GEOSPATIAL INSIGHT MACHINE LEARNING ENGINEER PRACTICAL TEST

As part of the selection process, we ask all candidates for Machine Learning positions to complete a practical assessment. Your submission will provide a point of discussion during the face-to-face interview.

In this test, you will be asked to perform some basic image processing and machine learning tasks and give an interpretation of the results. The ideal solution will contain one jupyter notebook with the solutions to the tasks, perhaps with some supporting explanations where appropriate.

Take as much time as you need to complete the tasks, in any order, but we'd suggest to spend no more than 2 hours even if you do not complete all the tasks. Please ensure everything you do is presented in your submission, without executing steps elsewhere (say, on the command line or other scripts that are not submitted).

Requirements

We recommend that you use python using Anaconda, which will simplify the installation of some of the libraries you need for this tutorial.

Install the gdal and rasterio packages using:

```
conda install -c conda-forge gdal rasterio=1.0
```

If you have problems installing GDAL, have a look online for solutions depending on your OS.

Datasets

Three sets of data are provided for training a binary classifier:

- **Training.** The following two files can be used as training data:
  - data/train/train_input.tif. An 8-band satellite image. Each band is represented by a separate channel of the input image. Each 8-channel pixel from the image will be an input for training a binary classifier. The satellite bands are as follows (in order):
    - coastal
    - blue
    - green
    - yellow
    - red
    - red-edge
    - near infrared 1
    - near infrared 2
  - data/train/train_target.tif. A single-channel image of the same height and width as the input image. The target image pixel values are all either 0 or 255. The pixels represent binary ground-truth data for some feature of the corresponding pixels of the input image.
- **Validation.** The following two files can be used as validation data:
  - data/validation/validation_input.tif. An 8-band satellite image as for training data.
  - data/validation/validation_target.tif. A single-channel image as for training data.
- **Test.** The following file can be used as test data:
  - data/test/test_input.tif. An 8-band satellite image as for training and validation data.

Tasks

1. **Image processing and visualisation.** For the training input image:
   a) Open the image using *rasterio* and convert it to a numpy array.
   b) Print the array shape and type.
   c) Given that that the array dimensions are in the form (channels, height, width), normalize each channel using min/max normalization. Your normalization code will be reused later.
   d) Visualize the red, green and blue channels of the normalized array as an RGB image.
2. **Logistic regression.**
   a) Using your favourite ML package, train a binary logistic regression on the training images. Your model should take as input a min/max normalized 8-band input pixel and predict the corresponding value of the single-channel target pixel.
   b) How many trainable parameters does your logistic regression model have? Describe your calculation.
   c) Apply your trained model to the normalized validation input image. Use the validation target image to compute a relevant quality measure for your model.
   d) Apply your trained model to the normalized test input image. Visualize the results of the output.
   e) Visualize the trained parameters of your trained model as a 1-d vector. Explain the values and describe how they affect the behavior of your binary classifier?
3. **Convolutional neural network.**
   a) Define a CNN with the following layers:

Layer	Filter size	Number of filters	Activation
1	8 x 3 x 3	32	RELU
2	32 x 3 x 3	1	Linear

   b) How many trainable parameters does your CNN model have? Describe your calculation.
   c) Initialize your CNN with random weights. Do not train your CNN. To ensure model input dimensions are correct, apply the initial randomized model to the normalized training input image.
   d) Suppose another, hypothetical, CNN classifier takes planar RGB images as input. Suppose the input layer contains multiple 3x11x11 convolution filters. After training, the filters of the input layer are visualized and one of them appears as:

In general terms, describe the role for which this filter has become trained.