

Approximating Probability Distributions by ReLU Networks

Manuj Mukherjee[†]Aslan Tchamkerten[‡]Mansoor Yousefi[‡]

Abstract—How many neurons are needed to approximate a target probability distribution using a neural network with a given input distribution and approximation error? This paper examines this question for the case when the input distribution is uniform, and the target distribution belongs to the class of histogram distributions. We obtain a new upper bound on the number of required neurons, which is strictly better than previously existing upper bounds. The key ingredient in this improvement is an efficient construction of the neural nets representing piecewise linear functions. We also obtain a lower bound on the minimum number of neurons needed to approximate the histogram distributions.

I. INTRODUCTION

Neural networks have long been used as classifiers, whose underlying task is function computation. The theory of function approximation via neural networks is now well-studied [1]–[7]. Recently, through the development of *variational autoencoders* [8] and *generative adversarial networks (GAN)* [9], neural networks are also being used to approximate probability distributions.

In the distribution approximation setting, a neural network is provided with an input random variable X (seed) distributed according to some distribution that can be easily generated, e.g., Gaussian or uniform. The task of the neural network is to compute a function f such that the distribution of $f(X)$ closely approximates a target distribution.

A natural question to ask is the following. Given a seed distribution and a target distribution, what is the minimum number of neurons, i.e., the minimum size of the neural network, needed to approximate the target distribution within a given accuracy? The work of Bailey and Telgarsky [10] tackled this question for a variety of seed-target pairs. In particular, given a network of a fixed depth, [10] provided an upper bound on the minimum network size needed to approximate the target $\text{unif}[0, 1]^h$ using the seed $\text{unif}[0, 1]^l$, where $h > l$, $h, l \in \mathbb{N}$. The authors also showed that this upper bound is tight up to constant factors.

Building upon [10], [11] examined the task of approximating generalisations of the uniform distribution, referred to as *histogram distributions* (see Definition 2). The authors in [11] constructed neural networks which are able to approximate n -tiled histogram distributions using uniform seeds up to a

given accuracy. However, [11] does not comment on whether their construction is optimal with respect to the size of the network.

In this work, we examine the optimality of the construction in [11]. Given uniform seed, approximation error ϵ , and a target histogram distribution P of tile parameter n (see Definition 2), we construct a neural net which approximates P within ϵ . Compared to the construction in [11] (see Table I), our construction uses strictly fewer neurons in various regimes of ϵ and n . For example, in the regime where ϵ is fixed and $n \rightarrow \infty$, we achieve a network size of $O(n^{\frac{3}{2}})$ improving upon $\Theta(n^2)$ in [11]. In other regimes, our construction uses at most the same number of neurons as that of [11]. The key ingredient that we used to reduce the number of neurons is an efficient construction of the neural networks that compute piecewise affine functions studied in [12]. Instead of representing a piecewise affine function by a straightforward neural net of depth 2 and size linear in the number of affine pieces, [12] constructs a deeper neural network computing the same function. Suitably choosing the depth of this neural net, we are able to reduce the size of the network compared to the straightforward network.

To examine the optimality of our construction, we developed a lower bound on the number of neurons needed by neural networks of a fixed depth to approximate histogram distributions. The bound is developed by extending the techniques of [10]. In certain regimes, such as when $\epsilon \rightarrow 0$ and n is fixed, both our construction and that of [11] turn out to be optimal. However, in most other regimes, the lower bound on the number of neurons does not match the upper bound in our construction. The optimal construction remains unknown.

The rest of the paper is organised as follows. The problem of approximating distributions and our main result are formally stated in Section II, along with the comparison of our construction with that of [11]. Section III proves the main result by constructing a neural network of reduced size. Section IV states the lower bound on the minimum size needed to approximate histogram distributions. The paper is concluded in Section V.

II. PRELIMINARIES

A *fully-connected feedforward rectified linear unit (ReLU)* neural network is formally defined as follows.

Definition 1. A map $\phi : \mathbb{R}^{m_0} \rightarrow \mathbb{R}^{m_L}$ is a *fully connected feedforward ReLU network of depth L and size N* , if there exists integers $m_1, m_2, \dots, m_{L-1} > 0$, such that

$$\bullet \quad \phi(x) = M_L(\text{ReLU}(M_{L-1}(\text{ReLU}(\dots \text{ReLU}(M_1(x))), \text{where } M_i : \mathbb{R}^{m_{i-1}} \rightarrow \mathbb{R}^{m_i}, 1 \leq i \leq L, \text{ are affine maps,$$

[†]M. Mukherjee is with The Alexander Kofkin Faculty of Engineering, Bar Ilan University, Ramat Gan, Israel. Email: mukherm@biu.ac.il.

[‡]A. Tchamkerten and M. Yousefi are with Telecom Paris, Institut Polytechnique de Paris, Palaiseau, France. Email: {aslan.tchamkerten, yousefi}@telecom-paris.fr.

and $\text{ReLU}(x) = (\max\{0, x_i\} : 1 \leq i \leq m)$, for any $x \in \mathbb{R}^m$.

- $\sum_{i=1}^L m_i = N$. In other words, the size is simply the total number of neurons.

Notice that widths need not be the same. Throughout this paper, we consider neural networks with input dimension $m_0 = 1$ and output dimension $m_L = 2$,¹ unless otherwise stated. Denote the set of all neural networks with size N and depth L by $\mathcal{N}(N, L)$.

The goal of this work is to approximate n -tiled histogram probability distributions at the output. These are distributions that are obtained by quantizing $[0, 1]^2$ into n^2 uniform squares and assigning probability masses to each of them:

Definition 2. A probability distribution supported on $[0, 1]^2$ is an n -tiled histogram if its pdf P is

$$P(x) = \sum_{\mathbf{k} \in K} w_{\mathbf{k}} \mathbf{1}\{x \in c_{\mathbf{k}}\}, \quad (1)$$

where $K = \{(k_1, k_2) : k_i = 0, \dots, n-1\}$, $c_{\mathbf{k}} = [\frac{k_1}{n}, \frac{k_1+1}{n}] \times [\frac{k_2}{n}, \frac{k_2+1}{n}]$, $w_{\mathbf{k}} > 0$, and $\sum_{\mathbf{k}} w_{\mathbf{k}} = n^2$. The set of all n -tiled histogram distributions supported on $[0, 1]^2$ is denoted by $\mathcal{E}[0, 1]_n^2$.

We shall denote the uniform distribution on $[0, 1]$ by U . Given any random variable $X \sim P(x)$ and any function f , the distribution of $Y = f(X)$ shall be denoted by $f\#P$.

To choose a measure of distance between distributions, we first note that for any neural network $\phi : \mathbb{R} \rightarrow \mathbb{R}^2$ and any $P \in \mathcal{E}[0, 1]_n^2$, conventional measures like *total variation* (TV) or *relative entropy* between $\phi\#U$ and P saturate, i.e., $\text{TV}(P, \phi\#U) = 1$ and $D(P||\phi\#U) = \infty$. This is due to the fact that $\phi(U)$ is always supported on a collection of line segments in \mathbb{R}^2 where P puts no mass. Refer to [13, Section 2] for a detailed discussion. Therefore, following the convention of neural network literature, we choose the *Wasserstein distance*.

Definition 3. For a pair of distributions P and Q supported on \mathbb{R}^2 , denote by $\Pi(P, Q)$ the set of couplings, i.e., joint distributions on $\mathbb{R}^2 \times \mathbb{R}^2$ whose marginals are respectively P and Q . Then, the Wasserstein distance between P and Q is given by

$$W(P, Q) \triangleq \inf_{\pi \in \Pi(P, Q)} \int_{\mathbb{R}^2 \times \mathbb{R}^2} \|x - y\| d\pi(x, y),$$

where $\|\cdot\|$ denotes the ℓ_2 -norm.

We say that a neural network $\phi : \mathbb{R} \rightarrow \mathbb{R}^2$ ϵ -approximates $P \in \mathcal{E}[0, 1]_n^2$ with uniform seed if $W(P, \phi\#U) \leq \epsilon$.

Definition 4. We say that N is an achievable size for ϵ -approximating $\mathcal{E}[0, 1]_n^2$ with uniform seed if, for any $P \in \mathcal{E}[0, 1]_n^2$, there exists an integer $L > 0$ and a neural net $\phi \in \mathcal{N}(N, L)$ which ϵ -approximates P .

¹The key take-away of this work is the reduction in network size through the application of a new construction for representing linear splines (see Lemma 6). We therefore restrict attention to two-dimensional output for simplicity.

Denote by $N^*(n, \epsilon)$ the minimum network size needed for ϵ -approximating $\mathcal{E}[0, 1]_n^2$ with uniform seed.

Our main result, stated next, gives an upper bound on $N^*(n, \epsilon)$.

Theorem 1. Fix $\epsilon > 0$. The minimum number of neurons needed to ϵ -approximate $\mathcal{E}[0, 1]_n^2$ satisfies

$$N^*(n, \epsilon) \leq O(n^{\frac{3}{2}}) + \max\left\{1, \left\lceil \log\left(\frac{2\sqrt{2}}{n\epsilon}\right) \right\rceil\right\} O(n).$$

We prove Theorem 1 in Section III where we construct a neural net of size $N = \Theta(n^{\frac{3}{2}}) + \max\left\{1, \left\lceil \log\left(\frac{2\sqrt{2}}{n\epsilon}\right) \right\rceil\right\} \Theta(n)$ and depth $L = \Theta(n) + n \max\left\{1, \left\lceil \log\left(\frac{2\sqrt{2}}{n\epsilon}\right) \right\rceil\right\}$, which ϵ -approximates distributions in $\mathcal{E}[0, 1]_n^2$.

In Table I we compare our construction to that of [11].² Our construction performs at least as good as the construction of [11] in all regimes of ϵ and n , and sometimes better.

III. THE ACHIEVABILITY PART: PROOF OF THEOREM 1

The proof is based on two steps. Given $P \in \mathcal{E}[0, 1]_n^2$, a piecewise linear function $\phi : \mathbb{R} \rightarrow \mathbb{R}^2$ is constructed, which satisfies $W(P, \phi\#U) \leq \epsilon$. This step is based on [11, Theorem 4.4]. In the second step, we construct a ReLU neural network with desired size N and that realizes ϕ exactly. This construction is based on results from [12] for representing *free-knot linear splines* and sums of functions using neural networks. The improvement over [11] results from this second step.

A. Constructing a piecewise linear function ϕ

We will keep this section short as it covers material from [11]. We shall state only the relevant details that will be required in order to state the function ϕ .

Define the triangular function $g : [0, 1] \rightarrow \mathbb{R}$

$$g(x) = \begin{cases} 2x, & x \in [0, \frac{1}{2}] \\ 2 - 2x, & x \in [\frac{1}{2}, 1] \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

Denote the composition of s triangular functions by $g_s \triangleq \underbrace{g_1 \circ g_1 \circ \dots \circ g_1}_{s \text{ times}}$. The resulting function g_s is the “sawtooth”

$$g_s(x) = \begin{cases} 2^s(x - i2^{-s+1}), & x \in [2i, 2i+1] \times 2^{-s}, \quad \forall i \in I_s, \\ -2^s(x - (i+1)2^{-s+1}), & x \in [2i+1, 2i+2] \times 2^{-s}, \quad \forall i \in I_s, \end{cases} \quad (3)$$

where $I_s = \{0, 1, \dots, 2^{s-1} - 1\}$, and $g_s(x) = 0$ when $x \notin [0, 1]$.

Denote by Σ_m the set of *continuous piecewise affine functions* $h : [0, 1] \rightarrow [0, 1]$ having m breakpoints in $(0, 1)$. Constructing ϕ when $m_L = 1$ is straightforward.

²The result in Theorem 4.4 of [11] is originally stated in terms of depth and a parameter called ‘connectivity’. The bound on the size $N^*(n, \epsilon)$ stated here can be obtained by looking at the construction used in the proof.

Regime	Our construction	Construction in [11]
n fixed	$N = \Theta(\log(\frac{1}{\epsilon}))$	$N = \Theta(\log(\frac{1}{\epsilon}))$
$\epsilon \rightarrow 0$	$L = \Theta(\log(\frac{1}{\epsilon}))$	$L = \Theta(\log(\frac{1}{\epsilon}))$
$n \rightarrow \infty$	$N = \Theta(n^{\frac{3}{2}})$	$N = \Theta(n^2)$
ϵ fixed	$L = \Theta(n)$	$L = \Theta(1)$
$n \rightarrow \infty$	$N = \Theta(n^{\frac{3}{2}})$	$N = \Theta(n^2)$
$\epsilon = \Theta(n^{-\theta}), \theta > 1$	$L = \Theta(n \log n)$	$L = \Theta(\log n)$
$n \rightarrow \infty$	$N = \Theta(n^2)$	$N = \Theta(n^2)$
$\epsilon = \Theta(2^{-n})$	$L = \Theta(n^2)$	$L = \Theta(n)$

TABLE I: Table comparing the sizes and depths of neural networks constructed by us and [11] for different regimes of n and ϵ . Our construction uses deeper nets to cut down on the size of the network.

Lemma 1. For every $P \in \mathcal{E}[0, 1]_n$, there exists an $f \in \Sigma_{n-1}$ satisfying $f \# U = P$.

Proof: The result follows by noting that $F_P^{-1} \in \Sigma_{n-1}$ if $P \in \mathcal{E}[0, 1]_n$, where F_P denotes the cumulative distribution function of P . Since $F_P^{-1} \# U = P$, $f = F_P^{-1}$. ■

To generalize the construction from $m_L = 1$ to $m_L = 2$, consider $p(y_1, y_2) = p(y_1)p(y_2|y_1)$. The component $p(y_1)$ can be constructed from Lemma 1 by marginalization. Let the weights of P for each part $c_{k_1, k_2} = [\frac{k_1}{n}, \frac{k_1+1}{n}] \times [\frac{k_2}{n}, \frac{k_2+1}{n}]$ be denoted by w_{k_1, k_2}^P . Denote by P_{marg} the n -tiled histogram distribution on $[0, 1]$ whose weight are given by $w_k^{P_{\text{marg}}} = \sum_{j=0}^{n-1} w_{k,j}^P$, for all $k \in \{0, 1, \dots, n-1\}$. In other words, P_{marg} is the marginal distribution of P along the first coordinate. Let $\phi_{\text{marg}} \in \Sigma_{n-1}$ be the function given by Lemma 1 which satisfies $\phi_{\text{marg}} \# U = P_{\text{marg}}$. For the component $p(y_2|y_1)$, let P_i , $0 \leq i \leq n-1$, be the n -tiled histogram distributions in $[0, 1]$ with weights given by $w_k^{P_i} = \frac{nw_{i,k}^P}{\sum_{j=0}^{n-1} w_{j,k}^P}$, for all $k \in \{0, 1, \dots, n-1\}$. The distribution P_i are simply the marginals of P along the second coordinate conditioned on the fact that the first coordinate lies in $[\frac{i}{n}, \frac{i+1}{n}]$. Let $\phi_i \in \Sigma_{n-1}$ be the functions given by Lemma 1, which satisfy $\phi_i \# U = P_i$, for all $0 \leq i \leq n-1$.

With these two ingredients, we can now present the function $\phi : [0, 1] \rightarrow [0, 1]^2$ that approximates an n -tiled histogram distribution P .

Lemma 2 (Theorem 4.4 of [11]). Let $P \in \mathcal{E}[0, 1]_n^2$. The corresponding function $\phi : [0, 1] \rightarrow [0, 1]^2$ defined by

$$\phi(x) = (\phi_{\text{marg}}(x), \sum_{i=0}^{n-1} \phi_i(g_s(n\phi_{\text{marg}}(x) - i))), \quad (4)$$

where $g_s(\cdot)$ is defined in (3), satisfies $W(P, \phi \# U) \leq \frac{2\sqrt{2}}{n2^s}$ for any $s \in \mathbb{N}$.

Proof: See the proof of Theorem 4.4 of [11]. ■

Hence, by choosing $s = \max\{1, \lceil \log(\frac{2\sqrt{2}}{n\epsilon}) \rceil\}$, we have $W(\phi \# U, P) \leq \epsilon$ for any $P \in \mathcal{E}[0, 1]_n^2$.

B. Constructing a ReLU neural network

In this section we construct a ReLU neural net that realises the function ϕ given by Lemma 2. The proof is based on

realizing each component in (4) by a ReLU network efficiently. We begin by proving the required lemmas.

Lemma 3 (Pass Lemma). If $f : [0, 1] \rightarrow [0, 1]$ satisfies $f \in \mathcal{N}(N, L)$, then for any $L' \geq L$, $f \in \mathcal{N}(N + L' - L, L')$.

Proof: This is possible by simply adding $L' - L$ layers each of which contain only one neuron, and define the maps M_i , for all $L + 1 \leq i \leq L'$ to be identity maps. Note that the ReLU of the added layers have no effect since the function takes values in $[0, 1]$. ■

Lemma 4 (Parallel Lemma). Let $f, g : [0, 1] \rightarrow [0, 1]$ satisfy $f \in \mathcal{N}(N_1, L_1)$ and $g \in \mathcal{N}(N_2, L_2)$. Then the function $h : [0, 1] \rightarrow [0, 1]^2$ given by $h = (f, g)$ satisfies $h \in \mathcal{N}(N_1 + N_2 + |L_1 - L_2|, \max\{L_1, L_2\})$.

Proof: Without loss of generality, assume $L_1 \geq L_2$. Then, by the Lemma 3, we have $g \in \mathcal{N}(N_2 + |L_1 - L_2|, L_1)$. To get the network h , we simply stack the net $f \in \mathcal{N}(N_1, L_1)$ on top of net $g \in \mathcal{N}(N_2 + |L_1 - L_2|, L_1)$, and ensure that the weights of the connections between neurons in f to neurons in g is zero. Formally, let the affine maps for f and g be denoted by $M_i^{(f)}$ and $M_i^{(g)}$, for all $1 \leq i \leq L_1$. Then, define the new neural net $h : [0, 1] \rightarrow [0, 1]^2$ by using the concatenated affine maps $M_i = (M_i^{(f)}, M_i^{(g)})$, for all $1 \leq i \leq L_1$. It is easy to see that $h = (f, g)$ and $h \in \mathcal{N}(N_1 + N_2 + |L_1 - L_2|, \max\{L_1, L_2\})$. ■

Lemma 5 (Compose Lemma). Let $f, g : \mathbb{R} \rightarrow \mathbb{R}$. If $f \in \mathcal{N}(N_1, L_1)$ and $g \in \mathcal{N}(N_2, L_2)$, then for any $p, q \in \mathbb{R}$, $f \circ (pg + q) \in \mathcal{N}(N_1 + N_2 - 1, L_1 + L_2 - 1)$.

Proof: We shall use superscripts (f) and (g) to denote affine maps from the neural nets realising f and g . The network $f \circ (pg + q)$ is simply generated by adding the net f in series with net g , deleting the output node of g , and connecting the neurons from layer $L_2 - 1$ of g to the first layer of f using suitable weights.

Formally, let the affine maps $M_{L_2}^{(g)}$ and $M_1^{(f)}$ be expressed as $M_{L_2}^{(g)}(x) = A^{(g)}x + b^{(g)}$ and $M_1^{(f)}(x) = A^{(f)}x + b^{(f)}$, where $A^{(g)} \in \mathbb{R}^{1 \times m_{L_2-1}^{(g)}}$, $A^{(f)} \in \mathbb{R}^{m_1^{(f)} \times 1}$, $b^{(g)} \in \mathbb{R}$, $b^{(f)} \in \mathbb{R}^{m_1^{(f)}}$. Now, define the new affine map $M : \mathbb{R}^{m_{L_2-1}^{(g)}} \rightarrow \mathbb{R}^{m_1^{(f)}}$ as $M(x) = Ax + b$, where $A \in \mathbb{R}^{m_1^{(f)} \times m_{L_2-1}^{(g)}}$, $b \in \mathbb{R}^{m_1^{(f)}}$, and

$A = pA^{(f)}A^{(g)}$, $b = (pb^{(g)} + q)A^{(f)} + b^{(f)}$. Now, consider the net of depth $L_1 + L_2 - 1$ defined by the affine maps $M_i = M_i^{(g)}$, for all $1 \leq i \leq L_2 - 1$, $M_{L_2} = M$, and $M_{L_2+i} = M_{1+i}^{(f)}$, for all $1 \leq i \leq L_1 - 1$. It is easy to see that this net has size $N_1 + N_2 - 1$ and realises the map $f \circ (pg + q)$. ■

Next, we state a couple of important technical result from [12] on representing functions from Σ_m and sums of functions.³

Lemma 6 (Spline Lemma, Theorem 3.1 of [12]). *Let $f \in \Sigma_m$. Then f can be realised by a ReLU network with $L = \max\left\{2, 2\left\lceil \frac{m}{(W-2)\lfloor \frac{W-2}{6} \rfloor} \right\rceil + 1\right\}$ and $N = W(L-1) + 1$, for any $W \geq 8$.*

Proof: See Theorem 3.1 of [12]. ■

Note that $f \in \Sigma_m$ can always be realised using a depth-2 ReLU network of size $m + 3$ in a straightforward manner. The spline lemma, however, uses a deeper net and realizes $f \in \Sigma_m$ more efficiently, thereby saving up on the number of neurons. Below we explain how the Spline Lemma does this.

The Spline Lemma is proved by viewing the set of piecewise affine functions with m breakpoints as a vector space of dimension m , and constructing a suitable basis for this space. This vector space representation is more efficient than direct computation of the sum (4). As a result, any piecewise affine function h of $m_W \triangleq (W-2)\lfloor \frac{W-2}{6} \rfloor$ breakpoints, where $W \geq 8$ is a parameter, can be expressed as a sum $h(x) = \sum_{k=1}^{W-2} T_k$, where $T_k = \sum_{j=1}^{W-2} c_{j,k} \text{ReLU}(x - \beta_j)$, where $c_{j,k}, \beta_j \in \mathbb{R}$. The function $h = \sum_{k=1}^{W-2} T_k$ can be realised by a network of three layers with the two intermediate layers having $W-2$ nodes each, and given that the second intermediate layer is activation free. A simple argument (see [12, Remark 3.1]) shows that this artificial technical requirement can be removed by simply adding two more neurons to both the intermediate layers, and adjusting some weights suitably. Now, any $f \in \Sigma_m$ can be expressed as $f = \sum_{i=1}^{\lceil \frac{m}{m_W} \rceil} h_i$, where h_i are piecewise affine functions with m_W breakpoints, and can be realised using the neural networks of size $2(W-2) + 1$, depth 3, and activation free second layer as described above. The result in the Spline Lemma then follows by realising the sum in f using a network constructed according to the following Add Lemma (Lemma 7). The proof of the Spline Lemma includes the technical argument to remove the artificial restriction of the second layer of each h_i being activation free.

Lemma 7 (Add Lemma, Proposition 4.2(ii) of [12]). *Consider ℓ functions $f_i : [0, 1] \rightarrow \mathbb{R}$, $1 \leq i \leq \ell$, with $f_i \in (N_i, L_i)$. Then the sum $f = \sum_{i=1}^{\ell} f_i$ satisfies $f \in \mathcal{N}(\sum_{i=1}^{\ell} (N_i + 2L_i - 2) - \ell + 1, \sum_{i=1}^{\ell} L_i - \ell + 1)$.*

Proof: The proof involves putting the networks in series and adding a pair of nodes to each intermediate layer, one

³The results in [12] are stated in terms of width of network given by $W = \max_{1 \leq i \leq L_1} m_i$. Moreover, by the convention of [12], the depth is $L-1$ instead of L . We have therefore suitably modified the statements according to our convention, and also expressed the statements in terms of size and not width.

to pass the input, and the other to pass the partial sum. See Proposition 4.2(ii) and Remark 3.1 of [12] for details. ■

With these lemmas in hand, we begin the construction of a net for ϕ . Firstly, recall that $\phi_{\text{marg}}, \phi_i \in \Sigma_{n-1}$, for all $0 \leq i \leq n-1$. Hence, choosing $W = \lceil \sqrt{n} \rceil$ in the Spline Lemma (Lemma 6), we have for all sufficiently large n that $\phi_{\text{marg}}, \phi_i \in \mathcal{N}(\Theta(\sqrt{n}), \Theta(1))$. Next, observe that $g_1 \in \mathcal{N}(4, 2)$. This follows from the fact that $g_1(x) = M_2(\text{ReLU}(M_1(x)))$, where $M_1(x) = [1 \ 1 \ 1]^T x + [0 \ -\frac{1}{2} \ -1]^T$, and $M_2(x) = [2 \ -4 \ 2]^T x$. Hence, by recalling that $g_s = \underbrace{g_1 \circ g_1 \circ \dots \circ g_1}_{s \text{ times}}$, the repeated application of Compose

Lemma (Lemma 5) yields $g_s \in \mathcal{N}(3s+1, s+1)$. Therefore, an application of Compose Lemma (Lemma 5) yields that for any $0 \leq i \leq n-1$, $\phi_i(g_s(n\phi_{\text{marg}}(\cdot) - i)) \in \mathcal{N}(3s + \Theta(\sqrt{n}), s + \Theta(1))$. Finally, by applying the Add Lemma (Lemma 7), we get $\sum_{i=0}^{n-1} \phi_i(g_s(n\phi_{\text{marg}}(\cdot) - i)) \in \mathcal{N}(\Theta(n^{\frac{3}{2}}) + s\Theta(n), ns + \Theta(n))$. Now, recalling that $\phi_{\text{marg}} \in \mathcal{N}(\Theta(\sqrt{n}), \Theta(1))$, a use of the Parallel Lemma (Lemma 4) gives $\phi \in \mathcal{N}(\Theta(n^{\frac{3}{2}}) + s\Theta(n), ns + \Theta(n))$. By Lemma 2, since a choice of $s = \max\{1, \lceil \log(\frac{2\sqrt{2}}{n\epsilon}) \rceil\}$ yields $W(\phi \# U, P) \leq \epsilon$, we have $N^*(n, \epsilon) \leq O(n^{\frac{3}{2}}) + \max\{1, \lceil \log(\frac{2\sqrt{2}}{n\epsilon}) \rceil\} O(n)$, which completes the proof of Theorem 1.

IV. THE CONVERSE PART: LOWER BOUND ON THE NUMBER OF NEURONS

It is difficult to directly obtain lower bounds $N^*(n, \epsilon)$. Instead we focus on a related quantity, the minimum number of neurons required by a neural network of a fixed depth L to ϵ -approximate histogram distributions. For this section, we shall be considering $\mathcal{E}[0, 1]_n^d$ for any general $d \geq 2$, and all definitions from Section II are modified accordingly. For instance, we now consider neural nets whose output dimension is d , i.e., $\phi : \mathbb{R} \rightarrow \mathbb{R}^d$.

Formally, define $N^*(n, \epsilon, L)$ to be the minimum achievable size for a network of depth L to ϵ -approximate $\mathcal{E}[0, 1]_n^d$. The following theorem states the lower bound $N^*(n, \epsilon, L)$.

Theorem 2. *Fix $n, \epsilon > 0$, $L \geq 1$. The minimum achievable size for a network of depth L to ϵ -approximate $\mathcal{E}[0, 1]_n^d$ satisfies*

$$N^*(n, \epsilon, L) \geq \max\left\{L, L\left[\frac{1}{e}\left(\frac{C(d)}{n\epsilon}\right)^{\frac{d-1}{L}} - 1\right]\right\},$$

$$\text{where } C(d) \triangleq \frac{d}{d-1} \left(\frac{2\Gamma(\frac{d+1}{2})\Gamma(\frac{3}{2})}{\pi^{\frac{d}{2}}\sqrt{d}} \right)^{\frac{1}{d-1}}.$$

Based on Theorem 2, and setting $d = 2$, Table II marks whether our neural network in Section III-B and the network constructed in [11] are optimal for various regimes of ϵ and n . Here, optimality implies whether the neural net ϵ -approximating $\mathcal{E}[0, 1]_n^2$ using some depth L has its size N to be equal in orders of magnitude to the bound on $N^*(n, \epsilon, L)$ provided in Theorem 2. For example, we see that in the regime where $\epsilon \rightarrow 0$ and n is fixed, both our construction and that of [11] are optimal. However, in most regimes we are unable to guarantee optimality. We believe the problem stems from the weakness

Regime	Optimality	
	Our construction	Construction in [11]
$\epsilon \rightarrow 0$ n fixed	Yes	Yes
$n \rightarrow \infty$ ϵ fixed	Not known	Not known
$n \rightarrow \infty$ $\epsilon = \Theta(n^{-\theta}), \theta > 1$	Not known	Not known
$n \rightarrow \infty$ $\epsilon = \Theta(2^{-n})$	Yes	Not known

TABLE II: Table documenting the optimality of different constructions for different regimes of ϵ and N . ‘Not known’ means that the construction does not meet the bound in Theorem 2.

in lower bound. For example, consider the regime where ϵ is fixed and $n \rightarrow \infty$. Using a depth $L = \Theta(1)$, the bound in Theorem 2 reduces $N^*(n, \epsilon, L) \geq \Theta(1)$. This is clearly loose, since a constant sized network should not be able to ϵ -approximate $\mathcal{E}[0, 1]_n^2$ as n grows.

We only provide a very brief outline of the proof of Theorem 2 due to the lack of space. The details can be found in [14]. The proof of Theorem 2 is an extension of a method used in [10]. Note that the image of any function $f : \mathbb{R} \rightarrow \mathbb{R}^d$ realized by a ReLU net is supported on at most ζ 1-dimensional affine hyperplanes in \mathbb{R}^d . The following lemma from [10] bounds the number ζ based on the size N and depth L of the network.

Lemma 8 (Lemma 2.1 of [10]). *Let $f : \mathbb{R} \rightarrow \mathbb{R}^d$ is realized by a ReLU net of size N and depth L . Then, the image of f is supported on at most ζ 1-dimensional affine hyperplanes on \mathbb{R}^d , where*

$$\zeta \leq (e(\frac{N}{L} + 1))^L.$$

Proof: See Lemma 2.1 of [10]. ■

The following theorem then bounds the distance between n -tiled histogram distribution from $\mathcal{E}[0, 1]_n^d$ and any distribution supported on ζ 1-dimensional affine hyperplanes in \mathbb{R}^d .

Theorem 3. *Let $P \in \mathcal{E}[0, 1]_n^d$ and Q be a distribution supported on any ζ 1-dimensional affine hyperplanes in \mathbb{R}^d . Then*

$$W(P, Q) \geq C(d)\zeta^{-\frac{1}{d-1}}n^{-1},$$

$$\text{where } C(d) \triangleq \frac{d}{d-1} \left(\frac{2\Gamma(\frac{d+1}{2})\Gamma(\frac{3}{2})}{\pi^{\frac{d}{2}}\sqrt{d}} \right)^{\frac{1}{d-1}}.$$

Proof: See the long version of this document [14]. ■

It is easy to see that Theorem 2 follows by simply plugging in the bound on ζ from Lemma 8 to the expression in Theorem 3, and noting the trivial bound $N \geq L$.

V. CONCLUDING REMARKS

In this work, we studied the problem of approximating n -tiled histogram distributions by ReLU neural networks with uniform seed. We obtained a new upper bound on the achievable size of the network, which is tighter than existing results. The main

ingredient we leveraged to tighten the bound is an efficient construction of neural networks representing free knot linear splines studied in [12]. We also computed a lower bound on achievable size, but unfortunately the bounds do not match. We believe that the lower bound is loose and tightening it shall be a future direction of research.

REFERENCES

- [1] A. R. Barron, “Approximation and estimation bounds for artificial neural networks,” *Mach. Learn.*, vol. 14, no. 1, p. 115–133, Jan. 1994.
- [2] M. Telgarsky, “Benefits of depth in neural networks,” ser. Proceedings of Machine Learning Research, V. Feldman, A. Rakhlin, and O. Shamir, Eds., vol. 49. PMLR, 23–26 Jun 2016, pp. 1517–1539.
- [3] D. Yarotsky, “Error bounds for approximations with deep relu networks,” *Neural Networks*, vol. 94, pp. 103 – 114, 2017.
- [4] M. Mehrabi, A. Tchamkerten, and M. Yousefi, “Bounds on the approximation power of feedforward neural networks,” ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 3453–3461.
- [5] R. Arora, A. Basu, P. Mianjy, and A. Mukherjee, “Understanding deep neural networks with rectified linear units,” in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018*.
- [6] P. Grohs, D. Perekrestenko, D. Elbrächter, and H. Bölcskei, “Deep neural network approximation theory,” *CoRR*, vol. abs/1901.02220, 2019. [Online]. Available: <http://arxiv.org/abs/1901.02220>
- [7] H. Lee, R. Ge, T. Ma, A. Risteski, and S. Arora, “On the ability of neural nets to express distributions,” ser. Proceedings of Machine Learning Research, S. Kale and O. Shamir, Eds., vol. 65. PMLR, 07–10 Jul 2017, pp. 1271–1296.
- [8] D. P. Kingma and M. Welling, “Auto-encoding variational bayes,” in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014*.
- [9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in Neural Information Processing Systems 27*, 2014, pp. 2672–2680.
- [10] B. Bailey and M. Telgarsky, “Size-noise tradeoffs in generative networks,” in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018, p. 6490–6500.
- [11] D. Perekrestenko, S. Müller, and H. Bölcskei, “Constructive universal high-dimensional distribution generation through deep relu networks,” in *Int. Conf. on Machine Learning (ICML)*, 2020.
- [12] I. Daubechies, R. A. DeVore, S. Foucart, B. Hanin, and G. Petrova, “Nonlinear approximation and (deep) relu networks,” *CoRR*, vol. abs/1905.02199, 2019.
- [13] M. Arjovsky and L. Bottou, “Towards principled methods for training generative adversarial networks,” in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017*.
- [14] M. Mukherjee, A. Tchamkerten, and M. Yousefi, “Approximating probability distributions by relu networks,” *arXiv:2101.09973*.