OPT 3 – COMPUTER VISION (2022 / UB)

REPORT: **Task 2**

Group members:

Johnny Núñez Cano (1), <jnunezca11@alumnes.ub.edu>, johnnynunez
Qijun Jin (2), <qjinjinj9@alumnes.ub.edu>, QJ

1. SUMMARY OF CONTRIBUTIONS

In this practice, we are trying to predict the ages of the people, given a set of metadata such as gender, ethnicity, and facial expression. We will apply the ResNet50 as our backbone model of the neural network. In addition to this model, it will be nested with a set of fully-connected layers to improve the prediction's accuracy. More concretely, we will set up some enhancements over the following points:

1- Sample weights

It is a method used in some machine learning models to assign a customized weight for each group of the training sample that can benefit the model to classify more properly and make a decision boundary.

As we are facing this task with unbalanced datasets, there are certain groups that are more popular than others, so in this case, we have to weightless those popular groups in front of others. By providing these weights, those rarely-seen groups that could be classified by age, ethnicity, facial expression, and gender will have more weights. We have looked in Keras about the definition of sample weights as:

“sample_weights functionality adds more importance to some samples than others during training that the loss function used by the model is weighted per sample not per class. It changes the way the loss is calculated.”

2- Model and hyperparameters

We have refined our model by considering several combinations of aggregated layers. In concrete, we have tried to aggregate layers of different sizes and fully-connected layers within some dropout layers. Moreover, we have also set several values for learning rate with $1e^{4}$, $5e^{5,}$ and $1e^{5}$, but the result of the model has not improved at all. Therefore, we have tried to change the type of the loss by combining with early stopping patience up to 20.

2. EXPERIMENTAL SETUP & DISCUSSION OF THE RESULTS

M1 - Baseline model - all layers trainable - without samples weights:

This is the model of the practice that is provided to us as the reference of the result to compare with the following implementation of our models, some of them with sample weights.

M2 - Baseline model - last layers trainable - with sample weights default:

This is the model that has been improved using the sample weights customized, in this case, we are using the division of the group only by age. We are only using the last layers as trainable.

M3 - Baseline model - all layers trainable - pre-entrained last layers with sample weights default:

This is the model in which all of the layers are trainable. The last layers are trained with the configuration of stage II. In this case, we are still using the division by age for obtaining each subgroups.

M4 - Model improved - last layers trainable - with sample weights personalized:

This is the model that has been trained with stage I and stage II. We are not using the pre-trained models as we want to get the result from scratch. We have also used the sample weights customized, meaning that the weighted groups are separated by age with 20 years old of difference until 60 and each group of this is separated by its ethnicities. From here, the total training set is separated into 9 different groups, and with the group of age from 60 to 100, we can get 10 subgroups of the training set.

M5 - Baseline model - all layers trainable - with sample weights personalized:

With M5 model, we have used the same configuration as before, except that the layers of the network are all trainable. The fully-connected layers aggregated after the layer of size 512 are:

Dropout	Nodes	Dropout	Nodes	Dropout	Nodes	Dropout	Nodes
0.1	256	0.1	128	0.05	64	0.05	32

Comparison of error for all the experiments:

Model	Learning rate	Training strategy	Gender bias	Expression bias	Ethnicity bias	Age bias	MAE
M1	1e-5	1	0.1370	2.35	2.8	5.45	9.59
M2	1e-5	2	0.0580	1.78	4.8	4.39	7.35
M3	1e-5	2	0.0370	1.45	2.352	3.52	**6.670045**
M4	1e-5	1	0.26	1.15	2.352	3.16	**6.940045**
M5	1e-5	2	0.12	0.78	0.322	2.85	**8.01**

Table 1: Comparison table of error and bias, results obtained for all models

As we can observe in the table, our best model M5 is the last one which is trained with all the layers as trainable and with sample weights customized. In comparison with model M3, we can say that the bias is improved using sample weights. But the mean absolute error that we got is higher than other results on Codalab. In detail, we can see that using the stage I training without sample weights and then using stage II with sample weights improved the results. In this case, even if the best result of MAE is obtained with sample weights by training both stages, we get a lot of biases. Therefore, we have trained the last layers without sample weights and then applied sample weights in stage II. As we cannot get low MAE with low bias, we conclude that our network is maybe too complex to train, so without dropouts, we are overfitting very fast the model and getting worse results.

Plots of the experiments:

For this first comparison, we have plotted the training curves using the model M4-variant and M4.

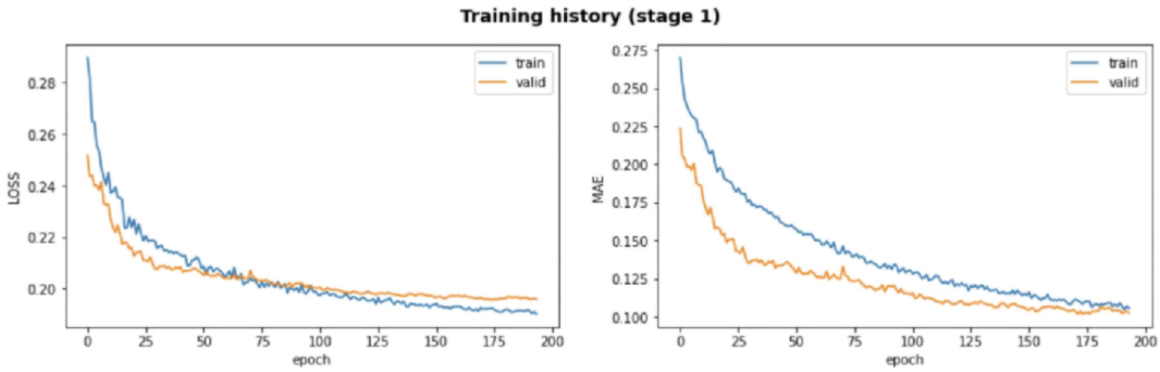

Figure 1: illustrative example of training curve on M4-variant.

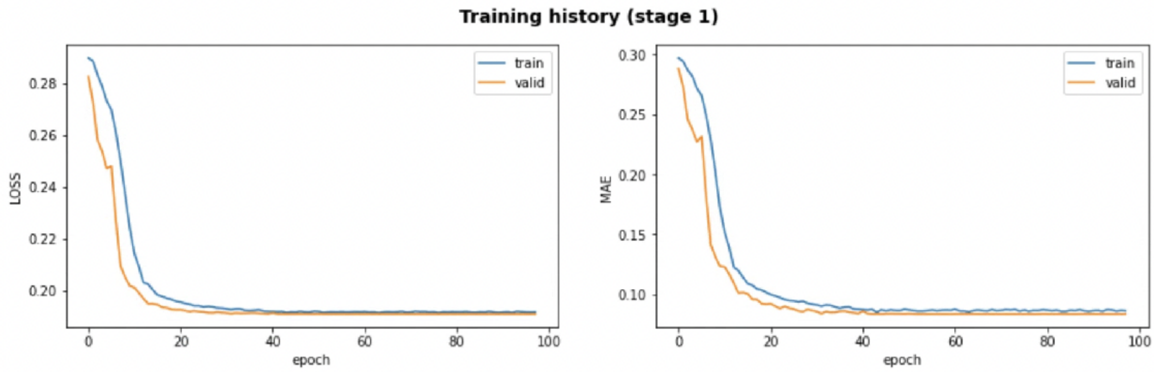

Figure 2: illustrative example of training curve on M4.

As we can see in the above plots, both models are able to be learned. With respect to the one without sample weights, the model with sample weights is getting micro jumps in the training. We think that it is probable that the model can be able to fit the network due to the fact that the model M4-variant has used 48 subgroups in comparison with the final model M4 which has only used 10 subgroups. The excess sample weights cannot fit well the curve. In figure 2, we can see that the curve is much smoother. Even in both cases, we have achieved very similar results on MAE.

This figure shows the pretrained model without sample weights:

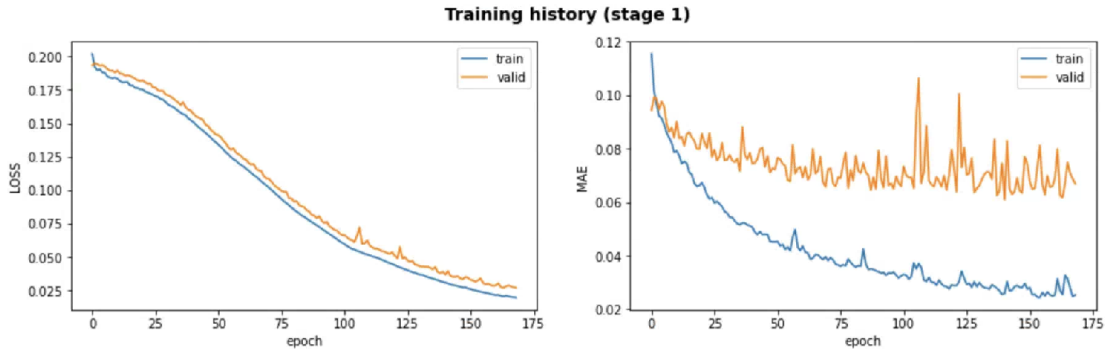

Figure 3: illustrative example of training curve on M5-variant.

As we can see in the above figure, it is confirmed that having 48 subgroups for sample weights produces noises in the training curve - MAE loss. Therefore, we used the M5 with 10 subgroups which is the one we have used to make the prediction for the Codalab competition. We have thought that adding too many subgroups can result in a strategy that is very unbalanced for certain minority groups.

3. FINAL REMARKS

To conclude, we have seen that using sample weights is a good strategy to ponderate the minor classes by giving them higher weights in front of other classes. In practice, we can see that this technique is good when they are a few strategically separable groups. When we separate the dataset into all categorically separable groups, the result of the model can be worse than the one with only a few separated groups. More concretely, when there are too many groups, the sample weights can result in adding noise to the model. In the end, by reducing the number of the groups to around 10, we obtain the best position in the Codalab.