Capstone project proposal

Kaggle Iceberg Classifier Chanllenge: <https://www.kaggle.com/c/statoil-iceberg-classifier-challenge>

# Identify the problem

This problem is about predicting icebergs in images from satellites. The goal of the project is to find out the best predicting technique for this problem and improve the accuracy of prediction.

# The client

The client is the remote sensing company. They can use the technique to process satellite data and predict certain threaten conditions to ships and oil platforms in the ocean.

# The data

For each dataset to be predicted, there are two image channels (band1 and band2) and an incidence angle data. There are 1604 datasets in the training.json file. In the 1604 datasets, 753 of them are “is\_iceberg”. There are 8425 datasets in the test.json file. All the images in datasets are preprocessed, with the object roughly in the middle of the image. Some of the incident angle data are missing.

To acquire more data for the training purpose, I will use the pseudo sampling, since there are a lot more data in the test dataset. I will only use the dataset predicted with a fair model and with a more confident score (more than 0.9 or less than 0.1 in a 0-1 scale).

# Approaches

Convolutional neural network (Covnet) consists of feature layers and classify layers, is a powerful tool for image classification. It makes sense focusing on this technique to get the best prediction accuracy. This technique works well finding edges and identify features in the image. Usually, it does not need the feature engineering required for other classifiers. However, the rules of design Covnet architecture are not clearly developed. And the number of dataset is much smaller than the number of parameters, so overfitting will be a problem. One solution is finding a success developed Covnet with similar datasets and find-tuning some of the parameters. I’m not sure if I can find one retrained Covnet similar enough with our dataset.

The ideal method would be only using the feature layers in a typical Covnet architecture to generate object edges and fine features. Then, instead of using a full connected neural network classifier usually found in the Covnet architecture, use one of the simpler classifier to avoid overfitting.

Other approaches are using simpler classifier such as decision tree, support vector machine and k-nearest neighbor. Before using those classifies, feature engineering will be conducted. For example, I will probably use object detecting library (OpenCV) to detect the object in the image; and then find out the shape, size and brightness of the object; finally, feed the data into classifiers.

Characters used in the prediction:

1. Shape, size.
2. Brightness in band1 and band2.
3. Incident angle data – not sure how that affect the prediction yet.

Things make the prediction complex:

1. The background of the image varies due to wind condition and incident angle.
2. Some of the iceberg may have a ship-like shape/size.

# Deliverables

Code, the milestone report and the final report.

In the coming reports, I will report the predicting accuracy from the about approaches, along with the process of Covnet fine-tuning, Covnet visualization, feature engineering, explosive data analysis.

Useful links:

1. <http://cs231n.github.io/convolutional-networks/> -- Karpathy intro to Covnet
2. <https://kaggle2.blob.core.windows.net/forum-message-attachments/69182/2287/A%20practical%20theory%20for%20designing%20very%20deep%20convolutional%20neural%20networks.pdf> -- Practical theory about design Covnet
3. <https://flyyufelix.github.io/2016/10/03/fine-tuning-in-keras-part1.html> --How to fine-tuning Covnet.