CPSC 532D — 10. UNIVERSAL APPROXIMATION

Danica J. Sutherland

University of British Columbia, Vancouver

Fall 2024

In our motivation of SRM in Chapter 9, we talked about wanting to use an $\mathcal{H}$ so big that the approximation error $\inf_{h \in \mathcal{H}} L_{\mathcal{D}}(h) - L_{\text{bayes}}$ is zero. What kinds of $\mathcal{H}$ satisfy that?

To keep things simple, we'll think about $\mathcal{Y} \subseteq \mathbb{R}$ today.

One example would be the set of all functions $\mathcal{X} \to \mathcal{Y}$. This way leads a million mathematical counterexamples of being able to do even super basic things like computing expectations, let alone being able to learn.

A milder set to target is the set of all continuous functions. If there's a continuous function achieving the Bayes error, then this immediately guarantees that the approximation error would be zero.

DEFINITION 10.1. 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 $\|f\|_\infty = \sup_{x \in \mathcal{X}} |f(x)|$.

Recall that if f and g are elements of a function space and $a \in \mathbb{R}$, we have that $af + g$ is the function mapping x to $af(x) + g(x)$. So, $\|f - g\|_\infty = \sup_{x \in \mathcal{X}} |f(x) - g(x)|$ is one possible distance metric on functions.

The following result suggests that this is a reasonable (if strict) way to calculate distances between functions.

PROPOSITION 10.2. *Suppose that $\ell(h, (x, y)) = l_y(h(x))$ for $l_y : \mathbb{R} \to \mathbb{R}$. Then $L_{\mathcal{D}}$ is $\left(\mathbb{E}_{(x,y) \sim \mathcal{D}} \|l_y\|_{\text{Lip}} \right)$-Lipschitz with respect to $\|h - g\|_\infty$.*

Proof. We have that

$$|L_{\mathcal{D}}(h) - L_{\mathcal{D}}(g)| = \left| \mathbb{E}_{(x,y) \sim \mathcal{D}} l_y(h(x)) - \mathbb{E}_{(x,y) \sim \mathcal{D}} l_y(g(x)) \right| \leq \mathbb{E}_{(x,y) \sim \mathcal{D}} \left| l_y(h(x)) - l_y(g(x)) \right|$$

$$\leq \mathbb{E}_{(x,y) \sim \mathcal{D}} \|l_y\|_{\text{Lip}} |h(x) - g(x)| \leq \left(\mathbb{E}_{(x,y) \sim \mathcal{D}} \|l_y\|_{\text{Lip}} \right) \|h - g\|_\infty. \quad \square$$

10.1 DENSENESS

Even if the "target function" isn't continuous, the approximation error could still be zero.

EXAMPLE 10.3. Consider $\mathcal{X} = \mathbb{R}$ and the true labels being determined by the discontinuous function $y = \mathbb{1}(x > 0)$. Although this function isn't in $C(\mathcal{X})$, you can get

---

For more, visit https://cs.ubc.ca/~dsuth/532D/24w1/.

arbitrarily close to it, e.g. by taking the continuous functions

$$f_\sigma(x) = \begin{cases} 0 & \text{if } x \le 0 \\ x/\sigma & \text{if } 0 \le x \le \sigma \\ 1 & \text{if } x \ge \sigma. \end{cases}$$

The 0-1 loss here is

$$L_\mathcal{D}(f_\sigma) = \Pr\big(x \in (0, \sigma)\big)\, \mathbb{E}\Big[1 - \frac{x}{\sigma} \mid x \in (0, \sigma)\Big] < \Pr\big(x \in (0, \sigma)\big).$$

As $\sigma \to 0$, we have $L_\mathcal{D}(f_\sigma) \to 0$ regardless of $\mathcal{D}$. Thus, $\inf_{h \in C(\mathcal{X})} L_\mathcal{D}(h) = 0$, even though there is no $h \in C(\mathcal{X})$ with $L_\mathcal{D}(h) = 0$. Therefore the approximation error, in this case, is zero.

$C(\mathcal{X})$ can approximate many interesting function classes. We can frame this with the following definition from metric topology:

Definition 10.4. Let $\mathcal{G} \subseteq \mathcal{F}$ for some metric space $\mathcal{F}$. We say that $\mathcal{G}$ *is dense in* $\mathcal{F}$ with respect to the metric ρ if, for every $f \in \mathcal{F}$, $\inf_{g \in \mathcal{G}} \rho(g, f) = 0$.

That is, for every point in $f \in \mathcal{F}$ that isn't in $\mathcal{G}$, you need to be able to get *arbitrarily close* to f with points in $\mathcal{G}$.

A canonical example is that the set of rational numbers is dense in the set of real numbers.

Proposition 10.5. *Suppose that* $\mathcal{H}$ *is dense in* $\mathcal{F}$ *with respect to* $\|\cdot\|_\infty$, *and use loss* $\ell(h, (x, y)) = l_y(h(x))$ *with finite* $\mathbb{E}_{(x,y) \sim \mathcal{D}} \|l_y\|_{\text{Lip}}$. *Then* $\inf_{h \in \mathcal{H}} L_\mathcal{D}(h) = \inf_{f \in \mathcal{F}} L_\mathcal{D}(f)$.

Proof. Let $M = \mathbb{E}_{(x,y) \sim \mathcal{D}} \|l_y\|_{\text{Lip}} < \infty$. Choose $(f_1, f_2, \dots)$ to be a sequence in $\mathcal{F}$ such that $L_\mathcal{D}(f_i) \to \inf_{f \in \mathcal{F}} L_\mathcal{D}(f)$. For each f_i, choose a $g_i \in \mathcal{G}$ such that $\|f_i - g_i\|_\infty \le \frac{1}{i}$, which is possible because $\mathcal{G}$ is dense in $\mathcal{F}$. Then, by Proposition 10.2, $|L_\mathcal{D}(g_i) - L_\mathcal{D}(f_i)| \le M \|g_i - f_i\|_\infty \le \frac{M}{i} \to 0$, and thus $(L_\mathcal{D}(g_i))$ converges to the same point as $(L_\mathcal{D}(f_i))$. $\square$

10.2 UNIVERSAL APPROXIMATORS

There are many variants of universality [see e.g. SFL10]; this is a reasonable baseline.

Definition 10.6. We call a hypothesis class $\mathcal{H}$ of functions $\mathcal{X} \to \mathbb{R}$ *universal* if $\mathcal{H} \cap C(\mathcal{X})$ is dense in $C(\mathcal{X})$ with respect to $\|\cdot\|_\infty$.

The following property is known as *separating compact sets*. It establishes that thresholding functions in a universal hypothesis class can shatter any set, so that $\text{VCdim}(\text{sgn} \circ \mathcal{H}) = \infty$. It also implies that the Rademacher complexity is infinite.

Finite sets are compact. **Proposition 10.7.** *Let* $V, W \subset \mathcal{X}$ *be disjoint compact sets, and let* $\mathcal{H}$ *be universal. Choose any* $a \ge 0$. *Then there exists an* $h \in \mathcal{H}$ *such that* $h(x) > a$ *for all* $x \in V$, *and* $h(x) < -a$ *for all* $x \in W$.

Proof. Define $\rho_V(x) = \min_{v \in V} \|x - v\|$, and likewise ρ_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 $\rho_V(x) = 0$ then

$\rho_W(x) > 0$, and vice versa. Thus the following g is well-defined and continuous:

$$g(x) = 2a \frac{\rho_V(x) - \rho_W(x)}{\rho_V(x) + \rho_W(x)}.$$

If $x \in V$, then $\mathcal{D}_V(x) = 0$, and so $g(x) = -2a$ for $x \in V$. Likewise, $g(x) = 2a$ for $x \in W$. Thus, any $h \in \mathcal{H}$ with $\|h - g\|_\infty < a$ will satisfy the property we want. Since g is continuous and $\mathcal{H}$ is dense in $C(\mathcal{X})$, such an h must exist. $\qquad\square$

This result implies that, at least for binary classifiers, it's impossible to PAC-learn a universal $\mathcal{H}$. Depending on the $\mathcal{H}$, though, we may be able to nonuniformly learn it with SRM or similar algorithms. If we use a decomposition $\mathcal{H} = \mathcal{H}_1 \cup \mathcal{H}_2 \cup \cdots$ for $\mathcal{H}_1 \subset \mathcal{H}_2 \subset \cdots$, then even though the approximation error in all of $\mathcal{H}$ is zero, the approximation error in $\mathcal{H}_k$ might not be. As we consider $\mathcal{H}_k$ for increasing k, we trade off higher estimation error for lower approximation error. When $\mathcal{H}$ is universal, there might be some $\mathcal{H}_k$ where we can achieve zero approximation error (if there's some $h \in \mathcal{H}$ achieving the minimal loss, also called the *well-specified* setting). We might, though, only have the approximation error of $\mathcal{H}_k$ going to zero as k increases, called a *misspecified setting*; this would be true e.g. in Example 10.3 with $\mathcal{H}_k = \{f : \|f\|_{\mathrm{Lip}} \le k\}$.

Well-specified doesn't imply realizable; you might have $\inf_{h \in \mathcal{H}} \mathrm{L}_\mathcal{D}(h) > 0$.

10.3 UNIVERSAL APPROXIMATION OF NEURAL NETWORKS

As you may have heard before (probably invoked in somewhat mystical ways), classes of neural networks are universal.

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 $\mathrm{ReLU}(z) = \max\{z, 0\}$, tanh, or $\mathrm{sigmoid}(z) = \frac{1}{1+\exp(-z)}$. Other choices include $\mathrm{softmax}(z) = (\exp(z_j))_j / \sum_j \exp(z_j)$, max pooling, attention operators, and so on.

On A3 Q3, you bounded the Rademacher complexity for some such networks, with some assumptions on σ_k, $\mathcal{D}$, bounds on W_k, and that $b_k = 0$. (There are [slightly] better bounds than this one; we'll talk about this a bit soon.) Your bound didn't explicitly depend on the number of parameters, just on their norms.

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.

10.3.1 *Constructive proofs*

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

THEOREM 10.8. *Let $g : [0, 1] \to \mathbb{R}$ be M-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 $N = \lceil M/\varepsilon \rceil$*

using threshold activations $\sigma(t) = \mathbb{1}(t \geq 0)$, and a linear output unit.

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

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

We're going to construct

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

as a two-layer network. To do this, let $a_0 = g(0)$, and for $i \geq 1$ 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

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

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 $k = \max\{k : b_k \leq x\}$. Then, since g is M-Lipschitz,

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

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 10.9. *Let $g : [0,1]^d \to \mathbb{R}$ be continuous. For any $\varepsilon > 0$, choose $\delta > 0$ such that $\|x - x'\|_\infty \leq \delta$ implies $|g(x) - g(x')| \leq \varepsilon$. Then there is a three-layer ReLU network f with $\Omega\left(\frac{1}{\delta^d}\right)$ ReLU nodes satisfying $\int_{[0,1]^d} |f(x) - g(x)|\, dx \leq 2\varepsilon$.*

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 details, see Theorem 2.1 of Telgarsky [Tel21]. $\qquad \square$

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.

10.3.2 *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 10.10 (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: $\mathcal{F} \subseteq C(\mathcal{X})$.*

- *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 $C(\mathcal{X})$ with respect to $\|\cdot\|_\infty$.

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

Proposition 10.11. *The set of functions $\mathcal{F}_{\exp}$ is dense in $C(\mathcal{X})$, 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

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

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

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 $x_1 = x_2$. $\qquad\square$

Proposition 10.12 ([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 $\|f_0 - g\|_\infty \le \varepsilon/2$. 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 $\|f - f_0\|_\infty \le \varepsilon/2$. □

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.) These are for shallow, wide networks, but if you use a deep, narrow network you can get away even with polynomial activations [KL20].

There are also a variety of other results. Maybe most important is an infinite-width construction of Barron [Bar93]; also see Section 3 of [Tel21] or Section 9.3 of [Bach24].

10.4 CIRCUIT COMPLEXITY

We won't go into depth on this perspective, but it's definitely worth knowing it exists. Shalev-Shwartz and Ben-David [SSBD14, 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.

10.5 INTERPRETATION

"Neural networks can do anything!!"

(You don't hear "decision trees 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, as in Example 10.3. So, there is *some* neural network that can approximate "what's the next bit in the response of a very smart human 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 *really really big* decision tree that can do that.

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.

6

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; [Bach24] 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

[Bach24]   Francis Bach. *Learning Theory from First Principles*. Draft version. August 2024.

[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.

[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.

[SSBD14]   Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[Tel21]   Matus Telgarsky. *Deep learning theory lecture notes*. October 2021.