TOWARDS UNDERSTANDING REGULARIZATION IN BATCH NORMALIZATION

Ping Luo[†*] **Xinjiang Wang**[‡*] **Wenqi Shao**[‡*] **Zhanglin Peng**[‡]
[†]The Chinese University of Hong Kong [‡]SenseTime Research

ABSTRACT

Batch Normalization (BN) improves both convergence and generalization in training neural networks. This work understands these phenomena theoretically. We analyze BN by using a basic block of neural networks, consisting of a kernel layer, a BN layer, and a nonlinear activation function. This basic network helps us understand the impacts of BN in three aspects. First, by viewing BN as an implicit regularizer, BN can be decomposed into population normalization (PN) and gamma decay as an explicit regularization. Second, learning dynamics of BN and the regularization show that training converged with large maximum and effective learning rate. Third, generalization of BN is explored by using statistical mechanics. Experiments demonstrate that BN in convolutional neural networks share the same traits of regularization as the above analyses.

1 INTRODUCTION

Batch Normalization (BN) is an indispensable component in many deep neural networks (He et al., 2016; Huang et al., 2017). Experimental studies (Ioffe & Szegedy, 2015) suggested that BN improves convergence and generalization by enabling large learning rate and preventing overfitting in training. Understanding BN theoretically is a key question.

Notations. This work denotes a scalar and a vector by using lowercase letter (*e.g.* x) and bold lowercase letter (*e.g.* $\mathbf{x}$) respectively. BN is investigated in a single-layer perceptron that is a building block of deep models, consisting of a kernel layer, a BN layer, and a nonlinear activation function. Its forward computation can be written by

$$y = g(\hat{h}), \;\; \hat{h} = \gamma\frac{h-\mu_{\mathcal{B}}}{\sigma_{\mathcal{B}}} + \beta \;\text{ and }\; h = \mathbf{w}^{\mathsf{T}}\mathbf{x}, \tag{1}$$

where $g(\cdot)$ denotes an activation function such as ReLU, h and $\hat{h}$ are the hidden values before and after normalization, $\mathbf{w}$ and $\mathbf{x}$ are kernel weight vector and network input respectively. In BN, $\mu_{\mathcal{B}}$ and $\sigma_{\mathcal{B}}$ represent mean and standard deviation of h. They are estimated within a batch of M samples. γ is a scale parameter and β is a shift parameter.

Despite the simplicity of the above basic network, it builds up the blocks of deep networks. It has been widely adopted in theoretical studies such as proper initialization (Krogh & Hertz, 1992; Advani & Saxe, 2017), dropout (Wager et al., 2013), weight decay and data augmentation (Bös, 1998). Our analyses assume that neurons at the BN layer are independent similar to (Salimans & Kingma, 2016; van Laarhoven, 2017), as the mean and the variance are estimated individually for each neuron. But we get rid of Gaussian assumption on the network input and the weight vector in theorem 1 that is our main result, meaning our assumption is milder than those in (Yoshida et al., 2017; Ba et al., 2016; Salimans & Kingma, 2016). Overall, several frequently-used notations are summarized in Table 2 in Appendix for reference.

1.1 HIGHLIGHTS OF RESULTS

Out main results are organized below in three aspects.

• First, Sec.2 decomposes BN into population normalization (PN) and gamma decay, which is an explicit regularization form of $\mu_{\mathcal{B}}$ and $\sigma_{\mathcal{B}}$. These statistics have different impacts: (1) $\mu_{\mathcal{B}}$ discourages reliance on a single neuron and encourages different neurons to have equal magnitude, in the sense that corrupting individual neuron does not harm generalization. This phenomenon was also found

[*]The first three authors contribute equally. Corresponding to pluo.lhi@gmail.com, {wangxinjiang, shaowenqi, pengzhanglin}@sensetime.com.

empirically in a recent work (Morcos et al., 2018), but has not been established analytically. (2) $\sigma_{\mathcal{B}}$ reduces kurtosis of the input distribution as well as correlations between neurons. (3) The regularization strengths of these statistics are *inversely proportional* to the batch size M, indicating that BN with large batch would decrease generalization. (4) Removing either one of $\mu_{\mathcal{B}}$ and $\sigma_{\mathcal{B}}$ could imped convergence and generalization.

• Second, by using ordinary differential equations (ODEs), Sec.3 shows that gamma decay enables the network trained with BN to converge with *large maximum and effective learning rate*, leading to faster training speed compared to the network trained without BN or trained with weight normalization (WN) (Salimans & Kingma, 2016) that is a counterpart of BN.

• Third, Sec.4 compares generalization errors of BN, WN, and vanilla SGD by using statistical mechanics. The "large-scale" regime is of interest, where number of samples P and number of neurons N are both large but their ratio P/N is finite. In this regime, the generalization errors are quantified both analytically and empirically.

Numerical results in Sec.5 show that BN in CNNs has the same traits of regularization as disclosed by the above analyses.

2 A Probabilistic Interpretation of BN

We treat $\mu_{\mathcal{B}}$ and $\sigma_{\mathcal{B}}$ as random variables to derive regularization of BN. Since one sample $\mathbf{x}$ is seen many times in training and at each time it is presented with other samples in a batch that is drawn randomly, $\mu_{\mathcal{B}}$ and $\sigma_{\mathcal{B}}$ can be treated as injected random noise for $\mathbf{x}$.

Loss Function. Training a neural network typically involves minimizing a negative log likelihood function with respect to a set of parameters $\theta = \{\mathbf{w}, \gamma, \beta\}$. Then the loss function can be defined by

$$\frac{1}{P}\sum_{j=1}^{P}\ell(\hat{h}^j) = -\frac{1}{P}\sum_{j=1}^{P}\log p(y^j|\hat{h}^j;\theta) + \zeta\|\mathbf{w}\|_2^2, \tag{2}$$

where $p(y^j|\hat{h}^j;\theta)$ represents the likelihood function of the network and P is number of training samples. As Gaussian distribution is often employed as prior distribution for the weight parameters, we have a weight decay $\zeta\|\mathbf{w}\|_2^2$ (Krizhevsky et al., 2012) that is a popular technique in deep learning.

Prior. By following (Teye et al., 2018), we find that BN also induces Gaussian priors for $\mu_{\mathcal{B}}$ and $\sigma_{\mathcal{B}}$. We have $\mu_{\mathcal{B}} \sim \mathcal{N}(\mu_{\mathcal{P}}, \frac{\sigma_{\mathcal{P}}^2}{M})$ and $\sigma_{\mathcal{B}} \sim \mathcal{N}(\sigma_{\mathcal{P}}, \frac{\rho+2}{4M})$, where M is batch size, $\mu_{\mathcal{P}}$ and $\sigma_{\mathcal{P}}$ are population mean and standard deviation respectively, and ρ is kurtosis that measures the peakedness of the distribution of h. These priors tell us that $\mu_{\mathcal{B}}$ and $\sigma_{\mathcal{B}}$ would produce Gaussian noise in training. There is a tradeoff regarding noise. For example, when M is small, training could diverge due to large noise. This is supported by experiment of BN (Wu & He, 2018) where training diverges when $M = 2$ in ImageNet (Russakovsky et al., 2015). When M is large, the noise is reduced because $\mu_{\mathcal{B}}$ and $\sigma_{\mathcal{B}}$ get close to $\mu_{\mathcal{P}}$ and $\sigma_{\mathcal{P}}$. It is known that $M > 30$ would provide a moderate noise, as the sample statistics converges in probability to the population statistics by the weak Law of Large Numbers. This is also supported by experiment (Ioffe & Szegedy, 2015) where BN with $M = 32$ already works well in ImageNet.

2.1 A Regularization Form

The loss function in Eqn.(2) can be written as an expected loss by integrating over the priors of $\mu_{\mathcal{B}}$ and $\sigma_{\mathcal{B}}$, that is, $\frac{1}{P}\sum_{j=1}^{P}\mathbb{E}_{\mu_{\mathcal{B}},\sigma_{\mathcal{B}}}[\ell(\hat{h}^j)]$ where $\mathbb{E}[\cdot]$ denotes expectation. We show that $\mu_{\mathcal{B}}$ and $\sigma_{\mathcal{B}}$ impose regularization on the scale parameter γ, but result in different regularization strengths. In theorem 1, we employ ReLU activation function as a concrete example that is widely used in practice. In general, the results can be extended to the other activation functions as shown in Appendix C.2.

Theorem 1 (Regularization of $\mu_{\mathcal{B}}, \sigma_{\mathcal{B}}$)**.** *Let $\ell(\hat{h})$ be the loss function of BN and the activation function be ReLU. Then*

$$\frac{1}{P}\sum_{j=1}^{P}\mathbb{E}_{\mu_{\mathcal{B}},\sigma_{\mathcal{B}}}\ell(\hat{h}^j) \simeq \frac{1}{P}\sum_{j=1}^{P}\ell(\bar{h}^j) + \zeta(h)\gamma^2 \quad \text{and} \quad \zeta(h) = \underbrace{\frac{\rho+2}{8M}F_\gamma}_{\text{from } \sigma_{\mathcal{B}}} + \underbrace{\frac{1}{2M}\frac{1}{P}\sum_{j=1}^{P}\sigma(\bar{h}^j)}_{\text{from } \mu_{\mathcal{B}}}, \tag{3}$$

where $\bar{h}^j = \gamma\frac{h^j-\mu_{\mathcal{P}}}{\sigma_{\mathcal{P}}} + \beta$ represents the population normalization (PN) and $h^j = \mathbf{w}^{\mathsf{T}}\mathbf{x}^j$. $\zeta(h)$ is a data-dependent coefficient of gamma decay, ρ is the kurtosis of distribution of h, F_γ represents Fisher Information Matrix (FIM) of γ, and $\sigma(\cdot)$ is a sigmoid function.

From theorem 1, we have several observations that are of both theoretical and practical values.

• First, it decomposes BN into population normalization (PN) and gamma decay. PN replaces the batch statistics in BN by population statistics. In gamma decay, computation of $\zeta(h)$ is data-dependent, making it differed from weight decay where the coefficient is determined empirically. In essentials, Eqn.(3) represents the randomness in BN in a deterministic way, not only enabling us to apply methodologies such as ODEs and statistical mechanics to analyze BN, but also inspiring us to imitate BN's performance by WN without computing batch statistics in numerical study.

• Second, PN is closely connected to WN that is independent from sample mean and variance. WN (Salimans & Kingma, 2016) is defined by $\upsilon\frac{\mathbf{w}^T\mathbf{x}}{||\mathbf{w}||_2}$ that normalizes the weight vector to have unit variance, where υ is a learnable parameter. Let each diagonal element of the covariance matrix of $\mathbf{x}$ be a and all the off-diagonal elements be zeros. $\bar{h}^j$ can be rewritten as

$$\bar{h}^j = \gamma\frac{\mathbf{w}^T\mathbf{x}^j - \mu_{\mathcal{P}}}{\sigma_{\mathcal{P}}} + \beta = \upsilon\frac{\mathbf{w}^T\mathbf{x}^j}{||\mathbf{w}||_2} + b, \tag{4}$$

where $\upsilon = \frac{\gamma}{a}$ and $b = -\frac{\gamma\mu_{\mathcal{P}}}{a||\mathbf{w}||_2} + \beta$. Eqn.(4) removes the estimations of statistics and eases our analyses of regularization for BN.

• Third, $\mu_{\mathcal{B}}$ and $\sigma_{\mathcal{B}}$ produce different parts in $\zeta(h)$. The strength from $\mu_{\mathcal{B}}$ depends on the expectation of $\sigma(\bar{h}^j) \in [0,1]$, which represents excitation or inhibition of a neuron, meaning that a neuron with larger output may exposure to larger regularization, encouraging different neurons to have equal magnitude. This is consistent with empirical result (Morcos et al., 2018) which prevented reliance on single neuron to improve generalization. The strength from $\sigma_{\mathcal{B}}$ works as a complement for $\mu_{\mathcal{B}}$. For a single neuron, F_γ represents the norm of gradient, implying that BN punishes large gradient norm. For multiple neurons, F_γ is the FIM of γ, meaning that BN would penalize correlations among neurons. Both $\sigma_{\mathcal{B}}$ and $\mu_{\mathcal{B}}$ are important, removing either one of them would imped performance.

We observe in experiments in Sec.5 that BN in CNNs share similar traits of regularization. However, in deep models the priors for $\sigma_{\mathcal{B}}$ and $\mu_{\mathcal{B}}$ become multivariate Gaussian distributions where relationships between layers may not be neglected. In this case, we didn't find meaningful analytical form for BN.

3 Optimization with Regularization

We show that BN converges with large maximum and effective learning rate (lr) that are larger than a network trained without BN. Our result explains why large lr can be used in practice in BN (Ioffe & Szegedy, 2015). Our analyses are conducted in three stages. First, we establish dynamical equations of a teacher-student (T-S) model in thermodynamic limit and acquire the fixed point. Second, we investigate eigenvalues of the corresponding Jacobian matrix at this fixed point. Finally, we calculate the maximum and the effective lr.

Teacher-Student Model. We first introduce useful techniques from statistical mechanics (SM). In SM, a student network is dedicated to learn relationship between an input and an output with a weight vector $\mathbf{w}$ as parameters. It is useful to characterize behavior of the student by using a teacher network that uses $\mathbf{w}^*$ as a ground-truth vector. We treat the single-layer perceptron as the student, which is optimized by minimizing the euclidian distance between its output and the supervision provided by a teacher without BN. The student and the teacher have identical activation function.

Loss Function. We define a loss function of the above T-S model by $\frac{1}{P}\sum_{j=1}^{P}\ell(\mathbf{x}^j) = \frac{1}{P}\sum_{j=1}^{P}\big[g(\mathbf{w}^{*\mathsf{T}}\mathbf{x}^j) - g(\sqrt{N}\gamma\frac{\mathbf{w}^{\mathsf{T}}\mathbf{x}^j}{\|\mathbf{w}\|_2})\big]^2 + \zeta\gamma^2$. Here $g(\mathbf{w}^{*\mathsf{T}}\mathbf{x}^j)$ represents supervision from the teacher, while $g(\sqrt{N}\gamma\frac{\mathbf{w}^{\mathsf{T}}\mathbf{x}^j}{\|\mathbf{w}\|_2})$ is the output of student trained to mimic its teacher. The student is defined by Eqn.(4) where $\nu = \sqrt{N}\gamma$ and the bias term is merged into $\mathbf{w}$. This loss function represents BN using WN with gamma decay and it is sufficient to study the lr of different approaches. Let $\theta = \{\mathbf{w}, \gamma\}$ be a set of parameters updated by SGD, *i.e.* $\theta^{j+1} = \theta^j - \eta\frac{\partial\ell(\mathbf{x}^j)}{\partial\theta^j}$ where η denotes a learning rate. The update rules for $\mathbf{w}$ and γ are

$$\mathbf{w}^{j+1} - \mathbf{w}^j = \eta\delta^j(\frac{\gamma^j\sqrt{N}}{\|\mathbf{w}^j\|_2}\mathbf{x}^j - \frac{\tilde{\mathbf{w}}^{j\mathsf{T}}\mathbf{x}^j}{\|\mathbf{w}^j\|_2^2}\mathbf{w}^j) \text{ and } \gamma^{j+1} - \gamma^j = \eta(\frac{\delta^j\sqrt{N}\mathbf{w}^{j\mathsf{T}}\mathbf{x}^j}{\|\mathbf{w}^j\|_2} - \zeta\gamma^j), \tag{5}$$

where $\tilde{\mathbf{w}}$ denotes a normalized weight vector of the student, that is, $\tilde{\mathbf{w}} = \sqrt{N}\gamma\frac{\mathbf{w}}{\|\mathbf{w}\|_2}$, and $\delta^j = g'(\tilde{\mathbf{w}}^{j\mathsf{T}}\mathbf{x}^j)[g(\mathbf{w}^{*\mathsf{T}}\mathbf{x}^j) - g(\tilde{\mathbf{w}}^{j\mathsf{T}}\mathbf{x}^j)]$ represents the gradient[1] for clarity of notation.

Order Parameters. As we are interested in the "large-scale" regime where both N and P are large and their ratio P/N is finite, it is difficult to examine a student with parameters in high dimensions directly. Therefore, we transform the weight vectors to order parameters that fully characterize interactions between the student and the teacher. In this case, the parameter vector can be reparameterized by using a vector of three elements including γ, R, and L. In particular, γ measures norm of the normalized weight vector $\tilde{\mathbf{w}}$, that is, $\tilde{\mathbf{w}}^{\mathsf{T}}\tilde{\mathbf{w}} = N\gamma^2\frac{\mathbf{w}^{\mathsf{T}}\mathbf{w}}{\|\mathbf{w}\|_2^2} = N\gamma^2$. The parameter R measures angle (overlapping ratio) between the weight vectors of student and teacher. We have $R = \frac{\tilde{\mathbf{w}}^{\mathsf{T}}\mathbf{w}^*}{\|\tilde{\mathbf{w}}\|\|\mathbf{w}^*\|} = \frac{1}{N\gamma}\tilde{\mathbf{w}}^{\mathsf{T}}\mathbf{w}^*$ where the norm of the ground-truth vector is $\frac{1}{N}\mathbf{w}^{*\mathsf{T}}\mathbf{w}^* = 1$. Moreover, L represents norm of the original weight vector $\mathbf{w}$ and $L^2 = \frac{1}{N}\mathbf{w}^{\mathsf{T}}\mathbf{w}$. With the above definitions, relationship between R and L can be represented by $RL = \frac{1}{N}\mathbf{w}^{\mathsf{T}}\mathbf{w}^*$.

3.1 Learning Dynamics of Order Parameters

Now we transform update equations (5) by using order parameters. To this end, we define three variables γ^2, L^2 and RL. The update rule for variable γ^2 can be obtained by $\left(\gamma^2\right)^{j+1} - \left(\gamma^2\right)^j = \frac{1}{N}\left[2\eta\delta^j\tilde{\mathbf{w}}^{j\mathsf{T}}\mathbf{x}^j - 2\eta\zeta\left(\gamma^2\right)^j\right]$ following update rule of γ. Similarly, the update rules for variables L^2 and RL are $\left(RL\right)^{j+1} - \left(RL\right)^j = \frac{1}{N}\left(\frac{\eta\gamma^j}{L^j}\delta^j\mathbf{w}^{*\mathsf{T}}\mathbf{x}^j - \frac{\eta R^j}{L^j}\delta^j\tilde{\mathbf{w}}^{j\mathsf{T}}\mathbf{x}^j\right)$ and $\left(L^2\right)^{j+1} - \left(L^2\right)^j = \frac{1}{N}\left[\frac{\eta^2(\gamma^2)^j}{(L^2)^j}{\delta^j}^2\mathbf{x}^{j\mathsf{T}}\mathbf{x}^j - \frac{\eta^2}{N(L^2)^j}{\delta^j}^2(\tilde{\mathbf{w}}^{j\mathsf{T}}\mathbf{x}^j)^2\right]$ by multiplying both sides of (5) by $\mathbf{w}^*$.

To define the learning dynamics, we turn the above update rules into ODEs. We take γ^2 as an example. Its differential equation can be defined by $\frac{d\gamma^2}{dt} = \lim_{\Delta t\to 0}\frac{(\gamma^2)^{j+1} - (\gamma^2)^j}{\Delta t} = 2\eta\langle\delta\tilde{\mathbf{w}}^{\mathsf{T}}\mathbf{x}\rangle_{\mathbf{x}} - 2\eta\zeta\gamma^2$, where $t = \frac{j}{N}$ is a normalized sample index that can be treated as a continuous time variable. We have $\Delta t = \frac{1}{N}$ that approaches zero in the thermodynamic limit when $N \to \infty$. $\langle\cdot\rangle_{\mathbf{x}}$ denotes expectation over the distribution of $\mathbf{x}$. The differential equations of $\frac{dRL}{dt}$ and $\frac{dL^2}{dt}$ can be defined in the same way. We simplify notations by representing $I_1 = \langle\delta\tilde{\mathbf{w}}^{\mathsf{T}}\mathbf{x}\rangle_{\mathbf{x}}$, $I_2 = \langle\delta^2\mathbf{x}^{\mathsf{T}}\mathbf{x}\rangle_{\mathbf{x}}$ and $I_3 = \langle\delta\mathbf{w}^{*\mathsf{T}}\mathbf{x}\rangle_{\mathbf{x}}$. We obtain a dynamical system

$$\frac{d\gamma}{dt} = \eta\frac{I_1}{\gamma} - \eta\zeta\gamma, \quad \frac{dR}{dt} = \eta\frac{\gamma}{L^2}I_3 - \eta\frac{R}{L^2}I_1 - \eta^2\frac{\gamma^2 R}{2L^4}I_2 \quad \text{and} \quad \frac{dL}{dt} = \eta^2\frac{\gamma^2}{2L^3}I_2. \tag{6}$$

More results are provided in Appendix C.4.

3.2 Fixed Point of the Dynamical System

To investigate lr of BN, we derive the fixed point of (6) by setting $d\gamma/dt = dR/dt = dL/dt = 0$. The fixed points of BN, WN, and vanilla SGD (without BN and WN) are given in Table 1. In the thermodynamic limit, the optima for (γ_0, R_0, L_0) would be $(1, 1, 1)$. Our main interest is the overlapping ratio R_0 between the student and the teacher, because it optimizes the direction of the weight vector. We see that R_0 for all three approaches attain optimum '1'. Intuitively, in BN and WN, this optimal solution does not depend on L_0 because their weight vectors are normalized. In other words, WN and BN are easier to optimize than vanilla SGD where both R_0 and L_0 have to be optimized. In BN, γ_0 depends on the activation function. For ReLU, we have $\gamma_0^{bn} = \frac{1}{2\zeta+1}$, meaning that norm of the normalized weight vector relies on the decay factor. In WN, we have $\gamma_0^{wn} = 1$ as WN has no regularization on γ.

	(γ_0, R_0, L_0)	$\eta_{\max}\ (R)$	$\eta_{\text{eff}}\ (R)$
BN	$(\gamma_0, 1, L_0)$	$\left(\frac{\partial(\gamma_0 I_3 - I_1)}{\gamma_0\partial R} - \zeta\gamma_0\right) / \frac{\partial I_2}{2\partial R}$	$\frac{\eta\gamma_0}{L_0^2}$
WN	$(1, 1, L_0)$	$\frac{\partial(I_3 - I_1)}{\partial R} / \frac{\partial I_2}{2\partial R}$	$\frac{\eta}{L_0^2}$
SGD	$(1, 1, 1)$	$\frac{\partial(I_3 - I_1)}{\partial R} / \frac{\partial I_2}{2\partial R}$	η

Table 1: Comparisons of fixed points, $\eta_{\max}$ for R, and η_{eff} for R. A fixed point is denoted as (γ_0, R_0, L_0).

3.3 Maximum and Effective Learning Rates

With the above fixed points, we derive the maximum and the effective lr. Specifically, we analyze eigenvalues and eigenvectors of the Jacobian matrix corresponding to (6). We are interested in the

[1] $g'(x)$ denotes the first derivative of $g(x)$.

lr to approach R_0. We find that this optimum value only depends on its eigenvalue denoted as λ_R, $\lambda_R = \frac{\partial I_2}{\partial R}\frac{\eta\gamma_0}{2L_0^2}(\eta_{\max} - \eta_{\mathrm{eff}})$, where $\eta_{\max}$ and η_{eff} are maximum and effective lr (proposition 1 in Appendix C.4). They are given in Table 1. We demonstrate that $\lambda_R < 0$ if and only if $\eta_{\max} > \eta_{\mathrm{eff}}$, such that the fixed point R_0 is stable for all approaches (proposition 2 in Appendix C.5). It is able to show that $\eta_{\max}$ of BN ($\eta^{bn}_{\max}$) is larger than WN and SGD, enabling R to converge with a larger learning rate. With ReLU, we find that $\eta^{bn}_{\max} \geq \eta^{\{wn,sgd\}}_{\max} + 2\zeta$ (proposition 3 in Appendix C.6). Moreover, the effective lr's in Table 1 are consistent with previous work (van Laarhoven, 2017).

4 Generalization Analysis

To investigate generalization of BN, we adopt a teacher-student model with identity activation function, which minimizes a loss function $\frac{1}{P}\sum_{j=1}^{P}(\bar{y}^j - y^j)^2$, where $\bar{y}$ represents the teacher's output and y is the student's output. We compare BN with WN+gamma decay and vanilla SGD. All of them share the same teacher network whose output is defined by $\bar{y} = \mathbf{w}^{*\mathsf{T}}\mathbf{x} + \epsilon$, where $\mathbf{x}$ is drawn from $\mathcal{N}(0, \frac{1}{N})$ and ϵ is an unobserved Gaussian noise. We are interested to see how different methods resist this noise.

For **vanilla SGD**, the student is computed by $y = \mathbf{w}^{\mathsf{T}}\mathbf{x}$ with $\mathbf{w}$ being the weight vector to optimize, where $\mathbf{w}$ has the same dimension as $\mathbf{w}^*$ to be a realizable learning problem. The loss function of vanilla SGD is $\ell^{sgd} = \frac{1}{P}\sum_{j=1}^{P}(\mathbf{w}^{*\mathsf{T}}\mathbf{x}^j - \mathbf{w}^{\mathsf{T}}\mathbf{x}^j)^2$, whose solution asymptotically approaches the Moore–Penrose pseudo inverse solution $\mathbf{w} = \left(\mathbf{x}^{\mathsf{T}}\mathbf{x}\right)^{+}\mathbf{x}^{\mathsf{T}}\bar{\mathbf{y}}$. For **BN**, the student is defined as $y = \gamma\frac{\mathbf{w}^{\mathsf{T}}\mathbf{x} - \mu_{\mathcal{B}}}{\sigma_{\mathcal{B}}} + \beta$. As our main interest is the weight vector, we freeze the bias similar to vanilla SGD by setting $\beta = 0$. Therefore, the loss function is written as $\ell^{bn} = \frac{1}{P}\sum_{j=1}^{P}\left(\mathbf{w}^{*\mathsf{T}}\mathbf{x}^j - \gamma(\mathbf{w}^{\mathsf{T}}\mathbf{x}^j - \mu_{\mathcal{B}})/\sigma_{\mathcal{B}}\right)^2$. For **WN+gamma decay**, the student is computed similar to Eqn.(4) by $y = \sqrt{N}\gamma\frac{\mathbf{w}^{\mathsf{T}}\mathbf{x}}{\|\mathbf{w}\|_2}$. Then the loss function is defined by $\ell^{wn} = \frac{1}{P}\sum_{j=1}^{P}\left(\mathbf{w}^{*\mathsf{T}}\mathbf{x}^j - \sqrt{N}\gamma\frac{\mathbf{w}^{\mathsf{T}}\mathbf{x}^j}{\|\mathbf{w}\|_2}\right)^2 + \zeta\|\gamma\|_2^2$. In the T-S model with identity unit, expression of ζ becomes $\zeta = \frac{1}{2M}$ after applying theorem 1 (Appendix C.1). With the above definitions, the three approaches are studied under the same T-S model, where their generalization errors can be strictly compared with the other factors ruled out.

4.1 Generalization Errors

We provide closed-form solutions of the generalization errors for vanilla SGD and WN+gamma decay. They are compared with numerical solutions of BN.

vanilla SGD. The solution of generalization error depends on the rank of correlation matrix $\mathbf{\Sigma} = \mathbf{x}^{\mathsf{T}}\mathbf{x}$. Here we define an effective load $\alpha = P/N$ that is the ratio between number of samples P and number of input neurons N (number of learnable parameters). The generalization error denoted as $\epsilon^{sgd}_{\mathrm{gen}}$ can be acquired by using the distribution of eigenvalues of $\mathbf{\Sigma}$ following (Advani & Saxe, 2017). If $\alpha < 1$, $\epsilon^{sgd}_{\mathrm{gen}} = 1 - \alpha + \alpha\epsilon^2/(1-\alpha)$. Otherwise, $\epsilon^{sgd}_{\mathrm{gen}} = \epsilon^2/(1-\alpha)$ where ϵ is the injected noise to the teacher. The values of $\epsilon^{sgd}_{\mathrm{gen}}$ with respect to α are plotted in blue curve in the top of Fig.1. It first decreases but then increases as α increases from 0 to 1, $\epsilon^{sgd}_{\mathrm{gen}}$ diverges at $\alpha = 1$, and it would decrease again when $\alpha > 1$.

WN+gamma decay. The decay term turns the correlation matrix to $\mathbf{\Sigma} = \left(\mathbf{x}^{\mathsf{T}}\mathbf{x} + \zeta\mathbf{I}\right)$ that is positive definite. Following statistical mechanics (Krogh & Hertz, 1992), the generalization error is $\epsilon^{wn}_{\mathrm{gen}} = \delta^2\frac{\partial(\zeta G)}{\partial\zeta} - \zeta^2\frac{\partial G}{\partial\zeta}$ where $G = 1 - \alpha - \zeta + \left(\zeta + (1+\sqrt{\alpha})^2\right)^{\frac{1}{2}}\left(\zeta + (1-\sqrt{\alpha})^2\right)^{\frac{1}{2}}/2\zeta$. We see that $\epsilon^{wn}_{\mathrm{gen}}$ can be computed quantitatively given the values of ζ and α. Let the variance of noise ϵ injected to the teacher be 0.25. Fig.1 top shows that no other curves could outperform the curve when $\zeta = 0.25$, a value

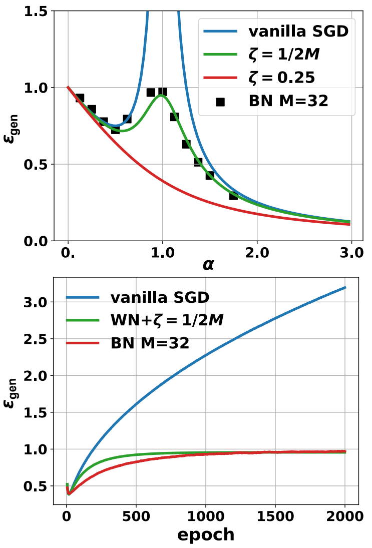


Figure 1: **Top:** generalization error ϵ_{gen} *v.s.* effective load α. 'WN+gamma decay' has two cases $\zeta = \frac{1}{2M}$ and $\zeta = 0.25$. BN has $M = 32$. **Bottom:** generalization error at $\alpha = 1$.

equal to the noise magnitude. The ζ smaller than 0.25 would exhibit overtraining around $\alpha = 1$, but they still perform significantly better than vanilla SGD.

Numerical Solutions of BN. We employ SGD with $M = 32$ to find solutions of $\mathbf{w}$ for BN. The generalization error is evaluated as difference between the validation and the training loss. The number of input neurons is 4096 and the number of training samples can be varied to change α. The results are marked as black squares in the top of Fig.1. After applying theorem 1 to the T-S model, BN is equivalent to WN+gamma decay when $\zeta = \frac{1}{2M}$. It is seen that BN gets in line with the curve of ‘ζ=1/2M’ ($M = 32$) and thus quantitatively validates our derivations. Their generalization errors are further compared in the bottom of Fig.1 at $\alpha = 1$, where vanilla SGD clearly diverges, while BN and WN+gamma decay are comparable.

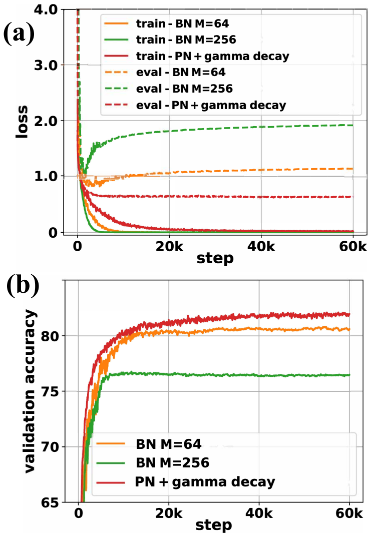


Figure 2: Training and evaluation loss in (a) and validation accuracy in (b).

5 Experiments in CNNs

This section shows that BN in CNNs follows similar traits of regularization as our analyses. We employ a 6-layered CNN similar to (Salimans & Kingma, 2016). For all experiments, the network architecture is fixed while the normalization layers can be changed. We adopt CIFAR10 (Krizhevsky, 2009) that contains 60k images of 10 categories (50k images for training and 10k images for test). All models are trained by using SGD with momentum on a single GPU, while the initial learning rates are scaled proportionally for different batch sizes (Goyal et al., 2017). In order to study regularization of BN, we discard any other trick such as weight decay and data augmentation. More empirical setting can be found in Appendix B.

Evaluation of Theorem 1. We compare BN with PN+gamma decay where the population statistics and the regularization coefficient are estimated by using sufficient amount of training samples. BN trained with a normal batch size $M = 64$ is treated as baseline in Fig.2. When batch size increases, BN would imped both loss and accuracy. For example, when increasing M to 256, performance decreases because the regularization from the batch statistics reduces in large batch, resulting in overtraining (see the gap between train and validation loss when $M = 256$).

In comparison, we train PN by using 10k training samples to estimate statistics. This further reduces regularization. We see that the release of regularization can be complemented by gamma decay, making PN even outperformed BN. This empirical result verifies our derivation of regularization for BN. Similar trend can be observed by experiment in a downsampled version of ImageNet (see Appendix B.1). Nevertheless, we would like to point out that PN+gamma decay is of interest in theoretical analysis, but it is computation-demanding when applied in practice because evaluating $\mu_{\mathcal{P}}$, $\sigma_{\mathcal{P}}$ and $\zeta(h)$ may require sufficiently large number of samples.

Study of Regularization. We study the regulation strengthes of vanilla SGD, BN, WN, WN+mean-only BN, and WN+variance-only BN. Fig.3 compares their training and validation losses. We see that the generalization error of BN is much lower than WN and vanilla SGD. The reason has been disclosed in this work: stochastic behaviors of $\mu_{\mathcal{B}}$ and $\sigma_{\mathcal{B}}$ in BN improves generalization.

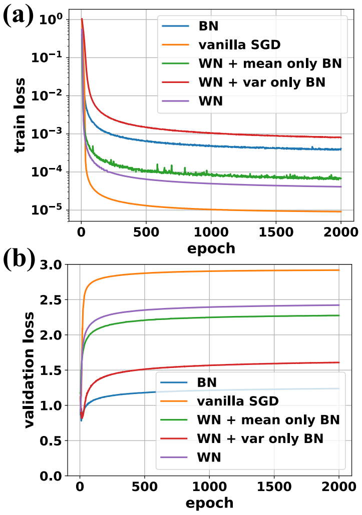


Figure 3: Study of regularization.

To investigate $\mu_{\mathcal{B}}$ and $\sigma_{\mathcal{B}}$ individually, we decompose their contributions by running a WN with mean-only BN as well as a WN with variance-only BN, to simulate their respective regularization. As shown in Fig.3, improvements from the mean-only and the variance-only BN over WN verify our conclusion that noises from $\mu_{\mathcal{B}}$ and $\sigma_{\mathcal{B}}$ have different regularization strengths. Both of them are essential to produce good result.

Parameter Norm. We further demonstrate impact of BN to the norm of parameters. We compare BN with vanilla SGD. A network is first trained by BN in order to converge to a local minima where the parameters do not change much. At this local minima, the weight vector is frozen and denoted as $\mathbf{w}^{bn}$. Then this network is finetuned by using vanilla SGD with a small learning rate 10^{-3} and its kernel parameters are initialized by $\mathbf{w}^{sgd} = \gamma \frac{\mathbf{w}^{bn}}{\sigma}$, where σ is the moving average of $\sigma_{\mathcal{B}}$.

Fig.7 in Appendix B.2 visualizes the results. As $\mu_{\mathcal{B}}$ and $\sigma_{\mathcal{B}}$ are removed in vanilla SGD, it is found from the last two figures that the training loss decreases while the validation loss increases, implying that reduction in regularization makes the network converged to a sharper local minimum that generalizes less well. The magnitudes of kernel parameters $\mathbf{w}^{sgd}$ at different layers are also displayed in the first four figures. All of them increase after freezing BN, due to the release of regularization on these parameters.

Batch size. To study BN with different batch sizes, we train different networks but only add BN at one layer at a time. The regularization on the γ parameter is compared in Fig.4 (a) when BN is located at different layers. The values of γ^2 increase along with the batch size M due to the weaker regularization for the larger batches. The increase of γ^2 also makes all validation losses increased as shown in Fig.4 (b).

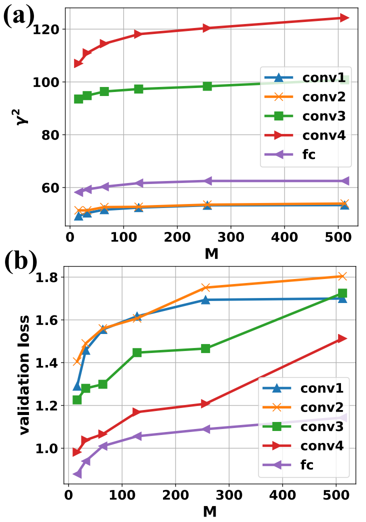


Figure 4: Values of γ^2 increase along with M as shown in (a), due to the lack of regularization in large batch, making the validation losses increased as well in (b).

BN+dropout. Despite the better generalization of BN with smaller batch sizes, large-batch training is more efficient in real cases. Therefore, improving generalization of BN with large batch is more desiring. However, gamma decay requires estimating the population statistics that increases computations. We also found that treating the decay coefficient as a constant hardly improves generalization for large batch. Therefore, we utilize dropout as an alternative to compensate for the insufficient regularization. Dropout has also been analytically viewed as a regularizer (Wager et al., 2013). We add a dropout after each BN layer to impose regularization.

Fig.5 plots the results. The generalization of BN deteriorates significantly when M increases from 64 to 256. This is observed by the much higher validation loss (top) and lower validation accuracy (bottom) when $M = 256$. If a dropout layer with ratio 0.125 is added *after* each BN layer for $M = 256$, the validation loss is suppressed and accuracy increased by a great margin. This superficially contradicts with the original claim that BN reduces the need for dropout (Ioffe & Szegedy, 2015). There are two differences between our study and previous work.

First, in pervious study the batch size was fixed at a quite small value (*e.g.* 32), at which the regularization was already quite strong. Therefore, an additional dropout could not further cause better regularization, but on the contrary increases the instability in training and yields a lower accuracy. However, our study explores relatively large batch that degrades the regularization of BN, and thus dropout with a small ratio can complement. Second, usual trials put dropout before BN and cause BN to have different variances during training and test. In contrast, dropout follows BN in this study and thus the problem can be alleviated. The improvement by applying dropout after BN has also been observed by a recent work (Li et al., 2018).

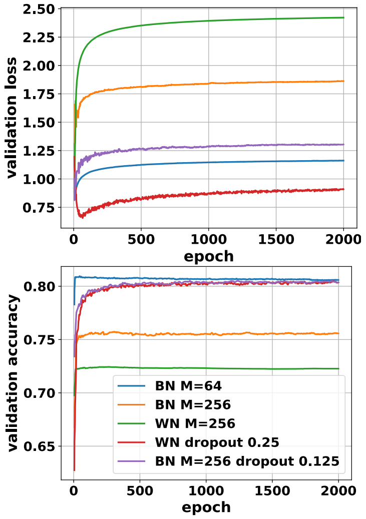


Figure 5: BN and WN with dropout.

WN+dropout. Since BN can be treated as WN trained with regularization in this study, combining WN with regularization should be able to match the performance of BN. As WN outperforms BN in running speed (without calculating statistics) and it suits better in RNNs than BN, an improvement of its generalization is also of great importance. Fig.5 shows that WN can also be regularized by dropout. We apply dropout after each WN layer with ratio 0.25. We found that the improvement on both validation accuracy and loss is surprising. The accuracy increases from 0.73 to 0.80, surpassing ‘BN M=256’ and on par with ‘BN M=64’.

6 Related Work

Neural Network Analysis. Many studies analysed neural networks (Opper et al., 1990; Saad & Solla, 1996; Bs & Opper, 1998; Pennington & Bahri, 2017; Zhang et al., 2017b; Brutzkus & Globerson, 2017; Raghu et al., 2017; Mei et al., 2016; Tian, 2017). For example, for a multilayer network with linear activation function, Glorot & Bengio (2010) explored its SGD dynamics and Kawaguchi (2016) showed that every local minimum is global. Tian (2017) studied the critical points and convergence behaviors of a 2-layered network with ReLU units. Zhang et al. (2017b) investigated a teacher-student model when the activation function is harmonic. In (Saad & Solla, 1996), the learning dynamics of a committee machine were discussed when the activation function is error function $\mathrm{erf}(x)$. Unlike previous work, this work analyzes regularization emerged in BN and its impact to both learning and generalization, which are still unseen in the literature.

Normalization. Many normalization methods have been proposed recently. For example, BN (Ioffe & Szegedy, 2015) was introduced to stabilize the distribution of input data of each hidden layer. Weight normalization (WN) (Salimans & Kingma, 2016) decouples the lengths of the network parameter vectors from their directions, by normalizing the parameter vectors to unit length. The dynamic of WN was studied by using a single-layer network (Yoshida et al., 2017). Moreover, Li et al. (2018) diagnosed the compatibility of BN and dropout (Srivastava et al., 2014) by reducing the variance shift produced by them. van Laarhoven (2017) showed that weight decay has no regularization effect when using together with BN or WN. Ba et al. (2016) demonstrated when BN or WN is employed, back-propagating gradients through a hidden layer is scale-invariant with respect to the network parameters. Santurkar et al. (2018) gave another perspective of the role of BN during training instead of reducing the covariant shift. They argued that BN results in a smoother optimization landscape and the Lipschitzness is strengthened in networks trained with BN. However, both analytical and empirical results of regularization in BN are still desirable. Our study explores regularization, optimization, and generalization of BN in the scenario of online learning.

Regularization. Ioffe & Szegedy (2015) conjectured that BN implicitly regularizes training to prevent overfitting. Zhang et al. (2017a) categorized BN as an implicit regularizer from experimental results. Szegedy et al. (2015) also conjectured that in the Inception network, BN behaves similar to dropout to improve the generalization ability. Gitman & Ginsburg (2017) experimentally compared BN and WN, and also confirmed the better generalization of BN. In the literature there are also implicit regularization schemes other than BN. For instance, random noise in the input layer for data augmentation has long been discovered equivalent to a weight decay method, in the sense that the inverse of the signal-to-noise ratio acts as the decay factor (Krogh & Hertz, 1992; Rifai et al., 2011). Dropout (Srivastava et al., 2014) was also proved able to regularize training by using the generalized linear model (Wager et al., 2013).

7 Discussions and Future Work

This work investigated regularization emerged in BN. By utilizing a single-layer perceptron, BN was decomposed into PN and gamma decay, where the regularization strengths from $\mu_\mathcal{B}$ and $\sigma_\mathcal{B}$ are different and their impacts in training were explored. Moreover, convergence and generalization of BN with regularization were derived and compared with vanilla SGD, WN, and WN+gamma decay, showing that BN enables training to converge with large maximum and effective learning rate, as well as leads to better generalization. Our analytical results explain many existing empirical phenomena. Experiments in CNNs showed that BN in deep networks share the same traits of regularization. Furthermore, a combination of dropout and BN might ameliorate BN when batch size goes large. Our result also encourages us to combine WN and dropout, outperforming BN in some senses without estimating batch statistics.

In future work, we are interested in finding analytical form of regularization for BN in deep networks, although it might involve multivariate Gaussian prior distributions, making it a non-trivial problem. Moreover, investigating the other normalizers such as instance normalization (IN) (Ulyanov et al., 2016) and layer normalization (LN) (Ba et al., 2016) is also important. Understanding the characteristics of these normalizers should be the first step to analyze some recent best practices such as group normalization (Wu & He, 2018) that merged IN and LN, and switchable normalization (Luo et al., 2018) that chose BN, IN, and LN in each normalization layer. Furthermore, devising an efficient counterpart of gamma decay is desirable in the community and will be investigated in the future, as it may improve generalization of WN that is independent of batch statistics.

REFERENCES

Madhu S. Advani and Andrew M. Saxe. High-dimensional dynamics of generalization error in neural networks. *arXiv:1710.03667 [physics, q-bio, stat]*, October 2017. URL `http://arxiv.org/abs/1710.03667`. arXiv: 1710.03667.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv:1607.06450*, 2016.

Alon Brutzkus and Amir Globerson. Globally optimal gradient descent for a convnet with gaussian inputs. In *ICML*, 2017.

Siegfried Bös. Statistical mechanics approach to early stopping and weight decay. *Physical Review E*, 58(1):833, 1998.

Siegfried Bs and Manfred Opper. Dynamics of batch training in a perceptron. In *Journal of Physics A: Mathematical and General*, volume 31(21), pp. 4835, 1998.

Igor Gitman and Boris Ginsburg. Comparison of Batch Normalization and Weight Normalization Algorithms for the Large-scale Image Classification. *arXiv:1709.08145 [cs]*, September 2017. URL `http://arxiv.org/abs/1709.08145`. arXiv: 1709.08145.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour. *arXiv preprint arXiv:1706.02677*, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

Kenji Kawaguchi. Deep learning without poor local minima. In *NIPS*, 2016.

Alex Krizhevsky. Learning multiple layers of features from tiny images. In *Technical Report*, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

Anders Krogh and John A. Hertz. Generalization in a linear perceptron in the presence of noise. *Journal of Physics A: Mathematical and General*, 25(5):1135, 1992.

Xiang Li, Shuo Chen, Xiaolin Hu, and Jian Yang. Understanding the disharmony between dropout and batch normalization by variance shift. In *arXiv:1801.05134*, 2018.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *arXiv:1608.03983*, 2016.

Ping Luo, Jiamin Ren, and Zhanglin Peng. Differentiable learning-to-normalize via switchable normalization. *arXiv:1806.10779*, 2018.

Song Mei, Yu Bai, and Andrea Montanari. The landscape of empirical risk for non-convex losses. In *arXiv:1607.06534*, 2016.

Ari S. Morcos, David G.T. Barrett, Neil C. Rabinowitz, and Matthew Botvinick. On the importance of single directions for generalization. In *ICLR*, 2018.

M. Opper, W. Kinzel, J. Kleinz, and R. Nehl. On the ability of the optimal perceptron to generalise. In *Journal of Physics A: Mathematical and General*, volume 23(11), pp. 581, 1990.

Jeffrey Pennington and Yasaman Bahri. Geometry of neural network loss surfaces via random matrix theory. In *ICML*, 2017.

Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl Dickstein. On the expressive power of deep neural networks. In *ICML*, 2017.

Salah Rifai, Xavier Glorot, Yoshua Bengio, and Pascal Vincent. Adding noise to the input of a model trained with a regularized objective. *arXiv:1104.3250 [cs]*, April 2011. URL `http://arxiv.org/abs/1104.3250`. arXiv: 1104.3250.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. In *ICJV*, 2015.

David Saad and Sara A. Solla. Dynamics of on-line gradient descent learning for multilayer neural networks. In *NIPS*, 1996.

Tim Salimans and Diederik P. Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *arXiv:1602.07868*, 2016.

Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How Does Batch Normalization Help Optimization? (No, It Is Not About Internal Covariate Shift). *arXiv:1805.11604 [cs, stat]*, May 2018. URL `http://arxiv.org/abs/1805.11604`. arXiv: 1805.11604.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. In *Journal of Machine Learning Research*, 2014.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. *arXiv:1512.00567 [cs]*, December 2015. URL `http://arxiv.org/abs/1512.00567`. arXiv: 1512.00567.

Mattias Teye, Hossein Azizpour, and Kevin Smith. Bayesian uncertainty estimation for batch normalized deep networks. In *ICML*, 2018.

Yuandong Tian. An analytical formula of population gradient for two-layered relu network and its applications in convergence and critical point analysis. In *ICML*, 2017.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv:1607.08022*, 2016.

Twan. van Laarhoven. L2 regularization versus batch and weight normalization. In *arXiv:1706.05350*, 2017.

Stefan Wager, Sida Wang, and Percy Liang. Dropout Training as Adaptive Regularization. *arXiv:1307.1493 [cs, stat]*, July 2013. URL `http://arxiv.org/abs/1307.1493`. arXiv: 1307.1493.

Yuxin Wu and Kaiming He. Group normalization. *arXiv:1803.08494*, 2018.

Yuki Yoshida, Ryo Karakida, Masato Okada, and Shun ichi Amari. Statistical mechanical analysis of online learning with weight normalization in single layer perceptron. In *Journal of the Physical Society of Japan*, 2017.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, , and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017a.

Qiuyi Zhang, Rina Panigrahy, and Sushant. Sachdeva. Electron-proton dynamics in deep learning. In *arXiv:1702.00458*, 2017b.

Appendices

A Notations

Table 2: Several notations are summarized for reference.

$\mu_\mathcal{B}, \sigma^2_\mathcal{B}$	batch mean, batch variance
$\mu_\mathcal{P}, \sigma^2_\mathcal{P}$	population mean, population variance
$\mathbf{x}, y$	input of a network, output of a network
$\bar{y}$	ground truth of an output
$h, \hat{h}$	hidden value before and after BN
$\bar{h}$	hidden value after population normalization
γ, β	scale parameter, shift parameter
$g(\cdot)$	activation function
$\mathbf{w}, \mathbf{w}^*$	weight vector, ground truth weight vector
$\tilde{\mathbf{w}}$	normalized weight vector
M, N, P	batch size, number of neurons, sample size
α	an effective load value $\alpha = P/N$
ζ	regularization strength (coefficient)
ρ	Kurtosis of a distribution
δ	gradient of the activation function
$\eta_\text{eff}, \eta_\text{max}$	effective, maximum learning rate
R	overlapping ratio (angle) between $\tilde{\mathbf{w}}$ and $\mathbf{w}^*$
L	norm (length) of $\mathbf{w}$
$\lambda_\text{max}, \lambda_\text{min}$	maximum, minimum eigenvalue
ϵ_gen	generalization error

B More Empirical Settings and Results

All experiments in Sec.5 are conducted in CIFAR10 with a CNN architecture similar to (Salimans & Kingma, 2016) that is summarized as 'conv(3,32)-conv(3,32)-conv(3,64)-conv(3,64)-pool(2,2)-fc(512)-fc(10)', where 'conv(3,32)' represents a convolution with kernel size 3 and 32 channels, 'pool(2,2)' is max-pooling with kernel size 2 and stride 2, and 'fc' indicates a full connection. We follow a configuration for training by using SGD with a momentum value of 0.9 and continuously decaying the learning rate by a factor of 10^{-4} each step. For different batch sizes, the initial learning rate is scaled proportionally with the batch size to maintain a similar learning dynamics (Goyal et al., 2017).

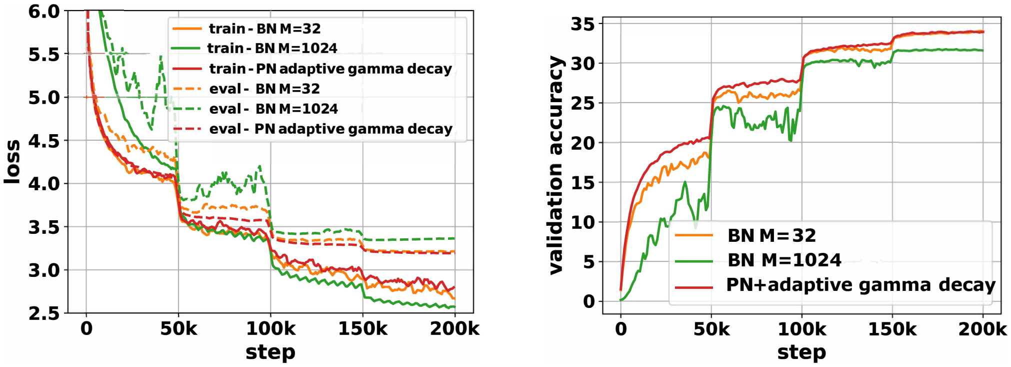
B.1 Results in downsampled ImageNet

Besides CIFAR10, we also evaluate theorem 1 by employing a downsampled version of ImageNet (Loshchilov & Hutter, 2016), which contains identical 1.2 million data and 1k categories as the original ImageNet, but each image is scaled to 32×32. We train ResNet18 in downsampled ImageNet by following the training protocol used in (He et al., 2016). In particular, ResNet18 is trained by using SGD with momentum of 0.9 and the initial learning rate is 0.1, which is then decayed by a factor of 10 after 30, 60, and 90 training epochs.

In downsampled ImageNet, we observe similar trends as those presented in CIFAR10. For example, we see that BN would imped both loss and accuracy when batch size increases. When increasing M to 1024 as shown in Fig.6, both the loss and validation accuracy decrease because the regularization from the random batch statistics reduces in large batch size, resulting in overtraining. This can be seen by the gap between the training and the validation loss. Nevertheless, we see that the reduction of regularization can be complemented when PN is trained with adaptive gamma decay, which makes PN performed comparably to BN in downsampled ImageNet.

B.2 Impact of BN to the Norm of Parameters

We demonstrate the impact of BN to the norm of parameters. We compare BN with vanilla SGD, where a network is first trained by BN in order to converge to a local minima when the parameters do not change much. At this local minima, the weight vector is frozen and denoted as $\mathbf{w}^{bn}$. Then this



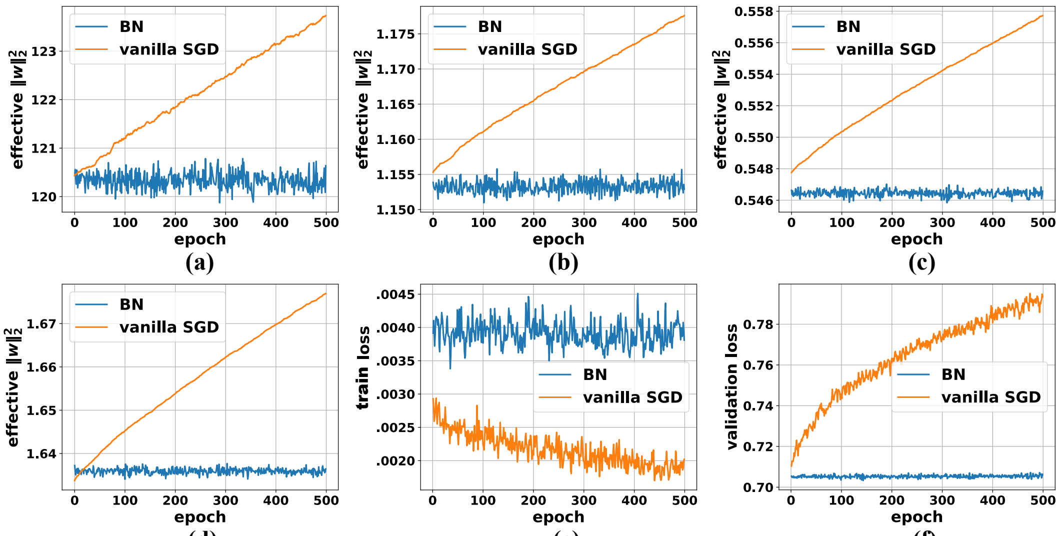
(a) Comparisons of train and validation loss.

(b) Comparisons of validation accuracy.

Figure 6: **Results of downsampled ImageNet.** (a) plots training and evaluation loss. (b) shows validation accuracy. The models are trained on 8 GPUs.

network is finetuned by using vanilla SGD with a small learning rate 10^{-3} with the kernel parameters initialized by $\mathbf{w}^{sgd} = \gamma\frac{\mathbf{w}^{bn}}{\sigma}$, where σ is the moving average of $\sigma_{\mathcal{B}}$.

Fig.7 below visualizes the results. As $\mu_{\mathcal{B}}$ and $\sigma_{\mathcal{B}}$ are removed in the vanilla SGD, it is found from the last two figures that the training loss decreases while the validation loss increases, meaning that the reduction in regularization makes the network converged to a sharper local minimum that generalizes less well. The magnitudes of kernel parameters $\mathbf{w}^{sgd}$ at different layers are also displayed in the first four figures. All of them increase after freezing BN, due to the release of regularization on these parameters.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 7: **Study of parameter norm.** Vanilla SGD is finetuned from a network pretrained by BN on CIFAR10. The first four figures show the magnitude of the kernel parameters in different layers in finetuning, compared to the effective norm of BN defined as $\gamma\frac{\|\mathbf{w}\|}{\sigma_{\mathcal{B}}}$. The last two figures compare the training and validation losses in finetuning.

C Proof of Results

C.1 Proof of Theorem 1

Theorem 1 (Regularization of $\mu_{\mathcal{B}}, \sigma_{\mathcal{B}}$)**.** *Let ζ be the strength (coefficient) of the regularization and the activation function be ReLU. Then*

$$\frac{1}{P}\sum_{j=1}^{P}\mathbb{E}_{\mu_{\mathcal{B}},\sigma_{\mathcal{B}}}\ell(\hat{h}^j) \simeq \frac{1}{P}\sum_{j=1}^{P}\ell(\bar{h}^j) + \zeta\gamma^2,$$

$$\text{and }\ \zeta = \underbrace{\frac{\rho+2}{8M}F_{\gamma}}_{\text{from } \sigma_{\mathcal{B}}} + \underbrace{\frac{1}{2M}\frac{1}{P}\sum_{j=1}^{P}\sigma(\bar{h}^j)}_{\text{from } \mu_{\mathcal{B}}},$$

where $\bar{h}^j = \gamma\frac{\mathbf{w}^{\mathsf{T}}\mathbf{x}^j-\mu_{\mathcal{P}}}{\sigma_{\mathcal{P}}} + \beta$, ρ is the kurtosis of the distribution of $\mathbf{w}^{\mathsf{T}}\mathbf{x}$, F_{γ} is a Fisher Information Matrix of γ, and $\sigma(\cdot)$ is a sigmoid function.

Proof. Let $\hat{h}^j = \gamma\frac{\mathbf{w}^T\mathbf{x}^j-\mu_{\mathcal{B}}}{\sigma_{\mathcal{B}}} + \beta$ and $\bar{h}^j = \gamma\frac{\mathbf{w}^T\mathbf{x}^j-\mu_{\mathcal{P}}}{\sigma_{\mathcal{P}}} + \beta$. We prove theorem 1 by performing a Taylor expansion on a function $A(\hat{h}^j)$ at $\bar{h}^j$, where $A(\hat{h}^j)$ is a function of $\hat{h}^j$ defined according to a particular activation function. The negative log likelihood function of the single-layer perceptron can be generally defined as $-\log p(y^j|\hat{h}^j) = A(\hat{h}^j) - y^j\hat{h}^j$, which is similar to the loss function of the generalized linear models with different activation functions.

$$\begin{aligned}
\frac{1}{P}\sum_{j=1}^{P}\mathbb{E}_{\mu_{\mathcal{B}},\sigma_{\mathcal{B}}}[l(\hat{h}^j)] &= \frac{1}{P}\sum_{j=1}^{P}\mathbb{E}_{\mu_{\mathcal{B}},\sigma_{\mathcal{B}}}\left[A(\hat{h}^j) - y^j\hat{h}^j\right] \\
&= \frac{1}{P}\sum_{j=1}^{P}(A(\bar{h}^j) - y^j\bar{h}^j) + \frac{1}{P}\sum_{j=1}^{P}\mathbb{E}_{\mu_{\mathcal{B}},\sigma_{\mathcal{B}}}\left[-y^j(\hat{h}^j - \bar{h}^j) + A(\hat{h}^j) - A(\bar{h}^j)\right] \\
&= \frac{1}{P}\sum_{j=1}^{P}l(\bar{h}^j) + \frac{1}{P}\sum_{j=1}^{P}\mathbb{E}_{\mu_{\mathcal{B}},\sigma_{\mathcal{B}}}\left[(A'(\bar{h}^j) - y^j)(\hat{h}^j - \bar{h}^j)\right] \\
&+ \frac{1}{P}\sum_{j=1}^{P}\mathbb{E}_{\mu_{\mathcal{B}},\sigma_{\mathcal{B}}}\left[\frac{A''(\bar{h}^j)}{2}(\hat{h}^j - \bar{h}^j)^2\right] \\
&= \frac{1}{P}\sum_{j=1}^{P}l(\bar{h}^j) + R^f + R^q,
\end{aligned}$$

where $A'(\cdot)$ and $A''(\cdot)$ denote the first and second derivatives of function $A(\cdot)$. The first and second order terms in the expansion are represented by R^f and R^q respectively. To derive the analytical forms of R^f and R^q, we take a second-order Taylor expansion of of $\frac{1}{\sigma_{\mathcal{B}}}$ and $\frac{1}{\sigma_{\mathcal{B}}^2}$ around σ_P, it suffices to have

$$\frac{1}{\sigma_{\mathcal{B}}} \approx \frac{1}{\sigma_{\mathcal{P}}} + (-\frac{1}{\sigma_{\mathcal{P}}^2})(\sigma_{\mathcal{B}} - \sigma_{\mathcal{P}}) + \frac{1}{\sigma_{\mathcal{P}}^3}(\sigma_{\mathcal{B}} - \sigma_{\mathcal{P}})^2$$

and

$$\frac{1}{\sigma_{\mathcal{B}}^2} \approx \frac{1}{\sigma_{\mathcal{P}}^2} + (-\frac{2}{\sigma_{\mathcal{P}}^3})(\sigma_{\mathcal{B}} - \sigma_{\mathcal{P}}) + \frac{3}{\sigma_{\mathcal{P}}^4}(\sigma_{\mathcal{B}} - \sigma_{\mathcal{P}})^2.$$

By applying the distributions of $\mu_\mathcal{B}$ and $\sigma_\mathcal{B}$ in the paper, R^f can be derived as

$$\begin{aligned}
R^f &= \frac{1}{P}\sum_{j=1}^{P}\mathbb{E}_{\mu_\mathcal{B},\sigma_\mathcal{B}}\left[(A'(\bar{h}^j)-y^j)(\hat{h}^j\bar{h}^j)\right]\\
&= \frac{1}{P}\sum_{j=1}^{P}\mathbb{E}_{\mu_\mathcal{B},\sigma_\mathcal{B}}\left[(A'(\bar{h}^j)-y^j)\left(\gamma\frac{\mathbf{w}^T\mathbf{x}^j-\mu_\mathcal{B}}{\sigma_\mathcal{B}}-\gamma\frac{\mathbf{w}^T\mathbf{x}^j-\mu_\mathcal{P}}{\sigma_\mathcal{P}}\right)\right]\\
&= \frac{1}{P}\sum_{j=1}^{P}\mathbb{E}_{\mu_\mathcal{B},\sigma_\mathcal{B}}\left[(A'(\bar{h}^j y^j)\left(\gamma\mathbf{w}^T\mathbf{x}^j\left(\frac{1}{\sigma_\mathcal{B}}-\frac{1}{\sigma_\mathcal{P}}\right)+\gamma\left(-\frac{\mu_\mathcal{B}}{\sigma_\mathcal{B}}+\frac{\mu_\mathcal{P}}{\sigma_\mathcal{P}}\right)\right)\right]\\
&= \frac{1}{P}\sum_{j=1}^{P}\gamma(A'(\bar{h}^j)-y^j)(\mathbf{w}^T\mathbf{x}^j-\mu_\mathcal{P})\mathbb{E}_{\sigma_\mathcal{B}}\left[\frac{1}{\sigma_\mathcal{B}}-\frac{1}{\sigma_\mathcal{P}}\right]\\
&= \frac{1}{P}\sum_{j=1}^{P}\frac{\rho+2}{4M}\gamma(A'(\bar{h}^j)-y^j)\frac{\mathbf{w}^T\mathbf{x}^j-\mu_\mathcal{P}}{\sigma_\mathcal{P}}.
\end{aligned}$$

This R^f term can be understood as below. Let $h=\frac{\mathbf{w}^T\mathbf{x}-\mu_\mathcal{P}}{\sigma_\mathcal{P}}$ and the distribution of the population data be p_{xy}. We establish the following relationship

$$\begin{aligned}
\mathbb{E}_{(x,y)\sim p_{xy}}\mathbb{E}_{\mu_\mathcal{B},\sigma_\mathcal{B}}\left[(A'(\bar{h})-y)h\right] &= \mathbb{E}_{\mu_\mathcal{B},\sigma_\mathcal{B}}\mathbb{E}_{x\sim p_x}\mathbb{E}_{y|x\sim p_{y|x}}\left[(A'(\bar{h})-y)h\right]\\
&= \mathbb{E}_{\mu_\mathcal{B},\sigma_\mathcal{B}}\mathbb{E}_{x\sim p_x}\left[(\mathbb{E}\left[y|x\right]-\mathbb{E}_{y|x\sim p_{y|x}}\left[y\right])h\right]\\
&= 0.
\end{aligned}$$

Since the sample mean converges in probability to the population mean by the Weak Law of Large Numbers, for all $\epsilon>0$ and a constant number K ($\exists K>0$ and $\forall P>K$), we have $\left|R^f-\mathbb{E}_{(x,y)\sim p_{xy}}\mathbb{E}_{\mu_\mathcal{B},\sigma_\mathcal{B}}\left[(A'(\bar{h})-y)h\right]\right|<\frac{\rho+2}{4M}\epsilon$. The above equation means that R^f is sufficiently small given moderately large number of data points P (the above inequality holds when $P>30$).

On the other hand, R^q can be derived as

$$\begin{aligned}
R^q &= \frac{1}{P}\sum_{j=1}^{P}\mathbb{E}_{\mu_\mathcal{B},\sigma_\mathcal{B}}\left[\frac{A''(\bar{h}^j)}{2}(\hat{h}^j-\bar{h}^j)^2\right]\\
&= \frac{1}{P}\sum_{j=1}^{P}\frac{A''(\bar{h}^j)}{2}\mathbb{E}_{\mu_\mathcal{B},\sigma_\mathcal{B}}\left[(\gamma\frac{\mathbf{w}^T\mathbf{x}^j-\mu_\mathcal{B}}{\sigma_\mathcal{B}}+\beta-\gamma\frac{\mathbf{w}^T\mathbf{x}^j-\mu_\mathcal{P}}{\sigma_\mathcal{P}}+\beta)^2\right]\\
&= \frac{1}{P}\sum_{j=1}^{P}\frac{A''(\bar{h}^j)}{2}\mathbb{E}_{\mu_\mathcal{B},\sigma_\mathcal{B}}\left[(\gamma\mathbf{w}^T\mathbf{x}^j)^2(\frac{1}{\sigma_\mathcal{B}}-\frac{1}{\sigma_\mathcal{P}})^2-2\gamma\mu_\mathcal{P}\mathbf{w}^T\mathbf{x}^j(\frac{1}{\sigma_\mathcal{B}}-\frac{1}{\sigma_\mathcal{P}})^2+(\frac{\mu_\mathcal{B}}{\sigma_\mathcal{B}}-\frac{\mu_\mathcal{P}}{\sigma_\mathcal{P}})^2\right]\\
&\simeq \frac{1}{P}\sum_{j=1}^{P}\frac{\gamma^2A''(\bar{h}^j)}{2}\left((\mathbf{w}^T\mathbf{x}^j-\mu_\mathcal{P})^2\mathbb{E}_{\mu_\mathcal{B},\sigma_\mathcal{B}}\left[(\frac{1}{\sigma_\mathcal{B}}-\frac{1}{\sigma_\mathcal{P}})^2\right]+\mathbb{E}_{\mu_\mathcal{B},\sigma_\mathcal{B}}\left[\left(\frac{\mu_\mathcal{B}-\mu_P}{\sigma_\mathcal{B}}\right)^2\right]\right)\\
&= \frac{1}{P}\sum_{j=1}^{P}\frac{\gamma^2A''(\bar{h}^j)}{2}\left((\frac{\mathbf{w}^T\mathbf{x}^j-\mu_\mathcal{P}}{\sigma_\mathcal{P}})^2\frac{\rho+2}{4M}+\frac{1}{M}(1+\frac{3(\rho+2)}{4M})\right).
\end{aligned}$$

Note that $\frac{\partial^2 l(\bar{h}^j)}{\partial\gamma^2}=A''(\bar{h}^j)(\frac{\mathbf{w}^T\mathbf{x}^j-\mu_\mathcal{P}}{\sigma_\mathcal{P}})^2$, we have $F(\gamma)=\frac{1}{P}\sum_{j=1}^{P}A''(\bar{h}^j)(\frac{\mathbf{w}^T\mathbf{x}^j-\mu_\mathcal{P}}{\sigma_\mathcal{P}})^2$ been an estimator of the Fisher Information Matrix (FIM) with respect to the scale parameter γ, according to the definition of Fisher information. Then, by neglecting $O(1/M^2)$ high-order term in R^q, we get

$$R^q\simeq\frac{\rho+2}{8M}F(\gamma)\gamma^2+\frac{\mu_{d^2A}}{2M}\gamma^2,$$

where μ_{d^2A} indicates the mean of the second derivative of $A(h)$.

C.2 Theorem 1 with ReLU

For the ReLU non-linear activation function, that is $f(h) = \max(h, 0)$, we use its continuous approximation softplus function $f(h) = \log(1 + \exp(h))$ to derive the partition function $A(h)$. In this case, we have $\mu_{d^2A} = \frac{1}{P}\sum_{j=1}^{P} \sigma(\bar{h}^j)$. Therefore, we have $\zeta = \frac{\rho+2}{8M} F_\gamma + \frac{1}{2M}\frac{1}{P}\sum_{j=1}^{P}\sigma(\bar{h}^j)$ as shown in theorem 1.

C.3 Theorem 1 with Identity activation Function

For a loss function in the form $L = \frac{1}{P}\sum_{j=1}^{P}\left(\mathbf{w}^{*\mathsf{T}}\mathbf{x}^j - \gamma(\mathbf{w}^{\mathsf{T}}\mathbf{x}^j)/\sigma_{\mathcal{B}}\right)^2$, we have $F_\gamma = 2\lambda$ and the regularization contribution from $\mu_{\mathcal{B}}$ can be neglected. We also have $\rho = 0$ for Gaussian input distribution. The exact expression of theorem 1 is also possible for such linear regression problem. Under the condition of Gaussian input $\mathbf{x} \sim \mathcal{N}(0, 1/N)$, $h = \mathbf{w}^{\mathsf{T}}\mathbf{x}$ is also a random variable satisfying a normal distribution $\mathcal{N}(0, 1)$, it can be derived that $\mathbb{E}\left(\sigma_{\mathcal{B}}^{-1}\right) = \frac{\sqrt{M}}{\sqrt{2}\sigma_{\mathcal{P}}}\frac{\Gamma\left(\frac{M-2}{2}\right)}{\Gamma\left(\frac{M-1}{2}\right)}$ and $\mathbb{E}\left(\sigma_{\mathcal{B}}^{-2}\right) = \frac{M}{\sigma_{\mathcal{P}}^2}\frac{\Gamma\left(\frac{M-1}{2}-1\right)}{\Gamma\left(\frac{M-1}{2}\right)}$, therefore

$$\zeta = \lambda\left(1 + \frac{M\Gamma\big((M-3)/2\big)}{2\Gamma\big((M-1)/2\big)} - \sqrt{2M}\frac{\Gamma\big((M-2)/2\big)}{\Gamma\big((M-1)/2\big)}\right).$$

Furthermore, the expression of ζ can be simplified as $\zeta = \frac{3}{4M}$. If the bias term is neglected in a simple linear regression, contributions from $\mu_{\mathcal{B}}$ to the regularization term is neglected and thus $\zeta = \frac{1}{4M}$. Note that if one uses mean square error without being divided by 2 during linear regression, the values for ζ should be multiplied by 2 as well, where $\zeta = \frac{1}{2M}$.

□

C.4 Dynamical Equations

Here we discuss the dynamical equations of BN. Let the length of teacher's weight vector be 1, that is, $\frac{1}{N}\mathbf{w}^{*T}\mathbf{w}^* = 1$. We introduce a normalized weight vector of the student as $\widetilde{\mathbf{w}} = \sqrt{N}\gamma\frac{\mathbf{w}}{\|\mathbf{w}\|}$. Then the overlapping ratio between teacher and student, the length of student's vector, and the length of student's normalized weight vector are $\frac{1}{N}\widetilde{\mathbf{w}}^T\mathbf{w}^* = QR = \gamma R$, $\frac{1}{N}\widetilde{\mathbf{w}}^T\widetilde{\mathbf{w}} = Q^2 = \gamma^2$, and $\frac{1}{N}\mathbf{w}^{\mathsf{T}}\mathbf{w} = L^2$ respectively, where $Q = \gamma$. And we have $\frac{1}{N}\mathbf{w}^T\mathbf{w}^* = LR$.

We write the dynamical equations of BN as follows,

$$\begin{cases} \frac{dQ}{dt} = \eta\frac{I_1}{Q} - \eta\zeta Q, \\ \frac{dR}{dt} = \eta\frac{Q}{L^2}I_3 - \eta\frac{R}{L^2}I_1 - \eta^2\frac{Q^2R}{2L^4}I_2, \\ \frac{dL}{dt} = \eta^2\frac{Q^2}{2L^3}I_2, \end{cases}$$

where $I_1 = \langle\delta\tilde{\mathbf{w}}^{\mathsf{T}}\mathbf{x}\rangle_{\mathbf{x}}$, $I_2 = \langle\delta^2\mathbf{x}^{\mathsf{T}}\mathbf{x}\rangle_{\mathbf{x}}$, and $I_3 = \langle\delta\mathbf{w}^{*\mathsf{T}}\mathbf{x}\rangle_{\mathbf{x}}$, which are the terms presented in $\frac{d\gamma^2}{dt}$, $\frac{dRL}{dt}$, and $\frac{dL^2}{dt}$. They are used to simplify notations.

Proposition 1. *Let λ_Q^{bn}, λ_R^{bn} be the eigenvalues of the Jacobian matrix at $\theta_0 = (Q_0, 1, L_0)$ corresponding to the order parameters Q and R respectively in BN. Then*

$$\begin{cases} \lambda_Q^{bn} = \frac{\eta}{Q_0}\frac{\partial I_1}{\partial Q} - \eta\zeta Q_0, \\ \lambda_R^{bn} = \frac{\partial I_2}{2\partial R}\frac{\eta Q_0}{2L_0^2}(\eta_{\max}^{bn} - \eta_{\text{eff}}^{bn}), \end{cases}$$

where $\eta_{\max}^{bn}$ and η_{eff}^{bn} are the maximum and effective learning rates respectively in BN.

Proof. Firstly, note that the change of L will not change I_1, I_2 and I_3. Thus we have $\frac{\partial I_1}{\partial L} = \frac{\partial I_2}{\partial L} = \frac{\partial I_3}{\partial L} = 0$. And we also neglect the term proportional to η^2 because of the learning rate decays to a small value when converged. At fixed point $R_0 = 1$, we have $\partial(QI_3 - I_1)/\partial Q = 0$, thus the Jacobian of BN is

$$J^{bn} = \begin{bmatrix} \frac{\eta}{Q_0}\frac{\partial I_1}{\partial Q} - 2\eta\zeta & \frac{\eta}{Q_0}\frac{\partial I_1}{\partial R} & 0 \\ 0 & \frac{\eta}{L_0^2}\left(\frac{Q_0 \partial I_3}{\partial R} - \frac{\partial I_1}{\partial R} - \zeta Q_0^2\right) - \frac{\eta^2 Q_0^2}{2L_0^4}\frac{\partial I_2}{\partial R} & 0 \\ 0 & \frac{\eta^2 Q_0^2}{2L_0^3}\frac{\partial I_2}{\partial R} & 0 \end{bmatrix},$$

and the eigenvalues of J^{bn} can be obtained by inspection

$$\begin{cases} \lambda_Q^{bn} = \frac{\eta}{Q_0}\frac{\partial I_1}{\partial Q} - 2\eta\zeta, \\ \lambda_R^{bn} = \frac{\eta}{L_0^2}\left(\frac{Q_0 \partial I_3}{\partial R} - \frac{\partial I_1}{\partial R} - \zeta Q_0^2\right) - \frac{\eta^2 Q_0^2}{2L_0^4}\frac{\partial I_2}{\partial R} = \frac{\partial I_2}{\partial R}\frac{\eta Q_0}{2L_0^2}\left(\eta_{\max}^{bn} - \eta_{\text{eff}}^{bn}\right), \\ \lambda_L^{bn} = 0. \end{cases}$$

Since $\gamma_0 = Q_0$, we have $\eta_{\max}^{bn} = (\frac{\partial(\gamma_0 I_3 - I_1)}{\gamma_0 \partial R} - \zeta\gamma_0)/\frac{\partial I_2}{2\partial R}$ and $\eta_{\text{eff}}^{bn} = \frac{\eta\gamma_0}{L_0^2}$. □

C.5 STABLE FIXED POINTS OF BN

Proposition 2. *When activation function is ReLU or sigmoid, then (i) $\lambda_Q^{bn} < 0$, and (ii) $\lambda_R^{bn} < 0$ iff $\eta_{\max}^{bn} > \eta_{\text{eff}}^{bn}$.*

Proof. Firstly, when activation function is ReLU, we derive $I_1 = \frac{Q(\pi R + 2\sqrt{1-R^2} + 2R\arcsin(R))}{4\pi} - \frac{Q^2}{2}$, which gives

$$\frac{\partial I_1}{\partial Q} = -Q + \frac{\pi R + 2\sqrt{1-R^2} + 2R\arcsin(R)}{4\pi}.$$

Therefore at the fixed point of BN $\theta_0^{bn} = (\frac{1}{2\zeta+1}, 1)$, we have

$$\lambda_Q^{bn} = \eta(\frac{1}{Q_0}\frac{\partial I_1}{\partial Q} - 2\zeta) = \eta(\frac{1}{Q_0}(-1 + \frac{1}{2Q_0} - 2\zeta) = -\zeta - \frac{1}{2} < 0.$$

Note that $\mathbf{x}^T\mathbf{x}$ approximately equals 1. We get

$$\begin{aligned} I_2 &= \int\limits_{u,v} [g'(s)\,(g(t) - g(s))]^2 DsDt \\ &= \int_0^{+\infty}\int_0^{+\infty} v^2 DsDt + \int_0^{+\infty}\int_{-\infty}^{+\infty} s^2 DsDv - 2\int_0^{+\infty}\int_{-\infty}^{+\infty} stDsDt \\ &= \frac{Q^2}{2} + \frac{\pi R + 2R\sqrt{1-R^2} + 2\arcsin(R)}{4\pi} - \frac{Q(\pi R + 2\sqrt{1-R^2} + 2R\arcsin(R))}{2\pi}. \end{aligned}$$

At the fixed point we have $\frac{\partial I_2}{\partial R} = -Q_0 < 0$. Therefore, we conclude that $\lambda_R^{bn} < 0$ iff $\eta_{\max}^{bn} > \eta_{\text{eff}}^{bn}$.

Secondly, when activation function is sigmoid, we employ $\mathrm{erf}(x/\sqrt{2})$ to analyze sigmoid function. Therefore, we have

$$\begin{aligned} I_1 &= \int\limits_{u,v} [g'(u)\,(g(v) - g(u)\,u] DuDv \\ &= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} g'(u)g(v)uDuDv - \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} g'(u)g(u)uDuDv. \end{aligned}$$

By using the Fourier Transformation of integral on multivariate Gaussian probability density, that is,

$$\begin{aligned} &\int \frac{dx_1 \cdots dx_n}{\sqrt{(2\pi)^n |C|}} \exp\{-\frac{1}{2}X^T C^{-1} X\} \times f_1(x_1) \cdots f_n(x_n) \\ &= \int \frac{dy_1 \cdots dy_n}{\sqrt{(2\pi)^n}} \exp\{-\frac{1}{2}Y^T C Y\} \times \tilde{f}_1(y_1) \cdots \tilde{f}_n(y_n), \end{aligned}$$

where $\tilde{f}_j(y) = \int \frac{dx}{\sqrt{2\pi}} f_j(x) e^{iyx}$.

It suffices to have

$$I_1 = \frac{2QR}{(\pi + \pi Q^2)\sqrt{2 + 2Q^2 - Q^2R^2}} - \frac{2Q^2}{(\pi + \pi Q^2)\sqrt{1 + 2Q^2}}$$

and

$$I_2 = \frac{4}{\pi\sqrt{1 + 2Q^2}}(\arcsin \frac{1}{1 + 3Q^2} - \arcsin \frac{1}{\sqrt{1 + 3Q^2}\sqrt{2(1 + 2Q^2) - 2Q^2R^2}}).$$

At the fixed point $\theta_0^{bn} = (\frac{1}{2\zeta+1}, 1)$, it can be shown that $\frac{\partial(I_1/Q)}{\partial Q} < 0$ and thus $\lambda_Q^{bn} = \eta(\frac{\partial(I_1/Q)}{\partial Q} - \zeta) < 0$, we have $\frac{\partial I_2}{\partial R} < 0$. We also conclude that $\lambda_R^{bn} < 0$ iff $\eta_{\max}^{bn} > \eta_{\mathrm{eff}}^{bn}$. □

C.6 Maximum Learning Rate of BN

Proposition 3. *When the activation function is ReLU, then $\eta_{\max}^{bn} \geq \eta_{\max}^{\{wn,sgd\}} + 2\zeta$, where $\eta_{\max}^{bn}$ and $\eta_{\max}^{\{wn,sgd\}}$ indicate the maximum learning rates of BN, WN, and vanilla SGD respectively.*

Proof. From the above results, we have $I_1 = \frac{Q(\pi R + 2\sqrt{1-R^2} + 2R\arcsin(R))}{4\pi} - \frac{Q^2}{2}$, which gives $\partial I_1/\partial R \geq 0$ at the fixed point of BN. And we get

$$\begin{aligned}
I_3 &= \int\limits_{u,v} [g'(s)\,(g(t) - g(s)\,t] DsDt \\
&= \int\limits_{u,v} g'(s)g(t)tDsDt - \int\limits_{u,v} g'(s)g(s)tDsDt \\
&= \int_0^{+\infty}\int_0^{+\infty} t^2 DsDt - \int_0^{+\infty}\int_{-\infty}^{+\infty} stDsDt \\
&= \frac{\pi + 2R\sqrt{1-R^2} + 2\arcsin(R)}{4\pi} - \frac{QR}{2}.
\end{aligned}$$

Then we derive that $\frac{\partial I_2}{\partial R} < 0$ and $\frac{\partial(I_3 - I_1)}{\partial R} / \frac{\partial I_2}{2\partial R}$ has the same value at their respective fixed point of BN, WN and vanilla SGD. At the fixed point of BN, $Q_0 = \gamma_0 = \frac{1}{2\zeta+1} < 1$, then we have

$$\begin{aligned}
\eta_{\max}^{bn} &= (\frac{\partial(\gamma_0 I_3 - I_1)}{Q_0 \partial R} - \zeta\gamma_0) / \frac{\partial I_2}{2\partial R} \\
&= \frac{\partial(I_3 - I_1)}{\partial R} / \frac{\partial I_2}{2\partial R} + (1 - \frac{1}{\gamma_0})\frac{\partial I_1}{\partial R} / \frac{\partial I_2}{2\partial R} - \zeta\gamma_0 / \frac{\partial I_2}{2\partial R} \\
&\geq \frac{\partial(I_3 - I_1)}{\partial R} / \frac{\partial I_2}{2\partial R} + 2\zeta \\
&= \eta_{\max}^{\{wn,sgd\}} + 2\zeta.
\end{aligned}$$

□