Image Interpretation – Assignment 3

The aim of this exercise is to create a model that automatically labels crop types from optical satellite (in this case Sentinel-2) image time series. Your model (classifier) will be trained in a supervised manner with a provided training set. Then, you will assess the performance of the model (accuracy, training/inference runtime, memory requirements) and generate a map with model's prediction for the test region. You will work with single-pixel time series from *ZueriCrop* dataset [1]. The temporal length of the input data is 71 which covers Sentinel-2 Level-2A bottom-of-atmosphere reflectance between January and December 2019. Each time series contains 4 bands with a ground sampling distance (GSD) of 10 meters (B2=Blue, B3=Green, B4=Red, B8=Near Infrared). In the provided dataset there are 13 different crop types: Meadow, Winter wheat, Maize, Pasture, Sugar beets, Winter barley, Winter rapeseed, Vegetables, Potatoes, Wheat, Sunflowers, Vines, Spelt. For a given test example, your classifier should assign one of these crop classes.

The problem will be solved in three different groups:

1. The first group should implement and train a recurrent neural network (RNN) that receives an input single-pixel time series and predicts the corresponding class.
2. The second group should implement and train a temporal convolutional network (TCN) that receives an input single-pixel time series and predicts the corresponding class.
3. The third group should study more conventional machine learning algorithms. The features should be extracted using conventional techniques, e.g computing Normalized Difference Vegetation Index (NDVI). Random Forest classifier (RF) and Hidden Markov model (HMM) should be implemented and trained.

Additional Information:
Training and test datasets are provided in two separate HDF5 files. They contain (i) data with the size $N \times T \times C$, where N is the number of pixels, T is the temporal length, and C is the number of channels and (ii) corresponding labels with the size N.

Note that test set is not a validation set, it should **not** be used for model selection. Each group should divide the training dataset into training and validation on its own. Only after training the model should be evaluated on the test data. For evaluation, each group must create a confusion matrix to capture correct and incorrect classifications. The confusion matrix should be used to quantify the performance of the model using appropriate metrics e.g overall accuracy.

Provided scripts:

1. Pytorch dataloader
2. Visualizer code for generating map from predictions

Submitting Results:
To submit the results, each group should create a Git repository (e.g., Gitlab/Github) that contains their code. The Git repository should **not** be used to only deliver the result (only a single commit), but also to facilitate collaboration and provide an overview of the development process.

In addition to the code, each group must write a technical report of 3-4 pages. The report should include the following:

- A description of the chosen method including training details, how data is handled and so on.
- Quantitative and qualitative (i.e generated crop map on the test region) performance analysis.
- Discussion on results e.g why certain crops are difficult to predict by the model.
- Discussion on the limitation of the method for a given task.
- References

The technical report may discuss failed approaches if they help to show the decisions that led to the final method.

All submissions should be sent to ozgur.turkoglu@geod.baug.ethz.ch. You can also send an invitation for the Git repository to this email.

The deadline for submission is 15 December 23:59.

In-class presentation will take place on 16 December at 08.50.

Although you may use any framework to solve this exercise these libraries are **highly recommended**:

- Python + virtualenv / Anaconda
- jupyter / colab
- h5py for data access
- PyTorch / Tensorflow for neural networks
- XGBoost / LightGBM for fast training of gradient boosted decision trees.
- scikit-learn for conventional machine learning, feature extraction and model selection

A few hints and typical pitfalls:

- Find a way to create an appropriate validation set for training.
- Testing (evaluation on dataset_testset.h5) is performed only once - after the whole training process is completed.
- Find appropriate metrics and and a way to track them. Prototyping new models only makes sense if you can measure the improvement.
- Perform analysis for training/inference time and memory requirement.
- Be aware that you can subsample the temporal dimension, it would be very useful to speed up development.
- Be aware that test set also has *Unknown* class which is kept for visualization purpose.
- You may create a very simple baseline before using larger models - Develop your actual model against this baseline.
- Check your predictions qualitatively.
- You can use only a small portion of the training data when developing a model to speed up development.

References

[1] M. O. Turkoglu, S. D'Aronco, G. Perich, F. Liebisch, C. Streit, K. Schindler, and J. D. Wegner, "Crop mapping from image time series: deep learning with multi-scale label hierarchies," *Remote Sensing of Environment*, 2021.