|  |  |
| --- | --- |
| СОГЛАСОВАНО | УТВЕРЖДАЮ |
| Сторона ЗАКАЗЧИКА    Д. В. Попов  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022 г. | Сторона ИСПОЛНИТЕЛЯ  Профессор кафедры  ИАНИ ННГУ, д.т.н.  Н.В. Старостин  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022 г. |
|  |  |

**ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ РЕШЕНИЯ ЗАДАЧИ РЕДУКЦИИ ПРОСТРАНСТВА МНОГОМЕРНЫХ ФУНКЦИЙ**

**Руководство программиста**

**Этап 2. Разработка программной документации**

**ОКР «РАЗРАБОТКА НЕЙРОННОЙ СЕТИ СПЕЦИАЛЬНОГО ВИДА (АВТОЭНКОДЕРА) ДЛЯ РЕШЕНИЯ ЗАДАЧИ РЕДУКЦИИ ПРОСТРАНСТВА МНОГОМЕРНЫХ ФУНКЦИЙ»**

**(Шифр ПО «Enc»)**

Ответственный исполнитель

\_\_\_\_\_\_\_\_\_\_\_\_ В.А. Куликов

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_2022 г.

2022 г.**АННОТАЦИЯ**

Руководство программиста представляет собой информацию по содержанию, правилам работы и настройки программы «Enc». Программа выполняет запуски разного типа автоэнкодеров над различными функциями, подсчитывает ошибку и выводит результаты работы в выходные файлы.

Содержание

1. [Общие сведения о программе. 4](#_Toc93253337)
2. [Характеристики программы. 5](#_Toc93253338)
3. [Обращение к программе. 6](#_Toc93253339)
4. [Входные и выходные данные. 8](#_Toc93253340)

# Общие сведения о программе.

* 1. Назначение программы.

ПО «Enc» предназначен для решения задачи редукции пространства многомерных функций.

* 1. Функции программы.

ПО «Enc» выполняет следующие операции:

* Считывание параметров выполнения программы из консоли.
* Считывание параметров и весов для автоэнкодера, при условии наличия файлов, содержащих эту информацию.
* Генерация случайных данных для параметров функции одним из трех методов.
* Обучение автоэнкодеров.
* Оптимизация параметров автоэнкодера методом полного перебора (не используется из-за долгого времени работы).
* Оптимизация параметров автоэнкодера алгоритмом эффективной глобальной оптимизации.
* Запуск автоэнкодеров на параметрах, полученных из входных файлов.
* Запись в файлы параметров и весов, полученных при обучении автоэнкодера.
* Сохранение диаграмм отклонений для каждого из параметров.
  1. Системные требования.

Для функционирования ПО «Enc» ПЭВМ должны удовлетворять следующим требованиям: процессор Ryzen 5 или Intel i5, видеокарта GTX 1050Ti, оперативная память не менее 8 ГБ DDR4, HDD не менее 124 GB, клавиатура, мышь, интернет-доступ.

ПО «Enc» должно быть работоспособно при установленном python3, numpy, smt, tensorflow, sobol\_seq, keras.

# Характеристики программы.

ПО «Enc» является консольным приложением. Все возможные параметры для передачи в командную строку описаны в разделе «Обращение к программе». Основные характеристики ПО «Enc»:

1. Программа состоит из следующих модулей:

* DataGenerator – Данный класс генерирует случайные данные для параметров функции тремя различными способами: полностью случайно в пределах области определения, методом sobol, методом латинского гиперкуба LSH.
* Normalizer – модуль, реализующий класс нормировки/денормировки данных.
* Function – модуль реализует класс функции, а также содержит стандартные функции func\_1, func\_2, func\_3, func\_4.
* AutoEncoder\_class – модуль содержит класс автоэнкодера и позволяет создать автоэнкодеры всех типов, а также содержит все необходимые функции для работы автоэнкодера.
* Optimizer – модуль содержит класс оптимизации параметров автоэнкодера, который по результатам запусков автоэнкодеров, определяет лучшие параметры для выбранной функции.
* Error\_class – модуль реализует класс подсчета ошибки на основе средней абсолютной ошибки.
* Calculate\_error – модуль, предназначенный для запуска из консоли и вывода ошибок.
* Training\_model – модуль, предназначенный для запуска из консоли и обучения автоэнкодеров.

1. Сохранение параметров автоэнкодеров осуществляется в файлы .txt (Saved Models/Params). Сохранение весов автоэнкодеров осуществляется в файлы .h5 (Saved Models/Weights). Сохранение графиков ошибок для автоэнкодеров с различными параметрами осуществляется в файлы .png (Saved Models/Graphs).
2. Валидация результатов обучения оценивается с помощью ПМ Calculate\_error.
3. Основные функции программы:

* Обучение автоэнкодера для выбранной функции.
* Запуск автоэнкодера на готовых параметрах и весах, а также вывод ошибок.

# Обращение к программе.

Высокоуровневый дизайн представленный на рисунке 1 имеет два сценария запуска:

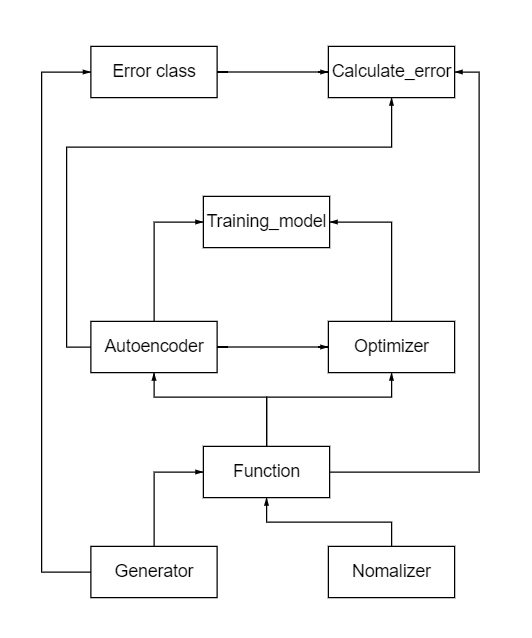


Рисунок 1 – Высокоуровневый дизайн.

*Сценарий 1. Обучение автоэнкодера.*

Для запуска программы необходимо:

* открыть командную строку ОС;
* перейти в директорию с исполняемым файлом программы командой:

>cd C:\Users\{ИМЯ\_ПОЛЬЗОВАТЕЛЯ}\encoderProject\Code\Scripts

* прописать команду:

> python3.9 training\_models.py [-h] [-f {func\_1, func\_2, func\_3, func\_4, all}] [-a {dense, deep, vae, all}] [-i [ITER], --iter [ITER]]

Сначала вызывается скрипт training\_models.py, затем прописываются возможные аргументы:

«-h» – на консоль выводится все возможные аргументы

«-f» – выбор функции нейронной сети

* func\_1
* func\_2
* func\_3
* func\_4
* all

«-a» – выбор автоэнкодера для нейронной сети, где

* dense – сжимающий автоэнкодер
* deep – глубокий автоэнкодер
* vae – вариационный автоэнкодер
* all – использование всех автоэнкодеров для обучения

«-i» – Количество эпох подбора гиперпараметров автоэнкодера (по умолчанию 25)

Описание работы программных модулей:

1. Пользователь вводит:

> python3.9 training\_models.py [-h] [-f {func\_1, func\_2, func\_3, func\_4, all}] [-a {dense, deep, vae, all}] [-i [ITER], --iter [ITER]]

1. В модуле training\_model создается экземпляр тестируемой функции.
2. В модуле training\_model создается экземпляр класса-оптимизатора.
3. В модуле optimize\_class происходит генерация параметров и нормировка данных.
4. В модуле класса-оптимизатора создается экземпляр класса автоэнкодера с выбранными параметрами.
5. Класс-оптимизатор начинает подбор параметров.
6. Класс-опримизатор производит обучение автоэнкодеров с выбранными параметрами и подает на выход наилучшую по отклонению комбинацию параметров автоэнкодера.
7. Класс-оптимизатор сохраняет веса и параметры автоэнкодеров в файлы.

*Сценарий 2. Подсчет ошибки и вывод графиков*.

1. Пользователь вводит:

> python3.9 calculate\_error.py [-h] [-f {func\_1, func\_2, func\_3, func\_4, all}] [-a {dense, deep, vae, all}]

1. В модуле calculate\_error создается экземпляр тестируемой функции.
2. В модуле calculate\_error создается экземпляр класса автоэнкодера с выбранными параметрами и загруженными весами.
3. В модуле calculate\_error создается экземпляр класса подсчета ошибки.
4. С помощью модуля error\_class производится подсчет средних общих ошибок и ошибок по параметрам.
5. В модуле error\_class сохраняется график отклонений по параметрам.

# Входные и выходные данные.

Согласно сценариям обращения к программе, существуют два сценария, которые порождают входные/выходные данные.

*Сценарий 1.*

Входные параметры:

* Выбранный тип функции
* Выбранный тип автоэнкодера
* Количество эпох.

Выходные параметры:

Файлы с весами обученного автоэнкодера представлены на рисунке 2.

* Веса обученного автоэнкодера.

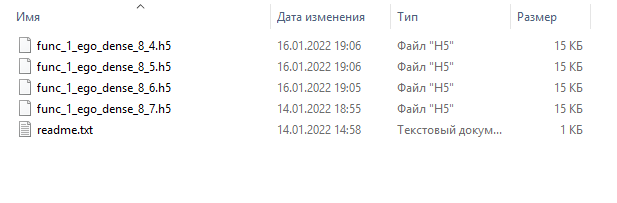


Рисунок 2. - Файлы с весами обученного автоэнкодера.

* Оптимальные гиперпараметры автоэнкодера по итогам процесса оптимизации параметров показаны на рисунке 3.

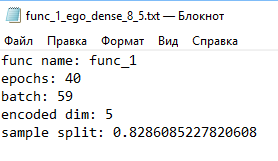


Рисунок 3 – Оптимальные гиперпараметры автоэнкодера

*Сценарий 2.*

Входные параметры:

* Веса обученного автоэнкодера.
* Оптимальные параметры автоэнкодера по итогам процесса оптимизации параметров.
* Выбранный тип функции.
* Выбранный тип автоэнкодера.

Выходные параметры:

* Графики отклонений по параметрам показаны на рисунке 4.

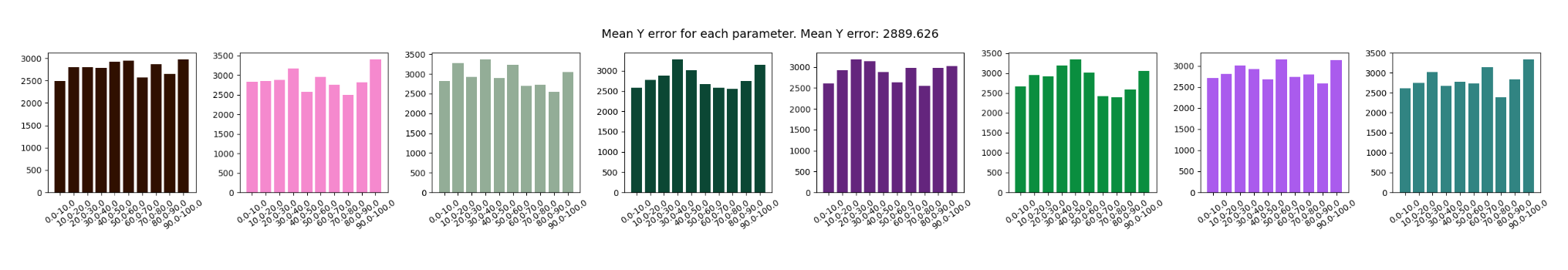


Рисунок 4 – Графики отклонений по параметрам