Improving Model Accuracy for Imbalanced Image Classification Tasks by Adding a Final Batch Normalization Layer: An Empirical Study

V. Kocaman, O. M. Shir, T. Baeck

Research Question
❖ Learning from small samples in highly imbalanced image classification problems
- In real world, especially in agriculture and healthcare, the anomalies are rare and it is usually expensive, time consuming or impossible to collect them.
- In this kind of highly imbalanced classification problems, DL frameworks favor majority classes over minority classes (generalisation power of DL).
- Under covariate shift (dataset shift), train and test set come from different distributions and model fails to predict samples which haven’t seen during training.
- What is the most effective approach to enable learning of minority classes?

Possible solutions for learning from small samples in DL:
○ Get more data
○ Transfer learning (fine tuning)
○ Data augmentation
○ Cost Loss
○ Going deeper and ensembling
○ Autoencoders
○ Prior knowledge (Domain adaptation)
○ Class-balanced loss (CBL)
○ Batch Normalization (BN)?

Batch Norm (BN) is a widely adopted technique that is designed to combat internal covariate shift and to enable faster and more stable training of DNNs. It is an operation added to the model before activation which normalizes the inputs and then applies learnable scale (γ) and shift (β) parameters to preserve model performance.

TABLE I: Averaged F1 test set performance values over 10 runs, alongside BN’s total improvement, using 10 epochs with VGG19, with/without BN and with Weighted Loss (WL) without BN.

<table>
<thead>
<tr>
<th>Plant</th>
<th>Class</th>
<th>Without final BN</th>
<th>With WL (no BN)</th>
<th>With final BN (no WL)</th>
<th>BN total improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apple</td>
<td>Unhealthy</td>
<td>0.2942</td>
<td>0.7947</td>
<td>0.9562</td>
<td>0.1615</td>
</tr>
<tr>
<td></td>
<td>Healthy</td>
<td>0.7075</td>
<td>0.8596</td>
<td>0.9577</td>
<td>0.0981</td>
</tr>
<tr>
<td>Pepper</td>
<td>Unhealthy</td>
<td>0.7237</td>
<td>0.8939</td>
<td>0.9575</td>
<td>0.0636</td>
</tr>
<tr>
<td></td>
<td>Healthy</td>
<td>0.8229</td>
<td>0.9121</td>
<td>0.9558</td>
<td>0.0437</td>
</tr>
<tr>
<td>Tomato</td>
<td>Unhealthy</td>
<td>0.5688</td>
<td>0.8671</td>
<td>0.9786</td>
<td>0.1115</td>
</tr>
<tr>
<td></td>
<td>Healthy</td>
<td>0.7708</td>
<td>0.9121</td>
<td>0.9780</td>
<td>0.0659</td>
</tr>
</tbody>
</table>

**Conclusion**
- Placing an additional BN layer just before the output layer has a considerable impact in terms of minimizing the training time and test error for minority classes in highly imbalanced datasets.
- Upon adding the final BN layer the F1 test score is increased from 0.2942 to 0.9562 for the unhealthy Apple minority class, from 0.7237 to 0.9575 for the unhealthy Pepper and from 0.5688 to 0.9780 for the unhealthy Tomato when WL is not used. This suggests that the accuracy improvement is significant.
- The highest gain in test F1 score for both classes (majority vs. minority) is achieved just by adding a final BN layer, resulting in a more than three-fold performance boost on some configurations.
- Trying to maximize validation and train losses may not be an optimal way of getting a high F1 test score for minority classes.
- Having a higher train and validation loss but high validation accuracy would lead to higher F1 test scores for minority classes in less time.
- The final BN layer in imbalanced classification problems has a salutatory effect on both the probability associated with the predicted class label and reduction in prediction variance. That is, the model may perform better even if it is not confident enough while making a prediction.
- Lower values in the softmax output may not necessarily indicate lower confidence level, leading to another discussion why softmax output may not serve as a good uncertainty measure for DNNs. A model can be uncertain in its predictions even when having a high softmax output.