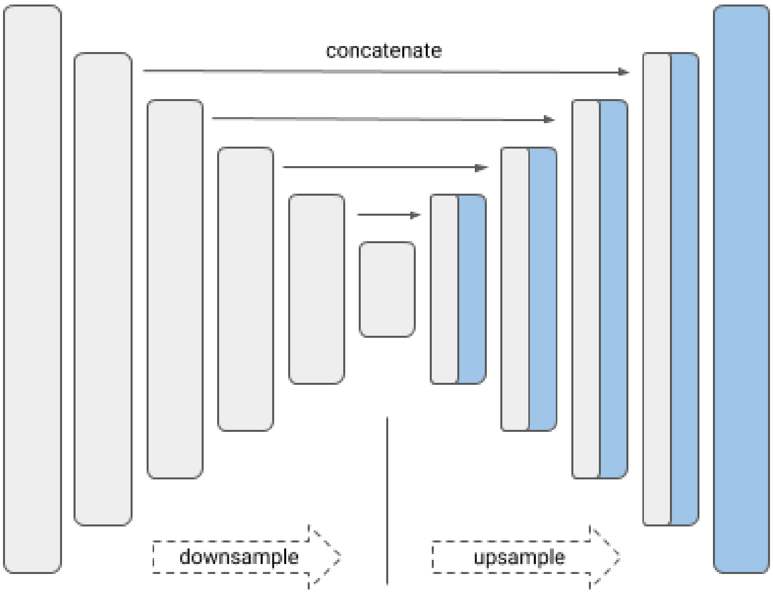
Cloud segmentation project

Matej Cief, Tomas Mizera

Checkpoint

Overview

We use Unet for predicting masks based on input images (data analysis can be found in `analysis/data_analysis.ipynb`) with backbone `resnet` that extracts features and passes it to Unet.



Challenges & solutions

While working on project we came across several challenges:

- run-length encoding (as described in analysis), labeled data provided from kaggle and had to somehow transfer this encoding to mask image. We found several functions that transforms this encoding to images and used them.
- data streaming to model preventing memory overflow. Generator class was introduced - it also transforms data (e.g. run-length encoding to image and so on.)
- we also spent nice amount of time searching for possibilities to predict mask, not only category of clouds (not 4 output neurons, but entire convolutional layer)

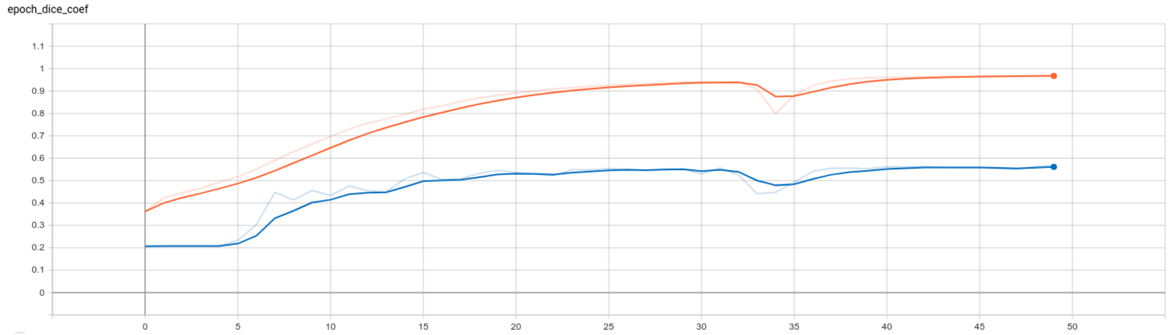
Files

- `main.py` - model definition and training
- `data/generator.py` - code for Generator class handling data manipulation and streaming to model
- `models/util.py` - does dice coef

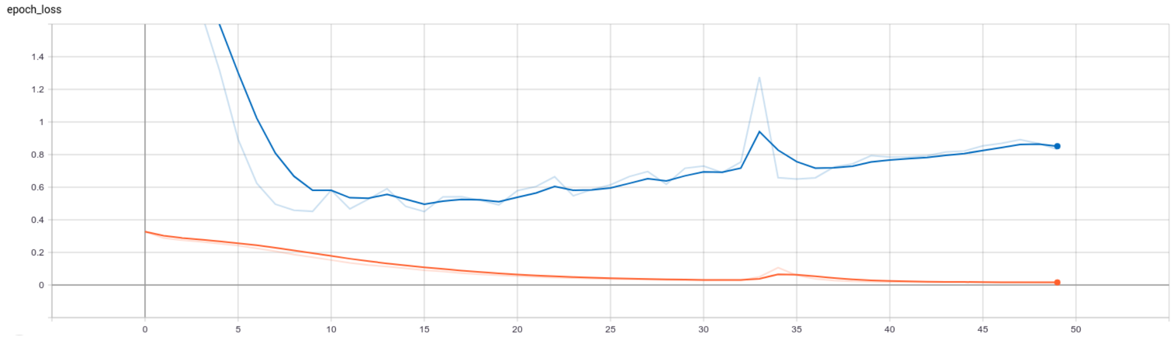
First complete run

Even though we were able to run more runs, our first big run had 50 epochs and these are results:

Legenda: Validation data - Train data



Model accuracy on 50 epochs



Model loss on 50 epochs

We can see that current run stagnates at 20-25 epochs and after that loss function starts to arise. It would make sense to stop training after this amount of epochs - this feature will be implemented in final submission.