Function Norms for Neural Networks*

Amal Rannen-Triki †[1], Maxim Berman[1], Vladimir Kolmogorov[2], and Matthew B. Blaschko[1]

[1]KU Leuven, Belgium; email: name.surname@esat.kuleuven.be
[2]Institute of Science and Technology, Austria; email: vnk@ist.ac.at

Abstract

Deep neural networks (DNNs) have become increasingly important due to their excellent empirical performance on a wide range of problems. However, regularization is generally achieved by indirect means, largely due to the complex set of functions defined by a network and the difficulty in measuring function complexity. There exists no method in the literature for additive regularization based on a norm of the function, as is classically considered in statistical learning theory. In this work, we study the tractability of function norms for deep neural networks with ReLU activations. We provide, to the best of our knowledge, the first proof in the literature of the NP-hardness of computing function norms of DNNs of 3 or more layers. We also highlight a fundamental difference between shallow and deep networks. In the light on these results, we propose a new regularization strategy based on approximate function norms, and show its efficiency on a segmentation task with a DNN.

1. Introduction

In the recent years, computer vision has benefited from the growth of neural network applications. Most of the recent results indicate that larger networks provide better performance. Many works link this increase to the expressivity of the function classes encoded by DNNs. [2] and [13] link expressivity to depth, and define a complexity measures (e.g. transitions, activation patterns) that grow exponentially with the number of layers. [12] takes a differential geometry perspective to show that in certain regimes, the global curvature of the neural network function grows exponentially with depth but not width, showing a clear advantage of deep networks over shallow ones.

With the continual increase of the networks size and complexity, it seems natural to look into maximum a posteriori estimation of the parameters, or equivalently to regularize the training with carefully designed additive measure of the network complexity.

Looking back at classical machine learning algorithms with regularized risk, it appears that direct regularization has often been achieved by penalization of a norm of a function. In the case of linear functions (e.g. Tikhonov regularization [19]), penalizing the parameter vector corresponds to a penalization of a function norm as a straightforward result of the Riesz representation theorem [15]. In the case of reproducing kernel Hilbert space (RKHS) regularization (including splines [21]), penalizing the parameters corresponds by construction to a function norm regularization [20, 16].

In the case of deep neural networks, similar approaches have been applied directly to the parameter vectors, resulting in an approach referred to as weight decay [8]. In contrast to the previously mentioned Hilbert space approaches, this does not directly penalize a measure of function complexity, such as a norm. Indeed, it is straightforward to see that this is not even a function of the mapping encoded by the network, as different weights can result in the same mapping.

In this work, we study the tractability of computing function norms for neural networks. We prove for the first time (to the best of our knowledge) that computing any function norm of a DNN with three or more layers with rectified linear unit (ReLU) activation functions [6] as a function of its parameter values is NP-hard. Moreover, we highlight a fundamental difference between ReLU networks of 2 layers and networks of 3 or more layers by constructing a polynomial-time computable norm for the shallow architectures. This result motivates the use of approximation strategies to be able to use function norm related measures for DNN regularization. We empirically test this new regularization strategy on a segmentation task with a DNN.

2. Function norm based regularization

We consider the supervised training of the weights W of a deep neural network (DNN) given a training set $\mathcal{D} = \{(x_i, y_i)\} \in (\mathcal{X} \times \mathcal{Y})^n$, where $\mathcal{X} \subseteq \mathbb{R}^d$ is the input space

*The first workshop on Statistical Deep Learning for Computer Vision, in *Seoul, Korea*, 2019. Copyright by Author(s).

†This author is currently affiliated with Deepmind.

and $\mathcal{Y} \subseteq \mathbb{R}^s$ the output space. Let $f : \mathcal{X} \to \tilde{\mathcal{Y}}$ be the function encoded by the neural network. We aim to minimize the risk $\mathcal{R}(f) = \int \ell(f(x), y)\, \mathrm{d}P(x, y)$, where P is the underlying joint distribution of the input-output space. As this distribution is generally inaccessible, empirical risk minimization approximates the risk integral by

$$\hat{\mathcal{R}}(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(x_i), y_i), \tag{1}$$

where the elements from the dataset $\mathcal{D}$ are supposed to be i.i.d. samples drawn from $P(x, y)$.

When the number of samples n is large, the empirical risk (1) is a good approximation of the risk integral. In the small-sample regime, however, better control of the generalization error can be achieved by adding a regularization term to the objective.

In the statistical learning theory literature, this is most typically achieved through an additive penalty [20, 10]

$$\arg\min_{f} \hat{\mathcal{R}}(f) + \lambda \Omega(f), \tag{2}$$

where Ω is a measure of function complexity. The regularization biases the objective towards "simpler" candidates in the model space.

In machine learning, using the norm of the learned mapping appears as a natural choice to control its complexity. This choice limits the hypothesis set to a ball in a certain topological set depending on the properties of the problem. In an RKHS, the natural regularizer is a function of the Hilbert space norm: for the space $\mathcal{H}$ induced by a kernel k, $\|f\|_{\mathcal{H}}^2 = \langle f, f \rangle_{\mathcal{H}}$. Several results showed that the use of such a regularizer results in a control of the generalization error [5, 21, 1]. In the context of function estimation, for example using splines, it is customary to use the norm of the approximation function or its derivative in order to obtain a regression that generalizes better [22].

However, for neural networks, defining the best prior for regularization is less obvious. The topology of the function set represented by a neural network is still fairly unknown, which complicates the definition of a proper complexity measure.

Nevertheless, if the activation functions are continuous, any function encoded by a network is in the space of continuous functions. Moreover, supposing the input domain $\mathcal{X}$ is compact, the network function has a finite L_q-norm

$$\|f\|_q = \left(\int \|f(x)\|_q^q \, \mathrm{d}\mu(x) \right)^{\frac{1}{q}}, \tag{3}$$

where the inner norm represents the q-norm of the output space.

In Sec. 4, we will focus on the special case of L_2. This function space has attractive properties, being a Hilbert space.

However, because of the high dimensionality of neural network function spaces, the optimization of function norms is not an easy task. Indeed, the exact computation of any of these function norms is intractable for networks with ReLU activations, as we show in the following section.

3. Tractability of function norm computation

In this section, we will study separately the networks of 2 or less layers and the networks of 3 or more layers.

3.1. Shallow networks

Definition 1 *Let $\mathcal{H}$ be an RKHS associated with the kernel k over the topological space X. k is **characteristic** [17] if the mapping $P \mapsto \int_X k(., x) dP(x)$ from the set of all Borel probability measures defined on X to $\mathcal{H}$ is injective.*

For example, the Gaussian kernel over $\mathbb{R}^d$ is characteristic.

Proposition 1 *Given a 2-layer neural network f mapping from $\mathbb{R}^d$ to $\mathbb{R}$ with m hidden units, and a kernel k characteristic over $\mathbb{R}^d$, there exists a function norm $\|f\|$ that can be computed in a quadratic time in m and the cost of evaluation of k (assuming we allow a square root operation). For example, for a Gaussian kernel, the cost of the kernel evaluation is linear in d and the function norm can be computed in $\mathcal{O}(m^2 d)$ (assuming that we allow square root and exponential operations).*

We define a norm on two layer ReLU networks by defining an inner product through a RKHS construction based on a characteristic kernel [17] after normalization of the network weights. The kernel must be characteristic to guarantee that the resulting norm is zero iff $f = \mathbf{0}$. The exact construction is provided in [14, Appendix F] due to space constraints here.

3.2. Deep networks

Proposition 2 *For f defined by a deep neural network (of depth greater or equal to 3) with ReLU activation functions, deciding if any function norm $\|f\| = 0$ from the weights of a network is NP-hard.*

To prove this statement, we construct a network of depth 3 for which the Max-Cut problem [7] can be reduced in linear time to deciding if the norm of the function is zero.

Definition 2 (Cut) *Given a graph $G = (V, E)$, a cut is a partition of the vertex set V into two disjoint subsets $S \subseteq V$ and $V \setminus S$. The size of the cut is $|\operatorname{cut}(G, S)| = |\{\{i, j\} \in E, \;\; i \in S, \; j \in V \setminus S\}|$.*

Problem 1 (Max-cut decision problem) *Given a graph G and an integer k, decide if there is a cut of at least size k in G. We denote the truth of this statement as* $\text{Max-Cut}(G, k)$.

Given graph G, we define function $f : \mathbb{R}^V \to \mathbb{R}$ as follows:

$$
\begin{aligned}
f(x) &= h\left(\sum_{(i,j)\in E} f_{ij}(x)\right) \\
h(z) &= \max(0, z-k+1) \\
f_{ij}(x) &= g_1(x_i - x_j) + g_1(x_j - x_i) \\
g_1(z) &= g_0(z-1) \\
g_0(x) &= \varepsilon^{-1}[\max(0, x+\varepsilon) - 2\max(0,x) + \max(0, x-\varepsilon)]
\end{aligned}
$$

where $\varepsilon = \frac{1}{4|V|}$. Note that $g_0(z), g_1(z)$ place some non-zero values in the ε neighborhood of $z = 0$ and $z = 1$, respectively, and zero elsewhere. Furthermore, $g_0(0) = 1$ and $g_1(1) = 1$ (see Figure 1).

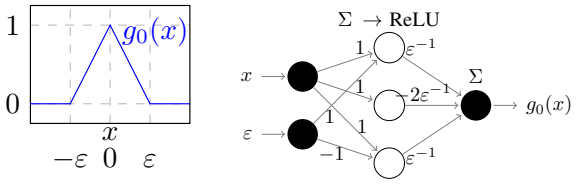

Figure 1: Plot of function g_0, and network computing this function.

Lemma 1 *f is continuous by composition of continuous functions.*

Corollary 1 $\|f\| = 0 \iff f(x) = 0$ *everywhere.*

Proposition 3 $\|f\| \neq 0$ *iff* Max-Cut$(G, k) =$ *true.*

Proof: First, suppose that Max-Cut(G, k) is true. Let S define a partition with $|\operatorname{cut}(G, S)| \geq k$. For $i \in S$, set $x_i = 1$. For $j \in V \setminus S$ set $x_j = 0$. We have

$$\sum_{\{i,j\}\in E} f_{ij}(x) = |\operatorname{cut}(G, S)| \geq k. \tag{4}$$

This implies that $f(x) \geq 1$ and thus $\|f\| > 0$ by Corollary 1.

Now suppose that $\|f\| > 0$, then $f(x) > 0$ for some $x \in \mathbb{R}^V$. Define $E' = \{\{i, j\} \in E : f_{ij}(x) > 0\}$, and let $\mathcal{C}$ be the set of connected components of graph (V, E'). For node $i \in C \in \mathcal{C}$ denote $y_i = x_i - \min_{j\in C} x_j$ and $z_i = \operatorname{round}(y_i)$. Let $S = \{i \in V \mid z_i \text{ is even}\}$. We will show next that $|\operatorname{cut}(G, S)| \geq k$, thus proving the claim.

Condition $f(x) > 0$ means that $\sum_{\{i,j\}\in E} f_{ij}(x) > k-1$. Since $f_{ij}(x) \in [0, 1]$ for each $\{i, j\} \in E$, there must be at least k edges in E with $f_{ij}(x) > 0$. Thus, $|E'| \geq k$, and so it suffices to show that $E' \subseteq \operatorname{cut}(G, S)$. By construction,

$$|y_i - y_j| = |x_i - x_j| \in (1-\varepsilon, 1+\varepsilon) \qquad \forall (i, j) \in E'. \tag{5}$$

We claim that $|y_i - z_i| \leq \frac{1}{4}$ for each $i \in V$. Indeed, let $C \in \mathcal{C}$ be the component containing i. Pick $i^* \in \arg\min_{i\in C} x_i$, then $y_{i^*} = 0$. Let P be a path from i^* to i in (V, E'), then $y_i = \sum_{(i',j')\in P}(y_{j'} - y_{i'})$. Eq. (5) now gives $|y_i - \operatorname{round}(y_i)| \leq |P| \cdot \varepsilon \leq |V| \cdot \varepsilon = \frac{1}{4}$, as claimed.

Now consider edge $\{i, j\} \in E'$. Conditions $|y_i - z_i| \leq \frac{1}{4}$, $|y_j - z_j| \leq \frac{1}{4}$ and $|y_j - y_i| \in (\frac{1}{2}, \frac{3}{2})$ imply that $|z_i - z_j| = 1$, and therefore $\{i, j\} \in \operatorname{cut}(G, S)$.

□

Corollary 2 *For a neural network with ReLU activations and three or more layers, deciding whether the function has non-zero norm is NP-hard by polynomial-time reduction from the max-cut decision problem.*

That hardness of a three layer network implies hardness of deeper networks is straightforward as we may implement the identity function by

$$x = \max(0, x) - \max(0, -x). \tag{6}$$

Theorem 1 (Riesz representation theorem [3]) *For $\mathcal{H}$ a Hilbert space an $L : \mathcal{H} \mapsto \mathbb{R}$ or $\mathbb{C}$ a bounded linear functional, there exists a unique vector $h_0 \in \mathcal{H}$ such that:*

$$\forall h \in \mathcal{H}, L(h) = \langle h, h_0 \rangle, \tag{7}$$

and we have $\|L\| = \|h_0\|$.

Proposition 1 and Proposition 2, in combination with the Riesz representation theorem for linear functions (Theorem 1), shows dichotomy in the complexity of norm computation of (deep) neural networks: for two layer networks, we have polynomial time [14] computation, while for networks of depth 3 or higher with ReLU activations, norm computation is NP-hard (Corollary 2).

4. Regularization with approximate norm

In the previous section, we have shown that the exact computation of any function norm for a DNN with ReLU activations is intractable, motivating the need of a stochastic approximation. Assuming the measure μ in the definition of the L_q-norm is a probability measure Q, the function norm can be written as $\|f\|_{2,Q} = \mathbb{E}_{z\sim Q}\left[\|f(z)\|_2^2\right]^{1/2}$. Moreover, assuming we have access to i.i.d samples $z_j \sim Q$, this weighted L_2-function norm can be approximated by

$$\left(\frac{1}{m}\sum_{i=1}^{m} \|f(z_i)\|_2^2\right)^{\frac{1}{2}}. \tag{8}$$

For samples outside the training set, empirical estimates of the squared weighted L_2-function norm are U-statistics of order 1, and have an asymptotic Gaussian distribution to which finite sample estimates converge quickly as $\mathcal{O}(m^{-1/2})$ [9]. In the next experiments, we show the efficiency of such a regularization strategy in a semantic image segmentation task with DNNs.

(a) ground truth (b) weight decay (c) weight decay + function norm

Figure 2: ENet outputs, after training on 500 samples of Cityscapes, without (b) and with (c) weighted function norm regularization (standard Cityscape color palette – black regions are *unlabelled* and not discounted in the evaluation.

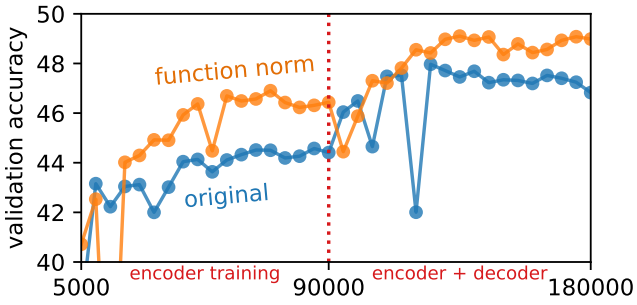

Figure 3: Evolution of the validation accuracy of ENet during training with the network's original regularization settings, and with added weighted function norm regularization.

Regularized training of ENet: We consider the training of ENet [11], a network architecture designed for fast image segmentation, on the Cityscapes dataset [4]. As regularization plays a more significant role in the low-data regime, we consider a fixed random subset of $n = 500$ images of the training set of Cityscapes as an alternative to the full 2975 training images. We compare training ENet similarly to the author's original optimization settings, in a two-stage training of the encoder and the encoder + decoder part of the architecture, using weighted cross-entropy loss. We use Adam with a base learning rate of $2.5 \cdot 10^{-4}$ with a polynomially decaying learning rate schedule and 90000 batches of size 10 for both training stages. We found the validation performance of the model trained under these settings with all images to be 60.77% mean IoU; this performance is reduced to 47.15% when training only on the subset.

We use our proposed approximate function norm regularization using unlabeled samples taken from the 20000 images of the "coarse" training set of Cityscapes, disjoint from the training set. Figure 3 shows the evolution of the validation accuracy during training. We see that the added regularization leads to a higher performance on the validation set. Figure 2 shows a segmentation output with higher performance after adding the regularization.

Results: In our experiments, we were not able to observe an improvement over the baseline in the same setting with a state-of-the-art semi-supervised method, mean-teacher [18]. We therefore believe the observed effect to be attributed to the effect of the regularization. The impact of semi-supervision in the regime of such high resolution images for segmentation is however largely unknown and it is possible that a more thorough exploration of unsupervised methods would lead to a better usage of the unlabeled data.

5. Discussion and Conclusions

While function norms played a central role in regularization for classical machine learning algorithms, regularized training of neural networks has focused mainly on indirect control of complexity. We have shown here for the first time that norm computation in a low fixed depth neural network with ReLU activations is NP-hard, elucidating some of the challenges of working with DNN function classes. This result constitutes a fundamental difference with shallower networks, for which we can construct a polynomial time norm based on the weights. This result motivates the use of stochastic approximations to weighted norm, which is readily compatible with stochastic gradient descent optimization strategies. We empirically validated the expected effect of the employed regularizer with experiments on the training of ENet on Cityscapes.

Acknowledgments

A. Rannen-Triki, M. Berman and M.B. Blaschko acknowledge support from FWO (grant G0A2716N), an Amazon Research Award, an NVIDIA GPU grant, and the Facebook AI Research Partnership.

V. Kolmogorov is supported by the European Research Council under the European Unions Seventh Framework Programme (FP7/2007-2013)/ERC grant agreement no 616160.

References

[1] Olivier Bousquet and André Elisseeff. Stability and generalization. *Journal of Machine Learning Research*, 2(Mar):499–

526, 2002.

[2] Nadav Cohen, Or Sharir, and Amnon Shashua. On the expressive power of deep learning: A tensor analysis. In *Conference on Learning Theory*, pages 698–728, 2016.

[3] John B Conway. *A Course in Functional Analysis*, volume 96. Springer Science & Business Media, 1994.

[4] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[5] Federico Girosi and Tomaso Poggio. Networks and the best approximation property. *Biological cybernetics*, 63(3):169–176, 1990.

[6] Richard HR Hahnloser, Rahul Sarpeshkar, Misha A Mahowald, Rodney J Douglas, and H Sebastian Seung. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 405(6789):947, 2000.

[7] Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

[8] Anders Krogh and John A Hertz. A simple weight decay can improve generalization. *Advances in Neural Information Processing Systems*, 4:950–957, 1995.

[9] Alan James Lee. *U-Statistics: Theory and Practice*. CRC Press, 1990.

[10] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[11] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016.

[12] Ben Poole, Subhaneil Lahiri, Maithra Raghu, Jascha Sohl-Dickstein, and Surya Ganguli. Exponential expressivity in deep neural networks through transient chaos. In *Advances in neural information processing systems*, pages 3360–3368, 2016.

[13] Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl Dickstein. On the expressive power of deep neural networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2847–2854. JMLR. org, 2017.

[14] Amal Rannen Triki, Maxim Berman, and Matthew B. Blaschko. Stochastic weighted function norm regularization. *CoRR*, abs/1710.06703, 2017.

[15] Frédéric Riesz. *Sur une espece de geometrie analytique des systemes de fonctions sommables*. Gauthier-Villars, 1907.

[16] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels*. MIT Press, 2001.

[17] Bharath K Sriperumbudur, Kenji Fukumizu, and Gert RG Lanckriet. Universality, characteristic kernels and rkhs embedding of measures. *Journal of Machine Learning Research*, 12(Jul):2389–2410, 2011.

[18] Antti Tarvainen and Harri Valpola. Weight-averaged consistency targets improve semi-supervised deep learning results. *arXiv preprint arXiv:1703.01780*, 2017.

[19] Andrey Nikolayevich Tikhonov. Solution of incorrectly formulated problems and the regularization method. *Soviet Math. Dokl.*, 4:1035–1038, 1963.

[20] Vladimir Vapnik. *Statistical learning theory*. Wiley, 1998.

[21] Grace Wahba. *Spline models for observational data*, volume 59. Siam, 1990.

[22] Grace Wahba. Splines in nonparametric regression. *Encyclopedia of Environmetrics*, 2000.

Function Norms for Neural Networks: Supplementary Material*

Amal Rannen-Triki †[1], Maxim Berman[1], Vladimir Kolmogorov[2], and Matthew B. Blaschko[1]

[1]KU Leuven, Belgium; email: name.surname@esat.kuleuven.be
[2]Institute of Science and Technology, Austria; email: vnk@ist.ac.at

1. On the complexity of neural network norm computation

1.1. Two-layer networks

We will define a norm on two layer ReLU networks by defining an inner product through a RKHS construction.

A two layer network with a single output can be written as

$$f(x) = w_1^T \sigma(W_2 x) \tag{1}$$

where $w_1 \in \mathbb{R}^m$ and $W_2 \in \mathbb{R}^{m \times d}$, and $\sigma(x) = \max(0, x)$ taken element-wise. In the following, such a network is represented by: (w_1, W_2), and we note:

Lemma 1 (Addition) *Let $u = (u_1, U_2)$ and $v = (v_1, V_2)$ be two functions represented by a 2-layer neural network. Then, the function $u + v$ is represented by $\left(\begin{pmatrix} u_1 \\ v_1 \end{pmatrix}, \begin{pmatrix} U_2 \\ V_2 \end{pmatrix}\right)$.*

Lemma 2 (Scalar multiplication) *Let $u = (u_1, U_2)$ be a function represented by a 2-layer neural network. Then, the function αu is represented by $(\alpha u_1, U_2)$.*

These operations define a linear space. A two-layer network f is preserved when scaling the ith row of W_2 by $\alpha > 0$ and the ith entry of w_1 by α^{-1}. We therefore assume that each row of W_2 is scaled to have unit norm, removing any rows of W_2 that consist entirely of zero entries.[1] Now, we define an inner product as follows:

Definition 1 (An inner product between 2-layer networks) *Let k be a characteristic kernel [7] [2] over $\mathbb{R}^d$. Let u and v be two-layer networks represented by $(u_1, U_2) \in \mathbb{R}^{m_u} \times \mathbb{R}^{m_u \times d}$ and $(v_1, V_2) \in \mathbb{R}^{m_v} \times \mathbb{R}^{m_v \times d}$, respectively, where no row of U_2 or V_2 is a zero vector, and each row has unit norm. Define*

$$\langle u, v \rangle_{\mathcal{H}} := \sum_{i=1}^{m_u} \sum_{j=1}^{m_v} [u_1]_i [v_1]_j k([U_2]_{i,:}, [V_2]_{i,:}), \tag{2}$$

where $[M]_{i,:}$ denotes the ith row of M, which induces the norm $\|f\|_{\mathcal{H}} := \sqrt{\langle f, f \rangle_{\mathcal{H}}}$.

We note that k must be characteristic to guarantee the property of a norm that $\|f\|_{\mathcal{H}} = 0 \iff f = \mathbf{0}$.

This inner product inherits the structure of the linear space defined above. Using the addition (Lemma 1) and scalar multiplication (Lemma 2) operations, verifying that Equation (2) satisfies the properties of an inner product is now a basic exercise.

We may take Equation (2) as the basis of a constructive proof that two-layer networks have polynomial time computable norm. To summarize, to compute such a norm, we need to:

1. Normalize $w = (w_1, W_2)$ so W_2 has rows with unit norm, and no row is a zero vector, which takes $\mathcal{O}(md)$ time;
2. Compute $\langle w, w \rangle$ according to Equation (2), which is quadratic in m times the complexity of $k(x, x')$;
3. Compute $\sqrt{\langle w, w \rangle}$.

Therefore, assuming we allow square roots as operations, the constructed norm can be computed in a quadratic time in the cost of the evaluation of k. For example, for a Gaussian kernel $k(x, x') := \exp(-\gamma \|x - x'\|^2)$, and allowing exp as operation, the cost of the kernel evaluation is linear in the input dimension and the cost of the constructed norm is quadratic in the number of hidden units.

*The first workshop on Statistical Deep Learning for Computer Vision, in *Seoul, Korea*, 2019. Copyright by Author(s).

†This author is currently affiliated with Deepmind.

[1]The choice of vector norm is not particularly important. For concreteness we may assume it be L_1 normalized, which when considering rational weights with bounded coefficients, preserves polynomial boundedness after normalization.

[2]See main paper, Definition 1

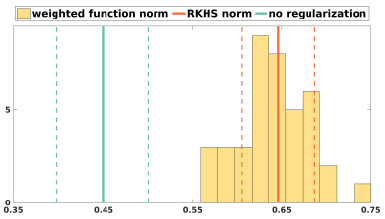

Figure 1: Histogram of accuracies with weighted function norm on the Oxford Flowers dataset over 10 trials with 4 different regularization sample sizes, compared to the mean and standard deviation of RKHS norm performance, and the mean and standard deviation of the accuracy obtained without regularization.

2. Additional experiments

2.1. Oxford Flowers classification with kernelized logistic regression

This experiment shows that the approximate L_2 norm and the RKHS norm have similar behavior on the test data. We consider the 17 classes Oxford Flower Dataset, composed of 80 images per class, and precomputed kernels that have been shown to give good performance on a classification task [5, 6]. We have taken the mean of Gaussian kernels as described in [1]. To test the effect of the regularization, we train the logistic regression on a subset of 10% of the data, and test on 20% of the samples. The remaining 70% are used as potential samples for regularization. For both regularizers, the regularization parameter is selected by a 3-fold cross validation. For the approximate norm regularization, we used a 4 different sample sizes ranging from 20% to 70% of the data. This procedure is repeated on 10 different splits of the data for a better estimate. The optimization is performed by quasi-Newton gradient descent, which is guaranteed to converge due to the convexity of the objective. Figure 1 shows the means and standard deviations of the accuracy on the test set obtained without regularization, and with regularization using the RKHS norm, along with the histogram of accuracies obtained with the weighted norm regularization with the different sample sizes and across the 10 trials. This figure demonstrates the equivalent effect of both regularizers.

The use of the weighted function norm is more useful for DNNs, where we showed that polynomial time exact norms do not exist. The next experiment shows the efficiency of the proposed regularization strategy other regularization strategies: Weight decay [4], dropout [2] and batch normalization [3].

References

[1] Peter V. Gehler and Sebastian Nowozin. On feature combination for multiclass object classification. In *International Conference on Computer Vision*, pages 221–228, 2009.

[2] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv:1207.0580*, 2012.

[3] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, 2015.

[4] Anders Krogh and John A Hertz. A simple weight decay can improve generalization. *Advances in Neural Information Processing Systems*, 4:950–957, 1995.

[5] Maria-Elena Nilsback and Andrew Zisserman. A visual vocabulary for flower classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1447–1454, 2006.

[6] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*, 2008.

[7] Bharath K Sriperumbudur, Kenji Fukumizu, and Gert RG Lanckriet. Universality, characteristic kernels and rkhs embedding of measures. *Journal of Machine Learning Research*, 12(Jul):2389–2410, 2011.