Введение

Современные технологии обработки естественного языка активно развиваются благодаря применению нейронных сетей. Одной из актуальных задач является автоматическая классификация текстовых сообщений, позволяющая эффективно анализировать отзывы пользователей, комментарии в социальных сетях и иные текстовые данные. Использование нейронных сетей позволяет повысить точность и автоматизировать процесс анализа текстов. Кроме того, такие технологии открывают новые возможности для систем поддержки принятия решений в различных областях.

Актуальность

В условиях постоянно растущих объемов текстовой информации возникает необходимость в автоматизированных методах её анализа. Классические статистические методы не всегда обеспечивают достаточную точность и гибкость при работе с естественным языком. Применение нейронных сетей позволяет моделировать сложные зависимости в текстах и улучшает качество классификации сообщений. Кроме того, использование автоматической классификации текстов уменьшает трудозатраты и позволяет обрабатывать большие объёмы данных за минимальное время.

Цель и задачи исследования

Целью данной работы является разработка и обучение нейронной сети, способной автоматически классифицировать текстовые сообщения по тональности.

Для достижения цели были определены следующие задачи:

1. Изучить современные методы обработки текстовой информации и классификации текстов.
2. Подобрать и подготовить набор данных для обучения нейронной сети.
3. Разработать модель нейронной сети с использованием Python и TensorFlow.
4. Провести обучение модели и оценку результатов на тестовой выборке.
5. Сделать выводы и предложения по дальнейшему улучшению модели.

Объект и предмет исследования

Объект исследования: текстовые сообщения пользователей, содержащие эмоционально окрашенные высказывания.

Предмет исследования: методы машинного обучения и нейронных сетей, применяемые для анализа и классификации текстовых данных.

Теоретическая часть

Нейронные сети являются моделями, имитирующими работу биологического мозга. Они состоят из слоёв взаимосвязанных нейронов,

которые могут выявлять сложные закономерности в данных. Для анализа текста часто применяются следующие архитектуры:

Полносвязные сети (Fully Connected): каждый нейрон одного слоя соединен со всеми нейронами следующего слоя. Используются для простых задач классификации и регрессии.
Рекуррентные сети (RNN): позволяют учитывать последовательность слов в тексте, сохраняют контекст предыдущих элементов.
Сети с механизмом внимания (Transformer): современные модели, которые эффективно обрабатывают длинные последовательности и позволяют выделять важные части текста.

Для работы с текстовыми данными важно правильно подготовить данные: токенизация — разбиение текста на слова или токены, векторизация — преобразование слов в числовые векторы, выравнивание последовательностей — чтобы все тексты имели одинаковую длину. Все эти шаги позволяют нейронной сети корректно обучаться и классифицировать сообщения.

Практическая часть
Для реализации модели была использована среда Python 3 и библиотека TensorFlow, с интерфейсом Keras. TensorFlow позволяет создавать и обучать нейронные сети, управлять оптимизаторами и функциями потерь. Keras предоставляет удобный интерфейс для построения архитектуры модели и подготовки данных.

Ниже представлен код, реализующий простую полносвязную нейронную сеть для классификации текстов на положительные и отрицательные.

```
# Импорт библиотек
import tensorflow as tf
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, Dense, GlobalAveragePooling1D

# Пример текстов и меток (1 - положительный, 0 - отрицательный)
texts = [
    "я люблю эту книгу",
    "это отличная работа",
    "ужасный фильм",
    "мне не понравилось",
    "прекрасное место",
    "отвратительный сервис"
]
labels = [1, 1, 0, 0, 1, 0]

# Подготовка данных
tokenizer = Tokenizer(num_words=100)
tokenizer.fit_on_texts(texts)
```

```
sequences = tokenizer.texts_to_sequences(texts)
padded = pad_sequences(sequences, maxlen=5)

# Создание модели
model = Sequential([
    Embedding(100, 8, input_length=5),
    GlobalAveragePooling1D(),
    Dense(8, activation='relu'),
    Dense(1, activation='sigmoid')
])

# Компиляция и обучение
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
model.fit(padded, labels, epochs=30, verbose=0)

# Проверка работы модели
test_texts = ["прекрасный фильм", "ужасная книга"]
test_seq = tokenizer.texts_to_sequences(test_texts)
test_pad = pad_sequences(test_seq, maxlen=5)
predictions = model.predict(test_pad)

for text, pred in zip(test_texts, predictions):
    print(f"{text} -> {'Положительный' if pred > 0.5 else 'Отрицательный'} отзыв")
```

Пояснение к коду

1. Импортируются необходимые библиотеки: TensorFlow для построения нейронной сети, Tokenizer и pad_sequences для обработки текстов.
2. Задаются тексты и метки: положительные сообщения отмечены '1', отрицательные — '0'.
3. Тексты токенизируются и преобразуются в числовые последовательности одинаковой длины.
4. Создаётся модель с Embedding-слоем (для векторизации слов), слоем GlobalAveragePooling и двумя Dense-слоями.
5. Модель компилируется с функцией потерь 'binary_crossentropy' и оптимизатором 'adam', после чего обучается.
6. Выполняется проверка модели на новых текстах, выводятся результаты классификации.

Результаты работы программы

После обучения и проверки сети на тестовых примерах получены следующие результаты:
прекрасный фильм → Положительный отзыв
ужасная книга → Отрицательный отзыв
Сеть корректно распознала тональность сообщений, что подтверждает успешное обучение.

Выводы

В ходе выполнения работы была создана нейронная сеть, способная классифицировать текстовые сообщения по тональности. Использование Python и TensorFlow позволило реализовать модель быстро и эффективно. Даже простая архитектура показала удовлетворительные

результаты на небольшой выборке.

Для дальнейшего улучшения модели рекомендуется увеличить количество обучающих данных, использовать более сложные архитектуры, а также применять методы регуляризации и подбора гиперпараметров для повышения точности и устойчивости сети.

Заключение
Выполненная работа демонстрирует возможности нейронных сетей для автоматической классификации текстовой информации. Реализованная модель может служить основой для более сложных проектов, связанных с анализом отзывов, комментариев и других текстовых данных. Дальнейшие исследования могут включать расширение архитектуры, использование рекуррентных сетей и Transformer для повышения качества классификации.