Icing on the Cake: An Easy and Quick Post-Learnig Method You Can Try After Deep Learning

Tomohiko Konno* **and** **Michiaki Iwazume**

AI Science Research and Development Promotion Center
National Institute of Information and Communications Technology, Tokyo Japan

Abstract

We found an easy and quick post-learning method named "Icing on the Cake" to enhance a classification performance in deep learning. The method is that we train only the final classifier again after an ordinary training is done.
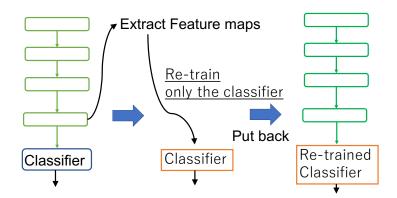
Figure 1: The sketch of the proposed method.("Icing on the Cake"). Left: Train a deep neural network as usual. Center: Extract features of input data as estimation from the layer before the final classifier, and then train the final classifier again by the extracted features. Right: Put the re-trained classifier back to the network. The accuracies in cifar100 in ResNet56 is improved from 66% to 73% in our experiment

1   Introduction and Related Works

We have seen a great success in deep learning [1]. Now, deep leanings can do a lot of things including language translation, image classification, image generation [2, 3], image generation from text [4–7], text generation from image [8, 9], unsupervised machine translation [10–12], playing Go (reinforcement learning) [13], and even network design of deep learning by themselves [14], to raise a few. The achievements are so many, that we cannot exhaustively enough at all.

Six years have passed since the advent of surprise to deep learning [15]. However, there is no such symptom that the progress in deep learning research would slow-down at all. To the contrary, we see that the research has been accelerated, and the realm that deep learning achieves has been spreading even at this moment. We, human, cannot catch up the progress. We do not still understand deep learning yet. In the present paper, we will show another phenomena, which we found by chance.

---

*Corresponding Author: tomohiko@nict.go.jp

Our proposed method named "Icing on the Cake" enhances the accuracies of image classification problems in deep learnings by an easy and quick way. We do not fully understand the reason yet, and the phenomena is interesting to the community. It is another learning method as: early stopping [16] that stops epochs early timing to prevent overfitting; dropout [17] that deactivates some nodes when training; batch normalization [18]; group normalization [19]; He initialization [20]; Xavier initialization [21]; mix-up (between-class learning; Sample Parings) [22–24] that mixes datas and labels to enhance learning; and so forth. We add another one to the existing list.

2 Proposed Method: Icing on the Cake

The method named "Icing on the Cake" consists of what follows. The sketch of the idea is illustrated in Fig. 1.

Step-1: Train a deep neural network as usual.

Step-2: Extract features from the layer just before the final classifier as estimation.

Step-3: Train only the final layer again by the extracted features.

Step-4: Put the final classifier back to the original network.

After the procedure, accuracies increase.

2.1 An Easy Way to Implement

In the implementation, we do not have to put the re-trained classifier back to the network. We have an easier equivalent method. First, we input test data, and then extract the features of the test data, and then input the features into the re-trained classifier. This is illustrated in Fig. 2.
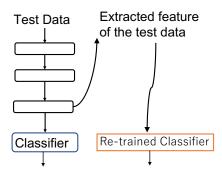

Figure 2: An easy way to implement when estimate

We have two good things in "Icing on the Cake".

1. It is easy to try.
2. It does not take time for re-training to finish.

It is not very rare that it takes a week or so to train deep neural networks. However, since "Icing on the Cake" is just to train the final layer only, it does not take time. It is done in about five minutes or so. It is really icing on the cake. We do not strongly insist that "Icing on the Cake" works all the cases. We cannot prove it for all the networks to hold as of now. The method is worth trying, and it does not hurt.

2.2 Experiments

ResNet [25] with varying depth from 11 to 56 for the100-image classification problem in Cifar100 [26] are investigated. Batch size is 128, optimizer is Adam, epochs are 200, and data augmentation is used. We use Keras [27]. The result is summarized in Tab. 1, and illustrated in Fig. 3. The averages
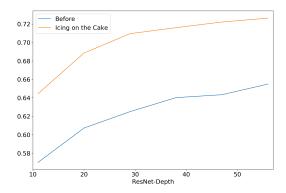
Figure 3: Accuracies in ResNet with varying depth from 11 to 56 for 100-image classification in cifar100 before and after "Icing on the Cake".

are taken over ten times. After "Icing on the Cake" the accuracies always get bigger than before, and the standard deviations get smaller.

In other experiments, ResNet56 and DenseNet[2] [28] are investigated. The 100-image and 10-image classifications are done in Cifar100 and in Cifar10. The results are summarized in Tab. 2 and Tab. 3. Batch size is 128, optimizer is Adam, epochs are 200, and data augmentation is used. "Icing on the Cake" is abbreviated as ICK in the tables. The results show that after "Icing on the Cake" the accuracies get clearly bigger than before. The results are the averages over ten data points for each. After "Icing on the Cake" the accuracies get bigger, and the standard deviations get smaller.

ResNet-Depth	Before	Icing on the Cake
11	0.57 (0.010)	0.64 (0.003)
20	0.61 (0.014)	0.69 (0.005)
29	0.63 (0.021)	0.71 (0.005)
38	0.64 (0.016)	0.72 (0.004)
47	0.64 (0.020)	0.72 (0.003)
56	0.66 (0.015)	0.73 (0.003)

Table 1: The experiments in ResNet with varying depth from 11 to 56. Accuracies and stand deviations are written.

	ResNet56	DenseNet
Before	0.66 (0.015)	0.59 (0.023)
ICK	0.73 (0.003)	0.79 (0.0049)

Table 2: The classifications in Cifar100

	ResNet56	DenseNet
Before	0.885 (0.012)	0.86 (0.024)
ICK	0.92 (0.0025)	0.91 (0.0057)

Table 3: The classifications in Cifar10

2.3 Comments

We do not fully understand the reason why "Icing on the Cake" works. We did experiments in an effort to find a counter example. However, we could not. Rather, we collected more positive evidences at last. The reason why "Icing on the Cake" works is not yet fully investigated, and is open to the community.

3 Conclusion

We found that classification performance in deep learnings get improved with an easy and quick method named "Icing on the Cake". The method is just to train the final classifier only again by the extracted features.

---

[2]Depth=40, DenseBlock=3, and $k = 12$

3

It is not that we strongly insist that every deep learning is improved by "Icing on the Cake", but that there is a possibility that it works well. It is very easy to try, and it does not take time. It is just to train the last one layer. It is worth trying. It is really icing on the cake.

A   Other experiments in WideResNet

WideResNet–16–8 [29] is investigated for 100-image classification problem in cifar100, and summarized in Tab. 4. The batch size is 128, optimizer is Adam, and the epochs are 200. The differences in Wide ResNets is not as big as in Resnet and DenseNet. However, after "Icing on the Cake", the accuracies get always bigger than before as in the table.

Before	Icing on the Cake
0.72	0.73
0.72	0.74
0.72	0.74
0.72	0.73
0.72	0.73
0.72	0.74
0.73	0.74
0.73	0.74
0.72	0.73

Table 4: The experiments in WideResNet–16–8

References

[1] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

[2] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[3] J. Johnson, A. Gupta, and L. Fei-Fei. Image Generation from Scene Graphs. *ArXiv e-prints*, April 2018.

[4] S. Hong, D. Yang, J. Choi, and H. Lee. Inferring Semantic Layout for Hierarchical Text-to-Image Synthesis. *ArXiv e-prints*, January 2018.

[5] S. Sharma, D. Suhubdy, V. Michalski, S. Ebrahimi Kahou, and Y. Bengio. ChatPainter: Improving Text to Image Generation using Dialogue. *ArXiv e-prints*, February 2018.

[6] Z. Zhang, Y. Xie, and L. Yang. Photographic Text-to-Image Synthesis with a Hierarchically-nested Adversarial Network. *ArXiv e-prints*, February 2018.

[7] K. Chen, C. B. Choy, M. Savva, A. X. Chang, T. Funkhouser, and S. Savarese. Text2Shape: Generating Shapes from Natural Language by Learning Joint Embeddings. *ArXiv e-prints*, March 2018.

[8] D. Li, Q. Huang, X. He, L. Zhang, and M.-T. Sun. Generating Diverse and Accurate Visual Captions by Comparative Adversarial Learning. *ArXiv e-prints*, April 2018.

[9] R. Luo, B. Price, S. Cohen, and G. Shakhnarovich. Discriminability objective for training descriptive captions. *ArXiv e-prints*, March 2018.

[10] G. Lample, A. Conneau, L. Denoyer, and M. Ranzato. Unsupervised Machine Translation Using Monolingual Corpora Only. *ArXiv e-prints*, October 2017.

[11] M. Artetxe, G. Labaka, E. Agirre, and K. Cho. Unsupervised Neural Machine Translation. *ArXiv e-prints*, October 2017.

[12] G. Lample, M. Ott, A. Conneau, L. Denoyer, and M. Ranzato. Phrase-Based Neural Unsupervised Machine Translation. *ArXiv e-prints*, April 2018.

[13] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[14] H. Pham, M. Y. Guan, B. Zoph, Q. V. Le, and J. Dean. Efficient Neural Architecture Search via Parameter Sharing. *ArXiv e-prints*, February 2018.

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[16] Lutz Prechelt. Early stopping-but when? In *Neural Networks: Tricks of the trade*, pages 55–69. Springer, 1998.

[17] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[18] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[19] Y. Wu and K. He. Group Normalization. *ArXiv e-prints*, March 2018.

[20] K. He, X. Zhang, S. Ren, and J. Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *ArXiv e-prints*, February 2015.

[21] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[22] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond Empirical Risk Minimization. *ArXiv e-prints*, October 2017.

[23] Y. Tokozume, Y. Ushiku, and T. Harada. Between-class Learning for Image Classification. *ArXiv e-prints*, November 2017.

[24] H. Inoue. Data Augmentation by Pairing Samples for Images Classification. *ArXiv e-prints*, January 2018.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[26] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[27] Francois Chollet et al. Keras. `https://github.com/fchollet/keras`, 2015.

[28] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016.

[29] S. Zagoruyko and N. Komodakis. Wide Residual Networks. *ArXiv e-prints*, May 2016.