

### Список литературы:

1. Как свёрточные нейросети имитируют работу мозга [Электронный ресурс] – 2023. – Режим доступа: <https://practicum.yandex.ru/blog/svertochnye-neyronnye-seti/> . — Дата доступа: 15.11.2023.
2. Алгоритм глубокого обучения для выявления заболевания коронавирусом (COVID-19) с помощью компьютерной томографии [Электронный ресурс] – 2021. – Режим доступа: <https://peerj.com/articles/cs-345/> . — Дата доступа: 15.11.2023.
3. Свёрточные нейронные сети [Электронный ресурс] – 2023. – Режим доступа: <https://medium.com/@prathammodi001/convolutional-neural-networks-for-dummies-a-step-by-step-cnn-tutorial-e68f464d608f> . — Дата доступа: 15.11.2023.

УДК 004.8

### ПРИЛОЖЕНИЕ «ПОМОЩНИК ПО НАПИСАНИЮ ТЕКСТОВ»

*Д. С. Удом*

*«Гродненский государственный университет имени Янки Купалы», г. Гродно  
Научный руководитель: И. Б. Просвирнина, кандидат  
физико-математических наук, доцент*

Цель работы: разработать и реализовать помощник по написанию текстов; создать свою модель и обучить её написанию текста.

Интеллектуальные технологии обработки данных – это компьютеризированные технологии, используемые для обработки и анализа больших объемов данных. Они включают в себя инструменты и методы, основанные на статистическом анализе, машинном обучении и математике. Интеллектуальный анализ данных позволяет извлекать ценные знания из больших объемов данных, например, для принятия стратегических решений в бизнесе. Системы интеллектуальной обработки данных могут использоваться в различных областях, таких как аудит, авиация, банковское дело и другие. Они могут включать в себя технологии обработки естественного языка (NLP), которые позволяют извлекать информацию из объемных неструктурированных текстов.

Генерация текста с использованием искусственного интеллекта (ИИ) представляет собой процесс создания текстового контента с помощью алгоритмов машинного обучения и нейронных сетей. Эти технологии позволяют быстро и эффективно генерировать уникальный контент для различных целей, таких как написание статей, создание маркетинговых материалов и научных публикаций. Инструменты генерации текста на основе ИИ предлагают следующие преимущества:

1) Сокращение времени на создание контента и повышение эффективности работы.

2) Возможность создания уникальных текстов, трудно отличимых от текстов, созданных человеком.

3) Генерация большого объема уникального контента для использования в различных целях.

4) Поддержка нескольких языков и создание аудио контента с естественным голосом.

Использование инструментов генерации текста на основе ИИ может быть полезным для тех, кто ищет быстрый и эффективный способ получить качественный контент без необходимости владения специальными навыками программирования.

Такие технологии могут быть полезны в различных областях, включая маркетинг, образование, научные исследования и другие. Однако, при использовании сгенерированного контента важно учитывать его качество и соответствие целям и задачам организации или проекта.

Рассмотрим, как разработать и реализовать помощник по написанию текстов. Создадим свою модель и обучим её для написания текста. Данный алгоритм и модель обучения реализовывается с помощью языка программирования Python 3 в среде разработки Google Colab.

- Необходимо импортировать нужные библиотеки и подготовить данные для нейронной сети.

- Необходимые библиотеки: Counter, torch, numpy.

Токенизируем текст, превращаем его в индексы. Достаем все уникальные символы: буквы, пробелы, знаки препинания в тексте и каждому символу присваиваем число.

Создаем прямой и обратный словари (индексы в символы). Проходим по всему тексту и превращаем с помощью составленного словаря каждый символ в индекс. Генерируем из последовательности сгенерированных индексов батчи для обучения нейронной сети. Достаем несколько случайных строк из текста фиксированной длины. Генерируем сразу обучающую выборку и цель (таргет) для нее.

Реализуем функцию, которая генерирует (предсказывает) текст с помощью обученной нейронной сети.

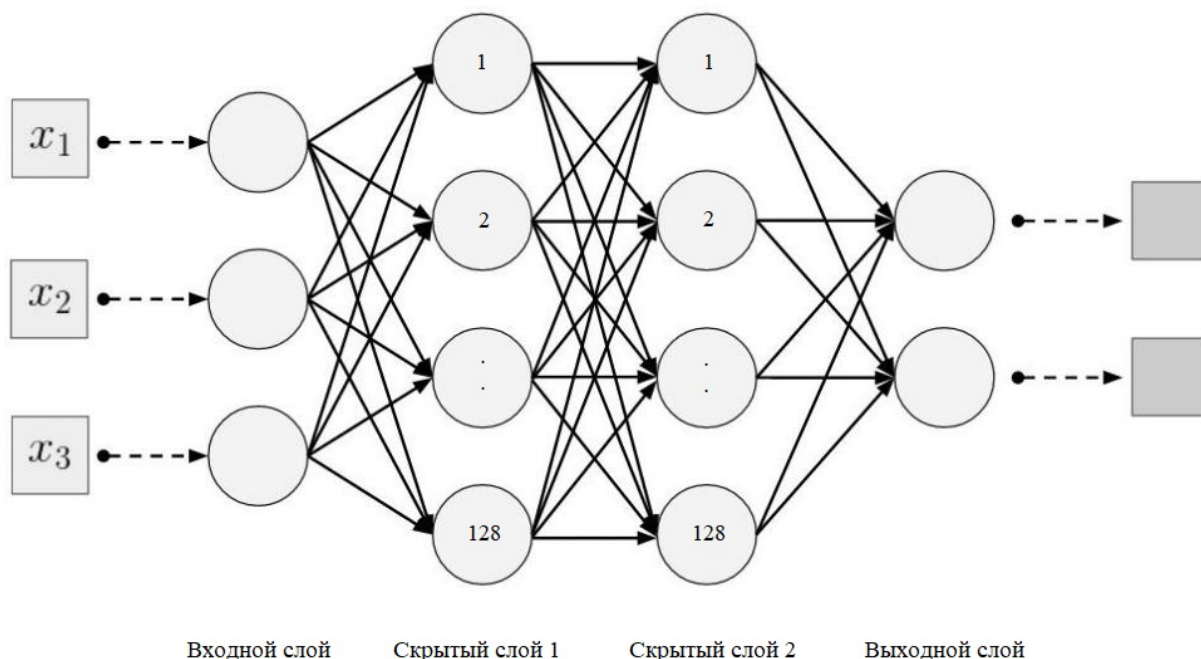
Создаем класс нейронной сети. Превращаем каждый символ на входе сети в вектор. «Скармливаем» эти векторы LSTM слою. Выходы из LSTM слоя пропускаем через Dropout.

Отправляем выход из Dropout на линейный слой размерности словаря, чтобы на выходе получить столько чисел, сколько символов в словаре.

Создаем нейросеть и обучаем ее. Модель нейронной сети, выглядит так:

```
model = TextRNN(input_size=len(idx_to_char), hidden_size=128, embedding_size = 128, n_layers=2).
```

Эта строка создает экземпляр класса TextRNN в библиотеке torch. Этот класс предназначен для обработки текста с использованием рекуррентных нейронных сетей (RNN), где параметры: input\_size (размер входного алфавита); hidden\_size (размер скрытого состояния рекуррентной нейронной сети); embedding\_size (размерность вложения слов); n\_layers (количество слоев в RNN).



*Рисунок 1 - Слои нейронной сети*

Выбираем функцию потерь, оптимизатор и настраиваем расписание, по которому меняем шаг оптимизатора. В нашем случае снижаем шаг в 2 раза, если ошибка не падает 5 шагов подряд. Подаем в нейронную сеть разные части текста и обучаем ее делать все меньше и меньше ошибок, когда она предсказывает следующую букву по предыдущему тексту.

Таким образом, был разработан программный продукт представляющий собой мощный инструмент, основанный на искусственном интеллекте, который способен автоматически создавать текстовое содержимое. Он обладает рядом преимуществ, включая повышение производительности, возможность персонализации контента и создание большого объема текстов.

Программа генерации текста может быть использована в различных областях, таких как маркетинг, реклама, журналистика, литература и другие. Она сокращает время и усилия, затрачиваемые на написание текстов, и позволяет сосредоточиться на более творческих и стратегических задачах.

Однако, стоит учитывать ограничения программы генерации текста. Возможны неточности и ошибки в сгенерированном контенте, поэтому необходимо проводить проверку и редактирование текста для обеспечения его качества и точности. Также важно учитывать этические аспекты использования сгенерированного контента, чтобы избежать манипуляций информацией или создания фейковых материалов.

В целом, программа генерации текста предоставляет возможности для более эффективного и быстрого создания текстового контента. Она является полезным инструментом, но требует сочетания с экспертизой и контролем, чтобы достичь наилучших результатов.

### Список литературы

1. Федоров, Д. Ю. Программирование на языке высокого уровня Python: учебное пособие для прикладного бакалавриата / Д. Ю. Федоров. – 2-е изд., перераб. и доп. – Москва : Издательство Юрайт, 2019. – 161 с. – (Бакалавр. Прикладной курс).
2. Бринк Х. Машинное обучение Наука и искусство построения алгоритмов, которые извлекают знания из данных / Х. Бринк, Д. Ричардс, М. Феверолф. — Санкт-Петербург: «Питер», 2017. — 330 с.
3. Вьюгин В.В. Математические основы машинного обучения и прогнозирования / В.В. Вьюгин. — Москва: «Московский центр непрерывного математического образования», 2014. — 305 с.