*SIP* (2020), vol. 9, e9, page 1 of 7 © The Authors, 2020.

doi:10.1017/ATSIP.2020.7

ORIGINAL PAPER

# Theoretical analysis of skip connections and batch normalization from generalization and optimization perspectives

YASUTAKA FURUSHO AND KAZUSHI IKEDA

*Deep neural networks (DNNs) have the same structure as the neocognitron proposed in 1979 but have much better performance, which is because DNNs include many heuristic techniques such as pre-training, dropout, skip connections, batch normalization (BN), and stochastic depth. However, the reason why these techniques improve the performance is not fully understood. Recently, two tools for theoretical analyses have been proposed. One is to evaluate the generalization gap, defined as the difference between the expected loss and empirical loss, by calculating the algorithmic stability, and the other is to evaluate the convergence rate by calculating the eigenvalues of the Fisher information matrix of DNNs. This overview paper briefly introduces the tools and shows their usefulness by showing why the skip connections and BN improve the performance.*

**Keywords:** Deep neural networks, ResNet, Skip connections, Batch normalization

Received 19 December 2019; Revised 24 January 2020

## I. INTRODUCTION

Deep neural networks (DNNs) have been changing the history of machine learning in terms of performance [1–4]. Although their high performance originates from their exponential expressive power owing to the depth [5–8], such deep networks are difficult to train owing to the so-called vanishing gradient. In fact, a classic feedforward network with 56 layers had a larger empirical risk than one with 20 layers [9], implying that the network is not fully trained. To overcome this degradation problem, many heuristics have been proposed and some of them improved their performance. In particular, skip connections in the residual networks (ResNet) [9,10] and batch normalization (BN) [11] enable extremely deep NNs (1202 layers) to be trained with a small empirical risk and a small expected risk. In addition, a ResNet skipping two layers showed better performance than a ResNet skipping one layer or a standard feedforward neural network [9].

In the case of the linear model, the expected risk and empirical risk have been theoretically evaluated. The model selection theory such as AIC [12] and MDL [13] evaluated the expected risk by measuring the gap between a trained model and a true model that generate data using the Cramer–Rao bound. The convex analysis evaluated how fast the empirical risk decreases by calculating properties of loss landscape such as the strong convexity, Lipschitzness, and smoothness [14]. However, these theoretical analyses cannot be applied to recently proposed DNN techniques since DNNs have singular points [15,16] and are non-convex [17] even when its activation function is the identity. The singular points make the Fisher information matrix degenerate and thus the Cramer–Rao bound doesn't hold, which implies that the classical model selection theory cannot be applied. The non-convexity also makes the convex analysis difficult to apply.

Regardless of the difficulty, the recent popularity of DNNs has promoted the development of new methodologies as below for theoretical analyses of DNN techniques [18–26]. One is to calculate the algorithmic stability to evaluate the generalization gap defined as the difference between the expected risk and empirical risk. The other is to calculate the eigenvalues of the Fisher information matrix of DNNs to evaluate how fast the empirical risk decreases around the minimal point.

Since this method is widely applicable and has succeeded in quantifying the effectiveness of skip connections and BN, this overview paper briefly introduces the method and shows how to apply it to DNN techniques.

## II. PROBLEM FORMULATION

### A) Samples for training

Let the training set be denoted by $S = \{z(n)\}_{n=1}^{N}$, where each training example $z(n)$ consists of an input $x(n)$ and the

Nara Institute of Science and Technology, Ikoma, Nara, 8916-5, Japan

**Corresponding author:**
Kazushi Ikeda
Email: kazushi@is.naist.jp

corresponding target $y(n)$. An example $z(n) = (x(n), y(n))$ is independently identically chosen from a probability distribution $\mathcal{D}$ on the joint space $\mathcal{Z}$ of the input space $\mathcal{X}$ and the output space $\mathcal{Y}$. Note that the indices of the examples are omitted if they are clear from the context.

### B) Training of deep neural network

The DNN $f : \mathcal{X} \times \Theta \to \mathcal{Y}$ with parameters $\theta \in \Theta$ predicts the corresponding target $y \in \mathcal{Y}$ for a given input $x \in \mathcal{X}$, where $\Theta$ is the parameter space. Its performance is measured on the basis of the expected risk,

$$R(\theta) = \mathbb{E}_z \left[ \ell(z, \theta) \right], \tag{1}$$

where $\ell(z, \theta) = \frac{1}{2} \left\| f(x; \theta) - y \right\|^2$ is the squared loss. The parameters are trained by the gradient descent (GD),

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta R_S(\theta_t), \tag{2}$$

to minimize the empirical risk,

$$R_S(\theta) = \frac{1}{|S|} \sum_{z \in S} \ell(z, \theta), \tag{3}$$

instead of the expected risk because the data distribution $\mathcal{D}$ is not known, where $\theta_t$ and $\eta$ denote the output of the GD at the $t$th update and the learning rate, respectively. Note that the parameters $\theta_o$ are initialized according to the method specified in each subsequent analysis section.

### C) Decomposition of expected risk

The expected risk is decomposed into two components,

$$R(\theta_t) = \underbrace{R(\theta_t) - R_S(\theta_t)}_{\text{generalization gap}} + \underbrace{R_S(\theta_t)}_{\text{empirical risk}} . \tag{4}$$

The generalization gap measures the difference between the expected risk and empirical risk, while the empirical risk expresses how fast the GD optimizes the parameters. Recent analytical techniques evaluate each of them as described next.

## III. GENERALIZATION GAP

### A) Formulation of ResNets

We evaluate the effectiveness of ResNets by deriving upper bounds of the generalization gaps of the following linear DNNs: $f : \mathbb{R}^D \times \Theta \to \mathbb{R}^D$, where $\theta$ denotes the parameters of each NN.

**MLP:**

$$f(x; \theta) = \prod_{l=1}^{L} W^l x, \tag{5}$$

**ResNet1:**

$$f(x; \theta) = \prod_{l=1}^{L} (W^l + I) x, \tag{6}$$

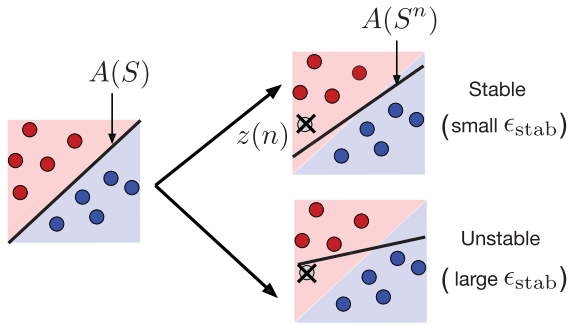


**Fig. 1.** Algorithmic stability. It measures how much the removal of one example $z(n)$ from the training set $S$ affects the trained model $A(S^n)$.

**ResNet2:**

$$f(x; \theta) = \prod_{l=1}^{L/2} (W^{2l} W^{2L-1} + I) x. \tag{7}$$

Although these DNNs are linear with respect to the input $x$ and have the same expressive ability, they are nonconvex with respect to the parameter $\theta$ and have different parameter representations.

### B) Algorithmic stability

A training algorithm $\mathcal{A}$ receives the training set $S$ and outputs a trained model $\mathcal{A}(S)$. The algorithmic stability measures how much the removal of one example $z(n)$ from the training set $S$ affects the trained model $\mathcal{A}(S^n)$ in terms of the expected loss, where $S^n = S \backslash z(n)$ (Fig. 1).

**Definition 1** (Definition 4 in [18]). *The training algorithm $\mathcal{A}$ is pointwise hypothesis stable if there exists $\epsilon_{stab}$ such that $\forall n \in [N]$,*

$$\mathbb{E}_{\mathcal{A},S} \left[ |\ell(z(n), \mathcal{A}(S)) - \ell(z(n), \mathcal{A}(S^n))| \right] \leq \epsilon_{stab}, \tag{8}$$

*where the expectation is taken with respect to the randomness of the algorithm $\mathcal{A}$ and the training set $S$.*

A stable algorithm $\mathcal{A}$ with small $\epsilon_{stab}$ outputs a trained model with a small generalization gap in the framework of statistical learning theory.

**Theorem 1** (Theorem 11 in [18]). *If the training algorithm $\mathcal{A}$ is pointwise hypothesis stable, the following holds with probability at least $1 - \delta$ :*

$$R(A(S)) - R_S(A(S)) \leq \sqrt{\frac{M^2 + 12MN\epsilon_{stab}}{2N\delta}}, \tag{9}$$

*where $M$ is an upper bound of the loss function.*

The algorithmic stability $\epsilon_{stab}$ of the GD depends on the flatness of the loss landscape around a global minimum.

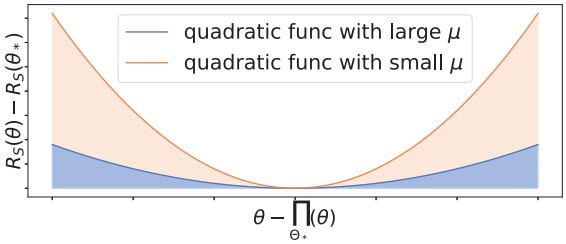


**Fig. 2.** Excess risk is smaller than a quadratic function of parameters $\theta$. The constant $\mu$ for the PL condition controls its flatness.

**Definition 2** (Definition 4 in [19]). *The empirical risk $R_S$ satisfies the Polyak–Lojasiewicz (PL) condition with a constant $\mu$ if the following holds:*

$$\forall \theta_t \in \Theta, \quad R_S(\theta_t) - R_S(\theta^*) \leq \frac{1}{2\mu} \|\nabla_\theta R_S(\theta_t)\|^2, \qquad (10)$$

*where $\theta_*$ is a global minimum.*

Here, the constant $\mu$ for the PL condition expresses the flatness of the loss landscape around a global minimum. If the empirical risk $R_S$ satisfies the PL condition with $\mu$ and is $\beta$-smooth, that is, the gradient is $\beta$-Lipschitz, then the following inequality holds:

$$R_S(\theta_t) - R_S(\theta_*) \leq \frac{\beta^2}{2\mu} \left\| \theta_t - \prod_{\Theta_*}(\theta_t) \right\|^2, \qquad (11)$$

where $\prod_{\Theta_*}(\theta_t)$ is the projection of $\theta_t$ on the set of global minima $\Theta_*$. This shows that an excess risk is smaller than a quadratic function of parameters and that the constant $\mu$ controls its flatness (Fig. 2).

A training algorithm has better stability if it converges faster and its loss function has flatter minima.

**Theorem 2** (Theorem 3 in [19]). *Suppose that the empirical risk $R_S$ satisfies the PL condition with $\mu$ and the loss function is $\alpha$-Lipschitz. If the training algorithm $\mathcal{A}$ converges parameters to the global minima $\theta_*$ with $\|\theta_t - \theta_*\| \leq \epsilon_t$, it is pointwise hypothesis stable, that is,*

$$\epsilon_{stab} \leq 2\alpha\epsilon_t + \frac{2\alpha^2}{\mu(N-1)}. \qquad (12)$$

## C) Upper bounds of the generalization errors

We applied the above stability analysis to MLP, ResNet1, and ResNet2 under Assumptions 1 and 2 [24].

**Assumption 1.** *The input correlation matrix is the identity, $\sum_{(x,y)\in S} xx^T = I$.*

**Assumption 2.** *The eigenvalues of the output–input correlation matrix $\sum_{(x,y)\in S} yx^T$ are greater than one.*

These assumptions are rather weak since a dataset satisfies Assumption 1 if it is preprocessed by principal component analysis (PCA) whitening. In addition, the PCA-whitened MNIST dataset satisfies Assumption 2.

**Table 1.** PL condition of the $L$-layer linear DNNs.

| Model | Constant $\mu$ for the PL condition | Reference |
|---|---|---|
| MLP | $La_{\min}^{2L-2}/C$ | [19] |
| ResNet1 | $L(1-a_{\max})^{2L-2}/C$ | [20] |
| ResNet2 | $L(1-a_{\max}^2)^{L-2}a_{\min}^2/C$ | [24] |

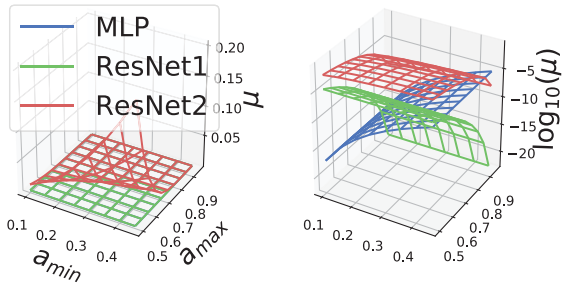


**Fig. 3.** Constants $\mu$ for the PL-condition of the 10-layer linear DNNs.

**Table 2.** Convergence of the $L$-layer linear DNNs.

| Model | Speed of the parameter convergence $\epsilon_t$ |
|---|---|
| MLP & ResNet1 | $(1-\eta L\gamma^{2(L-1)})^t\epsilon_0$ |
| ResNet2 | $\left(1-\eta L \underbrace{\frac{\gamma^2-1}{2\gamma^2}}_{\leq 1}\gamma^{2(L-1)}\right)^t \epsilon_0$ |

**Theorem 3** (Theorems 3 and 4 in [24]). *Initialize the linear DNNs by orthogonal initialization [27]. Then, under Assumptions 1 and 2, ResNet2 has flatter minima, as shown in Table 1, where $a_{\min}$ and $a_{\max}$ are the minimum and maximum singular values of the weights, during training, respectively, and $C$ is a constant (Fig. 3). In addition, its parameters converge slower than the other DNNs, as shown in Table 2, where $\gamma$ is the minimum singular value of the transform by the layer during training.*

**Remark 1.** *Theorem 3 implies that ResNet2 has a smaller generalization gap than MLP or ResNet1 when the parameters are updated by the GD a sufficient number of times.*

## D) Numerical experiments

To confirm the validity of the above analyses and the applicability to the DNNs with the ReLU activation function, $\phi(\cdot) = \max\{0, \cdot\}$, some numerical experiments were carried out. The dataset was the MNIST dataset [28] after PCA whitening, so that $\forall d \in [D], \mathbb{E}[x_d] = 0$ and $\mathrm{Var}(x_d) = 1$, and projection into the principal subspace of 10 dimensions.

We initialized the DNNs with 10 hidden units in each layer by the orthogonal initialization and trained these by the GD. During the training, the training loss, the test loss, and the approximate value of the stability $\epsilon_{stab}$,

$$\frac{1}{N}\sum_{n=1}^{N} |\ell(z(n), A(S)) - \ell(z(n), A(S^n))|, \qquad (13)$$

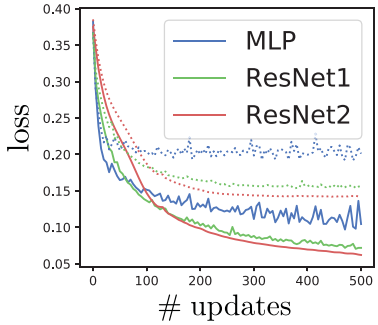


**Fig. 4.** Training loss (solid lines) and test loss (dotted lines) of the 10-layer DNNs with the ReLU activation.

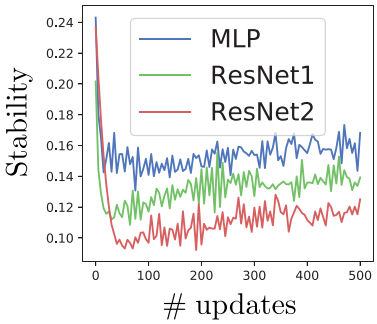


**Fig. 5.** Approximation of the stability $\epsilon_{stab}$ of the 10-layer DNNs with the ReLU activation.

were calculated every five updates (Figs. 4 and 5). The results show that ResNet2 had a greater stability and a smaller generalization gap than the other DNNs and that the analyses are valid even for the DNNs with the ReLU activation.

## IV. EMPIRICAL RISK

### A) Formulation of batch normalization

We evaluate the effectiveness of BN by deriving the empirical risk of the following DNNs with the ReLU activation $\phi(\cdot) = \max\{0, \cdot\}$:

**ResNet:**

$$u_i^l = \phi(h_i^{l-1}), \quad h_i^l = \sum_{j=1}^{D} W_{i,j}^l u_j^l + h_i^{l-1}, \qquad \hat{y} = 1^T h^L. \tag{14}$$

**ResNet with BN:**

$$u_i^l = \phi(\mathrm{BN}(h_i^{l-1})), \quad \mathrm{BN}(h_i^l) = \frac{h_i^l - \mathbb{E}_x\left[h_i^l\right]}{\sqrt{\mathrm{Var}\left(h_i^l\right)}},$$

$$h_i^l = \sum_{j=1}^{D} W_{i,j}^l u_j^l + h_i^{l-1}, \quad \hat{y} = 1^T h^L. \tag{15}$$

Here, $h^\circ = W^\circ x$ is the projection of the input, and the expectation of the BN is taken with respect to the input in the batch of the GD. Without the loss of generality, the projection matrix is a square matrix initialized by Xavier initialization [29], and the inputs in the training set are normalized to $\mathbb{E}_x[x_i] = 0$ and $\mathrm{Var}(x_i) = 1$.

### B) Hessian and Fisher information matrices

The empirical risk $R_S(\theta_t)$ is approximated by the second-order Taylor expansion around the minima $\theta_*$,

$$R_S(\theta_t) = R_S(\theta_*) + \frac{1}{2}(\theta_t - \theta_*)^T H(\theta_*)(\theta_t - \theta_*), \tag{16}$$

where $H(\theta_*) = \nabla_\theta \nabla_\theta R_S(\theta_*)$ is the Hessian matrix. The Hessian matrix is decomposed as $H(\theta_*) = U \Lambda U^T$, where $U$ and $\Lambda$ are a unitary square matrix comprising the eigenvectors and a diagonal matrix filled with the eigenvalues, respectively, which simplifies the empirical risk to

$$R_S(\theta_t) = R_S(\theta_*) + \frac{1}{2} v_t^T \Lambda v_t, \tag{17}$$

where $v_t = U^T(\theta_t - \theta_*)$, and the GD to

$$v_{t+1} = (I - \eta\Lambda)\, v_t. \tag{18}$$

Let $\lambda_{\min}$ and $\lambda_{\max}$ be the minimum and maximum eigenvalues of $H(\theta_*)$, respectively. The GD converges when the learning rate is $\eta = 2/\lambda_{\max}$ and it converges fastest when the learning rate is $\eta = 1/\lambda_{\max}$. The fastest convergence rate is the reciprocal of the condition number, $\lambda_{\max}/\lambda_{\min}$ [21], as is well known in adaptive filtering theory [30]. However, the Hessian matrix and its eigenvalues are difficult to calculate owing to the complicated structure of DNNs.

Recently, the Fisher information matrix (FIM) of $p(x, y; \theta)$,

$$F(\theta) = \mathbb{E}_x\left[\nabla_\theta \log p(x;\theta)\, \nabla_\theta \log p(x;\theta)^T\right], \tag{19}$$

has been found to approximate the Hessian matrix of a DNN, $f(x;\theta)$, where $p(x, y; \theta) = p(x)p(y|x;\theta)$, $p(y|x;\theta) = \mathcal{N}(f(x;\theta), 1)$, and $p(x)$ is the probability of the input. In this case, the FIM is rewritten as

$$F(\theta) = \mathbb{E}_x\left[\nabla_\theta f(x;\theta)\, \nabla_\theta f(x;\theta)^T\right] \tag{20}$$

and the following holds:

$$H(\theta) = F(\theta) - \mathbb{E}_x\left[(y - f(x;\theta)) \cdot \nabla_\theta \nabla_\theta f(x;\theta)\right], \tag{21}$$

the second term of which is negligible when the error is small. In addition, the eigenvalues of the FIM of a sufficiently wide neural network do not change during training [31,32].

### C) Bounds of the empirical risk

We calculated the eigenvalues of the FIMs of the naive ResNet and the ResNet with BN averaged over the random He initialization [33] (expected FIM) under Assumptions 3 and 4 [25].

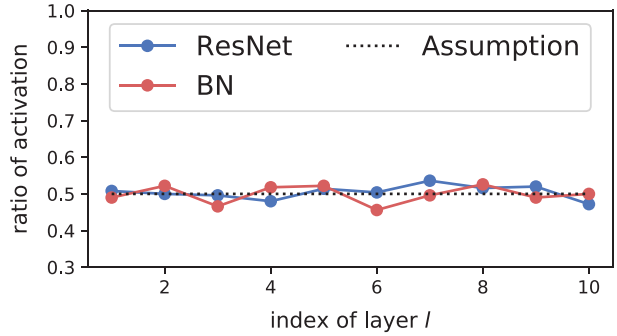


**Fig. 6.** Activation rate of the hidden units in each layer.

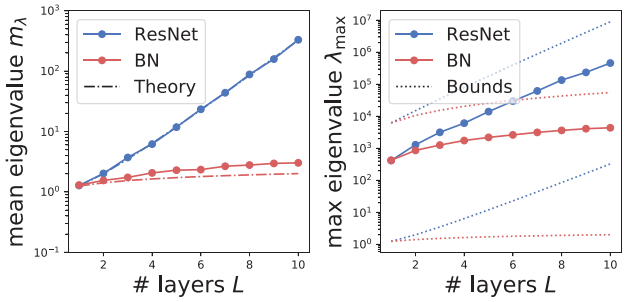


**Fig. 7.** Mean eigenvalues and maximum eigenvalues of the expected FIM.

**Assumption 3.** *The forward signal $u_i^l$ is independent of the backward error signal $\partial f(x;\theta)/\partial h_i^l$.*

**Assumption 4.** *Half of the hidden units per layer are active $\phi'(h_i^l) = 1$.*

Although Assumption 3 is rather unrealistic, some theorems have been derived on the basis of Assumption 3 and their results were in agreement with those of numerical experiments [22,34,35]. In addition, the binary class PCA-whitened MNIST dataset satisfies Assumption 4 (Fig. 6).

**Theorem 4** (Modification of Table 1 in [25]). *Under Assumptions 3 and 4, the maximum eigenvalue $\lambda_{max}$ of the expected FIM of the ResNet grows exponentially with the depth,*

$$m_\lambda \le \lambda_{max} \le (L+1)D^2 m_\lambda,$$
$$m_\lambda = \frac{L+4}{4L+4} \cdot 2^L, \tag{22}$$

*where $m_\lambda$ is the mean of all the eigenvalues $\{\lambda_i\}_{i=1}^{(L+1)D^2}$.*

**Remark 2.** *The learning rate of the GD must be exponentially small with respect to the depth of the ResNet for convergence of the parameters to the minima.*

**Theorem 5** (Modification of Table 1 in [25]). *Under Assumptions 3 and 4, BN relaxes the exponential growth of the eigenvalue to $L \log L$ order at most,*

$$m_\lambda \le \lambda_{max} \le (L+1)D^2 m_\lambda, \tag{23}$$
$$m_\lambda = \frac{H_{L+1}+1}{2}, \tag{24}$$

*where $H_L = \sum_{k=1}^{L} \frac{1}{k}$ is the harmonic number.*

**Remark 3.** *BN enables the GD to use a larger learning rate than that of the ResNet for convergence of the parameters to the minima.*

Note that our discussion is focused on the minima $\theta_*$. This is justified by the fact that the GD and the stochastic GD make the parameters into the minima under some conditions [31,36,37].

## D) Numerical experiments

To confirm the validity of the above analyses, some numerical experiments were carried out. The dataset was a subset of the MNIST dataset [28] with class labels of 0 and 1 after the PCA whitening so that $\forall d \in [D], \mathbb{E}[x_d] = 0$, and $\mathrm{Var}(x_d) = 1$, with projection into the principal subspace of 50 dimensions.

We initialized the ResNet and ResNet with BN, which have 50 hidden units in each layer, by the He initialization, calculated the mean eigenvalues and maximum eigenvalues of the expected FIMs of these DNNs, and found that the mean eigenvalues were in agreement with the theoretical values and that the maximum eigenvalues were bounded by the theoretical upper and lower bounds (Fig. 7).

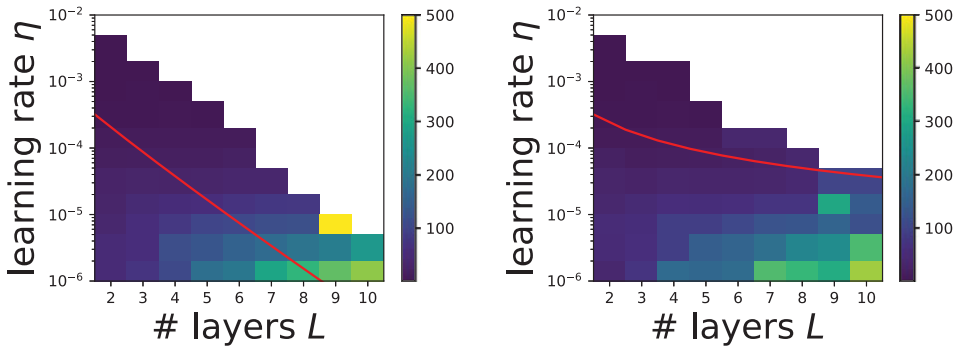


**Fig. 8.** Training loss under various settings. Read lines: theoretical lower-bounds of the maximum learning rates. White color: divergence (the loss >1000). (a) ResNet. (b) ResNet with BN.

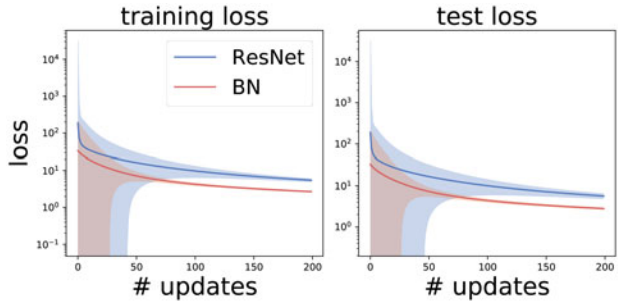


**Fig. 9.** Loss of the 4-layer DNNs (solid line: average, shadowed area: within one s.d.)

In addition, the convergence properties of the ResNet and ResNet with BN were numerically examined. Each algorithm with various numbers of layers $L$ and learning rates $\eta$ updated the parameters 50 times for each run and the training loss was averaged over five runs (Fig. 8). It was found that the algorithms converged if the learning rate was less than the lower bounds of the stable convergence, that is, $2/(\text{upper bound of } \lambda_{\max})$.

From a more practical viewpoint, we evaluated the training loss and test loss of the ResNet and ResNet with BN at each update, where each algorithm used an optimal learning rate, $\eta = 1/(\text{upper bound of } \lambda_{\max})$ (Fig. 9). The result shows that the BN accelerates the convergence and increases the stability.

## V. CONCLUSION

Some theoretical tools have been developed to analyze the theoretical properties of DNNs. We applied them to DNNs with new techniques such as skip connections and BN and showed why and how they improve the performance of DNNs. Skip connections reduce the generalization gap of standard DNNs, and the reductions are greater when the connections skip two layers at once, by smoothing the loss landscape around the minima. BN enables the GD to use a larger learning rate for convergence and accelerates training by smoothing the entire loss landscape. These analytical techniques may help researchers develop new DNN models.

## FINANCIAL SUPPORT

This work was supported in part by JSPS-KAKENHI grant numbers 18J15055 and 18K19821, and the NAIST Big Data Project initials. For example, “This work was supported by the Wellcome Trust (A.B., grant numbers XXXX, YYYY), (C.D., grant number ZZZZ); the Natural Environment Research Council (E.F., grant number FFFF); and the National Institutes of Health (A.B., grant number GGGG), (E.F., grant number HHHH)”. Where no specific funding has been provided for research, please provide the following statement: “This research received no specific grant from any funding agency, commercial or not-for-profit sectors.”

## STATEMENT OF INTEREST

None.

## REFERENCES

[1] Schmidhuber F.: Deep learning in neural networks: An overview. *Neural Netw.*, **61** (2015), 85–117.

[2] LeCun Y.; Bengio Y.; Hinton G.: Deep learning. *Nature*, **521** (2015), 436–444.

[3] Russakovsky O.; Deng J.H.; Su J.; Krause J.; Satheesh S.; Ma S.; Huang Z.; Karpathy A.; Khosla A.; Bernstein M.; Berg A.C.; Li F.F.: ImageNet large scale visual recognition challenge. *Int. J. Comput. Vis.*, **115** (2010), 211–252.

[4] Silver D. *et al.*: Mastering the game of go with deep neural networks and tree search. *Nature*, **550** (2016), 484–489.

[5] Bengio Y.: Learning deep architectures for AI. *Found. Trends® Mach. Learn.*, **2** (2009), 1–127.

[6] Montufar G.F.; Pascanu R.; Cho R.; Bengio Y.: On the number of linear regions of deep neural networks, On the number of linear regions of deep neural networks, 2014, 2924–2932.

[7] Telgarsky M.: Benefits of depth in neural networks, in *Conference on Learning Theory*, New York, PMLR 49 (2016), 1517–1539.

[8] Raghu M.; Poole B.; Kleinberg J.; Ganguli S.; Sohl-Dickstein J.: On the expressive power of deep neural networks, in *International Conference on Machine Learning*, Sydney, PMLR 70 (2017), 2847–2854.

[9] He K.; Zhang X.; Ren S.; Sun J.: Deep residual learning for image recognition, in *IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, 2016, 770–778.

[10] He K.; Zhang X.; Ren S.; Sun J.: Identity mappings in deep residual networks, in *European Conference on Computer Vision*, Amsterdam, 2016, 630–645.

[11] Ioffe S.; Szegedy C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift, in *International Conference on Machine Learning*, Lille, 2015, 448–456.

[12] Akaike H.: A new look at the statistical model identification. *IEEE Trans. Automat. Contr.*, **19** (1974), 716–723.

[13] Grunwald P.: Model selection based on minimum description length. *J. Math. Psychol.*, **44** (2000), 133–152.

[14] Bottou L.; Frank E.C.; Jorge N.: Optimization methods for large-scale machine learning. *SIAM Rev.*, **60** (2018), 223–311.

[15] Fukumizu K.; Akaho S.; Amari S.: Critical lines in symmetry of mixture models and its application to component splitting, in *Advances in Neural Information Processing Systems*, Vancouver, 2003, 865–872.

[16] Amari S.; Hyeyoung P.; Tomoko O.: Geometrical singularities in the neuromanifold of multilayer perceptrons, in *Advances in Neural Information Processing Systems*, Vancouver, 2002, 343–350.

[17] Kawaguchi K.: Deep learning without poor local minima, in *Advances in Neural Information Processing Systems*, Barcelona, 2016.

[18] Bousquet O.; Elisseeff A.: Stability and generalization. *J. Mach. Learn. Res.*, **2** (2002), 499–526.

[19] Charles Z.; Papailiopoulos D.: Stability and generalization of learning algorithms that converge to global optima, in *International Conference on Machine Learning*, Stockholm, PMLR 80 (2018), 745–754.

[20] Hardt M.; Ma T.: Identity matters in deep learning, in *International Conference on Learning Representations*, Toulon, 2017.

[21] LeCun Y.; Bottou L.; Orr G.B.; Muller K.R.: Efficient backprop, in *Neural Networks: Tricks or the Trade*, 2012, 9–48.

[22] Karakida R.; Akaho S.; Amari S.: Universal statistics of Fisher information in deep neural networks: Mean field approach, in *International Conference on Artificial Intelligence and Statistics*, Okinawa, 2019.

[23] Furusho Y.; Kubo T.; Ikeda K.: Roles of pre-training in deep neural networks from information theoretical perspective. *Neurocomputing*, **248** (2017), 76–79.

[24] Furusho Y.; Liu T.; Ikeda K.: Skipping two layers in ResNet makes the generalization gap smaller than skipping one or no layer, in *INNS Big Data and Deep Learning Conference*, Sestri Levante, 2019, 349–358.

[25] Furusho Y.; Ikeda K.: Effects of skip-connection in ResNet and batch-normalization on Fisher information matrix, in *INNS Big Data and Deep Learning Conference*, Sestri Levante, 2019, 341–348.

[26] Furusho Y.; Ikeda K.: Theoretical analysis of Fixup initialization for fast convergence and high generalization, in *International Conference on Machine Learning: Understanding and Improving Generalization in Deep Learning Workshop*, Long Beach, 2019, 23.

[27] Saxe A.M.; McClelland J.L.; Ganguli S.: Exact solutions to the nonlinear dynamics of learning in deep linear neural networks, in *International Conference on Learning Representation*, Banff, 2014.

[28] LeCun Y.; Bottou L.; Bengio Y.; Haffner P.: Gradient-based learning applied to document recognition. *Proc. IEEE*, **86** (1998), 2278–2324.

[29] Glorot X.; Bengio Y.: Understanding the difficulty of training deep feedforward neural networks, in *International Conference on Artificial Intelligence and Statistics*, PMLR 9 (2010), 249–256.

[30] Haykin S.: Adaptive Filter Theory, 4th ed., Prentice Hall, Upper Saddle River, NJ, 2002.

[31] Jacot A.; Gabriel F.; Hongler C.: Neural tangent kernel: convergence and generalization in neural networks, in *Advances in Neural Information Processing Systems*, Montreal, 2018, 8580–8589.

[32] Karakida R.; Akaho S.; Amari S.: The normalization method for alleviating pathological sharpness in wide neural networks, in *Advances in Neural Information Processing Systems*, Vancouver, 2019, in press.

[33] He K.; Zhang X.; Ren S.; Sun J.: Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification, in *IEEE International Conference on Computer Vision*, Santiago, 2015, 1026–1034.

[34] Poole B.; Lahiri S.; Sohl-Dickstein J.M.; Ganguli S.: Exponential expressivity in deep neural networks through transient chaos, in *Advances in Neural Information Processing Systems*, Barcelona, 2016.

[35] Yang G.; Schoenholz S.: Mean field residual networks: On the edge of chaos, in *Advances in Neural Information Processing Systems*, Long Beach, 2017, 2865–2873.

[36] Ge R.; Huang F.; Jin C.; Yuan Y.: Escaping from saddle points–online stochastic gradient for tensor decomposition, in *Conference on Learning Theory*, Paris, PMLR 40 (2015), 1–46.

[37] Lee J.; Jason D.; Scimochowitz M.; Jordan M.; Recht B.: Gradient descent only converges to minimizers, in *Conference on Learning Theory*, New York, PMLR 49 (2016), 1246–1257.

**Yasutaka Furusho** received his B.E. from National Institute of Technology, Kumamoto College, Japan, in 2015 and his M.E. from Graduate School of Information Science, Nara Institute of Science and Technology, Japan, in 2017. He is currently working toward his D.E. at Nara Institute of Science and Technology. His research interests include the analysis of neural networks based on statistical learning theory and information geometry, and its application to the model selection of neural networks.

**Kazushi Ikeda** received his B.E., M.E., and Ph.D in Mathematical Engineering and Information Physics from the University of Tokyo in 1989, 1991, and 1994, respectively. He was a research associate with the Department of Electrical and Computer Engineering of Kanazawa University from 1994 to 1998. He was a research associate of Chinese University of Hong Kong for three months in 1995. He was with Graduate School of Informatics, Kyoto University, as an associate professor from 1998 to 2008. Since 2008, he has been a full professor of Nara Institute of Science and Technology. He was the editor-in-chief of the Journal of the Japanese Neural Network Society and is currently an action editor of Neural Networks and an associate editor of IEEE Transactions on Neural Networks and Learning Systems.