

Классификация газов с помощью глубокой нейронной сети

Стрижкин Денис Александрович

Актуальность и Обзор Задачи

Детектирование и определение параметров газов — широко распространенная задача.

Примеры использования:

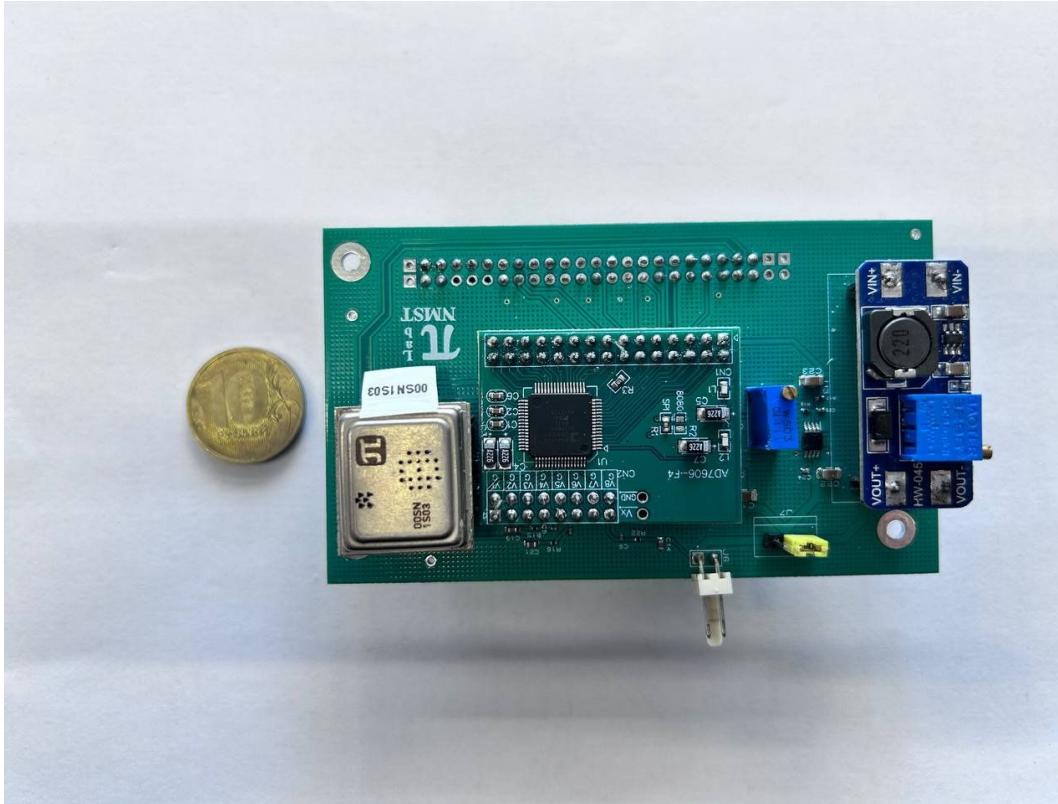
- Мониторинг токсичных летучих веществ (например, Толуола) на производстве с применением химикатов.
- Использование в системах «Умный Нос» для распознавания запахов.

Принцип работы сенсора:

- Основной элемент — чувствительная область из осажденного оксида.
- Измеряется изменение сопротивления датчика, температуры и формы сигнала при осаждении молекул газа.

Проблема: Прямой связи типа и свойств газов с показаниями датчика нет, а количество измеряемых параметров достаточно велико.

Датчик



Датасет

Подготовлен массив данных для 4 видов газов

Виды газов: Бензол, Сероводород, Толуол и о-Ксилол

Параметры: Состоит из **9 признаков (Features)**, измеренных в течение ~ 600 секунд с шагом 0.2 секунды при разных температурах.

	V_h	P	R_h	T	V_m	R_m	dR/dT	Triangle	Iter	Gas	Type
0	3.539844	0.024944	502.341182	154.026314	7.436060	297555.1236	-3535.219560	295420.2720	-623.24	Bensol	
1	3.539287	0.024957	501.928926	153.363342	7.494658	306630.5212	-3652.793340	304654.3252	-623.28	Bensol	
2	3.539093	0.024962	501.771883	153.110792	7.521848	311117.9496	-3722.063704	309080.8132	-623.40	Bensol	
3	3.538885	0.024965	501.654895	152.922657	7.543197	314710.8504	-3773.482740	312555.5384	-623.28	Bensol	
4	3.538905	0.024961	501.730000	153.043435	7.559057	317421.1488	-3805.046956	315429.9652	-623.28	Bensol	

Обработка данных

- Реализован алгоритм обработки данных для создания датасета в формате таблицы.
- Кодирование названий газов (строк) в числа методом **Label Encoder** (scikit-learn).
- Разбиение на тренировочный и тестовый датасеты в соотношении **80/20**.

Архитектура Модели

- **Подбор архитектуры:** Архитектура (включая **Learning Rate** алгоритма Adam) была подобрана с помощью модуля **GridSearch** (scikit-learn).
- **Функция потерь (Loss Function):** **Cross-Entropy Loss** — часто используется в задачах классификации, вычисляет разницу между истинным и предсказанным распределениями вероятностей.
- **Оптимизатор:** **Adam** — используется для оптимизации коэффициентов сети на основе вычисленных ошибок. Известен эффективностью при работе с большими массивами данных.
- **Обучение:** Использовался модуль **DataLoader** для итеративной загрузки данных небольшими частями (батчами).

Архитектура Модели



Результаты Классификации

Общая Точность: Точность классификации, полученная на тестовом наборе данных, составила **89.74%**.

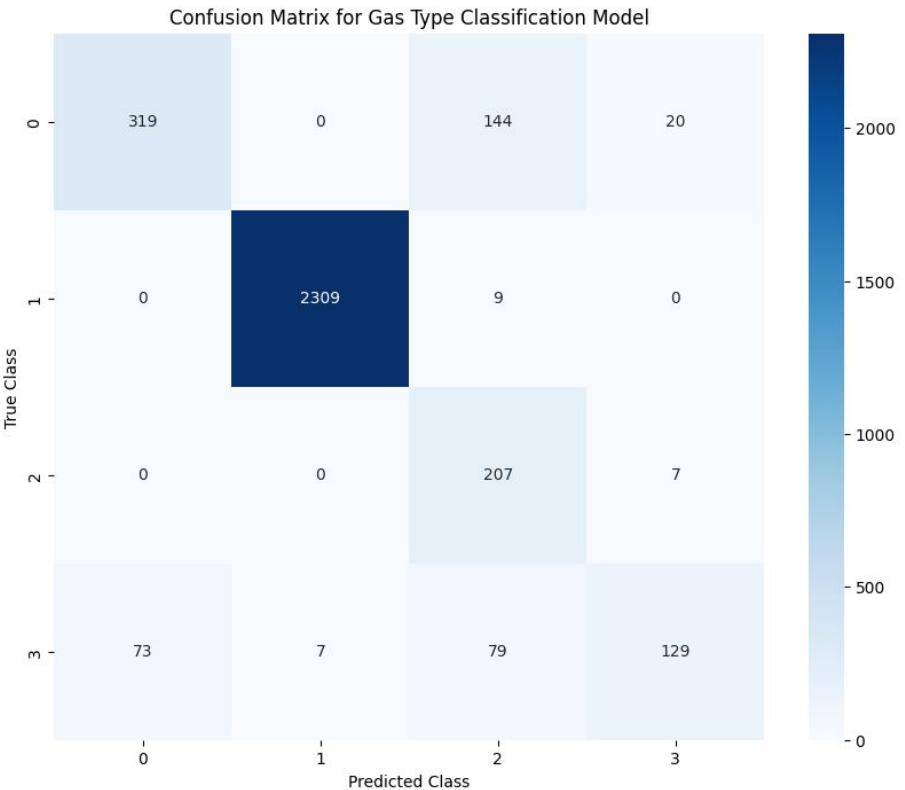
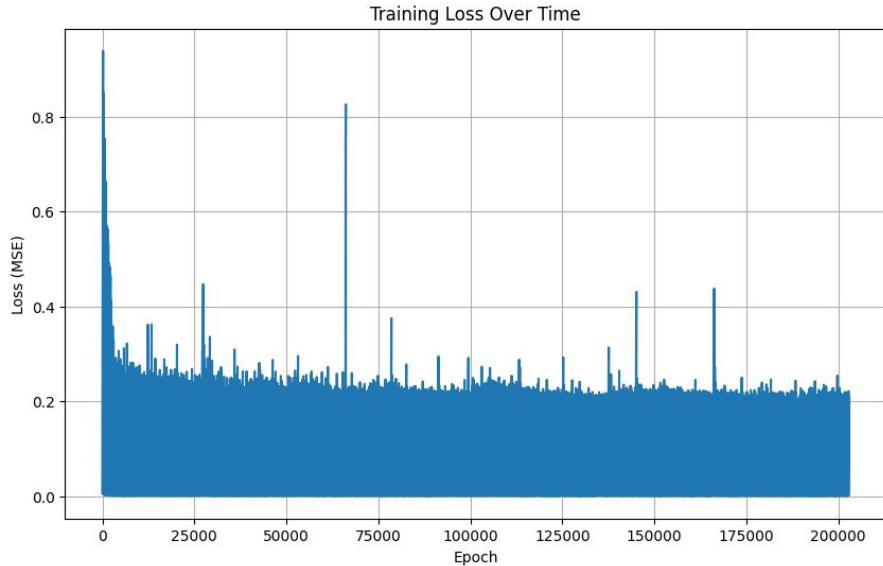
Количество эпох: Обучение проводилось в течение **20000 эпох**.

Анализ по газам:

- **Лучший результат:** У **Сероводорода (H₂S, класс 1)**, что объясняется наибольшим объемом данных для этого газа.
- **Худший результат:** У **о-Ксилола (класс 2)**. Вероятная причина — ошибки, допущенные при сборе данных для о-ксилола.

Проблема различимости: Плохая различимость Бензола, Толуола и о-Ксилола может быть связана с их схожим химическим составом.

Результаты Классификации



Benzene – 0, Toluene – 3, H₂S – 1, O-Xylene – 2

Анализ и Выводы

- График ошибок предсказания показал сильные дрожания и постепенный выход на установившееся значение.
- Периодические выбросы ошибок и не совсем корректное поведение могут означать **переобучение модели**

Создана и оптимизирована полносвязная нейронная сеть с достигнутой точностью **89.74%**

Необходимо устранить недочеты в структуре модели

Для устранения переобучения: Добавить **Dropout-слои** для стабилизации подбора коэффициентов.

Для повышения точности:

- Пересмотреть и исправить ошибки в исходных данных.
- Поискать **дополнительные измеряемые параметры** для улучшения различимости газов со схожим химическим составом.

GUI для Тестирования и Визуализации Данных

Цель инструмента: Создание наглядного и интерактивного способа тестирования данных с датчика и различных версий обученной модели.

Реализация: Приложение с графическим интерфейсом (GUI), разработанное на **Python**.

Используемые библиотеки:

- **Raylib:** Использовалась для отрисовки графического пользовательского интерфейса.
- **PyTorch (torch):** Применялась для загрузки обученной модели

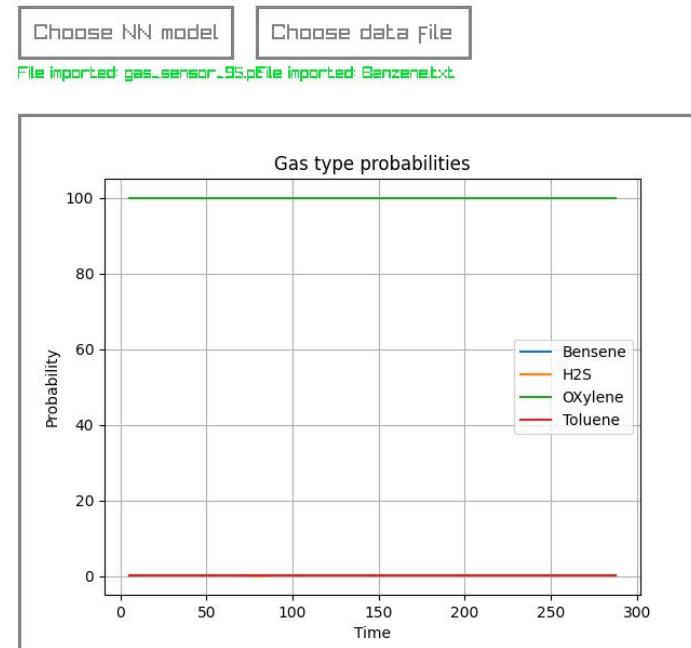
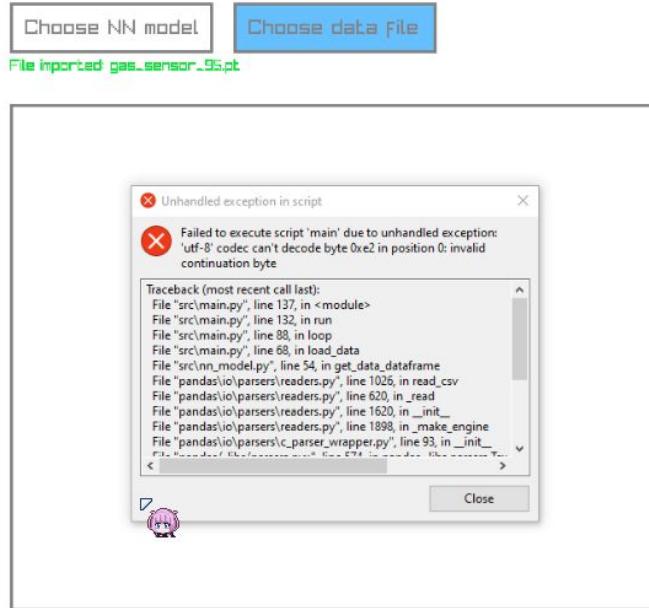
GUI для Тестирования и Визуализации Данных

Функциональность:

- Чтение и обработка исходных данных с датчика с помощью **pandas** и **Label Encoder**.
- Возможность выбора и загрузки файла модели для быстрого тестирования разных версий.
- **Визуализация результата:** По нажатию кнопки строится **график распределения вероятностей** принадлежности газа к каждому из 4 классов.

Преимущество: Позволяет быстро и наглядно **тестировать данные и версии модели** в реальном времени.

GUI для Тестирования и Визуализации Данных



Run model

Спасибо за внимание!