CPSC 532D — 11. UNIVERSAL APPROXIMATION

Danica J. Sutherland
University of British Columbia, Vancouver
Fall 2023

Estimation error bounds in an RKHS are relatively simple: if $\mathcal{F}$ is the RKHS with kernel $k(x, x') = \langle \varphi(x), \varphi(x') \rangle_{\mathcal{F}}$, and $\mathcal{H}_B = \{f \in \mathcal{F} : \|f\|_{\mathcal{F}} \le B\}$, we can use exactly the same Rademacher bound as before (Section 3.2 of the Rademacher notes) to see that

$$\operatorname{Rad}(\mathcal{H}_B|_{S_x}) \le \frac{B}{\sqrt{m}} \sqrt{\frac{1}{m} \sum_i \left\|\varphi(x_i)\right\|_{\mathcal{F}}^2} = \frac{B}{\sqrt{m}} \sqrt{\frac{1}{m} \sum_i k(x_i, x_i)}.$$

With a linear kernel, $k(x, x) = \|x\|^2$ and so this becomes exactly the result we had before (as it should be). But with many kernels, such as the Gaussian, $k(x, x) = 1$ and so we get simply $B/\sqrt{m}$.

Our previous style of bounds then immediately work for constrained ERM in $\mathcal{H}_B$. Our SVM algorithms also work for kernel SVMs, where now $\|w\|$ becomes $\|f\|_{\mathcal{F}}$: for instance, if we know that $\mathcal{D}$ is such that $\mathbb{E}\, k(x, x) \le C^2$ and is perfectly separable by a predictor $f^* \in \mathcal{F}$, then hard SVMs achieve

$$L_{\mathcal{D}}^{0-1}(\operatorname{sgn} \circ \hat{h}) \le \frac{1}{\sqrt{m}} \left(2C \|f^*\|_{\mathcal{F}} + \sqrt{\tfrac{1}{2} \log \tfrac{1}{\delta}} \right).$$

But what kind of approximation error can we expect?

DEFINITION 1. For a metric space $\mathcal{X}$, $C(\mathcal{X})$ denotes the Banach space of continuous functions $\mathcal{X} \to \mathbb{R}$, with norm given by $\|f\|_\infty = \sup_{x \in \mathcal{X}} |f(x)|$.

1 UNIVERSAL KERNELS

DEFINITION 2. Consider a kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, where $\mathcal{X}$ is a compact metric space, with RKHS $\mathcal{F}$ of functions $\mathcal{X} \to \mathbb{R}$. Let $C(\mathcal{X})$ denote the space of all continuous functions $\mathcal{X} \to \mathbb{R}$. k is *universal* if cF is dense in $C(\mathcal{X})$: for any continuous target function $g : \mathcal{X} \to \mathbb{R}$ and any $\varepsilon > 0$, there exists an $f \in \mathcal{F}$ such that

$$\|f - g\|_\infty = \sup_{x \in \mathcal{X}} |f(x) - g(x)| \le \varepsilon.$$

PROPOSITION 3. *Let $V, W \subset \mathcal{X}$ be disjoint compact sets, and let k be a universal kernel on $\mathcal{X}$. Then there exists an $f \in \mathcal{F}$ such that $f(x) > 0$ for all $x \in V$, and $f(x) < 0$ for all $x \in W$.*

Finite sets are compact.

This means that for any universal RKHS $\mathcal{F}$, we can shatter any finite set without "repeats," so $\operatorname{VCdim}(\operatorname{sgn} \circ \mathcal{F}) = \infty$, and $\operatorname{Rad}(\mathcal{F}|_{S_x}) = \infty$. But since the VC dimension of homogeneous linear classifiers is d, no kernel with a finite-dimensional feature map can be universal.

For any S_x, we can find some function to show $\operatorname{Rad}(\mathcal{F}|_{S_x}) > 0$, and then we can just arbitrarily scale that function up by a constant while remaining in $\mathcal{F}$ to push the Rademacher as high as we want.

For more, visit https://cs.ubc.ca/~dsuth/532D/23w1/.

Proof. Define $\text{dist}_V(x) = \min_{v \in V} \|x - v\|$, and likewise dist_W. Since the sets are compact, we can use just min instead of inf, and they'll still be well-defined continuous functions in $C(\mathcal{X})$. Since the sets are compact and disjoint, if $\text{dist}_V(x) = 0$ then $\text{dist}_W(x) > 0$, and vice versa. Thus the following g is well-defined and continuous:

$$g(x) = \frac{\text{dist}_V(x) - \text{dist}_W(x)}{\text{dist}_V(x) + \text{dist}_W(x)}.$$

But if $x \in V$, then $\mathcal{D}_V(x) = 0$, and so $g(x) = -1$ for $x \in V$, and likewise $g(x) = 1$ for $x \in W$. Thus, any $f \in \mathcal{F}$ with $\|f - g\|_\infty < 1$ will satisfy the property we want, which is called *separating compact sets*. But universality implies such an f must exist. □

proposition 4. *The Gaussian kernel* $\exp\left(-\frac{1}{2\sigma^2}\|x - x'\|^2\right)$ *is universal for any* $\sigma > 0$.

This can be proved via the Stone-Weierstrass theorem [more soon, but for full deatails see SC08, Section 4.6], or via Fourier properties [SC08, Exercise 4.12]; there are also versions that work for non-compact $\mathcal{X}$ [SFL10, with a Fourier approach].

Some infinite-dimensional kernels are not universal. One silly example is the kernel on $\mathbb{R}^d$ given by $k(x, x') = \exp(-(x_1 - x'_1)^2)$, a Gaussian kernel that only looks at the first coordinate. A less silly example is the distance kernel $k(x, x') = \|x\| + \|x'\| - \|x - x'\|$ on compact subsets of $\mathbb{R}^d$, which is an excellent kernel for certain applications but isn't "quite" universal [SSGF13, Appendix B].

Note that this isn't assuming realizability; we might have $L_{\mathcal{D}}(f_{\mathcal{D}}) > 0$.

We can also think about the Bayes predictor, e.g. for square loss the *regression function* $f_{\mathcal{D}}(x) = \mathbb{E}_{(x,y)\sim\mathcal{D}}[y \mid x]$. If $f_{\mathcal{D}} \in \mathcal{F}$, called the *well-specified* setting then constrained ERM with a large enough bound, or SRM, or similar algorithms, will have zero approximation error. Because $\mathcal{F}$ is only *dense in* $C(\mathcal{X})$ and not equal to $C(\mathcal{X})$, though, in the *misspecified* setting where $f_{\mathcal{D}} \notin \mathcal{F}$ we can always get closer and closer to $f_{\mathcal{D}}$ with increasing norm. This means there's always some approximation error as a function of the allowed norm of h. In either case, how close you can get to $f_{\mathcal{D}}$ with any finite $\|h\|_{\mathcal{F}}$ is a function of "how hard" the problem is.

2 Universal Approximation of Neural Networks

The situation is similar for neural networks, where the result is more famous and treated more mystically.

A *feedforward neural network* (or *multilayer perceptron*, MLP) is a function defined hierarchically as

$$f(x) = f^{(D)}(x) \qquad f^{(k)}(x) = \sigma_k(W_k f^{(k-1)}(x) + b_k) \qquad f^{(0)}(x) = x,$$

where $W_k \in \mathbb{R}^{d'_k \times d_{k-1}}$, $b_k \in \mathbb{R}^{d'_k}$, and $\sigma_k : \mathbb{R}^{d'_k} \to \mathbb{R}^{d_k}$; usually, $d_k = d'_k$. Typically $\sigma_D(z) = z$, while intermediate *hidden layers* use nonlinear activations. Many common choices are componentwise, such as $\text{ReLU}(z) = \max\{z, 0\}$, tanh, or $\text{sigmoid}(z) = \frac{1}{1+\exp(-z)}$. Other choices include $\text{softmax}(z) = (\exp(z_j))_j / \sum_j \exp(z_j)$, max pooling, attention operators, and so on.

We also assumed $b_k = 0$ *for simplicity. This could potentially be handled by making* σ_k *always add a constant 1 dimension to its output, though our proof also assumed* σ *had an elementwise structure.*

The solutions to A2 Q3 bound the Rademacher complexity for some such networks, with some assumptions on the σ_k, the data distribution, and various norms of the parameters. (There are [slightly] better bounds than this one; we'll talk about this a bit soon.) Like for kernels, this bound is based on the norm of the various weight matrices; it doesn't depend on the number of parameters.

It's worth noting now that neural networks are usually trained via stochastic gradient descent, but this non-convex optimization can be difficult: in general, it's NP-hard, even to optimize a single ReLU unit with square loss [GKMR21]. We'll talk more about optimization soon.

2.1 *Constructive proofs*

The following result is easy to understand, and extremely simple, but is indicative of universal approximation results in general.

THEOREM 5. *Let $g : [0,1] \to \mathbb{R}$ be ρ-Lipschitz. For any $\varepsilon > 0$, there exists a network f such that $\|f - g\|_\infty \le \varepsilon$, where the network has one hidden layer of width $N = \lceil \rho/\varepsilon \rceil$ using threshold activations $\sigma(t) = \mathbb{1}(t \ge 0)$, and a linear output unit.*

Proof. We're going to construct a piecewise-constant approximation to g. For $i \in \{0, \dots, N-1\}$, let $b_i = \frac{i\varepsilon}{\rho}$, i.e.

$$b_0 = 0, \qquad b_1 = \frac{\varepsilon}{\rho}, \qquad \cdots, \qquad b_{N-1} = \left(\left\lceil \frac{\rho}{\varepsilon} \right\rceil - 1\right)\frac{\varepsilon}{\rho} < \frac{\rho}{\varepsilon} \cdot \frac{\varepsilon}{\rho} = 1.$$

We're going to construct

$$f(x) = \begin{cases} g(0) & \text{if } 0 \le x < b_1 \\ g(b_1) & \text{if } b_1 \le x < b_2 \\ & \vdots \\ g(b_{N-1}) & \text{if } b_{N-1} \le x \le 1 \end{cases}$$

as a two-layer network. To do this, let $a_0 = g(0)$, and for $i \ge 1$ let $a_i = g(b_i) - a_{i-1}$, so that

$$\sum_{i=0}^{k} a_i = g(0) + (g(b_1) - g(0)) + (g(b_2) - (g(b_1) - g(0))) + \cdots = g(b_k).$$

Thus the desired f is just

$$f(x) = \sum_{i=0}^{N-1} a_i \, \mathbb{1}(x \ge b_i),$$

which is a network of the desired form: the first layer has a weight matrix of all ones, and a bias vector collecting the negatives of the thresholds b_i, while the second layer has weights collecting the a_i and no offset.

Now, consider any input x, and let $k = \max\{k : b_k \le x\}$. Then, since g is ρ-Lipschitz,

$$|g(x) - f(x)| \le \underbrace{|g(x) - g(b_k)|}_{\le \rho|x - b_k|} + \underbrace{|g(b_k) - f(b_k)|}_{0} + \underbrace{|f(b_k) - f(x)|}_{0} \le \rho\frac{\varepsilon}{\rho} = \varepsilon. \qquad \square$$

You could use a narrower network by depending on the total variation *of g, how much it "wiggles" up and down: if g is pretty flat in some region, there's no need to keep putting points there, you only need a new one when g changes more than ε.*

We could do a similar thing with ReLU networks, using piecewise-linear approximations rather than piecewise-constant.

Here's a similar result in $\mathbb{R}^d$:

THEOREM 6. *Let $g : [0,1]^d \to \mathbb{R}$ be continuous. For any $\varepsilon > 0$, choose $\delta > 0$ such that $\|x - x'\|_\infty \le \delta$ implies $|g(x) - g(x')| \le \varepsilon$. Then there is a three-layer ReLU network f with*

δ exists for any ε, since continuous functions on compact domains are uniformly continuous, and $\|\cdot\|_2$ and $\|\cdot\|_\infty$ are equivalent.

$\Omega\left(\frac{1}{\delta^d}\right)$ *ReLU nodes satisfying* $\int_{[0,1]^d} |f(x) - g(x)| \, \mathrm{d}x \leq 2\varepsilon$.

Proof (sketch). Approximate the continuous g by a piecewise-constant h, with pieces given by hyper-rectangles. Construct a two-layer ReLU net to check whether the input x is in each hyper-rectangle. Put those networks side-by-side as the first two layers of f, so that the second hidden layer is just an indicator vector of which hyper-rectangle x is in; use a linear readout layer to set any value on those pieces.

For more details, see Telgarsky [Tel, Theorem 2.1]. □

Notice the *curse of dimensionality*: the size of the network depends exponentially on the dimension, which for deep learning is typically *at least* hundreds, perhaps millions or more. This isn't just a proof artifact; it's necessary to approximate arbitrary continuous functions. The construction also needs really large weights, and has a really bad Lipschitz constant; it also only gives an L_1 approximation bound, not sup-norm like before.

2.2 *Non-constructive bound via Stone-Weierstrass*

We can actually get a sup-norm bound with only one hidden layer a different way, using the celebrated Stone-Weierstrass approximation theorem from analysis.

theorem 7 (Stone-Weierstrass, special case). *Let $\mathcal{X}$ be a compact metric space. Suppose $\mathcal{F}$ is a set of functions from $\mathcal{X} \to \mathbb{R}$ such that:*

- *Each $f \in \mathcal{F}$ is continuous: $\mathcal{F} \subseteq \mathrm{C}(\mathcal{X})$.*
- *For each $x \in \mathcal{X}$, there is at least one $f \in \mathcal{F}$ with $f(x) \neq 0$.*
- *For all $f, g \in \mathcal{F}$ and $\alpha \in \mathbb{R}$, we have $\alpha f + g \in \mathcal{F}$ and $fg = (x \mapsto f(x)g(x)) \in \mathcal{F}$.* ($\mathcal{F}$ *is an* algebra.)
- *For each $x \neq x' \in \mathcal{X}$, there is at least one $f \in \mathcal{F}$ with $f(x) \neq f(x')$.* ($\mathcal{F}$ separates points.)

Then $\mathcal{F}$ is dense in $\mathrm{C}(\mathcal{X})$: for any continuous function $g : \mathcal{X} \to \mathbb{R}$ and any $\varepsilon > 0$, there is some $f \in \mathcal{F}$ such that $\|f - g\|_\infty = \sup_{x \in \mathcal{X}} |f(x) - g(x)| \leq \varepsilon$.

You may have heard of the Weierstrass theorem, which shows that polynomial functions are dense in $\mathrm{C}(\mathcal{X})$; this is a generalization.

proposition 8. *The set of functions $\mathcal{F}_{\exp}$ is dense in $\mathrm{C}(\mathcal{X})$, where*

$$\mathcal{F}_{\exp} = \left\{ x \mapsto \sum_{i=1}^{m} a_i \exp(w_i \cdot x) : m \geq 1; w_1, \ldots, w_m \in \mathbb{R}^d; a_1, \ldots, a_m \in \mathbb{R} \right\}.$$

Notice that $\mathcal{F}_{\exp}$ is a set of one-hidden-layer neural networks with exponential hidden activations and *unbounded width*.

Proof. We just need to show that it satisfies the conditions of Stone-Weierstrass. The

first two are clear. For $f(x) = \sum_{i=1}^{m} a_i \exp(w_i \cdot x)$ and $g(x) = \sum_{i=1}^{m'} a'_i \exp(w'_i \cdot x)$, we have

$$\alpha f + g = \left(x \mapsto \sum_{i=1}^{m} (\alpha a_i) \exp(w_i \cdot x) + \sum_{i=1}^{m'} a'_i \exp(w'_i \cdot x) \right) \in \mathcal{F}_{\exp}$$

$$f g = \left(x \mapsto \sum_{i=1}^{m} \sum_{j=1}^{m'} a_i a'_i \exp((w_i + w'_j) \cdot x) \right) \in \mathcal{F}_{\exp}.$$

To show $\mathcal{F}_{\exp}$ separates x_1 and x_2, consider $f(x) = \exp((x_1 - x_2) \cdot x)$, so that

$$\frac{f(x_1)}{f(x_2)} = \frac{\exp\left(\|x_1\|^2 - x_2 \cdot x_1\right)}{\exp\left(x_1 \cdot x_2 - \|x_2\|^2\right)} = \exp\left(\|x_1\|^2 - 2x_1 \cdot x_2 + \|x_2\|^2\right) = \exp\left(\|x_1 - x_2\|^2\right),$$

which is one iff $x_1 = x_2$. □

(The proof that Gaussian kernels are universal is very similar.)

PROPOSITION 9 ([HSW89]). *Let* $\sigma : \mathbb{R} \to \mathbb{R}$ *be continuous with* $\lim_{z \to -\infty} \sigma(z) = 0$, $\lim_{z \to \infty} \sigma(z) = 1$. *Then* $\mathcal{F}_\sigma$ *is dense in* $C(\mathcal{X})$, *where* $\mathcal{F}_\sigma$ *is defined as*

$$\mathcal{F}_\sigma = \left\{ x \mapsto \sum_{i=1}^{m} a_i \sigma(w_i \cdot x) : m \geq 1; w_1, \ldots, w_m \in \mathbb{R}^d; a_1, \ldots, a_m \in \mathbb{R} \right\}.$$

Proof (sketch). For any continuous target g, first find an $f_0 \in \mathcal{F}_{\exp}$ such that $\|f_0 - g\|_\infty \leq \varepsilon/2$. Now, find some coefficients such that

$$\exp(z) \approx \sum_j c_j \sigma(t_j z)$$

is sufficiently accurate so that when we replace each $\exp(w_i \cdot x)$ in f_0 by $\sum_i c_i \sigma(t_i w_i \cdot x)$, we find an $f \in \mathcal{F}_\sigma$ such that $\|f - f_0\|_\infty \leq \varepsilon/2$. □

More generally, this works if σ is anything that's not a polynomial [LLPS93]. (A shallow network with fixed-degree polynomial activations is itself a polynomial of fixed degree.)

There are also a variety of other results. Maybe most important is an infinite-width construction of Barron [Bar93]; also see Section 3 of [Tel] or Section 9.3 of [Bach23].

It's also worth noting that while these results are for shallow, wide networks, universal approximation is also possible with deep, narrow networks [KL20].

3 CIRCUIT COMPLEXITY

We won't go into depth on this perspective, but it's definitely worth knowing it exists. Shalev-Shwartz and Ben-David [SSBD, Chapter 20] overview the general basic results, but the standard classic text seems to be Parberry [Par94]. There's also recent work, particularly on Transformers.

The short version:

- Two-layer networks with threshold activations can represent all functions from $\{\pm 1\}^d \to \{\pm 1\}$. Since computers always represent things as binary strings, that's pretty powerful.
- But, it takes exponential width to do that.
- But, for any Boolean function that can computed with maximal runtime T, there exists a network of size $\mathcal{O}(T^2)$ that implements that function.

4 Interpretation

"Neural networks can do anything!!"

(You don't hear "Gaussian kernels can do anything!!" as often, but it's just as true....)

These results mean that, for any (continuous) function (on a bounded domain) that we'd like to approximate, there *is* some neural net that can closely approximate that behaviour. Continuous functions also aren't a huge limit: they can closely approximate lots of noncontinuous functions too. So, there is *some* neural network that can approximate "what's the next byte a very smart human would say in response to a Unicode string of length at most 128,000 bytes." But that network is going to be *very* large (in parameter count and also weight norm). There's also a function in a Gaussian RKHS that can do that, but it has *really really* big norm.

So, does ERM in a large enough hypothesis class, or SRM, or whatever other learning algorithm, necessarily generalize? Maybe not.

Also, for neural networks ERM is NP-hard; does gradient descent approximate it well? Maybe not.

But, are these constructions with *enormous* norms indicative of the actual norm required for functions we care about? Maybe not.

One way to help answer these questions is to characterize what kinds of functions have large norms. This is mostly beyond the scope of this course, but the typical traditional scheme is based on functions in Sobolev classes; [Bach23] has a bunch of material on this. There's also recent work on, say, constructing Transformers to do some particular task, as an existence proof of approximation for *that* task (rather than universally).

References

[Bach23] Francis Bach. *Learning Theory from First Principles*. April 2023 draft.

[Bar93] Andrew R. Barron. "Universal Approximation Bounds for Superpositions of a Sigmoidal Function." *IEEE Transactions on Information Theory* 39 (3 1993), pages 930–45.

[GKMR21] Surbhi Goel, Adam Klivans, Pasin Manurangsi, and Daniel Reichman. "Tight Hardness Results for Training Depth-2 ReLU Networks." *ITCS*. 2021. arXiv: 2011.13550.

[HSW89] Kurt Hornik, Maxwell Stinchcombe, and Halber White. "Multilayer Feedforward Networks are Universal Approximators." *Neural Networks* 2 (1989), pages 359–366.

[KL20] Patrick Kidger and Terry Lyons. "Universal Approximation with Deep Narrow Networks." *COLT*. 2020. arXiv: 1905.08539.

[LLPS93] Moshe Leshno, Vladimir Ya. Lin, Allan Pinkus, and Shimon Schocken. "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function." *Neural Networks* 6.6 (1993), pages 861–867.

[Par94] Ian Parberry. *Circuit complexity and neural networks*. MIT Press, 1994.

[SC08] Ingo Steinwart and Andreas Christmann. *Support Vector Machines*. Springer, 2008.

[SFL10] Bharath K. Sriperumbudur, Kenji Fukumizu, and Gert R. G. Lanckriet. "On the relation between universality, characteristic kernels and RKHS embedding of measures." *AISTATS*. 2010. arXiv: 1003.0887.

[SSBD] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[SSGF13] Dino Sejdinovic, Bharath K. Sriperumbudur, Arthur Gretton, and Kenji Fukumizu. "Equivalence of distance-based and RKHS-based statistics in hypothesis testing." *Annals of Statistics* 41.5 (Oct. 2013), pages 2263–2291.

[Tel] Matus Telgarsky. *Deep learning theory lecture notes*. Version: 2021-10-27 v0.0-e7150f2d (alpha). 2021.