Predicting Brain Age with Convolutional Neural Networks

Claudia Greco

April, 2018

1. Abstract

The human brain changes across the adult lifespan, it is possible to identify the signs of aging in a brain MRI (magnetic resonance imaging) images and even to determine a brain age. The difference between brain age and chronological age can reveal conditions such as dementia, schizophrenia, diabetes, disorders that make the brain appear older. The goal of this project is to accurately predict the chronological brain age based on MRI data. We accomplished this by using a predictive modelling approach based on deep learning, specifically convolutional neural networks (CNN), and applied to neuroimaging data.

Section 2 introduces a more detail explanation of the problem that we addressed. Section 3 describes the process of data acquisition and data cleaning. Section 4 presents the details of the model as well as the data preprocessing and network topology. Finally, Sections 5 and 6 discusses the results and future work, respectively.

Keywords: Neuroimaging, Deep Learning, Convolutional Neural Networks, Brain Age.

2. Introduction

Determining brain age from an MRI scan has always been a time-consuming business. The analysis is long because the MRI data has to be heavily processed before it can be interpreter. This pre-processing includes the removal from the image of non-brain tissue such as the skull, the classification of white matter, gray matter, and other tissue, and the removal of image artefacts along with various data-smoothing techniques[1].

Now, brain age can be predicted very quicker based on neuroimaging data using deep learning approaches to model healthy brain ageing. If the predicted brain age of a person differs a lot from its chronological age, this difference can reflect a neurodegenerative condition.

The goal of this project is to accurately predict the chronological brain age based on structural MRI data, by using convolutional neural networks.

3. Dataset

All imaging data used were T1 weighted, structural magnetic resonance imaging (MRI) brain scans from the publicly available IXI dataset[2]. After cleaning and dropping some images, we got 493 healthy people, male and females, between 20 and 86 years old. Distribution is shown in Figure 1. None had any kind of neurological condition that might influence their brain age. Their brain age should match their chronological age.
More details below:

Male: 45.4% (224/493)
Female: 54.6% (269/493)
Mean age = 49.2 years
St Dev = 16.4 years
Min age= 20
Max age = 86 years

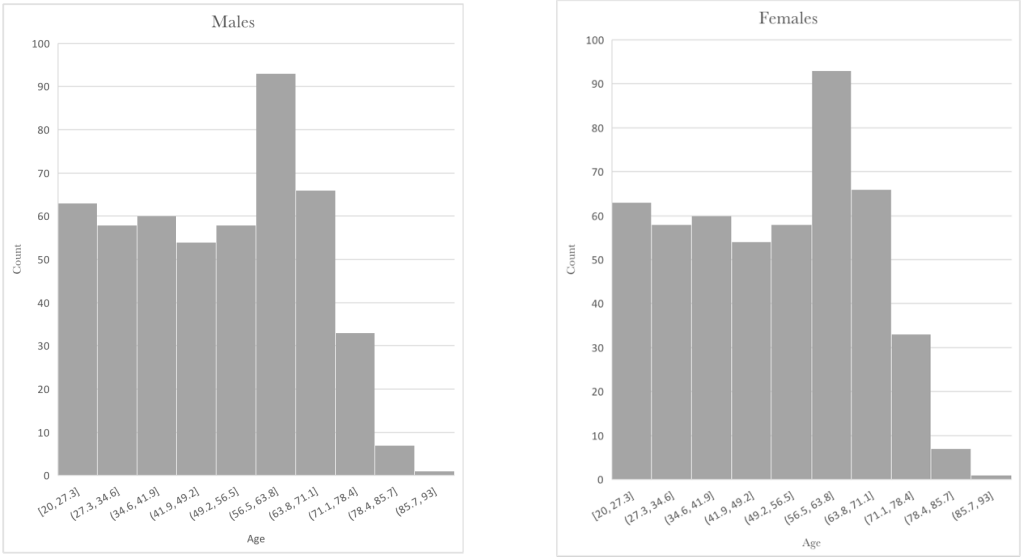


Fig 1. Dataset distribution by Age and Sex

4. Model

Convolutional Neural Network in `Keras` (v 2.1.4) was used to predict chronological age in healthy individuals. CNN can learn the correspondence between patterns in structural neuroimaging data and an age label, and then fit MRI images as independent variables to predict chronological age as the dependent variable.

The project was developed using Python v 3.6, Keras v 2.1.4, in a personal laptop MacOS High Sierra, 2.3 GHz Intel Core i5, 8GB 2133 MHz LPDDR3.

4.1 Preprocessing

Images were in Nifti format, and with a shape of (256,256,150). MRI images were loaded using NiBabel package. As a first approach, we decided to take one slice per volume, specifically slice number 130 and to reduce one dimension and use 2D Images for modeling. The decision of taking specifically 130 is because is the deep that shows a good picture of the brain.

The only data processing required for the deep learning machine is to ensure the consistency of the image orientation and the pixel dimensions between images. Examples of neuroimaging input data is shown in Figure 2.

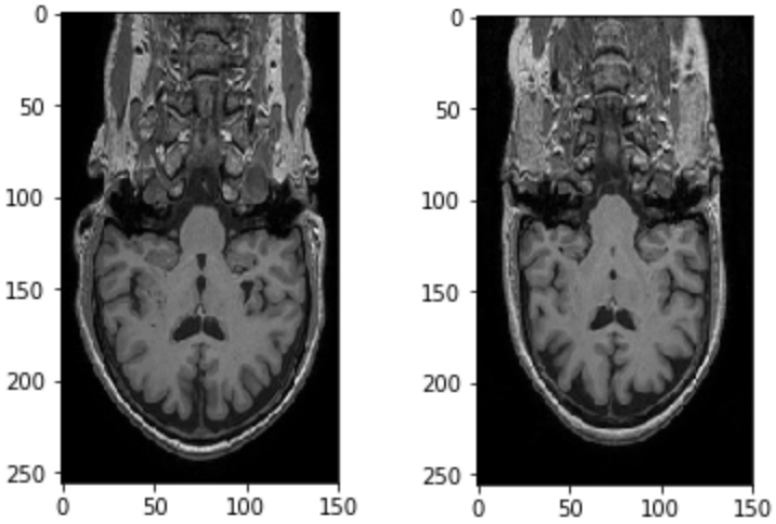


Fig 2. Left: MRI of a 20 years-old person. Right: MRI of a 86 years-old person

We train_test_split the data leaving 70% for training and 30% for testing, and used MinMaxScaler (sklearn.preprocessing) to scaling our pixels to (0,1). We build three models, one of them using ImageDataGenerator (keras.preprocessing.image) to do some preprocessing and realtime data augmentation, and the others two models without this step.

4.2 Topology

Our Convolutional Neural Network uses MRI grey scaled images of a size 256h × 150w as inputs. The output to be predicted is a single scalar representing the chronological age. The topology is described below and illustrated in Figure 3.

1. Input Layer: 2D Convolution with:
    - input shape = (256, 150, 1)
    - 90 filters
    - filter_size = kernel_size =(5, 5)
    - strides = (1,1)
    - activation function = ReLu

2. MaxPooling with:
    - pool size = (2, 2)
3. 2D Convolution with:
    - 180 filters
    - filter_size = kernel_size = (4, 4)
    - strides = (1,1)
    - activation function = ReLu

4. MaxPooling with:
    - pool size = (2, 2)

5. Dropout of 0.6

6. Flatten

7. Dense Layer with
    - 2048 Neurons
    - activation function = ReLu

8. Dropout of 0.6

9. Output Layer: Dense Layer with
    - 1 Neuron
    - activation function = ReLu

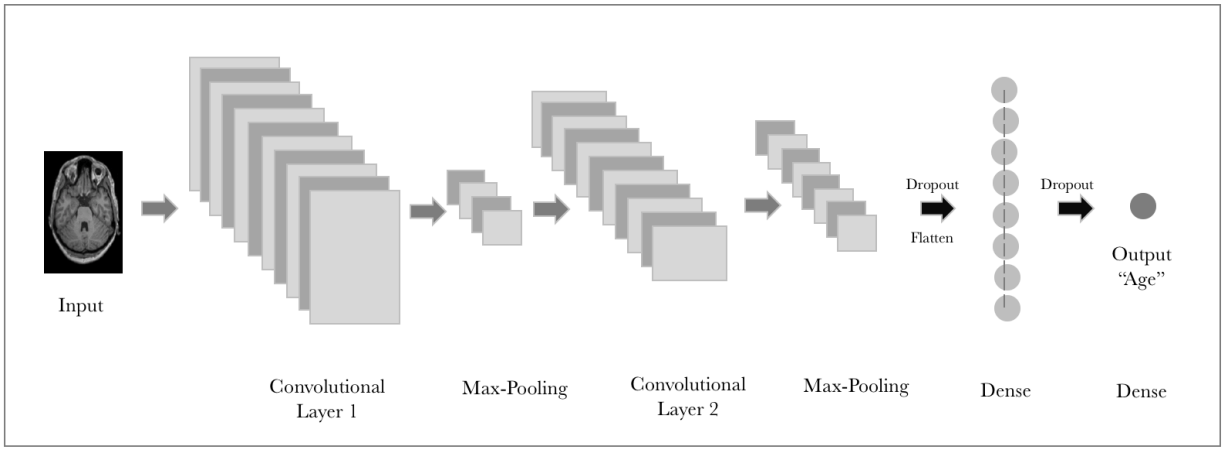


Fig 3. Network Topology

4.3 Modeling

We built three different models, all of them had the same topology and the network weights were trained by minimizing the mean absolute error (mae) using adam optimization algorithm, train on 345 samples (70%) and testing on 148 samples (30%).

Model 1: Our simplest model, no data augmentation and no hyper-parameter tunning.

Model 2: Used ImageDataGenerator for do preprocessing and real time data augmentation. All datasets were augmented by generating additional artificial training images. The data augmentation strategy consisted of performing rotation of 40 degree and randomly shift images horizontally and vertically by 0.01.

Model 3: For hyper-parameter optimization we used the wrapper Hyperas (Keras + Hyperopt) and defined hyper-parameter ranges to tune and used our best_model to predict.

	Model 1 No Data Augmentation No HyperParameters Tunning	Model 2 Data Augmentation No HyperParameters Tunning	Model 3 No Data Augmentation HyperParameters Tunning
	EarlyStopping: monitor='val_loss', min_delta=0.01, patience=10, mode='auto'	ImageDataGenerator rotation_range=40, width_shift_range=0.1, height_shift_range=0.1, horizontal_flip=True	Hyperas – optim Hyperas search space: hp.choice('Conv2D', [15,30]), hp.uniform('Dropout', 0, 1) max_evals=2
Batch Size	128	128	128
Epochs	50 – stopped at 47	50	50
Time for train	13min 14s	27min 51s	3hs 11min 25s
MAE **Baseline: 15.3 years**	10.8 years	16.18 years	11.9 years

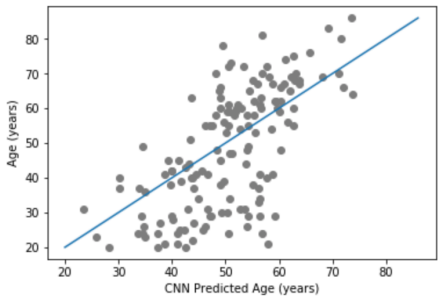
Fig 4. Models performance

5. Results

The best model was Model 1, with a mean absolute error of 10.8 years, and got a better performance than our baseline of 15.3 years. Details from Model 1 are shown in Figure 5.

In Model 2, data augmentation provided no benefits at all. We also hypothesis that the images are already clean that combining features don't really add any additional information.

Model 3 ran two times (max_evals = 2), looking for the parameters for a Conv2D layer and a Dropout. Parameters of best run were {'Conv2D': 30, 'Dropout': 0.3838088604298333} with a mean absolute error of 11.9 years.



Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 252, 146, 15)	390
max_pooling2d_1 (MaxPooling2	(None, 126, 73, 15)	0
conv2d_2 (Conv2D)	(None, 123, 70, 30)	7230
max_pooling2d_2 (MaxPooling2	(None, 61, 35, 30)	0
dropout_1 (Dropout)	(None, 61, 35, 30)	0
flatten_1 (Flatten)	(None, 64050)	0
dense_1 (Dense)	(None, 1000)	64051000
dropout_2 (Dropout)	(None, 1000)	0
dense_2 (Dense)	(None, 1)	1001

Total params: 64,059,621
Trainable params: 64,059,621
Non-trainable params: 0

None

Fig 5. Model 1: Left: Predictions vs True values. Right: model.summary()

6. Next Steps

Three main things should be done to better make predictions:

- First, get more data
- Second work with fully volumetric images and use 3D convolutions.
- Third, with more computational resources, increase hyper-parameters to the Hyperas search space.

References

https://arxiv.org/ftp/arxiv/papers/1612/1612.02572.pdf

[1] https://www.technologyreview.com/s/603112/deep-learning-machine-uses-mri-scans-to-determine-your-brain-age/

[2] http://brain-development.org/ixi-dataset/