Meta-learning Sparse Implicit Neural Representations

Jaeho Lee♠* Jihoon Tack♠* Namhoon Lee♦ Jinwoo Shin♠

♠KAIST ♦UNIST *equal

jaeho-lee@kaist.ac.kr

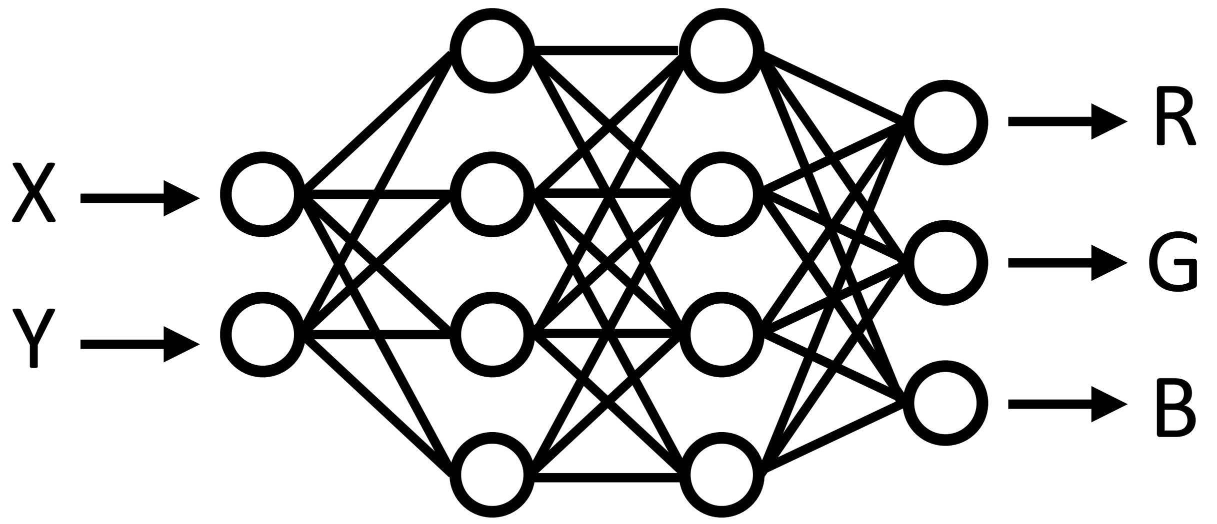
Summary. Implicit Neural Representation (INR) is a new paradigm to represent each data point as a neural net.
We propose a compute efficient meta-learning method to train sparse INRs of a dataset.

arXiv GitHub

Background

INR represents each data as a neural network approximating coordinate-to-signal mappings.

For instance, an RGB image can be represented by a function $f_\theta : \mathbb{R}^2 \to \mathbb{R}^3$



INRs became popular, as it has several practical advantages over classical methods (e.g., pixels).

Challenge: Scalability

If we want to represent a large-scale dataset, we need a huge computation & storage capacity.

Q. Can we efficiently train Sparse INRs for a large number of data?
≈ How can we efficiently train sparse models for multiple related datasets?

Framework

We formulate this problem as learning a good "Sparse Initial INR" which requires small training budget to fit each data.

Formally, we can write as

$$\min_{\theta^{(0)}} \min_{\substack{M \in \{0,1\}^d \\ \|M\|_0 \le \kappa}} \frac{1}{N} \sum_{j=1}^{N} \mathscr{L}_j\big(M \odot \theta^{(t)}(T_j, \theta^{(0)})\big)$$

Here, $(M, \theta^{(0)})$ is the mask & initial parameter, $\theta^{(t)}(T_j, \theta^{(0)})$ denote a t-step SGD-updated $\theta^{(0)}$ to fit the image T_j, and $\mathscr{L}_j$ denotes the loss with respect to the image T_j.

Method: Meta-Sparse INR

We propose using a three-step procedure.

1. Train an initialized INR over a set of images using a meta-learning algorithm (e.g., MAML).

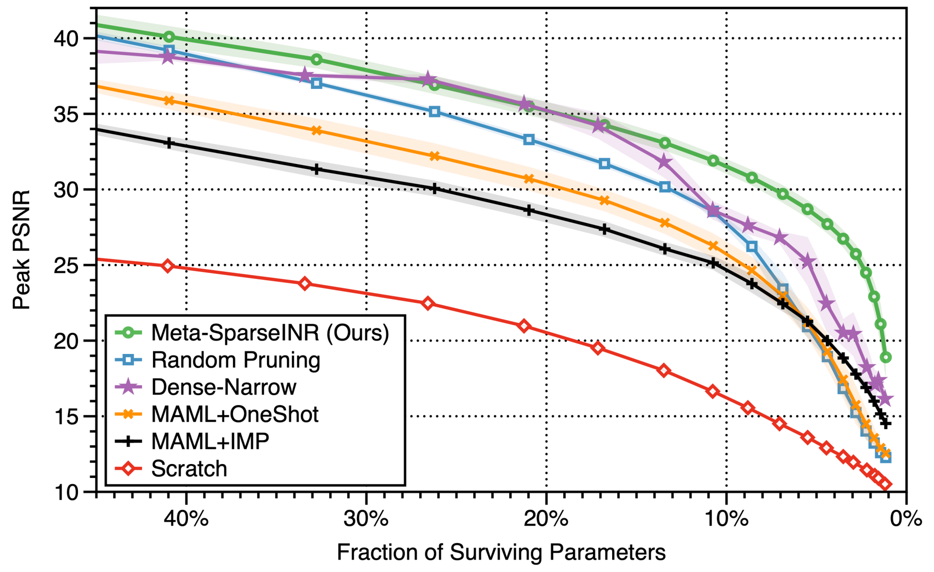
2. Prune a fraction of surviving weights using the magnitude-based pruning.

3. Retrain the INR, and go back to **2** (if needed)

*Note: Pruning-at-initialization methods didn't outperform dense-narrow baselines, even for a single image case.

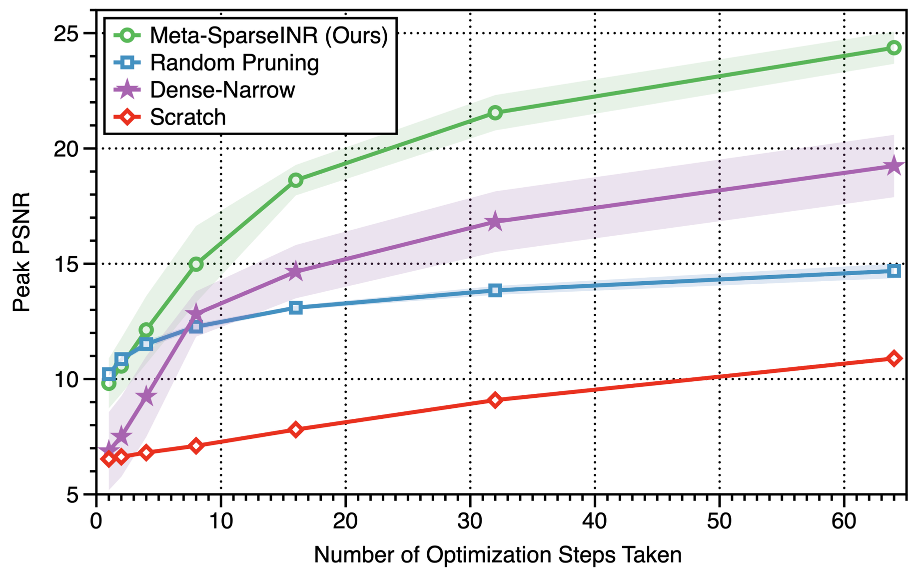
Experiment

Our method outperforms dense-narrow baselines when given a fixed fine-tuning budget.

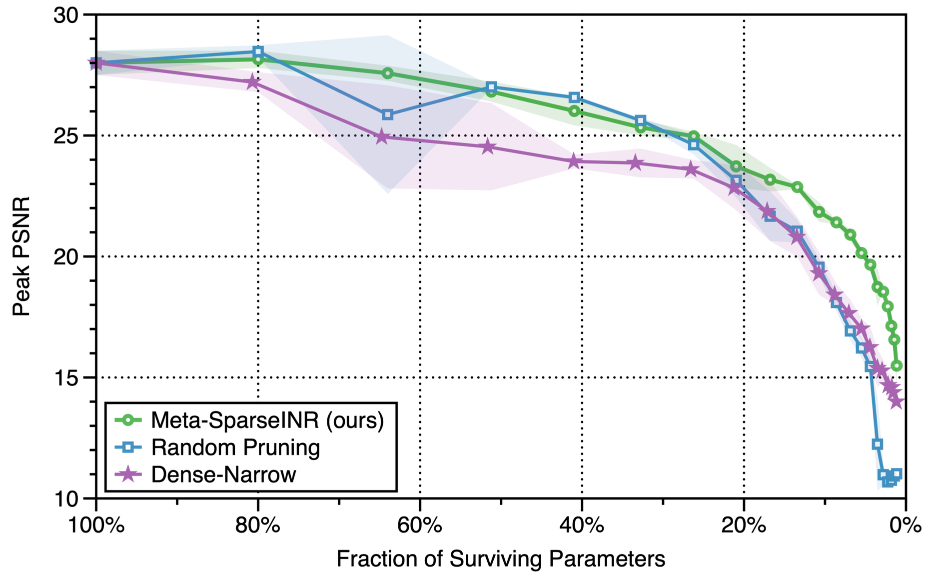


Dataset	Method	PSNR	#Params
CelebA	Meta-SparseINR (Ours)	27.71	8,704
CelebA	Random Pruning	26.25	17,000
CelebA	Dense-Narrow	27.68	17,708
Imagenette	Meta-SparseINR (Ours)	25.74	8,704
Imagenette	Random Pruning	24.09	17,000
Imagenette	Dense-Narrow	24.76	14,212
SDF	Meta-SparseINR (Ours)	49.92	8,704
SDF	Random Pruning	47.48	17,000
SDF	Dense-Narrow	44.61	26,978

SIREN on CelebA, Imagenette and SDF, when one use 100 SGD steps for fitting.



PSNR vs. number of optimization step



Two-step optimization