

Отчет об исследовании

Сегментация сейсмических изображений для
обнаружения солей с использованием CNN.

Байдаченко Валерия Алексеевна

Оглавление

Постановка задачи.....	2
Исходные данные	2
Метрики качества	3
Анализ	4
Предварительный анализ данных	5
Описание решения	5
Обучение моделей.....	6
Результаты	6
Визуализация результатов.....	7
Выводы.....	8
Список использованных источников.....	9

Постановка задачи

Сейсмические данные собираются с использованием сейсморазведки. В ходе сейсмической съемки методом отраженных волн сейсмические волны создаются на поверхности или вблизи нее при помощи сейсмического источника – это может быть динамит, молот, вибратор, пневматический источник, гидропушка или иное устройство, способное передавать энергию вглубь земли. Сейсмические волны распространяются от источника и проходят сквозь слои геологического разреза. На границах между различными типами пород происходит изменение физических свойств, в частности, плотности и скорости распространения сейсмических волн и их произведения, сейсмического импеданса.

Некоторая часть сейсмической энергии отражается от геологических границ и достигает поверхности. На поверхности отраженные волны регистрируются сейсмоприемниками. Они представляют собой электромеханические устройства - микрофоны, геофоны, гидрофоны или акселерометры (датчики ускорения), преобразующие энергию отраженных волн в электрические сигналы.

Сигналы записываются и обрабатываются на компьютере для получения изображений недр, производится их интерпретация для получения информации, необходимой для поисково-разведочных работ, разработки месторождений и добычи нефти и газа.

Существует несколько областей в мире, где есть огромное количество соли в недрах. Одна из проблем сейсмического изображения состоит в том, чтобы идентифицировать часть недр, которая является солью. Плотность соли обычно составляет $2,14 \text{ г / см}^3$, что ниже, чем у большинства окружающих ее пород. Сейсмическая скорость соли составляет $4,5 \text{ км / с}$, что обычно выше, чем у окружающих ее пород. Эта разница создает резкое отражение на границе раздела солевых осадков. Обычно соль - это аморфная порода без особой внутренней структуры. Это означает, что внутри соли, как правило, не так много отражающей способности, если только внутри нее нет отложений. Необычно высокая сейсмическая скорость соли может создать проблемы с сейсмической визуализацией. Соляные отложения важны для углеводородной промышленности, так как они обычно образуют хорошие нефтяные ловушки.

Но, к сожалению, чрезвычайно трудной задачей, которая следует за этапом предварительной обработки, является интерпретация сейсмических изображений. Точная интерпретация сейсмических изображений требует многолетнего опыта работы в области геофизики. Ошибки дорогостоящие: неправильные интерпретации могут стоить миллионы долларов нефтяным и газовым компаниям, если они будут копать в земле с ложными прогнозами углеводородов. Кроме того, существует множество рисков, связанных с бурением. Следовательно, очень важный вопрос в этом направлении заключается в том, можем ли мы использовать возможности компьютеров, чтобы помочь геофизикам точно составить карту областей на изображениях, которые соответствуют присутствию солей(углеводородов).

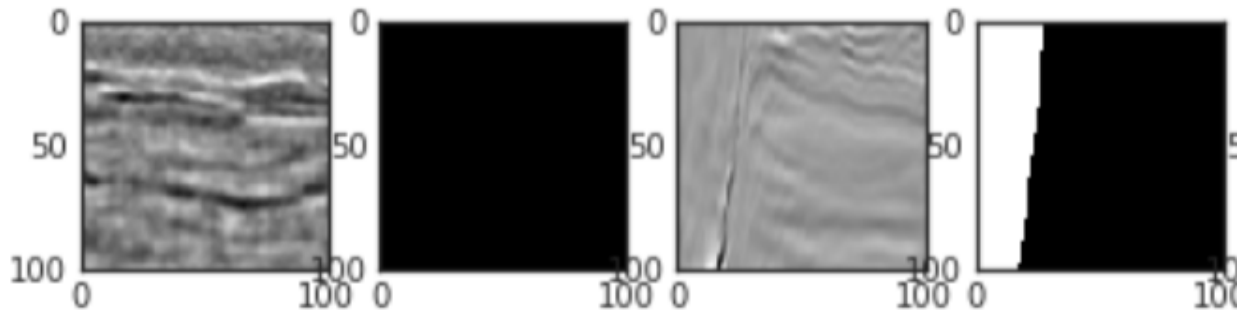
Цель данного проекта – сегментация регионов, содержащих соль. Работа проводится в рамках соревнования на Kaggle <https://www.kaggle.com/c/tgs-salt-identification-challenge/overview>

Исходные данные

Данные представляют собой набор изображений, выбранных в различных местах случайным образом в недрах. Изображения имеют размер 101×101 пикселей, и каждый пиксель классифицируется как соль или осадок. В дополнение к сейсмическим изображениям, есть данные о глубине отображаемого местоположения, предоставляется

для каждого изображения. Обучающий набор содержит 4000 сейсмических изображений, а тестовый 18000 изображений.

Пример изображений и масок



Метрики качества

Это соревнование оценивается по средней точности при различных порогах пересечения по объединению (IoU). IoU предлагаемого набора пикселей объекта и набора пикселей истинного объекта вычисляется как:

$$IoU(A, B) = \frac{A \cap B}{A \cup B}.$$

Метрика охватывает диапазон порогов IoU, в каждой точке рассчитывается среднее значение точности. Пороговые значения находятся в диапазоне от 0,5 до 0,95 с размером шага 0,05: (0,5, 0,55, 0,6, 0,65, 0,7, 0,75, 0,8, 0,85, 0,9, 0,95). Другими словами, при пороге 0,5 предсказанный объект считается «успехом», если его IoU больше 0,5.

При каждом пороговом значении t точность вычисляется на основе количества истинных положительных результатов (TP), ложных отрицательных значений (FN) и ложных положительных результатов (FP), полученных в результате сравнения прогнозируемого объекта со всеми истинными объектами:

$$\frac{TP(t)}{TP(t) + FP(t) + FN(t)}.$$

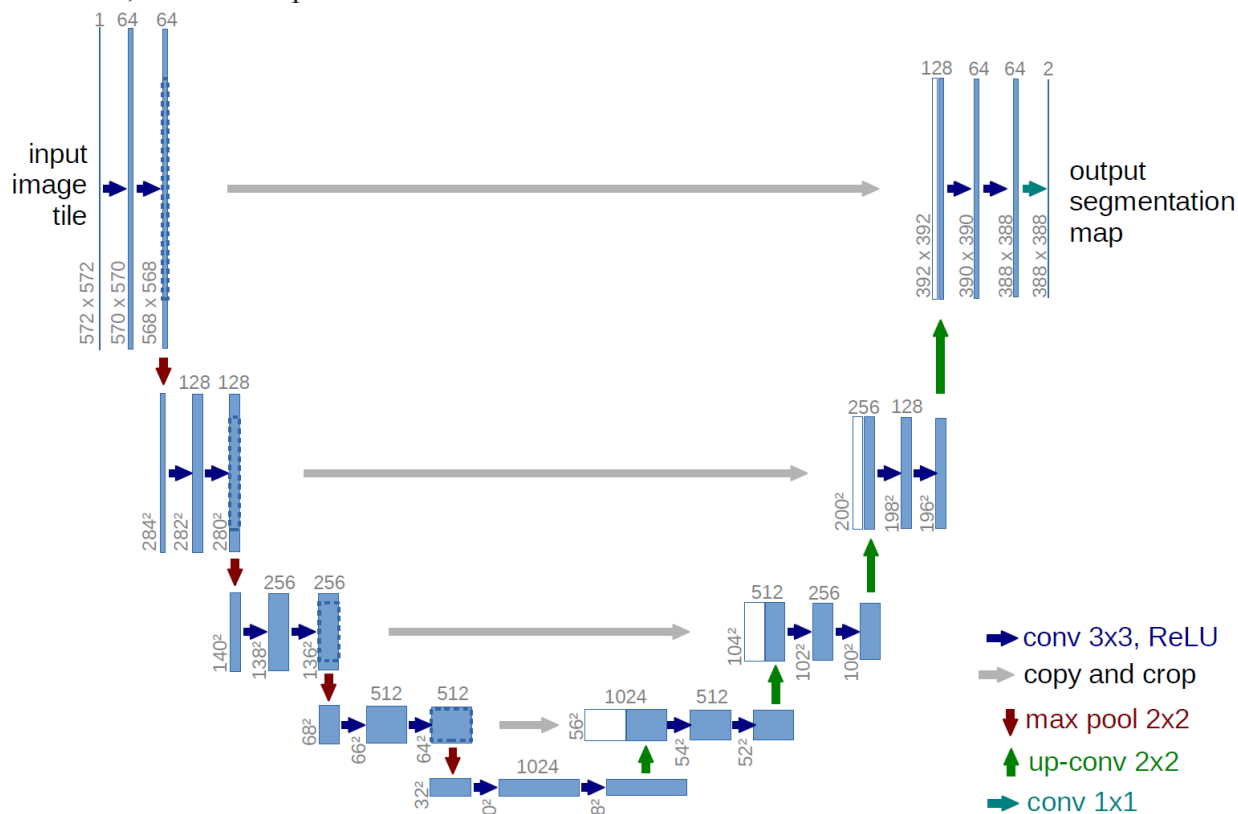
Средняя точность одиночного изображения затем рассчитывается как среднее из вышеуказанных значений точности при каждом пороге IoU:

$$\frac{1}{|thresholds|} \sum_t \frac{TP(t)}{TP(t) + FP(t) + FN(t)}.$$

Наконец, оценка, возвращаемая метрикой соревнования, представляет собой среднее значение, полученное по индивидуальной средней точности каждого изображения в тестовом наборе данных.

Анализ

Для проведения исследований я взяла хорошо изученную U-net, это отличная сеть для сегментации картинок. Сеть очень распространена в таких соревнованиях и есть много описаний, тонкости применения и т.д.



U-Net — это **свёрточная нейронная сеть**, которая была создана в 2015 году для сегментации биомедицинских изображений в отделении Computer Science **Фрайбургского университета**. Архитектура сети представляет собой полносвязную свёрточную сеть, модифицированную так, чтобы она могла работать с меньшим количеством примеров (обучающих образов) и делала более точную сегментацию.

Сеть содержит сжимающий путь (слева) и расширяющий путь (справа), поэтому архитектура похожа на букву U, что и отражено в названии. На каждом шаге мы удваиваем количество каналов признаков.

Сжимающий путь похож на типичную свёрточную сеть, он содержит два подряд свёрточных слоя 3x3, после которых идет слой ReLU и пулинг с функцией максимума 2x2 с шагом 2.

Каждый шаг расширяющего пути содержит слой, обратный пулингу, который расширяет карту признаков, после которого следует свертка 2x2, которая уменьшает количество каналов признаков. После идет конкатенация с соответствующим образом обрезанной картой признаков из сжимающего пути и две свертки 3x3, после каждой из которых идет ReLU. Обрезка нужна из-за того, что мы теряем пограничные пиксели в каждой свертке. На последнем слое свертка 1x1 используется для приведения каждого 64-компонентного вектора признаков до требуемого количества классов.

Всего сеть имеет 23 свёрточных слоя.

Предварительный анализ данных

Мной был сделан расчет площади солевого покрытия и на основе этого признака было сделано деление на классы с целью дальнейшего использования этого признака для стратифицированного деления на обучающий и валидационный наборы данных. Примерно 40% изображений – это изображения без соли.

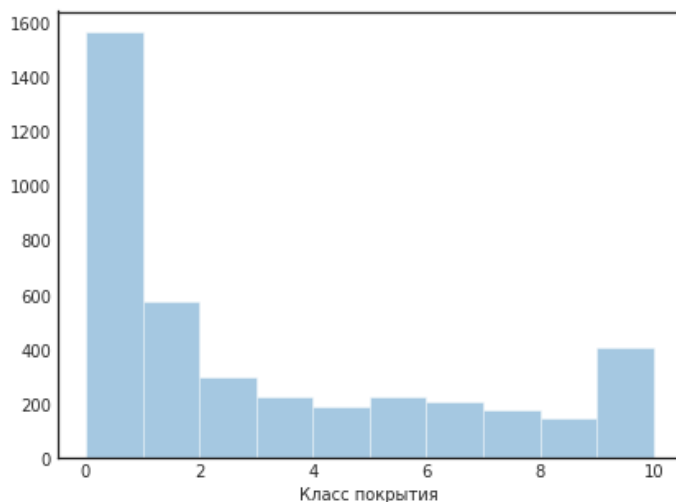
```
train_df['coverage_class'].value_counts(normalize=True)
```

```
:
```

0	0.39050
1	0.14400
2	0.07400
10	0.06125
5	0.05625
3	0.05600
6	0.05125
4	0.04650
7	0.04350
9	0.03975
8	0.03700

Name: coverage_class, dtype: float64

Смотрим, как выглядит гистограмма по этому признаку.



Описание решения

Работа с данными

Для деления данных на тренировочный и валидационный наборы я использовала созданный признак 'coverage_class'. В тренировочный датасет вошли 3200 изображений, в валидационный 800.

Так как данных было не так уж и много, то я использовала аугментацию данных. Мы можем сделать только отображение по горизонтали, потому что на сейсмических изображениях (сейсмограммах, разрезах и др.) имеют дело с временами прихода волн, отраженных от границ, а не с глубинами границ. Для сред, близких к горизонтально-слоистым, координаты времен и глубин связаны простым переводным коэффициентом – скоростью. Обычно скорость постепенно растет с глубиной, что соответствует плавному изменению коэффициента при переходе от глубинной к временной вертикальной оси. Часто бывает, что скорости точно неизвестны и поэтому изображения оставляют по вертикали во временном масштабе.

Архитектура сетей

Во время работы над данной задачей я пробовала разные архитектуры сетей

Первый эксперимент.

U-net без тюнингов. Перед подачей изображения на вход делаю resize (128,128).
Обучение проходит 60 эпох, butch_size = 32.

Второй эксперимент.

Использую библиотеку segmentation model. «Предобученная» модель ResNet34 с весами ImageNet.

https://github.com/qubvel/segmentation_models/tree/master/segmentation_models

Перед подачей изображения на вход делаю resize (128,128).

Обучение проходит 100 эпох, batch_size = 16.

Третий эксперимент.

Тюнингованная U-Net

Использую dropout, транспонированную свертку Conv2DTranspose, batchnormalization, residual block. Делаю аугментацию изображений. И делаю стратифицированное деление на обучающую и валидационную выборку.

Обучение проходит 100 эпох, batch_size = 16.

Обучение моделей

Обучение моделей происходило на Kaggle. На Kaggle есть ограничения по времени 6 часов, по оперативной памяти 13Гб, поэтому я не делала обучение более 100 эпох, batch_size тоже был выбран исходя их данных ограничений.

Результаты

Результаты исследования представлены в таблице.

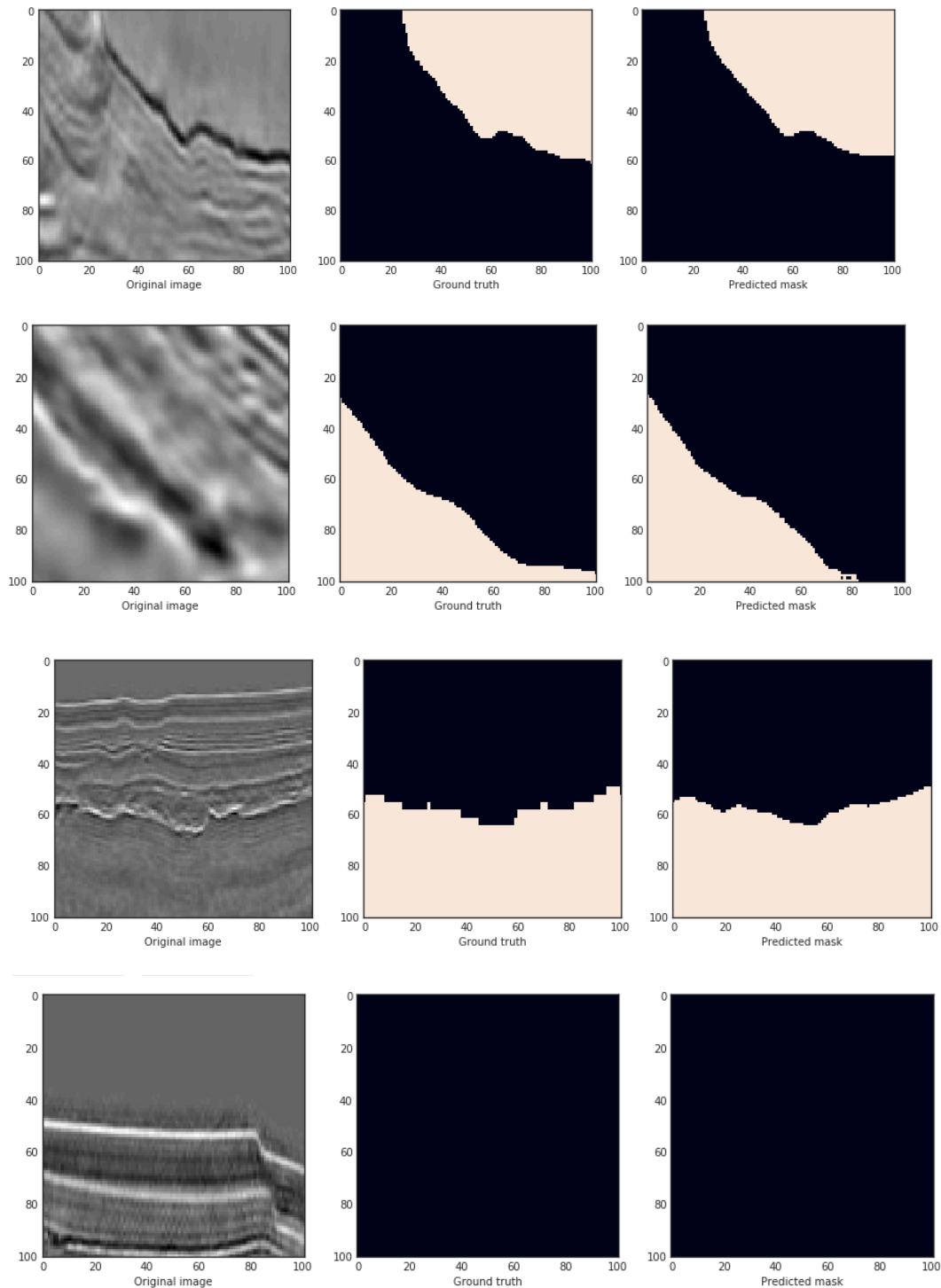
Model	Private score	Public score
U-net	0,67665	0,63864
U-net (ResNet34)	0,76995	0,74211
U-net (ResBlock, Aug, Strat)	0,8341	0,80866

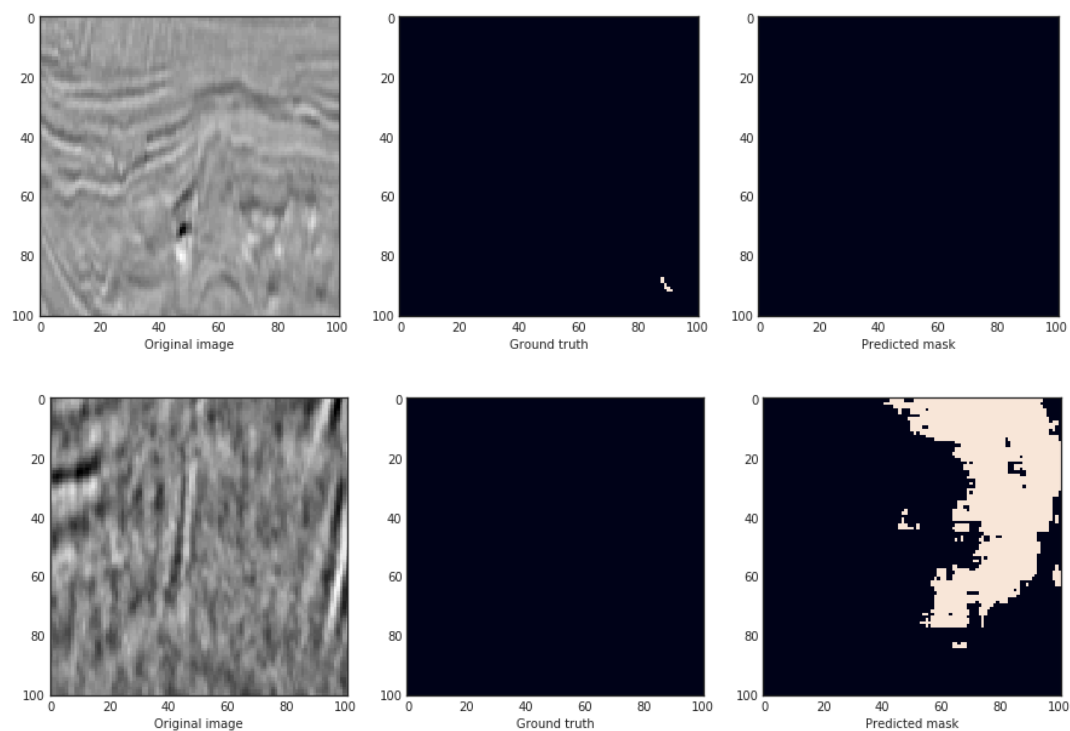
Лучший результат получился у тюнингованной модели, что вполне логично.

Визуализация результатов

Ниже представлены предсказания модели U-net (ResBlock, Aug, Strat) на валидационной выборке.

Модель обрабатывает достаточно неплохо.





Выводы

Меня устроил результат моей работы над данной задачей. Я смогла добиться неплохих результатов, получила практический опыт создания нейронных сетей, подбора параметров сетей.

Конечно, до результатов победителей данного конкурса мне не удалось дотянуться, но можно продолжать работать в данном направлении. Можно усложнить сеть, добавить дополнительные блоки, попробовать использовать «кастомные» функции потерь.

Список использованных источников

1. Воскресенский Ю.Н. Построение сейсмических изображений. Учебное пособие для вузов. – М.: РГУ нефти и газа, 2006, 116 с.
http://lerv.deg.gubkin.ru/file.php?file=../../1/dfwikidata/Voskresenskij.JU.N.Postroenie.sejsmicheskij.izobrazhenij.%28M.,RGUNG%29%282006%29%28T%29_GsPs_.pdf
2. Kaggle TGS Salt Identification Challenge <https://www.kaggle.com/c/tgs-salt-identification-challenge/overview>
3. U-Net: Convolutional Networks for Biomedical Image Segmentation
<https://arxiv.org/pdf/1505.04597.pdf>
4. Deep Residual Learning for Image Recognition <https://arxiv.org/pdf/1512.03385.pdf>
5. Segmentation model https://github.com/qubvel/segmentation_models