Norm-based generalisation bounds for deep multi-class convolutional neural networks

Ledent, Antoine; Mustafa, Waleed; Lei, Yunwen; Kloft, Marius

Norm-based generalisation bounds for deep multi-class convolutional neural networks

Antoine Ledent [1], Waleed Mustafa [1], Yunwen Lei [2] and Marius Kloft [1]

[1]Department of Computer Science, TU Kaiserslautern, 67653 Kaiserslautern, Germany
[2]School of Computer Science, University of Birmingham, Birmingham B15 2TT, United Kingdom

Abstract

We show generalisation error bounds for deep learning with two main improvements over the state of the art. (1) Our bounds have no explicit dependence on the number of classes except for logarithmic factors. This holds even when formulating the bounds in terms of the L^2-norm of the weight matrices, where previous bounds exhibit at least a square-root dependence on the number of classes. (2) We adapt the classic Rademacher analysis of DNNs to incorporate weight sharing—a task of fundamental theoretical importance which was previously attempted only under very restrictive assumptions. In our results, each convolutional filter contributes only once to the bound, regardless of how many times it is applied. Further improvements exploiting pooling and sparse connections are provided. The presented bounds scale as the norms of the parameter matrices, rather than the number of parameters. In particular, contrary to bounds based on parameter counting, they are asymptotically tight (up to log factors) when the weights approach initialisation, making them suitable as a basic ingredient in bounds sensitive to the optimisation procedure. We also show how to adapt the recent technique of loss function augmentation to our situation to replace spectral norms by empirical analogues whilst maintaining the advantages of our approach.

Introduction

Deep learning has enjoyed an enormous amount of success in a variety of engineering applications in the last decade (Krizhevsky, Sutskever, and Hinton 2012; He et al. 2016; Karras, Laine, and Aila 2018; Silver et al. 2018). However, providing a satisfying explanation to its sometimes surprising generalisation capabilities remains an elusive goal (Zhang et al. 2017; Du et al. 2019; Asadi, Abbe, and Verdu 2018; Goodfellow, Shlens, and Szegedy 2015). The statistical learning theory of deep learning approaches this question by providing a theoretical analysis of the generalisation performance of deep neural networks (DNNs) through better understanding of the complexity of the function class corresponding to a given architecture or training procedure.

This field of research has enjoyed a revival since 2017 with the advent of learning guarantees for DNNs expressed in terms of various norms of the weight matrices and classification margins (Neyshabur, Bhojanapalli, and Srebro 2018; Bartlett, Foster, and Telgarsky 2017; Zhang, Lei, and Dhillon 2018; Li et al. 2019; Allen-Zhu, Li, and Liang 2019). Many improvements have surfaced to make bounds non-vacuous at realistic scales, including better depth dependence, bounds that apply to ResNets (He, Liu, and Tao 2019), and PAC-Bayesian bounds using network compression (Zhou et al. 2019), data-dependent Bayesian priors (Dziugaite and Roy 2018), fast rates (Suzuki 2018), and reduced dependence on the product of spectral norms via data-dependent localisation (Wei and Ma 2019; Nagarajan and Kolter 2019). A particularly interesting new branch of research combines norm-based generalisation bounds with the study of how the optimisation procedure (stochastic gradient descent) implicitly restricts the function class (Cao and Gu 2019; Du et al. 2019; Arora et al. 2019; Zou et al. 2018; Jacot, Gabriel, and Hongler 2018; Frankle and Carbin 2019). One idea at the core of many of these works is that the weights stay relatively close to initialisation throughout training, reinforcing lucky guesses from the initialised network rather than constructing a solution from scratch. Thus, in this branch of research, it is critical that the bound is negligible when the network approaches initialisation, i.e., *the number of weights involved is not as important as their size*. This observation was first made as early as in (Bartlett 1998).

Despite progress in so many new directions, we note that some basic questions of fundamental theoretical importance have remain unsolved. (1) How can we remove or decrease the dependence of bounds on the number of classes? (2) How can we account for weight sharing in convolutional neural networks (CNNs)? In the present paper, we contribute to an understanding of both questions.

Question (1) is of central importance in extreme classification (Prabhu and Varma 2014), where we deal with an extremely high number of classes (e.g. millions). (Bartlett, Foster, and Telgarsky 2017) showed a bound with no explicit class dependence (except for log terms). However, this bound is formulated in terms of the $L^{2,1}$ norms of the network's weight matrices. If we convert the occurring $L^{2,1}$ norms into the more commonly used L^2 norms, we obtain a square-root dependence on the number of classes.

Regarding (2), (Li et al. 2019) showed a bound that accounts for weight sharing. However, this bound is valid only under the assumption of orthonormality of the weight matrices. The assumption of unit norm weights—which is violated by typical convolutional architectures (GoogLeNet, VGG, Inception, etc.)—makes it difficult to leverage the generalisation gains from small weights, and it is a fortiori not easy to see how the bounds could be expressed in terms of distance to initialisation.

In this paper, we provide, up to only logarithmic terms, a complete solution to both of the above questions. First, our bound relies only on L^2 norms at the last layer, yet it has no explicit (non-logarithmic) dependence on the number of classes. In deep learning, no generalization bound other than ours has ever achieved a lack of non-logarithmic class dependency with L^2 norms. Second, our bound accounts for weight sharing in the following way. The Frobenius norm of the weight matrix of each convolutional filter contributes only *once* to the bound, regardless of how many times it is applied. Furthermore, our results have several more properties of interest: (i) We exploit the L^∞-continuity of nonlinearities such as pooling and ReLu to further significantly reduce the explicit width dependence in the above bounds. (ii) We show how to adapt the recent technique of loss function augmentation to our setting to replace the dependence on the spectral norms by an *empirical* Lipschitz constant with respect to well chosen norms. (iii) Our bounds also have very little explicit dependence on architectural choices and rely instead on norms of the weight matrices expressed as distance to initialisation, affording a high degree of architecture robustness compared to parameter-space bounds. In particular, our bounds are negligible as the weights approach initialisation.

In parallel to our efforts, (Long and Sedghi 2020) recently made progress on question (2), providing a remedy to the weight-sharing problem. Their work, which appeared at ICLR 2020, is independent of ours. This can be observed from the fact that their work and ours were first preprinted on arXiv on the very same day. Their approach is completely different from ours, and both approaches have their merits and disadvantages. We provide an extensive discussion and comparison in Section H.

Related Work

We now discuss the related work on the statistical learning theory (SLT) of DNNs. The SLT of neural networks can be dated back to 1970s, based on the concepts of VC dimension, fat-shattering dimension (Anthony and Bartlett 2002), and Rademacher complexities (Bartlett and Mendelson 2002). Here, we focus on recent work in the era of deep learning.

Let $(x_1, y_1), \dots, (x_n, y_n)$ be training examples independently drawn from a probability measure defined on the sample space $\mathcal{Z} = \mathcal{X} \times \{1, \dots, K\}$, where $\mathcal{X} \subset \mathbb{R}^d$, d is the input dimension, and K is the number of classes. We consider DNNs parameterized by weight matrices $\mathcal{A} = \{A^1, \dots, A^L\}$, so that the prediction function can be written $F_{\mathcal{A}}(x) = A^L \sigma_{L-1}\big(A^{L-1}\sigma_{L-2}\big(\cdots A^1 x\big)\big)$, where L is the depth of the DNN, $A^l \in \mathbb{R}^{W_{l-1} \times W_l}$, $W_0 = d, W_L = K$, and $\sigma_i : \mathbb{R}^{W_i} \mapsto \mathbb{R}^{W_i}$ is the non linearity (including any pooling and activation functions).

When providing PAC guarantees for DNNs, a critical quantity is the Rademacher complexity of the network obtained after appending any loss function. The first work in this area (Neyshabur, Tomioka, and Srebro 2015) therefore focused on bounding the Rademacher complexity of networks satisfying certain norm conditions, where the last layer is one-dimensional. They apply the concentration lemma and a peeling technique to get a bound on the Rademacher complexity of the order $O\big(\frac{2^L}{\sqrt{n}} \prod_{i=1}^L \|A^i\|_{\mathrm{Fr}}\big)$, where $\|A\|_{\mathrm{Fr}}$ denotes the Frobenius norm of a matrix A. (Golowich, Rakhlin, and Shamir 2018) showed that this exponential dependency on the depth can be avoided by an elegant use of the contraction lemma to obtain bounds of the order $O\big((\sqrt{L}/\sqrt{n}) \prod_{i=1}^L \|A^i\|_{\mathrm{Fr}}\big)$.[1] The most related work to ours is the spectrally-normalized margin bound by (Bartlett, Foster, and Telgarsky 2017) for multi-class classification. Writing $\|A\|_\sigma$ for the spectral norm, the result is of order $\tilde{O}(M/\gamma)$ with

$$M = \frac{1}{\sqrt{n}} \prod_{i=1}^L \|A^i\|_\sigma \left(\sum_{i=1}^L \frac{\|A^{i\top}\|_{2,1}^{\frac{2}{3}}}{\|A^i\|_\sigma^{\frac{2}{3}}} \right)^{\frac{3}{2}}, \qquad (1)$$

where $\|A\|_{p,q} = \Big(\sum_j \big(\sum_i |A_{ij}|^p \big)^{\frac{q}{p}} \Big)^{\frac{1}{q}}$ is the (p, q)-norm, and γ denotes the classification margin.

At the same time as the above result appeared, the authors in (Neyshabur, Bhojanapalli, and Srebro 2018) used a PAC Bayesian approach to prove an analogous result [2], where $W = \max\{W_0, W_1, \dots, W_L\}$ is the width:

$$\tilde{O}\left(\frac{L\sqrt{W}}{\gamma\sqrt{n}} \left(\prod_{i=1}^L \|A^i\|_\sigma \right) \left(\sum_{i=1}^L \frac{\|A^i\|_{\mathrm{Fr}}^2}{\|A^i\|_\sigma^2} \right)^{\frac{1}{2}} \right). \qquad (2)$$

These results provide solid theoretical guarantees for DNNs. However, they take very little architectural information into account. In particular, if the above bounds are applied to a CNN, when calculating the squared Frobenius norms $\|A^i\|_{\mathrm{Fr}}^2$, the matrix A^i is the matrix representing the linear operation performed by the convolution, which implies that the weights of each filter will be summed as many times as it is applied. This effectively adds a dependence on the square root of the size of the corresponding activation map at each term of the sum. Furthermore, the L^2 version of the above bound (1) includes a dependence on the square root of the number of classes through the maximum width W of the network. This square-root dependence is not favorable when the number of classes is very large. Although many efforts have been performed to improve the class-size dependency in the shallow learning literature (Lauer 2018; Guermeur 2002, 2007; Koltchinskii and Panchenko 2002; Guermeur 2017; Musayeva, Lauer, and Guermeur 2019; Mohri, Rostamizadeh, and Talwalkar 2018; Lei et al. 2019), extensions of those results to deep learning are missing so far.

In late 2017 and 2018, there was a spur of research effort on the question of fine-tuning the analyses that provided the above bounds, with improved dependence on depth (Golowich, Rakhlin, and Shamir 2018), and some bounds for

[1] Note that both of these works require the output node to be one dimensional and thus are not multiclass

[2] Note that the result using formula (2) can also be derived by expressing (1) in terms of L^2 norms and using Jensen's inequality

recurrent neural networks (Chen, Li, and Zhao 2019; Zhang, Lei, and Dhillon 2018)). Notably, in (Li et al. 2019), the authors provided an analogue of (1) for convolutional networks, but only under some very specific assumptions, including orthonormal filters. Those conditions are not satisfied by the typical convolutional architectures (GoogLeNet, VGG, Inception, etc.).

Independently of our work, (Long and Sedghi 2020, to appear at ICLR 2020) address the weight-sharing problem using a parameter-space approach. Their bounds scale roughly as the square root of the number of parameters in the model. In contrast to ours, their employed proof technique is more similar to (Li et al. 2019): it focuses on computing the Lipschitz constant of the functions with respect to the parameters. The result by (Long and Sedghi 2020) and ours, which we contrast in detail in Section , both have their merits. In nutshell, the bound by (Long and Sedghi 2020) remarkably comes along without dependence on the product of spectral norms (up to log terms), thus effectively removing the exponential dependence on depth. Our result on the other hand comes along without an explicit dependence on the number of parameters, which can be very large in deep learning. As already noted in (Bartlett 1998), this property is crucial when the weights are small or close to the initialisation.

Lastly, we would like to point out that, over the course of the past year, several techniques have been introduced to replace the dependence on the product of spectral norms by an empirical version of it, at the cost of either assuming smoothness of the activation functions (Wei and Ma 2019) or a factor of the inverse minimum preactivation (Nagarajan and Kolter 2019). Slightly earlier, a similar bound to that in (Long and Sedghi 2020) (with explicit dependence on the number of parameters) had already been proved for an unsupervised data compression task (which does not apply to our supervised setting) in (Lee and Raginsky 2019). Recently, another paper addressing the weight sharing problem appeared on arXiv (Lin and Zhang 2019). In this paper, which was preprinted several months after (Long and Sedghi 2020) and ours, the authors provided another solution to the weight sharing problem, which incorporates elements from both our approach and that of (Long and Sedghi 2020): they bound the L^2-covering numbers at each layer independently, but use parameter counting at each layer, yielding *both* an unwanted dependence on the number of parameters in each layer (from the parameter counting) *and* a dependence on the spectral norms from the chaining of the layers.

Further related work includes the following. (Du et al. 2018) showed size-free bounds for CNNs in terms of the number of parameters for two-layer networks. In (Sedghi, Gupta, and Long 2019), the authors provided an ingenious way of computing the spectral norms of convolutional layers, and showed (interestingly) that regularising the network to make them approach 1 for each layer is both feasible and beneficial to accuracy. Several researchers have also provided interesting insights into DNNs from different perspectives, including through model compression (Neyshabur, Bhojanapalli, and Srebro 2018), capacity control by VC dimensions (Harvey, Liaw, and Mehrabian 2017), and the implicit restriction on the function class imposed by the optimisation procedure (Arora et al. 2018; Zhou et al. 2019; Neyshabur et al. 2019, to appear; Suzuki 2018; Du et al. 2019; Jacot, Gabriel, and Hongler 2018; Arora et al. 2019).

Contributions in a Nutshell

In this section, we state the simpler versions of our main results for specific examples of neural networks. The general results are described in in more technical detail in Section A.

Fully Connected Neural Networks

In the fully connected case, the bound is particularly simple:

Theorem 1 (Multi-class, fully connected)**.** *Assume that we are given some fixed reference matrices $M^1, M^2, \dots, M^L$ representing the initialised values of the weights of the network. Set $\widehat{R}_\gamma(F_\mathcal{A}) = (1/n)(\#(i : F(x_i)_{y_i} < \gamma + \max_{j\neq y_i} F(x_i)_j))$, where # denotes the cardinality of a set. With probability at least $1-\delta$, every network $F_\mathcal{A}$ with weight matrices $\mathcal{A} = (A_1, A_2, \dots, A_L)$ and every margin $\gamma > 0$ satisfy:*

$$\mathbb{P}(\arg\max_j(F_\mathcal{A}(x)_j) \neq y) \leq \widehat{R}_\gamma(F_\mathcal{A}) + \tag{3}$$

$$\widetilde{\mathcal{O}}\left(\frac{\max_{i=1}^n \|x_i\|_{\mathrm{Fr}} R_\mathcal{A}}{\gamma\sqrt{n}} \log(\bar{W}) + \sqrt{\frac{\log(1/\delta)}{n}}\right), \tag{4}$$

where $W = \bar{W} = \max_{i=1}^L W_i$is the maximum width of the network, and

$$R_\mathcal{A} := \rho_L \max_i \|A^L_{i,\bullet}\|_{\mathrm{Fr}} \left(\prod_{i=1}^{L-1} \rho_i \|A^i\|_\sigma\right) \tag{5}$$

$$\left(\sum_{i=1}^{L-1} \frac{(\|A^i - M^i\|_{2,1}^{2/3}}{\|A^i\|_\sigma^{2/3}} + \frac{\|A^L\|_{\mathrm{Fr}}^{2/3}}{\max_i \|A^L_{i,\bullet}\|_{\mathrm{Fr}}^{2/3}}\right)^{\frac{3}{2}}. \tag{6}$$

Note that the last term of the sum does not explicitly contain architectural information, and assuming bounded L^2 norms of the weights, the bound only implicitly depends on W_i for $i \leq L-1$ (through $\|A^i - M^i\|_{2,1} \leq \sqrt{W_{i-1}}\|A^i - M^i\|_{2,1}$), but not on W_L (the number of classes). This means the above is a class-size free generalisation bound (up to a logarithmic factor) with L^2 norms of the last layer weight matrix. This improves on the earlier $L^{2,1}$ norm result in (Bartlett, Foster, and Telgarsky 2017). To see this, let us consider a standard situation where the rows of the matrix A^L have approximately the same L^2 norm, i.e., $\|A^L_{i,\bullet}\|_2 \asymp a$. (In Section I in the Appendix, we show that similar conditions hold except on a subset of weight space of asymptotically vanishing measure and further discuss possible behaviour of the norms.) In this case, our bound involves $\|A^L\|_{\mathrm{Fr}} \asymp \sqrt{W_L}a$, which incurs a square-root dependency on the number of classes. As a comparison, the bound in (Bartlett, Foster, and Telgarsky 2017) involves $\|(A^L)^\top\|_{2,1} \asymp W_L a$, which incurs a linear dependency on the number of classes. If we further impose an L_2-constraint on the last layer as $\|A^L\|_{\mathrm{Fr}} \leq a$ as in the SVM case for a constant a (Lei et al. 2019), then our bound would enjoy a logarithmic dependency while the bound in (Bartlett, Foster, and Telgarsky 2017) enjoys a square-root dependency. This cannot be improved without also changing the dependence on n. Indeed, if it could, we would be able to get good guarantees for classifiers working on fewer examples than classes. Furthermore, in the above bound, the dependence on the spectral norm of A^L in the other terms of the sum is reduced to a dependence on $\max_i \|A^L_{i,\boldsymbol{\cdot}}\|_2$[3]. Both improvements are based on using the L^∞-continuity of margin-based losses.

Convolutional Neural Networks

Our main contribution relates to CNNs. To avoid blinding the reader with notation, we present first simple versions of our results.

Two-layers The topic of the present paper is often notationally cumbersome, which imposes an undue burden on readers. Therefore, we first present a particular case of our bound for a two-layer network composed of a convolutional layer and a fully connected layer with a single input channel, *with explicit pre chosen norm constraints*[4]. Note that the restrictions are purely based on notational and reader convenience: more general results are presented later and in the Appendix.

2-layer Notation: Consider a two layer network with a convolutional layer and a fully connected layer. Write d, C for the dimensions of the input space and the number of classes respectively. We write w for the *spacial* dimension of the hidden layer *after pooling*[5]. Write A^1, A^2 for the sets of weights of the first and second layer, with the weights appearing only once in the convolutional case. A^2 is a matrix whilst A^1 can be arranged as a tensor or unfolded as a matrix. The matrix, which we denote by $\tilde{A}^1$, representing the convolution operation presents the weights of the matrix A_1 repeated as many times as the filters are applied. For any input $x \in \mathbb{R}^d$, we write $|x|_0$ for the maximum L^2 norm of *any single convolutional patch* of x. For instance, if x is an image and the network applies 3×3 convolutions with stride 1, $|x|_0$ would be the maximum L^2 norm of any sub-image of size $3 \times 3 \times m$ where m is the number of channels. The network is represented by the function

$$F(x) = A^2 \sigma(\tilde{A}^1 x),$$

where σ denotes the non linearities (including both pooling and activation functions). As above, M^1, M^2 are the initialised weights.

Theorem 2. *Let $a_1, a_2, a_*, b_0, b_1 > 0$. Suppose that the distribution over inputs is such that $|x|_0 \leq b$ a.s. With probability $> 1-\delta$ over the draw of the training set, for every network $\mathcal{A} = (A^1, A^2)$ with weights satisfying $\|(A^1 - M^1)^\top\|_{2,1} \leq a_1$, $\|A^2 - M^2\|_{\mathrm{Fr}} \leq a_2$ and $\sup_{c \leq C} \|A^2_{c,\boldsymbol{\cdot}}\|_2 \leq a_*$, if $\sup_{i \leq n} \|\tilde{A}^1 x_n\|_2 \leq b_1$, then*

$$\mathbb{P}\left(\arg\max_j (F_{\mathcal{A}}(x)_j) \neq y\right) \tag{7}$$
$$\leq \widehat{R}_\gamma(F_{\mathcal{A}}) + 3\sqrt{\frac{\log(\frac{2}{\delta})}{2n}} + \frac{\mathcal{C}}{\sqrt{n}} \mathcal{R} \left[\log_2(n^2 \mathcal{D})\right]^{\frac{1}{2}} \log(n),$$

where $\mathcal{C}$ is an absolute constant,

$$\mathcal{R}^{2/3} = \left[b_0 a_1 \max\left(\frac{1}{b_1}, \frac{\sqrt{w} a_*}{\gamma}\right)\right]^{2/3} + \left[\frac{b_1 a_2}{\gamma}\right]^{2/3}, \tag{8}$$

and the quantity in the log term is $\mathcal{D} = \max(b_0 a_1 \bar{W} a_*/b_1, b_1 a_2 C/\gamma)$ where $\bar{W}$ is the number of hidden neurons before pooling.

Remarks:

1. Just as in the fully connected case, the implicit dependence on the number of classes is only through an L^2 norm of the full last layer matrix. b_1 is a an upper bound on the L^2 norms of hidden activations.
2. a_1 is the norm of the filter matrix A^1, which counts each filter only once regardless of how many times it is applied. This means our bound enjoys only logarithmic dependence on input size for a given stride, and *takes weight sharing into account*.
3. As explained in more detail at the end of Appendix H, there is also no explicit dependence on the size of the filters and the bound is stable through up-resolution. In fact, there is no explicit non logarithmic dependence on architectural parameters, and the bounds converges to 0 as a_1, a_2 tend to zero (in contrast to parameter space bounds such as (Long and Sedghi 2020)).
4. a_* replaces the spectral norm of A^2, and is only equal to the maximum L^2 norm of the second layer weight vectors corresponding to each class. This improvement, just as the improved dependence on the number of classes, comes from better exploiting the continuity of margin based losses with respect to the L^∞ norm.
5. The spectral norm of the first layer matrix $\tilde{A}_1$ is not necessary and is absorbed into an empirical estimate of the hidden layer norms. The first term in the max relates to the estimation of the risk of a test point presenting with a hidden layer norm higher than (a multiple of) b_1.
6. b_0 refers to the maximum L^2 norm of a single convolutional patch over all inputs and patches. In particular, the bound *exploits the sparsity of connections in CNNs.*

A result for the multi-layer case We assume we are given training and testing points $(x, y), (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ drawn iid from any probability distribution over $\mathbb{R}^d \times \{1, 2, \ldots, C\}$. We suppose we have a convolutional architecture so that for each filter matrix $A^l \in \mathbb{R}^{m_l \times d_l}$ from layer $l-1$ to layer l, we can construct a larger matrix $\tilde{A}^l$ representing the corresponding (linear) convolutional operation. The 0^{th} layer is the input,

[3] Replacing the $L^{2,1}$ norm by a L^2 norm without accumulating factors of the numbers of classes is the more substantial contribution. On the other hand, the replacement of the spectral norm is down to better Lipschitz management and has probably been achieved elsewhere. We know of one paper that provides a similar improvement under specific assumptions (the last layer weights being fixed and initialised as independent Bernouilli distributions) (Zou et al. 2018)

[4] It is common practice to leave the post hoc step to the reader in this way. Cf.,e.g., (Long and Sedghi 2020))

[5] This is less than the number of convolutional patches in the input and is not influenced by the number of filters applied.

whist the L^{th} layer is the output/loss function. We write w_l for the *spacial* width at layer l, W_l for the total width at layer l (including channels), and W for $\max_l W_l$. For simplicity of presentation, we assume that the activation functions are composed only of ReLu and max pooling.

Theorem 3. *With probability $\geq 1-\delta$, every network $F_{\mathcal{A}}$ with fliter matrices $\mathcal{A} = \{A^1, A^2, \ldots, A^L\}$ and every margin $\gamma > 0$ satisfy:*

$$\mathbb{P}\left(\arg\max_j (F_{\mathcal{A}}(x)_j) \neq y\right)$$

$$\leq \widehat{R}_\gamma(F_{\mathcal{A}}) + \widetilde{\mathcal{O}}\left(\frac{R_{\mathcal{A}}}{\sqrt{n}}\log(\bar{W}) + \sqrt{\frac{\log(1/\delta)}{n}}\right), \quad (9)$$

where $\bar{W}$ is the maximum number of neurons in a single layer (before pooling) and

$$R_{\mathcal{A}}^{2/3} = \sum_{l=1}^{L}(T_l)^{2/3}$$

for where $T_l =$

$$B_{l-1}(X)\|(A^l - M^l)^\top\|_{2,1}\sqrt{w_l}\max_{U\leq L}\frac{\prod_{u=l+1}^{U}\|\tilde{A}^u\|_{\sigma'}}{B_U(X)}$$

if $l \leq L-1$ and for $l = L$, $T_l =$

$$\frac{B_{L-1}(X)}{\gamma}\|A^L - M^L\|_{\mathrm{Fr}}.$$

Here, w_l is the width at layer l after pooling. By convention, $b_L = \gamma$, and for any layer l_1, $B_{l_1}(X) := \max_i \left|F^{0\to l_l}(x_i)\right|_{l_1}$ denotes the maximum l^2 norm of any convolutional patch of the layer l_1 activations, over all inputs. For $l \leq L-1$, $\|\tilde{A}_l\|_{\sigma'} \leq \|\tilde{A}_l\|$ denotes the maximum spectral norm of any matrix obtained by deleting, for each pooling window, all but one of the corresponding rows of $\tilde{A}$. In particular, for $l = L$, $\|\tilde{A}^L\|_{\sigma'} = \rho_L \max_i \|A^L_{i,\bullet}\|_2$.Here $A^L_{i,\bullet}$ denotes the i'th row of A^L, and $\|\cdot\|_2$ denotes the Frobenius norm.

Similarly to the two layer case above, a notable property of the above bounds is that the norm involved is that of the matrix A^l (the filter) instead of $\tilde{A}^l$ (the matrix representing the full convolutional operation), which means we are only adding the norms of each filter once, regardless of how many patches it is applied to. As a comparison, although the generalization bound in (Bartlett, Foster, and Telgarsky 2017) also applies to CNNs, the resulting bound would involve the whole matrix $\widetilde{A}$ ignoring the structure of CNNs, yielding an extra factor of O_{l-1} instead of $\sqrt{w_l}$, where O_l denotes the number of convolutional patches in layer l: through exploiting weight sharing, we remove a factor of $\sqrt{O_{l-1}}$ in the l^{th} term of the sum compared to a standard the result in (Bartlett, Foster, and Telgarsky 2017), and we remove another factor of $\sqrt{O_{l-1}/w_l}$ through exploitation of the L^∞ continuity of max pooling and our use of L^∞ covering numbers.

A further significant improvement is in replacing the factor $\|X\|_{2,2}\prod_{i=1}^{l-1}\|\tilde{A}_i\|_\sigma$ from the classic bound by $B_{l-1}(X)$, which is the maximum L^2 norm of a single convolutional patch. This implicitly removes another factor of $\sqrt{O_{l-1}}$, this time from the local connection structure of convolutions.

We note that it is possible to obtain more simple bounds without a maximum in the definition of T_l by using the spectral norms to estimate the norms at the intermediary layers.

Empirical spectral norms; Lipschitz augmentation

A commonly mentioned weakness of norm-based bounds is the dependence on the product of spectral norms from above. In the case of fully connected networks, there has been a lot of progress last year on how to tackle this problem. In particular, it was shown in (Nagarajan and Kolter 2019) and in (Wei and Ma 2019) that the products of spectral norms can be replaced by empirical equivalents, at the cost of either a factor of the minimum preactivation in the Relu case (Nagarajan and Kolter 2019), or Lipschitz constant of the *derivative* of the activation functions if one makes stronger assumptions (Wei and Ma 2019). In the appendix, we adapt some of those techniques to our convolutional, ReLu situation and find that the quantity $\rho_l^{\mathcal{A}}$ can be replaced in our case by: $\rho_l^{\mathcal{A}} = \max\left(\max_i \max_{\tilde{l}\geq l}\frac{\rho^{\mathcal{A},x_i}_{l_1\to l_2}}{B_{l_2}(X)}, \max_i \max_{\tilde{l}\geq l}\frac{\theta^{\mathcal{A},x_i}_{l_1\to l_2}}{E_{l_2}(X)}\right)$ where $E_l(X)$ denotes the minimum preactivation (or distance to the max/second max in max pooling) at layer l for over every input, $\rho^{\mathcal{A},x_i}_{l_1\to l_2}$ (resp. $\theta^{\mathcal{A},x_i}_{l_1\to l_2}$) is the Lipschitz constant of gradient of $F^{l_1\to l_2}$ with respect to the norms $|\cdot|_{\infty,l_1}$ and $|\cdot|_{l_2}$(resp. $|\cdot|_{\infty,l_1}$ and $|\cdot|_\infty$). These quantities can easily be calculated explicitly: if $M = \nabla_{F^{0\to l_1}(x_i)}F^{l_1\to l_2}$ so that locally around $F^{0\to l_1}(x_i)$, $F^{l_1\to l_2}(x) = Mx$, then $\theta^{\mathcal{A},x_i}_{l_1\to l_2} = \|M^\top\|_{1,\infty}$ and $\rho^{\mathcal{A},x_i}_{l_1\to l_2} = \max_M \|M'\|_{1,2}$ where M' runs over all sub matrices of M obtained by keeping only the rows corresponding to a single convolutional patch of layer l_2.

Note that an alternative approach is to obtain tighter bounds on the worst case Lipschitz constant. Theorem A.1 in the Appendix is a variation of Theorem 3 involving the explicit worst case Lipschitz constants across layers instead of spectral norms. These quantities can then be independently bounded, or made small via regularisation using recent techniques developed in, e.g., (Fazlyab et al. 2019; Latorre, Rolland, and Cevher 2020).

General proof strategy

Some key aspects of our proofs and general results rely on using the correct norms in activation spaces. On each activation space, we use the norm $|\cdot|_\infty$ to refer to the maximum absolute value of each neuron in the layer, the norm $|\cdot|_l$ to refer to the the maximum l^2 norm of a single convolutional patch (at layer l) and $|\cdot|_{\infty,l}$ for the maximum l^2 norm of a single pixel viewed as a vector over channels. Using these norms, we can for each pair of layers l_1, l_2 define the Lipschitz constant $\rho_{l_1\to l_2}$ is the Lipschitz constant of the subnetwork $F^{l_1\to l_2}$ with respect to the norms $|\cdot|_{\infty,l_1}$ and $|\cdot|_{l_2}$. Using those norms we can formulate a cleaner extention of Theorem 3 where the quantity $R_{\mathcal{A}}$ can be replaced by

$$\Bigg[\sum_{l=1}^{L-1}\left(B_{l-1}(X)\|A^l - M^l\|_{2,1}\max_{\tilde{l}>l}\frac{\rho_{l\to\tilde{l}}}{B_{\tilde{l}}(X)}\right)^{2/3} + \left(\frac{B_{L-1}(X)}{\gamma}\|A^L - M^L\|_{\mathrm{Fr}}\right)^{2/3}\Bigg]^{3/2},$$

where for any layer l_1, $B_{l_1}(X) := \max_i \left|F^{0\to l_1}(x_i)\right|_{l_1}$ denotes the maximum l^2 norm of any convolutional patch of the layer l_1 activations, over all inputs. $B_L(X) = \gamma$. Our proofs derive this result, and the Theorems above follow. cf. Theorem A.1[6].

In the rest of this Section, we sketch the general strategy of the proof, focusing on the (crucial) one-layer step. At this point, we need to introduce notation w.r.t. the convolutional channels: we will collect the data matrix of the previous layer in the form of a tensor $X \in \mathbb{R}^{n\times U\times d}$ consisting of all the convolutional patch stacked together: if we fix the first index (sample i.d.) and the second index (patch i.d.), we obtain a convolutional patch of the corresponding sample. For a set of weights $A \in \mathbb{R}^{d\times m}$, the result of the convolutional operation is written XA where is defined by $(XA)_{u,i,j} = \sum_{o=1}^{d} X_{u,i,o}A_{o,j}$ for all u, i, j.

A key step in bounding the capacity of NN's is to bound the covering numbers of individual layers. Recall the definition.

Definition 1 (Covering number)**.** Let $V \subset \mathbb{R}^n$ and $\|\cdot\|$ be a norm in $\mathbb{R}^n$. The covering number w.r.t. $\|\cdot\|$, denoted by $\mathcal{N}(V,\epsilon,\|\cdot\|)$, is the minimum cardinality m of a collection of vectors $\mathbf{v}^1,\ldots,\mathbf{v}^m \in \mathbb{R}^n$ such that $\sup_{\mathbf{v}\in V}\min_{j=1,\ldots,m}\|\mathbf{v}-\mathbf{v}^j\| \le \epsilon$. In particular, if $\mathcal{F}\subset\mathbb{R}^{\mathcal{X}}$ is a function class and $X = (x_1, x_2, \ldots, x_n) \in \mathcal{X}^n$ are data points, $\mathcal{N}(\mathcal{F}(X),\epsilon,(1/\sqrt{n})\|\cdot\|_2)$ is the minimum cardinality m of a collection of functions $\mathcal{F} \ni f^1,\ldots,f^m : \mathcal{X}\to\mathbb{R}$ such that for any $f\in\mathcal{F}$, there exists $j \le m$ such that $\sum_{i=1}^{n}(1/n)\left|f^j(x_i) - f(x_i)\right|^2 \le \epsilon^2$. Similarly, $\mathcal{N}(\mathcal{F}(X),\epsilon,\|\cdot\|_\infty)$ is the minimum cardinality m of a collection of functions $\mathcal{F}\ni f^1,\ldots,f^m : \mathcal{X}\to\mathbb{R}$ such that for any $f\in\mathcal{F}$, there exists $j\le m$ such that $i \le n, \quad \left|f^j(x_i) - f(x_i)\right| \le \epsilon$.

If we apply classical results on linear classifiers as is done in (Bartlett, Foster, and Telgarsky 2017) (where results on L^2 covering numbers are used) by viewing a convolutional layer as a linear map directly, we cannot take advantage of weight sharing. In this work, we circumvent this difficulty by applying results on the L^∞ covering numbers of classes of linear classifiers to a different problem where each "(convolutional patch, sample, output channel)" combination is mapped into a higher dimensional space to be viewed as a single data point, as explained below. A further reduction in explicit dependence on architectural parameters is achieved by leveraging the L^∞-continuity of margin based loss functions, ReLu activation functions, and pooling. We will make use of the following proposition from (Zhang 2002) (Theorem 4, page 537).

Proposition 4. *Let $n, d \in \mathbb{N}$, $a, b > 0$. Suppose we are given n data points collected as the rows of a matrix $X \in \mathbb{R}^{n\times d}$, with $\|X_{i,\cdot}\|_2 \le b, \forall i = 1,\ldots,n$. For $U_{a,b}(X) = \{X\alpha : \|\alpha\|_2 \le a, \alpha\in\mathbb{R}^d\}$, we have*

$$\log\mathcal{N}\left(U_{a,b}(X),\epsilon,\|\cdot\|_\infty\right) \le \frac{36a^2b^2}{\epsilon^2}\log_2\left(\frac{8abn}{\epsilon}+6n+1\right).$$

Note that this proposition is much stronger than Lemma 3.2 in (Bartlett, Foster, and Telgarsky 2017). In the latter, the cover can be chosen independently of the data set, and the metric used in the covering is an L^2 average over inputs. In Proposition 4, the metric used in the covering is a maximum over all inputs, and the data set must be chosen in advance, though the size of the cover only depends (logarithmically) on the sample size[7].

Using the above proposition on the auxiliary problem based on (input, convolutional patch, ouput channel) triplets, we can prove the following bounds for the one layer case:

Proposition 5. *Let positive reals (a, b, ϵ) and positive integer m be given. Let the tensor $X \in \mathbb{R}^{n\times U\times d}$ be given with $\forall i \in \{1,2,\ldots,n\}, \forall u \in \{1,2,\ldots,U\}, \quad \|X_{i,u,\cdot}\|_2 \le b$. For any choice of reference matrix M, we have*

$$\log\mathcal{N}\left(\{XA : A\in\mathbb{R}^{d\times m}, \|A-M\|_{\mathrm{Fr}} \le a\},\epsilon,\|\cdot\|_\infty\right) \le \frac{36a^2b^2}{\epsilon^2}\log_2\left[\left(\frac{8ab}{\epsilon}+7\right)mnU\right],$$

where the norm $\|\cdot\|_\infty$ is over the space $\mathbb{R}^{n\times U\times m}$.

Sketch of proof: By translation invariance, it is clear that we can suppose $M = 0$. We consider the problem of bounding the L^∞ covering number of $\{(v_i^\top X^j)_{i\le I, j\le J} : \sum_{i\le I}\|v_i\|_2^2 \le a^2\}$ (where $X^j \in \mathbb{R}^{d\times n}$ for all j) with only logarithmic dependence on n, I, J. Here, I plays the role of the number of output channels, while J plays the role of the number of convolutional patches. We now apply the above Proposition 4 on the $nIJ \times dI$ matrix constructed as follows:

$$\begin{pmatrix} X^1 & 0 & \ldots & 0 \\ 0 & X^1 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & X^1 \\ X^2 & 0 & \ldots & 0 \\ 0 & X^2 & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & X^2 \\ X^3 & 0 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ X^J & 0 & \ldots & 0 \\ 0 & X^J & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & X^J \end{pmatrix}^\top,$$

with the corresponding vectors being constructed as $(v_1, v_2, \ldots, v_I) \in \mathbb{R}^{dI}$.

[6] Note that our assumption that the *worst case* Lipschitz constants are bounded removes some of the interactions between layers, yielding a simpler final formula compared to (Wei and Ma 2019; Nagarajan and Kolter 2019)

[7] We note that the proof is also much more obscure, although it is far more approachable to prove an analogous result with a squared log term instead, by going via the shattering dimension.

If we compose the linear map on $\mathbb{R}^{n\times d}$ represented by $(v_1, v_2, \ldots, v_I)^\top$ with k real-valued functions with L^∞ Lipschitz constant 1, the above argument yields comparable bounds on the $\|\cdot\|_2$ covering number of the composition, losing a factor of $\sqrt{k}$ only (for the last layer, $k = 1$, and for convolutional layers, k is the number of neurons in the layer left after pooling).

The proposition above is only enough to deal with the last layer, or a purely l^2 version of our bounds. To prove Theorem 3, which involves $\|\cdot\|_{2,1}$ norms, it is necessary to show the following extension:

Proposition 6. *Let positive reals* (a, b, ϵ) *and positive integer* m *be given. Let the tensor* $X \in \mathbb{R}^{n\times U\times d}$ *be given with* $\forall i \in \{1, 2, \ldots, n\}, \forall u \in \{1, 2, \ldots, U\}$, $\|X_{i,u,\cdot}\|_2 \leq b$. *For any fixed* M*:*

$$\log \mathcal{N}\left(\{XA : A \in \mathbb{R}^{d\times m}, \|A - M\|_{2,1} \leq a\}, \epsilon, \|\cdot\|_*\right)$$
$$\leq \frac{64a^2b^2}{\epsilon^2}\log_2\left[\left(\frac{8ab}{\epsilon} + 7\right) mnU\right],$$

where the norm $\|\cdot\|_*$ *over the space* $\mathbb{R}^{n\times U\times m}$ *is defined by* $\|Y\|_* = \max_{i\leq n}\max_{j\leq U}\left[\sum_{k=1}^m Y_{i,j,k}^2\right]^{\frac{1}{2}}$.

Sketch of proof: We first assume fixed bounds on the L^2 norms $\|A^{i,\cdot}\|_2 = a_i$ of each filter, and use Proposition 5 with $m = 1$ for each output channel with a different granularity ϵ_i. We then optimize over the choice of ϵ_i, and make the result apply to the case where only $a = \sum_i a_i \geq \|A\|_{2,1}$ is fixed in advance by l^1 covering the set of possible choices for $(a_1, a_2, \ldots, a_m)$ for each a, picking a cover for each such choice and taking the union. We accumulate a factor of 2 because of this approach, but to our knowledge, it is not possible to rescale the inputs by factors of $\sqrt{a_i}$ as was done in (Bartlett, Foster, and Telgarsky 2017), as the input samples in an L^∞ covering number bound must be chosen in advance.

We can now **sketch the proof of the Theorem 7 :** we use the loss function

$$l(x_i, y_i) = \max\left[\lambda_{b_1}(\|\sigma(\tilde{A}^1 x_i)\|_2 - b_1),\right.$$
$$\left.\lambda_\gamma\left(\max_{j\neq y}(A^2\sigma(\tilde{A}^1 x_i))_j - (A^2\sigma(\tilde{A}^1 x_i))_{y_i}\right)\right],$$

where for any $\theta > 0$ the *ramp loss* λ_θ is defined by $\lambda_\theta = 1 + \min(\max(x, -\theta), 0)/\theta$. This loss incorporates the following two failure scenarios: (1) the L^2 norm of the hidden activations exceed a multiple of b_1 (2) The activations behave normally but the network still outputs a wrong prediction. Since pooling is continuous w.r.t. the infty norm, the above results for the one layer case applied to a layer yields an ϵ cover of hidden layer w.r.t to the L^∞ norm. The contributions to the error source (1) therefore follows directly from the first layer case. The contribution of the 1st layer cover error to (2) must be multiplied $1/\gamma$ and the Lipschitz constant of A^2 with respect to the $||_{\infty,1}$ and L^∞ norms respectively, which we estimate by $\sqrt{w}a_*$ since the Euclidean norm of the deviation from the cover at the hidden layer is bounded by $\sqrt{w}$ times the deviation in $||_{\infty,1}$ norm [8].

Remarks and comparison to concurrent work

We have addressed the main problems of weight sharing and dependence on the number of classes. As mentioned earlier, (Long and Sedghi 2020) have recently studied the former problem independently of us. It is interesting to provide a comparison of their and our main results, which we do briefly here and in more detail in the Appendix.

The bound in (Long and Sedghi 2020) scales like $\mathcal{C}\sqrt{\frac{\mathcal{W}(\sum_{l=1}^L s_l - \log(\gamma)) + \log(1/\delta)}{n}}$, where s_l is an upper bound on the spectral norm of the matrix corresponding to the l^{th} layer, γ is the margin, and $\mathcal{W}$ is the number of parameters, taking weight sharing into account by counting each parameter of convolutional filters only once. The idea of the proof is to bound the Lipschitz constant of the map from the set of weights to the set of functions represented by the network, and use dimension-dependent results on covering numbers of finite dimensional function classes. Remarkably, this doesn't require chaining the layers, which results in a lack of a non logarithmic dependence on the product of spectral norms. Note that the term $\sum_{l=1}^L s_l$ comes from a log term via the inequality $\prod(1 + s_i) \leq \exp(\sum s_i)$.

On the other hand, the bound scales at least as the square root of the number of parameters, even if the weights are arbitrarily close to initialisation. In contrast, our bound (3) scales like $O(\sqrt{1/n})$ up to log terms when the weights approach initialisation. Furthermore, if we fix an explicit upper bound on the relevant norms (cf.Theorem C.2) [9], **the bound then converges to zero** as the bounds on the norms go to zero. In a refined treatment via the NTK literature (cf. (Arora et al. 2019)), explicit bounds would be provided for those quantities via other tools. In addition, a small modification of the proofs can make the constant towards which the post hoc bounds converges at initialisation arbitrarily small at the cost of slightly worse log terms away from intialisation[10].

Finally, note that the main advantages and disadvantages of our bounds compared to (Long and Sedghi 2020) are connected through a tradeoff in the proof where one can decide which quantities go inside or outside the log. In particular, it is not possible to combine the advantages of both. We refer the reader to Appendix H for a more detailed explanation.

Conclusion

We have proved norm-based generalisation bounds for deep neural networks with significantly reduced dependence on certain parameters and architectural choices. On the issue of class dependency, we have completely bridged the gap between the states of the art in shallow methods and in deep learning. Furthermore, we have, simultaneously with (Long and Sedghi 2020), provided the first satisfactory answer to the weight sharing problem in the Rademacher analysis of neural networks. Contrary to independent work, our bounds are norm-based and are negligible at initialisation.

[8] Recall this $||_{\infty,1}$ norm is a supremum over the spacial locations of the L^2 norms over the channel directions.

[9] The bounds in (Long and Sedghi 2020) and other works deal only with this case, leaving the post hoc case to the reader

[10] The post hoc step from Thm C.2 to (e.g.) Thm 3 is a discrete equivalent to setting priors on the maximum norms $\|(A_l - M_l)^\top\|_{2,1}$.

References

Allen-Zhu, Z.; Li, Y.; and Liang, Y. 2019. Learning and Generalization in Overparameterized Neural Networks, Going Beyond Two Layers. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32*, 6155–6166. Curran Associates, Inc.

Anthony, M.; and Bartlett, P. 2002. *Neural Network Learning: Theoretical Foundations*. ISBN 978-0-521-57353-5. doi: 10.1017/CBO9780511624216.

Arora, S.; Du, S. S.; Hu, W.; Li, Z.; and Wang, R. 2019. Fine-Grained Analysis of Optimization and Generalization for Overparameterized Two-Layer Neural Networks. *arXiv e-prints* arXiv:1901.08584.

Arora, S.; Ge, R.; Neyshabur, B.; and Zhang, Y. 2018. Stronger Generalization Bounds for Deep Nets via a Compression Approach. In Dy, J.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, 254–263. Stockholm, Sweden: PMLR.

Asadi, A.; Abbe, E.; and Verdu, S. 2018. Chaining Mutual Information and Tightening Generalization Bounds. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31*, 7234–7243. Curran Associates, Inc.

Bartlett, P. L. 1998. The sample complexity of pattern classification with neural networks: the size of the weights is more important than the size of the network. *IEEE Transactions on Information Theory* 44(2): 525–536. doi:10.1109/18.661502.

Bartlett, P. L.; Foster, D. J.; and Telgarsky, M. J. 2017. Spectrally-normalized margin bounds for neural networks. 6240–6249. Curran Associates, Inc.

Bartlett, P. L.; and Mendelson, S. 2002. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research* 3(Nov): 463–482.

Cao, Y.; and Gu, Q. 2019. Generalization Bounds of Stochastic Gradient Descent for Wide and Deep Neural Networks. *arXiv e-prints* arXiv:1905.13210.

Chen, M.; Li, X.; and Zhao, T. 2019. On Generalization Bounds of a Family of Recurrent Neural Networks.

Du, S. S.; Wang, Y.; Zhai, X.; Balakrishnan, S.; Salakhutdinov, R. R.; and Singh, A. 2018. How Many Samples are Needed to Estimate a Convolutional Neural Network? In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31*, 373–383. Curran Associates, Inc.

Du, S. S.; Zhai, X.; Poczos, B.; and Singh, A. 2019. Gradient Descent Provably Optimizes Over-parameterized Neural Networks. In *International Conference on Learning Representations*.

Dziugaite, G.; and Roy, D. 2018. Data-dependent PAC-Bayes priors via differential privacy .

Fazlyab, M.; Robey, A.; Hassani, H.; Morari, M.; and Pappas, G. J. 2019. Efficient and Accurate Estimation of Lipschitz Constants for Deep Neural Networks. *CoRR* abs/1906.04893.

Frankle, J.; and Carbin, M. 2019. The Lottery Ticket Hypothesis: Finding Sparse, Trainable Neural Networks. In *International Conference on Learning Representations*.

Golowich, N.; Rakhlin, A.; and Shamir, O. 2018. Size-Independent Sample Complexity of Neural Networks. In Bubeck, S.; Perchet, V.; and Rigollet, P., eds., *Proceedings of the 31st Conference On Learning Theory*, volume 75 of *Proceedings of Machine Learning Research*, 297–299. PMLR.

Goodfellow, I.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. In *International Conference on Learning Representations*.

Guermeur, Y. 2002. Combining Discriminant Models with New Multi-Class SVMs. *Pattern Analysis & Applications* 5(2): 168–179. ISSN 1433-7541. doi:10.1007/s100440200015.

Guermeur, Y. 2007. VC Theory of Large Margin Multi-Category Classifiers. *Journal of Machine Learning Research* 8: 2551–2594.

Guermeur, Y. 2017. Lp-norm Sauer–Shelah lemma for margin multi-category classifiers. *Journal of Computer and System Sciences* 89: 450 – 473. ISSN 0022-0000. doi: https://doi.org/10.1016/j.jcss.2017.06.003.

Harvey, N.; Liaw, C.; and Mehrabian, A. 2017. Nearly-tight VC-dimension bounds for piecewise linear neural networks. In Kale, S.; and Shamir, O., eds., *Proceedings of the 2017 Conference on Learning Theory*, volume 65 of *Proceedings of Machine Learning Research*, 1064–1068. Amsterdam, Netherlands: PMLR.

He, F.; Liu, T.; and Tao, D. 2019. Why ResNet Works? Residuals Generalize. *arXiv e-prints* arXiv:1904.01367.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *CVPR*, 770–778. IEEE Computer Society.

Jacot, A.; Gabriel, F.; and Hongler, C. 2018. Neural Tangent Kernel: Convergence and Generalization in Neural Networks. *CoRR* abs/1806.07572.

Karras, T.; Laine, S.; and Aila, T. 2018. A Style-Based Generator Architecture for Generative Adversarial Networks. *CoRR* abs/1812.04948.

Koltchinskii, V.; and Panchenko, D. 2002. Empirical Margin Distributions and Bounding the Generalization Error of Combined Classifiers. *Ann. Statist.* 30(1): 1–50. doi: 10.1214/aos/1015362183.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In Pereira, F.; Burges, C. J. C.; Bottou, L.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems 25*, 1097–1105. Curran Associates, Inc.

Latorre, F.; Rolland, P.; and Cevher, V. 2020. Lipschitz constant estimation of Neural Networks via sparse polynomial optimization. In *International Conference on Learning Representations*. URL https://openreview.net/forum?id=rJe4_xSFDB.

Lauer, F. 2018. Error bounds with almost radical dependence on the number of components for multi-category classification, vector quantization and switching regression. In *Conférence sur l'Apprentissage automatique (CAp) - French Conference on Machine Learning (FCML)*, Proc. of the French Conference on Machine Learning (CAp/FCML). Rouen, France.

Lee, J.; and Raginsky, M. 2019. Learning Finite-Dimensional Coding Schemes with Nonlinear Reconstruction Maps. *SIAM Journal on Mathematics of Data Science* 1: 617–642. doi:10.1137/18M1234461.

Lei, Y.; Dogan, Ü.; Zhou, D.; and Kloft, M. 2019. Data-Dependent Generalization Bounds for Multi-Class Classification. *IEEE Trans. Information Theory* 65(5): 2995–3021. doi:10.1109/TIT.2019.2893916.

Li, X.; Lu, J.; Wang, Z.; Haupt, J.; and Zhao, T. 2019. On Tighter Generalization Bounds for Deep Neural Networks: CNNs, ResNets, and Beyond.

Lin, S.; and Zhang, J. 2019. Generalization Bounds for Convolutional Neural Networks.

Long, P. M.; and Sedghi, H. 2020. Size-free generalization bounds for convolutional neural networks. In *International Conference on Learning Representations*.

Mohri, M.; Rostamizadeh, A.; and Talwalkar, A. 2018. *Foundations of Machine Learning*. Adaptive Computation and Machine Learning. Cambridge, MA: MIT Press, 2 edition. ISBN 978-0-262-03940-6.

Musayeva, K.; Lauer, F.; and Guermeur, Y. 2019. Rademacher complexity and generalization performance of multi-category margin classifiers. *Neurocomputing* 342: 6 – 15. ISSN 0925-2312. doi:https://doi.org/10.1016/j.neucom.2018.11.096. Advances in artificial neural networks, machine learning and computational intelligence.

Nagarajan, V.; and Kolter, J. Z. 2019. Deterministic PAC-Bayesian generalization bounds for deep networks via generalizing noise-resilience. *CoRR* abs/1905.13344.

Neyshabur, B.; Bhojanapalli, S.; and Srebro, N. 2018. A PAC-Bayesian Approach to Spectrally-Normalized Margin Bounds for Neural Networks. In *International Conference on Learning Representations*. openreview.net.

Neyshabur, B.; Li, Z.; Bhojanapalli, S.; LeCun, Y.; and Srebro, N. 2019, to appear. The role of over-parametrization in generalization of neural networks. In *International Conference on Learning Representations*.

Neyshabur, B.; Tomioka, R.; and Srebro, N. 2015. Norm-Based Capacity Control in Neural Networks. In GrÃŒnwald, P.; Hazan, E.; and Kale, S., eds., *Proceedings of The 28th Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, 1376–1401. Paris, France: PMLR.

Prabhu, Y.; and Varma, M. 2014. FastXML: A Fast, Accurate and Stable Tree-classifier for Extreme Multi-label Learning. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, 263–272. New York, NY, USA: ACM. ISBN 978-1-4503-2956-9. doi:10.1145/2623330.2623651.

Sedghi, H.; Gupta, V.; and Long, P. M. 2019. The Singular Values of Convolutional Layers. In *International Conference on Learning Representations*.

Silver, D.; Hubert, T.; Schrittwieser, J.; Antonoglou, I.; Lai, M.; Guez, A.; Lanctot, M.; Sifre, L.; Kumaran, D.; Graepel, T.; Lillicrap, T.; Simonyan, K.; and Hassabis, D. 2018. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science* 362(6419): 1140–1144. ISSN 0036-8075. doi:10.1126/science.aar6404.

Suzuki, T. 2018. Fast generalization error bound of deep learning from a kernel perspective. In Storkey, A.; and Perez-Cruz, F., eds., *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, 1397–1406. Playa Blanca, Lanzarote, Canary Islands: PMLR.

Wei, C.; and Ma, T. 2019. Data-dependent Sample Complexity of Deep Neural Networks via Lipschitz Augmentation. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; dAlché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32*, 9725–9736. Curran Associates, Inc.

Zhang, C.; Bengio, S.; Hardt, M.; Recht, B.; and Vinyals, O. 2017. Understanding deep learning requires rethinking generalization.

Zhang, J.; Lei, Q.; and Dhillon, I. S. 2018. Stabilizing Gradients for Deep Neural Networks via Efficient SVD Parameterization. In *ICML*, volume 80 of *Proceedings of Machine Learning Research*, 5801–5809. PMLR.

Zhang, T. 2002. Covering Number Bounds of Certain Regularized Linear Function Classes. *J. Mach. Learn. Res.* 2: 527–550. ISSN 1532-4435. doi:10.1162/153244302760200713. URL https://doi.org/10.1162/153244302760200713.

Zhou, W.; Veitch, V.; Austern, M.; Adams, R. P.; and Orbanz, P. 2019. Non-vacuous Generalization Bounds at the ImageNet Scale: a PAC-Bayesian Compression Approach. In *International Conference on Learning Representations*. openreview.net.

Zou, D.; Cao, Y.; Zhou, D.; and Gu, Q. 2018. Stochastic Gradient Descent Optimizes Over-parameterized Deep ReLU Networks. *CoRR* abs/1811.08888.