CENG 483 - Introduction to Computer Vision Spring 2017-2018 Take Home Exam 2 Age Estimation from Facial Image

Duygu Dogan
e1941962@ceng.metu.edu.tr

***Abstract*—This report aims to analyze implemented an age estimation system based on deep learning methods and come to a conclusion by a comparison.**

***Index Terms*—Age estimation, Deep Learning, PyTorch**

I. Introduction

The main purpose of age estimation systems is to determine age of the person in a query image.

Here in this report, methodologies of age estimation system will be explained firstly. Then, the process of selecting parameters and their effects will be told. Lastly, finding better ways to improve the performance will be observed, compared and concluded.

II. Used Technologies

- Python 3.6
- Anaconda
- Jupyter Notebook

III. Implementation

The code that this report based on have been borrowed from available resources. [1]

In order to implement an accurate age estimation system and to observe the effects of layers and parameters for our optimizer, the base code was run with layer_number = {0,1,2,3} and different learning rates & epoch numbers which will be explained on sections below.

IV. Results and Interpretation

A. Rationale Behind the Parameter Choices

a) Number of Epochs:

As can be seen from Table 1, when epoch size equals to 10000, it causes over-fitting (training loss is getting smaller, yet validation loss is getting greater.) In order to prevent this, by comparing the accuracies and the difference between training and validation loss of different epoch sizes as well, 2000 is the most accurate number to be chosen as epoch size for layer0.

It also can be seen from Figure 2, when epoch number gets greater than 2000, the difference between training and validation loss gets greater as well. In order to avoid this, 2000 is the most rationale choice for epoch size to be picked for Layer 1 as well.

Layer 2 and 3 was also implemented by this rationality. Thus, 2000 was chosen as epoch number for all layers.

epoch size	training accuracy	validation accuracy	training loss	validation loss
500	0.5666	0.587	157.8924	185.4425
1000	0.5938	0.5815	147.1486	177.0235
2000	0.6036	0.5865	142.1042	174.8374
3000	0.6062	0.587	139.7753	174.2244
5000	0.6458	0.563	137.5090	170.1258
10000	0.6206	0.5815	135.4689	175.3012

TABLE I: Effects of Epoch Number for Layer 0

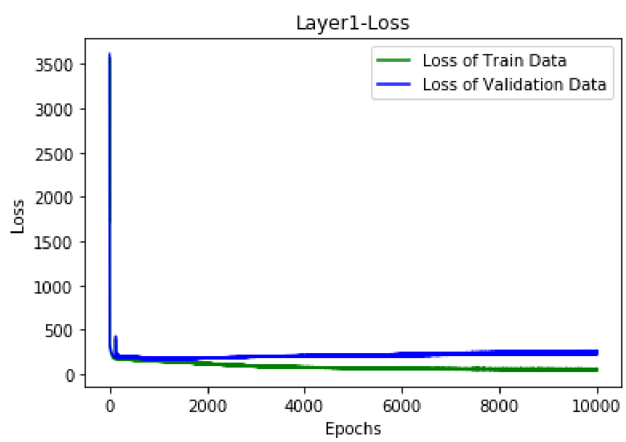


Fig. 1: Over Fitting on Layer 1

b) Learning Rate:

As can be interpreted from Table 2, when learning rate gets larger than 0.01, it causes some disagreements. After 0.01, learning rate affects accuracies and loss histories negatively. Besides, the difference between training and validation loss is getting greater and that can be interpreted as converging to over-fitting. Also, since the difference between accuracy and loss history of 0.005 learning rate and those of 0.01

learning rate is negligibly small, choosing 0.01 as learning rate parameter is more rational by comparing the difference between training and validation loss.

learning rate	training accuracy	validation accuracy	training loss	validation loss
0.005	0.6172	0.583	139.3228	170.8780
0.01	0.6036	0.5875	142.0253	170.6348
0.03	0.523	0.5645	173.0030	210.0635
0.05	0.522	0.4105	234.9851	255.7691
0.1	0.2658	0.184	435.7170	643.3201

TABLE II: Effects of Learning Rate for Layer 0

c) Nodes:

Table 3 represents the effects of changing node numbers for Layer 1. Choosing 256 as node number seems more rationale than choosing the others by looking at the accuracies. However, validation loss is more than twice the training loss and the difference between validation accuracies of node size = 256 and node size = 64 is negligibly small. Thus, by comparing accuracies, loss histories and the loss differences of training and validation, 64 has been chosen as node number of layer 1.

As can be seen from Table 4, validation losses are quite similar for all node pairs. For that reason, most accurate node numbers can be chosen by looking at validation accuracies. Thus, node numbers for layer 2 was chosen as (128,64).

By interpreting this process of layer 1 and 2, we can easily conclude that decreasing the node numbers gradually is the most effective method in order to get more accurate results. This method can be applied to Layer 3 in order to chose more rationale node numbers as well. Thus, (128,64,32) was picked as node numbers for Layer 3.

node	training accuracy	validation accuracy	training loss	validation loss
256	0.7096	0.6215	86.9202	192.1315
128	0.7086	0.6155	92.7371	191.1053
64	0.696	0.6175	109.0660	183.6473
32	0.7142	0.589	106.8716	180.7302
16	0.6976	0.5865	128.0664	167.4126

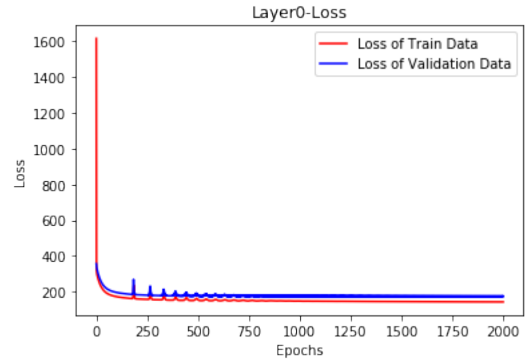
TABLE III: Effects of Nodes for Layer 1

node	training accuracy	validation accuracy	training loss	validation loss
256,128	0.6552	0.534	160.4240	180.2103
128,64	0.6852	0.6235	100.2939	192.8039
128,16	0.7036	0.6075	97.4649	183.9501
64,32	0.6714	0.6125	109.7315	192.8489
32,16	0.7112	0.6175	94.5625	187.3507

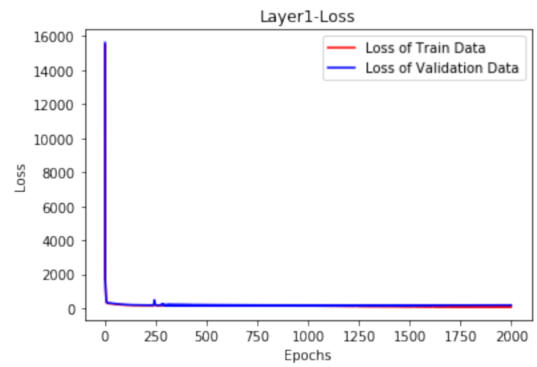
TABLE IV: Effects of Nodes for Layer 2

B. Effects of the Number of Layers

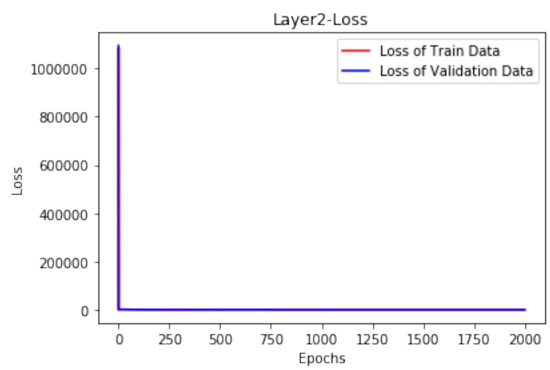
Figure 2 is based on the parameters that have been chosen and explained above. All of those layers on the figure have higher learning rates. For the functions with higher learning rates, parameters are bouncing around chaotically, unable to settle in a nice spot in the optimization landscape. [2]



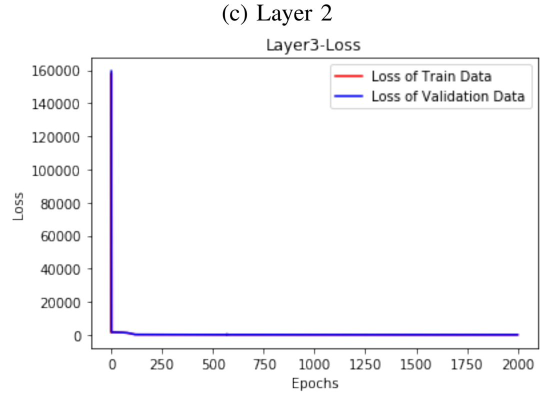
(a) Layer 0



(b) Layer 1



(c) Layer 2



(d) Layer 3

Fig. 2: Loss Histories of Different Layers

C. *Self Portrait*

The self portraits on the Figure 3 were tested with the trained model and the results can be seen on Table 5.

(a) Self Portrait 1 (b) Self Portrait 2

Fig. 3: Self Portraits

layer	estimated age for 1st portrait	estimated age for 2nd portrait
0	37.5439	35.5466
1	38.6970	31.7833
2	32.5321	30.4000
3	32.1033	30.4775

TABLE V: Estimated Ages for Two Portraits

The results are more different than expected. Layer 1 gave the most erroneous result surprisingly and it can be resulted from the parameter choices. Also, the difference between third layer and others was expected more than it is now.

By looking at the estimated age difference between two portraits, we can assume that either the corresponding ages of the images which contain glasses in the training data are smaller than the others or the features around eye area affect the age in a negative way.

V. Conclusion

As can be seen on Table 6, validation and training accuracies are highly similar yet, both are low for all layers. That can be solved by applying parameter adaptive learning rate methods and some hyper-parameter optimization.

layer	training accuracy	validation accuracy
0	0.6036	0.5865
1	0.696	0.6175
2	0.6852	0.6235
3	0.5916	0.6145

TABLE VI: Accuracies for Different Layers

References

[1] (n.d.). Retrieved May 14, 2018, from https://nbviewer.jupyter.org/url/user.ceng.metu.edu.tr/g̃cinbis/courses/Spring18/CENG483/slides/pytorch2_neuralnets.ipynb

[2] Convolutional Neural Networks for Visual Recognition. (n.d.). Retrieved May 14, 2018, from http://cs231n.github.io/neural-networks-3/#baby