The Universal Approximation Theorem

1/12/23

I've read in the fast.ai book about the *universal approximation theorem*. It's described vaguely, but since I am a mathematician by training, I'm going to try to do the following in this post: first, I'll guess at the precise statement of the theorem. Then, I'll look the precise statement up. And finally, I'll try to extract some lessons from the exercise.

So, here's the guess: Let $g : \mathbb{R} \to \mathbb{R}$ be the function

$$g(x) = \max(x, 0).$$

This is the ReLU/rectified linear unit function. Given any other continuous function $f : \mathbb{R}^n \to \mathbb{R}$, any $\epsilon > 0$, and any compact subset $S \subset \mathbb{R}^n$, there exist constants $\{c_i^j\}_{i=1,j=1}^{i=n,j=m}$, $\{b_i\}_{i=1}^n$, $\{d_j\}_{j=1}^m$, and w such that

$$\left| f(x^1, \ldots, x^n) - w - \sum_{j=1}^m d_j g\left(b_i + \sum_{i=1}^n c_i^j x^i \right) \right| < \epsilon, \quad \forall x \in S.$$

Taking a look here, we see that this version of the theorem is called the "arbitrary-width" version of the theorem. The only thing which is different between the above statement and the reference in Wikipedia is that Wikipedia informs us that the theorem applies for any continuous function which is not polynomial in place of g (the ReLU function is not polynomial because it is not identically zero but has infinitely many zeroes). All the other differences are a matter of differences in notation but not content; the biggest such difference is that f is allowed on the Wikipedia page to have codomain $\mathbb{R}^k$ for some k; but this follows from my case by the triangle inequality.

On the Wikipedia page, there are other versions of the theorem. The most interesting one to me is the one which allows one to fix m (the "width" of the network) to be bounded by $n + m + 2$ by allowing arbitrarily many layers in the network, i.e. by combining the various $d_j g$ terms as the inputs to more copies of g. This represents that tradeoff between depth and width that I've learned about. This works if g is any non-affine function. Apparently, and this is really cool to me, it's possible to determine the minimum required depth for a fixed f and ϵ.

Finally, there is a version of the theorem that, by choosing a suitable candidate for g, one can put a global bound on both the depth and width of the network! I wonder if this choice of g gives significant performance improvements in practice...