1. Why Machine Learning for processing data of Copernicus Satellite Sensors?

- The amount of data from remote sensing satellites that has to be processed, dramatically increased in the recent years, especially with the Copernicus program.
- The processing is even further challenging since there are near real time (NRT) requirements for many products.
- Therefore, the retrieval algorithms not only have to be accurate, but also very fast aswell.
- In recent years, the application of machine learning techniques, especially neural networks, has become increasingly popular in order to improve the performance of classical retrieval algorithms.
- A successful example is the use of neural networks for the retrieval of the operational CLOUD product of the Sentinel-5 Precursor satellite (S5P).

2. Inversion with a radiative transfer model vs inversion with a neural network

- Atmospheric retrieval can often be formulated in terms of mathematical inversion problems.
- The goal is to find a set of parameters $\mathbf{x}$ that minimize the residual $\mathbf{r}(\mathbf{x}) = \mathbf{y}_\text{obs} - \mathbf{y}_\text{mod}$ between a known vector $\mathbf{y}$ and the mapping of the parameters $\mathbf{F}(\mathbf{x})$, where $\mathbf{F}$ is a predefined function.
- In the case of atmospheric retrieval algorithms $\mathbf{x}$ represents the state of the atmosphere, $\mathbf{y}$ is a measured spectrum and $\mathbf{F}$ is a radiative transfer model (RTM) that predicts the spectrum $\mathbf{F}(\mathbf{x})$.
- For the inversion algorithm the specific implementation of $\mathbf{F}$ is not relevant: it can be a complex RTM or a fast neural network (NN).
- GOODRT (GOAL Direct RTM) is an example for an inversion algorithm with a RTM as a forward model.
- It produces the S5P ozone total column product, is computationally very expensive but has no NRT requirements.
- ROCCIN (Retrieval Of Cloud Information using Neural Networks) is an example for an inversion algorithm with NNs as forward models.
- It is part of the S5P CLOUD product and has strict NRT requirements.

3. How to get from a radiative transfer model to a neural network

- In order to replace the RTM of an inversion algorithm by a NN a general method was developed which is applicable to arbitrary RTMs and thus can be used for many retrieval algorithms.

   Training a NN from the RTM
   
   1. Smart sampling
   
   2. RTM
   
   3. Scaling of the data
   
   4. Training of the NN
   
   5. Validation of the NN
   
   6. Deployment of the NN

- It consists of the following steps:
  1. Smart sampling: The training data needed for the NN consists of input/output pairs. In case of the ROCCIN algorithm the input consists of up to seven parameters:
   - Surface parameters (surface height, surface albedo)
   - Geometry (solar zenith angle, viewing zenith angle, relative azimuth angle)
   - Cloud properties (cloud height, cloud optical thickness) - in case of cloudy scenes.
   - Samples of this input space are chosen with the Halton sequence. Additionally, importance sampling can be used to account for the distribution of the different parameters.
  2. Generation of the training data:
   - The corresponding outputs are generated using the RTM.
   - For ROCCIN, these are spectra in the O$_2$ A-band, calculated by the RTM VLIDORT.
   - The final results are then saved together with the inputs in a HDF-CDF file.
  3. Scaling of the data:
   - Inputs and outputs of the training data are scaled to the interval [0,1] to improve the stability of the weights during the training process.
  4. Training of the NN:
   - Tools based on keras were implemented which allow:
     - Easy definition of the network topology, activation functions and training parameters.
     - Saving of the network as well as metadata in a hdf5 file.
     - Iterative training by loading of pre-trained networks.
  5. Validation:
   - After the training, the NN is validated with an independent data set.
  6. Deployment of the NN:
   - A neural network module was developed which implements:
     - Reading of NNs defined in hdf5 files at runtime.
     - Transparent scaling of inputs and outputs.
     - Computation of the derivatives.
     - Support of arbitrary network topologies and different activation functions.

5. Application

- Neural networks for SSP already used for operational CLOUD product.
- Neural networks for S4 implemented in first version of CLOUD processor.
- Current clear-sky neural network for S4 has slightly better performance than clear-sky network for SSP, cloudy network for S4 needs further improvement.
- Different NNs for different scene types and cloud models are in use (clear-sky, cloudy CRD, cloudy CAL). Development for other cloud types (ice-clouds) is ongoing.

4. Evaluation

- Finding a neural network with the optimal performance is very challenging and depends on the input (many aspects impact the results).
  1. Scaling of the data.
  2. Topology of the network.
  3. Activation Functions of the neurons.
  4. Sampling of the parameters for the training set.
- Other aspects not yet investigated: Optimizer + learning-rate, number of samples, batch size.
- For the evaluation of Sentinels-4 cloud-sky spectra (5 input, 345 output parameters) were used.
  1. Scaling:
   - Scaling both, inputs and outputs, is necessary.
  2. Topology:
   - Selecting the network topology is hard as it is very arbitrary (key factors: complexity, number hidden layers).
  3. Activation Functions:
   - Activation functions are responsible for nonlinearity of the NN.
     - The following activation functions were evaluated: relu, sigmoid, tanh.
     - Activation with tanh is the preferred activation function: best relative error (Figure 4.3), smoothest, no discontinuities (Figure 4.4).
  4. Sampling:
   - Parameter distribution in training set is significant.
   - Importance Sampling.
   - Comparison for surface height with actual vs. Uniform distribution.
- Despite worse validation loss during training, NNs trained with actual parameter distribution lead to better results.

6. Conclusion

- Neural networks offer a way to drastically increase the performance of classical retrieval algorithms.
- Many orders of magnitudes faster than RTMs, successful use in CLOUD product for SSP (operational) and S4.
- Neural network lifecycle chain offers general procedure for replacing RTM by NN.
- Allows use of specific NNs for specific problems (e.g. for different types of clouds).
- Neural networks can provide sufficient accuracy to replace RTM.
- Finding best structure configuration is challenging, approaches to evaluate and determine well suited structures have been presented, more investigations are ongoing.
- Neural networks allow for new possibilities for inversion algorithms.
- Computational performance allows many forward model calls, gradients available.
- Possibility for global optimization techniques.

**Figure 1.1**: cloud fraction (CF), cloud top pressure (CTP) and the cloud optical thickness (COT) from the operational S5P CLOUD product from 08-08-2019.

**Figure 2.2**: The solver can either use a RTM or a NN as a forward model.

**Figure 3.1**: Illustration of the complete NN lifecycle - from data sampling to deployment.

**Figure 4.1**: Relative errors of NNs trained with input and/or output scaling or no scaling at all.

**Figure 4.2**: Validation losses during training for different NN topologies.

**Figure 4.3**: Relaxed representation of the relative root mean square errors for NNs with different activation functions.

**Figure 4.4**: Mean spectra radiance depending on the cosine of the relative azimuth angle for NNs with different activation functions.

**Figure 4.5**: Surface distribution for the best training set.

**Figure 4.6**: Validation loss during training for NNs with different training sets.

**Figure 4.7**: Relative errors of NNs with different training sets.

**Table 1**: Computational performance for generating 250000 spectra.