Laboratory 15: TensorFlow semantic image segmentation

In this laboratory exercise, you will train a variant of reduced FireNet (Gupta et al. 2023) with an attached modified DeepLabv3+ (Chen et al. 2018) semantic segmentation module to decern land cover[1] from high–resolution (50 cm/pixel) RGB satellite images of the French Hautes–Alpes.

The data are a reprocessed subset of Guérin et al. (2021; https://github.com/koechslin/swin-transformer-semantic-segmentation) with the images cut into smaller tiles (each 1,000 $\times$ 1,000 pixel image became 49 128 $\times$ 128 pixel images), the tiles filtered to remove those with more than 1% missing data, and the building and road classes merged into a single anthropogenic class. In total there are 6,650 images: 4,602 (69.2%) training, 1,024 (15.4%) validation, and 1,024 (15.4%) testing. The classes are not particularly well–balanced: the largest class, by area, is dense forest—it occupies 25.5 $\times$ more area than the smallest class (anthropogenic)[2].

Tasks

(1) Create a working environment to build and train the model.

  (a) Make a working project directory by typing `mkdir alps && cd alps` in the terminal.

  (b) Download the segmented data archive by typing `for k in {0..6}; do wget 'https://github.com/dpl10/phytoinformatics2023/raw/main/IGN-data'$k'.tar'; done` in the terminal. Answer question (1).

  (c) Extract the archive by typing `tar -x -M -fIGN-data{0..6}.tar` in the terminal. Three .tfr files should be output.

  (d) Download the model script by typing `wget https://raw.github.com/dpl10/phytoinformatics2023/main/deeplabFirenet.py` in the terminal.

  (e) Download the training and testing scripts by typing `TYPES=('test' 'train'); for TYPE in "${TYPES[@]}"; do wget 'https://github.com/dpl10/phytoinformatics2023/raw/main/'$TYPE'Segmentation.py'; done` in the terminal.

  (f) Make the scripts executable by typing `chmod +x *.py` in the terminal.

(2) Start the Docker instance created for Laboratory 14 by typing `docker run -u $(id -u):$(id -g) --rm -it -v "$PWD:/tmp" -w /tmp tensorflow:2.12.0` in the terminal.'

(3) Construct a randomly initialized DeepLabv3+FireNet model by typing `./deeplabFirenet.py -a 6 -f selu -i 128 -o deepFire.h5 -r 1234567890` in the terminal. Answer question (2).

(4) View the untrained model in netron using the graphical interface or a web browser (https://netron.app).

(5) Make a directory to save the training result by typing `mkdir denovo` in the terminal.

(6) Train the model by typing `time ./trainSegmentation.py -a 6 -b 64 -c -e 48 -f ce+clr+a -i 128 -l 0.01 -m deepFire.h5 -o denovo -r 1234567890 -t IGN-train.tfr -v IGN-validation.tfr` in the terminal. Depending upon your computer, this step will take between 1 minute and 3 hours. Answer question (3).

[1] classes: unknown (0), dense forest (1), sparse forest (2), moor (3), herbaceous formation (4), and anthropogenic (5)

[2] pixel area: (0) 14,219; (1) 45,008,122; (2) 35,484,201; (3) 17,608,687; (4) 9,074,259; (5) 1,764,112

(7) Evaluate the best–trained model by typing `./testSegmentation.py -a 6 -b 64 -g 0 -i 128 -m $(ls -ltr denovo/*/best-model.h5 | awk '{print $9}' | tail -1) -t IGN-test.tfr` in the terminal. Answer question (4).

(8) Close the Docker image by typing `exit` in the terminal.

Questions (https://forms.gle/s64em7v5UQ7xG5zX6)

(1) For task (1)(b), explain what each part of the command does.

(2) For task (3):

  (a) How many parameters does this model have?
  (b) How many parameters are trainable?
  (c) What is the size of the output layer?

(3) For task (6):

  (a) Explain what each part of the command does.
  (b) How long (real time == wall clock time) did it take to train the model?
  (c) What was the model's accuracy and mean IoU on the validation dataset?
  (d) Did the model overfit during training?

(4) For task (7):

  (a) What was the model's accuracy and mean IoU on the test dataset?
  (b) Is this very different from the corresponding values on the validation dataset?
  (c) Would the model benefit from additional epochs of training?

Literature cited

Chen, L.-C., Y. Zhu, G. Papandreou, F. Schroff & H. Adam. 2018. Encoder–decoder with atrous separable convolution for semantic image segmentation. arXiv 1802.02611.

Gupta, K. D., , D. K. Sharma, S. Ahmed, H. Gupta, D. Gupta & C.-H. Hsu. 2023. A novel lightweight deep learning–based histopathological image classification model for IoMT. Neural Processing Letters 55: 205–228.

Guérin, K., E.and Oechslin, C. Wolf & B. Martinez. 2021. Satellite image semantic segmentation. arXiv 2110.05812.

Due by 2:00 PM May 16.