 doi:10.20944/preprints202207.0139.v1

# Epitopological sparse deep learning via network link prediction: a brain-inspired training for artificial neural networks

**Yingtao Zhang, Alessandro Muscoloni, Carlo Vittorio Cannistraci***
Center for Complex Network Intelligence (CCNI), Tsinghua Laboratory of Brain and Intelligence (THBI)
Department of Computer Science, Tsinghua University
Department of Biomedical Engineering, Tsinghua University
{zhangyingtao1024, alessandro.muscoloni}@gmail.com
*Corresponding author: Carlo Vittorio Cannistraci (kalokagathos.agon@gmail.com)

## Abstract

Sparse training (ST) aims to improve deep learning by replacing fully connected artificial neural networks (ANNs) with sparse ones. ST is promising but at an early stage, therefore it might benefit to borrow brain-inspired learning paradigms such as epitopological learning (EL) from complex network intelligence theory. EL is a field of network science that studies how to implement learning on networks by changing the shape of their connectivity structure (epitopological plasticity). EL was conceived together with the Cannistraci-Hebb (CH) learning theory according to which: the sparse local-community organization of many complex networks (such as the brain ones) is coupled to a dynamic local Hebbian learning process and contains already in its mere structure enough information to partially predict how the connectivity will evolve during learning. One way to implement EL is via link prediction: predicting the existence likelihood of each nonobserved link in a network. CH theory inspired a network automata rule for link prediction called CH3-L3 that was recently proven to be very effective for general purpose link prediction. Here, starting from CH3-L3 we propose a CH training (CHT) approach to implement epitopological sparse deep learning in ANNs. CHT consists of three parts: kick start pruning, to hint the link predictors; epitopological prediction, to shape the ANN topology; and weight refinement, to tune the synaptic weights values. Experiments on MNIST and CIFAR10 datasets compare the efficiency of CHT and other ST-based algorithms in speeding up the ANN training across epochs. While SET leverages random evolution and RigL adopts gradient information, CHT is the first algorithm in ST that learns to shape sparsity by using the sparse topological organization of the ANN.

## 1 Introduction

Artificial Intelligence (AI) is developing deep learning algorithms because of their superior performance in various fields, such as Natural Language Processing and Computer Vision. However, to achieve even higher performance, AI models are becoming deeper and more complex, resulting in a crisis of over-parameterization[1], which is also straining computing resources. Researchers are therefore focusing on how to compress the models more effectively. There are many possible ways to compress the model, including Knowledge Distillation[2], Model Quantification[3, 4], Sparse training (ST)[1, 5, 6, 7, 8, 9], etc. However, ST is one of the closest to the original intent of designing Artificial Neural Networks (ANNs). At the outset of the process of designing ANNs, researchers drew lessons from Brain Neural Networks (BNNs)[10]. Hebbian learning was introduced in 1949[11], and

Preprint. Under review.

 doi:10.20944/preprints202207.0139.v1

it is summarized in the axiom: "neurons that fire together wire together". This could be interpreted in two ways: changing the synaptic weights (weight plasticity) and changing the shape of synaptic connectivity (epitopological plasticity)[12, 13, 14]. For long period AI research mainly focused on leveraging weight plasticity by learning on fully connected topologies. Recently, epitopological plasticity is on the verge because ST research tries to compress ANNs connectivity by carving sparse architectures. In this study we will investigate whether it is possible to improve ST by exploiting new complex network intelligence paradigms derived from BNNs, which indeed are mostly sparsely connected.

Current ST includes four main directions: pruning, which was introduced in an article by Lecun et al[15]. in 1989; dynamic sparse training (DST), which was introduced in an article by Mocanu et al. in 2018[7] with the algorithm sparse evolutionary training (SET); NeuroEvolution of Augmenting Topologies (NEAT)[16]; and RigL[6], which is a gradient-based sparse training method. This study aims to introduce a fifth direction called Cannistraci-Hebb training (CHT), which stems from the epitological learning[12, 13, 14] and Cannistraci-Hebb network automata theory for link prediction[17]. They are new brain-inspired concepts recently introduced in network science. The work related with these previous studies is discussed in section 2 below.

EL is a field of network science that studies how to implement learning on networks by changing the shape of their connectivity structure (epitopological plasticity). One way to implement EL is via link prediction: predicting the existence likelihood of each nonobserved link in a network. CH theory inspired a network automata rule for link prediction called CH3-L3 that works on bipartite networks[17]. Here, starting from CH3-L3 we propose a CH training (CHT) approach to implement epitopological sparse deep learning in ANNs. CHT consists of three parts: kick start pruning (KSP), to hint the link predictors; epitopological prediction via CH3-L3, to shape the ANN topology; and weight refinement, to tune the synaptic weights values. The KSP provides EL with a hint from the original sparse network in order to predict new links based on the structured network. Epitopological prediction via CH3-L3 shows the efficiency of EL in finding the proper subnetwork for sparse training: a new solution to find the Lottery Ticket of ANNs[5]. Finally, in the later period of epitopological prediction, the new predicted connections significantly overlap with removed ones. This means that the sparse network has already formed some structural features and its structure turns to be stable. There is no need to run forward anymore the CH3-L3 predictor of EL. Hence, to save computational running time, we propose to truncate with an early stop the epitopological prediction phase when the link overlap rate is higher than a significant level, and in the last epochs to refine the weights learning of the sparse topology.

Computational experiments on MNIST[18] and CIFAR10[19] datasets compare the efficiency of CHT and other ST-based algorithms in speeding up the ANN training across epochs. While SET leverages random evolution to progress towards a scale-free topology and RigL adopts gradient information to suggest how to update the connections, CHT is the first algorithm in ST that learns to shape sparsity by using the topological organization of the ANN.

## 2 Related Work

In this section, we will introduce some basic notions of sparse training, epitopological learning and Cannistraci-Hebb theory for network automata link prediction, with the aim to support our proposed Cannistraci-Hebb training methodology for epitological sparse deep learning in ANNs.

### 2.1 From pruning and dynamic sparse training to epitological learning

ST research in AI started with pruning, which was introduced from an article by Lecun et al. in 1989[15]. The classic pruning (CP) paradigm refers to removing unimportant weights after training and finetuning the weights of existing links to adapt the sparse structure. Following this, various pruning paradigms emerged such as spring mushrooms[20, 21, 22, 23]. Han et al.[20] updated the CP paradigm and extended it to Iterative Pruning (IP). After some other studies in ST[24, 16], Mocanu et al.[7] proposed a first dynamic sparse training algorithm called SET, which is a paradigm closer to a neurobiological perspective. Differently from the CP and previous ST methods, SET initializes the network with a random sparse state and evolves the network structure with both pruning and regrowing simultaneously. They stated that it was inspired by a biological phenomenon: synaptic shrinking during sleep[25]. In order to renormalize the overall synaptic strength, the weakest synapses

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 July 2022 doi:10.20944/preprints202207.0139.v1

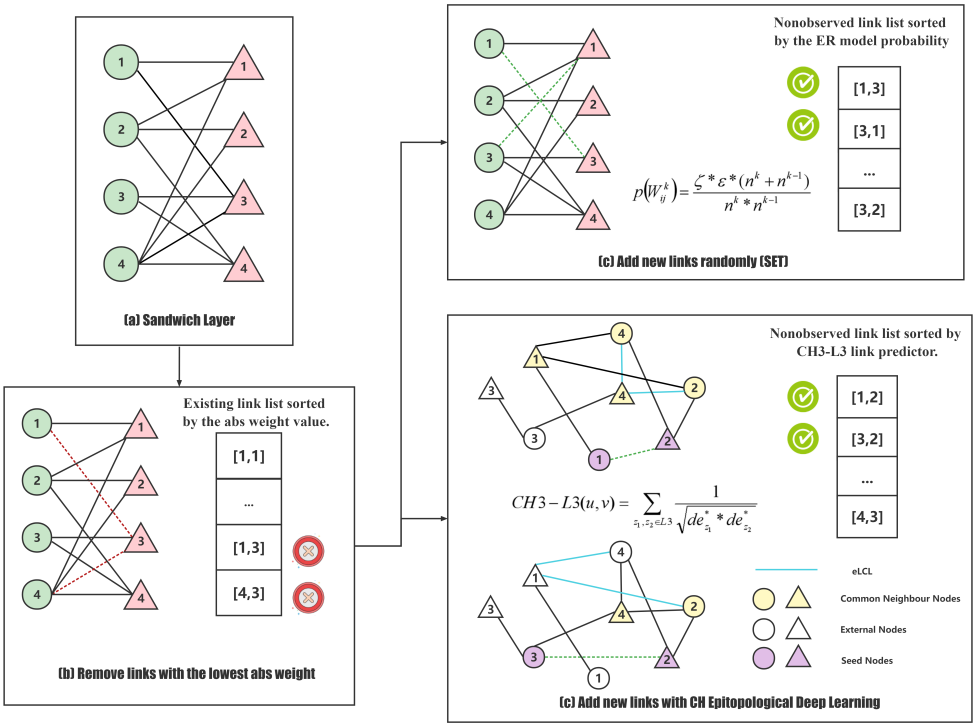


Figure 1: Illustration of CH Epitopological Theory and SET. **(a)** Sandwich layer is a new concept to define the architecture between two consecutive layers of the ANN. **(b)** Removal mechanism by lowest absolute weights (for both ESDL and SET). **(c)** Different addition mechanisms between SET (randomly) and CH Epitopological Learning (based on link prediction for instance by top-ranked CH3-L3 scores).

shrink during sleep, while the strongest synapses remain unchanged, then new synapses regrow when individuals awake. To simulate this procedure, Mocanu et al. introduced SET by pruning the weights that are closest to zero and stochastically regrowing new weights[7].

Besides, a more recent theory is the lottery ticket hypothesis (LTH)[5]. According to the LTH, there must exist a subnetwork of the original fully connected network that can attain a similar performance to the original network within the same training epochs. How to seek that lottery ticket (the specific subnetwork) became the topic of investigation among following studies[6, 1, 9].

RigL[6] is a gradient-based sparse training method that utilizes the instantaneous gradient information of nonobserved links to grow new links. Further notes on all these ST algorithms are provided in the Method section in suppl. info. However, most of them focused their attention on the basic training information, without taking into consideration that sparse ANNs can be seen as a special case of complex networks. Therefore, in this paper, we borrow epitopological leaning that is a new brain-inspired paradigm from network science and introduce it into the field of sparse training.

### 2.2 Epitopological Learning and Cannistraci-Hebb network automata theory for link prediction

EL is a field of network science that studies how to implement learning on networks by changing the shape of their connectivity structure. This is also called epitopological plasticity, because plasticity means "to change shape" and epitopological means "via a new topology". EL was conceived together with the Cannistraci-Hebb (CH) learning theory according to which: the sparse local-community organization of many complex networks (such as the brain ones) is coupled to a dynamic local Hebbian learning process and contains already in its mere structure enough information to partially predict how the connectivity will evolve during learning. The rationale is that, in any complex network with local-community organization, cohort of nodes (neurons in the case of brain networks)

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 July 2022 doi:10.20944/preprints202207.0139.v1

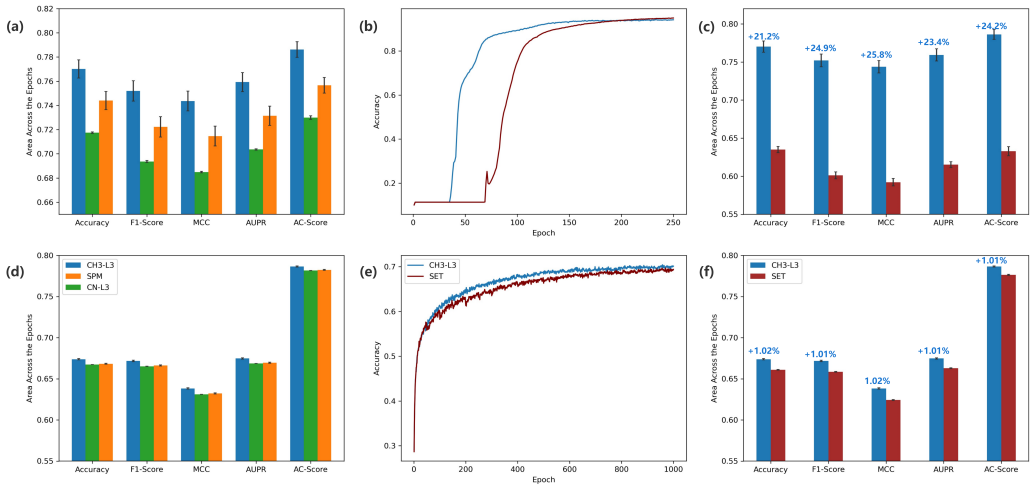


Figure 2: Comparison of ESDP and SET. Panels **(a,b,c)** report the results for the MNIST dataset (250 epochs at steps of 1 epoch), whereas panels **(d,e,f)** for the CIFAR10 dataset (1000 epochs at steps of 50 epochs). **(a,d)** For each epitopological learning method, the barplots show the AAE value (mean and standard error over 3 repetitions) related to different performance metrics (Accuracy, F1-Score, MCC, AUPR, AC-score). **(b,e)** The plots show the accuracy curve over the epochs comparing the best epitopological learning method (CH3-L3) with SET. **(c,f)** For CH3-L3 and SET, the barplots show the AAE value (mean and standard error over 3 repetitions) related to different performance metrics. The percentual increment of CH3-L3 is reported in **(c,f)**.

tend to be co-activated (fire together) and to learn by forming new connections between them (wire together) because they are topologically isolated in the same local community[12].

One way to implement EL is via link prediction: predicting the existence likelihood of each nonobserved link in a network. CH theory was recently proven to be very effective for general purpose link prediction[17, 26, 27], and inspired a network automata rule for link prediction called CH3-L3 that works on bipartite networks[17]. Here, starting from CH3-L3 we propose a CH training (CHT) approach to implement epitopological sparse deep learning in each sandwich architecture (the bipartite subnetwork formed by two consecutive layers) of ANNs.

## 3 Epitopological sparse deep learning and Cannistraci-Hebb training

### 3.1 Epitopological sparse deep learning

Following the second interpretation of the Hebbian Learning rule, we propose epitopological sparse deep Learning (ESDL), which implements EL in the deep learning field. ESDL changes the perspective of ANNs from weights to topology. In particular, we focus on the sparse topology of each sandwich layer: a bipartite sub-network composed of two consecutive layers of the ANN. The first step of ESDL is to initialize the sparse connections of each sandwich layer, for which we can refer to any sparse initialization such as, for instance, the random sparse initialization used by SET and set links randomly according to the ER model. During training, ESDL performs two distinct processes. The first process is the evolution of the sparse topology weights, in which ESDL conducts the backpropagation as standard deep learning techniques, but strictly based on a L0 regularization constraint. The weight evolution is performed after each batch and the entire training set is recycled after every epoch. The second process is the epitopological evolution, which is the key factor in ESDL and is carried out at intervals of epochs, named update interval[6]. The epitopological evolution process is further divided into two stages: removal and prediction. In the removal stage, for each sandwich layer, we remove the links with weights having the lowest absolute value, since they can be considered the least useful for the model performance. In order to maintain constant the sparsity of

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 July 2022 doi:10.20944/preprints202207.0139.v1

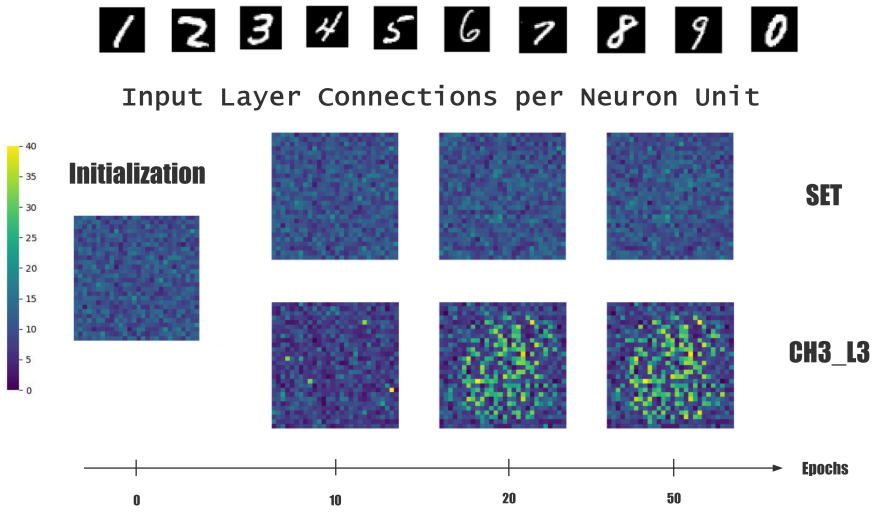


Figure 3: Connections of input layer neurons in MNIST. The heatmaps represent the node degree for each neuron unit in the input layer, equal to the number of connections between the input neuron and neurons in the next layer. Heatmaps are shown for the initialization stage (epoch 0) and for three successive learning stages (epoch 10, 20 and 50), comparing the CH3-L3 and SET methods.

the sandwich layer, the same number of links are added based on the highest likelihood scores given by any link prediction method (we found that CH3-L3 offers the best performance among 3 tested methods) when applied to that sandwich layer. Fig. 1 illustrates the main differences between the evolution steps in Epitopological Learning and SET.

The choice of the hyper-parameters also influences the efficiency of ESDL. Apart from learning rate and batch size, a critical factor is the Update Interval. If the interval is too small, weights are removed without adequate training and the topological structure is not yet informative enough for an accurate link prediction, leaving the risk to fall into a local optimum. A proper tuning of the update interval is less relevant for other algorithms such as SET, since the links are added randomly, and RigL, which adds new links based on the gradient information.

### 3.2 Cannistraci-Hebb training

ESDL is a link prediction based methodology which can make its best use when the network has already formed some relevant structural features. But, the link predictor efficiency might be impaired by a sparse random connectivity initialization. On the other hand, when the link prediction does not vary across epochs, there is not anymore necessity to perform ESDL. Therefore, to address these concerns, we propose a novel 3-step training procedure named Cannistraci-Hebb training (CHT).

- *Kick start pruning (KSP)*: the goal of this mechanism is to provide ESDL with the initial prompt. KSP firstly trains a fully connected network for a small number of epochs (we tested 1 and 50 epochs in MNIST, and 50 epochs only in CIFAR10 for time reasons) and then prunes it to the required sparsity by removing the links with the lowest absolute weights. It is a dense-to-sparse strategy which can form some relevant structural features and give a better hint to the link predictor.
- *Epitopological prediction*: this step corresponds to ESDL (introduced in section 2.2), which evolves the sparse topological structure starting from KSP initialization rather than random. After several evolution epochs, the overlap between links removed and added can reach a high level, meaning that the network has achieved a stable structure and ESDL will continuously remove and add mostly the same links, slowing down the training. For each sandwich layer, when the overlap rate has reached a certain threshold (we considered the significant level of 0.9), we perform an early stop of the epitopological prediction.
- *Weight refinement*: when all the sandwich layers has stopped the epitopological prediction, the model starts to learn and refine only the weights using the obtained network structure.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 July 2022 doi:10.20944/preprints202207.0139.v1

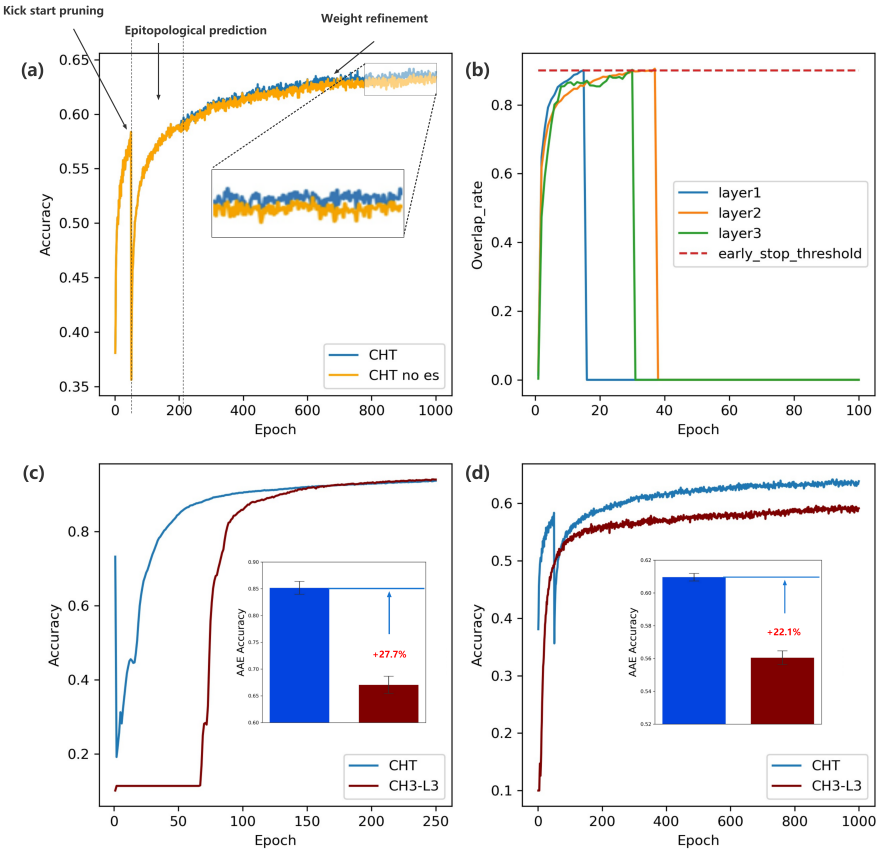


Figure 4: Comparison between CHT and CH3-L3. **(a)** The panel shows the three training periods of the CHT procedure: KSP, epitopological prediction and weight refinement. **(b)** The panel shows the overlap rate between new links and removed links for each layer, highlighting the necessity of applying an early-stop mechanism. **(c)** The plot reports the accuracy curve of CHT for 1-epoch-KSP and CH3-L3 in MNIST dataset (250 epochs). The curve of CHT for 50-epochs-KSP is provided in Fig. 6 in Suppl. Information. **(d)** The plot reports the accuracy curve of CHT for 50-epochs-KSP and CH3-L3 in CIFAR10 dataset (1000 epochs). Panels **(c,d)** contain an embedded subpanel showing the AAE-accuracy value of each algorithm (mean and standard error over 3 repetitions). The percentual increment of CHT is reported in **(c,d)**.

## 4 Result

In order to assess the effectiveness of ESDL and CHT with respect to SET and RigL, we implement these sparse training techniques within a Multi-Layer Perceptron (MLP) and we compare the performance obtained in a computer vision classification task using two different datasets. The first dataset is MNIST, for which we adopt a MLP with the architecture of 784x1000x1000x1000x10. The second dataset is CIFAR10, for which the architecture is 3072x4000x1000x4000x10. In the initial tests proposed in Fig. 2 to fairly compare with SET, we use the SET strategy based on the ER model to initialize the first three sandwich layers with a given sparsity and we set the last sandwich layer as full connected[7], since it directly affects the classification output. The other hyperparameters of the different experiments are indicated in Table 1.

Fig. 2 shows the comparison of ESDL and SET for both MNIST and CIFAR10 datasets. In particular, panels (a,d) report the Area Across the Epochs (AAE) related to several performance metrics for three ESDL methods based on different link predictors: CH3-L3, SPM and CN-L3. CH3-L3 confirmed to be the best link predictor for both datasets, therefore we compared its performance to SET, which is shown in panels (b,c,e,f). According to both the accuracy curve and the AAE values, ESDL based on CH3-L3 manifests an improvement compared to SET (around +25% in MNIST and +1%

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 July 2022 doi:10.20944/preprints202207.0139.v1

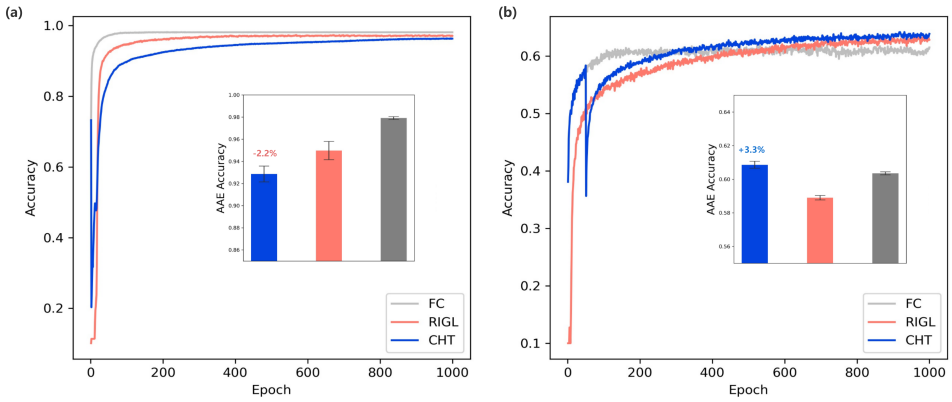


Figure 5: Comparison between CHT, RigL and FC. The figure reports the accuracy curve of CHT, RigL and FC in datasets: **(a)** MNIST, 1-epoch-KSP on 1000 epochs (the result with 50-epochs-KSP is in suppl. info.); **(b)** CIFAR10, 50-epoch-KSP on 1000 epochs. Each panel contains an embedded subpanel showing the AAE-accuracy value of each algorithm (mean and standard error over 3 repetitions). The percentual increment/decrement in the embedded subpanel is between CHT and the RigL.

in CIFAR10). The reason can be extracted from Fig. 3 in the MNIST dataset, images contain representations of numbers around the central area. The information content is therefore mostly distributed around the central area, which deserves more connections than neuron units associated to peripheral area. In Fig. 3, we show the number of connections (node degree) of each input unit after 10, 20 and 50 epochs, and we highlight that ESDL (based on CH3-L3) can evolve a more informative sub-network faster than SET.

In Fig. 4a we show the three steps of the CHT process: KSP, epitopological prediction and weight refinement. Fig. 4b shows the overlap rate between new links and removed links for each layer, highlighting the necessity of applying an early-stop mechanism, since the rate approaches a threshold of 0.9 after only 20-40 epochs. In Fig. 3(c,d) we compare CHT and ESDL (based on SET initialization and CH3-L3 prediction as in Fig. 2), validating the importance of adopting the KSP before the epitopological prediction. In MNIST, thanks to 1-epoch-KSP, the CHT reaches a high accuracy much faster than ESDL (+27.7% in AAE), but the final epoch accuracy is comparable. Results of CHT for 50-epoch-KSP are in suppl. info. In CIFAR10, CHT for 50-epoch-KSP reaches a high accuracy much faster than ESDL (+22.1% in AAE) and also the final epoch accuracy is remarkably higher.

As final analysis, we show the comparison of CHT, RigL and fully connected (FC) in Fig. 5. As commented in the Method section in suppl. info., RigL is a typical pseudo-sparse realization. However, CHT with only 1-epoch-KSP (the result with 50-epoch-KSP is in suppl. Info) shows a comparable final epochs performance in MNIST (Fig. 5a) and a better learning speed in CIFAR10 (50-epocks-KSP used, Fig. 5b) , meaning that CHT is promising because it is not inferior to the current SOTA algorithm RigL despite using less information. Indeed, CHT is the first method to add new links (after removal) exploiting the mere structural information of the sparse sandwich subnetwork topology, whereas RigL is pseudo-sparse because it exploits the entire gradient information of the fully connected ANN.

Table 1: The hyperparameter set for each figure

| | sparsity | learning rate | batch size | update interval | zeta | dropout |
|---|---|---|---|---|---|---|
| Fig. 2 MNIST | 0.99 | 0.0001 | 32 | 1 | 0.3 | 0 |
| Fig. 2 CIFAR10 | 0.88 | 0.01 | 128 | 50 | 0.3 | 0.3 |
| Fig. 4 MNIST | 0.99 | 0.0001 | 32 | 5 | 0.3 | 0 |
| Fig. 4 CIFAR10 | 0.99 | 0.01 | 128 | 10 | 0.3 | 0.3 |
| Fig. 5 MNIST | 0.99 | 0.0001 | 32 | 5 | 0.3 | 0 |
| Fig. 5 CIFAR10 | 0.99 | 0.01 | 128 | 10 | 0.3 | 0.3 |

## 5 Conclusion

By translating the brain-inspired paradigm of epitopological learning from complex network intelligence theory into sparse training field, we introduce CHT: an algorithm for epitopological sparse deep training of neural networks. CHT has three phases. It is initialized with kick start pruning, to hint the link predictors; progresses with epitopological learning prediction, to shape the ANN topology; and finalizes with weight refinement, to tune the synaptic weights values. Empirical validations on classification of computer vision datasets show the efficacy of each of these three steps and CHT in comparison to previous ST baseline and state-of-the-art algorithms. CHT represents a completely new direction to implement sparse training. While SET leverages random evolution to progress towards a scale-free topology and RigL adopts gradient information to suggest how to update the connections, CHT is the first algorithm in sparse training that learns to shape sparsity by using the topological organization of the ANN. In brief, we could simplify saying that CHT is the first to predict sparsity using sparsity.

## References

[1] Shiwei Liu, Lu Yin, Decebal Constantin Mocanu, and Mykola Pechenizkiy. Do we actually need dense over-parameterization? in-time over-parameterization in sparse training. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 6989–7000. PMLR, 2021.

[2] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015.

[3] Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[4] Wenlin Chen, James T. Wilson, Stephen Tyree, Kilian Q. Weinberger, and Yixin Chen. Compressing neural networks with the hashing trick. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2285–2294. JMLR.org, 2015.

[5] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[6] Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. Rigging the lottery: Making all tickets winners. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 2943–2952. PMLR, 2020.

[7] Decebal Constantin Mocanu, Elena Mocanu, Peter Stone, Phuong H Nguyen, Madeleine Gibescu, and Antonio Liotta. Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nature communications*, 9(1):1–12, 2018.

[8] Hesham Mostafa and Xin Wang. Parameter efficient training of deep convolutional neural networks by dynamic sparse reparameterization. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 4646–4655. PMLR, 2019.

[9] Shiwei Liu, Tianlong Chen, Xiaohan Chen, Zahra Atashgahi, Lu Yin, Huanyu Kou, Li Shen, Mykola Pechenizkiy, Zhangyang Wang, and Decebal Constantin Mocanu. Sparse training via boosting pruning plasticity with neuroregeneration. pages 9908–9922, 2021.

 doi:10.20944/preprints202207.0139.v1

[10] Warren S. McCulloch and Walter H. Pitts. A logical calculus of the ideas immanent in nervous activity. In Margaret A. Boden, editor, *The Philosophy of Artificial Intelligence*, Oxford readings in philosophy, pages 22–39. Oxford University Press, 1990.

[11] Donald Hebb. The organization of behavior. emphnew york, 1949.

[12] Gregorio Alanis-Lobato Carlo Vittorio Cannistraci and Timothy Ravasi. From link-prediction in brain connectomes and protein interactomes to the local-community-paradigm in complex networks. *Sci Rep*, 3(1613), 2013.

[13] Carlo Vittorio Cannistraci. Modelling self-organization in complex networks via a brain-inspired network automata theory improves link reliability in protein interactomes. *Sci Rep*, 8(1):2045–2322, 10 2018.

[14] Vaibhav et al Narula. Can local-community-paradigm and epitopological learning enhance our understanding of how local brain connectivity is able to process, learn and memorize chronic pain? *Applied network science*, 2(1), 2017.

[15] Yann LeCun, John S. Denker, and Sara A. Solla. Optimal brain damage. In David S. Touretzky, editor, *Advances in Neural Information Processing Systems 2, [NIPS Conference, Denver, Colorado, USA, November 27-30, 1989]*, pages 598–605. Morgan Kaufmann, 1989.

[16] Kenneth O. Stanley and Risto Miikkulainen. Evolving neural networks through augmenting topologies. *Evolutionary Computation*, 10(2):99–127, 2002.

[17] A. Muscoloni, U. Michieli, and C.V. Cannistraci. Adaptive network automata modelling of complex networks. *preprints*, 2020.

[18] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86(11):2278–2324, 1998.

[19] Alex Krizhevsky. Learning multiple layers of features from tiny images. pages 32–33, 2009.

[20] Song Han, Jeff Pool, John Tran, and William J. Dally. Learning both weights and connections for efficient neural network. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 1135–1143, 2015.

[21] Trevor Gale, Erich Elsen, and Sara Hooker. The state of sparsity in deep neural networks. *CoRR*, abs/1902.09574, 2019.

[22] Alex Renda, Jonathan Frankle, and Michael Carbin. Comparing rewinding and fine-tuning in neural network pruning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[23] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[24] Decebal Constantin Mocanu, Elena Mocanu, Phuong H. Nguyen, Madeleine Gibescu, and Antonio Liotta. A topological insight into restricted boltzmann machines. *Machine Learning*, 104(2):243–270, September 2016.

[25] Graham H Diering, Raja S Nirujogi, Richard H Roth, Paul F Worley, Akhilesh Pandey, and Richard L Huganir. Homer1a drives homeostatic scaling-down of excitatory synapses during sleep. *Science*, 355(6324):511–515, 2017.

[26] Tao Zhou. Progresses and challenges in link prediction. *CoRR*, abs/2102.11472, 2021.

[27] Tao Zhou, Yan-Li Lee, and Guannan Wang. Experimental analyses on 2-hop-based and 3-hop-based link prediction algorithms. *Physica A: Statistical Mechanics and its Applications*, 564:125532, 2021.

 doi:10.20944/preprints202207.0139.v1

[28] Simone Daminelli, Josephine Maria Thomas, Claudio Durán, and Carlo Vittorio Cannistraci. Common neighbours and the local-community-paradigm for topological link prediction in bipartite networks. *New Journal of Physics*, 17(11):113037, nov 2015.

[29] Claudio Durán, Simone Daminelli, Josephine M Thomas, V Joachim Haupt, Michael Schroeder, and Carlo Vittorio Cannistraci. Pioneering topological methods for network-based drug–target prediction by exploiting a brain-network self-organization theory. *Briefings in Bioinformatics*, 19(6):1183–1202, 04 2017.

[30] Mark EJ Newman. Clustering and preferential attachment in growing networks. *Physical review E*, 64(2):025102, 2001.

[31] Nazanin Movarraei and Maruti Shikare. On the number of paths of lengths 3 and 4 in a graph. *International Journal of Applied Mathematical Research*, 3(2):178–189, 2014.

[32] Linyuan Lü, Liming Pan, Tao Zhou, Yi-Cheng Zhang, and H. Eugene Stanley. Toward link predictability of complex networks. *Proceedings of the National Academy of Sciences*, 112(8):2325–2330, 2015.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 July 2022 doi:10.20944/preprints202207.0139.v1

# Supplementary Information

## A Methods

### A.1 Sparse Training Algorithms

#### A.1.1 SET (baseline algorithm)

SET[7] is the first DST-based algorithm introduced in the field of ANNs. Differently from Pruning, SET makes the evolution of sparse neural networks more dynamic by continuously exploring the network structure. SET follows a 4-step procedure: (1) initialize the network topology at random according to the ER model; (2) train the model for one epoch; (3) remove zeta percentage of links with the lowest absolute weights and randomly add the same number of links; (4) repeat steps 2-3 until the pre-set number of training epochs has been reached. 'zeta' represents the fraction of existing links removed/added in each evolutionary epoch. As an easy-to-implement algorithm, SET is also effective in practice. In addition, Mocanu et al.[7] demonstrates that the topology of the sandwich layers gradually evolves to a scale-free network organization after convergence of the DST procedure.

#### A.1.2 RigL (state of the art algorithm)

RigL[6] follows a similar procedure to SET, but implements a different technique for growing new links. In order to maximize the use of the training information, RigL activates as new links the currently nonobserved links with the highest absolute values of instantaneous gradient. Besides, RigL applies two mechanisms to promote the performance of ST. It stretches the interval of the structure evolution to extend the learning time of the weight values, which results in a more rational pruning of the corresponding weighted links. In addition, RigL applies a decay function of $\alpha$, referred to 'zeta' in this paper, to make the model focusing on the weights in the late period of training. Given the excellent performance of RigL in various datasets[6], we consider it as the SOTA method in comparison with the CHT technique introduced in this study.

### A.2 Link Prediction Methods

#### A.2.1 Link prediction on bipartite sandwich layers

In this study, we adopt three link prediction methods (CH3-L3, CN-L3, SPM) that can be used both on monopartite and bipartite networks. Here, we exploit a particular pipeline for using them on bipartite networks. Let's consider a bipartite sandwich layer composed of N1 nodes in the first layer and N2 nodes in the second layer, with corresponding binary adjacency matrix *B* of size [*N1, N2*] such that $B_{u,v} = 1$ if there is a connection between nodes *u* and *v* and $B_{u,v} = 0$ otherwise. Given *B*, we build an equivalent monopartite adjacency matrix *X* of size [*N, N*] composed of all $N = N1 + N2$ nodes from both sandwich layers, and with connections between the same node pairs that are also connected in the sandwich layer. The monopartite adjacency matrix *X* is given in input to the monopartite link prediction methods, which provide in output likelihood scores for all possible $\frac{N*(N-1)}{2}$ monopartite undirected node pairs. Out of all these pairs, the $N1 * N2$ bipartite node pairs are extracted and they represent the actual result of the bipartite link prediction process on the sandwich layer.

#### A.2.2 CH3-L3

The Cannistraci-Hebb (CH) theory has been introduced as a revision of the local-community-paradigm (LCP)[12, 13, 28, 29] theory and it has been formalized within the framework of network automata[17]. While the LCP paradigm emphasized the importance to complement the information related to the common neighbours with the interactions between them (internal local-community-links), the CH rule is based on the local isolation of the common neighbours by minimizing their interactions external to the local community (external local-community-links). In particular, Cannistraci-Hebb (CH) network automata on paths of length n are all the network automata models that explicitly consider the minimization of the external local-community-links within a local community characterized by paths of length n[17]. In this study, we consider the CH network automaton CH3 on paths of length 3 (CH3-L3), whose mathematical formula is:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 July 2022 doi:10.20944/preprints202207.0139.v1

$$CH3\_L3(u,v) = \sum_{z_1,z_2 \in L3} \frac{1}{\sqrt{(1+de_{z_1}) * (1+de_{z_2})}} \tag{1}$$

where: *u* and *v* are the two seed nodes of the candidate interaction; $z_1$, $z_2$ are the intermediate nodes on the considered path of length three; $de_{z_1}, de_{z_2}$ are the respective external node degrees; and the summation is executed over all the paths of length three. Node pairs with tied CH3-L3 score are sub-ranked according to the associated shortest paths correlation (SPcorr) score, as described in the original study[17].

#### A.2.3 CN-L3

A standard baseline method adopted in link prediction studies is the common neighbors (CN) index[30]. Considered in the perspective of network automata on paths of length 2, the CN rule corresponds to the number of paths of length 2 (CN-L2). In this study, which is focused on bipartite networks (having no paths of length 2 between node pairs) as suggested in previous study[28], we rely on path of length 3 to define CN. Therefore, we adopt as baseline predictor a rule termed CN-L3, corresponding to the number of paths of length 3 between node pairs.

The mathematical formula[31] is:

$$CN\_L3(u,v) = (X^3)_{u,v} - X_{u,v} * (d_u + d_v - 1) \tag{2}$$

where: *u* and *v* are the two seed nodes of the candidate interaction; $d_u, d_v$ are the respective node degrees; $X$ is the binary adjacency matrix of the network.

#### A.2.4 Structural Perturbation Method (SPM)

The structural perturbation method (SPM) relies on a theory similar to the first-order perturbation in quantum mechanics[32]. A high-level description of the procedure is the following: (1) randomly remove 10% of the links from the network adjacency matrix $X$, obtaining a reduced network $X' = X - R$, where R is the set of removed links; (2) compute the eigenvalues and eigenvectors of $X'$; (3) considering the set of links $R$ as a perturbation of $X'$, construct the perturbed matrix $X^P$ via a first-order approximation that allows the eigenvalues to change while keeping fixed the eigenvectors; (4) repeat steps 1-3 for 10 independent iterations and take the average of the perturbed matrices $X^P$. The link prediction result is given by the values of the average perturbed matrix, which represent the scores for each node pair. The higher the score the greater the likelihood that the interaction exists. The idea behind the method is that a missing part of the network is predictable if it does not significantly change the structural features of the observable part, represented by the eigenvectors of the matrix. If this is the case, the perturbed matrices should be good approximations of the original network[32].

### A.3 Purely sparse operations for sparse training

Most of the sparse training implementations are actually not purely sparse. It is often used a binary 'mask' matrix to store the adjacency matrix of the sandwich layer, which is multiplied element-wise with the weight matrix in order to perform pseudo-sparse operations. In this article, we consider important to implement purely sparse operations across the entire training period, which allows for scalability of the implementation to large networks. We adopt 3 lists in order to store the information of ESDL during the training period:

- **position list:** it stores the positions of all the existing links with corresponding indexes of input units and output units.
- **weight list:** it stores the weight values associated to existing links.
- **gradient list:** it stores the gradient information associated to existing links.

Since ESDL keeps the sparsity of each sandwich layer constant during training, the three lists have the same fixed length, equal to the number of existing links. The information content of the three lists is matched, meaning that for each list the element at a given index is associated to the same

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 July 2022 doi:10.20944/preprints202207.0139.v1

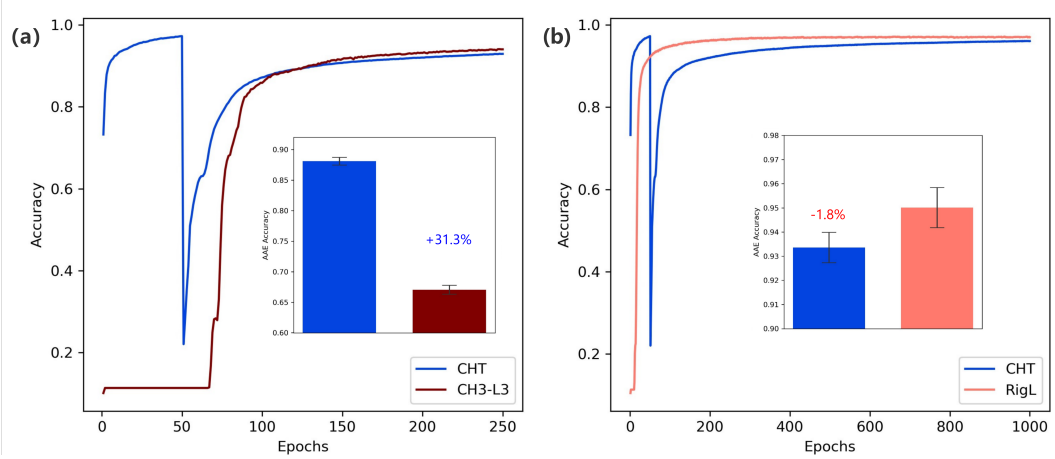


Figure 6: CHT for 50-epochs-KSP. **(a)** The panel reports the accuracy curve of CHT for 50-epochs-KSP and CH3-L3 in MNIST dataset (250 epochs). **(b)** The panel reports the accuracy curve of CHT for 50-epochs-KSP and RigL in MNIST dataset (1000 epochs). Each panel contains an embedded subpanel showing the AAE-accuracy value of each algorithm (mean and standard error over 3 repetitions). The percentual increment/decrement of CHT for 50-epochs-KSP is reported in **(a,b)**.

existing link. At every structural evolution step, the information related to the new links (position, weight, gradient) will be introduced at the indexes of the removed links, therefore replacing the old information. This realization makes the total training period with purely sparse operations, which reduces the computational time and space requirements of ESDL.

### A.4 AC-Score for performance evaluation

In this study we introduce the AC-score as an additional metric to evaluate the performance of each algorithm in classification tasks. Given the number of actual positives ($P$), actual negatives ($N$), true positives ($TP$) and true negatives ($TN$), the formula of the AC-score is as follow:

$$AC - Score = \frac{2 * \frac{TP}{P} * \frac{TN}{N}}{\frac{TP}{P} + \frac{TN}{N}} \tag{3}$$

AC-Score effectively compensates for the shortcoming of other evaluation measures. With respect to the F1-score, it considers also the influence of $TN$, which makes the evaluation more comprehensive. In addition, AC-Score overcomes the unsuitability of accuracy in imbalanced datasets.

## B Datasets

### B.1 MNIST

MNIST[18] is a computer vision dataset consisting of 60,000 train samples and 10,000 test samples. The labels include a total of 10 categories, corresponding to the numbers ‘0’ to ‘9’. Each image has been normalized and centered in the image, which has a fixed size (28x28 pixels) and a value from 0 to 1. As the input layer for MLP, all the samples are flattened and converted to a 1D array of 784 features. We set the dimensions of each sandwich layer of the MLP to 784x1024x1024x1024x10 for MNIST. The first three sandwich layers are sparse, while the last layer is fully connected.

### B.2 CIFAR10

CIFAR10[19] is also a computer vision dataset consisting of 50,000 train samples and 10,000 test samples. The samples are divided into 10 categories, including some daily objects and animals. We use the same data augmentation method as SET[7]. As the input layer for MLP, each sample has a fixed size (3x32x32) and is flattened to a 1D array containing 3072 features. We set the dimensions

 doi:10.20944/preprints202207.0139.v1

of each sandwich layer of the MLP to 3072x4000x1000x4000x10 for CIFAR10. The first three sandwich layers are sparse, while the last layer is fully connected.