Regression with Deep Neural Networks: Generalization Error Guarantees, Learning Algorithms, and Regularizers

Jaweria Amjad
Department of E&EE
University College London
London, UK
jaweria.amjad.16@ucl.ac.uk

Zhaoyan Lyu
Department of E&EE
University College London
London, UK
uceelyu@ucl.ac.uk

Miguel R.D. Rodrigues
Department of E&EE
University College London
London, UK
m.rodrigues@ucl.ac.uk

***Abstract*—We present new data-dependent characterizations of the generalization capability of deep neural networks based data representations within the context of regression tasks. In particular, we propose new generalization error bounds that depend on various elements associated with the learning problem such as the complexity of the data space, the cardinality of the training set, and the input-output Jacobian of the deep neural network. Moreover, building upon our bounds, we propose new regularization strategies constraining the network Lipschitz properties through norms of the network gradient. Experimental results show that our newly proposed regularization techniques can deliver state-of-the-art performance in comparison to established weight-based regularization.**

I. Introduction

Deep neural networks (DNN) have led to remarkable performance in various machine learning tasks [1]. These networks consist of multiple layers of simple linear affine transformations followed by point-wise non-linearities, therefore exhibiting huge capacities making them prone to significant overfitting as shown in a recent landmark paper [2]. However, considerable empirical evidence continues to showcase counter-intuitively that very deep models generalize remarkably well in various applications [3] .

Significant efforts are therefore being devoted by the scientific community to characterize the learning properties of deep neural networks in highly over-parameterized settings [4]. On the one hand, a number of theoretical oriented works have focused on characterizations of the generalization properties of DNNs by offering bounds on their generalization error [5]- [7]. On the other hand, empirical oriented works have also been conducted via a series of carefully crafted experiments to study the inductive bias of stochastic gradient descent algorithms [3]. Most of these works, however have focused on deep learning classifiers, with less attention being paid to the performance of deep neural networks for regression problems. In this paper, we make progress on this front whereby – by building upon the robustness and generalization framework [8] – we offer new characterizations of the generalization ability of deep neural networks, when used to learn functions applicable to regression tasks. In particular:

- We derive new generalization error guarantees applicable to such tasks under the ℓ_p-loss.
- Then, building upon our proposed generalization error guarantees, we design new a regularizer allowing to learn deep neural network based data representations applicable to regression settings. Such regularizers explicitly constrain the operator norm of the network input-output Jacobian matrix.
- We also present computationally efficient methods to estimate the spectral norm of the aforementioned Jacobian matrix in order to accelerate the neural network learning process and test their performance on classical denoising tasks.
- Finally, we demonstrate empirically that our bounds – which, in contrast with existing ones, alleviate the exponential dependence of generalization error on network depth – lead to a regularization strategy offering superior generalization results in comparison with existing regularization strategies enforcing Lipschitz continuity.

Of particular relevance, Jacobian regularization has already been shown to result in an improved generalization performance in deep learning classifiers [9], [10]. However, the fact that this phenomenon extends to regression tasks as well has not been explored before in literature to the best of our knowledge.

Notation: Lower case boldface characters denote vectors, upper case boldface characters represent matrices and sets are represented using calligraphic fonts. $\|.\|_{p,q}$ represents the operator norm induced by ℓ_p and ℓ_q vector norms of the matrix argument. In turn, the operator $\|.\|_p$ denotes the ℓ_p norm of a vector argument. $\mathcal{N}_{\mathcal{D}}\left(\psi/2, \rho\right)$ represents the covering number of a metric space $(\mathcal{D}, \rho)$ using balls of radius $\psi/2$.

II. Problem Setup

We are interested in problems involving the estimation of a functional relationship between data points $\mathbf{x} \in \mathcal{X}$ and target points $\mathbf{y} \in \mathcal{Y}$ based on a set of examples $\mathcal{S} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i \leq m}$ drawn i.i.d. from the space $\mathcal{D} = \mathcal{X} \times \mathcal{Y}$ according to an unknown probability measure μ. In this setting, we restrict our attention to learning problems associated with regression tasks where the input data points $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^{N_x}$ and the output data points $\mathbf{y} \in \mathcal{Y} \subseteq \mathbb{R}^{N_y}$. We assume that $\mathcal{X}$ and $\mathcal{Y}$ are compact and we use ℓ_q and ℓ_p to measure distances in $\mathcal{X}$ and $\mathcal{Y}$ respectively. We also assume that the space $\mathcal{D} = \mathcal{X} \times \mathcal{Y}$ is compact with respect to the product metric ρ [11].

In this supervised learning setting, we restrict our attention to the use of deep neural networks $f_{\mathcal{S}} : \mathbb{R}^{N_x} \to \mathbb{R}^{N_y}$ that are trained on the training set $\mathcal{S}$ to learn the underlying map between $\mathcal{X}$ and $\mathcal{Y}$. Such a feed forward deep neural network can be represented as a composition of d layer-wise mappings delivering an output $f_{\mathcal{S}}(\mathbf{x}) \in \mathbb{R}^{N_y}$, given the input $\mathbf{x} \in \mathbb{R}^{N_x}$ as follows:

$$f_{\mathcal{S}}(\mathbf{x}) = \left(f_{\theta_d} \circ \dots f_{\theta_1}\right)(\mathbf{x}; \Theta)$$

Our goal is to characterize the quality of such learnt deep neural network using a well-known measure quantifying the generalization capability of a machine learning models. In particular, we will use the generalization error (GE) associated with the learnt deep neural network given by:

$$GE(f_{\mathcal{S}}) = |l_{\text{exp}}(f_{\mathcal{S}}) - l_{\text{emp}}(f_{\mathcal{S}})| \tag{1}$$

corresponding to the difference between the expected and empirical losses given by:

$$l_{\text{exp}}(f_{\mathcal{S}}) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mu}[l(f_{\mathcal{S}}, (\mathbf{x}, \mathbf{y}))],$$
$$l_{\text{emp}}(f_{\mathcal{S}}) = \frac{1}{m} \sum_i l(f_{\mathcal{S}}, (\mathbf{x}_i, \mathbf{y}_i))$$

where $l(\cdot, \cdot)$ is the pre-specified loss function. For the purpose of our analysis, we will characterize the performance of a deep neural network regressor $f_{\mathcal{S}}(\cdot)$ with respect to the standard ℓ_p-loss function, which – for an input $\mathbf{x} \in \mathcal{X}$ and ground-truth $\mathbf{y} \in \mathcal{Y}$ – is given by:

$$l(f_{\mathcal{S}}(\cdot), (\mathbf{x}, \mathbf{y})) = \|\mathbf{y} - f_{\mathcal{S}}(\mathbf{x})\|_p \tag{2}$$

Our ensuing analysis offers bounds to the generalization error in (1) of deep feed-forward neural networks based regressors as a function of a number of relevant quantities. These quantities include the covering number of the sample space $\mathcal{D}$, the size of the training set $\mathcal{S}$, and properties of the network encapsulated in its input-output Jacobian matrix given by:

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial f_{\mathcal{S}}(\mathbf{x})_1}{\partial \mathbf{x}_1} & \cdots & \frac{\partial f_{\mathcal{S}}(\mathbf{x})_1}{\partial \mathbf{x}_{N_x}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{\mathcal{S}}(\mathbf{x})_{N_y}}{\partial \mathbf{x}_1} & \cdots & \frac{\partial f_{\mathcal{S}}(\mathbf{x})_{N_y}}{\partial \mathbf{x}_{N_x}} \end{bmatrix}$$

III. Main Results

We now develop bounds to the generalization error associated with deep neural networks by leveraging the *algorithmic robustness* framework in [8]. The following definition introduces the notion of a robust learning algorithm.

Definition 1 (*Algorithmic Robustness*)**.** A learning algorithm is said to be $(K, \epsilon(\mathcal{S}))$-robust if the space $\mathcal{D}$ can be partitioned into K disjoint sets $\mathcal{K}_k$, $k = 1, \dots, K$, such that for all $(\mathbf{x}_i, \mathbf{y}_i) \in \mathcal{S}$ and all $(\mathbf{x}, \mathbf{y}) \in \mathcal{D}$

$$\begin{aligned}(\mathbf{x}_i, \mathbf{y}_i), (\mathbf{x}, \mathbf{y}) \in \mathcal{K}_k \implies \\ |l(f_{\mathcal{S}}, (\mathbf{x}_i, \mathbf{y}_i)) - l(f_{\mathcal{S}}, (\mathbf{x}, \mathbf{y}))| \leq \epsilon(\mathcal{S})\end{aligned} \tag{3}$$

The algorithmic robustness framework has already been used to derive generalization bounds for deep learning classifiers [9], [12]–[14]. However, these works consider uniformly bounded loss functions in their analyses and therefore the obtained bounds do not carry over immediately to regression problems due to additional technical difficulties arising from the use of unbounded loss functions. We therefore re-purpose existing analyses in order to understand how deep neural networks can underlie the construction of input-output functional relations for regression tasks under ℓ_p loss functions.

Our analysis – which generalizes a key result in [9] – builds upon a simple characterization of the Lipschitz continuity of a deep neural network, based on the use of ℓ_p and ℓ_q norms to measure distances on the network input and output respectively as follows:

$$\|f_{\mathcal{S}}(\mathbf{x}') - f_{\mathcal{S}}(\mathbf{x}'')\|_p \leq \sup_{\mathbf{x} \in \text{conv}(\mathcal{X})} \|\mathbf{J}(\mathbf{x})\|_{p,q} \|\mathbf{x}' - \mathbf{x}'\|_q \tag{4}$$

where $conv(.)$ is the convex hull of a set.

We can now describe the robustness of a deep neural network regressor trained under an ℓ_p-loss.

Theorem 1. *(Robustness under ℓ_p loss) A DNN $f_{\mathcal{S}}(\cdot)$ trained on a training set $\mathcal{S}$ under the ℓ_p-loss in* (2) *is*

$$\left(\mathcal{N}_{\mathcal{D}}\left(\psi/2, \rho\right), \left(1 + \sup_{\mathbf{x} \in conv(\mathcal{X})} \|\mathbf{J}(\mathbf{x})\|_{p,q}\right)\psi\right) - robust$$

for any $\psi > 0$ and $\mathcal{N}_{\mathcal{D}}\left(\psi/2, \rho\right) < \infty$.

Proof. See Appendix. □

Finally, leveraging Theorem 3, we can also describe a GE bound for a deep neural network regressor trained under a ℓ_p-loss.

Theorem 2. *A DNN $f_{\mathcal{S}}(\cdot)$ trained on the training set $\mathcal{S}$ under the ℓ_p-loss in* (2) *obeys with probability $1 - \zeta$, for any $\zeta > 0$, the GE bound given by:*

$$\begin{aligned} GE(f_{\mathcal{S}}) \quad \leq \quad & \left(1 + \sup_{\mathbf{x} \in conv(\mathcal{X})} \|\mathbf{J}(\mathbf{x})\|_{p,q}\right)\psi \\ & + M\sqrt{\frac{2\mathcal{N}_{\mathcal{D}}\left(\psi/2, \rho\right)\log(2) + 2\log\left(1/\zeta\right)}{m}} \end{aligned}$$

for any $\psi > 0$ and $M < \infty$.

Proof. See Appendix □

A. Discussion

Some comments about the nature of this bound are in order. The bound in Theorem 2 consists of two terms:

- The second term captures the interplay between the cardinality of the training set $\mathcal{S}$ and the complexity

Algorithm 1: Esitmation of the spectral norm of $\mathbf{J}$

```
Input: Mini-batch $\mathcal{B}$, number of power iterations $n$.
Output: Maximum singular value, $\sigma$, of the matrix $\mathbf{J}$.
for $(\mathbf{y}, \mathbf{x}) \in \mathcal{B}$ do
    Initialize $\{\mathbf{u}\} \in \mathbb{R}^{N_x}$
    $i \leftarrow 0$
    while $i < n$ do
        $\mathbf{v} \leftarrow vjp(f(\mathbf{y}), \mathbf{y}, \mathbf{u})$
        $\mathbf{u} \leftarrow jvp(f(\mathbf{y}), \mathbf{y}, \mathbf{v})$
        $i \leftarrow i + 1$.
    $\sigma \leftarrow \|\mathbf{u}\|_2 / \|\mathbf{v}\|_2$
```

of the data space measured via its covering number. Intuitively, the generalization error decreases with the increase in the cardinality of the training set, and it also decreases with the decrease in the covering number of the data space. It is generally recognized that real-world data is associated with spaces exhibiting small intrinsic dimension – hence bounded covering numbers [9] – where the optimal covering ball radius can be evaluated using network characteristics [9], [14].

- The first term is associated with the Lipschitz constant of the loss function that, in turn, is proportional to the Lipschitz constant of the deep neural network for the loss function. This – in sharp contrast with existing parameter dependent bounds [15] – then suggests that the generalization capability of a deep neural network does not deteriorate exponentially with the network depth, in view of the fact that the Lipschitz constant of a deep neural network depends on a network input-output Jacobian operator norm. In fact, in agreement with empirical results reported in [10] for DNN classifiers, we also show experimentally that the generalization error for DNN regressors tends to be directly proportional to the operator norm of the network input-output Jacobian matrix.

Overall, in view of a direct dependence between the generalization ability and the operator norm of the network input-output Jacobian matrix, our bounds also suggest an entirely new regularization strategy that can outperform existing neural network regularization techniques such as weight decay, weight orthogonalization [12] and penalizing the frobenius norm of the jacobian [9], [16] as shown in the sequel.

IV. Jacobian Based Regularization

Building upon the insights associated with our bound, we now propose a new regularization strategy applicable to scenarios where one adopts an ℓ_2-metric both on the network input and output.

In particular, by adopting such an Euclidean metric, the bounds suggest that the generalization ability of a deep neural network depends on the maximum value of the spectral norm of the network input-output Jacobian matrix over the convex hull of the input space. We therefore propose a new regularization technique that penalizes the sum of the spectral norm of the network Jacobian matrix evaluated at the various

Algorithm 2: Computation of the jvp.

```
Input: Mini-batch $\mathcal{B}$, model outputs $f(\mathbf{y})$, vector $\mathbf{v}$.
Output: $\mathbf{J}\mathbf{v}$
Initialize a dummy tensor $\mathbf{d}$.
$\mathbf{g} \leftarrow vjp(f(\mathbf{y}), \mathbf{y}, \mathbf{d})$
$\mathbf{u} \leftarrow vjp(\mathbf{g}, \mathbf{p}, \mathbf{v})$
return $\mathbf{u}$
```

training samples within the training set, leading immediately to the training objective given by

$$\frac{1}{m}\sum_{i=1}^{m} l(f_{\mathcal{S}}, \mathbf{s}_i) + \beta \sum_{i=1}^{m} \sigma_{\max}(\mathbf{J}(\mathbf{x}_i))^2 \tag{5}$$

where $\sigma_{\max}(\cdot)$ represents the spectral norm of its matrix argument. The hyper-parameter β balances between the desire to minimize the empirical error and the desire to minimize the spectral norm of the network Jacobian[1].

Note that this new training objective encourages the network to explore solutions in regions in the parameter space associated with Jacobians with small spectral norms which – owing to the nonlinear nature of a deep neural network – allows for a broader parameter search space. This is contrast to more restrictive techniques such as (1) weight decay that constrains the network weights to exhibit small norms (hence, these techniques do not take into account the correlation between the rows of weight matrices) or else (2) weight orthogonalization that constrains the weights to lie on Stiefel manifold [12], [14].

A. Efficient Computation of the Jacobian Based Regularizer

The challenge associated with the use of the training objectives in (5) relates to the computation of the Spectral norm of the Jacobian matrix $\mathbf{J}$ because computing and storing the Jacobian matrix of deep neural networks incurs huge cost. A computationally efficient algorithm to approximate the spectral norm of the Jacobian has already been proposed and is given in Algorithm 1 [17]. We however, provide them here for the sake of completeness.

The procedure leverages the power method [18] to approximate the spectral norm of the Jacobian based regularization term in (5). It starts by choosing (randomly) an initial (nonzero) approximation of the left singular vector $\mathbf{u}$ in $\mathbb{R}^{N_x}$ associated with the highest singular value of the matrix $\mathbf{J}$. It then leverages the automatic differentiation to iteratively compute the Jacobian vector product and vector Jacobian product as follows:

$$\mathbf{v} \leftarrow \left[\frac{df(\mathbf{y})}{d\mathbf{y}}\right]^T \mathbf{u}, \qquad \mathbf{u} \leftarrow \left[\frac{df(\mathbf{y})}{d\mathbf{y}}\right] \mathbf{v}$$

We fix the number of iterations to $n = 3$ since these result in sufficiently accurate numerical approximations of the spectral norm. The spectral norm σ is then equal to $\|\mathbf{u}\|_2 / \|\mathbf{v}\|_2$.

[1]Note that the Rectified Linear Unit (ReLU) defined as $\phi(x) = \max(0, x)$ is the non-linearity of choice in most DNNs. It has an undefined derivative for $x = 0$. However it should be noted that the nondifferentiable points lie in a set of measure zero and gradient based optimization algorithms never reach such points in reality and therefore the Jacobian matrix and the its norms can be computed almost every realizable point [9].

TABLE I
TEST PSNR (DB) AND SSIM FOR A 3-LAYER DNCNN, TRAINED FOR 140 EPOCHS ON BSD300.

	40 samples		200 samples		400 samples	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Wdecay	25.56	0.52	27.11	0.61	28.17	0.68
Wortho	25.42	0.51	26.91	0.61	27.72	0.67
Jfro	26.78	0.53	26.92	0.63	27.55	0.68
Jspectral	27.22	0.58	27.26	0.64	27.94	0.70

The algorithm exploits the reverse and forward mode automatic differentiation to compute the vector Jacobian product $vjp(\cdot,\cdot,\cdot)$, and the Jacobian vector products $jvp(\cdot,\cdot,\cdot)$ respectively. All major deep learning frameworks offer support for the computation of reverse mode vector Jacobian product. The forward mode Jacobian vector product can easily be computed via the reverse mode automatic differentiation using the method described in Algorithm 2 [19].

Note again that the merit of Algorithm 1 lies in computing the spectral norm of Jacobian without explicitly computing the Jacobian itself that is prohibitive in high-dimensional settings.

V. EXPERIMENTS

We now conduct experiments to gauge the effectiveness of the proposed regularizer on the classical image denoising problem involving the reconstruction of a clean image given a noisy version of the image (corrupted with additive Gaussian noise with mean 0 and variance 0.01). We use the 3-layer version of the classical DnCNN model from [20] with 32 filters of size 3×3 followed by ReLU in first layer, 32 filters of dimension 3×3 followed by batch normalization and ReLU in the second layer and 1 filter of size 3×3 in the third layer. We use 64×64 cutouts of the greyscale images taken from BSD300 dataset [21] for training, validation and testing. Three different training sets of size 40, 200 and 400 samples are considered. For each of these training sets we fix the validation and test sets of size 1750 samples each. An SGD based opimizer is used for trainig and the hyperparamter β is tuned on the validation set using grid search.

We compare the performance of our proposed gradient based regularizer (5) (referred to as Jspectral) with Frobenius norm of the Jacobian (Jfro) [9], [16] and other conventional weight based regularizers such as weight decay (Wdecay) and weight orthogonalization (Wortho) [12]. The performance of the network trained with these regularization methods is reported in Table I. Our results show that Jspectral outbeat the competing regularizers both in terms of Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM). Note that, although, for large training samples the PSNR of the reconstructed images achieved using Jspectral and Wdecay regularization eventually becomes comparable if the network is trained for sufficiently large number of epochs. However, our results (not included in the paper due to the shortage of space), show that Jacobian based regularizers converge much faster than Weight based regularization techniques[2]. A visual comparison of the reconstruction quality with various regularizers can be seen in Figure 1.

To conclude, in line with our analysis, we offer one additional result showcasing that generalization indeed appears to be intimately related to the spectral norm of the network Jacobian. In particular, Figure 2 compares the value of the network input-output Jacobian spectral norm for a deep neural network trained under standard SGD with no regularization (referred to as Vanilla) and a deep neural network trained under SGD for various regularization strategies. It can be seen that in all the cases the spectral norm of the Jacobians decreases gradually with the increase in the number of training epochs. Therefore, Jspectral regularization has the potential to improve generalization further. These trends apply not only to this denoising tasks but also other regression tasks such as compressed sensing [22].

VI. CONCLUSIONS

We have studied the generalization behaviour of deep neural networks by building upon the robustness framework. In particular, we have offered new generalization bounds applicable to regression problems – that encapsulate key quantities associated with the learning problem, including the complexity of the data space, the cardinality of the training set, and the network Lipschitz constant. Notably, our bounds have also led to an entirely new regularization strategy – based on the penalization of the spectral norm of the network Jacobian – that clearly outperform existing regularizers.

APPENDIX

Proof of Theorem 1. We can establish a $\psi/2$ cover of $\mathcal{D}$ such that $K \le \mathcal{N}_{\mathcal{D}}(\psi/2, \rho)$ such that $\forall(\mathbf{x}', \mathbf{y}') \in \mathcal{S}$ and $(\mathbf{x}'', \mathbf{y}'') \in \mathcal{D}$, if $(\mathbf{x}', \mathbf{y}')$ and $(\mathbf{x}'', \mathbf{y}'')$ correspond to the same partition, then $\rho((\mathbf{x}', \mathbf{y}'), (\mathbf{x}'', \mathbf{y}'')) \le \psi$.

Let us now consider two data points $(\mathbf{x}', \mathbf{y}') \in \mathcal{S}$ and $(\mathbf{x}'', \mathbf{y}'') \in \mathcal{D}$ associated with one of the partitions. Then

$$\begin{aligned} & |l(f_{\mathcal{S}}, (\mathbf{x}', \mathbf{y}')) - l(f_{\mathcal{S}}, (\mathbf{x}'', \mathbf{y}''))| \\ = & \left| \|\mathbf{y}' - f_{\mathcal{S}}(\mathbf{x}')\|_p - \|\mathbf{y}'' - f_{\mathcal{S}}(\mathbf{x}'')\|_p \right| \\ \overset{(a)}{\le} & \|\mathbf{y}' - f_{\mathcal{S}}(\mathbf{x}') - \mathbf{y}'' + f_{\mathcal{S}}(\mathbf{x}'')\|_p \\ \overset{(b)}{\le} & \|\mathbf{y}' - \mathbf{y}'\|_p + \|f_{\mathcal{S}}(\mathbf{x}') - f_{\mathcal{S}}(\mathbf{x}'')\|_p \\ \overset{(c)}{\le} & \|\mathbf{y}' - \mathbf{y}''\|_p + \max_{\text{conv}(\mathcal{X})} \|\mathbf{J}(\mathbf{x})\|_{p,q} \|\mathbf{x}' - \mathbf{x}''\|_q \\ \overset{(d)}{\le} & \left(1 + \sup_{\mathbf{x} \in \text{conv}(\mathcal{X})} \|\mathbf{J}(\mathbf{x})\|_{p,q}\right) \rho((\mathbf{x}', \mathbf{y}'), (\mathbf{x}'', \mathbf{y}'')) \quad (6) \end{aligned}$$

The inequalities $(a), (b)$ and (c) hold due to reverse triangle inequality, Minkowski-inequality and eq. (4), respectively, where (d) holds trivially because sup metric ρ upper bounds

[2]The networks regularized using Jspectral converge in relatively fewer number of epochs (≈ 50) as compared to the Wdecay (≈ 100).

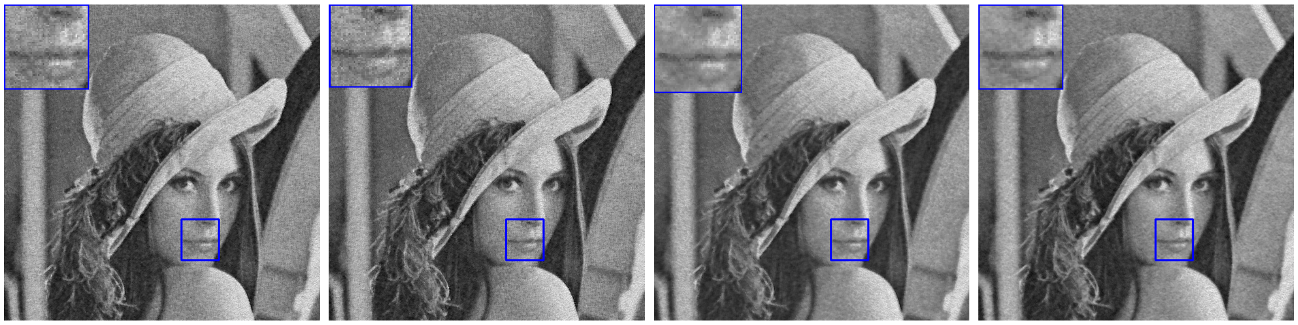
Fig. 1. Sample results of the denoised images for $m = 40$ using a 5-layer DnCNN [20]. (Left to Right) Wdecay (PSNR = 28.57), Wortho (PSNR = 28.02), Jfro (PSNR = 29.89), Jspectral (PSNR = 30.36).

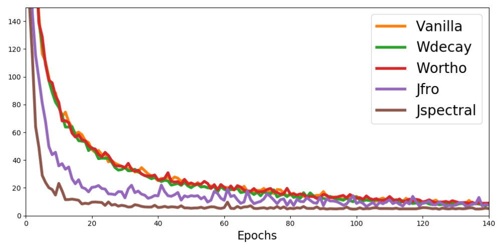

Fig. 2. Sum of the spectral norms of the network input-output Jacobian evaluated on training samples $\sum_{i=1}^{m} \sigma_{\max}(\mathbf{J}(\mathbf{x}_i))^2$ versus number of training epochs. The neural network is trained using SGD under different regularizers, for the image denoising task.

the distance metric on data space. It now follows immediately from (3) that

$$|l(f_{\mathcal{S}}, (\mathbf{x}', \mathbf{y}')) - l(f_{\mathcal{S}}, (\mathbf{x}'', \mathbf{y}''))| \leq \left(1 + \sup_{\mathbf{x} \in \text{conv}(\mathcal{X})} \|\mathbf{J}(\mathbf{x})\|_{p,q}\right)\psi$$

and the theorem follows. □

Proof of Theorem 2. The GE of a robust deep neural network based classifier follows immediately from the robustness result. In particular, it has been shown in [8], that with probability greater than $1 - \zeta$

$$GE \leq \epsilon(\mathcal{S}) + M\sqrt{\frac{2K\log(2) + 2\log(1/\zeta)}{m}} \quad (7)$$

where M represents the maximum value of loss over all the samples in the sample space $\mathcal{D}$ that can be shown to be finite for a Lipschitz continuous deep neural network [22]. Now, Theorem 1 shows that K can be upper bounded by $\mathcal{N}_{\mathcal{D}}(\psi/2, \rho)$, it also shows that $\epsilon(\mathcal{S}) \leq \left(1 + \sup_{\mathbf{x} \in \text{conv}(\mathcal{X})} \|\mathbf{J}(\mathbf{x})\|_{p,q}\right)\psi$, leading immediately to the result. □

References

[1] S. Dargan, M. Kumar, M. R. Ayyagari, and G. Kumar, "A survey of deep learning and its applications: A new paradigm to machine learning," *Archives of Computational Methods in Engineering*, pp. 1–22, 2019.

[2] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," *arXiv preprint arXiv:1611.03530*, 2016.

[3] P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, and I. Sutskever, "Deep double descent: Where bigger models and more data hurt," *arXiv preprint arXiv:1912.02292*, 2019.

[4] S. S. Du, X. Zhai, B. Poczos, and A. Singh, "Gradient descent provably optimizes over-parameterized neural networks," *arXiv preprint arXiv:1810.02054*, 2018.

[5] Y. Cao and Q. Gu, "Generalization error bounds of gradient descent for learning over-parameterized deep relu networks." in *AAAI*, 2020, pp. 3349–3356.

[6] B. Neyshabur, Z. Li, S. Bhojanapalli, Y. LeCun, and N. Srebro, "Towards understanding the role of over-parametrization in generalization of neural networks," *arXiv preprint arXiv:1805.12076*, 2018.

[7] G. K. Dziugaite and D. M. Roy, "Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data," 2017.

[8] H. Xu and S. Mannor, "Robustness and generalization," *Machine learning*, vol. 86, no. 3, pp. 391–423, 2012.

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

[10] R. Novak, Y. Bahri, D. A. Abolafia, J. Pennington, and J. Sohl-Dickstein, "Sensitivity and generalization in neural networks: an empirical study," *arXiv preprint arXiv:1802.08760*, 2018.

[11] N. Weaver, *Lipschitz algebras*. World Scientific, 1999.

[12] M. Cisse, P. Bojanowski, E. Grave, Y. Dauphin, and N. Usunier, "Parseval networks: Improving robustness to adversarial examples," *arXiv preprint arXiv:1704.08847*, 2017.

[13] J. Amjad, J. Sokolić, and M. R. Rodrigues, "On deep learning for inverse problems," in *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, 2018, pp. 1895–1899.

[14] K. Jia, S. Li, Y. Wen, T. Liu, and D. Tao, "Orthogonal deep neural networks," *arXiv preprint arXiv:1905.05929*, 2019.

[15] D. Jakubovitz, R. Giryes, and M. R. Rodrigues, "Generalization error in deep learning," *arXiv preprint arXiv:1808.01174*, 2018.

[16] J. Hoffman, D. A. Roberts, and S. Yaida, "Robust learning with jacobian regularization," *arXiv preprint arXiv:1908.02729*, 2019.

[17] C. Anil, J. Lucas, and R. Grosse, "Sorting out lipschitz function approximation," in *International Conference on Machine Learning*, 2019, pp. 291–301.

[18] R. Mises and H. Pollaczek-Geiringer, "Praktische verfahren der gleichungsauflösung." *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 9, no. 2, pp. 152–164, 1929.

[19] J. Townsend, "A new trick for calculating Jacobian vector products," https://j-towns.github.io/2017/06/12/A-new-trick.html, 2017, accessed: 2020-01-17.

[20] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[21] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th Int'l Conf. Computer Vision*, vol. 2, July 2001, pp. 416–423.

[22] J. Amjad, Z. Lyu, and M. R. Rodrigues, "Model-aware regularization for learning approaches to inverse problems," *arXiv preprint arXiv:2006.10869*, 2020.