## МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского

Институт информационных технологий, математики и механики

Кафедра математического обеспечения и суперкомпьютерных технологий

Направление подготовки

02.04.02. Фундаментальная информатика и информационные технологии

Направленность образовательной программы магистерская программа «Компьютерная графика и моделирование живых и технических систем»

Отчёт

по методам глубокого обучения для решения задач компьютерного зрения

на тему:

«Применение переноса обучения для решения задачи, поставленной во второй лабораторной работе»

> Квалификация (степень) магистр

> > Форма обучения очная

Выполнили: студ	центы группы 381706-3М
X	Храмов Илья Валерьевич
	Подпись
P	еунова Ольга Алексеевна
	Подпись
F	Зоеводин Андрей Михайлович
	Подпись

Н. Новгород2018 г.

# Содержание

Постановка задачи	3
Тренировочные и тестовые наборы данных	
Метрика качества решения	
Разработанные программы/скрипты	
Тестовые конфигурации сетей	
Результаты	
Литература	12

### Постановка задачи

### Цель

Цель настоящей работы состоит в том, чтобы исследовать возможности переноса обучения для решения целевой задачи, выбранной изначально для выполнения практических работ.

#### Задачи

Выполнение практической работы предполагает решение следующих задач:

- 1. Поиск исходной задачи (близкой по смыслу к целевой задаче) и поиск натренированной модели для решения исходной задачи.
- 2. Выполнение трех типов экспериментов по переносу знаний (типы экспериментов описаны в лекции).
- 3. Сбор результатов экспериментов.

### Тренировочные и тестовые наборы данных

#### Целевая задача

Задача — классификация комиксов. Данные получены из [5]. 86 классов сокращены до 14 классов с целью убрать классы, в которых выборка не репрезентативна. Размер изображений в каждом классе 224\*224. Изображения 3 канальные.

No	Категории	Размер тренировочной выборки	Размер тестовой выборки
1	Aquaman v7	1088	282
2	Batgirl v4	1088	293
3	Batman v2	1509	384
4	Batwing	616	167
5	Batwoman	739	187
6	Catwoman v4	1047	256
7	Green Arrow	1040	256
8	Green Lantern	1301	345
9	Harley Quinn	671	165
10	Nightwing v3	558	147
11	Red Lanterns	767	197
12	Sinestro	619	143
13	Supergirl v6	701	200
14	Wonder Woman	1233	292
		12977	3314

Таблица 1. Размер выборки в каждом классе тренировочного и тестового множеств целевой задачи.

Изображения хранятся в формате JPEG.

На вход сети подаются бинарные файлы с расширениями .rec (изображения), .idx (индексы изображений) (Рис. 1).

### Исходная задача

Исходная задача — ImageNet.

## **Binary Record**

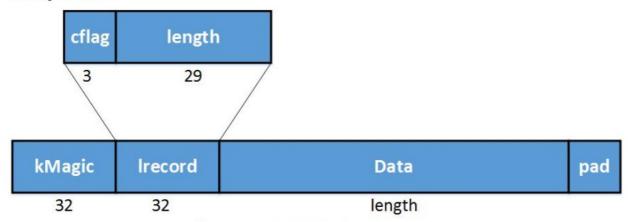


Рисунок 1. Формат хранения в MXNet [8].

- kMagic начало записи;
- lrecord длина (length) и продолжительность записи (cflag);
- Data данные;
- pad пространство для выравнивания до 4 байт.

## Метрика качества решения

В качестве метрики для оценки качества решения задачи [5] выбрана "Точность" ("Ассигасу"). В терминологии MXNet — это отношение количества правильно предсказанных сэмплов к общему количеству сэмплов (Рис. 2).

$$\operatorname{accuracy}(y,\hat{y}) = rac{1}{n} \sum_{i=0}^{n-1} \mathbb{1}(\hat{y_i} == y_i)$$

Рисунок 2. Определение "Accuracy" и МАЕ в MXNet [7]. Более подробную информацию о метриках можно найти в [7].

## Разработанные программы/скрипты

- reduce\_dataset.py скрипт для вычленения из оригинального набора данных наиболее репрезентативных категорий.
- prepare dataset.py скрипт для подготовки данных под mxnet.
- lab5.py скрипт для обучения автокодировщика.
- resize dataset.py скрипт для изменения размерности данных.
- load\_dataset.py функции для загрузки данных в скрипт обучения.
- blocks.py основные конструкционные блоки сетей.
- fit.py универсальный скрипт обучения, формирующий все необходимые отчёты.
- parse log.py скрипт, формирующий сводную таблицу из журнала обучения.

### Тестовые конфигурации сетей

На рисунке 3 представлена сверточная сеть squeeznet 1.1, которую обучали на наборе данных ImageNet.

На рисунке 4 представлена сверточная сеть Сеть 1 (переделанная squeeznet 1.1), построенная для решения целевой задачи.

Варианты переноса обучения:

- 1. Сеть 1 обученная с нуля.
- 2. Сеть 1 с инициализированными параметрами сверточных слоев из squeeznet 1.1 (параметры фиксируются при обучении) и новым классификатором.
- 3. Сеть 1 с инициализированными параметрами сверточных слоев из squeeznet 1.1 (параметры не фиксируются при обучении) и новым классификатором.

Рисунок 3. Сеть squeeznet 1.1.

Рисунок 4. Сеть 1.

## Результаты

В таблице 2 приведены конфигурация системы и программное обеспечение, с помощью которых проводилось обучение и тестирование построенных моделей.

Параметры	Версия
Операционная система	Windows10
GPU	NVIDIA GeForce GTX 1080  (частота процессора — OC Mode – GPU Boost Clock: 1835 MHz, GPU Base Clock: 1695 MHz Gaming Mode (Default) - GPU Boost Clock: 1809 MHz, GPU Base Clock: 1670 MHz; шина передачи данных – PCI Express 3.0; видеопамять - GDDR5X 8GB; количество ядер — 2560; частота памяти - 10010 MHz; интерфейс памяти — 256-bit;)
CUDA	9.2
Python	3.7.1
MXNet	1.3.0

Таблица 2. Конфигурация системы.

## В таблице 3 приведены параметры обучения.

Варианты переноса обучения	Оптимизатор	Скорость обучения	Количество эпох	Размер batch
1	SGD (стохастическ ий градиентный спуск)	0.004	120	10
2	SGD	0.0004	120	10
3	SGD	0.0004	120	10

Таблица 3. Параметры обучения.

V	Варианты переноса обучения		
Характеристики	1	2	3
Среднее время обучения за одну эпоху, с	31,96	19,07	32,1
Качество решения на тренировочном наборе (Accuracy), %	82,71	50,21	94,27
Качество решения на тестовом наборе (Ассигасу), %	47,43	46,16	57,22
Номер эпохи с достигнутым максимальным качеством решения на тренировочном наборе	120	100	116
Максимальное качество решения на тренировочном наборе (Accuracy), %	82,71	51,67	95,05
Номер эпохи с достигнутым максимальным качеством решения на тестовом наборе	103	114	86
Максимальное качество решения на тестовом наборе (Accuracy), %	50,75	46,56	58,1

Таблица 4. Результаты экспериментов. Конфигурация сетей приведена в "Тестовые конфигурации сетей".

### Анализ результатов

Из представленных результатов можно сделать выводы:

- 1. Перенос обучения помог улучшить точность на тренировочной выборке (Сеть6, лаб.раб.3: 42,72).
- 2. Выборка нерепрезентативная. Малое количество изображений на один класс, как тренировочной, так и тестовой выборок. Некоторые изображения в тестовой и тренировочной выборках слишком отличаются (например, тренировочная выборка нарисован Бэтман на черном фоне, тестовая выборка нарисован Бэтман на светлом фоне). Герои одних комиксов могут встречаться в других комиксах (например, Harley Quinn в Ваtman и наоборот).
- 3. Плохой выбор скорости обучения. Если данный параметр слишком большой, то точность будет приблизительно 11% (измерялось для разных конфигураций сетей), то есть мы будем бесконечно «прыгать» через точку минимума. Если мы берем параметр слишком маленьким, то мы "застреваем" в локальном минимуме.
- 4. Можно попробовать сети, содержащие другую архитектуру Inception.

### Литература

- 1. MNIST dataset [http://yann.lecun.com/exdb/mnist].
- 2. OpenCV [http://opencv.org].
- 3. Материалы Летней межвузовской школы 2016 [https://github.com/itseez-academy/itseez-ss-2016- theory], [https://github.com/itseez-academy/itseez-ss-2016-practice].
- 4. Лекциипоглубокомуобучению:https://sites.google.com/site/kustikovavalentina/studentam/kurs-glubokoe-obucenie,2018.
- 5. Исходные данные <a href="https://www.kaggle.com/cenkbircanoglu/comic-books-classification:">https://www.kaggle.com/cenkbircanoglu/comic-books-classification:</a> kaggle datasets download -d cenkbircanoglu/comic-books-classification.
- 6. Документация MXNet <a href="http://mxnet.incubator.apache.org/test/tutorials/">http://mxnet.incubator.apache.org/test/tutorials/</a>.
- 7. Метрики в MXNet <a href="https://mxnet.incubator.apache.org/api/python/metric/metric.html">https://mxnet.incubator.apache.org/api/python/metric/metric.html</a>.
- 8. Формат хранения данных в MXNet: <a href="https://mxnet.incubator.apache.org/architecture/note-data-loading.html">https://mxnet.incubator.apache.org/architecture/note-data-loading.html</a>.
- 9. Репозиторий исходных кодов: <a href="https://github.com/okondratieva/DeepLearning">https://github.com/okondratieva/DeepLearning</a>.