Resilience of Pattern Recognition in Multi-layered Neural Networks

Tahereh Mozaffari[1]*, Mahta Ramezanian[1]

Abstract
In order to study the functionality of the neural networks, we have to develop tools to predict their behavior in accordance with errors, failures and environmental changes, which manipulate their topology. Separated roles of the systems dynamics and topology allows us to use unique tools to learn about resilience function of the system, but to do so, we need to design realistic models of neural networks. In this project, we developed a multi-layered neural network and we used two of the fastest learning algorithms to train it some retrieval patterns. Finally, we investigated resilience of the pattern recognitions by applying perturbations on the links.

Keywords
restricted Boltzmann machine — Hopfield model — learning — Neuronal dynamics — Resilience — multi-layered

[1] *Department of Physics, Sharif University of Technology, Tehran, Iran*
***Corresponding authors**: ta.mozaffari@gmail.com, ramezanian@physics.sharif.edu

Contents

Introduction

The resilience of a complex network can be roughly defined as the ability of the system in adjusting to disturbances, both internal and external, in order to remain functional [1]. In a neural network we define functionality a parameter related to the ability of learning. We intend to perturb the topology of the network and determine how the functionality will be affected. In this regard, we eliminate a random neuron and test if the network can still recognize the given image. We examine the network after this perturbation 100 times and the functionality is the ratio of the right recognition to 100. As it is mentioned in the Universal resilience patterns in complex networks[2], the β_{eff} is the key parameter which indicates whether the network is capable of recognition or not. the β_{eff} is defined as

$$\beta_{eff} = \frac{\mathbb{I}^T A s^{in}}{\mathbb{I}^T A \mathbb{I}}$$

Where A is the adjacent matrice of the network and s^{in} is the ingoing weighted degrees.

In this project, we devised a neural network, capable of learning random*** patterns regardless of their rotational status with a tolerence for noises. For this purpose we had to use a two-layered neural network, the first layer, with 1600 neurons, for recognition of rotation and the second one, with 400 neurons, for eliminating the noise.

We need a network with multiple types of nodes and different roles, it has suddenly become very fashionable to study this kind of networks with multiple layers since they learn the nonlinearity at the same time as the linear discriminant. They implement linear discriminants in a space where the inputs have been mapped nonlinearly. They admit simple algorithms where the form of the nonlinearity can be learned from training data.

Additionally, since in order to calculate of the resilience and the observation of the*** gozar several disturbations needs to be applied, it is essential to move beyond simple networks and investigate more complicated but more realistic frameworks. For example, more noisy and transformed pattern can be recognized, or edges exhibit heterogeneous features: they can have different weights [3] [4] [5], exist only between nodes that belong to different sets (e.g., bipartite networks) [6], or be active only at certain times [7] [8].

In what follows, we describe the network we are studying, learning algorithms and the calculation of the resilience parameters. In the end, we investigate the impact of the applied disturbances to the resilience of learning in the studied network.

1. Methods

Our job can be categorized into three parts:

1. Forming the neural network
2. Training the network

Figure 1. An example of retrieval (left) and given (right) pattern. the given sample is rotated and noisy.

3. Calculation of the resilience parameters

1.1 Neural network

The neural network is consist of two layers:

1. Layer 1 with 1600 neurons
2. Layer 2 with 400 neurons

each neuron in layer 2 is connected to 5 neurons in Layer 1 and is fully connected to the other neurons in its own layer. Detailed features like noise are being investigated in layer 1 and coarse features like rotation and translation matched in the second layer. In layer 1, weights have been set according to Hebbian rule:

$$w_{ij} = \frac{1}{N}\sum_{\mu=1}^{p} \varepsilon_i^{\mu}\varepsilon_j^{\mu}$$

For pattern μ, if the bits corresponding to neurons i and j are equal, then the product $\varepsilon_i^{\mu}\varepsilon_j^{\mu}$ will be positive. This would, in turn, have a positive effect on the weight w_{ij} and the values of i and j will tend to become equal. The opposite happens if the bits corresponding to neurons i and j are different.

1.2 Learning algorithms

In the layer 1, we use Hopfield model to update the spins. For the Hopfield Networks, this is implemented in the following manner, when learning p patterns:

$$w_{ij} = \frac{1}{N}\sum_{\mu=1}^{p} \varepsilon_i^{\mu}\varepsilon_j^{\mu}$$

For pattern μ, if the bits corresponding to neurons i and j are equal, then the product $\varepsilon_i^{\mu}\varepsilon_j^{\mu}$ will be positive. This would, in turn, have a positive effect on the weight w_{ij} and the values of i and j will tend to become equal. The opposite happens if the bits corresponding to neurons i and j are different.

In order to use these layers for learning relevant features of the patterns, we implemented deep learning technique for layer 2 and the interconnection spins. To do so, We construct an exact mapping from the variational renormalization group, first introduced, and deep learning architectures based on Restricted Boltzmann Machines (RBMs). We illustrate these ideas using the nearest-neighbor Ising Model in one and two-dimensions. Eventually, we used deep learning algorithms to employ a generalized renormalization group technique to learn relevant features from data. [9] Therefore, Spins and interconnection weights in layer 2 update follow the Boltzmann restricted machine equations:

$$P(x) = \frac{1}{1+\exp^{-x}}$$

$$s_j^{l1} = P\sum_i w_{ij} s_i^{l2}$$

$$s'^{l2}_i = P\sum_j w_{ij} s'^{l1}_j$$

$$\Delta w_{ij} = \eta\left(\sum_i s_i^{l1} s_j^{l2} - \sum_i s'^{l1}_i s'^{l2}_j\right)$$

Deep learning Calculations in layer 2 is as same as layer 1 but their input has a lower resolution. One can easily conclude that from the fact that the second layer has fewer neurons than the first one (also known as decimation-based renormalization transformations). This has been illustrated in the Figure 2. The spins at layer 2 have a natural interpretation as the decimated spins when performing the RG transformation in the layer 1. This implies that the hidden spins in the second layer of the DNN are also described by the RG transformed Hamiltonian with a coupling between neighboring spins. Repeating this argument for spins coupled between the second and third layers and so on, one obtains the deep learning architecture shown in Figure 2 which implements decimation.[9]

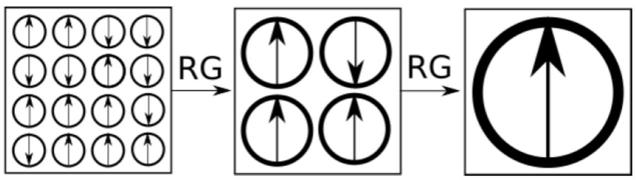

Figure 2. Deep learning architecture in the one-dimensional Ising Model.

In the layer 2, weights are being updated using energy function of the neural network. This is calculated as:

$$E = -(\sum_{i,j=1} w1_{ij}x_i x_j + \sum_{i,j=1} w2_{ij}x_i x_j + 2\alpha \sum_{i,j=1} w12_{ij}x_i x_j)$$

Where $\alpha$is 0.25 [10] and $w1$, $w2$ and $w12$ respectively represent weights in layer 1, layer 2 and interconnection edges. Notice that the edges are indirected. The energy function decreases as the system gets updated in each step until it reaches a local minimum, corresponding to an attractor.

After all these, standard equilibrium Monte Carlo technique is being utilized, that means spins are being changed with the probability of $\exp^{-\Delta E}$. Where ΔE is the difference between the energy of two subsequent steps. which is derived by Hopfield model in layer 1 and by restricted Boltzmann machine in layer 2. Spins of the layer 1 update as

$$P(s_i^{l1}) = g(\sum_i w_{ij} s_j^{l2})$$

Subsequently, the inputs of layer 2 will be updated. That means spins in layer 2 connecting to the new spins will be evaluated again by restricted Boltzmann machine in the new state.

Preception of rotation To decide if the given pattern is a rotation of a retrieval pattern, an angle $\theta$is being chosen randomly. Then we calculate ΔE by rotating all the spins in layer 2 corresponding to θ. The rotation is accepted with the probability of $\exp^{-\Delta E}$. Where ΔE is defined as

$$\Delta E = \Delta E_1 + \Delta E_2 + \Delta E_{12}$$

and

$$\Delta E_i = \Delta E_i' - \Delta E_i$$

Where E_i represents the energy of layer i and prime sign indicates the state after the rotation. For the layer 2, energies can be calculated as follow

$$E_2 = -\frac{1}{2}\sum_{rr'} w_{rr'} S(r) S(r') - \sum_r \varepsilon(r) S(r)$$

Where the respective weights are being calculated like the ordinary Hopfield:

$$w_{rr'} = \frac{1}{N}\sum_{\mu} P(r)^{\mu} P(r')^{\mu}$$

Notice that R is the rotation matrice, r is the position of the neuron, and

$$\varepsilon(r,t) = h_0 S^0(\hat{R}(\theta)r)$$

$$E'(\theta) \simeq -\frac{1}{2}\beta^2 h_0^2 \sum_{rr'} w_{rr'} S(\hat{R}(\theta)r) S(\hat{R}(\theta)r')$$

$$= -\frac{1}{2} N \beta^2 h_0^2 \sum_{\mu} (M_0^{\mu}(R(\theta)))^2$$

Then if the rotation is accepted (which occures with the probability of $\exp^{-\Delta E}$), the angle of rotation changes for next step is

$$\theta(t+1) = \theta(t) - \frac{\delta H}{\delta \theta} + \eta(t)$$

$\eta(t)$ is white noise.

We used 4 retrieval patterns and we know that all the patterns have finite correlation (Figure 2)

$$\frac{1}{N}\sum_{\mu} P(r)^{\mu} P(r')^{\mu} \sim \exp(-\frac{\theta}{\theta_c})$$

[10]

1.3 Resilience function

After $n = 100$ steps. We calculate the overlap and mean of weights. To do so, let's assume the system dynamics is as follows

$$\frac{dx}{dt} = f(\beta, x)$$

the β parameter is related to environmental conditions. According to what preceeds, one can see that the retrevial patterns are the fixed points of this equation. In a the network we are studying, the state of each node is affected by the state of its immediate neighbours. Therefore, we characterize the effective state of the system using the average nearest-neighbour activity

$$x_{eff} = \frac{\mathbb{I}^T A \mathbf{x}}{\mathbb{I}^T A \mathbb{I}}$$

Because

$$\beta_{eff} = \frac{\mathbb{I}^T A s^{in}}{\mathbb{I}^T A \mathbb{I}}$$

h is

$$h_i = \sum_j w_{ij}$$

Therefore,

$$\beta_{eff} = \sum_i h_i^2$$

So x_{eff} can also be derieved as [2]

$$h_i = \sum_j w_{ij}$$

$$x_{eff} = m^{\mu} = \sum_i p_i^{\mu} S_i h_i$$

In this project, we disturb the network in two ways:

- For changing the topology of the network m links are removed randomly
- Each weight shifts by a factor a (for m links) : $w_{ij} = aw_{ij}$

2. Results and Discussion

The resilience pattern of a complex system is effectively is not universal and depends on the system's state parameter space. Once, however, because of the separation of the systems dynamics and topology, we map the system into -space we can accurately predict the systems response to diverse perturbations and correctly identify the critical points where the system loses its resilience. The role of the network topology is fully captured by the β_{eff}. This parameter predicts density, heterogeneity, and symmetry of the system, three structural factors affecting system resilience. They predict system response to different perturbations [2] and suggest potential intervention strategies to avoid the loss of resilience or principles for designing resilient neural networks that can successfully cope with perturbations. We believe the more mature result can shed light on the methods of making biological neural network optimally resilient. What follows is the resilience of our neural network in response to specified perturbations.

2.0.1 Link loss

Firstly we eliminate random links. In case of eliminating links from layer 1 or 2 (Figures 3 and 4), the result are as expected: A critical value for the β_{eff}^c which is approximately 0.63 for layer 1 and 0.19 for layer 2. There is a transition in this critical value, where below that, the activity of network is nearly zero. In Figure 4, one can see that there is a sudden reduction of activity in $\beta_{eff} = 0.4$ too, yet it is not as catastrophic as the transition in $\beta_{eff} = 0.19$.

However, the response to interconnection link loss between layer 1 and layer 2 (Figure 5) is odd. We see irregularities in the behavior of the network, which can be a result of limitations on the computation (e.g. few number of intermediate links) or a matter of complexity that our team is not aware of. Nonetheless, the transition in $\beta_{eff} = 0.65$ is clear, although meaningless!

The most bizarre result happens in Figure 6. Here, we concluded that although sparse networks with $\beta_{eff} < 0.22$ are inactive, a vast number of interconnection links are destructive as well. As one can see in Figure 6, activity of the network reduces significantly for $\beta_{eff} > 1.2$

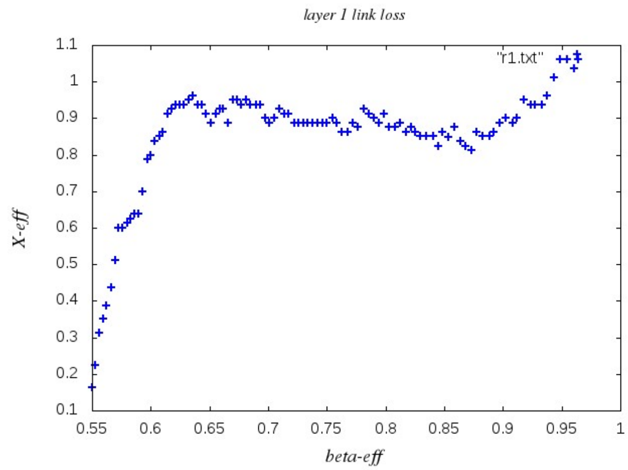

Figure 3. Layer 1 link loss

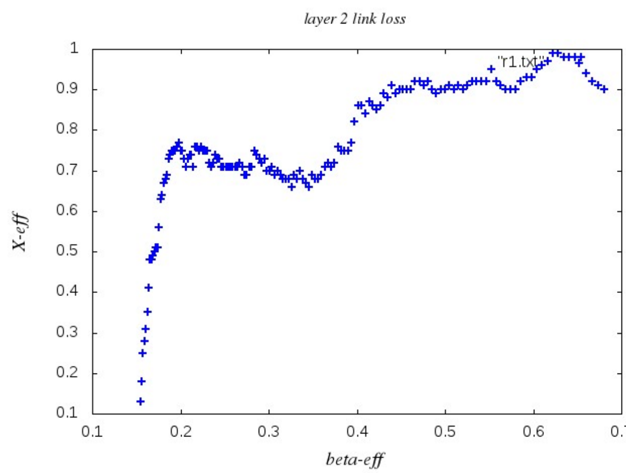

Figure 4. Layer 2 link loss

2.0.2 Weight Reduction

The response to this perturbation is smooth and predictable. For $\beta_{eff} > 0.6$ in layer 1 and more than 0.7 in layer 2, the collapse of the neural network would be avoided. These sharp transitions indicate that regardless of the network structure and the form of perturbation, the state of the system is fully determined by β_{eff}.

In all these samples, resilience increases with β_{eff}, as the larger is β_{eff}, the deeper is the system into the active state. Consequently, resilience is governed by three topological characteristics, where dense, symmetric and heterogeneous

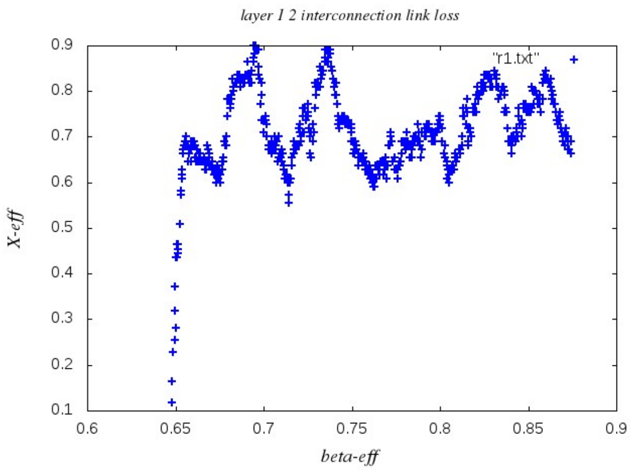

Figure 5. Layer 12 interconnection link loss

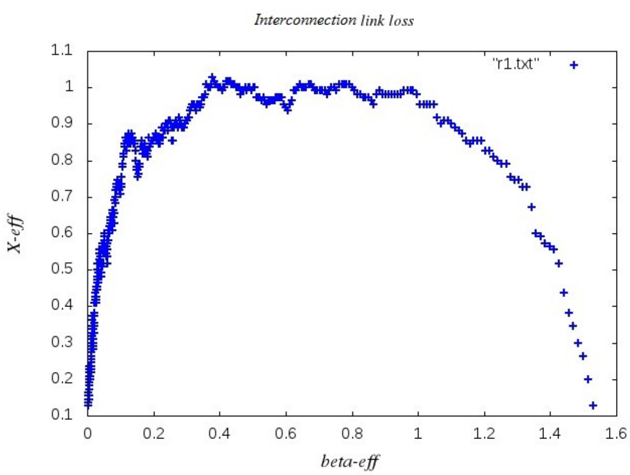

Figure 6. Interconnection link loss

networks are most resilient (large β_{eff}), and sparse, antisymmetric and heterogeneous networks are least resilient (small β_{eff}).

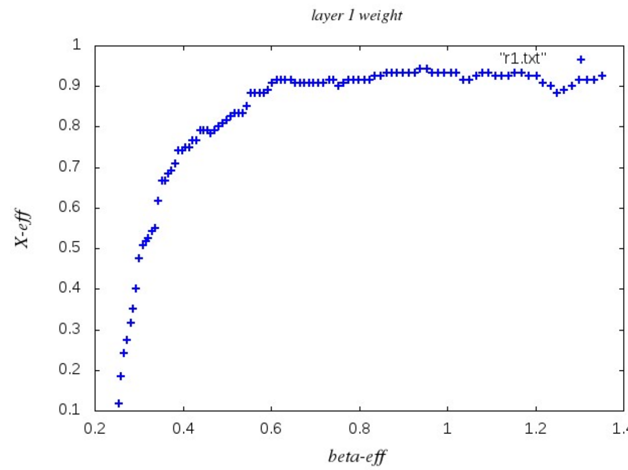

Figure 7. Weight reduction in layer 1

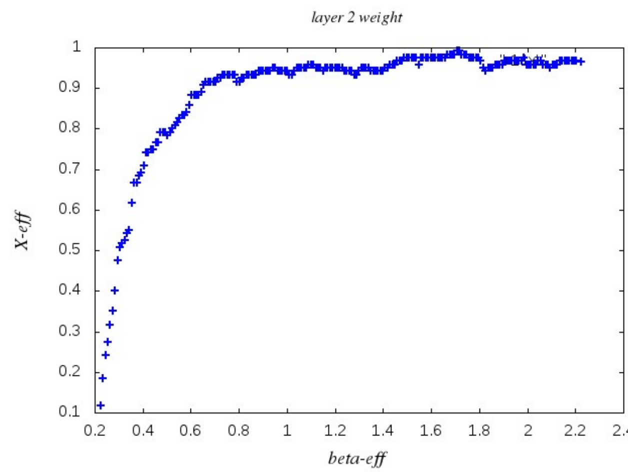

Figure 8. Weight reduction in layer 2

References

[1] Erez K. Ben-Avraham D. & Havlin S Cohen, R. Resilience of the internet to random breakdown. *Phys. Rev. Lett.*, 85:4626 4628, 2000.

[2] Baruch Barzel & Albert-Lszl Barabsi. Gao, Jianxi. Universal resilience patterns in complex networks. *Nature*, 530.7590:307–312, 2016.

[3] Alain Barrat, Marc Barthelemy, Romualdo Pastor-Satorras, and Alessandro Vespignani. The architecture of complex weighted networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(11):3747–3752, 2004.

[4] Mark EJ Newman. Random graphs with clustering. *Physical review letters*, 103(5):058701, 2009.

[5] Stanley Wasserman and Katherine Faust. *Social network analysis: Methods and applications*, volume 8. Cambridge university press, 1994.

[6] Newman MEJ. Networks: an introduction, 2010.

[7] Petter Holme. Modern temporal network theory: a colloquium. *The European Physical Journal B*, 88(9):1–30, 2015.

[8] Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P Gleeson, Yamir Moreno, and Mason A Porter. Multilayer networks. *Journal of complex networks*, 2(3):203–271, 2014.

[9] Pankaj Mehta and David J Schwab. An exact mapping between the variational renormalization group and deep learning. *arXiv preprint arXiv:1410.3831*, 2014.

[10] VS Dotsenko. Neural networks: translation-, rotation-and scale-invariant pattern recognition. *Journal of Physics A: Mathematical and General*, 21(15):L783, 1988.