Controlled vs. Automatic Processing: A Graph-Theoretic Approach to the Analysis of Serial vs. Parallel Processing in Neural Network Architectures

Sebastian Musslick[1,*], Biswadip Dey[2,*], Kayhan Özcimder[1,2,*],
Md. Mostofa Ali Patwary[3], Theodore L. Willke[3], and Jonathan D. Cohen[1]
[1]Princeton Neuroscience Institute, Princeton University, Princeton, NJ 08544, USA.
[2]Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, NJ 08544, USA.
[3]Parallel Computing Lab, Intel Corporation, Santa Clara, CA 95054, USA.
[*]Equal Contribution, Corresponding Author: *musslick@princeton.edu*

Abstract

The limited ability to simultaneously perform multiple tasks is one of the most salient features of human performance and a defining characteristic of controlled processing. Based on the assumption that multitasking constraints arise from shared representations between individual tasks, we describe a graph-theoretic approach to analyze these constraints. Our results are consistent with previous numerical work (Feng, Schwemmer, Gershman, & Cohen, 2014), showing that even modest amounts of shared representation induce dramatic constraints on the parallel processing capability of a network architecture. We further illustrate how this analysis method can be applied to specific neural networks to efficiently characterize the full profile of their parallel processing capabilities. We present simulation results that validate theoretical predictions, and discuss how these methods can be applied to empirical studies of controlled vs. and automatic processing and multitasking performance in humans.

Keywords: multitasking; cognitive control; capacity constraint

Introduction

The human ability to carry out multiple tasks concurrently – a longstanding focus of cognitive research – presents an interesting puzzle. In some domains, humans can fluidly execute a large number of behaviors concurrently (e.g., locomote, navigate, talk, and bimanually gesticulate). However, in other domains, this capacity is strikingly limited (e.g., conduct mental arithmetic while constructing a grocery list). In addition to their obvious practical importance, constraints on multitasking are also of theoretical significance. Any general theory of cognition must address how choices are made among the limited set of behaviors that can be carried out at a given time (Anderson, 2013; Lieder & Griffiths, 2015; Kurzban, Duckworth, Kable, & Myers, 2013; Shenhav, Botvinick, & Cohen, 2013), and thus the sources of such limitations occupy a central role in cognitive theory.

Whether a set of tasks can or cannot be carried out concurrently has often been attributed to a fundamental distinction between automatic and controlled processing, with the former relying on parallel processing mechanisms and the latter on a limited capacity, serial processing (Posner & Snyder, 1975; Shiffrin & Schneider, 1977). However, this begs a fundamental question: why are control-dependent processes capacity limited? Early theories, as well as some of the most successful unified theories of cognition (e.g., ACT-R) have assumed that this constraint reflects an intrinsic, structural property of the control system itself (e.g., limited capacity of working memory). However, alternative accounts have suggested that limitations in multitasking capacity reflect local properties of the mechanisms used for task execution, rather than an intrinsic property of the control system itself. According to such accounts, constraints on multitasking arise when two tasks call upon the same local resources (e.g., representations specific to the tasks) for different purposes (Allport, 1980; Meyer & Kieras, 1997; Navon & Gopher, 1979; Salvucci & Taatgen, 2008) and thus cannot be performed at the same time.[1]

Building on this idea, it has been proposed that a fundamental purpose of control mechanisms is to prevent crosstalk, by limiting the engagement of representations used by multiple processes ("mulituse representations") to a single purpose (e.g., task) at any given time (e.g. Cohen, Dunbar, & McClelland, 1990; Botvinick, Braver, Barch, Carter, & Cohen, 2001). From this perspective, constraints on multitasking of control-demanding processes reflect the *purpose* of control, rather than an intrinsic limit in control mechanisms. To the extent that the processing pathways required to perform different tasks rely on shared (i.e., multiuse) representations, not only do they become increasingly reliant on control (to specify the current intended use, and avoid crosstalk from competing uses), but the multitasking capacity of the network becomes limited (i.e., driven toward serial processing). In other words, control mechanisms are guilty by association, rather than themselves the source of constraints on multitasking.

One question that might be asked is: how does the constraint on multitasking imposed by pathway overlap scale with network size? A naive assumption might be that, in large networks (such as the brain), the constraint is relatively weak, and thus is inadequate to explain the prohibitive constraints apparent in human control-dependent processing. We have addressed this question in previous work, by examining the effects of pathway overlap (multiuse of representations) in

[1]Multitasking (and apparent parallelism) can, in some situations, be achieved by rapid switching between serial processes (as is common in computers). Here, we focus on forms of multitasking that reflect truly concurrent processing, sometimes referred to as perfect timesharing or pure parallelism. In the General Discussion, we consider how our findings concerning the conditions for such parallelism relate to the capability for rapid serial processing.

two types of networks of varying size (Feng et al., 2014). We found that even modest degrees of pathway overlap produced a strikingly strong constraint on parallel processing that was nearly scale invariant. This supported the idea that constraints in human multitasking may reflect representational multiuse, rather than a limitation intrinsic to control mechanisms themselves. However, while this work was suggestive, it relied on numerical simulations that were restricted to a limited range of parameters. It also failed to provide a clear path from these theoretical ideas to empirical validation.

Here, we conduct an exhaustive analysis of the relationship between pathway overlap and parallel processing in single-layered, feed-forward, non-linear networks. Our findings validate and extend those of Feng et al. (2014), identifying additional factors that influence the relationship between pathway overlap and parallel processing capability. We also show how these analysis methods can be used to fully specify the multitasking capabilities of a network, and validate derived theoretical predictions in simulated neural networks. Critically, we suggest how this method could also be applied to empirical data to determine the multitasking capabilities of natural agents in realistically large task spaces. Finally, we discuss related results using these methods to examine the interaction between pathway overlap, learning and generalization.

Graph-Theoretic Approach to Parallel vs. Serial Processing Capability

Following Feng et al. (2014), we consider single-layered, feedforward networks with N input and N output layer components. Each component represents an input or output dimension (vector subspace), and the connection from an input to an output component constitutes the processing pathway for a given task (defined as a unique mapping from all possible vectors in input subspace to corresponding vectors in the output subspace, that is independent of the mappings for all other combinations of input and output components in the network). The network can be represented as a directed bipartite graph $\mathcal{G}_B = (\mathcal{V}, \mathcal{E})$, in which the node set $\mathcal{V}$ can be partitioned into two disjoint sets of nodes $\mathcal{V}_{in}$ and $\mathcal{V}_{out}$, representing the input and output layer components respectively. Moreover, an edge $(i, j) \in \mathcal{E} \subseteq \mathcal{V}_{in} \times \mathcal{V}_{out}$ represents a directed pathway from the input layer to the output layer in the network (i.e., a task). We introduce the matrix $A = [a_{ij}] \in \{0,1\}^{N \times N}$ to represent the network structure and define its elements such that $a_{ij} = 1$ when, $(i, j) \in \mathcal{E}$, $i \in \mathcal{V}_{in}$, $j \in \mathcal{V}_{out}$ and $a_{ij} = 0$ otherwise.

Pathway Overlap and Interference

The matrix A, extracted from the adjacency matrix of the bipartite graph, captures the overall network structure, since by definition the graph is directed and has no self-loops. In particular, it represents the degree to which pathways overlap (i.e., share representations): the sum of each row of matrix A reflects the multiuse of input representations (out-degree of input nodes), the sum of each column reflects the same for output representations (in-degree of output nodes), and together these indicate the degree of pathway overlap in the network. We assume that such overlap produces interference, prohibiting performance of the tasks involved. We formalize three types of interference, as shown in Fig 1. Convergent interference (shown in green) occurs when two sources of input compete to determine a common output. In addition, we consider divergence (shown in red) as a form of interference in our analysis. Although this does not pose an impediment to performance (i.e., it is possible to generate two distinct responses to the same input), it represents a restriction on the number of independent sources of input (and therefore number of tasks) that the system can process at once, and thus can be treated formally as a type of interference in our analysis of multitasking capability. Finally, we consider a third, indirect form of interference that supervenes on the first two (shown in blue). In this case, the two tasks in question do not directly interfere with one another. However, their simultaneous engagement would necessarily engage a third task (shown in purple) that would produce interference; accordingly the two tasks shown in blue can not be performed simultaneously. It is important to note that not only the *amount* of interference (of the forms just described), but also how it is distributed over the network impacts multitasking performance. Here, for simplicity, we assume a uniform distribution of pathways among the input and output components[2], which means the pathway overlap P is equal to the in-degree and out-degree of each component in the network.

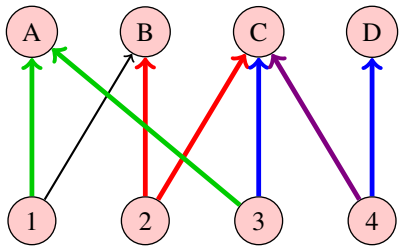
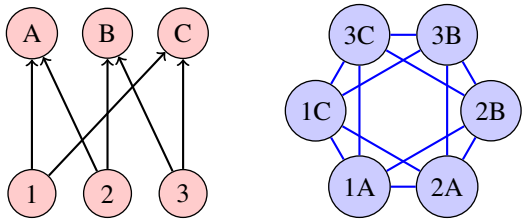


Figure 1: Illustration of the three types of interference considered in our analysis (see *text*).

To quantify multitasking capability, we begin by constructing an interference graph $\mathcal{G}_I$ associated with the original bipartite graph $\mathcal{G}_B$. By assigning each edge of the original graph $\mathcal{G}_B$ to a node in $\mathcal{G}_I$, each node in the $\mathcal{G}_I$ is used to represent a task. Interference between tasks is then represented by assigning edges to pairs of nodes in $\mathcal{G}_I$ if the tasks represented by those nodes are subject to any of the three forms of interference defined above (formally, this corresponds to the square of the line graph of $\mathcal{G}_B$). The adjacency relationships between nodes in $\mathcal{G}_I$ thus describe which tasks in the original network can be executed concurrently (i.e., in parallel). This, in turn, can be used to identify the maximum multitasking capability of the network, as discussed in the next section.

[2]Such a uniform distribution is also reflective of a relatively broad range of distributions in constraining multitasking.



(*a*)Bipartite Graph - $\mathcal{G}_B$ (*b*)Interference Graph - $\mathcal{G}_I$

Figure 2: The first subfigure (2a) illustrates a network of size $N = 3$ and pathway overlap $P = 2$ (in-degree and out-degree of each component is 2). The second subfigure (2b) shows the associated interference graph with 6 nodes, and each of these nodes has 4 neighbors due to interferences.

Maximum Independent Set (MIS) as a Measure of Multitasking Capability

Identifying the multitasking (maximum parallel processing) capability in the original network can be cast as the problem of finding the largest set of nodes in the interference graph wherein no two nodes are adjacent. This is formally known as the *maximum independent set* (MIS). Finding the MIS of a graph is an NP-hard problem, that has been studied extensively in the graph theory literature (Tarjan & Trojanowski, 1977). Figure 3 summarizes the effect of pathway overlap and network size on the MIS for networks with uniform pathway distribution (comparable to Feng et al., 2014), confirming that parallel processing capability is severely constrained by pathway overlap in a manner that is virtually scale invariant for network size (source code available at `github.com/musslick/CogSci-2016`). In the sections that follow, we show how these analysis tools can be used to infer the particular parallel processing capabilities of specific networks, validate predictions made based on extracted interference graphs in simulations of network multitasking performance, and describe how these tools could be used to infer similar information regarding human performance from neuroimaging data.

Application to Neural Network Models

In the previous section we introduced graph-theoretic analyses to investigate factors affecting the parallel processing capability in simplified network structures. Here, we examine the extent to which these analyses can be applied to more complex models (such as artificial neural networks) and empirical data (e.g. neuroimaging analyses). We will describe how neural representations of tasks can be used to generate predictions about how many and which combinations of tasks a network (or person) can perform in parallel (a space of possibilities that grows combinatorially with the number of tasks, and thus quickly becomes intractable to direct empirical inquiry), based on measurements of single task performance (that grows only linearly in the number of tasks). These analyses may provide useful diagnostic tools for exhaustively assessing multitasking capabilities based on amounts of data that are practical to acquire.

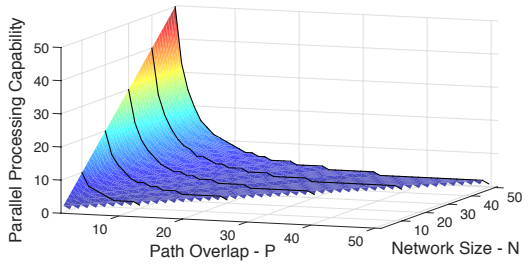


Figure 3: Network parallel processing capability as a function of pathway overlap (P) and network size (N) for networks with uniform pathway distributions.

Network Architecture and Processing

We focus on a network architecture that has been used to simulate a wide array of empirical findings concerning human performance (e.g. Cohen et al., 1990; Cohen, Servan-Schreiber, & McClelland, 1992; Botvinick et al., 2001). Such networks typically consist of four layers (see Figure 4): an input layer with two partitions, one of which represents the current stimulus and projects to an associative layer, and another that encodes the current task and projects to both the associative and output layers; an associative (hidden) layer that projects to the output layer; and an output layer that represents the network's response. Input units are clamped to either 0 or 1 to represent the current stimulus and task. These values are multiplied by the matrix of connection weights from the input layers to the associative layer, and then passed through a logistic function to determine the pattern of activity over the units in the associative layer. This pattern is then used (together with projections from the task units in the input layer) to determine the pattern of activity over the output layer. The latter provides a response pattern that is evaluated by computing its mean squared error (MSE) with respect to the correct (task-determined) output pattern.

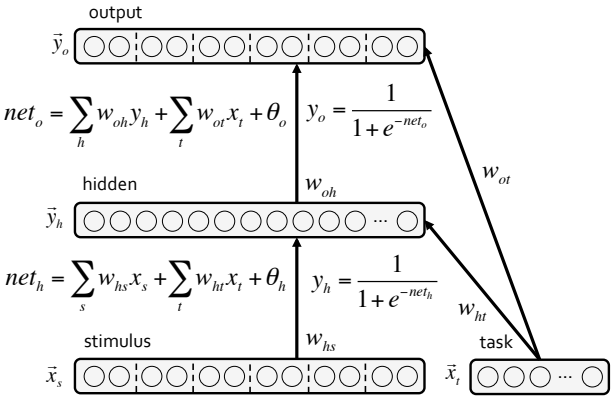


Figure 4: Feedforward neural network used in simulations. The input layer is composed of stimulus vector $\overrightarrow{x_s}$ and task vector $\overrightarrow{x_t}$. The activity of each element in the associative layer $y_h \in \overrightarrow{y_h}$ is determined by all elements x_s and x_t and their respective weights w_{hs} and w_{ht} to y_h. Similarly, the activity of each output unit $y_o \in \overrightarrow{y_o}$ is determined by all elements y_h and x_t and their respective weights w_{oh} and w_{ot} to y_o. A bias of $\theta = -2$ is added to the net input of all units y_h and y_o.

Stimulus input units are structured according to dimensions (subvectors of the stimulus pattern), each of which is comprised of a set of feature units with only one feature unit activated per dimension. Similarly, output units are organized into response dimensions, with only one response unit permitted to be active per dimension. Each task is represented by a single task input unit that is associated with a set of unique, one-to-one mappings between the input units in one stimulus dimension and the output units in one response dimension, and that is independent of the mappings for all other tasks. Here, we focus on networks ($N = 6$) in which there are six input dimensions comprised of two features each, and six output dimensions comprised of two responses each. Such networks support a total of $6 * 6 = 36$ possible tasks; and, since each stimulus input dimension consists of two features, $2^6 = 64$ possible input patterns per task (including both task-relevant and task-irrelevant features). The number of hidden layer units in each network is set to 200 to avoid constraining the network to low-dimensional mappings of the input space. Networks are initialized with a set of small random weights and then trained using the backpropagation algorithm (David E. Rumelhart & Williams, 1986) to produce the task-specified response for all stimuli in each task. That is, the network is trained to generate the response for the corresponding stimulus in the task-relevant dimension, while suppressing responses in all other response dimensions.

Extracting Directed Bipartite Graph from Task Representations

Our analysis focuses on the representations (patterns of activity) over the associative and output units, insofar as these reflect the computations carried out by the network required to perform each task. In particular, we are interested in the characteristics of these representations for each task, how they compare across tasks, and how these factors correspond to multitasking performance. The representations associated with each task can be characterized by calculating, for each unit in the associative and output layers, the mean of its activity over all of the stimuli for a given task; this mean pattern of activity can then be used as a representation of the task[3]. Correlating these patterns of activity across tasks yields a task similarity matrix that can be examined separately for the associative and output layers of the network. This can then be used to assess the extent to which different tasks rely on similar or different representation within each layer of the network. Figure 5 provides an example of such similarity matrices (thresholded for similarity correlations above $r > 0.8$). Tasks that have similar representations over the associative layer can be inferred to rely on the same input dimension – that is, they share an input component in the bipartite graph representation of the network – and tasks that are similar at the output layer can be inferred to share an output component.

[3]A formally equivalent analysis could be carried out using the weight matrix of the network. Here we focus on patterns of activity, as these may serve as useful predictors for patterns of activity observed in empirical data, such as fMRI and/or unit recordings.

Accordingly, a bipartite graph (of the type shown in Figure 3) can be constructed by measuring the patterns of activity observed in the network while it performs each individual task. This can then be analyzed, using the graph-theoretic methods described above, to examine the full multitasking profile of the network – that is, both the maximum concurrency (parallel processing) capability of the network and, perhaps more interestingly, the exact profile of which combinations of tasks can and cannot be performed concurrently (see Figure 6). This procedure is substantially more efficient, and scales more gracefully (linearly with size of the network) than determining the multitasking profile by simulating and examining performance of the network for all combinations of tasks (which scales factorially).

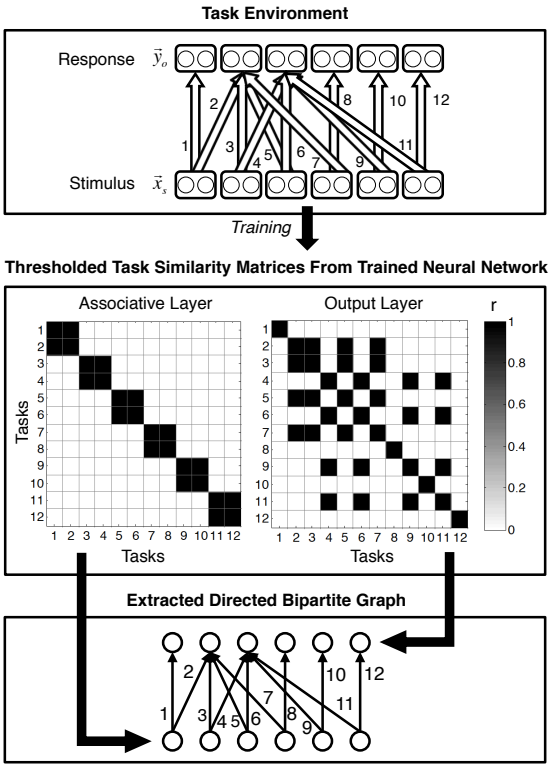


Figure 5: A task environment consists of 12 possible tasks represented as input-output mappings. The bipartite graph can be extracted from thresholded task similarity matrices that are obtained from task activity correlations at the associative layer and output layer of a trained network.

Simulation Experiment

To validate the methods described above, we compared simulated network performance with analysis predictions for 100 networks of size $N = 6$, each trained on a different subset of 12 randomly sampled tasks (source code available at `github.com/musslick/CogSci-2016`). Tasks were chosen subject to the constraint that each stimulus dimension was associated with two tasks. For each network we extracted a bipartite graph from the task similarity matrices as outlined above. Figure 5 shows the results for an example network, from which a bipartite graph was generated that recovered

the exact task structure imposed during training. That is, the network learned to use similar associative layer representations for tasks that involved the same stimulus dimension (e.g. tasks 1 & 2 in Figure 5), and learned similar output representations for tasks involving the same response dimensions (e.g. tasks 2, 3, 5 & 7 in Figure 5).

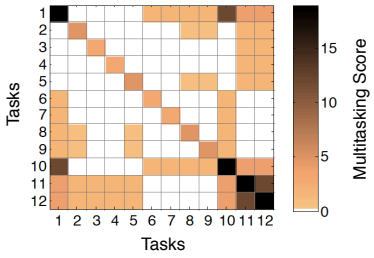


Figure 6: Extracted adjacency matrix of interference graph. Off-diagonal colored entries indicate all tasks that can be paired with a given task. Colors of off-diagonal elements indicate the number of all possible multitasking conditions in which the corresponding task-pairing can occur. Colors of diagonal elements indicate the number of all multitasking combinations in which the corresponding task can occur.

For each network, a bipartite graph can be constructed and used to extract a corresponding interference graph. The adjacency matrix of the latter indicates which tasks can be paired. This can be used, in turn, to identify all combinations of tasks that can be performed concurrently (see example in Figure 6), as well as the MIS which specifies the greatest number of tasks that can be performed concurrently. For each of the 100 trained networks, we extracted its interference graph and computed the multitasking performance (MSE) for all combinations of tasks that belonged to an independent set. We compared this to a random sample (of identical size) composed of combinations of tasks in which two or more of the tasks did not belong to an independent set. Figure 7 shows that these analyses yield accurate predictions about which tasks can be performed concurrently and which cannot. As the network was never trained on multitasking, concurrent performance of tasks from independent sets can still produce some error. However, the performance for those tasks is markedly and reliably better than multitasking performance for combinations of tasks in which not all belong to the same independent set, $t(98) = 232.34, p < .001$ (2 tasks); $t(98) = 132.03, p < .001$ (3 tasks); $t(79) = 29.64, p < .001$ (4 tasks).

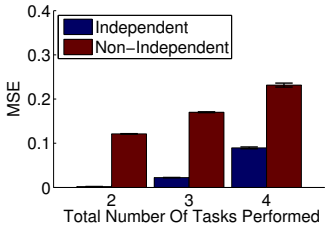


Figure 7: Multitasking performance for sets of identified independent and non-independent tasks. Error bars indicate the standard error of the mean for multitasking conditions of networks trained in different task environments.

General Discussion and Conclusion

We have introduced a graph-theoretic approach to compute the multitasking (parallel processing) capability of feedforward, single-layer non-linear networks. This was achieved by generating an interference graph from the directed bipartite graph representation of the network, which provides a compact representation of its multitasking capabilities. Identifying the MIS in the interference graph reveals the maximum number of concurrent tasks that can be executed without performance loss. The interference graph can also be used to identify all combinations of tasks that can be performed in parallel. We have shown that, consistent with previous work (Feng et al., 2014), introducing even modest amounts of pathway overlap induces dramatic constraints on multitasking capability. We then illustrated how the graph-theoretic analysis can be applied to specific networks, using the patterns of activity associated with individual tasks to characterize the full profile of the network's multitasking capabilities.

At a practical level, these methods suggest possibilities for empirical research. For example, if patterns of neural activity (measured using direct neuronal recordings and/or fMRI) can be identified for a set of individual tasks, then the analyses described above can be used to predict multitasking performance for all possible combinations of tasks in the set. The measurements required to carry out this analysis grow linearly with the number of tasks in the set, whereas the number of measurements required to characterize the interactions among them from behavior would grow factorially with set size. That is, these analyses may be particularly useful in situations in which exhaustively assessing the entire space of task combinations is empirically impractical.

Theoretically, the approach provides a formal framework for studying the relationship between learned task representations and controlled (serial) vs. automatic (parallel) processing. Specifically, it permits quantitative analysis of how task environment and learning impact the tradeoff between compactness of representation (associated with serial, control-dependent processing) and multitasking capability (associated with parallel, automatic processing). In related work others (e.g. Caruana, 1997; Bengio, Courville, & Vincent, 2013; Saxe, McClelland, & Ganguli, 2013) have shown that compact, "multiuse" representations not only make more efficient use of network resources (e.g. fewer associative units) but also are likely to arise most quickly (especially in hierarchically structured environments), and support greater generalization during learning. However, the present work illustrates the costs this incurs with regard to parallelism of processing: as pathway overlap (mulituse of representations) increases, processing in the network is rapidly driven to be serial, and becomes reliant on control mechanisms to avoid cross-talk. We suggest that this tension underlies the tradeoff between controlled and automatic processing observed in human performance, and that constraints on the capacity for human multitasking reflect this tension.

While we have focused on forms of multitasking arising

from concurrent parallelism, our findings are likely to have implications even when multitasking is achieved by rapidly switching between tasks. One of the most robust findings in the cognitive literature is the cost associated with switching between tasks, reflecting at least in part carry-over effects that the representations of one task have on the next (Yeung, Nystrom, Aronson, & Cohen, 2006, for a review see Kiesel et al., 2010). To the extent that such carry-over effects reflect interference from shared representations, then this may determine the speed and/or accuracy with which sequential switching can be achieved in a manner similar to its impact on pure parallelism. That is, multiuse of representations may define a continuum from pure sequential processing, though rapid task switching, to pure parallelism, and the methods we describe may provide a way of analyzing networks to determine where they lie along this continuum.

There are a variety of network parameters that can impact the extent to which multiuse representations arise during training (including weight initialization, regularization constraints, number of hidden units, etc.). Here, we have focused on a simple network parameterization (two-layered, feedforward, with random weight initialization and no regularization) as a first assessment of the usefulness of the analytic approach. Another simplification in our treatment was the construction of binary bipartite and interference graphs, by thresholding the real-valued correlation matrix of network representations. Additional simulation results (not reported) suggest that the analysis methods we report are robust across a wide range of thresholds and learned task representations. However, a generalization of the method to address graded interference effects (e.g., using weighted graphs) is an important avenue for future research. More generally, it will be important to explore the extent to which the methods and analyses we describe can be extended to networks with more complex and realistic architectures (e.g., multi-layered and/or recurrent, with varying pathway distributions and graded degrees of interference). We hope that the work described here will encourage a proliferation of efforts along these lines.

References

Allport, D. A. (1980). Attention and performance. *Cognitive psychology: New directions*, *1*, 12–153.

Anderson, J. R. (2013). *The architecture of cognition*. Psychology Press.

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *35*(8), 1798–1828.

Botvinick, M. M., Braver, T. S., Barch, D. M., Carter, C. S., & Cohen, J. D. (2001). Conflict monitoring and cognitive control. *Psychological review*, *108*(3), 624.

Caruana, R. (1997). Multitask learning. *Machine learning*, *28*(1), 41–75.

Cohen, J. D., Dunbar, K., & McClelland, J. L. (1990). On the control of automatic processes: a parallel distributed processing account of the stroop effect. *Psychological review*, *97*(3), 332.

Cohen, J. D., Servan-Schreiber, D., & McClelland, J. L. (1992). A parallel distributed processing approach to automaticity. *The American journal of psychology*, 239–269.

David E. Rumelhart, G. E. H., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*, 533–536.

Feng, S. F., Schwemmer, M., Gershman, S. J., & Cohen, J. D. (2014). Multitasking vs. Multiplexing: Toward a normative account of limitations in the simultaneous execution of control-demanding behaviors. *Cognitive, Affective, & Behavioral Neuroscience*, *14*(1), 129-146.

Kiesel, A., Steinhauser, M., Wendt, M., Falkenstein, M., Jost, K., Philipp, A. M., & Koch, I. (2010). Control and interference in task switchinga review. *Psychological bulletin*, *136*(5), 849.

Kurzban, R., Duckworth, A., Kable, J. W., & Myers, J. (2013). An opportunity cost model of subjective effort and task performance. *The Behavioral and brain sciences*, *36*(6), 661–679.

Lieder, F., & Griffiths, T. L. (2015). When to use which heuristic: A rational solution to the strategy selection problem. In *37th cognitive science society conference, tx.*

Meyer, D. E., & Kieras, D. E. (1997). A computational theory of executive cognitive processes and multiple-task performance: Part i. basic mechanisms. *Psychological review*, *104*(1), 3.

Navon, D., & Gopher, D. (1979). On the economy of the human-processing system. *Psychological review*, *86*(3), 214.

Posner, M., & Snyder, C. (1975). attention and cognitive control,. In *Information processing and cognition: The loyola symposium* (pp. 55–85).

Salvucci, D. D., & Taatgen, N. A. (2008). Threaded cognition: an integrated theory of concurrent multitasking. *Psychological review*, *115*(1), 101.

Saxe, A. M., McClelland, J. L., & Ganguli, S. (2013). Learning hierarchical category structure in deep neural networks. In *Proceedings of the 35th annual meeting of the cognitive science society* (pp. 1271–1276).

Shenhav, A., Botvinick, M. M., & Cohen, J. D. (2013). The expected value of control: an integrative theory of anterior cingulate cortex function. *Neuron*, *79*(2), 217–240.

Shiffrin, R. M., & Schneider, W. (1977). Controlled and automatic human information processing: Ii. perceptual learning, automatic attending and a general theory. *Psychological review*, *84*(2), 127.

Tarjan, R. E., & Trojanowski, A. E. (1977). Finding a Maximum Independent Set. *SIAM Journal on Computing*, *6*(3), 537-546. doi: 10.1137/0206038

Yeung, N., Nystrom, L. E., Aronson, J. A., & Cohen, J. D. (2006). Between-task competition and cognitive control in task switching. *The Journal of Neuroscience*, *26*(5), 1429–1438.