Structural Drivers of Function in Information Processing Networks

Ann Hermundstad, Kevin Brown, Danielle Bassett, and Jean Carlson
Department of Physics, University of California, Santa Barbara, California 93106-9530
Email: ann@physics.ucsb.edu

***Abstract*—Structural configuration drives functionality in a range of different natural and artificial information processing systems. In this study, we use feedforward neural networks to evaluate the impact of structural variations on the ability of a network to learn and retain representations of external information. Performance is evaluated by statistically analyzing the error in the representations produced by parallel and layered networks during supervised, sequential function approximation. By varying the initial network state and the time given to learn the information, we identify tradeoffs between configurations that optimize for the best versus worst case scenarios and for the production of accurate versus retainable and generalizable representations of information. We show that these tradeoffs are maintained in larger networks and for variations in the information presented to the networks. By characterizing the curvature, depth, and participation of network connections about local error landscape minima, we find that variations in landscape structure give rise to the observed tradeoffs in performance. Consistently deep, narrow minima enable parallel networks to produce highly accurate solutions at the cost of more frequent failure in retention and generalizability. In contrast, variability in the depth and curvature of local minima enables layered networks to produce coarse but generalizable solutions at the cost of hindering consistent accuracy. Identifying structural drivers of functional performance is crucial for understanding both successes and limitations of information processing systems.**

I. Introduction

Structural configuration plays a crucial role in determining the functional performance of both artificial and biological information processing systems. For example, the structure of artificial systems can be carefully constructed [9], [10] to efficiently and accurately perform a specific function. Similarly, biological neuronal networks display a range of different structural motifs that enable the performance of disparate functions [4], [5].

In systems that must balance competitive processes, such as flexible learning and stable memory, variations in structural configuration may reveal functional tradeoffs in performance. We use feedforward neural networks, for which both the network structure and the external information can be precisely controlled, to systematically evaluate the error in representations of information produced during supervised, sequential one-dimensional function approximation. Our approach, however, is very different from studies that seek the "optimal" network structure to accurately perform a single task; rather, we identify structural features that impact learning and memory performance. Across a range of parallel and layered topologies, we find inherent tradeoffs in network function that arise solely from variations in underlying structure. These tradeoffs include optimization for best versus worst case scenarios and optimization for producing accurate versus retainable and generalizable representations of information.

In the remainder of the paper, we discuss the extent to which network configurations differ in their ability to both learn and retain information. Additional details, methodological considerations, and applications to neuronal systems can be found in [6].

II. Model

We evaluate the performance of feedforward, backpropagation (FFBP) artificial neural networks for the task of supervised, sequential, one-dimensional function approximation. The construction of our network model is consistent with standard FFBP neural network models [17]. We consider the five distinct topologies shown in Figure 1a. Each network has 12 hidden nodes arranged into h layers of ℓ nodes per layer. Nodes in adjacent layers are connected via variable, unidirectional weights. Interlayer connectivities were chosen in order to roughly maintain the same total number of adjustable parameters per network, N_p, noted in Figure 1a.

All networks are constructed using identical nodes with sigmoidal transfer functions $s(x) = 1/(1+e^{(-x)})$ and variable thresholds θ. The output $y = s(\sum_{p=1} \omega_p x_p - \theta)$ of each node is a function of the sum of its inputs x_p weighted by the variable connection strengths ω_p. Representing the threshold as $\theta = \omega_0 x_0$, where $x_0 = 1$, allows us to organize all adjustable parameters into a single, N_p-dimensional weight vector $\vec{\omega}$.

During training, each network is presented with a training pattern of N_d pairs of input x_d and target y_d values, denoted $(\vec{x}, \vec{y})$. The set of variable weights $\vec{\omega}$ is iteratively updated via the Polak-Ribiere conjugate gradient (PRCG) descent method with an adaptive step size [18], [19] in order to minimize the output error $E(\vec{\omega})$, a process theoretically analogous to searching an error landscape for a local or global minimum. We use online training, for which $E(\vec{\omega})$ is the sum of squared errors between the network output $y(\vec{\omega})$ and target output y calculated after all N_d points are presented, $E(\vec{\omega}) = \sum_d (y_d(\vec{\omega}) - y_d)^2/2$.

To simultaneously study learning and memory processes, we present information to the network in two sequential training sessions. We use a biologically-motivated pseudorehearsal technique during the second training session to preserve memory of the first session. This involves retraining the network

978-1-4673-0323-1/11/$26.00 ©2011 IEEE

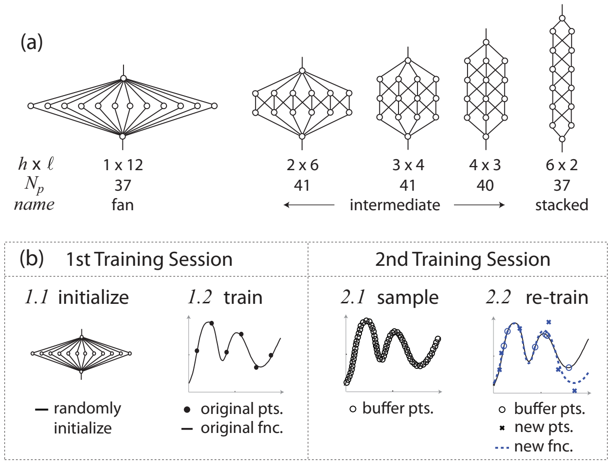

Fig. 1. Network configurations and training task. (a) Network configurations considered in this study. Indicated below each network are the number of hidden layers h and nodes per layer ℓ, the total number of adjustable parameters N_p, and the name by which we refer to the network. (b) Illustration of the sequential learning task described in the text applied to the fan network.

with new information and a *representation* of the original information [15]. The steps of this process are shown in Figure 1b and are described below:

First Training Session

Step 1.1 - Initialize: The network is initialized with randomly chosen weights ("randomly initialized state").

Step 1.2 - Train: Each network trains on six fixed "original" points, $(\vec{x}^{(o)}, \vec{y}^{(o)})$, that represent the information we wish the network to remember in subsequent training sessions. The values of these points, chosen to be evenly spaced in x and random in y, are identical for all five networks. Each network is given 10^5 iterations ("unlimited" training time) to generate a functional representation f_o of $(\vec{x}^{(o)}, \vec{y}^{(o)})$.

Second Training Session

Step 2.1 - Sample: Each network begins the second training session with the set of weights that generate f_o ("sampled state"). Each network randomly samples a pseudo-pool of 1000 buffer points from f_o, subsets of which are used in the following step to simulate memory rehearsal.

Step 2.2 - Re-train: The network is given a "limited" training time of 500 iterations to re-train on six randomly chosen new points $(\vec{x}^{(n)}, \vec{y}^{(n)})$ and six buffer points $(\vec{x}^{(b)}, \vec{y}^{(b)})$ randomly selected from the pseudo-pool. We repeat the second training session 1000 times to generate a distribution of solutions $\{f_n\}$ of the new and buffer points.

III. Results

We evaluate the performance of all five networks shown in Figure 1a during the sequential training task. To ensure the robustness of the results, we additionally evaluate the performance of larger networks consisting of 18 nodes arranged into configurations with $(h\text{x}\ell = 1\text{x}18, 2\text{x}9, 3\text{x}6, 6\text{x}3, 9\text{x}2)$. Lastly, we train the networks shown in Figure 1a using a permuted set of values for the original points $(\vec{x}^{(o)}, \vec{y}^{(o)})$.

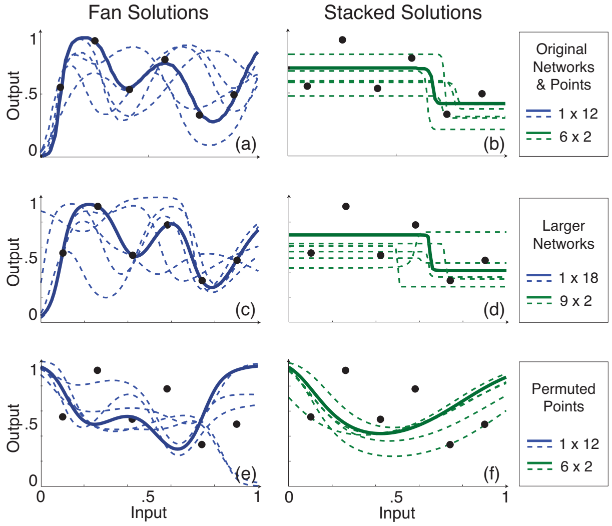

Fig. 2. Network solutions. Panels (a) and (b) show solutions produced respectively by the fan and stacked networks, indicating the approximations f_o (solid curve) of the original points (point markers) and subsets of approximations $\{f_n\}$ (dashed curves) of the new and buffer points. During the first training session, the fan network achieves a lower error by fitting the original points with a high order polynomial, while the stacked network produces a higher error fit that averages over variation in the y-values of the original points. Subsequent approximations $\{f_n\}$ retain the features these features of f_o. These results are consistent for larger networks, shown in panels (c) and (d), and for permutations of the original points, shown in panels (e) and (f).

A. Tradeoffs in Learning and Memory Tasks

Fan and Stacked Networks: We see qualitative differences in the solutions f_o and $\{f_n\}$ produced by the fan and stacked networks, shown respectively in Figures 2a and 2b. As the specific form of f_o depends on the randomly initialized network state (see the following section), we use solutions f_o that are representative of average network performance over a range of randomly initialized states.

The fan network accurately fits all six original points with a high order polynomial, while the stacked network produces a coarser solution that averages over the variation in the original points (solid curves in Figures 2a and 2b). Similar features are observed using larger networks (Figures 2c and 2d) and permutations of the original training points (Figures 2e and 2f). Increasing the size of the networks results in more pronounced features, such as the sharper kinks produced by the stacked network (Figure 2d), in the solutions f_o and $\{f_n\}$. Training on permuted original points results in a more accurate fit produced by the fan as compared to the stacked network, which averages over the variation in the permuted points (Figure 2f).

Intermediate Networks: We compute the errors $\{E_n^{(o)}\}$ and $\{E_n^{(n)}\}$ in the solutions produced by all five networks shown in Figure 1a, and we find tradeoffs in performance across the full range of configurations.

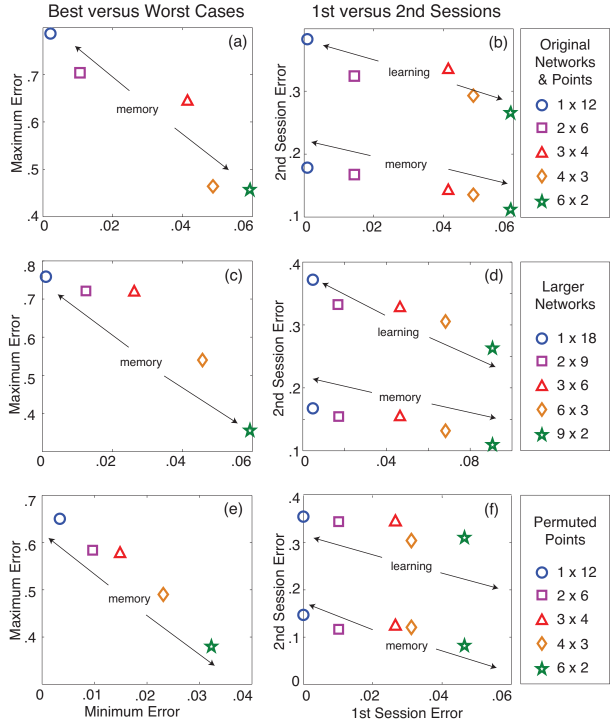

Fig. 3. Tradeoffs in network learning and memory. Panel (a) shows the best successes and worst failures in memory measured with respect to solutions $\{f_n\}$, where increasing h/ℓ decreases the maximum value of $\{E_n^{(o)}\}$ at the cost of increasing its minimum value. Panel (b) shows the average performance during the first versus second training session, measured with respect to solutions f_0 and $\{f_n\}$, where increasing h/ℓ increases $E_o^{(o)}$ achieved during the first session but decreases $\langle\{E_n^{(n)}\}\rangle$ and $\langle\{E_n^{(o)}\}\rangle$ achieved during the second session. These results are consistent for larger networks, shown in panels (c) and (d), and for permutations of the original points, shown in panels (e) and (f).

In Figure 3a, we see a tradeoff between optimization for the best case (maximization of success) versus worst case (minimization of failure) scenarios in retention, whereby lower minimum values of $\{E_n^{(o)}\}$ correspond to larger maximum values. Parallel networks maximize successful retention by producing lower minimum error values, while layered networks minimize failure in retention by producing lower maximum error values.

We furthermore find a tradeoff between performance during the first versus second training sessions, as shown in Figure 3b. Lower values of $E_o^{(o)}$ produced during the first training session correspond to larger values of both $\langle\{E_n^{(n)}\}\rangle$ and $\langle\{E_n^{(o)}\}\rangle$ produced during the second training session. This suggests a tradeoff between the production of accurate versus retainable and generalizable representations of information. Parallel networks produce more accurate solutions during initial training, while layered networks are better able to retain and generalize coarser solutions during subsequent training.

These tradeoffs are observed in larger networks (Figures 3c and 3d) and for permutations of the original points (Figures 3e and 3f).

B. Variable Training Conditions

To understand how tradeoffs arise from variations in structure, we probe the features of the underlying error landscapes that each network must navigate during the training process.

Identifying Local Landscape Minima: To identify landscape minima, we give each network "unlimited" training time to produce representations of the original points. We repeat this training process 500 times with different randomly initialized states in order to generate a distribution of solutions $\{f_o\}$. The errors $\{E_o^{(o)}\}$ in these solutions then correspond to local minima within the error landscape.

The cumulative distribution function (CDF) of $\{E_o^{(o)}\}$, shown in Figure 4a, reveals that the fan network consistently finds zero error minima. The intermediate and stacked networks find both zero error and high error minima with probabilities that respectively decrease and increase as h/ℓ increases. The maximum error produced by the stacked network, $E_o^{(o)*}$, corresponds to the minimum error achieved by fitting the original points with a horizontal line.

The landscapes produced by larger networks show minima of similar error values but of varying frequency than the landscapes produced by smaller networks (Figure 4c). In comparison, training on the permuted set of original points generates error landscapes whose minima have different error values than those produced using the unpermuted set of original points. Because the values of the original points remain constant under permutation, the stacked network produces the same maximum error value of $E_o^{(o)*}$ (Figure 4e).

These distributions were additionally used to generate the results in the previous section, where the solutions f_o shown in Figure 2 were chosen because their error was representative of the distribution averages in Figures 4a, 4c, and 4e.

Temporal Constraints: To investigate the effect of temporal constraints, we train each network on the original points with 1000 sets of randomly chosen weights but terminate training after 100 iterations. The increased number of randomly initialized states allows us to better resolve the edges of the error distributions shown in Figures 4b, 4d, and 4f.

Once training time is limited, all distributions shift toward higher error values. The stacked network maintains the abrupt cutoff near $E_o^{(o)*}$, while all other distributions extend far beyond this value. Larger networks find minima with similar error values to those found by smaller networks, but they vary in the frequency with which they find these minima. Larger layered networks more frequently produce linear solutions near the cutoff $E_o^{(o)*}$, while larger parallel networks produce more frequent catastrophic error values (Figure 4d). In comparison, training on the permuted set of original points reveals that networks find minima more widely distributed in error but show similar behavior near the edges of the distributions.

C. Dependence on Error Landscape Structure

To better understand differences in network performance in the presence and absence of temporal constraints, we examine the properties of local minima within the error landscapes produced by the five networks shown in Figure 1a.

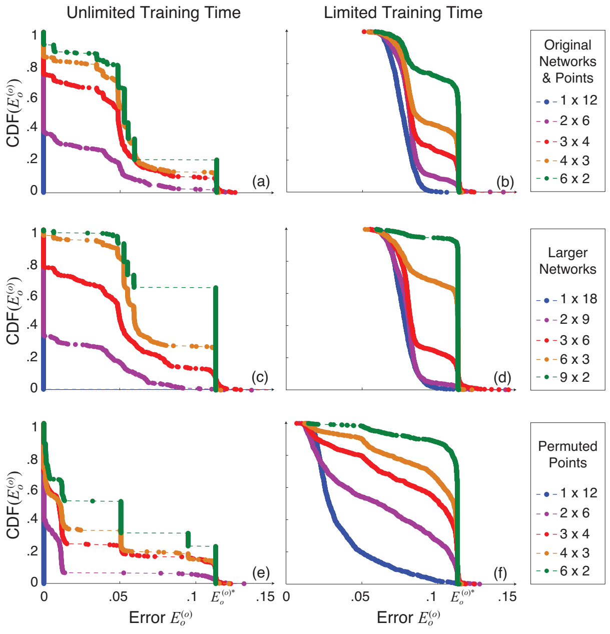

Fig. 4. Distribution of error minima: CDFs of $\{E_o^{(o)}\}$ given (a) unlimited and (b) limited training time for the five networks shown in Figure 1a. (a) The fan network consistently finds zero error solutions, while all other networks find solutions with a range of error values. The stacked network produces a maximum upper limit of $E_o^{(o)*} = 0.1131$, which corresponds to fitting all points with a horizontal line. (b) All distributions shift toward higher error values. The stacked network maintains the hard limit at $E_o^{(o)*}$, while all other networks produce error values that greatly exceed this value. These results are consistent for larger networks, shown in panels (c) and (d), and for permutations of the original points, shown in panels (e) and (f).

Characterizing Landscape Features: We characterize error minima by the direction of highest local landscape curvature, which specifies the combination of weight adjustments that produces the largest change in error. We adopt the terminology used in previous studies and refer to directions with high and low curvature as stiff and sloppy, respectively [20], [21]. Stiff and sloppy directions are found by diagonalizing the error Hessian $H_{pq} = \partial^2 E/\partial\omega_p\partial\omega_q$ evaluated at the set of weights that produces the local error minimum. For computational efficiency, we use the approximate Levenberg-Marquardt (LM) Hessian [22], which agrees well with the stiffest eigenvectors of H and is equivalent to H when a given model perfectly fits data [20], [21]. The LM Hessian takes the form:

$$\frac{\partial^2 E}{\partial\omega_p\partial\omega_q} \approx \sum_{d=1}^{N_D} \frac{\partial r_d^{(o)}}{\partial\omega_p}\frac{\partial r_d^{(o)}}{\partial\omega_q}, \tag{1}$$

where $r_d^{(o)} = (y_d(\vec{\omega}) - y_d^{(o)})$ is the dth residual.

We diagonalize the LM Hessian about each of the 500 minima whose error values $\{E_o^{(o)}\}$ are shown in Figure 4a. Each error minimum produces a set of N_p eigenvalues λ and normalized eigenvectors $\vec{\xi}$, which give the degrees and directions of stiffness in weight space. The following analysis focuses on the stiffest eigenvector $\{\vec{\xi}^{(1)}\}$ as it most strongly controls relevant behavior about each landscape minimum.

Motion along stiff directions may depend on the fraction of network connections that must be significantly adjusted, a quantity measured by the participation ratio $\rho^{(1)} = \sum_q (\xi_q^{(1)})^4$ [23]. $\rho^{(1)}$ is a dimensionless quantity that ranges from a delocalized minimum of $1/N_P$, for which all components have equal weight $1/\sqrt{N_P}$, to a localized maximum of 1, for which a single component carries unit weight.

For the set of minima with error values $\{E_o^{(o)}\}$, we quantify $\{\rho^{(1)}\}$ and $\{\lambda^{(1)}\}$ of $\{\vec{\xi}^{(1)}\}$. The covariances $\mathrm{C}_{E,\rho} = \mathrm{Cov}(E_O^{(O)}, \rho^{(1)})$ and $\mathrm{C}_{E,\lambda} = \mathrm{Cov}(E_O^{(O)}, \lambda^{(1)})$ in these quantities are shown by the ellipses centered about their average values in Figures 5(a) and 5(b), respectively.

Figure 5 highlights the variability in basin structure within and between the networks. As h/ℓ increases, the variance in $\{E_o^{(o)}\}$, $\{\rho^{(1)}\}$, and $\{\lambda^{(1)}\}$ increase. Higher variance leads to lower confidence in predicting the success of the network, but it also suggests that the network has more options when exploring its error landscape. The orientations of covariance ellipses for each landscape provide information regarding the relationships between the error and depth of local minima and the participation of network connections about these minima. For a given value of h/ℓ, larger values of $E_o^{(o)}$ correspond to smaller values of $\lambda^{(1)}$ and larger values of $\rho^{(1)}$. Higher error minima therefore tend to be shallower and require the adjustment of fewer weights.

Landscape Features and Successful Performance: Variations in landscape structure provide insight into the way each network searches for solutions. In particular, fan solutions are characterized by low error and participation ratio, indicating that the fan network must adjust nearly all of its weights in order to navigate zero error basins. In contrast, stacked solutions span a range of error values. The corresponding basins are characterized by a variety of eigenvalues and participation ratios, indicating that the stacked network can navigate many types of basins by adjusting variable numbers of weights. Shallow, high error basins can be found by the stacked network through the adjustment of few connections. Narrow, low error basins, found by both the fan and stacked networks, require fine tuning of a larger number of connections.

Landscape characteristics help explain the results shown in Figures 3 and 4. Given unlimited training time, landscape variability is disadvantageous and can prevent a network from finding a low error minimum. Once time is limited, however, landscape variability can be advantageous in preventing failure by providing the network with high error, shallow basins that can be navigated with the adjustment of relatively few connections.

IV. Discussion

In this study, we investigated the tradeoffs in learning and memory performance that arise from structural complexity. None of the configurations considered here simultaneously mastered both learning and memory tasks, a sensitivity that

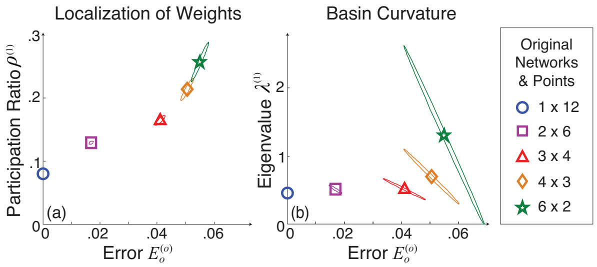

Fig. 5. Properties of error landscapes: covariances in (a) $\{\rho^{(1)}\}$ and $\{E_o^{(o)}\}$ and (b) $\{\lambda^{(1)}\}$ and $\{E_o^{(o)}\}$ for error landscape minima produced by the five networks shown in Figure 1. For each network, the values of $\{E_o^{(o)}\}$ are taken from the distributions shown in panel (a) of Figure 4. Covariances, indicated by ellipses, are centered about their average values, indicated by markers. The semimajor axis of each ellipse marks the direction of maximum covariance. Increasing h/ℓ increases both the average and variance in all three quantities. For a given network, larger values of $E_o^{(o)}$ are often linked to smaller values of $\lambda^{(1)}$ and larger values of $\rho^{(1)}$.

may explain the large variability of architectural motifs evident in large-scale biological and technical systems.

The parallel "fan" and layered "stacked" networks best illustrate the observed tradeoffs in performance. The fan network produces accurate solutions at the cost of potential misrepresentation when retaining and generalizing these solutions. In contrast, the stacked network produces coarser solutions that are more easily retained and generalized. Qualitatively similar behavior is observed for larger networks and for permutations of the original training points, suggesting that these tradeoffs are not a consequence of the network size or specific choice of external information but rather arise solely from variations in structure.

Variations in underlying error landscape structure provides insight into these differences in performance. Deep, narrow landscape minima enable the fan network to produce consistently accurate solutions given unlimited training time. If time is limited, however, the fan network can fail to find these minima and thereby produce highly erroneous solutions. In contrast, variability in depth and curvature of error landscape minima enable the stacked network to quickly find coarse solutions in short amounts of time. If time is unlimited, however, the presence of local minima can hinder the stacked network from finding consistently accurate solutions. While parallel configurations are often preferred in artificial neural network studies due to their efficiency and accuracy, these results suggest the use of layered configurations when performance criteria favor generalizability and minimization of failure over specificity and high accuracy.

The use of small networks and limited training time was crucial to our analysis and allowed us to isolate the performance tradeoffs that we expect to be maintained in larger systems. In particular, the intermediate networks, which are structurally composed of several adjacent stacked networks, share features of both parallel and layered configurations. The performance of these networks may help predict the behavior of larger composite systems, such as cortical layers composed of structurally distinct columns [1] or modular divide-and-conquer networks [24]. In considering complex network systems, we anticipate that underlying structural complexity will continue to impact performance through functional tradeoffs.

Acknowledgment

This work was supported by the David and Lucile Packard Foundation and the Institute for Collaborative Biotechnologies through contract no. W911NF-09-D-0001 from the U.S. Army Research Office.

References

[1] Mountcastle, V B (1997) The columnar organization of the neocortex, *Brain* 120(4):701-722.

[2] Jain, A K, Murty, M N, Flynn, P J (1999) Data clustering: a review, *ACM Comput. Surv.* 31(3):264-323.

[3] Chittka, L, Niven, J (2009) Are bigger brains better?, *Curr. Biol.* 19(21):R995-R1008.

[4] Bassett, D S, et al. (2010) Efficient physical embedding of topologically complex information processing networks in brains and computer circuits, *PLoS Comput Biol* 6(4):e1000748.

[5] Reid, A T, Krumnack, A, Wanke, E, Kotter, R (2009) Optimization of cortical hierarchies with continuous scales and ranges, *Neuroimage.* 47(2):611-617.

[6] Hermundstad, A M, Brown, K S, Bassett, D S, Carlson, J M (2011) Learning, memory, and the role of neural network architecture, *PLoS Comput. Biol.* 7(6):e1002063.

[7] Ress, D, Glover, G H, Liu, J, Wandell, B (2007) Laminar profiles of functional activity in the human brain, *NeuroImage* 34(1):74-84.

[8] Atencio, C A, Schreiner, C E (2010) Columnar connectivity and laminar processing in cat primary auditory cortex, *PLoS One* 5(3):e9521.

[9] Bakoglu, H B (1990) *Circuits, Interconnections, and Packaging for VLSI*, (Addison Wesley).

[10] Galushkin, A I (2007) *Neural Networks Theory*, (Springer, Berlin).

[11] Fukushima, K (1988) Neocognitron: a hierarchical neural network capable of visual pattern recognition, *Neural Networks* 1(2):119-130.

[12] Robinson, A J (1994) An application of recurrent nets to phone probability estimation, *IEEE Trans. Neur. Net.* 5(2):298-305.

[13] Ratcliff, R (1990) Connectionist models of recognition memory: constraints imposed by learning and forgetting functions, *Psychol. Rev.* 97(2):285-308.

[14] Sharkey, N E, Sharkey, A J C (1995), An analysis of catastrophic interference, *Connect. Sci.* 7:301-330.

[15] Robins, A (1995) Catastrophic forgetting, rehearsal, and pseudorehearsal, *Connect. Sci.* 7(2):123-146.

[16] Robins, A, McCallum, S (1998) Catastrophic forgetting and the pseudorehearsal solution in Hopfield-type networks, *Connect. Sci.* 10(2):121-135.

[17] Rojas, R (1996) *Neural Networks: A Systematic Introduction*, (Springer, Berlin).

[18] Fletcher, R, Reeves, C M (1964) Function minimization by conjugate gradients, *Computer J.* 7(2):149-154.

[19] Polak, E, Ribiere, G (1969) Note sur la convergence de methodes de directions conjugees, *Rev. Franc. Informat. Rech. Oper.* 16:35-43.

[20] Brown, K S, Sethna, J P (2003) Statistical mechanical approaches to models with many poorly known parameters, *Phys. Rev. E* 68:021904.

[21] Brown, K S, Hill, C C, Calero, G A, Myers, C R, Lee, K H, Sethna, J P, Cerione, R A (2004) The statistical mechanics of complex signaling networks: nerve growth factor signaling, *Phys. Biol.* 1:184-195.

[22] Fletcher, R (1987) *Practical Methods of Optimization, 2nd ed.*, (Wiley, New York).

[23] Mello, P A, Kuma, N (2004) *Quantum transport in mesoscopic systems: complexity and statistical fluctuations*, (Oxford, New York).

[24] Fu, H-C, Lee, Y-P, Chiang, C-C, Pao, H-T (2001) Divide-and-conquer learning and modular perceptron networks, *IEEE Tran. Neural Networks* 12(2):250-263.