Script

March 21, 2019

1 This demo uses dataset of raw T1 structual MRI images collected by Steve.

2 After normalization.

3 With mean value subtracted

541 subjects in total.
from 6 to 85 years old.
10% for test, 90% for training.
Because of the incompatible in the pixdim and dim, we get rid of three images from the dataset:

- A00058503
- A00058952
- A00059344

Two images are missing according to the phenotypic information:

- A00040181
- A00039084

```
In [1]: %load_ext autoreload
        %autoreload 2
        %matplotlib inline
```

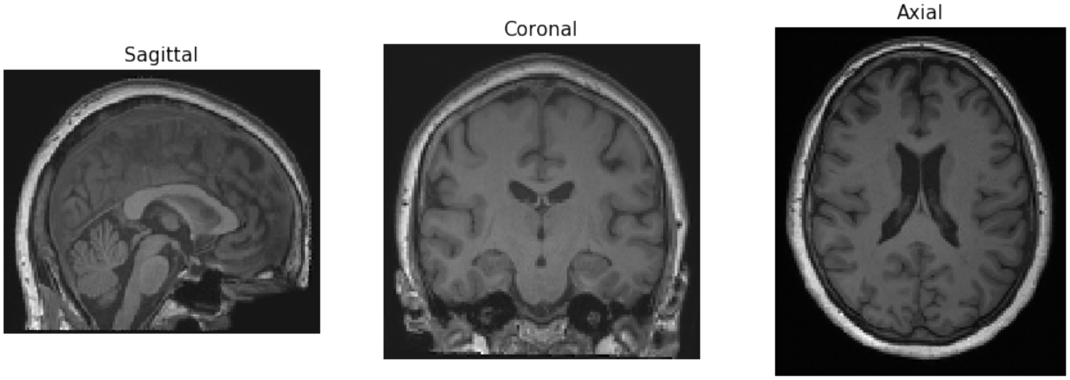
```
In [2]: from preprocess import *
```

3.1 1. The raw image

3.1.1 Some images looks like this:

```
In [3]: nii_img = nib.load('/media/woody/Elements/Steve_age_data/ANAT_normalized/A00008326.nii')
        npy_img = nii_img.get_data()
        print2d(npy_img)
```

```
Dimension:  (121, 145, 121)
```



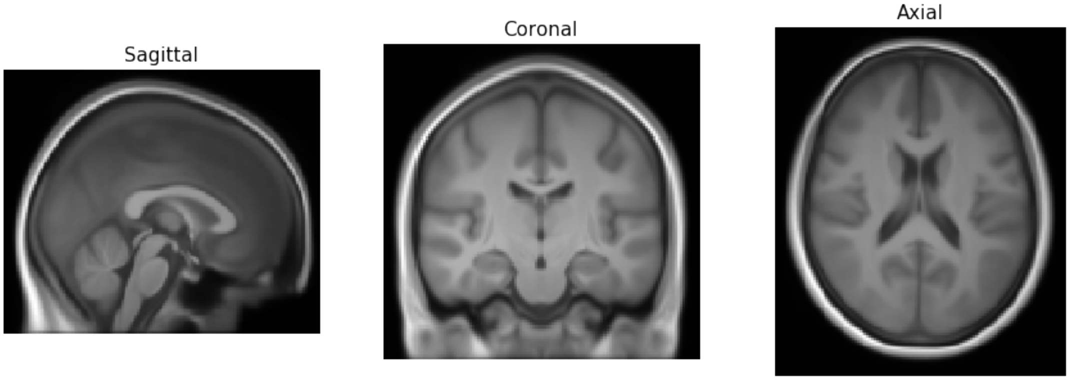
3.1.2 The mean image of all the training data is computed and is subtracted from all training and test data.

3.1.3 It is worth noting that the test data does not contribute to the mean image. This is because the training data, and only training data, needs to have zero mean for better training performance.

3.1.4 How does the mean values look like:

```
In [4]: mean_npy = np.load('./data_npy/mean_npy.npy')
        print2d(mean_npy)
```

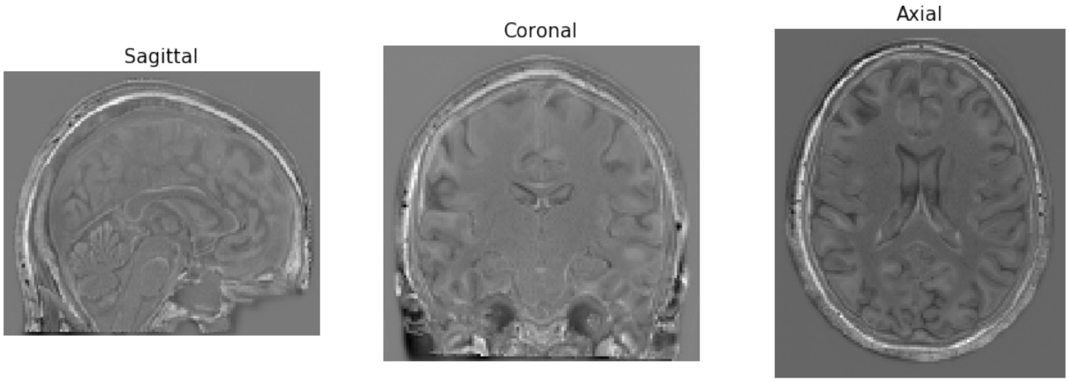
```
Dimension:  (121, 145, 121)
```



3.1.5 Finally, the images fed into the model look like this:

```
In [6]: final_img = np.load('./data_npy/mean_subtracted/A00008326.npy')
        print2d(final_img)
```

```
Dimension:  (121, 145, 121)
```



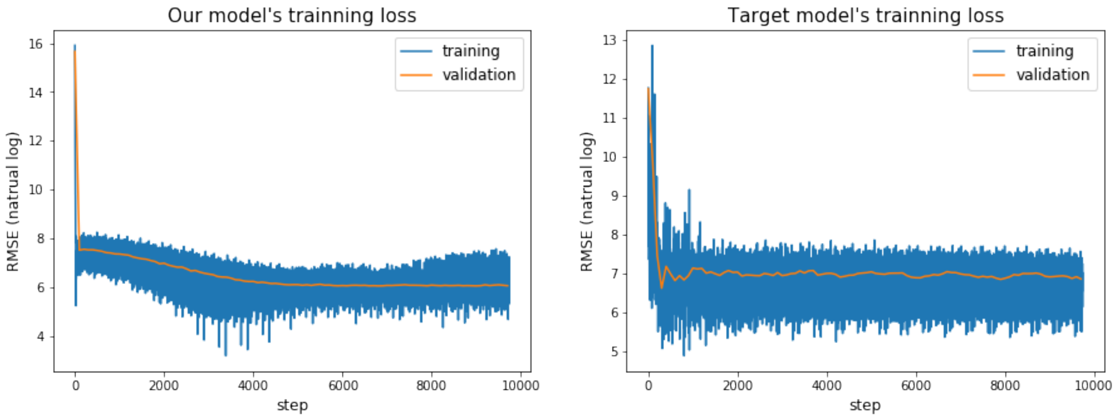
3.2 2. Training and test results

- We collect the best shot who has the smallest loss value (RMSE) in the training process.
- The validation uses the test dataset.
- Our model uses 3D-CNN with dropout, batch normalization and L2 regularization.
- Target model uses 3D-CNN with dropout.

```
In [7]: def subdraw(ax,filename):
            arr = np.load(filename)
            ax.plot(arr[0], np.log(arr[1]),label='training')
            ax.plot(arr[2], np.log(arr[3]),label='validation')
            ax.legend(fontsize=12)
            ax.set_xlabel('step',fontsize=12)
            ax.set_ylabel('RMSE (natrual log)',fontsize=12)

        f, (ax1, ax2) = plt.subplots(1, 2, figsize=(15,5))
        subdraw(ax=ax1,filename='./img/demo_1_1_pltdata_2019.03.21.15:49:07.npy')
        ax1.set_title('Our model\'s trainning loss',fontsize=15)
        subdraw(ax=ax2,filename='./img/demo_1_2_pltdata_2019.03.21.19:07:32.npy')
        ax2.set_title('Target model\'s trainning loss',fontsize=15)
```

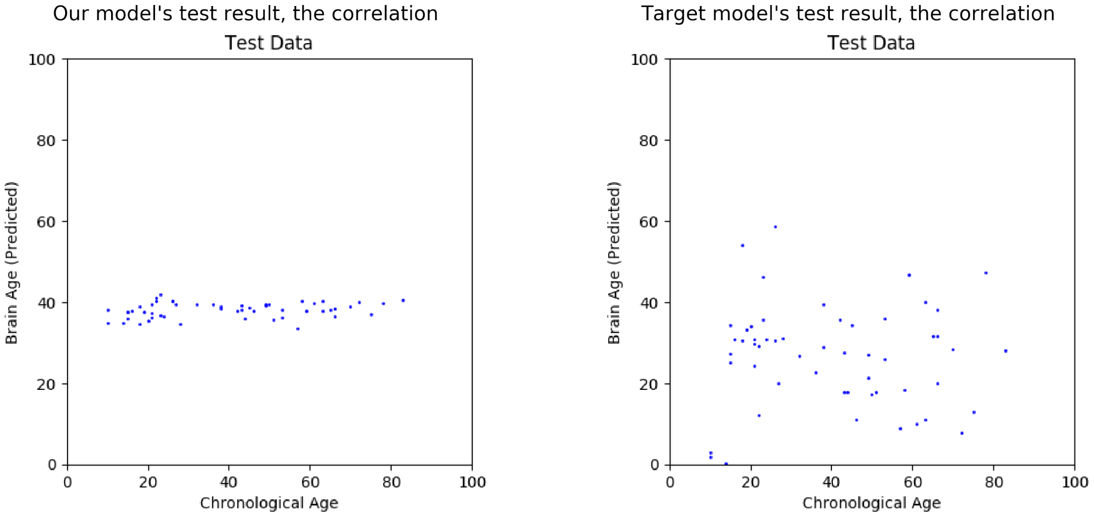
```
Out[7]: Text(0.5, 1.0, "Target model's trainning loss")
```



3.2.1 Let's check the correlation between the predicted age and the chronological age.

```
In [8]: f, (ax1, ax2) = plt.subplots(1, 2, figsize=(40,20))
        ax1.imshow(plt.imread('./img/demo_1_1_test.png'))
        ax1.axis('off')
        ax1.set_title('Our model\'s test result, the correlation',fontsize=40)
        ax2.imshow(plt.imread('./img/demo_1_2_test.png'))
        ax2.axis('off')
        ax2.set_title('Target model\'s test result, the correlation',fontsize=40)
```

```
Out[8]: Text(0.5, 1.0, "Target model's test result, the correlation")
```



3.3 Guess:

3.3.1 There's some possibility that in former demos the models suffer some overfitting problem.

3.3.2 Try to figure it out ...

```
In [ ]:
```