

# Задание по практикуму №1

Иван Сомов  
группа 317  
3 курс

15 марта 2018 г.

## 1 Описание работы

В данном отчете содержатся результаты экспериментов, проведенных при выполнении практического задания №1 «Свёрточные автокодировщики для улучшения качества классификации изображений» по курсу «Практикум на ЭВМ 2017-2018» на кафедре ММП ВМК МГУ. Были проведены необходимые эксперименты, и по ним были получены соответствующие выводы. Данное задание было направлено на ознакомление с нейронными сетями и концепцией transfer learning.

## 2 Эксперименты

### 2.1 Эксперимент №1

В этом эксперименте требовалось протестировать на train/test выборках мультиномиальную логистическую регрессию и метод, основанных на деревьях.

Первый подход был реализован в виде однослойной нейронной сети. Результаты работы сети на тестовом датасете приведены в Табл.1. В качестве второго метода использовался xgboost на деревьях. Результаты так же в Табл.1.

Таблица 1: Бейслайны для последующих экспериментов

Метод	Точность	Log loss
Мультиномиальная логистическая регрессия	36 %	6.63828
xgboost	45.9 %	1.55202

### 2.2 Эксперимент №2

В этом пункте предлагалось реализовать свёрточную сеть и посмотреть, как влияют на время и качество различные параметры модели.

### 2.2.1 Исследование влияния размера ядра свертки

Как видно на Рис.2.2.1, с увеличением размера ядра свертки время обработки изображений растет практически линейно, так как число дополнительно обработанных пикселей с увеличением размерности ядра на 1 равно  $(n+1)^2(k-1)^2 - n^2k^2 = (2n+1)(k-1)^2 - n^2(2k+1)$ , где  $n$  – размер свёртки,  $k = \frac{image\_side - n + 1}{stride}$ . Таким образом, ввиду большого  $k$  второе слагаемое мало, по сравнению с первым.

Также можем наблюдать, что нет явной зависимости точности и потерь от размера фильтра. Это означает, что для каждой модели следует индивидуально подбирать размерность свёртки.

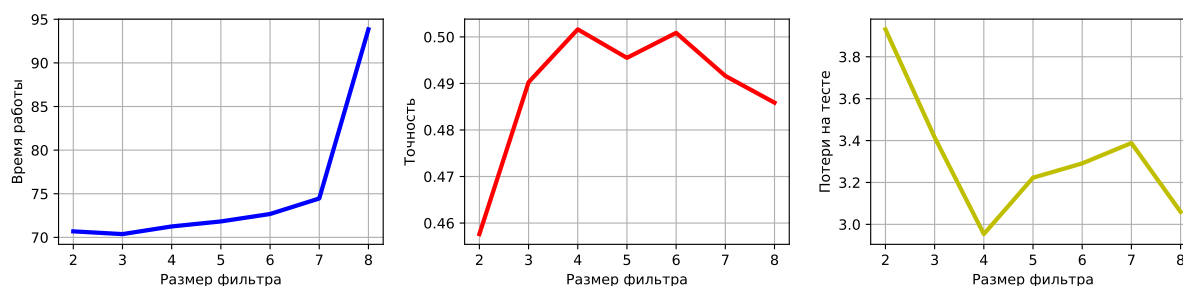


Рис. 2.2.1

### 2.2.2 Исследование влияния количества фильтров

На 2.2.2 уже прослеживается линейная зависимость времени от числа фильтров в блоке, так как количество вычислений увеличивается на число обработанных пикселей одним фильтром.

С увеличением числа фильтров точность и потери в среднем увеличивается и уменьшается, соответственно. Тем не менее слишком большой свёрточный слой может превести к переобучению. Получаем, что размер блока нужно так же подбирать, как и размерность свёртки. К тому же следует регулировать пулинг-слой, который фактически выполняет функцию «bottleneck», пытаясь выделить важные признаки<sup>1</sup>.



Рис. 2.2.2

<sup>1</sup>К вопросу о переобучении сети.

### 2.2.3 Исследование влияния количества блоков в сети

При росте числа блоков время растет экспоненциально. Также могут возникнуть те же проблемы с переобучением, что и в предыдущем пункте. Решение – правильный подбор параметров модели. Результаты приведены на 2.2.3.

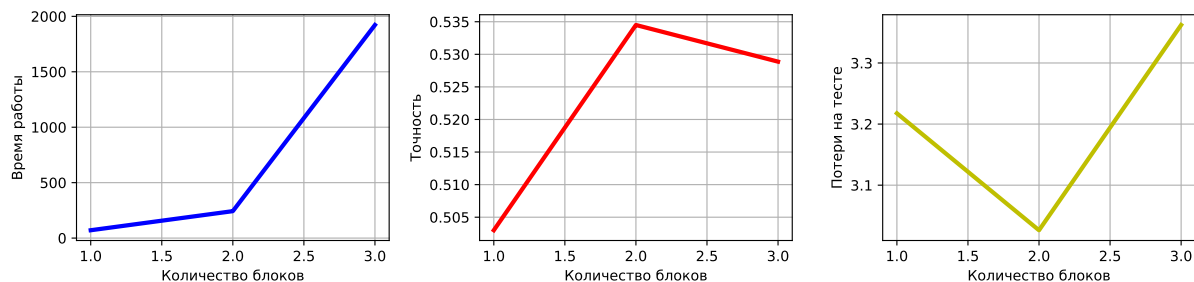


Рис. 2.2.3

### 2.2.4 Исследование стратегий использования momentum

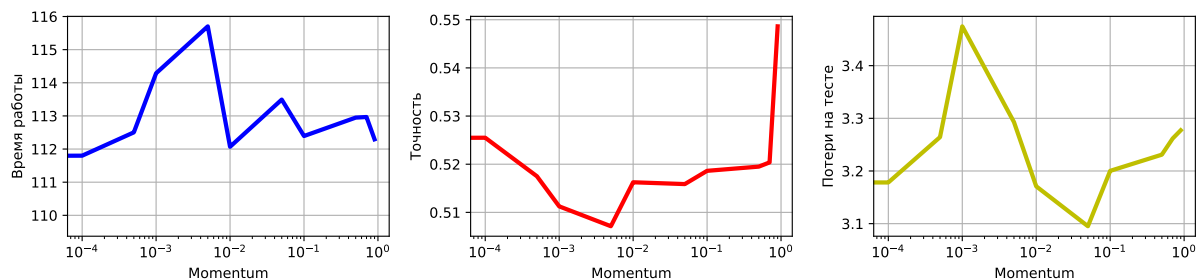


Рис. 2.2.4

По 2.2.4 становится понятно, что параметр momentum требуется отрегулировать для конкретной задачи.

## 2.3 Эксперимент №3

В этом эксперименте предлагалось реализовать свёрточный автокодировщик, обученный по неразмеченной части датасета, а также подобрать его оптимальные параметры.

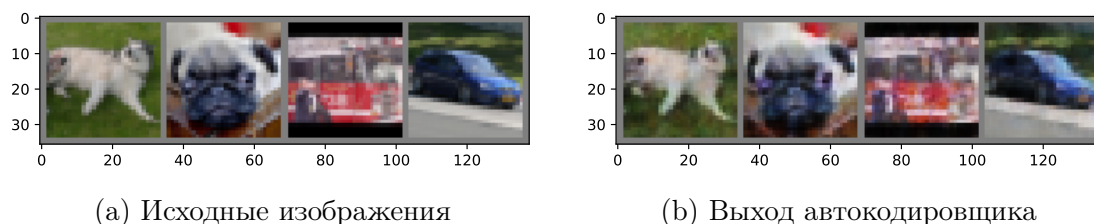


Рис. 2.3.1: Работа автокодировщика

Для сокращения времени перебора параметров автокодировщика был сделан ряд предположений о его устройстве:

- структура фиксируется симметричной
- размерность сжатых данных лежит в границах от 800 до  $3 * 32 * 32 - 1000$
- нелинейностями на выходе автокодировщика могут быть функции  $\tanh$  и Leaky ReLU
- параметры задают корректный автокодировщик

Интуиция введения второго ограничения заключается в том, что автокодировщик с одной стороны не должен слишком сильно сжимать данные, а с другой стараться выделить существенные признаки. Нелинейности подбирались так, чтобы на выходе они выдавали не только положительные значения.

Результат работы наилучшего автокодировщика по метрике MSE представлен на 2.3.1.

## 2.4 Эксперимент №4

Эксперимент №4 состоял в том, что требовалось воспользоваться новым признаковым пространством, порожденным кодирующей частью автокодировщика. А именно проанализировать результаты работы методов из первого исследования на преобразованных признаках.

Таблица 2: Сравнение результатов на разных признаковых пространствах

Метод	Результат	Бейслайны	После преобразования
Мультиномиальная логистическая регрессия	Точность	36 %	55 %
	Log loss	6.63828	2.78158
xgboost	Точность	45.9 %	47.9 %
	Log loss	1.55202	1.46529

Из Табл.2 видно, что качество значительно улучшилось для первого метода по сравнению с подходом, основанным на деревьях. Это, видимо, связано с тем, что выборка стала лучше линейно разделяться. Также повышению качества способствовало понижение размерности пространства, и, как следствие, устранение неинформативных признаков.

Итак, размеченные данные действительно помогли улучшить качество на тесте.

## 2.5 Эксперимент №5

В этом пункте было предложено попробовать модифицировать процесс обучения нейросети, а именно использовать слои кодировщика в качестве первого блока сети.

Параметры блоков, не относящихся к энкодеру, были обучены на тренировочной части датасета, в то время как первый блок дообучался. В итоге, после подбора

необходимых параметров, получилось достичь качества на тестовой выборке в 58.35 %, в то время как точность сетей из второго эксперимента не превышала 56 %.

Таким образом, автокодировщики можно использовать как нелинейную альтернативу PCA и пробовать обучать их для последующего повышения качества методов.