Learning Neural Networks with Sparse Activations

Pranjal Awasthi PRANJALAWASTHI@GOOGLE.COM
Google Research

Nishanth Dikkala NISHANTHD@GOOGLE.COM
Google Research

Pritish Kamath PRITISH@ALUM.MIT.EDU
Google Research

Raghu Meka RAGHUM@CS.UCLA.EDU
University of California, Los Angeles

Abstract

A core component present in many successful neural network architectures, is an MLP block of two fully connected layers with a non-linear activation in between. An intriguing phenomenon observed empirically, including in transformer architectures, is that, after training, the activations in the hidden layer of this MLP block tend to be extremely sparse on any given input. Unlike traditional forms of sparsity, where there are neurons/weights which can be deleted from the network, this form of *dynamic* activation sparsity appears to be harder to exploit to get more efficient networks.

Motivated by this we initiate a formal study of PAC learnability of MLP layers that exhibit activation sparsity. We present a variety of results showing that such classes of functions do lead to provable computational and statistical advantages over their non-sparse counterparts. Our hope is that a better theoretical understanding of *sparsely activated* networks would lead to methods that can exploit activation sparsity in practice.

Keywords: Multilayer Perceptrons, PAC Learning, Activation Sparsity, Rademacher Complexity

1. Introduction

In recent years, transformer based deep neural networks (Vaswani et al., 2017) and the subsequent development of large language models have marked a paradigm shift in the fields of natural language processing and computer vision (Brown et al., 2020; Chowdhery et al., 2022; Chen et al., 2022b; Dosovitskiy et al., 2020). These models have significantly improved performance across various tasks, setting new benchmarks and enabling previously unattainable breakthroughs. However, the computational cost of training and deploying these models, especially the largest variants, presents a significant challenge. A notable portion of these models' computational and parameter overhead is attributed to the Multi-Layer Perceptron (MLP) layers. These layers are integral to the transformer architecture, playing a crucial role in its ability to solve many different tasks.

Despite their efficacy, the resource-intensive nature of these models has spurred a wave of research focused on enhancing their efficiency (Banner et al., 2019; Frankle and Carbin, 2018; Gholami et al., 2022; Hinton et al., 2015; Anil et al., 2018; Harutyunyan et al., 2023). Among the various strategies explored for improving the inference efficiency of large transformers, attempting to sparsify the transformer is a promising approach.

A motivation for exploiting sparsity is rooted in an intriguing empirical observation made in recent works (Li et al., 2023) regarding the behavior of MLP layers within large transformer models. Post-training, these layers tend to exhibit a high degree of sparsity in their activations; often each input activates as low as 3% of the neurons in the MLP layers, suggesting a natural emergence of sparsity in activations. This leads to these MLP layers behaving like key-value lookups (Geva et al., 2020). The extremely low sparsity (3%) suggests that there might be significant room to sparsify the MLP layers leading to both training and inference efficiency. In addition, such sparsity also helps with interpretability of transformers by disentangling neurons corresponding to distinct concepts (Elhage et al., 2022). Moreover, through extensive ablation studies Li et al. (2023) observe that this phenomenon is highly prevalent. It occurs in convolutional networks (CNNs), as well as in vanilla fully connected feedforward networks.

Despite the potential benefits, effectively harnessing dynamic sparsity has proven challenging. Although, there have been many recent efforts (Li et al., 2023; Grimaldi et al., 2023; Liu et al., 2023; Dong et al., 2023; Csordás et al., 2023; Mirzadeh et al., 2023), they have led to limited success. None of the approaches achieve speedups (either in training or in inference) anywhere close to the the potential factor of 33x that is suggested by 3% sparsity. Moreover, by explicitly enforcing sparsity via methods such as choosing only the top-k activations, the quality of the model degrades in some cases.

A key reason for the hardness in exploiting activation sparsity is that this form of sparsity is *dynamic* in nature and is input-dependent (i.e., not a fixed pattern). While each input example activates a small number of neurons, the overall sparsity pattern cannot be localized to a small subset of the model weights. For instance, the dynamic nature precludes the use of typical weight quantization or pruning based methods to exploit sparsity empirically. On the other hand, having a non-localized sparsity pattern is crucial in ensuring the model has rich expressiveness.

The above observations suggest that post-training, large transformer networks belong to an intriguing function class that is highly expressive yet exhibits high sparsity. Given the challenges in exploiting this behavior in practical settings, in this work, we initiate a theoretical study of the statistical and computational properties of such functions in the probably approximately correct (PAC) learning framework (Valiant, 1984).

We introduce the class of *sparsely activated* MLPs. We focus on the case of depth-1 MLPs with n input units and s hidden units with the standard ReLU activations. We define the class $\mathcal{H}_{n,s,k}$ as the class of depth-1 ReLU networks in n-dimensions with the promise that on each input in the support of the data distribution, at most k of the s hidden units are active:

Definition 1 (Sparsely Activated Networks) *Let $\sigma(\cdot)$ denote the ReLU activation, namely $\sigma(z) := \max\{z, 0\}$. The class $\mathcal{H}_{n,s,k}$ consists of hypotheses of the form $h(x) = \sum_{j=1}^{s} u_j \sigma(\langle w_j, x\rangle - b_j)$ with the property that for all x in the support of the distribution, it holds that $|\{j : \langle w_j, x\rangle - b_j > 0\}| \leq k$.*

Note that this sparsity differs from *dead sparsity*, where some neurons are never active on any of the inputs, and consequently, can be deleted from the network without impacting its functionality. The form of dynamic sparsity we study can be crucial for the networks to be more expressive. We provide a couple of examples of useful functions represented using sparsely activated networks here:

- **Junta functions:** The class of functions on n variables which depend on only a p-sized subset ($p < n$) of the variables is known as p-junta functions. Sparse parities are a canonical example of junta functions. We show in Theorem 13 that we can represent $\log(s)$-juntas using $\mathcal{H}_{n,s,1}$.

- **Indexing function:** Consider the function $\mathsf{Index}_b : \{-1, 1\}^{b+2^b} \to \{0, 1\}$, where $\mathsf{Index}_b(z)$ is the x-th bit of y (-1 mapped to 0), where x is the integer represented by the first b bits of z in binary representation, and y is the remaining 2^b bits vector. This can be represented as a 1-sparse activation network of size 2^b (i.e., in $\mathcal{H}_{b+2^b,2^b,1}$): $\mathsf{Index}_b((x,y)) = \sum_{\alpha\in\{-1,1\}^b} \sigma(\langle w_\alpha, z\rangle - b + \frac{1}{2})$ where the first b coordinates of w_α are α and the α-th coordinate among the last 2^b coordinates is $\frac{1}{2}$. On input $z = (x,y)$, only the neuron corresponding to $\alpha = x$ is activated, and the output is precisely $\frac{1}{2}y_x + \frac{1}{2}$.

In both the examples presented above, removing any of the s neurons will change the functionality of the network. However, each weight vector w_i is quite sparse. In Appendix A, we present an example of a sparsely activated network where even the weight vectors w_i are not sparse. Hence, in general, it is not clear if sparsely activated networks can be represented with fewer neurons or sparse weight vectors.

In order to provide learning guarantees, we have to assume an upper bound on the *scale* of u, w_j's and b_j's. We will use the following natural scaling for the paper:

Definition 2 *Let $\mathcal{H}^{W,B}_{n,s,k} \subseteq \mathcal{H}_{n,s,k}$ consisting of h given as $h(x) = \sum_{j=1}^{s} u_j\sigma(\langle w_j, x\rangle - b_j)$, satisfying $\|u\|_\infty \cdot \max_{j\in[s]} \|w_j\|_2 \le W$ and $\|u\|_\infty \cdot \max_{j\in[s]} |b_j| \le B$.*

We then consider the problem of learning sparsely activated networks efficiently. We consider the domain to be the Boolean hypercube $\mathcal{X} = \{1,-1\}^n$ as a natural first-step and as a domain where sparsely activated networks can compute non-trivial functions. The Boolean hypercube provides a setting where the function can be sparse everywhere in the domain while maintaining expressiveness; this appears harder in the continuous setting. For instance, if the inputs are Gaussian over $\mathbb{R}^n$, one likely needs the biases in the ReLU units to be very large to enforce 1-sparsity. This suggests that, in the continuous domain, more non-standard distributions are likely necessary to obtain a rich class of functions which are sparse everywhere in the domain. Hence for theoretical simplicity we focus on functions on the Boolean hypercube.

Even with the sparsity assumption, the class $\mathcal{H}_{n,s,1}$ is likely hard to learn in polynomial time (or even quasi-polynomial time) under an arbitrary distribution on the hypercube. In particular, we show that *parities* on the hypercube on k variables can be computed by $\mathcal{H}_{k^2,2k,1}$, with coefficient vectors of norm at most $O(k)$. Thus, $\mathcal{H}_{n,O(\sqrt{n}),1}$ need $2^{\Omega(\sqrt{n})}$ queries in the powerful Statistical Queries (SQ) model (see Section 4 for details). We also show cryptographic hardness results for learning $\mathcal{H}_{n,s,1}$ under generic distributions on the hypercube.

Theorem 3 (Informal; see Section 4) *Any SQ algorithm for learning $\mathcal{H}^{O(n^{0.75}),O(n)}_{n,O(\sqrt{n}),1}$ under arbitrary distributions over the hypercube either requires $2^{-\Omega(\sqrt{n})}$ tolerance or $2^{\Omega(\sqrt{n})}$ queries.*

Assuming the hardness of learning with rounding *problem with polynomial modulus, there is no* $\mathsf{poly}(n,s,W,B,1/\varepsilon)$ *run-time algorithm to (ε,δ)-PAC learn $\mathcal{H}^{W,B}_{n,s,1}$.*

Learning under uniform distribution. Given the above hardness results, it is natural to consider distributional assumptions as is often done for related classes in learning theory (e.g., Klivans et al. (2004); Kane (2014) etc.). Our main result is that when the input distribution is uniform over the n-dimensional hypercube, $\{1,-1\}^n$, the class $\mathcal{H}^{W,B}_{n,s,k}$ can be learned in time $n^{\mathsf{poly}(k\log(ns))}$:

Theorem 4 (Informal; see Theorem 8) *There exists an (ε,δ)-PAC learning algorithm for $\mathcal{H}^{W,B}_{n,s,k}$ with respect to the uniform distribution over $\{1,-1\}^n$ that has sample complexity and run-time $n^{\mathsf{poly}(k\log(ns))/\varepsilon^2}\log(1/\delta)/\varepsilon$ (suppressing dependence on W, B).*

As our learning algorithm works by performing linear regression over low-degree monomial basis (a.k.a. the *low-degree algorithm*), the guarantees work even in the *agnostic* or *non-realizable* setting by standard arguments (e.g., Klivans et al. (2004)). For simplicity, we focus on the realizable setting as the algorithm and analysis do not change for the agnostic case.

For sparsity $k = 1$, the above run-time is $n^{O(\mathsf{poly}(\log(ns))/\varepsilon^2)}$. As we showed above, $\mathcal{H}_{n,s,1}$ can simulate juntas of size $\log_2 s$ over n variables. Thus, a quasi-polynomial run-time is the best we can do under a widely believed conjecture on the hardness of learning juntas.

The guarantee above is in stark contrast to what is achievable for general one-layer size s ReLU networks under the uniform distribution over the hypercube. One-layer size-s networks can simulate parities on $\min(n, s)$ variables. They thus cannot be learned even under the uniform distribution on the hypercube by SQ algorithms with less than $2^{\Omega(\min(n,s))}$ queries. Further, even for non-SQ algorithms, as shown in (Chen et al., 2022a), quasi-polynomial run-time with respect to the uniform distribution on the hypercube is impossible under widely studied cryptographic assumptions.

The proof of Theorem 4 is via Fourier analysis and the *low-degree algorithm*. The main ingredient is to show that the *average-sensitivity* of functions in $\mathcal{H}_{n,s,k}$ is at most $O(k^4(\sqrt{n}\log(ns)))$. We then use this bound the *noise-sensitivity* of functions in $\mathcal{H}_{n,s,k}$. The latter implies the existence of a low-degree approximation by exploiting Klivans et al. (2004) which is enough to obtain the theorem. See Section 3 for details.

Learning under general distributions. We also show that $\mathcal{H}^{W,B}_{n,s,k}$ can be learnt under general distributions with smaller sample complexity than would be required without the sparsity condition, in the case when $s \gg kn$. In particular, we show the following.

Theorem 5 (Informal; see Theorem 17) *There exists an (ε, δ)-PAC learning algorithm for $\mathcal{H}^{W,B}_{n,s,k}$ over $\{1, -1\}^n$ that has sample complexity $\widetilde{O}\left(ksn/\varepsilon^2\right)$ (suppressing dependence on W, B, δ).*

By contrast, the class $\mathcal{H}^{W,B}_{n,s,s}$ (that is, size-s networks without activation sparsity) requires a sample complexity of $\Omega(s^2/\varepsilon^2)$. To prove the above, we provide a bound on the Rademacher complexity of the class $\mathcal{H}^{W,B}_{n,s,k}$ that has an improved dependence on s.

Taken together, our results demonstrate that leveraging dynamic activation sparsity is theoretically possible for both computational and statistical benefits. We hope that further theoretical study of the class of sparsely activated networks could pave the way for more efficient training and inference methods for deep architectures, including transformer-based models where these sparsely activated networks have been observed to arise in practice.

1.1. Related Work

Our work is motivated by recent empirical observations on the extreme sparsity observed in the MLP layers of trained transformer models (Li et al., 2023; Shen et al., 2023). The works of Li et al. (2023); Peng et al. (2023) propose theoretical explanations of why this phenomenon occurs. However, ours is the first work to formally study sparsely activated networks in the PAC learning setup and quantify their computational and statistical advantages. Motivated by the observation on sparsity, recent work has also studied the connections between the MLP layers and key-value memory lookups (Sukhbaatar et al., 2019; Lample et al., 2019; Geva et al., 2020).

There have also been recent works on designing networks with explicitly enforced sparsity structure. One such line of work concerns mixture of experts models (Shazeer et al., 2017; Fedus et al.,

2022) where each input is independently routed to one or two MLP blocks among a set of experts. An alternate way to enforce sparsity is to introduce a top-k operation after each MLP layer that zeros out most of the activations (Csordás et al., 2023; Li et al., 2023). In particular, Li et al. (2023) propose a top-k transformer along these lines. However, due to the top-k operation being relatively slow on accelerator hardware, this technique does not yield wall-clock speedup for either training or inference.

In another recent work Liu et al. (2023) propose to train a small predictor network to predict the activated indices at each MLP layer. There has also been work to explore enforcing block sparsity constraints and weight tying in the model weights themselves (Dong et al., 2023), as well as efforts to enforce static sparsity that is not input dependent (Frantar and Alistarh, 2023). However such methods haven't been effective for language modeling via transformer models and have been much more successful in classification domains that have a small number of output labels.

Significantly more attention has been given to sparsifying attention layer computation (Zaheer et al., 2020; Choromanski et al., 2020; Wang et al., 2020; Gu and Dao, 2023). Instead, our focus in this work here is understanding the sparsity behavior of the MLP layer.

2. Preliminaries

We consider the problem of learning real-valued functions over the input space $\mathcal{X} = \{-1, 1\}^n$, to small expected ℓ_2-squared error, namely for the underlying distribution $\mathcal{D}$ over $(x, y) \in \mathcal{X} \times \mathbb{R}$, our goal is the minimize the population loss of a predictor $f : \mathcal{X} \to \mathbb{R}$ given as $\mathcal{L}_{\mathcal{D}}(f) := \mathbb{E}_{(x,y)\sim\mathcal{D}}\, \ell(f(x), y)$ where $\ell(\hat{y}, y) := \frac{1}{2}(\hat{y} - y)^2$. For any dataset $S \in (\mathcal{X} \times \mathbb{R})^*$, we denote the empirical loss as $\mathcal{L}_S(f) := \frac{1}{|S|}\sum_{(x,y)\in S} \ell(f(x), y)$.

For any hypothesis class $\mathcal{H} \subseteq \mathbb{R}^{\mathcal{X}}$, we say that $\mathcal{D}$ is $\mathcal{H}$-realizable, if there exists $h^\star \in \mathcal{H}$ such that $h^\star(x) = y$ holds with probability 1 for $(x, y) \sim \mathcal{D}$. Following the standard definition of *probably approximately correct* (PAC) learning (Valiant, 1984), we say that a learning algorithm $\mathcal{A}$ (ε, δ)-PAC learns $\mathcal{H}$ with sample complexity $m(\varepsilon, \delta)$ if for all $\mathcal{H}$-realizable distributions $\mathcal{D}$ over $\mathcal{X} \times \mathbb{R}$, and for $S \sim \mathcal{D}^{m(\varepsilon,\delta)}$, it holds with probability at least $1-\delta$ that $\mathcal{L}_{\mathcal{D}}(\mathcal{A}(S)) \le \varepsilon$. We say that a learning algorithm $\mathcal{A}$ (ε, δ)-PAC learns $\mathcal{H}$ under distribution $\mathcal{P}$ (over $\mathcal{X}$) if the learning guarantee holds for all $\mathcal{H}$-realizable $\mathcal{D}$ with the marginal over $\mathcal{X}$ being $\mathcal{P}$. In particular, we use $\mathcal{U}$ to denote the uniform distribution over $\mathcal{X}$.

2.1. Fourier Analysis and the Low-Degree Algorithm

Any function $f : \{-1, 1\}^n \to \mathbb{R}$, has a unique Fourier representation given as $\sum_{T\subseteq[n]} \hat{f}(T)\chi_T(x)$ where $\chi_T(x) := \prod_{j\in T} x_i$. The degree of f, denoted $\deg(f)$, is the largest k such that $\hat{f}(T) \neq 0$ for some T with $|T| = k$. The ℓ_2 norm of f under the uniform distribution is defined as $\|f\|_2^2 := \mathbb{E}_{x\sim\mathcal{U}}\, f(x)^2$ (O'Donnell, 2014).

We define the ℓ_2 sensitivity of f at x as $\mathsf{sen}_f(x) := \frac{1}{4}\sum_{i\in[n]}(f(x) - f(x^{\oplus i}))^2$, where $x^{\oplus i}$ is x with the i-th bit flipped; the scaling factor of $1/4$ means that for $f : \{-1, 1\}^n \to \{-1, 1\}$, sensitivity can be interpreted as $\mathsf{sen}_f(x) = |\{i : f(x) \neq f(x^{\oplus i})\}|$. The average ℓ_2^2 sensitivity $\mathsf{AS}(f)$ is defined as $\mathbb{E}_{x\sim\mathcal{U}}\,[\mathsf{sen}_f(x)]$. For any x, let $N_\rho(x)$ denote the distribution obtained by flipping each coordinate of x with probability $(1-\rho)/2$. The ρ-noise sensitivity of f is $\mathsf{NS}_\rho(f) := \mathbb{E}_{x\sim\mathcal{U}, y\sim N_\rho(x)}\, \frac{1}{4}(f(x) - f(y))^2$.

A connection between noise sensitivity and Fourier concentration was first observed in Klivans et al. (2004). We state this connection below, along with other basic facts about Fourier coefficients.

Claim 2.1 *[See Klivans et al. (2004)] The following properties hold for all $f : \{-1,1\}^n \to \mathbb{R}$:*

- $\|f\|_2^2 = \sum_{T\subseteq[n]} \hat{f}(T)^2$, *and*
- $\mathsf{NS}_\rho(f) = \sum_{T\subseteq[n]} \frac{1}{2}(1-\rho^{|T|})\hat{f}(T)^2$, *and hence* $\sum_{T:|T|>d} \hat{f}(T)^2 \leq 2\cdot\mathsf{NS}_\rho(f)/(1-\rho^d)$.

We also need a bound on the average sensitivity of a single halfspace which is known to be $O(\sqrt{n})$. We require a more fine-grained version from Kane (2014) which quantifies the dependence on the bias of the halfspace.

Lemma 6 (Kane (2014)) *Let $g : \mathcal{X} \to \{0,1\}$ be a halfspace: $g(x) = \mathbb{1}\{\langle w,x\rangle \leq b\}$ and $\mathbb{E}[g] = p$. Then, $\mathsf{AS}(g) = O(p\sqrt{n}\log(1/p))$.*

Proof Without loss of generality, we can assume that the coefficients of w are positive. This makes g a monotone function which is non-decreasing in each coordinate. Now, for $i \in [n]$, and $x \sim \mathcal{U}$,

$$\mathbb{E}[x_i g(x)] = \frac{1}{2}\sum_{x\in\mathcal{X}} \left(x_i g(x) - x_i g(x^{\oplus i})\right) = \mathbb{E}[(g(x) - g(x^{\oplus i}))^2],$$

where the second equality is due to the non-decreasing nature of g and that $g(x)$ takes values in $\{0,1\}$. Therefore,

$$\mathsf{AS}(g) = \tfrac{1}{4}\,\mathbb{E}_x\left[g(x)\textstyle\sum_{i=1}^n x_i\right],$$

the claim now follows from Lemma 6 of Kane (2014). ■

Low-degree algorithm. We recall the standard *low-degree* algorithm and its guarantees for learning hypothesis classes that exhibit low-degree Fourier concentration (see e.g., Klivans et al. (2004) for details). For any hypothesis class $\mathcal{H} \subseteq (\mathcal{X} \to \mathbb{R})$, let $C_{\mathcal{H}} := \sup_{h\in\mathcal{H},x\in\mathcal{X}} h(x)$.

Lemma 7 *For hypothesis class $\mathcal{H} \subseteq (\mathcal{X} \to \mathbb{R})$ such that $\sum_{T:|T|>d} \hat{h}(T)^2 \leq \varepsilon$ for all $h \in \mathcal{H}$, there exists an $(O(\varepsilon),\delta)$-PAC learning algorithm for $\mathcal{H}$ with $O(n^d C_{\mathcal{H}}^2 \log(1/\delta)/\varepsilon)$ sample and time complexity.*

The algorithm operates by performing *polynomial regression*, that is, linear regression in the basis of monomials of degree at most d. The algorithm achieves the desired error because $g(x) := \sum_{T:|T|\leq d} \hat{h}(T)\chi_T(x)$ is such that $\|g-h\|_2^2 = \sum_{T:|T|>d} \hat{h}(T)^2 \leq \varepsilon/2$, and hence there exists a good solution to the polynomial regression problem.

3. Learning over Uniform Distribution

In this section we provide a learning algorithm for k-sparsely activated networks under the uniform distribution.

Theorem 8 *There exists an (ε,δ)-PAC learning algorithm for $\mathcal{H}_{n,s,k}^{W,B}$ with respect to the uniform distribution over $\mathcal{X}$ that has sample complexity and run-time $O(n^d k^2 (W\sqrt{n}+B)^2 \log(1/\delta)/\varepsilon)$ for $d = \Theta((k^8 W^4 \log(ns)^4 + k^6 B^4 \log s)/\varepsilon^2)$*

At a high level, we show that all hypotheses in $\mathcal{H}_{n,s,k}^{W,B}$ exhibit low-degree Fourier concentration and hence can be learned over the uniform distribution using the low-degree algorithm (Lemma 7). To show Fourier concentration, we bound the noise sensitivity of sparse-activated networks by first showing a bound on the average sensitivity and then converting this to a bound on noise sensitivity.

Lemma 9 *For all $h \in \mathcal{H}_{n,s,k}^{W,B}$, it holds that $\mathsf{AS}(h) \le O\left(k^4 W^2 \sqrt{n} \log(ns) + k^3 B^2 \sqrt{\log s}\right)$.*

Proof Consider $h \in \mathcal{H}_{n,s,k}^{W,B}$ given as $h(x) = \sum_{j=1}^{s} u_j \sigma(\langle w_j, x\rangle - b_j)$. For any $R \subseteq [s]$, let $\ell_R(x) = \langle w^R, x\rangle - b^R$ for $w^R := \sum_{j\in R} u_j w_j$ and $b^R := \sum_{j \in R} u_j b_j$. Since $\max_j |u_j| \cdot \max_j \|w_j\| \le W$ and $\max_j |u_j| \cdot \max_j |b_j| \le B$, it follows that $\|w^R\| \le |R| \cdot W$ and $|b^R| \le |R| \cdot B$. For any $x \in \mathcal{X}$, let $R_x \subseteq [s]$ be defined as $R_x := \{j \in [s] : \langle w_j, x\rangle > b_j\}$. Since h is k-sparse, we have that $|R_x| \le k$ and hence $\|w^{R_x}\| \le kW$ and $|b^{R_x}| \le kB$. It is easy to see that for $h \in \mathcal{H}_{n,s,k}^{W,B}$ it holds that $h(x) = \ell_{R_x}(x)$ for all $x \in \mathcal{X}$.

The average sensitivity of h is given as

$$\begin{aligned} \mathsf{AS}(h) &= \mathbb{E}_x \left[\textstyle\sum_{i=1}^{n} \frac{1}{4} \left(h(x) - h(x^{\oplus i}\right)^2\right] \\ &= \mathbb{E}_x \left[\textstyle\sum_{i=1}^{n} \frac{1}{4} \left(h(x) - h(x^{\oplus i})\right)^2 \cdot \mathbb{1}\{R_x = R_{x^{\oplus i}}\}\right] && \text{(U)} \\ &\quad + \mathbb{E}_x \left[\textstyle\sum_{i=1}^{n} \frac{1}{4} \left(h(x) - h(x^{\oplus i}\right)^2 \cdot \mathbb{1}\{R_x \ne R_{x^{\oplus i}}\}\right] && \text{(V)} \end{aligned}$$

We bound term (U) as,

$$\begin{aligned} \text{(U)} &= \mathbb{E}_x \left[\textstyle\sum_{i=1}^{n} \frac{1}{4} \left(\ell_{R_x}(x) - \ell_{R_x}(x^{\oplus i})\right)^2 \cdot \mathbb{1}\{R_x = R_{x^{\oplus i}}\}\right] \\ &\le \mathbb{E}_x \left[\textstyle\sum_{i=1}^{n} \frac{1}{4} \left(w_i^{R_x}\right)^2\right] = \mathbb{E}_x \left[\frac{1}{4}\|w^{R_x}\|^2\right] \le \frac{k^2 W^2}{4}\,. \end{aligned}$$

We bound term (V) as follows using the inequality $(a-b)^2 \le 2(w^2 + b^2)$,

$$\begin{aligned} \text{(V)} &= n\,\mathbb{E}_{x,i} \left[\frac{1}{4} \left(h(x) - h(x^{\oplus i}\right)^2 \cdot \mathbb{1}\{R_x \ne R_{x^{\oplus i}}\}\right] \\ &\le n\,\mathbb{E}_{x,i} \left[\frac{1}{2} \left(h(x)^2 + h(x^{\oplus i})^2\right) \cdot \mathbb{1}\{R_x \ne R_{x^{\oplus i}}\}\right] \\ &= n\,\mathbb{E}_{x,i} \left[h(x)^2 \cdot \mathbb{1}\{R_x \ne R_{x^{\oplus i}}\}\right] \qquad \text{(by symmetry)} \end{aligned}$$

For $g_j(x) := \mathbb{1}\{\langle w_j, x\rangle > b_j\}$, we have that

$$\Pr_{x,i}[R_x \ne R_{x^{\oplus i}}] \le \frac{1}{n} \textstyle\sum_{j=1}^{s} \sum_{i=1}^{n} \Pr_x[g_j(x) \ne g_j(x^{\oplus i})] = \frac{1}{n} \sum_{j=1}^{s} \mathsf{AS}(g_j)$$

Note that $\sum_{j=1}^{s} g_j(x) \le k$ (by k-sparsity), and hence for $p_j = \mathbb{E}_x[g_j(x)]$, we have that $\sum_{j=1}^{s} p_j \le k$. From Lemma 6, we have that $\mathsf{AS}(g_j) \le p_j \sqrt{n \log(1/p_j)}$. Thus,

$$\Pr_{x,i}[R_x \ne R_{x^{\oplus i}}] \le \frac{1}{n} \sum_{j=1}^{s} p_j \sqrt{n \log(1/p_j)} \le \frac{k\sqrt{\log(s/k)}}{\sqrt{n}}$$

where we use concavity of $p\sqrt{\log(1/p)}$ for $p \in (0,1)$. For each $R \subseteq [s]$ with $|S| \leq k$, we have by Hoeffding bound that for some sufficiently large c and $t = ckW\sqrt{\log(n^k s)} + kB$,

$$\begin{aligned}
&\Pr_{x\sim\mathcal{U}}\left[\exists R \subseteq [s] : |R| \leq k \text{ and } \left|\langle w^R, x\rangle - b^R\right| > t\right] \\
&\leq \Pr_{x\sim\mathcal{U}}\left[\exists R \subseteq [s] : |R| \leq k \text{ and } \left|\langle w^R, x\rangle\right| > t - \left|b^R\right|\right] \\
&\leq 2n^k \exp\left(\frac{-(t-|b^R|)^2}{2\|w^R\|^2}\right) \leq \frac{1}{(ns)^4},
\end{aligned}$$

Hence, in particular we have that

$$\Pr_x[|\ell_{R_x}(x)| \geq ck^{1.5}W\sqrt{\log(ns)} + kB] \leq \frac{1}{n^4 s^4}$$

And for all x, we also have that $|\ell_{R_x}(x)| \leq kW\sqrt{n} + kB$ holds with probability 1. Thus, we can upper bound (V) as,

$$\begin{aligned}
(\text{V}) &\leq n \cdot \left[\left(\frac{k\sqrt{\log(s/k)}}{\sqrt{n}} - \frac{1}{(ns)^4}\right)(ck^{1.5}W\sqrt{\log(ns)} + kB)^2 + \frac{1}{(ns)^4}\cdot(kW\sqrt{n}+kB)^2\right] \\
&\leq O\left(k^4W^2\sqrt{n}\log(ns) + k^3B^2\sqrt{\log s}\right)
\end{aligned}$$

Combining the bounds on (U) and (V) completes the proof. ■

Next, we can use the bound on average sensitivity to bound the noise sensitivity of functions in $\mathcal{H}^{W,B}_{n,s,k}$. To do so we use an argument attributed to Peres for converting bounds on average sensitivity to bounds on noise sensitivity, allowing us to get better low-degree approximations.

Lemma 10 *For any $h \in \mathcal{H}^{B}_{n,s,k}$,*

$$\mathsf{NS}_\rho(h) = \sqrt{(1-\rho)} \cdot O(k^4W^2\log^2(ns/(1-\rho)) + k^3B^2\sqrt{\log s}).$$

The proof of Lemma 10 is provided in Appendix B.

Proof of Theorem 8 We combine Claim 2.1, Lemma 7 and Lemma 10. Fix an error parameter ε. Then, by Lemma 10, there is a constant $c > 0$, such that for

$$1 - \rho = c\varepsilon^2 \cdot \min\left\{\frac{\log(knsW^2/\varepsilon)}{k^8W^4\log^4(ns)}, \frac{1}{k^6B^4\log s}\right\}$$

any $h \in \mathcal{H}^{W,B}_{n,s,k}$, satisfies

$$\mathsf{NS}_\rho(h) \leq \varepsilon/3$$

Thus, we can choose a suitable $d = \Theta((k^8W^4\log(ns)^4 + k^6B^4\log s)/\varepsilon^2)$, such that by Claim 2.1,

$$\textstyle\sum_{T:|T|>d} \hat{f}(T)^2 \leq \frac{\varepsilon}{3(1-\rho^d)} \leq \frac{\varepsilon}{d(1-\rho)} \leq \varepsilon\,.$$

Finally, note that $C_{\mathcal{H}^{W,B}_{n,s,k}} = k(W\sqrt{n} + B)$; since at most k neurons are active on any input, and each neuron can at most contribute $W\sqrt{n} + B$. Thus, the theorem now follows from combining the above with Lemma 7. The run-time and sample complexity will be $O(n^d\log(1/\delta)/\varepsilon)$ where d is as above. ■

Remark 11 *Theorem 8 can be extended to hold in case of the hypothesis class where k-sparsity need not hold for all inputs $x \in \mathcal{X}$, but holds with probability at least $1 - 1/\mathsf{poly}(n, s)$ over the input distribution, that is, $\Pr_{x \sim \mathcal{U}}[\#\{i \in [s] : \langle w_i, x\rangle + b_i > 0\} > k] \leq 1/\mathsf{poly}(n, s)$. This is by decomposing $\mathsf{AS}(h)$ into (U), (V) and a third term handling x for which the k-sparsity is violated.*

4. Lower Bounds for Learning $\mathcal{H}_{n,s,1}$

Note that the previous section implies a quasi-polynomial time learning algorithm for the class $\mathcal{H}_{n,s,1}$ of 1-sparsely activated networks. We next show that a quasi-polynomial run-time is likely necessary for learning $\mathcal{H}_{n,s,1}$ under the uniform distribution and stronger lower bounds under arbitrary distributions.

Sparse Activations Can Simulate Juntas We first show that our proposed learning algorithms for the case of the uniform distribution have near-optimal runtime under a widely believed conjecture on the hardness of learning juntas. Let $\mathcal{J}_{n,p}$ denote the set of Boolean functions $f : \{1, -1\}^n \to \{-1, 1\}$ that only depend on at most p variables.

Conjecture 12 (Hardness of learning Juntas) *(see e.g. Mossel et al. (2003); Feldman et al. (2011)) There is no (ε, δ)-PAC learning algorithm for learning $\mathcal{J}_{n,p}$ under the uniform distribution on the hypercube that runs in time $n^{o(p)}$.*

The conjecture implies that there is no learning algorithm for $\mathcal{H}_{n,s,1}$ that runs in $n^{o(\log s)}$ time.

Theorem 13 *Assuming Conjecture 12, there is no (ε, δ)-PAC learning algorithm for $\mathcal{H}_{n,s,1}^{W,B}$ for $W = \sqrt{\log_2 s}$ and $B = \log_2 s$ over $\mathcal{U}$ that runs in $n^{o(\log s)}$ time.*

Proof We show that $\mathcal{H}_{n,s,1}^{\sqrt{p},p} \supseteq \mathcal{J}_{n,p}$ for all $p \leq \lfloor \log_2 s \rfloor$, that is, for $p \leq \lfloor \log_2 s \rfloor$ any p-junta $f \in \mathcal{J}_{n,p}$ can be expressed as $\sum_{j \in [s]} u_j \sigma(\langle w_j, x\rangle + b_j)$ where $\|u\|_\infty \leq 1$ and $\|w_j\|_2 \leq \sqrt{\log_2 s}$. Suppose w.l.o.g. that f depends on $x_1, \ldots, x_p$. Let $w_1, \ldots w_{2^p}$ be distinct vectors that take all possible ± 1 values in the first p coordinates, and are 0 on other coordinates. Let $u_j = f(x)$ for any x such that $x_i = w_{ji}$ for all $i \in [p]$ and $j \in [2^p]$. Let $w_j = \mathbf{0}$ and $u_j = 0$ for all $j > 2^p$. It is now easy to verify that for all $x \in \mathcal{X}$,

$$f(x) = \textstyle\sum_{j \in [2^p]} u_j \sigma(\langle w_j, x\rangle - p + 1), \quad \text{since } \sigma(\langle w_j, x\rangle - p + 1) = \mathbb{1}\{x_i = w_{ji} \text{ for all } i \in [p]\}$$

Thus, the theorem follows under the assumption of Conjecture 12. ■

Hardness Under Arbitrary Distributions We next show that one-sparse activation networks over $\{1, -1\}^n$ can simulate parities of size $\Omega(\sqrt{n})$. Fix an integer m, and for $S \subseteq [m]$, let $\chi_S : \{1, -1\}^m \to \{0, 1\}$ be defined by $\chi_S(y) = 1$ if and only if $\sum_{i \in S} y_i$ is even. Now, we can use the following simple identity (similar identities were used for similar purposes for example in Klivans and Sherstov (2006))

$$\chi_S(y) = \textstyle\sum_{a \in \{-m, \ldots, m\} : a \text{ even}} 2\sigma\left(\frac{1}{2} - \left(\sum_{i \in S} y_i - a\right)^2\right).$$

Note that for any $y \in \{1, -1\}^m$, at most one ReLU node is active. This is not quite enough to capture $\mathcal{H}_{n,s,1}$ as the function inside the ReLUs are not linear. To fix this, we linearize the quadratic function by increasing the dimension. For $y \in \{1,-1\}^m$, let $x(y) \in \{1,-1\}^{m \times m}$ be defined as follows:

$$x(y)_{ij} = \begin{cases} y_i & \text{if } i = j \\ y_i y_j & \text{if } i \neq j \end{cases}.$$

Let $n = m^2$ and identify $\{1,-1\}^n$ with $\{1,-1\}^{m \times m}$ in the natural way. Observe that for any $S \subseteq [m]$, $a \in [-m, m]$, there exists a vector $w_{S,a} \in \mathbb{R}^n$, $b_{S,a} \in \mathbb{R}$ such that

$$\tfrac{1}{2} - \left(\sum_{i \in S} y_i - a\right)^2 = \langle w_{S,a}, x(y) \rangle - b_{S,a}.$$

In particular, we can take $b_{S,a} = |S| + a^2 - 1/2$, and $w_{S,a}[i,j] = -1$ if $i \neq j \in [m]$ and $w_{S,a}[i,i] = 2a$. Note that $\|w_{S,a}\|_2 = O(m^{1.5}) = O(n^{3/4})$ and $|b_{S,a}| = O(m^2) = O(n)$.

In summary, there exists a distribution $\mathcal{D}$ on $\{1,-1\}^{m \times m}$ such that learning parities over $\{1,-1\}^m$ under the uniform distribution is implied by learning $\mathcal{H}^{O(m^{1.5}),O(m^2)}_{m^2,2m,1}$ under the distribution $\mathcal{D}$. The first part of Theorem 3 now follows from standard lower bounds for learning parities.

SQ Hardness Consider a class of functions, denoted by C, that maps $\mathbb{R}^n$ to $\mathbb{R}$, and let $\mathcal{D}$ be a distribution over $\mathbb{R}^n$.

In the Statistical Query (SQ) model, as described by Kearns (1998), the learner interacts with the data through an SQ oracle. For a bounded query function $\phi : \mathbb{R}^n \times \mathbb{R} \to [-1,1]$ and a tolerance $\tau > 0$, the oracle can return any value v such that the absolute difference $|v - \mathbb{E}_{x \sim D}[\phi(x, f(x))]| \leq \tau$. The goal in SQ learning is to learn an approximation to the unknown concept only using few queries as above with reasonable tolerance. We will use the following classical theorem:

Theorem 14 ((Blum et al., 1994)) *Any SQ algorithm for learning the class of parities over $\{1,-1\}^m$ within error $1/3$ under the uniform distribution over the hypercube with tolerance τ requires $\Omega(2^m \tau^2)$ queries.*

The first part of Theorem 3 follows immediately from the above and the fact that parities on m variables can be computed in $\mathcal{H}^{O(m^{1.5}),O(m^2)}_{m^2,O(m),1}$ as described.

Cryptographic Hardness We sketch the argument here. Following Chen et al. (2022a), our starting point will be the *Learning with Rounding* (LWR) problem (Banerjee et al., 2012):

Definition 15 *For moduli $p, q \in \mathbb{N}$, $w \in \mathbb{Z}_q^m$, let $f_w : \mathbb{Z}_q^m \to \mathbb{Z}_p$ by $f_w(y) := (\langle w, y \rangle \bmod q) \bmod p$.*

In the $\mathsf{LWR}_{p,q,m}$ problem the *secret* $w \in \mathbb{Z}_q^m$ is drawn uniformly at random and we are given samples of the form $(y, f_w(y))$ where y is uniform over Z_q^m. The goal is to output a hypothesis that achieves a small error in predicting the label $f_w(\cdot)$. It is conjectured that there is no $\mathsf{poly}(m,p,q)$ algorithm for $\mathsf{LWR}_{p,q,m}$.

Conjecture 16 (See Banerjee et al. (2012)) *There is no $\mathsf{poly}(p,q,m)$ run-time algorithm to solve the $\mathsf{LWR}_{p,q,m}$ with probability at least $2/3$ (over the random choice of w and the samples).*

We show that an efficient algorithm for $\mathcal{H}_{n,s,1}$ functions under arbitrary distributions on the hypercube will contradict this assumption.

Consider an instance of the $\mathsf{LWR}_{p,q,m}$ problem. First, map $y \in \mathbb{Z}_q^m$ to $z(y) \in \{1,-1\}^r$ for $r = O(m \log q)$ by considering the binary representation of the integers in y. Next, let $\lambda : [q^2m] \to [p]$ be such that $\lambda(i) = (i \mod q) \mod p$. Note that for every $w \in \mathbb{Z}_q^m$, we can find a vector $v(w) \in \mathbb{R}^r$ such that $\langle v(w), z(y)\rangle = \langle w, y\rangle$. Then,

$$f_w(y) = \lambda(\langle v(w), z(y)\rangle).$$

Now, observe that we can write

$$\lambda(\langle v(w), z(y)\rangle) = \textstyle\sum_{a\in[q^2m]} 2\lambda(a)\sigma\left(\frac{1}{2} - (\langle v(w), z(y)\rangle - a))^2\right).$$

Note that in the conversion $z(y) \in \{1,-1\}^r$ and $v(w) \in \mathbb{R}^r$. Further, for any input y, only one of the ReLUs will be active. However, the above is not quite in $\mathcal{H}_{n,s,1}$ as we have a quadratic function inside the ReLU. Just as we did for parities, we can fix this issue by linearizing the quadratic form. Let $n = r^2$, and define $x(y) \in \{1,-1\}^{r\times r}$ by setting $x(y)_{ij} = z(y)_i z(y)_j$ if $i \neq j$ and $x(y)_{ii} = z(y)_i$. Then, just as in our argument for parities, there exists a *lifted* weight vector $W_{w,a} \in \{1,-1\}^n$ and $b_{w,a}$ such that

$$\frac{1}{2} - (\langle v(w), z(y)\rangle - a))^2 = \langle W_{w,a}, x(y)\rangle - b_{w,a}.$$

In addition, it is easy to check that $\|W_{w,a}\|_2, |b_{w,a}| = \mathsf{poly}(q,m)$. In particular, we get that for every $w \in \mathbb{Z}_q^m$, there exists a function F_w in $\mathcal{H}_{r^2,O(q^2m),1}$ such that for every $y \in \mathbb{Z}_q^m$,

$$f_w(y) = F_w(x(y)),$$

where $x(y) \in \{1,-1\}^{r^2}$ is the embedding as defined above and in showing SQ hardness. The second part of Theorem 3 now follows from the conjectured hardness of $\mathsf{LWR}_{p,q,m}$; we omit the minor details.

5. Learning under General Distributions

We now show the statistical advantage associated with sparsely activated neural networks over general distributions. In particular, we show that

Theorem 17 *There exists a (ε,δ)-PAC learning algorithm for any $\mathcal{H}^{W,B}_{n,s,k}$ with sample complexity $m(\varepsilon,\delta) = O\Big(\frac{(WR+B)^2 ksn \log(\frac{k(R+B)}{\varepsilon}) + \log(\frac{1}{\delta})}{\varepsilon^2}\Big)$.*

This result even holds in a more general setting where the input space $\mathcal{X} \subset \mathbb{R}^n$ and $\|x\| \leq R$ for all $x \in \mathcal{X}$. To begin with we will again consider the class of 1-sparsely activated networks, i.e., $\mathcal{H}^{W,B}_{n,s,1}$. We will discuss extensions to $\mathcal{H}^{W,B}_{n,s,k}$ towards the end of the section.

We use Rademacher complexity to establish the bound in Theorem 17. Given a set of examples $S = \{x_1, x_2, \dots, x_m\}$ the empirical Rademacher complexity (Shalev-Shwartz and Ben-David, 2014) is defined as $\mathcal{R}_{\mathcal{H}}(S) := \mathbb{E}_\zeta\left[\max_{h\in\mathcal{H}} \frac{1}{m}\sum_{i=1}^m \zeta_i h(x_i)\right]$, where $\zeta_1, \dots, \zeta_m$ are $\{-1,+1\}$ valued Rademacher random variables. For $\mathcal{H}$, let $C_{\mathcal{H}} := \sup_{h\in\mathcal{H}, x\in\mathcal{X}} h(x)$.

Lemma 18 (see Shalev-Shwartz and Ben-David (2014)) *For any class $\mathcal{H}$ mapping $\mathcal{X}$ to $\mathbb{R}$, there exists an (ε, δ)-PAC learning algorithm for $\mathcal{H}$ with sample complexity $m(\varepsilon, \delta)$ equal to the smallest m such that for a large enough constant c, it holds that*

$$c \cdot \left(C_{\mathcal{H}} \mathbb{E}_S[\mathcal{R}_{\mathcal{H}}(S)] + \sqrt{\frac{\log(1/\delta)}{m}} \right) \leq \varepsilon\,.$$

Theorem 17 will follow from bounding the Rademacher complexity $\mathcal{R}_{\mathcal{H}}(S)$. Recall that in the absence of any sparsity assumption, existing results (Anthony et al., 1999) on the Rademacher complexity of 1-hidden layer ReLU networks with input dimensionality n and s hidden units lead to a bound of $\frac{(WR+B)s}{\sqrt{m}}$.[1] We will show that the main statistical advantage that comes from sparsity is that the dependence on the number of hidden units s can be made sub-linear, albeit at the expense of an explicit dependence on the input dimensionality n. In particular we will prove the following theorem.

Theorem 19 *It holds that*

$$\mathcal{R}_{\mathcal{H}^{W,B}_{n,s,1}}(S) \;\leq\; \frac{(WR+B)\sqrt{sn\log(m(R+B))}}{\sqrt{m}}. \tag{1}$$

Proof For a given hypothesis $u, w_1, \ldots, w_s \in \mathcal{H}^{W,B}_{n,s,1}$ and for any $j \in [s]$, let I_j be the subset of the m examples that activate neuron j, i.e., $I_j = \{i \in [m] : \langle w_j, x_i \rangle - b_j \geq 0\}$. Since each I_j is determined by a halfspace in n dimensions, by the Sauer-Shelah lemma (Shalev-Shwartz and Ben-David, 2014) there can be at most $O(m^n)$ such subsets.

Next, we have

$$\begin{aligned}
\mathcal{R}_{\mathcal{H}^{W,B}_{n,s,1}}(S) \;&:=\; \mathbb{E}_\zeta\left[\max\nolimits_{u,w_1,\ldots,w_s \in \mathcal{H}^{W,B}_{n,s,1}} \tfrac{1}{m} \textstyle\sum_{i=1}^m \zeta_i \sum_{j=1}^s u_j \sigma(\langle w_j, x_i\rangle - b_j)\right] && (2)\\
&=\; \mathbb{E}_\zeta\left[\max\nolimits_{u,w_1,\ldots,w_s \in \mathcal{H}^{W,B}_{n,s,1}} \tfrac{1}{m} \textstyle\sum_{j=1}^s \sum_{i\in I_j} \zeta_i u_j (\langle w_j, x_i\rangle - b_j)\right] && (3)\\
&\leq\; \mathbb{E}_\zeta\left[\max\nolimits_{u,w_1,\ldots,w_s \in \mathcal{H}^{W,B}_{n,s,1}} \tfrac{1}{m} \textstyle\sum_{j=1}^s \sum_{i\in I_j} \zeta_i u_j \, \langle w_j, x_i\rangle\right]\\
&\quad+\; \mathbb{E}_\zeta\left[\max\nolimits_{u,w_1,\ldots,w_s \in \mathcal{H}^{W,B}_{n,s,1}} \tfrac{1}{m} \textstyle\sum_{j=1}^s \sum_{x_i\in I_j} \zeta_i u_j b_j\right] && (4)
\end{aligned}$$

We will bound the above two terms separately via standard concentration inequalities. For the second term note that for any fixed I_j, the random variable $\sum_{i\in I_j} \zeta_i$ is sub-Gaussian with norm $O(\sqrt{|I_j|})$. Hence we for any fixed I_j the following holds (Vershynin, 2018)

$$\mathbb{P}\left[\left|\textstyle\sum_{i\in I_j} \zeta_i\right| > t\sqrt{|I_j|}\right] \leq 2e^{-\frac{t^2}{c}}, \tag{5}$$

where $c > 0$ is an absolute constant. Via the union bound we get that with probability at least $1 - O(m^n e^{-t^2/c})$, all sets I_j simultaneously satisfy the above inequality.

Hence we get the following bound on the second term.

$$\mathbb{E}_\zeta\left[\max_{u,w_1,\ldots,w_s \in \mathcal{H}^{W,B}_{n,s,1}} \frac{1}{m} \sum_{j=1}^s \sum_{x_i \in I_j} \zeta_i u_j b_j\right] \;\leq\; \frac{1}{m}\sum_{j=1}^s t|u_j b_j|\sqrt{|I_j|} + O(m^n e^{-t^2/c})\frac{1}{m}\sum_{j=1}^s |u_j b_j||I_j|. \tag{6}$$

[1] Better bounds are possible under stronger assumptions on the network weights (Wei et al., 2019).

From the fact that the activations are 1-sparse we get that $\sum_{j=1}^{s} |I_j| = m$. This implies that $\sum_{j=1}^{s} \sqrt{|I_j|} \leq \sqrt{sm}$. Furthermore, using the fact that $\max_j |u_j b_j| \leq B$ we get

$$\mathbb{E}_\zeta \left[\max_{u,w_1,\ldots,w_s \in \mathcal{H}_{n,s,1}^{W,B}} \frac{1}{m} \sum_{j=1}^{s} \sum_{x_i \in I_j} \zeta_i u_j b_j \right] \leq \frac{1}{\sqrt{m}} tB\sqrt{s} + O(m^n e^{-t^2/c})B. \tag{7}$$

Setting $t = 2\sqrt{nc\log(mB)}$ we get that the second term is bounded by

$$\mathbb{E}_\zeta \left[\max_{u,w_1,\ldots,w_s \in \mathcal{H}_{n,s,1}^{W,B}} \frac{1}{m} \sum_{j=1}^{s} \sum_{x_i \in I_j} \zeta_i u_j b_j \right] \leq \frac{4B\sqrt{snc\log(mB)}}{\sqrt{m}}. \tag{8}$$

Similarly, we next bound the first term. Note that for any fixed I_j, and any coordinate $p \in [n]$, sub-Gaussian concentration (Vershynin, 2018) implies that

$$\mathbb{P}\left(\left| \sum_{i \in I_j} \zeta_i x_{i,p} \right| > t\sqrt{\sum_{i \in I_j} x_{i,p}^2} \right) \leq 2e^{-\frac{t^2}{c}}. \tag{9}$$

■

Via a union bound over all the n coordinates and all possible subsets I_j we get that with probability at least $1 - 2nm^n e^{-\frac{t^2}{c}}$, all sets I_j simultaneously satisfy

$$\left\| \sum_{i \in I_j} \zeta_i x_i \right\| \leq tR\sqrt{|I_j|}. \tag{10}$$

Using the above we can bound the first term as

$$\begin{aligned}
\mathbb{E}_\zeta \left[\max_{u,w_1,\ldots,w_s \in \mathcal{H}_{n,s,1}^{W,B}} \frac{1}{m} \sum_{j=1}^{s} \sum_{x_i \in I_j} \zeta_i u_j \langle w_j, x_i \rangle \right] &\leq \mathbb{E}_\zeta \left[\max_{u,w_1,\ldots,w_s \in \mathcal{H}_{n,s,1}^{W,B}} \frac{1}{m} \sum_{j=1}^{s} \left\langle u_j w_j, \sum_{i \in I_j} \zeta_i x_i \right\rangle \right] && (11) \\
&\leq \frac{1}{m} \sum_{j=1}^{s} |u_j| \|w_j\| \mathbb{E}_\zeta \left[\left\| \sum_{i \in I_j} \zeta_i x_i \right\| \right] && (12) \\
&\leq \frac{W}{m} \sum_{j=1}^{s} \left(tR\sqrt{|I_j|} + 2nm^n e^{-\frac{t^2}{c}} R|I_j| \right). && (13)
\end{aligned}$$

Recall from above that $\sum_{j=1}^{s} \sqrt{|I_j|} \leq \sqrt{sm}$. Furthermore, setting $t = 2\sqrt{n\log(mR)}$ we get that the first term is bounded by

$$\mathbb{E}_\zeta \left[\max_{u,w_1,\ldots,w_s \in \mathcal{H}_{n,s,1}^{W,B}} \frac{1}{m} \sum_{j=1}^{s} \sum_{i \in I_j} \zeta_i u_j \langle w_j, x_i \rangle \right] \leq 4\frac{WR\sqrt{ns\log(mR)}}{\sqrt{m}}. \tag{14}$$

Combining the bounds for the first and the second terms, we get the desired claim.

Generalization to k-sparsely activated networks. The above analysis extends in a straightforward manner to the class $\mathcal{H}_{n,s,k}$, i.e., the class of networks where each input activates at most k hidden units.

To extend the bound in Theorem 19 we note that using the fact that k-sparsity implies that $\sum_{j\in[s]}|I_j| \leq km$ we get that

$$\mathcal{R}_{\mathcal{H}_{n,s,k}^{W,B}}(S) \;\leq\; \frac{(WR+B)\sqrt{snk\log(km(R+B))}}{\sqrt{m}}. \tag{15}$$

Note that in contrast to the classical bounds on Rademacher complexity of general norm bounded 1-layer neural networks the bound in Theorem 19 above has a sub-linear dependence on s. However we incur an explicit dependency on the input dimensionality.

We suspect that this is a limitation of our proof technique and conjecture that the right dependence should not have any explicit dependence on the input dimension n.

Conjecture 20 *The class $\mathcal{H}_{n,s,k}^{W,B}$ of k-sparsely activated neural networks satisfies*

$$\mathcal{R}_{\mathcal{H}_{n,s,k}^{W,B}}(S) \leq \tfrac{(WR+B)\sqrt{sk}}{\sqrt{m}}. \tag{16}$$

6. Discussion & Future Directions

Motivated by the empirical phenomenon of activation sparsity in MLP layers of large transformer models, in this work we proposed and studied the problem of PAC learning the class of sparsely activated neural networks. This is a novel concept class with many interesting properties. The form of input-dependent sparsity present in this class of functions makes it distinct from the typical sparse function classes studied in literature. The main conceptual insight from our work is that despite the empirical challenges in leveraging sparsity, activation sparsity can provably provide both computational and statistical benefits.

Several open questions come out of our work. While we provide algorithms with near optimal running time for the case of the uniform distribution, it would be interesting to design learning algorithms under arbitrary distributions that are provably better than the $O((ns)^n)$-time algorithms that exist for general 1-layer ReLU networks (Goel et al., 2020). As mentioned in Section 5 we strongly suspect that the dependence on the input dimension n in the Rademacher complexity bound of Theorem 19 is suboptimal. While we primarily considered networks that are sparsely activated for all inputs, it might be interesting to also consider sparsely activated with high probability over input distributions, as we briefly alluded to in Remark 11 although in that case, the probability of not being sparsely activated was very small. Finally, it would be interesting to explore practical algorithms for leveraging sparsity based on our theoretical insights.

Acknowledgments

We thank anonymous reviewers for their comments that helped improve the presentation.

References

Rohan Anil, Gabriel Pereyra, Alexandre Passos, Robert Ormandi, George E Dahl, and Geoffrey E Hinton. Large scale distributed neural network training through online distillation. *arXiv preprint arXiv:1804.03235*, 2018.

Martin Anthony, Peter L Bartlett, Peter L Bartlett, et al. *Neural network learning: Theoretical foundations*, volume 9. cambridge university press Cambridge, 1999.

Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 719–737. Springer, 2012.

Ron Banner, Yury Nahshan, and Daniel Soudry. Post training 4-bit quantization of convolutional networks for rapid-deployment. *Advances in Neural Information Processing Systems*, 32, 2019.

Avrim Blum, Merrick Furst, Jeffrey Jackson, Michael Kearns, Yishay Mansour, and Steven Rudich. Weakly learning dnf and characterizing statistical query learning using fourier analysis. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 253–262, 1994.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Sitan Chen, Aravind Gollakota, Adam R. Klivans, and Raghu Meka. Hardness of noise-free learning for two-hidden-layer neural networks. In *Neural Information Processing Systems (NeurIPS)*, 2022a. URL http://papers.nips.cc/paper_files/paper/2022/hash/45a7ca247462d9e465ee88c8a302

Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*, 2022b.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *arXiv preprint arXiv:2009.14794*, 2020.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

Róbert Csordás, Kazuki Irie, and Jürgen Schmidhuber. Approximating two-layer feedforward networks for efficient transformers. *arXiv preprint arXiv:2310.10837*, 2023.

Harry Dong, Beidi Chen, and Yuejie Chi. Towards structured sparsity in transformers for efficient inference. In *Workshop on Efficient Systems for Foundation Models@ ICML2023*, 2023.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Nelson Elhage, Tristan Hume, Catherine Olsson, Neel Nanda, Tom Henighan, Scott Johnston, Sheer ElShowk, Nicholas Joseph, Nova DasSarma, Ben Mann, Danny Hernandez, Amanda Askell, Kamal Ndousse, Andy Jones, Dawn Drain, Anna Chen, Yuntao Bai, Deep Ganguli, Liane Lovitt, Zac Hatfield-Dodds, Jackson Kernion, Tom Conerly, Shauna Kravec, Stanislav Fort, Saurav Kadavath, Josh Jacobson, Eli Tran-Johnson, Jared Kaplan, Jack Clark, Tom Brown, Sam McCandlish, Dario Amodei, and Christopher Olah. Softmax linear units. *Transformer Circuits Thread*, 2022. https://transformer-circuits.pub/2022/solu/index.html.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *The Journal of Machine Learning Research*, 23(1): 5232–5270, 2022.

Vitaly Feldman, Homin K. Lee, and Rocco A. Servedio. Lower bounds and hardness amplification for learning shallow monotone formulas. In *Conference on Learning Theory (COLT)*, volume 19 of *JMLR Proceedings*, pages 273–292. JMLR.org, 2011. URL http://proceedings.mlr.press/v19/feldman11a/feldman11a.pdf.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning*, pages 10323–10337. PMLR, 2023.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913*, 2020.

Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. A survey of quantization methods for efficient neural network inference. In *Low-Power Computer Vision*, pages 291–326. Chapman and Hall/CRC, 2022.

Surbhi Goel, Aravind Gollakota, and Adam Klivans. Statistical-query lower bounds via functional gradients. *Advances in Neural Information Processing Systems*, 33:2147–2158, 2020.

Matteo Grimaldi, Darshan C Ganji, Ivan Lazarevich, and Sudhakar Sah. Accelerating deep neural networks via semi-structured activation sparsity. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1179–1188, 2023.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Hrayr Harutyunyan, Ankit Singh Rawat, Aditya Krishna Menon, Seungyeon Kim, and Sanjiv Kumar. Supervision complexity and its role in knowledge distillation. *arXiv preprint arXiv:2301.12245*, 2023.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Daniel M. Kane. The average sensitivity of an intersection of half spaces. In *Symposium on Theory of Computing (STOC)*, pages 437–440. ACM, 2014. doi: 10.1145/2591796.2591798. URL https://doi.org/10.1145/2591796.2591798.

Michael J. Kearns. Efficient noise-tolerant learning from statistical queries. *J. ACM*, 45(6):983–1006, 1998. doi: 10.1145/293347.293351. URL https://doi.org/10.1145/293347.293351.

Adam R. Klivans and Alexander A. Sherstov. Cryptographic hardness for learning intersections of halfspaces. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2006), 21-24 October 2006, Berkeley, California, USA, Proceedings*, pages 553–562. IEEE Computer Society, 2006. doi: 10.1109/FOCS.2006.24. URL https://doi.org/10.1109/FOCS.2006.24.

Adam R. Klivans, Ryan O'Donnell, and Rocco A. Servedio. Learning intersections and thresholds of halfspaces. *J. Comput. Syst. Sci.*, 68(4):808–840, 2004. doi: 10.1016/J.JCSS.2003.11.002. URL https://doi.org/10.1016/j.jcss.2003.11.002.

Guillaume Lample, Alexandre Sablayrolles, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Large memory layers with product keys. *Advances in Neural Information Processing Systems*, 32, 2019.

Zonglin Li, Chong You, Srinadh Bhojanapalli, Daliang Li, Ankit Singh Rawat, Sashank J. Reddi, Ke Ye, Felix Chern, Felix X. Yu, Ruiqi Guo, and Sanjiv Kumar. The lazy neuron phenomenon: On emergence of activation sparsity in transformers. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=TJ2nxciYCk-.

Zichang Liu, Jue Wang, Tri Dao, Tianyi Zhou, Binhang Yuan, Zhao Song, Anshumali Shrivastava, Ce Zhang, Yuandong Tian, Christopher Re, and Beidi Chen. Deja vu: Contextual sparsity for efficient LLMs at inference time. In *International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pages 22137–22176. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/liu23am.html.

Iman Mirzadeh, Keivan Alizadeh, Sachin Mehta, Carlo C Del Mundo, Oncel Tuzel, Golnoosh Samei, Mohammad Rastegari, and Mehrdad Farajtabar. Relu strikes back: Exploiting activation sparsity in large language models. *arXiv preprint arXiv:2310.04564*, 2023.

Elchanan Mossel, Ryan O'Donnell, and Rocco A. Servedio. Learning juntas. In *Symposium on Theory of Computing (STOC)*, pages 206–212. ACM, 2003. doi: 10.1145/780542.780574. URL https://doi.org/10.1145/780542.780574.

Ryan O'Donnell. *Analysis of boolean functions*. Cambridge University Press, 2014.

Ze Peng, Lei Qi, Yinghuan Shi, and Yang Gao. Theoretical explanation of activation sparsity through flat minima and adversarial robustness. *arXiv preprint arXiv:2309.03004*, 2023.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

Kai Shen, Junliang Guo, Xu Tan, Siliang Tang, Rui Wang, and Jiang Bian. A study on relu and softmax in transformer. *arXiv preprint arXiv:2302.06461*, 2023.

Sainbayar Sukhbaatar, Edouard Grave, Guillaume Lample, Herve Jegou, and Armand Joulin. Augmenting self-attention with persistent memory. *arXiv preprint arXiv:1907.01470*, 2019.

Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

Colin Wei, Jason Lee, Qiang Liu, and Tengyu Ma. On the margin theory of feedforward neural networks, 2019. URL `https://openreview.net/forum?id=HJGtFoC5Fm`.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297, 2020.

Appendix A. Example of a Sparsely Activated Network without Weight Sparsity

There are interesting functions (beyond juntas/parities) that are sparsely activated but do not have weight sparsity. E.g.: suppose $\log_2 s < n$. Consider $b = \log_2 s$, $q = n - b$, and look at $F : \{-1,1\}^b \times \{-1,1\}^q \to \mathbb{R}$, of the form $\sum_{\alpha \in \{1,-1\}^b} \sigma(\langle w_\alpha, y\rangle + \Gamma \cdot (\langle x, \alpha\rangle - b))$, where the input is (x, y). When $\Gamma = \sqrt{q}$, this network is 1-sparsely activated for all inputs, and when $\Gamma = \Theta(\sqrt{\log s})$, the function is 1-sparse with probability $1 - 1/\mathsf{poly}(s)$ under the uniform distribution on $\{-1,1\}^{b+q}$. Remark 11 shows that our results continue to hold in such a setting. Intuitively, such functions are similar to Indexing; they return the function $\sigma(\langle w_x, y\rangle)$ for all (or most) of the input space, where w_x can depend arbitrarily on the x part of the input.

Appendix B. Proof of Lemma 10

Proof of Lemma 10 Given a $\rho \in [-1,1]$, let $r = \lfloor 2/(1-\rho) \rfloor$. We describe an alternate way to sample $(x, N_\rho(x))$. First sample $z \in \{\pm 1\}^n$ uniformly at random and partition the n coordinates of z into the r buckets $\{A_e \subseteq [n]\}_{e=1}^r$ at random (each coordinate is included in exactly one of these buckets uniformly and independently). For each A_e, sample $v_e \in \{\pm 1\}$ uniformly at random. Multiply the coordinates of A_e by v_e and concatenate all the buckets to get x. Choose one bucket b at random and flip v_b to get $v^{\oplus b}$. Multiply the coordinates of A_e by $v^{\oplus b}$ to get y. Observe that (x, y) are distributed exactly the same as $(x, N_\rho(x))$. Now, given $h(x) = \sum_{j=1}^s u_j \sigma(\langle w_j, x\rangle - b_j)$, define

$$H_z(v) = \sum_{j=1}^{s} u_j \sigma(\langle w_j', v\rangle - b_j),$$

where $w_{je}' = \sum_{l \in A_e} w_{jl} z_l$. Clearly $h(x) = H_z(v)$. Hence,

$$\begin{aligned} \mathsf{NS}_\rho(h) &= \mathbb{E}[(h(x) - h(y))^2] \\ &= \frac{1}{r} \mathop{\mathbb{E}}_{z, \{A_e\}} \left(\sum_{b=1}^{r} \mathop{\mathbb{E}}_{v} (H_z(v) - H_z(v^{\oplus b}))^2 \right) \\ &= \frac{1}{r} \mathop{\mathbb{E}}_{z, \{A_e\}} [\mathsf{AS}(H_z)]. \end{aligned} \tag{17}$$

From Lemma 9,

$$\begin{aligned} &\mathsf{AS}(H_z) \le O\left(k^4 W'^2 \sqrt{r} \log(rs) + k^3 B^2 \sqrt{\log s}\right), \\ &\text{where } W' := \max_{j \in [s]} |u_j| \cdot \|w_j\|. \end{aligned}$$

To bound W' we need to bound

$$\max_{j \in [s]} \|w_j'\|_2^2 = \max_{j \in [s]} \sum_{e=1}^r \left(\sum_{i \in A_e} w_{ji} z_i\right)^2.$$

For any $j \in [s]$, we have from measure concentration

$$\begin{aligned} &\Pr_z \left[\left|\sum_{i \in A_e} w_{ji} z_i\right| > t\right] \le 2 \exp\left(-\frac{t^2}{4 \sum_{i \in A_e} w_{ji}^2}\right) \\ \implies &\Pr_z \left[\left|\sum_{i \in A_e} w_{ji} z_i\right| > 2\sqrt{2 \log(nsr) \sum_{i \in A_e} w_{ji}^2}\right] \le \frac{1}{(nsr)^2} \end{aligned}$$

Now we use that $\sum_{e=1}^{r} \sum_{i \in A_e} w_{ji}^2 = \|w_j\|_2^2$.

$$\implies \Pr_z \left[\sum_{e=1}^{r} \left(\sum_{i \in A_e} w_{ji} z_i \right)^2 > 8 \log(nsr) \|w_j\|_2^2 \right] \leq \tfrac{1}{n^2 s^2 r}$$
$$\implies \Pr_z \left[\forall j \in [s],\ \|w'_j\|_2^2 \leq 8 \log(nsr) \|w_j\|_2^2 \right] \geq 1 - \frac{1}{n^2 sr}.$$

Combining with the fact that $\|w'_j\|$ is always at most $W\sqrt{n}$, we get that

$$\mathop{\mathbb{E}}_z \left[\max_{j \in [s]} \|w'_j\|_2^2 \right] \leq O(\log nsr) \|w_j\|_2^2.$$

Combining the above with (17), we get

$$\begin{aligned} \mathsf{NS}_\rho(h) &= \frac{1}{r} \mathop{\mathbb{E}}_{z, \{A_e\}} [\mathsf{AS}(H_z)] \leq \frac{O(W^2 k^4 \log^2(nrs)^2 + k^3 B^2 \sqrt{\log s})}{\sqrt{r}} \\ &= \sqrt{(1-\rho)} O(k^4 W^2 \log^2(ns/(1-\rho)) + k^3 B^2 \sqrt{\log s}). \end{aligned}$$

The claim now follows. ■