

Министерство образования Новосибирской области
ГБПОУ НСО «Новосибирский авиационный технический колледж имени Б.С. Галушака»

РАЗРАБОТКА GAN НЕЙРОСЕТИ

Пояснительная записка к курсовому проекту

ПМ.02 Осуществление интеграции программных модулей

МДК.02.01 Технология разработки программного обеспечения

НАТКиГ.202000.010.000ПЗ

Разработал:
студент группы ПР-21.106
Шестаков И.В.

2023

Содержание

Введение.....	3
1 Описание предметной области	4
2 Выбор технологии, языка и среды программирования	7
3 Анализ и уточнение требований к программному продукту	8
3.1 Выбор методов и разработка основных алгоритмов решения задачи.....	8
4 Разработка структурной схемы программного продукта.....	12
4.1 Описание используемых процедур и библиотечных функций	12
4.2 Спецификация программы	13
5 Проектирование интерфейса пользователя	14
5.1 Разработка форм ввода-вывода информации.....	14
Заключение	15
Литература	16
Приложение А Техническое задание	17
Приложение В Блок схемы.....	29

					НАТКиГ.201200.010.000ПЗ							
Изм.	Лист	№ докум	Подпись	Дата								
Разраб		Шестаков И.В.			Разработка GAN нейросети				Литера	Лист	Листов	
Пров		Чекушкина О. О.							у		2	50
									ПР-21.106			
Н. Контр		Тышкевич Е. В.										
Утв		Тышкевич Е. В.										

Введение

Нейронные сети вдохновлены биологической нейронной системой человека. Нейронные сети состоят из искусственных нейронов, которые объединяются в слои и обрабатывают данные, выполняя различные вычислительные задачи. Они используются для решения широкого спектра задач, включая распознавание образов, классификацию, регрессию, генерацию данных и многое другое.

В последние годы диффузионные модели приобрели особую актуальность в области искусственного интеллекта. Этот подход, основанный на принципах диффузии, нашел широкое применение в решении сложных задач, связанных с обработкой информации, анализом данных и прогнозированием. С ростом количества пользовательских данных в социальных медиа, эти модели могут быть использованы для моделирования динамики распространения информации, влияния мнений и предсказания трендов в обществе.

Целью курсового проекта является создания генерирующей изображения нейросети.

Задачами курсового проекта в связи с указанной целью являются:

- изучение предметной области;
- рассмотрение проекта с точки зрения пользователя для выявления функций приложения;
- написание кода нейросети;
- тестирование полученного продукта.

Объект исследования – генерирующая нейросеть.

1 Описание предметной области

Диффузионная модель базируется на принципах диффузии, представляя собой метод анализа и моделирования распространения информации, изменений или воздействия в сложных системах. В сущности, эти модели описывают, как новые идеи, технологии или явления распространяются внутри определенной области или среды.

Основной элемент диффузионной модели — это сеть узлов, представляющих собой индивидуальные сущности, такие как люди, компьютеры, или другие агенты. Связи между узлами определяют взаимодействие и передачу информации.

Пользователь, генерирующий изображение может:

1. Ввод данных:

- Подготовка входных данных (описание желаемого результата);
- Уточнение положительных и отрицательных запросов.

2. Настройка параметров генерации:

- Выбор размера генерируемого изображения;
- Задание количества генерируемых изображений;
- Настройка дополнительных параметров.

3. Запуск генерации:

- Отправка входных данных в модель;
- Получение на выходе сгенерированных нейросетью изображений.

В процессе обучения диффузионной модели для генерации изображений, модель учится захватывать сложные зависимости и структуры в данных. Она выявляет контекстуальные зависимости между пикселями, изучает распределение признаков в обучающем датасете и настраивает свои параметры с использованием стохастического градиентного спуска. Модель также обучается управлять градиентными изменениями в процессе диффузии, что влияет на конечный результат генерации изображения.

Однако, в ходе обучения могут возникнуть различные проблемы. В частности, модель может столкнуться с проблемой переобучения, когда она слишком хорошо подстраивается под обучающий датасет, теряя способность к обобщению на новые данные. Также, возможны трудности с управлением шумом и деталями в генерируемых изображениях. Баланс между детализацией и сохранением структуры представляет собой сложную задачу.

Процесс обучения нейросети, можно разделить на несколько ключевых этапов:

1. Подготовка данных:

– В начале обучения выбирается датасет, содержащий изображения, которые модель будет использовать в процессе обучения. Этот датасет разнообразен и представляет собой множество изображений, которые модель должна научиться воспроизводить.

2. Инициализация Модели:

– Диффузионная модель инициализируется с определенными параметрами и структурой. Эта структура обычно включает в себя узлы и связи, которые моделируют внутренние зависимости в данных.

3. Распространение информации:

– Процесс обучения основан на идее "распространения" информации внутри модели. Модель последовательно применяет диффузионные шаги, где каждый шаг представляет собой изменение состояния модели в направлении, учитывающем структуру данных.

4. Оптимизация параметров:

– Во время распространения модель настраивает свои параметры так, чтобы минимизировать разницу между сгенерированными изображениями и изображениями из датасета.

Обучение модели требует большого набора реальных изображений в качестве примеров. Модель изучает контекстуальные зависимости между пикселями в изображениях. Это позволяет ей генерировать структурно согласованные и реалистичные изображения. Чем больше и разнообразнее

тренировочные данные, тем лучше модели сможет научиться генерировать реалистичные изображения.

Диффузионные модели могут использоваться для генерации фотографических изображений людей, животных, пейзажей и других объектов. Они позволяют создавать высококачественные изображения, неотличимые от реальных фотографий.

Они открывают большие возможности для синтеза фотореалистичных изображений, которые могут использоваться в различных областях - от развлечений до науки.

В заключение, диффузионные модели в области генерации изображений открывают перспективные возможности для совершенствования синтеза визуальных контентов. Этот подход, опирающийся на принципы диффузии, предоставляет уникальные инструменты для моделирования сложных зависимостей в данных и эффективной генерации качественных изображений.

					НАТКиГ.202000.010.000ПЗ	Лист
						6
Изм.	Лист	№ докум	Подпись	Дата		

2 Выбор технологии, языка и среды программирования

Средой программирования выбрана программа Visual Studio Code, так как она предоставляет легкую и интуитивно понятную среду разработки. Её установка и настройка занимают минимум времени, что важно для быстрого старта проекта. К тому же VS Code обладает отличной поддержкой языка программирования Python. Интеграция с Python расширением обеспечивает возможность работы с виртуальными средами, авто дополнение кода, проверку синтаксиса и быстрое исправление ошибок.

Языком программирования выбран Python, так как он является одним из наиболее популярных языков программирования в области машинного обучения и исследовательских задач. Это обеспечивает доступ к богатому экосистему библиотек и инструментов, специально созданных для работы с данными и моделями машинного обучения. Синтаксис Python является простым и читаемым, что облегчает понимание и сопровождение кода. Это особенно важно в научном и исследовательском программировании, где ясность и понятность кода приобретают ключевое значение.

Python обладает обширной библиотекой для работы с данными, научных вычислений и машинного обучения. Библиотеки, такие как NumPy, Pandas, TensorFlow и Keras, предоставляют мощные инструменты для создания и обучения моделей, а также обработки данных.

Выбор технологий обоснован стремлением использовать мощные фреймворки TensorFlow и Keras для создания и обучения моделей глубокого обучения, интеграцией с предварительно обученными моделями текста TensorFlow Hub и TensorFlow Text, а также применением метода гауссовской диффузии для генерации изображений.

3 Анализ и уточнение требований к программному продукту

3.1 Выбор методов и разработка основных алгоритмов решения задачи

GENERATING AI включает в себя реализацию гауссовской диффузии, модели для обработки текста с использованием BERT, а также диффузионной модели для генерации изображений.

Процесс диффузии представляет собой постепенное преобразование структурированного сигнала (например, изображения) в шум. Путем моделирования диффузии можно генерировать зашумленные изображения на основе обучающих изображений и обучать нейронную сеть устранению шума. С использованием обученной сети можно воспроизводить обратный процесс диффузии, то есть создавать изображения из шума.

Когда уровень шума достаточно низок, можно использовать условные гауссианы для установления переходов цепочки дискретизации в прямом процессе. Простая параметризация прямого процесса является результатом соединения этого знания с предположением Маркова:

$$q(x_{1:T}|x_0) := \prod_{t=1}^T q(x_t|x_{t-1}), q(x_t|x_{t-1}) := N(x_t; \mu_{t-1}, \sigma_t^2), \quad (1)$$

где $x_{1:T}$ — обозначает последовательность случайных переменных $\{x_1, x_2, \dots, x_T\}$;

x_0 — начальное состояние;

$q(x_{1:T} | x_0)$ — Это условное распределение всех состояний от x_1 до x_T при условии начального состояния x_0 ;

$\prod_{t=1}^T$ — произведение по времени от $t=1$ до T ;

$q(x_t|x_{t-1})$ — условное распределение состояния x_t при условии предыдущего состояния x_{t-1} ;

$N(x_t; \mu_{t-1}, \sigma_t^2)$ — гауссово распределение для переменной x_t . Здесь N обозначает нормальное (гауссово) распределение, аргументы функции — среднее (μ_{t-1}), стандартное отклонение (σ_t^2), где μ — параметр, а β — некоторый коэффициент.

Класс `GaussianDiffusion` представляет собой утилиту для реализации гауссовской диффузии. Этот метод используется для генерации шума и диффузии входных данных. Гауссовская диффузия является процессом, который вводит случайные шумы в изображения, постепенно увеличивая их дисперсию с течением времени.

После прямой диффузии начинается обратный процесс. Модель учится обращать этот процесс распространения во время обучения, чтобы получать свежие данные. Модель изучает совместное распределение как $(x_0:T)$ результат исходного уравнения чистого гауссовского шума где обнаружены зависящие от времени параметры гауссовых переходов. Так же учитывается формулировка Маркова, согласно которой данное распределение обратного диффузионного перехода зависит исключительно от предыдущего временного шага.

Для обработки текста используется модель BERT (Bidirectional Encoder Representations from Transformers). Эта модель способна создавать контекстные эмбединги для текстовых данных, улавливая их семантические свойства и взаимосвязи между словами. Эмбединги текста, полученные с использованием BERT, используются в дальнейшем в процессе генерации изображений. BERT обладает трансформерной архитектурой, которая позволяет модели эффективно обрабатывать последовательности данных, сохраняя контекстуальные зависимости между словами. На листинге 1 приведен код для преобразования запроса в эмбединг.

Листинг 1 – Преобразование текста в эмбединг

```
def process_text(text_batch):  
    text_preprocessed = bert_preprocess_model(text_batch)  
    bert_results = bert_model(text_preprocessed)  
    return bert_results["pooled_output"]
```

Таким образом, BERT преобразует текстовую информацию в числовое представление, сохраняя контекстуальные и семантические зависимости между словами. Полученные эмбединги затем используются в процессе генерации изображений с учетом текстового контекста.

Диффузионная модель представлена классом `DiffusionModel`. Этот класс использует нейросетевую модель (`network`), обученную для генерации изображений, и предоставляет методы для генерации изображений с использованием гауссовской диффузии.

Приведенный код создает экземпляр диффузионной модели, использует ее для генерации изображений на основе текстового описания ("Clear room") и визуализирует результаты. Экстраполяция шума (`ex_rate`) позволяет управлять уровнем вариации в генерируемых изображениях.

Обучение модели происходит в цикле, включающем следующие шаги, реализованные в коде:

- Выборка Данных: извлечение батчей из обучающего набора данных, представляющих изображения и соответствующие текстовые описания;
- Случайная Генерация Временных Шагов: используется случайная выборка временных шагов для применения гауссовской диффузии к изображениям;
- Обновление Весов ЕМА Модели: после каждого шага обучения веса модели обновляются в соответствии с бегущим средним (ЕМА) для стабилизации результатов.

Обратная марковская модель, которая максимизирует вероятность обучающих данных, используется для обучения диффузионной модели. Практически говоря, обучение аналогично уменьшению вариационной верхней границы отрицательной логарифмической вероятности:

$$E [-\log p_{\theta}(x_0)] \leq E q - \log p_{\theta}(x_{0:T}) q(x_{1:T} | x_0) = E q - \log p(x_T) - \sum_{t=1}^T \log p_{\theta}(x_t - 1 | x_t) q(x_t | x_t - 1) = : L, \quad (2)$$

Где $E [-\log p_{\theta}(x_0)]$ – ожидание отрицательного логарифма вероятности начального состояния x_0 по распределению, заданному моделью с параметрами θ .

$E q - \log p_{\theta}(x_{0:T}) q(x_{1:T} | x_0)$ – вариационная верхняя граница для отрицательной логарифмической вероятности всей последовательности состояний от x_0 до x_T .

$E q - \log p(x_T) - X_t \geq 1 \log p(\theta(x_{t-1}|x_t)) q(x_t|x_{t-1})$ – разложение вариационной верхней границы. Первое слагаемое относится к вероятности конечного состояния x_T , а второе слагаемое представляет собой сумму логарифмов условных вероятностей переходов между состояниями x_t и x_{t-1} .

L – вариационная нижняя граница для отрицательной логарифмической вероятности, которую нужно максимизировать в процессе обучения.

Используя обученную модель, предсказывается зашумленное изображение на основе случайного шума:

$$x_t = x_{t-1} + \sqrt{\beta_t} * noise_t, \quad (3)$$

где x_t – предсказанное зашумленное изображение;

x_{t-1} – предыдущее зашумленное изображение;

β_t – параметр функции, вычисленный на этапе обучения;

$noise_t$ – случайный шум.

Обучение модели и выборка из неё можно представить в виде алгоритмов на рисунке 1:

Algorithm 1 Training	Algorithm 2 Sampling
1: repeat 2: $\mathbf{x}_0 \sim q(\mathbf{x}_0)$ 3: $t \sim \text{Uniform}(\{1, \dots, T\})$ 4: $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 5: Take gradient descent step on $\nabla_{\theta} \ \epsilon - \epsilon_{\theta}(\sqrt{\alpha_t} \mathbf{x}_0 + \sqrt{1 - \alpha_t} \epsilon, t)\ ^2$ 6: until converged	1: $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ 2: for $t = T, \dots, 1$ do 3: $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ if $t > 1$, else $\mathbf{z} = \mathbf{0}$ 4: $\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{1 - \alpha_t}{\sqrt{1 - \alpha_t}} \epsilon_{\theta}(\mathbf{x}_t, t) \right) + \sigma_t \mathbf{z}$ 5: end for 6: return \mathbf{x}_0

Рисунок 1 – алгоритмы обучения и выборки модели

Алгоритм генерации изображения и процесса обучения модели можно представить в виде блок-схем, изображенных в приложении Б на рисунке Б.1.

4 Разработка структурной схемы программного продукта

4.1 Описание используемых процедур и библиотечных функций

Код включает в себя разнообразные процедуры и классы, предназначенные для выполнения различных задач. Среди ключевых функций кода можно выделить процедуры, ответственные за загрузку моделей, определение классов, связанных с гауссовской диффузией, а также модели диффузии. Кроме того, реализованы методы, направленные на генерацию изображений и их последующую визуализацию. Описание процедур представлено в таблице 1.

Таблица 1 – Описание процедур и классов

Процедуры и классы	Описание
@app.route('/') index()	Декоратор для маршрута главной страницы Функция, возвращающая шаблон index.html
@app.route('/generate_images', methods=['POST']) generate_images()	Декоратор для маршрута обработки POST-запросов генерации изображений Функция, обрабатывающая POST-запрос, генерирующая изображения и возвращающая их в формате base64
__init__(self, beta_start=1e-4, beta_end=0.02, timesteps=1000, clip_min=-1.0, clip_max=1.0)	Инициализация параметров гауссовской диффузии
q_mean_variance(self, x_start, t)	Расчет среднего значения и дисперсии на текущем временном шаге
q_sample(self, x_start, t, noise)	Вычисление диффузионных образцов на временном шаге t
predict_start_from_noise(self, x_t, t, noise)	Предсказание начального состояния из шума
q_posterior(self, x_start, x_t, t)	Вычисление среднего значения и дисперсии апостериорной диффузии
p_mean_variance(self, pred_noise, x, t, clip_denoised=True)	Вычисление среднего значения, дисперсии и логарифма дисперсии предсказанного шума
p_sample(self, pred_noise, x, t, clip_denoised=True)	Генерация выборки из модели диффузии
__init__(self, network, ema_network, timesteps, gdf_util, ema=0.999)	Инициализация модели диффузии
generate_images(self, num_images, annotation, negative_prompt, ex_rate, size)	Генерация изображений с использованием гауссовской диффузии
plot_images(self, epoch, logs, num_rows, num_cols, figsize, annotation, ex_rate, size)	Отображение сгенерированных изображений

Процедуры используют различные библиотеки для обработки текста, загрузки моделей, генерации изображений и других задач. В таблице 2 представлено описание библиотечных функций, используемых в коде.

Таблица 2 – Описание библиотечных функций

Библиотечные функции	Описание
tf.keras.models.load_model(model_path)	Загрузка модели из файла
tf.random.normal(shape, dtype)	Генерация тензора с случайными значениями из нормального распределения
cv2.imencode('.png', image)	Кодирование изображения в формат PNG
np.linspace(start, end, num, dtype)	Генерация массива чисел с постоянным шагом между начальным и конечным значениями
np.cumprod(array, axis)	Вычисление кумулятивного произведения элементов массива
plt.subplots(num_rows, num_cols, figsize)	Создание сетки для отображения нескольких графиков
ax.imshow(image)	Отображение изображения на графике
base64.b64encode(buffer)	Кодирование бинарных данных в формат base64
webbrowser.open(url)	Открытие URL в веб-браузере
request.form.get(key, default)	Получение значения из формы POST-запроса
os.path.dirname(os.path.realpath(file))	Получение каталога текущего исполняемого файла
hub.KerasLayer(tfhub_handle)	Создание слоя Keras из модели TensorFlow Hub

4.2 Спецификация программы

Файлы программного продукта представлены в таблице 3.

Таблица 3 – Файлы, входящие в программный продукт

Наименование	Назначение	Примечание
dev/templates/index.html	Файл разметки WEB-сайта	Файл html и css разметки, отображаемой на странице сайта, а также для хранения POST-запросов, привязанных к определенным объектам
dev/main.py	Файл кода для генерации изображений и отправки запросов на сайт	С помощью него запускается сайт и принимаются POST-запросы на генерацию изображения, а также для его вывода
dev/model.h5	Файл с сохраненными данными обученной модели	Хранит в себе веса обученной модели, необходимых для генерации изображений
Training/Training.ipynb	Файл кода для обучения модели	Реализация обучения диффузионной модели, сохраняемой в .h5 файле

5 Проектирование интерфейса пользователя

5.1 Разработка форм ввода-вывода информации

Курсовой проект представляет собой WEB-сайт для генерации изображений с использованием пользовательских параметров. На рисунке 2 изображен дизайн сайта, показывающий его интерфейс и функционал.

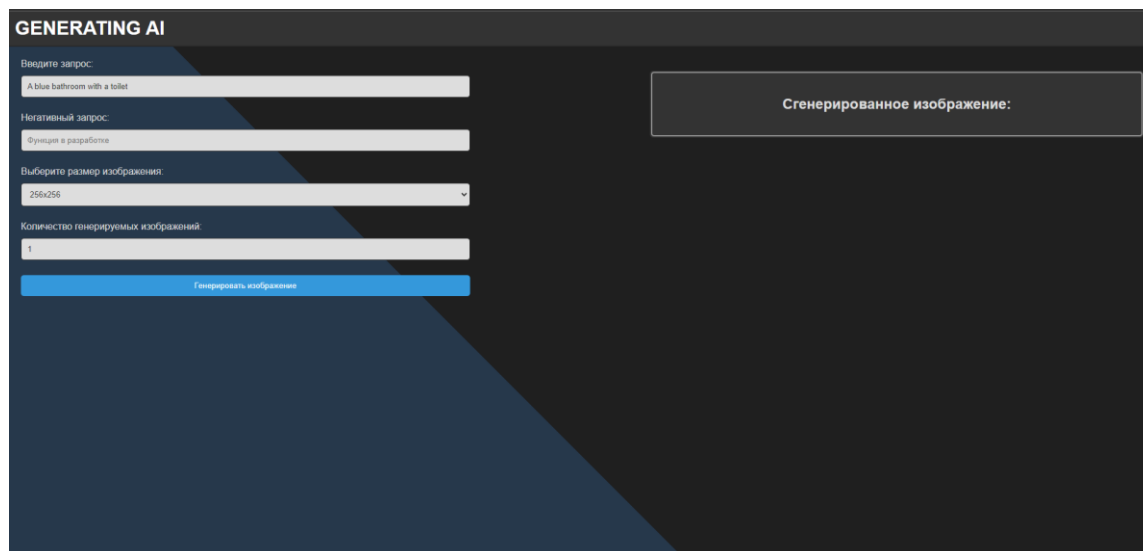


Рисунок 2 – дизайн сайта

Визуально выделены ключевые элементы: поле ввода запроса, где пользователи могут указать текстовый запрос или ключевые слова, а также интерфейс выбора размера изображения.

Дополнительно, на сайте предусмотрена возможность выбора количества изображений, которые пользователь хочет сгенерировать с использованием заданных параметров. После ввода всех необходимых данных, пользователь может нажать кнопку запуска генерации изображения.

Важным элементом функционала является поле вывода сгенерированных изображений, где пользователь может увидеть результаты в виде изображений, созданных в соответствии с их запросом, выбранными размерами и указанным количеством.

Заключение

В результате разработки диффузионной модели, основанной на принципах диффузии была реализована основная функция — генерация изображения, позволяющая создавать уникальные изображения на основе запросов. А также дополнительные функции — выбор размера изображения и их количество при генерации.

Интерфейс WEB-сайта разработан с учетом удобства использования и минимального количества действий для достижения пользовательской цели. Страница сайта не перегружена лишней информацией и содержит только основной функционал, необходимый для корректной работы сайта.

Процесс обучения диффузионной модели включает в себя несколько ключевых этапов, начиная с подготовки разнообразного датасета. Инициализация модели, распространение информации и оптимизация параметров осуществляются в ходе обучения, где модель учится захватывать сложные зависимости и структуры в данных.

В заключение, диффузионные модели в области генерации изображений представляют собой перспективный подход, обеспечивающий уникальные инструменты для моделирования сложных зависимостей в данных и эффективной генерации качественных визуальных контентов. Этот метод, основанный на принципах диффузии, открывает новые горизонты для усовершенствования синтеза визуальных материалов в различных областях.

Литература

1 Keras [Электронный ресурс]: Документация по диффузионным моделям: <https://keras.io/examples/generative/ddim/>

2 Tensorflow [Электронный ресурс]: Документация по библиотеки Tensorflow: <https://www.tensorflow.org/?hl=ru>

3 METANIT [Электронный ресурс]: Документация по языку программирования Python – Режим доступа к руководству: <https://metanit.com/python/>

4 HashDork [Электронный ресурс]: Введение в диффузионные модели: <https://hashdork.com/ru/диффузионная-модель/>

5 Unite [Электронный ресурс]: Принципы обучения диффузионных моделей: <https://www.unite.ai/ru/diffusion-models-in-ai-everything-you-need-to-know/>

Приложение А

Министерство образования Новосибирской области
ГБПОУ НСО «Новосибирский авиационный технический колледж
имени Б.С. Галуцака»

GENERATING AI

Техническое задание
НАТКиГ.202000.010.000ПЗ

Разработал:

Студент группы: ПР-21.106

___. ___.20 ____ /Шестаков И.В./

Проверила:

___. ___.20 ____ /Чекушкина О.О./

2023

					НАТКиГ.202000.010.000ПЗ	Лист
Изм.	Лист	№ докум	Подпись	Дата		17

Содержание

Введение.....	19
1 Основания для разработки	20
2 Назначения разработки.....	21
3 Исходные данные, цели и задачи	22
3.1 Требования к функциональным характеристикам	22
3.2 Цель работы	22
3.3 Решаемые задачи	22
4 Требования	23
4.1 Требования к функциональным характеристикам	23
4.1.1 Для пользователя.....	23
4.1.2 Исходные данные	23
4.1.3 Результаты	23
4.2 Требования к надёжности	23
4.3 Условия эксплуатации	23
4.4 Требования к составу и параметрам технических средств.....	24
4.5 Требования к информационной и программной совместимости.....	24
4.6 Требования к маркировке и упаковке	24
4.7 Требования к транспортированию и хранению	24
4.8 Специальные требования	24
5 Требования к программной документации	25
6 Техничко-экономические показатели	26
7 Стадии и этапы разработки	27
8 Порядок контроля и приёмки.....	28

Введение

Настоящее техническое задание распространяется на разработку нейросети с веб-интерфейсом. Основной задачей системы является генерация изображений, удовлетворяющих заданным параметрам, таким как размер изображения и количество изображений. Веб-интерфейс обеспечивает удобный способ взаимодействия пользователя с системой, позволяя ему вводить необходимые параметры и получать результат в виде сгенерированного изображения.

Наименование приложения: «GENERATING AI».

Краткая характеристика области применения: нейросеть представляет собой скрипт с веб-интерфейсом, разрабатываемой с целью автоматической генерации изображений по запросу.

Наименование темы разработки – «Разработка GAN нейросети».

					НАТКиГ.202000.010.000ПЗ	Лист
Изм.	Лист	№ докум	Подпись	Дата		19

1 Основания для разработки

Разработка программы GENERATING AI обоснована потребностью в решении задачи генерации изображений с использованием искусственных нейронных сетей. На фоне быстрого развития современных технологий и растущего внимания к искусственному интеллекту, возникает необходимость в создании эффективных средств автоматизации процессов генерации разнообразных изображений.

Главной целью разработки программы является повышение эффективности работы в различных областях, где генерация изображений играет важную роль. Программа GENERATING AI позволит автоматизировать и улучшить этот процесс, обеспечивая быстрое и высококачественное создание изображений, соответствующих заданным параметрам.

Применение искусственных нейронных сетей в генерации изображений обладает потенциалом улучшить результаты в различных областях, включая дизайн, медицину, образование, развлечения и другие. Способность программы подстраиваться под различные требования пользователей, такие как размер изображения и его количество, делает ее универсальным инструментом для разнообразных задач и отраслей.

2 Назначение разработки

Основное назначение программы GENERATING AI состоит в создании инновационного программного продукта, предоставляющего средство для автоматической генерации разнообразных данных с использованием нейросетей. Этот продукт разрабатывается с целью обеспечить пользователей различных категорий (включая исследователей, художников, авторов контента и многих других) мощным инструментом для генерации изображений, что повысит их производительность и креативные возможности в различных областях деятельности.

					НАТКиГ.202000.010.000ПЗ	Лист
Изм.	Лист	№ докум	Подпись	Дата		21

3 Исходные данные, цели и задачи

3.1 Исходные данные

Исходные данные, необходимые при разработке нейросети:

- Информация о генеративных моделях;
- Наборы данных для обучения и тестирования;
- Требования к типам генеративных моделей (например, GAN, VAE, RNN);
- Архитектура нейросети.

3.2 Цель работы

Целью работы является разработка и внедрение программы GENERATING AI, способной генерировать изображения на основе нейросетей.

3.3 Решаемые задачи

- Генерация изображений на основе заданных параметров;
- Обеспечение возможности настройки параметров генерации пользователем.

4 Требования

4.1 Требования к функциональным характеристикам

4.1.1 Для пользователя

Пользователь может:

- Загрузка исходных данных – моделей;
- Генерация изображений;
- Возможность настройки параметров генерации;
- Экспорт сгенерированных данных.

4.1.2 Исходные данные

Программа принимает на вход модель, загруженную пользователем и настройки параметров: запрос, размер изображения, количество изображений. При обучении так же используются наборы данных: изображения и текстовые описания изображений. Текстовое описание хранится в отдельном .json файле.

4.1.3 Результаты

Программа должна предоставлять сгенерированные изображения в соответствии с параметрами, заданными пользователем.

4.2 Требования к надежности

Программа должна обеспечивать надежную работу и предотвращать возможные сбои и ошибки

4.3 Условия эксплуатации

Программа должна работать на платформах Windows. При работе пользователя с разрабатываемой информационной системой не должно возникать проблем, так как система обладает понятным программным интерфейсом.

					НАТКиГ.202000.010.000ПЗ	Лист
Изм.	Лист	№ докум	Подпись	Дата		23

4.4 Требования к составу и параметрам технических средств

Требований к аппаратному обеспечению нет.

4.5 Требования к информационной и программной совместимости

Программа должна быть совместима с современными форматами изображений. Также следует обеспечить совместимость с популярными форматами экспорта данных.

4.6 Требования к маркировке и упаковке

Программа поставляется через интернет.

4.7 Требования к транспортировке и хранению

Не применимо.

4.8 Специальные требования

Программа должна быть снабжена графическим интерфейсом.

5 Требования к программной документации

Состав программной документации должен включать в себя:

- Руководство пользователя с инструкциями по установке и использованию программы;
- Описание программы;
- Пояснительная записка;
- Описание применения;
- Техническую документацию, описывающую архитектуру программы и принципы работы нейросетей.

6 Технико-экономические показатели

Технико-экономические показатели включают в себя затраты на разработку, включая исследование и обучение алгоритмов, а также подготовку данных. Необходимо учесть аппаратные затраты, такие как стоимость вычислительного оборудования и энергозатраты. Затраты на обслуживание, мониторинг и обновления также важны. Экономическая эффективность оценивается прогнозируемыми доходами и сравнением с альтернативами.

					НАТКиГ.202000.010.000ПЗ	Лист
Изм.	Лист	№ докум	Подпись	Дата		26

7 Стадии и этапы разработки

В таблице 1 указаны стадии разработки курсового проекта.

Таблица 1 – Стадии разработки

№	Название этапа	Срок, даты, %	Отчётность
1.	согласование требований и планирование	30.11.23	
2.	создание программы и тестирование	25.12.23	
3.	установка программы на целевых платформах	10.01.24	
4.	обеспечение стабильной работы программы и внесение необходимых изменений	10.02.24	

8 Порядок контроля и приёмки

Контроль качества разработки и исполнения ТЗ будет осуществляться в ходе каждого этапа разработки. Программа должна проходить тестирование на соответствие требованиям.

Защита осуществляется перед комиссией по специальности.

Срок защиты определяется руководителем КП.

					НАТКиГ.202000.010.000ПЗ	Лист
Изм.	Лист	№ докум	Подпись	Дата		28

Приложение Б

(информационное)

Блок схемы

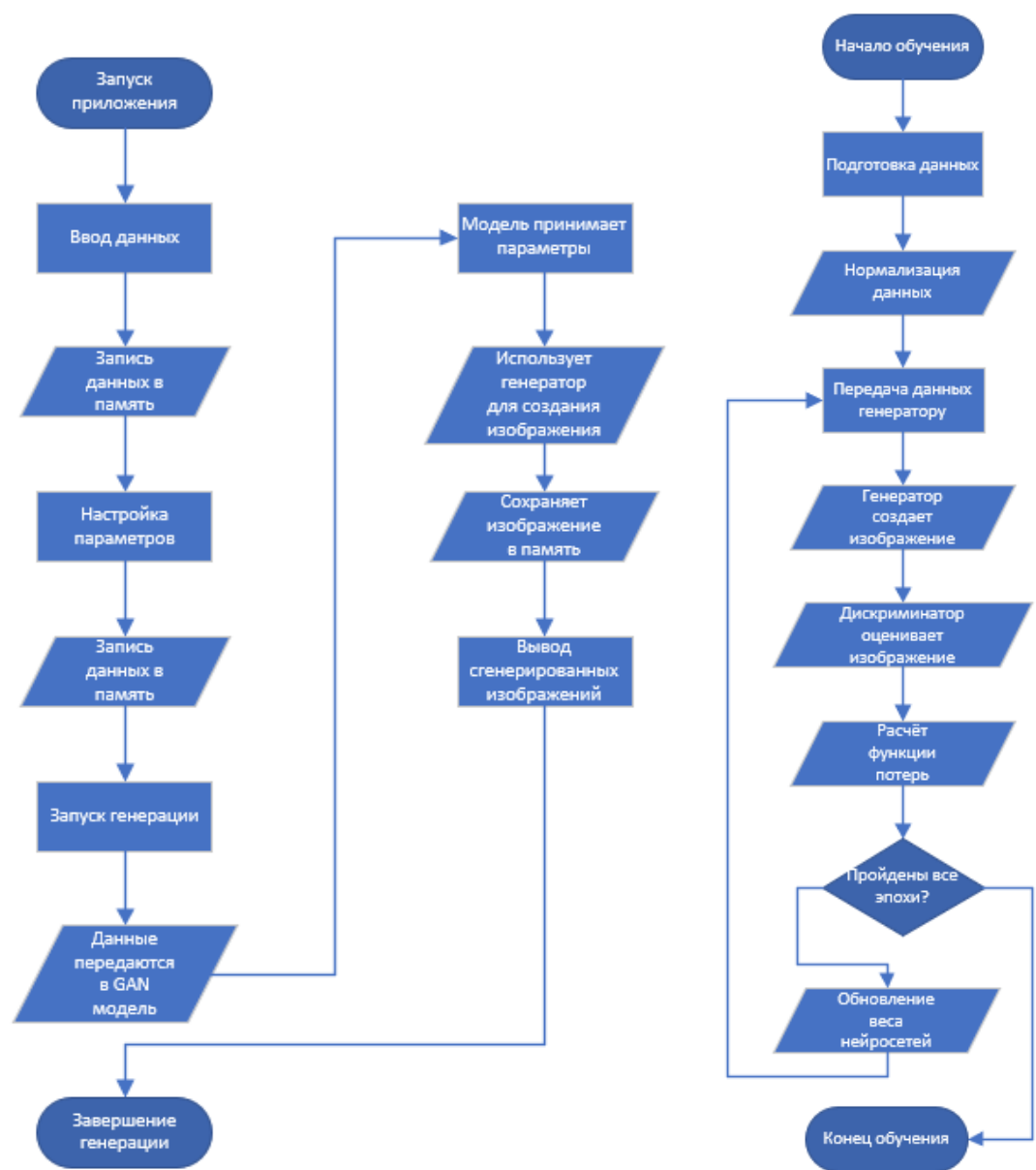


Рисунок Б.1 – блок-схемы алгоритмов

					НАТКиГ.202000.010.000ПЗ			
Изм.	Лист	№ докум.	Подпись	Дата	Блок схемы алгоритма генерации изображения и процесса обучения модели			
Разраб.	Шестаков И.В.							
Провер.	Чекушкина О.О.							
Реценз.								
Н.Контр	Тышкевич Е.В.							
Утверд.	Тышкевич Е.В.				ПР-21.106			
					Лит.	Масса	Масштаб	
							1 : 1	
					Лис	2	Листов	43