

ИССЛЕДОВАНИЕ ВОЗМОЖНОСТЕЙ ФРЕЙМВОРКА TENSORFLOW ДЛЯ ПРОГНОЗИРОВАНИЯ ВРЕМЕННЫХ РЯДОВ

Е.В. Унжакова, И.А. Ботыгин
Томский политехнический университет
E-mail: katerina.unz@gmail.com

Введение

Задача прогнозирования будущих значений временного ряда весьма актуальна в наши дни и её решение играет важнейшую роль в различных сферах науки, техники, медицины, экономики, политики и т.д. Для прогнозирования временных рядов используют статистические и структурные модели [1]. Одна из структурных моделей, получившая широкое распространение в наши дни – это модель на искусственных нейронных сетях (ИНС). Способность приложений на базе искусственных нейронных сетей к обобщению и выделению скрытых зависимостей между входными и выходными данными предоставляет уникальные возможности и для задач прогнозирования.

Подготовка данных

Вначале необходимо подготовить данные для обучения и проверки работы ИНС, выбрать архитектуру, основные характеристики и параметры обучения.

Используем набор метеорологических данных с сайта ФГБУ «ВНИИГМИ-МЦД». Для экспериментов были выбраны данные по станции Томск за последние 10 лет (до 2018 г.) по 24 признакам [2]. Наблюдения в выборке каждые 3 часа – это 8 наблюдений за 24 часа. Прогнозирование планируется осуществлять на период в 7 дней по 4 признакам: температура воздуха по сухому термометру, относительная влажность воздуха, максимальная скорость ветра, атмосферное давление на уровне станции.

В соответствии с выбранным сроком прогнозирования (на 7 следующих дней) делаем смещение по времени в данных на 56 шагов ($7 \times 8 = 56$). Конвертируем выборки Pandas в массивы NumPy. ИНС работает лучше всего на значениях между -1 и 1, поэтому отмасштабируем данные между 0 и 1, используя библиотеку Scikit-Learn [3].

Данные для обучения состоят из 26241 наблюдений, тестовые данные – из 2916 наблюдений, в целях ускорения процесса обучения ИНС из полной базы учебных данных были созданы группы более коротких последовательностей данных для обучения.

Обучение с фреймворком TensorFlow

Для программной реализации модели прогнозирования был выбран фреймворк TensorFlow, являющийся открытым программным обеспечением для машинного обучения. Фреймворк TensorFlow был создан в компании

Google и поддерживает много крупномасштабных интеллектуальных приложений [3].

TensorFlow хорошо адаптирован к многопоточной работе, как на CPU, так и GPU (графический процессор), опираясь на архитектуру CUDA. Для данного исследования используются расчёты на CPU, поскольку имеющийся в используемом компьютере GPU не поддерживает требуемую версию технологии CUDA.

Чтобы TensorFlow мог обучать модель нужно создать функцию потерь и алгоритм оптимизации.

Одна очень распространённая функция потерь называется средней квадратичной ошибкой (Mean Squared Error). Она диагностирует, как близко вывод модели соответствует настоящим выходным сигналам. Однако, её использование для ранних временных шагов может вызвать искажение последующих результатов, следовательно, укажем период разогрева в 50 шагов времени.

Функция потерь является важным показателем для оценки производительности оптимизатора. Оптимизаторы обновляют весовые параметры, чтобы минимизировать функцию потерь.

RMSProp (Root Mean Square Propagation) – среднеквадратичное распространение ошибки. В оптимизаторе RMSProp скорость обучения настраивается автоматически и для каждого параметра выбирается разная скорость.

Создание и обучение ИНС

Keras API – библиотека для глубокого машинного обучения, обладает модульной минималисткой и легко расширяемой архитектурой, работает поверх библиотеки TensorFlow [4].

Построение ИНС требует настройки слоев модели. Архитектура создаваемой ИНС содержит два плотных слоя: GRU (управляемый рекуррентный слой) и Dense (базовый регулярный слой).

В GRU используется фильтр обновления и фильтр сброса и его можно применять, когда нужно быстро обучить модель, а вычислительных мощностей недостаточно.

В результате получилась рекуррентная модель искусственной нейронной сети, которая после компиляции готова для обучения. Для целей обучения (по причине использования CPU), выставляем невысокие параметры: 25 эпох и 25 шагов обучения – это 625 наблюдений.

В итоге обучение длилось 18 минут 24 сек. и завершилось на 25 эпохе, со значением функции потерь – 0.01803.

Получение результатов обучения

Рассмотрим результаты обучения. Учебные данные нейросеть видела в процессе обучения, в отличие от тестовых. На графиках обозначение «True.» – настоящие результаты (зелёный), «Pred.» – предсказанные ИНС (синий). Серым цветом выделен период разогрева (рис. 1).

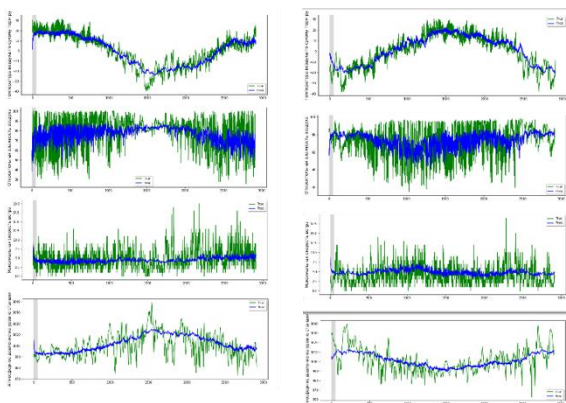


Рис. 1. Результат обучения ИНС на учебных данных (слева) и на тестовых данных (справа)

Как видно на графиках ИНС достаточно точно предсказывает значения, однако следует увеличить количество эпох и шагов обучения, а также добавить функцию обратного вызова EarlyStopping, которая останавливает обучение, когда ошибка перестанет уменьшаться. При компиляции модели добавить метрику assuagasy, которая покажет точность предсказания на тестовом наборе данных.

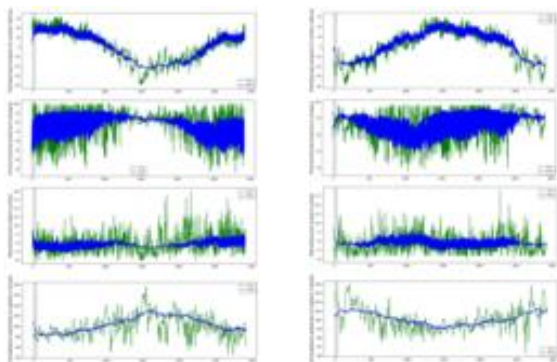


Рис. 2. Результат обучения с новыми параметрами

При количестве 50 эпох и 50 шагов обучение длилось 15 минут 38 сек. (с более мощным CPU) и завершилось на 42 эпохе, так как сработала функция остановки. Точность предсказания на тестовом наборе данных – 74%. Значение функции потерь уменьшилось до 0.01673 и на графиках увеличилась точность предсказания (рис. 2).

Применим созданную рекуррентную ИНС для прогнозирования временного ряда на 7 дней вперёд (январь 2019 г.). Сравнивая результаты с данными сайта World Weather, можно точно сказать, что ИНС очень близка к реальным измерениям (рис. 3).

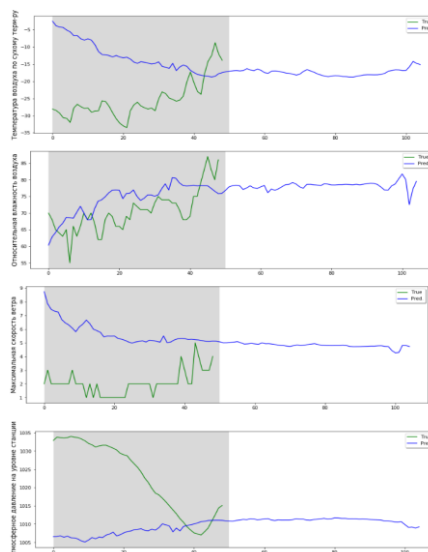


Рис. 3. Предсказанные значения по целевым признакам

Заключение

Исследование показало определённую эффективность выбранных методов для решения подобных задач прогнозирования временных рядов. Кроме того, обучение созданной ИНС достаточно хорошо использует доступные вычислительные возможности, поэтому с более мощным CPU (или подходящим GPU), а также варьируя количество эпох, шагов за эпоху и длину последовательностей можно добиться значительного улучшения скорости обучения, качества предсказания и точности в процентном отношении.

Список использованных источников

1. Обзор методов прогнозирования // URL <https://ivan-shamaev.ru/overview-forecast-methods/#i-14> (дата обращения: 21.11.2019)
2. Специализированные массивы для климатических исследований // URL <http://aisori-m.meteo.ru/waisori/index.xhtml?idata> (дата обращения: 17.11.2019)
3. Жерон, Орельен. Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow. Пер. с англ. — СПб.: ООО «Альфа-книга»: 2018. — 688 с.
4. Антонио, Д. Библиотека Keras – инструмент глубокого обучения. Реализация нейронных сетей с помощью библиотек Theano и TensorFlow / Д. Антонио, П. Суджит; пер. с англ. — Москва: ДМК Пресс, 2018. — 294 с.