

Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра интеллектуальных информационных технологий

**Отчет по лабораторной работе №1
по курсу «МРЗвИС»
на тему:
«Сжатие графической информации линейной рециркуляционной
сетью»**

Выполнил студент группы 021703:

Чиж Г.И.

Проверил:

Жук А. А.

**МИНСК
2022**

Цель: Ознакомиться, проанализировать и получить навыки реализации модели линейной рециркуляционной сети для задачи сжатия графической информации.

Задание: Реализовать модель линейной рециркуляционной сети с адаптивным шагом обучения с нормированными весами.

Программа реализована на языке Python.

В процессе работы исходные изображения преобразуются в трёхмерные матрицы (с помощью библиотеки `matplotlib`), разбиваются на квадратные блоки (размер блоков задаётся пользователем), каждый блок проходит через двухслойную нейронную сеть: в первом преобразовании вектор цветов блока сжимается до количества нейронов первого скрытого слоя, во втором преобразовании он разжимается до исходного состояния. Сжатие и разжатие происходит путём перемножения матриц входа и весов (матриц весов две: для сжатия и разжатия). В процессе обучения обучающая выборка из L блоков (в нашем случае L это общее количество блоков) проходит через нейронную сеть и корректирует матрицы весов такое количество раз, которое нужно, чтобы снизить суммарную квадратическую ошибку до порогового значения. Корректировка матриц и подсчёт ошибки происходит по известным заданным формулам.

После обучения двух матриц весов до состояния, когда ошибка приемлема, каждый блок проходит через эти матрицы и сохраняется в векторах, которые потом трансформируются в готовые матрицы изображений. Из этих матриц с помощью библиотеки `matplotlib` потом формируются нужные изображения.

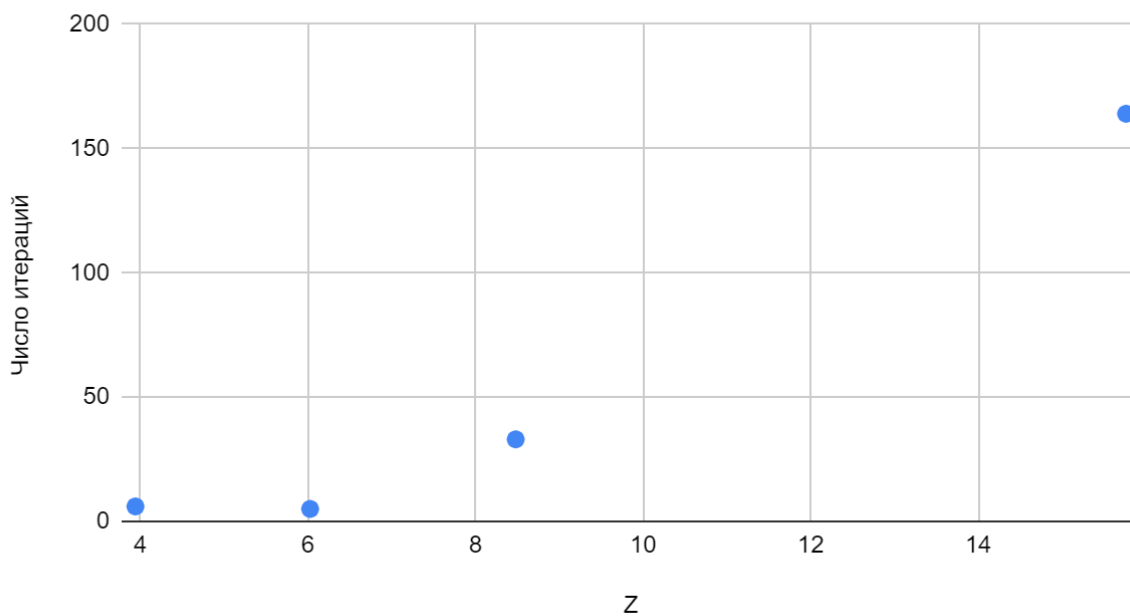
Хотя первый скрытый слой (сжатое изображение) необязательно должен быть выводимым в качестве изображения, он выводится в целях наглядности (а потому количество нейронов на этом слое должно соответствовать формуле $a^2 * 3$, $a \in \mathbb{Z}$ (кратно 3, так как количество цветов в пикселе 3; квадрат, так как сжатые блоки квадратные)).

Графики

1. График зависимости количества итераций от коэффициента сжатия Z
 $e = 2500$ $\alpha = 0.0001$

Z	Число итераций		N	L	p
3,941565991	6		48	3249	12
15,76387345	164		48	3249	3
6,026307436	5		75	2025	12
8,480014244	33		108	1764	12

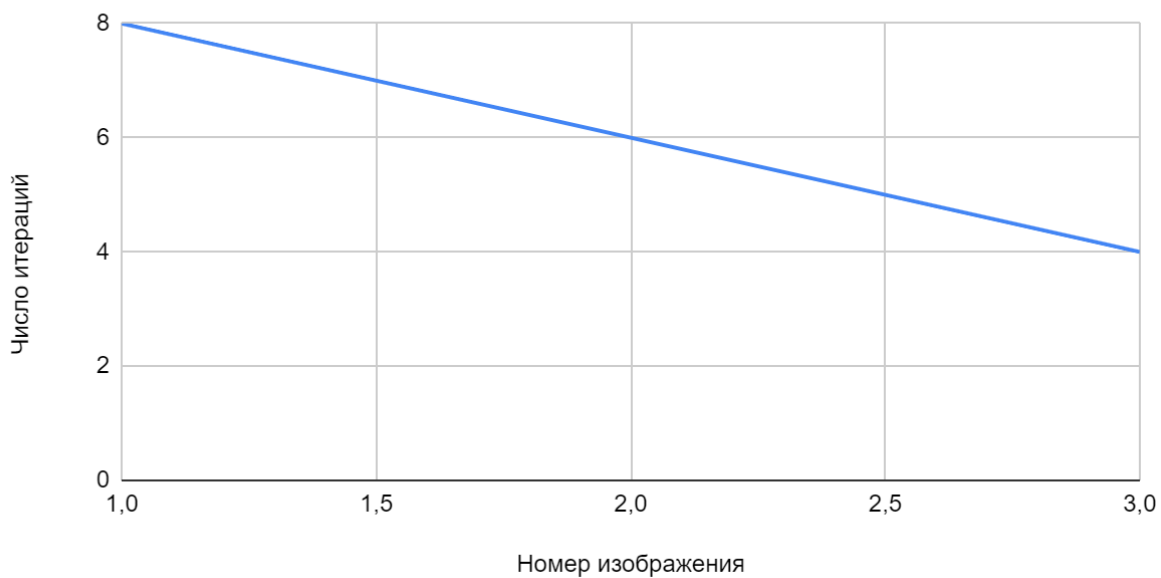
Число итераций относительно параметра "Z"



2. График зависимости числа итераций для разных изображений
 $e = 2500$ $\alpha = 0.0001$ $p = 12$ размер блоков = 4×4

Номер изображения	Число итераций		Размеры
1	8		340x604
2	6		256x256
3	4		225x225

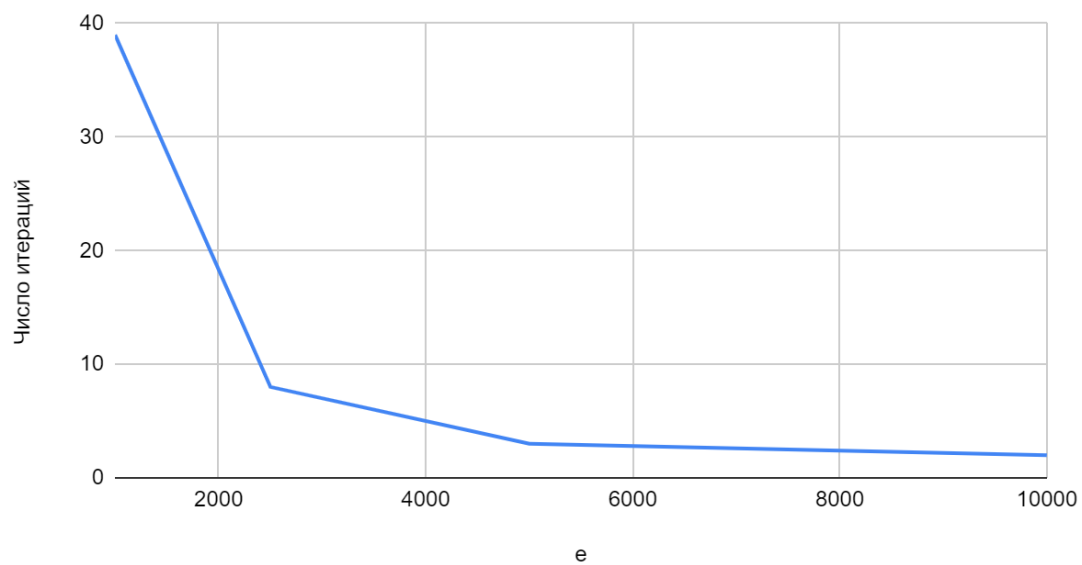
Число итераций относительно параметра "Номер изображения"



3. График зависимости числа итераций от максимальной ошибки ϵ
 $\alpha = 0.0001$ $p = 12$ размер блоков = 4×4

ϵ	Число итераций
1000	39
2500	8
5000	3
10000	2

Число итераций относительно параметра "e"

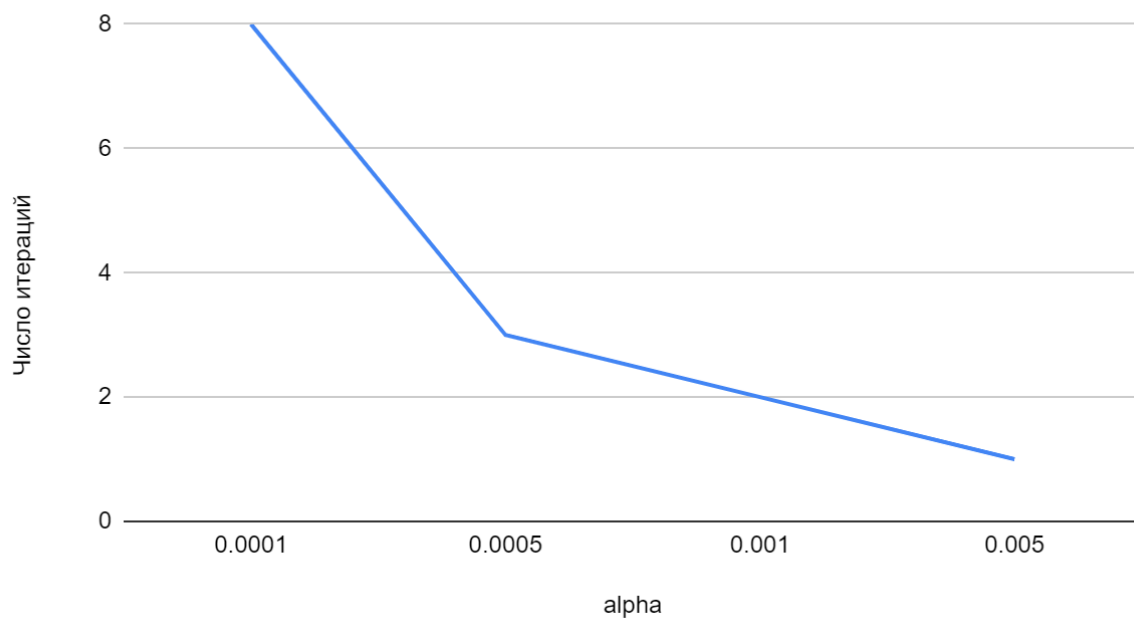


4. График зависимости количества итераций от коэффициента обучения alpha

$\epsilon = 2500$ $p = 12$ размер блоков = 4x4

alpha	Число итераций
0.0001	8
0.0005	3
0.001	2
0.005	1

Число итераций относительно параметра "alpha"



Выводы:

Количество итераций растёт в нелинейной зависимости от коэффициента сжатия.

Число итераций растёт с размером картинки.

Число итераций падает с ростом максимальной ошибки (за счёт конечного совпадения восстановленной картинки с оригиналом)

Число итераций растёт с увеличением коэффициента обучения (за счёт точности снижения ошибки)