Age Estimation From Facial Image

Onur Tirtir
Department of Computer Engineering
Middle East Technical University
Ankara, Turkey
onur.tirtir@ceng.metu.edu.tr

***Abstract*—Purpose of this work is to make age estimation for a given image. The Age Estimation (AE) system we devised is based on processing feature vectors -retrieved by using *ResNet-18*- with a series of fully-connected layers with *Rectified Linear Units(ReLU)*.**

***Index Terms*—Age Estimation, ResNet-18, Rectified Linear Units, RMSprop, Zero-One Loss, Mean Squared Error(MSE), Fully-Connected Layer, L2 Penalty, Dropout**

I. Introduction

The pipeline of our system is composed of ResNet-18 and neural network to process feature vectors retrieved from ResNet-18. For each image, we derived a feature vector -size of 512- for both to present a semantic meaning along with these images and not to interpret images as just an accumulation of pixels. We focused only on design of neural network mentioned above. After retrieving feature vectors, we experimented several neural network models. These models are composed of $0,\ 1,\ 2\ and\ 3$ fully-connected layers, i.e no dropouts are applied between layers. We applied ReLU after each fully-connected layer except the last one. Trivially, it seems not to be a harmful decision to truncate age estimations done negatively to 0 but that would deceive our neural network reducing the error factitiously for negative estimations. We preferred to use *RMSprop Optimizer* to optimize our neural network with several configurations. We have done mini-batch training to achieve memory limits and get a chance for faster convergence [3]. We will use MSE loss to measure how far we are from ground truth and Zero-One accuracy at some points to measure our accuracy on validation set. In the following sections, we will describe how the key decisions affected our experiments' accuracy for validation set. These sections are highly inter-laved because of the nature of neural network training. To explain, we will also inspect effect of increasing $nepochs$ while we are observing behaviors of several neural network configurations in following section. We will also examine suitability of the default $lrate = 0.001$ and $wdecay = 0.0$ parameters for different number of layers and their several configurations, in addition to the subsections specially experimenting them.

II. Overview on Key Decisions

We defined following objectives as crucial to be well-tuned for our experiments:

i Number of layers
ii Number of neurons, $nneurons$ for each layer
iii Number of training epochs, $nepochs$
iv Learning rate, $lrate$ (as a parameter of RMSprop Optimizer)
v Weight decay, $wdecay$ (as a parameter of RMSprop Optimizer for L2 penalty)
vi Mini-Batch size, $bsize$.

III. Layer Configuration

In this section, we kept parameters other than number of layers and neurons for each layer constant such that $lrate = 0.001$, $wdecay = 0.0$ and $bsize = 32$conventionally. Whenever we needed to add another layer, we simply have chosen $nneurons$ as multiplies of 2 again by convention. To find optimal number of epochs to train we inspected plots. These plots can be interpreted such that:

i Both training and validation losses are decreasing magnificently up to some number of epochs. (possible underfit region)
ii Then training loss continues to decrease but validation loss either remains stable of increases over epochs (possible overfit region)

A. No Hidden Layers

Observing Fig. 1, losses with training and validation sets seem to be strictly decreasing to some point as we increase $nepochs$. We expect for our model to overfit to training data as we increase $nepochs$. On the contrary, we see that after some point, loss for training data almost does not decrease, even. We have just an accuracy of 0.573 for this configuration. Hence this configuration directs us to a very inadequate model to fit to training set. We also derived that $lrate = 0.001$ looks like higher than it should be. However, since weakness -no hidden layer- of this model is obvious, it is not a candidate model for us.

B. One Hidden Layer

For this subsection, we experimented different number of neurons -64, 128, and 256-. We will observe Fig. 2 along this subsection. We see that as the number of neurons increase, loss retrieved from training set is decreasing while the loss obtained from validation set is increasing. This leads us to think overfit case as *the gap [2] between training -validation set loss* is increasing obviously. Another observation is that as we decrease number of neurons -64-, we get a higher loss for training set while we get slightly lower loss for

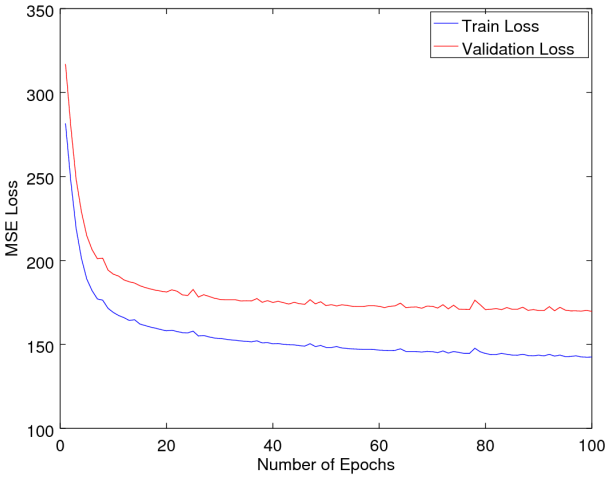

Fig. 1: MSE Loss for Training and Validation Sets

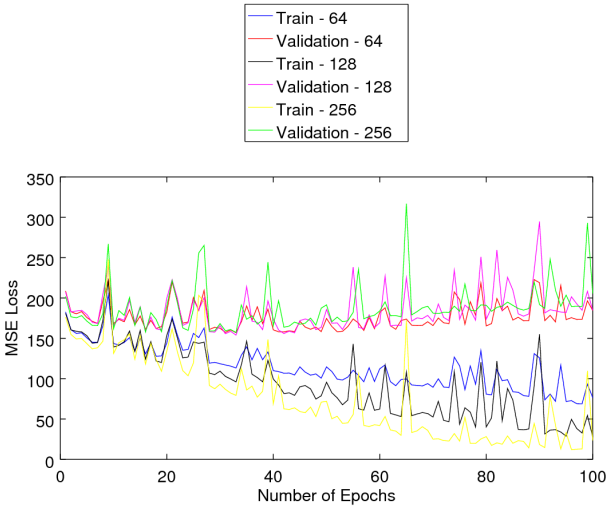

Fig. 2: MSE Loss For Varying Number of Neurons For One Hidden Layer

validation set comparing with other two configurations. So we derived that increasing number of neurons for a one layer architecture causes more over-fitting to training data. Inspecting optimal $nepochs$, we have 0.630 accuracy for 64 neurons with $nepochs = 45$, 0.627 accuracy for 128 neurons with $nepochs = 30$ and 0.629 accuracy for 256 neurons with $nepochs = 30$. We see that approximately (± 10) around these $nepochs$ we have a local maximum for accuracy on validation set. Having to do that much training does not seem to be reasonable but brings the insufficiency of our model into our mind with only *one hidden layer*. Hence we should try on more complex models. In addition to observation on overfit made in previous subsection, difference between *gaps* [2] is not only because of possible overfit of more neurons but underfit of 64 neurons. We also derived that $lrate = 0.001$ appears to be low for this layer configuration considering not all the 100 epochs but first 50 epochs as an upper bound to optimal $nepochs$ values just found. Observing plots, we see a higher $lrate$ would have been a better choice considering rough shape of plots [1]. We choose the configuration giving the optimal accuracy -one hidden layer with 256- as a candidate configuration, as it better fits to training set -possibly as it have more neurons than the others-.

C. Two Hidden Layers

In this subsection, we kept number of neurons in first hidden layer constant (256) and experimented for different number of neurons $(64,\ 128\ and\ 256)$ in second hidden layer. We will inspect Fig. 3 along this subsection. As we see, 3 configurations *two hidden layer* case behave similar to each other. Different than the experiments in previous subsection, the *gap* [2] between training and validation loss is more obvious. Also, default $lrate$ seems to be more suitable for these configurations [1]. Inspecting $nepochs$ for optimal accuracies for these 3 configurations; we have 0.624 accuracy for $256,\ 64$ neurons with $nepochs = 14$, 0.631 accuracy for $256,\ 128$ neurons with $nepochs = 11$ and 0.628 accuracy for $256,\ 256$ neurons with $nepochs = 8$ for these 2 hidden layers. Comparing these, we derived that as the number of neurons in second hidden layer decreases, $nepochs$ to get optimal accuracy increases. This is reasonable since a larger layer fits to training data more quickly as it is more capable to memorize what it learned. Since that is the only obvious difference between these 3 configurations, the configuration having $256,\ 256$ neurons is determined as a candidate configuration. Observing plots, we see $lrate$ seems to be appropriate [1].

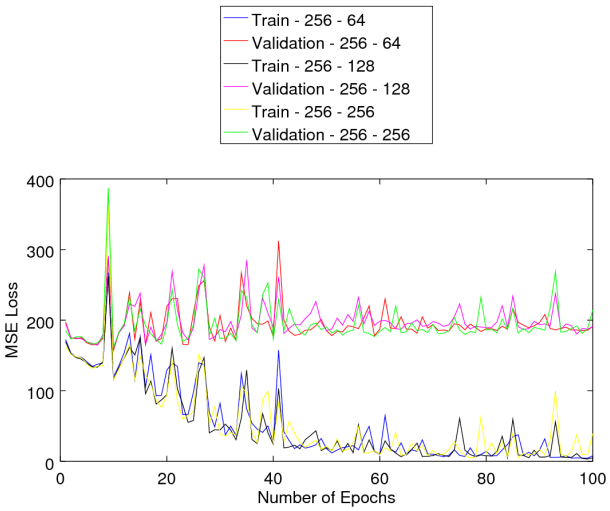

Fig. 3: MSE Loss For Varying Number of Neurons In Second Hidden Layer

D. Three Hidden Layers

We will inspect Fig. 4 along this subsection. $nneurons$ as $256,\ 128,\ 64$ respectively for each hidden layer to experiment

the effect of gradually summarizing the features. Again we observe same overfit problem -called as gap [2]- in related figure, as in previous subsection. Plots in that figure have a very similar behavior with the ones in previous subsection. Differently, We have 0.634 accuracy with $nepochs = 11$.

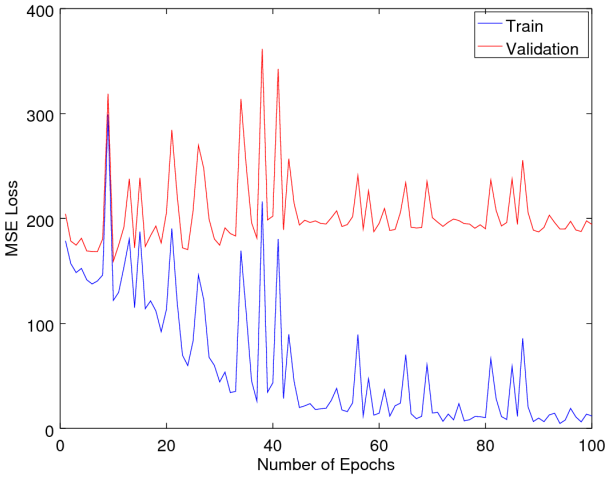

Fig. 4: MSE Loss For Three Hidden Layer Summarizing Architecture

E. Overall Observations

Unfortunately we still have a considerable *gap* [2] between validation and training set losses -mostly the case for high number of $nepochs$ for our experiments- is actually a poor training problem. Considering $nepochs$ around 10 based on above subsections, it would be appropriate to eliminate *one hidden layer* candidate. We already had eliminated *no hidden layer* architecture in its related subsection. Remaining two candidates, we have chosen the *three hidden layer* with $nneurons$ $256,\ 128,\ 64$ respectively and $nepochs = 11$ of training as **final architecture** in further sections as it gave a slightly better accuracy for validation set. The poor training problem may also arise from having a training set of limited size, using a straight fully connected neural network -i.e no dropouts- and having insufficient number of hidden layers or inappropriate choice of $lrate$, $wdecay$ and $bsize$ parameters. In further experiments, we will *tune* these last 3 parameters.

F. Learning Rate

In this subsection, we kept parameters other than $lrate$ constant such that $wdecay = 0.0$, $bsize = 32$ and using *three hidden layer* with $nneurons$ $256,\ 128,\ 64$ -according to above section- and experimented validation set accuracies over different $lrate$s. Observing rough shape of MSE Loss on training set in Fig. 4 and validating in Fig. 5 according to Zero-One accuracy, $lrate = 0.001$ is the optimal $lrate$. From now on, we will use $lrate = 0.001$ in our **final architecture**.

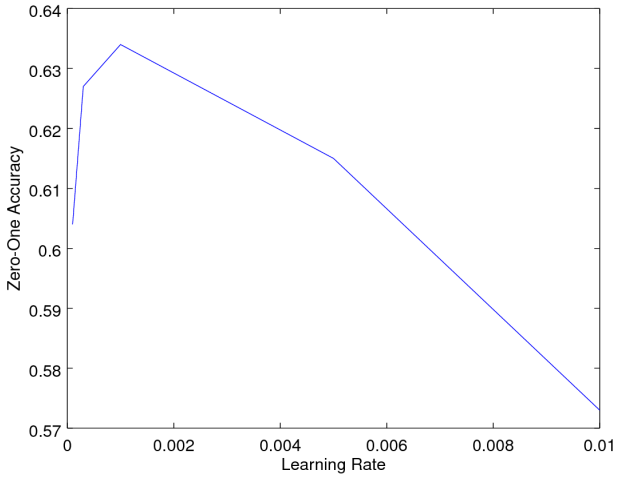

Fig. 5: Accuracy Over Different Learning Rates

G. Batch Size

In this subsection, we kept parameters other than $bsize$ constant such that $wdecay = 0.0$ and $lrate = 0.001$ - $lrate$ was chosen recently- and using *three hidden layer* with $nneurons$ $256,\ 128,\ 64$, and we experimented validation set accuracies over different $bsize$s. Validating in Fig. 6 according to Zero-One accuracy, $bsize = 32$ is the optimal $bsize$. From now on, we will use $bsize = 32$ in our **final architecture**.

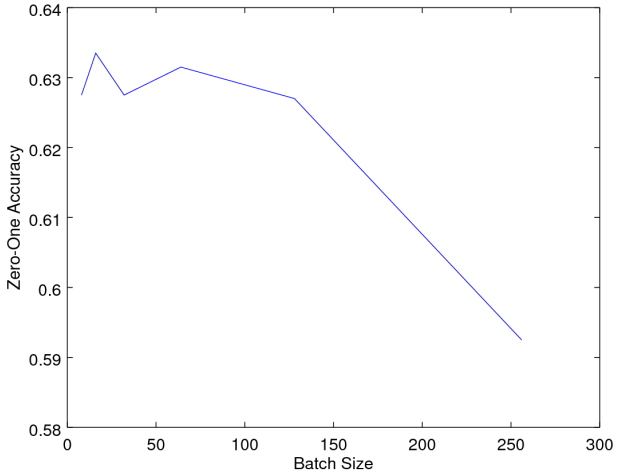

Fig. 6: Accuracy Over Different Batch Sizes

H. Weight Decay

In this subsection, after defining parameters other than $wdecay$ in previous subsections such that $bsize = 32$, $lrate = 0.001$ and selecting *three hidden layer* with $nneurons$ $256,\ 128,\ 64$, and we experimented validation set accuracies over different $wdecay$s. Validating according to Zero-One accuracy, we have 0.630 accuracy for $wdecay = 0.0$, 0.629 accuracy for $wdecay = 0.001$, 0.626 accuracy for $wdecay =$

0.01, 0.630 accuracy for $wdecay = 0.1$, 0.630 accuracy for $wdecay = 1$, and 0.630 accuracy for $wdecay = 10$. We can say that as we penalize weights more, we get slightly worse results. This is probably because our weights learn better when they become larger. Hence we have chosen $wdecay$ to be equal to 0.0. From now on, we will use $wdecay = 0.0$ in our **final architecture**.

IV. Evaluation of Some Estimations

Best estimation is done for $v1466.jpg$, with an error of $1.1978e-5$ ages -numerically 0-. Average-accurate estimation is done for $v460.jpg$, with an error of 9.864 ages. Worst estimation is done for $v43.jpg$, with an error of 77.438 ages.

Comparing best and average-accurate, the person in $v460.jpg$ has sunglasses which deceived our architecture probably because of *occlusion*. One could say that the orientation of the head in $v460.jpg$ affected the performance. However, inspecting $v32.jpg$ -also contains a strangely oriented head-, we have only 2 ages of difference between ground truth and estimation. Hence we infer that the major reason of 9.864 *ages difference* is sunglasses. Comparing worst estimation with others, $v43.jpg$ is actually a picture, not a photograph. Hence that image is not *coherent* with the majority of the images used for training. For instance, the person in that image has a smaller and stranger head comparing with the images from training set. Hence the worst estimation is done for that image.

V. Evaluation of Self Portrait

Final architecture predicted age as 26.57 for Fig. 7a and as 31.22 for Fig. 7b. The reason for this is again the sunglasses's occlusion effect as mentioned in above section. However, these predictions are both valid according to *Zero-One Accuracy* as they are far from ground truth -22- less than ± 10. Other than sunglasses's, we have no deceiving effect in these photos. To explain, the photos are high-quality, they have no noise and the background has not a deceiving color -like skin color- or patterns on it to fake *ResNet-18*. These lead us to our average-accurate results we get from validation set.

VI. Conclusion

After testing on validation set and inspecting behavior on some validation set images and 2 self portraits, we have determined to use a *three hidden layer* architecture having 256, 128 and 64 neurons in respective layers with learning rate 0.001, mini-batch size 32 and 0 weight decay based on our experiments, which gave us 0.635 accuracy in validation set.

(a) First Self Portrait

(b) Second Self Portrait

Fig. 7: Self Portraits

References

[1] A. Karphy, "CS231n Convolutional Neural Networks for Visiul Recognition", Loss Function, http://cs231n.github.io/assets/nn3/learningrates.jpeg, 2017

[2] A. Karphy, "CS231n Convolutional Neural Networks for Visiul Recognition", Train/Val Accuracy, http://cs231n.github.io/assets/nn3/accuracies.jpeg, 2017

[3] S. Khirirat, H. R. Feyzmahdavian, M. Johansson, Mini-Batch Gradient Descent: Faster Convergence Under Data Sparsity, pp. 2880–2887, December 2017