

# Обработка трехмерных облаков точек с помощью нейронных сетей

Кирилл Муравьев

25 декабря 2019 г.

В настоящее время трехмерные облака точек (point clouds) широко используются в робототехнике и компьютерном моделировании. Например, основными данными для управления и навигации беспилотных автомобилей являются облака точек, поступающие с лидаров. Также облако точек является одним из самых удобных форматов для воссоздания трехмерной модели объекта. Часто возникают задачи классификации или сегментации набора точек в трехмерном пространстве. Для решения подобных задач нужна высокоуровневая обработка точек с учетом геометрии, цвета и других характеристик. До появления нейронных сетей задачи классификации объектов по облакам точек решались классическими методами, основанными на различных способах вычисления дескрипторов по окрестностям точек, например, [6], [3]. Однако такие методы обладают недостаточно высоким качеством работы, поэтому в последнее время для обработки облаков точек применяются в основном методы, основанные на нейронных сетях.

В данном тексте рассмотрим два метода генерации признаков по набору точек с помощью нейронных сетей: пространственный и графовый. В пространственном методе облака точек переводятся бинарные сетки заполненности (occuрансу grid), по которым с помощью трехмерных сверток вычисляются целевые признаки. В графовом методе на облаке точек задается структура графа, по которому вычисляются целевые признаки с помощью графовых сверточных сетей.

## 1. Постановка задачи

Дан набор точек в трехмерном пространстве:

$$P \in \mathbb{R}^{n \times 3} = \{p_i = (x_i, y_i, z_i), i = 1, \dots, n\}$$

В задаче классификации облаков точек нужно восстановить принадлежность объекта к одному из  $K$  заданных классов:

$$f(P) \sim c \in \{1, \dots, K\}$$

В задаче сегментации облаков точек нужно восстановить принадлежность каждой точки к одному из  $K$  классов:

$$f(P) \sim \{c_1, \dots, c_n\}, c_i \in \{1, \dots, K\}$$

В задаче детекции объекта нужно найти ограничивающий параллелепипед (bounding box) минимального размера, содержащий объект определенного класса:

$$f(P) \sim (x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max})$$

$$\forall i : c_i = c \hookrightarrow x_i \in [x_{min}, x_{max}], y_i \in [y_{min}, y_{max}], z_i \in [z_{min}, z_{max}]$$

## 2. Пространственный метод

### 2.1. Описание метода

В пространственном методе проводится дискретизация облаков точек - все пространство разбивается трехмерной прямоугольной сеткой на ячейки определенного размера. Если в ячейке есть точка из облака, она помечается как занятая, иначе - как свободная. Так получается трехмерная дискретная структура, над которой можно выполнять обычные свертки:

$$X \in \{0, 1\}^{H \times W \times D};$$
$$X_{i,j,k} = \begin{cases} 0, & \exists p_i = (x_i, y_i, z_i) : x_i \in [ih, (i+1)h], y_i \in [jh, (j+1)h], z_i \in [kh, (k+1)h] \\ 1, & \text{otherwise} \end{cases},$$

где  $h$  - шаг сетки. Также можно вместо флага заполненности помещать в ячейки сетки другие признаки, например, цвет точки, содержащейся в ячейке.

Трехмерная свертка не сильно отличается от двумерной. Она выглядит следующим образом:

$$(K * X)_{i,j,k} = \sum_{i'=-n}^n \sum_{j'=-m}^m \sum_{k'=-q}^q X_{i-i',j-j',k-k'} K_{i'+m,j'+n,k'+q}$$

В общем виде сверточную нейронную сеть можно представить следующим образом:

$$f(X) = (f_1 \circ f_n \circ \dots \circ f_1)(x),$$

$$f_i(X) = \sigma_i(K_i * X),$$

где функции  $\sigma_i$  задают различные нелинейности и нормализации.

## 2.2. Современные нейросетевые архитектуры

На сегодняшний день существует множество нейросетевых архитектур для обработки облаков точек с помощью пространственного метода. Рассмотрим некоторые из них.

Одна из таких архитектур представлена в работе [5]. В этой работе решается задача детекции автомобилей в облаке точек. Облако точек дискретизируется трехмерной сеткой с шагом 10 см. Сеть состоит из трех сверточных слоев с размером ядра 4, и двух слоев развертки (deconvolution). Первый слой развертки предсказывает наличие автомобиля в клетке пространства, а второй - размер параллелепипеда, содержащего автомобиль. Эксперименты в этой работе проводились на коллекции данных KITTI [2]. Перекрытие предсказанного положения автомобиля с истинным составило от 70% до 90% на разных сценах.

Еще одна архитектура для обработки облаков точек представлена в работе [4]. В ней решается задача сегментации облаков точек. Облако точек дискретизируется сеткой с шагом 30 см и затем подается на вход сети, состоящей из трех трехмерных сверток с ядром размера 5 и трехмерной субдискретизацией (3D max pooling) и одного полносвязного слоя для предсказания класса точки. Эксперименты проводились на коллекции облаков точек, собранных с лидаров в городской среде. Точность сегментации при 7 классах достигла 93%.

## 2.3. Преимущества и недостатки метода

Основное преимущество пространственного метода - наличие пространственной структуры, которая позволяет извлекать признаки с помощью сверточных нейронных сетей. Трехмерные свертки можно эффективно вычислять на многопоточном компьютере или на видеокарте. Использование большого количества сверток позволяет добиться высокого качества классификации, сегментации и детекции объектов по облакам точек.

Еще одно несомненное преимущество пространственного метода - легкость реализации. Эффективное параллельное вычисление трехмерных сверток реализовано в большинстве фреймворков глубокого обучения. Дискретизация облака точек также является простым алгоритмом и может быть легко и эффективно реализована.

К недостаткам пространственного метода следует отнести высокую ресурсоемкость, которая неизбежно возникает при использовании трехмерных матриц и сверток на них. Трехмерные свертки требуют не только больших вычислительных затрат, но и больших объемов памяти для хранения весов и промежуточных тензоров. Поэтому применение нейронных сетей с пространственным методом на маломощных бортовых вычислителях может быть затруднено.

## 3. Графовый метод

### 3.1. Описание метода

В графовом методе с помощью определенного алгоритма на облако точек наносится структура графа. Вершинами графа являются точки исходного облака. Ребра задаются чаще всего по расстояниям между точками. Например, один из вариантов задания ребер - провести от каждой точки ребро к ее  $k$  ближайшим соседям в исходном пространстве.

Свертка на графах задается с помощью реберных функций - каждому ребру графа сопоставляется функция, имеющая настраиваемые параметры. Значением свертки в вершине является агрегация (сумма, максимум или другая) реберных функций по всем ребрам, входящим в эту вершину:

$$x'_j = \sum_{(i,j) \in E} h_\theta(x_i, x_j) \text{ или } x'_j = \max_{(i,j) \in E} h_\theta(x_i, x_j)$$

### 3.2. Современные нейросетевые архитектуры

Одной из самых популярных нейросетей для сегментации облаков точек является архитектура PointNet [7]. В этой архитектуре к исходным точкам применяется серия матричных преобразований и полносвязных слоев, а затем - агрегация с помощью global max pooling. Эту агрегацию можно интерпретировать как свертку на графе без ребер, где функция  $h_\theta(i, j)$  зависит только от вершины  $i$ . Несмотря на простоту архитектуры, PointNet обладает довольно высоким качеством работы - точность сегментации на датасете ModelNet40 [10] составила 89.2%.

Еще более высокого качества сегментации удалось добиться с помощью архитектуры PointNet++ [8], которая является усовершенствованной версией архитектуры PointNet. Точность сегментации на датасете ModelNet40 составила 90.7%. В этой архитектуре используются графы, в которых точки объединены в кластеры - компоненты связности. Каждая точка в кластере соединена с центроидом. Агрегация при свертке осуществляется максимизацией.

Еще одна работа по сегментации облаков точек с помощью свертки на графах - PCNN [1]. В этой работе за счет применения операций Extension и Restriction, а также свертки с RBF-ядром удалось добиться точности сегментации в 93.2%.

Одна из последних работ (на декабрь 2019 года), посвященная задаче сегментации облаков точек - это EdgeConv [9]. В архитектуре, представленной в этой работе, на точках строится так называемый kNN-граф: точка соединяется ребрами с  $k$  ближайшими к ней точками в исходном пространстве. В качестве реберной функции  $h_\theta$  берется комбинация вектора ребра и его выходной вершины:

$$(h_\theta(i, j))_m = ReLU(\theta_m(x_i - x_j) + \phi_m x_i)$$

В качестве функции агрегации взят максимум:

$$x'_{im} = \max_{(i,j) \in E} ReLU(\theta_m(x_i - x_j) + \phi_m x_i)$$

Архитектура сети для сегментации состоит из преобразования полносвязным слоем, четырех слоев графовых свертки и серии полносвязных слоев, предсказывающих классы точек. Точность сегментации на датасете ModelNet40 составила 93.5%, что выше, чем у остальных вышеперечисленных архитектур.

### 3.3. Преимущества и недостатки

Основным преимуществом графового метода обработки облаков точек является низкая ресурсоемкость. Так как структура графа не задает дополнительных размерностей, то для хранения сети и вычисления результата памяти нужно ненамного больше, чем для хранения исходного облака точек. Так, модель EdgeConv [9] занимает 21 Мб памяти, модель PointNet++ [8] - 12 Мб. Скорость работы графовых моделей также достаточно высокая - 27 мс на видеокarte у модели EdgeConv и 163 мс у модели PointNet++.

Еще одно преимущество графового метода - его вариативность. Можно задать разные структуры графов и разные функции для свертки, что позволяет наилучшим образом адаптировать архитектуру нейронной сети под конкретную задачу.

Основным недостатком графового метода является трудоемкость реализации. Свертки на графах не реализованы в стандартных фреймворках для глубокого обучения, и их эффективная реализация для параллельных вычислений может занять немалое время.

## Список литературы

- [1] Matan Atzmon, Haggai Maron, and Yaron Lipman. Point convolutional neural networks by extension operators. *arXiv preprint arXiv:1803.10091*, 2018.
- [2] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.
- [3] Yulan Guo, Mohammed Bennamoun, Ferdous Sohel, Min Lu, and Jianwei Wan. 3d object recognition in cluttered scenes with local surface features: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2270–2287, 2014.
- [4] Jing Huang and Suyu You. Point cloud labeling using 3d convolutional neural network. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 2670–2675. IEEE, 2016.

- [5] Bo Li. 3d fully convolutional network for vehicle detection in point cloud. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1513–1518. IEEE, 2017.
- [6] Min Lu, Yulan Guo, Jun Zhang, Yanxin Ma, and Yinjie Lei. Recognizing objects in 3d point clouds with multi-scale local features. *Sensors*, 14(12):24156–24173, 2014.
- [7] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017.
- [8] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017.
- [9] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 38(5):146, 2019.
- [10] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.