2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)

RadiX-Net: Structured Sparse Matrices for Deep Neural Networks

Ryan A. Robinett[1] and Jeremy Kepner[1,2]

[1]MIT Department of Mathematics, [2]MIT Lincoln Laboratory Supercomputing Center

***Abstract*—The sizes of deep neural networks (DNNs) are rapidly outgrowing the capacity of hardware to store and train them. Research over the past few decades has explored the prospect of sparsifying DNNs before, during, and after training by pruning edges from the underlying topology. The resulting neural network is known as a sparse neural network. More recent work has demonstrated the remarkable result that certain sparse DNNs can train to the same precision as dense DNNs at lower runtime and storage cost. An intriguing class of these sparse DNNs is the X-Nets, which are initialized and trained upon a sparse topology with neither reference to a parent dense DNN nor subsequent pruning. We present an algorithm that deterministically generates RadiX-Nets: sparse DNN topologies that, as a whole, are much more diverse than X-Net topologies, while preserving X-Nets' desired characteristics. We further present a functional-analytic conjecture based on the longstanding observation that sparse neural network topologies can attain the same expressive power as dense counterparts.**

***Index Terms*—sparse neural networks, sparse matrices, artificial intelligence**

I. Introduction

As research in artificial neural networks progresses, the sizes of state-of-the-art deep neural network (DNN) architectures put increasing strain on the hardware needed to implement them [1], [2]. In the interest of reduced storage and runtime costs, much research over the past decade has focused on the sparsification of artificial neural networks [3]–[13]. In the listed resources alone, the methodology of sparsification includes Hessian-based pruning [3], [4], Hebbian pruning [5], matrix decomposition [9], and graph techniques [10]–[13]. Yet all of these implementations are alike in that a DNN is initialized and trained, and then edges deemed unnecessary by certain criteria are pruned.

Unlike most strategies for creating sparse DNNs, the X-Net strategy presented in [14] is sparse "*de novo*"—that is, X-Nets are neural networks initialized upon sparse topologies. X-Nets are observed to train as well on various data sets as their dense counterparts, while exhibiting reduced memory usage [14], [15]. Further, by offering sparse alternatives to fully-connected and convolutional layers—X-Linear and X-Conv layers, respectively—X-Nets exhibit such performance on not only generalized DNN tasks, but also image recognition tasks canonically reserved for convolutional neural networks [9].

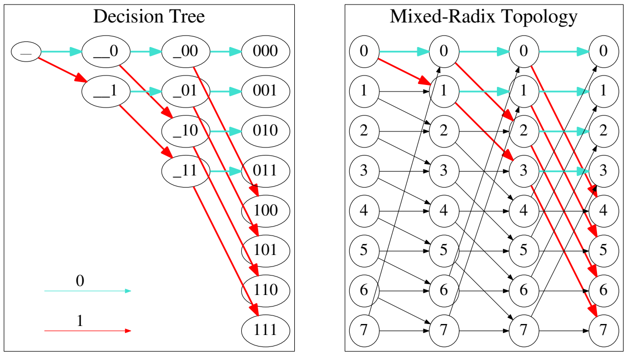


Fig. 1. Construction of the mixed-radix topology defined by $\mathcal{N} = (2, 2, 2)$ using overlapping decision trees. (left) A four-layer binary decision tree. (right) A four-layer mixed radix topology composed of eight offset decision trees.

X-Net layers are constructed using properties of expander graphs, which give X-Nets the properties of sparsity and path-connectedness (see Mathematical Preliminaries) [14], [16]. Random X-Linear layers achieve path-connectedness probabilistically, while explicit X-Linear layers, constructed from Cayley graphs, aim to achieve path-connectedness deterministically [14]. As an artifact of their construction from Cayley graphs, explicit X-Linear layers are required have the same number of nodes as adjacent layers. This constrains the kinds of X-Net topologies which may be constructed deterministically.

We propose RadiX-Nets as a new family of *de novo* sparse DNNs that deterministically achieve path-connectedness while allowing for diverse layer architectures. Instead of emulating Cayley graphs, RadiX-Nets achieve sparsity using properties of mixed-radix numeral systems, while allowing for diversity in network topology through the Kronecker product [17]. Additionally, RadiX-Nets satisfy symmetry, a property which both guarantees path connectedness and precludes inherent training bias in the underlying sparse DNN architecture.

II. Mathematical Preliminaries

Understanding RadiX-Nets' graph-theoretic construction and underlying mathematical properties requires defining a few concepts. RadiX-Nets are composed of sub-nets that are herein referred to as mixed-radix topologies. Mixed-radix

This material is based in part upon work supported by the NSF under grant number DMS-1312831. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

978-1-5386-5555-9/19/$31.00 ©2019 IEEE
DOI 10.1109/IPDPSW.2019.00051

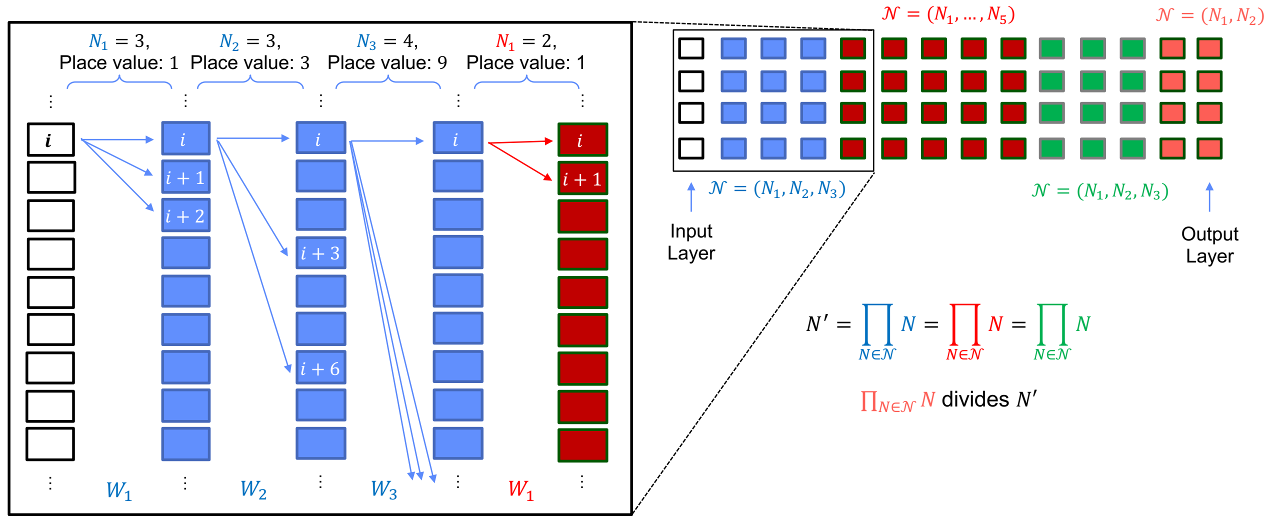

N_1 = 3, Place value: 1
N_2 = 3, Place value: 3
N_3 = 4, Place value: 9
N_1 = 2, Place value: 1
i
i
i + 1
i + 2
i
i + 3
i + 6
i
i
i + 1
W_1
W_2
W_3
W_1
$\mathcal{N}$ = (N_1, …, N_5)
$\mathcal{N}$ = (N_1, N_2)
$\mathcal{N}$ = (N_1, N_2, N_3)
Input Layer
$\mathcal{N}$ = (N_1, N_2, N_3)
Output Layer
$N' = \prod_{N\in\mathcal{N}} N = \prod_{N\in\mathcal{N}} N = \prod_{N\in\mathcal{N}} N$
$\prod_{N\in\mathcal{N}} N$ divides N'

Fig. 2. A RadiX-Net *prior* to Kronecker product is a layered graph wherein each layer is a mixed-radix topology. (left) A single mixed-radix topology within a concatenation of mixed-radix topologies, defined by mixed-radix system $\mathcal{N} = (3, 3, 4)$. (top right) A concatenation of the mixed-radix topologies defined by $\mathcal{N}, \mathcal{N}, \mathcal{N}$, and $\mathcal{N}$. The mixed-radix topologies are concatenated such that the output nodes of one are identified label-wise with the input-nodes of the next. (bottom right) Strict relationships between $\mathcal{N}, \mathcal{N}, \mathcal{N}$, and $\mathcal{N}$ allow for RadiX-Nets to satisfy sparsity, symmetry, and path-connectedness.

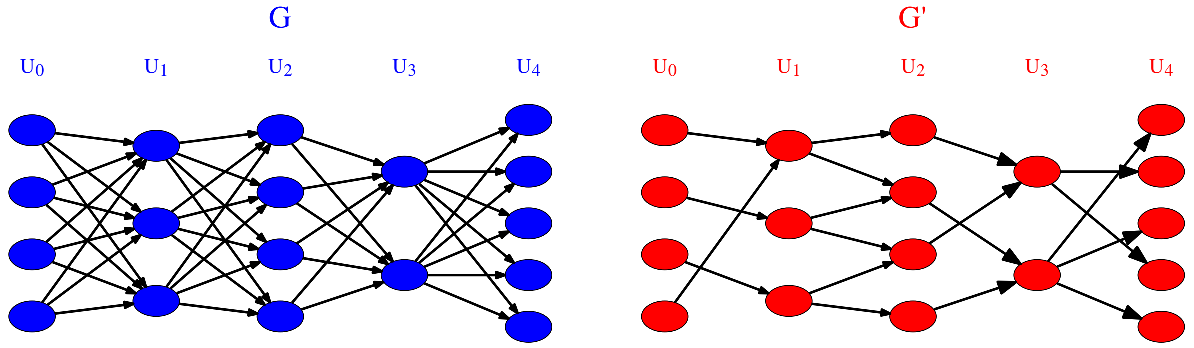
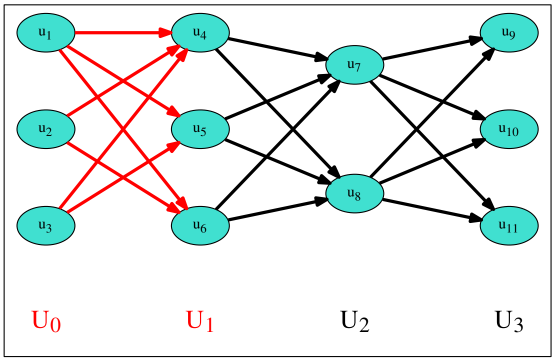


Fig. 3. Feedforward neural network topologies (FNNTs) G, G' built on the same ordered collection of nodes $\mathcal{U}$. For every ordered collection of nodes, there exists a unique fully-connected FNNT; for $\mathcal{U}$ in this example, this happens to be G.

topologies are based on properties of mixed-radix number systems, and can be constructed from overlapping decision trees (see Figure 1). A mixed-radix numeral system is the sole parameter used to uniquely specify a mixed-radix topology. Mixed-radix topologies are a kind of feedforward neural net topology (FNNT), which is a layered graph wherein all vertices in one layer point only to some number of vertices in the next. The adjacency matrix of an FNNT is uniquely defined by the adjacency submatrices corresponding to each of its layers. Essentially, RadiX-Net topologies are constructed from Kronecker products of mixed-radix adjacency submatrices and dense DNN adjacency submatrices (see Figure 5). The main properties of interest in RadiX-Nets are path-connectedness—which ensures each output depends upon all inputs—and symmetry, which ensures that there is the same number of paths between each input and output.

Mixed-Radix Numeral System: Let $\mathcal{N} = (N_1, \ldots, N_L)$ be an ordered set of L integers greater than 1. Let $N' = \prod_{i=1}^{L} N_i$. All such $\mathcal{N}$ implicitly define a numeral system which bijectively represents all integers in $\{0, \ldots, N'-1\}$. That is, the set of ordered sets

$$\{(n_1, \ldots, n_L) \mid n_i \in \{0, \ldots, N_i - 1\}\}$$



$$A_1 = \left(\begin{array}{c|c} \mathbf{0}_{3,3} & \mathbf{W} \\ \hline \mathbf{0}_{3,3} & \mathbf{0}_{3,3} \end{array}\right) = \left(\begin{array}{ccc|ccc} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right)$$

$$A = \left(\begin{array}{c|ccc} & \mathbf{W} & & \\ \mathbf{0}_{8,3} & & \mathbf{1}_{3,2} & \\ & & & \mathbf{1}_{2,3} \\ \hline \mathbf{0}_{3,3} & & \mathbf{0}_{3,8} & \end{array}\right)$$

Fig. 4. An FNNT G, together with the matrices A_1 and A. A_1 is the adjacency matrix of the restriction G_1 of G to the the nodes $U_0 \cup U_1$. By definition, then, $\mathbf{W}$ is the adjacency submatrix of G_1. For both A_1 and A, the entry at i, j is nonzero if and only if there exists a node from u_i to u_j.

maps bijectively to $\{0, \ldots, N' - 1\}$ by the map

$$(n_1, \ldots, n_L) \longleftrightarrow \sum_{i=1}^{L} \left(n_i \prod_{j=1}^{i-1} N_j \right).$$

Mixed-radix numeral systems arise naturally in numerous graph-theoretic constructions, such as decision trees (see Figure 1).

Feedforward Neural Net Topology (FNNT): An FNNT G with $n + 1$ layers of nodes—including input and output layers—is an $(n + 1)$-partite directed graph with independent components $U_0, \ldots, U_n$ satisfying the constraints that

- if there exists an edge from $u \in U_i$ to $v \in U_j$, then $j = i + 1$, and
- the out-degree of $u \in U_i$ is nonzero for all $i < n$.

Adjacency Submatrix of an FNNT: Say G is an FNNT. Let G_i be the restriction of G to the set of nodes $U_{i-1} \cup U_i$ and the set of edges from U_{i-1} to U_i in G. We define $m_i = |U_{i-1}|$ and $n_i = |U_i|$ for all i. Up to a permutation of indices, the adjacency matrix of G_i is of the form

$$\left(\begin{array}{c|c} \mathbf{0}_{m_i,m_i} & \mathbf{W}_i \\ \hline \mathbf{0}_{n_i,m_i} & \mathbf{0}_{n_i,n_i} \end{array}\right)$$

for some $\mathbf{W}_i$, where $\mathbf{0}_{a,b}$ is the $a \times b$ matrix of zeros. We refer to $\mathbf{W}_i$ as the adjacency submatrix of the restriction G_i.

Conversely, say that an ordered set $\mathcal{W} = (\mathbf{W}_1, \ldots, \mathbf{W}_n)$ of matrices is such that

- the only nonzero entries of $\mathbf{W}_i$ are ones for all i, and
- no column of $\mathbf{W}_i$ is the zero vector.

If the number of columns in $\mathbf{W}_{i-1}$ equals the number of rows in $\mathbf{W}_i$ for all $i \in \{1, \ldots, n\}$, then $\mathcal{W}$ defines a unique FNNT with $n + 1$ layers of nodes.

Path-Connectedness: We define path-connectedness as follows: let G be an FNNT with $n + 1$ layers of nodes. G is path-connected if, for every $u \in U_0$ and every $v \in U_n$, there exists a path from u to v.

Symmetry: We define symmetry as follows: let G be an FNNT with $n{+}1$ layers of nodes. G is symmetric if there exists a positive integer m such that, for all $u \in U_0$ and all $v \in U_n$, there exist exactly m paths from u to v. If G is symmetric, it is path-connected. If G has adjacency matrix A, then G satisfies symmetry if and only if, up to some permutation of A,

$$A^n = \left(\begin{array}{c|c} \mathbf{0}_{n,M-n} & m\mathbf{1}_{n,n} \\ \hline \mathbf{0}_{M-n,M-n} & \mathbf{0}_{M-n,n} \end{array}\right),$$

where M is the number of nodes in G, $\mathbf{1}_{a,b}$ is the $a \times b$ matrix of ones, and m is some positive integer.

Density of an FNNT An ordered collection $(U_0, \ldots, U_n)$ of sets of nodes implicitly defines a unique, fully-connected DNN topology—namely, the FNNT such that, for all $i \in \{1, \ldots, n\}$, there exists an edge from u to v for all $u \in U_{i-1}$ and all $v \in U_i$. The number of edges in this DNN topology is equal to $\sum_{i=1}^{n} |U_{i-1}||U_i|$. We define the density of an FNNT G as the ratio of the number of edges in G to the number of edges in the DNN topology defined by the ordered set of independent components of G. By this construction, the highest possible density of an FNNT is one, while the lowest is $\frac{\sum_{i=1}^{n} |U_{i-1}|}{\sum_{i=1}^{n} |U_{i-1}||U_i|}$.

III. RadiX-Net Topologies

A. Constructing RadiX-Net Topologies

We construct RadiX-Net topologies using mixed-radix topologies as building blocks, as motivated by Figure 2.

Mixed-Radix Topologies: Let L be a positive integer, and let $\mathcal{N} = (N_1, \ldots, N_L)$, where N_i is an integer greater than 1 for all i. Let $N' = \prod_{N \in \mathcal{N}} N$, and let U_i be a set of N' nodes—with labels $0, \ldots, N' - 1$—for all $i \in \{0, \ldots, L\}$. For all $i \in \{1, \ldots, L\}$, we create edges from node j in U_{i-1} to node $j + n \prod_{j=1}^{i-1} N_j \pmod{N'}$ in U_i for all $n \in \{0, \ldots, N_i - 1\}$.

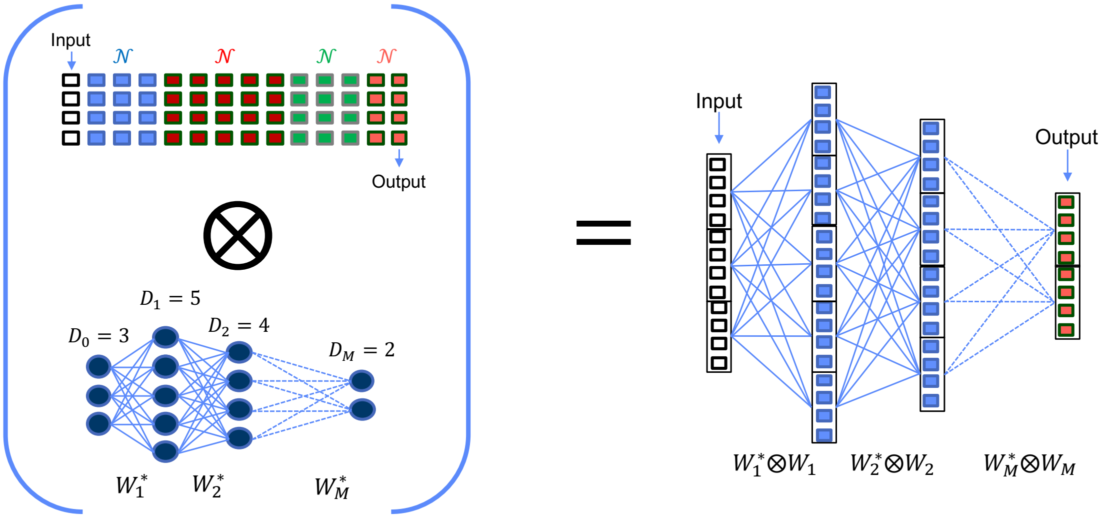


Fig. 5. The final step of RadiX-Net construction involves Kronecker products of adjacency submatrices of mixed-radix topologies and adjacency submatrices of an arbitrary dense deep neural network with the same number of layers. The number of vertices in each layer of the dense deep neural networks provides an additional set of parameters by which a wide range of RadiX-Nets can be defined.

```
Algorithm  Generating RadiX-Net Topologies
  // returns $\overline{\mathcal{W}}$ generating unique FNNT (see Approach)
  $\mathcal{W} \leftarrow$ empty_array()
  $N' \leftarrow \prod_{N\in\mathcal{N}_1} N$
  // see (2):
  $\mathbf{P} \leftarrow$ permutation_matrix($N'$)
  for $\mathcal{N} \in \mathcal{N}^*$ do
      pv $\leftarrow$ 1
      for $N \in \mathcal{N}$ do
          $\mathbf{W} \leftarrow \mathbf{0}_{N',N'}$
          for $j = 0$ to $N - 1$ do
              // see (1):
              $\mathbf{W} \leftarrow \mathbf{W} + \mathbf{P}^{j*\mathrm{pv}}$
          $\mathcal{W}$.append($\mathbf{W}$)
          pv $\leftarrow$ pv*$N$
  $\overline{\mathcal{W}} \leftarrow$ empty_array()
  // see (3):
  for $\mathbf{W}_i \in \mathcal{W}$ do
      $\mathbf{W}^* \leftarrow \mathbf{1}_{D_{i-1},D_i}$
      $\overline{\mathcal{W}}$.append($\mathbf{W}^* \otimes \mathbf{W}_i$)
  return $\overline{\mathcal{W}}$
```

Fig. 6. An algorithm for generating the RadiX-Net topology defined by list $\mathcal{N}^* = (\mathcal{N}_1, \ldots, \mathcal{N}_M)$ of mixed-radix numeral systems and list $\mathcal{D} = (D_0, \ldots, D_{\overline{M}})$ of positive integers.

Let $\mathbf{W}_i$ be the adjacency submatrix defining the edges from U_{i-1} to U_i. By construction, we have that

$$\mathbf{W}_i = \sum_{j=0}^{N_i-1} \mathbf{P}^{j\nu_i}, \tag{1}$$

where $\nu_i = \prod_{k=1}^{i-1} N_k$ and $\mathbf{P}$ is the permutation matrix

$$\left(\begin{array}{ccc|c} 0 & \ldots & 0 & 1 \\ \hline & & & 0 \\ & \mathbf{I}_{N'-1} & & \vdots \\ & & & 0 \end{array}\right), \tag{2}$$

$\mathbf{I}_n$ being the $n \times n$ identity matrix. We refer to the resulting graph as the mixed-radix topology induced by $\mathcal{N}$.

RadiX-Net Topologies: Here, we formally construct RadiX-Net topologies using mixed-radix topologies, adjacency submatrices, and the Kronecker product, as motivated by Figure 5. For an informal programmatic construction, see Figure 6.

RadiX-Net topologies are uniquely defined by an ordered set $\mathcal{N}^* = (\mathcal{N}_1, \ldots, \mathcal{N}_M)$ of mixed-radix numeral systems $\mathcal{N}_i = (N_1^i, \ldots, N_{L_i}^i)$ together with an ordered set $\mathcal{D}$ of positive integers. We require that

1) there exists a positive integer N' such that $N' = \prod_{N\in\mathcal{N}_i} N$ for all $i \in \{1, \ldots, M-1\}$, and
2) $\prod_{N\in\mathcal{N}_M} N$ divides N'.

Let $\overline{M} = \sum_{i=1}^{M} L_i$, the total number of radices in $\mathcal{N}^*$; we further require that $\mathcal{D} = (D_0, \ldots, D_{\overline{M}})$ consist of $\overline{M} + 1$ integers satisfying $D_i \ll N'$ for all i.

We construct a RadiX-Net G using $\mathcal{N}^*$ and $\mathcal{D}$ as follows: let G_i be the mixed-radix topology induced by $\mathcal{N}_i$. Identifying the output nodes of G_i with the input nodes of G_{i+1} creates an $\overline{M}$-layer FNNT with ordered set $\mathcal{W} = (\mathbf{W}_1, \ldots, \mathbf{W}_{\overline{M}})$ of adjacency submatrices of the form (1)[†]. Similarly, $\mathcal{D}$ implicitly defines a unique dense DNN topology H on an ordered collection $U_0, \ldots, U_{\overline{M}}$ of nodes satisfying $|U_i| = D_i$. The ordered set of adjacency matrices of H is $\mathcal{W}^* = (\mathbf{W}_1^*, \ldots, \mathbf{W}_{\overline{M}}^*)$, where $\mathbf{W}_i^*$ is the $D_{i-1} \times D_i$ matrix of ones. We define G as the unique FNNT defined by

$$\overline{\mathcal{W}} = (\mathbf{W}_1^* \otimes \mathbf{W}_1, \ldots, \mathbf{W}_{\overline{M}}^* \otimes \mathbf{W}_{\overline{M}}) \tag{3}$$

(see Mathematical Preliminaries).

Mixed-radix and RadiX-Net topologies satisfy symmetry, and therefore path-connectedness. Proofs for this assertion, as

[†] We refer to such an FNNT as an *extended mixed-radix topology* (see Appendix).

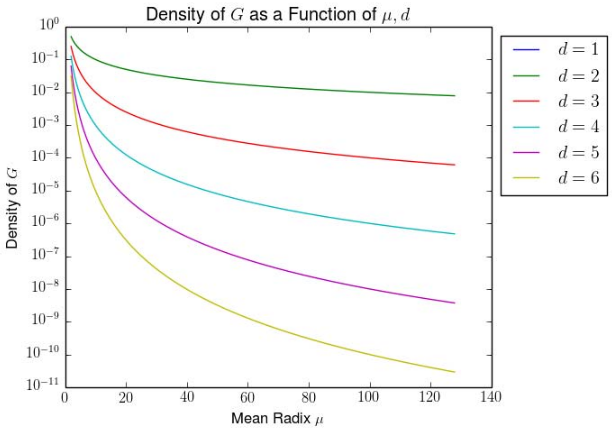


Fig. 7. If G is the RadiX-Net topology defined using $\{\overline{N}_i\}, \{D_i\}$ as defined in (4), and $\{\overline{N}_i\}$ has sufficiently small variance, then the density of a G varies as a function of μ and d (see (6)).

well as the number of paths from any node u in the input layer to a node v in the output layer for each family of topologies, can be found in the Appendix.

B. Asymptotic Sparsity of RadiX-Nets

Say G is the RadiX-Net topology generated by $\mathcal{N}^* = (\mathcal{N}_1, \ldots, \mathcal{N}_M), \mathcal{D} = (D_0, \ldots, D_{\overline{M}})$. Further say $\mathcal{N}_i = (N_{i,1}, \ldots, N_{i,L_i})$ for all i, and let N' be the integer satisfying $N' = \prod_{N \in \mathcal{N}_i} N$ for all $i \in \{1, \ldots, M-1\}$. If we define

$$(\overline{N}_1, \ldots, \overline{N}_{\overline{M}}) := (N_{1,1}, \ldots, N_{1,L_1}, N_{2,1}, \ldots, N_{M,L_M}),$$

then the density Δ_G of G is given by

$$\Delta_G = \left(\frac{1}{N'}\right)\left(\frac{\sum_{i=1}^{\overline{M}} \overline{N}_i D_{i-1} D_i}{\sum_{i=1}^{\overline{M}} D_{i-1} D_i}\right). \tag{4}$$

Let μ be the mean value of $\{\overline{N}_i\}$. When $\{\overline{N}_i\}$ has sufficiently small variance, it follows immediately from (4) that

$$\Delta_G \approx \frac{\mu}{N'}. \tag{5}$$

This implies that when $\{\overline{N}_i\}$ has small variance, the sparsity of G is negligibly affected by $\{D_i\}$.

We define $d = \log_\mu N'$. For sufficiently small variance of the $\overline{N}_i$, we can assume that d is approximately equal to some integer, with which we can write

$$\Delta_G \approx \frac{1}{\mu^{d-1}}. \tag{6}$$

Concretely, μ corresponds to the average radix of each mixed-radix numeral system used to construct G, and d corresponds to the number of radices used to construct each mixed-radix numeral system[‡][*]. The effect of μ and d on the sparsity of G is shown in Figure 7.

IV. Conclusions & Future Work

This paper presents the RadiX-Net algorithm, which deterministically generates sparse DNN topologies that, as a whole, are much more diverse than X-Net topologies while preserving X-Net's desired characteristics. In a related effort, benchmarking RadiX-Net performance in comparison to X-Net, dense DNN, and other neural network implementations can be found in [15]. Furthermore, RadiX-Net is used in [18] to construct a neural net simulating the size and sparsity of the human brain.

Prabhu *et al.* and Alford *et al.* come at the end of a long history of sparse neural network research [3]–[15]. This collective body mutually corroborates the following assertion: Sparse neural networks can train to the same arbitrary degree of precision as their dense counterparts. While the reduced training time of sparse neural nets can be attributed to having fewer parameters, there is no intuitive reason as to why sparse networks should demonstrate the same expressive power—as some have put it—as dense counterparts.

Naïvely, should sparse networks have the same expressive power as dense networks, dense and pruned networks would be obsolete, as *de novo* sparse networks achieve the expressive power of both while exceeding the training speed of both. Because the corpus of research in sparse networks seems unanimous on the subject, it would behoove the field to become more objective about what is meant when discussing expressive power, as is done in [19]–[21]. As demonstrated by [22], functional analysis provides a powerful language with which to describe the abilities and limitations of neural networks rigorously. In Section IV.B, we present a functional-analytic conjecture based on the mentioned experimental findings, which the authors intend to prove at a later date. Posing and proving such conjectures would direct future research in artificial neural networks more prudently than would experimental results alone.

A. Preliminaries for Conjecture

The most sturdy theoretical ground upon which artificial neural nets stand is Cybenko's Universality Theorem. Though the original statement of the theorem is stronger than the corollary below, this corollary captures the significance of the Universality Theorem in the field of artificial neural networks.

Corollary. Let $\sigma : \mathbb{R} \to \mathbb{R}$ be a continuous function such that $\lim_{t \to \infty} \sigma(t) = 1$ and $\lim_{t \to -\infty} \sigma(t) = 0$ (let us call this function sigmoidal). Further, let $\mathcal{C}_n$ be the space of continuous functions on $I_n = [0,1]^n$ with metric topology defined by supremum norm $d(f,g) = \sup_{\vec{x} \in I_n} |f(\vec{x}) - g(\vec{x})|$. Lastly, let S be the set of functions of the form

$$G(\vec{x}) = \sum_{j=1}^{N} \alpha_j \sigma\left(\vec{y}_j^T \vec{x} + \theta_j\right), \tag{7}$$

where N is a natural number, α_j and θ_j are real numbers, and $\vec{y}$ is an element of $\mathbb{R}^n$. The set S is dense $\mathcal{C}_n$. □

We adopt some of the language of this corollary to make our conjecture connect more immediately to the literature.

[‡] Per bullet 2) in Section III.A, this excludes the last mixed-radix numeral system.

[*] Note that this assumption is contingent on $\{N_i\}$ having sufficiently small variance.

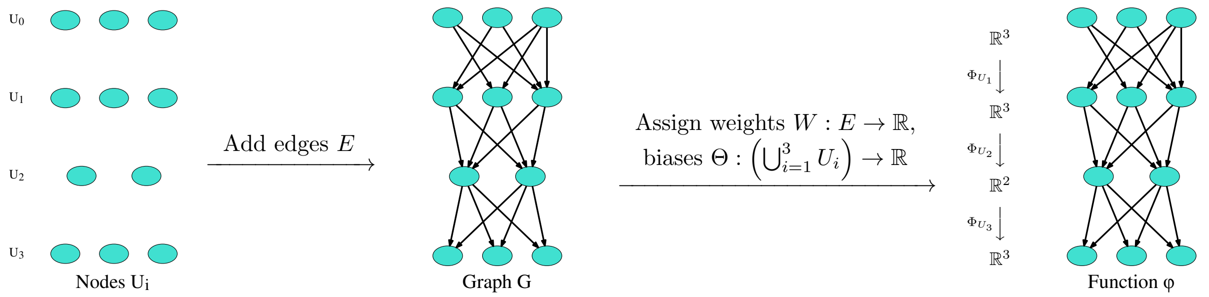


Fig. 8. Treating all activation functions as equal, a feedforward neural network (FNN) $\mathcal{G}$ is uniquely determine by an finite ordered collection of nodes $\mathcal{U}$, a set of edges E that makes $\mathcal{U}$ into an FNNT, a weights function $W : E \to \mathbb{R}$, and a bias function $\Theta : \left(\bigcup_{i \neq 0} U_i\right) \to \mathbb{R}$. Given m is the highest index of the $U_i \in \mathcal{U}$, $\mathcal{G}$ induces the unique function $\varphi : \mathbb{R}^{|U_0|} \to \mathbb{R}^{|U_m|}$ defined by $\varphi = \Phi_m \circ \ldots \circ \Phi_0$, where the element of $\Phi_i(\vec{x})$ corresponding to $u \in U_i$ is equal to $\varphi_u(\vec{x})$ (see (8)).

Let σ, I_n, $\mathcal{C}_n$, and d be as defined above. We define a feedforward neural network (FNN) as an FNNT G, with set of edges E, together with a map $W : E \to \mathbb{R}$ assigning a weight w to each edge and a map $\Theta : \bigcup_{i=1}^{m} U_i \to \mathbb{R}$—where m is the number of non-input layers in G—assigning a bias θ to each non-input node. We associate with each FNN $\mathcal{G}$ the unique map $\varphi : \mathbb{R}^{|U_0|} \to \mathbb{R}^{|U_m|}$ defined by the following:

- let $\tilde{E} : \bigcup_{i=1}^{m} U_i \to E$ map each node u to the set of edges going into u;
- for all $u_i \in U_0$, let $\varphi_{u_i}(x_1, \ldots, x_{|U_0|}) = x_i$;
- for all $i \in \{1, \ldots, m\}$ and for all $v \in U_i$, let

$$\varphi_v(\vec{x}) = \sigma\left(\Theta(v) + \sum_{(u,v)\in\tilde{E}(v)} W(u,v)\varphi_u(\vec{x})\right); \quad (8)$$

- assuming $U_m = \{u_1, \ldots, u_{|U_m|}\}$, we define

$$\varphi(\vec{x}) = \left(\varphi_{u_1}(\vec{x}), \ldots, \varphi_{u_{|U_m|}}(\vec{x})\right). \quad (9)$$

Let $\mathcal{U} = (U_0, U_1, \ldots)$ be an infinite ordered collection of finite sets of nodes such that $|U_0| = n$. Let $\mathfrak{D}$ be the unique fully-connected FNNT on $\mathcal{U}$, and let $\mathcal{S}$ be some sparse FNNT on $\mathcal{U}$ satisfying symmetry. We define $\mathfrak{D}_N$ and $\mathcal{S}_N$ as the unique FNNTs constructed by restricting $\mathfrak{D}$ and $\mathcal{S}$, respectively, to the set of nodes $\bigcup_{i=0}^{N} U_i$, introducing a new node v, and creating and edge from u to v for all $u \in U_N$. Finally, let $\mathbb{D}_N$ and $\mathbb{S}_N$ be the sets of continuous functions which can be represented as FNNs on $\mathfrak{D}_N$ and $\mathcal{S}_N$, respectively.

B. Functional-Analytic Conjecture

Due to the findings of Prabhu *et al.*, Alford *et al.*, and others, we are convinced that *de novo* sparse neural network topologies exhibit the same expressive power of fully-connected DNN topologies in the following way.

Conjecture. For all $\mathbb{X} \subset \mathcal{C}_n$, we define

$$\delta(\mathbb{X}) = \sup_{f\in\mathcal{C}_n}\left[\inf_{g\in\mathbb{X}}\left(d(f,g)\right)\right]. \quad (10)$$

If $\delta(\mathbb{D}_N)$ is in $O(N^{-p})$ for some p, then $\delta(\mathbb{S}_N)$ is also in $O(N^{-p})$. □

Acknowledgment

The authors wish to acknowledge the following individuals for their contributions and support: Simon Alford, Alan Edelman, Vijay Gadepally, Chris Hill, Hayden Jananthan, Lauren Milechin, Richard Wang, and the MIT SuperCloud team.

References

[1] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, June 2015.

[2] J. Kepner, V. Gadepally, H. Jananthan, L. Milechin, and S. Samsi, "Sparse deep neural network exact solutions," in *High Performance Extreme Computing Conference (HPEC)*, IEEE, 2018.

[3] Y. LeCun, J. S. Denker, and S. A. Solla, "Optimal brain damage," in *Advances in neural information processing systems*, pp. 598–605, 1990.

[4] B. Hassibi and D. G. Stork, "Second order derivatives for network pruning: Optimal brain surgeon," in *Advances in neural information processing systems*, pp. 164–171, 1993.

[5] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[6] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[7] S. Srinivas and R. V. Babu, "Data-free parameter pruning for deep neural networks," *CoRR*, vol. abs/1507.06149, 2015.

[8] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding," *CoRR*, vol. abs/1510.00149, 2015.

[9] B. Liu, M. Wang, H. Foroosh, M. Tappen, and M. Penksy, "Sparse convolutional neural networks," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 806–814, June 2015.

[10] J. Kepner and J. Gilbert, *Graph Algorithms in the Language of Linear Algebra*. SIAM, 2011.

[11] J. Kepner, M. Kumar, J. Moreira, P. Pattnaik, M. Serrano, and H. Tufo, "Enabling massive deep neural networks with the graphblas," in *High Performance Extreme Computing Conference (HPEC)*, IEEE, 2017.

[12] M. Kumar, W. Horn, J. Kepner, J. Moreira, and P. Pattnaik, "Ibm power9 and cognitive computing," *IBM Journal of Research and Development*, 2018.

[13] J. Kepner and H. Jananthan, *Mathematics of Big Data: Spreadsheets, Databases, Matrices, and Graphs*. MIT Press, 2018.

[14] A. Prabhu, G. Varma, and A. M. Namboodiri, “Deep expander networks: Efficient deep networks from graph theory,” *CoRR*, vol. abs/1711.08757, 2017.

[15] S. Alford and J. Kepner, “Training sparse neural networks,” in *MIT Undergraduate Research Technology Conference*, IEEE, 2018.

[16] S. P. Vadhan, *Pseudorandomness.* now, 2012.

[17] C. F. Loan, “The ubiquitous kronecker product,” *Journal of Computational and Applied Mathematics*, vol. 123, no. 1, pp. 85 – 100, 2000. Numerical Analysis 2000. Vol. III: Linear Algebra.

[18] R. Wang and J. Kepner, “Building a brain,” unpublished.

[19] Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang, “The expressive power of neural networks: A view from the width,” in *Advances in Neural Information Processing Systems 30* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), pp. 6231–6239, Curran Associates, Inc., 2017.

[20] V. Khrulkov, A. Novikov, and I. Oseledets, “Expressive power of recurrent neural networks,” in *International Conference on Learning Representations*, 2018.

[21] M. Raghu, B. Poole, J. Kleinberg, S. Ganguli, and J. Sohl-Dickstein, “On the expressive power of deep neural networks,” in *Proceedings of the 34th International Conference on Machine Learning* (D. Precup and Y. W. Teh, eds.), vol. 70 of *Proceedings of Machine Learning Research*, (International Convention Centre, Sydney, Australia), pp. 2847–2854, PMLR, 06–11 Aug 2017.

[22] G. Cybenko, “Approximation by superpositions of a sigmoidal function,” *Mathematics of Control, Signals and Systems*, vol. 2, pp. 303–314, Dec 1989.

Appendix

For purposes of simplifying Theorem 1, we use the following two lemmas. Lemma 2 discusses *extended mixed-radix topologies*, which we define as RadiX-Net topologies generated by $\mathcal{N}^*, \mathcal{D} = (D_0, \ldots, D_{\overline{M}})$ satisfying $D_i = 1$ for all i.

Lemma 1. Mixed-radix topologies satisfy symmetry, and the number of paths from an input node u to an output node v is one.

Proof. This follows directly from the definition of a mixed-radix numeral system. □

Lemma 2. Let G be the extended mixed-radix (EMR) topology defined by some $\mathcal{N}^* = (\mathcal{N}_1, \ldots, \mathcal{N}_M)$ satisfying the RadiX-Net constraints (see Section III: RadiX-Net Topologies). G satisfies symmetry, and the number of paths from an input node u to an output node v is $(N')^{M-1}$, where N' is the integer satisfying $N' = \prod_{N \in \mathcal{N}_i} N$ for all $i \in \{1, \ldots, M-1\}$.

Proof. We show this by induction. Say that, for some positive integer M, all EMR topologies G defined by some $\mathcal{N}^* = (\mathcal{N}_1, \ldots, \mathcal{N}_M)$ satisfy symmetry. Let $\mathcal{N}_+^* = (\mathcal{N}_1, \ldots, \mathcal{N}_M, \mathcal{N}_{M+1})$ for some $\mathcal{N}_{M+1}$ satisfying the RadiX-Net constraints, and let G_+ be the EMR topology induced by $\mathcal{N}_+^*$. Recall that G_+ is formed from the disjoint union of the MR topologies G_i (generated by $\mathcal{N}_i$) by identifying $U_{i-1,L_{i-1}}$ and $U_{i,0}$ for all i (here, U_{i,L_i} and $U_{i,0}$ simply refer to the output and input layers, respectively, of G_i). Because G_{M+1} is an MR topology, Lemma 1 guarantees that there exists exactly one path from u to v for all $u \in U_0^{M+1}$ and all $v \in U_{L_{M+1}}^{M+1}$. By hypothesis, for some positive integer m, there exist exactly m paths from $\tilde{u} \in U_{1,0}$ to $\tilde{v} \in U_{M,L_M}$ for all such $\tilde{u}, \tilde{v}$. Because U_{M,L_M} and $U_{M+1,0}$ are identified, this implies that for every path from $\tilde{u} \in U_{1,0}$ to $\tilde{v} \in U_{M,L_M}$, there exists exactly one path from $\tilde{u}$ to $v \in U_{M+1,L_{M+1}}$ which passes through $\tilde{v}$. Further, because there are $|U_{M+1,0}|$ such $\tilde{v}$, there exist exactly $m|U_{M+1,0}|$ paths from $\tilde{u}$ to v for all choices of $\tilde{u}, v$. By induction from the case $M = 1$ (i.e. Lemma 1), G_+ satisfies symmetry, and $m = \prod_{i=2}^{M} |U_{i,0}| = (N')^{M-1}$. □

Theorem 1. Let G be the RadiX-Net topology defined by some $\mathcal{N}^* = (\mathcal{N}_1, \ldots, \mathcal{N}_M), \mathcal{D} = (D_0, \ldots, D_{\overline{M}})$ satisfying the RadiX-Net constraints. We order the layers $\overline{U}_0, \ldots, \overline{U}_{\overline{M}}$ of G in the natural way, where $\overline{U}_0$ and $\overline{U}_{\overline{M}}$ are the input and output layers, respectively, of G. G satisfies symmetry, and the number of paths from input node u to output node v is given by $(N')^{\overline{M}-1} \left(\prod_{i=1}^{\overline{M}-1} D_i \right)$, where N' is the integer satisfying $N' = \prod_{N \in \mathcal{N}_i} N$ for all $i \in \{1, \ldots, M-1\}$.

Proof. Let $\mathbf{A}$ be the adjacency matrix of G, and let $\mathbf{W}_i^*, \mathbf{W}_i$ be as defined in (3). We define $\kappa = N' \sum_{i=0}^{\overline{M}} D_i$, $\alpha = N' D_0$, and $\beta = N' D_{\overline{M}}$. Up to a permutation, $\mathbf{A}$ is of the form

$$\left(\begin{array}{c|ccc} & \mathbf{W}_1^* \otimes \mathbf{W}_1 & & \\ \mathbf{0}_{\kappa-\beta,\alpha} & & \ddots & \\ & & & \mathbf{W}_{\overline{M}}^* \otimes \mathbf{W}_{\overline{M}} \\ \hline \mathbf{0}_{\beta,\alpha} & & \mathbf{0}_{\beta,\kappa-\alpha} & \end{array} \right). \quad (11)$$

Therefore, the following statements hold.

$$\begin{aligned} \mathbf{A}^{\overline{M}} &= \left(\begin{array}{c|c} \mathbf{0}_{\alpha,\kappa-\beta} & \prod_{i=1}^{\overline{M}} (\mathbf{W}_i^* \otimes \mathbf{W}_i) \\ \hline \mathbf{0}_{\kappa-\alpha,\kappa-\beta} & \mathbf{0}_{\kappa-\alpha,\beta} \end{array} \right) \\ &= \left(\begin{array}{c|c} \mathbf{0}_{\alpha,\kappa-\beta} & \left(\prod_{i=1}^{\overline{M}} \mathbf{W}_i^* \right) \otimes \left(\prod_{i=1}^{\overline{M}} \mathbf{W}_i \right) \\ \hline \mathbf{0}_{\kappa-\alpha,\kappa-\beta} & \mathbf{0}_{\kappa-\alpha,\beta} \end{array} \right) \end{aligned}$$

The deduction above is consequent of the mixed-product property of the Kronecker product [17]. It is easy to show that

$$\prod_{i=1}^{\overline{M}} \mathbf{W}_i^* = \left(\prod_{i=1}^{\overline{M}-1} D_i \right) \left(\mathbf{1}_{D_0, D_{\overline{M}}} \right), \quad (12)$$

where $\mathbf{1}_{a,b}$ is the $a \times b$ matrix of ones. By Lemma 2, it holds that

$$\prod_{i=1}^{\overline{M}} \mathbf{W}_i = (N')^{\overline{M}-1} \left(\mathbf{1}_{N',N'} \right). \quad (13)$$

Therefore,

$$\mathbf{A}^{\overline{M}} = \left(\begin{array}{c|c} \mathbf{0}_{\alpha,\kappa-\beta} & (N')^{\overline{M}-1} \left(\prod_{i=1}^{\overline{M}-1} D_i \right) \left(\mathbf{1}_{\alpha,\beta} \right) \\ \hline \mathbf{0}_{\kappa-\alpha,\kappa-\beta} & \mathbf{0}_{\kappa-\alpha,\beta} \end{array} \right).$$

So G satisfies symmetry, and for all input nodes u and output nodes v, there exist exactly $(N')^{\overline{M}-1} \left(\prod_{i=1}^{\overline{M}-1} D_i \right)$ paths from u to v. □