Методы машинного обучения

Лекция 15

Глубокие НС

Интерпретация работы НС

Глубокое обучение (Deep learning)

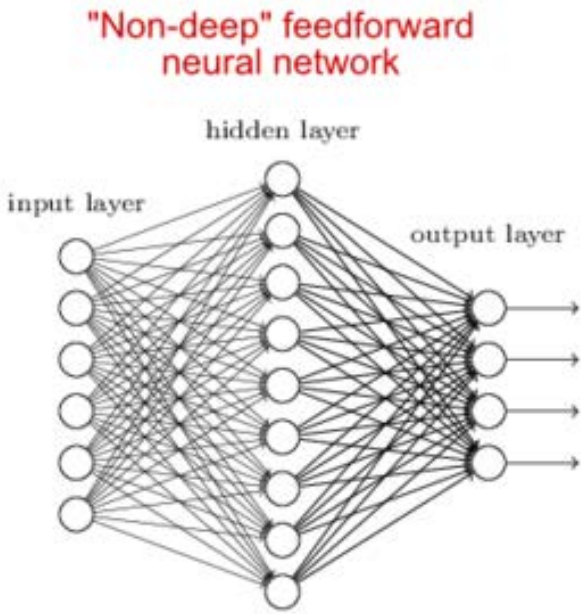
Глубокое обучение — это совокупность методов машинного обучения, основанных на обучении представлениям (*feature/representation learning*), а не специализированных алгоритмах под конкретные задачи, и которые:

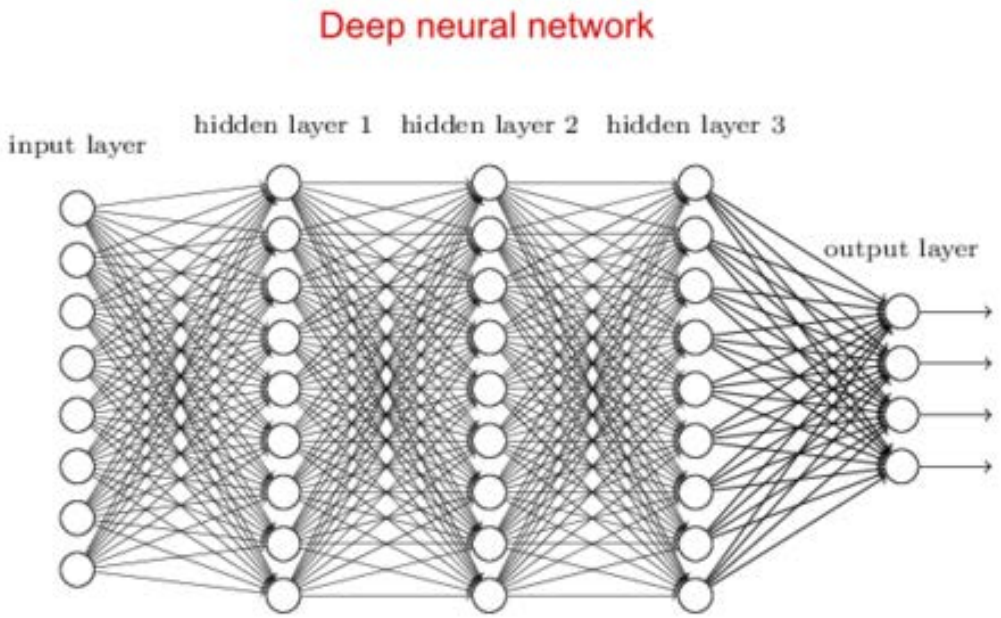
- Используют многослойную систему нелинейных фильтров для извлечения признаков с преобразованиями. Каждый последующий слой получает на входе выходные данные предыдущего слоя;
- Могут сочетать алгоритмы обучения с учителем, с частичным привлечением учителя, без учителя, с подкреплением;
- Формируют в процессе обучения слои выявления признаков (*features*) на нескольких уровнях представлений, которые соответствуют различным уровням абстракции; при этом признаки организованы иерархически — признаки более высокого уровня являются производными от признаков более низкого уровня.

Как правило, глубокое обучение предназначено для работы с большими объемами данных и использует сложные алгоритмы для обучения модели.

Глубокие нейронные сети
Deep Neural Network

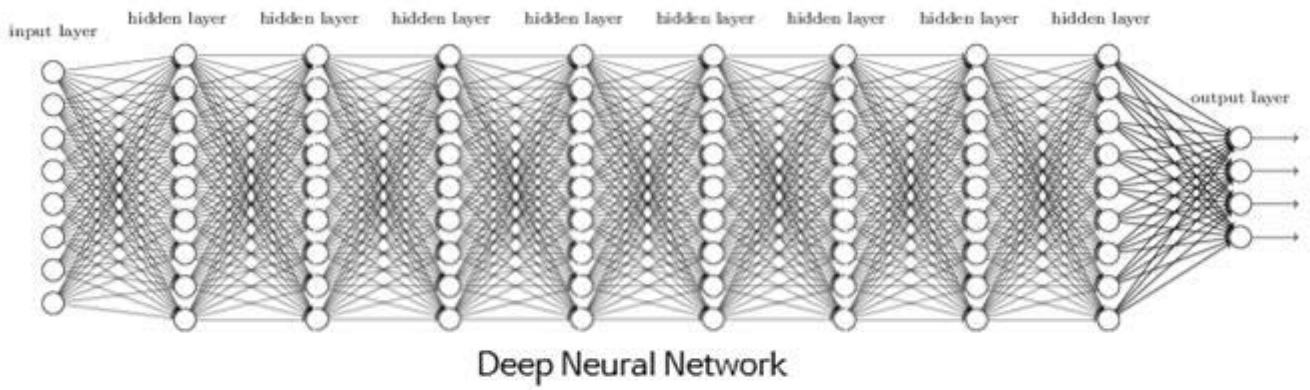
Нейронные сети с числом скрытых слоев большим единицы называются глубокими. Они могут содержать меньшее число нейронов в каждом слое, чем сети с одним скрытым слоем, реализующие то же самое отображение, однако строгой методики сопоставления таких сетей не существует.

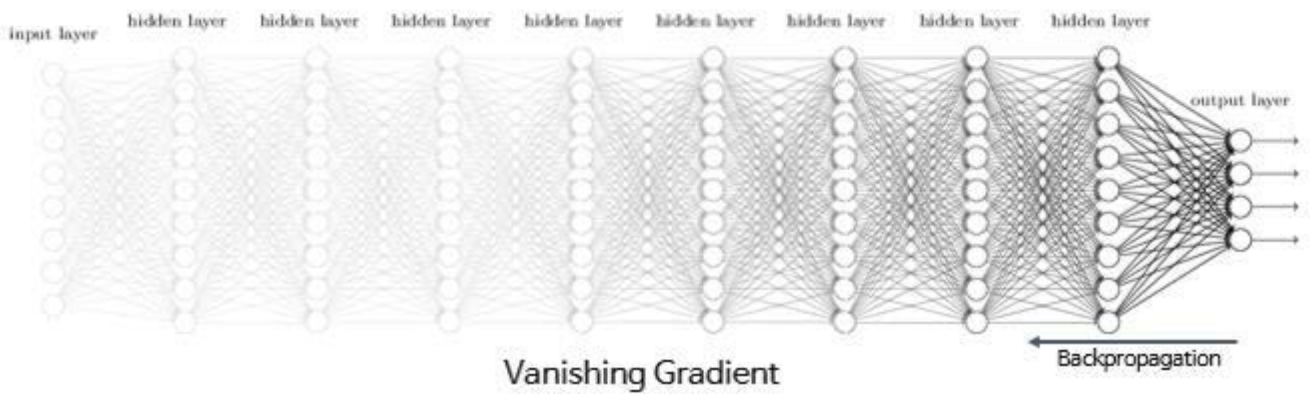




Основные проблемы глубоких полносвязных нейронных сетей

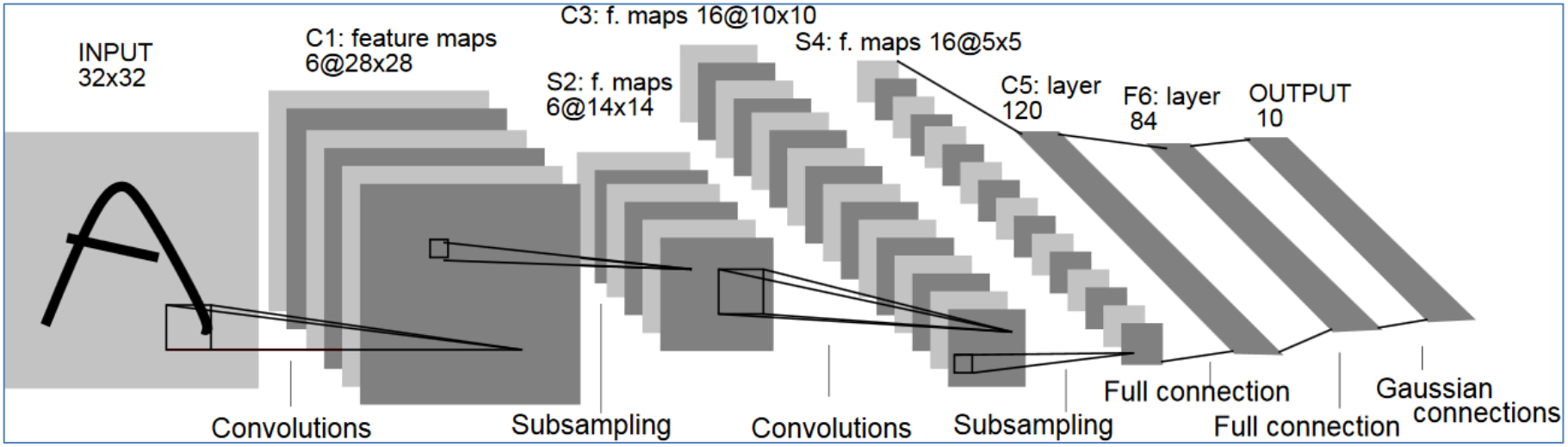
- большое число настраиваемых параметров;
- затухающими градиентами при обучении с помощью алгоритма обратного распространения ошибки (Backpropagation);
- полностью игнорируется топология локального взаиморасположения входных величин.





Свёрточная нейронная сеть

- Неокогнитрон (*Neocognitron*) —это иерархическая многослойная нейронная сеть сверточного типа, предложена Кунихико Фукусимой в 1980 году, способная к робастному распознаванию образов, обучаемая по принципу «обучение без учителя».
- В 1998 году Яну Лекуну удалось использовать алгоритм обратного распространения ошибки для обучения глубокой сверточной нейронной сети для решения задачи распознавания рукописных ZIP-кодов.

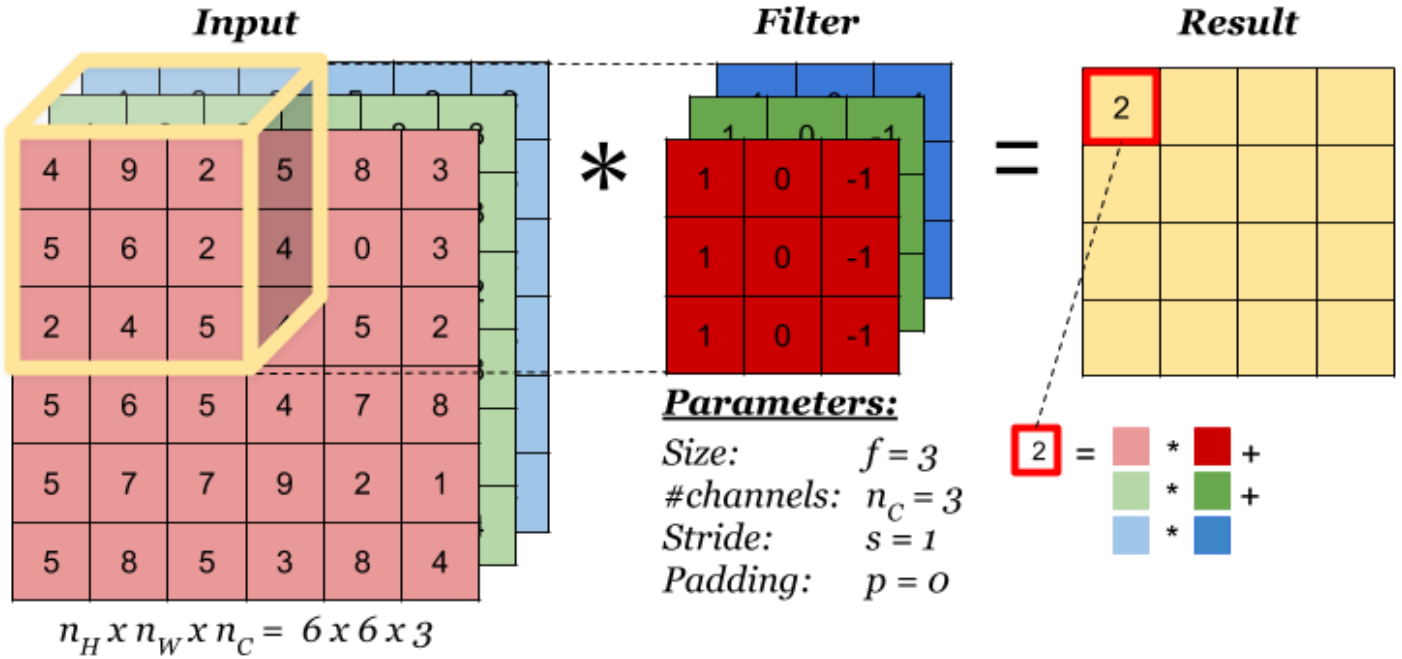


Архитектура сверточной нейронной сети LeNet-5, использованной для распознавания цифр.

Операция свертки (Convolution)

Операция свертки (Convolution) аналогична использованию малого фильтра, например, размером 3 × 3 с шагом 1 ко всему изображению. Применение одного такого фильтра фактически является построением некой карты нахождения определенного признака на изображении. Каждый фильтр соответствует одному нейрону.

Свертка - операция над парой матриц *A (размера $n_x \times n_y$) и B (размера $m_x \times m_y$), результатом которой является матрица C=A∗B размера ($n_x - m_x + 1) \times (n_y - m_y + 1$). Каждый элемент результата вычисляется как скалярное произведение матрицы B и некоторой подматрицы A такого же размера (подматрица определяется положением элемента в результате). То есть,* $C_{i,j} = \sum_{u=0}^{m_x-1} \sum_{v=0}^{m_y-1} A_{i+u,j+v} B_{u,v}$.

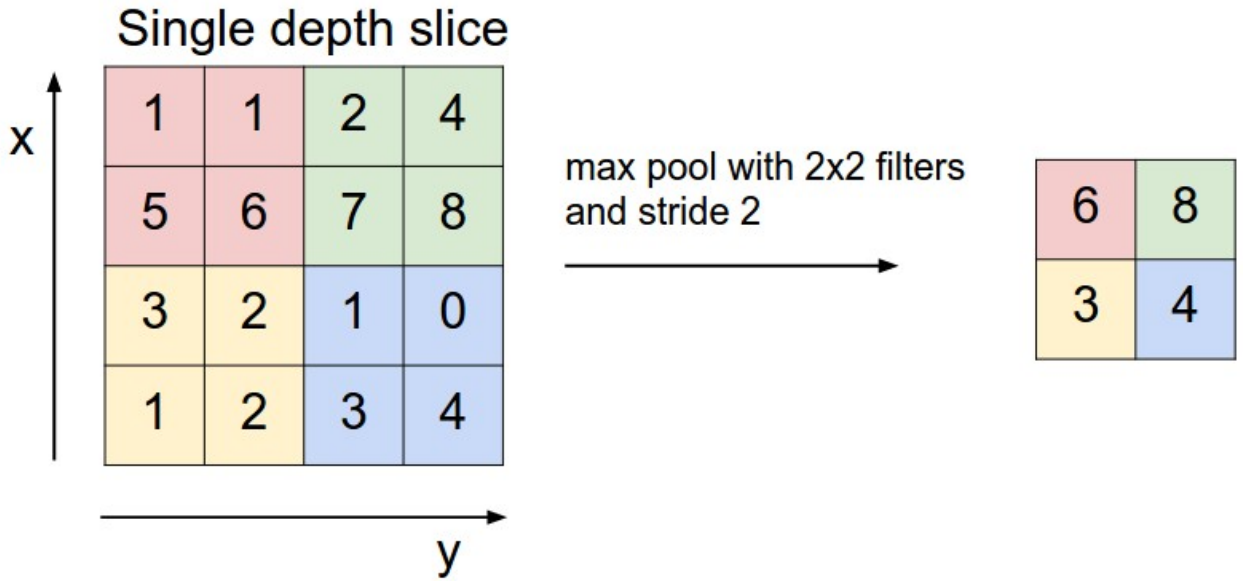


Пулинг / сабсемплинг (Subsampling)

Пулинг (или сабсемплинг) используется для уменьшения размерности. Он основан на том факте, что изображения обладают свойством локальной скоррелированности пикселей, т.е. соседние пиксели, как правило, не сильно отличаются друг от друга. Таким образом, если из нескольких соседних пикселей получить какой-либо агрегат, то потери информации будут незначительными. Рекомендуемый размер объединения составляет 2 × 2.

Основные цели пулингового слоя:

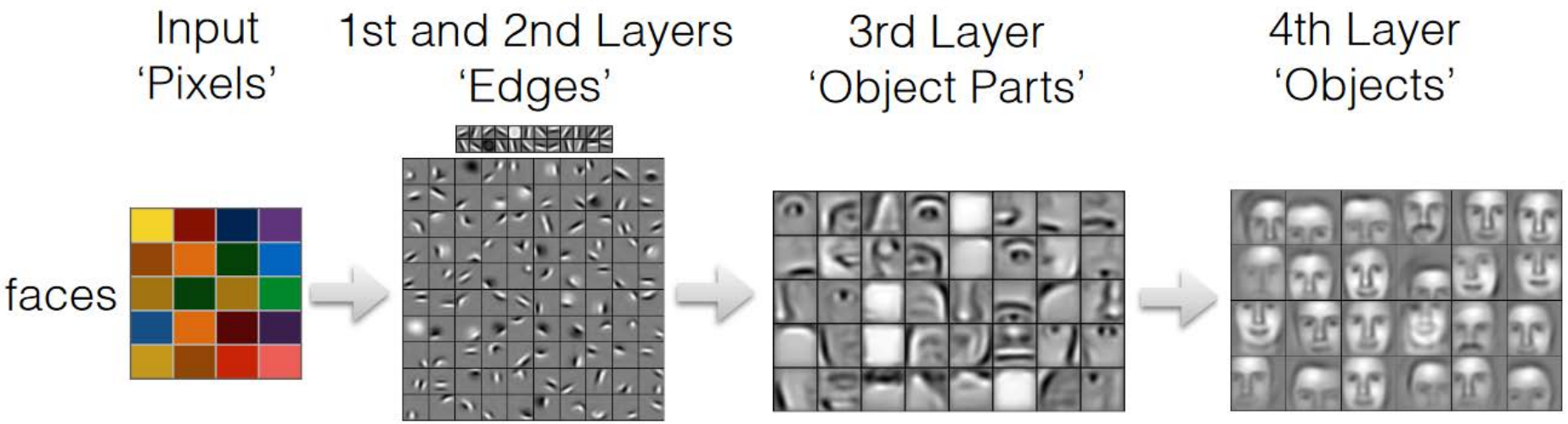
- уменьшение изображения, чтобы последующие свертки оперировали над большей областью исходного изображения;
- увеличение инвариантности выхода сети по отношению к малому переносу входа;
- ускорение вычислений.



Общий принцип функционирования

На вход сети подается изображение и к нему последовательно применяются операции свертки (Convolution), которые чередуются с пулингом (Subsampling) несколько раз, а затем полученные данные проходят через набор полносвязных слоев.

Тем самым сверточные нейронные сети, позволяют создавать модели, состоящие из множества слоев, которые способны обучаться представлениям данных с различными уровнями абстракции.



Ключевые этапы реализации глубоких сетей

Можно выделить восемь ключевых этапов для реализации глубоких сетей:

- Аугментация данных.
- Предобработка данных.
- Инициализация.
- Выбор функций активации.
- Процесс обучения.
- Регуляризация.
- Визуализация.
- Ансамбли глубоких сетей.

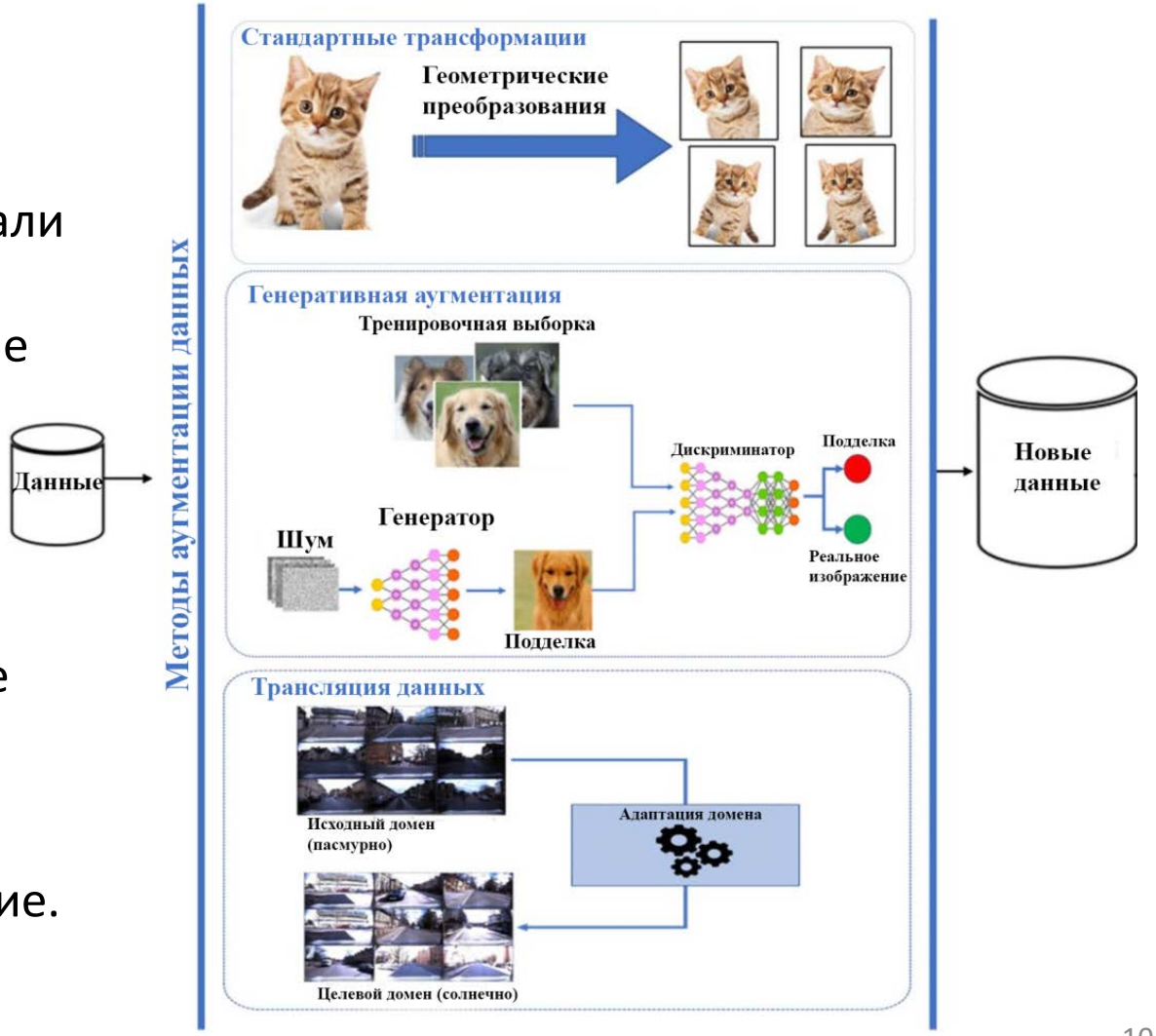
Аугментация данных

Аугментация данных – это методика создания дополнительных обучающих данных из уже имеющихся.

Самыми популярными (классическими) методами аугментации являются:

- отражение по горизонтали (horizontal flip);
- случайное кадрирование (random crop);
- изменение цвета (color jitter).

Возможно применение различных комбинаций, например, одновременно выполнять поворот и случайное масштабирование.

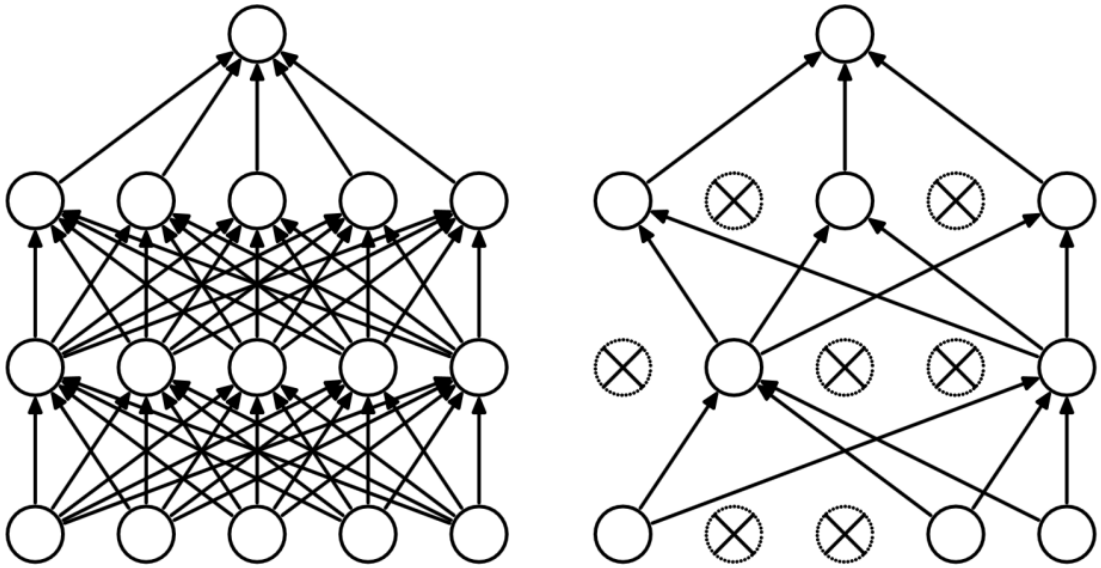


Регуляризация

Регуляризация позволяет осуществлять контроль емкости нейронной сети, что способствует предотвращению переобучения. Основная идея заключается в учете дополнительной информации, которая имеет вид штрафа за сложность модели. Также регуляризация позволяет ограничивать значения весовых коэффициентов и осуществлять отбор наиболее важных факторов, которые сильнее всего влияют на результат.

Основные методы регуляризации применительно к нейронным сетям: L1 и L2-регуляризация, ограничения нормы вектора весов, дропаут.

Dropout регуляризация нейроной сети:



Извлечение правил из полносвязной нейронной сети в задачах классификации

Под извлекаемой логической закономерностью будем понимать легко интерпретируемое правило, выделяющее из обучающей выборки достаточно много объектов какого-то одного класса и практически не выделяющее объекты остальных классов. Правила, выражающие закономерности, формулируются на языке логических предикатов первого порядка вида:

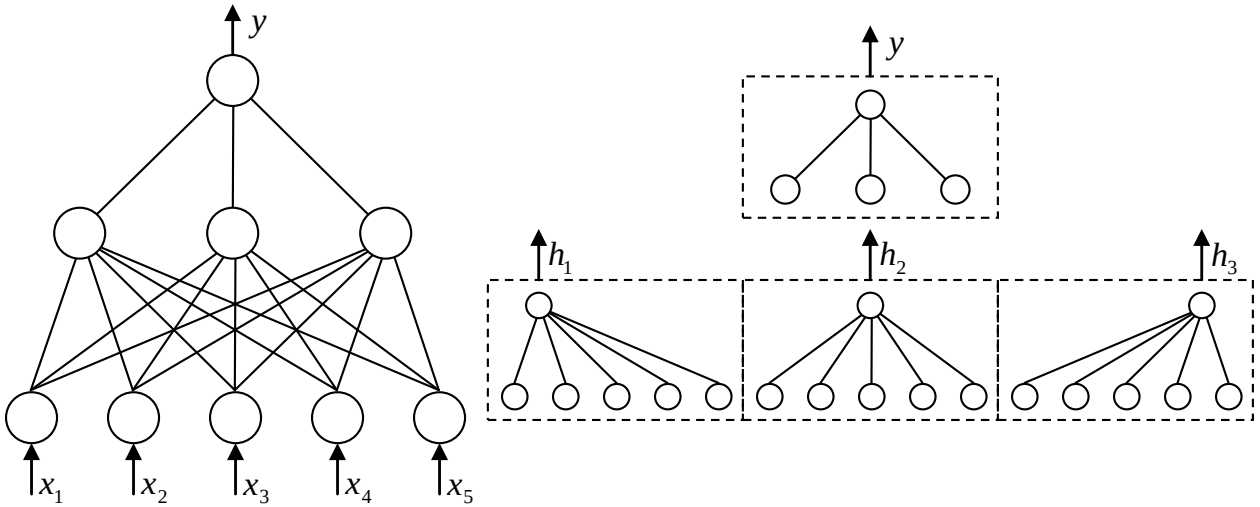
ЕСЛИ *(условие1) И (условие2) И … И (условиеN)* ***ТО*** *(вывод).*

Можно выделить следующие основные подходы:

- извлечение локальных правил из совокупности простейших однослойных сетей, на которые разделяется построенная многослойная модель;
- построение глобальных правил, которые характеризуют классы на выходе непосредственно через значения входных параметров.

Извлечение локальных правил из обученных нейронных сетей

Основные этапы метода NeuroRule:
Этап 1. Обучение нейронной сети.
Этап 2. Прореживание нейронной сети.
Этап 3. Подготовка к извлечению правил,
кодирование признаков классифицируемых объектов.
Этап 4. Извлечение правил.

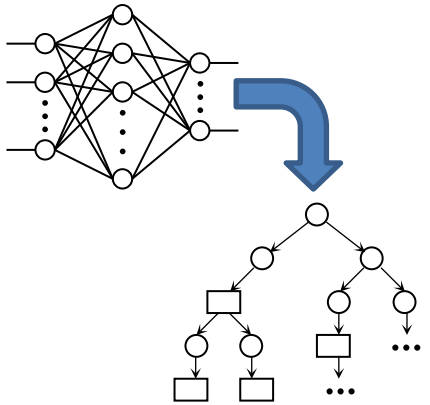


Основным недостатком является наличие жестких ограничений на архитектуру нейросети, число элементов, связей и вид функций активации, т.е. отсутствие универсальности и масштабируемости.

Извлечение глобальных правил из обученных нейронных сетей

Построение глобальных правил, которые характеризуют классы на выходе непосредственно через значения входных параметров.

Данный подход осуществляет построение дерева решений на основе знаний, заложенных в обученную нейросеть, причем достаточно того, что сеть является неким «черным ящиком» или «Оракулом/Экспертом», которому можно задавать вопросы и получать от него ответы.

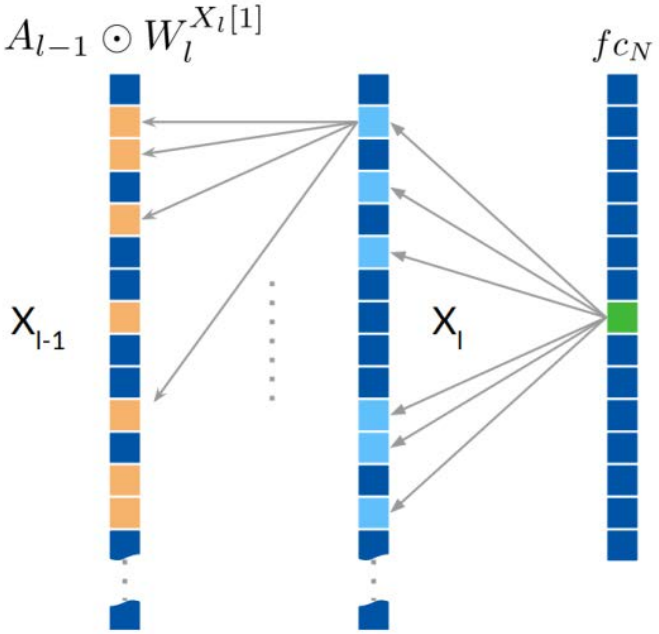


- Достоинство заключается в обобщающей способности искусственных нейронных сетей, что позволяет получать более простые деревья решений. Использование такого «Эксперта» позволяет компенсировать недостаток данных, наблюдающийся при построении деревьев решений на нижних уровнях.
- Недостаток заключается в отсутствии прозрачности внутреннего функционирования сети.

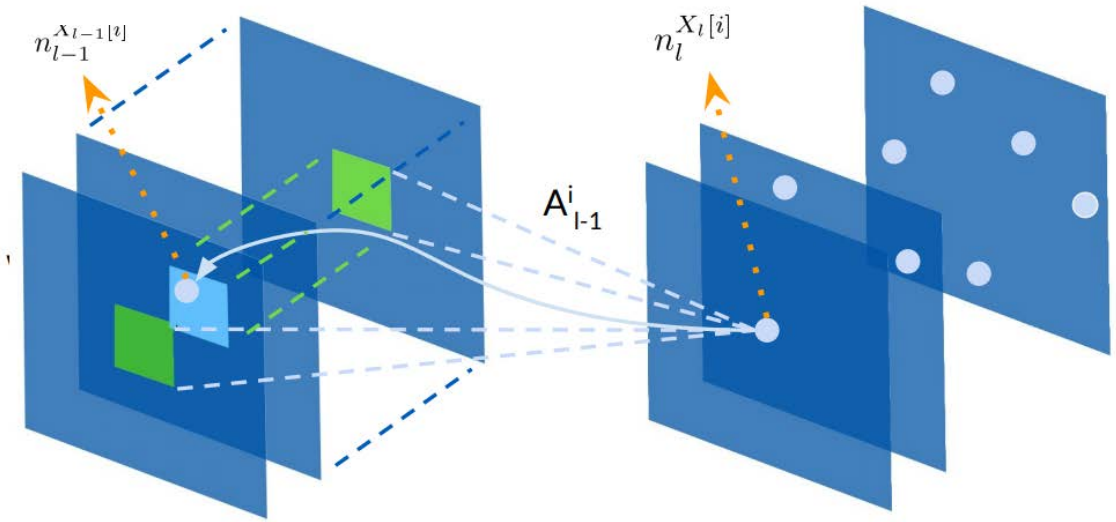
Дополнительным важным направлением является оценка чувствительности влияния значений входных параметров на выход сети. Для этого значения выбранного входа варьируются в области его определения, в то время как остальные параметры остаются фиксированными и отслеживаются изменения в выходе сети. Знания, полученные из этой формы анализа, могут быть представлены в виде таких правил, как:

«ЕСЛИ X уменьшается на 5%, ТО Y увеличивается на 8%».

Интерпретация работы нейронных сетей - Обратное прохождение сигнала

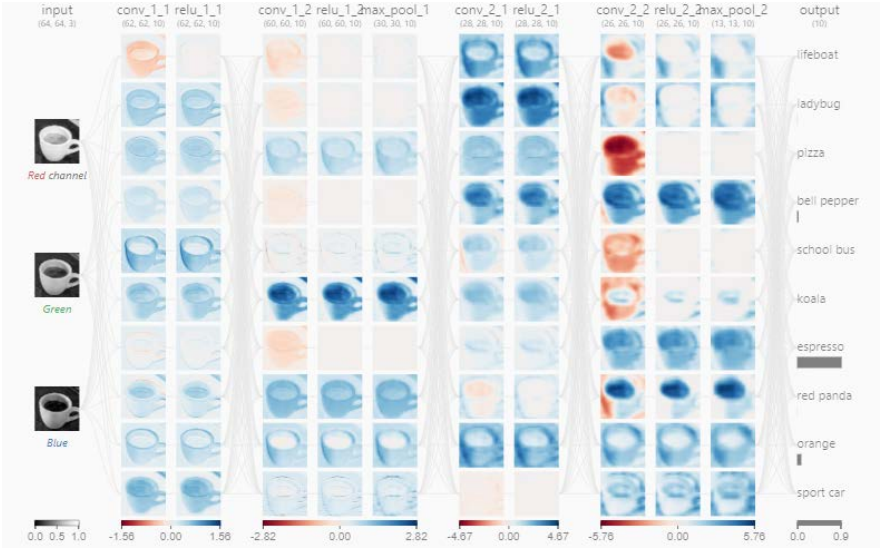


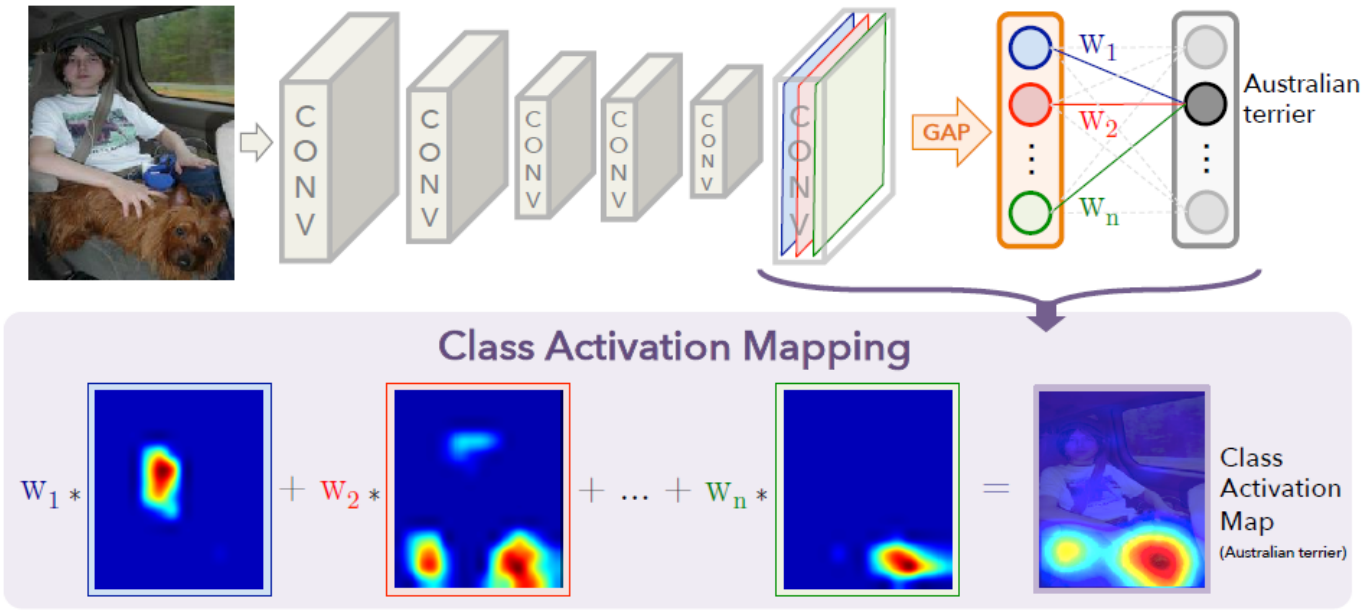
- Обратное прохождение полносвязанных слоев.
- Обратное прохождение сверточных слоев.



Обобщение и интерпретация функционирования сверточных слоев

Одним из подходов, для обобщения и интерпретации результатов функционирования сверточных слоев является метод Class Activation Mapping (CAM), который представляет собой взвешенную карту активации, созданную для каждого изображения, что помогает определить область, на которую акцентирует внимание НС, при классификации изображения.

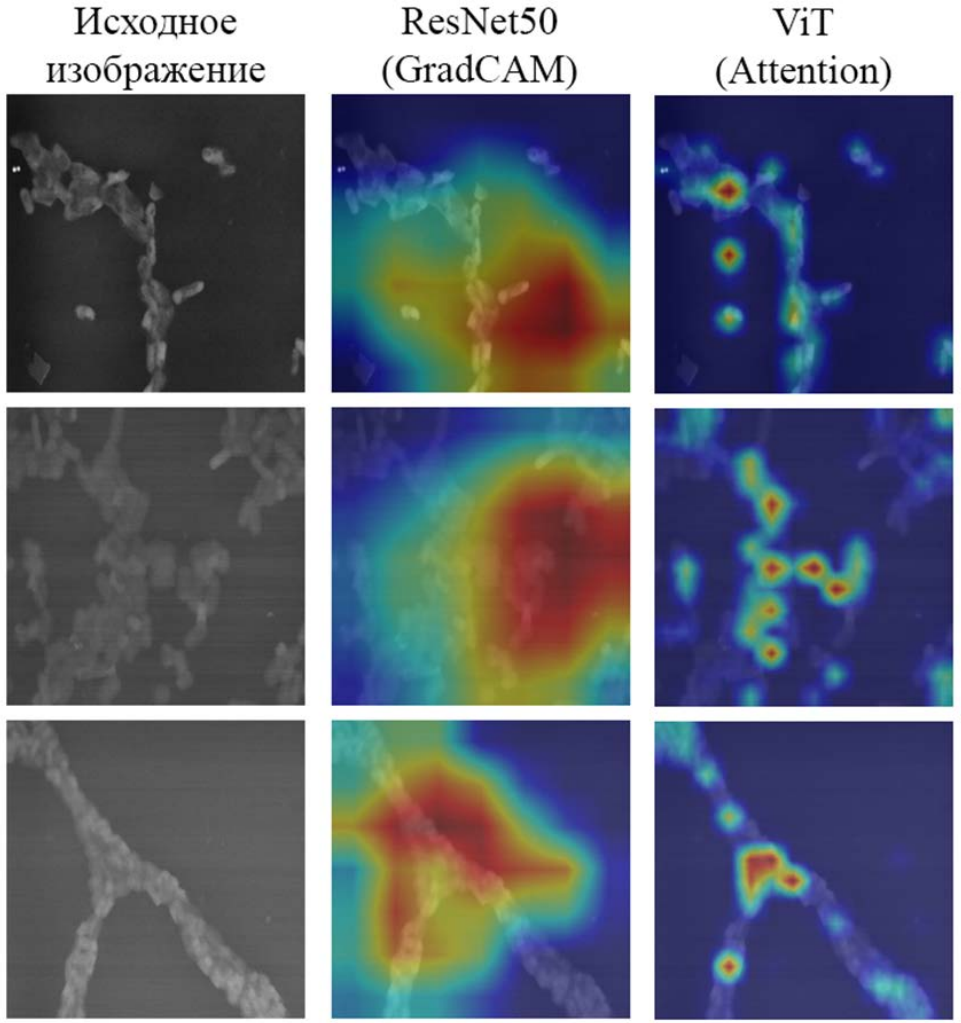




Методы интерпретации GradCAM и Vision Transformer

Интерпретируемость двух популярных архитектур глубокого обучения — ResNet50 и Vision Transformer (ViT-224) — при задаче классификации микроскопических изображений бактерий. Представлена работа встроенных карт-внимания Vision Transformer и пост-интерпретация с помощью Grad-CAM для ResNet50. Показаны тепловые карты с выделенными зонами с наибольшим влиянием на прогноз модели.

Механизм внимания демонстрирует более сконцентрированные участки, однако в обоих способах интерпретации работы моделей наблюдаются фоновые зоны с высокой значимостью на прогноз модели. Это свидетельствует, что фоновый шум может иметь влияние на формирование прогноза, что в целом негативно влияет на способность генерализации моделей.



«Доверенный ИИ»

Объединяет программные инструменты и методики для противодействия принципиально новым угрозам, возникающим на всех этапах жизненного цикла соответствующих технологий:

- Доверенные фреймворки и библиотеки машинного обучения.
- Инструменты проверки наличия аномалий в наборах данных.
- Инструменты оценки устойчивости обученных моделей к атакам.
- Инструменты для повышения доверия к предобученным моделям.
- Методы защиты моделей от атак на этапе эксплуатации.
- Методы объяснения моделей.
- Методы обнаружения дрейфа данных.
- Методы выявления предвзятости моделей.

ИСП РАН (https://www.ispras.ru/ai-center/)

Угрозы ИИ

Страхи перед ИИ

- ИИ принимает решения, которые невозможно объяснить или предсказать
- ИИ может ошибаться, а ответственность за эти ошибки размыта
- ИИ может быть предвзятым и манипулировать мнением
- ИИ собирает и использует персональные данные без явного согласия
- ИИ внедряется везде и его использование может выйти из-под контроля
- ИИ эволюционирует и изменяет алгоритмы без внешнего контроля

Конкретные риски

- Потеря контроля над критическими инфраструктурами
- Финансовые потери из-за ошибок ИИ
- Утечки и неправомерное использование данных
- Репутационные риски из-за ошибок ИИ
- Юридическая ответственность за решения ИИ на пользователе не способном управлять ИИ
- Нарушение конфиденциальности и сбор данных без согласия
- Потеря контроля над личной информацией
- Манипуляция сознанием и поведенческая реклама
- Опасность автоматизированных решений без возможности апелляции

Атаки на ИИ

С 2019 по 2024 год было опубликовано около 17000 научных статей, которые описывают состязательные атаки на ИИ.

По данным ресурса https://owasp.org/www-project-machine-learning-security-top-10/

Top 10 Machine Learning Security Risks:

- ML01:2023 Input Manipulation Attack
- ML02:2023 Data Poisoning Attack
- ML03:2023 Model Inversion Attack
- ML04:2023 Membership Inference Attack
- ML05:2023 Model Theft
- ML06:2023 AI Supply Chain Attacks
- ML07:2023 Transfer Learning Attack
- ML08:2023 Model Skewing
- ML09:2023 Output Integrity Attack
- ML10:2023 Model Poisoning

- Инъекция (Prompt Injection)

  Атака по смыслу схожа с sql-инъекцией. Но на текущий момент, это уже что-то вроде социальной инженерии для ИИ. В контексте атак на ИИ её относят к jailbreak.
- Инфекция (Infection)

  Заражение вредоносным ПО. В OWASP один из примеров – это атака на цепочку поставок.
- Уклонение (Evasion)

  Данные на вход для ИИ модифицируют незначительно. Нарушитель стремиться заставить ИИ ошибиться.
- Отравление ИИ (Poisoning)

  Отравление данных, которые поступают в ИИ. Доказано, что достаточно внести 0,001% ошибок в данные, чтобы результаты оказались аномальными и неверными.
- Извлечение (Extraction)

  Допустим, кто-то иногда делает запросы в модель и медленно собирает из нее данные. В конце концов атакующий соберет достаточно информации, чтобы восстановить данные на своей стороне.

Спасибо за внимание