Разработка пакета нейросетевой аппроксимации дифференциальных уравнений DEGANN

Алимов П.Г., СПбГУ, Санкт-Петербург st076209@student.spbu.ru,
Гориховский В.И., СПбГУ, Санкт-Петербург v.gorikhovskii@spbu.ru

Аннотация

Во многих современных областях наук возникает необходимость моделирования динамических систем, и для ускорения вычисления дифференциальных уравнений исследователи прибегают к аппроксимации решения дифференциальных уравнений с помощью нейронных сетей. При этом на текущий момент отсутствует инструментарий, позволяющий быстро пользоваться нейронными сетями для аппроксимации.

Данная работа посвящена разработке Python пакета DEGANN для нейросетевой аппроксимации дифференциальных уравнений по заданным требованиям к полученному решению.

Введение

Во многих современных областях наук возникает необходимость моделирования динамических систем.

- В работе [1] авторам требовалось многократное нахождение решения системы дифференциальных уравнений с помощью прямого моделирования методом Монте-Карло (DSMC). Для уменьшения вычислительной сложности расчётов авторы разработали и обучили нейронную сеть, позволяющую аппроксимировать численные решения.

- В исследовании [2] авторы проводили моделирование биологических сетей, содержащих более 10 узлов. Для решения этой задачи была реализована многослойная нейронная сеть, основанная на перцептронах.

- В работе [3] сделана попытка смягчить ограничения традиционного численного подхода к моделированию проводимости ионного канала с помощью применения нейронных сетей.

Несмотря на то, что в решении множества различных задач успешно применяются нейронные сети, оказывается, что не существует универсального инструмента, позволяющего быстро пользоваться таким подходом. Текущие наработки предлагают только решения частных случаев.

- В исследовании 2017-2019г. Raissi et al. были разработаны нейронные сети для аппроксимации решений небольшого набора уравнений в частных производных: Бюргера, Шрёдингера, Навье-Стокса, Аллен-Кана и Кортевега — де Фриза [4].

- Flamant et al. разрабатывают универсальную нейронную сеть, которая может аппроксимировать решение любого дифференциального уравнения [5]. Полученный инструмент не подходит для быстрых серийных вычислений в связи со сложностью самой НС.

Таким образом создание решения, позволяющего автоматически подбирать топологию и параметры обучения нейронной сети для аппроксимации решения заданного дифференциального уравнения, является актуальной.

Архитектура решения

Для создания решения, предназначенного для автоматического поиска подходящей топологии нейронной сети, необходимо создать модуль, позволяющий генерировать настраиваемые НС. Для этого нужно сделать обёртку над существующим инструментом для работы с нейронными сетями или собственную реализацию.

Также необходимо определить методы оптимизации в пространстве топологий нейронных сетей. И для достижения автоматизации должна быть реализована экспертная система, которая автоматически предлагает метапараметры для методов оптимизации, которую в свою очередь возвращают подходящую топологию нейронной сети.

Таким образом решение на высоком уровне должно представлять собой экспертную систему. А экспертная система, в свою очередь, должна представлять собой оболочку над библиотекой генерации нейронных сетей, которая позволяет по заданным параметрам создавать и обучать нейронные сети, осуществлять их сохранение и загрузку в виде набора весов, производить построение исполняемого кода на C++ для проведения серийных вычислений. Общая архитектура решения представлена на рисунке 1 и относится к классу слоистых архитектур.

Эксперименты

Для анализа точности, эффективности и производительности алгоритмов поиска оптимальной топологии, а также определения подходящих параметров алгоритмов поиска были поставлены эксперименты.

Рис. 1: Общее представление архитектуры пакета DEGANN.

Дизайн экспериментов

Для проведения экспериментов были выбраны семь уравнений:

$$y' + 3y = 0,\ y(0) = 1 \tag{LF-ODE1}$$

$$y = ln(1+x) \tag{Log}$$

$$y'' + 100y = 0,\ y(0) = 0, y'(0) = 10 \tag{LH-ODE1}$$

$$y = e^{-\frac{(x-0.5)^2}{0.08}} \tag{Gauss}$$

$$y' + \frac{y-2x}{x+0.1} = 0,\ y(0) = 20 \tag{NL-ODE1}$$

$$y' = 0,\ y(0) = 1 \tag{LF-ODE2}$$

$$y = \frac{\sin(5x) * \log_2(u+1)}{\sqrt{1+t}} \tag{Multidim}$$

Результаты экспериментов

Таблица 1 описывает вероятность в процентах получения, в результате обучения, нейронной сети, которая преодолевает порог в 25% по функции потерь MeanAPE, для случайного поиска и метода имитации отжига SAM_lin_lin. Можно заметить, что на всех уравнениях, кроме LH-ODE1, Multidim, оба алгоритма дают большую вероятность успешного обучения НС, при чём случайный поиск даёт результат быстрее, чем метод имитации отжига. Но на LH-ODE1 и Multidim метод имитации отжига даёт вероятность сходимости на порядок больше.

Таблица 1: Таблица с вероятностью успешного обучения нейронной сети на ДУ с порогом 25% по функции потерь MeanAPE

Название уравнения	Случайный поиск			SAM_lin_lin		
Размер данных	50	150	400	50	150	400
Log	42.5%	80%	21.9%	57.1%	60.6%	22.2%
Gauss	47.6%	76.9%	83.3%	22.2%	44.4%	35.7%
NL-ODE1	100%	100%	100%	100%	100%	100%
LF-ODE1	90.9%	100%	100%	100%	100%	100%
LF-ODE2	100%	100%	100%	100%	100%	100%
LH-ODE1	0.9%	0%	7.4%	2.1%	0%	8.3%
Multidim	2.6%	<0.1%	26.6%	4.5%	0.2%	26.6%

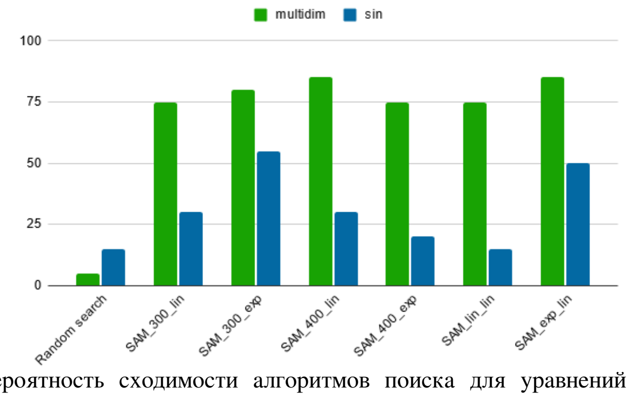

Рис. 2: Вероятность сходимости алгоритмов поиска для уравнений LH-ODE1, Multidim и ограничения 10% по метрике MeanAPE.

На рисунке 2 изображена гистограмма, отражающая вероятность получения результата, подходящего под заданные условия, для алгоритмов случайного поиска и метода имитации отжига. Видно, что метод имитации отжига гарантирует большую вероятность сходимости, чем случайный поиск, и при этом разные конфигурации лучше подходят под разные задачи.

Таким образом, агрегируя выводы, получаем следующие результаты.

- Более чем в 90% случаев случайного поиска достаточно для получения подходящей топологии нейронной сети.

- При определённых конфигурациях метод имитации отжига сходится на уравнениях (LH-ODE1, Multidim) в 3-4 раза чаще, чем случайный поиск.
- На уравнениях помимо (LH-ODE1, Multidim) для нахождения подходящей топологии нейронной сети алгоритмам требуется обучить менее 100 моделей.
- При увеличении ограничений на требуемое значение функции потерь вероятность сходимости алгоритма случайного поиска убывает быстрее, чем вероятность сходимости метода имитации отжига для поиска оптимальной топологии НС.

Экспертная система

По результатам проведённых экспериментов были выделены возможные метки, описывающие поданную на вход задачу, и в зависимости от набора меток экспертная система проводит конфигурацию конвейера алгоритмов поиска топологии нейронных сетей. Также реализован графический интерфейс над экспертной системой, делающий её более простой в использовании.

Архитектура

По результатам экспериментов были выделены метки, описывающие ограничения на получаемое решение, и общая архитектура представленная на рисунке 3. По меткам строятся ограничения и выбираются алгоритмы и их параметры, которые с наибольшей вероятностью удовлетворят запрос пользователя. Метки представлены в следующем списке.

1. Требуемая точность решения ($precision$).
2. Тип функции, который принимает решение дифференциального уравнения ($type$).
3. Время работы метода feedforward (predict) обученной нейронной сети ($work\ time$).
4. Размер тренировочной выборки данных ($data\ size$).

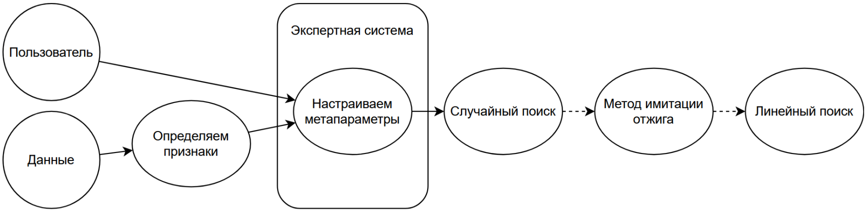

Рис. 3: Поток действий при работе с экспертной системой.

Пример работы

Для валидации работоспособности экспертной системы перед ней были поставлены две задача аппроксимации модели нагруженного гармонического осцилятора с учётом свободных вращений (FHO-FR). Данная модель взята из ВКР Андрея Исакова. Были выставлены следующие требования к решению:

- максимальная точность аппроксимации — для метки $precision$ было установлено значение $maximal$;
- скорость аппроксимации не является критическим фактором — для метки $work\ time$ было установлено значение $long$.

Также в качестве подсказок системе были переданы $unknown$ и $very\ small$ в качестве значений меток $precision$ и $data\ size$ соответственно.

В результате запуска параметризованных экспертной системой алгоритмов поиска была получена обученная нейронная сеть в виде словаря параметров, которую в дальнейшем можно экспортировать как в виде параметров, так и в качестве функции на C++. Полученная нейронная сеть содержала три скрытых слоя по 15, 17 и 10 нейронов в каждом соответственно и показала ошибку в 1% по функции потерь MeanAPE на валидационных данных, что говорит о правильном определении топологии НС для данной задачи и положительной оценке работы экспертной системе в том числе и в реальных задачах.

Заключение

В рамках данной работы были получены следующие результаты.

1. Реализованы алгоритмы случайного, линейного поисков и метод имитации отжига для поиска оптимальной топологии в пространстве топологий нейронных сетей.

2. Выполнено экспериментальное исследование, показавшее, что метод имитации отжига на функциях с несколькими независимыми переменными и функциях вида синусоиды получает требуемый результат в 3-4 раза чаще, чем случайный поиск, но при работе обучает в среднем большее количество нейронных сетей и дольше работает.

3. Реализован и опубликован пакет DEGANN[1], включающий в себя библиотеку генерации нейронных сетей, алгоритмы поиска в пространстве топологий нейронных сетей и экспертную систему для автоматического подбора подходящей топологии нейронной сети для аппроксимации заданного дифференциального уравнения.

Список литературы

[1] Aksenova, Olga A. and Khalidov, Iskander A. Simulation of unstable rarefied gas flows in a channel for different Knudsen numbers // AIP Conference Proceedings 2132, 180009 (2019). https://aip.scitation.org/doi/abs/10.1063/1.5119667

[2] Mao Guo, Zeng Ruigeng, Peng Jintao, Zuo Ke, Pang Zhengbin, Liu Jie. Reconstructing gene regulatory networks of biological function using differential equations of multilayer perceptrons // BMC Bioinformatics 23, 503 (2022). https://doi.org/10.1186/s12859-022-05055-5

[3] Lei Chon Lok, Mirams Gary R. Neural Network Differential Equations For Ion Channel Modelling // Front. Physiol. 12:708944. https://www.frontiersin.org/articles/10.3389/fphys.2021.708944

[4] Raissi M., Perdikaris P., Karniadakis G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations // Journal of Computational Physics, Volume 378, Pages 686-707. https://www.sciencedirect.com/science/article/pii/S0021999118307125

[5] Flamant Cedric, Protopapas Pavlos, Sondak David. Solving Differential Equations Using Neural Network Solution Bundles // https://arxiv.org/abs/2006.14372

[1]Доступен по ссылке: https://pypi.org/project/degann/.