Глава 1

Введение

Автоматическая постановка и дополнение обучающих задач востребованное направление в сфере образования. Алгоритмические методы позволяют разрешать классические проблемы образования, включающие составление методической литературы, пресечение недобросовестной кооперации обучающихся при индивидуальном контроле знаний, формирование индивидуальной образовательной траектории.

Работы также показывают успешное применение автоматической генерации для формирования индивидуальной образовательной траектории тематически однородного, но разноуровневого по сложности. Тем не менее предложенные подходы требуют значительных временных экспертов для создания новых методических курсов.

Стремительное развитие генеративного моделирования в областях естественного языка [**radford2019language**] и машинного зрения [**rombach2022highresolution**][**song2020ge**
определили новые подходы в известных задачам нотариального консультирования.

Глава 2

Методика составления педагогических задач

Педагогическая задача является основой образовательного процесса и играет ключевую роль в достижении учебных целей. Её цель состоит в том, чтобы обеспечить студентам определённые образовательные возможности и помочь им развить необходимые знания, умения и навыки.

При создании педагогической задачи важно учитывать не только содержание обучения, но и индивидуальные особенности студентов, их уровень знаний и способности. Педагогическая задача должна быть четко сформулирована, чтобы студенты могли понять, что от них требуется, и чувствовать уверенность в выполнении задания.

Важным аспектом педагогической задачи является её реалистичность и актуальность. Задача должна иметь практическую ценность и быть связанной с реальными жизненными ситуациями или профессиональными задачами. Это поможет стимулировать интерес и мотивацию студентов к изучению материала.

Педагогическая задача также должна предоставлять возможность для развития критического мышления и применения знаний на практике. Она должна быть структурированной и обеспечивать возможность оценки выполнения студентами поставленной задачи. Критерии оценки должны быть ясными и объективными, чтобы обеспечить справедливую оценку достижения учебных целей.

Реализация педагогической задачи может включать использование различных методов обучения и оценки, таких как групповая работа, проектная деятельность, обсуждения, решение проблемных ситуаций и другие. Это позволит стимулировать активное участие студентов в образовательном процессе и способствовать их полноценному развитию.

2.1 Методическая задача

2.2 Виды задач

2.3 Содержание педагогической задачи

2.4 Структура методического материала

Задачник - это учебный ресурс, который обычно организован в виде последовательного набора задач, объединенных по общей теме или концепции. Он служит для систематизации и структурирования материала, предоставляя студентам или исследователям возможность практического применения знаний и навыков.

Структура задачника обычно состоит из разделов или тематических блоков, которые охватывают определенные аспекты изучаемой области. Каждый блок начинается с введения в тему, где обозначаются ключевые понятия и основные принципы, а также могут даваться краткие объяснения теоретических аспектов.

Задачи внутри блоков обычно распределены в порядке возрастания сложности или последовательности углубления в изучаемую тему. Начальные задачи могут быть более простыми и базовыми, позволяя студентам получить предварительное понимание концепции, после чего следуют более сложные задачи, требующие более глубокого анализа и применения полученных знаний.

Каждая задача обычно сопровождается пояснениями или инструкциями, которые помогают студентам понять, как решить задачу, и обычно включает в себя краткое описание цели задачи и указания на соответствующие теоретические материалы.

Важно, чтобы задачник обеспечивал разнообразие заданий, таких как теоретические задачи, практические задания и примеры, а также давал возможность студентам проверить свои знания и навыки через различные виды задач.

В конце задачника обычно предоставляются дополнительные ресурсы, такие как дополнительная литература, ссылки на онлайн-ресурсы или рекомендации по дальнейшему обучению, что помогает студентам расширить свои знания и глубже понять изучаемую тему.

Глава 3

Методы машинного обучения для работы с текстом и изображениями

В рамках секции будут описаны методы, применяемые в генеративном моделировании для решения задачи генерации задач.

3.1 Использование нейросетевых подходов

Нейронные сети представляют собой вычислительные модели, состоящие из узлов, называемых нейронами, организованных в слои. Каждый нейрон взвешивает входные сигналы, представленные как вектор $\mathbf{x} = (x_1, x_2, ..., x_n)$, с весами $\mathbf{w} = (w_1, w_2, ..., w_n)$ и смещением b, где n - количество входов, x_i - i-й входной сигнал, w_i - весовой коэффициент i-го входа, b - смещение (bias). На выходе нейрона производится линейная комбинация входов с весами и смещением:

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

Полученная сумма z затем подвергается нелинейному преобразованию при помощи функции активации $f(z)$, которая определяет активацию нейрона:

$$y = f(z)$$

Функция активации обычно вводится для добавления нелинейности в модель, что позволяет нейронной сети моделировать сложные нелинейные зависимости в данных. Некоторые из распространенных функций активации включают в себя сигмоидальную функцию (σ), гиперболический тангенс (tanh), ReLU (Rectified Linear Unit) и их вариации.

В случае многослойной нейронной сети, выходы нейронов одного слоя становятся входами для следующего слоя, образуя цепочку преобразований. Процесс передачи данных через нейроны последовательных слоев называется прямым распространением (forward propagation).

Нейронные сети обучаются путем настройки весов $\mathbf{w}$ и смещений b с использованием алгоритмов оптимизации, таких как градиентный спуск. Во время обучения модель минимизирует функцию потерь L, которая оценивает разницу между предсказанным результатом и истинным значением:

$$L = \frac{1}{N} \sum_{i=1}^{N} L(y_i, \hat{y}_i)$$

где N - количество обучающих примеров, y_i - истинное значение, $\hat{y}_i$ - предсказанное значение.

Long-Short Term Mermory

Transformer

Модель Transformer является архитектурой глубокого обучения, предназначенной для обработки последовательных данных, таких как тексты или временные ряды. Она была предложена в статье "Attention is All You Need"и стала одной из самых инновационных архитектур в области обработки естественного языка.

Основной компонент модели Transformer - это механизм внимания (Attention Mechanism). Он позволяет модели сосредоточиться на наиболее важных частях входных данных при выполнении задач, таких как машинный перевод или обработка текста.

Механизм внимания в Transformer состоит из трех основных частей:

1. Query, Key, Value (QKV): Это три набора весов, которые модель изучает во время обучения. Они используются для вычисления весов входных данных и определения их важности для каждого элемента. Формально, для каждого элемента x_i во входных данных, вычисляются query q_i, key k_i и value v_i векторы:

$$q_i = x_i W_q,$$

$$k_i = x_i W_k,$$

$$v_i = x_i W_v,$$

где W_q, W_k, W_v - матрицы весов, которые модель обучает.

2. Attention Scores: После вычисления query и key векторов, для каждого элемента x_i вычисляются attention scores e_{ij} по формуле:

$$e_{ij} = \frac{q_i \cdot k_j}{\sqrt{d_k}},$$

где d_k - размерность ключевого (или запросового) пространства. Этот шаг позволяет модели оценить важность каждого элемента входных данных для каждого другого элемента.

3. Веса внимания (Attention Weights): Attention scores преобразуются в веса внимания α_{ij} с помощью функции softmax:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{j'} \exp(e_{ij'})}.$$

Эти веса показывают, какую важность модель придает каждому элементу данных при решении конкретной задачи.

После вычисления весов внимания, они умножаются на соответствующие значения (value) и суммируются, чтобы получить итоговый взвешенный вектор, который представляет собой выход механизма внимания.

Механизм внимания в Transformer может быть использован в нескольких вариантах: внутри блоков кодировщика и декодировщика для обработки последовательных данных в машинном переводе, внутри блоков самовнимания (self-attention) для обработки последовательных данных в других задачах, таких как генерация текста или анализ сентимента.

Таким образом, механизм внимания в модели Transformer позволяет ей эффективно обрабатывать и анализировать последовательные данные, учитывая их важность и контекст.

Низкоранговый адаптеры

Модель Transformer является архитектурой глубокого обучения, предназначенной для обработки последовательных данных, таких как тексты или временные ряды. Она была предложена в статье "Attention is All You Need"и стала одной из самых инновационных архитектур в области обработки естественного языка.

Основной компонент модели Transformer - это механизм внимания (Attention Mechanism). Он позволяет модели сосредоточиться на наиболее важных частях входных данных при выполнении задач, таких как машинный перевод или обработка текста.

Механизм внимания в Transformer состоит из трех основных частей:

1. Query, Key, Value (QKV): Это три набора весов, которые модель изучает во время обучения. Они используются для вычисления весов входных данных и определения их важности для каждого элемента. Формально, для каждого элемента x_i во входных данных, вычисляются query q_i, key k_i и value v_i векторы:

$$q_i = x_i W_q,$$

$$k_i = x_i W_k,$$

$$v_i = x_i W_v,$$

где W_q, W_k, W_v - матрицы весов, которые модель обучает.

2. Attention Scores: После вычисления query и key векторов, для каждого элемента

x_i вычисляются attention scores e_{ij} по формуле:

$$e_{ij} = \frac{q_i \cdot k_j}{\sqrt{d_k}},$$

где d_k - размерность ключевого (или запросового) пространства. Этот шаг позволяет модели оценить важность каждого элемента входных данных для каждого другого элемента.

3. Веса внимания (Attention Weights): Attention scores преобразуются в веса внимания α_{ij} с помощью функции softmax:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{j'} \exp(e_{ij'})}.$$

Эти веса показывают, какую важность модель придает каждому элементу данных при решении конкретной задачи.

После вычисления весов внимания, они умножаются на соответствующие значения (value) и суммируются, чтобы получить итоговый взвешенный вектор, который представляет собой выход механизма внимания.

Механизм внимания в Transformer может быть использован в нескольких вариантах: внутри блоков кодировщика и декодировщика для обработки последовательных данных в машинном переводе, внутри блоков самовнимания (self-attention) для обработки последовательных данных в других задачах, таких как генерация текста или анализ сентимента.

Таким образом, механизм внимания в модели Transformer позволяет ей эффективно обрабатывать и анализировать последовательные данные, учитывая их важность и контекст.

3.2 Генеративные подходы

Энергетические методы

Energy-based подходы в машинном обучении представляют собой методы моделирования, основанные на понятии энергии системы. В этом подходе модель оценивает энергию данных, а затем использует эту оценку для различных задач, таких как классификация, регрессия или генерация данных. Основная идея состоит в том, что модель стремится минимизировать энергию для реальных данных и увеличивать ее для фиктивных (сгенерированных) данных.

Пусть $\mathbf{x}$ — наблюдаемая переменная (например, вектор признаков), а $E(\mathbf{x}; \theta)$ — энергия, присвоенная данным $\mathbf{x}$ моделью с параметрами θ. Тогда модель может быть описана следующим образом:

$$E(\mathbf{x}; \theta)$$

Здесь θ — параметры модели, которые подлежат обучению.

Energy-based модели могут быть использованы для решения различных задач. Например, в задаче классификации модель может устанавливать низкую энергию для данных из правильного класса и высокую для данных из неправильного класса. Для регрессии модель может предсказывать энергию, близкую к целевому значению.

Обучение energy-based моделей часто включает минимизацию функции потерь, которая может быть определена как разница между энергией реальных данных и сгенерированных данных. Один из часто используемых подходов - это минимизация отклонения (discrepancy) между энергиями реальных данных $\mathbf{x}$ и сгенерированных данных $\tilde{\mathbf{x}}$:

$$L(\theta) = E(\mathbf{x}; \theta) - E(\tilde{\mathbf{x}}; \theta)$$

Такой подход позволяет модели стремиться к минимизации энергии для реальных данных и максимизации энергии для сгенерированных данных.

Energy-based подходы имеют широкий спектр применений и используются в различных областях, включая глубокое обучение, генеративные модели и обучение без учителя. Они представляют собой мощный инструмент для моделирования данных с использованием концепции энергии, что позволяет справляться с различными задачами в машинном обучении.

Модели потоков

3.2.1 Диффузионные модели

Диффузионные модели, также известные как модели с диффузией, представляют собой класс вероятностных моделей, используемых для генерации данных, включая изображения. Основная идея диффузионных моделей состоит в том, чтобы последовательно преобразовывать начальное изображение, добавляя к нему случайный шум на каждом шаге. Это делается путем итеративного применения условных вероятностных моделей, которые моделируют распределение пикселей изображения. Этот процесс постепенно улучшает качество изображения, приближая его к реальным данным из обучающего набора. Важным преимуществом диффузионных моделей является их способность генерировать высококачественные изображения без необходимости использования генеративно-состязательных сетей или других архитектур глубокого обучения. Кроме того, эти модели позволяют контролировать процесс генерации изображений, например, регулируя уровень шума или изменяя другие параметры. Таким образом, диффузионные модели представляют собой эффективный и гибкий метод генерации изображений, который находит применение в различных областях, включая компьютерное зрение, графический дизайн и искусственный интеллект.

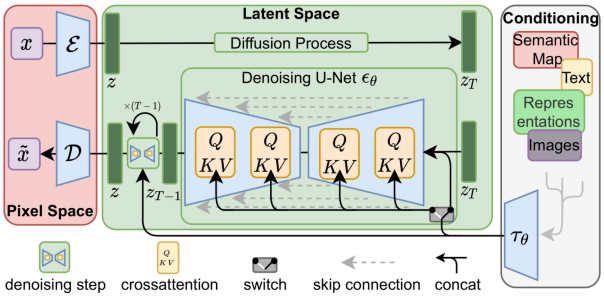

Рис. 3.1: Иллюстрация модели Stable Diffusion [**stablediffusion**]

3.2.2 Оптимальный транспорт

Задача оптимального транспорта (Optimal Transport) является одним из ключевых понятий в области математической экономики, теории вероятностей, теории изображений и машинного обучения. Она представляет собой проблему определения оптимального способа перемещения массы из одной распределенной системы в другую с минимальными затратами или стоимостью. Основная идея заключается в том, чтобы найти наилучшее соответствие между двумя распределениями, учитывая их форму и объем.

Постановка задачи от Монге начинается с рассмотрения двух вероятностных распределений μ и ν на двух метрических пространствах $\mathcal{X}$ и $\mathcal{Y}$. Задача состоит в поиске отображения $T : \mathcal{X} \to \mathcal{Y}$, которое переводит распределение μ в распределение ν, минимизируя некоторую функцию стоимости. Функция стоимости обычно является мерой сходства между элементами из $\mathcal{X}$ и $\mathcal{Y}$, такой как квадрат расстояния. Математически задача Монге формулируется как:

$$\inf_{\gamma \in \Pi(\mu,\nu)} \int_{\mathcal{X}\times\mathcal{Y}} c(x,y)\, d\gamma(x,y),$$

где $\Pi(\mu,\nu)$ обозначает множество всех возможных совместных распределений γ на $\mathcal{X} \times \mathcal{Y}$ с фиксированными маргинальными распределениями μ и ν, а $c(x,y)$ — функция стоимости перевозки массы из x в y.

С другой стороны, постановка задачи от Кантаровича вводит понятие потенциала. Задача состоит в поиске потенциала $\phi : \mathcal{X} \to \mathbb{R}$, который минимизирует функционал стоимости:

$$\inf_{\phi} \left(\int_{\mathcal{X}} \phi(x)\, d\mu(x) + \int_{\mathcal{Y}} \psi(y)\, d\nu(y) \right),$$

где ψ — обратная функция к ϕ. Таким образом, отображение $T : \mathcal{X} \to \mathcal{Y}$ получается из градиента потенциала.

Обе постановки задачи имеют широкий спектр приложений, включая согласование распределений в машинном обучении, обработку изображений, экономическую теорию и транспортную логистику. Оптимальный транспорт представляет собой мощный математический инструмент для анализа и моделирования различных

процессов с перемещением массы.

3.3 Обработка естественного языка

Раздел имеет две части представление языка и его генерация.

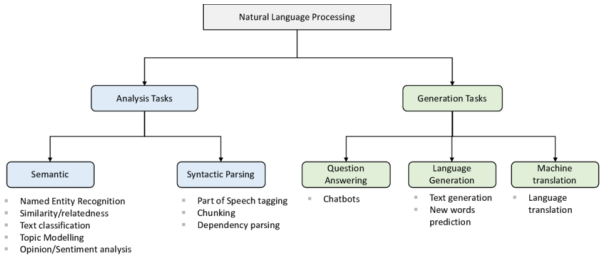

Рис. 3.2: Таксономия современных подходов обработки естественного языка

Анализ естественного языка это межпредметная дисциплина.

3.3.1 Представление

Лемматизация

Лемматизация представляет собой процесс нормализации текста, целью которого является приведение слов к их базовой форме или лемме. В контексте обработки естественного языка (Natural Language Processing, NLP), лемматизация является важным этапом предварительной обработки текста, который позволяет уменьшить размер словаря и улучшить качество анализа.

Формально, лемматизация выражается в преобразовании слова w в его лемму $\text{lemma}(w)$ с использованием правил и алгоритмов, учитывающих морфологические особенности языка. Лемма представляет собой каноническую, нормализованную форму слова, которая может быть использована для обобщения различных грамматических форм одного и того же слова.

Математически, процесс лемматизации может быть представлен как отображение слова w в его лемму $\text{lemma}(w)$:

$$\text{lemma}(w) = \text{лемма}$$

Например, для слова "бегу его леммой будет слово "бежать". Лемматизация помогает уменьшить словарь слов и снизить размерность пространства признаков в текстовых данных, что положительно влияет на производительность алгоритмов обработки текста, таких как классификация или кластеризация.

Применение лемматизации часто сопровождается предварительным шагом токенизации, в котором текст разбивается на отдельные слова или токены. Это позволяет

применить лемматизацию к каждому слову в тексте независимо от контекста. Лемматизация часто используется в различных областях NLP, включая информационный поиск, анализ тональности, машинный перевод и другие.

Векторное представление

Практически востребованной оказалась дистрибутивная гипотеза [**Schutze**], легшая в основу алгоритма [**NIPS2013_9aa42b31**].

В генеративном моделировании естественного языка, встает задача представления слов в виде векторов в многомерном пространстве, что позволяет моделировать семантические и синтаксические аспекты текста в компактной форме. Это представление, известное как "векторное вложение"или "embedding позволяет выразить смысловые и лингвистические свойства слов, используемых в языке.

Формально, векторное вложение $\mathbf{e}_w$ слова w представляет собой векторное представление этого слова в многомерном пространстве:

$$\mathbf{e}_w = (e_{w1}, e_{w2}, ..., e_{wd})$$

где d - размерность пространства вложения (число измерений), e_{wj} - j-ая компонента вектора вложения $\mathbf{e}_w$.

Эти векторные представления обычно изучаются и извлекаются из больших корпусов текстов с использованием различных алгоритмов, таких как word2vec, GloVe (Global Vectors for Word Representation), FastText и другие. Они обладают свойством сохранения семантической близости слов в пространстве вложения: слова, которые часто встречаются в похожих контекстах, имеют близкие векторные представления.

Векторные вложения слов играют важную роль в генеративном моделировании естественного языка, так как они позволяют моделям представлять слова в виде непрерывных числовых значений, которые могут быть использованы как входные данные для алгоритмов машинного обучения. Это позволяет моделям эффективно изучать зависимости между словами и генерировать тексты семантически богатые и лингвистически осмысленные.

3.3.2 Генерация

N-граммы

N-граммы представляют собой последовательности из n элементов в тексте или последовательности символов, где n обозначает количество элементов в последовательности. Элементы могут быть символами, словами или более крупными фрагментами текста в зависимости от контекста применения. Анализ n-грамм является важным методом в обработке естественного языка (Natural Language Processing, NLP) для изучения частотности последовательностей слов или символов в текстовых

данных.

Формально, n-грамма ngram_n длины n в тексте T определяется как последовательность n элементов, где каждый элемент x_i может быть символом, словом или другими единицами текста:

$$\text{ngram}_n = (x_1, x_2, ..., x_n)$$

Использование n-грамм в анализе текста позволяет оценивать частотность последовательностей слов или символов и изучать лингвистические характеристики текста, такие как структура, стиль и тематика. Кроме того, n-граммы могут использоваться в задачах моделирования языка, предсказания следующего слова в предложении, а также в машинном переводе и других приложениях обработки естественного языка.

Авторегрессионая модель

Авторегрессионные модели в генеративном моделировании естественного языка представляют собой класс статистических моделей, которые моделируют вероятностное распределение последовательности слов или символов в тексте. Эти модели базируются на предположении о зависимости текущего элемента последовательности от предыдущих элементов, что позволяет им учитывать контекст и последовательность в текстовых данных.

Формально, в авторегрессионной модели вероятность появления последовательности $W = (w_1, w_2, ..., w_T)$ слов или символов определяется как произведение вероятностей каждого слова при условии предыдущих:

$$P(W) = P(w_1) \cdot P(w_2|w_1) \cdot P(w_3|w_1, w_2) \cdot \ldots \cdot P(w_T|w_1, w_2, ..., w_{T-1})$$

где $P(w_t|w_1, w_2, ..., w_{t-1})$ - вероятность появления слова w_t при условии всех предыдущих слов $w_1, w_2, ..., w_{t-1}$.

Авторегрессионные модели могут быть реализованы с использованием различных подходов, включая марковские модели, рекуррентные нейронные сети и модели с авторегрессионными свойствами, такие как GPT (Generative Pre-trained Transformer) и LSTM (Long Short-Term Memory). Они находят применение в широком спектре задач обработки естественного языка, включая генерацию текста, машинный перевод, синтез речи и другие.

Одной из ключевых особенностей авторегрессионных моделей является их способность учитывать контекст и последовательность слов в тексте, что позволяет им генерировать качественные тексты с учетом структуры и семантики. Тем не менее, выбор подходящей модели и обучение ее требуют значительных вычислительных ресурсов и экспертных знаний в области машинного обучения и обработки естественного языка.

Attention

Механизм внимания (attention) в генеративном моделировании естественного языка является ключевым компонентом в нейронных сетях, позволяющим модели фокусироваться на определенных частях входных данных при выполнении задач обработки текста. Этот механизм позволяет модели адаптироваться к различным контекстам и динамически выделять важные элементы во входных последовательностях.

Формально, предположим, что у нас есть входные данные $X = (x_1, x_2, ..., x_T)$ и контекст C, а также текущее состояние скрытого слоя модели h_t. Механизм внимания вычисляет вектор внимания α, который определяет важность каждого элемента входной последовательности на текущем временном шаге:

$$\alpha_t = \text{softmax}(f(h_t, X))$$

где f - функция, которая вычисляет важность каждого элемента входной последовательности, а softmax применяется для получения нормированных весов внимания.

С использованием вектора внимания α, взвешенная сумма контекста C вычисляется как:

$$context_t = \sum_{i=1}^{T} \alpha_{ti} x_i$$

Полученный контекст используется в дальнейших вычислениях модели для выполнения задач, таких как генерация текста или классификация.

Механизм внимания позволяет модели сосредоточиться на наиболее значимых частях входных данных в каждый момент времени, что делает его особенно полезным для задач, требующих адаптивности и контекстного понимания, таких как машинный перевод, генерация текста и вопросно-ответные системы. Этот механизм стал ключевым инструментом в области генеративного моделирования естественного языка, позволяя моделям эффективно работать с различными типами данных и контекстами.

3.4 Обработка изображений

3.4.1 Компьютерное зрение

Методы решения задач компьютерного зрения включают как классические подходы, так и глубокие нейронные сети.

Классические методы основываются на внутренней структуре и симметрии изображения, его семантике и характеристиках объектов на нем. Эти методы включают в себя алгоритмы обработки изображений, фильтрацию, выделение признаков (например, метод гистограмм градиентов или методы локтевых точек), шаблонное

сопоставление и классификацию на основе характеристик объектов.

Глубокие методы, основанные на сверточных нейронных сетях (CNN), стали широко распространенными и эффективными в решении задач компьютерного зрения. Эти методы автоматически изучают признаки изображений на различных уровнях абстракции, начиная от низкоуровневых признаков, таких как грани и текстуры, до высокоуровневых семантических признаков, связанных с объектами и их распознаванием. CNN состоит из нескольких слоев, включая сверточные, пулинговые и полносвязные слои, которые работают в совокупности для изучения и классификации изображений.

Эти два класса методов часто комбинируются для достижения лучших результатов в решении задач компьютерного зрения. Например, глубокие нейронные сети могут использоваться для извлечения признаков из изображений, а затем классические методы могут применяться для анализа этих признаков и решения конкретных задач, таких как детекция объектов или сегментация изображений.

Сверточные нейронные сети

Сверточные нейронные сети (CNN)[**lecun1989handwritten**] представляют собой класс глубоких нейронных сетей, специально разработанных для обработки структурированных данных, таких как изображения. Они эффективно работают за счет своей способности автоматически извлекать иерархические признаки из входных данных, что делает их особенно подходящими для задач компьютерного зрения.

Основными компонентами сверточных нейронных сетей являются сверточные слои, пулинг слои и полносвязанные слои. Сверточные слои выполняют операции свертки над входными данными с использованием фильтров или ядер, чтобы извлечь локальные пространственные признаки, такие как грани, углы и текстуры. Это позволяет модели обнаруживать абстрактные особенности изображений на разных уровнях детализации.

Пулинг слои предназначены для уменьшения пространственных размеров активаций, полученных после сверточных операций, путем объединения значений пикселей в заданных областях. Это позволяет модели быть инвариантной к небольшим трансляциям объектов на изображении и уменьшает количество параметров, что способствует предотвращению переобучения и повышению эффективности вычислений.

Полносвязанные слои обычно располагаются в конце архитектуры нейронной сети и используются для объединения высокоуровневых признаков, извлеченных предыдущими слоями, в предсказания или классификации. Эти слои обладают полным соединением со всеми активациями предыдущего слоя и представляют собой типичные слои нейронных сетей, в которых каждый нейрон связан с каждым нейроном предыдущего слоя.

Во время обучения сверточной нейронной сети параметры каждого слоя оптимизируются с использованием методов оптимизации, таких как обратное распростра-

нение ошибки и стохастический градиентный спуск, с целью минимизации заданной функции потерь. Этот процесс позволяет модели настраивать свои параметры для эффективного извлечения признаков и выполнения конкретной задачи, такой как классификация изображений или сегментация объектов.

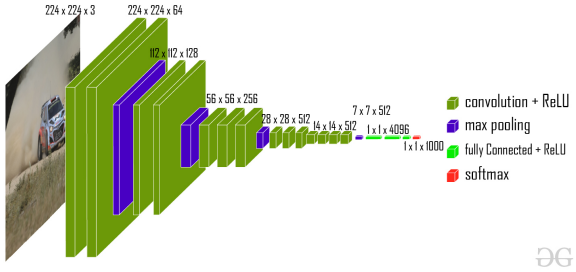

Рис. 3.3: Архитектура VGG16 [**simonyan2014very**]

Архитектуры U-Net 3.4 и ResNet 3.3 являются двумя широко используемыми моделями в области компьютерного зрения, обе из которых имеют уникальные характеристики и применяются в различных задачах.

U-Net - это архитектура, разработанная для сегментации изображений, особенно в медицинском изображении. Ее особенностью является использование симметричной структуры, включающей свертки (downsampling) для уменьшения размерности и деконволюционные слои (upsampling) для восстановления пространственного разрешения. Основное преимущество U-Net заключается в способности эффективно обрабатывать маленькие объекты и сохранять детали на всех уровнях.

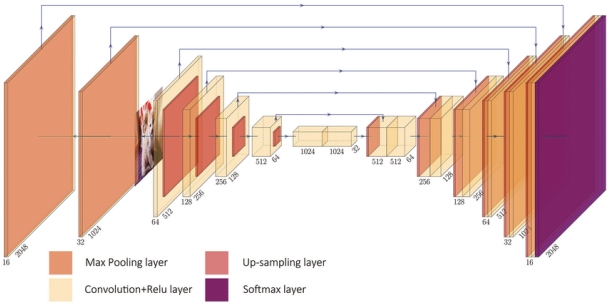

Рис. 3.4: Архитектура Unet [**ronneberger2015u**]

ResNet, с другой стороны, известен своей глубокой архитектурой с использованием блоков с пропуском (skip connections), которые обеспечивают плавное обучение глубоких сетей. Основное преимущество ResNet заключается в способности обучать глубокие модели с минимальным ухудшением производительности (проблема затухающих градиентов), благодаря использованию пропускающих соединений.

В целом, U-Net обладает преимуществами в задачах, где важна точность восстановления деталей и сегментация объектов, особенно на малых объектах. ResNet, напротив, лучше подходит для задач классификации и детекции объектов в больших наборах данных, благодаря способности обучать глубокие модели с высокой стабильностью.

Аугментация

Методы аугментации изображений в компьютерном зрении представляют собой техники, используемые для увеличения размера и разнообразия тренировочного набора данных путем применения различных преобразований к изображениям. Целью аугментации является создание дополнительных вариаций изображений, что помогает улучшить обобщающую способность моделей машинного обучения и уменьшить риск переобучения.

Основные методы аугментации включают в себя изменение размера изображения (путем масштабирования), повороты, отражения, изменение яркости, контраста и насыщенности цветов, а также добавление шума или размытия. Дополнительно, могут применяться специфические трансформации, такие как сдвиги, обрезки или изменение геометрии изображения.

Применение методов аугментации позволяет модели машинного обучения обучаться на более разнообразных данных, что способствует повышению их устойчивости к различным условиям и изменениям в данных во время работы. Кроме того, аугментация может помочь справиться с проблемой несбалансированных классов и улучшить обобщающую способность моделей.

Эффективное использование методов аугментации требует тщательного анализа особенностей конкретной задачи и выбора соответствующих трансформаций, которые помогут улучшить качество модели без искажения смысла изображений. Кроме того, важно проводить проверку и оценку результатов аугментации с целью избежать нежелательных эффектов и обеспечить сбалансированное улучшение качества моделей компьютерного зрения.

Детекция объектов

3.4.2 Выделение объектов

[kirillov2023segment]

Модель YOLO (You Only Look Once) представляет собой популярную архитектуру для обнаружения объектов на изображениях. Ее основной идеей является выполнение обнаружения объектов и классификации в одной сети, что делает ее быстрой и эффективной.

Порядок работы модели YOLO начинается с входного изображения, которое подается на вход нейронной сети. Затем изображение проходит через сверточные слои, которые извлекают признаки из изображения на различных уровнях абстракции.

Далее, полученные признаки пропускаются через сверточные слои, которые прогнозируют боксы (ограничивающие рамки) для объектов и их вероятности принадлежности к различным классам. Эти сверточные слои производят прогнозы на основе якорей (anchors), которые представляют разные размеры и соотношения сторон боксов.

После этого выполняется пост-обработка, включающая подавление неоднородных предсказаний (non-maximum suppression), чтобы получить финальные прогнозы объектов. Этот шаг удаляет лишние дубликаты и уверенно прогнозирует объекты с наибольшей уверенностью (confidence).

В результате работы модели YOLO получается набор боксов с классами и оценками уверенности, представляющих объекты, найденные на изображении. Эта информация может быть использована для обнаружения объектов и их классификации в реальном времени.

Выделение объектов

Алгоритм bounding box представляет собой метод в области компьютерного зрения, направленный на выделение прямоугольной области, охватывающей объекты на изображении. Основная цель алгоритма состоит в определении минимального прямоугольника, который содержит объект, сохраняя при этом минимальные потери информации.

Принцип работы алгоритма bounding box заключается в следующих шагах:

1. Подготовка изображения: Изображение предварительно обрабатывается для улучшения качества и подготовки к дальнейшему анализу. 2. Обнаружение объектов: Проводится анализ изображения с целью выявления интересующих областей, используя различные методы, такие как выделение краев, сегментация или классификация. 3. Вычисление ограничивающих рамок: Для каждого обнаруженного объекта определяется минимальный прямоугольник, который полностью охватывает его. 4. Визуализация результатов: Ограничивающие рамки визуализируются на изображении для дальнейшего анализа или использования.

Математически алгоритм bounding box может быть описан следующим образом:

Пусть $P = \{p_1, p_2, ..., p_n\}$ — множество точек, описывающих объект на изображении.

Координаты верхнего левого угла ограничивающей рамки (x_{min}, y_{min}) и координаты нижнего правого угла (x_{max}, y_{max}) определяются как:

$$x_{\min} = \min_{p \in P}(p_x),$$

$$y_{\min} = \min_{p \in P}(p_y),$$

$$x_{\max} = \max_{p \in P}(p_x),$$

$$y_{\max} = \max_{p \in P}(p_y).$$

Таким образом, алгоритм bounding box позволяет эффективно выделять и описывать объекты на изображении, что является важным инструментом в области компьютерного зрения и обработки изображений.

Распознание символов

Оптическое распознавание символов (OCR) представляет собой процесс автоматического преобразования текста, представленного в виде изображения или сканированного документа, в электронный текстовый формат. Термин "оптическое"в данном контексте указывает на использование оптических средств, таких как камеры или сканеры, для захвата изображений символов с физических носителей, например, бумаги.

Процесс OCR включает в себя несколько этапов, начиная с захвата изображения и заканчивая распознаванием символов и созданием электронного текста. Первоначально изображение документа подвергается предварительной обработке, такой как удаление шума или коррекция искажений. Затем происходит сегментация изображения, то есть разделение его на отдельные символы или группы символов.

Далее, при помощи алгоритмов распознавания, включающих методы машинного обучения и компьютерного зрения, символы на изображении анализируются и сопоставляются с соответствующими символами из набора знаков. Этот этап включает в себя распознавание формы символов, их контекста и других характеристик, что позволяет определить, какие символы были изображены на сканированном документе.

В завершение, распознанные символы объединяются в слова, предложения и абзацы, формируя полноценный текстовый документ. Точность и эффективность процесса OCR зависят от качества изображения, используемых алгоритмов распознавания, а также от языка и структуры текста. В современных приложениях OCR широко используются в различных областях, включая сканирование документов, распознавание номеров автомобильных номеров, оптическое чтение рукописных текстов и другие приложения, где требуется автоматическое извлечение текста из изображений.

Глава 4

Описание работы

Работа по созданию ассистента выполнялась поэтапно.

Исходным этапом работы являлось создание корпуса педагогических задач, извлеченных из открытых источников российских учебников. Процесс сбора данных осуществлялся при помощи технологий оптического распознавания символов (OCR), включая методы, разработанные в рамках данного исследования.

В главе приведено описание поставленных по трем направлениям. Сбор данных, генерация текста задачи и сопровождающей иллюстрации.

4.1 Подготовка данных для обучения

Цель этапа составить два набора данных для обучения: параллельный корпус изображение-текст и наиболее полное описание задачи

4.1.1 Разметка изображений

В состав открытого тренировочного набора входит более десятка тысяч аннотированных изображений Результат моделирования предоставлены на открытых ресурсах[1]

Разметка

Данные были собраны из открытых источников [**libmipt**][**mathedu**].

Для получения обучающей выборки была проведена разметка части датасета. Каждое изображение включает в себя текстовую информацию, а также различные чертежи и формулы, характерные для данной области знаний.

Процесс разметки включал создание аннотаций для каждого изображения, а именно выделение границ объектов, таких как текстовые блоки, формулы и черте-

[1] `https://github.com/NMashalov/Generative-modeling-appliance-for-creating-educational-tasks` и `https://huggingface.co/datasets/NMashalov/task_illustrations_dataset`

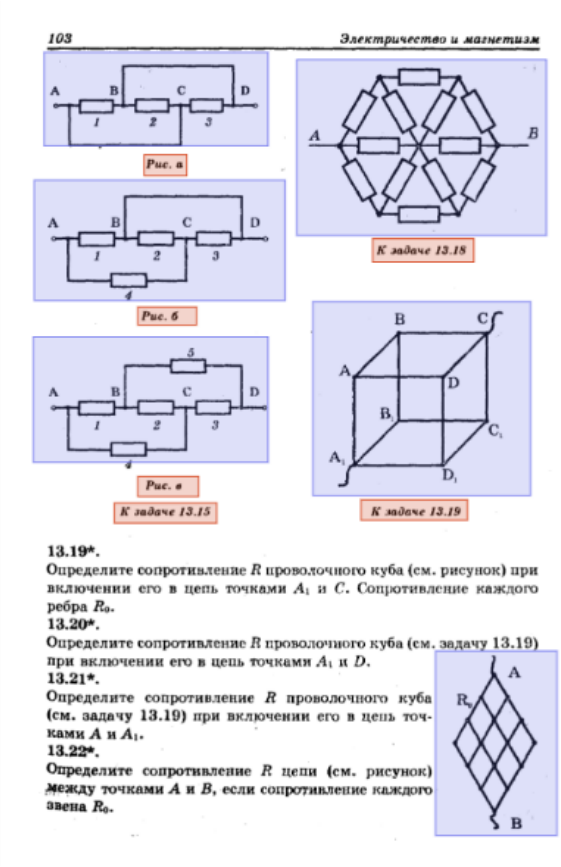

Рис. 4.1: Пример аннотированной иллюстрации из книги Генденштейн, Кирик, Гельфгат: 1001 задача по физике

жи. Этот процесс требовал точности и внимательности для корректного определения границ объектов на изображении и их соответствия с аннотациями.

Для расширения датасета и обеспечения его разнообразия была применена аугментация данных. Применялись повороты, масштабирование, изменение освещения и отражение, позволили создать дополнительные вариации входных данных. Это способствовало увеличению разнообразия обучающей выборки и повышению устойчивости модели к различным вариациям данных, что важно для обеспечения ее эффективности в реальных условиях различной разметки страницы.

Обучение нейросети для завершения разметки

Для обучения на полученных данных была использована нейронная сеть YOLO. Эта архитектура нейронной сети имеет способность эффективно дообучаться на небольших выборках данных, что позволяет достигать удовлетворительных результатов.

Для ситуаций, где число аннотаций и число изображений на изображении не совпадало, применялся алгоритм на двудольном графе, направленный на максимизацию числа пар.

4.1.2 Распознание текста

Для распознавания текста на изображениях была использована модель Nougat [**blecher2023nougat**] являющаяся адаптацией идеи представленной в статье Donut [**kim2022ocr**]. В отличие от других открытых решений, данная модель обладает способностью распознавать не только естественный язык, но и формульные выражения. Для адаптации модели к работе с русским языком потребовалось провести дополнительное дообучение.

Для обучения модели был использован сервис MathPix, предоставляющий высокое качество распознавания математических выражений. Размер датасета составил около 500 изображений. Результаты распознавания были представлены в формате Markdown.

- Был замен словарь токенизатор модели

4.2 Создание иллюстрации

Для обучения на парах текст изображение была использована модель StableDiffusion[**rombac**

Для обучения использовался низкоранговый адаптер Lora.

Изображение генерируется не в размере 512x512, а 128x128. А после трасируется для повышения разрешения

FID не объективен поскольку работает с черно-белым, но он очень высок порядка 1.5

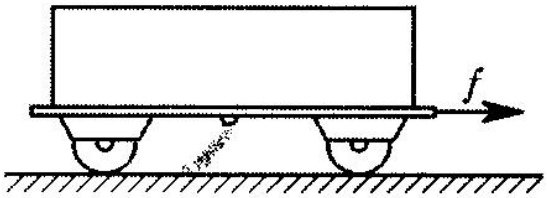

Рис. 4.2: Результат по запросу "Графическое изображение тележки"

4.3 Создание ассистента

Глава 5

Заключение

5.1 Заключение

Работа представляет

5.1.1 Благодарности

Автор благодарит кафедру инновационной педагогики за предоставление консультационной информации в области права и . Отдельная благодарность моему научному руководителю Щербакову Дмитрию Евгеньевичу за возможность работы в актуальной и современной тематики генеративного моделирования

Список литературы для корпуса

- Колмогоров А. Н. и др. Алгебра и начала анализа : учебное пособие для 10-го класса средней школы
- Алгебра и начала анализа, 9 класс / под ред. А. Н. Колмогорова. — 1975
- Перельман Я. И. Живая геометрия теория и задачи. — 1930
- Извольский Н. А. Геометрия на плоскости (планиметрия). — 1
- Беляева Э. С., Монахов В. М. Экстремальные задачи. — 1977
- Александров А. Д. и др. Геометрия пробный учебник для 9—10 классов средней школы. — 1983
- Гарднер М. Есть идея! — 1982
- Шень А. Х. Геометрия в задачах. — 2017
- Дорф П. Я. Наглядные пособия по математике и методика их применения в средней школе. — 1960
- Лиман М. М. Практические задачи по геометрии. — 1961

- Скопец З. А., Жаров В. А. Задачи и теоремы по геометрии (планиметрия). — 1962
- Клопский В. М. и др. Геометрия : учебное пособие для 9—10 классов / В. М. Клопский, З. А. Скопец, М. И. Ягодовский ... 1978. — 256 с.
- Овчинкин В.А. Сборник задач по общему курсу физики. Часть 1. Механика. Термодинамика и молекулярная физика - 2005
- Овчинкин В.А., Раевский А.О., Ципенюк Ю.М. Сборник задач по общему курсу физики. Часть 3. Атомная и ядерная физика. Строение вещества 2009г
- Козел С.М., Лейман В.Г., Локшин Г.Р., Овчинкин В.А., Прут Э.В. Сборник задач по общему курсу физики. Часть 2. Электричество и магнетизм. Оптика 2000 г
- Беклемишев Д.В. Курс аналитической геометрии и линейной алгебры 2005г
- Генденштейн, Кирик, Гельфгат: 1001 задача по физике с ответами, указаниями, решениями