Emergence and Relevance of Criticality in Deep Learning

Juyong Song,1,2,3 Matteo Marsili,3, * and Junghyo Jo1,2,4, †

1Asia Pacific Center for Theoretical Physics, Pohang, Gyeongbuk 37673, Korea
2Department of Physics, Pohang University of Science and Technology, Pohang, Gyeongbuk 37673, Korea
3The Abdus Salam International Centre for Theoretical Physics, Strada Costiera 11, 34014 Trieste, Italy
4School of Computational Sciences, Korea Institute for Advanced Study, Seoul 02455, Korea

(Dated: November 1, 2017)

Deep learning has been successfully applied to various tasks, but its underlying mechanism remains unclear. Neural networks map input data to hidden states in deep layers. As deeper layers have fewer degrees of freedom, subsets of data are transformed into identical states in the deep layers. In this sense, deep learning can be considered as a hierarchical data grouping process. In this Letter, we discover that deep learning forces the size distributions of the data clusters to follow power laws with a different power exponent within each layer. In particular, we identify a critical layer where the cluster size distribution obeys a reciprocal relationship between rank and frequency, also known as Zipf’s law. Deep learning ensures balanced data grouping by extracting similarities and differences between data. Furthermore, we verify that the data structure in the critical layer is most informative to reliably generate patterns of training data. Therefore, the criticality can explain the operational excellence of deep learning and provide a useful concept for probing optimal network architectures.

Introduction. – Deep learning (DL) is similar to physics modeling in the sense that it extracts relevant features from data that can be used for discriminating and generating data [1]. DL has state-of-the-art performance in various fields, including image/voice recognition and language translation [2]; however, why DL works so well remains unclear [3, 4]. Recently, DL has been applied to learn phases and phase transitions in physics [5]. The great success of DL has been explained by hierarchical feature extraction, in which primitive features are extracted in shallow layers near the input layer, while more abstract features emerge in deeper layers after processing the primitive features [2, 6]. However, this algorithmic explanation alone does not answer the following quantitative questions: how relevant are the extracted features for representing the data, and what are the optimal depth and width of the layers for effective data representation?

DL is an effective data grouping method [6–8] in which information is propagated from the input layer through progressively deeper layers. Considering that typical artificial neural networks have narrower architectures in the deeper layers, subsets of input data are transformed to identical states in hidden layers due to the dimension reduction. The lower-dimensional states in the hidden layers act as labels to represent subsets of the input data. Note that data grouping can be considered to be a generalization of data classification based on the output state in discriminative models. Here, the label frequency, corresponding to the subset size, encodes the data grouping in deep neural networks. Given a set of data, what is the most informative grouping to extract the relevant features? If each sample in the data was separately clustered as a distinct group by ignoring the similarities between samples, or if the entire dataset was clustered into a single group by ignoring the differences between samples, no information would be obtained about the data. In addition to these two extreme groupings, any alternative uniform grouping would be similarly uninformative. Two information measures have been proposed to evaluate the relevance of data grouping [9, 10]: the state entropy $H[s]$, which measures the resolution of sample states $s$ in data; and the frequency entropy $H[k]$, which measures the variability of state frequencies $k(s)$. It has been shown that the most informative data grouping structure follows power laws [9]. In this Letter, we study the intrinsic data grouping structures in DL in terms of these two information measures.

Zipf’s law, which is ubiquitously observed in nature [11], is a special power law with an exponent close to unity. The inverse relation between rank and frequency implies the existence of a few frequent patterns and numerous rare patterns. The origin and function of Zipf’s law have been discussed with respect to the information processing in language and communication evolution [12, 13]. Zipf’s law has also been observed in the activity patterns of a real neural network [14], although probing its functionality in the animal brain is a formidable task. Furthermore, the scaling law is related to the criticality in thermodynamics [15]. For energy-based generative models, the log frequency of a state $s$ can be defined as an energy, $E(s) = -\log k(s)$. Since different states can have the same frequency, we denote the degeneracy of frequency (or energy $E_k$) as $m_k$. When the energy degeneracy obeys a power law, $m_k \propto k^{-\beta-1}$, the energy $E_k$ is linearly related to the corresponding entropy,

$$S_k \equiv \log k m_k = \beta E_k + \text{constant}, \tag{1}$$

where $\beta^{-1}$ can be naturally interpreted as an effective temperature. The energy-entropy linearity induces large energy fluctuations with diverging specific heat, $C = -\beta^2(d\beta/dE)^{-1}$, for constant $\beta = dS/dE$.

In this Letter, we briefly introduce a generative DL model. Then, using an artificial neural network, we discover that the data grouping process in DL forces each layer to follow a power law with its characteristic exponent. Specifically, we identify a critical layer that obeys Zipf’s law, such as the real neural network. The generation ability is maximized at the critical layer, which verifies that the criticality is functional.
for faithfully generating patterns of training data.

Unsupervised deep learning. – Among the various DL models, we adopted the deep belief network (DBN), a representative energy-based generative model [16, 17]. A DBN is composed of stacks of restricted Boltzmann machines (RBMs) (Fig. 1a). Each RBM consists of one visible layer and one hidden layer with restricted connections, i.e., visible nodes are not connected to other visible nodes and hidden nodes are not connected to other hidden nodes. Thus, nodes in the same layer are indirectly connected through the nodes in the neighboring layer within an RBM stack. Given a visible and hidden state \((v, h)\), the RBM defines an energy function,

\[
E(v, h; \theta) \equiv -v^T W h - v \cdot a - h \cdot b,
\]

where \(\theta \equiv (W, a, b)\) is the model parameters. Specifically, the matrix \(W\) represents the symmetric coupling strengths between the visible and hidden nodes, and the vectors \(a\) and \(b\) control the biases of the visible and hidden states. Then, a certain state \((v, h)\) has the following probability,

\[
P(v, h; \theta) \equiv \frac{\exp(-E(v, h; \theta))}{Z(\theta)},
\]

with \(Z(\theta) = \sum_v \sum_h \exp(-E(v', h'; \theta))\). Hereafter, for brevity, we omit \(\theta\) from the equations unless necessary. Disconnection between the nodes in the same layer allows the RBM to factorize the probability

\[
P(v, h) = \prod_i P(h_i | v) P(v) = \prod_j P(v_j | h) P(h)
\]

to conditional probabilities. We use the first equation to generate hidden states for given visible states and the second equation to generate visible states for given hidden states. The forward and backward propagations are stochastic, and repeated propagation achieves Gibbs sampling for the hidden and visible layers, respectively [18].

We first need to determine the parameter \(\theta\) that can reliably reproduce data \(v\). Suppose we have \(M\) datasets, \(\{v^{(1)}\}_{\mu=1}^M\), and each dataset has \(N\) components, \(v^{(\mu)} = (v_1^{(\mu)}, \ldots, v_N^{(\mu)})\). For independent datasets, the data likelihood given \(\theta\) is \(\mathcal{L}(\theta) = \prod_{\mu=1}^M P(v^{(\mu)}; \theta)\). Then, the RBM gives the following log-likelihood,

\[
\log \mathcal{L}(\theta) = \sum_{\mu=1}^M \log \sum_h P(v^{(\mu)}, h; \theta),
\]

through marginalization for all possible hidden states \(h\). Learning through the Boltzmann machine algorithm optimizes \(\theta\) by maximizing the log-likelihood [19]. After the learning is completed, we propagate the input data \(\{v^{(\mu)}\}_{\mu=1}^M\) forward to the first hidden layer and obtain \(M\) hidden states \(\{h_1^{(\mu)}\}_{\mu=1}^M\). Here, we denote the visible and the first hidden layer as \(V\) and \(H_1\), respectively. The hidden states \(\{h_1^{(\mu)}\}_{\mu=1}^M\) for \(H_1\) serve as the input data for the second hidden layer \(H_2\) (Fig. 1a). Then, we optimize \(\theta\) for the second RBM stack and repeat the training for the remaining RBM stacks [20].

Informative data grouping. – After learning is completed, the DBN transforms \(\{v^{(\mu)}\}_{\mu=1}^M\) to \(\{h_1^{(\mu)}\}_{\mu=1}^M, \ldots, \{h_k^{(\mu)}\}_{\mu=1}^M, \ldots\) in the hidden layers. Owing to the narrowing of the DBNs toward deeper layers, some distinct states on the shallow layers are transformed to identical states in the deep layers (Fig.1b). Therefore, the data representation can be considered to be a hierarchical data grouping based on the hidden states (Fig.1c). We examine the state statistics for each layer to quantify the data grouping in each DBN layer (Fig.1d). Given \(M\) hidden states, \(\{h_k^{(\mu)}\}_{\mu=1}^M\) for a specific layer, we identify \(L(\leq M)\) distinct states, \(\{s^{(\nu)}\}_{\nu=1}^L\). Then, we count the frequency of state \(s^{(\nu)}\):

\[
k_\nu \equiv k(s^{(\nu)}) = \sum_{\mu=1}^M \delta_{s^{(\mu)}, s^{(\nu)}}.
\]

The state distinguishability (resolution) can be quantified by the Shannon entropy,

\[
H[s] = -\frac{L}{M} k_\nu \log \frac{k_\nu}{M} = -\sum_{k=1}^{k_{\max}} \frac{km_k}{M} \log \frac{k}{M},
\]
where the second description is for the frequency \( k \). Here, \( m_k = \sum_{\mu=1}^{L} \delta_{k,k_\mu} \) denotes the frequency degeneracy, and \( k_{\text{max}} \) denotes the maximum state frequency. Note that the normalization condition is \( \sum_{\nu=1}^{L} k_{\nu} = \sum_{k=1}^{k_{\text{max}}} km_k = M \).

We consider each hidden layer as a different model for data representation. Different hidden layers (models) may extract different features from the data, and it is important to note that the extracted features are encoded in the frequency of the hidden states. If two distinct states \( s^\mu \) and \( s^\nu \) have the same frequency (\( k_{\mu} = k_{\nu} \)), the two states cannot be differentiated. However, if every state has a distinct frequency, the frequency information becomes highly valuable to the identification of relevant features in the data grouping. Therefore, the variability of the state frequency can represent the goodness of feature extraction. The uncertainty of the state frequency can be quantified by another Shannon entropy.

\[
H[k] = -\sum_{k=1}^{k_{\text{max}}} \frac{km_k}{M} \log \frac{km_k}{M}. \tag{8}
\]

On the basis of the state and frequency entropies \( H[s] \) and \( H[k] \), we examined the data grouping of the MNIST data [21]. The data contain \( M = 60,000 \) samples of hand-written digits. Each sample represents a 28 \times 28 pixel image \( (N = 784) \), where each pixel has a real value between 0 and 1. Our DBN architecture has one visible \((V)\) and ten hidden \((H_1, \ldots, H_{10})\) layers that have a decreasing number of nodes \((784-500-250-120-60-30-25-20-15-10-5)\) from \(V\) to \(H_{10}\). After learning was completed, we obtained \( \{h_\ell^\mu\}_{\mu=1}^{M}, \ell \in \{1, \ldots, 10\} \) by propagating the input data \( \{v^\mu\}_{\mu=1}^{M} \) forward to the hidden layers. Again, we emphasize that DL is an agglomerative data grouping, especially in the case of narrowing DBN architectures (Fig. 1c). We computed \( H[s] \) and \( H[k] \) for the ten hidden layers (Fig. 2a). As the layer size shrinks, the state entropy \( H[s] \) decreases monotonically due to the dimension reduction. By contrast, the frequency entropy \( H[k] \) increases up to the eighth hidden layer \( H_8 \) and then decreases. The widely used k-means clustering [22], however, has slightly lower \( H[k] \) than that of the DL. To illustrate the detailed data grouping, we plotted the frequency degeneracy \( m_k \) of the hidden states for different layers (Fig. 2b). Surprisingly, the frequency degeneracy always follows a power law, \( m_k \propto k^{-\beta-1} \), with a different exponent within each layer; however, k-means clustering does not follow a power laws (Fig. 2c). We note that power-law distributions are special in the sense that they maximize \( H[k] \) constrained by a fixed \( H[s] = R \) [9]. This can be easily shown by means of the method of Lagrange multipliers.

\[
\mathcal{F} = H[k] + \lambda_1(H[s] - R) + \lambda_2(\sum_k km_k - M), \tag{9}
\]

where the second constraint comes from the normalization condition. Then, the maximization condition \((\delta \mathcal{F}/\delta m_k=0)\) leads to \( m_k \propto k^{-\lambda_1-1} \), giving rise to \( \lambda_1 = \beta \). The power-law \( m_k \) distributions define a universal \( H[s]\)-\( H[k] \) curve. Here, small \( \beta \) ignores the state distinguishability (resolution), whereas large \( \beta \) emphasizes resolution. One may be tempted to interpret \( \beta^{-1} \) as an effective temperature, because small \( \beta \) (high temperature) corresponds to erroneous data grouping with low resolution. Finally, we confirmed that inducing power law \( m_k \) is an intrinsic property of the RBM regardless of the optimization (learning) of \( \theta \). We generated the hidden states \( \{h_\ell^\mu\}_{\mu=1}^{M} \) before learning with a random \( \theta \); and found that they also showed power law \( m_k \) (Fig. 2c). Furthermore, the corresponding \( H[s] \) and \( H[k] \) were on the universal \( H[s]\)-\( H[k] \) curve, although their specific positions shifted depending on the learning status (Fig. 2a).

Hidden layers represent the input data with different state distinguishability. The distinguishability of shallow layers \( H_1, H_2, \text{ and } H_3 \) is too high to detect similarities between samples in the data; thus, the frequency of distinct samples has poor variability. By contrast, the distinguishability of deep layer \( H_{10} \) is too low to detect differences between samples,
which also leads to the poor frequency variability. Critical layer $H_6$, however, has moderate distinguishability that can group samples into various sizes of clusters with distributions that follow the scale-invariant Zipf’s law with $\beta = 1$. Note that Zipf’s law emerges at $H_k$ maximizing $H[k] + H[s]$, not at $H_k$ maximizing $H[k]$ alone (Fig. 2a). The critical layer successfully detects similarities and differences between samples simultaneously. If one considers the measure $H[k] + H[s]$ as a balance between state distinguishability and frequency variability, Zipf’s law ($m_k \propto k^{-2}$ or $p_k \equiv km_k \propto k^{-1}$) maximizes

$$F' = H[k] + H[s] + \lambda_2 \sum_k km_k - M. \quad (10)$$

**Criticality and pattern generation.** – Is the criticality functional for DL? We examined the generation performance of the generative DBN. After our DBN optimized $\theta$ and learned the hand-written digits, we obtained equilibrium states for each hidden layer. We repeated the backward and forward propagation between $H_\ell$ and $H_{\ell-1}$ 10,000 times for Gibbs sampling to obtain the equilibrium states for the $\ell$th hidden layer, starting from 60,000 random initial states for $H_\ell$. Then, we generated digit images in the visible layer $V$ by propagating the equilibrium states in $H_\ell$ all the way back to $V$. The generated digits appeared different depending on the starting layer $H_\ell$ (Figs. 3a-c). Shallow layer $H_2$ generated heterogeneous digit samples, including some odd-looking digits (Fig. 3a). By contrast, deep layer $H_{10}$ generated stereotyped samples (Fig. 3c). The DBN learned uniformly distributed digits (approximately 6,000 training samples for each digit (0 to 9)). We examined whether the generated digits followed the original distribution of the training digit samples. The generated samples did not have labels (0 to 9), in contrast to the MNIST training samples, which have true labels. Therefore, we labeled the generated samples by means of a classification machine for the hand-written digits with an accuracy of 1.6% error [17]. Then, we quantified the generation ability as the inverse of the Kullback-Leibler divergence

$$\frac{1}{D(P||Q)} = \left[ \sum_{\text{label}=0}^9 P(\text{label}) \log \frac{P(\text{label})}{Q(\text{label})} \right]^{-1} \quad (11)$$

between the two label distributions, $P(\text{label})$ and $Q(\text{label})$, for the training and generated samples. Shallow layers $H_1$, $H_2$, and $H_3$ generated rich digit samples, but their label distribution $Q(\text{label})$ deviated substantially from $P(\text{label})$. The state entropy of the shallow layers is too high to detect relevant features in the data. On the other hand, deep layer $H_{10}$ generated stereotyped samples, and their $Q(\text{label})$ also deviated from $P(\text{label})$. Finally, critical layer $H_6$, which balanced the state entropy $H[s]$ and the frequency entropy $H[k]$ with the critical exponent $\beta \approx 1$, showed the highest generation ability (Fig. 3d). Therefore, the balanced measure, $H[k] + H[s]$, is a good indicator of the generation performance.

We investigated this conclusion with different DBN architectures for the MNIST data and for the OCR data (lower-case letter images) [23]. Regardless of the network architecture and data, we confirmed that the critical layers always showed the highest generation ability (Fig. 3e and 3f).

**Conclusions.** – We studied DL in terms of information-theoretic measures. Our findings are twofold. First, DL is a data grouping labeled by hidden states. We discovered that DL forces the size distribution of grouped data to follow power laws with a different exponent $\beta$ for every hidden layer. From an information theory perspective, the power laws maximize the frequency entropy given a fixed state entropy [9, 10]; thus, DL maximizes the size variability of grouped data at each hidden layer with the respective resolution. However, how the RBMs generate the power laws remains an open question, although Schwab et al. have shown that Zipf’s law ($\beta = 1$) can naturally emerge without fine tuning in the thermodynamic limit when fluctuating hidden variables affect systems [24].

Second, we identified a critical layer, obeying Zipf’s law,
among the hidden layers. In the typical architectures of deep neural networks, deeper layers from the input layer have fewer degrees of freedom and nodes. Therefore, shallow layers have many nodes that contribute to high sample distinguishability, whereas deep layers have few nodes, which inevitably results in low sample distinguishability. However, both cases result in low frequency variability of the distinguishable samples. The critical layer balances the sample distinguishability and frequency variability. Therefore, the criticality, observed in a real neural network [14], could be functional for optimal pattern generation in artificial neural networks.

Information theory provides a promising approach for understanding DL. Information bottleneck theory has quantified the information transfer through deep neural networks [25, 26]. Here, we propose that the state entropy and frequency entropy are useful local measures that can help to design an efficient network architecture for DL. For example, one can stop adding hidden layers or nodes when they start to show the criticality. Furthermore, the two information measures can be used to probe different neural networks with similar performance.

We thank Saeed Saremi for the discussion during the initial stage of this study; and Ji Hyun Bak and Changbong Hyeon for their helpful comments. This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF), funded by the Ministry of Education (2016R1D1A1B03932264), and the Max Planck Society, Gyeongsangbuk-Do and Pohang City (J.J.), and partially supported by the ICTP through the OEA-AC-98 (J.S.).

Supporting Material: Simulation details

The Boltzmann machine algorithm updates \( \theta \) to maximize the log-likelihood of \( \mathcal{L}(\theta) \) in Eq. (5). The updating formulation for the parameter is given by

\[
\theta' = \theta + \alpha \frac{\partial \log \mathcal{L}(\theta)}{\partial \theta}
\]

with learning rate \( \alpha \). The probability gradient for the coupling strength \( W_{ij} \) is derived as

\[
\frac{\partial \log \mathcal{L}(\theta)}{\partial W_{ij}} = \sum_{\mu} \sum_{h} v_{i}^{\mu} h_{j}^{\mu} P(v^{\mu}, h^{\mu}) - \sum_{v} \sum_{h} v_{i} h_{j} P(v, h).
\]

The following probability gradients are obtained in a similar manner:

\[
\frac{\partial \log \mathcal{L}(\theta)}{\partial a_{i}} = \sum_{\mu} \sum_{h} v_{i}^{\mu} P(v^{\mu}, h^{\mu}) - \sum_{v} \sum_{h} v_{i} P(v, h),
\]

\[
\frac{\partial \log \mathcal{L}(\theta)}{\partial b_{i}} = \sum_{\mu} \sum_{h} h_{i} P(v^{\mu}, h^{\mu}) - \sum_{v} \sum_{h} h_{i} P(v, h).
\]

To compute the gradients, we use Gibbs sampling, called the contrastive divergence (CD) method [27], instead of directly obtaining the joint probabilities, \( P(v^{\mu}, h^{\mu}) \) and \( P(v, h) \). A restricted Boltzmann machine has a special structure where the nodes in the same layer are not directly coupled. Therefore, the probability of visible/hidden node activity can be written as a product of the conditional probabilities of the individual nodes in a layer. The conditional probability for forward propagation from \( v \) to the \( j \)th hidden node \( h_{j} \) is

\[
P(h_{j} \mid v) = \frac{1}{1 + \exp(\Delta E_{j})},
\]

where \( \Delta E_{j} \equiv E(v, h) - E(v, f_{j}(h)) \) and \( f_{j}(h) \) is a flip operation for \( h_{j} \). Similarly, the conditional probability for backward propagation from \( h \) to the \( j \)th visible node \( v_{j} \) is

\[
P(v_{j} \mid h) = \frac{1}{1 + \exp(\Delta E_{j})},
\]

where \( \Delta E_{j} \equiv E(v, h) - E(f_{j}(v), h) \), and \( f_{j}(v) \) is the flip operation for \( v_{j} \). We conduct Gibbs sampling with these conditional probabilities by propagating input data \( v^{\mu}(0) \) forward and backward \( n \) times: \( v^{\mu}(0) \mapsto h^{\mu}(0) \mapsto v^{\mu}(1) \mapsto h^{\mu}(1) \mapsto \cdots \mapsto v^{\mu}(n) \mapsto h^{\mu}(n) \). Then, the Gibbs sampling can approximate Eq. (S2) as

\[
\frac{\partial \log \mathcal{L}(\theta)}{\partial W_{ij}} = \frac{1}{M} \sum_{\mu=1}^{M} v_{i}^{\mu}(0) h_{j}^{\mu}(0) - v_{i}^{\mu}(n) h_{j}^{\mu}(n).
\]

To find the global minimum more efficiently, we adopted the mini-batch method by using multiple batches of data instead of considering all the data at once. The randomly grouped batches introduce stochasticity to reduce the likelihood of becoming trapped in local minima. We used 100 samples for a batch with \( n = 4 \) CD steps, and computed the gradient in Eq. (18). Then, we continued to compute the gradient with different batches. Each parameter update for a batch is called an epoch. We used the persistent CD method in which the final state \( (v^{\mu}(n), h^{\mu}(n)) \) given a batch is used for the initial state of the following batches [28]. We used real values between 0 and 1 for the visible node activities and binary values.
state frequencies because the real-valued states $h$ were too close to the true values: $3294$, $1140$, $1847$, $919$, $4200$, $801$, $2380$, $663$, $4172$, $189$, $816$, $2556$, $1415$, $4494$, $3619$, $1283$, $140$, $2570$, $1090$, $1672$, $2258$, $664$, $179$, $413$, $1033$, and $991$. We successfully classified the original data and found frequencies very diverse.

In addition to the MNIST hand-written digit data, we applied our method to the OCR data [23]. The data contain 52152 samples of lowercase letters. Each sample represents a $16 \times 8$ pixel image ($N = 128$), where each pixel has a binary value ($0$ or $1$). We used 44800 samples for training, 3720 samples to validate the hyperparameters, and 3632 samples for testing. The original data contain the following frequencies of letters from a to z: $3333$, $1148$, $1849$, $920$, $4233$, $721$, $2441$, $619$, $3890$, $164$, $785$, $2892$, $1410$, $4523$, $3655$, $1310$, $74$, $2584$, $1075$, $1696$, $2373$, $584$, $150$, $392$, $986$, and $990$. We used a discriminative machine to classify generated samples into lowercase letters. We verified that the machine could successfully classify the original data and found frequencies very close to the true values: $3294$, $1140$, $1847$, $919$, $4200$, $801$, $2380$, $663$, $4172$, $189$, $816$, $2556$, $1415$, $4494$, $3619$, $1283$, $140$, $2570$, $1090$, $1672$, $2258$, $664$, $179$, $413$, $1033$, and $991$. Then, we assessed the generation performance and confirmed that the critical layer ($\beta = 1$) showed the highest generation ability for the OCR data given a network architecture ($28$-320-160-120-100-95-90-85-75-70 for $V$, $H_1$, $\cdots$, $H_9$).