On the Generalization of Neural Networks

Nikolaos Tsilivis, Alex Zou

Foundations of Machine Learning (CSCI-GA 2566) final project
May 2023

Abstract

Neural Networks are complex, composite, parameterized functions that are being trained with some variant of gradient descent to learn a target concept. Despite their large capacity, they are very effective in generalizing to samples different than the training ones. In this short survey, we review explanations that have been proposed for this generalization behavior of neural networks.

1 Introduction

Machine Learning can be broadly defined as computational methods using experience to improve performance or to make accurate predictions [11]. In particular, for the task of classification, a computational method, or *hypothesis*, predicts one class out of many possible ones. We typically have access to a finite amount of *training* data, which we use for learning one hypothesis (from a set of, possibly infinite, candidates), and we are concerned with the question of how well our hypothesis will *generalize* to new points (how accurately will predict their classes).

As of 2023, most classification problems are tackled with Deep Learning techniques [10]. Large Neural Networks compute hierarchical representations of the input and they are being trained with some variant of gradient descent on vast amounts of data. Herein, we will consider the most simple version of a multi-layer neural network. Let $\mathcal{X} \subseteq \mathbb{R}^d$ be an input space, $\mathcal{Y}$ be an output space of cardinality $|\mathcal{Y}| = k$, $A_1, A_2, \dots, A_L$ be real matrices (called weight matrices) of size $n_1 \times d, n_2 \times n_1, \dots, n_{L-1} \times k$, and $\sigma : \mathbb{R} \to \mathbb{R}$ be a ρ-Lipschitz function (called activation function), then a neural network $f : \mathcal{X} \to \mathbb{R}^k$ is the function

$$f(x) = A_L \sigma \left(A_{L-1} \sigma \left(\dots \sigma(A_1 x) \dots \right) \right), \tag{1}$$

where σ above is overloaded to be applied elementwise to its input.

Such models have been applied successfully in domains like image recognition [9], speech analysis [6], text analysis [5], biology [8] and chemistry [13], that is they are able to predict correctly on unseen data. Let us introduce some notation to start formalizing these notions, and restrict ourselves, to the case of binary classification with $\mathcal{Y} = \{\pm 1\}$. We denote by $\mathcal{D}$ a distribution over $\mathcal{X} \times \mathcal{Y}$, and we assume access to m training pairs, $S = \{(x_1, y_1), \dots, (x_m, y_m)\}$ drawn i.i.d. from $\mathcal{D}$. We use S to select a hypothesis h from a hypothesis class $\mathcal{H}$ of binary-valued functions. We care about the prediction of h on samples from $\mathcal{D}$ and quantify this with the *true error*: $R(h) = \mathbb{P}_{(x,y)\sim\mathcal{D}} [h(x) \neq y]$. Since $\mathcal{D}$ is most often unknown to us, we only have access to the *empirical error*: $\hat{R}_S(h) = \mathbb{P}_{(x,y)\sim S} [h(x) \neq y] = \frac{1}{m} \sum_{i=1}^{m} 1_{h(x_i) \neq y_i}$. In controlling the difference between the two errors, the *VC-dimension* of $\mathcal{H}$, $VC(\mathcal{H})$ is important; with probability at least

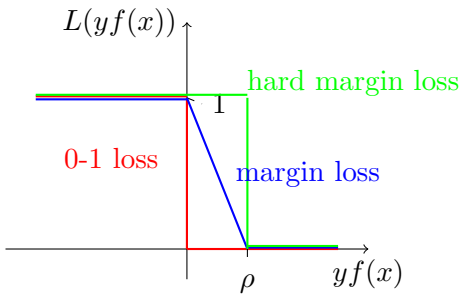

Figure 1: The 0-1 loss commonly used in classification versus the margin loss which grants scale sensitive generalization bounds.

$1-\delta$ (over the random draw of S from $\mathcal{D}$), for any $h \in \mathcal{H}$ it holds [11]

$$R(h) \leq \hat{R}_S(h) + \mathcal{O}\left(\sqrt{\frac{\log(m/VC(\mathcal{H}))}{m/VC(\mathcal{H})}}\right). \tag{2}$$

From the above inequality, one can see that by using m samples, with $m \in \Omega(VC(\mathcal{H}))$, they can be sure that if the empirical error is small, then the true error will also be small (i.e. h will generalize). The issue with neural networks, however, starts from the empirical observations that, in order to generalize, they typically require much less samples than their VC dimension (for typical activation functions it is a function of the number of parameters). For instance, AlexNet that fueled the deep learning revolution in Computer Vision [9] has roughly 60m parameters and it is able to generalize from training on only 1.2m labeled data.

One issue with the previously sketched analysis is that the VC-dimension is a scale-insensitive measure of complexity. It doesn't take into account the confidence of the hypothesis on its predictions. An analysis that circumvents this is based on the so-called *Rademacher Complexity* of a hypothesis class and the notion of *margin*. For a function class $G = \{g : Z \to [a, b]\}$, we define its *empirical Rademacher Complexity* on a sample $S = (z_1, \ldots, z_m)$ as the maximum correlation of a $g \in G$ with binary noise:

$$\hat{\mathfrak{R}}_S(G) = \mathbb{E}_\sigma\left[\sup_{g \in G} \frac{1}{m}\sum_{i=1}^{m} \sigma_i g(z_i)\right], \tag{3}$$

where $\sigma_i \sim Unif(\{-1, +1\})$. Typically G is thought as being a family of functions consisting of a loss function composed with hypotheses from an $\mathcal{H}$. For instance, the loss could be the 0-1 loss, which we used in our definition of error previously. Another loss that penalizes predictions with low confidence $|f(x)|$ is the so-called *margin loss* (see Figure 1):

$$L_\rho(yf(x)) = \begin{cases} 1, & \text{if } yf(x) < 0 \\ 1 - \frac{yf(x)}{\rho}, & \text{if } 0 \leq yf(x) \leq \rho \\ 0, & \text{if } yf(x) > \rho. \end{cases} \tag{4}$$

Based on the above definitions, we can obtain a general generalization bound that is now scale sensitive. Let $\mathcal{H}$ be a real valued hypothesis class, and let ρ be a positive number. With probability at least $1-\delta$, for any $h \in \mathcal{H}$ it holds [11]

$$R(h) \leq \frac{1}{m}\sum_{i=1}^{m} L_\rho(y_i f(x_i)) + \frac{2}{\rho}\hat{\mathfrak{R}}_S(\mathcal{H}) + 3\sqrt{\frac{\log(2/\delta)}{2m}}. \tag{5}$$

We call this analysis scale-sensitive, because, if the predictor h has 0 empirical margin loss on all points for a large ρ (large, correct, confidence), then the complexity term $\frac{2}{\rho}\hat{\mathfrak{R}}_S(\mathcal{H})$ will be small, and hence the bound tight. Therefore, in order to argue about the generalization of a neural network, one must compute (or upper bound) the Rademacher complexity (eq. (3)) for the hypothesis class consisting of neural networks of the form of eq. (1)[1]. Most of the results that we will present follow this route.

The rest of this survey is organised as follows: Section 2 reviews the first generalization bound for neural networks that was scale sensitive and dates back in 1996 [2]. Section 3 makes a leap in time and presents a generalization bound that follows closely the general techniques we presented above and succeeds in upper bounding the Rademacher Complexity by what is termed as the *spectral complexity* of the network [3]. In section 4, we will present a similar bound [12], that follows from the so-called PAC-Bayes framework[2]. Section 5 presents a fundamentally different idea based on the idea of compressing networks [1]. Finally, Section 6 presents some experiments on measuring bounds with trained neural networks in various settings.

2 Beyond VC-dimension type bounds

As already mentioned, for typical neural networks, like multilayer threshold ones, their VC dimension scales at least as fast as the number of weights of the network [4]. However, scale sensitive analyses demonstrate that it is the magnitude of the weights that play a crucial role in generalization, as suggested by the (explicitly titled) paper "For valid generalization, the size of the weights is more important than the size of the network" [2].

That work considered the task of binary classification, real valued hypotheses classes $\mathcal{H}$ and proved a bound on the generalization error of any $h \in \mathcal{H}$ using the notion of γ fat-shattering dimension of $\mathcal{H}$.

Definition 2.1. *Let $\gamma > 0$ be a scale parameter. We say that a sequence $x_1, \ldots, x_m$ is γ-shattered by $\mathcal{H}$, if there exists $r_1, \ldots, r_m$ such that for all binary labels $t_1, \ldots, t_m \in \{\pm 1\}^m$, there is an $h \in \mathcal{H}$ such that*

$$(h(x_i) - r_i)\, t_i \geq \gamma \quad \forall i \in [m]. \tag{6}$$

*The **fat-shattering dimension** of $\mathcal{H}$ at γ, denoted $\mathrm{fat}_{\mathcal{H}}(\gamma)$, is the size of the largest γ-shattered sequence.*

This definition captures the idea of margin, already presented in the introduction, so it comes as no surprise that the generalization bound that follows resembles that of eq. (5).

Theorem 2.2. *Let $\gamma \in (0, 1)$. With probability at least $1 - \delta$, for any $h \in \mathcal{H}$ it holds:*

$$R(h) \leq \frac{1}{m}\sum_{i=1}^{m} L_{\gamma}^{0-1}(y_i f(x_i)) + \sqrt{\frac{2\left(\mathrm{fat}_{\mathcal{H}}(\gamma/16)\ln\left(34em/\mathrm{fat}_{\mathcal{H}}(\gamma/16)\right)\log(578m) + ln(4/\delta)\right)}{m}}, \tag{7}$$

where $L_{\gamma}^{0-1}(f(x)y)$ is the hard margin loss (shown with green in Figure 1).

[1] It is instructive for the reader to try to compute eq. (3) for a linear function. What kind of constraints would we like our predictor to satisfy?

[2] The PAC-Bayes framework originally argues about randomized predictors. However, it is often presented in the literature as a competing framework for explaining generalization for *derandomized* predictors as well, like neural networks. Its edge (if any) over the type of analysis we presented in our introduction, is unclear as it can be shown to stem from Rademacher Complexity type analysis (see Appendix H in [7])

We remark that indeed this bound is qualitatively similar to that of eq. (5): first, the margin loss can always be upper bounded by the hard margin loss, and, second both of the complexity terms decrease with increasing margin ρ / scale γ.

The fat-shattering dimension, as the Rademacher Complexity, of a function class that is a bounded linear combination of other functions can be upper bounded by a function of the dimension of the individual ones. As such, for a neural network of the form (1) with sigmoid activations and a constraint on the group norm of all the weight matrices and on the maximum element of the input, $\|A_i\|_{1,\infty} \leq C, \|x\|_\infty \leq B$, one can show $\text{fat}_{\mathcal{H}}(\gamma) \in \tilde{\mathcal{O}}(B^2 C^{L(L+1)/2} \log d/\gamma^{2L})$, where d is the dimension of the input. If this upper bound gets combined with Theorem 2.2, then one gets what is, to the best of our knowledge, the first generalization bound that appeared in the literature that does not explicitly depend on the amount of the parameters, or the width, of the network. Notice, however that there is an exponential dependence on the depth of the network L.

The proof of Theorem 2.2 goes through covering numbers for the hypothesis class $\mathcal{H}$ with respect to the ℓ_∞ norm. The upper bound on the fat-shattering dimension of the network uses also covering numbers techniques and Maurey's sparsification lemma.

3 Spectrally-normalized margin

Between 1996 and 2017 a lot of things happened, but one of them was that neural networks became fashionable again, and questions of the past came to the surface once again; how possibly can neural networks trained with gradient descent generalize if they are able to fit literally any training data [14]?

One quantity that seems important for the sensitivity of neural networks seems to be the product of the spectral norms of the weight matrices. Indeed, for a network of the form (1), it is:

$$\begin{aligned}
\|f(x) - f(y)\|_2 &= \|A_L\sigma\left(A_{L-1}\sigma\left(\dots\sigma(A_1x)\dots\right) - A_L\sigma\left(A_{L-1}\sigma\left(\dots\sigma(A_1y)\dots\right)\right\|_2 \\
&\leq \|A_L\|_\sigma\|\sigma\left(A_{L-1}\sigma\left(\dots\sigma(A_1x)\dots\right) - \sigma\left(A_{L-1}\sigma\left(\dots\sigma(A_1y)\dots\right)\right\|_2 \\
&\leq \rho\|A_L\|_\sigma\|A_{L-1}\sigma\left(\dots\sigma\left(A_1x\right)\dots\right) - A_{L-1}\sigma\left(\dots\sigma\left(A_1y\right)\dots\right)\|_2 \\
&\leq \dots \\
&\leq \left(\prod_{l=1}^{L}\rho\|A_l\|_\sigma\right)\|x-y\|_2,
\end{aligned} \tag{8}$$

where the inequalities hold from the definition of the spectral norm (operator norm 2) and the fact that the activation was assumed to be ρ-Lipschitz. The bound presented in this section will depend on this quantity, $\prod_{l=1}^{L}\rho\|A_l\|_\sigma$, which can be thought as the Lipschitz constant of the whole network. One thing to notice is that whether or not it increases exponentially with depth depends on just the maximum singular values of the weight matrices (if all of them are less than 1, then the product becomes smaller with increasing depth).

Now, we are in a position to present the bound. It is about multi-class classification with k classes and is based on the margin-based analysis that we outlined in the introduction. The difference, now, is that the margin is defined as the minimum difference between the outputs of the network from the true one, i.e. $M(x,y) = f_y(x) - \max_{y'\neq y} f_{y'}(x)$. For each weight matrix A_i, there is a corresponding reference matrix M_i appearing in the bound. We denote by $\|\cdot\|_{p,q}$ the p,q group norm of a matrix, which corresponds to first computing the p-th norm of the columns and then the q-th norm of those values viewed as a vector. Let also W denote the maximum amongst $d, n_1, n_2, \dots, n_{L-1}, k$ (i.e. the width of the network).

Theorem 3.1. *Let weight matrices $A_1, \ldots, A_L$, reference matrices $M_1, \ldots, M_L$, and let $X \in \mathbb{R}^{m \times d}$ be the data matrix (each row is a different input). For any $\gamma > 0$, with probability $1 - \delta$, for any $h \in \mathcal{H}$ (where is $\mathcal{H}$ contains all the networks of the form (1)) it holds:*

$$R(h) \leq R_\gamma(h) + \tilde{\mathcal{O}}\left(\frac{\|X\|_F C_h \ln H}{\gamma m} + \sqrt{\frac{ln(1/\delta)}{m}}\right), \tag{9}$$

*where the **spectral complexity**, C_h, is defined as*

$$C_h = \left(\prod_{i=1}^{L} \rho_i \|A_i\|_\sigma\right) \left(\sum_{i=1}^{L} \frac{\|A_i^\intercal - M_i^\intercal\|_{2,1}^{2/3}}{\|A_i\|_\sigma^{2/3}}\right)^{3/2}. \tag{10}$$

Remarks.

- Despite being a bound for k-class classification, there is no explicit dependence on the number of classes.
- The bound depends logarithmically on the width.
- As mentioned already, the first term of the spectral complexity can be made to decrease with depth (if all the spectral norms are less than one), yet the second term (the sum) is upper bounded by a polynomial of depth. To see that, set $\{M_i\}_{i=1}^L$ to 0, then $\|A_i^\intercal\|_{2,1} \geq \|A_i^\intercal\|_F = \|A_i\|_F \geq \|A_i^\intercal\|_\sigma$, thus the second term is upper bounded by $\sqrt{L^3}$. Hence, there is an unavoidable polynomial dependence on the depth on this bound, which is a bit annoying (very deep neural networks, provided they can be optimized, can generalize just fine!).

The proof consists of 3 main steps. First, we derive a generic margin-bound as in equation (5), but for multi-class classification. Then, the Rademacher Complexity of the neural networks family is upper bounded by an expression that involves the ℓ_2 covering number of this function class, with a standard tool called Dudley's entropy integral bound (it is proven by considering a sequence of coverings at different scales, and invokes, amongst others, Massart's lemma and Riemann's integral formula). The most technical part of the proof covers inductively the hypothesis class, covering a layer at a time. To cover each individual layer, the following lemma is being proven.

Lemma 3.2. *Let conjugate exponents (p, q) and (r, s) be given with $p \leq 2$, as well as positive reals (a, b, ϵ) and positive integer c. Let matrix $X \in \mathbb{R}^{m \times d}$ be given with $\|X\|_p \leq b$. Then*

$$\ln \mathcal{N}\big(\{XA : A \in \mathbb{R}^{d \times c}, \|A\|_{q,s} \leq a\}, \epsilon, \|\cdot\|_2\big) \leq \lceil \frac{a^2 b^2 c^{2/r}}{\epsilon^2} \rceil \ln(2dc), \tag{11}$$

where $\mathcal{N}(\mathcal{F}, \epsilon, \|\cdot\|))$ denotes the covering number of a function class $\mathcal{F}$ at scale ϵ with respect to a norm $\|\cdot\|$.

4 PAC-Bayesian analysis grants margin bounds

The previous bound seemed to explain certain quantities that are important for generalization, however its proof is fairly technical. A similar, but weaker[3], bound [12] can be given starting from the PAC-Bayes framework that is about randomized predictors, and is arguably using simpler techniques.

[3]For a discussion, please refer to the original paper [12].

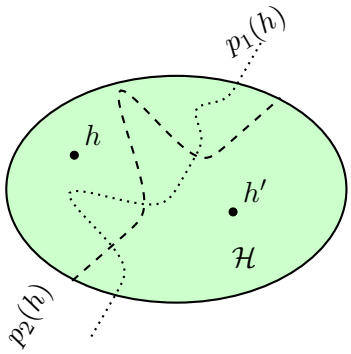

Figure 2: An illustration of the PAC-Bayes framework. A hypothesis class $\mathcal{H}$ is furnished with probability distributions over the hypotheses, and one reasons about the error of randomized predictors, as defined by different distributions.

Informally speaking, the PAC-Bayes framework treats the hypotheses as random variables, with some being more likely than others, and reasons about the expected error of a predictor that follows a specific distribution Q (often called "posterior") based on the distance from a "prior" distribution over the same space (see Figure 2 for an illustration). However, one can "derandomize" the generalization bounds with a lemma that looks like follows:

Lemma 4.1. *Let $f_w(x) : \mathcal{X} \to \mathbb{R}^k$ be any predictor (i.e. neural network) with parameters w, and P be any distribution on the parameters that is independent of the training data. Then, for any $\gamma, \sigma > 0$, with probability $\geq 1 - \sigma$ over the training set of size m, for any w, and any random perturbation u s.t $\mathbb{P}_u[\max_{x \in \mathcal{X}} |f_{w+u}(x) - f_w(x)| \leq \frac{\gamma}{4}] \geq \frac{1}{2}$, we have:*

$$L_0(f_w) \leq \hat{L}_\gamma(f_w) + 4\sqrt{\frac{KL(w+u||P) + \ln \frac{6m}{\sigma}}{m-1}} \tag{12}$$

We will use Lemma 4.1 to derive the bound. The summary of the rest of the proof is as follows: To derive a margin bound, we just need to find a proper distribution for u and P, where u satisfies constraint $\mathbb{P}_u[\max_{x \in \mathcal{X}} |f_{w+u}(x) - f_w(x)| \leq \frac{\gamma}{4}] \geq \frac{1}{2}$. Then we get the bound simply by calculating the $KL(w+u||P)$ in eq. (12).

5 Compression of Networks

6 Extensions

References

[1] Sanjeev Arora et al. "Stronger Generalization Bounds for Deep Nets via a Compression Approach". In: *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*. Ed. by Jennifer G. Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, 2018, pp. 254–263.

[2] Peter L. Bartlett. "For Valid Generalization the Size of the Weights is More Important than the Size of the Network". In: *Advances in Neural Information Processing Systems 9, NIPS, Denver, CO, USA, December 2-5, 1996*. Ed. by Michael Mozer, Michael I. Jordan, and Thomas Petsche. MIT Press, 1996, pp. 134–140.

[3] Peter L. Bartlett, Dylan J. Foster, and Matus Telgarsky. "Spectrally-normalized margin bounds for neural networks". In: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. by Isabelle Guyon et al. 2017, pp. 6240–6249.

[4] Eric B. Baum and David Haussler. "What Size Net Gives Valid Generalization?" In: *Advances in Neural Information Processing Systems 1, [NIPS Conference, Denver, Colorado, USA, 1988]*. Ed. by David S. Touretzky. Morgan Kaufmann, 1988, pp. 81–90.

[5] Tom B. Brown et al. "Language Models are Few-Shot Learners". In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*. Ed. by Hugo Larochelle et al. 2020.

[6] "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups". In: *IEEE Signal Process. Mag.* 29.6 (2012), pp. 82–97. DOI: 10.1109/MSP.2012.2205597. URL: https://doi.org/10.1109/MSP.2012.2205597.

[7] Dylan J. Foster et al. "Hypothesis Set Stability and Generalization". In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. Ed. by Hanna M. Wallach et al. 2019, pp. 6726–6736.

[8] John M. Jumper et al. "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596 (2021), pp. 583–589.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*. Ed. by Peter L. Bartlett et al. 2012, pp. 1106–1114.

[10] Yann LeCun, Yoshua Bengio, and Geoffrey E. Hinton. "Deep learning". In: *Nature* 521.7553 (2015), pp. 436–444.

[11] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. Adaptive computation and machine learning. MIT Press, 2012. ISBN: 978-0-262-01825-8.

[12] Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. "A PAC-Bayesian Approach to Spectrally-Normalized Margin Bounds for Neural Networks". In: *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*.

[13] David Pfau et al. "Ab-Initio Solution of the Many-Electron Schrödinger Equation with Deep Neural Networks". In: *ArXiv* (2019).

[14] Chiyuan Zhang et al. "Understanding deep learning requires rethinking generalization". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. 2017.