Machine Learning Engineer Nanodegree

Capstone Proposal

Felipe Alves
September 3rd, 2018

Proposal

Domain Background

Several areas of Earth with large accumulations of oil and gas also have huge deposits of salt below the surface. Unfortunately, knowing where large salt deposits are precisely is very difficult. Professional seismic imaging still requires expert human interpretation of salt bodies. This leads to very subjective, highly variable renderings. More alarmingly, it leads to potentially dangerous situations for oil and gas company drillers [1].

The motivation to work in this project is to dig deep into Machine Learning algorithms and to work with images.

Problem Statement

The problem stated by TGS (the company sponsoring this Kaggle challenge) is to build an algorithm that automatically and accurately identifies if a subsurface target is salt or not [1]. The algorithm will be trained and tested with the images provided by TGS. The results will be evaluated using Intersection over Union (IoU) metric.

Datasets and Inputs

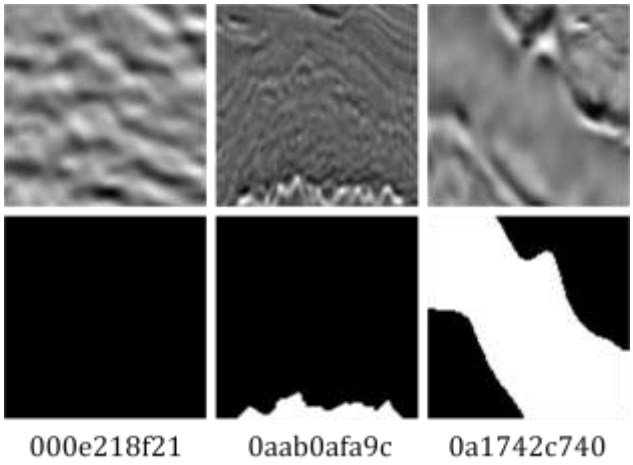
In this project, we will use the TGS Salt Identification Challenge dataset, available on Kaggle [1]. The dataset contains 26,000 images in total. Table 1 explains how they are distributed. The images are 101x101x3 PNG format, though they have 3 channels they are all greyscale.

Table 1 – Images from the dataset

Category	Number of Images	Description
Train	4,000	Images used for training the model
Mask	4,000	Images used as masks for training set (ground-truth)
Test	18,000	Images used for testing the model

Each image is named with a unique id that can be found in the "depths.csv" file, available together with the images. Below are 3 examples of images found in the train set and their respective ground-truth.

Figure 1 – Different images and their masks



The white area in the masks shows where the salt is. Id 000e218f21 has no salt at all.

Solution Statement

The solution will require two neural network architectures. The first one is Inception V3, a convolutional neural network pre-trained on ImageNet dataset [2], which will act as a backbone for U-net. U-net is a cnn developed for biomedical image segmentation [3]. The model will be trained for 30 epochs with patience set to 5 to avoid overfitting. The results will be evaluated using IoU metric.

Benchmark Model

The benchmark model will be Jesper solution using only U-net [4]. His model uses only U-net and achieved an IoU score of 0.62. This project will try to enhance his model by using transfer learning from Inception V3 with pre loaded weights from ImageNet.

Evaluation Metrics

The metric used to quantify the performance of both the benchmark model and the solution model is Intersection over Union metric. This is the metric used by Kaggle in this challenge.

Intersection over Union is an evaluation metric used to measure the accuracy of an object detector [5]. For this project, it will compare the ground-truth mask with the predicted mask. Intersection over Union can be mathematically defined as:

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union}$$

Project Design

The project will be developed in Python 3.6.4. The main libraries used in this project are *pandas, numpy, matplotlib, skimage, keras* and *tensorflow*. The theoretical workflow for approaching a solution consists of the following.

Firstly resize and load all train images and masks to match the required input for Inception V3. Secondly U-net model will be imported using Inception v3 as backbone with ImageNet pre-trained weights [6]. Then the IoU metric will be defined to evaluate the results.

The model will be trained on a p2.xlarge instance of Amazon EC2 (AMI) with epochs set to 30 and patience set to 5. The best weights for this problem will be saved on 'model-tgs-salt-1.h5' file. The training set will use 90% of the original set (3,600 images), the other 10% (400 images) are for validation.

To test the model, all the images will be resized to match the required input and they will be tested using the best weights generated by the training step. The resulting masks will then be resized back to 101x101 (original size) and be transformed into a csv file, ready for submission.

References

1 - https://www.kaggle.com/c/tgs-salt-identification-challenge

2 - https://www.tensorflow.org/tutorials/images/image_recognition

3 - https://en.wikipedia.org/wiki/U-Net

4 - https://www.kaggle.com/jesperdramsch/intro-to-seismic-salt-and-how-to-geophysics

5 - https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/

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