Applied Data Science:

Weakly Supervised Learning

- Prepared by -

Sangmin Lee
Donglai Xu
Ferra Suryani
Woonsup Kim
Yayuan Wang

Columbia University

STAT GR5243 / GU4243 | Professor Ying Liu

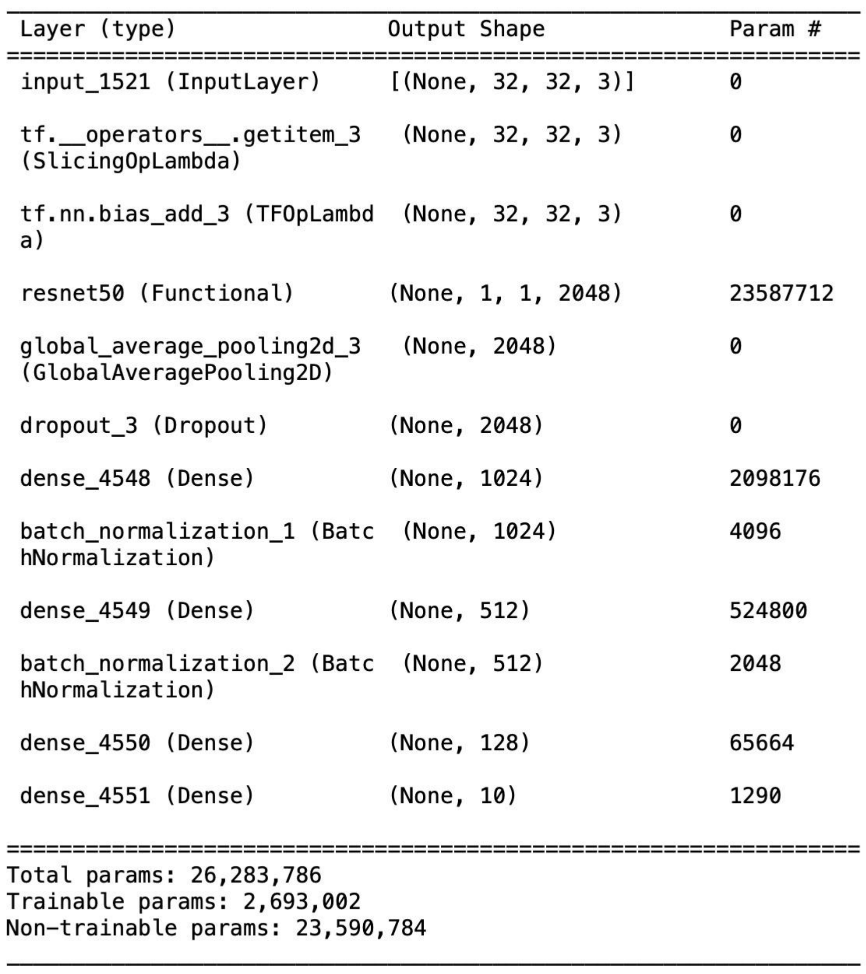
November 2, 2022

Problem Statement

A client is creating a mobile AI program that accurately classifies images. Two major concerns for the client are the portability of the program and the computational efficiency. Their current practice is to use simple logistic regression to classify the images. However, this method has two main issues; the first is that the logistic regression method is not sophisticated enough, and the second is that current practice assumes that the labels are clean and are learning from an untrustworthy source. Keeping in mind the portability and computational efficiency, our group will propose different methods that will be an improvement compared to the current practice.

Model I (ResNet50)

One of the main issues with the current practice was that the logistic regression method was not sophisticated to classify the images well. To improve on this, we utilized transfer learning to predict the labels of the images. By comparing the result from different transfer models, the ResNet50 model was selected as it gave us the best performance. Following the ResNet50 model, we added several dense layers with the activation function “ReLu”, dropout layers, and batch normalization layers to prevent overfitting. Figure 1 displays the architecture of Model I.



```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1521 (InputLayer)     [(None, 32, 32, 3)]       0

 tf.__operators__.getitem_3   (None, 32, 32, 3)        0
 (SlicingOpLambda)

 tf.nn.bias_add_3 (TFOpLambd  (None, 32, 32, 3)        0
 a)

 resnet50 (Functional)       (None, 1, 1, 2048)        23587712

 global_average_pooling2d_3   (None, 2048)             0
 (GlobalAveragePooling2D)

 dropout_3 (Dropout)         (None, 2048)              0

 dense_4548 (Dense)          (None, 1024)              2098176

 batch_normalization_1 (Batc  (None, 1024)             4096
 hNormalization)

 dense_4549 (Dense)          (None, 512)               524800

 batch_normalization_2 (Batc  (None, 512)              2048
 hNormalization)

 dense_4550 (Dense)          (None, 128)               65664

 dense_4551 (Dense)          (None, 10)                1290

=================================================================
Total params: 26,283,786
Trainable params: 2,693,002
Non-trainable params: 23,590,784
_________________________________________________________________
```

Figure 1. Summary of Model I

Evaluation of Model I

The training process for Model I took around **30 minutes,** and it returned a training accuracy of approximately 0.29 and a validation accuracy of roughly 0.24. A summary of Model I's performance is produced in Figure 2.

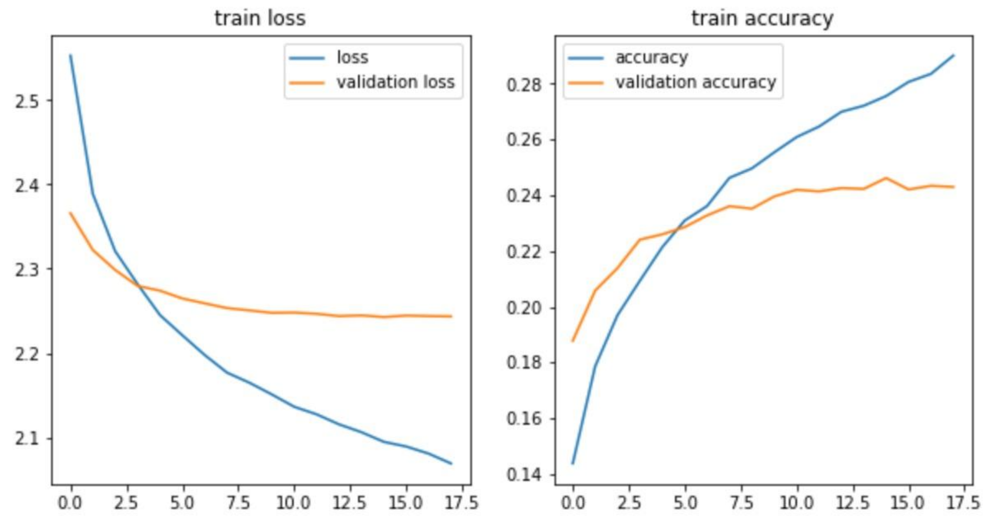


Figure 2. Summary of Model I performance

Model II (LeNet)

While the more sophisticated model of ResNet50 improved the classification, it still assumes that the labels are clean and are learning from an untrustworthy source. To improve this, we added the weakly supervised learning feature LeNet to clean the noisy data before feeding the data to ResNet50. LeNet attempts to clean the noisy data by learning and training only from the clean labels and applying this to correct the noisy labels. After this correction, the new/cleaned dataset is fed into Model I.

Evaluation of Model II

The training process for Model I took around **40 minutes,** and it returned a training accuracy of approximately 0.52 and a validation accuracy of roughly 0.47. A summary of the performance of Model I can be seen in Figure 3.

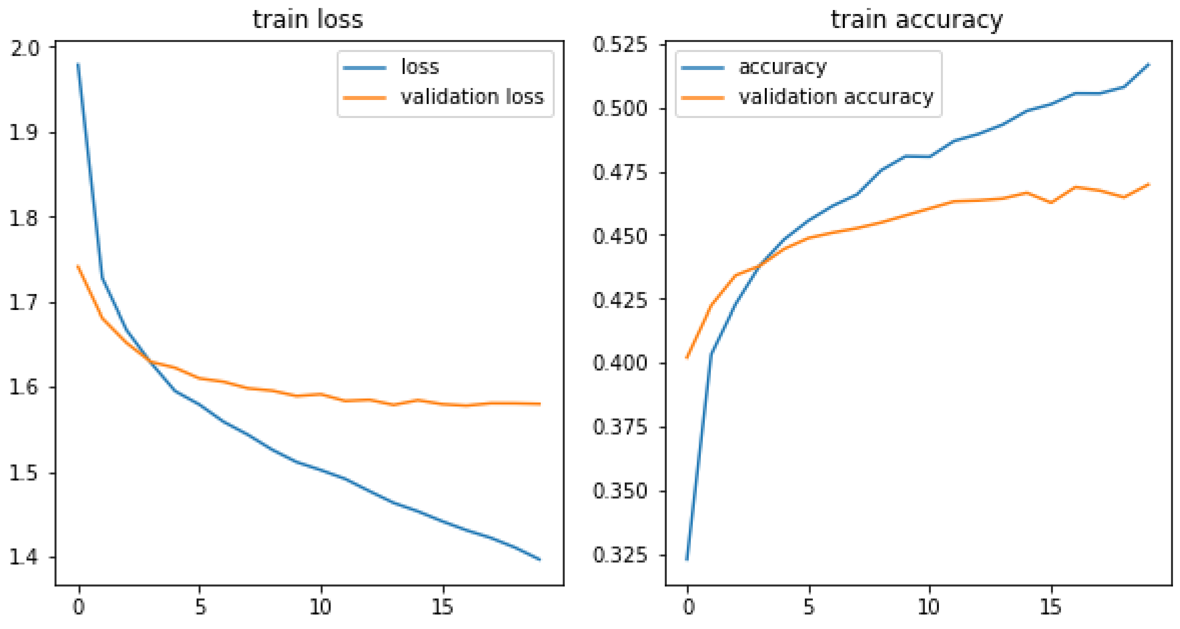
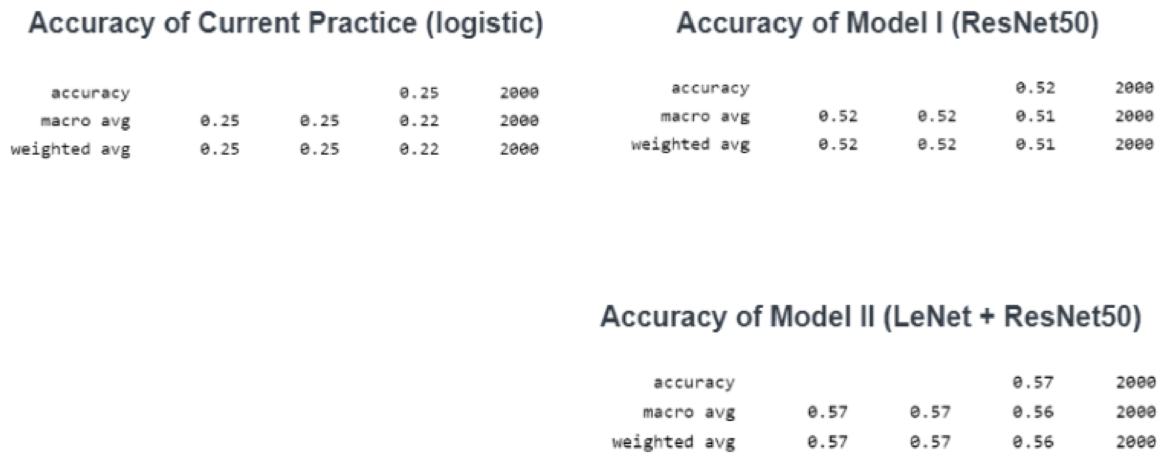


Figure 3. Summary of Model II Performance

Conclusion and Recommendation

A comparison of accuracy was made between the three models using clean data, and figure 4 summarizes this comparison.



Accuracy of Current Practice (logistic)

```
    accuracy                           0.25      2000
   macro avg       0.25      0.25      0.22      2000
weighted avg       0.25      0.25      0.22      2000
```

Accuracy of Model I (ResNet50)

```
    accuracy                           0.52      2000
   macro avg       0.52      0.52      0.51      2000
weighted avg       0.52      0.52      0.51      2000
```

Accuracy of Model II (LeNet + ResNet50)

```
    accuracy                           0.57      2000
   macro avg       0.57      0.57      0.56      2000
weighted avg       0.57      0.57      0.56      2000
```

Figure 4. Comparison of Accuracy

Current logistic regression method had a very poor accuracy of 0.25. Therefore, a more sophisticated model like ResNet50 is recommended due to the improved accuracy of 0.52. While adding a LeNet layer to clean the noisy data improved the accuracy further to 0.57, the processing time took too long and exceeded the desirable 30 minutes. Therefore, due to the computational inefficiency, we recommend using Model I (ResNet50) to develop the app.