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**

**Кафедра математического обеспечения и суперкомпьютерных технологий**

Направление подготовки «Прикладная математика и информатика»

Магистерская программа «Системное программирование»

**Отчет по лабораторной работе**

**«Разработка полностью связанной нейронной сети»**

Выполнили:

студенты группы 381603м4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Гладилов, Волокитин, Левин,

Новак

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ под

Нижний Новгород

2017

**содержание**

[1 Постановка задачи 3](#_Toc500364255)

[1.1 Постановка задачи 3](#_Toc500364256)

[1.2 Математическое описание задачи 3](#_Toc500364257)

[2 Тренировочные и тестовые наборы данных 5](#_Toc500364258)

[2.1 Описание набора данных 5](#_Toc500364259)

[2.2 Метрика качества решения задачи 5](#_Toc500364260)

[3 Формат хранения данных 6](#_Toc500364261)

[4 Формат входа сети 7](#_Toc500364262)

[5 Описание файлов репозитория 8](#_Toc500364263)

[6 Тестовые конфигурации сетей 9](#_Toc500364264)

[7 Результаты экспериментов 13](#_Toc500364265)

[8 Итоги 14](#_Toc500364266)

.

# Постановка задачи

## Постановка задачи

В рамках выполнения данной лабораторной работы необходимо изучить, реализовать и протестировать различные конфигурации нейронных сетей, для выбранной задачи компьютерного зрения с использованием библиотеки глубокого обучения. Список задач, которые необходимо решить в данной лабораторной работе:

* Выбор библиотеки глубокого обучения, установка и тестирование.
* Выбор практической задачи компьютерного зрения для выполнения практических работ.
* Разработка программ/скриптов для подготовки тренировочных и тестовых данных
* Разработка различных архитектур нейронных сетей (различные типы слоев, их число, типы функции активаций и.т.д) в формате, который принимается выбранной библиотекой.
* Обучение разработанных глубоких моделей.
* Тестирование обученных глубоких моделей.

В качестве практической задачи нашей группой была выбрана тема: Предсказание возраста и пола человека по фотографии его лица.

## Математическое описание задачи

Общую структуру сети можно изобразить следующим образом:

Вхходной слой

Выходной слой

Скрытые слои

Входной слой

1. Общая схема сети

Содержание внутренних слоев зависит от того, какого типа сеть мы рассматриваем (многослойная полносвязная, сверточная, и.т.д).

На входе имеется набор данных представляющих из себя трехканальные изображения размера .

В случае задачи классификации, классической метрикой ошибки является кросс-энтропия:

Где – число элементов в выборке, – истинная метка, - полученная метка.

Задача обучения нейронной сети сводится к задаче оптимизации функции ошибки по всем синаптическим весам:

Для обучения нейронной сети обычно используется метод обратного распространения ошибки. Метод обратного распространения ошибки определяет стратегию выбора параметров сети с использованием градиентных методов оптимизации. Градиентные методы на каждом шаге уточняют значения параметров, по которым проводится оптимизация, согласно формуле:

где определяет сдвиг значений параметров, – скорость обучения – параметр обучения, который определяет «скорость» движения в направлении минимального значения функции, – направление в многомерном пространстве параметров нейронной сети. В классическом методе обратного распространения ошибки направление движения совпадает с направлением антиградиента.

Общая схема работы метода обратного распространения ошибки:

1. Инициализация весов
2. Прямой проход нейронной сети. Вычисление сигналов нейронов и значений функций активации
3. Вычисление значения целевой функции и градиента этой функции.
4. Обратный проход нейронной сети в направлении от выходного слоя к входному слою, и корректировка синаптических весов.
5. Повторение этапов 2 – 4 до момента выполнения критериев остановки

# Тренировочные и тестовые наборы данных

## Описание набора данных

В качестве набора данных был использована часть данных из набора (<https://data.vision.ee.ethz.ch/cvl/rrothe/imdb-wiki/>). Набор представляет из себя фотографии знаменитостей полученных с сайта Wikipedia. Всего доступно 62,328 картинок, каждая из которых имеет следующее описание, доступное в файле .mat:

* **dob –** дата рождения
* **fhoto\_taken –** год, когда было сделано фото
* **full\_path –** путь до изображения
* **gender:** - пол, 0 если женщина 1 если мужчина, NaN если не определен
* **name** – имя
* **face\_location –** координаты лица (для полных картинок)

Данное описание позволяет при помощи скриптов получить необходимые разметку в формате .lst или json состоящую из:

* Путь до картинки
* Пол

1. Примеры изображений из набора:

Исходный набор был разбит на тестовую и тренировочную выборки в соотношении:

* 80% - тренировочная выборка (47000 примеров)
* 20% - тестовая выборка (12000 примеров)

## Метрика качества решения задачи

Для оценки качества полученной модели используются данные которые предоставляешь библиотека CAFFE, при запуске в тестовом режиме. Точность определяется по формуле:

# Формат хранения данных

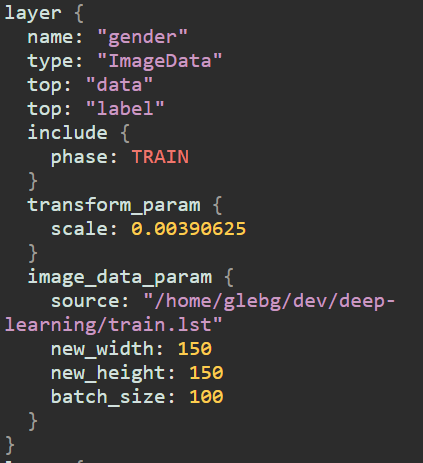
На сайте представлены архивы содержащие картинки в формате jpg. На выбор предоставляются две версии данных:

* Полная картинка
* Часть изображения, содержащая лицо

Каждый из архивов содержит файл с разметкой, структура которого описана в разделе 2. Изображения представлены в различных разрешениях, для удобства использования они были приведены к разрешению с применением библиотеки OpenCV.

# Формат входа сети

Для описания входа сети в библиотеке Caffe используется слой ImageData.



Описание значений параметров слоя:

* top – Указывает на то, какие данные приходят в слой, в данном случае это исходная картинка и метка класса
* phase – TRAIN ( TEST) –режим в котором используется слой
* transform\_param – описание преобразований над входными данными. В данном случае выполняется нормировка на
* Source – файл \*.lst где хранятся изображения и метки класса
* new\_width/new\_height – размеры входного тензора
* batch\_size – размер пачки картинок.

В данной модели мы используем RGB картинки уменьшенные до размера и нормализованные на

# Описание файлов репозитория

Структура проекта:

* Readme.md – общее описание задачи
* Lab02 – папка содержащая выполненные результаты по работе

Отчет по проделанной работе находится в Lab02/report

Исходные скрипты для запуска сети, полученные модели, а также \*.prototxt находятся в Lab02/src

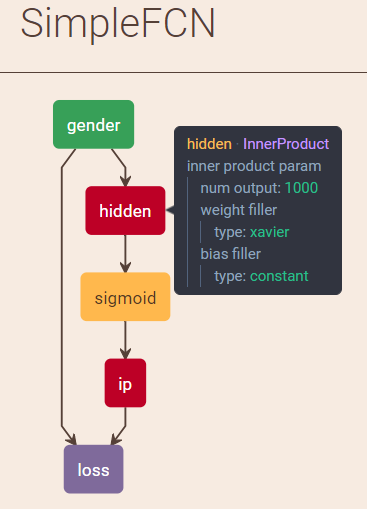
* Train.lst – описание тренировочной выборки
* Test.lst – описание тестовой выборки
* Wiki.mat – исходная разметка
* Resize.py – скрипт для изменения размеров изображений
* Exp1 …Exp6 – папки с результатами экспериментов

Каждая папка включает в себя четыре файла:

* gender\_fcn.prototxt – файл с описанием сети в Caffe-формате
* gender\_fcn\_solver.prototxt – описание параметров обучения
* Shell скрипт для запуска фаз тренировки и тестирования
* Out.log – сохраненный вывод библиотеки Caffe, который содержит подробное описание фаз тренировки, фазы тестирования, а также результаты

# Тестовые конфигурации сетей

* **SimpleFCN:**



1. Cеть с одним скрытым слоем, имеющим 1000 нейронов

Описание конфигурационного файла:

test\_interval: 1000

base\_lr: 0.05

momentum: 0.9

weight\_decay: 0.0005

lr\_policy: "step"

stepsize: 2500

gamma: 0.1

display: 0

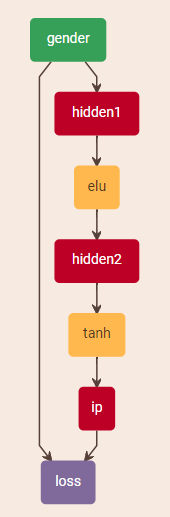
max\_iter: 10000

snapshot: 500

solver\_mode: GPU

type: "SGD"

* **Elu+Tanh**



1. Два скрытых слоя. 400 и 50 нейронов.

Описание конфигурационного файла:

test\_interval: 1000

base\_lr: 0.05

momentum: 0.9

weight\_decay: 0.0005

lr\_policy: "step"

stepsize: 2500

gamma: 0.1

display: 0

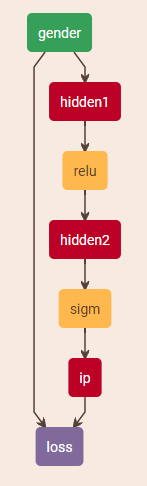
max\_iter: 10000

snapshot: 500

solver\_mode: GPU

type: "SGD"

* **Relu+Sigmoid**

****

1. Два скрытых слоя.400 и 50 нейронов. Активации Relu+Sigmoid

Описание конфигурационного файла:

test\_interval: 500

base\_lr: 0.01

momentum: 0.9

weight\_decay: 0.0005

lr\_policy: "fixed"

display: 500

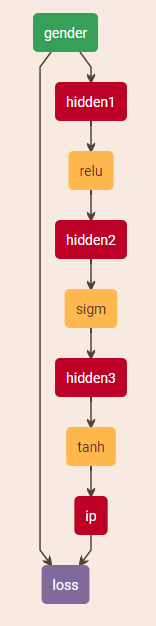
max\_iter: 2000

snapshot: 500

solver\_mode: GPU

type: "SGD"

* **Three layers**



1. Три скрытых слоя. 400, 100, 20 нейронов соответственно

Описание конфигурационного файла:

test\_interval: 500

base\_lr: 0.01

momentum: 0.9

weight\_decay: 0.0005

lr\_policy: "fixed"

display: 500

max\_iter: 2000

snapshot: 500

solver\_mode: GPU

type: "SGD"

# Результаты экспериментов

|  |  |  |
| --- | --- | --- |
| **Конфигурация сети** | **Время обучения**  **(секунды)** | **Точность** |
| **SimpleFCN+sigm** | **720** | **0.8338** |
| **SimpleFCN+tanh** | **534** | **0.786** |
| **Elu+tanh** | **540** | **0.7878** |
| **Three layers** | **560** | **0.8158** |
| **Relu+Sigmoid** | **653** | **0.8156** |

Кроме этого, был произведен ряд экспериментов, результаты которых оказались менее удачными. Например на сети с тремя скрытыми слоями (1000, 750 и 250), а также Relu активацией loss выдала .

На некоторых конфигурациях не достигалось нужной точности. Конфигурации моделей доступны в репозитории.

# Итоги

В рамках данной лабораторной работы были решены следующие задачи:

* Выбрана библиотека глубокого обучения (Caffe). Произведена её установка как на кластер так и на локальную машину. Подтверждена корректность установки.
* Выбрана практическая задача компьютерного зрения.
* Разработаны скрипты для подготовки данных, тренировки и тестирования сетей.
* Разработаны различные архитектуры нейронных сетей.
* Произведено обучение и тестирование разработанных глубоких моделей.