Deep Learning with Medical Image

Task description

Our goal is to determine if a tumor's invasiveness $\geq 75\%$ using deep learning model. In this task, "EIC negative" and "EIC positive" stands for invasiveness $\geq 75\%$ and invasiveness < 75 respectively.

Dataset

The dataset is from a doctor in Kaohsiung Medical University Chung-Ho Memorial Hospital who is now doing some researchs about medical image and computer science.
It contains:

1. Dicom files: For each patient, left and right breast have their own labels. Which means that left and right breast chould have different invasiveness. There are two image for each side of breast. During training, our goal is to use these two images as the input of model and predict if the breast is EIC positive.
2. A csv file which records the label of each breast.
3. A csv file which records the clinical data of each breast.

Data preprocessing

First, drop out all the data with NA.
The remaining number of data points is 1001.
Then, I seperate them into 10% test data and 90% training data. Further, 10% of training data is used for validation.
Because of the imbalance of EIC positive and EIC negative, I use some image augmentation skills such as copy, rotate, scale and center crop to balance the training dataset.
So the final number of dataset is:

dataset	EIC positive number	EIC negative number
train	213	597
valid	27	63
test	22	79

The ratio of EIC positive and EIC nagative are silimar in three dataset.

- note that image augmentation methods are only applied on the training part of the dataset and the table only shows the original data number.

Model design

For each breast, we have two images and a vector of clinical data.
To combine all this together, I use two pretrained pytorch.resnet models (resnet18) for images and combine the output with clinical data. Later, the combined output would pass through a fully connected layer and become the output of the entire model.
The input size of image is 200x200, all the image would be center cropped before getting into the model

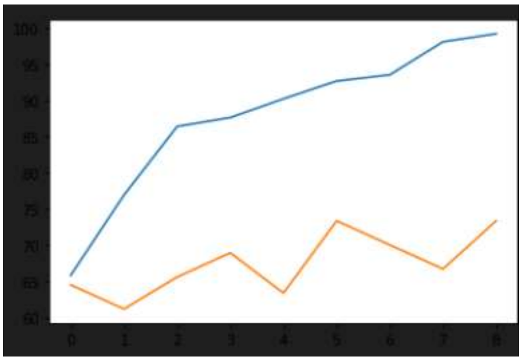
Optimizer and loss function

I use CrossEntropyLoss as my loss function and SGD as optimizer.

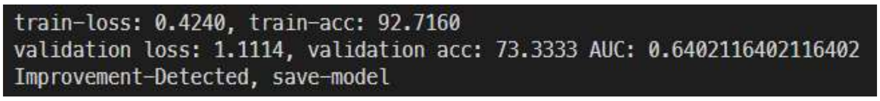
result

For the model selection, I choose the model with highest AUC score during training.

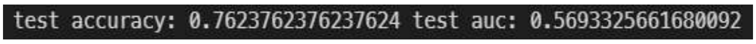
- Without balanced
  learning rate = 0.001, momentum = 0.9
  The training accuracy can reach 100% however, the vaidation accuracy is just around 60 ~ 70%.



The validation accuracy and AUC score

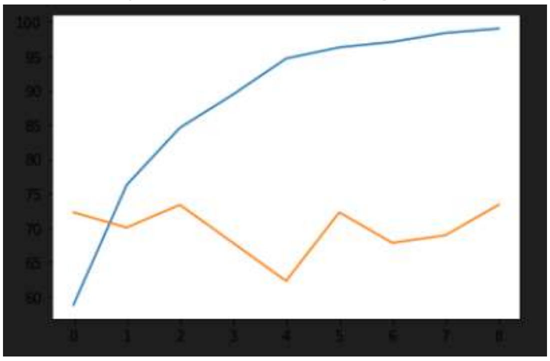


```
train-loss: 0.4240, train-acc: 92.7160
validation loss: 1.1114, validation acc: 73.3333 AUC: 0.6402116402116402
Improvement-Detected, save-model
```

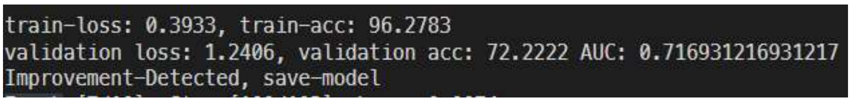


```
test accuracy: 0.7623762376237624 test auc: 0.5693325661680092
```

- Image augmented through copy the training as 3 times larger:
  learning rate = 0.001, momentum = 0.9
  By the image below, we can see that the training accuracy are close to 100%, however the validation accuracy is still not so good.

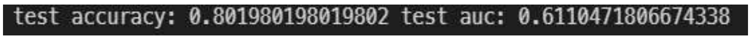


The validation accuracy and AUC score



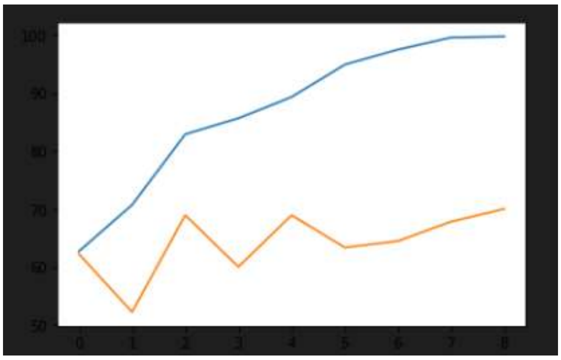
```
train-loss: 0.3933, train-acc: 96.2783
validation loss: 1.2406, validation acc: 72.2222 AUC: 0.716931216931217
Improvement-Detected, save-model
```

The test accuracy and AUC score

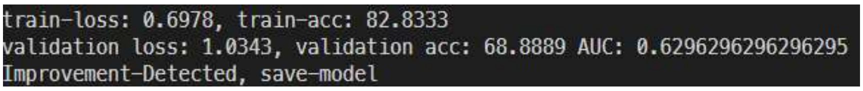


```
test accuracy: 0.801980198019802 test auc: 0.6110471806674338
```

- Image augmented by randomly pick images and randomly rotate and scale them.
  learning rate = 0.001, momentum = 0.9
  The validation accuracy is a little bit worse than before.

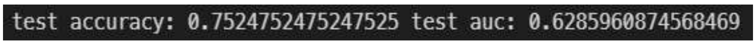


The validation accuracy and AUC score



```
train-loss: 0.6978, train-acc: 82.8333
validation loss: 1.0343, validation acc: 68.8889 AUC: 0.6296296296296295
Improvement-Detected, save-model
```

The test accuracy and AUC score



```
test accuracy: 0.7524752475247525 test auc: 0.6285960874568469
```

conclusion

For what I have tried right now, we can learne d that resnet do detect something while training. However overfitting is a huge problem. I have tried to do some image augmentation to the dataset, but the model still not work so well.
As for the imbalance data, we can see that the accuracy for imbalance dataset is silimir to balanced dataset. But test dataset is really imbalance, we should consider the AUC score. We can see that a balanced dataset could make a better model with higher AUC score.

What to do next

1. Solve the problem of overfitting.
2. Try other pretrain model in the merged model.
3. Model visualization