Unsupervised Chaos Clusters

Rajarshi Banerjee

August 2, 2022

Abstract

Deep neural networks(DNN) have demonstrated that they are capable of learning complex patterns enabling the models to make accurate predictions when trained with sufficient labeled data. However, the real world is populated with information which is generally unstructured and unannotated and to learn from such data involves the application of unsupervised learning algorithms. Prior theoretical analysis on the generalizability of DNNs revealed that neural networks trained using labeled data behaves like a chaotic system [10]. In this paper we propose an approach to learn better unsupervised representations by training a DNN auto-encoder which indirectly incentives this chaotic behavior by being trained to maximize its Lyapunov exponents. We show our results on the MNIST and the Fashion-MNIST datasets and compare the unsupervised embedding learnt by our approach to that of a simple autoencoder.

1 Introduction

The advent and application of deep neural networks (DNNs) in various disciples have heralded unforeseen improvements - from tasks as simple as playing Atari games [11] to as complex as understanding the problem of protein folding [12]. This is achieved by the network model learning about the data through the back-propagation algorithm [13] and then generating a output appropriate for the task at hand. However, most DNNs are trained using data which is properly structured and annotated (or making use of a reward function in the previous example) and as such gathering it can be an expensive endeavour. In the unsupervised learning setup, the model does not have access to the ground truth of the data points and needs to learn discriminatory features such that they can be separated into meaningful clusters.

In this paper, we propose an approach of learning the aforementioned features by maximizing the intrinsic chaos in the model. Chaos theory [16] is the study of systems where states diverge exponentially and unpredictably from their expected trajectory when the input given to the system is perturbed ever so slightly. Various systems in real life like the weather [14], stock prices [9] and even societal changes [1] are chaotic systems. Such systems are notoriously difficult to model and recent analysis into the non linear dynamics of DNNs suggest that their state-of-the-art performances can be attributed to the model becoming a chaotic system [10].

In essence, our approach towards unsupervised learning relies on two important components: the reconstruction loss which is induced organically by using an Autoencoder architecture [6] for the neural network where only a limited information is transferred to the decoder depending on the size of the bottle-neck layer and then by maximizing the Lyapunov exponent [5] of the network. The Lyapunov exponent is a quantity that characterizes the rate of change of infinitesimally close trajectories in dynamic systems and in general can be used to measure the chaotic nature of a system, in context of a neural network one can approximate this quantity by comparing the general output of the network to the output produced by a slightly perturbed input as visualized in Figure (1).

Note that the two loss functions are trying to accomplish to two almost different objectives, maximization of the Lyapunov exponent increases the variation of the output of the network while the reconstruction loss decreases the variation of between the input and the output of the network. The rationale behind this is that the internal embeddings learnt by the model for datapoints belonging to separate classes will have a greater distance between them as compared to those belonging to the same classes.

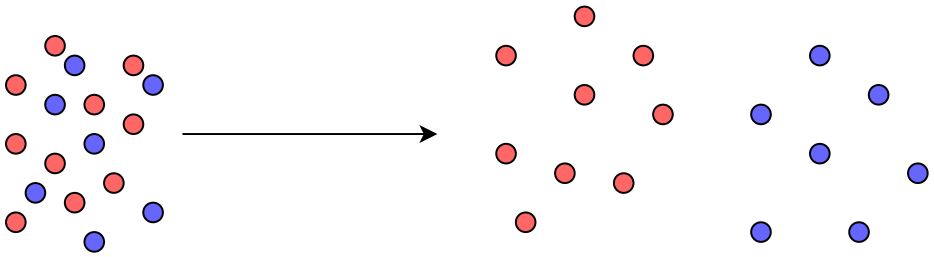


Figure 1: By maximizing the Lyapunov exponents we incentivize the neural network to maximize the distance between the output of the datapoints along with minimizing the reconstruction loss which prevents the embedding from going too far apart.

We report the results of the clusters formed using our methodology and compare it against that of a simple Autoencoder on the MNIST and Fashion-MNIST dataset using only two classes, this is because experimentally it was observed that performance detoriates rapidly when more than two clusters are involved. Thus, rather than being a holistic end to end unsupervised learning solution it could be more practical to use it as a self supervised learning task for more involved unsupervised learning algorithms like SCAN [15].

2 Related Work

The most successful unsupervised learning approaches are end to end approaches incorporating pipelines which combine feature learning and clustering: Unsupervised Deep Embedding for Clustering Analysis [17] (DEC) made use of KL Divergence to cluster data points, DeepCluster [3] leveraged the inherent architecture of CNNs to iteratively cluster and train on prior labels predicted by the network which was improved upon by DeeperCluster [4]. All these approaches relies on cycles of training the network using prior labels based on initial predictions followed by clustering. A slightly different approach proposed the maximization of mutual information between input images and its augmentations [8, 7].

Representation learning [2] relies on self-supervised learning with various pre-designed tasked to cluster datapoints. SCAN [15] used a two step approach for feature learning and clustering where it combined self-supervised learning to achieve state-of-the-art unsupervised learning classification. Our approach deviates considerably from all the previous methods by solely relying on chaos maximization and thus could be used a a pre-designed task for better classification performance.

3 Method

3.1 Lyapunov exponents

Given a dynamic system Z with states that change over time $\{Z_0, Z_1, \ldots, Z_t\}$ and two initial starting states: $Z_0 = x_0$ and $\delta Z_0 = x_0 + \delta(0)$, the lyapunov exponent or the lyapunov characteristic exponent λ is defined as:

$$\left|\delta Z_t\right| \approx e^{\lambda t}\left|\delta Z_0\right| \tag{1}$$

where $Z_i, \delta Z_i, x_o, \delta(0) \in \mathcal{R}^n$ and $\epsilon > \delta(0) > 0$. Since the state space of the system lies in the vector space $\mathcal{R}^n$, the spectrum of Lyapunov exponents are denoted by $\lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$. The largest lyapunov exponent is generally know as the Maximal Lyapunov Exponent (MLE) as it determines the predictability of the system, a system is considered a chaotic system when the MLE is a postive number. This is because the initial separation of between Z_0 and the δZ_0 after t time period becomes:

$$\|\delta(t)\| \approx \|\delta(0)\|e^{\lambda t} \tag{2}$$

Note that equation (2) is a consequence of the evolution of the state space of the system given by equation (1). Formally, the MLE is defined as:

$$\lambda = \lim_{t \to \infty} \lim_{|\delta Z_0| \to 0} \frac{1}{t} \ln |\delta Z_t| \tag{3}$$

Equation (3) is valid for continuous systems. For discrete systems like a neural network where each state is a change in the embedding of the input x_0 as it passes from one layer to another such that $x_{n+1} = f_n(x_n)$ where f_n is the n^{th} layer of the neural network (inclusive of activation function) the MLE is given by:

$$\lambda = \lim_{n \to \infty} \frac{1}{n} \sum_{i=0}^{n-1} \ln |f_n'(x_i)| \tag{4}$$

For our purposes we approximate the derivative of the equation (4) as the ratio of difference of perturbed outputs to that of perturbed inputs, i.e. $\frac{\delta(t)}{\delta(0)}$

3.2 Chaos Loss

As a loss function to the neural network we want to maximize the intrinsic chaos in it and thus would want to minimize $l_c(x_0) = -\lambda$:

$$\begin{aligned} \min_f l_c(x_0) &= \min -\lambda \\ &= \min - \lim_{n \to \infty} \frac{1}{n} \sum_{i=0}^{n-1} \ln |f_n'(x_i)| \\ &\approx \min -\mathbb{E}_n[\ln f_n'(x_i)] \\ &= \min -\mathbb{E}_n[f_n'(x_i)] \\ &= \min - \frac{\|x(n) - x_\delta(n)\|}{\|x(0) - x_\delta(0)\|} \end{aligned} \tag{5}$$

We use the formulation of approximate derivative as specified by Husheng Li [10] in computing the MLE by taking multiple perturbations of the input $x(0)$ with $x_\delta(0) = x(0) + \mathcal{N}(0, 1)$ to obtain the final expression:

$$L_c(x_0) = \min_f \max_{x_\delta} l_c(x_0) \tag{6}$$

3.3 Choas Autoencoder Loss

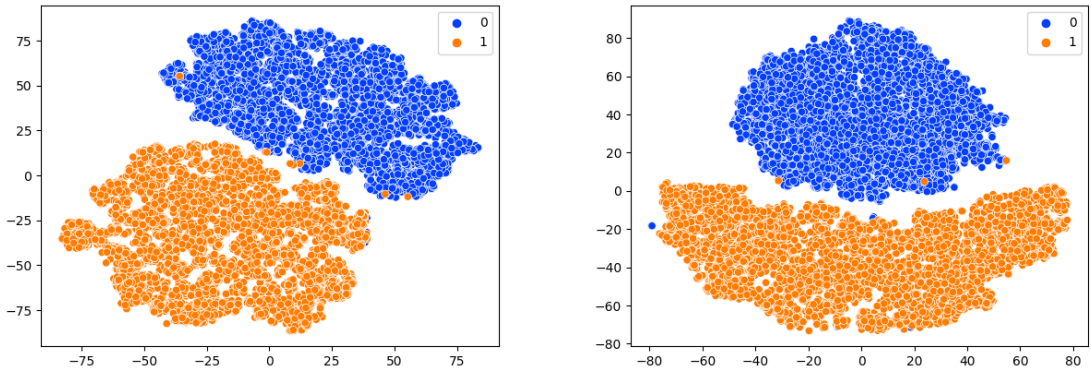
Along with the loss function mentioned in equation (6) the reconstruction loss is also incorporated in order to ensure the output does not completely deviate from the original input. Combining reconstruction loss from the autoencoder ensures that the variation induced by increasing the Lyapunov exponent does not degenerate into noise. Thus, we obtain the final loss function as:

$$L(x_0) = \gamma * \|x(n) - x(0)\|^2 + (1 - \gamma) * L_c(x_0) \tag{7}$$

where γ is a hyper-parameter which weighs the simple reconstruction loss and the chaos loss. $\gamma = 1$ will yield the traditional reconstruction error loss used in an Autoencoder.

4 Experimental Results

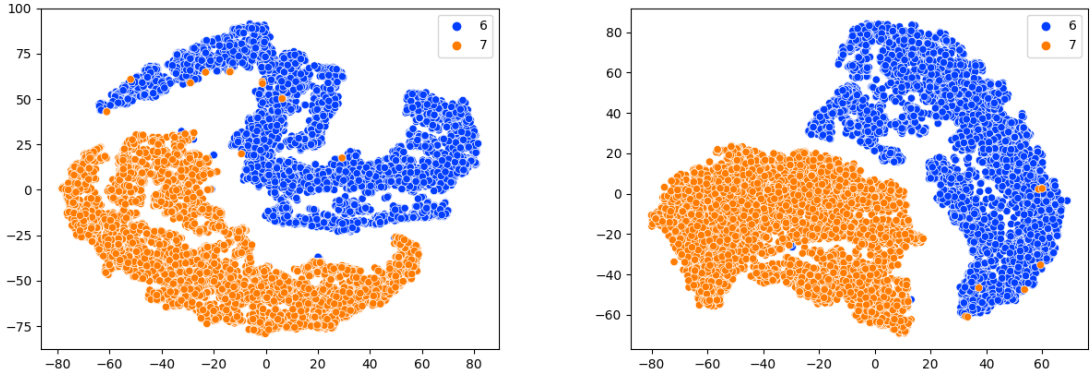
We report the results on the MNIST dataset with datapoints consisting of labels 0 and 1 and Fashion-MNIST dataset with datapoints consisting of labels 6 and 7. The neural network used is a simple CNN with 4 layers.



(a) Autoencoder TSNE embedding

(b) Chaos Autoencoder TSNE embedding

Figure 2: MNIST TSNE plots for (a) Autoencoder and (b) Chaos Autoenconder. Notice how in (b) how the elements belonging to the same class are more compactly positioned.



(a) Autoencoder TSNE embedding

(b) Chaos Autoencoder TSNE embedding

Figure 3: FMNIST TSNE plots for (a) Autoencoder and (b) Chaos Autoenconder. As can be observed the datapoints in (b) can more easily be separated by a linear classifier compared to the points in (a).

Metrics	**Autoencoder embedding**	**Chaos autoencoder embedding**
Rand Score	0.859	0.918
Adjusted Rand Score	0.718	0.836
Adjusted Mutual Information Score	0.657	0.773
Normalized Mutual Information Score	0.657	0.773
Homogeneity Score	0.648	0.769
Completeness Score	0.666	0.778
V-measure score	0.657	0.773
Fowlkes-Mallows scores	0.862	0.919

Table 1: Comparison of clusters for the MNIST dataset

Metrics	Autoencoder embedding	Chaos autoencoder embedding
Rand Score	0.643	0.665
Adjusted Rand Score	0.285	0.332
Adjusted Mutual Information Score	0.322	0.379
Normalized Mutual Information Score	0.322	0.379
Homogeneity Score	0.300	0.355
Completeness Score	0.348	0.407
V-measure score	0.322	0.379
Fowlkes-Mallows scores	0.675	0.694

Table 2: Comparison of clusters for the Fashion-MNIST dataset

References

[1] A. Albert. *Chaos and society*, volume 29. Ios Press, 1995.

[2] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[3] M. Caron, P. Bojanowski, A. Joulin, and M. Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149, 2018.

[4] M. Caron, P. Bojanowski, J. Mairal, and A. Joulin. Unsupervised pre-training of image features on non-curated data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2959–2968, 2019.

[5] J. B. Dingwell. Lyapunov exponents. *Wiley encyclopedia of biomedical engineering*, 2006.

[6] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[7] W. Hu, T. Miyato, S. Tokui, E. Matsumoto, and M. Sugiyama. Learning discrete representations via information maximizing self-augmented training. In *International conference on machine learning*, pages 1558–1567. PMLR, 2017.

[8] X. Ji, J. F. Henriques, and A. Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9865–9874, 2019.

[9] I. Klioutchnikov, M. Sigova, and N. Beizerov. Chaos theory in finance. *Procedia computer science*, 119:368–375, 2017.

[10] H. Li. Analysis on the nonlinear dynamics of deep neural networks: Topological entropy and chaos. *arXiv preprint arXiv:1804.03987*, 2018.

[11] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[12] K. M. Ruff and R. V. Pappu. Alphafold and implications for intrinsically disordered proteins. *Journal of Molecular Biology*, 433(20):167208, 2021.

[13] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[14] J. Slingo and T. Palmer. Uncertainty in weather and climate prediction. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 369(1956):4751–4767, 2011.

[15] W. Van Gansbeke, S. Vandenhende, S. Georgoulis, M. Proesmans, and L. Van Gool. Scan: Learning to classify images without labels. In *European conference on computer vision*, pages 268–285. Springer, 2020.

[16] G. Williams. *Chaos theory tamed*. CRC Press, 1997.

[17] J. Xie, R. Girshick, and A. Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487. PMLR, 2016.