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. Вспользуется скелетное представление символа получаемый с помощью алгорита Л.М.Местецкого [1]. Предлагается подход, который позволяет работать с графами небольшого размера, классификации графов скелетного представления. Предлогаеммый алгоритм описывает вектором каждую вершину, через агрегацию информации о соседних вершинах. Выделяет базисные представления графа для каждого класса. Классифицирует полученное описание. Данный алгоритм имеет несложную с вычислительной точки зрения структуру. Проведены эксперименты по классификации рукописных цифр.

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

1 Введение

Изучается метод распознавания текста. Каждая буква представляется через скелетное представление и решается задача классификации скелетного представления. Изображение символа рассматривается, как двумерная фигура, ограниченная конечным множеством полигонов. Полигоны получаются из бинаризации изображения. Под бинаризацией понимается следующее: все пиксели, цвет которых не черный в нотации цветов от 0 до 255, становятся белыми. Далее полученное изображение обрабатывается алгоритмом Л.М.Местецкого [1], основанного на прямом построении обобщенной триангуляции Делоне. Множество граничных элементов фигуры пораждает скелетное графовое представление символа, с достаточным качеством точности положения вершин. Под точностью положения вершин понимается, что вершина находится внутри фигуры и максимально возможно отдалена от границы. Иными словами, если представить символ как толстую линию, то вершина обязана лежать в центре этой толстой линии. Этот алгоритм возвращает матрицу смежности графа, координаты вершин и радиус окружности (толщина линии в данной точке).

Цель исследования предложить сверточную нейронную сеть на графе скелетного представления символа. Задача —требуется преобразовывать графовое представление символа в вектор, для дальнейшего решения задачи классификации.

2 Введение в задачу классификации скелетного представления

Исследуемая проблема состоит в том, что требуется предложить регулярное описание графа с нерегулярной структурой, т.е. граф, подлежащий обработке, имеет произвольное число вершин, не имеет строгую структуру связей, а его вершины не упорядочены. Обзор этой проблемы и подходы к ее решению можно описанны в [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. [4,5] Также существует подход использующий медиальное

представление [6]. Существует ряд статей Липкиной А.Л. и Л.М.Местецкий Посвященных решению нашей задачи, например [7], [8].

Для решения задачи необходимо подобрать признаковое описание графового представления (координаты вершин и их валентность, степень, тольщина линий и их комбинации) описывающие символ. Для их подбора можно воспользоваться например [9].

Большенство подходов для свертки на графе предназначены либо для предсказания существования ребра между вершинами (Link Prediction), либо для классификации вершин (Node Classification). Эти задачи схожи, потому что они решаются через представление вершины в виде вектора, для дальнешего погружения в метрической пространство. Данные методы применимы для графов, у которых каждая вершина имеет не более одного признака. Также локальное описание графа должно быть разнообразно, что может выполняться только в графах с большим число вершин. Эти критерии не выполняется в нашем случае, т.к. граф, описывающий скелетное представление символа, имеет следующие ограничения: 1) большенство вершин имеет степень не больше двух, 2) число вершин в графе редко превышает 25, 3) каждая вершина имеет 4 признака. Таким образом получаем, что известные методы плохо применимы для решения данной задачи.

В [10] описан подход к решению задачи. В ней разобран алгоритм сжатия графа. Он состоит из трех частей: 1) применяется node2vec [11], 2) на полученном векторном представлении вершин применяется PCA (principal component analysis). Это позволяет получить значимые "направления"верши графа, иными словами показать общий вид вершин графа. 3) полученные направления классифицируются.

В статье GraphSAGE [12], предложен метод решения задачи представления вершины в виде вектора. Он основан на модификации рекурентной нейронной сети. В ней 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.

В данной работе предлагается метод, основанный на приминении подхода GraphSAGE и PCA для агрегации векторных описаний вершин и дальнейшего анализа. Общая схема алгоритма показана на рис. 3.

3 Постановка задачи

Задано 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, метка класса $y \in \mathbb{Y}$—значение цифры. Вычислительный эксперимент использует коллекцию картинонок MNIST [13].

Для описания решения задачи зададим пространство неориентированных графов $\mathbf{G} = (\mathbf{N}, \mathbf{E}, \lambda)$, где $\mathbf{N}$-множество вершин, $\mathbf{E} \subseteq (\mathbf{N} \times \mathbf{N})$-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 $\mathbf{n} \in \mathbf{N}$. Данный граф получается из скелетного представления $f_1\colon \mathbf{I} \to \mathbf{G} \in G$. Каждая вершина графа получает метку $\mathbf{l} \in \mathbf{L} = \mathbb{R}^4_+$. Эти 4 числа, называются 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. Пример такого графа показан на рис. 1.

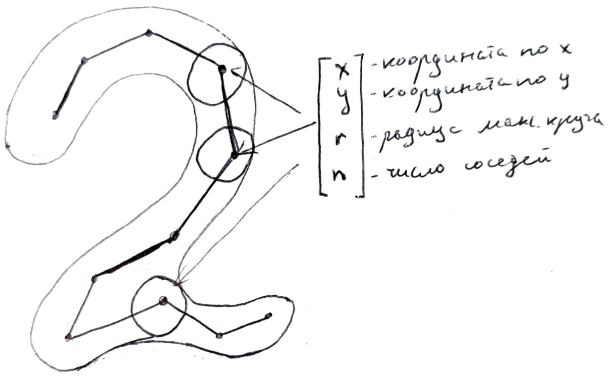


Рис. 1 Пример графа на рукописной цифре 2.

Для описания представления изображения графом используется алгоритм Л.М. Местецкого.

Для классификации графа 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 $= |\mathbb{Y}|$). Нобходимо получить $\overrightarrow{\mathbf{d}}$ – δ-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 $\mathbb{Y}$.

При данных условиях, постановка задачи выражается через минимизацию перекрестной энтропии между прогнозируемым и фактическим значением. 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 $= = 0$.

4 Описание базового алгоритма

В качестве базового алгоритма взят подход из статьи [14], 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 коллекции изображений MNIST [13] 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

Рис. 2 Пример схемы свертки в статье про GraphSAGE.

Предложен алгоритм решения задачи классификации графа скелетного представления, состоящий из трех частей. Первая, свертка на графе, производится с примененим рекурентной нейронной сети, схематичное ее отображение показано на рис 2. Эта идея аналогична GraphSAGE. Схема данной части алгоритма представленна в Алгоритме 1. Вторая выделение главных компонент, используется идея приминения PCA из [10]. Затем, векторные предсатвления вершин агрегируются и данные передаются в полносвязныйе слои для классификации графа. Общую схему алгоритма с применением PCA показана на рис. 3. Так как PCA достаточно вычислительно сложная опперация, по сравнению с остальными операциями в предлагаемом алгоритме, поэтому, после того, как сеть достаточно обучилась. Критерием этого, является стабилизация ошибки. В эксперименте применялся критерий раннего отсанова после 8 эпох. Находятся главные компоненты, для каждого класса графа, и агрегируются полученные вектора, предполагается, что полученные вектора описывают базисный граф каждого класса. Замена PCA происходит с помощью представления входного графа через координаты векторов базисных графов. Такое упрощание уменьшает вычислительную сложность алгоритма в 1.5 раза. Модифицированная общая схему алгоритма показана на рис 4.

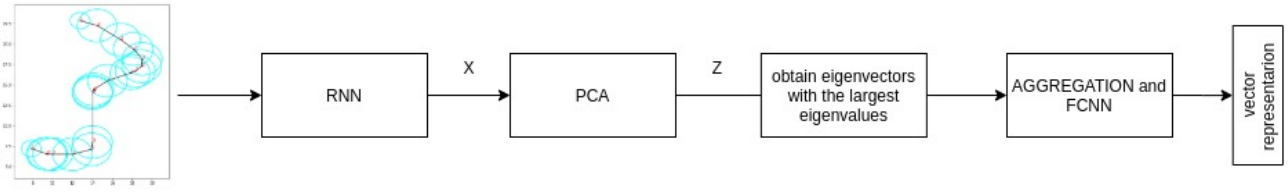


Рис. 3 Общая схема нейронной сети с применением PCA.

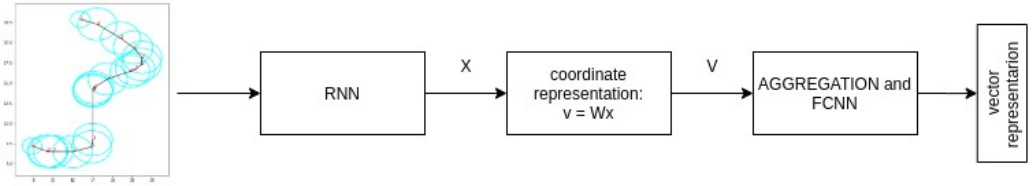


Рис. 4 Общая схема нейронной сети без использования применением PCA, в данном случае PCA заменяется на координатное представление каждого вектора описывающего вершину.

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**

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

  end

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

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

end

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

6 Вычислительный эксперимент

Используется коллекция изображений MNIST [13]. Перед скелетизацией изображения были бинаризуются. Все символы, не являющиеся совершенно черными, переводятся в белые, этого требует алгоритм скелетизации. Приминяется алгоритм Л.М.Местецкого [1]. Получается скелет, который является неориентированным графом. Каждая его вершина описывается координатами, степенями вершин и максимальными радиусами кругов, вписанных в цифры.

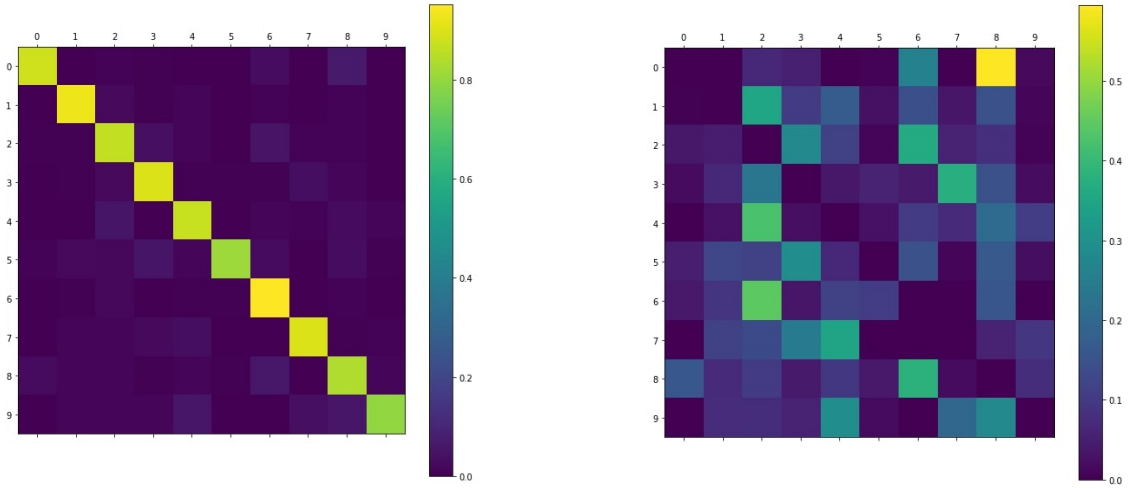
После скелетонизации, на полученных графах обучается предложенная сеть. Иными словами решается задача минимизации (1). Первая проблема, с которой может возникнуть при обучении это то, что необходимо правильно инициализировать веса свертки. В эксперименте все веса инициализированны случайно. Поэтому, получится, что алгоритм который используется для PCA в некоторых сдучаях не будет сходиться. Это происходит из-за того, что рассматриваемые графы имеют достаточно однородную структуру, и получается, что вектора вершин будут сонаправленны. Это решается приминением идеи мультистарта. Когда кривая ошибки стабилизируется, (на данном этапе у нас точность предсказания равна 55%), PCA заменяется на координатное представление и сеть дообучается.

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 87%.

7 Анализ ошибки

На рисунке 5(а) показанна гистограмма определения класса нашего алгоритма, по оси y отображаются истинные значения класса, по оси предсказанный алгоритмом класс. На рисунке 5(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. 5 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. 7. Приведя небольшое исследование мы определили, что человек по скелетному представлению может верно определить класс с точностью 84%, полученную гистаграмму можно увидеть на рис. 6.

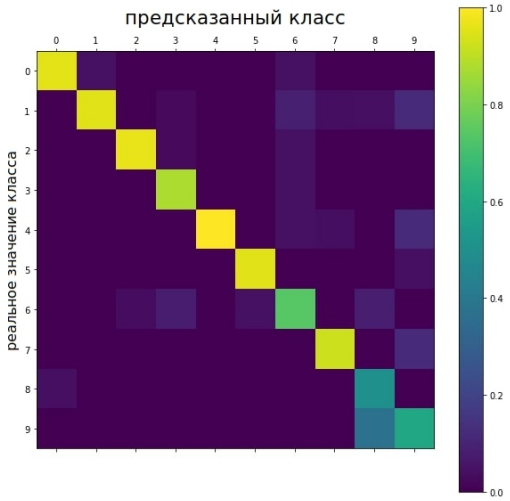


Рис. 6 гистограмма определения класса человеком, по оси y отображаются истинные значения класса, по оси предсказанный алгоритмом класс.

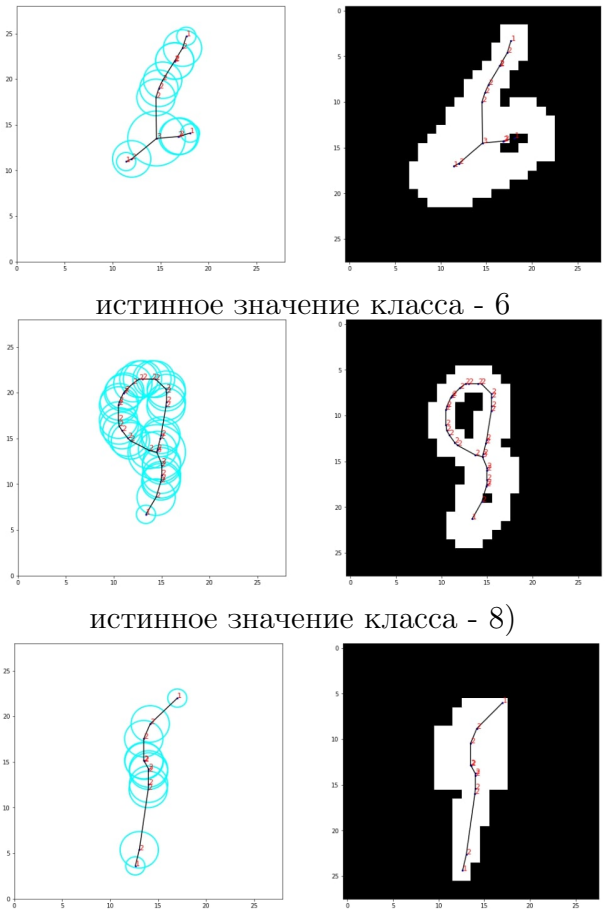


Рис. 7 Примеры скелетных представлений, которые по которым невозможно верно определить истинный класс.

8 Заключение

Проведенный эксперимент показал, что построение модели сверточной нейронной сети на графе скелетного представления символа может дать конкурентноспособный результат. Тем не менее, такие модели все еще проигрывают по качеству обученным сверточным нейронным сетям. В следствии анализа ошибок было обнаружено несколько ошибок алгоритма скелетизации и сделан вывод о том, что для дальнейшего улучшения качества модели имеет смысл улучшать алгоритм получения скелетного представления.

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] Solomatin A.A. *Распознавание текста на основе скелетного представления толстых линий и сверточных сетей*, 2018.

[5] Kutsevol P.N. *Распознавание текста на основе скелетного представления толстых линий и сверточных сетей*, 2018.

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

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

.

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

.

[9] 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.

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

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

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

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

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

Поступила в редакцию