

# Распознавание текста на основе скелетного представления линий с шириной и сверточных сетей

Рябов Никита Андреевич

Московский физико-технический институт

Научный руководитель: Рейер Иван Александрович, к.т.н., ФИЦ ИУ РАН

- Современные методы распознавания текста основаны на нейросетевых моделях, но:
  - Требуют больших объемов разметки
  - Чувствительны к шумам и искажениям
  - Ограничены в интерпретируемости результатов
- Графовые методы могут улучшить качество распознавания за счет структурного представления текста
- Графовые методы могут улучшить качество распознавания за счет структурного представления текста

- Разработать метод распознавания текста, основанный на графовом представлении, который:
- Использует топологические и семантические связи символов и слов
- Повышает устойчивость к искажениям
- Снижает зависимость от обучающих данных

## Исследуются

Различные варианты обработки графового представления символов с помощью нейросетевых методов.

## Требуется

Предложить архитектуру модели, дающую лучшее качество по сравнению с современными методами при том же количестве параметров

- Бинаризация
- Скелетизация методом из статьи Mestetskiy L., Semenov A.  
Binary Image Skeleton - Continuous Approach
- Полученный граф подается на нейронную сеть

# Графовые свертки: спектральная

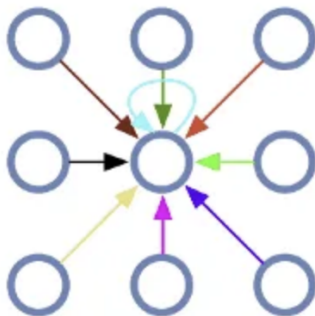
$D$  - диагональная матрица, где  $D_{ii}$  - количество исходящих из  $i$  ребер

$A$  - матрица смежности графа

$f$  - функция активации

Тогда результат свертки:

$$H^{t+1} = f(D^{-\frac{1}{2}}(A + I)D^{-\frac{1}{2}} H^t \Theta)$$



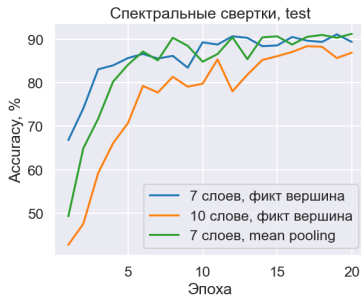
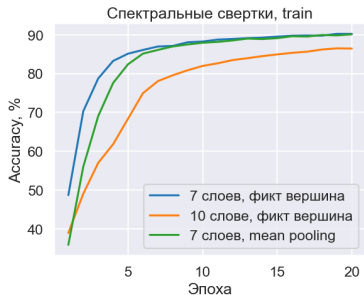
Graph

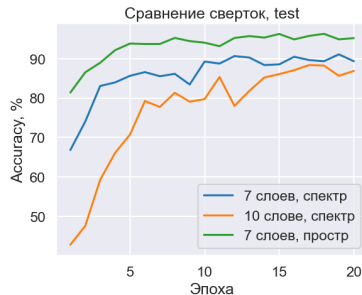
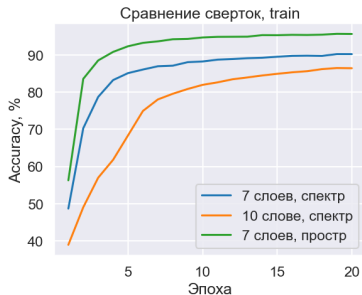
- Фиктивная вершина, скрытое состояние которой - выход сети
- Mean pooling



Архитектура сетей:

- Несколько сверток с функцией активации ReLU
- Пулинг
- Линейный полносвязный слой + SoftMax





# Эксперименты: пространственная свертка

params: 40138

```
Epoch: 001, Train Loss: 1.2371, Train Acc: 0.5627, Test Acc: 0.8140
Epoch: 002, Train Loss: 0.5095, Train Acc: 0.8358, Test Acc: 0.8654
Epoch: 003, Train Loss: 0.3580, Train Acc: 0.8853, Test Acc: 0.8898
Epoch: 004, Train Loss: 0.2898, Train Acc: 0.9083, Test Acc: 0.9217
Epoch: 005, Train Loss: 0.2446, Train Acc: 0.9235, Test Acc: 0.9385
Epoch: 006, Train Loss: 0.2168, Train Acc: 0.9326, Test Acc: 0.9373
Epoch: 007, Train Loss: 0.2059, Train Acc: 0.9367, Test Acc: 0.9374
Epoch: 008, Train Loss: 0.1870, Train Acc: 0.9423, Test Acc: 0.9527
Epoch: 009, Train Loss: 0.1837, Train Acc: 0.9433, Test Acc: 0.9447
Epoch: 010, Train Loss: 0.1730, Train Acc: 0.9471, Test Acc: 0.9407
Epoch: 011, Train Loss: 0.1660, Train Acc: 0.9488, Test Acc: 0.9318
Epoch: 012, Train Loss: 0.1668, Train Acc: 0.9490, Test Acc: 0.9527
Epoch: 013, Train Loss: 0.1653, Train Acc: 0.9493, Test Acc: 0.9574
Epoch: 014, Train Loss: 0.1537, Train Acc: 0.9535, Test Acc: 0.9536
Epoch: 015, Train Loss: 0.1483, Train Acc: 0.9532, Test Acc: 0.9628
Epoch: 016, Train Loss: 0.1462, Train Acc: 0.9542, Test Acc: 0.9490
Epoch: 017, Train Loss: 0.1454, Train Acc: 0.9538, Test Acc: 0.9578
Epoch: 018, Train Loss: 0.1452, Train Acc: 0.9547, Test Acc: 0.9632
Epoch: 019, Train Loss: 0.1389, Train Acc: 0.9570, Test Acc: 0.9492
Epoch: 020, Train Loss: 0.1409, Train Acc: 0.9566, Test Acc: 0.9523
Best Test Accuracy: 0.9632
```

- Полученная точность не достигает точности LeNet5, но выдает хорошее качество, используя меньше обучаемых параметров
- Пространственная свертка дает лучший результат, но требует больше параметров на слой
- При этом на достаточно маленьких моделях тоже получается получить приемлемую точность