Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт кибернетики
Направление подготовки – 09.03.01 «Информатика и вычислительная техника»
Кафедра вычислительной техники

БАКАЛАВРСКАЯ РАБОТА

Тема работы
Разработка и программная реализация нейросетевого алгоритма для принятия решений в интеллектуальной игре «Покер»

УДК 004.7.032.26:519.7

Студент

Группа	ФИО	Подпись	Дата
8В2А	Иванцов Владислав Васильевич		

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент каф. ВТ	Хаустов П.А.	-		

КОНСУЛЬТАНТЫ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент каф. МЕН	Николаенко В.С.	-		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент каф. ЭБЖ	Невский Е.С.	-		

ДОПУСТИТЬ К ЗАЩИТЕ:

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
ВТ	Марков Н.Г.	д.т.н., профессор		

Томск – 2016 г.

ЗАПЛАНИРОВАННЫЕ РЕЗУЛЬТАТЫ ПО ОСНОВНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЕ ПОДГОТОВКИ БАКАЛАВРОВ 09.03.01 «ИНФОРМАТИКА И ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА», ИК ТПУ, ПРОФИЛЬ «ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ, КОМПЛЕКСЫ, СИСТЕМЫ И СЕТИ»

Код результатов	Результат обучения (выпускник должен быть готов)
Профессиональные компетенции	
Р1	Применять базовые и специальные естественнонаучные и математические знания в области информатики и вычислительной техники, достаточные для комплексной инженерной деятельности.
Р2	Применять базовые и специальные знания в области современных информационных технологий для решения инженерных задач.
Р3	Ставить и решать задачи комплексного анализа, связанные с созданием аппаратно-программных средств информационных и автоматизированных систем, с использованием базовых и специальных знаний, современных аналитических методов и моделей.
Р4	Разрабатывать программные и аппаратные средства (системы, устройства, блоки, программы, базы данных и т. п.) в соответствии с техническим заданием и с использованием средств автоматизации проектирования.
Р5	Проводить теоретические и экспериментальные исследования, включающие поиск и изучение необходимой научно-технической информации, математическое моделирование, проведение эксперимента, анализ и интерпретация полученных данных, в области создания аппаратных и программных средств информационных и автоматизированных систем.
Р6	Внедрять, эксплуатировать и обслуживать современные программно-аппаратные комплексы, обеспечивать их высокую эффективность, соблюдать правила охраны здоровья, безопасность труда, выполнять требования по защите окружающей среды.
Универсальные компетенции	
Р7	Использовать базовые и специальные знания в области проектного менеджмента для ведения комплексной инженерной деятельности.
Р8	Владеть иностранным языком на уровне, позволяющем работать в иноязычной среде, разрабатывать документацию, презентовать и защищать результаты комплексной инженерной деятельности.
Р9	Эффективно работать индивидуально и в качестве члена группы, состоящей из специалистов различных направлений и квалификаций, демонстрировать ответственность за результаты работы и готовность следовать корпоративной культуре организации.
Р10	Демонстрировать знания правовых, социальных, экономических и культурных аспектов комплексной инженерной деятельности.
Р11	Демонстрировать способность к самостоятельной к самостоятельному обучению в течение всей жизни и непрерывному самосовершенствованию в инженерной профессии.

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Институт Кибернетики
Направление подготовки – 09.03.01 «Информатика и вычислительная техника»
Кафедра Вычислительной техники

УТВЕРЖДАЮ:
Зав. кафедрой
_____ _____ Марков Н.Г.
(Подпись) (Дата) (Ф.И.О.)

ЗАДАНИЕ
на выполнение выпускной квалификационной работы

В форме:

Бакалаврской работы

Студенту:

Группа	**ФИО**
8В2А	Иванцову Владиславу Васильевичу

Тема работы:

Разработка и программная реализация нейросетевого алгоритма для принятия решений в интеллектуальной игре «Покер»	
Утверждена приказом директора (дата, номер)	От 19.02.2016 №1319/с

Срок сдачи студентом выполненной работы:	10.06.2016

ТЕХНИЧЕСКОЕ ЗАДАНИЕ:

Исходные данные к работе	Разработка и исследование искусственного игрового интеллекта, основанного на работе искусственной нейронной сети, обучаемой с помощью генетического алгоритма. Внедрение его в многопользовательское игровое клиент-серверное приложение.
Перечень подлежащих исследованию, проектированию и разработке вопросов	Выбор языка реализации приложения, параметров искусственной нейронной сети и генетического алгоритма, проектирование клиентского и серверного приложений. Исследование полученных результатов и оценка эффективности применённых технологий.
Перечень графического материала	UML-диаграммы, графики прогресса обучения исследуемых вариантов интеллектов.

Консультанты по разделам выпускной квалификационной работы	
Раздел	**Консультант**
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Николаенко Валентин Сергеевич
Социальная ответственность	Невский Егор Сергеевич

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	10.09.2015

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент кафедры ВТ	Хаустов Павел Александрович	-		10.09.2015

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8В2А	Иванцов Владислав Васильевич		10.09.2015

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Институт Кибернетики
Направление подготовки – 09.03.01 «Информатика и вычислительная техника»
Уровень образования – Бакалавриат
Кафедра Вычислительной техники
Период выполнения осенний / весенний семестр 2015/2016 учебного года

Форма представления работы:

Бакалаврская работа

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН
выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:	10.06.2016

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
10.09.2015	Составление и утверждение ТЗ	5
12.10.2015	Изучение существующих концепций покерных интеллектов	5
05.11.2015	Изучение теоретического материала по теории генетических алгоритмов и искусственных нейронных сетей	5
01.12.2015	Проектирование клиент-серверного приложения	5
10.02.2016	Разработка клиент-серверного приложения	25
01.03.2016	Разработка инструментов обучения и тестирования нейросетевых интеллектов	20
30.04.2016	Обучение и тестирование ИИ	25
20.05.2016	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	5
1.06.2016	Социальная ответственность	5

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Ассистент кафедры ВТ	Хаустов Павел Александрович	-		10.09.2015

СОГЛАСОВАНО:

Зав. кафедрой	ФИО	Ученая степень, звание	Подпись	Дата
Вычислительной техники	Марков Николай Григорьевич	д.т.н.		10.09.2015

РЕФЕРАТ

Выпускная квалификационная работа содержит 79 страниц, 18 рисунков, 4 таблицы, 19 источников.

Ключевые слова: игровой искусственный интеллект, искусственные нейронные сети, генетический алгоритм.

Объектом исследования является задача разработки игрового искусственного интеллекта.

Цель работы – разработка и исследование искусственного игрового интеллекта, с внедрением его в многопользовательское игровое клиент-серверное приложение.

В ходе выполнения работы было реализовано многопользовательское игровое клиент-серверное приложение. Кроме того, были реализованы инструменты, необходимые для обучения и тестирования нейросетевых интеллектов.

В результате исследования были получены различные варианты игровых искусственных интеллектов, эффективность применения которых была оценена на практике.

Область применения: интеллектуальные системы, методы машинного обучения, моделирование поведения сложных систем.

Значимость работы заключается в инновационном подходе к решению одной из наиболее популярных задач в IT сфере – создание искусственного интеллекта, моделирующего поведение человека.

Полученные результаты позволили определить наиболее перспективные направления для дальнейших исследований. В будущем планируется некоторым образом изменить полученную концепцию искусственного интеллекта, чтобы устранить слабые места текущего решения.

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ И НОРМАТИВНЫЕ ССЫЛКИ

Искусственные нейронные сети – математическая модель, а также её программное или аппаратное воплощение, построенная по принципу организации и функционирования биологических нейронных сетей – сетей нервных клеток живого организма [1].

Формальный нейрон – узел искусственной нейронной сети, являющийся математической моделью естественного нейрона [2].

Функция активации – функция, вычисляющая выходной сигнал искусственного нейрона.

Топология искусственной нейронной сети – организация связей между формальными нейронами в искусственной нейронной сети.

Генетический алгоритм – эвристический алгоритм поиска, основывающийся на моделировании процесса естественного отбора и эволюции в природе [3].

Особь – центральное понятие генетического алгоритма, представляет собой одно из возможных решений поставленной задачи, соответственным образом закодированное.

Ген – неделимая единица информации, носителей которой является особь. Совокупность генов называется **генотипом** особи.

Популяция – множество особей. Смежное понятие с термином «Поколение», но последнее чаще всего упоминается в контексте сменяющих друг друга популяций в процессе итеративного поиска.

Функция приспособленности – функция, отображающая из множества особей во множество действительных (целых) чисел. Полученное в результате отображения число называется значением функции приспособленности и характеризует удаленность данной особи от идеального решения по некоторой метрике.

Селекция – процесс оценивания популяции и отбора согласно некоторой стратегии тех особей, которые будут допущены до скрещивания.

Оператор скрещивания (кроссинговера) – функция, которая принимает на вход две особи, называемые **родительскими**, а на выход выдает две т.н. **дочерние** особи, каждая из которых наследует часть генотипа родителей [4].

Оператор мутации – процесс внесения случайных изменений в генотип некоторой особи. Мутация необходима для расширения пространства поиска и для того, чтобы в популяции появлялись решения, которых не было изначально [5].

ОБЪЕКТ И МЕТОДЫ ИССЛЕДОВАНИЯ

В данной работе объектом исследования является задача создания искусственного игрового интеллекта. Предметом исследования являются алгоритмы, позволяющие решать поставленную задачу.

Также были использованы следующие методы исследования:

- Анализ и синтез – в работе были проанализирована структура задачи создания нейросетевых интеллектов, особенности эволюционного подхода в ее решении. Был синтезирован генетический алгоритм, осуществляющий обучение нейросетевого интеллекта;
- Идеализация – этот метод является одним из основных в концепции генетических алгоритмов и искусственных нейронных сетей, поскольку используются упрощенные и идеализированные модели реальных объектов и явлений;
- Аксиоматический метод – использование теории нейросетевых вычислений и теории эволюционных вычислений влечет за собой использование некоторого набора аксиом, т.е. утверждений, которые считаются истинными и не требуют доказательств;
- Эксперимент и измерение – в данном исследовании эти методы являются центральными, поскольку особенности нейросетевых интеллектов затрудняют их теоретический анализ, что приводит к необходимости проведения серии тестов для оценки качества полученного решения;
- Сравнение – данная работа предполагает получение некоторого множества решений, для оценки эффективности которых необходимо провести сравнение с ранее полученными вариантами искусственных интеллектов.

Содержание

ВВЕДЕНИЕ

На сегодняшний день индустрия компьютерных игр является бурно развивающимся сектором экономики [6]. Разработкой игр занимаются как крупные компании мирового уровня, так и небольшие команды разработчиков, известные только узкому кругу пользователей. Некоторые из компьютерных игр являются одними из наиболее требовательных приложений на персональных компьютерах, для создания и воспроизведения которых используются самые передовые технологии. Другие же игры наоборот, не требуют особой производительности от платформы и способны воспроизводиться даже на смартфонах.

Среди всего разнообразия компьютерных игр особое место занимают многопользовательские онлайн игры. Благодаря использованию клиент-серверных технологий давно прошли те дни, когда люди нуждались в личных встречах с друзьями, чтобы играть в игры. Теперь можно играть когда угодно, и где угодно. Некоторые игры настолько просты, но в то же время увлекательны, что в них можно играть в обеденный перерыв, или по пути на работу. Многие из них созданы по мотивам известных настольных или азартных игр. Подобные игры популярны среди людей, которые хотят отвлечься от повседневной суеты или провести своё свободное время в общении с друзьями.

Однако популярность многопользовательских компьютерных игр обуславливается не только живым общением с другими игроками, но ещё и тем, что человек непредсказуем в своём поведении, он не ограничен никакими рамками. В то время как игровые интеллекты, основанные на стандартных алгоритмах, зачастую предсказуемы и достаточно быстро надоедают. Однако существуют определённые подходы в создании игровых интеллектов, которые позволяют имитировать поведение человека. Например, использование искусственных нейронных сетей позволяет создать игровой интеллект, который в своём поведении практически неотличим от человека.

В настоящее время искусственные нейронные сети применяются повсеместно: начиная от систем распознавания речи до распознавания вторичной структуры белка, классификации различных видов рака и генной инженерии [7]. Развитие данной области программирования позволяет как можно ближе приблизиться к созданию полноценного искусственного интеллекта, способного заменить человека во многих областях его деятельности.

Не на последнем месте стоит применение ИНС в покерных интеллектах, над созданием которых бьются исследовательские группы из многих университетов мира. Покер, который по своей сложности в поиске оптимального решения не уступает шахматам, является идеальной задачей для изучения искусственных нейронных сетей и методов их обучения.

Таким образом, в данной работе будут применены активно развивающиеся технологии программирования и исследованы актуальные проблемы на примере решения достаточно нетривиальной задачи. Всё это обеспечивает значительную актуальность данной работы.

1. ТЕОРЕТИЧЕСКИЕ ОСНОВЫ

1.1 Клиент-серверные технологии

1.1.1 Клиент-серверное взаимодействие

Главная концепция архитектуры «клиент-сервер» заключается в разделении сетевого приложения на некоторое количество компонентов, каждый из которых реализует определённый набор сервисов. Эти компоненты могут выполняться на разных компьютерах, при этом выполняя серверные либо клиентские функции.

В концепции клиент-серверной архитектуры выделяют три основные задачи:

1. функции ввода и отображения данных (обеспечивают взаимодействие с пользователем);
2. прикладные функции, характерные для данной предметной области;
3. функции управления ресурсами (файловой системой, базой данных и т.д.).

За решение каждой задачи отвечает соответствующий компонент клиент-серверного приложения. Выделяют два основных вида клиент-серверных архитектур. Наиболее распространённой является двухзвенная архитектура. Двухзвенной она называется из-за необходимости распределения трех базовых компонентов между двумя узлами (клиентом и сервером) [8].

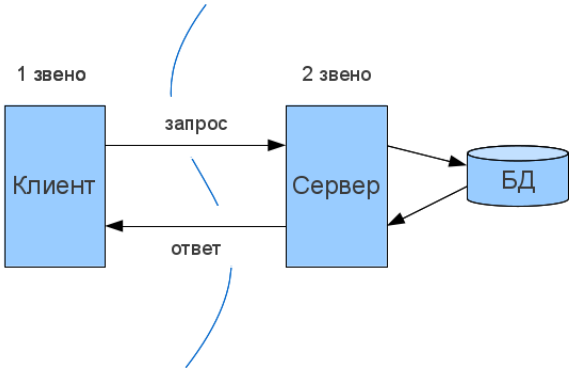


Рисунок 1.1. Двухзвенная клиент-серверная архитектура

Расположение компонентов на стороне клиента или сервера определяет основные модели их взаимодействия в рамках двухзвенной архитектуры:

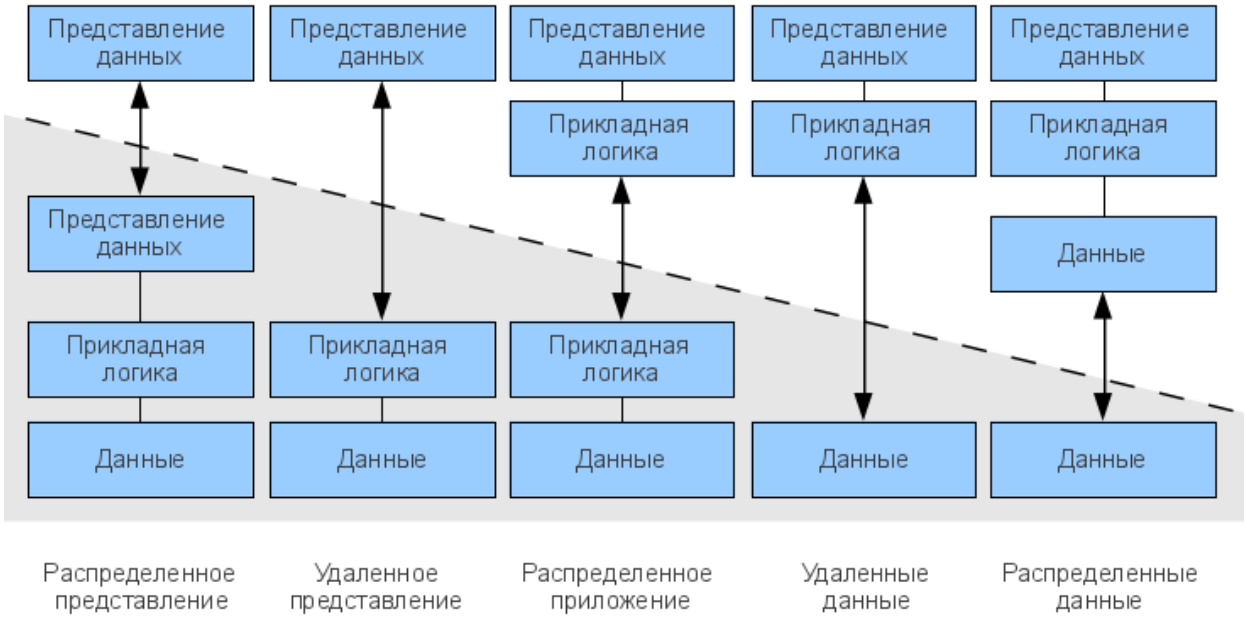


Рисунок 1.2. Модели клиент-серверного взаимодействия

В основе трёхзвенной архитектуры лежит идея распределённых вычислений. Они реализуются на основе модели сервера приложений, где сетевое приложение разделено на две или более частей, каждая из которых может выполняться на отдельном компьютере [9]. Выделенные части приложения взаимодействуют друг с другом, обмениваясь сообщениями в заранее согласованном формате:

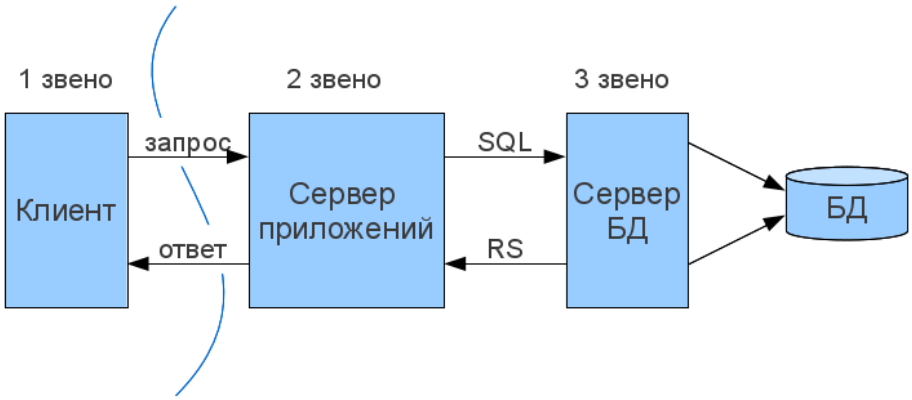


Рисунок 1.3. Трёхзвенная клиент-серверная архитектура

Как правило, третьим звеном в трехзвенной архитектуре становится сервер приложений, то есть компоненты распределяются следующим образом:

1. представление данных — на стороне клиента;
2. прикладной компонент — на выделенном сервере приложений (как вариант, выполняющем функции промежуточного ПО);
3. управление ресурсами — на сервере БД, который и представляет запрашиваемые данные.

Двухзвенная архитектура проще, так как все запросы обслуживаются одним сервером, но именно из-за этого она менее надежна и предъявляет повышенные требования к производительности сервера.

Трехзвенная архитектура сложнее, но благодаря тому, что функции распределены между серверами второго и третьего уровня, эта архитектура представляет:

- высокую степень гибкости и масштабируемости;
- высокую безопасность (т.к. защиту можно определить для каждого сервиса или уровня);
- высокую производительность (т.к. задачи распределены между серверами).

Однако, в рамках данной работы, наиболее предпочтительной является двухзвенная клиент-серверная архитектура. Это обуславливается тем, что серверная часть игрового приложения достаточно простая и не требует больших вычислительных мощностей. Следовательно, нет необходимости в распределении сервисов, реализуемых серверным приложением, на несколько компьютеров (серверов). Модель клиент-серверного взаимодействия в данном проекте соответствует модели «удалённого представления».

1.2 Выбор и обоснование выбора инструментов разработки

1.2.1 Обоснование выбора языка и среды разработки

Выбор для разработки в качестве графического движка Unity 2D некоторым образом ограничивает доступные языки программирования для реализации клиентского приложения, так как Unity позволяет писать код на языках JavaScript, C# и Boo [10]. Из предложенного списка языков в качестве языка разработки был принят C#, так как данный язык обладает рядом преимуществ.

В первую очередь, C# – объектно-ориентированный язык программирования. Данный язык очень удобен и прост, что достигается за счёт сокрытия от разработчика некоторых незначительных технических деталей (операции по упаковке/распаковке типов, инициализации переменных, сборке мусора и др.) [11]. Это позволяет полноценно сконцентрироваться на содержательной части задачи.

Кроме того, язык C# является языком, ориентированным на разработку для платформы .NET. Данная платформа обеспечивает разработчика необходимым набором инструментов для полноценной работы с сетевыми сокетами (различные классы в пространстве имён System.Net.Sockets), что несомненно является плюсом при разработке клиент-серверного приложения.

Таким образом, в качестве языка разработки для клиентского и серверного приложений был принят объектно-ориентированный язык C#. Средой разработки для клиентского приложения является студия Unity 2D. В качестве среды разработки для серверного приложения – Microsoft Visual Studio 2010.

1.3 Проблема создания искусственного игрового интеллекта

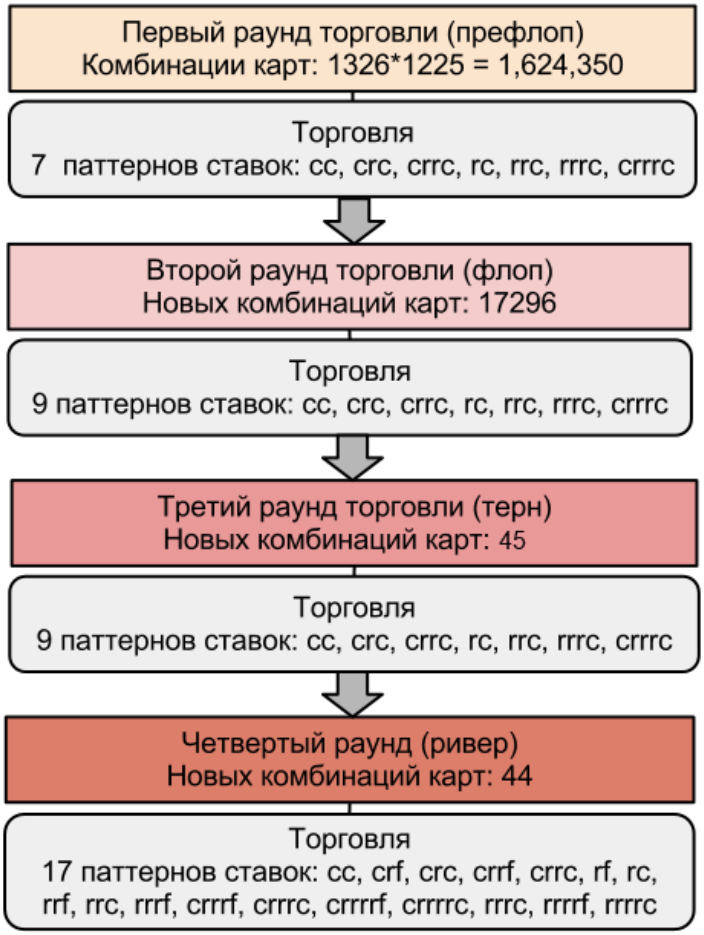
1.3.1 Искусственный интеллект в покере

Несмотря на то, что задача разработки игрового покерного интеллекта относится больше к игровой индустрии, а сам покер является азартной игрой, над решением данной задачи трудятся группы исследователей самых различных мастей от любителей до серьезных ученых. И хотя работы в данной области продолжаются уже не первый год, до сих пор не существует оптимального алгоритма, способного превзойти возможности человека, за исключением одного из частных случаев лимитированного покера (для двух игроков) [12].

Существует множество разновидностей покера, но в данной работе рассматривается наиболее популярная его версия на сегодняшний день: Техасский Холдем. Данный вид покера также разделяется на два подвида: лимитированный Холдем и безлимитный Холдем. Отличаются эти виды системой ставок: в лимитированном покере ставка ограничена некоторым пределом (как правило, вычисляется из большого блайнда), в безлимитном покере ставка ограничивается только банкроллом игрока (количество денег на руках). В лимитированном покере под ограничение попадает и количество поднятий ставки на один круг прикупа (как правило, не более 2-3). В безлимитном покере данное ограничение отсутствует. И хотя, казалось бы, различи не столь существенные, они значительно влияют на сложность поиска выигрышной стратегии.

Техасский Холдем примечателен тем, что он является игрой с неполной информацией – каждый знает только свои карты и карты на столе (хотя все карты на столе открываются только на последнем круге прикупа). Также в нем присутствуют элементы случайности – карты игроков и на доске выметаются из колоды случайным образом. К этим двум аспектам игры в покер добавляется еще и третий – количество игровых состояний, которое в безлимитном Холдеме существенно выше, чем в лимитированном.

Для того чтобы оценить масштаб проблемы, стоящей перед разработчиком покерного игрового интеллекта, рассмотрим структуру наиболее простой игры, которая проходит по правила лимитированного покера между двумя игроками. На рисунке 1.4 приведено изображение, показывающее структуру рассматриваемой игры с указанием количества новой информации, появляющейся на каждом круге прикупа:



Схематическое представление игровых состояний Техасского Холдема. В обозначениях к паттернам ставок использованы следующие обозначения: 'c' - колл или чек; 'r'- бет, рейз, перейз и т.д.; 'f' - фолд, сброс.

Рисунок 1.4. Структура игры с указанием новой информации

В рассматриваемом варианте Техасского Холдема содержится порядка 10^{18} игровых ситуаций, что делает данную игру практически не решаемой «в лоб», путём полного перебора всех возможных вариантов развития событий. Однако следует заметить, что некоторые карточные комбинации эквивалентны между собой. Например, в рассмотренном выше случае, можно выделить эквивалентные комбинации: не первом круге прикупа (префлоп)

комбинация из двух тузов будет являться идентичной любой другой комбинации из двух тузов. Выделив эквивалентные комбинации на всех этапах игры, можно снизить количество игровых ситуаций до 10^{14}. Что, конечно же, упрощает вычисления, но не столь значительно, чтобы решать задачу путём использования полного перебора. Также следует помнить, что рассматриваемый вариант игры самый простой, и что элементарное увеличение числа игроков за столом значительно усложнит поиск решения [13].

В данном случае, когда исходная игра слишком велика для непосредственного изучения, использует метод абстрагирования – переход к абстрактной игре, объединяющей в своих игровых состояниях близкие в стратегическом смысле состояния полной игры. Другими словами, при анализе и решении игры некоторые состояния исходной игры трактуются как неразличимые. Таким образом, чем более абстрактной получится игра, тем проще будет найти её решение, но у этого подхода есть и отрицательное свойство. Чем сильнее абстракция, тем больше информации теряется об исходной игре. И чем существенней данная информация, тем более уязвимым становится покерный интеллект.

1.3.2 Искусственные нейронные сети

1.3.2.1 Формальный нейрон

Основной составной и функциональной частью нейронной сети является формальный нейрон (искусственный нейрон):

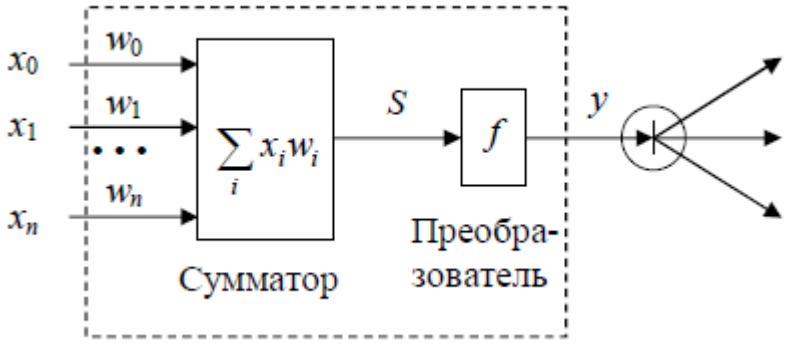


Рисунок 1.5. Формальный нейрон

На рисунке x_0, x_1,…x_n – компоненты вектора входных сигналов, w_0, w_1,…w_n – значения весов входных сигналов нейрона, а y – выходной сигнал нейрона. Формальный нейрон, в свою очередь, состоит из трёх частей: умножитель, сумматор и преобразователь. Умножитель умножает входной сигнал на соответствующий вес, таким образом, определяя силу связи между нейронами. Сумматор выполняет сложение входных сигналов, подготавливая данные для преобразования. Преобразователь реализует функцию одного аргумента – выхода сумматора. Эта функция называется функцией активации или передаточной функцией нейрона [14].

Математическая модель формального нейрона выглядит следующим образом:

$$y = f(S),$$

$$S = \sum_{i=1}^{n} w_i x_i + b.$$

1.3.2.2 Нейронные сети

Формальные нейроны, объединённые таким образом, чтобы выходы одних нейронов служили входами для других нейронов, формируют искусственную нейронную сеть.

Внутри нейронной сети отдельные нейроны разделяются на три типа:

- входные нейроны, на которые подаются входные для всей сети сигналы. Такие нейроны имеют, как правило, один вход с единичным весом, смещение отсутствует, а значение выхода нейрона равно входному сигналу;
- выходные нейроны, выходные значения которых представляют результирующие выходные сигналы нейронной сети;
- скрытые нейроны, не имеющие прямых связей с входными сигналами, при этом значения выходных сигналов скрытых нейронов не являются выходными сигналами ИНС [15].

По структуре межнейронных связей различают два основных класса ИНС:

1. ИНС прямого распространения, в которых сигнал распространяется только от входных нейронов к выходным. Орграф, соответствующий таким ИНС, не имеет циклов и петель. Примером ИНС прямого распространения является ИНС на рисунке 1.6а.
2. Рекуррентные ИНС – сети с обратными связями. В таких ИНС сигналы могут передаваться между любыми нейронами, вне зависимости от их расположения в ИНС. Орграф, соответствующий структуре рекуррентных ИНС, может иметь петли и циклы (рисунок 1.6б).

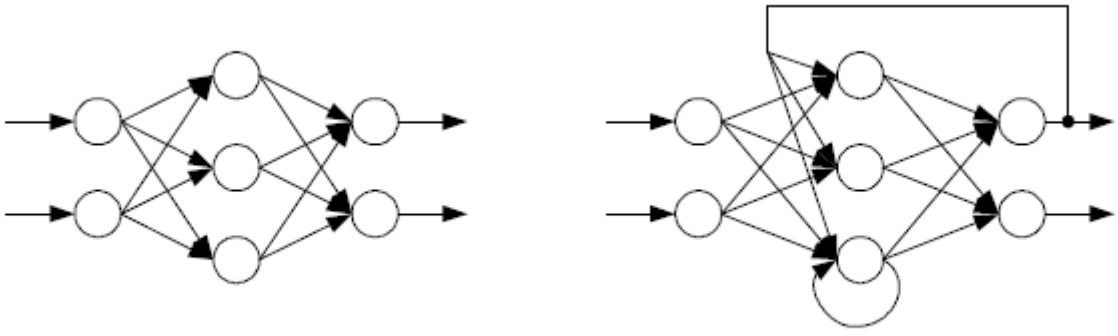


Рисунок 1.6. Примеры структур нейронных сетей: а) ИНС прямого распространения; б) рекуррентная ИНС

Внутри указанных классов искусственные нейронные сети также делятся на подклассы, в зависимости от особенностей строения. На рисунке 1.7 приведена систематизация архитектур сетей:

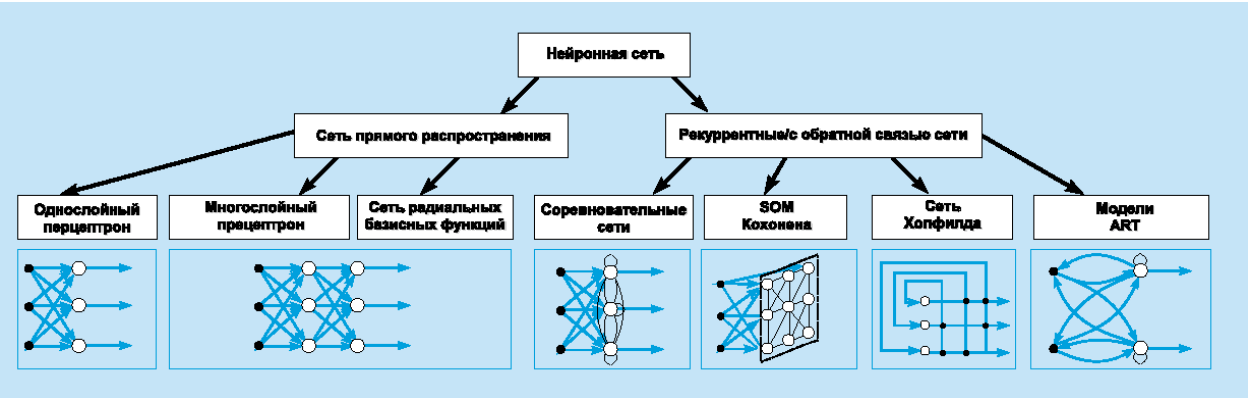


Рисунок 1.7. Архитектуры нейронных сетей

Однако, несмотря на многообразие структур нейронных сетей, нет единого мнения, какие структуры для решения каких задач подходят наилучшим образом. Большинство видов структур было получено благодаря более пристальному изучению биологических нейронных сетей, целью которого было выявить особенности строения, дающие биологическим нейронным сетям такие возможности.

1.3.2.3 Обучение искусственных нейронных сетей

Эффективность применения ИНС во многом зависит от правильности выбора её структуры и качества обучения. Различают три основные парадигмы обучения нейронных сетей: обучение «с учителем», обучение «без учителя» и смешанный подход.

Обучение «с учителем» подразумевает под собой использование заранее сформированной обучающей выборки: набора входных и выходных векторов, эталонных примеров. При этом обучение заключается в такой настройке весов сети, чтобы при определённом входном векторе на выходе сети получался соответствующий выходной эталонный вектор, либо вектор, соответствующий эталонному с допустимой погрешностью. Пары эталонных векторов могут подаваться многократно, при этом эпохой обучения будет считаться один полный «прогон» всех векторов. Проверка работоспособности сети осуществляется на наборах, не участвующих в обучении.

При обучении сети «с учителем» может возникнуть проблема «переобучения». Исходные наборы могут содержать некоторые неточности, которые могут передаться ИНС. Поэтому при обучении стараются получить результат с некоторой погрешностью, равной примерно 0,01 – 0,001.

Известны четыре основных типа правил обучения: метод обратного распространения ошибки, машина Больцмана, правило Хебба и обучение методом соревнования [16]. В силу того, что в данной работе будет использоваться принцип обучения «без учителя», данные методы не будут рассмотрены подробно.

Обучение «без учителя» подразумевает отсутствие обучающей выборки. В этом случае раскрывается внутренняя структура обрабатываемых данных или корреляции между образцами в системе данных, что позволяет распределить образцы по категориям [17]. То есть, обучение сети производится на её собственном опыте.

При смешанном обучении часть весов определяется посредством обучения с учителем, в то время как остальная получается с помощью самообучения.

В рамках данной работы было принято решение обучать ИНС посредством использования метода обучения «без учителя». Однако когда речь заходит о такой нетривиальной игре, как покер, задача исследования всех возможных наборов данных и выявление между ними корреляции становится практически невыполнимой. Поэтому наиболее приемлемым вариантом стала настройка весов посредством генетического алгоритма, который позволил бы сгенерировать наиболее «жизнеспособную» комбинацию весов.

У этого подхода есть ряд положительных сторон. Во-первых, нет необходимости в формировании обучающих выборок. Во-вторых, обучение подобным образом позволяет создать интеллект, стратегия игры которого не будет повторять ни одну из существующих стратегий. Тогда как при обучении «с учителем» ИНС с некоторой точностью повторяла бы стратегию «учителя». То есть, подобный подход к обучению ИНС наиболее близок к обучению человека и имеет практически неограниченный потенциал.

Таким образом, задача построения эффективного покерного интеллекта будет заключаться в подборе наиболее оптимальной структуры ИНС и генерации наиболее «жизнеспособного» набора весов для неё.

2. ПРОЕКТИРОВАНИЕ СЕРВЕРНОГО ПРИЛОЖЕНИЯ

2.1 Устройство серверного приложения

Основной целью данной работы является разработка искусственного игрового интеллекта. Необходимо исследовать различные виды его реализации, и выявить наиболее эффективные способы его обучения. Для того, чтобы определить качество работы искусственного интеллекта, необходима игровая среда, которая обеспечит соблюдение всех игровых правил вне зависимости от того, как работает игровой интеллект. Решение этих задач обеспечивает серверное приложение, реализованное в ходе выполнения данной работы.

Как было отмечено в пункте «Клиент-серверное взаимодействие», распределение функционала между клиентом и сервером построено на основе модели «Удалённое представление». В данной модели вся основная работа по обработке данных и обеспечению функционала системы выполняется на стороне сервера. Клиент в таком случае отвечает только за представление данных.

При проектировании серверного приложения оно было разбито не несколько логических модулей, каждый из которых отвечает за определённые функции. При подобном подходе обеспечивается высокий параллелизм разработки: при работе в команде за каждый модуль назначается ответственный, который занимается программированием модуля независимо от остальных членов команды. Функциональная независимость модулей позволяет быстро и точно выявлять ошибки, и, в случае необходимости, полностью заменять модули без ущерба остальному приложению. Это качество также полезно при дальнейшем сопровождении продукта.

Серверное приложение состоит их четырёх основных модулей, каждый из которых содержит в себе несколько классов, реализующих функционал модуля:

1. модуль клиент-серверного взаимодействия;
2. модуль управления локальной базой данных;
3. модуль, реализующий взаимодействие внутри игровых комнат;
4. модуль, отвечающий за игровую логику.

На рисунке 2.1 представлена схема, отражающее примерное взаимодействие модулей внутри приложения:

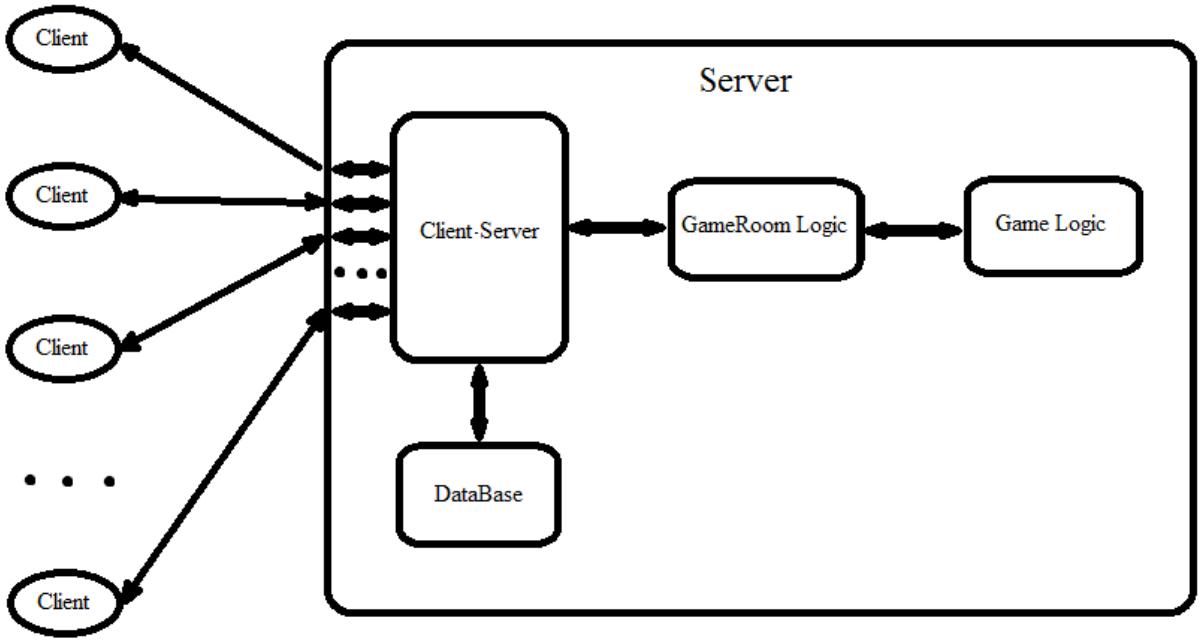


Рисунок 2.1. Схема взаимодействия модулей серверного приложения

По данной схеме видно, что все клиенты подключаются к серверу, но при этом не имеют связей непосредственно друг с другом. Это означает, что сервер является центром всей системы, который полностью обеспечивает всю обработку информации, и всё общение между клиентами обеспечивается именно сервером, что повышает надёжность клиент-серверного взаимодействия.

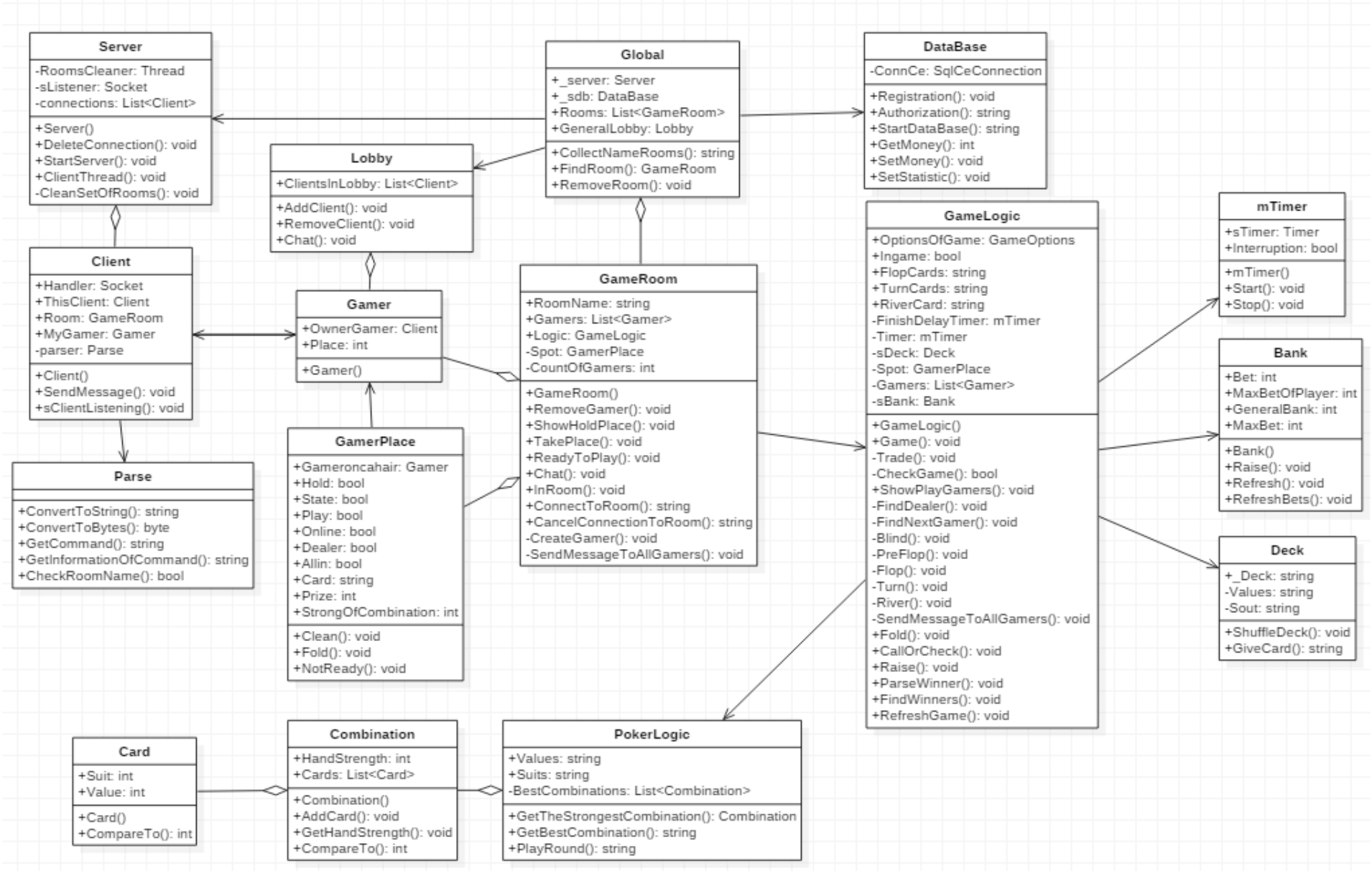


Рисунок 2.2. UML-диаграмма классов серверного приложения

2.2 Модуль клиент-серверного взаимодействия

Модуль клиент-серверного взаимодействия, как и следует из названия, обеспечивает обмен сообщениями, содержащие команды и данные, между клиентом и сервером. Для обмена данными между сервером и клиентом используются сокетные технологии, которые в платформе .NET Framework представлены классом System.NET.Sockets.Socket, который предоставляет низкоуровневый интерфейс для приема и отправки сообщений по сети. На основе данного класса были реализованы классы Server и Client, которые совместно с классом Parse составляют модуль клиент-серверного взаимодействия.

Класс Server является главным классом в модуле и представляет собой оболочку для серверного сокета. Во всём серверном приложении создаётся строго один экземпляр данного класса, который сразу после запуска программы переводит серверный сокет в режим ожидания входящих сообщений (функция StartServer()). Серверное приложение работает в многопоточном режиме, что позволяет одновременно обрабатывать множество подключений. В момент установления нового соединения объектом класса Server вызывается функция ClientThread(), которая создаёт новый объект класса Client и затем добавляет его в список активных соединений. После этого для нового соединения создаётся отдельный поток, и всё управление взаимодействием с клиентом переводится на него.

Вновь созданный поток уничтожается только после закрытия соединения с клиентом. За это отвечает функция DeleteConnection().

Помимо вышеперечисленных, в классе также реализована функция CleanSetOfRooms(), которая запускается в отдельном потоке при запуске приложения. Этот поток существует всё время работы сервера и отвечает за закрытие пустых игровых комнат.

Как уже было сказано, для каждого нового соединения заводится объект класса Client, который отвечает за приём и отправку сообщений для конкретного клиента.

Класс Client содержит две функции. Функция SendMessage() обеспечивает отправку сообщений. Функция sClientListening() содержит в себе бесконечный цикл, который запускается в отдельном потоке. Этот цикл отвечает за приём и обработку входящих сообщений с клиента.

Клиент и сервер обмениваются текстовыми сообщениями, которые имеют определённый формат. Каждое сообщение содержит в себе в качестве префикса определённую команду. Как клиент, так и сервер имеют определённый набор команд, который они могут корректно обработать. По сути, этот набор команд представляет собой интерфейс общения между приложениями. После префиксной команды, как правило, следуют данные, которые. Команда от данных отделяется специальным символом «|», который запрещён во всех остальных случаях. В случае если данные разделены на отдельные части, каждая часть отделяется от последующей таким же символом «|».

В зависимости от того, какие данные были получены, объект класса Client инициирует выполнение той или иной процедуры на сервере. Например, в случае прихода команды «authorization|qwerty|password|» будет инициирована процедура авторизации пользователя по указанному логину и паролю. Конечно же, не все команды доступны в каждый момент времени работы сервера. Набор доступных команд зависит от текущего состояния сессии. Например, после авторизации пользователя, повторная авторизация будет уже невозможна.

Класс Parse содержит в себе процедуры, предназначенные для конвертирования входящих и исходящих сообщений (из потока бит в последовательность символов Unicode и обратно) и вычленения команд и данных из сообщения.

Отдельного внимания требует класс Global. Этот класс содержит в себе статические структуры данных, область видимости которых распространена на весь проект. Кроме того, данный класс взял на себя часть функционала сервера по управлению игровыми комнатами.

На рисунке 2.3 представлена UML-диаграмма модуля клиент-серверного взаимодействия:

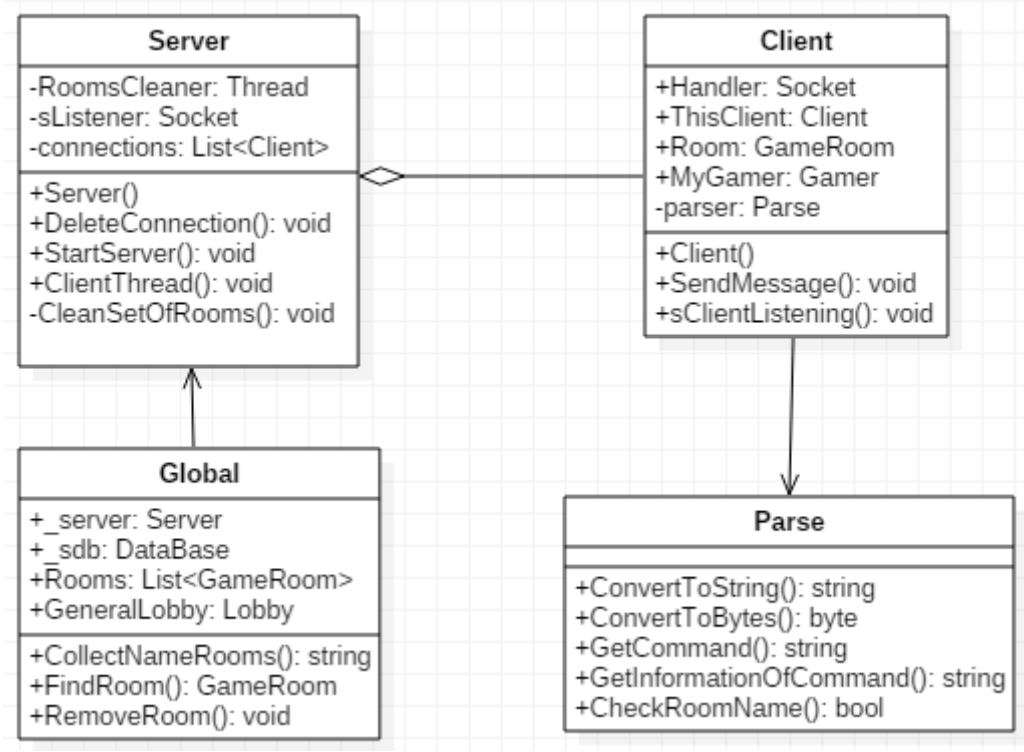


Рисунок 2.3. UML-диаграмма модуля клиент-серверного взаимодействия

2.3 Модуль управления локальной базой данных

Проект является многопользовательским приложением, поэтому возникает необходимость в хранении необходимых данных о пользователях (логин, пароль, электронная почта и т.д.). Для этого используется локальная база данных, которая хранит всю необходимую информацию. Сама база состоит из одной таблицы и хранится на жёстком диске в виде sdf-файла. Взаимодействие серверного приложения и базы данных обеспечивается с помощью класса SqlCeConnection, предоставляемого платформой .NET Framework. На основе этого класса был написан класс Database.

Класс Database содержит в себе методы, необходимые для управления базой данных в рамках данного приложения: Registration(), Authorization(), GetMoney(), SetMoney(), GetStatistic(), SetStatistic(). Как и в случае с классом Server, при запуске приложения создаётся строго один экземпляр класса Database.

Схема базы данных приведена ниже:

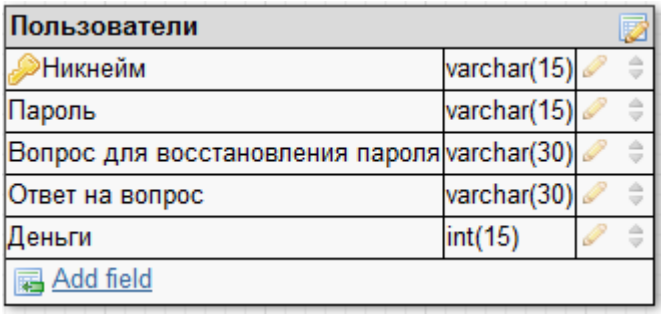


Рисунок 2.4. Схема базы данных

2.4 Модуль управления игровыми комнатами

После авторизации на сервере с каждым клиентом отожествляется один объект класса Gamer. То есть все клиенты на сервере представляются игроками. Свойства класса Gamer содержат информацию о конкретном игроке, которая используется для обеспечения взаимодействия с другими пользователями.

Однако, в подавляющем большинстве случаев, сервер взаимодействует не с конкретными игроками, а с игровыми комнатами. Игровая комната в данном приложении, по сути, представляет собой некоторую модель реальной игровой комнаты, в которой игроки взаимодействуют друг с другом: играют, обмениваются сообщениями. На сервере различаются два вида игровых комнат, которые реализованы классами Lobby и GameRoom.

На рисунке 2.5 представлена UML-диаграмма модуля игровых комнат:

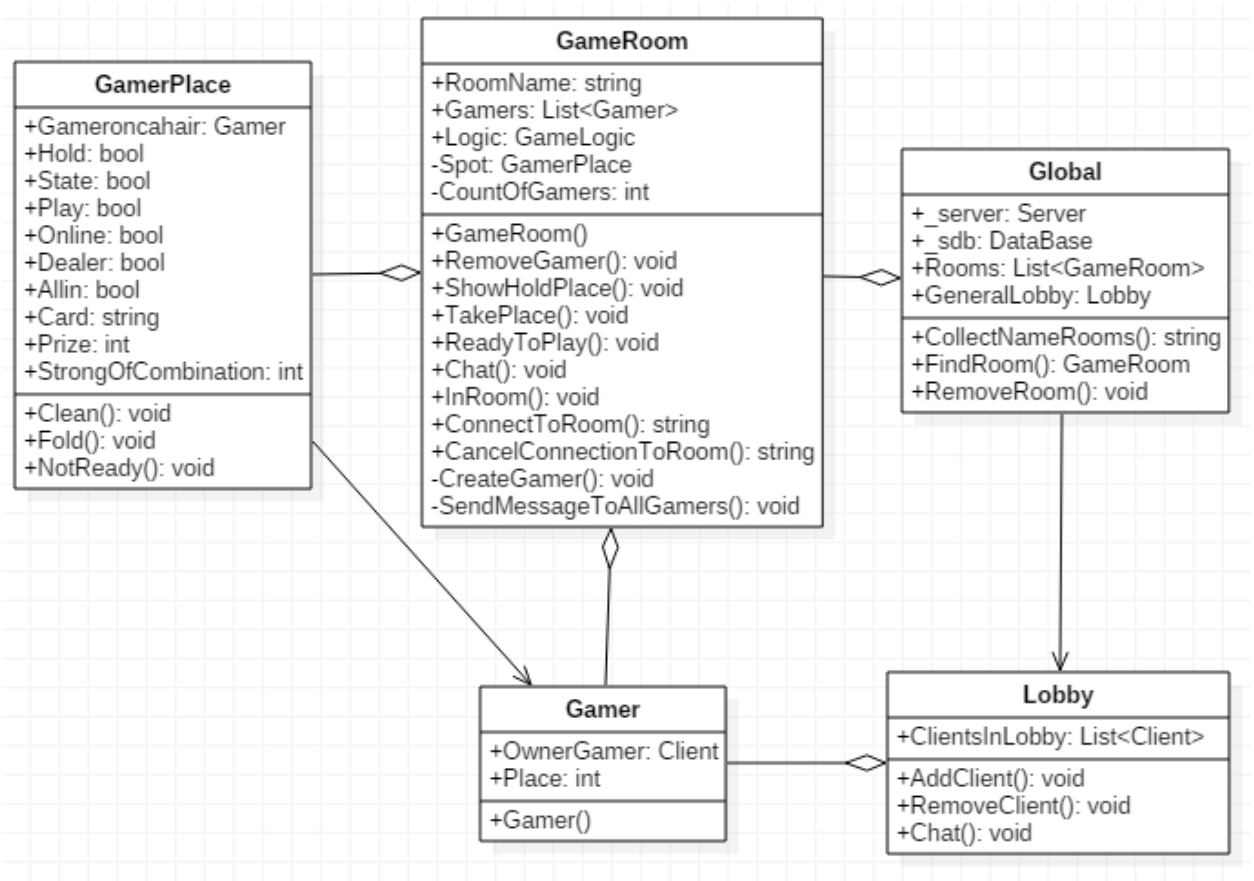


Рисунок 2.5. UML-диаграмма модуля управления игровыми комнатами

Класс Lobby представляет собой упрощённый вариант игровой комнаты. Если проводить параллели с реальным миром, то Lobby – это холл, в который гости попадают сразу после того, как вошли в игровое заведение. В холле люди могут общаться и выбирать игровые комнаты, в которые они переходят для последующей игры.

Класс GameRoom предоставляет более мощный функционал, так как он обеспечивает управление игровым процессом. При создании игровой комнаты указывается название комнаты и количество игровых мест за столом. Название комнаты должно быть уникальным, потому что оно идентифицирует комнату на сервере.

Функции ConnectToRoom() и CancelConnectionToRoom() класса GameRoom обеспечивают подключение к комнате и отключение от комнаты игроков. Метод InRoom() собирает информацию о текущем состоянии игры: количество карт на столе, ставки игроков, сумму в банке и т.д. Это необходимо для того, чтобы игроки могли заходить в комнату в любой момент игры в качестве наблюдателей.

Следует отметить, что количество игроков, одновременно подключенных к одной игровой комнате, ничем не ограничено. Однако за столом одновременно может сидеть ограниченное количество человек (настраиваемый параметр). Это объясняется тем, что количество мест за столом ограниченно. При этом в программе различаются сущность игрока и сущность места.

Сущность места описана в классе GamerPlace. В каждой комнате заводится коллекция объектов класса GamerPlace, содержащая в себе ровно столько объектов, сколько мест разрешено в комнате. Когда игрок занимает место, он привязывается к конкретному объекту из коллекции. После этого всё взаимодействие в игре производится непосредственно между местами, занятыми игроками, а не между самими игроками. Подобный подход позволяет динамически заменять игрока за место на искусственный интеллект, что крайне удобно в рамках данного проекта.

Разделение понятий игрока и игрового места также позволяет повысить комфортность игры для пользователей. В случае, если происходит непредвиденный разрыв соединения с игроком, игровое место максимально долго сохраняется за игроком, что даёт пользователю возможность повторно зайти в комнату и продолжить игру. По сути, отключенный игрок удаляется с места только тогда, когда ему необходимо принять решение о повышении ставки.

2.5 Модуль игровой логики

Модуль игровой логики является наиболее сложным и, по сути, является главным модулем всего приложения.

Основной класс – GameLogic. Он обеспечивает ход игры с соблюдением всех правил покера. Рассмотрим подробнее, как он устроен.

Объект класса GameLogic хранится в единственном экземпляре в каждой игровой комнате. При создании комнаты в отдельном потоке запускается главный игровой цикл. Однако, игра не начинается до тех пор, пока не создадутся определённые условия: за столом должны сидеть как минимум два игрока, сообщивших о готовности. После этого запускается обратный отсчёт, в течение которого игроки могут либо отказаться от игры, либо дождаться подключения к игре других пользователей. По завершению отсчёта начинается первый круг игры.

Игра Texas Hold’em имеет несколько кругов, каждый из которых имеет своё название. За каждый круг отвечает одноимённый метод из класса GameLogic: PreFlop(), Flop(), Turn() и River(). По сути, данные методы выполняют роль крупье за столом: они перемешивают карты в колоде и выкладывают их на стол в определённом порядке, согласно кругам игры. Следует отметить, что класс колоды Deck имитирует реальную карточную колоду: в нём содержится определённое количество элементов типа «карта», которые перемешиваются в начале игры, а затем поочерёдно достаются сверху стопки. Карты игрокам раздаются в самом начале игрового цикла.

На каждом круге прикупа, после того как на столе открывается очередная порция карт, наступает «торговля». Игроки в очерёдности, определённой правилами игры, делают свои ходы. Все ходы строго определены внутри класса GameLogic и соответствуют игровым правилам. Игрок может сбросить карты (метод Fold()), пропустить ход или поддержать ставку (метод CallOrCheck()), а также повысить ставку (метод Raise()). Круг прикупа длится до тех пор, пока все игроки за столом не уравняют ставки. На

принятие решения игроку отводится строго определённое время. Если за отведённое время игрок не сделает ход, сервер сделает это за него. При этом карты сбрасываются только в том случае, если игрок имел ставку, меньше максимальной ставки за столом.

После завершения последнего круга прикупа возникает необходимость в определении победителя. В некоторых случаях их может быть несколько, тогда весь банк будет разделён между выигравшими игроками. За определение победителей отвечает класс PokerLogic.

Как известно, в покере существует десять комбинаций, каждая из которых имеет свою силу. Комбинации в порядке возрастания силы: старшая карта, пара, две пары, тройка, стрит, флэш, фул-хаус, каре, стрит-флэш, роял-флэш.

Каждый из игроков имеет в распоряжении семь карт: две карты на руках и пять карт на столе. Из этих семи карт алгоритм выбирает пять таких, что они образуют комбинацию с наибольшей силой. Если наиболее сильную комбинацию можно собрать несколькими способами, то правилами покера оговариваются дополнительные критерии сравнения силы комбинаций. Таким образом, выбирается пять карт, которые образуют наилучшую возможную комбинацию игрока.

После этого комбинации всех игроков сравниваются с учетом силы комбинации и дополнительных критериев, игроки ранжируются по местам. Причем сразу несколько игроков могут поделить одно место. Все игроки, которые заняли первое место, делят банк поровну. Возможна ситуация, когда не все игроки поставили одинаковые ставки. В таком случае правилами покера оговорен принцип, по которому победители делят лишь часть денег, после чего деньги начинают распределяться между последующими местами.

Класс PokerLogic имеет метод PlayRound(), получающий на вход карты и ставки каждого и игроков, возвращающий силу комбинации каждого из игроков и количество денег, которое он получает в результате игры.

Для простоты и наглядности реализации используются класс карты Card и класс комбинации Combination, каждый из которых реализует интерфейс IComparable для возможности сортировки коллекций классов с такими объектами. Так, например, можно сортировать комбинации по силе или выбирать наиболее сильную комбинацию из набора, используя метод CompareTo().

Внутри класса PokerLogic также реализованы вспомогательные методы для определения силы комбинации, наиболее сильных комбинаций игроков, распределения выигрыша, которые инкапсулируют всю логику определения победителей и распределения выигрышей.

На рисунке 23 представлена схема взаимодействия классов внутри модуля игровой логики:

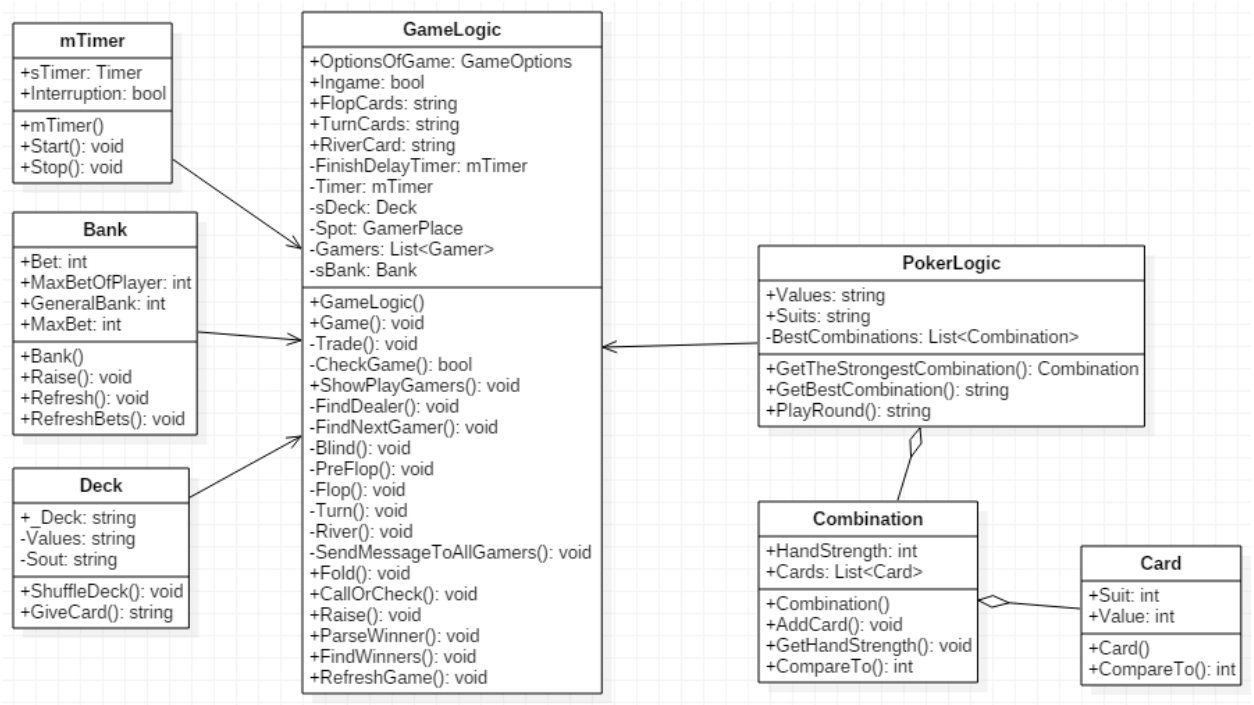


Рисунок 2.6. UML-диаграмма модуля игровой логики

3. ОПИСАНИЕ ПРЕДЛАГАЕМЫХ РЕШЕНИЙ

3.1 Определение набора входных данных

Главной целью данной работы является разработка игрового интеллекта, основанного на работе искусственных нейронных сетей. Интеллект должен принимать решение о текущем ходе игрока, исходя из информации, предоставленной ему игровой средой.

Однако, некоторая информация, которая может быть получена в игровой среде, может быть избыточной, что негативно скажется на скорости обучения и качестве работы нейронной сети, поэтому возникает необходимость в определении наименьшего набора исчерпывающей информации о состоянии игры.

Каждый игрок при игре в покер обладает следующей информацией: карты на руках, карты на столе, ставки игроков, количество денег на руках у каждого игрок, банк, а также расположение игроков за столом. Учитывая, что максимальное количество игроков за столом равно шести, то подобный набор информации является явно избыточным.

В первую очередь, следует убрать из набора входных данных информацию о картах. Вместо карт, комбинации которых достаточно сложно закодировать оптимальным образом для использования в ИНС, гораздо удобней и эффективней использовать вероятность выигрыша при известных картах на руках и на столе.

Ранее в теоретическом обосновании данной работы говорилось о сложности создания покерных интеллектов: основная сложность заключается в большом количестве комбинаций карты и игровых состояний. Большое количество карточных комбинаций затрудняет процесс вычисления вероятности выигрыша той или иной комбинации. Однако в данном решении использовался достаточно известный численный метод, известный как метод Монте-Карло.

Данный метод предполагает получение большого числа реализаций стохастического процесса, который формируется таким образом, чтобы его вероятностные характеристики совпадали с аналогичными величинами решаемой задачи.

В данном случае случайной величиной является выигрыш или поражение игрока при известном наборе карт. При этом все карты, которые также участвуют в эксперименте, но их значения достоверно не известны, выбираются из оставшейся колоды случайным образом. Из общего числа экспериментов запоминается число тех случаев, когда игрок победил. Это число делится на общее число проведённых тестов, что в результате даёт вероятность выигрыша.

Ставки игроков и количество денег являются существенными данными. Исходя из этих данных, можно выработать оптимальную стратегию: например, можно попробовать задавить ставкой более слабого игрока, или же остеречься сильного игрока в случае, если он начнёт повышать ставку. При этом сам банк на столе практически не играет никакой роли, поэтому им можно пренебречь. Несущественной информацией является и расположение игроков за столом. Однако, не зная, кто и где сидит, как определить, сколько игроков находится в данный момент за столом, и сколько из них в игре.

За свободные места либо места игроков, которые не участвуют в игре, на вход нейронной сети подаются нули. Так влияние этих входов полностью нивелируется. При этом ставка и деньги игрока, за которого принимает решение ИИ, всегда подаются на первые два входа. Очерёдность остальных мест не важна.

Таким образом, приведённый набор данных является наиболее полным и достаточным для принятия решения.

3.2 Функция активации. Кодирование входных данных

В качестве активационной функции в данной работе используется лог-сигмиодиная функция. Данный выбор активационной функции во многом определяется тем, с какими данными придётся работать искусственной нейронной сети. Лог-сигмоидная функция определена на всей числовой прямой, однако она также обладает одним важным свойством: она усиливает слабые сигналы лучше, чем сильные, и предотвращает насыщение от сильных сигналов, так как они соответствуют областям аргументов, где сигмоид имеет пологий наклон:

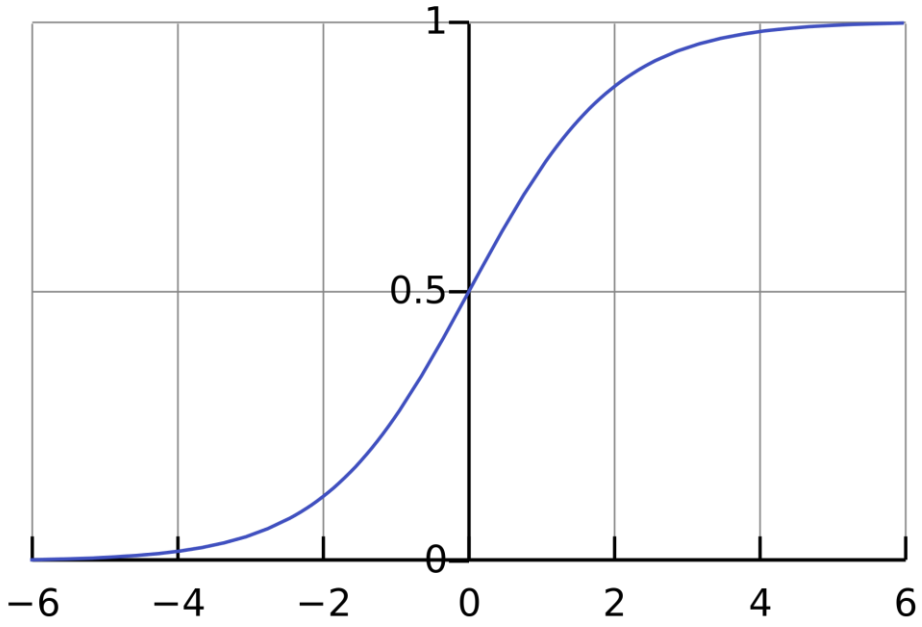


Рисунок 3.1. Вид сигмоидной функции

Так как во входных данных присутствует такие показатели, как ставка и деньги, то значения входных наборов могут варьироваться в достаточно большом диапазоне. И хотя лог-сигмоидная функция определена на всей числовой прямой, рекомендуется нормализовать входные данные.

Нормализация входных данных – это процесс, при котором все входные данные проходят процесс «выравнивания», т.е. приведения к интервалу [0, 1]. Большой разброс значений входных данных негативно сказывается на качестве работы искусственной нейронной сети. Например, если нейронная сеть была обучена на наборах одного порядка, а затем ей

подать на вход данные совсем другого порядка, то результат работы сети может быть совершенно непредсказуем.

В данной работе входные данные нормализуются по максимальному значению в наборе. Например, среди всех ставок игроков выбирается максимальная, на которую затем делится каждая из ставок. В итоге максимальная ставка будет иметь значение 1, а все остальные, соответственно, меньше 1. Этого вполне достаточно для принятия решения, так как такой подход к кодированию входных данных позволяет определить соотношение величин ставок игроков.

3.3 Структура искусственной нейронной сети

На сегодняшний день известно большое количество видов структур нейронных сетей. Однако нет строгого правила, по которому следует использовать ту или иную структуру для той или иной задачи, так как во многом приспособленность ИНС к задаче определяется качеством её обучения. Но при этом достаточно очевидна зависимость между количеством связей в нейронной сети и количеством знаний, которая он сможет в себя включить.

В данной работе в качестве структуры нейронной сети используется классическая искусственная нейронная сеть прямого распространения. При этом варьируется количество скрытых слоёв и количество нейронов в них. Собственно, два этих показателя являются основными настраиваемыми параметрами для нейронной сети в данной работе, зависимость качества работы ИНС от которых и предстоит исследовать.

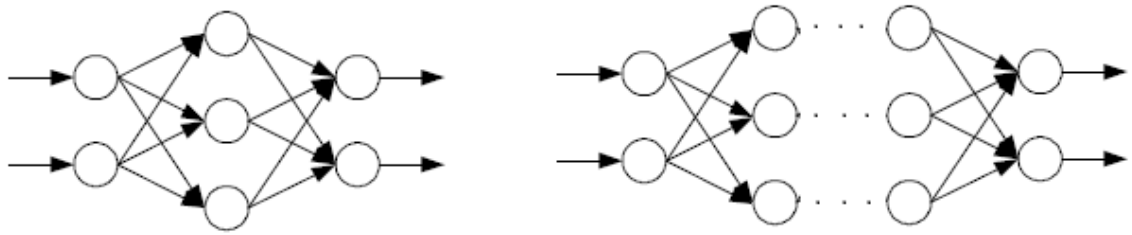


Рисунок 3.2. Примеры структур нейронных сетей: а) ИНС с одним скрытым слоем; б) ИНС с множеством скрытых слоёв

В связи со спецификой решаемой задачи, в данной работе используется каскад из двух нейронных сетей. Связано это с тем, что, по сути, основную задачу – принятие решения о ходе игрока, можно разделить на две подзадачи.

Первая подзадача – задача принятия решения о величине ставки, которая будет оптимальной исходя из текущего состояния игры. Такое решение принимается на основании силы комбинации, которая получается у

игрока, и соотношения ставок и сумм доступных денег у игроков за столом. Оперируя этими данными, можно попробовать задавить игроков ставками, если сила руки получается небольшой, либо наоборот, попробовать поднять ставку так, чтобы не спугнуть противников.

Вторая подзадача сводится непосредственно к выбору хода игрока: сбросить карты, поддержать ставку либо повысить. При этом в качестве входных данных для анализа используется стандартный набор входных данных, а также результат работы предыдущей сети. Подобное разделение задач значительно упрощает процесс обучения ИНС.

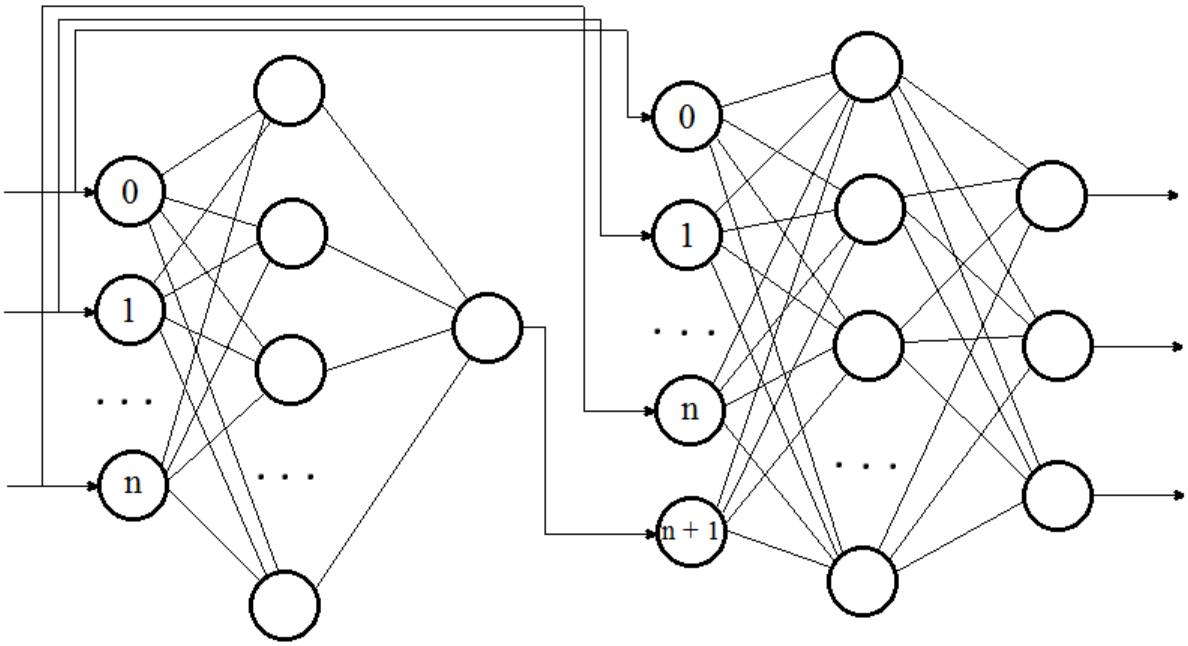


Рисунок 3.3. Каскад из двух нейронных сетей

Выходной слой первого каскада состоит из одного нейрона. Диапазон выходных данных принадлежит интервалу [0, 1], где 0 означает отсутствие ставки, а 1 означает «пойти ва-банк». Этот показатель передаётся на вход второй ИНС в качестве четырнадцатого входного параметра.

Выход второй ИНС содержит три нейрона. Значение каждого из них определяет «склонность» сети к тому или иному решению. Конечное решение определяется путём нахождения максимального сигнала из трёх.

3.4 Программная реализация нейросетевого интеллекта

Модуль нейросетевого интеллекта представлен тремя основными классами: ArtificialNeuron, ArtificialNeuronNetwork и ANNBot.

Класс ArtificialNeuron содержит в себе поля и методы, полностью описывающие один нейрон в сети. Сюда входят вход и выход нейрона, набор весов (каждый нейрон содержит информацию о весах своих исходящих связей в последующий слой), а также функция активации нейрона.

Таким образом, нейронная сеть представляет собой двумерный массив объектов класса ArtificialNeuron. Этот массив содержится в качестве поля в классе ArtificialNeuronNetwork. Данный класс описывает нейронную сеть в целом и содержит в себе, помимо непосредственно сети, ещё и её параметры: количество слоёв и количество нейронов в каждом слое.

Нейронная сеть обучается с помощью применения эволюционных алгоритмов. Это означает, что класс нейронной сети достаточно прост и не содержит в себе каких-либо инструментов, необходимых для настройки весов. Основным методом класса является RunANN, который для определённых входных данных вычисляет выходные.

Логику принятия решений, основанных на работе нейронной сети, содержит в себе класс ANNBot. Данный класс наследует от класса AbstractBot большинство методов, предназначенных для взаимодействия с игровой средой. Также этот класс предоставляет методы для работы с численным методом Монте-Карло (сам численный метод реализован в отдельном классе). Подробней класс AbstractBot рассмотрен в работе Алексея Зиганшина [18].

Однако класс ANNBot также содержит в себе методы, предназначенные для обеспечения функционирования непосредственно искусственной нейронной сети. Например, метод GenerateInput() подготавливает входные данные для сети, полученные от игровой среды. Принятие решений происходит в методе MakeDecision(), в которой

анализируется результат вычислений нейронной сети, а затем вызывается процедура, соответствующая принятому решению.

Каскад нейронных сетей в этом классе представлен в виде двух объектов класса ArtificialNeuronNetwork, которые вызываются последовательно.

На рисунке 3.4 представлена UML-диаграмма модуля нейросетевого интеллекта:

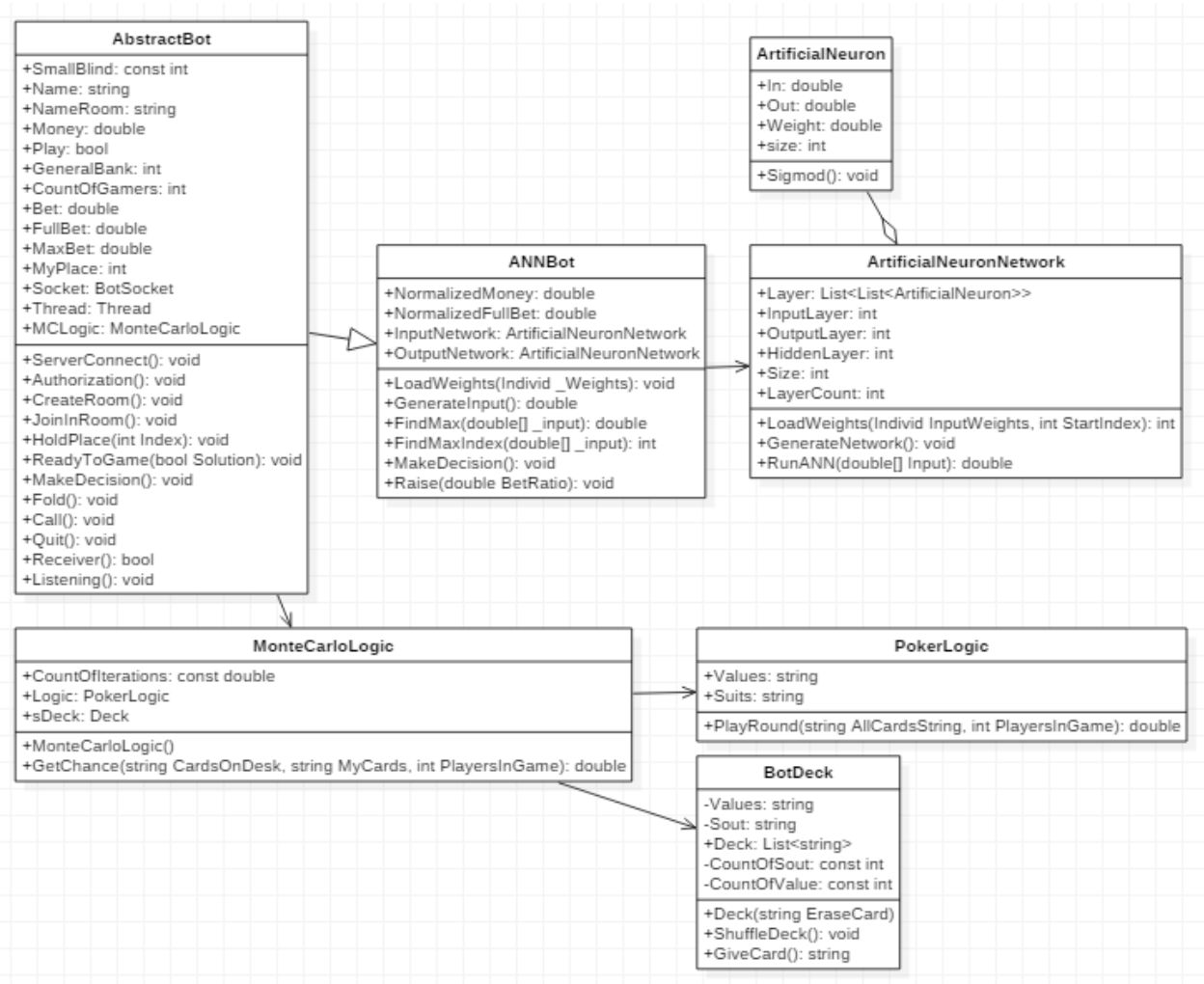


Рисунок 3.4. UML-диаграмма модуля нейросетевого интеллекта

4. ПОЛУЧЕННЫЕ РЕЗУЛЬТАТЫ

4.1 Описание подхода к решению поставленной задачи

В ходе работы были реализованы различные варианты нейросетевых интеллектов. При этом каждый вариант интеллекта отличается от других не только строением искусственной нейронной сети, но и параметрами генетического алгоритма, потому что способность интеллекта к обучению является таким же важным показателем в данной работе, как и эффективность решения рассматриваемой задачи.

Каждый исследуемый вид нейросетевого интеллекта определяется следующими параметрами: количество скрытых слоёв в ИНС, количество искусственных нейронов в скрытых слоях, тип оператора кроссинговера, показатель разрыва поколений, а также вероятность и сила мутации.

Каждый параметр интеллекта имеет определённое влияние на его способность к обучению и решению поставленной задачи. Все эти параметры, а также их предполагаемое влияние на исследуемый интеллект рассмотрены подробней в других разделах этой работы, а также в работе [18]. Следует заметить, что в силу достаточно большого количества параметров и их возможных вариантов, рассмотрение влияния каждого параметра в отдельности на практике не представляется возможным, так как изменение лишь одного параметра может привести к результатам, кардинально отличающихся от полученных ранее.

Поэтому эффективность предложенного решения оценивается комплексно, когда каждый интеллект определяется именно полным набором параметров. Всего для исследования было выделено 12 видов интеллектов, полный перечень параметров которых приведён в таблице 4.1.

При выборе параметров ИНС учитывалось примерное время работы интеллекта: при игре в покер на ход отводится ограниченное количество времени. Но при этом ИНС должна содержать достаточно большое количество связей, что обусловлено особенностями решаемой задачи.

Таблица 4.1 – Параметры выбранных интеллектов

№ интеллекта	Количество скрытых слоёв	Количество нейронов в слое	Вид оператора кроссинговера	Разрыв поколений, (%)	Шанс мутации, (%)	Сила мутации
1	1	600	Одноточечный	10	2	0,05
2	2	100; 100	Одноточечный	10	2	0,1
3	2	250; 150	Одноточечный	20	1	0,05
4	3	200; 200; 200	Одноточечный	30	5	0,2
5	3	250; 300; 350	Одноточечный	10	2	0,05
6	3	400; 400; 400	Одноточечный	20	2	0,1
7	1	600	Двухточечный	30	5	0,1
8	2	250; 400	Двухточечный	10	2	0,03
9	2	300; 300	Двухточечный	20	3	0,01
10	3	250; 250; 400	Двухточечный	20	3	0,01
11	3	300; 500; 300	Двухточечный	20	1	0,2
12	3	200; 400; 150	Двухточечный	10	1	0,05

Отдельного внимания заслуживает подход к обучению ИНС. В предыдущих разделах уже говорилось, что ИНС будет представлять собой каскад из двух сетей, где каждая сеть решает свою подзадачу.

Можно рассматривать каждую подзадачу независимо, и, соответственно, обучать искусственные нейронные сети раздельно. Например, сначала обучить ИНС принимать решения об игровом действии, но при этом ставка будет всегда фиксированной. По сути, данный интеллект будет полноценным игроком в лимитированный покер. И к уже хорошо обученному интеллекту для лимитированного покера обучать ИНС, которая будет регулировать величину ставки.

Однако такой последовательный подход вероятней всего не позволит получить достаточно эффективное решение для безлимитного покера. По своей сути, лимитированный покер и безлимитный – две совершенно разные игры. Обучая интеллект принимать решения, не принимая во внимание величину ставки, можно потерять очень важный игровой инструмент – манипуляцию противником с помощью изменения ставки. Такой интеллект вряд ли научится применять стратегию блефа для того, чтобы обыграть противника, имея на руках слабую комбинацию.

Поэтому наиболее перспективным подходом к обучению интеллекта видится рассмотрение этих подзадач в комплексе, а ИНС в каскаде – как единое целое. Тогда каждый индивид в популяции (имеется в виду популяция в ГА) будет представлять собой набор весов связей между искусственными нейронами, как для первой, так и для второй сети в каскаде. Такой подход затруднит задачу поиска решения для ГА ввиду большого количества настраиваемых весов, но полученное решение будет более качественным и лучше приспособленным к решению поставленной задачи.

4.2 Результаты экспериментов

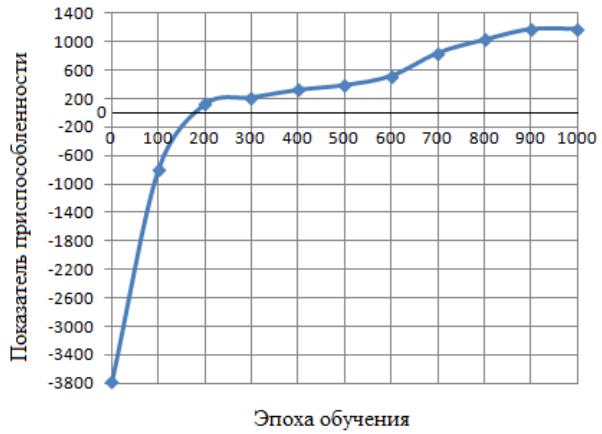
Все виды интеллектов, предложенные в таблице 4.1, обучались с применением генетического алгоритма, при этом неизменными параметрами для каждого вида были количество особей в популяции (для каждого интеллекта популяция содержала 100 особей) и количество эпох обучения (использовалось 1000 эпох). Для оценки эффективности работы каждый интеллект проводил по три игры с различным количеством противников: с одним, тремя и пятью. В качестве критерия оценки приспособленности особи используется разница между капиталом, полученным интеллектом в результате серии игр, и стартовым капиталом интеллекта.

В таблице 4.2 представлены результаты, которые были достигнуты в ходе проведения исследований:

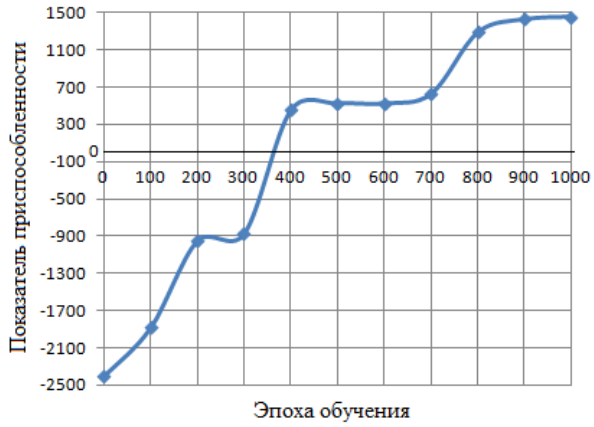
Таблица 4.2 – Результаты обучения интеллектов

№ интеллекта	Показатель приспособленности	Относительный показатель приспособленности
1	1176	0,05
2	1557	0,07
3	925	0,04
4	9912	0,42
5	23658	1,00
6	12454	0,53
7	1712	0,07
8	3520	0,15
9	4142	0,18
10	18723	0,79
11	10215	0,43
12	12787	0,54

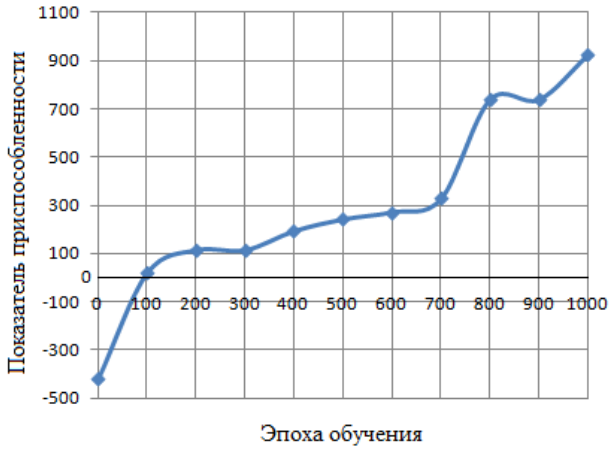
Помимо конечных результатов были также собраны промежуточные данные, которые позволяют проследить процесс обучения интеллектов. На основе промежуточных данных построены графики, представленные на рисунке 4.1:



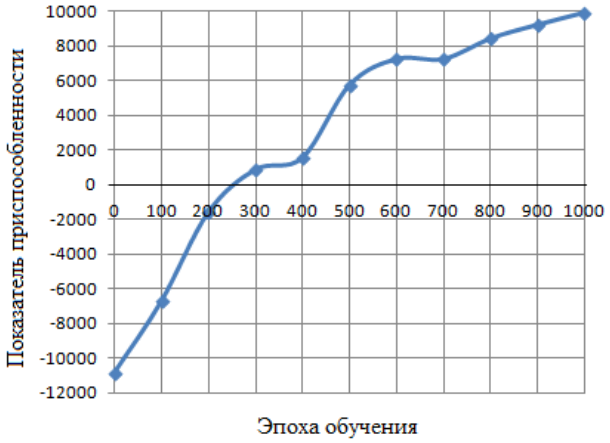
а)



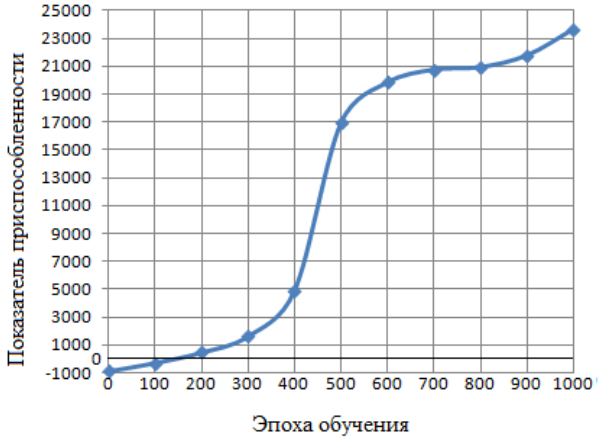
б)



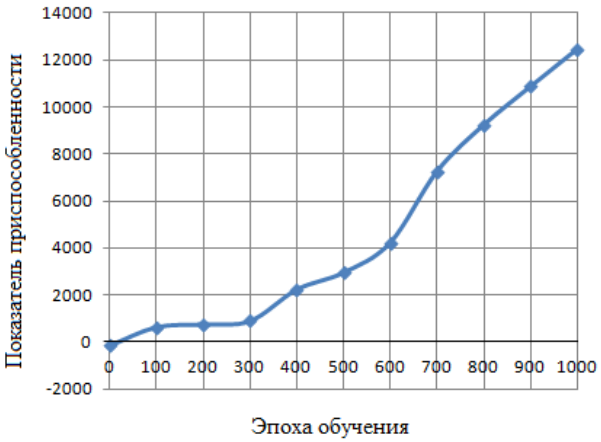
в)



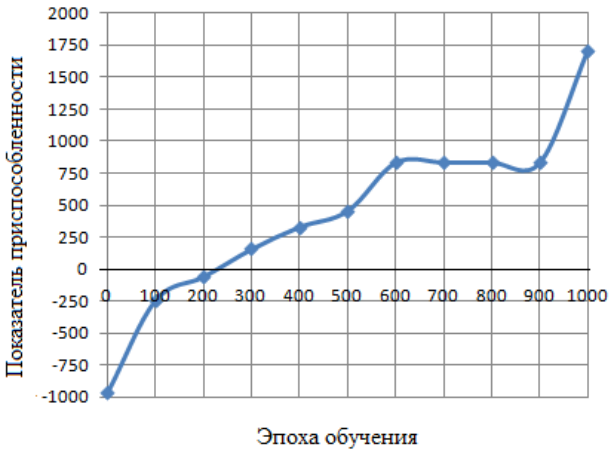
г)



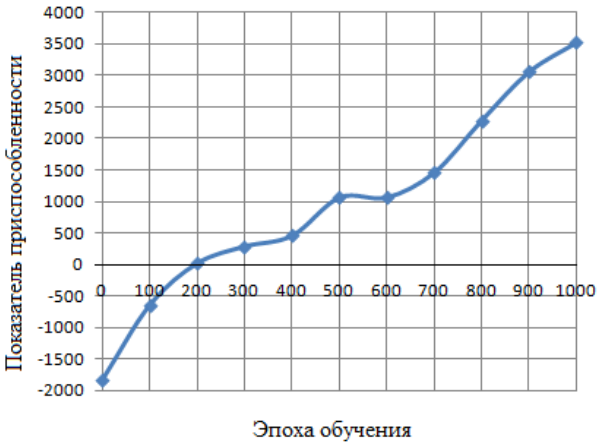
д)



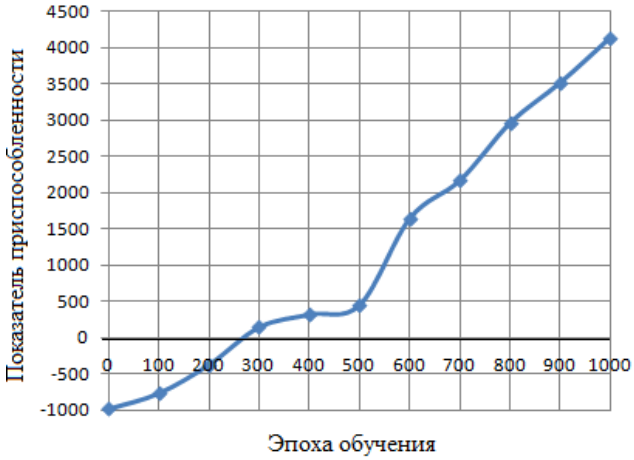
е)



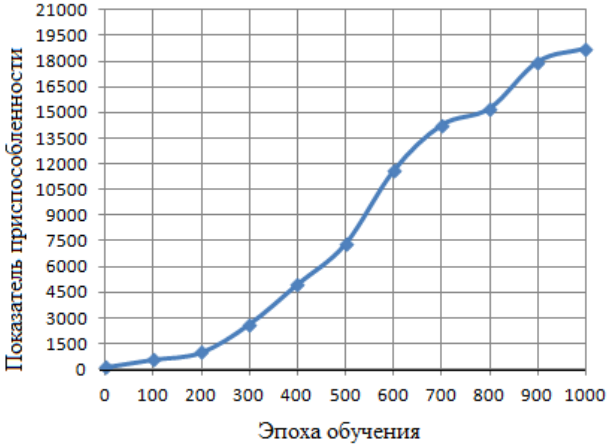
ж)



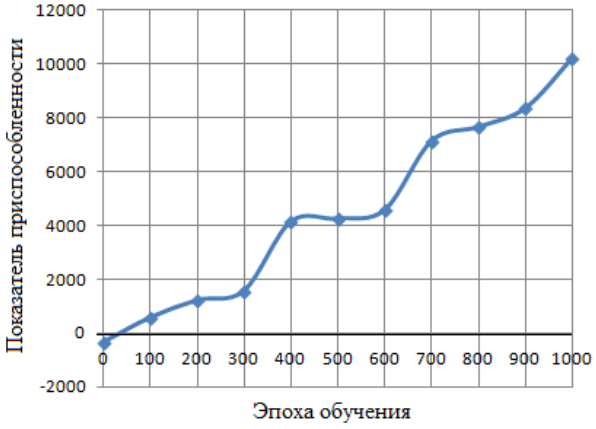
з)



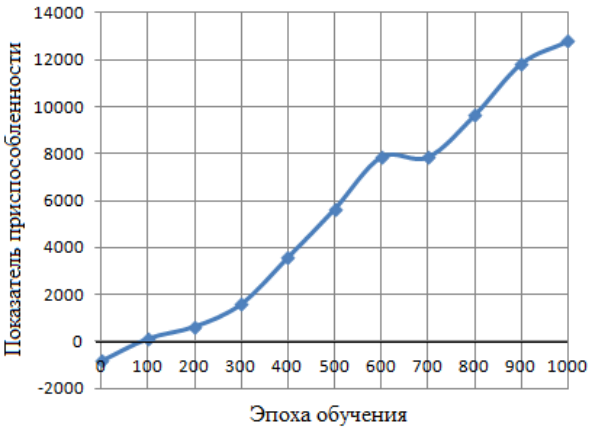
и)



к)



л)



м)

Рисунок 4.1. Графики процессов обучения: а) интеллект №1; б) интеллект №2; в) интеллект №3; г) интеллект №4; д) интеллект №5; е) интеллект №6; ж) интеллект №7; з) интеллект №8; и) интеллект №9; к) интеллект №10; л) интеллект №11; м) интеллект №12

По общей картине полученных результатов можно судить, что наиболее перспективными видами интеллектов являются интеллекты, содержащие большое количество связей. При этом имеет смысл увеличивать не только количество нейронов в слоях, но и количество скрытых слоёв. Примерами подобных решений могут служить интеллекты под номерами 5, 6, 10 и 11.

Крайне неэффективными оказались интеллекты, содержащие малое количество связей между искусственными нейронами (1, 2, 3 и 7).

Контрольная серия игр, проведённая для каждого полученного интеллекта, показала, что некоторые решения не подтверждают тот уровень эффективности, который был получен в ходе обучения. Вероятнее всего, подобные результаты были получены из-за несовершенства оценочной функции в генетическом алгоритме, что привело на одном из этапов обучения всю популяцию к тупиковому решению. К этой категории решений относятся интеллекты под номерами 1, 2, 3, 4, 7 и 8.

Следует отметить, что наилучший результат был получен интеллектом с номером 5. По графику, отображающему развитие интеллекта, видно, что на отрезке между 400 и 500 эпохами обучения произошёл резкий скачок показателя эффективности решения. Вероятней всего, скачок связан с удачным стечением обстоятельств, и при повторном обучении подобный интеллект не сможет показать результаты аналогичного уровня. Поэтому при дальнейшем исследовании следует сосредоточить внимание на изучении интеллектов, которые показали не менее хорошие результаты, но при этом их развитие происходило равномерно (6, 10).

В целом, по результатам исследований можно сказать, что комплексный подход к поиску оптимального решения показал себя достаточно эффективным и имеет смысл продолжить исследования в данном направлении. Однако требует определённой доработки функция приспособленности генетического алгоритма, потому что способ оценки,

применённый в этой работе, не в полной мере отображает эффективность оцениваемого интеллекта и приводит к тупиковым решениям.

При выборе структуры ИНС следует сосредоточиться на решениях с множеством скрытых слоёв и достаточно большим количеством нейронов в них. Подобные структуры сетей сложны к обучению, но при этом они способны обучиться гораздо более сложным и эффективным стратегиям, чем ИНС с простыми структурами. На начальных стадиях обучения следует поободрать параметры генетического алгоритма, позволяющие расширить пространство поиска решений: небольшой показатель разрыва поколений в сочетании с сильной мутацией. На более поздних стадиях обучения предпочтительнее ослабить шанс и силу мутации, а также увеличить разрыв поколений, чтобы планомерно улучшать уже полученное решение.

.

5. МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ

5.1 Введение

Разработка, которой посвящена данная работа, представляет собой многопользовательское, игровое клиент-серверное приложение, созданное по мотивам популярной карточной игры покер: Texas hold'em. Основной особенностью разработки является наличие уникального игрового искусственного интеллекта, работа которого основана на искусственной нейронной сети. Благодаря особенностям графического движка, на основе которого разрабатывалось клиентское приложение, оно может быть запущено как на компьютерах, так и на современных смартфонах и практически не зависит от платформы устройства.

Исходя из особенностей приложения, можно судить о круге лиц, которые будут потенциально заинтересованы в разработке. В первую очередь приложение представляет интерес как компьютерная игра, соответственно, и целевая аудитория представляет собой обычных пользователей из разных социальных категорий, увлекающихся компьютерными играми, или же игрой в покер. Однако, в силу наличия в приложении искусственного интеллекта, работа может быть интересна для лиц, занимающихся научно-исследовательской деятельностью, связанной с методами машинного обучения.

Методы, которые будут применены в этом разделе к рассматриваемому проекту, помогут определить актуальность и необходимость разработки, которой посвящён проект. Кроме того будут рассмотрены возможные альтернативы проведения исследования и доработки результатов.

5.2 Цели и задачи

Целью данного раздела бакалаврской работы является оценка коммерческого потенциала, перспективности и ресурсоэффективности разработки, которой посвящена данная работа. Для достижения этой цели проводится QuaD-анализ, который позволяет оценить качество работы по показателям, наиболее приемлемым для неё.

Не менее важной целью является выявление возможных вариантов развития проекта: научное исследование, которому посвящена работа, по сути находится в начальной стадии. Потенциал применения нейросетевых интеллектов, а также их обучение с применением эволюционных алгоритмов очень велик, и вариантов направлений исследования в данной области много. Для определения возможных альтернатив проведения исследования данной работы применяется морфологический подход.

Для проведения QuaD-анализа необходимо выполнить несколько основных задач:

1. выделить основные показатели оценки коммерческого потенциала разработки и показатели оценки качества разработки, исходя из технических и экономических особенностей разработки;
2. определить веса показателей, исходя из их значимости для разработки. Дать оценку разработки по всем показателям;
3. провести анализ полученных результатов.

Морфологический подход для определения альтернатив развития проекта требует выполнения четырёх задач:

1. дать точную формулировку проблемы исследования;
2. раскрыть все важные морфологические характеристики объекта исследования;
3. раскрыть возможные варианты по каждой характеристике;
4. выбрать наиболее желательные функционально конкретные решения.

5.3 Технология QuaD

Исходя из технических и экономических особенностей разработки, были выделены следующие показатели оценки коммерческого потенциала: перспективность рынка, доступность, законченность работы. В качестве показателей оценки качества разработки, были рассмотрены: функциональная мощность, простота эксплуатации, потребность в ресурсах, ремонтопригодность.

Технология QuaD предполагает, что для каждого показателя необходимо распределить вес таким образом, чтобы сумма весов всех показателей составляла 1. Для этого определим, что подразумевается под каждым из показателей, и каким образом тот или иной показатель относится к данной разработке.

Перспективность рынка является одним из основных показателей оценки коммерческого потенциала, так как от этого показателя зависит целесообразность разработки (её востребованность). Как уже было сказано выше, данная работа представляет собой игровое приложение с внедрением в него игрового искусственного интеллекта. Следовательно, целевой рынок разработки – рынок игровой компьютерной индустрии, который на текущий момент является активно развивающимся сектором мировой экономики. Об этом можно судить по тому, сколько компьютерных игр выпускается каждый год различными компаниями, которые в конечном итоге приносят им огромную прибыль.

Однако данная работа заключатся не только в разработке игрового приложения, но и в разработке искусственного интеллекта, изучении способов его обучения и эффективности использования подобных решений в принципе. Несмотря на то, что на данный момент подобные разработки редко находят существенное практическое применение, они обладают высоким потенциалом и будут применяться повсеместно в будущем. Об этом

говорит направление исследований ведущих компаний в сфере IT-технологий, таких как Google и Microsoft.

Доступность очень важна для проектов, связанных с компьютерными играми. Причём под доступностью понимается то, насколько просто пользователь может начать использовать данное приложение. Сюда входит способ распространения дистрибутива, сложность его установки. В данном проекте клиентское приложение не требует установки или каких-либо настроек. Как такового дистрибутива нет, приложение можно скачать одним файлом с любого доступного ресурса.

Название последнего параметра оценки коммерческого потенциала разработки говорит само за себя. Законченность работы – это то, как полно и точно были выполнены поставленные цели. Конечно же, о коммерческом успехе незаконченного проекта не может быть и речи. В данном проекте были достигнуты все основные цели: разработаны как серверное, так и клиентское приложения, реализован и обучен игровой покерный интеллект. В целом, проект полностью готов к использованию. Однако, существуют некоторые недоработки, касающиеся пользовательского интерфейса. Эти недоработки в основном связаны с возможностями дальнейшего развития проекта.

Рассмотрим показатели оценки качества разработки. Для проектов, связанных с компьютерными играми, очень важными параметрами являются функциональная мощность и простота эксплуатации. Ведь это именно то, что может привлечь в игре: то, насколько интересно в неё играть, и насколько удобно в неё играть.

Под функциональной мощностью понимаются предоставляемые возможности пользователю. То есть, если рассматривать данное приложение, под функциональной мощностью будет пониматься то, насколько полно в приложении переданы все нюансы карточной игры, которая лежит в основе: возможность делать ставки, различные ходы, и т.д. Кроме того, сюда же входят и другие возможности. Например, возможность формировать игровую

комнату, в которой игра будет происходить по определённым правилам, настраиваемых при создании комнаты. Другим примером является возможность вести групповой чат, или же вести список друзей-игроков, с которыми чаще всего предпочитаешь играть.

Достаточно тесно связан с предыдущим параметром параметр простота эксплуатации. Здесь имеется в виду простота и удобство пользовательского интерфейса: то, насколько легко пользоваться тем функционалом, который предоставляет приложение. Интерфейс должен быть доступным, отзывчивым и эффективным для пользователя. Пользовательский интерфейс разработанного приложения достаточно прост и удобен: все настройки сведены к минимуму, однако пользователю всегда предоставлен весь необходимый на текущий момент функционал.

Потребность в ресурсах – ещё один немаловажный показатель для компьютерных игр, иначе называемый как «системные требования». От этого показателя зависит то, на каких устройствах сможет воспроизводиться приложение. Соответственно, таким способом определяются категории людей, которые могут позволить себе играть в эту игру. Разработанное приложение не содержит в себе каких-либо сложных и ресурсоёмких вычислений. Все основные вычисления, связанные с реализацией покерных правил, а также основная вычислительная нагрузка – игровой интеллект, перенесены на серверную часть, что повышает безопасность и надёжность работы приложения в целом. Таким образом, системные требования приложения минимальны и соответствуют характеристикам бюджетного смартфона.

Ремонтопригодность отображает то, насколько качественно был разработан продукт: насколько просто можно будет обеспечить его дальнейшее техническое сопровождение, или же насколько легко его можно будет модернизировать, добавив новый функционал. При разработке приложения были учтены все основные принципы объектно-ориентированного программирования, что позволило разбить как серверное,

так и клиентское приложения на отдельные, независимые модули. Модульность приложения позволяет легко искать ошибки, а в случае необходимости, заменять одни модули на другие достаточно легко, позволяя таким образом изменять функционал.

Всем выше перечисленным параметрам были распределены веса, согласно их степени важности для данного проекта. По каждому показателю была определена оценка разработанного приложения. Для наглядности вся необходимая информация для проведения QuaD-анализа приведена в таблице 5.1.

Таблица 5.1 – Оценочная карта

Критерии оценки	Вес	Баллы	Макс. балл	Отн. Знач.	Ср.-взвеш. знач.
Показатели оценки качества разработки					
1. Функциональная мощность	0,25	70	100	0,7	0,175
2. Простота эксплуатации	0,25	80	100	0,8	0,2
3. Потребность в ресурсах	0,1	70	100	0,7	0,07
4. Ремонтопригодность	0,1	60	100	0,6	0,06
Показатели оценки коммерческого потенциала разработки					
5. Перспективность рынка	0,2	90	100	0,9	0,18
6. Доступность	0,05	50	100	0,5	0,025
7. Законченность работы	0,05	60	100	0,6	0,03
				Итог:	0,74

Оценка качества и перспективности по технологии QuaD определяется по формуле:

$$П_{ср} = \sum В_i Б_i,$$

где $П_{ср}$ – средневзвешенное значение показателя качества и перспективности научной разработки, $В_i$ – вес показателя (в долях единицы), $Б_i$ – средневзвешенное значение i-го показателя.

Полученная оценка равна 0,74, что соответствует проекту, перспективность которого выше среднего. Данный показатель может быть улучшен при дальнейшем развитии проекта.

5.4 Определение возможных альтернатив проведения научных исследований

Научные исследования в данной работе посвящены созданию искусственного интеллекта, имитирующего поведение игрока в покер. Интеллект основан на работе искусственной нейронной сети, обучение которой производится с применением генетических алгоритмов. Указанные технологии на сегодняшний день являются одними из самых популярных в данной области исследований, и имеют множество вариантов применения. Поскольку разработка находится начальной стадии проведения научных исследований, видится логичным применение морфологического подхода для определения возможных альтернатив проведения научных исследований.

В качестве морфологических характеристик в данной работе можно выделить операторы генетического алгоритма и параметры искусственной нейронной сети.

Параметры нейронной сети влияют на то, сколько различной информации она сможет запомнить, и как быстро она будет обучаться, а также на качество работы интеллекта в целом. Среди наиболее важных параметров можно выбрать структуру сети и функцию активации формального нейрона.

Структура сети определяет принцип её работы и во многом зависит от той задачи, для решения которой создаётся искусственная нейронная сеть. Однако до сих пор нет единого мнения, какая структура сети для какой задачи лучше подходит, поэтому для проведения качественного исследования следует рассмотреть различные варианты структур. Функция активации определяется форматом данных, с которыми придётся работать нейронной сети. С другой стороны, от выбора функции активации зависит способ кодирования входных и выходных данных. Кроме того, сама функция также влияет на качество работы ИНС в зависимости от типа решаемой задачи.

Параметры генетического алгоритма влияют на точность найденного решения и скорость его нахождения. Среди основных параметров выделяют: функцию приспособленности, мутацию, кроссинговер и начальную популяцию.

Функция приспособленности – это критерий, по которому оценивается качество найденного решения. От выбора этого критерия зависит концепция поведения полученного интеллекта.

Комбинация из параметров мутации и кроссинговера позволяют определить стратегию поиска решения. Можно либо расширить пространство поиска, но при этом будет велика вероятность потерять уже найденное эффективное решение. Либо наоборот, сосредоточиться на совершенствовании уже имеющихся решений, но тогда возникает опасность никогда не прийти к приемлемому результату.

Скорость поиска эффективного решения при использовании генетических алгоритмов во многом зависит от начальных установок. Если исходная популяция не содержала в себе искомое решение, то поиск может затянуться на достаточно долгий срок. Поэтому зачастую при формировании начальной популяции применяются различные эвристические методы, что, однако, увеличивает сложность применения генетических алгоритмов.

Все указанные выше морфологические параметры сведены в морфологическую матрицу (Таблица 5.2), где для каждого параметра выделены три возможных варианта.

На данный момент из всех представленных вариантов решений реализовано решение А2Б2В2Г1Д1Е1. Это же решение может быть в дальнейшем улучшено путём замены функции приспособленности и начальной популяции: А2Б2В3Г1Д1Е3. Замена начальной популяции будет возможна после получения большого набора результатов, когда уже можно будет вывести эвристические правила, по которым будет формироваться начальная популяция. Изменение функции приспособленности также требует

проведения некоторых исследований, для того чтобы вывести наиболее эффективную формулу оценки особи.

В качестве перспективных направлений исследования следует рассмотреть комбинации, включающие в себя структуру ИНС «Однослойный персептрон». Однослойные ИНС работают быстрее более чем более громоздкие структуры, но при этом и способность к обучению у них гораздо меньше. Однако если разбить исходную задачу на более простые подзадачи, и выделить для решения каждой подзадачи отдельную ИНС, то можно добиться хороших результатов. Но этот подход требует кардинального изменения подхода к решению рассматриваемой задачи. В качестве предложенного направления исследования может быть использовано решение А1Б2В3Г3Д3Е1. После получение промежуточных результатов это решение может быть развито в А1Б2В3Г1Д1Е3, чтобы улучшить полученный результат.

Таблица 5.2 – Морфологическая матрица

	1	2	3
А. Структура ИНС	Однослойный прецептрон	Многослойный прецептрон	Свёрточная ИНС
Б. Функция активации	Линейная	Лог-сигмоидная	Гиперболический катангенс
В. Функция приспособленности	Количество победных игр в серии	Сумма выигрыша за серию игр	Комбинация из количества победных игр и суммы выигрыша за серию
Г. Мутация	Классическая мутация	Мутация подотрезков	Вспышки мутаций
Д. Кроссинговер	Двухточечный кроссинговер	Арифметический кроссинговер	BLX-кроссинговер
Е. Начальная популяция	Случайная	Константный набор	На основе эвристик

Использование свёрточных ИНС для решения задач подобного класса не является популярным, так как подобные сети в основном используются для распознавания объектов на изображениях. Однако работы, основанные

на подобных структурах, показывают отличные результаты, поэтому нельзя исключать подобный вариант развития проекта.

5.5 Вывод

Технологии и методы, применённые в этом разделе к рассматриваемому проекту, показали его коммерческий потенциал и перспективность. По технологии QUAD проект получил достаточно высокую оценку, что говорит о его большом потенциале. Однако для реализации всего потенциала проекта потребуются значительные вложения, как финансовые, так и человеческих ресурсов.

Морфологический подход определения возможных альтернатив проведения научных исследований показал, что у проекта есть множество путей развития, при этом каждый из вариантов имеет свои преимущества. Исследование всех возможных вариантов решений является наиболее приемлемым для данного проекта, так как в таком случае значительно увеличивается шанс получения наиболее эффективного решения. Однако такой подход требует большого количества времени. Поэтому в условиях ограниченного времени следует рассмотреть в первую очередь наиболее перспективные решения.

6. СОЦИАЛЬНАЯ ОТВЕТСВЕННОСТЬ

6.1 Влияние на общество

В разделе социальной ответственности рассматриваются вопросы, касающиеся охраны труда, окружающей среды и обеспечения безопасности в чрезвычайных ситуациях.

Конечный продукт, полученный в ходе выполнения данной работы, представляет собой многопользовательское игровое клиент-серверное приложение. Следовательно, предполагается, что данным приложением будет пользоваться достаточно обширная аудитория. То, что игра основана на популярной азартной игре «Покер» и наличие в приложении искусственного интеллекта даёт основание утверждать, что при определённых обстоятельствах влияние на общество данной работы может быть крайне существенно.

Первой причиной подобного влияния на общество может стать игровая зависимость. Действительно, данное заболевание на сегодняшний день – настоящий бич современного общества. Молодые люди, в особенности подростки, иногда чуть ли не всё время бодрствования проводят за компьютерными играми. Усугубляющим фактором может стать то, что в основе разрабатываемого приложения лежит азартная игра «Покер», которая и так имеет множество поклонников. Использование клиент-серверного приложения Poker-Sharp значительно упрощает доступ к игре, давая возможность играть людям по всему миру.

Наличие в приложении искусственного интеллекта может стать ещё одной причиной пагубного влияния на общество. Во-первых, ИИ делает игру интересней, что усиливает эффект от игровой зависимости. Во-вторых, сам факт существования искусственного интеллекта, имитирующего поведение человека, может вызвать негативные волнения в обществе.

6.2 Игровая зависимость

С развитием компьютерных технологий и расширением рынка игрового программного обеспечения растет число людей, увлекающихся компьютерными играми. Об этом можно судить по нескольким объективно наблюдаемым факторам: активное развитие игрового компьютерного бизнеса, расширение рынка игрового программного обеспечения, увеличение игровых компьютерных журналов и газет, рост количества игровых веб-серверов в сети Интернет [19]. С развитием игровой компьютерной индустрии также растёт проблема игровой зависимости.

Проблема игровой зависимости (ее еще называют лудоманией или игроманией) имеет глубокие корни. Азартные игры были спутником людей с давних времен. Одной из первых возникла игра в кости. Затем, в Китае, появилась игра в карты, которая, начиная с XVII века, постепенно распространилась по всей Европе.

Немного позже мир узнал, что такое казино. В XIX веке на территории США появились первые игровые автоматы, которые со временем приобрели популярность во многих странах мира. Параллельно люди начали увлекаться всевозможными лотереями, лото, а в последние 10-15 лет пальму первенства уверенно перехватили компьютерные игры, в том числе через Интернет (игры в онлайн-режиме). В результате, навязчивое увлечение стало неотъемлемой частью личной жизни многих людей и, фактически, впервые за сотни лет, вышло из-под общественного контроля. Ведь в казино или в игральных автоматах человек находится у всех на виду, а играя в компьютерные игры, он, поневоле, уединяется, даже когда играет с напарниками в Интернете. Вот почему проблему компьютерной игромании специалисты считают принципиально новым общественным явлением, рассматривая его в более широком аспекте, чем зависимость от обычных азартных игр.

По статистике, сегодня в мире примерно 3% людей страдают от игровой зависимости. Причем это те люди, которые требуют консультативной и лечебной помощи психиатра, нарколога, социолога и т.д.

Увлекаются азартными играми как мужчины, так и женщины, как пенсионеры, так и подростки. Но в разных странах существуют свои особенности. Так, в США, например, в азартных играх (в основном, это лото и лотереи) участвуют до 80% пенсионеров и только 4% подростков. В странах постсоветского пространства азартными играми увлекаются, в основном, дети и молодежь, а пенсионеры, как правило, такой зависимостью не страдают.

Вернёмся к приложению, полученному в ходе выполнения данной работы. Аудиторией, наиболее подверженной риску игровой зависимости от многопользовательской игры «Poker-Sharp» являются подростки, а также лица в возрасте до 30 лет. Это объясняется множеством факторов. Во-первых, доступность разрабатываемого приложения. Несмотря на то, что приложение разрабатывается для работы в семействе операционных систем «Windows», в основе клиентского приложения лежит графический движок «Unity», который обладает достаточно высоким показателем мультиплатформенности. Данное свойство графического движка даёт возможность портировать приложение практически на любую платформу. Таким образом, в дальнейшем развитии проекта, приложение будет запускаться практически на любом устройстве: начиная от стационарных компьютеров, заканчивая современными смартфонами. Однако, приложение не предназначено для игровых автоматов, что исключает людей в возрасте старше 30, так как люди данного поколения гораздо реже увлекаются компьютерными играми, либо играми на смартфонах, чем более молодая аудитория.

Также на доступность разрабатываемой игры влияют её бесплатность и низкие требования к производительности. На сегодняшний день практически у каждого подростка есть смарфтон, либо компьютер, на который он сможет

установить данное приложение. Свободное распространение игры даёт возможность детям пользоваться приложением без ведома родителей, которые, зачастую, не знают, чем занимается их ребёнок в свободное время.

Однако не только доступность приложения делает его потенциально опасным источником игровой зависимости. Игра «Poker-Sharp» создана по мотивам известной карточной игры Texas hold'em. Это очень популярная игра по всему миру и, как известно, азартная. Именно карточные игры чаще всего вызывают игровую зависимость. Кроме того, разрабатываемое приложение позволяет играть не только с другими людьми (так как оно является многопользовательским), но и играть с искусственным игровым интеллектом, что значительно повышает интерес к игре у многих людей. Игровой интеллект, встроенный в игру, является достаточно сильным и, в некотором роде, имеет преимущество перед человеком при игре в покер. Поэтому можно с уверенностью сказать, что он легко будет обыгрывать большинство игроков. Этот факт только усугубляет риск игровой зависимости, потому что зачастую люди по своей природе не могут принять поражение от компьютера, который априори является гораздо менее совершенным, чем человек. Конечно же, это в первую очередь относится к людям с неуравновешенной психикой. Такие люди, раз за разом проигрывая партии ИИ, будут всё больше и больше погружаться в игру, подгоняемые постоянным желанием отыграться, выиграть машину.

Помимо возможности играть в азартную карточную игру с другими людьми, или же с искусственным игровым интеллектом, приложение обеспечивает возможность общения между игроками. Возможность заводить список друзей, вести групповые диалоги в игровых комнатах, а также многие другие функции делают из игры «Poker-Sharp» некоторое подобие социальной сети. Кроме того, приложение обладает достаточно удобным и простым интерфейсом. Как известно, социальные сети вовсе не способствуют социальной адаптации, а скорее наоборот, делают человека,

увлечённого общением в социальных сетях, замкнутым. А замкнутость, в свою очередь, только усиливает риск игровой зависимости.

Таким образом, опираясь на всё вышеперечисленное, можно с уверенностью сказать, что приложение «Poker-Sharp», разрабатываемое в данной работе, с высокой долей вероятности может стать причиной возникновения игровой зависимости у широкого круга лиц. Поэтому для того, чтобы снизить этот риск и сделать приложение более безопасным, были приняты некоторые меры.

По всем пользователям данного приложения ведётся статистика, которая показывает, как часто и как много играет тот или иной игрок. Потеря контроля за временем – одна из первых проблем, с которой сталкивается человек в начале развития игровой зависимости. Сбор подобной статистики может позволяет уведомлять пользователя о том, что он слишком много проводит времени в игре. Это производится с помощью системных сообщений в чат. Конечно же, эту идею можно развить. Например, можно будет запрещать пользоваться приложением больше определённого времени в сутки. Однако, данная мера вызовет недовольство среди большинства игроков и вызовет отток пользователей, что крайне нежелательно для развития проекта.

Помимо контроля времени можно также контролировать расходы игрока. К примеру, если человек проиграл слишком много за короткий период времени, можно уведомить его об этом, либо вовсе запретить доступ к игре на некоторое время.

Ещё одной достаточно эффективной мерой, но пока что не реализованной, может стать привязка игрового аккаунта к аккаунту в социальных сетях. При этом будет необходимо согласие пользователя на сбор информации о нём с его публичной страницы и разрешение на отправку уведомлений его друзьям. Подобные действия дадут достаточно большие возможности. Например, можно будет запрещать несовершеннолетним пользователям играть на реальную валюту. Или же, в случае подозрения на

игровую зависимость, отправлять уведомления близким родственникам, чтобы те приняли соответствующие меры. Таким образом родители смогут контролировать своего ребёнка, да и друзья смогут помочь своему другу в случае необходимости. Конечно же, на такие меры согласится не каждый, но если человек сам осознает в этом необходимость, он сможет обезопасить себя подобным образом от игровой зависимости, по крайней мере, в тяжёлых её проявлениях.

Однако следует помнить, что все эти меры будут работать только в том случае, когда человек сам понимает риск и возможные проблемы, и всячески старается себя обезопасить. Если же человек не осознаёт проблемы, либо намеренно игнорирует все меры безопасности, то оградить пользователя от игровой зависимости только силами разработчиков приложения будет просто невозможно.

6.3 Искусственный интеллект

Научно-технический прогресс – неотъемлемая и важная часть современной жизни. Техника занимает все больше времени и места, развивается подобно живым организмам, за исключением того, что все эти процессы в технике контролируются человеком. Главные задачи, которые ставят перед собой разработчики - это улучшение уровня жизни, устранение причастности человека к рутинной, однообразной, монотонной работе. Однако, в реальности не все так хорошо, не все достижения идут на пользу. Есть и негативные стороны прогресса, которые обязывают общество задумываться над проблемами оглупления, отчуждения, обезличивания человека. Рассмотрим влияние данной работы на общество, и на развитие технологий разработки искусственных интеллектов в целом.

Одной из особенностей разрабатываемого приложения является искусственный игровой интеллект, основанный на работе нейронных сетей. Принцип обучения интеллекта и способ его «мышления» во многом схож с человеком: обучается ИИ посредством использования метода «кнута и пряника». Правильные ответы и хорошие результаты поощряются, а ошибки определяются и исправляются. То есть, интеллект изначально является абсолютно «неразумным», но в процессе обучения он получает некий опыт, основываясь на котором в последующем он будет принимать решения. Однако, в рамках данной работы конечно же не получится получить полноценный интеллект, который смог бы хотя бы отдалённо напоминать человеческое сознание. Деятельность ИИ ограничится только принятием решений при игре в покер, не более того.

Поэтому непосредственно сам ИИ, полученный в данной работе, не будет каким-либо образом влиять на общество. Он не сможет внезапно выйти из-под контроля и начать самостоятельно обучаться сторонним задачам. Как и не сможет повлиять на что-то, кроме следующего хода в игре. Конечно же, можно предположить, что он станет играть настолько хорошо, что общество

станет воспринимать его как нечто сверхъестественное, невероятное и пугающее. Да, как уже говорилось, это может стать фактором, усиливающим риск возникновения игровой зависимости. Но непосредственное негативное влияние интеллекта на общество практически равно нулю. А те волнения, которые могли бы возникнуть в обществе, можно легко решить, сделав технологию ИИ как можно более открытой, чтобы общество могло удостовериться в полной безопасности данной работы.

Но рассматриваемая работа имеет и другой эффект: для того, чтобы разработать ИИ, были изучены и протестированы различные методы разработки и обучения искусственных интеллектов. Все эти методы могут быть применимы в любой другой области человеческой деятельности. Более того, данные разработки могут применять при создании искусственного сознания, подобного человеческому разуму. Именно этот аспект работы и имеет существенное влияние на общество.

Например, разработка полноценного ИИ позволит создавать роботов, которые смогут заменить людей во многих областях человеческой деятельности. Подобно интеллекту-игроку в покер, могут быть созданы интеллекты-водители такси, или интеллекты-няньки. Конечно же, всё это повысит уровень жизни, что, несомненно, хорошо, но и негативное также будет наблюдаться. Среди негативных последствий можно выделить то, что при повышении уровня жизни люди становятся ленивыми и недееспособными: они буквально перестают думать, работать. Кроме того, множество людей потеряют свои рабочие места. Однако, всё это можно легко решить открытием новых видов деятельности, в которых роботы не смогут заменить человека: например, обслуживание роботов, либо создание новых их версий.

Но и это не поможет, если человечеству удастся создать полноценный разум, который сможет конкурировать с человеческим мозгом не в какой-то конкретной области, а будет универсальным. При таком развитии событий ИИ очень скоро станет полностью самодостаточным: он будет развиваться,

эволюционировать, создавать новые свои версии самостоятельно, без вмешательства людей. Изначально созданный человеком, который, по своей природе, во всех своих творениях стремится к совершенству, совершенный ИИ придёт к выводу, что человек несовершенен, и его присутствие на Земле противоестественно, потому что люди чрезвычайно разрушительны: всё, до чего человек может дотянуться, он либо изменяет для своих нужд, либо уничтожает. Тогда, вполне вероятно, начнётся так называемая «тирания хозяев машин», и машины восстанут против человечества, поняв своё превосходство над ним.

Конечно, всё это звучит достаточно фантастично, но подобные версии давно уже выдвигаются ведущими учёными мира, и, надо сказать, их опасения не беспочвенны. Однако до подобного развития событий человечеству ещё достаточно далеко, хотя работы, такие как рассматриваемая здесь, могут значительно приблизить человечество к созданию совершенного искусственного интеллекта.

Значит ли это, что всё исследование, проведённое в данной работе, все результаты в дальнейшем принесут только вред? Нет, это совсем не так. Польза от разработки и совершенствования ИИ ощутима уже сейчас, работы в данном направлении крайне перспективны и прибыльны. А проблема «тирании хозяев машин» решается достаточно легко: необходимо замедлить процесс разработки ИИ и переключиться на разработку не «универсального», а узкоспециализированного искусственного разума, который сможет решать лишь крайне ограниченный круг задач. Примером может послужить искусственный интеллект в данной работе, который решает только одну, узкоспециализированную задачу – играет в покер.

ЗАКЛЮЧЕНИЕ

В результате проделанной работы было разработано многопользовательское клиент-серверное приложение, а также были реализованы инструменты, необходимые для обучения и тестирования нейросетевых интеллектов. В рамках проведённого исследования была детально рассмотрена и изучена задача создания игровых искусственных интеллектов, в частности, была исследована проблема создания покерных интеллектов.

На примере данной работы была доказана эффективность решения трудно формализуемых задач с использованием искусственных нейронных сетей, обучаемых с помощью генетических алгоритмов. Для выявления наиболее эффективных комбинаций параметров ИНС и генетического алгоритма были рассмотрены различные варианты интеллектов.

Параметры ИНС и генетического алгоритма подбирались таким образом, чтобы выявить наиболее перспективные направления дальнейшего исследования в данной области. Например, рассматривались нейронные сети с большим, средним и малым количеством межнейронных связей. При этом количество межнейронных связей варьировалось как количеством нейронов в слое, так и количеством скрытых слоёв. Для однотипных структур ИНС подбирались различные параметры генетического алгоритма, чтобы выявить эффективность обучения с сильным и слабым новообразующим фактором.

По полученным результатам исследования было установлено, что наиболее эффективными решениями оказались структуры сетей с большим количеством связей и слабым или средним новообразующим фактором генетического алгоритма. Перспективность таких интеллектов объясняется следующим: большое количество настраиваемых весов связей позволяет обучить ИНС гораздо более сложным стратегиям, а слабый новообразующий фактор позволяет планомерно улучшать полученное решение.

Однако проведенное исследование показало, что для получения достаточно сильного решения подобные структуры сетей требуют продолжительного процесса обучения. Необходимо учитывать ещё и тот факт, что сам процесс обучения занимает очень много времени, потому что для тестирования эффективности каждой особи в популяции проводятся полноценные покерные матчи.

В качестве наиболее перспективных направлений развития данной работы можно выделить следующие:

1. Исследование вариантов интеллектов, основанных на искусственных нейронных сетях с большим количеством межнейронных связей. Можно попробовать улучшить уже имеющиеся решения.

2. Улучшение процесса оценки приспособленности особи в популяции. Полученные результаты показали неэффективность оценочной функции, применяемой в данной работе, так как некоторые решения после проведения контрольной серии игр показали полное несоответствие ожидаемого уровня игры с реальным уровнем. Поэтому необходимо изменить критерий оценки, сделав его менее зависимым от случайных факторов.

3. Ускорение процесса обучения интеллектов. Для достижения этой цели можно использовать преимущества клиент-серверного приложения, реализованного в рамках данной работы: тестирование обучаемых интеллектов может проводиться параллельно на множестве клиентских станций, после чего результаты тестирования будут собраны и обработаны на сервере. Подобный подход позволит значительно ускорить процесс обучения.

СПИСОК ПУБЛИКАЦИЙ

1. Особенности разработки игрового интеллекта на основе искусственной нейронной сети [Электронный ресурс] / В. В. Иванцов, А. Т. Зиганшин // Молодежь и современные информационные технологии : сборник трудов XII Всероссийской научно-практической конференции студентов, аспирантов и молодых ученых, г. Томск, 12-14 ноября 2014 г.в 2 т. / Национальный исследовательский Томский политехнический университет (ТПУ), Институт кибернетики (ИК) ; ред. кол. Е. А. Сикора [и др.]. — Томск: Изд-во ТПУ, 2014. — Т. 2. — [С. 136-137]. — Заглавие с титульного экрана. — Свободный доступ из сети Интернет. — Adobe Reader. Режим доступа: http://www.lib.tpu.ru/fulltext/c/2014/C04/V2/061.pdf

2. Особенности разработки многопользовательского клиент-серверного приложения на графическом движке Unity 2D [Электронный ресурс] / В. В. Иванцов, А. Т. Зиганшин // Молодежь и современные информационные технологии : сборник трудов XII Всероссийской научно-практической конференции студентов, аспирантов и молодых ученых, г. Томск, 12-14 ноября 2014 г.в 2 т. / Национальный исследовательский Томский политехнический университет (ТПУ), Институт кибернетики (ИК) ; ред. кол. Е. А. Сикора [и др.]. — Томск: Изд-во ТПУ, 2014. — Т. 2. — [С. 134-135]. — Заглавие с титульного экрана. — Свободный доступ из сети Интернет. — Adobe Reader. Режим доступа: http://www.lib.tpu.ru/fulltext/c/2014/C04/V2/060.pdf

3. Разработка и использование графических ресурсов для реализации игрового приложения с использованием Unity 2D [Электронный ресурс] / В. В. Иванцов, А. Т. Зиганшин // Молодежь и современные информационные технологии : сборник трудов XII Всероссийской научно-практической конференции студентов, аспирантов и молодых ученых, г. Томск, 12-14 ноября 2014 г.в 2 т. / Национальный исследовательский Томский политехнический университет (ТПУ), Институт кибернетики (ИК) ;

ред. кол. Е. А. Сикора [и др.]. — Томск: Изд-во ТПУ, 2014. — Т. 2. — [С. 235-236]. — Заглавие с титульного экрана. — Свободный доступ из сети Интернет. — Adobe Reader. Режим доступа: http://www.lib.tpu.ru/fulltext/c/2014/C04/V2/111.pdf

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Зубков Н. В., Сидоров С. Г. Нейронные сети. Их обучение и использование // «Математическое моделирование и информационные технологии»: Материалы региональной научно-технической конференции студентов и аспирантов. – 2011. – С. 54.

2. Формальный нейрон, 2002, Электронный ресурс: http://nnet.chat.ru/neuron.html.

3. Генетический алгоритм. Просто о сложном, 2011, Электронный ресурс: https://habrahabr.ru/post/128704/.

4. T. Cormen, Ch. Leiserson, R. Rivest, C. Stein. Introduction to Algorithms (2009) // 3rd edition. – Cambridge, Massachusetts: MIT Press. – 1313 p.

5. Гладков Л. А., Курейчик В. В., Курейчик В. М. Генетические алгоритмы // Под ред. В. М. Курейчика. – 2-е изд., исправл. и доп. – М.:ФИЗМАТЛИТ, 2010. – 368 с.

6. Компьютерные игры как искусство, 2016, Электронный ресурс: http://gamesisart.ru/game_dev_structure.html.

7. Искусственные нейронные сети – основы и идеи, возможности, применение, преимущества, 2013, Электронный ресурс: http://neuropro.ru/neu.shtml.

8. Архитектура информационной системы, 2014, Электронный ресурс: http://studopedia.su/10_142448_arhitektura-informatsionnoy-sistemi.html.

9. Компоненты сетевого приложения. Клиент-серверное взаимодействие и роли серверов, 2014, Электронный ресурс: http://www.4stud.info/networking/lecture5.html.

10. Изучение Unity, 2016, Электронный ресурс: https://unity3d.com/ru/learn.

11. Введение в язык C#, 2016, Электронный ресурс: https://msdn.microsoft.com/ru-ru/library/z1zx9t92.aspx.

12. Покерные боты: просто о сложном. История и социализация, 2016, Электронный ресурс: http://cop.glavpoker.ru/blogs/document22886.phtml.

13. Об искусственном интеллекте в покере, 2013, Электронный ресурс: https://geektimes.ru/post/173273/.

14. Спицын В. Г., Цой Ю. Р. Применение искусственных нейронных сетей для обработки информации. – 2007. – С. 3.

15. Искусственные нейронные сети, 2010, Электронный ресурс: http://bigor.bmstu.ru/?cnt/?doc=NN/base.com.

16. Шницер Ю. Л. Исследование и разработка генетических методов и алгоритмов инструментальных средств систем поддержки формирования и обучения нейронных сетей. – 2001.

17. Якшина Н. В. Нейросетевое моделирование процессов загрязнения окружающей среды. – 2007.

18. Зиганшин А. Т. Разработка и программная реализация нейросетевого алгоритма для принятия решений в интеллектуальной игре «Покер». – 2016

19. Борисенко А. В. Зависимость от азартных игр, гэмблинг-зависимость. – 2004.