Team Brainforest – Planet: Understanding the Amazon from space

Thomas de Bel   Leonieke van den Bulk   Tanja Crijns
Chris Kamphuis   Joris van Vugt   Pieter Wolfert

June 19th 2017

1 Introduction

This report is written for the course *Machine Learning in Practice (NWI-IMC030)* at Radboud University. We entered the competition *Planet: Understanding the Amazon from space*[1] as the second and last project for this course.

The machine learning competition we picked was hosted by Planet, a company which gathers satellite imagery and provides this to businesses, developers and researchers. Every minute the world loses 25 hectares of forest, of which most in the amazon rainforest. By automating the classification of satellite imagery (provided by Planet) local governments and policy makers can act faster on deforestation and human encroachment. The data for this competition consists of so called chips which make up one satellite scene. The chips have a resolution of 256 by 256 pixels, which is equal to 221.7 acres, and they contain red, blue, green and near-infrared channels.

The aim of the competition was to detect the right label(s) for each chip, which was evaluated using the F_2 score. In total there are 17 labels which can describe a chip: there are labels for weather conditions (cloudy, partially cloudy, haze, clear), labels for land condition (road, habitation, water, bare ground, agriculture, selective logging, slash burn), mining related (conventional mine, artisinal mine) and rainforest related (primary, blooming). Every image has at least two labels, one for the weather condition (which are mutually exclusive) and at least one describing the environment on the surface.

2 Approach

2.1 Data

The data consists of 40,479 images in the train set and 61,191 images in the test set. All images have a size of 256x256 and are available in both `.jpg` with and `.tiff` format. The `.jpg` files contained RGB channels with values between 0 and 255. The `.tiff` files contained 4 bands of data: RGB and the near infrared channel (NIR). The advantage of the `.tiff` files is that the NIR channel can be used to discern living from non-living objects. In our networks we used this NIR channel to calculate the normalized difference vegetation/water index (NDVI/NDWI) for each pixel in an image. These calculated channels provide extra information as they have a high contrast for vegetation and water. A problem with the data was that the support for each label differed greatly. For instance, there are only 100 images for the label ’conventional mine’, while there are 37513 images with label ’primary’ in the training set. We applied test-time augmentation of flipping (horizontally and vertically) and randomly rotating the image, along with the predictions of the normal image.

2.2 Methods

2.2.1 Single label

The initial idea was to train a network separately for each label and combine the results for submission. The labels were divided across the team members in the herafter mentioned division with respective approaches. If no specific approach is mentioned, a standard approach

[1] `https://www.kaggle.com/c/planet-understanding-the-amazon-from-space`

of training separate convolutional neural networks for each label was chosen:

1. *Weather*: cloudy, partly cloudy, haze and clear. The weather labels are mutually exclusive, hence it was appropriate to train one network on all four labels. The first approach was to predict with a convnet and the second approach was to predict with random forest and hand-crafted features such as the mean, standard deviation and detected edges.
2. *Mining*: artisinal mine and conventional mine.
3. *Vegetation*: primary and blooming. The primary and blooming labels work well together because blooming can only occur when there's primary. Three different convnets were trained; one for blooming, one for primary and one for primary without blooming versus primary with blooming.
4. *Winding, road-like structures*: water, road and selective logging. The approach here was to train three separate convnets for each label. A network was also trained with NIR data for water, as this could have been relevant for this particular label.
5. *Commonly concurring labels*: blow down, slash burn and cultivation
6. *Ground types commonly without rainforest*: habitation, agriculture and bare ground

2.2.2 Multi-label

Another approach is to train a single network to predict all labels. This requires the output layer to use the sigmoid activation function rather than softmax. This network can then be optimized via either computing the binary-crossentropy loss for each output or using the multi-label soft margin loss. The benefits of using a single network to predict all classes are twofold. First, the network is able to learn correlations between labels, such as the previously mentioned correlation between primary and blooming. Second, the network can easily reuse features that are useful for detecting a partical label. This way, labels that have fewer examples in the training set can still be classified effectively.

We trained various network architectures including DenseNet [1], Feature Pyramid Networks [2], VGG16 [3] and ResNet34 [4]. These networks all performed well on the ImageNet challenge and were suggested on the discussion forum on Kaggle[2].

2.2.3 DenseNet

DenseNet [1] is a network architecture that we used a lot during this competition. We decided to use this architecture, because at the moment it achieves state of the art performance on ImageNet and CIFAR. Given that we got a lot of data in this competition, using a DenseNet architecture seems promising.

During the competition it gave the best results up to a week before the deadline. A DenseNet architecture consists of Dense blocks. Dense blocks are series of convolutional layers, where the input of each layer is the output of all the previous layers in the block. When training the networks we tried different architectures. We trained DenseNets with depths of 40, 64 and 121. In the DenseNet paper weight decay was used when training the networks. We trained networks with both some weight decay and no weight decay. We trained networks using a compression rate of 0.5 and with no compression. As input for these networks we used images of sizes 32x32, 92x92 and 128x128. We trained models using 3 channel images (RGB from JPG), 5 channel images (RGB from JPG + NDVI/NDWI) and 8 channel images (RGB from jpg + NDVI/NDWI + RGB from tif).

3 Results

3.1 Single label

All separate networks got worse precision, recall and F_2-scores than a simple multi-label

[2] https://www.kaggle.com/c/planet-understanding-the-amazon-from-space/discussion/33559#185642

convolutional neural network did for all labels.

3.2 Multi-label

Although we intially used the binary-crossentropy loss for optimizing multi-label networks, we later found that multi-label soft margin loss performed better on this task. The results are presented in Table1. Feature Pyramid Networks (FPN) trained with 3 input channels performed best. Using pretrained networks also seemed to increase performance a little, but there are no pretrained networks with 5 input channels unfortunately.

3.3 DenseNet

Using DenseNet we found that the deeper the network, the better the results were on the validation and test set. In general the networks using no weight decay performed better than the networks using weight decay. We found that using no compression worked better than using a compression rate of 0.5. Using images of greater sizes improved the performance of the networks slightly, but it was very costly in terms of computational time. Using more input channels also improved the performance of the model. Using only the NIR data from the TIF images was however sufficient. When also taking into account the RGB channels of the TIF data the DenseNet performance did not increase.

3.4 Ensemble

Our final score was achieved by ensembling <INSERT NETWORKS>. These predictions were averaged. Next, the predictions of each model were averaged and a threshold was determined per label by doing line search for the highest F_2 score on the validation set.

4 Discussion

In the end our best score was achieved using X with a F2-score of Y. This resulted in the Zth place (per the 19th of June 2017) on the Kaggle leaderboard, which includes us in the top 10%.

Overall we are satisfied with the final result. However, as always, there is some room for discussion and improvement. Most striking was that on the leaderboard the number one with a score of 0.93356 and the number hundred with a score of 0.92324 only differed approximately 0.01, which is very little. This means that there clearly is a ceiling in what the current models can learn from this data and that classifying a few labels extra correctly can do a lot for your score on the leaderboard but does not necessarily mean that your model is that much better.

During the competition it was also brought to light on the discussion board that at least 10% of the .tiff data labels were incorrect. This is probably the reason that the best results were achieved using the .jpg files, which is a shame as we think that especially the infrared channel contains a lot of useful information which could have increased performance significantly.

The most important thing we will take away from this competition is that multi-label learning is a lot more effective than single-label learning in a multi-label problem, especially when there are large correlations between the labels.

All in all, we have learned quite a lot in the two competitions we have done for this course. From new network architectures to new approaches to learning to work together efficiently as a team in a big project. In conclusion, we found the Kaggle competitions to have been an educational and successful experience.

5 References

[1] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*, 2016.

[2] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid net-

Network	Image size	Input channels	Test F_2
DenseNet121	128x128	3	0.9218
DenseNet121 (pretrained)	128x128	3	0.9247
ResNet34	128x128	3	0.9254
Feature Pyramid Network	112x112	3	**0.9270**
Feature Pyramid Network	112x112	5	0.9237

Table 1: Performance of various popular network architecures. Networks with 3 input channels were trained with the RGB channels from jpg images. Networks with 5 input channels also used NDVI and NDWI derived from tiff files.

works for object detection. *arXiv preprint arXiv:1612.03144*, 2016.

[3] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

Appendix

5.1 Code repository

A git repository with all code used in this project can be found on Github[3].

5.2 Group Work Distribution

All:
Weekly group meetings and one supervisor meeting. The labels that are mentioned herafter are explained in more detail in section 2.2.1.

Tanja:
Team captain preparations. Trained convolutional neural networks for the *'Ground types commonly without rainforest'* labels. Trained experimental settings of the multi-label network.

Joris:
Briefly worked on blow down, slash burn and cultivation . Trained many different multi-label networks and ensembled them.

Leonieke:
Trained on weather labels for the single label classification using both convolutional neural networks and RandomForests. Trained multi-label model.

Thomas:
Trained single label networks for road and water. Experimented with networks using the NIR channel and implemented NDVI and NDWI.

[3] `https://github.com/TanjaCrijns/MLIP-Brainforest`

Chris:
Created train/validation split. Made a data loader to easily load data for single label networks. Trained networks for the mine labels. Trained several variations of Densenet.

Pieter:
Trained several approaches on primary and blooming labels (all vs. rest, one vs. rest) using different networks (smaller vs. bigger convolutional networks and pretrained networks). Trained multi-label model.