

Министерство науки и высшего образования Российской Федерации
Санкт-Петербургский политехнический университет Петра Великого
Институт прикладной математики и механики
Высшая школа теоретической механики

Работа допущена к защите

Директор высшей школы

_____ А.М. Кривцов

«___» _____ 2021 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

ПОСТРОЕНИЕ МЕТАМОДЕЛИ ПРИТОКА ФЛЮИДА К ВЕРТИКАЛЬНОЙ СКВАЖИНЕ ПО ДАННЫМ ИЗ ГИДРОДИНАМИЧЕСКОГО СИМУЛЯТОРА

по направлению подготовки

01.03.03 Механика и математическое моделирование

профиль

01.03.03_01 Механика и математическое моделирование сред с
микроструктурой

Выполнил

студент гр. 3630103/70101

А.А. Муравцев

Руководитель

доцент, к. т. н.

Е.Ю. Журавлёва

Санкт-Петербург

2021

САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО
Институт прикладной математики и механики

УТВЕРЖДАЮ
Директор высшей школы

А.М. Кривцов

« » 2021 г.

ЗАДАНИЕ
по выполнению выпускной квалификационной работы

студенту Муравцеву Александру Алексеевичу, группы 3630103/70101

1. Тема работы: Построение метамодели притока флюида к вертикальной скважине по данным из гидродинамического симулятора
2. Срок сдачи студентом законченной работы: 14 июня 2021 года
3. Исходные данные по работе: статьи по теме работы; программа гидродинамического моделирования ECLIPSE Blackoil; пакет прикладных программ MATLAB; дистрибутив языка программирования Python: Anaconda
4. Содержание работы (перечень подлежащих разработке вопросов):
Построить компьютерную модель пласта в гидродинамическом симуляторе
Построить метамодели притока флюида к вертикальной скважине с помощью искусственных нейронных сетей и градиентного бустинга на основе деревьев регрессии
Провести визуализацию точности прогноза построенных метамоделей для двух сценариев разработки нефтяной залежи
Применить построенную метамодель для оценки неопределенности суммы дебитов нефти методом Монте-Карло
5. Перечень графического материала (с указанием обязательных чертежей): Отсутствуют
6. Консультанты по работе: Отсутствуют
7. Дата выдачи задания 1 февраля 2021 года

Руководитель ВКР _____ Е.Ю. Журавлёва
(подпись) инициалы, фамилия

Задание принял к исполнению 1 февраля 2021 года
(дата)

Студент _____ А.А. Муравцев
(подпись) инициалы, фамилия

РЕФЕРАТ

На 95 с., 44 рисунка, 3 таблицы, 15 приложений.

МАШИННОЕ ОБУЧЕНИЕ, ЦИФРОВАЯ МОДЕЛЬ ПЛАСТА, ЛАТИНСКИЙ ГИПЕРКУБ, МНОГОСЛОЙНЫЙ ПЕРЦЕПТРОН, ГРАДИЕНТНЫЙ БУСТИНГ

В данной работе проводилось построение и исследование метамодели притока флюида к вертикальной скважине. Представлено два подхода к построению метамодели: с помощью градиентного бустинга на основе деревьев регрессии и с помощью искусственных нейронных сетей. В гидродинамическом симуляторе построена компьютерная модель пласта, на которой проводились эксперименты с целью генерации данных для обучения метамодели. Проведена визуализация точности прогноза обученных нейронных сетей. Метамодель применена для оценки неопределенности суммы дебитов нефти методом Монте-Карло.

THE ABSTRACT

95 pages, 44 pictures, 3 tables, 15 appendices.

MACHINE LEARNING, DIGITAL RESERVOIR MODEL, LATIN HYPERCUBE, MULTILAYER PERCEPTRON, GRADIENT BOOSTING

In this work, the construction and study of a metamodel of fluid inflow to a vertical well was carried out. Two approaches to building a metamodel are presented: using gradient boosting based on regression trees and using artificial neural networks. In the hydrodynamic simulator, a computer model of the reservoir was built, on which experiments were carried out in order to generate data for training the metamodel. Visualization of the forecast accuracy of trained neural networks has been performed. The metamodel is applied to estimate the uncertainty of the sum of oil production rates using the Monte Carlo simulation.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
ГЛАВА 1. ПОДГОТОВКА ДАННЫХ ДЛЯ ОБУЧЕНИЯ МЕТАМОДЕЛИ	7
1.1 Построение гидродинамической модели пласта	7
1.2 Генерация кейсов для расчёта	12
1.3 Считывание данных из выходных файлов	16
ГЛАВА 2. ОБУЧЕНИЕ МЕТАМОДЕЛИ	17
2.1 Нейронная сеть.....	17
2.2 Градиентный бустинг на основе деревьев регрессии	24
ГЛАВА 3. АНАЛИЗ ТОЧНОСТИ ПРОГНОЗА И ВИЗУАЛИЗАЦИЯ РЕЗУЛЬТАТОВ.....	28
3.1 Диаграммы размаха ошибок.....	28
3.2 Сравнение результатов (симулятор/метамодель).....	37
3.3 Визуализация прогноза при изменении одного из входных параметров.....	39
ГЛАВА 4. ОЦЕНКА НЕОПРЕДЕЛЁННОСТИ СУММЫ ДЕБИТОВ МЕТОДОМ МОНТЕ-КАРЛО	44
ЗАКЛЮЧЕНИЕ	47
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	49
ПРИЛОЖЕНИЯ 1-15	52

ВВЕДЕНИЕ

При решении многих современных инженерных задач используется компьютерное моделирование вместо проведения реальных экспериментов. Однако компьютерный эксперимент при моделировании сложной физической системы с несколькими входными и выходными параметрами может занимать длительное время. Встаёт вопрос о сокращении времени расчёта.

Увеличение скорости расчёта особенно актуально при решении задач оптимизации или оценке неопределённости результата расчёта, так как при решении этих задач требуется проводить большое количество экспериментов при разных значениях варьируемых входных параметров рассматриваемой модели.

Для решения подобных задач исследователи используют аппроксимации, которые с высокой точностью воспроизводят результаты исходной модели, но при этом существенно возрастает скорость расчёта. Такие аппроксимации называют метамоделями.

Построение метамоделей осуществляется с помощью алгоритмов машинного обучения (МО) с учителем. В пространстве варьируемых входных параметров генерируется обучающая выборка, для которой проводятся эксперименты на полноценной компьютерной модели. Далее на основе этих данных с помощью инструментов МО проводится аппроксимация исходной модели на всём пространстве варьируемых параметров.

Существует огромное количество инструментов МО для аппроксимации данных: многослойные перцептроны, искусственные нейронные сети радиально-базисных функций, регрессия на основе гауссовских процессов, метод опорных векторов, деревья регрессии, градиентный бустинг на основе деревьев регрессии и т. д.

В данной работе рассмотрены два подхода к построению метамоделей: с помощью искусственных нейронных сетей [14] и с помощью градиентного бустинга на основе деревьев регрессии [9].

Проводится построение метамодели притока флюида в скважину на основе данных, полученных при расчёте на гидродинамическом симуляторе ECLIPSE Blackoil. Осуществляется аппроксимация зависимостей дебитов нефти в различные моменты времени от четырёх параметров геологии пласта.

В задачах нефтяного инжиниринга метамодели применяются при решении задач оптимизации. Например, перед использованием химических методов увеличения нефтеотдачи (ХМУН) осуществляется поиск оптимальных концентраций полимера, сурфактанта и щёлочи с целью увеличения коэффициента извлечения нефти (КИН) у месторождений на поздней стадии разработки.

Цель данной выпускной квалификационной работы: построить метамодель притока флюида к вертикальной скважине и провести сравнение двух возможных подходов к построению метамоделей при разных характеристиках аппроксимируемых данных, описывающих зависимость дебитов нефти от параметров геологии пласта.

ГЛАВА 1. ПОДГОТОВКА ДАННЫХ ДЛЯ ОБУЧЕНИЯ МЕТАМОДЕЛИ

1.1 Построение гидродинамической модели пласта

Данные для обучения метамодели подготовлены при расчёте на симуляторе ECLIPSE Blackoil.

ECLIPSE Blackoil – программа гидродинамического моделирования (моделирования фильтрационных процессов), включающая в себя набор уравнений, основанных на материальном балансе и законах фильтрации, а также численных схемах их решения для различных типов коллекторов, флюидов, степеней структурной и геологической сложности и систем разработки. Расчеты выполняются на ячеистой модели пласта (рис. 1), где внутри каждой ячейки свойства (пористость, проницаемость, пластовое давление, насыщенность и т. д.) распределены равномерно и могут изменяться во времени.

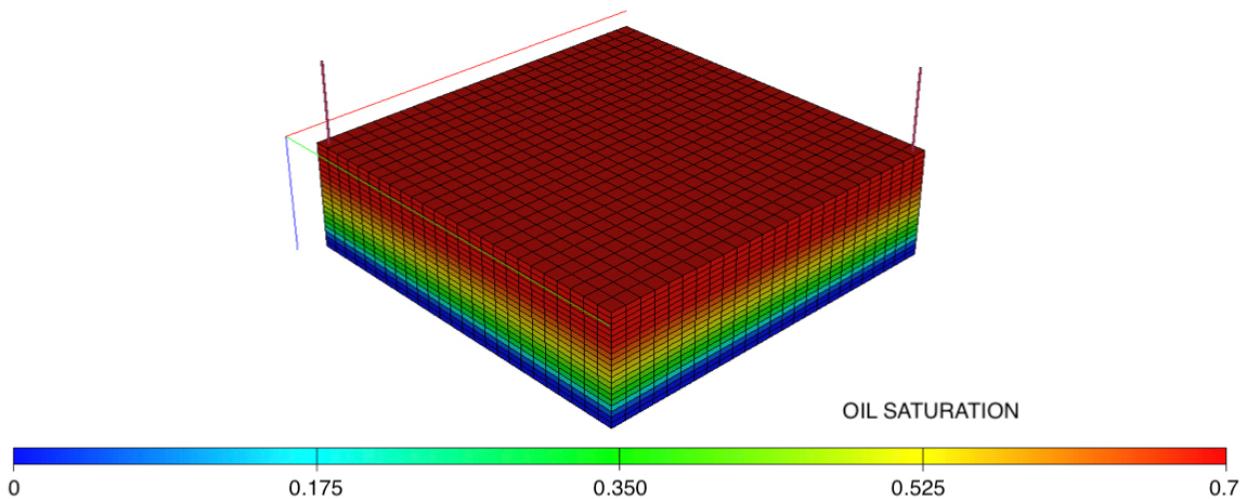


Рис. 1. Пример ячеистой модели пласта в симуляторе ECLIPSE

В модели Blackoil (нелетучей нефти) предполагается, что пластовый флюид состоит из нефти, газа и воды, которые рассматриваются как массовые компоненты, при этом только газ может быть растворен в нефти и воде. Нефть и вода не смешиваются и не обмениваются массами. В расчетах учитываются упругие свойства породы и флюидов, а также влияние гравитационных и капиллярных сил. Используется конечно-разностная схема, по которой расчет

потоков происходит между соседними ячейками через грани ячеек модели. На рис. 2 представлена схема потока между ячейками. Также указаны параметры ячеек, которые участвуют в законе Дарси.



Рис. 2. Поток между ячейками гидродинамической модели

Закон Дарси [1] для потока между ячейками:

$$Q_p = -\bar{k} \cdot \bar{A} \cdot \frac{\bar{k}_{rp}}{\bar{\mu}_p \cdot \bar{B}_p} \cdot \left(\frac{P_i - P_{i-1}}{\bar{\Delta x}} \right) = -\bar{k} \cdot \bar{A} \cdot \frac{\bar{k}_{rp}}{\bar{\mu}_p \cdot \bar{B}_p} \cdot \left(\frac{P_i - P_{i-1}}{\frac{\Delta x_{i-1} + \Delta x_i}{2}} \right), \quad (1)$$

где Q_p – расход, \bar{k} , \bar{A} , \bar{k}_{rp} , $\bar{\mu}_p$, \bar{B}_p – средние проницаемость, площадь, ОФП, вязкость и объёмный коэффициент; индекс p – фаза (нефть, газ, вода).

Значения средних параметров, входящих в закон Дарси, вычисляются следующим образом:

- для произведения проницаемости на площадь используется взвешенное среднее гармоническое произведений этих величин в ячейках:

$$\bar{k} \cdot \bar{A} = \frac{\Delta x_{i-1} + \Delta x_i}{\frac{\Delta x_{i-1}}{(k \cdot A)_{i-1}} + \frac{\Delta x_i}{(k \cdot A)_i}}; \quad (2)$$

- для физических свойств фаз используется среднее арифметическое этих величин в ячейках:

$$\overline{\mu_p} = \frac{\mu_{i-1} + \mu_i}{2}; \quad \overline{B_p} = \frac{B_{i-1} + B_i}{2}; \quad (3)$$

- для ОФП фаз используется значение в ячейке, из которой происходит отток. В случае направления потока из ячейки $i - 1$ в ячейку i будет использоваться ОФП ячейки $i - 1$:

$$\overline{k_{rp}} = (k_{rp})_{i-1}. \quad (4)$$

Для притока в скважину при установившемся режиме используется формула Дюпюи [1] отдельно для каждой фазы с учётом подвижности фаз:

$$Q_{ok} = \frac{2\pi \cdot (k \cdot h \cdot k_{ro})_k}{\mu_o \cdot \left(\ln \frac{r_{pk}}{r_w} + S_k \right)} \cdot (\bar{P}_k - P_{wfk}), \quad (5)$$

$$Q_{wk} = \frac{2\pi \cdot (k \cdot h \cdot k_{rw})_k}{\mu_w \cdot \left(\ln \frac{r_{pk}}{r_w} + S_k \right)} \cdot (\bar{P}_k - P_{wfk}), \quad (6)$$

где Q – дебит скважины, k и h – проницаемость и мощность пласта соответственно, μ – вязкость флюида, k_r – относительная фазовая проницаемость (ОФП), r_p и \bar{P} – эквивалентный радиус и среднее давление в ячейке соответственно, r_w и P_{wf} – радиус скважины и забойное давление соответственно, S – скин-фактор; дополнительные индексы o , w и k обозначают величины для нефти, величины для воды и номер слоя ячеистой модели соответственно.

Общий приток жидкости в скважину из всех перфорированных ячеек:

$$Q_t = \sum_{k=1}^n Q_{tk} = \sum_{k=1}^n (Q_{ok} + Q_{wk}) = \sum_{k=1}^n \left[\lambda_{tk} \frac{2\pi \cdot (k \cdot h)_k}{\ln \frac{r_{pk}}{r_w} + S_k} \cdot (\bar{P}_k - P_{wfk}) \right], \quad (7)$$

где $\lambda_{tk} = \lambda_{ok} + \lambda_{wk} = \left(\frac{k_{ro}}{\mu_o} + \frac{k_{rw}}{\mu_w} \right)_k$ и n – количество перфорированных ячеек.

В общем случае ECLIPSE предлагает пользователю выбор метода решения [25]. По умолчанию используется полностью неявный метод, обеспечивающий устойчивость при больших временных шагах. При использовании этого метода обеспечивается заданная точность решения нелинейных уравнений, и погрешность материального баланса сохраняется пренебрежимо малой. Для решения нелинейных уравнений используется метод итераций Ньютона, при этом матрица фильтрационных коэффициентов разложима по всем переменным, что обеспечивает квадратичную скорость сходимости. При решении сильно нелинейных задач используются различные методы ускорения сходимости. Система линейных уравнений на каждой ньютоновской итерации решается методом Гнездовой Факторизации (Nested Factorization) с ускорением за счет применения метода Ортомин (Orthomin).

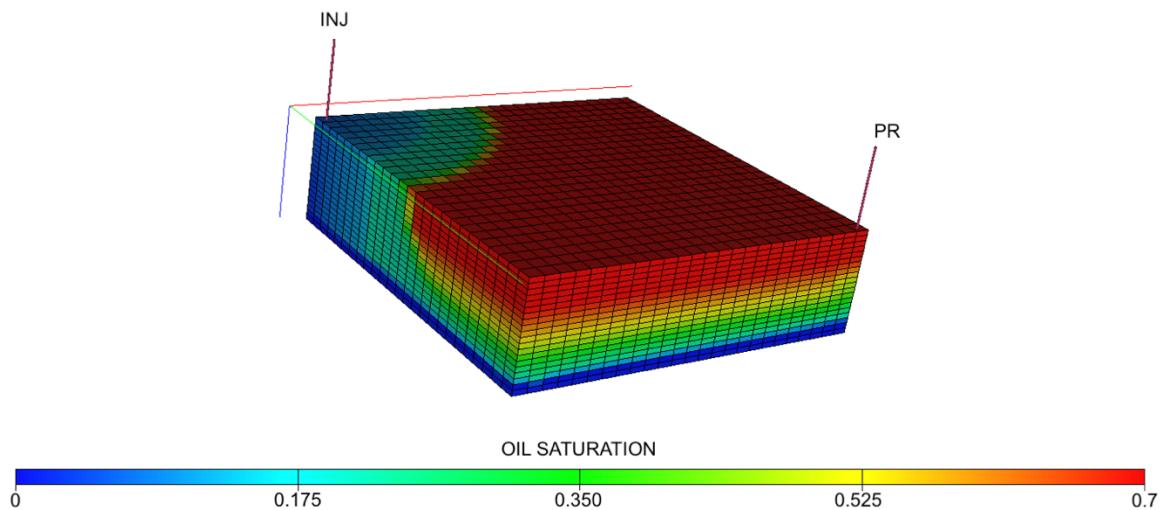


Рис. 3. Гидродинамическая модель пласта

В данной работе полная гидродинамическая модель пласта, которая используется для генерации обучающих данных метамодели, представляет собой одну четверть пятиточечной схемы заводнения. Расположение двух вертикальных скважин показано на рис. 3. Размерность равномерной сетки составляет 25 ячеек по X, 25 по Y и 40 (20 для матриц + 20 для трещин) по Z.

Размер ячейки по горизонтали фиксирован и составляет 20 метров. По вертикали размер ячейки 1.0 м.

Используется опция двойной пористости/проницаемости (dual porosity, dual permeability system). Данная опция применяется для моделирования залежей с сильно трещиноватыми коллекторами. Каждый блок сетки представляется в виде матрицы породы и трещины. Течение в пласте с двойной пористостью/проницаемостью происходит в основном по трещинам. Кроме того, осуществляется массообмен между каждой матрицей и соответствующей ей трещиной. При использовании опции двойной пористости/проницаемости (в отличие от опции только двойной пористости) могут присутствовать перетоки флюидов [25] между соседними матрицами ячеистой модели пласта.

Также в расчетах используется опция химического заводнения с добавлением поверхностно-активного вещества (ПАВ), которое применяется для уменьшения коэффициента поверхностного натяжения на границе раздела вода-нефть. Задаётся концентрация ПАВ при закачке в нагнетательную скважину.

Рассматриваются 2 сценария разработки нефтяной залежи:

- 1) изначально пластовое давление низкое, поэтому в одной из скважин нагнетание воды с ПАВ начинается с самого начала расчёта с целью увеличения и поддержания пластового давления;
- 2) сначала обе скважины добывающие и работают на естественном режиме.

Когда суммарный дебит падает ниже 20 кубометров в сутки, одна из скважин переводится в режим нагнетания воды с ПАВ.

Пример входного файла-шаблона (с расширением DATA) для расчета симулятором ECLIPSE с комментариями по всем используемым опциям и ключевым словам приведен в приложении 1.

Результаты расчетов записываются в текстовый файл с расширением RSM и включают данные по дебитам нефти, воды, объемам закачки и т. д.

1.2 Генерация кейсов для расчёта

При изменении значений параметров в файле-шаблоне изменяются значения результатов расчёта на симуляторе. Для обучения метамодели необходимо провести расчёт на некоторой выборке точек пространства варьируемых параметров. Существует множество способов генерации выборки точек, в которых будет проводиться расчёт. Различают два типа выборок: фиксированная и дополняемая. Фиксированная выборка генерируется один раз, проводятся расчёты в выбранных точках и на основе результатов обучается окончательная метамодель. В дополняемой выборке после построения метамодели анализируются полученные результаты и принимается решение о добавлении точек в определённых областях пространства с целью дальнейшего увеличения точности прогноза.

Дополняемая выборка позволяет построить точную метамодель при небольшом количестве компьютерных экспериментов, так как автоматически проводится анализ построенных метамоделей и определяются области, где необходимо провести дополнительные компьютерные эксперименты. Подробное описание алгоритмов реализации дополняемой выборки приведено в [12], [13] и [17].

В данной работе используется фиксированная выборка, которая относительно проще в реализации, так как нет необходимости запускать расчёт на симуляторе напрямую из исполняемой программы и не требуется алгоритмов для поиска области, на которой необходимо уточнить построенную метамодель.

Варьируются 4 параметра геологии пласта: пористость матрицы (PORO MATRIX), проницаемость матрицы (PERMX=PERMY MATRIX), пористость трещин (PORO CRACK) и проницаемость трещин (PERMX=PERMY CRACK). Границы диапазонов варьирования каждого из параметров представлены в таблице 1.

Таблица 1. Варьируемые параметры

	Пористость матрицы	Проницаемость матрицы, мД	Пористость трещин	Проницаемость трещин, мД
Минимальное значение	0.14	0.9	0.005	50
Максимальное значение	0.24	10	0.05	2000

В пространстве варьируемых параметров задаётся фиксированное множество точек, в которых будет проводиться расчёт на симуляторе. В качестве такого множества точек выбран латинский гиперкуб.

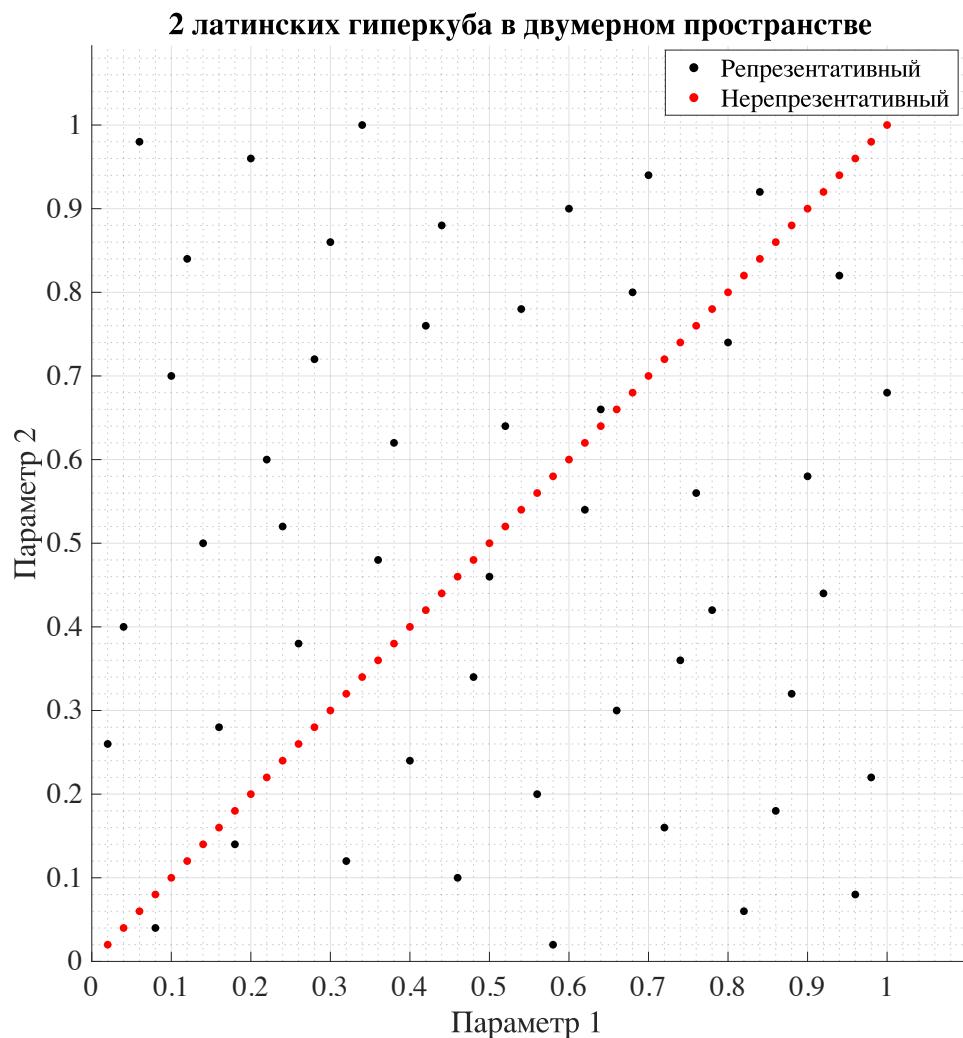


Рис. 4. Примеры латинских гиперкубов

Латинский гиперкуб создаётся путём деления каждого измерения пространства варьируемых параметров на m интервалов равного размера и размещения ровно одной точки в каждом интервале для каждого измерения. Внутри выбранных ячеек, на которые разделяется пространство варьируемых параметров, точка размещается по заранее заданному шаблону (например, в одной из вершин).

При генерации латинского гиперкуба решаются задачи оптимизации [18], [20] с целью построения репрезентативного гиперкуба, точки которого наиболее равномерно распределены по всему пространству варьируемых параметров. На рис. 4 представлены примеры репрезентативного и нерепрезентативного латинских гиперкубов из $m = 50$ точек в двумерном пространстве. Код генерации этих гиперкубов приведён в приложении 3.

Для последующего обучения метамодели в четырёхмерном пространстве генерируются три репрезентативных латинских гиперкуба: для обучения, валидации и тестирования. Количество точек в валидационной выборке составляет 20% от суммы количества точек в обучающей и валидационной выборках. Количество точек в тестовой выборке выбрано в 2 раза больше, чем в обучающей, с целью дальнейшего более полного исследования качества (точности) построенной метамодели.

Обучающая выборка используется для подбора параметров метамодели, валидационная – для остановки обучения при переобучении, тестовая – для контроля точности построенной метамодели.

Так как заранее неизвестен характер исследуемой зависимости, то количество точек P на обучающей выборке выбирается произвольно. Далее на основе анализа результатов построенной метамодели может быть принято решение увеличить или уменьшить количество точек на обучающей выборке.

На рис. 5 изображены проекции сгенерированных латинских гиперкубов. Код генерации латинских гиперкубов представлен в приложении 4.

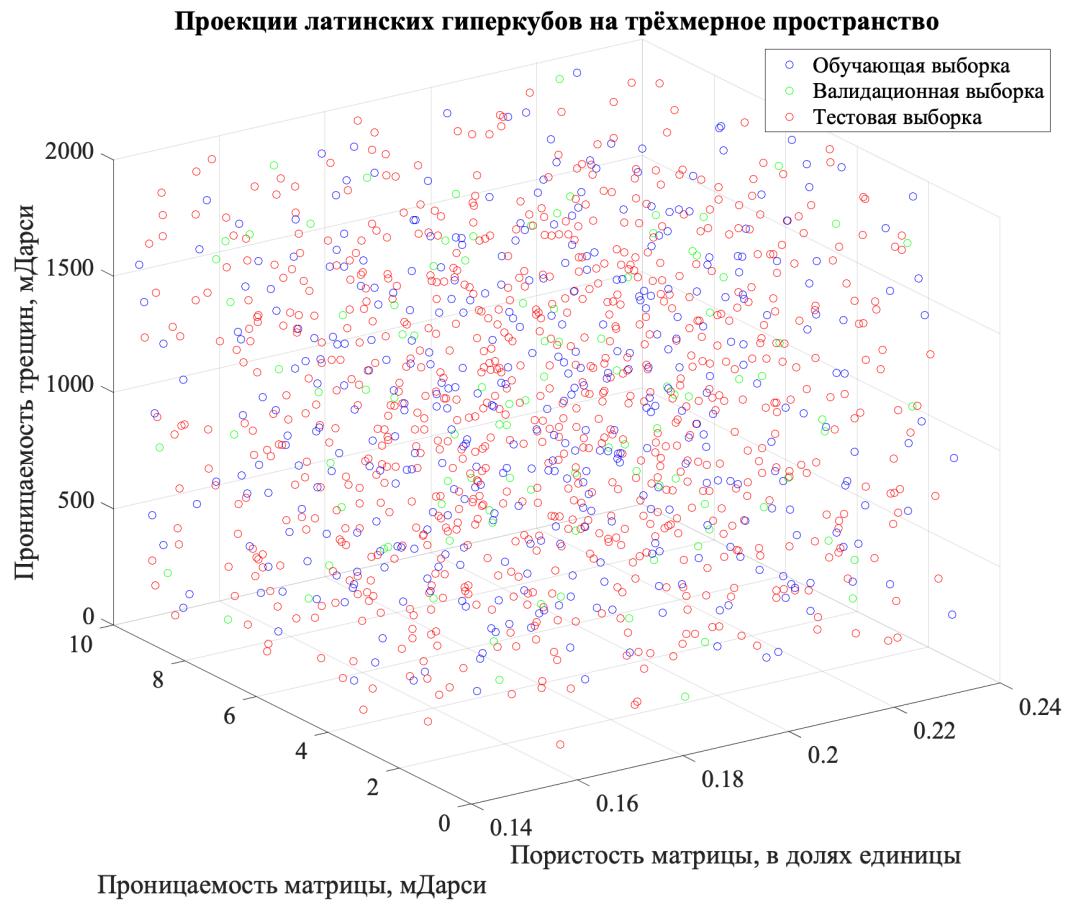


Рис. 5. Проекции трёх латинских гиперкубов

Далее в файле-шаблоне редактируются строки, содержащие информацию о значениях варьируемых параметров (подставляются значения из сгенерированных латинских гиперкубов). Создаются файлы для расчёта на симуляторе ECLIPSE. Код генерации файлов представлен в приложении 5.

1.3 Считывание данных из выходных файлов

После расчётов на гидродинамическом симуляторе проводится считывание выходных файлов. Файлы формата RSM содержат информацию о дебитах и накопленной добычи воды и нефти в различные моменты времени. Файлы формата UNRST содержат информацию о нефтенасыщенностях и водонасыщенностях в ячейках пласта в различные моменты времени. Данные, полученные из выходных файлов формата RSM, будут использованы для обучения метамодели. Код считывания выходных файлов симулятора представлен в приложении 6.

После считывания выходных файлов в рабочем пространстве (Workspace) MATLAB есть все необходимые данные для последующего обучения метамодели.

Эти данные могут быть объединены в таблицу формата csv для последующего считывания сторонними программами построения метамоделей:

```
writematrix([input;FOPR_target], 'cases1.csv')
```

ГЛАВА 2. ОБУЧЕНИЕ МЕТАМОДЕЛИ

2.1 Нейронная сеть

Перед обучением необходимо выбрать архитектуру нейронной сети, которая определяет каким образом и в каком порядке проводятся преобразования входного вектора в выходной вектор.

В данной работе выбран двухслойный перцептрон [26], архитектура которого представлена на рис. 6.

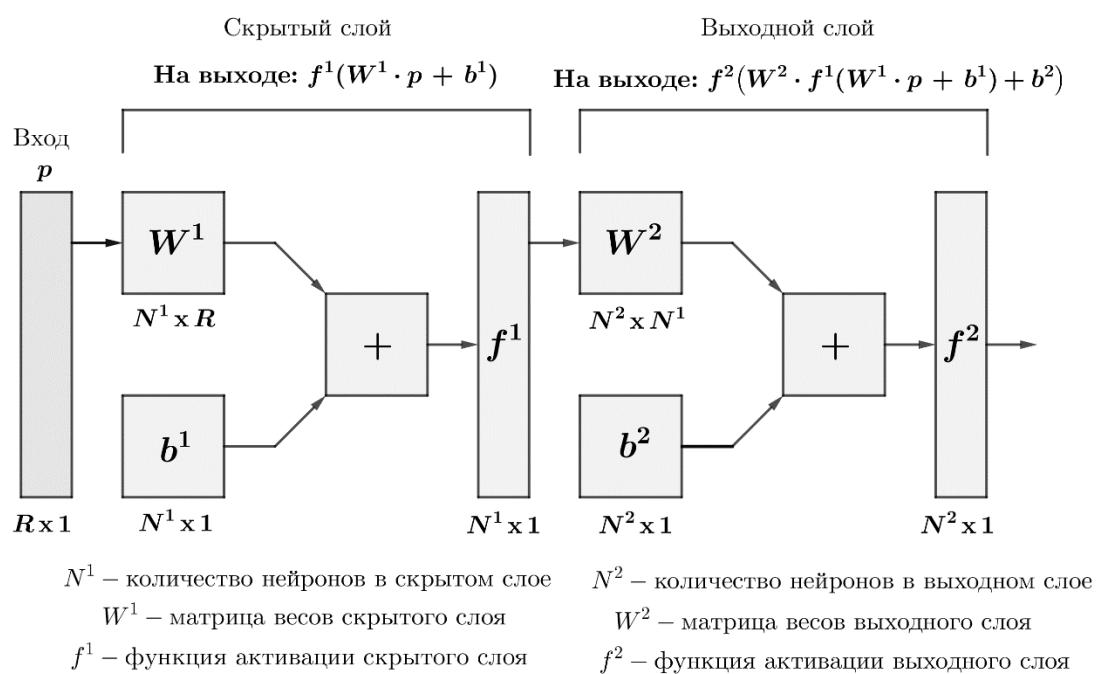


Рис. 6. Архитектура двухслойного перцептрона

Входные данные: вектор p из четырёх параметров геологии пласта (значения из сгенерированных латинских гиперкубов).

Целевые данные: вектор значений дебитов нефти по месяцам, полученный при расчёте на гидродинамическом симуляторе (значения из выходных файлов формата RSM).

Количество нейронов в скрытом слое N^1 варьируется для достижения максимальной точности.

В данной задаче варьировались 4 параметра, между которыми при генерации латинского гиперкуба корреляции не заданы (рис. 7) и каждый из которых существенно влияет на результат расчёта, поэтому извлечения признаков [8] (объединения нескольких параметров с целью уменьшения размерности пространства входных параметров) не проводилось.

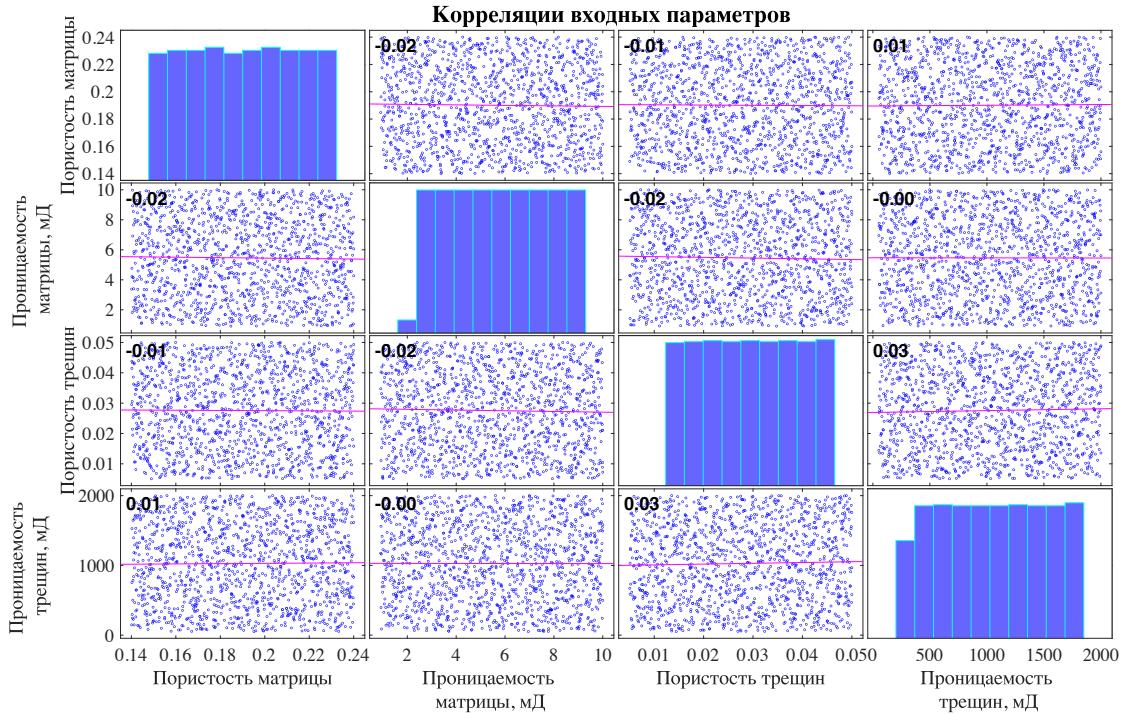


Рис. 7. Корреляции между варьируемыми параметрами
corrplot(input')

Так как входные данные имеют разные единицы измерения и диапазоны значений, то перед обучением нейронной сети проведена нормализация этих данных к диапазону $[-1; 1]$ по формуле:

$$X_{\text{норм}} = \frac{2(X - X_{\min})}{X_{\max} - X_{\min}} - 1. \quad (8)$$

Аналогично проведена нормализация целевых данных.

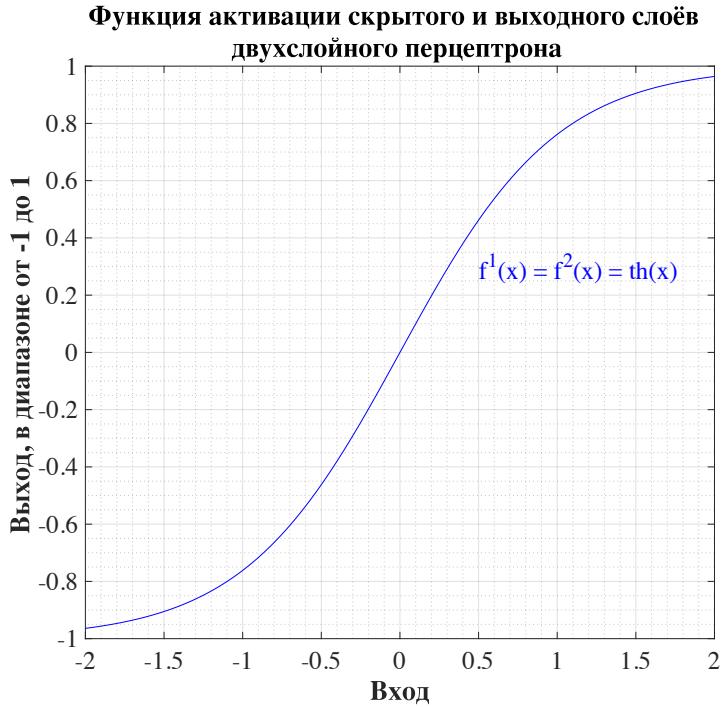


Рис. 8. Используемая функция активации

С учётом специфики решаемой задачи (в предсказанных значениях дебитов нефти не должно быть отрицательных значений) выбраны функции активации (рис. 8) на скрытом и выходном слоях двухслойного перцептрона:

$$f^1(x) = f^2(x) = \text{th } x = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \quad (9)$$

Совместный выбор способа нормализации (8) и функций активации (9) накладывает ограничение на допустимые выходные значения рассматриваемой нейронной сети: выходные значения при любых значениях входных параметров лежат в диапазоне между минимальным и максимальным значениями дебита нефти на обучающей выборке.

Настройка параметров метамодели в матрицах весов W^1, W^2 и в векторах отклонений b^1, b^2 осуществляется в процессе обучения с использованием метода масштабируемых сопряжённых градиентов [16].

В качестве функции потерь (количественной характеристики ошибки принятия решений двухслойным перцептроном) выбрана среднеквадратичная ошибка прогноза нейронной сети относительно целевых значений:

$$mse = \frac{1}{N} \sum_{i=1}^N (y_i - t_i)^2, \quad (10)$$

где N – количество прогнозируемых нейронной сетью значений, y_i – прогноз нейронной сети, t_i – целевое значение, полученное при расчёте на симуляторе.

Обучение останавливается, если в течение 1000 предыдущих эпох среднеквадратичная ошибка прогноза на валидационной выборке не уменьшилась.

Код построения и обучения двухслойного перцептрона представлен в приложении 8.

После нескольких попыток обучения нейронной сети сформулирован алгоритм, по которому в дальнейшем проводилось обучение двухслойного перцептрона:

- 1) сгенерировать гиперкуб произвольного (представляемого разумным для данной задачи) размера и около десяти раз провести обучение двухслойного перцептрона с произвольным количеством нейронов в скрытом слое; выбрать нейронную сеть с наименьшей среднеквадратичной ошибкой прогноза на валидационной выборке; далее проанализировать графики функции потерь в процессе обучения выбранной нейронной сети для трёх выборок (обучающей, валидационной и тестовой);
- 2) если mse на обучающей выборке существенно меньше, чем на валидационной, то необходимо вернуться к пункту 1, уменьшив количество нейронов в скрытом слое;
- 3) если mse на обучающей и валидационной выборках отличаются незначительно и достигнута требуемая точность, то необходимо перейти к визуализации результатов;
- 4) если mse на обучающей и валидационной выборках отличаются незначительно, но не достигнута требуемая точность, то необходимо вернуться к пункту 1, увеличив при этом размер гиперкубов.

Обучение метамодели проведено при:

- 1) первом сценарии разработки и $P = 400$ точках на обучающей выборке;
- 2) первом сценарии разработки и $P = 800$ точках на обучающей выборке;
- 3) втором сценарии разработки и $P = 800$ точках на обучающей выборке.

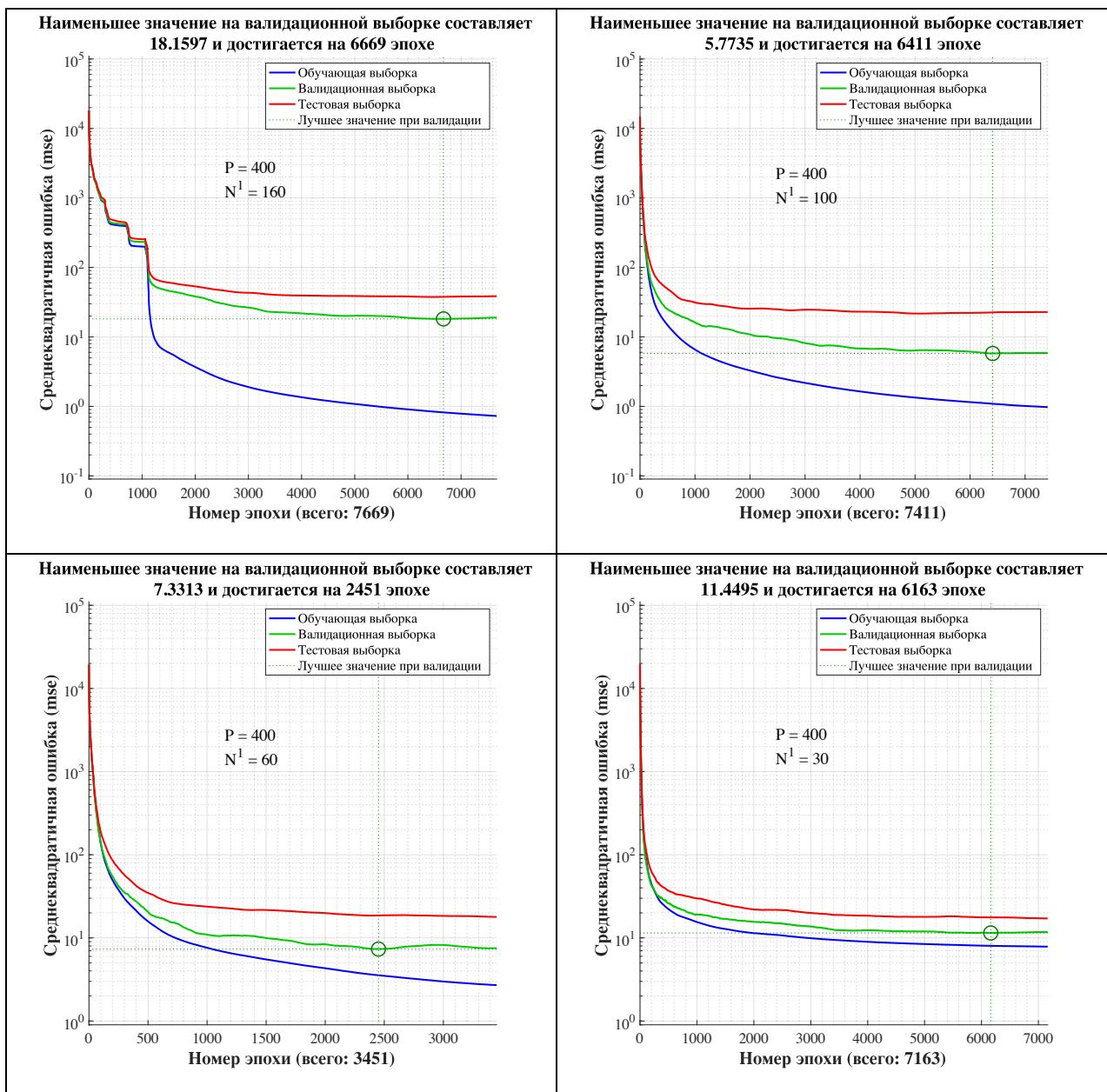


Рис. 9. Визуализация истории процесса обучения четырёх перцептронов с разным количеством нейронов N^1 в скрытом слое ($P = 400$, сценарий разработки 1)

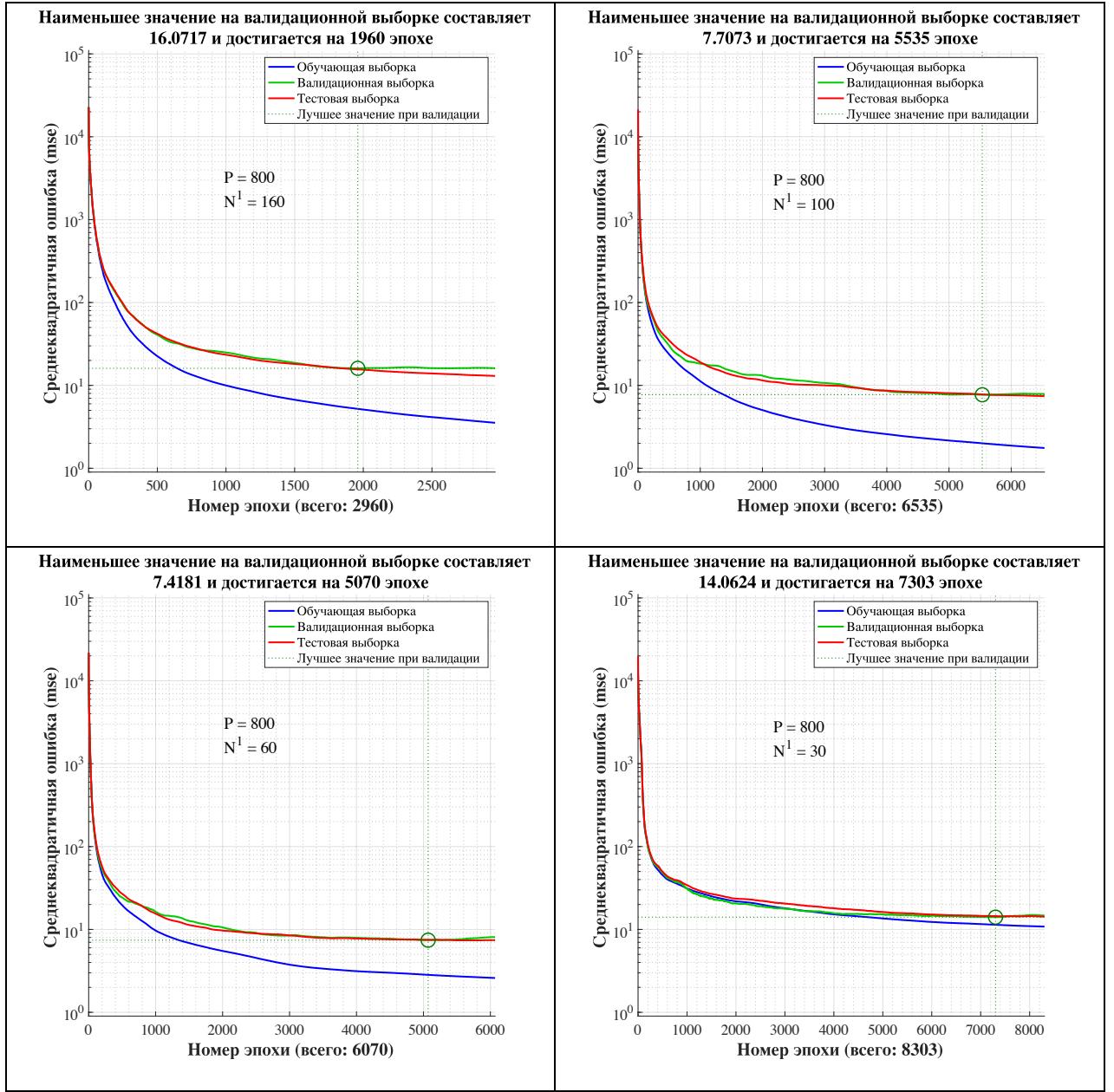


Рис. 10. Визуализация истории процесса обучения четырёх перцептронов с разным количеством нейронов N^1 в скрытом слое ($P = 800$, сценарий разработки 1)

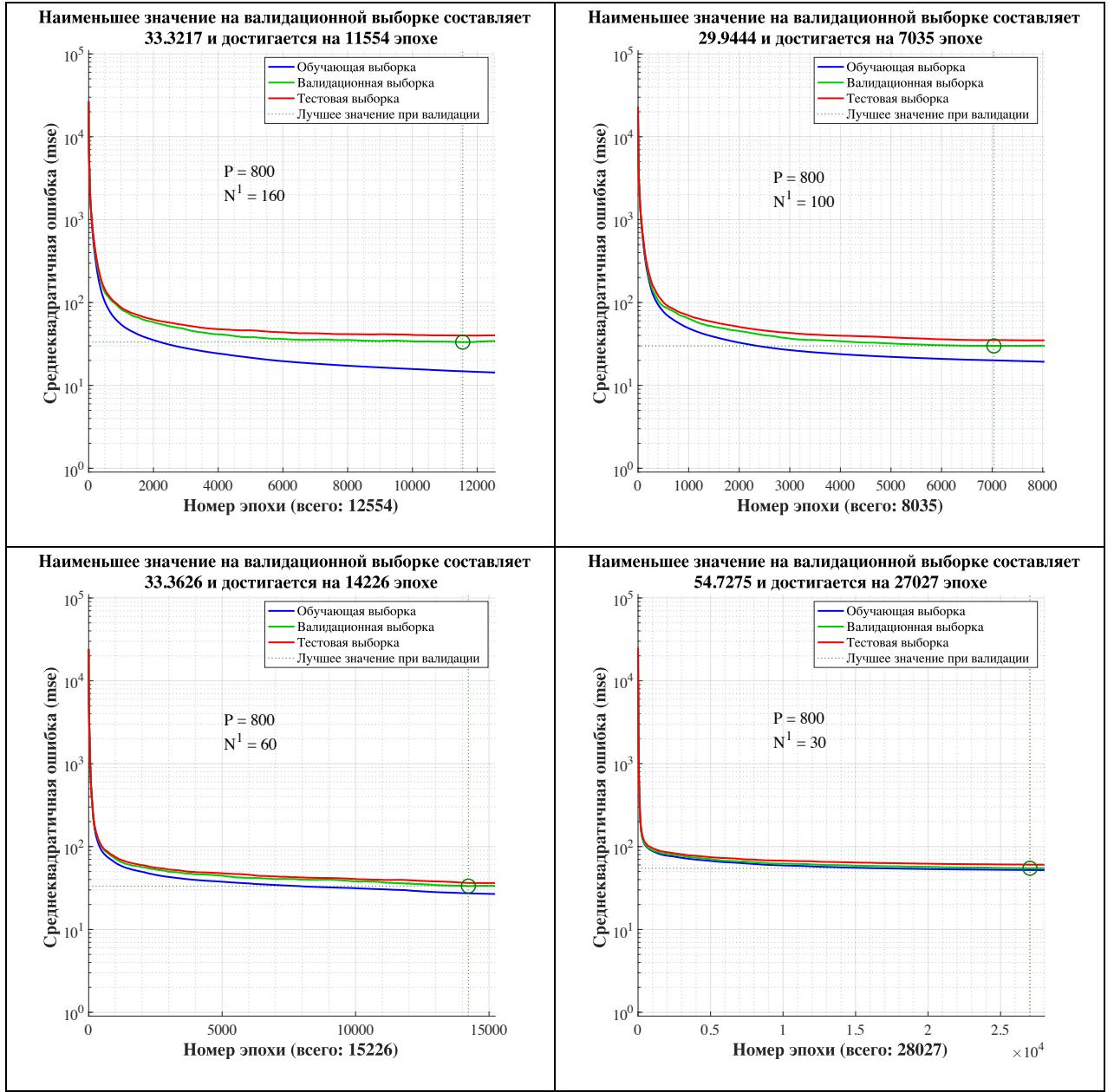


Рис. 11. Визуализация истории процесса обучения четырёх перцептронов с разным количеством нейронов N^1 в скрытом слое ($P = 800$, сценарий разработки 2)

В следующей главе для каждого из перцептронов, истории обучения которых представлены на рис. 9 – 11, будет проведена визуализация ошибок прогноза для всех рассчитанных на симуляторе кейсов.

2.2 Градиентный бустинг на основе деревьев регрессии

Деревья регрессии представляют собой иерархическую структуру, состоящую из узлов принятия решения (логических утверждений в отношении того или иного входного параметра) для прогнозирования результирующего значения (рис. 12).

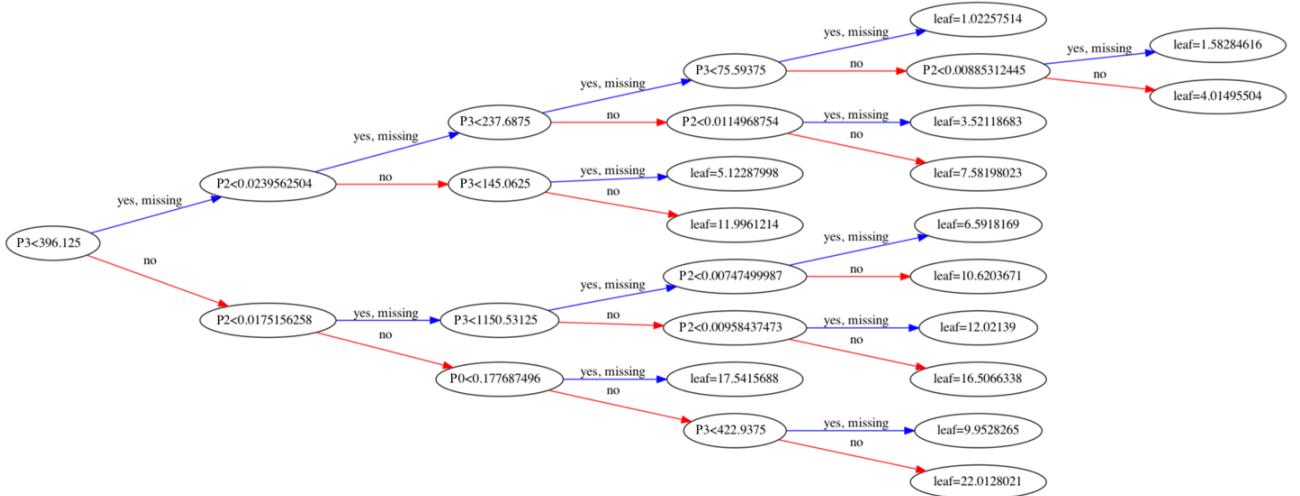


Рис. 12. Пример дерева регрессии

Градиентный бустинг – техника построения моделей на основе ансамбля деревьев регрессии. На каждой итерации к обученному ансамблю добавляется ещё одно дерево регрессии, которое предсказывает отклонение всего ансамбля, тем самым уменьшая ошибку предсказания всей модели.

В данной работе градиентный бустинг реализован с помощью библиотеки XGBoost языка программирования Python. Код представлен в приложении 14.

По графикам функции потерь (рис. 13 – 14) ошибка прогноза XGBoost больше, чем ошибка прогноза нейронной сети.

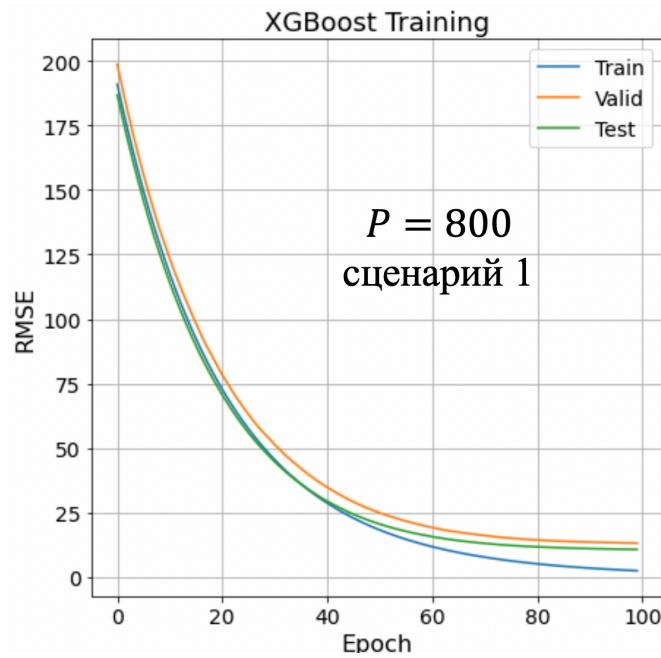


Рис. 13. Визуализация истории процесса обучения XGBoost модели (сценарий 1)

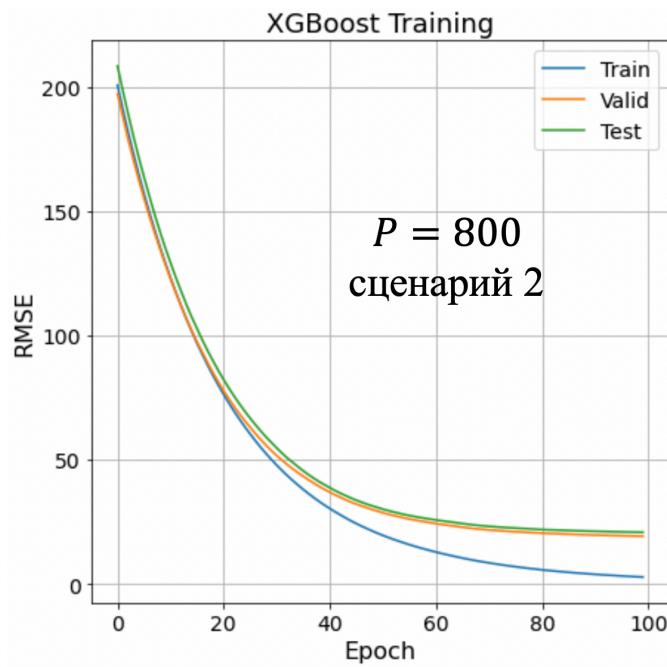


Рис. 14. Визуализация истории процесса обучения XGBoost модели (сценарий 2)

Построенные XGBoost модели позволяют оценить влияние каждого из входных параметров на результат прогноза (рис. 15).

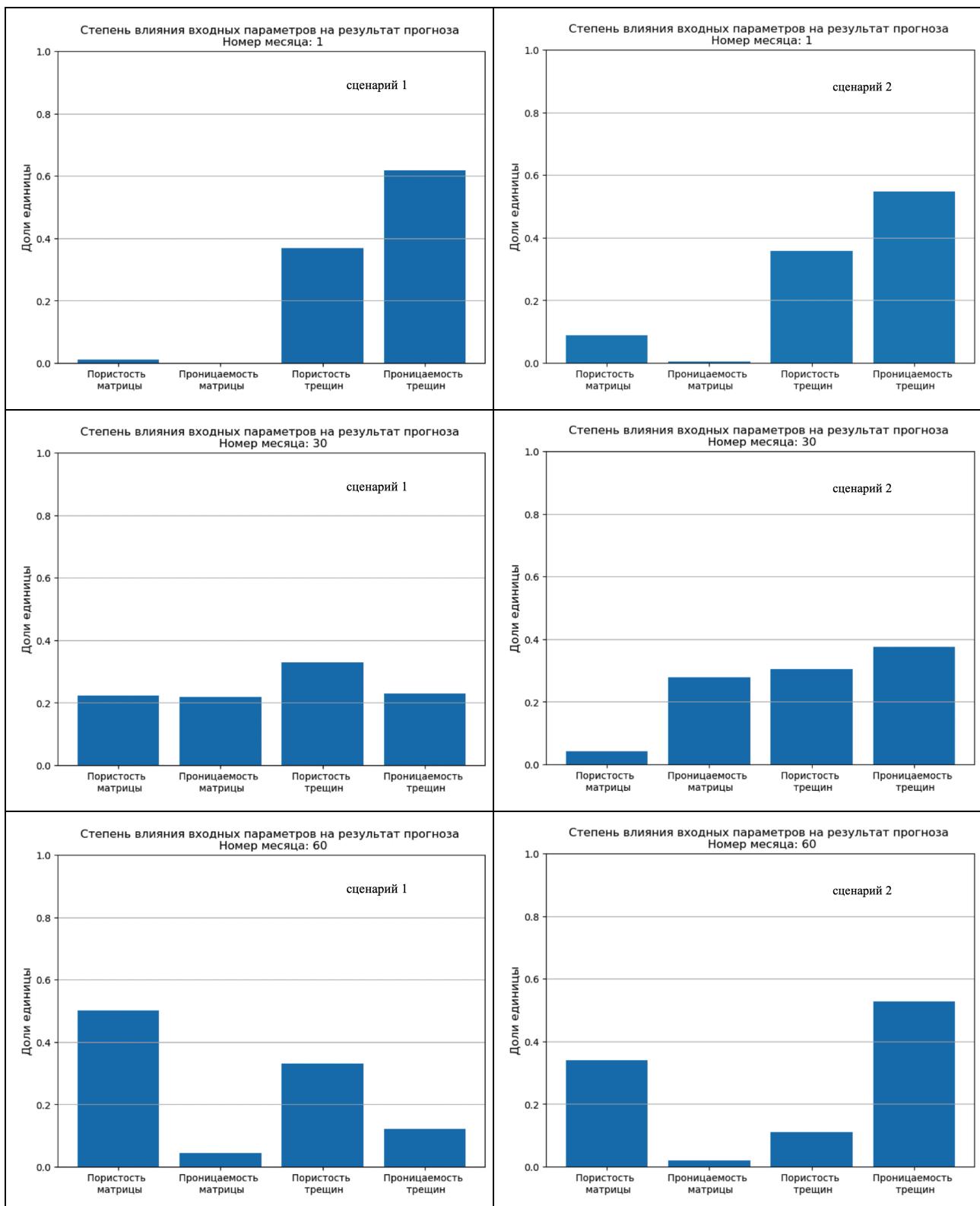


Рис. 15. Оценка влияния входных параметров на результат прогноза

В первый месяц на величину дебита нефти существенно влияют пористость и проницаемость трещин, так как фильтрация жидкостей осуществляется преимущественно по трещинам. Далее (месяц №30) каждый из

параметров оказывает существенное влияние на результат расчёта: в процессе фильтрации участвуют и матрица, и трещины. Пористость матрицы становится особенно важной после продолжительного времени (месяц №60), так как от пористости матрицы существенно зависят первоначальные запасы нефти.

ГЛАВА 3. АНАЛИЗ ТОЧНОСТИ ПРОГНОЗА И ВИЗУАЛИЗАЦИЯ РЕЗУЛЬТАТОВ

3.1 Диаграммы размаха ошибок

Чтобы получить наиболее полное представление о точности построенных метамоделей, генерируются диаграммы размаха ошибок прогноза для каждого из месяцев на всех выборках (обучающей, валидационной и тестовой).

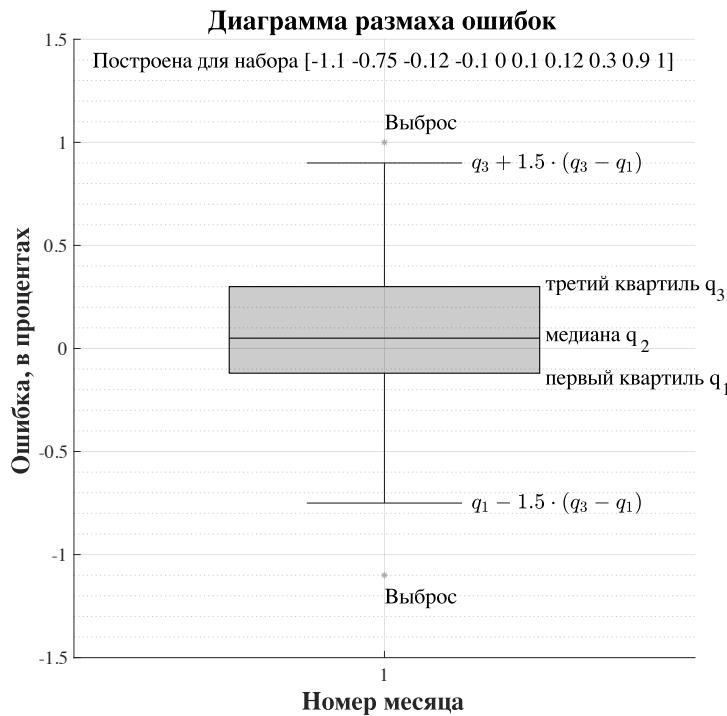


Рис. 16. Пример диаграммы размаха ошибок

На диаграмме (рис. 16) отображаются:

- 1) медиана всех данных q_2 ;
- 2) первый квартиль q_1 (медиана данных, лежащих слева от медианы всех данных);
- 3) третий квартиль q_3 (медиана данных, лежащих справа от медианы всех данных);
- 4) выбросы (значения, которые не попадают в интервал от $q_1 - 1.5 \cdot (q_3 - q_1)$ до $q_3 + 1.5 \cdot (q_3 - q_1)$).

На рис. 17 – 28 представлены диаграммы размаха ошибок прогноза нейронной сети для двух сценариев разработки нефтяной залежи при варьировании количества нейронов N^1 в скрытом слое и разном количестве точек P на обучающей выборке. При расчётах использованы нейронные сети, графики функции потерь которых представлены на рис. 9 – 11.

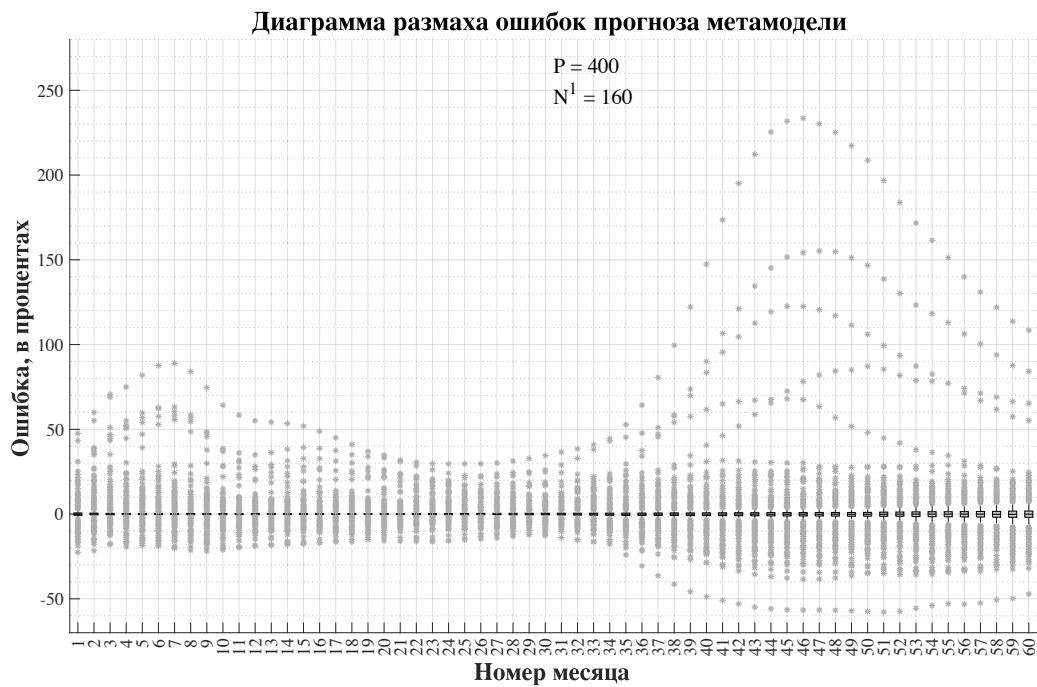


Рис. 17. Диаграмма размаха ошибок прогноза нейронной сети
($P = 400, N^1 = 160$, сценарий разработки 1)

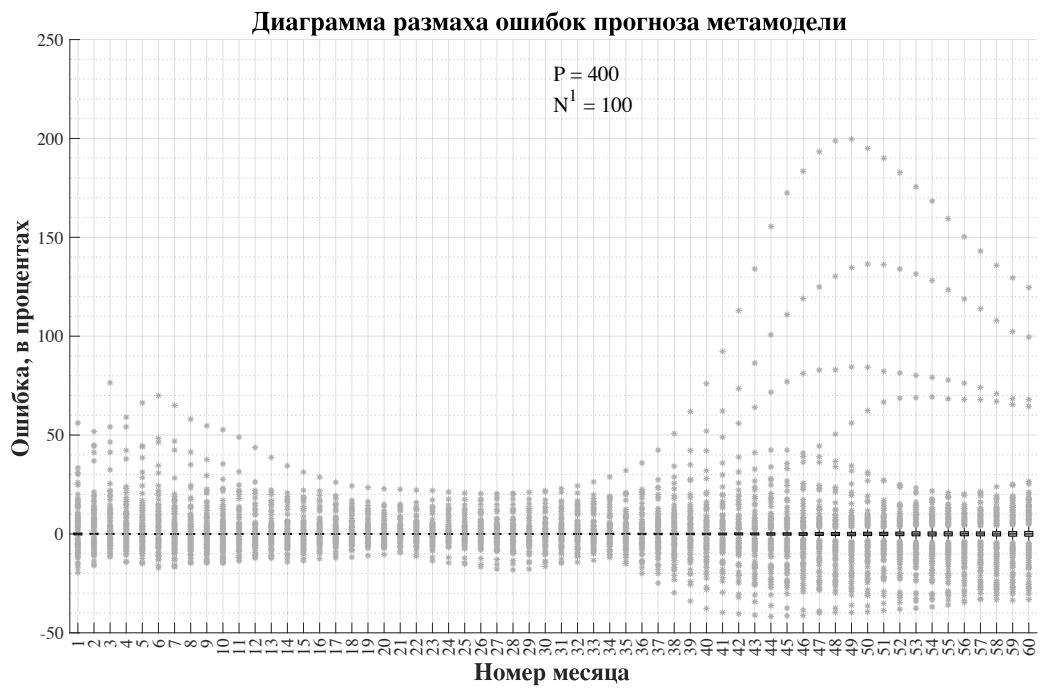


Рис. 18. Диаграмма размаха ошибок прогноза нейронной сети
($P = 400$, $N^1 = 100$, сценарий разработки 1)

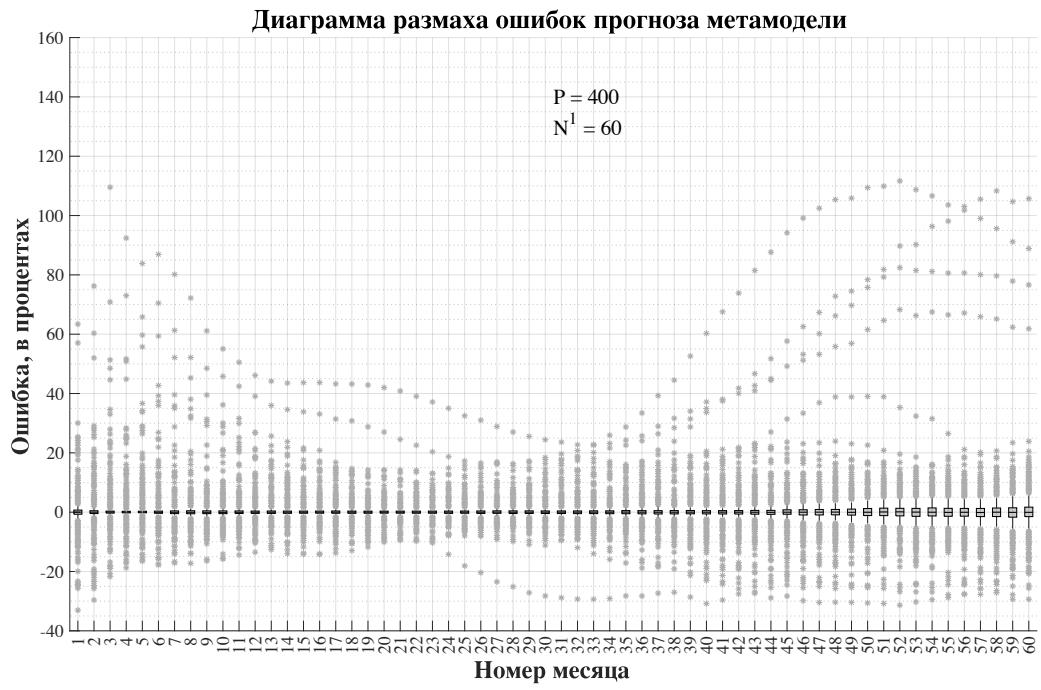


Рис. 19. Диаграмма размаха ошибок прогноза нейронной сети
($P = 400$, $N^1 = 60$, сценарий разработки 1)

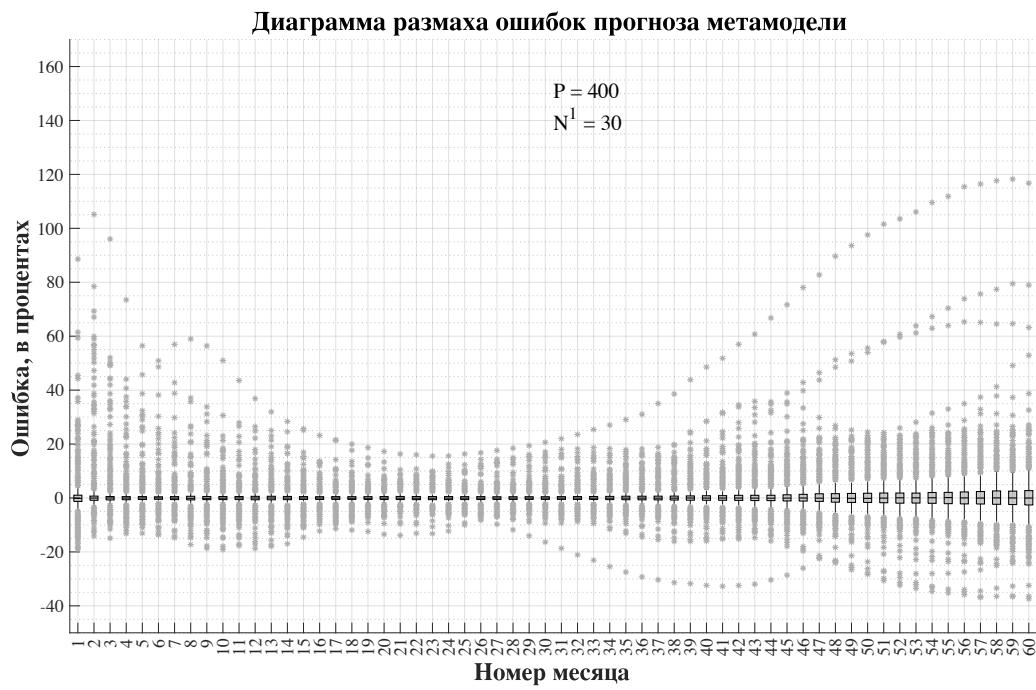


Рис. 20. Диаграмма размаха ошибок прогноза нейронной сети
($P = 400$, $N^1 = 30$, сценарий разработки 1)

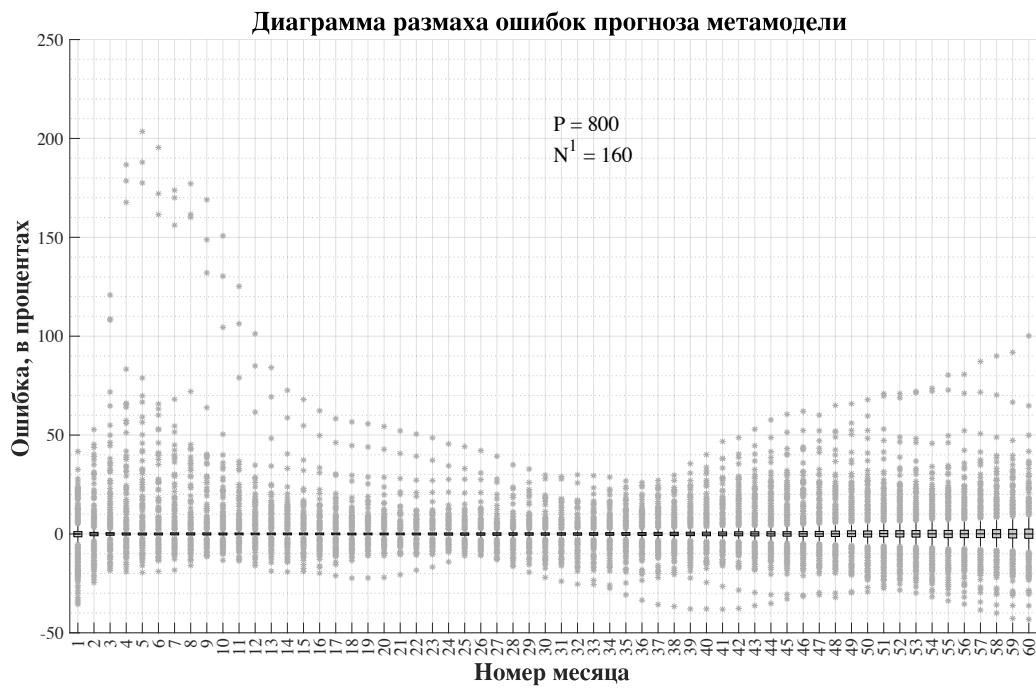


Рис. 21. Диаграмма размаха ошибок прогноза нейронной сети
($P = 800$, $N^1 = 160$, сценарий разработки 1)

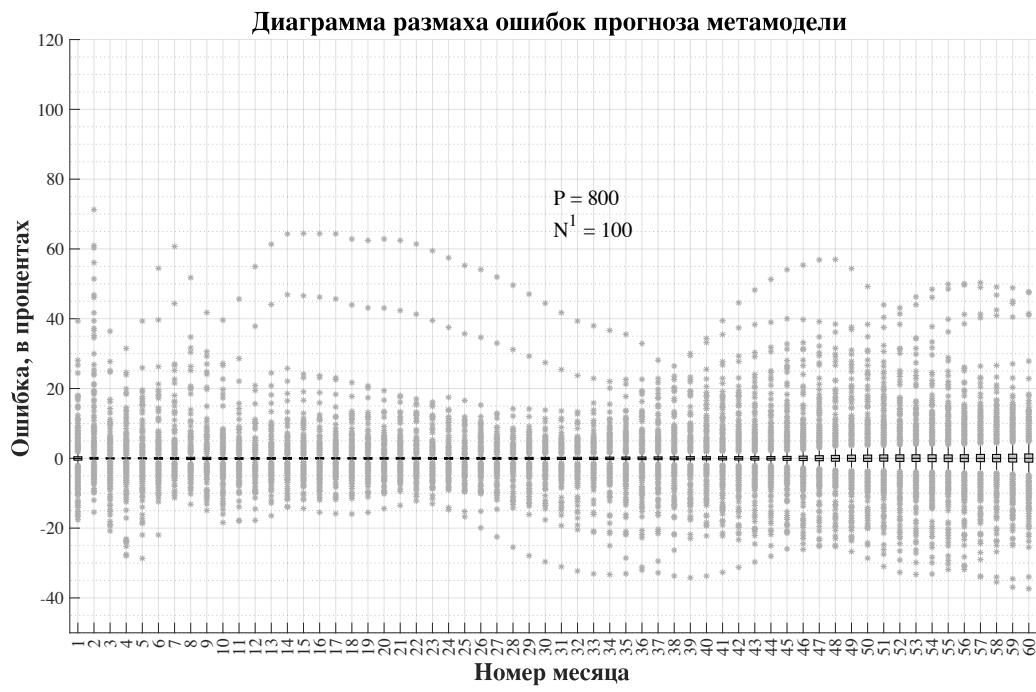


Рис. 22. Диаграмма размаха ошибок прогноза нейронной сети
($P = 800, N^1 = 100$, сценарий разработки 1)

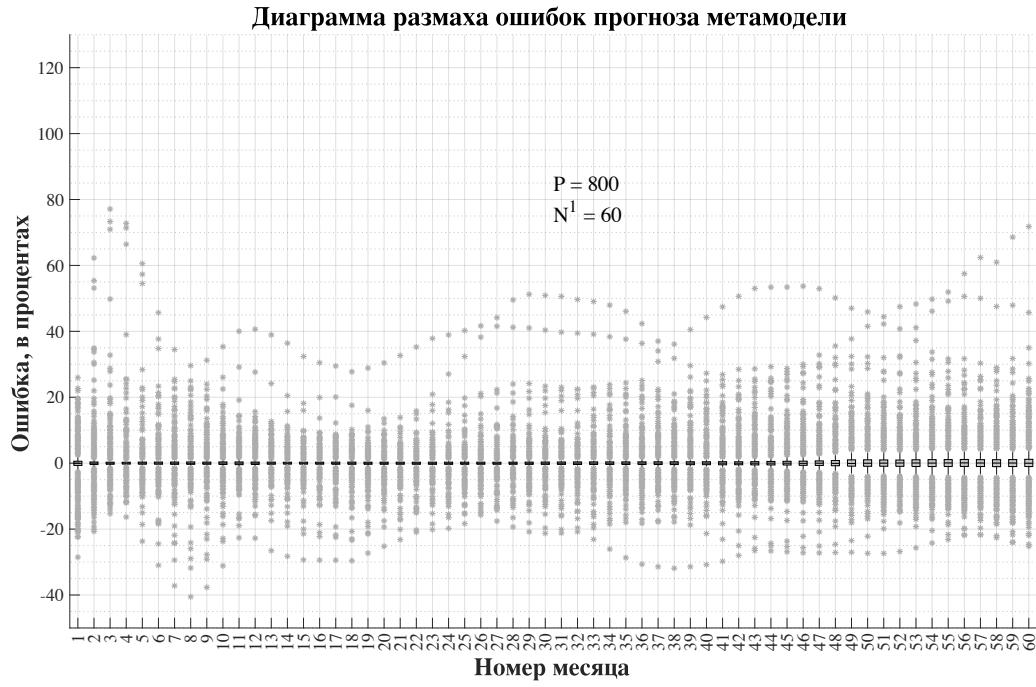


Рис. 23. Диаграмма размаха ошибок прогноза нейронной сети
($P = 800, N^1 = 60$, сценарий разработки 1)

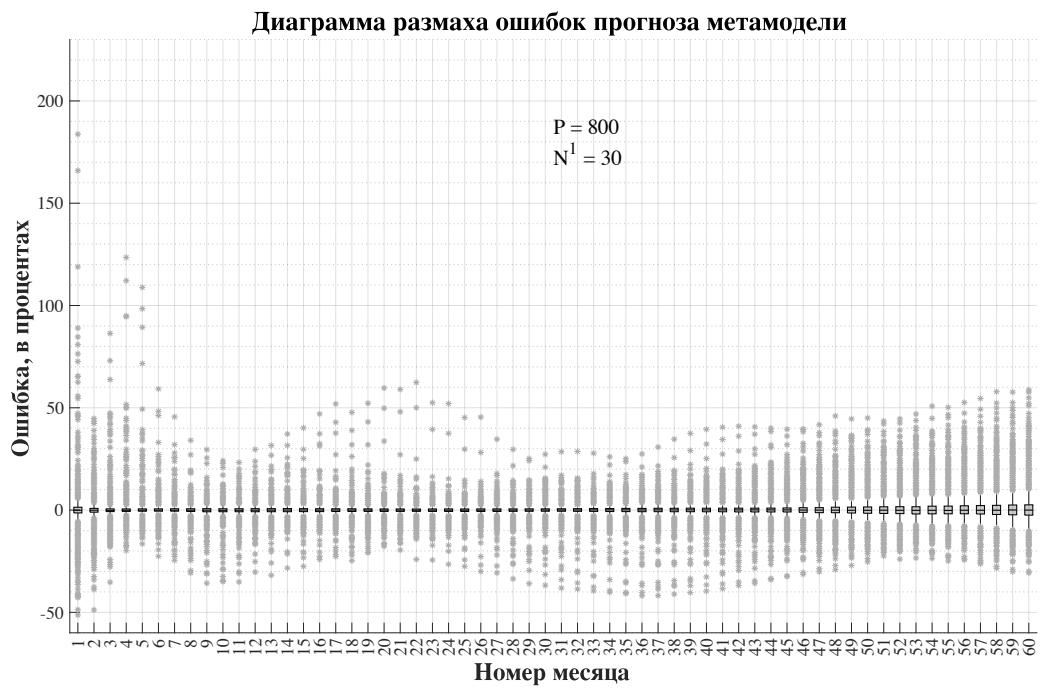


Рис. 24. Диаграмма размаха ошибок прогноза нейронной сети
($P = 800, N^1 = 30$, сценарий разработки 1)

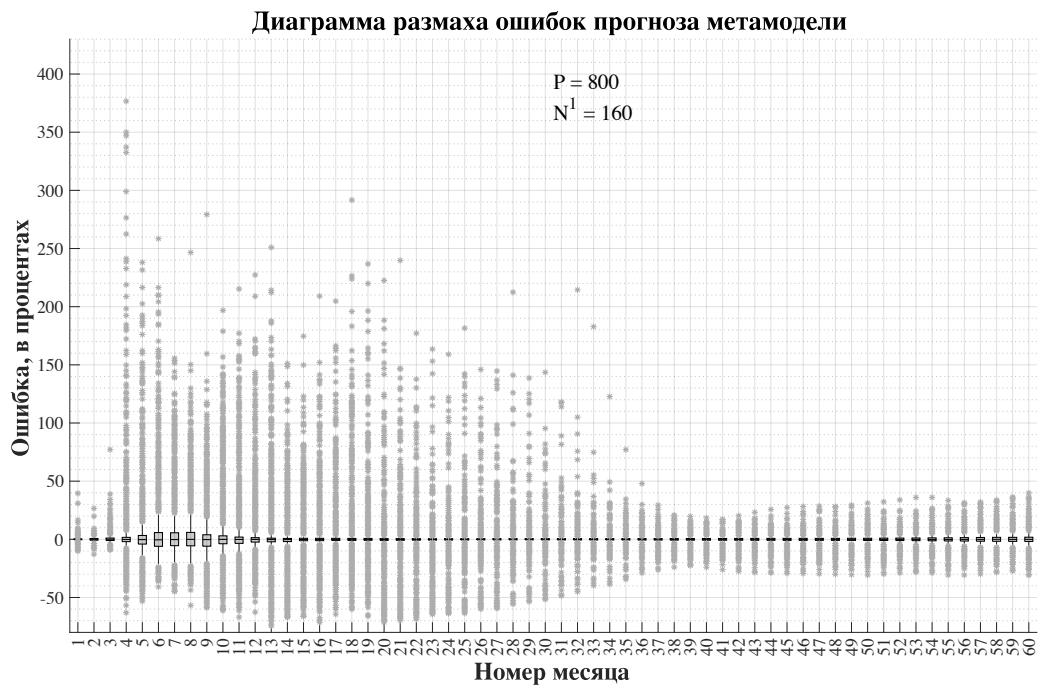


Рис. 25. Диаграмма размаха ошибок прогноза нейронной сети
($P = 800, N^1 = 160$, сценарий разработки 2)

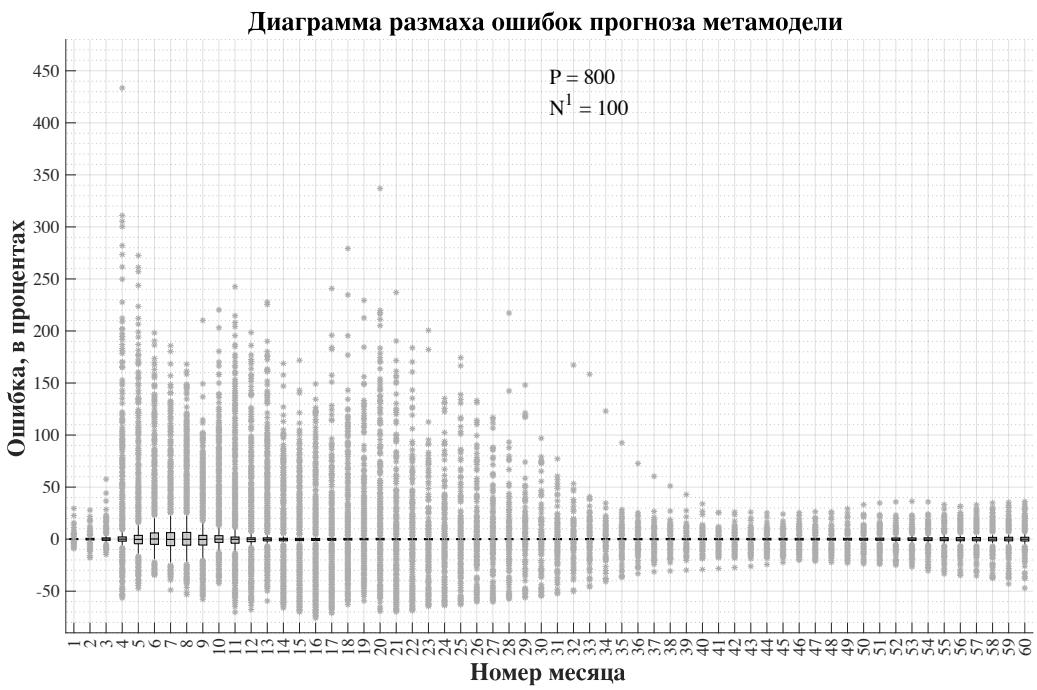


Рис. 26. Диаграмма размаха ошибок прогноза нейронной сети
($P = 800$, $N^1 = 100$, сценарий разработки 2)

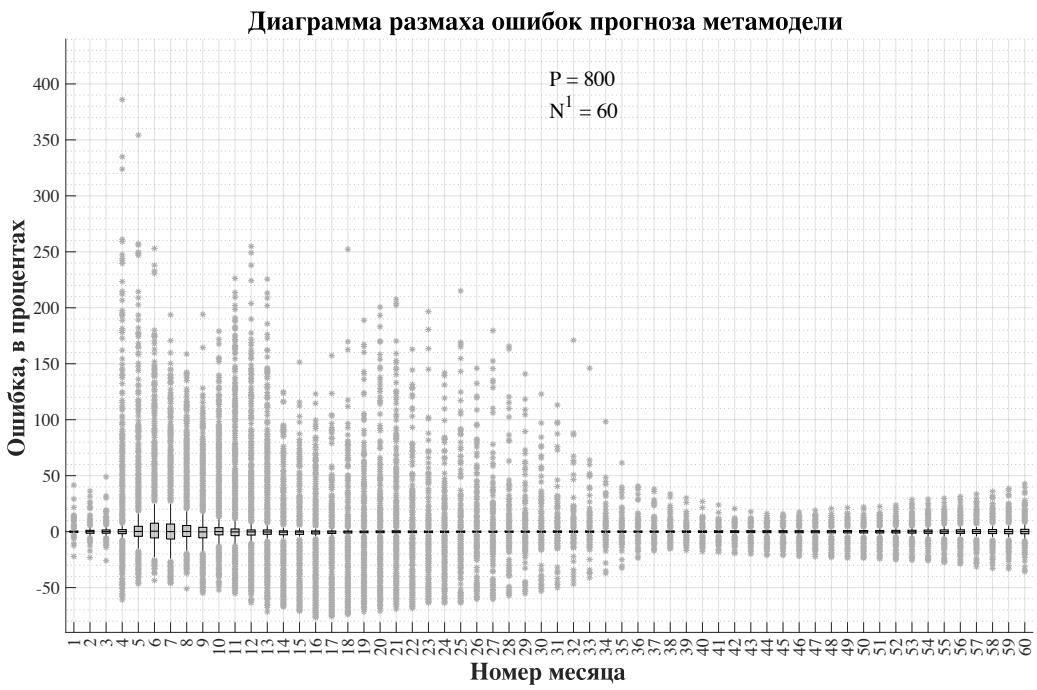


Рис. 27. Диаграмма размаха ошибок прогноза нейронной сети
($P = 800$, $N^1 = 60$, сценарий разработки 2)

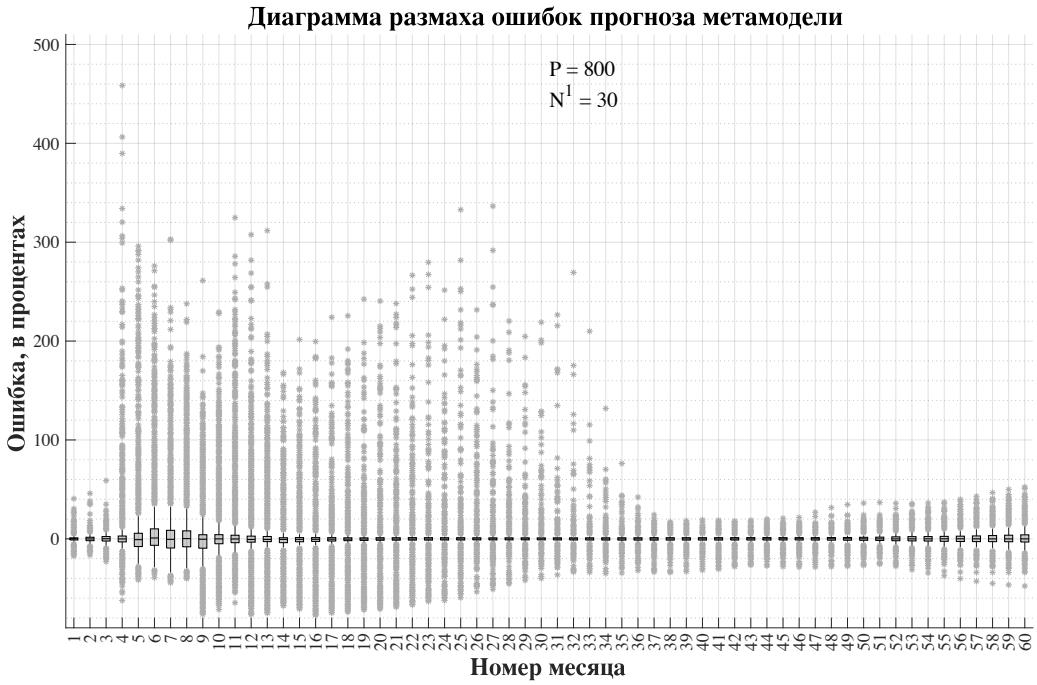


Рис. 28. Диаграмма размаха ошибок прогноза нейронной сети
($P = 800$, $N^1 = 30$, сценарий разработки 2)

На основе анализа истории процесса обучения двухслойных перцептронов (см. рис. 9 – 11) и диаграмм размаха ошибок прогноза (см. рис. 17 – 28) выбраны нейронные сети, которые будут использованы для визуализации результатов прогноза. Высокие относительные ошибки прогноза нейронной сети возникают при невысоких значениях дебита нефти (сценарий 1) или при прогнозе дебита для месяцев, на которых для некоторых кейсов осуществлялся перевод одной из скважин в режим нагнетания (сценарий 2).

На рис. 29 – 30 представлены диаграммы размаха ошибок прогноза XGBoost метамодели для двух сценариев разработки нефтяной залежи для первых десяти месяцев. При расчётах использованы XGBoost модели, графики функции потерь которых представлены на рис. 13 – 14.

По диаграммам размаха (см. рис. 29 и рис. 30) ошибки прогноза XGBoost модели больше, чем ошибки прогноза нейронной сети (см. рис. 23 и рис. 27 соответственно).

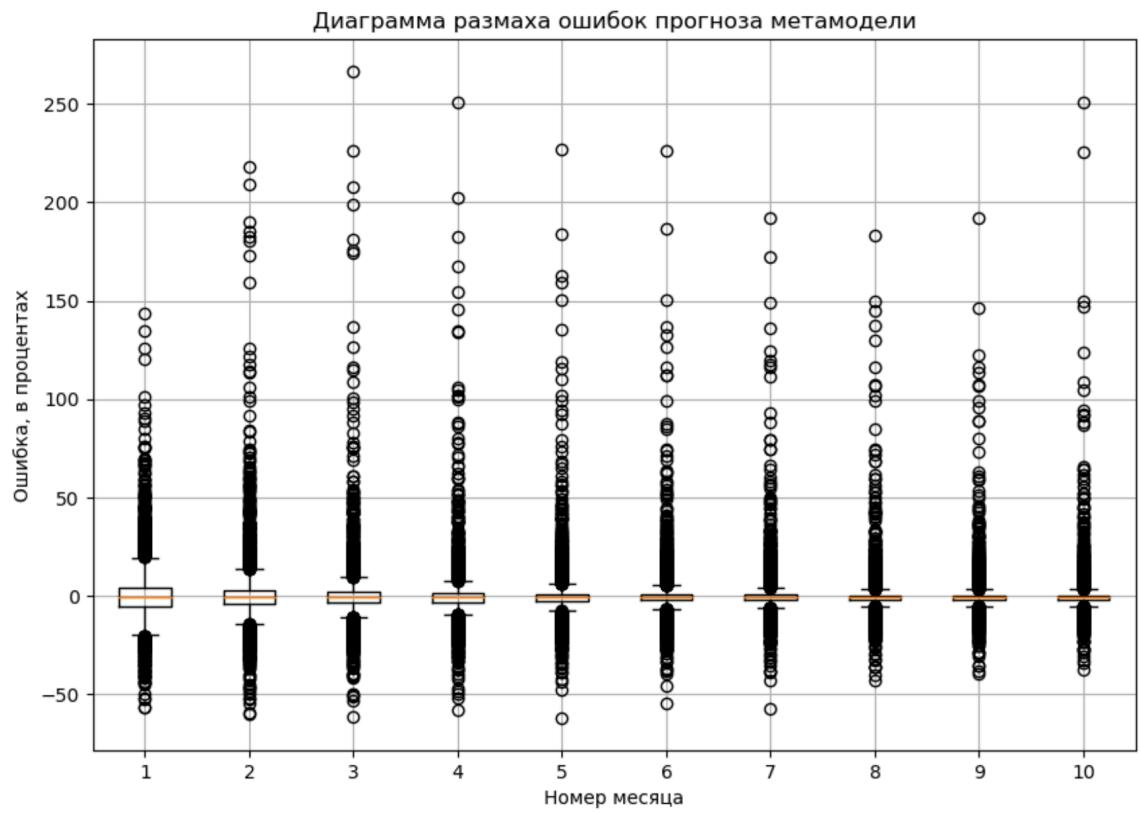


Рис. 29. Диаграмма размаха ошибок прогноза XGBoost
($P = 800$, сценарий 1)

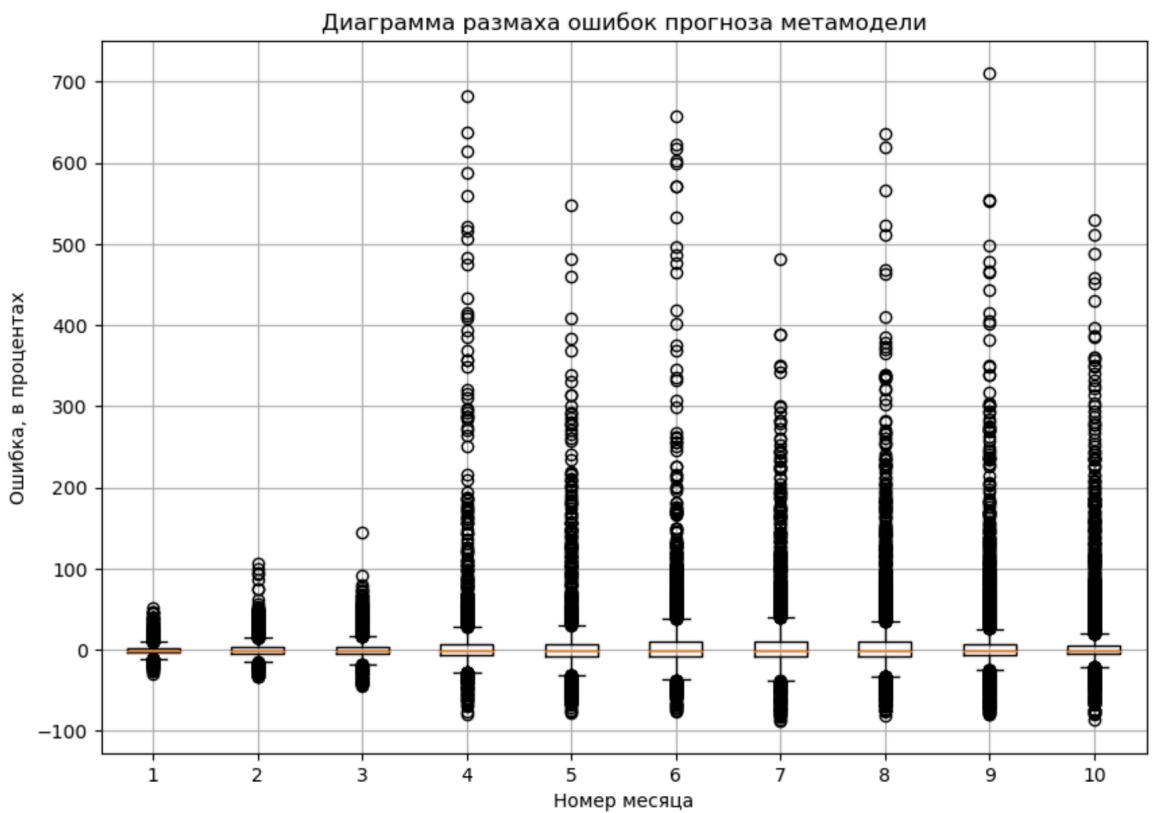


Рис. 30. Диаграмма размаха ошибок прогноза XGBoost
($P = 800$, сценарий 2)

3.2 Сравнение результатов (симулятор/метамодель)

Построены графики (рис. 31 – 32) зависимости дебита нефти от времени для всех кейсов, рассчитанных на симуляторе. Значения дебитов получены при расчёте на симуляторе, а также с помощью нейронной сети.

Код построения графиков представлен в приложении 10.

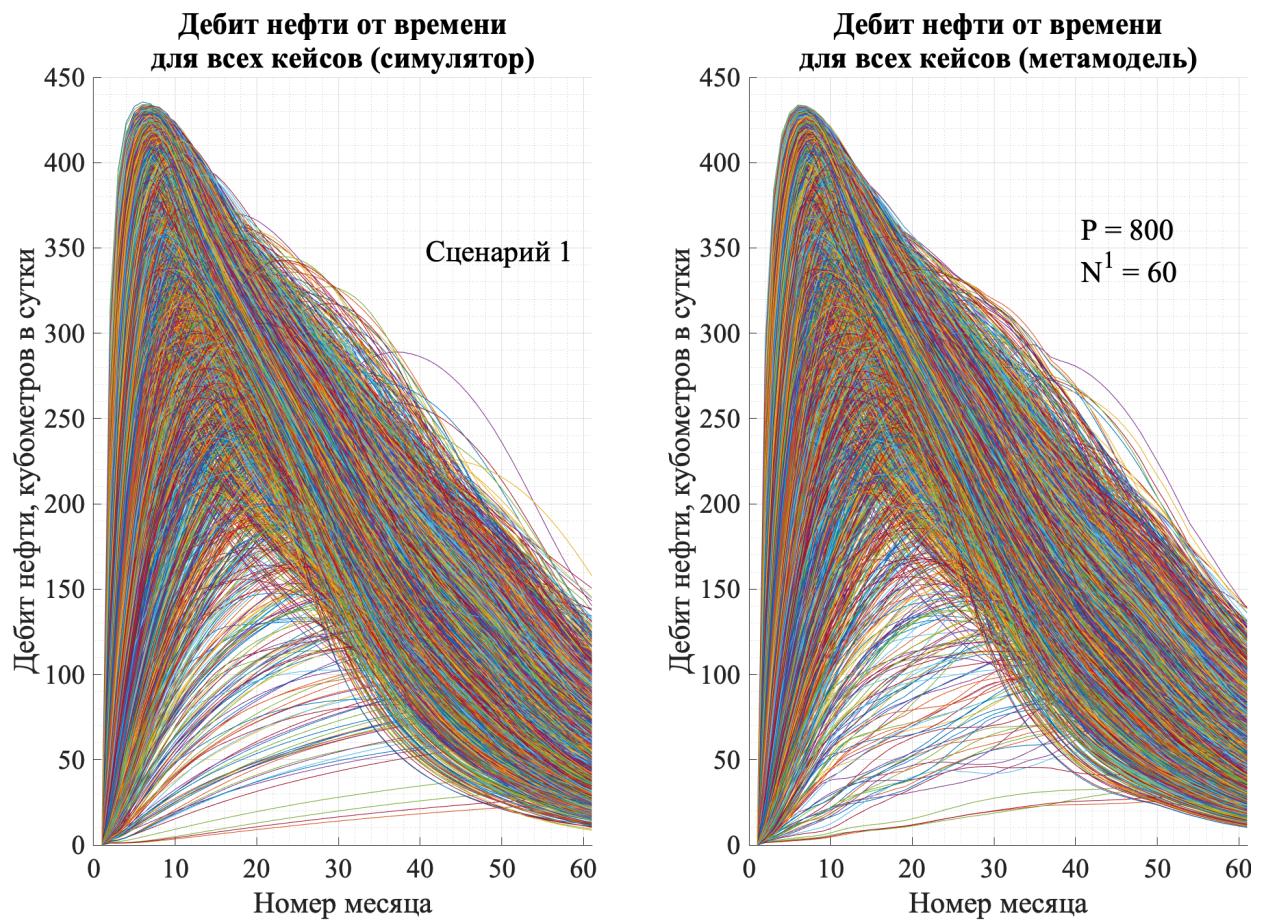


Рис. 31. Графики зависимости дебита нефти от времени для всех кейсов (сценарий 1)

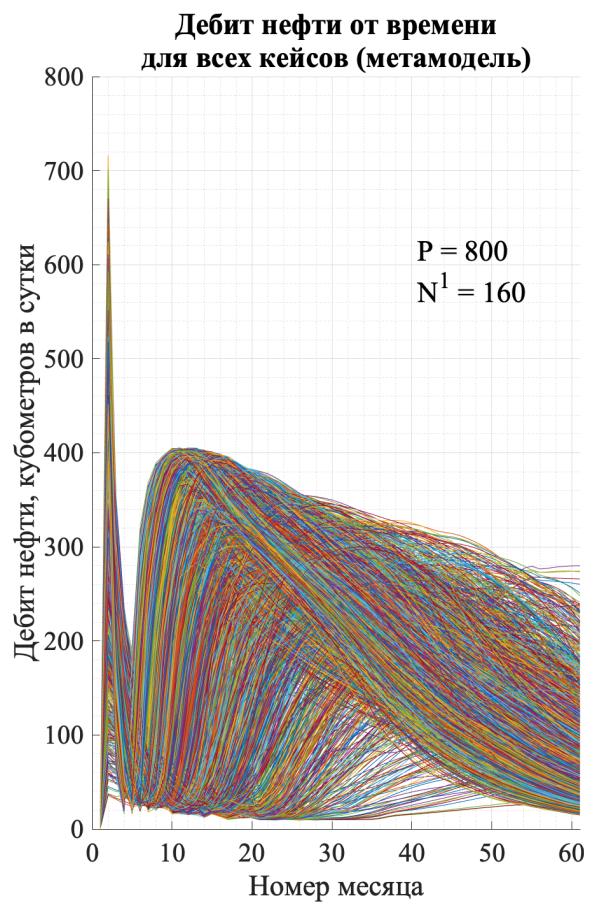
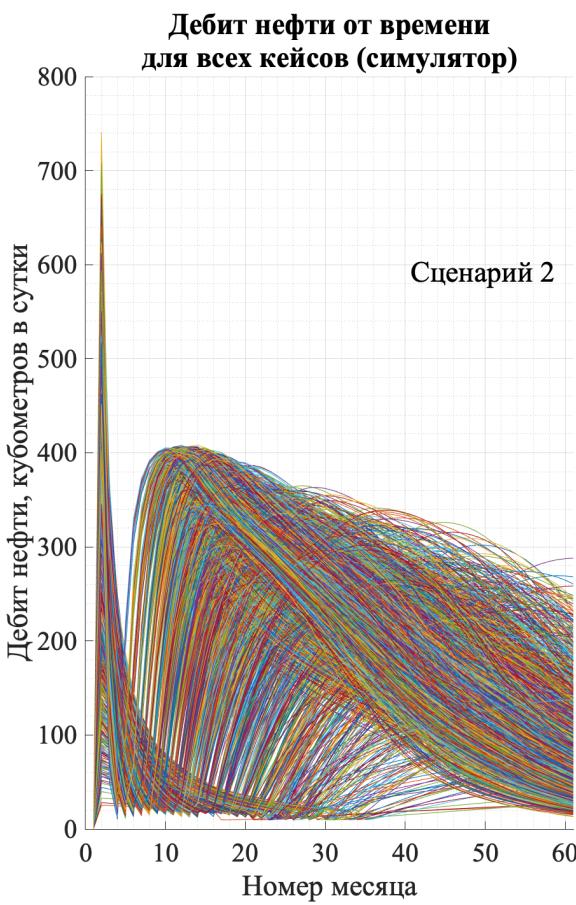


Рис. 32. Графики зависимости дебита нефти от времени для всех кейсов (сценарий 2)

3.3 Визуализация прогноза при изменении одного из входных параметров

Построены графики зависимости (рис. 33 – 40) дебитов нефти от времени при варьировании каждого из входных параметров метамодели: 3 параметра фиксируются, а четвёртый варьируется с постоянным шагом.

Значения фиксированных параметров представлены в таблице 2.

Таблица 2. Значения фиксированных параметров

Пористость матрицы	Проницаемость матрицы, мД	Пористость трещин	Проницаемость трещин, мД
0.2	5	0.02	1500

Код построения графиков представлен в приложении 11.

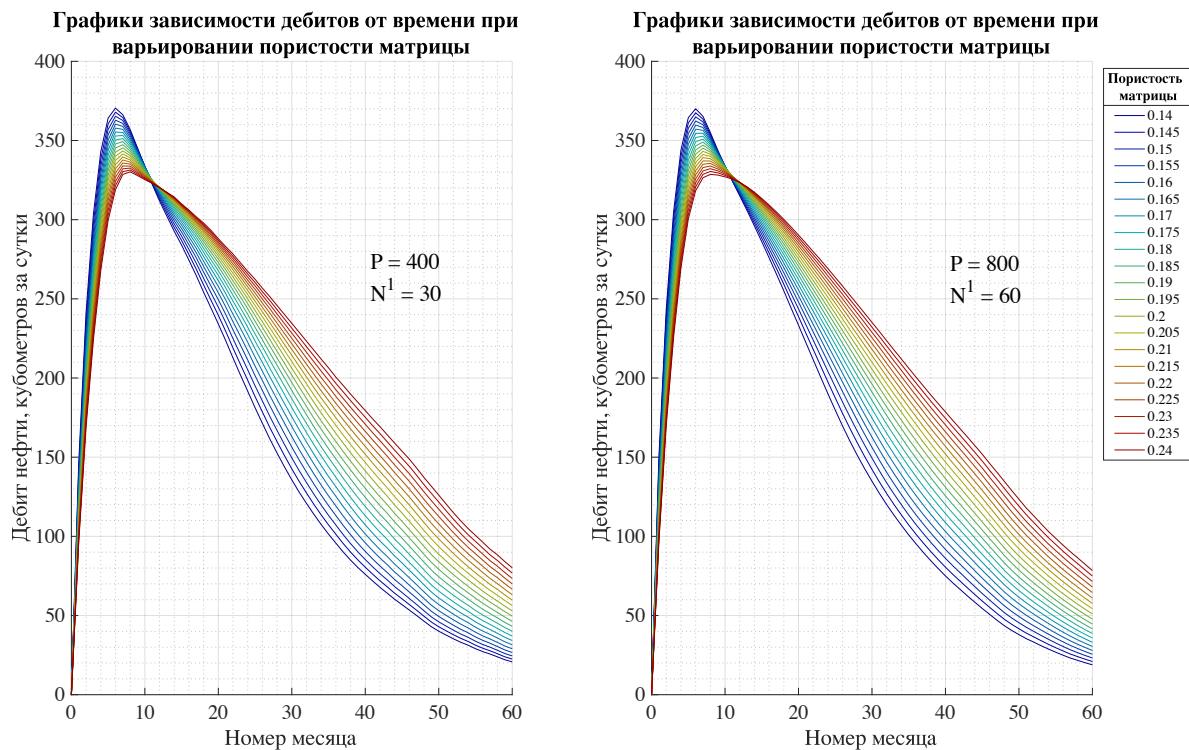


Рис. 33. Графики зависимости дебитов нефти от времени при варьировании пористости матрицы (сценарий 1)

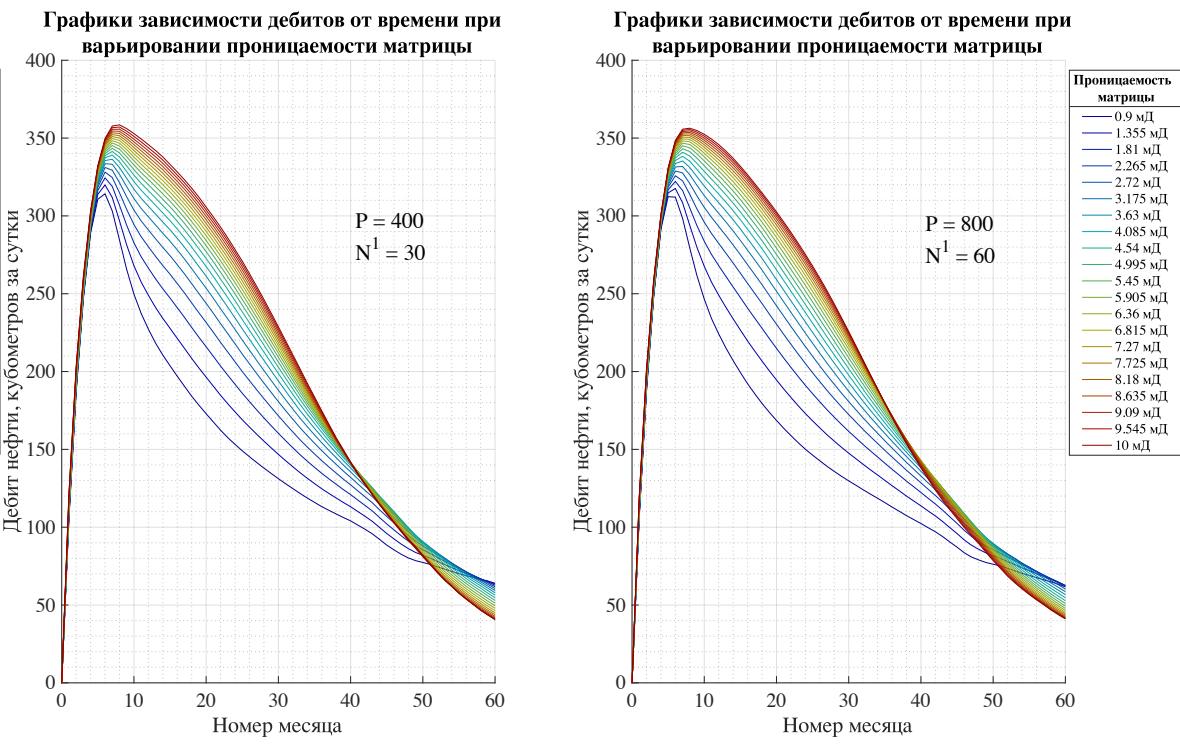


Рис. 34. Графики зависимости дебитов нефти от времени при варьировании проницаемости матрицы (сценарий 1)

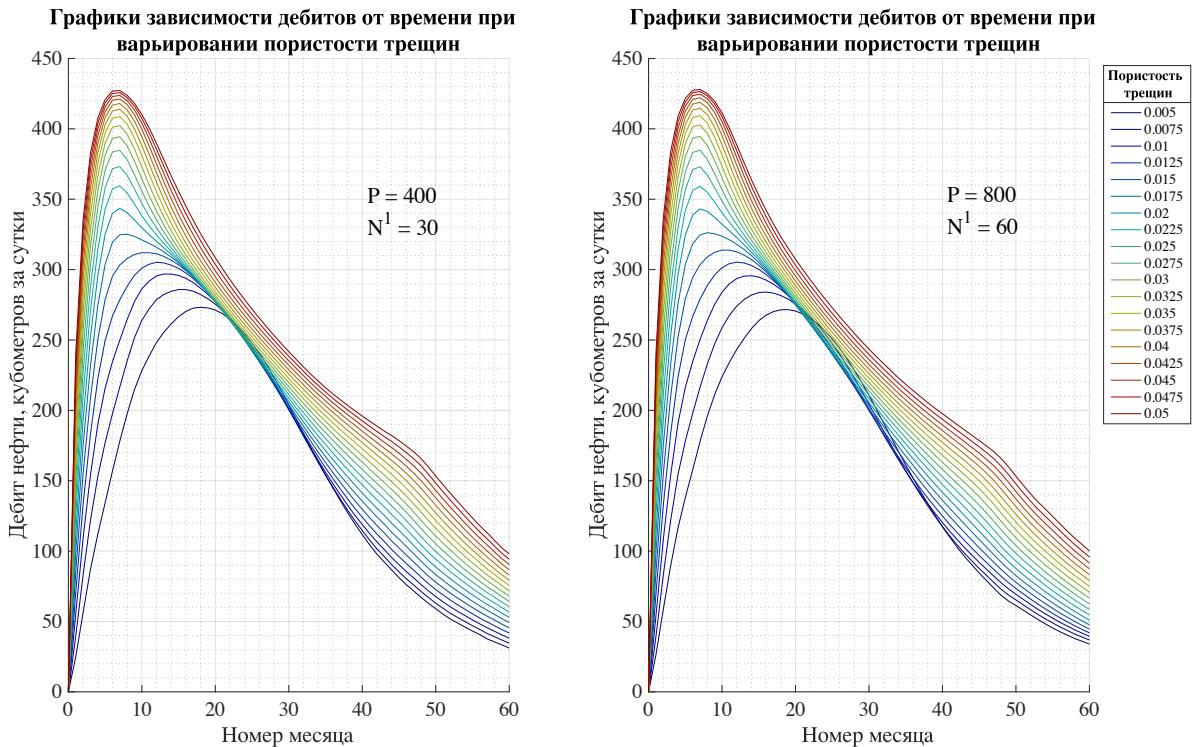


Рис. 35. Графики зависимости дебитов нефти от времени при варьировании пористости трещин (сценарий 1)

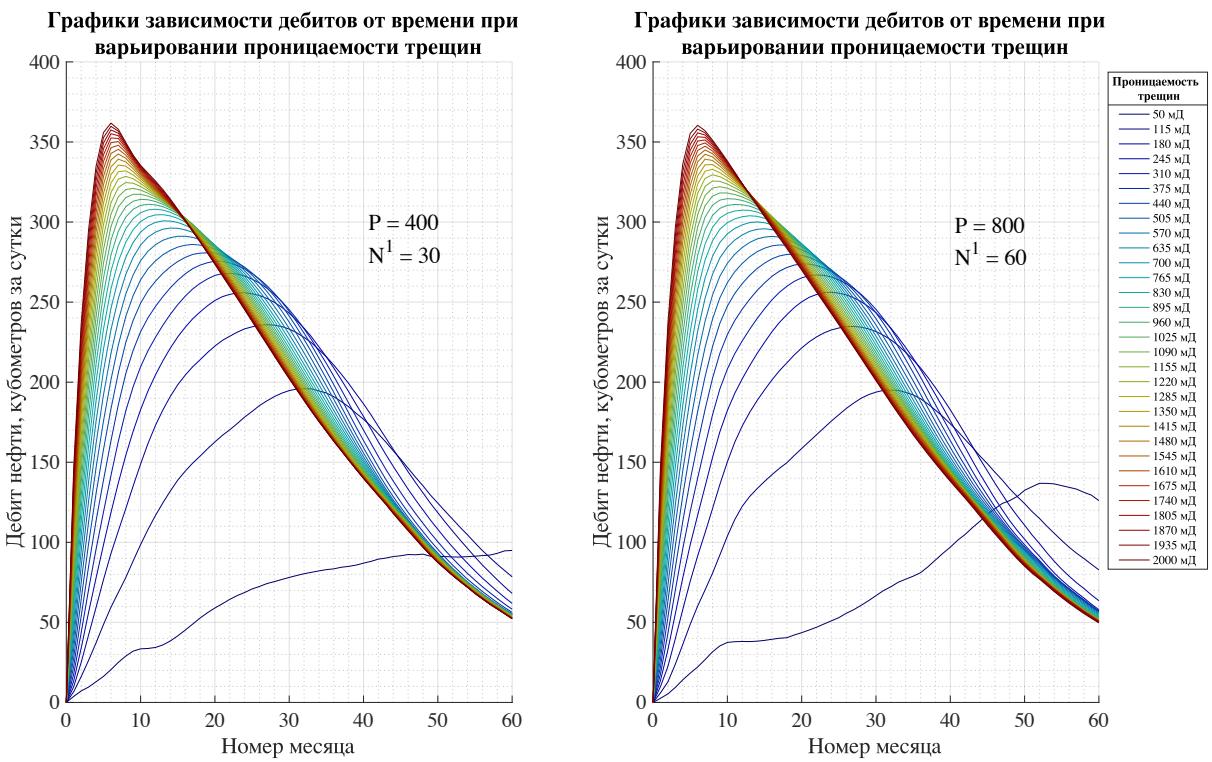


Рис. 36. Графики зависимости дебитов нефти от времени при варьировании проницаемости трещин (сценарий 1)

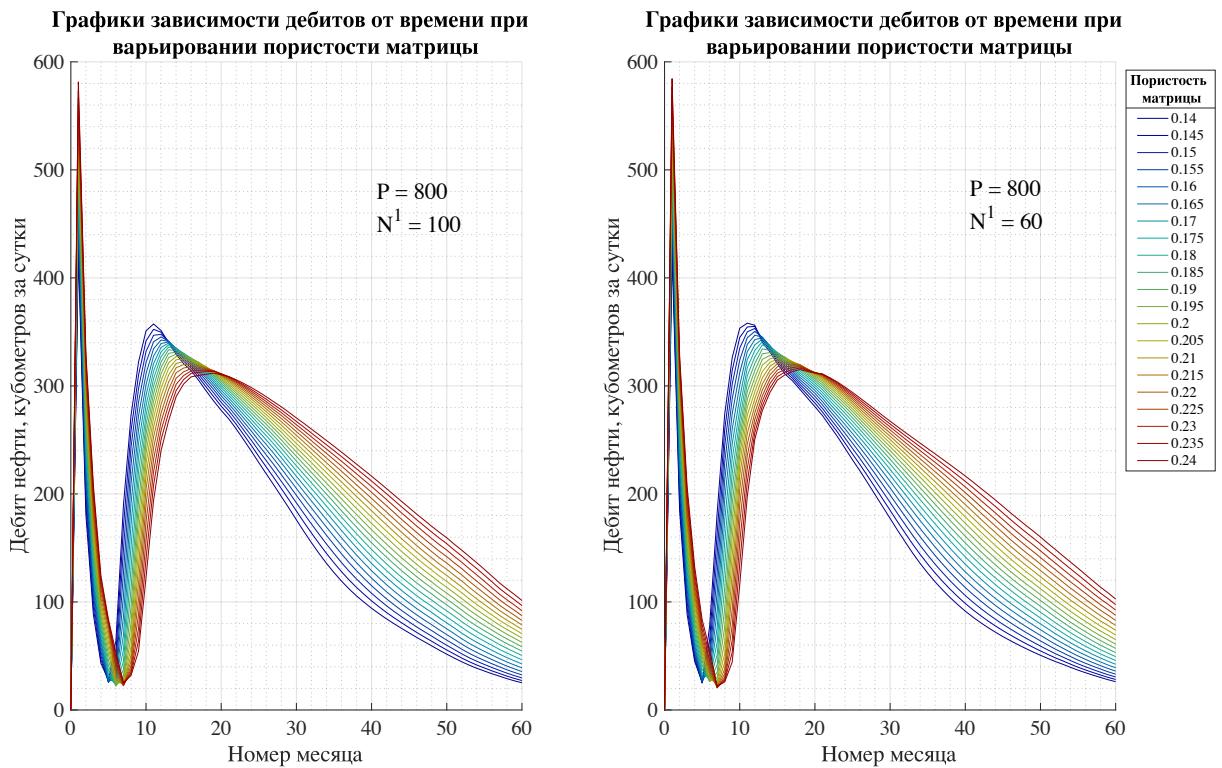


Рис. 37. Графики зависимости дебитов нефти от времени при варьировании пористости матрицы (сценарий 2)

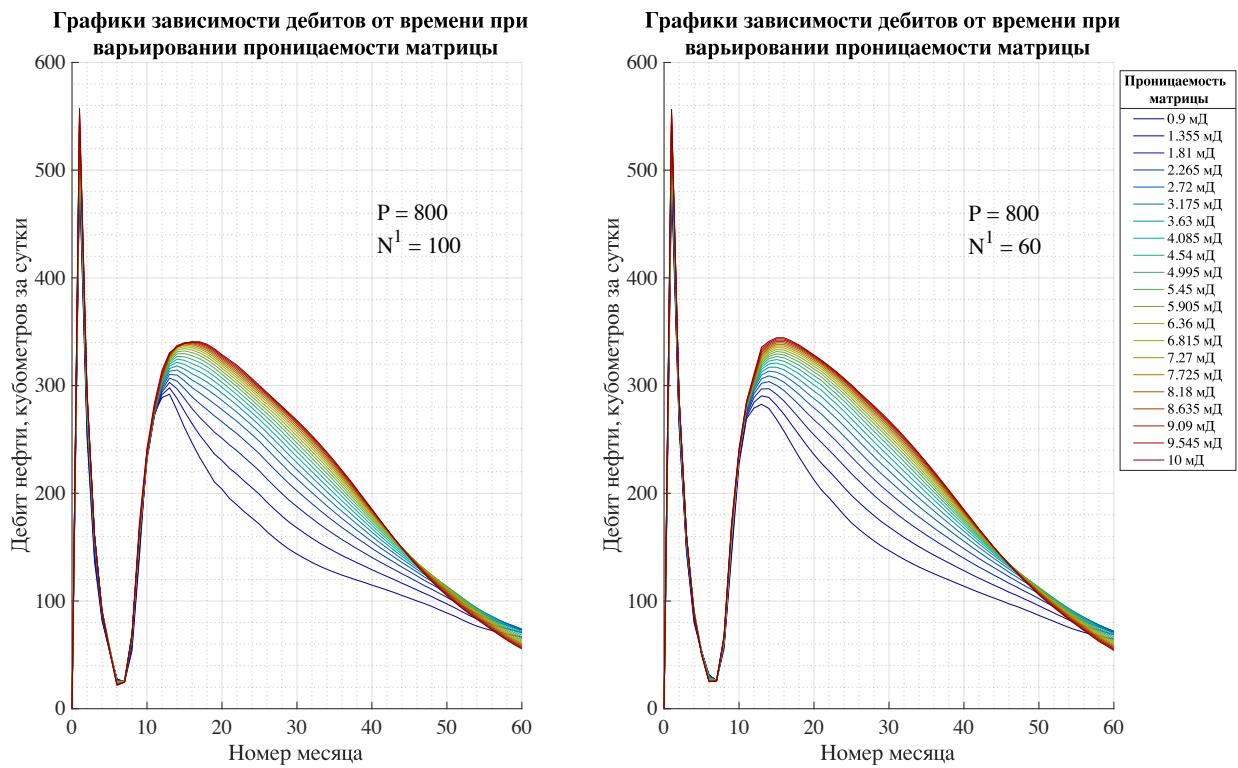


Рис. 38. Графики зависимости дебитов нефти от времени при варьировании проницаемости матрицы (сценарий 2)

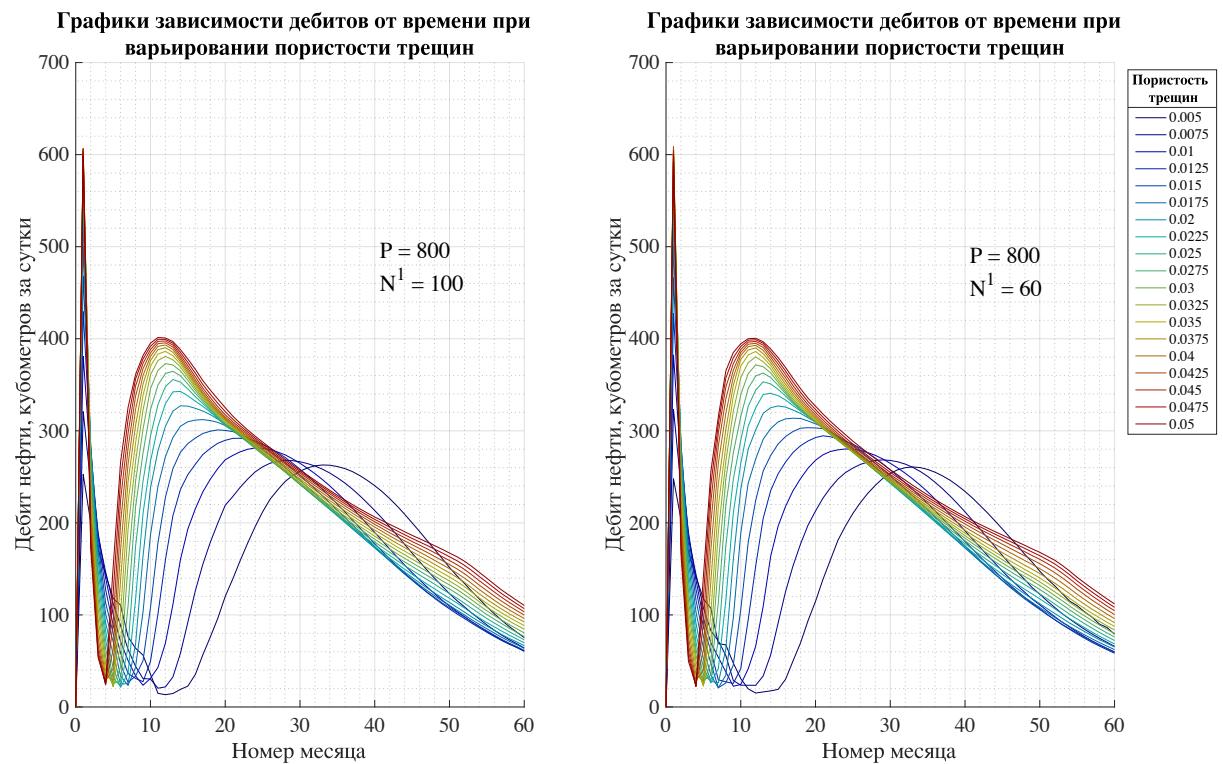


Рис. 39. Графики зависимости дебитов нефти от времени при варьировании пористости трещин (сценарий 2)

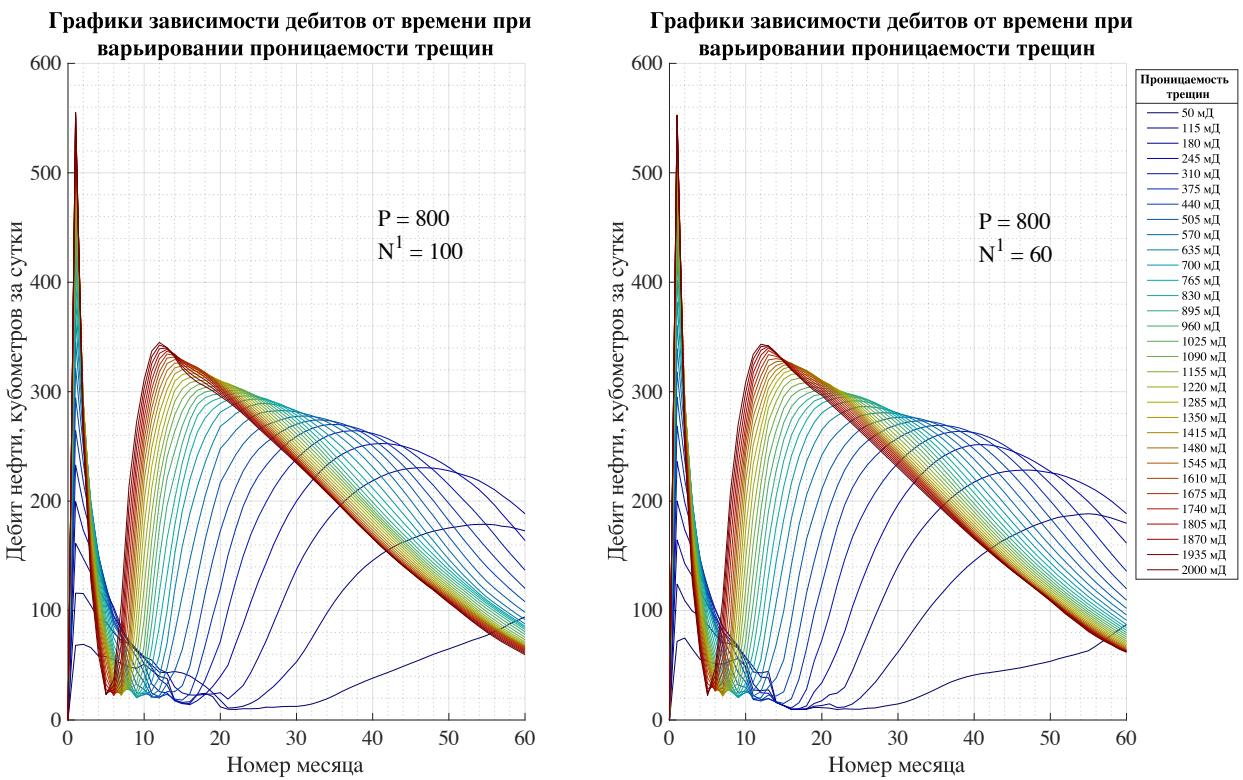


Рис. 40. Графики зависимости дебитов нефти от времени при варьировании проницаемости трещин (сценарий 2)

Также проведена анимация подобных графиков зависимости дебитов нефти от времени при варьировании всех входных параметров метамодели. Код приведён в приложении 12.

ГЛАВА 4. ОЦЕНКА НЕОПРЕДЕЛЁННОСТИ СУММЫ ДЕБИТОВ МЕТОДОМ МОНТЕ-КАРЛО

Оценка неопределённости методом Монте-Карло используется, когда неизвестны точные значения некоторых входных параметров модели, но известны распределения этих значений. Результатом оценки неопределённости является распределение выходного параметра модели.

Алгоритм:

- 1) случайно генерируются N сценариев (каждый из сценариев представляет собой вектор значений входных параметров, сгенерированных случайным образом по заданным распределениям);
- 2) проводится расчёт для каждого сценария;
- 3) строится распределение выходных данных, определяется среднее, минимальное и максимальное значения.

По представленному алгоритму проведена оценка неопределённости суммы дебитов нефти методом Монте-Карло. Расчёты проводились на основе построенных нейронных сетей. Код представлен в приложении 13.

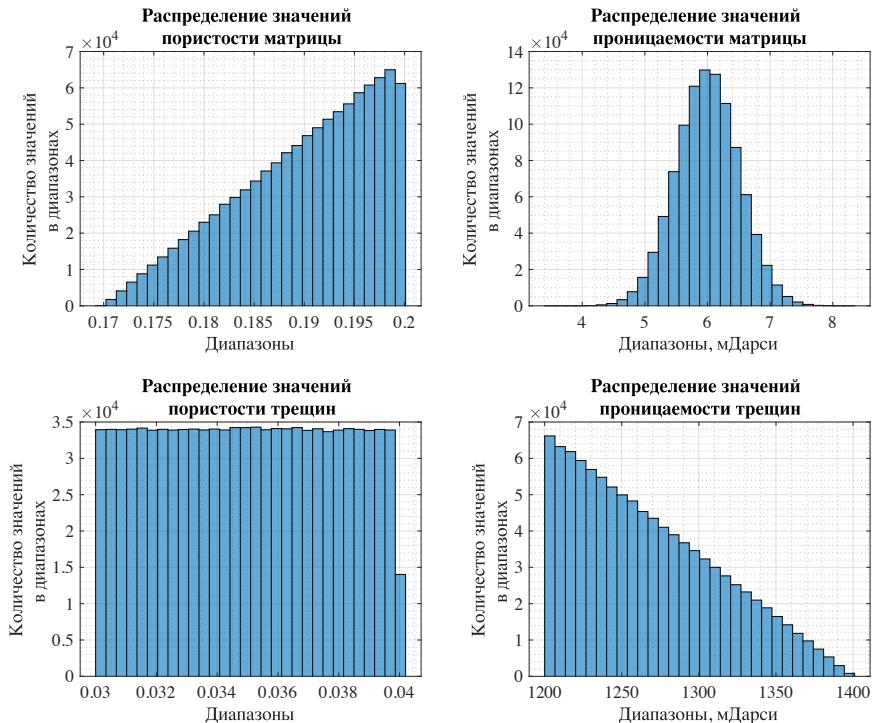


Рис. 41. Распределения входных параметров метамодели

Распределения входных параметров [27] представлены на рис. 41:

- 1) пористость матрицы – треугольное распределение [29] с параметрами распределения $a = 0.17, b = 0.2, c = 0.2$;
- 2) проницаемость матрицы – нормальное распределение [28] с параметрами распределения $mu = 6, sigma = 0.5$;
- 3) пористость трещин – равномерное распределение [30] с параметрами распределения $lower = 0.03, upper = 0.04$.
- 4) проницаемость трещин – треугольное распределение [29] с параметрами распределения $a = 1200, b = 1200, c = 1400$.

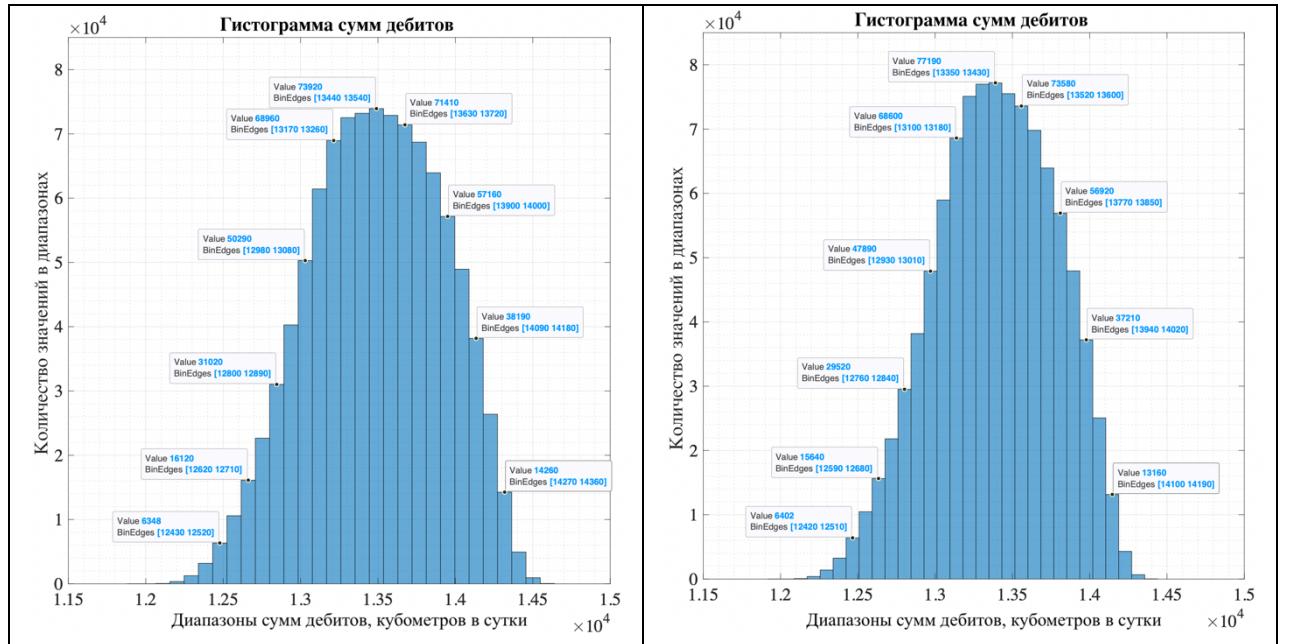


Рис. 42. Гистограммы сумм дебитов (сценарий 1 / сценарий 2)

Распределения выходного параметра (суммы дебитов) для двух сценариев изображены на рис. 42.

В таблице 3 при числе сценариев N представлены время расчёта с использованием метамодели t_{mm} и предполагаемое время расчёта на гидродинамическом симуляторе t_s . Среднее время расчёта одного кейса на симуляторе t_{sa} определено при расчёте кейсов, использованных для обучения метамодели.

Таблица 3. Время расчёта с помощью метамодели и на симуляторе

N	t_{sa} , с	t_{mm} , с	t_s , с
10^6	18	2.6	$18 \cdot 10^6$ (≈ 7 месяцев)

Таким образом, с помощью построенной метамодели возможно практически мгновенно проводить оценку неопределённости, а на гидродинамическом симуляторе подобные расчёты занимают продолжительное время.

ЗАКЛЮЧЕНИЕ

В данной работе были построены метамодели притока флюида к вертикальной скважине. Для обучения метамоделей использовались данные, полученные при расчёте на гидродинамическом симуляторе ECLIPSE Blackoil. Представлено 2 подхода к построению метамоделей: с помощью градиентного бустинга на основе деревьев регрессии и с помощью двухслойного перцептрона. Проведена визуализация точности прогноза построенных метамоделей. Нейронная сеть применена для оценки неопределённости результата расчёта.



Рис. 43. Система для построения метамоделей

Реализована система для построения метамоделей (рис. 43) с безграничными возможностями расширения. Возможные варианты расширения:

- 1) строить разные гидродинамические модели пласта в симуляторах ECLIPSE Blackoil, tNavigator и т. п.;
- 2) рассмотреть не только вертикальные, но и горизонтальные скважины;

- 3) увеличить количество варьируемых параметров (например, исследовать характер зависимостей дебита нефти от времени при варьировании параметров закачки или при варьировании физико-химических параметров флюидов, учитывая при этом корреляции между ними, определённые при проведении физических экспериментов [11]);
- 4) изменить способ генерации обучающей выборки (при выборе дополняемых в процессе обучения выборок [12], [13], [17] необходимо пересмотреть структуру всей системы для построения метамоделей);
- 5) использовать более современные технологии машинного обучения для построения архитектуры и обучения метамоделей; например, предсказывать поле насыщенностей (рис. 44), но перед обучением сжимать выходные данные из UNRST файлов [22];

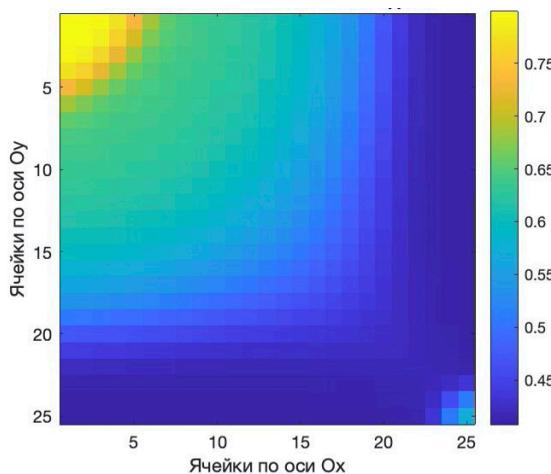


Рис. 44. Распределение водонасыщенности в одном из горизонтальных слоёв гидродинамической модели

- 6) придумать новые способы визуализации точности прогноза и результатов;
- 7) применить построенные метамодели для решения различных практических задач: оптимизации, генерации результатов при наложенных связях между входными параметрами и т. п.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] Баранов В. Е., Куреленков С. Х., Воробьёва Л. В. – Прикладное моделирование пласта – Издательство Томского Политехнического университета, 2012
- [2] Баренблatt Г. И. – Анализ размерностей. Учебное пособие – Москва, 1987
- [3] Баренблatt Г. И., Ентов В. М., Рыжик В. М. – Движение жидкостей и газов в природных пластах – Москва «Недра», 1984
- [4] Каспарьян Э. В., Козырев А. А., Иофис М. А., Макаров А. Б. – Геомеханика – Москва «Высшая школа», 2006
- [5] Мирзаджанзаде А. Х., Хасанов М. М., Бахтизин Р. Н. – Моделирование процессов нефтегазодобычи. Нелинейность, неравновесность, неопределенность – Москва, Ижевск, 2005
- [6] Седов Л. И. – Методы подобия и размерности в механике – Москва «Наука», 1977
- [7] Симонов М., Ахметов А., Темирчев П., Коротеев Д., Костоев Р., Бурнаев Е., Оседецов И. – Применение технологий машинного обучения для быстрого 3D моделирования притока к скважине в элементе разработки – Общество инженеров нефтегазовой промышленности, SPE-191593-18RPTC-RU, 2018
- [8] Brownlee J. – Discover Feature Engineering, How to Engineer Features and How to Get Good at It, Machine Learning Mastery:
<https://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/>
- [9] Brownlee J. – XGBoost With Python. Gradient Boosted Trees with XGBoost and scikit-learn. Machine Learning Mastery, 2018
- [10] Burkardt J. – Improved Hypercube Sampling:
https://people.math.sc.edu/Burkardt/m_src/ihc/ihc.html
- [11] Cosse R. – Basics of Reservoir Engineering. Oil and Gas Field Development Techniques, 1993

- [12] Crombecq K. – Surrogate Modelling of Computer Experiments with Sequential Experimental Design – University of Antwerp, Ghent University, 2011
- [13] Gorissen D. – Grid-Enabled Adaptive Surrogate Modeling for Computer Aided Engineering – University of Antwerp, Ghent University, 2010
- [14] Hagan et al. – Neural Network Design, 2nd Edition, eBook
- [15] Mohaghegh S. – Proxy Modeling for Fluid Flow in Porous Media (articles):
<https://shahab.pe.wvu.edu/research/RefereedJournals.htm>
- [16] Moller, M. – A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning. *Neural Networks*, Vol. 6, 1993, pp. 525-533
- [17] Mueller et al. – Algorithm Development for Black-Box Global Optimization, Berkeley Lab: <https://ccse.lbl.gov/Research/Optimization/index.html>
- [18] Pan G., Ye P., Wang P. – A Novel Latin Hypercube Algorithm via Translational Propagation, *ScientificWorldJournal*, 2014. Published online 2014 Sep 2, doi: 10.1155/2014/163949
- [19] Temirchev P., Simonov M., Kostoev R., Burnaev E., Oseledets I., Akhmetov A., Margarit A., Sitnikov A., Koroteev D. – Deep neural networks predicting oil movement in a development unit. *Journal of Petroleum Science and Engineering*. Available online 20 September 2019
- [20] Viana F., Venter G., Balabanov V. – An Algorithm for Fast Optimal Latin Hypercube Design of Experiments. *International Journal for Numerical Methods in Engineering*, 2010
- [21] Код для считывания данных из файлов с расширением RSM:
<https://www.mathworks.com/matlabcentral/answers/452195-reading-content-of-a-text-file-by-readtable>
- [22] Код для считывания данных из файлов с расширением UNRST:
https://www.mathworks.com/matlabcentral/fileexchange/47607-read_ecl-m
- [23] Код для увеличения размера шрифта на графиках:
<https://www.mathworks.com/matlabcentral/fileexchange/67644-supersizeme>
- [24] ECLIPSE 100. Справочное руководство

[25] ECLIPSE. Version 2017.1. Technical Description

[26] MATLAB Shallow Neural Network Toolbox Manual:

<http://matlab.izmiran.ru/help/toolbox/nnet/>

[27] MATLAB Documentation, Create probability distribution object:

<https://www.mathworks.com/help/stats/makedist.html>

[28] MATLAB Documentation, Normal Distribution:

<https://www.mathworks.com/help/stats/prob.normaldistribution.html>

[29] MATLAB Documentation, Triangular Distribution:

<https://www.mathworks.com/help/stats/prob.triangulardistribution.html>

[30] MATLAB Documentation, Uniform Distribution:

<https://www.mathworks.com/help/stats/prob.uniformdistribution.html>

ПРИЛОЖЕНИЕ 1

Файл-шаблон DATA (сценарий 2)

Текст файла-шаблона (расширение DATA), содержащий информацию о построенной гидродинамической модели и подаваемый на расчёт в ECLIPSE. Сценарий разработки №2: сначала обе скважины - добывающие и работают в естественном режиме. Затем, после падения суммарного дебита по скважинам до 20 кубометров в сутки, одна из скважин переводится в режим нагнетания воды с ПАВ.

```
--Первый раздел файла входных данных
RUNSPEC
-- Название
TITLE
DPDP
-- Количество ячеек в направлениях X,Y,Z
DIMENS
    25    25    40 /
-- В данной задаче используется модель двойной пористости
DUALPORO
-- В данной задаче используется модель двойной проницаемости
DUALPERM
-- В данной задаче присутствует нефтяная фаза
OIL
-- В данной задаче присутствует водная фаза
WATER
-- В данной задаче присутствует ПАВ
SURFACT
-- Метрические единицы измерения
METRIC
-- Размерности таблиц
TABDIMS
    2    1    13   13    1   20 /
-- Размерности регионов
REGDIMS
    2    1    0    0    / 
-- Размерности массивов, использующихся для описания скважин и их групп
WELLDIMS
    3    55    3    3    /
-- Число итераций при пересчете показателей скважин
NUPCOL
    4    /
-- Дата начала моделирования
START
    1 'JAN' 2021 /
-- Размер стека, используемого при решении системы линейных уравнений
NSTACK
```

```

8 /
-- Выходные файлы унифицированы
UNIFOUT
-- Геометрия используемой для моделирования сетки, а также различные
-- свойства породы для
-- каждого сеточного блока
GRID =====
-- Ввод сеточных данных только для ячеек матрицы
DPGRID
-- Присваивает значение заданной константы элементам заданного массива
внутри текущего бокса
EQUALS
-- Размеры блоков сетки в направлении X (горизонтальном) для текущего
входного бокса
'DX' 20.000000 , , , , , ,
-- Размеры блоков сетки в направлении Y (горизонтальном) для текущего
входного бокса.
'DY' 20.000000 , , , , , ,
-- Значения глубин верхних граней блоков сетки для текущего бокса.
'TOPS' 2600 , 1, 25 , 1 , 25 , 1, 1 /
-- Размеры блоков сетки в направлении Z (вертикальном) для текущего
входного бокса
'DZ' 1.0 , 1, 25 , 1, 25 , 1, 20 /
-- Отношение нефтенасыщенной мощности к общей мощности для ячеек текущего
входного бокса.
'NTG' 1.0000000 , 1, 25 , 1, 25 , 1, 20 /
-- Пористость матрицы
'PORO' 0.19373 /
-- Проницаемость по X матрицы
'PERMX' 4.6128 /
-- Проницаемость по Y матрицы
'PERMY' 4.6128 /
-- Проницаемость по Z матрицы
'PERMZ' 0.30 /
-- Значения глубин верхних граней блоков сетки для текущего бокса.
'TOPS' 2600 , 1, 25 , 1 , 25 , 21, 21 /
-- Размеры блоков сетки в направлении Z (вертикальном) для текущего
входного бокса
'DZ' 1.0 , 1, 25 , 1 , 25 , 21, 40/
-- Отношение нефтенасыщенной мощности к общей мощности для ячеек текущего
входного бокса.
'NTG' 1.00 , 1, 25 , 1, 25 , 21, 40 /
-- Пористость трещин
'PORO' 0.026637 /
-- Проницаемость по X трещин
'PERMX' 1461.7633 /
-- Проницаемость по Y трещин
'PERMY' 1461.7633 /
-- Проницаемость по Z трещин
'PERMZ' 300.0000 /
/
-- Устанавливает данные раздела GRID подлежащие выводу
RPTGRID
  'DZ'
  'PERMX'
  'PERMY'

```

```

'PORO'
'TOPS'
/
-- Коэффициент сигма взаимосвязи матриц и трещин для всей сетки
SIGMA
0.12 /
-- Устанавливает поля входного файла данных
COLUMNS
1 132 /
-- Контролирует вывод данных
DEBUG
6*0 1 0 /
-- Создает исходный файл для пакета GRAF
INIT
/
-- Раздел EDIT включает в себя команды для модификации поровых объемов
значений
-- глубин центров блоков, проводимостей и др.
EDIT
-- Переопределяет текущий входной бокс
BOX
1 25 1 1 20 20 /
-- Умножает вычисляемый поровый объем внутри текущего бокса
MULTPV
25*1000 /
/
-- Закрывает входной бокс и определяет всю сетку в качестве текущего
входного бокса
ENDBOX
/
BOX
1 25 25 25 20 20 /
-- Умножает вычисляемый поровый объем внутри текущего бокса
MULTPV
25*1000 /
/
ENDBOX
/
BOX
1 1 2 24 20 20 /
MULTPV
23*1000 /
/
ENDBOX
/
BOX
25 25 2 24 20 20 /
MULTPV
23*1000 /
/
ENDBOX
/
-- Раздел входных данных PROPS содержит свойства пластовых флюидов и
породы, зависящие от
-- давления и насыщенности
PROPS =====

```

```

-- Относительная проницаемость воды и капиллярное давление как функции Sw.
SWFN
  .3000  .0000  0.135
  .5000  .0200  0.068
  .7000  .2000  0.027
  1.0    1.0    .0
/
  .0000  .0000  0.0000
  2600.0 1.0000  .0000
/
-- Относительная проницаемость нефти как функция So
SOF2
  .2000  .0000
  .7000  1.0
/
  .0000  .0000
  2600.0 1.0000
/
-- Объемный коэффициент, сжимаемость и вязкость воды
PVTW
  270 1.030 4.6E-5 0.34 0.0 /
-- Объемный коэффициент и вязкость дегазированной нефти как функции
давления
PVDO
  200    1.0    0.47
  280    0.999   0.47
  300    0.998   0.47
/
-- Сжимаемость породы
ROCK
  270 .3E-5 /
-- Плотности флюидов в стандартных условиях
DENSITY
  850. 1000. 10. /
-- Задает связь вязкости раствора от концентрации ПАВ
SURFVISC
  0.0    0.34
  30.    5.0   /
-- Задает связь адсорбции ПАВ на горной породе и концентрации ПАВ
SURFADS
  0.0    0.0000
  1.0    0.0005
  30.0   0.0005 /
  0.0    0.0000
  1.0    0.0005
  30.0   0.0005 /
-- Задает связь поверхностного натяжения на границе вода-нефть и
концентрации ПАВ
SURFST
  0.0    0.05
  1.0    1.0E-6
  30.0   1.0E-6 /
-- Определяет свойства изменения капиллярного давление для модели ПАВ
SURFCAPD
  -9    0.0

```

```

-4.5 0.0
-2    1.0
10    1.0 /
              

-9     0.0
-4.5   0.0
-2     1.0
10    1.0 /
-- Определяет свойства породы для модели ПАВ
SURFROCK
2600 2650 /
2 2650 /
/

-- В разделе REGIONS выделяются области сетки для
-- задания функций насыщенности 'SATNUM' (относительных проницаемостей и
-- капиллярных давлений), подсчета запасов 'FIPNUM' и др.
REGIONS =====
-- 
EQUALS
'SATNUM ' 1 , 1, 25 , 1, 25 , 1, 20 /
'SATNUM ' 2 , 1, 25 , 1, 25 , 21, 40 /
'FIPNUM ' 1 , 1, 25 , 1, 25 , 1, 20 /
'FIPNUM ' 2 , 1, 25 , 1, 25 , 21, 40 /
'SURFNUM ' 2 , 1, 25 , 1, 25 , 1, 20 /
'SURFNUM ' 2 , 1, 25 , 1, 25 , 1, 40 /
/
-- Раздел SOLUTION содержит необходимые данные для задания начального
-- состояния (давление, насыщенности) каждого блока сетки
-- моделируемого пласта.
SOLUTION =====
-- Глубины межфлюидных контактов и другие параметры равновесия
EQUIL
2600 230 2618 /
-- Вывод данных раздела SOLUTION
RPTSQL
'PRES' 'SWAT'
/
-- 
RPTRST
PRESSURE SWAT /
-- В разделе SUMMARY указываются переменные, которые должны быть записаны
-- в Summary файлы после каждого временного шага моделирования
SUMMARY =====
-- Забойное давление
WBHP
/
-- Темп отбора нефти в скважине (дебит скважины)
WOPR
/
-- Темп отбора воды в скважине (дебит скважины)
WwPR
/
-- Скорость закачки воды по месторождению
FWIR
-- Дебит нефти по месторождению

```

FOPR
 -- Накопленная добыча нефти по месторождению
FOPT
 -- Накопленная добыча воды по месторождению
FWPT
 -- Дебит воды по месторождению
FWPR
 -- Дебит жидкости по месторождению
FLPR
 -- Накопленная добыча жидкости по месторождению
FLPT
 -- Пластовое давление
FPR
 -- Поровый объем, содержащий нефть
FOPV
 -- Поровый объем, содержащий воду
FWPV
 -- Задает табличный вывод данных из summary-файлов
 -- в файл печати в конце расчета
RUNSUM
 -- Задает выдачу итогов расчета в RSM файл
SEPARATE
 -- Раздел SCHEDULE содержит данные о режимах работы скважин в процессе
 -- моделирования (установленные значения управляющих параметров и
 -- пределы их изменения) и моменты времени выдачи промежуточных
 -- отчетов
SCHEDULE ======
 -- Параметры вывода результатов моделирования
RPTSCHED
 'PRES' 'SWAT' 'RESTART=1'
 /
 --
MESSAGES
 9* 777 3* /
 /
 --
RPTONLY
 -- ввод новых скважин
WELSPECS
 '1' 'G' 1 1 2600 'OIL' /
 '2' 'G' 25 25 2600 'OIL' /
 'INJ' 'G' 1 1 2600 'WAT' /
 /
 -- Определение связи между скважинами и
 -- блоками сетки, вскрываемыми ими
COMPDAT
 '1' ' 1 1 21 31 'OPEN' 0 .0 157E-3 /
 '2' ' 25 25 21 31 'OPEN' 0 .0 157E-3 /
 'INJ' ' 1 1 31 40 'OPEN' 0 .0 157E-3 /
 /
 -- Управления добывающей скважиной
WCONPROD
 '1' 'OPEN' 'BHP' 5* 40 3* 6* 1* /
 '2' 'OPEN' 'BHP' 5* 40 3* 6* 1* /
 /
 -- Управления нагнетательной скважиной

```
WCONINJE
'INJ' 'WATER' 'SHUT' 'RESV' 1* 500 1500.0          6* /
/
-- Экономические ограничения на дебиты скважин
WECON
'1' 10 1* 0.9 1* 1* WELL NO 'INJ' /
/
-- Задает концентрацию ПАВ нагнетаемом потоке в скважине
WSURFACT
'INJ' 12.0 /
/
-- Интервалы до следующего момента времени выдачи
-- результатов расчетов
TSTEP
60*30 /
-- Останавливает расчет
END
```

ПРИЛОЖЕНИЕ 2

Файл-шаблон DATA (сценарий 1)

Текст файла-шаблона (расширение DATA), содержащий информацию о построенной гидродинамической модели и подаваемый на расчёт в ECLIPSE. Сценарий разработки №1: изначально давление в пласте низкое, поэтому в одной из скважин для поддержания давления с самого начала расчёта производится нагнетание воды с ПАВ.

```
RUNSPEC
TITLE
DPDP
DIMENS
    25    25    40   /
DUALPORO
DUALPERM
OIL
WATER
SURFACT
METRIC
TABDIMS
    2    1    13    13    1    20   /
REGDIMS
    2    1    0    0   /
WELLDIMS
    3    55    3    3   /
NUPCOL
    4   /
START
    1 'JAN' 2021   /
NSTACK
    8   /
UNIFOUT
GRID      =====
DPGRID
EQUALS
'DX'      ' 20.000000 , , , , , , /'
'DY'      ' 20.000000 , , , , , , /'
'TOPS'    ' 2600 , 1, 25 , 1 , 25 , 1, 1 /'
'DZ'      ' 1.0 , 1, 25 , 1, 25 , 1, 20 /'
'NTG'     ' 1.0000000 , 1, 25 , 1, 25 , 1, 20 /'
'PORO'    0.1755   /
'PERMX'   3.5617   /
'PERMY'   3.5617   /
'PERMZ'   0.30   /
'TOPS'    ' 2600 , 1, 25 , 1 , 25 , 21, 21 /'
'DZ'      ' 1.0 , 1, 25 , 1 , 25 , 21, 40 /'
'NTG'     ' 1.00 , 1, 25 , 1, 25 , 21, 40 /'
```

```

'PORO'  0.01445   /
'PERMX'  1668.5   /
'PERMY'  1668.5   /
'PERMZ'  300.000  /
/
RPTGRID
    -- Report Levels for Grid Section Data
    --  'DX'
    --  'DY'
    'DZ'
    'PERMX'
    'PERMY'
    --  'MULTX'
    --  'MULTY'
    'PORO'
    'TOPS'
    --  'PORV'
    --  'DEPTH'
    --  'TRANX'
    --  'TRANY'
    --  'ALLNNC'
    /
--SIGMA FOR 10 BY 10 BY 30 FT BLOCKS
SIGMA
    0.12   /

COLUMNS
2 132  /

COLUMNS
1 132  /

DEBUG
6*0 1 0  /

INIT
/

EDIT
BOX
1 25 1 1 20 20  /

MULTPV
25*1000 /
/
ENDBOX
/
BOX
1 25 25 25 20 20  /

MULTPV
25*1000 /
/
ENDBOX
/
BOX

```

```

1 1 2 24 20 20 /

MULTPV
23*1000 /
/
ENDBOX
/
BOX
25 25 2 24 20 20 /

MULTPV
23*1000 /
/
ENDBOX
/
PROPS      =====
SWFN
    .3000  .0000  0.135
    .5000  .0200  0.068
    .7000  .2000  0.027
    1.0     1.0     .0
/
    .0000  .0000  0.0000
    1.0000 1.0000  .0000
/
SOF2
    .2000  .0000
    .7000  1.0
/
    .0000  .0000
    1.0000 1.0000
/
PVTW
    270 1.030 4.6E-5 0.34 0.0 /
PVDO
    200     1.0     0.47
    280     0.999   0.47
    300     0.998   0.47
/
ROCK
    270 .3E-5 /
DENSITY
    850. 1000. 10. /
SURFVISC
    0.0     0.34
    30.     5.0    /
SURFADS
    0.0     0.0000
    1.0     0.0005
    30.0    0.0005 /
    0.0     0.0000
    1.0     0.0005

```

```

30.0      0.0005  /

SURFST
0.0      0.05
1.0      1.0E-6
30.0     1.0E-6  /

SURFCAPD
-9       0.0
-4.5     0.0
-2       1.0
10      1.0  /

-9       0.0
-4.5     0.0
-2       1.0
10      1.0  /

SURFROCK
1 2650 /
2 2650 /

--RPTPROPS
--    -- PROPS Reporting Options
--
--'SURFVISC'
/
/
REGIONS      =====

EQUALS
'SATNUM   ' 1 , 1, 25 , 1, 25 , 1, 20 /
'SATNUM   ' 2 , 1, 25 , 1, 25 , 21, 40 /
'FIPNUM   ' 1 , 1, 25 , 1, 25 , 1, 20 /
'FIPNUM   ' 2 , 1, 25 , 1, 25 , 21, 40 /
--'ROCKNUM ' 1 , 1, 6 , 1, 6 , 1, 3 /
--'ROCKNUM ' 2 , 1, 6 , 1, 6 , 4, 6 /
'SURFNUM   ' 2 , 1, 25 , 1, 25 , 1, 40 /
/

SOLUTION      =====
EQUIL
2600 50 2618 /
--RPTSOL
--'PRES' 'SOIL' 'SWAT' 'RESTART=1' 'OILAPI' 'FIPTR=2' 'TBLK' 'FIPPLY=2'
-- /
RPTSOL
'PRES'  'SWAT'
/
RPTRST
PRESSURE SWAT /
-- RPTRST
-- PRESSURE SOIL SWAT /

```

```

SUMMARY      =====
WBHP
/
WOPR
/
WWPR
/
FWIR
FWIT
FOPR
FOPT
FWPT
FWPR
FLPR
FLPT
FPR
FOPV
FWPV
RUNSUM
SEPARATE
SCHEDULE   =====
--RPTSCHED
--'PRES' 'SOIL' 'SWAT' 'RESTART=2' 'FIP=2' 'WELLS=2' 'SUMMARY=2'
-- /
RPTSCHED
'PRES' 'SWAT' 'RESTART=1'
/
MESSAGES
 9* 777 3* /

-- RPTRST
-- SURFBLK VISC /
/
RPTONLY
WELSPECS
--'1' 'G'    1 1 2600 'OIL' /
'2' 'G'    25 25 2600 'OIL' /
'INJ' 'G'    1 1 2600 'WAT' /
/

COMPDAT
--'1'      ' 1 1 21 31 'OPEN' 0 .0 157E-3 /
'2'      ' 25 25 21 31 'OPEN' 0 .0 157E-3 /
'INJ'     ' 1 1 31 40 'OPEN' 0 .0 157E-3 /
/
WCONPROD
--'1' 'OPEN' 'BHP' 5* 40 3* 6* 1* /
'2' 'OPEN' 'BHP' 5* 40 3* 6* 1* /
/
WCONINJE
'INJ' 'WATER' 'OPEN' 'RESV' 1* 500 1500.0 6* /
/
--WECON
--'1' 10 1* 0.9 1* 1* WELL NO 'INJ' /
--/

```

```
WSURFACT
  'INJ' 12.0 /
/
TSTEP
60*30 /
END
```

Латинские гиперкубы (50 точек)

Код генерации репрезентативного и нерепрезентативного латинских гиперкубов из 50 точек в двумерном пространстве:

```
%репрезентативный и нерепрезентативный латинские гиперкубы в двумерном
пространстве (m=50)
%в следующей строке использована функция ihs(dim_num, point_num,
duplication, seed) с сайта [10]. Но перед командой return в теле функции
ihs добавлена строка x = x./point_num; с целью нормализации гиперкуба к
диапазону [0;1]
lh_example1 = ihs(2, 50, 30, 5);
%нерепрезентативный латинский гиперкуб
lh_example2(1:2,1)=[0.02 0.02];
for i=2:50
    lh_example2(1:2,i)=lh_example2(1:2,i-1)+0.02;
end
%визуализация гиперкубов
text_size_scale = 1.9;
font = 'Times New Roman';
figure('Position', [10 10 750 750], 'DefaultTextFontName', font,
'DefaultAxesFontName', font)
scatter(lh_example1(1,:),lh_example1(2,:),27,'filled','black','DisplayName',
'Репрезентативный')
hold on
scatter(lh_example2(1,:),lh_example2(2,:),27,'filled','r','DisplayName',
'Нерепрезентативный')
grid on
grid minor
title('2 латинских гиперкуба в двумерном пространстве')
xlabel('Параметр 1')
ylabel('Параметр 2')
xlim([0 1.02])
ylim([0 1.095])
legend
%в следующей строке использована функция supersizeme(varargin) с сайта
[23] с целью увеличения шрифта текста
supersizeme(text_size_scale)

clear font i lh_example1 lh_example2 text_size_scale
```

ПРИЛОЖЕНИЕ 4

Латинские гиперкубы (для метамодели)

Код генерации латинских гиперкубов в пространстве варьируемых параметров:

```
%генерируется выборка методом латинского гиперкуба
%Варьируются: пористость матрицы, проницаемость матрицы, пористость трещин, проницаемость трещин

points_train = 400; %количество точек для обучения метамодели
N_train = 4; %количество варьируемых параметров
lb_train = [0.14 0.9 0.005 50]; %нижние границы диапазонов каждого из варьируемых параметров
ub_train = [0.24 10 0.05 2000]; %верхние границы диапазонов каждого из варьируемых параметров
%в следующей строке использована функция ihs(dim_num, point_num, duplication, seed) с сайта [10]. Однако перед командой return в теле функции ihs добавлена строка x = x./point_num; с целью нормализации гиперкуба к диапазону [0;1]
Xla_train = ihs(N_train, points_train, 5, 7)'; %гиперкуб, нормализованный к диапазону [0;1]
Dla_train = bsxfun(@plus,lb_train,bsxfun(@times,Xla_train,(ub_train-lb_train))); %гиперкуб в пространстве варьируемых параметров

points_valid = 100; %количество точек для валидации метамодели
N_valid = 4;
lb_valid = [0.14 0.9 0.005 50];
ub_valid = [0.24 10 0.05 2000];
Xla_valid = ihs(N_valid, points_valid, 5, 7)';
Dla_valid = bsxfun(@plus,lb_valid,bsxfun(@times,Xla_valid,(ub_valid-lb_valid)));

points_test = 800; %количество точек для тестирования метамодели
N_test = 4;
lb_test = [0.14 0.9 0.005 50];
ub_test = [0.24 10 0.05 2000];
Xla_test = ihs(N_test, points_test, 5, 7)';
Dla_test = bsxfun(@plus,lb_test,bsxfun(@times,Xla_test,(ub_test-lb_test)));

%объединение данных в таблицу, которая будет подаваться в качестве входных данных при обучении метамодели
input = [Dla_train' Dla_valid' Dla_test'];

%визуализация проекций латинских гиперкубов на трёхмерное пространство
text_size_scale = 1.9;
font = 'Times New Roman';
figure('DefaultTextFontName',font,'DefaultAxesFontName',font)
scatter3(Dla_train(:,1),Dla_train(:,2),Dla_train(:,4), 'b','DisplayName',
'Обучающая выборка')
```

```

hold on
scatter3(Dla_valid(:,1),Dla_valid(:,2),Dla_valid(:,4),'g','DisplayName',
'Валидационная выборка')
hold on
scatter3(Dla_test(:,1),Dla_test(:,2),Dla_test(:,4),'r','DisplayName',
'Тестовая выборка')
title('Проекции латинских гиперкубов на трёхмерное пространство')
xlabel('Пористость матрицы, в долях единицы')
ylabel('Проницаемость матрицы, мДарси')
zlabel('Проницаемость трещин, мДарси')
legend
%в следующей строке использована функция supersizeme(varargin) с сайта
[23] с целью увеличения шрифта текста
supersizeme(text_size_scale)

%очистка параметров, которые не будут использоваться в дальнейшем
clear points_test points_train points_valid N_test N_train N_valid font
text_size_scale

```

Генерация файлов для расчёта

Код генерации файлов, предназначенных для расчёта в гидродинамическом симуляторе:

```
%создаются файлы для расчёта в гидродинамическом симуляторе (этот код запускается 3 раза, при каждом запуске вручную меняются только значения sampling (= Dla_train, = Dla_valid или = Dla_test) и folder_number (= 1, = 2 или = 3)). После третьего запуска необходимо сохранить Workspace в текущую папку командой save('1'), чтобы в дальнейшем не потерять сгенерированные данные
sampling = Dla_train; %Dla_train – обучающая выборка, сгенерированная методом латинского гиперкуба на предыдущем этапе и находящаяся в Workspace
folder_number = 1; %номер–название папки, куда будут помещаться сгенерированные файлы

cases_count = length(sampling); %количество файлов для генерации

%генерация файла для командной строки. Файл необходим, чтобы все сгенерированные файлы запускались на расчёт в симулятор автоматически из командной строки
for i=1:cases_count
    cmd_matrix(i,1)={'C:\ecl\2017.1\bin\pc_x86_64\eclipse'};
    cmd_matrix{i,1}=cmd_matrix{i,1}+" "+string(i);
end

mkdir(string(folder_number)); %создание новой папки

writecell(cmd_matrix,string(folder_number)+'/'+string(folder_number), 'QuoteStrings',false);

fileList=dir([string(folder_number)+'/*.txt']);
file = fullfile(fileList(1).folder, fileList(1).name);
[tempDir, tempFile] = fileparts(file);
status = copyfile(file, fullfile(tempDir, [tempFile, '.cmd']));
delete(file)

%чтение файла–шаблона, хранящего в себе всю информацию (о гидродинамической модели), необходимую для расчёта на симуляторе
file_exp = textread('D_template1.DATA', '%s', 'delimiter', '\n', ...
    ' whitespace', '');

%далее в файле–шаблоне заменяются строки, содержащие информацию о значениях варьируемых параметров (подставляются значения из латинского гиперкуба). В цикле происходит сохранение файлов для последующего расчёта на симуляторе
for i=1:cases_count
```

```

%номера строк 59, 61, 63, 73, 75 и 77 соответствуют файлу-шаблону в
приложении 1
file_exp{59,1}="''PORO' "+string(sampling(i,1))+"/";
file_exp{61,1}="''PERMX' "+string(sampling(i,2))+"/";
file_exp{63,1}="''PERMY' "+string(sampling(i,2))+"/";
file_exp{73,1}="''PORO' "+string(sampling(i,3))+"/";
file_exp{75,1}="''PERMX' "+string(sampling(i,4))+"/";
file_exp{77,1}="''PERMY' "+string(sampling(i,4))+"/";

writecell(file_exp,string(folder_number)+"/D"+string(i)+".dat",'QuoteStr
ings',false);

end

%файлы сохранились в формате .dat. Меняем на .DATA
fileList=dir([string(folder_number)+'*.*dat']);

for i = 1:numel(fileList)
    file = fullfile(fileList(i).folder, fileList(i).name);
    [tempDir, tempFile] = fileparts(file);
    status = copyfile(file, fullfile(tempDir, [tempFile, '.DATA']));
    delete(file)
end

%очистка параметров, которые не будут использоваться в дальнейшем
clear cases_count cmd_matrix file file_exp fileList folder_number
clear i sampling status tempDir tempFile

```

Считывание результатов расчёта

Код считывания результатов из файлов, сгенерированных симулятором. После считывания необходимо сохранить Workspace в текущую папку командой `save('1')`, чтобы в дальнейшем не потерять данные. Файлы имеют формат RSM (содержат информацию о дебитах нефти и т.д.) и UNRST (содержат информацию о насыщенностиях и т.д.):

```

folder_number = 1; %номер папки с результатами симулятора на обучающей
%выборке

for i=1:length(Dla_train)
    %в следующей строке использована функция readRunFar(file) с сайта
    [21]. Код этой функции был дополнен для корректного чтения всех RSM
    файлов и приведён в следующем приложении 7.

    res_train{i}=readRunFar(string(folder_number)+"/D"+string(i)+".RSM");
    %результаты из .RSM файлов в виде массива таблиц
    FOPR_train(:,i)=res_train{i}.FOPR; %таблица дебитов нефти; каждый
    %столбец – это значения дебитов по месяцам для кейса, соответствующего
    %номеру столбца
end

folder_number = 2; %номер папки с результатами симулятора на
%валидационной выборке

for i=1:length(Dla_valid)
    res_valid{i}=readRunFar(string(folder_number)+"/D"+string(i)+".RSM");
    FOPR_valid(:,i)=res_valid{i}.FOPR;
end

folder_number = 3; %номер папки с результатами симулятора на тестовой
%выборке

for i=1:length(Dla_test)
    res_test{i}=readRunFar(string(folder_number)+"/D"+string(i)+".RSM");
    FOPR_test(:,i)=res_test{i}.FOPR;
end

%объединение данных в таблицу, которая будет подаваться в качестве
%целевых данных при обучении метамодели
FOPR_target = [FOPR_train FOPR_valid FOPR_test];

%очистка параметров, которые не будут использоваться в дальнейшем
clear folder_number i

```

```

folder_number = 1; %номер папки с результатами симулятора на обучающей
 выборке

%далее приведён цикл, в котором считываются выходные данные из UNRST
файлов. Была проведена визуализация данных, но в этой работе данные из
UNRST файлов не анализируются и нигде не используются, так как не
выяснены причины нестабильной работы функции read_ecl(file). Для
некоторых кейсов функция не считывает UNRST файлы (выдаёт ошибку).
for i=1:length(Dla_train)
    %в следующей строке использована функция read_ecl(file) с сайта [22]
    out{i}=read_ecl(string(folder_number)+'/D'+string(i)+'.UNRST');
    SWAT_train(:,i)=out11{i}.SWAT(:,8); %таблица значений
водонасыщенности (состояние на восьмой месяц) по ячейкам
гидродинамической модели пласта
end

%очистка параметров, которые не будут использоваться в дальнейшем
clear folder_number i

```

Код функции readRunFar(file)

Дополненный код функции readRunFar(file) с сайта [21]:

```

function t=readRunFar(file)
% return table containing data from input file
fid=fopen(file,'r');
% read the fullfile as cell array, convert to string array, ignore the
FORTRAN formfeed

c=textscan(fid,'%s','delimiter','\n','Whitespace','','','CommentStyle','1',
'CollectOutput',0);
fid=fclose(fid);
c=string(c{:});
ix=find(contains(c,"SUMMARY"))-1; % get section starting lines
N=numel(ix); % the number of sections in
file % the start of second column
iyr=find(contains(c,"YEARS")); % of data
iyr=strfind(c(iyr(1)),"YEARS")-1;
c(ix(2):end)=extractAfter(c(ix(2):end),iyr); % remove the TIME column
after first set
iend=find(char(c(end))~=' ',1,'last'); % find last data on last
section
d=c(1:ix(2)-1); % begin a new data array with
first section % join subsequent sections to
for i=2:N-1
first
d=strcat(d, c(ix(i):ix(i+1)-1));
end
temp = extractBefore(c(ix(N):end),iend+1);
if length(temp)>length(d)
temp(7)=[];
end
d=strcat(d, temp); % trash the beginning header
lines
d=d(~all(char(d)==' '|char(d)=='_',2));% and the other extraneous
text
data=str2num(char(d(4:end))); % convert to numeric
vnames=split(d(1)); % get the variable names from
header
vnames=vnames(~(vnames=="")); % split() leaves empties...
vnames=categorical(vnames,unique(vnames,'stable'));% create ordered
categorical variable
cats=categories(vnames); % the category names for
logical addressing
t=table(data(:,1),'VariableNames',cats(1)); % initial table entry
for i=2:numel(cats) % and build the output table
t=[t table(data(:,vnames==cats(i)), 'VariableNames',cats(i))];
end
end

```

Код построения нейронной сети

Код построения и обучения двухслойного перцептрана:

```

x = input; %входной вектор нейронной сети
t = FOPR_target; %вектор целевых значений на выходе нейронной сети

net = network; %инициализация нейронной сети
net.name = 'Метамодель для предсказания дебитов нефти (FOPR)';

net.numInputs = 1; %количество входных параметров нейронной сети (один
вектор)

net.numLayers = 2; %количество слоёв нейронной сети

net.biasConnect = [1; 1]; %связи векторов отклонений
net.inputConnect = [1; 0]; %связи входного слоя
net.layerConnect = [0 0; 1 0]; %связи между слоями
net.outputConnect = [0 1]; %связи выходного слоя

net.inputs{1}.name='Вход'; %название входа
net.inputs{1}.exampleInput=input(:,1); %пример структуры входного
вектора
net.inputs{1}.range=[lb_train;ub_train]'; %диапазоны варьирования
каждого значения во входном векторе
net.inputs{1}.processFcns = {'removeconstantrows','mapminmax'};
%применяемые методы предобработки входного вектора

net.layers{1}.name='Скрытый слой'; %название первого слоя нейронной сети
net.layers{1}.size=30; %количество нейронов на первом слое
net.layers{1}.transferFcn='tansig'; %функция активации первого слоя
net.layers{1}.initFcn='initnw'; %функция инициализации значений (перед
обучением) на первом слое

net.layers{2}.name='Выходной слой'; %название второго слоя нейронной
сети
net.layers{2}.size=60; %количество нейронов на втором слое
net.layers{2}.transferFcn='tansig'; %функция активации второго слоя
net.layers{2}.initFcn='initnw'; %функция инициализации значений (перед
обучением) на втором слое

net.outputs{2}.name='Выход'; %название выхода
net.outputs{2}.processFcns={'removeconstantrows','mapminmax'};
%применяемые методы предобработки выходного вектора

net.adaptFcn = 'adaptwb';
net.derivFcn = 'defaultderiv';

%Способ разделения получаемых на обучение данных на выборки (обучающую,
валидационную и тестовую)
net.divideFcn = 'divideind';

```

```

net.divideMode = 'sample';
net.divideParam.trainInd = 1:length(Dla_train(:,1));
net.divideParam.valInd = length(Dla_train(:,1))...
    +1:length(Dla_train(:,1))+length(Dla_valid(:,1));
net.divideParam.testInd =
length(Dla_train(:,1))+length(Dla_valid(:,1))+...
1:length(Dla_train(:,1))+length(Dla_valid(:,1))+length(Dla_test(:,1));

net.initFcn = 'initlay';
net.performFcn = 'mse'; %функция потерь
net.performParam.regularization = 0; %коэффициент регуляризации
net.performParam.normalization = 'none'; %нормализация ошибок
%относительно целевых данных
net.plotFcns = {'plotperform', 'plottrainstate', 'ploterrhist',...
    'plotregression', 'plotfit'}; %графики, генерируемые в процессе
обучения
net.trainFcn='trainscg'; %метод обучения нейронной сети (метод
масштабируемых сопряжённых градиентов)
net.trainParam.epochs=50000; %максимальное количество эпох
net.trainParam.max_fail=1000; %максимальное количество неудач улучшения
прогноза на валидационной выборке

% Обучение нейронной сети
[net,tr] = train(net,x,t);

% Анализ точности построенной нейронной сети
y = net(x); %прогноз
e = gsubtract(t,y); %разность целевых и предсказанных значений
performance = perform(net,t,y) %mse

% Перерасчёт ошибки прогноза (mse) для обучающей, валидационной и
тестовой выборок
trainTargets = t .* tr.trainMask{1};
valTargets = t .* tr.valMask{1};
testTargets = t .* tr.testMask{1};
trainPerformance = perform(net,trainTargets,y)
valPerformance = perform(net,valTargets,y)
testPerformance = perform(net,testTargets,y)

%Относительная ошибка прогноза
e_rel=e./t*100;

%Гистограмма относительных ошибок для всех предсказанных значений (на
обучающей+валидационной+тестовой выборках)
figure
ploterrhist(e_rel);

%Диаграмма размаха для всех предсказанных значений (на
обучающей+валидационной+тестовой выборках)
figure
boxplot(e_rel(1:length(t(:,1)),1:length(t(1,:))))'

%Сохранение всех данных о построенной нейронной сети в структуру в
Workspace
network_name = 'temp1'; %название файла с данными (временный файл)

```

```

save(network_name, '-regexp', '^(!struct_.*$).'); %сохранение файла (всё
из Workspace, кроме struct_)
struct_net1_trainscg1 = load(network_name); %загрузка файла в Workspace
в виде структуры. При очередной попытке обучения метамодели название
переменной необходимо изменить для её отдельного сохранения, а не
перезаписи имеющейся
delete temp1.mat; %удаление файла
clearvars -except struct* %удаление всех переменных из Workspace, кроме
struct_
load('1.mat') %загрузка исходных данных для дальнейших попыток обучения
метамодели

%Просмотр архитектуры нейронной сети
% struct_vis = struct_net1_trainscg1
% view(struct_vis.net)

%Просмотр графиков, генерируемых в процессе обучения
%figure, plotperform(struct_vis.tr)
%figure, plottrainstate(struct_vis.tr)
%figure, ploterrhist(struct_vis.e)
%figure, plotregression(struct_vis.t,struct_vis.y)
%figure, plotfit(struct_vis.net,struct_vis.x,struct_vis.t)

```

Визуализация

Код визуализации истории процесса обучения (функции потерь) нейронной сети и размаха ошибок прогноза метамодели:

```
%Визуализация истории процесса обучения (функции потерь) и размаха ошибок прогноза нейронной сети

%в следующей строке необходимо выбрать структуру с информацией о
%построенной нейросети, для которой проводится визуализация
struct_net=struct_net1_trainscg1;

text_size_scale = 1.7;
font = 'Times New Roman';
figure('Position', [10 10 600
600], 'DefaultTextFontName', font, 'DefaultAxesFontName', font)
plotperform(struct_net.tr)
text(struct_net.tr.num_epochs/3,struct_net.tr.perf(1,1)/10,{['P =
'+string(length(struct_net.Dla_train)), 'N^1 =
'+string(struct_net.net.layers{1}.size)}, 'FontName', font, 'FontSize', 11)
title({'Наименьшее значение на валидационной выборке составляет
', string(struct_net.valPerformance)+ ' и достигается на
'+string(struct_net.tr.best_epoch)+ ' эпохе'})
xlabel('Номер эпохи (всего
'+string(struct_net.tr.num_epochs)+ ')', 'FontSize', 12, 'FontWeight', 'bold')
)
ylabel('Среднеквадратичная ошибка
(mse)', 'FontSize', 12, 'FontWeight', 'bold')
legend('Обучающая выборка', 'Валидационная выборка', 'Тестовая
выборка', 'Лучшее значение при валидации')
grid on
grid minor
supersizeme(text_size_scale)

errors = -struct_net.e_rel;
%errors = -struct_net.e_rel(:,input(4,:)>400 & input(3,:)>0.01);
figure('Position', [10 10 1200
700], 'DefaultTextFontName', font, 'DefaultAxesFontName', font)
b=boxchart(errors(2:length(errors(:,1)),1:length(errors(1,:))))';
b.MarkerStyle='*';
b.MarkerColor='#ABADAE';
b.BoxFaceColor = 'black';
text(length(struct_net.FOPR_test(:,1))/2,round(max(max(errors)), -1),{['P =
'+string(length(struct_net.Dla_train)), 'N^1 =
'+string(struct_net.net.layers{1}.size)}, 'FontName', font, 'FontSize', 12)
title('Диаграмма размаха ошибок прогноза метамодели', 'FontSize', 15)
xlabel('Номер месяца', 'FontWeight', 'bold', 'FontSize', 14)
ylabel('Ошибка, в процентах', 'FontWeight', 'bold', 'FontSize', 14)
grid on
```

```
grid minor
ylim([round(min(min(errors)), -1)-10 round(max(max(errors)), -1)+50])
supersizeme(text_size_scale)

clear struct_net text_size_scale font errors b
```

Сравнение симулятор/метамодель

Код для построения графиков дебита нефти с целью сравнения картин, получаемых для симулятора и метамодели:

```
%Визуализация дебитов нефти для всех кейсов (сравнение: симулятор и
метамодель)

%в следующей строке необходимо выбрать структуру с информацией о
построенной нейросети, для которой проводится визуализация
struct_net=struct_net1_trainscg1;

text_size_scale = 2;
font = 'Times New Roman';
figure('Position', [10 10 1100
700], 'DefaultTextFontName', font, 'DefaultAxesFontName', font)
net1 = struct_net.net;
subplot(1,2,1)
hold on
cases_count=length(struct_net.FOPR_target(1,:));
plot(struct_net.FOPR_target(1:end,1:cases_count));
text(2*length(struct_net.FOPR_target(:,1))/3,4*max(max(struct_net.FOPR_t
arget))/5, 'Сценарий 1', 'FontName', font, 'FontSize', 11)
title({'Дебит нефти от времени', 'для всех кейсов (симулятор)'})
xlabel('Номер месяца')
ylabel('Дебит нефти, кубометров в сутки')
grid on
grid minor
subplot(1,2,2)
hold on
plot(net1(struct_net.input(1:end,1:cases_count)))
text(2*length(struct_net.FOPR_target(:,1))/3,4*max(max(struct_net.FOPR_t
arget))/5, {'P = '+string(length(struct_net.Dla_train)), 'N^1 =
'+string(struct_net.net.layers{1}.size)}, 'FontName', font, 'FontSize', 11)
title({'Дебит нефти от времени', 'для всех кейсов (метамодель)'})
xlabel('Номер месяца')
ylabel('Дебит нефти, кубометров в сутки')
grid on
grid minor
supersizeme(text_size_scale)

clear text_size_scale font net1 cases_count struct_net
```

Визуализация прогноза метамодели

Код для визуализации зависимостей дебита нефти от времени при варьировании одного из входных параметров метамодели:

```
%Визуализация зависимостей дебита нефти при варьировании одного из
входных параметров
```

```

tic

text_size_scale = 2;
font = 'Times New Roman';

%в следующих двух строках необходимо выбрать структуры с информацией о
нейросетях, для которых будет проводиться визуализация
struct1 = struct_net1_trainscg1;
struct2 = struct_net1_trainscg2;

%нейросети из выбранных структур
net1 = struct1.net; %первая нейросеть, для которой проводится
визуализация
net2 = struct2.net; %вторая нейросеть, для которой проводится
визуализация

%выбранные значения входных параметров, если они не варьируются
poro_matrix = 0.2;
perm_matrix = 5;
poro_crack = 0.02;
perm_crack = 1500;

%варьируется пористость матрицы
figure('Position', [10 10 1300
850], 'DefaultTextFontName', font, 'DefaultAxesFontName', font)
subplot(1,2,1)
hold on
slice_array=0.14:0.005:0.24;
color_array=jet(length(slice_array))/1.5;
flag=0;
for i=slice_array
    flag=flag+1;
    new_result = net1([i;perm_matrix;poro_crack;perm_crack]);
    plot(0:length(new_result)-
1,new_result,'DisplayName',string(i),'Color',color_array(flag,:));
end
text(2*length(struct1.FOPR_target(:,1))/3,4*max(new_result)/5,{['P =
'+string(length(struct1.Dla_train)), 'N^1 =
'+string(struct1.net.layers{1}.size)}, 'FontName', font, 'FontSize', 11)
title(compose('Графики зависимости дебитов от времени при \
варьировании пористости матрицы'))
xlabel('Номер месяца')
```

```

ylabel('Дебит нефти, кубометров за сутки')
xlim([0,60])
grid on
grid minor
leg = legend('show');
leg.FontSize = 7;
title(leg,compose('Пористость \n матрицы'))

subplot(1,2,2)
hold on
slice_array=0.14:0.005:0.24;
color_array=jet(length(slice_array))/1.5;
flag=0;
for i=slice_array
    flag=flag+1;
    new_result = net2([i;perm_matrix;poro_crack;perm_crack]);
    plot(0:length(new_result)-
1,new_result,'DisplayName',string(i),'Color',color_array(flag,:));
end
text(2*length(struct2.FOPR_target(:,1))/3,4*max(new_result)/5,{['P =
'+string(length(struct2.Dla_train)), 'N^1 =
'+string(struct2.net.layers{1}.size)}, 'FontName', font, 'FontSize', 11)
title(compose('Графики зависимости дебитов от времени при \n
варьировании пористости матрицы'))
xlabel('Номер месяца')
ylabel('Дебит нефти, кубометров за сутки')
xlim([0,60])
grid on
grid minor
leg = legend('show');
leg.FontSize = 7;
title(leg,compose('Пористость \n матрицы'))
supersizeme(text_size_scale)

%варьируется проницаемость матрицы
figure('Position', [10 10 1300
850], 'DefaultTextFontName', font, 'DefaultAxesFontName', font)
subplot(1,2,1)
hold on
slice_array=0.9:0.455:10;
color_array=jet(length(slice_array))/1.5;
flag=0;
for i=slice_array
    flag=flag+1;
    new_result = net1([poro_matrix;i;poro_crack;perm_crack]);
    plot(0:length(new_result)-1,new_result,'DisplayName',string(i)+'
мД','Color',color_array(flag,:));
end
text(2*length(struct1.FOPR_target(:,1))/3,4*max(new_result)/5,{['P =
'+string(length(struct1.Dla_train)), 'N^1 =
'+string(struct1.net.layers{1}.size)}, 'FontName', font, 'FontSize', 11)
title(compose('Графики зависимости дебитов от времени при \n
варьировании проницаемости матрицы'))
xlabel('Номер месяца')
ylabel('Дебит нефти, кубометров за сутки')
xlim([0,60])

```

```

grid on
grid minor
leg = legend('show');
leg.FontSize = 7;
title(leg,compose('Проницаемость \n матрицы'))

subplot(1,2,2)
hold on
slice_array=0.9:0.455:10;
color_array=jet(length(slice_array))/1.5;
flag=0;
for i=slice_array
    flag=flag+1;
    new_result = net2([poro_matrix;i;poro_crack;perm_crack]);
    plot(0:length(new_result)-1,new_result,'DisplayName',string(i)+'
мД','Color',color_array(flag,:));
end
text(2*length(struct2.FOPR_target(:,1))/3,4*max(new_result)/5,{ 'P =
'+string(length(struct2.Dla_train)), 'N^1 =
'+string(struct2.net.layers{1}.size)}, 'FontName', font, 'FontSize', 11)
title(compose('Графики зависимости дебитов от времени при \n
варьировании проницаемости матрицы'))
xlabel('Номер месяца')
ylabel('Дебит нефти, кубометров за сутки')
xlim([0,60])
grid on
grid minor
leg = legend('show');
leg.FontSize = 7;
title(leg,compose('Проницаемость \n матрицы'))
supersizeme(text_size_scale)

%варьируется пористость трещин
figure('Position', [10 10 1300
850], 'DefaultTextFontName', font, 'DefaultAxesFontName', font)
subplot(1,2,1)
hold on
slice_array=0.005:0.0025:0.05;
color_array=jet(length(slice_array))/1.5;
flag=0;
for i=slice_array
    flag=flag+1;
    new_result = net1([poro_matrix;perm_matrix;i;perm_crack]);
    plot(0:length(new_result)-
1,new_result,'DisplayName',string(i), 'Color',color_array(flag,:));
end
text(2*length(struct1.FOPR_target(:,1))/3,4*max(new_result)/5,{ 'P =
'+string(length(struct1.Dla_train)), 'N^1 =
'+string(struct1.net.layers{1}.size)}, 'FontName', font, 'FontSize', 11)
title(compose('Графики зависимости дебитов от времени при \n
варьировании пористости трещин'))
xlabel('Номер месяца')
ylabel('Дебит нефти, кубометров за сутки')
xlim([0,60])
grid on
grid minor

```

```

leg = legend('show');
leg.FontSize = 7;
title(leg,compose('Пористость \n трещин'))

subplot(1,2,2)
hold on
slice_array=0.005:0.0025:0.05;
color_array=jet(length(slice_array))/1.5;
flag=0;
for i=slice_array
    flag=flag+1;
    new_result = net2([poro_matrix;perm_matrix;i;perm_crack]);
    plot(0:length(new_result)-
1,new_result,'DisplayName',string(i),'Color',color_array(flag,:));
end
text(2*length(struct2.FOPR_target(:,1))/3,4*max(new_result)/5,{['P =
'+string(length(struct2.Dla_train)), 'N^1 =
'+string(struct2.net.layers{1}.size)},'FontName',font,'FontSize',11)
title(compose('Графики зависимости дебитов от времени при \n
варьировании пористости трещин'))
xlabel('Номер месяца')
ylabel('Дебит нефти, кубометров за сутки')
xlim([0,60])
grid on
grid minor
leg = legend('show');
leg.FontSize = 7;
title(leg,compose('Пористость \n трещин'))
supersizeme(text_size_scale)

%варьируется проницаемость трещин
figure('Position', [10 10 1300
850], 'DefaultTextFontName',font, 'DefaultAxesFontName',font)
subplot(1,2,1)
hold on
slice_array=50:65:2000;
color_array=jet(length(slice_array))/1.5;
flag=0;
for i=slice_array
    flag=flag+1;
    new_result = net1([poro_matrix;perm_matrix;poro_crack;i]);
    plot(0:length(new_result)-1,new_result,'DisplayName',string(i)+'
мД','Color',color_array(flag,:));
end
text(2*length(struct1.FOPR_target(:,1))/3,4*max(new_result)/5,{['P =
'+string(length(struct1.Dla_train)), 'N^1 =
'+string(struct1.net.layers{1}.size)},'FontName',font,'FontSize',11)
title(compose('Графики зависимости дебитов от времени при \n
варьировании проницаемости трещин'))
xlabel('Номер месяца')
ylabel('Дебит нефти, кубометров за сутки')
xlim([0,60])
grid on
grid minor
leg = legend('show');
leg.FontSize = 6;

```

```

title(leg,compose('Проницаемость \n трещин'))

subplot(1,2,2)
hold on
slice_array=50:65:2000;
color_array=jet(length(slice_array))/1.5;
flag=0;
for i=slice_array
    flag=flag+1;
    new_result = net2([poro_matrix;perm_matrix;poro_crack;i]);
    plot(0:length(new_result)-1,new_result,'DisplayName',string(i)+'
мД','Color',color_array(flag,:));
end
text(2*length(struct2.FOPR_target(:,1))/3,4*max(new_result)/5,{['P =
'+string(length(struct2.Dla_train)), 'N^1 =
'+string(struct2.net.layers{1}.size)},'FontName',font,'FontSize',11)
title(compose('Графики зависимости дебитов от времени при \
варьировании проницаемости трещин'))
xlabel('Номер месяца')
ylabel('Дебит нефти, кубометров за сутки')
xlim([0,60])
grid on
grid minor
leg = legend('show');
leg.FontSize = 6;
title(leg,compose('Проницаемость \n трещин'))
supersizeme(text_size_scale)

toc

clear slice_array color_array flag new_result leg i text_size_scale
clear font struct1 struct2 net1 net2 poro_matrix perm_matrix
clear poro_crack perm_crack

```

Прогноз метамодели в движении

Код для визуализации зависимости дебита нефти от времени и входных параметров нейросети (в движении):

```
%Визуализация зависимости дебита нефти от времени и входных параметров
нейросети (в движении)
```

```
%в следующей строке необходимо выбрать структуру с информацией о
построенной нейросети, для которой проводится визуализация
struct_vis = struct_net1_trainscg1;

%нейросеть, для которой проводится визуализация
net_vis = struct_vis.net;

figure('Position', [10 10 1250 700])
for j=50:10:2000
clf
subplot(2,2,1)
hold on
flag=0;
str={};
slice_array=1:1:10;
color_array=jet(length(slice_array))/1.5;
for i=slice_array
new_result = net_vis([0.14;i;0.005;j]);
plot(new_result,'DisplayName',string(i),'Color',color_array(flag+1,:));
flag=flag+1;
str{flag}=string(i)+ ' мД';
end
title(compose('Пористость матрицы: 0.14; проницаемость матрицы указана в
легенде;\n пористость трещин: 0.005; проницаемость трещин: '+string(j)+'
мД'))
xlabel('Номер месяца')
ylabel('Дебит нефти, кубометров за сутки')
grid on
grid minor
leg=legend(str);
title(leg,{['P = ' +string(length(struct_vis.Dla_train)), 'N^1 = '
+string(struct_vis.net.layers{1}.size)]})
subplot(2,2,2)
hold on
flag=0;
str={};
for i=slice_array
new_result = net_vis([0.24;i;0.005;j]);
plot(new_result,'DisplayName',string(i),'Color',color_array(flag+1,:));
flag=flag+1;
str{flag}=string(i)+ ' мД';
end
```

```

title(compose('Пористость матрицы: 0.24; проницаемость матрицы указана в
легенде;\n пористость трещин: 0.005; проницаемость трещин: '+string(j)+'
мД'))
xlabel('Номер месяца')
ylabel('Дебит нефти, кубометров за сутки')
grid on
grid minor
leg=legend(str);
title(leg,{ 'P = '+string(length(struct_vis.Dla_train)), 'N^1 =
'+string(struct_vis.net.layers{1}.size)})
subplot(2,2,3)
hold on
flag=0;
str={};
for i=slice_array
new_result = net_vis([0.14;i;0.05;j]);
plot(new_result,'DisplayName',string(i),'Color',color_array(flag+1,:));
flag=flag+1;
str{flag}=string(i)+' мД';
end
title(compose('Пористость матрицы: 0.14; проницаемость матрицы указана в
легенде;\n пористость трещин: 0.05; проницаемость трещин: '+string(j)+'
мД'))
xlabel('Номер месяца')
ylabel('Дебит нефти, кубометров за сутки')
grid on
grid minor
leg=legend(str);
title(leg,{ 'P = '+string(length(struct_vis.Dla_train)), 'N^1 =
'+string(struct_vis.net.layers{1}.size)})
subplot(2,2,4)
hold on
flag=0;
str={};
for i=slice_array
new_result = net_vis([0.24;i;0.05;j]);
plot(new_result,'DisplayName',string(i),'Color',color_array(flag+1,:));
flag=flag+1;
str{flag}=string(i)+' мД';
end
title(compose('Пористость матрицы: 0.24; проницаемость матрицы указана в
легенде;\n пористость трещин: 0.05; проницаемость трещин: '+string(j)+'
мД'))
xlabel('Номер месяца')
ylabel('Дебит нефти, кубометров за сутки')
grid on
grid minor
leg=legend(str);
title(leg,{ 'P = '+string(length(struct_vis.Dla_train)), 'N^1 =
'+string(struct_vis.net.layers{1}.size)})
supersizeme(1.3)
drawnow
end

clear struct_vis net_vis j flag str slice_array color_array i new_result
clear leg

```

Оценка неопределённости

Код для оценки неопределённости суммы дебитов нефти методом Монте-Карло:

```

zMonte_n=1e+6;
net_Monte=struct_net1_trainscg1.net;

%zMonte_poro_matrix = ((randn(zMonte_n,1)*0.01)+0.2)';
zMonte_poro_matrix_distr =
makedist('Triangular','a',0.17,'b',0.2,'c',0.2);
zMonte_poro_matrix = random(zMonte_poro_matrix_distr,zMonte_n,1)';

%zMonte_perm_matrix = ((randn(zMonte_n,1)*1)+6)';
zMonte_perm_matrix_distr = makedist('Normal','mu',6,'sigma',0.5);
zMonte_perm_matrix = random(zMonte_perm_matrix_distr,zMonte_n,1)';

%zMonte_poro_crack = ((randn(zMonte_n,1)*0.004)+0.03)';
zMonte_poro_crack_distr = makedist('Uniform','lower',0.03,'upper',0.04);
zMonte_poro_crack = random(zMonte_poro_crack_distr,zMonte_n,1)';

%zMonte_perm_crack = ((randn(zMonte_n,1)*40)+1000)';
zMonte_perm_crack_distr =
makedist('Triangular','a',1200,'b',1200,'c',1400);
zMonte_perm_crack = random(zMonte_perm_crack_distr,zMonte_n,1);

text_size_scale = 1.7;
font = 'Times New Roman';
figure('Position', [10 10 1000
800], 'DefaultTextFontName', font, 'DefaultAxesFontName', font)
subplot(2,2,1)
histogram(zMonte_poro_matrix,30);
title(compose('Распределение значений \n пористости матрицы'))
ylabel({'Количество значений', 'в диапазонах'})
xlabel('Диапазоны')
grid on
grid minor
subplot(2,2,2)
histogram(zMonte_perm_matrix,30);
title(compose('Распределение значений \n проницаемости матрицы'))
ylabel({'Количество значений', 'в диапазонах'})
xlabel('Диапазоны, мД')
grid on
grid minor
subplot(2,2,3)
histogram(zMonte_poro_crack,30);
title(compose('Распределение значений \n пористости трещин'))
ylabel({'Количество значений', 'в диапазонах'})
xlabel('Диапазоны')
grid on

```

```

grid minor
subplot(2,2,4)
histogram(zMonte_perm_crack,30);
title(compose('Распределение значений \n проницаемости трещин'))
ylabel({'Количество значений','в диапазонах'})
xlabel('Диапазоны, мД')
grid on
grid minor
supersizeme(text_size_scale)

text_size_scale = 2;
tic
zMonte_result =
sum(net_Monte([zMonte_poro_matrix;zMonte_perm_matrix;zMonte_poro_crack;z
Monte_perm_crack]));
toc
figure('Position', [10 10 800
800], 'DefaultTextFontName', font, 'DefaultAxesFontName', font)
histogram(zMonte_result,30);
xlim([1.15 1.5]*1E+4)
ylim([0 8.5]*1E+4)
title('Гистограмма сумм дебитов')
ylabel('Количество значений в диапазонах')
xlabel('Диапазоны сумм дебитов, кубометров в сутки')
grid on
grid minor
supersizeme(text_size_scale)

```

XGBoost модель

Код построения метамодели с помощью Python XGBoost. Метамодель строится на основе данных, считанных в MATLAB из RSM файлов, сгенерированных симулятором ECLIPSE. Данные из MATLAB выгружаются в виде таблицы командой `writematrix([input;F0PR_target], 'cases1.csv');`. Далее сгенерированная таблица считывается средствами Python в JupyterLab:

[1]:

```
import numpy as np
import pandas as pd
import xgboost as xg
from sklearn.metrics import mean_squared_error as MSE
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
import time
import pylab as pl
from IPython import display
```

[2]:

```
# Считывание данных из csv файла
data = pd.read_csv("cases3.csv",header=None)

# Количество месяцев расчёта на симуляторе
month_count = 61

# Необходимо указать количество кейсов для обучения, валидации и
тестирования
train_count = 400
```

```
valid_count = 100
test_count = 800

# Разделение считанных данных на обучающую, валидационную и тестовую
# выборки
data_train = data.iloc[:,0:train_count]
data_valid = data.iloc[:,train_count:train_count+valid_count]
data_test = data.iloc[:,train_count+valid_count:train_count
                     +valid_count+test_count]
```

[3]:

```
# Вывод таблицы с данными в JupyterLab
```

```
# в таблице: столбцы - кейсы расчитанные на симуляторе (каждый
# со своими значениями входных параметров)
```

```
# строка 0 - пористость матрицы (входной параметр)
```

```
# строка 1 - проницаемость матрицы (входной параметр)
```

```
# строка 2 - пористость трещин (входной параметр)
```

```
# строка 3 - проницаемость трещин (входной параметр)
```

```
# строки 4-64 - дебиты нефти по месяцам 0-60 (выходные параметры)
```

```
data
```

[4]:

```
# Массив для построенных XGBoost метамоделей (для каждого месяца своя
# модель)
XGB_models = []

# В цикле для каждого месяца строится своя XGBoost метамодель
for month_current in range(1,month_count):

    # Разделение входных и целевых данных
    X_train,T_train = data_train.iloc[:-month_count,:].transpose(), \
                      data_train.iloc[month_current-month_count,:]
    X_valid,T_valid = data_valid.iloc[:-month_count,:].transpose(), \
                      data_valid.iloc[month_current-month_count,:]
    X_test,T_test = data_test.iloc[:-month_count,:].transpose(), \
                     data_test.iloc[month_current-month_count,:]

    xgb_r = xg.XGBRegressor(learning_rate=0.05,n_estimators=100,
                           max_depth=100,gamma=0)
    #xgb_r = xg.XGBRegressor(learning_rate=0.009,n_estimators=1000,
    #                       min_child_weight=0.03,max_depth=10)
    #xgb_r = xg.XGBRegressor()

    # Наборы, на которых будут проводиться обучение, валидация и тестирование
    eval_set = [(X_train,T_train),(X_valid,T_valid),(X_test,T_test)]

    # Обучение
    xgb_r.fit(X_train,T_train,eval_metric=["rmse"],
               eval_set=eval_set,verbose=False)
    XGB_models.append(xgb_r)
```

```

# Вывод истории процесса обучения (значений функции потерь)
if month_current == 7:
    results=xgb_r.evals_result()
    epochs=len(results['validation_0']['rmse'])
    x_axis=range(0,epochs)
    fig, ax = plt.subplots()
    plt.rcParams['figure.figsize']=[7,7]
    ax.plot(x_axis,results['validation_0']['rmse'],label='Train')
    ax.plot(x_axis,results['validation_1']['rmse'],label='Valid')
    ax.plot(x_axis,results['validation_2']['rmse'],label='Test')
    ax.legend(fontsize=14)
    plt.xlabel('Epoch',fontsize=16)
    plt.xticks(fontsize=14)
    plt.ylabel('RMSE',fontsize=16)
    plt.yticks(fontsize=14)
    plt.title('XGBoost Training',fontsize=18)
    plt.grid(which='both')
    plt.show()

```

[5]:

```

# Прогноз
months_for_forecast = 11 #количество месяцев для прогноза
rel_errors = {}
for month_current in range(1,months_for_forecast):
    X,T = data.iloc[:-month_count,:].transpose(), \
           data.iloc[month_current-month_count,:]
    Y = XGB_models[month_current-1].predict(X)
    abs_err = list(np.subtract(Y,T))
    rel_errors[str(month_current)] = np.divide(abs_err,T)*100

```

```

plt.figure(figsize=(10, 7), dpi=100)
plt.boxplot(rel_errors.values())
plt.grid(which='both')
plt.title('Диаграмма размаха ошибок прогноза метамодели')
plt.xlabel('Номер месяца')
plt.ylabel('Ошибка, в процентах')

#print(XGB_models[month_current-1].feature_importances_)
#rel_err #ошибка по кейсам на валидационной выборке в процентах

```

[6]:

```

# Визуализация степени влияния входных параметров на результат расчёта (для
# каждого месяца)

```

```
%matplotlib inline
```

```

bars = ('Пористость \n матрицы','Проницаемость \n матрицы',
        'Пористость \n трещин','Проницаемость \n трещин')
x_pos = np.arange(len(bars))
for i in range(1,month_count):
    plt.clf()
    plt.figure(figsize=(8, 6), dpi=100)
    plt.ylim((0,1))
    plt.ylabel('Доли единицы', fontsize=12)
    plt.grid(axis='y')
    plt.title('Степень влияния входных параметров на результат прогноза '
              + '\n Номер месяца: ' + str(i))
    plt.bar(range(len(XGB_models[i-1].feature_importances_)),
            XGB_models[i-1].feature_importances_, label='d')
    plt.xticks(x_pos,bars)

```

```
display.clear_output(wait=True)
display.display(pl.gcf())
#if i==1:
#    time.sleep(10)
time.sleep(0.3)
```

[7]:

```
# Построение графика зависимости дебитов от времени для произвольных
# значений входных параметров внутри диапазонов обучения
new_case=pd.DataFrame(list(zip([0.18],[5],[0.03],[1000])),
                      columns=['0','1','2','3'])
FOPR_y=[0]
for i in range(0,60):
    FOPR_y.append(XGB_models[i].predict(new_case)) #предсказание для нового
    #кейса
plt.plot(FOPR_y)
plt.grid()
plt.xlabel('Номер месяца')
plt.ylabel('Дебит, кубометров в сутки')
plt.title("Зависимость дебита нефти от времени \n (расчёт на метамодели)")
```

ПРИЛОЖЕНИЕ 15

Построенные нейронные сети

Ссылка на файлы с расширением mat, которые содержат данные, полученные при расчёте на симуляторе и используемые при обучении метамодели, а также структуры, содержащие всю информацию о построенных нейронных сетях:

<https://drive.google.com/drive/folders/18JnnKxQAZX31Vny2sOqWMrsC-7ka7zAd>

1_net_P=400_Scenario_1.mat: первый сценарий разработки и $P = 400$ точек на обучающей выборке.

2_net_P=800_Scenario_1.mat: первый сценарий разработки и $P = 800$ точек на обучающей выборке.

3_net_P=800_Scenario_2.mat: второй сценарий разработки и $P = 800$ точек на обучающей выборке.

Таблица приложения 15 (о данных в mat файлах)

lb_train; lb_valid; lb_test	Массив нижних границ диапазонов варьируемых параметров (для обучающей, валидационной и тестовой выборок)
ub_train; ub_valid; ub_test	Массив верхних границ диапазонов варьируемых параметров (для каждой из выборок)
Xla_train; Xla_valid; Xla_test	В строках таблиц координаты точек из сгенерированных латинских гиперкубов, нормализованных к диапазону [0; 1]

Dla_train; Dla_valid; Dla_test	В строках таблиц координаты точек из сгенерированных латинских гиперкубов
input	Таблица входных данных для нейронной сети
res_train; res_valid; res_test	Таблицы с данными из выходных RSM файлов симулятора
FOPR_train; FOPR_valid; FOPR_test	Значения дебитов нефти из RSM файлов
FOPR_target	Целевые значения дебитов для нейронной сети
struct_net1_trainscg1, struct_net1_trainscg2, ...	Структуры, содержащие всю информацию о построенных нейронных сетях