Юрий Почанин

Цифровая трансформация промышленных предприятий

Юрий Почанин

Цифровая трансформация промышленных предприятий

«Автор»
2026

Почанин Ю. С.

Цифровая трансформация промышленных предприятий /
Ю. С. Почанин — «Автор», 2026

В научно- популярной книге описаны основные понятия технологий и инструментов для цифровой трансформации промышленного предприятия. Отображены вопросы об искусственном интеллекте, нейронных сетях, машинном и глубоком обучении. Описан зарубежный опыт по созданию цифровых бизнес – экосистем в промышленности. Дана информация по следующим вопросам: управление данными, бизнес модели, ERP – планирование ресурсов предприятия, описаны сети для передачи данных между физическими объектами, оснащенными встроенными средствами взаимодействия с использованием технологий для связи друг с другом и внешней средой в том числе интернет вещей – 1оТ, цифровые двойники, большие данные, компьютерное зрение, роботизация, автоматизация рабочих мест, облачные технологии.

Юрий Почанин
Цифровая трансформация промышленных предприятий

Цифра ВВЕДЕНИЕ

Цифровая трансформация позволяет решать на новом уровне непрерывно усложняющиеся задачи, стоящие перед промышленными предприятиями. В основе инновационных изменений лежит переход к цифровым и умным производственным средам. Процессы, ранее зависевшие от бумажной документации и ручного труда, обретают иную форму благодаря безбумажному проектированию и автоматизированному производству. Цифровая трансформация промышленности трансформируется через интеграцию концепций «Индустрия 4.0» и «Фабрики будущего». Использование виртуального моделирования позволяет создавать точные прототипы, что существенно сокращает время и затраты. Технологии интернета вещей расширяют возможности мониторинга и управления, собирая данные в режиме реального времени и создавая межмашинные взаимодействия. Важнейшую роль играют технологии больших данных и предиктивной аналитики, обеспечивающие возможность прогнозирования и оптимизации процессов. Внедрение искусственного интеллекта и робототехники способствует повышению эффективности и снижению человеческого фактора в операции. Облачные вычисления обеспечивают гибкость и доступность данных, что важно для распределённых и виртуальных фабрик. Аддитивное производство, получающее всё большее распространение, привносит гибкость в изготовление сложных компонентов и изделий. Оно позволяет производить адаптивные решения, что особенно важно в условиях быстро меняющихся рыночных требований. Умные фабрики, характеризующиеся высокой степенью автоматизации, предоставляют новое измерение эффективности в условиях постоянных изменений. Здесь управление осуществляется в реальном времени, что позволяет быстро адаптироваться к новым вызовам. Это достигается через синергию технологий интернета вещей, больших данных и передовых информационных систем. Процесс цифровой трансформации в промышленности оказывается мощным катализатором для уменьшения эксплуатационных расходов, улучшения эффективности производственного цикла и обеспечения высокого уровня адаптивности бизнеса, что способствует ускорению разработки и внедрения новаторских продуктов, позволяет реализовать массовую кастомизацию и применение гибких производственных систем. Лидирующими экономиками в этой технологической эволюции выступают азиатские гиганты, государства Европейского союза и Северная Америка, что отображает их приверженность передовым технологическим практикам и инновациям. Компания Axenix опубликовала в начале 2024 г. топ цифровых трендов промышленности в России и мире, представленный на рис.1. В рамках шести ключевых технологических тенденций, актуальных как для России, так и для международного сообщества, совпадения наблюдаются лишь в двух аспектах. Первое совпадение касается развития передовых инструментов искусственного интеллекта, включая машинное обучение (ML) и продвинутую аналитику. Второе — технология RPA (Robotic Process Automation), представляющая собой автоматизацию бизнес-процессов с помощью программных роботов. В остальных технологических сегментах пути России и мирового сообщества различаются.

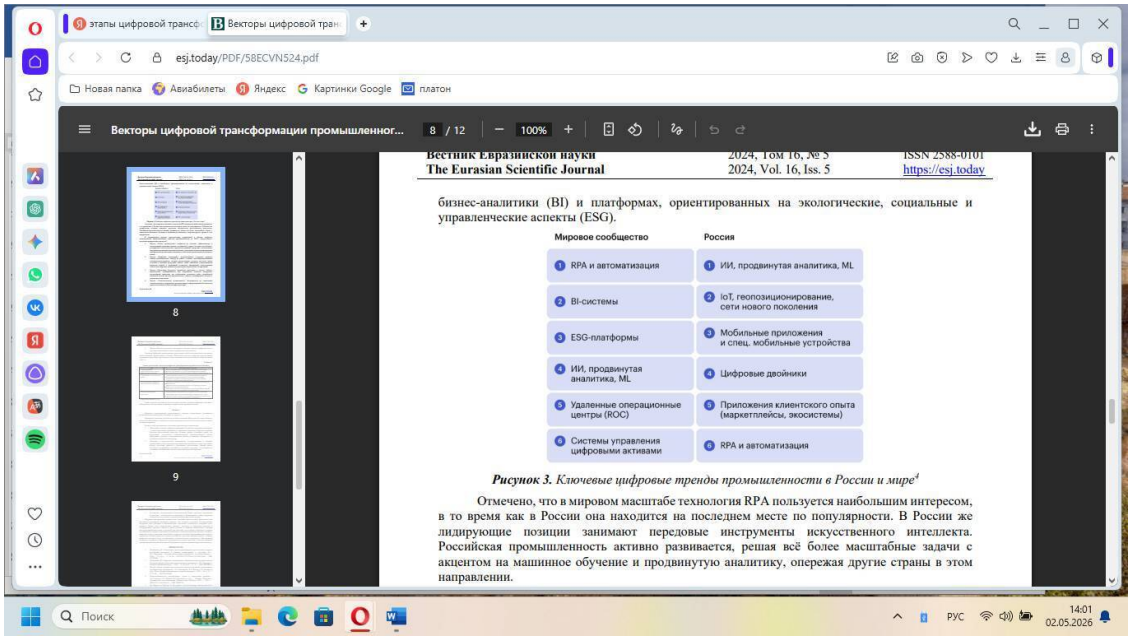


Рис. 1. Ключевые цифровые тренды промышленности в России и мире

В мировом масштабе технология RPA пользуется наибольшим интересом, в то время как в России она находится на последнем месте по популярности. В России же лидирующие позиции занимают передовые инструменты искусственного интеллекта. Российская промышленность активно развивается, решая всё более масштабные задачи с акцентом на машинное обучение и продвинутую аналитику, опережая другие страны в этом направлении. В утвержденном перечне стратегических направлений в области цифровой трансформации обрабатывающих отраслей промышленности до 2030 г. предусмотрено несколько приоритетных проектов:

1. Проект «Умное производство» направлен на усиление эффективности в использовании основных активов, материалов и сырья. Он также способствует расширению технологических горизонтов компаний, внедрению отечественных программных решений в ключевых областях, увеличению количества предприятий, применяющих предиктивную аналитику и технологии промышленного интернета вещей;

2. Проект «Цифровой инжиниринг» предусматривает ускорение процесса выхода промышленных товаров на рынок, создание мультифункциональных площадок-маркетплейсов, которые предоставляют ресурсы для всего цикла создания и продажи продукции. Проект также занимается стандартизацией форматов данных и увеличивает количество предприятий, использующих технологии цифровых двойников и проведение виртуальных тестирований. 3. Проект «Продукция будущего» занимается переходом к системе гибкого производства, адаптированного под потребности клиентов, внедрением предиктивной аналитики для оптимизации ремонтных работ, разработкой сервисной модели продаж промышленных товаров и расширением доступа к передовым технологиям.

4. Проект «Технологическая независимость» фокусируется на укреплении технологического суверенитета, включая защиту информационной безопасности критически важных объектов инфраструктуры.

5. Проект «Интеллектуальная господдержка» включает переход к цифровой модели государственной финансовой поддержки промышленности.

Стратегия цифровой трансформации представляет собой последовательное внедрение бизнес-решений, организованное поэтапно. Реализация стратегии цифровой трансформации

укрупненно может быть представлена в виде поэтапной последовательности бизнес-решений, таблица 1.

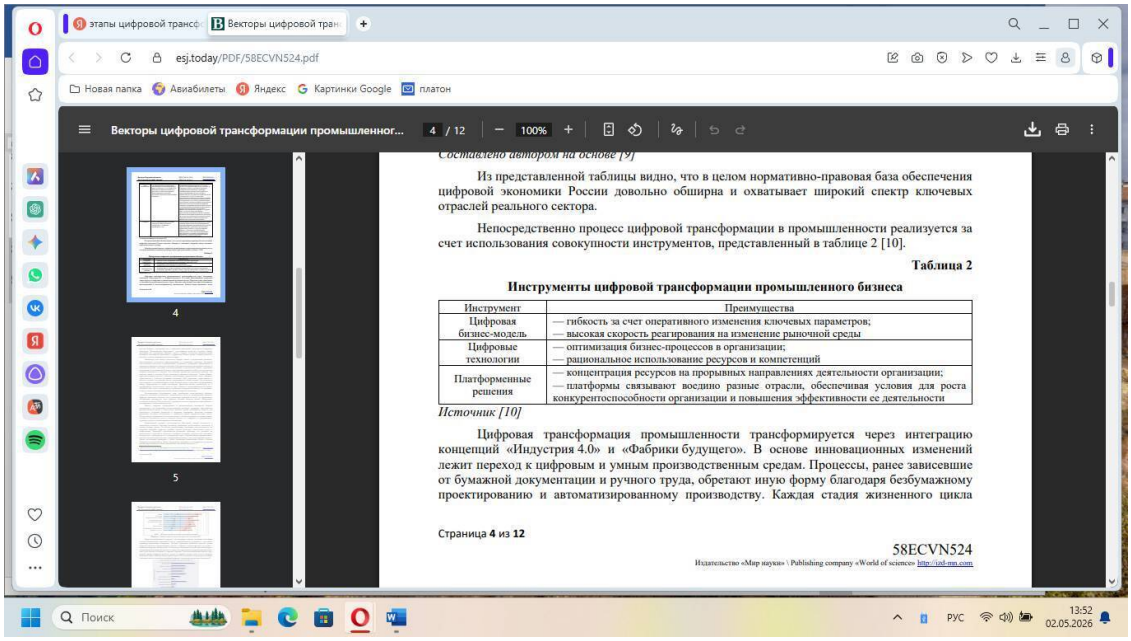
Таблица 1 Этапы реализации стратегии цифровой трансформации промышленного бизнеса

Этап	Ключевые бизнес-решения
1.Формулирование стратегических задач в рамках цифровой трансформации	-многие промышленные предприятия ставят перед собой задачи повышения эффективности, укрепление конкурентных позиций и обеспечении безопасности своего бизнеса в процессе цифровизации.
2.Проведение технологического аудита	-анализ ключевых аспектов для улучшения производительности бизнес-процессов через внедрение технологических решений; -выявление потребностей в финансировании и его источников.
3. Реорганизация и цифровое обновление бизнес-процессов	-интеграция новейших технологических решений и обеспечение их взаимодействия; -переосмысливание организации рабочего пространства, включая разработку систем удаленной работы; -адаптация и обучение сотрудников в соответствии обновленными бизнес-процессами и измененным рабочими ролями.
4. Разработка цифровой экосистемы	- трансформация корпоративной культуры и методов управления; -налаживание процессов взаимодействия с партнерами на основе аналитики больших данных и использования цифровых инструментов; - переход к модели самообслуживающейся организации.

Применение новейших технологий, включая концепции Индустрии 4.0, искусственного интеллекта и блокчейна, способствует повышению производительности и укреплению позиций на мировых рынках. Важно укреплять партнерские отношения с иностранными компаниями и организациями в сфере цифровых технологий для успешного проникновения на зарубежные рынки. Цифровая трансформация должна стать ключевым стратегическим приоритетом для российских предприятий, желающих укрепить свои позиции за рубежом. Государственная поддержка через субсидии и образовательные программы способствует ускорению этого процесса. Кроме того, цифровизация улучшает логистику и управление рисками в международной торговле, сокращает время доставки и повышает прозрачность операций благодаря применению таких технологий, как блокчейн, что особенно актуально в условиях нестабильности мировой экономики и изменений в глобальных цепочках поставок. Активное внедрение новых технологий, инвестиции в цифровые решения, переквалификация кадров и укрепление кибербезопасности позволят российской экономике укрепить процесс создания цифрового контура и будут способствовать технологическому суверенитету. При активной поддержке государства и грамотной экономической политике возможно ускорение цифровой трансформации экономики Российской Федерации и создание надежной цифровой экосистемы. Процесс цифровой

трансформации в промышленности реализуется за счет использования совокупности инструментов, представленный в таблице 2.

Таблица 2 Инструменты цифровой трансформации промышленного бизнеса



Составлено автором на основе [9]

Из представленной таблицы видно, что в целом нормативно-правовая база обеспечения цифровой экономики России довольно обширна и охватывает широкий спектр ключевых отраслей реального сектора.

Непосредственно процесс цифровой трансформации в промышленности реализуется за счет использования совокупности инструментов, представленный в таблице 2 [10].

Таблица 2

Инструменты цифровой трансформации промышленного бизнеса

Инструмент	Преимущества
Цифровая бизнес-модель	— гибкость за счет оперативного изменения ключевых параметров; — высокая скорость реагирования на изменение рыночной среды
Цифровые технологии	— оптимизация бизнес-процессов в организации; — рациональное использование ресурсов и компетенций
Платформенные решения	— концентрация ресурсов на прорывных направлениях деятельности организации; — платформы связывают воедино разные отрасли, обеспечивая условия для роста конкурентоспособности организации и повышения эффективности ее деятельности

Источник [10]

Цифровая трансформация промышленности трансформируется через интеграцию концепций «Индустрия 4.0» и «Фабрики будущего». В основе инновационных изменений лежит переход к цифровым и умным производственным средам. Процессы, ранее зависевшие от бумажной документации и ручного труда, обретают иную форму благодаря безбумажному проектированию и автоматизированному производству. Каждая стадия жизненного цикла

Страница **4** из **12**

58ECVN524

Издательство «Мир наук» \ Publishing company «World of sciences» http://izd-mn.com

Применение новейших технологий, включая концепции Индустрии 4.0, искусственного интеллекта и блокчейна, способствует повышению производительности и укреплению позиций на мировых рынках. Важно укреплять партнерские отношения с иностранными компаниями и организациями в сфере цифровых технологий для успешного проникновения на зарубежные рынки. Цифровая трансформация должна стать ключевым стратегическим приоритетом для российских предприятий, желающих укрепить свои позиции за рубежом. Государственная поддержка через субсидии и образовательные программы способствует ускорению этого процесса. Кроме того, цифровизация улучшает логистику и управление рисками в международной торговле, сокращает время доставки и повышает прозрачность операций благодаря применению таких технологий, как блокчейн, что особенно актуально в условиях нестабильности мировой экономики и изменений в глобальных цепочках поставок. Активное внедрение новых технологий, инвестиции в цифровые решения, переквалификация кадров и укрепление кибербезопасности позволят российской экономике укрепить процесс создания цифрового контура и будут способствовать технологическому суверенитету. При активной поддержке государства и грамотной экономической политике возможно ускорение цифровой трансформации экономики Российской Федерации и создание надежной цифровой экосистемы.

Успешная цифровая трансформация зависит от умного производства или стратегического внедрения ключевых технологий. К ним относятся:

1.Промышленный Интернет вещей: IoT, компонент «Индустрии 4.0», объединяет оборудование, инструменты и системы для сбора и обмена данными на всех производственных участках. Это обеспечивает мониторинг в реальном времени, диагностическое обслуживание и оптимизацию активов.

2. ИИ и машинное обучение. ИИ и машинное обучение анализируют огромные объемы данных для выявления закономерностей, прогнозирования спроса, автоматизации проверок и принятия более взвешенных решений.

3.Облачные вычисления. Облачные платформы предлагают масштабируемую инфраструктуру, позволяющую производителям получать доступ к данным и приложениям из любого места, сотрудничать в реальном времени и сокращать накладные расходы на ИТ.

4. Расширенная аналитика. Аналитика на основе данных позволяет выполнять диагностическое обслуживание, контроль качества и оптимизацию цепочки поставок, сокращая простои и отходы.

5. Робототехника и автоматизация: роботизированные системы и интеллектуальная автоматизация повышают точность, сокращают число ошибок и повышают производительность, особенно в повторяющихся и опасных задачах.

6. Цифровые двойники: цифровые двойники, продукт отрасли 5.0, реплицируют физические активы или процессы в виртуальной среде. Это позволяет выполнять моделирование, тестирование и оптимизацию перед физическим внедрением.

7. Аддитивное производство (3D-печать): 3D-печать обеспечивает быстрое прототипирование и гибкое производство, поддерживает массовую настройку и сокращение отходов материала.

В цифровой трансформации сборочных производств и логистических процессов первостепенными являются задачи объединения машин в сеть, аналитики данных, оптимизации производительности и контроля потока создания ценности. Мониторинг и анализ производственных данных производится следующим способом. На полевом уровне присутствуют оцифрованные данные со стороны машин, доступные по протоколу OPC UA, а также проприетарные контроллеры, к которым нужно подключиться с помощью определенного коннектора. Кроме того, есть дополнительные данные, которые требуется собрать с производственной среды (вибрационные показатели, температурный режим и др.), чтобы предсказать выход из строя того или иного оборудования. В этом процессе в качестве обеспечительной инфраструктуры эффективно использование функционала IoT-шлюзов — программно-аппаратных комплексов, которые позволяют подключиться:

-к линии, не модифицируя ее программу электроавтоматики и используя прямое подключение посредством протоколов OPC UA и проприетарных протоколов;

-к датчикам, которые на линии в данный момент не присутствуют, но будут внедрены в процессе цифровизации.

Затем данные могут агрегировать и отправлять к верхнему уровню ИТ-систем для последующей аналитики, анализа и интеграции с ERP- и MES-системами предприятия. Другими словами, информация собирается из существующей производственной среды и становится доступной на уровне ИТ-решений предприятия, которыми могут выступать:

-реляционные базы данных;

-MES-, ERP-системы;

-облачные решения с достаточно высоким функционалом распределенных вычислений (Microsoft Azzure, MySQL и др.).

Цифровая трансформация позволяет решать на новом уровне непрерывно усложняющиеся задачи, стоящие перед промышленными предприятиями. В настоящее время в целом решена задача автоматизации управления технологическими процессами с использованием цифровых АСУТП. Они позволяют вести управление процессами в замкнутом контуре по предопределенным алгоритмам, реализовывать оптимальные стратегии управления с применением систем усовершенствованного управления и автоматически выполнять последовательности операций (например, пуск и останов оборудования, или исполнение рецептур многостадийных периодических процессах).

В отличие от автоматизации технологического процесса, задачи управления производством в массе своей не автоматизированы. В перечень задач управления производством входят, например, подготовка и контроль выполнения производственных планов, задачи оптими-

зации и контроля производственных режимов, задачи контроля состояния и эффективности промышленных активов и основного оборудования, вопросы безопасности и надежности оборудования, вопросы безопасности персонала, контроля выбросов и множество других.

В настоящее время большая часть данных и документов предприятия хранится в виде файлов и цифровых данных. Перевод данных, ранее предоставлявшихся на бумажных носителях, в цифровую форму без изменения вида и содержания данных и документов известен как оцифровка. На этом этапе не происходит никаких изменений в существующих бизнес-процессах.

По мере внедрения технологий предприятия получают все большее количество данных в цифровой форме. В конечном счете становится возможным создание полного описания предприятия в цифровом виде — цифровой копии (цифрового двойника). Процесс внедрения цифровых технологий для создания такой цифровой копии называется цифровизацией.

Цифровая копия создается для всех стадий жизненного цикла производства, от проектирования до эксплуатации и ремонта. Существующие бизнес-процессы автоматизируются и ускоряются с применением данных цифровой копии производства вместо документов, а также программного обеспечения. По мере того, как цифровая копия становится все более точной и полной, разрабатываются и внедряются новые эффективные бизнес-процессы, которые заменяют старые.

ГЛАВА 1. ПОНЯТИЯ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

Искусственный интеллект (ИИ) — это способность вычислительных систем выполнять задачи, обычно связанные с человеческим интеллектом, такие как обучение, рассуждение, решение проблем, восприятие и принятие решений. ИИ использует алгоритмы, которые позволяют компьютеру обрабатывать большие объёмы данных и находить в них закономерности. На основе этих закономерностей он может делать выводы, предсказывать события или принимать решения.

Искусственным интеллектом называют комплекс программ, разработанных с целью воспроизведения навыков, присущих человеку. Это способность заниматься решением проблем, планированием, пополнять запас своих знаний, улучшать подход к выполнению поставленных задач в ходе работы над ними.

Искусственный интеллект — это искусственная система, имитирующая интеллектуальную деятельность человека. Основополагающей работой, заложившей фундамент для создания искусственных моделей нейронов и нейронных сетей, явилась работа Уоррена С. Мак-Каллока и Вальтера Питтса "Логическое исчисление идей, относящихся к нервной активности", опубликованная в 1943 г., в которой была предложена модель формального нейрона.

Имитация интеллектуальной деятельности человека может быть осуществлена разными способами. В связи с этим назовем три основных направления исследований в искусственном интеллекте: эвристическое (или информационное), бионическое и эволюционное.

Эвристическое, или информационное, направление исследований в искусственном интеллекте включает специалистов, занимающихся созданием машинных способов решения интеллектуальных задач, а также созданием программ для вычислительных машин, решающих такие задачи. При этом как будут устроены подобные программы, насколько близки или далеки будут те способы, которыми они достигают поставленной цели по сравнению с человеческими способами, абсолютно не имеет никакого значения. Главное — конечный результат, его совпадение с результатом, получаемым человеком при решении той же задачи.

Бионическое направление исследований в искусственном интеллекте изучает процессы, протекающие в мозгу человека, когда он решает задачи. Программы для вычислительной машины создаются для имитации процессов получения результатов решения у человека и для изучения этих процессов. Ученые, работающие в бионическом направлении, пытаются воссоздать техническими средствами сам объект, в котором бы протекали процессы, схожие с пси-

хическими процессами, проявляющимися у человека во время решения задач. Такие исследователи специально конструируют сети искусственных нейронов и другие аналоги, присущие нервной системе человека.

Эволюционное моделирование- направление в искусственном интеллекте, в основе которого лежат принципы и понятийный аппарат, заимствованные из эволюционной биологии и популяционной генетики и объединяющие компьютерные методы (генетические алгоритмы, генетическое программирование, эволюционное программирование и эволюционные стратегии) моделирования эволюционных процессов в искусственных системах.

Эволюционное программирование– это метод оптимизации, основанный на моделировании процесса биологической эволюции. Здесь основной упор делается на связь между родительскими особями и их потомками, а изменения происходят только путем мутаций, без скрещивания. Каждое решение характеризуется набором параметров и способностью к изменению. Процесс оптимизации происходит путем последовательного создания новых поколений решений, где каждое следующее поколение создается на основе лучших представителей предыдущего.

Развитие систем искусственного интеллекта продолжается стремительными темпами, меняя окружающую действительность на наших глазах. Мы уже увидели умные беспилотные автомобили, чипы с ИИ, мощные онлайн-инфраструктуры Azure-Microsoft на базе искусственного интеллекта и множество других блестящих изобретений.

Система искусственного интеллекта принимает входные данные в виде речи, текста, изображения, а затем обрабатывает их, применяя различные правила и алгоритмы. После обработки система выдает результат, т. е. успех или неудачу, при вводе данных. Затем результат оценивается посредством анализа, открытия и обратной связи. Наконец, система использует свои оценки для корректировки входных данных, правил и алгоритмов, а также целевых результатов. Этот цикл продолжается до тех пор, пока не будет достигнут желаемый результат.

Когнитивные вычисления направлены на воссоздание мыслительного процесса человека в компьютерной модели. Технология стремится имитировать логику работы человеческого сознания и улучшить взаимодействие между людьми и машинами. Это происходит, например, через распознавание ИИ человеческого языка и значения изображений для последующего самообучения. Реализовать систему когнитивных вычислений сегодня возможно только в таких сложных машинах, как суперкомпьютер IBM Watson.

В целом искусственный интеллект представляет собой набор моделей и методов, который способен на основе полученной информации сделать те или иные выводы. Общая характеристика для всех моделей – способность извлечь знания из набора данных. Что-то вроде вычисление значения функции с миллионами и миллиардами переменных. Кроме того, ИИ— это наука на стыке математики, биологии, психологии, кибернетики и ещё кучи всего. Она изучает технологии, которые позволяют человеку писать «интеллектуальные» программы и учить компьютеры решать задачи самостоятельно. Главная задача ИИ— понять, как устроен человеческий интеллект, и смоделировать его.

Главными теоретическими проблемами были и остаются следующие проблемы. Центральная проблема искусственного интеллекта — это проблема представления знаний в компьютере. Здесь немаловажным является вопрос: «Что́ такое знание?» Следующий вопрос: «Как представлять знания?» — возникает сразу, если мы собираемся использовать их с применением компьютера.

Решение проблемы представления знаний опирается на исследования в области компьютерной лингвистики и в области компьютерной логики. Компьютерная лингвистика лежит в основе естественно-языкового общения с компьютером и автоматического перевода, а компьютерная логика служит для формализации всего богатства человеческих рассуждений.

Другая главная проблема искусственного интеллекта — это проблема выявления и исследования интеллектуальных мета процедур человека. Сложность этой проблемы обусловлена специфичностью устройства человеческого мозга, его полушарной асимметрией. Дело в том, что наш мозг состоит из двух полушарий, левого и правого. Левое полушарие строго логично, рационально. Мета процедуры, характерные для него, могут быть описаны словесно, они могут быть формализованы в четкие алгоритмы, реализуемые на современных компьютерах с архитектурой Неймана-Тьюринга. Правое полушарие имеет дело (можно сказать «мыслит») чувственными образами. Интуитивные рассуждения, озарения, вещие сны и т. п. являются, по-видимому, результатом работы именно правого полушария

По прогнозам ученых, дальнейшее развитие исследований в искусственном интеллекте приведет: — к смене парадигмы (модели) ЗНАНИЯ + ВЫВОД парадигмой ЗНАНИЕ + ОБОСНОВАНИЕ. Это позволит в интеллектуальных системах использовать:

-методы обоснования и аргументы;

-современные языки программирования, ориентированные на вывод одних знаний из других;

-совершенствовать инструментальные средства искусственного интеллекта, в частности, языков программирования, ориентированных на обоснование;

- модернизацию архитектуры вычислительных машин пятого и последующих поколений, сейчас модернизация идет в 4-х направлениях: гигантские суперкомпьютеры; нейробионическое направление (сотни тысяч процессоров с программируемой конфигурацией); территориально удаленные компьютеры и базы с высокоскоростными каналами связи;

-специальные процессоры для обработки зрительных образов и знаний и для проведения рассуждений автономно, т. е. без помощи человека;

-появление методов распараллеливания решения задач на уровне архитектурных решений о структуре компьютера и на логически-теоретическом уровне;

-появление новых моделей представления знаний, позволяющих проводить обработку интегрированной информации (символической, текстовой, зрительной, акустической, тактильной;

-синтез разнотипных экспертных систем, которые будут использоваться совместно для выработки решений, т. е. как консилиум экспертных систем разного типа

Любая программная система, создаваемая в рамках искусственного интеллекта, всегда ориентирована на использование знаний. Знания, выраженные на естественном языке, черпаются из книг, статей и других источников и в том виде, в котором содержатся в этих источниках, не могут быть использованы для обработки на компьютере. Требуется выбрать подходящий способ их формализации (представления) для получения возможности обработки знаний на вычислительных машинах. Сама обработка знаний на компьютере заключается в получении по определенным правилам вывода других знаний на основе имеющихся.

Первичными базовыми понятиями искусственного интеллекта являются понятия знание, представление знаний и вывод. Знаниями принято называть хранимую (в компьютере) информацию, формализованную в соответствии с определенными структурными правилами, которую компьютер может автономно использовать при решении проблем по таким алгоритмам, как логические выводы. Знания можно разделить на факты (фактические знания), правила (знания для принятия решений) и метазнания (знания о знаниях).

Для того чтобы манипулировать всевозможными знаниями о реальном мире с помощью компьютера, необходимо сначала представить их в виде, пригодном для использования на компьютере. С помощью ИИ автоматизируют работу, повышают эффективность и решают сложные задачи в разных областях.

Искусственный интеллект может помочь человеку в следующем.

1.Автоматизировать рутинные процессы. ИИ экономит время и ресурсы человека. Например, чат-боты в службах поддержки заменяют операторов — они обрабатывают стандартные запросы.

2.Обрабатывать большие данные. ИИ способен анализировать огромные объемы информации и находить закономерности, которые трудно обнаружить человеку. Например, в маркетинге ИИ анализирует поведение пользователей и предлагает персонализированные рекомендации.

3.Улучшать точность и скорость. ИИ используют там, где нужны высокая точность и скорость принятия решений. В медицине системы на базе искусственного интеллекта помогают диагностировать заболевания — они изучают снимки и результаты анализов.

4.Повышать удобство и качество жизни. ИИ внедряют в бытовую технику, транспорт, приложения и устройства. Это делает их более умными и удобными. Например, умные дома с голосовыми ассистентами управляют освещением, температурой и безопасностью.

5.Развивать инновации. ИИ открывает новые возможности в науке, технике и других областях. Например, с его помощью быстрее разрабатывают новые лекарства.

6.Оптимизировать производства. Роботы на производственных линиях повышают производительность и снижают затраты.

Принципы работы искусственного интеллекта включают в себя машинное обучение, нейронные сети и обработку естественного языка, что делает эту технологию все более востребованной и эффективной в различных сферах деятельности.

Одним из основных принципов работы искусственного интеллекта является машинное обучение. Этот процесс позволяет программам и системам самостоятельно учиться на основе опыта и данных, что делает их все более эффективными и точными в выполнении задач.

Другим важным принципом является нейронные сети, которые представляют собой модель обработки информации по принципу работы человеческого мозга. Нейронные сети позволяют искусственному интеллекту анализировать данные, выявлять закономерности и делать предсказания на основе полученных знаний.

Нейросети — это один из подходов к созданию ИИ, который вдохновлён системой нейронов в мозге. Вместо того чтобы писать сложные алгоритмы для решения задач, нейросети обучаются на основе большого количества данных и находят в них закономерности. Чтобы работать с нейросетями, не нужно быть учёным. Например, можно освоить профессию инженера машинного обучения. Он работает с данными и создаёт на их основе алгоритмы машинного обучения, которые помогают решать прикладные задачи.

ИИ можно разделить на несколько подкатегорий, таких как машинное обучение, глубокое обучение, обработка естественного языка (NLP) и компьютерное зрение. Машинное обучение включает в себя создание алгоритмов, которые могут обучаться на данных и делать прогнозы или принимать решения без явного программирования. Глубокое обучение, в свою очередь, является подкатегорией машинного обучения и использует нейронные сети для анализа данных и принятия решений.

Обработка естественного языка (NLP) занимается взаимодействием между компьютерами и человеческим языком, что позволяет системам ИИ понимать, интерпретировать и генерировать человеческий язык. Компьютерное зрение, с другой стороны, фокусируется на анализе и интерпретации визуальной информации из окружающего мира.

Базовые направления в рамках искусственного интеллекта и их соотношения представлены на рис. 1.1. Еще одним ключевым принципом является обработка естественного языка. С помощью алгоритмов искусственного интеллекта системы могут понимать и обрабатывать человеческую речь, а также генерировать тексты и отвечать на вопросы, что является важным в различных областях, включая бизнес.

Рис.1.1. Соотношение базовых направлений в рамках искусственного интеллекта

Система искусственного интеллекта принимает входные данные в виде речи, текста, изображения, а затем обрабатывает их, применяя различные правила и алгоритмы. После обработки система выдает результат, т. е. успех или неудачу, при вводе данных. Затем результат оценивается посредством анализа, открытия и обратной связи. Наконец, система использует свои оценки для корректировки входных данных, правил и алгоритмов, а также целевых результатов. Этот цикл продолжается до тех пор, пока не будет достигнут желаемый результат.

Когнитивные вычисления направлены на воссоздание мыслительного процесса человека в компьютерной модели. Технология стремится имитировать логику работы человеческого сознания и улучшить взаимодействие между людьми и машинами. Это происходит, например, через распознавание ИИ человеческого языка и значения изображений для последующего самообучения. Реализовать систему когнитивных вычислений сегодня возможно только в таких сложных машинах, как суперкомпьютер IBM Watson.

ГЛАВА 2. НЕЙРОННЫЕ СЕТИ

Нейронные сети - это модель, вдохновленная работой человеческого мозга. Они состоят из множества связанных между собой искусственных нейронов, которые обрабатывают информацию и принимают решения на основе полученных данных

Нейронная сеть — это последовательность нейронов, соединенных между собой синапсами. Структура нейронной сети пришла в мир программирования прямиком из биологии. Благодаря такой структуре, машина обретает способность анализировать и даже запоминать различную информацию. Нейронные сети также способны не только анализировать входящую информацию, но и воспроизводить ее из своей памяти. Другими словами, нейросеть это машинная интерпретация мозга человека, в котором находятся миллионы нейронов, передающих информацию в виде электрических импульсов.

Нейронные сети не программируются в привычном смысле этого слова, они обучаются. Возможность обучения— одно из главных преимуществ нейронных сетей перед традиционными алгоритмами. Технически обучение заключается в нахождении коэффициентов связей между нейронами. В процессе обучения нейронная сеть способна выявлять сложные зависимости между входными и выходными данными, а также выполнять обобщение. Это значит, что в случае успешного обучения сеть сможет вернуть верный результат на основании дан-

ных, которые отсутствовали в обучающей выборке, а также неполных и/или «зашумлённых», частично искажённых данных.

Компоненты нейросети:

1. Нейроны. Аналогичные нервным клеткам в мозге, они первыми обрабатывают данные и передают информацию дальше.

2.Весовые коэффициенты. Эти параметры определяют, какое значение придается каждому входному сигналу.

3.Слои. Каждая нейросеть состоит из трех основных типов слоев.

Входной слой- — отвечает за получение исходных данных, таких как изображения, текст или числовые значения. Картинка раскладывается на пиксели, каждый из которых поступает на отдельный нейрон.

Скрытые слои-— выполняют основную работу по обработке информации. В этих слоях происходит множество вычислений, которые трансформируют входные данные и создают более сложные представления. Чем больше скрытых слоев, тем сложнее задачи может решать нейросеть.

Выходной слой— предоставляет конечный результат. Например, это может быть классификация изображения (что изображено на фото) или предсказание (какие тренды ожидаются в будущем).

Эти слои соединены между собой, создавая сложную сеть связей. Каждый нейрон скрытого слоя связан с несколькими нейронами предыдущего и следующего слоев, что позволяет им обрабатывать информацию на различных уровнях абстракции.

Также есть нейрон смещения и контекстный нейрон. Нейрон смещения или bias нейрон используется в большинстве нейросетей. Особенность этого типа нейронов заключается в том, что его вход и выход всегда равняются 1 и они никогда не имеют входных синапсов. Нейроны смещения могут, либо присутствовать в нейронной сети по одному на слое, либо полностью отсутствовать, 50/50 быть не может. Соединения у нейронов смещения такие же, как у обычных нейронов — со всеми нейронами следующего уровня, за исключением того, что синапсов между двумя bias нейронами быть не может. Следовательно, их можно размещать на входном слое и всех скрытых слоях, но никак не на выходном слое, так как им попросту не с чем будет формировать связь.

Нейрон смещения нужен для того, чтобы иметь возможность получать выходной результат, путем сдвига графика функции активации вправо или влево.

Синапс-это связь между двумя нейронами. У синапсов есть 1 параметр — вес. Благодаря ему, входная информация изменяется, когда передается от одного нейрона к другому. Допустим, есть 3 нейрона, которые передают информацию следующему, тогда у нас есть 3 веса, соответствующие каждому из этих нейронов. У того нейрона, у которого вес будет больше, та информация и будет доминирующей в следующем нейроне (пример — смешение цветов). На самом деле, совокупность весов нейронной сети или матрица весов — это своеобразный мозг всей системы. Именно благодаря этим весам, входная информация обрабатывается и превращается в результат. Важно помнить, что во время инициализации нейронной сети, веса расставляются в случайном порядке.

Теперь, чтобы понять, как же работают нейронные сети, давайте взглянем на ее составляющие, рис.2.1. и их параметры. Нейрон — это вычислительная единица, которая получает информацию, производит над ней простые вычисления и передает ее дальше.

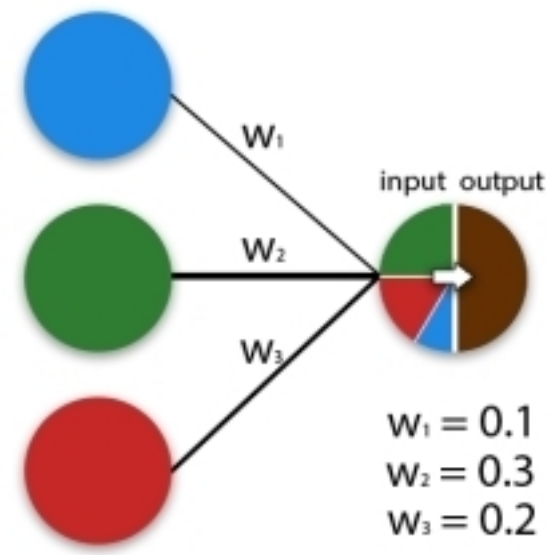


Рис.2.1. Схема передачи информации

Рассмотрим пример, рис. 2.2, где изображена часть нейронной сети, и буквами I обозначены входные нейроны, буквой H — скрытый нейрон, а буквой w — веса.

Из формулы видно, что входная информация — это сумма всех входных данных, умноженных на соответствующие им веса. Тогда дадим на вход 1 и 0. Пусть w1=0.4 и w2 = 0.7 Входные данные нейрона H1 будут следующими: 1*0.4+0*0.7=0.4.

Теперь, когда у нас есть входные данные, мы можем получить выходные данные, подставив входное значение в функцию активации (подробнее о ней далее). Затем, когда у нас есть выходные данные, мы передаем их дальше. И так, мы повторяем для всех слоев, пока не дойдем до выходного нейрона. Запустив такую сеть, в первый раз мы увидим, что ответ далек от правильного, потому что сеть не натренирована.

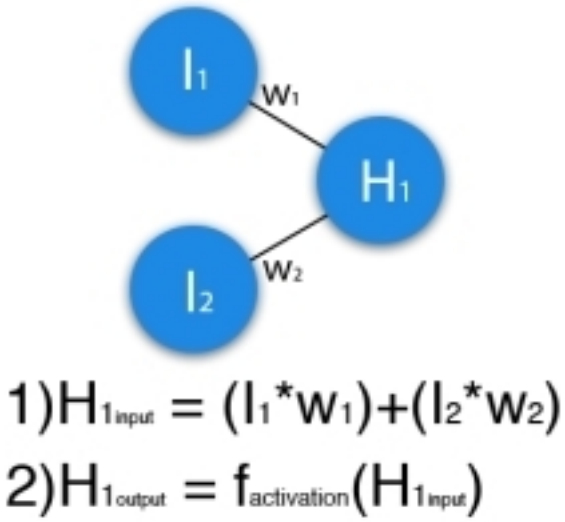


Рис.2.2. Часть нейронной сети

Чтобы улучшить результаты мы будем ее тренировать. Но прежде, чем узнать, как это делать, давайте введем несколько терминов и свойств нейронной сети.

Модель внутренней структуры искусственного нейрона представлена на рис.2.3.

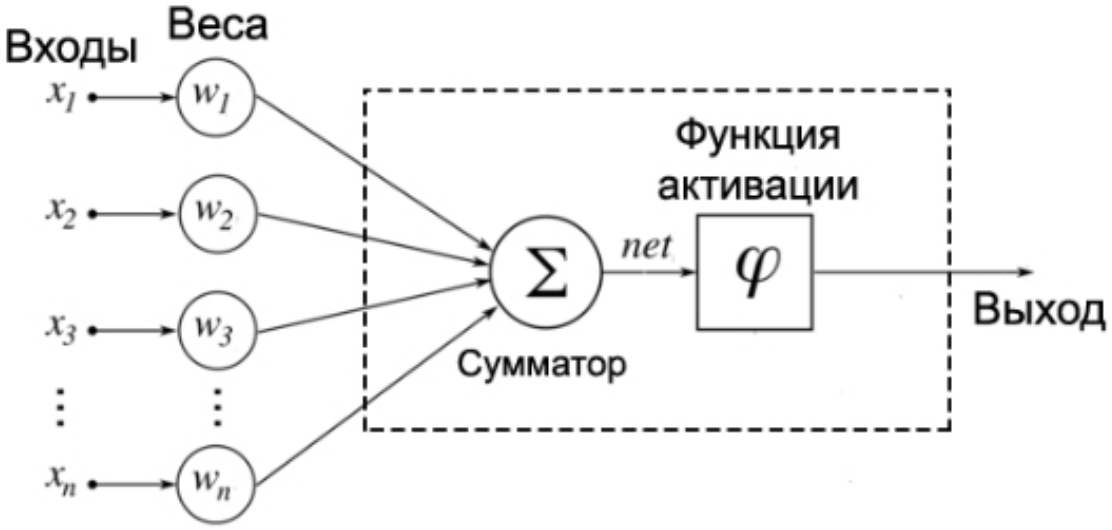


Рис. 2.3. Модель внутренней структуры искусственного нейрона

У каждого нейрона, в том числе и у искусственного, должны быть какие-то входы, через которые он принимает сигнал. Поступившие на входы сигналы умножаются на свои веса. Сигнал первого входа x1 умножается на соответствующий этому входу вес w1. В итоге получаем x1w1. И так до n-ого входа. В итоге на последнем входе получаем xnwn.

Теперь все произведения передаются в сумматор. Он просто суммирует все входные сигналы, умноженные на соответствующие веса:

x1w1+x2w2+⋯+xnwn=∑i=1nxiwi

Когда необходимо коротко записать большое выражение, состоящее из суммы повторяющихся/однотипных членов, то используют знак сигмы.

Рассмотрим простейший вариант записи:

∑i=15i=1+2+3+4+5

Таким образом снизу сигмы мы присваиваем переменной-счетчику I стартовое значение, которое будет увеличиваться, пока не дойдет до верхней границы (в примере выше это 5).

Верхняя граница может быть и переменной. Приведу пример такого случая. Пусть у нас есть n магазинов. У каждого магазина есть свой номер: от 1 до n. Каждый магазин приносит прибыль. Возьмем какой-то (неважно, какой) I -ый магазин. Прибыль от него равна pi. Если мы хотим посчитать общую прибыль от всех магазинов (обозначим ее за P), то нам пришлось бы писать длинную сумму:

P=p1+p2+⋯+pi+⋯+pn

Как видно, все члены этой суммы однотипны. Тогда их можно коротко записать следующим образом:

P=∑i=1npi

Словами: «Просуммируй прибыли всех магазинов, начиная с первого и заканчивая n-ым». В виде формулы это гораздо проще, удобнее и красивее.

Результатом работы сумматора является число, называемое взвешенной суммой. Взвешенная сумма (net) — сумма входных сигналов, умноженных на соответствующие им веса.

net=∑i=1nxiwi

Роль сумматора очевидна– он агрегирует все входные сигналы (которых может быть много) в какое-то одно число – взвешенную сумму, которая характеризует поступивший на нейрон сигнал в целом.

Для понимания роли последнего компонента искусственного нейрона –функции активации –рассмотрим следующий пример. У одного искусственного нейрона задача– решить, ехать ли отдыхать на море. Для этого на его входы мы подаем различные данные. Пусть у нашего нейрона будет 4 входа:

-стоимость поездки;

-какая на море погода;

-текущая обстановка с работой;

-будет ли на пляже закусочная

Все эти параметры будем характеризовать 0 или 1. Соответственно, если погода на море хорошая, то на этот вход подаем 1. И так со всеми остальными параметрами.

Если у нейрона есть четыре входа, то должно быть и четыре весовых коэффициента. В нашем примере весовые коэффициенты можно представить как показатели важности каждого входа, влияющие на общее решение нейрона. Веса входов распределим следующим образом:5,4,1,1.

Нетрудно заметить, что очень большую роль играют факторы стоимости и погоды на море (первые два входа). Они же и будут играть решающую роль при принятии нейроном решения. Пусть на входы нашего нейрона мы подаем следующие сигналы:1,0,0,1. Умножаем веса входов на сигналы соответствующих входов:5,0,0,1. Взвешенная сумма для такого набора входных сигналов равна 6:

net=∑i=14xiwi=5+0+0+1=6

Как нейрон должен решить, ехать на море или нет? Очевидно, нам нужно как-то преобразовать нашу взвешенную сумму и получить ответ. Это позволит выполнить функция активации. Функция активации является фундаментальной составляющей нейронных сетей, и понимание принципов работы, а также областей применения, способствует созданию более эффективных сетей.

Функция активации преобразует входные данные в выходные для их последующей передачи на следующий слой. Эти функции определяют, должен ли нейрон быть активирован или нет, основываясь на взвешенной сумме его входов плюс смещение. Без функций активации нейронные сети были бы просто линейными моделями, неспособными решать сложные задачи, такие как распознавание образов, обработка естественного языка и другие передовые приложения ИИ. В модель нейрона добавляется нелинейный преобразователь, который реализует нелинейную функцию, которая называется функцией активации. По сути, нейрон представляет собой элементарный вычислитель, характеризующийся входным/выходным состоянием и функцией активации. На основе элементарных вычислителей, каждый из которых получает сигналы и посылает другим вычислителям, составляются ИНС разной топологии.

Функция активации определяет, следует ли активировать нейрон, вычисляя взвешенную сумму входных данных и добавляя смещение. Это помогает модели принимать сложные решения и делать прогнозы, вводя нелинейности в выходные данные каждого нейрона. Эта нелинейность позволяет сети изучать более сложные закономерности, которые невозможно изучить с помощью чисто линейной модели, например:

-моделирование функций, которые не являются линейно разделяемыми;

-увеличение способности сети формировать несколько границ принятия решений на основе комбинации весовых коэффициентов и смещений.

В процессе обучения эти весовые коэффициенты и смещения обновляются на основе ошибки, возникающей на выходе процесс известен как обратное распространение ошибки.

Функции активации обеспечивают обратное распространение ошибки, предоставляя градиенты, необходимые для обновления весовых коэффициентов и смещений.

Без нелинейности даже глубокие сети были бы ограничены решением только простых, линейно разделяемых задач. Функции активации позволяют нейронным сетям моделировать очень сложные распределения данных и решать сложные задачи глубокого обучения. Добавление функций нелинейной активации обеспечивает гибкость и позволяет сети извлекать из данных более сложные и абстрактные шаблоны.

Для разных типов искусственных нейронов используют самые разные функции активации. Далее мы подробно рассмотрим самые известные функции активации.

1.Функция единичного скачка-самый простой вид функции активации. Выход нейрона может быть равен только 0 или 1. Если взвешенная сумма больше определенного порога b, то выход нейрона равен 1. Если ниже, то 0. Как ее можно использовать? Предположим, что мы поедем на море только тогда, когда взвешенная сумма больше или равна 5. Значит наш порог равен 5. В нашем примере взвешенная сумма равнялась 6, а значит выходной сигнал нашего нейрона равен 1. Итак, мы едем на море. Однако, если бы погода на море была бы плохой, а также поездка была бы очень дорогой, но имелась бы закусочная и обстановка с работой нормальная (входы: 0011), то взвешенная сумма равнялась бы 2, а значит выход нейрона равнялся бы 0. Итак, мы никуда не едем.

В общем, нейрон смотрит на взвешенную сумму и если она получается больше его порога, то нейрон выдает выходной сигнал, равный 1.

Графически эту функцию активации можно изобразить следующим образом, рис.2.4.

На горизонтальной оси расположены величины взвешенной суммы. На вертикальной оси — значения выходного сигнала. Как легко видеть, возможны только два значения выходного сигнала: 0 или 1. Если взвешенная сумма равна порогу или больше него, то функция выдает 1. Все предельно просто.

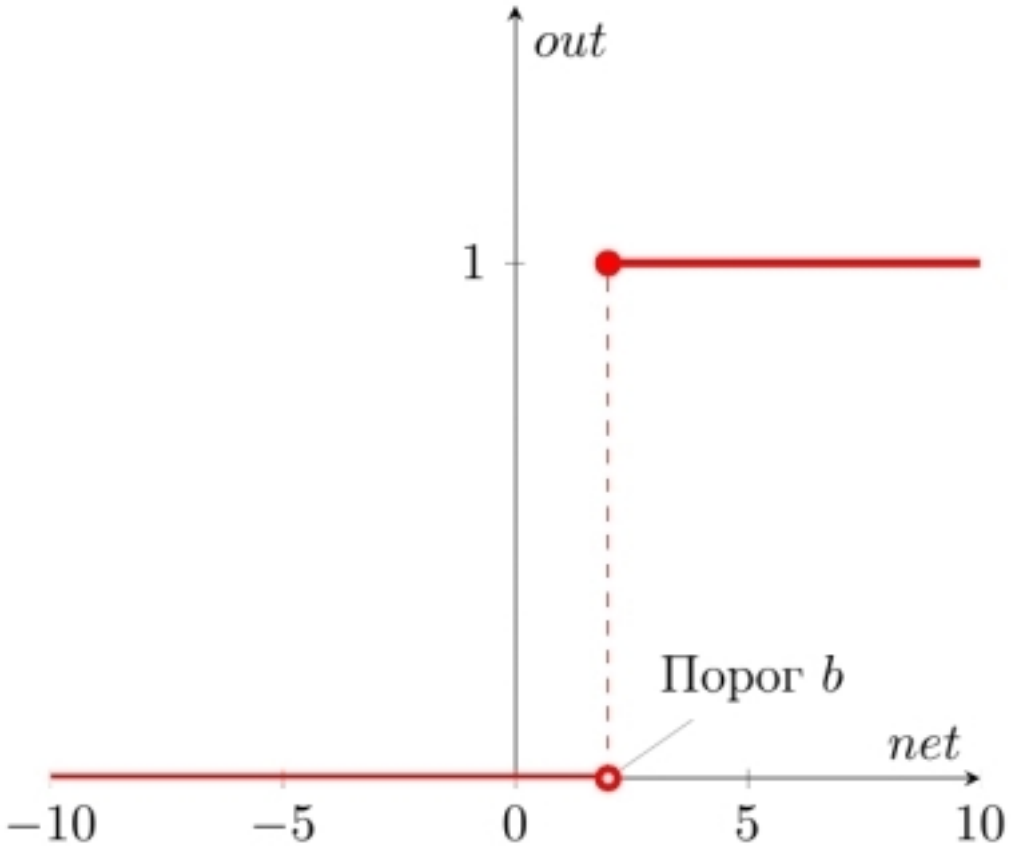


Рис.2.4. Функция единичного скачка

Теперь, запишем эту функцию активации математически. Почти наверняка вы сталкивались с таким понятием, как составная функция. Это когда мы под одной функцией объединяем несколько правил, по которым рассчитывается ее значение. В виде составной функции функция единичного скачка будет выглядеть следующим образом:

out(net)={0,net<b1,net≥b

В этой записи нет ничего сложного. Выход нейрона (out) зависит от взвешенной суммы (net) следующим образом: если net (взвешенная сумма) меньше какого-то порога (b), то out (выход нейрона) равен 0. А если net больше или равен порогу b, то out равен 1.

2. Логистическая функция. На самом деле существует целое семейство сигмоидальных функций, некоторые из которых применяют в качестве функции активации в искусственных нейронах. Все эти функции обладают некоторыми очень полезными свойствами, ради которых их и применяют в нейронных сетях. Эти свойства станут очевидными после того, как вы увидите графики этих функций.

Самая часто используемая в нейронных сетях сигмоида —логистическая функция, рис.2.5.

График этой функции выглядит достаточно просто. Если присмотреться, то можно увидеть некоторое подобие английской буквы S, откуда и пошло название семейства этих функций.

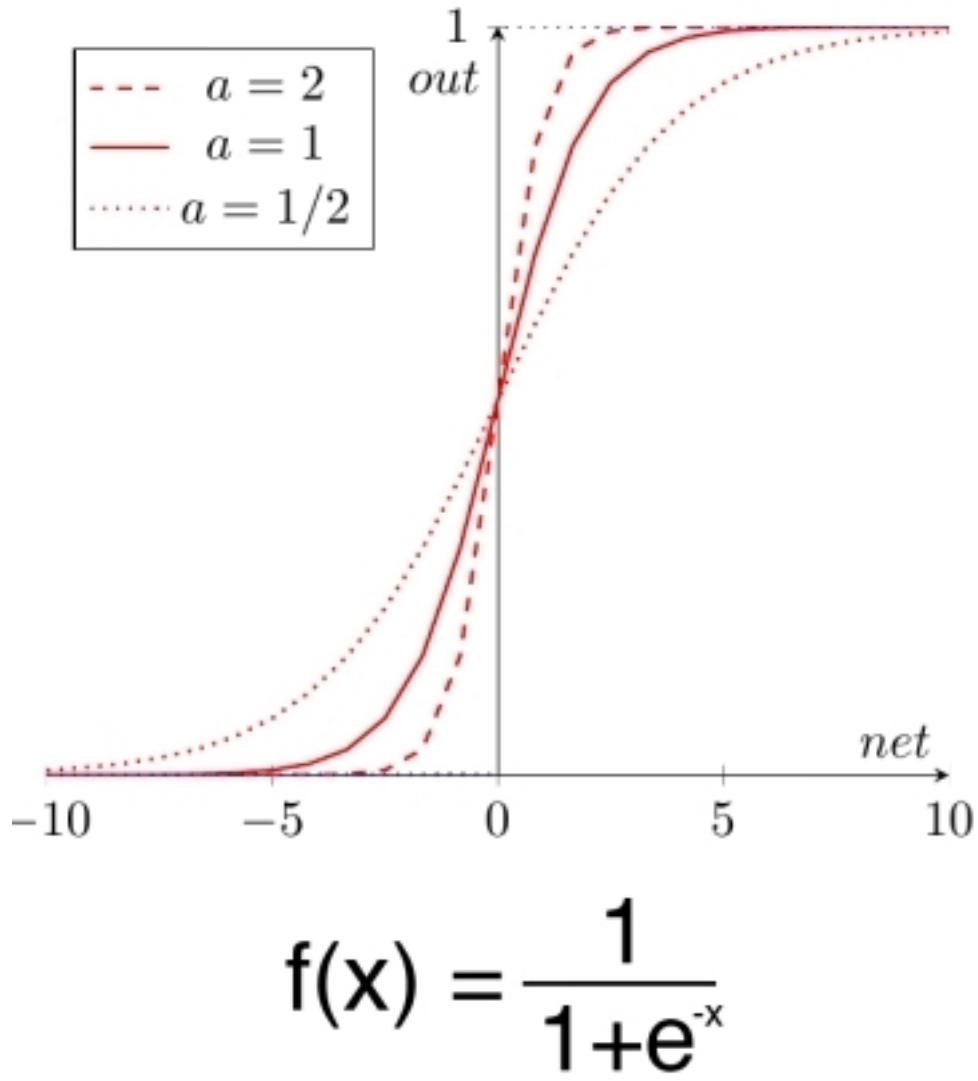
А вот так она записывается аналитически:

out(net)=11+exp(−a·net)

Параметр “a”-это число, которое характеризует степень крутизны функции. На рис. 6.6 представлены логистические функции с разным параметром a.

Вспомним наш искусственный нейрон, определяющий, надо ли ехать на море. В случае с функцией единичного скачка все было очевидно. Мы либо едем на море (1), либо нет (0).

Здесь же случай более приближенный к реальности. Мы до конца полностью не уверены. Тогда использование логистической функции в качестве функции активации приведет к тому, что вы будете получать цифру между 0 и 1. Причем чем больше взвешенная сумма, тем ближе выход будет к 1 (но никогда не будет точно ей равен). И наоборот, чем меньше взвешенная сумма, тем ближе выход нейрона будет к 0. Например, выход нашего нейрона равен 0.8. Это значит, что он считает, что поехать на море все-таки стоит. Если бы его выход был бы равен 0.2, то это означает, что он почти наверняка против.



$$f(x) = \frac{1}{1+e^{-x}}$$

Рис.2.5. Логистическая функция (разные a)

Логистическая функция имеет следующие свойства:

-она является «сжимающей» функцией, то есть вне зависимости от аргумента (взвешенной суммы), выходной сигнал всегда будет в пределах от 0 до 1;

-она более гибкая, чем функция единичного скачка – ее результатом может быть не только 0 и 1, но и любое число между ними;

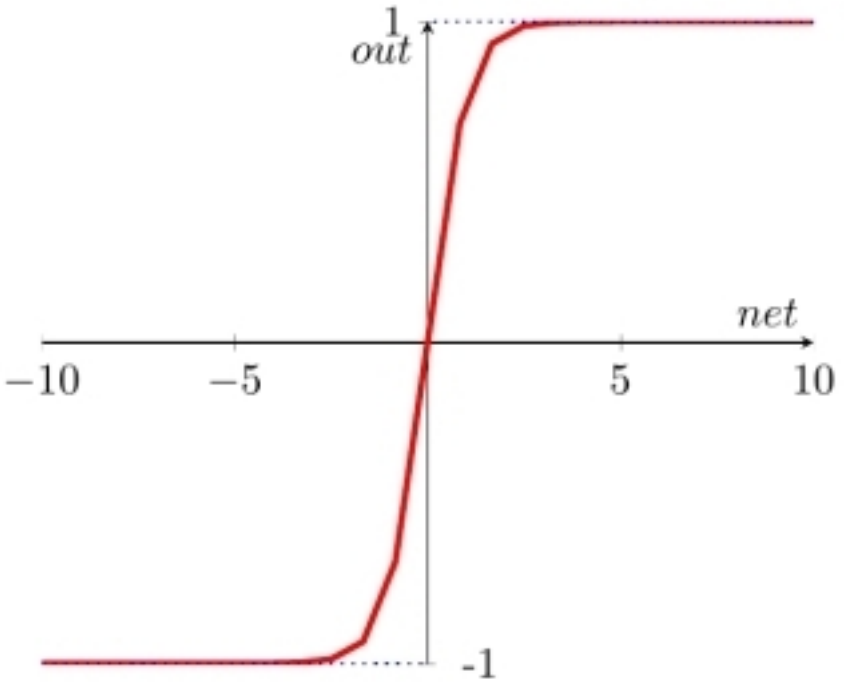
-во всех точках она имеет производную, и эта производная может быть выражена через эту же функцию.

Именно из-за этих свойств логистическая функция чаще всего используются в качестве функции активации в искусственных нейронах.

3. Гиперболический тангенс. Он применяется в качестве функции активации биологами для более реалистичной модели нервной клетки. Такая функция позволяет получить на выходе значения разных знаков (например, от -1 до 1), что может быть полезным для ряда сетей. Функция записывается следующим образом:

out(net)=tanh(neta)

В формуле параметр а также определяет степень крутизны графика этой функции. График этой функции выглядит следующим образом, рис.2.6.



$$f(x) = \frac{e^{2x} - 1}{e^{2x} + 1}$$

Рис.2.6 Гиперболический тангенс

Гиперболический тангенс обладает всеми полезными свойствами, которые имеет и логистическая функция.

4.*Тренировочный сет* — это последовательность данных, которыми оперирует нейронная сеть. В нашем случае исключающего или (xor) у нас всего 4 разных исхода, то есть у нас будет 4 тренировочных сета: 0xor0=0, 0xor1=1, 1xor0=1,1xor1=0.

5.Итерация. Это своеобразный счетчик, который увеличивается каждый раз, когда нейронная сеть проходит один тренировочный сет. Другими словами, это общее количество тренировочных сетов, пройденных нейронной сетью.

6.Эпоха. При инициализации нейронной сети эта величина устанавливается в 0 и имеет потолок, задаваемый вручную. Чем больше эпоха, тем лучше натренирована сеть и соответственно, ее результат. Эпоха увеличивается каждый раз, когда мы проходим весь набор тренировочных сетов, в нашем случае, 4 сетов или 4 итераций.

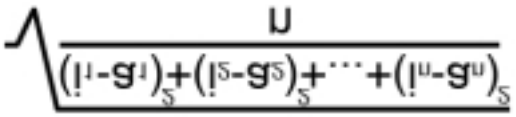
7.Ошибка — это процентная величина, отражающая расхождение между ожидаемым и полученным ответами. Ошибка формируется каждую эпоху и должна идти на спад. Если этого не происходит, значит, вы что-то делаете не так. Ошибку можно вычислить разными путями, но мы рассмотрим лишь три основных способа: Mean Squared Error (далее MSE), Root MSE и Arctan. Здесь нет какого-либо ограничения на использование, как в функции активации, и вы вольны выбрать любой метод, который будет приносить вам наилучший результат. Стоит лишь учитывать, что каждый метод считает ошибки поразному. У Arctan, ошибка, почти всегда, будет больше, так как он работает по принципу: чем больше разница, тем больше ошибка.

У Root MSE будет наименьшая ошибка, поэтому, чаще всего, используют MSE, которая сохраняет баланс в вычислении ошибки.

1.MSE

$$\frac{(i_1-a_1)^2+(i_2-a_2)^2+\ldots+(i_n-a_n)^2}{n}$$

2.*Root MSE*



3.Arctan

$$\frac{\arctan^2(i_1-a_1)+\ldots+\arctan^2(i_n-a_n)}{n}$$

Принцип подсчета ошибки во всех случаях одинаков. За каждый сет, мы считаем ошибку, отняв от идеального ответа, полученный. Далее, либо возводим в квадрат, либо вычисляем квадратный тангенс из этой разности, после чего полученное число делим на количество сетов.

Виды искусственных нейронных сетей. Искусственные нейронные сети состоят из совокупности искусственных нейронов. Возникает логичный вопрос– а как располагать/соединять друг с другом эти самые искусственные нейроны? Как правило, в большинстве нейронных сетей есть так называемый входной слой, который выполняет только одну задачу– распределение входных сигналов остальным нейронам. Нейроны этого слоя не производят никаких вычислений.

Есть разные виды нейронных сетей, каждый из которых используется для определенных целей.

1. Однослойные нейронные сети. В однослойных нейронных сетях сигналы с входного слоя сразу подаются на выходной слой. Он производит необходимые вычисления, результаты

которых сразу подаются на выходы. Выглядит однослойная нейронная сеть следующим образом, рис.2.7.

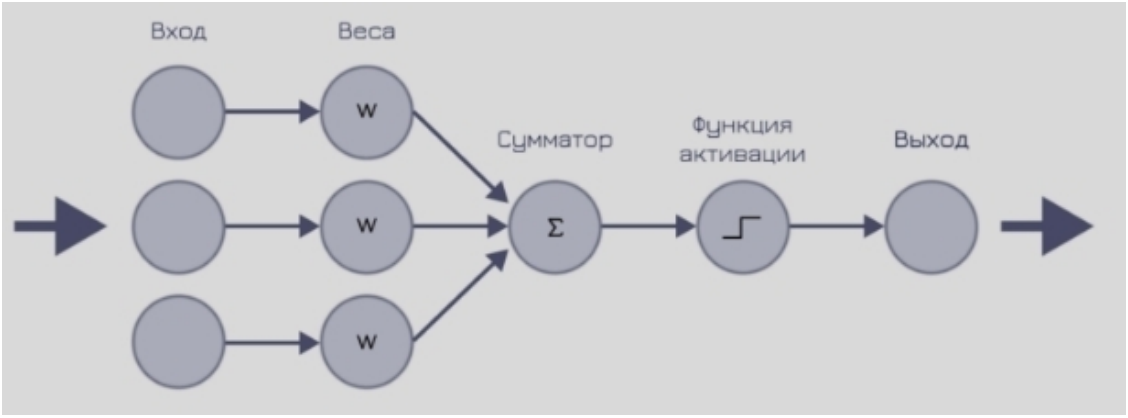


Рис.2.7. Однослойная нейронная сеть

Модель ее состоит из четырех основных компонентов: входа, веса, сумматора и функции активации. Входные данные перемножаются с весами, суммируются и поступают на вход функции активации.

2. Многослойные нейронные сети. Многослойная нейронная сеть — одна из самых базовых архитектур. Она состоит из искусственных нейронов, которые объединяются в слои. Нейрон из одного слоя связан с каждым нейроном из следующего слоя, поэтому такие нейронные сети часто называют полно связными, рис.2.8. Чаще всего их используют для обработки числовых данных или в составе других нейронных сетей.

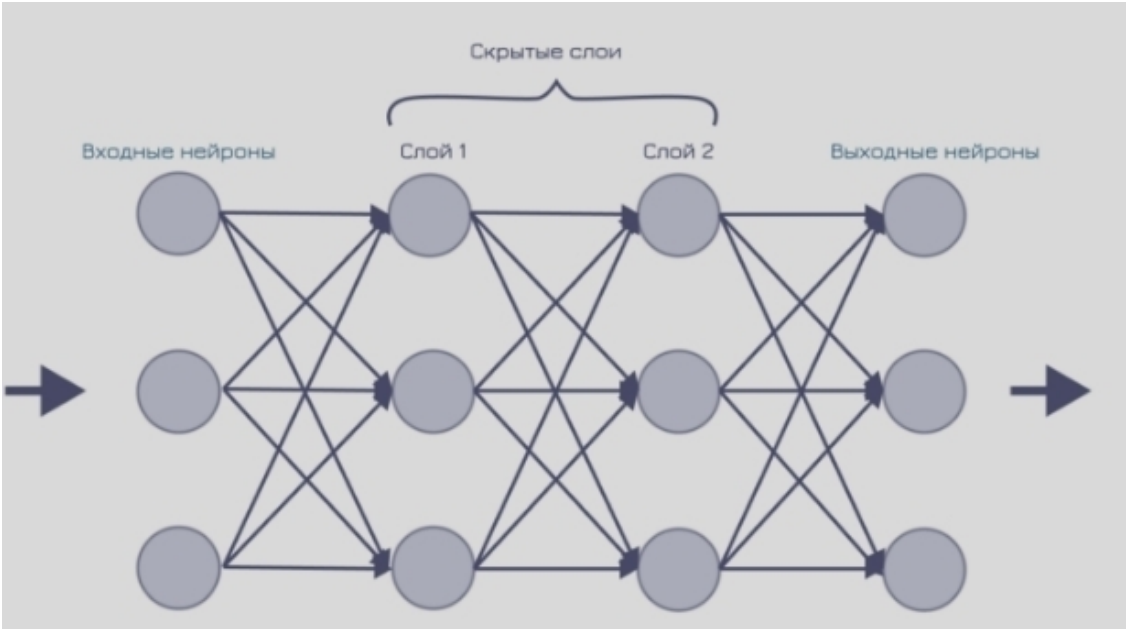


Рис.2.8. Многослойная нейронная сеть

Такие сети, помимо входного и выходного слоев нейронов, характеризуются еще и скрытым слоем (слоями). Понять их расположение просто– эти слои находятся между входным и

выходным слоями. Такая структура нейронных сетей копирует многослойную структуру определенных отделов мозга.

Название скрытый слой получил неслучайно. Дело в том, что только относительно недавно были разработаны методы обучения нейронов скрытого слоя. До этого обходились только однослойными нейросетями. Многослойные нейронные сети обладают гораздо большими возможностями, чем однослойные.

3.Сети прямого распространения— искусственные нейронные сети, в которых сигнал распространяется строго от входного слоя к выходному. В обратном направлении сигнал не распространяется. Такие сети широко используются и вполне успешно решают определенный класс задач: прогнозирование, кластеризация и распознавание.

4.Сети с обратными связями. В сетях такого типа сигнал может идти и в обратную сторону, рис.2.9.

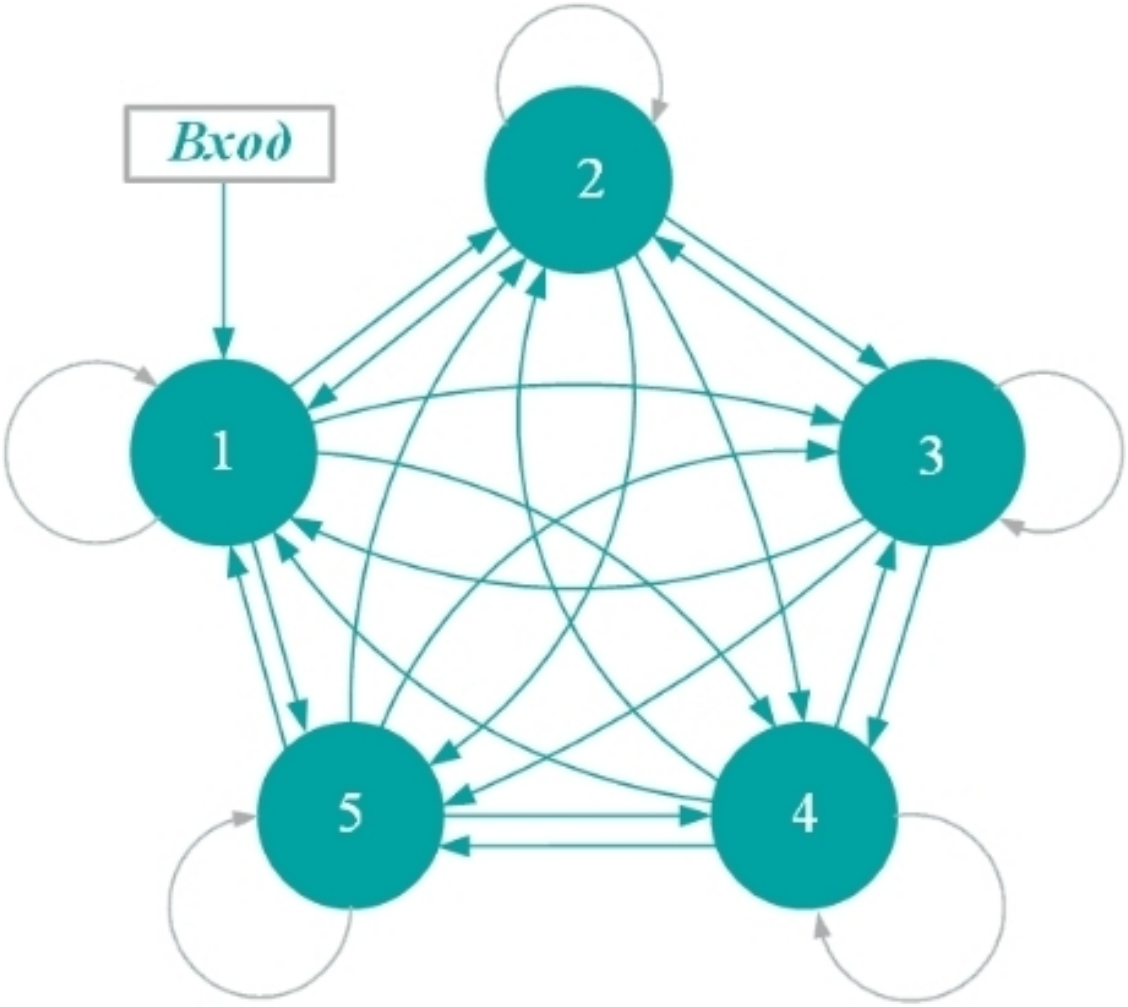


Рис. 2.9. Сеть с обратной связью

Дело в том, что в сетях прямого распространения выход сети определяется входным сигналом и весовыми коэффициентами при искусственных нейронах. А в сетях с обратными связями выходы нейронов могут возвращаться на входы. Это означает, что выход какого-нибудь нейрона определяется не только его весами и входным сигналом, но еще и предыдущими выходами (так как они снова вернулись на входы).

Возможность сигналов циркулировать в сети открывает новые, удивительные возможности нейронных сетей. С помощью таких сетей можно создавать нейросети, восстанавливающие или дополняющие сигналы. Другими словами, такие нейросети имеют свойства кратковременной памяти (как у человека).

Работа нейронных сетей основана на трех этапах: получение данных, их обработка и принятие решения.

1. Получение входных данных. Данные — это основа работы нейронной сети. На этом этапе сеть получает информацию. Это могут быть изображения, текст, звук или числовые данные. Нейросеть преобразует данные в числовой формат, который может быть обработан. Например, для задачи классификации изображений сеть принимает пиксели и преобразует их в числовые значения.

2.Обработка через слои нейронов. Каждый слой нейронной сети выполняет специфическую обработку данных:

-входной слой передает данные в виде чисел;

-скрытые слои извлекают ключевые признаки, например, для изображения это может быть контур или текстура;

-выходной слой преобразует данные в результат: например, вероятность того, что на фото изображена кошка.

Отдельной взятый нейрон отвечает за простую операцию, но в совокупности сеть способна решать сложные задачи. Важным компонентом обработки является функция активации. Она помогает нейронам «решить», какие данные важны, а какие можно игнорировать.

3.Обучение модели. Когда нейронная сеть готова и инициализирована, у нее случайные веса — они еще не настроились под нужный результат. Такая нейросеть называется необученной. Ее надо обучить на определенные действия. Процесс обучения бывает ручным и автоматическим и выглядит обычно так. Нейросети дают на вход разные данные, она анализирует их, а потом ей сообщают, каким должен быть правильный ответ. Сеть устроена так, что будет «стремиться» подогнать веса синапсов, чтобы выдавать верные результаты.

Для эффективного обучения нужно много повторений. Иначе нейронная сеть будет работать неточно — ведь входные данные могут серьезно различаться, а она окажется натренирована только на один возможный вариант. Поэтому обучение проводится в несколько итераций и эпох.

Итерация — это одно прохождение тренировочного сета. Эпоха — это количество полных прохождений всех сетов. Чем больше эпох, тем лучше натренирована нейросеть.

После обучения можно давать нейронной сети входные данные уже без подсказок. Она будет давать ответы на основе весов, которые подсчитала в процессе обучения. Затем нейронная сеть запускается в работу. На вход поступает какая-то информация или запрос. Входной слой нейронной сети обрабатывает ее и переводит в понятный машине вид — в числовые наборы. Затем эти наборы передаются нейронам. Нейроны по формулам, которые в них заложены, обрабатывают информацию. Как именно реагировать на разные детали этих данных, определяют коэффициенты — их нейросеть разработала при обучении. По сути, эти коэффициенты работают как память: нейросеть «вспоминает», как следует реагировать на похожие кластеры информации с известными ей признаками. Данные передаются дальше по нейронной сети, проходит разные слои и типы нейронов. В конечном итоге на последнем слое нейросеть может сделать вывод. На выход подается ее финальная «реакция» на запрос.

ГЛАВА 3. МАШИННОЕ ОБУЧЕНИЕ

Машинное обучение представляет собой подобласть искусственного интеллекта, которая позволяет компьютерным системам обучаться на основе опыта и данных. Для построения таких методов используются средства математической статистики, численных методов, методов оптимизации, теории вероятностей, теории графов, различные техники работы с данными

в цифровой форме. С помощью машинного обучения, компьютерные программы могут самостоятельно улучшать свою производительность и адаптироваться к новым задачам.

3.1. Типы машинного обучения

Типы машинного обучения в широком смысле можно разделить на четыре основные парадигмы в зависимости от типа данных и целей обучения:

-контролируемое обучение,

-неконтролируемое обучение,

-полу контролируемое обучение,

-обучение с подкреплением.

Классификацию типов машинного обучения, с учетом целевой переменной и решаемых задач можно представит в следующем виде, рис.3.1.

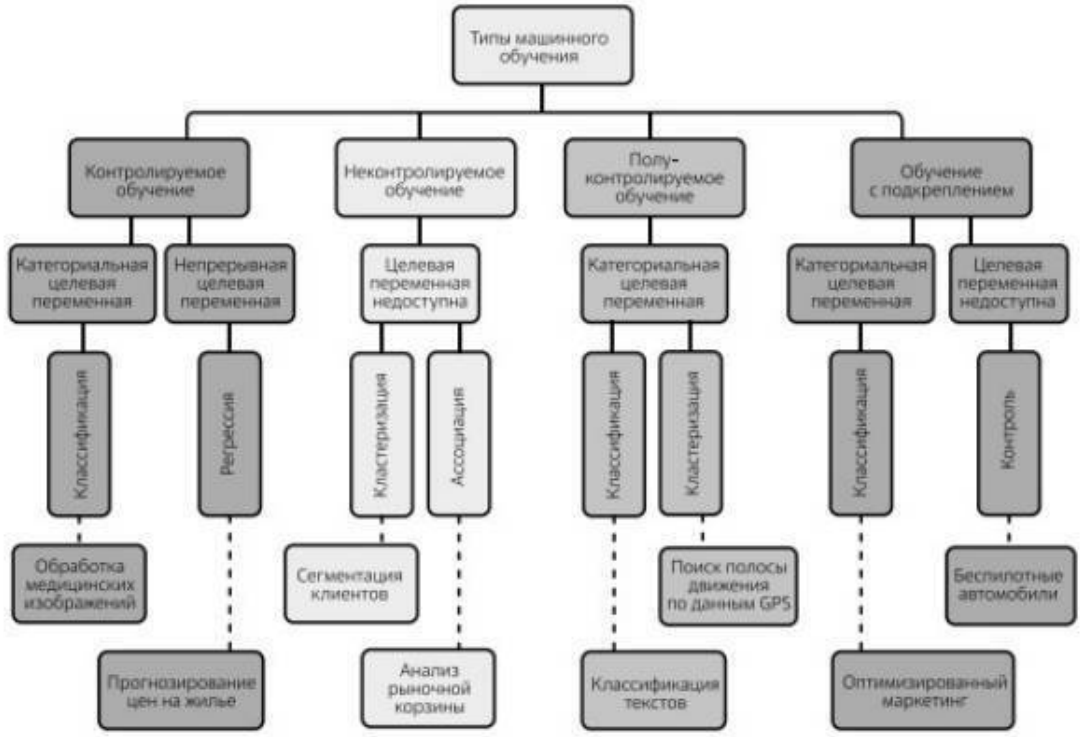


Рис.3.1. Классификация типов машинного обучения с учетом решаемых задач.

3.1.1. Контролируемое обучение (обучение с учителем)

Контролируемое обучение — это фундаментальный подход к машинному обучению и искусственному интеллекту. Он предполагает обучение модели с использованием размеченных данных, где каждому входному значению соответствует правильное выходное значение. Разметка данных — это процесс добавления значимых и информативных меток к набору данных, чтобы облегчить машинным алгоритмам понимание и обработку информации. Без нее алгоритмы машинного обучения терялись бы в море неструктурированных данных.

Алгоритм обучения состоит из входных признаков и соответствующих выходных меток. Процесс работает следующим образом. Моделям предоставляется набор данных для обучения, который включает входные данные (признаки) и соответствующие выходные данные (метки или целевые переменные). Во время обучения алгоритм обрабатывает обучающие данные, изучая взаимосвязи между входными характеристиками и выходными метками. Это достигается путём настройки параметров модели для минимизации разницы между её прогнозами и фактическими метками.

После обучения модель оценивается с помощью тестового набора данных для измерения её точности и производительности. Затем производительность модели оптимизируется путём настройки параметров и использования таких методов, как перекрестная проверка для сбалансирования смещения и дисперсии. Используя размеченные входы и выходы, модель может сопоставлять входные данные и полученные результаты на точность и постепенно обучаться.

Эта модель затем используется для прогнозирования меток новых данных. Если метки дискретные (например, «да» или «нет»), то модель будет решать задачу классификации (например, определять, какие сообщения являются спамом). Если же метки непрерывные (числа), то модель будет решать задачу регрессии – к примеру, предсказывать динамику цен на недвижимость

Контролируемое машинное обучение использует наборы обучающих данных для достижения желаемых результатов. Например, вы хотите научить машину прогнозировать, сколько времени вам понадобится, чтобы доехать до дома с работы. Здесь вы начинаете с создания набора помеченных данных. Эти данные включают в себя: погодные условия, время суток, каникулы. Все эти детали являются вашими входными данными в этом примере контролируемого обучения. Результатом является количество времени, которое потребовалось, чтобы вернуться домой в этот конкретный день. Инстинктивно понимаешь, что если на улице дождь, то дорога домой займет больше времени. Но машине нужны данные и статистика.

Первое, что вам необходимо создать, — это обучающий набор, который будет содержать общее время в пути и соответствующие факторы, такие как погода, время и т. д. На основе этого обучающего набора ваша машина может обнаружить прямую зависимость между количеством дождя и временем, которое вам понадобится, чтобы добраться домой. Таким образом, он устанавливает, что чем сильнее идет дождь, тем дольше вам придется ехать, чтобы вернуться домой. Он также может увидеть связь между временем, когда вы уходите с работы, и временем, когда вы будете в дороге.

Это начало вашей модели данных. Это начинает влиять на то, как дождь влияет на то, как люди водят машину. Также становится заметно, что все больше людей путешествуют в определенное время суток.

Для реализации контролируемого машинного обучения понадобятся следующие алгоритмы: регрессии, логистической регрессии, классификации, наивные байесовские классификаторы, деревья решений, машина опорных векторов.

Проблемы, с которыми сталкиваются при контролируемом машинном обучении:

-нерелевантная входная функция, присутствующая в обучающих данных, может дать неточные результаты;

-подготовка и предварительная обработка данных всегда представляют собой сложную задачу;

-точность страдает, когда в качестве обучающих данных вводятся невозможные, маловероятные и неполные значения.

Преимущества контролируемого машинного обучения:

-обучение под наблюдением позволяет собирать данные или производить вывод данных на основе предыдущего опыта;

-помогает оптимизировать критерии производительности, используя опыт;

-помогает решать различные типы реальных вычислительных задач.

Недостатки контролируемого обучения:

-граница принятия решения может быть переобучена, если в вашем обучающем наборе нет примеров, которые вы хотите иметь;

-во время обучения классификатора нужно выбрать множество хороших примеров;

-сортировка больших данных может стать настоящим испытанием;

-обучение контролируемому обучению требует много вычислительного времени.

3.1.2. Неконтролируемое обучение (обучение без учителя)

При неконтролируемом обучении алгоритмы машинного обучения используются для анализа и группирования наборов неразмеченных данных. Эти алгоритмы выявляют шаблоны в данных без вмешательства человека (поэтому они «неконтролируемые»). Обучение без учителя работает путем анализа немаркированных данных для выявления закономерностей и взаимосвязей. Данные не помечены никакими предопределенными категориями или результатами, поэтому алгоритм должен находить эти закономерности и взаимосвязи самостоятельно. Это может быть сложной задачей, но она также может быть очень полезной, поскольку позволяет получить представление о данных, которые не были бы очевидны из помеченного набора данных.

Входные данные для моделей обучение без учителя следующие:

-неструктурированные данные: могут содержать зашумленные (бессмысленные) данные, пропущенные значения или неизвестные данные;

-немаркированные данные, которые содержат только значение входных параметров, целевого значения (выходных данных) нет. Данные легко собрать по сравнению с помеченным при контролируемом подходе.

Модели неконтролируемого обучения используются для выполнения трех основных задач— кластеризации, ассоциации и снижения размерности:

1. Кластеризация— это метод интеллектуального анализа данных, применяемый для группирования неразмеченных данных исходя из их сходств и различий. Например, в рамках алгоритмов кластеризации по K-средним похожие точки данных объединяются в группы, где значение K представляет размер группы и степень структурированности. Этот метод подходит для сегментации рынка, сжатия изображений и т.д.

2. Ассоциация— метод неконтролируемого обучения, в котором для выявления взаимосвязей между переменными и заданным набором данных используются определенные правила. Метод изучения правил ассоциации выявляет некоторые очень полезные взаимосвязи между параметрами большого набора данных и, в основном, используется для анализа рыночной корзины, который помогает лучше понять взаимосвязь между различными продуктами. Например, магазины используют алгоритмы, основанные на этой технике, чтобы выяснить взаимосвязь между продажей одного товара без учителя и продажами другого на основе поведения покупателей.

3. Снижение размерности. Снижение размерности при предварительной обработке данных может превратить огромный набор данных с сотнями функций в меньший, но по-прежнему значимый набор ключевых измерений, который легче обрабатывать и визуализировать. Представьте набор данных из 100 характеристик учащихся (рост, вес, оценки и т.д.). Чтобы сосредоточиться на ключевых чертах, сведите его всего к двум характеристикам: росту и оценкам, что упростит визуализацию или анализ данных. Этот метод часто используется на этапе обработки данных, например, когда авто кодировщики удаляют помехи из визуальных данных для повышения качества изображения.

Неконтролируемое обучение может обнаружить аномальные шаблоны данных. Благодаря этому банки могут обнаруживать мошеннические транзакции, которые не соответствуют обычной схеме расходов. Также это можно использовать для выявления аномалий в структуре сетевого трафика, которые могут быть признаком кибератаки.

Неконтролируемое обучение группирует гены, которые экспрессируются сходным образом, тем самым помогая исследователям определить функцию гена и возможные мишени для лекарств.

Алгоритмы неконтролируемого обучения могут быть задействованы в анализе медицинских карт пациентов и выявлять группы пациентов с похожими симптомами, что может быть очень полезно при ранней диагностике пациента или планировании лечения.

Ключевые проблемы обучения без учителя:

-зашумленные данные: выбросы и шум могут искажать шаблоны и снижать эффективность алгоритмов;

-зависимость от предположений: алгоритмы часто полагаются на предположения (например, формы кластеров), которые могут не соответствовать реальной структуре данных;

-риск переобучения: переобучение может произойти, когда модели фиксируют шум вместо значимых закономерностей в данных;

-ограниченное руководство: отсутствие ярлыков ограничивает возможность направлять алгоритм к конкретным результатам;

-интерпретируемость кластеров: результаты, такие как кластеры, могут не иметь четкого значения или соответствия категориям реального мира;

-чувствительность к параметрам: многие алгоритмы требуют тщательной настройки гиперпараметров, таких как количество кластеров в k-средних;

-отсутствие достоверности: в обучении без учителя отсутствуют помеченные данные, что затрудняет оценку точности результатов.

Классификация больших данных в рамках контролируемого обучения— непростая задача. Однако получаемые на выходе результаты точны и надежны. И наоборот, неконтролируемое обучение позволяет обрабатывать большие объемы данных в режиме реального времени. Однако в этом случае не хватает прозрачности в отношении кластеризации данных и существует более высокий риск получения неточных результатов. Выходом из ситуации является частично контролируемое обучение.

3.1.3. Полу контролируемое обучение

. Полу контролируемое обучение (частично контролируемое обучение) представляет собой гибридный подход, который использует преимущества как обучения под присмотром, так и без присмотра в виде единого автоматизированного анализа. Полу контролируемое обучение — это случай, когда обучение происходит на небольшом количестве размеченных и большом количестве неразмеченных данных, рис. 3.2.

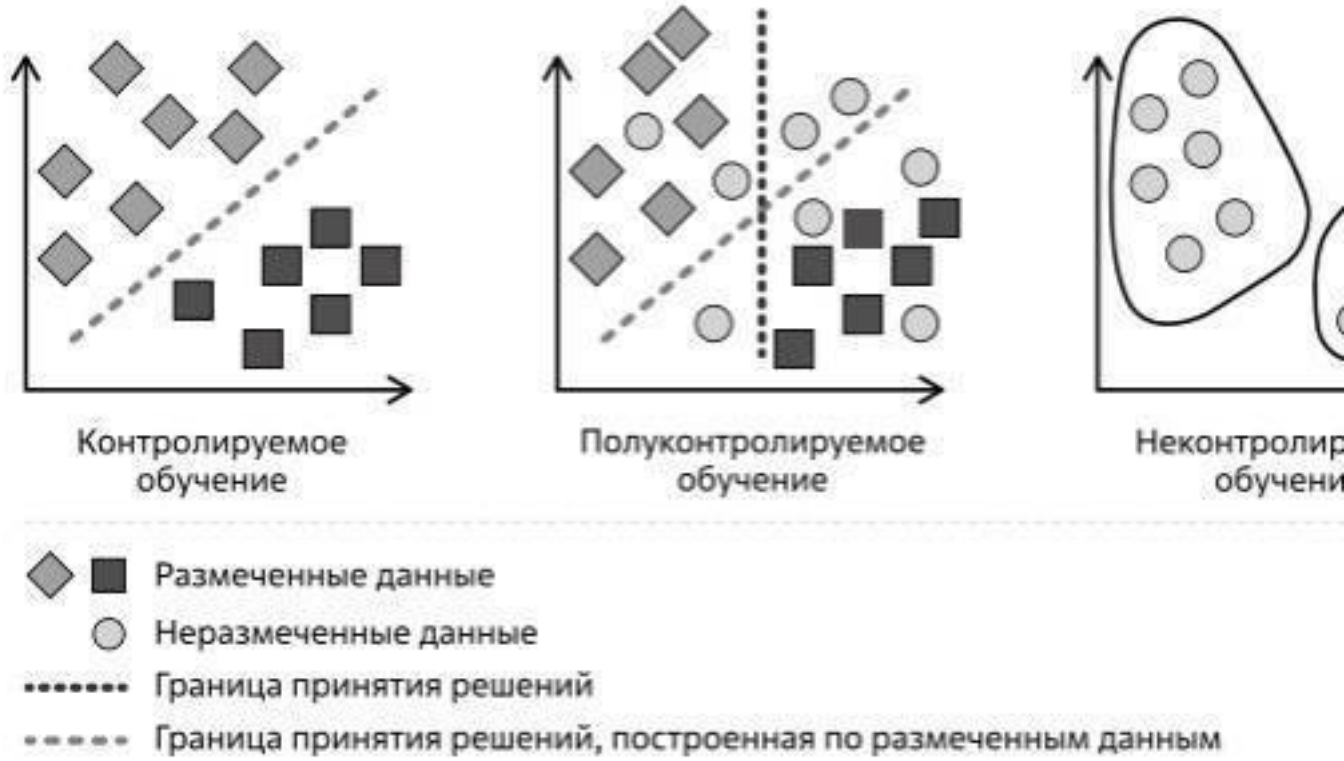


Рис.3.2. Полу контролируемое обучение

Затем оно использует шаблоны и взаимосвязи, извлеченные из помеченных данных, для расшифровки немаркированных данных.

На первый взгляд такой подход может показаться излишеством. Как оказалось, автоматизация связи между машинным обучением под присмотром и без присмотра может значительно улучшить процессы и сэкономить массу времени. Получение большого объема правильно маркированных данных может быть дорогостоящим, трудоемким и отнимать много времени. В то же время предоставление моделям самим делать полезные выводы из кучи немаркированных данных может привести к менее чем удовлетворительным результатам. Это лучшее решение, использующее эффективность сортировки данных при обучении без присмотра и предельную точность обучения под присмотром.

Требуется время, чтобы разобраться в наиболее интересных результатах обучения без учителя. Затем человек должен решить, какие наборы данных стоят потраченных усилий, вычислительных затрат и других ресурсов, которые идут на более глубокий анализ (и не забывайте, что требуется время и усилия, чтобы действительно направить нужные данные в нужном направлении). Автоматизация этих медленных, дорогостоящих и сложных шагов может значительно ускорить общий анализ и снизить вероятность человеческой ошибки.

Создание размеченных данных для машинного обучения — это длительный процесс, который часто требует квалифицированных специалистов в предметной области и достаточно высоких затрат. Процесс сбора немаркированных данных намного легче и дешевле.

Соответственно, затраты на процесс обучения с использованием только размеченных данных могут быть слишком большими, и полу контролируемое обучение может быть решением проблемы. В последнем случае используются две выборки с данными из одних и тех же классов (размеченная и неразмеченная).

3.1.4. Обучение с подкреплением

Обучение с подкреплением, (RL) — это метод машинного обучения, который учит программу взаимодействовать со средой, чтобы получить более высокую награду. Процесс состоит из нескольких компонентов:

-агент — программа машинного обучения или автономная система;

-среда — окружающий мир или виртуальное пространство, в котором агент принимает решение;

-действие — действие, которое агент совершает в среде, чтобы повлиять на ее состояние;

-состояние — конфигурация элементов внутри среды;

-вознаграждение — это положительная, отрицательная или нейтральная оценка за выполненное действие (то есть награда или наказание);

-совокупное вознаграждение — сумма всех наград, которые агент получает за серию действий в определенной задаче.

RL напоминает человеческое обучение, потому что оно также основано на методе проб и ошибок и системе вознаграждения. Представьте, как ребенок учится ходить. Каждый раз, когда он делает шаг и не падает, получает похвалу от родителей. Если ребенок упадет, становится больно, и он учится избегать таких действий в будущем. Спустя множество повторений ребенок запоминает правильные движения и начинает ходить уверенно и стабильно. RL работает аналогичным образом, но с использованием алгоритмов и вычислительных методов.

Метод обучения с подкреплением основано на трех принципах:

1.Марковский процесс принятия решений (MDP) —формальная структура для описания задач обучения. Она помогает понять, как агент должен действовать, чтобы достичь своей цели. MDP включает описание всех возможных ситуаций, в которых может оказаться агент, все возможные действия, которые он может предпринять, и то, как они влияют на ситуацию.

Также MDP описывает, какие награды получает агент за свои действия и насколько важны будущие награды по сравнению с текущими.

2.Политика —стратегия, которую агент использует для выбора действий в каждом состоянии. Она может быть простой — случайный выбор действий — или сложной, например, при использовании нейросети, обученной выбирать лучшие действия.

3.Функция ценности —помогает агенту понять, какие состояния или действия ведут к лучшим результатам в будущем. С помощью политики и функции ценности агент учится на своем опыте и принимает решения, которые максимизируют награду.

Обучение по методу RL происходит следующим образом. Агент наблюдает текущее состояние среды. Затем он выбирает действие исходя из своей политики. После выполнения действия агент получает награду и переходит в новое состояние. На основе обратной связи он обновляет свои оценки и улучшает политику. Например, если действие привело к высокой награде, агент запоминает это как хорошее действие. А если к низкой — будет стараться избегать его в будущем.

Типы алгоритмов обучения с подкреплением. Существует несколько типов алгоритмов, потому что разные задачи и условия среды требуют различных подходов к обучению. Используются различные алгоритмы, такие как Q-обучение, методы градиента политики, методы Монте-Карло и обучение временным различиям. Глубокое обучение с подкреплением основано на применении глубоких нейронных сетей для обучения с подкреплением. Одним из примеров алгоритма глубокого обучения RL является оптимизация политики региона доверия (TRPO).

Все эти алгоритмы можно разделить на две большие категории.

1. RL на основе моделей. RL на основе моделей обычно используется в тех случаях, когда среда четко определена и неизменна, а тестирование реальных сред затруднено. Сначала агент разрабатывает внутреннее представление (модель) среды. Решение выполняет действия в окружающей среде и отмечает новое состояние и ценность вознаграждения. Оно связывает переход от действия к состоянию со значением вознаграждения. По завершении модели агент моделирует последовательности действий на основе вероятности оптимального совокупного вознаграждения. Затем решение дополнительно присваивает значения самим последовательностям действий. Таким образом, агент разрабатывает различные стратегии в среде для достижения желаемой конечной цели.

2.RL без моделей. RL без моделей наиболее эффективна в больших, сложных, трудно поддающихся описанию средах, а также в неизвестных и изменчивых средах, где тестирование на основе среды не имеет существенных недостатков. Агент не разрабатывает внутреннюю модель среды и ее динамики. Вместо этого в среде используется метод проб и ошибок, чтобы оценить и отметить пары действие– состояние, а также последовательности пар действие– состояние для разработки политики.

Пример. Представьте себе беспилотный автомобиль, который должен ориентироваться в городском потоке машин. Дороги, дорожное движение, поведение пешеходов и множество других факторов могут сделать среду очень динамичной и сложной. ИИ-команды на начальных этапах обучают транспортное средство в смоделированной среде. Транспортное средство выполняет действия в зависимости от текущего состояния, получая вознаграждения или наказания. Со временем, преодолев миллионы миль по различным виртуальным сценариям, транспортное средство обучается действиям, наиболее соответствующим каждому состоянию, не моделируя всю динамику движения. При внедрении в реальный мир транспортное средство использует изученную политику, но продолжает совершенствовать ее, добавляя новые данные.

Если вы привыкли работать с задачами обучения с учителем, то в случае RL действует немного иная логика. Вместо того, чтобы создавать алгоритм, который обучается на наборе пар «факторы — правильный ответ», в обучении с подкреплением необходимо научить агента

взаимодействовать с окружающей средой, самостоятельно генерируя эти пары. Затем на них же он будет обучаться через систему наблюдений, выигрышей) и действий.

Сравнительная схема машинного обучения с подкреплением и классическим контролируемым обучением показано на рис.3.3.

Машинное обучение с подкреплением напоминает обучение с учителем, однако, в данном случае в роли «учителя» выступает окружающая (настоящая или виртуальная) среда, которая показывает, как агенты должны действовать, чтобы получить наибольшее суммарное вознаграждение, рис.3.3 б.

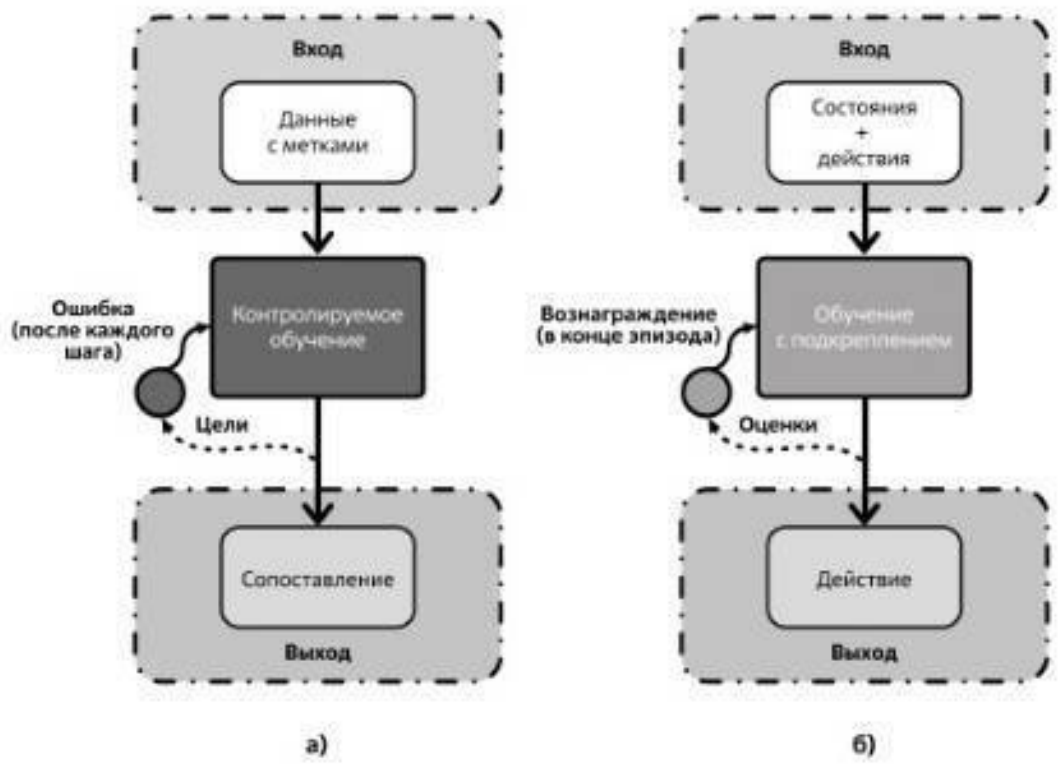


Рис. 3.3. Сравнительная схема машинного обучения

Если в случае, рис. 3.3 а, мы отвечаем на вопрос «К какому классу относится исследуемый объект?», то в случае, рис. 3.3 б, мы пытаемся ответить на вопрос «Что нужно сделать, чтобы достичь желаемого результата?». При этом мы, как правило, не знаем, хороший или плохой выбор мы делаем на каждом шаге, а получаем «вознаграждение» (подкрепление) после завершения эпизода обучения.

Преимущества обучения с подкреплением . Обучение с подкреплением (RL) имеет много преимуществ, но наиболее распространенные следующие

1.Экономия на обучении. В отличие от машинного обучения с учителем, для оценки ответов и обучения модели методом RL не нужно привлекать экспертов. Агенты сами «до обучаются» и оценивают решения, и это сильно экономит расходы.

2.Простые нейронные сети. Для RL часто используют простые нейросети, которым нужны небольшие вычислительные мощности. Это экономит затраты на оборудование. А еще такие модели обучаются быстрее, потому что у них меньше параметров и слоев по сравнению с большими сетями.

3.Не требуется разметка данных. Разметка — это процесс добавления аннотаций или меток к данным, которые помогают сделать их более понятными для моделей. Агенты используют необработанные данные из среды, в которой они будут работать.

4.Решает задачи с отложенным вознаграждением. RL может решать задачи, в которых мы не можем сразу узнать последствие каждого действия. Например, в шахматах правильность хода можно оценить только после завершения партии. В этом случае RL-агент не анализирует каждый ход отдельно, а рассматривает всю партию как последовательность действий. Победа дает высокое вознаграждение, и модель учится выбирать ходы, которые увеличивают шансы на победу, даже если краткосрочные результаты не всегда положительны.

5.Адаптивные и гибкие агенты. RL-модель получает обратную связь от окружающей среды и корректирует свои действия для адаптации к новым условиям. Благодаря этому системы могут работать в динамичных средах.

6.Высокая эффективность в сложной среде. Алгоритмы RL можно использовать в сложных средах со множеством правил и зависимостей. В той же среде человек может быть не в состоянии определить наилучший путь, даже обладая превосходными знаниями о среде. Вместо этого алгоритмы RL без моделей быстро адаптируются к постоянно меняющимся средам и находят новые стратегии для оптимизации результатов.

7.Оптимизация для достижения долгосрочных целей. Поскольку обучение RL ориентировано на максимальное увеличение вознаграждения в долгосрочной перспективе, оно эффективно для сценариев, в которых действия влекут за собой длительные последствия. Такой тип особенно хорошо подходит для реальных ситуаций, когда обратная связь по каждому шагу не всегда доступна, поскольку позволяет извлечь уроки из отсроченных вознаграждений. Например, решения о потреблении или хранении энергии могут иметь долгосрочные последствия. Обучение RL можно использовать для оптимизации энергоэффективности и затрат в долгосрочной перспективе. При наличии соответствующей архитектуры агенты RL также могут обобщать выученные стратегии для выполнения похожих, но не идентичных задач.

8.Финансовые прогнозы. Динамика финансовых рынков сложна, статистические свойства со временем меняются. Алгоритмы RL могут оптимизировать долгосрочную прибыль, учитывая транзакционные издержки и адаптируясь к рыночным изменениям. Например, алгоритм наблюдает за правилами и закономерностями фондового рынка, прежде чем тестировать действия и регистрировать соответствующие вознаграждения. Алгоритм динамически создает значение функции и разрабатывает стратегию максимального увеличения прибыли.

Обучение с подкреплением помогает системам корректировать свое поведение после каждого действия и становиться умнее

Используют обучение с подкреплением в следующих областях.

1.В электронной коммерции и поисковых системах для создания персонализированных рекомендаций. Алгоритмы RL изучают поведение пользователя, его предпочтения и историю поисков, чтобы предложить именно те товары или контент, которые будут ему интересны. Например, так работает подборка видеороликов на YouTube.

2. Роботы. RL обучает роботов навигации в сложных и динамичных средах. Агенты получают обратную связь из окружающей среды и корректируют свои действия для выполнения задач. Так роботы-пылесосы учатся передвигаться по квартире и обходить препятствия. А промышленные роботы — контролировать перемещение множества деталей по линиям и сборку изделий.

3.Машины с автопилотом. С помощью RL автономные машины учатся адаптироваться к дорожным условиям, оптимизировать маршруты и парковаться в ограниченных пространствах. Автомобили Tesla используют RL для обучения в реальных условиях вождения. Система получает данные от сенсоров и камер, анализирует их и принимает решения на основе накопленного опыта.

4.Боты для игр. RL обучает ботов реагировать на поведение других игроков. Агенты получают обратную связь на основе игровых результатов и корректируют свои стратегии. Например, компания DeepMind создала бота AlphaGo, который с помощью RL обыграл чемпионов мира по игре Го.

ГЛАВА 4. ГЛУБОКОЕ ОБУЧЕНИЕ

Глубокое обучение является мощным шагом в развитии искусственного интеллекта. Это подраздел машинного обучения, который использует искусственные нейронные сети для эмуляции работы человеческого мозга. Глубокое обучение позволяет создавать модели ИИ, которые способны распознавать сложные образы, работать с естественными языками и предсказывать результаты на основе огромного объема данных.

Процесс обучения называется глубоким, так как структура искусственных нейронных сетей состоит из нескольких входных, выходных и скрытых слоев. Каждый слой содержит единицы, преобразующие входные данные в сведения, которые следующий слой может использовать для определенной задачи прогнозирования. В отличие от традиционных методов машинного обучения, глубокое обучение способно автоматически извлекать признаки из данных, что делает его особенно эффективным для обработки больших объемов информации. Это позволяет моделям глубокого обучения решать задачи, которые ранее были недоступны для традиционных алгоритмов.

Глубокое обучение, как правило, представляют собой сети с прямой связью, в которых данные передаются от входного уровня к выходному уровню без обратной связи. Сначала создается карта виртуальных нейронов и назначаются случайные числовые значения или «веса» соединениям между ними. Веса и входные данные умножаются и возвращают выходной сигнал от 0 до 1. Если сеть не точно распознала конкретный шаблон, алгоритм будет корректировать весовые коэффициенты, до тех пор, пока не определит коэффициенты, правильно обрабатывающие данные.

Еще одним важным свойством нейронных сетей в глубоком обучении является их способность к передаче знаний. После обучения на большом наборе данных, нейронная сеть может быть использована для решения аналогичных задач на новых данных. Это открыло двери к созданию широкого спектра приложений, включая автономные автомобили, медицинскую диагностику, рекомендательные системы и многие другие.

В целом, нейронные сети играют важную роль в глубоком обучении, обеспечивая мощный инструментарий для анализа и обработки сложных данных. Их способность к обучению на основе опыта и передаче знаний позволяет достичь высокой точности при решении различных задач в области искусственного интеллекта.

Применение алгоритма обратного распространения ошибки является одним из известных методов, используемых для глубокого обучения нейронных сетей прямого распространения (такие сети ещё называют многослойными персептронами). после каждого прохода по сети обратное распространение выполняет проход в обратную сторону и регулирует параметры модели (веса и смещения). Обратное распространение играет решающую роль в совершенствовании нейронных сетей с течением времени и вычисляет градиент функции потерь относительно каждого веса с использованием правила цепочки, что позволяет эффективно обновлять веса.

С обратным распространением процесс обучения становится автоматизированным, и модель может самостоятельно настраиваться для оптимизации своей производительности.

Алгоритм обратного распространения включает два основных этапа: прямой переход и обратный переход.

В прямом проходе входные данные подаются во входной слой. Эти входные данные в сочетании с их соответствующими весами передаются скрытым слоям.

Например, в сети с двумя скрытыми слоями (h1 и h2, рис.4.1, выходные данные из h1 служат входными данными для h2.

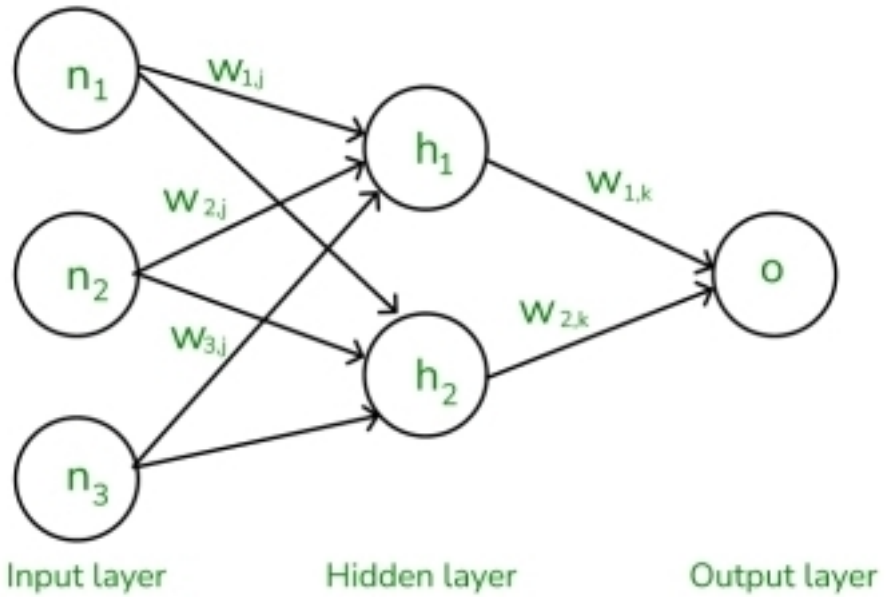


Рис.4.1. Прямой переход с использованием весов и смещений

Перед применением функции активации к взвешенным входным данным добавляется смещение. К каждому скрытому слою применяется функция активации, подобная ReLU (исправленная линейная единица), которая возвращает входные данные, если они положительные, или ноль в противном случае. Это добавляет нелинейности, позволяя модели изучать сложные взаимосвязи в данных. Наконец, выходные данные с последнего скрытого уровня передаются на выходной уровень, где активируется функция, такая как softmax, которая преобразует взвешенные выходные данные в вероятности для классификации.

При обратном прохождении ошибка (разница между прогнозируемым и фактическим результатом) распространяется обратно по сети для корректировки весов и смещений. Одним из распространенных методов вычисления ошибок является среднеквадратичная ошибка (MSE), задается как: MSE= (PredictedOutput−Actual Output)2MSE= (Прогнозируемый результат−Фактический результат)2

После вычисления ошибки сеть корректирует веса, используя градиенты, которые вычисляются с помощью правила цепочки. Эти градиенты указывают, насколько следует скорректировать каждый вес и смещение, чтобы минимизировать ошибку на следующей итерации. Обратный переход продолжается слой за слоем, гарантируя, что сеть обучается и улучшает свою производительность. Функция активации через свою производную играет решающую роль в вычислении этих градиентов во время обратного распространения.

Обратное распространение- мощный алгоритм глубокого обучения, в основном, используемый для обучения искусственных нейронных сетей, в частности, сетей прямой связи.

При обратном распространении часто используются алгоритмы оптимизации, такие как градиентный спуск или стохастический градиентный спуск. Алгоритм вычисляет градиент, используя цепное правило из calculus, что позволяет ему эффективно перемещаться по сложным слоям в нейронной сети для минимизации функции затрат.

Цель обучения нейросети при использовании алгоритма обратного распространения ошибки — это такая подстройка весов нейросети, которая позволит при приложении некоторого множества входов получить требуемое множество выходов нейронов (выходных нейронов). Можно назвать эти множества входов и выходов векторами. В процессе обучения пред-

полагается, что для любого входного вектора существует целевой вектор, парный входному и задающий требуемый выход. Эту пару называют обучающей. Работая с нейросетями, мы обучаем их на многих парах. Распространение сигнала нейросети в процессе обучения нейросети представлено на рис.4.2.

Рис. 4.2. Распространение сигнала нейросети в процессе обучения

При пошаговой реализации метода обратного распространения ошибки необходимо выполнить следующие действия:

1.инициализировать синаптические веса случайными маленькими значениями,

2. выбрать из обучающего множества очередную обучающую пару; подать на вход сети входной вектор,

3. выполнить вычисление выходных значений нейронной сети.

4. посчитать разность между выходом нейросети и требуемым выходом (речь идёт о целевом векторе обучающей пары).

5. скорректировать веса сети в целях минимизации ошибки,

6. повторять для каждого вектора обучающего множества шаги 2-5, пока ошибка обучения нейронной сети на всём множестве не достигнет уровня, который является приемлемым.

Алгоритм обратного распространения ошибки позволяет нейронным сетям с глубокой архитектурой обучаться на больших наборах данных и достигать высокой точности в различных задачах, таких как классификация, распознавание образов и генерация текста.

В первую очередь к глубокому обучению относятся следующие методы и их вариации:

-определённые системы обучения без учителя, такие как, сеть глубоких убеждений, генеративно-состязательная сеть,

-определённые системы обучения с учителем, такие как свёрточная нейронная сеть, которая вывела на новый уровень технологии распознавания образов,

-рекуррентные нейронные сети, позволяющие обучаться на процессах во времени,

-рекурсивные нейронные сети, позволяющие включать обратную связь между элементами схемы и цепочками

4.2. Сеть глубоких убеждений

Сеть глубоких убеждений (Deep Belief Network, DBN) — это вероятностный алгоритм глубокого обучения без учителя. Слои в DBN действуют как детектор признаков. После применения обучения, DBN может быть, до обучена с применением обучения с учителем. DBN можно представить как композицию ограниченных машин Больцмана (RBM) и автоэнкодеров.

Сети глубоких убеждений используют несколько математических концепций, сочетая теорию вероятностей со структурами нейронных сетей. По своей сути, они используют ограниченные машины Больцмана (RBM) для послойного обучения, которые основаны на вероятностных графических моделях.

1. Модель, основанная на энергии. Каждый RBM в DBN представляет собой модель, основанную на энергии. Для RBM с видимыми единицами измерения v и скрытыми единицами измерения h энергетическая функция определяется как:

$$E(v,h) = -\sum_{я} a_{я} v_{я} - \sum_{j} b_{j} h_{j} - \sum_{я,j} v_{j} h_{j} w_{ij}$$

Здесь ai и bj являются терминами предвзятости, а wij представляет веса между единицами измерения.

2. Распределение вероятностей. Вероятность данного состояния RBM определяется распределением Больцмана:

$$P(v,h) = \frac{e^{-E(v,h)}}{Z}$$

где Z - функция разделения, коэффициент нормализации, вычисляемый как сумма по всем возможным парам видимых и скрытых единиц измерения.

3. Обучение с использованием контрастивной дивергенции. RBM обычно обучаются с использованием метода, называемого контрастивной дивергенцией (CD). Этот метод аппроксимирует градиент логарифмического правдоподобия и обновляет веса wij, а также смещает ai, bj, чтобы максимизировать вероятность обучающих данных в рамках модели.

В DBN эти RBM собраны в стопку. Скрытый слой одного RBM служит видимым слоем для следующего. После такого неконтролируемого послойного обучения вся сеть может быть точно настроена с использованием контролируемых методов, таких как обратное распространение, целью которого является минимизация разницы между прогнозируемым результатом и фактической меткой обучающих данных

RBM представляет собой мощные генеративные модели, которые широко используются для различных приложений, таких как уменьшение размерности, изучение функций и совместная фильтрация. RBM являются строительными блоками DBN и представляет собой двухуровневую нейронную сеть, которая изучает распределение вероятностей входных данных. Каждый уровень в DBN обычно является RBM.

В самом простом варианте автоэнкодер – это нейронная сеть, рис.4.3, которая сначала кодирует входной сигнал в некоторое скрытое состояние, размерность которого, как правило, меньше размерности входного сигнала, а затем, из скрытого состояния снова разворачивает (декодирует) данные в другое, новое состояние.

Размерности входных и выходных векторов, в общем случае, могут отличаться. Например, можно попробовать обучить автоэнкодер масштабировать изображения. В другом примере, сжатия данных, декодер должен как можно точнее воспроизвести входное изображение, опираясь только на вектор скрытого состояния. Сам же этот вектор будет представлять сжатое изображение.

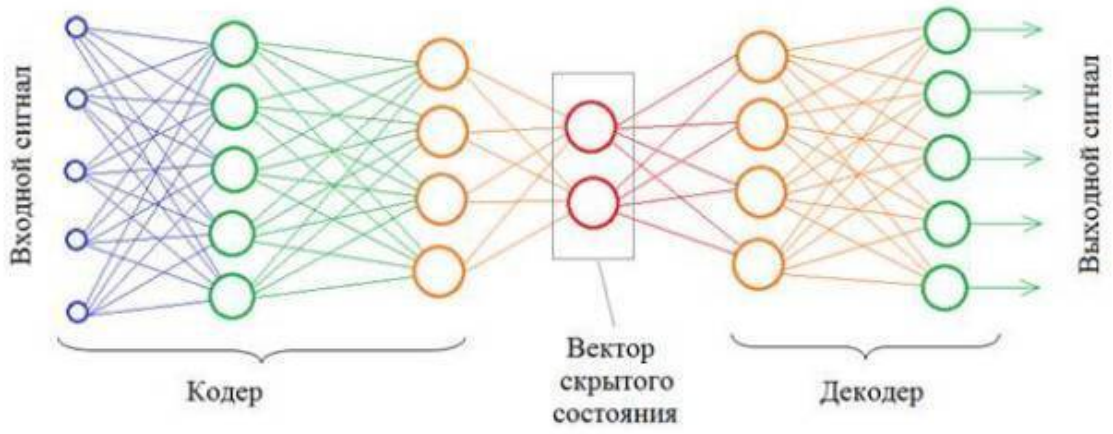


Рис. 4.3. Структура автоэнкодера

Сеть глубоких убеждений - это один из типов глубоких нейронных сетей, состоящая из нескольких скрытых слоев, в которых нейроны внутри одного слоя не связаны друг с другом, но связаны с нейронами соседнего слоя.

DBN состоят из нескольких уровней случайных, или случайно определяемых, блоков. Каждый уровень в DBN предназначен для извлечения различных функций из входных данных, при этом нижние уровни идентифицируют базовые шаблоны, а более высокие уровни распознают более абстрактные концепции. Эта структура позволяет DBN эффективно изучать сложные представления данных, что делает их особенно полезными для таких задач, как распознавание изображений и речи, где входные данные многомерны и требуют глубокого понимания.

DBN работают в две основные фазы: предварительное обучение и точная настройка. На этапе предварительного обучения сеть учится представлять входные данные слой за слоем. Каждый уровень обучается независимо как RBM, что позволяет сети эффективно изучать сложные представления данных. На этом этапе сеть изучает распределение вероятностей входных данных, что помогает ей понять базовую структуру данных.

На этапе точной настройки DBN настраивает свои параметры для конкретной задачи, такой как классификация или регрессия. Обычно это делается с использованием метода, известного как обратное распространение, при котором оценивается производительность сети при выполнении задачи, а ошибки используются для обновления параметров сети. Этот этап часто включает обучение под наблюдением, когда сеть обучается с использованием помеченных данных.

DBN используют стохастические модули, которые принимают решения вероятностно. Эта стохастическая природа позволяет сети исследовать и усваивать более сложные закономерности в данных.

Базы данных обучаются послойно, что эффективно и помогает в изучении глубоких представлений данных.

Алгоритм используется на этапе предварительной подготовки DBN. Каждый уровень обучается независимо от других, что упрощает процесс обучения.

На этапе точной настройки обратное распространение используется для задач обучения под наблюдением. Она настраивает параметры сети для повышения ее производительности при выполнении конкретных задач.

Внедрение глубоких сетей доверия (DBNs). Для реализации сетей глубоких убеждений (DBN) сначала необходимо установить numpy, pandas и scikit-учиться !pip install numpy pandas scikit-learn.

Приведенный код описывает процесс создания сети глубоких убеждений (DBN) с использованием Python. Вот пошаговое объяснение:

1.Импорт библиотек. Импортированы основные библиотеки Python для обработки данных (numpy, pandas), модели машинного обучения (scikit-learn) и глубокого обучения (tensorflow).

2.Загрузить набор данных: Набор данных MNIST, представляющий собой набор изображений рукописных цифр размером 28x28 пикселей, извлекается с помощью fetch_openml из scikit-learn. Этот набор данных обычно используется для сравнительного анализа алгоритмов классификации.

3.Предварительная обработка. Набор данных разбивается на обучающий и тестовый наборы с помощью train_test_split. Затем данные масштабируются с помощью StandardScaler для их нормализации, что часто приводит к повышению производительности нейронных сетей.

4.Уровень RBM. Ограниченная машина Больцмана инициализируется с заданным количеством компонентов и скоростью обучения. RBM - это неконтролируемые нейронные сети, которые находят закономерности в данных путем реконструкции входных данных.

5.Уровень классификатора. Для конечного уровня прогнозирования выбран классификатор логистической регрессии. Логистическая регрессия-это простая, но эффективная линейная модель для задач классификации.

6.Конвейер DBN. RBM и модель логистической регрессии объединены в конвейер. Это позволяет последовательно применять RBM (для извлечения признаков) с последующей логистической регрессией (для классификации).

7.Обучение. Конвейер, формирующий DBN, обучается на предварительно обработанных обучающих данных (X_train_scaled). RBM изучает функции, которые затем используются моделью логистической регрессии для классификации цифр.

8.Оценка.Наконец, производительность обученной DBN оценивается на тестовом наборе. Точность классификации (dbn_score) выводится для обеспечения количественного показателя того, насколько хорошо работает модель.

Эта реализация DBN использует простой, но эффективный набор моделей для обучения на основе данных и выполнения цифровой классификации. Уровни RBM действуют как детекторы признаков, преобразуя исходные значения интенсивности пикселей в более полезные представления для классификации модели логистической регрессии.

4.3. Генеративно-состязательная нейросеть

Генеративно-состязательная нейросеть (GAN) – это модель машинного обучения, умеющая имитировать заданное распределение данных. GAN состоят из двух нейронных сетей, одна из которых обучена генерировать данные, а другая – отличать смоделированные данные от реальных (отсюда и «состязательный» характер модели).

GAN состоят из двух основных компонентов: генератора и дискриминатора. Эти две сети работают вместе в процессе, который можно сравнить с игрой между двумя соперниками.

Генератор — это нейронная сеть, которая создает новые данные, основываясь на случайном шуме. Его задача — генерировать данные, которые настолько похожи на реальные, что дискриминатор не сможет отличить их от настоящих. Генератор учится создавать данные, которые максимально приближены к реальным, используя обратную связь от дискриминатора.

Генератор может быть построен на основе различных архитектур нейронных сетей, таких как полно связные сети или сверточные нейронные сети (CNN). Важно, чтобы генератор мог эффективно преобразовывать случайный шум в данные, которые выглядят реалистично.

Дискриминатор — это нейронная сеть, которая оценивает данные и пытается определить, являются ли они реальными или сгенерированными. Его задача — стать настолько точным, чтобы уметь различать настоящие данные и подделки, созданные генератором. Дискриминатор получает на вход как реальные данные, так и данные, созданные генератором, и учится различать их.

Дискриминационные алгоритмы пытаются классифицировать входные данные. Учитывая особенности полученных данных, они стараются определить категорию, к которой они относятся. Дискриминатор также может быть построен на основе различных архитектур нейронных сетей, таких как сверточные нейронные сети. Основная цель дискриминатора — минимизировать ошибку классификации, чтобы точно определять, какие данные являются реальными, а какие — сгенерированными.

Процесс обучения состоит из следующих этапов.

1.Инициализация. Генератор и дискриминатор инициализируются случайными весами. Это начальный этап, когда обе сети еще не обучены и их выходные данные могут быть случайными и некачественными.

2.Генерация данных. Генератор создает данные на основе случайного шума. Эти данные могут быть изображениями, текстами, звуками или любыми другими типами данных, в зависимости от задачи.

3.Оценка данных. Дискриминатор оценивает как реальные данные, так и данные, созданные генератором. Он пытается определить, какие данные являются настоящими, а какие — подделками.

4.Обновление весов. Весы генератора и дискриминатора обновляются на основе их ошибок. Генератор стремится уменьшить ошибку дискриминатора, а дискриминатор — увеличить свою точность. Этот процесс повторяется множество раз, пока генератор не научится создавать данные, которые дискриминатор не сможет отличить от реальных.

Полезно сравнить генеративные состязательные сети с другими нейронными сетями, такими как автокодеры (автоэнкодеры) и вариационные автокодеры.

Автокодеры кодируют входные данные в векторы. Они создают скрытое или сжатое представление необработанных данных. Они полезны при уменьшении размерности: вектор, служащий в качестве скрытого представления, сжимает необработанные данные в меньшее количество. Автокодеры могут быть сопряжены с так называемым декодером, который позволяет восстанавливать входные данные на основе их скрытого представления, как и в случае с машиной Больцмана.

Конец ознакомительного фрагмента.