Statistics 440 - Case Study 3
Kaggle Competition:
Statoil/C-CORE Iceberg Classifier Challenge

Barinder Thind
301193363
In Collaboration with: Matthew Reyers, Brad Smallwood, Ryan Sheehan
Simon Fraser University

Abstract

For our final case study, we were tasked with trying to predict whether or not a given image is an iceberg or a ship for a kaggle competition. There were numerous challenges in all the different components that made up this case study such as for: data reading, data cleaning, and model building. This was largely due to the fact that we had never dealt with images before or data in this format. Ultimately, we were fairly successful in developing a group of convolutional neural nets that, after being stacked, had a log-loss of 0.1456. This put us in the top 100 (and in Bronze) for this kaggle competition of over 1900 participants.

Keywords: Mixture Models, CNN, Image Analysis, keras/tensorflow, xgboost

---

I. Introduction

This is our last case study and for it, we were asked to look into solving a kaggle data science competition problem about identifying whether an object is an iceberg or a ship. These predictions are made using image data that has been fast fourier transformed. As this was our first attempt at image analysis of any kind, this case study posed unique challenges when compared with all other data analysis we had done thus far. For example, the data came in a format we had never seen before - each observation came with two variables that were not just single digits or characters - they were lists of numbers. On top of that, we had to look into different methods of analyzing this data including introductory deep learning techniques.

The following few sections are organized as follows: I first present a detailed description of the dataset along with some comments about the models we used to get our predictions. Next, I delve into the particular results, including our ranking in the competition. Finally, I conclude

by providing some ways to improve the results in the future and summarizing everything that went into my results.

II. Methodology

In this section, I describe the data and the models we attempted.

1. Data Description

The data is split into a training and test set. Within the training set, there are five variables:

- Id: A unique value corresponding to each image
- band-1, band-2: 75x75 Pixel list of the flattened images
- inc-angle: The incidence angle of which the image was taken
- is-iceberg: Whether or not this particular image is an iceberg or not

Similarly, for the test set, we had the same set of variables excluding the one corresponding to what the image actually it. Finally, there are 1604 observations in the training set whereas there are 8424 in the test set.

2. Model Building

There were a number of ways for us to approach this problem. First however, we had to clean the data. Looking at the images, it was obvious that the "noise" pixels had the potential to contribute (in a harmful way) to our model accuracy. In order to avoid this issue, we had to "smooth" out these pixels. Dr. Jack Davis's code was instrumental in tackling this problem and ultimately, we decided to use a mixture model. In particular, with the help of the optim() function in R, we implemented the maximization expectation algorithm. Here is an example of its results:

Next, we had to convert this data into a particular format such that it is appropriate for model building. Initially, we transformed the data set such that each pixel was now a variable. This made it easy to use with various models such as the random forest and the basic neural net. We

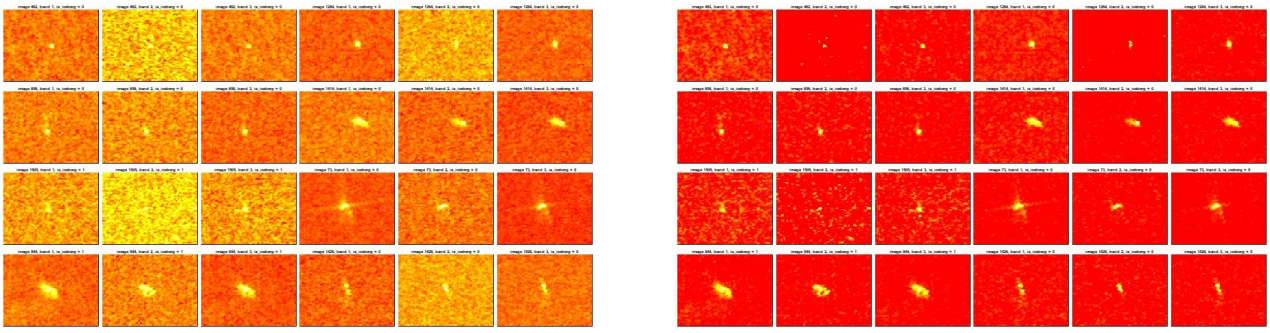

Figure 1: [Left] - Uncleaned Data — [Right] - Cleaned Data

also used the LASSO to try and eliminate some of the noise variables. Ultimately, the results for these models were not the present[1].

Looking at some of the kernels presented by others in the competition, we attempted to implement a convolutional neural net (CNN). A CNN is a particular type of neural net that isn't fully connected i.e each neuron is connected to only a few neurons in the previous layer. On top of that, neurons share weights whereas in the traditional (fully connected) neural net, each parameter had its own weight. In a CNN, we take into account adjacent pixels (or variables) and look for correlations between them. The following image illustrates this:

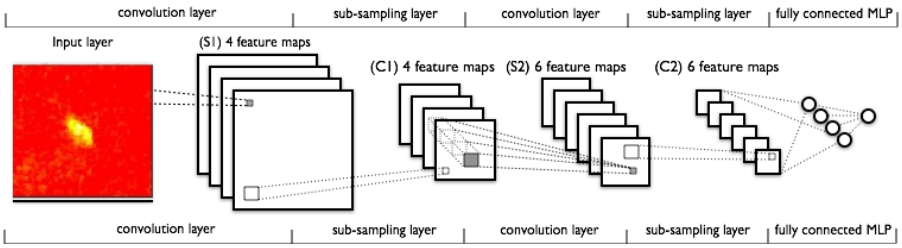

Figure 2: Convolutional Neural Net Illustration

You can see that from our image, the convolutional neural net takes some chunk of pixels, filters it through "feature maps" and using this information, repeated for the rest of the pixels, it makes a (binary, in our case) prediction about what that image is.

[1]Table 1. has these results

3. Evaluation

Lastly, in order for us to be evaluated, the kaggle competition used the log loss metric which is defined as follows:

$$L = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log\, p_i + (1 - y_i) \log\,(1 - p_i)].$$

This is the usual metric for binary responses. It is relatively unforgiving as it heavily penalizes classifiers that are confident about an incorrect classification.

III. Results & Discussion

Initially, we attempted various methods on the expanded data set[2]. The results for each set of these is summarized in the following table:

Model	Log Loss
LASSO + Neural Net	0.57832
Full Data Neural Net	0.53291
PCA + Random Forest	0.62910
Full Data LDA	1.1293
Random Forest	0.47820
Full Data Logistic Regression	0.89301
XGBoost	0.43271
PCA + Logistic Regression	1.2531

Table 1. of models and their corresponding results.

These were not the best and it could be because the methods we applied were just not suitable for this kind of data. In fact, the CNN is specifically built for this sort of data and thus, we applied it next. Using various kernels as references, we created many variations on our neural

[2]The data set in which there was a variable for each pixel

nets. The following is our ranking for our best, purely model-based, submission into the kaggle competition[3]:

Figure 3: Kaggle Ranking

Finally, to try and improve this prediction, we attempted stacking our score. The result from stacking was actually worse than the one from our best CNN. This could be a fault of ours as our implementation may not have been the best. We had no prior experience using such a method to improve our scores and we have made critical errors in getting to our results. Another "stacking" technique was to try and avoid the impact of wrong extreme scores (probability predictions $>> 0.9$ or $<< 0.1$) by taking weighted averages across our predictions. This made sense to us because, for example, if we had two sets of predictions in which we had diametrically opposed predictions for some particular observation, then averaging them would reduce the impact of the wrong one (which one of them must be). We did this with a plethora of predictions we made along with predictions by others and got the following ranking on kaggle[4]:

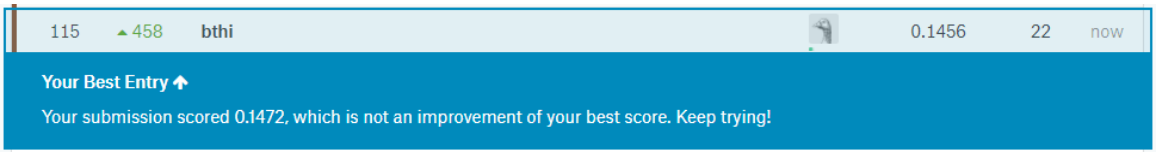

Figure 4: Final Kaggle Ranking with Stacking

IV. Conclusion

Our challenge here was to try and predict whether or not an image is an iceberg or a ship. While there were numerous challenges, we still were able to place in the top third of the kaggle competition (at the time of this case study submission). We tried a myriad of models to try and get a log loss score that is as low as possible. Ultimately, the best method was the one that took advantage of the CNN. For the future, we will look at understanding and using stacking

[3]My groupmate, Brad Smallwood, made this particular submission
[4]This is lower than any kernel or accessible data set we could have got

methods to - hopefully - improve this score and move ourselves up in the rankings! Regardless, this was a great learning experience.

V. References

- Dr. Jack's Code & Lectures
- https://www.kaggle.com/devm2024/transfer-learning-with-vgg-16-cnn-aug-lb-0-1712
- https://www.kaggle.com/dimitrif/keras-with-data-augmentation-lb-0-1826
- https://www.kaggle.com/mihaskalic/keras-straightforward

Word Count: 1053 [5]

Total # of Kaggle Submissions: 37 [6]

Lowest Kaggle Score: 0.1456

[5] Excluding the abstract

[6] Between Brad and I