

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

Нижний Новгород
2017

СОДЕРЖАНИЕ

1	ПОСТАНОВКА ЗАДАЧИ	3
1.1	Постановка задачи	3
1.2	Математическое описание задачи	3
2	ТРЕНИРОВОЧНЫЕ И ТЕСТОВЫЕ НАБОРЫ ДАННЫХ	5
2.1	Описание набора данных	5
2.2	Метрика качества решения задачи	5
3	ФОРМАТ ХРАНЕНИЯ ДАННЫХ	6
4	ФОРМАТ ВХОДА СЕТИ	7
5	ОПИСАНИЕ ФАЙЛОВ РЕПОЗИТОРИЯ	8
6	ТЕСТОВЫЕ КОНФИГУРАЦИИ СЕТЕЙ	9
7	РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ	13

1 Постановка задачи

1.1 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.

1.2 Математическое описание задачи

Общую структуру сети можно изобразить следующим образом:

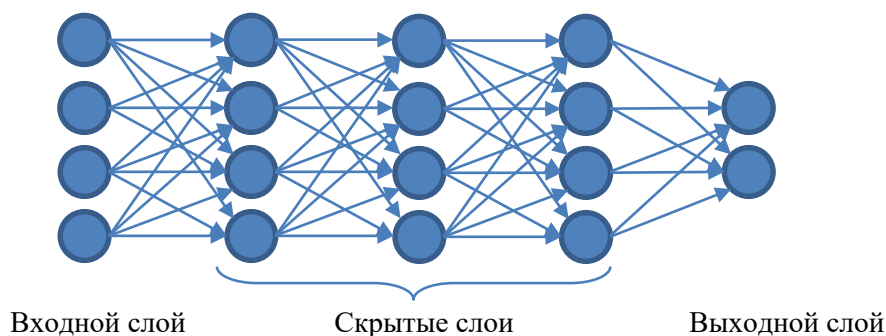


Рис. 1. Общая схема сети

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 $M \times N$.

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

$$E(w) = -\frac{1}{L} \sum_{k=1}^L (y^k \ln u^k + (1 - y^k) \ln(1 - u^k)).$$

Где L – число элементов в выборке, y^k – истинная метка, u^k – полученная метка.

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

$$\min_w E(w)$$

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

$$w(k + 1) = w(k) + \Delta w,$$

где $\Delta w = \eta p(w)$ определяет сдвиг значений параметров, $\eta, 0 < \eta < 1$ – *скорость обучения* – параметр обучения, который определяет «скорость» движения в направлении минимального значения функции, $p(w)$ – 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

2 Тренировочные и тестовые наборы данных

2.1 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
- **photo_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



Рис. 2. 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)

2.2 Метрика качества решения задачи

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

$$accuracy = \frac{TRUE}{N}, \text{ где } N - \text{общее количество примеров}$$

3 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 150x150 с применением библиотеки OpenCV.

4 Формат входа сети

Для описания входа сети в библиотеке Caffe используется слой ImageData.

```
layer {
  name: "gender"
  type: "ImageData"
  top: "data"
  top: "label"
  include {
    phase: TRAIN
  }
  transform_param {
    scale: 0.00390625
  }
  image_data_param {
    source: "/home/glebg/dev/deep-
learning/train.lst"
    new_width: 150
    new_height: 150
    batch_size: 100
  }
}
```

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 255
- 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 150x150 и нормализованные на 255

5 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
- 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

6 Тестовые конфигурации сетей

- SimpleFCN:

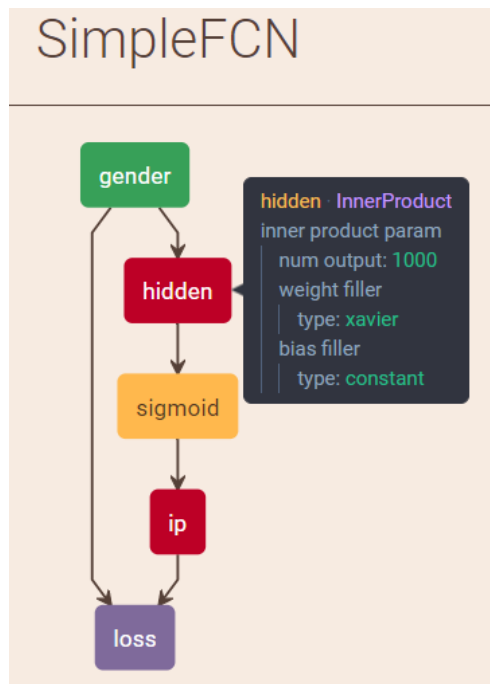


Рис. 3. Сеть с одним скрытым слоем, имеющим 1000 нейронов

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**

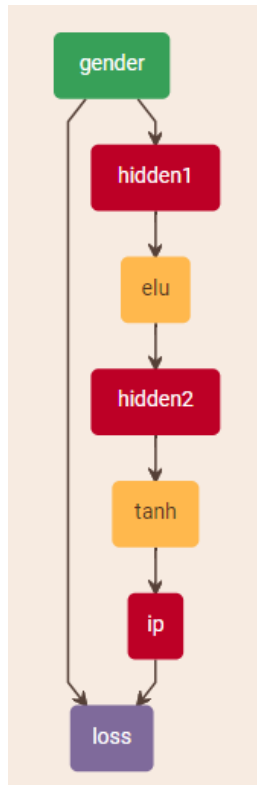


Рис. 4. Два скрытых слоя. 400 и 50 нейронов.

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

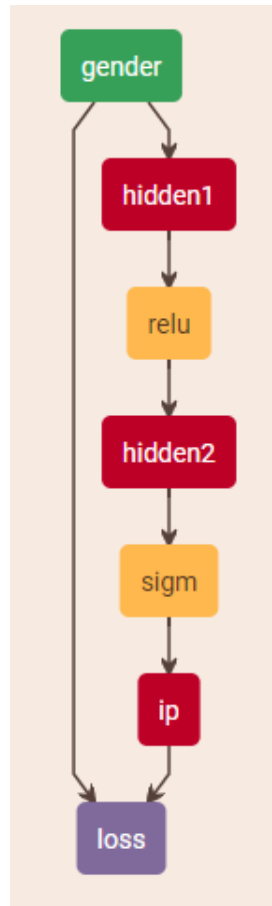


Рис. 5. Два скрытых слоя.400 и 50 нейронов. Активации Relu+Sigmoid

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**

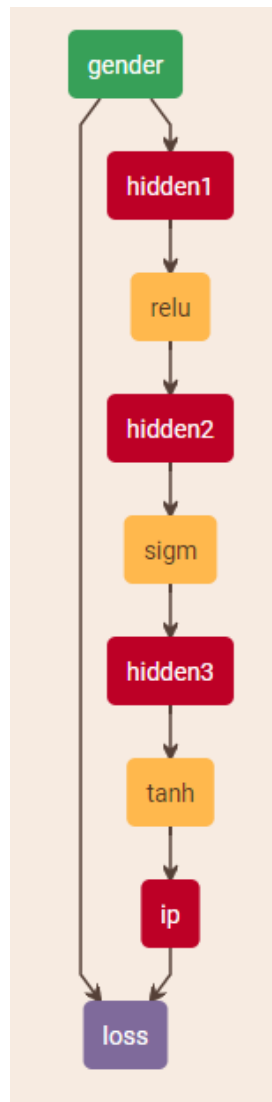


Рис. 6. Три скрытых слоя. 400, 100, 20 нейронов соответственно

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"
```

7 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 *NaN*.

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