TeamTea Task2

Method:

Preprocess:

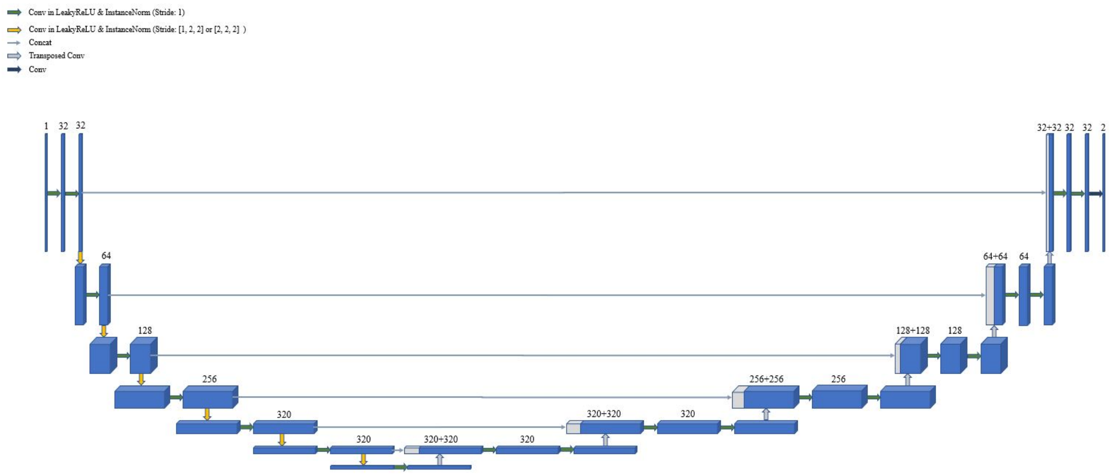
Convert the data of each case into a four-dimensional numpy array and the corresponding pickle file. In the four-dimensional array(CXYZ), the last array[-1,:,:,:] of the first dimension stores the result of segmentation and annotation. The front of the first dimension stores data of different modalities, such as FLAIR, T1, T2, etc. in MRI data. Array[0,:,:,:] represents the intensity data of FLAIR sequence imaging, array[1,:,:,:] represents T1-weighted intensity data, and so on. If it is only monomodal, the length of the first dimension of the four-dimensional array is only 2, which respectively represent image data and label data. The last three dimensions of the four-dimensional array represent the three-dimensional data represented by the three coordinates of x, y, and z. For the original image data, the value represents the intensity, and for the annotation result, the three-dimensional data values of the last three dimensions are 0, 1 respectively. , 2... indicates different label categories.

Image cropping is to crop a three-dimensional medical image to its non-zero area. The specific method is to find a smallest three-dimensional bounding box in the image, the value outside the bounding box area is 0, and the image is cropped using this bounding box. Compared with before cropping, the cropped image has no effect on the segmentation result, but the image size can be reduced to avoid useless calculations and improve calculation efficiency. Cropping is divided into three main steps. The first step is to generate a three-dimensional non-zero template nonzero_mask based on the four-dimensional image data (C, X, Y, Z) to indicate which areas in the image are non-zero. Different modalities have corresponding three-dimensional data, resulting in different three-dimensional nonzero_masks, and the non-zero template of the entire four-dimensional image is the union of the non-zero templates of each modal. Finally, the binary_fill_holes function of the scipy library is called to fill the generated nonzero_mask. The second step is to determine the size and position of the bounding_box for clipping according to the generated non-zero template. In the code, it is necessary to find the minimum and maximum coordinates of the nonzero_mask value of 1 on the x, y, and z three coordinate axes. The third step is to crop each modality of the image in turn according to the bounding_box, and then recombine them together.

The purpose of resampling is to solve the problem of inconsistency in the actual space size represented by a single voxel voxel in different images in some three-dimensional medical image data sets. Because the convolutional neural network only operates in the voxel space, it ignores the size information in the actual physical space. In order to avoid this difference, it is necessary to resize different image data in the voxel space to ensure that the actual physical space represented by each voxel in different image data is consistent. First, determine the size of the target space for resampling. Put the spacing of each data in a list, and count the median value of each dimension spacing. Next, determine the target size of each image according to target_spacing. The product of spacing and shape of each image is a fixed value, which represents the size of the entire image in actual space. Finally, call the resize function in the skimage library to resize each image.

The purpose of normalization is to make the gray value of each image in the training set have the same distribution. For non-CT image normalization, only the gray information of a single image is used to calculate the mean and variance.

Network architecture:

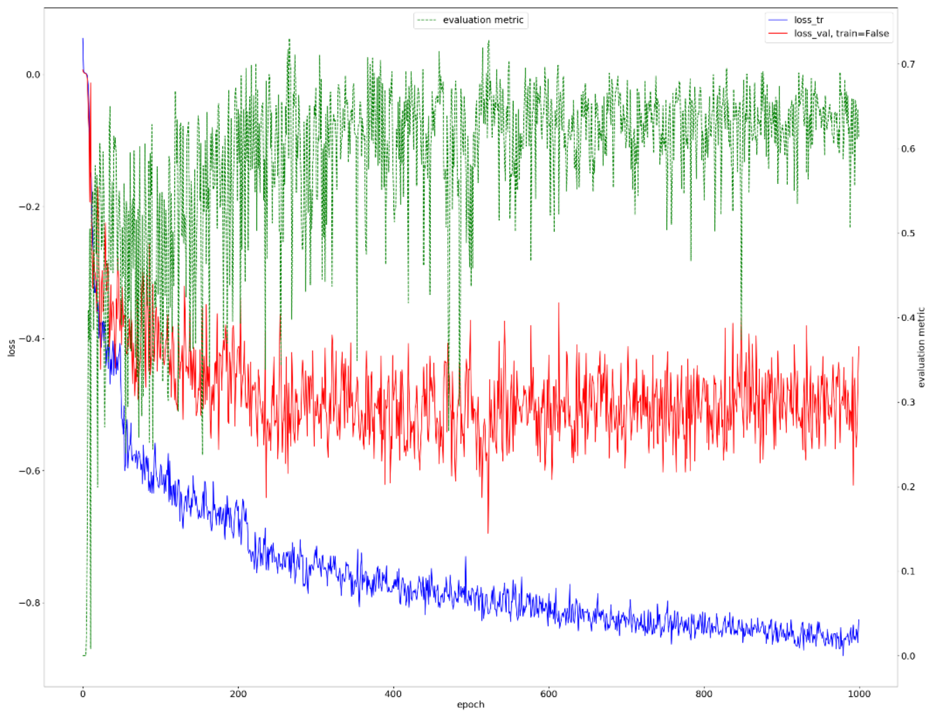


Train:

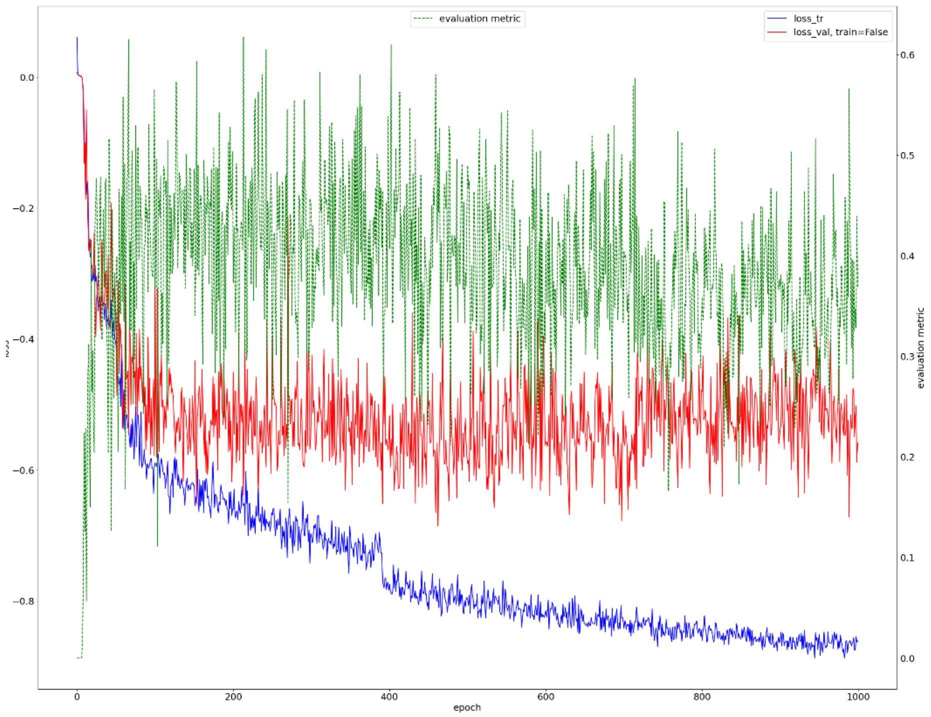
We use 3D nnU-Net in Task2 and two folds(0, 1) of the 5-fold-cross-validaton is trained by nnUNetTrainerV2. We set the initial learning rate as 0.01, set the maximum epoch as 1000, set the momentum as 0.90 and use SGD optimizer.

Result:

Train loss and validation loss:



fold 0


evaluation metric
loss_tr
loss_val, train=False
epoch
evaluation metric

fold 1