Age Regression from Brain MRI
Group: 21

Ling Qin, Junchen Zhao, Yurun Song
{lq519, jz2119, ys3919}@imperial.ac.uk

1 Summary

Predicting the age of patient from a brain MRI scan can have diagnostic value for a number of diseases that may cause structural changes and potential damage to the brain. A discrepancy between the predicted age and the real, chronological age of a patient might indicate the presence of disease. This requires an accurate predictor of brain age which may be learned from a set of healthy reference subjects, given their brain MRI data and their actual age.
The objective for the coursework is to implement different supervised learning approaches for age regression from brain MRI. We used the data from a total of 652 healthy subjects, that is split into different development sets and a held-out test set on which we will evaluate our final prediction accuracy.

2 Part A

This part requires an image segmentation and an feature extraction to the brain tissues. Then the age of the patient can be predicted by the linear relationship between the age and the tissue features.

2.1 Image Segmentation

Firstly, the U-Net structure was chosen to be the training model for image segmentation. U-Net allows the training network to spread context information towards higher resolution layers. This was achieved by the upsampling part which increases the number of feature channels. The learning rate was set to be 0.0002 since the divergent behavior appeared in the training step with learning rate 0.001. The final loss after the test step was about 0.13. The dice scores for CSF, GM and WM were calculated by applying LabelOverlapMeasuresImageFilter() method. In this way, the similarities between reference image and the predicted image were detected.

2.2 Feature Extraction and Age Regression

In the feature extraction part, the variable vols saved the absolute volume for each tissue. Then, an original and a normalized vols were plotted out. For the final age regression part, three regression models were applied: the Bayesian regression model, the Ordinary Least Squares model and the Lasso Regression. The mean absolute errors (MAE) and the r2 scores for the cross-validation step are as follows.

Regression Model	**MAE (r2)**
Bayesian regression	7.706 (0.714)
Ordinary Least Squares	7.697 (0.714)
Lasso	9.800 (0.597)

Table 1: Regression Performance

3 Part B

3.1 Part B-1

In this part, our task is to do the pre-processing on the feature space for the grey matter maps before doing the PCA dimension reduction for the feature space of the grey matter maps. In order to achieve this task, we firstly read through the grey matter map file directory and read all of the images using the SimpleITK. Then, after we read through the image file, we wrote the image file which was downsampled by the factor of 2 and transformed to discrete Gaussian distribution using the SimpleITK. After we finished the image writing process, we used the SimpleITK to transform the image file to an array and store the image array as a numpy array which has size of (652, 109350).

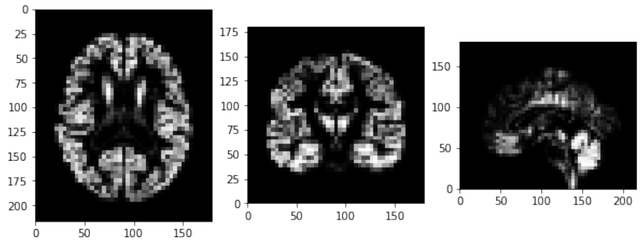

Figure 1: This is our grey image in the grey matter map.

3.2 Part B-2

In this part, our task is to do the PCA dimension reduction for the feature space of the grey matter maps. In order to achieve this, we used the sklearn to split the image numpy array, which was preprocessed in the **part B-1**, and the age array in the meta data to training class and testing class as **Xtrain, Xtest, ytrain, ytest**. Then, we defined the PCA model with **random seed** 42 and 0.95 **number of component**. After we defined the PCA model, we fitted the **Xtrain** to the PCA model and performed the PCA dimensionality reduction on both of the **Xtrain** and **Xtest**.

3.3 Part B-3

In this part, our task is to use different regressors to fit the training data of both of the image data and age data, **Xtrain, ytrain** which are splited into 2 different training folds, after we applied both of them to the PCA model for the dimensionality reduction. In this part, we used 3 different regressors to fit the data: **Linear Regressor, Polynomial Regressor, KNN Regressor**. After applying and fitting the data to the regressor, we could finally use the fitted regressor to get the prediction by fitting test data to the learned regressor model. By having the predictions by applying these 3 different regressors, we could finally calculate the **Mean Absolute Error(mae)**and **data correlation(r2)**. The final result is showing below.

- **In the training fold 1**
- Linear Regression: mae=5.602565 r2=0.861700
- Poly Regression: mae=13.256441 r2=0.313250
- KNNeighbour Regression: mae=7.960890 r2=0.728203
- **In the training fold 2**
- Linear Regression: mae=5.513735 r2=0.858106

- Poly Regression: mae=13.386714 r2=0.309692
- KNNeighbour Regression: mae=7.230828 r2=0.761953

The visualization is showing below:

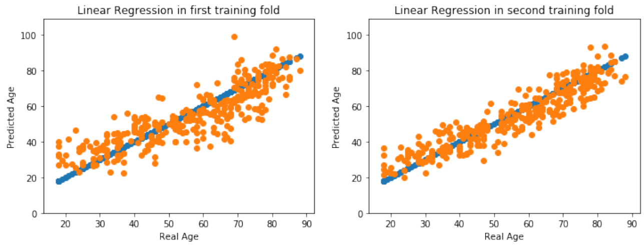

Figure 2: Linear regression on the 2 training folds.

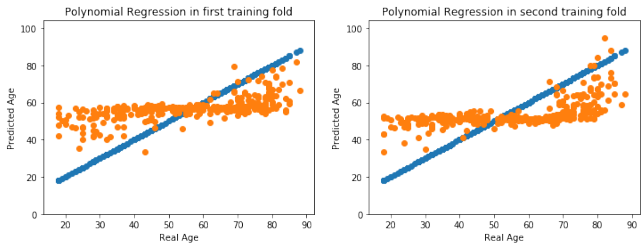

Figure 3: Polynomial regression on the 2 training folds.

Figure 4: KNN regression on the 2 training folds.

4 Part C

4.1 Resampling

The 3D Grey matter maps needs to be resampled first. We divided them into 5 groups of size. All of the tissues were trying to fill the whole 3D Grey matter maps as much as possible. There are 5 kinds of samples:

- size [32, 32, 32] with image spacing [6, 6, 6]
- size [48, 48, 48] with image spacing [4, 4, 4]
- size [64, 64, 64] with image spacing [3, 3, 3]
- size [80, 80, 80] with image spacing [2.5, 2.5, 2.5]
- size [96, 96, 96] with image spacing [2, 2, 2]

Any larger size would lead to run out of the memory in 2 fold cross validation. Therefore, we stopped at size of 96 and the size increased by 16 units every time.
Here are the corresponds graphs we got by using LeNet3D, which simply replaced the fatten layer (FC1 layer) to equal the output from Conv layer. The input of FC1 layer is extremely large when we have size of 96 up to 148176 and the output of FC1 layer is still 120.

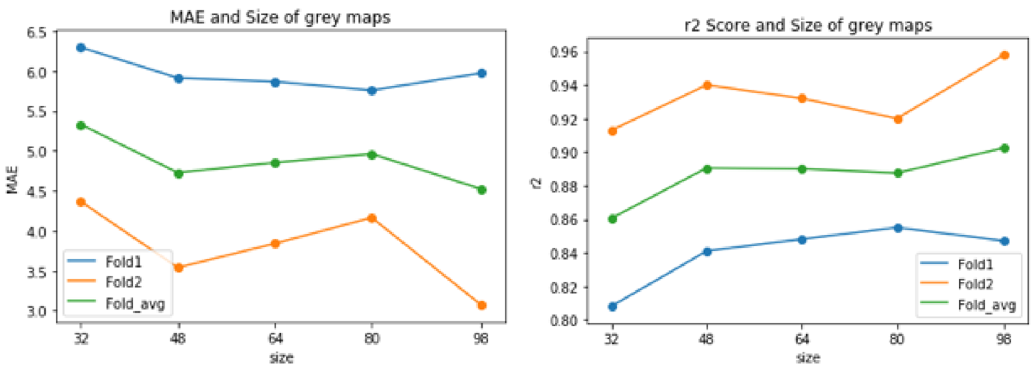

Figure 5: Scatter plots for age regression with resampling size

Here are the two figures5 that MAE and r2 with regard to the size of grey maps. It can be observed from the graph that MAE has a slight tendency to decrease with the larger grey maps while r2 shows the increase. In general, the smaller MAE the better performance the model is. R2 score show the correlation of the prediction and real truth value, which indicates how stable the model is. Therefore, based on the computing memory we have, 96 is the best size of grey matter maps.

4.2 Training

Applying MinMaxScaler for the ages first and then inversing the scale when calculating MAE. The best hyperparameters of training we found: Epochs: 20, Batch size: 15, Adam optimizer with learning rate 0.001 and MSE loss. Besides, the accuracy was measured by r2 score in the testing. For map size like 96, the test loss stopped decreasing at epochs 10 and size 32 merely saturated at epoch 20.

4.3 Performance

With 2 Folds Cross Validation for size 96, using following three Neural Network LeNet 3D, CovLeNet 3D and CustomCNN 3D, which contains 3 Conv3D layer and 3 Linear layer (2 relu non-linear layer).

Neural Network	MAE1 (r2)	MAE2 (r2)	Average
LetNet 3D	5.974 (0.847)	3.075 (0.958)	4.524 (0.903)
FC LeNet 3D	6.244 (0.821)	2.943 (0.961)	4.593 (0.891)
Custom CNN 3D	6.699 (0.799)	5.733 (0.848)	6.216 (0.824)

Table 2: Performance

From the table2, we can see that the performance of CustomCNN is not better than above two. We believe this is mainly because the 3 consecutive pooling layers actually drops too much important information. Therefore, features extracted by the Conv3D layer are no longer accurate any more, which indirectly lead to incorrect prediction in regression. Meanwhile, it proves again the idea that the result of FC LetNet is quite similar to LetNet and fully convolutional neural network indeed does not affect the original accuracy.

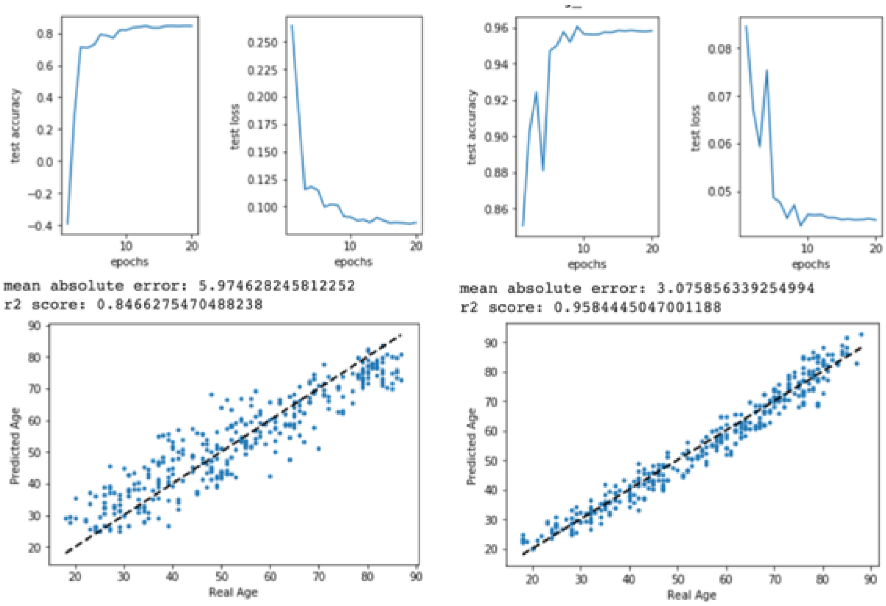

Figure 6: Performance of size 96 LeNet3D

The above figure6 shows the first and second Cross Validation results for size 96 LeNet, which is also the best result we get for the part3. The average ages error is about 4.524 and r2 score is about to 0.903 .

5 Age Regression Results

In this coursework, we implemented different supervised learning approaches for the age regression tasks from brain MRI in part A, B and C.
In part A, it's required to implement image segmentation and an feature extraction on the brain tissues so that the age of the patient can be predicted by the linear relationship between the age and the tissue features. In order to achieve our goals: image segmentation, feature extraction, age regression, we decided to use U-Net structure to train our model for the image segmentation task**(1)**; we decided to save our extracted features, the absolute volume for each tissues, as variable vols for the regression task**(2)**; we decided to implement three different regressors for the regression task: Bayesian regression model, the Ordinary Least Squares model and the Lasso Regression**(3)**. The detailed results of the age regression is in the part A section.
In part B, we did the pre-processing on the feature space for the grey matter maps before doing the PCA dimension reduction for the feature space of the grey matter maps. After we finished the pre-processing part, we implemented the PCA dimension reduction for the feature space of the grey matter maps. Finally, we used different regressors to fit the training data of both of the image data

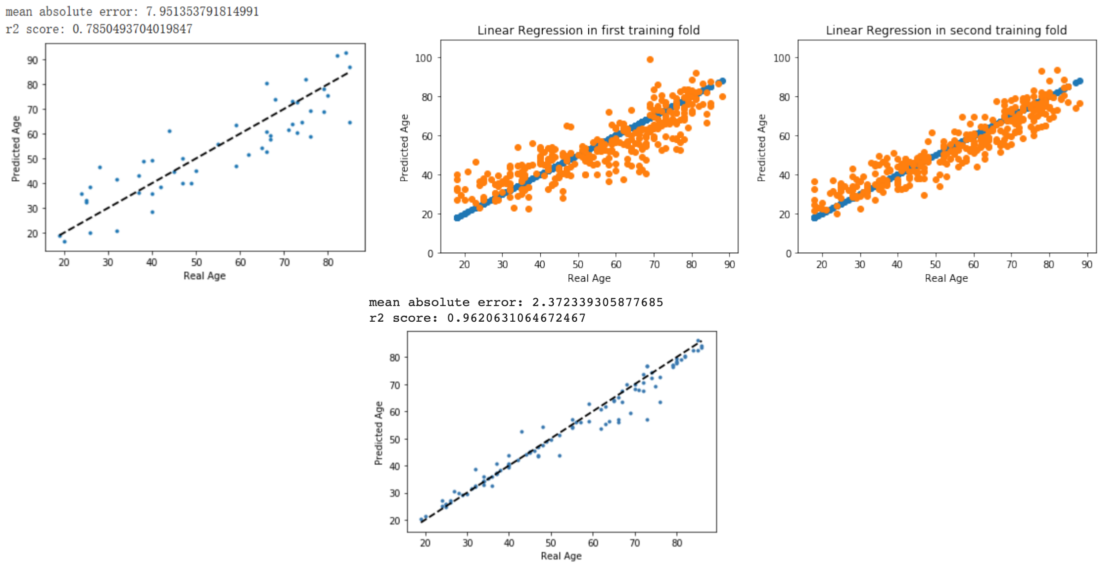

Figure 7: Scatter plots for age regression for three different approaches.

and age data including: **Linear Regressor, Polynomial Regressor, KNN Regressor**. Because, in general, the smaller MAE the better performance the model is and R2 score show the correlation of the prediction and real truth value, which indicates how stable the model is, the best regressor is the **Linear Regressor** due to the fact that it generates the lowest Mean Absolute error and r2. The detailed results of the regression is in the part B section.

In part C, we wanted to prove our idea: the result of FC LetNet is quite similar to LetNet and fully convolutional neural network indeed doesn't affect the original accuracy. So, we resampled the 3D grey matter maps at first and we divided the resampled 3D grey matter maps into 5 groups of size. All of the tissues were trying to fill the whole 3D Grey matter maps as much as possible. Then, we generated the corresponds MAE and r2 score graphs by using LeNet3D, which simply replaces the fatten layer (FC1 layer) to equal the output from Conv layer**(detail is in the part C section)**. Then, we started the training process by using the resampled 3D grey matter maps dataset and we applied the MinMaxScaler for the ages first and inverse the scale when calculating MAE. Finally, we came to the stage of evaluation. We applied three Neural Networks: **LeNet 3D, CovLeNet 3D and CustomCNN 3D** and proved that the result of FC LetNet is quite similar to LetNet and fully convolutional neural network indeed doesn't affect the original accuracy. The detailed results is in the part C section.