E0 306: Deep Learning: Theory and Practice 14^{th} February 2019

Scribe Notes for Lecture 11

Lecturer: Amit Deshpande *Scribe*: Rahul Raychaudhury

1 Recap

1.1 Notion of Approximation

In the previous lecture we developed two notions of approximating a function. One notion is approximating a function everywhere in a certain interval.

Definition : h is a pointwise ϵ-approximation of f in the interval $[a, b]$ if $|f(x) - h(x)| \leq \epsilon$ for all $x \in [a, b]$.

The other notion is to define a measure on an interval and try to minimize the expected squared error (with respect to the measure) on that interval.

Definition : h is an ϵ-approximation of f with respect to measure μ on interval $[a, b]$ if $E_{x\sim\mu}[(f(x) - h(x))^2] = \int_a^b (f(x) - h(x))^2 u(x)dx \leq \epsilon$.

The function we are trying to approximate could be defined on R^d and μ could be a measure on R^d. In general we are interested in how good our approximation can be if we take our hypothesis from a certain concept class. Specifically, we are interested in the class of neural networks using a certain activation function, of a certain depth/width and range of weights. Barron's Theorem gave a quantitative answer to this question with respect to the second notion of approximation for neural networks with one hidden layer and no activation on the output neuron.

1.2 Barron's Theorem

Barron's Theorem(informal): For every 'nice' $f : B \to R$ where B is bounded and subset of R^d and μ gives a probability density on B, there exists a function $h(x) = \sum_{k=1}^{r} \alpha_k \sigma(w^T x + b_k) + c_0$ with $r = O(\frac{C}{\epsilon})$ such that $E_{x\sim\mu}[(f(x) - h(x))^2] = \int_B (f(x) - h(x))^2 \mu(x)dx \leq \epsilon$. Note that C depends on both f and μ.
Proof idea:
Step 1: The proof basically takes the function f, looks at its Fourier Transform and shows that the function $f(x) - f(0)$ lies in the convex hull of functions of the form $cos(w^T x + b)$ with some bounds on w and b. These bounds depend on both f and μ.
$f(x) - f(0) \in Conv(cos(w^T x + b)_{w,b})$

Step 2: The proof shows a function $h(x)$ that is close to $f(x) - f(0)$ in the following way. $E_{x\sim\mu}[((f(x) - f(0)) - h(x))^2] \leq \epsilon$ where $h(x) = \sum_{k=1}^{O(\frac{1}{\epsilon})} \alpha_k cos({w_k}^T x + b_k)$. To obtain such a function h the proof uses the fact that since $f(x)-f(0)$ lies in the convex hull of trigonometric functions, and hence can be written as an expected function with some probability over these trigonometric functions. Then by taking $O(\frac{1}{\epsilon})$ iid samples of them and taking their average the required h is obtained due to Chebyshev's bound.

The specifics of the proof is covered in the previous lecture. Note that the proof is non-constructive, given a certain function that satisfies the niceness requirements the proof does not give a way to construct the neural network the theorem promises. Also, Barron's Theorem goes further and shows that we can do the above for sigmoids whereas above we have shown for cosines.

2 Low degree Polynomial Approximations

In Barron's Theorem we showed how trigonometric functions were used to approximate a function and then one went from trigonometric functions to step functions to sigmoids. In this section we try to do the opposite in some sense. We try to approximate the sigmoidal/ReLU activation using low degree polynomials. These approximations give us insight into the expressiveness of ReLU/sigmoid networks.

2.1 Approximating the Sigmoid activation

The fact that the sigmoid function can be approximated as closely as desired within an interval follows from the Weierstrass theorem in real analysis. The highlight of the following approximation is that it is relatively low degree.

2.1.1 Chebyshev Polynomials

Consider the following polynomials, $p_0(z) = 1$, $p_1(z) = z$, $p_{n+1} = 2zp_n(z) - p_{n-1}(z)$.
These are called Chebyshev polynomials of the first kind. Let α_n denote the projection of the function f on the nth Chebyshev Polynomial with respect to the measure $\mu(x) = \frac{1}{\pi}\frac{1}{\sqrt{1-x^2}}$. $\alpha_i =< f, p_i >= \int_{-1}^{1} p_i(z)f(z)u(z)dz$. Any continuous function f on the interval $[-1, 1]$ can be written as $\sum_{n=0}^{\infty} \alpha_n p_n$. Truncating the series after some threshold $n = N$ gives a degree N polynomial approximation. Observe that the Chebyshev polynomials form an orthonormal basis, that is, $< p_i, p_j >= \int_{-1}^{1} p_i(z)p_j(z)u(z)dz$ equals 0 if $i \neq j$ and 1 if $i = j$.

2.1.2 Using Chebyshev Polynomials to obtain low degree approximation

Let $\sigma(z) := \frac{1}{1+e^{-4Lz}}$. We want to approximate this function using a low degree polynomial. Our approach is to take the projection of σ upto some degree d on the span of the Chebyshev polynomials of the first kind.
$Proj_{\leq d} := \sum_{i=0}^{i=d} \alpha_i p_i$. If we can show that the Chebyshev coefficients fall exponentially in n

for our function then it would imply a low degree approximation. This turns out to be true but we skip the proof as it is long and unenlightening.
Proposition[2]: $|\alpha_k| \leq \frac{\frac{1}{L}+\frac{2}{\pi}}{(1+\frac{\pi}{4}L)^k}$ for all k. Note that the coefficients are with respect to σ as defined above.
Now observe that $|\sigma(z) - \sum_{i=0}^{d} \alpha_i p_i| = |\sum_{i=d+1}^{\infty} \alpha_i p_i| \leq \sum_{i=d+1}^{\infty} |\alpha_i| \leq \frac{4+\frac{8L}{\pi}}{(1+\frac{\pi}{4}L)^d}$ when $z \in [-1,1]$ since $\sigma([-1,1] \in [0,1])$ and $p_k([-1,1]) \in [-1,1])$.
Thus, we can obtain an ϵ approximation of σ with a polynomial of degree $O(log(\frac{1}{\epsilon}))$ in the interval $[-1,1]$.
We know that a feed-forward neural network can be thought of as a tree. We also know that composition of polynomials gives a new polynomial with degree at most the product of the original polynomials. These facts along with the previously shown polynomial approximation of the sigmoidal activation suggests implies that a sigmoidal feed-forward network of depth l can be approximated by a polynomial of depth $O(log(\frac{1}{\epsilon})^l)$ since we perform l compositions.

2.2 Approximating the ReLU activation

Next, we show that the ReLU activation can be approximated with pointwise error ϵ using a polynomial of degree $O(\frac{1}{\epsilon})$ in the interval $[-1,1]$r. The proof relies on a known result in approximation theory called Jackson's Theorem which gives a polynomial approximation to the absolute value function. Using the absolute value function to approximate other functions is a common technique in approximation theory.

Jackson's Theorem : There exists a polynomial $\tilde{p}(x)$ of degree $O(\frac{1}{\epsilon})$ such that for all $x \in [-1,1]$, $||x| - \tilde{p}(x)| \leq \frac{\epsilon}{2-\epsilon}$. Note that $\epsilon \in (0,1)$.

Proposition[1]: Let $r(x) := max(0,x)$ and $\epsilon \in (0,1)$. There exists a polynomial $p(x)$ of degree $O(\frac{1}{\epsilon})$ such that for all $x \in [-1,1]$, $|r(x) - p(x)| \leq \epsilon$ and $p([-1,1]) \subseteq [0,1]$.

Proof: Observe that $r(x) = \frac{x+|x|}{2}$. Let $\tilde{p}(x)$ be a polynomial of degree $O(\frac{1}{\epsilon})$ such that for all $x \in [-1,1]$, $||x| - \tilde{p}(x)| \leq \frac{\epsilon}{2-\epsilon}$. We have such a polynomial from Jackson's Theorem.
Let $\bar{p}(x) := \frac{\tilde{p}(x)+x}{2}$. Clearly, for all $x \in [-1,1]$, $|r(x) - \bar{p}(x)| \leq \frac{\epsilon}{2(2-\epsilon)}$.
Let $p(x) := \frac{(2-\epsilon)}{2}(\bar{p}(x) - \frac{1}{2}) + \frac{1}{2}$.
Thus, for $x \in [-1,1]$, $|r(x) - p(x)| = \frac{\epsilon}{2}|r(x)| + \frac{(2-\epsilon)}{2}(r(x) - \bar{p}(x)) + \frac{1}{2}(|\frac{(2-\epsilon)}{2} - 1|) \leq \epsilon$.
Clearly, $p([-1,1]) \subseteq [0,1]$.

Although the above proof is non-constructive we can obtain similar bounds using standard interpolation methods. Therefore the above result suggests a ReLU network of l layers can be approximated using a polynomial of degree $(\frac{1}{\epsilon})^l$ in the interval $[-1,1]$.
,

2.3 Expressive Power

If we treat the degree of the polynomial required to approximate a function as a measure of the complexity of that function, it appears that ReLU networks are much more complicated than sigmoidal networks ($log(\frac{1}{\epsilon})^l$ versus $(\frac{1}{\epsilon})^l$). This is a justification for why one might want to use ReLU networks for learning complicated concepts. However, there are downsides as the ReLU function does not have nice properties like smoothness, ease of computing the gradient etc unlike the sigmoid.
Polynomials can also be thought of as a measure of how much a function fluctuates. If a polynomial fluctuates a lot in an interval than it must have a lot of zeros which would imply it has high degree. One can use polynomial approximations with other techniques to address questions like how expressive power changes with width/depth or choice of activation function. The following result by Telgarsky in the paper *Benefits of depth in Neural Networks*[3] uses these ideas. The proof uses approximation using multiple polynomials rather than one.

Theorem(Telgarsky[3]): There exists a ReLU Feed-Forward Neural Network with $O(k^3)$ layers with $O(1)$ neurons per layer and $O(1)$ distinct parameters such that any $\frac{1}{100}$ approximation using $O(k)$ layers requires $2^{\Omega(k)}$ neurons.

The result gives a feed-forward network such that if we want to approximate said network with another network using cube root many layers we have to pay an exponential price in the number of neurons used.

References

[1] Surbhi Goel, Varun Kanade, Adam R. Klivans, and Justin Thaler. Reliably learning the relu in polynomial time. In *Proceedings of the 30th Conference on Learning Theory, COLT 2017, Amsterdam, The Netherlands, 7-10 July 2017*, pages 1004–1042, 2017.

[2] Shai Shalev-Shwartz, Ohad Shamir, and Karthik Sridharan. Learning kernel-based halfspaces with the 0-1 loss. *SIAM J. Comput.*, 40(6):1623–1646, 2011.

[3] Matus Telgarsky. Benefits of depth in neural networks. In *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*, pages 1517–1539, 2016.