|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ *Робототехники и комплексной автоматизации*

КАФЕДРА *Системы автоматизированного проектирования (РК-6)*

**ОТЧЕТ О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

по дисциплине:

Введение в искусственный интеллект

|  |  |  |
| --- | --- | --- |
| Студент |  | Абидоков Рашид Ширамбиевич |
| Группа |  | РК6-11М |
| Вариант |  | 1 |
| Тема лабораторной работы |  | Реализация искусственной нейронной сети |

Студент **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Абидоков Р. Ш.**

*подпись, дата фамилия, и.о.*

Преподаватель **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Федорук В. Г.**

*подпись, дата фамилия, и.о.*

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Москва, 2020 г.*

Оглавление

[Задание на лабораторную работу 3](#_Toc62683568)

[Теоретические сведения 3](#_Toc62683569)

[Примеры работы программы 5](#_Toc62683570)

[Описание программной реализации 6](#_Toc62683571)

# Задание на лабораторную работу

Разработать, используя язык C/C++, двухслойную нейронную сеть с линейными функциями активации, обеспечить ее обучение для решения задач сжатия данных с потерями.

# Теоретические сведения

*Искусственная нейронная сеть* – сеть, в качестве вершин которой выступают искусственные нейроны. ИНС осуществляет преобразование вектора входных сигналов (воздествий) в вектор выходных сигналов. Выходной сигнал каждого нейрона формируется путем применения *функции активации* к взвешенной сумме входных сигналов.

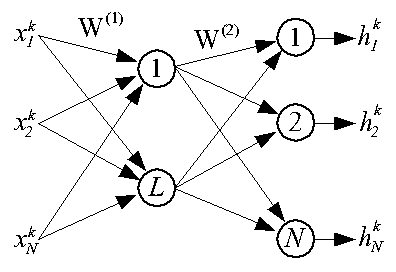


Рис 1. Схема двухслойной ИНС

Для сети *с линейной функцией активации* выход нейрона равен непосредственно взвешенной сумме. Т.е. вектор выходных сигналов каждого слоя формируется путем умножения вектора входных сигналов на матрицу весов. В этом случае для двухслойной ИНС с линейной функцией активации

где вектор входных сигналов k-го объекта, матрица весов первого слоя, матрица весов второго слоя, вектор выходных сигналов k-го объекта.

Обучение сети, состоящее в оптимальном подборе весов, составляющих матрицы и , подразумевает минимизацию целевой функции в виде:

где количество объектов выборки, размерность выходного вектора.

Поскольку в задаче сжатия изображения стоит задача минимизации отклонения полученного изображения от начального,

и целевая функция принимает вид

Для отыскания минимума используется метод обратного распространения ошибки. В рамках данного метода уточнение весовых коэффициентов производится формулой метода градиента

С учетом линейности функции активации выражения для частных производных выходного слоя:

Для первого слоя:

# Примеры работы программы

Для работы используются черно-белые изображения в формате .png с разрешением 256x256 пикселей. Исходное изображение разбивается на прямоугольные части, которые пропускаются через сеть, затем собирается выходное изображение. Обучение сети производилось с помощью изображения, приведенного на Рис. 2



Рис 2. Исходное изображение

Пример работы при различных размерах ячейки

|  |  |  |
| --- | --- | --- |
|  |  |  |
| C:\Study\Introduction-to-AI\lab4\src\img\0-in.bmp |  |  |

Пример работы при различном количестве нейронов

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

# Описание программной реализации

Программа написана на языке C++ с использованием компилятора gcc. Для работы с изображениями используется библиотека cimg.

Алгоритм работы программы (см. Листинг 1):

1. С помощью функции load\_images загружаются изображения с диска, формируется вектор изображений
2. Вектор изображений функцией images\_to\_matrices преобразуется в вектор матриц размерности 256x256
3. Каждая из матриц функцией matrices\_to\_cell\_vecs преобразуется в набор матриц размерности , где – заданные числа, а затем данные матрицы переразбиваются в вектора размерности для последующей подачи в нейронную сеть
4. Создается нейронная сеть, по первому изображению в выборке происходит обучение
5. Все изображения пропускаются через сеть
6. Пункты 1-3 выполняются в обратном порядке, собираются выходные изображения

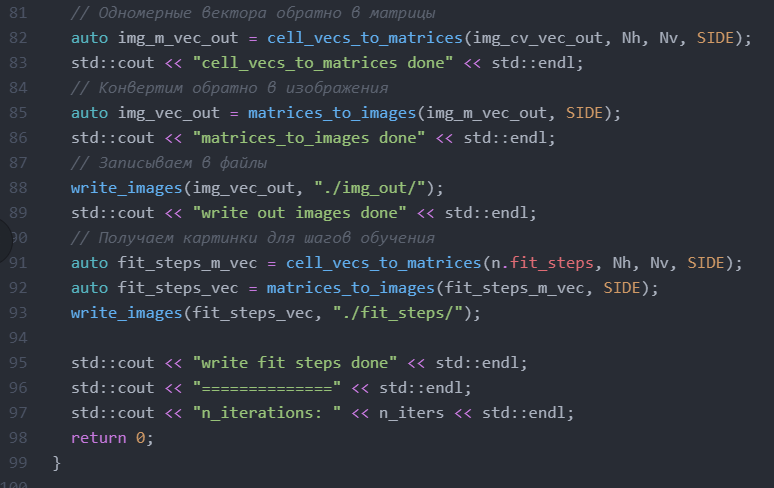
Алгоритм обучения нейронной сети (см. Листинг 2):

1. Веса инициализируются малыми случайными числами из интервала – интервал подобран эмпирически
2. Значение квадратичной ошибки на предыдущем проходе задается очень большим числом
3. Совершается проход по обучающему объекту (представляющему собой изображение в виде набора векторов размерности ) – для каждого вектора вычисляется выход сети, находятся ошибки на втором и первом слоях, веса модифицируются по формулам, приведенным ранее
4. Вычисляется значение квадратичной ошибки на текущем проходе, сравнивается с предыдущим – если разница меньше определенной или количество проходов достигло максимального, обучение завершается. Иначе – совершается еще один проход.

Листинг 1. Функция main



Листинг 1 (продолжение). Функция main



Листинг 2. Метод network.fit

