REGULARIZING ACTIVATIONS IN NEURAL NETWORKS VIA DISTRIBUTION MATCHING WITH THE WASSERTEIN METRIC

Anonymous authors
Paper under double-blind review

ABSTRACT

Regularization and normalization have become an indispensable component in deep learning because it enables faster training and improved generalization performance. We propose the projected error function regularization loss (PER) that encourages activations to follow the standard normal distribution. PER randomly projects activations to one dimensional space and computes the regularization in the projected space. PER acts like the Pseudo-Huber loss in the projected space, enabling robust regularization for training deep neural networks. In addition, PER can capture interaction between hidden units by projection vector drawn from unit sphere. By doing so, PER minimizes the upper bound of the Wasserstein distance of order one between an empirical distribution of activations and the standard normal distribution. To the best of the authors' knowledge, this is the first work to regularize activations concerning the target distribution in the probability distribution space. We evaluate the proposed method on image classification task and word-level language modeling task.

1 INTRODUCTION

Training of deep neural networks is very challenging because of vanishing and exploding gradient problem (Hochreiter, 1998; Glorot & Bengio, 2010), existence of many flat regions and saddle points (Shalev-Shwartz et al., 2017), and the shattered gradient problem (Balduzzi et al., 2017). To remedy these issues, various methods for controlling hidden activations have been proposed such as normalization (Ioffe & Szegedy, 2015; Huang et al., 2018), regularization (Littwin & Wolf, 2018), initialization (Mishkin & Matas, 2015; Zhang et al., 2019), and architecture design (He et al., 2016).

Among various techniques of controlling activations, one well-known and successful path is controlling their first and second moments. Back in 1990s, it has been known that the neural network training can be benefited from normalizing input statistics so that samples have zero mean and identity covariance matrix (LeCun et al., 1998; Schraudolph, 1998). This idea motivated batch normalization (BN) that considers hidden activations as the input to the next layer and normalizes scale and shift of the activations (Ioffe & Szegedy, 2015).

Recent works show the effectiveness of different sample statistics of activations for normalization and regularization. Deecke et al. (2018) and Kalayeh & Shah (2019) normalize activations to several modes with different scales and translations. Variance constancy loss (VCL) implicitly normalizes the fourth moment by minimizing the variance of sample variances, which enables adaptive mode separation or collapse based on their prior probabilities (Littwin & Wolf, 2018). In addition, BN is extended to whiten activations (Huang et al., 2018; 2019), and to normalize general order of central moment in the sense of L^p norm including L^0 and L^∞ (Liao et al., 2016; Hoffer et al., 2018).

In this paper, we propose a new regularization method, called projected error function regularization (PER), that regularizes activations in probability distribution space with the Wasserstein metric. Specifically, PER encourages the distribution of activations to be close to the standard normal distribution. PER shares a similar strategy that dictates the desired distribution of activations with previous normalization/regularization methods such as BN and VCL. However, previous approaches are capable of concerning single, or few, sample statistics of activations. On the contrary, PER presents new perspective of concerning the *target distribution* $\mathcal{N}(0, I)$ for controlling the activations. By concerning the distribution itself, PER can implicitly consider various statistical characteristics simultaneously, e.g. all order of moments and correlation between hidden units. The extensive experiments on multiple challenging tasks show the efficiency of PER.

2 Related works

Since BN has been proposed, many normalization methods (Salimans & Kingma, 2016; Lei Ba et al., 2016; Ulyanov et al., 2016; Wu & He, 2018; Kingma & Dhariwal, 2018) have been suggested by normalizing activations to have a sample mean β and sample standard deviation γ. Even though its theoretical aspects on regularization and optimization are still being actively investigated (Santurkar et al., 2018; Kohler et al., 2018; Bjorck et al., 2018; Yang et al., 2019), many modern deep learning architectures employ BN as an essential building block for better performance and stable training.

Based on the work of Ioffe & Szegedy (2015), Huang et al. (2018; 2019) proposed normalization technique whitening the activation of each layer. These additional constraints on statistical relationship between activations show an significant improvement in generalization performance of residual networks. Although correlations, or statistical dependency between activations, are not explicitly constrained, dropout prevents activations from being activated at the same time, called co-adaptation, by randomly dropping the activations (Srivastava et al., 2014), the weights (Wan et al., 2013), and the spatially connected activations (Ghiasi et al., 2018).

Considering the BN as forcing activations to have learned value of norm of each unit in L^2 space, there are extensions that use other norms. Streaming normalization (Liao et al., 2016) explores the normalization of a different order of central moment with L^p norm for general p. Similarly, Hoffer et al. (2018) explores L^1 and L^∞ normalization, which enable low precision computation. Littwin & Wolf (2018) proposes a regularization loss that reduces the variance of sample variances of activation which is closely related to the fourth moment.

Initialization schemes such as balancing variances of each layer (Glorot & Bengio, 2010; He et al., 2015), bounding scale of activation and gradient in residual networks (Mishkin & Matas, 2015; Balduzzi et al., 2017; Gehring et al., 2017; Zhang et al., 2019), and norm preserving (Saxe et al., 2013) can be thought as stablizing activations at the initial state. Although it cannot be guaranteed that the desired initial state of activations is maintained during the course of training unlike normalization and regularization approaches, experimental evidences show that an initialization scheme can stabilize the learning process as well.

Recently, the Wasserstein metric have gained much popularity in a wide range of applications in deep learning with some nice properties such as being a metric in probability distribution space without requiring common supports of two distributions. For instance, it is successfully applied to a multi-labeled classification loss function (Frogner et al., 2015), gradient flow of policy update in reinforcement learning (Zhang et al., 2018), training of generative models (Arjovsky et al., 2017; Gulrajani et al., 2017; Kolouri et al., 2019), and capturing long term semantic structure in sequence-to-sequence language model (Chen et al., 2019). However, to the best of the authors' knowledge, PER is the first work regularizing activations in the Wasserstein probability distribution space.

3 Projected error function regularization

We consider a neural network with L layers each of which have d_l hidden units in layer l. Let $\mathcal{T} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^n$ be n training samples which are assumed to be i.i.d. samples drawn from a probability distribution $P_{\mathbf{x},\mathbf{y}}$. In this paper, we consider the optimization by stochastic gradient descent where we are given mini-batch of b samples randomly drawn from $\mathcal{T}$ at each training iteration. For i-th element in the mini-batch, the neural network recursively computes:

$$\boldsymbol{h}_i^l = \phi\left(\boldsymbol{W}^l \boldsymbol{h}_i^{l-1} + \boldsymbol{b}^l\right) \tag{1}$$

where $\boldsymbol{h}_i^0 = \boldsymbol{x}_i \in \mathbb{R}^{d_0}$, $\boldsymbol{h}_i^l \in \mathbb{R}^{d_l}$ is an i-th element of activation in layer l, ϕ is an activation function. In the case of recurrent neural networks (RNNs), the recursive relationship takes the form of:

$$\boldsymbol{h}_{t_i}^l = \phi\left(\boldsymbol{W}_{rec}^l \boldsymbol{h}_{t-1_i}^l + \boldsymbol{W}_{in}^l \boldsymbol{h}_{t_i}^{l-1} + \boldsymbol{b}^l\right) \tag{2}$$

where $\boldsymbol{h}_{t_i}^l$ is an i-th element of activation in layer l at time t and $\boldsymbol{h}_{0_i}^l$ is an initial state. Without loss of generality, we focus on activations in layer l and the mini-batch of samples $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^b$. Throughout this paper, we let f^l be a function made by compositions of recurrent relation in equation 2 up to layer l, i.e., $\boldsymbol{h}_i^l = f^l(\boldsymbol{x}_i)$, and f_j^l be a j-th output of a function f^l.

We are interested in the problem of controlling a set of hidden activations $\left\{\boldsymbol{h}_i^l\right\}_{i=1}^b$ observed in mini-batch. Before introducing our method, we review BN and its variants as controlling activations by concerning the norm in $L^p(\mathbb{R}^{d_0})$ which is the space of measurable functions whose p-th power of absolute value is Lebesgue integralable with norm of $f \in L^p(\mathbb{R}^{d_0})$ is given by:

$$\| f \|_p = \left(\int_{\mathbb{R}^{d_0}} |f(\boldsymbol{x})|^p dP_{\mathbf{x}}(\boldsymbol{x}) \right)^{1/p} < \infty \tag{3}$$

where $P_{\mathbf{x}}$ is the unknown probability distribution generating training samples $\{\boldsymbol{x}_i\}_{i=1}^n$. Since we have no access to $P_{\mathbf{x}}$, it is approximated by the empirical measure of mini-batch samples $\nu_{\mathbf{x}} = \frac{1}{b}\sum_{i=1}^b \delta_{\boldsymbol{x}_i}$ where $\delta_{\boldsymbol{x}}$ is the Dirac unit mass on $\boldsymbol{x}$.

BN and its variants normalize L^p norm of centralized activations[1], then scale and shift the normalized activations by learnable parameters. That is, the normalization methods ψ^l at layer l can be represented by composition of a normalizing function ψ_p^l and a learnable linear function ψ_s^l:

$$\psi^l(h_{ij}^l) = \psi_s^l(\psi_p^l(h_{ij}^l)) = \gamma_j^l \psi_p^l(h_{ij}^l) + \beta_j^l \qquad \psi_p^l(h_{ij}^l) = \frac{h_{ij}^l - \bar{\mu}_j}{\left(\sum_k \frac{1}{b}|h_{kj}^l - \bar{\mu}_j|^p\right)^{1/p}} \tag{4}$$

where h_{ij}^l is j-th unit of $\boldsymbol{h}_i^l$, $\bar{\mu}_j = \frac{1}{b}\sum_k h_{kj}^l$ is the sample mean, and β_j^l and γ_j^l is a learnable shift and scale parameters. We can see that ψ_p^l gives the constant norm $\| \psi_p^l \circ f_j^l \|_p = 1$ for any unit j and any empirical measure, i.e. samples of mini-batch. Therefore, the L^p norm of the function to j-th unit is bounded as: $\| \psi^l \circ f_j^l \|_p \leq \| \gamma_j^l \psi_p^l \circ f_j^l \|_p + \| \beta_j^l \|_p = \gamma_j^l + \beta_j^l$.

Instead of constraining norm of f_j^l to have certain value, PER concerns the 1-Wasserstein distance between empirical distribution of activations and the standard normal distribution in the probability distribution space $\mathcal{P}(\mathbb{R}^{d_l})$. Specifically, PER adopts a *soft constraint* approach that minimizes the upper bound of the Wasserstein distnace which will be proved in section 3.1. Let $\nu_{\mathbf{h}^l} = \frac{1}{b}\sum_i \delta_{\boldsymbol{h}_i^l} \in \mathcal{P}(\mathbb{R}^{d_l})$ be an empirical measure of hidden activations computed for mini-batch at layer l. Then, the loss and the gradient of PER for $\nu_{\mathbf{h}^l}$ are defined as:

$$\mathcal{L}_{per}(\nu_{\mathbf{h}^l}) = \frac{1}{b}\sum_{i=1}^b \mathbb{E}_{\boldsymbol{\theta}\sim U(\mathbb{S}^{d_l-1})} \left[\langle \boldsymbol{\theta}, \boldsymbol{h}_i^l \rangle \text{erf}\left(\frac{\langle \boldsymbol{\theta}, \boldsymbol{h}_i^l \rangle}{\sqrt{2}} \right) + \sqrt{\frac{2}{\pi}} \exp\left(-\frac{\langle \boldsymbol{\theta}, \boldsymbol{h}_i^l \rangle^2}{2} \right) \right] \tag{5}$$

$$\nabla_{\boldsymbol{h}_i^l} \mathcal{L}_{per}(\nu_{\mathbf{h}^l}) = \frac{1}{b}\mathbb{E}_{\boldsymbol{\theta}\sim U(\mathbb{S}^{d_l-1})} \left[\text{erf}\left(\langle \boldsymbol{\theta}, \boldsymbol{h}_i^l / \sqrt{2} \rangle \right) \boldsymbol{\theta} \right] \tag{6}$$

where $\mathbb{S}^{d_l-1}$ is the unit sphere in $\mathbb{R}^{d_l}$ and $U(\mathbb{S}^{d_l-1})$ is the uniform distribution on $\mathbb{S}^{d_l-1}$. In this paper, expectation over $U(\mathbb{S}^{d_l-1})$ will be approximated by the Monte Carlo method with s number of samples. Therefore, PER results in simple modification of the backward pass as in Alg. 1. As shown in the Fig. 1, $\mathcal{L}_{per}$ acts like the Pseudo-Huber loss $g(x) = \sqrt{1+x^2} - 1$ in the projected space. The Pseudo-Huber loss is smooth approximation of the Huber loss (Huber, 1964), and it is widely used in the context of the robust statistics (Barron, 2019). This robustness can prevent explosion of activation regularization loss due to outliers having large values that are prevalent in forward pass of deep neural networks without a normalization technique.

In addition, PER captures interaction between hidden units unlike activation norm regularization loss that is widely used in RNNs (Merity et al., 2017). Consider L^p activation norm as $\frac{1}{b}\sum_i \|$

[1] In the case of case when apply normalization techniques before non-linear activation function, the result of presented analysis will be applied to $\boldsymbol{W}^l f^{l-1} + \boldsymbol{b}^{(l)}$. However, in both cases, these techniques normalizes the target vectors the in the same way. Therefore, we analyze the situation when BN is applied to activations to directly contrast effects of BN and the proposed method even though applying BN to pre-activation is more common in literature.

Algorithm 1 Backward pass under PER

Input The number of Monte Carlo evaluations s, an activation for i-th sample $\boldsymbol{h}_i$, the gradient of the loss $\nabla_{\boldsymbol{h}_i}\mathcal{L}$, a regularization coefficient λ

```
$\boldsymbol{g} \leftarrow \boldsymbol{0}$
**for** $k \leftarrow 1$ to $s$ **do**
    Sample $\boldsymbol{v} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
    $\boldsymbol{\theta} \leftarrow \boldsymbol{v} / \parallel \boldsymbol{v} \parallel_2$
    Project $h'_i \leftarrow \langle \boldsymbol{h}_i, \boldsymbol{\theta} \rangle$
    $g_k \leftarrow \text{erf}\left(h'_i/\sqrt{2}\right)$
    $\boldsymbol{g} \leftarrow \boldsymbol{g} + g_k \boldsymbol{\theta}/s$
**end for**
**return** $\nabla_{\boldsymbol{h}_i}\mathcal{L} + \lambda \boldsymbol{g}$
```

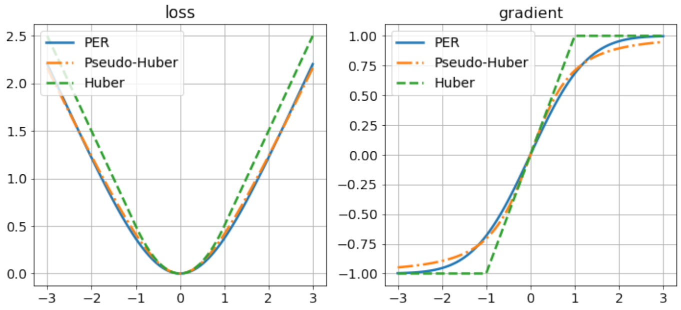


Figure 1: Illustration of PER loss and gradient in $\mathbb{R}$. Herein, PER loss is shifted by c so that $\mathcal{L}_{per}(\delta_0) - c = 0$. Huber loss is defined as $h(x) = |x| - 0.5$ in $|x| > 1$ and $h(x) = x^2/2$ in $|x| \leq 1$.

$\boldsymbol{h}_i^l \parallel_p^p = \frac{1}{b}\sum_{i,j}|h_{ij}^l|^p = \frac{1}{b}\sum_{i,j}|\langle \boldsymbol{h}_i^l, \boldsymbol{e}_j \rangle|^p$ where $\{\boldsymbol{e}_j\}_{j=1}^{d_l}$ is the natural basis of $\mathbb{R}^{d_l}$. That is, the activation norm regularization can be thought as computing the regularization loss of activations by projecting them using the natural basis. However, PER use more rich classes of projection vectors $\theta \sim U(\mathbb{S}^{d_l-1})$, encoding interaction between hidden units into the regularization loss.

3.1 Distribution matching with the Wasserstein metric

To understand the properties of PER, we examine the Wasserstein distance between activations and $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. The Wasserstein metric of order p between two probability measures μ and ν is defined by:

$$W_p(\mu, \nu) = \left(\inf_{\pi \in \prod(\mu,\nu)} \int_{\Omega \times \Omega} d^p(\boldsymbol{x}, \boldsymbol{y}) \pi(d\boldsymbol{x}, d\boldsymbol{y}) \right)^{1/p} \tag{7}$$

where $\prod(\mu, \nu)$ is the set of all joint probability measures on $\Omega \times \Omega$ having the first and the second marginals μ and ν, respectively.

Lemma 1. *Let $\mu \in \mathcal{P}(\mathbb{R})$ be the Gaussian measure defined as $\mu(\mathbb{A}) = \frac{1}{\sqrt{2\pi}} \int_{\mathbb{A}} \exp\left(-\frac{1}{2}x^2\right) dx$ and $\nu_{\mathrm{h}} \in \mathcal{P}(\mathbb{R})$ be an empirical measure of observations defined as $\nu_{\mathrm{h}} = \frac{1}{b}\sum_i \delta_{h_i}$. Then, $\mathcal{L}_{per}(\nu_{\mathrm{h}})$ is an upper bound of $W_1(\mu,\nu_{\mathrm{h}})$.*

Proof. In $\mathcal{P}(\mathbb{R})$, the 1-Wasserstein $W_1(\mu, \nu)$ can be formulated as (Rachev & Rüschendorf, 1998):

$$W_1(\mu, \nu_{\mathrm{h}}) = \int_0^1 |F_\mu^{-1}(z) - F_{\nu_{\mathrm{h}}}^{-1}(z)| dz = \int_{-\infty}^{\infty} |F_\mu(x) - F_{\nu_{\mathrm{h}}}(x)| dx \tag{8}$$

where F_μ and $F_{\nu_{\mathrm{h}}}$ are cumulative distribution functions (CDFs) of measures μ and ν_{h}, respectively. We have $|F_\mu - F_{\nu_{h_i}}| \in L^1(\mathbb{R})$ where $\nu_{h_i} = \delta_{h_i}$ for given h_i. Therefore, applying the Minkowski

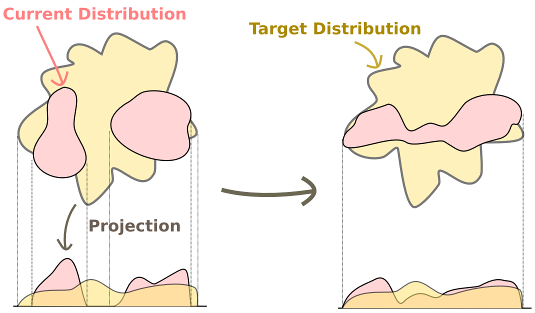


Figure 2: Illustration of minimization of the sliced Wasserstein distance between the current distribution and the target distribution. Note that it only concerns a distance in projected dimension.

inequality to equation 8 gives:

$$\int_{-\infty}^{\infty} |F_\mu(x) - \frac{1}{b}\sum_{i=1}^{b} 1_{h_i \leq x}| dx \leq \frac{1}{b}\sum_i \int_{-\infty}^{\infty} |F_\mu(x) - 1_{h_i \leq x}| dx$$
$$= \frac{1}{b}\sum_i \left(x_i \text{erf}\left(\frac{x_i}{\sqrt{2}}\right) + \sqrt{\frac{2}{\pi}} \exp\left(-\frac{x_i^2}{2}\right)\right) = \mathcal{L}_{per}(\nu_{\text{h}}) \quad (9)$$

which completes the proof. □

To extend the Lemma 1 from $\mathcal{P}(\mathbb{R})$ to $\mathcal{P}(\mathbb{R}^{d_l})$, we consider the sliced Wasserstein distance (Rabin et al., 2011) which approximates the Wasserstein distance in a high dimensional distribution space by projecting the distributions to $\mathbb{R}$ (Fig. 2). It is proved by that sliced Wasserstein and Wasserstein are equivalent metrics (Santambrogio, 2015; Bonnotte, 2013). The sliced Wasserstein of order one can be formulated as:

$$SW_1(\mu, \nu) = \int_{\mathbb{S}^{d-1}} W_1(\mu_{\boldsymbol{\theta}}, \nu_{\boldsymbol{\theta}}) d\lambda(\boldsymbol{\theta}) \quad (10)$$

where $\mu_{\boldsymbol{\theta}}$ and $\nu_{\boldsymbol{\theta}}$ represent the measures projected at the angle $\boldsymbol{\theta}$, and λ is an uniform measure on $\mathbb{S}^{d-1}$.

Corollary 2. *For the Gaussian measure $\mu \in \mathcal{P}(\mathbb{R}^{d_l})$ and the empirical measure of activations $\nu_{\mathbf{h}} = \frac{1}{b}\sum_i \delta_{\boldsymbol{h}_i}$, $\mathcal{L}_{per}(\nu_{\mathbf{h}})$ is an upper bound of $SW_1(\mu,\nu_{\mathbf{h}})$.*

Proof. We have $\mu_{\boldsymbol{\theta}}(A) = \frac{1}{\sqrt{2\pi}} \int_A \exp\left\{-\frac{1}{2}x^2\right\} dx$ for any choice of $\boldsymbol{\theta}$. Then, applying Lemma 1 to equation 10 yields the desired result:

$$SW_1(\mu, \nu) = \int_{\mathbb{S}^{d-1}} \int_{-\infty}^{\infty} |F_{\mu_{\boldsymbol{\theta}}}(x) - \frac{1}{b}\sum_i 1_{\langle \boldsymbol{h}_i^l, \boldsymbol{\theta}\rangle \leq x}| dx d\lambda(\boldsymbol{\theta})$$
$$\leq \frac{1}{b}\sum_i \int_{-\infty}^{\infty} \left(\langle \boldsymbol{h}_i^l, \boldsymbol{\theta}\rangle \text{erf}\left(\frac{\langle \boldsymbol{h}_i^l, \boldsymbol{\theta}\rangle}{\sqrt{2}}\right) + \sqrt{\frac{2}{\pi}} \exp\left(-\frac{\langle \boldsymbol{h}_i^l, \boldsymbol{\theta}\rangle^2}{2}\right)\right) d\lambda(\boldsymbol{\theta}) = \mathcal{L}_{per}(\nu_{\mathbf{h}}) \quad (11)$$

□

The use of $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ as the target can be motivated by the natural gradient (Amari, 1998) that enables parameter update to steepest descent direction in a Riemannian manifold. In addition to this, Roux et al. (2008) shows that the natural gradient direction corresponds to maximizing the probability of non-increasing generalization error. For gradient direction, natural gradient corrects the direction by multiplying the inverse Fisher information matrix $\boldsymbol{F}^{-1}$. In Raiko et al. (2012) and Desjardins et al. (2015), under the independence assumption between forward and backward passes and activations of different layers, the Fisher information matrix is formulated as a block diagonal matrix each of which block is defined by:

$$\boldsymbol{F}_l = \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})\sim(\mathbf{x},\mathbf{y})}\left[\frac{\partial \mathcal{L}}{\partial \text{vec}(\boldsymbol{W}^l)} \frac{\partial \mathcal{L}}{\partial \text{vec}(\boldsymbol{W}^l)}^T\right] = \mathbb{E}_{\boldsymbol{x}}\left[\boldsymbol{h}^{l-1}\boldsymbol{h}^{l-1^T}\right] \mathbb{E}_{(\boldsymbol{x},\boldsymbol{y})}\left[\frac{\partial \mathcal{L}}{\partial \boldsymbol{a}^l} \frac{\partial \mathcal{L}}{\partial \boldsymbol{a}^l}^T\right] \quad (12)$$

Table 1: The top-1 error rates of ResNet on CIFAR-10. Lower is better. All numbers are rounded to two decimal places. Boldface indicates minimum error.

Model	Method	Test error (%)
ResNet-56	Vanilla	7.21
	BN	6.95
	PER	**6.72**
ResNet-110	Vanilla	6.90
	BN	6.62
	PER	**6.19**

where vec$(\boldsymbol{W}^l)$ is vectorized $\boldsymbol{W}^l$, $\boldsymbol{h}^{l-1} = f^{l-1}(\boldsymbol{x})$, and $\boldsymbol{a}^l = \boldsymbol{W}^l f^{l-1}(\boldsymbol{x}) + \boldsymbol{b}^l$ for $\boldsymbol{x} \sim \mathrm{x}$.

From the equation 12, it have been empirically shown that faster training and improved generalization performance through making $\frac{1}{b}\sum_i \boldsymbol{h}_i^l \boldsymbol{h}_i^{l^T} \approx \boldsymbol{I}$ for making standard gradient to be close to natural gradient through zero mean and unit variance activations (LeCun et al., 1998; Schraudolph, 1998; Wiesler et al., 2014; Glorot & Bengio, 2010; Raiko et al., 2012) and decorrelated activations (Huang et al., 2018; Cogswell et al., 2015; Xiong et al., 2016). In this perspective, PER is expected to enjoy the same advantages by matching $\nu_{\mathbf{h}^l}$ to $\mathcal{N}(0, I)$, thereby promoting $\frac{1}{b}\sum_i \boldsymbol{h}_i^l \boldsymbol{h}_i^{l^T} \approx \boldsymbol{I}$.

While the sliced Wasserstein in equation 10 and its gradient can be exactly computed, we work with its upper bound because it removes the sorting operations for evaluating the inverse CDF of empirical distribution. Therefore, it requires no computational cost for sorting and enables distributed and large-batch training by removing dependency of gradient computation on batch dimension.

4 Experiments

This section illustrates the effectiveness of PER through experiments on different benchmark tasks with various datasets and architectures. We compare PER with BN normalizing the first and second moments and VCL regularizing fourth moments. In addition, we also compare PER with L^1 and L^2 activiation norm regularization which share similar behavior on certain areas in the projected space. Along with the benchmark experiments, we also analyze the impact of PER on the behavior of networks. Throughout all experiments, we use 256 number of slices for computation of PER and same regularization coefficient for activations in all layers.

4.1 Image classification in CIFAR-10 and CIFAR-100

We first evaluate PER in image classification task in CIFAR (Krizhevsky et al., 2009). We first evaluate PER with ResNet (He et al., 2016) in CIFAR-10. In this experiments, PER is compared with BN and vanilla networks initialized by fixup initialization (Zhang et al., 2019). We match the experimental details in training under BN with He et al. (2016) and under PER and vanilla with Zhang et al. (2019). Herein, we search the regularization coefficient over { 3e-4, 1e-4, 3e-5, 1e-5 }. Table 1 presents the results of CIFAR-10 with ResNet-56 and ResNet-110. PER outperforms BN as well as vanilla networks in both architectures. Especially, PER improves the test errors by 0.49 % and 0.71% in ResNet-56 and ResNet-110 without BN, respectively.

We also performed experiments with the deep ELU networks which examined in VCL Littwin & Wolf (2018). The deep ELU networks is a modified 11-layer CNN described in Clevert et al. (2015). Alongside of ELU, experiments with Leaky ReLU and ReLU are performend. We match the experimental settings in Littwin & Wolf (2018) except that we used 10x less learning rate for bias parameters and use of additional scalar bias after ReLU and Leaky ReLU based on Zhang et al. (2019). Again, we search the regularization coefficient over { 3e-4, 1e-4, 3e-5, 1e-5 }. In the case of ReLU and Leaky ReLU in CIFAR-100, we search { 3e-6, 1e-6, 3e-7, 1e-7 } because of divergence of training with PER in these setting. As shown in Table 2, PER shows best performance on four configurations among six configurations. In other cases, PER also results in comparable performance to BN or VCL giving at most 0.16 % less than best performing method. Herein, we want to note that

Table 2: The top-1 error rates of deep ELU network on CIFAR-10 and CIFAR-100. Lower is better. All numbers are rounded to two decimal places. Boldface indicates minimum error.

Activation	Method	CIFAR-10	CIFAR-100
ReLU	Vanilla	8.43	29.45
	BN	7.53	**29.13**
	VCL	7.80	30.30
	PER	**7.21**	29.29
LeakyReLU	Vanilla	6.73	26.50
	BN	6.38	26.83
	VCL	6.45	26.30
	PER	**6.29**	**25.50**
ELU	Vanilla	6.74	27.53
	BN	6.69	26.60
	VCL	**6.26**	25.86
	PER	6.42	**25.73**

the PER have no additional parameters unlike BN requiring parameters for each channel in every layer (2.5K total) and VCL requiring parameters for each location and channel in every layer (350K total).

4.2 Language modeling in PTB and WikiText2

We evaluate PER in word-level language modeling task in PTB (Mikolov et al., 2010) and WikiText2 (Merity et al., 2016). We apply PER loss to LSTM with two layers having 650 hidden units with and without reuse embedding (RE) in Inan et al. (2016) and Press & Wolf (2016), and variational dropout (VD) in Gal & Ghahramani (2016). We used the same configurations with Merity et al. (2017) except clipping gradient at 0.25 instead of 10 and train for 60 epochs instead of 80. PER is compared with recurrent BN (RBN) because BN is not directly applicable to LSTM (Cooijmans et al., 2016). PER is also compared with L^1 and L^2 activation norm regularizations. Herein, the search space of regularization coefficient is {3e-4, 1e-4, 3e-5 }. In the case of L^1 and L^2 penalties in PTB, we search additional hyperparameters { 1e-5, 3e-6, 1e-6, 3e-6, 1e-6, 3e-7, 1e-7 } because the searched coefficients seem to constrain the capacity.

We list in Table 3 the perplexity comparison of methods on PTB and WikiText2. While all regularization techniques shows somewhat regularization effects by improving test perplexity, PER gives best test perplexity except LSTM and RE-LSTM in PTB dataset. We also note that naively applying RBN often reduce performance especially when VD is used unlike PER. For instance, RBN increase test perplexity of VD-LSTM by about 5 in PTB and WikiText2.

4.3 Closeness to the standard normal distribution.

To examine the effect of PER on the closeness to $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$, we investigate distributional characteristics of activations under PER in deep ELU networks used in the previous benchmark task. We first analyze distribution of $\nu_{\boldsymbol{h}_j^l} = \frac{1}{n}\sum_i \delta_{h_{ij}^l}$ for some unit j and layer l (Fig. 3). In the analysis, BN shows somewhat stable distribution in a sense that distributional shift between two consecutive iterations due to the nature of normalization. On the contrary, activation distribution of vanilla method and PER result in somewhat noisy distributions. However, we observed that PER prevents explosion of variance and pushes the mean to zero. As shown in the Fig. 3, variances of $\nu_{\boldsymbol{h}_j^6}$ under PER and Vanilla are very high in the beginning of training. However, as training the network, the variance keeps decreasing towards one under PER. Similarly, biased means of $\nu_{\boldsymbol{h}_j^3}$ and $\nu_{\boldsymbol{h}_j^9}$ at early stage of learning are recovered under PER.

Since the distribution of single activation only capture the scalar, $SW_1(\mathcal{N}(\mathbf{0}, \boldsymbol{I}), \nu_{\mathbf{h}^l})$ is also examined (Fig. 4). Herein, the sliced Wassertein distance is computed by approximating the Gaussian

Table 3: Validation and test perplexity on PTB and WikiText2. Lower is better. All numbers are rounded to one decimal places. Boldface indicates minimum perplexity.

		PTB		WikiText2	
Model	Method	Valid	Test	Valid	Test
LSTM	Vanilla	123.2	122.0	138.9	132.7
	L^1 penalty	119.6	**114.1**	137.7	130.0
	L^2 penalty	120.5	115.2	136.0	131.1
	RBN	**118.2**	115.1	156.2	148.3
	PER	118.5	114.5	**134.2**	**129.6**
RE-LSTM	Vanilla	114.1	112.2	129.2	123.2
	L^1 penalty	112.2	108.5	128.6	122.7
	L^2 penalty	116.6	**108.2**	126.5	123.3
	RBN	113.6	110.4	138.1	131.6
	PER	**110.0**	108.5	**123.2**	**117.4**
VD-LSTM	Vanilla	84.9	81.1	99.6	94.5
	L^1 penalty	84.9	81.5	98.2	92.9
	L^2 penalty	84.5	81.2	98.8	94.2
	RBN	89.7	86.4	104.3	99.4
	PER	**84.1**	**80.7**	**98.1**	**92.6**
RE-VD-LSTM	Vanilla	78.9	75.7	91.4	86.4
	L^1 penalty	78.3	75.1	90.5	86.1
	L^2 penalty	79.2	75.8	**90.3**	86.1
	RBN	83.7	80.5	95.5	90.5
	PER	**78.1**	**74.9**	90.6	**85.9**

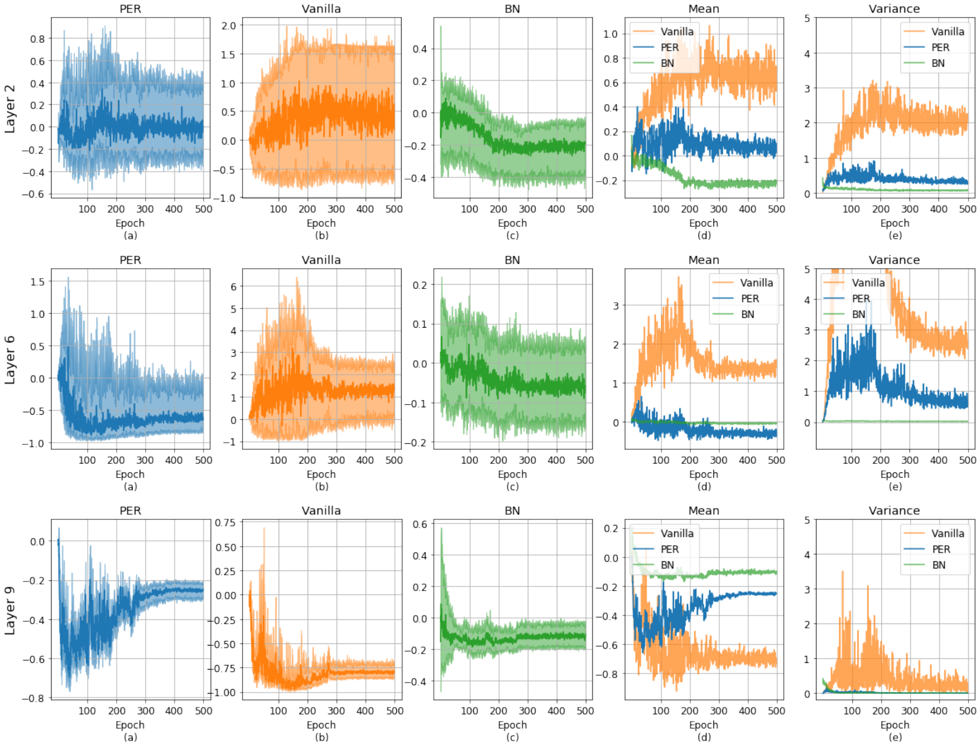
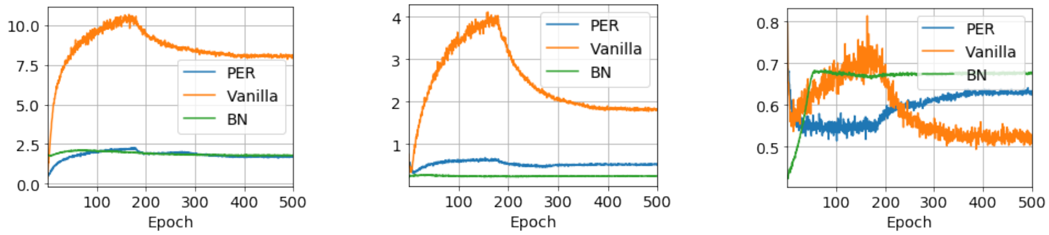


Figure 3: Evolution of distributions of $\nu_{\boldsymbol{h}_i^3}$, $\nu_{\boldsymbol{h}_j^6}$, and $\nu_{\boldsymbol{h}_j^9}$ for fixed randomly drawn i, j, k on training set. (a)-(c) represent values (0.25, 0.5, 0.75) quantiles under PER, Vanilla, and BN. (d) and (e) represent the mean and variance of activations. Variance is clipped at 5 for better visualization.



(a) $SW_1(\mathcal{N}(\mathbf{0}, \boldsymbol{I}), \nu_{\mathbf{h}^3})$

(b) $SW_1(\mathcal{N}(\mathbf{0}, \boldsymbol{I}), \nu_{\mathbf{h}^6})$

(c) $SW_1(\mathcal{N}(\mathbf{0}, \boldsymbol{I}), \nu_{\mathbf{h}^9})$

Figure 4: Closeness to the standard normal distribution in terms of the Wasserstein metric

measure using the empirical measure of samples drawn from $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ as in Rabin et al. (2011). As similar to the previous result, while normalization methods with initialization $\beta_j^l = 0$ and ${}_j^l = 1$ can constrain activations close to $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$ in the sliced Wasserstein metric sense, PER can also effectively control the distribution of activations without such normalization. This confirms that the regularization loss of PER prevent the distribution of activation from drifting away from the target distribution.

5 Conclusion

We proposed the regularization loss that minimizes the upper bound of the 1-Wasserstein distance between the standard normal distribution and the distribution of activations. PER differs from the existing methods that act on sample statistics rather than a distribution itself. Our experimental results in image classification and language modeling show that PER outperforms or shows a comparable performance to sample statistics based approaches (BN and VCL) as well as L^1 and L^2 activation norm regularization. The benchmark performances show the somewhat marginal but consistent regularization effects. The analysis on activations' distribution during training verifies that PER can stabilize probability distribution of activation without normalization techniques. Considering that the regularization loss can be easily applied to a wide range of tasks without changing architectures or training strategies unlike BN, we believe that the results indicate the valuable potential of regularizing networks in the probability distribution space as a future direction of research.

The idea of regularizing activations with metric in probability distribution space can be extended to many useful applications encoding task-specific priors. For instance, one can investigate the Laplace distribution to promote sparsity activation behavior. In addition, the empirical distribution of pretrained networks can be used as the target distribution. For instance, to prevent catastrophic forgetting, activation distribution can be regularized so that it does not drift away from the activation distribution from the previous tasks unlike constraining the weight updates in parameter l_2 space (Kirkpatrick et al., 2017) or in function L^2 space (Benjamin et al., 2018).

References

Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pp. 214–223, 2017.

David Balduzzi, Marcus Frean, Lennox Leary, JP Lewis, Kurt Wan-Duo Ma, and Brian McWilliams. The shattered gradients problem: If resnets are the answer, then what is the question? In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 342–350, 2017.

Jonathan T Barron. A general and adaptive robust loss function. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4331–4339, 2019.

Ari S Benjamin, David Rolnick, and Konrad Kording. Measuring and regularizing networks in function space. *arXiv preprint arXiv:1805.08289*, 2018.

Nils Bjorck, Carla P Gomes, Bart Selman, and Kilian Q Weinberger. Understanding batch normalization. In *Advances in Neural Information Processing Systems*, pp. 7694–7705, 2018.

Nicolas Bonnotte. *Unidimensional and evolution methods for optimal transportation*. PhD thesis, Paris 11, 2013.

Liqun Chen, Yizhe Zhang, Ruiyi Zhang, Chenyang Tao, Zhe Gan, Haichao Zhang, Bai Li, Dinghan Shen, Changyou Chen, and Lawrence Carin. Improving sequence-to-sequence learning via optimal transport. *arXiv preprint arXiv:1901.06283*, 2019.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

Michael Cogswell, Faruk Ahmed, Ross Girshick, Larry Zitnick, and Dhruv Batra. Reducing overfitting in deep networks by decorrelating representations. *arXiv preprint arXiv:1511.06068*, 2015.

Tim Cooijmans, Nicolas Ballas, César Laurent, Çağlar Gülçehre, and Aaron Courville. Recurrent batch normalization. *arXiv preprint arXiv:1603.09025*, 2016.

Lucas Deecke, Iain Murray, and Hakan Bilen. Mode normalization. *arXiv preprint arXiv:1810.05466*, 2018.

Guillaume Desjardins, Karen Simonyan, Razvan Pascanu, et al. Natural neural networks. In *Advances in Neural Information Processing Systems*, pp. 2071–2079, 2015.

Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya, and Tomaso A Poggio. Learning with a wasserstein loss. In *Advances in Neural Information Processing Systems*, pp. 2053–2061, 2015.

Yarin Gal and Zoubin Ghahramani. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems*, pp. 1019–1027, 2016.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1243–1252, 2017.

Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Dropblock: A regularization method for convolutional networks. In *Advances in Neural Information Processing Systems*, pp. 10727–10737, 2018.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 249–256, 2010.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pp. 5767–5777, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international Conference on Computer Vision*, pp. 1026–1034, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02): 107–116, 1998.

Elad Hoffer, Ron Banner, Itay Golan, and Daniel Soudry. Norm matters: efficient and accurate normalization schemes in deep networks. In *Advances in Neural Information Processing Systems*, pp. 2160–2170, 2018.

Lei Huang, Dawei Yang, Bo Lang, and Jia Deng. Decorrelated batch normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 791–800, 2018.

Lei Huang, Yi Zhou, Fan Zhu, Li Liu, and Ling Shao. Iterative normalization: Beyond standardization towards efficient whitening. *arXiv preprint arXiv:1904.03441*, 2019.

Peter J Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, pp. 73–101, 1964.

Hakan Inan, Khashayar Khosravi, and Richard Socher. Tying word vectors and word classifiers: A loss framework for language modeling. *arXiv preprint arXiv:1611.01462*, 2016.

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

Mahdi M Kalayeh and Mubarak Shah. Training faster by separating modes of variation in batch-normalized models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pp. 10215–10224, 2018.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

Jonas Kohler, Hadi Daneshmand, Aurelien Lucchi, Ming Zhou, Klaus Neymeyr, and Thomas Hofmann. Towards a theoretical understanding of batch normalization. *arXiv preprint arXiv:1805.10694*, 2018.

Soheil Kolouri, Phillip E. Pope, Charles E. Martin, and Gustavo K. Rohde. Sliced wasserstein auto-encoders. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=H1xaJn05FQ`.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Yann LeCun, Leon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural Networks: Tricks of the Trade*, pp. 9–50. Springer, 1998.

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

Qianli Liao, Kenji Kawaguchi, and Tomaso Poggio. Streaming normalization: Towards simpler and more biologically-plausible normalizations for online and recurrent learning. *arXiv preprint arXiv:1610.06160*, 2016.

Etai Littwin and Lior Wolf. Regularizing by the variance of the activations' sample-variances. In *Advances in Neural Information Processing Systems*, pp. 2115–2125, 2018.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Stephen Merity, Bryan McCann, and Richard Socher. Revisiting activation regularization for language rnns. *arXiv preprint arXiv:1708.01009*, 2017.

Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*, 2010.

Dmytro Mishkin and Jiri Matas. All you need is a good init. *arXiv preprint arXiv:1511.06422*, 2015.

Ofir Press and Lior Wolf. Using the output embedding to improve language models. *arXiv preprint arXiv:1608.05859*, 2016.

Julien Rabin, Gabriel Peyré, Julie Delon, and Marc Bernot. Wasserstein barycenter and its application to texture mixing. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pp. 435–446. Springer, 2011.

Svetlozar T Rachev and Ludger Rüschendorf. *Mass Transportation Problems: Volume I: Theory*, volume 1. Springer Science & Business Media, 1998.

Tapani Raiko, Harri Valpola, and Yann LeCun. Deep learning made easier by linear transformations in perceptrons. In *Artificial Intelligence and Statistics*, pp. 924–932, 2012.

Nicolas L Roux, Pierre-Antoine Manzagol, and Yoshua Bengio. Topmoumoute online natural gradient algorithm. In *Advances in neural information processing systems*, pp. 849–856, 2008.

Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *Advances in Neural Information Processing Systems*, pp. 901–909, 2016.

Filippo Santambrogio. Optimal transport for applied mathematicians. *Birkäuser, NY*, 55:58–63, 2015.

Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? In *Advances in Neural Information Processing Systems*, pp. 2483–2493, 2018.

Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

Nicol Schraudolph. Accelerated gradient descent by factor-centering decomposition. *Technical report/IDSIA*, 98, 1998.

Shai Shalev-Shwartz, Ohad Shamir, and Shaked Shammah. Failures of gradient-based deep learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3067–3075. JMLR. org, 2017.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

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

Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *International Conference on Machine Learning*, pp. 1058–1066, 2013.

Simon Wiesler, Alexander Richard, Ralf Schlüter, and Hermann Ney. Mean-normalized stochastic gradient for large-scale deep learning. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 180–184. IEEE, 2014.

Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European Conference on Computer Vision*, pp. 3–19, 2018.

Wei Xiong, Bo Du, Lefei Zhang, Ruimin Hu, and Dacheng Tao. Regularizing deep convolutional neural networks with a structured decorrelation constraint. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pp. 519–528. IEEE, 2016.

Greg Yang, Jeffrey Pennington, Vinay Rao, Jascha Sohl-Dickstein, and Samuel S Schoenholz. A mean field theory of batch normalization. *arXiv preprint arXiv:1902.08129*, 2019.

Hongyi Zhang, Yann N Dauphin, and Tengyu Ma. Fixup initialization: Residual learning without normalization. *arXiv preprint arXiv:1901.09321*, 2019.

Ruiyi Zhang, Changyou Chen, Chunyuan Li, and Lawrence Carin. Policy optimization as wasserstein gradient flows. *arXiv preprint arXiv:1808.03030*, 2018.