10-707: Advanced Deep Learning **Spring 2020**

Lecture 2: January 15

Lecturer: Andrej Risteski *Scribes: Amrit Singhal, Jin Shang, Yijie Sun, Zihao Ding*

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

2.1 Three pillars of supervised learning

Supervised learning is to minimize a training loss l over a class of predictors $\mathcal{F}$:

$$\hat{f} = \arg\min_{f \in \mathcal{F}} \sum_{(x,y):\text{training examples}} l(f(x), y) \tag{2.1}$$

Three pillars of supervised learning:

1. Representational power (How expressive is the class $\mathcal{F}$)
2. Optimization (How to minimize the training loss efficiently)
3. Generalization (How does $\hat{f}$ perform on unseen samples)

2.2 Universal expressivity of neural networks

Expressivity: how complicated a function could be that a neural network can represent given a specific structure and certain amount of volumn.

In this section, the ultimate theorem to be proved is that **neural networks are universal approximators**:

Theorem 2.1 *Given any Lipschitz f: $\mathbb{R}^d \to \mathbb{R}$, a shallow (3-layer) neural network with $(\frac{1}{\epsilon})^d$ neurons can approximate it to within ϵ error.*

This can be further quantified to:

Theorem 2.2 *For any L-Lipschitz function f: $[0,1]^d \to \mathbb{R}$, there is a 3-layer neural network $\hat{f}$ with $O(d(\frac{L}{\epsilon})^d)$ ReLU neurons, s.t.*

$$\int_{[0,1]^d} \left| f(x) - \hat{f}(x) \right| dx \leq \epsilon$$

Proof: The proof can be divided into 3 parts:

1. Query the values of f on a fine grid
2. Approximate f as linear combination of "queries"
3. Approximate the indicators using ReLUs

- **Part I**

  If a function f: $[0,1]^d \to \mathbb{R}$ is L-Lipschitz:

$$\forall x, y \in [0,1]^d, |f(y) - f(x_i)| \leq L \, ||x - y||_\infty \tag{2.2}$$

  Consider $P = (C_1, C_2, \cdots, C_N)$ a partition of $[0,1]^d$ into cells of side lengths at most δ.
  If we have a set of obervations of L-Lipschitz function f: $[(x_1, f(x_1)), (x_2, f(x_2)), \cdots, (x_N, f(x_N))]$, $x_i \in C_i$, by Lipschitzness:

$$\begin{aligned} \forall i, y \in C_i : |f(y) - f(x_i)| &\leq L \, |y - x_i| \\ &\leq L\delta \end{aligned} \tag{2.3}$$

  Therefore, we build a grid on values of f using the observations.

- **Part II**

  In this part we try to prove that such a grid can be represented by a piecewise linear function $g(x)$:

$$g(x) = \sum_{i=1}^{N} 1_{x \in C_i} f(x_i)$$

Lemma 2.3 *$g(x) = \sum_{i=1}^{N} 1_{x \in C_i} f(x_i)$ satisfies $\sup_{x \in [0,1]^d} |f(x) - g(x)| \leq L\delta$.*

Proof: Let $x \in C_i$. Then $1_{x \in C_i} = 1$ and $1_{x \in C_j} = 0$ for $j \neq i$.
So, $g(x) = f(x_i)$.
By equation 2.3, $|f(x) - g(x)| = |f(x) - f(x_i)| \leq L\delta$. ■

- **Part III**

  In this part, we try to construct $g(x)$ using a 3-layer neural network $\hat{f}(x)$ with ReLU activation, so that based on Lemma 2.3 we can finally prove Theorem 2.2.

  Because $g(x)$ is a piecewise linear function (indicators) of cells C_i, we need to prove the indicator for each cell can be represented by a 2-layer network:

Lemma 2.4 *Let $C \subseteq \mathbb{R}^d$ be a cell, $C = x : x \in [l_i, r_i], i \in d$. Then, there exists a 2-layer network $\tilde{\tilde{h}}(x)$ of size O(d) and ReLU activation, s.t.*

$$\int_{x \in [0,1]^d} \left| \tilde{\tilde{h}}(x) - 1(x \in [l_i, r_i], i \in d) \right| dx \to 0$$

Proof: For $\tau \geq 0, x \in R$ and σ for ReLU function:

$$|1(x \geq 0) - (\sigma(\tau x) - \sigma(\tau x - 1))| = \begin{cases} \leq 1, \; 0 \leq x \leq \dfrac{1}{\tau} \\ 0, \text{ otherwise} \end{cases} \tag{2.4}$$

$$1(x \in [l_i, r_i], i \in d) = 1(\sum_{i=1}^{d}(1(x_i \geq l_i) + 1(x_i \leq r_i)) \geq 2d) \tag{2.5}$$

$$\tilde{1}(x) = \sigma(\tau x) - \sigma(\tau x - 1) \tag{2.6}$$

$$h(x) = 1(\sum_{i=1}^{d}(1(x_i \geq l_i) + 1(x_i \leq r_i)) \geq 2d) \tag{2.7}$$

$$\tilde{h}(x) = 1(\sum_{i=1}^{d}(\tilde{1}(x_i \geq l_i) + \tilde{1}(x_i \leq r_i)) \geq 2d) \tag{2.8}$$

$$\tilde{\tilde{h}}(x) = \tilde{1}(\sum_{i=1}^{d}(\tilde{1}(x_i \geq l_i) + \tilde{1}(x_i \leq r_i)) \geq 2d) \tag{2.9}$$

According to 2.4:

$$\begin{aligned}\int_{x\in[0,1]^d} \left|\tilde{h}(x) - h(x)\right| dx &\leq \int_{x\in[0,1]^d} \left|1(\sum_{i=1}^{d}(1(x_i \geq l_i) + 1(x_i \leq r_i)) \neq \sum_{i=1}^{d}(\tilde{1}(x_i \geq l_i) + \tilde{1}(x_i \leq r_i)))\right| dx \\ &\leq \int_{x\in[0,1]^d} (\sum_{i=1}^{d} 1(1(x_i \geq l_i) \neq \tilde{1}(x_i \geq l_i)) + \sum_{i=1}^{d} 1(1(x_i \leq r_i) \neq \tilde{1}(x_i \leq r_i)))dx \\ &\leq 1 * 2 * d * \frac{1}{\tau} \\ &= \frac{2d}{\tau}\end{aligned} \tag{2.10}$$

According to equation 2.4, if $\tilde{\tilde{h}}(x) - \tilde{h}(x) \neq 0$, $\sum_i \tilde{1}(x_i \geq l_i) + \tilde{1}(x_i \leq r_i) - 2d \in (0, \frac{1}{\tau})$,
so $\exists i : x_i \in (l_i, l_i + \frac{1}{\tau})$ or $x_i \in (r_i, r_i - \frac{1}{\tau})$.
Integration of these ranges over all dimensions makes x bounded by $\frac{2d}{\tau}$, so:

$$\int_{x\in[0,1]^d} \left|\tilde{\tilde{h}}(x) - \tilde{h}(x)\right| dx \leq \frac{2d}{\tau} \tag{2.11}$$

Combining equation 2.10 and 2.11, we have:

$$\begin{aligned}\int_{x\in[0,1]^d} \left|\tilde{\tilde{h}}(x) - h(x)\right| dx &= \int_{x\in[0,1]^d} \left|\tilde{\tilde{h}}(x) - \tilde{h}(x) + \tilde{h}(x) - h(x)\right| dx \\ &\leq \int_{x\in[0,1]^d} (\left|\tilde{\tilde{h}}(x) - \tilde{h}(x)\right| + \left|\tilde{h}(x) - h(x)\right|)dx \\ &\leq \frac{2d}{\tau} + \frac{2d}{\tau} \\ &= \frac{4d}{\tau}\end{aligned} \tag{2.12}$$

■

To conclude, we represent f using a linear combination of queries (whether input is in cell C_i). Because the indicator for each cell can be approximated precisely with a 2-layer network with ReLU activation function, the function g can be constructed by a 3-layer neural network composed of such sub-networks.

Finally, we need to determine the size of this network. Because:

- $\tilde{\tilde{h}}$ is a 2-layer net with ReLU activations and $O(d)$ nodes
- f is L-Lipschitz with an input of dimension d
- The size of each grid along each dimension id ϵ

The size of this 3-layer neural network is $O\left(d\left(\frac{L}{\epsilon}\right)^d\right)$. ■

2.3 Escaping the curse of dimensionality

One potential vulnerability of theorem 2.2 is the size of the neural network is involved with the dimension of input (curse of dimensionality). In this section we'll prove this can be gotten rid of if the function can be represented by functions of certain frequencies.

The Fourier basis for "nice" functions from $\mathbb{R}^d \to \mathbb{R}$ consists of basis functions:

$$e_\omega(x) = e^{i\langle\omega,x\rangle} = cos(\langle\omega,x\rangle) + isin(\langle\omega,x\rangle), \omega \in \mathbb{R}^d \tag{2.13}$$

The Fourier integral theorem gives coefficients for this basis:

$$\hat{f}(\omega) = \int_{\mathbb{R}^d} f(x)e^{-i\langle\omega,x\rangle}dx \tag{2.14}$$

$$f(x) = \int_{\mathbb{R}^d} \hat{f}(\omega)e^{i\langle\omega,x\rangle}d\omega \tag{2.15}$$

Barron constant:

$$C = \int_{\mathbb{R}^d} \|\omega\| \left|\hat{f}(\omega)\right| d\omega \tag{2.16}$$

Theorem 2.5 *For any f: $\mathbb{B} \to \mathbb{R}$, $\mathbb{B} = \{x \in \mathbb{R}^d : \|x\| \leq 1\}$, there exists a 3-layer neural network $\hat{f}$ with $O(\frac{C^2}{\epsilon})$ neurons and sigmoid activation, s.t.*

$$\int_{\mathbb{B}} (f(x) - \hat{f}(x))^2 dx \leq \epsilon$$

Proof: The proof can be divided into 3 parts:

1. Write $f(x)$ as infinite convex combination of cosine-like activations
2. Prove that $f(x)$ with small Barron constant can be approximated using small number of cosine-like activations
3. Prove the cosine non-linearities can be approximated by sigmoid non-linearities

- **Part I**

  By Fourier integral theorem:

$$\begin{aligned} f(x) &= f(0) + \int_{\mathbb{R}^d} \hat{f}(\omega)(e^{i\langle\omega,x\rangle} - 1)d\omega \\ &= f(0) + \int_{\mathbb{R}^d} \left|\hat{f}(\omega)\right| (e^{i(b_\omega + \langle\omega,x\rangle)} - e^{ib\omega})d\omega \end{aligned} \tag{2.17}$$

where $\left|\hat{f}(\omega)\right|$ is the magnitude, b_ω is the phase.

Because f is a real-valued function, only the real part of the above expression will survive (also we neglect the constant part $f(0)$):

$$\begin{aligned} f(x) &= \int_{\mathbb{R}^d} \left|\hat{f}(\omega)\right| (\cos(b_\omega + \langle \omega, x\rangle) - \cos(b_\omega)) d\omega \\ &= \int_{\mathbb{R}^d} \frac{\left|\hat{f}(\omega)\right| \|\omega\|}{C} \left(\frac{C}{\|\omega\|} (\cos(b_\omega + \langle \omega, x\rangle) - \cos(b_\omega)) \right) d\omega \end{aligned} \tag{2.18}$$

where $C = \int_{\mathbb{R}^d} \left|\hat{f}(\omega)\right| \|\omega\| \, d\omega$ is the Barron's constant.
Thus we have obtained the expression as convex combination of cosine-like activations.

- **Part II**

  We'll use subsampling to prove that there is a set S of ωs, s.t.

$$f(x) \approx \frac{1}{|S|} \sum_{\omega \in S} \frac{C}{\|\omega\|} (\cos(b_\omega + \langle \omega, x\rangle) - \cos(b_\omega)) \tag{2.19}$$

The strategy is to choose $\omega \in \mathbb{R}$ to add to S with probability $\frac{\left|\hat{f}(\omega)\right| \|\omega\|}{C}$ for r times.
Let $g_i = \frac{C}{\|\omega_i\|} (\cos(b_{\omega_i} + \langle \omega_i, x\rangle) - \cos(b_{\omega_i}))$, $g = \frac{1}{r} \sum_{i=1}^{r} g_i$, then we have:

$$\mathbb{E}[g_i] = f \tag{2.20}$$

$$\begin{aligned} \mathbb{E}_x \mathbb{E}_g[(g(x) - f(x))^2] &= \mathbb{E}_x \mathbb{E}_{g_i}[(\sum_i (\frac{1}{r} g_i - \frac{1}{r} f))^2] \\ &= \frac{1}{r^2} \mathbb{E}_x \mathbb{E}_{g_i}[(\sum_i (g_i - f))^2] \\ &= \frac{1}{r^2} (\mathbb{E}_x \mathbb{E}_g[(g_i - f)^2] + \sum_{i \neq j} \mathbb{E}_x \mathbb{E}_{g_i, g_j}[(g_i - f)(g_j - f)]) \\ &= \frac{1}{r^2} (\mathbb{E}_x \mathbb{E}_g[(g_i - \mathbb{E}[g])^2] + \sum_{i \neq j} \mathbb{E}_x[\mathbb{E}_{g_i}[(g_i - f)] \, \mathbb{E}_{g_j}[(g_j - f)]]) \\ &= \frac{1}{r} (\mathbb{E}_x \mathbb{E}_g[g^2] - \mathbb{E}_x \mathbb{E}_g[g^2]) + 0 \\ &\leq \frac{1}{r} \mathbb{E}_x \mathbb{E}_g[g^2] \\ &\leq \frac{1}{r} \max_\omega \mathbb{E}_x[g_\omega^2] \end{aligned} \tag{2.21}$$

Then we use the fact that cos is 1-Lipschitz. (You can show this by using the fact that cos is bounded between -1 and 1 and apply Mean Value Theorem). Hence,

$$|(\cos(b_w + \langle w, x\rangle) - \cos(b_w))| \leq |\langle w, x\rangle| \leq \|w\| \|x\|$$

Therefore,

$$\left(\frac{C}{\|w\|} (\cos(b_w + \langle w, x\rangle) - \cos(b_w)) \right)^2 \leq C^2 \|x\|^2 \leq C^2$$

Then integrating gives us exactly the expectation of $\mathbb{E}_x[g_\omega^2]$, thus

$$\forall w : \int_{x\in\mathbb{B}} \left(\frac{C}{\|w\|}(\cos(b_w + \langle w, x\rangle) - \cos(b_w))\right)^2 dx \le C^2$$

Simply substituting this new bound into 2.21, we get

$$\mathbb{E}_g\mathbb{E}_x[(g-f)^2] \le \frac{C^2}{r}$$

By the definition of expectation, there exists a width r network g, with cosine-line activation, s.t.

$$\exists g, \mathbb{E}_x[(g(x) - f(x))^2] \le \frac{C^2}{r}$$

- **Part III**

  The last step is to approximate the cosine-like activation function with sigmoids. First, for convenience, we denote:

$$g_w(x) = \frac{C}{\|w\|}\left(\cos\left(b_w + \langle w, x\rangle\right) - \cos\left(b_w\right)\right).$$

  The approximation is given by the following lemma:

  Lemma 2.6 *There exists a 2-layer neural net G_0 of size $O\left(\frac{1}{\epsilon}\right)$ with sigmoid activations, s.t.*

$$\sup_{x\in\mathbb{B}} |G_0(x) - g_w(x)| \le \epsilon.$$

  Instead of proving $g_w(x)$ directly, let's define an auxillary function h

$$h_w(y) := \frac{C}{\|w\|}\left(\cos\left(b_w + \|w\|y\right) - \cos\left(b_w\right)\right).$$

  Note that

$$g_w(x) = h_w\left(\left\langle \frac{w}{\|w\|}, x\right\rangle\right)$$

  is a composition of a linear function and h_w, so it suffices to approximate h with sigmoid activations. First observe that h is C-Lipschitz by taking its derivative:

$$|h_w'(y)| = |C\sin\left(b_w + \|w\|y\right)| \le C.$$

  Then we can follow the idea for proving the first theorem: Grid the interval $[-1, 1]$ into intervals $[l_i, r_i]$ of size ϵ/C. Pick arbitrary $y_i \in [l_i, r_i]$, we have:

$$\sup_{x\in[-1,1]} \left|\sum_i 1\left(y \in [l_i, r_i]\right) h_w\left(y_i\right) - h_w(y)\right| \le \epsilon$$

  Finally, we observe that we can write the indicator function as the difference of step functions.

$$1(y \in [l_i, r_i]) = 1(y \ge l_i) - 1(y \ge r_i)$$

  So, it suffices to approximate the step function using a sigmoid.

For approximating a step function using a sigmoid, observe that

$$\lim_{\tau \to \infty} \sup_{x \in [0,1]^d} \left| 1(x \geq a) - \frac{1}{1 + e^{-\tau(x-a)}} \right| = 0$$

This allows us to drive the error as small as required, by taking a τ that is large enough.

Thus, we can approximate the required function using a 2-layer neural network (first layer for obtaining the sigmoids to approximate the step function, and the second layer to perform the linear combination).

■

2.4 Concluding thoughts

1. All the results that been presented and proven in the lecture are **existential** i.e. they prove that a good approximator exists. But we do not deal with the problem of finding one efficiently (much less so using gradient descent).
2. The choice of non-linearity in the the theorems in very flexible i.e. all the presented results can be shown to hold true for other choices of non-linearity as well.
3. There are many other similar results available, such as results showing the deep, but narrow networks are universal estimators.