ML Theory Lecture 8

Matus Telgarsky

Today we'll finish the "succinct" and representation topics by showing how to multiply with ReLU networks, and with some comments on generative models.

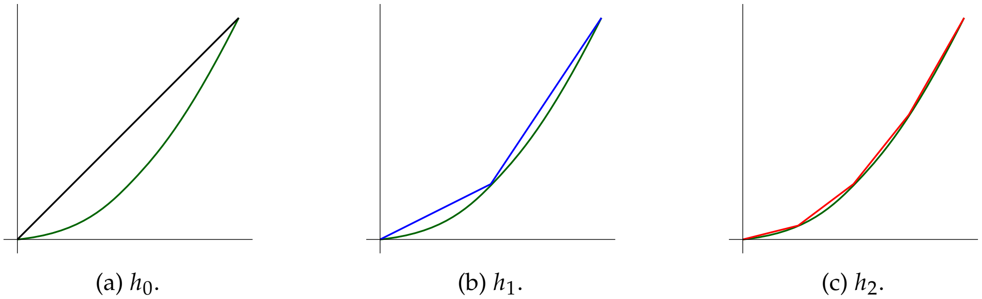
1 Squaring with neural nets

This section will show how to implement $x \mapsto x^2$; this beautiful construction is due to Yarotsky (2016).

Let's try to brute force the simplest possible idea: let's consider a sequence $(h_0, h_1, \ldots)$ of piecewise affine interpolants of x^2 along $[0,1]$, where $h_i(x) = x^2$ at the points

$$S_i := \left(\frac{0}{2^i}, \frac{1}{2^i}, \ldots, \frac{2^i}{2^i}\right).$$

That is to say, $h_i(x) = x^2$ for $x \in S_i$, and otherwise h_i interpolates between those points.



(a) h_0.

(b) h_1.

(c) h_2.

For now, this construction is not so nice for ReLU approximation: it seems as though we'll need 2^i ReLUs to approximate h_i; it turns out we'll need only $\mathcal{O}(i)$!

Let's try to understand what is happening inductively. The base case $i = 0$ is easy: $h_0(x) = x$.

Consider h_{i+1} and h_i for $i \geq 0$. Since h_i is correct on S_i and $S_i \subseteq S_{i+1}$, then $h_{i+1}(x) = h_i(x)$ for $x \in S_i$.

It remains to consider $x \in S_{i+1} \setminus S_i$, meaning $x = (2j+1)/2^{i+1}$ for some $j \in \{0, \ldots, 2^i - 1\}$. For convenience, write $\epsilon = 1/2^{i+1}$; then

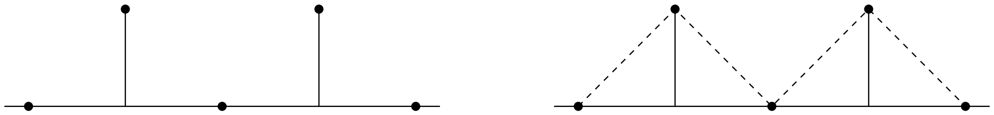
$$h_i(x) - h_{i+1}(x) = \frac{1}{2}\left((x-\epsilon)^2 + (x+\epsilon)^2\right) - x^2 = \left(x^2 + \epsilon^2\right) - x^2 = \epsilon^2.$$

The key thing here is the gap is a constant!

Putting the two cases together, for $x \in S_{i+1}$,

$$h_{i+1}(x) = h_i(x) - \underbrace{\frac{1}{4^{i+1}} \mathbb{1}[x \in S_{i+1} \setminus S_i]}_{\star}.$$

Now let's consider the behavior of the interpolation, so we can discuss $x \in [0,1]$. The difference $h_{i+1}(x)$ must be piecewise affine and indeed it must be the linear interpolation of the term $\star$ above, as depicted in the following plots.



(a) ⋆.

(b) Piecewise affine interpolation.

This interpolation is a familiar function: for $x \in [0,1]$,

$$h_{i+1} = h_i - \frac{\Delta^{i+1}}{4^{i+1}}.$$

Since $h_0(x) = x$,

$$h_i(x) = h_0(x) + \sum_{j=0}^{i-1} \Big(h_{j+1}(x) - h_j(x)\Big) = x - \sum_{j=1}^{i} \frac{\Delta^j(x)}{4^j}.$$

Since all terms can be written with ReLUs, this gives the following.

Theorem 1.1 (Yarotsky (2016)). *The functions $(h_i)_{i\geq 0}$ defined above satisfy the following properties.*

1. *h_i is the piecewise-affine interpolation of x^2 along $[0,1]$ with interpolation points S_i.*
2. $\sup_{x\in[0,1]} |h_i(x) - x^2| \leq 4^{-i-1}$.
3. *h_i can be written as a ReLU network consisting of $2i$ layers and $5i$ nodes.*
4. *Any ReLU network f with $\leq L$ layers and $\leq N$ nodes satisfies*

$$\int_{[0,1]} (f(x) - x^2)^2 \,\mathrm{d}x \geq \frac{1}{5760(2N/L)^{4L}}.$$

Proof. 1. This was handled above.

2. Fix i, and set $\tau := 2^{-i}$, meaning τ is the distance between interpolation points. The error between x^2 and h_i is thus bounded above by

$$\sup_{x\in[0,1-\tau]} \sup_{z\in[0,\tau]} \frac{\tau - z}{\tau}\left(x^2\right) + \frac{z}{\tau}(x+\tau)^2 - (x+z)^2 = \frac{1}{\tau} \sup_{x\in[0,1-\tau]} \sup_{z\in[0,\tau]} 2xz\tau + z\tau^2 - 2xz\tau - \tau z^2$$
$$= \frac{1}{4\tau} \sup_{x\in[0,1-\tau]} \frac{\tau^3}{4} = \frac{\tau^2}{4} = 4^{-i-1}.$$

3. The relu network is as follows. It contains Δ^i, but also an "accumulation line" where it first passes forwards x, and thereafter subtracts off each $\Delta^j/4^j$ as it is computed. Δ^i requires $2i$ layers and $3i$ nodes, and the accumulation line is a single chain of 1 node per each of $2i$ layers.

4. By a bound from last lecture, $N_A(f) \leq (2N/L)^L$. Using a symbolic package to differentiate, for any interval $[a,b]$,

$$\min_{(c,d)} \int_{[a,b]} (x^2 - (cx+d))^2 \,\mathrm{d}x = \frac{(b-a)^5}{180}.$$

Let S index the subintervals of length at least $1/(2N)$ with $N := N_A(f)$, and restrict attention to $[0,1]$. Then

$$\sum_{[a,b]\in S} (b-a) = 1 - \sum_{[a,b]\notin S} (b-a) \geq 1 - N/(2N) = 1/2.$$

Consequently,

$$\begin{aligned}\int_{[0,1]}(x^2-f(x))^2\,\mathrm{d}x &= \sum_{[a,b]\in P_A(f)}\int_{[a,b]\cap[0,1]}(x^2-f(x))^2\,\mathrm{d}x\\ &\geq \sum_{[a,b]\in S}\frac{(b-a)^5}{180}\\ &\geq \sum_{[a,b]\in S}\frac{(b-a)}{2880N^4}\geq\frac{1}{5760N^4}.\end{aligned}$$

□

Remark 1.2. From squaring we can get many other things. First of all,

$$(x,y)\mapsto xy=\frac{1}{2}\left((x+y)^2-x^2-y^2\right).$$

Once we have multiplication, we get polynomials, which give smooth functions essentially via Taylor expansion (Yarotsky, 2016). We can also approximate division and thus rational functions (Telgarsky, 2017). ◇

2 Probability distributions

We also discussed how to represent various probability distributions. Here is a brief summary: *[highly abridged]*

- One classical model is to approximate a density with $1/n\sum_{i=1}^n k(x_i,x)$, where $k(\cdot,\cdot)$, is a density kernel, for instance a Gaussian. By our representation results for RBF SVMs, this can approximate any continuous density. Note that KDE also gives an algorithm, albeit a simple one: sample $(x_i)_{i=1}^n$ from a continuous density, and plug them into the estimator above. Note that this estimator suffers a bad curse of dimension.
- We mentioned Gaussian Mixtures, which have some similarities to KDE, but there a beneficial scenarios where they are more succinct.
- We also discussed generative networks. These operate by first sampling $x\sim\mu$, where μ is some efficiently sampleable distribution, and thereafter outputting $f(x)$, where f is a neural network. In the optimal transport literature, this is written $f\#\mu$, if random variable Y has distribution $f\#\mu$ and X has distribution μ, then $\Pr[Y\in S]=\Pr[X\in f^{-1}(S)]$.

  I mentioned that in the univariate case, we can prove we can represent some distributions in this way using the inverse CDF sampling method. In more general cases, there are similar things in the literature on optimal transport, but typically requiring equal input and output dimensions (Villani, 2003). For differing dimensions, you have to make space-filling curves, as in a recent paper by Bolton Bailey and I.

References

Matus Telgarsky. Neural networks and rational functions. In *ICML*, 2017.

Cedric Villani. *Topics in Optimal Transportation*, volume 58 of *Graduate Studies in Mathematics*. American Mathematical Society, 2003.

Dmitry Yarotsky. Error bounds for approximations with deep relu networks. 2016. `arXiv:1610.01145 [cs.LG]`.