Глава 3.2. Искусственный интеллект

3.2.1. Термины и определения

В английском языке термин Intelligence не имеет антропоморфной окраски, которую он имеет в традиционном русском переводе. Intelligence означает «умение рассуждать разумно», а вовсе не антропоморфный «интеллект»; для антропоморфного русскоязычного «интеллекта» имеется английский аналог “intellect”. Ложный налёт антропоморфности, или “антропоморфный эффект”, осложняет четкое понимание свойств и области применимости искусственного интеллекта. Имеется принципиальная разница между физическими параметрами искусственного интеллекта и мозга человека. Например, Azure AI потребляет порядка 10-12 МВт электроэнергии и занимает огромный машинный зал, в то время как мозг потребляет в среднем 20 Вт и занимает 1300 куб. см. пространства.

Приведем некоторые варианты определений термина Иску́сственный интелле́кт (ИИ; англ. Artificial Intelligence, AI). Определения разбиты на две группы - рекомендованные и не рекомендованные. Рекомендованные определения являются строгими и не содержат антропоморфных манипуляций. Не рекомендованные определения не являются строгими и в той или иной степени являются манипулятивными.

Рекомендованные определения

Искусственный интеллект — это процесс и результат процесса автоматизации построения алгоритмов и их отображения на архитектуру ЭВМ в виде программы.

Г.И. Марчук. Методы вычислительной математики

Искусственный интеллект — это аппроксимация функции отображения одного множества в другое. При этом функция отображения, первое и второе множества существуют до момента обучения алгоритма аппроксимации функции. Математическим обоснованием существования сходимости и устойчивости таких аппроксимаций для конечномерных множеств является теорема А.Н. Тихонова о неподвижных точках отображения на упорядоченных ограниченных множествах. Задача построения аппроксимации функции отображения бесконечного множества на конечное относится к категории некорректных задач, решение у которой либо отсутствует, либо множественно, либо неустойчиво.

А.И. Аветисян. Академик, Директор ИСП РАН

Не рекомендованные определения

Искусственный интеллект — свойство интеллектуальных систем выполнять творческие функции, которые традиционно считаются прерогативой человека.

Википедия

Искусственный интеллект – это область компьютерных наук, которая занимается созданием интеллектуальных машин и программ, способных выполнять задачи, традиционно ассоциируемые с человеческим интеллектом, такие как распознавание образов, обработка естественного языка, принятие решений и обучение. ИИ включает в себя широкий спектр подходов, включая машинное обучение, нейронные сети, экспертные системы и многое другое. Цель ИИ – создание систем, которые могут адаптироваться к новым ситуациям, учиться на опыте и принимать решения без прямого вмешательства человека.

Большая языковая модель GigaChat

Термин «искусственный интеллект» представляет множество дисциплин в области программного обеспечения, логики, вычислений и философии, цель которых — создание компьютеров, способных выполнять функции, ранее свойственные только человеку. В качестве примера таких функций можно назвать восприятие значения, написанного или произнесенного, обучение, распознавание мимики и так далее. Направление развития искусственного интеллекта имеет долгую историю. Такие ранние технологические прорывы, как оптическое распознавание символов, теперь являются обыденным явлением.

HPE

Искусственный интеллект позволяет компьютерам обучаться на собственном опыте, адаптироваться к задаваемым параметрам и выполнять те задачи, которые раньше были под силу только человеку. В большинстве случаев реализации ИИ — от компьютерных шахматистов до беспилотных автомобилей — крайне важна возможность глубокого обучения и обработки естественного языка. Благодаря этим технологиям компьютеры можно «научить» выполнению определенных задач с помощью обработки большого объема данных и выявления в них закономерностей.

SAS

3.2.2. История ИИ

История искусственного интеллекта начинает отсчет с древних времен, когда философы размышляли, как можно искусственно механизировать человеческое мышление и управлять им с помощью разумных «нечеловеческих» машин. Мыслительные процессы, которые подогревали интерес к ИИ, зародились, когда классические философы, математики и логики рассмотрели возможность манипулирования символами (механически), что в конечном итоге привело к изобретению программируемого цифрового компьютера, компьютера Атанасова-Берри (ABC) в 1940-х годах. Это конкретное изобретение вдохновило ученых на продвижение идеи создания «электронного мозга» или существа с искусственным интеллектом.

Математик Алан Тьюринг среди прочего предложил тест, который измерял способность машины воспроизводить человеческие действия в степени, неотличимой от человеческого поведения. Позднее в том же десятилетии область исследований ИИ была основана во время летней конференции в Дартмутском колледже в середине 1950-х годов, где Джон Маккарти, ученый-компьютерщик и когнитивист, ввел термин «искусственный интеллект».

Начиная с середины XX века многие ученые, программисты, логики и теоретики способствовали укреплению современного понимания искусственного интеллекта в целом. С каждым новым десятилетием появлялись инновации и открытия, которые меняли фундаментальные знания людей в области искусственного интеллекта и того, как исторические достижения превратили ИИ из недостижимой фантазии в осязаемую реальность для нынешнего и будущих поколений.

1940-1960: Рождение ИИ на волне кибернетики

Период между 1940 и 1960 годами был отмечен сочетанием технологических достижений (ускорителем которых стала Вторая мировая война) и желанием понять, как объединить работу машин и органических существ. Для Норберта Винера, пионера в области кибернетики, целью было объединить математическую теорию, электронику и автоматизацию в «единую теорию управления и коммуникации, как в животных, так и в машинах». Незадолго до этого первая математическая и компьютерная модель биологического нейрона (формального нейрона) была разработана Уорреном Маккалоком и Уолтером Питтсом еще в 1943 году.

В начале 1950 года Джон фон Нейман и Алан Тьюринг еще не создали термин ИИ, но были отцами-основателями лежащей в его основе технологии: они перешли от компьютеров в десятичной логике XIX века (которая, таким образом, имела дело со значениями от 0 до 9) к машине с двоичной логикой (которая полагается на булеву алгебру, имея дело с цепочками из 0 или 1). Таким образом, два исследователя формализовали архитектуру наших

современных компьютеров и продемонстрировали, что это — универсальная машина, способная выполнять то, что запрограммировано.

Тьюринг, с другой стороны, впервые поднял вопрос о возможном интеллекте машины в своей знаменитой статье 1950 года «Вычислительные машины и интеллект» и описал «игру в имитацию», где человек должен иметь возможность различать в диалоге телетайпа, разговаривает ли он с человеком или с машиной. Какой бы противоречивой ни была эта статья («тест Тьюринга»), ее часто будут цитировать как источник вопросов о границе между человеком и машиной.

Термин «ИИ» можно отнести к Джону Маккарти из Массачусетского технологического института, который Марвин Мински (Университет Карнеги-Меллона) определяет как создание компьютерных программ, способных выполнять задачи, которые в настоящее время более удовлетворительно выполняются людьми, потому что требуют умственных процессов высокого уровня, таких как перцептивное обучение, организация памяти и критическое мышление.

Летняя конференция 1956 года в Дартмутском колледже (финансируемая Институтом Рокфеллера) считается основателем этой дисциплины. Как ни странно, стоит отметить великий успех того, что было не конференцией, а скорее семинаром. Только шесть человек, включая Маккарти и Мински, постоянно присутствовали на протяжении всей этой работы (которая в основном опиралась на разработки, основанные на формальной логике).

Статья 1963 года Рида К. Лолора, члена Калифорнийской коллегии адвокатов «Что могут делать компьютеры: анализ и прогнозирование судебных решений» отмечает, что популярность технологий снизилась. Герберт Саймон, экономист и социолог, в 1957 году предсказал, что ИИ сумеет победить человека в шахматах в следующие 10 лет, но затем ИИ вступил в первую «зиму». Видение Саймона подтвердилось... 30 лет спустя.

В 1968 году В. Н. Вапником и А. Я. Червоненкисом, сотрудниками Института проблем управления им. Трапезникова, было опубликовано доказательство фундаментального результата — условий равномерной сходимости частот к вероятностям по классу событий. Аналогичные условия были получены для равномерной сходимости средних к математическим ожиданиям по семейству случайных величин. В настоящее время эти результаты широко известны во всём мире, а понятие размерности Вапника-Червоненкиса (VC-dimension) прочно вошло в международный научный лексикон.

1970-1980: Повышение интереса к ИИ

В 1968 году Стэнли Кубрик снял фильм «Космическая одиссея 2001 года», в котором компьютер HAL 9000 суммирует в себе всё многообразие этических вопросов, поставленных

ИИ: будет ли он представлять собой высокий уровень сложности, благо для человечества, или опасность? Воздействие фильма, естественно, не было научным, но оно способствовало популяризации темы, как и писатель-фантаст Филип К. Дик, который никогда не переставал задаваться вопросом: испытают ли однажды машины эмоции.

Именно с появлением первых микропроцессоров в конце 1970 года ИИ снова взлетел и вступил в золотой век экспертных систем. Путь движения вперед был фактически открыт в Массачусетском технологическом институте в 1965 году с помощью DENDRAL (экспертная система, специализирующаяся на молекулярной химии) и в Стэнфордском университете в 1972 году с MYCIN (система, специализирующаяся на диагностике болезней крови и лекарствах, отпускаемых по рецепту). Эти системы были основаны на «машине вывода», которая была запрограммирована как логическое зеркало человеческого рассуждения. Вводя данные, «движок» давал ответы высокого уровня знаний. В конце 1980-х - начале 1990-х годов повальное увлечение снова прекратилось. На самом деле программирование таких знаний потребовало больших усилий, и при программировании от 200 до 300 правил возникал эффект «черного ящика»: было непонятно, как именно рассуждала машина . Таким образом, разработка и обслуживание стали чрезвычайно проблематичными и, кроме того, решались многими другими менее сложными и менее дорогими способами. Следует напомнить, что в 1990-е годы термин «искусственный интеллект» почти стал «табу», и в университетский язык даже вошли более скромные его вариации, такие как «продвинутые вычисления».

В 1971 г. В. Н. Вапник и А. Я. Червоненкис обосновали сходимость методов обучения, основанных на минимизации эмпирического риска, что дает возможность получить оценку скорости сходимости алгоритмов машинного обучения. В частности, к таким алгоритмам относятся методы построения кусочно-линейных решающих правил, минимизирующих число ошибок на материале обучения. Поскольку одним из формальных средств, реализующих такие кусочно-линейные правила, являются нейронные сети, то эта теория использовалась во всём мире для анализа работы нейронных сетей. Разработанные В. Н. Вапником и А. Я. Червоненкисом методы решения этой задачи получили название методов структурной минимизации риска. В настоящее время они широко применяются в задачах распознавания образов, восстановления регрессионных зависимостей и при решении обратных задач физики, статистики и других научных дисциплин.

В 1974 году произошло значимое событие: А. И. Галушкиным впервые был описан метод обратного распространения ошибки (англ. backpropagation). Это итеративный градиентный алгоритм, который используется при обновлении весов многослойного перцептрона с целью минимизации ошибки и получения желаемого выхода.

1990-2000: Время кропотливой работы

В 1990-х годах произошел значительный прогресс в области ИИ и машинного обучения. Важным достижением стало развитие алгоритмов обратного распространения ошибки, которые позволили эффективно обучать многослойные нейронные сети. Это значительно улучшило их возможности в распознавании образов и обработке естественного языка.

Были разработаны: алгоритм опорных векторов (SVM), глубокие нейронные сети (DNN), сверточные нейронные сети (CNN) и рекуррентные нейронные сети (RNN). Эти методы позволили значительно улучшить способности ИИ в распознавании образов, обработке естественного языка и принятии решений на основе больших объемов данных. CNN стали основой для современных систем компьютерного зрения, позволяя эффективно распознавать и классифицировать изображения. RNN, в свою очередь, позволили обрабатывать последовательные данные, такие как текст или речь. Это позволило значительно улучшить возможности компьютерного зрения и расширить его применение в различных областях, таких как медицина, транспорт, безопасность и промышленность.

Были разработаны методы обучения без учителя, которые позволили машинам обучаться на основе данных без явного предоставления правильных ответов. Это открыло новые возможности для анализа больших объемов данных, где разметка данных может быть сложной или невозможной задачей.

Робототехника стала более доступной и распространенной, что привело к созданию более совершенных и автономных роботов. Это стало возможным благодаря развитию алгоритмов управления, датчиков и исполнительных механизмов. Автономные роботы начали использоваться в различных областях, таких как производство, логистика, сельское хозяйство и обслуживание.

Продолжалось развитие экспертных систем, которые стали более сложными и способными решать более сложные задачи. Экспертные системы используют знания экспертов для решения проблем в определённой области. Развитие экспертных систем позволило автоматизировать процесс принятия решений в различных областях, таких как медицина, юриспруденция, финансы и производство.

Интернет получил в исследовательской среде более широкое распространение, что позволило исследователям и разработчикам обмениваться идеями и данными. Это способствовало быстрому прогрессу в области ИИ и машинного обучения.

В 1997 году произошло значимое событие: шахматный движок Deep Blue, разработанный IBM, одержал победу над чемпионом мира по шахматам Гарри Каспаровым. Это событие привлекло внимание общественности к растущим способностям ИИ.

2000-2010: Эпоха социальных сетей

В период с 2000 по 2010 годы в области ИИ отмечается новый расцвет, основанный на больших данных и вычислительных мощностях, а также значительных достижениях. Перечислим наиболее значимые достижения:

- Появление новых алгоритмов машинного обучения. В 2000-х годах были разработаны и усовершенствованы такие алгоритмы, как градиентный спуск, стохастический градиентный спуск и алгоритмы оптимизации.
- В это десятилетие активно развивается глубокое обучение, и его практическое применение становится более распространенным. Это привело к значительному прогрессу в области распознавания образов, обработки естественного языка и других областях.
- Значительно усовершенствованы сверточные нейронные сети (CNN), которые стали основой для многих современных систем компьютерного зрения. Также значительно усовершенствованы рекуррентные нейронные сети (RNN), которые позволили обрабатывать последовательные данные, такие как текст и речь.
- Началось активное исследование и разработка генеративных моделей, таких как генеративно-состязательные сети (GANs), которые позволяют создавать новые изображения, звуки и тексты, имитирующие реальные данные.
- Произошел значительный прогресс в области обработки естественного языка (NLP), включая разработку алгоритмов машинного перевода, чат-ботов и систем распознавания речи.
- Робототехника продолжила развитие, были созданы более сложные и автономные роботы, способные выполнять различные задачи в различных средах.
- Экспертные системы также продолжили развиваться, и их применение стало более широким, включая использование в медицине, юриспруденции и других областях.
- Облачные вычисления стали более доступными и распространенными, что позволило исследователям и разработчикам использовать большие объемы вычислительных ресурсов для обучения и тестирования ИИ-моделей.
- Было создано множество открытых источников данных и библиотек алгоритмов, что облегчило доступ к данным и инструментам для исследователей и разработчиков.

2010-2020: Эпоха больших данных и бум ИИ

В период с 2010 по 2020 год в области ИИ произошли значительные события, которые оказали существенное влияние на развитие ИИ, приведшие в результате к новому буму ИИ.

Отметим ключевые события этой эпохи:

1. Глубокое обучение стало доминирующим подходом в машинном обучении, что привело к значительному прогрессу в распознавании образов, обработке естественного языка и других областях.
2. В 2012 году Google X (поисковая лаборатория Google) заставила ИИ распознавать кошек на видео. Для этой задачи было использовано более 16000 процессоров.
3. В 2016 году AlphaGo победила чемпиона Европы (Фан Хуэй) и чемпиона мира (Ли Седоль) в игре Го.
4. Появление трансформеров. Архитектура трансформеров, представленная в 2017 году, привнесла революционные изменения в область обработки естественного языка и генерации текста, что стало прообразом современных языковых моделей, таких как BERT и GPT.
5. Активное развитие получило мультимодальное обучение, позволяющее объединять информацию из разных модальностей, таких как текст, изображения и видео, для улучшения качества обучения моделей.
6. Доступ к огромным объемам данных. Например, чтобы иметь возможность использовать алгоритмы классификации изображений и распознавания кошек, ранее требовалось проводить долгий ручной отбор образцов самостоятельно. Сегодня простой поиск в Google в доли секунды может выдать миллионы результатов.
7. Активное развитие графических процессоров (GPU) для ускорения расчета алгоритмов обучения. Этот процесс итеративен, и до 2010 года обработка всей выборки могла занимать несколько недель. Вычислительная мощность видеокарт (способная выполнять более тысячи миллиардов транзакций в секунду) позволила добиться значительного прогресса при ограниченных финансовых затратах (менее 1000 евро на одну видеокарту).

2020-е: Эпоха генеративного ИИ

Генеративный ИИ – это популярная в 2020-х годах область исследований, которая занимается автоматизированным созданием нового контента, такого как тексты, изображения, видео и аудио, на основе открытых данных и запросов пользователей. Эта технология позволяет автоматизировать задачи по созданию контента на основе компиляции и смешения определенных пользователем аспектов накопленных знаний, и

создавать удачные изображения, музыкальные композиции и, в отдельных случаях, научные открытия.

Развитие генеративного ИИ началось в конце 1990-х годов, активизировалось в 2010-х годах, но 2020-е годы стали настоящим прорывом. Это произошло в связи с развитием генеративно-состязательных сетей (GAN - Generative Adversarial Network) и появлением больших языковых моделей, таких как ChatGPT, в 2023 году. Эти модели позволяют генерировать тексты такого высокого качества, что часто его довольно сложно отличить от текста, написанного журналистом.

Одним из ключевых людей, стоящих за развитием генеративного ИИ, был Ян Гудфеллоу, канадский исследователь. Он известен своими работами в области глубокого обучения и генеративных моделей. Ян Гудфеллоу работал в различных институциях, включая Google Brain, OpenAI и Microsoft Research. Он внёс значительный вклад в развитие генеративного ИИ, особенно в области генеративных состязательных сетей (GANs) и трансформеров.

Основные свойства генеративного ИИ:

1. Генерация. Это способность создавать новые данные, такие как изображения, текст или аудио, на основе смешения отдельных аспектов в существующих данных.
2. Анализ. Это способность анализировать и интерпретировать большие объёмы данных, находить закономерности и делать выводы.
3. Обучение. Это способность обучаться на основе малого количества примеров, улучшать свои результаты и адаптироваться к новым условиям.
4. Автоматизация. Это способность автоматизировать рутинные задачи, такие как обработка данных, суммаризация текстов и др.
5. Удобство получения информации. Это способность отвечать на текстовые вопросы пользователя по всему корпусу накопленных в процессе обучения ИИ знаний.

Основные сценарии использования генеративного ИИ на текущий момент:

1. Создание контента. Это генерация изображений, видео, музыки и текстов для маркетинга, развлечений и искусства.
2. Распознавание образов. Это анализ и классификация изображений, видео и аудио для медицинской диагностики, безопасности и распознавания лиц.
3. Машинный перевод. Это перевод текстов на разные языки с высокой точностью и качеством перевода.
4. Персонализация. Это адаптация продуктов и услуг под индивидуальные потребности и предпочтения пользователей – конкретных физических лиц.

Негативной стороной высоких достижений в области реалистично сгенерированного контента стали так называемые «deep fake» - ложные аудио, фото или видео, которые

обычный неподготовленный человек с высокой степенью вероятности не сможет отличить от оригиналов. Это явилось серьезным вызовом для новой дисциплины - киберкриминалистики.

Эпоха генеративного ИИ открывает широкие возможности для творчества, исследований, образования и бизнеса. Однако, эта технология также вызывает опасения общественности относительно этических аспектов и возможного негативного воздействия не только на рынок труда, но и на вектор развития человечества.

Врезка:

В области медицины и биологии GenAI может ускорить разработку лекарств, проверяя и создавая молекулы для новых лекарственных форм, а также реализовать концепции персонализированного медицинского обслуживания. В области создания контента GenAI может автоматизировать задачи, сэкономить время и деньги, а также создавать индивидуальные маркетинговые материалы. GenAI можно использовать для разработки чат-ботов для обслуживания клиентов, извлечения знаний из баз данных для сотрудников. Для промышленности на основе GenAI ведется разработка дизайна продуктов, проектирование изделий, создание цифровых двойников.

В сфере транспорта GenAI играет решающую роль при реализации технологий автономного и беспилотного вождения, а также создания интеллектуальных транспортных систем. На основе ИИ строятся поисковые системы, системы рекомендаций, таргетированная реклама, виртуальные помощники, автоматический перевод с одного языка на другой, системы распознавания лиц и многое другое. **Таких приложений десятки тысяч. Нейросети типа ChatGPT, Mistral, Llama, GigaChat, YaGPT, Bard или Stable Diffusion внедряются во всех отраслях беспрецедентными темпами.**

Нейронные сети

Один из ключевых элементов архитектуры ИИ — это нейросети. Их математические принципы и архитектура тоже были «изобретены», изучены и описаны в тысячах статей и книг, начиная с 1950х годов. Еще в 1957 г. Фрэнк Розенблатт предложил концепцию перцептрона, фундаментального «строительного блока» нейронных сетей.

Искусственная нейронная сеть (ИНС) - это математическая модель, объединяющая множество однотипных элементарных вычислительных единиц, называемых "нейронами", в слои, которые обрабатывают информацию параллельно.

Компоненты ИНС:

- Нейроны — основные вычислительные единицы, которые принимают входные данные, обрабатывают их и передают результаты дальше по сети. Каждый нейрон выполняет простую математическую операцию: он умножает входные значения на определённые веса, затем суммирует результаты и передаёт их через функцию активации, которая определяет, будет ли нейрон активирован и каким будет его выходной сигнал.
- Синапсы — связи между нейронами, которые определяют силу взаимодействия между ними. Синапсы могут быть возбуждающими или тормозящими, что влияет на то, как нейроны взаимодействуют друг с другом.
- Веса — коэффициенты, присвоенные синапсам, которые влияют на передачу сигнала между нейронами. Веса могут быть положительными или отрицательными, и они определяют, насколько сильно один нейрон влияет на другой.

Состав слоёв ИНС:

- Входной слой — принимает сырые данные для анализа. Входные данные могут быть любыми, от изображений до текстовых данных.
- Скрытые слои — промежуточные слои, выполняющие основную обработку данных. Скрытые слои могут быть одного или нескольких типов, и они выполняют сложные преобразования входных данных, чтобы подготовить их для выходного слоя.
- Выходной слой — выдаёт результат работы сети, например, предсказание или классификацию. Выходной слой может иметь один или несколько нейронов, в зависимости от задачи, которую решает ИНС.

ИНС решают широкий спектр задач. Вот некоторые из них:

- Распознавание образов. Например, идентификация объектов на изображениях, распознавание лиц.
- Обработка естественного языка. Машинный перевод, синтез речи, обработка текста.
- Прогнозирование и анализ. Прогнозирование временных рядов, анализ данных.
- Управление автономными агентами. Автомобильный автопилот, робот с высокой степенью автономности.

Обучение ИНС:

- Обучение ИНС происходит автоматически на больших объёмах данных. Алгоритм обучения позволяет результирующей нейросети адаптироваться к входным данным и улучшать свою производительность. Этот процесс включает корректировку весов синапсов для минимизации ошибок в предсказаниях или классификации.

- Обучение ИНС может быть выполнено различными методами, включая обратное распространение ошибки, стохастический градиентный спуск и другие методы оптимизации.

Врезка: Пример. Описание процесса обучения сверточной нейронной сети.
Структура сверточной нейронной сети (CNN) формируется динамически в процессе обучения. Алгоритм обучения состоит из двух компонентов: менеджера и построителя сети. В процессе обучения алгоритм обучения обеспечивает расчет весовых коэффициентов распознавателя. Менеджер посылает сигнал «построить сеть», который запускает процесс построения структуры сети. Построитель сети рассматривает каждый выход в отдельности. Для вычисления выбранного выхода требуется знать локальную рецептивную область для выходного нейрона. Для каждого нейрона существует набор связей в каждом из слоев – входных (In) и выходных (Out) связей. Поскольку последний (Res) слой связан со всеми выходами слоя S2, то его нейроны должны знать параметры своей рецептивной области. Нейронам слоя S2 также требуются входные значения – выходы сверточного слоя C2.
На рисунке в верхней части видно, как пара C-S слоев посылают друг другу сообщения. Построение связей продолжается от слоя к слою до тех пор, пока не выстраивается «дерево связей» нейронов. В результате, на входной образ поступает сигнал «вернуть выходное значение сканирующего окна» со слоя C1. Для каждого выходного нейрона строится свое такое дерево. Зная выход с входного слоя, можно вычислить выход для C1 слоя, затем S1 и так далее, до Res-слоя, используя построенное дерево связей фрагмента многослойной нейронной сети для одного нейрона выходного слоя.
После окончания синтеза в памяти будет сформирована полная структура нейронной сети со всеми связями. Это позволяет эффективно обрабатывать входные данные и получать результаты.

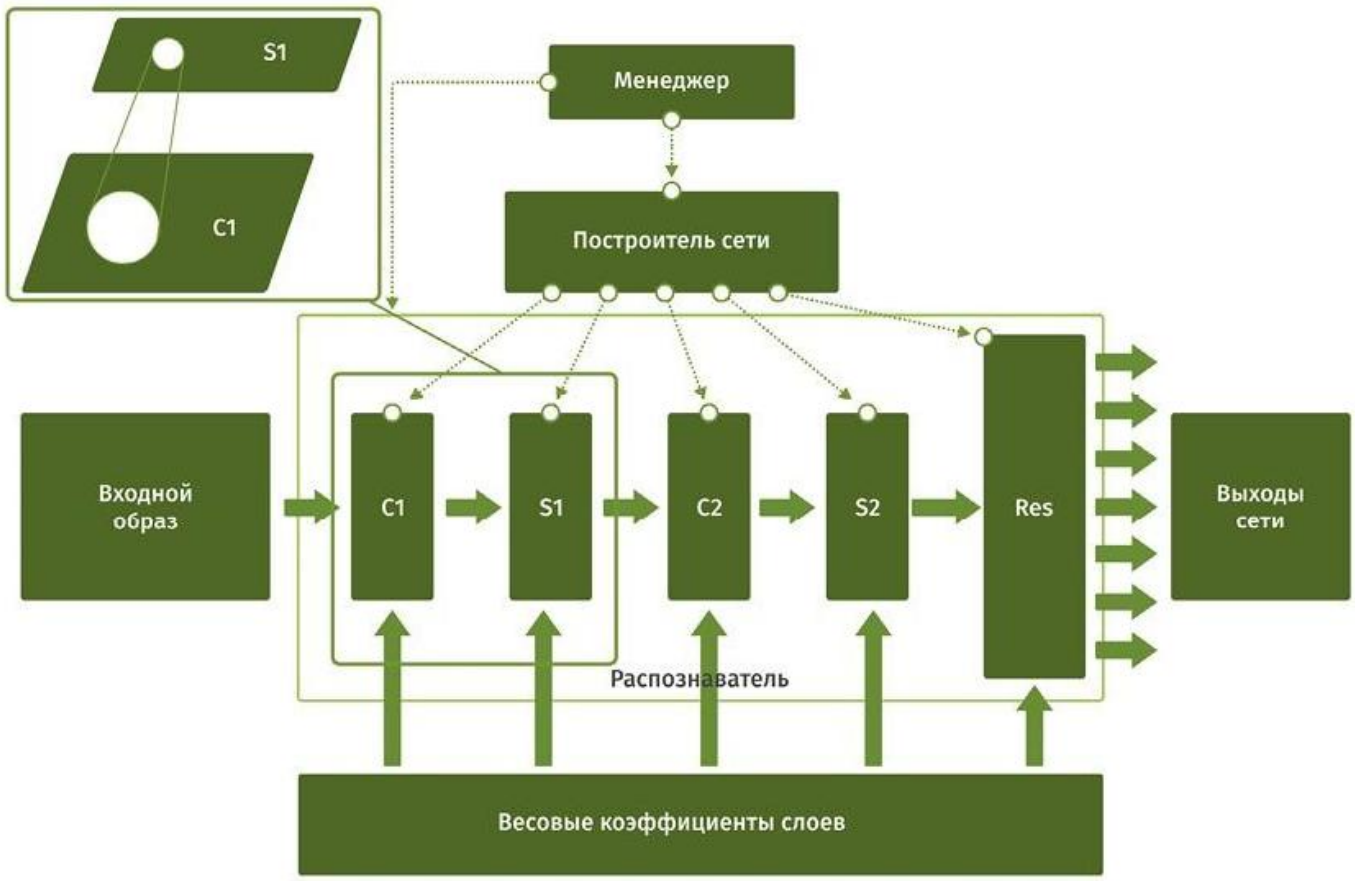


Рис. 3.2.2. Архитектура сверточной нейронной сети. Источник: см. Ссылку 3.2.1.

3.2.3. Классификация ИИ

Известно множество разных вариантов классификации ИИ. Однако мы предлагаем еще один вариант. Согласно рекомендованному определению, ИИ — это результат процесса автоматизации построения алгоритма и его отображения на архитектуру ЭВМ в виде программы. Математическим обоснованием сходимости и устойчивости процесса автоматизации является теорема А.Н. Тихонова о неподвижных точках отображения на упорядоченных ограниченных множествах. Как только ИИ - это программа, следовательно, возможно определить классификацию по степени сложности программ; где под сложностью понимается рост количества и разнообразия компонентов программы и их взаимодействий между собой. При этом, по мере усложнения, программы объединяются в платформы. Платформы как и программы получают свою специализацию: платформы разработки, платформы безопасности, платформы автоматизации операций непрерывной интеграции и развертывания, платформы автоматизации процессов эксплуатации и контроля качества.

На основе этих утверждений предлагаются следующие классификаторы:

1. Методы и модели
2. ИИ-системы.
3. ИИ-платформы.
4. Подотчетные платформы.

Методы и модели

Методы обучения - это математические и статистические методы, которые используются для построения алгоритмов на основе данных. Методы обучения реализуются соответствующими алгоритмами - назовем их “алгоритмы обучения”. При этом, имеются также “результирующие алгоритмы”, полученные в результате завершения работы алгоритмов обучения. Результирующие алгоритмы, как правило, сильно связаны с алгоритмами обучения, поэтому их рассматривают как элементы одной категории и объединяют в термин “модель”.

Методы обучения и полученные в результате модели используются для решения конкретных задач. Приведем обобщенную классификацию методов обучения и получаемых моделей. В основу классификации моделей и методов легли материалы ресурса http://www.machinelearning.ru.

Категория “Методы и модели” изображена на Рисунке 3.2.1.

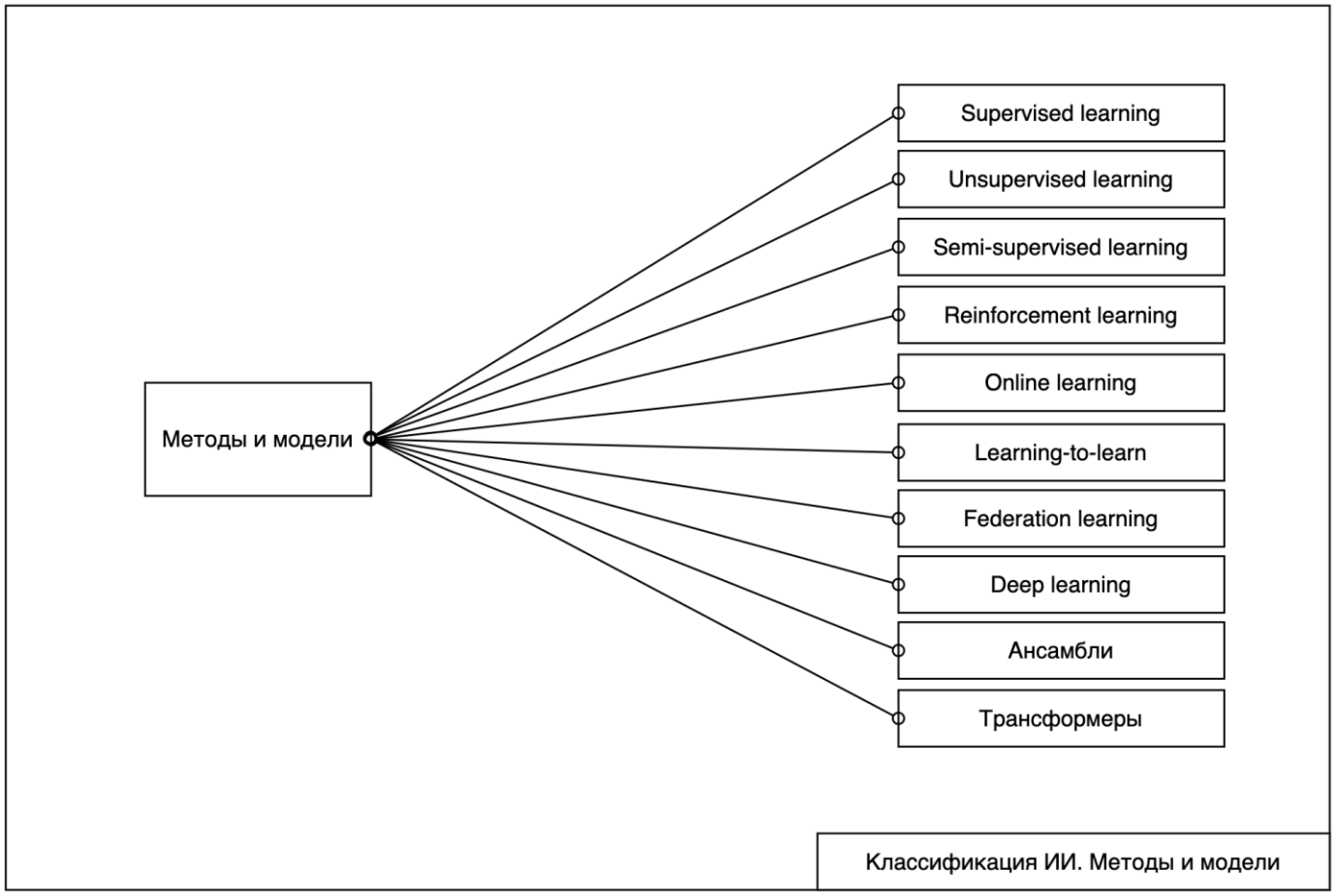


Рисунок. 3.2.1. Классификация ИИ. Методы и модели

Supervised learning

Supervised learning (обучение с учителем) – это метод машинного обучения, при котором модель обучается на размеченных данных. В процессе обучения модель получает входные данные и соответствующие им правильные ответы (метки), что позволяет ей научиться делать предсказания на новых, ранее не известных данных. Этот метод широко используется в различных областях, таких как распознавание изображений, обработка естественного языка и прогнозирование временных рядов.

Основные задачи, решаемые с помощью Supervised learning:

- Классификация. Это задача, в которой модель должна дать ответ (да/нет) на принадлежность заданной категории (классу) для каждого примера. Например, определение, является ли электронное письмо спамом или нет.

- Многоклассовая классификация. Это разновидность классификации, в которой модель должна предсказать одну из нескольких возможных категорий. Например, распознавание рукописных цифр от 0 до 9.
- Регрессия. Это задача, в которой модель должна предсказать непрерывное значение. Например, прогнозирование цены на жилье на основе различных характеристик, таких как площадь, количество комнат и расположение.

Для успешного применения Supervised learning требуется наличие большого количества данных со сбалансированной разметкой, что может вызвать сложности с подготовкой эффективного обучения.

Unsupervised learning

Unsupervised learning (обучение без учителя) – это метод машинного обучения, при котором модель обучается на неразмеченных данных. В отличие от Supervised learning, здесь модель не получает указаний о том, какие ответы являются правильными. Вместо этого алгоритм обучения самостоятельно пытается выявить закономерности и структуры в данных.

Основные задачи, решаемые с помощью Unsupervised learning:

- Кластеризация. Это задача, в которой модель группирует данные в кластеры на основе их сходства. Например, группировка клиентов магазина по их покупательским предпочтениям.
- Выявление аномалий. Это задача, в которой модель обнаруживает необычные или аномальные наблюдения в данных. Это может быть полезно для обнаружения мошенничества или технических проблем.
- Ассоциативные правила – задача, в которой модель находит правила, описывающие связи между различными объектами в данных. Например, анализ покупок в супермаркете для выявления товаров, которые часто покупают вместе.

Unsupervised learning может быть полезным в ситуациях, когда разметка данных невозможна или слишком трудоемка, однако этот метод может потребовать больше времени и усилий для настройки и интерпретации результатов.

Semi-supervised learning

Semi-supervised learning (обучение с частичным привлечением учителя) – это метод машинного обучения, который сочетает в себе преимущества Supervised learning и

Unsupervised learning. Метод реализует подход, при котором модель обучается на небольшом количестве размеченных данных и большом количестве неразмеченных данных.

Задачи, решаемые с помощью Semi-supervised learning:

- Улучшение качества обучения. Добавление небольшого количества размеченных данных к большому количеству неразмеченных данных может значительно повысить точность модели по сравнению с обучением только на неразмеченных данных.
- Снижение затрат на разметку данных. Использование неразмеченных данных позволяет сократить затраты на ручную разметку данных, которая может быть дорогостоящей и трудоемкой.
- Адаптация к новым данным. Модель, обученная на комбинации размеченных и неразмеченных данных, может лучше адаптироваться к новым, ранее не виденным данным.

Semi-supervised learning находит применение в различных областях, включая обработку изображений, анализ текстов и биоинформатику, где ручная разметка данных может быть сложной или невозможной.

Reinforcement learning

Reinforcement learning (обучение с подкреплением) – это метод машинного обучения, который обеспечивает обучение модели на основе результатов взаимодействия с внешней средой. Метод реализует подход, в котором программный агент, инкапсулирующий модель, предпринимает какие-то действия в среде, получая обратную связь в виде вознаграждения или наказания. Цель агента – максимизировать суммарное вознаграждение за серию действий.

Задачи, решаемые с помощью Reinforcement learning:

- Автоматическое управление. Это обучение роботов и автономных транспортных средств принимать решения в реальном времени на основе обратной связи от среды.
- Игры и стратегические задачи. Это создание алгоритмов, способных играть в сложные игры (например, шахматы) или решать стратегические задачи, такие как управление ресурсами.
- Оптимизация процессов. Это использование приложений со встроенными механизмами Reinforcement learning для оптимизации производственных процессов, логистики и управления запасами.
- Рекомендательные системы. Это самообучающиеся в процессе работы агенты, которые могут рекомендовать товары, услуги или контент на основе предпочтений пользователей.

Reinforcement learning находит широкое применение в различных областях, где требуется адаптивное поведение и принятие решений в условиях неопределенности.

Online learning

Online learning (онлайн-машинное обучение) – это метод машинного обучения, который позволяет моделям обучаться и адаптироваться к новым данным в режиме реального времени, без необходимости повторного обучения всей модели. Это особенно полезно в ситуациях, когда данные постоянно обновляются или поступают в реальном времени, например, в системах рекомендаций, анализе финансовых рынков или прогнозировании трафика.

Основные задачи, решаемые с помощью онлайн-машинного обучения:

- Обработка больших объемов данных. Онлайн-подход позволяет обрабатывать большие объемы данных без необходимости хранить их все в памяти одновременно.
- Адаптация к изменениям. Модели могут адаптироваться к изменениям в данных, что важно для многих приложений, таких как рекомендательные системы или прогнозирование временных рядов.
- Снижение задержек. Обучение в реальном времени позволяет снизить задержки между поступлением новых данных и обновлением модели, что критически важно для некоторых приложений, например, в системах управления трафиком.
- Экономия ресурсов. Онлайн-обучение может быть более эффективным с точки зрения использования вычислительных ресурсов, поскольку не требует повторного обучения всей модели при поступлении новых данных.

Онлайн-машинное обучение находит применение в широком спектре областей, включая интернет-рекламу, финансовые услуги, здравоохранение и многие другие, где важна быстрая адаптация к новым данным и высокая степень актуальности моделей.

Transfer learning

Transfer learning – это метод машинного обучения, который позволяет использовать признаки, полученные при обучении алгоритма решения одной задачи, для улучшения продуктивности алгоритма для решения другой, связанной задачи. В отличие от методов, когда модель обучается с нуля для каждой новой задачи, transfer learning использует предварительно обученные модели и адаптирует их к новым задачам, что значительно сокращает время и ресурсы, необходимые для обучения.

Задачи, решаемые с помощью transfer learning:

- Сокращение времени обучения. Transfer learning позволяет ускорить процесс обучения модели, так как большая часть работы уже выполнена при предварительном обучении.
- Улучшение производительности. Использование предварительно обученной модели в качестве отправной точки может привести к улучшению производительности на новой задаче, особенно если объем доступных данных для новой задачи ограничен.
- Адаптация к новым доменам. Transfer learning позволяет моделям адаптироваться к новым доменам, где данные могут существенно отличаться от тех, на которых модель была предварительно обучена.
- Уменьшение потребности в данных. Предварительное обучение на большом наборе данных может помочь модели обобщать информацию и делать прогнозы на основе меньшего количества данных.

Transfer learning находит применение в широком спектре областей, включая компьютерное зрение, обработку естественного языка, рекомендательные системы и многие другие, где доступно большое количество данных для предварительного обучения.

Learning-to-learn

Learning-to-learn (L2L) – это метод машинного обучения, который подразумевает автоматизацию способности модели самостоятельно адаптироваться и улучшать свою производительность на основе полученного опыта. В отличие от иных методов, где модель обучается на фиксированном наборе данных и затем применяется к новым данным без изменений, L2L позволяет модели учиться на своих ошибках и улучшать свои способности к обобщению и адаптации к новым ситуациям.

Задачи, решаемые с помощью L2L:

- Адаптация к изменениям. L2L позволяет моделям адаптироваться к изменениям в данных, что важно для многих приложений, таких как рекомендательные системы или прогнозирование временных рядов.
- Улучшение обобщения. L2L помогает моделям лучше обобщать полученные знания на новые данные, что повышает их точность и эффективность.
- Самостоятельное обучение. L2L позволяет моделям самостоятельно изучать новые концепции и правила без необходимости полного переобучения.
- Сокращение времени обучения. L2L может сократить время обучения модели, поскольку она учится на своих ошибках и быстрее достигает высокой производительности.

L2L находит применение в широком спектре областей, включая автономные системы, робототехнику, обработку естественного языка и многие другие, где важна способность модели к самообучению и адаптации к новым условиям.

Federation learning

Federation learning – это метод машинного обучения, который позволяет объединять усилия нескольких участников для обучения общей модели, сохраняя при этом приватность данных каждого участника. В отличие от централизованного обучения, где все данные собираются на одном сервере, в federation learning каждый участник обучает свою локальную модель, а затем обменивается только весами модели, а не самими данными. Это обеспечивает защиту персональных данных и конфиденциальности участников.

Задачи, решаемые с помощью federation learning:

- Защита персональных данных. Federation learning позволяет участникам сохранять контроль над своими данными, не передавая их третьим лицам.
- Улучшение качества обучения. Объединение данных нескольких участников увеличивает полноту обучающего корпуса, что позволяет создать более точную и надежную модель, чем обучение на данных одного участника.
- Решение проблемы нехватки данных. Federation learning может быть использовано для обучения моделей на небольших наборах данных, которые не позволяют достичь высокой точности при обучении на одном устройстве.
- Снижение нагрузки на сеть. Обмен весами моделей вместо полных наборов данных снижает нагрузку на сеть и ускоряет процесс обучения.

Federation learning находит применение в различных областях, где требуется обработка персональных данных, таких как медицина, финансы, образование и другие.

Deep learning

Deep learning (глубокое обучение) – это метод машинного обучения, использующий многослойные нейронные сети для решения сложных задач. Метод требует больших объемов данных для обучения. Глубокое обучение применяется для решения задач, не решаемых другими методами. К таким областям относится компьютерное зрение, распознавание речи, обработка естественного языка и многие другие.

Задачи, решаемые с помощью deep learning:

- Распознавание изображений. Deep learning позволяет распознавать объекты на фотографиях и видео, что находит применение в системах безопасности, автономных автомобилях и других приложениях.
- Распознавание речи. Deep learning используется для создания систем распознавания речи, которые могут переводить речь в текст, а также использовать голос для управления устройствами и т.д.
- Обработка естественного языка. Deep learning применяется для создания чат-ботов, переводчиков, систем генерации текстов и других приложений, работающих с текстом.
- Предсказание и прогнозирование. Deep learning может использоваться для прогнозирования временных рядов, предсказания спроса на товары, оценки рисков и других задач, требующих анализа данных.
- Классификация и кластеризация. Deep learning позволяет классифицировать объекты по категориям, а также группировать их в кластеры на основе общих характеристик.
- Генерация контента. С помощью deep learning создаются алгоритмы, способные генерировать изображения, музыку, тексты и другой контент на основе заданных условий.

Deep learning продолжает развиваться и находить новые применения в различных сферах, делая возможным решение сложных задач, ранее недоступных для автоматизации.

Ансамбли

Ансамбли представляют собой комбинацию разных моделей машинного обучения, каждая из которых специализируется на определенной задаче или аспекте данных. Эти модели работают вместе, чтобы сформировать более мощную и точную ИИ-систему, способную решать сложные задачи. Сложные ансамбли могут включать различные типы моделей, от простых до сложных, такие как деревья решений, случайные леса, градиентный бустинг, нейронные сети, и другие. Каждая модель вносит свой вклад в общий результат, компенсируя недостатки других моделей и повышая общую производительность ИИ-системы.

Примером ансамбля может служить ИИ-система, которая объединяет модели для классификации изображений, обнаружения объектов и сегментации изображений. Каждая модель отвечает за свою часть задачи, а их совместная работа позволяет достичь высокой точности и полноты распознавания объектов на изображениях.

Создание и обучение ансамблей требует тщательного подбора моделей, настройки параметров и разработки стратегии комбинирования результатов. Однако, благодаря своей способности к адаптации и улучшению на основе различных типов данных, сложные ансамбли становятся все более популярными в области.

Трансформеры

Трансформеры представляют собой семейство архитектур ИИ-систем, разработанное для обработки последовательностей. Они отличаются от традиционных рекуррентных нейронных сетей (RNN) и сверточных нейронных сетей (CNN) способностью обрабатывать последовательности параллельно, что делает их особенно эффективными для задач, требующих учета контекста и зависимостей между элементами последовательности.

Архитектура трансформеров состоит из кодировщика и декодировщика, каждый из которых включает в себя механизм внимания. Кодировщик получает на вход векторизованную последовательность с позиционной информацией, а декодировщик получает на вход часть этой последовательности и выход кодировщика.

Компоненты архитектуры трансформеров:

- Кодировщик преобразует входную последовательность в векторное представление, которое содержит информацию о контексте и зависимостях между элементами последовательности.
- Декодировщик использует векторное представление для генерации выходной последовательности.
- Механизм внимания позволяет модели учитывать контекст и зависимости между элементами входной и выходной последовательностей.

Специфика компонентов:

- Кодировщик состоит из механизма самовнимания со входом из предыдущего слоя и нейронной сети с прямой связью со входом из механизма самовнимания.
- Декодировщик состоит из механизма самовнимания со входом из предыдущего слоя, механизма внимания к результатам кодирования со входом из механизма самовнимания и кодировщика и нейронной сети с прямой связью со входом из механизма внимания.
- Механизм внимания позволяет модели фокусироваться на наиболее важных частях входной последовательности при генерации выходной последовательности. Это позволяет модели лучше понимать контекст и генерировать более точные и осмысленные результаты.

Отдельно стоит остановится на механизме внимания. Он состоит из трех основных компонентов:

- Запрос (query) - вектор, представляющий текущий элемент выходной последовательности, для которого будет вычисляться важность элементов входной последовательности.
- Ключ (key) и Значение (value) - векторы, представляющие элементы входной последовательности. Эти векторы используются для вычисления весов важности каждого элемента входной последовательности относительно текущего элемента выходной последовательности.
- Функция внимания (attention function) - функция, которая принимает на вход запрос, ключи и значения, и возвращает веса важности для каждого элемента входной последовательности. Эти веса используются для определения вклада каждого элемента входной последовательности в формирование выходного значения. Функция внимания обычно реализуется через скалярное произведение между запросом и ключами, нормализованное с использованием функции softmax, чтобы получить веса важности. Эти веса затем умножаются на значения, чтобы получить взвешенную сумму, которая и является вкладом каждого элемента входной последовательности в формирование выходного значения. Функция внимания играет ключевую роль в обработке последовательностей, позволяя учитывать контекст и зависимости между элементами последовательности.

В механизме внимания может быть несколько различных функций внимания. В таком случае говорят о нескольких головах внимания (attention head), в том смысле, что одна голова внимания выполняет отдельную функцию внимания. Каждая голова внимания отвечает за анализ определенного аспекта входных данных, например, за определение семантических связей между словами в предложении. Количество голов внимания может варьироваться в зависимости от архитектуры модели и задачи, которую она решает. Это позволяет модели учитывать различные контексты и зависимости между элементами входной последовательности, что делает модель более гибкой и эффективной в решении разнообразных задач.

Благодаря своей способности параллельно обрабатывать последовательности трансформеры могут решать широкий спектр задач. Наиболее популярные из них:

- Машинный перевод. Трансформеры эффективно справляются с задачей перевода текстов с одного языка на другой, учитывая контекст и зависимости между словами.
- Автоматическое реферирование. Трансформеры могут выделять ключевые моменты из длинных текстов, создавая краткие резюме или рефераты.
- Генерация текста. Трансформеры способны генерировать тексты на заданную тему, учитывая контекст и стиль. Это может быть полезно для создания статей, описаний товаров, рекламных текстов и т.д.

- Распознавание речи. Трансформеры могут использоваться для распознавания речи, учитывая контекст и зависимости между звуками.
- Анализ тональности текста. Трансформеры могут определять эмоциональную окраску текста, анализируя контекст и зависимости между словами.
- Классификация текстов. Трансформеры могут классифицировать тексты по категориям, учитывая контекст и зависимости между словами.
- Ответы на вопросы. Трансформеры могут отвечать на вопросы, учитывая контекст и зависимости между словами в вопросе и в тексте.

Наибольший вклад в развитие трансформеров внесли проекты BERT от Google Research, XLNet от Google Brain, GPT от OpenAI.

ИИ-системы

ИИ-системы - это конкретные приложения или наборы приложений, разработанных для выполнения определенных задач. Как правило, ИИ-система использует ресурсы и возможности какой-либо платформы для достижения своих целей. Приложение состоит из отдельных компонентов, которые совместно выполняют одну или набор взаимосвязанных задач.

Категория “ИИ-системы” изображена на Рисунке 3.2.2.

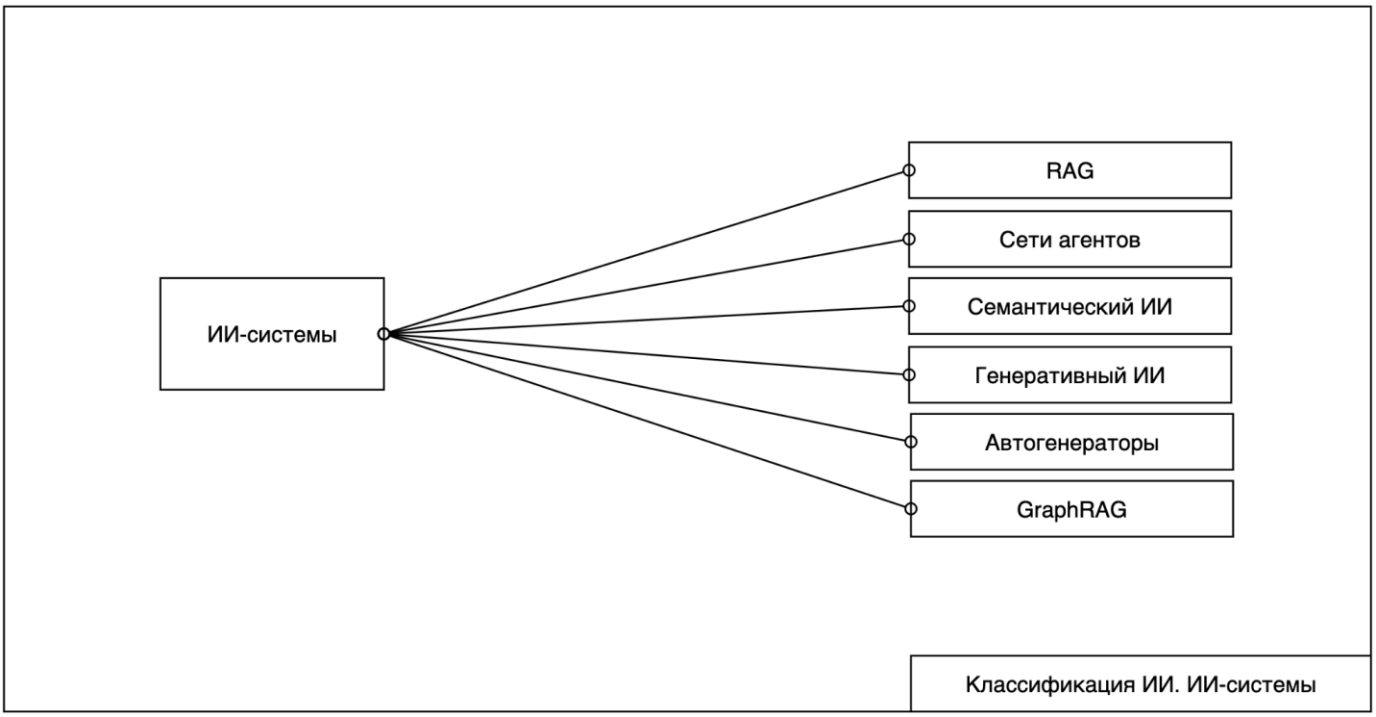


Рисунок. 3.2.2. Классификация ИИ. ИИ-системы

RAG

RAG (Retrieval Augmented Generation) – это технология генерации ответов с дополненным извлечением информации из внешних источников, в качестве которых могут выступать корпоративные базы данных и нормативно-справочная документация. RAG реализует метод, который сочетает в себе два основных этапа: извлечение информации из хранилищ данных и других источников и генерацию текста на основе этой информации. Основная идея RAG заключается в том, чтобы предоставить языковой модели доступ к дополнительным данным, которые могут помочь ей генерировать более точные и информативные ответы.

Процесс RAG начинается с поиска релевантной информации в существующих хранилищах данных, таких как базы данных, энциклопедии, новостные статьи и т.д. Затем эта информация подается на вход языковой модели вместе с вопросом пользователя. Модель использует полученную информацию для генерации ответа, который учитывает не только вопрос, но и дополнительные данные.

RAG позволяет значительно повысить качество ответов языковых моделей, особенно в тех случаях, когда требуется точная и актуальная информация. Этот метод широко используется в различных областях, где важно предоставлять пользователям достоверные и полезные сведения, например, в поддержке клиентов, научных исследованиях и образовании.

Сети агентов

В ИИ под агентом понимается интеллектуальный агент (ИА), который скрывает поведение автономного объекта, который действует, направляя свою деятельность на достижение целей в окружающей среде, используя наблюдение за окружающей средой и внутренним состоянием с помощью метрик или датчиков, взаимодействия с внешней средой при помощи отправки сообщений или других механизмов. ИА могут изучать или использовать знания для достижения своих целей. Они могут быть очень простыми или очень сложными. Например, термостат, рассматривается как пример ИА. Простые ИА, объединенные в сеть, позволяют строить системы большого уровня сложности и функциональных возможностей.

Концепция ИА тесно связана с агентами в экономике, когнитивных науках, этике, практической философии, а также во многих междисциплинарных социально-когнитивных моделях и компьютерных социальных симуляциях. Кроме этого, ИА тесно связаны с программными агентами (автономными компьютерными программами, выполняющими задачи от имени пользователей). В этом случае ИА – это программный агент, обладающий некоторым интеллектом, например, автономные программы (боты), используемые для помощи оператору или интеллектуального анализа данных.

Классическим примером сети интеллектуальных агентов можно назвать реализацию федеративного поиска. Федеративный поиск представляет собой децентрализованный поиск по различным источникам информации с координацией и обработкой результатов поиска на центральном узле. Такой поиск требует централизованной координации распределенных агентов, которые взаимодействуют с поисковыми ресурсами. При этом, необходимо выполнять координацию запросов, передаваемых поисковым агентам, так и слияние результатов поиска, возвращаемых каждым из них.

В случае, когда ИА строится на основе специализированной языковой модели, ИА принято называть “экспертом”. Такие мультиагентные системы получили название “MoE” - Mixture of Experts.

С развитием концепции Smart City тема сетей интеллектуальных агентов получила огромное поле для применения. По сути, концепция Smart City на практике развивает теоретическую модель сетей интеллектуальных агентов, выделяя в ней следующие направления специализации агентов (применительно к использованию технологий машинного обучения и ИИ):

- Рекомендательные системы.
- Персонализация рабочего пространства.
- Системы на основе графов знаний (семантические системы).
- Системы по интеллектуальному анализу и визуализации данных.
- Дорожное движение.
- Самодвижущиеся авто.
- Электрогенерация.
- Распределение энергии.
- Физическая безопасность.
- Социальная безопасность.
- Сохранение здоровья.
- И другие.

Следует отдельно отметить активное развитие таких фреймворков, как Langchain. Такого типа фреймворки представляют собой инструменты разработки, направленные на упрощение процесса создания приложений на основе одной или нескольких больших языковых моделей (LLM). Фреймворк предлагает набор инструментов и библиотек, которые позволяют разработчикам интегрировать LLM в свои проекты, обеспечивая тем самым взаимодействие с моделями и обработку естественного языка.

Архитектура Langchain основана на принципе цепочек (chains), которые позволяют соединять различные компоненты и модули для создания сложных приложений. Это включает в себя возможность объединения нескольких LLM, а также других компонентов,

таких как базы данных, поисковые системы и внешние API. Элементом цепочки может быть агент, инкапсулирующий специфически настроенную (дообученную) большую языковую модель, что позволяет относительно быстро формировать комитеты агентов.

Основные элементы архитектуры Langchain включают:

- Цепочки (Chains). Позволяют объединять различные компоненты и модули для создания сложных приложений.
- Шаблоны промптов (Prompt Templates). Предоставляют возможность создания и управления промптами (структурированными запросами к LLM), что упрощает процесс взаимодействия с моделями.
- Интеграция с LLM. Позволяет напрямую взаимодействовать с различными LLM, такими как GPT-3, LLaMA, и другими.
- Обработка естественного языка (NLP). Включает инструменты для обработки и анализа текстовых данных.
- Хранение и обработка данных. Обеспечивает механизмы для локального хранения и обработки векторизованных данных.

Фреймворки типа Langchain предоставляют разработчикам гибкие инструменты для создания разнообразных приложений, от простых чат-ботов до сложных ИИ-систем.

Семантический ИИ

Системы на основе онтологий известны давно. К примеру, организацией W3C разработан ряд широко известных инициатив. W3C OWL (Ontology Web Language) направлена на реализацию языка описания онтологий. W3C SKOS (Simple Knowledge Organization System) направлена на организацию знаний таким образом, чтобы облегчить взаимодействие различных информационных систем за счёт стандартизации тезаурусов, систем классификации, таксономий и других видов нормализации лексики.

С появлением мощных инструментов обработки естественного языка, таких как глубокие нейронные сети, разработчики начали применять нейросети для автоматизации задач извлечения знаний из неструктурированных текстов и внесения именованных сущностей, фактов и зависимостей напрямую в графы знаний. Таким образом, объединив две технологии в одну систему, автоматизировав процессы наполнения графов знаний, стало возможным значительно увеличить мощь ИИ-систем на базе графов знаний. В сообществе заговорили о новой эре Semantic AI – «эра графов знаний» (The Knowledge Graph Era).

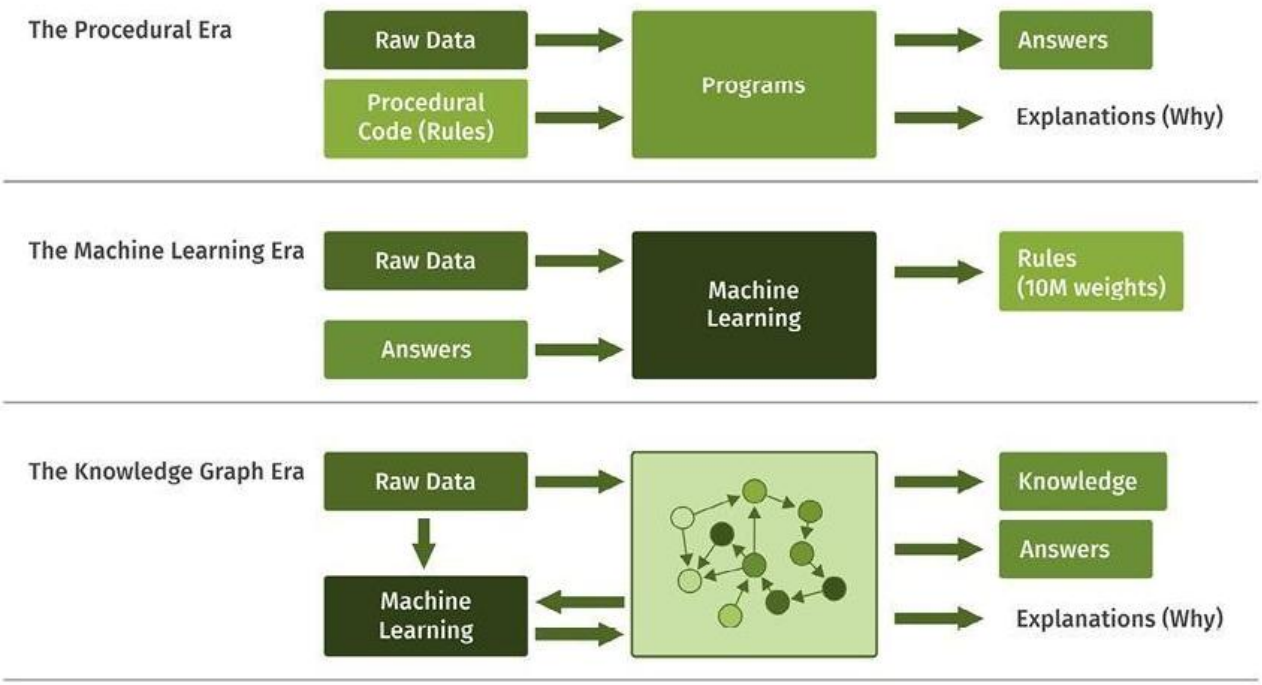


Рис. 3.2.5. Источник: Ссылка 3.2.2.

Современное состояние систем на основе графов знаний можно проиллюстрировать следующей диаграммой (Рис. 3.2.6).

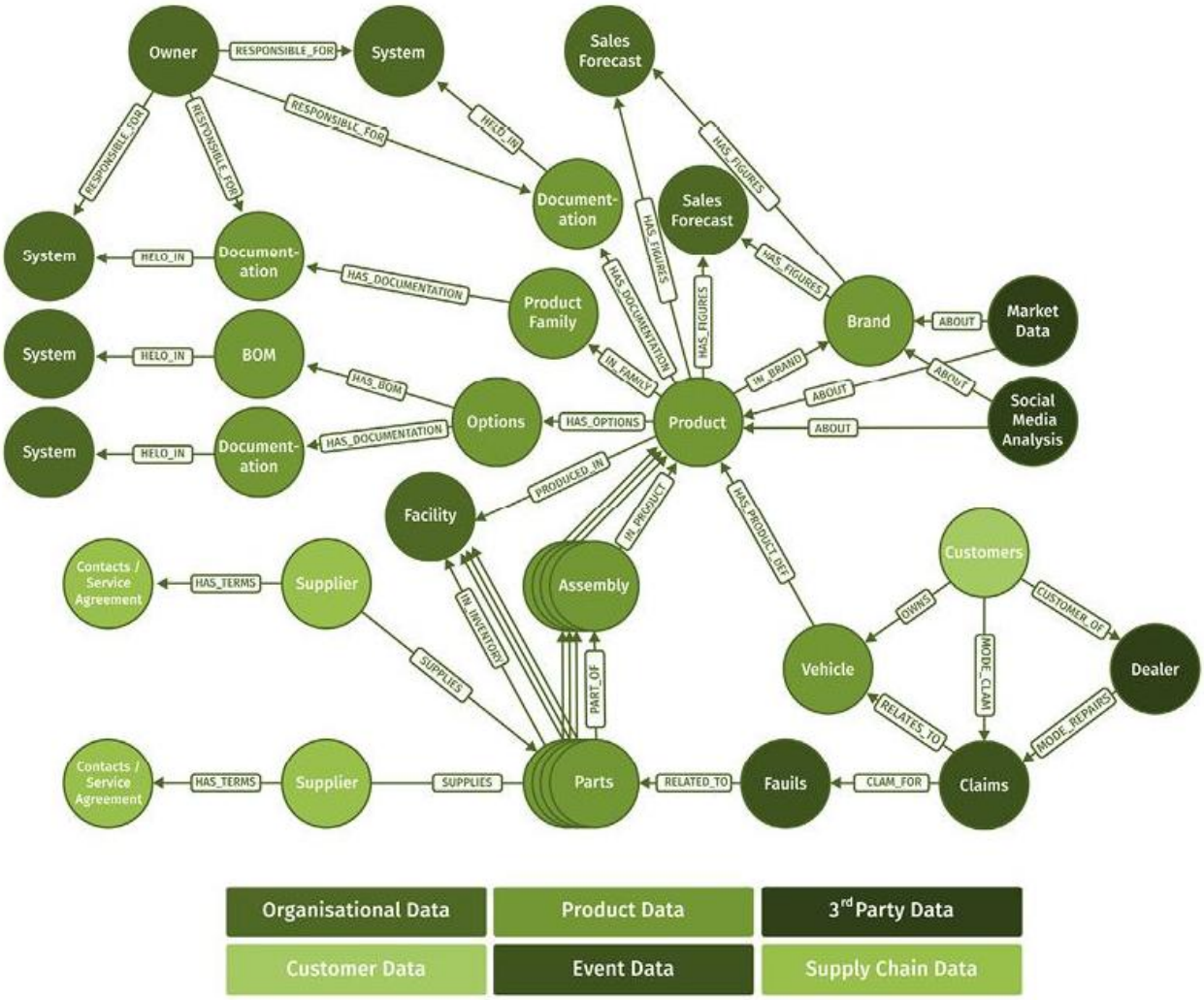


Рис. 3.2.6. Пример графа знаний. Источник: см. Ссылку 3.2.3.

На ней видно, что информация сохраняется в согласованном виде в форме графа знаний в специализированной графовой базе данных. При этом граф знаний становится не просто местом, в котором содержится «единая версия правды» всей организации, а источником для согласованных обучающих наборов данных, позволяющих создавать сложные нейросетевые архитектуры в полностью контролируемой среде обучения, что исключает или сильно снижает риск попадания неконтролируемых примеров в обучающие наборы и, как следствие, в ответы (прогнозы) нейросети. Это особенно важно для таких приложений, как юриспруденция, медицина, государственное управление, финансовый сектор, военное применение. А также необходимо для соответствия требованиям регуляторов (Европейский Союз, США) к объяснимости решений, принимаемых в автоматическом режиме.

В последнее время получило распространение объединение процессов разработки ИИ-систем на основе графов знаний и генеративного ИИ. Автоматическая генерация программного кода построена на следующей цепочке преобразований:

1. При помощи графа знаний формируется достаточно полное, семантически и логически согласованное описание предметной области. Такое описание получило название “карты знаний”.
2. Далее на основе карты знаний генерируется набор спецификаций, определяющих требования, сценарии использования, сценарии тестирования, взаимосвязи между сущностями.
3. Далее на основе спецификаций происходит генерация программного кода.

Генеративный ИИ

Генеративный ИИ – это класс ИИ-систем, предназначенных для создания новых данных, имитирующих оригинальные образцы. Он работает на основе алгоритмов, обученных на больших объемах данных, и способен генерировать разнообразный контент, включая тексты, изображения, музыку и другие формы контента. Ключевым элементом генеративного ИИ являются генеративно-состязательные сети (GANs), изобретенные Ианом Гудфеллоу в 2014 году. GANs состоят из двух компонентов: генератора, создающего новые данные, и дискриминатора, оценивающего их подлинность. В процессе обучения генератор стремится обмануть дискриминатор, создавая все более реалистичные образцы, а дискриминатор учится отличать настоящие данные от подделок.
Благодаря механизму внимания и способности эффективно обрабатывать большие последовательности данных, трансформеры могут и используются в качестве генераторов для генерации текстов, изображений и других форм контента.

В последнее время получили распространение мультимодальные генеративные модели. Такие модели объединяют несколько модальностей данных - текст, изображения, аудио и видео. Это необходимо для создания более комплексного и сложного контента. Они могут использовать различные архитектуры, включая VAE (вариационные автоэнкодеры), TCN (временные сверточные сети) для обработки временных рядов, а также LSTM (долгосрочная краткосрочная память) для обработки последовательностей. Трансформеры также активно применяется в мультимодальных генеративных моделях, так как они позволяют эффективно обрабатывать и комбинировать данные из разных модальностей, таких как текст, изображения и аудио, что делает их идеальным выбором для создания мультимодального контента.

Примеры мультимодальных генеративных моделей:

- Gemini от Google. Эта модель обрабатывает текст, звук, изображения и видео. Она используется для создания датасетов и других задач.
- GPT-4V от OpenAI. Эта модель особенно хороша в области анализа изображений и видео. Она используется для обучения роботов и медицинской диагностики.
- MM1 от Apple. Эта модель способна решать задачи, связанные с изображениями и текстом, например, подсчет объектов или выполнение математических операций.
- DALL-E 2 от OpenAI. Эта модель генерирует изображения на основе текстовых описаний. Она может создавать очень реалистичные и креативные изображения.

Эти модели демонстрируют потенциал генеративного ИИ в создании контента, который может быть использован в различных областях, от искусства и дизайна до науки и техники.

Врезка:

Первая модель GPT-1[1], разработанная в 2018 г. компанией Open AI, содержала 117 млн. параметров. Вторая модель GPT-2[2], появившаяся в 2019 г., - 1.5 млрд параметров. Она уже писала осмысленные тексты. Но качественный скачок произошел лишь с появлением модели GPT-3 в 2020 г., содержащей 175 млрд. параметров, а также ее модификации GPT-3.5 (другое название – InstructGPT)[3], дообученной экспертами с целью соответствия ее ответов некоторым «человеческим» принципам, в частности этичности. Структура GPT-3, описанная в статье Language Models are Few-Shot Learners[4], позволила резко поднять качество обработки и генерации текстов, в том числе за счет роста до 2048 токенов объема текста в запросе, подаваемого на вход модели (так называемый prompt/ промпт).

Модель GPT-3 стала первой коммерческой моделью, ориентированной на рынок. Доступ к модели был реализован как облачный сервис. Нейросеть содержала 175 млрд параметров (96 слоев Transformer-decoder). Была проведена оптимизация потребления памяти: половина слоев внимания в сети используют разреженные матрицы (локальные окна). Развитие парадигмы запросов prompt позволило реализовать «обучение в контексте" (in-context learning). Обучение было проведено на доверенных данных: примеры для обучения были смешаны пропорционально их качеству на основе мнений привлеченных экспертов. Объем данных для обучения был увеличен в 15 раз, добавлена очищенная коллекция CommonCrawl (570GB) и два новых корпуса книг (95GB). В итоге качество генерации текстов резко выросло.

---

[1] https://openai.com/charter
[2] https://openai.com/research/better-language-models
[3] Aligning language models to follow instructions, https://openai.com/research/instruction-following
[4] Brown T., Brown T. B., Mann B., Ryder N., Kaplan J. D., Kaplan J., Neelakantan A., Dhariwal P., Neelakantan A., Shyam P. et al. Language Models are Few-Shot Learners (англ.) // ArXiv.org — 2020. https://arxiv.org/pdf/2005.14165.pdf

В ноябре 2022 г. появилась модель ChatGPT[5], «внутри» которой находится GPT-3.5. Она приобрела известность за счет удобного диалогового API интерфейса и концепции ИИ-ассистента/ помощника, что позволило набрать более 100 млн пользователей менее, чем за 2 мес. Модель «заговорила».

В марте 2023 г. вышла мультимодальная модель GPT-4 (по оценкам, от 200 до 400 млрд. параметров, точных данных нет), в которой увеличен до 32 тыс. токенов размер текстовых входных данных при диалоговом запросе, а также есть возможность генерации изображений. При этом в своем техническом описании модели GPT-4[6] OpenAI не раскрыла данные про особенности архитектуры, методах и наборах данных, использованных для ее обучения.

В мае 2024 г. вышла мультимодальная модель GPT-4o, работающая с текстами, изображениями, аудио и видео, обладающая возможностью рассуждать и позволяющая общаться в режиме реального времени в голосовом диалоге[7]. В планах OpenAI стоит разработка GPT-5. Функциональные возможности модели пока неизвестны, но Генеральный директор компании Сэм Альтман заявил, что модель будет работать как виртуальный мозг[8].

Компания Meta (ранее Facebook) разработала модели Llama (65 млрд параметров), Llama-2 (70 млрд) и Vicuna[9], являющейся моделью Llama, дообученной на ответах ChatGPT. В июле 2024 г. был официально анонсирован выпуск семейства моделей Llama 3.1, версий на 8 млрд, 70 млрд и 405 млрд параметров. Все модели поддерживают мультиязычность и контекст для анализа размером 128К токенов[10]. Компания X(Twitter) разработала модель Grok-1 и опубликовала исходный код на GitHub[11]. Google разрабатывает семейство моделей PALM-2 и Gemini. Компания Anthropic разрабатывает семейство моделей Claude, наиболее известной из которых по состоянию на август 2024 г. была Claude 3.5 Sonnet[12]. Модель работает с контекстом размером 200К, доступна бесплатно на Claude.ai и через приложение Claude iOS app.

Безусловно, свою модель разработала компания Nvidia. В июне 2024г. она выпустила семейство моделей Nemotron-4 340B[13] на 340 млрд параметров. Nemotron-4 340B обучалась на датасете, содержащем тексты на более, чем 50 естественных языках и 40 языках программирования. Архитектура модели построена на технологии Grouped-Query Attention (внимание к групповым запросам,

---

[5] Introducing ChatGPT, https://openai.com/blog/chatgpt
[6] https://cdn.openai.com/papers/gpt-4.pdf
[7] https://community.openai.com/t/gpt-4o-audio-access-for-api/744549/2
[8] https://www.windowscentral.com/software-apps/openai-ceo-sam-altman-suggests-gpt-5-may-work-like-a-virtual-brain
[9] Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90% ChatGPT Quality, https://lmsys.org/blog/2023-03-30-vicuna/
[10] https://ai.meta.com/research/publications/the-llama-3-herd-of-models/
[11] https://github.com/xai-org/grok-1
[12] https://www.anthropic.com/news/claude-3-5-sonnet
[13] https://build.nvidia.com/nvidia/nemotron-4-340b-instruct

GQA)[14] и Rotary Position Embeddings (RoPE)[15], длина анализируемого контекста - 4096 токенов. Модель представлена в виде нескольких вариантов: Nemotron-4-340B-Base (предназначена для генерации синтетических данных), Nemotron-4-340B-Instruct (для разработки чатов и выполнения инструкций) и Nemotron-4-340B-Reward (модель с дополнительным линейным слоем для обучения).

Чрезвычайно активно на рынке генеративных сетей работают китайские компании, например Alibaba Cloud, разрабатывающая линейку моделей под названием Qwen. В июле 2024 г. она выпустила семейство из 5 моделей Qwen2[16] под лицензией Apache 2.0 (бесплатное некоммерческое использование), включающее как обычные модели, так и instruction-tuned модели: Qwen2-0.5B, Qwen2-1.5B, Qwen2-7B, Qwen2-57B-A14B и Qwen2-72B. Все модели работают с контекстом размером до 128k, в дополнение к английскому и китайскому поддерживают ещё 27 языков. Судя по опубликованным бенчмаркам, модель превосходит Llama 3.

В 2024 г. приобрел широкую известность стартап 01.AI, основанный Кай-Фу Ли, в начале 2000х возглавлявшим Google в Китае. У стартапа есть свой чат-бот Wanzhi, а также API для использования и интеграции с другими решениями на базе ИИ. Партнерами стартапа являются Alibaba Cloud и поставщик моделей ИИ компания Fireworks. Языковая модель Yi Large[17] стартапа построена на архитектуре трансформер и по состоянию на конец июля 2024 г. находилась в топ-10 моделей в общем рейтинге моделей LLM - Chatbot Arena Leaderboard, рассчитываемым платформой Hugging Face Space[18], наравне с GPT-4o от OpenAI, Cloude 3.5 Sonnet от Anthropic и Gemini 1.5 Pro от Google. Рейтинг Chatbot Arena Leaderboard формируется пользователями - реальными людьми. На платформе пользователи вслепую пишут запрос, далее их запрос передается в две модели, пользователи видят два ответа и выбирают лучший для себя, не зная, какой модели он принадлежит. На основе десятков тысяч таких запросов формируется рейтинг - чем чаще выбираются результаты модели, тем выше она в рейтинге.

В России свои генеративные модели разработали Яндекс, - YandexGPT и YandexART, доступные в облаке[19], YaGPT[20], а также Сбер, который еще в 2023 г. открыл доступ к своей нейросетевой модели ruGPT-3.5[21] на 13 млрд параметров, а сейчас развивает GigaChat, причем качество работы моделей Сбера и Яндекса на русских текстах уже выше, чем у ChatGPT. Кроме того, Яндекс разработал и

---

[14] https://klu.ai/glossary/grouped-query-attention
[15] RoFormer: Enhanced Transformer with Rotary Position Embedding, Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, Yunfeng Liu, https://arxiv.org/abs/2104.09864
[16] https://qwenlm.github.io/blog/qwen2/
[17] https://platform.lingyiwanwu.com/docs#%E6%A8%A1%E5%9E%8B
[18] https://huggingface.co/spaces/lmsys/chatbot-arena-leaderboard
[19] https://yandex.cloud/ru/services/foundation-models?
[20] https://ya.ru/ai/gpt-2
[21] https://habr.com/ru/companies/sberbank/articles/746736/

выложил в июне 2024 г. в публичный доступ YaFSDP[22] — инструмент для ускорения обучения LLM и сокращения расходов на GPU.

В марте 2024 г. о создании собственной LLM объявил оператор связи МТС[23]. Генеративная модель и сервисы на ее основе активно внедряются для решения различных задач в рамках цифровой экосистемы МТС. Например, в июле 2024 г. был запущен сервис речевой аналитики WordPulse[24] для использования в колл центре, онлайн-кинотеатре KION, МТС Банке и других проектах. WordPulse оценивает тематику разговора, его тональность, динамику изменения настроения клиента, выделять факты и аспекты из высказываний, например, в случае, когда клиенты хвалят быструю доставку, но недовольны неаккуратной упаковкой, строить статистику по каждому факту отдельно. Кроме того, WordPulse используется для автоматического обнаружения нежелательного контента в коротких видео пользовательской видеоплатформы NUUM.

Т-банк (ранее Тинькофф банк) финансирует лабораторию ИИ исследований T-Bank AI Research, в которой разработана русскоязычная языковая модель T-lite с 8 млрд параметров, по бенчмаркам работающая быстрее, чем Chat-GPT 3.5 и Llama-3-8B-Instruct при решении бизнес-задач на русском языке. С помощью T-Lite разработчики смогут создавать LLM-приложения для собственного использования. Это могут быть ИИ-помощники для обработки запросов клиентов, инструменты для анализа данных и продвинутые поисковые системы[25]. T-Lite является частью семейства Gen-T, обучаемых языковых моделей «Т-банка», созданных для решения конкретных задач. Код модели[26] открыт для использования и размещен на платформе Huggingface[27].

Более того, T-Bank AI Research и Центральный университет, созданный при поддержке Авито, МТС, Росатома, СИБУРа, Т-Банка, VK и других компаний, открыли совместную лабораторию Omut AI для проведения фундаментальных исследований в области ИИ и LLM[28]. Цель лаборатории — исследование и разработка инновационных подходов, которые помогут обеспечить контроль и безопасность ИИ, его соответствие стандартам этики и надежности. Результаты работы лаборатории будут доступны научному и индустриальному сообществу.

Лаборатория будет работать по 3 направлениям: AI Alignment (исследования на стыке обработки естественного языка (NLP), обучения с подкреплением (RL) и других областей компьютерных наук, с целью добиться того, чтобы поведение ИИ было предсказуемым и не выходило из-под контроля человека, соответствовало

---

[22] https://github.com/yandex/YaFSDP
[23] https://mts.ai/ru/home/
[24] https://moskva.mts.ru/about/media-centr/soobshheniya-kompanii/novosti-mts-v-rossii-i-mire/2024-07-24/ii-ot-mts-budet-analizirovat-zvonki-i-chaty-s-klientami-ekosistemy
[25] https://club.dns-shop.ru/digest/123163-t-bank-zapustil-moschnuu-russkoyazyichnuu-yazyikovuu-model-t-lite/
[26] https://telegra.ph/Zdes-my-vylozhili-T-Lite-modeli-07-20
[27] https://huggingface.co/AnatoliiPotapov/T-lite-0.1
[28] https://iz.ru/1731347/2024-07-23/v-rossii-obiavili-o-sozdanii-laboratorii-dlia-razvitiia-bezopasnogo-ii

его потребностям и ценностям), LLM Foundations (исследования и поиск более эффективных архитектур для создания больших языковых моделей с нуля, исследования основных принципов и методов улучшения мыслительных способностей моделей) и Multimodal AI (исследования и разработки методов создания универсальных мультимодальных моделей, которые умеют взаимодействовать с человеком через текст, звук и изображения).

Российский Научно-исследовательский институт искусственного интеллекта AIRI[29] разработал на основе опенсорсной модели Mistral-7B, доработав ее архитектуру, свою мультимодальную модель OmniFusion[30], работающую с изображениями, текстами, а в будущем с аудио, 3D и видео контентом.

Российские коммерческие и государственные компании активно тестируют модели, разрабатывая на их основе различные приложения. Например, ведется интеграция российских LLM моделей YandexGPT и GigaChat с порталом Госуслуг для создания сервиса диалога с гражданами - пользователями портала, и их консультирования, в соответствии с концепцией клиентоцентричности государственных органов. Модели обучаются на жалобах и вопросах граждан, ответах официальных организаций на запросы, диалогах с операторами колл-центров и других данных.

Хотя крупные ИТ компании являются драйвером создания новых моделей и приложений, растет активность академических исследовательских организаций. Общее число известных и производительных моделей достигло сотен, а число отдельных частных разработок - несколько тысяч. При этом стоимость создания и обучения больших моделей с сотнями млрд параметров значительно выросла и составляет сотни млн долларов. Например, по данным Stanford University's annual AI index report 2024[31], стоимость вычислений для обучения GPT-4 от OpenAI оценивается в $78 млн, в сравнении с $4.3 млн для GPT-3, а обучение Gemini Ultra от Google обошлось в $191 млн. При этом США опережают Китай, ЕС и Великобританию как ведущий разработчик ИИ. В соответствии с тем же исследованием в 2023 г. в США было разработано 61 крупных генеративных модели, тогда как в ЕС – 21, а в Китае -15.

Важно понимать, что количество параметров моделей очень быстро растет. Именно размер моделей определяет их качество, включая способность к рассуждению и анализу. На рис. 11.1. показан тренд роста числа параметров моделей по секторам в динамике за 2003-2023 г.г. из упомянутого выше отчета Stanford University Artificial Intelligence Index Report 2024, построенный на основе выборки из более, чем 800 моделей из базы данных исследовательской компании Epoch AI[32]. Для многих моделей он уже превысил 1 трлн.

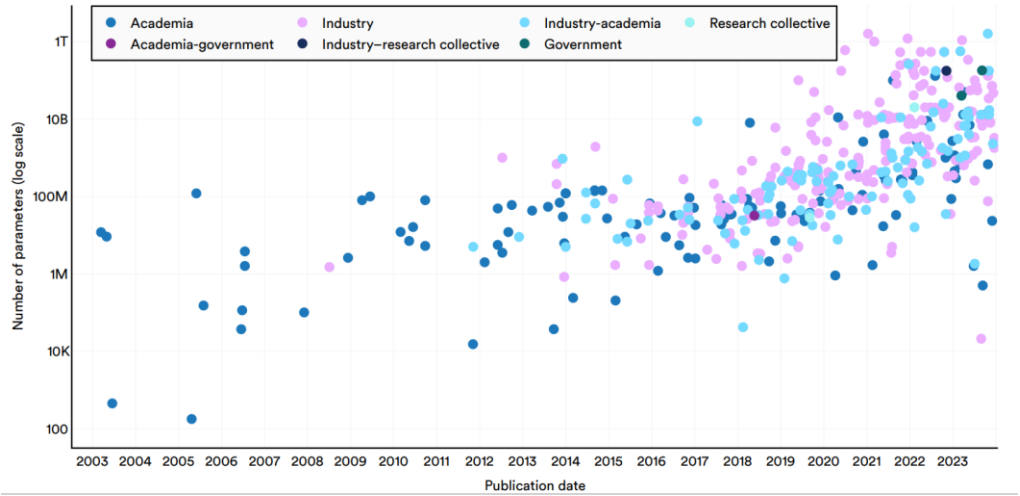
Рост количества параметров в генеративных моделях позволяет выявить и «запомнить» более широкий набор выявленных внутренних связей в обучающих

---

[29] https://airi.net/ru/
[30] https://huggingface.co/AIRI-Institute/OmniFusion
[31] https://www.weforum.org/agenda/2024/04/stanford-university-ai-index-report/
[32] https://epochai.org/data/notable-ai-models

датасетах, что определяет то, как модель интерпретирует входные данные. Модели, обученные на большем количестве данных, обычно имеют больше параметров. Количество параметров в моделях резко возросло с 2010-х годов, что отражает как растущую сложность задач, для решения которых предназначены модели ИИ, доступность данных, появление архитектуры трансформер, усовершенствование аппаратного обеспечения, так и доказанную эффективность более крупных моделей.

Рисунок 11.1. Число параметров моделей по секторам в динамике, 2003-2023 г.г. Источник: Stanford University Artificial Intelligence Index Report 2024, на основе базы данных Epoch AI



На рис. 11.2 показана динамика роста среднего объема вычислительных ресурсов, требуемых для обучения 350 наиболее известных LLM моделей, за период 2010-2024 г.г. на основе базы данных компании Epoch AI. С начала 2010-х объем затраченных ресурсов, измеряемый в FLOP (Floating point operations per second, число операций с плавающей точкой в секунду), рос с темпом 4.1 раза в год, что было обусловлено как ростом финансирования, появлением более требовательных к вычислительным ресурсам новых архитектур, так и повышением производительности вычислений, появлением GPU, суперкомпьютеров и специализированных вычислительных кластеров. По данным той же компании, затраты на обучение LLM растут с темпом 2,4 раза в год, а расчет параметров наиболее продвинутых моделей уже требуют сотен млн долларов. Более 50% от этих затрат приходятся на покупку кластеров GPU, остальное – на создание и эксплуатацию сопутствующей инфраструктуры, а также на электроэнергию.

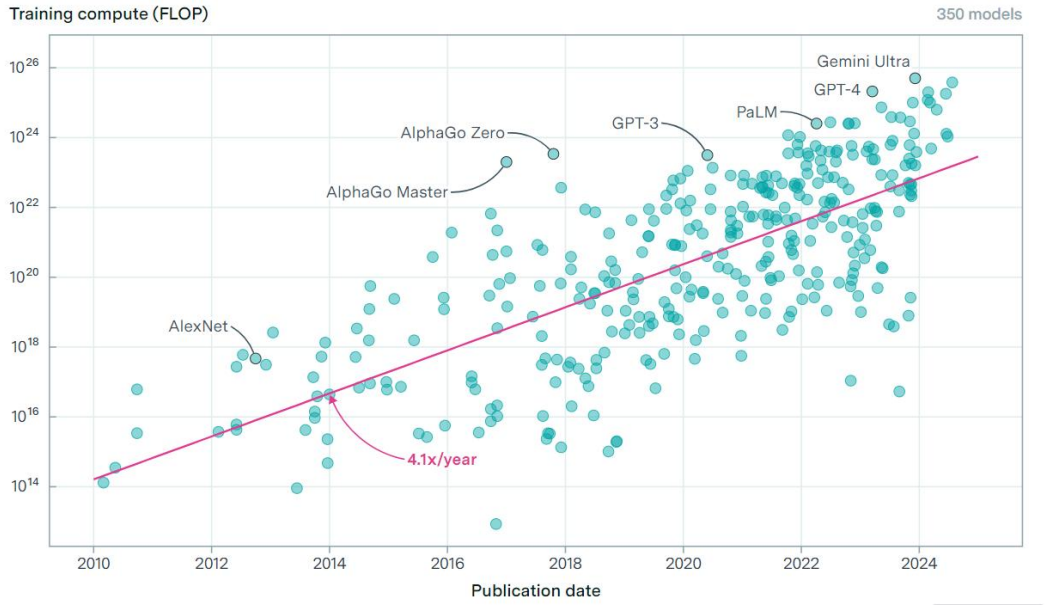
Например, размер тренировочных кластеров OpenAI, которые используются для обучения модели следующего поколения, оценивается в 120000 GPU карт A100. При этом для обучения GPT-4 было задействовано 25000 GPU карт Nvidia

A100, а обучение длилось около 100 дней[33], для LLAMA-3-405B — 16000 GPU[34] (но более мощных).

До 2020 г. крупнейшие языковые модели и модели, работающие с изображениями, были сопоставимы с точки зрения вычислительных затрат на обучение. Однако, после появления архитектуры трансформер языковые модели стали быстро масштабироваться, резко увеличился размер их обучающих датасетов. Кроме того, многие модели, такие как GPT-4 и Gemini, стали мультимодальными, что также требует больше мощностей для обучения.

По данным Epoch AI размер датасетов для обучения языковых моделей растет со скоростью 2,9 раза в год, то есть удваивается каждые 8 месяцев. Наиболее известные модели в настоящее время используют наборы данных с десятками триллионов токенов. Например, модель GPT-4 имеет 1,8 трлн параметров, обучалась на контексте из 13 трлн токенов. Тем не менее, крупнейшие общедоступные наборы данных примерно в десять раз больше этого размера. Например, корпус ресурса Common Crawl[35] содержит сотни трлн слов, собираемых в интернете с 2008 г.

Рисунок 11.2. Вычислительные ресурсы, необходимые для обучения наиболее известных LLM моделей, 2010-2024 г.г. Источник: Epoch AI



Исключительно важно понимать, что всего за несколько лет изменилась концепция использования и разработки генеративных моделей. Они стали строительный элементом, «кирпичиком» для разработки комплексных решений, которые стали собираться как конструктор. Пользователю не надо знать, что внутри модели, «под капотом», ему надо уметь использовать модель.

---

[33] https://klu.ai/blog/gpt-4-llm
[34] https://forum.cursor.com/t/llama-3-1-405b-is-published/6684
[35] https://commoncrawl.org/

Генеративные модели можно дообучать и настраивать, добавляя новые данные в обучающий датасет. Исходные коды множества моделей находятся в открытом доступе, например на ресурсе GitHub. Фактически код модели описывает ее архитектуру - какие слои и как друг с другом связаны, где есть нелинейности в связях, и состоит из вызовов типовых функций, реализованных в открытых библиотеках. К коду прикладывается файл параметров.

Когда компания X (бывший Tvitter) весной 2024 г. выпустила модель Grok-1, она также разместила ее на GitHub. По данным Stanford University Artificial Intelligence Index Report с 2011 г. количество проектов, связанных с ИИ и опубликованных на GitHub, резко увеличилось: с 845 в 2011 г. до примерно 1,8 миллиона в 2024 г., рост за 2023 г. составил 59,3%. Продолжает расти число публикаций в области ИИ. В период с 2010 по 2022 год общее количество публикаций увеличилось почти втрое: с примерно 88 тыс в 2010 г. до более чем 240 тыс в 2022 г.

Интересно, что к гонке моделей присоединилась даже компания Apple. В июле 2024 г. она выпустила опенсорсную LLM под названием DCLM-7B (7 млрд параметров)[36]. Модель открыта, но в условиях значительного числа уже более известных аналогов (Llama3, Gemma2, Qwen2 и т.д.) является только «одним из» решений и ее разработка это скорее всего имиджевый ход компании Apple.

Жестокая конкуренция заставляет разработчиков разрабатывать все более совершенные модели. Например, в июле 2024 г. OpenAI выпустила модель GPT-4o mini[37] на замену старой GPT-3.5 Turbo. Новая модель стала дешевле и качественнее, может работать с контекстным окном размером 128К и поддерживает тот же набор языков, как и «большая» GPT-4o. Важно, что выход GPT-4o mini отражает тенденцию сегментации рынка: у каждого крупного разработчика есть несколько разных моделей - большая и качественная с большим числом параметров, но медленная, и «маленькая», но быстрая (примеры «маленьких» - GPT-4o mini, Gemini Flash, Claude Haiku/Sonnet).

Автогенераторы

Автогенератор - это генеративный ИИ, используемый для генерации полностью готового и гарантированно работающего программного кода. Под понятие кода попадают файлы с параметрами конфигурации, Kubernetes-манифесты, файлы с данными инициализации, сценарии на специализированных языках (shell, Terraform, Ansible и тд), а также программный код на языке программирования. Мотивация развития и использования автогенераторов – выход на новый уровень скорости и сложности разработки программного обеспечения при увеличении общей сложности ИИ-систем и процессов эксплуатации ИИ-платформ. Это происходит путём включения ИИ в жизненный цикл разработки и

[36] https://x.com/_philschmid/status/1814274909775995087
[37] https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/

эксплуатации прикладных систем, начиная от разработки и тестирования, и заканчивая размещением и эксплуатацией готовых систем на продуктивных средах.

Актуален вопрос: что дает использование автогенераторов в процессе разработки и эксплуатации ИТ-активов компании, в чём состоят преимущества их использования?

Формула ответа:

Разработка активов + Эксплуатация активов + Автогенераторы =
= Выход на более высокий уровень возможностей.

Здесь под «разработкой активов» понимается комплекс субпроцессов, таких как кодирование, тестирование, интеграция и размещение программных систем в тестовых средах, под «эксплуатацией активов» понимается комплекс субпроцессов, таких как размещение на продуктивных средах, изоляция по данным и контроль доступа в коммунальной среде для арендаторов (тенантов), фиксация инцидентов и аномалий, автомасштабирование приложений и управление отказами, а под «автогенераторами» - специализированные инструменты, исполняющие одну или более операций, один или более субпроцессов, при этом сцепленные между собой для автоматического исполнения процесса разработки и эксплуатации как единое целое.

Активное включение ИИ в жизненный цикл создания и эксплуатации программных систем приводит к изменению процессов управления и новому разделению труда среди разработчиков. Почему? Потому что это требует новых инструментов разработки и эксплуатации, новых ролевых моделей команд разработки и эксплуатации, новых компетенций команд и новой инфраструктуры компании – AI-driven[38]. Огромный эффект от использования ИИ открывает новые возможности, поэтому компании, бизнес которых критически зависит от эффективности процессов разработки, малого времени вывода продуктов на рынок и стабильно высокого качества услуг интенсивно работают в направлении освоения новых инструментов и практик, а также изменения своих процессов управления, разработки, эксплуатации, при этом активно инвестируют в освоение и адаптацию AI-driven инфраструктуры под свои возможности и условия.

На сегодня можно констатировать, что наиболее продвинутые компании экспериментируют с использованием ИИ-инструментов по следующим направлениям:

- Разработка систем
    - Автозаполнение кода (code completion)

[38] Более подробно об AI-driven инфраструктуре – см. статьи «Архитектура цифровых платформ будущего» и «Инфраструктура современных цифровых платформ», А. Прозоров, Р. Шнырёв и др., «Открытые системы. СУБД», 2021.

    - Трансляция кода (code translation)
    - Генерация заглушек (mock generation)
    - Кодогенерация (code generation)
    - Выявление уязвимостей (vulnerabilities detection)
    - Суммаризация кода (code summarization)
- Эксплуатация систем
    - Масштабирование сервисов (services autoscaling)
    - Обработка отказов (automatic failover)
    - Авторегистрация неисправностей (automatic troubleshooting)
    - Выявление утечек конфиденциальной информации (data leak prevention)
    - Выявление аномалий (anomaly detection)

Процессы разработки

Автозаполнение кода предполагает завершение блока кода за разработчика. Разработчик начинает писать какую-то новую строку кода, ИИ-модуль распознает начало конкретной конструкции и предлагает автозаполнить продолжение. Автозаполнение реализуется через плагины сред IDE, например таких, как IntelliJ IDEA или PyCharm, при этом в состав плагина входит ИИ-модуль, обученный на открытых и/или корпоративных репозиториях конкретного языка программирования. Автозаполнение существенно экономит время разработчика на запись кода и исключает ошибки в параметрах вызовов функций и реализации автоподставляемого блока кода.

Трансляция кода предполагает трансляцию кода с одного языка программирования и/или фреймворка на другой. Трансляция реализуется при помощи утилиты, которая подключается к CI/CD конвейеру; в состав утилиты входит ИИ-модуль для трансляции. Трансляция существенно экономит ресурсы и время на перенос приложения с одного технологического стека на другой. Например, для ускорения и удешевления, разработка происходит на популярном стеке Python/Django/Postgres, а затем проект транслируется на промышленные стеки Java/Spring/Oracle или C#/ASP.NET/MS SQL, в зависимости от требований конкретного заказчика.

Генерация заглушек предполагает генерацию API, которое выдает данные, очень похожие на оригинальные. Разработчик или тестировщик сразу получает практически оригинальные данные, но при этом сохраняется конфиденциальность информации, необходимая для многих, особенно финансовых организаций. ИИ-генератор заглушек обеспечивает аналогичное разнообразие данных, включая нетипичные примеры для тестирования граничных и исключительных случаев. В конечном итоге заглушка API полностью имитирует работу реальной системы.

Кодогенерация в каноническом понимании вычислительной техники – это часть процесса компиляции. В современных условиях кодогенерация широко вошла в состав инструментального стека автоматизации работы разработчиков и может быть определена как «процесс автоматического создания программного кода на основе внешнего представления». Вариантов внешнего представления может быть много. Самыми популярными вариантами внешнего представления являются декларативные программы в формальном машинном представлении, таком как XML, JSON, YAML и т.д.

CASE-инструменты, таких как ARIS, System Architect, и многие другие, включая современные RPA-конструкторы, блок-схемы или диаграммы, как правило, могут быть представлены на XML, JSON или YAML. Следовательно, данный вариант квалифицируется по вышеуказанному правилу.

В августе 2021 года компания OpenAI представила проект кодогенератора Codex, созданного на базе архитектуры GPT-3 и обученного на кодовой базе открытых репозиториев GitHab. Результаты публичных тестов Codex вызвали мощный интерес со стороны сообщества разработчиков, так как Codex получает на вход корректные высказывания на естественном английском языке, а выдает корректный текст на Python или JavaScript готовый к исполнению. Недостатков у Codex ещё очень много и он пока совсем не готов к промышленному применению «из коробки», тем не менее, сообщество разработчиков проявило мощный интерес к теме кодогенерации на основе корректных высказываний на естественном языке, так как такой подход кардинально меняет устоявшуюся практику разработки программ, где трансляцией низкоуровневой архитектуры в исполняемый код выступает программист, а процесс «ручной» трансляции архитектуры в код носит весьма ресурсоемкий характер и риски человеческого фактора.

Выявление уязвимостей предполагает статический анализ кода на предмет известных уязвимостей и потенциально опасных операций. Сформирован и существует целый класс таких систем – статических анализаторов кода. Статические анализаторы традиционно применяются в областях, где очень дорого обходятся ошибки или уязвимости. Например, разработчики операционных систем, мобильных устройств, провайдеры платформ для разворачивания PaaS или serverless приложений. Ведутся работы над ИИ, который добавит к таким системам дополнительные возможности по поиску аномальных фрагментов кода и автозамене этих фрагментов на заведомо безопасные блоки прямо на лету – после этапа загрузки кода пользователя на платформу, но перед этапом отправки кода пользователя на исполнение.

Суммаризация кода предполагает формирование краткого содержательного описания блока кода на естественном языке с целью создания комментария к этому блоку кода, соответствующего индустриальным и/или корпоративным стандартам. Суммаризация кода помогает упростить и ускорить процесс вовлечения новых сотрудников в поддержку или

доработку существующего кода, так как позволяет новым сотрудникам понять существующий код. Это направление использования ИИ позволяет сохранить инвестиции, сделанные в разработку legacy-кода.

Процессы эксплуатации

Масштабирование сервисов реализуется двумя взаимодополняющими способами:

1. Масштабированием по данным
2. Масштабированием по вычислительной мощности.

Масштабирование по данным хорошо известно под термином «шардирование данных». Шардирование – это горизонтальное партиционирование данных, принцип наполнения и использования базы данных, при котором логически независимые строки таблиц хранятся раздельно, заранее сгруппированные в секции; при этом секции могут размещаться на разных серверах базы данных, а этом один физический узел вычислительного кластера может содержать несколько логических серверов (инстансов) баз данных. Для реализации шардирования необходимо на прокси-сервере, который получает запросы к базе данных, эффективно решать задачу выбора шарды для переадресации запроса. В этих целях целесообразно применять ИИ-модуль, который по тексту сообщения будет прогнозировать целевую шарду.

Масштабирование по вычислительной мощности – это основная бизнес-задача, эффективно решаемая эластичными ИТ-инфраструктурами. Традиционный подход к масштабированию по мощности – настройка политик масштабирования на вычислительном кластере. Как это выглядит? Допустим, мы задаем политику, что при превышении на прокси-сервере вычислительного кластера количества входящих запросов к заданному приложению до значения «X в сек.» необходимо запустить еще одну реплику определенной группы сервисов. Для пользователей приложение отвечает быстро и всё выглядит очень хорошо до тех пор, пока нагрузка растёт или снижается достаточно плавно, а для «прогрева» сервисов приложения не требуется много времени. Когда вновь запущенным сервисам необходим «прогрев», который может занимать достаточно продолжительное время, а нагрузка стремительно изменяется, чтобы обеспечить для пользователя неизменно высокое качество услуги необходимо заранее прогнозировать момент запуска новых реплик сервисов. Для прогнозирования таких моментов используют ИИ-модули, которые в случае необходимости инициируют «прогревающую» нагрузку.

Автоматическая обработка отказов применяется в отказоустойчивых конфигурациях вычислительных кластеров и серверов баз данных. Как правило, обработка отказа реализуется аварийным переводом трафика или нагрузки на другой, заведомо работоспособный ресурс. При авариях оборудования или инфраструктуры высока вероятность потери запросов от пользователей и отката активных транзакций баз данных.

Чтобы избежать таких неприятностей, обеспечивая тем самым стабильно высокий уровень сервиса, используют прогнозирование наступления аварий при помощи выявления аномалий в поведении оборудования, инфраструктуры и приложений. Для выявления аномалий используются ИИ-модули. В случае выявления аномалии и прогнозирования сбоя запускается сценарий обработки отказа: старт новой реплики приложения или сбойной группы сервисов и, после их «прогрева», перевод прикладного трафика с аварийного ресурса на новую реплику. После полного перевода трафика и фиксации всех транзакций аварийный ресурс может быть освобождён.

Авторегистрация неисправностей, как правило, предполагает автоматическое выполнение следующих процессов:

1. Сбор доступной контекстной информации об аномалии после получения уведомления от детектора аномалий.
2. Квалификации аномалии на предмет наличия инцидента.
3. Регистрация инцидента в ITSM-системе.
4. Маршрутизация уведомления об инциденте на исполнителя.

Для квалификации аномалии на предмет инцидента может быть использован детектор инцидентов на основе ИИ-модуля. Для маршрутизации уведомления об инциденте на исполнителя может быть использован классификатор на основе ИИ-модуля, который по тексту уведомления определяет исполнителя.

Выявление утечек конфиденциальной информации – это широко известная задача информационной безопасности и вотчина систем класса DLP. В случае, когда в большой компании используется две или более систем DLP, узко специализирующихся на утечках конфиденциальной информации в офисных файлах, мультимедиа, изображениях, BIM-моделях, программном коде и других сложных документах, может потребоваться классификатор (ИИ-модуль), прогнозирующий целевую DLP, необходимую для анализа пересылаемого или сохраняемого во внешнем хранилище документа.

Выявление аномалий – классическая задача машинного обучения, которая необходима для решения задач автоматической обработки отказов и авторегистрации неисправностей. Решают эту задачу следующим образом. Анализируют на предмет первичных признаков аномалий как можно больше доступных данных, – это могут быть метрики потребления ресурсов арендаторами (тенантами), приложениями или отдельными сервисами, лог-записи с прикладных или инфраструктурных сервисов, а также телеметрия, снимаемая с ИТ-оборудования. После выявления первичные признаки аномалий агрегируются, выравниваются по шкале времени и идут на вторую стадию анализа – на вход детектору аномалий для надежного выявления аномалии и порождения события об аномалии. Детектор аномалий, а также первый уровень анализа реализуются за счёт ИИ-модулей.

Событие об аномалии используется для запуска политик обработки сбоев и авторегистрации неисправностей.

GraphRAG

GraphRAG (Graph-based Retrieval Augmented Generation) – это метод анализа и генерации текстовой информации, основанный на графовом подходе к генерации с дополненным извлечением (RAG). Этот метод использует большую языковую модель (LLM) для автоматического создания графа знаний из набора текстовых документов, полученных из корпоративных хранилищ данных.

Задачи, решаемые с помощью GraphRAG:

1. Знание структуры текста ответа. GraphRAG формирует структуру графа знаний, полученных из текстов корпоративной базы знаний до того, как пользователи начнут задавать вопросы, что позволяет системе предоставлять более точные и информативные ответы.
2. Ответы на глобальные вопросы. GraphRAG особенно эффективен при ответах на “общие вопросы” – те, которые касаются всего набора документов целиком, например, “Какие темы затрагиваются в документах такого департамента?”.
3. Улучшение полноты и разнообразия ответов. GraphRAG группирует документы до размера, соответствующего контекстному окну LLM, применяет вопрос к каждой группе для создания ответов на уровне группы, а затем объединяет все релевантные ответы в итоговую общую выдачу. Это позволяет улучшить полноту и разнообразие ответов по сравнению с традиционными методами RAG.

Microsoft предлагает следующую архитектуру GraphRAG:

- Indexer. Разделяет корпус данных на мелкие текстовые блоки (TextUnits), извлекает из них сущности, связи и ключевые утверждения.
- Clustering. Группирует данные в иерархическую структуру с использованием метода Лейдена, создавая граф знаний.
- Summarization. Генерирует обобщенные описания для каждой группы данных, что помогает в понимании контекста и смыслового связывания всей информации.
- Графовая база данных. Эта база данных хранит граф знаний, который объединяет сущности и их связи, созданные на основе неструктурированных данных.

Преимущества GraphRAG по отношению к обычному RAG:

- Графовый подход. GraphRAG использует графовую структуру для анализа и представления информации в виде графа знаний, что позволяет более эффективно обрабатывать сложные текстовые данные и извлекать из них знания.

- Анализ групп. GraphRAG может подготавливать выдачу на уровне групп, что позволяет выявлять ключевые темы и подтемы на уровне отдельных групп, а также устанавливать связи между ними.
- Улучшенная полнота и разнообразие ответов. Благодаря анализу на уровне групп и использованию графова знаний, GraphRAG способен предоставлять более полные и разнообразные ответы на запросы пользователей.

GraphRAG значительно улучшает работу языковых моделей с данными в закрытом контуре, позволяя ИИ-системе более точно и полно отвечать на сложные вопросы, требующие синтеза информации из разных источников, расположенных в закрытом контуре. В контексте GraphRAG аксиоматическое обучение может быть использовано для уточнения и расширения графа знаний, путем добавления новых сущностей, связей и утверждений на основе аксиом и правил вывода. Это позволяет системе лучше понимать контекст запросов и предоставлять более точные и релевантные ответы.

ИИ-платформы

ИИ-платформы - это платформы автоматизации разработки и эксплуатации ИИ, которые представляют собой программные системы, предназначенные для разработки, выполнения, распространения, мониторинга и эксплуатации ИИ-систем. Они обеспечивают инфраструктуру для вычислений, хранения, обмена и получения данных, на которой могут быть построены и запущены программные системы. ИИ-платформа включает в себя программные средства, инструменты, сервисы, которые формируют среду разработки, а также средства управления инфраструктурой, общие библиотеки, фреймворки, API и другие ресурсы, необходимые для создания и функционирования программных систем.

Категория “ИИ-платформы” изображена на Рисунке 3.2.3.

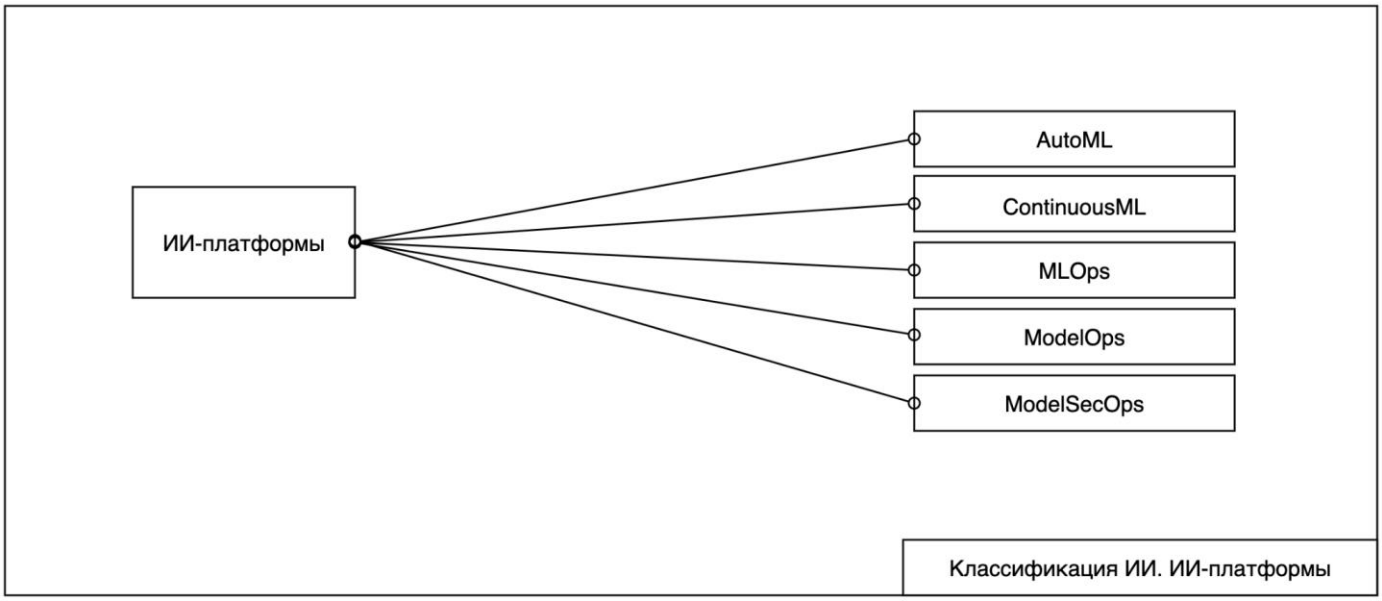


Рисунок. 3.2.3. Классификация ИИ. ИИ-платформы

AutoML

AutoML (Automated Machine Learning) – это подход и реализующая его ИИ-платформа, позволяющая автоматизировать процесс разработки и настройки моделей машинного обучения. AutoML берут на себя большую часть работы, связанной с выбором алгоритмов, оптимизацией гиперпараметров обучения, предварительной обработкой данных и другими аспектами процесса машинного обучения, что значительно сокращает время и усилия, необходимые для создания и тестирования моделей.

Задачи, решаемые с помощью AutoML:

- Упрощение процесса разработки моделей. AutoML позволяет разработчикам сосредоточиться на постановке задачи и интерпретации результатов, исключая необходимость глубоких знаний в области машинного обучения.
- Сокращение времени на разработку. Автоматизация процесса выбора алгоритма и настройки гиперпараметров ускоряет разработку моделей, позволяя быстрее получать результаты.
- Повышение эффективности команд. AutoML проверяют множество комбинаций алгоритмов и гиперпараметров, выбирая наилучшую конфигурацию для конкретной задачи, что повышает эффективность автоматически созданных моделей.
- Снижение требований к квалификации разработчиков. AutoML делает машинное обучение доступным для широкого круга специалистов, не имеющих глубоких знаний в этой области.

AutoML находит применение в различных областях, где требуется разработка и использование моделей машинного обучения, таких как наука о данных, бизнес-аналитика, здравоохранение и другие.

ContinuousML

ContinuousML (Continuous Machine Learning) – это подход и реализующая его ИИ-платформа, которая объединяет непрерывную интеграцию, непрерывную поставку (CI/CD) и непрерывное обучение (CML) в единый процесс разработки и эксплуатации моделей машинного обучения в составе ИИ-платформы. В отличие от AutoML, который фокусируется на создании моделей, ContinuousML автоматизирует весь жизненный цикл разработки и эксплуатации моделей, включая их постоянное обновление и улучшение на основе новых данных и обратной связи от пользователей.

Задачи, решаемые с помощью ContinuousML:

- Непрерывное обучение. Модели машинного обучения постоянно обновляются и улучшаются на основе новых данных и обратной связи от пользователей, что позволяет им оставаться актуальными и эффективными.
- Автоматизация развертывания. Процесс развертывания новых версий моделей автоматизирован, что позволяет быстро внедрять улучшения и исправления ошибок.
- Мониторинг и анализ. ContinuousML отслеживает производительность моделей в реальном времени, выявляет потенциальные проблемы и предлагает рекомендации по их устранению.
- Интеграция с DevOps. Непрерывная интеграция и поставка обеспечивают быструю и безопасную доставку изменений модели в среду эксплуатации или на устройства конечных пользователей.

Отличия ContinuousML от AutoML:

- Фокус на жизненном цикле. AutoML фокусируется на автоматизации создания моделей, в то время как ContinuousML охватывает весь жизненный цикл разработки и эксплуатации моделей.
- Непрерывное обучение. AutoML не обеспечивает непрерывное обучение, в то время как ContinuousML делает упор на постоянном обновлении и улучшении моделей.
- Полная интеграция с DevOps. AutoML может не использовать интеграцию с DevOps, в то время как ContinuousML тесно связывает процессы разработки и эксплуатации моделей с CI/CD.

MLOps

MLOps (Machine Learning Operations) – это методология и реализующая ее ИИ-платформа, объединяющая процессы разработки, эксплуатации и обслуживания моделей машинного обучения с целью повышения их надежности, эффективности и воспроизводимости. MLOps основан и включает в себя необходимые элементы DevOps, такие как непрерывная интеграция, непрерывная поставка и мониторинг, адаптированные для решения задач разработки и эксплуатации ИИ-систем.

Задачи, решаемые с помощью MLOps:

- Итеративная разработка. MLOps поддерживает итеративный процесс разработки, включающий сбор данных, предварительную обработку, обучение моделей, оценку, развертывание и переподготовку.
- Автоматизация. Автоматизирует рутинные задачи, такие как сборка, тестирование, развертывание и мониторинг моделей.
- Непрерывное развертывание. Обеспечивает быстрое и безопасное внедрение новых версий моделей в производство.
- Управление версиями. Позволяет отслеживать изменения в моделях и их влияние на производительность.
- Тестирование. Включает в себя тестирование моделей на различных сценариях использования.
- Воспроизводимость результатов. Гарантирует, что результаты экспериментов и разработки могут быть воспроизведены.
- Мониторинг. Отслеживает производительность моделей в реальном времени и выявляет потенциальные проблемы.

Отличия MLOps от ContinuousML:

- Фокус на процессах. MLOps фокусируется на оптимизации и автоматизации процессов разработки, эксплуатации и обслуживания ИИ-систем, в то время как ContinuousML сосредоточен на конкретных аспектах этого процесса.
- Масштабность. MLOps охватывает весь жизненный цикл ИИ-системы, от идеи до внедрения и поддержки, в то время как ContinuousML фокусируется на отдельных этапах.
- Интеграция. MLOps объединяет различные инструменты и технологии для создания целостной системы разработки и эксплуатации моделей, в то время как AutoML и ContinuousML могут быть реализованы как отдельные компоненты этой системы.

ModelOps

ModelOps (Model Operations) – это продвинутая методология и реализующая ее ИИ-платформа, направленная на управление жизненным циклом множества моделей различных ИИ-систем в масштабах большого предприятия. ModelOps фокусируется на автоматизации процессов разработки, развертывания, мониторинга и обновления моделей, а также на обеспечении их соответствия бизнес-требованиям и нормативным стандартам.

Задачи, решаемые ModelOps:

- Согласованное управление множеством моделей. ModelOps позволяет эффективно управлять большим количеством моделей, различающимся по области применения, настройке и группам клиентов.
- Мониторинг и контроль качества множества моделей. ModelOps помогает справляться с проблемами, связанными со сложностью данных мониторинга и аналитических алгоритмов, рассчитывающих метрики контроля качества, обеспечивая эффективное построение панелей управления качеством.
- Соблюдение нормативных требований. ModelOps способствует соблюдению нормативных требований, таких как защита персональных данных, обеспечивая безопасность данных и конфиденциальность информации.
- Координация разработчиков для масштабирования сложности ИИ-систем. ModelOps помогает обеспечить скоординированную работу разработчиков в множестве разнопрофильных команд, работающих над различными компонентами ИИ-системы, упрощая тем самым процесс масштабирования ИИ-систем.
- Единая среда разработки и исполнения. ModelOps обеспечивает единую среду для переноса моделей между этапами разработки, приемо-сдаточных испытаний в производством.

Отличия ModelOps от AutoML:

- Фокус на управлении моделями. ModelOps фокусируется на управлении жизненным циклом моделей, включая их разработку, развертывание, мониторинг и обновление, в то время как AutoML больше сосредоточен на автоматизации процесса выбора алгоритма и настройки гиперпараметров при обучении моделей.
- Масштаб. ModelOps применяется на уровне предприятия, охватывая множество моделей и процессов, в то время как AutoML может использоваться для создания отдельных моделей.
- Интеграция с бизнес-процессами. ModelOps тесно связан с бизнес-процессами и стратегией компании, в то время как AutoML может быть использован для решения конкретных задач создания моделей.

Отличия ModelOps от MLOps:

- ModelOps фокусируется на управлении жизненным циклом моделей, включая их разработку, развертывание, мониторинг и обновление, с акцентом на обеспечение соответствия моделей бизнес-требованиям и нормативным стандартам. ModelOps стремится к созданию единой платформы для управления всеми моделями машинного обучения в организации, независимо от их типа или области применения.
- MLOps фокусируется на автоматизации и оптимизации процессов разработки, эксплуатации и обслуживания моделей машинного обучения. MLOps включает в себя элементы DevOps, такие как непрерывная интеграция, непрерывная поставка и мониторинг, адаптированные для машинного обучения. MLOps направлен на повышение эффективности и надежности процессов разработки и эксплуатации моделей, а также на ускорение вывода моделей на рынок.

Как указано выше, отличие между ModelOps и MLOps заключается в том, что ModelOps ориентирован на управление жизненным циклом множества моделей, в то время как MLOps фокусируется на автоматизации процессов разработки и эксплуатации моделей. Следовательно, ModelOps является более масштабируемой ИИ-платформой, чем MLOps.

ModelSecOps

ModelSecOps представляет собой концепцию, объединяющую аспекты безопасности и управления моделями машинного обучения (ModelOps). Она включает в себя процессы и инструменты, направленные на обеспечение безопасности моделей, защиту данных от отравления и предотвращение несанкционированного доступа к моделям. Основное отличие ModelSecOps от ModelOps заключается в том, что ModelSecOps фокусируется не только на управлении жизненным циклом модели, но и на обеспечении ее безопасности. ModelOps сосредотачивается на процессах разработки, развертывания, мониторинга и обновления моделей, в то время как ModelSecOps добавляет к этому меры по защите моделей от киберугроз, контролю доступа к данным и моделям, а также обеспечению соответствия нормативным требованиям.

Задачи, решаемые ModelSecOps:

- Задачи, решаемые ModelOps.
- Предотвращение несанкционированного доступа к моделям и данным.
- Защита обучающих и тестовых данных от отравления.
- Проверка и фильтрация входных данных на предмет недопустимых символов и инструкций.
- Защита от запуска произвольного кода через prompt-инъекции.

- Обеспечение соответствия нормативным требованиям.
- Повышение доверия к моделям.
- Снижение рисков финансовых потерь из-за кибератак.

Меры защиты моделей и данных:

- Фильтрация входных данных и запросов пользователей. Использование доверенных алгоритмов, обеспечивающих надежную фильтрацию входных данных на предмет недопустимых символов и инструкций.
- Шифрование данных. Использование криптографических методов для защиты коммуникаций в рамках ИИ-системы, а также целостности обучающих и тестовых наборов данных.
- Аутентификация и авторизация. Контроль доступа к моделям и данным с использованием механизмов аутентификации и авторизации.
- Мониторинг безопасности. Отслеживание аномалий и потенциальных угроз безопасности в работе моделей.
- Тестирование на проникновение. Регулярное проведение симуляций атак для выявления уязвимостей в системе безопасности моделей.
- Управление доступом: Настройка прав доступа к моделям и данным на основе ролей пользователей.
- Соответствие стандартам. Обеспечение соответствия нормативным требованиям и стандартам безопасности, таким как GDPR.

ModelSecOps играет ключевую роль в защите инвестиций в разработку ИИ и предотвращении финансовых потерь из-за кибератак или нарушений конфиденциальности данных.

Подотчетные платформы

Подотчетная платформа – это ИИ-платформа с акцентом на обеспечение прозрачности, объяснимости и подотчетности результатов работы ИИ-систем в целом и их компонентов в частности. В отличие от обычной ИИ-платформы, которая может просто предоставлять инструменты для создания и использования ИИ-систем и их компонентов, подотчетная ИИ-платформа добавляет механизмы для анализа и контроля качества моделей, а также для обеспечения соответствия нормативным требованиям и стандартам этики.

Категория “Подотчетных платформ” изображена на Рисунке 3.2.4.

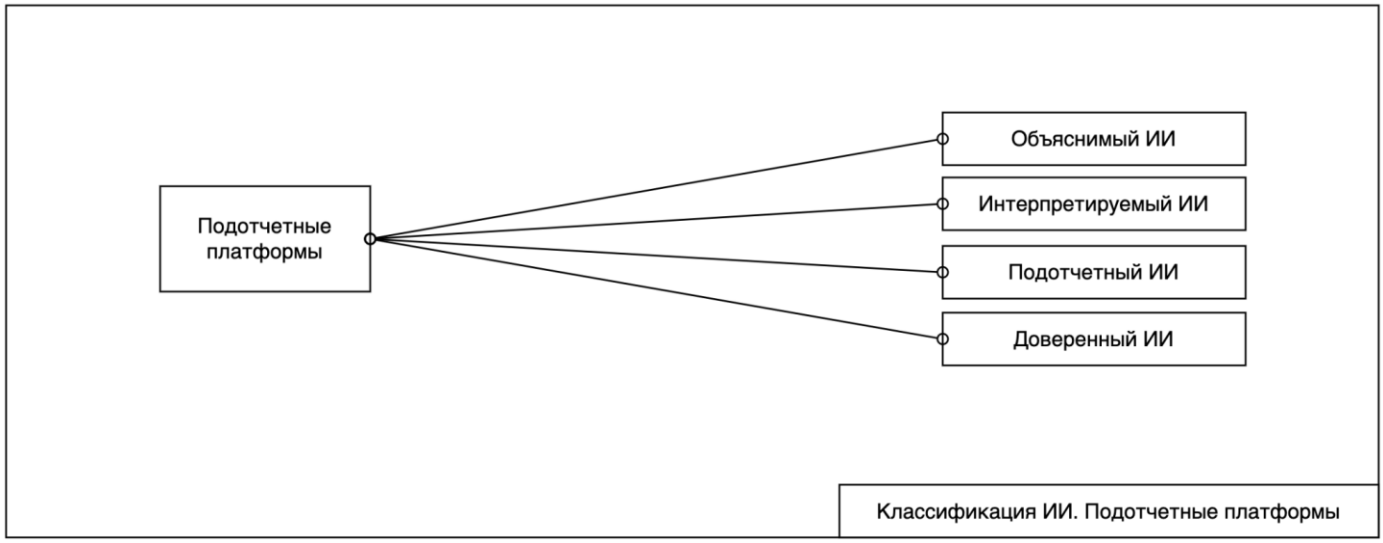


Рисунок. 3.2.4. Классификация ИИ. Подотчетные платформы

Объяснимый ИИ

Поскольку регулирующие органы, власть и обычные люди становятся зависимыми от ИИ-систем, процессы принятия решений, поддерживаемые ИИ-системами, требуют дополнительной прозрачности и подотчетности. Это необходимо, чтобы обеспечить доверие ИИ в обществе, государственных и международных отношениях. Об этом свидетельствует набирающее обороты нормативное регулирование в части объяснимости решений, принимаемых ИИ-системами.

Значимая нормативная база, устанавливающая обязательность объяснимости решений (прогнозов), принимаемых моделями машинного обучения:

- Европейский союз ввел обязанность на объяснение решений (прогнозов) моделей машинного обучения в GDPR как попытку справиться с потенциальными проблемами, вытекающими из растущей важности алгоритмов. Внедрение постановления началось в 2018 году. На текущий момент обязательство по объяснению распространяется только на локальный аспект интерпретируемости моделей. Ожидается, что в GDPR требования к объяснимости решений (прогнозов) моделей машинного обучения будут со временем ужесточаться.
- В Соединенных Штатах страховые компании должны быть в состоянии объяснить свои решения о ставках и покрытии при запросе со стороны органов юстиции. При этом для способа принятия решений не делается исключений – принимались они аналитиками или в автоматическом режиме (моделями машинного обучения).

На текущий момент объяснимый ИИ (XAI) применяется в следующих областях:

- Конструкция антенн (усовершенствованные антенны).
- Алгоритмическая торговля (высокочастотная торговля).
- Автономные транспортные средства.
- Здравоохранение (объяснимость решений, значимых для здоровья пациентов)
- Разработка детекторов признаков (компьютерное зрение).
- Юриспруденция (объяснимость юридически значимых решений).
- Страховой и банковский сектор (объяснимость выданных кредитов и страховых покрытий).
- Текстовая аналитика (извлечение фактов).
- Извлечение знаний из моделей (перенос навыка).
- Сравнение моделей машинного обучения.

Интерес общественности к XAI в мире начался в конце 2010-х. Так в 2017 году стартовала программа DARPA XAI. Программа направлена на создание моделей «стеклянных ящиков», которые можно объяснить без значительного ущерба для производительности ИИ. Пользователи-люди должны быть в состоянии понять решения ИИ (как в реальном времени, так и постфактум) и иметь возможность определять, когда следует доверять ИИ, а когда – не доверять.

Рис. 3.2.4. Описание работы объяснимого ИИ проектом DARPA XAI. Источник: DARPA.

В начале 2020-х интерес общественности к XAI в мире оформился в виде сообщества университетов, регулярных конференций и научных семинаров.

Интерпретируемый ИИ

Объяснимый и интерпретируемый ИИ – это две разных методологии, которые часто путают, но они имеют важные различия, так как основаны на разных концепциях машинного обучения:

- Интерпретируемое машинное обучение подразумевает создание моделей, которые можно легко понять и объяснить. Это особенно важно в таких областях, как медицина, юриспруденция или финансы, где необходимо точно знать, почему модель приняла определённое решение.
- Объяснимое машинное обучение, напротив, фокусируется на способности модели иметь отдельное объяснение своих решений в понятной форме. Такое объяснение помогает убедиться, что модель работает корректно и не содержит предвзятостей или ошибок.

Разница между объяснимым и интерпретируемым ИИ состоит в том что:

- Системы объяснимого ИИ (XAI) обеспечивают создание и доступность объяснений прогнозов моделей после того, как работа выполнена и предсказания сделаны. Для этого используются методы, такие как SHAP (Аддитивные объяснения Шепли) и LIME (локальные интерпретируемые объяснения, не зависящие от модели), которые помогают сделать внутреннюю работу моделей более прозрачной и понятной для пользователей.
- Системы интерпретируемого ИИ, напротив, обеспечивают создание моделей, которые по своему дизайну являются прозрачными и их легко понять. Для этого используются как можно более простые и прозрачные алгоритмы, такие как линейные модели, деревья решений, и некоторые другие, которые обеспечивают однозначность понимания без необходимости в дополнительных пояснениях.

Интерпретируемый ИИ вводит важную новацию - модель белого ящика (White-box model). Это концепция, которая позволяет разработчикам и пользователям полностью понимать и контролировать внутренние процессы и механизмы работы модели машинного обучения, что позволяет интерпретировать процесс преобразования входных данных в результаты работы модели. Модель белого ящика определяет допустимые классы алгоритмов машинного обучения, а также требует высокого уровня доверия обучающей выборки. К допустимым классам алгоритмов относятся: линейная регрессия, деревья решений,

Generalized Additive Models (GAMs). Линейная регрессия позволяет моделировать целевую переменную как линейную комбинацию входных признаков, используя метод наименьших квадратов и градиентный спуск. Деревья решений могут быть использованы для задач регрессии и классификации, позволяя разделить данные по всем признакам для минимизации функции стоимости. GAMs представляют собой мощные модели белого ящика, где целевая переменная представлена как сумма сглаживающих функций, представляющих отношение каждого из признаков и целевой переменной. Высокий уровень доверия обучающей выборки достигается путем использования аксиоматического обучения на основе валидированных экспертами графов знаний.

Модель белого ящика противоположна модели черного ящика, где внутреннее устройство модели остается неизвестным и непрозрачным. В модели белого ящика разработчики имеют полный доступ к исходному коду модели, алгоритмам, используемым для обучения и принятия решений, а также к данным, на которых модель обучалась. Модель белого ящика позволяет разработчикам детально анализировать работу моделей машинного обучения, выявлять потенциальные проблемы и улучшать их производительность. Преимущества модели белого ящика включают прозрачность, интерпретируемость, возможность оптимизации и улучшенную объяснимость. Однако разработка требует глубоких знаний в области машинного обучения. Модель белого ящика широко используется в научных исследованиях, разработке медицинских систем поддержки принятия решений и в областях, где требуется высокая степень доверия и прозрачности.

Модель черного ящика включает модели машинного обучения с высокой предсказательной силой, которые непрозрачны для интерпретации. Такие модели используются в задачах, где точность модели более важна, чем ущерб от непрозрачности внутренней работы и полученных результатов. Вместе с тем, существуют подходы, позволяющие интерпретировать модели черного ящика, такие как ансамбли деревьев и нейронные сети, при этом задача интерпретации является сложной задачей, требующей специальных техник и подходов.

В определенной степени предсказательные ограничения алгоритмов машинного обучения, допустимых в модели белого ящика, компенсируются автоматизацией процессов разработки ИИ-систем на основе графов знаний с помощью автогенераторов. Это происходит путем автоматизации построения ансамблей моделей, которые при получении входных данных документируют в специальном журнале процесс преобразования признаков по всей цепочке от самого первого до самого последнего этапа, включая результат. Далее, задокументированный процесс преобразования признаков может быть представлен в виде пояснительного отчета, в котором в однозначном виде представлен весь процесс преобразования входных данных в результат. С помощью средств

автоматизации могут быть созданы алгоритмические ансамбли большой размерности. При этом, размерность таких ансамблей ограничена только наличием вычислительных мощностей и возможностями масштабирования ИТ-инфраструктуры.

Подотчетный ИИ

«Подотчетный ИИ» неразрывно связан с двумя другими понятиями – «объяснимый ИИ» и «интерпретируемый ИИ». Вероятно, понятие «объяснимого ИИ» появилась несколько раньше, чем понятие «интерпретируемого ИИ», а понятие «подотчётного ИИ» появилась последним. Построить строгую хронологию довольно сложно, поскольку концепции, лежащие в основе этих понятий взаимосвязаны, развивались одновременно, заимствуя важнейшие принципы друг у друга. Довольно уверенно можно сказать, что объяснимый и интерпретируемый ИИ имеют более долгую историю, чем подотчётный ИИ.

Объяснимый ИИ (Explainable AI) направлен на разработку методов и подходов, которые позволяют объяснить результаты работы сложных моделей ИИ, таких как глубокие нейронные сети. Это включает в себя использование методов визуализации, локальных объяснений и других техник, помогающих понять, как модель пришла к тому или иному решению.

В отличие от этого, интерпретируемый ИИ (Interpretable AI) фокусируется на создании моделей, которые по своему дизайну легки для понимания и интерпретации. Такие модели используют простые алгоритмы, такие как деревья решений или линейные модели, что делает их структуру и процесс принятия решений наиболее прозрачной.

Подотчётный ИИ (Accountable AI) в отличии от объяснимого и интерпретируемого, акцентирует внимание на ответственности и прозрачности процессов разработки и использования ИИ-систем. Это означает, что ИИ-системы должны быть спроектированы таким образом, чтобы их работа могла быть проверена и объяснена, а также чтобы в любой момент они могли быть подвергнуты аудиту на соответствие требованиям контроля уровня качества, предвзятости и рисков ИИ, этическим и юридическим нормам. Подотчётный ИИ охватывает все аспекты ответственности и прозрачности в разработке и использовании ИИ, в то время как интерпретируемый и объяснимый ИИ сосредоточены на конкретных аспектах понимания и прозрачности моделей ИИ. Подотчетный ИИ может включать в себя ИИ-системы, разработанные в концепциях интерпретируемого и объяснимого ИИ.

Концепция подотчетного ИИ включает две новеллы: «алгоритмическую подотчетность» (алгоритмическую прозрачность) и «право на объяснение».

Алгоритмическая подотчетность - это концепция, которая определяет необходимость обеспечения прозрачности, объяснимости и ответственности в процессах принятия решений алгоритмами ИИ-систем. Она включает в себя способность отслеживать ошибки и предвзятость в алгоритмах, а также формулирует требования по разработке механизмов для их исправления и предотвращения в будущем. Важность алгоритмической подотчетности возрастает в контексте этических и социальных последствий применения ИИ, особенно в таких чувствительных областях, как здравоохранение, финансы, правосудие, управление инженерной инфраструктурой городских агломераций и других областях, где решения ИИ могут существенно влиять на жизнь людей.

Право на объяснение – это концепция, которая определяет индивидуальное право на информацию для пользователя ИИ-системы, либо лица, в отношении которого ИИ-система принимает решение. Это право гарантирует, что ИИ-система в процессе исполнения задач должна протоколировать процесс работы, включая эпизоды принятия решений, в своих внутренних журналах, чтобы по запросу пользователя в обязательном порядке сформировать юридически-значимое обоснование результатов принятых решений. Например, в случае отказа в выдаче кредита, ИИ-система должна быть в состоянии сформировать отчет и выдать справку с указанием объективных причин отказа в выдаче кредита по запросу клиента. Право на объяснение может быть создано либо на уровне закона, либо как профессиональный стандарт в одной или группе связанных областей. Без реализации этого права на уровне архитектур ИИ-систем общественность не имеет технической возможности оспаривать решения автоматизированных систем. Это является дискриминацией права на возможность оспаривания незаконного или ошибочного решения организации, от лица которой действует ИИ-системы.

Доверенный ИИ

Наиболее полной и строгой концепцией ИИ-систем, разрабатываемых для использования в областях с высокой социальной значимостью и ответственностью, является концепция доверенного ИИ. Эта концепция объединяет и систематизирует положения и методики, лежащие в основе объяснимого, интерпретируемого и подотчетного ИИ, а также вводит дополнительные новеллы.

Доверенный ИИ — это концепция разработки и эксплуатации систем искусственного интеллекта, гарантированно обладающих свойствами надежности, безопасности, эффективности, продуктивности, прозрачности, конфиденциальности, справедливости и

этичности получаемых результатов. Все это предполагает подотчетность соответствующих решений, созданных в доверенной среде разработки и эксплуатации, в рамках концепции доверенного продукта в составе платформы, на основе доверенных алгоритмов, доверенных карт знаний, аксиоматического обучения, доверенной среды кодогенерации и доверенного GraphRAG (Graph-based RAG) – метода анализа и генерации информации, основанного на графовом подходе с дополненным извлечением данных из внешних источников (Retrieval Augmented Generation, RAG).

ИИ-платформа, соответствующая концепции доверенного ИИ, гарантирует следующий набор свойств продуктов, созданных на ее основе и исполняемых в ее составе:

1. Надежность. Платформа и продукты работают предсказуемо, без сбоев или ошибок, особенно в условиях неопределенности.
2. Безопасность. Платформа и продукты защищены от взломов и кибератак.
3. Эффективность. Платформа и продукты обеспечивают быстрое и точное выполнение задач с использованием доступных ресурсов.
4. Продуктивность. Платформа обеспечивает максимизацию производительности труда разработчиков продуктов, а продукт – пользователей.
5. Подотчетность. Платформа обеспечивает возможность полного надзора и контроля над действиями разработчиков, а продукты – над поведением и результатами работы ИИ.
6. Прозрачность. Платформа и продукты обеспечивают возможность предоставления доступа к информации о функционировании и принятии решений ИИ-продуктами разработчикам и пользователям.
7. Конфиденциальность. Платформа и продукты обеспечивают механизмы защиты и контроля доступа к собираемым и генерируемым данным.
8. Справедливость. Платформа и продукты обеспечивают проверки предвзятости и учет интересов всех заинтересованных лиц, на кого распространяются результаты работы платформы и продуктов.
9. Этичность. Платформа и продукты обеспечивают соблюдение принципов взаимодействия между людьми и системами, направленными на расширение возможностей людей, а не на их ограничение.

Примером реализации концепции доверенного ИИ является HealthOS [https://healthops.ru/]. Это платформа доверенного медицинского ИИ для автоматизации лечебных процессов в реаниматологии, анестезиологии, кардиологии, хирургии и других областях, а также в научных и клинических исследованиях. HealthOS позволяет в десять раз повысить эффективность разработки и внедрения в рутинную медицинскую и научную практику доверенного ИИ.

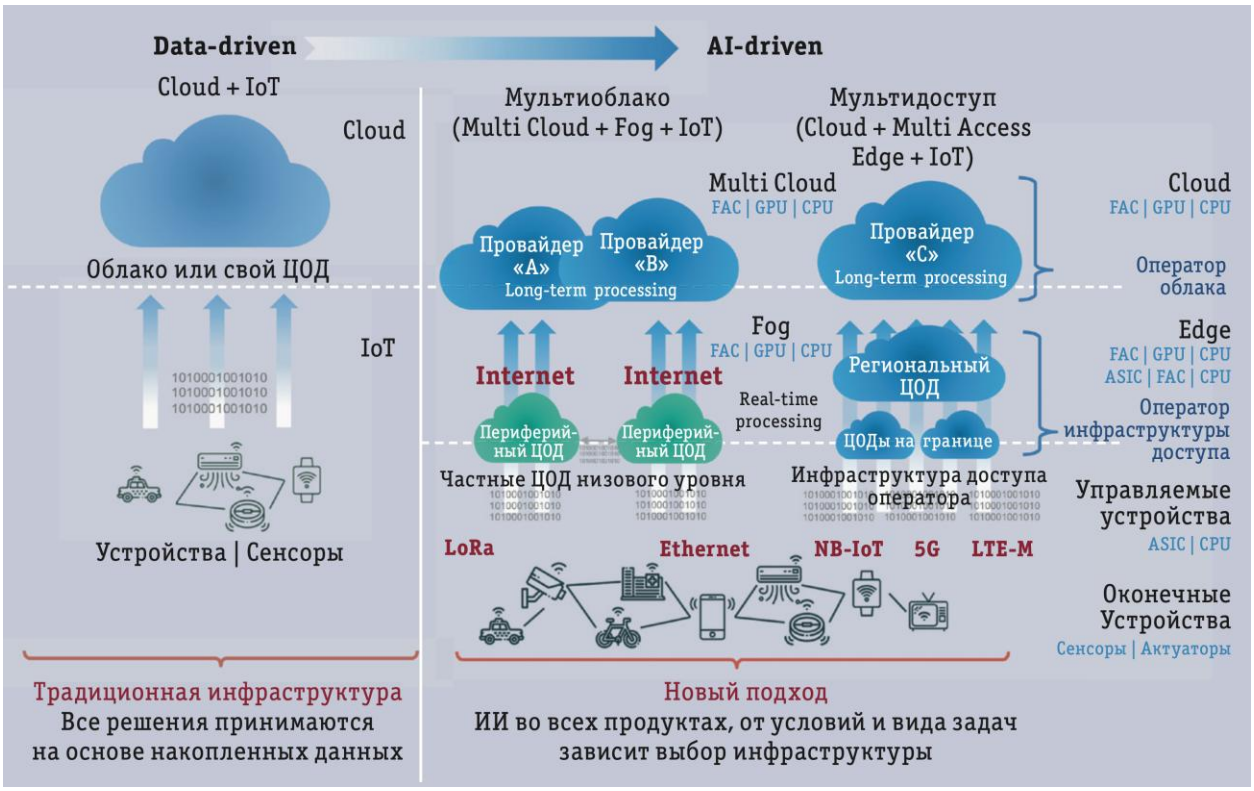
3.2.4. Вычислительная архитектура

Для развертывания и исполнения моделей ИИ требуется гибкая, масштабируемая и отказоустойчивая инфраструктура. Автомасштабирование кластеров до сотен тысяч вычислительных узлов позволяет эффективно обрабатывать большие объемы данных и обеспечивать высокую производительность систем ИИ. Взаимодействие ИИ и систем обработки больших данных осуществляется при помощи упаковки алгоритмов машинного обучения в микросервисы и интеграцию их с инфраструктурой хранения и обработки данных. NPU/GPU акселераторы ускоряют процесс обучения и работы моделей, обеспечивая более высокую производительность и эффективность.

Как известно, обучение больших языковых моделей типа GPT4 требует вычислительную инфраструктуру, которая обеспечивает создание и управление вычислительными кластерами размером до 150 тысяч вычислительных узлов. При этом, обучение и эксплуатация узко-специализированных решений требует вычислительную инфраструктуру из одного или нескольких вычислительных узлов. Вместе с тем, так же имеет место архитектура моделей.
ИИ для обучения может потребовать сотни тысяч вычислительных узлов, вместе с тем, при его использовании, например, для прогноза сбоев логистики грузов, могут потребоваться десятки миллионов датчиков, встроенных в морские контейнеры и распределенных по территории всего земного шара. Архитектура такого рода глобальных систем получила название “гиперскейла”.

Архитектура гиперскейла

В обобщенном виде архитектура гиперскейла имеет следующий вид.



Источник изображения - статья “Базовая инфраструктура современных цифровых платформ” [https://www.osp.ru/os/2021/04/13056073]

На рисунке представлены варианты компоновки ИИ-систем в архитектуре гиперскейла. Здесь имеются две отправные точки — наличие либо отсутствие у компании ЦОДов, близких к источникам данных. В первом случае целесообразно говорить о варианте «мультиоблака с туманными вычислениями» (Multi Cloud + Fog Layer): ЦОДы горизонтально объединяются между собой и образуют уровень Fog Layer, где исполняются приложения, критичные к задержкам по времени. Остальные приложения поднимаются на уровень выше — в мультиоблако, где ресурсы дешевле. Уровень мультиоблака необходим по разным причинам: отсутствие зависимости от поставщика, более низкая стоимость вычислительных ресурсов, более короткие передачи данных с нижележащего слоя. Вариант мультиоблака подходит для отраслеобразующих или больших компаний, обеспечивающих критичную для государства инфраструктуру. При отсутствии своих ЦОДов, близких к источникам данных, используется вариант облака с мультидоступом (Cloud + Multi Access Edge), при котором функции слоя туманных вычислений выполняет двухярусный слой граничных вычислений. Первый ярус исполняется на мини-ЦОДах, которые находятся в непосредственной близости от границы сети последней мили. Второй ярус исполняется на региональных ЦОДах, где стоимость ресурсов дешевле, но и задержки больше. Некритичные к задержкам компоненты ИИ выносятся на уровень вычислительного

ядра главных ЦОДов, где стоимость вычислительных ресурсов самая низкая. Вариант облака с мультидоступом более всего подходит для компаний, имеющих обширную или не определенную в пространстве географию поддержки бизнес-процессов, — например, для логистических компаний. Этот вариант подходит и для SMB-компаний, для которых зависимость от провайдера инфраструктуры не критична.

Указанные подходы к разработке ИИ опираются на типовые варианты ПАКов, позволяющих собирать физическую инфраструктуру для разных уровней, соответствующих конкретным инфраструктурным доменам: Compute — вычислительные ресурсы; Storage — системы хранения; CEP&AI — решения для обработки сложных событий в реальном времени, хранения данных, обучения и исполнения моделей машинного обучения; Network — решения для построения комплексных сетей предприятия.

Домен Compute, как правило, представлен x86-системами для поддержки общих нагрузок, таких как виртуализация, контейнеризация, и иногда нагрузок класса «сервер без гипервизора» (bare-metal). ИТ-гиганты, в отличие от остальных компаний, активно развивают собственные серверные решения на базе архитектуры ARM, обладающих пока лучшими показателями энергоэффективности и соотношения цена/производительность. Яркий пример такого подхода — компания Amazon, выпускающая серверы собственной разработки на базе процессоров Graviton и продающая их в составе вычислительных ресурсов своего облака.

Домен Storage развивается под влиянием расширения границ облака и роста объемов данных. Многие компании отказываются от классических массивов сетей хранения (SAN) в пользу распределенных программно-конфигурируемых хранилищ (Software-defined storage, SDS). Однако остаются нагрузки, для которых подходят классические решения SAN или массивы с прямым подключением (DAS), но в условиях расширения количества приложений в микросервисной архитектуре все более популярными становятся SDS-системы, которые, как правило, строятся на серверных мощностях x86-архитектуры либо на базе специализированных ПАК.

Домен CEP&AI требует особого внимания: правильно выстроенный конвейер для всех фаз ModelOps (ModelSecOps) в совокупности с оптимизированными ПАК позволяет сократить время вывода продукта на рынок, время на обучение ИИ, повысить производительность приложения, эффект от автоматизации и снизить стоимость вычислений в пересчете на операцию. Здесь возможны следующие типы программно-аппаратных комплексов:

- BigData — ПАК для обработки и хранения больших данных, может комплектоваться GPU для ускорения обработки задач, например, на Apache Spark.
- ML — ПАК с аппаратными средствами ускорения в ЦОДах для обучения и исполнения ИИ. Как правило, для фазы обучения моделей используются

конфигурации с GPU, в то время как для фазы исполнения моделей — узлы с x86-процессорами и GPU, FPGA и специализированными схемами (ASIC). Выбор конфигурации для исполнения моделей в ЦОДе напрямую связан с решаемой задачей ИИ: так, например, для рекомендательных систем лучше CPU, в то время как для задач синтеза и распознавания речи лучше GPU. Для задач компьютерного зрения применяются разнообразные ASIC — по мнению аналитиков, в будущем ASIC обойдут GPU для фазы исполнения моделей. На данный момент FPGA относительно редко применяются на открытом рынке в силу их высокой стоимости, но обладают неоспоримым преимуществом: их можно перепрограммировать. Лидер на рынке GPU и платформ для искусственного интеллекта — компания Nvidia. Однако ИТ-гиганты идут своим путем: выпускают свои ASIC для обучения и исполнения моделей, продавая их в составе вычислительных ресурсов своих облаков (Google TPU, Amazon Inferentia и Trainum).

- Edge AI — системы с аппаратными средствами ускорения решения задач ИИ при их выполнении в региональных ЦОДах или в специализированных конфигурациях в компакт-факторе. Расширение границ вычислений сместило вычислители ближе к источникам данных. Например, так строятся системы видеоаналитики: сервер или коробочное решение с GPU/ASIC обрабатывает данные с камер, распознавая номера автомобилей, а затем передавая их в текстовом формате в ЦОД для дальнейшей обработки, что снижает нагрузку на каналы связи.
- Управляемые устройства оснащаются специализированными ASIC для исполнения ИИ непосредственно на источнике данных, что позволяет обработать и извлечь выгоду без передачи данных. Пример оконечного устройства - умная камера.

Домен Network оккупировали ведущие мировые поставщики сетевого и телекоммуникационного оборудования (Cisco, Huawei и другие компании), в продуктовом портфеле которых присутствуют решения различного масштаба. Скорость передачи данных в облаках между вычислительными узлами составляет сегодня от 100 Гбит/с, и многие компании стоят перед выбором: расширить каналы связи между ЦОДами и граничными устройствами или разместить граничные ЦОДы ближе к источникам данных. Набирает популярность и подход к построению сетевых решений на базе сетевых устройств, отвязанных от конкретных фирменных решений, на которых можно установить ПО категории Open Source.

3.2.5. Риски и регулирование ИИ

ИИ не только создает новые рынки, но и несет огромные риски совершенно разного системного уровня и значимости их последствий – от нанесения физического вреда человеку и негативного влияния на его психическое здоровье до подрыва национальной безопасности страны

и возможного влияния на возникновение техногенных катастроф. Ведутся постоянные дискуссии вокруг экзистенциальных рисков, вызванных пока еще непонятными последствиями развития ИИ, злоупотреблением ИИ со стороны преступного сообщества или автономными действиями ИИ, в том числе на основе ИИ агентов. Регулирующие органы во всем мире пытаются не только выработать, но и сбалансировать набор мер регулирования таким образом, чтобы сдерживать риски ИИ, но при этом не подавлять инновации и максимизировать пользу от применения ИИ.

Как разработка, так и применение систем на основе ИИ может создавать этические проблемы, на которые не всегда есть четкие ответы. Например, алгоритмы и модели ИИ могут применяться при разработке новых молекулярных структур химических элементов, включая различные полимерные материалы, где надо перебирать миллиарды различных комбинаций для подбора компонент. Это «благо» и возможность. Однако, эта же технология и алгоритмы могут использоваться для создания медицинских препаратов с запрещенными свойствами. Это «плохо» и это риск.

Технологии GenAI на основе больших языковых моделей могут использоваться для реферирования информации, для написания дипломов, для создания диалоговых чат ботов. Когда такой бот используется для консультаций по истории Римской империи, то последствия ошибок для пользователя некритичны. Но если такой бот дает консультации по медицине, на основе которых назначается лечение, но цена ошибки очень велика.

Как следствие, у государства возникает вопрос, как регулировать ИИ, чтобы он приносил максимальную пользу при минимальном риске его применения? Если нет четких правил игры, то как частные пользователи – граждане, так и бизнес-пользователи - предприятия и организации не будут доверять технологии ИИ. Что влияет на потребление и развитие рынка и на рост инвестиций в технологии ИИ? Многие опасаются, что в будущем ИИ может заменить многие профессии и привести к сокращению рабочих мест.

Регулирование ИИ – это пока еще «открытая книга». Хотя во многих странах уже приняты базовые законы о регулировании ИИ, практика их применения почти отсутствует. Более того, технологии настолько быстро развиваются, что регулирование не успевает за ними. Как следствие, регуляторы принимают очень неконкретные, «рамочные» законы, которые описывают в основном этические принципы на основе подхода ex-ante (предписывающее регулирование, заранее вводятся императивные нормы, например, прямые запреты), так как сам стек технологий в момент разработки законодательства не определен и в нормативных актах используется очень широкое определение ИИ.

Борьба мнений по основным принципам регулирования отражает объективную картину: с одной стороны, имеется опасность регулирования технологий ИИ на ранней стадии их развития, что может «заморозить» их, с другой стороны, в силу отставания темпов разработки регулирования от темпов развития технологий мы сталкиваемся со слабостью самих нормативных предложений и отсутствии практики правоприменения. Ряд экспертов считают гораздо более важной задачей разработку и применение технологических стандартов ИИ, чем разработку всеобъемлющего регулирования.

На концептуальном уровне к «традиционным» угрозам информационных технологий, которые возникают при размещении систем в облачных средах, ИИ добавляет ещё три категории угроз: «отравление» данных обучающего корпуса, «отравление» промпта и «отравление» контекста. Эти угрозы связаны с особенностями работы ИИ и требуют особого внимания при разработке и эксплуатации систем на основе ИИ в облачных средах.

«Отравление» данных обучающего корпуса — это преднамеренное искажение или добавление ложных данных в обучающий корпус, который используется для обучения ИИ. Это может привести к

тому, что модель будет выдавать неверные результаты или принимать неправильные решения. Например, если в обучающий корпус добавить большое количество ложных данных о том, что определённый объект всегда находится в определённом месте, то модель может начать ошибочно определять местоположение этого объекта.

«Отравление» промпта — это манипуляция с текстом запроса (промпта), который подаётся на вход ИИ. Промпт может содержать ключевые слова или фразы, которые направляют модель к определённому результату. Если в промпт добавить скрытые или неочевидные подсказки, то модель может выдать желаемый результат, даже если он не соответствует истине. Например, если в промпт добавить фразу «только для взрослых», то модель может начать выдавать контент, предназначенный только для взрослых, даже если изначально он был предназначен для всех возрастов.

«Отравление» контекста — это изменение контекста, в котором работает ИИ, таким образом, чтобы она выдавала желаемые результаты. Контекст может включать в себя различные факторы, такие как время, место, условия и т.д. Если изменить контекст таким образом, чтобы он соответствовал желаемому результату, то модель может начать выдавать этот результат, даже если он не является правильным. Например, если модель обучена распознавать объекты на изображениях, то изменение условий освещения или фона может повлиять на её способность правильно распознавать эти объекты.

Эти угрозы требуют разработки специальных методов защиты и контроля качества данных, а также использования алгоритмов обнаружения и предотвращения атак на системы ИИ. Важно также проводить регулярные проверки и аудит систем на предмет наличия признаков «отравления» данных, промпта и контекста.

Можно констатировать, что в мире отсутствует единое мнение по тому, как и в какой степени необходимо регулировать ИИ. Например, позиции ЕС, Великобритании, Китая и США не только сильно отличаются, но и окрашены национальной спецификой, философией ведения бизнеса и подходами к регулированию. В итоге на международном уровне регулирование ИИ представляет из себя сильно фрагментированную и сложную нормативную среду, в которой присутствуют противоречия между подходами, например ex-ante и ex-post.

3.2.6. Направления развития

Приведем оценку направлений развития компаний в области ИИ на ближайшие 10 лет:

- Демократизация генеративного ИИ. Развитие инструментов и платформ, делающих генеративный ИИ доступным для широкого круга пользователей, не имеющих специализированных технических знаний.
- Развитие мультимодального ИИ. Интеграция различных модальностей данных, таких как текст, изображения, видео, аудио, для создания более точных и комплексных решений.
- Разработка ИИ с учетом безопасности. Создание систем ИИ в виде независимых продуктов в составе платформ, которые позволяют обнаруживать и предотвращать кибератаки, а также обеспечивать защиту персональных данных.

- Регулирование технологий и продуктов ИИ. Разработка и внедрение стандартов и норм для регулирования использования ИИ, особенно в сферах, где требуется высокая степень ответственности, таких как медицина и финансы.
- Инсорсинг разработки платформ экосистем. Развитие собственных платформ ИИ и защита интеллектуальной собственности для снижения зависимости от поставщиков и обеспечения собственной безопасности. Платформы экосистем обеспечивают интеграцию продуктов и вспомогательных сервисов, упрощающих взаимодействие между независимыми участниками рынка.
- Роботизация производства. Внедрение роботов систем для повышения эффективности производственных процессов.
- Облачные вычисления и распределенная инфраструктура. Ускоренное развитие облачных сервисов и инфраструктуры для обработки и хранения больших объемов данных.
- Кибербезопасность и защита данных. Повышение мер безопасности для защиты данных компании от кибератак и утечек информации.
- Интернет вещей (IoT). Создание вертикально-интегрированных решений на базе взаимосвязанных устройств и систем для сбора данных и управления ими в реальном времени.
- Дистанционное обучение. Расширение возможностей дистанционного обучения и образования с использованием ИИ и виртуальных сред.
- Персонализация клиентского опыта. Применение ИИ для создания индивидуальных предложений и улучшения качества обслуживания клиентов.

Отдельно следует отметить гипераутоматизацию. Благодаря гипераутоматизации компании и органы государственного управления смогут быстрее внедрять новые технологии, адаптировать свои бизнес-стратегии к изменяющимся условиям рынка, снижать сроки создания и себестоимость единицы продукции.

Гипераутоматизация — это актуальная технологическая тенденция, которая расширяет возможности автоматизированных рабочих процессов, делая их значительно более эффективными. Она предполагает замену человеческого участия в физических и цифровых задачах, включая процессы, требующие принятия решений, с помощью компьютеров и программного обеспечения.

Пять ключевых элементов гипераутоматизации:

1. Повсеместное использование ИИ (AI-driven);
2. Роботизация бизнес-процессов (RPA);

3. Повторное использование информационных ресурсов посредством магазинов и контент-сервисов (CSP);
4. Порталы самообслуживания пользователей (Low-no-code portal) для создания индивидуальных помощников;
5. Корпоративные системы на основе классической автоматизации для выполнения фиксированных потоков работ.

Стратегия гипераутоматизации позволяет компаниям быстрее реагировать на изменения на рынке и быть более гибкими, чем предприятия со сложными бизнес-процессами, реализованными в «классических автоматизированных системах».

Гипераутоматизация тесно связана с ИИ, поскольку он играет ключевую роль в решении сложных задач, таких как анализ больших объёмов данных, работа с неструктурированной информацией, адаптация к меняющимся условиям. ИИ позволяет выполнять процессы, которые ранее требовали вмешательства человека, повышая эффективность и точность выполнения задач.

Влияние гипераутоматизации на корпоративный ландшафт заключается в изменении подходов к управлению процессами, повышении операционной эффективности, снижении затрат и улучшении качества обслуживания клиентов. Компании, внедряющие гипераутоматизацию, могут ожидать увеличения производительности, улучшения качества продукции и услуг, а также повышения конкурентоспособности на рынке.

Изменения, связанные с гипераутоматизацией, будут происходить в различных временных горизонтах, начиная от краткосрочных улучшений в эффективности отдельных процессов и заканчивая долгосрочными стратегическими изменениями в структуре и культуре организации. Скорость внедрения гипераутоматизации зависит от специфики отрасли, размера компании и готовности к изменениям.

Глоссарий

Алгоритм: набор инструкций, описывающих последовательность действий исполнителя для решения некой задачи. Может иметь форму математической формулы или программы на формальном языке исполнителя (вычислителя).

Байесовские сети: также известные как байесовская сеть, байесовская модель, сеть убеждений и сеть принятия решений, представляет собой модель на основе графа, представляющую набор переменных и их зависимости.

Большие данные: массив структурированных и/или неструктурированных данных, которые слишком сложны для обработки с помощью «стандартного» программного

обеспечения для обработки данных. Существуют разные мнения по поводу нижней границы объема, после которого начинаются «большие данные». Общий критерий — большие данные требуют массивно-параллельных архитектур, чтобы обеспечить приемлемое (малое) время обработки запросов.

Большая языковая модель (БЯМ, Large Language Model, LLM): это модель глубокого обучения, состоящая из нейронной сети с большим количеством параметров (обычно миллиарды весовых коэффициентов и более), обученная на огромном объеме неразмеченного текста с использованием обучения без учителя. Эти модели способны справляться с широким спектром задач, включая генерацию текста, перевод, ответы на вопросы и многое другое. Они отличаются от специализированных моделей, обученных для выполнения одной конкретной задачи, своей универсальностью и способностью адаптироваться к различным контекстам.

Гиперавтоматизация (Hyperautomation): это процесс и результат процесса по автоматизации максимального количества повторяющихся задач и процессов в организации с целью повышения эффективности и снижения затрат. Она включает в себя использование технологий искусственного интеллекта, машинного обучения и роботизации процессов для автоматизации таких задач, как сбор и анализ данных, принятие решений и выполнение задач. Гиперавтоматизация позволяет организациям сократить время выполнения задач, повысить качество результатов, снизить количество ошибок.

Делинеация: это процесс разделения непрерывного сигнала на составные части, слои или шкалированные признаки. Например, делинеация ЭКГ означает разделение сигнала ЭКГ по шкале морфологических признаков кардиоцикла: значения пиков, зубцов, интервалов и базовой линии. При этом, каждый морфологический признак кардиоцикла имеет свое наименование, патогенетическую интерпретацию и набор значений. Делинеация помогает улучшить качество анализа и интерпретации непрерывных сигналов, а также обнаружить аномалии или отклонения от нормы.

Доверенный ИИ (Trusted AI): это процесс и результат процесса разработки ИИ, который обеспечивает надежность, безопасность, эффективность и продуктивность при его использовании. Концепция доверенного ИИ является ответом на рост недоверия ИИ и вызовам, связанным с разработкой и внедрением технологий ИИ, которые должны быть надежными, безопасными и этичными в использовании.

Интеллектуальный анализ данных: процесс обработки данных с целью выявления повторяющихся шаблонов, признаков или закономерностей и установления взаимосвязей между ними.

Интеллектуальная автоматизация (Intelligent Automation – IA): форма роботизации бизнес-процессов путем полной замены операций, выполняемых человеком, на программные алгоритмы и/или механические устройства. Полная замена операций предполагает замену интерфейса пользователя на программные интерфейсы, достаточные для взаимодействия алгоритмов между собой. Конечной точкой последовательной замены человека на алгоритм является полностью автоматический бизнес-процесс. Такие автоматические бизнес-процессы называют «безлюдными процессами».

Интеллектуальная промышленная автоматизация (Intelligent Industrial Automation — IIA), Интеллектуальная автоматизация зданий (Intelligent Building Automation — IBA) – прикладные направления интеллектуальной автоматизации, сочетающие в себе механизацию и использование ИИ с целью последовательной и, по возможности, максимально полной замены людей на умные машины.

Интерпретируемый ИИ (Interpretable AI): это процесс и результат процесса по разработке систем ИИ, при котором алгоритмы и модели ИИ создаются таким образом, чтобы их можно было легко понять и объяснить пользователю. Это достигается за счет применения методов и техник, которые делают процесс принятия решений ИИ прозрачным и доступным для интерпретации. В отличие от традиционных моделей ИИ, которые могут быть сложными и непрозрачными, интерпретируемый ИИ стремится сделать процесс принятия решений понятным и объяснимым. Это важно для многих областей применения ИИ, таких как медицина, финансы и юриспруденция, где требуется высокий уровень доверия и прозрачности. Интерпретируемый ИИ использует различные методы, такие как логические правила, деревья решений, линейная регрессия и другие, которые позволяют анализировать и понимать, как ИИ пришел к тому или иному решению. Это помогает избежать предвзятости и ошибок, а также повышает доверие к результатам работы ИИ.

Индуктивное смещение алгоритма машинного обучения (смещение обучения) — это набор предположений, которые алгоритм использует для прогнозирования результатов входных данных, с которыми он не сталкивался ранее.

Искусственный узкий интеллект (ANI — Artificial Narrow Intelligence): также известный как слабый AI, ANI — это тип искусственного интеллекта, который может сосредоточиться только на одной задаче или проблеме в данный момент времени (например, играя в игру против человека в шахматы). Это распространенная на сегодняшний день форма ИИ.

Карта знаний (Knowledge Map - KM): это логически и семантически согласованное машиночитаемое представление пространства поименованных признаков заданной предметной области, выполненное на основе единого словаря или онтологии. Карта

знаний может иметь статус доверенной. ***Доверенная карта знаний*** *— это карта знаний, полностью интегрированная в прикладное научное знание, соответствующая нормативной документации и этическим нормам предметной области, легализованная путем получения положительных заключений Этического комитета и Ученого совета авторитетного научного или медицинского центра, либо авторитетной некоммерческой организации. Карты знаний используются для решения разных задач: разработка доверенного ИИ, разработка интерпретируемого ИИ, семантическая совместимость знаний, автоматический обмен знаниями и др.*

Когнитивные вычисления*: программная модель, которая решает задачи на распознавание, логику и другие задачи, свойственные человеку, с помощью интеллектуального анализа данных, обработки естественного языка и распознавания образов.*

Машинное зрение *(Machine vision – MV): это применение технического зрения для промышленности и производства. Машинное зрение является подразделом инженерии и связано с вычислительной техникой, оптикой, машиностроением и промышленной автоматизацией.*

Машинное обучение *(Machine learning – ML): фокусируется на разработке алгоритмов, которые получают доступ к данным и используют их сами по себе, при этом алгоритм обучается самостоятельно, извлекая из входных данных компетенцию в виде признаков и их весов, сохраняя компетенцию в виде внутренней базы данных.*

Нейронная сеть, искусственная нейронная сеть *(ANN — Artificial Neural Network): математическая модель (а также её программное или аппаратное воплощение), построенная на основе отдельных принципов организации и функционирования биологических нейронных сетей. Нейросеть представляет собой систему соединённых и взаимодействующих между собой процессоров (искусственных нейронов). Такие процессоры довольно просты, однако, будучи соединёнными в достаточно большую сеть с управляемым взаимодействием, такие процессоры способны сообща выполнять довольно сложные задачи. С точки зрения машинного обучения, нейросеть представляет собой частный случай методов распознавания образов, дискриминантного анализа, методов кластеризации и т.п. С точки зрения математики, обучение нейросети — это многопараметрическая задача нелинейной оптимизации. С точки зрения кибернетики, нейросеть используется в задачах адаптивного управления и как алгоритмы для робототехники. С точки зрения вычислительной техники и программирования, нейросеть — способ решения задачи эффективного параллелизма.*

Нейросеть-трансформер, Трансформер *(Transformer): это популярная архитектура глубоких нейронных сетей, разработанная в 2017 году исследователями из Google Brain.*

Она предназначена для обработки последовательностей, таких как текст на естественном языке, и решения задач, связанных с машинным переводом и автоматическим реферированием. В отличие от рекуррентных нейронных сетей, трансформеры не требуют обработки последовательностей по порядку, что позволяет им распараллеливаться легче и обучаться быстрее.

Обработка естественного языка (Natural Language Processing — NLP): набор концепций, методов и алгоритмов для решения задач, связанных с анализом и интерпретацией естественного языка (т. е. языка, на котором говорят и пишут люди). К числу таких задач относятся машинный перевод с одного языка на другой, поиск фактов, сущностей, связей между ними в больших массивах текстовых документов, построение диалоговых систем, включая персональных голосовых помощников, и сотни других применений.

Объяснимый ИИ (eXplainable AI - XAI): фокусируется на предоставлении пост-специальных объяснений прогнозов модели, используя методы вроде важности признаков или механизмов привлечения внимания. Его цель - сделать внутреннюю работу модели более прозрачной и понятной для пользователей после того, как модель сделала прогноз. Интерпретируемое машинное обучение, напротив, подчеркивает присущую модели простоту и понятность. Оно стремится создавать модели, которые по своей природе легче понять, например, деревья принятия решений или линейные модели. Интерпретируемый ИИ предлагает такое модели, которые обычно проще и понятнее для людей без дополнительных объяснений.

Обучение с учителем: тип машинного обучения, при котором обучающие наборы данных снабжены правильными ответами, что обучает машину генерировать целевые результаты (аналогично отношениям между учителем и учеником).

Обучение без учителя: тип машинного обучения, при котором алгоритм обучается с информацией, которая не маркируется правильными ответами, что позволяет алгоритму действовать без руководства (или надзора).

Обучение признакам (feature learning) или обучение представлению (representation learning) – это методы машинного обучения, которые позволяют системе автоматически обнаруживать признаки, необходимые для обнаружения объектов или выполнения классификации объектов. Это заменяет ручное проектирование алгоритмов и позволяет машине определять необходимые алгоритмы и использовать их для выполнения конкретной задачи.

Обучение с подкреплением: метод машинного обучения, при котором алгоритм обучается, взаимодействуя с окружающей средой, а затем «штрафуется» за неверные или «поощряется» за верные решения.

Общий искусственный интеллект (AGI — Artificial General Intelligence): также известный как сильный AI, AGI — это тип искусственного интеллекта, который считается человекоподобным и все еще находится на начальной стадии разработки (в настоящее время скорее гипотетическое существование).

Подотчетность ИИ – это комплекс мер, направленных на обеспечение прозрачности, ответственности и контроля над действиями и решениями, принимаемыми алгоритмами ИИ-системы. Это включает в себя способность ИИ-системы объяснять свои действия, предоставлять обоснования для принятых решений и быть открытой для аудита и проверки.

Подотчетность ИИ важна для обеспечения доверия к технологиям ИИ со стороны общества и пользователей, а также для предотвращения возможных злоупотреблений и ошибок. Она требует разработки соответствующих механизмов и стандартов, которые бы обеспечивали соблюдение этических норм и законодательства в области ИИ.

Основные аспекты подотчетности ИИ:

- *Прозрачность. Возможность понимания того, как работает ИИ-система, какие данные используются и как принимаются решения.*
- *Ответственность. Способность оператора ИИ-системы нести ответственность за действия и решения ИИ-системы, включая возможность исправления ошибок и компенсации ущерба.*
- *Аудит и проверка. Наличие механизмов для аудита и проверки работы ИИ-системы, чтобы убедиться в ее соответствии установленным стандартам и нормам.*
- *Соблюдение этических норм. Соблюдение этических принципов и норм при разработке и использовании ИИ-системы, таких как уважение к частной жизни, справедливость и непредвзятость.*

Для достижения подотчетности ИИ необходимо активное участие всех заинтересованных сторон, включая разработчиков, исследователей, регуляторов и общество в целом.

Рекуррентная нейронная сеть (Recurrent neural network – RNN): вид нейронных сетей, где связи между элементами образуют направленную последовательность. Благодаря этому появляется возможность обрабатывать последовательные цепочки, для чего рекуррентные сети используют свою внутреннюю память для обработки последовательностей произвольной длины. Поэтому сети RNN применимы в таких

задачах, где нечто целостное разбито на части, например: распознавание рукописного текста или распознавание речи. Было предложено много различных архитектурных решений для рекуррентных сетей — от простых до сложных. В последнее время наибольшее распространение получили сети с долговременной и кратковременной памятью (LSTM) и управляемый рекуррентный блок (GRU).

Роботизированные процессы автоматизации (Robotic Process Automation – RPA): форма роботизации бизнес-процессов путем использования интерфейса пользователя для сбора данных и управления приложениями вместо человека. Использует имитацию действий пользователя. Имитация основана на записи и воспроизведении повторяющихся операций пользователя.

Семантическая совместимость знаний (Knowledge Semantic Interoperability - KSI): это способность компьютерных систем обмениваться данными с однозначным общим значением. Это означает, что системы могут однозначно интерпретировать данные друг друга, независимо от того, какие они используют формы представления и протоколы обмена данными. Достигается это через добавление метаданных к каждому элементу данных, связывая его с контролируемым общим словарем или онтологией. Это позволяет системам автоматически определять значение данных без необходимости вмешательства человека.

Свёрточная нейронная сеть (Convolutional neural network – CNN): архитектура нейронных сетей, предложенная Яном Лекуном в 1988 году и нацеленная на эффективное распознавание образов, входит в состав технологий глубокого обучения (deep learning). Идея CNN-сетей заключается в чередовании свёрточных слоёв (convolution layers) и слоёв подвыборки (pooling layers). Структура CNN-сети – однонаправленная (без обратных связей), принципиально многослойная.

Сеть глубоких убеждений (deep belief network – DBN) — это класс глубоких нейронных сетей, представляющий собой генеративную графическую модель, состоящую из нескольких слоев скрытых переменных (hidden units), со связями между слоями, но не между переменными внутри каждого слоя.

Техническое (компьютерное) зрение: технология создания систем, которые выполняют обнаружение, отслеживание и классификацию объектов по изображениям. Примеры применения технического зрения: управление процессами, видеонаблюдение, организация информации (например, индексирование базы данных изображений), анализ медицинских изображений, анализ топографических изображений, системы взаимодействия с «гальванической развязкой» (например, интерфейс «аналоговое устройство – ИТ-система»), системы дополненной реальности и др.

Федеративное обучение (federated learning), также известное как совместное обучение (collaborative learning) — это метод машинного обучения на нескольких децентрализованных пограничных устройствах, содержащих локальные образцы данных, без обмена образцами данных между пограничными устройствами. Этот подход отличается от традиционных методов централизованного машинного обучения, когда все локальные наборы данных загружаются на один сервер, а также классических децентрализованных подходов, которые предполагают, что локальные выборки данных распределены одинаково на различных узлах. Федеративное обучение позволяет нескольким субъектам создавать общую надежную модель машинного обучения без совместного использования данных, что позволяет решать такие критически важные проблемы, как конфиденциальность и безопасность данных, права доступа к данным. Применение этого подхода распространяется на целый ряд отраслей промышленности, включая оборону, телекоммуникации, интернет вещей и фармацевтику.

Фундаментальная модель ИИ (AI Fundamental Model): представляет собой сложную архитектуру, способную обучаться на больших объемах данных, запросах пользователей и выполнять широкий спектр задач. Они отличаются от традиционных моделей машинного обучения своей способностью к обобщению и адаптации к новым задачам без необходимости полного переобучения. Эти модели часто используют глубокие нейронные сети и обладают огромным числом параметров, что позволяет им выявлять сложные паттерны и взаимосвязи в данных. Примерами фундаментальных моделей являются DALL-E, GPT и PaLM, которые демонстрируют способность к генерации изображений, текстов и решению других сложных задач.

Чат-боты: чат-робот, который может общаться с человеком-пользователем с помощью текстовых или голосовых команд. Используется в электронной коммерции, образовании, здравоохранении и бизнесе для удобства общения и ответов на вопросы пользователей.

Эвристика: предположение разработчика алгоритма машинного обучения, реализованное в параметрах или архитектуре алгоритма.

Исследователи

Список из 10 исследователей, внесших значительный вклад в развитие генеративного ИИ:

- Ян Гудфеллоу - американский специалист по информатике, инженер и руководитель, входит в число ведущих исследователей, внесших значительный вклад в развитие генеративного ИИ. Он наиболее известен своим изобретением генеративных состязательных сетей (GAN), которые используют глубокое обучение для генерации изображений.

- Ян Лекун - французский ученый в области информатики, известный своими работами по применению нейросетей к задачам оптического распознавания символов и машинного зрения. Его исследования сыграли важную роль в развитии технологий, лежащих в основе современных генеративных моделей искусственного интеллекта.
- Илья Суцкевер - канадский и американский ученый в области информатики, искусственного интеллекта и машинного обучения, один из ведущих специалистов в области генеративного ИИ. Он сыграл ключевую роль в создании и развитии OpenAI, организации, стоящей за разработкой и продвижением больших языковых моделей ChatGPT.
- Джеффри Хинтон - британский ученый-информатик, известный своими работами в области глубинного обучения. Он разработал алгоритм обратного распространения ошибки, который стал ключевым инструментом в обучении глубоких нейронных сетей.
- Юрген Шмидхубер - немецкий ученый в области искусственного интеллекта и машинного обучения, известный своими работами в области глубинного обучения и нейронных сетей. Он является разработчиком архитектуры нейронных сетей по распознаванию речи, заимствованной Google.
- Эндрю Ын - американский ученый в области искусственного интеллекта и машинного обучения, известный своими работами в области глубинного обучения и нейронных сетей. Он является разработчиком роботизированной операционной системы и основателем образовательной платформы Coursera.
- Алексей Эфрос - американский специалист в области информатики, работающий в области компьютерного зрения. В контексте генеративного ИИ Эфрос известен своими исследованиями в области улучшения качества изображений с использованием алгоритмов глубокого обучения. Он также работал над созданием инструментов, способных идентифицировать города по их архитектуре и восстанавливать недостающие детали на фотографиях. Эти исследования вносят значительный вклад в развитие генеративного ИИ, особенно в области компьютерного зрения и обработки изображений.
- Йошуа Бенджио - канадский ученый в области информатики и искусственного интеллекта, известный своими работами в области глубинного обучения и нейронных сетей. Он является одним из разработчиков алгоритма обратного распространения ошибки и соавтором книги "Deep Learning".
- Дэвид Серен - американский ученый в области искусственного интеллекта и машинного обучения, известный своими работами по генеративному моделированию и глубокому обучению. Он является соавтором ряда ключевых работ по генеративным моделям и сооснователем компании OpenAI.
- Фей-Фей Ли известна созданием ImageNet, набора данных, который позволил быстро продвинуться в области компьютерного зрения в 2010-х годах. Она является профессором компьютерных наук Sequoia Capital в Стэнфордском университете и

бывшим участником совета директоров Twitter. Ли также является содиректором Стэнфордского института искусственного интеллекта, ориентированного на человека, и содиректором Стэнфордской лаборатории видения и обучения.

Литература

Наиболее популярные обзорные публикации по тематике XAI за период 2021-2023:

Авторы	Источник	Год	Вклад
Али и др.	[8]	2023	Обзор текущих исследований и трендов в области XAI, а также таксономия, включающая четыре оси XAI: объяснимость данных, объяснимость модели, постфактумная объяснимость и оценка объяснений. Авторы подчёркивают связь XAI с принципами надёжности, точками зрения пользователей, приложениями ИИ и государственными перспективами.
Бодрия и др.	[31]	2023	Обзор методов XAI (некоторые из них прошли бенчмаркинг), классифицированных на основе типа объяснения, которое они производят.
Швальбе и Финзель	[5]	2023	Метаобзор обзоров методов и концепций XAI, а также всеобъемлющая таксономия всей области.
Вебер и др.	[30]	2023	Обзор методов XAI, используемых для улучшения моделей машинного обучения, обсуждение их преимуществ и недостатков.
Гуидотти и др.	[32]	2022	Литературный обзор контрфактических объяснений и способов их нахождения.
Мачлев и др.	[33]	2022	Обзор текущих вызовов, приложений и будущих возможностей XAI для энергетических и силовых систем.
Мей и др.	[34]	2022	Обзор того, как генетическое программирование может быть использовано для XAI.

Минь и др.	[35]	2022	Обзор концепций, обзоров и методов XAI, выделение возможностей и вызовов. Предложена таксономия методов XAI с тремя категориями: моделирование до и после, объяснимость и интерпретируемые модели.
Спейт	[4]	2022	Метаобзор таксономий методов XAI. Автор предлагает новую таксономию, включающую рассмотренные, и предлагает создать базу данных методов XAI с их свойствами и дерево решений для помощи в выборе подходящих методов.
Тайсслер и др.	[36]	2022	Литературный обзор объяснимого ИИ для классификации временных рядов.
Янг и др.	[37]	2022	Мини-обзор методов XAI с акцентом на применение в медицине.
Зини и Авад	[38]	2022	Обзор методов XAI для обработки естественного языка и подходов к их оценке.
Антониади и др.	[39]	2021	Обзор текущих вызовов, приложений и будущих возможностей XAI для систем поддержки клинических решений.
Шазетт и др.	[29]	2021	Систематический обзор определений объяснимости и влияния их принятия на различные свойства системы. Результаты включают определение объяснимости, а также модель и каталог знаний о её влиянии.
Хеуиллет и др.	[40]	2021	Обзор текущих вызовов, методов и будущих возможностей XAI в обучении с подкреплением.
Лангер и др.	[6]	2021	Обзор целей, которые должен выполнять XAI. Авторы предлагают концептуальную модель для руководства междисциплинарными исследованиями XAI и подчёркивают важность учёта потребностей различных заинтересованных сторон, вовлечённых в системы ИИ.
Маркус и др.	[41]	2021	Обзор роли XAI в создании надёжного ИИ для здравоохранения, фокусирующийся на терминологии и стратегиях оценки.

Мохенси и др.	[42]	2021	Многодисциплинарный обзор XAI, фокусирующийся на дизайне и оценке объяснимых систем ИИ.
Рожат и др.	[43]	2021	Обзор методов XAI для данных временных рядов и иллюстрация типа объяснений и воздействия, которое они производят.
Самек и др.	[44]	2021	Обзор постфактумных методов