

СОГЛАСОВАНО

УТВЕРЖДАЮ

Сторона ЗАКАЗЧИКА

Сторона ИСПОЛНИТЕЛЯ

Профессор кафедры
ИАНИ ННГУ, д.т.н.

Попов Д.В.

Н.В. Старостин

«____» _____ 2022 г.

«____» _____ 2022 г.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ РЕШЕНИЯ ЗАДАЧИ РЕДУКЦИИ ПРОСТРАНСТВА МНОГОМЕРНЫХ ФУНКЦИЙ

Руководство оператора

Этап 2.3 Разработка программной документации

**«Разработка нейронной сети специального вида (автоэнкодера) для
решения задачи редукции пространства многомерных функций»**

(Шифр ПО «Enc»)

Ответственный исполнитель

_____ В.А. Куликов

«____» _____ 2021 г.

2022 г.

АННОТАЦИЯ

В данном руководстве описана структура, принципы работы, базовые понятия и интерфейс программного обеспечения «Enc», а также определены условия, необходимые для эффективного функционирования программного обеспечения и указана последовательность действий оператора при запуске и выполнении программы.

СОДЕРЖАНИЕ

1. НАЗНАЧЕНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	4
2. ТРЕБОВАНИЯ К ПРОГРАММНОМУ И АППАРАТНОМУ ОБЕСПЕЧЕНИЮ.....	4
2.1 Минимальный состав аппаратных средств	4
2.2 Минимальный состав программных средств.....	4
3. ВЫПОЛНЕНИЕ ПРОГРАММЫ	4
3.1 Загрузка и запуск программы	4
3.2 Запуск программы отрисовки.....	5
3.3 Этапы работы программы.....	6
3.4 Проверка корректности исходных данных	8
4. СООБЩЕНИЯ ОПЕРАТОРУ.....	8

1. НАЗНАЧЕНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

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

2. ТРЕБОВАНИЯ К ПРОГРАММНОМУ И АППАРАТНОМУ ОБЕСПЕЧЕНИЮ

2.1 Минимальный состав аппаратных средств

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.

2.2 Минимальный состав программных средств

ПО «Enc» должно быть разработано с использованием языка программирования Python.

ПО «Enc» должно быть работоспособно под управлением следующих операционных систем: Windows 10.

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

3. ВЫПОЛНЕНИЕ ПРОГРАММЫ

3.1 Загрузка и запуск программы тренировки нейронной сети

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:

```
> python training_models.py [-h] [-f {func_1, func_2, func_3, func_4, all}] [-a {dense, deep, vae, all}] [-i {iter}]
```

где:

сначала вызывается скрипт `training_models.py`,

затем прописываются возможные аргументы:

- -h – на консоль выводится все возможные аргументы
- -f – выбор функции нейронной сети
 - ♦ func_1, где $F(X) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2 + x_7^2 + x_8^2$, размерность пространства - 8, параметров ф-ции - 8
 - ♦ func_2, где $F(X) = (x_1 + x_2)^4 = x_1^4 + 4 x_1^3 x_2 + 6 x_1^2 x_2^2 + 4 x_1 x_2^3 + x_2^4$, размерность пространства 4, параметров ф-ции - 2 - спектр
 - ♦ func_3, где $F(X) = (x_1 - 100)^2 + (x_2 + 3)^3 + 5 (x_3 + 10)^2$, размерность пространства 6, параметров ф-ции - 3 - сдвинутый спектр
 - ♦ func_4, где $F(X) = (x_1 - 1)^2 + x_2^2 + x_3 + 2x_4 + x_5^3 + x_6$, размерность пространства 10, параметров ф-ции - 6 - сдвинутый спектр
- ♦ all - использование всех функций для обучения
- -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)

3.2 Запуск нейронной сети для получения результатов

Для запуска программы необходимо:

- произвести тренировку нейронной сети для нужной функции и автоэнкодера
- открыть командную строку ОС;
- перейти в директорию с исполняемым файлом программы командой:

```
>cd c:\Users\{ИМЯ_ПОЛЬЗОВАТЕЛЯ}\encoderProject\Code\Scripts
```

– прописать команду:

```
> python calculate_error.py [-h] [-f {func_1, func_2, func_3, func_4, all}] [-a {dense, deep, vae, all}]
```

где:

сначала вызывается скрипт `calculate_error.py`,

затем прописываются возможные аргументы:

- `-h` – на консоль выводится все возможные аргументы
- `-f` – выбор функции нейронной сети
 - ♦ `func_1`, где $F(X) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 + x_6^2 + x_7^2 + x_8^2$, размерность пространства - 8, параметров ф-ции - 8
 - ♦ `func_2`, где $F(X) = (x_1 + x_2)^4 = x_1^4 + 4x_1^3x_2 + 6x_1^2x_2^2 + 4x_1x_2^3 + x_2^4$, размерность пространства 4, параметров ф-ции - 2 - спектр
 - ♦ `func_3`, где $F(X) = (x_1 - 100)^2 + (x_2 + 3)^3 + 5(x_3 + 10)^2$, размерность пространства 6, параметров ф-ции - 3 - сдвинутый спектр
 - ♦ `func_4`, где $F(X) = (x_1 - 1)^2 + x_2^2 + x_3 + 2x_4 + x_5^3 + x_6$, размерность пространства 10, параметров ф-ции - 6 - сдвинутый спектр
- ♦ `all` - использование всех функций для получения результатов
- `-a` – выбор автоэнкодера для нейронной сети, где
 - ♦ `dense` – сжимающий автоэнкодер
 - ♦ `deep` – глубокий автоэнкодер
 - ♦ `vae` – вариационный автоэнкодер
 - ♦ `all` – использование всех автоэнкодеров для получения результатов

3.3 Этапы работы программы

В случае успешного завершения работы программы по тренировке нейронной сети консоль выведет сообщение с результатами тестирования нейронной сети (Рис. 1).

```
func_3 dense training
Opt params:
epochs = 34
batch = 16
encoded dim = 2
sample split = 83.89 % : 16.11 %
Opt mean Y error: [1114.40602335]
```

Рисунок 1.

После этого в папке encoderProject-master\Saved models\Params и encoderProject-master\Saved models\Weights будут сохранены гиперпараметры и веса для данной нейронной сети соответственно (рис. 2 и 3).

Имя	Дата изменения	Тип	Размер
func_1_ego_deep_8_7	25.12.2021 13:02	Текстовый докум...	1 КБ
func_1_ego_dense_8_7	25.12.2021 22:33	Текстовый докум...	1 КБ
func_3_ego_dense_6_1	28.12.2021 0:15	Текстовый докум...	1 КБ
func_3_ego_dense_6_2	26.12.2021 18:00	Текстовый докум...	1 КБ
readme	24.12.2021 15:07	Текстовый докум...	1 КБ

func_1_ego_deep_8_7 – Блокнот

Файл Правка Формат Вид Справка

func name: func_1
 epochs: 60
 batch: 16
 encoded dim: 7
 sample split: 0.9763547985242915

Рисунок 2.

ура > 2 курс > энкодер > encoderProject > Saved models > Weights

Имя	Дата изменения	Тип	Размер
func_1_ego_deep_8_4.h5	25.12.2021 13:01	Файл "H5"	17 КБ
func_1_ego_deep_8_5.h5	25.12.2021 12:15	Файл "H5"	17 КБ
func_1_ego_deep_8_6.h5	25.12.2021 13:02	Файл "H5"	17 КБ
func_1_ego_deep_8_7.h5	25.12.2021 13:00	Файл "H5"	17 КБ
func_1_ego_dense_8_4.h5	25.12.2021 22:32	Файл "H5"	14 КБ
func_1_ego_dense_8_5.h5	25.12.2021 22:22	Файл "H5"	14 КБ
func_1_ego_dense_8_6.h5	25.12.2021 22:32	Файл "H5"	14 КБ
func_1_ego_dense_8_7.h5	25.12.2021 22:28	Файл "H5"	14 КБ
func_2_ego_deep_4_1.h5	29.12.2021 16:44	Файл "H5"	17 КБ
func_2_ego_dense_4_1.h5	26.12.2021 16:27	Файл "H5"	14 КБ
func_3_ego_dense_6_1.h5	28.12.2021 0:15	Файл "H5"	14 КБ
func_3_ego_dense_6_2.h5	27.12.2021 23:22	Файл "H5"	14 КБ
readme	24.12.2021 15:07	Текстовый докум...	1 КБ

Рисунок 3.

При успешном запуске обученной сети на консоль выведется следующее сообщение (рис. 4):

```
K:\Моя папка\II КУРС Магистратуры\Autoencoder project\Code\Scripts>python calculate_error.py -f func_1 -a dense
Mean Y error func_1 dense: 3340.654
```

Рисунок 4.

Найти полученный график можно в папке encoderProject-master\Saved models\Graphs. Выглядит он следующим образом (рис. 5):

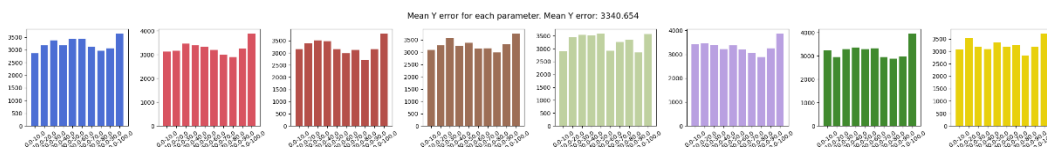


Рисунок 5.

3.4 Проверка корректности исходных данных

В случае возникновения ошибки система выводит на консоль сообщение об этом. В сообщении указывается информация о характере нарушения и местоположение ошибки. Пример обнаружения ошибки (рис. 6).

```
File "C:\Павлуша\учеба\Магистратура\2 курс\энкодер\encoderProject\Code\Scripts\generator_class.py", line 23, in __init__
    assert dim == len(val_range), "Размерность входных диапазонов не равна входной размерности!"
AssertionError: Размерность входных диапазонов не равна входной размерности!

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "training_models.py", line 57, in <module>
    func = TestFunctions.get_func(f_name)
  File "C:\Павлуша\учеба\Магистратура\2 курс\энкодер\encoderProject\Code\Scripts\function_class.py", line 63, in get_func
    return self.functions[name](self)
  File "C:\Павлуша\учеба\Магистратура\2 курс\энкодер\encoderProject\Code\Scripts\function_class.py", line 94, in func_4
    func = Function(f, 'func_4', 6, 4, data_range)
  File "C:\Павлуша\учеба\Магистратура\2 курс\энкодер\encoderProject\Code\Scripts\function_class.py", line 31, in __init__
    self.generator = DataGenerator(self.dim, self.data_range)
  File "C:\Павлуша\учеба\Магистратура\2 курс\энкодер\encoderProject\Code\Scripts\generator_class.py", line 28, in __init__
    raise AssertionError(e.args[0])
AssertionError: Размерность входных диапазонов не равна входной размерности!
```

Рисунок 6. Запись об ошибке

4. СООБЩЕНИЯ ОПЕРАТОРУ

Протокол работы системы содержит:

- сообщения об ошибке в работе программы (рис. 6)
- сообщения о результатах работы программы по тренировке нейронной сети (рис. 1)
- сообщения о результатах работы программы по запуску нейронной сети (рис. 4)