

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

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

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

магистерская диссертация

БЕЗЫТЕРАЦИОННОЕ РЕШЕНИЕ ЗАДАЧИ ТОПОЛОГИЧЕСКОЙ ОПТИМИЗАЦИИ УПРУГИХ ТЕЛ С ПОМОЩЬЮ МАШИННОГО ОБУЧЕНИЯ

по направлению подготовки (специальности)

01.04.03 Механика и математическое моделирование

Направленность (профиль)

01.04.03_04 Математическое моделирование процессов нефтегазодобычи

Выполнил

студент гр. 5040103/10401

А.И. Матвеева

Руководитель

доцент ВШМиПУ, ФизМех, к.т.н.

А.Д. Новокшенов

Санкт-Петербург

2023

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

УТВЕРЖДАЮ

Директор ВШТМиМФ

А.М. Кривцов

«__» _____ 20__ г.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы
студенту Матвеевой Анастасии Игоревне, гр. 5040103/10401

1. Тема работы: Безытерационное решение задачи топологической оптимизации упругих тел с помощью машинного обучения.
2. Срок сдачи студентом законченной работы: 26.05.2023.
3. Исходные данные по работе: опубликованные результаты исследований в данной области, данные о свойствах материалов, геометрические данные объекта, граничные условия и внешние нагрузки.
4. Содержание работы (перечень подлежащих разработке вопросов): обзор литературных источников по теме ВКР, создание нейронной сети для прогнозирования оптимальной топологии конструкции, проверка нейронной сети на оптимизируемой модели и сравнение результатов с численным методом.
5. Перечень графического материала (с указанием обязательных чертежей): не предусмотрено.
6. Дата выдачи задания: 27.02.2023.

Руководитель ВКР _____ А.Д. Новокшенов, доцент ВШМиПУ, ФизМех, к. т. н.

Задание принял к исполнению 27.02.2023

Студент _____ А.И. Матвеева

РЕФЕРАТ

На 55 с., 23 рисунка

МАШИННОЕ ОБУЧЕНИЕ, ГЕНЕРАТИВНО-СОСТЯЗАТЕЛЬНАЯ НЕЙРОННАЯ СЕТЬ, СВЕРТОЧНАЯ НЕЙРОННАЯ СЕТЬ, ТОПОЛОГИЧЕСКАЯ ОПТИМИЗАЦИЯ, АДАПТИВНЫЕ МАТЕРИАЛЫ, БИБЛИОТЕКА PYTORCH, МЕТОД СКОЛЬЗЯЩИХ АСИМПТОТ, SIMP МЕТОД

В данной выпускной квалификационной работе в качестве альтернативы традиционным методам топологической оптимизации деформируемых твердых тел предложен подход к топологической оптимизации на основе методов машинного обучения. Предложенный подход существенно уменьшает время, затрачиваемое на получение оптимального решения, и позволяют избежать использования ресурсоемких конечно-элементных вычислений на стадии получения оптимального решения. Все ресурсоемкие вычисления выполняются на стадии тренировки сети. Для предсказания оптимальной топологии используется двухэтапная модель, состоящая из сверточной и генеративно-состязательной нейронных сетей. На первом этапе сверточная нейронная сеть прогнозирует топологию в низком разрешении на основе информации о нагрузках и объемной доли. Затем, с помощью генеративно-состязательной нейронной сети, происходит уточнение и повышение разрешения полученной топологии. Эффективность предложенного метода была успешно демонстрируется на примере квадратной упругой пластинки. Также обсуждаются результаты и нерешенные проблемы метода.

ABSTRACT

55 pages, 23 pictures,

MACHINE LEARNING, GENERATIVE ADVERSARIAL NETWORK, CONVOLUTIONAL NEURAL NETWORK, TOPOLOGICAL OPTIMIZATION, ADAPTIVE MATERIALS, PYTORCH LIBRARY, MOVING ASYMPTOTES, SIMP METHOD

In this graduation thesis, as an alternative to traditional methods of topology optimization of deformable solids, an approach to topology optimization based on machine learning methods is proposed. The proposed approach significantly reduces the time spent on obtaining the optimal solution and allows to avoid the use of time-consuming finite element calculations at the stage of obtaining the optimal solution. All of time-consuming calculations are performed at the network training stage. A two-stage model, comprising a convolutional neural network (CNN) and a generative adversarial network (GAN), is used to predict the optimal topology. In the first stage, the CNN predicts the topology at a low resolution based on load information and volume fraction. Subsequently, the generated topology is refined and its resolution is increased using the GAN. The effectiveness of the proposed method has been successfully demonstrated on a square elastic plate. The results and unresolved issues of the method are also discussed.

СОДЕРЖАНИЕ

| | |
|--|----|
| ВВЕДЕНИЕ..... | 6 |
| ГЛАВА 1. ТОПОЛОГИЧЕСКАЯ ОПТИМИЗАЦИЯ | 8 |
| 1.1. Математическая постановка топологической оптимизации | 8 |
| 1.2. Проектные переменные | 10 |
| 1.3. Метод выполнения критерия оптимальности | 11 |
| 1.4. Метод скользящих асимптот..... | 14 |
| ГЛАВА 2. НЕЙРОННЫЕ СЕТИ..... | 18 |
| 2.1. Обзор методов машинного обучения в задаче топологической оптимизации..... | 18 |
| 2.1.1. Ускорение итеративного процесса с помощью методов машинного обучения | 19 |
| 2.1.2. Упрощение поиска производных с помощью методов машинного обучения | 20 |
| 2.1.3. Уменьшение пространства переменных проектирования с помощью методов машинного обучения | 21 |
| 2.1.4. Улучшение качества изображений в задаче топологической оптимизации с помощью методов машинного обучения | 23 |
| 2.1.5. Использование нейронных сетей для достижения безытерационной оптимизации | 24 |
| 2.2. Математическая модель нейрона..... | 27 |
| 2.3. Метод обратного распространения ошибки | 28 |
| 2.4. Функции активации..... | 32 |

| | |
|--|----|
| ГЛАВА 3. РАЗРАБОТКА ДВУХЭТАПНОЙ МОДЕЛИ ДЛЯ ПРЕДСКАЗАНИЯ ОПТИМАЛЬНОЙ ТОПОЛОГИИ УПРУГОЙ ПЛАСТИНЫ | 36 |
| 3.1. Сверточная нейронная сеть..... | 36 |
| 3.2. Генеративно-состязательная нейронная сеть..... | 40 |
| 3.3. Разработка двухэтапной модели..... | 43 |
| 3.4. Результаты..... | 48 |
| ЗАКЛЮЧЕНИЕ | 50 |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ..... | 51 |

ВВЕДЕНИЕ

Один из наиболее распространенных и актуальных методов численной оптимизации в области механики деформируемых твердых тел — это метод топологической оптимизации. В классической формулировке, топологическая оптимизация направлена на поиск оптимального распределения материала в заданной области с учетом заданных нагрузок и граничных условий с точки зрения повышения жесткости системы. Другими словами, топологическая оптимизация в механике деформируемых твердых тел — это метод, который помогает инженерам автоматизировать процесс разработки эффективных конструкций для различных задач. Обычно инженеры могут предложить начальную форму для деталей, но вопрос заключается в том, насколько эта форма оптимальна. Топологическая оптимизация позволяет найти наилучшую форму детали, учитывая заданные ограничения, такие как жесткость, масса или другие характеристики системы.

На сегодняшний день доступно множество численных методов для проведения топологической оптимизации. Среди них SIMP-метод (метод с пенализацией для твердых изотропных материалов), BESO-метод (метод эволюционной структурной оптимизации с двунаправленным поиском) и Level-set-методы (методы установления уровня). Важным компонентом в топологической оптимизации является метод конечных элементов, который обеспечивает моделирование поведения конструкции. Кроме того, применяются методы оптимизации, такие как методы нелинейного программирования и эволюционные алгоритмы для решения задач на поиск минимума или максимума функционала.

Методы нелинейного программирования, например, предоставляют мощные инструменты для решения оптимизационных задач с нелинейными ограничениями и целевыми функциями. Они основаны на математической теории оптимизации и используют итерационные алгоритмы для нахождения оптимального решения.

Эволюционные алгоритмы, в свою очередь, представляют собой стохастические методы оптимизации, вдохновленные процессами эволюции в природе. Они используют понятия, такие как генетическое кодирование, селекция,

мутация и скрещивание, для поиска оптимальных решений. Эволюционные алгоритмы особенно полезны в задачах с большими пространствами поиска или сложными нелинейными функциями.

С развитием аддитивного производства и инструментов проектирования, интерес к оптимизации топологии значительно возрос. Однако, несмотря на обширную математическую базу, необходимо отметить, что топологическая оптимизация требует значительных вычислительных ресурсов. В свете этого, появляется необходимость в разработке новых подходов, таких как методы машинного обучения, которые могут предоставить новые возможности для более эффективного и быстрого решения задач оптимизации топологии.

Таким образом, целью данной работы является разработка модели машинного обучения, позволяющей по входным параметрам, таким как, приложенная нагрузка, ограничения и объемная доля предсказать оптимальную топологию конструкции.

Для достижения поставленной цели рассматриваются следующие задачи:

1. Анализ литературных источников и изучение предложенных моделей для выявления преимуществ и недостатков, которые будут учтены при разработке собственной модели.
2. Разработка высокоэффективной архитектуры нейронных сетей, способных предсказывать оптимальное распределение материала в конструкции.
3. Генерация обучающих данных с использованием разработанного топологического оптимизатора, который позволяет создавать разнообразные варианты топологий конструкции на основе заданных параметров.
4. Обучение и настройка модели машинного обучения на сгенерированных данных для достижения оптимальных результатов.
5. Тестирование разработанной модели на упругой квадратной пластине для проверки ее эффективности и точности предсказания оптимальной топологии.

ГЛАВА 1. ТОПОЛОГИЧЕСКАЯ ОПТИМИЗАЦИЯ

1.1. Математическая постановка топологической оптимизации

На рисунке рассматривается упругое тело объемом V , ограниченное поверхностью S , подверженное кинематическим и статическим граничным условиям. Тело подвергается воздействию массовых сил \mathbf{K} , и поверхностным силам \mathbf{F} на границе S_2 .

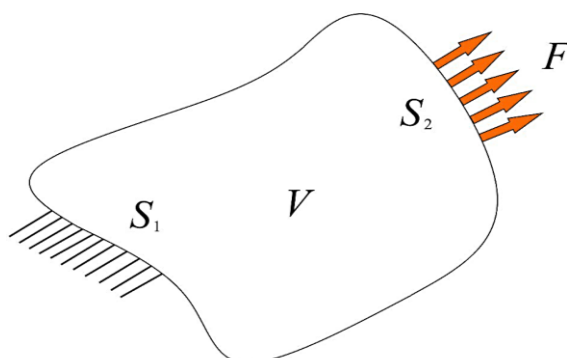


Рис.1.1. Постановка задачи топологической оптимизации

Работа внешних сил для упругого деформируемого тела (Рис.1.1), находящегося под действием кинематических и статических граничных условий, имеет вид:

$$a_{(e)}(\mathbf{u}) = \int_V \rho \mathbf{K} \cdot \mathbf{u} dV + \int_{S_2} \mathbf{F} \cdot \mathbf{u} dS \quad (1.1)$$

где \mathbf{K} – интенсивность массовых сил, \mathbf{F} – вектор сил, распределенный по поверхности S_2 , \mathbf{u} – вектор перемещения точек тела.

Элементарная работа внешних сил $\delta' a_{(e)}$ может быть отождествлена с вариацией потенциальной энергии деформации δa , равной вариации свободной энергии.

$$\delta' a_{(e)} = \delta a = \delta \int_V A dV = \int_V \rho \mathbf{K} \cdot \delta \mathbf{u} dV + \int_{S_2} \mathbf{F} \cdot \delta \mathbf{u} dS \quad (1.2)$$

где A – удельная потенциальная энергия.

Потенциальная энергия деформации имеет вид:

$$a = \rho_T \psi(\boldsymbol{\varepsilon}(\mathbf{u})) = \frac{1}{2} \boldsymbol{\sigma} \cdot \boldsymbol{\varepsilon} = \frac{1}{2} ({}^4C \cdot \boldsymbol{\varepsilon}) \cdot \boldsymbol{\varepsilon}, \quad (1.3)$$

где ρ_T – плотность тела в окрестности рассматриваемой материальной точки, ψ – свободная энергия, $\boldsymbol{\sigma}$ – тензор напряжений, $\boldsymbol{\varepsilon}$ – тензор деформаций, 4C – тензор упругости четвертого ранга.

В формуле (1.3) содержится утверждение, что работа внешних массовых и поверхностных сил на виртуальном перемещении $\delta \mathbf{u}$ точек упругого тела из положения равновесия, определяемого вектором \mathbf{u} , равна вариации потенциальной энергии деформации.

$$a_{(e)}(\delta \mathbf{u}) = \delta(\rho_T \psi(\boldsymbol{\varepsilon}(\mathbf{u}))) \quad (1.4)$$

Математическая формулировка поиска минимума податливости или максимума жесткости тела эквивалента поиску минимума работы внешних сил при заданных ограничениях:

$$\min_{\mathbf{u} \in U, E} a_{(e)}(\mathbf{u})$$

при ограничениях:

$$a_{(e)}(\delta \mathbf{u}) = \delta(\rho_T \psi(\boldsymbol{\varepsilon}(\mathbf{u}))), \text{ для любого } \delta \mathbf{u} \in U \quad (1.5)$$

$$V \leq aV_0, 0 < a < 1,$$

где V – объем в результате оптимизации, V_0 – первоначальный объем тела, U – поле кинематически-допустимых перемещений.

В математической постановке (1.5) присутствуют условие на объем V , который получится в результате топологической оптимизации.

Это ограничение связано с одной из основных целей топологической оптимизации: получение максимальной жесткости системы при заданных ограничениях на использование материальных ресурсов. Перемещения точек тела находятся из уравнения равновесия в методе конечных элементов задачи

топологической оптимизации с учетом заданных нагрузок. Таким образом, постановка задачи топологической оптимизации в терминах метода конечных элементов имеет следующий вид:

$$\min_{\mathbf{u} \in U, E_e} \mathbf{f}^T \mathbf{u}$$

при ограничениях $\mathbf{K}(E_e)\mathbf{u} = \mathbf{f}, E_e \in E_{\text{доп}},$

(1.6)

$$\mathbf{K} = \sum_{e=1}^N \mathbf{K}_e(E_e), \sum_{e=1}^N v_e = V = \alpha V_0, 0 < \alpha < 1$$

где \mathbf{K}_e – матрица жесткости элемента, \mathbf{f}^T – вектор сил, \mathbf{u} – вектор перемещений, v_e – объем конечного элемента с индексом e .

1.2. Проектные переменные

При топологической оптимизации в рамках метода конечных элементов, переменные проектирования используются для описания распределения материала в заданной области. Они определяют степень присутствия материала в каждом элементе. Подход заключается в использовании переменных плотности ρ в каждом элементе, указывающие на количество материала в данном элементе. Значения переменных плотности обычно выбираются в интервале от 0 до 1, где 0 соответствует полностью пустому элементу, а 1 соответствует полностью заполненному элементу. Таким образом, решение топологической оптимизации может представлять изображение, составленное из черных и белых пикселей. Черный пиксель соответствует наличию материала, а белый – его отсутствию. Для применяемых численных методов необходимо, чтобы проектная переменная являлась непрерывной величиной. Соответственно могут появиться промежуточные значения – серые пиксели, которые невозможно будет физически интерпретировать.

Для решения данной проблемы в топологической оптимизации применяется метод Solid Isotropic Material with Penalization (SIMP). В данном методе предполагается, что каждая точка изотропного материала обладает упругими свойствами, которые могут быть описаны следующим соотношением.

$$E(x) = \rho(x)^p E^0, \quad (1.7)$$

где ρ – фиктивная плотность ($0 \leq \rho \leq 1$), p – штрафной параметр.

Штрафной параметр в методе SIMP применяется для предотвращения возникновения промежуточных плотностей. Он вводится в математическую модель с целью управления изменением плотности материала в каждой точке тела. Экспериментально было получено, что оптимальное значения штрафного параметра варьируется в диапазоне от 3 до 5. Для меньших значений результат топологической оптимизации оказывается недостаточно четким за счет появления промежуточных плотностей. Для сильно больших значений штрафного параметра происходит значительное искажение конструкции, увеличение податливости, ухудшение сходимости.

Данный метод отлично зарекомендовал себя и активно применяется в программных обеспечениях топологической оптимизации (DS Simulia Tosca и Altair Optistruct).

1.3. Метод выполнения критерия оптимальности

Для соблюдения ограничений на смещения и объем необходимо изменить функцию Лагранжа, введя соответствующие множители Лагранжа. Целевая функция Лагранжа для заданных ограничений может быть представлена следующим образом:

$$L(\mathbf{u}, \rho) = a_{(e)}(\mathbf{u}) - \{\rho_T \psi(\boldsymbol{\varepsilon}(\mathbf{u}), \boldsymbol{\varepsilon}(\bar{\mathbf{u}})) - a_{(e)}(\bar{\mathbf{u}})\} + \Lambda \left(\int_V \rho(x) dV - V \right) + \int_V \lambda^+(x) (\rho(x) - 1) dV + \int_V \lambda^-(x) (\rho_{min} - \rho(x)) dV \quad (1.8)$$

где $\Lambda, \lambda^-(x), \lambda^+(x)$ – множители Лагранжа для ограничений, связанных с переменными проектирования ρ и объемом тела, $\bar{\mathbf{u}}$ – множитель Лагранжа для уравнения равновесия.

Оптимальное условие получается при дифференцировании функционала Лагранжа по переменным проектирования:

$$\frac{\partial L(\mathbf{u}, \rho)}{\partial \rho} = 0 \quad (1.9)$$

Предполагая, что $0 < \rho_{min} \leq \rho$ (что гарантирует единственность полей смещения), условие оптимальности по отношению к вариациям поля смещений \mathbf{u} дают $\bar{\mathbf{u}} = \mathbf{u}$. Условие для ρ принимает следующий вид:

$$\frac{\partial E_{ijkl}}{\partial \rho} \varepsilon_{ij}(\mathbf{u}) \varepsilon_{kl}(\mathbf{u}) = \Lambda - \lambda^- + \lambda^+ \quad (1.10)$$

$$\lambda^- \geq 0, \lambda^+ \geq 0, \lambda^-(x) (\rho_{min} - \rho(x)) = 0, \lambda^+(x) (\rho(x) - 1) = 0$$

Для промежуточных значений переменных проектирования $\rho_{min} < \rho < 1$ выражение принимает вид:

$$p\rho(x)^{p-1} E_{ijkl}^0 \varepsilon_{ij}(\mathbf{u}) \varepsilon_{kl}(\mathbf{u}) = \Lambda \quad (1.11)$$

Левая часть данного выражения является взвешенной потенциальной энергией деформации со множителем $p\rho(x)^{p-1}$. Согласно полученному условию, равенство энергии должно выполняться в каждой точке оптимального решения.

Допустим, на первой итерации фиктивная плотность уменьшилась в конечных элементах для удовлетворения ограничения на объем в соответствии с заданной объемной долей. Решив задачу теории упругости и вычислив взвешенную энергию в каждой точке тела, можно обнаружить, что взвешенная энергия распределена между точками тела неравномерно. Точки с повышенной энергией деформации указывают на недостаточную жесткость, в то время как точки с низкой энергией указывают на избыточную жесткость.

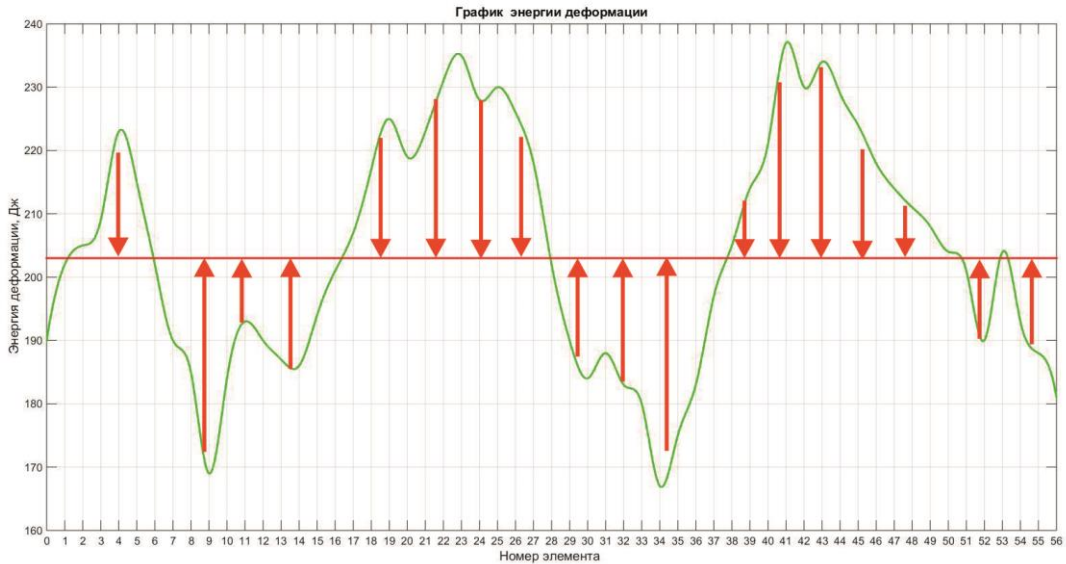


Рис.1.2. Распределение величины взвешенной энергии деформации в объеме тела (для наглядности представлен одномерный случай)

Для корректировки ρ_K в этих точек может быть применен следующий алгоритм:

$$\rho_{K+1} = \begin{cases} \max\{(1 - \xi)\rho_K, \rho_{min}\} & \text{если } \rho_K B_K^\eta \leq \max\{(1 - \xi)\rho_K, \rho_{min}\}, \\ \min\{(1 - \xi)\rho_K, 1\} & \text{если } \min\{(1 - \xi)\rho_K, 1\} \leq \rho_K B_K^\eta \\ \rho_K B_K^\eta & \text{в других случаях} \end{cases} \quad (1.12)$$

где K означает номер итерации

Переменная B_K в свою очередь определяется следующим выражением:

$$B_K = \Lambda_K^{-1} \rho \rho(x)^{p-1} E_{ijkl}^0 \varepsilon_{ij}(u_K) \varepsilon_{kl}(u_K) \quad (1.13)$$

где u_K – поле перемещений, полученное на итерации по номером K .

Для расчета поля перемещений u_K используется метод конечных элементов, основанный на уравнении равновесия. Таким образом, идея данного метода состоит в добавлении материала в области, где взвешенная энергия деформации превышает значение Λ , и удалении материала в противном случае.

Параметры η и ξ могут быть настроены для регулирования изменений, которые могут происходить на каждом шаге итерации, с целью обеспечения эффективности метода. Значение, которое примет плотность ρ_{K+1} на следующем

шаге, зависит от значения множителя Лагранжа Λ_K на текущем шаге, который в свою очередь определяется в соответствии с условием ограничения на объем. Объем на следующем шаге V_{K+1} является монотонно невозрастающей функцией от Λ_K и считается как сумма всех плотностей. Следовательно, вычисление значения Λ_K , удовлетворяющего ограничению на объем, может быть однозначно определено с использованием численной процедуры, например, методом бисекции или метода Ньютона. Стабильность итерационного метода зависит от выбора параметров, и поэтому параметры подбираются экспериментально. Обычно используются значения 0.5 для коэффициента η и 0.2 для коэффициента ξ .

1.4. Метод скользящих асимптот

Кристиан Сванберг в 1989 г предложил метод скользящих асимптот. Данный метод представляет собой метод нелинейного программирования и используется при решении задач оптимизации, постановка которых имеет следующий вид:

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^n} f_0(\mathbf{x}) \\ \text{при ограничениях: } f_i(\mathbf{x}) \leq \widehat{f}_i, i = \overline{1, \dots, m} \\ \underline{x}_j \leq x_j \leq \overline{x}_j, j = \overline{1, \dots, n} \end{aligned} \quad (1.14)$$

где $\mathbf{x} = (x_1, \dots, x_n)$ – вектор переменных проектирования, $f_0(\mathbf{x})$ – целевая функция, $f_i(\mathbf{x})$ – заданные ограничения (обычно на напряжения, перемещения), $\underline{x}_j, \overline{x}_j$ – пределы варьирования проектных переменных.

Успешно зарекомендовавший себя общий подход к решению таких проблем заключается в создании и решении последовательности явных подзадач по следующей итерационной схеме:

Шаг 0. Выбирается точка отсчета $\mathbf{x}^{(0)}$, номеру итерации присваивается $k = 0$.

Шаг 1. Значения функций $f_i(\mathbf{x}^{(k)})$ и их градиенты $\nabla f_i(\mathbf{x}^{(k)})$ рассчитываются в точке $\mathbf{x}^{(k)}$ для итерации с номером k для $i = \overline{0, m}$.

Шаг 2. На основе информации, полученной на первом шаге, создается подзадача оптимизации, осуществляя замену оригинальных f_i в исходной задаче аппроксимирующими функциями $f_i^{(k)}$.

Шаг 3. Полученное решение оптимизационной подзадачи на шаге 2 принимается за следующую точку для аппроксимации.

Процесс прерывается, когда выполняются определенные критерии сходимости.

Далее необходимо понять, как следует выбирать аппроксимирующие функции $f_i^{(k)}$. Рассмотрим шаг k и обозначим такие коэффициенты $L_j^{(k)}$ и $U_j^{(k)}$:

$$L_j^{(k)} < x_j^{(k)} < U_j^{(k)} \quad (1.15)$$

Аппроксимирующие функции для $i = \overline{0, m}$ имеют вид:

$$f_i^{(k)}(\mathbf{x}) = r_i^{(k)} + \sum_{j=1}^n \left(\frac{p_{ij}^{(k)}}{U_j^{(k)} - x_j} + \frac{q_{ij}^{(k)}}{x_j - L_j^{(k)}} \right), \quad (1.16)$$

где

$$p_{ij}^{(k)} = \begin{cases} (U_j^{(k)} - x_j^{(k)})^2 \frac{\partial f_i}{\partial x_j}, & \text{если } \frac{\partial f_i}{\partial x_j} > 0 \\ 0, & \text{если } \frac{\partial f_i}{\partial x_j} \leq 0 \end{cases} \quad (1.17)$$

$$q_{ij}^{(k)} = \begin{cases} 0, & \text{если } \frac{\partial f_i}{\partial x_j} \geq 0 \\ -(x_j^{(k)} - L_j^{(k)})^2 \frac{\partial f_i}{\partial x_j}, & \text{если } \frac{\partial f_i}{\partial x_j} < 0 \end{cases} \quad (1.18)$$

$$r_i^{(k)} = f_i^{(k)}(\mathbf{x}^{(k)}) - \sum_{j=1}^n \left(\frac{p_{ij}^{(k)}}{U_j^{(k)} - x_j^{(k)}} + \frac{q_{ij}^{(k)}}{x_j^{(k)} - L_j^{(k)}} \right) \quad (1.19)$$

Эти функции совпадают с исходными функциями с точностью до линейного приближения.

В силу того, что $p_{ij}^{(k)} \geq 0, q_{ij}^{(k)} \geq 0$, можно показать, что $f_i^{(k)}(\mathbf{x})$ – выпуклые функции. Название метода связано с наличием коэффициентов $L_j^{(k)}$ и $U_j^{(k)}$, которые являются асимптотами гипербол. Кривизна аппроксимирующих функций увеличивается с уменьшением расстояния от точки $x_j^{(k)}$ до $L_j^{(k)}$ и $U_j^{(k)}$, так как увеличиваются значения $\frac{\partial^2 f_j}{\partial x_j^2}(\mathbf{x}^{(k)})$. Чем сильнее увеличивать расстояние между коэффициентами и $x_j^{(k)}$, тем более линейной будет становиться функция для тех значений x , которые более близки к $x_j^{(k)}$. При устремлении коэффициентов к бесконечности: $L_j^{(k)} \rightarrow -\infty$ и $U_j^{(k)} \rightarrow +\infty$, $f_i^{(k)}$ становятся полностью линейными функциями от \mathbf{x} :

$$f_i^{(k)}(\mathbf{x}) = f_i(\mathbf{x}^{(k)}) + \sum_{j=1}^n \frac{\partial f_i}{\partial x_j} (x_j - x_j^{(k)}) \quad (1.20)$$

Эти функции идентичны аппроксимирующим функциям, которые применяются в широко известном методе последовательного линейного программирования. Таким образом, необходимо правильно подобрать значения коэффициентов на каждой итерации. Существуют некоторые рекомендации по выбору значений коэффициентов исходя из поведения целевой функции. Например, при наблюдении осцилляций в поведении функции следует выбирать значения коэффициентов близкими к $x_j^{(k)}$, если же поведение монотонно, то наоборот.

На втором шаге решается оптимизационная подзадача, поскольку применение аппроксимирующих функций приводит к выпуклой задаче с разделяющимися переменными, можно ввести функционал Лагранжа для дальнейшего анализа:

$$l(\mathbf{x}, \mathbf{y}) = f_0^{(k)}(\mathbf{x}) + \sum_{i=1}^m y_i (f_i^{(k)}(\mathbf{x}) - b_i) = \quad (1.21)$$

Опуская подробные вычисления, получаем:

$$r_0 - \mathbf{y}^T \mathbf{b} + \sum_{j=1}^n \left(\frac{p_{0j} + \mathbf{y}^T \mathbf{p}_j}{U_j - x_j} + \frac{q_{0j} + \mathbf{y}^T \mathbf{q}_j}{x_j - L_j} \right) = r_0 - \mathbf{y}^T \mathbf{b} + \sum_{j=1}^n l_j(x_j, \mathbf{y}) \quad (1.22)$$

где y_i – множители Лагранжа, $\mathbf{b} = \{b_1, \dots, b_m\}^T$, $b_i = \hat{f}_i - r_i$.

Путем взятия производных $l_j(x_j, \mathbf{y})$ по переменным проектирования и анализа знаков этих производных, получаем уникальное решение, при котором выполняется равенство $l'_{x_j}(\mathbf{x}, \mathbf{y}) = 0$.

$$x_j(\mathbf{y}) = \frac{(p_{0j} + \mathbf{y}^T \mathbf{p}_j)^{1/2} L_j + (q_{0j} + \mathbf{y}^T \mathbf{q}_j)^{1/2} U_j}{(p_{0j} + \mathbf{y}^T \mathbf{p}_j)^{1/2} + (q_{0j} + \mathbf{y}^T \mathbf{q}_j)^{1/2}} \quad (1.23)$$

Получаем двойственную задачу на поиск максимума функционала $W(\mathbf{y})$ после подстановки $l(\mathbf{x}, \mathbf{y})$ в выражение.

$$\begin{aligned} & \max_{\mathbf{y} \geq \mathbf{0}} W(\mathbf{y}) \\ & W(\mathbf{y}) = \min_x \{l(\mathbf{x}, \mathbf{y}); L_j < \alpha_j \leq x_j \leq \beta_j < U_j\} \end{aligned} \quad (1.24)$$

$W(\mathbf{y})$ является гладкой и вогнутой функцией, таким образом, данная подзадача успешно разрешается с помощью произвольного градиентного метода.

ГЛАВА 2. НЕЙРОННЫЕ СЕТИ

2.1. Обзор методов машинного обучения в задаче топологической оптимизации

В настоящее время много исследователей обращаются к использованию методов машинного обучения в задаче топологической оптимизации. Это обусловлено возможностью применения современных мощных алгоритмов обучения, которые позволяют находить оптимальные распределения материала с высокой эффективностью.

Использование методов машинного обучения в топологической оптимизации открывает новые возможности для решения сложных задач. Одним из таких примеров является поиск оптимального распределения нескольких материалов с различными свойствами. Также методы машинного обучения позволяют эффективно находить оптимальную топологию материала с учетом нелинейных свойств, таких как пластичность или вязкость. Кроме того, методы машинного обучения могут быть применены для оптимизации топологии материала с учетом динамических нагрузок и вибраций. Это позволяет создавать более прочные и устойчивые конструкции, способные справляться с динамическими воздействиями, такими как колебания или удары.

В топологической оптимизации часто существуют различные ограничения и условия, которые должны быть учтены. Например, может быть задана минимальная или максимальная толщина структуры, ограничения на объем или массу материала, а также напряжения. Методы машинного обучения позволяют интегрировать эти ограничения в процесс оптимизации и находить оптимальные решения, учитывающие все требования и условия. Таким образом, применение методов машинного обучения в топологической оптимизации позволяет решать сложные задачи, учитывать различные свойства материала и технологические требования, а также сокращать время и ресурсы, затрачиваемые на оптимизацию процесса проектирования структур.

Литературный обзор источников позволяет выделить основные цели применения существующих методов машинного обучения в задаче топологической оптимизации: ускорение итерационной процедуры, безытерационная оптимизация, мета-моделирование, уменьшение размерности пространства проектирования, усовершенствование оптимизатора, генеративный дизайн и постобработка.

Далее в данной главе будет представлен подробный обзор каждого из этих аспектов. Также будут рассмотрены различные подходы и методы машинного обучения, применяемые для достижения поставленных целей в топологической оптимизации. Кроме того, будут обсуждаться исследования, которые демонстрируют эффективность и преимущества применения методов машинного обучения в конкретных задачах топологической оптимизации. В результате этого обзора будет предоставлена обширная картина текущего состояния и будущих перспектив применения методов машинного обучения в области топологической оптимизации.

2.1.1. Ускорение итеративного процесса с помощью методов машинного обучения

В топологической оптимизации, как и в других классических итерационных методах с увеличением итераций изменение переменных проектирования начинает происходить медленнее. По мере сходимости метода на последних итерациях отклонение переменных проектирования может быть пренебрежительно мало, а время, затрачиваемое на этом этапе, велико. Поэтому часть исследований посвящены использованию нейронных сетей для ускорения итерационной процедуры с помощью предсказания оптимального решения на основе промежуточных, еще не полностью сходящихся решений.

Согласно далее рассмотренным исследованиям данный подход позволяет ускорить трудоемкий процесс топологической оптимизации. К примеру, в работе [22] авторы предложили использовать сверточные нейронные сети для достижения требуемого эффекта.

Для достижения лучшего результата ученые предложили прогнозировать результат по серии изображений, которые позволяют отследить характерные перемещения в процессе оптимизации. Например, в работе [28] с помощью сверточной LSTM (Long short-term memory) нейронной сети удалось спрогнозировать оптимальное распределение материала для двумерных структур, основываясь на начальной истории распределения плотности на выбранных итерациях. Также другими учеными была предложена нейронная сеть Pix2pix для создания отображения между промежуточной и конечной конфигурацией.

Кроме того, в работе [11] исследователи представили глубокую сеть доверия DBN (Deep belief network), которая используется совместно с SIMP методом. Промежуточный результат, полученный с помощью SIMP метода, подается на вход нейронной сети для прогнозирования оптимальной структуры, затем повторно уточняется с помощью SIMP метода. В последующей работе эти же авторы усовершенствовали данный метод с целью обеспечить его масштабируемость. Это означает, что метод может быть применен к задачам различного масштаба, независимо от их размеров или сложности.

В работе [18] была предложена гибридная структурная оптимизация, инженеры связали два процесса структурной оптимизации: оптимизацию топологии и параметрическую оптимизацию формы. Кроме того, они разработали коммуникационную модель на основе машинного обучения, которая может обмениваться геометрической информацией между модулями оптимизации топологии и формы для получения оптимальных результатов с точки зрения технологических ограничений.

2.1.2. Упрощение поиска производных с помощью методов машинного обучения

В задаче топологической оптимизации инженеры стремятся исследовать новые подходы для замены метода конечных элементов с использованием нейронных сетей. Такие подходы позволяют сократить время на поиск глобального минимума целевой функции с учетом определенных ограничений. Во многих

исследованиях было выдвинуто предложение по замене метода конечных элементов сверточной нейронной сетью. В работе [14] представили новую структуру, в которой репрезентативный объемный элемент и модель машинного обучения используются для построения непрерывных анизотропных эффективных свойств материала для одновременного проектирования общей конфигурации топологии и компоновки волоконного материала в функционально градуированных композитных структурах. В работе [27] расчет эффективной упругой жесткости метаматериала заменен нейронной сетью, осуществляющей прогноз в зависимости от четырех геометрических параметров элементарной ячейки. Некоторые авторы предложили схему онлайн-обучения, основное преимущество в том, что предварительная генерация обучающих данных не требуется. Уравнения равновесия решаются для модели с грубым конечно-элементным разбиением. Полученные поля перемещений и деформаций передаются вместе с переменной жесткостью на вход нейронной сети для прогнозирования производных на модели с мелким конечно-элементным разбиением. В работе [25] с помощью сверточной нейронной сети по имеющемуся распределению плотности производится анализ на чувствительность. В работе [2] обученная нейронная сеть, включающая в себя сверточную нейронную сеть с U-Net архитектурой и рекуррентную нейронную сеть с long-short term memory (LSTM), итеративно удаляет материал из сплошной среды без использования МКЭ.

2.1.3. Уменьшение пространства переменных проектирования с помощью методов машинного обучения

При увеличении размерности модели и ее конечно-элементного разбиения, время расчета значительно увеличивается из-за увеличения числа переменных проектирования. Большое количество переменных приводит к более сложным итерационным процедурам, требующим больше вычислительных ресурсов. Это может существенно ограничить эффективность и применимость классических методов оптимизации. Однако, с использованием нейронных сетей и других методов машинного обучения, возможно ускорить процесс оптимизации и снизить

размерность пространства проектирования, что ведет к более эффективным и быстрым результатам.

В статье [8] параметризуют задачу топологической оптимизации с помощью нейронной сети. Авторы предложили решать задачи оптимизации с помощью генерирования кандидатного решения нейронной сетью и последующего эволюционирования решения методами градиентного спуска по целевой функции податливости. Податливость получаемой конструкции оценивается с помощью дифференцируемой физической модели, что, собственно, и позволяет использовать градиентный спуск. В дальнейшем работе авторы применили подход параметризации к трехмерной области. Тестовые примеры иллюстрируют способность создавать более тонкие структуры. Аналогичным образом авторы работы [5] параметризуют плотность с помощью весов нейронной сети для того, чтобы плотность не зависела от конечных элементов. Позднее авторы распространили данный подход на задачу оптимального распределения двух и более материалов. В работе была добавлена возможность управлением толщины, с помощью проекции пространства Фурье.

Авторы в работе [8] представили двухэтапный метод глубокого онлайн-обучения, основанного на модели PED (predictor–evaluator network). Модель «Predictor» предсказывает оптимизированные топологии, а модель «evaluator» оценивает их во время обучения, вычисляя целевую функцию – податливость и минимизируя ее путем варьирования параметров нейронной сети «predictor». В дальнейших работах к подходу параметризации задачи добавляется нейронную сеть, предсказывающая поле перемещений, и затем прогнозируется оптимальное распределение материала. В работе [26] предлагают новый метод, который использует расширенный вариационный автоэнкодер для преобразования двумерных топологий в скрытое пространство меньшего размера. Сама топологическая оптимизация выполняется в скрытом пространстве, а не в пространстве изображений. В другой работе используют генеративно-состязательную нейронную сеть, которая обеспечивает удовлетворение

технологическим ограничениям, путем проведения топологической оптимизации в скрытом пространстве.

Другим возможным способом уменьшения количества переменных проектирования является отказ от параметризации с помощью конечно-элементного разбиения, и использование геометрических параметров в качестве проектных переменных. Данные методы имеют меньше проектных переменных, чем метод, основанный на плотности, и демонстрируют высокую вычислительную эффективность. Авторы статьи [16] предложили подход на основе Moving Morphable Component (MMC) совместно с машинным обучением. Для сокращения пространства параметров и обучающих данных используется метод главных компонент. С помощью данного метода получают основные компоненты переменных MMC и проводят нелинейную регрессию (k-ближайших соседей и метод опорных векторов SVM) для получения новой топологии. Авторы пытаются добиться оптимальную топологию почти мгновенно при изменении внешних воздействий. В статье [10] (2020) также использует подход на основе MMC, совмещенный с классификатором изображений на основе деревьев для оптимизации и объединенный с алгоритмом роя частиц. Классификатор позволяет определить является ли оптимизированное решение допустимым или нет. Это гарантирует, что неподходящие геометрические параметры не будут рассмотрены.

2.1.4. Улучшение качества изображений в задаче топологической оптимизации с помощью методов машинного обучения

Оптимизированные структуры зачастую имеют низкое качество, поэтому цель многих исследований стоит в том, чтобы повысить его. Предлагаемые подходы обеспечивают эффективное удаление шумов изображения, повышение детализации и контрастности, а также значительное улучшение границ структуры. С помощью нейронных сетей на этапе постобработки результатов оптимизации достигается превосходное качество оптимизированных структур. Нейронные сети способны выявить и корректировать недостатки и несовершенства, создавая изначально эстетически привлекательные и функционально эффективные результаты

топологической оптимизации. Это приводит к более точным и оптимальным решениям, значительно расширяя возможности и потенциал данного подхода в инженерных и научных приложениях.

Например, в работе [19] получают оптимальную топологическую структуру на грубом конечно-элементном разбиении. И затем при помощи нейронной сети с глубоким сверхвысоким разрешением или сверточной нейронной сети сверхвысокого разрешения (SRCNN) повышают качество изображения как в двухмерных, так и в трехмерных задачах.

В работе [23] использовали метод опорных векторов для достижения более гладкой и четкой границы путем классификации внешней части оптимизированной структуры на области с материалом или без. В работе [13] представили подход машинного обучения для создания оптимальной топологии в высоком качестве, учитывая обучающие данные с разными разрешениями. В статье представлено совместное использование дизъюнктивной модели нормальной формы и глубокой нейронной сети.

2.1.5. Использование нейронных сетей для достижения безытерационной оптимизации

Наконец, последняя проблема, которой уделяют большое внимание ученые, это полностью безытерационная оптимизация. Данный подход позволяет условиям проектирования, таким как граничные условия и нагрузки, сопоставить оптимальную топологию конструкции с помощью нейронной сети без проведения оптимизации.

В статье [1] представили сверточную нейронную сеть, которая помимо нагрузок и граничных условий принимает ограничение на объем в виде матрицы и прогнозирует оптимальное распределение материала с нелинейными эффектами. Для аналогичных входных параметров в [21] получили оптимальную топологию, используя генеративно-состязательную сеть. Авторы [15] разработали сверточную нейронную сеть предсказывающую оптимизированную конструкцию метаматериала для различной объемной доли и радиуса фильтрации. В некоторых

работах была предложена глубокая сверточная нейронная сеть, на вход которой передавали не только объемную долю, нагрузки и граничные условия, но также информацию о поле деформаций для обобщения этого подхода на различные граничные условия. В работе [20] предложили использовать условно генеративно-состязательную сеть (сGAN-type), в которой наряду с объемной долей, перемещениями, нагрузкой и граничными условиями используют плотность энергии деформации, напряжения фон Мизеса, полученные из МКЭ. Другие авторы предлагали работе на вход сверточной нейронной сети дополнительно подавать матрицу главных напряжений. Также сверточная нейронная сеть способна прогнозировать оптимальную структуру по полям перемещений, деформаций и объемной доле. В [2] предложили GAN с двумя дискриминаторами, способными учитывать не только механические ограничения, но и геометрические, а именно, – количество стержней. В работе [4] используется обучение с подкреплением, где последовательно происходит удаление элемента из структур, и модель получает вознаграждение за минимизацию податливости. В статье [4] использовали CNN и получали оптимальные структуры для различных значений коэффициентов Пуассона и объемной доли.

Для достижения прогноза результата с улучшенными инженерными характеристиками был разработан многоэтапный безытерационный процесс. Первый этап направлен на прогнозирование оптимальной топологии с учетом заданных нагрузок и ограничений. Этот этап позволяет получить начальное представление о желаемой структуре. Второй этап фокусируется на повышении качества изображения полученной топологии либо на уточнение требуемых характеристик.

В работе [24] была применена сверточная нейронная сеть для прогнозирования оптимальной структуры, основываясь на информации о силах, объемной доле и полях напряжений фон-Мизеса. Далее, с использованием условной генеративно-состязательной нейронной сети был проведен процесс уточнения, направленный на выявление областей, подверженных разрушению. Этот подход позволяет более точно прогнозировать места потенциального разрушения в оптимизированной

структуре, что имеет важное значение для обеспечения безопасности и надежности конструкций. Другими исследователями был предложен подход для задачи кондуктивного теплообмена. В данном подходе изображения топологий, близких к оптимальным, прогнозируются с использованием генеративной состязательной сети, а затем эти структуры уточняются с помощью генеративно-состязательной сети сверхразрешения (SRG) для получения высококачественных результатов. В работе [7] предложили двухэтапную модель с использованием нейронных сетей энкодер-декодер, включающую новую функции ошибки, чтобы уменьшить количество случаев структурного разъединения. В статье [17] авторы используют трансферное обучение исходной сверточной нейронной сети на данных с низким разрешением и гораздо меньший набор данных с высоким разрешением для точной настройки целевой сети. Целевая нейронная сеть получается из исходной путем добавления новых сверточных слоев, таким образом информация, полученная исходной нейронной сетью, переносятся на целевую для дальнейшей ее настройки. В работе [6] предложили использовать трехступенчатую модель. Сначала глубокая нейронная сеть получает оптимизированную структуру, затем с помощью сверточной нейронной сети уточняет ее границы, а далее алгоритм топологической оптимизации за 5 итераций обеспечивает ограниченность по объему.

Из проведенного литературного обзора следует, что существует большое количество исследований, посвященных данной теме, что подтверждает популярность и активное развитие данной области. В настоящей работе предлагается подход по использованию нейронной сети именно для безытерационной топологической оптимизации. В следующей главе будет подробно описана двухэтапная модель, объединяющая в себе сверточную нейронную сеть и генеративно-состязательную нейронную сеть. Для получения оптимальной топологии конструкции с помощью предложенной модели требуется лишь знать граничные условия, нагрузки и ограничение на объем.

2.2. Математическая модель нейрона

Линейный перцептрон, описанный Фрэнком Розенблаттом, является одной из первых моделей нейронных сетей. Он представляет собой простую математическую модель нейрона, которая используется для бинарной классификации. Все объекты в тренировочной выборке имеют 2 метки: 0 или 1. Задачей такого перцептрона является расставлять метки о новых, ранее не виденных объектах.

Рассмотрим математическую модель нейрона, введем следующие обозначения. Входные сигналы: $x = (x_1, \dots, x_d)^T \in \mathbb{R}^d$, синаптические веса: $\omega = (\omega_1, \dots, \omega_d)^T \in \mathbb{R}^d$, пороговое значение: $b \in \mathbb{R}$.

Модель, приведённую на Рис.2.1, предложили в 1943 г. МакКаллох и Питтс.

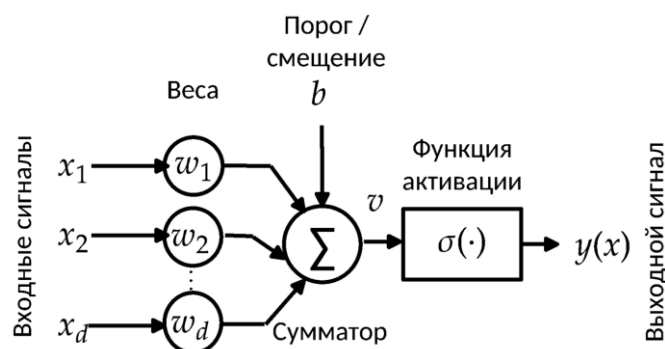


Рис.2.1. Модель перцептрона

Выход перцептрона имеет следующий вид:

$$y(x) = \sigma(\langle x, w \rangle + b) = \sigma\left(\sum_{j=1}^d \omega_j x_j + b\right) \quad (2.1)$$

Здесь $\sigma(x)$ имеет вид:

$$\sigma(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (2.2)$$

Вообще в качестве функции активации может быть выбрана любая другая нелинейная функция, о которой будет рассказано дальше. В таком случае, данный перцептрон можно будет назвать элементарной единицей нейронной сети.

2.3. Метод обратного распространения ошибки

Архитектура полносвязной нейронной сети включает в себя входной, выходной и промежуточные слои. Прямая связь между нейронами внутри одного слоя отсутствует, сигнал нейрона текущего слоя передается на вход нейрону следующего слоя. Для обучения такого вида нейронных сетей используется метод обратного распространения ошибки. В данном методе используется градиентный спуск для обновления весов сети на основе разности между выходом сети и ожидаемым значением. Процесс обучения начинается с случайной инициализации весов нейронной сети. Затем для каждого обучающего примера происходит прямой проход, при котором сигнал проходит через слои сети, и вычисляется ошибка. После этого происходит обратное распространение ошибки, где ошибка распространяется назад по сети, и для каждого нейрона вычисляется его вклад в эту ошибку. На основе этих вычислений обновляются веса сети с помощью градиентного спуска. Градиент показывает направление, в котором нужно изменить веса, чтобы уменьшить ошибку. Таким образом, производится корректировка весов сети, и процесс прямого прохода и обратного распространения ошибки повторяется множество раз, пока будет не достигнута желаемая точность.

Рассмотрим нейронную сеть, содержащую L слоев. Нейрон порядковым номер j принадлежит слою $L - 1$, нейрон i -ый слою $L - 2$.

Пусть нейрон с порядковым номер k находится в выходном слое L , нейрон с порядковым номер j находится в слое, нейрон с порядковым номером находится в слое.

Пусть в каждом слое находится фиксированное число нейронов: $m_L, m_{L-1}, m_{L-2}, \dots, m_1$. Для определенности присвоим итерации номер n , внутри одной итерации происходит прямой и обратный проход. Пусть w_{ji} обозначает весовой коэффициент, связывающий выход нейрона i с входом нейрона j .

Мерой эффективности обучения нейронной сети является функция потерь, которая может быть выбрана по усмотрению разработчика. Здесь в качестве примера будет приведена среднеквадратичная суммарная ошибка:

$$L_2 = \frac{1}{2} \sum_{i=1}^m (o_k - t_k)^2 \quad (2.3)$$

где t_k – ожидаемый выход нейрона, o_k – выход, генерируемый нейроном на итерации n .

Вектор градиента суммарной ошибки представляет собой набор частных производных функции ошибки по каждому весовому коэффициенту.

Знак каждой составляющей градиента функции ошибки указывает, в каком направлении нужно изменить соответствующий весовой коэффициент, чтобы эффективно снизить ошибку. Абсолютное значение каждой составляющей градиента показывает, насколько сильно нужно изменить соответствующий весовой коэффициент, чтобы достичь наибольшего улучшения в снижении ошибки.

Найдем частную производную целевой функции по синаптическим весам как производную сложной функции:

$$\frac{\partial L_2}{\partial w_{kj}} = \frac{\partial L_2}{\partial o_k} \frac{\partial o_k}{\partial s_k} \frac{\partial s_k}{\partial w_{kj}} \quad (2.4)$$

Во время прямого прохода сигнал, подающийся на вход нейронной сети, проходит через все слои сети к выходному слою. По мере продвижения сигнала каждый нейрон вычисляет взвешенную линейную комбинацию и применяет к ней функцию активации φ_k .

$$o_k = \varphi_k(s_k) \quad (2.5)$$

$$s_k = \sum_{j=0}^{m_{L-1}} w_{kj} o_j \quad (2.6)$$

Подставляя (2.3), (2.5), (2.6) в (2.4) и дифференцируя, получаем соотношения для частой производной функции ошибки:

$$\frac{\partial L_2}{\partial w_{kj}} = -(t_k - o_k) o_k \varphi'(s_k) \quad (2.7)$$

Для корректировки синаптических весов применяется дельта-правило, которое позволяет вычислить обновленные значения весов по следующей формуле:

$$w_{kj}(n + 1) = w_{kj}(n) + \Delta w_{kj}(n + 1) \quad (2.8)$$

$$\Delta w_{kj} = -r \frac{\partial L_2}{\partial w_{kj}}, \quad (2.9)$$

где λ – параметр скорости обучения.

Зная частную производную функции потерь, можно вычислить изменение весового коэффициента:

$$\Delta w_{kj} = r \delta_k o_k, \quad (2.10)$$

Формула для вычисления локального градиента δ_k имеет следующий вид:

$$\delta_k = -\frac{\partial L_2}{\partial s_k} = (t_k - o_k) \varphi'(s_k) \quad (2.11)$$

Согласно формуле (2.11), локальный градиент выходного нейрона k определяет необходимое изменение синаптического веса.

Корректировка веса нейрона выходного слоя и вычисление его локального градиента не вызывают особых трудностей. Однако, для нейрона, находящегося в скрытом слое, ситуация усложняется тем, что мы не знаем желаемого отклика этого нейрона. Но благодаря методу обратного распространения ошибки мы можем последовательно находить локальный градиент для нейрона скрытого слоя, используя информацию о локальных градиентах последующих слоев.

Запишем локальный градиент j нейрона скрытого слоя:

$$\delta_j = -\frac{\partial L_2}{\partial s_j} = -\frac{\partial L_2}{\partial o_j} \varphi'(s_j) \quad (2.12)$$

Вычислим частную производную:

$$\frac{\partial L_2}{\partial o_j} = \sum_{k=1}^{m_L} e_k \frac{\partial e_k}{\partial o_j}, \quad (2.13)$$

где $e_k = t_k - o_k$.

По правилу дифференцирования сложной функции, находим частную производную функции потерь:

$$\frac{\partial L_2}{\partial o_j} = \sum_{k=1}^{m_L} e_k \frac{\partial e_k}{\partial s_k} \frac{\partial s_k}{\partial o_j} \quad (2.14)$$

$$\frac{\partial e_k}{\partial s_k} = \frac{\partial (t_k - o_k)}{\partial s_k} = -\varphi'_k(s_k) \quad (2.15)$$

Поскольку взвешенная сумма выходных сигналов скрытого слоя является входом для выходного слоя, запишем:

$$s_k = \sum_{j=0}^{m_{L-1}} w_{kj} o_j \quad (2.16)$$

Подставляя в соотношение (2.14), получаем выражение:

$$\frac{\partial L_2}{\partial o_j} = - \sum_{k=1}^{m_L} e_k \varphi'_k(s_k) w_{kj} = - \sum_{k=1}^{m_L} \delta_k w_{kj} \quad (2.17)$$

Локальный градиент нейрона скрытого слоя принимает вид:

$$\delta_j = \sum_{k=1}^{m_L} \delta_k w_{kj} \varphi'(s_j) \quad (2.18)$$

Более подробно рассмотрим полученные формулы для локальных градиентов. В случае нейрона выходного слоя, для вычисления локального градиента необходимо знать желаемые значения и выходной сигнал нейронной сети. А в случае нейрона скрытого слоя, локальный градиент зависит от взвешенной суммы градиентов, вычисленных для нейронов следующего слоя. В обоих случаях

используется первая производная функции активации, поэтому важно, чтобы функция активации была дифференцируемой. Таким образом, обратное распространение ошибки выполняется итеративно, вычисляя градиенты нейронов от слоя к слою. Параметр r влияет на количество итераций, необходимых для завершения обучения. Рекомендованное значение r лежит в диапазоне от 0.1 до 0.001, и оптимальное значение определяется опытным путем.

При использовании низкого значения скорости обучения, алгоритм движется медленно, и траектория точки, двигающейся в противоположном направлении градиента, имеет гладкую форму.

Если значение скорости обучения высоко, то алгоритм сходится быстрее, но траектория движения принимает зигзагообразный вид.

Однако, при превышении определенного критического значения, алгоритм становится нестабильным.

2.4. Функции активации

Как ранее было сказано, функция активации — это математическая функция, которая применяется к выходу нейрона в нейронной сети. Функция активации вводит нелинейность в нейронную сеть, позволяя ей обрабатывать сложные и нелинейные зависимости в данных. Различные функции активации имеют разные свойства и характеристики, и выбор конкретной функции зависит от задачи и требований модели. В качестве примера ниже приведены некоторые функции активации.

Сигмоидная функция принимает любое вещественное число и возвращает значение в интервале от 0 до 1. Данная функция может быть полезная для решения задач бинарной классификации, оценки вероятностей наличия определенного события. Также функция имеет довольно гладкую производную, что позволяет использовать градиентный спуск во время обучения. Однако, следует отметить, что сигмоидная функция имеет свои недостатки, такие как насыщение градиента при очень больших или очень маленьких значениях входа. В таких случаях, другие

активационные функции, такие как ReLU или гиперболический тангенс, могут быть предпочтительнее.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.19)$$

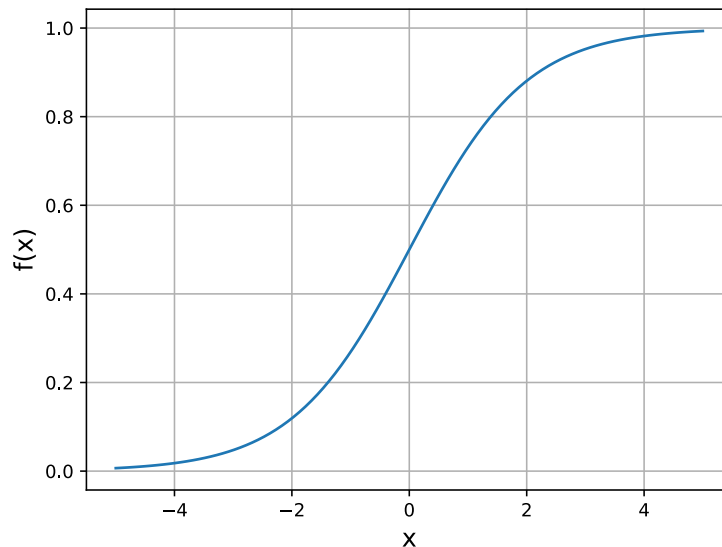


Рис.2.2. График сигмоидной функции

ReLU функция возвращает входное значение, если оно положительное, и ноль в противном случае. Такая функция значительно ускоряет процесс обучения нейронной сети и борется с проблемой «затухающий градиентов», так как ее градиент постоянен и не зависит от входа. Однако проблемой данной функции является недифференцируемость в нуле.

$$f(x) = \begin{cases} x, & \text{если } x > 0 \\ 0, & \text{если } x \leq 0 \end{cases} \quad (2.20)$$

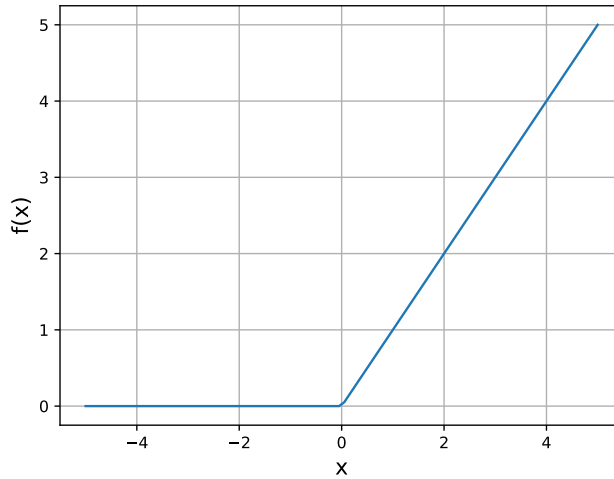


Рис.2.3. График функции ReLU

$$f(x) = \frac{e^{x_i}}{\sum_{j=1}^K e^{x_j}} \quad (2.21)$$

Softmax функция используется в нейронной сети для решения задач классификации, где требуется предсказать вероятности принадлежности входных данных к различным классам. Функция является гладкой и дифференцируемой, что облегчает обратное распространение ошибки в процессе обучения нейронных сетей. Softmax функция также имеет свои ограничения, такие как чувствительность к выбросам и потерю информации о относительных масштабах входных данных.

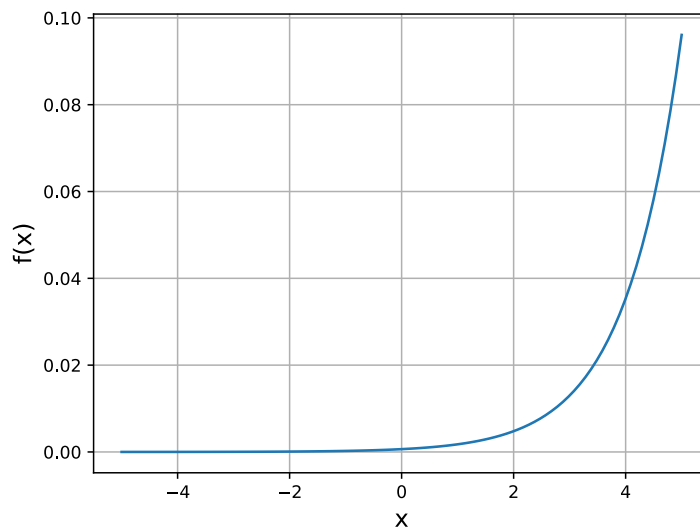


Рис.2.4. График функции Softmax

Тангенс гиперболический является одной из популярных функций активации в нейронных сетях. Функция имеет диапазон значений от -1 до 1, что делает ее более удобной для нормализации данных, в отличие от сигмоидной функции. Это особенно полезно в случае, когда данные имеют сильное отклонение или разброс. Также функция симметрична относительно нуля, что означает, что ее значения будут равны как положительным, так и отрицательным входам. Это свойство может быть полезным при моделировании данных, где существуют симметричные отношения или взаимосвязи между входами. Она широко используется в нейронных сетях для различных задач и может быть особенно полезна, когда требуется моделирование симметричных данных или нормализация входных значений.

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.22)$$

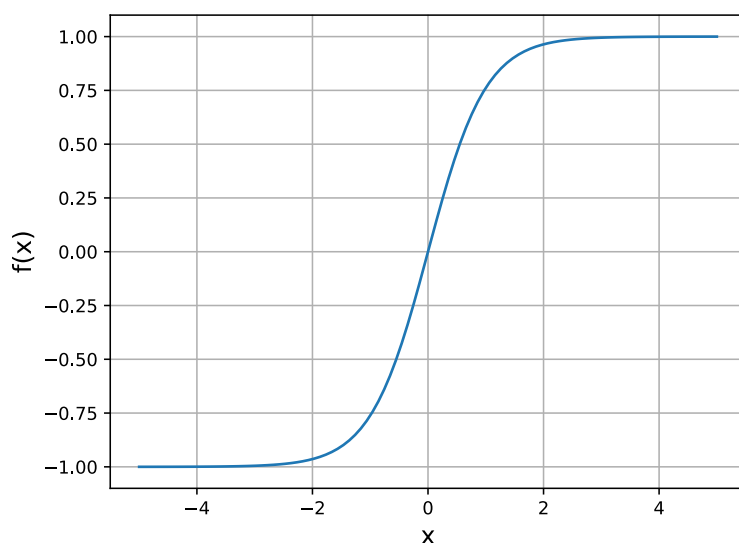


Рис.2.5. График функции Tanh

Каждая функция активации в нейронных сетях имеет свои преимущества и недостатки, и выбор конкретной функции зависит от специфики задачи. В целом, выбор функции активации осуществляется исходя из требований задачи, а также экспериментальным путем для достижения лучшей производительности и результатов обучения.

ГЛАВА 3. РАЗРАБОТКА ДВУХЭТАПНОЙ МОДЕЛИ ДЛЯ ПРЕДСКАЗАНИЯ ОПТИМАЛЬНОЙ ТОПОЛОГИИ УПРУГОЙ ПЛАСТИНЫ

3.1. Сверточная нейронная сеть

Идея разработки сверточных нейронных сетей была вдохновлена изучением зрительной коры головного мозга. Зрительная кора имеет несколько зон, каждая из которых специализирована на обработке определенных признаков поступающей информации. Эти зоны взаимодействуют друг с другом для создания полного восприятия. Кроме того, нейроны в данных зонах могут непоследовательно сообщаться между собой. В сверточных нейронных сетях отражаются все эти особенности, позволяя эффективно обрабатывать и распознавать визуальные данные.

В 1988 году Ян ЛеКун провел революционные исследования в области сверточных нейронных сетей и предложил концепцию "сверточного слоя" в нейронных сетях. Он разработал первые алгоритмы обратного распространения ошибки для сверточных нейронных сетей и продемонстрировал их успешность в задачах распознавания символов.

Затем в 1990-х годах Ян ЛеКун и его коллеги разработали и усовершенствовали модель сверточных нейронных сетей, известную как LeNet, которая стала одной из первых успешно примененных моделей для распознавания текста.

Однако, на протяжении длительного времени сверточные нейронные сети не получали широкого распространения из-за ограничений в вычислительных ресурсах и доступных данных для обучения. Это изменилось в 2012 году, когда модель под названием AlexNet, разработанная командой исследователей из Университета Торонто, победила в соревновании ImageNet Computer Vision. Модель AlexNet, названная в честь Алекса Крижевского, продемонстрировала успешные результаты сверточных нейронных сетей в области классификации изображений и привлекла широкое внимание к этой разработке.

С тех пор сверточные нейронные сети стали основным инструментом в области компьютерного зрения. Было предложено множество моделей, включая VGG, ResNet, U-Net и другие, которые превзошли предыдущие достижения в области классификации, сегментации и распознавания объектов на изображениях.

Основная идея сверточных нейронных сетей состоит в обработке участков изображения фильтрами с целью извлечения локальных признаков. Вместо обработки всего изображения, сверточные нейронные сети фокусируются на анализе малых участков, которые затем объединяются для формирования более высокоуровневых признаков. Это позволяет сетям автоматически выделять различные характеристики, такие как границы, текстуры, формы и другие локальные детали изображения. Такой подход к обработке данных снижает количество параметров и вычислительную сложность модели.

На практике, изображения имеют индивидуальную структуру и представляет собой массив векторов чисел. Если изображение черно-белое, то это массив интенсивностей, если цветное, то массив векторов состоит из трех чисел, которые означают интенсивности красного, зеленого и синего цветов. Таким образом, изображение представимо в виде нескольких прямоугольных матриц, каждая из которых характеризует интенсивность одного из цветовых каналов. А размер матрицы зависит от количества пикселей. То есть каждому пикселу соответствует вектор чисел, и его компоненты называются каналами.

После применения сверточного слоя будут также получаться матрицы с пространственной структурой, которые называются картами признаков. Каждая карта признаков представляет собой матрицу значений, где каждое значение соответствует выделенному признаку из окна исходного изображения. Операция свертки является линейным преобразованием вида:

$$y_{i,j}^l = \sum_{-d \leq a, b \leq d} W_{a,b} x_{i+a, j+b}^l \quad (3.1)$$

где $y_{i,j}^l$ — результат свертки на уровне l , $x_{i,j}^l$ — входные данные, $W_{a,b}$ — матрица весов размером $(2d + 1) \times (2d + 1)$.

После сверточной операции необходимо применить нелинейную функцию активации, она применяется к полученному тензору поэлементно. Помимо этого, еще существует операция субдискретизация (pooling). Данная операция применяется для обобщения выделяемых признаков для сокращения размерности тензоров по ширине и высоте. Исследования, которые проводились для сравнения различных операций субдискретизации, обычно оказывались в пользу max-pooling.

$$x_{i,j}^{l+1} = \max_{-d \leq a, b \leq d} z_{i+a, j+b}^l \quad (3.2)$$

где $z_{i+a, j+b}^l$ – выход сверточного слоя после применения нелинейной функции, d – размер окна субдискретизации.

На рисунке приведена операция субдискретизации MaxPool с фильтром размером 2x2 и шагом 2, поэтому размеры изображения уменьшаются в 2 раза.

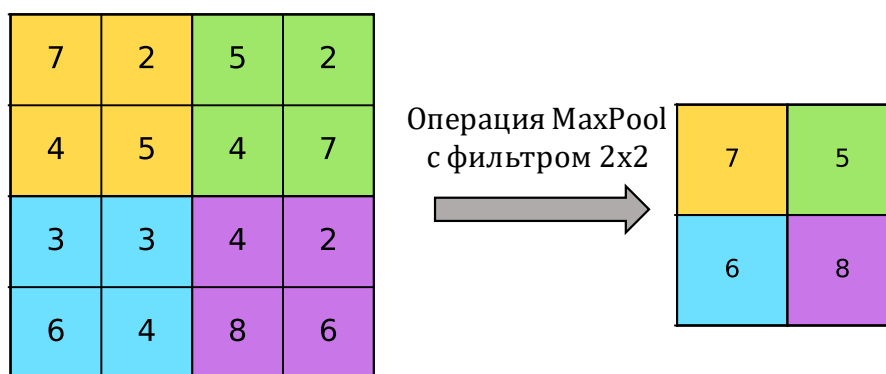


Рис.3.1. Пример субдискретизации

Также стоит отметить, что операции, отвечающие за понижение или повышение дискретизации выполняются без участия обучаемых параметров нейронной сети.

Размер фильтра обычно выбирают нечетным, так как он позволяет выделить один центральный пиксель, что обеспечивает однозначность операции свертки. В противном случае, могут потребоваться дополнительные шаги для обработки центрального пикселя.

На Рис.3.2 представлена архитектура полносверточной нейронной сети энкодер-декодер с понижающей и повышающей дискретизацией. Здесь

используется комбинация операции свертки слоев совместно субдискретизацией Max-Pooling в первой половине, и второй половине операция свертки комбинируется с интерполирующей операцией Upsampling для восстановления размеров входного изображения. За каждым двумя последовательными сверточными слоями следует операция дискретизации, которая уменьшает или увеличивает размер на $\frac{1}{2}$ от первоначального. Каждая операция свертки выполняется с фильтром размером 3×3 , за исключением первого слоя, где размер фильтра взят 2×2 , чтобы достичь четной размерности изображения. Также после каждой операции свертки происходит попиксельное применение ReLU функции. В последнем слое применяется сигмоидная функция для приведения выходных значений в диапазон от 0 до 1. Такие выходы можно интерпретировать как вероятности каждого пикселя в черно-белом изображении, где 0 означает белый пиксель, черный – 1.

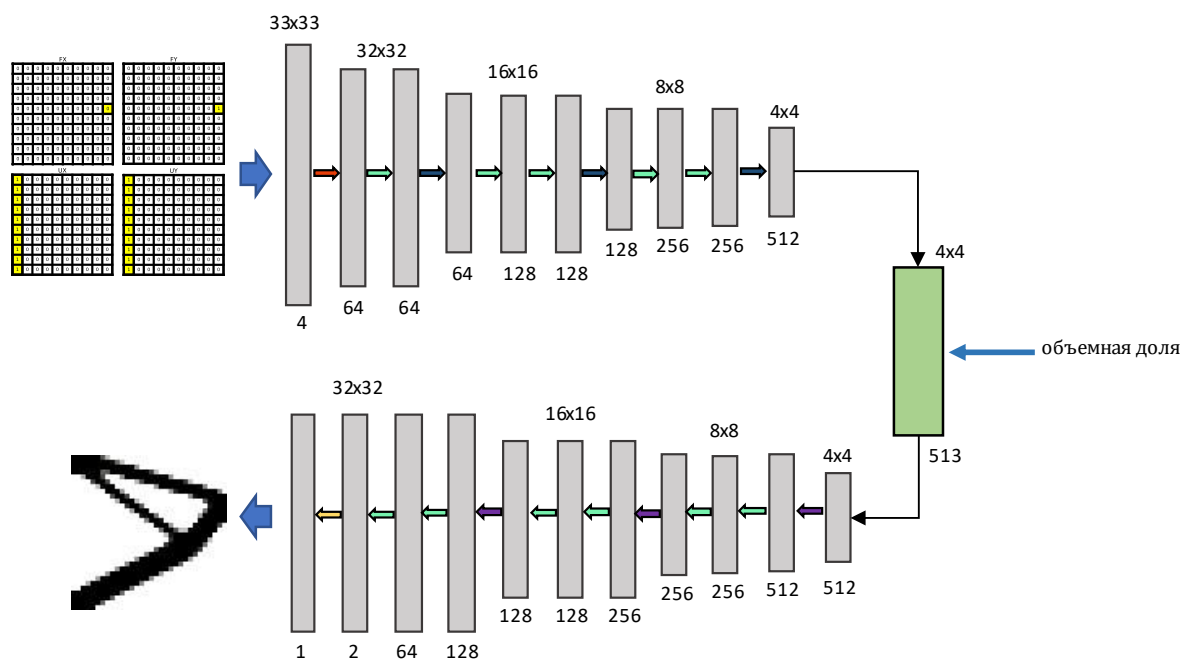


Рис.3.2. Архитектура сверточной нейронной сети

На Рис.3.3 отдельно выписаны названия слоев с размерами фильтров и значениями шагов для сверточной нейронной сети энкодер-декодер. Значение шага опущено и принято по умолчанию равным 1. Субдискретизация и повышающая дискретизация (интерполяция) имеет фильтр размером 2×2 .

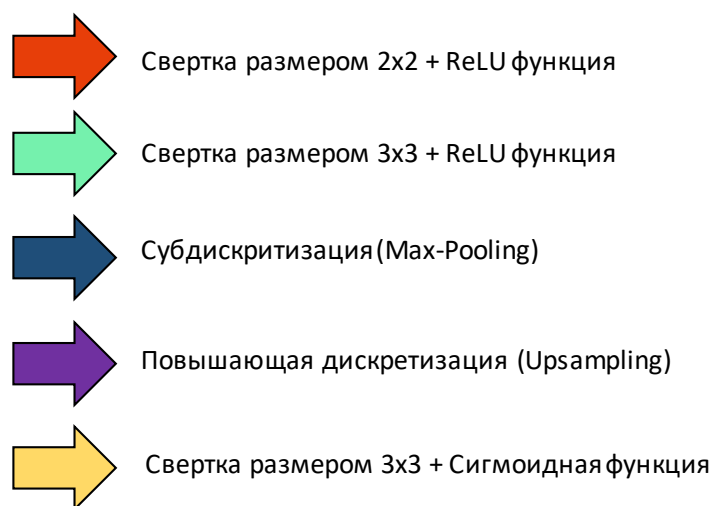


Рис.3.3. Слои сверточной нейронной сети

3.2. Генеративно-состязательная нейронная сеть

Генеративно-состязательная сеть (Generative adversarial network) была представлена в 2014 году Яном Гудфеллоу. В ее основе находится алгоритм машинного обучения, который включает две нейронных сети – генеративная нейронная сеть (генератор) и дискриминативная нейронная сеть (дискриминатор), которые работают против друг друга. Задача генератора – генерировать необходимый набор данных из скрытых переменных. В свою очередь дискриминатор старается отличить набор данных, заданный пользователем как истинный, от данных, созданных генератором, и отбраковать их в случае несоответствия. Процесс обучения одной сети и отбраковки другой продолжается до тех пор, пока не достигается минимаксное состояние равновесия. В случае достижения данного состояния, полученные сгенерированные данные становятся едва отличимы от исходных. Одним из главных плюсов представленной сети, это взаимообучение генеративной и дискриминативной моделей. Генеративная сеть пытается повысить процент ошибок у дискриминативной, в то время как дискриминативная улучшает точность распознавания.

Простейшая модель генеративно-состязательной нейронной сети представлена на Рис.3.4.

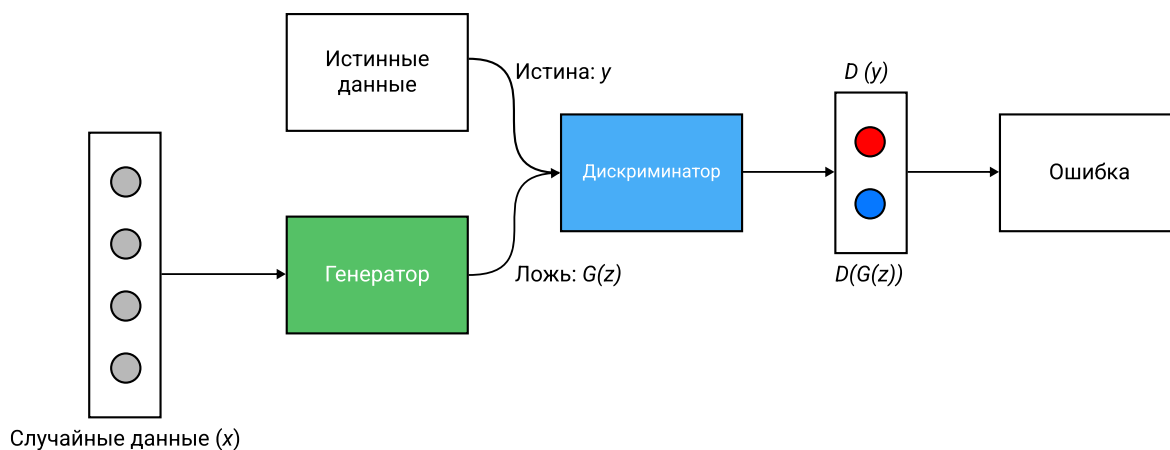


Рис.3.4. Схема генеративно-сопоставительной нейронной сети

Архитектура генератора изображена на Рис.3.5, ее можно разделить на несколько блоков: предварительный блок, остаточные блоки, блок повышения дискретизации, пост-остаточный блок, последний сверточный блок. Предварительный блок содержит сверточный слой и функцию активации ReLU. Данный блок из одноканального входного изображения создает карту признаков с 64 каналами. Далее следуют остаточные блоки, их количество равно 8 и архитектура такого блока представлена на Рис.3.6. Остаточный блок включает два последовательных сверточных слоя с функцией активацией ReLU и батч-нормализацией, на выходе данного блока происходит суммирование входа и выхода данного блока. В конце всех блоков также происходит суммирование входных признаков в первый остаточный блок и признаков, полученных на входе 8 блока. Пост-остаточный блок также содержит сверточный слой, функцию ReLU и батч-нормализацию. В блоке повышения дискретизации чередуется слой, отвечающий за масштабирование признаков с помощью кубической интерполяции и сверточный слой с ReLU функцией. Последний сверточный слой совместно с сигмоидной функцией активацией приводит значения к диапазону от 0 до 1.

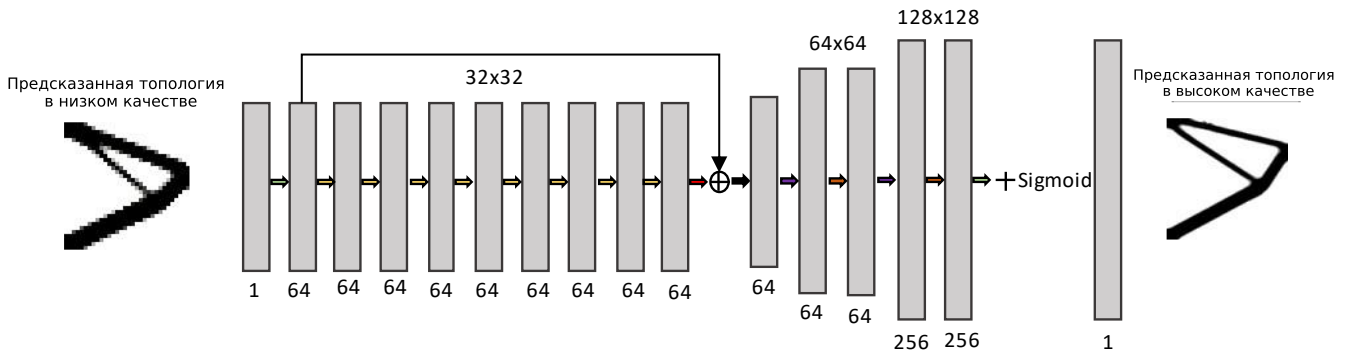


Рис.3.5. Архитектура генератора

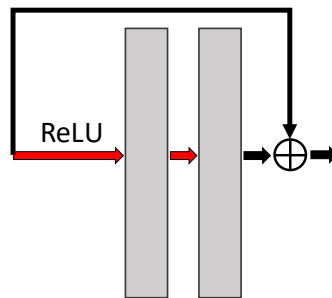


Рис.3.6. Архитектура простого блока с остаточной связью

Архитектура дискриминатора, представленная на Рис.3.7, состоит из нескольких последовательных слоев, которые выполняют операции свертки. Для сжатия размера тензора (ширины и высоты) также используется сверточная операция с шагом (stride) равным 2. В конце архитектуры дискриминатора располагается полносвязный слой, который совместно с сигмоидной функцией преобразует выход в виде скаляра, который указывает вероятность того, является ли изображение действительным или сгенерированным.

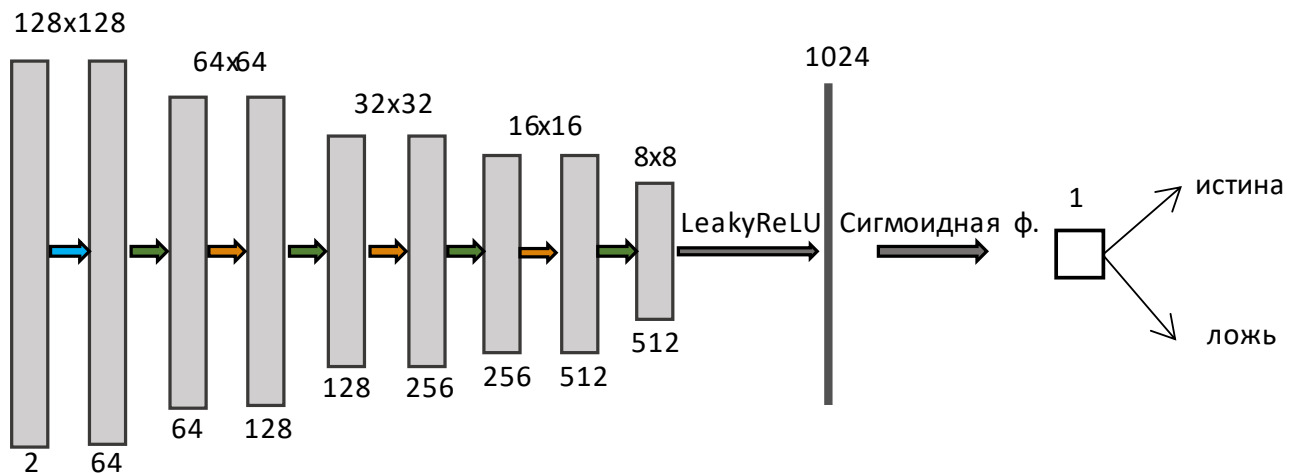


Рис.3.7. Архитектура дискриминатора

На Рис.3.8 отдельно выписаны названия слоев с размерами фильтров и значениями шагов для генеративно-состязательной нейронной сети. Если значение шага опущено, то по умолчанию оно принято равным 1.

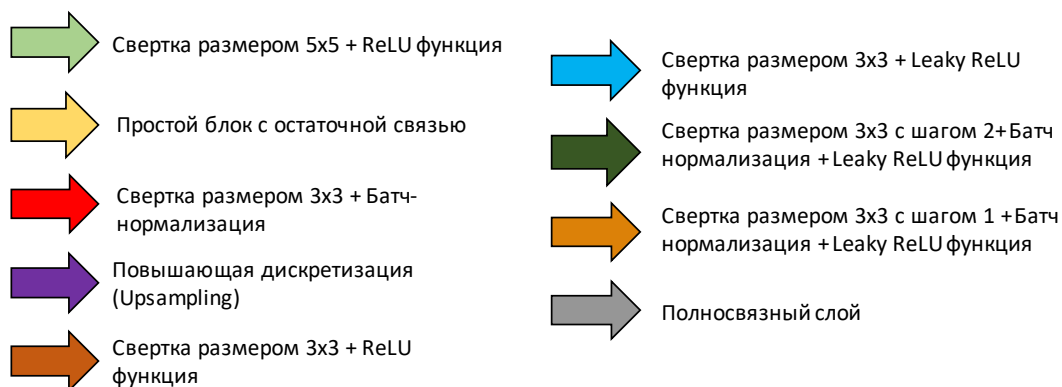


Рис.3.8. Слои генеративно-состязательной нейронной сети

3.3. Разработка двухэтапной модели

Двухэтапная модель была разработана с учетом особенностей тестового примера. Для демонстрации работы данного метода рассматривается изотропная квадратная упругая пластина, сторона которой равна 1 см. С одной стороны пластина жестко закреплена, с противоположной стороны действует сила в узле, равная $F = 1$ кН (см. Рис.3.9). Материал пластины обладает следующими характеристиками: модуль Юнга $E = 0.1603$ МПа, коэффициент Пуассона $\nu = 0.32$.

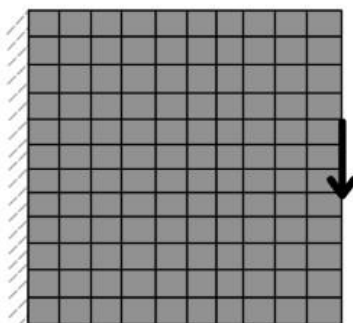


Рис.3.9. Геометрия и нагрузки рассматриваемого примера

Для обучения модели необходимо сгенерировать набор данных, состоящий из изображений низкого качества, высокого качества и четырех матриц, содержащих информацию о нагрузках и граничных условия. На Рис.3.10 изображена модель, используемая для получения оптимального топологии в виде изображений

размерами (32,32) и (128,128). Получение данных результатов осуществляется посредством метода конечных элементов и SIMP метода при ограничении на объемную долю в 30%.

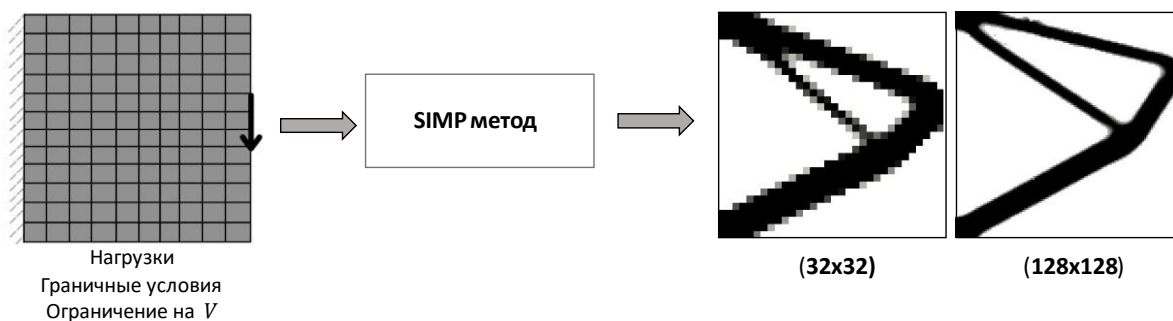


Рис.3.10. Генерация обучающих данных

Всего получено 20 топологий для разных точек приложения нагрузки. Для каждой пары оптимизированных топологий необходимо сформировать 4 матрицы размером (33, 33) и скалярную величину – объемную долю. Набор матриц содержит информацию о проекциях приложенной силы и проекциях перемещений, выступающих в виде ограничений. Пример четырех матриц размером (10, 10) приведен на Рис.3.11. Данный размер не имеет отношения к примеру, и был выбран для удобства представления данных. В матрицах, соответствующих проекциям сил F_x и F_y , содержатся значения проекций силы. В данном случае номер ячейки зависит от геометрического расположения узла конечно-элементной модели. У силы приложенной вертикально имеется только одна проекция $F_y = 1$, которая помещается в ячейку, выделенную желтым цветом. Аналогичная ситуация происходит и с перемещениями, так как левая грань полностью закреплена, то в первый столбец матрицы заполняется единицами. В общем случае сила может быть приложена к модели под углом в плоскости рисунка, в таком случае появится дополнительная проекция F_x .

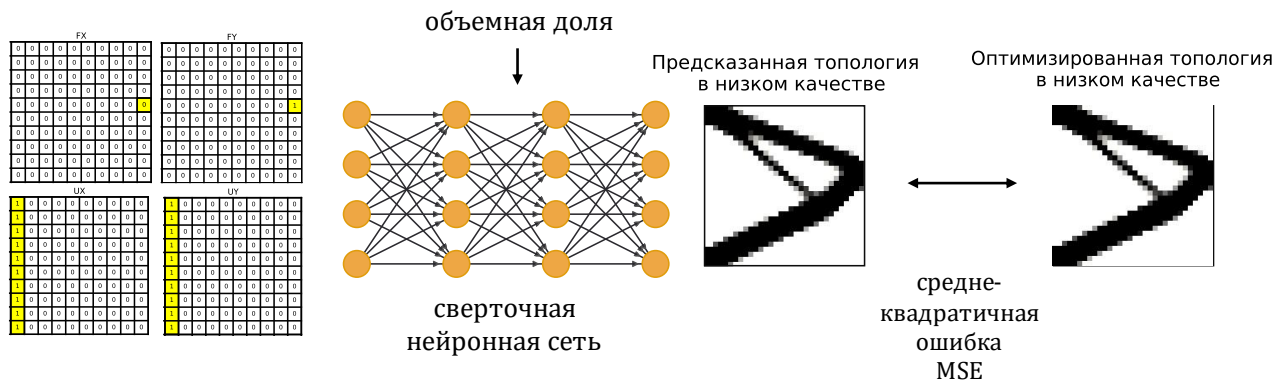


Рис.3.12. Обучение сверточной нейронной сети

В качестве функции потерь здесь выбрана суммарная среднеквадратичная ошибка между изображениями предсказанных и оптимизированных топологий. Для тренировочного процесса использовался ADAM оптимизатор, который представляет собой градиентный алгоритм оптимизации стохастических целевых функций, использовался для определения оптимальных весов сверточной нейронной сети. Процесс обучения был остановлен на 122 эпохе при достижении удовлетворительного минимального значения, которое составило 0.99%.

На третьем этапе происходит обучение генеративно-сопоставительной нейронной сети. Обучение начинается с дискриминатора, на вход которому подаются истинные и ложные данные. Задача дискриминатора состоит в том, чтобы обучиться различать ложные данные от истинных. Таким образом, истинным выходным данным дискриминатор должен присвоить метку 1, а ложным 0. Истинные данные содержат два изображения: предсказанная топология сверточной нейронной сетью с низким разрешением размером (32,32) и оптимизированная топология в высоком разрешении размеров (128,128). Ложные данные содержат также два изображения: предсказанная топология сверточной нейронной сетью с низким разрешением размером (32,32) и предсказанная топология генератором в высоком разрешении размеров (128,128). Для объединения изображений в пары необходимо интерполировать изображение размером (32,32) до размера (128,128).

Далее параметры нейронной сети дискриминатора «замораживаются» и начинается обучение генератора. Входные изображения низкого разрешения оптимизированной топологии пропускаются через слои генератора, далее

полученное изображение в высоком качестве совместно с изображением низкого качества подаются на вход дискриминатора в попытке, что дискриминатор присвоит метку 1. Таким образом, в процессе обратного распространения ошибки веса дискриминатора остаются неизменными, и генератор обучается таким образом, чтобы его выходное значение приближалось к истинному.

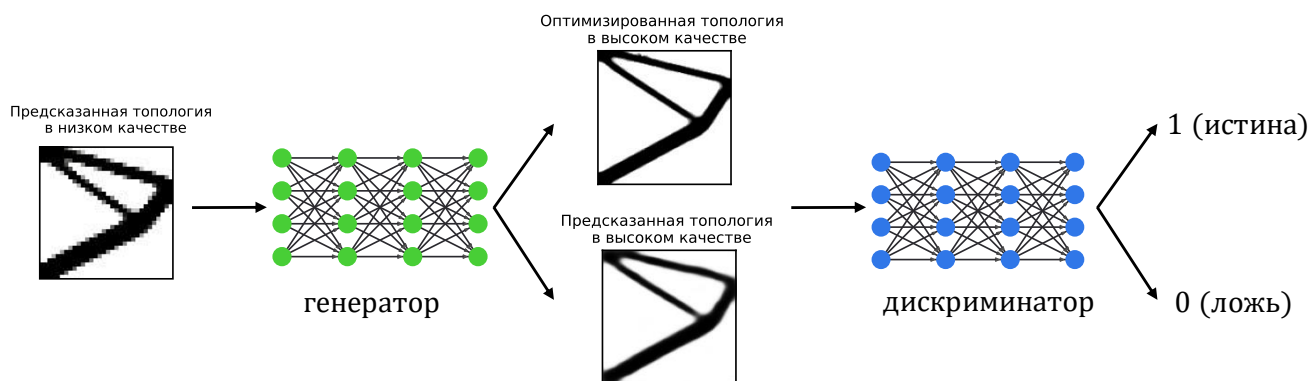


Рис.3.13. Обучение генеративно-сопоставительной нейронной сети

Для обучения генеративно-сопоставительной нейронной сети используется комбинация двух функций потерь: функция потерь дискриминатора и функция потерь генератора. Для обеих моделей используется функция кросс-энтропии:

$$L_{GAN} = E_{c,y \sim p_{data}(c,y)} [\log(D(c, y))] + E_{c,y \sim p_{data}(c,y)} [\log(1 - D(c, G(c)))] \quad (3.3)$$

где c – предсказанные изображения оптимальной топологии в низком разрешении, y – оптимизированные топологии в высоком разрешении, $G(c)$ – предсказанные изображения оптимальной топологии в высоком разрешении.

Также для улучшения обучения генератора добавляется среднеквадратичная ошибка, которая считает погрешность между изображениями оптимизированных топологий в высоком качестве и предсказанных генератором. Таким образом суммарная функция потерь имеет вид:

$$L_{\text{финальная}} = L_{GAN} + MSE(G(c), y) \quad (3.4)$$

Закрывающим этапом является тестирование нейронной сети. Для создания интегрированной модели, необходимо к сверточной нейронной сети подключить генератор (Рис.3.14). Производительность двухэтапной модели оценивается на тестовых примерах.

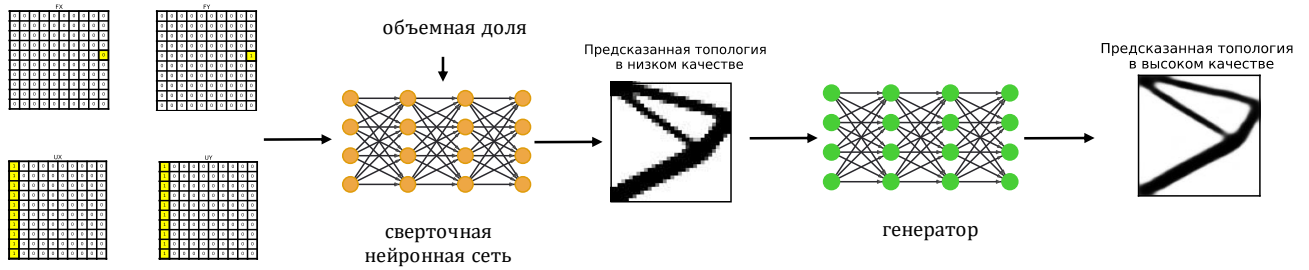


Рис.3.14. Предсказание двухэтапной модели

3.4. Результаты

Как упоминалось выше, сгенерированный набор данных был разделен в соотношении 7:3. Таким образом, результаты работы построенной модели будут демонстрироваться на оставшихся 6 примерах.

На рис.3.15.-3.17. приведены результаты предсказаний двухэтапной модели. Здесь выведены как результаты предсказания сверточной нейронной сети по входным параметрам: матрица нагрузки и граничных условий, объемной доли. В данных примерах объемная доля выбрана равной 0.3. Эти результаты обозначены как предсказанные модели в низком качестве. Также представлены результаты предсказания генератор в высоком качестве, на вход которого поступали выход сверточной нейронной сети.

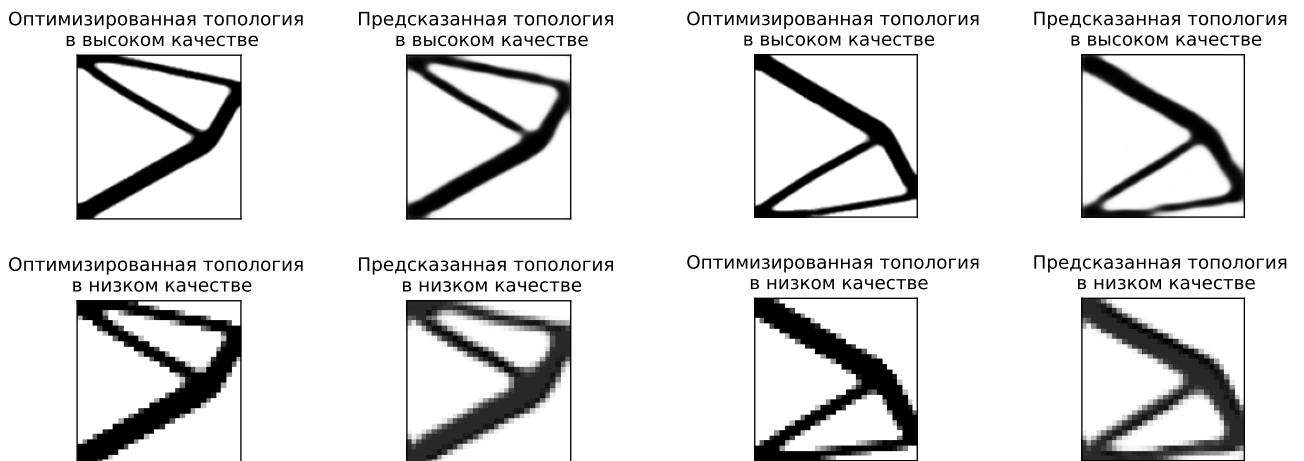


Рис.3.15. Предсказание двухэтапной модели

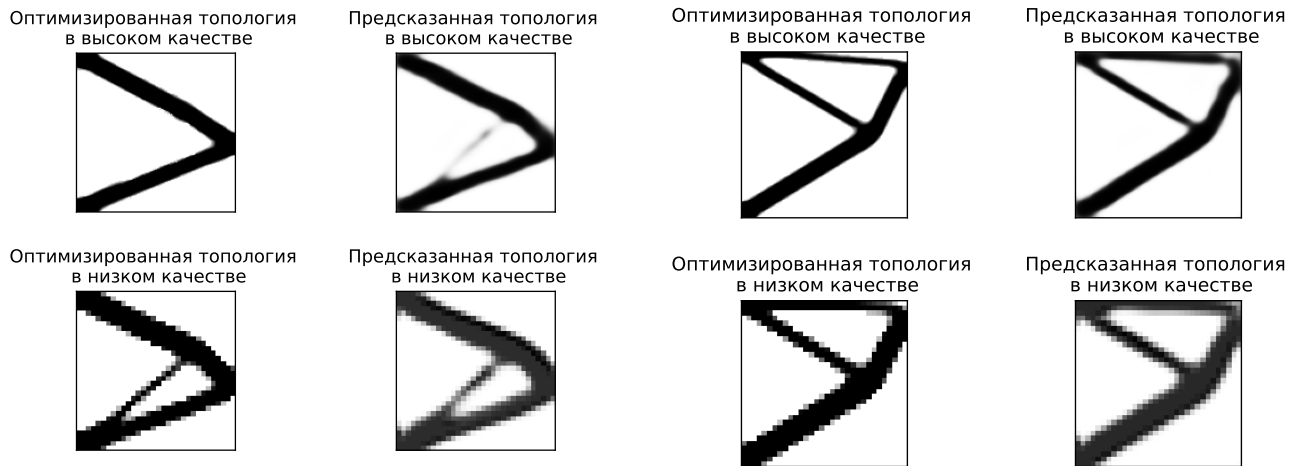


Рис.3.16. Предсказание двухэтапной модели

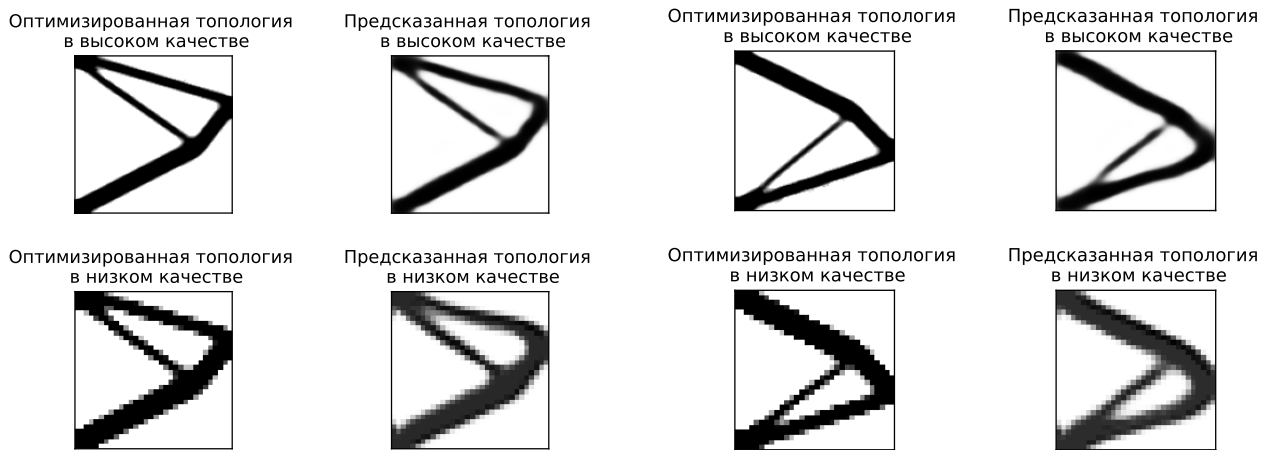


Рис.3.17. Предсказание двухэтапной модели

На представленных рисунках видно, что двухступенчатая модель успешно справилась с поставленной задачей и предсказала почти идентичные оптимальные топологии. Небольшая неточность достигается в области узких структур, а также на границах полученной топологии. Такая погрешность вполне приемлема, так как существенных потерь в качестве не наблюдается.

ЗАКЛЮЧЕНИЕ

В ходе данной работы была разработана двухэтапная модель. Набор данных оптимизированных топологий в сочетании с настройками оптимизации и граничными условиями генерировались с помощью кода на основе метода конечных элементов и SIMP метода. Каждый набор данных был создан как в низком качестве (32x32), так и в высоком (128,128) качестве при одинаковых граничных условиях, нагрузках и настройках оптимизации. Полученные наборы данных были разделены на обучающие и тестовые выборки. Сверточная нейронная сеть была обучена на сгенерированном обучающем наборе данных в низком разрешении, матрицах, содержащих информацию о нагрузках и граничных условиях, и скалярной величине - объемной доле. Затем для уточнения полученных данных была добавлена генеративно-состязательная нейронная сеть, которая обучалась как на изображениях оптимальных топологий низкого разрешения, так и высокого. Производительность интегрированной сети оценивалась на тестовом примере изотропной упругой пластины.

Наглядно показано, что предсказанные структуры высокого разрешения визуально аналогичны оптимизированным структурам высокого разрешения. Кроме того, эффективность данного подхода заключается в его способности предсказать структуру близкой к оптимально с меньшими затратами вычислительных ресурсов и без каких-либо итераций. Данный подход может пригодиться в случаях, когда требуется быстрый анализ и предсказание результатов в реальном времени.

Однако для дальнейшей работы над данной моделью необходимо учесть ряд ограничений. Например, добавить возможность прогнозировать оптимальную топологию для трехмерных задач. Сложность заключается в подготовке тренировочного набора данных и увеличении размерности самой нейронной сети. Таким образом, обучение может занять очень много времени и будет вызывать проблемы в подборе настройки модели. Также предложенную модель следует обобщить на различные граничные условия и ограничения оптимизации, такие как смещения или напряжения.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Abueidda D. W., Koric S., Sobh N. A., 2020. Topology optimization of 2D structures with nonlinearities using deep learning. *Computers & Structures*, 237, 106283.
2. Almasri W., Bettebghor D., Ababsa F., Danglade F., Adjed F., 2021, July. Deep Learning Architecture for Topological Optimized Mechanical Design Generation with Complex Shape Criterion. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems* (pp. 222-234). Springer, Cham.
3. Baki H., Kandula E., Sai K., Balaji S. Topology Optimization Using Convolutional Neural Network. In *Lecture Notes in Mechanical Engineering*, pages 301–307. Springer, 2020
4. Brown N., Garland A.P., Fadel G.M., Li G. Deep Reinforcement Learning for Engineering Design Through Topology Optimization of Elementally Discretized Design Domains. *Mater. Des.* 2022, 218, 110672.
5. Chandrasekhar A., Suresh K, 2021c. TOuNN: topology optimization using neural networks. *Structural and Multidisciplinary Optimization*, 63(3), 1135-1149.
6. Dustin B., Darshil P., Rahul R., and Gary F. D. Multi-stage deep neural network accelerated topology optimization. *Structural and Multidisciplinary Optimization*, 64(6):3473–3487, 12 2021. ISSN 1615-147X.
7. Gorkem C. A., Recep M. G.. Two-stage convolutional encoder-decoder network to improve the performance and reliability of deep learning models for topology optimization. *Structural and Multidisciplinary Optimization*, 63(4):1927–1950, 4 2021. ISSN 1615-147X.
8. Halle A., Campanile L. F., Hasse A. (2021). An Artificial Intelligence–Assisted Design Method for Topology Optimization without Pre-Optimized Training Data. *Applied Sciences*, 11(19), 9041
9. Hoyer S., Sohl-Dickstein, J., Greydanus S. (2019). Neural reparameterization improves structural optimization. arXiv preprint arXiv:1909.04240.

10. Jiang X., Wang H., Li Y., Mo K. (2020). Machine Learning based parameter tuning strategy for MMC based topology optimization. *Advances in Engineering Software*, 149, 102841.
11. Kallioras N. A., Kazakis G., Lagaros N. D. (2020). Accelerated topology optimization by means of deep learning. *Structural and Multidisciplinary Optimization*, 62(3), 1185-1212
12. Karlsson P., Pejryd L., Strömberg N. (2020, September). Generative Design Optimization and Characterization of Triple Periodic Lattice Structures in AlSi10Mg. In *International Conference on Additive Manufacturing in Products and Applications* (pp. 3-16). Springer, Cham
13. Keshavarzzadeh V., Alirezaei M., Tasdizen T., Kirby, R. M. (2021). Image-Based Multiresolution Topology Optimization Using Deep Disjunctive Normal Shape Model. *Computer-Aided Design*, 130, 102947
14. Kim C., Lee J., Yoo J. (2021). Machine learning-combined topology optimization for functionary graded composite structure design. *Computer Methods in Applied Mechanics and Engineering*, 387, 114158.
15. Kollmann H. T., Abueidda D. W., Koric S., Guleryuz E., & Sobh, N. A. (2020). Deep learning for topology optimization of 2D metamaterials. *Materials & Design*, 196, 109098.
16. Lei X., Liu C., Du Z., Zhang W., Guo X. (2019). Machine learning-driven real-time topology optimization under moving morphable component-based framework. *Journal of Applied Mechanics*, 86(1), 011004.
17. Mohammad M. B., Horea T. Ilies. GANTL: Towards Practical and Real-Time Topology Optimization with Conditional GANs and Transfer Learning. *Journal of Mechanical Design*, pages 1–32, 10 2021. ISSN 1050-0472.
18. Muñoz D., Nadal E., Albelda J., Chinesta F., Ródenas J.J. Allying topology and shape optimization through machine learning algorithms. *Finite Elements in Analysis and Design*, 204:103719, 7 2022. ISSN 0168874X.

19. Napier N., Sriraman S. A., Tran H. T., James K. A. (2020). An artificial neural network approach for generating high-resolution designs from low-resolution input in topology optimization. *Journal of mechanical design*, 142(1).
20. Nie Z., Lin T., Jiang H., Kara L. B. (2021). Topologygan: Topology optimization using generative adversarial networks based on physical fields over the initial domain. *Journal of Mechanical Design*, 143(3).
21. Sharpe C., Seepersad C. C. (2019, August). Topology design with conditional generative adversarial networks. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (Vol. 59186, p. V02AT03A062). American Society of Mechanical Engineers
22. Sosnovik I, Oseledets I. Neural networks for topology optimization. *Russian Journal of Numerical Analysis and Mathematical Modelling*, 2019, 34(4): 215–223.
23. Strömberg, N. (2020). Efficient detailed design optimization of topology optimization concepts by using support vector machines and metamodels. *Engineering Optimization*, 52(7), 1136-1148
24. Sumudu H., Udith H. Topologically optimal design and failure prediction using conditional generative adversarial networks. *International Journal for Numerical Methods in Engineering*, 122(23):6867–6887, 12 2021. ISSN 0029-5981.
25. Takahashi Y., Suzuki Y., Todoroki, A. (2019). Convolutional neural network-based topology optimization (cnn-to) by estimating sensitivity of compliance from material distribution
26. Tinghao G., Danny J. Lohan, Cang R., Max Yi Ren, and James T. Allison. An Indirect Design Representation for Topology Optimization Using Variational Autoencoder and Style Transfer. In *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Reston, Virginia, 1 2018. American Institute of Aeronautics and Astronautics. ISBN 978-1-62410-532-6
27. White D. A., Arrighi W. J., Kudo J., Watts S. E. (2019). Multiscale topology optimization using neural network surrogate models. *Computer Methods in Applied Mechanics and Engineering*, 346, 1118-1135

28. Younghwan J., Yonggyun Y., In Gwun Jang. Unit Module-Based Convergence Acceleration for Topology Optimization Using the Spatiotemporal Deep Neural Network. IEEE Access, 9:149766–149779, 2021. ISSN 2169-3536. doi: 10.1109/ACCESS.2021.3125014