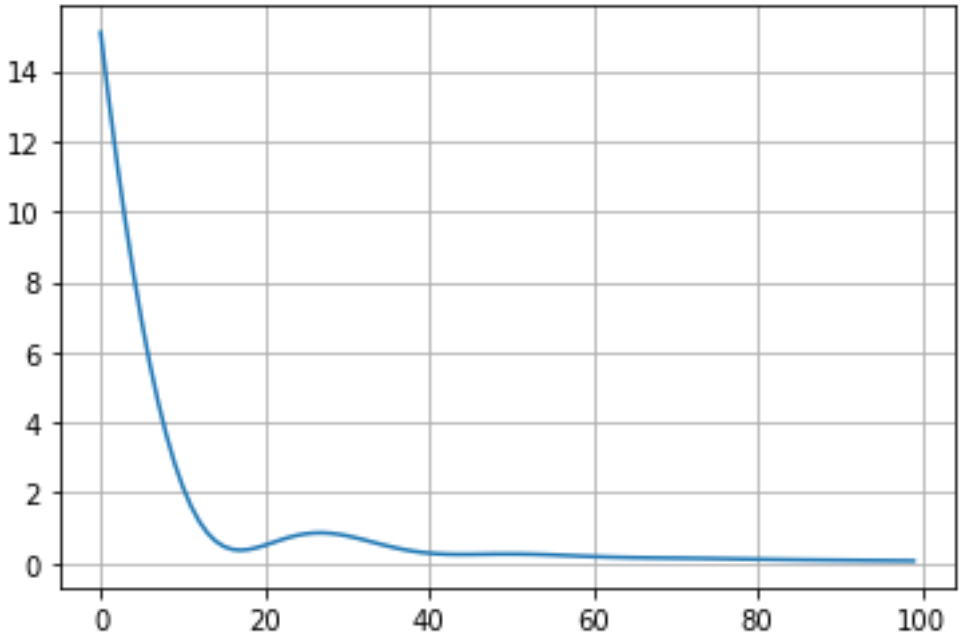

Рисунок 3 – График ошибки

ЛИТЕРАТУРА

1. Васильев, В.П. Аналитическая химия. Сборник вопросов упражнений и задач. / В.П. Васильев, Р.П. Морозова, Т.Д. Орлова; под ред. В.П. Васильева.- 2-е изд., перераб. и доп. – М.: Дрофа, 2003. – 320 с.