

Метод прореживания нейронных сетей в процессе обучения

Д.А. Новиков, Д.Ю. Буряк

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ имени М.В.Ломоносова,
Факультет вычислительной математики и кибернетики

Проблема



Глубокие нейронные сети имеют десятки скрытых слоев и сотни нейронов в каждом из них. Такие сети имеют миллионы, а иногда даже миллиарды параметров.



Сложности в использовании глубоких нейронных сетей без высокопараллельных вычислительных устройств.



Нейронные сети адаптируются к характерным чертам обучающей выборки, теряя способность к обобщению.

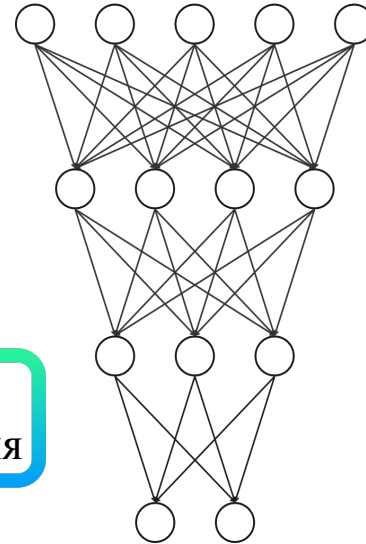
Прореживание нейронных сетей

Прореживание

Итеративное
прореживание

Прореживание в
процессе обучения

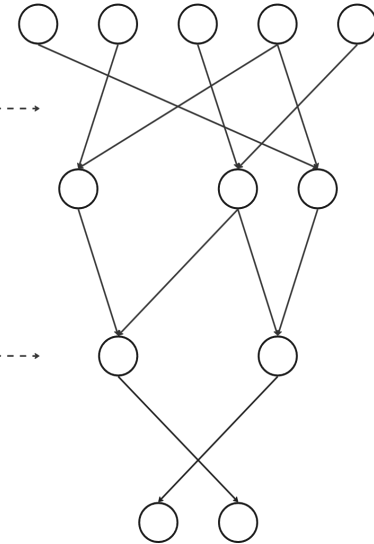
До прореживания



Исключение
связей

Исключение
нейронов

После прореживания



Основная цель данной работы - анализ существующих алгоритмов прореживания глубоких сверточных нейронных сетей и исследование подходов к повышению их эффективности.

Итеративное прореживание



Достоинства

- Доступно множество предварительно обученных моделей нейронных сетей
- Просты в реализации и могут превосходить более сложные методы



Недостатки

- Необходимость обучения нейронной сети до высокой точности перед проведением процедуры прореживания
- Необходимость хранения плотной модели нейронной сети

Прореживание в процессе обучения



Достоинства

- Хранение прореженной модели нейронной сети
- Отсутствие расходов на дообучение
- Сокращение времени вычислений



Недостатки

- Сложность в реализации
- Зависимость от архитектуры нейронной сети
- Сложность в подборе гиперпараметров для обучения

Обзор существующих подходов



Итеративное прореживание

- Прореживание на основе величины
- Прореживание на основе L_1 нормы
- Прореживание на основе расстояния Евклида



Прореживание в процессе обучения

- Прореженное эволюционное обучение
- Эволюционное обучение на основе градиента

Задача исследования

Рассмотрим задачу классификации изображений с помощью глубоких сверточных нейронных сетей. Дана выборка объектов, разделенных на разные классы, и архитектура нейронной сети, решающей данную задачу.

Необходимо предложить и реализовать новый метод прореживания и провести его сравнительное тестирование с существующими алгоритмами.



1 Провести анализ существующих алгоритмов, выбрать методы для исследования и реализации.



2 Предложить и реализовать алгоритм прореживания в процессе обучения.



3 Выбрать базу изображения и модели нейронных сетей.



4 Провести сравнительное тестирование реализованных методов.

Прореживание на основе L_1 нормы

- Исключение фильтров из сверточных слоев с L_1 нормой меньше определенного порогового значения.
- Норма фильтра: $\|x\| = \sum_i^n |x_i|$

Прореживание на основе расстояния Евклида

- Вычисление попарного расстояния Евклида для всех фильтров в сверточном слое.
- Исключение по одному из каждой пары фильтров, расстояние между которыми меньше определенного порогового значения.
- Расстояние Евклида: $d(p, q) = \sqrt{\sum_i^n (p_i - q_i)^2}$

Случайный выбор функций подмножества

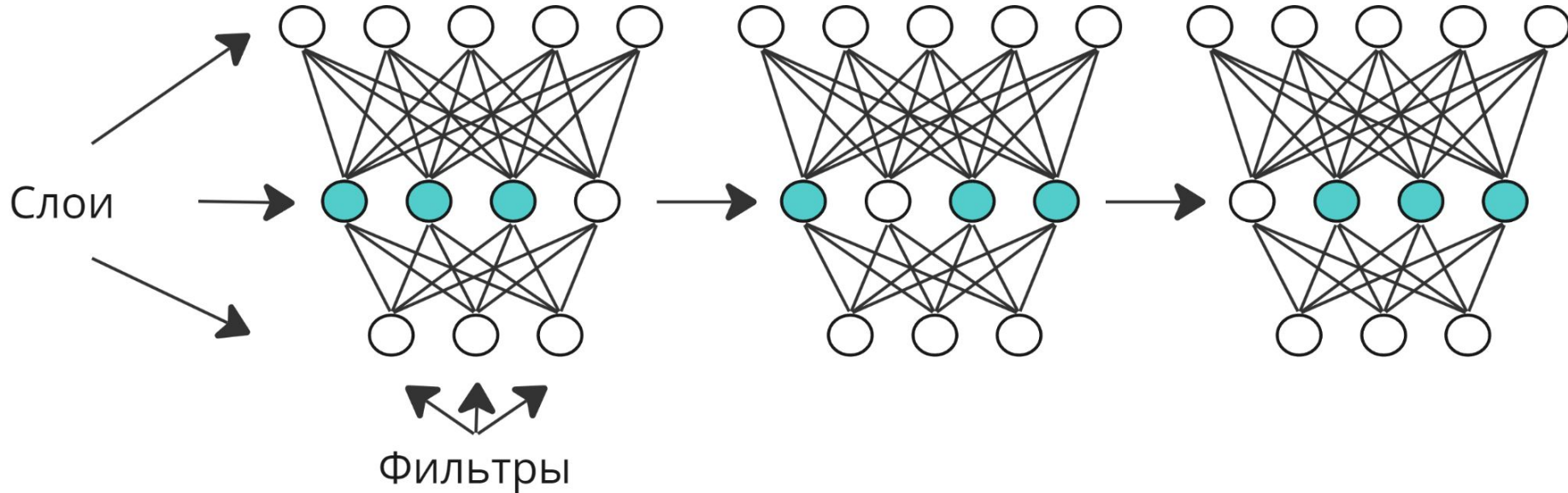
Случайный выбор функций подмножества - это алгоритм, который оценивает каждый признак с точки зрения его полезности в контексте многих других комбинаций признаков и затем формирует окончательное множество полезных признаков.



- Выбирается произвольное подмножество признаков
- Выполняется классификация, для данного подмножества
- Обновляется релевантность
 - Используемых признаков
 - Фиктивного множества признаков
- Процесс повторяется

Фиктивные признаки	Шаг 0	Шаг 1	Шаг k
Признак 1	$Q_0=0$	$Q_1=Q_0 + C_1+E(C)$	$Q_k=Q_{k-1} + C_k+E(C)$
.....			
Признак m	$Q_0=0$	$Q_1=Q_0 + C_1+E(C)$	$Q_k=Q_{k-1} + C_k+E(C)$

Прореживание в процессе обучения



- Каждому фильтру слоя ставится в соответствие значение релевантности r_j
- Вводится множество фиктивных параметров, с соответствующими значениями релевантности q_j
- Вычисляется значение функции потерь c_i
- Обновляются значения релевантности $new_r_j = r_j + c_i - E(c)$; $new_q_j = q_j + c_i - E(c)$

Релевантность фиктивных параметров обеспечивает базовый уровень r_{rand} . Чтобы найти набор фильтров, оказывающих значительное влияние на минимизацию функции потерь нейронной сети, проводится статистическая проверка: требуется, чтобы $p(r_j > r_{rand}) > \delta$ - фиксированный порог вероятности.

Прореживание в процессе обучения

```
1: for iter от 1 до количествоИтерацийОбучения do
2:   for k от 1 до количествоСлоевМодели do
3:     Исключаем часть множество фильтров слоя[k]
4:   end for
5:   Проводим итерацию обучения модели
6:   for k от 1 до количествоСлоевМодели do
7:     for j от 1 до количествоФиктивныхПараметровСлоя[k] do
8:       Обновляем значение релеватности  $q[j]$  фиктивного параметра[j]
9:     end for
10:    for j от 1 до количествоФильтровСлоя[k] do
11:      Обновляем значение релеватности  $r[j]$  фильтра[j]
12:      if ( $iter \bmod$  количествоИтерацийСбораСтатистики = 0) и
13: ( $p(r[j] > r\_base) \leq \delta$ ) then
14:        Удаляем фильтр[j]
15:      end if
16:      Восстанавливаем исключенные фильтры, если они не удалены
17:    end for
18:  end for
```

- Основан на идее метода «Случайный выбор функции подмножества».
- Задача классификации разбивается на набор классификаторов.
- Используются случайные подмножества признаков.
- Появляется возможность оценить качество каждого отдельного признака.

Описание тестовой базы изображений

- CIFAR-10
- 60000 RGB изображений
- 10 классов

Модели нейронных сетей



ResNet-18 (11.5 миллионов параметров),
обучена до точности (accuracy) 90.3%



VGG-16 (134 миллиона параметров),
обучена до точности (accuracy) 91%

Программная реализация



Все программные модули разработаны на языке **Python 3** с использованием фреймворка **PyTorch**.



Модуль удаления фильтров сверточных слоев в процессе обучения нейронной сети.

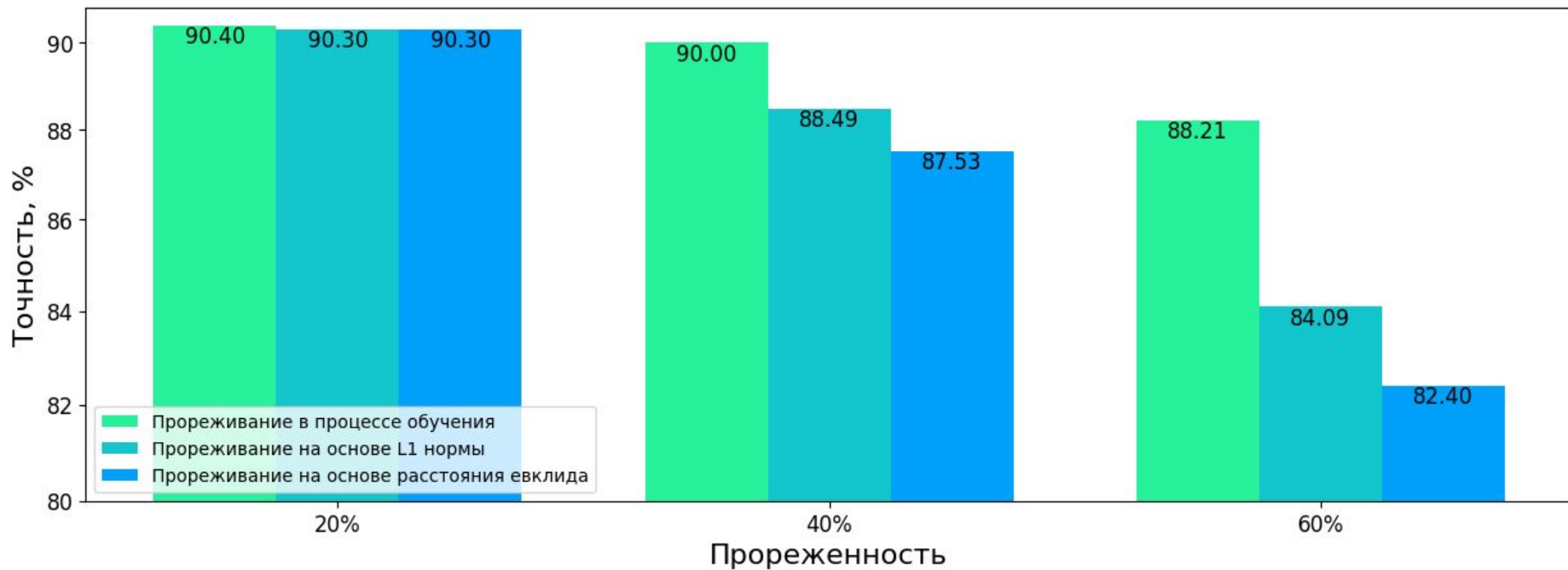


Модули удаления фильтров сверточных слоев на основе L_1 нормы и расстояния Евклида.

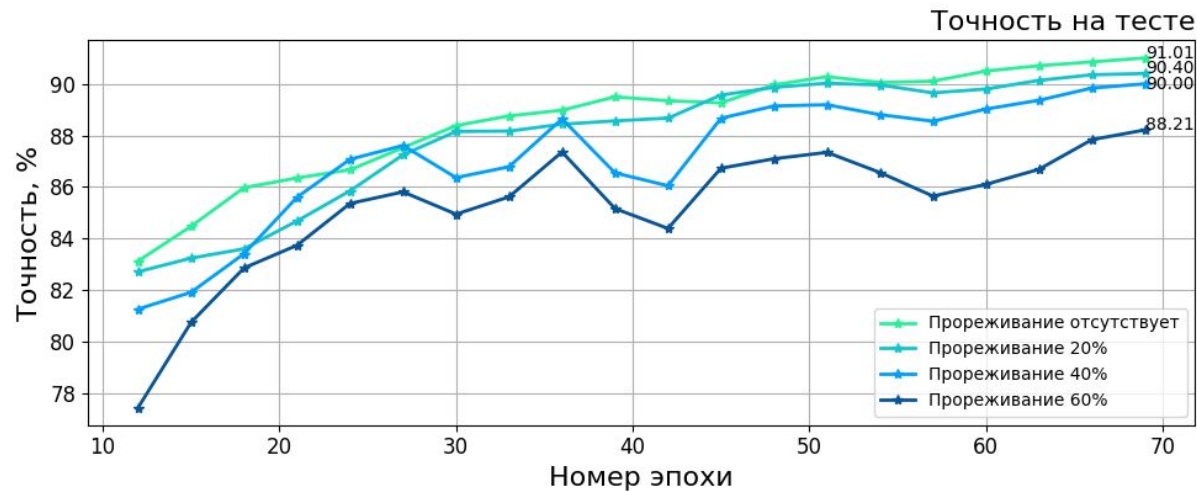
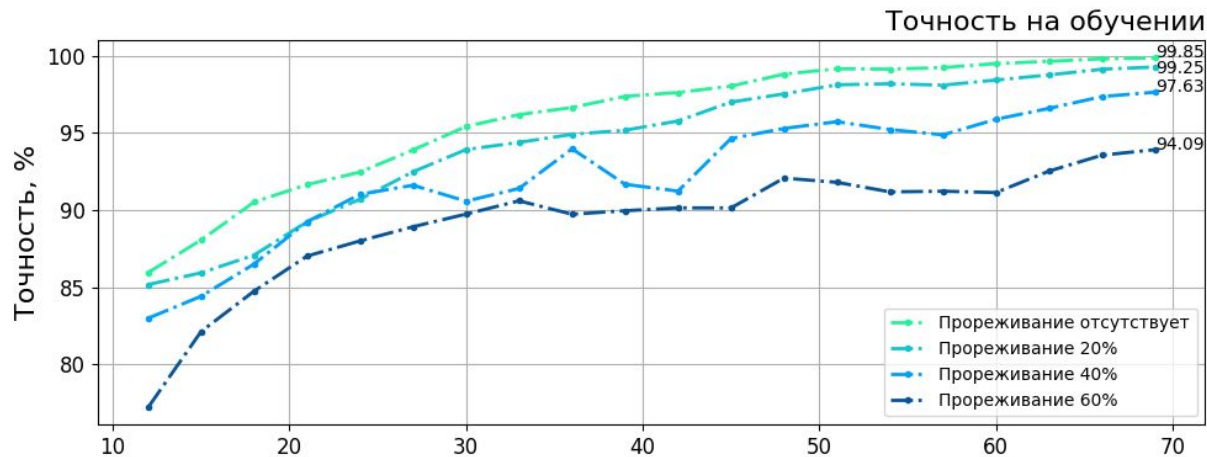


Все эксперименты были проведены в облачном сервисе **Google Colaboratory** с использованием аппаратного ускорителя **GPU**.

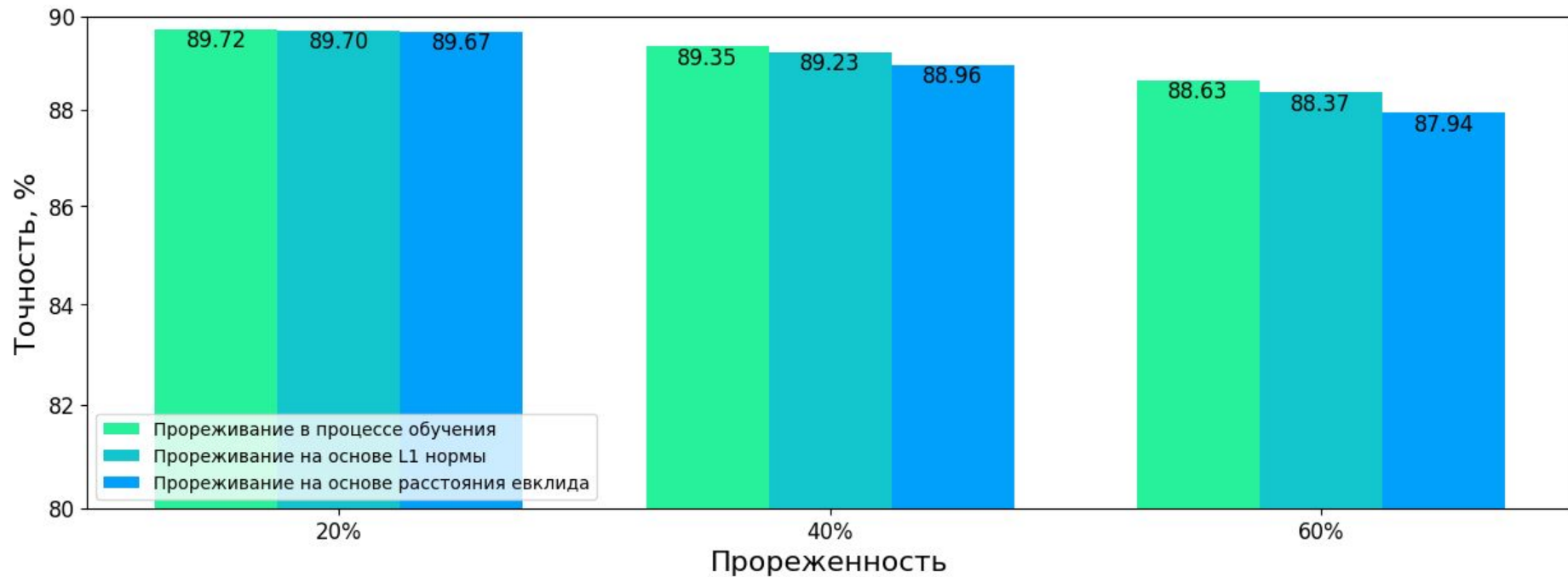
Результаты прореживания VGG-16



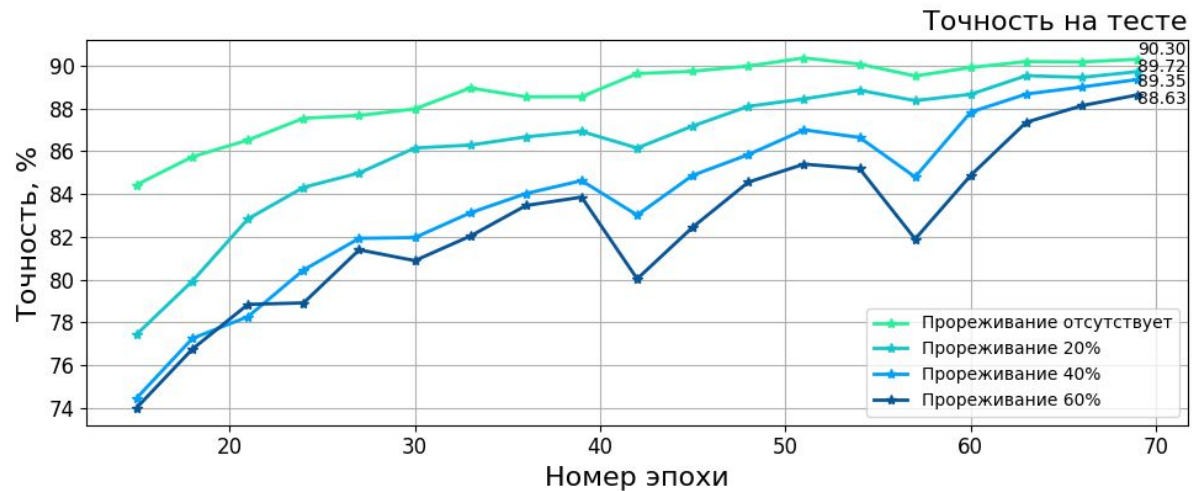
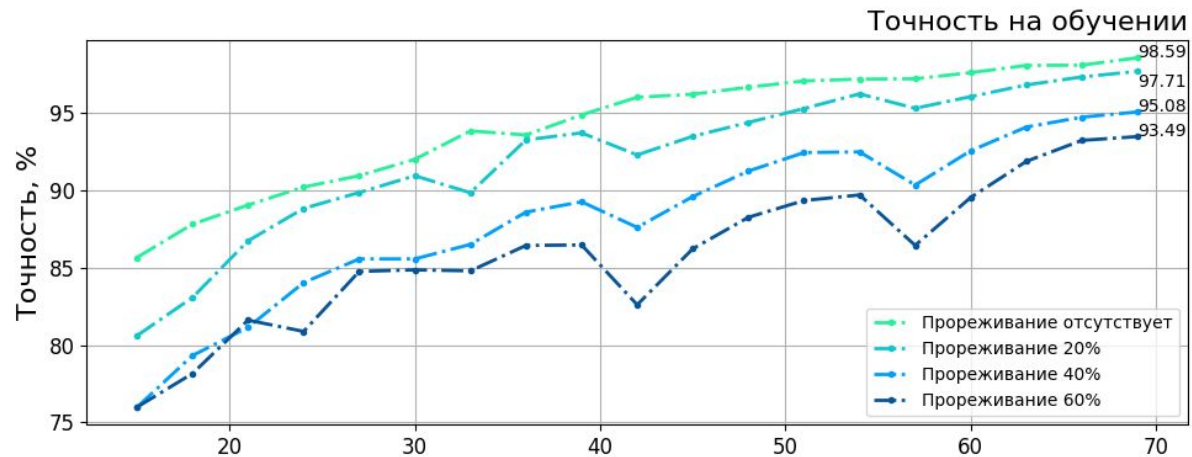
Результаты алгоритма прореживания VGG-16 в процессе обучения



Результаты прореживания ResNet-18



Результаты алгоритма прореживания ResNet-18 в процессе обучения



Основные результаты



Предложен и реализован метод прореживания сверточных нейронных сетей в процессе обучения.



Проведено экспериментальное исследование на задаче классификации изображений метода прореживания в процессе обучения по сравнению с методами прореживания на основе L_1 нормы и расстояния Евклида.



Из анализа полученных экспериментальных результатов следует, что при прореживании сверточной нейронной сети ResNet-18 на 60% точность предложенного метода превосходит точность метода прореживания на основе L_1 нормы на 0.26%, точность метода на основе расстояния Евклида на 0.69%. При аналогичном прореживании сверточной нейронной сети VGG-16 превосходит соответствующие методы на 4.12% и 5.81%.