CAN WE GAIN MORE FROM ORTHOGONALITY REGULARIZATIONS IN TRAINING DEEP NETWORKS?

A Thesis

by

NITIN KUMAR BANSAL

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Zhangyang (Atlas) Wang
Committee Members,	Alan Dabney
	Yoonsuck Choe
Head of Department,	Dilma Da Silva

December 2018

Major Subject: Computer Engineering

ABSTRACT

[1]This work seeks to answer the question: *as the (near-) orthogonality of weights is found to be a favorable property for training deep convolutional neural networks, how we can enforce it in more effective and easy-to-use ways?* Through this work we look to come up with novel orthogonality regularizations for training deep CNNs, utilizing various advanced analytical tools such as mutual coherence and Restricted Isometric Property. These plug-and-play regularizations can be conveniently incorporated into training almost any CNN without extra hassle. We then benchmark their effects on three state-of-the-art models: ResNet, WideResNet, and ResNeXt, on CIFAR-10 and CIFAR-100 and SVHN datasets. To validate method's efficacy across various distribution and dataset, we apply the best performing regularizer(SRIP), for different setting of WideResNet to ImageNet Dataset. We observe consistent performance gains after applying those proposed regularizations, in terms of both the final accuracies achieved, and accelerated and more stable convergences.

[1] Reprinted with permission from Abstract section of *Can We Gain More from Orthogonality Regularizations in Training Deep Networks?* by N. Bansal, X. Chen and Z. Wang, 2018,Advances in Neural Information Processing Systems 31 (NIPS 2018) pre-proceedings

ACKNOWLEDGMENTS

I would like to thank the Neural Information Processing Systems for allowing me to re-use my work titled ***Can We Gain More From Orthogonality Regularizationss in Training Deep CNNs?***, which was accepted in NIPS 2018 as a poster, as my Thesis report. I would also like to thank my Committee members and Chair for guidance and suggestions. Special thanks go to Dr. Bobak Mortazavi for agreeing to be an evaluator in my final presentation at a very short notice.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis committee consisting of Dr. Atlas Wang and Dr. Yoonsuck Choe of the Department of Computer Science and Engineering at Texas A&M University and Dr. Alan Dabney of the Department of Statistics at Texas A&M University. Xiaohan Chen, my peer at VITA Lab, at Texas A&M University, contributed during different phase of my thesis work in deriving and discussing theoretical concepts. GPU Resources provided by HPRC TAMU were also pivotal in smooth running of the experiments.

Funding Sources

Graduate study was supported by funding from part-time student worker positions and Teaching assistant fellowship at Texas A&M University.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

1. INTRODUCTION

1.1 Current Work & Its Relevance

[†] Despite the tremendous success of deep convolutional neural networks (CNNs) [17], their training remains to be notoriously difficult both theoretically and practically, especially for state-of-the-art ultra-deep CNNs. Potential reasons accounting for such difficulty lie in multiple folds, ranging from vanishing/exploding gradients [18], to feature statistic shifts [19], to the proliferation of saddle points [20], and so on. To address these issues, various solutions have been proposed to alleviate those issues, examples of which include parameter initialization [21], residual connections [1], normalization of internal activations [19], and second-order optimization algorithms [20].

This paper focuses on one type of structural regularizations: *orthogonality*, to be imposed on linear transformations between hidden layers of CNNs. The orthogonality implies energy preservation, which is extensively explored for filter banks in signal processing and guarantees that energy of activations will not be amplified [22]. Therefore, it can stabilize the distribution of activations over layers within CNNs [23, 24] and make optimization more efficient. Another way to analyze this would be to see through the prism of matrix norm and thus constraining the singular values of the Parameter Matrix of the model.[21] advocates orthogonal initialization of weight matrices, and theoretically analyzes its effects on learning efficiency using deep linear networks. The paper [21] argues importance of random orthogonal initialization for weights, as it provides conditions similar to that provided by unsupervised pre-training, and also helps in faithful propagation of gradients.Practical results on image classification using orthogonal initialization are also presented in [25]. More recently, a few works [26, 27, 28, 29, 30] look at (various forms of) enforcing orthogonality regularizations or constraints throughout training, as part of their specialized models for applications such as classification [29] or person re-identification [31]. They observed encouraging result improvements. However, a dedicated and thorough examination on the effects

[†] Reprinted with permission from Introduction section of *Can We Gain More from Orthogonality Regularizations in Training Deep Networks?* by N. Bansal, X. Chen and Z. Wang, 2018,Advances in Neural Information Processing Systems 31 (NIPS 2018) pre-proceedings

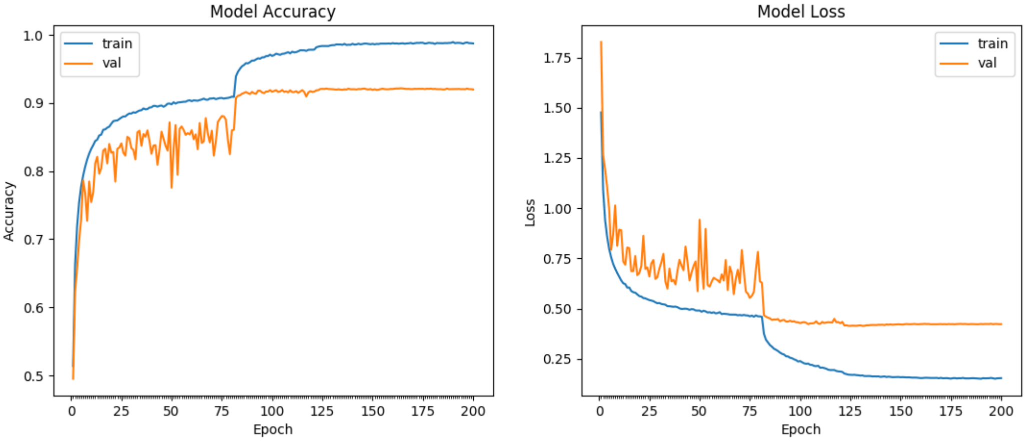


Figure 1.1: Training and Loss Curves : Original Regularizer

of orthogonality for training state-of-the-art general CNNs has been absent so far. Even more importantly, how to evaluate and enforce orthogonality for non-square weight matrices does not have a sole optimal answer. As we will explain later, existing works employ the most obvious but unnecessarily appropriate option, while we will introduce a series of more sophisticated regularizers that lead to larger performance gains.

1.2 Motivation and Our Focus

As we have already seen various difficulty, which we face while training a deep neural networks, and how orthogonality could be of help. We, as part of our work focus on achieving two goals:

- Improving Task Accuracy
- Improving Training Stability

Previous work in the field of orthogonality [29] or [31], have shown great promise in terms of achieving better task accuracy, but discuss little about the stability factor. We found that there is large fluctuations in training accuracy and loss, while training original Resnet Model, as shown in the Figure 1.1 between 20-80 epochs. Fluctuations seen could be attributed to unstable parameter space, or wild change in the network parameters, while the training process.

To draw the comparison and show the effectiveness of our method, we train the model based on one of our regularizer. Figure 1.2 shows the training curve achieved with our regularizer enforced,

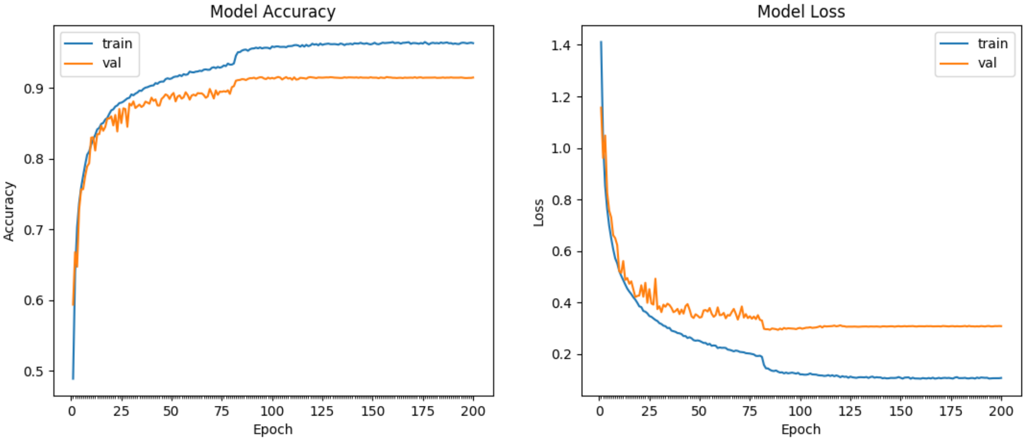


Figure 1.2: Training and Loss Curves : Orthogonal Regularizer

and we could see the stark difference between the curves achieved by 1.2 1.1 in term of stability, particularly in epochs ranging between 25-80.

This work aims to investigate and push forward various ways to enforce orthogonality regularizations on training deep CNNs. Specifically, we introduce three novel regularization forms for orthogonality, ranging from the double-sided variant of standard Frobenius norm-based regularizer, to utilizing Mutual Coherence (MC) and Restricted Isometry Property (RIP) tools [32, 33]. Those orthogonality regularizations have a plug-and-play nature, i.e., they can be incorporated with training almost any CNN without hassle. We extensively evaluate the proposed orthogonality regularizations on three state-of-the-art CNNs: ResNet-110 [1], ResNeXt [3], WideResNet [2]. In all experiments, we observe the consistent and remarkable accuracy boosts (e.g., **2.47%** in CIFAR-100 top-1 accuracy when using WideResNet), as well as accelerated and more stable convergences, *without any other change made to the original models*. It implies that many deep CNNs may have not been unleashed with their full powers yet, where orthogonality regularizations can help. Related to the concept of stable convergence, we use the term better *Parameter Space* instead, where a model in better *Parameter Space*, keeps/updates parameters during training in such a fashion, that there are very few fluctuations in validation accuracy, implying better parameter learning. Stability during the training process, is one of the most sought after property of a good model, predominantly in field of Generative Adversarial Networks. The orthogonal regularizers proposed in our work, could be further used in discriminator and generator module of the GANs to provide further stability and also have regularizing effect.

Our experiments further reveal that larger performance gains can be attained by designing stronger forms of orthogonality regularizations. We find the RIP-based regularizer, which has better analytical grounds to characterize near-orthogonal systems [34], to consistently outperform existing Frobenius norm-based regularizers and others.

2. RELATED WORK

2.1 Related Work Literature Survey

[†] To remedy unstable gradient and co-variate shift problems, [18, 35] advocated near constant variances of each layer's output for initialization. [19] presented a major breakthrough in stabilizing training, via ensuring each layer's output to be identical distributions which reduce the internal covariate shift. [36] further decoupled the norm of the weight vector from its phase(direction) while introducing independence between mini-batch examples, resulting in a better optimization problem. Orthogonal weights have been widely explored in Recurrent Neural Networks (RNNs) [37, 38, 39, 40, 41, 42] to help avoid gradient vanishing/explosion. [37] proposed a soft constraint technique to combat vanishing gradient, by forcing the Jacobian matrices to preserve energy measured by Frobenius norm. The more recent study [41] investigated the effect of soft versus hard orthogonal constraints on the performance of RNNs, the former by specifying an allowable range for the maximum singular value of the transition matrix and thus allowing for its small intervals around one.

In CNNs, orthogonal weights are also recognized to stabilize the layer-wise distribution of activations [23] and make optimization more efficient. [21, 25] presented the idea of orthogonal weight initialization in CNNs, which is driven by the norm-preserving property of orthogonal matrix: a similar outcome which BN tried to achieve. [21] analyzed the non-linear dynamics of CNN training. Under simplified assumptions, they concluded that random orthogonal initialization of weights will give rise to the same convergence rate as unsupervised pre-training, and will be superior than random Gaussian initialization. However, a good initial condition such as orthogonality does not necessarily sustain throughout training. In fact, the weight orthogonality and isometry will break down easily when training starts, if not properly regularized [21]. Several recent works [27, 28, 30] considered Stiefel manifold-based hard constraints of weights. [27] proposed a Stiefel

[†]Reprinted with permission from Related Work section of *Can We Gain More from Orthogonality Regularizations in Training Deep Networks?* by N. Bansal, X. Chen and Z. Wang, 2018,Advances in Neural Information Processing Systems 31 (NIPS 2018) pre-proceedings

layer to guarantee fully connected layers to be orthogonal by using Reimannian gradients, without considering similar handling for convolutional layers; their performance reported on VGG networks [43] were less than promising. [28] extended Riemannian optimization to convolutional layers and require filters within the same channel to be orthogonal. To overcome the challenge that CNN weights are usually rectangular rather than square matrices, [30] generalized Stiefel manifold property and formulated an Optimization over Multiple Dependent Stiefel Manifolds (OMDSM) problem. Different from [28], it ensured filters across channels to be orthogonal. A related work [26] adopted a Singular Value Bounding (SVB) method, via explicitly thresholding the singular values of weight matrices between a pre-specified narrow band around the value of one. The above methods [26, 27, 28, 30] all fall in the category of enforcing "hard orthogonality constraints" into optimization ([26] could be viewed as a relaxed constraint), and have to call the singular value decomposition (SVD) repeatedly during training. The cost of SVD on high-dimensional matrices is expensive even in GPUs, *which is one reason why we choose not to go for the "hard constraint" direction in this paper*. Moreover, since CNN weight matrices cannot exactly lie on a Stiefel manifold as they are either very "thin" or "fat" (e.g., $W^TW = I$ may never happen for an overcomplete "fat" W due to rank deficiency of its gram matrix), special treatments are needed to maintain such hard constraint. For example, [30] proposed group based orthogonalization to first divide an over-complete weight matrix into "thin" column-wise groups, and then applying Stiefel manifold constraints group-wise. The strategy was also motivated by reducing the computational burden of computing large-scale SVDs.

A recent work [29] explored orthogonal regularization, by enforcing the Gram matrix of each weight matrix to be close to identity under Frobenius norm. It constrains orthogonality among filters in one layer, leading to smaller correlations among learned features and implicitly reducing the filter redundancy. Such a soft orthonormal regularizer is differentiable and requires no SVD, thus being computationally cheaper than its "hard constraint" siblings. However, we will see later that Frobenius norm-based orthogonality regularization is only a rough approximation, and is inaccurate for "fat" matrices as well. The authors relied on a backward error modulation step, as well as similar group-wise orthogonalization as in [30]. We also notice that [29] displayed the strong advantage of enforcing orthogonality in training the authors' self-designed plain deep CNNs (i.e. without residual connections). However, they found fewer performance impacts when applying the same to training prevalent network architectures such as ResNet [1]. In comparison, our orthogonality regularizations can be added to CNNs as "plug-and-play" components, without bothering any assistance components. We observe evident improvements brought by them on most popular ResNet architectures.

Finally, we briefly outline a few works related to orthogonality in more general senses. One may notice that enforcing matrix to be (near-)orthogonal during training will lead to its spectral norm being always equal (or close) to one, which provides a potential link between regularizing orthogonality and spectrum. In [44], the authors showed that the spectrum of Extended Data Jacobian Matrix (EDJM) affected the network performance, and proposed a spectral soft regularizer that encourages major singular values of EDJM to be closer to the largest one. [45] claimed that the maximum eigenvalue of the Hessian predicted the generalizability of CNNs. Motivated by that, [8] penalized the spectral norm of weight matrices in CNNs. A similar idea was later extended in [46] for training generative adversarial networks, by proposing a spectral normalization technique to normalize the spectral norm/Lipschitz norm of the weight matrix to be one.

3. METHODOLOGY

3.1 Models and Architecture Used

The main task which we concentrate as to validate the effectiveness of our new regularizer is, Image Classification.Since the inception of AlexNet[17] in 2012, there have been plethora of Network which have come, and shown great improvement in task of Image Classification. VGG [47], has been one such model proposed, which gave better results when compared to AlexNet. Since, then work such as ResNet-110 [1], ResNeXt [3], WideResNet [2] and [11], have pioneered in utilizing the advantage of residual connections, to give state-of-art results for Classification task for all datasets.

3.1.1 ResNet[1]

[1] was the first of its kind of work, which talked about residual connections in the Deep CNN Models, which helped curb the problem of vanishing gradient in the Deep Network. The models proposed before ResNet[1], were failing to add to the accuracy, even after adding substantial number of layers. The accuracy instead saw a decrease as the model were made more deeper, mainly due to gradient instability and proliferation of saddle points. Even if there was some improvement seen in some cases, the computational cost was too expensive,as the we had to add millions of new parameter to see an improvement of $~$0.1-0.2%. As shown in various residual based models ResNet-110 [1], ResNeXt [3], WideResNet [2] and [11], residual connections seems to help, the gradient flow, as the residual connections provide an alternate path for the flow of gradients, and also acts as an regularizer by acting as an ensemble of CNNs, and thus reducing over-fitting due to increase in the number of parameters. As seen in the figure 3.1, Output of the one residual Block is calculated based not only on the Convolutional Layers the Input passes through but the input itself.At this point it will be apt to mention,there are two ways of defining the *Residual Block*,One where the activation is applied after the residual connection and one where the activation is applied before(pre-activation). The pre-activation version of the ResNet shows better training stability

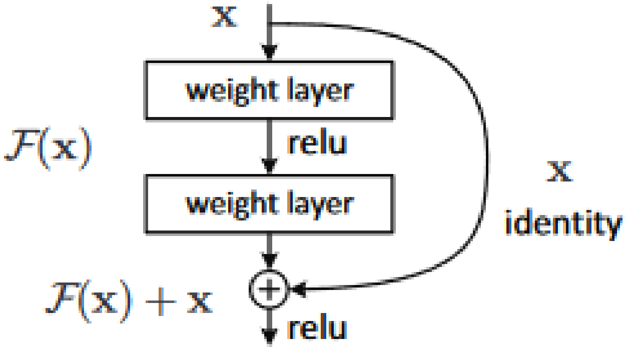
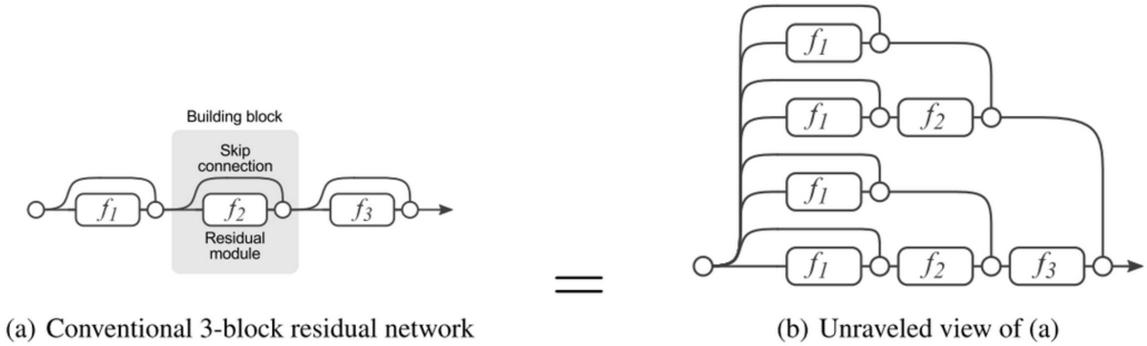

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 3.1: Residual Block : Reprinted From [1]

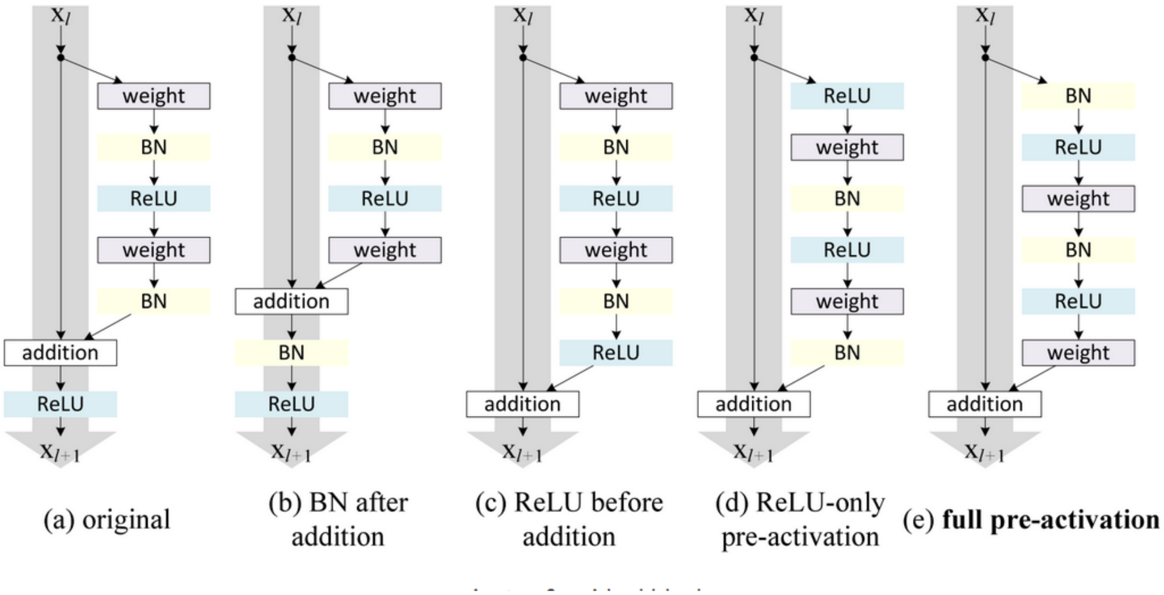


(a) Conventional 3-block residual network

(b) Unraveled view of (a)

Figure 3.2: Resnet Regularizer Ensemble Outlook: Reprinted From [10]

as argued in [11]. Figure 3.3 shows the different configuration of Residual Block. We had also mentioned earlier that ResNet model, acts as a regularizer too, We see in the Figure 3.2, how the ResNet could be visualized as a Ensemble of Networks,which can be used reduce variance of the resulting model and this acts an regularizer. The model itself consists of various blocks of Residual Blocks, stacked on top of each other, to make models of varying depth such as 34,50,101 or 110. Figure 3.4 shows a comparison between the VGG and a simple ResNet Model for ImageNet Dataset. There are slight modification in model, for different dataset, but most of it remains the same.



(a) original (b) BN after addition (c) ReLU before addition (d) ReLU-only pre-activation (e) **full pre-activation**

Figure 3.3: Pre-Activation ResNet: Reprinted From [11]

3.1.2 WideResNet [2]

WideResNet [2] offers a slightly alternate structure to the model proposed in [1]. [1] model also seems to be ineffective in terms of accuracy improvement, when the depth of the model increases. [2] argued about broadening the channel width, while keeping the kernel size same. Figure 3.5 shows the difference between the a simple ResNet block and a WideResNet block. Most of other architectural details remains the same, while further details are mentioned in Results and Summary section.

3.1.3 ResNext [3]

Model proposed in [3], is inspired both from [1] and the Google Inception Model [48], where the architecture is divided in to different paths, named as Cardinality of the Model.This model further emphasizes on increasing the width rather than the depth of the ResNet model, which results in further improvement in the network.Model uses all other features used in a basic ResNet model, except for it uses a different Residual Block, and a typical ResNext model is characterized by Depth-Width-Cardinality. Figure 3.6, shows the three version of visualizing the ResNext Building Blocks.

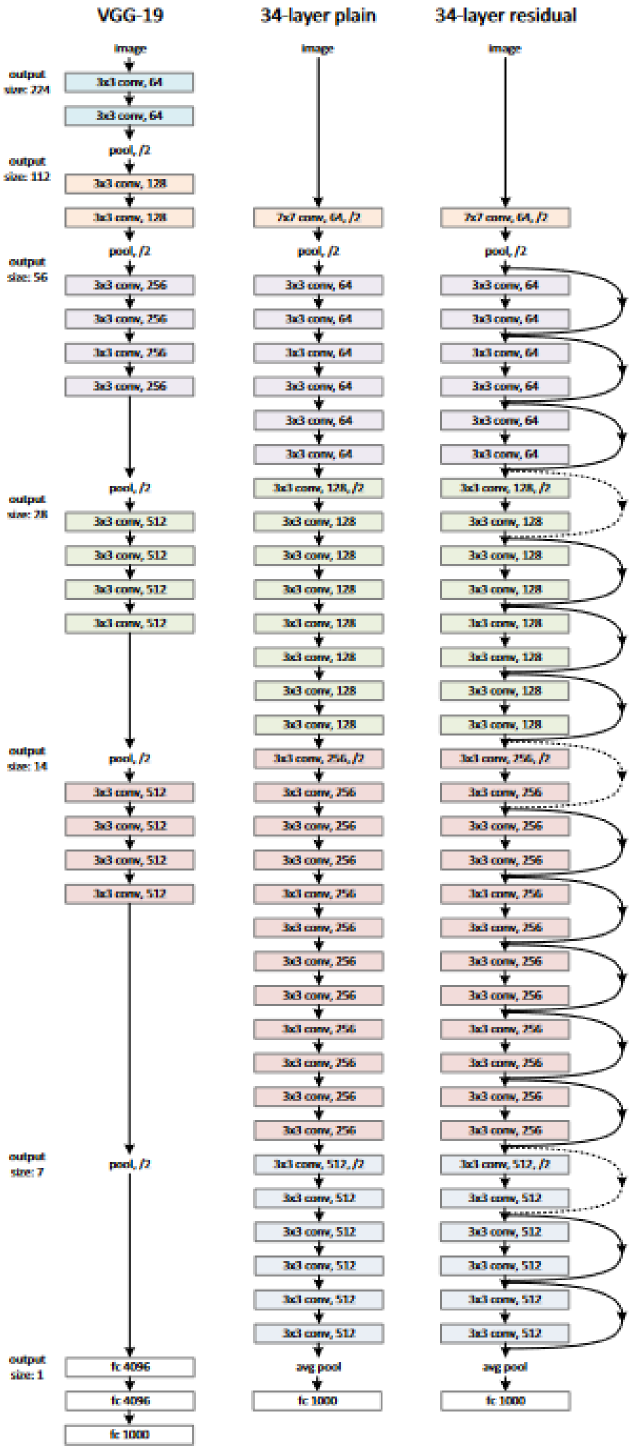


Figure 3.4: ResNet34 :Reprinted From [1]

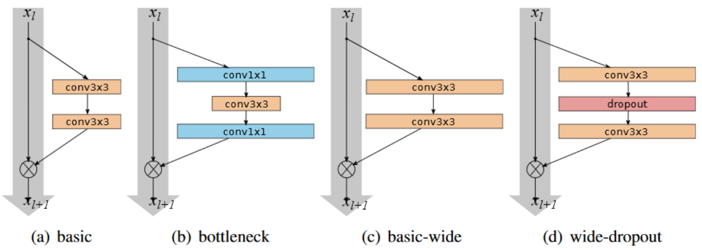


Figure 3.5: WideResNet: Reprinted From [2]

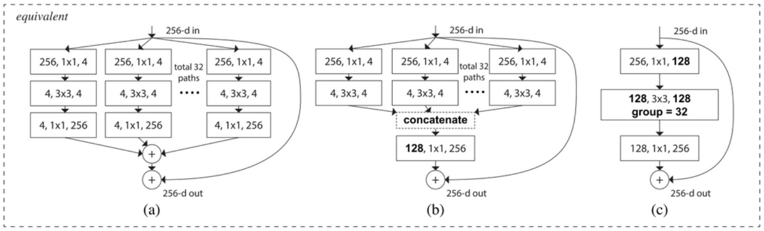


Figure 3.6: ResNext: Reprinted From [3]

3.1.4 DenseNet [4]

The last Model, in the family of vastly successful ResNet Family has been the DenseNet [4], which was the latest model to come out, beating all the previous models in most of the scenarios in the classification task. DenseNet [4] model looks to exploit the residual connections to an even greater extent by, connecting each layer with every other layer directly. This definitely make the model more convoluted and harder to interpret, but nevertheless outperformed other models unequivocally. Figure 3.7 explicitly shows how each layers are connected to each other in a typical DenseNet Model. However, Connecting every other layer with all the layers, brings with itself unnecessary complications, which the paper solves via proposing a comparatively simpler model, where only layers in a group are connected to each other, and a group consists of a predefined number of layers, which is a hyper-parameter. Figure 3.8 shows the simpler version of Original DenseNet model, which is actually being used.

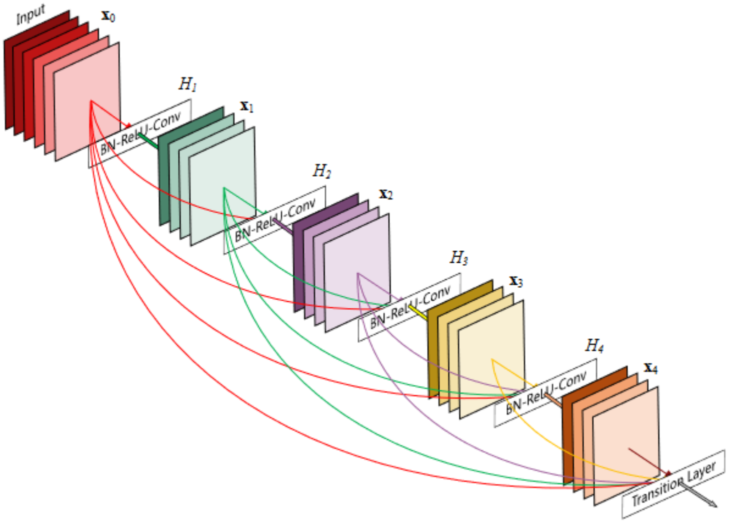


Figure 3.7: DenseNet Architecture: Reprinted From [4]

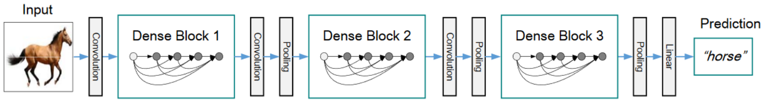


Figure 3.8: DenseNet Model: Reprinted From [4]

3.2 Deriving New Orthogonality Regularizations

† In this section, we will derive and discuss several orthogonality regularizers. Note that those regularizers are applicable to both fully-connected and convolutional layers. The default mathematical expressions of regularizers will be assumed on a fully-connected layer $W \in {}^{m\times n}$ (m could be either larger or smaller than n). For a convolutional layer $C \in {}^{S\times H\times C\times M}$, where S, H, C, M are filter width, filter height, input channel number and output channel number, respectively, we will first reshape C into a matrix form $W' \in m' \times n'$, where $m' = S \times H \times C$ and $n' = M$. All our regularizations are directly amendable to almost any CNN: there is no change needed on the network architecture, nor any other training protocol (unless otherwise specified).

†Reprinted with permission from Deriving New Orthogonality Regularizations section of *Can We Gain More from Orthogonality Regularizations in Training Deep Networks?* by N. Bansal, X. Chen and Z. Wang, 2018,Advances in Neural Information Processing Systems 31 (NIPS 2018) pre-proceedings

3.2.1 Baseline: Soft Orthogonality Regularization

[29] proposed to require the Gram matrix of the weight matrix to be close to identity, which we term as Soft Orthogonality (SO) regularization:

$$\text{(SO)} \qquad \lambda||W^TW - I||_F^2, \tag{3.1}$$

where λ is the regularization coefficient (the same hereinafter). It is a straightforward relaxation from the "hard orthogonality" assumption [27, 28, 30] under the standard Frobenius norm, and can be viewed as a different weight decay term limiting the set of parameters close to a Stiefel manifold rather than inside a hypersphere. The gradient is given in an explicit form: $4\lambda W(W^TW - I)$, and can be directly appended to the original gradient w.r.t. the current weight W.

However, SO (3.1) is flawed for an obvious reason: the columns of W could possibly to mutually orthogonal, if and only if W is under-complete ($m \leq n$). For over-complete W ($m > n$), its gram matrix $\in m \times m$ cannot be even close to identity, because its rank is at most n, making $||W^TW - I||_F^2$ a biased minimization objective. In practice, both cases can be found for layer-wise weight dimensions. The authors of [30, 29] advocated to further divide over-complete W into under-complete column groups to resolve the rank deficiency trap. In this paper, we choose to simply use the original SO version (3.1) as a fair comparison baseline.

The authors of [29] argued against the hybrid utilization of the original ℓ_2 weight decay and the SO regularization. They suggested to stick to one type of regularization all along training. Our experiments also find that applying both together throughout training will hurt the final accuracy. Instead of simply discarding ℓ_2 weight decay, we discover a *scheme change* approach which is validated to be most beneficial to performance, details on this can be found in Section 4.1.

3.2.2 Double Soft Orthogonality Regularization

The double soft orthogonality regularization extends SO in the following form:

$$\text{(DSO)} \qquad \lambda(||W^TW - I||_F^2 + ||WW^T - I||_F^2). \tag{3.2}$$

Note that an orthogonal W will satisfy $W^TW = WW^T = I$; an ove-rcomplete W can be regularized to have small $||WW^T - I||_F^2$ but will likely have large residual $||W^TW - I||_F^2$, and vice versa for an under-complete W. DSO is thus designed to cover both over-complete and under-complete W cases; for either case, at least one term in (3.2) can be well suppressed, requiring either rows or columns of W to stay orthogonal. It is a straightforward extension from SO.

Another similar alternative to DSO is "selective" soft orthogonality regularization, defined as: $\lambda||W^TW - I||_F^2$, if $m > n$; $\lambda||WW^T - I||_F^2$ if $m \leq n$. Our experiments find that DSO always outperforms the selective regularization, therefore we only report DSO results.

3.2.3 Mutual Coherence Regularization

The mutual coherence [33] of W is defined as:

$$\mu_W = \max_{i \neq j} \frac{|\langle w_i, w_j \rangle|}{||w_i|| \cdot ||w_j||}, \tag{3.3}$$

where w_i denotes the i-th column of W, $i = 1, 2, ..., n$. The mutual coherence (3.3) takes values between [0,1], and measures the highest correlation between any two columns of W. In order for W to have orthogonal (if $m \leq n$) or near-orthogonal (if $m > n$) columns, μ_W is expected to be as low as possible (zero if $m \leq n$).

We wish to suppress μ_W as an alternative way to enforce orthogonality. Assume W has been first normalized to have unit-norm columns, $\langle w_i, w_j \rangle$ is essentially the (i, j)-the element of the Gram matrix W^TW, and $i \neq j$ requires us to consider off-diagonal elements only. Therefore, we could equivalently re-express (3.3) into the following Mutual Coherence (MC) regularization term:

$$\text{(MC)} \qquad \lambda||W^TW - I||_\infty, \tag{3.4}$$

In practice, we implement (3.4) using two-stage components. We first fulfill the necessary preprocessing of normalizing $||W||_2 = 1$, via inserting a standard weight normalization with g fixed as 1 [36]. It implies an interesting possible implication of WN on the CNN weight orthogonality. Next, we add the $\lambda||W^TW - I||_\infty$ regularizer when computing the gradient w.r.t. W. Although we do not explicitly normalize the column norm of W to be one, we find experimentally that minimizing (3.4) often tends to implicitly encourage close-to-unit-column-norm W too, making the objective of (3.4) a viable approximation of mutual coherence (3.3)*.

The gradient of $||W^TW - I||_\infty$ could be explicitly solved by applying a smoothing technique to the non-smooth ℓ_∞ norm, e.g., [49]. However, it will invoke an iterative routine each time to compute ℓ_1-ball proximal projection, which is less efficient in our scenario where massive gradient computations are needed. In view of that, we turn to using auto-differentiation to approximately compute the gradient of (3.4) w.r.t. W.

3.2.4 Spectral Restricted Isometric Property Regularization

Recall that the RIP condition [32] of W assumes:

Assumption 1. *For all vectors $z \in ^n$ that is k-sparse, there exists a small $\delta_W \in (0, 1)$ s.t. $(1-\delta_W) \leq \frac{||Wz||^2}{||z||^2} \leq (1 + \delta_W)$.*

The above RIP condition essentially requires that every set of columns in W, with cardinality no larger than k, shall behave like an orthogonal system. If taking an extreme case with $k = n$, RIP then turns into another criterion that enforces the entire W to be close to orthogonal. Note that both mutual incoherence and RIP are well defined for both under-complete and over-complete matrices. We rewrite the special RIP condition with $k = n$ in the form below:

$$\frac{||Wz||^2}{||z||^2} - 1| \leq \delta_W, \; \forall z \in \mathbb{R}^n \tag{3.5}$$

Notice that $\sigma(W) = \sup_{z\in\mathbb{R}^n, z\neq\mathbf{0}} \frac{||Wz||}{||z||}$ is the spectral norm of W, i.e., the largest singular value of W. As a result, $\sigma(W^TW - I) = \sup\limits_{z\in\mathbb{R}^n, z\neq 0} |\frac{||Wz||^2}{||z||^2} - 1|$. In order to enforce orthogonality to W from an RIP perspective, one may wish to minimize the RIP constant δ_W in the special case $k = n$, which according to the definition should be chosen as $\sup\limits_{z\in\mathbb{R}^n, z\neq 0} |\frac{||Wz||^2}{||z||^2} - 1|$ as from (3.5). Therefore,

*We also tried to first normalize columns of W and then apply (3.4), without finding any performance benefits.

we end up equivalently minimizing the spectral norm of $W^TW - I$:

$$\text{(SRIP)} \qquad \lambda \cdot \sigma(W^TW - I). \tag{3.6}$$

It is termed as the Spectral Restricted Isometry Property (SRIP) regularization.

The above reveals an interesting hidden link: regularizations with spectral norms were previously investigated in [8, 46], through analyzing small perturbation robustness and Lipschitz constant. The spectral norm re-arises from enforcing orthogonality when RIP condition is adopted. But compared to the spectral norm (SN) regularization [8] which minimizes $\sigma(W)$, SRIP is instead enforced on $W^TW - I$. Also compared to [46] requiring the spectral norm of W to be exactly 1 (developed for GANs), SRIP requires *all singular values of W to be close to 1*, which is essentially **stricter**.

We again refer to auto differentiation to compute the gradient of (3.6) for simplicity. However, even computing the objective value of (3.6) can invoke the computationally expensive EVD. To avoid that, we approximate the computation of spectral norm using the power iteration method. Starting with a randomly initialized $v \in \mathbb{R}^n$, we iteratively perform the following procedure a small number of times (2 times by default) :

$$u \leftarrow (W^TW - I)v, v \leftarrow (W^TW - I)u, \sigma(W^TW - I) \leftarrow \frac{||u||}{||v||}. \tag{3.7}$$

3.3 Scheme Change for Regularization Coefficients

All the regularizers have an associated regularization coefficient denoted by λ, whose choice play an important role in the regularized training process. Correspondingly, we denote the regularization coefficient for the ℓ_2 weight decay used by original models as λ_2.

From experiments, we observe that fully replacing ℓ_2 weight decay with orthogonal regularizers will accelerate and stabilize training at the beginning of training, but will negatively affect the final accuracies achievable. We conjecture that while the orthogonal parameter structure is most benefit at the initial stage, it might be overly strict when training comes to the final "fine" stage, when we should allow for more flexibility for parameters. In view of that, we devise a switch scheme between

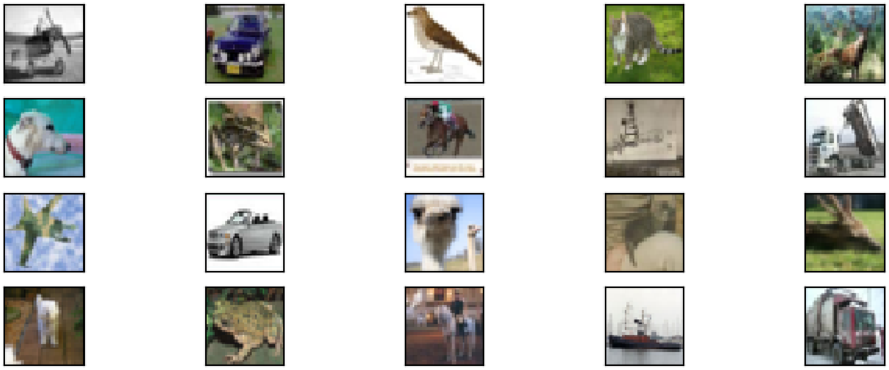


Figure 3.9: Sample Images: CIFAR 10

two regularization schemes at the beginning and late stages of training. Concretely, we gradually reduce λ (initially 0.1) by factors of 1e-3, 1e-4 and 1e-6, after 20, 50 and 70 epochs, respectively, and finally set it to zero after 120 epochs. For λ_2, we start with 1e-8; then for SO/DSO regularizers, we increase λ_2 to 1e-4/5e-4, after 20 epochs. For MC/SRIP regularizers, we find them insensitive to the choice of λ_2, potentially due to their stronger effects in enforcing W^TW close to I; we thus stick to the initial λ_2 throughout training for them. Such an empirical "scheme change" design is found to work nicely with all three models, benefiting both accuracy and efficiency.

3.4 Datasets

3.4.1 CIFAR 10 [5]

The CIFAR-10 dataset consists of 60000 32x32 color images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. The dataset is divided into five training batches and one test batch, each with 10000 images. The test batch contains exactly 1000 randomly-selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain exactly 5000 images from each class. Figure 3.9, shows a sample of the images in the CIFAR100 Dataset, these images were self generated.

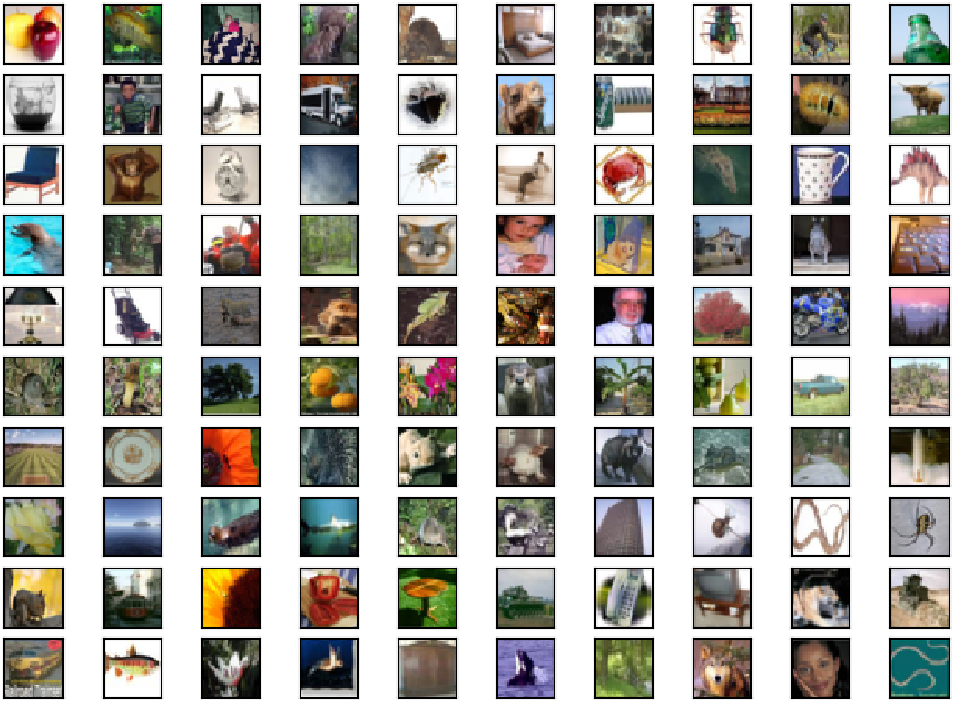


Figure 3.10: Sample Images: CIFAR 100

3.4.2 CIFAR 100 [5]

This dataset is similar to CIFAR-10 in terms of Image distribution, except it has 100 classes containing 600 images each. There are 500 training images and 100 testing images per class. The 100 classes in the CIFAR-100 are grouped into 20 super-classes. Each image comes with a "fine" label (the class to which it belongs) and a "coarse" label (the superclass to which it belongs).Figure 3.10, shows a sample of the images in the CIFAR100 Dataset, these images were self generated.

3.4.3 SVHN [6]

SVHN is a real-world image dataset for developing machine learning and object recognition algorithms with minimal requirement on data preprocessing and formatting. It can be seen as similar in flavor to MNIST (e.g., the images are of small cropped digits), but incorporates an order of magnitude more labeled data (over 600,000 digit images) and comes from a significantly harder, unsolved, real world problem (recognizing digits and numbers in natural scene images). SVHN is obtained from house numbers in Google Street View images. Figure 3.11, shows a sample of the images in the SVHN Dataset.

3.4.4 IMAGENET [7]

ImageNet is a Image database, which uses a model, which is similar to what is used for WordNet schema, to categorize different objects. The database was presented as a poster in 2009 CVPR. ImageNet still serves as the benchmark, for evaluating the performance of models which primarily has task such as Image Classification and Object Detection.

ImageNet by itself contains 14 million URLs of images. The annotation of the images is manually, and uses crowd-annotation tools such as Mechanical turk.For the Object detection task along with labels about the class of object, it provides bounding boxes for almost 1 million images. ImageNet 2012 classification dataset that consists of 1000 classes. The models are trained on the 1.28 million training images, and evaluated on the 50k validation images. There also has been an annual competition known as Imagenet Large Scale Visual Recognition Challenge(ILSVRC), where the existing state-of-art models for Image Classification and Object detection are applied on Imagenet Dataset, to see and verify its efficacy and to set new benchmark in terms of performance. Figure 3.12, shows a sample of the images in the Imagenet Dataset.

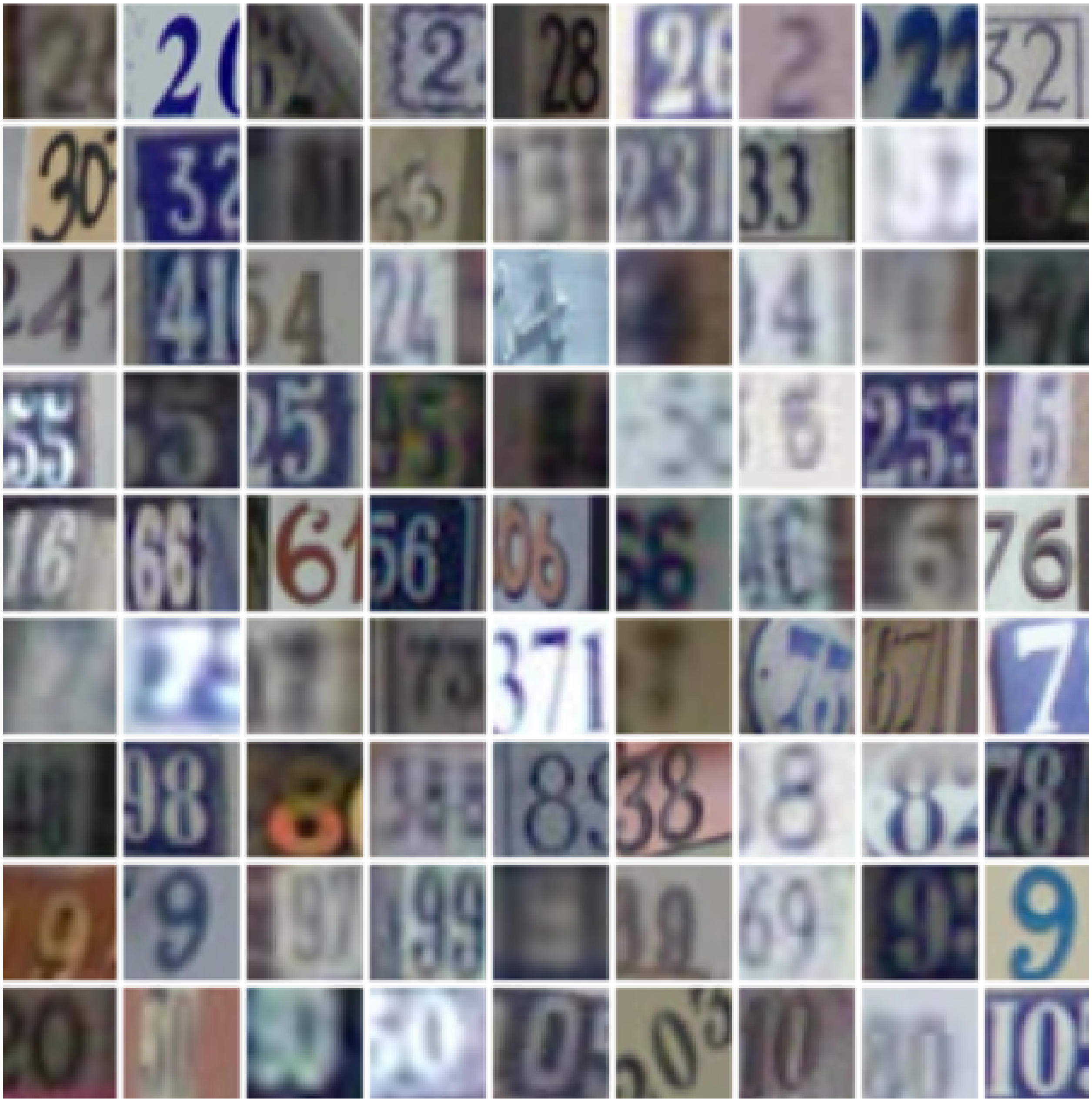


Figure 3.11: Sample Images: SVHN: Reprinted From [12]

Figure 3.12: Sample Images: Imagenet:Reprinted From [13]

4. RESULTS AND OBSERVATIONS

4.1 CIFAR10/100 Datasets

[†] We first explain the experimental set up used to verify the efficacy of different regularizers. We choose three popular state-of-the-art models: ResNet[1], Wide ResNet[2] and ResNext[3]. We will train each of the models with each of our proposed regularizers, and comparing their performance with the original version, in terms of both final achieved accuracy and convergence speed/stability. We choose CIFAR-10 and CIFAR-100 as our primary testbeds. They consist of 60,000 images of size 32×32 with a 5-1 training-testing split, divided into 10 and 100 classes respectively. All pre-processing and data augmentation are strictly identical to the original training protocols in [2, 1, 3]. All hyper-parameters and architectural details remain unchanged too, unless otherwise specified.

4.1.1 ResNet 110 - BottleNeck Residual Unit Model [1]

We employ a 110-layer ResNet Model [1], a very strong ResNet version, to evaluate our proposed regularizers here. The ResNet architecture uses Bottleneck Residual Units, with a formula setting given by $p = 9n + 2$, where *n* denotes the total number of convolutional blocks used and *p* the total depth of the model. We use the Adam optimizer to train the model for 200 epochs, with learning rate varied: starting with 1e-2, and then subsequently decreasing to 1e-3, 1e-5 and 1e-6, after 80, 120 and 160 epochs, respectively.

Several meaningful observations are found from Table 4.1. First, SRIP gives very competitive results: the best on CIFAR-10 and second best on CIFAR-100. Second, SO provides a surprisingly strong baseline here too, with second best on CIFAR-10 and the best on CIFAR-100. Besides, MC outperforms the original baseline and DSO (in the case even worse than original baseline), but remains inferior to SRIP and SO. Overall, we are able to gain as much as 0.49$\%$ on CIFAR-10

[†] Reprinted with permission from Experiments on Benchmark section of *Can We Gain More from Orthogonality Regularizations in Training Deep Networks?* by N. Bansal, X. Chen and Z. Wang, 2018,Advances in Neural Information Processing Systems 31 (NIPS 2018) pre-proceedings

Table 4.1: Top-1 Error Rates Achieved: ResNet 110 For CIFAR 10 and CIFAR 100. * Indicates Results During Our Experiment

Model	Depth-K-Cardinality	CIFAR-10	CIFAR-100	Regularizer
ResNet-110 [1]	110	7.04*	25.42*	None (original)
	110	6.78	**25.01**	SO
	110	7.04	25.83	DSO
	110	6.97	25.43	MC
	110	**6.55**	25.14	SRIP

(SRIP) and 0.41% on CIFAR-100 (SO), by simply enforcing orthogonality regularizations.

Figures 4.1 plots the training curves (in terms of validation accuracies w.r.t epoch numbers) of different methods on CIFAR-10 and CIFAR-100. We observe that all four regularized models have their training curves grow much faster in the initial training stage, and stay at higher accuracies throughout (most part of) training, compared to the un-regularized original version. Those regularized curves also tend be show less fluctuations, and grow more stably and smoothly. This observation becomes more stark in the region between epochs 20-80. Among them all, SRIP shows to provide the best training stability and efficiency, in addition to the highest/second highest final accuracies achieved.

4.1.2 Wide ResNet 28-10 Model [2]

For the Wide ResNet model [2], we have used depth 28 and k (width) 10, as this configuration gives the best accuracies for both CIFAR-10 and CIFAR-100, and is (relatively) computationally efficient. The model uses a Basic Block B(3,3), as defined in ResNet [1]. We have used the SGD optimizer with a Nesterov Momentum of 0.9 to train the model for 200 epochs. The learning rate starts at 1e-1, and is then decreased by a factor of 0.2, after 60, 120 and 160 epochs, respectively. We have followed all other settings of [2] identically. As shown in the Table 4.2, all four orthogonal regularizers significantly boost the accuracies over the original result. SRIP is the best performer in both datasets, giving rise to impressive **0.50%** and **2.47%** improvements on CIFAR-10 and

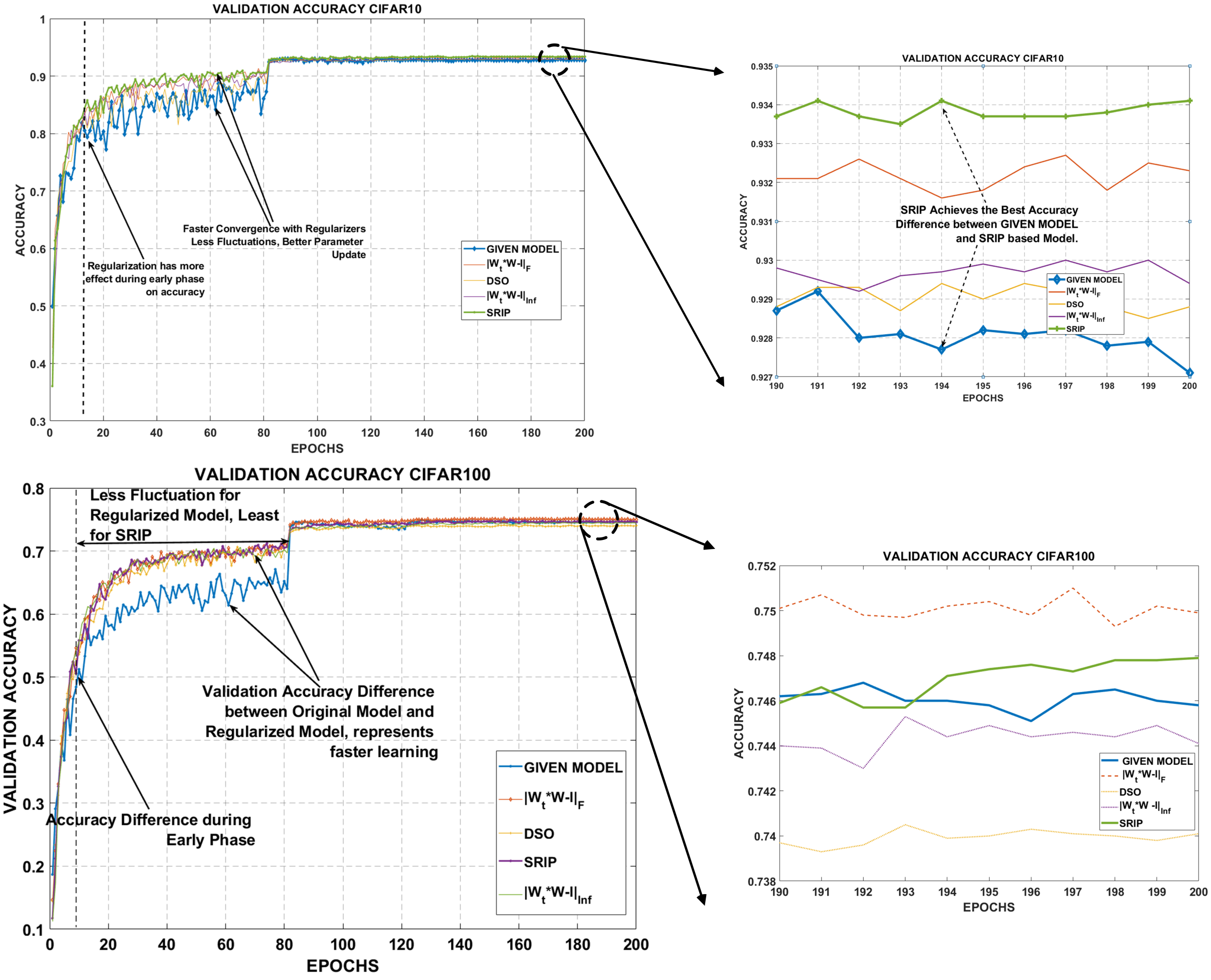
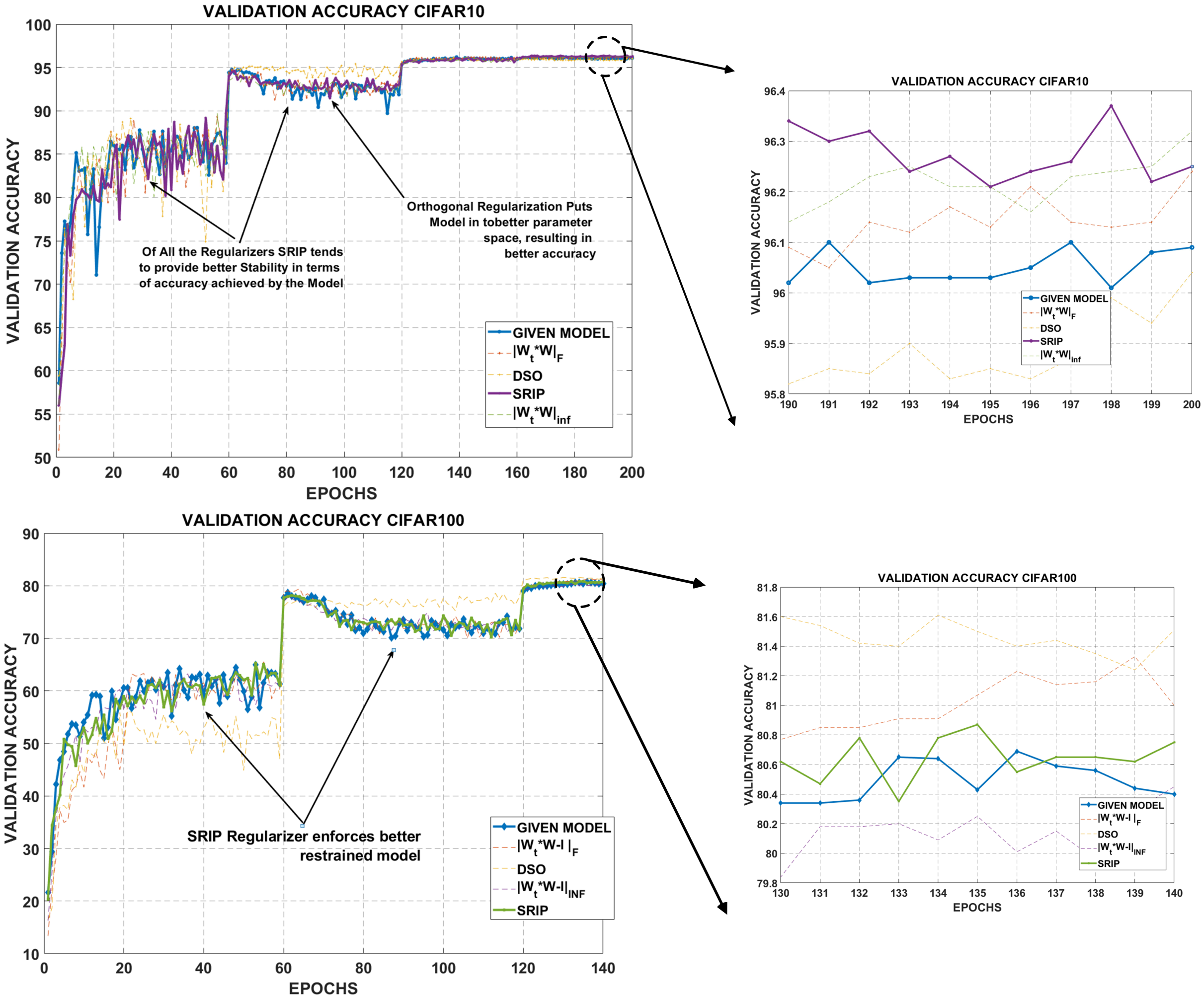


Figure 4.1: Validation Curves:ResNet-110. Top: CIFAR 10; Bottom: CIFAR 100



Of All the Regularizers SRIP tends to provide better Stability in terms of accuracy achieved by the Model

Orthogonal Regularization Puts Model in tobetter parameter space, resulting in better accuracy

SRIP Regularizer enforces better restrained model

Figure 4.2: Validation Curves:WideRes 28-10. Top: CIFAR 10; Bottom: CIFAR 100

Table 4.2: Top-1 Error Rates Achieved: Wide ResNet For CIFAR-10 and CIFAR-100. * Indicates Results During Our Experiments

Model	Depth-K-Cardinality	CIFAR-10	CIFAR-100	Regularizer
Wide ResNet[2]	28-10	4.16*	20.50*	None (original)
	28-10	3.76	18.56	SO
	28-10	3.86	18.12	DSO
	28-10	3.68	18.90	MC
	28-10	**3.66**	**18.03**	SRIP
	28-10	3.73	18.76	OMDSM [30]
	28-10	3.93	19.08	SN [8]

Table 4.3: Top-1 Error Rates Achieved: ResNext For CIFAR 10 and CIFAR 100. * Indicates Results During Our Experiments

Model	Depth-K-Cardinality	CIFAR-10	CIFAR-100	Regularizer
ResNext[3]	29-8-64	3.70*	18.53*	None (original)
	29-8-64	3.58	17.59	SO
	29-8-64	3.85	19.78	DSO
	29-8-64	3.65	17.62	MC
	29-8-64	**3.48**	**16.99**	SRIP
	29-8-64	3.54	17.27	SN [8]

CIFAR-100, respectively. MC performs only next to SRIP on CIFAR-10, while DSO performs the second best on CIFAR-100. SO still seems to be reasonably robust, ranking third among the four on both datasets. Figure 4.2 displays similar tendency as enforcing orthogonality makes training/validation curves smoother and more stable, and SRIP has the most positive impact in that regard. We also include the recent results (average accuracies) reported by [30] on improving Wide ResNet using hard Stiefel manifold constraints (**OMDSM**), which makes a fair comparison with ours, on soft regularization forms versus hard constraint forms of enforcing orthogonality. The OMDSM results lead to competitive accuracies, but are inferior to SRIP on both CIFAR-10 and CIFAR-100.

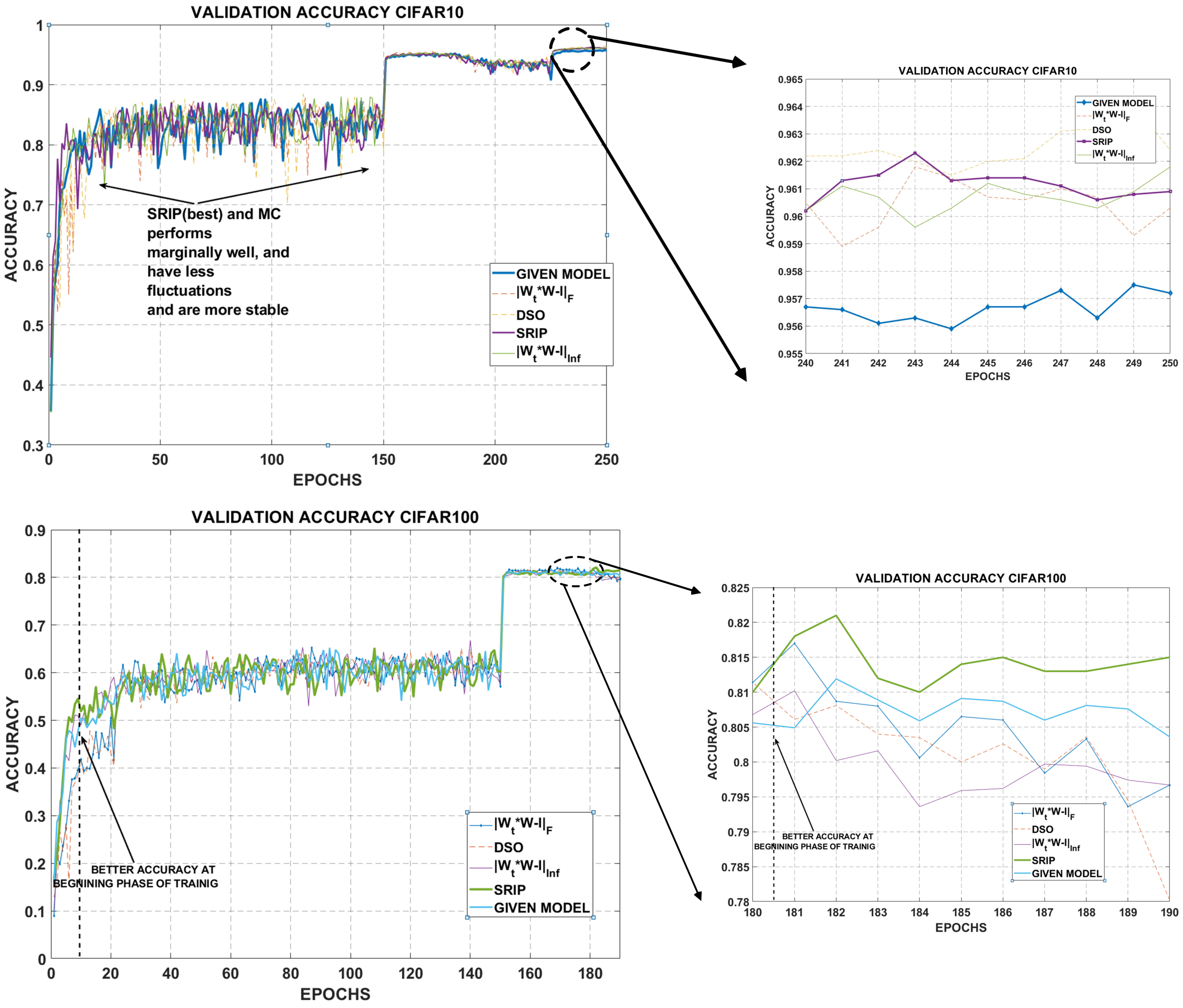


Figure 4.3: Validation Curves: Resnext. Top: CIFAR 10; Bottom: CIFAR 100

4.1.3 ResNext 29-8-64 Model [3]

For ResNext Model [3], we consider the 29-layer architecture with a cardinality of 8 and widening factor as 4, which reported the best state-of-the-art CIFAR-10/CIFAR-100 results compared to other contemporary models with similar amounts of trainable parameters. We use SGD optimizer with a Nesterov Momentum of 0.9 to train the model for 300 epochs. The learning starts from 0.1, and decays by a factor of 0.1 after 150 and 225 epochs, respectively

Table 4.3 shows that all orthogonality regularizers give rise to improved accuracies, except for DSO. SRIP is again the best performer among all in both datasets, with 0.22% gain on CIFAR-10 and 1.54% on CIFAR-100, respectively.Figure 4.3 shows the validation curve obtained for both the CIFAR Dataset. Similarly, SRIP is observed to also improve stability and convergence speed of training/validation curves (more notably on CIFAR-100), in particular at the initial stage.

4.1.4 Comparing SRIP with Spectral Regularization [8]

We compare the spectral norm (**SR**) regularization developed in [8]: $\frac{\lambda_s}{2}\sigma(W)^2$, with the authors' default $\lambda_s = 0.1$. All other settings in [8] have been followed identically. We apply the SN regularization to training the Wide ResNet-28-10 Model and the ResNext 29-8-64 Model. For the former, we obtain a top-1 error rate of 3.93% on CIFAR-10, and 19.08% on CIFAR-100. For the latter, the top-1 error rate is 3.54% for CIFAR-10, and 17.27% for CIFAR-100. As shown in Tables 4.2 and 4.3, we find that SRIP gives a consistent marginal improvement over SN in all cases. That reminds us of the previous discussion on the hidden link between the two, and suggests SRIP to be the better choice. Besides, both SN and SRIP are observed to lead to stabilized and smoother training curves.

4.1.5 Comparing SRIP with Jacobian based Regularization [9]

There has been a recent work on CNNs [9],which propounds the idea of using Norm of the Jacobian as a regularizer to the the model . The paper uses a Wide ResNet [2] with 22 layers of width 5, on CIFAR-10, which achieves an error rate **6.66%**, and with their proposed regularizer **5.68%**. We trained this model using SRIP over the same augmented full training set, achieving **4.28%** error, that shows a large margin over the CNN gradient norm-based regularizer.

Table 4.4: Top-5 Error Rates Achieved:ImageNet and (Top-1 Error) SVHN. * Indicates Results During Our Experiments

Model	Depth-K	ImageNet	Regularizer
ResNet34[1]	34	9.84	None
	34	9.68	OMDSM [30]
	34	8.32	SRIP
ResNet50[1]	50	7.02	None
	50	6.87	SRIP
Pre-Resnet34[11]	34	9.79	None
	34	9.45	OMDSM [30]
	34	8.79	SRIP
Model	Depth-K	SVHN	Regularizer
WideResNet[2]	16-8	1.63	None
	16-8	1.56	SRIP

4.2 ImageNet and SVHN Dataset

We ran extensive experiments on Imagenet Dataset for different configuration of Resnet[1], Pre-Resnet[11] and WideResnet[2] architectures. Results obtained with these architectures were later compared with, existing regularization methods. The experimental details and hyper-parameter settings for Imagenet dataset were kept consistent with the original model [1]. The Initial learning rate is set to 0.1, which is decreased at epoch 30,60,90 and 120 by a factor of 0.1. For all the Imagenet experiments, the initial value of λ constant was changed to 1e-6. The results achieved with ResNet34 and Pre-ResNet34 [1] were compared with the work in [30] ,achieving better Top-5 final accuracies for both ResNet34 and Pre-ResNet34 models. For Experiments related to SVHN, initial value of λ constant was made similar to that used for CIFAR dataset, which is 1e-2, and we used WideResnet [2] architecture to perform the experiments. The Table 4.4 summaries all the observation related to both the dataset under different settings.

5. SUMMARY AND INFERENCES

5.1 Summary Remarks and Insights

† From the extensive experiments with three state-of-the-art models on two popular benchmarks, we can conclude the following points:

- In response to the question in our title: *Yes, we can gain a lot from simply adding orthogonality regularizations into training*. The gains can be found in both final achievable accuracy and empirical convergence.

  For the former, the three models have obtained (at most) 0.49%, 0.50%, and 0.22% top-1 accuracy gains on CIFAR-10, and 0.41%, 2.47%, and 1.54% on CIFAR-100, respectively. For the latter, positive impacts are widely observed in our training and validation curves (Figure 4.1 as a representative example), in particular faster and smoother curves at the initial stage. Note that those impressive improvements are obtained with no other changes made.

- With its nice theoretical grounds, SRIP is also the best practical option among all four regularizations evaluated in this paper. It consistently performs the best in achieving the highest accuracy as well as accelerating/stabilizing training curves. It also outperforms other recent methods utilizing spectral norm [8] and hard orthogonality [30].

- Despite its simplicity (and potential estimation bias), SO is a surprisingly robust baseline and frequently ranks second among all four. We conjecture that SO benefits from its smooth form and continuous gradient, which facilitates the gradient-based optimization, while both SRIP and MC have to deal with non-smooth problems.

- DSO does not seem to be helpful. It often performs worse than SO, and sometimes even worse than the un-regularized original model. We interpret it by recalling how the matrix W

†Reprinted with permission from Summary Remarks and Insights section of *Can We Gain More from Orthogonality Regularizations in Training Deep Networks?* by N. Bansal, X. Chen and Z. Wang, 2018,Advances in Neural Information Processing Systems 31 (NIPS 2018) pre-proceedings

is constructed (Section 3 beginning): enforcing W^TW close to I has "inter-channel" effects (i.e., requiring different output channels to have orthogonal filter groups); whereas enforcing WW^T close to I enforce "intra-channel" orthogonality (i.e., same spatial locations across different filter groups have to be orthogonal). The former is a better accepted idea. Our results on DSO seems to provide further evidence (from the counter side) that orthogonality should be primarily considered for "inter-channel", i.e., between columns of W.

- MC brings in certain improvements, but not as significantly as SRIP. We notice that (3.4) will approximate (3.3) well only when W has unit columns. While we find minimizing (3.4) generally has the empirical results of approximately normalizing W columns, it is not exactly enforced all the time. As we observed from experiments, large deviations of column-wise norms could occur at some point of training and potentially bring in negative impacts. We plan to look for re-parameterization of W to ensure unit norms throughout training, e.g., through integrating MC with weight normalization [36], in future work.

- Comparison between SRIP and other existing Regularization techniques also seem to suggest, SRIP succeeds in providing better training parameter space. on CIFAR dataset, in [50], achieves an error rate of 5.68%, with our method we see an improvement of about 1.5%, on Wide-Resnet architecture.

- Verifying the efficacy of the model, we trained it on a large dataset such as ImageNet, for ResNet34[1], Pre-ResNet34[11] and ResNet 50[1], to see an improvement in terms of top-5 accuracy, from the model, which doesn't uses any regularization term.The improvement achieved ranges from 0.15% for ResNet 50 to as high as 1.5% for simple ResNet34. We also postulate that, as the number of layers are increased and Model becomes more deeper, the improvement would be more stark.

6. FUTURE WORK

6.1 Possible Extension of the Idea

The applications of this work, and the improvements achieved by the proposed regularizations could be used in various domain to improve the accuracy, stability and convergence of the model. We look to apply the same idea in both Sequence based models and vision based models to see the efficacy of our method and improvements it brings.We are primarily looking to see its effect on:

- Object Detection
- Person Re-identification
- Gan Training

6.1.1 Object Detection Task

There have been work in the past in the Object Detection task, as similar to Classification task, where enforcing orthogonality on the feature vectors, so that they respond to different classes independently has also shown improvements.All the state of art models [1],[2], [3], which have shown a subsequent Improvement in terms of Classification accuracy on CIFAR and ImageNet Dataset have followed up with verifying the efficacy of the proposed model with Object Detection task.

Specifically, Authors of [1] have run the model on PASCAL VOC [51] and COCO [52] dataset, on a model based on Faster R-CNN [14], with the backbone of the network changed from VGG [47] to Resnet. As an extension of our work, we look to employ the same idea,this time adding the new regularization term based on SRIP, in to the Backbone Network. As shown in the figure 6.1, In the first stage, we pass the Input Image, through a backbone network, which is a CNN model, used for Feature Extraction, named as *Conv Layers* in the figure. We look to enforce orthogonality to this network, and evaluate its performance both on region proposal generated and hence the final precision achieved.

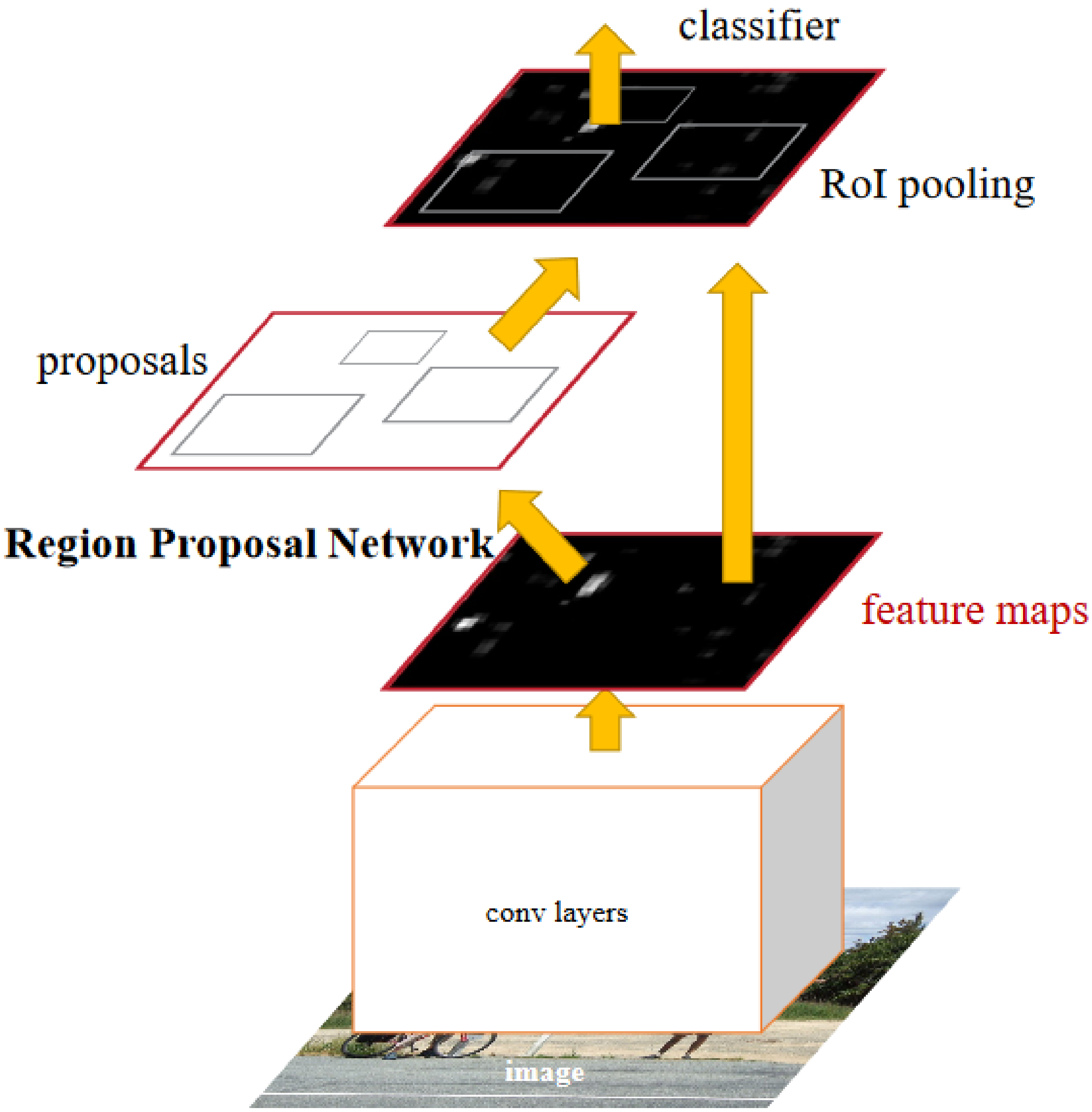


Figure 6.1: Faster RCNN:Reprinted From [14];

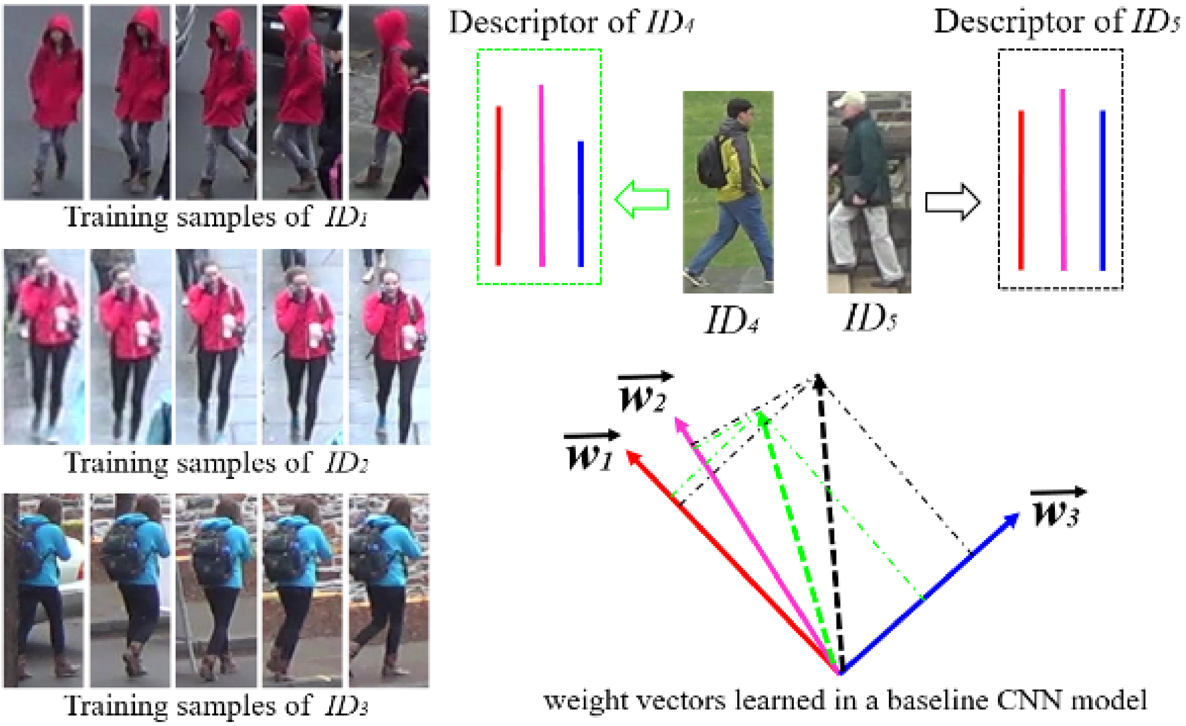

Training samples of ID_1
Training samples of ID_2
Training samples of ID_3
Descriptor of ID_4
Descriptor of ID_5
ID_4
ID_5
$\overrightarrow{w_2}$
$\overrightarrow{w_1}$
$\overrightarrow{w_3}$
weight vectors learned in a baseline CNN model

Figure 6.2: Weight Vectors Learned: Reprinted From [15];

6.1.2 Person Re-identification Task

Person Re-identification task, which is a task which comes in the ambit of task related to Face Recognition, could also use orthogonal property of weights to its advantage in improving its performance. Recent work in this domain, seems to validate this claim, as work shown in [15], shows improvement in re-ID accuracy for Market-1501, CUHK03, and Duke datasets. The model proposed propounds the same theory, emphasizing that enforcing orthogonality, encourages to produce more discriminative features which subsequently helps in performance gain. As shown in the figure 6.2, shows the feature weight vector learned for different persons, which could be made more discriminative by employing orthogonality. Paper proposes to use CaffeNet and ResNet architectures for training. We look to add our SRIP based regularizer to the base network, and see if it helps in further separating the features learned.

6.1.3 GAN Training Stability

Recent work in GANs and all its variant [53], have shown great promise in terms of generating target distribution, particularly for vision related tasks. One is also aware about the instability faced during training the GANs.Some of which are:

- Non-Convergence of Model Parameters, Where the Generator or the Discriminator never seems to converge.
- Unequal training,which leads to one of the Model becoming more powerful than other, hampers goal of the GAN. For Example, a strong Discriminator would lead to diminished gradient problem, where as a strong Generator could lead to a mode collapse, for a multi-mode target distribution.
- Training being *hyper-sensitive* to hyper parameters and regularizers.

Figure 6.3 shows a typical example of non-convergence of the value *xy* in a quintessential Nash equilibrium scenario. Miyato et al., proposed a method called spectral Normalization in [46], which constraints the singular values of the parameter matrix of the discriminator, which helps in regularizing the discriminator and hence curbing the problem of divergence and mode collapse. The method proposed in the model normalizes all the singular values of the weight matrix with, the largest singular value, using power iteration method. The model showed great promise in terms of achieving training stability and generating diverse image for CIFAR and ImageNet Dataset. Since, our best performing regularizer, which also penalizes in accordance with spectral norm, we think, enforcing this to either Discriminator or Generator or both should help stabilize the training of GANs. We are currently working towards replicating the results for DCGAN [54] on CelebA dataset and seeing the effect of our method compared to Spectral Normalization Method.

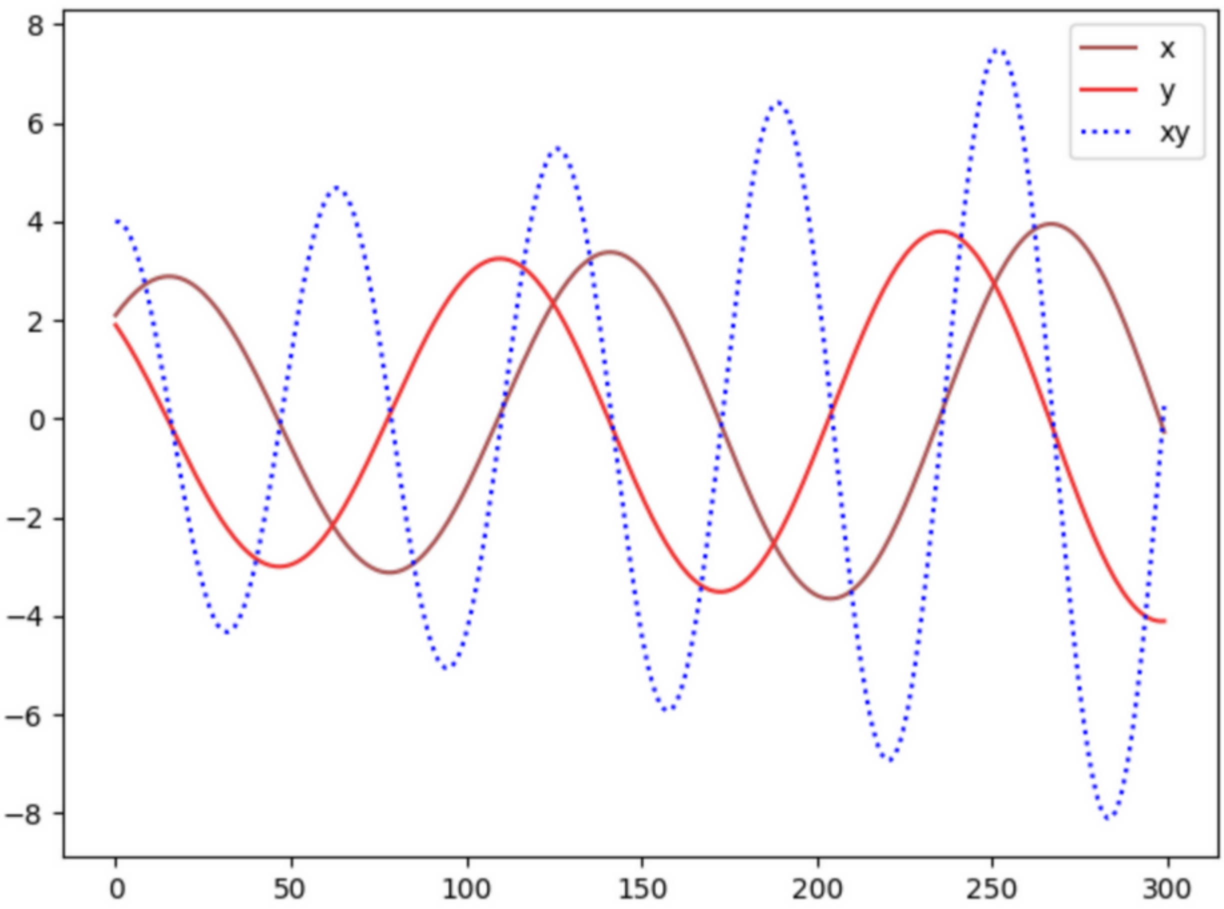


Figure 6.3: Nash Equilibrium: Players x and y: Reprinted From [16]

REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[2] S. Zagoruyko and N. Komodakis, “Wide residual networks,” *arXiv preprint arXiv:1605.07146*, 2016.

[3] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks,” in *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pp. 5987–5995, IEEE, 2017.

[4] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, “Densely connected convolutional networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, vol. 1, p. 3, 2017.

[5] A. Krizhevsky, “Learning multiple layers of features from tiny images.” `https://www.cs.toronto.edu/~kriz/cifar.html`, 2012.

[6] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng 2011.

[7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A Large-Scale Hierarchical Image Database,” in *CVPR09*, 2009.

[8] Y. Yoshida and T. Miyato, “Spectral norm regularization for improving the generalizability of deep learning,” *arXiv preprint arXiv:1705.10941*, 2017.

[9] J. Sokolic, R. Giryes, G. Sapiro, and M. R. Rodrigues, “Robust large margin deep neural networks,” *IEEE Transactions on Signal Processing*, vol. 65, Aug 2017.

[10] V. Fung, “Resnet and its variant models.” `https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035`, 2017.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," *CoRR*, vol. abs/1603.05027, 2016.

[12] "Sample pictures of svhn." `http://ufldl.stanford.edu/housenumbers`.

[13] "Sample pictures of imagenet." `https://www.crit-research.it`.

[14] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, (Cambridge, MA, USA), pp. 91–99, MIT Press, 2015.

[15] Y. Sun, L. Zheng, W. Deng, and S. Wang, "Svdnet for pedestrian retrieval," *CoRR*, vol. abs/1703.05693, 2017.

[16] J. Hui, "Why is it hard to train generative adversarial networks," 2018.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[18] X. Glorot and Y. 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.

[19] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, pp. 448–456, 2015.

[20] Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization," in *Advances in neural information processing systems*, pp. 2933–2941, 2014.

[21] A. M. Saxe, J. L. McClelland, and S. Ganguli, "Exact solutions to the nonlinear dynamics of learning in deep linear neural networks," *arXiv preprint arXiv:1312.6120*, 2013.

[22] J. Zhou, M. N. Do, and J. Kovacevic, “Special paraunitary matrices, cayley transform, and multidimensional orthogonal filter banks,” *IEEE Transactions on Image Processing*, vol. 15, no. 2, pp. 511–519, 2006.

[23] P. Rodríguez, J. Gonzalez, G. Cucurull, J. M. Gonfaus, and X. Roca, “Regularizing cnns with locally constrained decorrelations,” *arXiv preprint arXiv:1611.01967*, 2016.

[24] G. Desjardins, K. Simonyan, R. Pascanu, *et al.*, “Natural neural networks,” in *Advances in Neural Information Processing Systems*, pp. 2071–2079, 2015.

[25] D. Mishkin and J. Matas, “All you need is a good init,” *arXiv preprint arXiv:1511.06422*, 2015.

[26] K. Jia, D. Tao, S. Gao, and X. Xu, “Improving training of deep neural networks via singular value bounding,” *CoRR, abs/1611.06013*, 2016.

[27] M. Harandi and B. Fernando, “Generalized backpropagation,\'{E} tude de cas: Orthogonality,” *arXiv preprint arXiv:1611.05927*, 2016.

[28] M. Ozay and T. Okatani, “Optimization on submanifolds of convolution kernels in cnns,” *arXiv preprint arXiv:1610.07008*, 2016.

[29] D. Xie, J. Xiong, and S. Pu, “All you need is beyond a good init: Exploring better solution for training extremely deep convolutional neural networks with orthonormality and modulation,” *arXiv preprint arXiv:1703.01827*, 2017.

[30] L. Huang, X. Liu, B. Lang, A. W. Yu, and B. Li, “Orthogonal weight normalization: Solution to optimization over multiple dependent stiefel manifolds in deep neural networks,” *arXiv preprint arXiv:1709.06079*, 2017.

[31] Y. Sun, L. Zheng, W. Deng, and S. Wang, “Svdnet for pedestrian retrieval,” *arXiv preprint*, 2017.

[32] E. J. Candes and T. Tao, “Decoding by linear programming,” *IEEE transactions on information theory*, vol. 51, no. 12, pp. 4203–4215, 2005.

[33] D. L. Donoho, “Compressed sensing,” *IEEE Transactions on information theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[34] T. Zhang, “Sparse recovery with orthogonal matching pursuit under rip,” *IEEE Transactions on Information Theory*, vol. 57, no. 9, pp. 6215–6221, 2011.

[35] K. He, X. Zhang, S. Ren, and J. 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.

[36] T. Salimans and D. 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.

[37] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks,” in *International Conference on Machine Learning*, pp. 1310–1318, 2013.

[38] V. Dorobantu, P. A. Stromhaug, and J. Renteria, “Dizzyrnn: Reparameterizing recurrent neural networks for norm-preserving backpropagation,” *arXiv preprint arXiv:1612.04035*, 2016.

[39] M. Arjovsky, A. Shah, and Y. Bengio, “Unitary evolution recurrent neural networks,” in *International Conference on Machine Learning*, pp. 1120–1128, 2016.

[40] Z. Mhammedi, A. Hellicar, A. Rahman, and J. Bailey, “Efficient orthogonal parametrisation of recurrent neural networks using householder reflections,” *arXiv preprint arXiv:1612.00188*, 2016.

[41] E. Vorontsov, C. Trabelsi, S. Kadoury, and C. Pal, “On orthogonality and learning recurrent networks with long term dependencies,” *arXiv preprint arXiv:1702.00071*, 2017.

[42] S. Wisdom, T. Powers, J. Hershey, J. Le Roux, and L. Atlas, “Full-capacity unitary recurrent neural networks,” in *Advances in Neural Information Processing Systems*, pp. 4880–4888, 2016.

[43] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[44] S. Wang, A.-r. Mohamed, R. Caruana, J. Bilmes, M. Plilipose, M. Richardson, K. Geras, G. Urban, and O. Aslan, “Analysis of deep neural networks with extended data jacobian matrix,” in *International Conference on Machine Learning*, pp. 718–726, 2016.

[45] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, “On large-batch training for deep learning: Generalization gap and sharp minima,” *arXiv preprint arXiv:1609.04836*, 2016.

[46] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, “Spectral normalization for generative adversarial networks,” *arXiv preprint arXiv:1802.05957*, 2018.

[47] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *CoRR*, vol. abs/1409.1556, 2014.

[48] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich *CoRR*, vol. abs/1409.4842, 2014.

[49] Z. Lin, C. Lu, and H. Li, “Optimized projections for compressed sensing via direct mutual coherence minimization,” *arXiv preprint arXiv:1508.03117*, 2015.

[50] J. Sokolić, R. Giryes, G. Sapiro, and M. R. D. Rodrigues, “Robust large margin deep neural networks,” *IEEE Transactions on Signal Processing*, vol. 65, pp. 4265–4280, Aug 2017.

[51] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The PASCAL Visual Object Classes (VOC) challenge,” *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[52] T.-Y. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *ECCV*, 2014.

[53] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems 27* (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds.), pp. 2672–2680, Curran Associates, Inc., 2014.

[54] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” *CoRR*, vol. abs/1511.06434, 2015.