Sparse network initialization using deterministic Ramanujan graphs

Arindam Biswas [*1] **Suryam Arnav Kalra** [*2] **Pabitra Mitra** [2] **Biswajit Basu** [3]

Abstract

We introduce a sparsely connected neural network architecture inspired by Ramanujan graphs, which achieves performance comparable to dense networks. They are constructed from Cayley graphs of specific algebraic groups or as Ramanujan r-coverings of the full (k, l) bi-regular bipartite graph with $k + l$ vertices. This novel method employs zero-shot, data-independent, deterministic pruning at initialization, facilitating early identification of winning lottery tickets. Unlike traditional methods that rely on iterative processes to find these tickets, our technique identifies them at the outset. Our ultimate goal is to construct sparse, scalable Foundation Models. Experimental results demonstrate that our proposed architecture achieves competitive accuracy and sparsity ratios comparable to those obtained by previous pre-training pruning algorithms.

1. Introduction

Sparse neural architectures are attractive due to their parameter parsimony and reduced training time. Existence of sparse high performing subnetworks of a backbone dense network forms the basis of the well known lottery ticket hypothesis (Frankle & Carbin, 2019). Several approaches have been directed towards identifying winning lottery tickets with a minimal effort. Initial research were based on applying established pruning algorithms on a partially trained network (Renda et al., 2020; Fischer & Burkholz, 2022). Recently, a number of approaches has been suggested to obtain a sparse mask for pruning at initialization (PaI) (Frankle et al., 2020; Wang et al., 2021; Sreenivasan et al., 2022). These method use the structure of the initialized network, in a data dependent or independent manner, to prune the network to a high sparsity ratio (Sreenivasan et al., 2022; Lee et al., 2019a;b; Wang et al., 2020; Tanaka et al., 2020). Most of these techniques are multi-shot, obtaining desired connectivity structures from a random network initialization. Zero-shot pruning aims to construct an initialization topology without the need for iteration over network structures. We show that deterministic constructions of Ramanujan expander graphs can be effectively used for zero-shot pruning.

We propose a deterministic sparse network initialization technique based on Ramanujan graphs that are constructed either as Cayley graphs of certain algebraic groups or as Ramanujan r-coverings of the full (k, l) bi-regular bipartite graph on $k + l$ vertices. Prior approaches to using Ramanujan expander graphs for PaI have relied mainly on constructions based on iterated magnitude pruning techniques or generating random graphs. This often leads to the formation of irregular graph networks that do not strictly adhere to the rigorous definition of Ramanujan graphs. Our approach of constructing a deterministic Ramanujan network circumvents this problem. Ramanujan initializers using these bipartite graphs suitably represent the fully connected as well as the convolutional layers.

The sparse networks generated are data independent, structurally pre-defined, with a static mask across the training iterations. The deterministic construction algorithm does not introduce unwanted pseudorandomness in the generated graph, which was a focal point why in a recent work, the notion of IMDB score was introduced, see (Hoang et al., 2023).

Experimental results on benchmark image classification data sets show that Ramanujan sparse network initialization provides comparable performance with dense networks.

1.1. Related Work

Expander based winning lottery ticket generation has been studied in (Stewart et al., 2023). The methodology is based on generating random d-regular graphs for the bipartite layers. These graphs are Ramanujan with a high probability. A deep expander sparse network, the X-Net, is presented in (Prabhu et al., 2018b). It is constructed by sampling d-left regular graphs from the space of all bipartite graphs. Ramanujan graph based sparsity aware network initialization is proposed in (Esguerra et al., 2023).

[*]Equal contribution [1]Polynom, France
[2]Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur
[3]School of Engineering, Trinity College Dublin
. Correspondence to: Arindam Biswas <arin.math@gmail.com>, Suryam Arnav Kalra <suryamkalra35@gmail.com>.

Work presented at TF2M workshop at ICML 2024, Vienna, Austria. PMLR 235, 2024.

One-shot neural network pruning using spectral sparsification is presented in (Laenen, 2023). It is based on the effective resistance algorithm for obtaining spectrally sparse bipartite graphs. RadiX-Net (Kepner & Robinett, 2019) is a deterministic sparse neural architecture with mixed-radix topologies. It has desirable symmetry properties that preserves path connectedness and eliminates training bias. Connectivity properties are used in other graph theoretic initialization schemes that define an initial sparse network topology (Vysogorets & Kempe, 2023; Chen et al., 2022; 2023).

2. Research Gap and Contributions

Existing pruning at initialization techniques are often iterative and data dependent. Zero-shot data independent algorithms have advantages in terms of reduced computational overhead and generalization capabilities. Recently, there has been a flurry of works on the construction of pruned sparse networks based on expansion (Prabhu et al., 2018a; Kepner & Robinett, 2019; Pal et al., 2022; Stewart et al., 2023; Arnav Kalra et al., 2024). These methods use a random network structure and are data dependent. None could guarantee the following three properties at the same time: (i) Ramanujan property - allows us to construct the best possible expanders given a set of vertices and maintaining a high level of sparsity, (ii) Path-connectedness - a desirable property for all PaI architectures, and, (iii) High symmetricity (see for instance 3.3 and A.1) - a desirable property for computational purposes.

The principal contributions of our paper are:

1. Proposing a new technique of zero-shot pruning neural networks without using any data.
2. We present a deterministic Ramanujan graph construction technique for initializing sparse neural networks. To the best of our knowledge, no other work exists towards this direction.
3. Establishing that training sparse networks directly without previous pruning can work if the sparsification is done via the use of deterministic Ramanujan graphs. Previous research have indicated that vanilla training of sparse random networks are often unsuccessful to identify winning lottery tickets (Zhou et al., 2019).
4. In most previous works, for the identification of winning lottery ticket, sufficient to reach good generalization, is typically determined in an iterative fashion. However, zero-shot identification is more attractive (Tartaglione, 2022), which we develop in this work.

Further, identifying the sparse existant pathways and their trained weights can help in better explainabality and enables training with reduced computational effort.

3. Formulation of Sparse Neural Ramanujan Graphs

Definition 3.1 (Bipartite Ramanujan graphs)**.** Let $\Gamma = (V, E)$ be a d-regular $(d \geq 3)$ balanced bipartite graph. Let the eigenvalues of its adjacency matrix be $\lambda_n \leq \lambda_{n-1} \leq \cdots \leq \lambda_2 \leq \lambda_1$. Then Γ is said to be Ramanujan iff $|\lambda_i| \leq 2\sqrt{d-1}$, for $i = 2, \ldots, (n-1)$.

For an unbalanced $(d_1, d_2)-$biregular bipartite graph $(d_1, d_2 \geq 3)$, the condition of being Ramanujan changes to $|\lambda_i| \leq \sqrt{d_1 - 1} + \sqrt{d_2 - 1}$, for $i = 2, \ldots, (n-1)$. We see that when $d_1 = d_2$, it transforms to the usual definition. A detailed description of Ramanujan graphs can be found in (Hoory et al., 2006, sec. 5.3).

3.1. Regular Ramanujan graphs

Let $p, q \equiv 1 (mod\, 4)$ be distinct odd primes (the condition of $1 (mod\, 4)$ can be removed at the cost of making the analysis more technical and complicated). The graph $X^{p,q}$ is constructed using the following general method.

1. It is a Cayley graph (see Appendix A.1) on the subgroup of 2 by 2 matrices, $PGL_2(\mathbb{F}_q)$ where $\mathbb{F}_q$ is the finite field of characteristic q.
2. Consider the equation $a_0^2 + a_1^2 + a_2^2 + a_3^2 = p$. Jacobi's four square theorem states that there are $p + 1$ solutions to the equation $a_0^2 + a_1^2 + a_2^2 + a_3^2 = p$ with $a_0 > 0$ odd (i.e., $a_0 \equiv 1$ (mod 2)) and a_1, a_2, a_3 even. Now, for each such solution (a_0, a_1, a_2, a_3) consider the matrix $\begin{pmatrix} a_0 + ia_1 & a_2 + ia_3 \\ -a_2 + ia_3 & a_0 - ia_1 \end{pmatrix}$ where i is some fixed solution to $i^2 = -1$ (mod q).
3. Form the generating set S of the Cayley graph to be the set of these $(p + 1)$ matrices. Thus $X^{p,q} = Cay(PGL_2(\mathbb{F}_q), S)$.
4. The graphs are bipartite iff p is not a quadratic residue modulo q or in other words the Legendre symbol $\left(\frac{q}{p}\right) = -1$. The bipartite graphs $X^{p,q}$ will be $(p+1)$-regular, of size $\frac{q(q^2-1)}{2}$ by $\frac{q(q^2-1)}{2}$ and are Ramanujan (Lubotzky et al., 1988).

3.2. Construction of the fully connected layers

For the fully connected layers consisting of balanced bipartite graphs, we prune them at initialization in accordance with the Ramanujan graph structure of LPS. For this we select a prime q such that $\frac{q(q^2-1)}{2}$ by $\frac{q(q^2-1)}{2}$ is closest to the size of the original bipartite layer. We then select the prime p such that the Legendre symbol $\left(\frac{q}{p}\right) = -1$.

3.3. Bi-regular Ramanujan graphs and construction of convolutional layers

Fix a prime q and a $q \times q$ cyclic shift permutation matrix $P = [P]_{ij}$ with $[P]_{ij} = 1$ if $j = i - 1 \pmod q$ and 0 otherwise. Recall that the adjacency matrix of any $m \times n$ bipartite graph can be written as $Adj = \begin{pmatrix} 0_{m\times m} & B_{m\times n} \\ B^T_{m\times n} & 0_{n\times n} \end{pmatrix}$, where B is called the bi-adjacency matrix. Define the bi-adjacency matrix of this bipartite graph to be $B = \begin{pmatrix} I_q & I_q & \dots & I_q \\ I_q & P & \dots & P^{l-1} \\ I_q & P^2 & \dots & P^{2(l-1)} \\ \vdots & & & \\ I_q & P^{q-1} & \dots & P^{(q-1)(l-1)} \end{pmatrix}$ where I_q is the $q \times q$ identity matrix and P is as above. B is a $q^2 \times lq$ matrix and the bipartite graph is either $q^2 \times lq$ with bi-regularity (l, q) or symmetrically $lq \times q^2$ with bi-regularity (q, l). The graphs whose bi-adjacency matrices are represented as B (or B^T) are Ramanujan, see (Burnwal et al., 2021). The adjacency matrices constructed from the above graphs are also high structured being essentially collections of shift permutation matrices. This structure is lost when we pass to random constructions. These graphs are explicit realisations of the Ramanujan r-coverings of the full (k, l) bi-regular bipartite graph on $k + l$ vertices as shown in (Hall et al., 2018, cor 2.2). For pruning the convolution layers, we utilise the bi-regular Ramanujan graphs.

4. Experimental Methodology and Results

The goal of our experiments is to study the effectiveness of deterministic Ramanujan graph based sparse network initialization.

4.1. Datasets and architectures

The datasets used for the experiments are Cifar-10 and Cifar-100 (Krizhevsky, 2009). The experiments are performed over a variety of architectures including VGG13, VGG16, VGG19 (Simonyan & Zisserman, 2014), AlexNet (Krizhevsky et al., 2012), ResNet18 and ResNet34 (He et al., 2016) to show the robustness of our method. We proceed in two parts. In the first part, we prune the intermediate Fully Connected layers by replacing them with sparse Ramanujan Graph which is applicable for VGG13, VGG19 and AlexNet architectures. In the second part, we prune the whole network including the Convolution layers and the Fully Connected layers which is applicable for all the architectures considered in our experiment. The performance of the dense and the pruned networks are compared in each case. Finally, we compare the performance of our method against various state-of-the art PaI algorithms for VGG16 and the ResNet34 architectures. Training parameters for all of the architectures are same and are summarized in Table 1. We report accuracy on a randomly split 16% test set for all the experiments.

Table 1. Training Parameters for the experiment

Hyperparmeters	
Epochs	200
Train Batch Size	256
Test Batch Size	128
Learning Rate	0.1
LR Decay, Epoch	10x, [100, 150]
Optimizer	SGD
Weight Decay	0.0005
Momentum	0.9
Weight Initialization	Kaiming Uniform

4.2. Methods compared

The performance of the pruned networks are compared with that of corresponding dense networks. We have also compared our method against various pruning at initialization techniques such as Random (Liu et al., 2022), ERK (Evci et al., 2020; Mocanu et al., 2018), GraSP (Wang et al., 2020), and SynFlow (Tanaka et al., 2020). The number of iterations for SynFlow are 100 while for GraSP and ERK it is 1 keeping rest of the hyperparameters same as Table 1.

4.3. Results and discussion

Results for the first part of the experiment where only the intermediate fully connected layer is pruned, are summarized in Table 2 and 3 for the Cifar-10 and Cifar-100 datasets respectively. It can be observed that the Ramanujan graph construction allows us to extremely prune the fully connected layer upto **0.43%** while still retaining the accuracy as of the unpruned model.

Table 2. Accuracy of VGG and AlexNet when only the intermediate fully connected layer is pruned on Cifar-10 dataset

Dataset: Cifar-10			
Model	FC layer Size (Remaining Edge Percentage)		
	4096×4096 (Unpruned)	$\mathbf{2448 \times 110}$ (**1.6%**)	$\mathbf{2448 \times 30}$ (**0.43%**)
VGG13	92%	**91%**	**91%**
VGG19	92%	**92%**	**92%**
AlexNet	86%	**84%**	**86%**

For the second part of the experiment where we prune the complete network including the convolution layers and the fully connected layer, we could achieve an overall pruning percentage of ∼2% for VGG, ∼2.3% for AlexNet and ∼5% for the ResNet architectures. The accuracy of the models on the Cifar-10 and Cifar-100 datasets is given in Table 4.

Table 3. Accuracy of VGG and AlexNet when only the intermediate fully connected layer is pruned on Cifar-100 dataset

Dataset: Cifar-100			
Model	FC layer Size (Remaining Edge Percentage)		
	4096×4096 (Unpruned)	$\mathbf{2448 \times 110}$ ($\mathbf{1.6\%}$)	$\mathbf{2448 \times 30}$ ($\mathbf{0.43\%}$)
VGG13	66%	**66%**	**63%**
VGG19	66%	**67%**	**63%**
AlexNet	67%	**66%**	**66%**

Table 4. Accuracy of various architectures when the complete network is pruned including the Convolution and the FC layers.

Dataset: Cifar-10			
Model	Unpruned accuracy	Pruned Accuracy	Network Density
VGG13	92%	90%	1.7%
VGG16	93%	91%	5.3%
VGG19	92%	89%	2.4%
AlexNet	86%	82%	2.3%
ResNet18	87%	86%	5.6%
ResNet34	88%	86%	5.2%
Dataset: Cifar-100			
Model	Unpruned accuracy	Pruned Accuracy	Network Density
VGG16	70%	66%	5.3%
ResNet18	55%	54%	5.6%
ResNet34	57%	56%	5.2%

Finally, we compare the performance of the proposed Ramanujan sparse network initialization with other state-of-art pruning at initialization (PaI) techniques. The comparison of accuracy between various pruning at initialization (PaI) techniques at network density $\sim$5% is shown for the VGG16 and ResNet34 architectures in Table 5 and Table 6 on Cifar-10 and Cifar-100 datasets.

Table 5. Performance of various PaI methods on Cifar-10 dataset

Dataset: Cifar-10	
VGG16 (Network Density $\sim$5.3%)	
Method	Accuracy
Unpruned	93%
Our Method	91%
Random	89%
ERK	91%
SynFlow	**92%**
ResNet34 (Network Density $\sim$5.2%)	
Method	Accuracy
Unpruned	88%
Our Method	**86%**
Random	81%
ERK	**86%**
GraSP	**86%**

Table 6. Performance of various PaI methods on Cifar-100 dataset

Dataset: Cifar-100	
VGG16 (Network Density $\sim$5.3%)	
Method	Accuracy
Unpruned	70%
Our Method	**66%**
Random	60%
ERK	62%
SynFlow	65%
ResNet34 (Network Density $\sim$5.2%)	
Method	Accuracy
Unpruned	57%
Our Method	**56%**
Random	50%
ERK	**56%**
GraSP	**56%**

We can observe that our zero-shot method can achieve comparable accuracy to other iterative pruning at initialization techniques. It also significantly outperforms the random mask initialization. The pruned networks still maintain their accuracy with a slight reduction of around $1-2\%$ compared to their unpruned counterparts even at such low remaining weight percentage.

5. Conclusion and Future Work

We presented a deterministic, data independent, zero-shot method for constructing sparse neural network structures which upon weight initialization can be trained to a high accuracy. Experimental results on popular architectures and datasets demonstrate that close to unpruned network accuracy can be achieved using a very sparse network structure.

With the success of sparse deterministic Ramanujan neural networks, our further direction of work is to implement these in the case of transformers and study sparse Ramanujan transformer networks. The proposed deterministic construction technique is expected to significantly reduce the number of parameters and training time while maintaining accuracy.

References

Alon, N. Eigenvalues and expanders. *Combinatorica*, 6 (2):83–96, Jun 1986. ISSN 1439-6912. doi: 10.1007/BF02579166. URL https://doi.org/10.1007/BF02579166.

Alon, N. and Milman, V. D. $\lambda 1$, isoperimetric inequalities for graphs, and superconcentrators. *Journal of Combinatorial Theory, Series B*, 38(1):73–88, 1985.

Arnav Kalra, S., Biswas, A., Mitra, P., and Basu, B. Graph

expansion in pruned recurrent neural network layers preserve performance. In *The Second Tiny Papers Track at ICLR 2024, Tiny Papers @ ICLR 2024,*. OpenReview.net, 2024. URL https://openreview.net/pdf?id=hG5eu7ikDy.

Biswas, A. On a cheeger type inequality in cayley graphs of finite groups. *European Journal of Combinatorics*, 81:298–308, October 2019. doi: 10.1016/j.ejc.2019.06.009. URL https://doi.org/10.1016/j.ejc.2019.06.009.

Biswas, A. and Saha, J. P. A Cheeger type inequality in finite Cayley sum graphs. *Algebraic Combinatorics*, 4(3):517–531, 2021. doi: 10.5802/alco.166. URL https://alco.centre-mersenne.org/articles/10.5802/alco.166/.

Biswas, A. and Saha, J. P. Spectra of twists of cayley and cayley sum graphs. *Advances in Applied Mathematics*, 132:102272, January 2022. doi: 10.1016/j.aam.2021.102272. URL https://doi.org/10.1016/j.aam.2021.102272.

Biswas, A. and Saha, J. P. A spectral bound for vertex-transitive graphs and their spanning subgraphs. *Algebraic Combinatorics*, 6(3):689–706, 2023. doi: 10.5802/alco.278. URL https://alco.centre-mersenne.org/articles/10.5802/alco.278/.

Breuillard, E., Green, B., Guralnick, R., and Tao, T. Expansion in finite simple groups of lie type. *Journal of the European Mathematical Society*, 17(6):1367–1434, 2015. doi: 10.4171/jems/533. URL https://doi.org/10.4171/jems/533.

Burnwal, S. P., Sinha, K., and Vidyasagar, M. New and explicit constructions of unbalanced ramanujan bipartite graphs. *The Ramanujan Journal*, 57(3):1043–1069, April 2021. URL https://doi.org/10.1007/s11139-021-00384-0.

Chen, T., Chen, X., Ma, X., Wang, Y., and Wang, Z. Coarsening the granularity: Towards structurally sparse lottery tickets. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 3025–3039. PMLR, 17–23 Jul 2022.

Chen, Z., Xiang, J., Lu, Y., Xuan, Q., Wang, Z., Chen, G., and Yang, X. Rgp: Neural network pruning through regular graph with edges swapping. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–13, 2023. doi: 10.1109/TNNLS.2023.3280899.

Chung, F. A generalized alon-boppana bound and weak ramanujan graphs. *The Electronic Journal of Combinatorics*, 23(3), July 2016. doi: 10.37236/5933. URL https://doi.org/10.37236/5933.

Dodziuk, J. Difference equations, isoperimetric inequality and transience of certain random walks. *Transactions of the American Mathematical Society*, 284(2):787–794, 1984.

Esguerra, K., Nasir, M., Tang, T. B., Tumian, A., and Ho, E. T. W. Sparsity-aware orthogonal initialization of deep neural networks. *IEEE Access*, 2023.

Evci, U., Gale, T., Menick, J., Castro, P. S., , and Elsen, E. Rigging the lottery: Making all tickets winners. In *ArXiV*, 2020.

Fischer, J. and Burkholz, R. Plant'n'seek: Can you find the winning ticket? In *International Conference on Learning Representations*, 2022.

Frankle, J. and Carbin, M. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *7th International Conference on Learning Representations, ICLR*. OpenReview.net, 2019. URL https://openreview.net/forum?id=rJl-b3RcF7.

Frankle, J., Dziugaite, G. K., Roy, D., and Carbin, M. Pruning neural networks at initialization: Why are we missing the mark? In *International Conference on Learning Representations*, 2020.

Hall, C., Puder, D., and Sawin, W. F. Ramanujan coverings of graphs. *Advances in Mathematics*, 323:367–410, 2018. ISSN 0001-8708. doi: https://doi.org/10.1016/j.aim.2017.10.042. URL https://www.sciencedirect.com/science/article/pii/S0001870817303146.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *IEEE conference on computer vision and pattern recognition*, 2016.

Hoang, D., Liu, S., Marculescu, R., and Wang, Z. Revisiting pruning at initialization through the lens of ramanujan graph. In *International Conference on Learning Representations*, 2023.

Hoory, S., Linial, N., and Wigderson, A. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(4):439–561, 2006.

Kepner, J. and Robinett, R. Radix-net: Structured sparse matrices for deep neural networks. In *2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pp. 268–274, Los Alamitos, CA, USA, may 2019. IEEE Computer

Society. doi: 10.1109/IPDPSW.2019.00051. URL https://doi.ieeecomputersociety.org/10.1109/IPDPSW.2019.00051.

Krizhevsky, A. Learning multiple layers of features from tiny images. In *NEURIPS*, 2009.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *NEURIPS*, 2012.

Laenen, S. One-shot neural network pruning via spectral graph sparsification. In *TAGML Workshop, ICML*, 2023.

Lee, N., Ajanthan, T., Gould, S., and Torr, P. H. A signal propagation perspective for pruning neural networks at initialization. In *International Conference on Learning Representations*, 2019a.

Lee, N., Ajanthan, T., and Torr, P. Single-shot network pruning based on connection sensitivity. In *International Conference on Learning Representations,*, 2019b.

Liu, S., Chen, T., Chen, X., Shen, L., Mocanu, D. C., Wang, Z., , and Pechenizkiy, M. The unreasonable effectiveness of random pruning: Return of the most naive baseline for sparse training. In *International Conference on Learning Representations*, 2022.

Lubotzky, A., Phillips, R., and Sarnak, P. Ramanujan graphs. *Combinatorica*, 8:261–277, 1988.

Mocanu, D. C., Mocanu, E., Stone, P., Nguyen, P. H., Gibescu, M., , and Liotta, A. Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. In *Nature communications*, 2018.

Nilli, A. On the second eigenvalue of a graph. *Discrete Mathematics*, 91(2):207–210, 1991. ISSN 0012-365X. doi: https://doi.org/10.1016/0012-365X(91)90112-F. URL https://www.sciencedirect.com/science/article/pii/0012365X9190112F.

Pal, B., Biswas, A., Kolay, S., Mitra, P., and Basu, B. A study on the ramanujan graph property of winning lottery tickets. In *International Conference on Machine Learning*, pp. 17186–17201, 2022.

Prabhu, A., Varma, G., and Namboodiri, A. Deep expander networks: Efficient deep networks from graph theory. In *Computer Vision – ECCV 2018*, pp. 20–36. Springer International Publishing, 2018a. doi: 10.1007/978-3-030-01261-8_2. URL https://doi.org/10.1007%2F978-3-030-01261-8_2.

Prabhu, A., Varma, G., and Namboodiri, A. Deep expander networks: Efficient deep networks from graph theory. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 20–35, 2018b.

Renda, A., Frankle, J., and Carbin, M. Comparing rewinding and fine-tuning in neural network pruning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1gSj0NKvB.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2014.

Sreenivasan, K., Sohn, J.-y., Yang, L., Grinde, M., Nagle, A., Wang, H., Xing, E., Lee, K., and Papailiopoulos, D. Rare gems: Finding lottery tickets at initialization. In *Advances in Neural Information Processing Systems*, volume 35, pp. 14529–14540, 2022.

Stewart, J., Michieli, U., and Ozay, M. Data-free model pruning at initialization via expanders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4518–4523, 2023.

Tanaka, H., Kunin, D., Yamins, D. L. K., and Ganguli, S. Pruning neural networks without any data by iteratively conserving synaptic flow. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc.

Tartaglione, E. The rise of the lottery heroes: Why zero-shot pruning is hard. In *2022 IEEE International Conference on Image Processing (ICIP)*. IEEE, October 2022. doi: 10.1109/icip46576.2022.9897223. URL http://dx.doi.org/10.1109/ICIP46576.2022.9897223.

Vysogorets, A. and Kempe, J. Connectivity matters: Neural network pruning through the lens of effective sparsity. *J. Mach. Learn. Res.*, 24, 2023.

Wang, C., Zhang, G., and Grosse, R. Picking winning tickets before training by preserving gradient flow. In *International Conference on Learning Representations,*, 2020.

Wang, H., Qin, C., Bai, Y., Zhang, Y., and Fu, Y. Recent advances on neural network pruning at initialization. *arXiv preprint arXiv:2103.06460*, 2021.

Zhou, H., Lan, J., Liu, R., and Yosinski, J. Deconstructing lottery tickets: zeros, signs, and the supermask. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2019. Curran Associates Inc.

A. Appendix

A.1. Cayley graph

Let G be a group and let S be a subset of G that is closed under inversion i.e., $S = S^{-1}$. The corresponding Cayley graph $C(G, S)$ is a graph with vertex set the elements of G and edge set $\{(x, xs) : x \in G,\ s \in S\}$.

The adjacency matrix of a Cayley graph is highly symmetric in the sense that it contains more structure than that of adjacency matrices of general graphs. In fact often for sparse graphs, adjacency lists are stored in the memory. In case of a Cayley graph, one only needs to store the group elements G and the generating set S (of small size since we are focusing on sparse expanders), and the adjacency lists can be computed easily. This structural information is lost in case of random expander based networks.

A.2. Expander graphs and Ramanujan graphs

An expander graph is a structurally sparse graph that has strong connectivity properties. The connectivity can be quantified in different ways which give rise to different notions of expanders such as vertex expanders, edge expanders and spectral expanders. These notions are actually interrelated. In the following, a graph $\Gamma = (V, E)$ is a tuple consisting of a vertex set V and an edge set E which is a subset of $V \times V$.

A.2.1. Combinatorial expansion

Definition A.1 (vertex Cheeger constant)**.** The infimum of the quantity $\frac{|\delta(X)|}{|X|}$ where $\delta(X)$ denotes the outer vertex boundary of X i.e., the set of vertices in Γ which are connected to a vertex in X but do not lie in X as X runs over all non-empty subsets of V satisfying the condition with $|X| \leq \frac{|V|}{2}$ is known as the vertex Cheeger constant and is denoted by $\mathfrak{h}(\Gamma)$.

Definition A.2 (edge-Cheeger constant)**.** The edge boundary of a set S, denoted δS, is $\delta S =$ the set of edges going out from S to its complement. The edge Cheeger constant of Γ, denoted by $\mathbf{h}(\Gamma)$, is defined as: $\mathbf{h}(\Gamma) = \min \frac{|\delta S|}{D|S|}$ as S satisfies the following: $\{S \neq$ empty set, $|S| \leqslant \frac{n}{2}\}$ and D is the maximum degree of the graph Γ

The vertex Cheeger constant $\mathfrak{h}(\Gamma)$ and the edge Cheeger constant $\mathbf{h}(\Gamma)$ are related by the following equivalence

$$\frac{\mathfrak{h}(\Gamma)}{D} \leqslant \mathbf{h}(\Gamma) \leqslant \mathfrak{h}(\Gamma),$$

where D denotes the maximum degree of the graph (the degree of each vertex is the number of edges going out from the vertex). This allows one to speak about vertex expansion and edge expansion interchangeably. Having high combinatorial expansion means having high Cheeger constant, a desirable property for our case.

A.2.2. Spectral expansion

Given a finite undirected graph Γ the eigenvalues $\lambda_n \leqslant \cdots \leqslant \lambda_1$ of its adjacency matrix are all real and $\lambda_1 \leqslant D$ with equality iff the graph is D-regular. The spectra, i.e., the distribution of the eigenvalues convey a lot of information about the structure of the graphs. For instance, the quantity $\lambda_1 - \lambda_2$ (also known in the literature as the one sided spectral gap) quantifies the connectivity and the combinatorial expansion of the graph via the discrete Cheeger-Buser inequality, discovered independently by (Dodziuk, 1984) and by (Alon & Milman, 1985). A graph $\Gamma = (V, E)$ is said to be a spectral expander if the quantities $\{|\lambda_1| - |\lambda_2|, |\lambda_1| - |\lambda_k|\}$ are both bounded away from zero, where $k = n - 1$ if the graph is bipartite and $k = n$ otherwise.

A.2.3. Discrete Cheeger–Buser inequality

The discrete Cheeger–Buser inequality discovered independently by (Dodziuk, 1984) and by (Alon & Milman, 1985) allows one to pass from spectral expansion to combinatorial expansion. The inequality states that

$$\frac{\mathbf{h}(\Gamma)^2}{2} \leqslant \alpha_2 \leqslant 2\mathbf{h}(\Gamma),$$

where α_2 denotes the second smallest eigenvalue of the normalised Laplacian matrix of Γ and is related to the eigenvalues of the adjacency matrix via

$$\frac{\lambda_i}{D} \leqslant 1 - \alpha_i \leqslant \frac{\lambda_i}{d}\ \forall i = 1, 2, \ldots, n.$$

See (Chung, 2016) for details. From the above, it is easy to check that a high $|\lambda_1| - |\lambda_2|$ ensures a high $\mathbf{h}(\Gamma)$ and vice-versa. Thus, the two notions of expansion are inter-connected and every spectral expander remains a combinatorial expander. They are actually equivalent for some classes of graphs, for instance bipartite graphs (as the adjacency spectrum is symmetric about the origin), variants of algebraic graphs e.g., see (Breuillard et al., 2015; Biswas, 2019; Biswas & Saha, 2021; 2022; 2023) etc.

A.2.4. RAMANUJAN GRAPH BOUNDS, ALON-BOPANNA THEOREM

A d-regular graph is said to be a Ramanujan graph if $\max\{|\lambda_2|, |\lambda_k|\} \leqslant 2\sqrt{d-1}$. In the case of bipartite graphs, $\lambda_n = \lambda_1$ and $\lambda_{n-1} = \lambda_2$, hence the previous expression reduces to $|\lambda_2| \leqslant 2\sqrt{d-1}$. For fixed degree, with the sizes of the graphs growing larger and larger, these are the best possible expanders, as given by the Alon-Bopanna bound (Alon, 1986; Nilli, 1991).

Theorem A.3 (Alon-Boppana)**.** *For every d regular graph on n vertices,*

$$\lambda \geq 2\sqrt{d-1} - o_n(1).$$

The $o_n(1)$ term is a quantity that tends to zero for every fixed d as $n \to \infty$.

A.3. Experimental methodology

The q and l values used by the Ramanujan Graph construction for the convolution layers for various architectures is provided in Table 7.

Table 7. Values of q and l to generate Ramanujan graphs for layers of VGG, AlexNet and ResNet

VGG13			VGG19		
Conv Size	q	l	Conv Size	q	l
$[256 \times 256 \times 3 \times 3] \times 1$	13	177	$[256 \times 256 \times 3 \times 3] \times 3$	13	177
$[512 \times 256 \times 3 \times 3] \times 1$	19	121	$[512 \times 256 \times 3 \times 3] \times 1$	19	121
$[512 \times 512 \times 3 \times 3] \times 3$	19	242	$[512 \times 512 \times 3 \times 3] \times 7$	19	242
Conv to Linear Size	q	l	Conv to Linear Size	q	l
2448×25088	47	533	2448×25088	47	533
AlexNet			ResNet18		
Conv Size	q	l	Conv Size	q	l
$[384 \times 256 \times 3 \times 3] \times 1$	19	121	$[64 \times 64 \times 3 \times 3] \times 4$	7	82
$[384 \times 384 \times 3 \times 3] \times 1$	19	181	$[128 \times 64 \times 3 \times 3] \times 1$	11	52
$[256 \times 384 \times 3 \times 3] \times 1$	13	265	$[128 \times 128 \times 3 \times 3] \times 3$	11	104
			$[256 \times 128 \times 3 \times 3] \times 1$	13	88
			$[256 \times 256 \times 3 \times 3] \times 3$	13	177
			$[512 \times 256 \times 3 \times 3] \times 1$	19	121
			$[512 \times 512 \times 3 \times 3] \times 3$	19	242
Conv to Linear Size	q	l			
2448×25088	47	533			