Trimble / Bilberry : AI Engineer technical exercise
Part 1 : Technical Exercice

Chappe Aslan
April 2023

---

1. Dataset building & Data pre-processing

Browsing through the dataset provided, none of the images seems to be an obvious outlier, then there is no need to handfully remove some images.
Once raw data is ready, the first thing to do is to set up the dataset. Using the ImageFolder [1] function from torchvision, returning a dataset ready for classification.
All of the images from the dataset will undergo transforms such as Resize and Normalization in order to make it easier for the model to deal with it.

80% of the dataset will be for the training, 20% wil be for the validation.

2. Model building

a. Neural Network

In order to tackle a binary image classification problem, the most appropriate neural network to use is a CNN. The goal is to extract features from the images in order to build a model able to recognize if a picture is either a road or a field.
According to the task, a simple CNN with 3 layers should be able to give some consistent results.

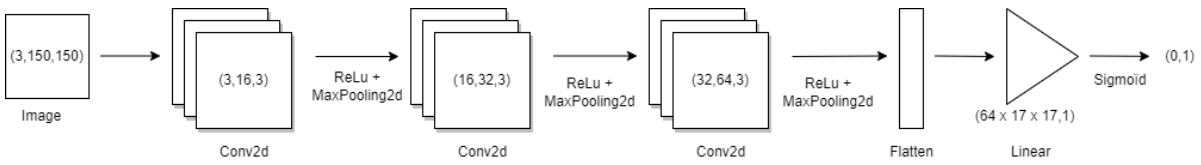

Figure 1. CNN Model for binary classification

The last linear layer will help us to get an output that once through a Sigmoid function will give a value between 0 and 1. Each of these two values represent either if the image is a road or a field (in our case, 0 for road and 1 for field). The closest to these values the output is, the more likely it is that the image contains a road or a field.

Parameters used for this model : Learning rate = 0.0001, Batch Size = 4, Epoch = 20

b. Loss & Optimizer

For this model the loss used is the Binary Cross Entropy (BCE), which is commonly used for binary classification problems. It measures the difference between the predicted probability and the true class value.
The optimizer used is the Adam optimizer, often used for training deep neural networks. It combines adaptive learning rate and the use of gradient moment.

3. Model Evaluation

Going through multiple settings, the best model obtained can identify correctly each of the 10 test images (except for 6 and 7) and is really accurate on pictures outside the dataset.
The model understands the data well enough. However, the training loss defines a pattern that can be understood as overfitting [Fig 2]. Another point is that the training is particularly impacted by the split made to the dataset and results can change according to it.

Image	Expected	Predicted
test_img 1	1	0.99
test_img 2	0	0.02
test_img 3	0	0.15
test_img 4	1	0.55
test_img 5	0	0.14
test_img 6	0/1	0.52
test_img 7	0	0.74
test_img 8	0	0.11
test_img 9	1	0.99
test_img 10	1	0.95

About image 6, the road is on the first plan, and fields are in the background. With the right settings, the prediction is balanced. The model seems to recognize both items in the picture but cannot decide which to clearly prioritize.

About image 7 being in the desert, results are not giving a road prediction. The dataset is not fitted for desert and this prediction is too uncertain. However, depending on the dataset split, the model can adapt to this image.

Using this model on images from Google Images is quite efficient. When not too far from the dataset, the results are positive.

Performances	Training	Validation
Loss	0.0944	0.1545
Accuracy	0.8949	0.9366

4. Correction / To go further

In order to create an optimized model, the obvious first step would be to expand much more the dataset. Indeed, this dataset is not dense enough for the model to train efficiently. Furthermore, the dataset does not cover enough situations of fields or roads (as we can see with the image 7 from the test dataset), and it might trick the model for these situations.
Another way to improve the model is, instead of a basic CNN, a ResNet [2] is much more valuable to extract features from images and easier to use.
For the most efficient and recent solution, Vision Transformers (ViT) [3] is the best model for image classification, with a large dataset and resources to run it.

5. Sources

[1] https://pytorch.org/vision/main/generated/torchvision.datasets.ImageFolder.html

[2] He, Kaiming et al. “Deep Residual Learning for Image Recognition.” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*

[3] Dosovitskiy, Alexey et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.” *ArXiv* abs/2010.11929

6. Additional Data

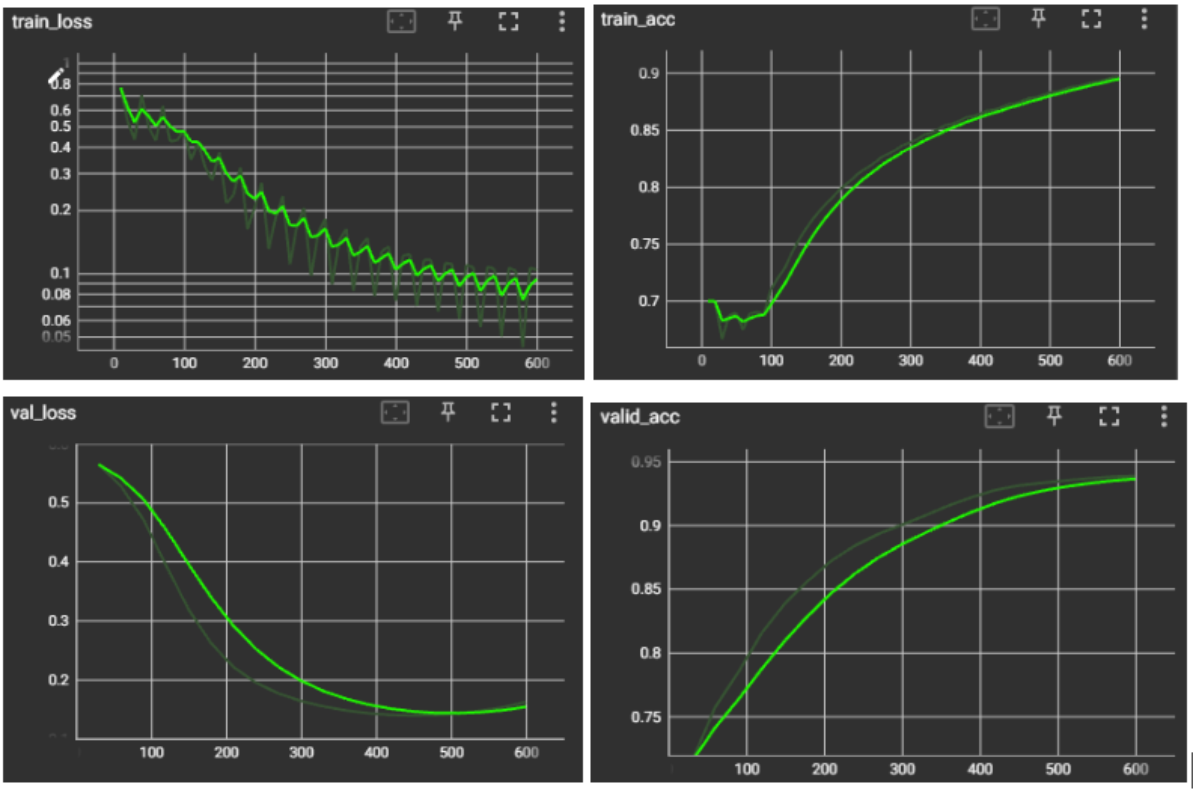

Figure 2. Training and validating curves