

Влияние архитектур кодировщиков на генерацию векторных представлений для моделирования 3D объектов через пространство выпуклых множеств

Д.Н. Грибанов

Самарский национальный исследовательский университет
им. академика С.П. Королева
Самара, Россия
gribanov.dn@ssau.ru

А.В. Мухин

Самарский национальный исследовательский университет
им. академика С.П. Королева
Самара, Россия
mukhin.av@ssau.ru

И.А. Килбас

Самарский национальный исследовательский университет
им. академика С.П. Королева
Самара, Россия
kilbas.ia@ssau.ru

Р.А. Парингер

Самарский национальный исследовательский университет
им. академика С.П. Королева
Самара, Россия
RusParinger@ssau.ru

Аннотация— В данной работе представлено исследование влияния различных архитектур на генерацию векторного представления трехмерных моделей объектов. Центральной моделью является автокодировщик: кодировщик создает векторное представление, а декодировщик на основе вектора моделирует 3D объект через пространство выпуклых множеств. В то время как многие исследования сосредоточены на оптимизации декодировщиков, текущая работа исследует влияние различных архитектур кодировщиков на качество генерации трехмерных моделей при обучении на новом формате данных. Также в данной работе предлагается новый метод обучения кодировщика с модификацией функции ошибки для генерации векторов представления.

Ключевые слова— трехмерные модели, векторное представление, нейронные сети, кодировщик

I. ВВЕДЕНИЕ

Для генерации трехмерных объектов в виде полигональной сетки (далее меш) были разработаны различные методы обучения. Часть работ деформирует шаблон в виде каркаса или сферы в меш [1]. Другая часть работ генерирует более простые виды представления 3D объектов, например, в виде облака точек [2] или в виде неявной функций [3]. Однако представление в виде облака точек имеет разреженный вид в сравнении с мешом. Представление в виде неявной функции имеет проблему интерпретируемости структуры полученного объекта для его трансформации в меш. Для решения проблемы был разработан метод моделирования через абстрактные формы. Подобный метод используется в системе DT-Net [4], где в качестве абстрактной формы используется пространство выпуклых множеств.

Система DT-Net состоит из автокодировщика: модуля кодировщика, что генерирует вектор представления и декодировщика, который из вектора моделирует трехмерный объект в виде меша. Декодировщик состоит из комбинации нейросетевых алгоритмов, из-за чего данная система работает лучше, чем аналоги. При разработке DT-Net основное исследование было сосредоточено на оптимизации декодировщика, который изначально обучается с

кодировщиком на воксельных данных. Когда обучение декодировщика закончено, возможно обучение нового кодировщика на новых видах данных, при этом декодировщик не участвует в обучении. В работе DT-Net обучается новый кодировщик на изображениях с модифицированной архитектурой ResNet18 [5]. Однако новый кодировщик значительно уступает в точности моделирования кодировщику на воксельных данных. В текущей работе исследуется влияние на генерацию векторного представления трехмерных моделей для их последующего моделирования различными архитектурами кодировщиков при использовании новых типов данных, в качестве которых взяты изображения. Также в данной работе представляется новый метод обучения кодировщика с модификацией функции ошибки.

II. НАБОР ДАННЫХ

Для обучения и тестирования нейронных моделей будет использоваться набор данных ShapeNetCore [6]. Всего из набора используется 13 классов, суммарно имеющих более 30000 моделей 3D объектов. На каждый объект имеется 24 рендера с различных ракурсов. Из работы DT-Net взято разбиение на обучающую и тестовую выборку.

III. КРАТКОЕ ОПИСАНИЕ ИССЛЕДОВАННЫХ АРХИТЕКТУР И РЕЗУЛЬТАТА МОДИФИКАЦИЙ ОБУЧЕНИЯ КОДИРОВЩИКА

Рассмотрим подробно оригинальную функцию ошибки. Пусть Z^I вектор представления полученный из кодировщика на основе изображений, а Z^V на основе вокселей, тогда исходная функция ошибки имеет вид:

$$L_{\text{enc}} = |Z^V - Z^I|^2 \quad (1)$$

Однако использование подобной функции ошибки не имеет градиента при обучении кодировщика для указания правильности генерации трехмерных объектов. В данной работе предлагается использовать комбинацию из ранее описанной функции ошибки и $L_{\text{align}}^{\text{dis}}$ (далее 3D функция ошибки) из исходной работы DT-Net. Тем самым при обучении нового кодировщика будет задействоваться кодировщик и декодировщик.

Для рассмотрения L_{align}^{dis} введем несколько понятий. Пусть имеется неотрицательное количество $N_p \in \mathbb{R}$ точек p_i , представляющие из себя координату в трехмерном пространстве. К точке p_i имеется значение функции заполненности o_i , указывающие внутри ли объекта данная точка (значение 1) или снаружи (значение 0). На основе вектора представления Z^I при помощи нейронной сети f из декодировщика генерируется топология объекта $\mathcal{T}^I = f(Z^I)$, из которой далее другая нейронная сеть O из декодировщика формирует значение функции заполненности в точке p_i . Основная задача данной функции ошибки обучить декодировщики аппроксимировать истинное значение функции заполненности o_i . В нашем случае не будут обучаться декодировщики, а лишь кодировщик для правильной генерации топологии $\mathcal{T}^I$. Итоговая функция ошибки имеет вид:

$$L_{align}^{dis} = \frac{1}{N_p} \sum_{i=1}^{N_p} [o_i * \max(O(\mathcal{T}^I, p_i), 0) + (1 - o_i) * (1 - \min(O(\mathcal{T}^I, p_i), 1))]. \quad (2)$$

Итоговая новая функция ошибки представляет из себя сумму L_{enc} и L_{align}^{dis} .

В качестве исследуемых архитектур взяты: ResNet18, ResNet34, ResNet50 [7]. Также были обучены вариации архитектур ResNet18 и ResNet34 с добавлением пакетной нормализаций (указаны как ResNet18BN, ResNet34BN) [8].

Итоги обучения различных архитектур представлены в таблице I. Для оценки моделирования используются две метрики: Light field distance (LFD) [9] и Chamfer distance (CD). В данной таблице так же представлены результаты для DT-Net и схожей работы BSP-NET [10].

Таблица I. СРАВНЕНИЕ РЕЗУЛЬТАТОВ РАЗНЫХ АРХИТЕКТУР И МЕТОДОВ ОБУЧЕНИЯ

Архитектура	3D функция ошибки	Время обучения	Метрика	
			CD (x1000)	LFD
ResNet18	x	10 часов	1,61	3917,38
ResNet18BN		10 часов	1,59	3962,81
ResNet34		12 часов	1,61	3901,00
ResNet34BN		12 часов	1,52	3874,96
ResNet50		16 часов	1,46	3787,63
DT-Net		12 часов	1,58	3866,41
BSP-NET	✓	12 часов	1,56	3946,31
DT-Net		10 дней	1,58	3883,75
ResNet50		12 дней	1,41	3794,59

Согласно таблице, добавление пакетной нормализации ведет к улучшению позиции определения вершин объекта по метрике CD, при этом внешний вид ухудшается по метрике LFD для ResNet18BN, однако для ResNet34BN результаты лучше по обоим метрикам. Результаты DT-Net и BSP-NET уступают ResNet34BN и ResNet50 по метрике CD, хотя ResNet34BN и

проигрывает в метрике LFD архитектуре DT-NET, но абсолютная разница не так значительна. Лучшее всего себя показала архитектура ResNet50, данная архитектура имеет наилучший показатель по метрикам CD и LFD. Использование 3D функции ошибки для оригинальной архитектуры DT-Net не сильно изменяет показания метрик, однако в случае архитектуры ResNet50 улучшается показание метрики CD, при этом метрика LFD не сильно отличаются относительно варианта с оригинальной функцией ошибки.

IV. ЗАКЛЮЧЕНИЕ

В данной работе было исследовано влияния различных архитектур нейронных сетей в качестве кодировщика на генерацию вектора представления для 3D моделирования моделей объектов на новых типах данных на примере изображений. В ходе исследования было показано, что использование ResNet50 приводит к улучшению точности генерации 3D моделей объектов, в сравнении с другими архитектурами и другими работами. При этом была представлена новая функция ошибки, позволяющая на примере ResNet50 улучшить точность расположения вершин согласно метрике CD.

V. БЛАГОДАРНОСТИ

Работа выполнена в рамках государственного задания по теме FSSS-2023-0006.

ЛИТЕРАТУРА

- [1] Ben-Hamu, H. Multi-chart generative surface modeling / H. Ben-Hamu, H. Maron, I. Kezurer, G. Avineri, Y. Lipman // ACM Transactions on Graphics (TOG). – 2018. – Vol. 37(6). – P. 1-15.
- [2] Qi, C.R. Pointnet: Deep learning on point sets for 3d classification and segmentation / C.R. Qi, H. Su, K. Mo, L.J. Guibas // Proceedings of the IEEE conference on computer vision and pattern recognition. – 2017. – P. 652-660.
- [3] Park, J.J. DeepSDF: Learning continuous signed distance functions for shape representation / J.J. Park, P. Florence, J. Straub, R. Newcombe, S. Lovegrove // Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. – 2019. – P. 165-174.
- [4] Hui, K. Neural template: Topology-aware reconstruction and disentangled generation of 3d meshes / K. Hui, R. Li, J. Hu, C. Fu // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. – 2022. – P. 18572-18582.
- [5] He, K. Deep residual learning for image recognition / K. He, X. Zhang, S. Ren, J. Sun // Proceedings of the IEEE conference on computer vision and pattern recognition. – 2016. – P. 770-778.
- [6] Chang, A.X. Shapenet: An information-rich 3d model repository / A.X. Chang, T. Funkhouser, L. Guibas et al. // arXiv:1512.03012. – 2015. DOI: 10.48550/arXiv.1512.03012.
- [7] He, K. Identity mappings in deep residual networks / K. He, X. Zhang, S. Ren, J. Sun // Proceedings of the Computer Vision-ECCV 2016: 14th European Conference. – 2016. – Vol. 4. – P. 630-645.
- [8] Ioffe, S. Batch normalization: Accelerating deep network training by reducing internal covariate shift / S. Ioffe, C. Szegedy // Proceedings of the International conference on machine learning. – 2015. – P. 448-456.
- [9] Chen, D. On visual similarity based 3D model retrieval / D. Chen, X. Tian, Y. Shen, M. Ouhyoung // Computer graphics forum. – Wiley Online Library. – 2003. – Vol. 22(3). – P. 223-232.
- [10] Chen, Z. Bsp-net: Generating compact meshes via binary space partitioning / Z. Chen, A. Tagliasacchi, H. Zhang // Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. – 2020. – P. 45-54.