Московский авиационный институт (национальный исследовательский университет)

Институт № 8 «Компьютерные науки и прикладная математика»

Кафедра № 806 «Вычислительная математика и программирование»

Выпускная квалификационная работа бакалавра
на тему:

«Применение дифференциальных уравнений для повышения эффективности обучения нейронных сетей»

Студент группы М8О-407Б-19: Будникова Валерия Павловна

Научный руководитель: Ревизников Дмитрий Леонидови д.ф.-м.н., профессор, профессор кафедры 806 МАИ

Москва – 2023

Актуальность темы

Актуальность темы данной работы связана с широким применением методов машинного обучения и нейронных сетей в настоящее время.

Модель нейронной сети с применением обыкновенных дифференциальных уравнений является альтернативным взглядом на структуру глубоких нейросетей. Работа сети основывается на использовании алгоритма, решающего обыкновенное дифференциальное уравнение.

Цель и задача работы

Цель – разработка нейросетевых архитектур, использующих концепцию численных методов решения обыкновенных дифференциальных уравнений.

Задачи:

- исследование связи обыкновенных дифференциальных уравнений с внутренним устройством нейронной сети;
- анализ перехода от концепции дискретных слоев к непрерывным;
- реализация алгоритмов численного решения обыкновенных дифференциальных уравнений;
- реализация нейросетевой архитектуры.

Постановка задачи

Дано:

- Искусственно сгенерированные данные различных математических функций.
- Различные параметризованные дифференциальные уравнения.

Необходимо:

- Разработать модель нейронной дифференциальной сети, которая смогла бы решать задачу аппроксимации и предсказывания функций динамики.
- Произвести тестирование данной модели на различных данных.
- Провести сравнение и анализ результатов.

Логика работы

1. Изучить численные методы решения обыкновенного дифференциального уравнения и реализовать данные методы.
2. Изучить работу и внутреннее устройство нейронной сети.
3. Реализовать механизм прямого и обратного распространения сети, с использованием обыкновенных дифференциальных уравнений.
4. Проанализировать внутреннюю функцию сети, определить, как она влияет на результат.
5. Подобрать и проанализировать различные параметры сети, для увеличения точности.
6. Сделать выводы, основываясь на качестве работы сети, ее структуре и используемых ресурсах.

Стек технологий

- **Язык программирования:** Python 3
- **Среда разработки:** редактор кода Visual Studio Code, среда разработки Jupyter notebook, облачная среда Google Colaboratory
- **Библиотеки:**
  - numpy - позволяет работать с многомерными массивами и матрицами
  - matplotlib - это библиотека для визуализации данных
  - torch — современная библиотека глубокого обучения, позволяет работать с многомерными массивами и матрицами, а также конструировать модели сетей

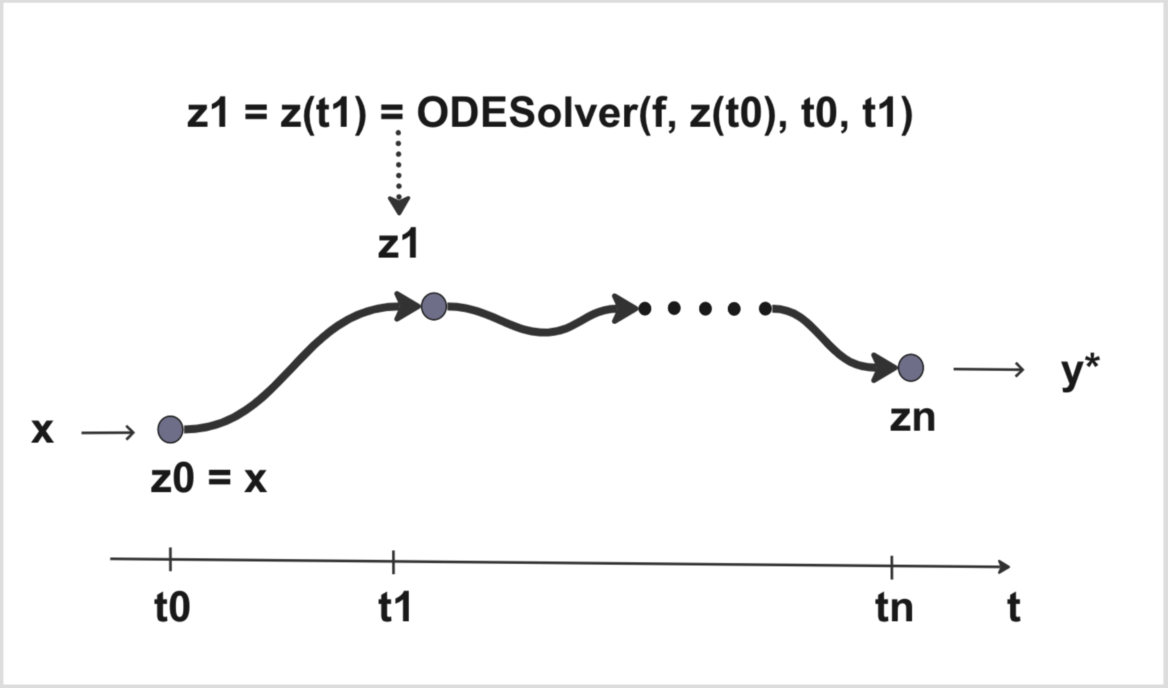
Принцип работы сети. Эволюция системы вперед.

x = z0 ⟶ ODESolver ⟶ z1 ⟶ ODESolver ⟶ z2 ⟶ ... ⟶ ODESolver ⟶ zn = y*

$$\frac{dz(t)}{dt} = f(t,\, z(t),\, q)$$

$$z(t0) = z0$$

$$z(t1) = ODESolver(f,\, z(t0),\, t0,\, t1)$$



Принцип работы сети. Эволюция системы.

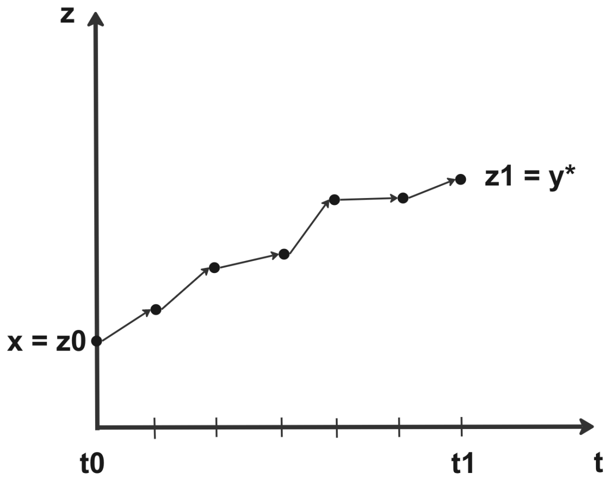
Рассмотрим пример аппроксимации функции на примере сети, состоящей из одного блока решателя обыкновенных дифференциальных уравнений.

Эволюция системы запускается из состояния z0 с параметризованной функцией динамики f. В блоке ODESolver используется любой выбранный метод решения ОДУ.

$$\frac{dz(t)}{dt} = f(t,\, z(t),\, q)$$

$$z(t0) = z0$$

x = z0 ⟶ **ODESolver** ⟶ **z1 = y***



После того, как система оказывается в новом состоянии z1, оно сравнивается с эталонным значением y.

Ошибка минимизируется варьированием параметров функции f.

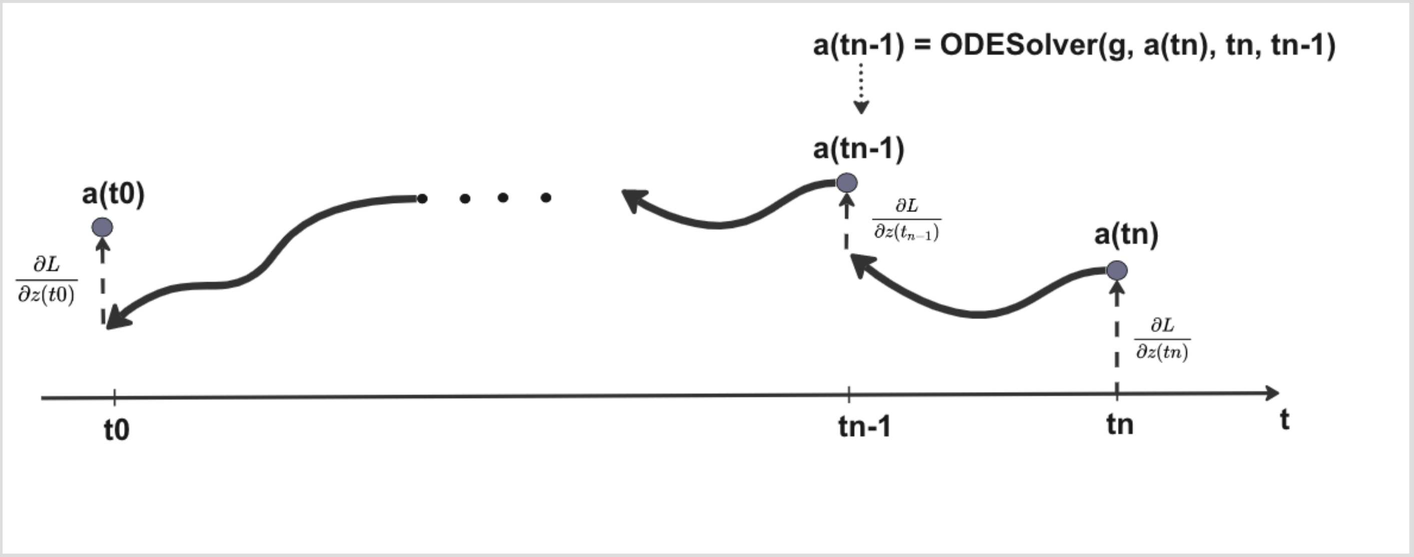
Принцип работы сети. Эволюция системы назад.

$$a(t) = \frac{\partial L}{\partial z(t)}$$

$$a0 \longleftarrow \boxed{\text{ODESolver}} \longleftarrow \cdots \longleftarrow a2 \longleftarrow \boxed{\text{ODESolver}} \longleftarrow a1 \longleftarrow \boxed{\text{ODESolver}} \longleftarrow an = \frac{\partial L}{\partial z(tn)}$$

$$\frac{da(t)}{dt} = -a(t)\,\frac{\partial f(t,\, z(t),\, q)}{\partial z} = g$$

$$a(tn) = \frac{\partial L}{\partial z(tn)}$$

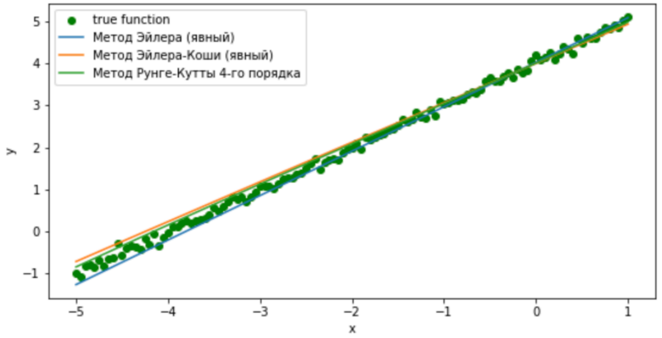


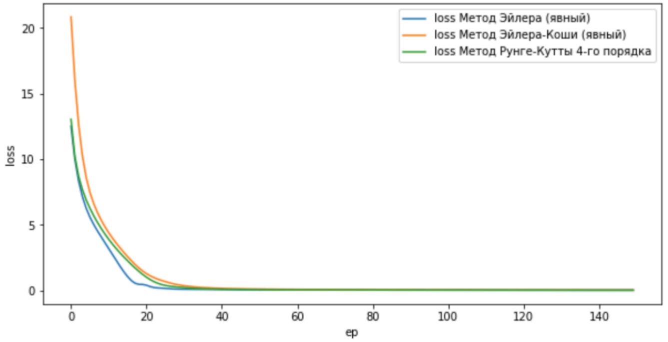
$$\frac{\partial L}{\partial q} = \int_{t_n}^{t_{n-1}} a(t)\,\frac{\partial f(z(t),\, t,\, q)}{\partial q}\,dt$$

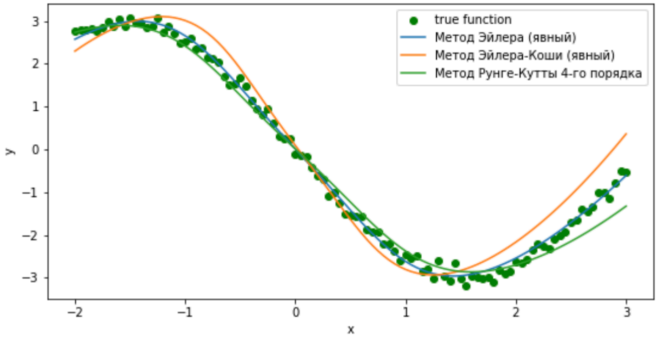
Описание программной разработки

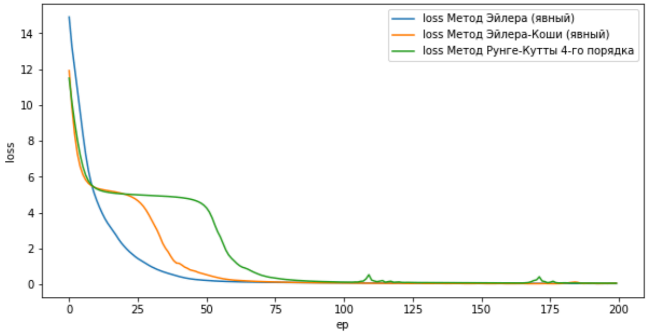
QR-код ссылки, где выложен код

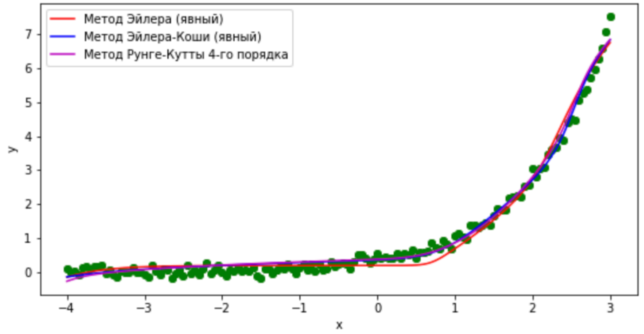
Результаты аппроксимации функций

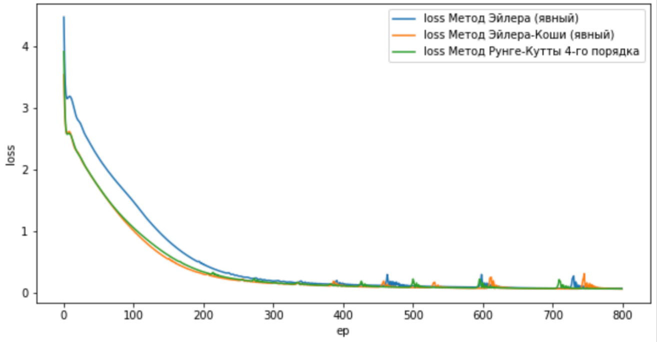




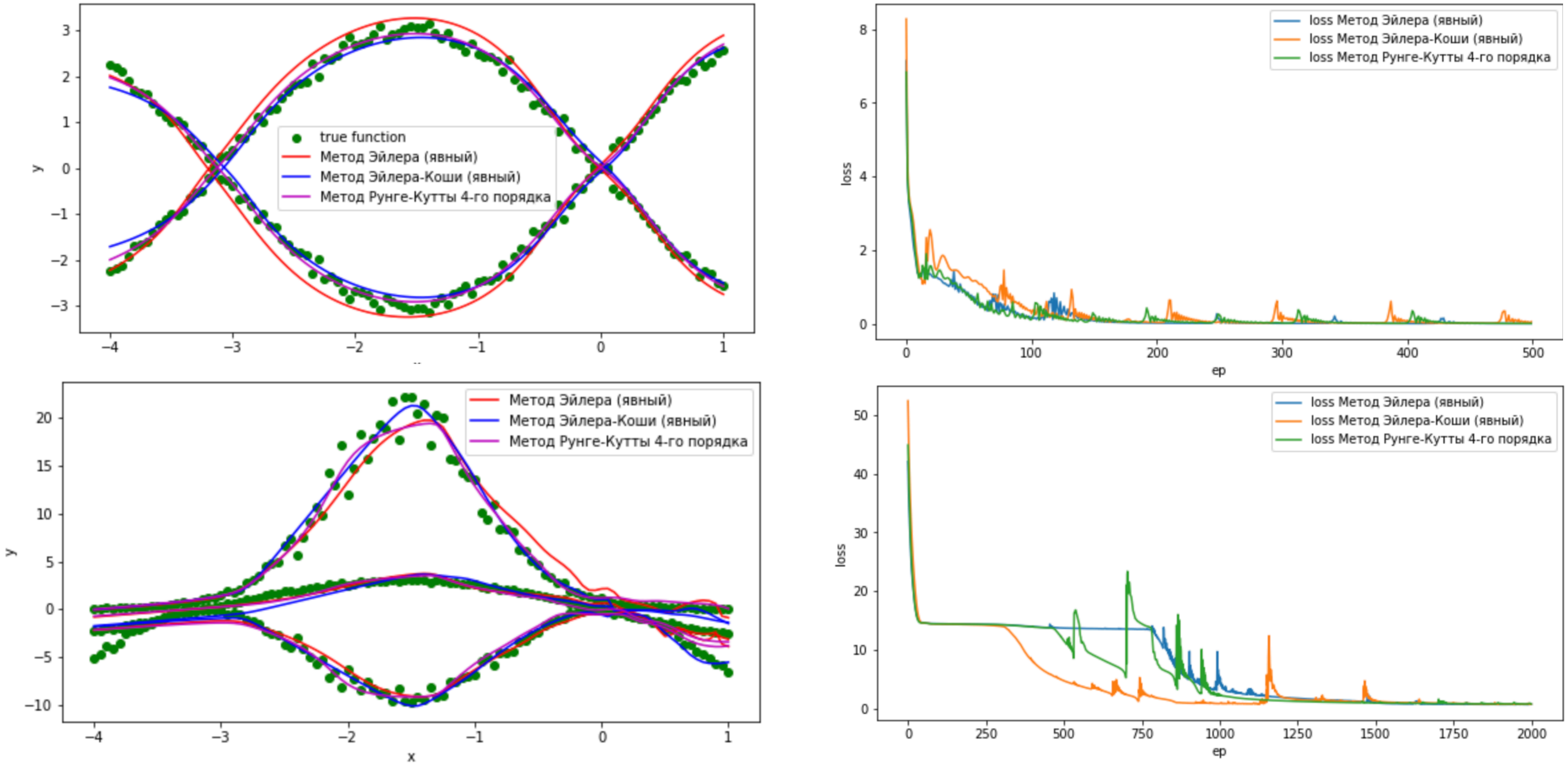




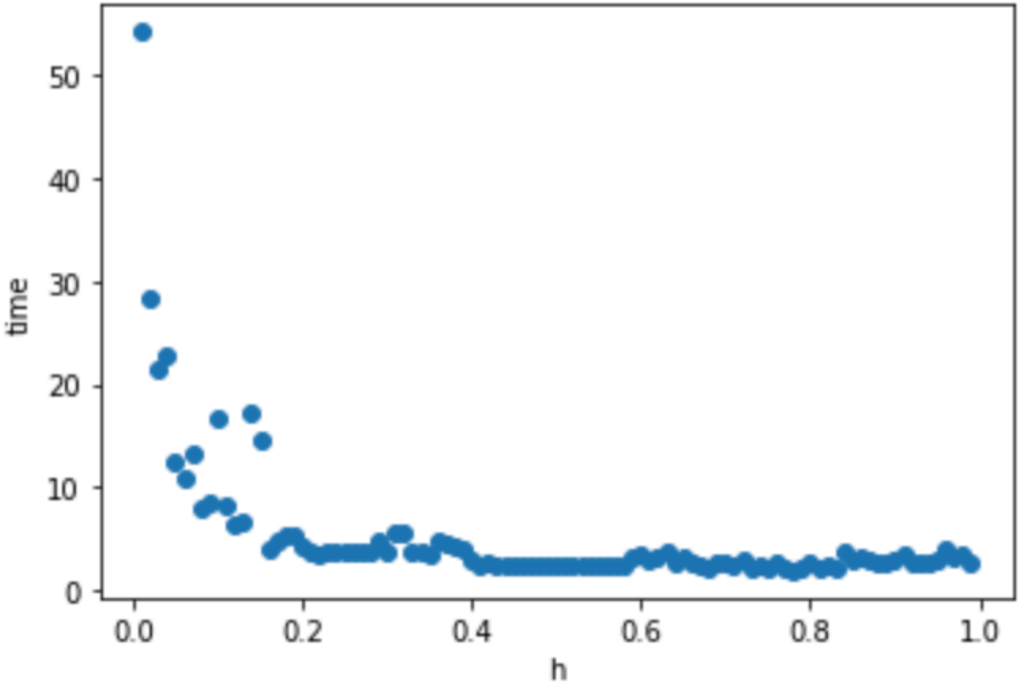


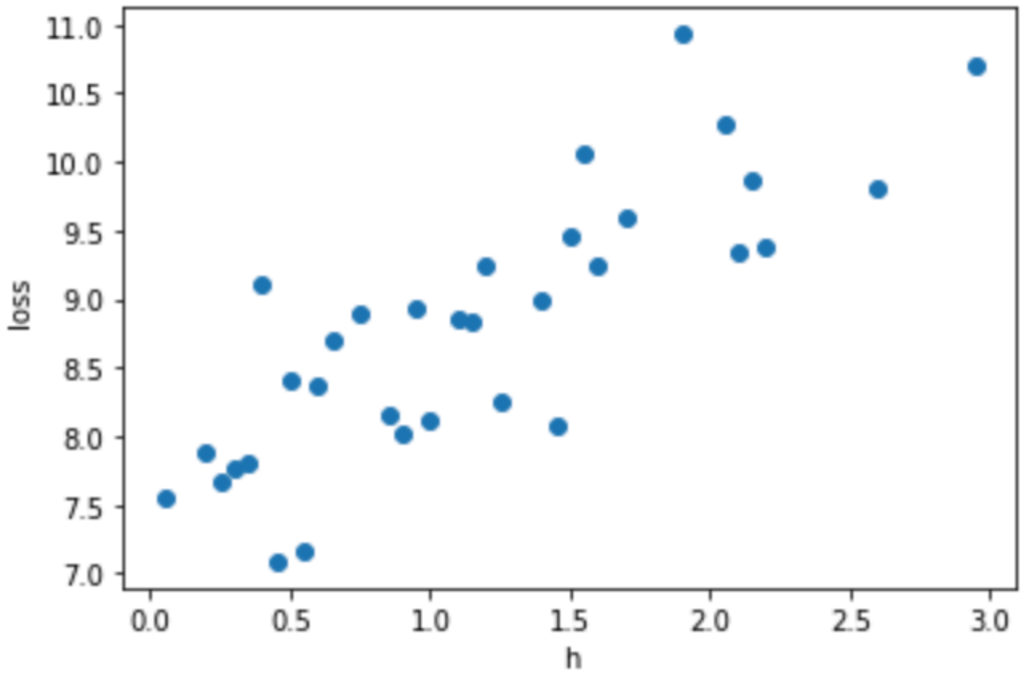


Результаты аппроксимации векторной функции

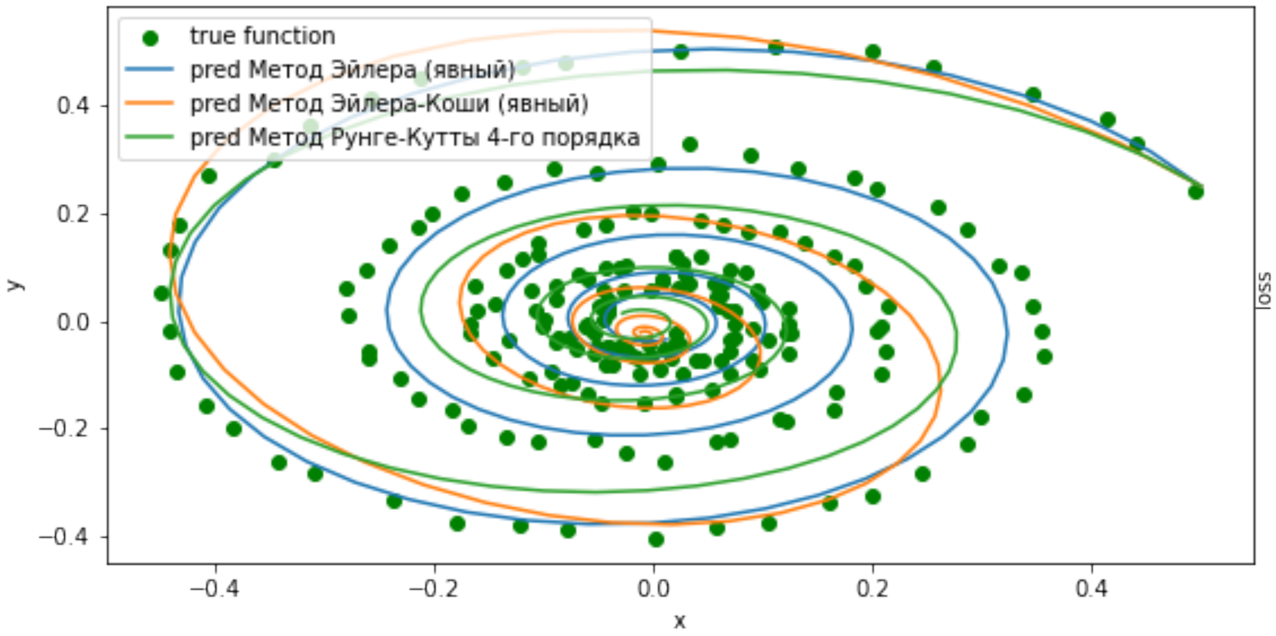


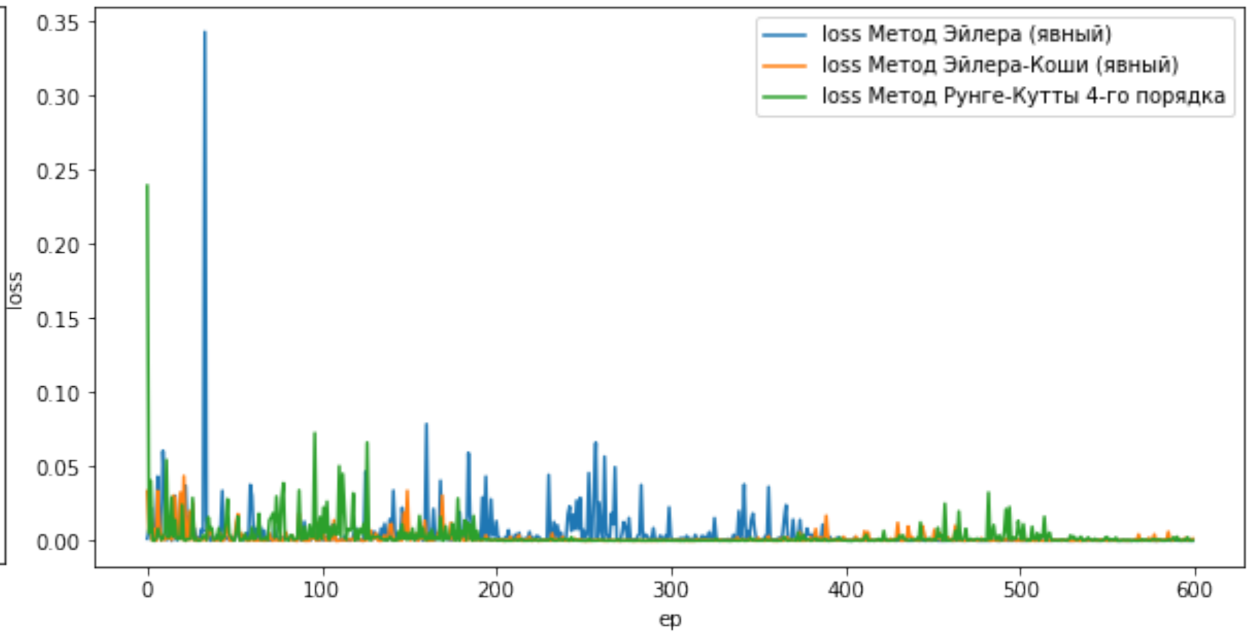
Зависимость времени работы сети и точности от параметра шага решателя





Результаты предсказания функции динамики





Оценка результата

- Данная модель затрачивает меньше ресурсов памяти при обучении, чем обычная глубокая нейронная сеть прямого распространения, что позволяет применять данную модель в ограниченных условиях, также появляется возможность адаптивного времени вычисления.
- При использовании линейных функции, а также полиномов и сложных функций в качестве внутренней функции сети, возможно аппроксимировать конечный круг функций.
- Комбинации матриц, представляющих из себя внутреннюю функцию сети позволяют расширить множество функций для аппроксимации.
- Данная сеть хорошо справляется с восстановлением функции динамики, необходимо задать в качестве функции сети аналогично функцию, состоящую из комбинации матриц, а предсказывать действие той или иной матрицы на функцию.