Генерация нейронных сетей для моделирования поведения динамических систем

Павел Геннадьевич Алимов, 20.Б11-мм
Научный руководитель: Инженер-исследователь СПБГУ В.И. Гориховский

Санкт-Петербургский государственный университет
Кафедра системного программирования

1 июня 2023г.

Введение

- Во многих современных областях физики, биологии, механики, экономики, математики используется моделирование динамических систем
- Сложностью моделирования является необходимость серийного нахождения решений дифференциальных уравнений
- Численные подходы к нахождению решения больших задач могут быть очень ресурсоёмкими
- Для аппроксимации решений дифференциальных уравнений могут использоваться нейронные сети прямого доступа [ВКР Полетанского В. 2021 г.]
- Разрабатываем генератор нейронных сетей с экспортом в качестве исполняемых файлов на C++

Актуальность

- Моделирование потока разреженного газа [Aksenova et al., 2019]
- Реконструкция биологических сетей [Mao et al., 2022]
- Моделирование ионных каналов [Lei et al., 2021]
- Моделирование колебательной и химической кинетики многоатомных газов [Kustova et al., 2022]

Современное состояние

Для каждой задачи разрабатывается свой метод или инструмент

- Использование нейронной сети для аппроксимации решения дифференциальных уравнений [Raissi et al. 2017-2019 г.]
- Численные методы основанные на нейронных сетях [Koryagin et al. 2019]
- Применение машинного обучения для выполнения промежуточных вычислений [Gorikhovskii et al. 2022]

Показана применимость нейронных сетей для аппроксимации решения дифференциальных уравнений [Полетанский В. 2022]

Цель

Разработка экспертной системы предлагающей оптимальную топологию и параметры обучения нейронной сети для аппроксимации функций и решения дифференциальных уравнений.
Подцели:

- разработка библиотеки позволяющей
  - генерировать нейронные сети по задаваемым параметрам
  - обучать сгенерированные нейронные сети на наборах данных
  - экспортировать полученные нейронные сети в виде C++ кода
- создание датасета

Задачи

- 5 Семестр:
  - выбор инструментария для реализации прототипа
  - выбор метрик, алгоритмов обучения и функций потерь для обучения нейронных сетей
  - реализация прототипа библиотеки
  - тестирование прототипа
- 6 Семестр
  - разработка архитектуры библиотеки для генерации нейронных сетей
  - реализация библиотеки
  - публикация библиотеки в качестве Python пакета
- 4 Курс
  - разработка экспертной системой
  - разработка оболочки для работы с экспертной системой
  - повышение производительности

Описание библиотеки

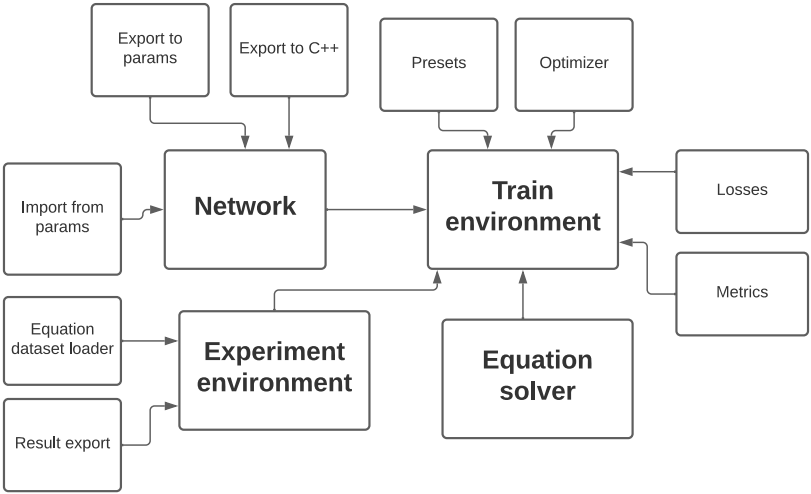


Рис. 1. Общая архитектура

Экспериментальная среда

shape	activations	loss_func	optimizer	epochs	loss	train_time
[1, 10, 10, 10, 10, 10, 10, 1]	['hard_sigmoid', 'hard_sigmoid', 'hard_sigmoid...	Huber	SGD	200	3.564775	10.177364
[1, 10, 10, 10, 10, 10, 10, 1]	['swish', 'swish', 'swish', 'swish', 'swish', ...	Huber	SGD	200	4.019524	10.181402
[1, 10, 10, 10, 10, 10, 10, 1]	['hard_sigmoid', 'hard_sigmoid', 'hard_sigmoid...	Huber	Adam	200	3.468842	10.288455
[1, 10, 10, 10, 10, 10, 10, 1]	['swish', 'swish', 'swish', 'swish', 'swish', ...	Huber	Adam	200	0.003190	10.293120
[1, 10, 10, 10, 10, 10, 10, 1]	['hard_sigmoid', 'hard_sigmoid', 'hard_sigmoid...	Huber	RMSprop	200	3.469033	10.671893
[1, 10, 10, 10, 10, 10, 10, 1]	['swish', 'swish', 'swish', 'swish', 'swish', ...	Huber	RMSprop	200	0.546987	10.654902
[1, 10, 10, 10, 10, 10, 10, 1]	['hard_sigmoid', 'hard_sigmoid', 'hard_sigmoid...	MeanAbsolutePercentageError	SGD	200	56.746208	13.934922
[1, 10, 10, 10, 10, 10, 10, 1]	['swish', 'swish', 'swish', 'swish', 'swish', ...	MeanAbsolutePercentageError	SGD	200	56.601738	14.641225
[1, 10, 10, 10, 10, 10, 10, 1]	['hard_sigmoid', 'hard_sigmoid', 'hard_sigmoid...	MeanAbsolutePercentageError	Adam	200	56.540363	16.393556
[1, 10, 10, 10, 10, 10, 10, 1]	['swish', 'swish', 'swish', 'swish', 'swish', ...	MeanAbsolutePercentageError	Adam	200	3.640137	13.015069
[1, 10, 10, 10, 10, 10, 10, 1]	['hard_sigmoid', 'hard_sigmoid', 'hard_sigmoid...	MeanAbsolutePercentageError	RMSprop	200	56.789879	25.429137
[1, 10, 10, 10, 10, 10, 10, 1]	['swish', 'swish', 'swish', 'swish', 'swish', ...	MeanAbsolutePercentageError	RMSprop	200	8.269400	24.360816
[1, 10, 10, 10, 10, 10, 10, 1]	['hard_sigmoid', 'hard_sigmoid', 'hard_sigmoid...	MeanSquaredError	SGD	200	24.164396	17.479131
[1, 10, 10, 10, 10, 10, 10, 1]	['swish', 'swish', 'swish', 'swish', 'swish', ...	MeanSquaredError	SGD	200	24.164421	18.342251
[1, 10, 10, 10, 10, 10, 10, 1]	['hard_sigmoid', 'hard_sigmoid', 'hard_sigmoid...	MeanSquaredError	Adam	200	24.164398	13.820068
[1, 10, 10, 10, 10, 10, 10, 1]	['swish', 'swish', 'swish', 'swish', 'swish', ...	MeanSquaredError	Adam	200	0.123417	13.529513
[1, 10, 10, 10, 10, 10, 10, 1]	['hard_sigmoid', 'hard_sigmoid', 'hard_sigmoid...	MeanSquaredError	RMSprop	200	24.167412	18.252851
[1, 10, 10, 10, 10, 10, 10, 1]	['swish', 'swish', 'swish', 'swish', 'swish', ...	MeanSquaredError	RMSprop	200	2.522867	18.131187

Рис. 3. Часть датасета

Тестирование

- Проведено unit-тестирование
- С помощью экспериментальной среды проведено общее тестирование работоспособности системы
- Проведено тестирование на отдельных дифференциальных уравнениях и системах дифференциальных уравнений

Генерация C++ кода

Язык	Форма нейронной сети	Размер данных	Время (мс)
Python	10-10-10-10-10-10	5 000	24050 ± 628
C++	10-10-10-10-10-10	5 000	35 ± 1
Python	10-10-10-10-10-10	25 000	117882 ± 129
C++	10-10-10-10-10-10	25 000	179 ± 9
Python	100-100-100	5 000	16621 ± 18
C++	100-100-100	5 000	531 ± 6
Python	100-100-100	25 000	83429 ± 635
C++	100-100-100	25 000	2636 ± 6
Python	500-500-500	5 000	18264 ± 63
C++	500-500-500	5 000	14688 ± 144
Python	500-500-500	25 000	91165 ± 81
C++	500-500-500	25 000	73915 ± 1022

Экспертная система

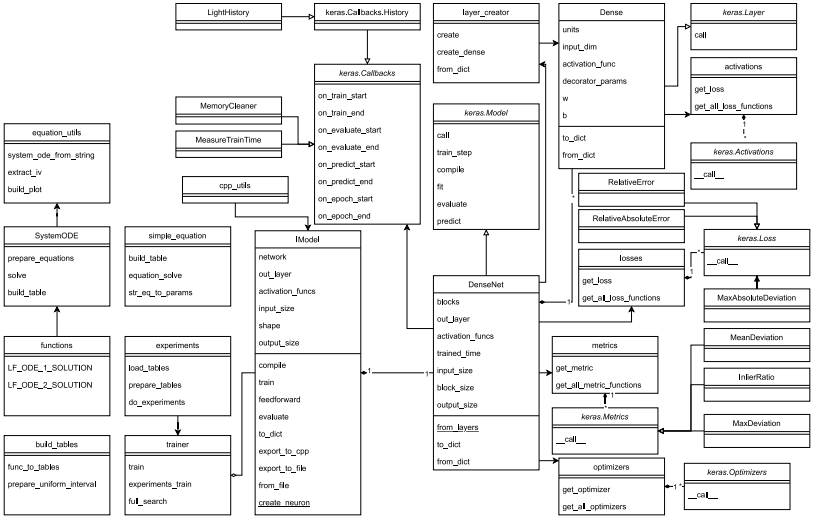
- Переборный вариант по заготовленным шаблонам
- Возвращает набор наиболее подходящих нейронных сетей
- Для одного дифференциального уравнения поиск набора удовлетворяющих нейронных сетей занял более двух суток

Результаты

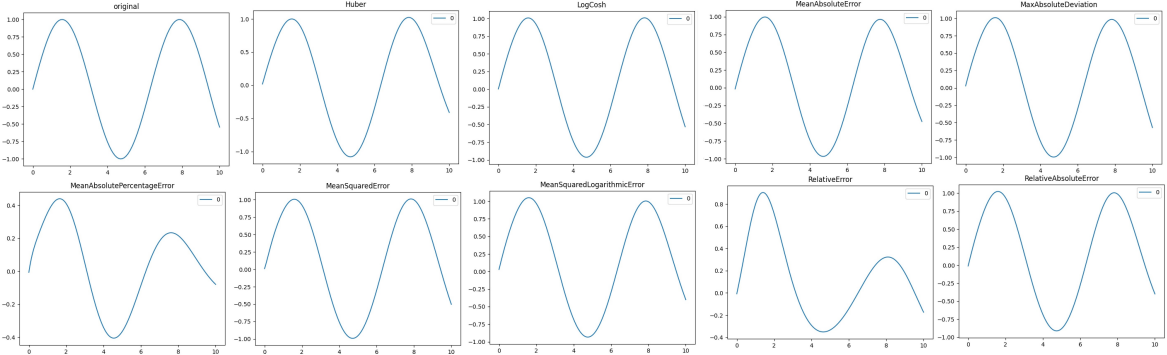
- Реализована библиотека[1]
- Выложен Python-пакет[2]
- Создана экспериментальная среда
- Проведено тестирование
- Сделана генерация C++ кода по нейронной сети
- Собираются данные для экспертной системы
- Реализован переборный вариант экспертной системы

[1] Исходный код библиотеки: https://github.com/Krekep/degann
[2] Ссылка на Python-пакет: https://pypi.org/project/degann/

Приложение 1. Архитектура библиотеки



Приложение 2. Предсказание синуса



Приложение 3. Предсказание решения дифференциального уравнения