УДК 004.89:519.876.5:658.78

ИНТЕГРАЦИЯ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ И МАШИННОГО ОБУЧЕНИЯ ДЛЯ ОПТИМИЗАЦИИ РАСПРЕДЕЛИТЕЛЬНОГО ЦЕНТРА

*Жданов А.С., студент;*

*ORCID: 0009-0004-2901-9108;  
E-mail: cazhdanov@mail.ru;  
Мокшин В.В., к.т.н., научный руководитель, доцент кафедры АСОИУ ФГБОУ ВО «Казанский национальный исследовательский технический университет им. А.Н. Туполева – КАИ», г. Казань, Россия*

INTEGRATION OF SIMULATION MODELING AND MACHINE LEARNING FOR DISTRIBUTION CENTER OPTIMIZATION

*Zhdanov A.S., student;*

*ORCID: 0009-0004-2901-9108;*

*E-mail: cazhdanov@mail.ru;*

*Mokshin V.V., Candidate of Technical Sciences, Scientific Supervisor, Associate Professor of the ASOIU Department, Kazan National Research Technical University named after A.N. Tupolev – KAI, Kazan, Russia*

Аннотация

Данная научно-исследовательская работа посвящена разработке и исследованию интеграции имитационного моделирования и методов машинного обучения для оптимизации работы распределительного центра. Основная цель исследования заключается в создании инновационного инструмента, который позволит не только оптимизировать операции распределительного центра, повышая его производительность и снижая операционные затраты, но и значительно улучшить качество обслуживания клиентов.

Задача исследования включала выбор и изучение имитационной модели распределительного центра в Anylogic, проведение оптимизационного эксперимента с использованием метода генетического алгоритма, экспорт результатов оптимизации в Excel файл и последующий анализ данных в Python на платформе Google Colab. Для подготовки данных использовались библиотеки Pandas, Numpy и Scikit-learn, обучение нейронной сети проводилось с применением инструментов TensorFlow Keras, а итоговые метрики и оценки качества модели визуализировались с помощью библиотек Matplotlib и Seaborn.

В работе детально рассматривается процесс оптимизации распределительного центра, начиная от выбора и детального изучения имитационной модели в Anylogic и заканчивая анализом данных с использованием нейронных сетей. Данные, полученные из Anylogic, были нормализованы и использованы для обучения нейронной сети, основная цель которой заключалась в предсказании значений Objective\_value на основе заданных параметров. Модель была обучена и протестирована на наборе данных, содержащем более 500 записей с различными параметрами и соответствующими значениями Objective\_value.

Результатом исследования стала разработка высокоточной нейронной сети, способной предсказывать значения Objective\_value для различных параметров распределительного центра c оценкой коэффициента детерминации r2\_score=0.88. Модель показала высокую точность предсказаний, что позволяет эффективно использовать её для оценки и оптимизации параметров распределительного центра. Были проведены углубленные анализы и визуализация результатов, что позволило лучше понять производительность модели.

Результаты исследований подтвердили, что интеграция имитационного моделирования и методов машинного обучения существенно улучшает процесс оптимизации распределительного центра. Разработанный инструмент позволяет тестировать и анализировать различные сценарии в виртуальной среде, что способствует принятию более обоснованных решений и повышению эффективности логистических операций.

Abstract

This research is dedicated to the development and exploration of the integration of simulation modeling and machine learning methods for optimizing the operations of a distribution center. The main goal of the research is to create an innovative tool that not only optimizes distribution center operations, thereby enhancing its productivity and reducing operational costs but also significantly improves customer service quality.

The research tasks included selecting and studying the simulation model of the distribution center in Anylogic, conducting an optimization experiment using a genetic algorithm, exporting optimization results to an Excel file, and subsequently analyzing the data in Python on the Google Colab platform. The data preparation utilized Pandas, Numpy, and Scikit-learn libraries, while the neural network was trained using TensorFlow and Keras tools, and the final metrics and model quality assessments were visualized using Matplotlib and Seaborn libraries.

The paper elaborately examines the process of optimizing the distribution center, starting from the selection and detailed study of the simulation model in Anylogic and ending with the data analysis using neural networks. The data obtained from Anylogic were normalized and used to train the neural network, the main goal of which was to predict Objective\_value values based on specified parameters. The model was trained and tested on a dataset containing over 500 records with various parameters and corresponding Objective\_value values.

The outcome of the research was the development of a highly accurate neural network capable of predicting Objective\_value values for various distribution center parameters, with a determination coefficient r2\_score=0.88. The model demonstrated high prediction accuracy, enabling effective evaluation and optimization of distribution center parameters. In-depth analyses and visualization of results were conducted to gain a better understanding of the model's performance.

The research findings confirmed that the integration of simulation modeling and machine learning methods significantly enhances the optimization process of the distribution center. The developed tool allows for testing and analyzing various scenarios in a virtual environment, facilitating the adoption of more informed decisions and improving the efficiency of logistical operations.

Ключевые слова: распределительный центр, нейронная сеть, имитационное моделирование, данные, машинное обучение, оптимизация, глубокое обучение, визуализация данных

Keywords: distribution center, neural network, simulation modeling, data, machine learning, optimization, deep learning, data visualization

**Введение**

Оптимизация распределительных центров (РЦ) представляет собой ключевую задачу для обеспечения эффективности логистических операций в различных отраслях [1]. С увеличением объемов и сложности данных, а также увеличением требований к скорости и точности принимаемых решений, возникает необходимость в инновационных подходах к управлению и оптимизации процессов в РЦ.

Одним из таких подходов является интеграция имитационного моделирования и машинного обучения. Имитационное моделирование (ИМ) — это метод, позволяющий создать модель реальной системы и провести эксперименты с этой моделью для анализа ее поведения и оценки различных стратегий управления [2]. С другой стороны, машинное обучение (МО) представляет собой набор методов и алгоритмов, которые позволяют компьютерам обучаться на данных и делать прогнозы или принимать решения на основе этого опыта.

Интеграция этих двух подходов открывает новые возможности для оптимизации работы РЦ. Путем анализа больших объемов данных с использованием методов МО и проверки различных стратегий управления с помощью ИМ можно достичь более точных и эффективных решений.

В данной научной работе рассматривается именно этот подход - интеграция ИЦ и МО для оптимизации работы РЦ.

Немаловажной частью работы является анализ подобных решений в России и в мире. В НИУ ВШЭ и БГИТУ активно ведутся исследования по интеграции ИМ и МО [3]. Отечественные компании: Сбербанк, Яндекс и X5 Retail Group, NFP [4] активно применяют методы МО для оптимизации своих логистических сетей. Российские исследователи активно участвуют в конференциях и публикуют результаты своих исследований, что способствует обмену опытом и улучшению применяемых методов. Ведущие западные университеты MIT [5] и TU Delft исследуют применение генетических алгоритмов и нейронных сетей для оптимизации логистических процессов. Компания Amazon и другие крупные компании мира внедряют передовые технологии для управления РЦ. Копания H2O.ai внедрила Driverless AI MOJO [6] в имитационную модель «Product Delivery» Anylogic для прогноза текущей температуры РЦ и спроса на каждого дистрибьютора.

**Инструменты и методы**

AnyLogic обеспечивает возможность создания и анализа моделей РЦ, оптимизации процесса распределения грузов и экспорта данных [7].

Выбрана модель «Distribution center» для исследования и оптимизации на Anylogic cloud. Модель имитирует основные процессы РЦ: разгрузку, сборку заказа и загрузку.

Обучение нейронной сети (НС) осуществляется благодаря использованию Pandas для работы с табличными данными, Numpy для массивов и матричных операций, Matplotlib и Seaborn для визуализации, Scikit-learn для разделения данных, кросс-валидации и оценки модели, TensorFlow.keras для создания НС. Стандартизация данных - StandardScaler для улучшения работы моделей, чувствительных к масштабу данных.

Batchnormalization и Dropout – методы, участвующее в построении архитектуры НС. Оптимизатор – Adam, mean\_squared\_error – функция потерь. Оценка модели с использованием различных метрик: MSE, MAE, RMSE, R², MSLE, MAPE, EVS.

**Оптимизация модели РЦ и экспорт результатов**

Для изучения поведения модели и улучшения её производительности можно воспользоваться оптимизацией модели AnyLogic [8]. Оптимизация включает в себя несколько прогонов модели с различными значениями параметров для нахождения наилучших. Оптимизационный эксперимент предоставляет два оптимизатора: Генетический оптимизатор использует эволюционный алгоритм для сохранения разнообразия решений и избегания субоптимальных вариантов, а OptQuest от OptTek Systems, Inc. является алгоритмом-«черным ящиком» для глобальной оптимизации. Процесс оптимизации [9] состоит из выбора оптимизационных параметров и запуска модели, вычисления значения целевой функции после прогона, анализа результата, изменения параметров и повторения процесса.

В модели РЦ необходимо оптимизировать параметры: initialUtilization - коэффициент использования склада (на сколько забит склад), palletTypesNum – количество видов груза, forkliftsNum – количество погрузчиков, numUnloadingDock – количество мест на выгрузку груза, numLoadingDock – количество мест на загрузку груза.

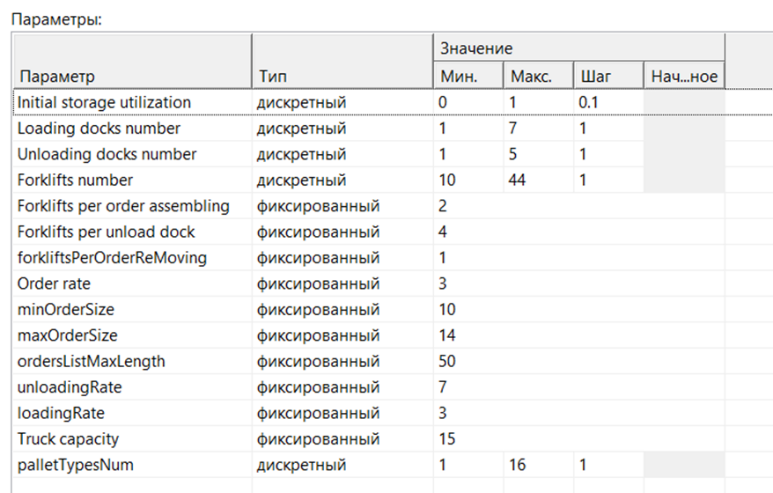


Рис.1. Дискретные и фиксированные параметры оптимизационного эксперимента

Добавляем целевую функцию, которую нужно минимизировать. Выбираем время загрузки грузовика (timeLoadTruck) и применяем к нему функцию getYMean(), которая возвращает среднее значение сохраненных элементов или 0, если элементов нет.

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рис.2. Целевая функция - время загрузки грузовика (timeLoadTruck)

Изображение выглядит как текст, снимок экрана, диаграмма, линия

Автоматически созданное описание

Рис.3. Результат оптимизации времени загрузки грузовика

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание Изображение выглядит как текст, снимок экрана, диаграмма, Шрифт

Автоматически созданное описание

Рис.4. Сравнение показателей до и после оптимизации среднего времени загрузки товара в грузовик (слева), утилизации загрузки груза (справа)

Параметры, которые были использованы в оптимизации, экспортируются в Excel файл для будущего анализа данных. В действиях, после итерации, в свойствах оптимизатора задаем значение в определенную ячейку на Java.

Изображение выглядит как текст, снимок экрана, линия, Шрифт

Автоматически созданное описание

Рис.5. Действия по экспорту оптимизационных данных из Anylogic в Excel файл

Далее необходимо подготовить данные к дальнейшему обучению и тестированию НС. Выберем столбцы с параметрами, которые изменяются при итерировании, и экспортируем в CSV (Comma-Separated Values). Получим подготовленные данные для импорта в python.

**Анализ данных результатов оптимизации распределительного центра и обучение нейронной сети**

Для анализа данных и обучения НС, которая будет предсказывать значение целевой переменной (Objective Value) на основе заданных параметров (Initial storage utilization, Loading docks number, Unloading docks number, Forklifts number, palletTypesNum), мы можем использовать библиотеки pandas, scikit-learn и tensorflow. Сам код написан в среде разработки Google Colab.

Первое, что необходимо сделать это загрузить данные из CSV-файла. Разделить данные на обучающую и тестовую выборки. Нормализовать. Построить и обучить НС. Оценить модель. Произвести кросс валидацию. Оценить результаты кросс валидации.

Pandas необходим для работы с табличными данными (DataFrame). Используется для загрузки, анализа и манипуляции данными. Numpy нужен для определение массивов и матричных операций, математических функций. Matplotlib - построение графиков и визуализация данных. Seaborn - расширенные возможности для визуализации данных, основанные на matplotlib. Sklearn (scikit-learn) – (train\_test\_split - разделение данных на обучающую и тестовую выборки, KFold - кросс-валидация, StandardScaler - нормализация данных, метрики (mean\_squared\_error, mean\_absolute\_error, r2\_score и др.) - оценка качества модели). Tensorflow.keras - (Sequential, Dense, Dropout, BatchNormalization - создание и настройка слоев НС, Adam - оптимизатор для обучения модели, model.compile(), model.fit(), model.predict() - компиляция, обучение и предсказание с помощью модели) [10]. Также необходимо подключиться к гугл диску для импорта датасета, для этого понадобится drive из google.colab [11].

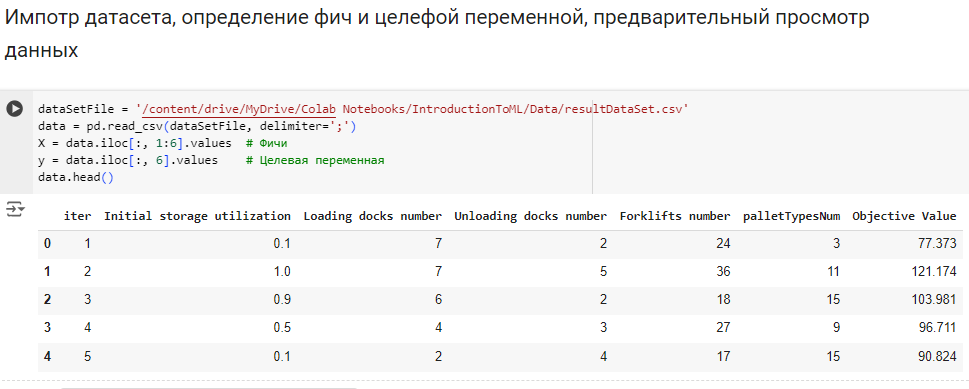


Рис.6. Импорт датасета

Далее разделим данные на обучающие и тестовые выборки. X - матрица признаков (features), где каждая строка представляет собой один образец (sample), а каждый столбец - один признак (feature). Вектор целевых переменных (targets), где каждый элемент соответствует целевой переменной для одного образца. Test\_size - определяет размер тестовой выборки. Может быть представлен в виде доли (от 0 до 1) или в виде абсолютного числа образцов. Random\_state - фиксирует начальное состояние генератора случайных чисел для воспроизводимости результата.

Стандартизация данных [12] является важным этапом предварительной обработки данных в МО. В данном случае используется StandardScaler из библиотеки scikit-learn, который стандартизирует признаки путем удаления среднего значения и масштабирования до единичной дисперсии. Это особенно полезно для моделей, которые чувствительны к масштабу данных, таких как линейная регрессия или методы, использующие градиентный спуск.

Изображение выглядит как текст, снимок экрана, Шрифт, алгебра

Автоматически созданное описание

Рис.7. Разделение данных на обучающую и тестовую выборки, нормализация данных

На следующем шаге строим саму модель, ее архитектуру из 4 нейронных слоев. После создания каждого из слоев используем нормализацию батчей Batchnormalization (для ускорения) [13] и dropout (выключатель нейронов с определенной вероятностью в каждой эпохе для предотвращения их совместной адаптации) [14]. Используем функцию активации ReLu. Компилируем модель с использованием оптимизатора Adam, где learning\_rate - размер шага, который оптимизатор делает в направлении минимума функции потерь. Метод compile в Keras используется для настройки процесса обучения модели. Он принимает функцию потерь (loss), оптимизатор (optimizer). Этот метод подготавливает модель для обучения, связывая оптимизатор и функцию потерь с моделью. Функция потерь определяет, как хорошо модель работает в сравнении с истинными значениями. В данном случае используется mean\_squared\_error (Среднеквадратическая ошибка)

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рис.8. Архитектура нейронной сети и компиляция модели

После компиляции модели можно начать обучать ее на тренировочных данных. Epochs - количество эпох, в течение которых модель будет обучаться. Batch\_size - размер батча (пакета), который используется для обновления градиентов и весов модели. Обучение модели происходит на батчах данных, а не на всех данных сразу, что позволяет использовать меньше оперативной памяти и ускоряет процесс обучения. Validation\_split - доля тренировочных данных. Verbose - уровень вывода информации во время обучения.

Изображение выглядит как текст, снимок экрана, Шрифт, линия

Автоматически созданное описание

Рис.9. Обучение модели

Проведем оценку модели [15], выведем основные метрики, определяющие успех обучения и наличия той или иной погрешности и ошибки.

MSE (Mean Squared Error) — среднеквадратичная ошибка. MSE вычисляется как среднее значение квадратов разностей между предсказанными и фактическими значениями. Хорошо выявляет большие ошибки.

MAE (Mean Absolute Error) — средняя абсолютная ошибка. MAE вычисляется как среднее значение абсолютных разностей между предсказанными и фактическими значениями.

RMSE (Root Mean Squared Error) — корень среднеквадратичной ошибки.

R² Score (Coefficient of Determination) — коэффициент детерминации. R² показывает долю дисперсии зависимой переменной, которая объясняется моделью. R² измеряет, насколько хорошо модель объясняет вариацию данных. Чем ближе значение R² к 1, тем лучше качество у модели, объяснительная сила.

MSLE (Mean Squared Logarithmic Error) — среднеквадратичная логарифмическая ошибка. MSLE вычисляется как среднее значение квадратов логарифмических разностей между предсказанными и фактическими значениями. MSLE полезен для задач с экспоненциальным ростом данных.

MAPE (Mean Absolute Percentage Error) — средняя абсолютная процентная ошибка. MAPE вычисляется как среднее значение абсолютных процентных разностей между предсказанными и фактическими значениями. MAPE измеряет среднюю величину ошибки в процентах. Полезна для данных, где масштаб ошибки важен.

EVS (Explained Variance Score) — Объясненная дисперсия. EVS измеряет, насколько хорошо предсказанные значения объясняют дисперсию истинных значений. EVS похожа на R², но учитывает только вариацию ошибки.

Далее выведем график потерь во время обучения. По графику видно, что с увеличением эпохи потери уменьшаются и стремятся к 0. Метрики после обучения: MSE: 50.082402437835135, MAE: 4.940964542463714, RMSE: 7.076892145414902, R^2 Score: 0.8741026529422047, MSLE: 0.006291146304758162, MAPE: 0.059373847214856834, EVS: 0.8746800215263353.

Изображение выглядит как текст, линия, График, снимок экрана

Автоматически созданное описание

Рис.10. График потерь во время обучения

Кросс-валидация — это метод оценки производительности модели МО, используемый для проверки её способности обобщать данные на новых, ранее не встречавшихся случаях. Кросс-валидация позволяет: оценить производительность модели на новых данных, выявить переобучение модели и настроить гиперпараметр, сравнить различные алгоритмы МО и выбрать наиболее подходящий.

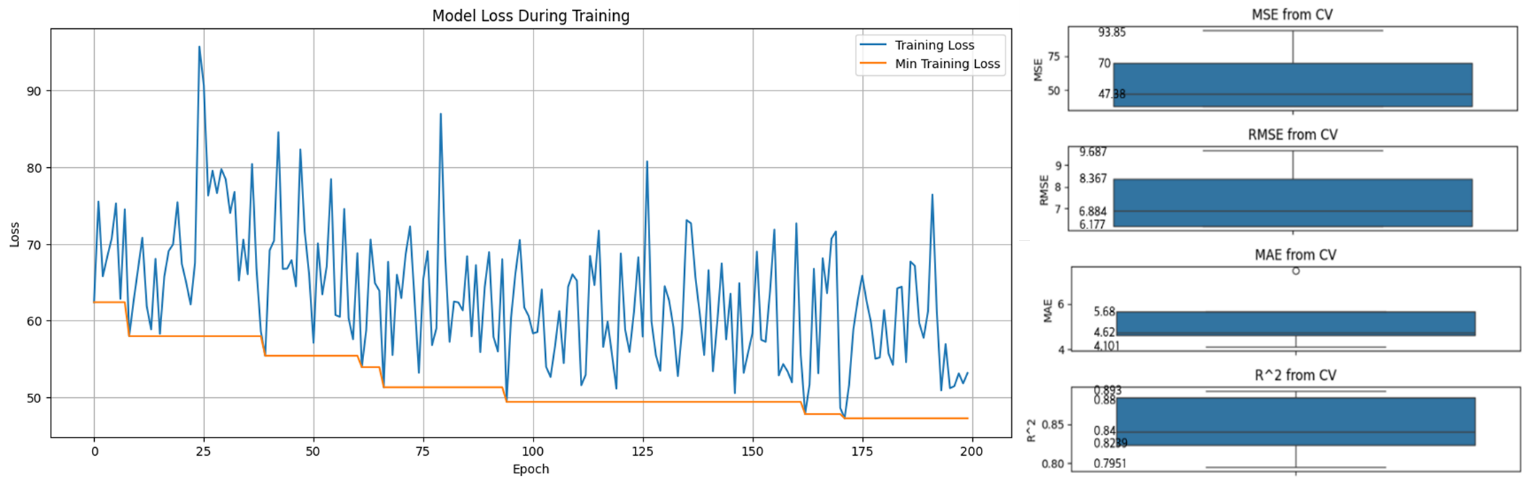


Рис.11. График потерь во время обучения и визуализация метрик после кросс валидации.

Заметим, что модель вывела Predicted Objective Value близкий к минимальному значению в нашем датасете, поэтому можно сказать, что данная комбинация параметров (0, 7, 1, 16, 2) соответствующая Initial storage utilization, Loading docks number, Unloading docks number, Forklifts number, palletTypesNum близка к оптимальному.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описаниеИзображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рис.12. Предсказание на новых данных и вывод метрик

Возможно, что данные в датасете могут меняться и усложняться. Для того, чтобы дообучить модель, необходимо загрузить обновленный датасет, разделить данные на обучающую и тестовую выборки, продолжить обучение модели на новых тестовых данных (см. рис. 13), провести оценку. После дообучения получили следующие метрики: MSE: 61.87045224170733, MAE: 5.971018330891927, RMSE: 7.865777281471129, R^2 Score: 0.849608089630033, MSLE: 0.009063899054494427, MAPE: 0.07617345820551844, EVS: 0.8498519526584907.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рис.13. Реализация дополнительного обучения модели на новых данных

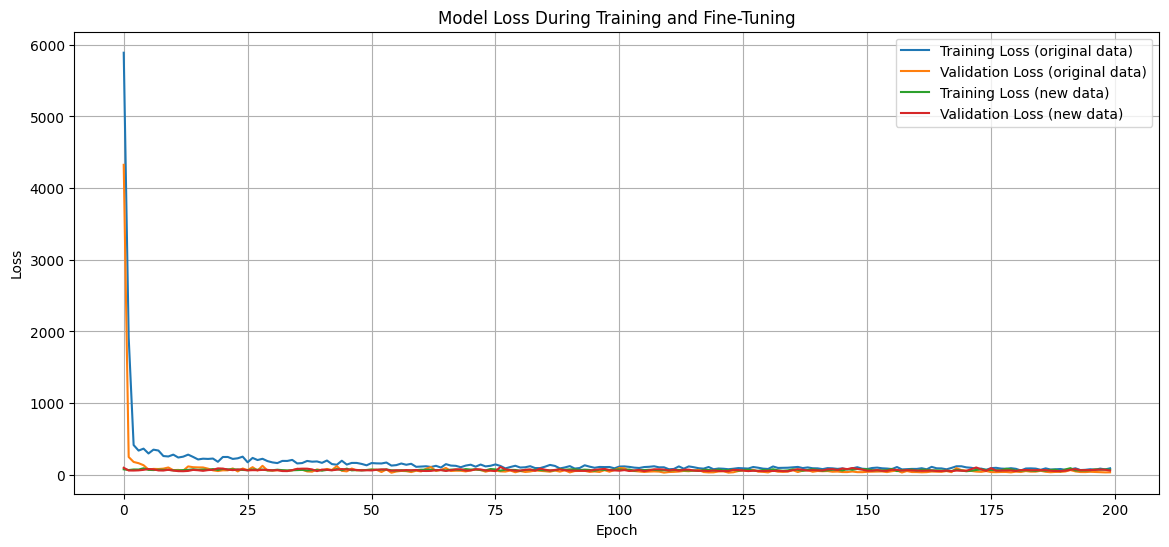


Рис.14. График потерь до и после обновления данных

**Заключение**

Исследование подтвердило, что оптимизация процессов РЦ, основанная на использовании ИМ и агентно-ориентированного подхода в среде AnyLogic, может значительно повысить его эффективность. Обученная НС для предсказания оптимизационного коэффициента по начальным параметрам (Initial storage utilization, Loading docks number, Unloading docks number, Forklifts number, palletTypesNum) продемонстрировало высокую точность. Метрики, полученные в ходе оценки модели, такие как MSE (50.0824), MAE (4.94), RMSE (7.07689), R^2 Score (0.8741), MSLE (0.00629), MAPE (0.05937) и EVS (0.87468), подтверждают высокую степень точности и надежности прогнозов, а также результаты дообучения показывают, что модель способна давать результат близкий к результату до обновления данных.

Оптимизация с использованием ИМ и искусственного интеллекта открывает новые возможности для улучшения логистических операций и управления запасами.

**Список литературы**

1. Wan L., Guan Z., Shao X. Simulation based study on the layout of an enterprise distribution center //Proceedings 2011 International Conference on Transportation, Mechanical, and Electrical Engineering (TMEE). – IEEE, 2011. – С. 279-283.

2. Dolgova O. I., Kryukov S. V. Simulation of business processes of service support of acquiring products in the AnyLogic software environment //St. Petersburg State Polytechnical University Journal. Economics. – 2021. – Т. 14. – №. 6. – С. 117.

3. Казаков О. Д., Азаренко Н. Ю. Комбинирование методов машинного обучения и имитационного моделирования социально-экономических процессов в системах поддержки принятия решений //Вестник РГРТУ. – 2020. – №. 71. – С. 97-107.

4. NFP консалтинговая компания. – URL: nfp2b.ru/2019/11/27/mashinnoe-obuchenie-v-imitatsionnyh-modelyah/ (дата обращения: 04.06.2024). – Текст: электронный.

5. MIT Center for Transportation & Logistics. – URL: ctl.mit.edu/pub/report/artificial-intelligence-machine-learning-roundtable (дата обращения: 04.06.2024). – Текст: электронный.

6. Anylogic: имитационное моделирование для бизнеса. – URL: ruanl.anylogic.online/features/artificial-intelligence/h2o-ai/ (дата обращения: 04.06.2024). – Текст: электронный.

7. Borshchev A. Multi‐method modelling: AnyLogic //Discrete‐event simulation and system dynamics for management decision making. – 2014. – С. 248-279.

8. Акельдов Д. В. Оптимизация работы распределительного центра в среде имитационного моделирования AnyLogic //Девятая всероссийская научно-практическая конференция по имитационному моделированию и его применению в науке и промышленности. – 2019. – С. 352-355.

9. Muravev D. et al. Multi-agent optimization of the intermodal terminal main parameters by using AnyLogic simulation platform: Case study on the Ningbo-Zhoushan Port //International Journal of Information Management. – 2021. – Т. 57. – С. 102-133.

10. Chollet F. Deep learning with Python. – Simon and Schuster, 2021.

11. Kanani P., Padole M. Deep learning to detect skin cancer using google colab //International Journal of Engineering and Advanced Technology Regular Issue. – 2019. – Т. 8. – №. 6. – С. 2176-2183.

12. Jo J. M. Effectiveness of normalization pre-processing of big data to the machine learning performance //The Journal of the Korea institute of electronic communication sciences. – 2019. – Т. 14. – №. 3. – С. 547-552.

13. Ioffe S., Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift //International conference on machine learning. – pmlr, 2015. – С. 448-456.

14. Srivastava N. et al. Dropout: a simple way to prevent neural networks from overfitting //The journal of machine learning research. – 2014. – Т. 15. – №. 1. – С. 1929-1958.

15. Raschka S. Model evaluation, model selection, and algorithm selection in machine learning //arXiv preprint arXiv:1811.12808. – 2018.