Chapter 1

1 Curse of Dimensionality and Neural Networks

We approach this chapter knowing the following

- **Universal approximation theorem**: Both shallow and deep networks are universal, that is they can approximate arbitrarily well any continuous function of n variables on a compact domain. [**?**].

  This is a classical result for shallow networks, and the same result holds for deep networks. Since $\{\text{Shallow Networks}\} \subseteq \{\text{Deep Networks}\}$, there too exists deep networks that can approximate any continuous function of n variables on a compact domain arbitrarily well.

- **Deep networks can beat the curse**: Theory on shallow networks is well understood and dates well back to at least the 1980s. However, the theory on deep networks is less well understood, for a review we can refer to [**?**]. Within this review we see results on how deep convolutional nets and recurrent nets have been able to beat the curse, in tasks such as image processing and speech recognition. We see that deep learning outperforms traditional machine-learning methods by automatically discovering intricate structures in high-dimensional data. Perhaps by use of hierarchical architectures, deep networks can exploit some form of structure in the data to beat the curse.

- ***"Are hierarchical architectures with more layers justifiable in terms of learning theory?"*** [**?**]

  We find that in many cases in past works, little attention was given to multi-layer networks as often shallow nets performed empirically as well [**?**]. However, [**?**] motivates a strong question regarding hierarchies; *"It seems that the learning theory of the type we have outlined does not offer any general argument in favor of hierarchical learning machines for regression or classification. This is somewhat of a puzzle since the organization of cortex - for instance visual cortex - is strongly hierarchical. At the same time, hierarchical learning systems show superior performance in several engineering applications."*. So there are occasions where deep networks can outperform shallow networks, and we would expect this to be the case by comparison to a neural network we already see in nature - the human brain. The question is now, that if we know such deep networks exist and can outperform shallow networks, can we justify this in terms of learning theory, and how do we construct them?

1.1 Shallow and Deep Networks

Definition 1.1 (Shallow Neural Network (Single Hidden Layer))**.** A shallow neural network with a single hidden layer is defined by the following components:

- **Input Layer:** Let $\mathbf{x} = (x_1, x_2, \dots, x_d)^T$ represent the input vector where d denotes the number of input features/dimensions, thus $\mathbf{x} \in \mathbb{R}^d$.

- **Hidden Layer:** The hidden layer consists of h neurons. Each neuron j applies a transformation to the input vector, characterized by a weight vector $\mathbf{w}_j \in \mathbb{R}^d$ and a scalar bias b_j. The activation function σ is applied to the linear combination of the inputs:

  $$z_j = \sigma(\mathbf{w}_j^T \mathbf{x} + b_j)$$

  The output of the hidden layer can be expressed as $\mathbf{z} = (z_1, z_2, \dots, z_h)^T \in \mathbb{R}^h$.

- **Output Layer:** Assuming a single output neuron, the final output y of the network is produced by applying another activation function ϕ to the linear combination of the hidden layer outputs:

$$y = \phi(\mathbf{v}^T \mathbf{z} + c)$$

where $\mathbf{v} \in \mathbb{R}^h$ and c is a scalar bias.

Definition 1.2 (Deep Neural Network)**.** A **deep neural network** is a function $f : \mathbb{R}^n \to \mathbb{R}^m$ constructed by a sequence of compositions of simpler functions, each representing a layer in the network. The architecture of a deep neural network can be described by a directed acyclic graph (DAG), denoted as $G = (V, E)$, where:

- V represents the set of vertices, with each vertex corresponding to a layer in the network.
- E represents the set of edges, where each directed edge $(u, v) \in E$ indicates that the output of layer u is the input to layer v.

Each vertex $v \in V$ in the graph is associated with a transformation function $f_v : \mathbb{R}^{n_v} \to \mathbb{R}^{m_v}$ and is characterized by the following components:

1. **Neurons**: Each layer v consists of m_v neurons. The number of neurons m_v in each layer can vary depending on the layer's role and design within the network.
2. **Activation Function**: Each layer includes an activation function $\sigma_v : \mathbb{R}^{m_v} \to \mathbb{R}^{m_v}$, which introduces non-linearities into the network, enabling it to model complex relationships.
3. **Weights and Biases**: Each connection $(u, v) \in E$ is characterized by a weight matrix $W_{uv} \in \mathbb{R}^{m_u \times n_v}$ and a bias vector $b_{uv} \in \mathbb{R}^{n_v}$, which are parameters that are learned during training.

The output of each layer v is given by:

$$y_v = \sigma_v(W_{uv} y_u + b_{uv})$$

where y_u is the output from the previous layer u. For the initial layer v_0 in the graph, the input $x \in \mathbb{R}^n$ is used:

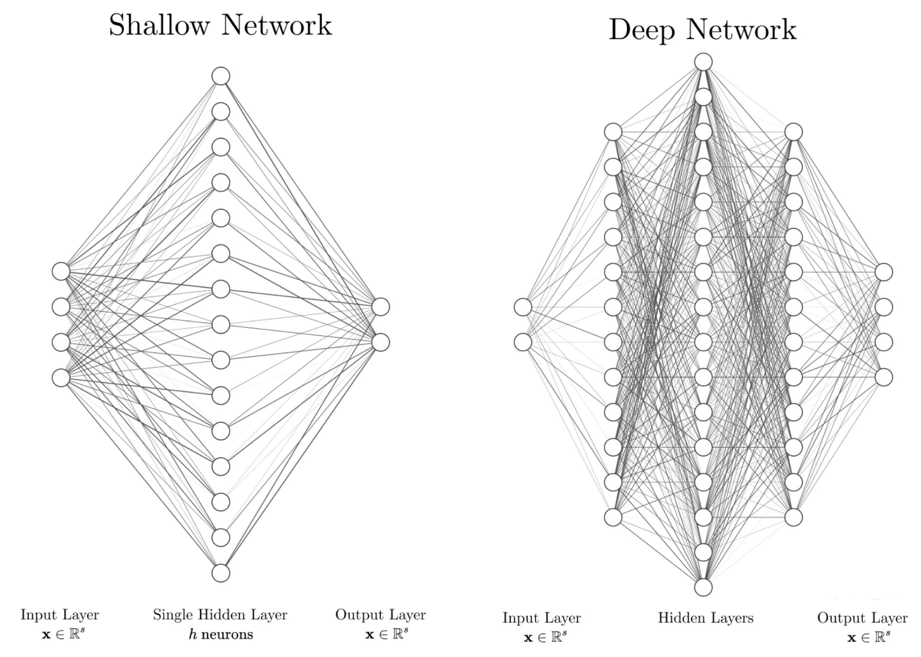
$$y_{v_0} = \sigma_{v_0}(W_{v_0} x + b_{v_0})$$

The overall output of the network $f(x)$ is the output of the final layer in the topology of G.

1.2 Examples of Deep Networks beating the curse

Deep neural networks have been applied successfully to a range of problems that involve high-dimensional data, showcasing their ability to beat the curse of dimensionality. Here are a few prominent examples:

- ResNet-152: A deep residual network with 152 layers, which has been shown to achieve state-of-the-art performance on the ImageNet dataset. [?]
- **Image Recognition**: In the realm of image recognition, Convolutional Neural Networks (CNNs) have become the standard. High-dimensional image data, which can include millions of pixels, is effectively managed by CNNs through hierarchical feature extraction and weight sharing. A landmark moment was when AlexNet dramatically improved performance on the ImageNet challenge, processing images with high resolution to accurately classify them into thousands of categories.


Shallow Network
Deep Network
Input Layer
$\mathbf{x} \in \mathbb{R}^s$
Single Hidden Layer
h neurons
Output Layer
$\mathbf{x} \in \mathbb{R}^s$
Input Layer
$\mathbf{x} \in \mathbb{R}^s$
Hidden Layers
Output Layer
$\mathbf{x} \in \mathbb{R}^s$

- **Natural Language Processing (NLP)**: Models like BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer) handle the high dimensionality of linguistic data, which includes extensive vocabularies and diverse syntactic structures. These models use attention mechanisms and embeddings to reduce dimensionality and capture semantic relationships within text, enabling them to excel in tasks like translation, text summarization, and sentiment analysis.
- **Speech Recognition**: Deep neural networks, specifically deep belief networks (DBNs) and recurrent neural networks (RNNs), have significantly improved the performance of speech recognition systems. These networks process raw audio data, which is inherently high-dimensional, to extract features and recognize speech patterns over time, even in noisy environments.
- **Medical Imaging**: Deep learning has also made substantial impacts in the analysis of medical images, such as MRI scans and CT scans, which are high-dimensional. Networks like U-Net have been specifically designed for biomedical image segmentation, efficiently processing these images to detect, for example, tumor boundaries against the backdrop of normal anatomical structures.
- **Financial Time Series Forecasting**: Despite the challenges posed by the noisy and non-stationary nature of financial markets, deep learning models like Long Short-Term Memory networks (LSTMs) have been used to forecast market movements based on high-dimensional input features, including historical price data, volume, and macroeconomic indicators.
- **Autonomous Vehicles**: The perception systems of autonomous vehicles are another area where deep neural networks manage high-dimensional sensory inputs, including LIDAR, radar, and video. These systems must process and interpret vast amounts of data in real-time to make driving decisions, an application where CNNs and RNNs have proven particularly effective.

These examples demonstrate how deep neural networks can effectively manage and exploit high-dimensional data across different domains, providing solutions where traditional methods fall short.

1.3 Why Sobolev Spaces?

In theoretical investigations of the degree of approximation, one typically makes an a priori assumption that the target function f, although itself unknown, belongs to some known class of functions.

In this report, we are interested in the Sobolev classes. By assuming our target function belongs to some smoothness class, we are enabled to make comments on the required complexity of a network to achieve a desired accuracy. This assumption is what gives us the theoretical guarantees we seek.

We can also leverage these assumptions in the training process, where we have seen Sobolev training in action already in the literature. [**?**], making use of more data than just input-output data pairs to train a network. We explore this later.

Our main method of improving upon the efficiency of a shallow network by use of deep networks will be via architecture solutions - assumptions on the smoothness and compositionality of the target function will be key in this process.

1.4 Main theorems

This section focuses on the results from Poggio, T. and Liao, Q. (2018) 'Theory I: Deep networks and the curse of dimensionality'. [**?**].

We take Π_n to denote the set all neural networks with complexity n, taken here as the total number of units in the network (e.g. shallow networks of n neurons in its hidden layer). Trivially we have $\Pi_n \subseteq \Pi_{n+1}$. We measure the degree of approximation error as:

$$\text{dist}(f - \Pi_n) = \inf_{g \in \Pi_n} \|f - g\|$$

The norm more classically associated for measuring the performance of a neural network is the L^2 norm, however the *sup* norm is used here to provide a stronger condition for the approximation error.

Note that if $dist(f, \Pi_n) = \mathcal{O}(n^{-\gamma})$, for some $\gamma > 0$ then a network of complexity $n = \mathcal{O}(\epsilon^{-\frac{1}{\gamma}})$ can guarantee an approximation error of at least ϵ.

Within this paper, the authors state two theorems, which are as follows:

1. **Theorem 1**: (Shallow Networks)

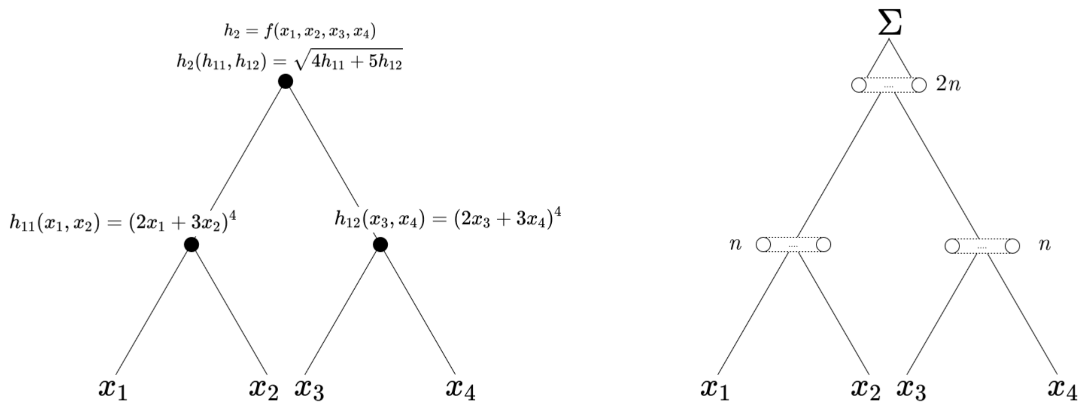
   It states that for a function $f \in W_m^n$, the set of all functions of n variables with continuous partial derivatives up to order $m < \infty$ such that $\|f\| + \sum_{1 \leq |\mathbf{k}|_1 \leq m} \left\|\mathbf{D}^k f\right\| \leq 1$, and an activation function $\sigma : \mathbb{R} \to \mathbb{R}$ that is infinitely differentiable and not a polynomial, the complexity of shallow networks that provide accuracy at least ε is

   $$N = O(\varepsilon^{-n/m})$$

2. **Theorem 2**: (Deep Networks - Binary Tree)

   It considers a function $f \in W_m^{2,n}$, the class of all compositional functions f of n variables with a binary tree architecture and constituent functions h in W_m^2 and a deep network with a compositional architecture.
   The activation function $\sigma : \mathbb{R} \to \mathbb{R}$ in this context is also infinitely differentiable and not a polynomial. The complexity of the network necessary to provide an approximation with accuracy at least ε is

$$N = O((n-1)\varepsilon^{-2/m})$$



$h_2 = f(x_1, x_2, x_3, x_4)$
$h_2(h_{11}, h_{12}) = \sqrt{4h_{11} + 5h_{12}}$
$h_{11}(x_1, x_2) = (2x_1 + 3x_2)^4$
$h_{12}(x_3, x_4) = (2x_3 + 3x_4)^4$
x_1 x_2 x_3 x_4
Σ
$2n$
n
n
x_1 x_2 x_3 x_4

Figure 1: Binary Tree Architecture

1.5 Spaces & Norms

We first take care to mention the spaces and norms we are using. If we intend to measure the quality of an approximation, we must carefully consider how to do this and how the use of various norms interact with one another, and what that implicates for our results. In particular we use 2 types of norms:

1.5.1 The L^p-norm ($\|\cdot\|_p$)

This norm is used to measure the approximation error between the function f and our approximation.

If $A \subseteq R^s$ is Lebesgue measurable, and $f : A \to R$ is a measurable function, we define the $L^p(A)$ norms of f as follows:

$$\|f\|_{p,A} = \begin{cases} \left(\int_A |f(x)|^p dx\right)^{1/p}, & \text{for } 1 \le p < \infty, \\ \text{ess sup}_{x\in A}|f(x)|, & \text{for } p = \infty. \end{cases}$$

The class of all functions such that $\|f\|_{p,A} < \infty$ is denoted by $L^p(A)$

Take $L^\infty(A)$ as the space of continuous functions on A.

Throughout this report we take $A = [-1,1]^s$, unless explicitly stated otherwise.

1.5.2 Sobolev Norm ($\|\cdot\|_{W^p_{r,s}}$)

We first define the space as follows:

Let $r \ge 1$ an integer and Q a cube in $\mathbb{R}^s$. The class $W^p_{r,s}(Q)$ consists of all functions with $r-1$ continuous partial derivatives on Q which in turn can be expressed (almost everywhere on Q) as indefinite integrals of functions in $L^p(Q)$. Alternatively, the class $W^p_{r,s}(Q)$ consists of functions which have, at almost all points of Q, all partial derivatives up to order r such that all of these derivatives are in $L^p(Q)$.

The Sobolev norm of $f \in W^p_{r,s}(Q)$ is defined by

$$\|f\|_{W^p_{r,s}(Q)} = \sum_{0\le|k|\le r} \left\|D^{\mathbf{k}} f\right\|_{p,Q}$$

where for the multi-integer $\mathbf{k} = (k_1, ..., k_s) \in \mathbb{Z}^s$, $0 \le \mathbf{k} \le r$ means that each component of $\mathbf{k}$ is nonnegative and does not exceed r, $|\mathbf{k}| := \sum_{j=1}^{s} |k_j|$ and

$$D^{\mathbf{k}} f = \frac{\partial^{|\mathbf{k}|} f}{\partial x_1^{k_1} \cdots \partial x_s^{k_s}}, \quad k \ge 0$$

Again $W_{r,s}^{\infty}(Q)$ will denote the class of functions which have continuous derivatives of order r and lower. As before, if $Q = [-1, 1]^s$, we will not mention it in the notation. Thus, we write $W_{r,s}^{p} = W_{r,s}^{p}([-1, 1]^s)$ etc.

2 Theorem 1

We state the theorem and proof from [?] as follows;

Let $1 \le d \le s, n \ge 1$ be integers, $f : \mathbb{R}^s \to \mathbb{R}$ and $\phi : \mathbb{R}^d \to \mathbb{R}$. A *generalized translation network* with n neurons evaluates a function of the form $\sum_{k=1}^{n} a_k \phi(A_k(\cdot) + \mathbf{b}_k)$ where the *weights* A_ks are $d \times s$ real matrices, the *thresholds* $\mathbf{b}_k \in \mathbb{R}^d$ and the *coefficients* $a_k \in \mathbb{R}$ $(1 \le k \le n)$. The set of all such functions (with a fixed n) will be denoted by $\Pi_{\phi;n,s}$.

We attempt to approximate f by elements of $\Pi_{\phi;n,s}$ on the domain $[-1, 1]^s$. For the $d = 1$ case we have $\Pi_{\phi;n,s}$ simply the collection of shallow networks with n neurons in their hidden layer.

This construction will not provide the best approximation, but will instead provide the optimal order of approximation. We have also, that the weights A_k's and the thresholds $\mathbf{b}_k$'s will be determined independently of the target function f.

We observe a notable assumption that we are able to sample from the target function at all prescribed points without noise.

2.1 Statement

We can make the following statement for the required complexity of a shallow network to approximate a function in a Sobolev space.

Let $1 \le d \le s$, $r \ge 1$, $n \ge 1$ be integers and $1 \le p \le \infty$. Let $\phi : \mathbb{R}^d \to \mathbb{R}$ be infinitely many times continuously differentiable in some open sphere in $\mathbb{R}^d$. It is further assumed that there exists a point b in this sphere such that

$$D^k \phi(b) \ne 0, \mathbf{k} \in \mathbb{Z}^d, \mathbf{k} \ge 0 \tag{1}$$

Note that the above condition, is equivalent to the statement that the function ϕ is not a polynomial. This is shown in [?].

Then, there exist $d \times s$ matrices $\{A_j\}_{j=1}^{n}$ with the following property: For any function f in the Sobolev space $W_{r,s}^{p}$, there exist coefficients $a_j(f)$ such that

$$\left\| f - \sum_{j=1}^{n} a_j(f) \phi(A_j(\cdot) + b_j) \right\|_p \le c n^{-r/s} \|f\|_{W_{r,s}^{p}} \tag{2}$$

Here, the functionals a_j are continuous linear functionals on $W_{r,s}^{p}$.

In particular, we measure the *degree of approximation* of f by the expression

$$E_{\phi;n,p} = \inf\{\|f - g\|_p : g \in \Pi_{\phi;n,s}\}$$

The quantity $E_{\phi;n,p}$ denotes the theoretically minimal error that can be achieved in approximating the function f in the L^p norm by generalized translation networks with n neurons each evaluating the activation function ϕ.

$$E_{\phi;n,p,r,s} := \sup\{E_{\phi;n,p}(f) : \|f\|_{W^p_{r,s}} \leq 1\} \leq cn^{-r/s}$$

where $E_{\phi;n,r,s,p}$ denotes the maximal error that can be achieved in approximating functions in $W^p_{r,s}$ by generalized translation networks with n neurons each evaluating the activation function ϕ, with the assumption that the target function $f \in W^p_{r,s}$ is properly normalised. Here we take the fact that that any function in $W^p_{r,s}$ can be normalized so that $\|f\|_{W^p_{r,s}} \leq 1$.

2.2 Proof

2.2.1 Idea for Proof

For every integer $m \geq r, \exists$ polynomial $P_m(f)$ of co-ordinatewise degree not exceeding m such that $\forall f \in W^p_{r,s}$.

$$\|f - P_m(f)\|_{p,[-\pi,\pi]^s} \leq \frac{c}{m^r}\|f\|_{W^{p*}_{r,s}}$$

(By co-ordinatewise degree, we mean that the degree of the polynomial in each variable is at most m). Aim to express each monomial in $P_m(f)$ with suitable derivative of ϕ, then take each derivative approximations and approximate it by an appropriate divided difference method involving $O(m^s)$ evaluations of ϕ

2.2.2 Useful Definitions and clarifications

- $|\mathbf{k}| = \sum_j k_j$
- $0 \leq \mathbf{k} \leq \mathbf{r} \iff 0 \leq k_j \leq r_j \forall j$
- $L^{p^*} := L^p([-\pi,\pi]^s)$, $W^{p^*}_{r,s} := W^p_{r,s}([-\pi,\pi]^s)$
- **Fourier Coefficients**: For a function $g \in L^{p^*} = L^p([-\pi,\pi]^s)$, its Fourier coefficients are given by the following, where $\mathbf{k}$ is a multi-index in $\mathbb{Z}^s$, and $\mathbf{t}$ is in the domain $[-\pi,\pi]^s$.

$$g(\mathbf{k}) := \frac{1}{(2\pi)^s}\int_{[-\pi,\pi]^s} g(\mathbf{t})e^{-i\mathbf{kt}}d\mathbf{t}, \quad \mathbf{k} \in \mathbb{Z}^s. \tag{3}$$

- **Partial Sums of the Fourier Series**: The partial sum $s_m(g,t)$ of the Fourier series of g is given as follows,

$$s_m(g(\mathbf{k}),t) := \sum_{-\mathbf{m}\leq\mathbf{k}<\mathbf{m}} g(\mathbf{k})e^{i\mathbf{kt}}, \quad \mathbf{m} \in \mathbb{Z}^s, \quad \mathbf{m} \geq 0, \quad \mathbf{t} \in [-\pi,\pi]^s, \tag{4}$$

- **de la Vallée Poussin Operator**: The de la Vallée Poussin operator $v_n(g,t)$ for a function g is defined as the average of the partial sums $s_m(g,t)$ where m ranges from n to $2n$. This operator is used to create a smoothed approximation of the function g.

$$v_n(g,\mathbf{t}) := \frac{1}{(n+1)^s}\sum_{n\leq\mathbf{m}\leq 2n} s_{\mathbf{m}}(g,t), \quad n \in \mathbb{Z}, \quad n \geq 0, \quad \mathbf{t} \in [-\pi,\pi]^s, \tag{5}$$

We occasionally omit the $\mathbf{k}$ argument of $g(\mathbf{k})$ for brevity.

Proposition 2.1 ([?]). *We have the following result:*

- *Given integers $r \geq 1$, s, $m \geq 1$, and $1 \leq p \leq \infty$.*
- *Let g be a function in the Sobolev space $W^{p^*}_{r,s}$.*

- *Then $v_m(g)$ is defined as a trigonometric polynomial of coordinate-wise order at most $2m$. This means that $v_m(g)$ is a polynomial composed of sine and cosine terms, where the highest frequency term has a frequency of $2m$.*

  Here $v_m(g)$ is defined as the de la Vallée Poussin operator applied to g, we omit the t dependence for brevity. $v_m(g) \equiv v_m(g(\mathbf{k}), \mathbf{t}) \equiv v_m(g, \mathbf{t})(\mathbf{k})$

- *And*

$$\|g - v_m(g)\|_{p,[-\pi,\pi]^s} \leq \frac{c}{m^r}\|g\|_{W^{p*}_{r,s}} \tag{6}$$

- *further,*

$$\sum_{0 \leq \mathbf{k} < 2m} |\hat{v_m}(g)(\mathbf{k})| \leq cm^{\alpha}\|g\|_{W^{p*}_{r,s}} \quad \textit{where} \quad \alpha = \frac{s}{\min(p,2)} \tag{7}$$

The idea follows to make a periodic function from a function on $[-1,1]^s$. The standard way to achieve this is via a cosine substitution; $x_j = \cos(t_j) 1 \leq j \leq s, \mathbf{x} \in [-1,1]^s, \mathbf{t} \in [-\pi,\pi]^s$. This substitution doesn't work as the L^p norms are no longer equal under this substitution.

From [**?**], we instead construct the following;$T : W^p_{r,s} \to W^p_{r,s}([-2,2]^s)$, such that the restriction of $T(f)$ on $[-1,1]^s$ is (almost everywhere) equal to f. The continuity of the operator T means that

$$\|T(f)\|_{W^p_{r,s}([-2,2]^s)} \leq c\|f\|_{W^p_{r,s}}$$

for every $f \in W^p_{r,s}$, This means that the operator T maps a function in the Sobolev space $W^p_{r,s}$ to a function in the same space but defined on a larger domain, $[-2,2]^s$, we construct this operator later.

If we find in practice we have an f already appropriately defined on $[-2,2]^s$, then we can simply work with f itself rather than $T(f)$, however our bounds would then depend on the respective Sobolev space; $W^p_{r,s}([-2,2]^s)$

We now define ψ be an infinitely differentiable function that takes the value 1 on $[-1,1]^s$ and 0 outside the interval $[-3/2,3/2]^s$. Then the function $T(f)\psi$ coincides with f on $[-1,1]^s$ almost everywhere, and is identically zero outside $[-3/2,3/2]^s$ and

$$\|T(f)\psi\|_{W^p_{r,s}([-2,2]^s)} \leq c\|f\|_{W^p_{r,s}} \tag{8}$$

We relabel $T(f)\psi$ as f for brevity. We then define a 2π-periodic function f^* from the function f (extended as above) using the transformation $x_j = 2\cos(t_j) 1 \leq j \leq s, \mathbf{x} \in [-1,1]^s, \mathbf{t} \in [-\pi,\pi]^s$.

$$f^*(\mathbf{t}) = f(\mathbf{x}) = f(2\cos(t_1), ..., 2\cos(t_s)) \quad \mathbf{t} \in [-\pi,\pi]^s \tag{9}$$

We find that $f^* \in W^{p^*}_{r,s}$. Using induction and the fact that f is identically zero outside $[-3/2,3/2]^s$, using the previous bound (**??**) we find

$$c_1\|f\|_{W^p_{r,s}} \leq \|f^*\|_{W^{p*}_{r,s}} \leq c_2\|f\|_{W^p_{r,s}} \tag{10}$$

We can now check that for any integer m, $v_m(f^*)$ is even and can be written as a sum of just cosines. We write for some set of coefficients $V_{\mathbf{k}}(f)$, the de la Vallée Poussin operator applied to f^* as

$$v_m(f^*, \mathbf{t}) = \sum_{0 \leq \mathbf{k} \leq 2m} V_{\mathbf{k}}(f) \prod_{j=1}^{s} \cos(k_j t_j) \tag{11}$$

For integer $k \geq 0$, we let T_k be the Chebyshev polynomial adapted to the interval $[-2,2]$ as defined by

$$T_k(2\cos t) = \cos(kt) \quad \text{for} \quad t \in [-\pi, \pi] \tag{12}$$

And for a multi-integer $\mathbf{k} \geq 0$, let

$$T_{\mathbf{k}}(\mathbf{x}) = \prod_{j=1}^{s} T_{k_j}(x_j) \quad \text{for} \quad \mathbf{x} \in \mathbb{R}^s \tag{13}$$

The polynomial $P_m(f)$ of co-ordinatewise degree at most $2m$ is then defined as

$$P_m(f, \mathbf{x}) = \sum_{0 \leq \mathbf{k} \leq 2m} V_{\mathbf{k}}(f) T_{\mathbf{k}}(\mathbf{x}) \quad \text{for} \quad \mathbf{x} \in \mathbb{R}^s \tag{14}$$

We relate $P_m(f)$ to $v_m(f^*)$ by the following

$$P_m(f, (2\cos t_1, ..., 2\cos t_s)) = v_m(f^*, \mathbf{t}) \quad \text{for} \quad \mathbf{t} \in [-\pi, \pi]^s \tag{15}$$

Going back to our key result and equation (**??**). We have that

$$\|f - P_m(f)\|_{p,[-\pi,\pi]^s} \leq \frac{c}{m^r} \|f\|_{W^{p*}_{r,s}} \tag{16}$$

and that

$$\sum_{0 \leq \mathbf{k} \leq 2m} |V_{\mathbf{k}}(f)| \leq c m^{\alpha} \|f\|_{W^{p*}_{r,s}} \quad \text{where} \quad \alpha = \frac{s}{\min(p, 2)} \tag{17}$$

To finish our proof we now have to construct an approximation to every polynomial. This is achieved by the following lemma.

Lemma 2.2. *States that given a function ϕ satisfying conditions from Theorem 1, for any integer $m \geq 1$ and any multi-integer $\mathbf{k} \in \mathbb{Z}^s$ with each component $\max_{1 \leq j \leq s} |k_j| \leq m$, there exists a function $G_{\mathbf{k},m,\epsilon} \in \Pi_{\phi;(6m+1)^s,s}$ such that*

$$\|T_{\mathbf{k}} - G_{\mathbf{k},m,\epsilon}\|_\infty \leq \epsilon \tag{18}$$

The weights and thresholds of each $G_{\mathbf{k},m,\epsilon}$ may be chosen from a fixed set with cardinality not exceeding $(6m+1)^s$

Proof. First consider the case when $d = 1$. We take the point $\mathbf{b}$ from (**??**), for $d = 1$ this is a real number, $\mathbf{b} = b$. Let ϕ be infinitely many times continuously differentiable on $[b - \delta, b + \delta]$

For a multi-integer $\mathbf{p} = (p_1, \ldots, p_s)$ and $\mathbf{x} \in \mathbb{R}^s$ we write

$$\mathbf{x}^{\mathbf{p}} := \prod_{j=1}^{s} x_j^{p_j} \quad \text{where we take } 0^0 = 1 \tag{19}$$

We see from

$$\phi_p(\mathbf{w}; \mathbf{x}) := \frac{\partial^{|p|}}{\partial w_{p_1} \ldots \partial w_{p_s}} \phi(\mathbf{w} \cdot \mathbf{x} + b) = x^{\mathbf{p}} \phi^{(|\mathbf{p}|)}(\mathbf{w} \cdot \mathbf{x} + b), \tag{20}$$

we conclude that

$$\mathbf{x}^{\mathbf{p}} = \left(\phi^{(|\mathbf{p}|)} \mathbf{w}(b)\right)^{-1} \phi_{\mathbf{p}}(\mathbf{0}; x) \tag{21}$$

We now apply our appropriate divided difference to replace $\phi_{\mathbf{p}}(\mathbf{0}; x)$. For multi-integers $\mathbf{p}$ and $\mathbf{r}$, we write

$$\binom{p}{r} := \prod_{j=1}^{s} \binom{p_j}{r_j}. \tag{22}$$

For any $h > 0$, the network defined by the formula

$$\Phi_{p,h}(\mathbf{x}) := h^{-|p|} \sum_{0 \le r \le p} (-1)^{|r|} \binom{p}{r} \phi(h(2r - p) \cdot \mathbf{x} + b) \tag{23}$$

is in $\Pi_{\phi;(p_1+1),\ldots,(p_s+1)}$, and represents a divided difference for $\phi_p(0; \mathbf{x})$. Further, we have

$$\|\Phi_{p,h} - \phi_p(0; \cdot)\|_\infty \le M_{\phi,m,s} h^2, \quad \max_{1 \le j \le s} \mid p_j \mid \le m, \mid h \mid \le \delta/(3ms) \tag{24}$$

where $M_{\phi,m,s}$ is a positive constant depending only on the indicated variables.
Now, we write $T_k(\mathbf{x}) := \sum_{0 \le p \le k} \tau_{\mathbf{k},\mathbf{p}} \mathbf{x}^{\mathbf{p}}$, and choose

$$h := h_{\phi,m,s} := \min \left\{ \delta, \min_{0 \le s \le 2m} \left(\frac{\epsilon}{M_{\phi,m,s} \sum_{0 \le p \le k} |\phi(b)|^{-1} |\tau_{k,p}|} \right)^{1/2} \right\}. \tag{25}$$

Then the above equation implies that the network $G_{k,m,\epsilon}$ defined by

$$G_{k,m,\epsilon}(\mathbf{x}) := \sum_{0 \le p \le k} \tau_{k,p} (\phi(b))^{-1} \Phi_{p,h_{\phi,m,s}}(\mathbf{x}), \tag{26}$$

satisfies the bound given in the lemme. For each $\mathbf{k}$, the weights and thresholds in $G_{k,m,\epsilon}$ are chosen from the set

$$\{(h_{\phi,m,s}, b) : b \in \mathbb{Z}^s, |r_j| \le 3m, 1 \le j \le s\}. \tag{27}$$

The cardinality of this set is $(6m + 1)^s$. Therefore, $G_{k,m,\epsilon} \in \Pi_{\phi;(6m+1)^s}$.
Next, if $d > 1$, and $\mathbf{b}$ is as in the original theorem, then we consider the univariate function

$$\sigma(x) := \phi(x, b_2, \ldots, b_d) \tag{28}$$

The function σ satisfies all the hypothesis of Theorem 1, with b_1 in place of $\mathbf{b}$. Taking into account the fact that $\sigma(\mathbf{w} \cdot \mathbf{x} + b_1) = \phi(A_{\mathbf{w}} \mathbf{x} + b)$ with

$$A_{\mathbf{w}} := \begin{pmatrix} \mathbf{w} \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \tag{29}$$

any network in $\Pi_{\sigma;n,s}$ is also a network in $\Pi_{\phi;n,s}$. Therefore, the case $d = 1$ implies the lemma also when $d > 1$. □

Theorem 1 Proof. Without loss of generality, we may assume that $n \ge 13^s$. Let $m \ge 1$ be the largest integer such that $(12m + 1)^s \le n$. We define $P_m(f) = \sum_{0 \le k \le 2m} V_k(f) T_k$ as before. We then define the network

$$N_n(f, x) := \sum_{0 \le k \le 2m} V_k(f) G_{k,2m,m-r-\alpha}(x) \tag{30}$$

is in $\Pi_{\phi;n,s}$ and satisfies

$$\|P_m(f) - N_n(f)\|_\infty \le c m^{-r} \|f\|_{W^p_r,s}. \tag{31}$$

Since $\|g\|_p \le 2^{s/p} \|g\|_\infty$ for all Lebesgue measurable functions g on $[-1, 1]$ we get from the earlier key result

$$\|f - P_m(f)\|_p \le \frac{c}{m^r} \|f\|_{W^p_r,s} \tag{32}$$

that,

$$\|f - N_n(f)\|_p \le cn^{-r/s}\|f\|_{W^p_{r,s}} \tag{33}$$

as required. Further, it is quite clear that the coefficients V_k are continuous linear functionals on L^p. Hence, the continuity assertion follows. $\square$