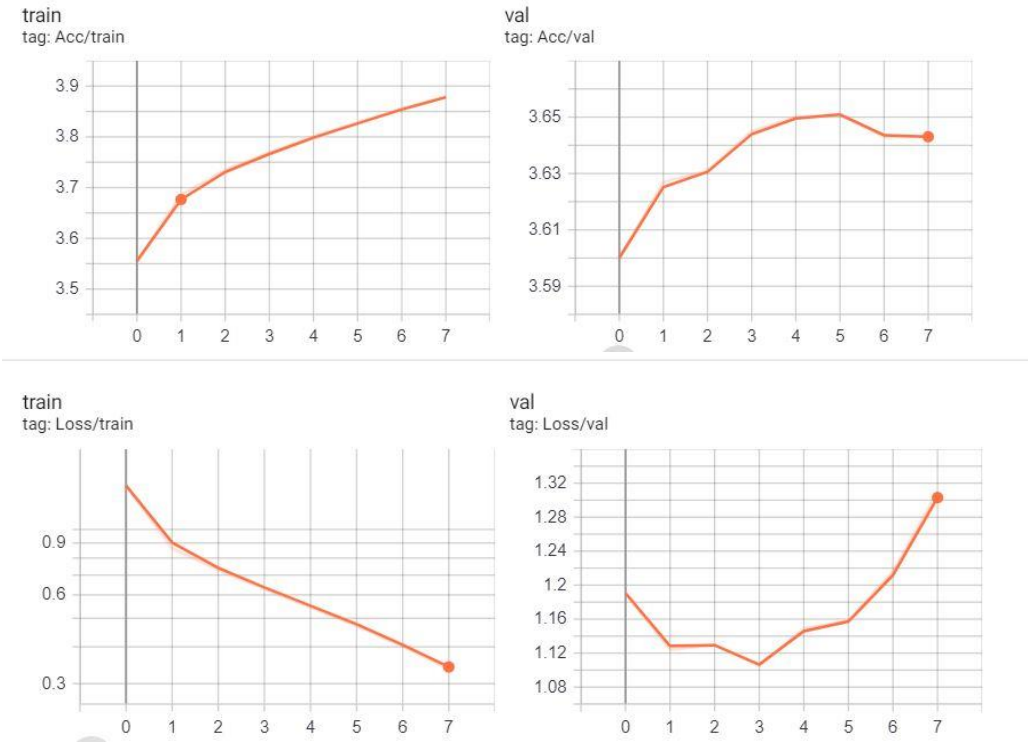
# Computer Vision Exercise

## Task 1: Semantic Segmentation

### Pretraining task:

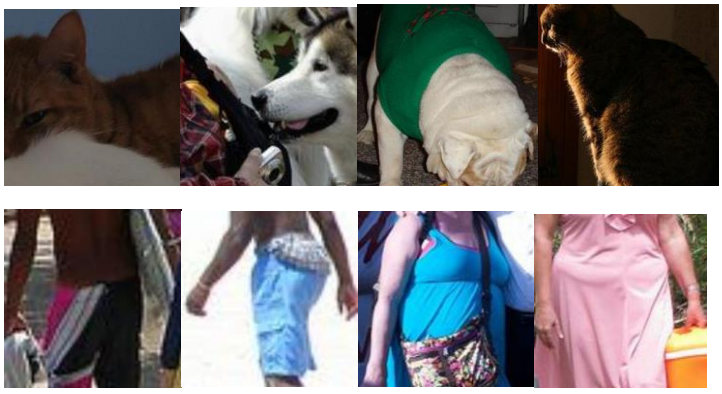
The pretrained Resnet helped to reduce training time. The initial training using 256 x 256 size images was taking too long for a single epoch (~ 3 hours) so I decided to reduce the size of the images to 12x x 128. This speeded up the training by a significant amount. After 3 epochs the validation loss started increasing and the validation accuracy began to flatten, so after waiting for 5 more epochs I stopped the training. The accuracy and loss curves are as follows:




### Nearest Neighbour Task

Despite the reduction in size the network seems to have learned some meaningful features as seen from the images. Using a size of 256 and 128, I obtained 2 sets of results.

*For 256 x 256*

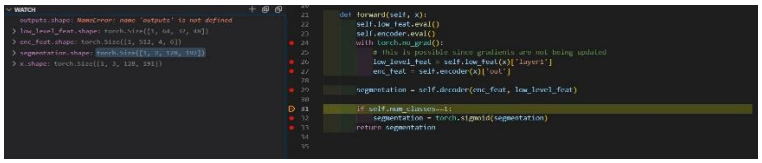


*For 128 x 128*

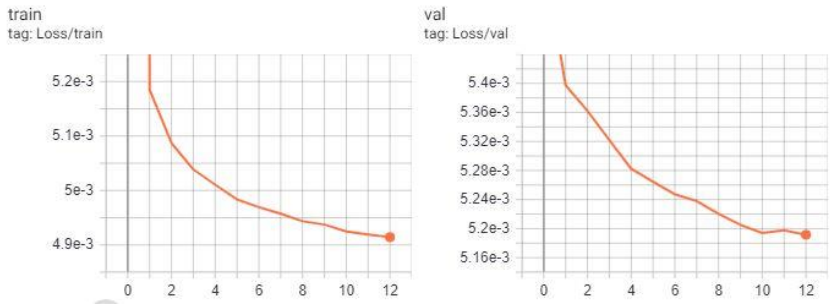
Overall observation is that the network gives more consistent results when the input size for the nearest neighbour task matches the input size for which the network was trained.

## Binary Segmentation task

For the Binary segmentation task, I used BCELogits loss and SGD optimizer without any scheduler. During the 1st epoch of validation, I realized that my input was being resized by one pixel after the decoder stage in the deeplabV. This was because in the deeplab decoder the forward method input variable took the low-level feature dimension and multiplied it by 4. Since my image had a width of 191 and the output of the decoder was 48*4 = 192, it caused a discrepancy. I rectified this by force resizing all images to 128 x 128 in the transforms.py file.



*Train and Validation loss curves are as follows:*




Segmentation masks obtained for the images after 12 epochs are already very similar to the input masks. However due to time constraints I was not able to complete the training and test my network out on fresh images. I have included the latest saved model from the binary segmentation task.