

Machine Learning Engineer Nanodegree

Capstone Proposal

Shehabul Hossain
November 17th, 2017

Proposal

Domain Background

Image classification is a topic of pattern recognition in computer vision. It is an approach to classify images based on the context of the image. Here “contextual” means the approach of focusing on the relationships of nearby pixels. In the late 1960s, different universities started computer vision which was pioneering artificial intelligence. Currently, there are different approaches that are used in image classification. One of the most popular methods is the Convolutional neural network (CNN). It is a deep learning approach used in machine learning which uses deep, feed-forward artificial neural networks that have been applied to analyzing visual imagery. One of the very first CNN was implemented in 1994 in the field of Deep Learning. CNN is being used even today to successfully classify images or extract data from images. Computer vision has come a long way since it's inception still, it has many challenges that are yet to overcome.

Problem Statement

Drifting icebergs are one kind of threat to navigation and activities in offshore areas. It can do serious damage to passing ships. Many companies use aerial reconnaissance and shore-based support to monitor environmental conditions and assess risks from icebergs. However, in harsh weather conditions, the only way is to monitor using satellite. Still, the data has to be processed manually in order to differentiate icebergs from other objects like a ship. It is a very tedious job to classify icebergs using satellite signals. To solve this problem machine learning can be used. The satellite collects data as an image and the objective is to create an image classifier that can find icebergs in images. A CNN can be used in this case as CNN's are very good at classifying images. The CNN model will take an image as an input and look for icebergs in that image. The output will be a number between 0 and 1 which will represent the probability that the image contains an iceberg.

Datasets and Inputs

Statoil, an international energy company operating worldwide, has worked closely with companies like C-CORE. C-CORE has been using satellite data for over 30 years and has built a computer vision based surveillance system. To keep operations safe an efficient a more efficient system can be implemented using machine learning. The company released the data in a Kaggle competition to find an efficient solution using machine learning.

The satellites that are used to detect icebergs are 600 kilometers above the earth using a radar that bounces a signal off an object and records the echo, then the data is translated into an image. The C-Band radar operates at a frequency that can see through darkness, rain, cloud and even fog. Echos from different objects are recorded and then translated into an image. An object will appear as a bright spot because it reflects more radar energy than its surroundings, but strong echoes can come from anything solid - land, islands, sea ice, as well as icebergs and ships. The energy reflected back to the radar is referred to as backscatter. Many things include winds affect the backscatter. High winds generate a brighter background and low winds generate darker. The Sentinel-1 satellite is a side-looking radar, which means it sees the image area at an angle (incidence angle). Generally, the ocean background will be darker at a higher incidence angle. You also need to consider the radar polarization, which is how the radar transmits and receives the energy. More advanced radars like Sentinel-1 can transmit and receive in the horizontal and vertical plane. Using this, you can get what is called a dual-polarization image.

Here, we have data with two channels: HH(transmit/received horizontally) and HV(transmit horizontally and received vertically). This can play an important role in classifying¹ as different objects tend to reflect energy differently. All the images are 75x75 images with two bands and we also have inc_angle which incidence the angel of which the image was taken.

We have two data files (train.json, test.json). The files consist of a list of images and for each image, we have 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).
- inc_angle - the incidence angle of which the image was taken. This field has some missing data marked as “na”, and those images with “na” incidence angles are all in the training data to prevent leakage.

- `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 `train.json` has 1604 rows and the `test.json` has 8424 rows.

Solution Statement

In order to solve the problem described above, we will use Deep Learning with a Convolutional Neural Network to create an image classifier. We will train the CNN with the training data so that it can correctly identify iceberg from other objects. The final model will take an image as an input and output a number between 0 and 1 predicting the probability that the image contains an iceberg. The feature `inc_angle` has a lot of missing values. As a result, it would not be a good idea to use this as a feature.

Benchmark Model

As this one is a binary classification problem with images as an input the benchmark model would be a single layer CNN. While training the CNN will take images as an input and then iterate over the image with its convolutional window in order to extract features in those images. As a binary classification, we can use a sigmoid function to the CNN which will give us the probability of the image being in either of the classes.

Evaluation Metrics

The results will be evaluated on the log loss between the predicted values and the ground truth. For each image in the test data there will be a predicted value from 0 to 1 which will represent the probability of the image containing an iceberg.

In multi-class version of the log loss metric at each observation is in one class and for each observation there is a output probability for each class. The metric is negative the log likelihood of the model that says each test observation is chosen independently from a distribution that places the submitted probability mass on the corresponding class, for each observation.

$$logloss = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^M y_{i,j} \log(p_{i,j})$$

where N is the number of observations, M is the number of class labels, \log is the natural logarithm, $y_{i,j}$ is 1 if observation i is in class j and 0 otherwise, and $p_{i,j}$ is the predicted probability that observation i is in class j .

Project Design

Programming language, frameworks and libraries

In this project, I am going to use python 3.6 with some helpful python libraries like pandas, numpy, matplotlib and seaborn to analysis the data. Then I will be using Keras which is an open source neural network library that runs on top of TensorFlow or Theano. In this case, I will be using TensorFlow as backend.

Workflow

First of all, I will be using pandas and numpy to analyze the data and try to find correlations between them. This analysis is done in order to understand the data better. As we have data for band_1 and band_2, we can treat them as two different channels to create two channeled image or we can make a third channel using their average and then create three channels which can be compared as a 3-channel RGB equivalent. If we think the data is not enough to find the trend in the data and also in order to reduce overfitting we could use image augmentation to generate more data from the existing data. Now that we have a 3-channel RGB equivalent data which is 75x75x3 in dimension, we can treat them as images and create a CNN and then train it so that it can find icebergs in any image given.

Convolutional Neural Networks (CNN)

CNN is one of the most popular ways to in Computer Vision nowadays. It does a great job extracting information from images. In our case, we want to extract the information that if an iceberg is present in an image or not. For this purpose, we are going to use CNN. We will be using transfer learning to modify some of the popular CNN models that are available like VGG-16, Resnet50, and InceptionV3. We will modify these pre-trained models to make a better prediction. Finally, with a satisfying result, we will create a pipeline that will input an image from the satellite and predict the probability of the image containing an iceberg.

References

- C. Bentes, A. Frost, D. Velotto and B. Tings, "Ship-Iceberg Discrimination with Convolutional Neural Networks in High Resolution SAR Images," Proceedings of EUSAR 2016: 11th European Conference on Synthetic Aperture Radar, Hamburg, Germany, 2016, pp. 1-4.
- J. Wu, "Introduction to Convolutional Neural Networks", National Key Lab for Novel Software Technology, Nanjing University, China, 2017

K. Jarrett, K. Kavukcuoglu, Y. LeCun, “What is the best multi-stage architecture for object recognition?”, 2009 IEEE 12th International Conference on Computer Vision, pp. 2146–2153. IEEE (2009)

A. Krizhevsky, I. Sutskever, G. E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing 25.”, Harvard, 2012

Statoil/C-CORE Iceberg Classifier Challenge, Kaggle, <https://www.kaggle.com/c/statoil-iceberg-classifier-challenge>
