Machine Learning for Imaging – Coursework Report
Age Regression from Brain MRI
Group: 69

Ziyang Yang, Yiming Zhang, Lichen Xue
{zy820, yz12120, lx320}@ic.ac.uk

1 Introduction

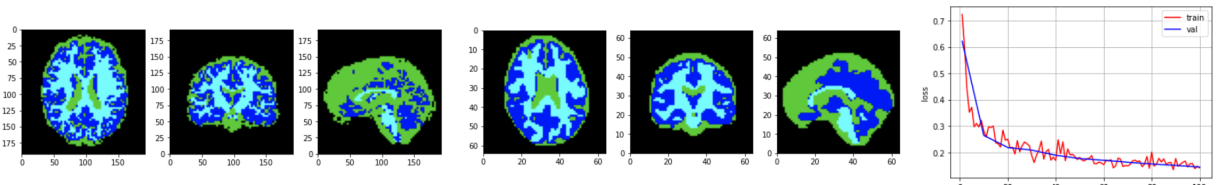
With a great potential of application, medical image processing is a popular area in machine learning. In this study, we aim to predict brain age from brain MRI. Part A and Part B describe two prediction models and their results for cross-validation. All the testing results are included and discussed in section 4.

2 Part A

Firstly, two image segmentation networks are implemented. One is a Le-Net model and another one is a simple network. Le-Net model consists of five convolutional layers for down-sampling, three transposed convolutional layers for up-sampling and one convolution layer for output. Because the Le-Net model has a long training time (about 30 mins), we shifted to a simpler network. The simple network includes 5 convolutional layers and the first 4 layers are followed by batch normalisation. The activation function for hidden layers is ReLU and the loss function is cross entropy.

Figure 1 and 2 show the training result for both models. Compared with original segmentation, it is obvious that main information is captured by Le-Net predicted segmentation while most details are lost. There is punctuation in training/validation curves as training goes on, this indicates the model is not stable. As for the simple network, details are captured in the predicted segmentation, while it contains some wrong information. From the training/validation curves, the loss decreases quickly and becomes steadily, this indicates the model is stable. Dice scores are shown in Table 1. Compared with Le-Net model, the simple network has a slightly higher dice score, which indicates the predicted segmentation is closer to the original segmentation.

Secondly, feature is calculated based on segmentation image. First, the 3D array from segmentation image is calculated. Then a label from background, CSF, GM and WM is assigned to each voxel. In unnormalised calculation, the plot illustrates the absolute number of voxels. In normalised calculation, the plot shows the ratio of each part to the total number of voxels. Figure 3 shows the simple network dice result in box plot. As background dice score box plot has many outliers, background voxels are omitted in ratio calculation. Figure 4 shows the visualisation for tissue volumes vs. age. It is obvious that after normalisation, CSF, GM and WM are closely grouped. Finally, 5 regression methods including Linear Regression, Lasso Lars Regression, K Neighbor Regression, Bayesian Ridge Regression and Multi-layer Perceptron(MLP) are trained based on normalised data. From the cross-validation result in Table 2, MLP is chosen with the best performance for MAE as 7.09 and r2 as 0.77.

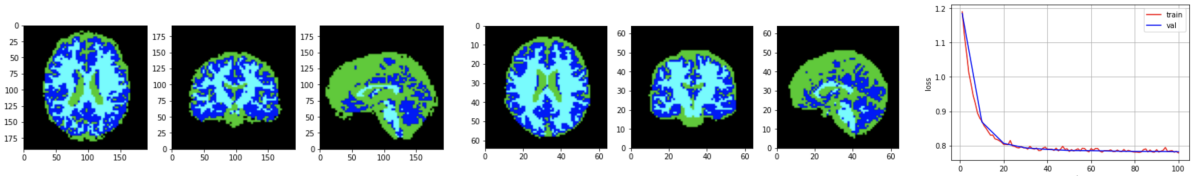


(a) Original Segmentation (b) Predicted Segmentation (c) Loss

Figure 1: Le-Net model running result

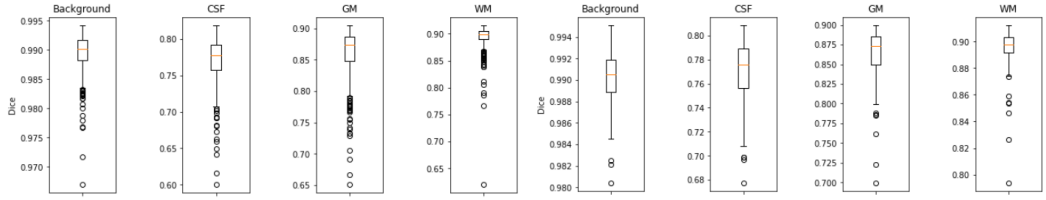
Validation Result	Le-Net Model	Simple Model
Loss	0.144	0.781
Background dice	0.991	0.990
CSF dice	0.702	0.784
GM dice	0.765	0.880
WM dice	0.798	0.902
Average dice score	0.814	0.890

Table 1: Model Validation Results



(a) Original Segmentation (b) Predicted Segmentation (c) Loss

Figure 2: Simple network model running result



(a) result of regression train data (b) result of regression test data

Figure 3: Simple 3d model dice result in box plot

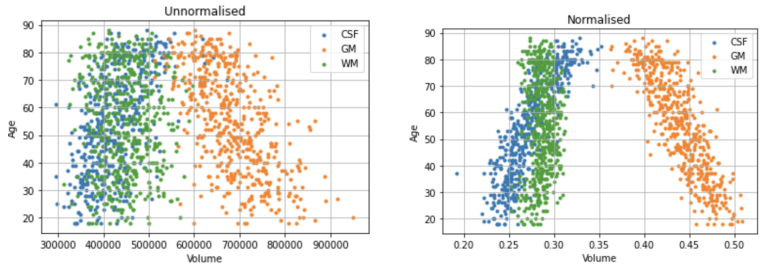


Figure 4: Tissue volumes vs. Age

Model	MAE-Fold 1	r2 - Fold 1	MAE-Fold 2	r2 - Fold 2
MLP	7.16	0.74	7.09	0.77
K Neighbors Regression	7.14	0.73	7.39	0.75
Linear Regression	7.51	0.72	7.71	0.74
Bayesian Ridge Regression	7.52	0.72	7.71	0.74
Lasso Regression	7.65	0.72	7.96	0.73

Table 2: Validation Results of Regression Models

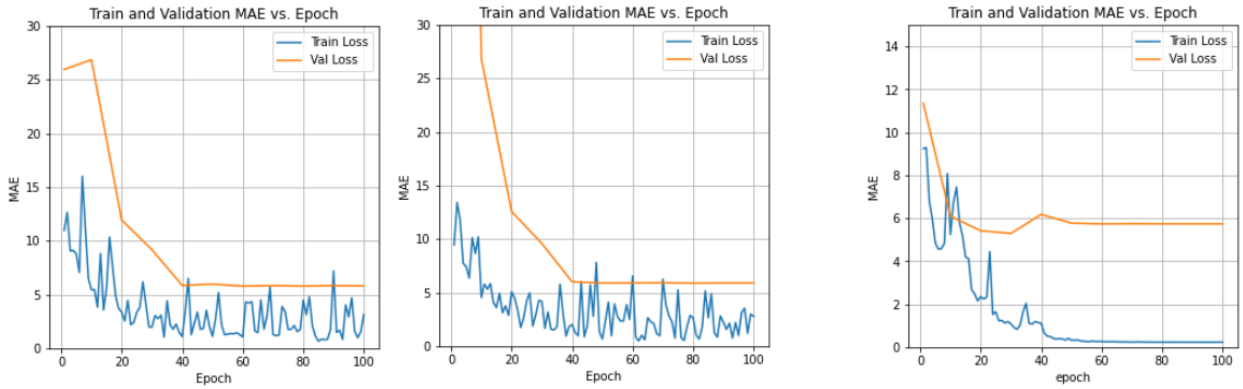
Figure 5: Validation Results of Regression Models

3 Part B

Based on the lesson learned from Part A, we started from a simple network and small parameters this time. The network model includes 6 layers, the first 2 layers are convolutional networks, followed by batch normalisation and max pooling, the next 2 layers are also convolutional networks, followed by batch normalisation, the last two layers are linear layers for regression purpose. The activation function for hidden layers is ReLU. There are many advantages of batch normalisation: the starting hyper parameters don't need to be carefully designed to gain a good result, and L2 or small weight

decay don't need to be used either. During training process, a scheduler is added to change the learning rate dynamically for every eight epochs in order to accelerate the convergence.

During the pre-process, images are normalized to zero mean and unit variance first, then they are re-sampled to (64, 64, 64) with image spacing is (3, 3, 3). These steps can help to transform the range of values for all the pixels in the image to a small range and adjust the element size for later usage. The training/validation curves for the cross-validation are shown in Figure 6. Cross-validation uses 500 dataset, 250 as training data and 250 as validation data. The final training use 500 dataset as training data and extra 47 dataset as validation data. In Figure 6a, training curve drops quickly from a relatively high value to a small value in the first 40 epochs, then it reduces punctually as training goes on. The validation loss curve stays same after 40 epochs and this indicates the model converges at this point. For the final training result (Figure 6b), the training loss become close to zero after 40 epochs, while the validation loss is still around 6. This seems like an overfitting issue. We tried to resolve this issue by adding a scheduler with gamma as 0.4, and the result shows that the scale of overfitting is reduced. Another possible reason is the validation dataset is small and the model is not well validated.



(a) Cross Validation Result(2 folds)

(b) Final Training Result

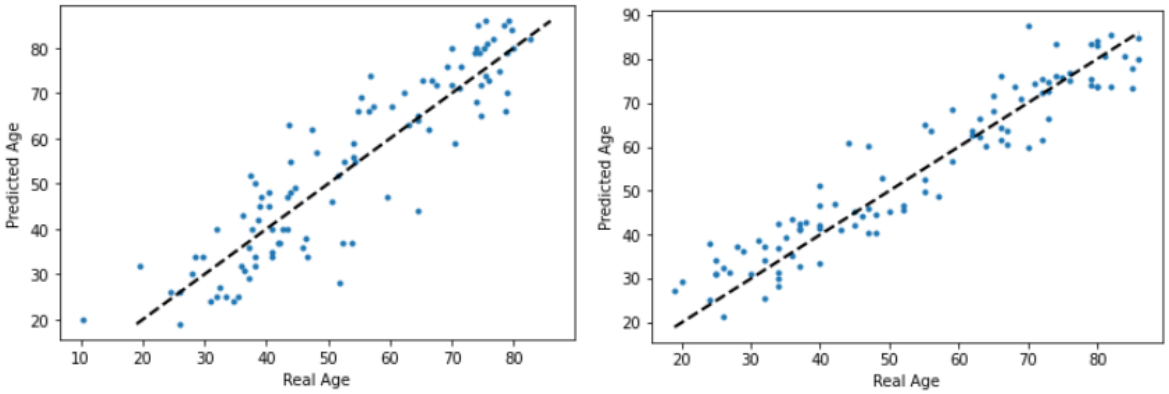
Figure 6: CNN Model Training Result

4 Age Regression Results

This section includes the testing results for MLP model from part A and CNN model from part B. The number of training dataset is 500 and the number of testing dataset is 100. Table 3 shows the result. Since each training and testing trial outputs a different MAE and r2, we use a range over 15 trials. For MLP, MAE is in range [6.7, 6.9], r2 score is in range [0.81, 0.83]. For CNN, MAE is in range [4.9, 5.3], r2 score is in range [0.89, 0.91]. The result indicates the performance of CNN is better than the performance of MLP. This also can be observed in Figure7. Compared with the scatter plot of MLP, the scatter plot of CNN model is closer to the line. This is because the MLP model uses normalised data from the feature calculation step, which is gained from image segmentation. Each of these steps might result in information distortion or information loss, which results in decline of prediction accuracy. While for CNN model, it uses the image as input and builds the network model on it directly, which helps to keep the helpful information.

Model	MAE	r2
MLP	6.8 ± 0.1	0.82 ± 0.01
CNN	5.1 ± 0.2	0.90 ± 0.01

Table 3: Test Results of Models



(a) MLP: MAE = 6.78, r2 = 0.82 (b) CNN: MAE = 5.15, r2 = 0.90

Figure 7: MLP and CNN Model Scatter Plot

During this study, we learned that it is better to start from a simple network when building a neural network model. Instead of implementing networks that generally have outstanding performance directly, networks such as Le-Net and U-Net can be referred as a guidance during model tuning. From our experiment, a relatively good result can be achieved from the simple network in this work.