

ПРИМЕНЕНИЕ ГРАФОВЫХ НЕЙРОННЫХ СЕТЕЙ С МЕХАНИЗМОМ ВНИМАНИЯ ДЛЯ СЕМАНТИЧЕСКОЙ СЕГМЕНТАЦИИ ОБЛАКОВ ТОЧЕК

Кадыров Р. А.

*Томский политехнический университет, ИШИТР,
8ВМ22, e-mail: rak32@tpu.ru*

Аннотация

В данной работе приводится обзор передовых моделей нейронных сетей для анализа облаков точек. Предложена новая архитектура модели нейронной сети с применением механизма внимания для анализа облаков точек в рамках задачи семантической сегментации. Разработанная модель протестирована на соответствующем наборе данных.

Ключевые слова: Облака точек, нейронные сети, графовые модели, механизм внимания.

Введение

Современные технологии в области компьютерного зрения и машинного обучения становятся ключевыми элементами в решении сложных задач визуального анализа сцен, таких как классификация объектов на сцене и сегментация отдельных объектов сцены.

Облака точек представляют собой многомерный источник информации, получаемый с лазерных датчиков, таких как LIDAR [1], позволяющий расширить возможности для восприятия окружающей среды при помощи компьютерного зрения.

Алгоритмы обработки облаков точек находят применение в различных областях. Например, в архитектуре [2] и градостроительстве они используются для создания точных 3D-моделей зданий и ландшафтов. С увеличением получаемого объема данных, возникает потребность в эффективном и быстром анализе.

Семантическая сегментация представляет собой актуальное направление в исследованиях компьютерного зрения. В контексте уличных сцен, где разнообразие объектов и их взаимодействие являются сложными задачами для традиционных методов, применение машинного обучения становится неотъемлемым инструментом для достижения точных и эффективных результатов.

Одним из способов обработки объектов, представленных в виде облаков точек является их преобразование в графовую структуру. Активное применение моделей машинного обучения для обработки графовых структур ведется с 2009 года [3, 4, 5, 6]. Появились модели, взаимодействующие с облаками точек как с графами, позволяя использовать расстояние между отдельными точками как важный признак [7, 8], что подтверждает актуальность разработки эффективного способа обработки облаков точек на основе графовых структур.

Целью данной работы является развитие идеи применения графовых нейронных сетей для сегментации облаков точек. В частности, улучшение современных моделей при помощи адаптированного механизма внимания.

Постановка задачи и выбор моделей

С развитием моделей, основанных на сверточных слоях [9] появился большой интерес в том, чтобы адаптировать эти методы для геометрических данных. Однако геометрические данные, в частности, облака точек имеют сильно отличающуюся от изображений структуру, поэтому первые работы в этой области предпринимали попытки как-то структурировать геометрические данные [10]. Впрочем, с развитием глубоких нейронных сетей была разработана модель PointNet [11], открывшая область геометрических глубоких нейронных сетей.

Механизм внимания, впервые примененный в графовых моделях, был представлен в работе авторов Petar Velićković et al., представивших модель Graph Attention Network (GAT) [12]. Схема работы механизма внимания в графовых моделях может быть представлена в виде следующего алгоритма:

1. На входе мы имеем наборы признаков всех вершин графа (формула 1).

$$h = \{\vec{h}_1, \vec{h}_1, \dots, \vec{h}_N\} \quad (1)$$

2. Далее идет обучаемое линейное преобразование в виде однослойного персептрона без функции активации, представленного в виде матрицы весов W . Признаки вершин умножаются на эту весовую матрицу

3. Получившиеся представления конкатенируются и перемножаются с вектором весов “а”, чтобы получить коэффициенты внимания (формула 2).

$$e_{ij} = a(W \vec{h}_i, W \vec{h}_j) \quad (2)$$

Вектор весов “а” представляет собой однослойный персептрон с функцией активации LeakyReLU на выходе.

4. Чтобы сравнить показатели внимания между вершинами, полученные коэффициенты нужно нормализовать. И, чтобы определить, какой узел важнее для какого узла требуется привести коэффициенты к одинаковому масштабу. Для этого на коэффициенты внимания применяется функция активации Softmax.

Для улучшения результатов применяется не один такой механизм, а множественные “головы внимания”. Другими словами, совокупность “механизмов внимания” с разными параметрами, работающие параллельно [13]. Такой механизм позволяет распределить внимание на разных признаках, при этом каждая отдельная “голова внимания” фокусируется на чем-то своем и тем самым расширяется захватываемый моделью контекст признаков.

В свою очередь, для задачи сегментации и классификации облаков точек существует модель, которая способна осуществлять преобразование облаков точек в граф для осуществления обобщения информации на основе графовой свертки – Dynamic Graph Convolutional Network (DGCNN) [7].

В этой работе рассматривается развитие этой модели - Linked Dynamic Graph Convolutional Network (LDGCNN) [8]. Отличие этой модели состоит в том, что авторы соединили признаки с разных слоев с признаками последующих слоев. Архитектура модели представлена на рис. 1.

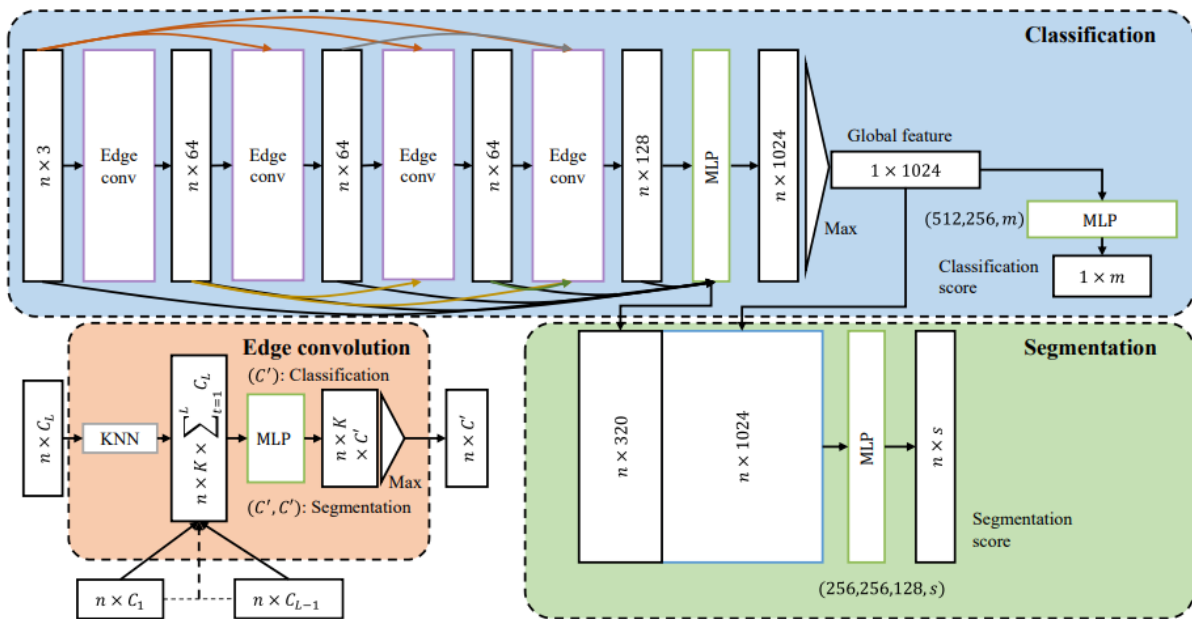


Рис. 1. Архитектура модели LDGCNN

Отличительными особенностями этих моделей является динамическое построение графов. Описанный ниже процесс происходит на каждом сверточном слое (EdgeConv):

1. Для каждой точки строится граф путем соединения ближайших точек и признаков с других слоев, при помощи метода к-ближайших соседей. Все ноды также имеют направленные грани, указывающие на эти ноды (self-loop edges).

2. Далее для каждой построенной грани высчитываются признаки (путем применения нелинейной функции с обучаемыми параметрами).

3. Применяется “свертка по граням”. Эта операция агрегирует признаки с прилегающих граней путем сложения или взятия максимума и обновляет признаковое пространство вершины.

Благодаря обновлению признакового пространства каждой точки, с каждым последующим слоем ближайшими соседями к каждой точке оказываются наиболее “осмысленные” точки, то есть более подходящие структурно.

Рассчитанные признаки далее могут использоваться как для предсказания класса облака точек, так и для сегментации облака точек на разные классы.

Архитектура LDGCNN использованием механизма внимания

Основное нововведение состоит в замене обычной графовой свертки на графовую свертку с использованием “механизма внимания” (рис. 2), описанную выше. Таким образом, это должно улучшить обобщающую способность модели за счет возможности выделять наиболее важные признаки.

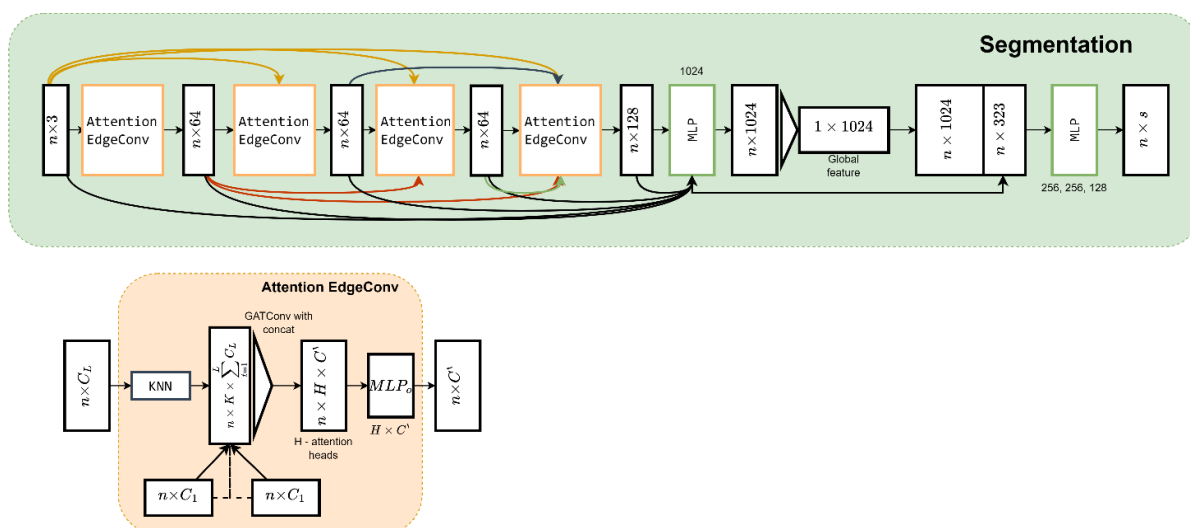


Рис. 2. Архитектура модели LDGCNN с использованием механизма внимания

Как уже говорилось, основное изменение заключается в замене механизма графовой свертки, поэтому сконцентрируемся на нем. Механизм работы следующий.

Также, используя алгоритм К-ближайших соседей, из облака точек формируется граф. Выходом графовой свертки с механизмом внимания является количество выходных признаков, умноженное на количество «голов внимания», используемых в механизме внимания. Для агрегации результатов с каждой «головы внимания» используется однослойный перцептрон. Таким образом, на выходе такой свертки будет аналогичный со входным набор данных с обновленным набором признаков для каждой точки из облака точек.

Выбор набора данных

Для апробации разработанной модели в задаче сегментации был выбран набор данных «Large-scale 3D shape reconstruction and segmentation from shapenet core55» [14], являющийся сокращенным вариантом набора данных – ShapeNetCore [15]. Он содержит облака точек различных объектов с разметкой в виде сегментации отдельных частей объектов.

Набор данных содержит 16881 облаков точек как с 16 глобальными категориями (самолет, стул, стол и т.п.), так и от 2 до 6 категорий частей объектов для задачи сегментации. Разделение производилось на обучающую и валидационную выборки размерами 14007 и 2874, соответственно. Примеры объектов из набора данных приведены на рис. 3.

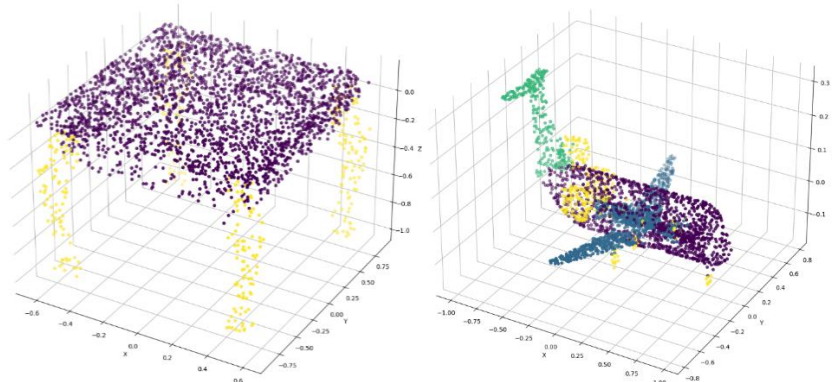


Рис. 3. Примеры облаков точек из использованного набора данных (слева – стол, справа – самолет)

Описание измеряемых метрик

Для оценки результатов использовались описанные ниже метрики.

Точность – количество точек, правильно отнесенных к классу, к количеству точек этого класса

Метрика «Intersection over Union», также известная как Jaccard index, — число от 0 до 1, показывающее, насколько у двух объектов (эталонного - ground true и измеряемого) совпадает внутренний «объем» [16].

Формально, для двух непустых множеств A и B, функция IoU определяется как показано в формуле 3.

$$IoU(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (3)$$

В качестве функции потерь использовалась «Negative Log-Likelihood Loss» (NLL loss). Она используется для оценки различий между распределением вероятностей, предсказанным моделью, и фактическими метками данных.

В ее основе – функция логарифмического правдоподобия (Log-Likelihood), она вычисляет логарифм вероятности предсказанного класса для каждого примера данных. Например, для бинарной классификации такая функция будет выглядеть как показано в формуле 4.

$$LL = \sum_{i=1}^N y_i \log(p_i) + (1 - y_i) \log(1 - p_i), \quad (4)$$

где N - количество примеров данных,

y_i - фактическая метка класса для i-го примера,

p_i – предсказанная вероятность принадлежности i-го примера к классу.

Соответственно, так как это функция правдоподобия, то она должна максимизироваться для достижения требуемого результата. Но если взять отрицательный логарифм, то потребуется минимизировать функцию и получится формула 5.

$$NLL = - \sum_{i=1}^N y_i \log(p_i) + (1 - y_i) \log(1 - p_i) \quad (5)$$

Для сравнения тестировалась реализация модели LDGCNN и ее модифицированная версия с использованием внимания, реализованные средствами PyTorch Geometric [17].

Конфигурации процесса обучения приведены в таблице 1.

Таблица 2

Конфигурации моделей

	LDGCNN	LDGCNN с использованием механизма внимания
Кол-во эпох	50	50
Размер батча		
Тип функции оптимизации	Adam	Adam
Скорость обучения	0.0001	0.0001

Тип функции потерь	negative log likelihood loss	negative log likelihood loss
Кол-во соседних точек для объединения в граф	20	20
Функция агрегации	Max	-
Кол-во голов внимания	-	4

Таким образом, были получены следующие результаты (таблица 2 и рис. 4).

Таблица 3

Результаты обучения

Название метрики	LDGCNN	LDGCNN с использованием механизма внимания
Точность	0.853	0.797
IoU	0.678	0.636
Значение функции потерь	0.548	0.638

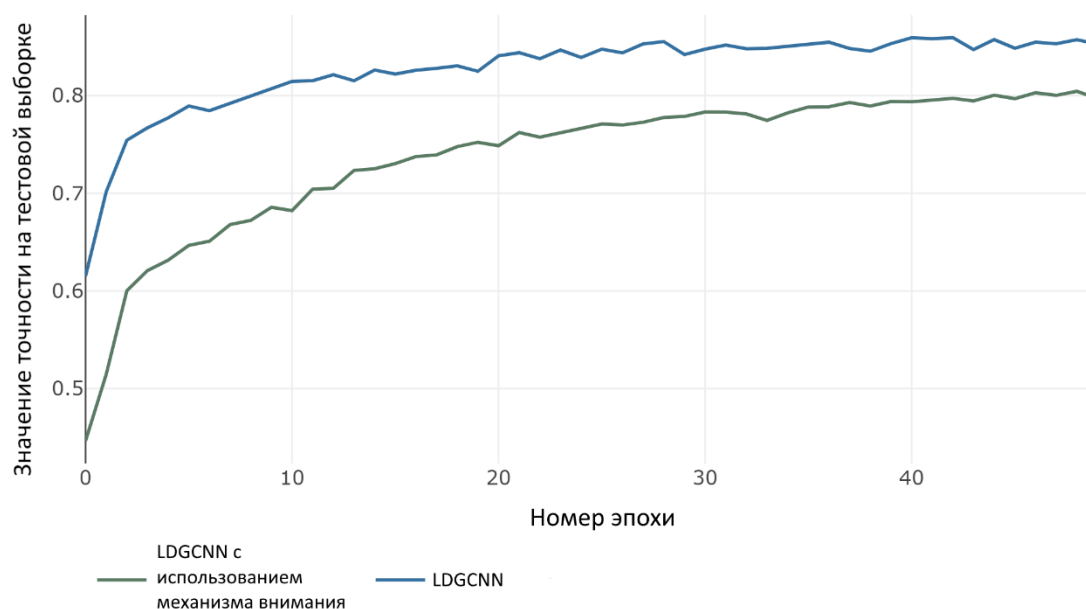


Рис. 4. График изменения точности обученных моделей от эпохи обучения

По результатам можно увидеть, что разработанная модель достаточно близка по основным метрикам к использованной в качестве базы модели LDGCNN, что, в целом, говорит о работоспособности и потенциале модели. Представленные результаты являются промежуточными и далее планируется предпринять шаги по улучшению метрик и скорости работы модели, а также исследовать влияние аугментации данных на качество предсказания.

Заключение

В результате научно-исследовательской работы была предпринята попытка внедрения механизма внимания в модель сегментации облаков точек. Полученная модель была обучена на подготовленном наборе данных и было произведено сравнение основных метрик с моделью, взятой в качестве основы. Таким образом удалось достичь близких метрик, и следующими шагами планируется улучшить результаты работы модели для повышения метрик.

Список использованных источников

1. Лидар // Википедия: сайт. – URL: ru.wikipedia.org/wiki/%D0%9B%D0%B8%D0%B4%D0%B0%D1%80 (дата обращения: 23.03.2024)
2. Облако точек. Как мы развиваем цифровые технологии в строительстве // Habr: сайт. – 2019. – URL: <https://habr.com/ru/companies/jetinfosystems/articles/464593/> (дата обращения: 23.03.2024).
3. Franco S. The Graph Neural Network Model // IEEE Transactions on Neural Networks. – 2009. – Vol. 20. – №. 1. – P. 61-80.
4. Thomas N. Kipf Semi-Supervised Classification with Graph Convolutional Networks / Thomas N. Kipf, Max Welling // ArXiv. – 2016. – 1609.02907v4.
5. Графовые нейронные сети // ИМТО: сайт. – 2022. – URL: https://neerc.ifmo.ru/wiki/index.php?title=Графовые_нейронные_сети (дата обращения: 23.03.2024).
6. Графовые нейронные сети // Школа Анализа данных: сайт. – URL: <https://education.yandex.ru/handbook/ml/article/grafovye-nejronnye-seti> (дата обращения: 23.03.2024).
7. Yue Wang Dynamic Graph CNN for Learning on Point Clouds / Y. Wang, Y. Sun, Z. Liu, S.E. Sarma, M.M. Bronstein, J. M. Solomon \ ArXiv. – 2019. – 1801.07829.
8. Kuangen Zhang Linked Dynamic Graph CNN: Learning on Point Cloud via Linking Hierarchical Features / K. Zhang, M. Hao, J. Wang, C.W. de Silva, C. Fu / ArXiv. – 2019. – P. 1904.10014.
9. Krizhevsky A. ImageNet Classification with Deep Convolutional Neural Networks // Communications of the ACM. – 2017. – Vol. 60. – № 6 – P. 84–90.
10. R. Qi C. Frustum PointNets for 3D Object Detection from RGB-D Data / Charles R. Qi, Wei Liu, Chenxia Wu, Hao Su, Leonidas J. Guibas / ArXiv. – 2018. – 1711.08488.
11. R. Qi, C. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation / Charles R. Qi, Hao Su, Kaichun Mo, Leonidas J. Guibas / ArXiv. – 2017. – 1612.00593.
12. Petar Velickoviĉ. Graph attention networks / P. Velickoviĉ, G. Cucurul, A. Casanova, A. Romero, P. Lio, Y. Bengio // ArXiv. – 2018. – 1710.10903.
13. Vaswani A. Attention Is All You Need / Ashish V., Noam S., Niki P., Jakob U., Llion J., Aidan N. G., Lukasz K., Illia P. / ArXiv. – 2017. – 1706.03762.
14. Large-scale 3D shape reconstruction and segmentation from shapenet core55 / ShapeNet: сайт. – 2017. – URL: <https://shapenet.cs.stanford.edu/iccv17/> (дата обращения: 23.03.2024)
15. ShapeNetCore // Papers With Code: сайт. – URL: <https://paperswithcode.com/dataset/shapenetcore> (дата обращения: 23.03.2024).
16. Intersection over Union (IoU) for object detection // pyimagesearch: сайт. – 2016. URL: <https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/> (дата обращения: 23.03.2024)
17. PyG Documentation // Pytorch Geometric: сайт. – URL: <https://pytorch-geometric.readthedocs.io/en/latest/> (дата обращения: 23.03.2024).