MEAN-FIELD ANALYSIS OF BATCH NORMALIZATION

Mingwei Wei
Department of Physics and Astronomy
Northwestern University
Evanston, IL 60202, USA
m.wei@u.northwestern.edu

James Stokes
Tunnel
New York, NY 10010, USA
james@tunnel.tech

David J Schwab
The Graduate Center
The City University of New York
New York, NY 10016 USA
Dschwab@gc.cuny.edu

ABSTRACT

Batch Normalization (BatchNorm) is an extremely useful component of modern neural network architectures, enabling optimization using higher learning rates and achieving faster convergence. In this paper, we use mean-field theory to analytically quantify the impact of BatchNorm on the geometry of the loss landscape for multi-layer networks consisting of fully-connected and convolutional layers. We show that it has a flattening effect on the loss landscape, as quantified by the maximum eigenvalue of the Fisher Information Matrix. These findings are then used to justify the use of larger learning rates for networks that use BatchNorm, and we provide quantitative characterization of the maximal allowable learning rate to ensure convergence. Experiments support our theoretically predicted maximum learning rate, and furthermore suggest that networks with smaller values of the BatchNorm parameter γ achieve lower loss after the same number of epochs of training.

1 INTRODUCTION

Deep neural networks have achieved remarkable success in the past decade on tasks that were out of reach prior to the era of deep learning (Krizhevsky et al., 2012; He et al., 2016b). Amongst the myriad reasons for these successes are powerful computational resources, large datasets, new optimization algorithms, and modern architecture designs (Russakovsky et al., 2015; Kingma & Ba, 2015). In many modern deep learning architectures, one key component is batch normalization (BatchNorm). BatchNorm is a module that can be introduced in layers of deep neural networks that normalizes hidden layer outputs to have a common first and second moment. Empirically, BatchNorm enables optimization using much larger learning rates, and achieves better convergence (Ioffe & Szegedy, 2015).

Despite significant practical utility, a theoretical understanding of BatchNorm is still lacking. A widely held view is that BatchNorm improves training by "reducing of internal covariate shift" (ICF) (Ioffe & Szegedy, 2015). Internal covariate shift refers to the change in the input distribution of internal layers of the deep network due to changes of the weights. Recent results (Santurkar et al., 2018), however, cast doubt on the ICF expalanation, by demonstrating that noisy BatchNorm increases ICF yet still improves training as in regular BatchNorm. This raises the question of whether the utility of BatchNorm is indeed related to the reduction of ICF. Instead, it is argued by Santurkar et al. (2018) that BatchNorm actually improves the Lipschitzness of the loss and gradient.

Meanwhile, dynamical mean-field theory (Sompolinsky & Zippelius, 1982), a powerful theoretical technique, has recently been applied by Poole et al. (2016) to ensembles of multi-layer random neural networks. This theory studies networks with an i.i.d. Gaussian distribution of weights and biases. Most recent work focuses on the analysis of order parameter flows and their fixed points

(Schoenholz et al., 2017; Xiao et al., 2018; Yang & Schoenholz, 2017), including their stability and decay rates. Importantly, Karakida et al. (2018) also successfully used mean-field analysis to estimate the spectral properties of the Fisher Information Matrix.

In this paper, we analytically quantify the impact of BatchNorm on the landscape of the loss function, by using mean-field theory to estimate the spectral properties of the Fisher Information Matrix (FIM) for typical batch-normalized neural networks. In particular, it is shown that BatchNorm reduces the maximal eigenvalue of the FIM provided that the normalization coefficient γ is not too large. By drawing on results linking Fisher Information to the geometry of the loss function, we explain how BatchNorm neural networks can be trained with a larger learning rate without leading to parameter explosion, and provide upper bounds on the learning rate in terms of the BatchNorm parameters. As an additional contribution motivated by our theoretical findings, we demonstrate an empirical correlation between the BatchNorm parameter γ and test loss. In particular, networks with smaller γ achieve lower loss after a fixed number of training epochs.

2 PRELIMINARIES

In our theoretical analysis, we employ the recent application of mean-field theory to neural networks which studies an ensemble of random neural networks with pre-defined i.i.d. Gaussian weights and biases. In this section, we provide background information and briefly recall the formalism of Karakida et al. (2018) which first computes spectral properties of the Fisher Information of a neural network and then relates it to the maximal stable learning rate.

2.1 FISHER INFORMATION MATRIX AND LEARNING DYNAMICS

Given a data distribution $\mathcal{D}$ over the set of instance-label pairs $\mathcal{X} \times \mathcal{Y}$, a family of parametrized functions $f_\theta : \mathcal{X} \to \mathcal{Y}$ and a loss function $l(f, y)$, our focus will be to ensure convergence of the following gradient descent with momentum update rule:

$$\theta_{t+1} = \theta_t - \eta \nabla L(\theta_t) + \mu(\theta_t - \theta_{t-1}) \ , \tag{1}$$

where $L(\theta)$ is the unobserved population loss,

$$L(\theta) := \mathbb{E}_{(x,y)\sim\mathcal{D}}\big[l(f_\theta(x), y)\big] \ . \tag{2}$$

In practice, the parameters are determined by minimizing an empirical estimate of equation 2 using a stochastic generalization (SGD) of the update rule equation 1. We neglect this difference by always working in the asymptotic limit of large sample size and moreover assuming full-batch gradient updates.

Suppose the loss function can be expressed in terms of a parametric family of positive densities as $l(f_\theta(x), y) =: -\log p_\theta(x, y)$. This assumption holds true for a large class of losses including squared loss and cross-entropy loss. Let I_θ denote the Fisher Information Matrix (FIM) associated with the parametric family induced by the loss,

$$I_\theta := \mathbb{E}_{(x,y)\sim\mathbb{P}_\theta}\left[\nabla_\theta \log p_\theta(x, y) \otimes \nabla_\theta \log p_\theta(x, y)\right] \ , \tag{3}$$

where $\otimes$ denotes Kronecker product and $\mathbb{P}_\theta$ denotes the probability distribution over $\mathcal{X} \times \mathcal{Y}$ with density $p_\theta(x, y)$. Recall that under suitable regularity conditions the following identity holds:

$$I_\theta = -\mathbb{E}_{(x,y)\sim\mathbb{P}_\theta}\left[\operatorname*{Hess}_\theta \log p_\theta(x, y)\right] \ , \tag{4}$$

where Hess_θ denotes the Hessian with respect to θ. The above right-hand side is closely related to the Hessian of the population loss,

$$\mathrm{Hess}\left(L(\theta)\right) = -\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\operatorname*{Hess}_\theta \log p_\theta(x, y)\right] \ , \tag{5}$$

where we interchanged the Hessian with the expectation value. In fact, if we assume that the estimation problem is well-specified so that there exist parameters θ_* such that the data distribution is generated by $\mathbb{P}_{\theta_*} = \mathcal{D}$, then we obtain the following equality between the Hessian of the population loss and the FIM evaluated at the optimal parameters,

$$\mathrm{Hess}\left(L(\theta_*)\right) = I_{\theta_*} \ . \tag{6}$$

If θ is initialized in a sufficiently small neighborhood of θ_*, then by expanding the population loss $L(\theta)$ to quadratic order about θ_* one can show that a necessary condition for convergence is that the step size is bounded from above by (LeCun et al., 2012; Karakida et al., 2018)[1],

$$\eta < \eta_* := \frac{2(1+\mu)}{\lambda_{\max}\big(\mathrm{Hess}(L(\theta_*))\big)} = \frac{2(1+\mu)}{\lambda_{\max}(I_{\theta_*})} \ , \tag{7}$$

where $\lambda_{\max}(M)$ denotes the largest eigenvalue of the matrix M. Rather than computing the optimal parameters θ_* directly, we follow the strategy of Karakida et al. (2018) by estimating the following quantity and arguing that the distribution of the weights and biases is not significantly impacted by the training dynamics,

$$\bar{\lambda}_{\max} := \mathop{\mathbb{E}}_{\theta}\left[\lambda_{\max}\left(I_\theta\right)\right] \ , \tag{8}$$

where $\mathbb{E}_\theta$ denotes the expectation value with respect to the weights and biases. This heuristic was shown to yield a remarkably accurate prediction of the maximal learning rate in (Karakida et al., 2018).

In this paper we adopt the data modeling assumption that the joint density factors as $p_\theta(x,y) = p(x)p_\theta(y\,|\,x)$ where $p(\cdot)$ denotes the probability density of the marginal distribution of the covariates, which is independent of θ. Under this factorization assumption, the FIM simplifies to

$$I_\theta = \mathbb{E}_{(x,y)\sim\mathbb{P}_\theta}\big[\nabla \log p_\theta(y\,|\,x) \otimes \nabla_\theta \log p_\theta(y\,|\,x)\big] \ . \tag{9}$$

Focusing on the Gaussian conditional model $p_\theta(y\,|x) \propto \exp(\frac{1}{2}\|f_\theta(x) - y\|_2^2)$, the FIM further simplifies to

$$I_\theta = \mathbb{E}_{x\sim\mathcal{D}}\big[\nabla_\theta f_\theta(x) \otimes \nabla_\theta f_\theta(x)\big] \ . \tag{10}$$

The family of parametrized functions $f_\theta : \mathbb{R}^{N_0} \to \mathbb{R}^{N_L}$ is chosen to be the family of functions computed by a multi-layer neural network architecture with N_0 input nodes, N_L output nodes and $L \geq 1$ layers. In this paper, we consider neural networks consisting of fully-connected (FC) and convolutional (Conv) layers, with and without batch normalization. The pointwise activation is denoted by σ, which is taken to be the rectified linear unit (ReLU) in this paper. Our analysis can be straightforwardly extended to other architectures and non-linearities. We use $h^l_\theta(x)$ to denote the output of layer l and the input to layer $l+1$. Clearly we have $h^0_\theta(x) = x$ and $h^L_\theta(x) = f_\theta(x)$.

3 Theory

In this section we focus on applying dynamical mean-field theory to study the effect of introducing batch normalization modules into a deep neural network by estimating the largest eigenvalue of the FIM. This estimate, in turn, provides an upper bound on the largest learning rate for which the learning dynamics is stable. This section is structured as follows: We first define various thermodynamic quantities (order parameters, 6 for fully-connected layers and 9 for convolutional layers) that satisfy recursion relations in the mean-field approximation. Then we present an estimate of $\bar{\lambda}_{\max}$ in terms of these order parameters, generalizing a result of Karakida et al. (2018). Using this estimate, we study how $\bar{\lambda}_{\max}$ and η_* are affected by BatchNorm and calculate their dependence on the BatchNorm coefficient γ. Detailed derivations of the order parameters, their recursions, and the associated eigenvalue bound are deferred to the Supplementary Material.

3.1 Fully Connected Layers

A general fully connected layer with input activation $h^l(x)$ and output pre-activation $z^{l+1}(x)$ is described by the affine transformation,

$$z^{l+1}(x) := W^{l+1}h^l(x) + b^{l+1} \ , \tag{11}$$

where $W^{l+1} \in \mathbb{R}^{N_{l+1}\times N_l}$, $b^{l+1} \in \mathbb{R}^{N_{l+1}}$ and N_l denotes the number of units in layer l. In the framework of mean-field theory, we will consider an ensemble of neural networks with Gaussian random weights and biases distributed as follows,

$$[W^{l+1}]_{ij} \sim N(0, \sigma_{\mathrm{w}}^2/N_l) \ , \qquad b^{l+1} \sim N(0, \sigma_{\mathrm{b}}^2\,\mathbb{I}_{N_{l+1}}) \ . \tag{12}$$

[1] In the quadratic approximation to the loss, the optimal learning rate is in fact $\eta_*/2$.

In the case of a standard fully connected layer, the input activation satisfies the recursions $h^l(x) = \sigma(z^l(x))$, where σ denotes the pointwise activation.

A batch-normalized fully connected layer, in contrast, satisfies the following recursion,

$$h^l(x) := \sigma\left(\frac{z^l(x) - \mu^l}{s^l} \odot \gamma_l + \beta_l\right) \ , \tag{13}$$

where $\odot$ denotes the elementwise (Hadamard) product, $\mu^l \in \mathbb{R}^{N_l}$ and $(s^l)^2 := s^l \odot s^l \in \mathbb{R}^{N_l}$ denote the mean and variance of the pre-activation layers with respect to the data distribution,

$$\mu^l := \mathop{\mathbb{E}}_{x}\big[z^l(x)\big] \ , \tag{14}$$

$$(s^l)^2 := \mathop{\mathbb{E}}_{x}\big[(z^l(x) - \mu^l)^2\big] \ . \tag{15}$$

The weights and biases are drawn from the same distributions as in the standard, no BatchNorm, case. In addition, we now have the BatchNorm parameters $\gamma^{l+1}, \beta^{l+1} \in \mathbb{R}^{N_{l+1}}$ which are assumed to be non-random for simplicity. We also fix $\beta_l = 0$

3.1.1 Order Parameters and Their Recursions

To investigate the spectral properties of the FIM, we define the following order parameters,

$$\Gamma_l := \frac{1}{N_l}\mathop{\mathbb{E}}_{\theta}\mathop{\mathbb{E}}_{x\sim\mathcal{D}}\left\|z^l(x)\right\|^2 \ , \qquad \widetilde{\Gamma}_l := \frac{1}{N_l}\mathop{\mathbb{E}}_{\theta}\left\|\mathop{\mathbb{E}}_{x\sim\mathcal{D}} z^l(x)\right\|^2 \ , \tag{16}$$

$$H_l := \frac{1}{N_l}\mathop{\mathbb{E}}_{\theta}\mathop{\mathbb{E}}_{x\sim\mathcal{D}}\left\|h^l(x)\right\|^2 \ , \qquad \widetilde{H}_l := \frac{1}{N_l}\mathop{\mathbb{E}}_{\theta}\left\|\mathop{\mathbb{E}}_{x\sim\mathcal{D}} h^l(x)\right\|^2 \ , \tag{17}$$

$$\Delta_l := \mathop{\mathbb{E}}_{\theta}\mathop{\mathbb{E}}_{x\sim\mathcal{D}}\left\|\delta^l(x)\right\|^2 \ , \qquad \widetilde{\Delta}_l := \mathop{\mathbb{E}}_{\theta}\left\|\mathop{\mathbb{E}}_{x\sim\mathcal{D}} \delta^l(x)\right\|^2 \ , \tag{18}$$

where $\|\cdot\|$ denotes the Euclidean norm and $\delta^l(x) := \frac{\partial f_\theta}{\partial z^l}(x)$. Here we assume that the data x are drawn i.i.d. from a distribution with mean 0 and variance 1, and also that the last layer is linear for classification. We then have the base cases: $H_0 = 0$, $\widetilde{H}_0 = 1$, $\Delta_L = \widetilde{\Delta}_L = 1$. The order parameters in the absence of BatchNorm satisfy the following recursions derived in Karakida et al. (2018),

$$\Gamma_l = \sigma_{\mathrm{b}}^2 + \sigma_{\mathrm{w}}^2 H_{l-1} \ , \qquad \widetilde{\Gamma}_l = \sigma_{\mathrm{b}}^2 + \sigma_{\mathrm{w}}^2 \widetilde{H}_{l-1} \ , \tag{19}$$

$$H_l = \frac{\Gamma_l}{2} \ , \qquad \widetilde{H}_l = \frac{\Gamma_l}{2\pi}\left(\sqrt{1-\tilde{c}^2} + \frac{\tilde{c}\pi}{2} + \tilde{c}\sin^{-1}\tilde{c}\right) \ , \tag{20}$$

$$\Delta_l = \frac{\sigma_{\mathrm{w}}^2}{2}\Delta_{l+1} \ , \qquad \widetilde{\Delta}_l = \frac{\sigma_{\mathrm{w}}^2\widetilde{\Delta}_{l+1}}{2\pi}\left(\frac{\pi}{2} + \sin^{-1}\tilde{c}\right) \ , \tag{21}$$

where $\tilde{c} := \widetilde{\Gamma}_l/\Gamma_l$. In the case of batch normalization we find the following recursions, which are derived in the Supplementary Material,

$$\Gamma_l = \sigma_{\mathrm{b}}^2 + \sigma_{\mathrm{w}}^2 H_{l-1} \ , \qquad \widetilde{\Gamma}_l = \sigma_{\mathrm{b}}^2 + \sigma_{\mathrm{w}}^2 \widetilde{H}_{l-1} \ , \tag{22}$$

$$H_l = \frac{\gamma_l^2}{2} \ , \qquad \widetilde{H}_l = \frac{\gamma_l^2}{2\pi} \ , \tag{23}$$

$$\Delta_l = \frac{\gamma_l^2\sigma_{\mathrm{w}}^2}{2}\frac{\Delta_{l+1}}{\Gamma_l} \ , \qquad \widetilde{\Delta}_l = \frac{\gamma_l^2\sigma_{\mathrm{w}}^2}{4}\frac{\widetilde{\Delta}_{l+1}}{\Gamma_l} \ . \tag{24}$$

3.2 Convolutional Layer

The mean field theory of convolutional layer was first studied by Xiao et al. (2018). In this paper, the results of the preceding section also apply to structured affine transformations including convolutional layers. Let $\mathcal{K}_l$ denote the set of allowable spatial locations of the the lth layer feature map and let $\mathcal{F}_{l+1}$ index the sites of the convolutional kernel applied to that layer. Let C_l denote the number of input channels. The output of a general convolutional layer is of the form,

$$z_\alpha^{l+1}(x) = \sum_{\beta\in\mathcal{F}_{l+1}} W_\beta^{l+1} h_{\alpha+\beta}^l(x) + b^{l+1} \ , \tag{25}$$

where $\alpha \in \mathcal{K}_{l+1}$, $W_\beta^{l+1} \in \mathbb{R}^{C_{l+1} \times C_l}$ and $b^{l+1} \in \mathbb{R}^{C_{l+1}}$. The weights and biases are now distributed as

$$[W_\alpha^{l+1}]_{ij} \sim N(0, \sigma_{\mathrm{w}}^2/N_l) \ , \qquad b^{l+1} \sim N(0, \sigma_{\mathrm{b}}^2 \, \mathbb{I}_{C_{l+1}}) \ . \tag{26}$$

where now $N_l := C_l|\mathcal{F}_{l+1}|$. As in the fully connected case, we consider convolutional layers with both vanilla activation functions of the form $h_\alpha^l(x) := \sigma(z_\alpha^l(x))$ as well as batch normalized convolutional layers, for which the input activations satisfy the recursive identity,

$$h_\alpha^l(x) := \sigma\left(\frac{z_\alpha^l(x) - \mu_\alpha^l}{s_\alpha^l} \odot \gamma_l + \beta_l\right) \ , \tag{27}$$

3.2.1 Order Parameters and Their recursions

Similar to the definitions for fully connected layer, we define the following set of order parameters:

$$\Gamma_l := \frac{1}{C_l} \mathop{\mathbb{E}}_{\alpha} \mathop{\mathbb{E}}_{\theta} \mathop{\mathbb{E}}_{x \sim \mathcal{D}} \left\| z_\alpha^l(x) \right\|^2 \ , \qquad \widetilde{\Gamma}_l := \frac{1}{C_l} \mathop{\mathbb{E}}_{\alpha} \mathop{\mathbb{E}}_{\theta} \left\| \mathop{\mathbb{E}}_{x \sim \mathcal{D}} z_\alpha^l(x) \right\|^2 \ , \tag{28}$$

$$H_l := \frac{1}{C_l} \mathop{\mathbb{E}}_{\alpha} \mathop{\mathbb{E}}_{\theta} \mathop{\mathbb{E}}_{x \sim \mathcal{D}} \left\| h_\alpha^l(x) \right\|^2 \ , \qquad \widetilde{H}_l := \frac{1}{C_l} \mathop{\mathbb{E}}_{\theta} \left\| \mathop{\mathbb{E}}_{x \sim \mathcal{D}} h_\alpha^l(x) \right\|^2 \ , \tag{29}$$

$$\Delta_l := \mathop{\mathbb{E}}_{\alpha} \mathop{\mathbb{E}}_{\theta} \mathop{\mathbb{E}}_{x \sim \mathcal{D}} \left\| \delta_\alpha^l(x) \right\|^2 \ , \qquad \widetilde{\Delta}_l := \mathop{\mathbb{E}}_{\alpha} \mathop{\mathbb{E}}_{\theta} \left\| \mathop{\mathbb{E}}_{x \sim \mathcal{D}} \delta_\alpha^l(x) \right\|^2 \ , \tag{30}$$

$$\widehat{\Gamma}_l := \frac{1}{C_l} \mathop{\mathbb{E}}_{\alpha \neq \beta} \mathop{\mathbb{E}}_{\theta} \left[\mathop{\mathbb{E}}_{x, x' \sim \mathcal{D}} \langle z_\alpha^l(x), z_\beta^l(x') \rangle \right] \ , \quad \widehat{H}_l := \frac{1}{C_l} \mathop{\mathbb{E}}_{\alpha \neq \beta} \mathop{\mathbb{E}}_{\theta} \left[\mathop{\mathbb{E}}_{x, x' \sim \mathcal{D}} \langle h_\alpha^l(x), h_\beta^l(x') \rangle \right] \ , \tag{31}$$

$$\widehat{\Delta}_l := \mathop{\mathbb{E}}_{\alpha \neq \beta} \mathop{\mathbb{E}}_{\theta} \left[\mathop{\mathbb{E}}_{x, x' \sim \mathcal{D}} \langle \delta_\alpha^l(x), \delta_\beta^l(x') \rangle \right] \ , \tag{32}$$

where now $\delta_\alpha^l := \partial f_\theta / \partial z_\alpha^l$ in analogy with the fully connected layer. The expectations over α and β are with respect to the uniform measure over the set of allowed indices. For a standard convolutional layer without BatchNorm, the order parameters can be shown to satisfy the following recursion relations:

$$\Gamma_l = \sigma_{\mathrm{b}}^2 + \sigma_{\mathrm{w}}^2 H_{l-1} \ , \qquad \widetilde{\Gamma}_l = \sigma_{\mathrm{b}}^2 + \sigma_{\mathrm{w}}^2 \widetilde{H}_{l-1} \ , \tag{33}$$

$$H_l = \frac{\Gamma_l}{2} \ , \qquad \widetilde{H}_l = \frac{\Gamma_l}{2\pi} \left(\sqrt{1 - \tilde{c}^2} + \frac{\tilde{c}\pi}{2} + \tilde{c} \sin^{-1} \tilde{c} \right) \ , \tag{34}$$

$$\Delta_l = \frac{\sigma_{\mathrm{w}}^2}{2} \Delta_{l+1} \ , \qquad \widetilde{\Delta}_l = \frac{\sigma_{\mathrm{w}}^2 \widetilde{\Delta}_{l+1}}{2\pi} \left(\frac{\pi}{2} + \sin^{-1} \tilde{c} \right) \ , \tag{35}$$

$$\widehat{\Gamma}_l = \sigma_{\mathrm{b}}^2 + \sigma_{\mathrm{w}}^2 \widehat{H}_{l-1} \ , \qquad \widehat{H}_l = \frac{\Gamma_l}{2\pi} \left(\sqrt{1 - \hat{c}^2} + \frac{\hat{c}\pi}{2} + \hat{c} \sin^{-1} \hat{c} \right) \ , \tag{36}$$

$$\widehat{\Delta}_l = \frac{\sigma_{\mathrm{w}}^2 \widehat{\Delta}_{l+1}}{2\pi} \left(\frac{\pi}{2} + \sin^{-1} \hat{c} \right) \ . \tag{37}$$

In the case of convolutional layers with BatchNorm, the following recursions hold:

$$\Gamma_l = \sigma_{\mathrm{b}}^2 + \sigma_{\mathrm{w}}^2 H_{l-1} \ , \qquad \widetilde{\Gamma}_l = \sigma_{\mathrm{b}}^2 + \sigma_{\mathrm{w}}^2 \widetilde{H}_{l-1} \ , \tag{38}$$

$$H_l = \frac{\gamma_l^2}{2} \ , \qquad \widetilde{H}_l = \frac{\gamma_l^2}{2\pi} \ , \tag{39}$$

$$\Delta_l = \frac{\gamma_l^2 \sigma_{\mathrm{w}}^2}{2} \frac{\Delta_{l+1}}{\Gamma_l} \ , \qquad \widetilde{\Delta}_l = \frac{\gamma_l^2 \sigma_{\mathrm{w}}^2}{4} \frac{\widetilde{\Delta}_{l+1}}{\Gamma_l} \ , \tag{40}$$

$$\widehat{\Gamma}_l = \sigma_{\mathrm{b}}^2 + \sigma_{\mathrm{w}}^2 \widehat{H}_{l-1} \ , \qquad \widehat{H}_l = \frac{\gamma_l^2}{2\pi} \ , \tag{41}$$

$$\widehat{\Delta}_l = \frac{\gamma_l^2 \sigma_{\mathrm{w}}^2}{4} \frac{\widehat{\Delta}_{l+1}}{\Gamma_l} \ , \tag{42}$$

where $\tilde{c} := \widetilde{\Gamma}_l / \Gamma_l$ and $\hat{c} := \widehat{\Gamma}_l / \Gamma_l$. The derivations of the recursion relations for both vanilla and batch-normalized convolutional layers are deferred to the Supplementary Material.

3.3 EIGENVALUE BOUND AND THERMODYNAMIC VARIABLES

The order parameters derived in the previous section are useful because they allow us to gain information about the maximal eigenvalue $\bar{\lambda}_{\max}$ of the FIM. We derived a generalization of (Karakida et al., 2018, Theorem 6) to allow for the inclusion of batch normalization and convolutional layers. In particular, we obtain a lower bound on the maximal eigenvalue $\bar{\lambda}_{\max}$ in terms of the previously introduced order parameters which satisfy the stated recursion relations in the mean-field approximation.

Claim 3.1. *If the layer dimension N_l of the fully connected layers and the number of channels C_l of the convolutional layers satisfy $N_l \gg 1$ and $C_l \gg 1$ for $0 < l < L$, we have,*

$$\bar{\lambda}_{\max} \geq \sum_{l \in [L]} f_l \ , \tag{43}$$

where

$$f_l = \begin{cases} N_{l-1} \widetilde{H}_{l-1} \widetilde{\Delta}_l, & \text{FC} \\ C_{l-1} |\mathcal{F}_l| \left[(|\mathcal{K}_l| - 1) \widehat{\Delta}_l + \widetilde{\Delta}_l \right] \left[(|\mathcal{K}_l| - 1) \widehat{H}_{l-1} + \widetilde{H}_{l-1} \right], & \text{Conv} \end{cases} . \tag{44}$$

The index sets $\mathcal{F}_l$ and $\mathcal{K}_l$ are defined in section 3.2. The order parameters are defined in the previous subsection.

Now we are ready to calculate the lower bound on $\bar{\lambda}_{\max}$ for a given model architecture by calculating the order parameters using their recursions. In the next section, we will focus on the numerical analysis of these recursion relations as well as present experiments that support our calculation.

4 NUMERICAL ANALYSIS AND EXPERIMENTS

In order to understand the effect of BatchNorm on the loss landscape, we theoretically compute $\bar{\lambda}_{\max}$ as a function of the BatchNorm parameter γ, for both fully connected and convolutional architectures (Fig. 1) with and without BatchNorm. For $\gamma \lesssim 3$ (typical for deep network initialization (Ioffe & Szegedy, 2015)) BatchNorm significantly reduces $\bar{\lambda}_{\max}$ compared to the vanilla networks. As a direct consequence of this, the theory predicts that batch normalized networks can be trained using significantly higher learning rates than their vanilla counterparts.

We tested the above theoretical prediction by training the same architectures on MNIST and CIFAR-10 datasets, for different values of η and γ, starting from randomly initialized networks with same variances employed in the mean-field theory calculations. As shown in Fig. 2, the $(\gamma, \log_{10} \eta)$-plane clearly partitions into distinct phases characterized by convergent and non-convergent optimization dynamics, and our theoretically predicted upper bound η_* closely agrees with the experimentally determined phase boundary. The experiment of vanilla network is shown in 6.4 as a baseline.

In addition to the striking match between our theoretical prediction and the experimentally determined phase boundaries, the experimental results also suggest a tendency for smaller γ-initiations to produce lower values of test loss after a fixed number of epochs, i.e. faster convergence. We leave detailed investigation of this initialization scheme to future work. Also, dark strips can be observed in the heatmaps indicate the optimal learning rates for optimization, which is around $\eta_*/2$ and consistent with LeCun et al. (2012) in the quadratic approximation to the loss. Our analysis also suggests that small γ initialization benefits the convergence of training. Additional experiments supporting this intuition can be found in Section 6.5 of Supplementary Material.

The architectural design for our experiments is as follows. In the fully connected architecture, we choose $L = 4$ layers with $N_l = 1000$ hidden units per layer except the final (linear) layer which has $N_L = 10$ outputs. Batch normalization is applied after each linear operation except for the final linear output layer. The convolutional network has a similar structure with $L = 4$ layers. The first three are convolutional layers with filter size 3, stride 2, and number of channels $C_1 = 30$, $C_2 = 60$, $C_3 = 90$. The final layer is a fully connected output layer to perform classification. The other architectural/optimization hyperparameters were chosen to be $\sigma_w^2 = 2$, $\sigma_b^2 = 0.5$, $\beta = 0$ and $\mu = 0.9$. Momentum μ here was set to be 0.9 to match the value frequently used in practice, which only affects the dependency of η_* on FIM.

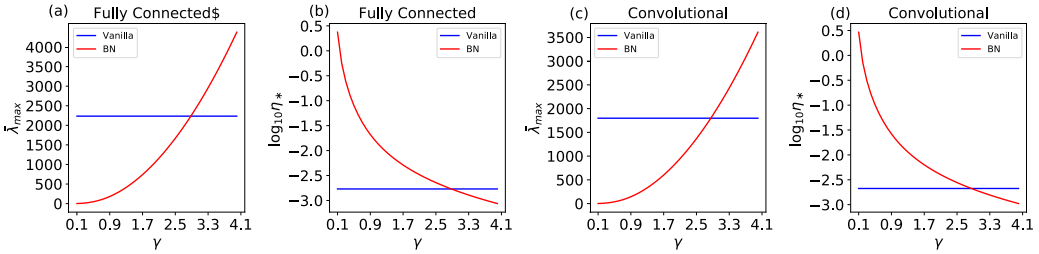

Figure 1: The maximum eigenvalue $\bar{\lambda}_{\max}$ and associated critical learning rate η_* for vanilla (blue) and BatchNorm networks (red) as a function of the BatchNorm parameter γ for different choices of architecture (fully-connected and convolutional), calculated by theory. (a, c) shows the flattening effect of BatchNorm on the loss function for a wide range of hyperparameters and (b, d) further show that for sufficiently small γ BatchNorm enables optimization with much higher learning rate than vanilla networks.

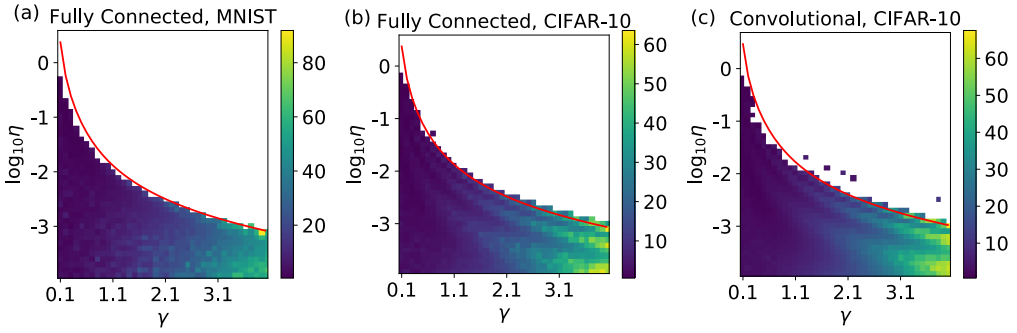

Figure 2: Heatmaps showing test loss as a function of $(\log_{10}\eta, \gamma)$ after 5 epochs of training for different choices of dataset and architecture. Results were obtained by averaging 5 random restarts. The white region indicates parameter explosion for at least one of the runs. The red line shows the theoretical prediction for the maximal learning rate η_*. The dark band on the heatmaps for CIFAR-10 approximately tracks the optimal learning rate $\eta_*/2$ in the quadratic approximation to the loss. Note the log scale for the learning rate, so the theory matches the experiments over three orders of magnitude for η.

5 Conclusion and Future Work

In this paper, we studied the impact of BatchNorm on the loss surface of multi-layer neural networks and its implication for training dynamics. By developing recursion relations for the relevant order parameters, the maximum eigenvalue of the Fisher Information matrix $\bar{\lambda}_{\max}$ can be estimated and related to the maximal learning rate. The theory correctly predicts that adding BatchNorm with small γ allows the training algorithm to exploit much larger learning rates, which speeds up convergence. The experiments also suggest that using a smaller γ results in a lower test loss for a fixed number of training epochs. This suggests that initialization with smaller γ may help the optimization process in deep learning models, which will be interesting for future study.

The close agreement between theoretical predictions and the experimentally determined phase boundaries strongly supports the validity of our analysis, despite the non-rigorous nature of the derivations. Although similar approaches have been used in other work (Poole et al., 2016; Schoenholz et al., 2017; Yang & Schoenholz, 2017; Xiao et al., 2018; Karakida et al., 2018), we hope that future work will place these results on a firmer mathematical footing. Furthermore, our BatchNorm analysis is not limited to the convolutional and fully-connected architectures we considered in this paper and can be extended to arbitrary feedforward architectures such as ResNets.

ACKNOWLEDGMENTS

This work was supported by National Science Foundation PHY-1734030 (DJS) and by the Simons Foundation program for the MMLS (DJS).

REFERENCES

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. *arXiv preprint arXiv:1603.05027*, 2016a.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016b.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Ryo Karakida, Shotaro Akaho, and Shun ichi Amari. Universal statistics of fisher information in deep neural networks: Mean field approach. *arXiv preprint arXiv:1806.01316*, 2018.

Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. pp. 10971105, 2012.

Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pp. 9–48. Springer, 2012.

Ben Poole, Subhaneil Lahiri, Maithra Raghu, Jascha Sohl-Dickstein, and Surya Ganguli. Exponential expressivity in deep neural networks through transient chaos. pp. 3360–3368, 2016.

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. 2015.

Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? (no, it is not about internal covariate shift). *arXiv preprint arXiv:1805.11604*, 2018.

Samuel S Schoenholz, Justin Gilmer, Surya Ganguli, and Jascha Sohl-Dickstein. Deep information propagation. In *International Conference on Learning Representations (ICLR)*, 2017.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Haim Sompolinsky and Annette Zippelius. Relaxational dynamics of the edwards-anderson model and the mean-field theory of spin-glasses. *Physical Review B*, 25(11):6860, 1982.

Lechao Xiao, Yasaman Bahri, Jascha Sohl-Dickstein, Samuel S. Schoenholz, and Jeffrey Pennington. Dynamical isometry and a mean field theory of cnns: How to train 10,000-layer vanilla convolutional neural networks. In *International Conference on Machine Learning (ICML)*, 2018.

Greg Yang and Samuel S. Schoenholz. Mean field residual networks: On the edge of chaos. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.

6 SUPPLEMENTARY MATERIAL

This section provides non-rigorous derivations of the order parameters, their recursions, and the associated eigenvalue bound. Despite the non-rigorous nature of these calculations, we remark that similar reasoning has been successfully used in a number of related works on mean-field theory, demonstrating impressive agreement with experiments (Poole et al., 2016; Schoenholz et al., 2017; Yang & Schoenholz, 2017; Xiao et al., 2018; Karakida et al., 2018).

6.1 RECURSIONS FOR FULLY CONNECTED LAYERS

Claim 6.1. *The forward recursions for* $0 \leq l \leq L-1$ *are* $\Gamma_{l+1} = \sigma_{\rm b}^2 + \sigma_{\rm w}^2 H_l$ *and* $\widetilde{\Gamma}_{l+1} = \sigma_{\rm b}^2 + \sigma_{\rm w}^2 \widetilde{H}_l$ *where* $H_l = \gamma_l^2/2$ *and* $\widetilde{H}_l = \gamma_l^2/(2\pi)$ *for* $l \in [L-1]$ *whereas* $H_0 = 1$ *and* $\widetilde{H}_0 = 0$.

Derivation. In general, we have

$$\frac{1}{N_{l+1}} \mathop{\mathbb{E}}_{\theta} \langle z^{l+1}(x), z^{l+1}(x') \rangle = \sigma_{\rm b}^2 + \frac{\sigma_{\rm w}^2}{N_l} \langle h_\theta^l(x), h_\theta^l(x') \rangle \ . \tag{45}$$

Thus, setting $l = 0$ we obtain $q^1 = \sigma_{\rm w}^2 + \sigma_{\rm b}^2$ if we assume $\mathop{\mathbb{E}}_{x} \|x\|^2 = N_0$, and $\widetilde{\Gamma}_1 = \sigma_{\rm b}^2$ since $\mathop{\mathbb{E}}_{x,x'} \langle x, x' \rangle = \langle \mathbb{E}\, x, \mathbb{E}\, x' \rangle = 0$.

Recall (for $l > 0$) $z^{l+1}(x) = W^{l+1} \sigma_l\big(u^l(x) \odot \gamma_l\big) + b^{l+1}$ where $u^l(x) = [z^l(x) - \mu^l]/s^l$ so

$$\frac{1}{N_{l+1}} \mathop{\mathbb{E}}_{\theta} \langle z^{l+1}(x), z^{l+1}(x') \rangle = \sigma_{\rm b}^2 + \frac{\sigma_{\rm w}^2}{N_l} \left\langle \sigma_l\big(u^l(x) \odot \gamma_l\big), \sigma_l\big(u^l(x') \odot \gamma_l\big) \right\rangle \ . \tag{46}$$

Therefore, setting $x = x'$ and taking expectation values over x gives,

$$\Gamma_{l+1} := \frac{1}{N_{l+1}} \mathop{\mathbb{E}}_{x,\theta} \|z^{l+1}(x)\|^2 \ , \tag{47}$$

$$= \sigma_{\rm b}^2 + \frac{\sigma_{\rm w}^2}{N_l} \mathop{\mathbb{E}}_{x,\theta} \|h^l(x)\|^2 \ , \tag{48}$$

$$= \sigma_{\rm b}^2 + \sigma_{\rm w}^2 \, H_l \ , \tag{49}$$

$$H_l := \frac{1}{N_l} \mathop{\mathbb{E}}_{x,\theta} \|h^l(x)\|^2 \ , \tag{50}$$

$$= \sigma_{\rm w}^2 \mathop{\mathbb{E}}_{x,\theta} \sigma_l^2\big(\gamma_l \, u^l(x)[1]\big) \ , \tag{51}$$

$$\simeq \sigma_{\rm w}^2 \int Dz \, \sigma_l^2 \left(\gamma_l \, z\right) \ , \tag{52}$$

$$= \frac{\gamma_l^2}{2} \ , \tag{53}$$

where we have approximated each component of the random vector $u^l(x)$ as a standard Gaussian.

Similarly, taking expectations over x, x' gives

$$\widetilde{\Gamma}_{l+1} = \sigma_{\rm b}^2 + \sigma_{\rm w}^2 \mathop{\mathbb{E}}_{x,x',\theta} \sigma_l\big(u^l(x)[1] \, \gamma_l\big) \sigma_l\big(u^l(x')[1] \, \gamma_l\big) \ . \tag{54}$$

Consider the approximation in which the random pair $(u^l(x)[1], u^l(x')[1])$ is Gaussian distributed with zero mean and covariance,

$$\Sigma^l := \begin{pmatrix} \Sigma^l_{xx} & \Sigma^l_{xx'} \\ \Sigma^l_{xx'} & \Sigma^l_{x'x'} \end{pmatrix} := \begin{pmatrix} \mathbb{E}\, u^l(x)[1]^2 & \mathbb{E}\, u^l(x)[1] u^l(x')[1] \\ \mathbb{E}\, u^l(x)[1] u^l(x')[1] & \mathbb{E}\, u^l(x')[1]^2 \end{pmatrix} \ . \tag{55}$$

Then the recursion becomes

$$\widetilde{\Gamma}_{l+1} \simeq \sigma_{\rm b}^2 + \sigma_{\rm w}^2 \int Dz_1 \, Dz_2 \, \sigma_l\Big(\sqrt{\Sigma^l_{xx}} \, z_1 \gamma_l\Big) \sigma_l \left[\sqrt{\Sigma^l_{yy}} \left(\tilde{c}_l \, z_1 + \sqrt{1 - (\tilde{c}_l)^2} \, z_2\right) \gamma_l\right] \ , \tag{56}$$

where $\tilde{c}_l := \Sigma^l_{xx'}/\sqrt{\Sigma^l_{xx}\Sigma^l_{x'x'}}$. Observe that by independence of x and y,

$$\mathop{\mathbb{E}}_{x,x'}\big[u_l(x)u_l(x')\big] = \mathop{\mathbb{E}}_{x,x'}\left[\frac{z_l(x)z_l(x') + \mu_l^2 - \mu_l\big(z_l(x) + z_l(x')\big)}{s_l^2}\right] \ , \tag{57}$$

$$= \frac{\mathbb{E}_x z_l(x)\mathbb{E}_y z_l(x') + \mu_l^2 - \mu_l\big(\mathbb{E}_x z_l(x) + \mathbb{E}_y z_l(x')\big)}{s_l^2} \ , \tag{58}$$

$$= 0 \ . \tag{59}$$

Thus $\Sigma^l_{xx'} = 0$ and consequently,

$$\widetilde{\Gamma}_{l+1} = \sigma_{\mathrm{b}}^2 + \sigma_{\mathrm{w}}^2 \widetilde{H}_l \ , \tag{60}$$

$$\widetilde{H}_l = \frac{\gamma_l^2}{2\pi} \ . \tag{61}$$

□

The derivation here assumes an infinitely large dataset. For a dataset of size m, an error is introduced from the non-zero ratio of $m/m_{x\neq x'}$ where $m_{x\neq x'}$ denotes the total number of sample pairs (x, x') where $x \neq x'$. We have $m_{x\neq x'} = m^2 - m$ and therefore the error is $O(1/m)$ which is negligible for most of the frequently-used datasets.

Claim 6.2. *The backward recursions for $l \in [1, L-1]$ are $\Delta_l = \frac{\gamma_l^2\sigma_{\mathrm{w}}^2}{2}\frac{\Delta_{l+1}}{\Gamma_l}$ and $\widetilde{\Delta}_l = \frac{\gamma_l^2\sigma_{\mathrm{w}}^2}{4}\frac{\widetilde{\Delta}_{l+1}}{\Gamma_l}$ with base cases $\Delta_L = \widetilde{\Delta}_L = 1$.*

Derivation. For each layer $l \in [L]$ and for each unit $i \in [N_l]$ define $\delta_l[i] \in \mathbb{R}^{N_L}$ by

$$\delta^l[i] := \frac{\partial h_\theta^L}{\partial z^l[i]} \ . \tag{62}$$

In particular, since we assume linear output $\delta^L[i] = e_i$ and thus

$$\Delta_L = \mathop{\mathbb{E}}_{x,\theta}\|\delta^L[\cdot]\|^2 \ = \mathbf{1} \ , \tag{63}$$

where $\mathbf{1} \in \mathbb{R}^{N_L}$ is the vector of ones. For ease of presentation and without loss of generality we restrict to the first component $\Delta_L[1] = 1$ and abuse notation by writing $\Delta_L = 1$. For $l < L$ we have by the chain rule,

$$\delta^l[i] = \sum_{k\in[N_{l+1}]} \frac{\partial z^{l+1}[k]}{\partial z^l[i]}\delta^{l+1}[k] \ , \tag{64}$$

$$= \frac{\gamma_l[i]}{s^l[i]}\sigma_l'\left(u^l[i]\,\gamma_l[i]\right)\sum_{k\in[N_{l+1}]}[W^{l+1}]_{ki}\delta^{l+1}[k] \ . \tag{65}$$

Hence,

$$\delta^l(x)[i]\delta^l(x') = \frac{\gamma_l[i]^2}{s^l[i]^2}\sigma_l'\left(u^l(x)[i]\,\gamma_l[i]\right)\sigma_l'\left(u^l(x')[i]\,\gamma_l[i]\right)\sum_{k,k'\in[N_{l+1}]}[W^{l+1}]_{ki}[W^{l+1}]_{k'i}\delta^{l+1}[k]\delta^{l+1}[k'] \ . \tag{66}$$

Following the usual assumption that the back-propagated gradient is independent of the forward signal we obtain,

$$\mathop{\mathbb{E}}_{\theta}\langle\delta^l(x), \delta^l(x')\rangle = \frac{\sigma_{\mathrm{w}}^2}{N_l}\sum_{i\in[N_l]}\mathop{\mathbb{E}}_{\theta}\left[\frac{\gamma_l[i]^2}{s^l[i]^2}\sigma_l'\left(u^l(x)[i]\,\gamma_l[i]\right)\sigma_l'\left(u^l(x')[i]\,\gamma_l[i]\right)\right]\mathop{\mathbb{E}}_{\theta}\langle\delta^{l+1}(x), \delta^{l+1}(x')\rangle \tag{67}$$

Setting $x = x'$ and taking expectation values over x we obtain,

$$\Delta_l := \mathop{\mathbb{E}}_{x,\theta} \|\delta^l(x)\|^2 \ , \tag{68}$$

$$\simeq \frac{\Delta_{l+1}}{\Gamma_l} \gamma_l^2 \sigma_{\rm w}^2 \int Dz\, \sigma_l'(\gamma_l z)^2 \ , \tag{69}$$

$$= \frac{\gamma_l^2 \sigma_{\rm w}^2}{2} \frac{\Delta_{l+1}}{\Gamma_l} \ . \tag{70}$$

Similarly, taking expectation values over x, x' we obtain,

$$\widetilde{\Delta}_l = \mathop{\mathbb{E}}_{x,x',\theta} \langle \delta^l(x), \delta^l(x') \rangle \ , \tag{71}$$

$$= \sigma_{\rm w}^2 \gamma_l^2 \frac{\widetilde{\Delta}_{l+1}}{\Gamma_l} \int Dz_1 Dz_2 \sigma_l'\Big(\sqrt{\Sigma^l_{xx}} z_1\, \gamma_l\Big) \sigma_l' \left[\sqrt{\Sigma^l_{yy}} \left(\tilde{c}_l\, z_1 + \sqrt{1-(\tilde{c}_l)^2}\, z_2\right) \gamma_l\right] \ , \tag{72}$$

$$= \frac{\sigma_{\rm w}^2 \gamma_l^2}{4} \frac{\widetilde{\Delta}_{l+1}}{\Gamma_l} \ . \tag{73}$$

□

6.2 Recursions for convolutional layers

Recall that the output of a general convolutional layer is of the form,

$$z_\alpha^{l+1}(x) = \sum_{\beta \in \mathcal{F}_{l+1}} W_\beta^{l+1} h^l_{\alpha+\beta}(x) + b^{l+1} \ , \tag{74}$$

where $\alpha \in \mathcal{K}_{l+1}$. As a concrete example, consider CIFAR-10 input of dimension $32 \times 32 \times 3$ which is mapped by a convolutional layer with 3×3 kernels, stride 2 and 20 output channels and no padding. Then $C_0 = 3$, $C_1 = 20$, $|\mathcal{K}_0| = 1024$ and $|\mathcal{F}_1| = 9$ and $|\mathcal{K}_1| = 225$.

We begin by deriving some useful identities for convolutional layers, before specializing to the batch-normalized and vanilla networks. For each channel $i \in [C_l]$, we have,

$$\mathop{\mathbb{E}}_{\theta} z_\alpha^{l+1}(x)[i] z_\beta^{l+1}(x')[i] = \sigma_{\rm b}^2 + \mathop{\mathbb{E}}_{\theta} \sum_{\substack{(\beta_1, j_1) \in \mathcal{F}_{l+1} \times [C_l] \\ (\beta_2, j_2) \in \mathcal{F}_{l+1} \times [C_l]}} [W^{l+1}_{\beta_1}]_{ij_1} [W^{l+1}_{\beta_2}]_{ij_2} h^l_{\alpha+\beta_1}(x)[j_1] h^l_{\beta+\beta_2}(x')[j_2] \ ,$$

$$= \sigma_{\rm b}^2 + \frac{\sigma_{\rm w}^2}{N_l} \sum_{\delta \in \mathcal{F}_{l+1}} \mathop{\mathbb{E}}_{\theta} \left\langle h^l_{\alpha+\delta}(x), h^l_{\beta+\delta}(x') \right\rangle \ . \tag{75}$$

Hence,

$$\frac{1}{C_{l+1}} \mathop{\mathbb{E}}_{\theta} \left\langle z_\alpha^{l+1}(x), z_\beta^{l+1}(x') \right\rangle = \sigma_{\rm b}^2 + \frac{\sigma_{\rm w}^2}{N_l} \sum_{\delta \in \mathcal{F}_{l+1}} \mathop{\mathbb{E}}_{\theta} \left\langle h^l_{\alpha+\delta}(x), h^l_{\beta+\delta}(x') \right\rangle \ , \tag{76}$$

where $N_l := C_l |\mathcal{F}_{l+1}|$.

Moreover, let us introduce the following shorthand,

$$\mathop{\mathbb{E}}_{\alpha} \|h^l_\alpha(x)\|^2 := \frac{1}{|\mathcal{K}_l|} \sum_{\alpha \in \mathcal{K}_l} \|h^l_\alpha(x)\|^2 \ , \tag{77}$$

$$\mathop{\mathbb{E}}_{\alpha} \langle h^l_\alpha(x), h^l_\alpha(x') \rangle := \frac{1}{|\mathcal{K}_l|} \sum_{\alpha \in \mathcal{K}_l} \left\langle h^l_\alpha(x), h^l_\alpha(x') \right\rangle \ , \tag{78}$$

$$\mathop{\mathbb{E}}_{\alpha \neq \beta} \left\langle h^l_\alpha(x), h^l_\beta(x') \right\rangle := \frac{1}{|\mathcal{K}_l|(|\mathcal{K}_l| - 1)} \sum_{\alpha, \beta \in \mathcal{K}_l \times \mathcal{K}_l : \alpha \neq \beta} \left\langle h^l_\alpha(x), h^l_\beta(x') \right\rangle \tag{79}$$

Then we can write the recursion relations as,

$$\Gamma_{l+1} := \frac{1}{C_{l+1}} \mathop{\mathbb{E}}_{\alpha} \mathop{\mathbb{E}}_{x,\theta} \|z_\alpha^{l+1}(x)\|^2 \ , \tag{80}$$

$$= \mathop{\mathbb{E}}_{x} \left[\sigma_{\rm b}^2 + \frac{\sigma_{\rm w}^2}{N_l} \sum_{\delta \in \mathcal{F}_{l+1}} \mathop{\mathbb{E}}_{\theta} \mathop{\mathbb{E}}_{\alpha} \|h_{\alpha+\delta}^l(x)\|^2 \right] \ , \tag{81}$$

$$= \sigma_{\rm b}^2 + \frac{\sigma_{\rm w}^2}{C_l} \mathop{\mathbb{E}}_{x,\theta} \mathop{\mathbb{E}}_{\alpha} \|h_\alpha^l(x)\|^2 \ , \tag{82}$$

$$= \sigma_{\rm b}^2 + \sigma_{\rm w}^2 \, H_l \ , \tag{83}$$

$$H_l := \frac{1}{C_l} \mathop{\mathbb{E}}_{x,\theta} \mathop{\mathbb{E}}_{\alpha} \|h_\alpha^l(x)\|^2 \ . \tag{84}$$

Furthermore we have,

$$\widetilde{\Gamma}_{l+1} = \frac{1}{C_{l+1}} \mathop{\mathbb{E}}_{\alpha} \mathop{\mathbb{E}}_{x,x',\theta} \langle z_\alpha^{l+1}(x), z_\alpha^{l+1}(x') \rangle \ , \tag{85}$$

$$= \mathop{\mathbb{E}}_{x,x'} \left[\sigma_{\rm b}^2 + \frac{\sigma_{\rm w}^2}{N_l} \sum_{\delta \in \mathcal{F}_{l+1}} \mathop{\mathbb{E}}_{\theta} \mathop{\mathbb{E}}_{\alpha} \langle h_{\alpha+\delta}^l(x), h_{\alpha+\delta}^l(x') \rangle \right] \ , \tag{86}$$

$$= \sigma_{\rm b}^2 + \sigma_{\rm w}^2 \, \widetilde{H}_l \ , \tag{87}$$

$$\widetilde{H}_l := \frac{1}{C_l} \mathop{\mathbb{E}}_{\alpha} \mathop{\mathbb{E}}_{x,x',\theta} \langle h_\alpha^l(x), h_\alpha^l(x') \rangle \ . \tag{88}$$

Finally, we have,

$$\widehat{\Gamma}_{l+1} := \mathop{\mathbb{E}}_{\alpha \neq \beta} \left[\mathop{\mathbb{E}}_{x,x',\theta} \langle z_\alpha^{l+1}(x), z_\beta^{l+1}(x') \rangle \right] \ , \tag{89}$$

$$= \mathop{\mathbb{E}}_{x,x'} \left[\sigma_{\rm b}^2 + \frac{\sigma_{\rm w}^2}{N_l} \sum_{\delta \in \mathcal{F}_{l+1}} \mathop{\mathbb{E}}_{\theta} \mathop{\mathbb{E}}_{\alpha \neq \beta} \langle h_{\beta+\delta}^l(x), h_{\alpha+\delta}^l(x') \rangle \right] \ , \tag{90}$$

$$= \mathop{\mathbb{E}}_{x,x'} \left[\sigma_{\rm b}^2 + \frac{\sigma_{\rm w}^2}{C_l} \mathop{\mathbb{E}}_{\theta} \mathop{\mathbb{E}}_{\alpha \neq \beta} \langle h_\beta^l(x), h_\alpha^l(x') \rangle \right] \ , \tag{91}$$

$$= \sigma_{\rm b}^2 + \sigma_{\rm w}^2 \widehat{H}_l \ , \tag{92}$$

$$\widehat{H}_l := \frac{1}{C_l} \mathop{\mathbb{E}}_{\alpha \neq \beta} \mathop{\mathbb{E}}_{x,x',\theta} \langle h_\alpha^l(x), h_\beta^l(x') \rangle \ , \tag{93}$$

where we used,

$$\sum_{\delta \in \mathcal{F}_{l+1}} \sum_{\alpha,\beta \in \mathcal{K}_l \times \mathcal{K}_l : \alpha \neq \beta} \left\langle h_{\alpha+\delta}^l(x), h_{\beta+\delta}^l(x') \right\rangle = \tag{94}$$

$$\sum_{\delta \in \mathcal{F}_{l+1}} \left[\left\langle \sum_{\alpha \in \mathcal{K}_l} h_{\alpha+\delta}^l(x), \sum_{\beta \in \mathcal{K}_l} h_{\beta+\delta}^l(x') \right\rangle - \sum_{\alpha \in \mathcal{K}_l} \left\langle h_{\alpha+\delta}^l(x), h_{\alpha+\delta}^l(x') \right\rangle \right] \tag{95}$$

$$= |\mathcal{F}_{l+1}| \left[\left\langle \sum_{\alpha \in \mathcal{K}_l} h_\alpha^l(x), \sum_{\beta \in \mathcal{K}_l} h_\beta^l(x') \right\rangle - \sum_{\alpha \in \mathcal{K}_l} \left\langle h_\alpha^l(x), h_\alpha^l(x') \right\rangle \right] \tag{96}$$

$$= |\mathcal{F}_{l+1}| \sum_{\alpha,\beta \in \mathcal{K}_l \times \mathcal{K}_l : \alpha \neq \beta} \left\langle h_\alpha^l(x), h_\beta^l(x') \right\rangle \ . \tag{97}$$

6.2.1 BATCH NORMALIZATION

Claim 6.3. *The forward recursions for* $0 \leq l \leq L-1$ *are* $\Gamma_{l+1} = \sigma_{\rm b}^2 + \sigma_{\rm w}^2 H_l$, $\widetilde{\Gamma}_{l+1} = \sigma_{\rm b}^2 + \sigma_{\rm w}^2 \widetilde{H}_l$ *and* $\widehat{\Gamma}_{l+1} = \sigma_{\rm b}^2 + \sigma_{\rm w}^2 \widehat{H}_l$ *where for each* $l \in [L-1]$ *we have* $H_l = \gamma_l^2/2$ *and* $\widetilde{H}_l = \widehat{H}_l = \gamma_l^2/(2\pi)$.

Derivation. Recalling that $h^l_\alpha(x) = \sigma_l\big(u^l_\alpha(x) \odot \gamma_l\big)$ where $u^l_\alpha(x) := \frac{z^l_\alpha(x) - \mu^l_\alpha}{s^l_\alpha}$ and substituting into equation 84, equation 88 and equation 93 we obtain,

$$H_l \simeq \int Dz\, \sigma_l^2(\gamma_l z) \ , \tag{98}$$

$$= \frac{\gamma_l^2}{2} \ , \tag{99}$$

$$\widetilde{H}_l \simeq \int Dz_1\, Dz_2\, \sigma_l(\gamma_l z_1)\sigma_l(\gamma_1 z_2) \ , \tag{100}$$

$$= \frac{\gamma_l^2}{2\pi} \ , \tag{101}$$

$$\widehat{H}_l \simeq \int Dz_1\, Dz_2\, \sigma_l(\gamma_l z_1)\sigma_l(\gamma_1 z_2) \ , \tag{102}$$

$$= \frac{\gamma_l^2}{2\pi} \ , \tag{103}$$

□

Claim 6.4. *The backward recursions are* $\Delta_l = \frac{\sigma_w^2 \gamma_l^2 \Delta_{l+1}}{2\Gamma_l}$, $\widetilde{\Delta}_l = \frac{\sigma_w^2 \gamma_l^2 \widetilde{\Delta}_{l+1}}{4\Gamma_l}$, *and* $\widehat{\Delta}_l = \frac{\sigma_w^2 \gamma_l^2 \widehat{\Delta}_{l+1}}{4\Gamma_l}$.

In order to derive the backward recursion, define (for each $l \in [L]$, $i \in [C_l]$, $\alpha \in \mathcal{K}_l$)

$$\delta^l_\alpha[i] = \frac{\partial h^L_\theta}{\partial z^l_\alpha[i]} \ . \tag{104}$$

Then by the chain rule,

$$\delta^l_\alpha[i] = \sum_{(\beta,k) \in \mathcal{K}_{l+1} \times [C_{l+1}]} \frac{\partial z^{l+1}_\beta[k]}{\partial z^l_\alpha[i]} \delta^{l+1}_\beta[k] \ . \tag{105}$$

Now,

$$z^{l+1}_\beta[k] = \sum_{(\beta',j) \in \mathcal{F}_{l+1} \times [C_l]} [W^{l+1}_{\beta'}]_{kj} \sigma_l\big(u^l_{\beta+\beta'}[j]\, \gamma_l[j]\big) + b^{l+1}[k] \ , \tag{106}$$

$$\frac{\partial z^{l+1}_\beta[k]}{\partial z^l_\alpha[i]} = \sum_{(\beta',j) \in \mathcal{F}_{l+1} \times [C_l]} [W^{l+1}_{\beta'}]_{kj} \sigma_l'\big(u^l_{\beta+\beta'}[j]\, \gamma_l[j]\big) \frac{\gamma_l[j]}{s^l_{\beta+\beta'}[j]} \delta_{ij} \delta_{\alpha,\beta+\beta'} \ , \tag{107}$$

$$= [W^{l+1}_{\alpha-\beta}]_{ki} \sigma_l'\big(u^l_\alpha[i]\, \gamma_l[i]\big) \frac{\gamma_l[i]}{s^l_\alpha[i]} \ . \tag{108}$$

Thus,

$$\delta^l_\alpha[i] = \frac{\gamma_l[i]}{s^l_\alpha[i]} \sigma_l'\left(u^l_\alpha[i]\, \gamma_l[i]\right) \sum_{(\beta,k) \in \mathcal{F}_{l+1} \times [C_{l+1}]} [W^{l+1}_\beta]_{ki} \delta^{l+1}_{\alpha-\beta}[k] \ . \tag{109}$$

By the distributional assumption on the weights, for each $(\alpha, \beta) \in \mathcal{K}_l \times \mathcal{K}_l$ we have,

$$\mathop{\mathbb{E}}_\theta \sum_{\substack{(\beta_1,k_1) \in \mathcal{F}_{l+1} \times [C_{l+1}] \\ (\beta_2,k_2) \in \mathcal{F}_{l+1} \times [C_{l+1}]}} [W^{l+1}_{\beta_1}]_{k_1 i} [W^{l+1}_{\beta_2}]_{k_2 i} \delta^{l+1}_{\alpha-\beta_1}[k_1] \delta^{l+1}_{\beta-\beta_2}[k_2] = \frac{\sigma_w^2}{N_l} \sum_{\delta \in \mathcal{F}_{l+1}} \mathop{\mathbb{E}}_\theta \left\langle \delta^{l+1}_{\alpha-\delta}(x), \delta^{l+1}_{\beta-\delta}(x') \right\rangle \ . \tag{110}$$

Thus, under the usual independence assumptions,

$$\mathop{\mathbb{E}}_\theta \left\langle \delta^l_\alpha(x), \delta^l_\beta(x') \right\rangle = \frac{\sigma_w^2}{N_l} \sum_{i \in [C_l]} \mathop{\mathbb{E}}_\theta \left[\frac{\gamma_l[i]^2}{s^l_\alpha[i] s^l_\beta[i]} \sigma_l'\left(u^l_\alpha(x)[i]\, \gamma_l[i]\right) \sigma_l'\left(u^l_\beta(x')[i]\, \gamma_l[i]\right) \right] \sum_{\delta \in \mathcal{F}_{l+1}} \mathop{\mathbb{E}}_\theta \left\langle \delta^{l+1}_{\alpha-\delta}(x), \delta^{l+1}_{\beta-\delta}(x') \right\rangle \ . \tag{111}$$

Setting $\alpha = \beta$, $x = x'$, averaging over α and taking the expectation value over x,

$$\Delta_l := \mathop{\mathbb{E}}_{\alpha} \mathop{\mathbb{E}}_{x,\theta} \|\delta^l_\alpha(x)\|^2 \ , \tag{112}$$

$$= \frac{\sigma_{\rm w}^2}{N_l} \frac{\gamma_l^2}{2\Gamma_l} C_l \mathop{\mathbb{E}}_{x,\theta} \sum_{\delta \in \mathcal{F}_{l+1}} \mathop{\mathbb{E}}_{\alpha} \|\delta^l_{\alpha-\delta}(x)\|^2 \ , \tag{113}$$

$$= \frac{\sigma_{\rm w}^2}{N_l} \frac{\gamma_l^2}{2\Gamma_l} C_l |\mathcal{F}_{l+1}| \mathop{\mathbb{E}}_{\alpha} \mathop{\mathbb{E}}_{x,\theta} \|\delta^l_\alpha(x)\|^2 \ , \tag{114}$$

$$= \frac{\sigma_{\rm w}^2 \gamma_l^2 \Delta_{l+1}}{2\Gamma_l} \ . \tag{115}$$

Similarly, setting $\alpha = \beta$, averaging over α and taking expectation values over x, x' gives

$$\widetilde{\Delta}_l := \mathop{\mathbb{E}}_{\alpha} \mathop{\mathbb{E}}_{x,x',\theta} \langle \delta^l_\alpha(x), \delta^l_\alpha(x') \rangle \ , \tag{116}$$

$$= \frac{\sigma_{\rm w}^2 \gamma_l^2 \Delta_{l+1}}{4\Gamma_l} \ . \tag{117}$$

Now,

$$\sum_{\substack{\alpha,\beta \in \mathcal{K}_{l+1} \times \mathcal{K}_{l+1} : \alpha \neq \beta \\ \delta \in \mathcal{F}_{l+1}}} \langle \delta^{l+1}_{\alpha-\delta}(x), \delta^{l+1}_{\beta-\delta}(x') \rangle \tag{118}$$

$$= \sum_{\delta \in \mathcal{F}_{l+1}} \left[\left\langle \sum_{\alpha \in \mathcal{K}_{l+1}} \delta^{l+1}_{\alpha-\delta}(x), \sum_{\beta \in \mathcal{K}_{l+1}} \delta^{l+1}_{\beta-\delta}(x') \right\rangle - \sum_{\alpha \in \mathcal{K}_{l+1}} \langle \delta^{l+1}_{\alpha-\delta}(x), \delta^{l+1}_{\alpha-\delta}(x') \rangle \right] \ , \tag{119}$$

$$= |\mathcal{F}_{l+1}| \left[\left\langle \sum_{\alpha \in \mathcal{K}_{l+1}} \delta^{l+1}_{\alpha}(x), \sum_{\beta \in \mathcal{K}_{l}} \delta^{l+1}_{\beta}(x') \right\rangle - \sum_{\alpha \in \mathcal{K}_{l+1}} \langle \delta^{l+1}_{\alpha}(x), \delta^{l+1}_{\alpha}(x') \rangle \right] \ , \tag{120}$$

$$= |\mathcal{F}_{l+1}| \sum_{\alpha,\beta \in \mathcal{K}_{l+1} \times \mathcal{K}_{l+1} : \alpha \neq \beta} \left\langle \delta^{l+1}_{\alpha}(x), \delta^{l+1}_{\beta}(x') \right\rangle \ . \tag{121}$$

Let us further assume that

$$\frac{1}{|\mathcal{K}_l|(|\mathcal{K}_l|-1)} \sum_{\alpha,\beta \in \mathcal{K}_l \times \mathcal{K}_l : \alpha \neq \beta} \mathop{\mathbb{E}}_{x,x',\theta} \left\langle \delta^{l+1}_{\alpha}(x), \delta^{l+1}_{\beta}(x') \right\rangle \tag{122}$$

$$= \frac{1}{|\mathcal{K}_{l+1}|(|\mathcal{K}_{l+1}|-1)} \sum_{\alpha,\beta \in \mathcal{K}_{l+1} \times \mathcal{K}_{l+1} : \alpha \neq \beta} \mathop{\mathbb{E}}_{x,x',\theta} \left\langle \delta^{l+1}_{\alpha}(x), \delta^{l+1}_{\beta}(x') \right\rangle \ . \tag{123}$$

Thus, taking expectation values over (x, x') and averaging over the allowable indices such that $\alpha \neq \beta$ we obtain,

$$\mathop{\mathbb{E}}_{\alpha \neq \beta} \left[\mathop{\mathbb{E}}_{x,x',\theta} \langle \delta^l_\alpha(x), \delta^l_\beta(x') \rangle \right] = \frac{C_l |\mathcal{F}_{l+1}|}{N_l} \frac{\sigma_{\rm w}^2 \gamma_l^2}{4\Gamma_l} \mathop{\mathbb{E}}_{\alpha \neq \beta} \left[\mathop{\mathbb{E}}_{x,x',\theta} \langle \delta^{l+1}_\alpha(x), \delta^{l+1}_\beta(x') \rangle \right] \ . \tag{124}$$

It follows that

$$\widehat{\Delta}_l := \mathop{\mathbb{E}}_{\alpha \neq \beta} \left[\mathop{\mathbb{E}}_{x,x',\theta} \left\langle \delta^l_\alpha(x), \delta^l_\beta(x') \right\rangle \right] \ , \tag{125}$$

$$= \frac{\sigma_{\rm w}^2 \gamma_l^2 \widehat{\Delta}_{l+1}}{4\Gamma_l} \ . \tag{126}$$

6.2.2 VANILLA CNN

Claim 6.5. *The forward recursions are* $\Gamma_{l+1} = \sigma_{\mathrm{b}}^2 + \sigma_{\mathrm{w}}^2 H_l$, $\widetilde{\Gamma}_{l+1} = \sigma_{\mathrm{b}}^2 + \sigma_{\mathrm{w}}^2 \widetilde{H}_l$ *and* $\widehat{\Gamma}_{l+1} = \sigma_{\mathrm{b}}^2 + \sigma_{\mathrm{w}}^2 \widehat{H}_l$ *where*

$$H_l = \frac{1}{2}\Gamma_l \ , \tag{127}$$

$$\widetilde{H}_l = \frac{\Gamma_l}{2\pi}\left[\sqrt{1-\tilde{c}^2} + \frac{\tilde{c}\pi}{2} + \tilde{c}\sin^{-1}(\tilde{c})\right] \ , \tag{128}$$

$$\widehat{H}_l = \frac{\Gamma_l}{2\pi}\left[\sqrt{1-\hat{c}^2} + \frac{\hat{c}\pi}{2} + \hat{c}\sin^{-1}(\hat{c})\right] \ , \tag{129}$$

and $\tilde{c} := \widetilde{\Gamma}_l/\Gamma_l$, $\hat{c} := \widehat{\Gamma}_l/\Gamma_l$.

Derivation. Substituting into equation 84, equation 88 and equation 93 we obtain,

$$H_l := \int Dz\, \sigma_l^2(\sqrt{\Gamma_l} z) \ , \tag{130}$$

$$= \frac{\Gamma_l}{2} \ , \tag{131}$$

$$\widetilde{H}_l = \int Dz_1\, Dz_2\, \sigma_l\big(\sqrt{\Gamma_l}\, z_1 \gamma_l\big)\sigma_l\left[\sqrt{\Gamma_l}\left(\tilde{c}_l\, z_1 + \sqrt{1-(\tilde{c}_l)^2}\, z_2\right)\gamma_l\right] \ , \tag{132}$$

$$= \frac{\Gamma_l}{2\pi}\left[\sqrt{1-\tilde{c}^2} + \frac{\tilde{c}\pi}{2} + \tilde{c}\sin^{-1}(\tilde{c})\right] \ , \tag{133}$$

$$\widehat{H}_l = \int Dz_1\, Dz_2\, \sigma_l\big(\sqrt{\Gamma_l}\, z_1 \gamma_l\big)\sigma_l\left[\sqrt{\Gamma_l}\left(\hat{c}_l\, z_1 + \sqrt{1-(\hat{c}_l)^2}\, z_2\right)\gamma_l\right] \ , \tag{134}$$

$$= \frac{\Gamma_l}{2\pi}\left[\sqrt{1-\hat{c}^2} + \frac{\hat{c}\pi}{2} + \hat{c}\sin^{-1}(\hat{c})\right] \ . \tag{135}$$

□

Claim 6.6. *The backward recursions are* $\Delta_l = \frac{\sigma_{\mathrm{w}}^2 \Delta_{l+1}}{2}$, $\widetilde{\Delta}_l = \frac{\sigma_{\mathrm{w}}^2 \widetilde{\Delta}_{l+1}}{2\pi}\left[\frac{\pi}{2} + \sin^{-1}(\tilde{c})\right]$ *and* $\widehat{\Delta}_l = \frac{\sigma_{\mathrm{w}}^2 \widehat{\Delta}_{l+1}}{2\pi}\left[\frac{\pi}{2} + \sin^{-1}(\hat{c})\right]$.

Derivation. In the backward direction, we have

$$\delta_\alpha^l[i] = \sum_{(\beta,k)\in\mathcal{K}_{l+1}\times[C_{l+1}]} \frac{\partial z_\beta^{l+1}[k]}{\partial z_\alpha^l[i]}\delta_\beta^{l+1}[k] \ , \tag{136}$$

$$= \sigma_l'\left(z_\alpha^l[i]\right) \sum_{(\beta,k)\in\mathcal{F}_{l+1}\times[C_{l+1}]} [W_\beta^{l+1}]_{ki}\delta_{\alpha-\beta}^{l+1}[k] \tag{137}$$

Under the usual independence assumptions,

$$\mathop{\mathbb{E}}_{\theta}\left\langle \delta_\alpha^l(x), \delta_\beta^l(x')\right\rangle = \frac{\sigma_{\mathrm{w}}^2}{N_l}\mathop{\mathbb{E}}_{\theta}\left\langle \sigma_l'\left(z_\alpha^l(x)\right), \sigma_l'\left(z_\beta^l(x')\right)\right\rangle \sum_{\delta\in\mathcal{F}_{l+1}} \left\langle \delta_{\alpha-\delta}^{l+1}(x), \delta_{\beta-\delta}^{l+1}(x')\right\rangle \ . \tag{138}$$

Thus,

$$\Delta_l := \mathop{\mathbb{E}}_{\alpha} \mathop{\mathbb{E}}_{x,\theta} \|\delta^l_\alpha(x)\|^2 \ , \tag{139}$$

$$= \frac{\sigma_{\mathrm{w}}^2 \Delta_{l+1}}{2} \tag{140}$$

$$\widetilde{\Delta}_l := \mathop{\mathbb{E}}_{\alpha} \mathop{\mathbb{E}}_{x,x',\theta} \langle \delta^l_\alpha(x), \delta^l_\alpha(x') \rangle \ , \tag{141}$$

$$= \frac{\sigma_{\mathrm{w}}^2 \widetilde{\Delta}_{l+1}}{2\pi} \left(\frac{\pi}{2} + \sin^{-1} \tilde{c} \right) \tag{142}$$

$$\widehat{\Delta}_l := \mathop{\mathbb{E}}_{\alpha \neq \beta} \left[\mathop{\mathbb{E}}_{x,x',\theta} \left\langle \delta^l_\alpha(x), \delta^l_\beta(x') \right\rangle \right] \ , \tag{143}$$

$$= \frac{\sigma_{\mathrm{w}}^2 \widehat{\Delta}_{l+1}}{2\pi} \left(\frac{\pi}{2} + \sin^{-1} \hat{c} \right) \ . \tag{144}$$

□

6.3 Derivation of Claim 3.1

Recall that the Fisher information matrix $I_\theta \in \mathbb{R}^{n \times n}$ is given by

$$I_\theta = \mathop{\mathbb{E}}_{x \sim \mathcal{D}} \left[\nabla_\theta f_\theta(x) \otimes \nabla_\theta f_\theta(x) \right] \ . \tag{145}$$

The claim is that the maximum eigenvalue of the Fisher Information Matrix is bounded as follows

$$\mathbb{E}_{(x,x') \sim \mathcal{D}} \langle \nabla_\theta f_\theta(x), \nabla_\theta f_\theta(x') \rangle \leq \lambda_{\max}(I_\theta) \leq \mathbb{E}_{x \sim \mathcal{D}} \langle \nabla_\theta f_\theta(x), \nabla_\theta f_\theta(x) \rangle \ . \tag{146}$$

The upper bound follows from convexity of $\lambda_{\max}(\cdot)$. Karakida et al. (2018) obtain the lower bound by considering the empirical estimate of the Fisher Information Matrix,

$$\widehat{I}_\theta = \frac{1}{m} \sum_{i=1}^m \nabla_\theta f_\theta(x_i) \otimes \nabla_\theta f_\theta(x_i) \ , \tag{147}$$

$$= \frac{1}{m} B_m^\top B_m \ . \tag{148}$$

where we have defined the matrix $B_m \in \mathbb{R}^{m \times n}$ with components $[B_m]_{ij} := \frac{\partial f_\theta(x_i)}{\partial \theta_j}$. We can then define a symmetric matrix $\widehat{H}_\theta \in \mathbb{R}^{m \times m}$ with the same eigenvalues as $\widehat{I}_\theta$,

$$\widehat{H}_\theta = \frac{1}{m} B_m B_m^\top \ , \tag{149}$$

The maximal eigenvalue of $\widehat{I}_\theta$ is thus computed by the Rayleigh quotient,

$$\lambda_{\max}(\widehat{I}_\theta) = \lambda_{\max}(\widehat{H}_\theta) = \max_{v : \|v\| = 1} \langle v, \widehat{H}_\theta \, v \rangle \ . \tag{150}$$

Letting $\mathbf{1} \in \mathbb{R}^m$ denote the vector of ones,

$$\lambda_{\max}(\widehat{I}_\theta) \geq \left\langle \frac{1}{\sqrt{m}} \mathbf{1}, \widehat{H}_\theta \, \frac{1}{\sqrt{m}} \mathbf{1} \right\rangle \ , \tag{151}$$

$$= \frac{1}{m^2} \sum_{i,j \in [m]} \left\langle \nabla_\theta f_\theta(x_i), \nabla_\theta f_\theta(x_j) \right\rangle, \tag{152}$$

which is the the plug-in estimator (V-statistic). Note that this bound can also be obtained using the population form of the Fisher information matrix using Jensen's inequality,

$$\lambda_{\max}(I_\theta) = \max_{v : \|v\| = 1} \langle v, I_\theta v \rangle \tag{153}$$

$$= \max_{v : \|v\| = 1} \mathop{\mathbb{E}}_{x \sim \mathcal{D}} \langle v, \nabla_\theta f_\theta(x) \rangle^2 \tag{154}$$

$$\geq \max_{v : \|v\| = 1} \left\langle v, \mathop{\mathbb{E}}_{x \sim \mathcal{D}} \nabla_\theta f_\theta(x) \right\rangle^2 \tag{155}$$

$$= \left\| \mathop{\mathbb{E}}_{x \sim \mathcal{D}} \nabla_\theta f_\theta(x) \right\|^2 \tag{156}$$

Thus, for a layered neural network we have

$$\bar{\lambda}_{\max}(\widehat{I}_\theta) \geq \frac{1}{m^2} \sum_{(x,x')} \sum_{l\in[L]} \mathop{\mathbb{E}}_{\theta} \langle \nabla_{\theta_l} f_\theta(x), \nabla_{\theta_l} f_\theta(x') \rangle \ , \tag{157}$$

$$=: \sum_{l\in[L]} f_l. \tag{158}$$

Specializing to a fully-connected neural network we obtain,

$$f_l = \frac{1}{m^2} \sum_{(x,x')} \mathop{\mathbb{E}}_{\theta} \left[\left\langle \frac{\partial f_\theta(x)}{\partial b_l}, \frac{\partial f_\theta(x')}{\partial b_l} \right\rangle + \left\langle \frac{\partial f_\theta(x)}{\partial W_l}, \frac{\partial f_\theta(x')}{\partial W_l} \right\rangle \right. \tag{159}$$

$$\left. + \left\langle \frac{\partial f_\theta(x)}{\partial \gamma_l}, \frac{\partial f_\theta(x')}{\partial \gamma_l} \right\rangle + \left\langle \frac{\partial f_\theta(x)}{\partial \beta_l}, \frac{\partial f_\theta(x')}{\partial \beta_l} \right\rangle \right] \ , \tag{160}$$

$$\simeq \frac{1}{m^2} \sum_{(x,x')} \mathop{\mathbb{E}}_{\theta} \left\langle \frac{\partial f_\theta(x)}{\partial W_l}, \frac{\partial f_\theta(x')}{\partial W_l} \right\rangle \ , \tag{161}$$

$$= \frac{1}{m^2} \sum_{(x,x')} \mathop{\mathbb{E}}_{\theta} \langle h^{l-1}(x), h^{l-1}(x') \rangle \langle \delta^l(x), \delta^l(x') \rangle \ , \tag{162}$$

$$\simeq \left[\frac{1}{m^2} \sum_{(x,x')} \mathop{\mathbb{E}}_{\theta} \langle h^{l-1}(x), h^{l-1}(x') \rangle \right] \left[\frac{1}{m^2} \sum_{(x,x')} \mathop{\mathbb{E}}_{\theta} \langle \delta^l(x), \delta^l(x') \rangle \right] \ , \tag{163}$$

$$= N_{l-1} \widetilde{H}_{l-1} \widetilde{\Delta}_l \ . \tag{164}$$

In the first approximation we use the fact that the terms with respect to b, γ, β are N_l times smaller than the term with respect to W. The second approximation uses the assumption that forward and backward order parameters are independent. The last approximation is for $m \gg 1$.

For convolutional layers we have

$$f_l = \frac{1}{m^2} \sum_{(x,x')} \mathop{\mathbb{E}}_{\theta} \left[\left\langle \frac{\partial f_\theta(x)}{\partial b_l}, \frac{\partial f_\theta(x')}{\partial b_l} \right\rangle + \left\langle \frac{\partial f_\theta(x)}{\partial W_l}, \frac{\partial f_\theta(x')}{\partial W_l} \right\rangle \right. \tag{165}$$

$$\left. + \left\langle \frac{\partial f_\theta(x)}{\partial \gamma_l}, \frac{\partial f_\theta(x')}{\partial \gamma_l} \right\rangle + \left\langle \frac{\partial f_\theta(x)}{\partial \beta_l}, \frac{\partial f_\theta(x')}{\partial \beta_l} \right\rangle \right] \ , \tag{166}$$

$$\simeq \frac{1}{m^2} \sum_{(x,x')} \mathop{\mathbb{E}}_{\theta} \left\langle \frac{\partial f_\theta(x)}{\partial W_l}, \frac{\partial f_\theta(x')}{\partial W_l} \right\rangle \ , \tag{167}$$

$$= \frac{1}{m^2} \sum_{(x,x')} \sum_{\alpha\in\mathcal{F}_l} \sum_{\beta,\beta'\in\mathcal{K}_l} \mathop{\mathbb{E}}_{\theta} \langle \delta^l_\beta(x), \delta^l_{\beta'}(x') \rangle \langle h^{l-1}_{\alpha+\beta}(x), h^{l-1}_{\alpha+\beta'}(x') \rangle \ , \tag{168}$$

$$\simeq \frac{1}{|\mathcal{K}_l|^2} \left[\frac{1}{m^2} \sum_{(x,x')} \sum_{\beta,\beta'\in\mathcal{K}_l} \mathop{\mathbb{E}}_{\theta} \langle \delta^l_\beta(x), \delta^l_{\beta'}(x') \rangle \right] \left[\frac{1}{m^2} \sum_{(x,x')} \sum_{\alpha\in\mathcal{F}_l} \sum_{\beta,\beta'\in\mathcal{K}_l} \mathop{\mathbb{E}}_{\theta} \langle h^{l-1}_{\alpha+\beta}(x), h^{l-1}_{\alpha+\beta'}(x') \rangle \right] \ , \tag{169}$$

$$= \left[|\mathcal{K}_l|(|\mathcal{K}_l|-1)\widehat{\Delta}_l + |\mathcal{K}_l|\widetilde{\Delta}_l \right] \left[\frac{\mathcal{C}_{l-1}|\mathcal{F}_l||\mathcal{K}_l|(|\mathcal{K}_l|-1)}{|\mathcal{K}_l|^2} \widehat{H}_{l-1} + \frac{\mathcal{C}_{l-1}|\mathcal{K}_l||\mathcal{F}_l|}{|\mathcal{K}_l|^2} \widetilde{H}_{l-1} \right] \ , \tag{170}$$

$$= N_{l-1} \left[(|\mathcal{K}_l|-1)\widehat{\Delta}_l + \widetilde{\Delta}_l \right] \left[(|\mathcal{K}_l|-1)\widehat{H}_{l-1} + \widetilde{H}_{l-1} \right] \ , \tag{171}$$

where $N_{l-1} = \mathcal{C}_{l-1}|\mathcal{F}_l|$. In the first approximation we use the fact that the terms with respect to b, γ, β are c_l times smaller than the term with respect to W. The second approximation uses the assumption that forward and backward order parameters are independent.

6.4 BASELINE

Baseline was an experiment of vanilla fully-connected networks trained on MNIST, with various σ_w weight initializations. Result is shown in Fig. 3.

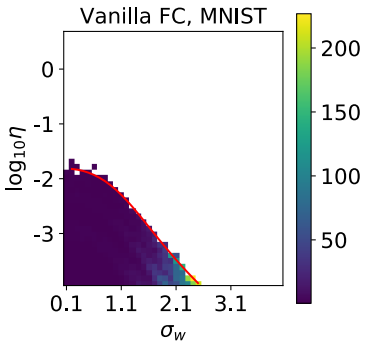

Figure 3: Heatmap showing test loss as a function of $(\log_{10}\eta, \gamma)$ after 5 epochs of training for vanilla fully-connected feed forward network where the relation between weight initialization variance and maximal learning rate is studied.

6.5 Additional Experiments

Our theory predicts that smaller γ has the effect of greatly reducing the λ_{max} of the FIM, and empirically networks with BatchNorm converge faster. Following this intuition, we performed additional experiments with VGG16 Simonyan & Zisserman (2014) and Preact-Resnet18 He et al. (2016a), with various γ initializations, fixed learning rate 0.1, momentum 0.9 and weight decay 0.0005, trained on CIFAR-10. The result is average over 5 different independent trainings. We find that smaller γ initialization indeed increases the speed of convergence, as shown below in Fig. 4.

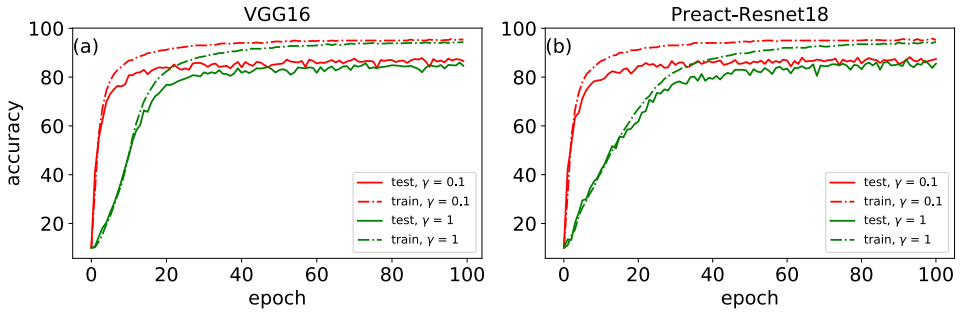

Figure 4: Learning curves of (a) VGG16 and (b) Preact-Resnet18 training on CIFAR-10, with the same hyperparameters except γ initialization. Results support that small γ initialization helps faster convergence.