Understanding the Amazon from Space

Anton Shvets

University of New Haven — GalvanizeU — July 21, 2017

Abstract

For this project my goal was to label satellite image chip data provided by Planet with various atmospheric and land labels. First I performed exploratory data analysis to visualize the distribution of labels and their correlations with one another. Visualizing the satellite images themselves also warned me about potential errors. After preprocessing the data I created a simple non-deep learning model as a reference to help me have a better understanding of the benefits and pitfalls of deep learning upon completion of this project. Next I overfit a few simple convolutional neural network (CNN) models to achieve 100% accuracy on the first few images. Finally after performing hyper-parameter tuning I had my best model ready which resulted in 95% test accuracy and achieved 85% on the fbeta (beta = 2) metric.

Introduction

The data for this Kaggle competition was provided by Planet, designer and builder of the worlds largest constellation of Earth-imaging satellites. The satellite image chips were of the Amazon forest. Deforestation in the Amazon Basin accounts for the largest share, contributing to reduced biodiversity, habitat loss, climate change, and other devastating effects. Better data about the location of deforestation and human encroachment on forests can help governments and local stakeholders respond more quickly and effectively. While considerable research has been devoted to tracking changes in forests, it typically depends on coarse-resolution imagery from Landsat (30 meter pixels) or MODIS (250 meter pixels). This limits its effectiveness in areas where small-scale deforestation or forest degradation dominate. The motivation of this challenge was to develop a robust method for dealing with high resolution imagery data.

Data Processing

Before setting out to build my model I did some exploratory data analysis to understand the type, size and distribution of my data. I concluded that the images contained 17 unique labels: four of those labels are atmospheric labels, 13 are various land use labels. There is a big label imbalance in the data set. The labels were one-hot encoded (technically multi-one hot encoded since on average there were 2.8 labels per satellite chip). The images were converted to an array of pixel intensity values which where standardize via simple division by the largest intensity value (255). Co-occurrence matrices confirmed that there was exactly one atmospheric label per image and if that label was not ‘cloudy’ there would be at least one land use label. Looking at the satellite images training data with their labels revealed that a portion of the images were mislabeled by hand which will lead to some funky results down the line (Figure 1).

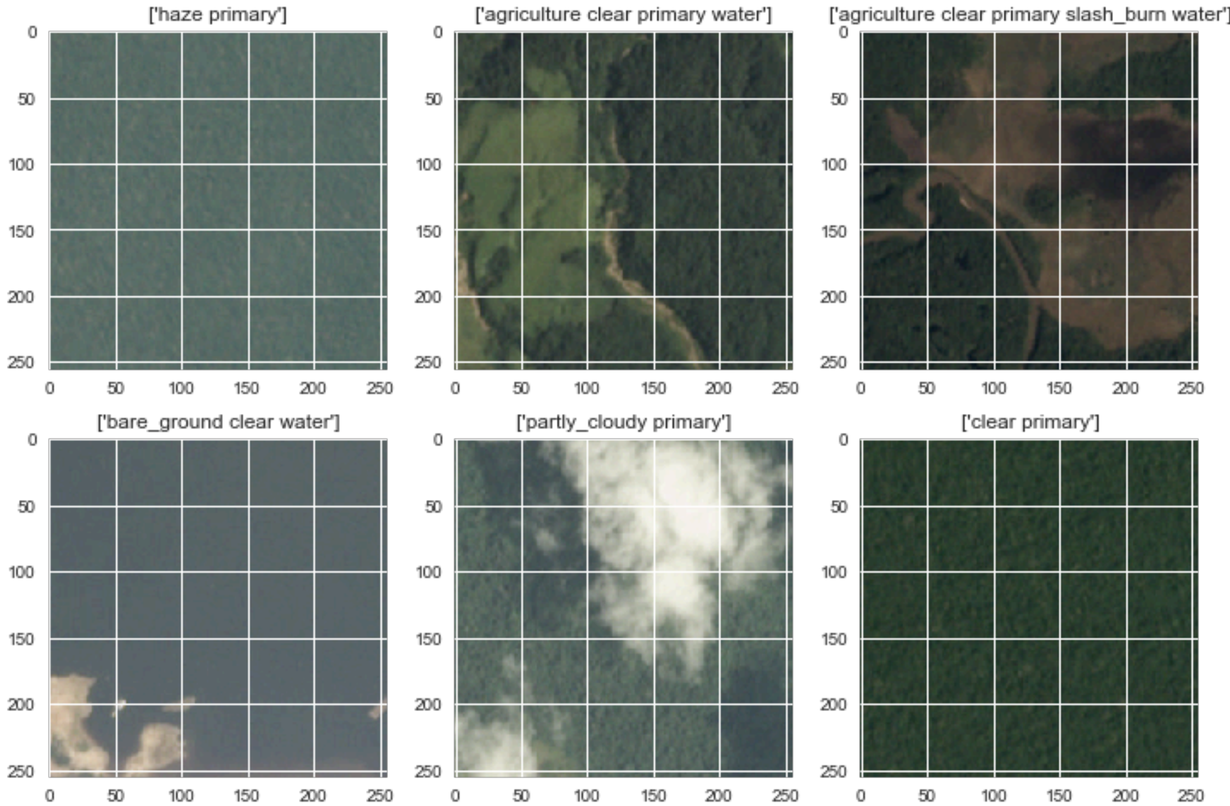
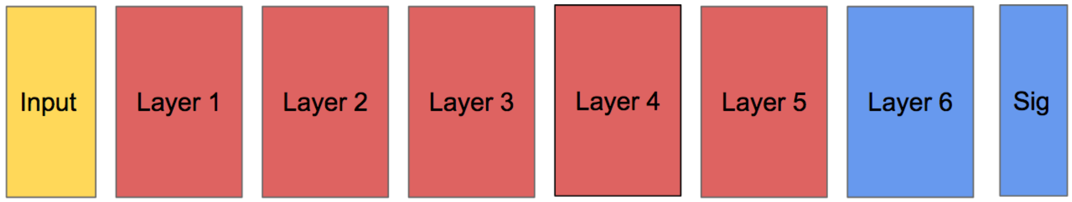


Figure 1. The above image shows six sample satellite image chips with their associated labels. Some of the images seem to be mislabeled: top row, middle image doesn't appear to have water while the bottom left image perhaps should be labeled as 'partly cloudy' instead of 'clear'. Mislabeling proved to be a significant cause of error in my analysis.

Training

To have any sort of perspective of how well my future 'best model' will be doing I needed to have a base model. I ended up with two. The first was a non deep-learning model so I could have an understanding of some some of the advantages of deep learning. For the non deep-learning model I used a Support Vector Classifier to predict only the atmospheric labels. I ended up with a 73% accuracy on the test data which is way better than guessing which would be about 25%. The second baseline model was a simple Neural Network with one flatten layer and one fully connected Dense layer. This model predicted all the labels with a 90% accuracy. Considering the big label imbalance in my data set, accuracy is not the best metric of choice. I decided to use Fbeta (beta = 2) for a simple reason that was due to the fact that the Kaggle competition required that to be the metric. (As opposed to an F1 score which is typically the go to for dealing with class imbalance the completion went with F2 as the metric of choice perhaps for its sensitivity for recall: tp / (tp + fn)). The simple Neural Network model only achieved .64 with Fbeta as the metric.

My next step involved overfitting a few simple CNN models in order to get 100% accuracy on the first five training points. This proved to be a useful exercise as it shed light on some future difficulties I might have predicting land use labels. It also helped me understand why binary-crossentropy is the ideal loss function for multi-label classification.



Layers 1-5: Conv2D (padding), Activation = ‘elu’, Conv2D, Activation, MaxPooling, Dropout

Layer 6: Flatten, Dense, Activation, Dense, Activation, Dropout, Dense, Sigmoid

Figure 2. The final architecture for my best model is shown above. Layers 1-5 were composed in the same way. Only the first Conv2D in each of those layers contained padding. The dropout rate was set to 0.1 and the learning rate for the model was 0.001. Layer six contained fully connected Dense layers with dropout to prevent overfitting and lower the number of parameters used. After the last fully connected Dense Layer a sigmoid activation function was used which is a typical go to last activation for multi-label classification.

Now that I had a basic understanding of what my network architecture will look like I set out to tune my hyper-parameters in order to get the best results. Although very time consuming even with the help of 2 GPU’s on the Google Cloud Platform I performed GridSearchCV to optimize my hyper-parameters. Through sheer brute force I discovered the best activation function for my network was ‘elu’: exponential linear unit. The number of hidden layers were optimized by balancing the time it took to train my network and the increase in the Fbeta score that the extra layer provided.

Upon optimizing all parameters and hyper-parameters I began training my best model on 80% of the total 40k+ images (10% of training data was used as validation). This model achieved a 95% accuracy and a Fbeta score of .85. The 5% increase in the accuracy is overshadowed by the 20% increase in the Fbeta score (see Figure 2 for final Architecture).

Error Analysis

My best model shows promising results, and the use of dropout and max pooling generalizes the model quite well. However, due to the label imbalance results could be further improved with more images showing the less frequent labels. Also upon visualizing some of the training images and their labels it is clear that some of them were mislabeled. Fixing this would require a manual check of all the images and labels.

Another source of error in my model comes from the difficulty of distinguishing rivers and roads. This error could be reduced by a more complex architecture with parallel models. My current model would be combined with one that incorporates the same images but taken with a near-infrared band (such images were also provided by Planet). The input images for the two models would be the same, but the two different filters would provide more insight that would help distinguish between some of the labels that confuse the current single model.

Conclusion

My goal was to correctly label satellite image chips provided by Planet with various atmospheric and land labels. I have accomplished this task with the help of a deep-learning model that incorporates repeated Convolutional Layers, Maxpooling, and various other Layers. My model achieve a 95% accuracy, but the best metric of choice was an F2 score for which I got 85%. The next step would be to incorporate near-infrared band images that could help the model distinguish between similar features, such as roads and rivers, that it gets mixed up at the moment.