Прогнозирование временных рядов

С использованием RNN и дифференциальных уравнений

Вихляев Егор, ММТ-21

Математическая постановка задачи

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 $y1(t)$ и $y2(t)$ - это функции времени, описывающие поведение системы. Тогда система дифференциальных уравнений имеет вид:

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

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

Для численного решения системы дифференциальных уравнений применим метод Рунге-Кутты четвертого порядка.Численное решение будет выполняться на заданном интервале времени $[0, T]$ с шагом h. Пусть начальные условия для системы заданы как y_1(0) =$_{96}$ y_10$_{10}$ и y_2(0) = y_20.

Математическая постановка задачи

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:

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

Методы решения

- Для численного решения системы линейных дифференциальных уравнений первого порядка используем метод Рунге-Кутты четвертого порядка. Этот метод выбран из-за его высокой точности и устойчивости при численном решении обыкновенных ДУ.
- Long Short-Term Memory (LSTM) — это тип рекуррентной нейрона сети (RNN), специально разработанный для обработки и предсказания данных временных рядов. 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).

Компьютерное моделирование

- В качестве подготовки данных для LSTM — решаем систему ДУ:

```
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

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

Code 3. Параметры и начальные условия

```
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

Компьютерное моделирование

- Далее переводим 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

Компьютерное моделирование

- Создаем модель:

```python
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

Компьютерное моделирование

- обучаем 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

Визуализация результатов

- Используем matplotlib для построения графиков истинных 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

Визуализация результатов

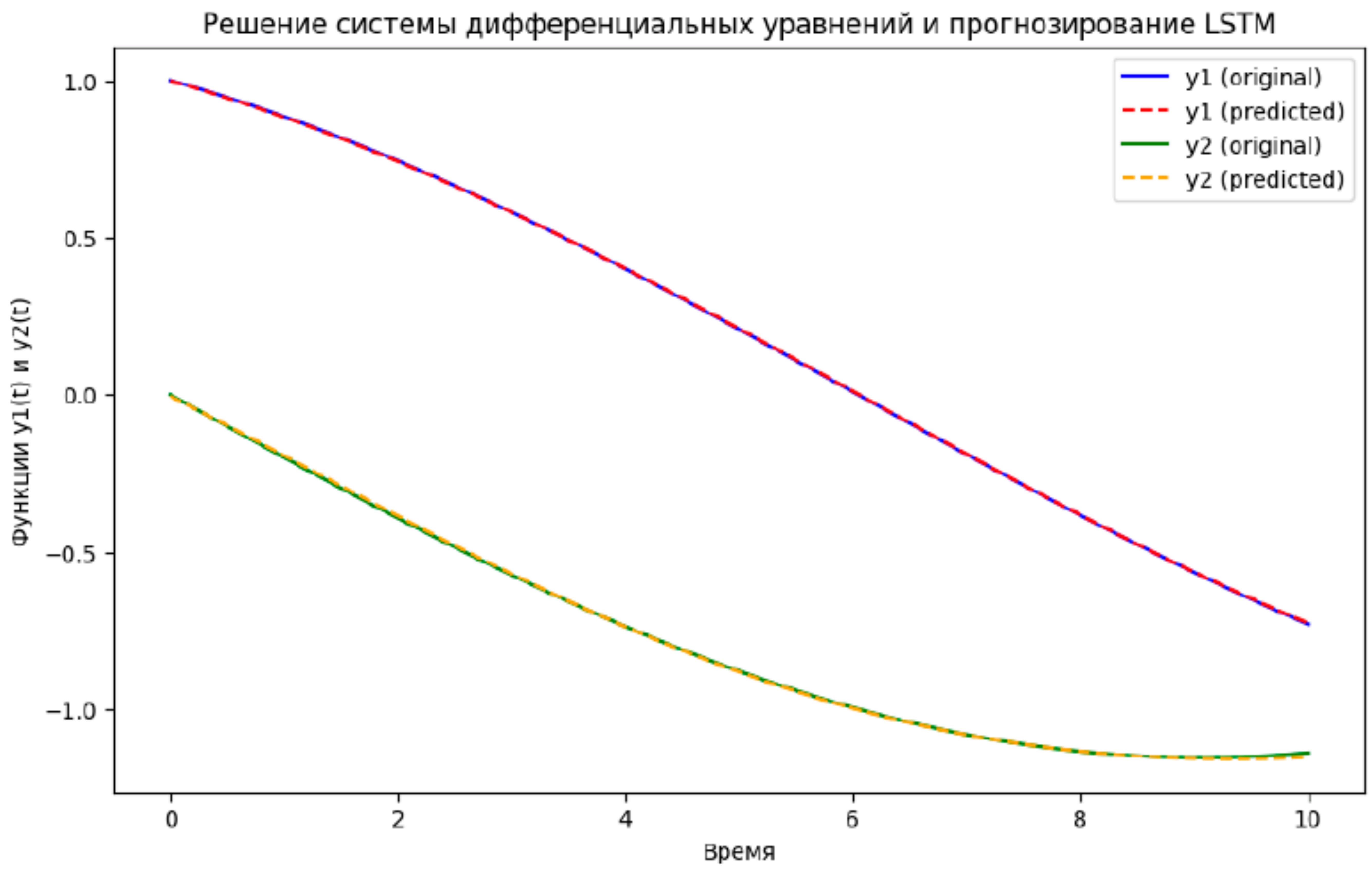

Figure 1. Графическая визуализация результатов

Визуализация результатов

```
MAE y_1 =  0.0036368336627955605
MAE y_2 =  0.004164645867341305
RMSE y_1 =  0.004439548863881781
RMSE y_2 =  0.005544029807856257
```

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.