Theoretical Deep Learning

Lecture 6: Two-layer neural nets and the Fourier analysis

July 29, 2021

Lecturer: Lei Wu *Scribe: Lei Wu*

A two-layer neural network is given by

$$f_m(x;\theta) = \frac{1}{m}\sum_{j=1}^{m} a_j \sigma(w_j^T x + c_j) = a\sigma(Wx + c),$$

where $a, c \in \mathbb{R}^m, W \in \mathbb{R}^{m\times d}$ and $\theta = \{a, W, c\}$ denote all the trainable parameters. See Figure 1 for a visualization of this network.

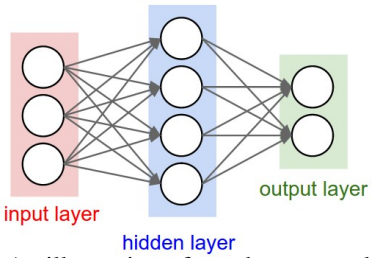

Figure 1: An illustration of two-layer neural networks.

- $\sigma : \mathbb{R} \mapsto \mathbb{R}$ is the (nonlinear) activation function, e.g., $\sigma(z) = \max(0, z)$ (ReLU), $\sigma(z) = \tanh(z)$. If z is a vector or matrix, $\sigma(z)$ should be understood in an element-wise manner.
- m denotes the number of neurons, which is referred to as the *network width*.
- If only allowing training of $\{\boldsymbol{a}_j\}$ with $\{\boldsymbol{b}_j, c_j\}$ fixed after the initialization, we obtain a random feature model, where the random feature is $\varphi(\boldsymbol{x}; \boldsymbol{b}, c) = \sigma(\boldsymbol{b}\cdot\boldsymbol{x} + c)$.

Activation functions. A list of commonly-used activation functions is given in Table 1. Some remarks

Saturating	Sigmoid	$\frac{1}{1+e^{-x}}$
	Tanh	$\frac{e^x-e^{-x}}{e^x+e^{-x}}$
Non-saturating	ReLU	$\max(0,x)$
	Leaky ReLU	$\max(ax,x)$, where a is a small value, e.g. 0.01
	Parametric ReLU	$\max(ax,x)$, with a **learnable**
	Softplus	$\ln(1+e^x)$
	GELU	$x\Phi(x)$
	SiLU	$x\sigma_{\text{sigmoid}}(\beta x)$

Table 1: Commonly used activation functions. $\Phi(\cdot)$ is the CDF of $\mathcal{N}(0,1)$.

goes as follows.

- ReLU stands for rectified linear unit, which is the most popular activation function in computer vision. But it lacks smoothness, which may be problematic in many applications.
- Softplus, Gaussian error linear unit (GELU), and sigmoid linear unit (SiLU) can be viewed as smoothed versions of ReLU. Currently, ReLU and ReLU variants are the most popular ones.
- The non-monotonic GELU and SiLU become very popular very recently, in particular in the pre-trained language models, such as BERT.
- For saturating activation functions, $\sigma'(z) \approx 0$ when $|z|$ is relatively large. This is bad for training. ReLU partially solves this problem. But there is a **dying ReLU** issue. For ReLU-activated nets, if a neuron is dead, it keeps dead for the whole training and cannot be re-activated anymore. Leaky ReLU is proposed to solve this issue.

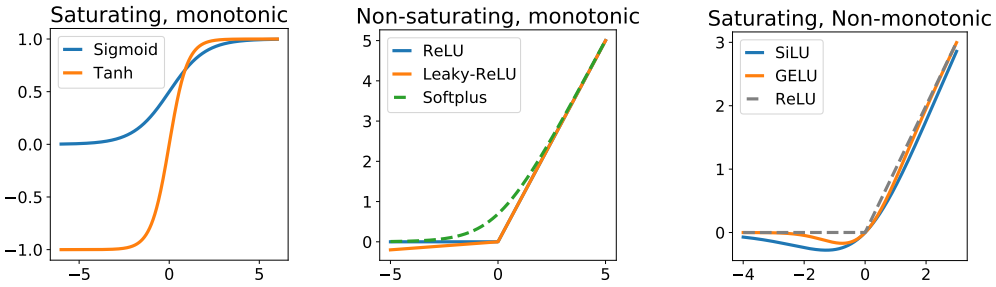

Figure 2: Comparison of several activation functions.

The set of functions that can be represented by two-layer neural nets is given by

$$\mathcal{F}_{\sigma,d} = \left\{ a\sigma(Wx+b) : a \in \mathbb{R}^m, c \in \mathbb{R}^m, W \in \mathbb{R}^{m\times d}, m \in \mathbb{N} \right\}.$$

Next, we study the approximation power of two-layer neural nets.

1 Universal approximation properties

Definition 1.1 (UAP)**.** Let $\mathcal{X}$ be a compact set. A function class $\mathcal{F}$ is said to be universal approximator if $\mathcal{F}$ is dense in $C(\mathcal{X})$ with respect to the uniform metric. This is equivalent to say that for any $f \in C(\mathcal{X})$ and $\varepsilon > 0$, there exists $f \in \mathcal{F}$ such that

$$\sup_{x\in\mathcal{X}} |f(x) - h(x)| \le \varepsilon.$$

Theorem 1.2 ([Siegel and Xu, 2020])**.** *Assume σ such that $\mathcal{F}_{\sigma,1}$ is dense in $C([0,1])$. Then, $\mathcal{F}_{\sigma,d}$ is dense in $C([0,1]^d)$.*

Proof. First, we assume that $\sigma \in C^\infty(\mathbb{R})$. Then, for any $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$,

$$\frac{\partial}{\partial w_i}\sigma(w^T x + b) = \lim_{\epsilon\to 0} \frac{\sigma(w^T x + \epsilon e_i^T x + b) - \sigma(w^T x + b)}{\epsilon} \in \overline{\mathcal{F}}_{\sigma,d}$$

for $i = 1, \ldots, d$. Similarly, for any $\alpha = (\alpha_1, \ldots, \alpha_d) \in \mathbb{N}^d$,

$$\frac{\partial}{\partial w^\alpha}\sigma(w^T x + b) = x^\alpha \sigma^{|\alpha|}(w^T x + b) \in \overline{\mathcal{F}}_{\sigma,d}.$$

Since $\mathcal{F}_{\sigma,1}$ is dense in $C([0,1])$, σ cannot be a polynomial. Hence, we can choose $w = 0$ and $b \in \mathbb{R}$ such that $\sigma^k(b) \neq 0$ for any $k \in \mathbb{N}$. Therefore, all the polynomials of the form $x_1^{\alpha_1} \cdots x_d^{\alpha_d}$ are in $\bar{\mathcal{F}}_{\sigma,d}$. This implies that $\bar{\mathcal{F}}_{\sigma,d}$ contains all the polynomials. By Weierstrass-Stone theorem, $\bar{\mathcal{F}}_{\sigma,d}$ is dense in $C(\Omega)$.

For non-smooth σ, since $\mathcal{F}_{\sigma,1}$ is dense in $C([0,1])$, we can use a two-layer neural net to approximate a smooth one. Then, the same results follow. □

The preceding problem implies that we only need to consider the one-dimensional case, where explicit constructive proof is always doable. The following lemma concerns the ReLU activation function.

Lemma 1.3. *Assume $\sigma(z) = \max(0, z)$. For any Lipschitz continuous function f, there exits a two-layer neural network $f_m(\cdot;\theta)$ such that*

$$\sup_{x\in[0,1]} |f_m(x;\theta) - f(x)| \lesssim \frac{\mathrm{Lip}(f)}{m}.$$

Proof. Let $h = \frac{1}{m}$ and $\{x_j = jh\}_{j=0}^m$ be the uniform grids in $[0,1]$. Let $t(x) = \max(1 - |x|, 0)$ be the triangular function. Then, the piecewise linear interpolator can be written as

$$\tilde{f}_m(x) = \sum_{j=0}^{m} f(x_i) t\left(\frac{x - x_i}{h}\right). \tag{1.1}$$

Consider the approximation error in the interval $[x_j, x_j + h]$: for $t \in [0, h]$,

$$\begin{aligned} |f(x_j + t) - \tilde{f}(x_j + t)| &= |f(x_j + t) - f(x_j) - \frac{f(x_j + h) - f(x_j)}{h} t| \\ &= |f'(\xi_1)t - f'(\xi_2)t| \lesssim \mathrm{Lip}(f)h. \end{aligned}$$

Hence,

$$\sup_{x\in[0,1]} |\tilde{f}_m(x) - f(x)| = \max_{j\in[m-1]} \sup_{t\in[0,h]} |f(x_j + t) - \tilde{f}(x_j + t)| \lesssim \mathrm{Lip}(f)h.$$

Notice that the triangular function can exactly represented with 3 ReLU neurons:

$$t(x) = \sigma(x + 1) + \sigma(x - 1) - 2\sigma(x).$$

Plugging it into (1.1), it shows that $\tilde{f}_m$ can be represented with a two-layer neural net with $3m$ neurons. □

Since the Lipschitz functions are dense in $C([0,1])$, we thus prove the UAP for the ReLU activation function. For other activation functions, we can use other constructive proofs. For a general proof, which holds for all the sigmoidal function:

$$\lim_{z\to-\infty} \sigma(z) = 0, \qquad \lim_{z\to\infty} \sigma(z) = 1, \tag{1.2}$$

we refer to [Cybenko, 1989].

2 Approximation with rates

Note that UAP does not show any superiority of neural nets over the classical methods, such as polynomials, spline, finite element methods, since all these methods also have UAP. To separate different methods, we need to estimate the approximation rate. Let us review some classical results.

- Approximating functions in $C(\mathcal{X})$ does not have rate.
- Lemma 1.3 can be extended to $d > 1$, where the rate is $O(\frac{1}{m^{1/d}})$. This means that to reach the accuracy ε, the number of parameters needed is ε^{-d}. For instance, taking $\varepsilon = 0.1, d = 20$, the number of parameters needed is 10^{20}. This issue is referred to as the *curse of dimensionality* (CoD).
- **High-order smoothness.** To obtain faster approximation rate, we need to consider smaller target function space. The classical approach is to add more smoothness by assume the high-order differentiability. Consider the Sobolev space defined by the Sobolev norm:

$$\|f\|_{H_d^s} = \left(\sum_{|\alpha| \leq s} |D^\alpha f|^2 \, \mathrm{d}x \right)^{1/2} < \infty.$$

  For $\mathcal{H}_d^s$, it has been shown that the minimax approximation rate of any methods is $O(m^{-s/d})$ (up to some constants). This suffers from CoD unless $s \gtrsim d$. In fact, when $s > d/2$, H_d^s is a RKHS. [LW: I should add references for the approximation of the Sobolev spaces.]
- **RKHS.** We have seen that approximation rate of random features is $O(1/m)$ for target functions in the associated RKHS. This rate avoids CoD because of the expectation representation of the functions.

One of the core tasks of theoretical deep learning is:

Identify the appropriate function classes, for which neural nets can approximate without CoD.

2.1 Avoid CoD via Fourier transform

The following procedure as first developed in [Jones, 1992]. Let $\hat{f}$ be the Fourier transform of f:

$$\hat{f}(\omega) = \frac{1}{(2\pi)^d} \int_{\mathbb{R}^d} f(x) e^{-i\omega^T x} \, \mathrm{d}x.$$

The Fourier inversion theorem shows

$$f(x) = \int \hat{f}(\omega) e^{i\omega x} \, \mathrm{d}\omega. \tag{2.1}$$

Naively thinking, we can think of (2.1) as a two-layer neural net with the activation function $\sigma(z) = e^{iz}$.

Let $\hat{f}(\omega) = |\hat{f}(\omega)| e^{ib(\omega)}$ be the polar decomposition of $\hat{f}(\omega)$. Then, we can rewrite (2.1) as follows

$$f(x) = \int |\hat{f}(\omega)| e^{i(b(\omega) + \omega^T x)} \, \mathrm{d}\omega = \int |\hat{f}(\omega)| \cos(b(\omega) + \omega^T x) \, \mathrm{d}\omega. \tag{2.2}$$

Assume $C_f = \int |\hat{f}(\omega)| \, \mathrm{d}\omega$ and let $\mathrm{d}\pi(\omega) = \frac{|\hat{f}(\omega)|}{C_f} \, \mathrm{d}\omega$. Then,

$$f(x) = C_f \, \mathbb{E}_{\omega \sim \pi}[\cos(\omega^T x + b(\omega))].$$

Thus, we represent the function as an expectation. Recall that the property of Monte-Carlo integration:

$$\mathbb{E}_{x \sim \rho}[h(x)] - \frac{1}{m} \sum_{j=1}^{m} h(x_j) \sim \frac{\mathrm{Var}(h)}{m},$$

where $x_1, \ldots, x_m$ are i.i.d. sampled from ρ. The following theorem shows that the similar result also hold for function approximation.

Theorem 2.1. *Let ρ be any probability distribution over $\mathbb{R}^d$. Assume $C_f = \int |\hat{f}(\omega)| \, \mathrm{d}\omega < \infty$, then there exists a two-layer neural net $f_m(\cdot; \theta)$ such that*

$$\|f_m(\cdot; \theta) - f\|^2_{L^2(\mathbb{P}_x)} \lesssim \frac{C_f^2}{m}.$$

Proof. Let $W = (\omega_1, \ldots, \omega_m)$ with $\{\omega_j\}$ being i.i.d. random variable sampled from π. Let

$$f_m(x; \tilde{\theta}) = \frac{1}{m} \sum_{j=1}^{m} C_f \cos(w_j^T x + b(w_j)) =: \frac{1}{m} \sum_{j=1}^{m} Z_j.$$

Moreover,

$$\begin{aligned} \mathbb{E}_W[Z_j - f(x)] &= 0 \\ \mathbb{E}_W[(Z_j - f(x))^2] &\le \mathbb{E}_W \, Z_j^2 \le C_f^2. \end{aligned} \tag{2.3}$$

Then, using the independence of Z_j, we have

$$\begin{aligned} \mathbb{E}_W[\|f_m(\cdot; \tilde{\theta}) - f\|^2_{L^2(\mathbb{P}_x)}] &= \mathbb{E}_x \, \mathbb{E}_W \, |\frac{1}{m} \sum_{j=1}^{m} (Z_j - f(x))|^2 \\ &= \mathbb{E}_x \, \frac{1}{m^2} \sum_{j=1}^{m} \mathbb{E} \, |Z_j - f(x)|^2 \le \frac{C_f^2}{m}, \end{aligned}$$

where the last inequality follows from (2.3). □

The preceding rate is a standard Monte-Carlo rate, which is independent of d. This explains the superiority of neural networks for approximating functions with $C_f = \mathrm{poly}(d)$. Note that C_f may depend on d. However, there are two issues.

- The cosine activation function is not often used in practice, though it is recently found effective in solving some scientific computing problems [Sitzmann et al., 2020].
- The input domain is $\mathbb{R}^d$. In practice, one often consider functions in a compact domain, e.g., the image where the pixel value lies in $[0, 1]$.

2.2 Barron's trick

Andrew R. Barron developed some tricks in [Barron, 1993] to solve these issues. Let Ω be a compact domain and define the dual norm

$$\|w\|_\Omega = \sup_{x\in\Omega} |w^T x|. \tag{2.4}$$

Let $\hat{w} = w/\|w\|_\Omega$. In the following, the dependence of Ω will be omitted for simplicity, but we will frequently use the property that $|\hat{w}^T x| \le 1, \forall x \in \Omega$.

Consider $f \in C(\Omega)$ and let f_e be a $L^1(\mathbb{R})$ extension of f. Since, $f(0) = \int \hat{f}_e(\omega)\,\mathrm{d}\omega$, we can express f as follows

$$\begin{aligned} f(x) - f(0) &= \int (e^{i\omega^T x} - 1)\hat{f}_e(\omega)\,\mathrm{d}\omega \\ &= \int \frac{e^{i\omega^T x} - 1}{\|\omega\|}\|\omega\|\hat{f}_e(\omega)\,\mathrm{d}\omega \\ &= \int \frac{\cos(\omega^T x + b(\omega)) - \cos(b(\omega))}{\|\omega\|}\|\omega\||\hat{f}_e(\omega)|\,\mathrm{d}\omega \\ &= \int g(\omega, x)\|\omega\||\hat{f}_e(\omega)|\,\mathrm{d}\omega, \end{aligned} \tag{2.5}$$

where

$$g(x, w) = \frac{\cos(\omega^T x + b(\omega)) - \cos(b(\omega))}{\|\omega\|}.$$

Assume that

$$C(f) := \int \|\omega\||\hat{f}(\omega)|\,\mathrm{d}\omega < \infty.$$

Then,

$$f(x) - f(0) = C_f\,\mathbb{E}_{\omega\sim\pi}[g(x, \omega)]. \tag{2.6}$$

Thus, we express f as an expectation. For a fixed ω, $g(x, \omega)$ only depends on $\omega^T x$. In other words, it is essentially an one-dimensional function. What remains is to show that $g(\cdot, \omega)$ can be approximated by two-layer neural nets.

Theorem 2.2. *Assume*

$$C_f = \inf_{f_e|_\Omega = f} \int (1 + \|\omega\|)|\hat{f}_e(\omega)| < \infty,$$

where the infimum is taken over all the $L^1(\mathbb{R})$ extension of f. Consider the sigmoidal activation function (1.2). Then, there exits a two-layer neural nets such that

$$\|f_m(\cdot; \theta) - f\|^2_{L^2(\rho)} \lesssim \frac{C_f^2}{m}.$$

Proof. First, write $g(x, \omega) = h(\hat{\omega}^T x; w)$ with $h(\cdot; w) : [-1, 1] \mapsto \mathbb{R}$ given by

$$h(t; w) = \frac{\cos(\|w\|t + b(w)) - \cos(b(w))}{\|w\|},$$

for which $\sup_{t\in[-1,1]} \max\{|h(t;w)|, |h'(t;w)|\} \le 1$. Let $H(t) = 1(t \ge 1)$ be the Heaviside step function. Then,

$$\begin{aligned} h(t;w) &= h(-1) + \int_{-1}^{t} h'(s;w)\,\mathrm{d}s \\ &= h(-1) + \int_{-1}^{1} h'(s;w) H(t-s;w)\,\mathrm{d}s, \end{aligned}$$

which means h can be represented by a two-layer neural nets activated by the step function. Plugging it into (2.6) yields

$$f(x) = f(0) + C_f' \,\mathbb{E}_{\omega\sim\pi}[h(-1;\omega)] + 2C_f' \,\mathbb{E}_{\omega\sim\pi}\,\mathbb{E}_{s\sim\mathrm{Unif}[-1,1]}[h'(s;\omega)H(\hat{\omega}^T x - s)], \tag{2.7}$$

where $C_f' = \int \|\omega\||\hat{f}_e(\omega)|\,\mathrm{d}\omega$. Thus, we write f in an expectation form. Using the fact that $\max\{h(-1;\omega), h'(s;\omega)\} \le 1$ and $|H(\hat{w}^T x - s)| \le 1$. The approximation error is bounded by

$$\begin{aligned} \text{app-err} &\lesssim \frac{C_f'^2 + f^2(0)}{m} \lesssim \frac{1}{m}\left((\int |\hat{f}_e(\omega)|\,\mathrm{d}\omega)^2 + (\int \|\omega\||\hat{f}_e(\omega)|\,\mathrm{d}\omega)^2 \right) \\ &\lesssim \frac{1}{m}\left(\int (1+\|\omega\|)|\hat{f}_e(\omega)|\,\mathrm{d}\omega \right)^2. \end{aligned}$$

Taking over all the $L^1(\mathbb{R})$ extension f_e, we complete the proof. □

3 An alternative Fourier analysis

3.1 Step functions

Notice that for $c > t > 0$,

$$e^{it} - 1 = i\int_0^t e^{is}\,\mathrm{d}s = i\int_0^c e^{is} H(t-s)\,\mathrm{d}s.$$

Combining with the case of $t < 0$, we have

$$e^{it} - 1 = i\int_0^c e^{is} H(t-s)\,\mathrm{d}s + i\int_0^{-c} e^{is} H(s-t)\,\mathrm{d}s. \tag{3.1}$$

Using this identity, we have

$$\begin{aligned} f(x) - f(0) &= \int (e^{i\omega^T x} - 1)\hat{f}_e(\omega)\,\mathrm{d}\omega \\ &= i\int_{\mathbb{R}^d}\int_0^{\|\omega\|} e^{is} H(\omega^T x - s)\,\mathrm{d}s\hat{f}_e(\omega)\,\mathrm{d}\omega + I_2 \\ &= i\int_{\mathbb{R}^d}\int_0^1 e^{i\|\omega\|t} H(\hat{\omega}^T x - t)\,\mathrm{d}t\|\omega\|\hat{f}_e(\omega)\,\mathrm{d}\omega + I_2 \qquad (s = \|\omega\|t) \\ &= i\int_{\mathbb{R}}\int_0^1 e^{i\|\omega\|t + b(\omega)} H(\hat{\omega}^T x - t)\|\omega\||\hat{f}_e(\omega)|\,\mathrm{d}t\,\mathrm{d}\omega + I_2 \end{aligned}$$

$$= \underbrace{-\int_{\mathbb{R}}\int_0^1 \sin(\|\omega\|t + b(\omega))H(\hat{\omega}^T x - t)\|\omega\||\hat{f}_e(\omega)|\,\mathrm{d}t\,\mathrm{d}\omega}_{I_1} + I_2.$$

Note that I_2 is similar to I_1 and accounts for the case $\omega^T x < 0$. We omit I_2 just for notation simplicity.

Hence, if $\int \|\omega\||\hat{f}_e(\omega)|\,\mathrm{d}\omega < \infty$, the $f(x)$ can be written as an expectation, which is an infinite-wide net.

3.2 ReLU activations

We can obtain similar Fourier-based characterization for the popular ReLU activation function. Notice that $\widehat{\nabla f} = i\omega\hat{f}$ leads to $\nabla f(0)^T x = \int_{\mathbb{R}^d} i\omega^T x \hat{f}_e(\omega)\,\mathrm{d}\omega$. Thus,

$$f(x) - \nabla f(0)^T x - f(0) = \int_{\mathbb{R}^d} (e^{i\omega^T x} - i\omega^T x - 1)\hat{f}_e(\omega)\,\mathrm{d}\omega. \tag{3.2}$$

Similar to the case of step function, $e^{i\omega^T x} - i\omega^T x - 1$ can be written as an integral form of ReLU function. The key technique is to use the identity:

$$e^{it} - it - 1 = \int_0^c \sigma(t-s)e^{is}\,\mathrm{d}s + \int_0^c \sigma(-t-s)e^{-is}\,\mathrm{d}s. \tag{3.3}$$

The two terms of the right hand side corresponds to $t > 0$ and $t \le 0$, respectively. Then,

$$\begin{aligned}
f(x) - \nabla f(0)^T x - f(0) &= \int_{\mathbb{R}^d} (e^{i\omega^T x} - i\omega^T x - 1)\hat{f}_e(\omega)\,\mathrm{d}\omega \\
&= \int_{\mathbb{R}^d}\int_0^{\|\omega\|} \sigma(\omega^T x - s)e^{is}\,\mathrm{d}s\hat{f}_e(\omega)\,\mathrm{d}\omega + I_2 \\
&= \int_{\mathbb{R}^d}\int_0^1 \sigma(\hat{\omega}^T x - s)e^{i\|\omega\|t}\,\mathrm{d}t\|\omega\|^2\hat{f}_e(\omega)\,\mathrm{d}\omega + I_2 \\
&= \underbrace{\int_{\mathbb{R}^d}\int_0^1 \cos(\|\omega\|t + b(\omega))\sigma(\hat{\omega}^T x - t)\|\omega\|^2|\hat{f}(\omega)|\,\mathrm{d}t\,\mathrm{d}\omega}_{I_1} + I_2,
\end{aligned} \tag{3.4}$$

where the I_2 is similar to I_1, accounting for the case $\omega^T x \le 0$. The explicit form of I_2 is omitted for notation simplicity. Hence, if $\int \|\omega\|^2|\hat{f}(\omega)|\,\mathrm{d}\omega < \infty$, (3.4) can be written as an expectation. Thus, Monte-Carlo discretization yields a similar rate.

3.3 general activation functions

The proceeding idea can be extended to the general ReLU^k activation function:

$$\mathrm{ReLU}^k(z) = \max(0, z^k).$$

Definition 3.1 (Spectral Barron norm)**.** For any $f \in C(\Omega)$, define

$$\|f\|_{\mathbb{F}_s} := \inf_{f_e|_\Omega = f} \int (1 + \|\omega\|)^k |\hat{f}_e(\omega)|\,\mathrm{d}\omega,$$

where the infimum is taken over all the $L^1(\mathbb{R}^d)$ extension.

Theorem 3.2. *If* $\|f\|_{\mathbb{F}_{k+1}} < \infty$*, then there exits a two-layer neural net* f_m *activated by* ReLU^k *such that*

$$\|f_m - f\|_{L^2(\rho)}^2 \lesssim \frac{\|f\|_{\mathbb{F}_{k+1}}^2}{m}.$$

For Fourier-based analysis of two-layer nets for more general activation functions, we refer to [Siegel and Xu, 2020].

References

[Barron, 1993] Barron, A. R. (1993). Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, 39(3):930–945.

[Cybenko, 1989] Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314.

[Jones, 1992] Jones, L. K. (1992). A simple lemma on greedy approximation in hilbert space and convergence rates for projection pursuit regression and neural network training. *The annals of Statistics*, pages 608–613.

[Siegel and Xu, 2020] Siegel, J. W. and Xu, J. (2020). Approximation rates for neural networks with general activation functions. *Neural Networks*, 128:313–321.

[Sitzmann et al., 2020] Sitzmann, V., Martel, J., Bergman, A., Lindell, D., and Wetzstein, G. (2020). Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33.