Transforming How We Diagnose Heart Disease

Hengcheng Zhu, Yuhui Zhang

1.Introduction

In cardiovascular physiology, the volume of the left ventricle (LV) is important in heart disease diagnosis. The difference of end diastole volume (EDV) and end systole volume (ESV), that is EDV-ESV, reflects the amount of blood pumped in one heart beat cycle (Stroke Volume, SV), which is often used in the diagnosis of heart diseases. Another frequently used indicator is the ratio (EDV-ESV)/EDV (Ejection Fraction, EF).

In this project, we are given 500 hundred peole's MRI scans with EDV and ESV labels.The task is to predict EDV and ESV in testing set.Since we do not have access to the test set, we just split the training set into two parts with ratio of 2.After preprocessing the dataset, we finally build two Neural Networks to predict the EDV and ESV on the 163 testing sample based on 330 training set.

2.Data Overview and Preprocessing

Now, we have a dataset of 500 subjects. For each subject, we have several observation for different trials, every observation consists 30 cardiac MRI images in DICOM format. These 30 2D cine images make up the cardiac cycle. To be noticed, for different subject, we have different number of trials.So it is important to firstly scale them into one unifed scheme.

The general preprocessing is first extracting all .dcm files from zip file, then construct a function to import all .dcm files into array in the python environment. We use Object Oriented Programming to process the folders in batch. As expected, there are 7 subjects with missing values so we just omit them. After the processing on all the train folders done, we can access many attributes of the Dataset objects which is stored as elements of 'dset' list. We are able to load data of 493 subjects with 7 subjects dropped.

To simplify this problem, we just take 'sax' images and ignore '2ch' and '4ch' images.We firstly tried the GSI's procedure to make question easier. Roughly, for each subject, we just resize and take average of all these images to make this problem easier. The first purpose of these operations is scaling the number of pixels in output for each subject . And the second purpose is extracting more representative feature from multiple images. Through reshaping, we will get a $493 \times 64 \times 64$ matrix. We can interpret it as every subject has 64×64 features. We call this as **Data1**.

However, the above method to deal with the X matrix will always decline the precision of our model for it ignores the properties of periodicity. Another way to deal with the data is that we take average of images in the same stage for every subject instead of taking average of all the images. Through this method, we will get a $493 \times 30 \times 64 \times 64$ matrix, which still reserve the property of periodicity. We can interpret it as every subject has $30 \times 64 \times 64$ features. We call this as **Data2**.

Then we split the two datasets into two parts, with 330 subjects as new training data and 163 subjects as testing data.

Also, since the outcome of our neural network is 600 values representing differernt probabilities(In sum to be 1), we use one hot encoding on target value to help us better evulate our model. In terms of the evaluation method, we set the result to be related to cumulative probabilities. To be specific, P0 represents the probability the volume is less than or equal to 0 mL, P1 represents the probability the volume is less than or equal to 1 mL and is larger than 0 ml, etc. So we round the target value to int and extend the target variable to one-hot vector of length 600.The specific evaluation criterion named **CRPS**

$$C = \frac{1}{600N} \sum_{m=1}^{N} \sum_{n=0}^{599} (P(y \leq n) - H(n - V_m))^2$$

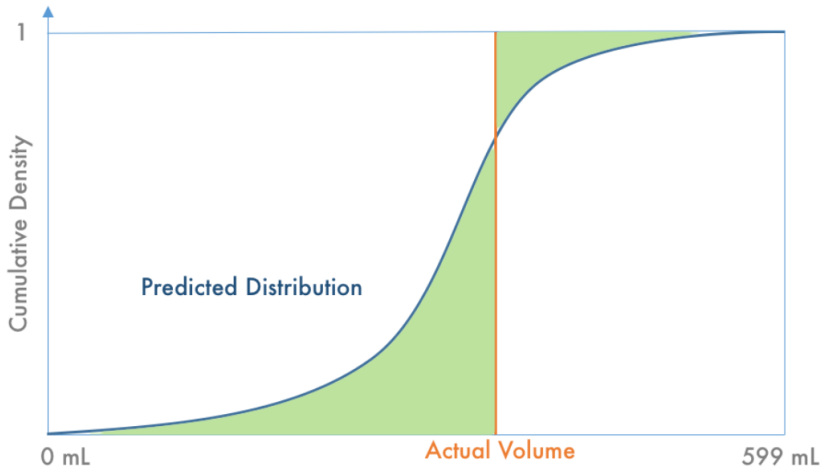


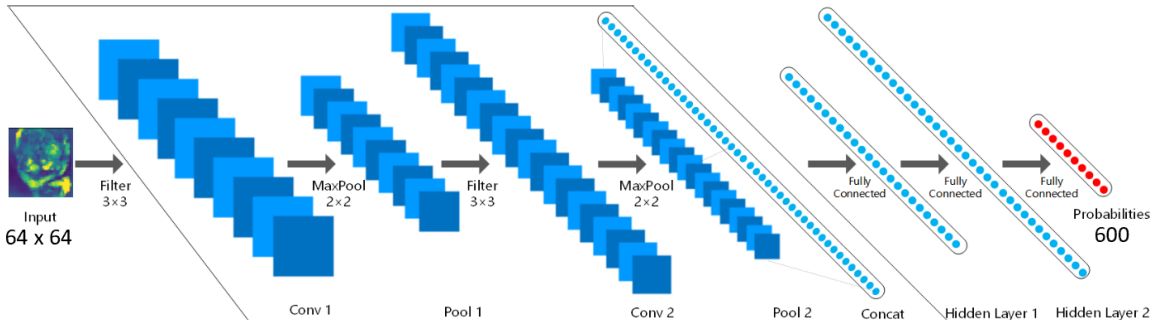
Figure 1: CRPS

where P is the predicted distribution, N is the number of rows in the test set(equal to twice the number of cases),V is the actual volume and $H(x)$ is the Heaviside step function($H(x) = 1$ for $x \geq 0$ and zero otherwise).

3 Neural Network

We bulit two neural networks, one is very simple Convolutional that is nearly the same as discussed in class and another is realtively complex residual neural network for **Data1** and **Data2** respectively.

For the first CNN bulit on **Data1**, the structure is as follows:



The final performance of our model is $CRPS = 0.0359$.Compared to the leaderboard, we found we rank 118/192.(just for fun).

We also tried another neural networks, residual network.Instead of training on the target value, we focus on the residual value.In order to overcome various problems when the traditional network has too many layers, including gradient dispersion or gradient disappearance, we have to change the mindset of traditional CNN. Similar to the boosting algorithm, the loss function of each tree is the negative gradient of the previous tree (that is, the residual) as the gradient boosting tree.For each layer of the Residual network, we built two ResidualBlock, with structure being convolutional layer $\rightarrow$ Normalization layer $\rightarrow$ relu-convolutional layer $\rightarrow$ Normalization layer.

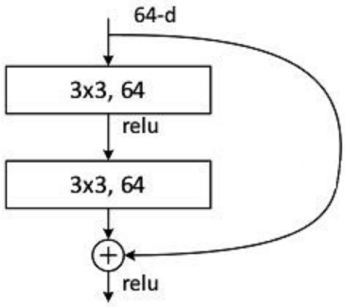
Therefore, the function we study is $y = (F(x) - y) + y$.The goal of training is that we would like to find a H, such that $H(x) = x$. Considering the difficulty, we study on $H(x) = F(x) + x$ instead, where $F(x)$ stands for the residual. The identity of the residual is easier to learn than the original parameters. To be specific, instead of studying $H(5.5) = 5.5$, we turn to study on $H(5) = F(5) + 5$, which equals to $F(5) = 0.5$. The new function is better to train.

After constructing the residual network, we build the entire network.

- 1.Build the ordinary CNN network and input the variable. Through a 3×3 convolution kernel, the normalization layer and the activation layer, the implicit variables are obtained.
- 2.Build layer in order(6 in total). In every layer, the number of blocks is layers[i]. Every layer is built to fit the residuals in the upper layer. Still take $H(x) = F(x) + x$ as an example.

a) block1:$F1(5) = 0.1$, $H1(5) = F1(5) + 5 = 5.1$,the fitted residual equals $5.5 - 5.1 = 0.4$
b) block2:$F2(5) = 0.2$, $H2(5) = F2(5) + H1(5) = 5.3$, the fitted residual equals $5.5 - 5.3 = 0.2$
c) block3:$F3(5) = 0.2$, $H3(5) = F3(5) + H2(5) = 5.5$,which is the target $H(5) = 5.5$

We perform this residual network on **Data2**, the final performance of it is 0.0342, which is a little bit better than CNN.



4 Discussion

Although we tried two neural networks to train the model, we outcome is very poor. We think the main reason for it is there is too many "noise" in every scans and here noise means other parts of the image apart from the ventricle.According some wining solution, they firstly conduct a image segmentation by using deep learning and consider it as the main parts of their pipeline. Due to the limited time, we just pool all the images after simple averaging into the neural network, which as expected would give very poor performance.We also push our code on the github:https://github.com/Jackie-Zhu17/PH244-PJ4.