

В наши дни есть такие задачи, которые имеют такую сложность, что решить их с помощью алгоритмов либо невозможно, либо требует огромные количества времени, вычислительной мощности и человеческих ресурсов. Для решения данных задач машинное обучение предлагает приближенное, но быстрое решение, основанное на конкретных примерах или данных, а не на формулах, которые покрывают не все случаи.

Цель данной работы - проанализировать класс таких задач, показать, почему машинное обучение с погрешностями лучше, чем точные алгоритмы, привести сравнения.

Что такое аналитическое решение?

Аналитическое решение — это точное выражение ответа через известные математические функции, операции и константы (например, формула корней квадратного уравнения).

Примеры задач, где аналитика не справляется:

- **Нелинейные дифференциальные уравнения** Например: Моделирование турбулентного течения жидкости (уравнения Навье–Стокса)
- **Задачи оптимизации с множеством переменных и ограничений** Например: Управление энергосистемой (оптимизация генерации и распределения электроэнергии)
- **Задачи с неструктуризованными данными** Например: Распознавание лиц, голоса, текста

Почему ML становится предпочтительным способом решения подобных задач?

В условиях, когда аналитическое решение либо отсутствует, либо требует большие вычислительные ресурсы, машинное обучение предлагает самый очень выгодный компромисс: вместо поиска точного ответа строится модель, способная давать достаточно точное решение за малое время. В реальных приложениях часто важнее не теоретическая точность, а обобщающая способность модели и её вычислительная эффективность.

Основную роль играет принцип регуляризации, сформированный А.Н. Тихоновым: если задача некорректна, то следует искать устойчивое приближенное решение, а не точное и нестабильное. Современные методы машинного обучения реализуют эту идею на практике, обеспечивая надежность даже при наличии шума в данных.

Согласно Гасникову и Баранову, "в задачах высокой размерности аналитические и даже классические численные методы теряют эффективность из-за проклятия размерности, тогда как методы машинного обучения, основанные на стохастической оптимизации, демонстрируют масштабируемость и адаптивность". Это особенно актуально для систем, где количество переменных может достигать сотни тысяч, а то и миллионы.

Примеры, где машинное обучение превосходит аналитику

Решение дифференциальных уравнений с помощью нейросетей

Дифференциальные уравнения в частных производных (УЧП) описывают сложные физические процессы — например, как течёт воздух вокруг самолёта или как распространяется тепло в металле. Традиционные методы решения таких уравнений (например, метод конечных разностей) разбивают пространство на мелкую сетку и вычисляют ответ в каждой её точке. Но если задача трёхмерная или зависит от многих переменных, такая сетка становится огромной.

Нейросети предлагают вместо сетки поиск формулы решения напрямую. Они ищут некую гибкую кривую, которая "подстраивается" под уравнение. Это работает в любом пространстве, без разбиения на точки. Такой подход даёт ответ за минуты или секунды, а ошибка обычно не превышает 5–10%. Для инженерных расчётов — это вполне допустимо.

Прогнозирование временных рядов

Классические статистические модели (например, ARIMA) хорошо работают, только если данные не скачут резко и не имеют шумов. Но в реальности данные на подобии продаж в магазине, цен на нефть, спроса на доставку почти всегда нестабильные, шумные и нелинейные.

Машинное обучение, особенно такие методы, как XGBoost, не требует, чтобы данные были «идеальными». Они учатся на истории: что было вчера, в праздники, в грозу — и на основе этого предсказывают будущее. В основном, именно такие модели сейчас используют российские компании для прогнозирования спроса и логистики. Ошибки в 5–15% считаются нормой — зато модель быстро адаптируется к новым условиям и не ломается при изменении тренда.

Распознание изображений и текста

Представим, что мы пытаемся написать четкие правила, по которым компьютер сможет распознать кошку от собаки. Например, если уши заостренные, то это кошка. Но, кошки бывают с опущенными ушами, а собаки могут находиться под таким ракурсом, что ушей вообще не будет видно. Попытки описать это всё общими формулами становятся бесполезными, ведь слишком много исключений и вариантов.

Машинное обучение решает данную задачу иначе. Компьютер сам учится на миллионах примеров. Нейросети смотрят на изображения и постепенно понимают, какие признаки важны. Точно также происходит и с текстом. Нейросети учатся понимать смысл, даже если фраза написана нестандартно.

Такие системы сегодня справляются практически так же хорошо, как люди. Допустим, из 100 изображений модель ошибается в 2-3 случаях — этого хватает, чтобы использовать её в реальной жизни: для диагностики снимков в больнице, распознавания лиц в камерах безопасности или автоматической модерации контента в онлайн-магазинах.

Допустимость погрешности: философия "достаточно хорошего" решения

Переход от идеального решения к концепции "достаточно хорошего" результата — ключевая идея современной прикладной информатики. Тихонов и Арсенин утверждали: "Практическая ценность решения определяется не его математической точностью, а устойчивостью, интерпретируемостью и вычислительной реализуемостью"

Погрешность в несколько процентов может быть оправдана, если она компенсируется:

- Значительным сокращением времени вычислений
- Возможностью обработки больших и шумных данных
- Адаптивностью модели к изменяющимся условиям

Заключение

Машинное обучение в эпоху сложных систем и огромного количества данных становится не заменителем аналитических методов, а инструментом первого выбора для задач, которые раньше нельзя было решить в силу отсутствия времени или ресурсов.

Основные примеры инструментов для решения подобных задач:

- Глубокие нейронные сети (**DNN, CNN, RNN, Transformers**) — для неструктурированных данных.
- Ансамблевые методы (**Random Forest, XGBoost, LightGBM**) — для табличных данных, интерпретируемых.
- Генетические алгоритмы / эволюционные стратегии — для оптимизации сложных функций.
- Reinforcement Learning — для задач принятия решений в динамичной среде.

Ссылки на более глубокие исследования

1. [Mastering the game of Go with deep neural networks and tree search](#)
2. [XGBoost: A Scalable Tree Boosting System](#)
3. [Deep learning](#)

Источники

- Goodfellow, I., Bengio, Y., Courville, A. — Deep Learning (MIT Press, 2016)

- Воронцов К. В. Лекции по машинному обучению.
- Тихонов А. Н., Арсенин В. Я. Методы решения некорректных задач.
- Баранов О. В., Гасников А. В. Методы оптимизации и машинное обучение: курс лекций.