
A Functional Perspective on Learning Symmetric Functions with Neural Networks

Aaron Zweig¹ Joan Bruna^{1,2}

Abstract

Symmetric functions, which take as input an unordered, fixed-size set, are known to be universally representable by neural networks that enforce permutation invariance. These architectures only give guarantees for fixed input sizes, yet in many practical applications, including point clouds and particle physics, a relevant notion of generalization should include varying the input size. In this work we treat symmetric functions (of any size) as functions over probability measures, and study the learning and representation of neural networks defined on measures. By focusing on shallow architectures, we establish approximation and generalization bounds under different choices of regularization (such as RKHS and variation norms), that capture a hierarchy of functional spaces with increasing degree of non-linear learning. The resulting models can be learned efficiently and enjoy generalization guarantees that extend across input sizes, as we verify empirically.

1. Introduction

Deep learning becomes far more efficient with prior knowledge of function invariants. This knowledge underlies architectural choices that enforce the invariance or equivariance in the network, including Convolutional Neural Networks (LeCun et al., 1998) which encode translation symmetries, and Graph Neural Networks (Scarselli et al., 2008) which encode conjugate permutation symmetries. For functions with invariance to permutation of the input elements, several universal architectures encode this invariance by treating the input as a set (Zaheer et al., 2017; Qi et al., 2017). However, these formulations assume a constant input size, which precludes learning an entire family of symmetric functions.

¹Courant Institute of Mathematical Sciences, New York University, New York ²Center for Data Science, New York University, New York. Correspondence to: Aaron Zweig <az831@nyu.edu>.

Such symmetric functions appear naturally across several domains, including particle physics, computer graphics, population statistics and cosmology. Yet, in most of these applications, the input size corresponds to a sampling parameter that is independent of the underlying symmetric function of interest. As a motivating example, consider the function family induced by the max function, where for varying N , $f_N(\{x_1 \dots x_N\}) = \max_{i \leq N} x_i$. It is natural to ask if a network can simultaneously learn all these functions.

In this work, we interpret input sets as an empirical measure defined over the base space \mathbb{I} , and develop families of neural networks defined over the space of probability measures *probability measures of \mathbb{I}* , as initially suggested in (Pevny & Kovarik, 2019; De Bie et al., 2019). We identify functional spaces characterized by neural architectures and provide generalization bounds that showcase a natural hierarchy among spaces of symmetric functions. In particular, our framework allows us to understand the question of generalizing across input sizes as a corollary. Our constructions rely on the theory of infinitely wide neural networks (Bengio et al., 2006; Rosset et al., 2007; Bach, 2017a), and provide a novel instance of *depth separation* leveraging the symmetric structure of the input.

Summary of Contributions: We consider the infinite-width limit of neural networks taking as domain the space of probability measures in order to formalize learning of symmetric function families. We prove a necessary and sufficient condition for which symmetric functions can be learned. By controlling the amount of non-linear learning, we partition the space of networks on measures into several function classes, proving a separation result among the classes as well as proving a generalization result and empirically studying the performance of these classes to learn symmetric functions on synthetic and real-world data.

Related Work Several works consider representing symmetric functions of fixed input size with invariant neural networks, and in particular there are two main universal architectures, DeepSets (Zaheer et al., 2017) and PointNet (Qi et al., 2017). An alternative generalization of DeepSets is given in (Maron et al., 2019), which proves the universality of tensor networks invariant to any subgroup of the

symmetric group. Regarding variable input size, the work from (Wagstaff et al., 2019) proves lower bounds on representation of the max function in the DeepSets architecture with a dependency on input size.

Separately, there is a wide literature considering neural networks that act on elements on functional data. These results mainly consider universal approximation (Sandberg & Xu, 1996; Stinchcombe, 1999; Rossi & Conan-Guez, 2005). The work (Mhaskar & Hahm, 1997) bears some similarity to the present work, as they prove a quantitative separation between the class of neural networks and the class of functionals with bounded norm, while our main result shows separations among several neural network classes.

The work most similar to ours are (Pevny & Kovarik, 2019; De Bie et al., 2019), which also normalize the DeepSets architecture to define a function on measures. However, they only prove the universality of this model, while we justify the model by classifying symmetric families that are representable and recovering generalization results. We also build on the framework given by (Bach, 2017a), which introduces function classes to characterize neural networks in the wide limit, and proves statistical generalization bounds to demonstrate the advantage of non-linear learning. Although we motivate our work from symmetric functions on finite sets, there are applications in multi-label learning (Frognier et al., 2015) and evolving population dynamics (Hashimoto et al., 2016) that require functions of measures.

Roadmap: We introduce notation and summarize the infinite-dimensional convex network theory theory (Bach, 2017a) in Section 2. In Section 3 we introduce measure networks and characterize their relationship to symmetric functions. Section 4 gives our main result, separating several classes of measure networks according to the degree of non-linear learning, and Section 5 introduces generalization results. Finally, we detail several experiments with the finite instantiation of measure networks in Section 6.

2. Preliminaries

2.1. Problem Setup

Let $\mathbb{I} \subseteq \mathbb{R}^d$ be a convex domain, and $N \in \mathbb{N}$. A *symmetric* function $f : \mathbb{I}^N \rightarrow \mathbb{R}$ is such that $f(x_1, \dots, x_N) = f(x_{\pi(1)}, \dots, x_{\pi(N)})$ for any $x \in \mathbb{I}^N$ and any permutation $\pi \in \mathcal{S}_N$. In this work, we are interested in learning symmetric functions defined independently of N . Let $\bar{\mathbb{I}} = \bigcup_{N=1}^{\infty} \mathbb{I}^N$, then $f : \bar{\mathbb{I}} \rightarrow \mathbb{R}$ is symmetric if f restricted to \mathbb{I}^N is symmetric for each $N \in \mathbb{N}$. Let \mathcal{F}_{sym} denote the space of symmetric functions defined on $\bar{\mathbb{I}}$. This setting is motivated by applications in statistical mechanics and particle physics, where N is a sampling parameter.

We focus on the realizable regression setting, where we

observe a dataset $\{(\mathbf{x}_i, f^*(\mathbf{x}_i)) \in \bar{\mathbb{I}} \times \mathbb{R}\}_{i=1, \dots, n}$ of n samples from an unknown symmetric function f^* , and \mathbf{x}_i are drawn iid from a distribution \mathcal{D} on $\bar{\mathbb{I}}$. The goal is to find a proper estimator $\hat{f} \in \mathcal{F}_{\text{sym}}$ such that the population error $\mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \ell(f^*(\mathbf{x}), \hat{f}(\mathbf{x}))$ is low, where ℓ is a convex loss.

Following a standard Empirical Risk Minimisation setup (Shalev-Shwartz & Ben-David, 2014; Bach, 2017a), we will construct hypothesis classes $\mathcal{F} \subset \mathcal{F}_{\text{sym}}$ endowed with a metric $\|f\|_{\mathcal{F}}$, and consider

$$\hat{f} \in \arg \min_{f \in \mathcal{F}; \|f\|_{\mathcal{F}} \leq \delta} \frac{1}{n} \sum_{i=1}^n \ell(f^*(\mathbf{x}_i), f(\mathbf{x}_i)), \quad (1)$$

where δ is a regularization parameter that is optimised using e.g. cross-validation. We focus on the approximation and statistical aspects of this estimator for different choices of \mathcal{F} ; solving the optimization problem (1) is not the focus of the present work and will be briefly discussed in Section 7.

2.2. Symmetric Polynomials

A simplest way to approximate symmetric functions is with symmetric polynomials. Combining Weierstrass approximation theory with a symmetrization argument, it can be seen that assuming $d = 1$, any symmetric continuous function $f : \mathbb{I}^N \rightarrow \mathbb{R}$ can be uniformly approximated by symmetric polynomials (see (Yarotsky, 2018) for a proof). There are several canonical bases over the ring of symmetric polynomials, but we will consider the one given by the power sum polynomials, given by $p_k(x) = \sum_{i=1}^N x_i^k$, with $x \in \mathbb{I}^N$.

Theorem 2.1 ((2.12) in (Macdonald, 1998)) *For any symmetric polynomial f on N inputs, there exists a polynomial q such that $f(x) = q(p_1(x), \dots, p_N(x))$.*

If q is linear, this theorem suggests a simple predictor for symmetric functions across varying N . If $x \in \mathbb{I}^M$, we can consider $x \mapsto \sum_{i=1}^N c_i \left(\frac{1}{M} p_i(x)\right) = \sum_{i=1}^N c_i \mathbb{E}_{y \sim \mu}(y^i)$ where $\mu = \frac{1}{M} \sum_{j=1}^M \delta_{x_j}$. The truncated moments of the empirical distribution given by x act as linear features, which yield an estimator over any input size M . We will consider a generalization of this decomposition, by moving beyond the polynomial kernel to a general RKHS (see Section 3.1).

2.3. Convex Shallow Neural Networks

By considering the limit of infinitely many neurons (Bengio et al., 2006; Rosset et al., 2007), (Bach, 2017a) introduces two norms on shallow neural representation of functions ϕ defined over \mathbb{R}^d . For a constant $R \in \mathbb{R}$, a fixed probability measure $\kappa \in \mathcal{P}(\mathbb{S}^d)$ with full support, a signed Radon measure $\nu \in \mathcal{M}(\mathbb{S}^d)$, a density $p \in L_2(d\kappa)$, and the notation that $\tilde{x} = [x, R]^T$, define:

$$\gamma_1(\phi) = \inf \left\{ \|\nu\|_{\text{TV}}; \phi(x) = \int_{\mathbb{S}^d} \sigma_\alpha(\langle w, \tilde{x} \rangle) \nu(dw) \right\} \text{ and}$$

$$\gamma_2(\phi) = \inf \left\{ \|p\|_{L_2}; \phi(x) = \int_{\mathbb{S}^d} \sigma_\alpha(\langle w, \tilde{x} \rangle) p(w) \kappa(dw) \right\}$$

where $\|\nu\|_{\text{TV}} := \sup_{|g| \leq 1} \int g d\nu$ is the *Total Variation* of ν and $\sigma_\alpha(t) = \max(0, t)^\alpha$ is the ReLU activation raised to the positive integer power α . These norms measure the minimal representation of ϕ , using either a Radon measure ν over neuron weights, or a density p over the fixed probability measure κ . The norms induce function classes:

$$\mathcal{F}_1 = \{ \phi \in C_0(\mathbb{I}) : \gamma_1(\phi) < \infty \},$$

$$\mathcal{F}_2 = \{ \phi \in C_0(\mathbb{I}) : \gamma_2(\phi) < \infty \}.$$

We also assume that the input domain \mathbb{I} is bounded with $\sup_{x \in \mathbb{I}} \|x\|_2 \leq R$.

These two functional spaces are fundamental for the theoretical study of shallow neural networks and capture two distinct regimes of overparametrisation: whereas the so-called *lazy* or kernel regime corresponds to learning in the space \mathcal{F}_2 (Chizat & Bach, 2018; Jacot et al., 2018), which is in fact an RKHS with kernel given by $k(x, y) = \mathbb{E}_{w \sim \kappa} [\sigma_\alpha(\langle w, \tilde{x} \rangle) \sigma_\alpha(\langle w, \tilde{y} \rangle)]$ (Bach, 2017a)¹, the mean-field regime captures learning in \mathcal{F}_1 , which satisfies $\mathcal{F}_2 \subset \mathcal{F}_1$ from Jensen’s inequality, and can efficiently approximate functions with hidden low-dimensional structure, as opposed to \mathcal{F}_2 (Bach, 2017a).

Finally, one can leverage the fact that the kernel above is an expectation over features to define a finite-dimensional random feature kernel $k_m(x, y) = \frac{1}{m} \sum_{j=1}^m \sigma_\alpha(\langle w_j, \tilde{x} \rangle) \sigma_\alpha(\langle w_j, \tilde{y} \rangle)$ with $w_j \stackrel{i.i.d.}{\sim} \kappa$, which defines a (random) RKHS $\mathcal{F}_{2,m}$ converging to \mathcal{F}_2 as m increases (Bach, 2017b; Rahimi & Recht, 2008). The empirical norm $\gamma_{2,m}$ can be defined similarly to γ_2 , where the density p is replaced by coefficients over the sampled basis functions $\sigma_\alpha(\langle w_j, \cdot \rangle)$.

2.4. Symmetric Neural Networks

A universal approximator for symmetric functions was proposed by (Zaheer et al., 2017), which proved that for any fixed N and $f_N \in \mathcal{F}_{\text{sym}}^N$ there must exist $\Phi : \mathbb{I} \rightarrow \mathbb{R}^L$ and $\rho : \mathbb{R}^L \rightarrow \mathbb{R}$ such that

$$f_N(x) = \rho \left(\frac{1}{N} \sum_{n=1}^N \Phi(x_n) \right). \quad (2)$$

However, universality is only proven for fixed N . Given a symmetric function $f \in \mathcal{F}_{\text{sym}}$ we might hope to learn ρ

¹Or a modified NTK kernel that also includes gradients with respect to first-layer weights (Jacot et al., 2018)

and Φ such that this equation holds for all N . Note that the fraction $\frac{1}{N}$ is not present in their formulation, but is necessary for generalization across N to be feasible (as otherwise the effective domain of ρ could grow arbitrarily large as $N \rightarrow \infty$).

Treating the input to ρ as an average motivates moving from sets to measures as inputs, as proposed in (Pevny & Kovarik, 2019; De Bie et al., 2019). Given $x \in \mathbb{I}^N$, let $\mu^{(N)} = \frac{1}{N} \sum_{i=1}^N \delta_{x_i}$ denote the empirical measure in the space $\mathcal{P}(\mathbb{I})$ of probability measures over \mathbb{I} . Then (2) can be written as $f_N(x) = \rho \left(\int_{\mathbb{I}} \Phi(u) \mu^{(N)}(du) \right)$.

3. From Set to Measure Functions

3.1. Neural Functional Spaces for Learning over Measures

Equipped with the perspective of (2) acting on an empirical measure, we consider shallow neural networks that take probability measures as inputs, with test functions as weights. We discuss in Section 3.2 which functions defined over sets admit an extension to functions over measures.

Let \mathcal{A} be a subset of $C_0(\mathbb{I})$, equipped with its Borel sigma algebra. For $\mu \in \mathcal{P}(\mathbb{I})$, and a signed Radon measure $\chi \in \mathcal{M}(\mathcal{A})$, define $f : \mathcal{P}(\mathbb{I}) \rightarrow \mathbb{R}$ as

$$f(\mu; \chi) = \int_{\mathcal{A}} \tilde{\sigma}(\langle \phi, \mu \rangle) \chi(d\phi). \quad (3)$$

where $\tilde{\sigma}$ is again a scalar activation function, such as the ReLU, and $\langle \phi, \mu \rangle := \int_{\mathbb{I}} \phi(x) \mu(dx)$. Crucially, the space of functions given by $f(\cdot; \chi)$ were proven to be dense in the space of real-valued continuous (in the weak topology) functions on $\mathcal{P}(\mathbb{I})$ in (Pevny & Kovarik, 2019; De Bie et al., 2019), and so this network exhibits universality.

Keeping in mind the functional norms defined on test functions in Section 2.3, we can introduce analogous norms for neural networks on measures. For a fixed probability measure $\tau \in \mathcal{P}(\mathcal{A})$, define

$$\|f\|_{1,\mathcal{A}} = \inf \left\{ \|\chi\|_{\text{TV}}; f(\mu) = \int_{\mathcal{A}} \tilde{\sigma}(\langle \phi, \mu \rangle) \chi(d\phi) \right\}, \quad (4)$$

$$\|f\|_{2,\mathcal{A}} = \inf \left\{ \|q\|_{L_2}; f(\mu) = \int_{\mathcal{A}} \tilde{\sigma}(\langle \phi, \mu \rangle) q(\phi) \tau(d\phi) \right\}, \quad (5)$$

where we take the infima over Radon measures $\chi \in \mathcal{M}(\mathcal{A})$ and densities $q \in L_2(d\tau)$. Analogously these norms also induce the respective function classes $\mathcal{G}_1(\mathcal{A}) = \{f : \|f\|_{1,\mathcal{A}} < \infty\}$, $\mathcal{G}_2(\mathcal{A}) = \{f : \|f\|_{2,\mathcal{A}} < \infty\}$. The argument in Appendix A of (Bach, 2017a) implies $\mathcal{G}_2(\mathcal{A})$ is an RKHS, with associated kernel $k_{\mathcal{G}}(\mu, \mu') = \int_{\mathcal{A}} \tilde{\sigma}(\langle \phi, \mu \rangle) \tilde{\sigma}(\langle \phi, \mu' \rangle) \tau(d\phi)$.

Moving from vector-valued weights to function-valued weights presents an immediate issue. The space $C_0(\mathbb{I})$ is infinite-dimensional, and it is not obvious how to learn a measure χ over this entire space. Moreover, our ultimate goal is to understand finite-width symmetric networks, so we would prefer the function-valued weights be efficiently calculable rather than pathological. To that end, we choose the set of test functions \mathcal{A} to be representable as regular neural networks.

Explicitly, using the function norms of Section 2.3, define $\mathcal{A}_{1,m} = \left\{ \phi; \phi(x) = \sum_{j=1}^m \beta_j \sigma_\alpha(\langle w_j, \tilde{x} \rangle), \|w_j\|_2 \leq 1, \|\beta\|_1 \leq 1 \right\}$ and $\mathcal{A}_{2,m} = \{ \phi \in \mathcal{F}_{2,m} : \gamma_{2,m}(\phi) \leq 1 \}$.

$\mathcal{A}_{1,m}$ thus contains functions in the unit ball of \mathcal{F}_1 that can be expressed with m neurons, and $\mathcal{A}_{2,m}$ contains functions in the (random) RKHS $\mathcal{F}_{2,m}$ obtained by sampling m neurons from κ . By definition $\mathcal{A}_{2,m} \subset \mathcal{A}_{1,m}$ for all m . Representational power grows with m , and observe that the approximation rate in the unit ball of \mathcal{F}_1 or \mathcal{F}_2 is in $m^{-1/2}$, obtained for instance with Monte-Carlo estimators (Bach, 2017a; Ma et al., 2019). Hence we can also consider the setting where $m = \infty$, with the notation $\mathcal{A}_{\{i,\infty\}} = \{ \phi \in \mathcal{F}_i : \gamma_i(\phi) \leq 1 \}$. Note also that there is no loss of generality in choosing the radius to be 1, as by homogeneity of σ any ϕ with $\gamma_i(\phi) < \infty$ can be scaled into its respective norm ball.

We now examine the combinations of \mathcal{G}_i with \mathcal{A}_i :

- $\mathcal{S}_{1,m} := \mathcal{G}_1(\mathcal{A}_{1,m})$; the measure χ is supported on test functions in $\mathcal{A}_{1,m}$.
- $\mathcal{S}_{2,m} := \mathcal{G}_1(\mathcal{A}_{2,m})$; χ is supported on test functions in $\mathcal{A}_{2,m}$.
- $\mathcal{S}_{3,m} := \mathcal{G}_2(\mathcal{A}_{2,m})$; χ has a density with regards to τ , which is supported on $\mathcal{A}_{2,m}$.
- The remaining class $\mathcal{G}_2(\mathcal{A}_{1,m})$ requires defining a probability measure τ over $\mathcal{A}_{1,m}$ that sufficiently spreads mass outside of any RKHS ball. Due to the difficulty in defining this measure in finite setting, we omit this class.

Note that from Jensen’s inequality and the inclusion $\mathcal{A}_{2,m} \subset \mathcal{A}_{1,m}$ for all m , we have the inclusions $\mathcal{S}_{3,m} \subset \mathcal{S}_{2,m} \subset \mathcal{S}_{1,m}$. And $\mathcal{S}_{3,m}$ is clearly an RKHS, since it is a particular instantiation of $\mathcal{G}_2(\mathcal{A})$. In the sequel we will drop the subscript m and simply write \mathcal{A}_i and \mathcal{S}_i .

These functional spaces provide an increasing level of adaptivity: while \mathcal{S}_2 is able to adapt by selecting ‘useful’ test functions ϕ , it is limited to smooth test functions that lie on the RKHS, whereas \mathcal{S}_1 is able to also adapt to more irregular

	First Layer	Second Layer	Third Layer
\mathcal{S}_1	Trained	Trained	Trained
\mathcal{S}_2	Frozen	Trained	Trained
\mathcal{S}_3	Frozen	Frozen	Trained

Table 1: Training for finite function approximation

test functions that themselves depend on low-dimensional structures from the input domain. We let $\|f\|_{\mathcal{S}_i}$ denote the associated norm, i.e. $\|f\|_{\mathcal{S}_1} := \|f\|_{1,\mathcal{A}_1}$.

Finite-Width Implementation: For any m , these classes admit a particularly simple interpretation when implemented in practice. On the one hand, the spaces of test functions are implemented as a single hidden-layer neural network of width m . On the other hand, the integral representations in (4) and (5) are instantiated by a finite-sum using m' neurons, leading to the finite analogues of our function classes given in Table 1. Specifically,

$$f(\mu) = \frac{1}{m'} \sum_{j'=1}^{m'} b_{j'} \tilde{\sigma} \left(\frac{1}{m} \sum_{j=1}^m c_{j',j} \int \sigma_\alpha(\langle w_{j',j}, \tilde{x} \rangle) \mu(dx) \right)$$

One can verify (Neyshabur et al., 2015) that the finite-width proxy for the variation norm is given by

$$\|f\|_1 = \frac{1}{m'} \sum_{j'} |b_{j'}| \|\phi_{j'}\|_1 \leq \frac{1}{mm'} \sum_{j',j} |b_{j'}| c_{j',j} \|w_{j',j}\|,$$

which in our case corresponds to the so-called path norm (Neyshabur et al., 2014). In particular, under the practical assumption that the test functions $\phi_{j'}$ are parameterized by two-layer networks with shared first layer, the weight vectors $w_{j',j}$ only depend on j and this norm may be easily calculated as a matrix product of the network weights. We can control this term by constraining the weights of the first two layers to obey our theoretical assumptions (of bounded weights and test functions in respective RKHS balls), and regularize the final network weights. See Section 6 and the Appendix for practical relaxations of the constraints.

3.2. Continuous Extension

In general, the functions we want to represent don’t take in measures $\mu \in \mathcal{P}(\mathbb{I})$ as inputs. In this section, we want to understand when a function f defined on the power set $f : \mathbb{I} \rightarrow \mathbb{R}$ can be extended to a continuous map $\bar{f} : \mathcal{P}(\mathbb{I}) \rightarrow \mathbb{R}$ in the weak topology, in the sense that for all $N \in \mathbb{N}$ and all $(x_1, \dots, x_N) \in \mathbb{I}^N$, $\bar{f} \left(\frac{1}{N} \sum_{i=1}^N \delta_{x_i} \right) = f(x_1, \dots, x_N)$.

Observe that by construction \bar{f} captures the permutation symmetry of the original f . Define the mapping $D : \mathbb{I} \rightarrow \mathcal{P}(\mathbb{I})$ by $D(x_1, \dots, x_N) = \frac{1}{N} \sum_{i=1}^N \delta_{x_i}$. Let $\hat{\mathcal{P}}_N(\mathbb{I}) := D(\mathbb{I}^N)$ and $\hat{\mathcal{P}}(\mathbb{I}) = \bigcup_{N=1}^{\infty} \hat{\mathcal{P}}_N(\mathbb{I})$, so that $\hat{\mathcal{P}}(\mathbb{I})$ is the set

of all finite discrete measures. For $\mu \in \hat{\mathcal{P}}(\mathbb{I})$, let $N(\mu)$ be the smallest dimension of a point in $D^{-1}(\mu)$, and let x be this point (which is unique up to permutation). Then define $\hat{f} : \hat{\mathcal{P}}(\mathbb{I}) \rightarrow \mathbb{R}$ such that $\hat{f}(\mu) = f_N(x)$.

We also write $W_1(\mu, \mu')$ as the Wasserstein 1-metric under the $\|\cdot\|_2$ norm (Villani, 2008). The following proposition establishes a necessary and sufficient condition for continuous extension of f :

Proposition 3.1 *There exists a continuous extension \bar{f} iff \hat{f} is uniformly continuous with regard to the W_1 metric on its domain.*

This result formalises the intuition that extending a symmetric function from sets to measures requires a minimal amount of regularity across sizes. We next show examples of symmetric families that can be extended to $\mathcal{P}(\mathbb{I})$.

3.3. Examples of Eligible Symmetric Families

Moment-based Functions: Functions based on finite-range interactions across input elements admit continuous extensions. For example, a function of singleton and pairwise interactions $f(x) = \rho\left(\frac{1}{N}\sum_{i=1}^N \phi_1(x_i), \frac{1}{N^2}\sum_{i_1, i_2=1}^N \phi_2(x_{i_1}, x_{i_2})\right)$ is a special case of the continuous measure extension $\bar{f}(\mu) = \rho(\langle \phi_1, \mu \rangle, \langle \phi_2, \mu \otimes \mu \rangle)$ when $\mu = D(x)$.

Ranking: Suppose that $\mathbb{I} \subseteq \mathbb{R}$. The max function $f_N(x) = \max_{i \leq N} x_i$ cannot be lifted to a function on measures due to discontinuity in the weak topology. Specifically, consider $\mu = \delta_0$ and $\nu_N = \frac{N-1}{N}\delta_0 + \frac{1}{N}\delta_1$. Then $\nu_N \rightarrow \mu$, but for \hat{f} as in Proposition 3.1, $\hat{f}(\nu_N) = 1 \neq 0 = \hat{f}(\mu)$.

Nevertheless, we can define an extension on a smooth approximation via the softmax, namely $g_N^\lambda(x) = \frac{1}{\lambda} \log \frac{1}{N} \sum_{i=1}^N \exp(\lambda x_i)$. This formulation, which is the softmax up to an additive term, can clearly be lifted to a function on measures, with the bound $\|g_N^\lambda - f_N\|_\infty \leq \frac{\log N}{\lambda}$. Although we cannot learn the max family across all N , we can approximate arbitrarily well for bounded N .

Counterexamples: Define the map $\Delta_k : \mathbb{R}^N \rightarrow \mathbb{R}^{kN}$ such that $\Delta_k(x)$ is a vector of k copies of x . Then a necessary condition for the function \hat{f} introduced in Proposition 3.1 to be uniformly continuous is that $f_N(x) = f_{kN}(\Delta_k(x))$ for any k . Intuitively, if f_N can distinguish the input set beyond the amount of mass on each point, it cannot be lifted to measures. This fact implies any continuous approximation to the family $f_N(x) = x_{[2]}$, the second largest value of x will incur constant error.

4. Approximation and Function Class Separation

4.1. Approximation of single ‘neurons’

In the same spirit as the ‘separations’ between \mathcal{F}_1 and \mathcal{F}_2 , we characterise prototypical functions that belong to \mathcal{S}_i but have poor approximation rates in \mathcal{S}_{i+1} for $i = \{1, 2\}$ in terms of the relevant parameters of the problem, the input dimensionality d and the bandwidth parameter m . Such functions are given by single neurons in a spherical input regime (details for this setting are given in the Appendix).

For the remainder of this work, we consider $\tilde{\sigma} = \sigma$ as the ReLU activation, and choose $\alpha = 2$ such that $\sigma_2(t) = \sigma(t)^2$ is the squared ReLU.

Theorem 4.1 (informal) *Assume $m = \infty$. For appropriate choices of the kernel base measures κ and τ , there exist f_1 with $\|f_1\|_{\mathcal{S}_1} \leq 1$ and f_2 with $\|f_2\|_{\mathcal{S}_2} \leq 1$ such that:*

$$\inf_{\|f\|_{\mathcal{S}_3} \leq \delta} \|f - f_2\|_\infty \gtrsim d^{-2} \delta^{-5/d},$$

$$\inf_{\|f\|_{\mathcal{S}_2} \leq \delta} \|f - f_1\|_\infty \gtrsim |d^{-11} - d^{-d/3} \delta|.$$

The choice of the squared ReLU activation in the parameterization of the test functions is required in the proof separating \mathcal{S}_1 and \mathcal{S}_2 . This follows from some properties of spherical harmonic parity and the decomposition of signed measures into probability measures.

These separations use the infinity norm rather than an appropriate L_2 norm, and therefore hold in a weaker norm than separation between \mathcal{F}_1 and \mathcal{F}_2 . Nevertheless, these separations confirm that symmetric network expressiveness is *graded* by the degree of non-linear learning.

Both results hold in the domain $m = \infty$, so from the concentration of the empirical kernel $k_m \rightarrow k$, with high probability these approximation lower bounds will still hold for sufficiently large m . In finite-width implementations, however, m may be sufficiently small that the random kernel more explicitly determines the expressiveness of $\mathcal{S}_{i,m}$. We experimentally test the presence of these depth separations with finite m in Section 6.

4.2. Approximation of variational symmetric function via Laplace method

Consider any symmetric family $f_N(x) = \arg \min_{t \in T} \langle \hat{\mu}_x, \phi_t \rangle$ where $\hat{\mu}_x$ is the empirical measure of x , ie, $\hat{\mu}_x = \frac{1}{N} \sum_i \delta_{x_i}$, T is a Euclidean subset, and $t \mapsto \phi_t$ is measurable. For example $T = \mathbb{R}$ and $\phi_t(x) = |t - x|$ yields f_N as the median.

Although this function family isn’t necessarily uniformly continuous in the weak topology, we highlight the option

of a Laplace approximation. Define $E_\mu(t) := \langle \mu, \phi_t \rangle$ and introduce the density $p_\beta(t) = \frac{1}{Z} e^{-\beta E_\mu(t)}$ where $Z = \int_T e^{-\beta E_\mu(t)} dt$ is the partition function. Then consider the Gibbs approximation $g_\beta(\mu) := \mathbb{E}_{p_\beta}[t] = \frac{1}{Z} \int_T t e^{-\beta E_\mu(t)} dt$.

One can verify (e.g. (Raginsky et al., 2017)) that $g_\beta \rightarrow g$ pointwise at a rate $\sim \frac{d \log(\beta+1)}{\beta}$. As g_β is continuous, by universality it can be represented in \mathcal{S}_i for all $i = \{1, 2, 3\}$. An approximation of g_β is given as a ratio of two shallow networks $g_\beta(\mu) = \frac{\int_T t \sigma_1(\langle \mu, \phi_t \rangle) dt}{\int_T \sigma_1(\langle \mu, \phi_t \rangle) dt}$, with $\sigma_1(u) = e^{-\beta u}$. However, the approximation rates blow-up as $\beta \rightarrow \infty$ with an exponential dependency on the dimension of T .

5. Generalization and Concentration

5.1. Generalization Bounds

Despite being a larger function class than \mathcal{F}_2 , the class \mathcal{F}_1 enjoys a nice generalization bound (Bach, 2017a). Crucially, this property is inherited when we lift to functions on measures, controlling the generalization of functions in \mathcal{S}_1 :

Proposition 5.1 *Assume for given δ , for all y the loss function $\ell(y, \cdot)$ is G -Lipschitz on $B_0(2R^2\delta)$, and $\ell(y, 0) \leq RG\delta$. Then with probability at least $1 - t$,*

$$\sup_{\|f\|_{\mathcal{S}_1} \leq \delta} \left| \mathbb{E}_{\mu \sim \mathcal{D}} \ell(f^*(\mu), f(\mu)) - \frac{1}{n} \sum_{i=1}^n \ell(f^*(\mu_i), f(\mu_i)) \right| \leq \frac{2RG\delta + 16R^4G\delta}{\sqrt{n}} + (4R^2G\delta + 2RG\delta) \sqrt{\frac{\log 1/t}{2n}}.$$

This proposition demonstrates that learning in \mathcal{S}_1 is not cursed by the dimension of the underlying input space \mathbb{I} . In other words, the main price for learning in \mathcal{S}_1 is not in generalization, despite the size of this class relative to \mathcal{S}_2 and \mathcal{S}_3 . In the absence of a lower bound on generalization error for the RKHS function classes, our experiments investigate the generalization of these models in practice.

Although d and N do not appear in this bound, these parameters nevertheless impact the generalization of our function classes \mathcal{S}_i . The input dimension controls the separation of the classes according to Theorem 4.1, and therefore larger d weakens the generalization of \mathcal{S}_2 and \mathcal{S}_3 ; compare Figure 1 and Figure 4 (in the Appendix) for how RKHS methods suffer in higher dimensions. Whereas large N and a natural choice of \mathcal{D} make generalization for \mathcal{S}_1 , and hence all three classes, nearly trivial, as discussed in section 5.2.

5.2. Concentration across Input Size

Consider the data distribution from which we sample, namely a measure from $\mathcal{P}(\mathbb{I})$ to sample finite sets. A natural way to draw data is to consider the following sampling

procedure: given $\xi \in \mathcal{P}(\mathcal{P}(\mathbb{I}))$ and $\Omega \in \mathcal{P}(\mathbb{N})$, draw $\mu \sim \xi$ and $N \sim \Omega$, sample N independent points $x_i \sim \mu$, and return $\{x_1, \dots, x_N\}$. If ξ is too peaked, this sampling process will concentrate very rapidly:

Proposition 5.2 *For $\xi = \delta_{\mu^*}$, the Rademacher complexity $\mathbb{E} \sup_{\|f\|_{\mathcal{S}_1} \leq \delta} \left| \frac{1}{n} \sum_{i=1}^n \epsilon_i f(\mu_i) \right| \lesssim \delta R(n^{-1/2} + R \mathbb{E}_{N \sim \Omega} [N^{-1/d}])$.*

Hence, the question of generalization across differently sized sets becomes trivial if N is large and d is small. In our experiments, $N \approx d$, so we will nevertheless choose $\xi = \delta_\mu$ for some $\mu \in \mathcal{P}(\mathbb{I})$. We consider more exotic data distributions over measures in the experiments on robust mean estimation in Section 6.2.

6. Experiments

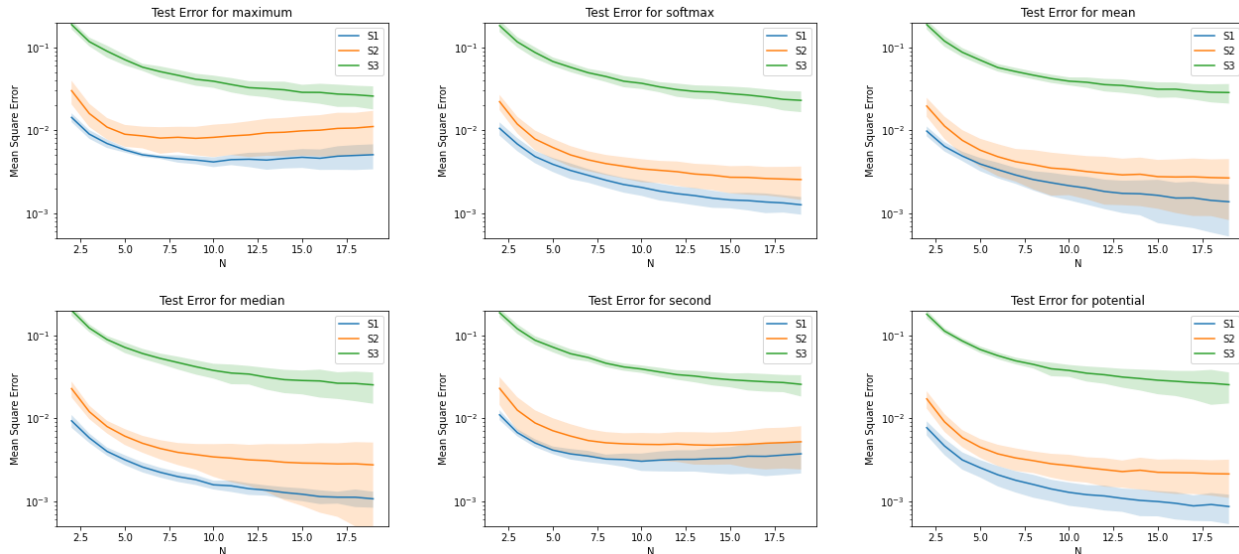
6.1. Symmetric Function Approximation

We consider the task of learning several common symmetric functions (see Figure 1). Our aim is to practically understand the approximation bounds of Theorem 4.1, as well as the generalization result of Proposition 5.1. Furthermore, by training and testing on sets of different sizes, we may consider how the models perform on out-of-distribution generalization across input size.

Experimental Setup: We instantiate our three function classes in the finite network setting, as outlined in Table 1. We use input dimension $d = 10$. For the finite realization of \mathcal{S}_1 , we use first hidden layer size $m = 100$ and second hidden layer size $h = 100$. Crucially, after fixing the finite architecture representing \mathcal{S}_1 , we scale up the width by 10 for the models with frozen weights. That is, the first hidden layer in \mathcal{S}_2 , and both hidden layers in \mathcal{S}_3 , have width equal to 1000. Increasing the width makes the \mathcal{S}_2 and \mathcal{S}_3 models strictly more powerful, and this setup allows us to inspect whether a larger number of random kernel features can compensate for a smaller, trained weight in approximation. For each model, we use its associated functional norm for regularization.

Each network is trained on a batch of 100 input sets. For our data distribution we consider the base domain $\mathbb{I} = [-3, 3]^d$, and the distribution over input measures ξ places all its mass on the uniform measure $U([-3, 3]^d)$. We choose to train with $N = 4$, i.e. all networks train on input sets of size 4, and test on sets of varying size. From the results we can measure out-of-distribution generalization of finite sets.

The one-dimensional symmetric functions are defined on sets of vectors by first applying inverse norms, i.e. $f_N(x) = \max_{1 \leq i \leq N} \|x_i\|_2^{-1}$. The potential function calculates the normalized gravitational potential among equal masses, i.e. $f_N(x) = \frac{2}{N(N-1)} \sum_{i < j} \frac{1}{\|x_i - x_j\|_2}$. The planted neuron and


 Figure 1: Test Error for $d = 10$ on the neural architectures of Section 3.1

smooth neuron are given as single-neuron networks, where following from the proof of Theorem 4.1, the planted neuron weight initialization is distinct from the model weight initialization. Further details are given in the Appendix.

We additionally consider an applied experiment on a variant of MNIST to observe how the finite-width implementations perform on real-world data, by first mapping images to point clouds. Due to space limitations, details and results are given in the Appendix.

Discussion: We observe in Figure 1 that \mathcal{S}_3 performs substantially worse in several cases, consistent with this function class being the smallest of those considered. The classes \mathcal{S}_2 and \mathcal{S}_1 are competitive for some functions, although we observe a trend where \mathcal{S}_1 still has better generalization performance. Therefore, the larger number of random kernel features doesn't compensate for training a smaller weight matrix in \mathcal{S}_1 , empirically confirming Theorem 4.1.

The test error on sets of larger size than the training data corroborates the conclusion of Proposition 3.1. The second-largest-element function generalizes extremely poorly, consistent with the observation in Section 3.3 that this function family cannot be approximated without constant error. In particular, all function classes more effectively generalize across different N on the softmax than the max, seeing as the latter lacks uniform continuity in measure space.

The other essential takeaway is the performance of the three models on the planted neurons in Figure 2. By using a distinct weight initialization for the neuron, its first layer will have very little mass under κ , and its first two layers will have little mass under τ , and therefore random features will not suffice to approximate this neuron. This is true

even with the scaling of \mathcal{S}_2 and \mathcal{S}_3 to enable more random kernel features, reiterating that these single neuron functions realize a meaningful separation between the classes. We observe a more similar performance of \mathcal{S}_1 and \mathcal{S}_2 on the smooth_neuron, as this function is chosen to be exactly representable with the random kernel features sampled by \mathcal{S}_2 . According to the function class inclusion it is still representable by \mathcal{S}_1 , but from Theorem 4.1 not efficiently representable by \mathcal{S}_3 , which is consistent with the results.

On increasing m , the standard deviations of \mathcal{S}_2 and \mathcal{S}_3 shrink with more random kernel features, but \mathcal{S}_1 still achieves the best approximation on the neuron. For the smooth neuron, \mathcal{S}_1 and \mathcal{S}_2 perform comparably, but \mathcal{S}_3 performs worse even for larger m . In Figure 3 we confirm the need for taking averages rather than sums in the DeepSets architecture, as the unnormalized model cannot generalize outside of the value of $N = 4$ where it was trained.

6.2. Robust Mean Estimation

Symmetric functions naturally arise in the context of empirical estimators. We consider specifically the task of robust mean estimation (Diakonikolas et al., 2017), where one seeks to estimate $\mathbb{E}_{X \sim P}[X]$ given samples drawn from the mixture distribution $(1 - \epsilon)P + \epsilon Q$. For simplicity, we consider an oblivious contamination model where the true distribution P and the noise distribution Q have similar mean vectors. Explicitly, each input set is derived as follows: we sample $m \sim \mathcal{N}(0, \sigma_m^2 I)$, $m' \sim \mathcal{N}(m, \sigma_m^2 I)$, and define $P = \mathcal{N}(m, \sigma_P^2 I)$ and $Q = \mathcal{N}(m', \sigma_Q^2 I)$. Then each input sets consist of N samples (X_1, \dots, X_N) where $X_i \stackrel{iid}{\sim} (1 - \epsilon)P + \epsilon Q$. Note that each input set is a corrupted sample with a different true mean vector m .

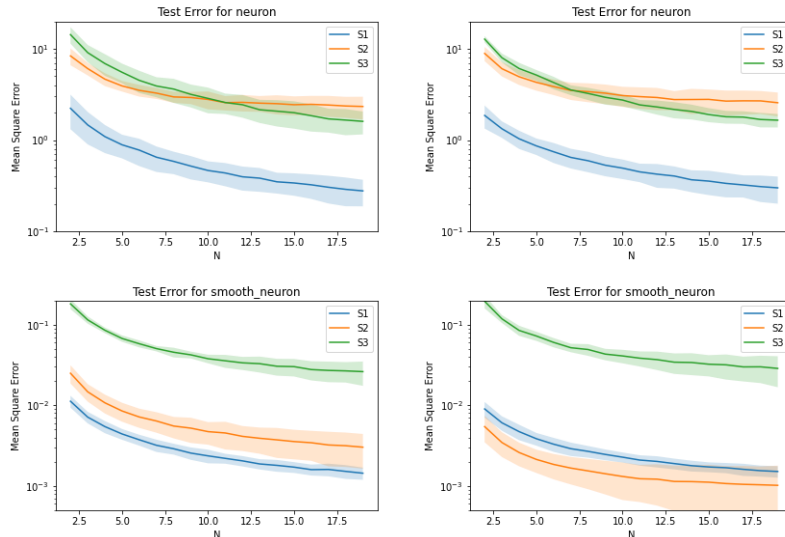


Figure 2: Planted neurons for $m = 100$ (left two) and $m = 200$ (right two). The smooth neuron has weights sampled consistently with \mathcal{F}_2 while the regular neuron has weights sampled distinctly from the network initialization.

	$N = 10$	$N = 20$	$N = 30$	$N = 40$
\mathcal{S}_1	0.335 ± 0.153	0.131 ± 0.018	0.091 ± 0.011	0.076 ± 0.011
\mathcal{S}_2	0.342 ± 0.153	0.137 ± 0.019	0.098 ± 0.012	0.082 ± 0.011
\mathcal{S}_3	0.361 ± 0.162	0.144 ± 0.020	0.103 ± 0.013	0.087 ± 0.013
Sample Mean	0.385 ± 0.172	0.153 ± 0.068	0.093 ± 0.042	0.096 ± 0.043
Geometric Median	0.321 ± 0.144	0.138 ± 0.062	0.087 ± 0.039	0.077 ± 0.034
Adversarial Estimator	0.612 ± 0.495	0.469 ± 0.550	0.417 ± 0.549	0.420 ± 0.564

Table 2: Mean squared test error for robust mean estimation among the finite model instantiations and baselines.

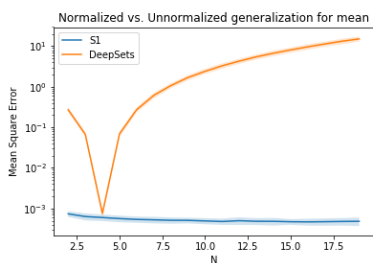


Figure 3: Test error for \mathcal{S}_1 versus unnormalized DeepSets architecture.

Experimental Setup: The network architecture is the same as above, with $d = 10$. We use $\sigma_m = 1$, $\sigma_{m'} = 2$, $\sigma_P = \sigma_Q = 1.5$, and $\epsilon = 0.2$. All networks train on sets of size $N = 20$, and test on sets of varying size, with mean squared error as the objective. As baselines we consider the naive sample mean, the geometric median, and the adversarially robust mean estimator proposed in (Diakonikolas et al., 2017). The results are given in Table 2.

Discussion: Although the variance is quite high due to the sampling procedure, performance in this setting confirms

that robust mean estimation also realizes the class separation, and that for this simple corruption model learning is competitive and in some cases superior to fixed estimators. In particular, the advantage of \mathcal{S}_1 over the baselines is most clear for $N = 20$, the setting where it was trained. Although the dependence of the fixed estimators on σ_P and σ_Q vanishes as $N \rightarrow \infty$, the dependence on these parameters is non-negligible in the regime where N is small, and therefore the robust mean may not generalize in the sense of Proposition 3.1. We explore training on different N sizes further in the Appendix. The poor performance of the adversarial estimator can mainly be attributed to the fact that the number of samples is considerably smaller than the setting studied in (Diakonikolas et al., 2017), weakening the concentration of the empirical covariance matrix on which this estimator relies.

7. Conclusion

In this work, we have analyzed learning and generalization of symmetric functions through the lens of neural networks defined over probability measures, which formalizes the

learning of symmetric function families across varying input size. Our experimental data confirms the theoretical insights distinguishing tiers of non-linear learning, and suggests that symmetries in the input might be a natural device to study the functional spaces defined by deeper neural networks. Specifically, and by focusing on shallow architectures, our analysis extends the fundamental separation between adaptive and non-adaptive neural networks from (Bach, 2017a) to symmetric functions, leading to a hierarchy of functional spaces $\mathcal{S}_3 \subset \mathcal{S}_2 \subset \mathcal{S}_1$, in which nonlinear learning is added into the parametrization of the network weights (\mathcal{S}_2), and into the parametrization of test functions (\mathcal{S}_1) respectively.

A crucial aspect we have not addressed, though, is the computational cost of learning in \mathcal{S}_1 through gradient-based algorithms. An important direction of future work is to build on recent advances in mean-field theory for learning shallow neural networks (Chizat & Bach, 2020; Ma et al., 2019; 2020; de Dios & Bruna, 2020).

Acknowledgements: We thank Raghav Singhal for helpful discussions regarding the proof of Theorem 4.1. This work has been partially supported by the Alfred P. Sloan Foundation, NSF RI-1816753, NSF CAREER CIF-1845360, and NSF CCF-1814524.

References

- Bach, F. Breaking the curse of dimensionality with convex neural networks. *The Journal of Machine Learning Research*, 18(1):629–681, 2017a.
- Bach, F. On the equivalence between kernel quadrature rules and random feature expansions. *The Journal of Machine Learning Research*, 18(1):714–751, 2017b.
- Bengio, Y., Roux, N. L., Vincent, P., Delalleau, O., and Marcotte, P. Convex neural networks. In *Advances in neural information processing systems*, pp. 123–130, 2006.
- Chizat, L. and Bach, F. A note on lazy training in supervised differentiable programming. *arXiv preprint arXiv:1812.07956*, 2018.
- Chizat, L. and Bach, F. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. *arXiv preprint arXiv:2002.04486*, 2020.
- De Bie, G., Peyré, G., and Cuturi, M. Stochastic deep networks. In *International Conference on Machine Learning*, pp. 1556–1565, 2019.
- de Dios, J. and Bruna, J. On sparsity in overparametrised shallow relu networks. *arXiv preprint arXiv:2006.10225*, 2020.
- Diakonikolas, I., Kamath, G., Kane, D. M., Li, J., Moitra, A., and Stewart, A. Being robust (in high dimensions) can be practical. In *International Conference on Machine Learning*, pp. 999–1008. PMLR, 2017.
- Efthimiou, C. and Frye, C. *Spherical harmonics in p dimensions*. World Scientific, 2014.
- Fournier, N. and Guillin, A. On the rate of convergence in wasserstein distance of the empirical measure. *Probability Theory and Related Fields*, 162(3-4):707–738, 2015.
- Frogner, C., Zhang, C., Mobahi, H., Araya, M., and Poggio, T. A. Learning with a wasserstein loss. In *Advances in neural information processing systems*, pp. 2053–2061, 2015.
- Hashimoto, T., Gifford, D., and Jaakkola, T. Learning population-level diffusions with generative rnns. In *International Conference on Machine Learning*, pp. 2417–2426, 2016.
- He, K., Zhang, X., Ren, S., and Sun, J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.
- Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pp. 8571–8580, 2018.
- Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- Ma, C., Wu, L., and E, W. Barron spaces and the compositional function spaces for neural network models. *arXiv preprint arXiv:1906.08039*, 2019.
- Ma, C., Wu, L., et al. The quenching-activation behavior of the gradient descent dynamics for two-layer neural network models. *arXiv preprint arXiv:2006.14450*, 2020.
- Macdonald, I. G. *Symmetric functions and Hall polynomials*. Oxford university press, 1998.
- Maron, H., Fetaya, E., Segol, N., and Lipman, Y. On the universality of invariant networks. In *International Conference on Machine Learning*, pp. 4363–4371, 2019.
- Mhaskar, H. N. and Hahm, N. Neural networks for functional approximation and system identification. *Neural Computation*, 9(1):143–159, 1997.

- Neyshabur, B., Tomioka, R., and Srebro, N. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014.
- Neyshabur, B., Tomioka, R., and Srebro, N. Norm-Based Capacity Control in Neural Networks. *arXiv:1503.00036 [cs, stat]*, April 2015. URL <http://arxiv.org/abs/1503.00036>. arXiv: 1503.00036.
- Pevny, T. and Kovarik, V. Approximation capability of neural networks on spaces of probability measures and tree-structured domains. *arXiv preprint arXiv:1906.00764*, 2019.
- Qi, C. R., Su, H., Mo, K., and Guibas, L. J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660, 2017.
- Raginsky, M., Rakhlin, A., and Telgarsky, M. Non-convex learning via stochastic gradient langevin dynamics: a nonasymptotic analysis. *arXiv preprint arXiv:1702.03849*, 2017.
- Rahimi, A. and Recht, B. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pp. 1177–1184, 2008.
- Rosset, S., Swirszcz, G., Srebro, N., and Zhu, J. ℓ_1 regularization in infinite dimensional feature spaces. In *International Conference on Computational Learning Theory*, pp. 544–558. Springer, 2007.
- Rossi, F. and Conan-Guez, B. Functional multi-layer perceptron: a non-linear tool for functional data analysis. *Neural networks*, 18(1):45–60, 2005.
- Sandberg, I. W. and Xu, L. Network approximation of input-output maps and functionals. *Circuits, Systems and Signal Processing*, 15(6):711–725, 1996.
- Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008.
- Shalev-Shwartz, S. and Ben-David, S. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.
- Stinchcombe, M. B. Neural network approximation of continuous functionals and continuous functions on compactifications. *Neural Networks*, 12(3):467–477, 1999.
- Villani, C. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.
- Wagstaff, E., Fuchs, F., Engelcke, M., Posner, I., and Osborne, M. A. On the limitations of representing functions on sets. In *International Conference on Machine Learning*, pp. 6487–6494, 2019.
- Yarotsky, D. Universal approximations of invariant maps by neural networks. *arXiv preprint arXiv:1804.10306*, 2018.
- Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R. R., and Smola, A. J. Deep sets. In *Advances in neural information processing systems*, pp. 3391–3401, 2017.