LECTURE NOTES & HOMEWORK SHEET ♯1
— MATHEMATICS OF DEEP LEARNING —

RÉMI GRIBONVAL

1. Overview of the course

A high-level view on neural networks. Neural networks provide flexible ways of describing parametrized families of functions $f_\theta : \mathbb{R}^d \to \mathbb{R}^k$ where θ denotes the corresponding parameters (to be detailed in due time). They are widely used in machine learning where the parameters are tuned on training data by running some optimization algorithm. Typically, for a supervised learning task, the training data is $(x_i, y_i)_{i=1}^n$ and the goal is to minimize some empirical risk $\hat{\mathcal{R}}(\theta) := \frac{1}{n}\sum_{i=1}^n \ell(f_\theta(x_i), y_i)$ which is an approximation of an ideal risk $\mathcal{R}^\star(\theta) := \mathbb{E}_{(X,Y)}\ell(f_\theta(x_i), y_i)$ under the (unknown) probability distribution from which the data is assumed to be drawn i.i.d. In the case of least squares regression, where $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}^k$ with $k = 1$, the conditional expectation $f(x) := \mathbb{E}(Y|X = x)$ minimizes the ideal risk, and the goodness of f_θ is measured by a discrepancy $d(f_\theta, f) := \mathbb{E}_X(f_\theta(X) - f(X))^2$. The success of neural networks in addressing a learning task is related to:

- **the "expressivity" of neural networks**, i.e., the capacity to approximate a given function f using a neural network f_θ. The landmark result is the so-called Universal Approximation Theorem, and other related mathematical questions revolve around **approximation theory**, to characterize which tradeoffs are achievable between approximation accuracy and number of network parameters (aka "complexity"), and how such tradeoffs may depend on "regularity" properties of f.
- **optimization algorithms**, to (approximately / empirically) minimize $\mathcal{R}^\star(\theta)$ or its empirical version $\hat{\mathcal{R}}(\theta)$. Computational scalability when the number of training samples is large is often adressed with **stochastic gradient descent** approaches, which convergence can be controlled under certain convexity and/or smoothness assumptions. Neural networks typically yield non-convex optimization problems, and guaranteeing convergence can sometimes be done by finely analyzing the shape of the **"landscape" of** $\theta \mapsto \mathcal{R}^\star(\theta)$ and/or the initialization of the algorithms.
- **statistical relevance**, depending on how many training samples (n) are available compared to the number of network parameters. An **overfitting** phenomenon is classically related to the notion of **VC-dimension**, but a so-called **"double descent"** has been recently observed, which can be explained in certain scenarios using tools from statistical physics and random matrix theory.

2. Generalities on neural networks

2.1. **Basic definitions and notations.** The realization of a neural network is a function $f_\theta : \mathbb{R}^d \to \mathbb{R}^k$ which depends on the network parameters (weights and biases) and its architecture (activation function $\varrho : \mathbb{R} \to \mathbb{R}$, aka nonlinearity, number of layers L, number of neurons per layer N_ℓ). One can define ϱ-networks of increasing complexity:

- **A neuron**: its realization is $g(x) = \varrho(\langle a, x\rangle + b),\ \forall x \in \mathbb{R}^d$ where $a \in \mathbb{R}^d$ and $b \in \mathbb{R}$
- **A scalar-valued shallow ϱ-network**, aka one-hidden-layer network: its realization is $g(x) = \sum_{i=1}^n c_i \varrho(\langle a_i, x\rangle + b_i) + d$. The **weight matrix** $A \in \mathbb{R}^{n\times d}$ with rows a_i, together with the **bias vector** $b = (b_i)_{i=1}^n$, coefficient vector $c = (c_i)_{i=1}^n$ define affine linear transforms $x \mapsto W_1(x) = Ax + b$ and $y \mapsto W_2(x) = \langle c, y\rangle + d$. The network is parameterized by $\theta = (W_1, W_2)$, and its realization reads $f_\theta = W_2 \circ \varrho \circ W_1$, where by abuse of notation we denote $\varrho : y = (y_i)_{i=1}^n \mapsto \varrho(y) = (\varrho(y_i))_{i=1}^n$.
- The realization of a **ϱ-network with L layers**, parameterized by $\theta = (W_\ell)_{\ell=1}^L$ where $W_\ell : \mathbb{R}^{N_{\ell-1}} \to \mathbb{R}^{N_\ell}$ are affine transforms, is $f_\theta = W_L \circ \varrho \ldots \circ W_1$. There are L affine layers and $L-1$ nonlinear layers, aka hidden layers, and $N_0 = d$, $N_L = k$.

To state approximation theorems we will denote $C(K)$ the Banach algebra of continuous real-valued functions on a compact set $K \subset \mathbb{R}^d$, equipped with the sup norm.

2.2. **Universal approximation.** Just as polynomials, neural networks have a universal approximation property.

Definition 1. *Single-hidden layer ϱ-networks have the* **universal approximation property (UAP)** *if the following holds: for every compact set $K \subset \mathbb{R}^d$, every continuous function $f \in C(K)$, and every $\epsilon > 0$, there exists an integer $n \geq 1$, weight vectors $a_i \in \mathbb{R}^d$, biases $b_i \in \mathbb{R}$ and coefficients $c_i \in \mathbb{R}$ such that*

$$\|f(\cdot) - \sum_{i=1}^n c_i \varrho(\langle a_i, \cdot\rangle + b_i)\|_{L^\infty(K)} \leq \epsilon.$$

Alternatively, denoting $\Sigma_n(\varrho)$ the set of realizations of single-hidden-layer ϱ-networks with n hidden neurons, which write $g(\cdot) = \sum_{i=1}^n c_i \varrho(\langle a_i, \cdot\rangle + b_i) + d$, the universal approximation property means that $\Sigma_\infty(\varrho) = \cup_{n\in\mathbb{N}} \Sigma_n(\varrho)$ is dense in $C(K)$, with respect to the sup-norm.

The UAP was initially proved [5, 2] with sigmoidal activation functions: non-decreasing functions $\varrho : \mathbb{R} \to \mathbb{R}$ such that $\lim_{t\to-\infty} \varrho(t) = 0$ and $\lim_{t\to+\infty} \varrho(t) = 1$ [7]. To what extent can the sigmoid-like assumption be weakened? A few counter-examples are helpful.

Exercise 1 (☕). *Show that if $\varrho : \mathbb{R} \to \mathbb{R}$ is affine (i.e. there are c, d such that $\varrho(t) = ct + d$) then any realization of a corresponding network, no matter how deep, is an affine function.*

Exercise 2 (☕). *Show that if $\varrho : \mathbb{R} \to \mathbb{R}$ is polynomial of degree at most r then any realization of a ϱ-network of depth at most L is a polynomial. Give a bound on its degree.*

These are the only counter-examples, if we restrict to continuous activation functions.

Theorem 1 ([6, Theorem 1])**.** *Let $\varrho : \mathbb{R} \to \mathbb{R}$ be continuous. The following are equivalent:*

(1) *single-hidden layer ϱ-networks have the universal approximation property*
(2) *ϱ is not a polynomial*

The proof exploits Stone-Weierstrass's theorem.

Theorem 2 (Stone-Weierstrass [9, 3])**.** *Consider K a compact metric space. A sub-algebra $\mathcal{A}$ of $C(K)$ is dense in $C(K)$ if, and only if*

(1) *it separates points: for every $x, y \in K$ there is $f \in \mathcal{A}$ such that $f(x) \neq f(y)$;*
(2) *for every $x \in K$ there is $f \in \mathcal{A}$ such that $f(x) \neq 0$.*

A classical example is with $\mathcal{A}$ the set of polynomials. Theorem 1 is proved in three steps.

- Step 1: reduction to the univariate case
- Step 2: reduction to the smooth case
- Step 3: proof for the univariate smooth case

In Step 1, using Stone-Weierstrass with $\mathcal{A}$ the set of all finite linear combinations of functions $\{x \mapsto \exp(\langle v, x\rangle),\ v \in \mathbb{R}^d\}$, it is sufficient to prove that for every $v \in \mathbb{R}^d$, the function $f : x \mapsto \exp(\langle v, x\rangle)$ can be approximated arbitrarily well in the sup-norm on any compact. It is thus sufficient to prove that the closure of $\Sigma_\infty(\varrho)$ contains $t \mapsto e^t$.

In Step 2, we smooth the activation function ϱ with a compactly supported C^∞ function ψ to obtain $\sigma = \psi \star \varrho$, which is C^∞, and show that for any $a, b \in \mathbb{R}$, the function $x \mapsto \sigma(ax + b) = \int_{\mathbb{R}} \varrho(ax + b - y)\psi(y)dy$ is in the closure of $\Sigma_\infty(\varrho)$. As a result, it is sufficient to show that $\Sigma_\infty(\sigma)$ is dense in $C(K)$ to establish that the same holds for $\Sigma_\infty(\varrho)$.

In Step 3, we show that whenever σ is C^∞ and not a polynomial, the closure of $\Sigma_\infty(\sigma)$ contains all monomials $x \mapsto x^k$, hence all polynomials. Since polynomials are dense in $C(K)$ (by Stone-Weierstrass again), this yields the conclusion. Indeed, by induction on the integer $k \geq 0$, for every compact K the closure of $\Sigma_{2^k}(\sigma)$ in $C(K)$ contains $x \mapsto \frac{\sigma^{(k)}(t)}{k!}x^k$ for every $t \in \mathbb{R}$. Given k, since σ is not a polynomial, there exists $t_k \in \mathbb{R}$ such that $\sigma^{(k)}(t_k) \neq 0$, hence $x \mapsto x^k$ is in the closure of $\Sigma_{2^k}(\sigma)$, hence also in the closure of $\Sigma_\infty(\sigma)$.

Exercise 3 (☕☕)**.** *Check carefully the density arguments, and fill in the missing details in the above proof. In particular: why is σ not a polynomial ?*

2.3. **A pathological activation function.** For any non-polynomial activation function, by the UAP, any $f \in C(K)$ is arbitrarily well approximated by a shallow network *provided the number of hidden neurons, n, grows to infinity.* If we allow some depth, this can also be achieved with a *fixed* number of neurons . . . for certain pathological activation functions.

Theorem 3 ([7, Theorem 4])**.** *Denote $\Sigma_n(\varrho)$ the set of realizations of ϱ-networks with $L = 3$ layers (two hidden layers) and at most n hidden neurons. There exists an* analytic and sigmoidal *activation function $\varrho : \mathbb{R} \to \mathbb{R}$ such that: for any dimension d, $\Sigma_n(\varrho)$ is dense in $C(K)$ with $K = [0, 1]^d$ and $n = 9d + 3$.*

In fact, there is an architecture of neural network in $\Sigma_n(\varrho)$ with $N_0 = d$, $N_1 = 3d$, $N_2 = 6d + 3$, $N_3 = 1$, such that any $f \in C(K)$ can be approximated arbitrarily well by tuning appropriately the $W = 21d^2 + 15d + 3$ weights as well as the n biases.

The detailed proof involves Kolmogorov's superposition theorem and will not be presented here. A simpler but still striking result has a more elementary proof

Theorem 4. *There is an activation function $\varrho : \mathbb{R} \to \mathbb{R}$, which is analytic almost everywhere, such that in the univariate case ($d = 1$) the set $\Sigma_1(\varrho)$ of shallow networks with $n = 1$ neuron is dense in $C(K)$ for every compact set $K \subset \mathbb{R}$.*

Proof. Wlog (up to dilations) we consider $K = [0, 1]$. By Stone-Weierstrass, the set $\mathcal{A}$ of polynomials with rational coefficients is dense in $C(K)$. As $\mathcal{A}$ is countable, consider an enumeration $\mathcal{A} = \{u_n\}_{n\in\mathbb{N}}$ and define ϱ as the concatenation of all functions $u_n : [0, 1] \to \mathbb{R}$. In other words, set $\varrho(x) := u_n(x - n)$ with $n = \lfloor x \rfloor$ for $x \geq 0$ (and $\varrho(x) = 0$ for $x < 0$). □

3. Focus on ReLU networks

From now on we focus on networks with the ReLU activation function

$$\varrho(t) = \texttt{ReLU}(t) = t_+ = \max(t, 0)$$

Exercise 4 (☕). *Explain how to implement the following functions (see Figure 1) with a* `ReLU`*-network:*

- *a sigmoid-like function, with a shallow network ($L = 2$);*
- *the absolute value function, with $L = 2$;*
- *a hat function, first with $L = 2$, then with $L = 3$ and a more narrow hidden layer;*
- *the soft-thresholding function, $f(t) = t(1 - 1/|t|)_+$;*
- *the identity function.*

Python exercise (optional): write a jupyter notebook showing these implementations and the graph representation of the corresponding networks (for example using `pytorch`*).*

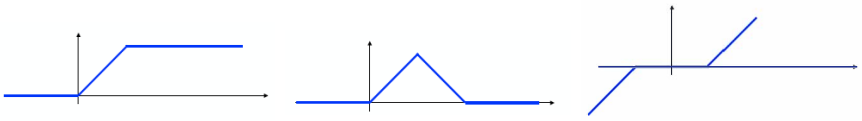

Figure 1. (left) sigmoid-like function; (middle) hat function; (right) soft-thresholding function

3.1. `ReLU`-networks and continuous piecewise linear functions.

Lemma 1. *The realization of a* `ReLU`*-network is continuous and piecewise affine linear (denoted CPwL). Start by proving it for a shallow network, then use composition.*

Exercise 5 (☕). *Prove it*

The converse is true for *univariate* CPwL functions ($d = 1$).

Lemma 2. *If $f : \mathbb{R} \to \mathbb{R}$ is CPwL then it is the realization of some shallow* `ReLU`*-network.*

Exercise 6 (☕). *Prove it.*

For CPwL functions $f : \mathbb{R}^d \to \mathbb{R}$ with $d > 1$, the converse no longer holds.

Exercise 7 (☕☕). *Prove that* $(x, y) \in \mathbb{R}^2 \mapsto \min(0, \max(x, y))$ *is the realization of a* `ReLU`*-network with* $L = 3$ *layers, but not the realization of a shallow* `ReLU`*-network.*

Lemma 3. *If* $f : \mathbb{R}^d \to \mathbb{R}$ *is CPwL, compactly supported, and nonzero, then it cannot be implemented as the realization of any shallow* `ReLU`*-network.*

Exercise 8 (☕). *Show that proving the result for* $d = 2$ *is enough to prove it for any* d*.*

Exercise 9 (Optional, ☕☕☕). *Prove the result for* $d = 2$*.*

Increasing the depth to $L \geq 3$ is thus necessary. For $d \in \{2, 3\}$, $L = 3$ is in fact sufficient to implement any CPwL function, as a consequence of the following theorem.

Theorem 5 ([1, Theorem 2.1]). *Every CPwL function* $f : \mathbb{R}^d \to \mathbb{R}$ *is the realization of a* `ReLU`*-network of depth* $L = \lceil \log_2(d + 1) \rceil + 1$*.*

In dimension $d \in \{2, 3\}$, $L = 3$ is necessary and sufficient. For $d \geq 4$, the minimum number of layers needed to implement any CPwL function is still unknown.

3.2. **Number of pieces and complexity of a network.** When f is CPwL, there exists a finite partition of $\mathbb{R}^d$ into connected sets such that the restriction of f to each of these sets is affine. The "number of pieces" of f is the smallest size among all such partitions. In the univariate case, n is thus the size of the smallest partition of $\mathbb{R}$ into intervals $I_1 = (-\infty, t_1)$, $I_2 = [t_1, t_2)$, $\dots I_{n-1} = [t_{n-2}, t_{n-1})$, $I_n = [t_{n-1}, \infty)$ such that f is affine on each I_i. To these n intervals correspond $n - 1$ breakpoints $a_1 < \ldots < a_{n-1}$.

For any ϱ, the complexity of a (shallow or deep) ϱ-network parameterized by $\theta = (W_1, \ldots, W_L)$, with $W_\ell : x \in \mathbb{R}^{N_{\ell-1}} \mapsto A_\ell x + b_\ell \in \mathbb{R}^{N_\ell}$, can be measured in terms of:

- the number of hidden neurons $N(\theta) := \sum_{\ell=1}^{L-1} N_\ell$;
- the number of connections (aka number of nonzero weights) $W(\theta) := \sum_{\ell=1}^{L} \|A_\ell\|_0$;
- the width is $\max_{1 \leq \ell \leq L-1} N_\ell$

Here, the ℓ^0 pseudo-norm of a matrix simply counts the number of nonzero entries.

The number of pieces of univariate shallow `ReLU`-networks is directly related to their complexity.

Exercise 10 (☕). *Consider* $f_\theta : \mathbb{R} \to \mathbb{R}$ *the realization of a shallow* `ReLU`*-network.*

- *Show that* $W(\theta) \leq 2N(\theta)$ *and that there exists a shallow* `ReLU`*-network* θ' *with the same realization* $f_{\theta'} = f_\theta$ *such that* $2N(\theta') = W(\theta') \leq W(\theta)$*.*
- *Show that the number of breakpoints of* $f_\theta = f_{\theta'}$ *is* $N(\theta') = W(\theta')/2$*.*

Bounds on the number pieces of deeper univariate `ReLU`-networks are also available.

Lemma 4 ([4, Lemma 5.19]). *There exist constants* $C_L, L \geq 2$ *with the following property. If* $f_\theta : \mathbb{R} \to \mathbb{R}$ *is the realization of a* `ReLU`*-network with* L *(affine) layers, then its number of pieces is bounded by* $C_L \min\left(N(\theta)^{L-1}, W(\theta)^{\lfloor L/2 \rfloor}\right)$*.*

The proof makes extensive use of results by Telgarsky [10].

Exercise 11 (Optional ☕☕☕). *Show that these bounds are sharp.*
Hint: use the sawtooth functions on $[0,1]$ defined recursively from the hat function $h_0 = \max(2x, 2-2x)$ as $h_{j+1} = h_j \circ h_0$ and their various implementations as `ReLU`-networks.
Python exercise (optional)provide a python implementation of these sawtooth functions as `ReLU` *network and illustrate both the realization and the graph of the network.*

4. From implementation to approximation

While `ReLU`-networks can only implement CPwL functions, they can be used to *approximate* any function f. How good is this approximation when we increase the network complexity ? Upper bounds are available under some assumptions on the regularity of f.

Theorem 6. *Assume f is 1-Lipschitz on $[0,1]$. For every $n \geq 1$ there is a shallow ReLU neural network with n neurons such that $\|f-g\|_{L^\infty([0,1])} \leq 2/(n+1)$.*

In Class Exercise 1. *Prove this result. Can we control the magnitude of the weights ?*

Solution 1. *Consider a regular grid $x_i = i/(n+1)$, $0 \leq i \leq n+1$. Let $g \in \Sigma_n$ be the CPwL function with n breakpoints $x_i, 1 \leq i \leq n$ such that $g(x_i) = f(x_i)$ for all i. For $0 \leq i \leq n$ and $x \in [x_i, x_{i+1}]$, since g is piecewise linear and f is 1-Lipschitz, we have*

$$|g(x) - g(x_i)| \leq |g(x_{i+1}) - g(x_i)| = |f(x_{i+1}) - f(x_i)| \leq |x_{i+1} - x_i| \leq 1/(n+1)$$

hence, using that $f(x_i) = g(x_i)$,

$$|f(x) - g(x)| \leq |f(x) - f(x_i)| + |f(x_i) - g(x)| \leq |x - x_i| + |g(x_i) - g(x)| \leq 2/(n+1)$$

The above result implies that the error of best approximation of a Lipschitz function by a shallow `ReLU`-network with n hidden neurons satisfies $E_n(f) := d(f, \Sigma_n) \leq 2|f|_{\text{Lip}}/(n+1)$. There are similar results in higher dimension.

Can we get even higher approximation rates by assuming more regularity on f ?

4.1. Lower approximation bounds for shallow networks.

Lemma 5. *Assume that $f \in C^3(\mathbb{R})$ is not affine. There is $C > 0$ s.t.: if g is piecewise affine with n pieces, then $\|f-g\|_{L^1(\mathbb{R})} \geq Cn^{-2}$.*

Proof. **Warmup**: consider $X^2 : x \mapsto x^2$, $\mathbb{P}_r$ the set of polynomials of degree at most r, and observe that $C_0 := \text{dist}(X^2, \mathbb{P}_1)_{L^1(K)} > 0$ with $K = [0,1]$. Also observe that

- $\exists x_0$ s.t. $f''(x_0) \neq 0$; wlog (shift and multiply) $x_0 = 0$ and $f''(x_0) > 2$;
- wlog (dilate and multiply) $f''(x) \geq 2$ on $K = [0,1]$ and $\|f'''\|_{L^\infty(K)} \leq 3C_0$.
  $g : x \mapsto \gamma^{-2} f(x\gamma)$; $g'(x) = \gamma^{-1} f'(x\gamma)$, $g''(x) = f''(x\gamma)$, $g'''(x) = \gamma f'''(x\gamma)$
- these shifts/multiplication/dilations rescale $\|f-g\|_{L^1(\mathbb{R})}$ by some constant $C(f)$.
- the restriction of g to $[0,1]$ has at most n pieces $I_i \subset [0,1]$

$$\|f-g\|_{L^1(\mathbb{R})} \geq \|f-g\|_{L^1([0,1])} \geq \sum_{i=1}^{n} \|f-g\|_{L^1(I_i)}$$

Main technical step (Taylor-like, postponed to Lemma 6): $\|f-g\|_{L^1(I_i)} \geq \frac{C_0}{2}|I_i|^3$

- Using Hölder inequality

$$\left(\sum_{i=1}^{n}|I_i|^3\right)^{1/3}\times n^{1-1/3}\geq\sum_{i=1}^{n}|I_i|\times 1=1.$$

Conclusion: $\|f-g\|_{L^1(\mathbb{R})}\geq C(f)\frac{C_0}{2}n^{-2}$. □

Lemma 6. *Let $C_0:=\mathrm{dist}(X^2,\mathbb{P}_1)_{L^1(K)}$ with $K=[0,1]$. Consider $f\in C^3(K)$ such that $f''\geq 2$ on $K=[0,1]$ and $C:=\|f'''\|_{L^\infty(K)}\leq 3C_0$. For any interval $I\subset K$ we have*

$$\|f-P\|_{L^1(I)}\geq\frac{C_0}{2}|I|^3,\quad\forall P\in\mathbb{P}_1.$$

Proof. Denote $I=[a,b]$;
Step 1: By change of variable, for each $P\in\mathbb{P}_1$ there are $Q,R\in\mathbb{P}_1$ such that

$$\|X^2-P\|_{L^1(I)}=(b-a)\|(b-a)^2X^2-Q\|_{L^1(K)}=(b-a)^3\|X^2-R\|_{L^1(K)}\geq C_0|I|^3.$$

Step 2: The Taylor expansion of f at a, $T=\alpha X^2+P$, $P\in\mathbb{P}_1$, $\alpha=f''(a)/2\geq 1$, satisfies $\|f-T\|_{L^\infty(I)}\leq\frac{C}{6}|I|^3\leq\frac{C_0}{2}|I|^3$ hence

$$\|f-T\|_{L^1(I)}\leq\|f-T\|_{L^\infty(I)}\leq\frac{C_0}{2}|I|^3.$$

Step 3: For each $Q\in\mathbb{P}_1$ there is $R\in\mathbb{P}_1$ such that, with $T=\alpha X^2+P$ as above,

$$\|f-Q\|_{L^1(I)}\geq\|T-Q\|-\|f-T\|=\alpha\|X^2-R\|-\|f-T\|\geq C_0|I|^3-\frac{C_0}{2}|I|^3=\frac{C_0}{2}|I|^3.$$

□

Corollary 1. *If $f\in C^3(\mathbb{R})$ is not affine then $d(f,\Sigma_n)_{L^1}\geq C(f)n^{-2}$, $\forall n\geq 1$.*

In particular, this holds for any compactly supported nonzero C^3 function.
Can we get a faster rate of approximation using deeper `ReLU`-networks ?

4.2. **Lower approximation bounds for networks of bounded depth.** Even with deep `ReLU`-networks, there is a limit to how well we can approximate C^3 functions.

Lemma 7. *If $f\in C^3(\mathbb{R})$ is not affine then $d(f,\Sigma_n^L)_{L^1}\geq C(f)n^{-2(L-1)}$, $\forall n\geq 1$, with Σ_n^L the set of realizations of `ReLU`-networks of depth at most L with at most n neurons.*

Exercise 12 (☕). *Prove this result and a similar one, with the set $\Sigma_n'^L$ of realizations of `ReLU`-networks of depth at most L with at most n* nonzero weights.
Hint: use Lemma 5 and Lemma 4.

Exercise 13 (Optional ☕☕☕). *Adapt Lemma 5 using the $\|\cdot\|_p$ norm, $0<p<\infty$, instead of the $\|\cdot\|_1$ norm. Adapt it to $f\in C^3(\mathbb{R}^d)$ which is not affine, in dimension $d>1$.*
Hint: see e.g. [8] where such results have been established.

Despite the limitations highlighted by Lemma 7, it can pay off to use deeper ReLU-networks. A basic building piece is the approximation of the square function, $x \mapsto x^2$, with accuracy exponential in the number of neurons / nonzero weights when depth is unconstrained [11, Proposition 2].

Lemma 8. *Consider $f : x \mapsto x^2$. There is $C > 0$ such that for every integer $j \geq 1$, there exists a* ReLU*-network θ with $N(\theta) \leq C \times j$, $W(\theta) \leq C \times j$, of width $\max_\ell N_\ell \leq C$, which realization f_θ approximates f to accuracy $4^{-(j+1)}$ in the sup-norm on $K = [0, 1]$.*

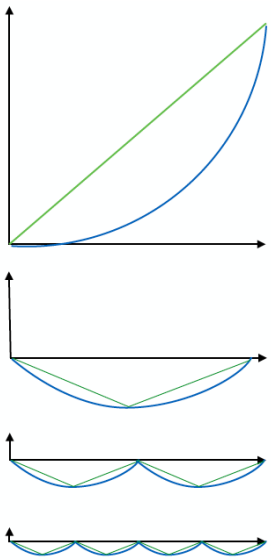

FIGURE 2. [top] Approximation of the square function f (blue) by $\mathtt{id}$ (green); [below] approximation of the residual $r_0 = f - \mathtt{id}$ (blue) by $-h_0/4$ (green); of $r_1 = r_0 + h_0/4$ (blue) by $-(1/4)^2 h_0 \circ h_0$ (green), etc.

Proof sketch. As illustrated on Figure 2, on $K = [0, 1]$, a natural approximation of f by a CPwL function with $n = 1$ piece is with $\mathtt{id}$. This leads to a residual $r_0 := f - \mathtt{id} : x \mapsto x^2 - x$, which is symmetric around $1/2$. A natural CPwL approximation of r_0 with $n = 2$ pieces is $-\frac{1}{4}h_0(x)$ with h_0 the hat function $h_0(x) = \min(2x, 2 - 2x)$. The residual $r_1 = r_0 + h_0/4$ is piecewise quadratic, on the intervals $[0, 1/2]$ and $[1/2, 1]$, and displays two shifted, dilated and scaled copies of r_0. This residual is well approximated by $-(1/4)^2 h_0 \circ h_0$, and the process can be repeated in a "fractal-like" matter. This yields a decomposition

$$f = \mathtt{id} + r_0 = \mathtt{id} - \tfrac{1}{4}h_0 + r_1 = \ldots = \mathtt{id} - \sum_{\ell=0}^{j} \frac{1}{4^{\ell+1}} h_\ell + r_{j+1},$$

where $h_j = h_0 \circ h_{j-1}$ is the sawtooth function with 2^j teeth (and 2^{j+1} linear pieces on $[0, 1]$). The truncated sum $f_j := \mathtt{id} - \sum_{\ell=0}^{j} \frac{1}{4^{\ell+1}} h_\ell$ yields $\|f - f_j\|_{L^\infty(K)} = \|r_{j+1}\|_{L^\infty(K)} = 4^{-(j+1)}$.

As h_0 is the realization of a `ReLU`-network (either shallow with $L = 2$, $N_1 = 3$, or deeper with $L = 3$, $N_1 = 2$, $N_2 = 1$), h_j is also a realization of a `ReLU`-network of controlled depth and complexity. Eventually, f_j is the realization of a `ReLU`-network is controlled by carefully reusing computations done for computing f_{j-1}. □

Exercise 14 (☕). *Fill in the missing details to exhibit a* `ReLU`*-network realizing f_j with the claimed complexity. What is the value of C? How deep is this network ?*
Python exercise (optional): write a jupyter notebook showing these implementations and the graph representation of the corresponding networks.

Exercise 15 (☕☕). *Can we hope to reduce the approximation error of f with the same complexity budget ? Can we hope to improve the lower bound in Lemma 7 ?*

Exercise 16 (☕). *Consider $M_d : (x_1, \dots, x_d) \mapsto \Pi_{i=1}^d x_i$.*
Show that for every $j \geq 1$, M_2 can be uniformly approximated on $[0,1]^2$ by the realization of some `ReLU`*-network θ of depth $L = O(j)$ with $N(\theta) = O(j)$ neurons, $W(\theta) = O(j)$ connections and accuracy exponentially decaying with j. Specify the achieved complexity and accuracy as a function of j.*
Same question with M_d on $[0,1]^d$ when d is a power of two. Same question for arbitrary d.
Python exercise (optional): write a jupyter notebook showing these implementations and the graph representation of the corresponding networks.

References

[1] Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. Understanding Deep Neural Networks with Rectified Linear Units. *ICLR*, 2018.
[2] George Cybenko. Approximation by superpositions of a sigmoidal function. *Math. Control. Signals Syst.*, 2(4):303–314, 1989.
[3] Louis de Branges. The Stone-Weierstrass theorem. *Proc. Amer. Math. Soc.*, 10(5):822–824, 1959.
[4] Remi Gribonval, Gitta Kutyniok, Morten Nielsen, and Felix Voigtlaender. Approximation spaces of deep neural networks. *arXiv.org*, May 2019.
[5] Kurt Hornik, Maxwell B Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.
[6] Moshe Leshno, Vladimir Ya Lin, Allan Pinkus, and Shimon Schocken. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, 6(6):861–867, 1993.
[7] Vitaly Maiorov and Allan Pinkus. Lower bounds for approximation by MLP neural networks. *Neurocomputing*, 25(1-3):81–91, 1999.
[8] Philipp Petersen and Felix Voigtländer. Optimal approximation of piecewise smooth functions using deep ReLU neural networks. *Neural Networks*, 108:296–330, 2018.
[9] M H Stone. The Generalized Weierstrass Approximation Theorem. *Mathematics Magazine*, 21(4):167–184, 1948.
[10] Matus Telgarsky. Benefits of depth in neural networks. *Journal of Machine Learning Research*, 49(June):1517–1539, June 2016.
[11] Dmitry Yarotsky. Error bounds for approximations with deep ReLU networks. *Neural Networks*, 2017.