# Multitasking Capability Versus Learning Efficiency in Neural Network Architectures

Sebastian Musslick[1,*], Andrew M. Saxe[2], Kayhan Özcimder[1,3],
Biswadip Dey[3], Greg Henselman[4], and Jonathan D. Cohen[1]

[1]Princeton Neuroscience Institute, Princeton University, Princeton, NJ 08544, USA.
[2]Center for Brain Science, Harvard University, Cambridge, MA 02138, USA.
[3]Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, NJ 08544, USA.
[4]Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104, USA.
[*]Corresponding Author: *musslick@princeton.edu*

**Abstract**

One of the most salient and well-recognized features of human goal-directed behavior is our limited ability to conduct multiple demanding tasks at once. Previous work has identified overlap between task processing pathways as a limiting factor for multitasking performance in neural architectures. This raises an important question: insofar as shared representation between tasks introduces the risk of cross-talk and thereby limitations in multitasking, why would the brain prefer shared task representations over separate representations across tasks? We seek to answer this question by introducing formal considerations and neural network simulations in which we contrast the multitasking limitations that shared task representations incur with their benefits for task learning. Our results suggest that neural network architectures face a fundamental tradeoff between learning efficiency and multitasking performance in environments with shared structure between tasks.

**Keywords:** multitasking; cognitive control; capacity constraint; learning; neural networks

## Introduction

Our limited capability to execute multiple tasks at the same time highlights one of the most fundamental puzzles concerning human processing, which must be addressed by any general theory of cognition (Shenhav, Botvinick, & Cohen, 2013; Kurzban, Duckworth, Kable, & Myers, 2013; Anderson, 2013): Why, for some tasks, is the human mind capable of a remarkable degree of parallelism (e.g., navigating a crowded sidewalk while talking to a friend), while for others its capacity for parallelism is radically limited (e.g., conduct mental arithmetic while constructing a grocery list)?

Early theories of cognition, that have continued to be highly influential, assert that the ability to multitask – that is, to carry out a set of tasks concurrently[1] – can be understood in terms of a fundamental distinction between automatic and controlled processing, with the former relying on parallel processing mechanisms (that can support multitasking) and the latter assumed to rely on a serial processing mechanism with limited capacity (Posner & Snyder, 1975; Shiffrin & Schneider, 1977) that can only support processing of a single task at a time. In this view, the constraints on the number of control-dependent tasks that can be executed at one time reflect an intrinsic property of the control system itself. However, alternative ("multiple-resource") accounts (Allport, 1980; Meyer & Kieras, 1997; Navon & Gopher, 1979; Salvucci & Taatgen, 2008) have suggested that multitasking limitations arise from local processing bottlenecks. That is, if two tasks share the same local resources (i.e. representations required to perform the tasks), then executing them simultaneously can lead to cross-talk and degraded performance. It has been argued that the very purpose of cognitive control is to prevent such cross-talk by limiting the number of active task processes engaged (Cohen, Dunbar, & McClelland, 1990; Botvinick, Braver, Barch, Carter, & Cohen, 2001). In this view, constraints in multitasking reflect the *consequences* of control doing its job, rather than limitations intrinsic to the mechanisms of control *itself*. This line of argument suggests that, to better understand the conditions under which multitasking is and is not possible, it is necessary to understand the extent to which the task processes involved share representations, and are thus subject to potential interference and the intervention of control to limit processing. This, in turn, raises the question of whether there are general principles of neural architectures that determine the use of shared representation, and how these interact with learning and processing.

One may argue that the constraints that shared representations impose on multitasking are negligibly small in a processing system as large as the human brain. However, simulation studies (Feng, Schwemmer, Gershman, & Cohen, 2014), followed by analytic work (Musslick et al., 2016) have found that the multitasking capability of a network can drop precipitously as a function of overlap between task processes (i.e. number of shared representations), and that this effect is relatively insensitive to the size of the network.

The findings above suggest that maximal parallel processing performance is achieved through the segregation of task pathways, by separating the representations on which they rely. This raises an important question: insofar as shared representation introduces the risk of cross-talk and thereby limitations in parallel processing performance, why would the brain prefer shared task representations over separate ones? Insights gained from the study of learning and representation in neural networks provide a direct answer to this question:

[1]Multitasking can, in some situations, be achieved by rapid sequential processing (e.g., switching between asynchronous serial processes, as is common in computers), rather than through true synchronous processing. Here, our focus is on forms of multitasking that reflect truly concurrent processing, sometimes referred to as perfect timesharing or pure parallelism.

Shared representations across tasks can support inference and generalization (Caruana, 1997). These benefits are strongly linked to the ability of neural networks to carry out "interactive parallelism", that is, the ability to learn and to process complex representations by simultaneously taking into account a large number of interrelated and interacting constraints (McClelland, Rumelhart, & Hinton, 1986).

In this study, we examine the tension between interactive parallelism that promotes learning efficiency through use of shared representations, on the one hand, and "independent parallelism" (i.e. the ability to carry out multiple processes independently), on the other hand. That is, we are interested in studying biases that promote shared representations over multitasking performance. We first demonstrate that the well-recognized (and valued) emergence of shared representations (Hinton, 1986) in response to extrinsic biases (i.e. shared structure in the task environment) leads to constraints in multitasking performance. In the second part, we introduce a formal characterization of a tradeoff between learning efficiency and multitasking performance and examine how intrinsic biases of the network toward the use of shared representations can expose this tradeoff in neural network simulations. The source code for all simulations is available at `github.com/musslick/CogSci-2017`.

## Neural Network Model

For the simulations described in the paper 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; Botvinick et al., 2001), including recent work on limitations in multitasking (Musslick et al., 2016). In this section we lay out the architecture of this network, its processing, as well as the task environments used to train it.

### Network Architecture and Processing

The network consists of two input layers, one of which represents the stimulus presented to the network and another that encodes the task that the network is instructed to perform on the stimulus. Stimulus input features can take any real value between 0 and 1 and can be grouped into stimulus dimensions that are relevant for a particular task. The network is instructed to perform a single task by clamping the corresponding task unit in the task layer to 1 while all other task units are set to 0. These stimulus and task input values are multiplied by a matrix of connection weights from the respective input layer to a shared 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 a set of direct projections from the task 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. Similar to stimulus features, output units can be grouped into response dimensions that are relevant for a particular task. Note that the weight projections from each task unit can act as control signals that bias processing towards task-relevant stimulus information represented at the associative and output layer.

In order to represent the task environment described below, the stimulus layer is compromised of 45 input units (features) and the task layer of nine task units. The output layer consists of 15 units and is organized into three response dimensions (with five units per response dimension.). The number of units in the associative layer is set to 100.

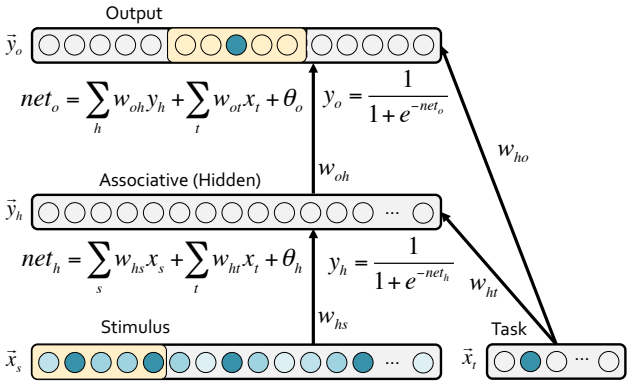


Figure 1: Feedforward neural network used in simulations. The input layer is composed of stimulus vector $\overrightarrow{x_s}$ and task vector $\overrightarrow{x_t}$. The activity of each element in the associative layer $y_h \in \overrightarrow{y_h}$ is determined by all elements $x_s$ and $x_t$ and 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 $y_o$. Blue shades in the input and output units (circles) correspond to unit values of $> 0$ and illustrate an example input pattern with its respective output pattern: The second task requires the network to map the vector of values in the first five stimulus input units to one out of five output units (yellow shade).

### Task Environment

Each task is defined as a mapping between a subspace of five stimulus features (referred to as a task-relevant stimulus dimension) onto five output units of a task-specific response dimension, so that only one of the five relevant output units is permitted to be active (see Fig. 1). The value of each stimulus feature is drawn from a uniform distribution $U[0,1]$. The rule by which 5 relevant stimulus features of any task-relevant stimulus dimension are mapped onto one of the 5 output units of the task-relevant response dimension corresponds to a non-linear function that was randomly generated[2] with a separate "teacher" network (cf. Seung, Sompolinsky, & Tishby, 1992), and is the same across tasks. However, tasks are considered to be independent in that they differ which stimulus dimension is linked to which response dimension.

The task environment across all simulations encompasses nine tasks. As illustrated in Fig. 2 groups of three tasks map

[2]Note that it is ensured that, for the uniform distribution $U[0,1]$ of stimulus unit activations in the task-relevant set of input units, every relevant output unit is equally likely to be required for execution.

onto the same response dimension. However, similarity between tasks could be varied by manipulating the overlap between their relevant stimulus dimensions. At the extremes, task environments can be generated such that tasks of different response dimensions relate to separate stimulus features (no feature overlap, Fig. 2a), or the same stimulus features (full feature overlap, e.g. tasks 1-3 in Fig. 2b).

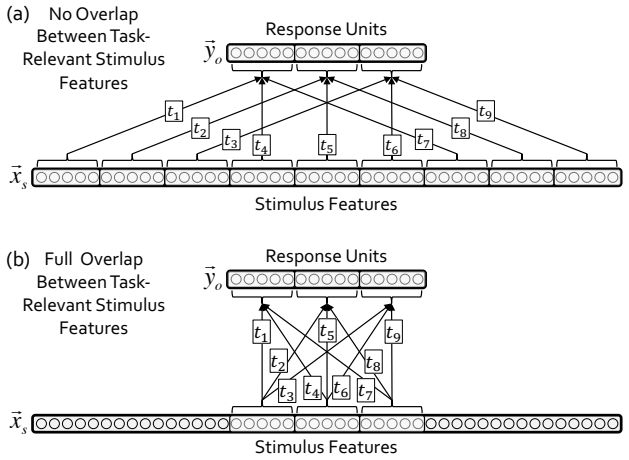


Figure 2: Task environments. For each task, the network was trained to map a subset of 5 stimulus features onto a subset of 5 output units. At the extremes tasks that were mapped onto different response dimensions (e.g. tasks 1-3) could either (a) rely on separate stimulus features or (b) completely overlap in terms of their relevant stimulus features.

Networks are initialized with a set of small random weights and then trained on all tasks using the backpropagation algorithm[3] (Rumelhart & Geoffrey E. Hinton, 1986) to produce the task-specified response for each stimulus.

## Multitasking Limitations Due to Shared Structure in the Task Environment

A key feature of neural networks is their ability to discover latent structure in the task environment, exploiting similarity between stimulus features in the form of shared representations (Hinton, 1986; Saxe, McClelland, & Ganguli, 2013). In this section we explore how the emergence of shared representations as a function of structural similarities between tasks can impact the multitasking performance of a network.

### Simulation Experiment 1: Shared Task Representations as a Function of Feature Overlap

In order to investigate the effect of structural similarities between tasks we generated task environments with varying overlap between task-relevant stimulus features. We define feature overlap as the number of relevant stimulus features that are shared between any pair of tasks linked to different response dimensions (see Fig. 3a). That is, two tasks involving two different response dimensions could either share no relevant stimulus features (cf. Fig. 2a), all five stimulus features (cf. Fig. 2b) or any whole number of features in between, resulting in 6 different task environments. We trained 100 networks in each of the environments. The networks were trained on all nine tasks with the same set of 50 stimulus samples until the network achieved an MSE of 0.01.

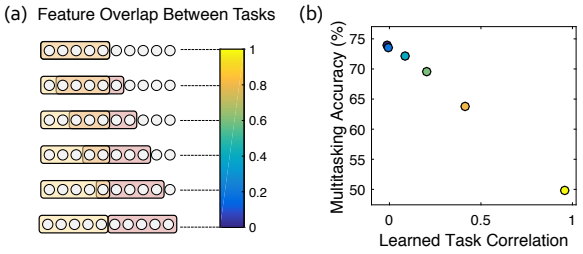


Figure 3: Effects of task similarity. (a) Networks were trained in task environments with varying degrees of feature overlap. Yellow and red shades highlight task-relevant stimulus features for two tasks involving different response dimensions. (b) Final multitasking accuracy of the network as a function of the learned similarity between tasks involving different response dimensions. Colors indicate the degree of feature overlap present in the task environment as illustrated in (a).

In order to assess the similarity of learned task representations we focus our analysis on the weights from the task units to the associative layer, insofar as these reflect the computations carried out by the network required to perform each task. For a given pair of tasks we compute the learned representational similarity between them as the Pearson correlation of their weight vectors to the associative layer.

We measured multitasking performance for pairs of tasks (of different stimulus and response dimensions) by activating two task units at the same time and evaluating the concurrent processing performance in the response dimensions relevant to the two tasks. The accuracy of a single task $A_{Single}$ can be computed as

$$A_{single} = \frac{a_c}{\sum_{i=1}^{5} a_i} \quad (1)$$

where $a_i$ is the activation of the $i$th output unit of the task-relevant response dimension and $a_c$ is the activation of the correct output unit. The multitasking accuracy is simply the mean accuracy of both engaged single tasks.

The simulation results confirm well-known explorations in neural networks (Hinton, 1986; McClelland & Rogers, 2003; Saxe et al., 2013) that task similarities in the environment can translate into similarities between learned task representations. Critically, this extrinsic bias toward the learning of shared representations negatively affected multitasking performance (Fig. 3b). To illustrate this, consider the simultaneous execution of tasks 1 and 5 in an environment as depicted in Fig. 2b. If the network learns similar representations at the associative layer for tasks 1 and 2 (note that both tasks rely on the same stimulus features), then executing task 1 will implicitly engage the representation of task 2 which in turn causes interference via its link to the response dimension of task 5.

[3] All reported results were obtained using gradient decent to minimize the MSE of each training pattern. However, we observed the same qualitative effects using the cross-entropy loss function.

## Multitasking Limitations due to Intrinsic Learning Biases

In addition to environmental biases that shape the learning of shared task representations there may be factors intrinsic to the neural system that can regulate the degree to which such representations are exploited in learning. In this section we introduce a formal analysis of how such biases can affect the tradeoff between learning efficiency and multitasking performance. We then use weight initialization as a learning bias in simulations to establish a causal relationship between the use of shared representations on the one hand, and resulting effects on learning and multitasking, on the other hand.

### Formal Intuitions on the Tradeoff between Learning Efficiency and Multitasking Capability

To gain formal intuition into the tradeoff between multitasking ability and learning speed, we consider a stripped-down version of the introduced network model that is amenable to analysis. In the full model, nonlinear interactions between the task units and the stimulus units occur in the associative layer. Here we assume a *gating model* in which these nonlinear interactions are carried out through gating signals that can zero out parts of the activity in the associative and output layers, or pass it through unchanged. The choice of which parts of each layer are gated through on each input is left to the designer (not learned, as in the full model).

We study the scheme depicted in Fig. 4 consisting of $M$ input and response dimensions with full feature overlap (cf. Fig. 2b). For the output layer, we assume that the gating variables automatically zero all but the task-relevant response dimensions. For the associative layer, we separate the hidden units into dimensions, one for each input dimension, and make the gating variables zero all representations except the one coming from the task-relevant input dimension (Fig. 4a).

Crucially, when the gating structure is known on a specific example, the output of the network is a linear function of the neurons that are on. Given this setting, the learning dynamics can be solved exactly using methods developed by Saxe, McClelland, and Ganguli (2014). The key advantage afforded by the gating scheme is depicted in Fig. 4a: the input-to-hidden weights for one input dimension can be shared by all tasks that rely on that input dimension. This leads to a factor $\sqrt{M}$ speedup in learning relative to learning a single task by itself (proof omitted due to space constraints).

However, with this gating system, multitasking is not possible: gating another task through to the output will lead to interference. To counteract this, the gating scheme must be changed: response dimensions can be divided into $Q$ groups, each with a dedicated set of hidden units (Fig. 4b). This allows tasks that use response dimensions in different output groups to be performed simultaneously. Hence a maximum of $Q$ tasks can be performed simultaneously, but weight sharing is reduced across tasks by a factor $Q$, slowing learning.

This analysis provides, at least in a simplified system, a quantitative expression of the fundamental tradeoff between

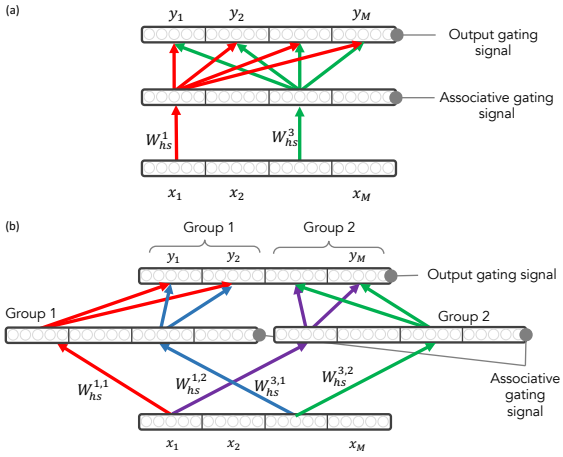


Figure 4: Gating model used for formal analysis. (a) Task information directly switches on or off task-relevant dimensions in the output and associative layers. This allows input-to-hidden weights to be shared across the $M$ different tasks corresponding to different response dimensions, increasing learning speed by a factor $\sqrt{M}$. However, two tasks that rely on different input dimensions cannot be multitasked due to crosstalk at the output (convergent red and green arrows). (b) Multitasking ability can be improved by separating response dimensions into $Q$ groups, each with a dedicated set of units in the associative layer. Gating now permits one task from each group to operate concurrently (red and green arrows no longer converge). However, weight sharing is limited to the group, yielding a learning speed of $\sqrt{M/Q}$.

learning speed and multitasking ability. Let $t$ be the number of iterations required to learn all tasks, $Q$ the maximum number of concurrently executable tasks, and $M$ the number of input/response response dimensions. Then

$$t^2 \propto Q/M \tag{2}$$

where the proportionality constant is related to the statistical strength of the input-output association for one task, the learning rate, and the error cut-off used to decide when learning is complete (Saxe, Musslick, & Cohen, 2017).

Due to the tradeoff in Eqn. (2), gating schemes that share more structure will learn more quickly. Hence generic, randomly initialized nonlinear networks will tend to favor shared representations, as shown in Simulation Experiment 1.

### Simulation Experiment 2: Effects of Learning Biases for Shared Representations

In Simulation 2 we focus on a bias intrinsic to the neural system, i.e. the initialization of the weights from the task layer. We use this factor to systematically examine how the use of shared representations facilitates the discovery of similarity structure while diminishing multitasking performance. To do so, we focus initially on a training environment in which tasks are maximally similar, as this is the condition in which there is most opportunity for exploiting shared representations. We then examine environments with 80% and 0% feature overlap between tasks, to test the generality of the observed effects.

To manipulate the bias towards shared task representations, we initialized the weights from the task units to the associative layer, varying the similarity among the weight vectors across tasks with the rationale that greater similarity should produce a greater bias toward the use of shared representations in the associative layer. Weight vectors for tasks relying on the same stimulus input dimensions were randomly initialized to yield a correlation coefficient of value $r$. The correlation value $r$ was varied from 0 to 0.975 in steps of 0.025 and was used to constrain initial weight similarities for 100 simulated networks per initial condition. The weight vectors for tasks of non-overlapping stimulus dimensions were uncorrelated. Finally, all task weights to the associative layer were scaled by a factor of 5 to enhance the effects of different initial task similarities. The networks were trained using the same parameters as reported for Simulation Experiment 1.

Simulation results indicate that networks with a higher similarity bias tend to develop more similar representations at the associative layer for those tasks (in terms of their final weight vector correlations), whereas a lower similarity bias leads to more distinct task representations at this layer. In environments with high feature overlap between tasks, stronger initial biases toward shared representations lead to increased learning speed (i.e. less iterations required to train the network), as similarities between tasks can be exploited (Fig. 5a). Critically, this comes at the cost of multitasking performance. Learning benefits gained from shared representations are less prevalent in environments with less feature overlap between tasks. However, effects of weight similarity biases on multitasking impairments remain (Fig. 5b).

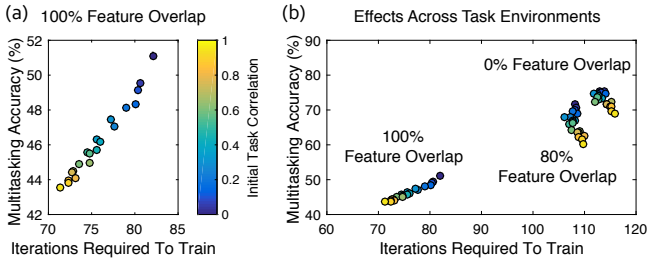


Figure 5: Effects of weight similarity bias. Mean multitasking accuracy (for two tasks simultaneously) plotted against the mean number of iterations required to train the network. Data points represent the mean measures across networks initialized with the same task similarity (constrained by task weight vector correlation) for tasks relying on the same stimulus dimensions. Effects are shown for (a) environments with 100% feature overlap between tasks, as well as (b) across environments with different feature overlap. Different data point clusters correspond to different training environments.

## General Discussion and Conclusion

The limited ability to perform multiple control-dependent tasks at the same time is one of the most salient characteristics of human cognition, and is universally considered a defining feature of cognitive control. Despite these facts, the sources of this capacity constraint associated with control remain largely unexplored. Here, we build upon the observation that multitasking limitations can arise from shared representations between tasks (Feng et al., 2014; Musslick et al., 2016), and use a combination of formal analysis and neural network simulations to examine biases towards shared representations that incur such costs in multitasking.

In the first part of this study, we build upon early insights of connectionism that shared representations emerge as a function of task similarities in the environment and demonstrate the deleterious consequences for multitasking performance. It has been shown that networks are capable of extracting similarities from a hierarchically structured input space (Hinton, 1986). Recent analytic and empirical work in the domain of semantic cognition paints a similar picture: neural systems may gradually discover shared structure in the task environment with a bias towards the initial formation of shared, low-dimensional representations (Saxe et al., 2013; McClelland & Rogers, 2003). Our simulation results are in line with these observations showing that shared task representations emerge as a function of high stimulus feature overlap between tasks and furthered the insight that such similarities in the task environment lead to multitasking limitations.

In the second part, we examined how intrinsic learning biases towards shared or separate representations (by means of weight initialization) can be used to expose a tradeoff between learning efficiency and multitasking performance. Early work in machine learning suggests that learning biases towards a particular representation can be understood as biases of the learner's *hypothesis space* (Baxter, 1995), that is, the set of all hypotheses a learner may use to acquire new tasks. We formalized this hypothesis space in terms of the amount of shared representations between tasks and showed how this mediates an inverse relationship between learning efficiency and interference-free multitasking. Our neural network simulations confirmed these analytical predictions, showing that a weight initialization bias towards shared representations enables faster learning if shared structure in the environment can be exploited, but incurs a cost for multitasking. A promising direction for future research may be to explore another prediction: our formalism suggests a role for such biases in regularizing the representational complexity of the network, thereby promoting generalization performance.

Our analyses indicate that neural learning systems, whether natural or artificial, are subject to a tension between "interactive parallelism" on the one hand, which exploits the fine grained structure of representations and similarity in the service of learning, and "independent parallelism" that supports concurrent processing of distinct tasks, on the other hand. A similar tension can be found in the domain of learning and memory. The complementary learning systems hypothesis proposes two separate learning systems, one system that relies on shared representations to support inference, as well as another system that uses separate representations to support independent encoding and retrieval of information

(McClelland, McNaughton, & O'Reilly, 1995). The latter system supports a form of independent parallelism for associational processes that is similar to the form of independent parallelism for executional processes described in this paper.

Altogether our results suggest that the brain may be confronted with balancing multitasking capability against extrinsic and intrinsic biases towards shared representations. A major goal for the development of artificial systems may be to systematically configure the balance between interactive and independent parallelism, as well as to exploit the relative advantages of each. Most efforts in complex neural architectures have focused predominantly on the discovery of shared representations for the purpose of inference and generalization (Bengio, Courville, & Vincent, 2013). However, one of the future challenges will be to explore the tension between learning efficiency and multitasking in networks with higher complexity (i.e. deep networks), as well as in more naturalistic task environments. We hope that this work will help inspire a proliferation of efforts to further explore this area.

## 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.

Baxter, J. (1995). Learning internal representations. In *Proceedings of the eighth annual conference on computational learning theory* (pp. 311–320).

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *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), 332–361.

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.

Hinton, G. E. (1986). Learning distributed representations of concepts. In *Proceedings of the 8th confererence of the Cognitive Science Society* (pp. 1–12). Hillsdale, NJ: Lawrence Erlbaum Associates.

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.

McClelland, J. L., McNaughton, B. L., & O'Reilly, R. C. (1995). Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, *102*(3), 419.

McClelland, J. L., & Rogers, T. T. (2003, April). The parallel distributed processing approach to semantic cognition. *Nature reviews. Neuroscience*, *4*(4), 310–322.

McClelland, J. L., Rumelhart, D. E., & Hinton, G. E. (1986). *The appeal of parallel distributed processing*. Cambridge, MA: MIT Press.

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.

Musslick, S., Dey, B., Özcimder, K., Patwary, M. M. A., Willke, T. L., & Cohen, J. D. (2016). Controlled vs. automatic processing: A graph-theoretic approach to the analysis of serial vs. parallel processing in neural network architectures. In *Proceedings of the 38th annual conference of the Cognitive Science Society* (pp. 1547–1552). Philadelphia, PA.

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).

Rumelhart, D. E., & Hinton, G. E. (1986). Learning representations by back-propagating errors. *Nature*, *323*, 533–536.

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).

Saxe, A. M., McClelland, J. L., & Ganguli, S. (2014). Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In Y. Bengio & Y. LeCun (Eds.), *International conference on learning representations.* Banff, Canada.

Saxe, A. M., Musslick, S., & Cohen, J. D. (2017). *A formal tradeoff between learning speed and multitasking ability in a simple neural network.* `http://www.people.fas.harvard.edu/~asaxe/multitasking.html`. (Retrieved May 13, 2017)

Seung, H., Sompolinsky, H., & Tishby, N. (1992). Statistical mechanics of learning from examples. *Physical Review A*, *45*(8), 6056.

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.