

Министерство науки и высшего образования Российской Федерации
Санкт-Петербургский политехнический университет Петра Великого
Физико-механический институт
Высшая школа теоретической механики и математической физики

Работа допущена к защите
Директор ВШТМиМФ
д.ф.-м.н., чл.-корр. РАН
_____ А.М. Кривцов
«__» _____ 2023 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА
Максимизация экономических параметров на этапе выбора системы
разработки месторождения при помощи нейросети

по направлению 01.03.03 Механика и математическое моделирование
по образовательной программе
01.03.03_03 Математическое моделирование процессов нефтегазодобычи

Выполнил

студент гр.5030103/90301

Р.Р. Галеев

Руководитель

Доцент ВШТМиМФ, к.ф.-м.н.

О.С. Лобода

Консультант

Ведущий эксперт ООО Газпромнефть НТЦ

А.П. Рощектаев

Санкт-Петербург

2023

**САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО**
Физико-механический институт
Высшая школа теоретической механики и математической физики

УТВЕРЖДАЮ

Директор ВШТМиМФ

А.М. Кривцов

«__» _____ 20__ г.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы

студенту Галееву Раулю Рустемовичу, гр. №5030103/90301

1. Тема работы: Максимизация экономических параметров на этапе выбора системы разработки месторождения при помощи нейросети.
2. Срок сдачи студентом законченной работы: 05.06.2023
3. Исходные данные по работе: гидродинамическая модель месторождения, данные об оптимальной системе разработки месторождения, модуль экономической оценки месторождения, документации языков программирования, справочная и научная литература по теме исследования.
4. Содержание работы (перечень подлежащих разработке вопросов): анализ научной литературы по теме работы, выбор оптимальных для задачи алгоритмов нейронной сети и машинного обучения, реализация кода на языке Python, тестирование на гидродинамической модели реального месторождения, анализ полученных результатов.
5. Перечень графического материала (с указанием обязательных чертежей): отсутствует.
6. Консультанты по работе: А.П.Рощектаев, ведущий эксперт ООО Газпромнефть НТЦ.
7. Дата выдачи задания 27.02.2023

Руководитель ВКР _____ О.С. Лобода, доцент ВШТМиМФ, к.ф.-м.н.

Задание принял к исполнению 27.02.2023

Студент _____ Р.Р.Галеев

РЕФЕРАТ

На 37 с., 22 рисунка, 2 таблицы.

ГЛУБОКОЕ МАШИННОЕ ОБУЧЕНИЕ С ПОДКРЕПЛЕНИЕМ, НАУКА О ДАННЫХ, НЕЙРОННЫЕ СЕТИ, ОБРАБОТКА ДАННЫХ, ОПТИМИЗАЦИЯ ПРОЦЕССОВ НЕФТЕДОБЫЧИ

В данной работе рассмотрена задача максимизации экономики месторождения при помощи нейронной сети. Даны общие понятия метода глубокого Q-обучения. Изучена модель месторождения и проведена обработка и анализ данных из модуля экономической оценки месторождения. Разработана реализация алгоритма глубокого Q-обучения с использованием трёхслойной нейронной сети на языке Python. Проведён анализ полученных результатов расчёта экономики месторождения.

THE ABSTRACT

37 pages, 22 pictures, 2 tables.

DEEP MACHINE REINFORCEMENT LEARNING, DATA SCIENCE, NEURAL NETWORKS, DATA PROCESSING, OPTIMIZATION OF THE OIL EXTRACTION PROCESSES

The given work is devoted to the problem of maximizing the economy of a deposit using a neural network. The general concepts of Deep Q-Learning method are given. The field model has been studied and the data from field economic evaluation module has been processed and analyzed. An implementation of the Deep Q-Learning algorithm using a three-layer neural network in Python has been developed. The results of field economics calculation have been analyzed.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	5
ГЛАВА 1. ОБЩИЕ СВЕДЕНИЯ О НЕЙРОННЫХ СЕТЯХ.....	7
1.1. Биологическое понятие нейрона	7
1.2. История создания искусственных нейронных сетей.....	8
1.3. Модель искусственной нейронной сети	9
1.4. Обучение нейронной сети	12
1.5. Обучение с подкреплением.....	13
ГЛАВА 2. АЛГОРИТМ ГЛУБОКОГО Q-ОБУЧЕНИЯ.....	16
2.1. Q-обучение.....	16
2.2. Эпсилон-жадная стратегия.....	17
2.3. Параметризованное Q-обучение.....	17
2.4. Глубокое Q-обучение	18
ГЛАВА 3. СОЗДАНИЕ НЕЙРОННОЙ СЕТИ И РЕАЛИЗАЦИЯ АЛГОРИТМА ГЛУБОКОГО Q-ОБУЧЕНИЯ ДЛЯ УПРАВЛЕНИЯ ТЕМПОМ ВВОДА СКВАЖИН НА МЕСТОРОЖДЕНИИ	19
3.1. Обработка данных.....	19
3.2. Создание нейронной сети.....	20
3.3. Реализация алгоритма глубокого Q-обучения для управления темпом ввода скважин на месторождении	23
ГЛАВА 4. ТЕСТИРОВАНИЕ АЛГОРИТМА И АНАЛИЗ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ	25
4.1. Гидродинамическая модель реального месторождения	25
4.2. Тестирование алгоритма и анализ траты вычислительных мощностей	25
4.3. Результаты	26
ЗАКЛЮЧЕНИЕ	34
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	36

ВВЕДЕНИЕ

На сегодняшний день стоит актуальным вопрос максимизации экономики месторождения. Была принята экономическая стратегия России на период до 2035 (ЭС-2035) [14], которая гласит о том, что целью развития энергетики Российской Федерации является, с одной стороны, максимальное содействие социально-экономическому развитию страны, а с другой стороны, - укрепление и сохранение позиций Российской Федерации в мировой энергетике, как минимум, на период до 2035 года. Для достижения данной цели требуется ускоренный переход к цифровой трансформации и интеллектуализации отраслей топливно-энергетического комплекса, в результате которых новое качество приобретут все процессы в сфере энергетики.

Актуальность данной работы состоит в разработке нового алгоритма подбора оптимальной стратегии для управления темпом ввода скважин на месторождении [13]. С развитием технологий теория нейронных сетей как научное направление получила большое количество практических реализаций, поэтому в данной работе для рассмотрения различных стратегий и выбора лучшей из них будет использована нейронная сеть и алгоритм глубокого Q-обучения. Такое решение поможет исключить влияние человеческого фактора при выборе стратегии темпа ввода скважин [6] для максимизации экономических параметров месторождения.

Цель работы заключается в разработке и реализации нового алгоритма, который будет способен на этапе выбора системы разработки подобрать оптимальную стратегию темпа ввода скважин для максимизации экономических показателей месторождения.

Для достижения заданной цели были сформулированы следующие задачи:

1. Изучить теоретические основы нейронных сетей.
2. Выбрать нейронную сеть и алгоритм обучения для поставленной задачи.
3. Реализовать выбранный алгоритм и нейронную сеть.

4. Обучить выбранную нейронную сеть.
5. Проанализировать и обработать входные данные о месторождении для подачи их в модель нейронной сети.
6. Протестировать созданный алгоритм для подбора оптимальной стратегии управления темпом ввода скважин на модели реального месторождения и проанализировать полученные результаты.

В качестве исходных данных дана модель реального месторождения и модуль экономической оценки месторождения.

ГЛАВА 1. ОБЩИЕ СВЕДЕНИЯ О НЕЙРОННЫХ СЕТЯХ

1.1 Биологическое понятие нейрона

В человеческом мозге находятся миллиарды нейронов. Нейроны человеческого мозга — это специализированные клетки нервной системы, которые передают и обрабатывают информацию в головном мозге и других частях тела. Они имеют уникальную структуру, которая позволяет им эффективно передавать сигналы друг другу и к другим клеткам в организме.

Нейрон головного мозга состоит из тела клетки (сомы), дендритов, аксона и окончаний аксона (терминалов). Тело клетки содержит ядро и другие органеллы, необходимые для жизнедеятельности клетки. Дендриты представляют собой короткие ветви, которые принимают входящие сигналы от других нейронов. Аксон — это длинный отросток, который передает сигналы от тела клетки к другим нейронам или эффекторным клеткам (например, мышцам). Окончания аксона (терминалы) располагаются вблизи других нейронов или эффекторных клеток и передают им сигналы в виде химических веществ, называемых нейромедиаторами. Нейроны головного мозга связаны между собой через синапсы, которые представляют собой контактные точки между окончанием аксона одного нейрона и дендритами другого. Эти связи образуют сложную сеть, которая обеспечивает выполнение различных функций головного мозга, включая обработку информации, управление движениями, регуляцию внутренних органов и т.д.

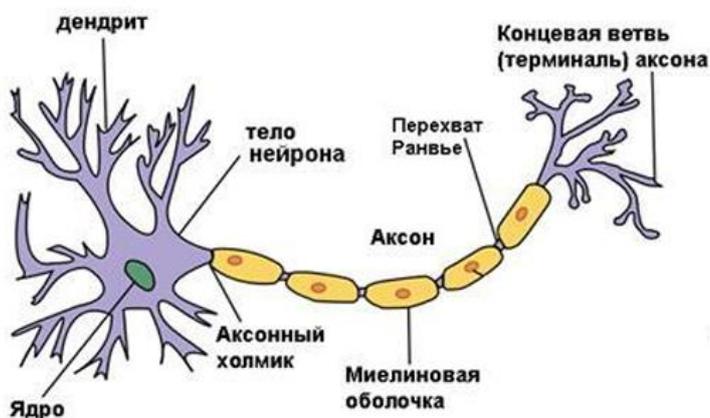


Рисунок 1.1. Строение нейрона

1.2 История создания искусственных нейронных сетей

История компьютерных моделей нейронных сетей берёт начало в середине XX века, когда учёные Уоррен Мак-Каллок и Уолтер Питтс создали вычислительную модель нейросети на основе математических алгоритмов и теории деятельности головного мозга. В 1957 году Фрэнк Розенблатт создал перцептрон – простую нейронную сеть, которая могла распознавать образы. Он использовал ее для классификации изображений. В 1960-х годах были созданы первые многослойные нейронные сети, которые могли решать более сложные задачи, такие как распознавание речи или обработка текста. В 1974 году советским учёным Галушкиным Александром Ивановичем был предложен первый математический алгоритм обучения нейронной сети. В 1980-х годах нейронные сети стали популярными в индустрии и научных кругах, и были использованы для решения различных задач, таких как распознавание образов, прогнозирование временных рядов и определение наличия определенных заболеваний. В 1990-х годах нейронные сети были заменены более эффективными алгоритмами машинного обучения, такими как метод опорных векторов и ансамблирование моделей. Однако, с развитием глубокого обучения в последние годы, нейронные сети стали снова популярными и используются для решения широкого спектра задач. В 2010 году появился термин Deep Learning (Глубокое обучение). Это построение и обучение нейронных сетей с большим числом слоёв (от нескольких десятков и более). Сейчас эта цифра достигает до нескольких сотен. Это стало доступно благодаря развитию графических процессоров, именно их архитектура параллельных вычислений и сделала возможным прорыв в реализации и обучении больших нейронных сетей. В 2012 году число нейронов искусственной нейронной сети достигло 10^6 нейронов. Для сравнения, приблизительное число нейронов в человеческом мозге — 10^9 . Однако, число связей искусственной нейронной сети мало.

1.3 Модель искусственной нейронной сети

Искусственные нейронные сети — это математические модели, которые имитируют работу нервной системы человека. Они состоят из нейронов, которые соединены между собой и передают информацию в форме сигналов. Искусственные нейронные сети используются для решения различных задач, таких как классификация, распознавание образов, прогнозирование и т. д. Они могут быть обучены на больших объемах данных и использоваться для автоматизации процессов в различных областях [5].

Самообучаемая искусственная нейронная сеть представляет собой систему соединённых и взаимодействующих между собой искусственных нейронов. Способность обучаться — одно из главных преимуществ нейронной сети перед классическими алгоритмами.

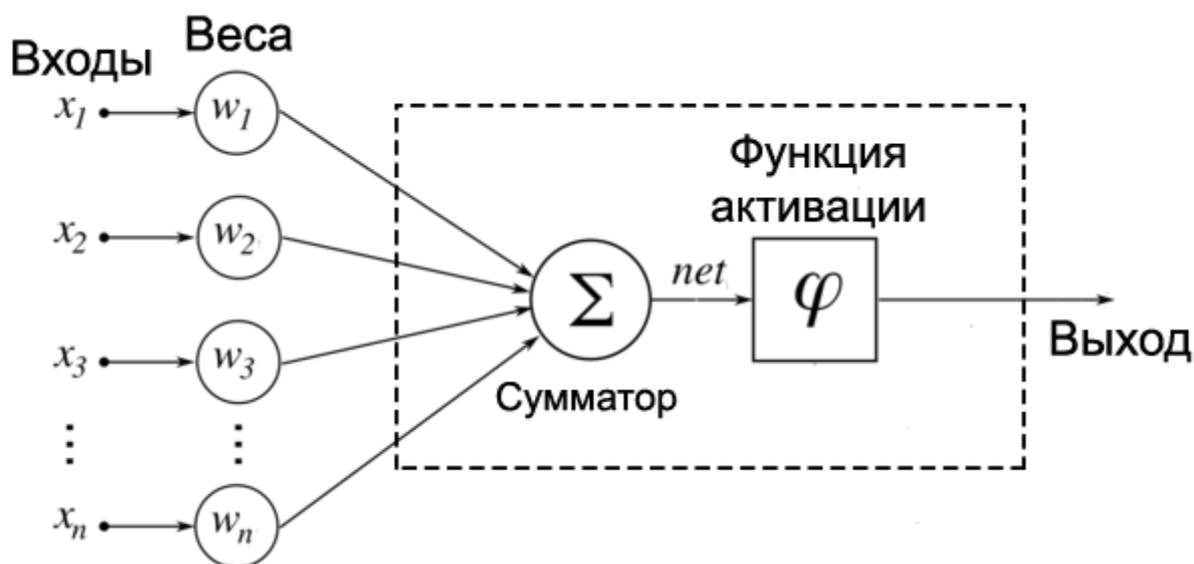


Рисунок 1.2. Схема искусственного нейрона

На рисунке 1.2. изображена схема искусственного нейрона, у которого есть n входов x_i . Каждый вход обладает весом w_i , на который умножается сигнал, проходящий по связи. Далее, произведение $x_i \cdot w_i$ суммируется. Полученную сумму называют net , она будет являться входом для функции активации, которая, в свою очередь, создаёт выходной сигнал.

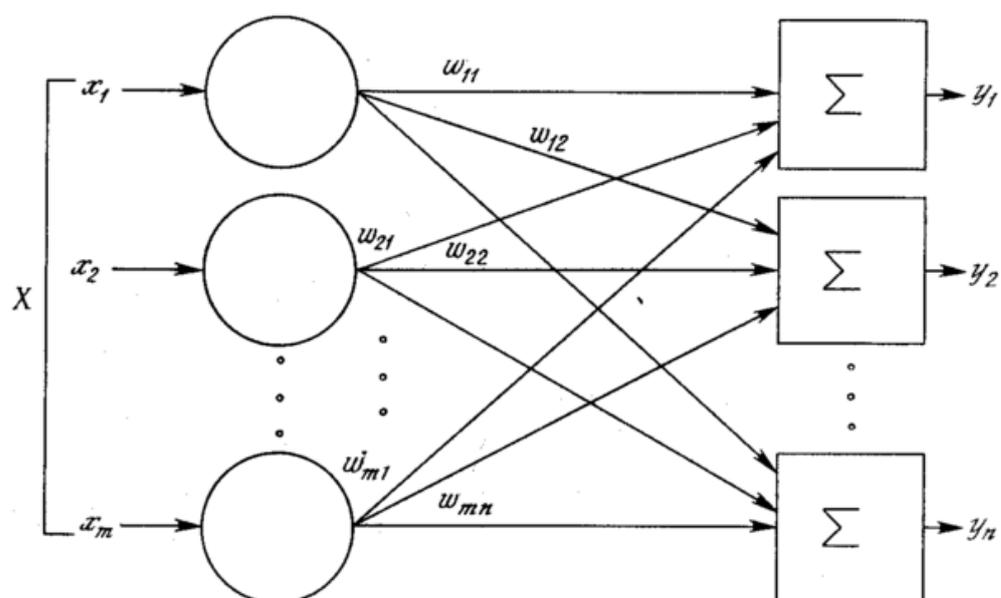


Рисунок 1.3. Схема однослойной нейронной сети

Однослойная нейронная сеть — это нейронная сеть, состоящая из одного слоя нейронов. Каждый нейрон в этом слое получает входные данные, обрабатывает их и сразу передает данные на выход сети. На рисунке 1.3. сигналы x_i поступают на входной слой, затем распределяются на выходные слои нейронов.

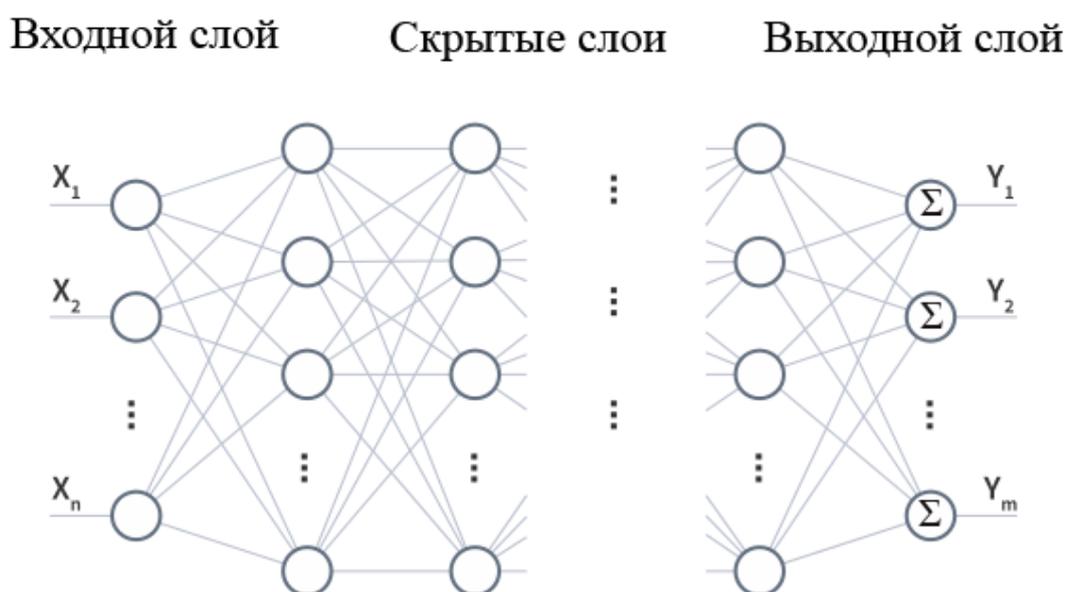


Рисунок 1.4. Схема многослойной нейронной сети

Многослойная нейронная сеть — это нейронная сеть, состоящая из нескольких слоев нейронов. Каждый слой обрабатывает входные данные и передает выходные данные следующему слою. В отличие от однослойной,

многослойная нейронная сеть помимо входного и выходного слоя обладает скрытыми слоями, поэтому информация проходит через несколько уровней обработки, что позволяет ей решать более сложные задачи.

Нейронные сети — это мощный инструмент для решения различных задач в области искусственного интеллекта. Они могут обрабатывать большие объемы данных и автоматически извлекать признаки, что делает их полезными для задач классификации, распознавания образов, обработки естественного языка и генерации новых данных.

Существует множество различных типов нейронных сетей, каждый из которых имеет свои особенности и применение в различных областях.

Прямое распространение является наиболее простым типом нейронной сети и широко используется для решения задач классификации и регрессии. Он состоит из набора входных узлов, скрытых слоев и выходного слоя. Примерами применения этого типа нейронной сети могут быть распознавание рукописных цифр, классификация изображений и прогнозирование цен на акции.

Рекуррентные нейронные сети используются для анализа временных последовательностей, таких как речь и текст. Они позволяют моделировать зависимости между последовательными элементами и имеют широкое применение в задачах машинного перевода, распознавания речи и анализа временных рядов.

Свёрточные нейронные сети используются для анализа изображений и видео. Они могут автоматически извлекать признаки из изображений, что делает их полезными для задач классификации и распознавания объектов. Свёрточные нейронные сети часто используются в компьютерном зрении, например, для распознавания лиц и автомобильных номеров.

Генеративные нейронные сети могут генерировать новые данные на основе имеющегося набора данных. Они используются для создания новых изображений, звуков и текстов. Примерами применения этого типа нейронной сети могут быть генерация фотографий, музыки и текстовых сообщений.

Состязательные нейронные сети состоят из двух нейронных сетей, одна из которых генерирует данные, а другая пытается их классифицировать. Они используются для создания новых изображений и обнаружения мошенничества. Примерами применения этого типа нейронной сети могут быть генерация фальшивых фотографий или обнаружение фальшивых документов.

В зависимости от задачи, которую необходимо решить, может использоваться один или несколько типов нейронных сетей. Например, для распознавания речи может использоваться рекуррентная нейронная сеть в сочетании со свёрточной нейронной сетью для анализа звуковых волн. Для обработки изображений может использоваться свёрточная нейронная сеть в сочетании с генеративной нейронной сетью для создания новых изображений.

Несмотря на то, что нейронные сети имеют много преимуществ, они также имеют и свои ограничения. Например, они могут быть чувствительны к шуму в данных и требуют большого количества обучающих данных для достижения высокой точности. Тем не менее, с развитием технологий нейронные сети становятся все более точными и эффективными, что делает их все более популярными в различных областях.

1.4 Обучение нейронной сети

Обучение нейронной сети – это процесс подбора связей в нейронной сети. Обучение с учителем заключается в последовательном предъявлении нейронной сети векторов наблюдений и последующей корректировке весовых коэффициентов так, чтобы выходное значение как можно ближе соответствовало требуемому отклику.

Алгоритм обратного распространения ошибки (back propagation) — это метод обучения многослойных нейронных сетей, который заключается в изменении весов связей между нейронами на основе разности между выходными значениями сети и ожидаемыми значениями [1]. Алгоритм начинается с передачи входных данных через сеть, после чего вычисляются ошибки на выходном слое:

$$E = \frac{1}{2} \sum_{i=1}^k (y - y')^2 \quad (1.1)$$

где k – число выходных нейронов сети, y – целевое значение, y' – фактическое, полученное на выходе значение.

Затем эти ошибки распространяются обратно через сеть, при этом каждому нейрону присваивается вес ошибки, который определяет его вклад в общую ошибку сети. На основе этих весов и производной функции активации вычисляются корректировки весов связей между нейронами [9], которые позволяют уменьшить ошибку на следующем проходе обучения:

$$\Delta w_{j,i}(n) = -\eta \frac{\partial E}{\partial w_{ij}} \quad (1.2)$$

где $w_{j,i}$ – вес i -ой связи j -ого нейрона, η – параметр скорости обучения.

Алгоритм повторяется до тех пор, пока ошибка не станет достаточно малой или не будет достигнуто максимальное количество итераций.

1.5 Обучение с подкреплением

Обучение с подкреплением (Reinforcement Learning) — это тип машинного обучения, который включает в себя обучение агента принимать решения, взаимодействуя с окружающей средой и получая обратную связь в виде награды или наказания [15]. Цель обучения с подкреплением заключается в том, чтобы научить агента принимать действия, которые максимизируют кумулятивную награду со временем [4].

В постановке задачи для обучения с подкреплением существуют следующие элементы:

1) Среда, которая может принимать какие-либо состояния $s \in S$, где S – пространство состояний. Среда в каждый момент времени может принимать только одно состояние s , которое может быть наблюдаемо неполностью.

2) Агент, который может совершать определённые действия $a \in A$, где A – множество (пространство) действий. Находясь в каждом состоянии, агент может совершить некое действие a .

3) Модель переходов $p(s'|a.s)$, которая является неизвестной – это та модель, по которой переходит данная система. В зависимости от текущего состояния s и совершаемого действия a , осуществляется переход в состояние s' .

4) Политика $\pi(a|s)$, по которой агент совершает действия. В зависимости от s даётся действию a некая вероятность.

5) Награда. Находясь в состоянии $(a|s)$ агент совершает действие a и получает награждение $r(s.a)$.

Простейшая схема алгоритма обучения с подкреплением изображена на рисунке 1.5.

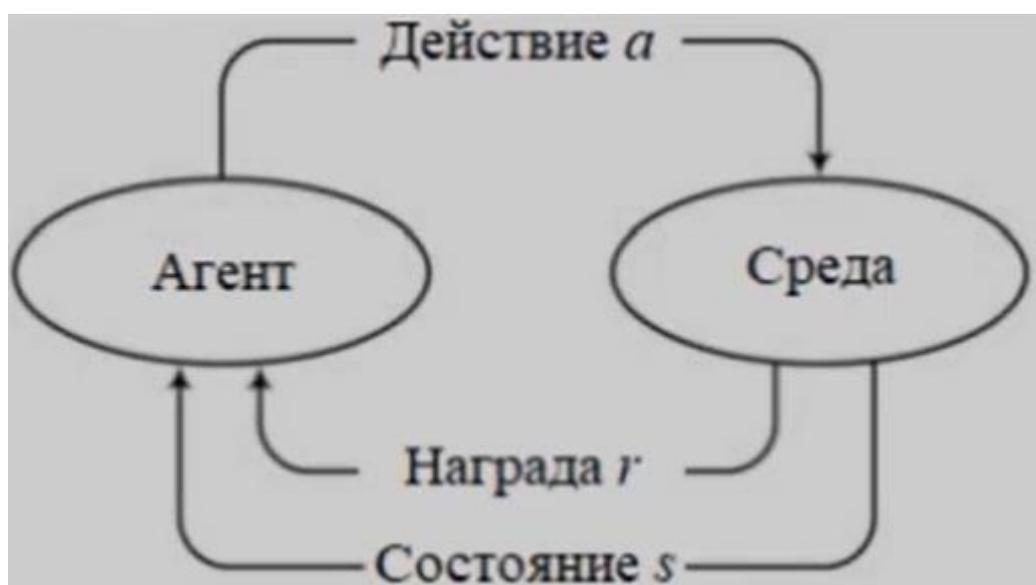


Рисунок 1.5. Алгоритм обучения с подкреплением.

Взаимодействие агента со средой называется эпизодом. Находясь в состоянии s_0 , агент совершает действие a_0 и переходит к состоянию s_1 . Затем, агент совершает действие a_1 и переходит к состоянию s_2 и так далее [9]. Последовательность таких состояний и действий называется траекторией.

Суммарная дисконтированная награда:

$$R_t = r(s_t, a_t) + \gamma r(s_{t+1}, a_{t+1}) + \gamma^2 r(s_{t+2}, a_{t+2}) + \dots \quad (1.3)$$

где $\gamma \in [0; 1]$ – ставка дисконтирования.

Суть задачи в максимизации награды: $R_t \rightarrow \max$.

Для решения данной задачи понадобятся также определённые функции:

1) Функция полезности состояний – это предсказание будущего вознаграждения, используется для оценки состояний.

$$V^\pi(s) = E_\pi[R_t | s_t = s] \quad (1.4)$$

Это математическое ожидание награды R_t по политике π , находясь в данный момент времени в состоянии s .

При переборе всех возможных действий при всех возможных стратегиях можно вычислить уравнение максимума всех возможных цепочек действий:

$$V^*(s) = \max_{\pi} V^\pi(s) \quad (1.5)$$

$$\begin{aligned} V^*(s) &= \max_{\pi} V^\pi(s) = \max_{a_0, a_1, \dots} E[R_0 | s_0 = s] = \\ &= \max_{a_0, a_1, \dots} E \left[r(s_0, a_0) + \gamma E_{p(s'|s_0, a_0)} [R_1 | s_1 = s'] \right] = \\ &= \max_{a_0, a_1, \dots} E \left[r(s_0, a_0) + \gamma E_{p(s'|s_0, a_0)} \underbrace{\max_{a_1, a_2, \dots} E[R_1 | s_1 = s']}_{V^*(s')} \right] = \\ &= \max_{a_0, a_1, \dots} E \left[r(s_0, a_0) + \gamma E_{p(s'|s_0, a_0)} V^*(s') \right] \end{aligned} \quad (1.6)$$

Последнее звено в данном уравнении называется уравнением Беллмана.

Агент должен выполнять такие действия a , чтобы максимизировать $V^*(s)$:

$$\pi^*(a|s) = \arg \max_a \left[r(s_0, a_0) + \gamma E_{p(s'|s_0, a_0)} V^*(s) \right] \quad (1.7)$$

следующем шаге:

$$V_{t+1}^* = \max_a \left[r(s_0, a_0) + \gamma E_{p(s'|s_0, a_0)} V_t^*(s) \right] \quad (1.8)$$

ГЛАВА 2. АЛГОРИТМ ГЛУБОКОГО Q-ОБУЧЕНИЯ

2.1 Q-обучение

Для качественного обучения одной лишь V -функции полезности состояния будет мало. Введём функцию полезности действий [10].

Функция полезности действий $Q(s, a)$ – математическое ожидание отдачи, полученной, начиная с состояния s и выбранного действия a , при выполнении стратегии π :

$$Q^\pi(s, a) = E_\pi[R_t | s_t = s, a_t = a] \quad (2.1)$$

$$Q^*(s, a) = \max_\pi Q^\pi(s, a) \quad (2.2)$$

Стратегия Q -функции:

$$\pi^*(a|s) = \arg \max_a Q^*(s, a) \quad (2.3)$$

Получение уравнения Беллмана для $Q^*(s, a)$:

$$\begin{aligned} Q^*(s, a) &= \max_{a_1, a_2, \dots} \left[r(s, a) + \gamma E_{p(s'|s, a)} E_{s_2, s_3} [R_1 | s_1 = s'] \right] = \\ &= r(s, a) + E_{p(s'|s, a)} \max_{a_1} \underbrace{\max_{a_2, a_3, \dots} E_{s_2, s_3} [R_1 | s_1 = s']}_{Q^*(s', a')} = \\ &= r(s, a) + \gamma E_{p(s'|s, a)} \max_{a'} Q^*(s', a') \end{aligned} \quad (2.4)$$

Данное выражение выполняется для всех состояний s , и для всех действий a .

Необходимо свести ошибку выражения для Q -функции к нулю, или её минимизировать, суммируя по всем действиям a :

$$\frac{1}{|S||A|} \sum_{s, a} \left[Q^*(s, a) - r(s, a) - \gamma E_{p(s'|s, a)} \max_{a'} Q^*(s', a') \right]^2 \rightarrow \min_{Q^*} \quad (2.5)$$

Чтобы минимизировать функцию, нужно двигаться в сторону её антиградиента (в сторону наискорейшего убывания функции). Необходимо посчитать функцию:

$$L = \left[Q^*(s, a) - r(s, a) - \gamma E_{p(s'|s, a)} \max_{a'} Q^*(s', a') \right]^2 \quad (2.6)$$

Далее нужно продифференцировать её по параметру Q^* , и двигаться в отрицательном направлении:

$$Q_{t+1}^*(s, a) = Q_t^*(s, a) - \alpha \nabla_Q * L(Q^*) \quad (2.7)$$

Вычисление градиента даёт следующее выражение:

$$Q_{t+1}^*(s, a) = Q_t^*(s, a) - 2\alpha \left[Q_t^*(s, a) - r(s, a) - \gamma \max_{a'} Q_t^*(s', a') \right] \quad (2.8)$$

Эта формула называется Q -обучением.

2.2. Эпсилон-жадная стратегия

Эпсилон-жадная стратегия — это стратегия выбора действия в алгоритмах обучения с подкреплением, таких как Q -обучение или глубокое Q -обучение. Она заключается в том, что агент выбирает наилучшее действие с вероятностью $(1 - \varepsilon)$ и случайное действие с вероятностью ε .

$$\pi(a|s) = \begin{cases} \arg \max Q(s, a), & \text{с вероятностью } (1 - \varepsilon) \\ \forall \text{ действие,} & \text{с вероятностью } \varepsilon \end{cases} \quad (2.9)$$

$$A_s = \arg \max Q(s, a) \quad (2.10)$$

A_s – оптимальные действия в состоянии s .

Жадная стратегия:

$$\pi(a|s) = \frac{1}{|A_s|} [a \in A_s] \quad (2.11)$$

Эпсилон-жадная стратегия:

$$\pi(a|s) = \frac{1 - \varepsilon}{|A_s|} [a \in A_s] + \frac{\varepsilon}{|A|} \quad (2.12)$$

2.3 Параметризованное Q -обучение

s, a – конечные множества, поэтому с помощью простого Q -обучения нельзя оценивать действия, которые задаются числами. В качестве решения этой проблемы можно параметризовать Q вектором параметров θ - $Q(s, a|\theta)$.

$$Q(s, a|\theta) = r(s, a) + \gamma E_{p(s'|s, a)} \max_{a'} Q(s', a'|\theta) \quad (2.13)$$

$$\frac{1}{|S||A|} \sum_{s, a} \left[Q(s, a|\theta) - r(s, a) - \gamma E_{p(s'|s, a)} \max_{a'} Q(s', a'|\theta) \right]^2 \rightarrow \min_{Q^*} \quad (2.14)$$

$$\theta_{t+1} = \theta_t - 2\alpha \left[Q_t^*(s, a) - r(s, a) - \gamma \max_{a'} Q_t^*(s', a') \right] * \nabla_{\theta} Q^*(s, a|\theta_t) \quad (2.15)$$

Теперь Q -функцию можно аппроксимировать при помощи искусственной нейронной сети. $\frac{\partial Q}{\partial \theta}$

2.4 Глубокое Q-обучение

Глубокое Q -обучение (Deep Q-Learning) — это метод обучения с подкреплением, который использует нейронные сети для оценки функции ценности и принятия решений в задачах, связанных с управлением. Он основан на алгоритме Q -обучения, который позволяет агенту находить оптимальную стратегию действий в среде, максимизируя награду.

Глубокое Q -обучение расширяет Q -обучение, добавляя глубокие нейронные сети для оценки функции ценности. Это позволяет агенту обучаться на более сложных и пространственных данных, чем традиционное Q -обучение. Вместо того, чтобы хранить таблицу значений Q -функции, глубокое Q -обучение использует нейронную сеть для оценки функции ценности.

Процесс обучения глубокого Q -обучения состоит из нескольких шагов: агент взаимодействует со средой, получая состояние и награду; затем он использует нейронную сеть для оценки функции ценности и выбирает действие на основе наилучшей оценки. После этого происходит обновление весов нейронной сети на основе полученной награды и нового состояния.

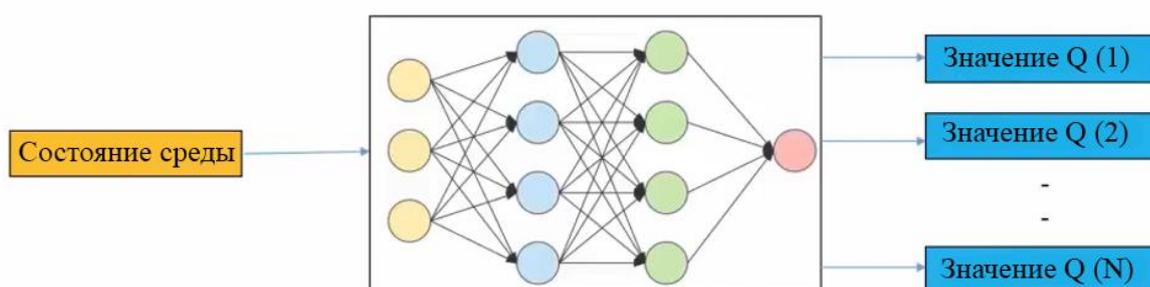


Рисунок 2.1. Глубокое Q-обучение

На рисунке 2.1. изображено, как в нейронную сеть подаётся состояние среды, и после вычислений на выходе получается массив действий Q , среди которых можно выбрать лучшее на основе наибольшей оценки.

ГЛАВА 3. СОЗДАНИЕ НЕЙРОННОЙ СЕТИ И РЕАЛИЗАЦИЯ АЛГОРИТМА ГЛУБОКОГО Q-ОБУЧЕНИЯ ДЛЯ УПРАВЛЕНИЯ ТЕМПОМ ВВОДА СКВАЖИН НА МЕСТОРОЖДЕНИИ

3.1 Обработка данных

Для произведения расчёта экономических показателей месторождения был использован модуль экономической оценки месторождения NumEx2.

Модуль экономической оценки принимает на вход гидродинамическую модель месторождения и её параметры, производится расчёт на N лет, на выход выводятся экономические показатели, которые были получены в результате расчёта.

Так как цель работы в автоматизации, и исключения влияния человеческого фактора при выборе стратегии управления темпом ввода скважин, нужно запускать расчёт в модуле не вручную, а подключаться к ней через консоль при помощи команды `os.system` с заданной гидродинамической моделью месторождения.

Для решения поставленной задачи необходимо подавать на вход модуля экономической оценки месторождения одномерный массив размера M , где M – количество добывающих скважин. Такой массив будет описывать когда, в каком порядке и в какой момент времени будет вводиться в эксплуатацию конкретная скважина, это позволит пользователю, либо нейронной сети управлять темпом ввода скважин.

Модуль на выходе выводит массив экономических показателей в формате `.txt`. Обработка полученных данных была осуществлена на языке Python с использованием библиотек `Pandas` и `NumPy`, так как этот язык программирования позволяет качественно работать с данными, а также их обрабатывать и анализировать при помощи вышеперечисленных библиотек. В качестве основного экономического параметра был выбран Net Present Value (NPV) – чистый дисконтированный доход с учётом инфляции. Этот параметр и будет являться наградой R в алгоритме глубокого Q -обучения.

3.2 Создание нейронной сети

В данной работе создание нейронной сети осуществлялось на языке Python с использованием библиотеки Tensorflow.Keras.

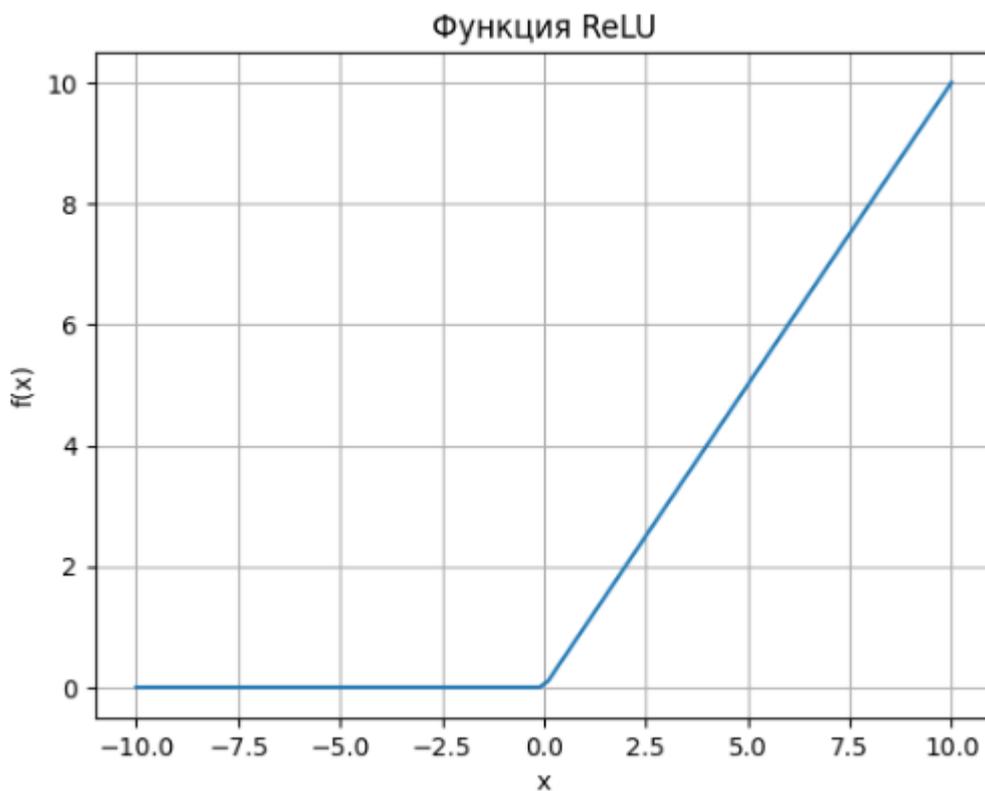
Необходимо избежать переобучения нейронной сети и трат больших мощностей компьютера, но при этом оставить достаточное качество обучения. Поэтому было решено, что оптимальным вариантом будет трёхслойная нейронная сеть, в каждом из слоёв по 64 нейрона. Входной слой содержит в себе $\frac{Y*360}{30}$ нейронов, где Y – количество лет для расчёта в модуле экономической оценки. Такое количество обусловлено тем, что на выходе модуль выводит экономические показатели, в частности, NPV, каждые 30 дней, и единицей измерения будут именно дни, и награду нейронная сеть будет, соответственно, получать каждые 30 дней.

Выходной слой будет содержать в себе K нейронов, где K – количество добывающих скважин на месторождении. Это обусловлено тем, что нейронная сеть будет выводить на выходе действия a , то есть, дни, в которые будут введены в эксплуатацию конкретные скважины.

Функции активации были выбраны следующим образом:

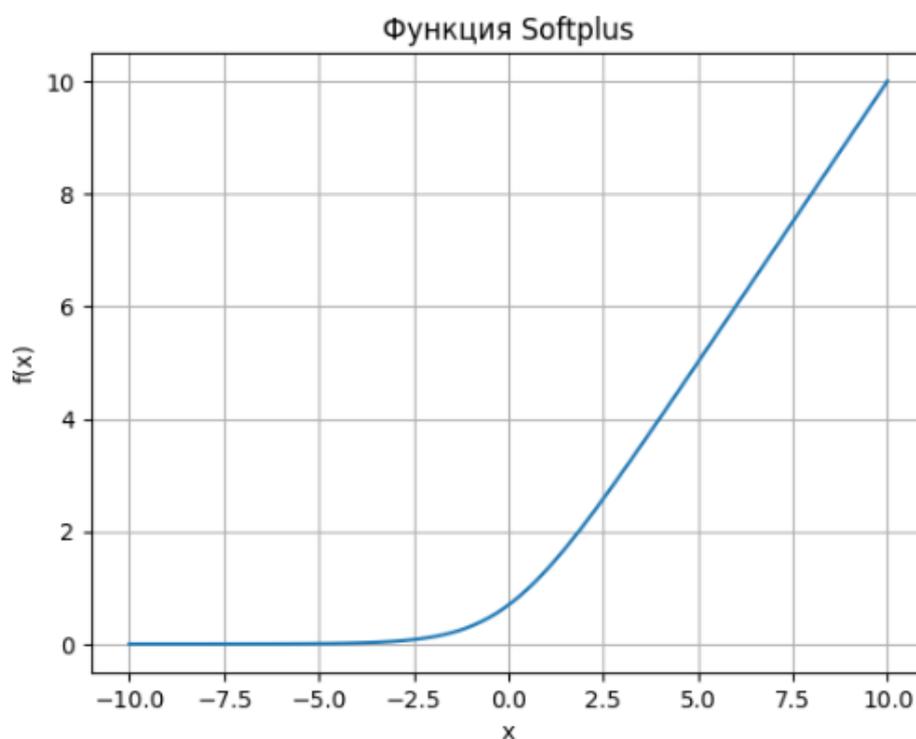
1. На входном и скрытых слоях функция активации *ReLU*:

$$f(x) = \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases} \quad (3.1)$$

Рисунок 3.1. Функция активации *ReLU*

2. На выходном слое использована функция активации *Softplus*, её можно рассматривать как гладкую версию функции *ReLU*:

$$f(x) = \log(1 + e^x) \quad (3.2)$$

Рисунок 3.2. Функция активации *Softplus*

Такой выбор функций активаций обусловлен тем, что значение функции в данной задаче не может быть меньше нуля, так как значения дней, в которые вводятся скважины в эксплуатацию, могут быть только положительными.

В качестве оптимизатора для искусственной нейронной сети был выбран оптимизатор Adam.

Adam (Adaptive Moment Estimation) — это оптимизатор для нейронных сетей, который использует комбинацию градиентного спуска и метода моментов. Он был предложен в 2014 году и является одним из наиболее популярных оптимизаторов в глубоком обучении [12].

Adam адаптивно изменяет скорость обучения для каждого параметра на основе истории градиентов и моментов. Это позволяет ему эффективно справляться с различными типами функций потерь и скоростями обучения.

Основные преимущества Adam включают:

- Быструю скорость сходимости
- Низкую вероятность застревания в локальных минимумах
- Адаптивную скорость обучения для каждого параметра
- Эффективную работу с большими наборами данных

В качестве функции потерь была выбрана функция Хьюбера.

Функция потерь Хьюбера используется в глубоком Q -обучении для оценки ошибки агента при обновлении его Q -функции. Она используется вместо среднеквадратичной ошибки, чтобы уменьшить влияние выбросов на обучение. Функция потерь Хьюбера имеет вид:

$$L = \begin{cases} \frac{1}{2}(Q_t - Q_p)^2, & |Q_t - Q_p| \leq \delta \\ \delta|Q_t - Q_p| - \frac{1}{2}\delta^2, & |Q_t - Q_p| > \delta \end{cases} \quad (3.3)$$

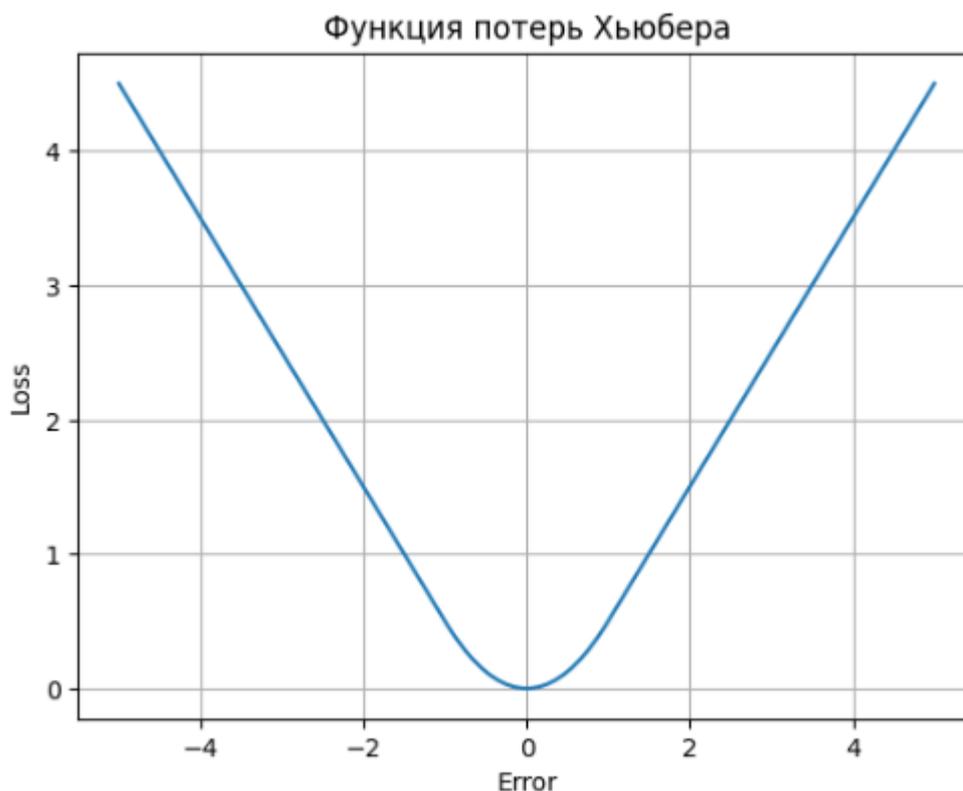


Рисунок 3.3. Функция потерь Хьюбера

3.3 Реализация алгоритма глубокого Q -обучения для управления темпом ввода скважин на месторождении

Принимая во внимание теорию, описанную во второй и третьей главах, можно реализовать алгоритм глубокого Q -обучения.

Средой s будет являться нефтяное месторождение. Агентом будет являться пользователь, или искусственная нейронная сеть, которая выбирает время для ввода в эксплуатацию конкретные добывающие скважины, тем самым совершая действие a [11]. Наградой $R(s, a)$ будет являться Net Present Value (NPV) – чистый дисконтированный доход с учётом инфляции, она будет зависеть от состояния и действий, совершаемых в среде.

Программная реализация алгоритма [7] включает в себя соединение к модулю экономической оценки через консоль с загрузкой гидродинамической модели месторождения, инициализацию функции для обучения, функцию обработки данных с модуля экономической оценки месторождения, создание модели нейросети, определение функции для выполнения действия в среде (ввода скважин в эксплуатацию) и получения следующего состояния и

вознаграждения, импорт и экспорт данных для модуля экономической оценки месторождения, обучение нейронной сети на нескольких эпизодах [8], аппроксимацию Q -функции при помощи нейронной сети, вывод результатов.

ГЛАВА 4. ТЕСТИРОВАНИЕ АЛГОРИТМА И АНАЛИЗ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ

4.1. Гидродинамическая модель реального месторождения

Тестирование алгоритма проводилось на гидродинамической модели реального месторождения (Оренбургское месторождение), представленной на рисунке 4.1:

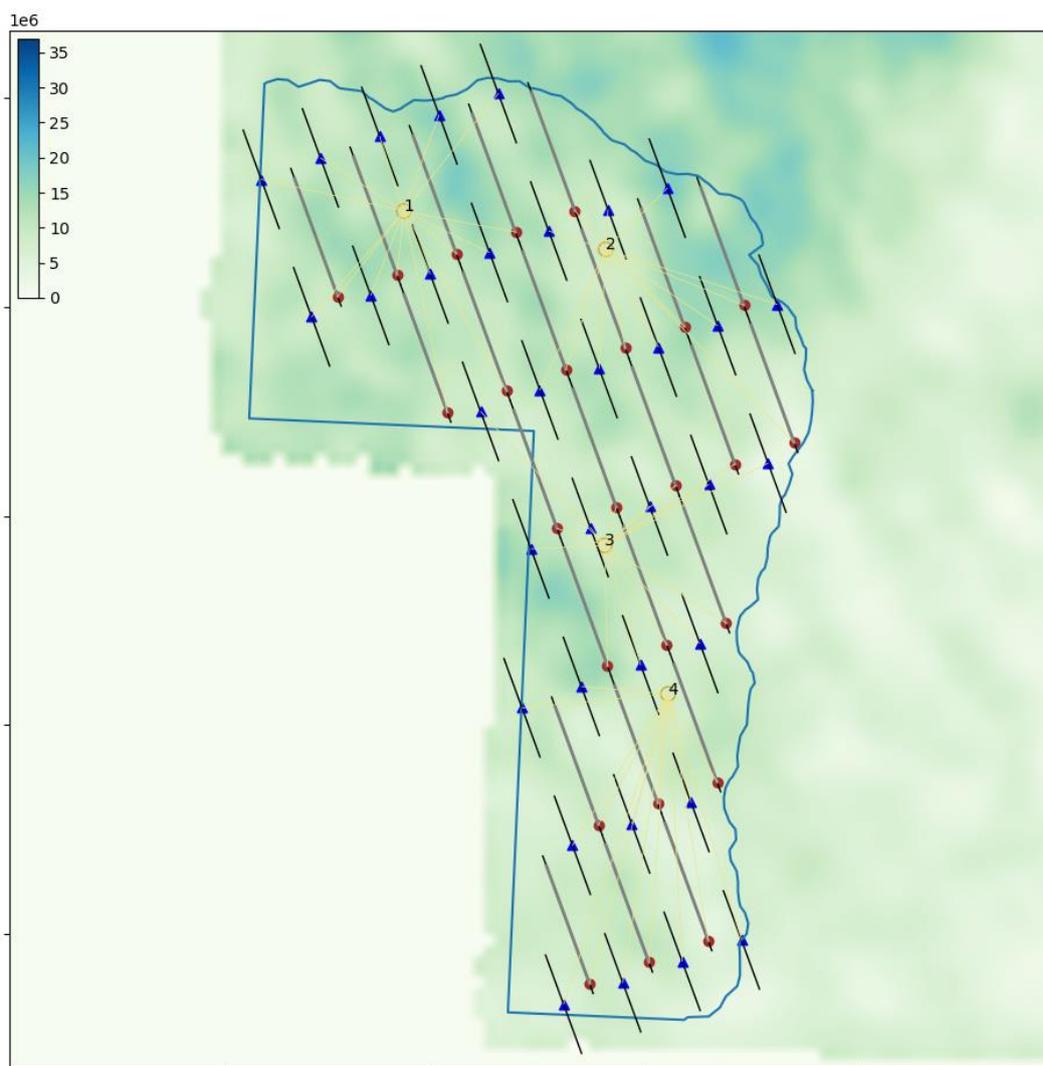


Рисунок 4.1. Гидродинамическая модель месторождения

4.2. Тестирование алгоритма и анализ траты вычислительных мощностей

Расчёт экономических показателей данного месторождения на 10 лет производится в среднем 2 минуты. Для алгоритма нужно произвести множество расчётов для обучения и подбора оптимальной стратегии. В данной работе количество расчётов (рассматриваемых нейронной сетью эпизодов) равно 25. Также для алгоритма требуется некоторое время для обучения нейронной сети

и аппроксимации Q -функции. При запуске алгоритма на процессоре Intel Core i7-10510U время вычислений 25 эпизодов составляет 1 час 15 минут.

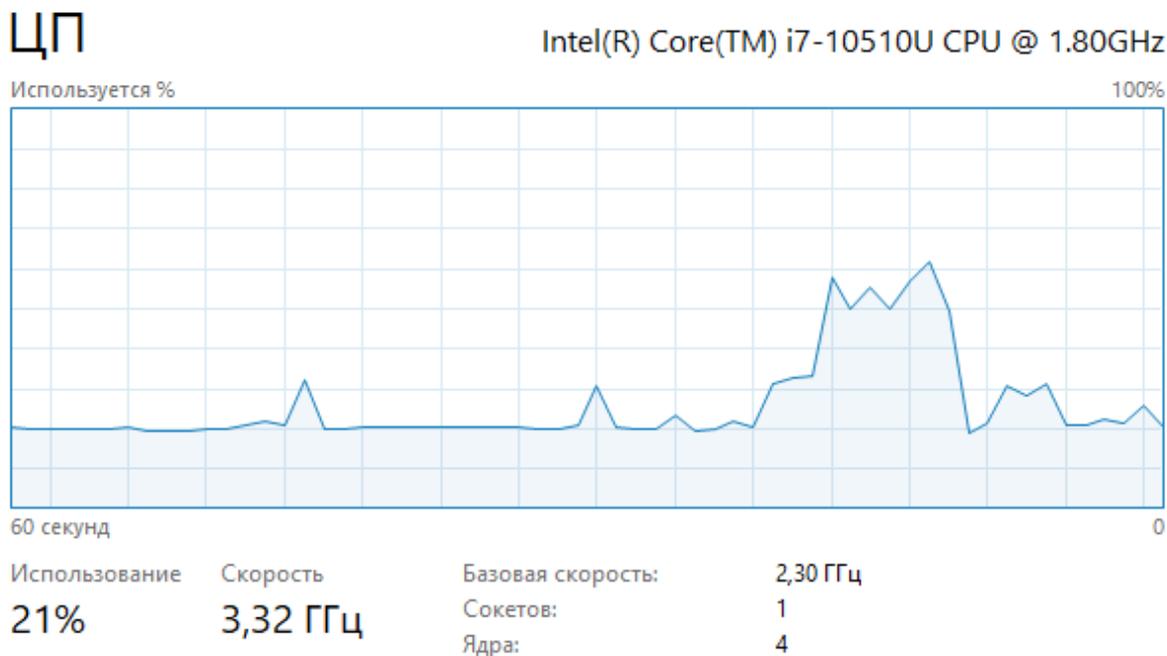


Рисунок 4.2. Затраты вычислительных мощностей при запуске алгоритма

Во время расчёта в модуле экономической оценки месторождения средняя нагрузка на процессор составляет 21%. Во время обучения нейронной сети и аппроксимации Q -функции нагрузка достигает значения 50%.

4.3. Результаты

В ходе расчётов, проведённых алгоритмом, была получена следующая стратегия для управления темпом ввода скважин:

Таблица 4.1.

Время запуска добывающих скважин

Имя скважины	доб(1)/наг(2)	Время запуска (день)	куст
1р	1	750	2
2р	1	450	2
3р	1	990	3
4р	1	510	2
5р	1	1230	3
6р	1	1050	3
7р	1	690	2
8р	1	570	2
9р	1	1380	4
10р	1	1110	3
11р	1	930	3

12р	1	630	2
13р	1	120	1
14р	1	1620	4
15р	1	1560	4
16р	1	870	3
17р	1	1170	3
18р	1	300	1
19р	1	0	1
20р	1	1320	4
21р	1	1500	4
22р	1	240	1
23р	1	180	1
24р	1	1440	4
25р	1	60	1

Таблица 4.2.

Время запуска нагнетательных скважин

Имя скважины	доб(1)/наг(2)	Время запуска (день)	Куст
1i	2	600	2
2i	2	1290	3
3i	2	540	2
4i	2	480	2
5i	2	1200	3
6i	2	660	2
7i	2	840	2
8i	2	1260	3
9i	2	960	3
10i	2	720	2
11i	2	780	2
12i	2	150	1
13i	2	1680	4
14i	2	1470	4
15i	2	1140	3
16i	2	1020	3
17i	2	810	2
18i	2	90	1
19i	2	30	1
20i	2	1530	4
21i	2	1710	4
22i	2	1350	4
23i	2	900	3
24i	2	1080	3
25i	2	330	1
26i	2	270	1
27i	2	1590	4

28i	2	1650	4
29i	2	1410	4
30i	2	210	1
31i	2	360	1
32i	2	1740	4
33i	2	420	1
34i	2	390	1

При использовании данной стратегии были получены следующие результаты чистого дисконтированного дохода с учётом инфляции:

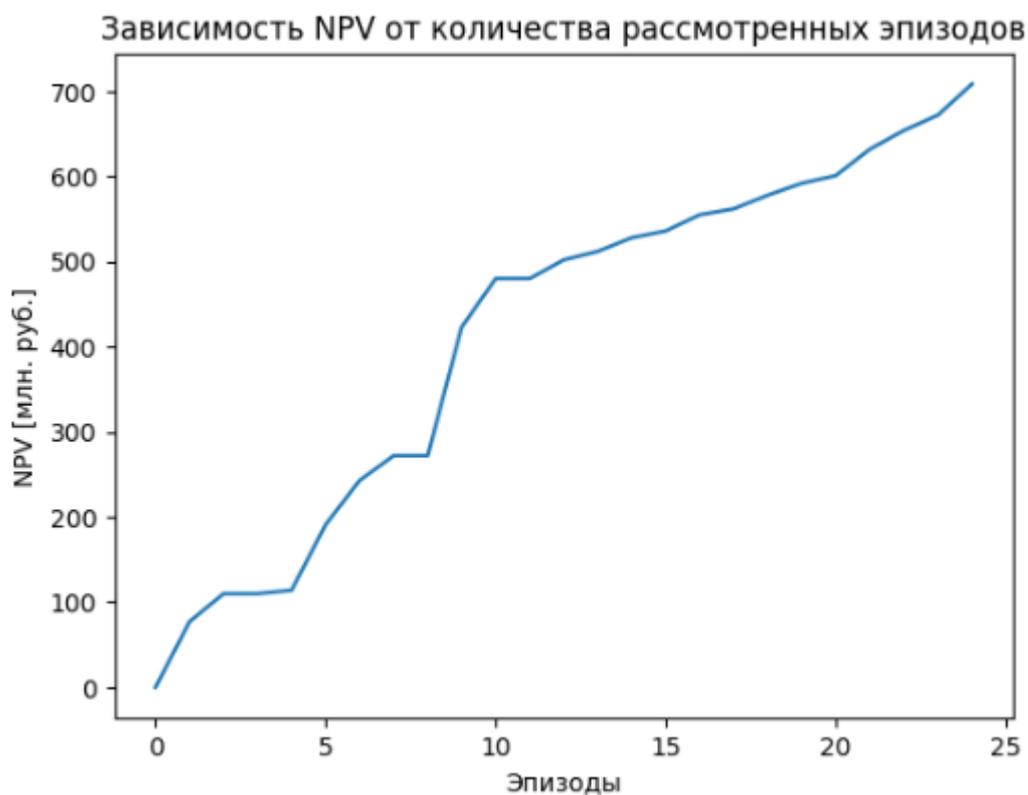


Рисунок 4.3. NPV

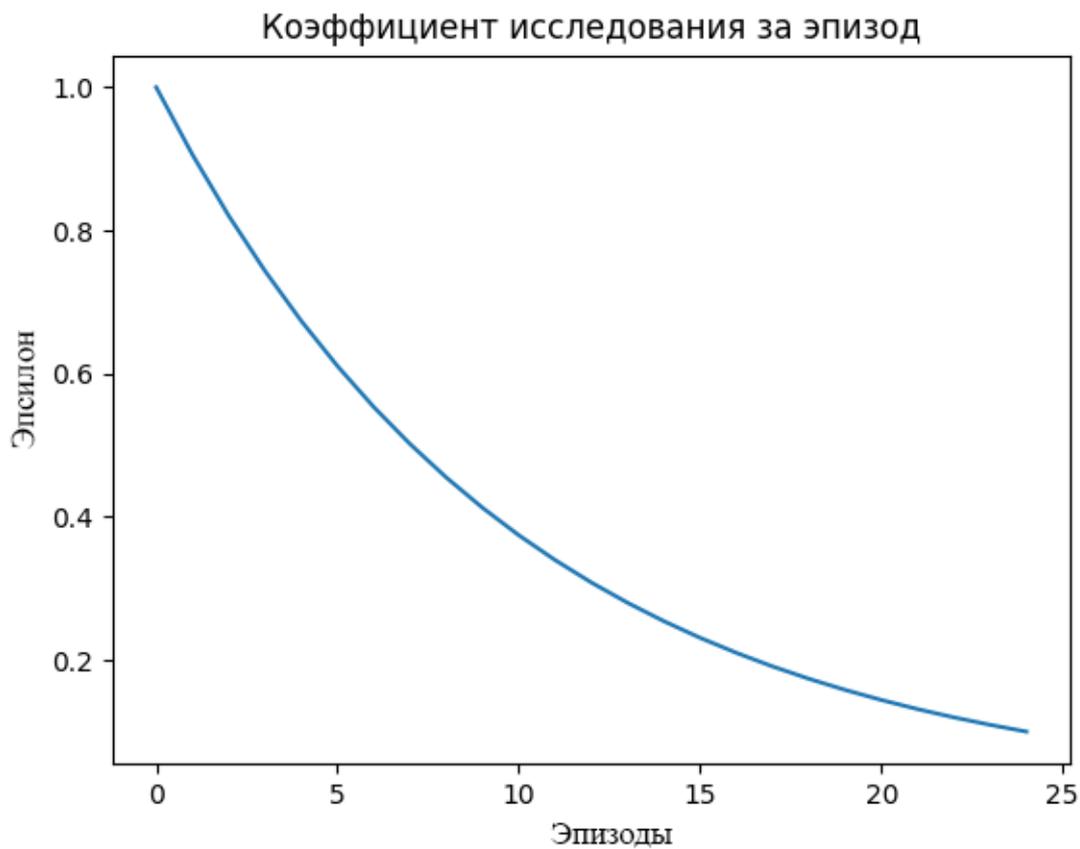
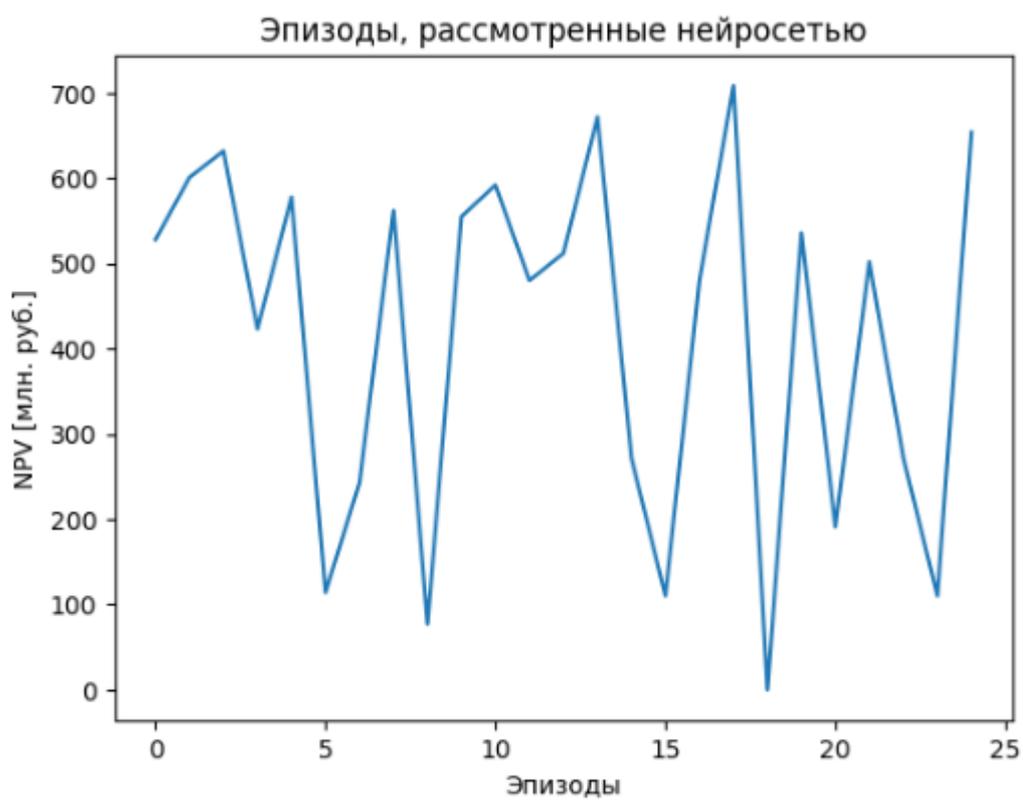
Рисунок 4.4. Коэффициент исследования ε 

Рисунок 4.5. Рассмотренные эпизоды

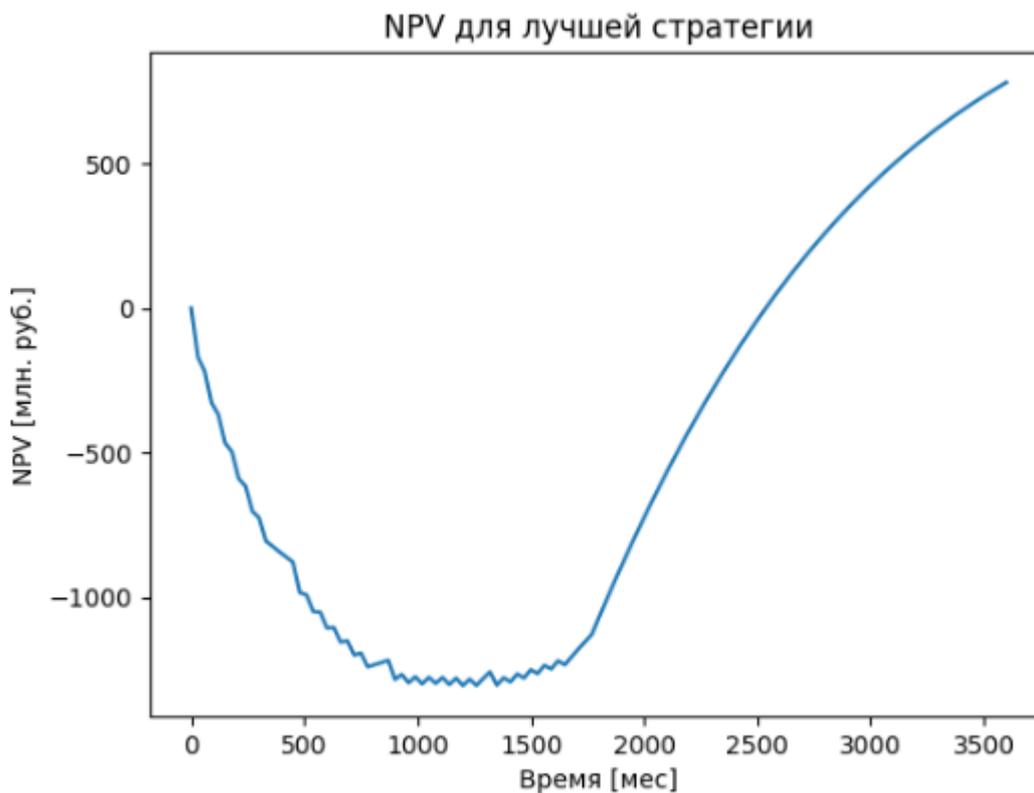


Рисунок 4.6. NPV для лучшей стратегии

При помощи алгоритма удалось увеличить NPV с 77 млн руб. (случайная стратегия) до 708 млн руб. (оптимальная стратегия для расстановки скважин) при расчёте на 10 лет для данной модели месторождения, процесс роста NPV с течением времени показан на рисунке 4.3.

На рисунке 4.4. показано уменьшение коэффициента исследования ε с течением времени. Это говорит о том, что в первые несколько эпизодов нейронная сеть генерировала случайную стратегию для темпа ввода скважин, но с течением времени среда исследовалась всё лучше и лучше, и нейронная сеть осознанно генерировала различные оптимальные стратегии.

Рисунок 4.5. иллюстрирует различные эпизоды, которые рассматривала нейросеть. В них алгоритм пробовал разные стратегии, и в конце среди рассмотренных стратегий алгоритмом была выбрана оптимальная стратегия, при которой NPV было максимально.

Далее был запущен расчёт в модуле экономической оценки месторождения вручную с использованием полученной стратегии:

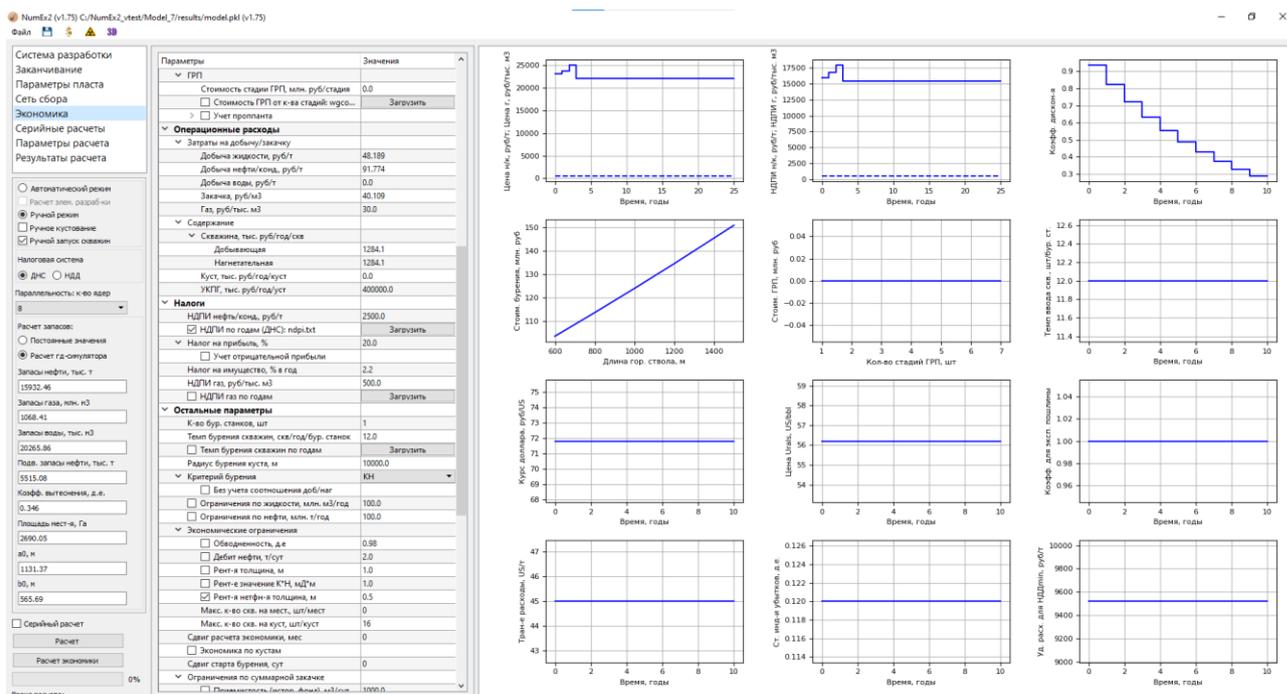


Рисунок 4.7. Интерфейс модуля экономической оценки месторождения

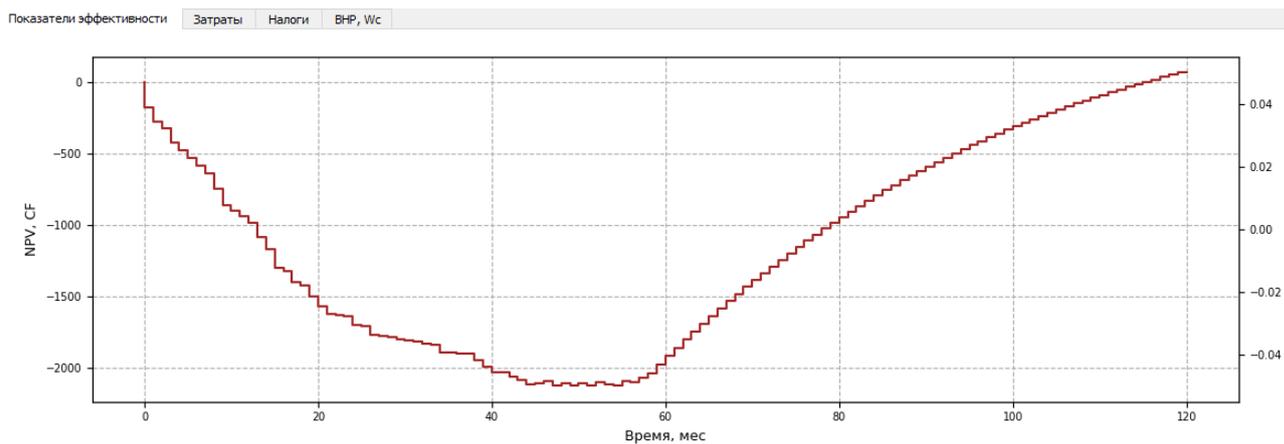


Рисунок 4.8. Показатели эффективности стратегии

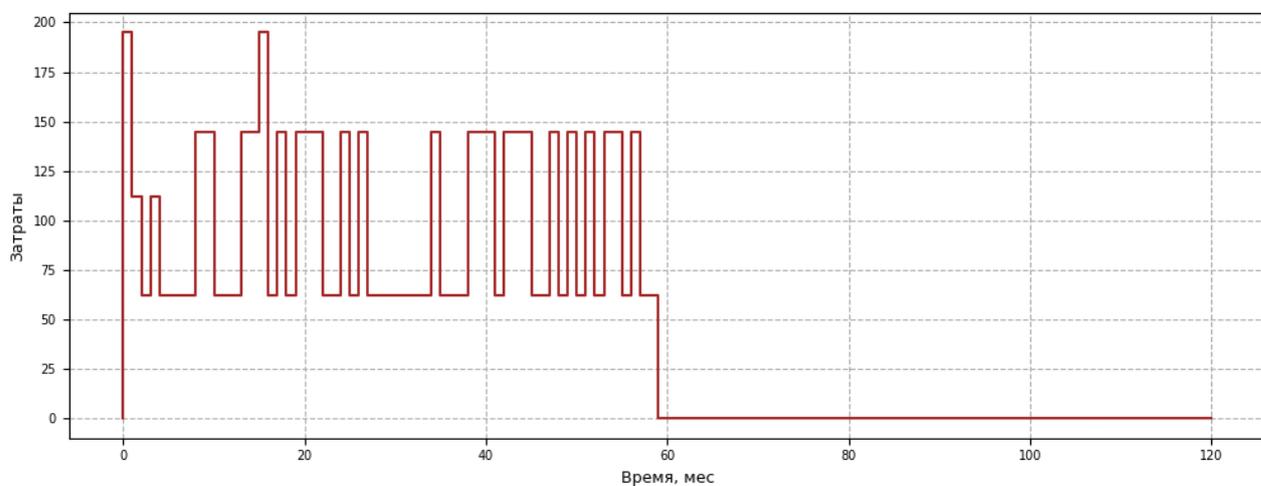


Рисунок 4.9. Затраты, млн. руб

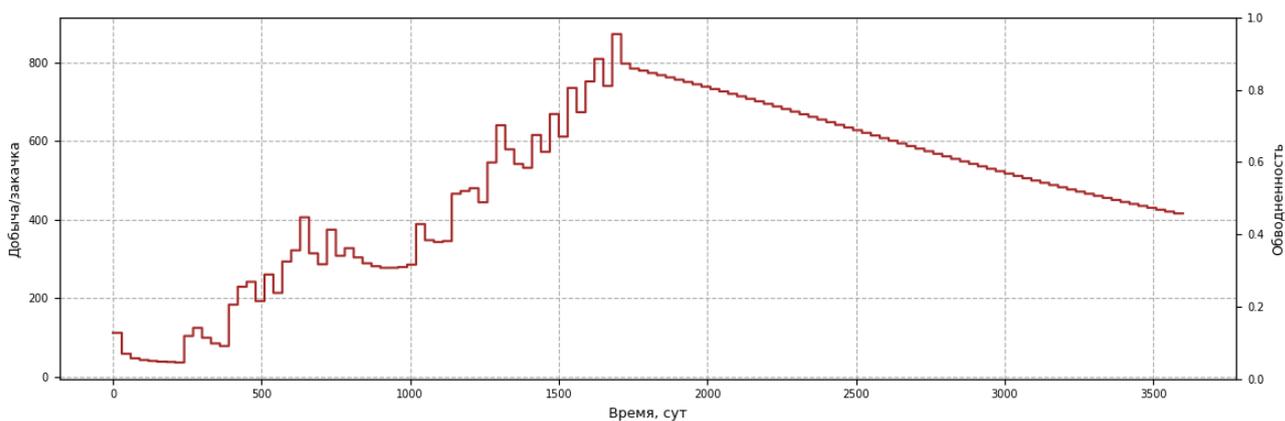


Рисунок 4.10. Дебит нефти, т/сут

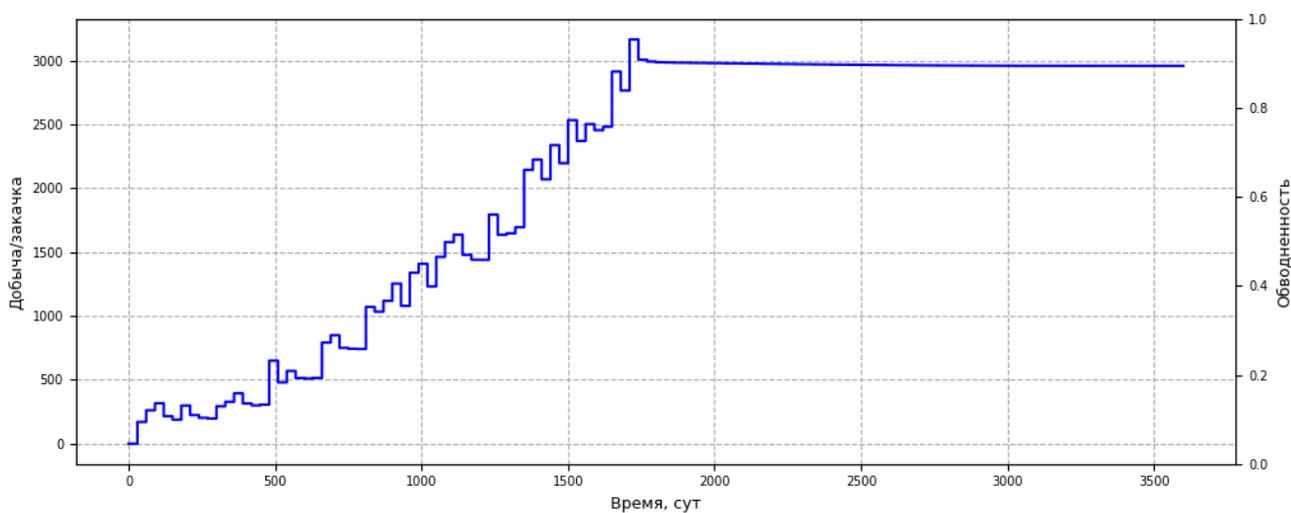
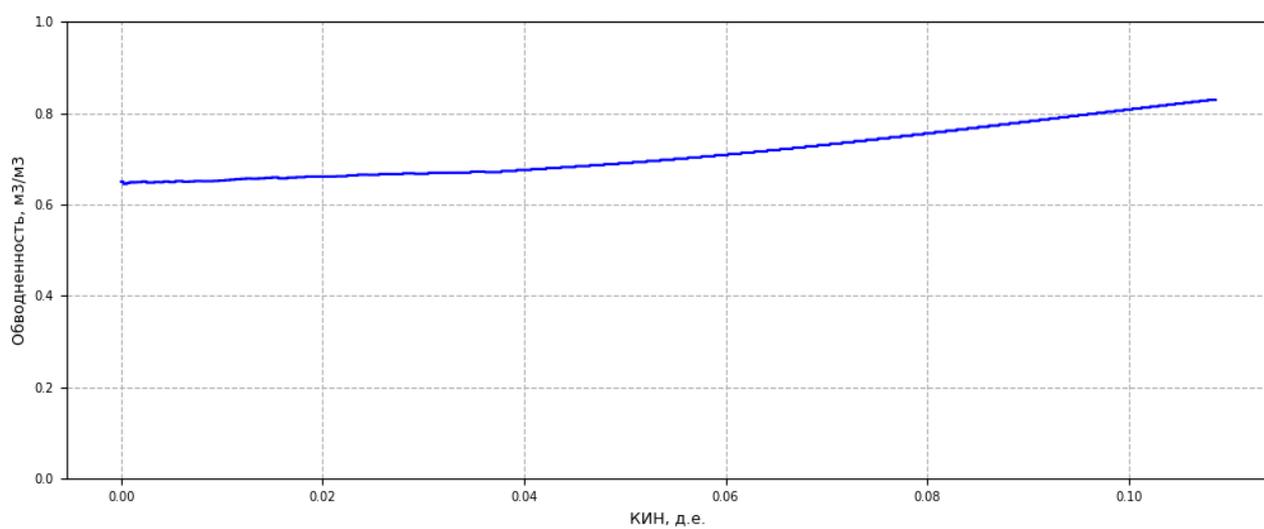
Рисунок 4.11. Закачка воды, м³/сут

Рисунок 4.12. Характеристика вытеснения

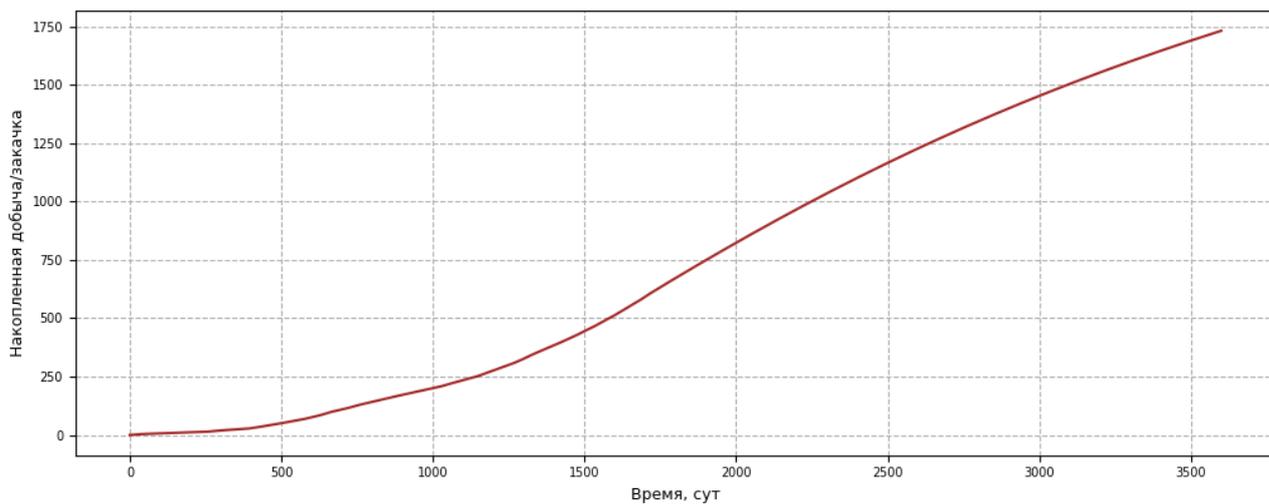


Рисунок 4.13. Накопленная нефть, тыс. т

Используя оптимальную стратегию, выработанную алгоритмом в модуле экономической оценки месторождения, интерфейс которого представлен на рисунке 4.6, можно получить данные об NPV за 120 месяцев (рисунок 4.7) и показатели дебита нефти на рисунке 4.8.

Рисунок 4.7 иллюстрирует то, что все скважины были введены в 59 месяц, поэтому до 59 месяца NPV лишь уменьшался. Далее, после того как все скважины были введены, начался рост NPV, так как больше скважин вводить не приходилось, и расходы шли только на содержание скважин, а они в свою очередь приносили прибыль.

ЗАКЛЮЧЕНИЕ

В результате выполнения выпускной квалификационной работы были изучены теоретические основы нейронных сетей и их принцип работы. Для выполнения поставленной цели была выбрана трёхслойная нейронная сеть и алгоритм глубокого Q -обучения. Программная реализация нейронной сети и алгоритма была осуществлена на языке программирования Python [2] с использованием библиотек Tensorflow.Keras, Pandas, NumPy, Matplotlib, os [3]. Нейронная сеть была обучена на 25 эпизодах, в которых был произведён расчёт на 10 лет для гидродинамической модели реального месторождения в модуле экономической оценки месторождения NumEx2 с предварительным анализом и обработкой данных, получаемых с данного модуля. После рассмотрения 25 эпизодов и расчёта длительностью 1 час 15 минут алгоритмом была выбрана оптимальная стратегия для управления темпом ввода скважин, и получен окончательный результат.

Для анализа результатов можно сравнить результат расчёта NPV при случайной стратегии в 77 млн руб., представленной в одной из первых эпизодов, рассмотренных алгоритмом и окончательный результат в 708 млн рублей, полученный с использованием оптимальной. Чистый дисконтированный доход с учётом инфляции удалось увеличить примерно в 10 раз, что показывает эффективность разработанного алгоритма и корректность теории, описывающей искусственные нейронные сети и алгоритм глубокого Q -обучения. Однако, недостатком данного алгоритма является долгое время обучения. Также существует множество других алгоритмов для обучения с подкрепления, такие как Policy Optimization, Policy Gradient, Proximal Policy Optimization, Actor-Critic, Twin-Delayed Deep Deterministic, Deep Deterministic Policy Gradient и другие. Но они либо требуют больших вычислительных мощностей, либо предназначены для задач иного типа, поэтому для конкретно данной в выпускной квалификационной работе был выбран алгоритм Deep Q-Learning.

Преимуществом разработанного алгоритма является частичное отсутствие влияния человеческого фактора при выборе оптимальной стратегии для управления темпом ввода скважин в эксплуатацию на месторождения, так как эпизоды рассматривает нейронная сеть, и спустя некоторое время расчётов она выбирает оптимальный алгоритм, при помощи которого можно максимизировать экономические параметры месторождения на этапе её выбора системы разработки.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Галушкин А. И. Нейронные сети: Основы теории — Москва: «Горячая линия-Телеком», 2017. — с. 43-49.
2. Плас Дж. Вандер. Python для сложных задач: наука о данных и машинное обучение. — СПб.: «Питер», 2018. — с. 331-375.
3. Рашка С., Мирджалили В. Python и машинное обучение. Машинное и глубокое обучение с использованием Python, scikit-learn и TensorFlow. — СПб.: ООО «Диалектика», 2019.
4. Ричард С. Саттон, Эндрю Дж. Барто Обучение с подкреплением: Введение. 2-е изд. / пер. с англ. А. А. Слинкина. — М.: «ДМК Пресс», 2020. — с. 152-174.
5. Чару Аггарвал Нейронные сети и глубокое обучение. Учебный курс — ООО «Диалектика», 2020. — с. 569-580.
6. Чоловский И. П. Спутник нефтегазопромыслового геолога: Справочник/ С 74 — М.: «Недра», 1989.— с. 199-229.
7. Ethem Alpaydin Introduction to Machine Learning – MIT Press, 2004 – p. 332 – 337.
8. Géron Aurélien, Hands on Machine Learning with Scikit Learn, Keras and Tensorflow Concepts, Tools and Techniques to Build Intelligent Systems // O'Reilly Media – 2019 – P. 114-115.
9. Hecht-Nielsen R. Theory of the backpropagation neural network // Neural networks for perception. – Academic Press, 1992. – P. 65-93.
10. Nazia Habib Hands-On Q-Learning with Python: Practical Q-learning with OpenAI Gym, Keras, and TensorFlow. — Packt Publishing, 2019. – 212 с.
11. Achiam J., Joshua P. Spinning Up in Deep Reinforcement Learning, 2018.
12. Breiman. L, Bagging predictors, Machine Learning. 24 (2): 123–140, 1996.
13. Большая энциклопедия нефти и газа, Темп – ввод – скважина, URL: <https://www.ngpedia.ru/id503239p1.html>. Дата обращения: 02.05.2023.
14. Министерство энергетики Российской Федерации: официальный сайт. – URL: <https://minenergo.gov.ru/>. Дата обращения 02.05.2023.

15. IBM, Cloud Education, Machine Learning, URL: <https://www.ibm.com/cloud/learn/machine-learning>. Дата обращения: 10.04.2023.

16. IBM, Prepare Your Data for AI and Data Science, URL: <https://www.ibm.com/garage/method/practices/code/data-preparation-ai-data-science/>. Дата обращения: 11.05.2023.