Schneider Electric European Hackathon

Challenge Data-Science: Zero Deforestation Mission

Ramón Sieira Martínez
Francisco Javier Martínez de Pisón
Jose Divasón

Steps

- **Tuning the learning rate, bath size and epochs**
- Augmentation filters
- Random search and ensemble

1) Several test were performed to find an optimal batch size, also adjusting the learning rate and finding an adequate value of epochs needed.

2) We performed some test with a baseline approach. The goal at this step is to have some idea about the performance of a simple model.

3) Different batch sizes (from 16 to 72) were tried

4) Surprisingly, we have obtained slower and worse results with large bath sizes, even though we also tried with different learning rates (keeping it constant, increasing it linearly with respect to the learning rate, etc.). This has been very surprising and an unexpected result, but due to lack of time we have not studied this problem further.

Therefore, we have proceeded with **16 as batch size, 0.00001 as learning rate and 100 epochs.**

Steps

- Tuning the learning rate, bath size and epochs
- Augmentation filters
- Random search and ensemble

Data augmentation is a fundamental task when training any deep learning model.

FINAL GOAL: find quickly augmentations that increase the score.

We use the Kornia library to perform data augmentation on GPU instead of using CPU.

We have tested different augmentation techniques that are available in the library (separately), using the "tf_efficientnet_b3_ns" model.

Due to the lack of time we have not performed many tests, only a few of them: rotations, horizontal and vertical flip, sharpness, saturations, contrasts, etc. We determined their maximum and minimum values.

RESULT: sharpness, saturations, rotations and blur seem to increase the score. With other filters (brightness, contrasts, etc) is not so clear if there is an improvement, so we did not perform any more tests with them.

Steps

- Tuning the learning rate, bath size and epochs
- Augmentation filters
- Random search and ensemble

A random search is performed with different backbones and values of the selected filters and within the ranges defined in the previous step:

```
saturation_min_max = (0.01, 0.20)
rotation_min_max = (0.01, 20.00)
sharpness_min_max = (0.06, 0.20)
blur_motion_min_max = np.array([5, 7])
```

The best models were trained again using pseudo-labelling, i.e., using the prediction that has been previously obtained from the test dataset.

Finally, we build a Weighted Blending Ensemble: we tested the combinations of the 2, 3, 4, ..., N best models that have been obtained, where N is optimized to get the best result. The weights of the models are obtained by optimization with the validation predictions.