A Catalog of Elemental Neural Circuits

Jacob Feldman
Dept. of Psychology, Center for Cognitive Science, Rutgers University

Abstract

It is well-known that abstract neurons of the McCulloch-Pitts type compute input-output functions that are formally equivalent to suitably defined Boolean functions. Boolean functions, in turn, can be classified (up to isomorphism of underlying logical structure) into a limited number of basic types, each of which is non-isomorphic to each other type. These basic types are conveniently expressed by means of a canonic logical formula involving a minimal number of constituent operations. Each such canonic formula can be translated into an equivalent representation as a neural circuit, resulting in a catalog of basic, distinct neural circuits. Each circuit in this catalog computes a distinct Boolean function, and every Boolean function is computed by one such network, making this a kind of "periodic table" of elemental neural circuits. This brief article presents and discusses this catalog of circuits. An intriguing possibility raised by this analysis is that neural circuits in biological networks might, like the circuits in this catalog, tend to take forms that are minimally complex given the functions they compute—a kind of "principle of neural parsimony."

The modern mathematical investigation of the properties of neural networks began with McCulloch and Pitts's model of an "abstract neuron," which sums D binary inputs, each either excitatory or inhibitory, and fires if and only if the total excitation meets or exceeds a predefined threshold θ. McCulloch and Pitts showed that networks of these simple devices were of extraordinary computational power, in fact equivalent to that of a Turing Machine or modern digital computer (McCulloch & Pitts, 1943).

Underlying their proof is the idea that any abstract neuron, or circuit of abstract neurons, computes a Boolean function: that is, a mapping from D binary variables (the inputs on the abstract "dendrites") to one variable (the output on the abstract "axon"). The mathematics of neural networks are thus inextricably tied up with the mathematics of Boolean functions. A similar relation exists between electrical circuit diagrams, of the type that illustrate textbooks on electrical engineering, and logical formulae, as one would find in a textbook of mathematical logic. The two representations are in a very

I am grateful to Randy Gallistel for helpful discussions and to NSF SBR-9875175 for support. Please direct comments to the author at jacob@ruccs.rutgers.edu.

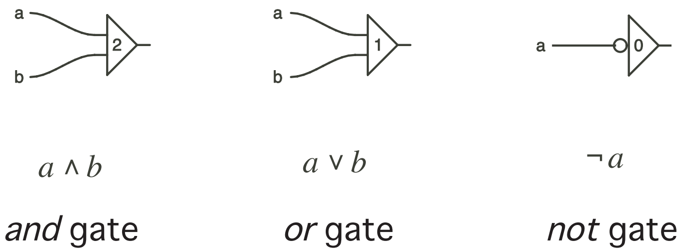

Figure 1. Renditions of the logical operators and (∧), or (∨) and not (¬) as abstract neurons. The number inscribed in each neuron is its threshold θ. Inputs marked with a small circle are inhibitory.

literal sense equivalent, and, as long recognized by electrical engineers, it is convenient to investigate the properties of real electrical circuits by translating them into equivalent symbolic expressions. Similarly, in this paper we catalog neural circuits by reference to the mathematical properties of the functions they compute.

Boolean functions

A simple translation of the conventional logical operators *and* (∧), *or* (∨) and *not* (¬) into equivalent abstract neurons is shown in Fig. 1. Each neuron has a threshold θ inscribed in its "body." The *and* neuron, for example, takes two inputs a and b, both excitatory, and has a threshold of 2. If both inputs are on (meaning that $a \wedge b$ holds) the sum is 2, meeting the threshold, and the neuron fires (releases an action potential). If either input is off, the sum is less than 2, and the neuron does not fire. Similarly, the *or* neuron's threshold is 1, so that it fires if either of its inputs is on. Both the *and* and *or* neurons are easily extended to an arbitrary number D of inputs. For *and*, the threshold is always set to D, so that it fires only if all of them are on. For *or*, the input is always set to 1, so it fires if *any* of them are on. For *not*, the translation is slightly more subtle, requiring an inhibitory input (Fig. 1, right).

Arbitrarily more complex Boolean functions may be created by connecting together larger numbers of the simple neurons shown in the figure. Because *and, or,* and *not* constitute a "complete basis" for Boolean functions, in this manner networks may be constructed corresponding to *any* Boolean function.

Isomorphism of Boolean functions

For a fixed number of inputs D, there are only a finite number 2^{2^D} of distinct input-output functions (there are 2^D possible combinations of inputs, and the circuit fires on some subset of them; there are 2^{2^D} such subsets). However, many of these distinct functions are, in a certain sense, *of the same type* as each other. For example the function $a \wedge \neg b$ and the distinct function $\neg a \wedge b$ are of "essentially the same type," in that they are the same if one simply swaps the labels a and b, which are essentially arbitrary.

When one formalizes and tabulates these equivalences, it turns out that for a fixed number D of inputs and a fixed number P of positive inputs (combinations that lead to an output of one), there are a very small number of basic distinct types. This classification is intrinsic to the space of Boolean functions; it does not depend in any way on the way in which these functions are expressed, whether in a logic-like notation or in some other way—such as a neural circuit—or whether this expression is in any sense minimal. The classification depends *only* on the structure of the functions themselves; that is, on the designated input-output mappings.

The classification of these types is thus very basic to our understanding of the intrinsic variety of possible input-output functions. A typology for up to $D = 4$ inputs was first given by Howard Aiken and his staff in the early days of digital computers (Aiken & the Staff of the Computation Laboratory at Harvard University, 1951); a good modern mathematical treatment can be found in Harrison (1965), and a more complete catalog with additional details can be found in Feldman (2003). One part of this classification that has become prominent in psychology are the six canonic type possible with $D = 3$ and $P = 4$, made famous by Shepard, Hovland, and Jenkins (1961), who studied their ease of learning (with the positive inputs rendered as the features of a class of objects to be memorized). With other values of D and P, the number of canonic forms varies in a somewhat complex manner; for example for $D = 4$ and $P = 4$ there are exactly nineteen types, and for $D = 4, P = 5$, twenty-seven.

As this typology stems purely from mathematical considerations and is not in any way tied to the details of logical notation, it is equally fundamental to the realm of neural circuits. Specifically, it refers to the variety of possible input-output functions that neural circuits may compute. When each canonic input-output function is rendered as a neural circuit, the result is literally a typology of functionally distinct circuits: every neural circuit that computes a Boolean function is isomorphic to exactly one circuit in the typology.

Boolean simplification and minimal formulae

A convenient way to render any Boolean function, and in particular each distinct type in the typology up to isomorphism, is by reference to its *minimal formula*: that is, the shortest propositional formula that expresses the same function. The minimal formula is a very useful window onto the properties of the function. For example the formula $(a \wedge b) \vee (a \wedge \neg b)$ can be reduced to $a \wedge (b \vee \neg b)$ and thus to a, clearly showing that its output really depends on only one of its inputs (a) and is independent of the other (b), which was not obvious from the original unreduced form.

The length of this shortest equivalent formula, called the Boolean complexity, is a measure of the inherent logical complexity of the function; that is, functions that can rendered as very short formulae are inherently "simple," while functions with no such reduction are inherently "complex." Moreover, the Boolean complexity seems to predict the difficulty with which human observers can learn the function, for example commit to memory a set of objects whose features match the positive inputs to the function (Feldman, 2000).

Unfortunately, computation of absolutely minimal formulae is intractable (Garey & Johnson, 1979), but simple heuristic techniques suffice to yield reasonably approximately minimal formula. Each heuristic reduction step can itself be rendered as the transformation of one neural circuit into another, smaller one (Fig. 2). Repeated application of these steps eventually results in a circuit that cannot be further reduced.

Just as the minimal logical formula shows off the structure of each function in a particularly clear way, the corresponding neural circuit clearly illustrates the basic neural computations inherent in each input-output function. In the catalog given below, each

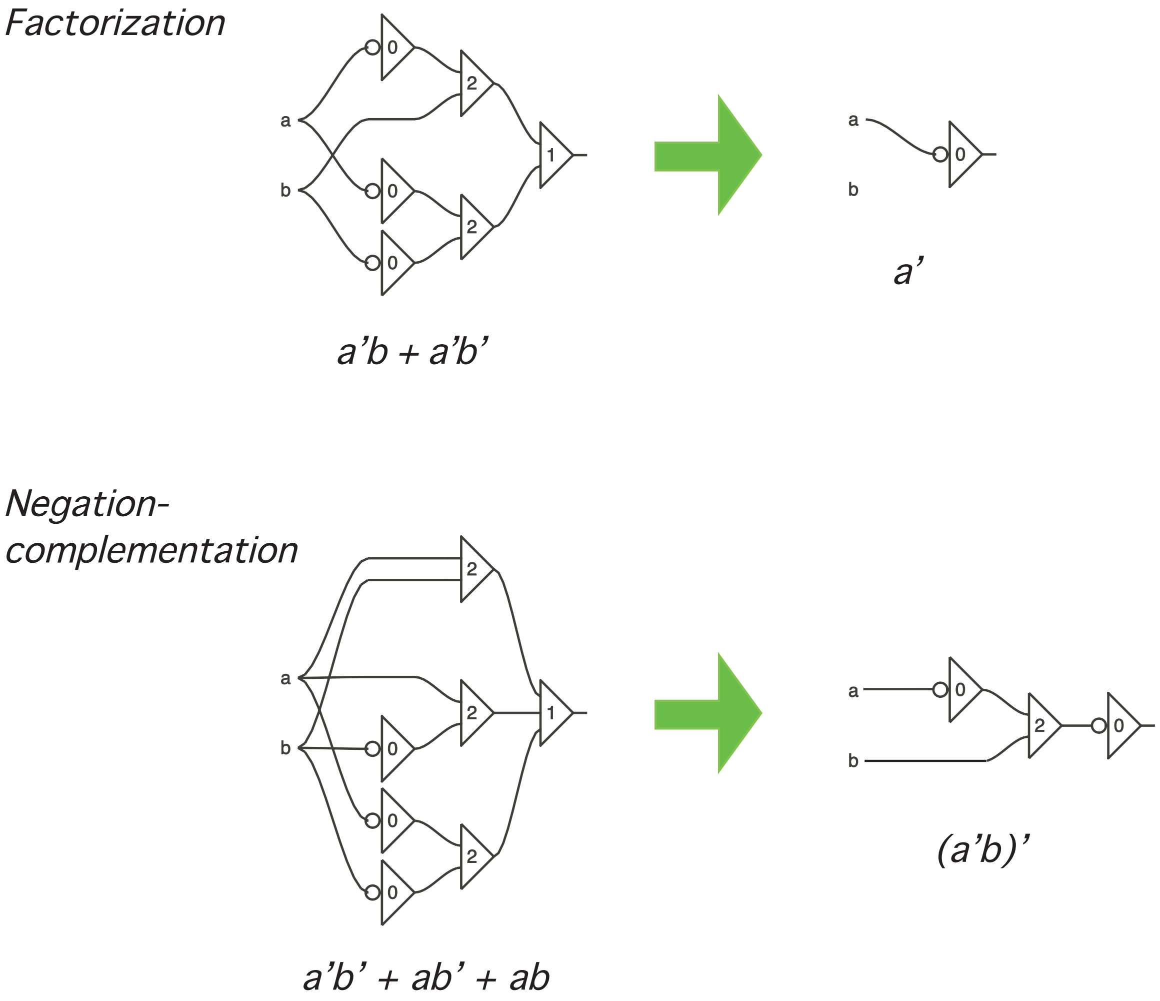

Figure 2. Illustrations of the two basic types of minimization steps involved in finding minimal formulae, factorization and negation-complementation, rendered as transformations of neural circuits. Boolean expressions are shown using the notation $a \wedge b = ab, a \vee b = a + b, \neg a = a'$.

neural circuit is given in this approximately minimal form so that its structure may be best appreciated.

A catalog of neural circuits

Fig. 3 gives the first few rows and columns of the catalog (the complete catalog is infinite, since it extends to arbitrarily large values of D and P). Each row is one value of D (the number of inputs), and each column is one value of P (the number of input combinations that lead to firing). As can be seen, each family comprises a large diversity of circuits, exhibiting large variations in complexity and topology.

Circuits at the upper left of the table, having small values of D and P, are generally relatively simple, with the very simplest the sole member of the $D = 1, P = 1$ family, consisting of a simple *not* neuron. Moving down and to the right, we encounter more

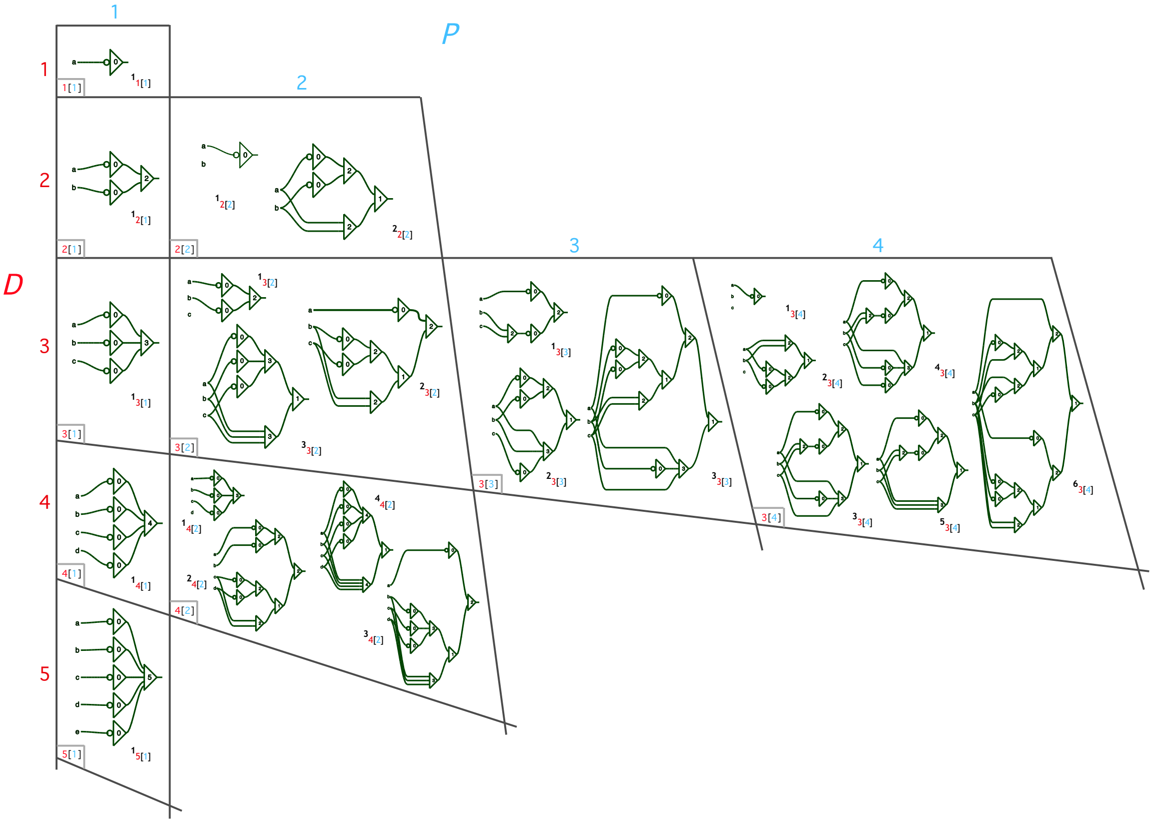

Figure 3. The "periodic table" of elemental neural circuits. Each cell in the table contains a family of circuits that each take D inputs and "fire" on exactly P distinct inputs. Every neural circuit on the table computes a distinct Boolean function, and every neural circuit that computes a Boolean function is isomorphic to one on the table. Each circuit is shown in approximately minimal form. The notation D[P] refers to the family with D dimensions and P combinations of inputs that yield a positive output. Circuits in family $D[P]$ are labeled $\mathbf{1}_{D[P]}, \mathbf{2}_{D[P]}, \ldots$. Order of these labels is arbitrary.

and more complex circuits. No matter how complex, though, each circuit is the simplest circuit that computes its particular input-output function (or at least, the simplest that can be produced by combinations of the reduction steps shown in Fig. 2). It is conceivable, though admittedly completely speculative, that the elemental circuits illustrated in the table might, up to functional equivalence, play the role of the "canonical microcircuits" sought by Douglas and Martin (1992) as the basic units of cortical machinery.

Minimization and learning

As discussed above, the process of finding the minimal circuit for each input-output function corresponds to algebraic simplification in the parallel realm of symbolic Boolean functions. The analog of this minimization process in the space of neural circuits is reorganization of circuits to produce simpler circuits, each reorganization step consisting of one of the operations shown in Fig. 2. An example of a complete circuit reorganization is shown step-by-step in Fig. 4.

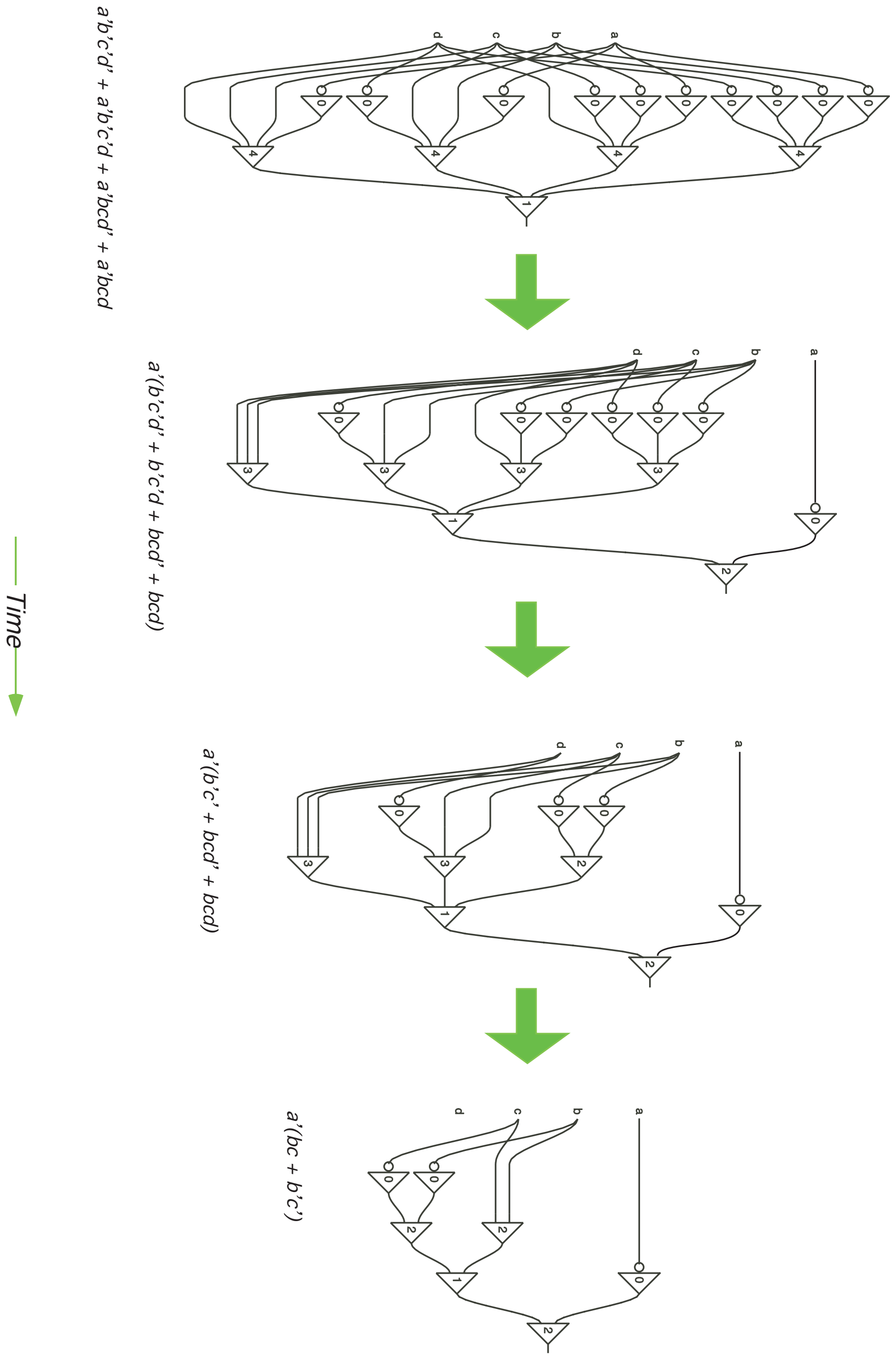
a'b'c'd' + a'b'c'd + a'bcd' + a'bcd
a
b
c
d
a'(b'c'd' + b'c'd + bcd' + bcd)
a'(b'c' + bcd' + bcd)
a'(bc + b'c')
Time

Figure 4. An example of the successive minimization of a neural network. Each successive circuit is equivalent to but simpler than the previous ones. The final (rightmost) circuit is irreducible. The complexity of this final circuit correlates with "subjective complexity" of a learned concept, suggesting that the process of minimization could correspond to the process of learning.

There is evidence from behavioral studies that this process of simplification corresponds to the process of *learning*. As mentioned above, subjects asked to memorize one of these input-output functions (in the form of objects whose features realize only "positive" inputs) are successful in doing so to an extent proportional to the Boolean complexity, i.e., to the length of the minimal formula (Feldman, 2000). This suggests that learning involves some process of simplification or dimensional reduction, with the final representation of the learned concept corresponding to the minimal formula. Of course this idea has many precedents in the literature of psychology (e.g., the Gestalt idea of *Prägnanz* or phenomenal simplicity) and computational neuroscience (e.g., the idea of relaxation and simulated annealing; Hinton & Sejnowski, 1986).

Hence it is natural to ask whether the process of learning might *literally* involve neural reorganization of a type related to the minimization shown in Fig. 4. This possibility is made more tantalizing by the recent discovery of neurogenesis in humans (Gould, Reeves, Graziano, & Gross, 1999), suggesting that learning may involve the addition and subtraction of neurons to existing circuits rather than simply changes in connection weights as usuall thought. What rules might govern the topological reorganization of neural networks? Simplification corresponding to Boolean minimization provides a concrete hypothesis—though admittedly a very speculative one. Indeed little biological evidence either for or against the idea of neural circuit simplicitation can be found in the existing neuroscientific literature, though complexity-minimization does play a prominent role in some theoretical treatments (e.g. Parberry, 1994).

Moreover, the idea that neural circuits continually reorganize into simpler ones carries with it an intriguing corollary. If this is true, than *stable neural circuits in the brain ought to always be minimal in form*—a kind of "principle of neural parsimony."

In turn this principle entails a concrete physiological prediction: neural circuits that are *not* minimal (e.g. the non-terminal forms in Fig. 4) *ought never to be observed in the brain.* Because the circuits given in the catalog above exhaust the logically distinct minimal forms, this prediction is equivalent to the prediction that all observed neural circuits will be isomorphic to one found in the catalog.

What is left out?

As will be clear to any working neuroscientist, a tremendous number of important properties of real neural circuits have been completely ignored in the above discussion.

Principal among these is the fact the real biological neurons are more complex and powerful than these simple abstract neurons (see Koch, 1999); indeed much of modern electrophysiology is devoted to the discovery of a more realistic computational model for the neuron. Strictly, though, the typology discussed above depends only on the fact that neurons compute Boolean functions, which is still true if all signals are either *on* (action potential) or *off* (no action potential), which is still a part of the standard model. If individual neurons compute different atomic functions than assumed above, though, needless to say the resulting circuit typologies will be differ from the illustrations given here. The typology would still be correct, but the resemblance between real circuits and canonic forms would be more abstract.

More seriously, another crucial factor ignored in the above is the *dynamics* of neural computation. The above circuits ignore the component of time completely, assuming that each neuron in effect waits for all of its inputs to be completely assembled before computing their sum. Of course this is unrealistic: indeed, many of the essential functional qualities of neural circuits probably relate to the change in population behavior over time (Sejnowski, 1976), time-dependent variations in firing rates (Brenner, Agam, Bialek, & de Ruyter van Steveninck, 1998), etc., all of which are ignored here.

Another factor left out is real-valued variation in weights along input connections. In most modern investigations of neural computation, such variations are usually thought to be the main mechanism whereby circuit function is updated over learning, while the topology of the network is usually assumed to be fixed. In real neural function it is unclear if the latter assumption is indeed realistic, or whether rather "function determines form" (Albertini & Sontag, 1993). But certainly the assumption that connection strength can vary smoothly is correct, and enormously impacts the computational power of the resulting network. Indeed the simplifying assumption at fault here is that circuits compute only Boolean rather than real-valued functions, which is obviously false.

These and other factors ignored here are without a doubt central to a complete understanding of the function of neural circuits. However their importance does not diminish the parallel role of logical function, i.e., the qualitative mathematical properties of the input-output function computed by the neural circuit. The above classification is offered in the hope of shedding light on the latter, while admittedly evading these other factors.

Summary and conclusion

This brief paper has presented a catalog of neural circuits, each of which serves as a canonic example of a single type in a mathematically well-defined typology of Boolean function classes. As discussed above, these basic function classes are well-known from mathematical logic, but had not previously been translated into equivalent neural networks. These circuits are certainly not exhaustive of real biological circuit types, because, as discussed, the typology only relates to the Boolean function computed by the circuit, and does not relate to other types of functions. However the principled nature of the typology raises the hope that other (perhaps more realistic) types of functions might be similarly classified, shedding some light on why biological circuits are organized as they are.

The more intriguing, but admittedly highly speculative, point raised by this analysis is the role of complexity-minimization in determing the architecture of real biological circuits. There is indeed little evidence at this time that reorganization of circuit topology plays an important role in biological learning, let alone reorganization in the direction of reduced complexity as illustrated by Fig. 4. It is the hope of the present author that this article might help to raise interest in this issue among neuroscientists, in order that evidence for or against this hypothesis might be found, with the hope that such evidence might lead in turn to a more complete understanding of the functional organization of neural circuits.

References

Aiken, H. H., & the Staff of the Computation Laboratory at Harvard University. (1951). *Synthesis of electronic computing and control circuits*. Cambridge: Harvard University Press.

Albertini, F., & Sontag, E. D. (1993). For neural networks, function determines form. *Neural networks*, *6*, 975–990.

Brenner, N., Agam, O., Bialek, W., & de Ruyter van Steveninck, R. (1998). Universal statistical behavior of neural spike trains. *Physical Review Letters*, *81*(18), 4000-4003.

Douglas, R. J., & Martin, K. A. C. (1992). Exploring cortical microcircuits: a combined anatomical, physiological, and computational approach. In T. McKenna, J. Davis, & S. F. Zornetzer (Eds.), *Single neuron computation.* Academic Press.

Feldman, J. (2000). Minimization of Boolean complexity in human concept learning. *Nature*, *407*, 630–633.

Feldman, J. (2003). A catalog of Boolean concepts. *Journal of Mathematical Psychology*, *47*(1), 98–112.

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: A guide to the theory of NP-completeness*. New York: Freeman.

Gould, E., Reeves, A. J., Graziano, M. S. A., & Gross, C. G. (1999). Neurogenesis in the neocortex of adult primates. *Science*, *286*, 548–552.

Harrison, M. A. (1965). *Intoduction to switching and automata theory*. New York: McGraw-Hill.

Hinton, G. E., & Sejnowski, T. J. (1986). Learning and relearning in Boltzmann machines. In D. E. Rumelhart, J. L. McClelland, & the PDP Research Group (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition.* Cambridge, MA: M.I.T. Press.

Koch, C. (1999). *Biophysics of computation: Information processing in single neurons*. New York: Oxford University Press.

McCulloch, W. S., & Pitts, W. H. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, *5*, 89–93. (Reprinted in W. S. McCulloch, "Embodiments of Mind"; Cambridge, MIT Press, 1965)

Parberry, I. (1994). *Circuit complexity and neural networks*. Cambridge: M.I.T. Press.

Sejnowski, T. J. (1976). On the stochastic dynamics of neuronal interation. *Biological cybernetics*, *22*, 203–211.

Shepard, R. N., Hovland, C. L., & Jenkins, H. M. (1961). Learning and memorization of classifications. *Psychological Monographs: General and Applied*, *75*(13), 1–42.