

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: студенты группы 381706-3М
Храмов Илья Валерьевич

Подпись

Реунова Ольга Алексеевна

Подпись

Воеводин Андрей Михайлович

Подпись

Н. Новгород
2018 г.

Содержание

Постановка задачи.....	3
Тренировочные и тестовые наборы данных.....	4
Метрика качества решения.....	6
Разработанные программы/скрипты.....	7
Тестовые конфигурации сетей.....	8
Результаты.....	9
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 288*432. Изображения 3 канальные.

№	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 .res (изображения), .idx (индексы изображений) (Рис. 1).

Исходная задача

Исходная задача подобна целевой задаче. Тестовая и тренировочная выборки составлены из оставшихся 72 классов (таблица 2).

№	Категории	Размер тренировочной выборки	Размер тестовой выборки
1	Batman & Robin Eternal	477	133
2	Batman and Robin v2	891	213
3	DC Sneak Peek	387	78
4	Earth 2 - World's End	507	128
5	Justice League	1344	315

6	Justice League 3000	245	59
7	Justice League 3001	227	62
8	Justice League Dark	817	223
9	Justice League International	240	54
10	Justice League of America	469	102
11	Justice League of America`s Vibe	189	31
12	Justice League United	348	82
13	Suicide Squad v4	538	153
14	Superman-Wonder Woman	561	119
		7240	1752

Таблица 2. Размер выборки в каждом классе тренировочного и тестового множеств исходной задачи.

Binary Record

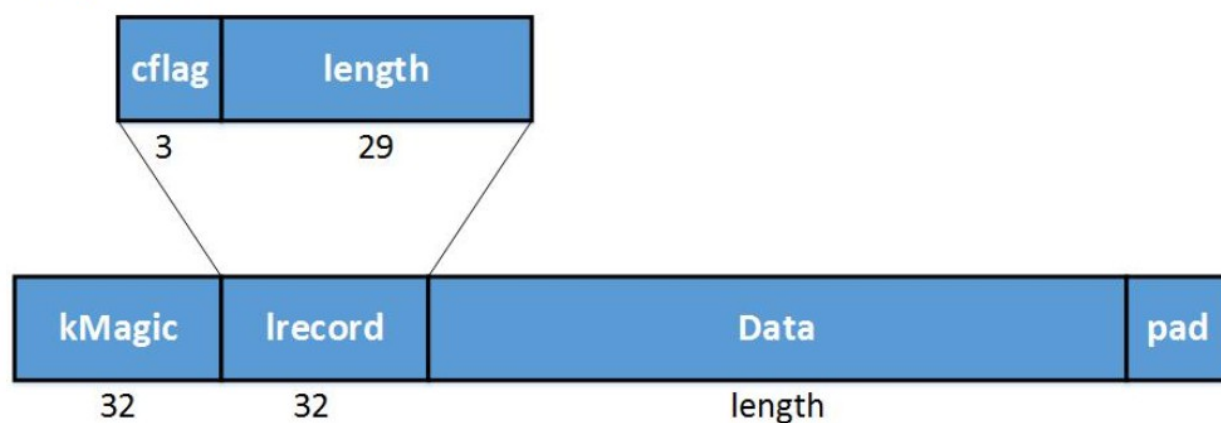


Рисунок 1. Формат хранения в MXNet [8].

- **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” (“Accuracy”). В терминологии MXNet — это отношение количества правильно предсказанных сэмплов к общему количеству сэмплов (Рис. 2).

$$\text{accuracy}(y, \hat{y}) = \frac{1}{n} \sum_{i=0}^{n-1} 1(\hat{y}_i == y_i)$$

Рисунок 2. Определение “Accuracy” и MAE 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 Сеть1, которую обучали для исходной задачи. Для решения целевой задачи были оставлены сверточные слои (фиксированные при обучении) и был добавлен новый классификатор (полносвязные слои). Сеть2 составлена аналогичным способом, за исключением того, что параметры сверточных слоёв меняются при обучении для целевой задачи. Сеть3 исходной задачи была полностью перенесена для решения целевой задачи с сохранением начальных значений параметров.

1.

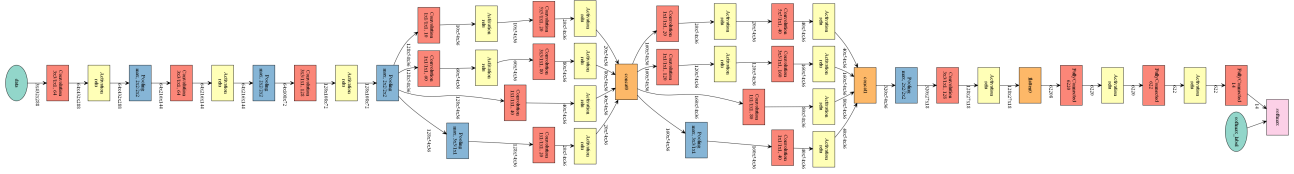


Рисунок 3. Сеть1.

2.

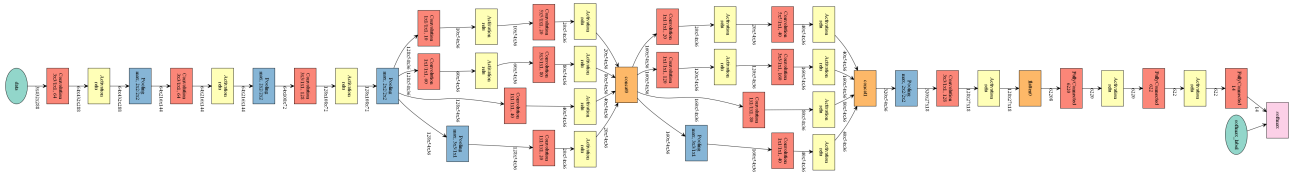


Рисунок 4. Сеть2.

3.

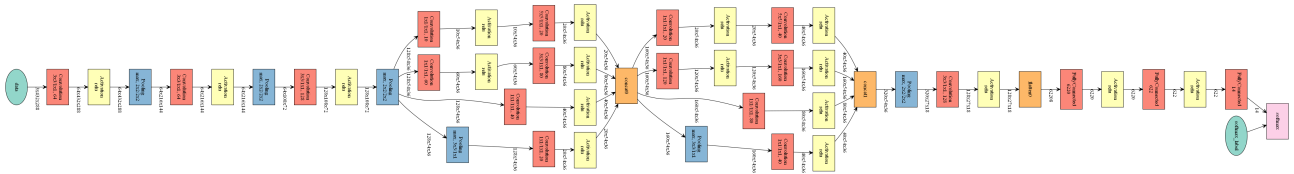


Рисунок 5. Сеть3.

Результаты

В таблице 3 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 3. Конфигурация системы.

В таблице 4 приведены параметры обучения.

Наименование сети	Оптимизатор	Скорость обучения	Количество эпох	Размер batch
Сеть1 (и)	SGD	0.0001	22	10
Сеть1 (ц)	SGD	0.0001	23	10
Сеть2 (и)	SGD	0.0001	22	10
Сеть2 (ц)	SGD	0.0001	25	10
Сеть3 (и)	SGD	0.0001	22	10
Сеть3 (ц)	SGD	0.0001	26	10

Таблица 4. Параметры обучения. И — параметры обучения для исходной задачи; Ц — параметры обучения для целевой задачи.

Задача	Характеристики	Наименование сети		
		Сеть1	Сеть2	Сеть3
Исходная задача	Среднее время обучения за одну эпоху, с	88,01	88,01	88,01
	Качество решения на тренировочном наборе (Assurasy), %	96,09	96,09	96,09
	Качество решения на тестовом наборе (Assurasy), %	30,63	30,63	30,63
	Номер эпохи с достигнутым максимальным качеством решения на тренировочном наборе	13	13	13
	Максимальное качество решения на тренировочном наборе (Assurasy), %	96,09	96,09	96,09
	Номер эпохи с достигнутым максимальным качеством решения на тестовом наборе	9	9	9
	Максимальное качество решения на тестовом наборе (Assurasy), %	30,97	30,97	30,97
Целевая задача	Среднее время обучения за одну эпоху, с	75,62	155,91	156,32
	Качество решения на тренировочном наборе (Assurasy), %	93,09	93,92	93,51
	Качество решения на тестовом наборе (Assurasy), %	30,48	29,88	29,73
	Номер эпохи с достигнутым максимальным качеством решения на тренировочном наборе	19	24	24

	Максимальное качество решения на тренировочном наборе (Assurasy), %	95,26	95,1	94,9
	Номер эпохи с достигнутым максимальным качеством решения на тестовом наборе	15	21	21
	Максимальное качество решения на тестовом наборе (Assurasy), %	30,9	30,06	30,09

Таблица 5. Результаты экспериментов. Конфигурация сетей приведена в “Тестовые конфигурации сетей”.

Анализ результатов

Из представленных результатов можно сделать выводы:

1. Перенос обучения не помог улучшить точность на тренировочной выборке (Сеть4, лаб.раб.3: было 31,33; max=34,13).
2. Сеть1, Сеть2, Сеть3 переобучились.
3. Выборка — нерепрезентативная. Малое количество изображений на один класс, как тренировочной, так и тестовой выборки. 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 Batman и наоборот).
4. 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.
5. Начальная обработка данных. Для конкретного размера изображения необходимо настраивать количество нейронов так, чтобы избежать “узкого горлышка”.

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. Лекции по глубокому обучению: <https://sites.google.com/site/kustikovavalentina/studentam/kurs-glubokoe-obucenie>, 2018.
5. Исходные данные — <https://www.kaggle.com/cenkbircanoglu/comic-books-classification>: kaggle datasets download -d cenkbircanoglu/comic-books-classification.
6. Документация MXNet — <http://mxnet.incubator.apache.org/test/tutorials/>.
7. Метрики в MXNet – <https://mxnet.incubator.apache.org/api/python/metric/metric.html>.
8. Формат хранения данных в MXNet: https://mxnet.incubator.apache.org/architecture/note_data_loading.html.
9. Репозиторий исходных кодов: <https://github.com/okondratieva/DeepLearning>.