Solution for AliProducts Challenge: Large-scale Product Recognition

Yunbo Peng*
AI Lab of Netease Game
gzpengyunbo@corp.netease.com

Yixin Chen*
AI Lab of Netease Game
yx.chen.cs@foxmail.com

Yue Lin
AI Lab of Netease Game
gzlinyue@corp.netease.com

1. Baseline

ResNet34[1] was adopted as our baseline model with rotation and random sized crop data augmentation and SGD optimizer with momentum. We found that the over-fitting issue was severe from observing training and validation loss curve. So we used cosine annealing warm restart learning rate[4] to jump out of the local minimal point and speed up the training processes by adjusting learning rate cyclically. There was another benefit that it made us possible to integrate multiple model snapshots produced in the training process. The length of the first cycle was 1 epoch. The number of epochs increased two times after a restart. We achieved 22.00% error rate in only 7 epochs on validation set with our baseline model. The initial learning rate was selected by the loss vs learning rate figure, which plotted by increasing the learning rate from 1e-6 to 1e1 within 2 epochs.

2. Over-fitting

The over-fitting issue is partially because of many categories of AliProducts dataset only have a few images. To overcome the issue further, we conducted experiments with a lot of data augmentation techniques, such as cutout, perspective transformation, etc., and the error rate was 20.24%. We set weight decay at 3e-4 to SGD optimizer, the performance got much better (17.62%). Labeling smoothing was also tried, we found that the bigger smoothing rate, e.g, 0.4, the better performance could achieve. The convergence was also slower accordingly. We haven't adopted label smoothing in our succeed experiments due to the overall performance was suppressed after combining a lot of methods with it and the convergence speed could not be tolerated. Moreover, we found dropout was quite effective to relief the over-fitting problem and obtain robust feature representation for fine-grained classification. The dropout layer was added after the global averaging pooling for all our models, and the dropout rate was set to 0.5. After combining what worked good for us, the performance was improved to 16.34%.

3. Noise Data

The AliProducts dataset contains a lot of noise labels. The noise data is a especially severe problem. Symmetric cross entropy(SCE)[6] proved that cross entropy exhibited over-fitting to noisy labels on easy classes and suffered from significant under learning on hard classes. They proposed a loss with a noise robust counterpart named reverse cross entropy(RCE) inspired by the symmetric KL-divergence. It was adopted to suppress the missing labeled image here. We achieved 14.24% after being applied to our experiments.

O2U-Net[3] is a method to clean up noisy dataset. They argued that the higher the normalized average loss of a sample in one training cycle, the higher the probability of being noisy labels. The method was naturally compatible with our cosine annealing warm restart learning rate. However, different manual training phases were required and it was quite not effective when training several models to ensemble. Thus we proposed to train several cycles to stable the training process as usual and guarantee that the normalized average loss was accurate firstly. Afterwards, the normalized average loss was treated as the samples weight. The next cycle used the samples weight generated by the previous cycle. The performance could be further improved to 13.84%.

4. Imbalanced Classes

Data imbalance is another problem we encountered in this dataset. To construct a balanced sub-dataset as well as ensure that the samples in the sub-dataset are clean and highly reliable, we used the model trained on the original training dataset to clean all the data. More concretely, we divided the data into two parts equally, and then trained models on each part of the data, and used the trained model to predict the other part of the data. In order to get clean samples, we only kept those samples whose prediction results are consistent with the original labels. In addition, we sampled the retained data to construct a balanced sub-dataset, and gave priority to samples with higher prediction scores in each category. In the competition, we actually used the SENet model to filter the data, and the number of

* Equal contribution.

samples retained in each category was set to 3. This method could improve the performance by 2%.

5. Backbone

After above methods were validated with our baseline model. We changed the backbone and trained several models for ensemble, including ResNeSt[7], RegNet[5], SENet154[2] and SEResNeXt[2]. Note that all backbone models we used were pretrained on ImageNet and all our models were trained on training set only. The best one was ResNeSt, which could boost the result to 10.98% on validation set without handing imbalanced classes. The result could be improved to 8.35% after finetuned by constructed balanced sub-dataset above with the single model. Test time augmentation(TTA) were also adopted. We used customized multi scale crops in our experiments.

6. Model Ensemble

In order to fuse the trained models of different backbones to decrease the error rate of classification, we used a class-based weighting method for model ensemble. We observed that each model had different classification capabilities in different categories. We hope that a model can be classified in the categories that it is good at. Therefore, we took the classification accuracy of different models in a specific category as referenced weights, and obtained weights of different models in a certain category through nonlinear scaling and normalization on referenced weights. Finally, each model will get a weight vector for different categories. In the test, weighted summation of the model prediction results by using weight vectors, we finally obtained the ensemble result of each sample. In the competition, we finally merged the results of all our models, and achieved 5.83% on validation set and 9.16% on the final test set.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[2] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[3] J. Huang, L. Qu, R. Jia, and B. Zhao. O2u-net: A simple noisy label detection approach for deep neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3326–3334, 2019.

[4] I. Loshchilov and F. Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[5] I. Radosavovic, R. P. Kosaraju, R. Girshick, K. He, and P. Dollár. Designing network design spaces. *arXiv preprint arXiv:2003.13678*, 2020.

[6] Y. Wang, X. Ma, Z. Chen, Y. Luo, J. Yi, and J. Bailey. Symmetric cross entropy for robust learning with noisy labels. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 322–330, 2019.

[7] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, Z. Zhang, H. Lin, Y. Sun, T. He, J. Mueller, R. Manmatha, et al. Resnest: Split-attention networks. *arXiv preprint arXiv:2004.08955*, 2020.