

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ**
Федеральное государственное автономное образовательное учреждение
высшего образования
**«Национальный исследовательский
Нижегородский государственный университет им. Н.И. Лобачевского»
(ННГУ)**

Институт информационных технологий, математики и механики

Направление подготовки: «Прикладная математика и информатика»
Магистерская программа: «Вычислительные методы и суперкомпьютерные
технологии»

Образовательный курс «Глубокое обучение»

ОТЧЕТ
по лабораторной работе №4

**Начальная настройка весов полностью связанных и
сверточных нейронных сетей с помощью библиотеки MXNet**

Выполнили:
студенты группы 381703-3м
Гладкова Татьяна
Крутоборезская Ирина
Крюкова Полина
Подчищаева Мария

Нижний Новгород
2018

Содержание

Цели	3
Задачи	4
Решаемая задача	5
Выбор библиотеки.....	6
Метрика качества решения задачи	6
Тренировочные и тестовые наборы данных.....	6
Конфигурации нейронных сетей	7
Разработанные программы/скрипты	8
Результаты экспериментов	8

Цели

Цель настоящей работы состоит в том, чтобы использовать методы обучения без учителя для настройки начальных значений весов сетей, построенных при выполнении предшествующих практических работ.

Задачи

Выполнение лабораторной работы предполагает решение *следующих задач*:

1. Выбор архитектур нейронных сетей, построенных при выполнении предшествующих практических работ.
2. Выбор методов обучения без учителя для выполнения настройки начальных значений весов сетей.
3. Применение методов обучения без учителя к выбранному набору сетей.
4. Сбор результатов экспериментов.

Решаемая задача

Была выбрана задача бинарной классификации: «кошки» - «собаки». Были использованы картинки из наборов данных <https://www.kaggle.com/tongpython/cat-and-dog> и <https://www.kaggle.com/c/dogs-vs-cats/data>. Получившийся набор состоит из 35029 изображений.



Рис. 1 Пример изображения из класса «кошки»



Рис. 2 Пример изображения из класса «собаки»

С помощью скрипта на python данные были преобразованы к размеру 64×64. С помощью скрипта `im2res.py`, который входит в библиотеку MXNet, изображения были сконвертированы в формат `.res`.

Выбор библиотеки

Для выполнения лабораторных работ выбрана библиотека MXNet для языка программирования Python.

На этапе проверки корректности установки библиотеки выполнена разработка и запуск тестового примера сети для решения задачи классификации рукописных цифр набора данных MNIST. Достигнута точность 0.9225.

Метрика качества решения задачи

В качестве метрики точности решения используется отношение угаданных животных ко всем в тестовой выборке:

$$Accuracy = \frac{Correctly\ answers\ count}{Images\ count}$$

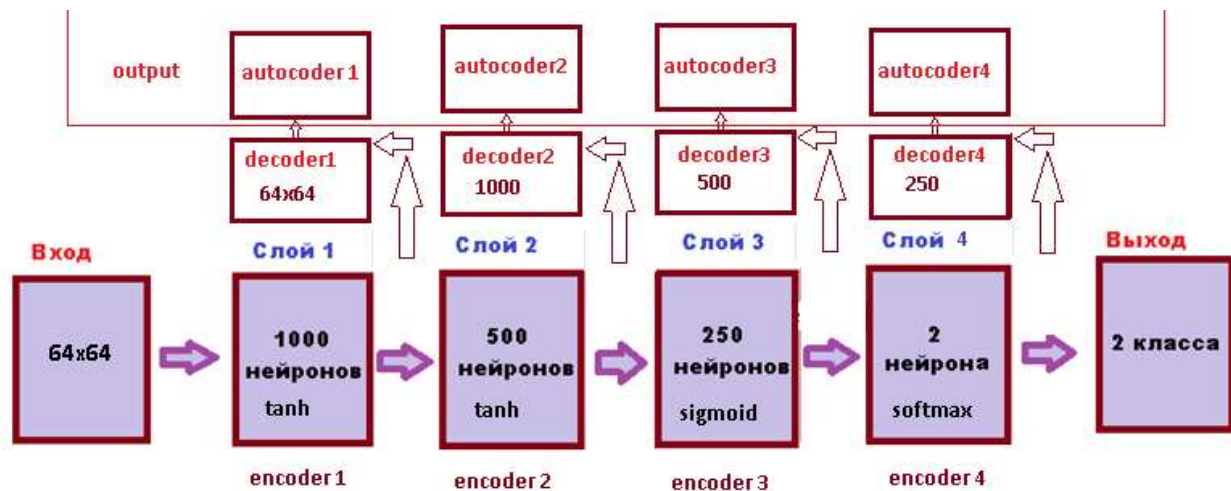
Тренировочные и тестовые наборы данных

В качестве тренировочной выборки используем тренировочную выборку первого и второго наборов данных, всего 16500 изображений котов и 16505 изображений собак. В качестве тестовой выборки используем тестовую выборку только из первого набора данных, т.к. во втором наборе данных тестовая выборка не разбита на изображения котов и собак. Всего в тестовой выборке 2042 изображения, котов и собак поровну.

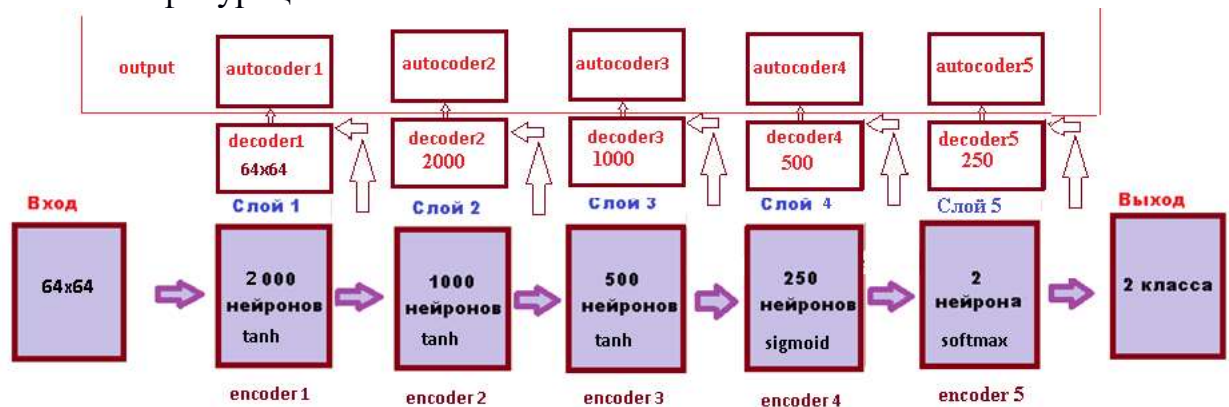
Конфигурации нейронных сетей

В данной работе были рассмотрены четыре конфигурации полносвязных нейронных сетей с 4-мя и 5-мя скрытыми слоями, которые были реализованы в лабораторной работе №2. Для них была реализована начальная настройка весов с помощью автокодировщиков. Для данной лабораторной работы данные были сжаты до размера 64x64.

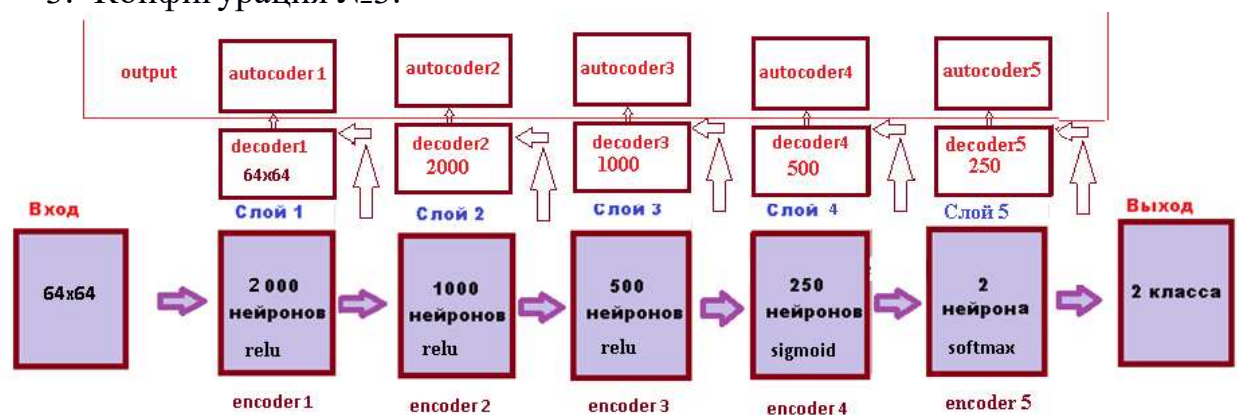
1. Конфигурация №1.



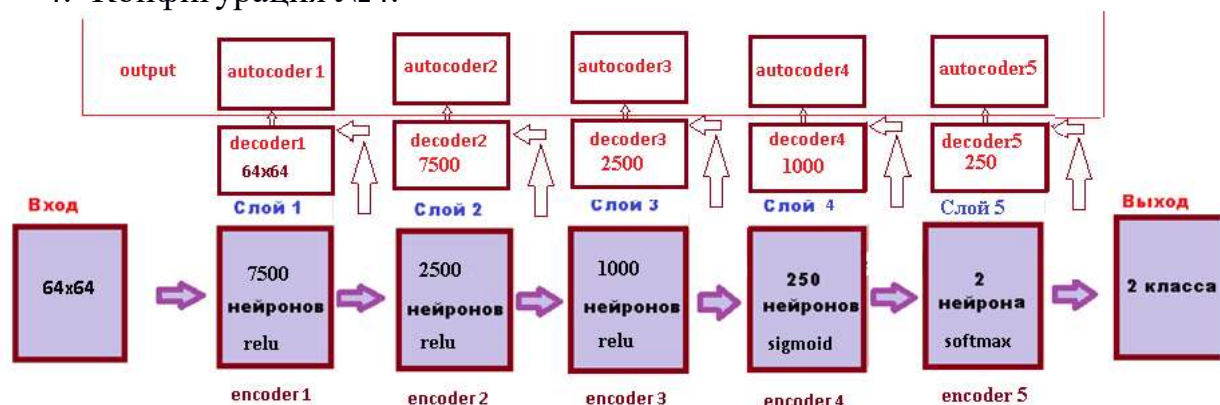
2. Конфигурация №2.



3. Конфигурация №3.



4. Конфигурация №4.



Разработанные программы/скрипты

В директории расположены четыре конфигурации построенных полносвязных нейронных сетей. Соответствия построенных конфигураций и конфигураций в директории:

- fcnn_tts.py – первая нейронная сеть tanh-tanh-sigmoid-softmax,
- fcnn_ttts.py – вторая нейронная сеть tanh-tanh-tanh-sigmoid-softmax,
- fcnn_rrrs1.py – третья нейронная сеть relu-relu-relu-sigmoid-softmax,
- fcnn_rrrs2.py – четвертая нейронная сеть relu-relu-relu-sigmoid-softmax,
- autoencoder.py – автокодировщик, считает предварительные параметры.

Результаты экспериментов

В работе рассмотрены 4 конфигурации.

Параметры обучения:

количество эпох – 1,

скорость обучения – 0.01.

№ Конфигурации	Точность на тестовом множестве	Время, с
1	0.5	25.35
2	0.5	46.56
3	0.67	37.83
4	0.70	135.99

В ходе обучения без учителя точность классификации сетей либо уменьшилась, либо осталась неизменной по сравнению с результатами из лабораторной работы №2 – случай, когда веса инициализируются случайными значениями. Это могло произойти из-за сжатия картинок с размера 128x128 до 64x64.