COSE474-2023F: Final Project Report
Multi-Residual Networks with ReLU Dropping

Chanbyeong Park

Abstract

본 연구에서는 residual networks가 ensemble과 같이 행동한다는 관점을 근거로 ensemble의 개수인 multiplicity를 최대로 만들어 성능 향상을 이끌기 위한 ReLU dropped multi-residual networks을 제안한다. 이는 ReLU dropped residual function으로 구성되어 기존 multi-residual networks에 비해 실질적인 ensemble 개수의 증가를 얻을 수 있으며, 1.7M parameter 모델로 CIFAR-10 데이터셋에 대해 5.41%의 오차율을 얻어 ResNet과 Multi-ResNet에 비해 더 높은 정확도를 보였다. 또한 residual networks의 성능은 단순히 valid path의 개수가 아니라 effective path의 개수와 상관관계를 보인다는 해석을 통해 더 효율적인 학습을 위한 방향을 제안하였다.

1. Introduction

Convolution Neural Networks(CNNs)(LeCun et al., 1989)는 컴퓨터비전의 문제를 해결하는 새로운 방법을 제시하여 큰 발전을 이끌었다. 이는 특히 이미지 분류에서도 뛰어난 성능을 보였는데, CNNs의 첫 등장 이후 AlexNet(Krizhevsky et al., 2012), VGG(Simonyan & Zisserman, 2015), GoogleNet(Szegedy et al., 2014) 등 더 깊은 모델으로 이어지며 이미지 분류의 정확도가 향상되었다. 다만 모델의 깊이가 깊어지면 gradient vanishing 또는 exploding이 발생한다는 문제가 있어 깊이를 더 늘리는 데 어려움이 있었다. Residual networks(ResNet)(He et al., 2015)는 identity mapping이라는 방식으로 이러한 문제를 방지하였고, 모델을 더욱 깊게 쌓을 수 있게 만들어 이미지 분류의 정확도를 향상시켰다.

ResNet의 높은 정확도는 identity skip-connection을 통해 gradient 값이 낮은 level의 layer에서도 보존될 수 있는 특성을 가지기 때문이라고 생각되었다. 하지만 ResNet은 residual block의 convolution layer를 지나거나 지나지 않는 다양한 path의 ensemble로 해석할 수 있으며, ResNet의 학습은 모델의 전체 깊이에 비해 상대적으로 얕은 깊이의 path에 주로 의존한다는 연구(Veit et al., 2016)가 있었다. 이에 따르면 모델의 gradient update는 주로 상대적으로 얕은 effective path에 의해 발생하며, ResNet의 성능은 가능한 모든 path인 valid path의 개수와 상관관계를 보였다.

Multi-residual networks(Multi-ResNet)(Abdi & Nahavandi, 2017)는 valid path의 개수인 multiplicity를 증가시키기 위해 기존 residual block에 residual function을 추가한 구조이다. 이는 깊이가 고정된 상황에서 multiplicity가 증가한다는 이점을 가져 기존의 모델의 비해 더 좋은 성능을 보였다. 다만 이는 실질적으로 multiplicity의 증가로 이어지지 않을 수 있다. Multi-ResNet의 residual function은 모두 같은 구조를 가지기 때문에 각 block의 residual function은 같은 구조와 같은 데이터로 학습을 수행한다. 이 경우 ensemble의 개수가 증가한다고 보기는 어려우며, residual function을 지나는 이항 분포의 확률을 증가시키는 것에 그칠 수 있다. 따라서 서로 다른 구조의 residual function을 사용해야 multiplicity가 증가한다고 할 수 있다. 또한, 모든 residual function과 identity mapping의 합인 edge의 개수가 일정한 경우에는 residual function의 추가가 오히려 multiplicity의 감소로 이어진다.

이러한 한계를 극복하기 위해 본 연구에서는 ReLU dropped multi residual networks(RDM-ResNet)을 제안한다. 이는 기본적으로 Multi-ResNet과 유사한 구조를 가지지만, CNNs에서 ReLU를 제거하는 경우 더 일반화된 특성을 얻어 기존 구조에 비해 향상된 성능을 얻을 수 있다는 연구(Zhao et al., 2018)에 따라, 추가되는 residual function을 ReLU dropped residual function으로 적용한다. 이를 통해 기존의 residual function과 다른 특성을 얻어 실질적인 multiplicity의 증가라는 이점을 얻을 수 있도록 한다. 또한, 전체 edge의 개수가 일정한 경우에는 residual function이 2개인 경우에 multiplicity가 최대가 되므로 residual function은 하나만 추가하는 구조를 가진다. 이 관계에 대한 내용은 Appendix A에 나타내었다.

본 연구의 기여는 다음과 같이 요약할 수 있다:

- 동일한 function 대신 ReLU dropped residual function을 multi-residual로 추가하였을 때, 다양한 특성의 ensemble을 얻어 성능이 향상될 수 있는지 확인한다.
- 동일한 크기의 모델에서 multi-residual block을 구성하는 function의 개수가 2개일 때 가장 큰 multiplicity를 얻어 성능이 향상될 수 있는지 확인한다.

2. Related Work

Residual Networks. Residual networks(He et al., 2015)는 identity mapping을 통한 residual learning을 적용하여 더 깊은 구조에서도 잘 학습되고 깊이 증가에 따른 성능

향상을 누릴 수 있다. 50-layer 이상의 깊은 모델은 1×1 convolution으로 차원을 늘리고 줄이는 bottleneck 구조를 사용하여 시간 복잡도가 너무 커지지 않도록 한다. 또한, residual block 내부의 batch normalization, ReLU, convolution의 순서를 적절히 배치한 pre-activation residual networks(He et al., 2016)는 성능 향상 및 더욱 깊은 모델에서의 학습을 가능하게 하였다.

Multi-Residual Networks. Multi-residual networks(Abdi & Nahavandi, 2017)는 residual networks가 얕은 networks의 ensemble로 작동한다는 관점에 근거하여 residual block의 residual function 개수를 증가시키는 모델이다. 이는 모델을 깊은 대신 넓게 만들어 성능을 향상시켰다. 또한, model parallelism을 통해 깊은 모델의 계산 복잡도를 15% 까지 향상시켰다.

ReLU Proportional Module. ReLU Proportional Module(Zhao et al., 2018)은 convolution layer와 ReLU가 N:M (N>M)의 비율을 이루는 구조로, 이는 추가적인 연산없이 더 일반화된 특성을 얻어 1:1 비율의 기존 모델에 비해 성능 향상을 보였다. 각 convolution layer 사이에는 batch normalization이 존재하기 때문에 두 convolution이 하나의 convolution으로 합쳐지지 않는다.

3. Methods

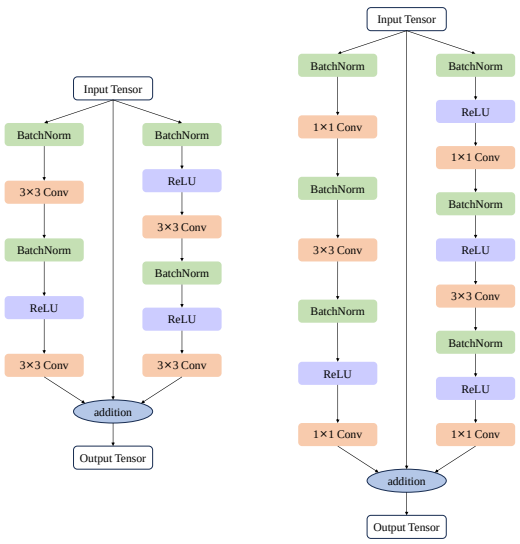

Figure 1. 왼쪽은 ReLU dropped multi-residual basic block, 오른쪽은 ReLU dropped multi-residual bottleneck block을 나타낸다.

본 연구에서 제안하는 RDM-ResNet은 기존의 residual block에 추가적인 residual function을 도입하여 multiplicity의 증가를 목표로 한다. 다만 실질적인 ensemble의 다양성을 위해 기존 block의 function과 다른, ReLU dropped residual function을 추가적인 residual function으로 적용한다. 또한, Appendix A에 따라 전체 edge의 개수가 동일한 경우 residual function의 개수가 2일 때 가장 큰 multiplicity를 얻기 때문에 모델의 크기가 일정할 때 더 효율적인 학습이 가능할 것으로 생각하여 residual function은 하나만 추가한다. Figure 1은 RDM-ResNet의 residual block을 나타낸 것이다. 이는 기존 residual networks의 basic과 bottleneck 구조 모두에 적용 가능하며, pre-activation residual block을 기반으로 하였다.

4. Experiments

4.1. Datasets

본 연구에서는 이미지 분류에 주로 사용되는 CIFAR-10 데이터셋(Krizhevsky et al., 2009)을 사용한다. 해당 데이터셋은 10개의 class를 가지며, 50,000개의 training 이미지와 10,000개의 test 이미지로 구성된다. 각 이미지는 모두 32×32 크기의 RGB 이미지다.

4.2. Experimental Design

전체적인 실험 구조는 ResNet(He et al., 2016)과 동일하게 설계하였다. 모든 입력 이미지는 각 채널에 대한 normalization을 적용하며, augmentation은 translation과 horizontal flip을 적용한다. 모델은 32×32의 이미지를 입력으로 받으며, 첫 3×3 convolutions를 적용한 후 $6n$개의 3×3 convolution layer를 쌓는다. $2n$개의 layer를 거칠 때마다 stride를 2로 설정하여 feature map 크기가 $\{32, 16, 8\}$로 변화하며, filter의 개수도 $\{16, 32, 64\}$로 구성된다. 모델의 마지막은 각 filter에 대한 global average pooling과 fully-connected layer, softmax로 구성된다. 따라서 전체 layer의 개수는 $6n + 2$가 된다. 다만 bottleneck 구조를 사용하는 모델은 $9n + 2$개의 layer를 갖는다. RDM-ResNet은 각 block에서 사용하는 residual function의 개수가 2배이므로 ResNet에서의 n 값의 절반을 사용하면 ResNet과 모델의 크기가 유사해진다.

실험의 hyperparameter 또한 거의 비슷하게 사용한다. stochastic gradient descent로 최적화를 진행하며, weight decay로 0.0001, momentum으로 0.9를 사용한다. Weight initialization과 BN도 ResNet과 동일하게 사용한다. Mini-batch size는 128이며, learning rate는 0.1로 시작하여 80, 120 epoch에서 각각 0.1씩 곱해 165 epoch까지 진행한다. 다만 learning rate warm up을 적용해 첫번째 epoch에서는 0.01의 learning rate를 사용한다.

본 실험에서는 각각 ReLU dropped multi-residual basic block과 ReLU dropped multi-residual bottleneck block을 사용하는 RDM-ResNet-56과 RDM-ResNet-83으로 실험을 수행한다. 이는 $n = 9$인 경우이며, 1.7M의 parameter를 갖는다. 이와 유사한 parameter 개수를 갖는 $n = 18$의 Pre-ResNet-110과 Pre-ResNet-164를 baseline으로 한다. 각 모델에 대해 5번씩 실험을 진행하였으며, 결과는 모든 시행의 중앙값으로 얻는다. 실험은 Google Colab으로 수행하였으며 CPU는 Intel Xeon 2.20 GHz, GPU는 NVIDIA V100을 사용하였다.

Table 1. CIFAR-10 데이터셋에 대한 각 모델의 test error rate 비교이다. r은 block의 residual function 개수를 나타내며, 각 모델의 parameter 개수는 모두 약 1.7M이다. 실험 결과는 5번의 수행에 대한 median(mean±std)의 꼴로 나타내었다.

MODEL	DEPTH	R	CIFAR-10(%)
PRE-RESNET (HE ET AL., 2016)	110	1	6.37
	164	1	5.46
MULTI-RESNET (ABDI & NAHAVANDI, 2017)	8	23	7.37
	14	10	6.42
	30	4	5.89
STOCHASTIC DEPTH (HUANG ET AL., 2016)	110	1	5.25
RDM-RESNET (MINE)	56	2	5.98(5.98±0.20)
	83	2	5.41(5.34±0.12)

4.3. Results

실험을 수행한 결과 및 기존 모델과의 비교는 Table 1에 나타냈으며, Figure 2은 학습 과정의 오차율 변화를 나타낸다. CIFAR-10에 대해 RDM-ResNet-56은 5.98%의 오차율을 얻었으며, RDM-ResNet-83은 5.41%의 오차율을 보였다. 각각 basic block과 bottleneck block을 기반으로 하는 모델임을 고려하면 둘 다 기존 ResNet보다 성능이 향상된 결과를 얻었음을 알 수 있다. 특히 RDM-ResNet-83은 다른 모든 Multi-ResNet보다도 높은 정확도를 보였다. 이는 ReLU dropped function을 적용한 multi-residual의 구조로 ensemble의 다양성을 얻어 더 효율적인 학습이 가능했기 때문이라고 볼 수 있다. Stochastic depth를 적용한 모델보다는 정확도가 높지 않았지만, 이는 서로 양립되는 구조가 아니므로 기존의 ResNet 대신 RDM-ResNet에 stochastic depth를 적용한다면 더 뛰어난 성능을 얻을 수 있다.

Multi-ResNet과 RDM-ResNet-56은 모두 basic block 구조를 기반으로 한다. RDM-ResNet-56은 Multi-ResNet의 8-layer와 14-layer 모델보다는 좋은 성능을 보였지만, 30-layer의 모델보다는 좋지 않은 성능을 보였다. 이러한 결과와 identity mapping은 parameter를 포함하지 않는다는 점을 고려하면 residual function이 2개인 경우 가장 효과적인 학습이 가능하다는 추측은 틀렸을 수 있다. 다만 전체 parameter의 개수가 일정하면서 multiplicity가 최대가 되는 경우는 residual function이 1인 기존 ResNet 구조이며, ResNet에 비해 다른 모델이 더 뛰어난 성능을 보이기 때문에 모델의 성능이 단순히 multiplicity에 의존한다는 가정부터 틀렸을 수 있다.

연구 결과를 통해 residual networks의 성능은 단순히 valid path의 개수와 상관관계를 가지는 것이 아니며, 학습에 큰 영향을 미치는 effective path의 개수와 상관관계를 가진다고 조심스럽게 추측해본다. Parameter의 개수가 동일하다면 깊이가 얕은 모델은 전체적으로 path의 깊이가 얕지만 residual function이 증가하여 function을 지나는 경우가 많아진다. 반대로 깊이가 깊은 모델은 전체적으로 path의 깊이가 깊지만 residual function을 지나는 경우가 적으며, multiplicity가 더 클 수 있다. 실험 결과 깊이가 깊거나 얕다고 꼭 성능이 좋지 않고 중간인 경우가 더 좋았던 이유를 이를 통해 설명할 수 있다.

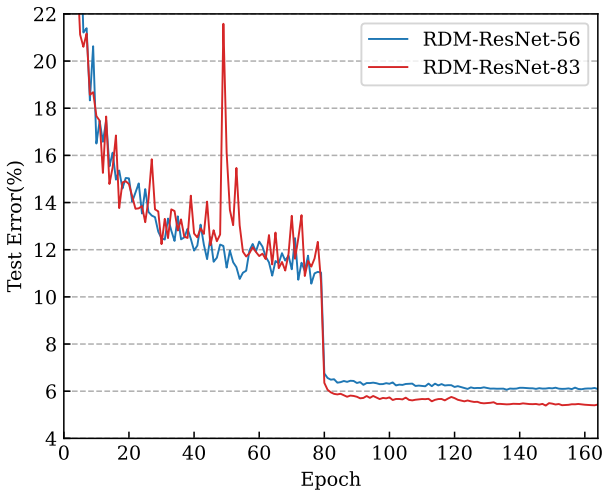

Figure 2. CIFAR-10 데이터셋에 대한 각 모델의 학습 과정에서 얻은 test error를 나타낸다. 각 곡선은 5번 시행의 평균을 나타낸다.

5. Conclusion

본 연구에서는 ResNet과 Multi-ResNet의 한계를 극복하기 위해 ReLU dropped multi-residual block을 사용하는 RDM-ResNet을 제안하였으며, basic과 bottleneck 기반 모델 모두 ResNet보다 향상된 정확도를 보여주었다. 이는 앞서 이야기했듯 ReLU dropped function으로 residual networks의 다양한 ensemble을 얻을 수 있기 때문으로 보인다. 또한 RDM-ResNet은 ResNet과 같이 간단한 구조로 구성되어 있기 때문에 활용 가능성이 크며, stochastic depth나 다양한 augmentation 등을 적용하면 정확도가 더욱 향상될 것이다.

RDM-ResNet에서 effective path를 이루는 layer의 범위가 $n_1 \leq k \leq n_2$일 때 effective path의 개수는 Appendix B에 따라 $(1+r)^b \int_{n_1/\alpha}^{n_2/\alpha} Bin(b, \frac{r}{1+r})$이다. 이러한 식 등을 통해 effective path가 최대가 되는 경우를 찾아, 그 때 실제로 가장 효율적인 학습이 가능한 지 확인할 수 있을 것이다. 다만 이를 위해서는 각 깊이 또는 parameter에 따른 effective path의 범위에 대한 연구가 선행되어야 한다.

References

Abdi, M. and Nahavandi, S. Multi-residual networks: Improving the speed and accuracy of residual networks, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition, 2015.

He, K., Zhang, X., Ren, S., and Sun, J. Identity mappings in deep residual networks, 2016.

Huang, G., Sun, Y., Liu, Z., Sedra, D., and Weinberger, K. Deep networks with stochastic depth, 2016.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C., Bottou, L., and Weinberger, K. (eds.), *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989. doi: 10.1162/neco.1989.1.4.541.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition, 2015.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. Going deeper with convolutions, 2014.

Veit, A., Wilber, M., and Belongie, S. Residual networks behave like ensembles of relatively shallow networks, 2016.

Zhao, G., Zhang, Z., Guan, H., Tang, P., and Wang, J. Rethink relu to training better cnns, 2018.

A. Edges and Multiplicity

모델의 전체 edge 개수 n은 block의 개수 b와 한 block의 edge 개수 k에 대해 $n = bk$의 관계를 가지며, multiplicity m은 $m = k^b$의 관계를 갖는다. 이는 $m = k^{n/k}$로 정리할 수 있다. Edge의 개수가 일정한 경우 n이 상수이므로 $y = k^{1/k}$가 최대일 때 multiplicity가 최대이다. 양변에 log를 취하면 $logy = \frac{1}{k}logk$을 얻으며, 양변을 미분하고 식을 정리하면 $\frac{dy}{dk} = \frac{1}{k^2}(1 - \log k)$을 얻는다. 해당 식이 0이 되는 경우는 $k = e$일 때이며, 따라서 multiplicity도 $k = e$일 때 최대가 된다. 다만 edge의 개수는 자연수만 가능하며, $k = 2$일 때와 $k = 4$일 때의 multiplicity 값이 동일하기 때문에 $k = 3$일 때 multiplicity가 최대가 된다. Block의 edge는 1개의 identity mapping과 r개의 residual function으로 구성되어 $k = 1 + r$이기 때문에 $r = 2$인, residual function이 2개인 경우에 multiplicity가 최대가 됨을 알 수 있다.

B. Parameters and Multiplicity

Multiplicity를 m, residual block의 개수를 b, residual function의 개수를 r이라고 하면 $m = (1 + r)^b$의 관계를 갖는다. 이때, residual function의 개수에 따라 각 block에서 function을 지날 확률은 $\frac{r}{1+r}$이다. 따라서 전체 block을 모두 지나며 residual function을 k개 지날 확률은 이항분포에 따라 $k \sim Bin(b, \frac{r}{1+r})$이다.이때 effective path를 이루는 layer의 범위가 $n_1 \leq k \leq n_2$라고 한다면 해당 경우의 확률을 모두 더해 multiplicity m과 곱하여 effective path의 개수를 얻을 수 있으므로 effective path의 개수 $e_p = (1 + r)^b \int_{n_1/\alpha}^{n_2/\alpha} Bin(b, \frac{r}{1+r})$이다. 이는 모델이 충분히 커서 첫번째와 마지막 layer는 무시할 수 있으며, 각 layer의 parameter 수는 동일하다는 가정 하에 성립한다.