Machine Learning Engineer Nanodegree

Capstone Proposal

Ronan Casey
November 3rd, 2018

Using Convolutional Neural Networks to Identify Icebergs in Dual-Polarization C-Band Radar Images.

Domain Background

Icebergs have been an ever-present obstacle for ocean-going vessels as long as that mode of transportation has existed. With much of the sea traffic consisting of commercial deliveries such as oil, there is a significant necessity in providing reliable results where iceberg identification is required in the interest of safe working conditions, protection of assets and environmental responsibility.

Tried and tested methods for identification come in the form of aerial reconnaissance and shore-based support. The combination of these methods will reliably identify the location of icebergs and conditions associated with heightening the risk of the presence of icebergs. However, in the remotest most inhospitable locations where these methods are unavailable, the only information available comes in the form of satellite data.

Problem Statement

The satellite radar works by pinging a radio wave towards the earth and recording the reflections that bounce back from the surface. The resulting data is converted to an image, where objects appear as bright spots in contrast to their surroundings. The problem presented means that anything solid will have a similar appearance. Land, islands, sea ice, icebergs and ships are all easily misclassified without detailed analysis. For the purposes of this project, the focus will be on differentiating icebergs from ships, exclusively. Processing the image data manually is demanding and monotonous for humans. Offloading the classification task to a computer would allow the human observers to focus all their energy on certifying the iceberg predictions. In this project I will use a deep learning approach to identify subtleties and features within each image and associated inc_angle value while outputting a probability related to the presence of an iceberg with 1 representing an iceberg sighting and 0 representing a ship sighting.

Datasets and Inputs

The data is provided in json format from the Kaggle Competition titled Statoil/C-CORE Iceberg Classifier Challenge. Each satellite image appears as a list item with the following fields:

id - the id of the image

band_1, band_2 - the flattened image data. Each band has 75x75 pixel values in the list, so the list has 5625 elements. Note that these values are not the normal non-negative integers in image files since they have physical meanings - these are float numbers with unit being dB. Band 1 and Band 2 are signals characterized by radar backscatter produced from different polarizations at a particular incidence angle. The polarizations correspond to HH (transmit/receive horizontally) and HV (transmit horizontally and receive vertically). This information will provide the bulk of the data used to train the neural network.

inc_angle - the incidence angle of which the image was taken. Note that this field has missing data marked as "na", and those images with "na" incidence angles are all in the training data to prevent leakage. Generally, the ocean background will be darker at a higher incidence angle. This information will provide the neural network with useful additional data when generating features.

is_iceberg - the target variable, set to 1 if it is an iceberg, and 0 if it is a ship. This field only exists in train.json

The labels are provided by human experts and geographic knowledge on the target.

Solution Statement

The iceberg identification problem presented is best suited to a deep learning solution. A Convolutional Neural Network (CNN) will be able to determine much finer levels of subtlety within the images than a human eye could ever accomplish. A CNN scans images to find outlines and shapes which can be combined to make unique identifiable features. These features can ultimately be identified in a previously unseen image to predict if a particular object is present, which in our case is an iceberg. Some additional preparation and processing steps will be required to make the data presentable to the CNN. The aim of this project is to design a CNN model using the Keras API for tensorflow that can reliably identify icebergs from satellite image data without the need for human involvement.

Benchmark Model

This model outlined in the solution above can be compared to many models available through the Kaggle Competition Leaderboard. Many of the benchmark models available are based on deep learning algorithms such as CNNs, but some are built using supervised learning algorithms such as Support Vector Machines (SVM).

The Kaggle Competion scores are measured using a log loss function. The log loss function is a measurement of the uncertainty associated with the model predictions.

The log loss function uses the values of each prediction (0, 1) and prediction probability (0.0 - 1.0) to measure the performance of a model. The greater the amount of correct predictions and the higher the probability attributed to those predictions will result in a lower log loss value. Models which achieve a lower log loss value are considered to be better performing.

Evaluation Metrics

The evaluation metric used in this project will be the same metric used in the Kaggle Competition, the log loss function. The formula for the log loss function is as follows:

$$logLoss=-1N\sum Ni=1(yi(logpi)+(1-yi)log(1-pi))$$

After training, the model will analyse previously unseen test data and compare the predictions with the test set labels. The results will be calculated using the log loss function where values closer to zero are optimal.

Project Design

The first step of this project is to collect all of the data from the Kaggle Competition and inspect it for maximizing the value it can present to the CNN.

The data used in this project is already well sorted and will only need a small amount of pre-processing before it can be used to fit the model. My approach to solving this problem will involve using all of the included training data (band_1, band_2, inc_angle) to train a CNN towards generating features which are associated with either icebergs or ships.

The "inc_angle" data in the training set is presented as incomplete with 'na' values present for a number of observations. The first thing to address will be to fill in these missing values appropriately using either a 0 value or an estimated value. My approach will try each of these methods to see which one is more advantageous.

Once the data is tidy, the flattened images must be converted to 2D arrays which can be inserted at the input layer of the CNN. The training data and test data will be processed individually and kept separate at all times. The CNN will contain two input layers. One layer for the 2D processed image data and a second layer for the inc_angle data of each image.

The hidden layer architecture and parameter values will be subject to experimentation during the design process. The output layer will consist of the targets in the training set (1 = iceberg, 0 = ship). The training images will be manipulated randomly to stop provide the CNN with extra image variety towards making the model more robust and less likely to over fit the training data. The final stage is to fit the model using our clean and processed data and evaluate the log loss performance using the test set.
