**Министерство науки и высшего образования Российской Федерации**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**“НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО”**

**Факультет программной инженерии и компьютерной техники (ПИКТ)**

**Направление подготовки (специальность) – 09.04.04 (Нейротехнологии и программная инженерия)**

**КУРСОВАЯ РАБОТА**

по дисциплине «Разработка параллельных и распределенных приложений для интеллектуальных информационных систем»

Тема: Разработка и распараллеливание нейронной сети для классификации данных

Обучающийся: Попович Александр Павлович. Р42222

**(Фамилия И.О.) (номер группы)**

Руководитель: Штенников Дмитрий Геннадьевич

**(Фамилия И.О)**

Работа защищена с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Подписи членов комиссии \_\_\_\_\_\_\_\_\_\_\_

(подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

Дата 30.12.2022

Санкт-Петербург

2022

**СОДЕРЖАНИЕ**

[Введение 3](#_Toc123234599)

[1. Анализ и подготовка датасета 4](#_Toc123234600)

[2. Реализация нейронной сети 7](#_Toc123234601)

[3. Распараллеливание нейронной сети 10](#_Toc123234602)

[4. Тестирование 13](#_Toc123234603)

[Заключение 15](#_Toc123234604)

[Список литературы 16](#_Toc123234605)

# Введение

В рамках курсовой работы необходимо разработать и распараллелить нейронную сеть для классификации данных. Датасетом является набор данных о раке молочной железы. Данные представлены в виде набора 30 признаков и двух классов: доброкачественные опухоли и злокачественные. Размерность составляет 569 примеров из которых 212 примеров злокачественно опухоли и 357 доброкачественной.

Нейронная сеть является простейшим персептроном с 30 входных нейронов и выходным нейроном.

Распараллеливание происходит за счет деления датасета на 4 батча. Для каждого батча создается поток, который обучается 10000 эпох. Веса передаются из потока в поток, что позволяет увеличивать точность нейронной сети и сократить время обучения.

# Анализ и подготовка датасета

Первым этапом для реализации нейронных сетей является подготовка, отчистка и нормирование исходных данных, на которых будет обучаться нейронная сеть. В зависимости от того, как будут подготовлены данные будет зависеть успешность обучения и работы нейронной сети. В курсовой работе нейронная сеть занимается задачей классификации или кластеризации – отнесению объектов к определенному классу по набору признаков.

Задачу классификации принято относить к обучению с учителем, т.е у нас заранее имеются данные с правильными ответами. Тогда алгоритм можно обучить на этом наборе данных, и далее применять его для предсказания.

Задачей курсовой работы является реализация нейронной сети для классификации раковых опухолей молочной железы. Датасет представлен в виде набора параметров и ключевой переменной. Набор параметров состоит из 30 признаков, а именно:

mean radius - средний радиус,

mean texture - средняя текстура,

mean perimeter - средний периметр,

mean area - средняя площадь,

mean smoothness - средняя гладкость,

mean compactness - средняя компактность,

mean concavity - средняя вогнутость,

mean concave points - средние вогнутые точки,

mean symmetry - средняя симметрия,

mean fractal dimension - средняя фрактальная размерность,

radius error - ошибка радиуса ошибка,

texture error - текстуры ошибка,

perimeter error - периметра ошибка,

area error - площади ошибка,

smoothness error - гладкости ошибка,

compactness error - компактности ошибка,

concavity error - вогнутости ошибка,

concave points error - вогнутых точек ошибка,

symmetry error - симметрии ошибка,

fractal dimension error - фрактальной размерности ошибка,

worst radius - наихудшего радиуса,

worst texture - наихудшая текстура,

worst perimeter - наихудший периметр,

worst area - наихудшая площадь,

worst smoothness - наихудшая гладкость,

worst compactness - наихудшая компактность,

worst concavity - наихудшая вогнутость,

worst concave points - наихудшая вогнутость точки,

worst symmetry - наихудшая симметрия,

worst fractal dimension - наихудшее фрактальное измерение.

Характеристики вычисляются на основе оцифрованного изображения тонкоигольного аспирата (FNA) массы молочной железы. Они описывают характеристики ядер клеток, присутствующих на изображении.

Целевой переменной являются классы диагноза: M - злокачественный, B - доброкачественный.

Для скачивания датасета необходимо воспользоваться листингом кода 1.

Листинг кода 1 – Скачивание датасета

import sklearn.datasets

breast\_cancer = sklearn.datasets.load\_breast\_cancer()

Следующим этапом является не менее важное разделение на обучающую выборку данных и тестовую. На практике для обучения нейронных сетей данные делят на три части, обучающую, тестовую и валидационную. Обучающий набор данных является самым большим из всех выборок и необходим для непосредственного обучения нейронной сети. Обычно он составляет от 60% - 90% всего набора данных. Размер обусловлен тем, что на достаточном количестве данных, нейронная сеть лучше подберет необходимые веса и смещения, что в следствии повысит точность предсказаний. Тестовым набором данных является часть данных, на которых будет производиться проверка точности предсказаний. Эти данные будут являться новыми для нейронной сети, из-за чего будет происходить объективная оценка точности. Но при использовании тестовых данных, нейронная сеть так же незначительно изменяет веса, что в следствии приводит к тому, что происходит обучение и на тестовых данных. Для этого используют последнюю часть данных, валидационные. Данный набор данных используется для финальной оценки точности нейронной сети. Валидационный набор, как и тестовый не известен нейронной сети. Финальной оценкой точности является результат, полученный на валидационном наборе данных. Данный подход деления датасета на 3 части уместен при большом исходном наборе данных. В курсовой работе используется датасет имеющий всего 569 случаев. Деление на 3 набора данных является малоэффективной при использовании малого объема данных. В следствии чего было принято использовать только обучающую и тестовые выборки. Разделение датасета происходит в следующих пропорциях, 90% для обучающей и 10 на тестовую выборку. Пример разделения датасета приведен на листинге кода 2.

Листинг кода 2 – Деление на обучающую и тестовые выборки.

*# Отделяем данные из датасета*

dataset = [(breast\_cancer.data[i][None, ...], breast\_cancer.target[i]) for i in range(len(breast\_cancer.target))]

trainSize = 0.9 *# Доля данных для обучения*

train\_len = int(len(dataset) \* trainSize) *# Размер данных для обучения*

*# Разделяем данные для обучения и для теста где 90% для обучения и 10% для теста*

X\_train, Y\_train = zip(\*dataset[:train\_len])

X\_test, Y\_test = zip(\*dataset[train\_len:])

X\_train = np.concatenate(X\_train, axis=0)

X\_test = np.concatenate(X\_test, axis=0)

После всех этапов подготовка данных окончена.

# Реализация нейронной сети

Нейронная сеть математическая модель, а также её программное или аппаратное воплощение, построенная по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма. Это понятие возникло при изучении процессов, протекающих в мозге, и при попытке смоделировать эти процессы. Первой такой попыткой были нейронные сети У. Маккалока и У. Питтса. После разработки алгоритмов обучения получаемые модели стали использовать в практических целях: в задачах прогнозирования, для распознавания образов, в задачах управления и др.

ИНС представляет собой систему соединённых и взаимодействующих между собой простых процессоров (искусственных нейронов). Такие процессоры обычно довольно просты (особенно в сравнении с процессорами, используемыми в персональных компьютерах). Каждый процессор подобной сети имеет дело только с сигналами, которые он периодически получает, и сигналами, которые он периодически посылает другим процессорам. И, тем не менее, будучи соединёнными в достаточно большую сеть с управляемым взаимодействием, такие по отдельности простые процессоры вместе способны выполнять довольно сложные задачи.

Нейронные сети не программируются в привычном смысле этого слова, они обучаются. Возможность обучения — одно из главных преимуществ нейронных сетей перед традиционными алгоритмами. Технически обучение заключается в нахождении коэффициентов связей между нейронами. В процессе обучения нейронная сеть способна выявлять сложные зависимости между входными данными и выходными, а также выполнять обобщение. Это значит, что в случае успешного обучения сеть сможет вернуть верный результат на основании данных, которые отсутствовали в обучающей выборке, а также неполных и/или «зашумленных», частично искажённых данных.

За основу нейронной сети для классификации использовался простейший перцептрон. Нейронная сеть представляет из себя класс с набором функций, каждая из которых необходима для анализа полученных результатов и обучения. Функция model приведенная на листинге кода 3.

Листинг кода 3 – Функция model

def model(self, x):

"Функция активации, ступенька"

return 1 if (np.dot(self.w, x) >= self.b) else 0

Данная функция отвечает за функцию активации нейрона. На вход функции подается исследуемые признаки. После матричного умножения признаков на веса нейронной сети и смещения в зависимости от полученного значения присваивается 1 класс, либо 0 класс.

Функция для предсказывания принимает на вход признаки, создает массив предсказаний, на случай если это не один набор, а несколько. Затем для каждого из набора признаков вызывается функция model и результат сохраняется в массив. На выходе получается готовый массив предсказаний. На листинге кода 4 приведена данная функция.

Листинг кода 4 – Функция predict

def predict(self, X):

"Предсказание нейронной сети"

Y = []

for x in X:

result = self.model(x)

Y.append(result)

return np.array(Y)

Для анализа полученных предсказаний есть функция расчета точности предсказаний. Данная функция принимает на вход полученные предсказания и истинные значения. Посредством сравнения предсказанного и истинного значения счетчик инкрементируется. По окончанию подсчета необходимо разделить полученное значение предварительной точности на длину массива предсказания. На листинге кода 5 приведена функция расчета точности.

Листинг кода 5 – Функция calc\_accuracy

def calc\_accuracy(self, Y\_pred\_test, Y\_test):

"Рассчет точности предсказаний"

correct = 0

for predict, test in zip(Y\_pred\_test, Y\_test):

if predict == test:

correct += 1

acc = correct / len(Y\_pred\_test)

return acc

Финальная функция обучения. Данная функция получает на входе наборы признаков и наборы ключевых переменных, количество эпох обучения и значение гиперпараметра скорости обучения. Суть обучения состоит в двух вложенных циклах. Основным циклом является количество эпох. Вложенный цикл берет каждый набор данных и предсказывает по начальным весам ключевую переменную. Когда предсказание ошибочно, происходит перерасчет весов, в ином случае веса не меняются. Так же записывается точность на обучающих данных. В случае если точность возрастает, веса сохраняются, иначе перерасчет весов не сохраняется. По окончанию циклов экземпляр класса сохраняет полученные окончательные веса, выводит график полученных значений точности на каждой итерации, выводит максимальную точность и эпоху на которой данная точность была достигнута. На листинге кода 6 представлена функция обучения.

Листинг кода 6 – Функция fit

def fit(self, X, Y, epochs = 1, lr = 1):

"Обучение нейронной сети"

accuracy = {}

max\_accuracy = 0

wt\_matrix = []

for i in range(epochs):

for x, y in zip(X, Y):

y\_pred = self.model(x)

if y == 1 and y\_pred == 0:

self.w = self.w + lr \* x

self.b = self.b - lr \* 1

elif y == 0 and y\_pred == 1:

self.w = self.w - lr \* x

self.b = self.b + lr \* 1

wt\_matrix.append(self.w)

accuracy[i] = self.calc\_accuracy(self.predict(X), Y)

if (accuracy[i] > max\_accuracy):

max\_accuracy = accuracy[i]

j = i

chkptw = self.w

chkptb = self.b

self.w = chkptw

self.b = chkptb

print(max\_accuracy,j)

plt.plot(list(accuracy.values()))

plt.xlabel("Epoch #")

plt.ylabel("Accuracy")

plt.ylim([0, 1])

plt.show()

return np.array(wt\_matrix)

# Распараллеливание нейронной сети

Для обучения нейронной сети необходимо создать экземпляр класса Perceptron листинг кода 7.

Листинг кода 7 – Класс нейронной сети

class Perceptron:

def \_\_init\_\_ (self):

"Объявление весов и смещения"

self.w = None

self.b = None

def model(self, x):

"Функция активации, ступенька"

return 1 if (np.dot(self.w, x) >= self.b) else 0

def predict(self, X):

"Предсказание нейронной сети"

Y = []

for x in X:

result = self.model(x)

Y.append(result)

return np.array(Y)

def calc\_accuracy(self, Y\_pred\_test, Y\_test):

"Рассчет точности предсказаний"

correct = 0

for predict, test in zip(Y\_pred\_test, Y\_test):

if predict == test:

correct += 1

acc = correct / len(Y\_pred\_test)

return acc

def fit(self, X, Y, epochs = 1, lr = 1):

"Обучение нейронной сети"

accuracy = {}

max\_accuracy = 0

wt\_matrix = []

for i in range(epochs):

for x, y in zip(X, Y):

y\_pred = self.model(x)

if y == 1 and y\_pred == 0:

self.w = self.w + lr \* x

self.b = self.b - lr \* 1

elif y == 0 and y\_pred == 1:

self.w = self.w - lr \* x

self.b = self.b + lr \* 1

wt\_matrix.append(self.w)

accuracy[i] = self.calc\_accuracy(self.predict(X), Y)

if (accuracy[i] > max\_accuracy):

max\_accuracy = accuracy[i]

j = i

chkptw = self.w

chkptb = self.b

self.w = chkptw

self.b = chkptb

print(max\_accuracy,j)

plt.plot(list(accuracy.values()))

plt.xlabel("Epoch #")

plt.ylabel("Accuracy")

plt.ylim([0, 1])

plt.show()

return np.array(wt\_matrix)

Для создания экземпляра класса воспользуемся листингом кода 8.

Листинг кода 8 – Создание экземпляра класса

perceptron1 = Perceptron()

Распараллеливание происходит за счет деления тестовой выборки на 4 части. Каждая часть выборки отправляется в отдельный поток. Таким образом за счет разделения данных на потоки обучение происходит быстрее. Исходя из тестирования, обучение на 40000 эпох без использования потоков затрачивает 1 минуту 25 секунд (Рисунок 1) и точностью 95%.

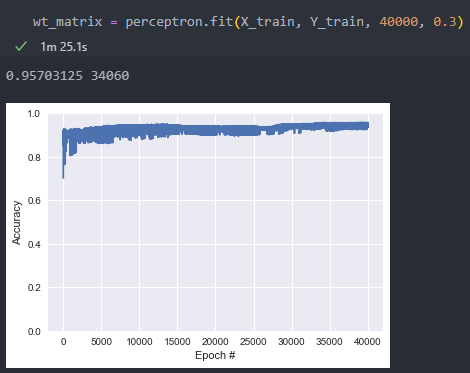


Рисунок 1 – Обучение без распараллеливания

При использовании потоков, время обучение сократилось до 47 секунд (Рисунок 2), что является почти в 2 раза быстрее, точность на обучающих данных составила 98%.



Рисунок 2 – Обучение с использованием потоков

Для поточного обучения необходимо создать новый экземпляр класса. Для каждого потока необходимо вызывать целевую функцию, а именно функцию обучения, передавать заранее разделенные данные, количество эпох обучения для каждого потока и скорость обучения.

После создания потока необходимо его запустить. Так же необходим цикл, который будет ждать окончания работы всех потоков.

# Тестирование

Результаты тестирования на тестовых данных, на которых не обучалась нейронная сеть для обучения без распараллеливания изображены на рисунке 3.

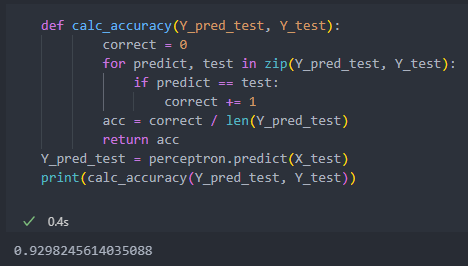


Рисунок 3 – Результаты точности обучения без распараллеливания

Точность составляет 93% при обучении на 40000 эпох, что является отличным результатом для классификации.

Для обучения с использованием поточного распараллеливания так же были произведены замеры точности, изображенные на рисунке 4.

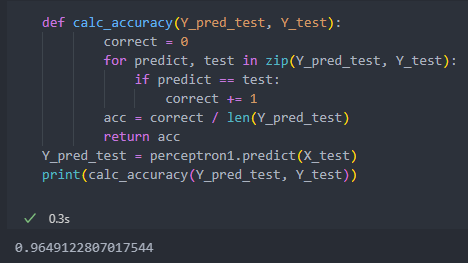


Рисунок 4 – Результаты точности обучения с использованием потоков

Точность составляет 97% при обучении с использованием 4х потоков с разделенным на 4 части тренировочных данных. Точность классификации раковых опухолей так же является высокой, что говорит нам о том, что использование потоков поможет сократить время обучения нейронной сети и не потерять точность предсказаний.

# Заключение

В результате выполнения курсовой работы была написана нейронная сеть по классификации раковых опухолей молочной железы. Были получены знания по написанию и работе нейронных сетей, функций активации и обучения. Написание нейронных сетей без использования библиотек оказалось сложной задачей и требующий большой теоретической базы. Были получены знания по распараллеливанию, которые в последствии позволят использовать их на практике в масштабных проектах. Потоки позволяют снизить необходимое время на обработку информации в разы, особенно при использовании большого объема данных. Цели курсовой работы были выполнены в полном объеме, а знания усвоены на практике.

# Список литературы

1. Барский, А.Б. Логические нейронные сети: Учебное пособие / А.Б. Барский. - М.: Бином, 2013. - 352 c.
2. Галушкин, А.И. Нейронные сети: основы теории / А.И. Галушкин. - М.: ГЛТ, 2012. - 496 c.
3. Куан Нгуен Полное руководство параллельного программирования на Python / Куан Нгуен: Packt Publishing Ltd, 2018 – Глава 3. Работа с потоками в Python

**Министерство науки и высшего образования Российской Федерации**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**“НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО”**

**Факультет программной инженерии и компьютерной техники (ПИКТ)**

**Направление подготовки (специальность) – 09.04.04 (Нейротехнологии и программная инженерия)**

**ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ (РАБОТУ)**

**Студент** Попович Александр Павлович Р42222

(Фамилия, И., О.) (группа)

**Руководитель** Штенников Д.Г., СПб НИУ ИТМО, факультет ПИиКТ, старший преподаватель

(Фамилия, И., О., место работы, должность)

**Дисциплина** Разработка параллельных и распределенных приложений для интеллектуальных информационных систем

**Наименование темы** Разработка и распараллеливание нейронной сети для классификации данных

**Задание** Разработать, обучить и протестировать модель нейронной сети для классификации опухолей молочной железы с использованием методов параллельного программирования. Оформить пояснительную записку.

Студент Дата «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022г.

Подпись Дата

Руководитель Дата «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022 г. Подпись Дата

**Министерство науки и высшего образования Российской Федерации**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**“НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО”**

**Факультет программной инженерии и компьютерной техники (ПИКТ)**

**Направление подготовки (специальность) – 09.04.04 (Нейротехнологии и программная инженерия)**

**ОТЗЫВ РУКОВОДИТЕЛЯ О ВЫПОЛНЕНИИ**

**КУРСОВОЙ РАБОТЫ**

**Студент** Попович Александр Павлович Р42222

(Фамилия, И., О.) (группа)

**Руководитель** Штенников Д.Г., СПб НИУ ИТМО, факультет ПИиКТ, старший преподаватель

(Фамилия, И., О., место работы, должность)

**Дисциплина** Разработка параллельных и распределенных приложений для интеллектуальных информационных систем

**Наименование темы** Разработка и распараллеливание нейронной сети для классификации данных

**ОЦЕНКА КУРСОВОГО ПРОЕКТА (РАБОТЫ)**

| **№**  **п/п** | **Показатели** | **Оценка** | | | |
| --- | --- | --- | --- | --- | --- |
| **5** | **4** | **3** | **0** |
| 1. | Способность к работе с литературными источниками, справочной литературой, Интернет-ресурсами и т. п. |  |  |  |  |
| 2. | Использование иностранных источников |  |  |  |  |
| 3. | Способность к анализу и обобщению информационного материала |  |  |  |  |
| 4. | Владение базовыми знаниями в профессиональной области |  |  |  |  |
| 5. | Владение базовыми знаниями в смежных областях |  |  |  |  |
| 6. | Владение навыками решения технических задач |  |  |  |  |
| 7. | Способность применять знания на практике |  |  |  |  |
| 8. | Уровень и корректность использования в работе методов численного моделирования, инженерных расчетов и статистической обработки данных |  |  |  |  |
| 9. | Владение навыками использования современных пакетов компьютерных программ и технологий |  |  |  |  |
| 10. | Владение навыками оформления отчетных материалов с применением современных пакетов программ |  |  |  |  |
| 11. | Качество оформления пояснительной записки (общий уровень грамотности, стиль изложения, качество иллюстраций, корректность цитирования и пр.\*\*) |  |  |  |  |
| 12. | Качество оформления презентации |  |  |  |  |
| 13. | Владение навыками публичного выступления и межперсональной коммуникации |  |  |  |  |
| 14. | Владение навыками планирования и управления временем при выполнении работы |  |  |  |  |
| **Итоговая оценка** | |  | | | |

\* - не оценивается (трудно оценить)

\*\* согласно рекомендациям

**Отмеченные достоинства:**

**Отмеченные недостатки:**

**Заключение:**

Студент Дата «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022 г.

Подпись Дата

Руководитель Дата «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022 г.

Подпись Дата

**Министерство науки и высшего образования Российской Федерации**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**“НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО”**

**Факультет программной инженерии и компьютерной техники (ПИКТ)**

**Направление подготовки (специальность) – 09.04.04 (Нейротехнологии и программная инженерия)**

**АННОТАЦИЯ НА КУРСОВОЙ ПРОЕКТ (РАБОТУ)**

**Студент** Попович Александр Павлович Р42222

(Фамилия, И., О.) (группа)

**Руководитель** Штенников Д.Г., СПб НИУ ИТМО, факультет ПИиКТ, старший преподаватель

(Фамилия, И., О., место работы, должность)

**Дисциплина** Разработка параллельных и распределенных приложений для интеллектуальных информационных систем

**Наименование темы** Разработка и распараллеливание нейронной сети для классификации данных

**ХАРАКТЕРИСТИКА КУРСОВОГО ПРОЕКТА (РАБОТЫ)**

**1. Цели и задачи работы**

Сформулированы при участии студента

Предложены студентом

Определены руководителем

Задачей данного курсового проекта является разработка, обучение и тестирование нейронной сети, для классификации опухоли молочной железы с использованием методов параллельного программирования

**2. Характер работы** Расчет Конструирование

Моделирование Другое, разработка

Работа носит теоретический характер.

**3. Содержание работы** Работа состоит из семи разделов; Введение, анализ и подготовка датасета, реализация нейронной сети, распараллеливание нейронной сети, тестирование, заключение. В каждом разделе приведено подробное описание проделанной работы

**4. Выводы**

По результатам выполнения курсовой работы удалось разработать, обучить и протестировать нейронную сеть для классификации опухолей молочной железы с использованием методов параллельного программирования.

Студент Дата «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022 г.

Подпись Дата

Руководитель Дата «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022 г

Подпись Дата

**Министерство науки и высшего образования Российской Федерации**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**“НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО”**

**Факультет программной инженерии и компьютерной техники (ПИКТ)**

**Направление подготовки (специальность) – 09.04.04 (Нейротехнологии и программная инженерия)**

**ГРАФИК ВЫПОЛНЕНИЯ КУРСОЙ РАБОТЫ**

**Студент** Попович Александр Павлович Р42222

(Фамилия, И., О.) (группа)

**Руководитель** Штенников Д.Г., СПб НИУ ИТМО, факультет ПИиКТ, старший преподаватель

(Фамилия, И., О., место работы, должность)

**Дисциплина** Разработка параллельных и распределенных приложений для интеллектуальных информационных систем

**Наименование темы** Разработка и распараллеливание нейронной сети для классификации данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Наименование этапа** | **Дата завершения** | | **Оценка и подпись руководителя** |
| **Планируемая** | **Фактическая** |
| 1 | Получение и уточнение задания | 30.09.2022 | 30.09.2022 |  |
| 2 | Изучение теоретического материала | 28.10.2022 | 28.10.2022 |  |
| 3 | Выбор датасета | 14.11.2022 | 14.11.2022 |  |
| 4 | Разработка нейронной сети | 27.11.2022 | 27.11.2022 |  |
| 5 | Обучение нейронной сети | 10.12.2022 | 10.12.2022 |  |
| 6 | Тестирование нейронной сети | 16.12.2022 | 16.12.2022 |  |
| 7 | Оформление пояснительной записки | 24.12.2022 | 24.12.2022 |  |

Студент Дата «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022 г.

Подпись Дата

Руководитель Дата «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022 г.

Подпись Дата