ResNets に対する新たな正則化手法 ShakeDrop の提案

山田 良博[1,a)]　岩村 雅一[1,b)]　黄瀬 浩一[1,c)]

概要

本稿では，ResNet 及びその派生手法に対する新たな確率的正則化手法，ShakeDrop を提案する．ShakeDrop は摂動を含む不安定な学習と通常の学習を確率的に切り替えることによって，摂動による正則化効果を享受しつつ，学習の安定化を実現した．ShakeDrop は多様な ResNet とその派生手法に適用可能であり，従来手法 Shake-Shake よりも利便性に優れ，一般物体認識精度が高い．CIFAR-100 を用いた実験では，従来の手法の認識精度を 3%以上改善し，投稿時点で世界最高の認識精度 87.81%を達成した．

1. はじめに

ResNet [1] は 100 層を超える多層の Convolutional Neural Network (CNN) を実現し，当時の一般物体認識の世界最高精度を達成した．ResNet の根幹を成す Residual Block は，入力 x に対する出力 $G(x)$ が畳み込み処理 $F(\cdot)$ を用いて下記の式で表される．

$$G(x) = x + F(x) \tag{1}$$

ResNet 以降，Residual Block が式 (1) と同じ形で表される Wide ResNet [2] や PyramdNet [3]，$G(x) = x + F_1(x) + F_2(x) + \ldots$ で表される ResNeXt [4]（図 1(a)）といった派生手法が提案され，相次いで当時の一般物体認識における世界最高精度を達成した．

ネットワーク構造の工夫が進む一方で，異なる形の工夫によって認識精度を向上させた手法が Shake-Shake [5]（図 1(b)）である．CNN は forward path と backward path からなる一連の学習過程によって，重みを更新する．通常は望ましい出力に近づくように重みを更新するところを，Shake-Shake は forward path と backward path で異なる乱数を掛ける「確率的な正則化」によって，敢えて計算を乱す．これは常識を覆す工夫であったが，Shake-Shake はこの工夫によって当時の一般物体認識における世界最高精度を更新した．しかし Shake-Shake は Residual Block が $G(x) = x + F_1(x) + F_2(x)$ で表される一部の ResNeXt を前提としているため，式 (1) の形で表される ResNet 等に導入出来ない．

本稿では，式 (1) で表される ResNet 等に適用できる，Shake-Shake と同様の確率的な正則化手法を提案する．ただし，単に Shake-Shake における摂動を式 (1) に持ち込むだけでは，強い摂動によって学習が不安定になる．そこで，学習の安定化のために，Stochastic Depth (ResDrop) [6] で提案された正則化手法を本来の使途とは異なる形で導入する．提案手法は摂動を含む不安定な学習と通常の学習を確率的に切り替えることによって，強い摂動の恩恵を享受しつつ，安定した学習を実現する．その際，Residual Block の $F(x)$ に負の係数をも乗じる強い摂動を加えることで，従来手法より認識精度を向上させている．

2. 提案手法

2.1 Shake-Shake の考察と 1-branch Shake

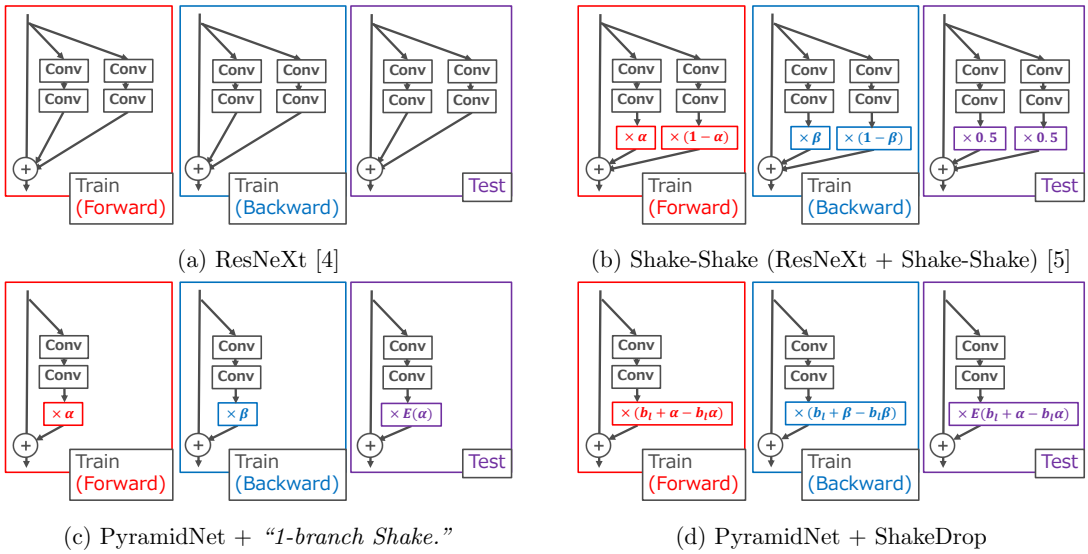
従来手法 Shake-Shake の確率的な正則化は以下の式で表される．

$$G(x) = \begin{cases} x + \alpha F_1(x) + (1-\alpha)F_2(x), & \text{in forward path} \\ x + \beta F_1(x) + (1-\beta)F_2(x), & \text{in backward path} \\ x + 0.5F_1(x) + 0.5F_2(x), & \text{otherwise.} \end{cases} \tag{2}$$

ただし α，β はそれぞれ $\alpha \in [0,1]$，$\beta \in [0,1]$ の一様乱数である．テスト時には α 及び $1-\alpha$ の期待値 0.5 を用いる．

式 (2) の解釈はこれまで与えられていないが，[7] の知見に基づくと，次のように解釈できる．まず，式 (2) から，Shake-Shake の foraward path では $F_1(x)$ と $F_2(x)$ の内分点を求めている．[7] では，２つのデータの特徴表現の加重和を取ることで，新たなデータが生成できることが示されている．すなわち，data augmentation である．したがって，Shake-Shake の forward path では，乱数 α に基づいて，学習データに含まれないデータが特徴空間で生成されていると考えられる．一方，backword path については，[5] で実験的に検討されており，β が α と離れる程，正しく重みが更新されないことが示されている．すなわち，Shake-Shake は，forward path では特徴空間で新たなデー

[1] 大阪府立大学 大学院工学研究科
[a)] yamada@m.cs.osakafu-u.ac.jp
[b)] masa@cs.osakafu-u.ac.jp
[c)] kise@cs.osakafu-u.ac.jp



(a) ResNeXt [4]　(b) Shake-Shake (ResNeXt + Shake-Shake) [5]

(c) PyramidNet + *"1-branch Shake."*　(d) PyramidNet + ShakeDrop

図 1: ネットワーク構造の模式図

タを生成し，backward path では重み更新を乱していると考えられる．

この知見に基づき，式 (1) の形で表される ResNet 等に適用可能な正則化として次式を考え，1-branch Shake と名付ける（図 1(c)）．

$$G(x) = \begin{cases} x + \alpha F(x), & \text{in forward path} \\ x + \beta F(x), & \text{in backward path} \\ x + 0.5F_1(x), & \text{otherwise.} \end{cases} \quad (3)$$

ただし α，β は Shake-Shake に習って，それぞれ $\alpha \in [0,1]$，$\beta \in [0,1]$ の一様乱数である．テスト時には α の期待値 0.5 を用いる．なお，[7] では，２つのデータの特徴表現の内分点のみでなく，ノイズを加えることで新たなデータを生成できることを示しており，data augmentation の観点からは，これでも十分に思える．しかし，単に α, β を適用するだけでは学習が安定しないことが簡単な実験で確認された[*1]．そのため，1-branch Shake では学習の安定化が必要になる．

2.2 ShakeDrop

1-branch Shake の否定的な結果は，学習を安定化させる工夫の必要性を示唆している．そこで，ResDrop を通常とは異なる目的で用いることを提案する．本来 ResDrop は，深い ResNet の学習時に勾配が消失する問題を解決すべく，確率的に選んだ一部の層を無視して学習を繰り返すという手法である．すなわち，学習時に見かけの層数を浅くすることにより，勾配消失を防ぐ手法である．しかし，この手法は 1-branch Shake にこのまま適用しても学習の安定化にはつながらないと考えられる．何故なら，1-branch Shake の問題は層が深いことでは無く，重みの更新時に著しい摂動が加えられている事である．そこで本研究では，学習時に一部の層を無視する代わりに，その部分だけ正しく重みが更新されないネットワークと置き換える．これにより，ネットワーク全体としては正しい重み更新が行われるものの，一部では正しくない重みが更新され，それらのバランスにより，ネットワークの学習が安定化しながら強い摂動が加わることが期待できる．この工夫を加えて学習を安定化した手法を ShakeDrop と呼ぶことにする．

提案手法 ShakeDrop には，確率 $p_l = 1 - \frac{l}{2L}$ のベルヌーイ分布の二値乱数 $b_l \in B(p_l)$ による制御を導入される．これは ResDrop で提案された学習安定化手法である Linear Decay Rule に基づく制御である．ただし，L と l は，入力から数えて L 個中 l 番目の Residual Block を表す．提案手法 ShakeDrop（図 1(d)）は次式で与えられる．

$$G(x) = \begin{cases} x + (b_l + \alpha - b_l\alpha)F(x), & \text{in forward path} \\ x + (b_l + \beta - b_l\beta)F(x), & \text{if backward path} \\ x + E(b_l + \alpha - b_l\alpha)F(x), & \text{otherwise.} \end{cases} \quad (4)$$

ただし α，β はそれぞれ $\alpha \in [-1,1]$，$\beta \in [0,1]$ の一様乱数である．また，$E(\cdot)$ は期待値である．α，β，b_l はパラメータ更新の度に決定され，二値乱数 b_l は制御に用いられる．$b_l = 1$ のとき α，β は打ち消され，式 (1) になる．$b_l = 0$ のとき，前向きと後ろ向きはそれぞれ $G(x) = x + \alpha F(x)$，$G(x) = x + \beta F(x)$ となり，式 (2) と同様の摂動を含む計算となる．テスト時には前向きの $(b_l + \alpha - b_l\alpha)$ の期待値を利用する．

予備実験の結果，提案手法 ShakeDrop は下記の条件を満たすネットワーク構造であれば機能することが分かった．
(1) 加算の直前に Batch Normalization (BN) が存在する．

[*1] PyramidNet-110 $\alpha = 270$ に 1-branch Shake を適用した場合，CIFAR-100 のエラー率が 77.99%だった．

表 1: CIFAR-100 における複数の α と β の範囲での PyramidNet+ShakeDrop の Top-1 error 率の 4 回平均 (%).

Case	α	β	Error (%)	Note
A	1	1	18.01	PyramidNet
B	0	0	17.74	PyramidDrop[9]
C	1	$[0,1]$	18.80	
D	1	$[-1,1]$	21.69	
E	$[0,1]$	1	38.48	
F	$[0,1]$	0	19.68	
G	$[0,1]$	$[0,1]$	18.27	
H	$[0,1]$	$[-1,1]$	20.61	
I	$[-1,1]$	1	18.68	
J	$[-1,1]$	0	17.28	
K	$[-1,1]$	$[0,1]$	**16.22**	
L	$[-1,1]$	$[-1,1]$	18.26	

(2) 加算の直後に ReLU が存在しない.

この 2 つの条件を満たすものの多くは，EraseReLU として提案されたネットワーク構造であり, 広範な検証の中でオリジナルの構造ののエラー率を下回る優れた構造であることが報告されている [8].

3. 実験

3.1 ShakeDrop における α と β の影響

α 及び β について表現可能なパラメータ範囲における実験結果を表 1 に示す．この結果から，α と β の組み合わせの効果を確認できる．B($\alpha = 0$，$\beta = 0$) の PyramidDrop（PyramidNet に ResDrop を適用）を除いて，A($\alpha = 1$，$\beta = 1$) の PyramidNet のエラー率を下回ったのは，J ($\alpha = [-1,1]$，$\beta = 0$) と K ($\alpha = [-1,1]$，$\beta = [0,1]$) だけだった．これらの中で，K が最も優れていた．$\alpha = 1$ と $\alpha \in [0,1]$ の場合は精度が改善しなかった．特に興味深いのは I ($\alpha \in [-1,1], \beta = 1$), J, K と L ($\alpha \in [-1,1], \beta \in [-1,1]$) の 4 つの条件の比較である．これらは全て同じ範囲の α (すなわち $\alpha \in [-1,1]$) の組み合わせである．J と K は PyramidNet よりも優れており，強い摂動が効果的であると言える．一方，同様に強い摂動を含む I と L は PyramidNet より劣っていた．これらの違いは以下のように解釈できる．I と J の比較から, 強い摂動が与えられた際，摂動を与えるよう決められた Residual Block では重みを更新すべきではない ($\beta = 0$ を採用し，$\beta = 1$ を避けるべき) と分かる．しかしながら，J ($\beta = 0$) と K ($\beta \in [0,1]$) の比較では，K ($\beta \in [0,1]$) が I ($\beta = 1$) や J ($\beta = 0$) より優れていた．これは摂動を与える Residual block も一部を更新すべきだと示唆している．一方，L ($\beta \in [-1,1]$) は I ($\beta = 1$) より優れていたが，J ($\beta = 0$) や K ($\beta \in [0,1]$) より劣っていた．これは β の範囲が負の値を含むと摂動が強すぎるために，認識精度の改善に寄与しないことを示している．

3.2 従来の正則化手法との比較実験

提案手法 ShakeDrop の有効性を確認するために, 一般物体認識用データセット CIFAR-100 [10]，Tiny ImageNet [*2] を用いて実験を行った.

Residual Block が式 (1) と $G(x) = x + F_1(x) + F_2(x)$ で表されるネットワークを用意し, 正則化なし (Vanilla), ResDrop，Shake-Shake，ShakeDrop（提案手法）の認識精度を比較した．$G(x) = x + F_1(x) + F_2(x)$ で表されるネットワークでは，確率的な正則化の導入方法として，正則化モジュールを add モジュールの前と後に入れる 2 パターンが考えられる．本稿では前に入れるのを Type-A，後に入れるのを Type-B と呼称し，それぞれについて検証を行う．ただし実験における一部のネットワークは，オリジナルのものではなく提案手法 ShakeDrop が適用可能な構造（EraseReLU や Wide ResNet に BN を導入）に変更した.

CIFAR-100 および Tiny ImageNet の結果を表 2 に示す．提案手法 ShakeDrop は表 2 中の大半の条件で最良値を示しており，優れた正則化として機能していた．また, 既存手法 Shake-Shake との比較において提案手法 ShakeDrop は優れた結果を示している．Type-A と Type-B については，Type-B の方が優れた認識精度を達成する傾向が確認された.

3.3 他手法との比較

最先端との手法の比較では, 1800 epoch の Cosine Learning Scheduling [5] と Random Erasing [12] を導入した PyramidNet-272 ($\alpha = 200$) で，提案手法 ShakeDrop が CIFAR-10/100 におけるエラー率 2.31%と 12.19%をそれぞれ達成し，正則化無し（Vanilla; その他の条件は同じ）のエラー率 3.42%と 16.66%を大幅に下回った．CIFAR-100 におけるエラー率 12.19%は投稿時点 (2018 年 3 月 22 日) での世界最高精度であった.

4. まとめ

一般物体認識において，従来手法 Shake-Shake に代わる新たな確率的正則化手法として ShakeDrop を提案し，ResNet 及びその派生手法での有効性を検証した．CIFAR-100 および Tiny ImageNet を用いた実験によって, 提案手法 ShakeDrop による認識精度の改善を確認し，CIFAR-100 において投稿時点での世界最高精度を達成した．本研究で提案した学習の安定化の方策は，ShakeDrop に適用した摂動以外でも有用な可能性があり，確率的正則化手法を今後研究する上での基盤技術になりうると考えている.

謝辞

[*2] Tiny ImageNet データセットは，ImageNet データセット [11] の一部から成るデータセットである．200 クラスから成り，各クラスには学習用に 500 枚，評価用に 50 枚が含まれている．https://tiny-imagenet.herokuapp.com/

表 2: CIFAR-100 及び Tiny-ImageNet における Top-1 error 率 (%). $^+$ は 4 回の結果の平均である.* は [3] の結果である.

(a) Residual Block が $G(x) = x + F(x)$ で表される場合 (ResNet, ResNeXt, Wide-ResNet PyramidNet)

Methods	Regularization	CIFAR-100 (%)	Tiny-ImageNet (%)
ResNet-110 <Conv-BN-ReLU-Conv-BN-add>	Vanilla	24.93	**41.24**
	ResDrop	22.88	42.50
	ShakeDrop	**21.70**	43.88
ResNet-164 Bottleneck <Conv-BN-ReLU-Conv-BN-ReLU-Conv-BN-add>	Vanilla	21.96	**36.52**
	ResDrop	20.35	38.09
	ShakeDrop	**19.58**	38.25
Wide-ResNet-28-10k <BN-ReLU-Conv-BN-ReLU-Conv-BN-add>	Vanilla	24.24	37.88
	ResDrop	26.64	45.80
	ShakeDrop	**20.50**	**34.29**
ResNeXt-29 8-64d <Conv-BN-ReLU-Conv-BN-ReLU-Conv-BN-add>	Vanilla	20.25	34.21
	ResDrop	20.28	**33.98**
	ShakeDrop	**18.66**	34.96
PyramidNet-110 α270 <BN-Conv-BN-ReLU-Conv-BN-add>	Vanilla	$^+$18.01	36.52
	ResDrop	$^+$17.74	33.97
	ShakeDrop	$^+$**15.78**	**30.70**
PyramidNet-272 α200 <BN-Conv-BN-ReLU-Conv-BN-add>	Vanilla	*16.35	-
	ResDrop	15.94	
	ShakeDrop	**14.96**	

(b) Residual Block が $G(x) = x + F_1(x) + F_2(x)$ で表される場合　(ResNeXt)

Methods	Regularization	CIFAR-100 (%)	Tiny-ImageNet (%)
ResNeXt-164 2-1-40d Bottleneck <Conv-BN-ReLU-Conv-BN-ReLU-Conv-BN-add>	Vanilla	21.75	38.56
	ResDrop Type-A	20.44	36.98
	ResDrop Type-B	20.21	37.08
	Shake-Shake	22.51	38.03
	ShakeDrop Type-A	**19.19**	37.61
	ShakeDrop Type-B	**18.66**	**36.75**
ResNeXt-29 2-4-64d Bottleneck <Conv-BN-ReLU-Conv-BN-ReLU-Conv-BN-add>	Vanilla	×	34.30
	ResDrop Type-A	20.13	34.04
	ResDrop Type-B	19.01	32.90
	Shake-Shake	18.82	33.30
	ShakeDrop Type-A	**18.49**	38.05
	ShakeDrop Type-B	**17.80**	**32.05**

本研究は，JST CREST #JPMJCR16E1，JSPS 科研費 #25240028 と#17H01803 補助による．

参考文献

[1] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” Proc. CVPR, 2016.

[2] S. Zagoruyko and N. Komodakis, “Wide residual networks,” Proc. BMVC, 2016.

[3] D. Han, J. Kim, and J. Kim, “Deep pyramidal residual networks,” Proc. CVPR, 2017.

[4] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks,” Proc. CVPR, 2017.

[5] X. Gastaldi, “Shake-shake regularization,” arXiv preprint arXiv:1705.07485v2, 2017.

[6] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Weinberger, “Deep networks with stochastic depth,” arXiv preprint arXiv:1603.09382v3, 2016.

[7] T. DeVries and G.W. Taylor, “Dataset augmentation in feature space,” Proc. ICLR Workshop, 2017.

[8] X. Dong, G. Kang, K. Zhan, and Y. Yang, “EraseReLU: A simple way to ease the training of deep convolution neural networks,” arXiv preprint 1709.07634, 2017.

[9] Y. Yamada, M. Iwamura, and K. Kise, “Deep pyramidal residual networks with separated stochastic depth,” arXiv preprint arXiv:1612.01230, 2016.

[10] A. Krizhevsky, “Learning multiple layers of features from tiny images,” Technical report, Univ. of Toronto, 2009.

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” Proc. CVPR, 2009.

[12] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, “Random erasing data augmentation,” arXiv preprint arXiv:1708.04896, 2017.