

НЕЙРОСЕТЕВОЙ АНАЛИЗ УРОВНЯ СТРЕССА ПО ТЕКСТОВЫМ ДАННЫМ

Кашко А.А., студент гр.353502

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Жвакина А.В. – канд. техн. наук, доцент

Аннотация. Работа представляет собой исследование и практическое применение современных технологий, которые помогают анализировать текстовые данные. Основное внимание уделено реализации примитивной модели на языке программирования Python с использованием библиотеки PyTorch и её обучению. Данный подход позволяет выявлять уровень стрессового состояния человека на основе лингвистических характеристик текста.

Ключевые слова. Нейросетевой анализ, уровень стресса, текстовые данные, PyTorch, бинарная классификация, лингвистические характеристики.

В условиях цифровой трансформации и роста онлайн-коммуникаций анализ психоэмоционального состояния человека становится все более востребованным. Одним из наиболее распространённых и негативно влияющих факторов является стресс, который может проявляться как на физиологическом, так и на поведенческом уровне, в том числе в речевой и письменной коммуникации. Особенно актуальной эта тема стала после пандемии COVID-19, которая привела к росту психоэмоциональных расстройств, включая тревожность и депрессию. По данным Всемирной организации здравоохранения, за последние пять лет уровень стресса среди населения вырос на 30%, а около 264 миллионов человек во всем мире страдают от депрессивных расстройств [1]. В таких условиях технологии, способные автоматически выявлять признаки стресса по тексту, могут стать важным инструментом для психологов, HR-специалистов и разработчиков сервисов ментального здоровья. В современном обществе всё больше людей выражают свои переживания и эмоции через текстовые сообщения, в том числе в форме постов и комментариев в социальных сетях. Это открывает возможности для анализа эмоционального состояния человека на основе его цифровой активности и может способствовать оказанию своевременной поддержки тем, кто в ней нуждается.

В связи с этим возрастают потребности в автоматизированных системах, способных определять признаки стрессового состояния по анализу текстовой информации. Такой подход может быть востребован в различных сферах – от систем психологической поддержки до мониторинга пользовательских сообщений в социальных сетях.

Целью данной работы является разработка и обучение примитивной нейросетевой модели, реализованной на языке программирования Python с использованием PyTorch, способной определять уровень стрессового состояния человека на основе анализа текстовых данных. Работа направлена на исследование возможностей применения простых архитектур в задаче распознавания эмоционального состояния и закладывает основу для дальнейшего усовершенствования подобных систем.

При выборе инструментария для реализации модели основное внимание уделялось удобству разработки, гибкости архитектуры и поддержке современных методов обучения нейросетей. PyTorch – это фреймворк машинного обучения с открытым исходным кодом, разработанный компанией Facebook. Он поддерживает создание сложных моделей нейронных сетей и их обучение с использованием GPU, что значительно ускоряет процесс вычислений. PyTorch выгодно отличается от конкурентов такими особенностями, как динамическое вычислительное графическое представление и простота интеграции с другими Python-библиотеками. Основные компоненты фреймворка: tensor – основной элемент для хранения данных, сходен с многомерными массивами в NumPy, но позволяет выполнять операции на GPU, module – базовый класс от которого наследуются все модели нейронных сетей, optimizer – инструмент для обновления параметров модели на основе градиентов (числовые значения, которые показывают в каком направлении и насколько сильно нужно изменить параметры модели) [2].

В рамках данной работы была реализована простая нейросетевая модель на языке программирования Python с использованием библиотеки PyTorch. Был подготовлен небольшой датасет, включающий примеры как стрессовых, так и нейтральных текстов, каждому из которых была присвоена соответствующая метка. Общий объём датасета составил около 200 текстов, поровну распределённых между двумя классами: стресс, нейтрально.

На этапе предварительной обработки данные подверглись базовой очистке: тексты были приведены к нижнему регистру, удалены знаки препинания, произведена токенизация. Далее, на основе частотного анализа, был сформирован словарь, каждой уникальной лексеме в котором соответствовал свой числовой идентификатор. Затем тексты были закодированы в виде последовательностей

фиксированной длины, пригодных для подачи в нейросетевую модель. Максимальная длина последовательности была выбрана равной 50 токенам.

Модель представляет собой однослойную нейросеть, которая использует в качестве входных данных текстовые последовательности, закодированные в числовые индексы. На первом этапе обработки входных данных используется слой эмбединга (`nn.Embedding`), который отображает каждый индекс в вектор фиксированной длины (в данной реализации – 16). Это позволяет преобразовать текстовые данные в более подходящий для нейросети формат. Затем применяется операция усреднения (`mean pooling`) по эмбединговым векторам, что приводит к одному вектору фиксированной длины, представляющему весь текст. Далее следует полносвязный слой (`nn.Linear`), на котором применяется функция активации (`ReLU`), помогающая моделировать нелинейные зависимости. Выходной слой модели – также полносвязный, с одним выходом, к которому применяется функция активации сигмоиды (`nn.Sigmoid`). Он преобразует результат в вероятность того, что текст связан с состоянием стресса.

Для классификации используется пороговое значение 0,5: если вероятность больше или равна порогу, то текст классифицируется как стрессовый, иначе – как нейтральный.

Обучение модели производилось на обучающей части выборки (80% данных, оставшиеся 20% – тестовая выборка) с использованием оптимизатора Adam и функции потерь `nn.BCELoss`, подходящей для бинарной классификации. Обучение проводилось в течение 10 эпох с размером батча 16 и `learning rate` 0.001. В ходе каждой эпохи производились проходы по всем обучающим данным с последующим обновлением весов модели. По завершении каждой эпохи отслеживались метрики качества на тестовой выборке.

Для оценки модели использовалась метрика `accuracy` – доля верно классифицированных текстов на тестовой выборке. Также дополнительно анализировались метрики `precision`, `recall` и `F1-score`, особенно важные при несбалансированных классах.

Модель была дополнена функцией (`predict(text: str)`), которая позволяет производить предсказания на новых текстах. Входной текст проходит те же этапы предобработки (очистка, токенизация, преобразование в индексы), после чего подаётся в модель. Результатом является вероятность, интерпретируемая как наличие или отсутствие признаков стресса.

Разработанная модель представляет собой базовый рабочий пример использования нейросетей для анализа эмоционального состояния человека на основе текстовых данных. Несмотря на простоту структуры модели и ограниченность датасета, она демонстрирует работоспособность и может быть использована в качестве отправной точки для более сложных решений.

Перед финальным этапом был проведён запуск разработанной нейросетевой модели (рисунок 1) на тестовом примере. На изображении ниже продемонстрирован результат работы функции `predict()`, в котором модель успешно определила эмоциональную окраску текста. Это подтверждает корректность всех этапов: от предварительной обработки данных до предсказания и финальной классификации. Полученная вероятность интерпретирована согласно установленному порогу, и модель уверенно отнесла текст к нужной категории, демонстрируя работоспособность и применимость решения на практике.

```
Epoch 6, Loss: 0.0531, Train Accuracy: 0.9866, Test Accuracy: 0.6562
Epoch 7, Loss: 0.0130, Train Accuracy: 1.0000, Test Accuracy: 0.6562
Epoch 8, Loss: 0.0135, Train Accuracy: 0.9933, Test Accuracy: 0.7031
Epoch 9, Loss: 0.0118, Train Accuracy: 0.9933, Test Accuracy: 0.6562
Epoch 10, Loss: 0.0188, Train Accuracy: 0.9933, Test Accuracy: 0.7031
Final Test Accuracy: 0.70
Текст: "Всё валится из рук. Просыпаюсь с ощущением, будто на мне лежит каменная плита
шники в кармане: только кажется, что нашла конец, а там – новый узел. День проходит в
ова."
Стресс: ДА (вероятность: 0.99)
Текст: "Все отлично, я спокоен и радуюсь жизни"
Стресс: НЕТ (вероятность: 0.00)
```

Рисунок 1 – Результат работы модели при предсказании эмоционального состояния текста

Для более детального анализа качества работы модели были построены различные графики, иллюстрирующие как процесс обучения, так и результативность классификации. Графики ошибок позволяют проследить динамику обучения на протяжении всех эпох, а гистограмма предсказанных

вероятностей и ROC-кривая дают представление о способности модели различать стрессовые и нейтральные тексты.

График точности на обучающем и тестовом наборах (рисунок 2) позволяет оценить, насколько эффективно модель классифицирует данные на протяжении всех эпох. По оси X отложены номера эпох, по оси Y – значения точности, отражающие долю верно классифицированных примеров. Рост точности на обучающей выборке указывает на успешное усвоение моделью закономерностей в данных, а рост тестовой точности – на способность модели обобщать информацию на новых примерах. Графики обучения и тестирования демонстрируют положительную динамику, постепенно растут и идут примерно параллельно, что свидетельствует о способности модели обобщать знания на новые данные.

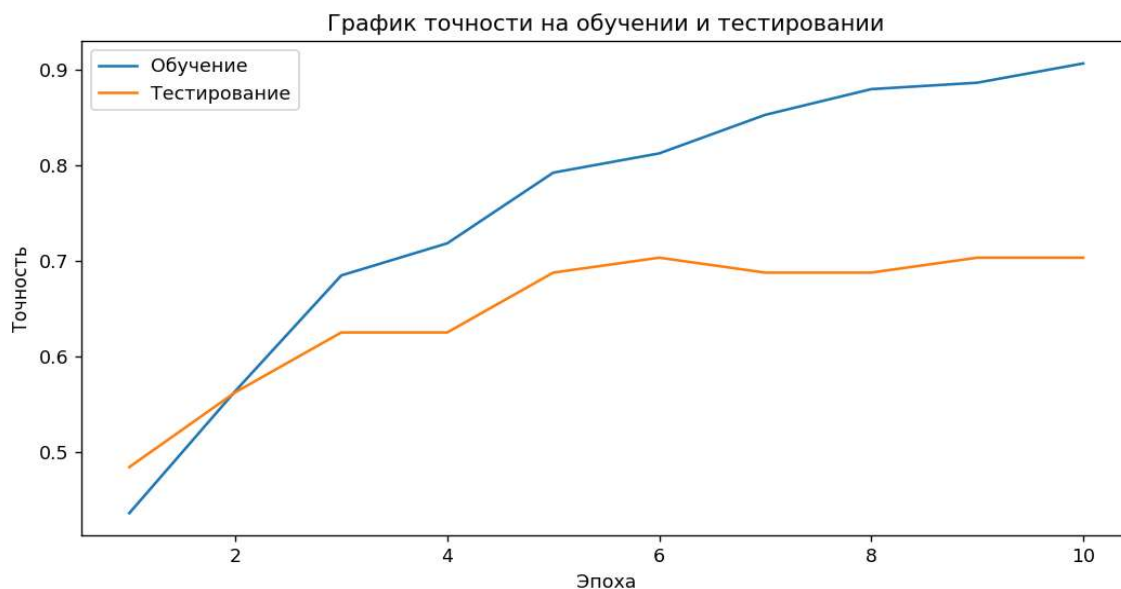


Рисунок 2 – График точности при обучении и тестировании

График потерь на обучении (рисунок 3) показывает, как изменяется средняя ошибка (потери) на обучающем наборе данных в процессе обучения. Потери измеряются с помощью функции BCELoss для бинарной классификации. Ось X отображает номера эпох, а ось Y – значения потерь. Чем ниже это значение, тем лучше нейросеть выполняет задачу классификации, минимизируя ошибку между предсказанным и фактическим значением. График потерь плавно снижается с каждой эпохой, это говорит о том, что модель обучается и её предсказания становятся более точными.

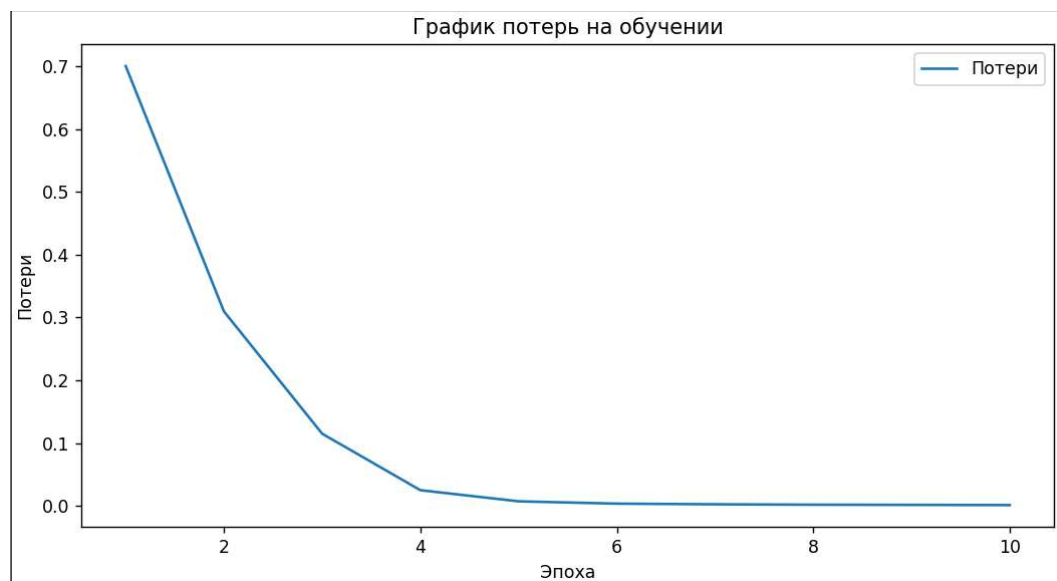


Рисунок 3 – График потерь при обучении

Гистограмма распределения предсказанных вероятностей (рисунок 4) отображает, как модель классифицирует данные на тестовом наборе. График имеет два пика: левый (около 0) – для класса "НЕТ"

(отсутствие стресса), правый (около 1) – для класса "ДА" (наличие стресса). Большая часть предсказаний сосредоточена вблизи крайних значений, что указывает на высокую уверенность модели. Небольшое количество примеров находится в центре диапазона (0.4–0.6), что может свидетельствовать о менее уверенных случаях. Распределение напоминает U-образную кривую, что типично для хорошо обученных бинарных классификаторов. Это говорит о том, что модель редко "колеблется" и чаще дает четкие предсказания.

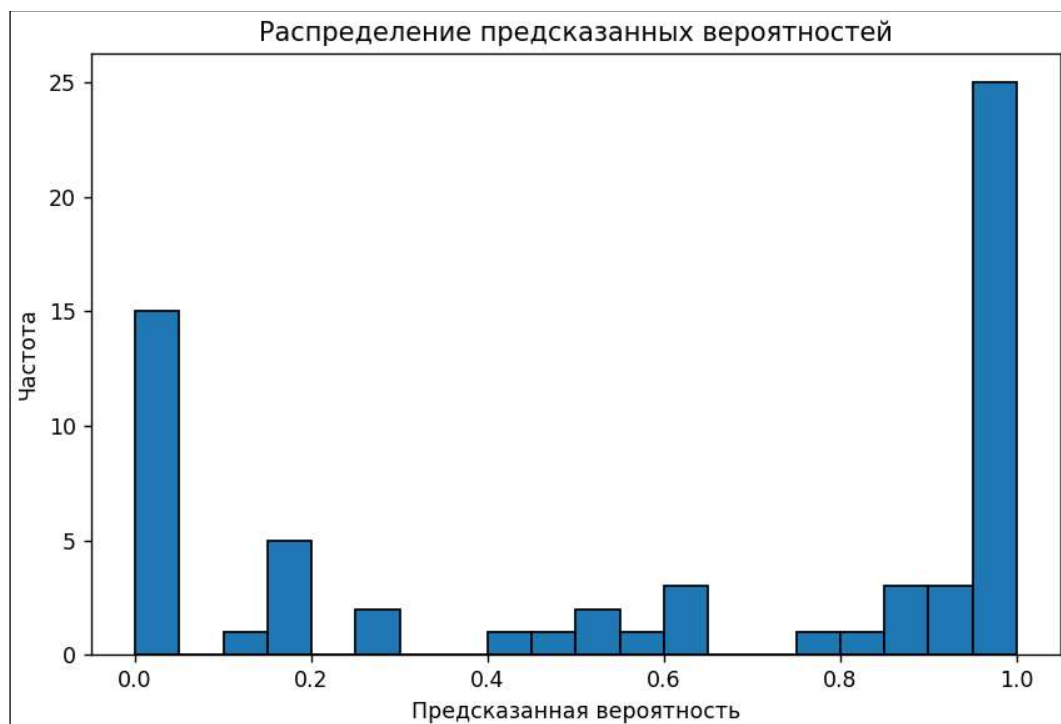


Рисунок 4 – Гистограмма распределения предсказанных вероятностей

ROC-кривая (рисунок 5) описывает взаимосвязь между чувствительностью модели (TPR, или true positives rate – доля истинно положительных примеров) и её специфичностью (описываемой в отношении долей ложноположительных результатов: $1 - \text{FPR}$). TPR, или чувствительность модели, является соотношением корректных классификаций положительного класса, разделённых на все положительные классы, доступные из набора данных. FPR – доля ложноположительных примеров, false positives rate. Это соотношение между ложными срабатываниями (количество прогнозов, ошибочно отнесённых в положительные), и всеми доступными отрицательными классами [3].

Ось X – доля ложных срабатываний (FPR), которые модель ошибочно классифицировала как положительные. Ось Y – доля истинных срабатываний (TPR): это доля положительных примеров, которые модель правильно классифицировала как положительные. ROC-кривая ближе к верхнему левому углу, это значит, что она правильно классифицирует больше положительных примеров при низкой доле ложных срабатываний. Площадь под ROC-кривой даёт количественную меру качества модели: чем она выше, тем лучше модель. $\text{AUC} = 0.9$ указывает на хорошее различие между положительными и отрицательными примерами.

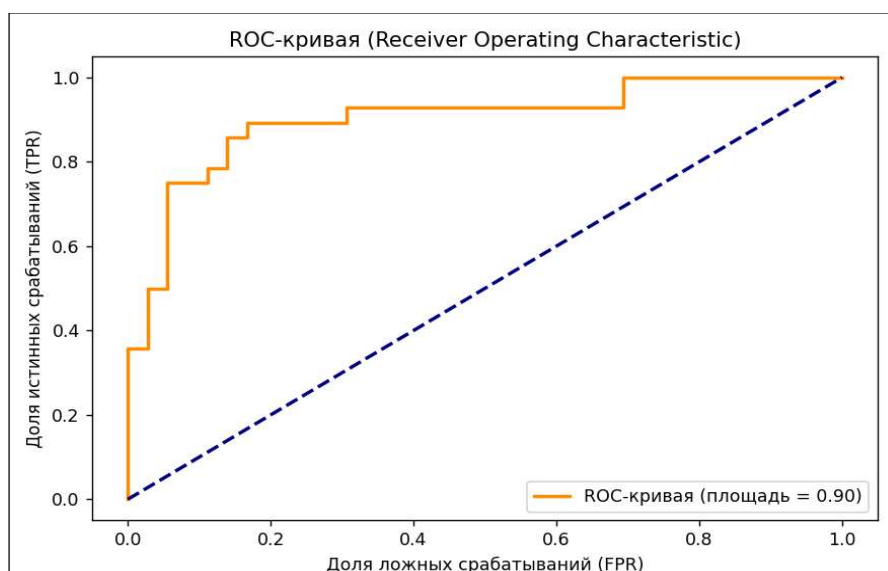


Рисунок 5 – ROC-Кривая

После построения графиков можно сказать, что модель демонстрирует хорошие результаты на задаче бинарной классификации.

Рассматривая направления для развития представленного подхода, следует отметить несколько ключевых аспектов, способных повысить эффективность и практическую значимость модели. Перспективы дальнейшего исследования: расширение набора данных за счёт включения большего количества текстов из различных источников (социальные сети, форумы, личные дневники) позволит улучшить обобщающую способность модели, замена однослойной нейросети на более продвинутые модели, такие как LSTM, GRU или трансформеры (например, BERT), может повысить точность классификации, интеграция анализа не только текстовых данных, но и других модальностей (аудио, видео) для более комплексной оценки эмоционального состояния, адаптация модели для использования в чат-ботах, системах мониторинга соц-сетей или мобильных приложениях для психологической поддержки.

Проведённое исследование продемонстрировало возможность применения простых нейросетевых моделей для анализа уровня стресса по текстовым данным. Несмотря на ограниченный объём датасета и простоту архитектуры, модель показала удовлетворительные результаты, что подтверждается метриками и визуализацией процесса обучения. Работа подчёркивает потенциал нейросетевых технологий в области психолингвистики и открывает направления для дальнейших исследований. В будущем подобные системы смогут стать частью инструментов для мониторинга психического здоровья, помогая своевременно выявлять людей, нуждающихся в поддержке.

Список использованных источников:

1. Депрессивное расстройство (депрессия) [Электронный ресурс] : World Health Organization. – Режим доступа: <https://www.who.int/news-room/fact-sheets/detail/depression>. – Дата доступа: 05.04.2025.
2. Начало работы с PyTorch: создание и обучение нейронных сетей на Python [Электронный ресурс] : Hexlet. – Режим доступа: <https://ru.hexlet.io/blog/posts/nachalo-raboty-s-pytorch-sozdanie-i-obuchenie-neyronnyh-setey-na-python>. – Дата доступа: 07.04.2025.
3. Как понять ROC-кривые с помощью Python [Электронный ресурс] : Habr. – Режим доступа: <https://habr.com/ru/companies/netologyru/articles/582756/>. – Дата доступа: 09.04.2025.

NEURAL NETWORK ANALYSIS OF STRESS LEVELS FROM TEXT DATA

Kashko A.A.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Zhvakina A.V. – PhD in Technical Sciences, Associate Professor

Annotation. The work is a study and practical application of modern technologies that help analyze text data. The main focus is on the implementation of a primitive model in the Python programming language using the PyTorch library and its training. This approach allows you to identify the level of stress in a person based on the linguistic characteristics of the text.

Keywords. Neural network analysis, stress level, text data, PyTorch, binary classification, linguistic features.