Project for Deep learning in medical imaging: Segmentation - MIC

Task 5: Fine tuning

Tobias Dorra, Hendrik Schick

February 5, 2020

1 Task

The task was to fine tune the Implementation.

2 Implementation

This week, we found a bug in the method, that is responsible for building the U-Net keras model from the parameters that the *Experiment Planning and Preprocessing* script from nnU-Net returned. This meant, that the model was missing one "U-Net stage" all the time. This is why this week we fixed that bug, retrained the model with the fixed architecture and evaluated the newly trained model again.

The new model architecture is now finally to big to show it on one Din-A4 page in this report. It is attached to the moodle submission as an image file, though.

When looking at the loss values, the performance of the model seems to have improved: While the old model had a final testing loss of 0.0148, the final test loss for the new model is 0.0080. Figure figure 1 shows, how the loss changes during the training. What is interesting in there is, that there is not too much of a difference between the training- and test loss any more. So, either, the new model architecture generalizes much better, or there is a problem with seperating the training- and test- dataset.

The new average values for the intersection over union for the individual classes are the following:

Class 0 (outside): 0.99819249

Class 1 (liver): 0.90915714

Class 2 (cancer): 0.33424284

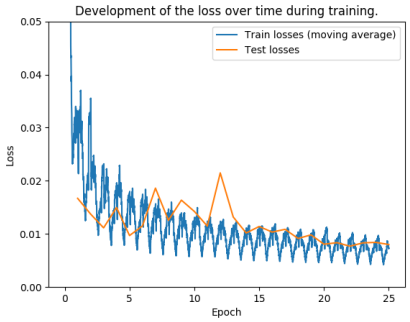

Figure 1: Development of the loss during training

So, detecting the liver now works reliably. This is also confirmed, when looking at the actual images of the predicted segmentations.

Detecting the cancer still works not that well. It can be expected, that the IOU for the cancer is not as high as the IOU for the other two classes, because the regions with cancer are usually quite small. So if the prediction is just of by a few voxels, this already decreases the IOU score a lot. However, when looking at the individual images, for three out of the ten test images, the detected cancer does not even overlap with the real cancer at all (IOU=0). Those three images are also the three images in the test set, that have the smallest cancer regions. So the network might have problems detecting small tumors.

Figure 2 shows a data case where the prediction of the cancer worked well.

Figure 3 shows a data case where the prediction worked not so well.

We also counted, how many voxels of which true class got (mis)classified as which other class. Here are the numbers for the two examples from figure 2 and 3:

For figure 2:

True class	**Predicted class**	**Number of voxels**
0 (outside)	0 (outside)	6.534.266
0 (outside)	1 (liver)	5.467
0 (outside)	2 (cancer)	62
1 (liver)	0 (outside)	11.075
1 (liver)	1 (liver)	174.966
1 (liver)	2 (cancer)	4.650
2 (cancer)	0 (outside)	474
2 (cancer)	1 (liver)	2.146
2 (cancer)	2 (cancer)	26.894

For figure 3:

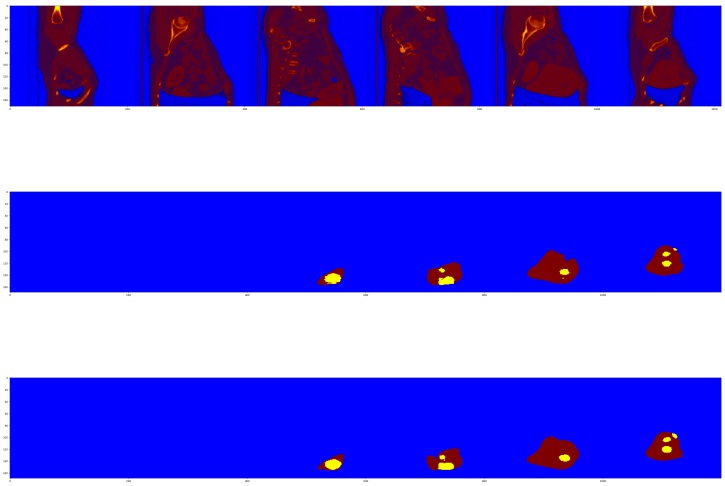

Figure 2: Good result
Top to bottom: input, ground truth, prediction
IOU values for this specific data case: 0.9973, 0.8823, 0.7857

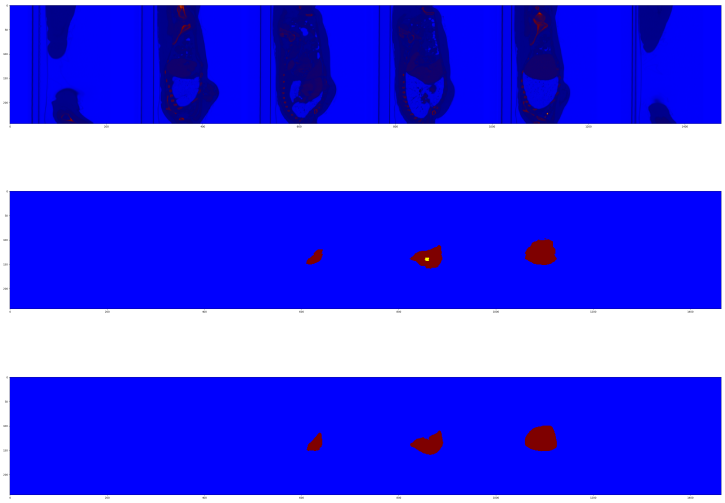

Figure 3: Bad result
Top to bottom: input, ground truth, prediction
IOU values for this specific data case: 0.9983, 0.8666, 0.3154

True class	Predicted class	Number of voxels
0 (outside)	0 (outside)	14.229.682
0 (outside)	1 (liver)	16.385
0 (outside)	2 (cancer)	0
1 (liver)	0 (outside)	6.853
1 (liver)	1 (liver)	154.104
1 (liver)	2 (cancer)	4
2 (cancer)	0 (outside)	0
2 (cancer)	1 (liver)	467
2 (cancer)	2 (cancer)	217

3 Problems and future tasks

We still need to do the same evaluation procedure for the prostate dataset, but we can reuse our existing code for that, so it should be done pretty quickly and easily.