



ITMO UNIVERSITY

Saint Petersburg, Russia

Распознавание изображений сейсмических срезов с помощью AutoML.

Как автоматизировать процесс обнаружения нефтяных резервуаров с помощью автоматического подбора архитектуры нейронной сети

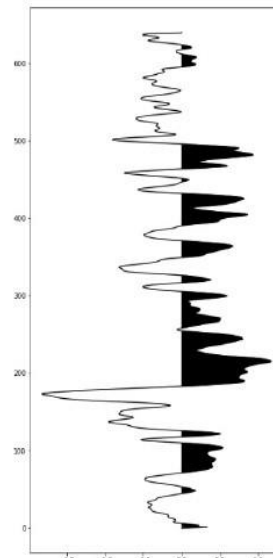
Ревин Илья

инженер Национального Центра Когнитивных Разработок

Санкт-Петербург - 2020

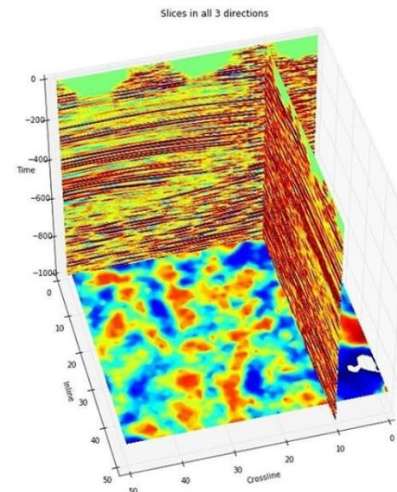
Решаемая задача – Анализ данных сейсмической разведки

- Анализ данных сейсмической разведки как правило сводится к анализу записей полученных с помощью сейсмодатчиков. Такие записи называются сейсмотрассами.
- Во ходе процесса предварительной обработки полученных сейсмотрасс мы получаем также сейсмическими разрезами (общепринятые названия inline, crossline, xline). Эти разрезы вместе образуют информационную модель месторождения с точки зрения сейсмической разведки. Такая модель называется сейсмическим кубом.
- Анализируя эти разрезы, мы можем получить представление о границах между геологическими элементами и определить местоположение интересующих нас элементов.



Sample of recorded seismic trace

Пример записи сейсмотрассы

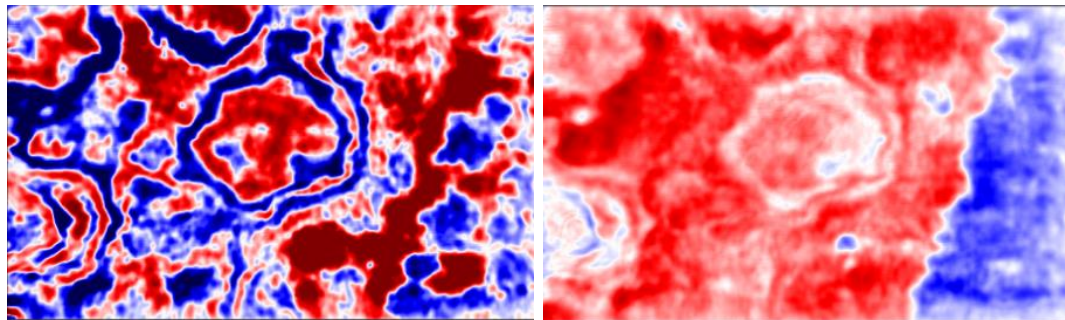
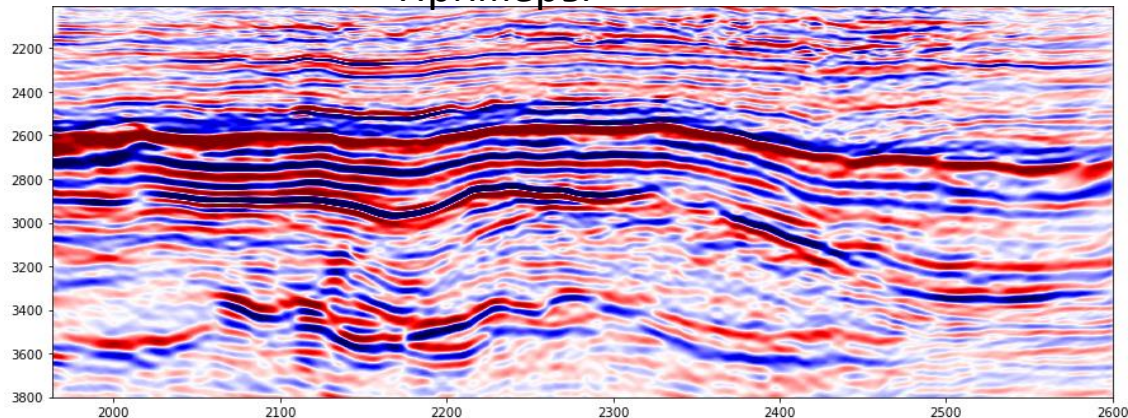


Пример сейсмического куба

Описание

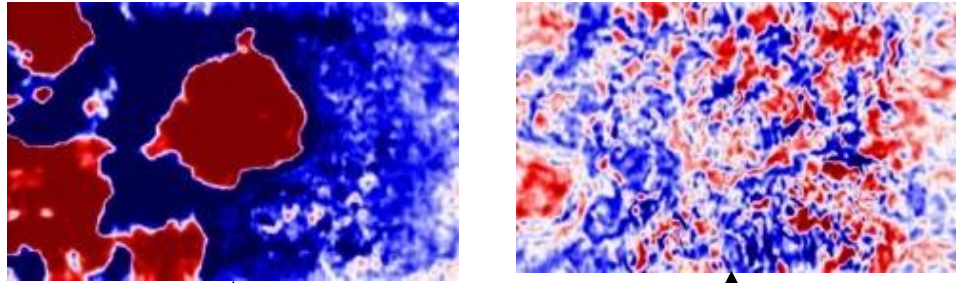
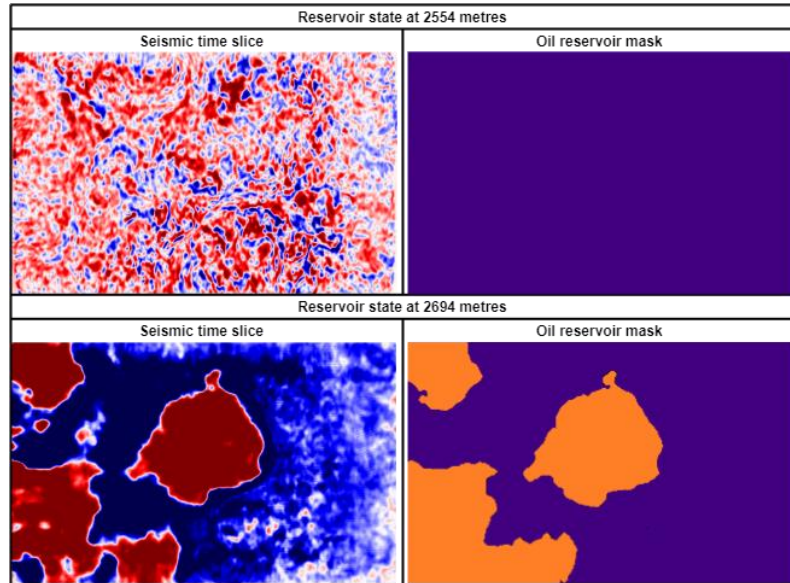
- **Анализ продольных срезов** сейсмического куба (ilines). Целью анализа срезов такого типа является нахождение границ между геологическими элементами. Таким образом можно установить **«мощность»** (например количество запасов нефти) геологического элемента.
- **Анализ поперечных срезов** сейсмического куба (crosslines). В данном анализе мы пытаемся установить локацию и геометрическую форму геологического элемента по всей глубине исследуемого месторождения.

Примеры



Постановка задач анализа сейсмических срезов с точки зрения Machine learning

- Детектирование и распознавание формы нефтяного резервуара по всей глубине месторождения.
- Получение информации о наличии потенциальных нефтяных резервуаров на сейсмических срезах без участия эксперта.



Семантической
сегментации

Бинарной
классификации
сейсмических срезов

Поиск оптимальной структуры нейронной сети и ее применение для задач

Постановка задачи бинарной классификации сейсмических срезов с математической точки зрения



Мы представляем сейсмические временные срезы как комбинацию двух множеств $Z_{crossline}$ и Z_{xline}^T . Они содержат в себе сеймотрассы по всей глубине сейсмического куба и так же в поперечном сечении. С математической точки зрения, такое представление можно сформулировать следующим образом:

$$Z_{crossline}, Z_{xline}^T = [(z_1), ..., (z_T)]$$

Где каждый вектор z содержит сеймотрассы для конкретной точки пространства сейсмического куба. Обозначим множество пар "объект, класс" $X \times Y$ как вероятностное пространство с неизвестной вероятностной мерой P , тогда:

$$P = P(S_i|y_i), S_i \in X, y_i \in Y$$

Существует такой тренировочный набор данных $X^m \in X \times Y$ порождаемый этим вероятностным пространством, который можно записать как:

$$X^m = [(S_1, y_1), ..., (S_m, y_m)]$$

Наша задача состоит в создании такого алгоритма, который для любого объекта $S_j \in X$, вычисляет вероятность P и строит функцию f , такую что:

$$f : X \rightarrow Y = \arg \max_i P(S_j|y_i)$$

Предварительная обработка данных сейсмической разведки

#Загрузка данных из файла

```
f = segyio.open(segyfile, ignore_geometry=True)1
```

все сейсмотрассы

```
traces = segyio.collect(f.trace)[:]
```

```
traces_count, traces_len = traces.shape
```

количество уникальных координат

```
ilines_count = np.unique(iline)
```

```
xlines_count = np.unique(xline)
```

границы месторождения

```
min_iline, max_iline = min(ilines_count), max(ilines_count)
```

```
min_xline, max_xline = min(xlines_count), max(xlines_count)
```

шаг пространственной сетки

```
grid_step = samples[1] - samples[0]
```

```
dilines = min(np.unique(np.diff(ilines_count)))
```

```
dxlines = min(np.unique(np.diff(xlines_count)))
```

Создание пространственной сетки

```
ilines = np.arange(min_iline, max_iline + dilines, dilines)
```

```
xlines = np.arange(min_xline, max_xline + dxlines, dxlines)
```

```
num_ilines, num_xlines = ilines.size, xlines.size
```

```
ilines_grid, xlines_grid =
```

```
np.meshgrid(np.arange(num_ilines),  
             np.arange(num_xlines),  
             indexing='ij')
```

индексирование сейсмотрасс

```
traces_indeces = np.full((num_ilines, num_xlines),  
                           np.nan)
```

```
idx_il = (iline - min_iline)// dilines
```

```
idx_xl = (xline - min_xline) // dxlines
```

```
traces_indeces[idx_il, idx_xl] = np.arange(traces_count)
```

```
exist_traces = np.logical_not(np.isnan(traces_indeces))
```

```
print('# traces doesnt exist:
```

```
{}'.format(np.sum(~exist_traces)))
```

создание сейсмического куба

```
cube = np.zeros((num_ilines, num_xlines, traces_len))
```

```
cube[ilines_grid.ravel()][exist_traces.ravel()],
```

```
      xlines_grid.ravel()][exist_traces.ravel()]] = traces
```

сохранение изображений crossline срезов по всей глубине месторождения (449 метров)

```
for i in range(0,449,1):
```

```
    fig, axs = plt.subplots(1, 1, figsize=(6, 6))
```

```
    axs.get_xaxis().set_visible(False)
```

```
    axs.get_yaxis().set_visible(False)
```

```
    axs.imshow(cube_small[...,i], cmap='seismic',
```

```
               vmin=-4, vmax=4, extent=(xline[0], xline[-1], iline[-1],  
               iline[0]))
```

```
    plt.savefig(r'./image_jpg/'+str(i)+'.png', dpi=300)
```

¹ URL: https://github.com/ITMO-NSS-team/oil-field-modelling/blob/master/seismic_analysis/reservoir_detection_with_cnn.ipynb

```
from skimage.io import imread, imshow, concatenate_images # Библиотека для работы с изображениями
from skimage.transform import resize # Библиотека для работы с изображениями
# Библиотека для работы с изображениями
ids = next(os.walk(".\images"))2
# Создаем пустые массивы для срезов и масок
X = np.zeros((len(ids), im_height, im_width, 1), dtype=np.float32)
y = np.zeros((len(ids), im_height, im_width, 1), dtype=np.float32)

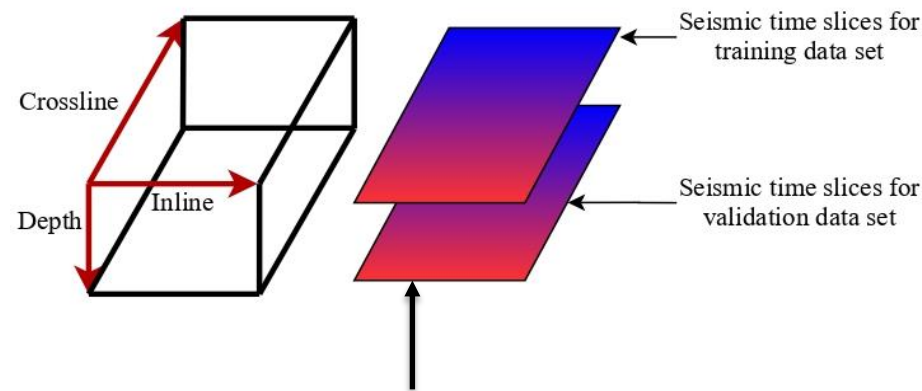
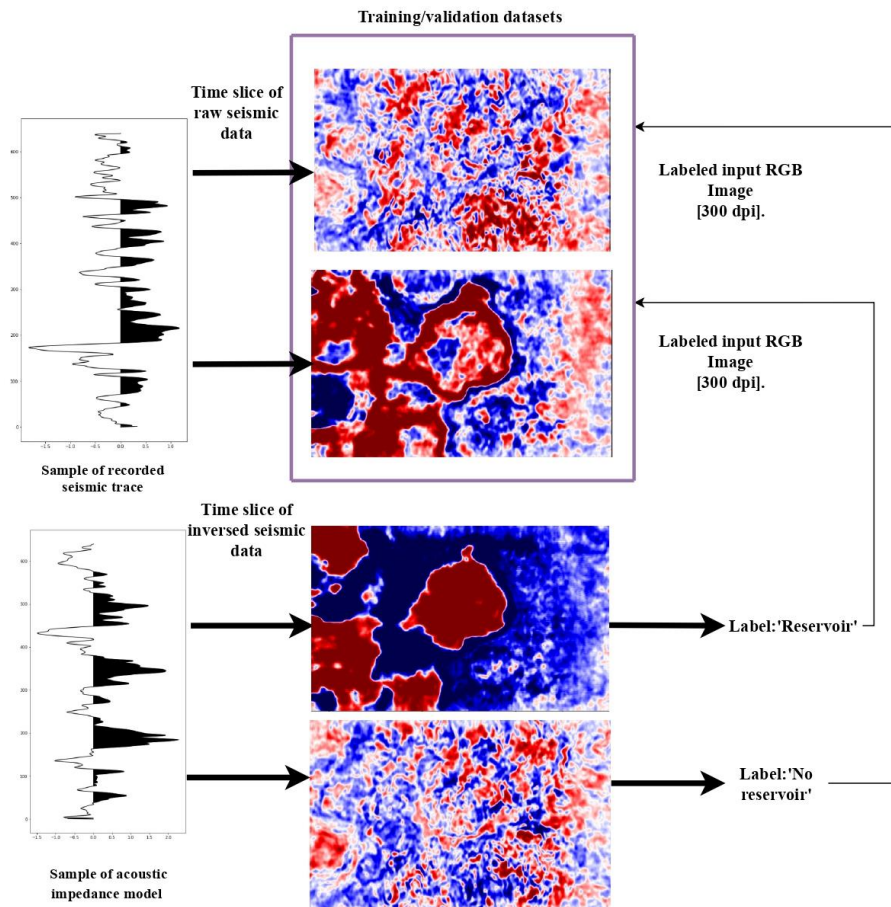
# Загрузка срезов и масок
for n, id_ in enumerate(ids):
    img = load_img(r"C:\Users\user\Desktop\seismic_OR\archive\images/"+id_, grayscale=True)
    x_img = img_to_array(img)
    x_img = resize(x_img, (im_height, im_width, 1), mode = 'constant', preserve_range = True)
    mask = img_to_array(load_img(r"C:\Users\user\Desktop\seismic_OR\archive\mask/"+id_,
    grayscale=True))
    mask = resize(mask, (im_height, im_width, 1), mode = 'constant', preserve_range = True)
    # Сохранение изображений в формате np.array
    X[n] = x_img/255.0
    y[n] = mask/255.0

# Дополнение данных
# Перевернутые образцы
X = np.append(X_train, [np.flipud(x) for x in X], axis=0)
y = np.append(y, [np.flipud(x) for x in y], axis=0)
# Повернутые на 90 градусов вокруг своей оси
X = np.append(X, [np.flip(x, axis = (0,1)) for x in X], axis=0)
y = np.append(y, [np.flip(x, axis = (0,1)) for x in y], axis=0)
```

Поскольку, в силу специфики прикладной области, в исходной тренировочной выборке оказалось всего лишь 450 изображений сейсмических срезов, была применена техника искусственного дополнения существующих изображений их вертикально и горизонтально отраженными копиями.

² URL: https://github.com/ITMO-NSS-team/oil-field-modelling/blob/master/seismic_analysis/oil_reservoir_semantic_segmentation.ipynb

Генерация тренировочных и валидационных наборов для AutoML моделей

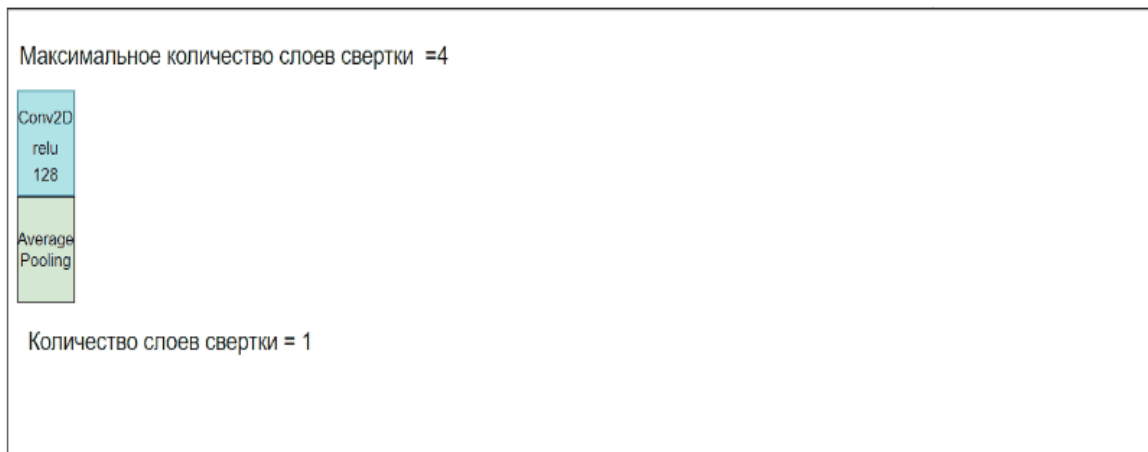


Генерация обучающих и валидационных наборов данных на основе изображений сейсмических срезов

Ручная разметка «сырых» сейсмических срезов с помощью инверсионного преобразования. «Преобразованные» сейсмические срезы делятся на 2 класса ("резервуар" и "отсутствие резервуара"). Полученный класс присваивается «сырому» сейсмическому срезу

Выбор моделей машинного обучения для решаемых задач

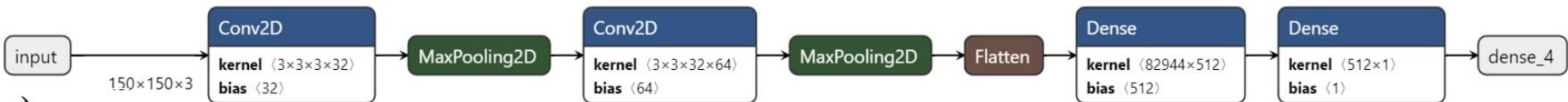
Мы рассматриваем сейсмические срезы как 2D-изображения к которым будем применять нейросети различных архитектур. Ручной выбор архитектуры сети требует как экспертных знаний в прикладной области, так и в области нейронных сетей. Целесообразным является использование алгоритма **NAS (Neural architecture search)** для поиска оптимальной архитектуры нейронной сети. Для того чтобы найти оптимальную архитектуру нейронной сверточной сети для задачи **бинарной классификации сейсмических срезов**, без непосредственного участия эксперта, было использовано расширение фреймворка **Fedot³** - эволюционный инструмент NAS.



В качестве модели для решения задачи семантической сегментации мы использовали модель с архитектурой **U-net**. Модель компилируется с помощью Adam optimizer, batch size = 32. В качестве функции потерь используется «binary cross-entropy loss function» поскольку существует только два состояния (нефтяной резервуар и отсутствие нефтяного резервуара).

³ URL: <https://github.com/ITMO-NSS-team/nas-fedot>

Сравнение результатов эволюционной оптимизации и ручного подбора архитектур сетей



Архитектуры глубоких моделей, использованных в экспериментальных исследованиях: а) базовая модель CNN, идентифицированная экспертом; б) модель CNN, полученная NAS

Как видно из рисунка, модель, полученная с использованием архитектуры NAS, имеет два дополнительных слоя свертки и два слоя пуллинга, а также «dropout» слой. Добавление отсеивающего слоя является регуляризацией модели и помогает решить проблему переобучения в процессе обучения.

Сравнение метрик качества для полученных архитектур

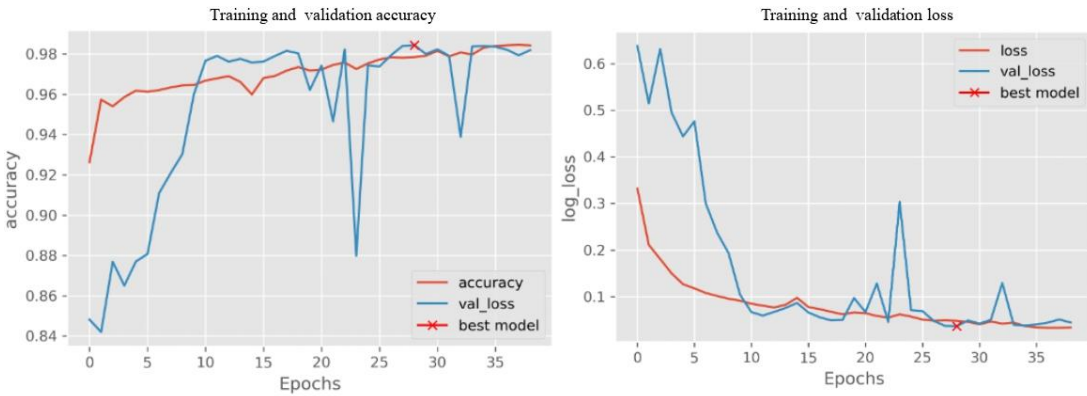
Model	Train		Validation				
	Accuracy	ROC AUC	Accuracy	ROC AUC	Precision	Recall	F1
CNN baseline	0.935	0.803	0.858	0.876	0.850	0.930	0.888
CNN NAS	0.972	0.907	0.954	0.953	0.907	0.907	0.907

Графики метрики качества и функции потерь



Изменение метрик для модели CNN

График слева показывает изменение значения метрики точности для обучающих и валидационных наборов данных. График справа показывает изменение значения функции потерь.



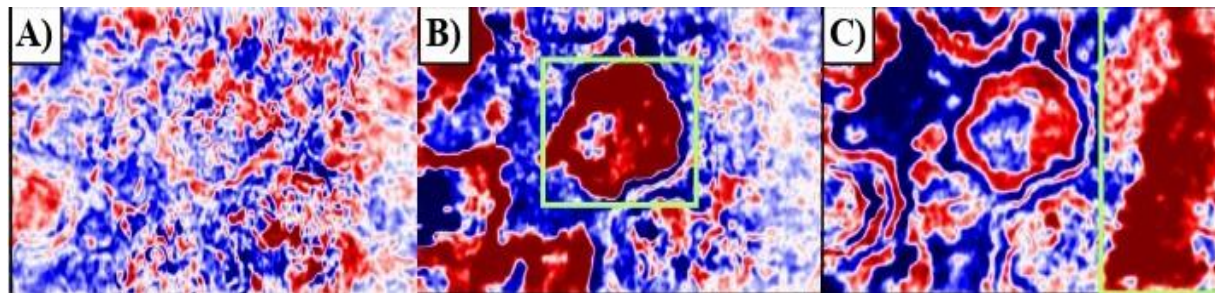
Изменение метрик для модели U-Net.

Красная линия на графике показывает изменение значения функции потерь для обучающего набора данных. Синяя линия на графике показывает изменение значения функции потерь для валидационного набора. Красный Крест – эпоха, в которую было достигнуто минимальное значение функции потерь для обучающих и валидационных наборов данных.

Анализ предсказаний полученных с помощью выбранных моделей машинного обучения

Примеры сейсмических срезов предсказанных с помощью модели NAS

- А) Пример среза с классом "без резервуара";
 - В) Пример правильного предсказанного моделью среза с классом "резервуар"
 - С) Пример неправильного предсказанного моделью среза с классом "резервуар"
- Зеленый квадрат - расположение нефтяного резервуара.



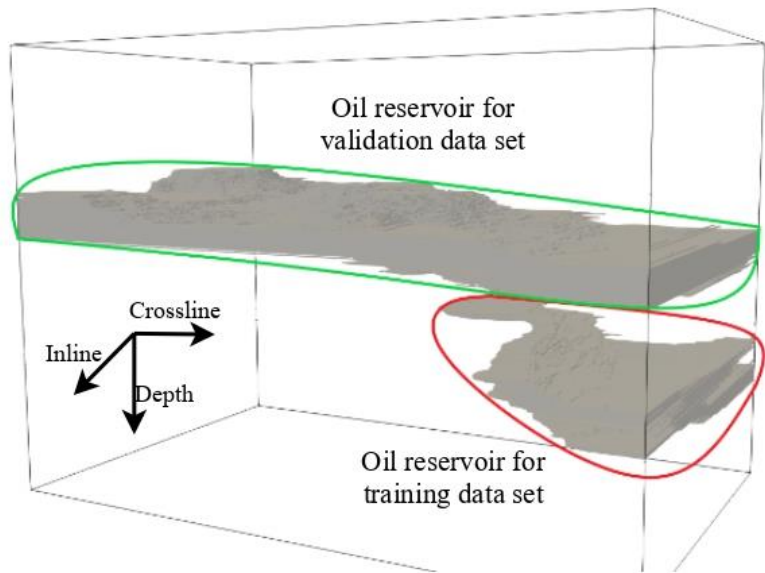
Сравнение исходного сейсмического среза и предсказанного с помощью
модели U-Net:

- А) Сейсмический временной срез
- В) Форма резервуара предсказанная экспертом
- С) Форма резервуара предсказанная моделью

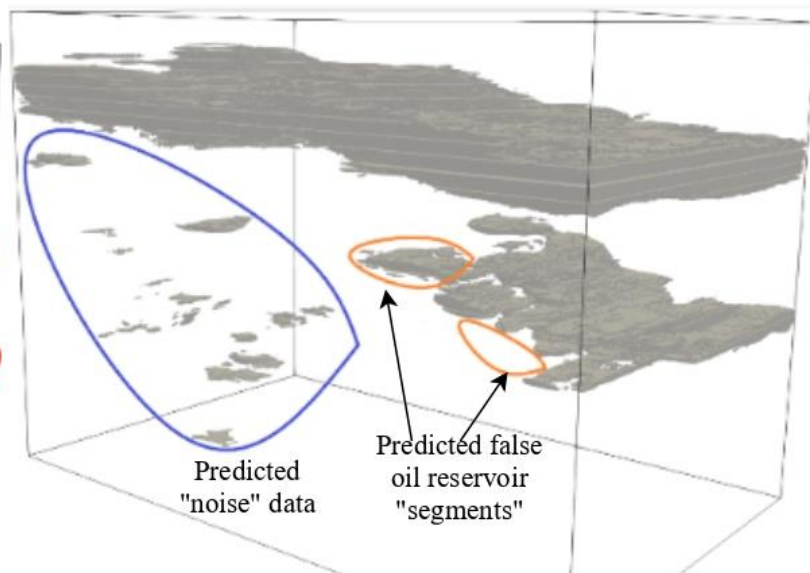


Объемная модель нефтяного резервуара

Ground true model



Predicted model



Как видно рисунка, наша модель смогла корректно восстановить нефтяной резервуар, входивший в обучающий набор, и достаточно надежно восстановить форму второго нефтяного резервуара, входившего в тестовый набор. Следует отметить, что погрешности в прогнозируемых результатах (область внутри синей кривой-это шум в данных, область внутри оранжевой кривой-это неверно предсказанные сегменты нефтяного резервуара в тестовом образце) не являются критичными для правильной интерпретации.

- Задача бинарной классификации изображений сейсмических срезов успешно решена;
- Искусственное дополнение данных – необходимый этап при работе с данными нефтяных месторождений;
- Модель бинарной классификации, полученная с помощью NAS, смогла достаточно точно идентифицировать с нефтяными резервуарами простой формы на изображении. Однако на основе анализа прогнозов становится ясно, что модель неверно классифицирует срезы с резервуарами более сложных конфигураций (вытянутые, неоднородные). Однако данный подход может использоваться в качестве предобработки «сырых» изображений сейсмических срезов.
- Анализ результатов эксперимента по семантической сегментации нефтяных резервуаров, показывает что предложенный подход позволяет выполнять сегментацию, близкую к результатам интерпретации человеком-экспертом. Данный метод позволяет ускорить обработку и анализ данных сейсморазведки, а также упростить процесс интерпретации для специалистов предметной области.

<https://github.com/ITMO-NSS-team/nas-fedot>

Реализация NAS для задачи классификации спутниковых снимков, описанная в презентации

<https://github.com/nccr-itmo/FEDOT>

Ядро фреймворка Fedot, прикладные примеры и бенчмарки.

<https://itmo-nss-team.github.io/FEDOT.Docs>

Документации к фреймворку

<https://github.com/ITMO-NSS-team/oil-field-modelling>

Реализация препроцессинга данных сейсмической разведки.

<https://itmo-nss-team.github.io>
https://t.me/NSS_group

Сайт и официальный новостной канал научной группы «Моделирование природных систем»

Спасибо за внимание!

Ревин Илья
ierevin@itmo.ru

ITMO *re than a*
UNIVERSITY