

МЕТОДЫ ПРЕДВАРИТЕЛЬНОЙ ОБРАБОТКИ ДАННЫХ ДЛЯ ОПТИМИЗАЦИИ ОБУЧЕНИЯ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ

Статья представляет комплексное исследование методов предварительной обработки данных для нейронных сетей. В работе систематизированы и детально проанализированы современные подходы к подготовке данных, включая нормализацию, работу с категориальными признаками, обработку пропущенных значений и специализированные методы для изображений и временных рядов. Особое внимание уделено практическим аспектам реализации методов предобработки и их влиянию на производительность нейросетевых моделей. Результаты исследования подчеркивают важность корректной предварительной обработки для достижения оптимальных показателей точности и скорости обучения.

ВВЕДЕНИЕ

Современные искусственные нейронные сети демонстрируют высокую эффективность при решении сложных задач машинного обучения, таких как распознавание изображений, обработка естественного языка и прогнозирование временных рядов. Однако качество их работы существенно зависит от корректности предварительной обработки входных данных.[1] Методы предварительной обработки позволяют решать ряд важных задач: устранение различий в масштабах и распределениях данных, фильтрацию шумов и артефактов, коррекцию несбалансированности классов, а также оптимизацию вычислительных процессов за счет уменьшения избыточности информации. Таким образом, разработка и применение эффективных методов предобработки данных представляет собой критически важный этап построения нейросетевых моделей, непосредственно влияющий на их производительность и качество работы. [2]

I. ОСНОВНЫЕ МЕТОДЫ ПРЕДОБРАБОТКИ ДАННЫХ

Нормализация и стандартизация данных:

Нормализация данных представляет собой процесс приведения числовых признаков к единому масштабу, что особенно важно для алгоритмов, основанных на расчетах расстояний или использующих градиентные методы оптимизации. На практике наиболее часто применяется min-max нормализация, которая линейно преобразует данные в заданный диапазон, обычно от 0 до 1. Стандартизация предполагает центрирование данных вокруг нуля с единичной дисперсией и особенно эффективна, когда данные содержат выбросы или когда предполагается их нормальное распределение.[1-3]

Работа с категориальными признаками:

Категориальные признаки требуют особого подхода при подготовке данных для нейросетевых моделей. Наиболее распространенным методом является One-Hot кодирование, создающее для каждой категории новый бинарный признак. Альтернативным подходом выступает целевое кодирование, при котором категории заменяются средним значением целевой переменной для объектов данной ка-

тегории, что особенно полезно при работе с признаками, имеющими большое количество категорий.[2]

Обработка пропущенных значений:

Пропущенные значения в данных могут существенно ухудшить качество моделей. Существует несколько стратегий обработки пропусков, включая удаление объектов или признаков с пропусками, заполнение статистическими показателями (средним или медианой), а также более сложные методы прогнозирования пропущенных значений с помощью моделей машинного обучения.[3]

Аугментация данных:

Аугментация данных играет особую важную роль при работе с небольшими наборами данных. Для изображений применяются геометрические преобразования, такие как повороты, отражения и масштабирование, а также цветовые преобразования, включая изменение яркости, контраста и насыщенности. Временные ряды обрабатываются с помощью временных сдвигов, добавления случайных колебаний и изменения масштаба значений.[4]

II. ПРАКТИЧЕСКАЯ РЕАЛИЗАЦИЯ МЕТОДОВ ПРЕДОБРАБОТКИ

Реализация методов предобработки в Python осуществляется с использованием популярных библиотек. Scikit-learn предоставляет инструменты для нормализации и стандартизации, работы с категориальными признаками и обработки пропущенных значений. TensorFlow и Keras предлагают специализированные решения для аугментации изображений и встроенной предобработки. Для расширенных задач можно использовать специализированные библиотеки, такие как Albumentations для работы с изображениями или Feature-engine для сложных методов предобработки.[5]

Ключевым аспектом практической реализации является создание воспроизводимых конвейеров обработки данных, что гарантирует согласованное применение преобразований к обучающим и тестовым наборам. Это особенно важно для обеспечения корректной оценки качества моделей.

ЗАКЛЮЧЕНИЕ

Проведенное исследование методов предварительной обработки данных позволяет сделать ряд важных выводов. Качественная предобработка данных является необходимым условием эффективного обучения нейросетевых моделей, существенно влияя как на скорость сходимости, так и на итоговую точность. Различные типы данных требуют специализированных подходов к обработке, что необходимо учитывать при построении конвейеров машинного обучения.

Современные библиотеки предоставляют широкие возможности для реализации сложных схем предобработки, однако их эффективное использование требует глубокого понимания принципов работы методов. Перспективными направлениями даль-

нейших исследований являются разработка адаптивных методов предобработки, создание эффективных алгоритмов для работы с мультимодальными данными и автоматизация выбора оптимальных преобразований.

Список литературы

1. Goodfellow I., Bengio Y., Courville A. Deep Learning. – MIT Press, 2016.
2. Géron A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. – O'Reilly Media, 2019.
3. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning. – Springer, 2009.
4. Chollet F. Deep Learning with Python. – Manning Publications, 2017.
5. Дюк В.А. Обработка данных на компьютере. – СПб.: Питер, 2018.

Рязанцев Никита Дмитриевич, магистрант кафедры информационных технологий автоматизированных систем, n.riazantsev@bsuir.by

Рязанцев Дмитрий Дмитриевич, магистрант кафедры информационных технологий автоматизированных систем, dmitryryaz@mail.ru.

Научный руководитель: Желяк Надежда Александровна, преподаватель кафедры информационных технологий автоматизированных систем БГУИР, доцент, кандидат технических наук, gznadya@gmail.com