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

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

1Department of Computer Science, TU Kaiserslautern, 67653 Kaiserslautern, Germany
2School 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,
We consider DNNs parameterized by weight matrices $A$. We now discuss the related work on the statistical learning weight-sharing problem. Their work, which appeared at ICLR 2017 (Neyshabur, Bhojanapalli, and Srebro 2018), used a PAC Bayesian approach to prove an analogous result, where $W = \max\{W_0, W_1, \ldots, W_L\}$ is the width:

$$M = \frac{1}{\sqrt{n}} \prod_{i=1}^{L} \|A^i\|_2 \left( \sum_{i=1}^{L} \frac{\|A^i\|_{\infty}^2}{\|A^i\|_2^2} \right) \gamma \frac{3}{2},$$

where $W = \max\{W_0, W_1, \ldots, W_L\}$ is the width:

$$M = \frac{1}{\sqrt{n}} \prod_{i=1}^{L} \|A^i\|_2 \left( \sum_{i=1}^{L} \frac{\|A^i\|_{\infty}^2}{\|A^i\|_2^2} \right) \gamma \frac{3}{2}. \tag{1}$$

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 $\|A^i\|_F^2$, 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 number of classes. In shallow learning (Lauer 2018; Guermeur 2002, 2011) have been performed to improve the class-size dependency in 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, Musyayevo, 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 spurt 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

$$\tilde{O}(\tilde{M}/\gamma)$$

with

$$M = \frac{1}{\sqrt{n}} \prod_{i=1}^{L} \|A^i\|_2 \left( \sum_{i=1}^{L} \frac{\|A^i\|_{\infty}^2}{\|A^i\|_2^2} \right) \gamma \frac{3}{2}. \tag{2}$$

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

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.
contributions to 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 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 \( M^1, M^2, \ldots, M^L \) representing the initialised values of the weights of the network. Set \( R_\gamma(F_A) = (1/n)(\#\{i : F(x)_{yi} < \gamma + \max_{j \neq y} F(x)_{ji}\}) \), where \# denotes the cardinality of a set. With probability at least \( 1 - \delta \), every network \( F_A \) with weight matrices \( A = (A_1, A_2, \ldots, A_L) \) and every margin \( \gamma > 0 \) satisfy:

\[
\Pr(\arg \max_i F_A(x)_i \neq y) \leq \tilde{R}_\gamma(F_A) + \frac{\sqrt{\log(1/\delta)}}{\sqrt{n}} \left( \sum_{i=1}^{n} \frac{\|x_i\|_{F^1} \|F_A\log(W) + \sqrt{\log(1/\delta)}/n \right),
\]

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

\[
R_A := \rho_L \max_i \|A_{L,i}\|_{F^1} \left( \prod_{i=1}^{L-1} \rho_i \|A^i\|_{\sigma} \right)
\]

\[
\left( \sum_{i=1}^{L-1} \frac{\|A^i - M^i\|_{2,1}^{2/3}}{\|A^i\|_{\sigma}^{2/3}} + \frac{\|A_{L-1}^L\|_{F^1}}{\max_i \|A_{L-1}^i\|_{F^1}^{2/3}} \right)^{2/3}.
\]

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 \( \|A^i - M^i\|_{2,1} \leq \sqrt{W_{i-1} \max_j \|A^j - M^j\|_{2,1}} \), 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 \( L_2 \) 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., \( \|A_{L,i}^L\|_2 \approx a \). (In Section 1 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 \( \|A_{L,i}^L\|_{F^1} \approx \sqrt{W_{L-1}} \), which incurs a square-root dependency on the number of classes. As a comparison, the bound in (Bartlett, Foster, and Telgarsky 2017) involves \( \|A_{L,i}^L\|_{2,1} \approx W_{L-1}^a \), which incurs a linear dependency on the number of classes. If we further impose an \( L_2 \)-constraint on the last layer as \( \|A_{L,i}^L\|_{F^1} \leq a \) 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 $A^2$ in the other terms of the sum is reduced to a dependence on $\max_j \|A_{2,j}\|_2^3$. Both improvements are based on using the $L^\infty$-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 oftennotationally 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 explicit pre chosen norm constraints. 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 spatial dimension of the hidden layer after pooling. 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 $3 \times 3$ 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 $\sigma$ denotes the non-linearities (including both pooling and activation functions). As above, $M^1, M^2$ are the initialised weights.

**Theorem 2.** Let $a_1, a_2, a_3, b_0, b_1 > 0$. 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 $A = (A^1, A^2)$ with weights satisfying $\|(A^1 - M^1)^\top\|_{2,1} \leq a_1, \|A^2 - M^2\|_{F^2} \leq a_2$ and $\sup_{c \leq C} \|A_{2,c}\|_2 \leq a_3$, if $\sup_{1 \leq n} \|A^1_{1:n}\|_2 \leq b_1$, then

$$P \left( \arg \max_j (F_A(x)_j) \neq y \right) \leq R_\gamma(F_A) + 3\sqrt{\frac{\log(\frac{D}{\epsilon})}{2n}} + \frac{C}{\sqrt{n}} \sqrt{\log_2(n^2D)} \frac{1}{2} \log(n),$$

where $C$ is an absolute constant,

$$R^{2/3} = \left[ b_0 a_1 \max \left( 1, \frac{\sqrt{w_a}}{b_1} \right) \right]^{2/3} + \left[ b_1 a_2 \right]^{2/3},$$

and the quantity in the log term is $D = \max(b_0 a_1 w_a, b_1, b_1 a_2 C/\gamma)$ where $W$ 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 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 zero as $a_1, a_2$ tend to zero (in contrast to parameter space bounds such as (Long and Sedghi 2020)).

4. $\alpha$ 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 $L^\infty$ 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,
whist the $L^{th}$ layer is the output/loss function. We write $w_l$ for the spatial width at layer $l$, $W_l$ for the total width at layer $l$ (including channels), and $W$ for max$\times$ $W_l$. 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 $F_A$ with filter matrices $A = \{A^1, A^2, \ldots, A^L\}$ and every margin $\gamma > 0$ satisfy:

$$\mathbb{P}\left( \arg\max_j\left(F_A(x)_j\right) \neq y \right) \leq \hat{R}_\gamma(F_A) + \tilde{O}\left( \frac{R_A}{\sqrt{n}} \log(W) + \sqrt{\frac{\log(1/\delta)}{n}} \right),$$

(9)

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

$$R^2_A = \sum_{l=1}^{L} (T_l)^{2/3}$$

for where $T_l = B_{l-1}(X)\| (A^l - M^l)^\top \|_{2,1} = \max_{u \leq L} \sqrt{\sum_{i=1}^{U} \| \tilde{A}^u_i \|_{\sigma'}} \bigg/ B_U(X)$

if $l \leq L - 1$ and for $l = L$, $T_l = B_{L-1}(X)\| A^L - M^L \|_{F_1}$.

Here, $w_l$ is the width at layer $l$ after pooling. By convention, $b_L = \gamma$, and for any layer $l_1$, $B_{l_1}(X) := \max_{i} \{ F^{0-l_1}(x) \} \rightarrow i$ denotes the maximum $\ell^2$ norm of any convolutional patch of the layer $l_1$ activations, over all inputs. For $l \leq L - 1$,

$$\| \hat{A} \|_{\sigma'} \leq \| \hat{A} \| \sigma'$$

denotes the maximum spectral norm of any matrix obtained by deleting, for each pooling window, all but one of the corresponding rows of $A$. In particular, for $l = L$,

$$\| \hat{A}^L \|_{\sigma'} = \rho_L \max_i \| A^L_i \|_{2,1}, \text{Here } A^L_i \text{ denotes the } i^{th} \text{ row of } A^L, \text{ and } \| \cdot \|_{2,1} \text{ denotes the Frobenius norm.}$$

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 $\hat{A}$ (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) applies also to CNNs, the resulting bound would involve the whole matrix $A$ 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 $L^\infty$ continuity of max pooling and our use of $L^\infty$ covering numbers.

A further significant improvement is in replacing the factor $\|X\|_{2,2} \prod_{i=1}^{L} \| A_i \|_{\sigma'}$ 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 $\rho_l^A$ can be replaced in our case by:

$$\rho_l^A = \max \left( \max_i \| \hat{A}_i \|_{\sigma'}(X), \max_i \| \hat{A}_i \|_{\sigma'}(X) \right)$$

where $E_l(X)$ denotes the maximum preactivation (or distance to the max/second max in max pooling) at layer $l$ for over every input, $\rho_{l-1}^{A_{l-1}}(x_1)$ is the Lipschitz constant of gradient of $F^{l-1} \rightarrow l_2$ with respect to the norms $\| \cdot \|_{1,1}$ and $\| \cdot \|_{2}$ (resp. $\| \cdot \|_{1,1}$ and $\| \cdot \|_{\infty}$). These quantities can easily be calculated explicitly: if $M = \nabla F^{l-1} (x_1) F^{l-1} \rightarrow l_2$ so that locally around $F^{l-1} (x_1)$, $F^{l-1} \rightarrow l_2 (x) = M x$, then

$$\rho_{l-1}^{A_{l-1}} = \| M \|_{1,\infty} \text{ and } \rho_{l-1}^{A_{l-1}} = \max_i \| M_i \|_{1,2}$$

where $M_i$ 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 $\| \cdot \|_{1}$ to refer to the the maximum $L^2$ norm of a single convolutional patch (at layer $l$) and $\| \cdot \|_{1,1}$ 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 $\rho_{l_1 \rightarrow l_2}$ is the Lipschitz constant of the subnetwork $F^{l_1} \rightarrow l_2$ with respect to the norms $\| \cdot \|_{1,1}$ and $\| \cdot \|_{2}$. Using those norms we can formulate a cleaner extention of Theorem 3 where the
quantity $R_A$ can be replaced by
\[
\left[ \sum_{l=1}^{L-1} \left( B_{l-1}(X) \| A^l - M^l \|_{2,1} \max_{l > j} B_{l-1}(X) \right) \right]^{2/3} \frac{1}{\gamma} \left( \frac{B_{L-1}(X)}{\gamma} \right)^{2/3} \frac{1}{3/2}
\]
where for any layer $l_1$, $B_{l_1}(X) := \max_{i \geq 1} |F_{l_1}(x_i)|_{l_1}$ denotes
the maximum $l^2$ norm of any convolutional patch of the first index ($l_1$ activations, over all inputs. $B_{l_1}(X) = \gamma$. Our proofs derive
this result, and the Theorems above follow. cf. Theorem A.1.

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 collect the matrix data 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 $X A$ where is defined by $(X A)_{u,i,j} =
\sum_{a=0}^{d} X_{u,i,A_{a,j}}$ 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. $\| \cdot \|$, de-
noted by $\mathcal{N}(V, \epsilon, \| \cdot \|)$, is the minimum cardinality $m$
of a collection of vectors $v_1^*, \ldots, v_m^* \in \mathbb{R}^n$ such that
$\sup_{v \in V} \min_{j=1,\ldots,m} \| v - v_j^* \| \leq \epsilon$. In particular, if $F \subset \mathbb{R}^X$ is a function class and $X = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$ are
data points, $\mathcal{N}(F(X), \epsilon, (1/\sqrt{n}) \| \cdot \|_{\infty}$) is the minimum cardinality $m$
of a collection of functions $F \ni f^1, \ldots, f^m : \mathcal{X} \to \mathbb{R}$ such that
for any $f \in F$, there exists $j \leq m$ such that $\sum_{i=1}^{n} |f_j(x_i) - f(x_i)|^2 \leq \epsilon^2$. Similarly,
$\mathcal{N}(F(X), \epsilon, \| \cdot \|_{\infty}$) is the minimum cardinality $m$
of a collection of functions $F \ni f^1, \ldots, f^m : \mathcal{X} \to \mathbb{R}$ such
that for any $f \in F$, there exists $j \leq m$ such that $i \leq n$, $|f_j(x_i) - f(x_i)| \leq \epsilon$.

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 $L^\infty$ covering numbers of classes
of linear classifiers to a different problem where each "(con-
volutional 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 $L^\infty$-continuity of margin based loss func-
tions, ReLu activation functions, and pooling. We will make
use of the following proposition from (Zhang 2002) (The-
orem 4, page 537).

*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)

**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
$\| X_{i,} \|_2 \leq b$, $\forall i = 1, \ldots, n$. For $U_{a,b}(X) = \{ X\alpha : \| \alpha \|_2 \leq a, \alpha \in \mathbb{R}^d \}$, we have
\[
\log \mathcal{N}(U_{a,b}(X), \epsilon, \| \cdot \|_{\infty}) \leq \frac{364a^2b^2}{\epsilon^2} \log_2 \left( \frac{8ab}{\epsilon} + 6n + 1 \right).
\]

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.

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 $(a, b, \epsilon)$ and positive integer $m$
be given. Let the tensor $X \in \mathbb{R}^{n \times U \times d}$ be given with
$X_{i,} \in \{ 1, 2, \ldots, n \}$, $\forall u \in \{ 1, 2, \ldots, U \}$, $\| X_{i,} \|_2 \leq b$. For
any choice of reference matrix $M$, we have
\[
\log \mathcal{N}(\{ X A : A \in \mathbb{R}^{d \times m}, \| A - M \|_{\infty} \leq a \}, \epsilon, \| \cdot \|_{\infty}) \leq \frac{364a^2b^2}{\epsilon^2} \log_2 \left( \frac{8ab}{\epsilon} + 7mnU \right),
\]
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 $L^\infty$ covering number of $\{ (v_{i,}^1)_{i,} \in I, J \leq j : \sum_{i,} |v_{i,}^1|_{2} \leq a^2 \}$ (where $X_{i,} \in \mathbb{R}^{d \times n}$ 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:

\[
\begin{pmatrix}
X^1 & 0 & \ldots & 0 \\
0 & X^1 & \ldots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \ldots & X^1 \\
X^2 & 0 & \ldots & 0 \\
0 & X^2 & \ldots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \ldots & X^2 \\
X^3 & 0 & \ldots & 0 \\
0 & X^3 & \ldots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \ldots & X^3 \\
\end{pmatrix}
\]

with the corresponding vectors being constructed as
$(v_{i,1}, v_{i,2}, \ldots, v_{i,J}) \in \mathbb{R}^{dJ}$.

*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
\[(v_1, v_2, \ldots, v_k)^T\] with \( k \) real-valued functions with \( L^\infty \)

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 \( (a, b, \epsilon) \) and positive integer \( m \) be given. Let the tensor \( X \in \mathbb{R}^{n \times U \times m} \) 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 \( M \):

\[
\log \mathcal{N}(\{X : A \in \mathbb{R}^{d \times m}, \|A - M\|_{2,1} \leq a, \epsilon, \|\cdot\|_\ast\}) \\
\leq \frac{6a^2b^2}{\epsilon^2} \log \left( \frac{(ab + 7)mnU}{\epsilon} \right),
\]

where the norm \( \|\cdot\|_\ast \) on the space \( \mathbb{R}^{n \times U \times m} \) is defined by

\[
\|Y\|_\ast = \max_{1 \leq u \leq U} \max_{1 \leq i \leq n} \sum_{k=1}^m Y_{i,u,k}^2.
\]

**Sketch of proof:** We first assume fixed bounds on the \( L^2 \) norms \( \|A^{i,:}\|_2 = a_i \) of each filter, and use Proposition 5 with \( m = 1 \) for each output channel with a different granularity \( \epsilon_i \). We then optimize over the choice of \( \epsilon_i \), and make the result apply to the case where only \( a \) is a \( \epsilon \)-covering 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 \( \sqrt{a} \) as was done in (Bartlett, Foster and Telgarsky 2017), as the input samples in an \( L^\infty \) covering number bound must be chosen in advance.

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

\[
l(x, y) = \max \left[ \lambda_0 \left( \max_{j \neq y} \|\sigma(A^{1,i}x_i)\|_2 - b_1 \right), \right.
\]

\[
\lambda_\epsilon \left( \max_{j \neq y} \left( A^2 \sigma(A^1_i x_i) \right)_j - \left( A^2 \sigma(A^1_i x_i) \right)_{y} \right),
\]

where for any \( \theta > 0 \) the ramp loss \( \lambda_\theta \) 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 infinity norm, the above results for the one layer case applied to a layer yields an \( \epsilon \) cover of hidden layer w.r.t. the \( L^\infty \) norm. The contributions to the error source (1) therefore follows directly from the first layer case. The contribution of the first layer cover error to (2) must be multiplied \( 1/\gamma \) and the Lipschitz constant of \( A^2 \) with respect to the \( \|\cdot\|_{\infty,1} \) and \( L^\infty \) norms respectively, which we estimate by \( \sqrt{w} \), since the Euclidean norm of the deviation from the cover at the hidden layer is bounded by \( \sqrt{w} \) times the deviation in \( \|\cdot\|_{\infty,1} \) norm.\(^8\)

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

**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
\[
C \sqrt{\frac{\sum s_i}{\gamma}} \log(\sqrt{l}/\gamma) + \log(1/\delta),
\]
where \( s_i \) is an upper bound on the spectral norm of the matrix corresponding to the \( i \)-th layer, \( \gamma \) is the margin, and \( W \) 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 \( \sum s_i \) 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 \( O(\sqrt{n}) \) 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 initialisation.\(^10\)

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 novel 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.

\(^9\)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 \( \|A_r - M_r\|_1 \).
Theoretical Foundations


Latorre, F.; Rolland, P.; and Cevher, V. 2020. Lipschitz constant estimation of Neural Networks via sparse polynomial


