ARTICLE
Demonstrations of Neural Network Computations Involving Students

Christopher J. May
Life Sciences Department, Carroll University, Waukesha, WI 53186

David Marr famously proposed three levels of analysis (implementational, algorithmic, and computational) for understanding information processing systems such as the brain. While two of these levels are commonly taught in neuroscience courses (the implementational level through neurophysiology and the computational level through systems/cognitive neuroscience), the algorithmic level is typically neglected. This leaves an explanatory gap in students' understanding of how, for example, the flow of sodium ions enables cognition. Neural networks bridge these two levels by demonstrating how collections of interacting neuron-like units can give rise to more overtly cognitive phenomena. The demonstrations in this paper are intended to facilitate instructors' introduction and exploration of how neurons "process information."

Key words: neural network;algorithmic level; demonstration; teaching; physiology;undergraduate

David Marr famously proposed that three levels of analysis are required to fully understand an information processing system such as the brain: the implementational level, the algorithmic level, and the computational level (Marr, 1982). The implementational level of analysis concerns the physical substrate of a system, such as neurons in the brain or electronic components in a computer. At this level, for example, neurophysiology examines the electrochemical activities of neurons. The algorithmic level addresses the information processing functions carried out by these activities. Artificial neural networks are particularly well suited for analyzing cortical algorithms because they are explicitly constructed as systems mapping inputs to outputs using nodes (neurons) and connections (synapses) inspired (at varying levels of implementational realism) by neurophysiology. Relatively simple examples include depictions of how the receptive fields of photoreceptors combine to give rise to the ganglion cells' center-surround receptive fields, which combine further downstream to create the receptive fields of simple cells in the primary visual cortex. Finally, the computational level identifies teleological explanations for a system's functioning. Methodologies such as functional magnetic resonance imaging, positron emission tomography, and transcranial magnetic stimulation, for example, are useful in triangulating the gross functions or goals of a brain area. Thus, we know that the prefrontal cortex plays a role in top-down attention; however the precise ways (viz., algorithms) by which the prefrontal cortex executes this role are still under active investigation.

Biopsychology and neuroscience textbooks typically give extensive treatment to the implementational and computational levels of analysis, but dedicate little coverage to the algorithmic level. The teaching of neural networks, in particular, is typically neglected in introductory and even advanced biopsychology/neuroscience courses. For example, John Pinel's *Biopsychology* (6th Ed.) progresses from "The Anatomy of the Nervous System" (Ch. 3) and "Neural Conduction and Synaptic Transmission" (Ch. 4) to "The Research Methods of Biopsychology" (Ch. 5) and "The Visual System" (Ch. 6). Breedlove, Rosenweiz, and Watson's *Biological Psychology* (5th Ed.) differs somewhat, by following chapters on neuroanatomy and neurophysiology with chapters on neuropharmacology and the endocrine system. In their chapter on neurophysiology, there are just five paragraphs which discuss neural circuits. These paragraphs focus on reflex arcs and neural convergence/divergence (pgs. 80, 82). Bob Garrett's *Brain and Behavior* (2nd Ed.) devotes just over a page to introducing neural networks (pgs. 43-44) by describing how some researchers have used neural networks to better understand the human brain and our cognitive abilities. Similarly, Gazzaniga, Ivry, and Mangun's popular *Cognitive Neuroscience* (1st Ed.) text discusses neural networks by highlighting a few cases where neural networks have been used to study the mind and brain. One notable exception to this pattern is Baars and Gage's recent textbook *Cognition, Brain, and Consciousness* (1st Ed.). Here, neural networks feature prominently in the chapter on neurons, and an accompanying appendix provides more detailed information. Generally, however, while mention may be made that most cognitive and behavioral functions arise from the coordinated activity of many interconnected neurons, neural networks are not discussed in any greater depth.

Neglecting the algorithmic level leaves significant gaps in students' understanding of the brain. Students commonly wonder, "What do sodium, ion flow, and action potentials have to do with perception, thinking, and emotion? What does it mean for a neuron to 'process information'? How is neurophysiology relevant to psychology?" The foundational idea of neuroscience, that the mind can be understood by studying the brain, is left underdeveloped.

Neural networks bridge neurophysiology with psychology by enabling 1) a redescription of neural processing in information processing terms, and 2) demonstrations of how collections of information processors give rise to more overtly cognitive phenomena.

The latter is particularly important. Because Interactions between neurons are highly non-linear, understanding the physiology of a single neuron sheds little light on the functioning of networks of neurons. Indeed, the brain is paradigmatic of a complex system, in which information processing functions are emergent rather than programmed. Emergent phenomena conform to the mantra, "the whole is greater than the sum of the parts" (see demonstrations below for examples). Complexity theorist Stuart Kaufman describes such systems as providing "order for free" (Kaufman, 1995). Without exploring precisely how this order (viz., information processing) can be provided for free (viz., as a result of non-linear neural interactions), a large explanatory gap will remain between the neuron and the brain.

My basic goals in conducting the demonstrations below are twofold: 1) to demystify brain function by illustrating how relatively complex processes can emerge from the interaction of simple elements, and 2) to dramatize this emergence by having students take on the role of a neuron, wherein they have no idea why they are enacting their prescribed role (viz., they have no knowledge of the larger goal of the network they constitute), but nonetheless are integral components in producing a relatively sophisticated product. In other words, students enact a debunking of the homuncular fallacy. While not necessary in a lower level course (e.g., Introductory Psychology), in upper level Neuroscience courses, I embed these demonstrations in a larger introduction to neural networks. Students are first given an information processing description of neural functioning, followed by a brief primer on the "integrate-and-fire" neural model, as outlined below.

MATERIALS AND METHODS

Individual neurons may be described as "detectors" (see O'Reilly and Munakata, 2000). Accordingly, the firing of a single neuron announces the presence of (detects) some set of input conditions, much like an "idiot light" in a car (Cummins and Poirier, 2004). For example, the seat-belt light on a car dashboard is either on or not, indicating that the seat belt is either plugged in or not. The gas light comes on when a certain threshold of fuel level has been crossed. In a home, a smoke detector fires if a certain level of smoke is present. Likewise, if a neuron receives sufficient excitatory input, it will fire. What an individual neuron detects depends on which neurons it is connected to and at what strength they are connected.

In an "integrate-and-fire" neural model, this firing threshold may be implemented by a step activation function (a sigmoid function is more biologically realistic, but step functions are easier to compute). A neuron first sums together all its excitatory and inhibitory inputs (integrates) and then passes that sum through an activation function, which determines the output of the neuron. With a step activation function, the neuron will fire if the integration of inputs to the neuron surpasses some threshold, and will not fire otherwise (in an all-or-none fashion, like an action potential). Equations 1 and 2 summarize this two-step integrate-and-fire neural model. NET represents the net input to an output neuron (j), summing, for all inputs (i), the product of the input activity (x) of an i^{th} neuron and the weight (w) between the i^{th} and j^{th} neurons. Weights correspond to the synaptic strength between two neurons and can range continuously from -1 (representing a strong inhibitory connection) to 1 (representing a strong excitatory connection). ACT takes NET as an input, and determines the output based on the threshold value.

(Eq. 1) $NET_j = \sum x_i w_{ij}$ for all i

(Eq. 2) $ACT_j = 0$ if $NET_j <$ threshold
1 if $NET_j >$ threshold

The particular function (mapping between input(s) and output(s)) computed by a neural network can be modified by changing the weights between neurons and/or their firing thresholds. By changing the set of neurons whose activity is required for the output to fire, these modifications may alter what an output neuron detects.

Demonstration 1

As a first example, consider a neural network consisting of just three neurons, two constituting an "input layer," which both project to a single "output" neuron, as in Figure 1. The firing threshold of the output neuron is set at 0.5, and the weights between each input-output pair are also set to be 0.5. Assume that the activity of input neurons is binary (0=off, 1=on). With two input units, there are four combinations of activation values. The conditions under which the output neuron will fire can be determined by constructing a truth table (Table 1). The activation of the output neuron is determined by plugging the activation values of the input neurons into Eqs. 1 and 2. For the particular pattern of weights and threshold in Figure 1, the output neuron will fire whenever any input is on, but will not fire if both inputs are off. This is the "inclusive-or" function.

This truth table (Table 1) can be derived in class by having students enact the neural network in Figure 1. To do this, the instructor should select three students. Each student will represent a neuron that either "fires" or is silent. Firing is indicated by the student either raising a particular sign, or their hand (this demonstration will be articulated assuming a sign). Students should be arranged into a triangular structure as in Figure 1. Each student should then be given a single note card with a particular set of instructions and a sign to hold up (for reasons that will become clear, ask the students to read their instructions, but not the sign they have been given):

- Student 1 (Input Layer)

Instructions:

If your shoulder is tapped, raise your sign and then tap the shoulder of the person directly in front of you.

Sign:

"Sally had a burger."

- Student 2 (Input Layer)

Instructions:

If your shoulder is tapped, raise your sign and then tap the shoulder of the person directly in front of you.

Sign:

"Sally had a Coke."

- Student 3 (Output Layer)

Instructions:

If your shoulder is tapped, raise your sign.

Sign:

"Sally had a burger OR a Coke."

The instructor can now conduct four tests to see how this student neural network responds. In the first test, the instructor taps the shoulder of one input student-neuron. That student will then raise their sign and tap the student in front of them, who will respond by raising their sign. The instructor or another student can then enter a 1 under the "Input1" column, a 0 under the "Input2" column, and a 1 under the Output column of a truth table on the white board. In the second test, the instructor taps the other input student-neuron, which will have the same effect of producing a response in the output student-neuron. Now enter a 0 under Input1, a 1 under Input2, and a 1 under Output. In the third test, the instructor taps both input student-neurons, which will also cause an output response. Finally, if the instructor does nothing, there should be no output neural activity, and so 0s can be entered under Input1, Input2, and Output.

For students watching this demonstration, the logical relationship between inputs and output should be fairly clear. The output ("Sally had a burger OR a Coke") fires when either one or both of the propositions ("Sally had a burger", "Sally had a Coke") is true (indicated by the firing of that input neuron). However, for the students enacting this computation, they will not know what function they helped implement. Since they have not read their signs, they simply raise their sign or do not, depending on if they received a tap. This is appropriate, since no single neuron in the brain "knows" the larger computations in which it participates. Asking students to share their instructions with the class, and then questioning them about what they think they were doing will dramatize this point. Again, while the student-neurons do not know the information-processing role they played in the large network, the audience will. This demonstrates emergence: clueless neurons following relatively simple instructions were integral to the production of a more complex function. That function is not reducible to the activity of any single neuron, but rather arises because of the coordinated activity of all the components (that is, the "whole is greater than the sum of its parts;" there is no "homunculus" in the brain).

Note that the particular values of the weights and threshold are not important, only the relationship between them matters. Thus, if the weights were increased to 0.9, as was the threshold, this network would still compute the inclusive-or function. Students often have difficulty appreciating this point initially, and so it may be necessary to repeat it with each demonstration. Because information processing in the brain is emergent, relationships take center-stage.

Note also that this algorithm for computing inclusive-or was implemented with a very different substrate than the brain uses (tapping people vs. electrochemical neurons). Indeed, any physical/biological setup in which the relations on the instruction cards obtain can be said to implement the inclusive-or algorithm. If a system could be constructed out of bubble-wrap, whereby the popping of one bubble systematically caused the popping of some other bubble(s), then that bubble-wrap system is a type of information processing device. This underscores that the algorithmic level is indeed distinct from the implementational level of analysis.

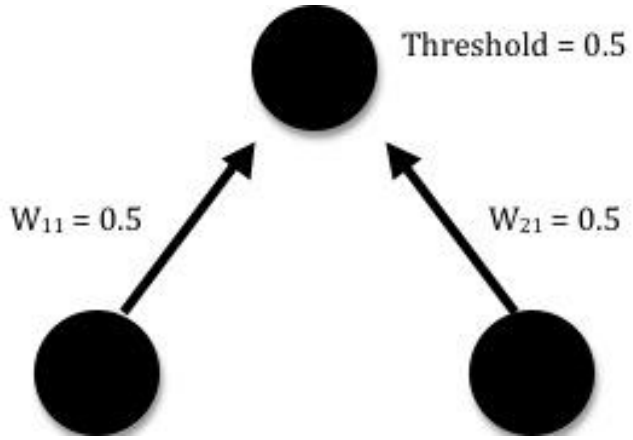

Figure 1. Neural network which computes the "inclusive-or" function. Weights (W) refer to the synaptic strength connecting either the first of second input unit to the sole output unit.

Input1	Input2	Output
0	0	0
0	1	1
1	0	1
1	1	1

Table 1. Truth table for "inclusive-or" function. 0 = off/false, 1 = on/true.

Demonstration 2

A second neural network demonstration enacts the "and" function. As mapped in Table 2, the output neuron is active here if and only if both inputs are on. To implement this, the threshold of the output neuron is elevated from 0.5 to 1.0 (Figure 2).

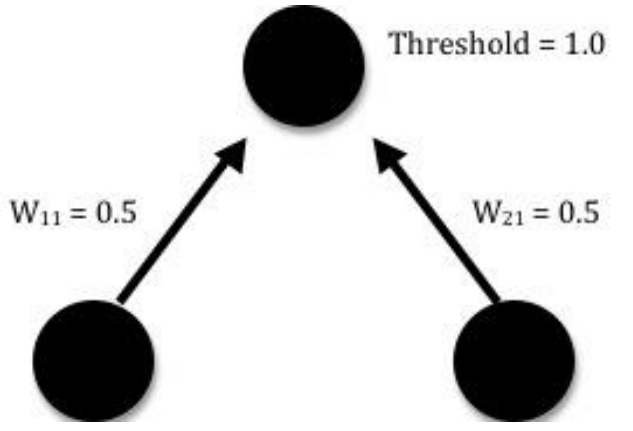

Figure 2. Neural network which computes the "and" function. Weights (W) refer to the synaptic strength connecting either the first of second input unit to the sole output unit.

Input1	Input2	Output
0	0	0
0	1	0
1	0	0
1	1	1

Table 2. Truth table for the "and" function. 0 = off/false, 1 = on/true.

Here again, three students should come to the front of the class and get in the formation depicted in Figure 2. Now students receive new instructions and signs:

- Student 1 (Input Layer)
Instructions:
If your shoulder is tapped, raise your sign and then tap the outside shoulder of the person in front of you.
Sign:
"Sally had a burger"

- Student 2 (Input Layer)
Instructions:
If your shoulder is tapped, raise you sign and then tap the outside shoulder of the person in front of you.
Sign:
"Sally had a Coke"

- Student 3 (Output Layer)
Instructions:
If BOTH of your shoulders are tapped, raise your sign. Otherwise, do nothing.
Sign:
"Sally had a burger AND a Coke"

By executing the same four combinations of input activity as in Demonstration 1, the rest of the class will clearly see that this network implements the "and" function. Once again, the student-neurons will not have much insight into the larger function they enacted.

To reiterate the importance of thinking relationally (and as a test for comprehension), students may be asked to come up with a different set of weights and threshold values that would also produce the "and" function.

Demonstration 3
A third neural network demonstration may be conducted to introduce students to two additional elements: inhibitory connections and "hidden" layers. Students sometimes have difficulty conceiving what role inhibition might play in the brain; this demonstration gives an example of one function, "exclusive-or," which requires inhibition. Exclusive-oris also a function that cannot be computed with just the two layers of neurons used in Demonstrations 1 and 2. Incidentally, this limitation of a two-layer network was pointed out in 1969 by Marvin Minsky and Seymour Papert, halting much neural network research until the mid-1980s. To compute exclusive-or, three layers of neurons are required: an input layer, a "hidden" layer (so-called because the activity of neurons in this layer is not directly manipulated, as with the input neurons, nor is their output activity necessarily visible, as with the output neurons), and an output layer.

In the exclusive-or function, the output neuron will fire if one of the input neurons is active, but not if both are active (Table 3). For this demonstration, five students need to be organized into the formation illustrated in Figure 3. Notice that there are two types of arrows between neurons in Figure 3, representing excitatory and inhibitory connections. To enact these connections, student-neurons in the input layer will produce two types of actions if they are tapped: tapping the student-neuron in front of them (representing an excitatory connection) and resting a hand on the shoulder of the person diagonal to them (representing an inhibitory connection). The activity of student-neurons in the hidden layer will, in turn, depend on the type of inputs they receive.

Input1	Input2	Output
0	0	0
0	1	1
1	0	1
1	1	0

Table 3. Truth table for "exclusive-or" function. 0 = off/false, 1 = on/true.

The instructions and signs to be given to each student-neuron are as follows:

- Student 1 (Input Layer)
Instructions:
If your shoulder is tapped:
1) with your outside hand, tap the shoulder of the person directly in front of you, raise your sign, and then
2) with your inside hand, rest it on the shoulder of the person diagonal to you and leave it there.
Sign:
"Sally had a burger"

- Student 2 (Input Layer)
Instructions:
If your shoulder is tapped:
3) with your outside hand, tap the shoulder of the person directly in front of you, raise your sign, and then
4) with your inside hand, rest it on the shoulder of the person diagonal to you and leave it there.
Sign:
"Sally had a Coke"

- Students 3 & 4 (Hidden Layer)
Instructions:
You will receive one of three possible signals: a) you will feel nothing, b) you will feel a tap on your

shoulder, or c) you will feel a tap on one shoulder and a hand resting on the other shoulder.

a) If you feel nothing, DO NOTHING.
b) If you feel a tap on your shoulder BUT DO NOT feel a hand resting on the other shoulder (wait a couple seconds to be sure), tap the shoulder of the person in front of you.
c) If you feel a tap on your shoulder AND a hand resting on your other shoulder (wait a couple seconds to be sure), DO NOTHING.

- Student 5 (Output Layer)

Instructions:

If your shoulder is tapped, raise your sign.

Sign:

"Sally had EITHER a burger OR a Coke."

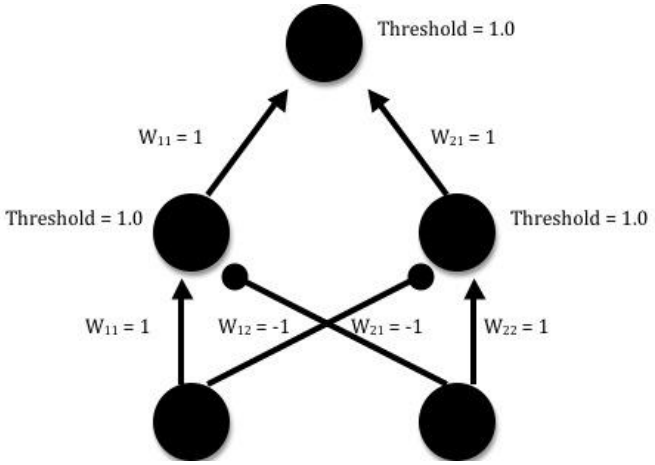

Figure 3. Neural network which computes the "exclusive-or" function. Weights (W) refer to the synaptic strength connecting neurons. Lines with arrowheads denote excitatory connections (and have positive weight values), whereas lines with ball heads denote inhibitory connections (and have negative weight values).

Again, the instructor should go through the four combinations of input activity to determine how the output student-neuron responds. As a comprehension test (either in class or on a test), the instructor may "lesion" one of the hidden layer neurons, and ask students to work out how the truth table they previously derived would be changed by this deletion.

DISCUSSION

The preceding demonstrations are rooted in the work of McCulloch and Pitts (1943), who first articulated how small networks of neurons may implement logical functions. Logical functions form the basis of modern computing. While computers and brains clearly have significant differences, pointing out this parallel to students may help them better understand how neurons could participate in computationally sophisticated processes. These demonstrations begin with the metaphor of a neuron as detector, and enact how relatively simple two and three-layer networks of neurons implement a few logical functions. Since McCulloch and Pitts pioneering work, it has been shown that a three-layer neural network (consisting of input, hidden, and output layers) can compute any continuous function, and a four-layer network (with two hidden layers) can compute any computable function (Cybenko, 1988). Thus, most neural networks, like standard computers, are so-called "Universal Computers." With the "exclusive-or" network, students are brought up to the level of architectural complexity that can compute most functions. This parallel between the brain and the computer can help students conceptually bridge the implementation level (e.g., neurons, electronic components) with the computational level (e.g., object recognition, Excel). While computers may be technically mysterious, they are not usually considered to be metaphysically mysterious. Linking the information processing of neurons, as demonstrated above, to computers in this way, then serves to demystify brain function.

These demonstrations necessarily abstract away from biological realism. For example, real neurons are noisy, not perfectly silent in the absence of stimulation. A sigmoid function is also a more realistic activation function than a step-function. In a sigmoid function, increasing excitatory input above threshold will increase the rate of action potentials, until some upper-limit of saturation is reached. Simulations with a high degree of neurophysiological realism require significant computational resources (in terms of memory and processing time), which severely limits the ability to also explore interactions between neurons. Nonetheless, to the extent that any model is useful, key processes hypothesized to be responsible for the overall functional dynamics of a system will be extracted and explored. For neurons, non-linearity is paramount. Non-linearity enables neurons to be effective detectors, and non-linearity underlies the complex information processing that can emerge from collections of interacting neurons. While the specific algorithm executed by an actual neural network in the brain is an empirical question (see O'Reilly and Munakata, 2000 for many examples), the three demonstrations detailed here provide a minimal set with which to bridge the implementational and computational levels of the brain.

These three demonstrations have been used in several classes, ranging from an introductory psychology class to an upper level neuroscience class. In all cases, the demonstrations accomplish the goal of helping to demystify the brain. They also engage students in material that many (particularly in an introductory class) may not find very interesting. In an upper-level class, these demonstrations introduce students to neural networks more generally. However, students usually require a fair amount of repetition and practice manipulating networks (e.g., by changing weight and threshold values, and determining the consequences of lesioning) before they are able to fully comprehend and articulate its dynamics. Students typically have little history in thinking about neurons from an information-processing perspective, and often need time to appreciate the importance of relationships between neurons. Nonetheless, these relatively simple demonstrations provide a vehicle to teach students how the electrochemical processing of neurons

relates to the information processing dynamics of the brain. The amount of time an instructor chooses to spend on the demonstrations depends on his/her goal. One or more of them can be done very quickly as a simple demonstration, or they could serve as the centerpiece for a larger discussion spanning several class periods. Through them, students should come to have a better understanding and appreciation of how neurophysiology connects with the high-level descriptions of brain processing given in cognitive and systems neuroscience.

REFERENCES

Baars BJ, Gage NM (2007) Cognition, brain, and consciousness: Introduction to cognitive neuroscience. New York: Academic Press.

Breedlove SM, Rosenzweig MR, Watson NV (2007) Biological Psychology: An introduction to behavioral, cognitive, and clinical neuroscience (5th Ed.). Sunderland, MA: Sinauer Associates, Inc.

Cummins R, Poirier P (2004) Representation and indication. In: Representation in Mind (Staines P, Slezak P eds), pp21-40. Elsevier.

Cybenko G (1988) Continuous valued neural networks with two hidden layers are sufficient. Technical report, Department of Computer Science, Tufts University, Medford, MA.

Garrett B (2009) Brain and behavior: An introduction to biological psychology (2nd Ed.). Los Angeles, CA: Sage.

Gazzaniga MS, Ivry RB, Mangun GR (1998) Cognitive neuroscience: The biology of the mind. New York, NY: Norton & Co.

Kaufman S (1995) At home in the universe: The search for the laws of self-organization and complexity. New York, NY: Oxford University Press.

Marr D (1982) Vision: A computational investigation into the human representation and processing of visual information. New York, NY: Henry Holt & Co., Inc.

McCulloch WS, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. Bull Math Biophys 5:115-137.

Minsky M, Papert S (1969) Perceptrons. Cambridge, MA: MIT Press.

O'Reilly RC, Munakata Y (2000) Computational explorations in cognitive neuroscience: Understanding the mind by simulating the brain. Cambridge, MA: The MIT Press.

Pinel JPJ (2006) Biopsychology (6th Ed.). New York, NY: Allyn & Bacon.

Received August 03, 2009; revised December 09, 2009; accepted December 12, 2009.

Address correspondence to: Dr. Christopher J. May, Life Sciences Department, 100 N. East Ave., Carroll University, Waukesha, WI 53186 Email: cmay@carrollu.edu