What it means by learning in a neural network: easing the knot

Atiya Ibnat Tasnim ATIYA.TASNIM@NORTHSOUTH.EDU
Rushmila Shehreen Khan RUSHMILA.KHAN.232@NORTHSOUTH.EDU
Md. Shahriar Karim SHAHRIAR.KARIM@NORTHSOUTH.EDU
North South University, Dhaka, Bangladesh

Abstract

What constitutes learning in humans and machines remains puzzling despite the unprecedented growth witnessed in both. Starting with a Perceptron and, in subsequent interrogation of multilayer perceptron (MLP), this paper revisits learning and searches for new learning signatures in artificial neural networks (ANN). Precisely, we consider the transport of the initial weight distribution to its final form while optimally balancing information entropy and the statistical complexity. Projecting the learning dynamics on the complexity-entropy plane helps identify optimal information organization internally within the network. As observed, training neural networks guided by complexity-entropy trade-off improves model performance and reproducibility. In continuation, we further assess depth dependence and information flow using entropy differences, the KL-divergence of the weight distributions between successive layers, and mutual information (MI) between the input, hidden layers, and output. Our findings indicate dependency between learning rate, neural network depth, and mutual information. Insights obtained so far in our ongoing analysis are relevant to applications ranging from explainable AI to understanding brain function.

1. Introduction

The Perceptron [13, 18], an artificial mimic of the biological neuron, drawing its existence and theoretical underpinnings from how neurons enable learning in the brain [2], sparked the era of Artificial Intelligence (AI) and applications. Learning in a neural network (NN) involves adjusting its weight distribution through data-intensive training. Despite striking dissimilarities on how a biological neuron and a Perceptron function [5], the gradients remain inseparable from the plasticity of biological neurons and the optimization sought in gradient descent in a perceptron-based network. In addition, a trained NN must generalize well [1, 7], be reproducible, and frugal in training cost and inference. Notably, the strength and structure of neural connectivity get tuned by the distributional shift of weight ($\mathbf{w}$) strength, forming a signature of learning. Many counterintuitive objectives align, apparently inexplicably, defining the persistent black-box nature of state-of-the-art deep learning models [19, 21]. Ultimately, easing the riddle of learning in neural networks is the primary focus of this work, from a single perceptron to an MLP, through the lens of information-theoretic concepts and metrics [21, 23, 25], and statistical complexity [3, 12] (Fig. 1).

Earlier works treated weight distribution tuning as an optimal transport problem [27]. But how the structural correlation between components within a model shapes the learning, or whether it contributes to the NNs' performance, in terms of generalizability, reproducibility, or computational efficiency, remains indecisive. Additionally, the weight distribution in neural networks remains inexplicably different from the experimentally observed log-normal distribution of weights of synaptic plasticity in the brain [16]. This inquires the pertinence of Euclidean geometry in analyzing

© A.I. Tasnim, R.S. Khan & M.S. Karim.

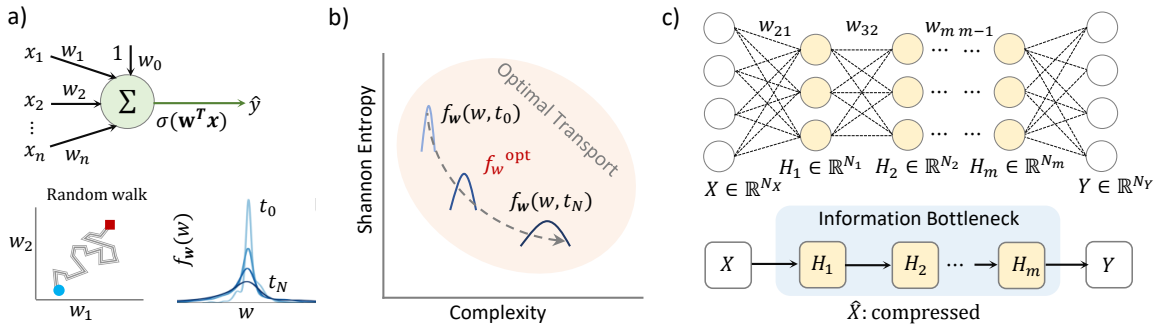

Figure 1: a) n-input perceptron (upper panel). An example random walk in the weight space $w_i \in \mathbb{R}^2$ (lower panel), and weight distribution $f_{\mathbf{w}}(w)$ transport as the training progresses (see Appendix, Fig. 7). b) The proposed 2D Complexity-Entropy (C-E) plane to oversee the optimal transport $f_{\mathbf{w}}(w)$ in learning. c) A sample MLP, amenable to deep linear network (DLN) as well, with notion of mutual information $I(X; H_i)$ flow as in Information Bottleneck (IB) method.

the synaptic plasticity in the brain and points to the possibility of non-Euclidean geometry playing a crucial role in NN being trained on non-Euclidean data [15]. This approach also, like the others [6, 14, 17], does not explain how learning shapes up in NNs and whether it is possible to analytically guide learning by structurally correlating the model components shaping the features being learnt through intensive training.

Recent evidence from neuroscience experiments suggests that brain connectomes are largely function-oriented [10, 20], attesting to the advantageous role of structural connectivity. For instance, the log-normal distribution of synaptic weights provides computational benefits by imposing a structural constraint [27]. Neuronal connectivity also has crucial role in its ability to learn, and in fact contributes to brains' process of generative computing through neuronal chaotic dynamics [8, 9, 11]. Motivated by the insights of the role of internal connectivity, we apply statistical complexity, $\mathrm{C}(\mathbf{w})$, to assess the structural correlation between system components, and interrogate the attainment of the final weight distribution on a 2D plane, namely the complexity-entropy (C-E) plane. In this work, we explore learning dynamics, myths, and beyond. Specifically, we interrogate a single-layer perceptron (SLP) and a multilayer perceptron (MLP) in search of the meaning of learning, that balances, measured on the C-E plane, between order and disorder in a neural network [12, 26]. To summarize, this continuing work seeks an optimal balance among the trifecta of order, information, and equilibrium in neural network learning. Specific outcomes so far obtained are-

- KL-divergence and entropy difference in successive layers reflects training error reduction.
- Training less, or indefinitely long enough, drives NN model to the C-E extrema, suggesting that NN efficiency may relate to optimality between the complexity (C) and Shannon entropy.
- Weight distribution that satisfies the optimality of C-E improves models' reproducibility while maintaining competitive classification accuracy.

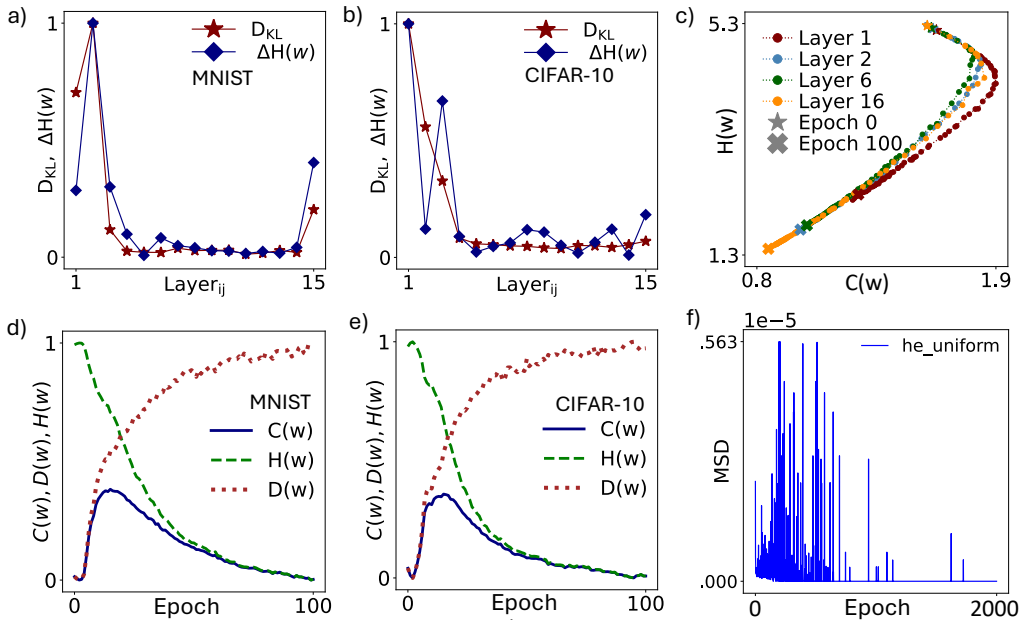

Figure 2: Comparison of Entropy and KL Divergence ($\Delta H(\mathbf{w})$ and $D_{KL}(\mathbf{w}_i||\mathbf{w}_{i+1})$): a) For MNIST trained on 16-layer MLP, b) Same configuration but for CIFAR-10. c) Beyond Layer 6, final epoch drives the system to non-optimal region of C-E plane. Entropy $\Delta H(\mathbf{w})$, Complexity $C(\mathbf{w})$ and Disequilibrium $D(\mathbf{w})$ of weights for MNIST (d) and CIFAR-10 (e) over training epochs. f) MSD progression of Perceptron over 2000 epochs

2. Model and Methods

2.1. Statistical Complexity: Complexity-Entropy (C-E) Plane

Statistical complexity, denoted as C($\mathbf{w}$), a measure of complexity for distribution of $\mathbf{w}$ describing a system, finds an analog in statistical mechanics [3, 12]. The stored information in the system is quantified using Shannon Entropy as $H(\mathbf{w}) = -\sum_{i=1}^{N} p_i \log p_i$. In crystal a specific state $\mathbf{w_c}$ is highly probable, so $H_c \approx 0$. The distance D, of distribution of $\mathbf{w}$ from a uniform distribution, is the disequilibrium evaluated as Jensen–Shannon divergence (JSD)

$$
\begin{aligned}
&D \triangleq \text{JSD, Assuming } f_{\mathbf{w}}(w) = P, \text{ and Uniform}(\mathbf{w}) = Q \\
&\text{JSD}(P||Q) = \frac{1}{2}D_{KL}(P||R) + \frac{1}{2}D_{KL}(Q||R), \text{ where } R = \frac{P+Q}{2} \\
&D_{KL}(A||B) = \sum_i A(i) \log \frac{A(i)}{B(i)}
\end{aligned} \tag{1}
$$

The well-defined order of crystal atoms produce a very sharp distribution of accessible states, resulting in low information storage. In contrast, the equiprobable accessible states of molecules in an ideal gas lead to maximum possible stored information in the system. The ideal-gas and crystal are analogous to fully-connected, highly-sparse networks, providing an avenue for navigating an optimal balance between the structure and information content of a system. In the context of NN,

the weight distribution $\mathbf{w}$ evolves as training proceeds, thereby affecting both the Shannon entropy $H(\mathbf{w})$ and the disequilibrium D, which measures the distance of $\mathbf{w}$ from its uniform distribution. So, learning by a NN is not only about accounting for the entropy of $\mathbf{w}$, but also about an increasing complexity $\mathrm{C}(\mathbf{w})$ indicating how efficient and accurate the NN is at organizing the information. Statistical complexity is defined as the product of KL divergence and JSD as follows:

$$C(\mathbf{w}) = H(\mathbf{w}) \times JSD(\mathbf{w}||\text{Uniform}(\mathbf{w})) \tag{2}$$

2.2. Information Bottleneck

Information bottleneck (IB) principle [24] facilitates extracting the relevant information contained in an input random variable (RV) X about the output RV Y. The impact of the internal organization and architecture of a neural network is shown to be of concern in studying information bottleneck tradeoffs [22]. The mutual information (MI), denoted as I(X; Y), between layers, is defined as

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \tag{3}$$

where $p(x,y)$ is the joint distribution of X, Y and $p(x), p(y)$ are respective marginal distributions. We apply IB framework to study a multi-layer perceptron (MLP), with varied layer sizes $(N = 2, 4, 8, 16)$ and learning rates $(\eta = 0.001, 0.0001, 0.00001)$. In MI calculation, the values are discretized by binning, as used in [4]. The hidden layers (T_i) satisfy the following relationship:

$$I(X;T_1) \geq I(X;T_2) \geq I(X,T_3) \geq \cdots \geq I(X;T_N) \tag{4}$$

3. Results

3.1. Entropy flow and KL-divergence across layer are qualitatively similar to error reduction

Entropy difference, $\Delta H(\mathbf{w})$, between layer L_i and L_{i+1} captures the difference in information between the layers. The difference, as in Fig. 2(a-b), gradually approaches its minimized level as the network goes deeper. It reassures that deepening an NN beyond a limit doesn't introduce much additional information, and holds for both MNIST (Fig. 2a) and CIFAR-10 (Fig. 2b). Specifically, around layer 6-7, both $(\Delta H(\mathbf{w})$ and $D_{KL}(\mathbf{w_i}||\mathbf{w}_{i+1})$ become sufficiently low. To further analyze, we project per-layer training response over all epochs on a Complexity-Entropy (C-E) plane that shows a gradual progression towards a peak complexity level and subsequent falling off to a low complexity region (see Fig. 3). Interestingly, rise-and-fall of complexity curve appears qualitatively preserved across depth variation of NNs (see Appendix, Fig. 8c). Complexity increases in early epochs and reverses its progression near epoch regions (25-50) that resembles the drastic reduction in training error (see inset of Fig. 3(a-b), left panel). Such trend of model complexity remains similar regardless of layer-wise (Fig. 3(a-b), right panel) or per-NN analysis (Fig. 3(a-b), left panel), as well as over alternative datasets (MNIST, Fig. 3a and CIFAR, Fig. 3b). An optimal region, as theorized in [12], lies near the projections of the training dynamics from epochs 25 to 50, thereby strengthening both complexity and entropy on the C-E plane. Projecting learning dynamics onto the C-E plane delineates a probable structural and connectivity tuning of the NN that optimally navigates the trifecta of order, information, and equilibrium. Intriguingly, the C-E curve appears almost identical beyond a certain depth that NN possesses (Appendix, Fig. 8c,d).

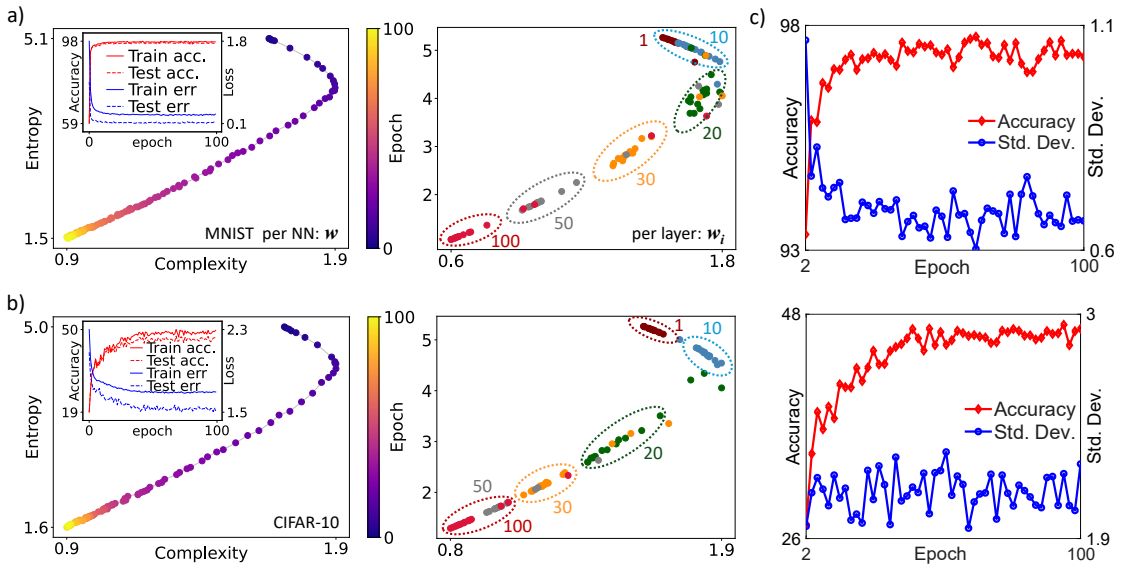

Figure 3: Model performance on Complexity-Entropy plane: a) considers all layer weights $\mathbf{w} = [\mathbf{w}_1, \mathbf{w}_2, \ldots \mathbf{w}_{16}]$ to form a weight distribution for $C(\mathbf{w})$ and entropy $H(\mathbf{w})$ calculation. b) Here, per layer (i) weight distribution, $p(\mathbf{w}_i)$, is used to calculate $C(\mathbf{w}_i)$ and $H(\mathbf{w}_i)$, and each circle represents C-E projection of all layers. a-b) For MNIST (upper panel) and CIFAR-10 (lower panel), early training epochs increase complexity, but this effect is reversed after about 10 epochs. Training error, shown in the inset plot, falls sharply by 10 epochs and does not decrease further thereafter. This trend remains similar for both datasets tried. c) Mean accuracy and standard deviation over 20 testing results using weights $\mathbf{w}$ obtained after different training epochs, both for MNIST (upper) and CIFAR-10 (lower). A low stdev near 50 epochs indicates dependency of reproducibility on complexity — MLP achieves improved reproducibility and competitive accuracy when trained in the optimal C-E region. (See Fig. 9 for the results on a different setup.)

3.2. Weights guided by the C-E plane achieve enhanced reproducibility

The C-E plane posits training on order and disequilibrium plane, where the optimality lies away from extrema of both [3, 12]. For instance, when MLP is trained for 100 epochs, the weight tuning drives the model towards a highly ordered (low information) and low disequilibrium region, compromising both the network complexity and the information stored in it (see Fig. 3). To assess how the system performs for C-E-guided weight distribution tuning, we calculated 20 accuracies, each computed 20 times on randomly selected 500 test samples, and reported the mean accuracy and standard deviation. As observed, the weight distribution obtained for epochs 25 to 50 leans towards a smaller standard deviation while maintaining competitive accuracy (both on MNIST, Fig. 3c, upper and CIFAR-10, Fig. 3c, lower panel), suggesting that improvement in reproducibility may relate to the optimal region identified in the C-E plane. Also, training an NN for epoch region 20-40, as identified on the C-E plane, of different layer sizes, model accuracy gradually increases regardless of the layer size considered (see Appendix, Fig. 8a).

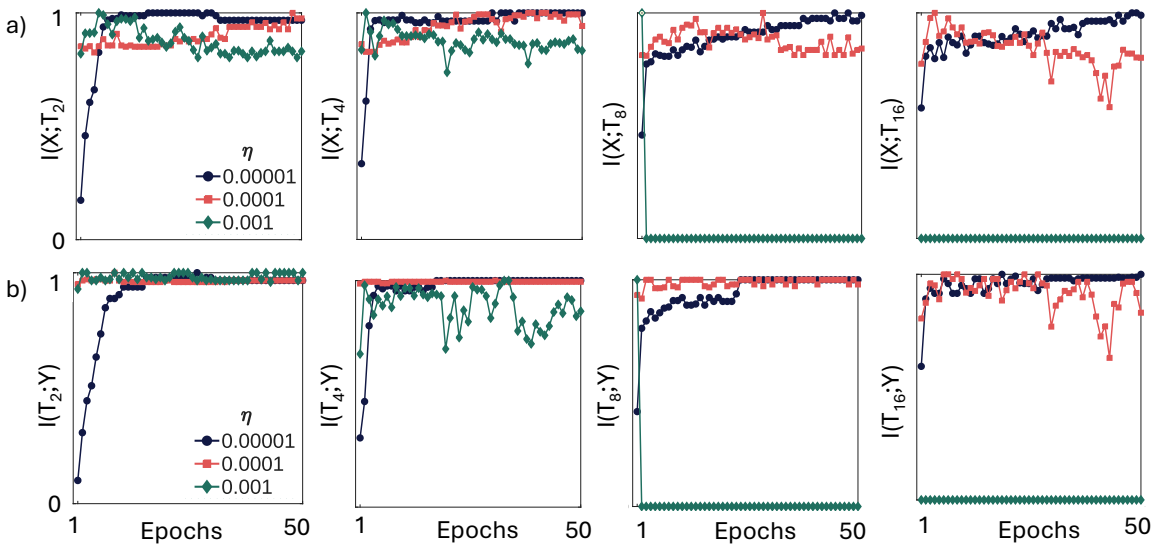

Figure 4: Information Planes for learning rate $\eta = 0.001$, 0.0001 and 0.00001 for layer sizes 2, 4, 8, 16. Evolution of MI (normalized) shown across 50 epochs. Top row (a) is the MI between input and the last hidden layer of network (T_N). Bottom row (b) is the MI between last hidden layer and the output.

3.3. Mutual Information and network-depth dependency

The evolution of MI between hidden layers and X or Y can be scrutinized either with respect to the layers' distance from X or Y, or in terms of the number of epochs of training. We examine the mutual information of the last hidden layer (Fig. 4). For higher values of η, larger networks show a decrease in MI with both X and Y. As observed, MI across layers depends on both η and the MLP's depth. For a higher η (0.001), MI beyond $I(X, T_8)$ diminishes, whereas it remains for a reduced (< 0.001) η. Also, MI changes erratically with a higher η than with a lower one, and this holds for $I(X; T_N)$ and $I(T_N; Y)$ both in an N-layer MLP. η dependency may further explain why smaller η results in more stable MI values earlier during training, and larger η diminishes availability of MI in hidden layers. Thus, optimal training dynamics may entail a trade-off between fluctuations in the MI and the learning rate. However, such observation is subject to extensive exploration across models and data and remains part of our ongoing tasks. See Appendix A for extended MI results.

4. Discussion

This paper explores the learning dynamics of neural networks by studying an MLP. We capture the dynamical transformation of weights from the initial condition to its final form and analyze it using information-theoretic and statistical approaches, including statistical complexity, the Complexity-Entropy plane, KL-divergence, MSD, and mutual information between the input, hidden layers, and output. While the initial traces of C-E dependency on model optimality are promising, its broader relevance remains contingent on whether C-E guidance is equally applicable to more complex datasets and models.

References

[1] Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. *Advances in neural information processing systems*, 32, 2019.

[2] Hans-Dieter Block. The perceptron: A model for brain functioning. i. *Reviews of Modern Physics*, 34(1):123, 1962.

[3] David P Feldman and James P Crutchfield. Measures of statistical complexity: Why? *Physics Letters A*, 238(4-5):244–252, 1998.

[4] Abdenour Hacine-Gharbi and Philippe Ravier. A binning formula of bi-histogram for joint entropy estimation using mean square error minimization. *Pattern Recognition Letters*, 101: 21–28, 2018.

[5] Uri Hasson, Samuel A Nastase, and Ariel Goldstein. Direct fit to nature: an evolutionary perspective on biological and artificial neural networks. *Neuron*, 105(3):416–434, 2020.

[6] Geoffrey E Hinton and Drew Van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the sixth annual conference on Computational learning theory*, pages 5–13, 1993.

[7] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

[8] Pedro Jiménez-González, Miguel C Soriano, and Lucas Lacasa. Leveraging chaotic transients in the training of artificial neural networks. *Physical Review Research*, 8(2):L022032, 2026.

[9] Christian Keup, Tobias Kühn, David Dahmen, and Moritz Helias. Transient chaotic dimensionality expansion by recurrent networks. *Physical Review X*, 11(2):021064, 2021.

[10] Alexei A Koulakov, Tomáš Hromádka, and Anthony M Zador. Correlated connectivity and the distribution of firing rates in the neocortex. *Journal of Neuroscience*, 29(12):3685–3694, 2009.

[11] Shuhong Liu, Nozomi Akashi, Qingyao Huang, Yasuo Kuniyoshi, and Kohei Nakajima. Exploiting chaotic dynamics as deep neural networks. *Physical Review Research*, 7(3):033031, 2025.

[12] Ricardo Lopez-Ruiz, Hector L Mancini, and Xavier Calbet. A statistical measure of complexity. *Physics letters A*, 209(5-6):321–326, 1995.

[13] Marvin Minsky and Seymour A Papert. *Perceptrons, reissue of the 1988 expanded edition with a new foreword by Léon Bottou: an introduction to computational geometry*. MIT press, 2017.

[14] Steven J Nowlan and Geoffrey E Hinton. Simplifying neural networks by soft weight sharing. In *The mathematics of generalization*, pages 373–394. CRC Press, 2018.

[15] Mathilde Papillon, Sophia Sanborn, Johan Mathe, Louisa Cornelis, Abby Bertics, Domas Buracas, Hansen J Lillemark, Christian Shewmake, Fatih Dinc, Xavier Pennec, et al. Beyond euclid: An illustrated guide to modern machine learning with geometric, topological, and algebraic structures. *Machine Learning: Science and Technology*, 6(3):031002, 2025.

[16] Roman Pogodin, Jonathan Cornford, Arna Ghosh, Gauthier Gidel, Guillaume Lajoie, and Blake A Richards. Synaptic weight distributions depend on the geometry of plasticity. In *International Conference on Learning Representations*, volume 2024, pages 36145–36168, 2024.

[17] Jorma Rissanen. Stochastic complexity and modeling. *The annals of statistics*, pages 1080–1100, 1986.

[18] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[19] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215, 2019.

[20] Louis K Scheffer, C Shan Xu, Michal Januszewski, Zhiyuan Lu, Shin-ya Takemura, Kenneth J Hayworth, Gary B Huang, Kazunori Shinomiya, Jeremy Maitlin-Shepard, Stuart Berg, et al. A connectome and analysis of the adult drosophila central brain. *elife*, 9:e57443, 2020.

[21] Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

[22] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 ieee information theory workshop (itw)*, pages 1–5. Ieee, 2015.

[23] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 ieee information theory workshop (itw)*, pages 1–5. Ieee, 2015.

[24] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[25] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[26] Marc Wiedermann, Jonathan F Donges, Jürgen Kurths, and Reik V Donner. Mapping and discrimination of networks in the complexity-entropy plane. *Physical Review E*, 96(4):042304, 2017.

[27] Weishun Zhong, Ben Sorscher, Daniel D Lee, and Haim Sompolinsky. A theory of learning with constrained weight-distribution. *arXiv preprint arXiv:2206.08933*, 2022.

Appendix A. Mutual Information between input, hidden layers, output

We train each MLP over 50 epochs for the MNIST dataset, for varied learning rates. The input to the MLP is X, the N-hidden layers are denoted as $T = \{T_1, T_2, \ldots, T_N\}$, and the output is Y. The information plane consists of two metrics, the MI between the input and hidden layer, $I(X;T_i)$, and the MI between hidden layer and the output, $I(T_i;Y)$. Supervised learning transforms the input variable X into a representation T(X) that facilitates accurate prediction of the output Y. In DNN, learning via Stochastic Gradient Descent (SGD) and backpropagation generates a Markov chain representation of T(X), known as the hidden layers, satisfying the following relationship:

$$I(X;T_1) \geq I(X;T_2) \geq I(X,T_3) \geq \cdots \geq I(X;T_N) \tag{5}$$

Evidently, as we move further away from the input into deeper layers, the MI between hidden layers and the input diminishes (see Fig. 5, 6).

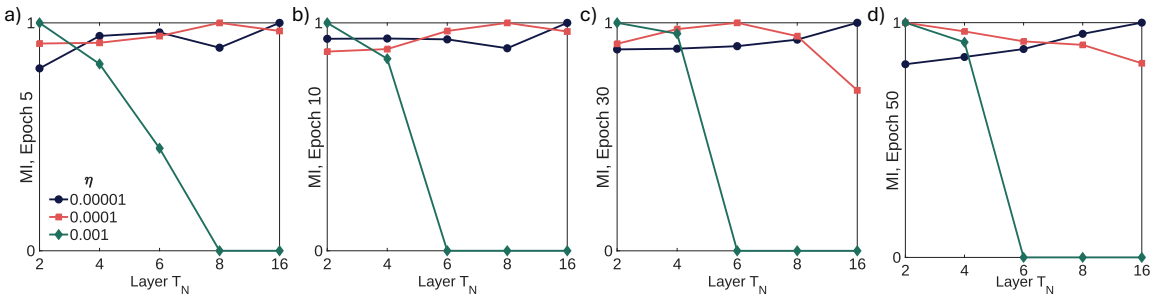

Figure 5: Whether and how the evolution or diminishment of mutual information may inform the optimal network size and learning rate of a system for some particular input data may be of interest. The MI between input X, and the last hidden layer of networks of size 2, 4, 6, 8, and 16 (along x-axis), at epoch number (a) 5, (b) 10, (c) 30, and (d) 50. For higher η, and earlier epochs, shallower networks can retain mutual information well, as evident from (a). The switching point, i.e. the number of layers beyond which the MI becomes zero, may relate to the accuracy and optimality of the model.

Appendix B. Cumulative MSD of Perceptron

B.1. Assessment of Perceptron weight displacement

For an n-input perceptron, Mean Square Displacement (MSD) per-epoch (e_i) is a measure of the average distance the weight vector $w = [w_1, w_2, \ldots w_m]$ traverse in the $w \in \mathbb{R}^n$ space each time perceptron fails to classify samples $\mathbf{x} = [x_1, x_2, \ldots x_n]$ and update following $w_{i+1} = w_i \pm \eta(y_i x_i)$ with η the learning rate. MSD $= \langle |\boldsymbol{w}(x_j + 1) - \boldsymbol{w}(x_j)|^2, \forall x_j \in \mathbf{x} \rangle$. In online training for the perceptron, the MSD calculation considered the displacement d_i for each error ($i = 1, 2, \ldots, N$) committed by the perceptron. After first error, $d_{i=1} = |\mathbf{w}_{\text{ref}} - \mathbf{w}_{i=1}|$, with $\mathbf{w}_{i=1}$ becoming new $\mathbf{w}_{\text{ref}}$ and so on. The final MSD for an epoch was the average of all the N displacements d_i.

Cumulative MSD captures the displacement of the weights accumulated over the epochs for consecutive errors. As shown in Fig. 7 (b,d,f,h), the cumulative MSD increases gradually for earlier

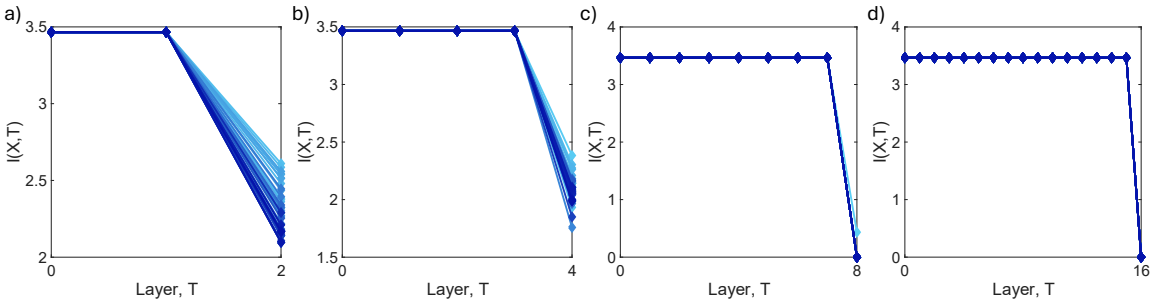

Figure 6: The intuitive relationship between hidden layers and inputs inferred from Eq. 5 - as training progresses, the resemblance between the hidden layer and input is reduced. For increasing depth of layers in a network, the MI $I(X,T)$ should be non-decreasing. Example shown for 2, 4, 8, and 16 layer networks for $\eta = 0.001$, a-d respectively. Color gradient pale blue to dark blue indicates first epoch (1) to last epoch (50).

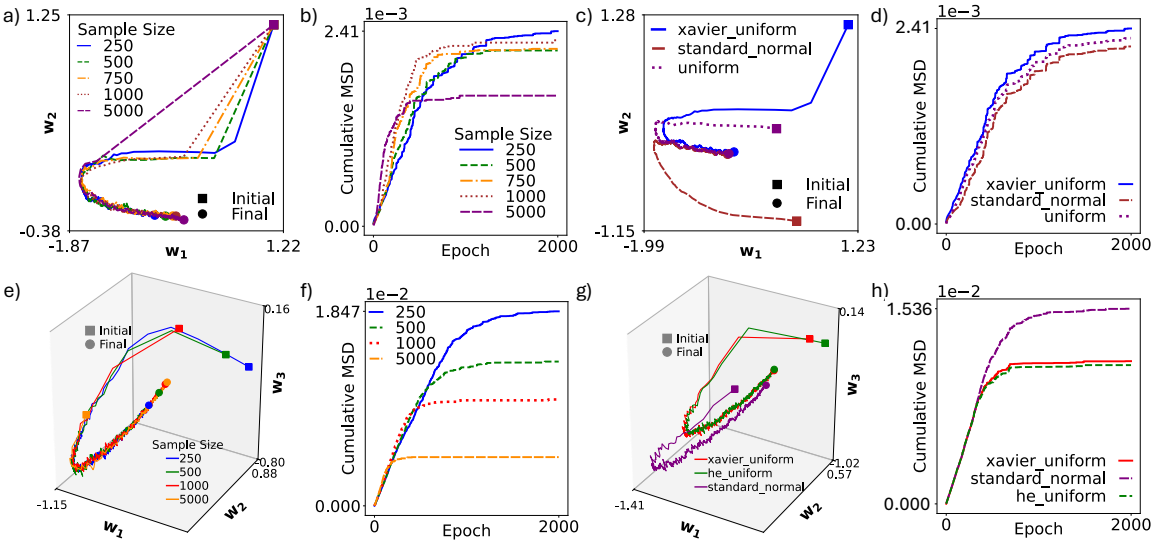

Figure 7: Weight trajectory and the corresponding cumulative MSD per epoch for varying sample sizes and initializations. Top row (a-d) includes results for 2-input Perceptron and bottom row (e-h) for 3-input Perceptron (see experimental setup in Appendix C). (a), (e) shows the weight trajectory for various sample sizes and (b), (f) are their cumulative MSD, (c), (g) shows the weight trajectory for various initializations and (d), (h) are their cumulative MSD.

epochs and reach a plateau near 500 epochs, suggesting a reduction in misclassifications in that region. This is reflected in the weight trajectories Fig. 7 (a,c,e,g) as large initial displacements of weights in earlier epochs progress to smaller ones. That is, the shift in weights becomes negligible as training progresses. For different initializations, the cumulative MSD trend remains generally

similar and regardless of initial positions, all weights reach to similar final positions. Larger sample sizes reach the plateau earlier since more misclassifications are corrected in the initial epochs. These observations are consistent for both 2-input and 3-input perceptron.

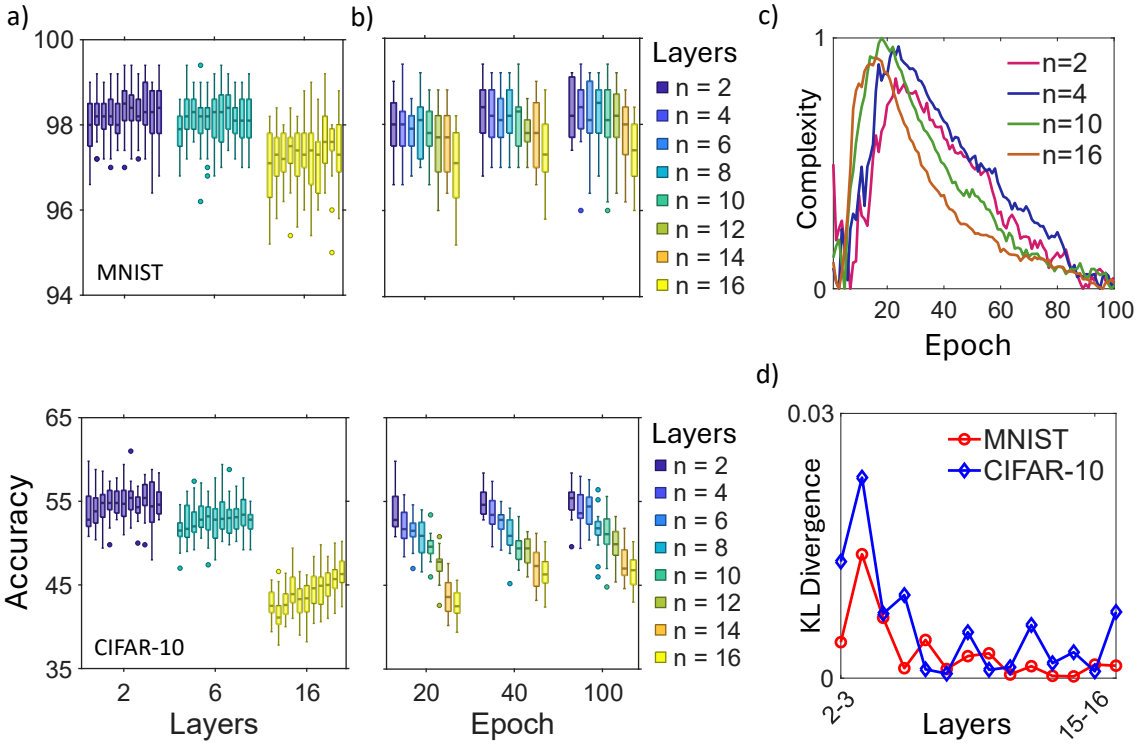

Figure 8: Accuracy for various epochs and network sizes for MNIST and CIFAR-10. (a) For networks with layer sizes 2, 6, and 16 within the epoch range 20-40. (b) shows the accuracy for layers 2, 4, 8, 10, 12, 14, and 16 across epochs 20, 40, and 100. (c) Normalized complexity for layers 2, 4, 10, and 16. (d) KL divergence of complexity between consecutive layer sizes. Beyond layer 6, the shift in the complexity distribution shows negligible change.

Appendix C. Experimental Setup

We ran two sets of experiments. In the first, an MLP with 16 hidden layers (512 neurons each), batch normalization, and 0.3 dropout was trained on MNIST (784 inputs) and CIFAR-10 (3072 inputs), both with 10 outputs. We used ReLU as activation between the hidden layers, and applied softmax to the output layer. The model was trained for 100 epochs with 32 batch size, Adam optimizer (learning rate = 0.001) and a weight decay of 1e-4. In the second setup, a single perceptron was trained on MNIST (converted to a binary classification problem: zero vs. non-zero), a synthetic 2D dataset, and a synthetic 3D dataset, each with 60,000 samples and an 80/20 train-test split.

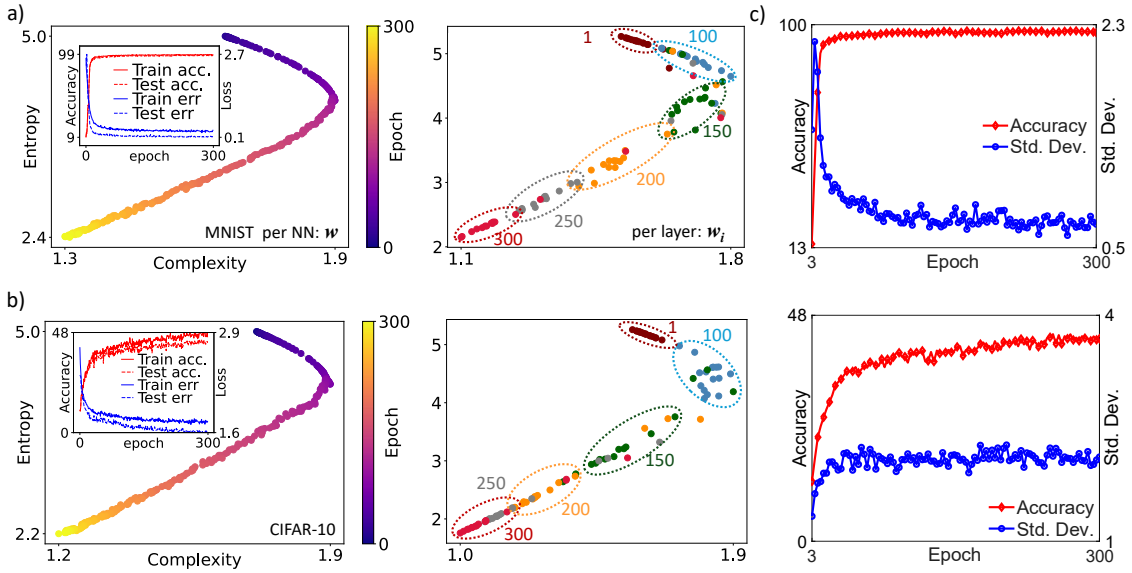

Figure 9: Complexity experiment on a different setup: MLP trained on 5000 training data for 300 epochs. Here, the reversal of the complexity in the C-E plane occurs around 120 epoch while the training error stops decreasing after 30 epochs. In that region, accuracy shows low standard deviation while remaining competitive.