Bone Age Prediction

Abhishek Nimmakayala, Akhil Kumar Baipaneni

Machine Learning 2

The George Washington University

Table of Contents

1. Introduction

1.1 Background on Bone age

A bone age study helps doctors estimate the maturity of a child's skeletal system. It's usually done by taking a single X-ray of the left wrist, hand, and fingers. It is a safe and painless procedure that uses a small amount of radiation. A child's bone age (also called the skeletal age) is assigned by determining which of the standard X-ray images in the atlas most closely match the appearance of the child's bones on the X-ray. A difference between a child's bone age and his or her chronological age might indicate a growth problem

The bone age study can help evaluate how fast or slowly a child's skeleton is maturing, which can help doctors diagnose conditions delay or accelerate physical growth and development. This test is usually ordered by pediatricians or pediatric endocrinologists

1.2 Problem statement

In this project we aim to train a model on the x-ray images of children's hands and predict the bone age of children, by doing so we can find any abnormalities in the growth of a child's bone and also find difference between a child's bone age and his or her chronological age which might indicate a growth problem

1.3 Motivation

There were two main aspects which motivates us to take up this project.

a) We can identify any abnormalities/growth problems in a child's growth
b) Since the bone age is assigned by a radiologist (a human) we can automate this process in the future and therefore overcome the errors made by human.

2. Methodology

2.1 Dataset description

The data set is obtained from kaggle and holds 9 GB of data. A total of 12611 images are available which are of good resolution (1514*2044 pixels). Along with the images there is a .CSV file which has image id, bone age, gender information.

2.2 Data preprocessing

We have used CV2 to resize all the images to 50*50. We used image data generator to generate more images. The number of training images before using data generator were 10088 and we generated 20,176 using image data generator. The following parameters were used in image data generator.

rotation_range=20,
width_shift_range=0.2,
height_shift_range=0.2,
zoom_range=0.3,
horizontal_flip=True,
brightness_range=[0.5, 1]

2.3 Data Visualization

The csv file contained the gender of the child along with their bone age, so we visualized the data to get a better understanding.

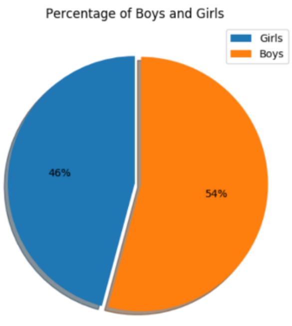


Figure 1.1

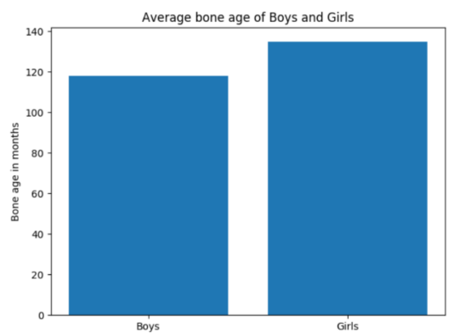


Figure1.2

From figure 1.1 we can observe that the total number of boys and girls in the dataset are almost equal. Figure 1.2 depicts the average bone age of boys and girls it was observed that the average bone age of girls was around 138 month and the average bone age of boys was 118 months.

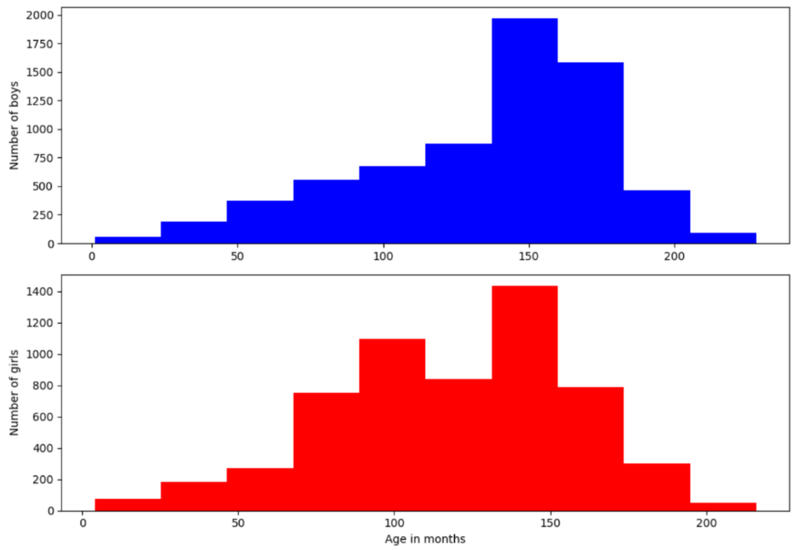


Figure 1.3

Figure 1.3 demonstrates the number of observations for each bone age for both girls and boys. It was seen that for boys bone age of 150 months had the highest number of observations (1900), It was the same case with girls as well bone age of 150 months had the highest number of observations (1300). For girls we had the least number of observations for bone age of 200 months while for boys it was 225 months.

2.4 Data Modelling

We have used 3 models in this project. The 1st model is a simple MLP, while the 2nd is a CNN and lastly the 3rd model is a Xception pre-trained network. Since we are trying to predict the bone age which is a continuous value we have used Mean squared error and Mean absolute error as our metrics to evaluate the model.

Model 1 Architecture:

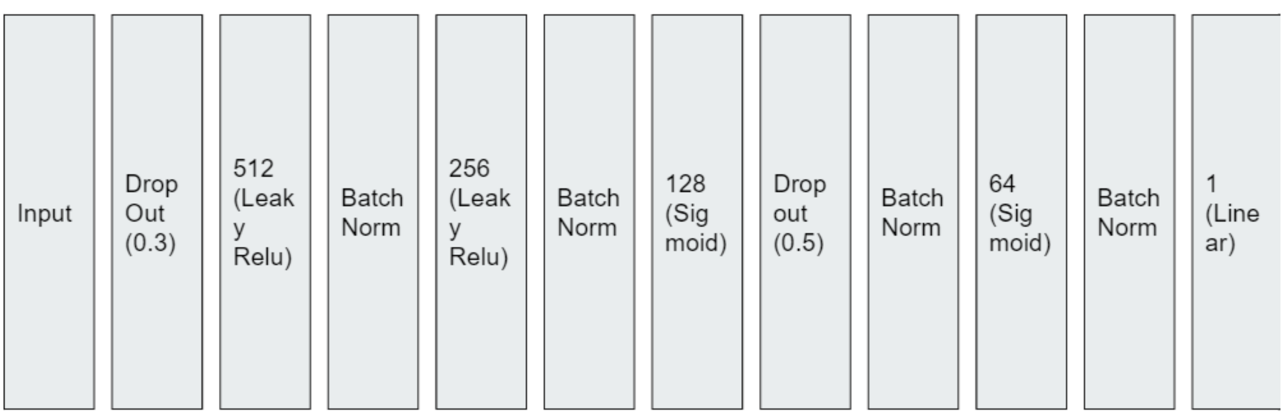


Figure 2.1

Figure 2.1 describes the simple MLP model's architecture. We have used a dropout in the 1st layer and 7th layer. There is a batch normalization layer in 3,5,8 & 10. Layers 2 & 4 have 512 and 256 neurons respectively and they have Leaky Relu as the activation function. We have used Laeky relu to overcome the dead activation problem in Relu. Layer 6 & 9 have 128 and 64 neurons with Sigmoid as the actication unit and the last layer is a linear layer with 1 neuron as the target is a continuous value.

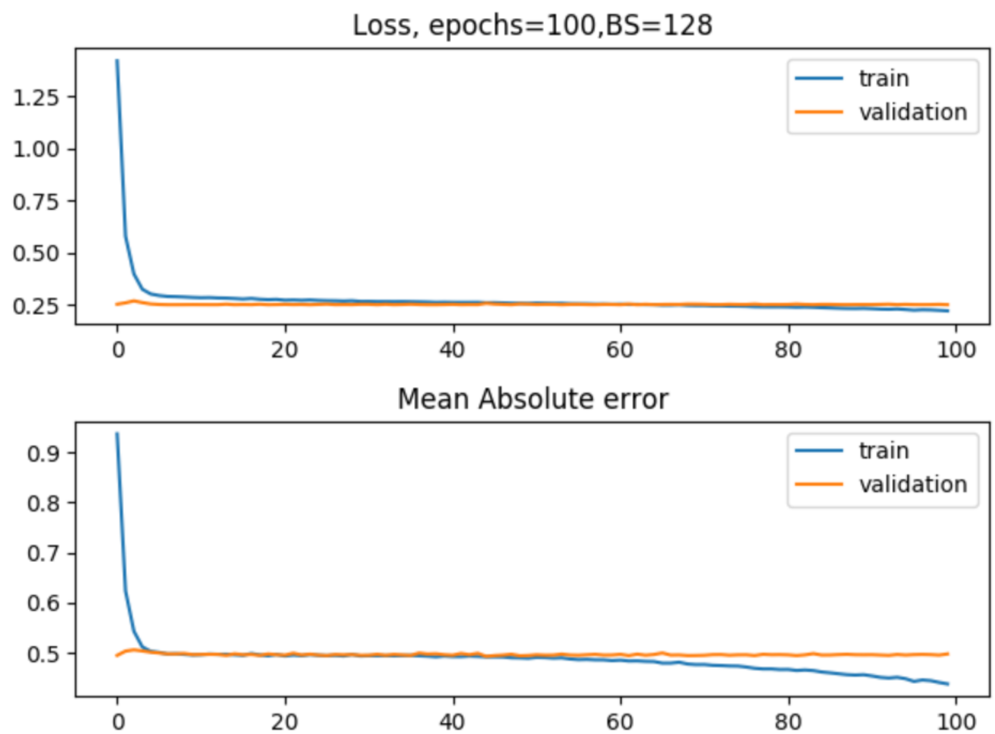


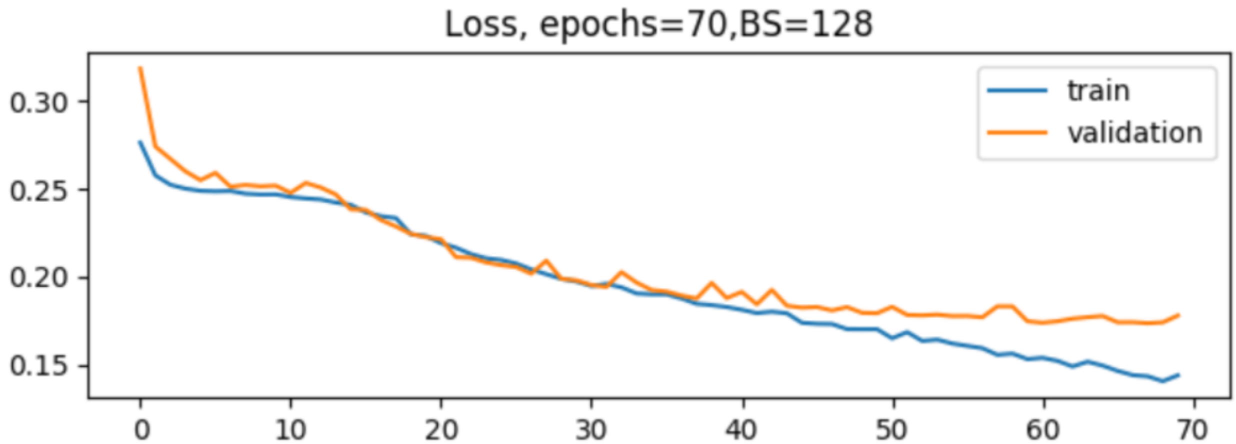
Figure 2.2

Figure 2.2 shoes that the training loss and Mean absolute error were almost constant which indicates that this model is not good enough to solve our problem. To have better results we used a CNN model next.

Model 2 Architecture:

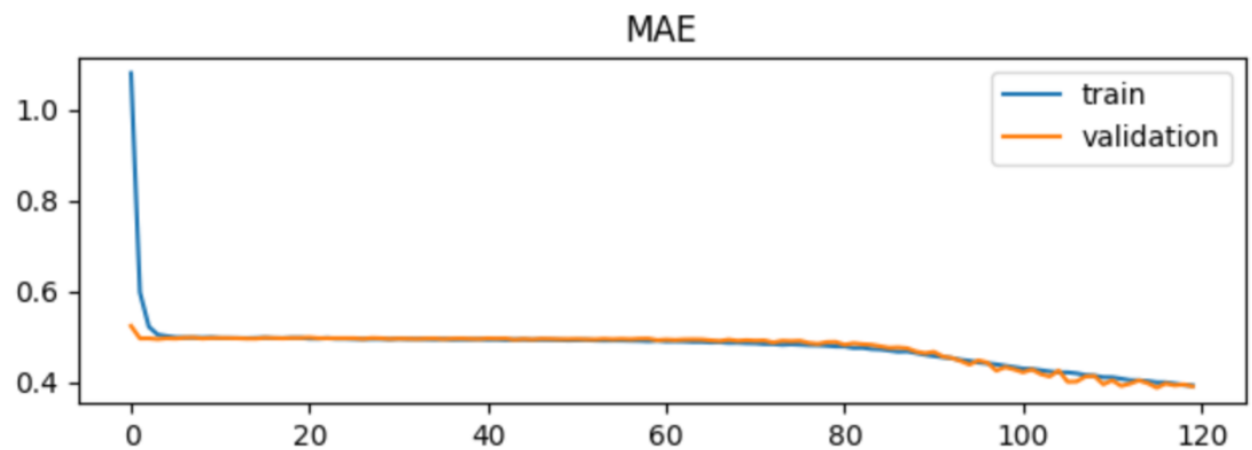
Layer (type)	Output Shape	Param #
conv2d_7 (Conv2D)	(None, 50, 50, 32)	896
conv2d_8 (Conv2D)	(None, 48, 48, 32)	9248
max_pooling2d_4 (MaxPooling2	(None, 24, 24, 32)	0
dropout_7 (Dropout)	(None, 24, 24, 32)	0
conv2d_9 (Conv2D)	(None, 24, 24, 64)	18496
conv2d_10 (Conv2D)	(None, 22, 22, 64)	36928
max_pooling2d_5 (MaxPooling2	(None, 11, 11, 64)	0
dropout_8 (Dropout)	(None, 11, 11, 64)	0
conv2d_11 (Conv2D)	(None, 11, 11, 128)	73856
conv2d_12 (Conv2D)	(None, 9, 9, 128)	147584
max_pooling2d_6 (MaxPooling2	(None, 4, 4, 128)	0
dropout_9 (Dropout)	(None, 4, 4, 128)	0
flatten_3 (Flatten)	(None, 2048)	0
dense_8 (Dense)	(None, 512)	1049088
dropout_10 (Dropout)	(None, 512)	0
dense_9 (Dense)	(None, 1)	513

The model 2 architecture is more robust than that of the model 1 architecture, as we can see there are more number of layers and also this model was trained for larger number of epochs. A total of 16 layers were used in the CNN model. This model has more number of dropout layers compared to the simple MLP model. Also there are three max-pooling layers in layer 3,7 & 11 respectively. The drop out layers were placed immediately after the max-pooling layers at 4,8 & 12 respectively. The CNN model results seem to be more reliable.



Number of epochs

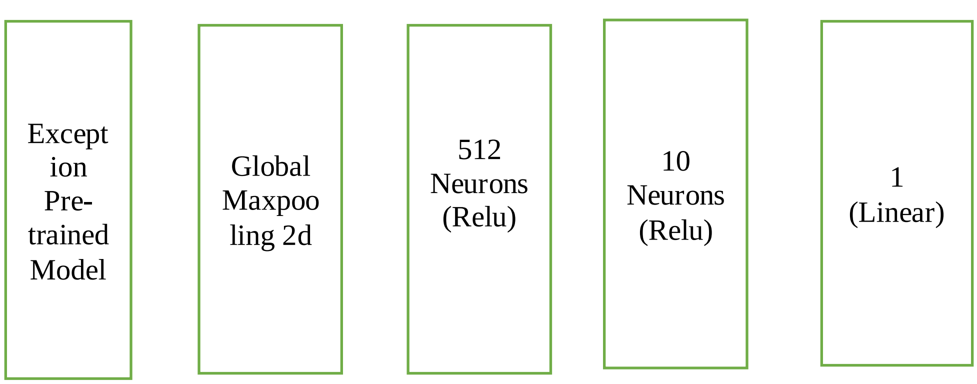
From the figure above we can clearly see that the validation loss is decreased as the number of epochs increased also. The validation loss decreases as the training loss decreases, this is a good indication that this CNN model is learning well.



Number of epochs

We can also see that the validation Mean absolute error also starts to coincide with the training mean absolute error as the number of epochs were increased.

Model 3 Architecture:



The above is the architecture of the pre-trained model. We have used global max-pooling in the 2nd layer and the 3rd and4th layers have 512 and 10 neurons with activation function as Relu. The last layer has 1 neuron with Linear as the activation function since the target is a continuous variable.

```
Epoch 37/200
253/252 [==============================] - 312s 1s/step - loss: 0.0873 - mae_in_months: 9.5760 - val_loss: 0.0911 - val_mae_in_months: 9.6506
Epoch 38/200
253/252 [==============================] - 311s 1s/step - loss: 0.0832 - mae_in_months: 9.3826 - val_loss: 0.0937 - val_mae_in_months: 9.8625
Epoch 39/200
253/252 [==============================] - 311s 1s/step - loss: 0.0834 - mae_in_months: 9.3657 - val_loss: 0.1090 - val_mae_in_months: 10.5230
Epoch 40/200
253/252 [==============================] - 311s 1s/step - loss: 0.0824 - mae_in_months: 9.3180 - val_loss: 0.0996 - val_mae_in_months: 10.2073
Epoch 41/200
253/252 [==============================] - 311s 1s/step - loss: 0.0782 - mae_in_months: 9.0573 - val_loss: 0.0913 - val_mae_in_months: 9.8012
Epoch 42/200
253/252 [==============================] - 312s 1s/step - loss: 0.0749 - mae_in_months: 8.7894 - val_loss: 0.0852 - val_mae_in_months: 9.3645
```

Figure 3.1

From Figure3.1 we can observe that the loss for this model was the best compared to all the other models.

3. Conclusions

- Out of the 3 models used Model 3 (Xception Pre-trained model) had the best results. The Mean absolute error was the least for Model 3.
- The second best model is the CNN model. The CNN model has a validation loss of 0.17, validation MAE of 0.34 and validation MSE of 0.18
- The Pre trained model had the least loss when compared to all the models
- The least performed model as expected was model 1 (the simple MLP model)

3.1 Limitations

- We generated only one extra image for every training image (double the training images) by using Image data generator, as we try triple or quadruple the number of training images using image data generator we had out of memory issues.

4. Future work

- The .csv file has gender information along with the bone age. But for all the 3 models we have used only the image as the training data, in the future we would like to use the gender information along with the image to train the models.

5. References:

Below are the resources we found very useful for developing our project.

- **https://kidshealth.org/en/parents/xray-bone-age.html**