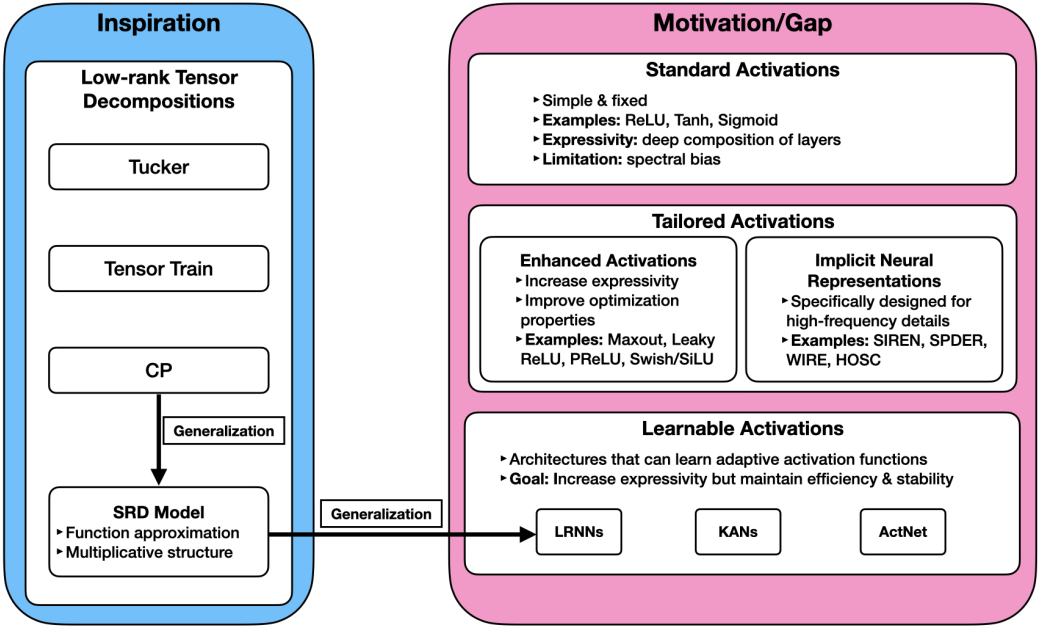
Deep Learning with Learnable Product-Structured Activations

Saanjali Maharaj, Prasanth B. Nair

University of Toronto Institute for Aerospace Studies


Inspiration
Low-rank Tensor Decompositions
Tucker
Tensor Train
CP
Generalization
SRD Model
Function approximation
Multiplicative structure
Generalization
Motivation/Gap
Standard Activations
Simple & fixed
Examples: ReLU, Tanh, Sigmoid
Expressivity: deep composition of layers
Limitation: spectral bias
Tailored Activations
Enhanced Activations
Increase expressivity
Improve optimization properties
Examples: Maxout, Leaky ReLU, PReLU, Swish/SiLU
Implicit Neural Representations
Specifically designed for high-frequency details
Examples: SIREN, SPDER, WIRE, HOSC
Learnable Activations
Architectures that can learn adaptive activation functions
Goal: Increase expressivity but maintain efficiency & stability
LRNNs
KANs
ActNet

Deep Low-Rank Separated Neural Networks (LRNNs)

The expressivity of LRNNs stems from a **multiplicative composition** of **learnable**, **univariate** functions.

LRNNs generalize low-rank tensor decomposition for function approximation to create a new class of deep neural network architectures with learnable product-structured activations.

Advantages:

- inherently capture multiplicative interations
- each neuron can independently learn a highly flexible activation function

General Shallow LRNN Model

We consider supervised learning with inputs $\mathrm{x} \in \mathbb{R}^d$ and outputs y, and seek a predictive model $\hat{y}(x)$.

$$\widehat{\mathbf{y}}_{\mathrm{lrnn}}(\mathbf{x}) = \sum_{\ell=1}^{r} \mathbf{s}_\ell \prod_{j=1}^{\bar{d}} (1 + \gamma\, g_j^\ell(z_j^\ell)), \quad \mathbf{z}^\ell = \mathbf{W}^\ell \mathbf{x} + \mathbf{b}^\ell,$$

where $r \in \mathbb{N}$ is the separation rank, $\mathbf{s}_\ell \in \mathbb{R}^K$ are weight vectors, $g_j^\ell : \mathbb{R} \to \mathbb{R}$ denotes a univariate component function, $\gamma = \bar{d}^{-1/2}$ is a scaling factor, $\mathbf{z}^\ell \in \mathbb{R}^{\bar{d}}$, $\mathbf{W}^\ell \in \mathbb{R}^{\bar{d} \times d}$, and $\mathbf{b}^\ell \in \mathbb{R}^{\bar{d}}$.

- LRNNs project the d-dimensional input to r latent vectors in $\mathbb{R}^{\bar{d}}$, then combine them through a sum-product operation.
- $(1 + \gamma\, g_j^\ell(z_j^\ell))$ ensures automatic relevance determination (ARD) and makes initialization more convenient.
- $\gamma = \bar{d}^{-1/2}$ plays a crucial role analogous to Xavier/He initialization in standard networks.

General Shallow LRNN Model

$$\widehat{\mathbf{y}}_{\mathrm{lrnn}}(\mathbf{x}) = \sum_{\ell=1}^{r} \mathbf{s}_\ell \prod_{j=1}^{\bar{d}} (1 + \gamma\, g_j^\ell(z_j^\ell)), \quad \mathbf{z}^\ell = \mathbf{W}^\ell \mathbf{x} + \mathbf{b}^\ell,$$

where $r \in \mathbb{N}$ is the separation rank, $\mathbf{s}_\ell \in \mathbb{R}^K$ are weight vectors, $g_j^\ell : \mathbb{R} \to \mathbb{R}$ denotes a univariate component function, $\gamma = \bar{d}^{-1/2}$ is a scaling factor, $\mathbf{z}^\ell \in \mathbb{R}^{\bar{d}}$, $\mathbf{W}^\ell \in \mathbb{R}^{\bar{d} \times d}$, and $\mathbf{b}^\ell \in \mathbb{R}^{\bar{d}}$.

- Hyperameters: separation rank (r), dimensionality of the linear projection layer ($\bar{d}$).
- g_j^ℓ can be parametrized by $r\bar{d}$ shallow neural networks (NNs).
- The NN parameters and the weight vectors ($\mathbf{s}_\ell$), are learnt by minimizing an appropriate loss function over the training dataset.

Variance-controlled Initialization

Lemma 1

The product-structured LRNN activation $\varphi(\mathbf{z}) = \prod_{j=1}^{\bar{d}}(1 + \gamma\, g_j(z_j))$ satisfies the following bounds:

$$(i)\ \mathrm{Var}[\varphi(\mathbf{z})] \ \leq\ e^{\sigma_g^2} - 1 \quad \text{and} \quad (ii)\ \sum_{k=1}^{\bar{d}} \mathrm{Var}[\partial\varphi(\mathbf{z})/\partial z_k] \ \leq\ \sigma_{g'}^2\, e^{\sigma_g^2}$$

where σ_g^2 and $\sigma_{g'}^2$ denote the variance of the component functions and their first-order derivatives, respectively, at initialization.

- Holds under mild assumptions on g_j at initialization (zero mean, finite variance).
- Activation variance and total gradient variance remain bounded independently of $\bar{d}$.
- This ensures stable forward and backward propagation through wide product structures.

Connection to Multilayer Perceptron (MLP)

The proposed LRNN model reduces to a standard shallow MLP in the special case where

- $\bar{d} = 1$,
- g_j^ℓ are replaced with standard activation functions.

Shallow MLP (r hidden neurons)

$$\mathbf{y}_{\text{mlp}}(\mathbf{x}) = \sum_{\ell=1}^{r} \mathbf{v}_\ell \, \sigma(z_\ell),$$

where $z_\ell = \mathbf{w}_\ell^T \mathbf{x} + b_\ell$ is a scalar projection of the input with $\mathbf{w}_\ell \in \mathbb{R}^d, \mathbf{v}_\ell \in \mathbb{R}^K$, and $b_\ell \in \mathbb{R}$ and $\sigma : \mathbb{R} \to \mathbb{R}$ is a standard MLP activation function.

Shallow LRNN (separation rank r)

$$\mathbf{y}_{\text{lrnn}}(\mathbf{x}) = \sum_{\ell=1}^{r} \mathbf{s}_\ell \, \varphi_\ell(\mathbf{z}^\ell),$$

where $\varphi_\ell(\mathbf{z}^\ell) = \prod_{j=1}^{\bar{d}} (1 + \gamma \, g_j^\ell(z_j^\ell))$ is the LRNN product-structured activation function with $\mathbf{z}^\ell = \mathbf{W}^\ell \mathbf{x} + \mathbf{b}^\ell$.

Distinction between MLP & LRNN

MLP	**LRNN**
All MLP neurons share the same *fixed* activation $\sigma : \mathbb{R} \to \mathbb{R}$ operating on scalar projections[1].	Each LRNN neuron learns its own *distinct learnable* activation function $\varphi_\ell : \mathbb{R}^{\bar{d}} \to \mathbb{R}$.
Standard activations have limitations such as spectral bias hindering representation of high-frequency details in signals.	LRNN activations achieve vector-to-scalar mapping through multiplicative compositions, enabling efficient representation of higher-order interactions that additive architectures struggle to capture.

[1] Maxout networks are a notable exception, also using vector-to-scalar mappings but through max operations rather than products.

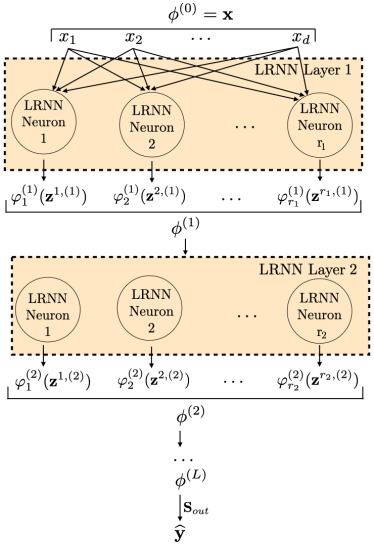
General Deep LRNN Architecture

Deep LRNN model with L layers

$$\widehat{\mathbf{y}}(\mathbf{x}) = \mathbf{S}^{\text{out}}(\phi^{(L)} \circ \phi^{(L-1)} \circ \ldots \circ \phi^{(1)})(\mathbf{x}),$$

where $\mathbf{S}^{\text{out}} \in \mathbb{R}^{K \times r_L}$ and $\phi^{(k)} : \mathbb{R}^{r_{k-1}} \to \mathbb{R}^{r_k}$, with $r_0 = d$ and $\phi^{(0)} = \mathbf{x}$.

- k-th hidden layer has r_k neurons.
- Maps a d-dimensional input to a K-dimensional output.
- LayerNorm applied to each $\phi^{(k)}$ for stable learning dynamics.



Universal Approximation by LRNNs

Theorem 1

If $f : [0,1]^d \to \mathbb{R}$ is a continuous function, then for every $\varepsilon > 0$, there exists an LRNN with suitably chosen separation rank r such that $\max_{\mathbf{x}\in[0,1]^d} \left| f(\mathbf{x}) - f_{\mathrm{lrnn}}(\mathbf{x}) \right| \leq \varepsilon$.

- Establishes universality analogous to that of standard MLPs.
- Just as the width of an MLP may grow with $1/\varepsilon$, the rank r of an LRNN can grow arbitrarily large to capture complex functions. Thus, "universal" here does not guarantee a small r unless the target function has low-rank/near-separable structure.

Curse of Dimensionality Mitigation

Theorem 2

For functions whose ANOVA decomposition is dominated by terms involving at most $m \ll d$ variables, LRNNs achieve approximation error ε with parameter complexity $\mathcal{O}(\text{poly}(d)/\varepsilon)$ rather than exponential in d.

- LRNNs can mitigate the curse of dimensionality for a class of structured functions since the parameter complexity grows only polynomially with d rather than the exponential scaling typical of generic approximators.
- LRNNs naturally encode sum-of-products structures matching ANOVA decompositions.
- Functions arising from physical systems often exhibit such decay in interaction order, making LRNNs particularly suitable for scientific computing applications.

Adaptive Spectral Bias Control

Lemma 2

When equipped with periodic activations (e.g., SIREN, SPDER), LRNNs with $\bar{d} > 1$ generate rich frequency spectra through combinatorial frequency synthesis. A single LRNN neuron with $\bar{d}$ components generates not only the $\bar{d}$ fundamental frequencies but also all $2^{\bar{d}} - 1$ possible sum and difference combinations.

- This multiplicative frequency synthesis contrasts with MLPs' additive synthesis, where each neuron contributes a single frequency pair.
- Consequently, LRNNs can represent complex spectra with fewer parameters, particularly for signals with harmonic relationships or intermodulation products.
- This explains their superior performance on audio and image representation tasks where the ability to capture high-frequency details is crucial.

Image Representation: Large-Scale Study

- Compared PSNR of LRNN-SPDER, SPDER, SIREN.
- All models have 200k parameters.
- ImageNet study: 1000 images, 3 random seeds per image.

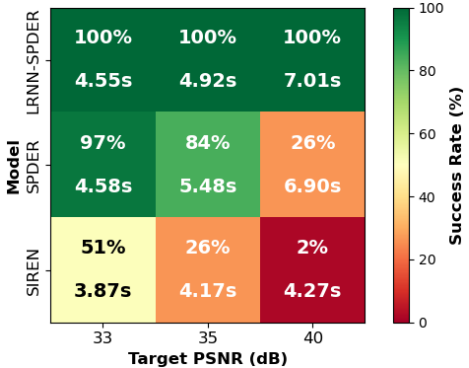


Figure 1: Average success rate and time for models to reach PSNR targets.

Image Representation: Large-Scale Study

Figure 2: Qualitative comparison of ImageNet reconstructions after 250 epochs. Selected examples demonstrate that LRNNs capture fine details significantly earlier in training than baseline models.

Audio Representation

- LRNN activations: $\sin(x)\arctan(x)$

Table 1: MSE loss and ρ_{AG} across architectures.

Method	MSE Loss ($\times 10^{-4}$)			
	bach	counting	reggae	reading
SIREN	1.21(0.28)	2.77(0.56)	21.5(6.3)	9.98(1.57)
SPDER	1.12(0.05)	2.29(0.55)	24.8(7.7)	8.88(2.45)
LRNN-SPDER	**0.10(0.01)**	**0.72(0.03)**	**7.93(0.11)**	**1.86(0.30)**
	ρ_{AG} (std $\times 10^{-4}$)			
SIREN	0.9986(5)	0.9906(15)	0.9769(11)	0.9193(94)
SPDER	0.9988(3)	0.9937(6)	0.9729(10)	0.9324(104)
LRNN-SPDER	**0.9999(0)**	**0.9967(2)**	**0.9860(2)**	**0.9862(31)**

Audio Representation

- LRNN activations: $\sin(x)\arctan(x)$

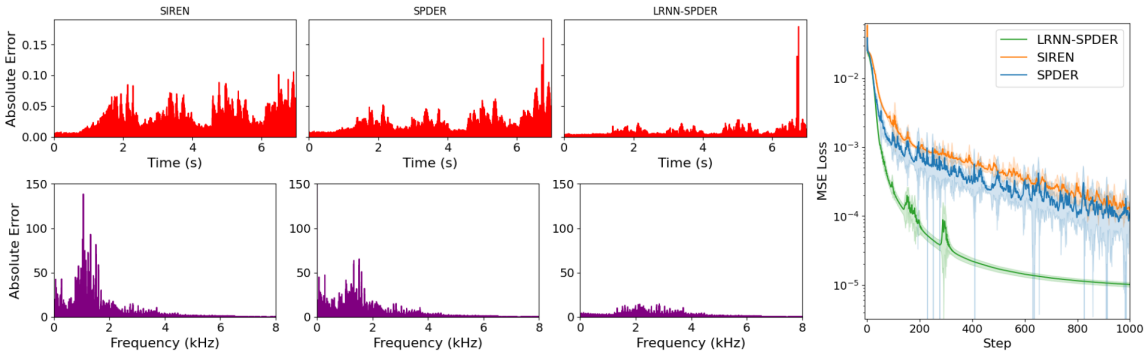


Figure 3: Absolute error in time and frequency domain and convergence of training MSE loss (mean $\pm 1\sigma$) for bach audio representation tasks for comparably sized models.

PDE Benchmark

- High-frequency Poisson PDE benchmark.
- LRNN activations: $\sin(x)$
- Compared LRNN to SIREN, MLPs, KANs for different parameter counts.

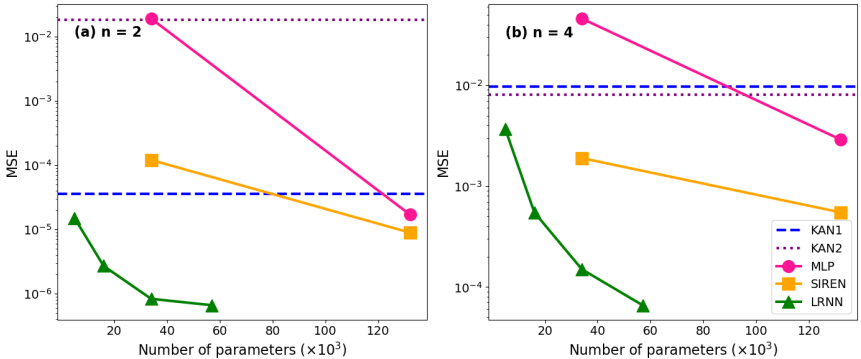


Figure 4: Results for the PDE benchmark.

CT Reconstruction

- Sparse-view Computed Tomography (CT) is vital for reducing patient radiation exposure, and INRs can reconstruct high-fidelity images from such limited data.
- Compared LRNN to WIRE, SIREN, Gauss, and ReLU with positional encoding on a 256×256 chest CT image task using $\sim$180k parameters for all models.
- LRNN's artifact-free reconstruction from limited projections has direct clinical implications for reducing patient radiation exposure while maintaining diagnostic quality.

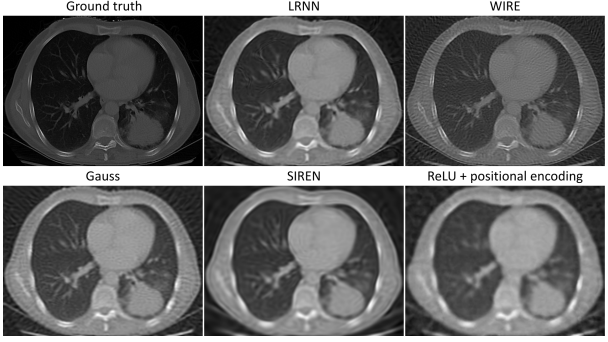


Table 2: CT model performance comparison.

Model	PSNR	SSIM
LRNN	**29.13**	**0.7455**
WIRE	28.83	0.6413
Gauss	27.84	0.6855
SIREN	27.46	0.6877
ReLU	26.89	0.6341

Conclusions

- **Novel architecture:** deep LRNNs effectively capture higher-order interations.
- **Theoretical analysis:** universal approximation, curse of dimensionality mitigation, adaptive spectral-bias control.
- **Numerical results:** significant potential across several domans.
- **Future research:** video modeling, unsteady PDEs, NeRFs, generalizable INR.
- **Refinement:** memory footprint of backward pass, kernel fusion, mixed-precision training.