Review

Network Design and the Brain

Saket Navlakha,[1,*] Ziv Bar-Joseph,[2] and Alison L. Barth[3]

Neural circuits have evolved to accommodate similar information processing challenges as those faced by engineered systems. Here, we compare neural versus engineering strategies for constructing networks. During circuit development, synapses are overproduced and then pruned back over time, whereas in engineered networks, connections are initially sparse and are then added over time. We provide a computational perspective on these two different approaches, including discussion of how and why they are used, insights that one can provide the other, and areas for future joint investigation. By thinking algorithmically about the goals, constraints, and optimization principles used by neural circuits, we can develop brain-derived strategies for enhancing network design, while also stimulating experimental hypotheses about circuit development and function.

Introduction: Synergies between Neurobiology and Computer Science

Rapid technological advances in neuroscience are transforming our understanding of brain structure and function. For example, breakthroughs in imaging and optics technologies have enabled the reconstruction of neural circuits, the visualization and tracking of individual synapses over time, and the ability to control and manipulate neural interactions; advances in mRNA analysis techniques have revealed transcriptomes unique to a myriad of neuronal cell types; and improvements in *in vitro* and *in vivo* electrophysiological recording techniques have led to remarkably precise descriptions of neural communication. These advances have also inspired new hardware architectures in engineering (neuromorphic computing [1–4]) and new machine learning algorithms in computer science (deep learning [5–7]). It is an exciting time.

Network design (see Glossary), both in the brain and in engineering, is a fundamental problem with many shared challenges and constraints. The first goal of this perspective is to describe how biological principles derived from neural circuit development can be generalized to improve how network architectures are constructed. The latter is often referred to as the network design problem by computer scientists, where the goal is to build a network, by defining its nodes and edges, with some optimization criteria in mind, such as maximizing efficiency while minimizing costs. This is a problem with a long history and with broad applications, from designing transport networks to **wireless networks** and computer chips [8]. These shared challenges and constraints suggest that there may be many principles studied by neuroscientists that have direct application to network design problems in computer science.

The second goal of this perspective is to examine how principles for network design used in computer science may also help neuroscientists understand the function and development of analogous processes in neural systems. Many frameworks have been developed to quantify the trade-offs inherent in different network structures and design strategies. The formalized measurements of costs, benefits, and constraints used to evaluate network function by engineers could be useful in providing neuroscientists concrete metrics to form and compare hypotheses about circuit organization.

[1]The Salk Institute for Biological Studies, Integrative Biology Laboratory, La Jolla, CA 92037, USA
[2]Carnegie Mellon University, Machine Learning Department, Computational Biology Department, Pittsburgh, PA 15213, USA
[3]Carnegie Mellon University, Center for the Neural Basis of Cognition, Department of Biological Sciences, Pittsburgh, PA 15213, USA

*Correspondence:
navlakha@salk.edu (S. Navlakha).

CrossMark

We first contrast neural circuit development via synaptic pruning, where connections are overproduced and then eliminated over time, versus engineered network design, where connections are usually added over time from an initially sparse topology, since adding connections that will soon be eliminated is deemed wasteful. We discuss the advantages and disadvantages of both approaches, when different design strategies may be useful, and how the study of one can shed light on the other. We then step back and extract from this pruning example five optimization principles often shared by neural and engineered systems. We conclude by describing potential avenues of future bidirectional research on network design using these principles as a guideline.

Synaptic Pruning during Development: A Network Design Perspective
Background
In the early development of neural circuits, nodes and edges (neurons and synapses) proliferate until roughly age 2 years in humans. Then, development shifts into a largely regressive process, where the number of nodes remains more or less unchanged, but where a majority (50–60%) of synapses are pruned by adulthood [9–12]. Although synaptic loss may seem uneconomical from an engineering point of view, there are two reasons why it may occur in the brain. First, this procedure reduces the amount of genetic information required to code for the trillions of connections made in the human brain [13]. Instead of requiring precise instructions to specify every connection, approximate rules can be applied and then be fine-tuned by experience. Second, even if some connections are genetically prespecified [14,15], this would not be desirable for the entire brain because the most appropriate subset of connections also depends on experience [16,17]. Thus, one challenge of development is to find the most appropriate subset of connections according to *a priori* unknown environmental stimuli. The brain finds this subset by overproducing and then eliminating infrequently used connections based on activity-dependent feedback [18,19] (Figure 1A).

Computer scientists have long been fascinated by how networks develop and evolve over time, be it transportation networks such as highways and roads, technological networks such as the Internet, or molecular networks such as protein interactions in the cell. Numerous models have been proposed to capture the dynamics of these developmental processes, including how new nodes are integrated into an existing network, how information propagates through the network, and how different topological structures, such as communities, form [20–25]. While some retraction events do occur — a road may close, a gene may be lost — these models treat these as relatively minor events and largely assume that the number of nodes and edges increases as the network develops over time. Other models have studied aging of nodes or
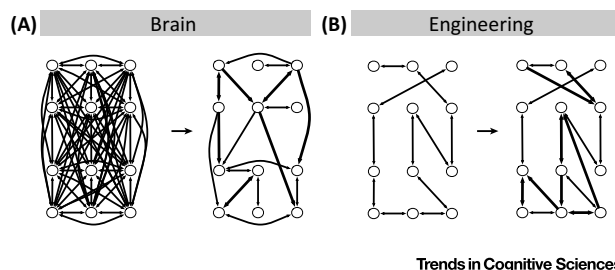

Figure 1. Pruning versus Growth Strategies for Network Design. (A) In the brain, connectivity is dense and then pruned over time. (B) In engineering, a sparse backbone topology (called a spanning tree) is often built to ensure that a communication path exists between any two nodes, and then new edges are added over time. Both strategies produce networks with the same number of nodes and edges at the end, but with different topologies. The thickness of an edge indicates the usage (strength or weight) of the connection.

Glossary
Computational (deep) neural networks: method for performing computational optimization loosely based on how neurons perform computation in the brain. These networks contain several layers (collection of nodes) that are connected by edges that encode a weight. In feed-forward models, the first layer encodes the input and each successive layer uses the values in the previous layer to perform computations, often based on the weighted sum of inputs. Deep neural networks refer to the number of layers in the network, which can range from few to hundreds.
Network design: networks consist of a collection of nodes that are connected to each other via edges. They range from fully connected (every node is connected to every other node) to spanning trees (see below). Network design is focused on determining how many edges to place and where to place them, such that some performance metric is optimized (e.g., minimizing time to relay messages, minimizing the number of redundant paths between nodes, or minimizing costs). Network design is an important aspect of many engineered and computational systems.
Network efficiency and robustness: common measures used to evaluate network performance. Efficiency measures how quickly a signal traverses from an input node to an output node through a path in the network. Robustness measures how tolerant signal propagation is to failures along the path.
Online algorithm: algorithm that processes inputs in a serial fashion, one or few at a time. In other words, the algorithm does not need to wait for the entire input in order to generate an initial output.
Randomized algorithm: algorithm that relies on random decisions as part of its logic. Randomized algorithms are effective for breaking symmetry and for overcoming adversaries since it is impossible to anticipate the exact behavior of a randomized algorithm. There are several cases in which randomized algorithms can be used to solve problems that deterministic

edges in a network [26–30], though the mechanisms driving retraction in these models is not intended to be neural inspired.

When designing networks, engineers often start by building a bare-bone topology and then adding connections over time based on need, demand, or budget (Figure 1B). So, why are neural pruning strategies rarely used in engineered network design? Two reasons come to mind. First, deliberately adding connections that will soon be removed is considered wasteful, particularly when the cost of making connections is high (e.g., building roads). Second, engineered network design is often done in a top-down centralized manner. For example, global traffic patterns of a city can be collected and analyzed together to determine which new road to add to best mitigate future congestion. With a central planner, it is easy to compute the potential value of a road without having to actually construct it. In the brain, building and maintaining synapses does impose energetic costs [31,32]; however, the brain is a distributed system, where no central planner observes all the information flowing through the network. In such a case, a different strategy is needed, one where the value of a connection is determined by building it and then seeing how frequently it is used.

Using Synaptic Pruning for Network Design

To formalize how pruning strategies may be useful for network design, Navlakha *et al*. [33] recently combined theory and experiments to compare pruning- versus growth-based strategies for a distributed network design problem. The main challenge of this problem is that the most appropriate subset of connections to keep depends on the activity observed in the network, which is not known in advance. Moreover, activity flows through the network in real time, and thus cannot be stored and analyzed in bulk.

To find this subset of connections, Navlakha *et al*. derived a neural algorithm based on hyper-connectivity followed by pruning using a simple activity-dependent '**use it or lose it**' rule, implemented by each synapse independently. In this rule, edges (synapses) that observe heavy traffic are more likely to be kept than edges that experience less traffic (Figure 2A,B). This strategy was compared with a growth-based algorithm [34], where a minimal topology, called a **spanning tree**, was initially built to guarantee that a communication pathway exists between every pair of nodes; local edges were then added to shortcut common routes observed in the network activity. The goal is to learn the appropriate structure or topology of the network that best matches patterns in the input activity. The final network produced by both of these approaches was evaluated based on two common graph-theoretic measures: **efficiency and robustness**.

Simulations of the pruning algorithm raised a new hypothesis about how the overall rate of synapse elimination during development plays a critical role in optimizing network structure (Figure 2C). The pruning algorithm observes some activity, prunes some low-use edges, observes more activity, and prunes more edges. The percentage of edges pruned in each such interval defines the pruning rate. Even though each pruning rate produces networks with the same initial and final density (i.e., the same number of nodes and edges at the beginning and end of the simulation) and each uses the same use-based principle to determine which synapse to eliminate, different pruning rates have a striking impact on the topology of the final networks constructed. The only variable changing across pruning rates is the percentage of synapses that are eliminated in each developmental time point.

Although the molecular and cellular mechanisms driving activity-dependent pruning have been extensively investigated for decades [17,35,36], this critical rate parameter has previously been neglected by experimentalists [9,12]. Using an electron microscopy staining technique that
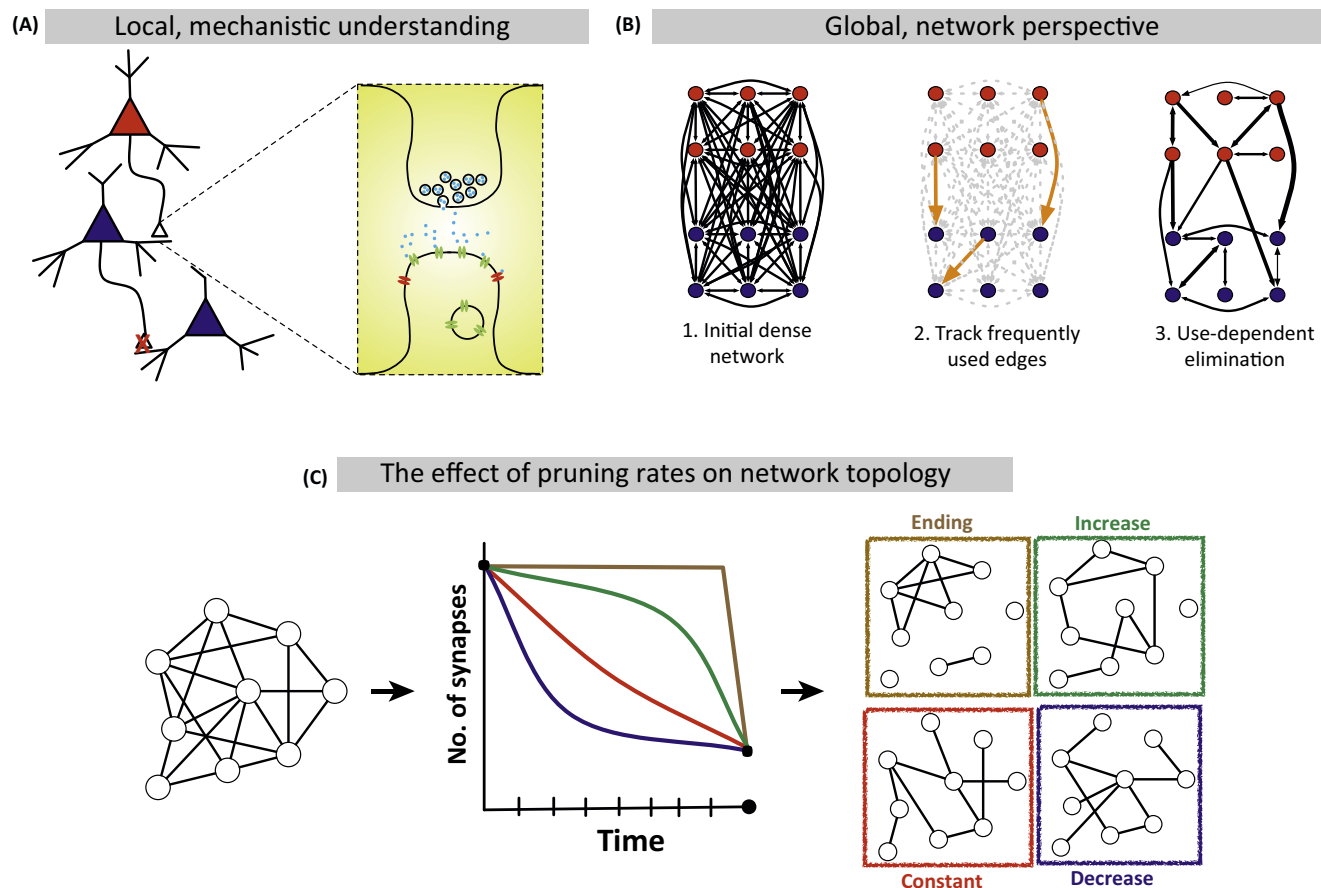
algorithms are not guaranteed to solve.
Signal interference: anything that disrupts a signal as it travels along a channel (edge) between a sender and a receiver.
Spanning tree: for a connected graph G, a spanning tree is a connected subgraph of G containing all the nodes in G and the minimum possible number of edges.
Use it or lose it: inspired by principles of Hebbian learning, an edge in a network is preferentially kept if it is frequently used to transmit information between nodes in the network; otherwise it is removed.
Wireless networks: collection of devices that interact wirelessly. These networks can be prespecified (e.g., a soccer robot team) or created in an *ad hoc* fashion (all cellphones within a given radius). Determining low-cost communication strategies for these networks is important for achieving robust and efficient network performance.

CellPress
REVIEWS

(A) Local, mechanistic understanding

(B) Global, network perspective

1. Initial dense network

2. Track frequently used edges

3. Use-dependent elimination

(C) The effect of pruning rates on network topology

No. of synapses

Time

Ending

Increase

Constant

Decrease

Trends in Cognitive Sciences

Figure 2. Synaptic Pruning for Network Design. (A) Prior work has largely focused on understanding the molecular and cellular mechanisms driving synaptic pruning at an individual synapse. Global aspects of this process, including the rate at which synapses are pruned, the impact of these rates on network topology, and the contrast of pruning-versus growth-based algorithms commonly used in engineering to construct networks, was not previously studied. (B) A new distributed network design algorithm based on hyperconnectivity followed by aggressive pruning. The algorithm identifies the least frequently used edges and prunes them over time. (C) Depiction of how four different pruning rates, which all start and end with the same number of nodes and edges, affect final network topology. In the middle panel, the x axis denotes time, and the y axis denotes the number of synapses in the network. For the decreasing rate, synapses are aggressively pruned early on, whereas for increasing rates, pruning is initially conservative. The network topology at the final time point is shown on the right. Differences in network topology affects the efficiency and robustness of signal propagation through the network.

pronounces electron opacity at synaptic contact zones [37,38], coupled with high-throughput machine learning analysis [39,40], Navlakha *et al.* counted >20 000 synapses across 41 animals and 16 time points in the developing mouse somatosensory (barrel) cortex. They determined that pruning rates decreased over time (i.e., a period of rapid elimination was followed by long period of slower elimination). They then translated these insights back to the network design domain and found, both theoretically and via simulation, that decreasing rates of pruning lead to 20–30% more efficient and robust networks compared to pruning using other rates, such as constant or increasing rates.

Why did decreasing rates perform better than other pruning rates? Intuitively, when starting from a random *tabula rasa* network, there are many connections that will not be important, and these are relatively easy to identify quickly. For example, when designing an airline network, it will not take long to learn that Los Angeles to New York City is an important and frequently used

route, whereas Des Moines to Fort Lauderdale is less common. Decreasing rates thus have two advantages. First, they remove many of these uncommon connections early and then provide a longer period of time for the remaining connections and pathways to be fine tuned. Second, they incur the least energetic cost amongst the pruning strategies, since many synapses are eliminated early on. Increasing rates have the advantage of being able to observe more data prior to making elimination decisions; however, such rates drastically alter network topology at later time-points, which could fragment the network.

Pruning-based algorithms have also been found to markedly improve both efficiency and robustness compared to growth-based methods for distributed networks. Adding edges to a sparsely connected network without a centralized controller is limiting because most edges are local, connecting nodes in the same vicinity. This hampers the ability to discover global connections that can dramatically improve network efficiency. Further, sparse networks (spanning trees) have many nodes whose removal can disconnect the network if not further connected; this reduces robustness. Prior work has shown that in Hopfield networks, pruning maximized resource utilization for memory storage [16]. Thus, using resources to temporarily build extra synapses leads to rapid convergence to the most appropriate subset of connections in the network, offering a new strategy for distributed network design.

Pruning as a Biological Principle For Network Construction

Biologically, learning and novel sensory experience in adulthood also results in the generation of new spines; a majority of which are selectively eliminated following a decreasing rate of pruning [41,42]. This suggests that learning-induced spine formation throughout the lifetime of an organism follows a similar rule of structural plasticity. Competition for inputs can also sculpt neural circuits, where 'losing' (i.e., redundant) inputs are pruned, and 'winning' inputs are strengthened and potentially gain additional synapses. Examples of this include the cerebellum [43] and neuromuscular junction [36]. Finally, arctic ground squirrels prune synapses during hibernation; upon arousal, synapses are overproduced and then pruned back to prehibernation levels after roughly 2 h [44,45]. This may represent a cost-savings mechanism during hibernation and a means to quickly adapt to a new environment after arousal.

Given the broad importance of adaptively determining appropriate connections in a network, it is perhaps not surprising that pruning-based strategies are used throughout the animal world. In the development of vasculature networks in zebrafish, vessel pruning follows a decreasing rate, where the majority of pruning events occurs shortly after fertilization [46]. Pruning is also observed in slime mold foraging networks [47], in ant pheromone trail networks, and in plant root foraging [48]. In all of these cases, pruning implements an explore–exploit strategy, where generating excess allows for quick discovery of the important. While the pruning mechanisms is clearly different across these examples, pruning appears to be a widely used principle of network construction in biology.

Practical Applications and Follow-up Work

Where might this pruning strategy be of practical use in engineering? As mentioned above, applications that can rely on central planning, such as building road networks, are usually best optimized in a *post hoc* manner. However, emerging technologies based on mobile, wireless, and sensor devices require a distributed optimization of network architectures. These applications have raised new computational challenges including the need for energy-efficient and flexible topologies [49,50]. For example, wireless networks that monitor physiological states in the body [51], that track volcanic activity in a hazardous environment, or that scan for enemy movement in a battlefield [52] are not easily serviceable and require that communication

pathways between nodes adapt to *a priori* unknown activity patterns. The main difference in these applications is that network connectivity can be modulated digitally (wirelessly), as opposed to by altering physical wiring.

Pruning strategies may also be applicable for designing efficient networks-on-chip circuits [53] and for training deep machine-learning networks [54,55]. Training a deep network is computationally and memory intensive, which can preclude their use in applications where such resources are not available. Standard learning approaches do modify connections weights but largely assume that architectures are fixed over time (e.g., all-to-all topologies). Pruning low-weight connections during training in a decreasing rate schedule may enable more effective network performance by reducing the complexity of a model without sacrificing its accuracy.

The insights derived here have also led to several follow-up studies. For example, while there are many models that describe how synaptic weights change as a function of activity (e.g., spike-timing dependent plasticity), less emphasis has been placed on understanding structural plasticity, that is, how network topology changes as a function of activity. Recent work has shown that using pruning strategies in a biological neural network can decrease the time to learn appropriate weights for inferring input–output mappings [56] and can be more robust in the presence of noise [57].

These findings also raise questions about whether pruning rates are perturbed in neuro-developmental disorders, such as Fragile X syndrome (too many synapses) or Rett syndrome (too few synapses), compared to neurotypical individuals that lie in a well-balanced trade-off in between. Interestingly, excessive connectivity can lead to **signal interference** – a major challenge in wireless networking [58] – where communication pathways meant to be distinct overlap with each other, reducing the speed and reliability of transmission.

Optimization Principles Shared by Neural and Engineered Networks

The above example highlights how an algorithmic perspective on neural development could inspire new algorithms for computer scientists and could raise new biological hypotheses about circuit construction. Next, we take a step back and extract from this example five optimization principles shared by many neural and engineered systems. These principles have been explored in other biological systems at the molecular level (e.g., gene regulation and genetic circuit design [59,60]); at the cellular level, excluding the brain [61]; and at the population level, such as the collective behavior of organisms [62,63]. However, they are not typically applied by neuroscientists when appraising network performance. Prior work has pointed out some additional similarities [50] – including the reliance on network interactions (Figure 3A) and distributed forms of computing (Figure 3B) – which we build upon here.

Efficiency

Efficiency provides a measure of the time versus resource cost of an action. For example, in engineered networks, efficiency could measure how long it takes to process an input signal, transform it, and then trigger an appropriate output response. A variety of network structures have been proposed to facilitate efficient information transfer (e.g., motifs and hubs), although some of these structures are more costly to build and maintain than others.

In the pruning example, we measured efficiency in terms of the time to propagate a signal from an input node to an output node through the network. Extra synapses, though costly to maintain, were used to quickly converge to optimal routing paths. In the brain more generally, resources are limited in numerous ways [64]: (i) volume of physical space, which is finite and
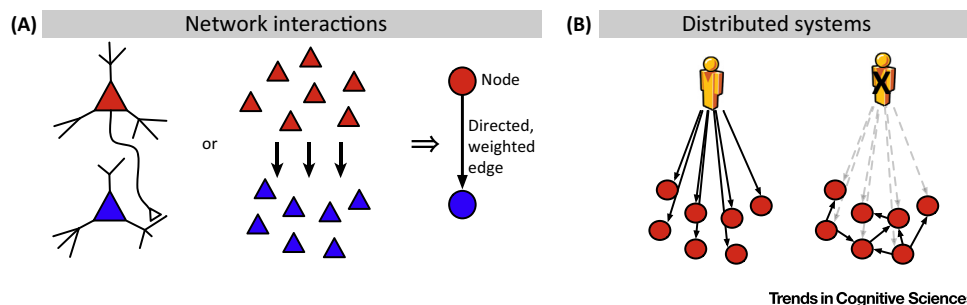
Figure 3. Two Common Features of Neural and Engineered Systems. (A) Networks are a common abstraction for representing neural communication. Nodes can be defined at multiple levels, including as individual neurons or as a population of neurons. Edges denote synapses or communication pathways and are often directed and weighted. Networks are embedded in physical space, which can impose wiring and volume constraints that affect connectivity. (B) Centralized systems contain a node that can monitor and control the entire network from one location. In distributed systems, each node has only a limited view of network activity, and coordination must arise emergently from the behavior of all nodes. In the central nervous system, there is likely a continuum between fully distributed and centralized architectures. For example, local hubs may collect and coordinate the activity of many small subcircuits. As observed in the pruning example, solving the same computational problem using a centralized versus a distributed algorithm requires markedly different strategies.

affects cell packing [65]; (ii) metabolic costs involved in supporting synaptic transmission, which can affect the number of available synapses [32]; and (iii) energetic costs associated with generating and propagating action potentials [66]. Two exemplars of neural efficiency that affect network design are Cajal's law of conduction delay and the wiring length minimization principle, which constrain neural arborization in many brain regions and species [67–72]; other examples include packet switching [73] and short input–output processing paths [74]. Understanding how to allocate limited resources to achieve the best performance per unit cost [75] is an essential component of network design.

Robustness
Robustness measures network resilience in the presence of external perturbations or internal noise. One common approach to enhancing robustness is to build redundancies, such that some components can fail without sacrificing function. Another approach is to transition from deterministic algorithms to probabilistic or **randomized algorithms** [76], which are only guaranteed to perform well on a task with some probability; the algorithmic challenge, then, is to ensure that this probability is very high. This approach is also becoming popular today as computing devices are getting smaller [77], since miniaturization often sacrifices reliability.

The brain has a remarkable ability to remain functional across a wide range of external conditions and internal noise. For example, >80% of all dopaminergic neurons must be lost from the substantia nigra before the onset of motor symptoms in patients with Parkinson's disease [78,79]. In the pruning example, we measured robustness to such node loss. Internal components themselves can also be highly unreliable; for example, many synapses have failure rates >60% [80,81]. Robustness can also be viewed in terms of maintaining an invariant. For example, neurons in primary visual cortex show similar line orientation tuning despite different stimulus strengths or visual contrast [82]; similarly, divisive normalization in olfactory circuits helps tune neural responses to be invariant to odor concentration [83].

While engineers do often think about building networks that are robust to external perturbations, it is less common to build reliable systems using internal components that are individually

very noisy. Constructing network architectures that remain responsive despite both external and internal perturbations is thus a primary challenge in network design.

Adaptation

Adaptation provides a measure of how well network performance (i.e., efficiency and robustness) changes in response to changing inputs or environments. This often requires modifying network topologies or connection strengths to better reflect new input or environmental statistics.

Neural systems are lionized for their ability to modify connection topologies and connection strengths based on experience [42,84–86]. In the pruning example, starting from a highly connected network, input activity was used to determine the most appropriate connections to keep; that is, the topology of the network was sculpted to reflect the structure of the input. More generally, while the rules regulating synaptic plasticity and synaptic renormalization [87] have been the subject of intense interest, less understood is how changes in local connectivity, brought about by changes in activity, are integrated into and influence global network structure. Some examples include the study of how heterogeneity in input probability can help balance excitation and inhibition [88] and how autaptic connections (i.e., self-edges) affect bursting behavior in network [89]. Computationally, adaptation remains a major challenge, especially in non-stationary or life-long machine learning applications [90,91], where systems are not tuned to one fixed task, but are rather autonomous.

Online Operation

Online operation is a requirement where input–output responses are calculated in real time, usually with little-to-no ability to pause for future data or to refer to previous data [92–94]. This contrasts with offline operation, which assumes that data can be aggregated together and complex operations can be performed on the entire dataset at once.

In the pruning example, inputs were provided one at a time, which necessitated a simple use it or lose it-based algorithm. In computer science, online algorithms are critical in many applications, for example, on the Internet when routing large volumes of live traffic, or in sensor networks, where each mobile device may be limited in their ability to store and analyze large amounts of data. Online algorithms only process a tiny handful of data at a time, which restricts the complexity of computations that can be performed but provides large benefits in terms of memory and speed. In the brain, there is likely a continuum in this regard, where a small portion of previous data (e.g., memories [95]) may be retrieved or fed back to determine the appropriate response to a new input. The main challenge of online algorithms is to approximate the performance of offline algorithms, despite having less time or fewer computing resources.

Scalability

Scalability is a property stating that an algorithm can solve a problem in a small or large network with similar computational logic.

Scalability is a common evolutionary byproduct, since it is often easier to build off an existing circuit to solve a more complex problem, as opposed to redesigning an entirely new circuit from scratch [96,97]. Examples include evolutionary scaling of neural components [98] and Rentian scaling, demonstrating similarities in computer chip architectures and neural circuits [99] (see also [100]). Furthermore, many circuit motifs are conserved across species and brain areas, indicating that similar computational logic may be used to solve information processing problems in small and large brains. In engineering, one common technique used to enhance scalability is to use distributed algorithms; the lack of reliance on a centralized controller means

that individual nodes, implementing the same logic, can be added to a network to scale the computation to a higher dimension.

Importantly, these goals may be at odds with one other. For example, highly efficient systems may not be as robust to perturbations, nor as flexible in response to changing environments. A key network design challenge is to determine how best to compromise between these criteria.

Potential Future Directions

Where else may a network design perspective provide new insights into circuit development and structure? Below, we describe three potential areas of joint investigation by neuroscientists and computer scientists that use the principles described above to motivate biological and computational questions.

Role of Silent Synapses during Development

Early formation of AMPA-silent synapses are vital for normal circuit development [101]. Silent synapses are those for which a presynaptic action potential fails to evoke a detectable postsynaptic signal in the receiving cell, despite being physically present. AMPA-silent synapses lie in a transitory state; they can be activated and integrated into existing pathways if recruited by activity, or irreversibly pruned otherwise [101–103]. The precision of this transition is vital, as premature or delayed transitioning of silent synapses has been implicated in several developmental pathologies [104]. While the molecular mechanisms of silent synapses have been intensely investigated [105], many details of this process remain elusive, including how and when silent synapses transition to different states and what affect these local decisions have on circuit function and topology. State- or condition-specific silencing or activation of synapses may enable information rerouting or signal amplification in neural networks. However, without tools that enable selective control of cell-type specific synapses, it will be difficult for neuroscientists to understand how silent synapses can regulate network output.
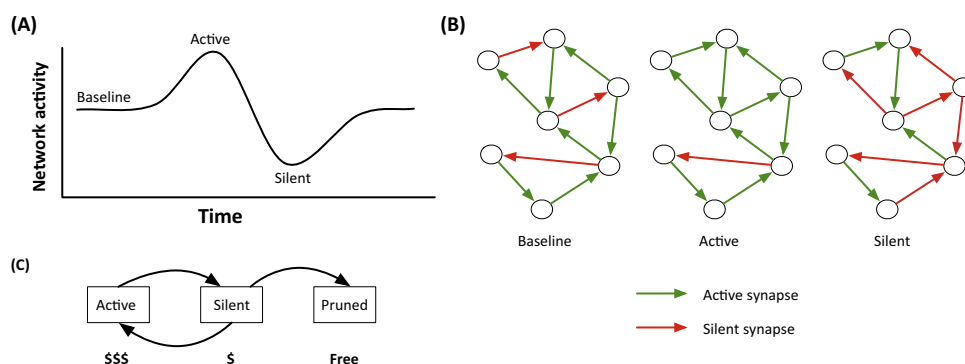
Trends in Cognitive Sciences

Figure 4. Computational Role of Silent Synapses. (A) Schematic showing how varying levels of network activity over time modulates the state of an individual synapse. Sustained, enhanced activity drives the synapse to become active, whereas periods of low activity causes the synapse to be silenced. (B) Network-level effects at three stages of network activity (baseline, active, and silent), showing how a population of synapses may change state depending on network activity, thus altering information flow. Red arrows indicate synapses that are silent; green arrows indicate synapses that are active. (C) Markov transition diagram depicting three synapse states. Each state has an associated cost to maintain: active synapses are most expensive, silent synapses less so, and pruned synapses are free. The arrows indicate possible transitions between states.

One way to investigate this, borrowed from engineering thinking, is to view a synapse abstractly as a three-state Markov chain: active ↔ silent → pruned (Figure 4). Each state has an associated cost to maintain: pruned synapses are 'free', but active synapses are more metabolically costly to maintain than silent synapses due to additional neurotransmitter release, protein synthesis, etc. The Markov model has three transition probabilities, and the goal is to understand how these transition probabilities are modulated by activity on the network and how they affect network adaptation. For example, the network could keep all synapses active at all times, but this might be energetically wasteful during times with low-volume traffic. In the mature brain, synapses may also fluctuate between active and silent states, controlled not by postsynaptic receptor content but by modulation of release probability [106]. These fluctuations may provide an opportunity for rerouting information flow without more costly anatomical rearrangements. The metrics discussed above could then be used to understand the trade-offs that different transitions strategies provide.

The notion of a 'silent connection' – one that is physically present but not used [106] – has been uncommon in engineering, but their use may enable more computationally flexible networks with the same physical substrate. State-dependent, rapidly reconfigurable networks may also be useful in designing field-programmable gate arrays (FGPAs), or in deep learning applications when the distribution of the input data could change over time [90].

Spontaneous Activity for Network Development and Maintenance

Spontaneous neural activity plays a critical role in guiding and refining connections during development [107,108] and in maintaining and renormalizing synaptic weights during quiescent periods [109]. Spontaneous activity can be generated at the earliest stages of sensory processing and propagated across the brain [110,111], or it can be generated by intrinsic activity in the network, as occurs in the neocortex [112–115]. The necessity of spontaneous activity for neural network maintenance and function is not well understood. Is spontaneous activity truly noise, or does it represent offline processing that naturally occurs in large recurrent networks? Indeed, there is a strong relationship between sensory-evoked patterns of activity and spontaneous activity at rest [116,117], suggesting that the patterns of synaptic connectivity provide a constraint to sequences of neural activity generated internally or through external triggers. It is also possible that spontaneous fluctuation of component activity is required for biological systems built of inherently unstable chemical (proteins, lipids, and nucleic acids) parts; however, systematic manipulation of spontaneous activity in engineered networks might reveal otherwise unexpected functions.

For example, in engineered systems, spontaneous activity is rarely used to construct networks but is commonly used as a maintenance and monitoring tool. In many applications, short messages (called keep-alive messages) are sent from one node (mobile device, or server) to another. These messages are used to check connection status and preserve connections from decay. One challenge is to adjust the frequency of these messages so they do not overburden the network nor interfere with regular activity (sensory-evoked stimuli). Spontaneous activity may also be helpful in neural networks to initialize parameters, which are later fine tuned using labeled data [118], or to encode priors that help the network anticipate likely future stimuli [119].

Computational Primitives across Multiple Scales

Networks are often designed modularly, consisting of collections of nodes (neurons) that interact in stereotyped ways, referred to as computational primitives [120,121] (Figure 5). We can consider a computational primitive at multiple levels; as pairs of contacts (A is connected to inhibitory neuron type B); as a small group of neurons (A is connected to inhibitory neuron type B, B is connected to
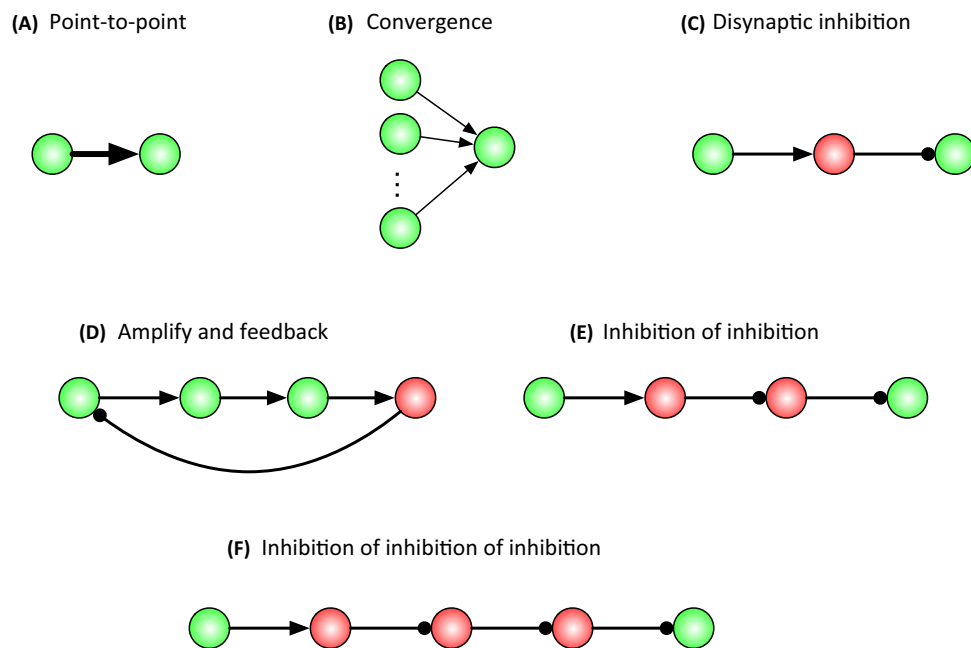
Figure 5. Examples of Computational Primitives in the Nervous System. Green nodes depict excitatory neurons. Red nodes depict inhibitory neurons. The thickness of the arrow (connection) depicts the strength of connection. (A) Point-to-point connections, where one neuron is strong enough to directly drive another. (B) Convergent connections, where a downstream neuron responds only to coincident activation of many input neurons. (C) Disynaptic inhibition, where activation of one neuron is sufficient to activate a neuron that inhibits a downstream neuron. (D) Signal amplification and feedback, where a series of excitatory neurons are amplified and then feedback to inhibit the first neuron. (E) Inhibition of inhibition, where an excitatory input neuron drives a sequence of inhibitory nodes to influence the activity of a downstream neuron. High basal activity of intermediate inhibitory nodes is required for the functionality of this motif. (F) Expanded inhibitory series, with multiple layers of inhibition, are also ubiquitous in neural circuits.

C, and A is never connected to C); or as larger assemblies composed of dozens to hundreds of neurons [122,123], such as the crustacean stomatogastric ganglion [124] as well as small interacting groups of inhibitory neurons in the neocortex, for example, the vasoactive intestinal polypeptide (VIP) to somatostatin neuron inhibitory motif observed across neocortical areas [125–127]. At a broader scale, the mammalian cortical column has been considered a computational primitive whose structural conservation and evolutionary expansion across multiple specialized regions of the neocortex indicates its utility [128].

Similarities between computational primitives observed at both small and multicellular scales in the brain can help us understand the function of these network structures. Take for example directional, serial inhibition (inhibition of inhibition [129], or inhibition of inhibition of inhibition) between GABAergic neurons in the neocortex or in the basal ganglia. Networks using serial inhibition require activity in some downstream inhibitory nodes, since inhibiting something that is not itself active will have no effect. However, persistent inhibitory firing is energetically costly. What are the computational advantages and disadvantages of serial inhibition compared, for example, to direct excitation? One potential answer is that serial inhibition provides a way to expand the dynamic range of output, where it can be both increased and decreased according to task demands and network state [106].

There are myriad examples of other computational primitives that are highly conserved in the brain (Figure 5) and that have a parallel in engineered systems, where they are built with

particular information processing features in mind. These shared primitives include feedforward activation and parallel motifs [130] for amplification of weak signals, gain control, or band-pass filtering, and feedback connections for modulatory control. Moreover, the logic of these motifs are strikingly similar to the complex operations that are carried by computers using Boolean operatives – for example, NAND, OR, and NOT gates – which serve as building blocks for all complex operations in electrical circuits. An understanding of the optimizations performed by these motifs may on the one hand help neuroscientists structure specific hypotheses about circuit organization and function across brain areas. On the other hand, using computational primitives as building blocks may provide new constraints for designing network topologies with specific objectives in mind, such as pruning in deep learning networks to extract features or prioritize information flow, or improving robustness in routing networks; the function of these motifs maybe also be better understood by their application within machine learning applications.

Concluding Remarks

In this perspective, we argued that principles from neural circuit construction can be used to inform the design of engineered networks; at the same time, engineering perspectives can also raise new testable hypotheses about neural circuit function. We grounded this claim in a recent example, where synaptic pruning was used to devise a novel distributed network design algorithm with marked advantages over conventional design strategies. This approach also led to a new hypothesis about how different pruning rates affect circuit topology. We then described five principles (efficiency, robustness, adaptation, online operation, and scalability) shared between the two domains, and highlighted potential future problems where a joint perspective may benefit both computer scientists and experimentalists (see Outstanding Questions). While we focused here on circuit changes during development, structural plasticity occurs throughout the lifetime of an organism; we hope some of the ideas presented here may also be useful towards their study [131].

Critical to this joint perspective is identifying the right level of abstraction in which to view circuit function. For example, in applying pruning rules to network construction, we generalized information flow so that individual nodes were sufficient to pass signals, in contrast to the brain, where postsynaptic firing typically requires convergent activity from many presynaptic inputs. Although abstraction always carries with it some loss, it can help identify and evaluate the role of general operating principles that might otherwise be hard to identify and properly weighted amongst a broad array of biophysical and anatomical measurements. Indeed, it is important to emphasize that our goals are not to replicate brain function by faithful recapitulation of neurophysiological properties [132]. Rather, we hope that analysis of brain-inspired circuit structures and design strategies can provide inspiration for novel algorithms not typically used in engineering. Conversely, evaluating network design strategies based on their efficiency, robustness, and costs – a perspective familiar to engineers and network scientists – may reveal new principles guiding circuit organization, including their information–transformation properties and their cost–benefit trade-offs.

To date, the relationship between network design scientists and neurobiologists has mostly consisted of two one-way streets. We hope this perspective will help enable a rich and reciprocal exchange of ideas between these two domains, building numerous and mutually beneficial interactions.

Outstanding Questions

In addition to pruning, how do other forms of structural plasticity that occur throughout the lifetime of an organism affect circuit structure? Can pruning strategies be used to improve the performance of machine learning networks?

Are pruning rates perturbed in neuro-developmental disorders, such as autism spectrum disorders (too many synapses) or Rett syndrome (too few synapses), or in adults that have learning deficiencies? What is the formal relationship between over-connectivity and signal interference (i.e., how much is too much)?

What role do silent synapses play during development, and how do their transitions between states (active, silent, or pruned) facilitate network adaptation while keeping costs down? What technologies need to be developed that enable selective control of cell-type specific synapses? Can these strategies be used to improve life-long learning for machine learning algorithms?

What role does spontaneous activity play for network maintenance and monitoring? How is network structure perturbed in the absence of spontaneous activity, or when its frequency changes?

What are the advantages and disadvantages of different computational primitives, or network motifs, when designing a network? Why are there so many seemingly wasteful motifs (inhibition of inhibition, inhibition of inhibition of inhibition) in neural circuits? Can we build a library of circuit motifs that can be used to classify circuit function and computational strategies across different brain regions and species? Can we use engineered circuits to evaluate the costs and benefits of each motif?

References

1. Dharmendra, S. *et al.* (2011) Cognitive computing. *Commun. ACM* 54, 62–71
2. Esser, S.K. *et al.* (2013) Cognitive computing systems: algorithms and applications for networks of neurosynaptic cores. *Proc. Int. Jt. Conf. Neural Netw.* 13, 1–10
3. Indiveri, G. *et al.* (2011) Neuromorphic silicon neuron circuits. *Front. Neurosci.* 5, 73
4. Chicca, E. *et al.* (2014) Neuromorphic electronic circuits for building autonomous cognitive systems. *Proc. IEEE* 102, 1367–1388
5. LeCun, Y. *et al.* (2015) Deep learning. *Nature* 521, 436–444
6. Goodfellow, I. *et al.* (2016) *Deep Learning,* MIT Press
7. Marblestone, A.H. *et al.* (2016) Toward an integration of deep learning and neuroscience. *Front. Comput. Neurosci.* 10, 94
8. Newman, M. (2010) *Networks: An Introduction,* Oxford University Press
9. Huttenlocher, P.R. (1979) Synaptic density in human frontal cortex – developmental changes and effects of aging. *Brain Res.* 163, 195–205
10. Markus, E.J. and Petit, T.L. (1987) Neocortical synaptogenesis, aging, and behavior: lifespan development in the motor-sensory system of the rat. *Exp. Neurol.* 96, 262–278
11. Bourgeois, J.P. and Rakic, P. (1993) Changes of synaptic density in the primary visual cortex of the macaque monkey from fetal to adult stage. *J. Neurosci.* 13, 2801–2820
12. White, E.L. *et al.* (1997) A survey of morphogenesis during the early postnatal period in PMBSF barrels of mouse SmI cortex with emphasis on barrel D4. *Somatosens. Mot. Res.* 14, 34–55
13. Cowan, W.M. *et al.* (1984) Regressive events in neurogenesis. *Science* 225, 1258–1265
14. White, J.G. *et al.* (1976) The structure of the ventral nerve cord of *Caenorhabditis elegans*. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* 275, 327–348
15. Goodman, C.S. *et al.* (1984) Cell recognition during neuronal development. *Science* 225, 1271–1279
16. Chechik, G. *et al.* (1998) Synaptic pruning in development: a computational account. *Neural Comput.* 10, 1759–1777
17. Stoneham, E.T. *et al.* (2010) Rules of engagement: factors that regulate activity-dependent synaptic plasticity during neural network development. *Biol. Bull.* 219, 81–99
18. Hebb, D.O. (1949) *The Organization of Behavior: A Neuropsychological Theory,* Wiley
19. Turney, S.G. and Lichtman, J.W. (2012) Reversing the outcome of synapse elimination at developing neuromuscular junctions *in vivo*: evidence for synaptic competition and its mechanism. *PLoS Biol.* 10, e1001352
20. Chakrabarti, D. and Faloutsos, C. (2006) Graph mining: laws, generators, and algorithms. *ACM Comput. Surv.* 38, 1–63
21. Leskovec, J. *et al.* (2005) Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proceedings of the 11th International ACM Conference on Knowledge Discovery and Data Mining*, pp. 177–187, ACM
22. Barabasi, A.L. and Albert, R. (1999) Emergence of scaling in random networks. *Science* 286, 509–512
23. Watts, D.J. and Strogatz, S.H. (1998) Collective dynamics of 'small-world' networks. *Nature* 393, 440–442
24. Vazquez, A. *et al.* (2003) Modeling of protein interaction networks. *ComPlexUs* 1, 38–44
25. Sporns, O. (2010) *Networks of the Brain* (1st edn), MIT Press

26. Chan, K.P. *et al.* (2004) Effects of aging and links removal on epidemic dynamics in scale-free networks. *Int. J. Mod. Phys. B* 18, 2534–2539
27. Saavedra, S. *et al.* (2008) Asymmetric disassembly and robustness in declining networks. *Proc. Natl. Acad. Sci. U. S. A.* 105, 16466–16471
28. Witten, T.M. (2015) Introduction to the theory of aging networks. *Interdiscip. Top. Gerontol.* 40, 1–17
29. Dorogovtsev, S.N. and Mendes, J.F. (2000) Evolution of networks with aging of sites. *Phys. Rev. E Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.* 62, 1842–1845
30. Safdari, H. *et al.* (2016) Fractional dynamics of network growth constrained by aging node interactions. *PLoS One* 11, e0154983
31. Laughlin, S.B. (2001) Energy as a constraint on the coding and processing of sensory information. *Curr. Opin. Neurobiol.* 11, 475–480
32. Laughlin, S.B. and Sejnowski, T.J. (2003) Communication in neuronal networks. *Science* 301, 1870–1874
33. Navlakha, S. *et al.* (2015) Decreasing-rate pruning optimizes the construction of efficient and robust distributed networks. *PLoS Comput. Biol.* 11, e1004347
34. Lynch, N.A. (1996) *Distributed Algorithms,* Morgan Kaufmann Publishers Inc.
35. Paolicelli, R.C. *et al.* (2011) Synaptic pruning by microglia is necessary for normal brain development. *Science* 333, 1456–1458
36. Lichtman, J.W. and Colman, H. (2000) Synapse elimination and indelible memory. *Neuron* 25, 269–278
37. Bloom, F.E. and Aghajanian, G.K. (1966) Cytochemistry of synapses: selective staining for electron microscopy. *Science* 154, 1575–1577
38. Bloom, F.E. and Aghajanian, G.K. (1968) Fine structural and cytochemical analysis of the staining of synaptic junctions with phosphotungstic acid. *J. Ultrastruct. Res.* 22, 361–375
39. Navlakha, S. *et al.* (2013) A high-throughput framework to detect synapses in electron microscopy images. *Bioinformatics* 29, 9–17
40. Chandrasekaran, S. *et al.* (2015) Unbiased, high-throughput electron microscopy analysis of experience-dependent synaptic changes in the neocortex. *J. Neurosci.* 35, 16450–16462
41. Xu, T. *et al.* (2009) Rapid formation and selective stabilization of synapses for enduring motor memories. *Nature* 462, 915–919
42. Yang, G. *et al.* (2009) Stably maintained dendritic spines are associated with lifelong memories. *Nature* 462, 920–924
43. Hashimoto, K. and Kano, M. (2013) Synapse elimination in the developing cerebellum. *Cell. Mol. Life Sci.* 70, 4667–4680
44. Popov, V.I. and Bocharova, L.S. (1992) Hibernation-induced structural changes in synaptic contacts between mossy fibres and hippocampal pyramidal neurons. *Neuroscience* 48, 53–62
45. von der Ohe, C.G. *et al.* (2006) Ubiquitous and temperature-dependent neural plasticity in hibernators. *J. Neurosci.* 26, 10590–10598
46. Chen, Q. *et al.* (2012) Haemodynamics-driven developmental pruning of brain vasculature in zebrafish. *PLoS Biol.* 10, e1001374
47. Tero, A. *et al.* (2010) Rules for biologically inspired adaptive network design. *Science* 327, 439–442
48. Kozlowski, T.T. (2012) *Shedding of Plants Parts,* Elsevier

49. Anastasi, G. *et al.* (2009) Energy conservation in wireless sensor networks: a survey. *Ad Hoc Netw.* 7, 537–568

50. Navlakha, S. and Bar-Joseph, Z. (2014) Distributed information processing in biological and computational systems. *Commun. ACM* 58, 94–102

51. Latré, B. *et al.* (2011) A survey on wireless body area networks. *Wirel. Netw.* 17, 1–18

52. Carle, J. and Simplot-Ryl, D. (2004) Energy-efficient area monitoring for sensor networks. *Computer* 37, 40–46

53. Cota, E. *et al.* (2017) *Reliability, Availability and Serviceability of Networks-on-Chip,* Springer Science

54. Han, S. *et al.* (2015) Learning both weights and connections for efficient neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems, NIPS'15*, pp. 1135–1143, Cambridge, MA, USA, MIT Press

55. Anwar, S. *et al.* (2017) Structured pruning of deep convolutional neural networks. *J. Emerg. Technol. Comput. Syst.* 13, 32 1–32:18

56. Spiess, R. *et al.* (2016) Structural plasticity denoises responses and improves learning speed. *Front. Comput. Neurosci.* 10, 93

57. Hiratani, H. and Fukai, T. (2016) Hebbian wiring plasticity generates efficient network structures for robust inference with synaptic weight plasticity. *Front. Neural Circuits* 10, 41

58. Jain, K. *et al.* (2003) Impact of interference on multi-hop wireless network performance. In *Proc. of the 9th Annual Intl. Conf. on Mobile Computing and Networking, MobiCom '03*, pp. 66–80, New York, NY, USA, ACM

59. Alon, U. (2006) *An Introduction to Systems Biology: Design Principles of Biological Circuits. Chapman & Hall/CRC Mathematical and Computational Biology,* Taylor & Francis

60. Lim, W.A. *et al.* (2013) Design principles of regulatory networks: searching for the molecular algorithms of the cell. *Mol. Cell* 49, 202–212

61. Navlakha, S. and Bar-Joseph, Z. (2011) Algorithms in nature: the convergence of systems biology and computational thinking. *Mol. Syst. Biol.* 7, 546

62. Feinerman, O. and Korman, A. (2013) *Theoretical Distributed Computing Meets Biology: A Review,* pp. 1–18, Berlin, Heidelberg, Springer

63. Gordon, D.M. (2016) The evolution of the algorithms for collective behavior. *Cell Syst.* 3, 514–520

64. Balasubramanian, V. (2015) Heterogeneity and efficiency in the brain. *Proc. IEEE* 103, 1346–1358

65. Zhang, K. and Sejnowski, T.J. (2000) A universal scaling law between gray matter and white matter of cerebral cortex. *Proc. Natl. Acad. Sci. U. S. A.* 97, 5621–5626

66. Lennie, P. (2003) The cost of cortical computation. *Curr. Biol.* 13, 493–497

67. Cuntz, H. *et al.* (2010) One rule to grow them all: a general theory of neuronal branching and its practical application. *PLoS Comput. Biol.* 6, e1000877

68. Chklovskii, D.B. *et al.* (2002) Wiring optimization in cortical circuits. *Neuron* 34, 341–347

69. Chen, B.L. *et al.* (2006) Wiring optimization can relate neuronal structure and function. *Proc. Natl. Acad. Sci. U. S. A.* 103, 4723–4728

70. Perez-Escudero, A. and de Polavieja, G.G. (2007) Optimally wired subnetwork determines neuroanatomy of *Caenorhabditis elegans*. *Proc. Natl. Acad. Sci. U. S. A.* 104, 17180–17185

71. Rivera-Alba, M. *et al.* (2011) Wiring economy and volume exclusion determine neuronal placement in the *Drosophila* brain. *Curr. Biol.* 21, 2000–2005

72. Rivera-Alba, M. *et al.* (2014) Wiring economy can account for cell body placement across species and brain areas. *Curr. Biol.* 24, R109–R110

73. Graham, D. and Rockmore, D. (2011) The packet switching brain. *J. Cogn. Neurosci.* 23, 267–276

74. Kaiser, M. and Hilgetag, C.C. (2006) Hilgetag. Nonoptimal component placement, but short processing paths, due to long-distance projections in neural systems. *PLoS Comput. Biol.* 2, e95

75. Bullmore, E. and Sporns, O. (2012) The economy of brain network organization. *Nat. Rev. Neurosci.* 13, 336–349

76. Motwani, R. and Raghavan, P. (1995) *Randomized Algorithms,* Cambridge University Press

77. Shi, W. *et al.* (2016) Edge computing: vision and challenges. *IEEE Internet Things J.* 3, 637–646

78. Bernheimer, H. *et al.* (1973) Brain dopamine and the syndromes of Parkinson and Huntington. Clinical, morphological and neurochemical correlations. *J. Neurol. Sci.* 20, 415–455

79. Fahn, S. (2003) Description of Parkinson's disease as a clinical syndrome. *Ann. N. Y. Acad. Sci.* 991, 1–14

80. Stevens, C.F. and Wang, Y. (1994) Changes in reliability of synaptic function as a mechanism for plasticity. *Nature* 371, 704–707

81. Murthy, V.N. *et al.* (1997) Heterogeneous release properties of visualized individual hippocampal synapses. *Neuron* 18, 599–612

82. Carandini, M. and Heeger, D.J. (2011) Normalization as a canonical neural computation. *Nat. Rev. Neurosci.* 13, 51–62

83. Olsen, S.R. *et al.* (2010) Divisive normalization in olfactory population codes. *Neuron* 66, 287–299

84. LeVay, S. *et al.* (1980) The development of ocular dominance columns in normal and visually deprived monkeys. *J. Comp. Neurol.* 191, 1–51

85. Clem, R.L. and Barth, A. (2006) Pathway-specific trafficking of native AMPARs by *in vivo* experience. *Neuron* 49, 663–670

86. Biane, J.S. *et al.* (2016) Thalamocortical projections onto behaviorally relevant neurons exhibit plasticity during adult motor learning. *Neuron* 89, 1173–1179

87. Fink, C.G. *et al.* (2013) A dynamical role for acetylcholine in synaptic renormalization. *PLoS Comput. Biol.* 9, e1002939

88. Landau, I.D. *et al.* (2016) The impact of structural heterogeneity on excitation-inhibition balance in cortical networks. *Neuron* 92, 1106–1121

89. Wiles, L. *et al.* (2017) Autaptic connections shift network excitability and bursting. *Sci. Rep.* 7, 44006

90. Mitchell, T.M. *et al.* (2015) In *Proc. of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25–30*, pp. 2302–2310, Austin, TX, USA

91. Suen, J.Y. and Navlakha, S. (2017) Using inspiration from synaptic plasticity rules to optimize traffic flow in distributed engineered networks. *Neural Comput.* 29, 1204–1228

92. Bottou, L. (1998) Online algorithms and stochastic approximations. In *Online Learning and Neural Networks* (Saad, D., ed.), Cambridge University Press, Cambridge, UK revised, Oct 2012

93. Shalev-Shwartz, S. (2012) Online learning and online convex optimization. *Found. Trends Mach. Learn.* 4, 107–194

94. Pehlevan, C. and Chklovskii, D.B. (2015) *A Normative Theory of Adaptive Dimensionality Reduction in Neural Networks,* ArXiv e-prints

95. Chaudhuri, R. and Fiete, I. (2016) Computational principles of memory. *Nat. Neurosci.* 19, 394–403

96. Lee, S. and Stevens, C.F. (2007) General design principle for scalable neural circuits in a vertebrate retina. *Proc. Natl. Acad. Sci. U. S. A.* 104, 12931–12935

97. Stevens, C.F. (2009) Darwin and Huxley revisited: the origin of allometry. *J. Biol.* 8, 14

98. Stevens, C.F. (2001) An evolutionary scaling law for the primate visual system and its basis in corticalfunction. *Nature* 411, 193–195

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

100. Jonas, E. and Kording, K.P. (2017) Could a neuroscientist understand a microprocessor? *PLoS Comput. Biol.* 13, e1005268

101. Atwood, H.L. and Wojtowicz, J.M. (1999) Silent synapses in neural plasticity: current evidence. *Learn. Mem.* 6, 542–571

102. Isaac, J.T. *et al.* (1995) Evidence for silent synapses: implications for the expression of LTP. *Neuron* 15, 427–434

103. Huang, X. *et al.* (2015) Progressive maturation of silent synapses governs the duration of a critical period. *Proc. Natl. Acad. Sci. U. S. A.* 112, E3131–E3140

104. Hanse, E. *et al.* (2013) AMPA-silent synapses in brain development and pathology. *Nat. Rev. Neurosci.* 14, 839–850

105. Kerchner, G.A. and Nicoll, R.A. (2008) Silent synapses and the emergence of a postsynaptic mechanism for LTP. *Nat. Rev. Neurosci.* 9, 813–825

106. Urban-Ciecko, J., Fanselow, E.E. and Barth, A.L. (2015) eocortical somatostatin neurons reversibly silence excitatory transmission via GABAb receptors. *Curr. Biol.* 25, 722–731

107. Katz, L.C. and Shatz, C.J. (1996) Synaptic activity and the construction of cortical circuits. *Science* 274, 1133–1138

108. McCormick, D.A. (1999) Spontaneous activity: signal or noise? *Science* 285, 541–543

109. Tononi, G. and Cirelli, C. (2014) Sleep and the price of plasticity: from synaptic and cellular homeostasis to memory consolidation and integration. *Neuron* 81, 12–34

110. Ackman, J.B. *et al.* (2012) Retinal waves coordinate patterned activity through-out the developing visual system. *Nature* 490, 219–225

111. Blankenship, A.G. and Feller, M.B. (2010) Mechanisms underlying spontaneous patterned activity in developing neural circuits. *Nat. Rev. Neurosci.* 11, 18–29

112. Fiser, J. *et al.* (2004) Small modulation of ongoing cortical dynamics by sensory input during natural vision. *Nature* 431, 573–578

113. Poulet, J.F. and Petersen, C.C. (2008) Internal brain state regulates membrane potential synchrony in barrel cortex of behaving mice. *Nature* 454, 881–885

114. Destexhe, A. (2011) Intracellular and computational evidence for a dominant role of internal network activity in cortical computations. *Curr. Opin. Neurobiol.* 21, 717–725

115. Garaschuk, O. *et al.* (2000) Large-scale oscillatory calcium waves in the immature cortex. *Nat. Neurosci.* 3, 452–459

116. Ch'ng, Y.H. and Reid, R.C. (2010, 20) Cellular imaging of visual cortex reveals the spatial and functional organization of spontaneous activity. *Front. Integr. Neurosci.* 4

117. Chiu, C. and Weliky, M. (2002) Relationship of correlated spontaneous activity to functional ocular dominance columns in the developing visual cortex. *Neuron* 35, 1123–1134

118. Erhan, D. *et al.* (2010) Why does unsupervised pre-training help deep learning? *J. Mach. Learn. Res.* 11, 625–660

119. Bray, D. (2014) Intrinsic activity in cells and the brain. *Mol. Biol. Cell* 25, 737–738

120. Lu, J. *et al.* (2017) Selective inhibitory control of pyramidal neuron ensembles and cortical subnetworks by chandelier cells. *Nat. Neurosci.* 20, 1377–1383

121. Hangya, B. *et al.* (2014) From circuit motifs to computations: mapping the behavioral repertoire of cortical interneurons. *Curr. Opin. Neurobiol.* 26, 117–124

122. Marr, D. (1969) A theory of cerebellar cortex. *J. Physiol. (Lond.)* 202, 437–470

123. Marr, D. (1970) A theory for cerebral neocortex. *Proc. R. Soc. Lond. B Biol. Sci.* 176, 161–234

124. Bargmann, C.I. and Marder, E. (2013) From the connectome to brain function. *Nat. Methods* 10, 483–490

125. Lee, S. *et al.* (2013) A disinhibitory circuit mediates motor integration in the somatosensory cortex. *Nat. Neurosci.* 16, 1662–1670

126. Fu, Y. *et al.* (2014) A cortical circuit for gain control by behavioral state. *Cell* 156, 1139–1152

127. Pi, H.J. *et al.* (2013) Kepecs. Cortical interneurons that specialize in disinhibitory control. *Nature* 503, 521–524

128. Mountcastle, V.B. (1998) *Perceptual Neuroscience: The Cerebral Cortex,* Harvard University Press

129. Pfeffer, C.K. *et al.* (2013) Inhibition of inhibition in visual cortex: the logic of connections between molecularly distinct interneurons. *Nat. Neurosci.* 16, 1068–1076

130. Milo, R. *et al.* (2002) Network motifs: simple building blocks of complex networks. *Science* 298, 824–827

131. van Ooyen, A. and Butz-Ostendorf, M. (2017) *The Rewiring Brain: A Computational Approach to Structural Plasticity in the Adult Brain,* Elsevier

132. Markram, H. *et al.* (2015) Reconstruction and simulation of neocortical microcircuitry. *Cell* 163, 456–492