Ischemic Stroke Lesion Segmentation using MRI brain images: Progres Report I 23-06-18

Carlos Uziel Perez Malla
s1781011

Abstract

This document is the first progress report on the master dissertation with title "Ischemic Stroke Lesion Segmentation using MRI brain images".

1. List of completed tasks

1. Setup environment for DeepMedic
2. Pre-process ISLES2017 data
3. Augment ISLES2017 data
4. Prepare transfer learning
5. Run first experiments

2. Summary of completed tasks

2.1. Setup environment for DeepMedic

The first step was to download DeepMedic repository[1] and install the necessary packages dependencies in a virtual environment. After that, I was successfully able to run the provided examples in said repository, both in CPU and GPU. Also, the directory structure of the whole project was defined and a git repository for it was set up[2].

2.2. Pre-process ISLES2017 data

In order to feed ISLES2017 data to DeepMedic, it needs to be accordingly pre-processed, following the instructions provided in the README. It is worth mentioning that images of ISLES2017 are already co-registered per subject. Each image was resampled to have a voxel size of 1mm*dF, where dF is the downsampling factor and goes from 0 to 1.0. It was necessary to downsample the images since, due to memory constraints, it was not possible to train more than two out of six of the given MRI channels.

Additionally, since the masks per subject are not part of the data, they had to be computed, since DeepMedic takes advantage of them. Once the masks were available, they were used to select the data within it for each channel and normalize it with zero mean and unit variance, as recommended in the README.

[1]https://github.com/Kamnitsask/deepmedic
[2]https://github.com/CarlosUziel/ischleseg

All of this was mainly done with the nibabel and nilearn python libraries. Figure 1 shows an example of the results of this process. Please note that background is not shown black due to normalization (0 is no longer the minimum value).

2.3. Augment ISLES2017 data

In the IPP report, it was mentioned that data augmentation could be useful for this task. Of the three alternatives proposed, sagittal axis reflection, elastic deformations and intensity variance, only the last one was implemented. Sagittal axis reflection was already present in DeepMedic, most likely a recent inclusion since DeepMedic is still being actively developed by its creator. After careful examination with Maria, elastic deformation did not make sense for this concrete problem.

Intensity variance consists of selecting the data within the ground truth labels of each subject and randomly vary the intensity values. The new values follow a gaussian distribution, whose mean and standard deviation are equal to the ones computed from the original intensity values. For each subject, a clone was created, being the only difference the intensity values within the ground truth labels. However, this was implemented in a way such that any number of clones can be added to the dataset.

This augmentation is done offline, since performing it online would have required to modify the internal code of DeepMedic, which could have been very problematic.

2.4. Prepare transfer learning

As suggested by Maria given the computational constraints the project was facing, a transfer learning approach was also developed. This simply involved taking the already trained model with channels ADC, CBF and MTT and train it again with the channels ADC, CBV and MTT. Luckily, DeepMedic offers a straightforward way of performing transfer learning so this was very easy to configure, nothing had to be implemented

2.5. Run first experiments

After all the above tasks were implemented, four different models were trained. First, one for each type of data, original pre-processed and pre-processed augmented. Second, transfer-learning was performed for each of the two aforementioned models. Results are shown in figures 2, 3, 4 and 5. It can be seen that data augmentation improves results substantially, whereas transfer learning doesn't seem to improve the results. It is also observable that there is an important overfitting factor, which may justify why adding more data yields much better results.

However, since this was the first time the models were trained fully, after analyzing the results a few errors were found. Concretely, the masks were incorrectly computed in some cases cropping the brains of a small portion of the number of subjects. Also, because of the way the down-sampling was being performed, it was not straightforward to up-sample the images again, causing multiple problems for post-processing. These errors should be fixed by now, and the new results are shown in 6, 7, 8 and 9. In this case, it is possible to observe some discontinuities in the graphs due to the fact that the training had to be stopped and resumed due to unexpected problems. Since DeepMedic stores the model at each epoch, if the training failed at any point during an epoch, the information of that epoch's progress would remain in the log, even if it would be recomputed when the training was resumed from the last saved epoch. Models are being run again to get rid of this noise.

It is worth mentioning that it is clear how the more robust pre-processing has positively affected the results. These new results also show that transfer-learning indeed improves results substantially when not using data augmentation. And that transfer-learning produces better results than data-augmentation. However, as mentioned earlier, these results are still not completely error-free, so should be taken with caution.

In order to obtain more robust results, ten randomly selected train and validation tests have been generated to run DeepMedic in all of them and be able to average the results, which will return a better estimate of the real performance of the system.

As part of this task, a small, still in development script is being coded for showing the final segmentation results in a visual way, comparing these results with the original images and labels.

3. Additional comments and further work

It was stated in the IPP report that part of this project was concerned with modifying the architecture of DeepMedic. However, after further discussion with Maria, this idea was discarded since that would have required more time that is available for a master dissertation. As a result, it was agreed that the focus should be on processing the data before and after the training process, thus treating the network as a black box.

From now on, and considering that there's always a possibility that more errors of the already implemented parts arise and slow down progress, this is the list for the planned tasks that are left:

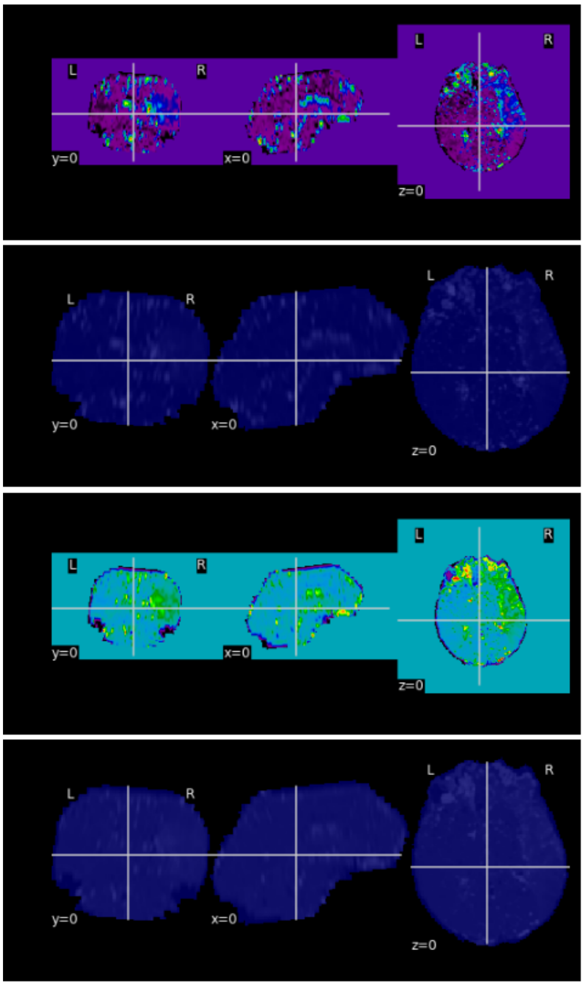

Figure 1. Example of Tmax (top) and TTP (bottom) channels of the same subject after the pre-processing pipeline. Masks are included.

1. Run all models again
2. Plot results and draw conclusions and/or possible improvements
3. Compute FA and MD from 4DPWI for postprocessing
4. Try another dataset

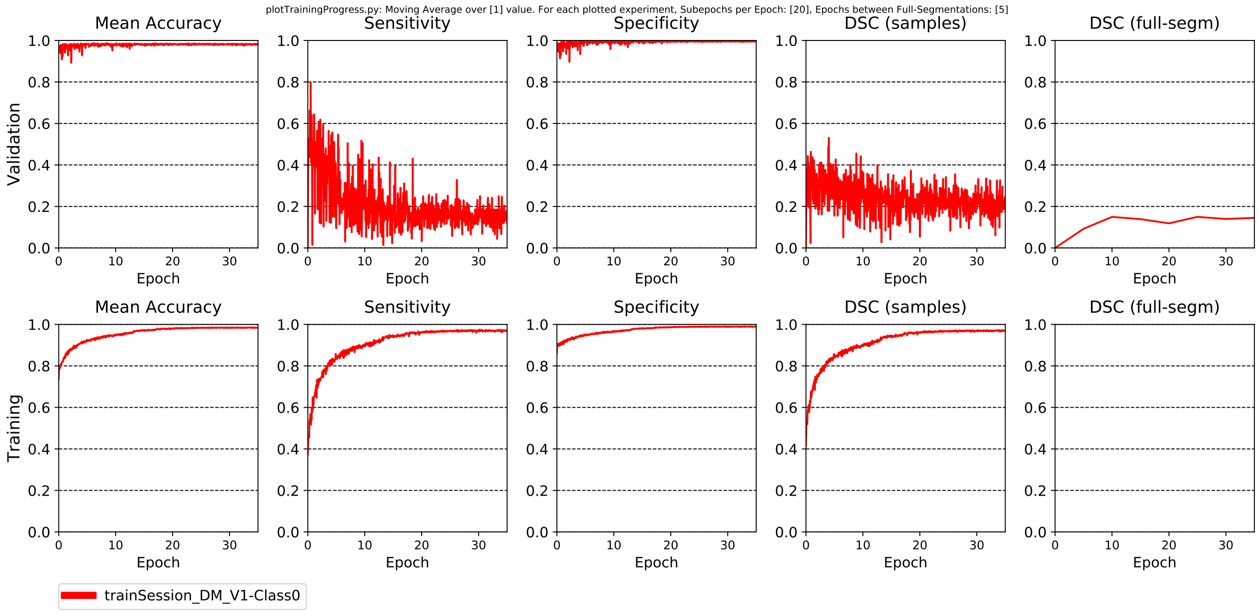

Figure 2. Plotted results of running DeepMedic with the original data pre-processed.

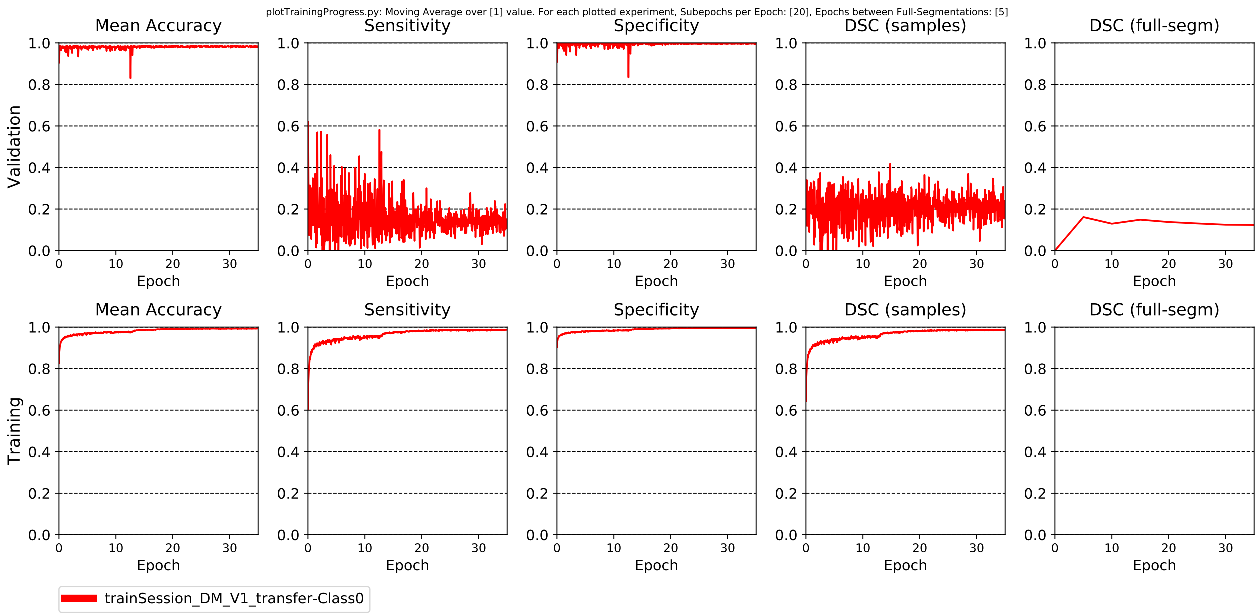

Figure 3. Plotted results of running through transfer learning DeepMedic with the original data pre-processed.

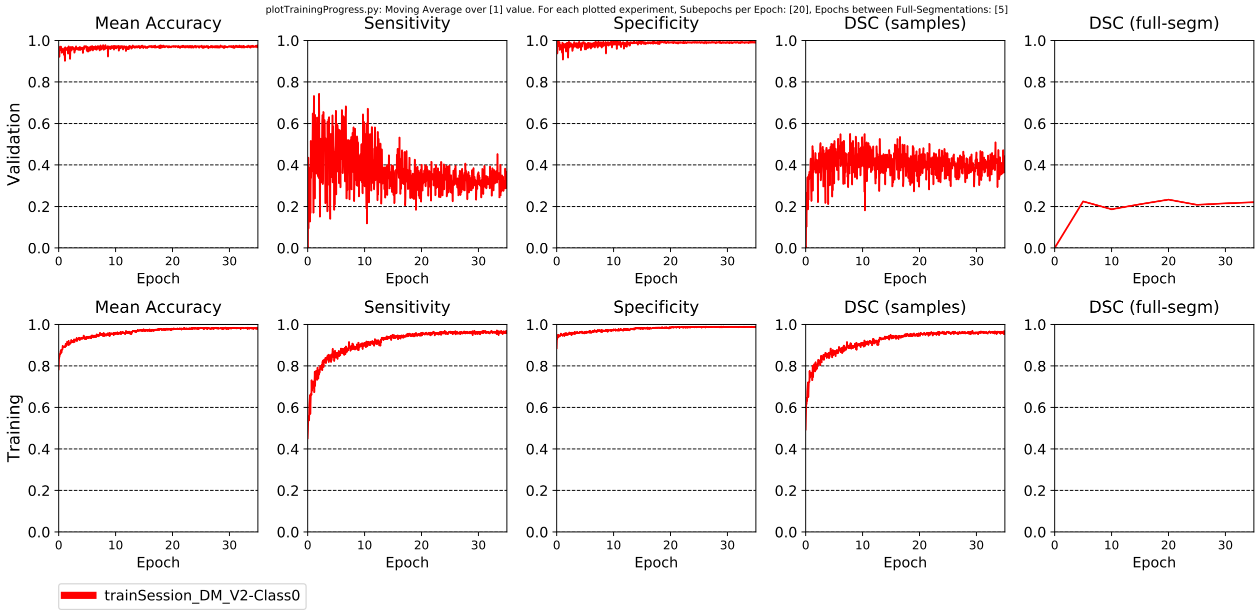

Figure 4. Plotted results of running DeepMedic with the augmented pre-processed data.

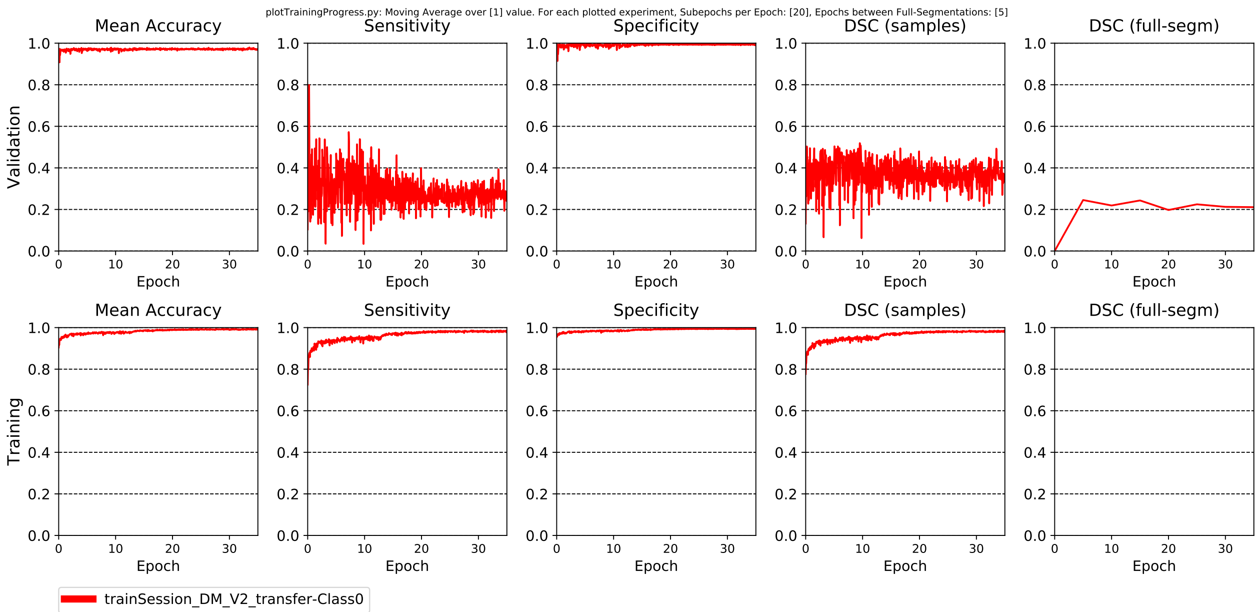

Figure 5. Plotted results of running through transfer learning DeepMedic with the augmented pre-processed data.

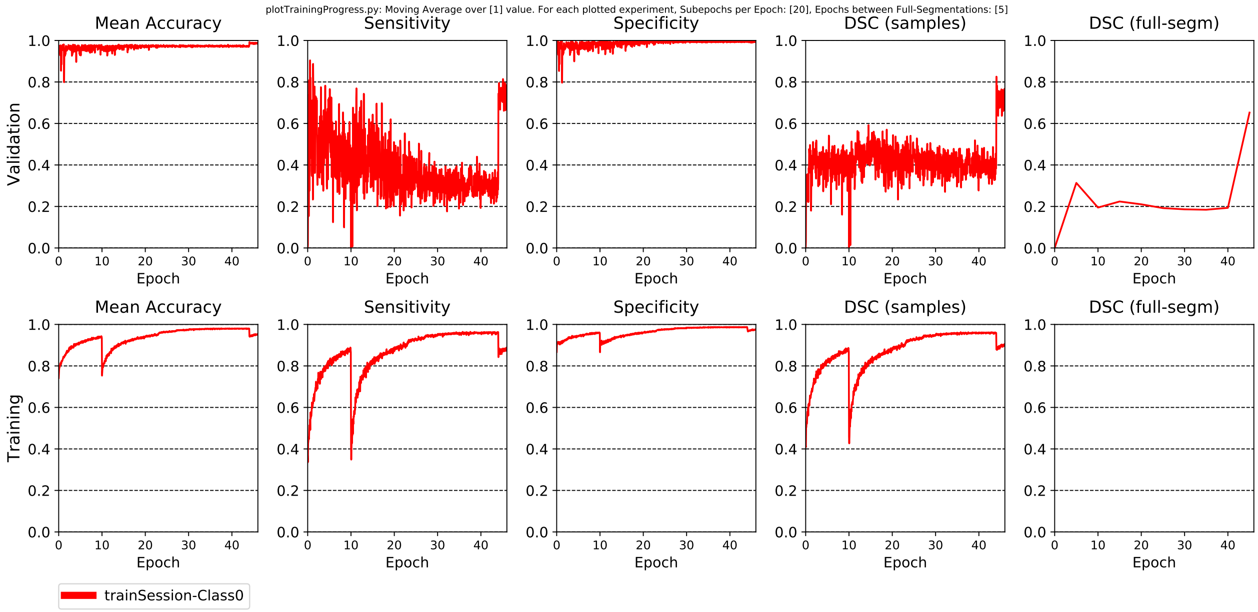

Figure 6. Plotted results of running DeepMedic with the original data pre-processed after fixing pre-processing.

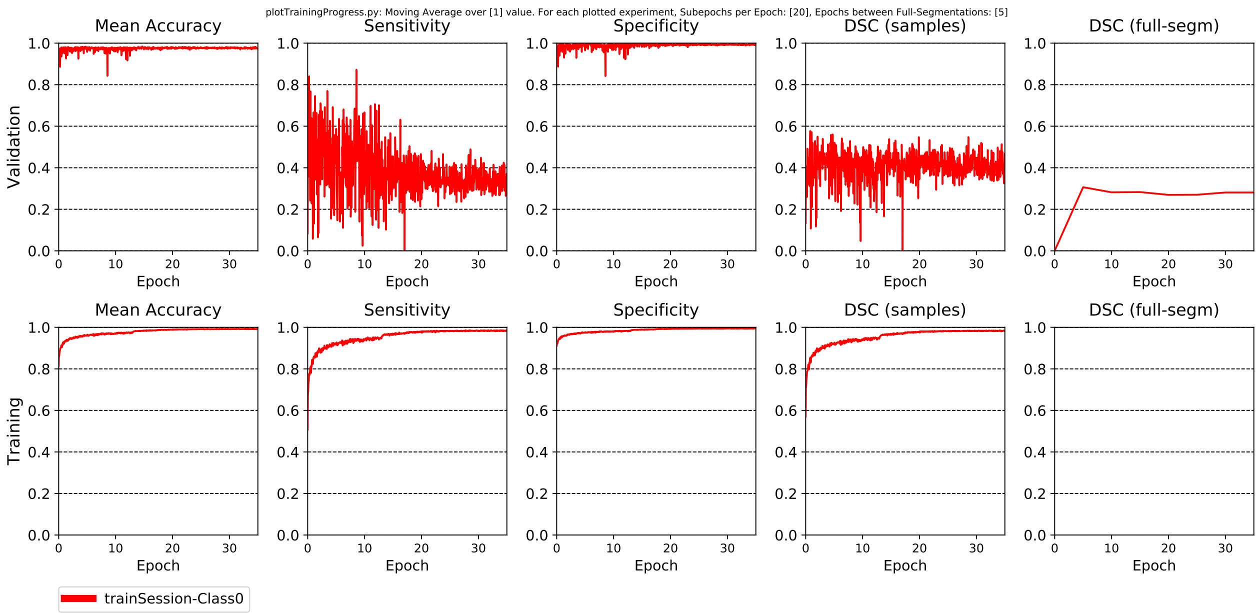

Figure 7. Plotted results of running through transfer learning DeepMedic with the original data pre-processed after fixing pre-processing.

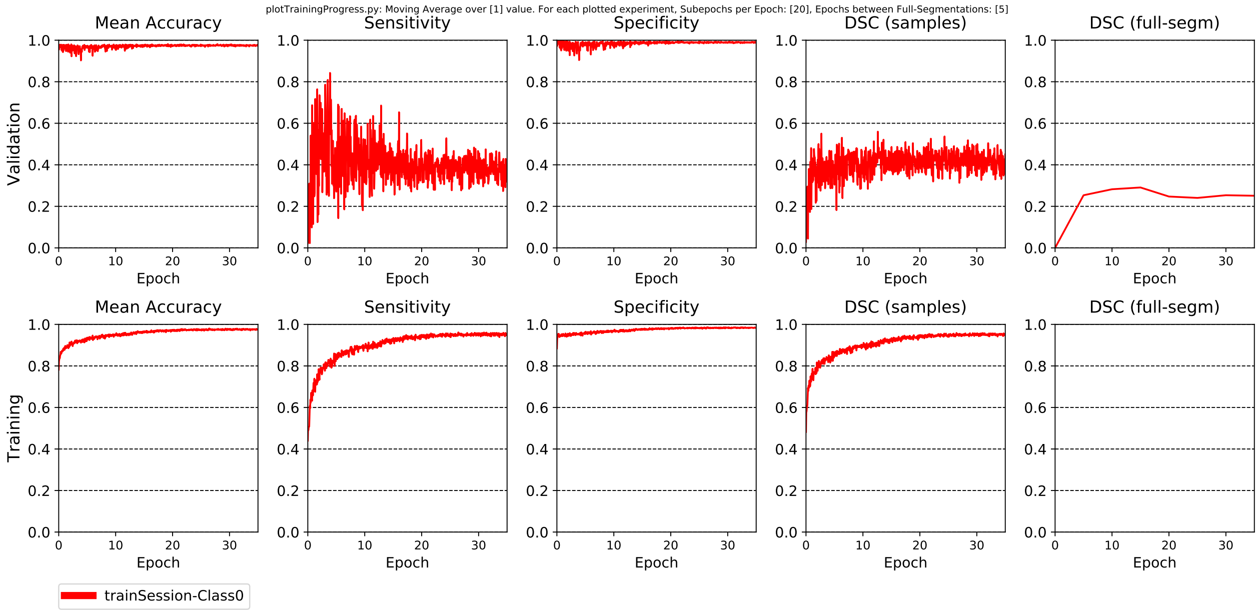

Figure 8. Plotted results of running DeepMedic with the augmented pre-processed data after fixing pre-processing.

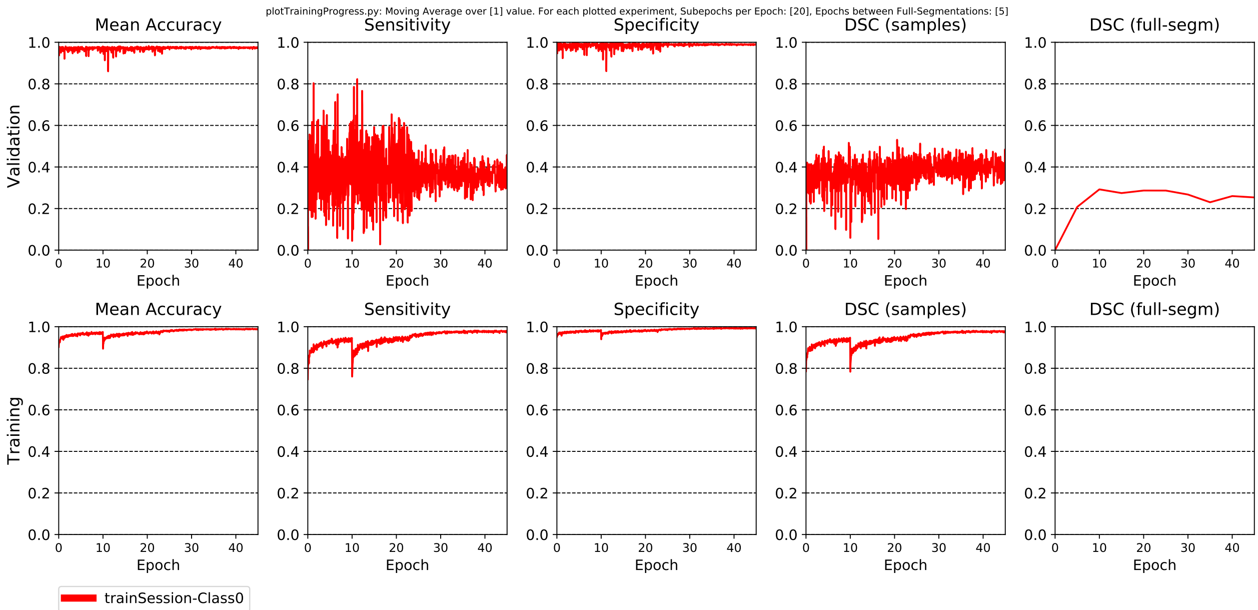

Figure 9. Plotted results of running through transfer learning DeepMedic with the augmented pre-processed data after fixing pre-processing.