# Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science

*Citation for published version (APA):*
Mocanu, D. C., Mocanu, E., Stone, P., Nguyen, H. P., Gibescu, M., & Liotta, A. (2018). Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nature Communications*, *9*(1), [2383]. https://doi.org/10.1038/s41467-018-04316-3

*DOI:*
10.1038/s41467-018-04316-3

*Document status and date:*
Published: 19/06/2018

*Document Version:*
Author's version before peer-review

***Please check the document version of this publication:***

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

*General rights*
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

*Take down policy*
If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Download date: 27. Aug. 2022

arXiv:1707.04780v1 [cs.NE] 15 Jul 2017

# Evolutionary Training of Sparse Artificial Neural Networks: A Network Science Perspective

**Decebal Constantin Mocanu · Elena Mocanu · Peter Stone · Phuong H. Nguyen · Madeleine Gibescu · Antonio Liotta**

**This article is a pre-print version!**

**Abstract** Through the success of deep learning, Artificial Neural Networks (ANNs) are among the most used artificial intelligence methods nowadays. ANNs have led to major breakthroughs in various domains, such as particle physics, reinforcement learning, speech recognition, computer vision, and so on. Taking inspiration from the network properties of biological neural networks (e.g. sparsity, scale-freeness), we argue that (contrary to general practice) Artificial Neural Networks (ANN), too, should not have fully-connected layers. We show how ANNs perform perfectly well with sparsely-connected layers. Following a Darwinian evolutionary approach, we propose a novel algorithm which evolves an initial random sparse topology (i.e. an Erdős-Rényi random graph) of two consecutive layers of neurons into a scale-free topology, during the ANN training process. The resulting sparse layers can safely replace the corresponding fully-connected layers. Our method allows to quadratically reduce the number of parameters in the fully conencted layers of ANNs, yielding quadratically faster computational times in both phases (i.e. training and inference), at no decrease in accuracy. We demonstrate our claims on two popular ANN types (restricted Boltzmann machine and multi-layer perceptron), on two types of tasks (supervised and unsupervised learning), and on 14 benchmark datasets. We anticipate that our approach will enable ANNs having billions of neurons and evolved topologies to be capable of handling complex real-world tasks that are intractable using state-of-the-art methods.

Decebal Constantin Mocanu
Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven, the Netherlands
E-mail: d.c.mocanu@tue.nl

Elena Mocanu
Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven, the Netherlands

Peter Stone
Department of Computer Science, The University of Texas at Austin, Austin, USA

Phuong H. Nguyen
Department of Electrical Engineering, Eindhoven University of Technology, the Netherlands

Madeleine Gibescu
Department of Electrical Engineering, Eindhoven University of Technology, the Netherlands

Antonio Liotta
Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven, the Netherlands

**Keywords** complex networks · evolutionary algorithms · deep learning · sparse artificial neural networks · restricted Boltzmann machines

## 1 Introduction

Through the success of deep learning (LeCun et al, 2015), Artificial Neural Networks (ANNs) are among the most used artificial intelligence methods nowadays. ANNs have led to major breakthroughs in various domains, such as particle physics (Baldi et al, 2014), reinforcement learning (Mnih et al, 2015), speech recognition, and so on (LeCun et al, 2015). Typically, ANNs have layers of fully-connected neurons (LeCun et al, 2015), which contain most of the network parameters (i.e. the weighted connections), leading to a quadratic number of connections with respect to their number of neurons. In turn, the network size is severely limited, due to obvious computational limitations.

By contrast to ANNs, biological neural networks have been demonstrated to have a sparse (rather than dense) topology (Strogatz, 2001; Pessoa, 2014), and also hold other important properties that are instrumental to learning efficiency. These have been extensively studied in (Bullmore and Sporns, 2009) and include scale-freeness (Barabási and Albert, 1999) and small-worldness (Watts and Strogatz, 1998). Nevertheless, ANNs have not evolved to mimic these topological features (Mocanu, 2016; Mocanu et al, 2016), which is why in practice they lead to extremely large models. Previous studies have demonstrated that, following the training phase, ANN models end up with weights histograms that peak around zero (Dieleman and Schrauwen, 2012; Yosinski and Lipson, 2012; Han et al, 2015). Moreover, in our previous work (Mocanu et al, 2015), we have hinted a similar fact. Yet, in the machine learning state-of-the-art, sparse topological connectivity is pursued only as an aftermath of the training phase (Han et al, 2015), which bears benefits only during the inference phase.

We claim that topological sparsity must be pursued since the ANN design phase, which leads to a substantial reduction in connections and, in turn, to memory and computational efficiency. At the same time, to be able to make use of standard training algorithms, e.g. Stochastic Gradient Descent (SGD), the structured multi-layer architecture of ANNs has to be preserved. Otherwise we would not be able to train large ANNs with a complete random sparse topology, due to the difficulty of finding suitable optimization algorithms.

In a recent paper, we introduced compleX Boltzmann Machines (XBMs), a sparse variant of Restricted Boltzmann Machines (RBMs), conceived with a sparse scale-free topology (Mocanu et al, 2016). XBMs outperform their fully-connected RBMs counterparts and are much faster, both in the training and the inference phases. Yet, being based on a fixed sparsity pattern, XBMs may fail to properly model the data distribution. To overcome this limitation, in this paper we introduce a Sparse Evolutionary Training (SET) procedure, which takes into consideration data distributions and creates sparse bipartite layers suitable to replace the fully-connected bipartite layers in any type of ANNs.

SET follows the natural simplicity of the Darwinian evolutionary approach, which was explored successfully in our previous work on evolutionary function approximation (Whiteson and Stone, 2006). Also, it has been explored for network connectivity in McDonnell and Waagen (1993), and for the layers architecture of deep neural networks (Miikkulainen et al, 2017). The bipartite ANN layers start from a random sparse topology (i.e. Erdős-Rényi random graph (Erdős and Rényi, 1959)), evolving through a random process during the training phase towards a scale-free topology. Remarkably, this process does not have to incorporate any constraints to force the scale-free topology. But our evolutionary algorithm is not arbitrary: it follows a phenomenon that takes place in real-world complex networks (such as biological neural networks, and protein interaction networks). Starting from an Erdős-Rényi random graph topology and throughout millenia of natural evolution, networks end up with a more structured connectivity, i.e. scale-free (Barabási and Albert, 1999) or small-world (Watts and Strogatz, 1998) topologies.

The remainder of this paper is organized as follows. Section 2 presents background knowledge mainly for the benefit of the less specialist reader. Section 3 introduces the proposed method, SET. Section 4 describes the experiments performed and discusses the results. Finally, Section 5 concludes the chapter and proposes future research directions.

## 2 Background

### 2.1 Artificial neural networks.

Artificial Neural Networks (Bishop, 2006) are mathematical models, inspired by biological neural networks, which can be used in all three machine learning paradigms (i.e. supervised learning (Hastie et al, 2001), unsupervised learning (Hastie et al, 2001), and reinforcement learning (Sutton and Barto, 1998)). These make them very versatile and powerful, as quantifiable by the remarkable success registered recently by the last generation of ANNs (also known as deep artificial neural networks or deep learning (LeCun et al, 2015)) in many fields from computer vision (LeCun et al, 2015) to gaming (Mnih et al, 2015; Silver et al, 2016). Just like their biological counterparts, ANNs are composed by neurons and weighted connections between these neurons. Based on their purposes and architectures, there are many models of ANNs, such as restricted Boltzmann machines (Smolensky, 1987), multi layer perceptron (Rosenblatt, 1962), convolutional neural networks (LeCun et al, 1998), recurrent neural networks (Graves et al, 2009), and so on. Many of these ANN models contain fully-connected layers. A fully-connected layer of neurons means that all its neurons are connected to all the neurons belonging to its adjacent layer in the ANN architecture. For the purpose of this paper, in this section we briefly describe two models that contain fully-connected layers, i.e. Restricted Boltzmann Machines (Smolensky, 1987) and multi layer perceptron (Rosenblatt, 1962).

A restricted Boltzmann machine is a two-layer, generative, stochastic neural network that is capable to learn a probability distribution over a set of inputs (Smolensky, 1987) in an unsupervised manner. From a topological perspective, it allows only interlayer connections. Its two layers are: the visible layer, in which the neurons represent the input data; and the hidden layer, in which the neurons represent the features automatically extracted by the RBM model from the input data. Each visible neuron is connected to all hidden neurons through a weighted undirected connection, leading to a fully-connected topology between the two layers. Thus, the flow of information is bidirectional in RBMs, from the visible layer to the hidden layer, and from the hidden layer to the visible layer, respectively. RBMs, beside being very successful in providing very good initialization weights to the supervised training of deep artificial neural network architectures (Hinton et al, 2006), are also very successful as stand alone models in a variety of tasks, such as density estimation to model human choice (Osogami and Otsuka, 2014), collaborative filtering (Salakhutdinov et al, 2007), information retrieval (Gehler et al, 2006), multi-class classification (Larochelle and Bengio, 2008), and so on.

Multi Layer Perceptron (Rosenblatt, 1962) (MLP) is a classical feed-forward ANN model that maps a set of input data to the corresponding set of output data. Thus, it is used for supervised learning. It is composed by an input layer in which the neurons represent the input

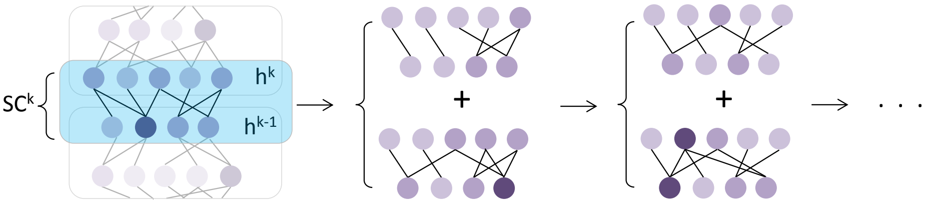

Fig. 1: An illustration of the SET procedure. For each Sparse Connected layer, SC$^k$ (left plot), of an ANN at the end of a training epoch a fraction of the weights, the ones closest to zero, are removed (top middle plot). Then, new weighs are added randomly in the same amount as the ones previously removed (bottom middle plot). Further on, a new training epoch is performed (right plot), and the procedure to remove and add weights is repeated. The process continues for a finite number of training epochs, as usual in the ANNs training.

data, an output layer in which the neurons represent the output data, and an arbitrary number of hidden layers in between, with neurons representing the hidden features of the input data (to be automatically discovered). The flow of information in MLPs is unidirectional, starting from the input layer towards the output layer. Thus, the connections are unidirectional and exist just between consecutive layers. Any two consecutive layers in MLPs are fully-connected. There are no connections between the neurons belonging to the same layer, or between the neurons belonging to layers which are not consecutive. In (Cybenko, 1989), it has been demonstrated that MLPs are universal function approximators, so they can be used to model any type of regression or classification problems.

In general, working with ANN models involves two phases: 1) training (or learning), in which the weighted connections between neurons are optimized using various algorithms (e.g. backpropagation procedure combined with stochastic gradient descent (Rumelhart et al, 1986; Bottou and Bousquet, 2008) used in MLPs, contrastive divergence (Hinton, 2002) used in RBMs) to minimize a loss function defined by their purpose; and 2) inference, in which the optimized ANN model is used to fulfill its purpose.

### 2.2 Scale-free complex networks.

Complex networks (e.g. biological neural networks, actors and movies, power grids, transportation networks) are everywhere, in different forms, and different fields (from neurobiology to statistical physics (Strogatz, 2001)). Formally, a complex network is a graph with non-trivial topological features, human- or nature-made. One of the most well-known and deeply studied type of topological features in complex networks is scale-freeness, due to the fact that a wide range of real-world complex networks have this topology. A network with a scale-free topology (Barabási and Albert, 1999) is a sparse graph (Del Genio et al, 2011) that approximately has a power-law degree distribution $P(d) \sim d^{-\gamma}$, where the fraction $P(d)$ from the total nodes of the network has $d$ connections to other nodes, and the parameter $\gamma$ usually stays in the range $\gamma \in (2, 3)$

## 3 Sparse Evolutionary Training (SET)

SET is detailed in Algorithm 1, and exemplified in Figure 1. Formally, let us defined a Sparse Connected ($SC^k$) layer in an ANN. This layer has $n^k$ neurons, collected in a vector $\mathbf{h}^k = [h_1^k, h_2^k, ..., h_{n^k}^k]$. Any neuron from $\mathbf{h}^k$ is connected to an arbitrary number of neurons belonging to layer below $\mathbf{h}^{k-1}$. The connections between the two layers are collected in a sparse weight matrix $\mathbf{W}^k \in \mathbf{R}^{n^{k-1} \times n^k}$. Initially, $\mathbf{W}^k$ is a Erdős-Rényi random graph, in which the probability of the connection between the neuron $h_i^k$ and $h_j^{k-1}$ to exist is given by:

$$p(W_{ij}^k) = \frac{\epsilon(n^k + n^{k-1})}{n^k n^{k-1}} \tag{1}$$

whereby $\epsilon \in \mathbf{R}^+$ is a parameter of SET controlling the sparsity level. If $\epsilon \ll n^k$ and $\epsilon \ll n^{k+1}$ then there is a linear number of connections (i.e. non-zero elements), $n^W = |\mathbf{W}^k| = \epsilon(n^k + n^{k-1})$, with respect to the number of neurons in the sparse layers. In the case of fully-connected layers the number of connections is quadratic, i.e. $n^k n^{k-1}$.

```
%Initialization;
initialize ANN model;
set $\epsilon$ and $\zeta$;
for each bipartite fully-connected (FC) layer of the ANN do
    replace FC with a Sparse Connected (SC) layer having a Erdős-Rényi topology given by $\epsilon$ and
      Eq.1;
end
initialize training algorithm parameters;
%Training;
for each training epoch $e$ do
    perform standard training procedure;
    perform weights update;
    for each bipartite SC layer of the ANN do
        remove a fraction $\zeta$ of the smallest positive weights;
        remove a fraction $\zeta$ of the highest negative weights;
        if $e$ is not the last training epoch then
            add randomly new weights (connections) in the same amount as the ones removed
              previously;
        end
    end
end
```

**Algorithm 1:** SET pseudocode

However, it may be that this random generated topology is not suited to the particularities of the data that the ANN model tries to learn. To overcome this situation, during the training process, after each training epoch, a fraction $\zeta$ of the smallest positive weights and of the highest negative weights of $SC^k$ is removed. These removed weights are the ones closest to zero, thus we do not expect that their removal will notably change the model performance (Han et al, 2015). Next, to let the topology of $SC^k$ to evolve so as to fit the data, an amount of new random connections, equal to the amount of weights removed previously, is added to $SC^k$. In this way, the number of connections in $SC^k$ remains constant during the training process. After the training ends, we keep the topology of $SC^k$ as the one obtained after the last weight removal step, without adding new random connections. Please note that the removal of the not important connections corresponds to the selection phase of natural

Table 1: Datasets characteristics. The data used in this paper have been chosen to cover a wide range of fields where ANNs have the potential to advance state-of-the-art, including biology, physics, computer vision, data mining, and economics.

| Experiments type | Dataset | | Dataset Properties | | | | |
|---|---|---|---|---|---|---|---|
| | | | Domain | Data type | Features [#] | Train samples [#] | Test samples[#] |
| Assessment of RBMs variants | UCI evaluation suite (Larochelle and Murray, 2011) | ADULT | households | binary | 123 | 5000 | 26147 |
| | | Connect4 | games | binary | 126 | 16000 | 47557 |
| | | DNA | genetics | binary | 180 | 1400 | 1186 |
| | | Mushrooms | biology | binary | 112 | 2000 | 5624 |
| | | NIPS-0-12 | documents | binary | 500 | 400 | 1240 |
| | | OCR-letters | letters | binary | 128 | 32152 | 10000 |
| | | RCV1 | documents | binary | 150 | 40000 | 150000 |
| | | Web | Internet | binary | 300 | 14000 | 32561 |
| | CalTech 101 | 16x16 | images | binary | 256 | 4082 | 2302 |
| | Silhouettes (Marlin et al, 2010) | 28x28 | images | binary | 784 | 4100 | 2307 |
| | MNIST | | digits | binary | 784 | 60000 | 10000 |
| Assessment of MLPs variants | MNIST | | digits | grayscale | 784 | 60000 | 10000 |
| | CIFAR10 | | images | RGB colors | 3072 | 50000 | 10000 |
| | HIGGS (Baldi et al, 2014) | | particle physics | real values | 28 | 10500000 | 500000 |

evolution (which typically is not a random process), while the the random addition of new connections corresponds to the mutation phase of natural evolution (which typically is a random process).

It is worth highlighting that in the initial phase of conceiving the SET procedure, the weight-removal and weight-addition steps after each training epoch were introduced intuitively. Still, in the last phases of preparing this paper we have found that there is a similarity between SET and a phenomenon which takes place in biological brains, named *synaptic shrinking during sleep*. This phenomenon has been demonstrated recently in two papers published in the Science journal in February 2017 (Diering et al, 2017; de Vivo et al, 2017). In short, it was found that during sleep the weakest synapses in the brain shrink, while the strongest synapses remain unaltered, supporting the hypothesis that one of the core functions of sleeping is to renormalize the overall synaptic strength increased while awake (de Vivo et al, 2017). By keeping the analogy, this is - in a way - what happens also with the ANNs during the SET procedure.

## 4 Experiments and results

### 4.1 Evaluation method.

We evaluate SET in two types of ANNs, restricted Boltzmann machine (Smolensky, 1987), and Multi Layer Perceptron (MLP) (LeCun et al, 2015), to experiment with both unsupervised and supervised learning. In total, we evaluate SET on 14 benchmark datasets, as detailed in Table 1, covering a wide range of fields in which ANNs are employed, such as biology, physics, computer vision, data mining, and economics. We also assess SET in combination with two different training methods, i.e. contrastive divergence (Hinton, 2002) and stochastic gradient descent (LeCun et al, 2015).

### 4.2 SET performance on restricted Boltzmann machines.

First, we have analyzed the performance of SET on a bipartite undirected stochastic ANN model, i.e. restricted Boltzmann machine (Smolensky, 1987), which is popular for its unsupervised learning capability (Bengio, 2009) and high performance as a feature extractor and density estimator (Osogami and Otsuka, 2014). The new model derived from the SET procedure was dubbed SET-RBM. In all experiments, we set $\epsilon = 11$, and $\zeta = 0.3$, performing a small random search just on the MNIST dataset, to be able to assess if these two meta-parameters are dataset specific or if their values are general enough to perform well also on different datasets.

There are few studies on RBM connectivity sparsity (Mocanu et al, 2016). Still, to get a good estimation of SET-RBM capabilities we compared it against RBM$_{FixProb}$ (Mocanu et al, 2016) (a sparse RBM model with a fixed Erdős-Rényi topology), fully-connected RBMs, and with the state-of-the-art results of XBMs from (Mocanu et al, 2016). We performed experiments on 11 benchmark datasets coming from various domains, as depicted in Table 1, using the same splitting for training and testing data as in (Mocanu et al, 2016). All models were trained for 5000 epochs using Contrastive Divergence (Hinton, 2002) (CD) with 1, 3, and 10 CD steps, a learning rate of 0.01, a momentum of 0.9, and a weight decay of 0.0002, as discussed in (Hinton, 2012). We evaluated the generative performance of the scrutinized models by computing the log-probabilities on the test data using Annealed Importance Sampling (AIS) (Salakhutdinov and Murray, 2008), setting all parameters as in (Mocanu et al, 2016; Salakhutdinov and Murray, 2008). We have used MATLAB for this set of experiments. We implemented SET-RBM and RBM$_{FixProb}$ ourselves; while for RBM and AIS we have adapted the code provided by (Salakhutdinov and Murray, 2008).

Figure 2 depicts the model's performance on all datasets, using varying numbers of hidden neurons; while Table 2 summarizes the results, presenting the best performer for each type of model for each dataset. In 7 out of 11 datasets, SET-RBM outperforms the fully-connected RBMs, while reducing the parameters by a few orders of magnitude. For instance, on the MNIST dataset, SET-RBM reaches -86.41 *nats*, with a 5.29-fold improvement over the fully-connected RBM, and a parameters reduction down to 2%. In 10 out of 11 datasets, SET-RBM outperforms XBM, which represents the state-of-the-art results on these datasets for sparse variants of RBM (Mocanu et al, 2016).

Figure 2 shows striking results on stability. While fully-connected RBMs show instability and over-fitting issues, the SET procedure stabilizes SET-RBMs and avoids over-fitting. This situation can be observed more often when a high number of hidden neurons is chosen (columns 2, 3, 5, 6, 8, and 9 of Figure 2). For instance, if we look at the DNA dataset, independently on the values of $n^h$ and $n^{CD}$ (Figure 2, third row), we may observe that SET-RBMs are very stable after they reach around -85 *nats*, having almost a flat learning behavior after that point. Contrary, on the same dataset, the fully-connected RBMs have a very short initial good learning behavior (for few epochs) and, after that, they go up and down during the 5000 epochs analyzed, reaching the minimum performance of -160 *nats* (Figure 2, third row, last column). We have to mention that these good stability and over-fitting avoidance capacity, are induced not just by the SET procedure, but also by the sparsity itself, as RBM$_{FixProb}$, too, has a stable behavior in almost all the cases.

We finally verified our initial hypothesis about sparse connectivity in SET-RBM. Figure 3 shows how the connectivity naturally evolves towards a scale-free topology. To assess this fact, we have used the null hypothesis from statistics (Everitt, 2002), which assumes that there is no relation between two measured phenomena. To see if the null hypothesis between the degree distribution of the hidden neurons and a power-law distribution can be rejected, we have computed the p-value (Nuzzo, 2014; Clauset et al, 2009) between them. To reject the null hypothesis the p-value has to be lower than a statistically significant threshold of 0.05. In all cases (all plots of Figure 3), looking at the p-values (y-axes to the right of the plots), we can see that at the beginning of the learning phase the null hypothesis is not rejected. This was to be expected, as the initial degree distribution of the hidden neurons is binomial due to the randomness of the Erdős-Rényi random graphs (Newman et al, 2001) used to initialize the

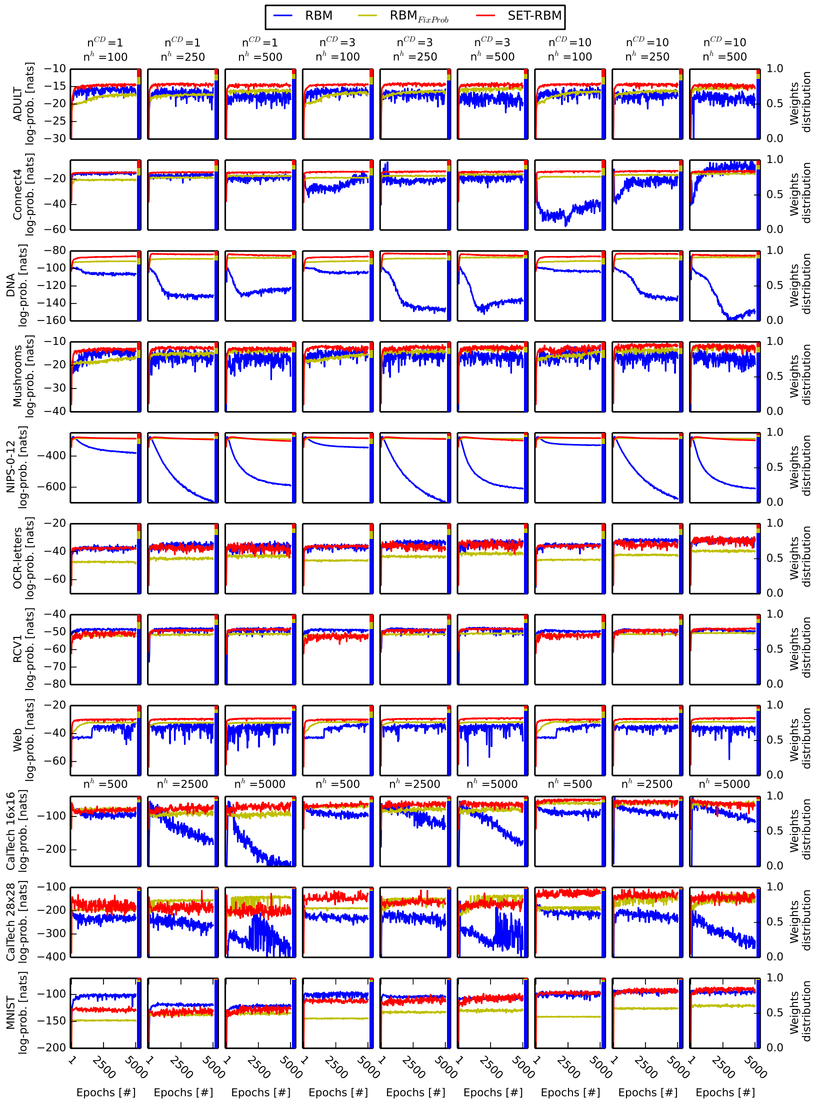

Fig. 2: Experiments with RBM variants using 11 benchmark datasets. For each model studied we have considered three cases for the number of Contrastive Divergence steps $n^{CD} = \{1, 3, 10\}$, and three cases for the number of hidden neurons ($n^h$). For the first 8 datasets (from top to bottom) we have used $n^h = \{100, 250, 500\}$, and for the last three datasets we have used $n^h = \{500, 2500, 5000\}$. The x-axes show the training epochs; the left y-axes show the average log-probabilities computed on the test data with AIS (Salakhutdinov and Murray, 2008); and the right y-axes (the stacked bar on the right part of the plots) reflect the fraction given by the $n^W$ of each model over the sum of the $n^W$ of all three models. Overall, SET-RBM outperforms the other two models in most of the cases. Also, it is interesting to see that SET-RBM and RBM$_{FixProb}$ are much more stable and do not present the over-fitting problems of RBM.

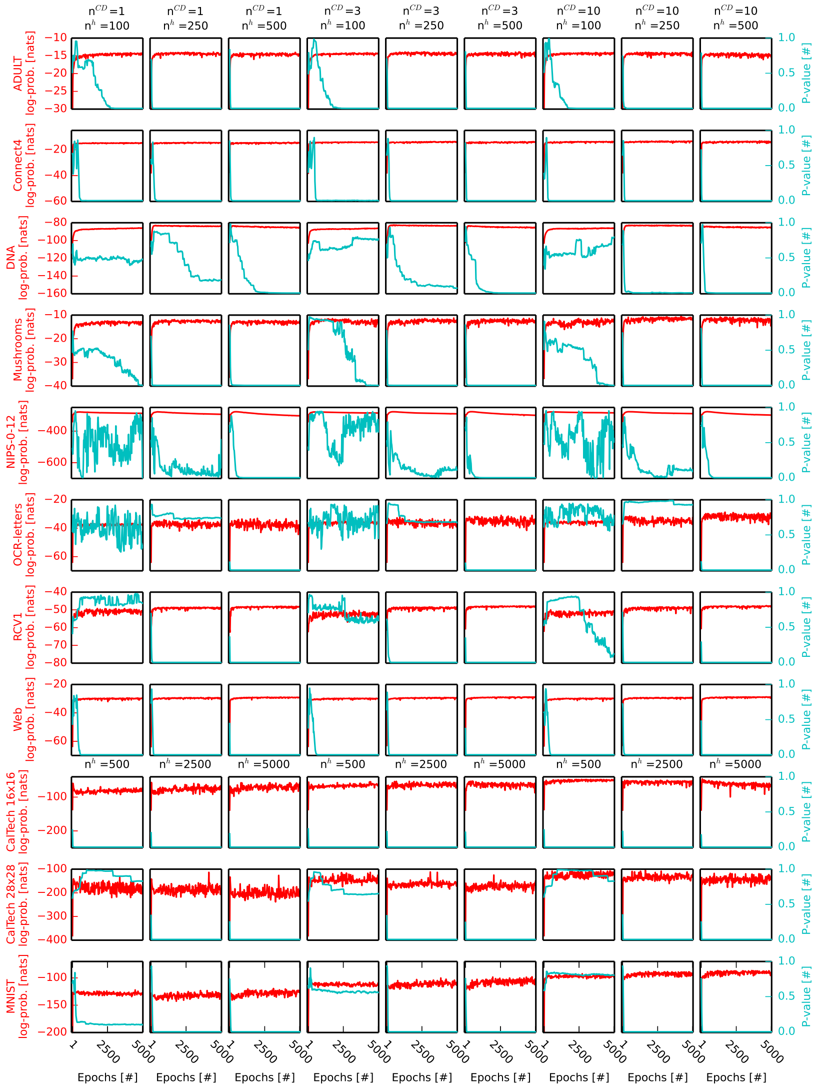

Fig. 3: SET-RBM evolution towards a scale-free topology. We have considered three cases for the number of Contrastive Divergence steps $n^{CD} = \{1, 3, 10\}$, and three cases for the number of hidden neurons ($n^h$). For the first 8 datasets (from top to bottom) we have used $n^h = \{100, 250, 500\}$, and for the last three datasets we have used $n^h = \{500, 2500, 5000\}$. The x-axes show the training epochs; the left y-axes (red color) show the average log-probabilities computed for SET-RBMs on the test data with AIS (Salakhutdinov and Murray, 2008); and the right y-axes (cyan color) show the p-values computed between the degree distribution of the hidden neurons in SET-RBM and a power-law distribution. We may observe that for models with a high enough number of hidden neurons, the SET-RBM topology always tends to become scale-free.

Table 2: Summarization of the experiments with RBM variants. On each dataset, we report the best average log-probabilities obtained with AIS on the test data for each model. $n^h$ represents the number of hidden neurons, $n^{CD}$ the number of CD steps, and $n^W$ the number of weights in the model.

| **Dataset** | | **RBM** | | | | **RBM**$_{FixProb}$ | | | | **SET-RBM** | | | | **XBM** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | **log-prob.** | $n^h$ | $n^W$ | $n^{CD}$ | **log-prob.** | $n^h$ | $n^W$ | $n^{CD}$ | **log-prob.** | $n^h$ | $n^W$ | $n^{CD}$ | **log-prob.** | $n^h$ | $n^W$ | $n^{CD}$ |
| UCI evaluation suite | ADULT | -14.91 | 100 | 12300 | 10 | -14.79 | 500 | 4984 | 10 | -13.85 | 500 | 4797 | 3 | -15.89 | 1200 | 12911 | 1 |
| | Connect4 | -5.01 | 500 | 63000 | 10 | -15.01 | 500 | 5008 | 10 | -13.12 | 500 | 4820 | 10 | -17.37 | 1200 | 12481 | 1 |
| | DNA | -85.97 | 500 | 90000 | 10 | -86.90 | 500 | 5440 | 10 | -82.51 | 250 | 3311 | 3 | -83.17 | 1600 | 17801 | 1 |
| | Mushrooms | -11.35 | 100 | 11200 | 10 | -11.36 | 500 | 4896 | 10 | -10.63 | 250 | 2787 | 10 | -14.71 | 1000 | 10830 | 1 |
| | NIPS-0-12 | -274.60 | 250 | 125000 | 3 | -282.67 | 500 | 8000 | 10 | -276.62 | 500 | 7700 | 3 | -287.43 | 100 | 5144 | 1 |
| | OCR-letters | -29.33 | 500 | 64000 | 10 | -38.58 | 500 | 5024 | 10 | -28.69 | 500 | 4835 | 10 | -33.08 | 1200 | 13053 | 1 |
| | RCV1 | -47.24 | 500 | 75000 | 3 | -50.34 | 500 | 5200 | 10 | -47.60 | 500 | 5005 | 10 | -49.68 | 1400 | 14797 | 1 |
| | Web | -31.74 | 500 | 150000 | 1 | -31.32 | 500 | 6400 | 10 | -28.74 | 500 | 6160 | 10 | -30.62 | 2600 | 29893 | 1 |
| CalTech 101 Silhouettes | 16x16 | -28.41 | 2500 | 640000 | 10 | -53.25 | 5000 | 42048 | 10 | -46.08 | 5000 | 40741 | 10 | -69.29 | 500 | 6721 | 1 |
| | 28x28 | -159.51 | 5000 | 3920000 | 3 | -126.69 | 5000 | 46272 | 10 | -104.89 | 2500 | 25286 | 10 | -142.96 | 1500 | 19201 | 1 |
| MNIST | | -91.70 | 2500 | 1960000 | 10 | -117.55 | 5000 | 46272 | 10 | -86.41 | 5000 | 44536 | 10 | -85.21 | 27000 | 387955 | 1:25 |

SET-RBMs topology. Subsequently, during the learning phase, we can see that, in many cases, the p-values decrease considerably at a statistical significant level under the $0.05$ threshold. When these situations occur, it means that the degree distribution of the hidden neurons in SET-RBM starts to approximate a power-law distribution. As to be expected, the cases with fewer neurons (e.g. Figure 3, fifth row, first column) fail to evolve to scale-free topologies, while the cases with more neurons always evolve towards a scale-free topology (Figure 3, columns 3, 6, and 9). To summarize, in 70 out of 99 cases studied, the SET-RBMs topology evolves clearly during the learning phase from an Erdős-Rényi topology towards a scale-free one.

### 4.3 SET performance on multi layer perceptron.

To better explore the capabilities of SET, we have also assessed its performance on classifications tasks based on supervised learning. We developed a variant of Multi Layer Perceptron (MLP) (LeCun et al, 2015), dubbed SET-MLP, in which the fully-connected layers have been replaced with sparse layers obtained through the SET procedure, with $\epsilon = 20$, and $\zeta = 0.3$. We kept the $\zeta$ parameter as in the previous case of SET-RBMs, while for the $\epsilon$ parameter we performed a small random search just on the MNIST dataset. We compared SET-MLP to a standard fully-connected MLP, and to a sparse variant of MLP having a fixed Erdős-Rényi topology, dubbed MLP$_{FixProb}$. For the assessment, we have used three benchmark datasets (Table 1), two coming from the computer vision domain (MNIST and CIFAR10), and one from particle physics (the HIGGS dataset (Baldi et al, 2014)). In all cases, we have used the same data processing techniques, network architecture, training method (i.e. Stochastic Gradient Descent (LeCun et al, 2015) with fixed learning rate of $0.01$, momentum of $0.9$, and weight decay of $0.0002$), and a dropout rate of $0.3$ (Table 3). The only difference between MLP, MLP$_{FixProb}$, and SET-MLP, consisted in their topological connectivity. We have used Python and the Keras library (Chollet, 2015) with Theano back-end (Al-Rfou et al, 2016) for this set of experiments. For MLP we have used the standard Keras implementation, while we implemented ourselves SET-MLP and MLP$_{FixProb}$ on top of the standard Keras libraries.

The results depicted in Figure 4 show how SET-MLP outperforms MLP$_{FixProb}$. Moreover, SET-MLP always outperforms MLP, while having two orders of magnitude fewer parameters. Looking at the CIFAR10 dataset, we can see that with only just $1\%$ of the weights of MLP, SET-MLP leads to significant gains. At the same time, SET-MLP has comparable results with state-of-the-art MLP models after these have been carefully fine-tuned.

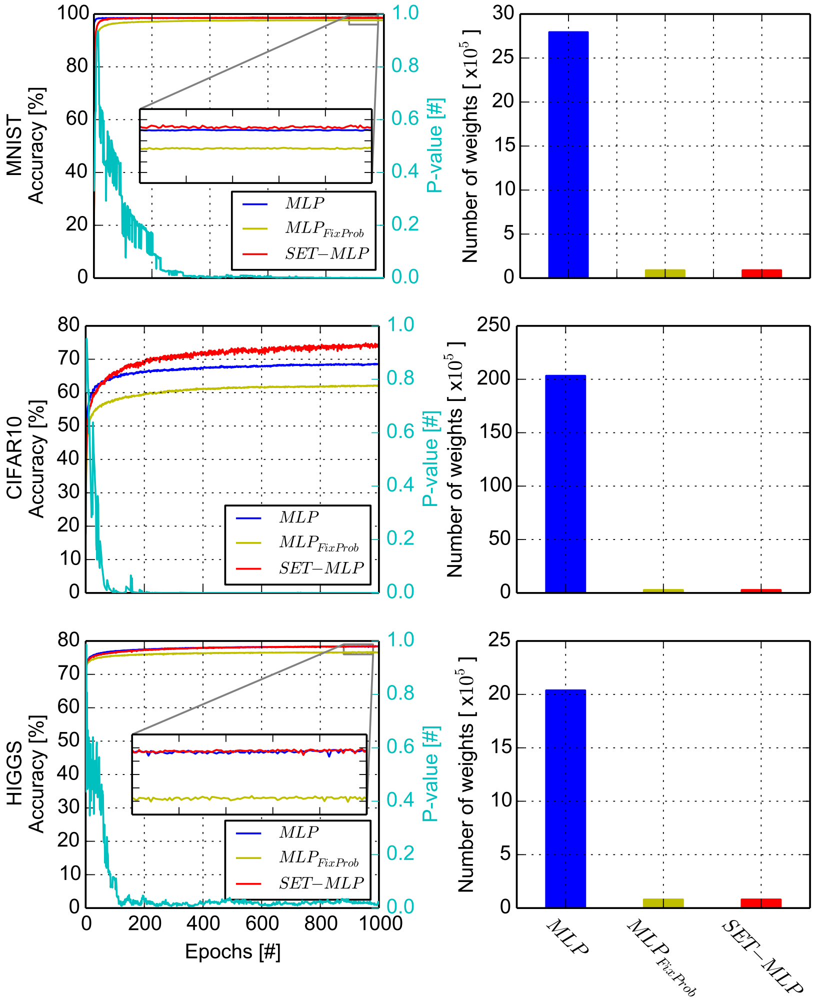

Fig. 4: Experiments with MLP variants using 3 benchmark datasets. The plots on the left side reflect models performance in terms of classification accuracy (left y-axes) over training epochs (x-axes); the right y-axes of the left plots give the p-values computed between the degree distribution of the hidden neurons of the SET-MLP models and a power-law distribution, showing how the SET-MLP topology becomes scale-free over training epochs. The bar plots on the right side depict the number of weights of the three models on each dataset. The most striking situation happens for the CIFAR10 dataset (second row) where the SET-MLP model outperforms drastically the MLP model, while having approximately 100 times fewer parameters.

Table 3: Summarization of the experiments with MLP variants. On each dataset, we report the best classification accuracy obtained by each model on the test data. $n^W$ represents the number of weights in the model.

| Dataset | Data | Architecture | Activation | MLP | | MLP$_{FixProb}$ | | SET-MLP | |
|---|---|---|---|---|---|---|---|---|---|
| | augmentation | | | Accuracy [%] | $n^W$ | Accuracy [%] | $n^W$ | Accuracy [%] | $n^W$ |
| MNIST | no | 784-1000-1000-1000-10 | SReLu | 98.55 | 2794000 | 97.68 | 89797 | 98.74 | 89797 |
| CIFAR10 | yes | 3072-4000-1000-4000-10 | SReLu | 68.70 | 20328000 | 62.19 | 278630 | 74.84 | 278630 |
| HIGGS | no | 28-1000-1000-1000-2 | SReLu | 78.44 | 2038000 | 76.69 | 80614 | 78.47 | 80614 |

To quantify, the second best MLP model in the literature on CIFAR10 reaches about $74.1\%$ classification accuracy (Urban et al, 2016) and has 31 million parameters: while SET-MLP reaches a better accuracy ($74.84\%$) having just about $0.3$ million parameters. Moreover, the best MLP model in the literature on CIFAR10 has $78.62\%$ accuracy (Lin et al, 2015), with about $12$ million parameters, while also benefiting from a pre-training phase (Hinton and Salakhutdinov, 2006; Hinton et al, 2006). Although we have not pre-trained the MLP models studied here, we should mention that SET-RBM can be easily used to pre-train a SET-MLP model to further improve performance.

Regarding the topological features, we can see from Figure 4 that, similarly to what was found in the SET-RBM experiments (Figure 3), the hidden neuron connections in SET-MLP rapidly evolve towards a power-law distribution.

Considering the different datasets under scrutiny, we should stress that we have assessed both image-intensive and non-image sets. On image datasets, Convolutional Neural Networks (CNNs) (LeCun et al, 2015) typically outperform MLPs. These, in fact, matches perfectly with the SET procedure. For instance, SET may be used to replace all CNNs fully connected layers with sparse evolutionary counterparts. The benefit would be two-fold: to reduce the total number of parameters in CNNs, and to permit the use of larger CNN models. However, CNNs are not viable on other types of high-dimensional data, such as biological data (e.g. (Danziger et al, 2006)), or theoretical physics data (e.g. (Baldi et al, 2014)). In those cases, MLPs will be a better choice. This is in fact the case of the HIGGS dataset (Figure 4, last row), where SET-MLP achieves $78.47\%$ classification accuracy and has about $90000$ parameters. Whereas, one of the best MLP models in the literature achieved a $78.54\%$ accuracy with three many times as many parameters (Lin et al, 2015).

The last but not the least, during all the experiments performed we observed that SET is quite stable with respect to the choice of meta-parameters $\epsilon$ and $\zeta$. There is no way to say that our choices offered the best possible performance, even if we fine-tuned them just on one dataset, i.e. MNIST, and we evaluated their performance on all 14 datasets. Still, we can say that a $\zeta = 0.3$ for both, SET-RBM and SET-MLP, and an $\epsilon$ specific for each model type, SET-RBM ($\epsilon = 11$) and SET-MLP ($\epsilon = 20$), were good enough to outperform state-of-the-art.

## 5 Conclusion

In this paper we have introduced SET, a simple procedure to replace ANNs fully-connected bipartite layers with sparse layers. We have validated our approach on 14 datasets (from different domains) and on two widely used ANN models, i.e. RBMs and MLPs. We have evaluated SET in combination with two different training methods, i.e. contrastive divergence and stochastic gradient descent, for unsupervised and supervised learning. We showed that SET is capable to *quadratically* reduce the number of parameters of bipartite neural networks layers, at no decrease in accuracy. In most of the cases, SET-RBMs and SET-MLPs outperform their fully-connected counterparts. Moreover, they always outperform their non-evolutionary counterparts, i.e. RBM$_{FixProb}$, and MLP$_{FixProb}$.

We can conclude that the SET procedure is coherent with real-world complex networks, whereby nodes connections tend to evolve into scale-free topologies (Barabási, 2016). This feature has important implications in ANNs: we could envision a computational time reduction by reducing the number of training epochs, if we would use for instance preferential attachment algorithms (Albert and Barabási, 2002) to evolve faster the topology of the bipartite ANN layers towards a scale-free one. Of course, this possible improvement has to be treated carefully, as forcing the model topology to evolve unnaturally faster into a scale-free topology may be prone to errors - for instance, the data distribution may not be perfectly matched.

SET can be widely adopted to reduce the fully-connected layers into sparse topologies in other types of ANNs, e.g. convolutional neural networks (LeCun et al, 2015), recurrent neural networks (LeCun et al, 2015), deep reinforcement learning networks (Mnih et al, 2015; Silver et al, 2016), and so on. SET may prove to be the basis of much larger ANNs, possibly on a billion-node scale to run in supercomputers. Also, it may lead to the building of small but powerfull ANNs which could be directly trained on low-resource devices (e.g. wireless sensor nodes, mobile phones), without the need of first training them on supercomputers and then to move the trained models to low-resource devices, as it is currently performed by the state-of-the-art Han et al (2015). These powerfull capabilities will be enabled by the linear relation between the number of neurons and the amount of connections between them yielded by SET. ANNs built with SET will have much more representational power, and better adaptive capabilities than the current state-of-the-art ANNs, and will push artificial intelligence well beyond its current boundaries.

## References

Al-Rfou R, Alain G, Almahairi A, et al CA (2016) Theano: A Python framework for fast computation of mathematical expressions. arXiv e-prints abs/1605.02688

Albert R, Barabási AL (2002) Statistical mechanics of complex networks. Rev Mod Phys 74:47–97, DOI 10.1103/RevModPhys.74.47

Baldi P, Sadowski P, Whiteson D (2014) Searching for exotic particles in high-energy physics with deep learning. Nature Communications 5:4308, DOI 10.1038/ncomms5308

Barabási AL (2016) Network science

Barabási AL, Albert R (1999) Emergence of scaling in random networks. Science 286(5439):509–512, DOI 10.1126/science.286.5439.509

Bengio Y (2009) Learning deep architectures for ai. Found Trends Mach Learn 2(1):1–127, DOI 10.1561/2200000006

Bishop CM (2006) Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag New York, Inc., Secaucus, NJ, USA

Bottou L, Bousquet O (2008) The tradeoffs of large scale learning. In: Platt J, Koller D, Singer Y, Roweis S (eds) Advances in Neural Information Processing Systems, vol 20, NIPS Foundation (http://books.nips.cc), pp 161–168

Bullmore E, Sporns O (2009) Complex brain networks: graph theoretical analysis of structural and functional systems. Nature Reviews Neuroscience 10(3):186–198

Chollet F (2015) keras. URL `https://github.com/fchollet/keras`

Clauset A, Shalizi CR, Newman MEJ (2009) Power-law distributions in empirical data. SIAM Rev 51(4):661–703, DOI 10.1137/070710111

Cybenko G (1989) Approximation by superpositions of a sigmoidal function. Mathematics of Control, Signals, and Systems (MCSS) 2(4):303–314, DOI 10.1007/bf02551274

Danziger SA, Swamidass SJ, Zeng J, Dearth LR, Lu Q, Chen JH, Cheng J, Hoang VP, Saigo H, Luo R, et al (2006) Functional census of mutation sequence spaces: the example of p53 cancer rescue mutants. IEEE/ACM Transactions on Computational Biology and Bioinformatics 3(2):114–125

Del Genio CI, Gross T, Bassler KE (2011) All scale-free networks are sparse. Phys Rev Lett 107:178,701, DOI 10.1103/PhysRevLett.107.178701

Dieleman S, Schrauwen B (2012) Accelerating sparse restricted boltzmann machine training using non-gaussianity measures. In: Bengio Y, Bergstra J, Le Q (eds) Deep Learning and Unsupervised Feature Learning, Proceedings, p 9

Diering GH, Nirujogi RS, Roth RH, Worley PF, Pandey A, Huganir RL (2017) Homer1a drives homeostatic scaling-down of excitatory synapses during sleep. Science 355(6324):511–515, DOI 10.1126/science.aai8355

Erdős P, Rényi A (1959) On random graphs i. Publicationes Mathematicae (Debrecen) 6:290–297

Everitt B (2002) The Cambridge dictionary of statistics. Cambridge University Press, Cambridge, UK; New York

Gehler PV, Holub AD, Welling M (2006) The rate adapting poisson model for information retrieval and object recognition. In: Proceedings of the 23rd International Conference on Machine Learning, ACM, New York, NY, USA, ICML ’06, pp 337–344, DOI 10.1145/1143844.1143887

Graves A, Liwicki M, Fernández S, Bertolami R, Bunke H, Schmidhuber J (2009) A novel connectionist system for unconstrained handwriting recognition. IEEE Trans Pattern Anal Mach Intell 31(5):855–868, DOI 10.1109/TPAMI.2008.137

Han S, Pool J, Tran J, Dally W (2015) Learning both weights and connections for efficient neural network. In: Cortes C, Lawrence ND, Lee DD, Sugiyama M, Garnett R (eds) Advances in Neural Information Processing Systems 28, Curran Associates, Inc., pp 1135–1143

Hastie T, Tibshirani R, Friedman J (2001) The Elements of Statistical Learning. Springer Series in Statistics, Springer New York Inc., New York, NY, USA

Hinton G (2012) A practical guide to training restricted boltzmann machines. In: Neural Networks: Tricks of the Trade, Lecture Notes in Computer Science, vol 7700, Springer, pp 599–619, DOI 10.1007/978-3-642-35289-8_32

Hinton GE (2002) Training Products of Experts by Minimizing Contrastive Divergence. Neural Computation 14(8):1771–1800, DOI 10.1162/089976602760128018

Hinton GE, Salakhutdinov RR (2006) Reducing the Dimensionality of Data with Neural Networks. Science 313(5786):504–507, DOI 10.1126/science.1127647

Hinton GE, Osindero S, Teh YW (2006) A fast learning algorithm for deep belief nets. Neural Comput 18(7):1527–1554, DOI 10.1162/neco.2006.18.7.1527, URL `http://dx.doi.org/10.1162/neco.2006.18.7.1527`

Larochelle H, Bengio Y (2008) Classification using discriminative restricted boltzmann machines. In: Proceedings of the 25th International Conference on Machine Learning, ACM, New York, NY, USA, ICML ’08, pp 536–543, DOI 10.1145/1390156.1390224

Larochelle H, Murray I (2011) The neural autoregressive distribution estimator. In: AISTATS, JMLR.org, JMLR Proceedings, vol 15, pp 29–37

LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11):2278–2324

LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444, DOI 10.1038/nature14539

Lin Z, Memisevic R, Konda K (2015) How far can we go without convolution: Improving fully-connected networks. arXiv preprint arXiv:151102580

Marlin BM, Swersky K, Chen B, de Freitas N (2010) Inductive principles for restricted boltzmann machine learning. In: AISTATS, JMLR.org, JMLR Proceedings, vol 9, pp 509–516

McDonnell JR, Waagen D (1993) Evolving neural network connectivity. In: IEEE International Conference on Neural Networks, pp 863–868 vol.2, DOI 10.1109/ICNN.1993.298671

Miikkulainen R, Liang JZ, Meyerson E, Rawal A, Fink D, Francon O, Raju B, Shahrzad H, Navruzyan A, Duffy N, Hodjat B (2017) Evolving deep neural networks. CoRR abs/1703.00548

Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G, Petersen S, Beattie C, Sadik A, Antonoglou I, King H, Kumaran D, Wierstra D, Legg S, Hassabis D (2015) Human-level control through deep reinforcement learning. Nature 518(7540):529–533, DOI 10.1038/nature14236

Mocanu DC (2016) On the synergy of network science and artificial intelligence. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, USA, July 9-15

Mocanu DC, Pokhrel J, Garella JP, Seppänen J, Liotou E, Narwaria M (2015) No-reference video quality measurement: added value of machine learning. Journal of Electronic Imaging 24(6):061,208, DOI 10.1117/1.JEI.24.6.061208

Mocanu DC, Mocanu E, Nguyen PH, Gibescu M, Liotta A (2016) A topological insight into restricted boltzmann machines. Machine Learning 104(2):243–270, DOI 10.1007/s10994-016-5570-z

Newman ME, Strogatz SH, Watts DJ (2001) Random graphs with arbitrary degree distributions and their applications. Physical review E 64(2):026,118

Nuzzo R (2014) Scientific method: Statistical errors. Nature 506(7487):150–152, DOI 10.1038/506150a

Osogami T, Otsuka M (2014) Restricted boltzmann machines modeling human choice. In: Advances in Neural Information Processing Systems 27, pp 73–81

Pessoa L (2014) Understanding brain networks and brain organization. Physics of Life Reviews 11(3):400 – 435

Rosenblatt F (1962) Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. Spartan

Rumelhart D, Hintont G, Williams R (1986) Learning representations by back-propagating errors. Nature 323(6088):533–536

Salakhutdinov R, Murray I (2008) On the quantitative analysis of deep belief networks. In: In Proceedings of the International Conference on Machine Learning, pp 872–879

Salakhutdinov R, Mnih A, Hinton G (2007) Restricted boltzmann machines for collaborative filtering. In: Proceedings of the 24th International Conference on Machine Learning, ACM, ICML '07, pp 791–798, DOI 10.1145/1273496.1273596

Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M, et al (2016) Mastering the game of go with deep neural networks and tree search. Nature 529(7587):484–489

Smolensky P (1987) Information processing in dynamical systems: Foundations of harmony theory. In: Rumelhart DE, McClelland JL, et al (eds) Parallel Distributed Processing: Volume 1: Foundations, MIT Press, Cambridge, pp 194–281

Strogatz SH (2001) Exploring complex networks. Nature 410:268–276, DOI 10.1038/35065725

Sutton RS, Barto AG (1998) Introduction to Reinforcement Learning, 1st edn. MIT Press, Cambridge, MA, USA

Urban G, Geras KJ, Kahou SE, Aslan O, Wang S, Caruana R, Mohamed A, Philipose M, Richardson M (2016) Do deep convolutional nets really need to be deep and convolutional? arXiv preprint arXiv:160305691

de Vivo L, Bellesi M, Marshall W, Bushong EA, Ellisman MH, Tononi G, Cirelli C (2017) Ultrastructural evidence for synaptic scaling across the wake/sleep cycle. Science 355(6324):507–510, DOI 10.1126/science.aah5982

Watts DJ, Strogatz SH (1998) Collective dynamics of 'small-world' networks. Nature 393:440–442

Whiteson S, Stone P (2006) Evolutionary function approximation for reinforcement learning. Journal of Machine Learning Research 7:877–917

Yosinski J, Lipson H (2012) Visually debugging restricted boltzmann machine training with a 3d example. In: Representation Learning Workshop, 29th International Conference on Machine Learning