2 Approximation properties

All exact science is dominated by the idea of approximation.

Bertrand Russell

We present reasons why Neural Networks structures can approximate very general objective functions. This is based on approximation theory and connection with the Finite Element theory (which is a particular approximation theory [1]). The justification follows from accuracy estimates. In numerical analysis these estimates are called **best approximation estimates**, that is there are obtained by means of theoretical constructions. The reader must be extremely attentive to the fact that best approximation estimates are by no means a guarantee that practical algorithms behave well. This is due to the fact that the mathematical techniques involved in the proofs of best approximation estimates are not related to the structure of algorithms. Such questions will be examined later by other methods.

2.1 Functional spaces

We use standard notations [88, 28, 1] for functions and functional spaces. In this text, no advanced properties of these functional spaces will be used, only the integrability conditions expressed below. The definitions are kept to the minimum.

Let $\Omega \subset \mathbb{R}^m$ be a connected subset of $\mathbb{R}^m$. The space $C^0(\Omega)$ is made with functions which are continuous over Ω. A function $f \in C^n(\Omega)$ has continuous derivatives up to order n, that is $\nabla^n f \in C^0(\Omega)^q$ with $q = m^n$.

A quadratically integrable function satisfies $\int_\Omega |f(\mathbf{x})|^2 dx < \infty$. The space of all quadratically integrable functions is denoted as $L^2(\Omega)$. It is endowed with the norm

$$\|f\|_{L^2(\Omega)} := \left(\int_\Omega |f(\mathbf{x})|^2 dx \right)^{1/2} < \infty.$$

The space $H^n(\Omega)$ is made with all functions which have n derivatives in $L^2(\Omega)$. It is endowed with the norm

$$\|f\|_{H^n(\Omega)} := \sum_{r=0}^{n} \left\| \nabla^r f \right\|_{L^2(\mathbb{R})^{m^r}} < \infty.$$

Consider a real number $1 \le s < \infty$. The space $L^s(\Omega)$ is made with all functions such that $\left(\int_\Omega |f(\mathbf{x})|^s dx\right) < \infty$. The space is endowed with the norm

$$\|f\|_{L^s(\Omega)} := \left(\int\limits_\Omega |f(\mathbf{x})|^s dx\right)^{1/s} < \infty.$$

The space $W^{n,s}(\Omega)$ is made with all functions which has n derivatives in $L^s(\Omega)$. It is endowed with the norm

$$\|f\|_{W^{n,s}(\Omega)} := \sum_{r=0}^{n} \left\|\nabla^r f\right\|_{L^s(\mathbb{R})^{m^r}} < \infty.$$

Note that $W^{n,2}(\Omega) = H^n(\Omega)$.

The previous material is generalized to $s = \infty$ by considering that $f \in L^\infty(\Omega)$ is such that

$$\|f\|_{L^\infty(\Omega)} := \sup_{\mathbf{x}\in\Omega}|f(\mathbf{x})| < \infty.$$

The space $W^{n,\infty}(\Omega)$ is made with all functions which has n derivatives in $L^\infty(\Omega)$. It is endowed with the norm

$$\|f\|_{W^{n,\infty}(\Omega)} := \sum_{r=0}^{n} \left\|\nabla^r f\right\|_{L^\infty(\mathbb{R})^{m^r}} < \infty.$$

2.2 Cybenko Theorem

The Cybenko Theorem has an historical importance in the field. It describes the asymptotic behavior of neural networks with one hidden layer of N neurons in the regime $N \to \infty$. However the Cybenko Theorem is a bit mysterious at first sight because its proof is non constructive. It is a combination of a contradiction argument with the Hahn-Banach Theorem. We note $I_d = [0,1]^d$.

Theorem 2.2.1 (Cybenko [18]). *Let S be any sigmoid. Then finite sums of the form*

$$G(\mathbf{x}) = \sum_{j=1}^{N} \alpha_j S\left(\langle \mathbf{y}_j, \mathbf{x}\rangle + \theta_j\right) \tag{2.1}$$

are dense in $C^0(I_d)$ equipped with the norm of the maximum.

The original proof is based on the following succession of theoretical arguments. The definition below needs complements in functional analysis for which we refer the reader to [88, 28].

Definition 2.2.2. *One says a sigmoid is discriminatory if it has the property. Let μ be a finite signed regular Borel measure μ with the property that*

$$\int_{I_d} \sigma(\langle \mathbf{y}, \mathbf{x}\rangle + \theta) d\mu(\mathbf{x}) = 0, \quad \forall (\mathbf{y}, \theta) \in \mathbb{R}^d \times \mathbb{R}.$$

Then $\mu = 0$.

Theorem 2.2.3. *Assume that σ is discriminatory. Then Theorem 2.2.1 holds.*

Proof. Take the set $\mathbb{S}$ made of all functions of the form (2.1). It is a linear sub-space of $C^0(I_d)$. Take its closure $\overline{\mathbb{S}}$ in $C^0(I_d)$ for the norm of the maximum. If $\overline{\mathbb{S}} = C^0(I_d)$, the proof is ended.

Assume on the contrary that $\overline{\mathbb{S}} \neq C^0(I_d)$. By the Hahn-Banach Theorem [28], there exists a bounded linear form $L \neq 0$ on $C^0(I_d)$ such that $L(\overline{\mathbb{S}}) = 0$. By the Riesz Theorem [28], there exists a finite signed regular Borel measure denoted as μ ($\mu \neq 0$ since $L \neq 0$) such that

$$L(h) = \int_{I_d} h(x) d\mu(x), \qquad \forall\, h \in C^0(I_d).$$

Taking $h(x) = \sigma(\langle \mathbf{y}, \mathbf{x}\rangle + \theta)$ for all $(\mathbf{y}, \theta) \in \mathbb{R}^d \times \mathbb{R}$ yields a contradiction. So the proof. □

It remains to prove the discriminatory property.

Lemma 2.2.4. *Any continuous monotone sigmoid is discriminatory.*

Proof. Take a preliminary $\varphi \in \mathbb{R}$, set a function $\sigma^{\lambda}_{\mathbf{y},\theta}(\mathbf{x}) = \sigma\left(\lambda(\langle \mathbf{y}, \mathbf{x}\rangle + \theta) + \varphi\right)$ and define two sets $\Pi_{\mathbf{y},\theta} = \{\mathbf{x} \mid \langle \mathbf{y}, \mathbf{x}\rangle + \theta = 0\}$ and $H_{\mathbf{y},\theta} = \{\mathbf{x} \mid \langle \mathbf{y}, \mathbf{x}\rangle + \theta > 0\}$. One has $0 = \int_{I_d} \sigma^{\lambda}_{\mathbf{y},\theta}(\mathbf{x}) d\mu(\mathbf{x})$. Passing to the limit $\lambda \to +\infty$ yields

$$0 = \sigma(\varphi)\mu\left(\Pi_{\mathbf{y},\theta}\right) + \mu\left(H_{\mathbf{y},\theta}\right).$$

Since φ is arbitrary, then $\mu\left(\Pi_{\mathbf{y},\theta}\right) = \mu\left(H_{\mathbf{y},\theta}\right) = 0$.
For a given $\mathbf{y} \in \mathbb{R}^d$, set $F_{\mathbf{y}}(h) = \int_{I_d} h(\langle \mathbf{y}, \mathbf{x}\rangle) d\mu(\mathbf{x})$. Since μ is a finite measure then F is a bounded functional on $L^\infty(I_d)$. From $\mu\left(H_{\mathbf{y},\theta}\right) = 0$, then $F_{\mathbf{y}}(h) = 0$ where h is the indicatrix function of an interval. By density of piecewise constant functions in $L^\infty(I_d)$, one has that $F_{\mathbf{y}}(h) = 0$ for all $\mathbf{y} \in \mathbb{R}^d$ and all $h \in L^\infty(I_d)$. That is $F_{\mathbf{y}} = 0$.
Then

$$\int_{I_d} e^{i\langle \mathbf{k}, \mathbf{x}\rangle} d\mu(x) = 0 \quad \forall \mathbf{k} \in \mathbb{R}^d.$$

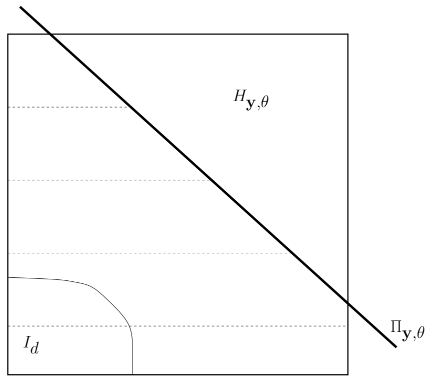

Fig. 2.1: Sketch of the sets $\Pi_{\mathbf{y},\theta}$ and $H_{\mathbf{y},\theta}$.

That is the Fourier transform of the measure is zero for all Fourier coefficients $\mathbf{k} \in \mathbb{R}^d$. So $\mu = 0$ as well [88][p. 176]. $\square$

Soon after the Cybenko, Hornick realized that a constructive proof is possible [45]. Much more powerful results in relation with the representation Theorem of Kolmogorov are possible, see [11] for a recent fully constructive proof.

2.3 Simple constructive proofs

We propose in this Section simple proofs of the Cybenko Theorem which have the advantage that sigmoids and ReLU-based activation functions are treated in an unified framework. As in the Cybenko Theorem, N is proportional to the number of neurons in the hidden layer which is unique.

2.3.1 Approximation in dimension one

Let us take a smooth real function $f^{\mathrm{obj}} \in C_0^1(\mathbb{R})$ with compact support in $I = [-1, 1]$. Note the Heaviside function $H(x) = 1$ for $x > 0$ and $H(x) = 0$ otherwise. One has the formula

$$f^{\mathrm{obj}}(x) = \int_{-\infty}^{x} (f^{\mathrm{obj}})'(y)dy = \int_{\mathbb{R}} (f^{\mathrm{obj}})'(y)H(x-y)dy. \tag{2.2}$$

We desire to consider approximations with sigmoid functions, as in the Cybenko Theorem. So we take one function S which is a sigmoid in the sense of Definition

1.2.5) and we pose $S_\varepsilon(x) = S\left(\frac{x}{\varepsilon}\right)$. Then we modify representation formula (2.2)

$$f_\varepsilon^{\mathrm{obj}}(x) = \int_{\mathbb{R}} (f^{\mathrm{obj}})'(y) S_\varepsilon(x-y)\, dy = \int_{|y|\le 1} (f^{\mathrm{obj}})'(y) S\left(\frac{x-y}{\varepsilon}\right) dy, \quad x \in \mathbb{R}. \tag{2.3}$$

A Riemann integration procedure with step size $0 < h \le 1$ such that $N = 1/h \in \mathbb{N}^*$ yields the function $f_{\varepsilon,N}$

$$f_{\varepsilon,N}(x) = \sum_{|i|\le N} (f^{\mathrm{obj}})'(ih) S\left(\frac{x-ih}{\varepsilon}\right) h. \tag{2.4}$$

Here the number of neurons in the hidden layer is equal to $2N+1$ and the activation function is a sigmoid.

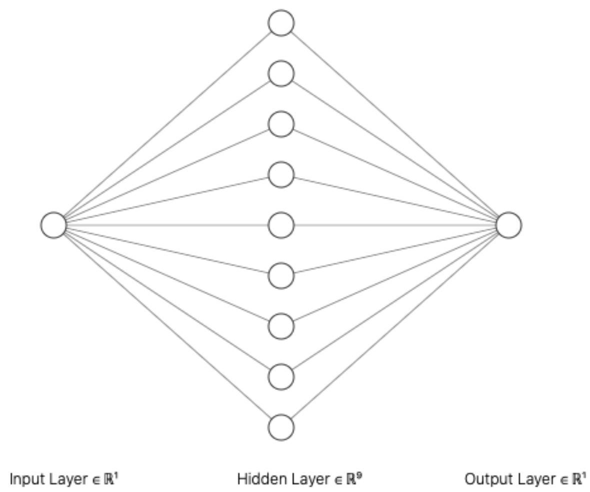

Fig. 2.2: Graph structure of the neural network for function (2.4).

Lemma 2.3.1. *The function $f_{\varepsilon,N}$ can be realized with a neural network with the sigmoid function as activation function and with one hidden layer of $2N+1$ neurons. With the notations (1.19)-(1.25), it corresponds to $(a_0, a_1, a_2) = (1, 2N+1, 1)$.*

Proof. For the hidden layer, take $W_0 = \frac{1}{\varepsilon}(1, \ldots, 1)^t \in \mathcal{M}_{2N+1,1}(\mathbb{R})$ and $b_0 = -W_0$. For the output layer, take $W_1 = \left((f^{\mathrm{obj}})'(ih)\right)_{|i|\le N} \in \mathcal{M}_{1,2N+1}(\mathbb{R})$ and $b_1 = 0$. The activation function is S for the hidden layer. The activation function of output layer is linear (it means no activation function). $\square$

Lemma 2.3.2. *Assume $f^{\text{obj}} \in C_0^0(I)$ and $S \in W^{2,\infty}(\mathbb{R})$ such that $H - S \in L^1(\mathbb{R})$. Assume $\varepsilon \to 0$ and $\varepsilon^2 N \to \infty$. Then*

$$\lim \left\| f^{\text{obj}} - f_{\varepsilon,N} \right\|_{L^\infty(I)} \to 0.$$

Proof. The triangular inequality is

$$\left\| f^{\text{obj}} - f_{\varepsilon,N} \right\|_{L^\infty(I)} \le \left\| f^{\text{obj}} - f_\varepsilon^{\text{obj}} \right\|_{L^\infty(I)} + \left\| f_\varepsilon^{\text{obj}} - f_{\varepsilon,N} \right\|_{L^\infty(I)}.$$

- One has $\left\| f^{\text{obj}} - f_\varepsilon^{\text{obj}} \right\|_{L^\infty(I)} \to 0$ as $\varepsilon \to 0$ by standard uniform continuity estimates on a closed interval. Indeed

$$\left| \left(f^{\text{obj}} - f_\varepsilon^{\text{obj}} \right)(x) \right| \le \| f^{\text{obj}} \|_{L^\infty(\mathbb{R})} \int_{\mathbb{R}} |H(y) - S_\varepsilon(y)|\, dy.$$

But $\int_{\mathbb{R}} |H(y) - S_\varepsilon(y)|\, dy = \int_{\mathbb{R}} |H(y/\varepsilon) - S(y/\varepsilon)|\, dy = \varepsilon \int_{\mathbb{R}} |H(y) - S(y)|\, dy$. One can write

$$\left\| f^{\text{obj}} - f_\varepsilon^{\text{obj}} \right\|_{L^\infty(I)} \le C\varepsilon. \tag{2.5}$$

- Then one has by standard quadrature estimates

$$\left\| f_\varepsilon^{\text{obj}} - f_{\varepsilon,N} \right\|_{L^\infty(I)} \le \| (f_\varepsilon^{\text{obj}})'' \|_{L^\infty(I)} N^{-1}.$$

But

$$f_\varepsilon^{\text{obj}}(x) = \int_{\mathbb{R}} f^{\text{obj}}(y) \frac{d}{dy} S\left(\frac{x-y}{\varepsilon} \right) dy \text{ so } (f_\varepsilon^{\text{obj}})'(x) = \int_{\mathbb{R}} f^{\text{obj}}(y) \frac{d^2}{dy^2} S\left(\frac{x-y}{\varepsilon} \right) dy$$

hence $\| (f_\varepsilon^{\text{obj}})'' \|_{L^\infty(I)} \le \frac{C}{\varepsilon^2}$ and one can write $\left\| f_\varepsilon^{\text{obj}} - f_{\varepsilon,N} \right\|_{L^\infty(I)} \le \frac{C}{\varepsilon^2 N}$.

- Finally $\left\| f_\varepsilon^{\text{obj}} - f_N \right\|_{L^\infty(I)} \le C \left(\varepsilon + \frac{1}{\varepsilon^2 N} \right)$ from which the claim proceeds. ☐

Remark 2.3.3. *A weakness of the result visible in the proof of Lemma 2.3.1 is that W_0 blows up like ε^{-1}, even if W_1 is bounded if $f^{\text{obj}} \in W^{1,\infty}(I)$. Most of the methods which are implemented tend on the contrary to control the magnitude of all weights, refer to Remark 5.5.3. So the theoretical result of the Lemma is not representative of the practice.*

Now we construct another approximation where the sigmoid in (2.4) is changed into the ReLU function. Since $R' = H$ (almost everywhere), an integration by part in (2.2) yields

$$f^{\text{obj}}(x) = \int_{\mathbb{R}} (f^{\text{obj}})''(y) R(x-y) dy$$

which is correct for $f \in C_0^2(\mathbb{R})$. Let us take an integer number $N \in \mathbb{N}^*$. A standard discretization procedure with Riemann quadrature formula with mesh size $h = \frac{1}{N} > 0$ yields the approximation

$$f_N(x) = \sum_{|i|\leq N} (f^{\mathrm{obj}})''(ih)R(x-ih)h. \tag{2.6}$$

The number of neurons in the hidden layer is equal to $2N+1$ as before, but now the activation function is ReLU.

Lemma 2.3.4. *The function f_N can be realized with a neural network with the ReLU function as activation function and with one hidden layer of $2N+1$ neurons with the ReLU function as activation function. As in Lemma 2.3.1, one has $(a_0, a_1, a_2) = (1, 2N+1, 1)$.*

Proof. Similar as in Lemma 2.3.1. Now $W_0 = (1, \dots, 1)^t \in \mathcal{M}_{N,1}(\mathbb{R})$. □

A comparison with the approximation with sigmoids show that the internal weight matrix W_0 is bounded.

Lemma 2.3.5. *Assume $f^{\mathrm{obj}} \in W_0^{\infty,3}(I)$. One has*

$$\left\| f^{\mathrm{obj}} - f_N \right\|_{L^\infty(I)} \leq C \left\| f^{\mathrm{obj}} \right\|_{W^{\infty,3}(I)} N^{-1}. \tag{2.7}$$

Proof. This is the standard error formula for Riemann quadratures. □

Algorithm 4 is used to approximate the function $f^{\mathrm{obj}}(x) = 1 - \cos 2\pi x$ with different number of neurons distributed over juste one hidden layer. The results are displayed in Figure 2.3. The objective function is symmetric left-right, however the approximation does not respect the symmetry. The error of approximation is mostly visible on the right hand side. Doubling the number of neurons decreases the error by a factor 2. It shows first order convergence with respect to the number of neurons. Overall the quality of this approximation is crude.

Next we modify the regularity of the objective function so that it is the same regularity as in Lemma 2.3.2.

Lemma 2.3.6. *Assume $f^{\mathrm{obj}} \in C_0^0(I)$. Then*

$$\lim_{N\to\infty} \left(\inf_{g_N} \left\| f^{\mathrm{obj}} - g_N \right\|_{L^\infty(I)} \right) = 0$$

where the g_N have the form (2.6).

```
Python-Keras-Tensorflow Initialization
N=40; dx=1./N
def f(x):                        return 1-np.cos(6.28*x)
def f_2(x):                      return 6.28*6.28*np.cos(6.28*x)
def init_W0(shape, dtype=None): return K.constant(np.array([np.ones(N)]))
def init_b0(shape, dtype=None): return K.constant(-np.linspace(0,1,N))
def init_W1(shape, dtype=None):
      array3=np.zeros(N)
      for i in range(0,N): array3[i]=f_2(i*dx)*dx
      return K.constant(np.transpose(np.array([array3])))
model = Sequential()
model.add(Dense(N, input_dim=1,name="lay_in",kernel_initializer=init_W0,
             use_bias=True, bias_initializer=init_b0, activation='relu'))
model.add(Dense(1,name="lay_out",kernel_initializer=init_W1,
             use_bias=False,activation='linear'))
x_p=np.linspace(0,1,N); y_p=model.predict(x_p); plt.plot(x_p,y_p,'-+')
```

Algorithm 4: A one-hidden-layer dense NN initialized with formula (2.6).

Proof. Let us consider the regularized function (2.3), still with the bound (2.5)

$$\left\| f^{\text{obj}} - f^{\text{obj}}_\varepsilon \right\|_{L^\infty(I)} \leq C\varepsilon.$$

Consider

$$f^\varepsilon_N(x) = \sum_{|i|\leq N} (f^{\text{obj}}_\varepsilon)''(ih)R(x-ih)h$$

which is a convenient regularization of (2.6). We ask the mollifier to be more regular with two derivatives, that is $\rho \in C^2_0(\mathbb{R})$. Then $S''_\varepsilon = \frac{1}{\varepsilon^3}\rho''\left(\frac{x}{\varepsilon}\right)$, so $\|f_\varepsilon\|_{W^{3,\infty}(I)} \leq C\varepsilon^{-3}$. By Lemma 2.3.5, $\left\| f^{\text{obj}}_\varepsilon - f^\varepsilon_N \right\|_{L^\infty(I)} \leq C\varepsilon^{-3}N^{-1}$. One has

$$\inf_{g_N} \left\| f^{\text{obj}} - g_N \right\|_{L^\infty(I)} \leq \left\| f^{\text{obj}} - f^\varepsilon_N \right\|_{L^\infty(I)} \leq \left\| f^{\text{obj}} - f^{\text{obj}}_\varepsilon \right\|_{L^\infty(I)} + \left\| f^{\text{obj}}_\varepsilon - f^\varepsilon_N \right\|_{L^\infty(I)}$$

so $\inf_{g_N} \left\| f^{\text{obj}} - g_N \right\|_{L^\infty(I)} \leq C\left(\varepsilon + \frac{1}{\varepsilon^3 N}\right)$. To end the proof, take ε small enough, for example $\varepsilon = N^{-\frac{1}{4}}$. So $\inf_{g_N} \left\| f^{\text{obj}} - g_N \right\|_{L^\infty(I)} \leq CN^{-\frac{1}{4}}$. It remains to let $N \to \infty$. □

It is possible the change the ReLU function so that it ressembles more a sigmoid (by using TReLu). We only provide the convenient modification of the integral representation formula (2.2) and other details are left as an exercice.

Lemma 2.3.7. *Assume $f^{\text{obj}} \in C^2_0(\mathbb{R})$. One has $f^{\text{obj}}(x) = \int_{\mathbb{R}} \alpha(y)T(x-y)dy + C$ where $\alpha(x) = \sum_{n\in\mathbb{N}}(f^{\text{obj}})''(x-n)$ and $C \in \mathbb{R}$ is a constant.*

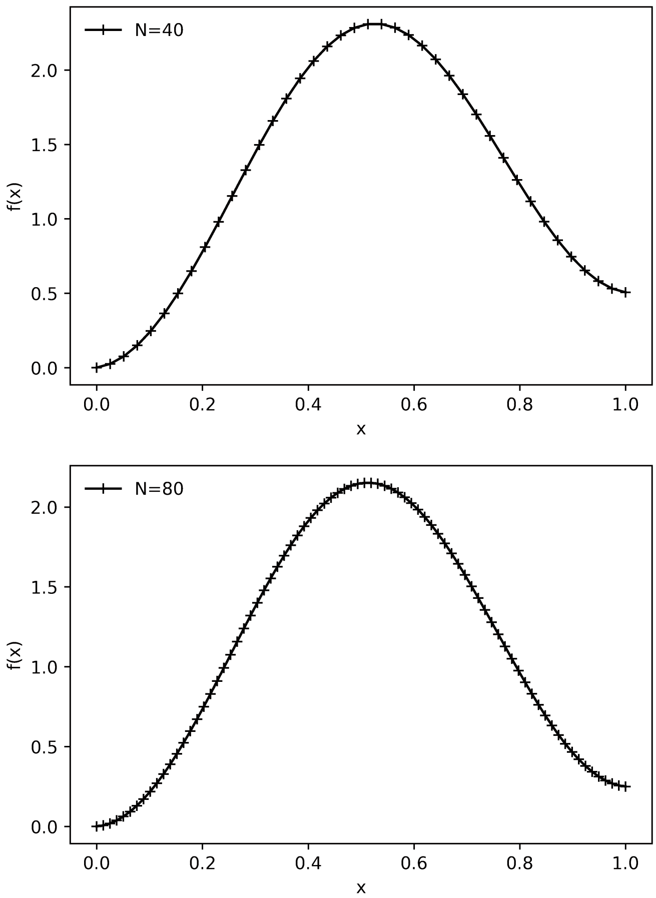

Fig. 2.3: Reconstruction of $f^{\text{obj}}(x) = 1 - \cos 2\pi x$ with Algorithm 3, code available at https://www.ljll.math.upmc.fr/despres/BD_fichiers/ReLU.py. On the left with 40 neurons, on the right with 80 neurons. One notices that the profiles do not have the left-right symmetry. This is due to the fact that the ReLU function is not left-right symmetric. Doubling the number of neurons decreases the error by a factor 2, which is characteristic of first order convergence with respect to the number of neurons.

Proof. Let us consider the function $g(x) = \int_{\mathbb{R}} \alpha(y)T(x-y)dy$. By definition of α, one has $\alpha = \beta'$ with $\beta(x) = \sum_{n\in\mathbb{N}} (f^{\text{obj}})'(x-n)$. One can write

$$g(x)(x) = \int_{\mathbb{R}} \beta'(y)T(x-y)dy = \int_{\mathbb{R}} \beta(y)T'(x-y)dy = \int_{x-1}^{x} \beta(y)dy.$$

Differentiation yields

$$g'(x) = \beta(x) - \beta(x-1) = \sum_{n\in\mathbb{N}} (f^{\text{obj}})'(x-n) - \sum_{n\in\mathbb{N}} (f^{\text{obj}})'(x-1-n) = (f^{\text{obj}})'(x).$$

So g and f^{obj} differ by a constant. $\square$

By using Lemma 1.2.9, it is also possible to generalize formula (2.6) to LReLU instead of ReLU.

2.3.2 Approximation in higher dimensions

Results similar as in the previous Section hold in any dimension. In higher dimension, a generalization of the formula (2.2) is given in the Lemma 2.3.8. Then as in Section 2.3.1, by regularization of the Heaviside functions, integration by parts and Riemann quadrature procedures, one can obtain in any dimensions formulas similar to (2.4) and (2.6).

The quadrature procedure for the integral $\int_{b\in\mathbb{R}} \cdots$ can be taken as the same to (2.4) or (2.6). However the quadrature procedure needed to discretize $\int_{\Omega\in S^d} \cdots$ requires sampling of the hypersphere S^d which is known to be a delicate topic. This is why the discretization of formula (2.8) will not be described in details.

Lemma 2.3.8. *Assume $f \in W^{2q,1}(\mathbb{R}^d)$, $|\mathbf{x}|f \in W^{2q,1}(\mathbb{R}^d)$ and $|\mathbf{x}|^2 f \in W^{2q,1}(\mathbb{R}^d)$. One has the formula*

$$f^{\mathrm{obj}}(\mathbf{x}) = C + \int\limits_{\Omega\in S^d} \int\limits_{b\in\mathbb{R}} w(\Omega, b) H(\Omega\cdot\mathbf{x} - b) d\Omega db \tag{2.8}$$

where $C \in \mathbb{R}$ is a constant, $S^d = \{|\Omega| = 1\} \subset \mathbb{R}^d$ is the unit ball and $w : S^d \times \mathbb{R} \to \mathbb{R}$ is a bounded function given in the proof.

To show (2.8), we will use Fourier techniques [88, 28]. The Fourier representation of f^{obj} writes $f^{\mathrm{obj}}(\mathbf{x}) = \frac{1}{2\pi}\int_{\mathbf{k}\in\mathbb{R}^d} \widehat{f^{\mathrm{obj}}}(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{x}} d\mathbf{k}$, so

$$\nabla f^{\mathrm{obj}}(\mathbf{x}) = \frac{i}{2\pi} \int\limits_{\mathbf{k}\in\mathbb{R}^d} \widehat{f}(\mathbf{k}) \mathbf{k} e^{i\mathbf{k}\cdot\mathbf{x}} d\mathbf{k}. \tag{2.9}$$

One has the formulas $\widehat{f^{\mathrm{obj}}}(\mathbf{k}) = \int_{\mathbf{x}\in\mathbb{R}^d} f^{\mathrm{obj}}(\mathbf{x}) e^{i\mathbf{k}\cdot\mathbf{x}} d\mathbf{x}$ and so

$$(|\mathbf{k}|^2 + 1)^q \widehat{f^{\mathrm{obj}}}(\mathbf{k}) = \int\limits_{\mathbf{x}\in\mathbb{R}^d} (-\Delta + 1)^q f^{\mathrm{obj}}(\mathbf{x}) e^{i\mathbf{k}\cdot\mathbf{x}} d\mathbf{x}.$$

By hypothesis, the integrand is integrable in $L^1(\mathbb{R}^d)$, so one obtains a formula which will be used in the proof of the next result

$$\left|\widehat{f^{\mathrm{obj}}}(\mathbf{k})\right| \le \frac{\|(-\Delta + 1)^q f\|_{L^1(\mathbb{R}^d)}}{(|\mathbf{k}|^2 + 1)^q} \le \frac{C}{(|\mathbf{k}|^2 + 1)^q}. \tag{2.10}$$

Proof of Lemma 2.3.8. The proof is in two steps.

• Our goal is firstly to find the weight function w. One has

$$f^{\text{obj}}(\mathbf{x}) = \int_{\Omega \in S^d} \left(\int_{b<\Omega\cdot\mathbf{x}} w(\Omega, b)db \right) = \int_{\Omega \in S^d} \left(\int_{-\infty}^{\Omega\cdot\mathbf{x}} w(\Omega, b)db \right)$$

where w is unknown at this stage. Differentiate

$$\nabla f^{\text{obj}}(\mathbf{x}) = \int_{\Omega \in S^d} w(\Omega, \Omega \cdot \mathbf{x})\Omega d\Omega.$$

Take w as the half Fourier transform of another unknown function z, that is

$$w(\Omega, a) = \int_{\mu>0} e^{i\mu a} z(\Omega, \mu)d\mu. \tag{2.11}$$

One gets

$$\nabla f^{\text{obj}}(\mathbf{x}) = \int_{\Omega \in S^d} \int_{\mu>0} e^{i\mu\Omega\cdot\mathbf{x}} z(\Omega, \mu)\Omega d\mu d\Omega.$$

Make the change of variable $\mathbf{k} = \Omega\mu$ that is $\mu = |\mathbf{k}|$ and $\Omega = \frac{\mathbf{k}}{|\mathbf{k}|}$. The equivalence of the measures writes $d\mathbf{k} = \mu^{d-1}d\mu d\Omega$. One gets

$$\nabla f^{\text{obj}}(\mathbf{x}) = \int_{\mathbf{k}\in\mathbb{R}^d} \left(\frac{z(\Omega, \mu)}{\mu^d} \right) \mathbf{k} e^{i\mathbf{k}\cdot\mathbf{x}} d\mathbf{k}.$$

Comparison with (2.9) shows a good choice of z is

$$z(\Omega, \mu) = \frac{i}{2\pi} \mu^d \widehat{f}^{\text{obj}}(\mu\Omega).$$

We note that (2.10) yields that $|z(\Omega, \mu)| \leq C\mu^d(\mu^2 + 1)^{-q}$. This function decreases like $O(\mu^{d-2q})$ for large $\mu > 0$.

For $2q - d > 1$, the function w is well defined by (2.11) because the integral is convergent. One has the bound

$$|w(\Omega, a)| \leq C \int_0^\infty \mu^d(\mu^2 + 1)^{-q} d\mu \leq \widetilde{C} < \infty.$$

which is bounded uniformly with respect to a, for all Ω. With this formula, one gets a bounded function w.

• Secondly, it is necessary to check that (2.8) makes sense, that is the integrand is integrable. It is enough to have a good estimate like

$$|w(\Omega, b)| \leq \frac{C}{1 + |b|^s}, \quad s > 1.$$

By d integration by parts on μ, there exists $C>0$ such that

$$|w(\Omega,a)|\le Ca^{-d},\quad a>0.$$

If $d\ge 2$, it is enough to to be sure the formula (2.8) is integrable. In view of (2.11), two integrations by parts with respect to a should be sufficient. It is possible if we have good estimates for

$$z''(\Omega,\mu)=\frac{id(d-1)}{2\pi}\mu^{d-2}\widehat{f}^{\mathrm{obj}}(\mu\Omega)$$
$$+\frac{2id}{2\pi}\mu^{d-1}\left\langle\Omega,\nabla\widehat{f}^{\mathrm{obj}}(\mu\Omega)\right\rangle+\frac{i}{2\pi}\mu^{d}\left\langle\nabla^2\widehat{f}^{\mathrm{obj}}(\mu\Omega)\Omega,\Omega\right\rangle.$$

In view of the assumptions and of the fact that, in Fourier, multiplication by $\mathbf{x}$ corresponds derivation with respect to $\mathbf{k}$, then (2.10) admits the generalization

$$\left|\nabla\widehat{f^{\mathrm{obj}}}(\mathbf{k})\right|\le\frac{\||\mathbf{x}|f^{\mathrm{obj}}\|_{L^1(\mathbb{R}^d)}}{(|\mathbf{k}|^2+1)^q}\ \text{and}\ \left|\nabla^2\widehat{f^{\mathrm{obj}}}(\mathbf{k})\right|\le\frac{\||\mathbf{x}|^2f^{\mathrm{obj}}\|_{L^1(\mathbb{R}^d)}}{(|\mathbf{k}|^2+1)^q}.\tag{2.12}$$

Then, with these hypotheses, (2.8) is correctly defined and the proof is ended. □

2.4 Connection with Finite Element Theory

We state known results which are that basic Finite Elements functions belong to the ensemble of functions generated by ReLU activation functions.

2.4.1 1D piecewise affine finite elements

Take a possibly non uniform mesh in dimension one

$$\dots<x_{j-1}<x_j<x_{j+1}<\dots\qquad\text{with}\ \lim_{j\to\pm\infty}x_j=\pm\infty.$$

We note the local mesh size $h_{j+\frac12}=x_{j+1}-x_j$. The global mesh size is defined by $h^+=\sup_{j\in\mathbb{Z}}(h_{j+\frac12})<\infty$. We make the assumption that the local mesh size is bounded from below, that is

$$h^-=\inf_{j\in\mathbb{Z}}(h_{j+\frac12})\ \text{is such that}\ h^->0.$$

Of course $h^-\le h^+$. Such notions are standard in the theory of Finite Element [1].

Definition 2.4.1. *The Finite Element P_1 function φ_j is*

$$\varphi_j(x) = \begin{cases} 0 & \text{for } x \le x_{j-1}, \\ \frac{x-x_{j-1}}{h_{j-\frac{1}{2}}} & \text{for } x_{j-1} \le x \le x_j \\ \frac{x_{j+1}-x}{h_{j+\frac{1}{2}}} & \text{for } x_j \le x \le x_{j+1} \\ 0 & \text{for } x_{j+1} \le x. \end{cases}$$

It has a compact support in $[x_{j-1}, x_{j+1}]$.

The family of basis functions $\left(\varphi_j\right)_{j\in\mathbb{Z}}$ is a basis of a discrete finite element space. We refer the reader to [1]. Our aim is to reinterpret the basis functions in terms of the ReLU function and alike.

Lemma 2.4.2. *One has the three different representations*

$$\begin{cases} \varphi_j(x) = R\left(\frac{1}{h_{j-\frac{1}{2}}}R(x-x_{j-1}) - \left(\frac{1}{h_{j-\frac{1}{2}}} + \frac{1}{h_{j+\frac{1}{2}}}\right)R(x-x_j)\right), \\ \varphi_j(x) = R\left(\frac{1}{h_{j+\frac{1}{2}}}R(x_{j+1}-x) - \left(\frac{1}{h_{j-\frac{1}{2}}} + \frac{1}{h_{j+\frac{1}{2}}}\right)R(x-x_j)\right), \\ \varphi_j(x) = T\left(T\left(\frac{x-x_{j-1}}{h_{j-\frac{1}{2}}}\right) + T\left(\frac{x_{j+1}-x}{h_{j+\frac{1}{2}}}\right) - 1\right). \end{cases} \tag{2.13}$$

Proof. Evident. The second formula is the symmetrized version of the first one. The third representation is symmetric left-right. □

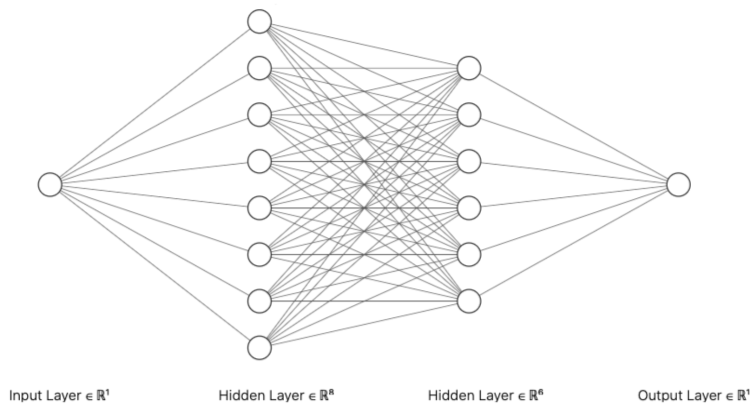

Fig. 2.4: Graph of the network which implements (2.14). The first hidden layer is two more neurons than the second hidden layer.

One can approximate a given function objective function $f^{\mathrm{obj}} \in C^0(\mathbb{R})$ with the well known Finite Element interpolation formula

$$f_h(x) = \sum_{j\in\mathbb{Z}} f^{\mathrm{obj}}(x_j)\varphi_j(x). \tag{2.14}$$

The objective function does not have necessarily a compact support for (2.14) to make sense (as it was the case in approximations (2.4) or (2.6)). The reason is that it is the basis functions φ_j's which have a compact support now. Nevertheless the next statement needs that f^{obj} has compact support.

Lemma 2.4.3. *Assume that $f^{\mathrm{obj}} \in C_0^0(\mathbb{R})$ has compact support in $I = [-1,1]$. The function f_h can be realized with a neural network with the ReLU function as activation function and with two hidden layers of $2/h^- + 1$ neurons. With the notations (1.19)-(1.25), it corresponds to $(a_0, a_1, a_2, a_3) = \big(1, 2/h^- + 1, 2/h^- + 1, 1\big)$.*

Proof. The number of $x_j \in [-1,1]$ is less or equal to $2/h^- + 1$. So the sum in formula (2.14) is restricted to these mesh points. The two hidden layers are used to calculate (any of) the Finite Element functions (2.13). □

Lemma 2.4.4. *One has $f(x_j) = f^{\mathrm{obj}}(x_j)$ for all $j \in \mathbb{Z}$. Assume moreover that $f^{\mathrm{obj}} \in W_0^2(\mathbb{R})$ with compact support in I, then*

$$\|f^{\mathrm{obj}} - f\|_{L^\infty(\mathbb{R})} \le \frac{1}{8}\sup_{j\in\mathbb{Z}}\left(\|(f^{\mathrm{obj}})''\|_{L^\infty(x_j,x_{j+1})} h_{j+\frac12}^2\right). \tag{2.15}$$

Proof. The function f is the P_1 Lagrange interpolation of f^{obj}, and the first part of the claim is immediate from (2.14). The second part is classical for Lagrange interpolation. For a given x such that $x_j < x < x_{j+1}$, take

$$\mu = \frac{f^{\mathrm{obj}}(x) - f(x)}{\frac{(x-x_j)(x-x_{j+1})}{2}}.$$

and consider

$$g(y) = f^{\mathrm{obj}}(y) - f(y) - \mu\frac{(y-x_j)(y-x_{j+1})}{2}.$$

By construction $g(x_j) = g(x_{j+1}) = g(x) = 0$ and $g''(y) = (f^{\mathrm{obj}})''(y) - \mu$. Since g vanishes at three different points, the Role Theorem used two times yields that there exists $y \in (x_j, x_{j+1})$ such that $g''(y) = 0$. So one can write

$$f^{\mathrm{obj}}(x) - f(x) = (f^{\mathrm{obj}})''(y)\frac{(x-x_j)(x-x_{j+1})}{2}.$$

The result then follows from $\big|(x-x_j)(x-x_{j+1})\big| \le \frac{h_{j+\frac12}^2}{4}$. □

```
Python-Keras-Tensorflow Initialization
N=40; dx=1./(N-1)
def f(x): return 1-np.cos(6.28*x)
def init_W0(shape,dtype=None):  return K.constant([np.ones(N)])
def init_b0(shape,dtype=None):  return K.constant(-np.linspace(0,1,N))
def init_W1(shape,dtype=None):  W=np.zeros((N,N));
    for i in range(0,N): W[i,i]= -2/dx
    for i in range(1,N): W[i-1,i]=1/dx
    return K.constant(W)
def init_W2(shape, dtype=None): array3=np.zeros(N)
    for i in range(0,N): array3[i]=f(i*dx)
    return K.constant(np.transpose(np.array([array3])))
model = Sequential()
model.add(Dense(N,input_dim=1,name="lay_in",kernel_initializer=init_W0,
              use_bias=True,bias_initializer=init_b0,activation='relu'))
model.add(Dense(N,input_dim=1,name="lay_hid",kernel_initializer=init_W1,
              use_bias=False,activation='relu'))
model.add(Dense(1,name="lay_out",kernel_initializer=init_W2,
              use_bias=False,activation='linear'))
x_p=np.linspace(0,1,N); y_p=model.predict(x_p); plt.plot(x_p,y_p,'-+')
```

Algorithm 5: A two-hidden-layer dense NN with initialization given by the Finite Element formula (2.14).

Algorithm 5 details an implementation of the Finite Element representation formula (2.14) for the reconstruction of the same function as in Algorithm 4. The numbers of hidden layers is 2. In Figure 2.5 one observes a gain in accuracy with respect to Figure 2.3. This is of course in accordance with the second-order theoretical prediction (2.15) which is more accurate than (2.7).

A natural question is to obtain a spacing of the mesh points so that the error is minimized. Not only this is a natural question from the viewpoint of the theory of approximation, but also one may think that this is what the optimization session of ML softwares (this is called the training) is all about. So let us consider a certain level of accuracy $\varepsilon > 0$, and let us try to obtain the bound $\|f^{\mathrm{obj}} - f\|_{L^\infty(\mathbb{R})} \le \varepsilon$. In view of Lemma 2.4.4, it is sufficient to satisfy

$$\frac{1}{8}\left(\|(f^{\mathrm{obj}})''\|_{L^\infty(x_j,x_{j+1})} h_{j+\frac12}^2\right) \le \varepsilon \text{ for all } j \in \mathbb{Z}.$$

One obtains

$$h_{j+\frac12} \le \sqrt{\frac{8\varepsilon}{\|(f^{\mathrm{obj}})''\|_{L^\infty(x_j,x_{j+1})}}} \text{ for all } j \in \mathbb{Z}.$$

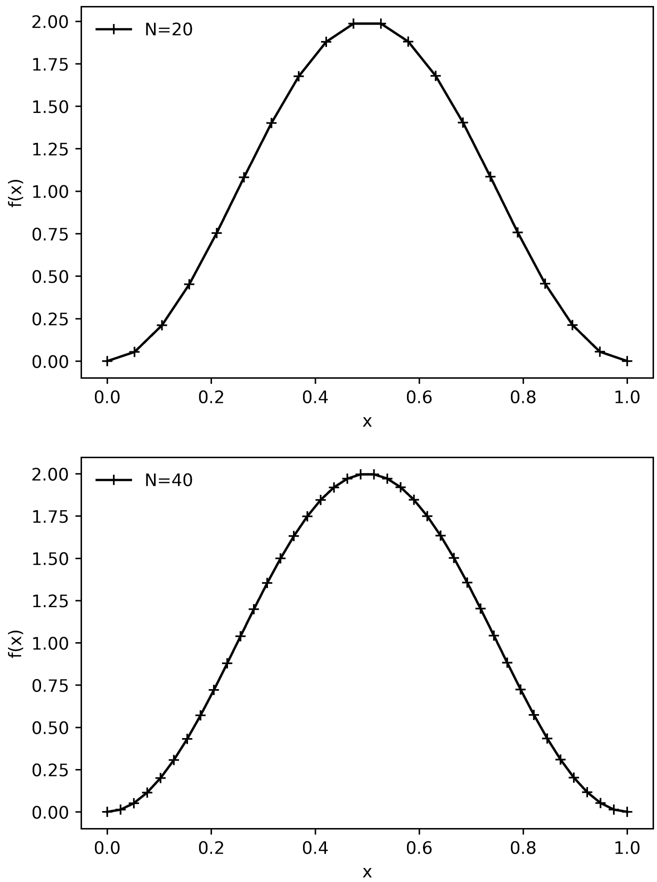

Fig. 2.5: Reconstruction of $f^{\mathrm{obj}}(x) = 1 - \cos 2\pi x$ with Algorithm 4. On the left with 20 neurons, on the right with 40 neurons. The results have the left-right symmetry.

2.4.2 2D piecewise affine Finite Elements

One starts with a triangular mesh of a closed polygon domain $\Omega \subset \mathbb{R}^2$

$$\Omega = \overline{\bigcup_{r \in \mathbb{Z}} T_r}$$

where all T_r's are disjoint open triangles such that $T_r \bigcap T_s = \emptyset$ for $r \neq s$. The index $0 < h < \infty$ will refer to an upper bound on the diameters of the triangle, that is $\mathrm{diam}(T_r) \leq h$ for all r. We consider that the number of triangles $N_h = \#\{T_r\}$ is finite, that is the triangulation is

$$\mathcal{T}_h = \{T_r\}_{r=1}^{N_h}.$$

The nodes $\mathbf{x}_j$'s are the summit of the triangles. A triangle is defined by 3 nodes. The list of all nodes is denoted as

$$\mathcal{N}_h = \{\mathbf{x}_j\}_j.$$