Using Machine Learning for processing Big Data of Copernicus Satellite Sensors at the Example of the TROPOMI / Sentinel-5 Precursor (S5P) and Sentinel-4 Cloud Product

Deutsches Zentrum für Luft- und Raumfahrt
German Aerospace Center

Fabian Romahn[1], Diego G. Loyola[1], Athina Argyrouli[2], Victor Molina Garcia[1], Jian Xu[1], Ronny Lutz[1], Robert J D Spurr[3], Sebastian Gimeno Garcia[4], Mattia Pedergnana[5]
(1) German Aerospace Center (DLR), (2) Technical University of Munich (TUM), (3) RT Solutions Inc., (4) EUMETSAT, (5) formerly German Aerospace Center (DLR)

EGU General Assembly

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 have not only 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) – in the upcoming Sentinel-4 (S4) CLOUD they will be used aswell

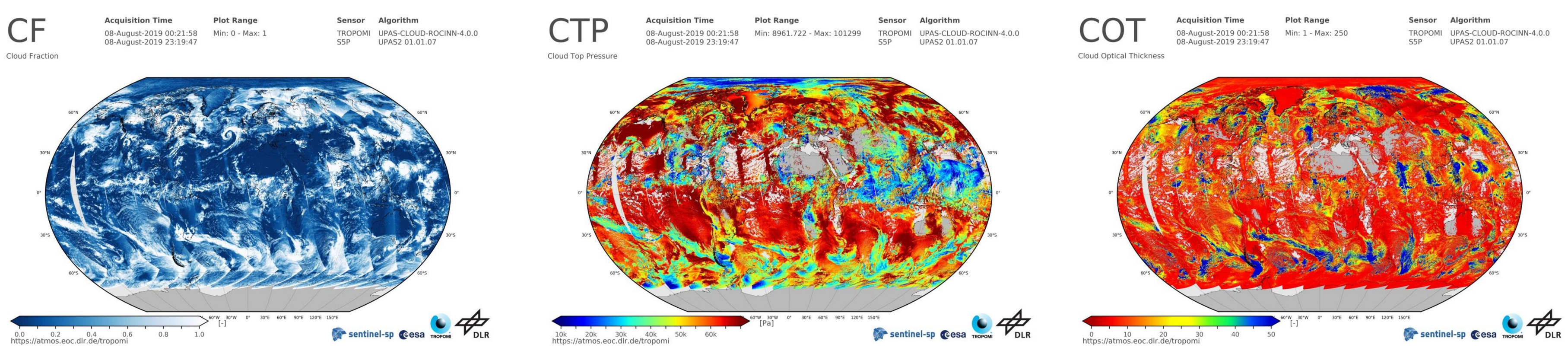

Figure 1.1: cloud fraction (CF), cloud top pressure (CTP) and the cloud optical thickness (COT) from the operational S5P CLOUD product from 08-08-2019

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
- There, the goal is to find a set of parameters x that minimize the residual $\|F(x) - y\|_2$ between a known vector y and the mapping of the parameters $F(x)$ - where F is a predefined function
- In the context of atmospheric retrieval algorithms x then represents the state of the atmosphere, y a measured spectrum and F a radiative transfer model (RTM) that predicts the spectrum $F(x)$
- For the inversion algorithm the specific implementation of F is not relevant – it can be a complex RTM or a fast neural network (NN)
- **GODFIT** (**GO**ME **D**irect **FIT**ting) is an example for an inversion algorithm with a RTM as forward model
  - It produces the S5P ozone total column product, is computationally very expensive but has no NRT requirements
- **ROCINN** (**R**etrieval **Of C**loud **I**nformation using **N**eural **N**etworks) 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

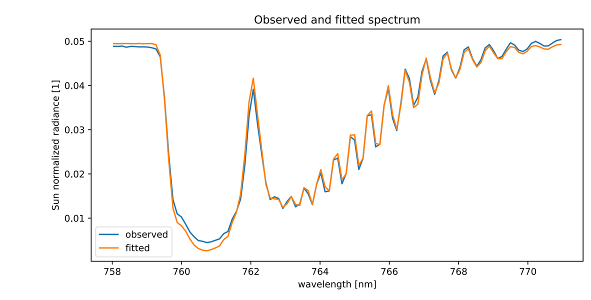

Figure 2.1: Example of an observed and fitted spectrum in the O_2 A-band – the fitted spectrum is a linear combination of a clear sky- and fully cloudy spectrum weighted by the cloud fraction

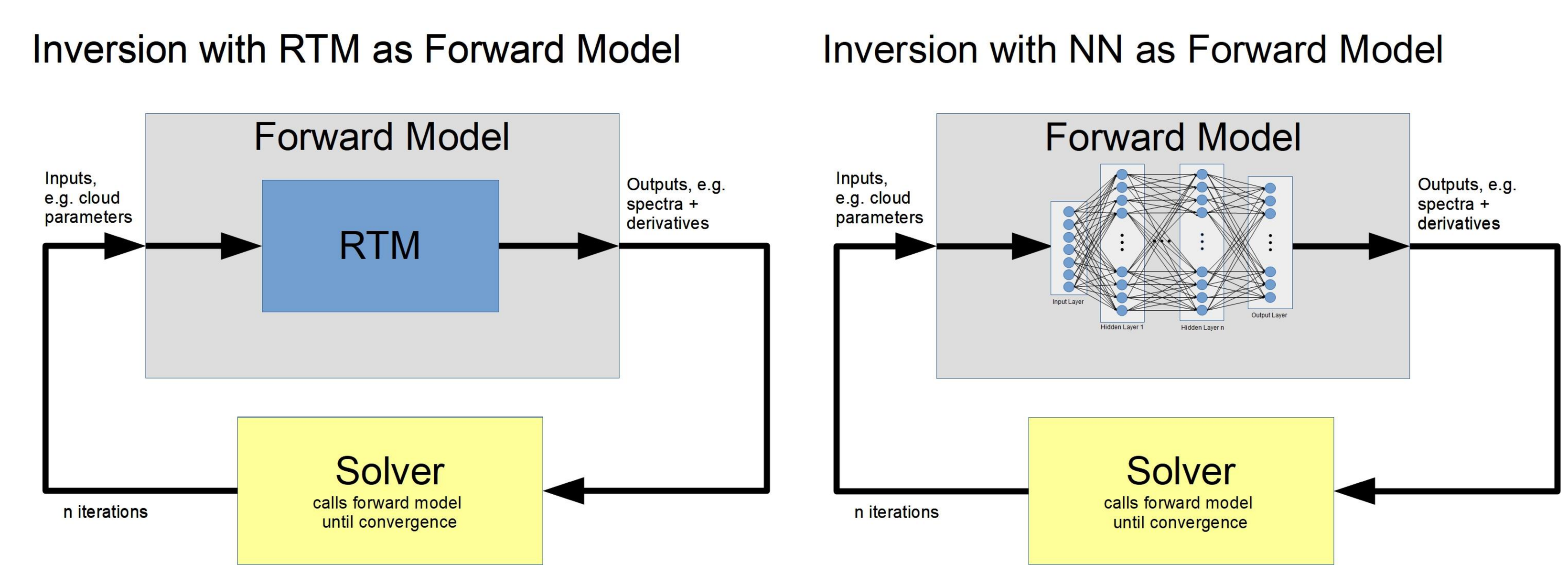

Figure 2.2: The solver can either use a RTM or a NN as forward model

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

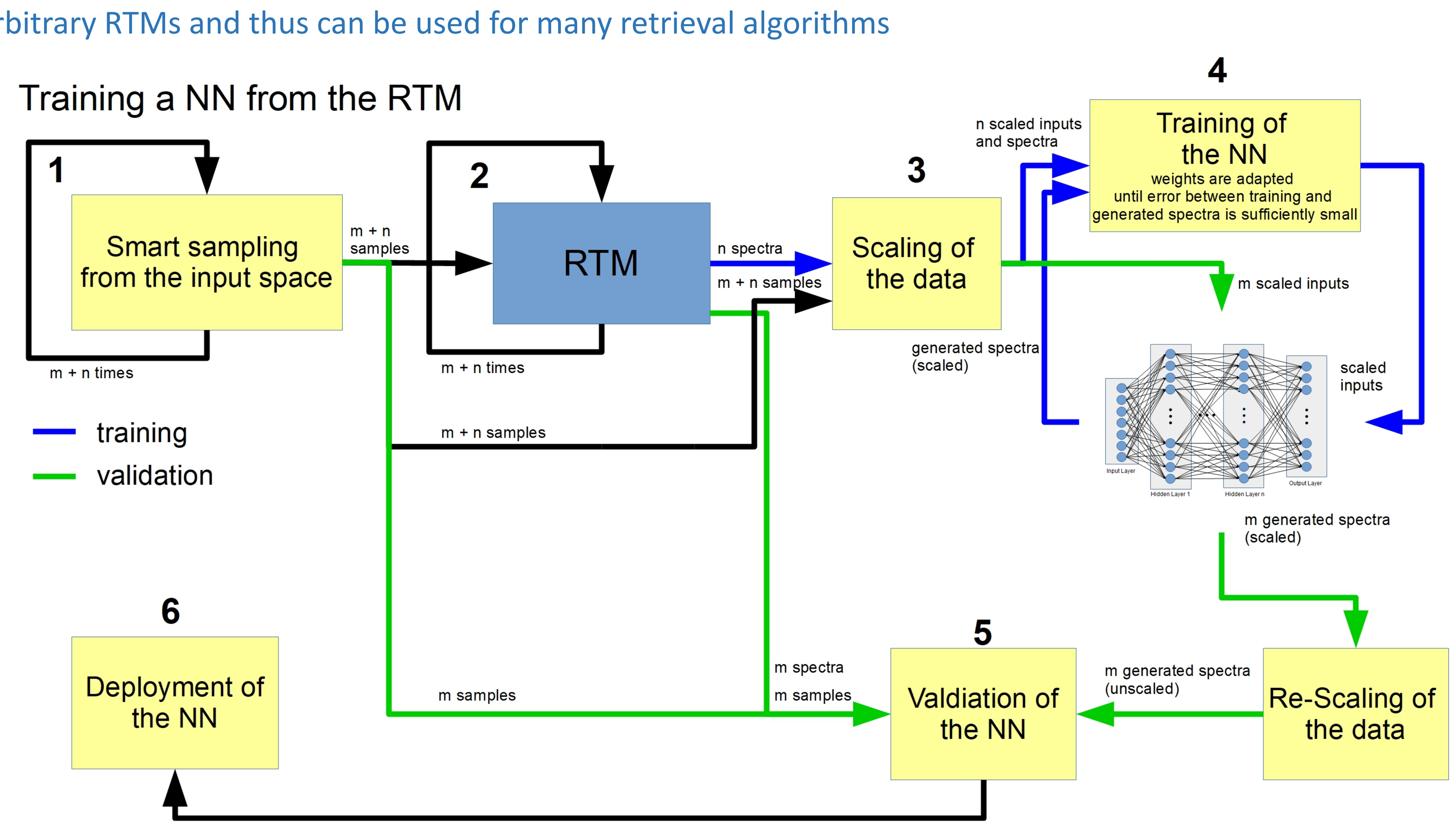

Figure 3.1: Illustration of the complete NN lifecycle – from data sampling to deployment

- It consits of the following steps:
  1. **Smart sampling:**
     The training data needed for the NN consists of input / output pairs. In case of the ROCINN algorithm the input consists of up to seven parameters:
     - Surface parameters (suface 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 ROCINN, 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 netCDF-4 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 an hdf5 file
     - iterative training by loading of pre-trained networks
  5. **Validation:**
     After the training, the NN is validated with an indenpendent 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

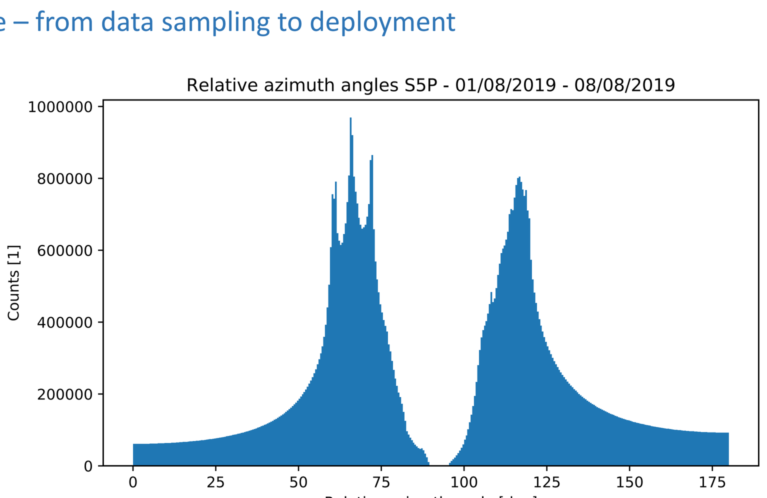

Figure 3.2: Histogram of the relative azimuth angles (RAA) from the S5P data of several days

4. Evaluation

1. **Precision**
   - Finding a suited structure of the NN is challenging
   - In the context of regression problems, only fully connected feed forward networks are considered
   - Then, the precision depending on the complexity (i.e. number of parameters) and the depth (i.e. number of hidden layers) was measured:
   a. **Complexity**
      - NNs with one hidden layer and a varying number of neurons were trained to generate clear sky spectra
      - Figure 4.1 shows that if the model is too simple it cannot reproduce the spectra correctly – however if it is too complex the precision slightly decreases again (overfitting)
      - The optimal number of parameters for this problem seems to be at between 20000 and 30000
      - Additionally, it can be seen that the scaling of the data offers a significant benefit to the precision
   b. **Depth**
      - NNs with an approx. fixed number of parameters and a varying number of hidden layers were trained
      - Figure 4.2 shows that NNs with three hidden layers are optimal

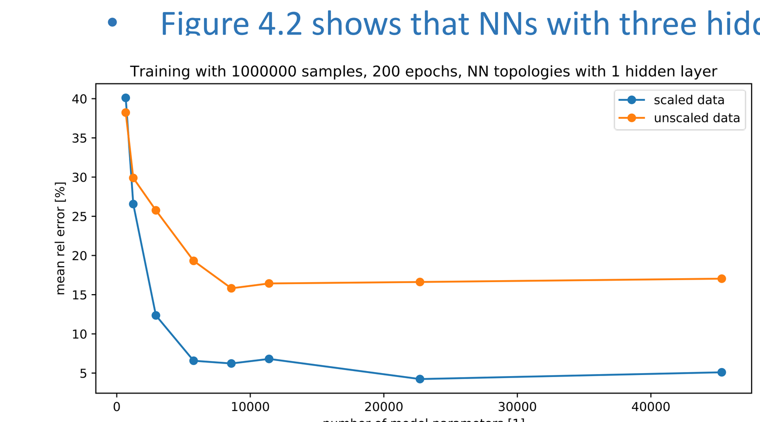

Figure 4.1: Precisions of NNs depending on the complexity and scaling of the data

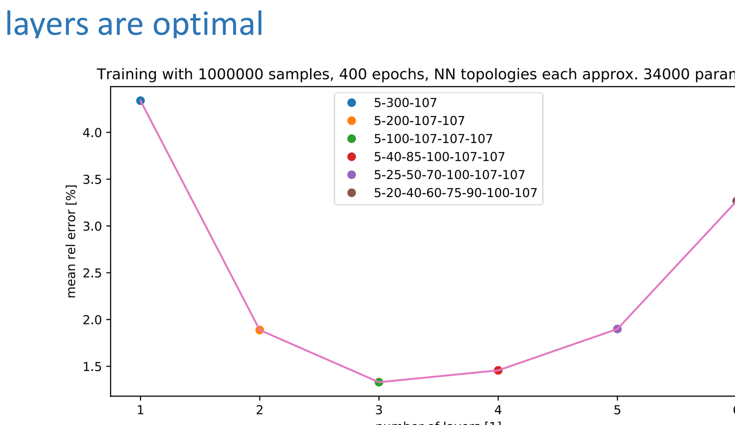

Figure 4.2: Precisions of NNs depending on the number of hidden layers

2. **Performance**
   - Table 1 shows that the execution time of NNs for generating 100000 clear sky spectra is about 6 to 7 orders of magnitude faster compared to the RTM
   - The calculation of the gradient (needed for the inversion) decreases the execution time by a factor of about 5 to 6 but it is still at least 5 orders of magnitude faster than the RTM (without gradient)
   - The execution- and training time of the NN increase with it's complexity

Forward Model	# parameters	exec. time (100000 spectra)	exec. time (100000 spectra) with gradient	training time	mean abs rel. error
RTM (VLIDORT 2.7) with 32 threads	-	6h 10m 59.801 s	-	-	-
NN clear sky (5, 5, 107)	672	0.05 s	0.37 s	17m 35s	40.11 %
NN clear sky (5, 200, 107)	22707	0.45 s	2.36 s	29m 39s	4.24 %
NN clear sky (5, 1000, 107)	113107	2.05 s	10.11 s	1h 01m 39s	5.10 %
NN clear sky (5, 200, 107, 107)	34263	0.64 s	3.92 s	38m 51s	1.89 %
NN clear sky (5, 100, 107, 107, 107)	34519	0.69 s	4.85 s	40m 13s	1.33 %
NN clear sky (5, 40, 85, 100, 107, 107)	34688	0.67 s	4.86 s	41m 35s	1.46 %
Operational NN clear sky (5, 100, 100, 107)	21507	0.62 s	3.48 s	n/a	2.79 %
Operational NN fully cloudy (7, 100, 100, 107)	21707	0.66 s	4.34 s	n/a	3.56 %

Table 4.1: Comparison of the RTM and different NNs regarding the execution- and training time as well as the precision (measured on Intel(R) Xeon(R) Gold 6152 CPU @ 2.10GHz with 88 cores)

3. **Operational application**
   - For the operational S5P CLOUD product, NNs for generating clear sky- and fully-cloudy spectra with structures of (5, 100, 100, 107) and (7, 100, 100, 107) are used
   - As can be seen in Figure 4.3, the clear sky NN performs better which is due to the reduced input space (5 vs 7 parameters)
   - Together with the previous results it can be seen that there is potential in improving the current NNs

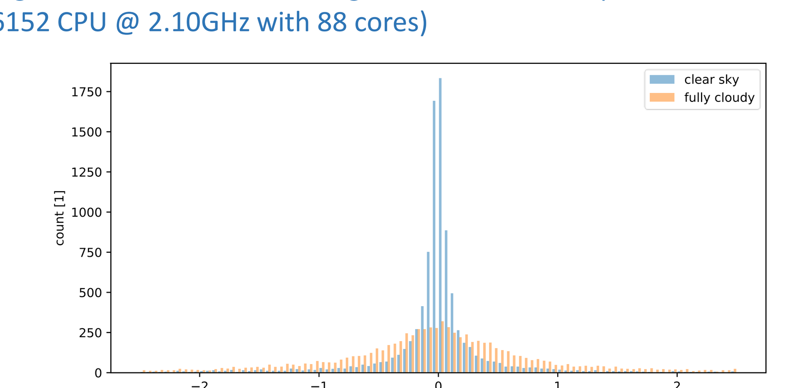

Figure 4.3: Relative errors of the clear sky and fully cloudy NNs of the operational S5P CLOUD product

5. Limitations

- NNs can have difficulties in handling discontinuities of the RTM function
- This can lead to unexpected results:
  - In Nadir scenes with VZA = 0° the spectrum is independent of the relative azimuth angle (RAA)
  - This is correctly modeled by the RTM
  - In the NN however, there is still a dependency of the RAA (Figure 5.1)
  - The NN spectra has always a jump in Nadir
  - This jump is then further propagated to the retrieved parameters
  - Solutions to this problem are:
    - Training of a separate NN for the Nadir region

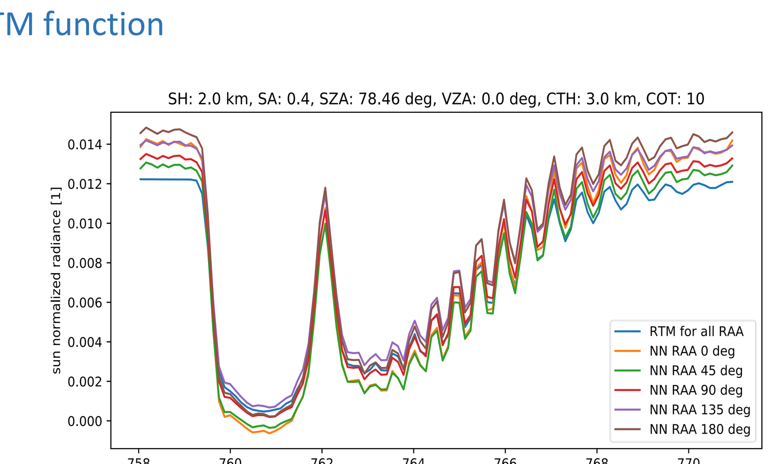

Figure 5.1: Generated spectra of the RTM and NN in a Nadir scene – the RTM spectra are identical for all RAA while the NN spectra differ

6. Conclusions

- Neural networks offer a way to drastically increase the performance of classical retrieval algorithms by being several orders of magnitude faster compared to RTMs
- Depending on the structure they can provide sufficient accuracy to replace the RTM - however finding an appropriate structure can be challenging
- Approaches to evaluate and thus determine the final structure have been presented
- The presented NN lifecycle chain offers a general way to replace a RTM with a NN
- Since NNs can have difficulties in handling discontinuities of the RTM function, unexpected side effects can occur (Nadir scenes) which can lead to further problems
- NNs instead of RTMs are being used successfully in the operational S5P CLOUD product
- NNs for the operational S4 CLOUD product are currently in development