



Министерство науки и высшего образования Российской Федерации
Муромский институт (филиал)
федерального государственного бюджетного образовательного учреждения высшего образования
**«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»**
(МИ ВлГУ)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Тема: «Разработка программы имитационного моделирования
моделей машинного обучения»

Выполнил:
Студент ИС-117
Минеев Р.Р.

Руководитель:
к. т. н., доц. каф. ИС
Щаников С.А.

Муром 2021

Актуальность темы

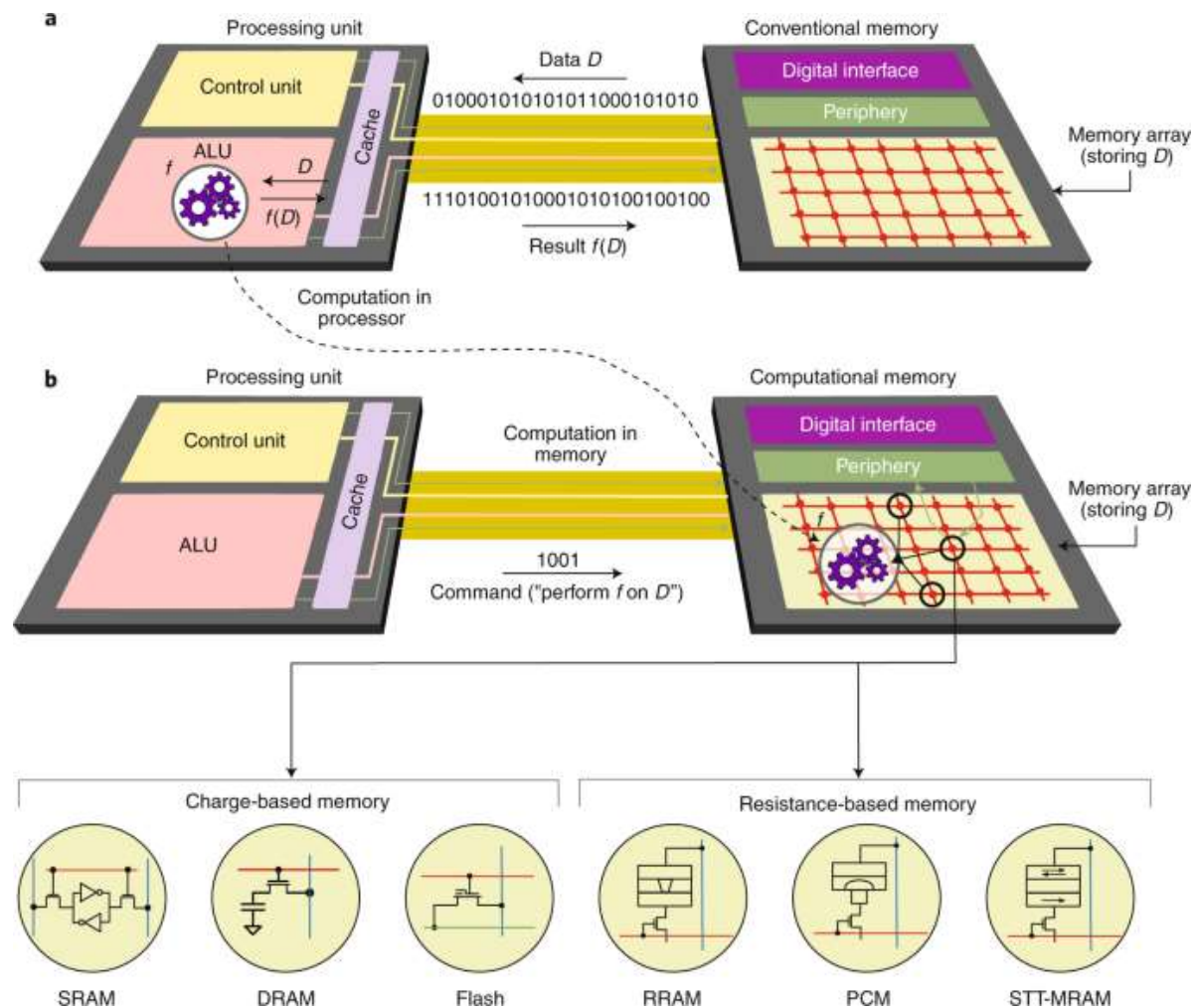


Рисунок 1 – организация вычислений в памяти

Цель и задачи

Цель работы: разработать программу, при помощи которой возможно определить, будут ли на ReRAM готовые модели нейронных сетей работать с приемлемой точностью.

Задачи:

- реализовать возможность тестировать любую модель нейронной сети;
- добавить алгоритмы имитации аналоговых помех для тестирования работы нейронных сетей;
- повысить производительность работы программы за счет использования параллельных вычислений в целях ускорения процесса тестирования;
- сформулировать выводы о работе реализованной программы тестирования.

Обзор программ моделирования ИИС



BaseGroup Labs
ТЕХНОЛОГИИ АНАЛИЗА ДАННЫХ



Общий принцип работы

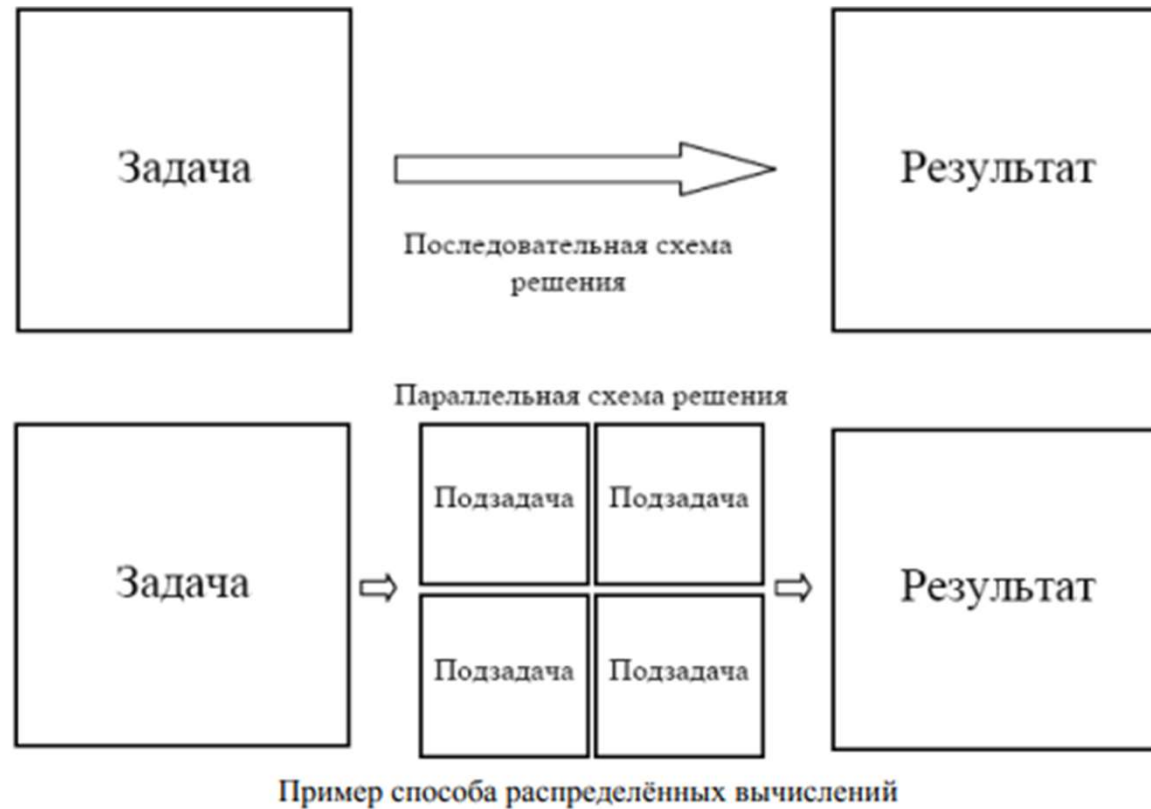
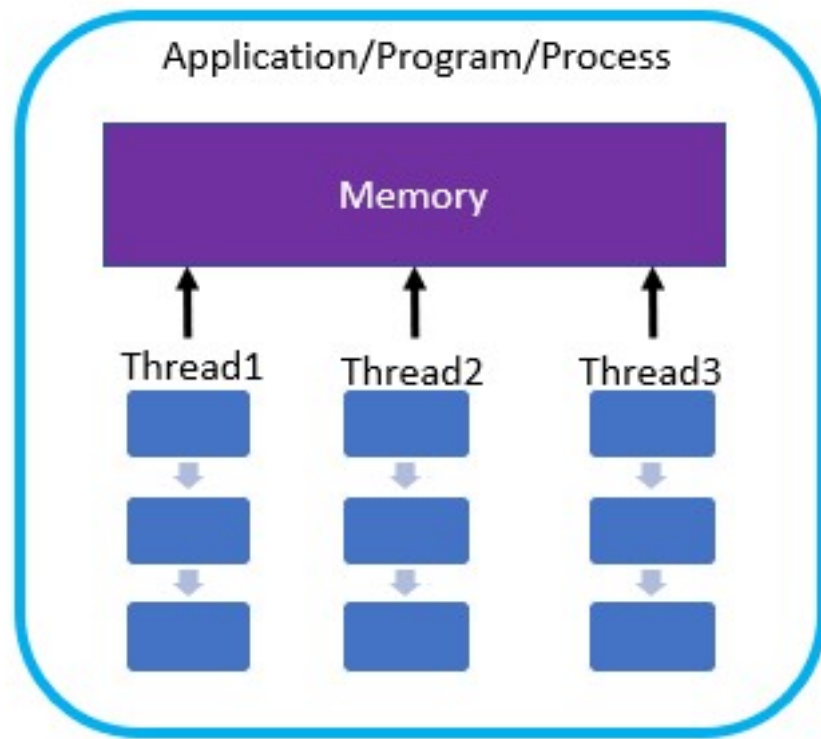
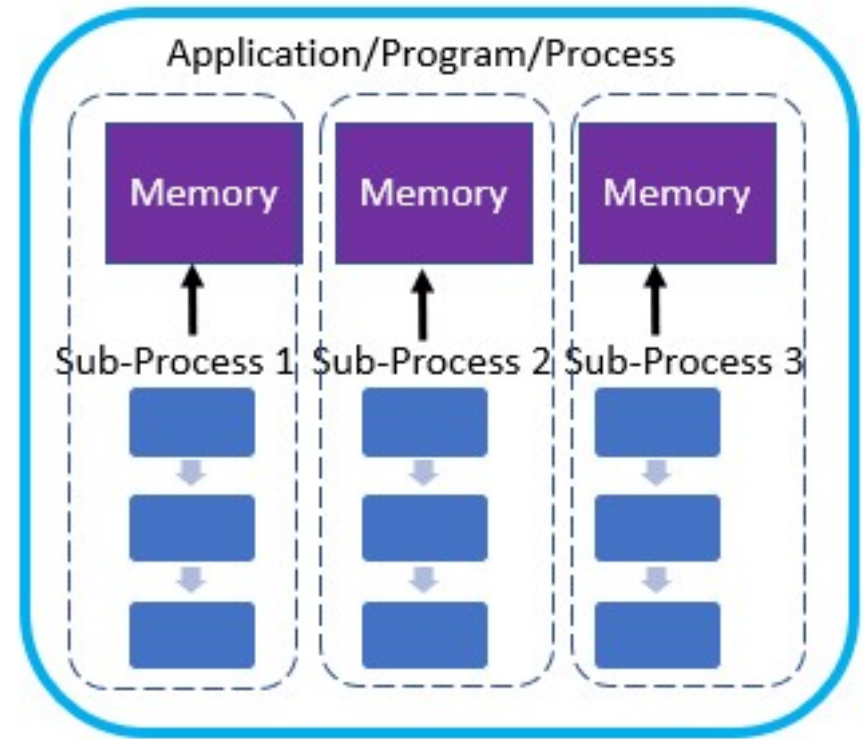


Рисунок 2 – Принцип работы программы

Проблема GIL и её решение



Multi-Threading



Multi-processing

Рисунок 3 – Методы параллельных вычислений

Проект программы

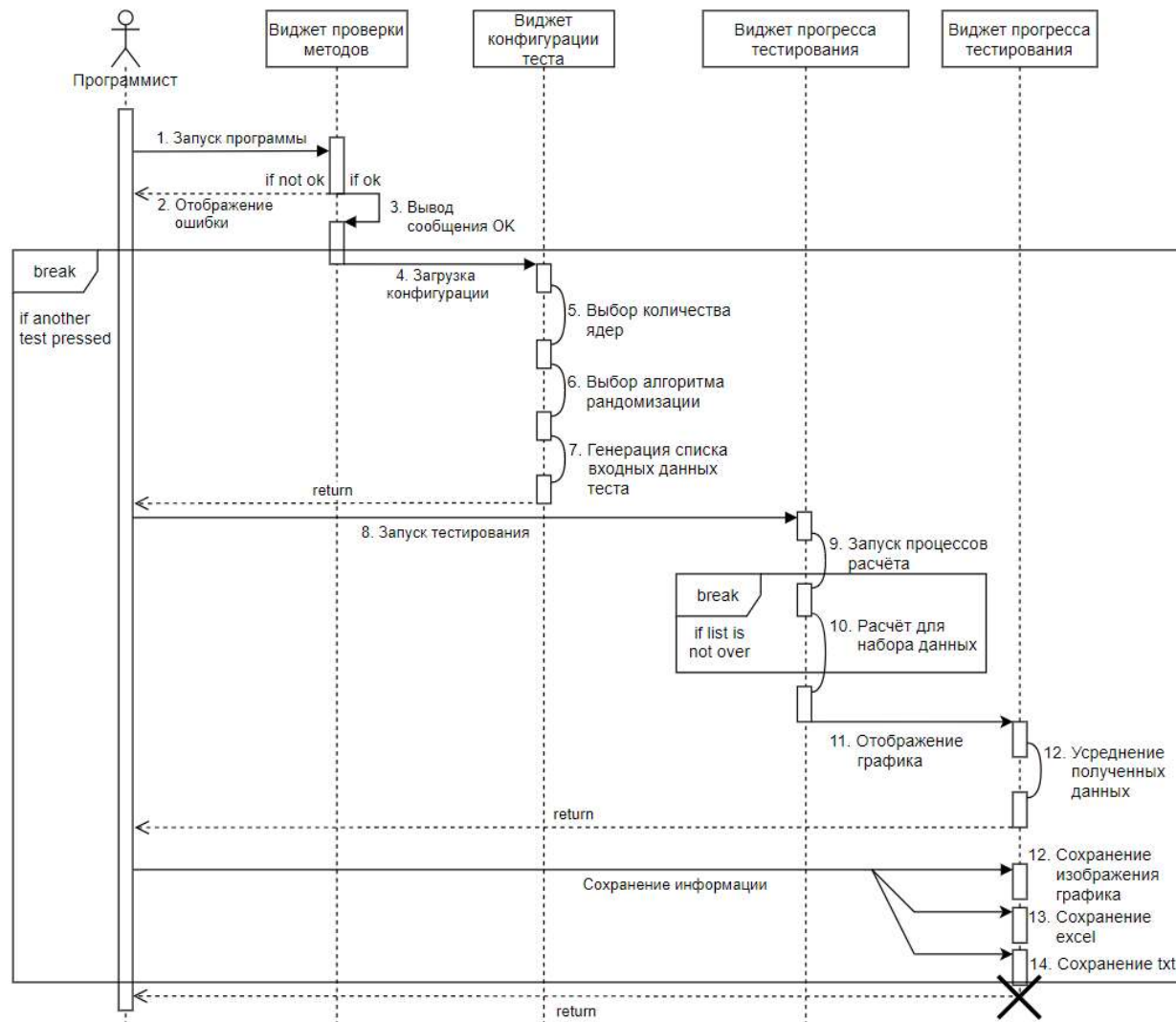


Рисунок 4 – Диаграмма последовательности

Алгоритмы имитации погрешностей весов

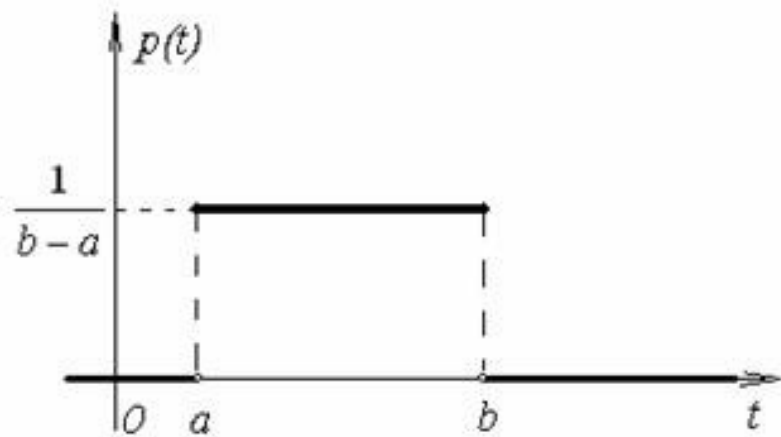


Рисунок 5 – График равномерного распределения

Параметр, указываемый в конфигурации теста: deviation

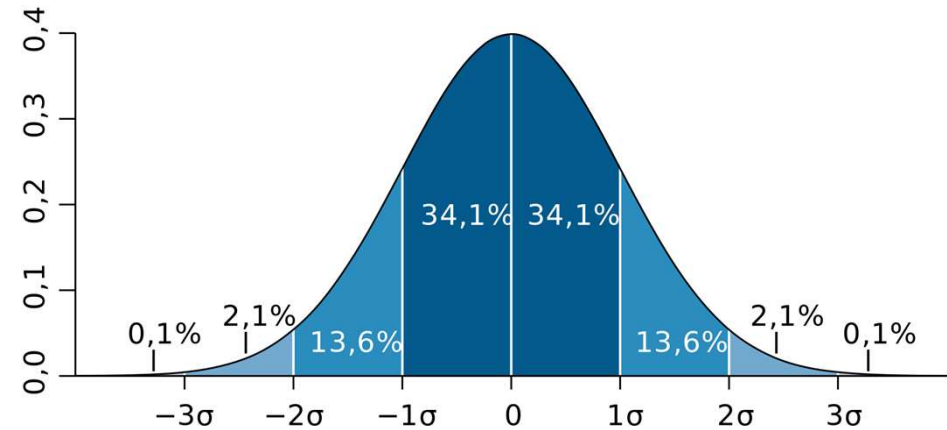


Рисунок 6 – График нормального распределения

Параметр, указываемый в конфигурации теста: sigma

Протокол взаимодействия с программой

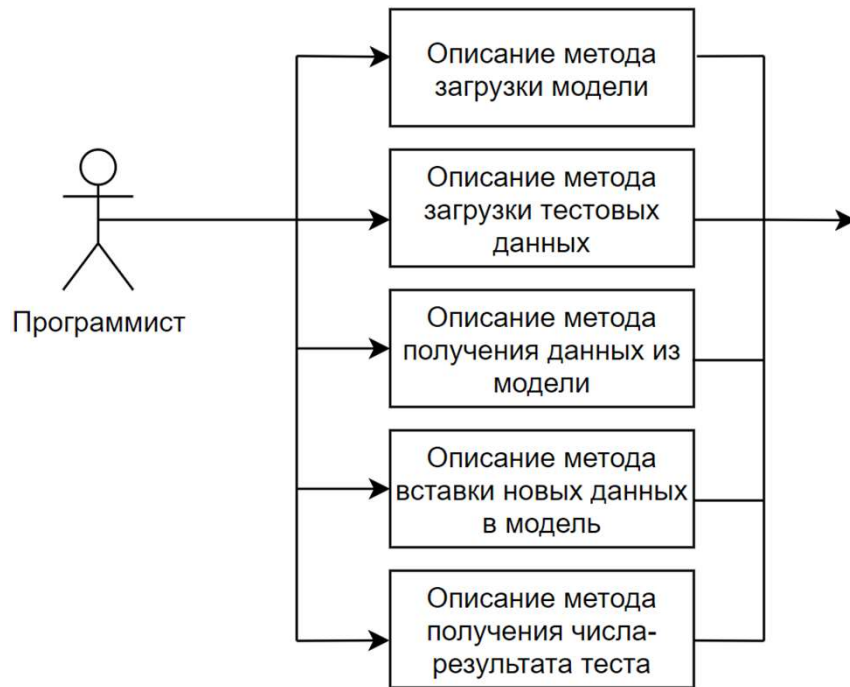
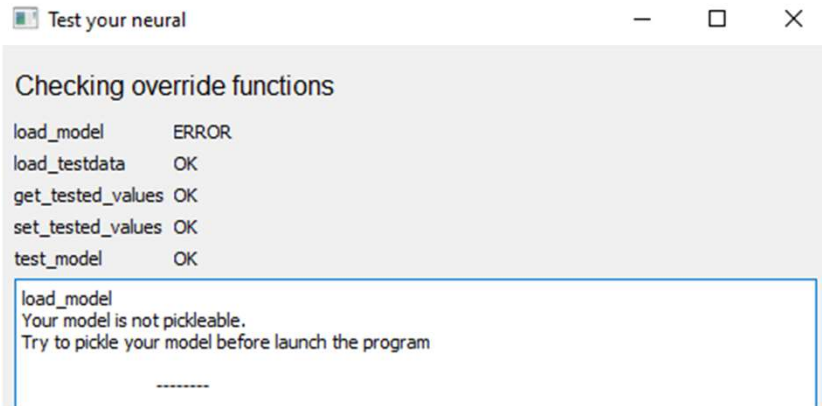


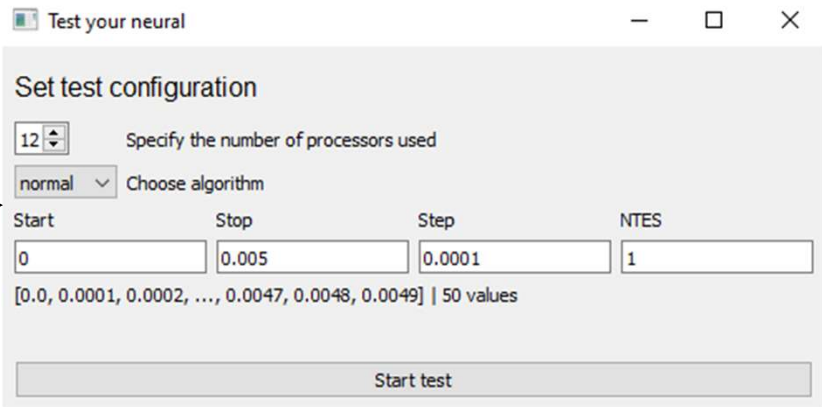
Рисунок 7 – Последовательность первого запуска программы

```
class NeuralCrashTest(NeuralCrash):  
    def __init__(self):  
        super().__init__()  
  
    def load_model(self):  
        self.model = load_model('../2 II Creation/fashion_mnist.h5')  
  
    def load_testdata(self):  
        (_, _), (testX, testY) = fashion_mnist.load_data()  
        testX = testX.reshape(testX.shape[0], 784) / 255  
        self.testdata = testX, testY  
  
    def get_tested_values(self):  
        return self.model.get_weights()  
  
    def set_tested_values(self, values):  
        new_model = self.model  
        new_model.set_weights(values)  
        return new_model  
  
    def test_model(self, model, data):  
        results = model.evaluate(data[0], data[1], batch_size=64)  
        return results[-1]
```

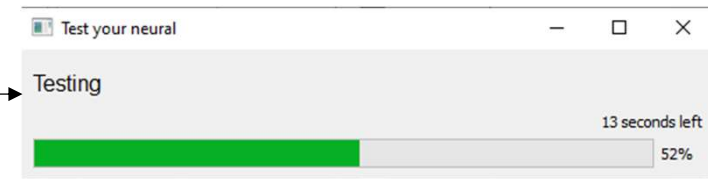
Реализованная программа



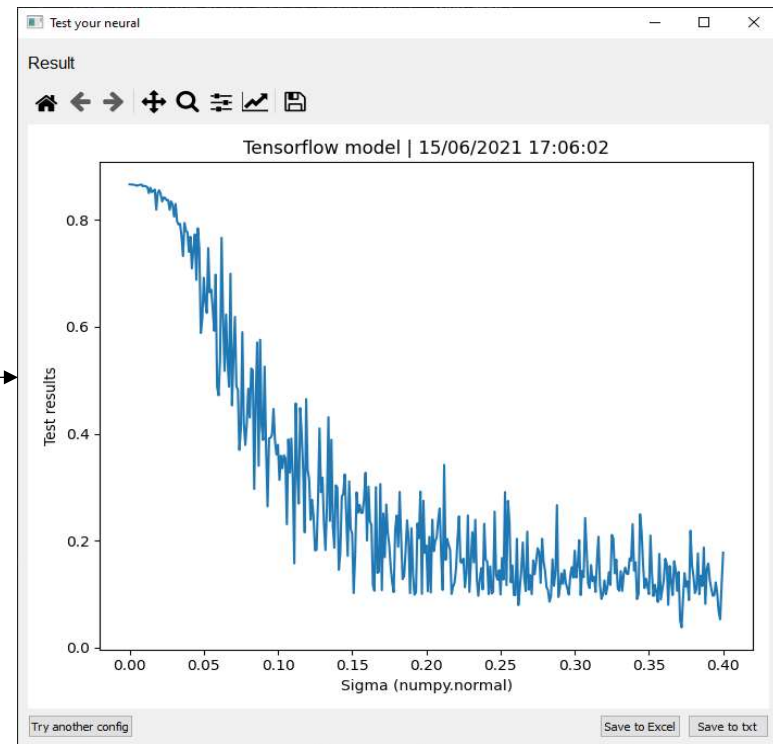
Виджет проверки пользовательских функций



Виджет настройки конфигурации тестирования



Виджет процесса тестирования

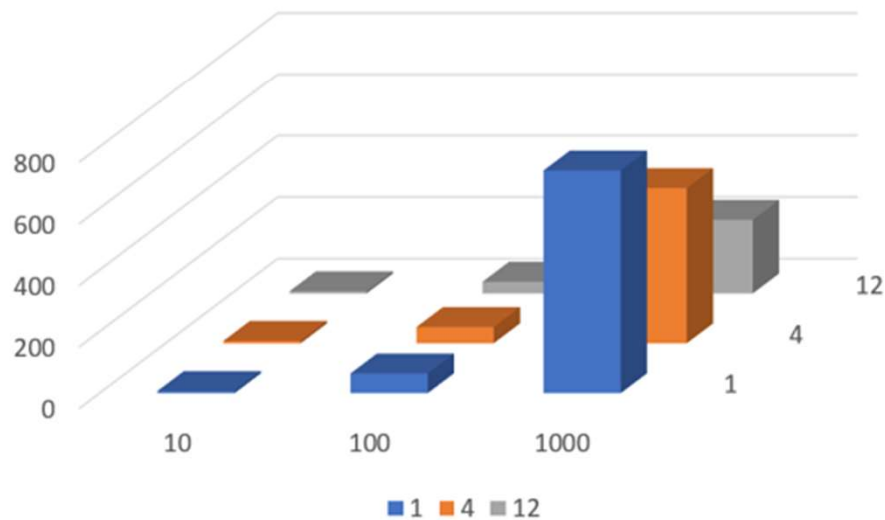


Виджет отображения результатов тестирования

Benchmark (Компьютер)

Ядра \ Значения	1	4	12
10	7	8	7
100	64	53	38
1000	723	504	239

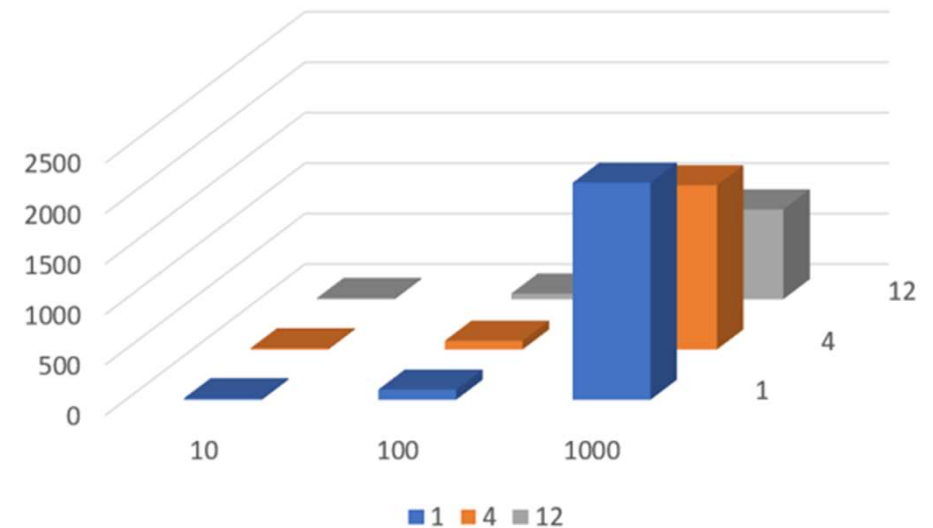
Время тестирования



Результаты тестирования модели Fashion_mnist

Ядра \ Значения	1	4	12
10	16	13	12
100	102	87	56
1000	2152	1628	890

Время тестирования

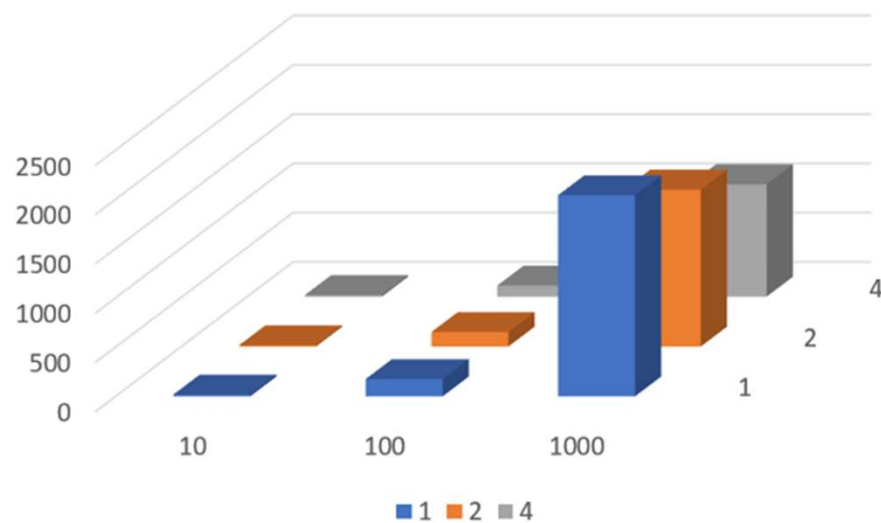


Результаты тестирования модели Fashion_mnist_big

Benchmark (Ноутбук)

Ядра \ Значения	1	2	4
10	18	13	17
100	176	148	113
1000	2042	1593	1142

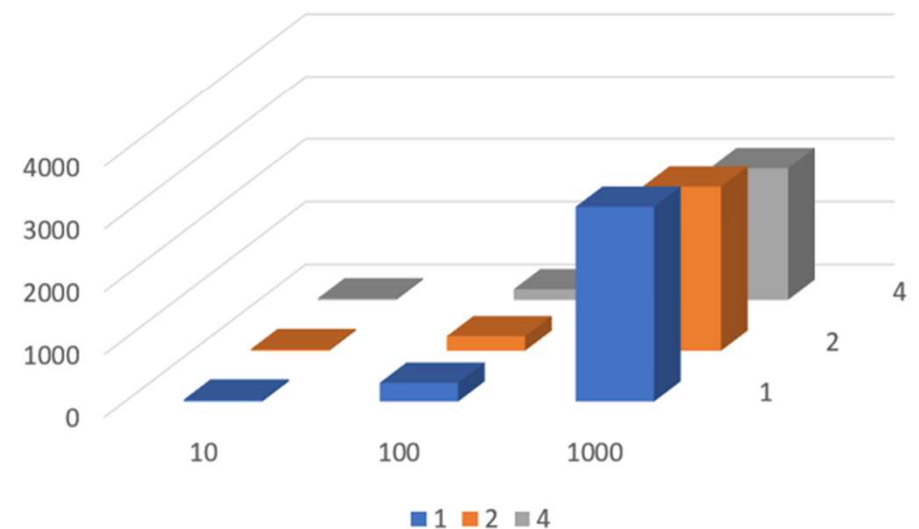
Время тестирования



Результаты тестирования модели Fashion_mnist

Ядра \ Значения	1	2	4
10	31	25	24
100	299	231	170
1000	3102	2615	2097

Время тестирования



Результаты тестирования модели Fashion_mnist_big

Заключение

Таким образом, в процессе разработки программы были выполнены следующие поставленные на работу задачи:

- реализована возможность тестировать любую модель нейронной сети;
- добавлены алгоритмы имитации аналоговых помех для тестирования работы нейронных сетей (алгоритм равномерного распределения и алгоритм нормального распределения);
- повышена производительность работы программы за счет использования метода распараллеливания процессов в целях сокращения времени тестирования;
- сформулированы выводы о работе реализованной программы тестирования.

Спасибо за внимание!