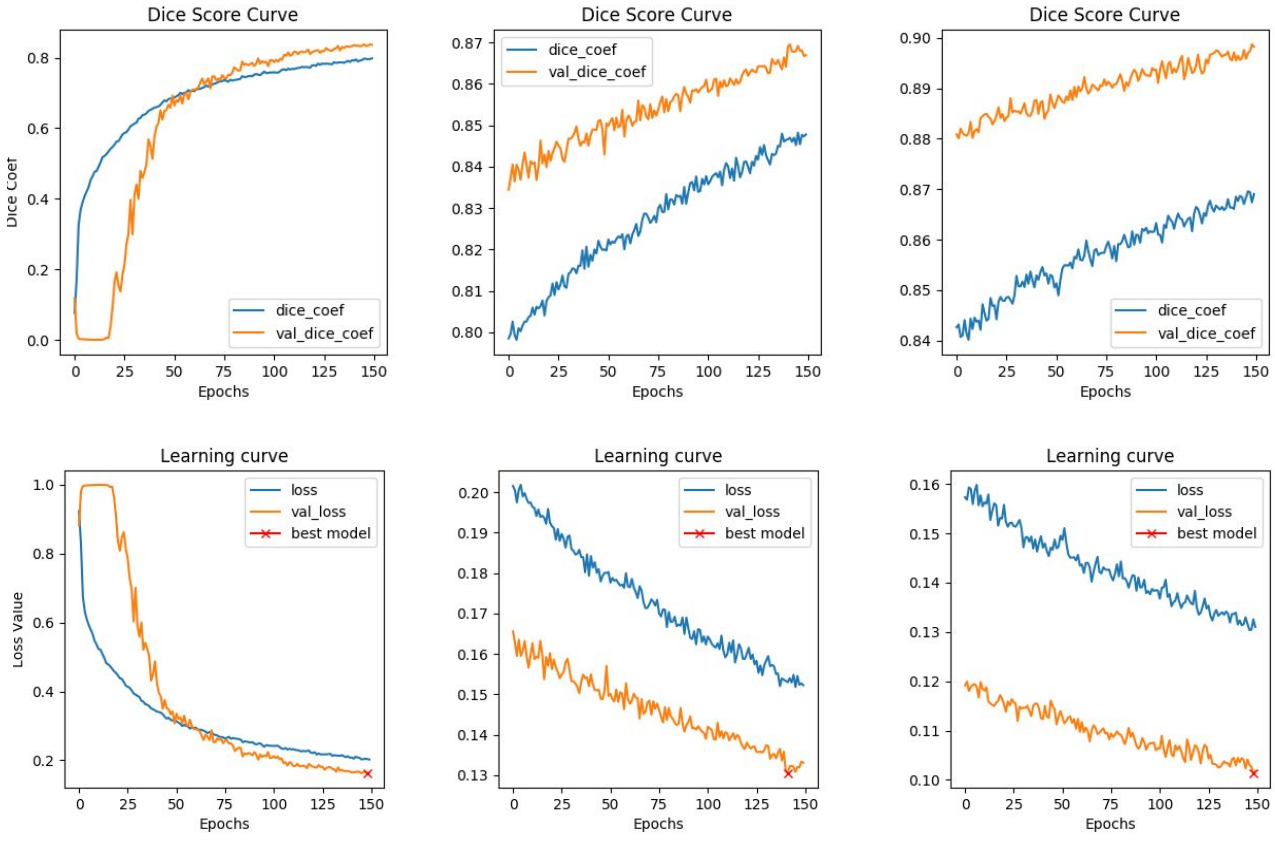
Students: Blanca Cabrera Gil & Laila Niazy
Course: CM2003 Deep Learning Methods for Medical Image Analysis

LAB 4

Improve the performance of the U-Net

Task 1: K-fold cross validation

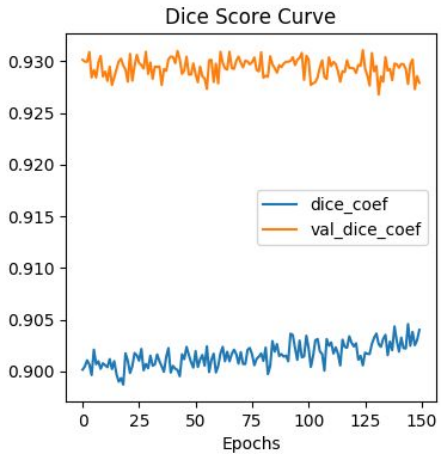
Train your network in this manner on the brain tumor MRI dataset that you have already worked on in the previous lab (path: *./MRI/Image/ for the MRI slices, /MRI/Mask/* for the tumor binary masks). Is the performance consistent across all folds? How would you deal with datasets for which some folds have a different performance than others?

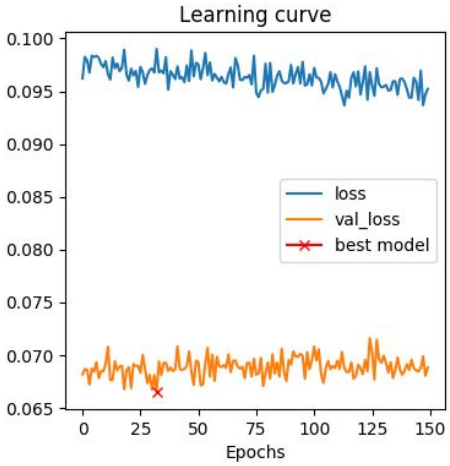


During our first run with 3- fold cross-validation, we forgot to delete the model after each run. In the end the model retrained on new data each run, so the model improved its

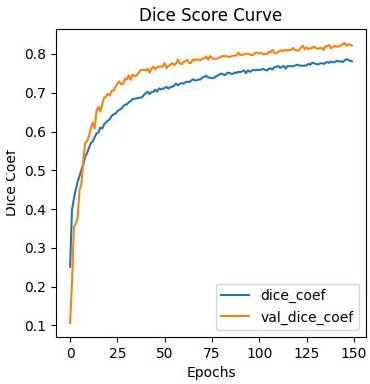
performance after each iteration as can be seen in the plots above the loss function decrease in each plot more and the dice coefficient increases. The plots can be read from left to right, with left being the first fold and the right being the third fold.

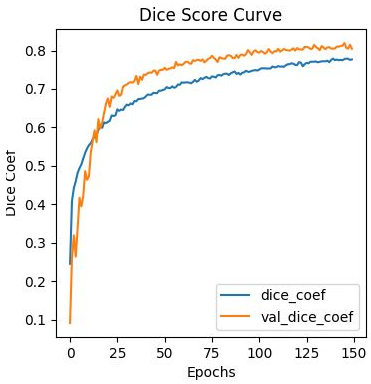
As a bonus task, you can then run more tests by increasing the number of folds (up to a maximum of 10 folds). Do you see any change in the performance?

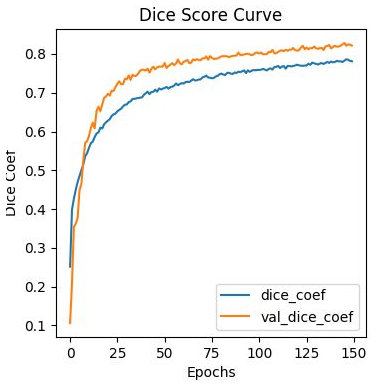




Since we made the same mistake in the bonus task while running 10-fold, we can observe the same phenomenon. The models performance increases with each fold. The plots above are of the final fold and we can see that the dice coefficient achieved a higher value in the final fold here than in the 3-fold cv above. The same can be said about the loss value. It is lower after 10-fold than after 3-fold.





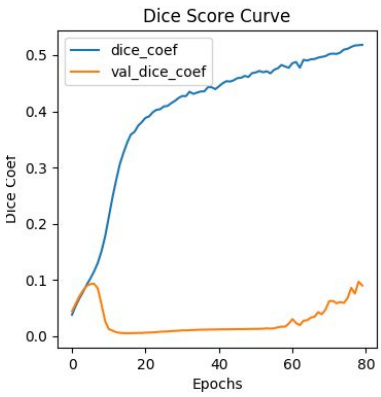


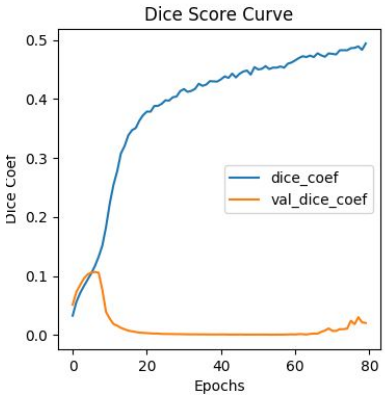
Moreover, for task 1 we had time to repeat the 3-fold cross validation. This time by reinitializing the model after each fold and thus correcting our previous mistake of retraining the model. The plot above of the dice coefficient show the final results and we

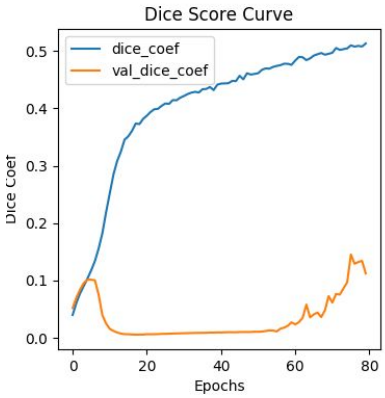
can see that the best achieved dice coefficient is pretty similar across all folds, which is what is expected.

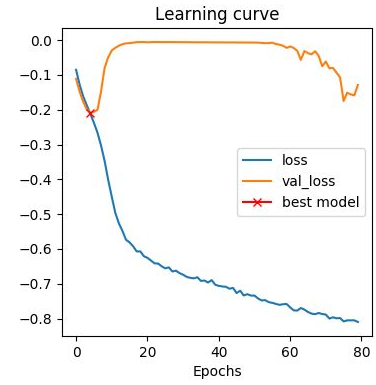
Task 2: Introducing weight maps

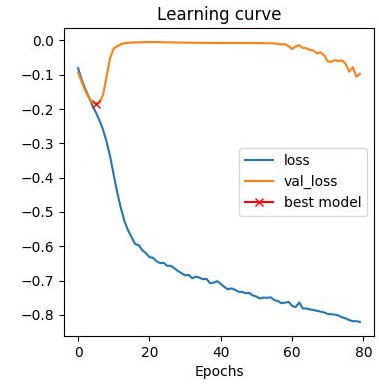
You can then finally start training your model. Can you observe a discrepancy between loss function and the traditional (un-weighted) dice coefficient evaluation metrics? Is it how you expected it to be? Which accuracy can you achieve?

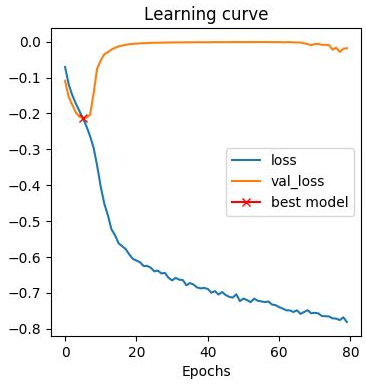










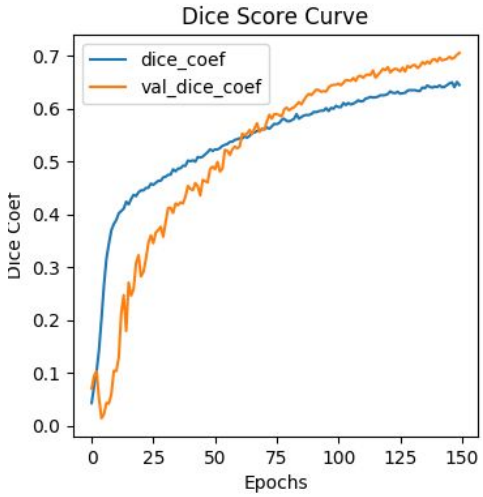


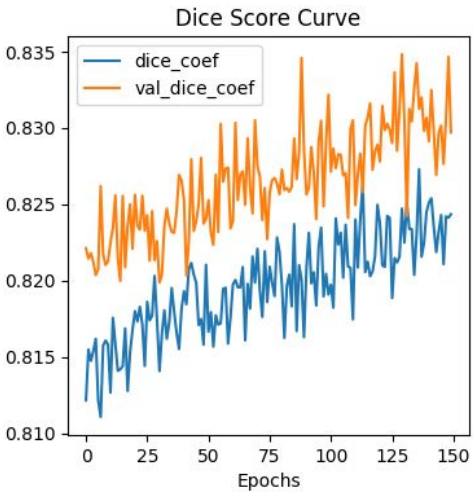
For task 2, we received surprising results. As we can see in the plots above the validation dice coeffiecient is not increased meaning its not learning and loss value for the validation is also not decreasing. We could not get to the bottom of this issue due to the server being down and we could not repeat the task. However, we think its overfitting the data due to our choose parameters because the training set is learning really well as can be seen in the plots. It only reaches about 0.5 dice coefficient because we used a fewer number of epochs to train due to time constraint and the

slowness of the server. We are sure if we trained the model for longer epochs the value of the dice coefficient would have been higher as in the other tasks.

Task 3: Adding autocontext

Please train the network with at least three cycles of training (T=2). Did the autocontext layer help for this segmentation task?





In the third task, we made the same mistake as in task 3 and retrain the model in each fold, so the performance after each fold increased. The two above plot of the dice score are of the first and last fold in the cycle with s = 1. We train the model for T=3 instead of two. The two plots below show the model performance in the first and final fold after 4 cycles. We can see the model performance definitely increased after each cycle with the dice coefficient increasing from 0.835 to 0.8525. Moreover, we assume that had we reinitialized the model after each fold, then the difference in performance between the first and the last cycle and between this approach and the cross validation would have been major with a higher dice score for using autocontext. Unfortunately since the training took along time, we were unable to repeat task 3 as we did task 1 to correct our mistake.