



Санкт-Петербургский государственный университет

Кафедра информатики

Реализация и внедрение архитектуры GRU в пакет нейросетевой аппроксимации ДУ DEGANN

Мурадян Денис Степанович, Алимов Павел Геннадьевич

Рекомендация к публикации: доцент Гориховский В. И.

Санкт-Петербург
2025

- Дифференциальные уравнения (ДУ) играют важную роль в науке и технике, однако численные методы их решения могут быть ресурсоёмкими и давать неточные результаты.
- Использование нейронных сетей для аппроксимации функций, описывающих ДУ, предлагает альтернативный подход.
- Функциональность DEGANN ограничена применением многослойного перцептрона (MLP), который не достаточно хорошо улавливает последовательные зависимости в данных.
- Рекуррентные нейронные сети, специально предназначены для работы с последовательностями и могут повысить эффективность аппроксимации.

Реализация и интеграция в DEGANN

- Разработан класс `TensorflowGRUNet` на базе `tf.keras.Model`.
- Основные этапы реализации:
 - ▶ Создание списка GRU-слоёв с активацией `tanh` для основного состояния и `sigmoid` для рекуррентного.
 - ▶ Последовательная обработка входных данных через GRU-слои с последующим выводом через полносвязный слой с линейной активацией для регрессионного предсказания.
- В классе реализованы методы для настройки оптимизатора, функции потерь, метрик и экспорта конфигурации, а также прямой проход данных.
- Интеграция в DEGANN осуществляется через параметр `net_type` с сохранением общего интерфейса.

- **Цель эксперимента:** сравнить сходимость двух архитектур при идентичных гиперпараметрах и временных промежутков обучения.
- Сравнивались две архитектуры:
 - ▶ GRUNet (на основе GRU)
 - ▶ DenseNet (традиционный MLP)
- Фиксированные параметры эксперимента:
 - ▶ 3 слоя, 30 нейронов в каждом.
 - ▶ Функция потерь: MSE; метрики: MAPE, R^2 .
 - ▶ Время обучения: 15, 30 и 45 секунд.

Выбор тестовых функций и их обоснование

- Для проверки моделей выбраны сложные для аппроксимации функции, каждая из которых предъявляет уникальные требования:
 - ★ $f_{\sin}(x) = \sin(10x)$ – периодическая функция с фиксированной частотой, демонстрирующая способность модели улавливать синусоидальные колебания.
 - ★ $f_{\text{hyperbol}}(x) = \frac{x^2 + 0.5}{x + 0.1}$ – функция с неравномерной зависимостью, где числитель задает квадратичный рост, а малые значения x подчёркиваются особенностями знаменателя.
 - ★ $f_{\text{hardsin}}(x) = \sin\left(\ln(x^{\sin(10x)})\right)$ функция объединяет синусоидальные колебания с логарифмическим масштабированием, что приводит к сложной нелинейной динамике.
- Такой набор функций позволяет всесторонне оценить эффективность сравниваемых архитектур.

Результаты эксперимента и анализ

- Для каждой архитектуры были подсчитаны средние значения метрик для разных временных интервалов:
 - ▶ **15 секунд:**
 - ★ GRUNet: $\text{MAPE} = 6.78$, $R^2 = 0.908$, Память = 112.09 MB.
 - ★ DenseNet: $\text{MAPE} = 15.51$, $R^2 = 0.844$, Память = 36.92 MB.
 - ▶ **30 секунд:**
 - ★ GRUNet: $\text{MAPE} = 3.90$, $R^2 = 0.983$, Память = 114.76 MB.
 - ★ DenseNet: $\text{MAPE} = 6.20$, $R^2 = 0.969$, Память = 39.75 MB.
 - ▶ **45 секунд:**
 - ★ GRUNet: $\text{MAPE} = 2.81$, $R^2 = 0.991$, Память = 118.41 MB.
 - ★ DenseNet: $\text{MAPE} = 5.41$, $R^2 = 0.978$, Память = 44.89 MB.
- Анализ показывает, что GRUNet обеспечивает более высокое качество аппроксимации при идентичных условиях обучения, несмотря на более высокие затраты памяти.

- Внедрение архитектуры GRU в DEGANN повышает точность аппроксимации ДУ.
- Преимущество GRUNet обусловлено её способностью учитывать последовательные зависимости, что приводит к более быстрой сходимости.
- Несколько более высокие затраты памяти не критичны для большинства практических задач.