

Учреждение образования  
«Белорусский государственный педагогический университет  
имени Максима Танка»

Физико-математический факультет  
Кафедра информатики и  
методики преподавания информатики

МЕТОДЫ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ПРИ РАЗРАБОТКЕ  
АДАПТИВНЫХ ОБРАЗОВАТЕЛЬНЫХ РЕСУРСОВ ПО ИНФОРМАТИКЕ

Курсовая работа студента  
240421 группы 3 курса  
Специальности физика-  
информатика  
Дневной формы  
получения образования  
\_\_\_\_\_ Жосткина  
Ивана Николаевича

Допущен к защите  
Заведующий кафедрой

\_\_\_\_\_

Защищена «\_\_»\_\_ 2024 г  
с отметкой «\_\_\_\_\_»

Научный руководитель  
Доцент, кандидат физико-  
математических наук  
Заборовский                      Георгий  
Александрович  
\_\_\_\_\_

Минск, 2024

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
ГЛАВА 1	ТЕОРЕТИЧЕСКИЕ
ОСНОВЫ АДАПТИВНЫХ ОБРАЗОВАТЕЛЬНЫХ РЕСУРСОВ И МЕТОДОВ	
ИСКУССТВЕННОГО ИНТЕЛЛЕКТА .....	5
1.1 Адаптивные образовательные ресурсы. ....	5
1.2 Искусственный интеллект .....	9
1.2.1 История ИИ.....	9
1.2.2 Интеллектуальные системы .....	11
1.2.3 Машинное обучение .....	12
1.3 Нейронные сети .....	15
ГЛАВА 2	МОДЕЛЬ
НЕЙРОННОЙ СЕТИ.....	21
2.1 Обучение нейронной сети .....	21
2.2 Инструменты для создания модели .....	23
2.3 Построение модели.....	26
2.3.1 Сбор данных.....	26
2.3.2 Работа с csv-файлами.....	29
2.3.3 Анализ данных .....	30
ЗАКЛЮЧЕНИЕ .....	32
СПИСОК ЛИТЕРАТУРЫ .....	33

## ВВЕДЕНИЕ

В современном мире технологии искусственного интеллекта (далее – ИИ) активно внедряются во все сферы жизни, включая образование. Разработка адаптивных образовательных ресурсов по информатике с использованием ИИ является важным направлением, так как позволяет создавать персонализированные учебные ресурсы, которые адаптируются под индивидуальные особенности и потребности каждого учащегося.

Проблемы искусственного интеллекта в образовании активно обсуждаются в научно-педагогическом сообществе [1]. Они охватывают широкий спектр вопросов, от технических аспектов создания и внедрения ИИ-технологий до этических и социальных последствий их использования.

Адаптивные образовательные ресурсы представляют собой программные системы, способные изменять образовательный контент и подходы к обучению в зависимости от поведения, предпочтений и результатов обучения пользователя. Использование ИИ в этих системах позволяет не только адаптировать учебный материал под уровень знаний и скорость обучения студента, но и предоставлять рекомендации по дальнейшему обучению, анализировать ошибки и предлагать способы их исправления.

Важность и актуальность применения методов искусственного интеллекта в образовательных ресурсах по информатике обусловлена стремительным развитием цифровых технологий и необходимостью подготовки специалистов, способных эффективно работать в условиях цифровой экономики. Адаптивные образовательные ресурсы, основанные на ИИ, могут стать ключевым инструментом в достижении этой цели, так как они предоставляют уникальную возможность для персонализации обучения и оптимизации учебного процесса.

С развитием технологий машинного обучения и нейронных сетей появляется всё больше инструментов для создания сложных адаптивных систем, которые могут анализировать большие объемы данных о студентах и их обучении. Эти данные включают в себя не только ответы на задания, но и время реакции, стиль обучения, частоту ошибок и многие другие параметры, которые могут быть использованы для формирования индивидуального учебного плана.

Однако, несмотря на значительные преимущества, использование ИИ в образовании сталкивается с рядом вызовов и проблем. К ним относятся вопросы конфиденциальности и безопасности данных, этические аспекты автоматизации образовательного процесса, а также потребность в разработке учебных материалов, которые были бы совместимы с адаптивными системами.

Целью данной работы является анализ данных и построение моделей, которые могут быть применены при разработке адаптивных образовательных ресурсов по информатике, а также оценка их потенциала и ограничений. Будут рассмотрены различные подходы к адаптации образовательного процесса, включая моделирование знаний, обучение с подкреплением, нейросетевые

алгоритмы и машинное обучение. Особое внимание будет уделено анализу существующих систем сбора и анализа данных.

В ходе работы будет проведен обзор литературы об адаптивных образовательных ресурсах, анализ их структуры и функционала. Также будут изучены основные принципы и методы ИИ, которые лежат в основе этих систем. В результате исследования предполагается выявить ключевые факторы, влияющие на успешность интеграции ИИ в образовательные ресурсы, и предложить рекомендации по их оптимизации и улучшению.

Курсовая работа будет направлена на изучение перспектив и ограничений использования методов ИИ в адаптивных образовательных ресурсах по информатике. Результаты исследования помогут определить наиболее эффективные подходы к интеграции ИИ в образовательный процесс и способствовать развитию инновационных методик обучения, которые будут отвечать требованиям современного образовательного пространства.

# ГЛАВА 1

## ТЕОРЕТИЧЕСКИЕ ОСНОВЫ АДАПТИВНЫХ ОБРАЗОВАТЕЛЬНЫХ РЕСУРСОВ И МЕТОДОВ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

### 1.1 Адаптивные образовательные ресурсы.

Перед началом разработки адаптивных образовательных ресурсов необходимо понять, что именно они из себя представляют.

**Адаптивный образовательный ресурс** — это система технологий, которые в каждый момент времени анализируют результаты обучения студента, учитывают его особенности и корректируют образовательную программу, а иногда и метод обучения. Эти системы помогают достичь таких образовательных целей, как сокращение затрат на обучение, формирование индивидуальной программы, увеличение эффективности обучения и поддержание высокого уровня вовлечённости в образовательный процесс.

Адаптивное обучение представляет собой подход, который максимально учитывает индивидуальные способности и потребности обучающегося. С активным развитием информационных технологий все большее применение в сфере образования находят электронные среды обучения, которые позволяют реализовать идеи адаптивного обучения на практике.

Использование адаптивных технологий предполагает интеграцию информационных и педагогических технологий, обеспечивающих интерактивность взаимодействия субъектов образования и продуктивность учебной деятельности учащегося с применением новых информационных технологий, обеспечивающих адаптивность в рамках образовательного процесса.

Системы электронного обучения с успехом выступают в качестве интерактивных средств обучения и контроля знания, предоставляя учащемуся теоретический материал в текстовом виде, аудио- и видеоформате, сообразно его уровню знаний, оценивая усвоение материала и, что самое главное, определяя траекторию его дальнейшего движения в рамках курса или учебного плана в целом.

Изучаемый материал предоставляется учащемуся в некоторый момент времени с учетом его накопленных знаний, успеваемости, опыта. В этом реализуется адаптация при обучении, происходит адаптированное представление материалов курса, тестирование, навигация. Так, учащемуся, показавшему высокие результаты при изучении предшествующего материала, требуется предоставлять материал и задания со сложностью выше среднего. Более простой материал и задания не обладают развивающим потенциалом. С другой стороны, учащийся с низкой подготовкой не в состоянии решить

сложные задания и разобрать материал повышенной сложности, что может в конечном счете привести к снижению мотивации.

В качестве примеров адаптивных образовательных ресурсов можно привести следующее:

1. Электронные учебные пособия - они предоставляют теоретический материал в текстовом виде, аудио- и видеоформате, сообразно уровню знаний учащегося.
2. Информационно-справочные источники - такие как хрестоматии, тексты из специальных словарей и энциклопедий, научной и научно-популярной литературы.
3. Электронные учебно-методические комплексы - включают в себя различные учебные модули, которые могут быть адаптированы под индивидуальные потребности учащегося.
4. Мультимедиа среды - предоставляют динамические изображения и анимационные модели, которые помогают учащимся лучше понять изучаемые процессы и явления.

Адаптивное обучение основывается на множестве определенных и хорошо апробированных моделей и процессов. Информация в системах адаптивного обучения необходима для представления знаний о предметной области и для моделирования поведения студентов в процессе обучения. Эту информацию можно разделить на три основные модели: модель предметной области, модель студента и модель адаптации.

**Модель предметной области** содержит информацию об изучаемом предмете и используется для поддержки адаптивного изучения курса. Модель предметной области выступает в качестве хранилища данных, которое содержит название разделов, тем, их содержание и навигационные ссылки, связанные со структурой представленных данных. Модель предметной области может также содержать информацию о студентах, имеющую непосредственное отношение к их учебной деятельности, пример, информацию об учебных проектах, участниках и их ролях.

Модель предметной области состоит из двух основных частей: содержания курса и системы предоставления знаний. Последняя должна быть в состоянии поддерживать любое содержание курса, а также легко адаптироваться к новым требованиям, которые могут быть предъявлены к содержанию курса. Крайне важным аспектом модели предметной области является взаимосвязь между элементами курса и навигацией, которая и позволяет реализовать идею адаптации при изучении материала. Модель предметной области предназначена для разработки структуры взаимосвязей между отдельными элементами курса и переходов между ними с учетом способностей и потребностей пользователей. Структура этих взаимосвязей должна обеспечивать возможность студентам перехода на требуемый элемент курса в рамках адаптивного обучения.

**Модель студента** является основным компонентом систем адаптивного обучения. Данная модель включает всю информацию о студенте: его прогресс в изучении предметной области, уровень усвоения, поведение и пр. Модель студента предполагает, что информация о студенте изменяется со временем, включая новые элементы и траекторию изучения курса по мере прохождения курса студентом. То есть содержит не только общую информацию о студенте, но отслеживает все действия студента в процессе адаптивного обучения в рамках электронной образовательной системы.

В модели студента представлена информация двух типов: связанная с предметной областью и не связанная с ней. Модель студента, связанная с предметной областью, фактически является моделью знаний студента. Она описывает уровень знаний студента, его понимание предмета или отдельных его разделов, ошибки, которые студент совершил в процессе изучения, прогресс студента в изучении предметной области, его оценки за тестирование и т.д.

Модель студента, не связанная с предметной областью, представляет информацию о навыках студента, основывается на его поведении. Эта информация включает в себя цели обучения, когнитивные способности студента, такие, например, как способность рассуждать, выстраивать ассоциации, его мотивацию, начальные знания и опыт, предпочтения и пр.

**Модель адаптации** включает в себя модель предметной области и модель студента. Процесс моделирования процесса адаптации при обучении начинается с выбора наиболее репрезентативных узлов на основе анализа потребностей студентов, описанных в модели студента. Рассматриваемые узлы могут быть классифицированы по различным видам знаний: базовые знания, включая знание определений, формул и других материалов; процедурные знания, относящиеся к методам и алгоритмам решения задач предметной области; и концептуальные знания, отражающие отношения между понятиями, которые полностью описывают предметную область. Разные виды знаний предполагают разные подходы при их изучении, следовательно, узлы будут представлять разные режимы изучения. Необходимо принять решение о том, какие объекты изучения в каких узлах должны быть представлены, так, чтобы они могли быть изученными студентами при прохождении соответствующих узлов.

Модель адаптации описывает адаптивное обучение на разных уровнях абстракции. В частности, модель адаптации определяет то, что может быть адаптировано, а также каким образом это должно быть адаптировано. Модель адаптации может определять это, в том числе и неявно. Уровни абстракции, на которых может быть определена адаптация, варьируются от конкретных правил, регламентирующих поведение во время обучения, вплоть до общих спецификаций логических взаимосвязей между субъектами адаптивного обучения. Наиболее успешные и широко известные системы адаптивного обучения используют модели адаптации, которые обобщенно определяют

поведение системы на основе свойств модели содержимого, например на основе взаимоотношений между субъектами контента.

Несмотря на то что подготовка электронного курса, применяемого в рамках адаптивного обучения, требует определенного времени, тем не менее его применение при адаптивном обучении дает ряд неоспоримых преимуществ. К таким достоинствам относится снижение нагрузки на преподавателя при проведении практических занятий, появляется возможность полностью уйти от лекционных занятий или перевести их в формат мини-лекций по запросу студентов во время контактной работы с преподавателем. Использование электронных курсов в рамках адаптивного обучения позволяет высвободить и аудиторный фонд.

С педагогической точки зрения, важным является вовлеченность студента в образовательный процесс. Возможность отслеживать свой прогресс, понимать свое место в потоке прохождения всего курса, иметь представление о собственном уровне усвоения материала – все это возбуждает и поддерживает интерес студента на протяжении всего процесса изучения курса. Ярким примером такого ресурса является система MOODLE (Рисунок 1.1), которая активно используется на различных уровнях образования в Республике Беларусь.

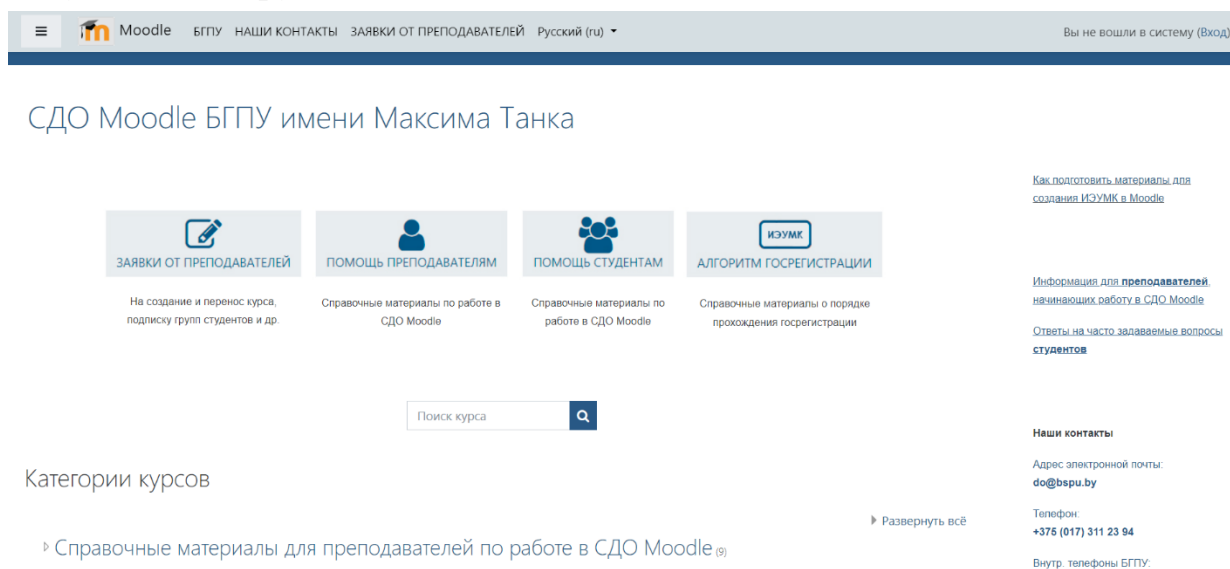


Рисунок 1.1 – Система дистанционного обучения Moodle



## 1.2 Искусственный интеллект

Что же такое искусственный интеллект (ИИ)? У разных авторов можно встретить свои определения понятия «искусственный интеллект». На сегодняшний день не существует одного общего определения. Остановимся на следующем определении ИИ.

**Искусственный интеллект** – это раздел информатики, изучающий различные аспекты моделирования мыслительной деятельности человека.

Таким образом, это наука, поставившая перед собой грандиозную цель – изучить и смоделировать мышление человека. Для этого необходимо ответить на сложные вопросы: какова природа мышления человека, какие процессы происходят в человеческом мозге, когда он видит, чувствует, думает, понимает?[2]

### 1.2.1 История ИИ

Искусственный интеллект (ИИ) начал формироваться как область знания в середине 20-го века, но его корни уходят в далекое прошлое, включая античные времена и средневековье. В последующие столетия, различные научные исследования, которые впоследствии стали основой для развития ИИ, продолжались.

С появлением компьютеров в 1940-х годах, ИИ начал развиваться как отдельное научное направление. В тот же период, Норберт Винер представил современную теорию управления, известную как кибернетика. Кибернетика изучает универсальные законы, которые применимы в системах автоматического управления, организации производства и человеческой нервной системы, и занимается вопросами информационного управления.

Термин “искусственный интеллект” впервые был использован в 1956 году в названии научного семинара в Дартмутском колледже в США. На этом семинаре обсуждались не столько вычислительные, сколько логические методы решения задач. Стоит отметить, что в английском языке слово “intelligence” означает “способность к логическому мышлению”, что не имеет такой фантастической коннотации, как в русском переводе.

Между 1956 и 1963 годами активно велись исследования по моделированию человеческого мышления и разработке первых компьютерных программ, основанных на этих моделях. В эти исследования были вовлечены специалисты из гуманитарных наук, такие как философы, лингвисты и психологи, а также ученые в области кибернетики. Были созданы и опробованы различные концепции и идеи.

#### ***1. Конец 50-х годов – модель лабиринтного поиска.***

Данный подход находит ответ на решение задачи, осуществляя поиск оптимального пути в пространстве состояний, двигаясь от входных данных к результату. Этот подход использовался в игровых программах (пятнашки, шашки, шахматы).

## ***2. Начало 60-х годов – эвристическое программирование.***

Эвристика представляет собой метод или правило, которое упрощает процесс нахождения решения задачи, избегая необходимости полного анализа всех возможных вариантов.

Эвристическое программирование — это методика, которая позволяет найти достаточно хорошее решение сложной проблемы с помощью применения известных эвристик. Это задача высокой сложности. Для изучения творческого процесса мышления человека американские ученые А. Ньюэлл и Г. Саймон провели эксперимент с группой студентов, не знакомых с математической логикой. Каждый участник пытался самостоятельно доказать математическую теорему, при этом все их рассуждения, идеи и неудачные попытки были записаны. На основе анализа этих данных исследователи выявили эвристики, которые затем были использованы для создания компьютерной программы под названием “Логик-теоретик” (1957 год, А. Ньюэлл, Г. Саймон, Дж. Шоу). С помощью “Логик-теоретика” были доказаны 38 теорем, что стало началом развития эвристического программирования.

## ***3. 1963-1970 годы – методы математической логики.***

В 1965 году Джон Алан Робинсон открыл правило резолюций, которое дает возможность автоматически доказать любую истинную теорему за ограниченное количество времени, исходя из аксиом. Параллельно с Робинсоном, выдающийся российский математик С. Ю. Маслов предложил альтернативный метод поиска доказательств, который впоследствии был назван в его честь. Метод Маслова предлагал другой подход к решению той же проблемы. В 1972 году был разработан язык логического программирования PROLOG.

Язык PROLOG открывает широкие возможности для решения задач, требующих логического рассуждения для нахождения ответа. Вычисления в рамках логического программирования заключаются в построении логического вывода, который представляет собой дерево, в корне которого находится доказываемое предложение, а ветви содержат связанные логически утверждения. Конечные точки этого дерева — это исходные данные, заданные в условиях задачи.

Конструкция дерева логического вывода включает в себя поиск связанных последовательностей утверждений, ведущих от исходных данных к цели, среди всех возможных путей. Этот набор путей называется пространством поиска для задачи. Поиск может осуществляться как в прямом направлении — от данных к цели, так и в обратном — от цели к данным. В прямом поиске из условий задачи выводятся новые утверждения, расширяя знания о задаче. В обратном поиске формируются гипотезы, которые должны соответствовать известным данным.

## ***4. Середина 1970-х годов – прорыв в США в развитии интеллектуальных систем, основанных на знаниях.***

Вместо поиска универсального алгоритма мышления пришла идея практического использования знаний экспертов в конкретной узкой области и создания на их основе компьютерной интеллектуальной системы. В США появились первые коммерческие экспертные системы. Появился подход к решению задач ИИ – представление знаний. Созданы первые экспертные системы для медицины и химии – MYCIN и DENDRAL. И лишь в начале 1980-х годов в Европе объявлена программа развития новых технологий, в которую включена проблематика ИИ.

**5. С середины 1980-х годов и по сегодняшний день** наблюдается активное развитие всех направлений ИИ. Увеличивается финансирование, создаются коммерческие интеллектуальные системы, ежегодно проводятся конференции по ИИ, издаются сотни научных журналов, книг по ИИ.

### **1.2.2 Интеллектуальные системы**

**Искусственные интеллектуальные системы** – автоматические системы, которые берут на себя отдельные функции интеллектуальной деятельности человека, например, принимать наилучшие решения, основываясь на ранее полученном опыте и анализе внешних воздействий.

Основное, что выделяет интеллектуальные системы среди прочих программных решений, — это их способность к обучению и аккумулированию знаний в процессе функционирования, а также к самоусовершенствованию и самообучению. Эти характеристики аналогичны свойствам человеческого разума: умение к обобщению, обучению, накоплению опыта и адаптации к новым условиям при решении задач. Именно благодаря этим качествам люди способны решать самые сложные задачи и быстро переключаться между различными типами задач. Следовательно, интеллектуальная система должна быть способна решать задачи, для которых не существует стандартных решений или алгоритмов.

Имеются и поведенческие определения ИИ, такие как “тест Тьюринга” от Алана Тьюринга. Этот тест предполагает взаимодействие человека и машины через обмен информацией, при этом они находятся в разных помещениях и не видят друг друга. Если в ходе такого общения невозможно определить, кто из собеседников является машиной, то машину можно признать обладающей интеллектом.

Тьюринг также предложил концепцию имитации мышления, согласно которой вместо попыток воссоздать интеллект взрослого, следует попытаться создать программу, имитирующую интеллект ребенка. Если такой “детский” интеллект будет правильно воспитан, он может превратиться в интеллект взрослого. Поэтому задача разделяется на две части: создание “программы-ребенка” и её последующее “воспитание”.

Современные разработчики интеллектуальных систем следуют этому подходу, так как заложить в систему полный объем знаний и связей между ними сразу невозможно.

### 1.2.3 Машинное обучение

**Машинное обучение** (Machine learning) – подраздел искусственного интеллекта, изучающий различные способы построения обучающихся алгоритмов. Под обучающимися алгоритмами понимаются алгоритмы, которые меняются (обучаются) каким-то образом в зависимости от входных данных.

Машинное обучение – очень обширная область знаний. Можно ведь по-разному определять слово «обучение» и каждый раз получать интересные результаты. Однако среди множества парадигм и подходов в машинном обучении выделяется одна очень интересная область – искусственные нейронные сети.

Машинное обучение основано на ряде принципов, которые обеспечивают его функционирование:

- **Данные.** В основе ML лежит использование данных. Обучающие данные предоставляют модели информацию о входных признаках и соответствующих правильных ответах. Чем более разнообразными, качественными и представительными являются данные, тем лучше модель сможет обучиться, распознавать образцы и осуществлять правильные прогнозы на новых данных.
- **Модель.** Представляет собой алгоритм или математическую функцию, которая преобразует входные данные в выходные. Модель выбирается в зависимости от задачи и типа данных. Она может быть линейной, деревом решений, нейронной сетью и т. д. Одна из ключевых целей машинного обучения – создание моделей, которые способны выдавать точные предсказания для новых данных, которые ранее не применялись в процессе обучения.
- **Обучение.** Процесс обучения состоит в подгонке модели к обучающим данным. Модель анализирует данные, выявляет закономерности и корректирует свои внутренние параметры так, чтобы минимизировать ошибку между предсказаниями модели и правильными ответами. Обучение может происходить с учителем (с правильными ответами), без учителя (без правильных ответов) или с подкреплением (с вознаграждениями или наказаниями). Вместо явного программирования модели получают знания из данных и корректируют свои параметры для достижения производительности.
- **Автоматизация.** ML стремится к автоматизации процессов и принятия решений, основанных на данных, без необходимости явного вмешательства человека. Алгоритмы ML способны выполнять сложные задачи с большой скоростью и точностью.

- **Оценка и тестирование.** После обучения модели необходимо оценить ее производительность на новых данных. Для этого используется тестовый набор данных, который модель не видела во время обучения. Оценка производится с помощью метрик, которые измеряют точность, полноту, F1-меру и другие характеристики модели. Это позволяет оценить, как модель справляется с задачей и определить необходимость дальнейшей доработки.
- **Обобщение.** Модель в ML должна быть способна делать точные прогнозы или принимать решения на новых, ранее неизвестных данных. Это свойство называется обобщением. Хорошая модель способна обобщать знания, выявлять общие закономерности и применять их к новым ситуациям.
- **Регуляризация и управление сложностью.** Когда модель становится сложной, существует риск переобучения, когда модель хорошо адаптируется к обучающим данным, но плохо обобщает на новые данные. Для контроля сложности моделей используются методы регуляризации, такие как L1 и L2 регуляризация.

Машинное обучение имеет применение как в повседневной жизни, так и в различных отраслях. Разберем несколько примеров.

Например, в повседневной жизни всем известны голосовые помощники Siri, Google Assistant и Алиса, которые используют машинное обучение для распознавания и понимания голосовых команд. Также во многих смартфонах сегодня есть функция автоматического распознавания лиц на фотографиях, которая позволяет организовать и классифицировать снимки по людям и создавать веселые видео на основе этих данных. Также одним из примеров машинного обучения являются рекомендательные системы в онлайн-платформах, такие как YouTube, КиноПоиск, Netflix, Яндекс Музыка, Spotify) которые предлагают персонализированные рекомендации по фильмам, музыке, книгам и пр.

Сфера, где мы сталкиваемся с машинным обучением почти каждый день – это финансы. Анализ финансовых данных для прогнозирования рыночных трендов и принятия решений о вложениях, определение мошеннических транзакций на основе аномалий в поведении клиентов и исторических данных, а также кредитный скоринг и оценка платежеспособности клиентов на основе их финансовой истории и других факторов.

Машинное обучение является ключевым компонентом в разработке автономных транспортных средств, позволяя им анализировать окружающую среду и принимать решения на основе данных с датчиков.

Существуют различные способы машинного обучения (Рисунок 1.2):

- Обучение с учителем, где модель обучается на основе данных, содержащих ответы (метки).

- Обучение без учителя, где модель ищет закономерности в данных без предварительно заданных ответов.
- Обучение с подкреплением, где модель учится на основе наград за выполнение определенных действий.

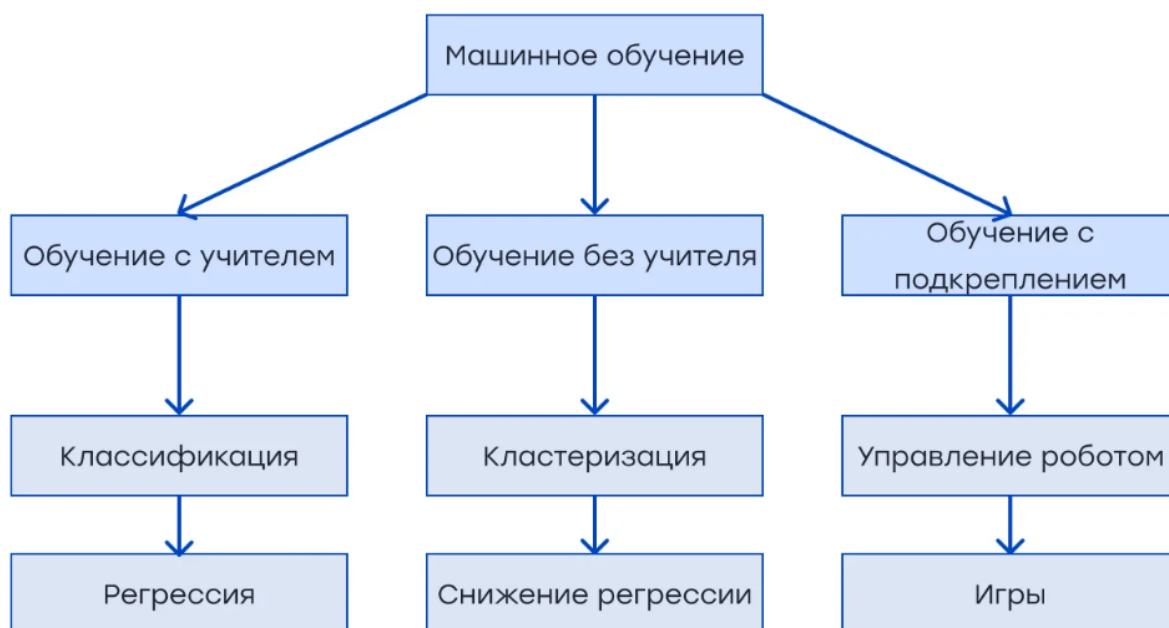


Рисунок 1.2 Способы машинного обучения

**Обучение с учителем** (Supervised learning) — это метод машинного обучения, при котором модель обучается на основе данных, содержащих входы и соответствующие им правильные ответы (метки). Этот процесс можно сравнить с обучением в школе, где учитель предоставляет ученику задачи и правильные ответы на них, а ученик учится решать подобные задачи самостоятельно.

Примеры задач:

- **Классификация** – определение принадлежности объекта к конкретному классу. Например, классификация электронных писем на спам.
- **Регрессия** – прогнозирование непрерывной целевой переменной. Например, предсказание цены недвижимости на основе ее характеристик.

**Проблемы и ограничения:** Одной из основных проблем обучения с учителем является необходимость в большом объеме размеченных данных. Сбор и разметка таких данных могут быть дорогостоящими и трудоемкими.

**Обучение без учителя** (Unsupervised learning) — это метод машинного обучения, при котором модель самостоятельно находит структуры в данных без явных указаний, что именно нужно искать. В отличие от обучения с учителем, где модель обучается на данных с известными ответами, здесь модель работает с неразмеченными данными, пытаясь самостоятельно выявить скрытые закономерности и взаимосвязи.

Примеры задач:

- **Кластеризация** – группировка схожих объектов внутри данных. Это помогает обнаружить неявные группы или кластеры в данных. Например, сегментация покупателей на основе их покупательского поведения.
- **Снижение размерности** – уменьшение размерности данных, сохраняя важные признаки и устраняя шум. Например, сжатие изображений без значительной потери информации.
- **Визуализация данных**: Обучение без учителя также может использоваться для визуализации многомерных данных в двух- или трехмерном пространстве, что помогает в анализе и интерпретации данных.

**Обучение с подкреплением** (Reinforcement Learning, RL) — это метод машинного обучения, при котором агент (искусственная система) учится принимать решения, выполняя действия в некоторой среде и получая обратную связь в виде наград или наказаний. Этот метод основан на принципе проб и ошибок, где агент экспериментирует с различными стратегиями и учится из опыта.

Примеры задач:

- **Управление роботом** – обучение робота совершать определенные действия в окружающей среде для достижения поставленных целей.
- **Игры** – обучение агента играть в игры, например, шахматы или видеоигры, чтобы достичь наивысшего возможного счета.

### 1.3 Нейронные сети

Биологические нейронные сети представляют собой совокупность биологических нейронов. Однако в таких сетях тоже много ненужных для обработки сигнала аспектов. Плюс ко всему нейронов в биологической нейросети очень много. Опять упрощаем: убираем ненужные химические и биологические компоненты, а также уменьшаем количество нейронов (Рисунок 1.3)

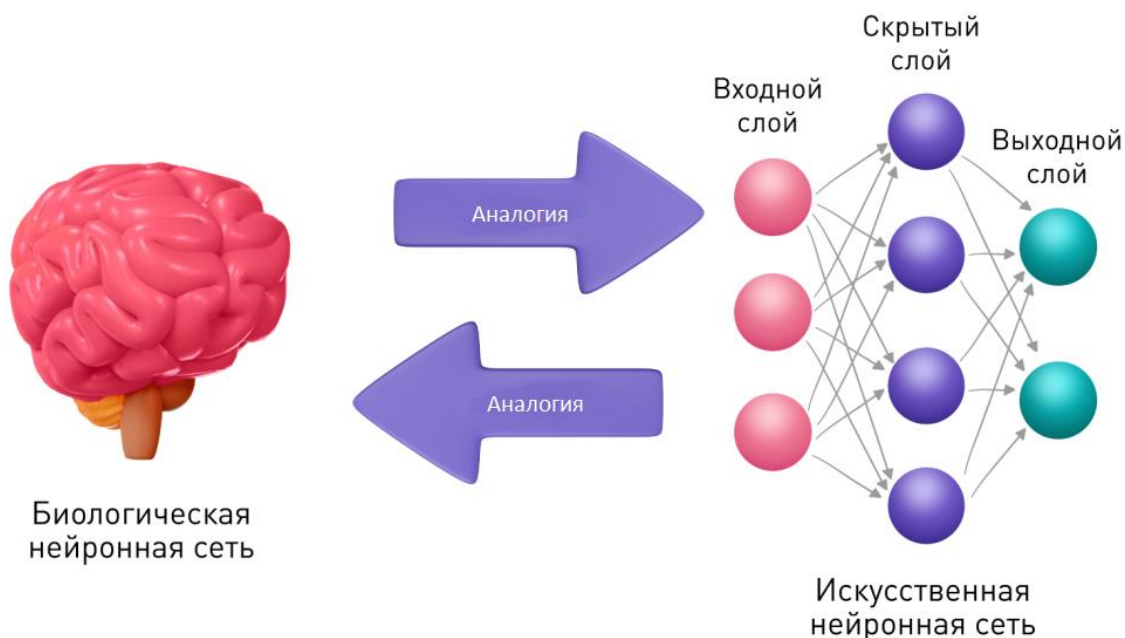


Рисунок 1.3 Искусственная нейронная сеть

У каждого биологического нейрона тысячи входов. Каждый из них соединен с выходами других нейронов. Значит, имеем тысячи синапсов на каждый нейрон. Синапс — связь между нейронами. Каждый синапс может либо усиливать, либо ослаблять проходящий через него сигнал. Более того, с течением времени синапсы могут меняться, а значит, будет меняться характер изменения сигнала. Если правильно подобрать параметры синапсов, то входной сигнал после прохода через нейронную сеть будет преобразовываться в правильный выходной сигнал.

Именно так и происходит преобразование множества входных сигналов в верное решение на выходе. Будем характеризовать каждую такую связь определенным числом, называемым весом данной связи. (Рисунок 1.4). Сигнал, прошедший через данную связь, умножается на вес соответствующей связи. Это ключевой момент в концепции искусственных нейронных сетей.

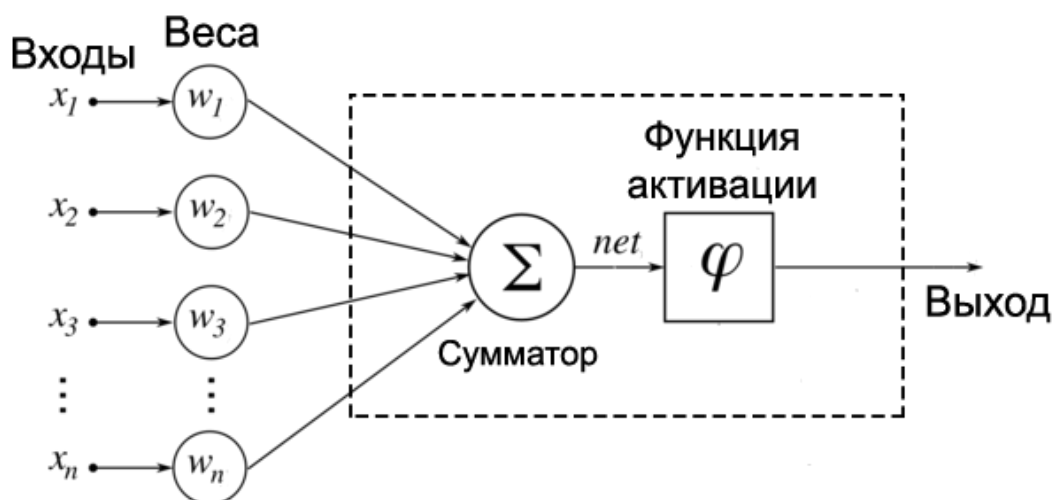


Рисунок 1.4 Модель нейрона



Поступившие на входы сигналы умножаются на свои веса (изображены кружками). Сигнал первого входа  $x_1$  умножается на соответствующий этому входу вес  $w_1$ . В итоге получаем  $x_1 w_1$ . И так до  $n$ -го входа. В итоге на последнем входе получаем  $x_n w_n$ .

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

$$x_1 w_1 + x_2 w_2 + \dots + x_n w_n = \sum_{i=1}^n x_i w_i$$

Результатом работы сумматора является число, называемое взвешенной суммой (net):

$$net = \sum_{i=1}^n x_i w_i$$

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

Для преобразования функции и получения ответа используют функцию активации. Она преобразует взвешенную сумму в какое-то число, которое и является выходом нейрона (выход нейрона обозначим переменной out). Для разных типов искусственных нейронов используют самые разные функции активации. В общем случае их обозначают символом  $\varphi(net)$ . Указание взвешенного сигнала в скобках означает, что функция активации принимает взвешенную сумму как параметр. Значение этой функции и является выходом нейрона (out):

$$out = \varphi(net)$$

Рассмотрим самые известные функции активации.

**Функция единичного скачка** – самый простой вид функции активации. Выход нейрона может быть равен только 0 или 1. (Рисунок 1.5) Если взвешенная сумма больше определенного порога  $b$ , то выход нейрона равен 1; если ниже, то 0.

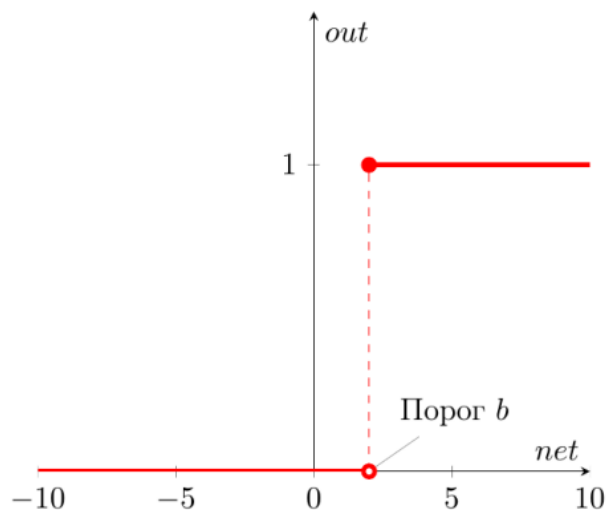


Рисунок 1.5 Функция единичного скачка

**Сигмоидальная функция.** Существует целое семейство сигмоидальных функций, некоторые из которых применяют в качестве функции активации в искусственных нейронах. Часто используемая в нейронных сетях сигмоида – логистическая функция (Рисунок 1.6).

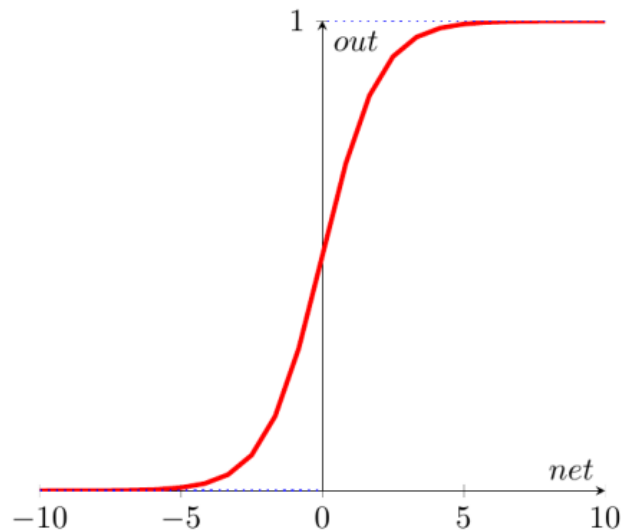


Рисунок 1.6 Сигмоидальная функция

График этой функции выглядит достаточно просто. Аналитически функция записывается следующим образом:

$$out(net) = \frac{1}{1 + e^{-a*net}}$$

где  $a$  – число, которое характеризует степень крутизны функции.

Это самые простые функции активации. Существует большое количество различных функций для разнообразных задач.

Искусственные нейронные сети состоят из совокупности искусственных нейронов. В зависимости от соединения и взаимодействия нейронов нейронные сети делятся на несколько видов.

**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 (Рисунок 1.7).

Входной слой обозначен кружками (он не считается за слой нейронной сети), а справа расположен слой обычных нейронов.

Нейроны соединены друг с другом стрелками. Над стрелками расположены веса соответствующих связей (весовые коэффициенты).

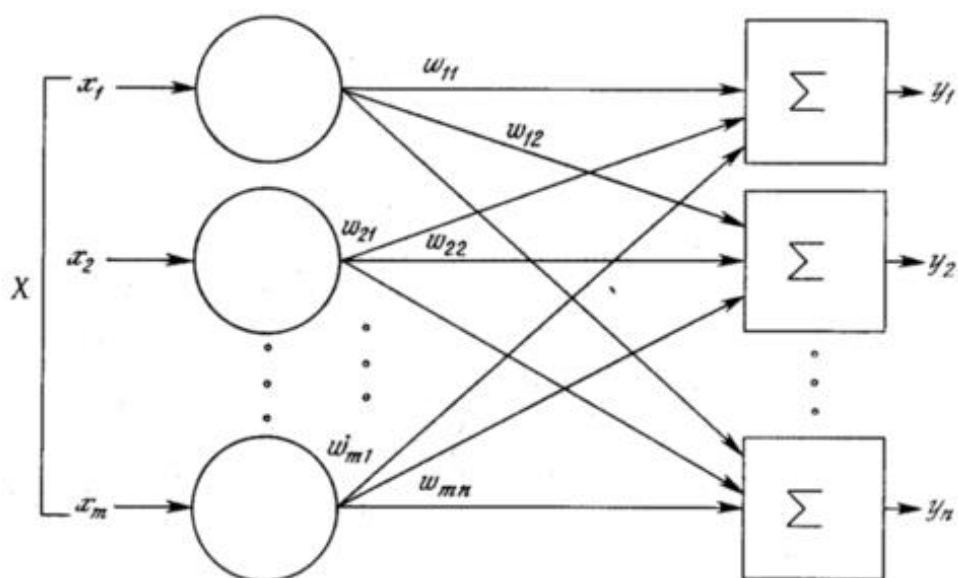


Рисунок 1.7 Однослойная нейронная сеть

**Многослойные нейронные сети.** Такие сети, помимо входного и выходного слоев нейронов, имеют еще и скрытый слой (слои). Эти слои находятся между входным и выходным слоями (Рисунок 1.8).

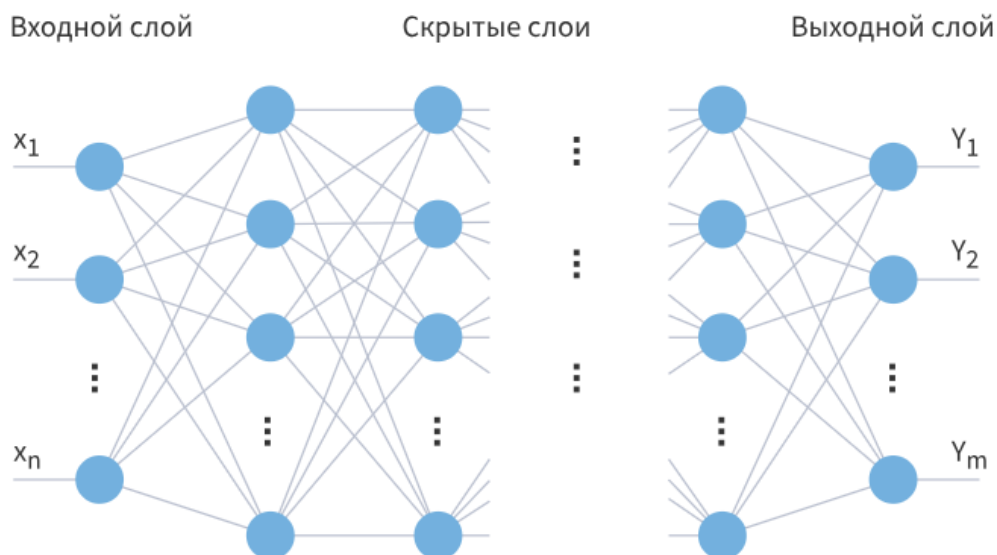


Рисунок 1.8 Многослойная нейронная сеть

Многослойные нейронные сети обладают гораздо большими возможностями, чем однослойные. Работу скрытых слоев нейронов можно сравнить с работой большого завода. Продукт (выходной сигнал) на заводе собирается по стадиям. После каждого станка получается какой-то промежуточный результат. Скрытые слои тоже преобразуют входные сигналы в некоторые промежуточные результаты.

**Сети прямого распространения.** Можно заметить одну интересную деталь на рисунках НС выше. Во всех примерах стрелки строго идут слева направо, то есть сигнал в таких сетях идет строго от входного слоя к выходному.

Сети прямого распространения (Feedforward neural network) (feedforward сети) – искусственные НС, 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.

**Сети с обратными связями.** В сетях такого типа сигнал может идти и в обратную сторону.

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 (Рисунок 1.9). 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).

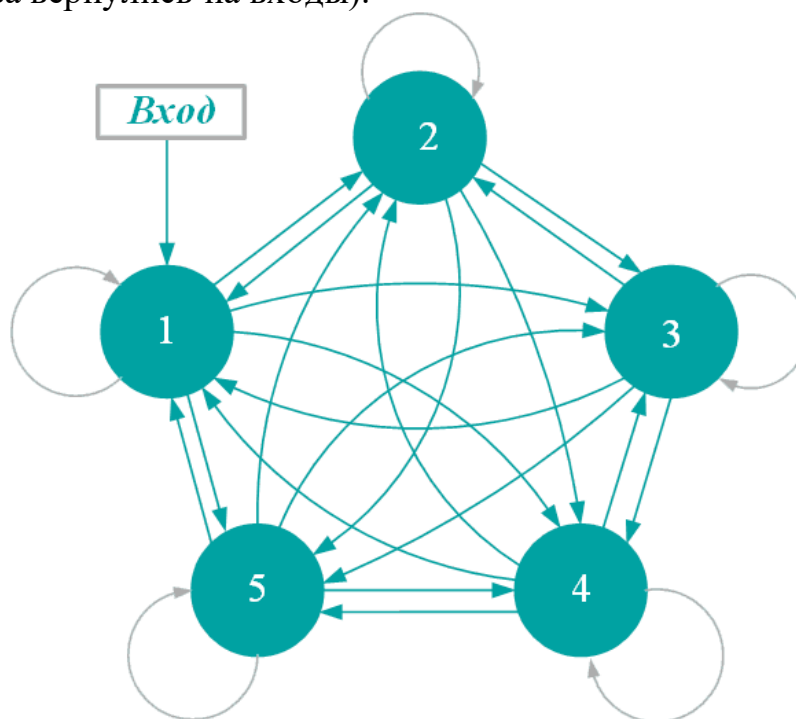


Рисунок 1.9 Сеть с обратными связями

Итак, под нейронной сетью обычно понимается структура, состоящая из связанных между собой нейронов. К настоящему времени предложено большое количество способов объединения нейронов в НС. Нейроны в сети расположены слоями. Очевидно, что для адекватного решения задачи функционирования сети нужно правильно выбрать значения весов связей между нейронами – обучить сеть.

## ГЛАВА 2

# МОДЕЛЬ НЕЙРОННОЙ СЕТИ

### 2.1 Обучение нейронной сети

**Обучение нейронной сети** (Training) – поиск такого набора весовых коэффициентов, при котором входной сигнал после прохода по сети преобразуется в нужный нам выходной.

Этот метод, применяемый к “обучению нейронных сетей”, также схож с процессами, происходящими в биологических нервных системах. Наш мозг формируется множеством взаимосвязанных нейронных сетей, каждая из которых состоит из нейронов с одинаковой функцией активации. Обучение происходит за счет модификации синапсов, которые модулируют силу входящего сигнала.

Тем не менее, существует важный аспект. Обучение сети с использованием только одного входного сигнала приведет к тому, что сеть просто запомнит правильный ответ, создавая иллюзию быстрого обучения. Однако при подаче даже слегка измененного сигнала, нейронная сеть даст некорректный результат. Нам необходимы нейронные сети, способные к обобщению, чтобы они могли распознавать лица на различных фотографиях, а не только на одной.

Именно с этой целью и создаются обучающие выборки.

- **Обучающая выборка** (Training set) – конечный набор входных сигналов (иногда вместе с правильными выходными сигналами), по которым происходит обучение сети. После обучения сети, то есть когда сеть выдает корректные результаты для всех входных сигналов из обучающей выборки, ее можно использовать на практике. Однако прежде производят оценку качества ее работы на так называемой тестовой выборке.
- **Тестовая выборка** (Testing set) – конечный набор входных сигналов (иногда вместе с правильными выходными сигналами), по которым происходит оценка качества работы сети.

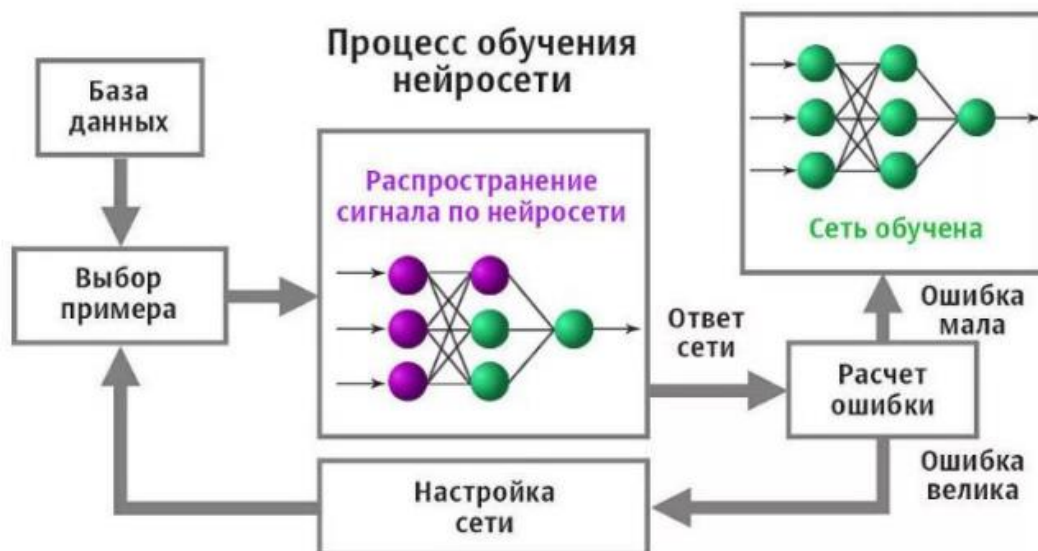


Рисунок 2.1 Обучение нейронной сети

Как можно обучать сеть? На рисунке 1.10 представлен примерный алгоритм обучения НС. Вспомним машинное обучение.

Итак, **обучение с учителем** (Supervised learning) – вид обучения сети, при котором ее веса меняются так, чтобы ответы сети минимально отличались от уже готовых правильных ответов.

Где взять правильные ответы? Если мы хотим, чтобы сеть узнавала лица, мы можем создать обучающую выборку на 1000 фотографий (входные сигналы) и самостоятельно выделить на ней лица (правильные ответы).

**Обучение без учителя** (Unsupervised learning) – вид обучения сети, при котором сеть самостоятельно классифицирует входные сигналы. Правильные (эталонные) выходные сигналы не демонстрируются.

Например, вы демонстрируете сети конфеты, пирожные и торты. Вы никак не регулируете работу сети. Вы просто подаете на ее входы данные об объекте. Со временем сеть начнет выдавать сигналы трех разных типов, которые и отвечают за объекты на входе.

Изучив структуру нейронных сетей и принципы их обучения можно перейти к практической реализации описанных методов.

## 2.2 Инструменты для создания модели

### *Язык программирования Python:*

— Плюсы:

- Простота использования, понятность и читаемость кода — проще, чем бейсик — идеально как для начинающих, так и для опытных разработчиков.
- Большое количество библиотек и фреймворков, которые предоставляют гибкие и мощные инструменты для разработки нейронных сетей.
- Активное сообщество разработчиков, которое обеспечивает поддержку, решение проблем и обмен опытом.

— Минусы:

- Python может быть несколько медленнее по сравнению с некоторыми другими языками программирования, что может быть проблемой при работе с большими объемами данных (Big Data).
- Некоторые библиотеки нейронных сетей могут быть менее стабильными или иметь ограниченную поддержку.
- Около 70-80% разработок в области нейронных сетей выполняется на Python. Это связано с его простотой, гибкостью и обширной экосистемой библиотек для машинного обучения, таких как TensorFlow, PyTorch, Keras.

### *Язык программирования C++:*

— Плюсы:

- Высокая производительность и эффективность — предпочтительный язык для разработки нейронных сетей с Big Data.
- Близость к «машинному» уровню, позволяющая оптимизировать код для конкретных аппаратных платформ.

— Минусы:

- Сложность использования и более сложный синтаксис по сравнению с Python.
- Менее развитое сообщество и меньшее количество готовых библиотек для работы с нейронными сетями.
- На нем выполняется 10-20% разработок в области нейронных сетей. Это язык низкого уровня, отсюда высокая производительность и эффективное использование ресурсов, особенно для больших и сложных моделей. Широко применяется в разработке фреймворков для нейронных сетей, таких как Caffe и TensorFlow.

### *Язык программирования Java:*

— Плюсы:

- Платформенная независимость — удобно для разработки нейронных сетей на разных операционных системах.
- Масштабируемость и надежность, способность работы с Big Data.
- Обширное сообщество разработчиков и богатая документация.

— Минусы:

- Более сложный синтаксис и более низкая производительность по сравнению с некоторыми другими языками.
- Малое количество специализированных библиотек и инструментов для разработки нейронных сетей. Среди имеющихся — библиотека DeepLearning4j.
- Примерно 5-10% разработок в области нейронных сетей выполняется на Java.

В апреле 2024 года Python продолжает занимать первую строчку (Рисунок 2.1) и увеличивать отрыв (его доля рынка выросла на 1.9% в сравнении с мартом 2024) в рейтинге самых популярных языков программирования по индексу TIOBE. Рейтинг составляет и ежемесячно обновляет компания TIOBE Software, главный офис которой находится в Нидерландах.

Место	Язык программирования	Доля рынка	Изменение доли рынка за прошедший месяц
1	 Python	16.41%	+1.90%
2	 C	10.21%	-4.20%
3	 C++	9.76%	-3.20%
4	 Java	8.94%	-4.29%
5	 C#	6.77%	-1.44%
6	 JavaScript	2.89%	+0.79%
7	 Go	1.85%	+0.57%
8	 Visual Basic	1.70%	-2.70%
9	 SQL	1.61%	-0.06%
10	 Fortran	1.47%	+0.88%

Рисунок 2.2 Рейтинг языков программирования на апрель 2024 года

Анализируя данную информацию языком программирования для построения модели нами был выбран Python. Но для более наглядного и простого



построения модели необходимо дополнительно установить необходимые библиотеки.

### ***Библиотека NumPy в Python:***

NumPy (сокращение от Numerical Python) — это фундаментальный пакет для высокопроизводительных научных вычислений и анализа данных в Python. Вот почему стоит использовать NumPy:

- **Высокопроизводительный объект многомерного массива (ndarray):** NumPy предоставляет эффективный объект многомерного массива, который позволяет быстро выполнять арифметические вычисления, ориентированные на массивы. Это особенно полезно при работе с данными в нейросетях.
- **Широкий спектр математических функций:** NumPy предлагает множество математических функций для выполнения операций с массивами, от базовой линейной алгебры до сложных преобразований Фурье. Эти функции позволяют эффективно работать с данными без необходимости явных циклов.
- **Интеграция с другими библиотеками:** NumPy хорошо взаимодействует с другими популярными библиотеками для научных вычислений, такими как SciPy, Pandas и Matplotlib. Это делает его незаменимым инструментом для анализа данных и разработки нейросетей.

### ***Библиотека Matplotlib в Python:***

Matplotlib — это библиотека для визуализации данных, которая позволяет создавать графики, диаграммы, гистограммы и другие визуальные элементы. Вот почему Matplotlib полезен:

- **Простота использования:** Matplotlib предоставляет простой и интуитивно понятный интерфейс для создания графиков. Это удобно при визуализации результатов нейросетей.
- **Гибкость и настраиваемость:** Вы можете настроить внешний вид графиков, добавить подписи к осям, легенды и многое другое. Это позволяет создавать информативные визуализации для презентаций и публикаций.
- **Интеграция с Jupyter Notebook:** Matplotlib хорошо работает в Jupyter Notebook, что упрощает визуализацию данных в процессе обучения нейросетей.

## 2.3 Построение модели

Адаптивный образовательный ресурс, который мы разрабатываем, представляет собой передовую систему, способную на автоматическое создание обширного спектра тестовых заданий, каждое из которых сопровождается множеством потенциальных вариантов ответов. Эта система уникальна тем, что она адаптируется к индивидуальным образовательным потребностям каждого учащегося.

В основе её работы должен быть сложный алгоритм, который анализирует предыдущие ответы студентов, чтобы определить их знания и навыки в определённой области. Затем, используя эту информацию, система генерирует новые вопросы, которые идеально подходят для дальнейшего развития учебных навыков учащегося.

Каждый вопрос тщательно подбирается и формулируется таким образом, чтобы не только оценить текущее понимание студентом материала, но и стимулировать критическое мышление и глубокое понимание предмета. Варианты ответов также разрабатываются с учётом различных уровней сложности, что позволяет студентам не только проверить свои знания, но и расширить их, выбирая более сложные ответы.

В целом, наш адаптивный ресурс будет представлять собой инновационное решение, которое обеспечивает персонализированный и эффективный подход к обучению, делая процесс образования более интерактивным, динамичным и мотивирующим. В качестве примера возьмем результаты теста, выполненного в системе дистанционного обучения Moodle.

### 2.3.1 Сбор данных

Экспорт результатов тестов в СДО Moodle может быть выполнен в различных форматах, каждый из которых имеет свои преимущества и недостатки.

#### ***В виде HTML-документа***

Плюсы:

- Удобство для чтения: HTML-формат легко читается в любом веб-браузере.
- Форматирование: Позволяет сохранять различные стили и форматирование текста.

Минусы:

- Не для обработки данных: Трудно использовать для дальнейшего анализа данных или обработки в других программах.
- Зависимость от браузера: Внешний вид может меняться в зависимости от используемого браузера.

#### ***В виде таблицы .xlsx***

Плюсы:

- Совместимость с Excel: Легко открывается и редактируется в Microsoft Excel.
- Функции анализа: Поддержка формул и инструментов для анализа данных.

Минусы:

- Необходимость наличия Excel: Для открытия файла требуется наличие программы Excel или совместимого программного обеспечения.
- Проблемы с совместимостью: Может возникнуть проблемы при открытии в других программах для работы с таблицами.

***В виде файла .csv***

Плюсы:

- Универсальность: Файлы CSV можно открывать в большинстве табличных редакторов.
- Простота: Простой формат, удобный для импорта и экспорта данных.

Минусы:

- Отсутствие форматирования: Не поддерживает сохранение стилей и форматирования.
- Ограниченное представление: Не подходит для сложного форматирования данных.

***В виде файла .txt***

Плюсы:

- Максимальная совместимость: Текстовые файлы можно открывать практически на любом устройстве.
- Простота редактирования: Легко редактировать в любом текстовом редакторе.

Минусы:

- Нет поддержки форматирования: Текстовые файлы не поддерживают форматирование.
- Трудности с организацией данных: Данные могут быть представлены неструктурированно, что затрудняет их анализ.

Эти форматы предоставляют различные возможности для работы с результатами тестов, и выбор конкретного формата зависит от целей и предпочтений пользователя. Например, для быстрого просмотра результатов подойдет HTML, а для глубокого анализа данных лучше использовать .xlsx или .csv.

Рассмотрим в первую очередь результаты в HTML и выберем оттуда вопросы, которые нам подойдут для наглядного примера. Но какие именно вопросы нам нужны? Если на вопрос все ответили правильно, то такой вопрос нам не подойдет, так как он слишком простой и нужной статистики для обучения мы не получим. То есть нам нужны вопросы, в которых есть те, кто совершил ошибки. Тут может быть несколько вариантов: либо вопрос задан некорректно, либо он слишком сложный.

СДО Moodle уже предлагает проанализированный тест (Рисунок 2.3).

Анализ структуры теста										
№	Тип вопроса	Название вопроса	Попытки	Индекс легкости	Стандартное отклонение	Балл случайного угадывания	Намеченный вес	Эффективный вес	Индекс дискриминации	Эффект
1	Случайный	Случайный (Вариант_01)	31	74.19%	44.48%		3,03%	3.64%	32.64%	38.60%
1.1	Множественный выбор	V_01_3д_1	5	80.00%	44.72%	20.00%	3,03%		85.57%	100.00%
1.2	Множественный выбор	V_03_3д_1	3	0.00%	0.00%	20.00%	3,03%			
1.3	Множественный выбор	V_04_3д_1	3	0.00%	0.00%	20.00%	3,03%			
1.4	Множественный выбор	V_05_3д_1	2	50.00%	70.71%	20.00%	3,03%		100.00%	100.00%
1.5	Множественный выбор	V_06_3д_1	3	100.00%	0.00%	20.00%	3,03%			
1.6	Множественный выбор	V_07_3д_1	3	100.00%	0.00%	20.00%	3,03%			
1.7	Множественный выбор	V_08_3д_1	2	100.00%	0.00%	20.00%	3,03%			
1.8	Множественный выбор	V_09_3д_1	5	100.00%	0.00%	20.00%	3,03%			
1.9	Множественный выбор	V_10_3д_1	5	100.00%	0.00%	20.00%	3,03%			
2	Случайный	Случайный (Вариант_02)	31	87.10%	34.08%		3,03%	2.62%	20.99%	28.68%
2.1	Множественный выбор	V_02_3д_2	4	50.00%	57.74%	20.00%	3,03%		-14.00%	-20.00%
2.2	Множественный выбор	V_03_3д_2	2	50.00%	70.71%	20.00%	3,03%		100.00%	100.00%
2.3	Множественный выбор	V_04_3д_2	2	100.00%	0.00%	20.00%	3,03%			

Рисунок 2.3 Анализ теста

Следуя нашим критериям, можно сделать вывод, что нас интересуют вопросы с индексом легкости менее 80%. Остальные 20% ошибок скорее всего будут случайными.

Примеры подходящих вопросов:

**Какое служебное слово используется в алгоритмической конструкции выбора из нескольких вариантов на языке C#?**

Модель ответа	Количество	Частота
do	0	0,00%
for	0	0,00%
if	5	55,56%
switch	4	44,44%
while	0	0,00%
Без ответа	0	0,00%

**С какого служебного слова в языке C# начинается цикл с постусловием?**

Модель ответа	Количество	Частота
while	0	0,00%
repeat	1	20,00%
for for	0	0,00%
do	3	60,00%
switch	1	20,00%
Без ответа	0	0,00%

**С какого служебного слова в языке C# начинается цикл с предусловием?**

Модель ответа	Количество	Частота
foreach	2	40,00%
for	0	0,00%
while	2	40,00%
repeat	1	20,00%
switch	0	0,00%
Без ответа	0	0,00%

### 2.3.2 Работа с csv-файлами

Когда мы сталкиваемся с задачей анализа результатов тестирования, особенно если речь идет о большом объеме данных, вручную обрабатывать каждый вопрос и ответ становится не только времязатратным, но и подверженным ошибкам процессом. В таких случаях, использование файла в формате CSV становится настоящим спасением.

Файлы CSV (Comma-Separated Values) представляют собой текстовые файлы, которые используют разделитель (чаще всего запятую) для структурирования табличных данных. Этот формат идеально подходит для работы с большими объемами информации, так как он позволяет быстро и эффективно обрабатывать данные с помощью программного обеспечения для работы с электронными таблицами или специализированных инструментов анализа данных.

Используя CSV-файл, мы можем легко импортировать результаты теста в такие программы, как Microsoft Excel или Google Sheets, где уже доступны мощные инструменты для сортировки, фильтрации и анализа данных. Это позволяет нам быстро находить закономерности, выявлять слабые и сильные стороны учащихся, а также адаптировать учебный материал под конкретные нужды группы или отдельного студента.

Более того, автоматизация процесса выбора вопросов с помощью CSV-файла значительно повышает точность анализа. Мы можем использовать различные алгоритмы для выявления наиболее частых ошибок или для определения вопросов, на которые студенты отвечают лучше всего. Это дает нам возможность не только оценить общий уровень знаний, но и тонко настроить образовательный процесс, делая его максимально эффективным и целенаправленным.

Работа с CSV файлами в Python может быть выполнена с помощью встроенного модуля csv.

**Чтение CSV файла:**

```
import csv

# Открытие файла для чтения
with open('test.csv', 'r', encoding='utf-8') as file:
    reader = csv.reader(file)
    for row in reader:
        print(row)
```

Однако, объем информации в файле слишком много, большая часть которой нам не нужна. Для фильтрации и извлечения только нужных нам данных из большого CSV файла, можно использовать библиотеку Pandas, которая предоставляет мощные инструменты для работы с табличными данными.

---

```
import pandas as pd

# Загрузка CSV файла
df = pd.read_csv('test.csv')

# Выбор нужных столбцов
needed_columns = df[['Модель ответа', 'Количество']]

print(needed_columns)
```

### 2.3.3 Анализ данных

Пусть количество вопросов определяет количество столбцов. В нашем тесте, как чаще всего бывает, пять вариантов ответа с одним верным. В таком случае 1 – ответ выбран, 0 – ответ не выбран. Количество тестируемых определяет количество строк. Получим матрицу значений. В качестве примера возьмем один из показанных ранее вопросов.

*С какого служебного слова в языке C# начинается цикл с постусловием?*

Исходя из ответов построим матрицу:

	While	Repeat	For for	Do	Switch
Учащийся 1	0	1	0	0	0
Учащийся 2	0	0	0	1	0
Учащийся 3	0	0	0	1	0
Учащийся 4	0	0	0	1	0
Учащийся 5	0	0	0	0	?

Какой цифрой заканчивается последняя строка? (Рисунок 2.4)

Импортируем модуль NumPy (должен быть установлен пакет NumPy). Берем веса по смыслу или случайно (цифры последнего столбца). В результате обучения программа поймет, какие веса подходят.

Создаем двумерный массив (матрицу) `teachMatrix` с цифрами без последнего столбца. Создаем массив `teachMatrixAnswers`, состоящий из цифр последнего столбца, не забыв перевернуть его в вертикальное положение функцией `.T`

Программа будет решать задачу методом подбора циклом `for`.

```

import numpy
from numpy.random import rand
def Activation (x):
    return 1 / (1 + numpy.exp (-x))
teachMatrix = numpy.array ([[0, 1, 0, 0],
                             [0, 0, 0, 1],
                             [0, 0, 0, 1],
                             [0, 0, 0, 1],
                             [0, 0, 0, 0]])
teachMatrixAnswers = numpy.array ([[0, 0, 0, 0, 1]]).T
numpy.random.seed (1)
sinaps = numpy.random.random ((4, 1))
for i in range (30000):
    inputLayout = teachMatrix
    output = Activation (numpy.dot (inputLayout, sinaps))
    err = teachMatrixAnswers - output
    adj = numpy.dot (inputLayout.T, err * (output * (1 - output)))
    sinaps += adj
print (output)

```

Рисунок 2.4 Обучение нейронной сети

Происходит подбор и перераспределение весов. Выводится результат, например:

```

= RESTART: F:\E
[[0.00411685]
 [0.0023682 ]
 [0.0023682 ]
 [0.0023682 ]
 [0.5       ]]

```

Что говорит нам о том, что из 30000 раз значение 1 будет в последнем столбце с вероятностью 0,5.

Если мы возьмем для контраста 30 повторений, то вероятность значений будет иметь большую погрешность:

```

=====
[[0.18474058]
 [0.08792165]
 [0.08792165]
 [0.08792165]
 [0.5       ]]

```

Эти показатели дают нам возможность сказать какие ответы наиболее часто встречаются, какие ошибки статистически совершаются чаще и корректен или нет изначальный вопрос.

## ЗАКЛЮЧЕНИЕ

Проект по созданию адаптивного образовательного ресурса является перспективным и многообещающим начинанием в сфере образовательных технологий. Он открывает новые горизонты для индивидуализации учебного процесса, предоставляя возможность каждому учащемуся получать знания в соответствии с его уникальными способностями и потребностями.

Сбор и анализ данных, проведенные в рамках исследования, позволили выявить ключевые параметры для создания эффективного обучающего контента, который может адаптироваться не только к различным уровням сложности, но и к специфике разнообразных учебных дисциплин. Это открывает путь к разработке универсальных образовательных ресурсов, которые будут способствовать повышению качества образования на всех уровнях.

Будущий адаптивный образовательный ресурс будет интегрировать передовые методы машинного обучения и искусственного интеллекта, чтобы предложить учащимся персонализированный обучающий опыт. Использование таких технологий позволит системе анализировать успеваемость учащихся в реальном времени и адаптировать учебный материал, делая его более понятным и доступным.

В целом, данный проект имеет потенциал изменить образовательную практику, предлагая новый подход к обучению, который будет полностью адаптирован под индивидуальные особенности каждого учащегося. Это не только улучшит процесс обучения, но и сделает его более интересным и мотивирующим для учащихся, что, в свою очередь, может привести к повышению общего уровня образовательных достижений.



## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Кадырова, Г. Р. Интеллектуальные системы : учебное пособие / Г. Р. Кадырова. – Ульяновск : УлГТУ, 2017. – 113 с.
2. Искусственный интеллект в образовании [Электронный ресурс]. – Режим доступа: <https://www.unesco.org/ru/digital-education/artificial-intelligence>. – Дата доступа: 14.05.2024.
3. Кан, К.А. Нейронный сети. Эволюция. – М: SelfPub 2018. – 288 с.
4. Постолит, А. В. Основы искусственного интеллекта в примерах на Python / А. В. Постолит. – СПб. : БХВ-Петербург, 2021. – 448 с.
5. Измайлова, М. А Роль искусственного интеллекта в построении адаптивной образовательной среды [Электронный ресурс]. – Режим доступа: <https://www.mir-nayka.com/jour/article/view/1642>. – Дата доступа: 12.05.2024.
6. Коровникова, Н. А. Искусственный интеллект в современном образовательном пространстве: проблемы и перспективы [Электронный ресурс]. – Режим доступа: <https://cyberleninka.ru/article/n/iskusstvennyy-intellekt-v-sovremennom-obrazovatelnom-prostranstve-problemy-i-perspektivy>. – Дата доступа: 14.05.2024.
7. Рашид, Тарик Создаем нейронную сеть. : Пер. с англ. — СПб. : ООО “Альфа-книга”, 2017. — 272 с. : ил. — Парал. тит. англ.