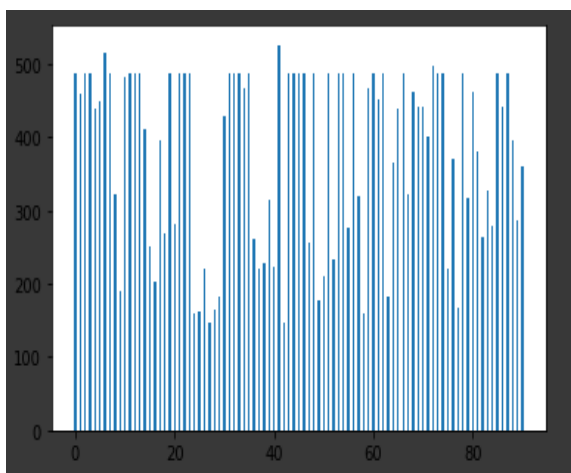


Final Term Project Report

2016314726 정영준

A given Dataset consists of 35,384 images classified into 91 classes. [Figure 1] shows the number of images classified for each class as a bar graph. Unbalanced Dataset with different image counts for each class, each class contains up to 526, at least 146 images, and an average of 379. From the entire Dataset, 20 images were randomly selected from each class, and half of them were divided into validation and test sets, so about 5% of the total data were set as validation sets and test set. Normalization (set by mean=0.49 and std=0.25) was applied to all images. Set the ResNet-19 as the initial model (120 Epoch, 0.1 Learning rate, Momentum SGD optimizer (momentum coefficient=0.9), 0.1 factor learning rate decay at 60,90 Epoch, 128 batch size) and the learning result is 97% accurate in the training set. Precision, Recall, and f1-score were printed for each class based on the validation set. F1-score appeared low in certain classes.



[Figure 1] Image Distribution by Class

24th, 37th, 52th, and 69th classes recorded f1-score less than 0.5 and representing images of chairs, kangaroos, otters and sea lions, respectively. Classes with significantly different distributions or easy confusion with different classes of images are mainly located in the lower ranks, and f1-score correlates with the number of images in each class with 0.16, with the two indicators having a slight positive correlation. As a result of using RandomCrop, ColorJitter, RandomRotation, and RandomGrayScale to reduce Overfitting by applying Data augmentation, accuracy of training set was 85% and 64% in the Validation set. The maximum accuracy of the validation set obtained by increasing the Weight Decay sequentially from 0.0005 was 66%. I formulated a hypothesis that the accuracy of the validation set is no longer improved because the image is 32x32 size small size, so significantly different image from the original image is trained when applying a specific Data Augmentation and dataset is imbalanced and consists mislabeled data. I tried to solve three problems using Label smoothing, Oversampling, and CutMix method. One-hot encoded correct answer labels are modified according to the smoothing factor, resulting in a decrease in value from 1 in the correct class and a rise in value from 0 in the remaining classes. And use modified value to calculate loss [Table 1].





Next, I tried to solve the imbalanced Dataset problem through Oversampling method. I adjusted the WeightRandomSampler of Pytorch to sample the image. Sampling were

Table 2: Top-1 classification accuracies of networks trained with and without label smoothing used in visualizations.

DATA SET	ARCHITECTURE	ACCURACY ($\alpha = 0.0$)	ACCURACY ($\alpha = 0.1$)
CIFAR-10	ALEXNET	86.8 ± 0.2	86.7 ± 0.3
CIFAR-100	ResNet-56	72.1 ± 0.3	72.7 ± 0.3
ImageNet	INCEPTION-V4	80.9	80.9

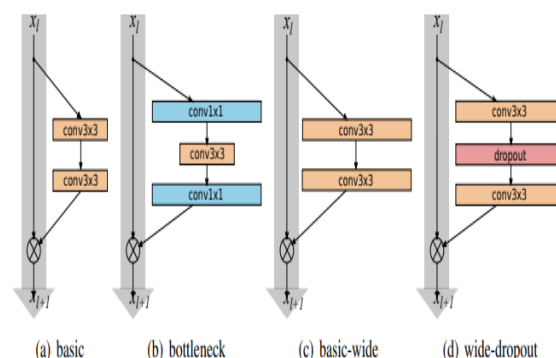
[Table 1] Label smoothing performance

made according to the number of images belonging to the class, so that, on average, images of all classes could equally affect loss and optimizer. Finally, Data Augmentation through CutMix method was applied. CutMix's mechanism is that randomly mix images belonging to a minibatch and cut a certain part of them and pastes them to the original image to create an image with two classes and predict them to yield loss [Table 2]. The class of the composite image is determined by the cutting ratio, so it can have the effect of using Cutout and Mixup at the same time. The probability of using Cutmix was 0.5 and the cutting ratio was sampled in Beta distribution (1,1). Using Cuitmix technique instead of previous used RandomCrop, ColorJitter, etc. And set the smoothing factor of Label smoothing to 0.1

	ResNet-50	Mixup [48]	Cutout [3]	CutMix
Image				
Label	Dog 1.0	Dog 0.5 Cat 0.5	Dog 1.0	Dog 0.6 Cat 0.4
ImageNet Cls (%)	76.3 (+0.0)	77.4 (+1.1)	77.1 (+0.8)	78.6 (+2.3)
ImageNet Loc (%)	46.3 (+0.0)	45.8 (-0.5)	46.7 (+0.4)	47.3 (+1.0)
Pascal VOC Det (mAP)	75.6 (+0.0)	73.9 (-1.7)	75.1 (-0.5)	76.7 (+1.1)

[Table 2] CutMix

and weighted sample image based on the number of images in each class of Training set using WeightedRandomSampler, and then start learn with the same model. The result was better than before: 80% Training Set accuracy and 74% on Validation set. The application of L2 Regularization did not affect the Validation set and resulted in lower accuracy of the Training set only. The models such as ResNet-50, ResNext, DenseNet, Wide-ResNet(WRN), SE-Net+ResNet, SE-Net+WRN, etc. were used to perform the learning under the same conditions. As a result of learning, the SENet+Wide-ResNet-28-10 model with pre-activation has reached the highest accurate. As the Convolution neural network deepens, the features of the initial layers have learned become more dilute and the model wouldn't learn well.



[Figure 2] WRN & ResNet Layer Block

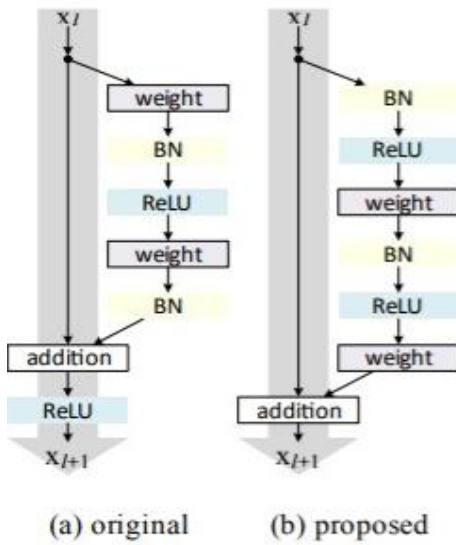
ResNet is a model structure that solves the problem by adding information from the initial input to the output. Wide-ResNet is a model with a structure that increases the channels of the Convolution layers from the normal ResNet's Residual block. I could get the best results when I used the © Basic-wide Block [Figure 2]. Wide-ResNet (WRN) is a model with a reduced depth and increased width structure in ResNet, with clear performance improvements compared to ResNet [Table 3]. The model with SE-Layer

	depth-k	# params	CIFAR-10	CIFAR-100
NIN [20]			8.81	35.67
DSN [14]			8.22	34.57
FitNet [24]			8.39	35.04
Highway [28]			7.72	32.39
ELU [6]			6.55	24.28
original-ResNet[1]	110	1.7M	6.43	25.16
	1202	10.2M	7.93	27.82
stoc-depth[14]	110	1.7M	5.23	24.58
	1202	10.2M	4.91	-
pre-act-ResNet[13]	110	1.7M	6.37	-
	164	1.7M	5.46	24.33
	1001	10.2M	4.92(4.64)	22.71
WRN (ours)	40-4	8.9M	4.53	21.18
	16-8	11.0M	4.27	20.43
	28-10	36.5M	4.00	19.25

[Table 3] WRN & ResNet Test Error rate

and pre-activation applied to the WRN-28-10 with the best performance was finally used for learning.

Pre-activation is applying the Batch Normalization, Relu activation function before the convolutional layers when passing through the Residual Block in the ResNet or the model derived from ResNet [Figure 3]. This structure is designed to keep shortcut information as much as possible in ResNet and to allow as little distortion of information in back propagation as possible, showing



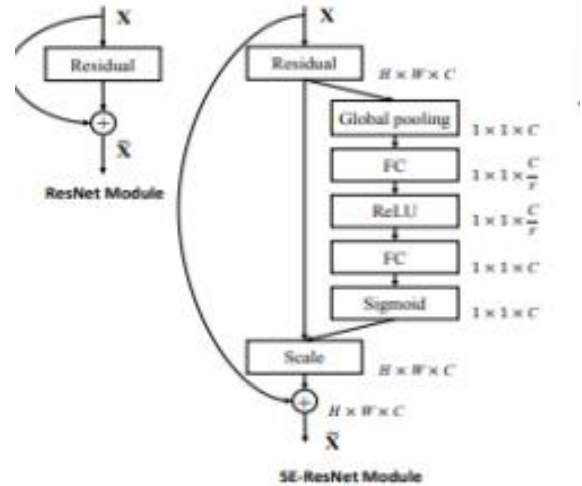
[Figure 3] Pre-activation

better performance than the normal structure [Table 4].

case	Fig.	ResNet-110	ResNet-164
original Residual Unit [1]	Fig. 4(a)	6.61	5.93
BN after addition	Fig. 4(b)	8.17	6.50
ReLU before addition	Fig. 4(c)	7.84	6.14
ReLU-only pre-activation	Fig. 4(d)	6.71	5.91
full pre-activation	Fig. 4(e)	6.37	5.46

[Table 4] Pre-activation performance

Squeeze-and-Excitation Networks(SE-Net) is a technique that compresses and re-extracts the information in Feature map to highlight important features and adds them to the Original Network, which can be applied to ResNet as shown in [Figure 4].



[Figure 4] SE-Net structure

I used all of these methods to create a pre-activation SE-Net+WRN-28-10 (depth=28, k=10) model. Adding RandomCrop for image augmentation brought about a better performance. This model recorded 85% accuracy in the training set and 79% accuracy in the valid set. The verification using the test set achieved 80% accuracy.

Reference

- [1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep Residual Learning for Image Recognition. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778
- [2] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, Youngjoon Yoo. CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features. arXiv:1905.04-899[cs.CV]
- [3] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, Kaiming He. Aggregated Residual Transformations for Deep Neural Networks. arXiv:161105431-[cs.CV]
- [4] Sergey Zagoruyko, Nikos Komodakis. Wide Residual Networks. arXiv:1605.07146 [cs.CV]
- [5] Rafael Müller, Simon Kornblith, Geoffrey Hinton. When Does Label Smoothing Help? arXiv:1906.-02629 [cs.LG]
- [6] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, Enhua Wu. Squeeze-and-Excitation Networks. arXiv:1709.01507-[cs.CV]