Data Challenge Preligens - Données Satellite

Andrew Caunes

November 21, 2021

1 Introduction

Le problème de la segmentation d'images satellitaire peut être envisagé sous de nombreux angles et beaucoup de méthodes éxistent pour y parvenir, avec des résultats plus ou moins satisfaisants. Dans le cas de ce data challenge, il était même envisageable de contourner le problème et ne pas réaliser une segmentation en apprenant directement à prédire un vecteur de proportion de chaque classes.

Pour autant, depuis quelques années, la méthode privilégiée et semblant fournir les meilleurs résultats dans ce type d'exercice est celle des réseaux de neurones, et plus particulièrement les réseaux convolutionnels. Je me suis penché exclusivement sur ce type de méthode pour cet exercice. Il semblerait après expérimentation des autres étudiants que ce soit la meilleure méthode.

2 Data Preprocessing

The dataset contain 18,491 training samples and 5043 testing samples. It's made of satellite images and segmentation masks for the training set, and only the images for the test set. There is no missing data. I started by separating the training set in a training set and a validation set, in order to tune hyperparameters.
With the validation set, we can run our models and test the performance with various data augmentation methods applied on the sets.
Through experimentation and based on previous experience, I found that it is best to apply merely some random rotations and flips to the images and masks. We can already start selecting a model.

3 Model Selection

We were provided with an initial UNet model with some tweaks specifically designed to improve performance on this kind of problem.
I first tried to implement different kinds of models including very basic CNNs with no contraction like in a UNet, but the training always came to a halt and the model was unable to improve very much. I then tried to use a UNet as it is often more efficient for the semantic segmentation problem. This network works similarly to an auto-encoder, with a phase where the widths and heights of the variables are reduced in order to apply a contraction, and then a expansion enlarges the variables again. This process is supposed to help the model capture the most important features and get rid of the unnecessary information.
I tried to tune the model by varying many hyperparameters including : Optimization method, learning rate, data augmentation, batch_size, architecture, number of layers, sizes of filters, pooling, and many others.
Unfortunately, the validation revealed that no tuning was able to beat the initial provided UNet model and its parameters.

4 Final Model and Scores

The final model is similar to the UNet initially provided, trained for 68 epochs with an Adam optimization method of learning_rate 1e-3.

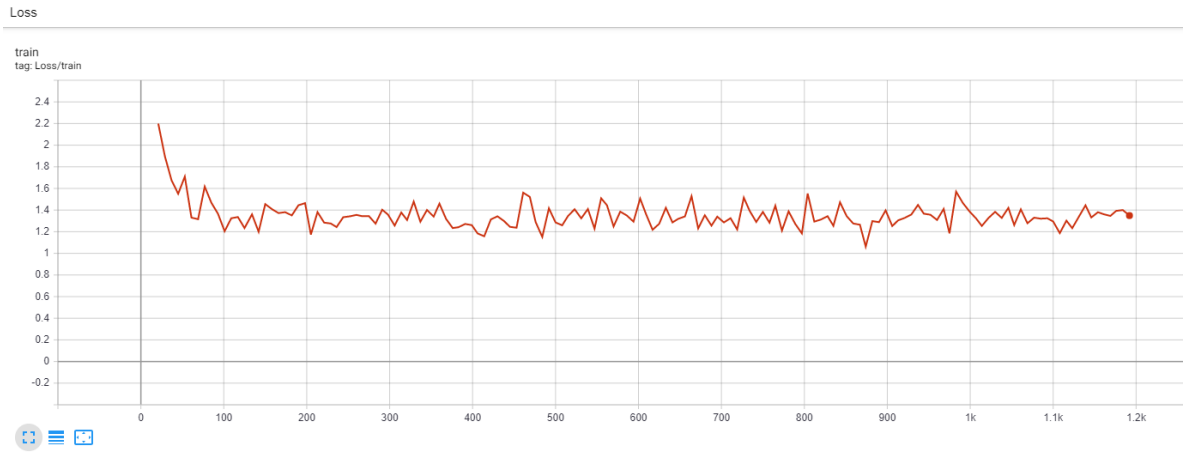

Figure 1: This frog was uploaded via the file-tree menu.

It uses 21 blocks of Batch Normalization and convolution with a ReLU activation. The total number of parameters is near 1,200,000.
The training took about 8 hours on a NVIDIA GTX1070 GPU.

The public score I obtained was 0.0614 while the private score will be disclosed later.

5 Conclusion

It appears that the neural network method is as expected the most efficient to harness the information contained in such a large dataset of images and generalize its acquired knowledge on new images. However, the training can be tricky and the optimal architecture and hyper-parameters for one particular task is not obvious.

Even considering an architecture that has been shown to perform well for a task - the UNet in our case - performance can go down really quickly with slight changes in the model, and a lot of validation is required, which implies a lot of computation time to tune a model to satisfaction.