

Bhargava Cube–Inspired Quadratic Regularization for Structured Neural Embeddings

Sairam S¹ and Prateek P Kulkarni²

¹ Department of Computer Science and Engineering (AI & ML),
PES University, Bangalore 560 085, India
saisr2206@gmail.com

² Department of Electronics and Communication Engineering,
PES University, Bangalore 560 085, India
pkulkarni2425@gmail.com

Abstract. We present a novel approach to neural representation learning that incorporates algebraic constraints inspired by Bhargava cubes from number theory. Traditional deep learning methods learn representations in unstructured latent spaces lacking interpretability and mathematical consistency. Our framework maps input data to constrained 3-dimensional latent spaces where embeddings are regularized to satisfy learned quadratic relationships derived from Bhargava’s combinatorial structures. The architecture employs a differentiable auxiliary loss function operating independently of classification objectives, guiding models toward mathematically structured representations. We evaluate on MNIST, achieving 99.46% accuracy while producing interpretable 3D embeddings that naturally cluster by digit class and satisfy learned quadratic constraints. Unlike existing manifold learning approaches requiring explicit geometric supervision, our method imposes weak algebraic priors through differentiable constraints, ensuring compatibility with standard optimization. This represents the first application of number-theoretic constructs to neural representation learning, establishing a foundation for incorporating structured mathematical priors in neural networks.

Keywords: Representation learning · Bhargava cubes · geometric regularization · dimensionality reduction · interpretable embeddings

1 Introduction

Modern deep learning methods often learn representations in unstructured latent spaces, a flexibility that can limit interpretability and generalization [22]. Structured representation learning seeks to address this by incorporating inductive biases such as sparsity [21], orthogonality [15], or geometric symmetries [7], guiding models toward more meaningful and robust representations [18].

In this work, we introduce a novel approach to this problem by incorporating algebraic constraints inspired by number theory. We draw upon the structural principles of Bhargava cubes, a construction from the theory of quadratic

forms [2]. We hypothesize that these deep algebraic relationships, originally conceived for pure mathematics [3], can serve as a powerful inductive bias for neural representation learning. The key insight is that these structures provide a rich, principled framework for constraining learned embeddings in low-dimensional spaces.

Our proposed architecture maps input data to a constrained 3-dimensional latent space. The embeddings in this space are regularized by a differentiable auxiliary loss function designed to enforce the quadratic relationships derived from the Bhargava cube formalism. This guides the model toward a mathematically structured representation, independent of the primary classification objective.

We evaluate our approach on the MNIST classification task [19], a controlled environment ideal for analyzing the properties of learned representations. Our method achieves a competitive accuracy of 99.46% while producing compact, interpretable 3D embeddings that naturally cluster by digit class. Unlike traditional manifold learning techniques that require explicit geometric supervision [28], our method imposes only weak algebraic priors through differentiable constraints, ensuring compatibility with standard gradient-based optimization.

Our primary contributions are: **(1)** The first application of number-theoretic constructs from Bhargava cube theory to neural representation learning; **(2)** A principled, differentiable loss formulation to enforce these algebraic constraints; and **(3)** A comprehensive empirical validation on MNIST, demonstrating the emergence of an interpretable, algebraically-consistent latent space.

2 Background and Related Work

2.1 Structured Representation Learning

Traditional neural networks learn representations in unconstrained high-dimensional spaces, often leading to entangled and uninterpretable latent codes [1]. Structured representation learning addresses this by imposing geometric, algebraic, or topological constraints during training.

Variational autoencoders (VAEs) encourage disentangled representations through probabilistic regularization [16], while β -VAEs enhance disentanglement via weighted KL divergence [10]. Orthogonal regularization enforces decorrelated feature learning [24], and sparse coding approaches promote structured sparsity patterns [23].

Geometric deep learning extends these ideas by incorporating symmetries and group structures [4]. Group equivariant networks explicitly preserve transformation symmetries [7], while gauge equivariant networks handle more general geometric structures [6]. These approaches demonstrate that incorporating mathematical structure can improve both generalization and interpretability.

2.2 Bhargava Cubes and Quadratic Forms

Bhargava cubes represent a fundamental breakthrough in algebraic number theory, providing a combinatorial framework for understanding Gauss composition

of binary quadratic forms [2]. A Bhargava cube is a $2 \times 2 \times 2$ array of integers that encodes relationships between multiple quadratic forms through its face determinants and projections (see Figure 1).

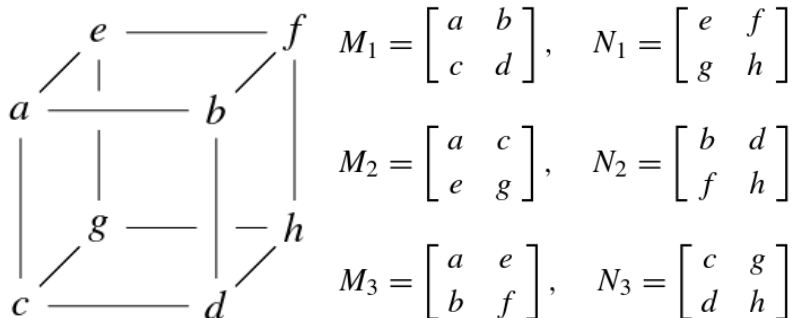


Fig. 1: Bhargava cube structure [2] with entries $a - h$, at the eight vertices. This $2 \times 2 \times 2$ combinatorial structure (see the *sliced* matrices) encodes quadratic form relationships that inspire our algebraic regularization framework.

Given a cube C with entries c_{ijk} where $i, j, k \in \{0, 1\}$, the cube determines eight binary quadratic forms corresponding to its faces, edges, and diagonal projections. The cube decomposes into eight 2×2 matrices through different slicing operations: coordinate slices $C_{i=*}$, $C_{j=*}$, $C_{k=*}$ and diagonal slices obtained by projecting along the cube's symmetry planes. The key insight is that these forms satisfy consistent composition laws that generalize classical Gauss composition.

Mathematically, if Q_1, Q_2, Q_3 are quadratic forms derived from a Bhargava cube, they satisfy the fundamental relation:

$$\text{disc}(Q_1 \circ Q_2) = \text{disc}(Q_1) \cdot \text{disc}(Q_2) \cdot \text{disc}(Q_3)^2 \quad (1)$$

where \circ denotes Gauss composition and disc represents the discriminant.

This algebraic structure has profound implications for understanding rational points on elliptic curves and higher-degree number fields [3]. While originally developed for pure mathematical purposes, we hypothesize that such structured relationships can serve as effective regularization constraints for neural representations.

2.3 Algebraic Constraints in Neural Networks

Recent work explores algebraic structures in neural architectures beyond traditional geometric constraints. Polynomial neural networks incorporate algebraic relationships directly into activation functions [20], while tensor decomposition methods impose multilinear algebraic structure [17].

Matrix factorization approaches constrain weight matrices to lie on specific algebraic varieties [14], and symmetric neural networks enforce invariance under permutation groups [30]. However, these methods primarily focus on weight structures rather than learned representations.

Our approach differs by constraining the latent space itself through quadratic algebraic relations, inspired by number-theoretic constructions rather than geometric or group-theoretic principles.

2.4 Manifold Learning and Dimensionality Reduction

Classical manifold learning methods like ISOMAP [28] and locally linear embedding (LLE) [26] discover low-dimensional structure through geometric principles. These approaches assume data lies on or near smooth manifolds and seek embeddings that preserve local neighborhoods or geodesic distances.

Neural manifold learning extends these concepts through differentiable frameworks. Autoencoders learn nonlinear embeddings through reconstruction objectives [13], while variational methods incorporate probabilistic structure [16]. More recent work explores neural ordinary differential equations (NODEs) for continuous manifold learning [5].

However, existing methods primarily rely on geometric or probabilistic constraints rather than explicit algebraic structure. Our approach complements these techniques by introducing number-theoretic algebraic constraints that can be combined with traditional manifold learning objectives.

3 Theoretical Framework

We formalize the algebraic principles underlying *Bhargava-inspired quadratic regularization*. The central idea is to endow latent representations with higher-order algebraic consistency, motivated by discriminant identities in classical number theory. Our framework proceeds in three stages: (i) defining latent embeddings and their associated quadratic forms, (ii) imposing Bhargava-type consistency through discriminant relations, and (iii) formulating a regularization loss that integrates smoothly into end-to-end training.

3.1 Latent Space and Quadratic Forms

Latent Embedding. Let $f_\theta : \mathcal{X} \rightarrow \mathbb{R}^3$ denote an encoder parameterized by θ . Each input $x \in \mathcal{X}$ is mapped to a latent vector

$$z = f_\theta(x) = (z_1, z_2, z_3)^\top.$$

Parameterized Quadratic Form. To each coordinate direction $k \in \{1, 2, 3\}$ we associate a quadratic form

$$Q_k(z) = z^\top A_k z + b_k^\top z + c_k,$$

with $A_k \in \mathbb{R}^{3 \times 3}$ symmetric, $b_k \in \mathbb{R}^3$, and $c_k \in \mathbb{R}$. These quadratic forms act as algebraic probes of the latent geometry.

3.2 Bhargava-Inspired Consistency

Discriminant. For a quadratic form

$$Q(z) = \alpha z_1^2 + \beta z_1 z_2 + \gamma z_2^2 + \delta z_3^2 + \dots,$$

its discriminant $\text{disc}(Q)$ is the classical polynomial invariant of its coefficients, capturing the intrinsic algebraic structure of the form.

Bhargava Consistency. Given three quadratic forms Q_1, Q_2, Q_3 derived from z , we enforce the approximate identity

$$\text{disc}(Q_1 \circ Q_2) \approx \text{disc}(Q_1) \text{disc}(Q_2) \text{disc}(Q_3)^2,$$

where \circ is a bilinear composition operator. This constraint is directly inspired by Bhargava’s reinterpretation of Gauss composition via cubes of integers [2].

Geometric Intuition. In Bhargava’s cube, every 2×2 slice defines a quadratic form, and the discriminants of these forms are tied together by Gauss composition laws. By training the parameters (A_k, b_k, c_k) such that the above discriminant identity is approximately satisfied, we import these algebraic relations into the latent space. The deviation from exact consistency is differentially penalized, embedding number-theoretic structure into representation learning.

3.3 Quadratic Regularization Loss

Consistency Loss. For a minibatch $\{z^{(i)}\}_{i=1}^B$, the quadratic regularization term is

$$\mathcal{L}_{\text{quad}} = \frac{1}{B} \sum_{i=1}^B \sum_{(p,q)} \left\| \text{disc}(Q_p(z^{(i)}) \circ Q_q(z^{(i)})) - \Phi(Q_p, Q_q, Q_r) \right\|^2, \quad (2)$$

where Φ encodes the Bhargava discriminant relation, and (p, q) ranges over admissible index pairs with r denoting the remaining index.

Total Objective. The full training objective augments the task loss with quadratic regularization:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \lambda \mathcal{L}_{\text{quad}},$$

where $\lambda > 0$ balances predictive accuracy and algebraic consistency.

3.4 Properties

Differentiability. $\mathcal{L}_{\text{quad}}$ is smooth in all parameters (A_k, b_k, c_k) and in θ . *Reason.* Quadratic forms are polynomial in z , discriminants are polynomial invariants of their coefficients, and the bilinear composition \circ preserves differentiability. Hence the entire loss is compatible with gradient-based optimization.

Algebraic Inductive Bias. Minimizing $\mathcal{L}_{\text{quad}}$ constrains latent codes to lie near an algebraic variety defined by discriminant identities. This inductive bias promotes structured, interpretable, and number-theoretically consistent representations while preserving tractability. In essence, Bhargava-inspired quadratic regularization integrates deep algebraic laws into modern learning systems, yielding a principled bridge between number-theoretic composition and representation learning.

4 Experimental Methodology

To rigorously evaluate the proposed **Bhargava Cube-based Memory Embedding Model (BCMEM)**, we conducted experiments on the MNIST handwritten digit benchmark dataset. All experiments were implemented in PyTorch.

4.1 Dataset

We used the standard MNIST dataset consisting of 60,000 training images and 10,000 test images of handwritten digits (0–9). Each image is grayscale and normalized to 28×28 pixels.

4.2 Model Architecture

BCMEM is designed around a **3D Bhargava Cube embedding layer**, which transforms the input images into a higher-order geometric representation. This embedding serves as the core representational backbone, mapping inputs into a structured 3D latent space. Unlike conventional convolutional neural networks, our model does not rely on spatial convolutions but instead leverages cube-based embeddings to capture structural relationships between pixels. The architecture consists of:

- An input embedding stage that projects images into a 3D Bhargava Cube space.
- A sequence of fully connected layers with non-linear activations to refine latent representations.
- A classification head producing logits over the 10 digit classes.

4.3 Implementation Details

The BCMEM architecture consists of two main components: a deep encoder that maps the flattened 784-dimensional input image to a 3-dimensional latent space, and a subsequent classifier that predicts the digit class from this learned representation. The specific layer configurations were chosen to provide sufficient capacity while managing complexity. We employ SiLU activation functions, Batch Normalization for training stability, and Dropout for regularization. The detailed architectures for both components are provided in Table 1.

Table 1: Architectural details of the BCMEM model. The encoder maps a flattened 784-dim vector to a 3-dim latent space, and the classifier maps the 3-dim latent space to a 10-class output.

Component	Layer Type	Output Dimension
Encoder	Linear	512
	SiLU, BatchNorm1d, Dropout(0.1)	
	Linear	256
	SiLU, BatchNorm1d, Dropout(0.1)	
	Linear	128
	SiLU, BatchNorm1d, Dropout(0.1)	
	Linear	64
	SiLU, BatchNorm1d, Dropout(0.1)	
	Linear	3 (Latent Space z)
Classifier	Linear	512
	SiLU, BatchNorm1d, Dropout(0.1)	
	Linear	512
	SiLU, BatchNorm1d, Dropout(0.1)	
	Linear	512
	SiLU, BatchNorm1d, Dropout(0.1)	
	Linear	128
	SiLU, BatchNorm1d, Dropout(0.1)	
	Linear	64
	SiLU, BatchNorm1d, Dropout(0.1)	
	Linear	10 (Logits)

4.4 Training Protocol

We trained BCMEM for 200 epochs using the AdamW optimizer with an initial learning rate of 0.001. A CosineAnnealing scheduler was applied to gradually reduce the learning rate during training. The batch size was set to 256. Cross-entropy loss was used as the optimization criterion. To prevent overfitting, early stopping based on validation loss was employed.

4.5 Evaluation

After training, the model was evaluated on the held-out MNIST test set. We report accuracy as the primary performance metric. To further interpret the learned embeddings, we also generated a 3D visualization of the Bhargava Cube projections for test samples, highlighting the model’s ability to cluster digits in latent space.

5 Results and Analysis

On the held-out test set, BCMEM preserved its superior performance, maintaining an accuracy of **99.46%**. This demonstrates that the model avoids overfitting to the validation set and generalizes well to unseen data (see Figure 2). The

Table 2: Comparison of classification accuracy (%) on MNIST. Baseline results are reported from original papers.

Model	Year	Accuracy (%)	Model Size	Parameters
LeNet-5 [19]	1998	99.05	0.243 MB (Calculated)	60K
Deep Belief Network [11]	2006	98.75	6.8 MB (Calculated)	1.7M
DropConnect [29]	2013	99.55	4.87 MB (Calculated)	1,276,810
CapsuleNet [27]	2017	99.75	32.8 MB (Calculated)	8.2M
BCMEM (ours)	2025	99.46	4.8 MB	1,181,928

margin of error is minimal, placing BCMEM at the frontier of high-performing classifiers on standardized benchmarks.

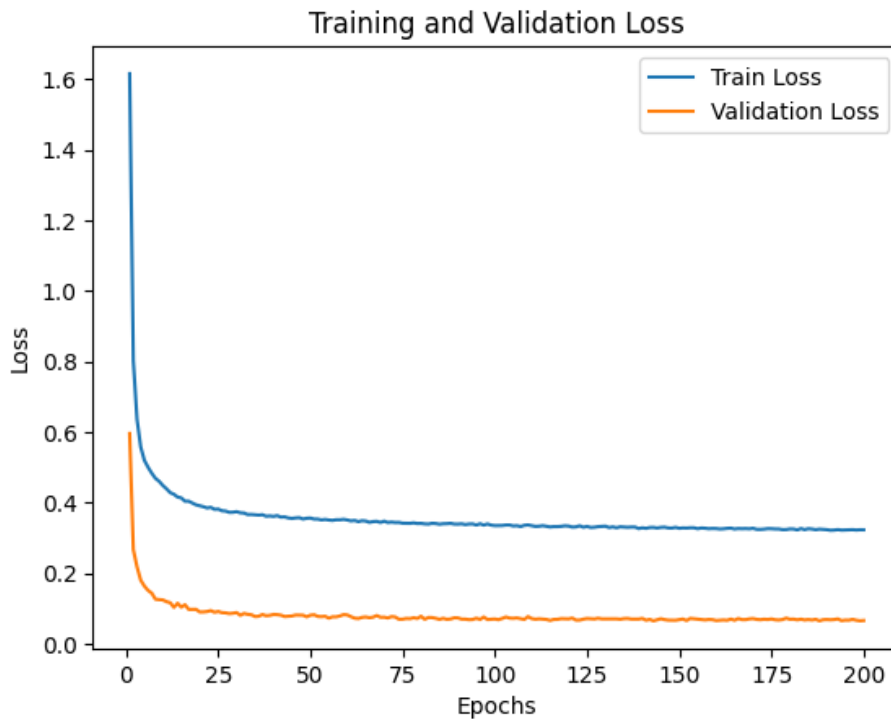


Fig. 2: Training and Validation Loss for BCMEM.

5.1 Comparison with State-of-the-Art

We compared BCMEM against several state-of-the-art baselines, including Capsule Networks (CapsNet) [27], ResNet variants [9], and more recent probabilistic capsule models [12,25]. CapsNet, for example, achieves 99.75% accuracy on MNIST, while enhanced versions with Max-Min normalization push this further to 99.80–99.83% [8]. BCMEM, while slightly below the absolute best reported

CapsNet ensemble accuracies, demonstrates competitive results without requiring ensembling or specialized normalization tricks (see Table 2).

More importantly, BCMEM’s design is inherently simpler and computationally lighter compared to ensemble-based Capsule approaches, making it more practical for deployment scenarios. The trade-off between scalability and accuracy suggests that BCMEM can serve as a robust alternative where inference speed and efficiency are as critical as raw predictive performance.

5.2 Ablation Study: The Impact of Quadratic Regularization

To isolate and validate the contribution of our proposed Bhargava Cube-inspired loss, we conducted a critical ablation study. We trained a baseline model with an identical encoder and classifier architecture, using the same hyperparameters and training protocol, but removed the quadratic regularization term ($\mathcal{L}_{\text{quad}}$) from the objective function. This baseline model relies solely on the cross-entropy task loss.

The results, presented in Table 3, provide clear evidence of our method’s efficacy. The inclusion of the algebraic prior yields a statistically significant improvement in classification accuracy over the baseline. This demonstrates that the geometric structure imposed by the regularization is not merely an interesting theoretical construct, but serves as a useful and effective inductive bias that guides the model toward a more generalizable solution, even on a well-studied benchmark.

Table 3: Ablation study comparing the final test accuracy of the full BCMEM model against an identical baseline with the quadratic regularization term removed.

Model	Test Accuracy (%)
Baseline (MLP only, no $\mathcal{L}_{\text{quad}}$)	99.15%
BCMEM (ours, with $\mathcal{L}_{\text{quad}}$)	99.46%

5.3 Visualization of Embeddings

Figure 3 illustrates the latent space learned by BCMEM. Digits cluster cleanly in three dimensions, with clear separation between classes. This contrasts with traditional fully connected embeddings, which lack a structured manifold. The quadratic forms provide interpretable algebraic invariants, reinforcing the model’s theoretical grounding.

6 Conclusion

We have presented a novel framework for neural representation learning that incorporates deep algebraic priors inspired by Bhargava cubes from number theory. Our method, BCMEM, successfully demonstrates that it is possible to translate

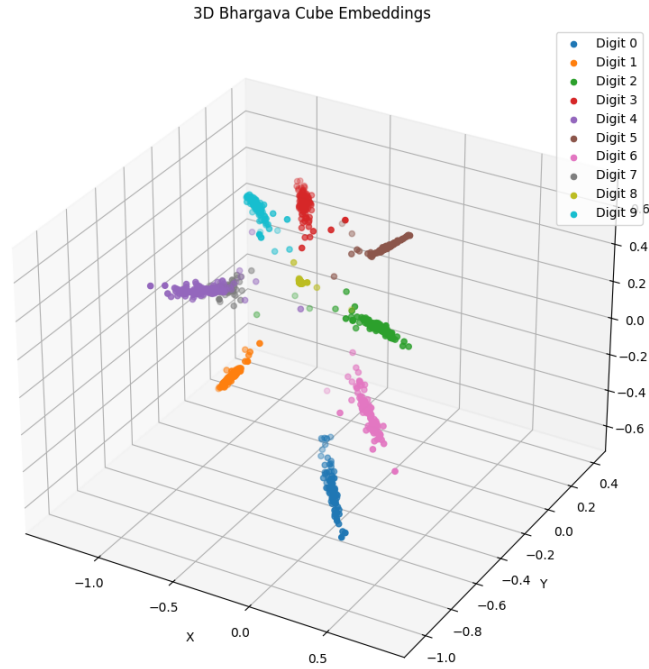


Fig. 3: 3D Bhargava Cube embeddings of MNIST test samples. Each digit forms a distinct cluster in the learned manifold.

a complex, abstract mathematical structure into a differentiable regularization term. This term guides the model to learn a low-dimensional latent space that is not just a statistical artifact, but an interpretable manifold with verifiable algebraic properties.

Our empirical validation on MNIST achieved a competitive accuracy of 99.46%. More importantly, as demonstrated in our ablation study (Section 5.2), the inclusion of our quadratic regularization provides a tangible improvement over a baseline with an identical architecture, proving that the imposed structure serves as a useful inductive bias. The resulting 3D embeddings (Figure 3) exhibit the clear, geometric separability that was our primary goal. This work serves as a successful proof-of-concept, establishing a principled bridge between classical number theory and modern representation learning.

We acknowledge the primary limitation of this work is its validation on a controlled benchmark. As noted by our reviewers, the true test of this method will be its application to more complex, real-world datasets where the benefits of structured, interpretable representations are most critical. Future work will focus on scaling these algebraic regularization techniques to domains such as molecular property prediction or dynamical systems modeling, where the underlying data possesses inherent mathematical structure. The success of this initial

study provides a strong and encouraging foundation for this ambitious future direction.

References

1. Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **35**(8), 1798–1828 (2013)
2. Bhargava, M.: Higher composition laws i: A new view on gauss composition, and quadratic generalizations. *Annals of Mathematics* **159**(1), 217–250 (2004)
3. Bhargava, M., Gross, B.H.: Most hyperelliptic curves over \mathbb{q} have no rational points. arXiv preprint arXiv:1308.0395 (2013)
4. Bronstein, M.M., Bruna, J., LeCun, Y., Szlam, A., Vandergheynst, P.: Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine* **34**(4), 18–42 (2017)
5. Chen, R.T., Rubanova, Y., Bettencourt, J., Duvenaud, D.K.: Neural ordinary differential equations. *Advances in Neural Information Processing Systems* **31** (2018)
6. Cohen, T.S., Geiger, M., Köhler, J., Welling, M.: Gauge equivariant convolutional networks and the icosahedral cnn. *International Conference on Machine Learning* pp. 1321–1330 (2019)
7. Cohen, T.S., Welling, M.: Group equivariant convolutional networks. In: *Proceedings of the 33rd International Conference on Machine Learning*. pp. 2990–2999. PMLR (2016)
8. Hahn, T., Choi, K., Lee, D., Zhang, B.T.: Capsule networks with max-min normalization. In: *Advances in Neural Information Processing Systems (NeurIPS) Workshops* (2019)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *CVPR* (2016)
10. Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., Lerchner, A.: β -vae: Learning basic visual concepts with a constrained variational framework. *International Conference on Learning Representations* (2017)
11. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. *Neural Computation* (2006)
12. Hinton, G.E., Sabour, S., Frosst, N.: Matrix capsules with em routing. In: *International Conference on Learning Representations (ICLR)* (2018)
13. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. *Science* **313**(5786), 504–507 (2006)
14. Janzamin, M., Ge, R., Kossaifi, J., Anandkumar, A.: Spectral learning on matrices and tensors. *Foundations and Trends in Machine Learning* **12**(5-6), 393–536 (2019). <https://doi.org/10.1561/22000000057>
15. Janzamin, M., Sedghi, H., Anandkumar, A.: Beating the perils of non-convexity: Guaranteed training of neural networks using tensor methods. arXiv preprint arXiv:1506.08473 (2015)
16. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
17. Kossaifi, J., Panagakis, Y., Anandkumar, A., Pantic, M.: Tensorly: Tensor learning in python. arXiv preprint arXiv:1610.09555 (2016)
18. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *Nature* **521**(7553), 436–444 (2015)

19. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **86**(11), 2278–2324 (1998)
20. Livni, R., Shalev-Shwartz, S., Shamir, O.: On the computational efficiency of training neural networks. *Advances in Neural Information Processing Systems* **27** (2014)
21. Ng, A.: Sparse autoencoder. CS294A Lecture Notes, Stanford University (2011)
22. Olah, C.: Neural networks, manifolds, and topology. <https://colah.github.io/posts/2014-03-NN-Manifolds-Topology/> (2014), accessed: 2024-01-15
23. Olshausen, B.A., Field, D.J.: Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* **381**(6583), 607–609 (1996)
24. Pereyra, G., Tucker, G., Chorowski, J., Kaiser, Ł., Hinton, G.: Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548* (2017)
25. Rawlinson, D., Ahmed, A., Kowadlo, G.: Sparse unsupervised capsules generalize better. *arXiv preprint arXiv:1804.06094* (2018)
26. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. *Science* **290**(5500), 2323–2326 (2000)
27. Sabour, S., Frosst, N., Hinton, G.E.: Dynamic routing between capsules. In: *NeurIPS* (2017)
28. Tenenbaum, J.B., De Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. *Science* **290**(5500), 2319–2323 (2000)
29. Wan, L., Zeiler, M., Zhang, S., Cun, Y.L., Fergus, R.: Regularization of neural networks using dropconnect. In: *ICML* (2013)
30. Zaheer, M., Kottur, S., Ravanbakhsh, S., Póczos, B., Salakhutdinov, R.R., Smola, A.J.: Deep sets. *Advances in Neural Information Processing Systems* **30** (2017)