E0 306: Deep Learning: Theory and Practice 12^{th} February 2019

Scribe Notes for Lecture 10
Expressive power of Neural Networks

Lecturer: Amit Deshpande *Scribe*: Janaky Murthy

In this lecture we will address the following questions:

1. What are the set of functions computable by a given NN architecture?
2. Given a family of functions $\mathcal{F}$, what is the size of the NN that is needed to compute it?

The *size* s of a neural network is the number of neurons it contains. We will refer to a feed forward network with an input layer, $l-1$ hidden layers and one output layer as l-NN.

Example 0.1 ***Toy Case 1:*** **2-*NN for a* n-*variate boolean function.*** *Given a boolean function* $f : \{0,1\}^n \mapsto \{0,1\}$*, can it be computed by a neural network having an input layer, 1-hidden layer and one output layer? If yes, determine the size of the network.*

Yes, we can construct such a NN for any arbitary boolean function f! For $i \in [1, 2^n]$, Let f_i denote the boolean function whose truth table has 1 at the i-th entry and 0 at other entries. Clearly $f_1, \ldots, f_{2^n}$ form a basis for the set of all n-variate boolean functions. In other words any n-variate boolean function f can be written as $f = a_1 f_1 + \ldots + a_{2^n} f_n$ where each $a_i \in \{0,1\}$. Hence it is sufficient to show that a neuron can compute f_i. The points in $\{0,1\}^n$ can be viewed as corners of an n-dimensional boolean hypercube (Figure 1. Each of the basis function f_i corresponds to assigning 0 to one of the corners and 1 to all other corners. Clearly such an assignment is linearly separable and hence can be computed by a neuron having threshold activation function.

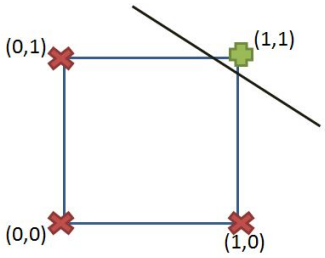

Figure 1: The basis functions are linearly seperable in 2 dimensional hypercube

The above example gives an upper bound on the size of neurons needed to compute any arbitary boolean function. But are $O(2^n)$ neurons necessary? In other words, what is the lower bound? Is there an explicit boolean function that needs $\Omega(2^n)$ size NN? These are even harder questions. We state few natural and intersting questions:

- Can we improve the 2^n bound by restricting to some interesting subset of boolean functions?
- What can we say about the bounds on the weights on the edges?
- Can we decrease the size by increasing the depth?
- Does increasing depth strictly increase the expressive power of the neural network?
- What are the set of functions that can expressed as a linear combination of some f_i's where the f_i's can be computed by a l-NN?

So far we have been looking at expressive power of NNs when $\mathcal{F}$ is the set of all boolean functions. But NNs can compute real valued functions. In the next section we'll see that NNs can *approximate* most of the real valued functions.

1 Universal approximation theorem

We first state the generic version of the Universal approximation theorem.

Theorem 1.1 *Generic-Universal Approx thm: Let $D \subseteq \mathbb{R}^n$ and $\mathcal{F}$ be some family of real valued functions $f : D \mapsto \mathbb{R}$. Then for all $f \in \mathcal{F}$ there is a neural network g that approximates f, i.e given any $\epsilon > 0$ there's a neural network g*

$$\forall f \in \mathcal{F}, x \in D : |f(x) - g(x)| < \epsilon \quad .$$

There are various versions of this theorem ([1], [2]) where it is proved by imposing conditions on the family $\mathcal{F}$, the domain D, the activation functions used in the neurons etc.

1.1 Barron's Theorem

Fourier Transform: Let $f : \mathbb{R}^n \mapsto R$ be a real valued function. The Fourier transform of f denoted by $F : \mathbb{R}^n \mapsto R$ is defined as:

$$F(w) = \int_x f(x) e^{-i<w,x>} dx$$

Smoothness Conditions: In our theorem we assume f satisfies the following smoothness conditions.

$$\int_w ||w|| F(w) dw \leq C$$

where C is some constant.

Theorem 1.2 *Let $x \in \mathbb{R}^n$ be picked from a distribution μ. Any continuous function $f : \mathbb{R}^n \mapsto \mathbb{R}$ that satisfies the smoothness conditions can be approximated by a 2-NN containing $O(\frac{1}{\epsilon^2})$ hidden units, i.e*

$$\mathbb{E}_{x \sim \mu}(||f(x) - g(x)||^2 \leq \epsilon$$

Proof Sketch. The proof idea is as follows: Write $f \approx f_1 + \ldots + f_k$ and show that each f_i can be approximately computed by a neuron. Write the width of the hidden layer, which is k in terms of the approximation factor.
Step 1: Write f as an infinite sum using fourier transform. Let $F(w)$ be the fourier transform of $f(x)$.

$$f(x) = \int_w F(w) e^{i<w,x>} dw$$

$F(w)$ can be written as $|F(w)|e^{i\theta_w}$. Substituting this in the above equation, we get:

$$\begin{aligned} f(x) &= \int_w |F(w)| e^{i(<w,x>+\theta_b)} dw \\ \implies f(x) &= \int_w |F(w)| cos(<w,x>+\theta_w) dw + i \cdot \int_w |F(w)| sin(<w,x>+\theta_w) dw. \end{aligned}$$

Since f is a real valued function, the second integral is 0.

$$\begin{aligned} f(x) &= \int_w |F(w)| cos(<w,x>+\theta_w) dw \\ \implies f(x) &= \int_w |F(w)| \frac{||w||}{C} cos(<w,x>+\theta_w) \frac{C}{||w||} dw \end{aligned}$$

Informal: Observe that the smoothness condition says that $f(x) = \int_w |F(w)| \frac{||w||}{C} dw \leq 1$ implying f is a convex combination of $cos(<w,x>+\theta_w) \frac{C}{||w||}$, i.e

$$f(x) = \sum_w \alpha_w cos(<w,x>+\theta_w) \frac{C}{||w||} \tag{1}$$

and $\sum_w \alpha_w = 1$. Equation 1 implies that f can be represented by a 2-NN with infinite number of cosine non-linearity in the hidden layer.

Step 2: Approximate the infinite sum by a finite sum. We now show that f can be well approximated by a finite number of terms in the RHS of Equation 1. Notice that the α_w's induce a natural probability distribution on w's. Pick r i.i.d samples $w_1, \ldots, w_r$ where each w_i is picked with probability α_{w_i}. Let g be the approximation of f computed from these finite samples. Our goal is to bound the $\mathbb{E}_{x\sim\mu}[\ ||f(x) - g(x)||\ ]$. It is sufficient to bound $||f - g||$. We use the following lemma due to [4] to bound $|f - g|$.

Lemma 1.3 *If f is in the closure of a convex hull of a set G in a Hilbert Space, with $||g_i|| < b$ for each $g_i \in G$ then for every $r \geq 1$ and every $k > b^2 - ||f||^2$, there exists $g_1, \ldots, g_n \in G$ such that*

$$||f - g|| \leq \frac{k}{r}$$

where g is in the convex hull of those n points.

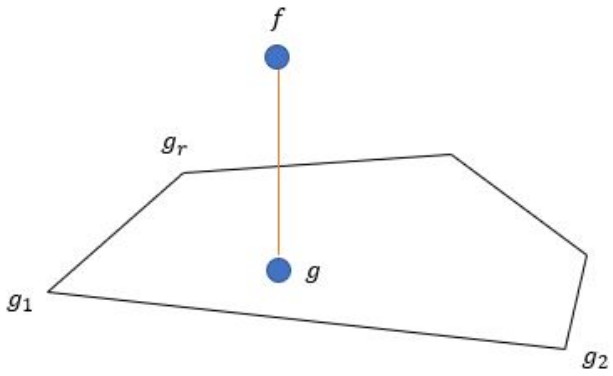

Figure 2: A point in the closure of the convex hull can be well approximated by a point in the convex hull

It turns out that in the proof of Lemma 1.3 (which uses law of large numbers) the g_i's are chosen exactly the same way we have described above. Figure 2 gives a pictorial interpretation of Lemma 1.3.

Step 3: Approximate cosine by some non linearity of your choice. After Step 2 we have that f is approximated as a finite sum of cosines. Now the task boils down to approximating cosine by the non linearity we are interested in, say sigmoid or ReLu. First we approximate approximate cosine as sum of step function (Figure 3) and then each step function by the desired nonlinearity (say sigmoid) (Figure 4) (The figures are taken from the lecture notes [3]

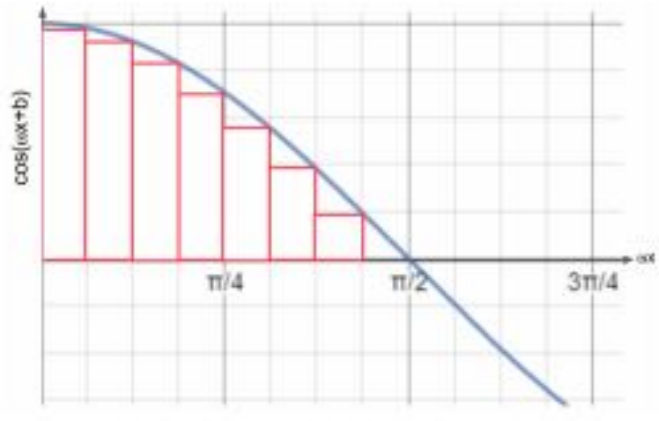

Figure 3: Approximating $cos(x)$ by step functions

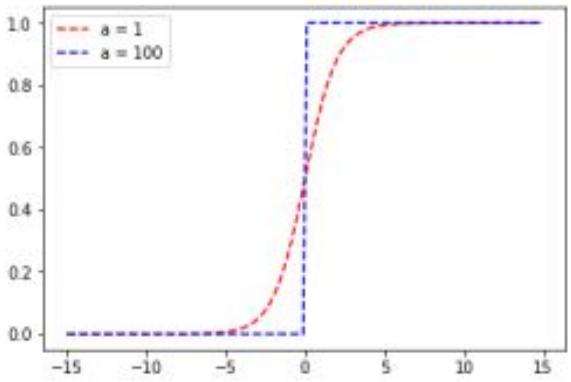

Figure 4: Approximating step function by sigmoid

References

[1] Andrew R Barron. "Universal approximation bounds for superpositions of a sigmoidal function". In: *IEEE Transactions on Information theory* 39.3 (1993), pp. 930–945.

[2] George Cybenko. "Approximation by superpositions of a sigmoidal function". In: *Mathematics of control, signals and systems* 2.4 (1989), pp. 303–314.

[3] Sanjeev Arora Neelesh Kumar Yuting Wang. *Lecture Notes: Expressiveness of Neural Networks*. URL: `https://www.cs.princeton.edu/courses/archive/fall18/cos597G/lecnotes/lec2.pdf`. 2018.

[4] Gilles Pisier. "Remarques sur un résultat non publié de B. Maurey". In: *Séminaire Analyse fonctionnelle (dit" Maurey-Schwartz")* (1980), pp. 1–12.