Научная статья

УДК 004.896
DOI 10.25205/1818-7900-2022-20-1-28-46

Поиск оптимальных 2D моделей нейронной сети U-net для решения задачи семантической сегментации томографических изображений гидратосодержащих образцов

Татьяна Олеговна Колесник [1]
Антон Альбертович Дучков [2]

[1, 2] Новосибирский государственный университет
Новосибирск, Россия

[2] Институт нефтегазовой геологии и геофизики им. А. А. Трофимука
Сибирского отделения Российской академии наук
Новосибирск, Россия

[1] t.kolesnik@g.nsu.ru, https://orcid.org/0000-0002-1362-2584
[2] DuchkovAA@ipgg.sbras.ru, https://orcid.org/0000-0002-7876-6685

Аннотация

Задача семантической сегментации изображений гидратосодержащих пород представляет собой многоклассовую классификацию пикселей каждого 2D-изображения во входном 3D объеме по классам «Гранула», «Флюид», «Гидрат». Классы гранулы и гидрата-флюида являются слабоконтрастными, поэтому для их разделения используется модель, построенная на основе архитектуры сверточной нейронной сети U-Net. Такое решение позволяет провести классификацию пикселей по классу гранулы с точностью более 90 %, в то время как стандартная классификация по пороговому значению имеет точность лишь 56 %.

В данной статье описывается процесс поиска оптимальных моделей архитектуры U-Net посредством настройки набора гиперпараметров. Учитывая ограниченное время обработки большого объема 3D томографических данных, требовалось найти реализацию сети U-Net для достижения наилучшего качества сегментации. С другой стороны, качество сегментации может быть улучшено, посредством анализа взаимосвязей между последовательно идущими изображениями в 3D объеме. Однако 3D реализация сети является очень ресурсоемкой с точки зрения обучения модели и ее работы в режиме вывода. В связи с этим также был осуществлен поиск гиперпараметров с условием достижения сопоставимого качества сегментации при существенном упрощении реализации 2D сети. В дальнейшем найденные оптимальные гиперпараметры 2D модели могут быть использованы для настройки 3D модели сегментации.

Проведя анализ моделей по критериям сходимости и результирующего качества сегментации, мы предложили 2 модели сегментации, оптимальные с точки зрения качества и сложности модели. Рассмотренные в работе подходы гипернастройки модели U-Net имеют общий характер и могут быть применены при работе с другими наборами данных, что позволит улучшить производительность вашего нейросетевого решения как в процессе обучения, так и на стадии его эксплуатации.

Ключевые слова

томография, газогидраты, сверточные нейронные сети, сегментация, U-Net, поиск по сетке гиперпараметров

Для цитирования

Колесник Т. О., Дучков А. А. Поиск оптимальных 2D моделей нейронной сети U-net для решения задачи семантической сегментации томографических изображений гидратосодержащих образцов // Вестник НГУ. Серия: Информационные технологии. 2022. Т. 20, № 1. С. 28–46. DOI 10.25205/1818-7900-2022-20-1-28-46

Search for Optimal 2D Models of the U-net Neural Network for Solving the Problem of Semantic Segmentation of Tomographic Images of Hydrate-Containing Samples

T. O. Kolesnik [1], **A. A. Duchkov** [2]

[1, 2] Novosibirsk State University
Novosibirsk, Russian Federation

[2] Trofimuk Institute of Petroleum Geology and Geophysics
of the Siberian Branch of the Russian Academy of Sciences
Novosibirsk, Russian Federation

[1] t.kolesnik@g.nsu.ru, https://orcid.org/0000-0002-1362-2584
[2] DuchkovAA@ipgg.sbras.ru, https://orcid.org/0000-0002-7876-6685

Abstract

The task of semantic segmentation of 2D-tomographic scans of hydrate-containing rocks is a multi-class classification of pixels of each input image in a set according to the classes “Granule”, “Fluid”, “Hydrate”. Now this is implemented in the form of segmentation by the “Granule” class using the convolutional architecture of the U-Net neural network and classification of pixels unclassified as “Granule” into the “Fluid” and “Hydrate” classes by the threshold value of pixel intensity.

Considering the limited processing time of a large volume of tomographic data, it is necessary to find a compromise between the complexity of the model and the quality of segmentation. On the other hand, it is also required to propose a second, simpler implementation of the network, to extend it to a 3D segmentation model.

The solution of these optimization problems is achieved by tuning the hyperparameters of the U-Net model. To determine which set of network hyperparameters is the best in a particular case, a partial search was performed over the hyperparameter grid, limited by the variables responsible for:

1) the number of trained filters in convolution operations;
2) learning the biases vector for output channels from convolutional operations;
3) choosing an algorithm to increase the resolution in the network decoder part.

This article describes the process of finding optimal models and provides an assumption about the possibilities for their improvement.

Keywords

Tomography, gas hydrates, convolutional neural networks, segmentation, U-Net, hyperparameter grid search

For citation

Kolesnik T. O., Duchkov A. A. Search for Optimal 2D Models of the U-net Neural Network for Solving the Problem of Semantic Segmentation of Tomographic Images of Hydrate-Containing Samples. *Vestnik NSU. Series: Information Technologies*, 2022, vol. 20, no. 1, p. 28–46. (in Russ.) DOI 10.25205/1818-7900-2022-20-1-28-46

Введение

В настоящее время проявляется устойчивый интерес к природным газогидратам как к альтернативному источнику метана [1; 2]. Разрабатываются перспективные методы добычи метана путем разложения газогидратов в горных породах на газ и воду. Одним из методов изучения процессов в горных породах является динамическая рентгеновская томография – повторное сканирование изменяющегося образца и построение его трехмерных изображений в разные моменты времени. Томография может быть проведена с использованием рентгеновского или синхротронного излучения, при этом скорость сканирования и получаемое разрешение для лабораторных устройств рентгеновского излучения на порядок ниже [1] [3].

[1] *Батранин А. В.* Методы и средства контроля основных параметров и характеристик рентгеновских томографов высокого разрешения: Дис. … канд. техн. наук: 05.11.13 / Нац. исслед. Том. политехн. ун-т. Томск, 2016. 148 с.

Для изучения структуры и процессов образования и диссоциации гидрата в породе были проведены лабораторные эксперименты с повторяющимся томографическим сканированием на синхротроне с разрешением 3,45 мкм [4; 5]. На протяжении нескольких часов каждые 15 минут проводилось сканирование трех областей исследуемого образца (сдвинутых по вертикали): область сканирования представляет цилиндр высотой 1,8 мм и диаметром 4,2 мм, время сканирования одной области составляет 70 секунд, а получаемый набор восстановленных данных содержит 512 изображений с разрешением 1224 × 1224 пикселей[2]. Общий объем данных для одного эксперимента составляет несколько терабайт, что исключает их интерпретацию в ручном режиме. В связи с этим актуальными являются задачи организации хранения, а также разработки автоматических методов анализа большого объема томографических данных. Для решения проблемы могут применяться алгоритмы стриминговой томографии, когда обработка отсканированных данных происходит на лету и завершается до начала следующего сканирования. Среди них широкую популярность приобрели алгоритмы сегментации данных [2].

Одним из ключевых этапов анализа томографических данных является сегментация – отнесение каждого вокселя в 3D объеме к определенному типу материала, из которого состоит образец. Сегментация данных необходима для проведения количественной оценки различных параметров исследуемого образца. На основании непосредственно сегментации или оценки характеристик породы может быть проведена локализация целевых объектов и явлений, что позволяет отбирать отдельные области или временные интервалы из общего потока данных, а также облегчает дальнейшую интерпретацию полученных результатов. Кроме того, сегментация данных позволяет построить трехмерную структуру образца.

Сегментация 2D КТ-изображений гидратосодержащих пород представляет собой задачу многоклассовой классификации изображений в оттенках серого. Необходимо каждому пикселю входного изображения присвоить одну из следующих меток класса: «Гранула», «Гидрат» и «Флюид». В настоящее время эта задача реализована в виде сегментации по классу «Гранула» и классификации пикселей, не классифицированных как «Гранула», на классы «Жидкость» и «Гидрат». Второй этап реализован за счет установления порогового значения интенсивности пикселя. Однако при классификации гранул этот метод позволяет достичь качества сегментации только 56 %, так как классы гранулы и гидрата-флюида являются слабоконтрастными. Поэтому на данный момент для их классификации применяется нейросетевое решение, основанное на архитектуре сети U-Net, которое позволяет повысить точность сегментации до 95 % [3].

Целью данной работы являлось проектирование, настройка гиперпараметров и поиск двух моделей (#A, #B) сегментации на основе архитектуры сети U-Net:

- модель #A должна являться наилучшим аппроксиматором с точки зрения максимизации оценки метрики качества на проверочном наборе при ограничении на время сегментации;
- модель #B должна являться существенным упрощением архитектуры модели с допустимой потерей точности до 90 %.

Модель #A в дальнейшем будет использована для сегментации изображений в стриминговой томографии, а также для генерации обучающей выборки для 3D модели сегментации. Модель #B станет основой для 3D модели сегментации, обладающей гораздо большей вычислительной сложностью с точки зрения процесса обучения и ее применения. Для начала рассмотрим особенности настраиваемой архитектуры нейронной сети U-Net.

[2] *Фокин М. И.* Применение нейронных сетей для анализа синхротронных томографических изображений, полученных в процессе формирования и диссоциации газогидратов в образцах: Магистерская дис.: 05.04.01 / Новосиб. гос. ун-т. Новосибирск, 2020. 56 с.

[3] *Колесник Т. О.* Применение сверточных нейронных сетей для сегментации томографических изображений гидратосодержащих образцов // Материалы 59-й Междунар. науч. студ. конф. МНСК-2021. Сер. инфор. технологии. Новосибирск: НГУ, 2021. С. 90.

1. Метод

1.1. Архитектура нейронной сети

U-Net – это полностью сверточная нейронная сеть, которая была создана в 2015 г. для сегментации биомедицинских изображений [6]. Архитектура U-Net, представленная на рис. 1, относится к классу автоэнкодеров. Нисходящие этапы сети называют частью кодировщика (ender'a), восходящие же относятся к части декодировщика сети (decoder'a) и олицетворяют собой сжимающий и расширяющий путь сети соответственно. В данной статье рассмотрена имплементация базовой архитектуры, отличающейся от оригинальной сети 2015 г. частью декодировщика сети. Далее последует краткое описание ее реализации, с использованием общепринятой терминологии и концепций светрточных нейронных сетей [4].

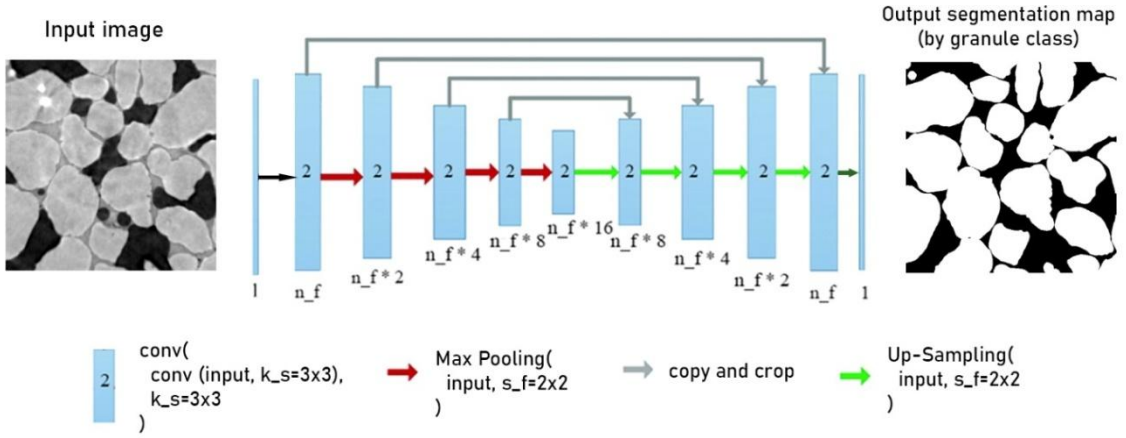

Рис. 1. Архитектура улучшаемой нейронной сети U-Net. Здесь n_f (num_filters) – базовое количество фильтров (в классической архитектуре U-Net 2015 $n\_f = 64$); k_s (kernel_size) – размер обучаемых сверточных фильтров в операциях свертки; s_f (scale_factor) – коэффициент понижения/повышения разрешения для операций Pooling / Upsampling

Fig. 1. Architecture of the improved neural network U-Net. Here n_f (num_filters) is the base number of filters (in the classical U-Net 2015 architecture $n\_f = 64$); k_s (kernel_size) is the size of trained convolutional kernels in convolution operations; s_f (scale_factor) is the coefficient of lowering/increasing the resolution for Pooling / Upsampling operations

Этапы сжимающего пути сети представляют собой применение двух последовательных слоев одношаговой (по вертикали и горизонтали) свертки с ядром 3 × 3 и с функцией активации ReLU, а также слоя максимального пулинга (MaxPooling) с ядом 2 × 2 с шагом 2. В процессе обучения фильтры свертки кодировщика настраиваются таким образом, чтобы выявить наилучшие признаки для классификации гранул. Так, при получении на вход одноканального изображения внутренней структуры исследуемого образца на выходе первого сверточного слоя кодировщика будет сгенерировано многоканальное признаковое представление изображения. Каждый канал в этом представлении можно интерпретировать как реакцию модели по соответствующему признаку / фильтру сверточного слоя на объект классификации во входном изображении.

С каждым сверточным слоем сжимающего пути обучаемые фильтры свертки увеличивают область кодировки исходного изображения – receptive field. Так, по окончании 1-го уровня в одном пикселе признакового представления будет закодирована информация о 6 × 6

[4] URL: https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks

пикселях исходного изображения, а по окончании 4-го этапа – о 76 × 76 пикселях [5] [7]. Соответствующие сверточные фильтры 1-го уровня будут отражать частные признаки классификации, например информацию об интенсивности пикселей, а фильтры 4-го уровня – форму классифицируемых объектов (гранул). Соответственно, максимальная глубина сети U-Net задает максимальный уровень анализа признаков. Так, последняя свертка на 5-м уровне сети будет просматривать область размером 140 × 140 пикселей исходного изображения. При разрешении томографа в 3,45 мкм в такую область попадает от одной до двух гранул среднего размера.

Каждый этап кодировщика призван увеличить количество каналов – карт признаков в выходном признаковом представлении изображения в два раза, за счет удвоения количества применяемых фильтров свертки в сверточных слоях, а также уменьшить размер этих карт признаков в два раза, посредством операции максимального пулинга. Часть декодировщика сети является обратной к кодировщику и призвана привести это признаковое представление к исходному размеру изображения. Каждый этап расширяющего пути содержит UpSumpling слой, обратный пулингу, увеличивающий размер входных карт признаков в два раза, и слои свертки, которые вдвое уменьшают количество наблюдаемых признаков. С целью повышения точности модели также применяется операция “copy and crop”, дополняющая признаковое представление в декодировщике признаковым представлением в кодировщике.

Для сопоставления результирующего многоканального признакового представления с целевой маской сегментации в последнем этапе декодера используется дополнительный слой свертки с ядром 1 × 1 для объединения всех сформированных карт-признаков (каналов). В результате работы сети для одного входного изображения будет сгенерирована одна, сопоставимая по размерам с оригинальным снимком, 2D-карта сегментации, отражающая реакцию модели на целевой класс гранулы для каждого пикселя входного изображения – некоторую количественную оценку уверенности сети в том, что классифицируемый пиксель относится к классу гранулы. Для получения вероятностной оценки выход сети был пропущен через сигмоидную функцию активации [6].

1.2. **Задание пространства поиска оптимальных моделей**

Можно выделить следующие характеристики модели сегментации на основе нейронной сети:

- обобщающая способность модели – определяется количеством параметров модели. Увеличение количества обучающих параметров увеличивает время сегментации и физический размер модели;
- обучающая способность модели – определяется скоростью сходимости. Если модель обладает лучшей сходимостью, ей потребуется меньше эпох на обучение;
- достигнутое качество сегментации на тестовой выборке.

Реализуемые целевые модели (#A, #B) для сегментации 2D томографических снимков гидратосодержащих образцов по классу гранулы представляют собой решение двух оптимизационных задач:

- задача #A – оптимизация с точки зрения достижения наилучшего качества сегментации (предположительно, больше 95 %) при ограничении на время сегментации (полная обработка 3D объемов (1536 изображений разрешением 1224 × 1224 пикселей^2) не должна превышать 15 минут).
- задача #B – оптимизация с точки зрения уменьшения количества обучаемых параметров модели и уменьшения сходимости сети (количества обучаемых эпох) при достижении сопоставимого или лучшего качества сегментации. Если для упрощенной модели #B удастся

[5] URL: https://www.baeldung.com/cs/cnn-receptive-field-size
[6] URL: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

достичь лучшего качества сегментации, она также будет являться решением оптимизационной задачи #A.

В данной работе в качестве минимизируемой функции потерь была использована функция Binary Cross Entropy (BCE) [8], и 1 – BCE в качестве метрики качества. Вероятностная характеристика BCE отражает «уверенность» модели в классификации каждого отдельного пикселя по классу гранулы, и с точки зрения метрики качества такая оценка является более строгой по сравнению с не менее распространенной метрикой Mean Intersection over Union [7] [9]. Mean IoU принимает на вход бинаризированное представление вероятностного выхода нейронной сети, при этом порог бинаризации может быть задан как 50 % или же подобран на стадии обучения с целью улучшения предсказания.

Процесс обучения модели и достижимое качество во многом зависят от инициализации обучаемых параметров модели. Обычно процесс обучения не является воспроизводимым, так как в качестве инициализаторов выбираются разные значения. Однако, зафиксировав индекс random_seed, с которого начнется считывание случайной последовательности чисел для инициализации ядер, мы можем сделать процесс обучения воспроизводимым. В задаче поиска оптимальной модели мы должны проанализировать влияние каждого гиперпараметра на характеристики сходимости модели и на достижимое качество сегментации. Для этого мы фиксируем значение анализируемого гиперпараметра и обучаем модели на инициализаторах из ограниченного множества. В качестве метода инициализации сверточных ядер в данной работе используется метод HeNormal, представляющий собой выборку из сокращенного нормального распределения значений [8]. Средняя оценка для гиперпараметра по всем инициализаторам будет являться основанием для его выбора или исключения из пространства поиска.

Определим базовое пространство поиска Z оптимальных моделей:

- |seed| = 3 – способ инициализации обучаемых параметров модели;
- n_f = {8, 16, 24, 32, 40, 48, 56, 64} – базовое количество обучаемых фильтров свертки в модели U-Net;
- conv_bias_trainable = {True, False} – обучение смещения для сверточных ядер.
- upsample_method = {Nearest Neighbor, Bilinear Interpolation, Subpixel Convolution} – способ повышения разрешения, не требующий обучения.

Вычислительная сложность декодировщика сети может быть уменьшена посредством обучения дополнительного сверточного слоя, следующего после операции Upsampling'a, с сокращением количества выходных фильтров. В данной работе были рассмотрены случаи с сохранением количества фильтров уровня (без обучения дополнительной свертки) и с их уменьшением в 2 и 4 раза для реализаций слоев с Nearest Neighbor или Bilinear Interpolation.

Заданное пространство Z определяет поиск моделей по критериям времени обучения и вывода, а также достижимого качества сегментации среди 336 моделей. Проанализировать все возможные комбинации этих гиперпараметров не представляется возможным. Однако мы можем изначально проанализировать модели по двум признакам: например, по инициализаторам и количеству обучаемых фильтров свертки, сократить пространство поиска и расширять эти модели другими гиперпараметрами, такими как подход к регуляризации данных и способ декодировки.

В сверточных нейронных сетях самой тяжелой операцией является операция свертки. Чем больше сеть будет иметь обучаемых ядер свертки, тем большее количество операций умножения и сложения потребуется выполнить для получения нового признакового представления. Поэтому сокращение пространства поиска следует начать именно с этого гиперпараметра.

[7] URL: https://keras.io/api/metrics/segmentation_metrics/
[8] URL: https://www.tensorflow.org/api_docs/python/tf/keras/initializers/HeNormal

1.3. **Особенности процесса тестирования моделей**

Исследуемый образец проходит сканирование каждые 15 минут. Естественным ограничением стриминговой томографии является требование, чтобы отсканированные данные были полностью обработаны до начала следующего сканирования. Под обработкой в данном случае понимается восстановление трех 3D объемов, их полная сегментация в несколько этапов (1. Сегментация по классу гранулы; 2. Вычитание пикселей, классифицированных как гранула; 3. Классификация оставшихся пикселей изображения) с последующим анализом результатов. На практике также могут применяться алгоритмы пред- / постобработки данных, что также увеличит время, необходимое для обработки входного потока изображений. В связи с этим в данной работе приводится только оценка времени сегментации по классу гранулы различными реализациями сети U-Net.

Для нахождения оптимальных моделей по сетке гиперпараметров были использованы ресурсы вычислительного кластера научно-образовательного центра НГУ-Газпромнефть. В целях воспроизводимости процесса обучения моделей весь процесс обучения был реализован на процессоре Intel® Xeon® Gold 6254 с кэшем 24,75 MB и 386 Гб ОЗУ. Помимо фиксации всех инициализаторов обучаемых параметров моделей, обработка данных также являлась унифицированной и производилась без средств асинхронной побатчевой оптимизации в процессе обучения моделей. Фиксация способа инициализации обучаемых параметров делает процесс обучения воспроизводимым, а анализ моделей на нескольких инициализаторах позволяет оценить вклад каждого варьируемого гиперпараметра в настройку модели. Процесс тестирования времени сегментации по классу гранулы был реализован как на CPU, так и на графическом ускорителе Tesla V100 с 16 GB видеопамяти, подключенному через PCIe v3. Важным критерием параллельной обработки является оптимальная загрузка процессоров и организация кэширования данных. В данной работе было проведено тестирование времени исполнения с варьированием количества изображений в батче, равное $B = 2^i$, где $i = 2, .., 9$ для CPU и $B = 2^j$, где $j = 1, .., 5$ для GPU. В качестве меры центральной тенденции была выбрана медиана.

2. **Описание исследования**

2.1. **Варьирование значения количества обучаемых фильтров свертки**

В архитектуре нейронной сети U-Net, приведенной на рис. 1, на первом уровне сети изображение кодируется 64 признаковыми представлениями. С каждым последующим уровнем кодировщика количество признаков увеличивается вдвое. Обозначим количество обучаемых фильтров свертки на первом этапе сети за *n_f*. Так, на k этапе кодировщика количество фильтров вычисляется по формуле $n\_f_k = 2^{k-1} * n\_f$. В U-net 2015 максимальная глубина сети $K = 5$, при этом порядок уровней в декодировщике является обратным к уровням в кодировщике, и количество блоков декодировщика равно $K - 1$.

Зададим подпространство z_1 пространства поиска Z по гиперпараметру *n_f* и зафиксируем 3 значения *seed* для инициализации сверточных весов случайными значениями в методе HeNormal. Таким образом, подпространство z_1 задано декартовым произведением множеств *n_f* и *seed*, где *n_f* = {8, 16, 24, 32, 40, 48, 56, 64}. Обучение всех 24 моделей, соответствующих подпространству z_1, происходило на протяжении 15 эпох с использованием алгоритма с адаптивной скоростью обучения Adam [10].

Прежде всего было необходимо проанализировать сложность полученных моделей и оценить время на сегментацию 3D объема, состоящего из 512 изображений с разрешением 1224×1224 px^2. При размере снимка, не кратного $2\text{^}K - 1$ (K – глубина сети), может произойти потеря данных в части кодировщика при операции максимального объединения, что приведет к несоответствию размеров представлений в кодировщике и декодировщике сети. Эту проблему можно решить, предварительно уменьшив размер входных снимков до 1024^2,

построить карту сегментации, затем увеличить ее размер до размера оригинальных снимков. В данной работе для понижения / повышения разрешения используется линейная интерполяция. Данная пред- / постобработка снимков и масок также включена в общее время, необходимое для сегментации по классу гранулы, а качество тестовых изображений было оценено по разрешению 1024^2, а не по оригинальному размеру снимков.

В табл. 1 (во втором столбце) приведено количество обучаемых параметров модели, а в табл. 2 отображено затраченное время на сегментацию одного объема в минутах без учета выгрузки отсегментированных масок сегментации в выходную директорию. Так, самая сложная модель с $n\_f = 64$ фильтрами имеет в 64 раза больше обучаемых параметров, чем модель с $n\_f = 8$, но при этом ей потребуется в среднем в 8 раз больше времени для сегментации одного 3D объема на CPU.

Таблица 1

Характеристики сложности моделей по количеству обучаемых фильтров свертки и достигнутое значение метрики качества Binary Cross Entropy (BCE) на проверочном наборе

Table 1

Model complexity characteristics by the number of trained convolution filters and the achieved value of the Binary Cross Entropy (BCE) quality metric on validation set

Количество базовых фильтров (n_f)	Количество параметров модели	Размер модели (МБ)	Усредненное по инициализаторам значение метрик качества 1-BCE для моделей на проверочном наборе (%)
8	490 256	5,78	86,25
16	1 960 864	22,61	90,03
24	4 411 824	50,66	92,15
32	7 843 136	89,93	91,77
40	12 254 800	140,41	91,56
48	17 646 816	202,12	92,23
56	24 019 184	275,05	92,33
64	31 371 904	359,19	91,86

Как было отмечено ранее, процесс тестирования во многом зависит от характеристик оборудования, размера данных и количества изображений в батче. Так, для такого входного размера снимков максимальный размер батча на GPU составил всего 4 снимка в формате float32 для всех моделей U-Net, сложность которых превышает 24 миллиона параметров. Это связано с тем, что для графического процессора требуется обеспечить одновременное хранение модели, данных и их промежуточное представления в глобальной памяти видеокарты. При организации работы на CPU также следует учитывать характеристики используемого оборудования. Так, например, самая тяжелая модель с базовым числом фильтров $n\_f = 64$ в своем последнем слое кодировщика будет работать с тензорным представлением снимка $B \times H / 2^4 \times W / 2^4 \times (n\_f = 64 * 2^4)$, где B – количество изображений в батче, H / W – высота / ширина входного изображения. При размере снимка $H = W = 1024$ и $B = 1$ для хранения промежуточного тензора потребуется

$$\frac{\frac{1024^2}{2^8} * 64 * 2^4 * 32}{1024^2 * 8} \text{ МБ} = 16 \text{ МБ [11].}$$

Максимальный тестируемый размер батча на CPU – 512. Тогда промежуточный размер тензора на $K = 5$ уровне сети U-Net занимает 8Гб, что приводит к увеличению количества обращений к глобальной памяти.

Таблица 2

Тестирование времени генерации масок сегментации для одного 3D объема на CPU и GPU

Table 2

Testing the generation time of segmentation masks for one 3D volume on CPU and GPU

Количество базовых фильтров (n_f)	Медианное время сегментации	
	CPU (мин)	GPU (мин)
8	0,88	0,35
16	1,41	0,4
24	2,04	0,5
32	2,74	0,58
40	3,75	0,79
48	4,68	0,93
56	5,9	1,12
64	7,02	1,3

Если потребуется сохранить результат работы модели на батче данных в директорию, время обработки на CPU займет от 0,1 до 0,16 минуты в зависимости от размера батча и сложности модели. При обработке батча моделью на GPU для сохранения результата требуется скопировать сгенерированные маски сегментации на CPU. В среднем время сохранения результата при обработке на GPU занимает порядка 0,16–0,2 минуты.

Размер батча на CPU варьировался с 4 до 512, при этом оптимальная конфигурация была достигнута при размере батча в 16 изображений для моделей, базовое число n_f фильтров в которых больше 16. Для моделей с числом фильтров $n\_f = 8$ и 16, батчи с 32–128 изображениями обеспечивают оптимальную загрузку процессора. На GPU максимальный размер батча составил 32 изображения для модели с $n\_f = 8$, для остальных моделей размер батча был уменьшен, чтобы избежать ошибок с переполнением памяти. Для всех моделей размер батча в 2 снимка обеспечил оптимальную загрузку видеокарты.

Таким образом, вычисление моделей на видеокарте позволяет добиться существенного прироста производительности для всех моделей. Так, для модели с базовым числом фильтров $n\_f = 8$ ускорение составляет порядка 200 %, а для модели с $n\_f = 64$ – 530 %. Однако для моделей с количеством фильтров $n\_f = 56$, 64 может возникнуть проблема их использования в стриминговой томографии, где доступ к GPU на узле может быть ограничен. Кроме того, так как весь процесс обучения выполняется на CPU небольшими батчами изображений, и требуется также вычислять значения алгоритма обратного распространения ошибки, обучение и настройка гиперпараметров этих моделей затруднено временными ограничениями доступа к вычислительному кластеру. При этом модель с $n\_f = 48$ фильтрами обладает меньшей вычислительной сложностью и на данном этапе анализа пространства гиперпараметров позволяет добиться лучшего или сопоставимого качества сегментации. В связи с этим далее

мы будем рассматривать модели со сложностью, не превышающей 20 миллионов параметров. Однако при необходимости после подбора оптимальных гиперпараметров для модели с $n\_f = 48$ эти же подходы могут быть применены к моделям с большим числом фильтров.

2.2. **Регуляризация на уровне обучения сверточных слоев**

При обучении сверточных слоев также можно задать обучение дополнительного вектора, отвечающего за смещение каждого выходного канала из слоя свертки [9]. Если предсказания модели смещены достаточно сильно относительно целевых ответов, может потребоваться увеличить количество эпох обучения, так как в течение нескольких из них настройка весов будет направлена не на выявление признаков классификации, а на уменьшение смещения. При анализе количества сверточных фильтров сети мы использовали всего лишь 15 эпох для обучения. Возможно, более простые модели могли недообучиться и не выявить важных закономерностей и свойств признаков. Поэтому обучение смещений позволит за несколько эпох привести эти модели к значениям, близким к оптимальным. Модели, обладающие большим числом фильтров, вследствие увеличенной обобщающей способности достаточно быстро минимизируют значение смещения, поэтому ожидается, что сходимость смещенной сети будет являться сопоставимой.

Альтернативным, более распространенным способом регуляризации выхода является подход побатчевой нормализации данных [12], который помимо смещения включает в себя настройку поканального стандартного отклонения. Нормализация на уровне сверточных слоев или их выходов в нейронной сети позволяет ввести регуляризацию обучаемых параметров, что приводит к улучшению сходимости сети и, соответственно, достижению лучшего качества сегментации за то же количество эпох.

В данной работе рассмотрен подход к регуляризации на уровне сверточных слоев. Операция смещения практически не потребляет ни память, ни вычислительное время, однако при этом благоприятно сказывается на процессе обучения. Требовалось провести анализ достигнутого качества сегментации на множестве z_1, расширенном гиперпараметром обучаемого сдвига в сверточном слое: $z_2 \subset Z$; $z_2 = z_1 \times$ conv_bias_trainable, где conv_bias_trainable – булево значение, отвечающее за обучение вектора смещения; conv_bias_trainable = *False* означает, что нормировка данных не применяется, т. е. $z_1 \times$ (conv_bias_trainable' = {*False*}) = z_1. Таким образом, требовалось протестировать $|z_2| = 48$ моделей (3 инициализатора * 8 значений фильтров n_f * на обучение векторов смещений), 24 из которых уже были обучены. Для сопоставления процесса обучения новые модели также были обучены на протяжении 15 эпох.

Сравним достижимое качество моделей с обучаемыми векторами смещений для каждой операции свертки сети (табл. 3).

Уже на этом этапе анализа времени сегментации мы можем разделить модели на две группы:

a) сложные (n_f = {32, 40, 48, 56, 64}) – решающие оптимизационную задачу уровня A;

b) простые (n_f = {8, 16, 24}) – решающие оптимизационную задачу уровня B;

Подход регуляризации на уровне сверточных слоев позволил существенно улучшить качество всех моделей, поэтому в дальнейшем, учитывая возросшее время на обучение сложных моделей, мы будем рассматривать для расширения гиперпараметрами сокращенное множество z_2':

$$z_2' \subset z_2;$$

$$z_2' = \left(n\_f' = \{8, 16, 24, 32, 40, 48\}\right) \times \left(\text{conv\_bias\_trainable} = \{True\}\right).$$

[9] URL: https://deepai.org/machine-learning-glossary-and-terms/bias-vector

Таблица 3

Характеристики сложности моделей по количеству обучаемых фильтров свертки и достигнутое значение метрики качества 1-BCE на проверочном наборе при настройке моделей по параметру регуляризации свертки

Table 3

Characteristics of model complexity by the number of trained convolution filters and the achieved value of the 1-BCE quality metric on validation set when tuning models by the convolution regularization parameter

Количество базовых фильтров (n_f)	Количество обучаемых весов сверток	Дополнительные обучаемые параметры смещения	Усредненное по инициализаторам значение метрик качества 1-BCE для моделей (%)	
			Conv	Conv + $\overline{bias}$
8	490 256	+ 737	86,25	89,17
16	1 960 864	+ 1 473	90,03	91,83
24	4 411 824	+ 2 209	92,15	93,15
32	7 843 136	+ 2 945	91,77	93,44
40	12 254 800	+ 3 681	91,56	93,35
48	17 646 816	+ 4 417	92,23	93,72
56	24 019 184	+ 5 153	92,33	92,78
64	31 371 904	+ 5 889	91,86	93,38

3. **Настройка способа декодировки сети**

Все ранее рассмотренные модели U-Net с глубиной $K = 5$ уровней уменьшали разрешение входного изображения в 2^4 раза и использовали интерполяцию Nearest Neighbor в части декодировщика сети для повышения разрешения закодированного признакового представления [10]. Так, с каждым уровнем сети, признаковое представление изображения будет дважды увеличено в размере посредством декодировки 1 пикселя во входном представлении 4 пикселями (2×2) с тем же значением интенсивности. Альтернативным способом является применение билинейной интерполяции, которая вычисляет промежуточные значения интенсивностей пикселей в генерируемом представлении на основании анализа четырех интенсивностей ближайших пикселей во входном представлении. Операция ближайшего соседа и интерполяция не изменяет количество каналов входного признакового представления.

Еще одним способом реализации слоя повышения дискретизации, не требующего обучения весовых параметров, является алгоритм субпиксельной свертки, основанный на операции «глубина в пространство» (depth to space) [11]. Субпиксельная конволюция использует понижение количества каналов входного тензора размером $H \times W \times C * r^2$ для генерации тензора большего размера $H * r \times W * r \times C$ [13]. Здесь H – высота представлений, W – ширина представлений, C – количество представлений, r – коэфициент повышения разрешения. Уменьшение количества просматриваемых признаковых представлений в операциях субпиксельной конволюции в 2^2 раза позволяет уменьшить вычислительную сложность следующей операции свертки и модели в целом, но также может сказаться на результирующем качестве сегментации в лучшую или худшую сторону. В оригинальной статье 2015 г. предлагается использовать слой up-convolution, который представляет собой комбинацию слоев Up-

[10] URL: https://www.tinaja.com/glib/pixintpl.pdf
[11] URL: https://www.tensorflow.org/api_docs/python/tf/nn/depth_to_space

Sampling + convolution (2 × 2) с уменьшением количества признаков в 2 раза [6]. В рассматриваемой имплементации дополнительная операция свертки не была использована.

Рассмотрим вычислительную сложность операции свертки на 4 уровне декодировщика сети интерполяционными подходами без обучения дополнительной свертки (рис. 2, *2.1*). Размер входного тензора для операции повышающей дискретизации имеет вид $H_{in} \times W_{in} \times C_{in}$ и равен $\frac{H}{2^4} \times \frac{W}{2^4} \times n\_f * 2^4$, где H_{in}, W_{in} и C_{in} – ширина, высота и количество каналов / признаков во входном признаковом представлении, а H, W – размер оригинального снимка. Слой upSampling'а увеличит размер тензора до $\frac{H}{2^3} \times \frac{W}{2^3} \times n\_f * 2^4$, после чего этот выходной тензор будет дополнен $n\_f * 2^3$ признаками из кодировщика операцией "copy and crop". Следующий сверточный слой с размером ядра $(K_H, K_W) = 3 \times 3$ делает преобразование этих $n\_f * (2^4 + 2^3) = n\_f * 24$ признаков (C_{in}) до $n\_f * 2^3$ каналов на выходе (C_{out}), сохраняя размер карт признаков. Сложность вычисления любого слоя в сети может быть измерена как общее количество операций с плавающей запятой (FLOP), необходимых для вычисления выходного представления [12]. Количество операций для сверточного слоя вида $(C_{out};\ K_H; K_W; C_{in})$ с размером выхода $H_{out} \times W_{out} \times C_{out}$ вычисляется по формуле 1:

$$FLOPS(Conv(C_{out};\ K_H; K_W; C_{in}))\ =\ H_{out} * W_{out} * C_{out} * K_H * K_W * C_{in} * 2 \qquad (1)$$

Таким образом, сложность вычисления первого сверточного слоя декодировщика равна

$$\boldsymbol{FLOPs}\left(\mathbf{Conv}\left(\boldsymbol{n_f} * \mathbf{2^3}; \mathbf{3}; \mathbf{3}; \boldsymbol{n_f} * \mathbf{24}\right)\right) = \frac{H}{2^3} * \frac{W}{2^3} * \left(n_f * 2^3\right) * 3 * 3 * \left(n_f * 24\right) * 2 =$$

$$= \mathbf{54} * \boldsymbol{H} * \boldsymbol{W} * \boldsymbol{n\_f^2}.$$

При этом количество обучаемых параметров только этого сверточного слоя будет равно (без учета обучаемого смещения каждого выхода) $C_{out} * K_H * K_W * C_{in} = (n\_f * 24) * 3 * 3 * (n\_f * 2^3) = 1728 * n\_f^2$. Следовательно, простая модель класса #B с $n\_f = 16$ будет иметь в 9 раз меньше обучаемых параметров для этого слоя, чем модель класса #A с $n\_f = 48$.

Теперь приведем вычислительную сложность первой свертки декодировщика в классической архитектуре сети U-Net 2015 (рис. 2, *2.2*). Здесь мы используем алгоритм интерполяции вместе с дополнительной сверткой, уменьшающей количество фильтров в 2 раза. Таким образом, после дополнения представления слоями из кодировщика мы получили количество слоев, соответствующее $n\_f * (2^3 + 2^3)$, вместо $n\_f * (2^4 + 2^3)$. Вычислительная сложность первой свертки кодировщика уменьшится, однако требуется также учитывать сложность вычисления дополнительной свертки. Полная сложность вычисления для двух сверток

$$\boldsymbol{FLOPS} = \begin{matrix} \boldsymbol{FLOPS}(\mathbf{Conv}(\boldsymbol{n\_f} * \mathbf{2^3}; \mathbf{2}; \mathbf{2}; \boldsymbol{n\_f} * \mathbf{2^4})) + \\ \boldsymbol{FLOPS}(\mathbf{Conv}(\boldsymbol{n\_f} * \mathbf{2^3}; \mathbf{3}; \mathbf{3}; \boldsymbol{n\_f} * \mathbf{2^4})) \end{matrix}$$

равна

$$FLOPS = \begin{matrix} \frac{H}{2^3} * \frac{W}{2^3} * \left(n_f * 2^3\right) * 2 * 2 * \left(n_f * 2^4\right) * 2 + \\ \frac{H}{2^3} * \frac{W}{2^3} * \left(n_f * 2^3\right) * 3 * 3 * \left(n_f * 2^4\right) * 2 \end{matrix} = (2^9 + 2^7 * 9) * {n_f}^2 * \frac{H}{2^3} * \frac{W}{2^3} * 2 =$$

$$= \mathbf{52} * \boldsymbol{H} * \boldsymbol{W} * \boldsymbol{n\_f^2},$$

т. е. вычислительная сложность классического декодировщика меньше, чем в рассматриваемой имплементации, при этом количество обучаемых параметров равно $(n\_f * 2^3) * 2 * 2 *$

[12] URL: https://www.thinkautonomous.ai/blog/?p=deep-learning-optimization

$(n\_f * 2^4) + (n\_f * 2^3) * 3 * 3 * (n\_f * 2^4) = 1664 * n\_f^2$, по сравнению с $1728 * n\_f^2$ для сети без обучения дополнительной свертки. Стоить отметить, что такой подход позволил не только уменьшить вычислительную сложность модели, но и добавить дополнительные обучаемые трансформации для слоя повышения разрешения, что может благоприятно сказаться на качестве обучения моделей.

Если же мы используем субпиксельную конволюцию для понижения разрешения (рис. 2, *2.3*), то на вход сверточного слоя поступит тензор размера $\frac{H}{2^3} \times \frac{W}{2^3} \times (n_f * (2^{(4-2)} + 2^3))$, тогда

$$\boldsymbol{FLOPS}\left(\mathbf{Conv}\left(\boldsymbol{n_f} * \mathbf{2^3}; \mathbf{3}; \mathbf{3}; \boldsymbol{n_f} * \mathbf{12}\right)\right) = \frac{H}{2^3} * \frac{W}{2^3} * \left(n_f * 2^3\right) * 3 * 3 * \left(n_f * 12\right) * 2 =$$

$$= \mathbf{27} * \boldsymbol{H} * \boldsymbol{W} * \boldsymbol{n\_f^2}.$$

Таким образом, операция преобразования глубины в пространство позволяет уменьшить вычислительную сложность модели в 2 раза.

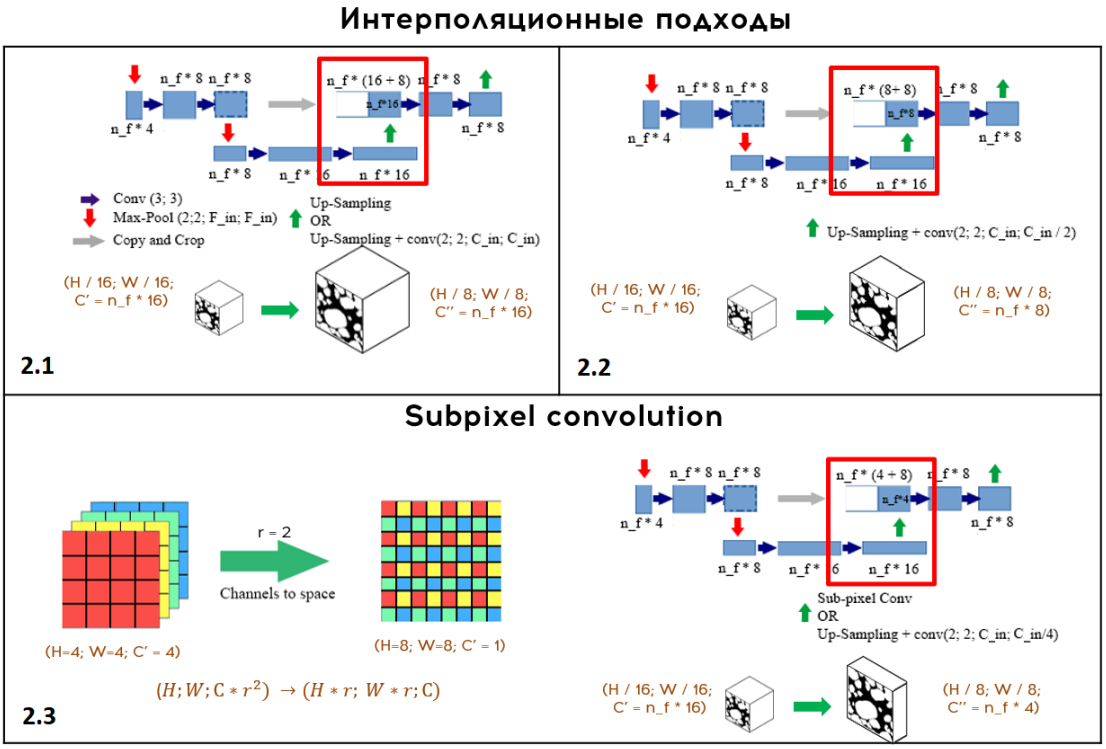

Рис. 2. Подходы к повышению разрешения признаковых представлений изображения:
2.1 – с сохранением количества признаков предыдущего уровня;
2.2 – с сокращением количества признаков в 2 раза;
2.3 – сС сокращением количества признаков в 4 раза

Fig. 2. Approaches to increasing the resolution of feature representations of an image:
2.1 – with the preservation of the number of features of the previous level;
2.2 – with a reduction in the number of features by 2 times;
2.3 – with a 4-fold reduction in the number of features

В данной работе в качестве эксперимента также были протестированы модели, использующие после слоя UpSamplinga свертку, уменьшающую количество признаковых представлений в 4 раза, по аналогии со слоем субпиксельной конволюции. Таким образом, после до-

полнения представления слоями из кодировщика мы получили количество слоев, соответствующее $n\_f * (2^2 + 2^3)$. Полная сложность вычисления для двух сверток

$$\boldsymbol{FLOPS} = \begin{matrix} \boldsymbol{FLOPS}(\mathbf{Conv}(\boldsymbol{n_f} * \mathbf{2^2}; \mathbf{2}; \mathbf{2}; \boldsymbol{n_f} * \mathbf{2^4})) + \\ \boldsymbol{FLOPS}(\mathbf{Conv}(\boldsymbol{n\_f} * \mathbf{2^3}; \mathbf{3}; \mathbf{3}; \boldsymbol{n\_f} * \mathbf{12})) \end{matrix}$$

равна

$$FLOPS = 2 * \frac{H}{2^3} * \frac{W}{2^3} * \left(\left(n_f * 2^2\right) * 2 * 2 * \left(n_f * 2^4\right) + \left(n_f * 2^3\right) * 3 * 3 * \left(n_f * 12\right)\right) = \\ = \mathbf{35} * \boldsymbol{H} * \boldsymbol{W} * \boldsymbol{n\_f^2}.$$

Количество обучаемых параметров при этом равно

$$(n\_f * 2^2) * 2 * 2 * (n\_f * 2^4) + (n\_f * 2^3) * 3 * 3 * (n\_f * 12) = 1120 * n\_f^2.$$

Новое пространство поиска $z_3 \subset Z$:

$$z_3 = z_2' \times decoder\_method = \{\begin{matrix} \text{Nearest Neighbor}, \\ \text{Bilinear Interpolation}, \\ \text{Nearest Neighbor} + conv(C_{in}; 2; 2; \mathrm{C_{out}} = C_{in}/2), \\ \text{Bilinear Interpolation} + conv(C_{in}; 2; 2; \mathrm{C_{out}} = C_{in}/2), \} \\ \text{Nearest Neighbor} + conv(C_{in}; 2; 2; \mathrm{C_{out}} = C_{in}/4), \\ \text{Bilinear Interpolation} + conv(C_{in}; 2; 2; \mathrm{C_{out}} = C_{in}/4), \\ \text{Subpixel Convolution} \end{matrix}$$

Для ускорения процесса обучения моделей, соответствующих пространству поиска z_3, был прменен подход Transfer learning [13]. Так, каждая модель использовала соответствующие множеству z_2 предобученные веса кодировщика сети, ранее оптимизированные со слоями Nearest Neighbor в этапе декодера, а часть декодировщика заменена в соответствии с анализируемым гиперпараметром повышения разрешения и заново инициирована теми же начальными значениями. Так как часть декодировщика сети не являлась обученной, первые 10 эпох веса кодировщика были зафиксированы. На протяжении последующих 20 эпох был применен подход fine-tuning [13], при котором происходило полное обучение модели. Таким образом, предобученный кодировщик позволил ускорить процесс обучения и изучить сходимость моделей на более поздних этапах обучения.

Требовалось протестировать $||z_3|| = 126$ моделей. В табл. 4 приведено качество сегментации моделей с группировкой (усреднением) по количеству базовых фильтров и способу декодировки. Результирующая сложность моделей приведена в табл. 5: во второй строке указано количество обучаемых параметров для моделей, не использующих понижение каналов в операции дискретизации, а в последующих строках – отклонение, численно отображающее уменьшение количества параметров при обучении дополнительной свертки 2 × 2 с понижением количества выходных каналов.

При обучении модели с базовым количеством фильтров *n_f* = 8 и сокращением количества параметров посредством обучения дополнительных трансформаций было отмечено, что качество сегментации и сходимость моделей сильно зависят от способа инициализации параметров переобучаемого декодировщика. При обучении моделей с большим числом параметров влияние начальных инициализаторов уменьшалось.

Интерполяция ближайшим соседом уступает по результирующему качеству сегментации билинейной интерполяции, но усложнение самого способа интерполяции увеличивает время на прохождение прямого и обратного проходов сети. Ранее при тестировании моделей по количеству базовых фильтров были определены размеры батчей, обеспечивающих наилучшую загрузку процессора и видеокарты. В табл. 6 приведена оценка времени сегментации

[13] URL: https://www.tensorflow.org/tutorials/images/transfer_learning?hl=en

одного 3D объема для моделей разного уровня сложности, вычисленная как среднее значение времени сегментации на 3-х повторных запусках модели при оптимальном размере батча. При запуске моделей на CPU было выявлено, что время вывода для моделей, использующих билинейную интерполяцию в части сетевого декодера, увеличилось в два раза. При тестировании на GPU время вывода при использовании разных интерполяционных подходов является сопоставимым. С другой стороны, обучение дополнительных сверточных слоев, несмотря на сокращение количества обучаемых параметров, приводит к увеличению времени прямого прохода сети (вывода).

Таблица 4

Достигнутое значение метрики качества 1-BCE на проверочном наборе при настройке моделей по способу декодировки с группировкой по количеству базовых обучаемых фильтров свертки *n_f*

Table 4

Achieved value of the 1-BCE quality metric on validation set when configuring models according to the decoding method with grouping by the number of basic trained convolution filters *n_f*

Способ декодировки	Число фильтров *n_f* (%)					
	8	16	24	32	40	48
Nearest Neighbor	89,33	92,16	93,21	93,30	93,52	93,63
Bilinear	89,34	92,15	93,26	93,43	93,58	93,72
Nearest Neighbor + conv(C_{in};2;2;C_{out}=C_{in}/2)	84,74	92,04	93,15	93,24	93,43	93,51
Bilinear + conv(C_{in};2;2;C_{out}=C_{in}/2)	86,01	92,18	93,30	93,35	93,47	93,61
Nearest Neighbor + conv(C_{in};2;2;C_{out}=C_{in}/4)	79,73	91,63	93,09	93,26	93,51	93,66
Bilinear + conv(C_{in};2;2;C_{out}=C_{in}/4)	79,76	91,65	93,16	93,33	93,60	93,72
SubConv	70,27	90,23	92,30	92,80	93,21	93,48

Таблица 5

Оптимизация вычислительной сложности моделей по сравнению с моделями, использующими ближайшего соседа или билинейную интерполяцию в качестве способа декодировки сети

Table 5

Optimization of computational complexity of models, compared to models using nearest neighbor or bilinear interpolation as a method of network deconvolution

Способ декодировки	Число фильтров *n_f*					
	8	16	24	32	40	48
Nearest Neighbor / Bilinear	490 993	1 962 337	4 414 033	7 846 081	12 258 481	17 651 233
Nearest/ Bilinear + conv(C_{in};2;2;C_{out}=C_{in}/2)	–5 440	–21 760	–48 960	–87 040	–136 000	–195 840
Nearest / Bilinear + conv(C_{in};2;2;C_{out}=C_{in}/4)	–51 680	–206 720	–465 120	–826 880	–1 292 000	–1 860 480
SubConv	–73 440	–293 760	–660 960	–1 175 040	–1 836 000	–2 643 840

Таблица 6

Тестирование времени генерации масок сегментации
для одного 3D объема на CPU и GPU

Table 6

Testing the generation time of segmentation masks
for one 3D volume on CPU and GPU

Способ декодировки	Усредненное по инициализаторам время сегментации на устройстве в мин.					
	Число фильтров *n_f*					
	CPU			GPU		
	24	32	40	24	32	40
Nearest Neighbor	2,34	3,27	4,29	0,41	0,49	0,66
Bilinear	4,45	5,99	7,55	0,41	0,48	0,65
Nearest Neighbor + conv(C_{in};2;2;C_{out}=C_{in}/2)	2,37	3,15	4,18	0,44	0,53	0,72
Bilinear + conv(C_{in};2;2;C_{out}=C_{in}/2)	4,52	6,04	7,72	0,44	0,53	0,72
Nearest Neighbor + conv(C_{in};2;2;C_{out}=C_{in}/4)	2,27	2,81	4,11	0,43	0,50	0,66
Bilinear + conv(C_{in};2;2;C_{out}=C_{in}/4)	4,31	5,60	7,50	0,43	0,49	0,66
SubConv	2,28	2,82	4,16	0,37	0,42	0,56

Операция субпиксельной конволюции позволяет добиться значительного упрощения сложности модели, но при этом увеличивает сходимость моделей – потребуется провести большее количество обучающих эпох, чтобы достигнуть сопоставимого качества сегментации. При этом модель, использующая дискретизацию ближайшим соседом и обучаемой сверткой с сокращением количества выходных каналов в 4 раза, позволяет добиться лучшего качества классификации пикселей за то же время вывода предсказания.

Для моделей с $n\_f = 16$ и $n\_f = 24$ фильтрами больший прирост качества был достигнут при обучении декодировщика с билинейной интерполяцией и дополнительной сверткой с понижением каналов в 2 раза. Однако для решения оптимизационной задачи #A модели большей сложности, использующие только интерполяцию ближайшим соседом, позволяют добиться лучшего качества сегментации. Несмотря на то что интерполяция ближайшим соседом в некоторых случаях дает большую погрешность приближения данных, можно попробовать добиться улучшения качества сегментации в процессе дообучении этих моделей.

Так как все модели с $n\_f = \{24, 32, 40, 48\}$, не использующие субпиксельную конволюцию части сетевого декодера, являются сопоставимыми по качеству сегментации, для выбора оптимальных моделей следует учитывать время на настройку обучающих параметров и время на получение маски сегментации по классу гранулы для трех 3D объемов КТ-изображений.

Выводы

Реализация базовой архитектуры нейронной сети не всегда является лучшим решением при ее применении в системах с ограниченным временем обработки входного потока данных. На основе анализа моделей архитектуры UNet по пространству гиперпараметров, огра-

ниченных переменными количества обучаемых фильтров в операциях свертки, регуляризации их выходов и способа увеличения разрешения в части сетевого декодера, был осуществлен поиск моделей сегментации, удовлетворяющих ограничениям оптимизационных задач. Такой поиск позволил получить первичное представление об аппроксимирующей способности моделей, обладающих меньшей вычислительной сложностью.

После анализа моделей по способу декодровки признакового представления на каждом уровне сети UNet, для группы моделей с базовым числом фильтров $n_f \in [24, 32, 40, 48]$ и применением линейной интерполяции для дискретизации признакового представления изображения в сети было достигнуто сопоставимое качество сегментации на валидационной выборке. Модели отличались только временем генерации масок сегментации.

- Для решения оптимизационной задачи класса #A может быть выбрана модель с 32 базовыми фильтрами *n_f* в слоях свертки с обучаемым смещением, использующая в качестве метода повышения дискретизации интерполяцию ближайшим соседом и дополнительный сверточный слой, сокращающий количество каналов представления в 4 раза. Время сегментации трех 3D объемов на CPU составляет порядка 8,5 минуты, на GPY – 1,5 минуты. Такая модель обладает в 4,7 раза меньшим числом параметров, чем классическая архитектура UNet, и позволяет достигнуть качества сегментации на тестовой выборке 1-BCE = 93,3 %.
- Решением оптимизационной задачи класса #B является модель с 24 базовыми фильтрами *n_f* в слоях свертки с обучаемым смещением. В качестве способа декодировки может быть использована интерполяция ближайшим соседом без сокращения количества каналов. Стоит отметить, что это же решение в случае дообучения модели может быть применено для оптимизационной задачи #A, при этом последний блок декодировщика может быть заменен билинейной интерполяцией, так как сложность модели в 1,5 раза меньше, чем модели класса #A.

Поскольку оптимальная 2D модель будет использоваться для создания обучающей выборки для 3D модели, полученные изображения требуют дополнительного анализа и доразметки. Отобразив разницу интенсивностей пикселей между целевой и небинаризированной прогнозируемой маской сегментации, можно выявить, на каких областях изображения модель выдает ошибочный ответ классификации. Так, на рис. 3 для первого изображения погрешность составляет 7,89 %, для второго – 9,04 %. Качество модели снижается при прогнозировании границ по классу гранулы и при классификации сильно засвеченных гранул. С другой стороны, эти области являются самыми трудными для ручной разметки, поэтому автоматизация с помощью нейронной сети может значительно сократить время подготовки наборов изображений.

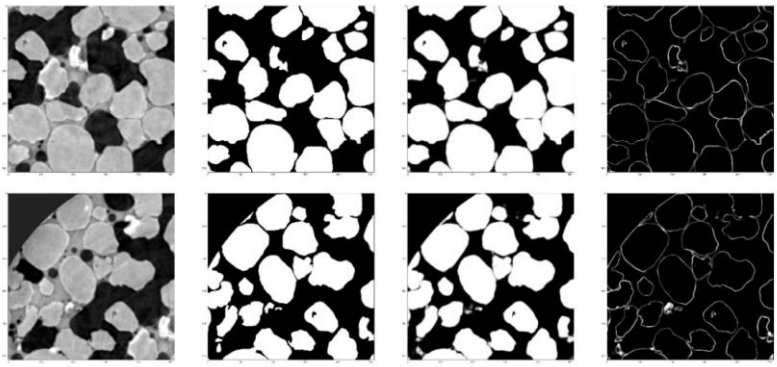

Рис. 3. Слева направо: изображение, таргетная маска сегментации по классу гранулы, сгенерированный вероятностный ответ модели, погрешность предсказания
Fig. 3. From left to right: Image, target segmentation mask by granule class, generated probabilistic response of the model, prediction error

Список литературы

1. **Chong Z. R., Yang S. H. B., Babu P., Linga P., Li X.-S.** Review of natural gas hydrates as an energy resource: prospects and challenges. *Applied Energy*, 2016, vol. 162, pp. 1633–1652. DOI 10.1016/j.apenergy.2014.12.061
2. **Makogon Y. F., Omelchenko R. Y.** Commercial gas production from Messoyakha deposit in hydrate conditions. *Journal of Natural Gas Science and Engineering*, 2013, vol. 11, pp. 1–6. DOI 10.1016/j.jngse.2012.08.002
3. **Кадыров Р. И.** Рентгеновская компьютерная томография в геологии: Учеб.-метод. пособие. Казань, Казан. федеральный ун-т, 2020. 37 с.
4. **Дробчик А. Н., Дугаров Г. А., Дучков А. А., Купер К. Э.** Акустические измерения и рентгеновская томография песчаных образцов, содержащих гидрат ксенона // Геофизические технологии. 2019. № 4. С. 17–23. DOI 10.18303/2619-1563-2019-4-17
5. **Nikitin V. V., Dugarov G. A., Duchkov A. A., Fokin M. I., Drobchik A. N., Shevchenko P. D., De Carlo F., Mokso R.** Dynamic in-situ imaging of methane hydrate formation and self-preservation in porous media. *Marine and Petroleum Geology*, 2020, vol. 115. DOI 10.1016/j.marpetgeo.2020.104234
6. **Ronneberger O., Fischer P., Brox T.** U-Net: Convolutional networks for biomedical image segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Lecture Notes in Computer Science*, 2015, vol. 9351, pp. 234–241. DOI 10.1007/978-3-319-24574-4_28
7. **Luo W., Li Y., Urtasun R., Zemel R.** Understanding the effective receptive eld in deep convolutional neural networks. In: Advances in Neural Information Processing Systems. Barcelona, 2016, pp. 4898–4906.
8. **Yi-de M., Qing L., Zhi-Bai Q.** Automated image segmentation using improved PCNN model based on cross-entropy. *Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing*, 2004, pp. 743–746. DOI 10.1109/ISIMP.2004.1434171
9. **Rahman M. A., Wang Y.** Optimizing intersection-over-union in deep neural net-works for image segmentation. *International symposium on visual computing*, 2016, pp. 234–244. DOI 10.1007/978-3-319-50835-1_22
10. **Kingma D., Ba J.** Adam: A Method for Stochastic Optimization. In: Proceedings of the 3rd International Conference on Learning Representations, 2014.
11. **Chollet F.** Deep Learning with Python Data representations for neural networks. Shelter Island, NY, Manning Publications, 2018, pp. 31–38.
12. **Ioffe S., Szegedy C.** Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Proceedings of the 32nd International Conference on Machine Learning (PMLR)*, 2015, vol. 37, pp. 448–456.
13. **Shi W., Caballero J., Huszar F., Totz J., Aitken A.P., Bishop R., Rueckert D., Wang Z.** Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1874–1883. DOI 10.1109/cvpr.2016.207

References

1. **Chong Z. R., Yang S. H. B., Babu P., Linga P., Li X.-S.** Review of natural gas hydrates as an energy resource: prospects and challenges. *Applied Energy*, 2016, vol. 162, pp. 1633–1652. DOI 10.1016/j.apenergy.2014.12.061
2. **Makogon Y. F., Omelchenko R. Y.** Commercial gas production from Messoyakha deposit in hydrate conditions. *Journal of Natural Gas Science and Engineering*, 2013, vol. 11, pp. 1–6. DOI 10.1016/j.jngse.2012.08.002

3. **Kadyrov R. I.** Rentgenovskaya kompyuternaya tomografiya v geologii. Uchebno-metodicheskoe posobie. Kazan, Kazan Federal University, 2020, 37 p. (in Russ.)
4. **Drobchik A. N., Dugarov G. A., Duchkov A. A., Kuper K. E.** Acoustic measurements and X-ray tomography of sand samples containing xenon hydrate. *Russian Journal of Geophysical Technologies*, 2019, no. 4, pp. 17–23. (in Russ.) DOI 10.18303/2619-1563-2019-4-17
5. **Nikitin V. V., Dugarov G. A., Duchkov A. A., Fokin M. I., Drobchik A. N., Shevchenko P. D., De Carlo F., Mokso R.** Dynamic in-situ imaging of methane hydrate formation and self-preservation in porous media. *Marine and Petroleum Geology*, 2020, vol. 115. DOI 10.1016/j.marpetgeo.2020.104234
6. **Ronneberger O., Fischer P., Brox T.** U-Net: Convolutional networks for biomedical image segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Lecture Notes in Computer Science*, 2015, vol. 9351, pp. 234–241. DOI 10.1007/978-3-319-24574-4_28
7. **Luo W., Li Y., Urtasun R., Zemel R.** Understanding the effective receptive eld in deep convolutional neural networks. In: Advances in Neural Information Processing Systems. Barcelona, 2016, pp. 4898–4906.
8. **Yi-de M., Qing L., Zhi-Bai Q.** Automated image segmentation using improved PCNN model based on cross-entropy. *Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing*, 2004, pp. 743–746. DOI 10.1109/ISIMP.2004.1434171
9. **Rahman M. A., Wang Y.** Optimizing intersection-over-union in deep neural net-works for image segmentation. *International symposium on visual computing*, 2016, pp. 234–244. DOI 10.1007/978-3-319-50835-1_22
10. **Kingma D., Ba J.** Adam: A Method for Stochastic Optimization. In: Proceedings of the 3rd International Conference on Learning Representations, 2014.
11. **Chollet F.** Deep Learning with Python Data representations for neural networks. Shelter Island, NY, Manning Publications, 2018, pp. 31–38.
12. **Ioffe S., Szegedy C.** Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Proceedings of the 32nd International Conference on Machine Learning (PMLR)*, 2015, vol. 37, pp. 448–456.
13. **Shi W., Caballero J., Huszar F., Totz J., Aitken A.P., Bishop R., Rueckert D., Wang Z.** Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1874–1883. DOI 10.1109/cvpr.2016.207

Информация об авторах

Татьяна Олеговна Колесник, студентка магистратуры
Антон Альбертович Дучков, заведующий лабораторией

Information about the Authors

Tatyana O. Kolesnik, Master's Student
Anton A. Duchkov, Head of the Laboratory

Статья поступила в редакцию 11.12.2021;
одобрена после рецензирования 01.02.2022; *принята к публикации 01.02.2022*
The article was submitted 11.12.2021;
approved after reviewing 01.02.2022; *accepted for publication 01.02.2022*