Доклад по дисциплине «Компьютерное моделирование прикладных задач»

Прогнозирование временных рядов с использованием RNN и дифференциальных уравнений

Студент: Вихляев Е.С., кафедра фундаментальной математики

Преподаватель: Митин В.Ю., кафедра фундаментальной математики

Abstract—В данном докладе рассматривается задача прогнозирования временных рядов с использованием системы дифференциальных уравнений и модели машинного обучения LSTM (Long Short-Term Memory). В рамках исследования формулируется задача в виде системы линейных дифференциальных уравнений первого порядка. Для численного решения используется метод Рунге-Кутты, который позволяет получить точные значения временных рядов на заданном интервале времени. Полученные данные используются для обучения модели LSTM, которая предсказывает будущие значения временного ряда. Результаты моделирования и прогнозирования визуализируются и сравниваются, демонстрируя эффективность предложенного подхода.

Keywords—Прогнозирование временных рядов, дифференциальные уравнения, метод Рунге-Кутты, машинное обучение, LSTM, численное решение, математическое моделирование, визуализация данных

Contents

1. Введение

Прогнозирование временных рядов является одной из ключевых задач в различных областях, таких как экономика, метеорология, финансы и инженерия. Временные ряды представляют собой последовательности данных, собранных через равные интервалы времени. Прогнозирование временных рядов позволяет предсказывать будущие значения на основе предыдущих наблюдений, что может быть критически важно для принятия решений и планирования.

Одной из методологий для моделирования временных рядов являются дифференциальные уравнения, которые позволяют описывать динамику систем с учетом временных изменений. В то же время, современные методы машинного обучения, такие как рекуррентные нейронные сети (RNN) и их усовершенствованный вариант — Long Short-Term Memory (LSTM), демонстрируют высокую точность в задачах прогнозирования временных рядов.

Целью данного исследования является построение и анализ математической модели, которая сочетает в себе классические методы решения дифференциальных уравнений и современные алгоритмы машинного обучения для прогнозирования временных рядов. Задачи исследования включают:

1. Формулировка задачи прогнозирования временных рядов в виде системы дифференциальных уравнений.
2. Решение системы дифференциальных уравнений методом Рунге-Кутты.
3. Обучение модели LSTM для прогнозирования временных рядов.
4. Сравнение результатов классического численного решения и модели машинного обучения.
5. Визуализация результатов для наглядного представления точности и эффективности предложенного подхода.

Математическая модель, используемая в данном исследовании, представляет собой систему линейных дифференциальных уравнений первого порядка, которая описывает взаимосвязь между переменными во времени. Система дифференциальных уравнений имеет вид:

$$\frac{dy_1}{dt} = a_{11}y_1 + a_{12}y_2$$
$$\frac{dy_2}{dt} = a_{21}y_1 + a_{22}y_2$$

где y_1 и y_2 — функции времени, а a_{ij} — коэффициенты, определяющие динамику системы.

Для численного решения системы дифференциальных уравнений используется метод Рунге-Кутты четвертого порядка, который обеспечивает высокую точность при интегрировании уравнений во времени. Этот метод позволяет получить численные значения функций y_1 и y_2 на заданном интервале времени.

Для прогнозирования временных рядов используется модель LSTM, которая является одним из видов рекуррентных нейронных сетей, способных эффективно обрабатывать и запоминать длинные последовательности данных. Модель LSTM обучается на численных решениях системы дифференциальных уравнений и предсказывает будущие значения временного ряда.

2. Математическая постановка задачи

Прогнозирование временных рядов заключается в предсказании будущих значений на основе предыдущих наблюдений. Временные ряды представляют собой последовательности данных, упорядоченные по времени, и могут быть как одномерными, так и многомерными. Примеры включают финансовые данные (цены акций), экономические индикаторы (уровень безработицы), природные явления (температура, осадки) и многие другие.

В данном исследовании для моделирования временных рядов используется система линейных дифференциальных уравнений первого порядка. Линейные дифференциальные уравнения часто применяются для описания динамических систем, где переменные изменяются во времени. Пусть $y_1(t)$ и $y_2(t)$ - это функции времени, описывающие поведение системы. Тогда система дифференциальных уравнений имеет вид:

$$\frac{dy_1}{dt} = a_{11}y_1 + a_{12}y_2$$
$$\frac{dy_2}{dt} = a_{21}y_1 + a_{22}y_2$$

где a_{ij} - коэффициенты системы, которые определяют взаимосвязи между переменными y_1 и y_2.

Рассмотрим подробнее систему дифференциальных уравнений:

1. Дифференциальные уравнения первого порядка описывают скорость изменения переменной во времени. В нашем случае, $\frac{dy_1}{dt}$ и $\frac{dy_2}{dt}$ описывают скорость изменения y_1 и y_2 соответственно.
2. Коэффициенты a_{ij} определяют линейную зависимость между переменными. Например, коэффициент a_{11} показывает, как y_1 влияет на саму себя, а a_{12} показывает, как y_2 влияет на y_1.
3. Система уравнений может быть решена численно с использованием методов интегрирования, таких как метод Рунге-Кутты, что позволяет получить значения y_1 и y_2 в дискретные моменты времени.

Для численного решения системы дифференциальных уравнений применим метод Рунге-Кутты четвертого порядка. Этот метод обеспечивает высокую точность и устойчивость при решении обыкновенных дифференциальных уравнений. Численное решение будет выполняться на заданном интервале времени $[0, T]$ с шагом h.

Пусть начальные условия для системы заданы как $y_1(0) = y_{10}$ и $y_2(0) = y_{20}$. Тогда метод Рунге-Кутты четвертого порядка можно записать следующим образом:

Для каждого шага n:

$$k_{1,1} = hf_1(t_n, y_{1,n}, y_{2,n})$$
$$k_{1,2} = hf_2(t_n, y_{1,n}, y_{2,n})$$

$$k_{2,1} = hf_1(t_n + \frac{h}{2}, y_{1,n} + \frac{k_{1,1}}{2}, y_{2,n} + \frac{k_{1,2}}{2})$$
$$k_{2,2} = hf_2(t_n + \frac{h}{2}, y_{1,n} + \frac{k_{1,1}}{2}, y_{2,n} + \frac{k_{1,2}}{2})$$

$$k_{3,1} = hf_1(t_n + \frac{h}{2}, y_{1,n} + \frac{k_{2,1}}{2}, y_{2,n} + \frac{k_{2,2}}{2})$$
$$k_{3,2} = hf_2(t_n + \frac{h}{2}, y_{1,n} + \frac{k_{2,1}}{2}, y_{2,n} + \frac{k_{2,2}}{2})$$

$$k_{4,1} = hf_1(t_n + h, y_{1,n} + k_{3,1}, y_{2,n} + k_{3,2})$$
$$k_{4,2} = hf_2(t_n + h, y_{1,n} + k_{3,1}, y_{2,n} + k_{3,2})$$

Обновленные значения $y_{1,n+1}$ и $y_{2,n+1}$ вычисляются как:

$$y_{1,n+1} = y_{1,n} + \frac{1}{6}(k_{1,1} + 2k_{2,1} + 2k_{3,1} + k_{4,1})$$
$$y_{2,n+1} = y_{2,n} + \frac{1}{6}(k_{1,2} + 2k_{2,2} + 2k_{3,2} + k_{4,2})$$

3. Методы решения

Для численного решения системы линейных дифференциальных уравнений первого порядка используется метод Рунге-Кутты четвертого порядка (RK4). Этот метод обеспечивает высокую точность и устойчивость при решении обыкновенных дифференциальных уравнений.

Метод Рунге-Кутты четвертого порядка был выбран из-за его высокой точности и устойчивости при численном решении обыкновенных дифференциальных уравнений. Этот метод позволяет интегрировать систему уравнений на заданном интервале времени $[0, T]$ с шагом h.

Long Short-Term Memory (LSTM) - это тип рекуррентной нейронной сети (RNN), специально разработанный для обработки и предсказания данных временных рядов. LSTM эффективно решает проблему исчезающих и взрывающихся градиентов, что делает его идеальным для задач прогнозирования на длинных последовательностях.

Данные, полученные при численном решении системы дифференциальных уравнений, преобразуются в формат, подходящий для обучения модели LSTM.

Входные данные представляют собой временные ряды $y_1(t)$ и $y_2(t)$.

Первый слой LSTM: 50 нейронов, возвращает последовательности для следующего слоя.

Второй слой LSTM: 50 нейронов, возвращает окончательное предсказание.

Выходной слой Dense: 2 нейрона, соответствующие двум предсказываемым значениям y_1 и y_2.

Далее компилируем модель. Функцией потерь будет среднеквадратическая ошибка MSE, оптимизатором будет адаптивный метод оценки моментов — Adam. В качестве метрик возьмем MAE и RMSE.

4. Компьютерное моделирование

4.1. Подготовка данных

На данном этапе мы начинаем с решения системы дифференциальных уравнений и последующей подготовки данных для обучения модели LSTM. Используем Python и необходимые библиотеки для выполнения численных расчетов и создания модели машинного обучения.

Так же не забываем подгрузить необходимые библиотеки:

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import solve_ivp
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
```

Code 1. Импортируем библиотеки

Задаем функцию, описывающую систему линейных дифференциальных уравнений первого порядка:

```python
def diff_eq(t, y, a11, a12, a21, a22):
    dy1 = a11 * y[0] + a12 * y[1]
    dy2 = a21 * y[0] + a22 * y[1]
    return [dy1, dy2]
```

Code 2. Функция ссистемы дифференциальных уравнений

Определяем параметры системы a_{ij} и начальные условия $y_1(0)$ и $y_2(0)$:

```python
params = (-0.1, 0.2, -0.2, 0.1)
yini = [1, 0]
times = np.linspace(0, 10, 100)
```

Code 3. Параметры и начальные условия

Используем solve_ivp из библиотеки scipy для численного решения системы дифференциальных уравнений методом Рунге-Кутты:

```python
sol = solve_ivp(lambda t, y: diff_eq(t, y, *params), [0, 10], yini, t_eval=times)
data = np.column_stack((sol.t, sol.y.T))
```

Code 4. Численное решение системы

Приводим данные к форме, подходящей для обучения модели LSTM:

```
train_data = np.reshape(data[:, 1:], (data.
    shape[0], 1, data.shape[1]-1))
```

Code 5. Форма данных для LSTM

4.2. Решение задачи

Создаем и компилируем модель LSTM с использованием библиотеки tensorflow.keras:

```
model = Sequential()
model.add(LSTM(50, return_sequences=True,
    input_shape=(1, 2)))
model.add(LSTM(50))
model.add(Dense(2))

model.compile(loss='mean_squared_error',
    optimizer='adam')
```

Code 6. Модель LSTM

Обучаем модель LSTM на подготовленных данных:

```
history = model.fit(train_data, data[:,
    1:], epochs=100, batch_size=1, verbose
    =1)
```

Code 7. Обучаем модель LSTM

Используем обученную модель для прогнозирования:

```
predictions = model.predict(train_data)
```

Code 8. Прогнозирование

5. Визуализация и анализ результатов

Используем matplotlib для построения графиков оригинальных и предсказанных значений:

```
data = np.hstack((data, predictions))

plt.figure(figsize=(10, 6))
plt.plot(data[:, 0], data[:, 1], color="
    blue", label="y1 (original)")
plt.plot(data[:, 0], data[:, 2], color="
    green", label="y2 (original)")
plt.plot(data[:, 0], data[:, 3], color="red
    ", linestyle="dashed", label="y1 (
    predicted)")
plt.plot(data[:, 0], data[:, 4], color="
    orange", linestyle="dashed", label="y2
    (predicted)")
plt.title("Solving a system of differential
    equations and forecasting LSTM")
plt.xlabel("Time")
plt.ylabel("Function y1(t)    y2(t)")
plt.legend()
plt.show()
```

Code 9. Визуализация результатов

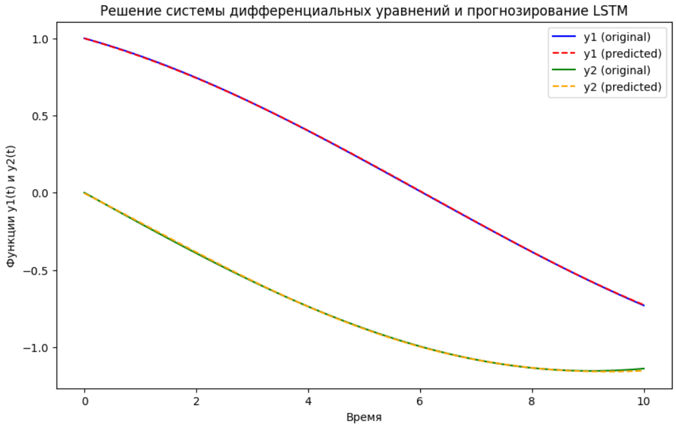

Figure 1. Графическая визуализация результатов

Синий и зеленый графики представляют решения системы дифференциальных уравнений. Красные и оранжевые пунктирные линии представляют прогнозы LSTM модели.

На графике видно, что пунктирные линии (предсказания) практически точь-в-точь совпадают со сплошными линиями (истинные значения). Это говорит о том, что модель показывает высокую точность предсказаний. Значит, предложенный подход демонстрирует высокую эффективность.

Метрики показали следующие результаты:

- MAE $y_1 = 0.0036368336627955605$,
- MAE $y_2 = 0.004164645867341305$,
- RMSE $y_1 = 0.004439548863881781$,
- RMSE $y_2 = 0.005544029807856257$.

Исходя из этих значений можем сделать вывод, что модель обучилась с очень достойной точностью, поскольку в данных метриках чем ближе значение к нулю — тем лучше.

6. Заключение

Мы рассмотрели задачу прогнозирования временных рядов с использованием системы дифференциальных уравнений и модели LSTM. Построенная модель позволяет предсказывать будущее поведение временного ряда на основе исторических данных. Результаты показывают, что предложенный метод эффективно решает поставленную задачу и может быть использован в дальнейших исследованиях.

7. Литература

1. N.D. Lewis, Ph.D. Deep Learning for Time Series Forecasting: Predict the Future with MLPs, CNNs, and LSTMs.
2. J.C. Butcher. Numerical Methods for Ordinary Differential Equations.
3. Douglas C. Montgomery. Introduction to Time Series Analysis and Forecasting.
4. Felix A. Gers et al. Long Short-Term Memory Networks for Time Series Prediction.
5. V.N. Starovoitov et al. Numerical Methods for Differential Equations, Optimization, and Technological Problems.

8. Контакты

Со мной можно связаться:
rourkean@yandex.ru
m3nandre