Распознавание текста на основе скелетного представления толстых линий и сверточных сетей

И. А. Рейер, В. В. Стрижов, М. С. Потанин, Д. Ожерелков, А. Булатов, В. А. Шокоров

reyer@forecsys.ru; strijov@gmail.com; mark.potanin@phystech.edu; ozherelkov97@gmail.com; ayd98@mail.ru; v.shokorov@yandex.ru

Исследуется метод распознавания символа, по его скелетному представлению, с применением сверточных нейронных сетей на графе скелета. В качестве входных данных используется скелетное представление символа, например, его можно получить используя Алгоритм Л.М.Местецкого. Предложеный алгоритм позволяет работать с графами небольшого размера, количество вершин которого не больше 30. Наша модель показала результат качества сравнимый с результатом человека. Также предложенный алгоритм имеет достаточно легкую с вычислительной точки зрения структуру и позволяет обрабатывать один граф за $0,3\mu s$. Проведены эксперименты по классификации рукописных цифр.

Ключевые слова: *распознавание текста, скелетное предствавление, GCNN, привилегированное обучение*

1 Введение

Мы решаем задачу распознавания текста, для этого рассматриваем изображения символа, как двумерную фигуру, ограниченной конечным множеством полигонов. Алгоритм Л.М.Местецкого, основанный на прямом построении обобщенной триангуляции Делоне Множества граничных элементов фигуры позволяет нам получить скелетное графовое представление данного символа, с достаточным качеством точности положения вершин. Цель исследования изучить возможность применения сверточной нейронной сети на скелетном представлении символа. Мы рассматриваем изображения символа, как двумерную фигуру, ограниченной конечным множеством полигонов, полученной из бинаризации изображения (В нашем случае под бинаризацией понимается следующее: все пиксели, цвет которых НЕ черный (в нотации цветов от 0 до 255), становятся белыми). Алгоритм Л.М.Местецкого [1] позволяет нам получить скелетное графовое представление данного символа, поэтому исследовательская часть нашей работы заключается в том, чтобы научиться качественно преобразовывать графовое представление символа в вектор, для дальнейшего решения задачи классификации. Сложность данного подхода заключается в нерегулярности структуры графа, объекты, подлежащие обработке, не упорядочены и могут иметь произвольную размерность и структуру связей. Обстоятельное исследование этой проблемы и возможных подходов к ее решению можно найти в статье [2]. Кроме канонического решения задачи обработки изображения CNN, похожая проблема решается в [3] там используется набор контуров и граф скелетного представления с последующим линейным классификатором SVM. Также существует подход использующий медиальное представление [4]. Существует ряд статей Липкиной А.Л. и Л.М.Местецкий Посвященных решению нашей задачи, например [5], [6].

Для решения задачи необходимо найти оптимальные параметры графового представления (такие как координаты вершин, их валентность, степень, тольщина линий и тд.) описывающие символ, для их подбора можно воспользоваться например [7].

2 Свертка на графе

Многие подходы созданные для свертки на графе предназначены либо для предсказания существования ребра между вершинами, известное, как Link Prediction, либо для классификации вершин (Node Classification). На самом деле это схожие задачи, потому что они решаются через представление вершину в виде вектора, для дальнешего сравнения расстояния между ними. Эти методы описания основываются на структуре связей и требуют, чтобы каждая вершина либо имела только одну метку, либо не имела вовсе, отсюда следует, что локальное описание графа должно быть разнообразно, что не выполняется в нашем случае, т.к. скелетонизация выдает граф, вершины которого часто имеют степень равную двум, и количество вершин в графе редко превышает 25, и вдобавок каждая вершина имеет 4 метки. Таким образом получаем, что известные методы плохо применимы для решения данной задачи.

Интересный подход был описан в статье [8], там предлагался алгоритм сжатия графа состоящий из трех частей: сначала применяется node2vec [9], затем на полученном векторном представлении вешин используется PCA (principal component analysis), что позволяет получить значимые "направления"верши графа, иными словами показать общий вид вершин графа, после чего, можно кслассифицировать полученное представление.

В статье про GraphSAGE [10], был предложен метод, который решает задачу представления вершины в виде вектора, там используеются модификация рекурентной нейронной сети, следующее значение вектора вершины представляется как нелинейная функция от старого значения и агрегированных значений соседей. Такой подход позволяет использовать в качестве метки вектор неединичной длины, но, как заявляют авторы, метод работает не графах с более чем 100000 вершинах.

Рис. 1 Пример схемы свертки в статье про GraphSAGE.

3 Постановка задачи

Пусть дано множество $\mathcal{A}$, состоящее из пар $(\mathbf{I}, y)$, где $\mathbf{I} \in \mathbb{I}$ - растровая черно-белая картинка размера $m \times n$ пикселей, в нотации цветов от 0 до 255, с изображением рукописной цифры, а $y \in \mathbb{Y}$ - метка класса - значение цифры. Примером набора таких даных, например, является-MNIST [11].

Задача заключается в построении модели, которая определяет по изображению цифры ее значание. Предлагаеммый подход заключается в использовании скелетонизации, для получения графового описания изображения, графовой сверточной сети, для представления графа в виде вектора фиксированной длины, полносвязной нейронной сети, для классификации полученного представления графа.

Для описания решения задачи зададим пространство неориентированных графов G, полученных с помощью скелетного представления $f_1 \colon \mathbf{I} \to \mathbf{G} \in G$, причем каждой вершине графа соответствует $\mathbf{l} \in \mathbf{L} = \mathbb{R}^4_+$, 4 числа, называемыми координатами вешины,

радиусом и количеством соседей. Координатами вершины является два числа, описывающие положение вершины на изобрадении, радиусом называется максимальный размер окружности, которую можно вписать в фигуру, представленной на бинаризованном изображении цифры.

При данной формулировке, задача разбивается на несколько подзадач, первая, нахождение оптимального представление изображения в виде графа, вторая, нахождение оптимальной модели для предсказания класса графа, другими словами какому классу принадлежит обрабатываемое изображение.

3.1 Описание подзадачи о представлении изображения графом

Для данной части используется алгоритм Л.М. Местецкого.

3.2 Описание подзадачи о классификации графа

Для этой части рассматривается множество сверточных графовых нейронных сетей $\mathfrak{C} = \{\mathbf{C}_1, \mathbf{C}_2, \dots\}$ и положительное целое число δ (т. е. ожидаемое количество классов $= |\mathbb{Y}|$), мы хотим получить $\overrightarrow{\mathbf{d}} - \delta$-мерное представление для каждого графа $\mathbf{G}_i \in \mathfrak{C}$, где $\forall i, \overrightarrow{\mathbf{d}_i}$ соответствует вероятности принадлежности этого графа к i-му классу из $\mathbb{Y}$.

Рассматриваются графы, $\mathbf{G} = (\mathbf{N}, \mathbf{E}, \lambda)$, где $\mathbf{N}$-множество вершин, а $\mathbf{E} \subseteq (\mathbf{N} \times \mathbf{N})$-множество ребер. Также для графа $\mathbf{G}$, должна существовать функция λ такая, что $\lambda\colon \mathbf{N} \to \mathbf{l}$, которая присваивает уникальную метку из множества $\mathbf{L}$ каждому узлу $\mathbf{n} \in \mathbf{N}$.

При данных условиях, постановка задачи выражается через минимизацию перекрестную энтропию между прогнозируемым и фактическим значение, т.к. Cross-Entropy Loss используется для создания уверенной модели, т.е. модели не только точно предсказывающей значение метки класса, но и делающей это с большей вероятностью.

$$\arg\min_{\mathbf{C} \in \mathfrak{C}} - \sum \sum_{i}^{\delta} \mathbf{d}_i \log \mathbf{y}_i \tag{1}$$

Где первая сумма берется по всем парам $(\mathbf{I}, y)$ из множества $\mathcal{A}, \mathbf{I} \in \mathbb{I}, y \in \mathbb{Y}, \mathbf{d} = \mathbf{C}(\mathbf{I})$ - δ-мерное представление для графа, $\mathbf{y}$ - вектор, y-ая координата которого $= 1$, остальные $= = 0$.

Таким образом задача заключается в минимизации (1) для графовой сверточной сети, общий вид которой показан на рис. 2, схема каждой состовляющей показана на рис. **??**, **??** соответственно.

4 Описание базового алгоритма

В качестве базового алгоритма взят подход из статьи [12], идея которого заключается в том, что вершинами искомого графа являются пиксели, поэтому каждая вершина получается с небольшим количеством соседей и данный граф имеющими схожую структуру, что и изображение, и в качестве свертки на вершинах используется линейное преобразование от векторных описаний соседей. Отличие данного алгоритма от сверточной нейронной сети заключается в том, что сеть оперирует понятиями присущими графу, другими словами на вход подается матрица смежности и метка на каждой вершине. Такой подход позволяет достичь точности на базе данных MNIST [11] 92%

Базовый алгоритм также решает на задачу минимизации (1), также находит минимум на множестве $\mathcal{A}$, состоящее из пар $(\mathbf{I}, y)$, при этом операция свертка происходит на

вершинах, которые являются пикселями самого изображении. Получаем, что задача переписывается в виде:

$$\arg\min_{\mathbf{w}} -\sum\sum_{i}^{\delta} \mathbf{d}_i \log \mathbf{y}_i \tag{2}$$

Где минимизация идет по всем весам сверточной сети, первая сумма берется по всем парам $(\mathbf{I}, y)$ из множества $\mathcal{A}$.

5 Описание алгоритма

Нами был предложен алгоритм решения задачи классификации графа скелетного представления, состоящий из 3 частей. Первая, свертка на графе, сделанная на подобии GraphSAGE, данную часть алгоритма можно увидеть на схеме Алгоритм 1, вторая выделение главных компонент, по аналогии с graph embend like cnn on photo, затем, полносвязныйе слои для классификации полученных компонент. Общую схему алгоритма с применением PCA можно увидеть на рис. 2. Так как PCA достаточно дорогостоящая по времени опперация, поэтому, после того, как сеть достаточно обучилась, мы находим все главные компоненты, для каждого класса графа, усредняем их и заменяем PCA на разложение вектора, описывающего какую-то вершину, на координаты в базисе усредненных главных компонент. Модифицорованную общую схему алгоритма можно увидеть на рис 3.

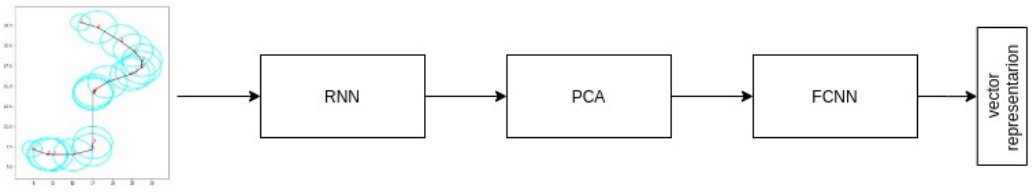


Рис. 2 Общая схема нейронной сети с применением PCA.

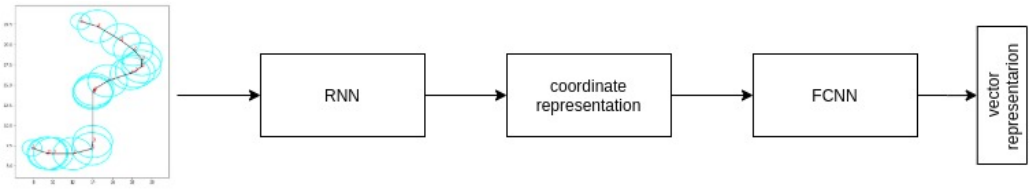


Рис. 3 Общая схема нейронной сети без использования применением PCA, в данном случае PCA заменяется на координатное представление каждого вектора описывающего вершину.

Алгоритм 1 Псевдокод для сверточной сети на графе

input : Граф $\mathbf{G} = (\mathbf{N}, \mathbf{E}, \lambda)$, напомним, что функция $\lambda\colon \mathbf{N} \to \mathbf{l}$ задает метки на вершинах; глубина расспространения K; матрица весов $\mathbf{W}^k, \forall k \in \{1 \dots K\}$; нелинейная функция σ; дифференцируемая агрегирующая функция $\mathrm{AGGREGATE}_k, \forall k \in \{1 \dots K\}$; функция описывающая соседей $\mathcal{N}\colon n \to 2^n, \mathcal{N}(n) = \{m \in \mathbf{E}\colon (m, n) \in \mathbf{E}\}$

output : векторное представление $\mathbf{z}_n \ \forall n \in \mathbf{N}$

$h_n^0 \leftarrow \lambda(n), \ \forall n \in \mathbf{N}$;

for $k = 1 \dots K$ **do**

  for $n \in \mathbf{N}$ **do**

    $h^k_{\mathcal{N}(n)} \leftarrow \mathrm{AGGREGATE}_k(\{h_m^{k-1}, \ \forall m \in \mathcal{N}(n)\})$;

    $h_n^k \leftarrow \sigma\big(\mathbf{W}^k \cdot \mathrm{CONCAT}(h_n^{k-1}, h^{k-1}_{\mathcal{N}(n)})\big)$;

  end

  $h_n^k \leftarrow h_n^k / \|h_n^k\|_2, \ \forall n \in \mathbf{N}$

end

$\mathbf{z}_n \leftarrow h_n^K, \ \forall n \in \mathbf{N}$

6 Вычислительный эксперимент

6.1 Данные

В качестве первичного набора данных используется библиотека MNIST [11]. Перед скелетизацией изображения были бинаризованы(все символы, не являющиеся совершенно черными, переводятся в белые), поскольку этого требует алгоритм скелетизации. Полученный скелет, являющийся ненаправленным графом, каждая вершина описывается координатами, степенями вершин и максимальными радиусами кругов, вписанных в цифры.

6.2 Приминение предложенного алгоритма

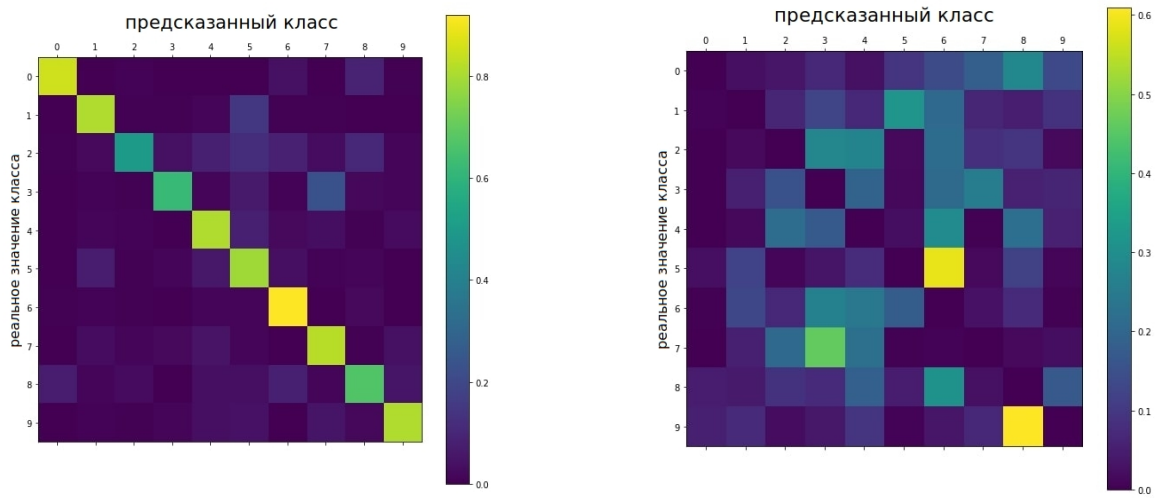
После скелетонизации, на полученных графах обучается предложенная сеть, первая проблема, с которой мы сталкиваемся при обучении это то, что необходимо правильно инициализировать веса свертки, так как если у нас веса инициализированны случайно, то получится, что алгоритм который используется для PCA не будет сходиться, потому что рассматриваемые графы имеют достаточно однородную структуру, поэтому вектора вершин все будут сонаправленны. Это решается несколькими запусками обучения. Далее, когда кривая ошибки сходится, (на данном этапе у нас точность предсказания равна 55%), после этого мы заменяем PCA на координатное представление и дообучаем сеть.

В ходе эксперимента подбираются оптимальные параметры сети такие как, число нейронов в скрытом слое, глубина работы алгоритма свертке на графе, функции активации и т.д. После чего мы получаем итоговую точность предсказания порядка 77%.

7 Анализ ошибки

На рисунке 4(а) показанна гистограмма определения класса нашего алгоритма, по оси y отображаются истинные значения класса, по оси предсказанный алгоритмом класс. На рисунке 4(б) показанна подобная гистограмма, кроме главной диагонали, отсюда видно, что алгоритм часто путает, например, 8 и 9.

Достаточно показательным фактом оказалаось то, что ответ сети оказался инвариантным относительно сдвига координат вершин, что соответствует логике. Так же если просмотреть особые случаи, то на мнигих входных данных даже человеку сложно определить истинное значение цифры, примеры таких вхдных данных можно рассмотреть



а) б)

Рис. 4 гистограмма определения класса нашего алгоритма, по оси y отображаются истинные значения класса, по оси предсказанный алгоритмом класс. На рисунке (б) показанна подобная гистограмма, кроме главной диагонали, отсюда видно, что алгоритм часто путает, например, 8 и 9.

на рис. 6. Приведя небольшое исследование мы определили, что человек по скелетному представлению может верно определить класс с точностью 84%, полученную гистаграмму можно увидеть на рис. 5.

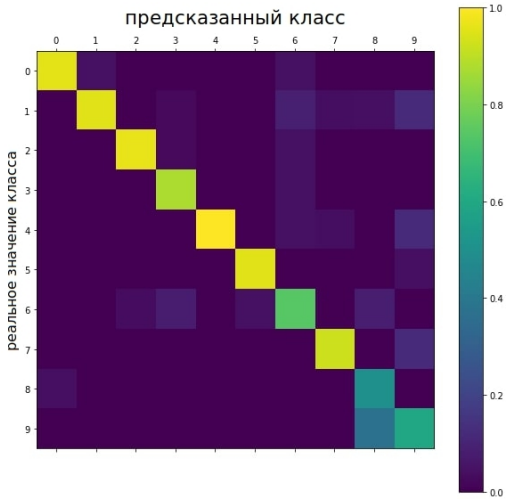


Рис. 5 гистограмма определения класса человеком, по оси y отображаются истинные значения класса, по оси предсказанный алгоритмом класс.

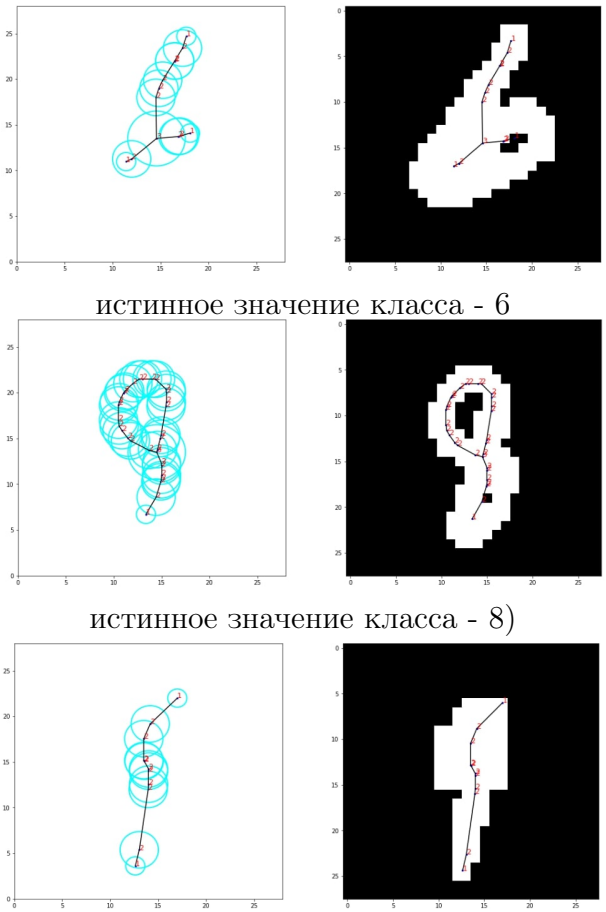


Рис. 6 Примеры скелетных представлений, которые по которым невозможно верно определить истинный класс.

Литература

[1] Л.М. Местецкий. Скелетизация многосвязной многоугольной фигуры на основе дерева смежности ее границы. *Сиб. журн. вычисл. математики.*

[2] Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *CoRR*, abs/1611.08097, 2016.

[3] Wei Shen, Xinggang Wang, Cong Yao, and Xiang Bai. Shape recognition by combining contour and skeleton into a mid-level representation. pages 391–400, 2014.

[4] Arseev Sergey Lomov Nikita. Neural networks for shape recognition by medial representation. *3nd International ISPRS Workshop on PSBB Moscow.*

[5] Л.М. Местецкий А.Л. Липкина. Классификация букв в изображениях на основе медиального представления. *Труды международной конференции по компьютерной графики и зрению "Графикон*

.

[6] Л.М. Местецкий А.Л. Липкина. Структурный подход к классификации букв в изображениях. *Труды международной конференции по компьютерной графики и зрению "Графикон*

.

[7] С.Ю. Желтов Ю.В. Визильтер, В.С. Горбацевич. Структурно-функциональный анализ и синтез глубоких конволюционных нейронных сетей. *Компьютерная оптика*, 43(5):886–900, 2019.

[8] Antoine Jean-Pierre Tixier, Giannis Nikolentzos, Polykarpos Meladianos, and Michalis Vazirgiannis. Classifying graphs as images with convolutional neural networks. *CoRR*, abs/1708.02218, 2017.

[9] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. *CoRR*, abs/1607.00653, 2016.

[10] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. *CoRR*, abs/1706.02216, 2017.

[11] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

[12] Boris Knyazev. *Tutorial on Graph Neural Networks for Computer Vision and Beyond (Part 1)*, 2019 (accessed August 4, 2019).

Поступила в редакцию