ARTICLE
Pencil-and-Paper Neural Networks: An Undergraduate Laboratory Exercise in Computational Neuroscience

Kevin M. Crisp, **Ellen N. Sutter**, **and Jacob A. Westerberg**
Neuroscience Program, St. Olaf College, Northfield, MN USA 55057.

Although it has been more than 70 years since McCulloch and Pitts published their seminal work on artificial neural networks, such models remain primarily in the domain of computer science departments in undergraduate education. This is unfortunate, as simple network models offer undergraduate students a much-needed bridge between cellular neurobiology and processes governing thought and behavior. Here, we present a very simple laboratory exercise in which students constructed, trained and tested artificial neural networks by hand on paper. They explored a variety of concepts, including pattern recognition, pattern completion, noise elimination and stimulus ambiguity. Learning gains were evident in changes in the use of language when writing about information processing in the brain.

Key words: artificial neural networks; learning; memory; Hebbian plasticity

Undergraduate neuroscientists struggle to understand and articulate the connections between cellular processes governing synaptic integration and excitability, and the higher-level cognitive processes of the brain. In one assessment, students were presented with a series of terms related to neuroscience, and were asked to classify them according to discipline. They classified the term "networks" with terms like "consciousness," "behavior" and "brain," but not with terms like "cells," "voltage-gated channels," "modeling" or "electrical potential (Crisp and Muir, 2012). This suggests that undergraduates, while able to master content at the cellular level and also at the behavioral level, perceive these subfields as independent domains rather than hierarchical levels of brain organization.

Undergraduates have a tendency to offer phenomenological explanations for many types of brain functions. While awe of the phenomena is valuable, mechanistic understanding is required for posing testable hypotheses, understanding the value of interdisciplinary approaches and developing self-guided habits of inquiry. For example, when we asked students how the brain can read distorted text that computers struggle to interpret, typical answers included "the brain is used to seeing texts in a variety of fonts," "we can use our memory" and "the brain can often fill in the gaps." Artificial neural networks are often capable of dealing with ambiguity, noise and missing data because of the robust and redundant nature of distributed representations. We reasoned that showing undergraduates how artificial neural networks could perform these brain-like tasks would help them develop a more mechanistic understanding of brain function.

Similarly, when we asked undergraduates why the brain struggles to learn associations between large numbers of similar stimuli (like Morse code characters or families of chemical structures), typical answers were quite phenomenological: "the patterns are very similar to each other," "it takes a lot of repetition," and "it would be easier if there was a pattern." Artificial neural networks also struggle with learning associations between indistinctive stimuli. In fact, they start making mistakes after learning a sufficiently large set of similar associations. Multiplication facts, for instance, are associations with a great deal of similarity between the stimuli. Adults tested on multiplication facts tend to produce a 25% error rate, with the majority of errors being non-random (Graham, 1987). For example, a test subject might incorrectly answer that 6 * 9 is 56 rather than 54, because 56 is a learned product and is approximately the right magnitude. Viscuso et al (1989) showed that an artificial neural network trained on multiplication facts also had an error rate of 30%, and made errors that were non-random and quite similar to those made by humans. We wondered if showing students that artificial neural networks could be used to explain certain errors in recall when learning associations between very similar stimuli, they would develop a more mechanistic understanding of learning-related confusion and forgetting.

McCulloch and Pitts (1943) proposed a simple but computationally powerful model of a neuron as a binary threshold unit. This seminal work challenged neuroscientists to consider the brain as a vast network of 10 billion interconnected parallel processors. While the McCulloch-Pitts model is so over-simplified as to raise serious questions about whether it is a fair representation of biological neurons at all, it does offer to serve as a cognitive bridge between synaptic plasticity and high-level computation. The model is inspired by cellular neurobiology in that artificial neurons integrate inputs from presynaptic neurons according to the relative strengths of individual synaptic connections, and fires only if the net input exceeds a threshold value:

$$n_i(t+1) = \theta\left(\sum_j w_{ij} n_j(t) - \mu_i\right), \qquad (1)$$

where n_i represents the activation (or "firing") state (0 or 1) of neuron-like unit i at time t. $\Theta(x)$ is a thresholding function that equals 1 if $x \geq \mu_i$ (the threshold at which unit i fires an impulse), and 0 if $x < \mu_i$. The weight w_{ij} represents the synaptic strength of the synapse from unit j onto unit i.

Arrays of McCulloch-Pitts neurons are capable of representing highly complex patterns. Hopfield (1982) developed an associative memory model from arrays of McCulloch-Pitts units, and showed that these models could store multiple patterns, or associations, superimposed across the same synapses, such that:

$$w_{ij} = \frac{1}{N}\sum_{\mu=1}^{p} \xi_i \, \xi_j \ , \qquad (2)$$

where N is the number of units in the network, p is the number of associations stored and ξ represents a pattern composed of an array of bits. Learning in Hopfield networks is Hebbian, such that if training necessitates that units i and j fire together, weight w_{ij} is increased to that end. Thus, in the Hopfield network, not only is information stored as a distributed representation across a matrix of synaptic weights, but multiple associations can be superimposed over the same matrix. McNaughton and Morris (1987) used a correlation matrix formalism to illustrate the principles of distributive associative memory in the context of hippocampal circuitry. This formalism simplified the procedures for training and recall from such networks and has been used to demonstrate such phenomena as distributed representations and cued recall.

From their inspiration from biological nerve cells to their computational power, these types of models present a valuable bridge to help undergraduate neuroscientists connect synaptic physiology to higher cognitive processes, such as pattern completion and associative learning. There are many free simulation tools that can offer students an opportunity to develop large-scale simulations and exposure to impressive processing power. However, computer-based simulation labs are often described by students as tedious, frustrating, and inefficient uses of their lab time (Crisp 2012). Because of this, students report plowing through the activities to get them done and learning less from these experiences than from hands-on laboratory exercises.

Here, we present a set of laboratory exercises that have been developed over several years in which students "trained" neural networks using simple correlation matrices on paper. The model, based on the formalism by McNaughton and Morris (1987), allowed students to teach the network associations between patterns of ones and zeros. The students observed how synaptic strengths change throughout the network in response to learning. They measured how many associations the network could reliably learn. They saw the difficulties in recall (output) that occur when input patterns are closely related. Pre- and post-exercise assessments were conducted to determine the extent to which the activity helped the students to understand the following concepts and learning objectives: pattern recognition in the presence of noise; pattern completion when presented with a partial cue; and saturation with minimally-discriminable stimuli.

MATERIALS AND METHODS

Pencil-and-Paper Neural Networks: In the prelab lecture (additional resources: http://pages.stolaf.edu/crisp/), students were asked to consider a simple example of learning in which a squirrel uses the positions of trees in a yard to located cached acorns. This problem could be solved by a simple artificial neural network consisting of an input layer in which the activity of the neurons represents the positions of the trees, and an output layer in which the pattern of neural activity represents the positions of the acorns (Fig. 1).

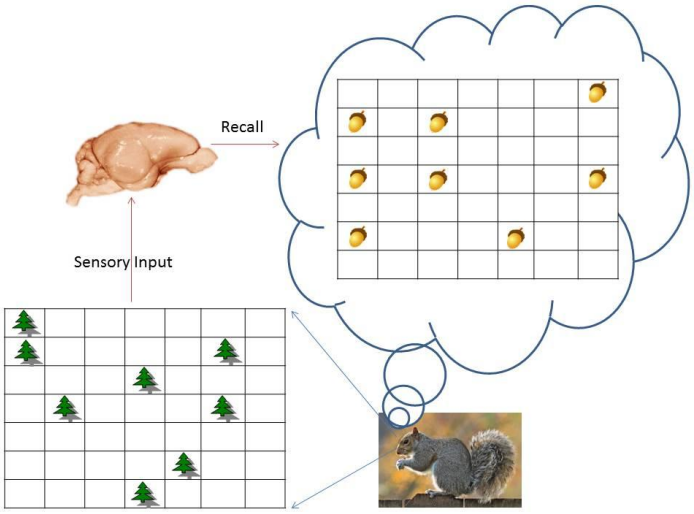


Figure 1. A very simple example of a task in which a squirrel associates a pattern of trees in a yard with a matching map of stashed acorns in the yard. Students were asked to consider a neural network with the task of recalling a map of acorns (output) when prompted with a map of trees (input).

They were told to assume that every neuron in the input layer synapses upon every neuron in the output layer, and that by varying the weights of these synapses, the network "learned" to associate a map of trees with a map of cached acorns (Fig. 2). The map can be represented as a vector of zeros and ones, where a zero indicates that nothing is present, and a one indicates that a tree (input layer) or acorn (output layer) is present. The synaptic connections can be represented as a matrix in which each synapse represents a specific synapse of one input neuron onto one output unit. This synaptic matrix is initially populated by zeros, but zeros change to one when synapses are strengthened during learning.

The students were shown how to teach the network by constructing pairs of arbitrary vectors (such as [1,0,1,1,0,0,0,0,1] and [1,0,0,0,1,0,0,1,1]) and training the network to associate them. This is done by applying Hebb's law in the strictest sense: for each neuron in the input pattern that is active (has a state of 1), every synapse from that unit onto an active unit in the output pattern is strengthened (Fig. 3). To keep the math very simple, strengthened synapses are changed only from zero to one. Note that in the model, synapses are never weakened (changed back to zero from one).

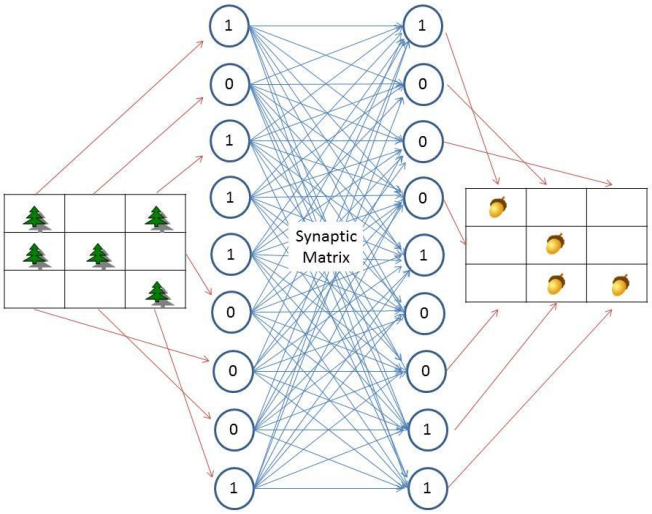


Figure 2. A simple two-layer neural network model that associates an input pattern with an output pattern. The input and output patterns are represented by vectors of zeros and ones. A neuron in the input layer is "active" if its corresponding value in the vector is a one, and it is "inactive" if its corresponding value in the vector is a zero. The synaptic matrix represents the connections of every neuron in the input layer onto every neuron in the output layer. Because every neuron in the network is connected to every other neuron, a layer of neurons can be represented as a 3x3 matrix or as a unidimensional vector.

Later, when the network is presented with the input pattern of activity as a cue, they observe that the output layer generates the appropriate learned output pattern of activity. This was shown to them to approximate Eq. 1. As each column in the matrix represents all of the synapses onto one particular output neuron, the students first calculated the sum of all the products of the weights in the column multiplied by the activity of the respective neuron in the input layer. Then, they applied a thresholding function (described below) is applied to this sum (Fig. 4).

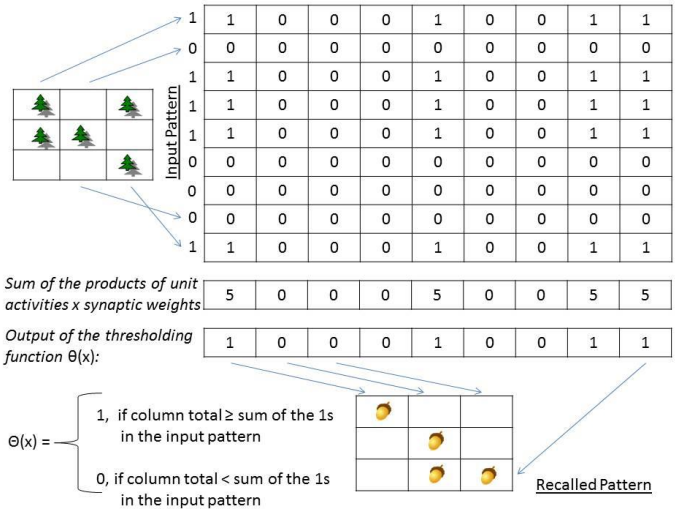


Figure 4. Simulating recall of the correct output pattern when the artificial nerve net is prompted with a trained input pattern. Recall is simulated by multiplying the input vector i by the synaptic matrix M such that the recalled pattern r = i*M (see Eq. 1). In this example, the column sum is five only if all the neurons that were active in the original pattern are also active during recall. The column sum represents the summed synaptic inputs in the dendritic tree of the output unit associated with that column, and the thresholding function reflects the all-or-nothing character of neuronal firing.

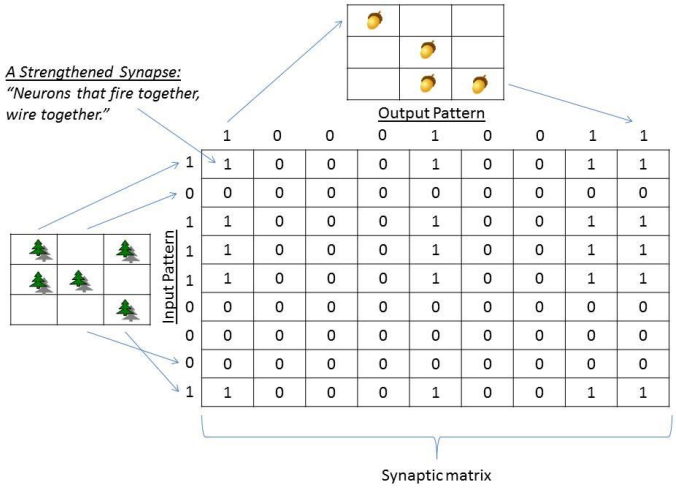


Figure 3. A synaptic matrix that has been trained to associate a nine-neuron input pattern of activity with a nine-neuron output pattern. Note that since every neuron talks to every other neuron, the simplification of the 2D map to a vector has no effect on the connectivity of the network. Each cell in the matrix represents a synapse from the corresponding input neuron onto the corresponding output neuron. After training, every synapse between an active input unit and an active output unit has a strength of 1.

We have simplified the math for the students after McNaughton and Morris (1987) by using only zero or one for each synaptic weight. Note from Eq. 2 that the correct synaptic weights would usually be a rational number between zero and one. The thresholding function θ(x) returns a one if the column sum of products is at least as great as the number of ones in the pattern at the time of learning (e.g., in the original pattern). Thus, if there were four ones in the original pattern, each unit in the output layer would be active if the sum of its inputs multiplied by their respective weights is greater than or equal to four. However, θ(x) returns zero if the sum of the products in a given column is less than the number of ones in the original pattern. This method is a simplification that makes the math easy for the students and places focus on the conceptual learning. It also means that the networks are usually tolerant of noisy input patterns that are close to the original training pattern but have an extra one in the inputs. However, because of this thresholding method, in order for the network to correctly interpret an incomplete input pattern (in which one of the ones in the input vector has been changed to zero), the column sums must be evaluated with respect to the new number of ones in the incomplete pattern rather than the number of ones in the original pattern.

Students were thus instructed that when applying the thresholding function, they should compare the column sum to the lesser of the two following values: (a) the sum of the ones in the input pattern currently presented to the network, or (b) the sum of the ones in the original input pattern with which the network was trained. This thresholding function preserves the use of very simple math and results in networks that are tolerant of noise and incomplete cues:

μ_i = the lesser of (a) the number of ones in the input pattern, or (b) the number of ones in the output pattern. (3)

$$\Theta = \begin{cases} 1, \text{ if } \sum_j w_{ij} n_j(t) \geq \mu_i, \text{ or} \\ 0, \text{ if } \sum_j w_{ij} n_j(t) < \mu_i \end{cases} \quad (4)$$

In-Lab Exercises: The students are presented with the exercises in Appendix A following the pre-lab lecture. A brief explanation of each exercise follows. Sample answers to the problem set are available upon request from the corresponding author (KM Crisp).

(1) The students are first instructed to draw a 4x4 grid representing the connections of 4 input neurons onto 4 output neurons. Each neuron is connected to each other neuron, and the strength of each synapse (initially zero) is represented by a one of the 16 squares in the grid. Note that row 1 in the grid represents the strengths of the synapses of input layer cell 1 onto each of the output layer cells, and so on.

Next, the students "train" their network. They must be made to understand that although our brains are always learning and recalling, neural networks are often conceived of as having discrete "training" and "testing" states. Synaptic strengths are only permitted to change in the training state; they are static in the recall phase. Training this network requires the students to come up with two arbitrary patterns, such as: (1,0,1,0), (0,0,1,1), (1,1,0,1), etc. One pattern will serve as the input pattern and be represented as a vertical vector to the left of the matrix; they represent the states (1 = firing or 0 = not firing) of the four input layer neurons. The other pattern will serve as the output pattern and be represented as a horizontal vector along the top of the matrix, where they represent the states of the four output layer neurons. The process of training simply involves applying Hebb's rule: anytime a presynaptic neuron is active and a postsynaptic neuron is active, the synapse between the two is strengthened (changed from zero to one). But note that the reverse is not true; ones are never changed back to zeros, or the network will forget what it previously learned!

Finally, the students "test" their network. The students present the original input pattern to the network, usually by writing it off to the left of the grid. Because each output unit will sum all of its inputs (times their respective synaptic strengths; see Eq. 1), the next step is to tally the sums of each column. Thus, for each synapse onto output neuron number one, the input neuron's state times its respective synaptic strength (onto output neuron one; e.g., column 1) is added to the tally. If the total column tally is greater or equal to the number of input neurons that are active in the input pattern (see Eqs. 3-4), then the state of the first output unit is 1. Otherwise, the state of the first output neuron is zero. This is repeated three more times for the remaining output layer neurons (columns 2-4). Finally, the calculated output states are compared to the expected output states (the original output pattern that the network was trained with) to see if the network has recalled correctly.

Students should be warned that their networks cannot learn an association in which either input or output state is (0,0,0,0), and that pairing input state (1,1,1,1) with output state (1,1,1,1) will immediately saturate all synapses so that the network will not be able to learn anything further in the next exercises. Finally, if the input pattern has only a single one, the next activity will not work.

(2) Next, the students test the ability of the network to recall an association even if the cue it is prompted with is incomplete. Recognizing patterns even when some data are missing is something that the human brain is very good at. This exercise is meant to show that this ability is an emergent property of the redundancy intrinsic to distributed representations across synapses. The students test the network with a pattern that is similar to the one it was trained on, but is missing a one. For example, if the network was trained to associate input pattern (1,0,1,0) with output pattern (0,0,1,1), for this exercise, it will be tested with input pattern (1,0,0,0) or (0,0,1,0). This is done precisely as the testing was done in exercise (1). The only difference is the pattern used as a cue from which to recall (calculate) the output. This exercise demonstrates a key learning objective (see Appendices B and C, Q2).

(3) Another fascinating feature of the brain is its ability to recognize patterns even when noise (e.g., static) obscures the input. This ability can also be explained in part by the robustness of associations that are stored in a distributed fashion across synapses, allowing networks to generalize across closely related inputs. This exercise also demonstrates a key learning objective (see Appendices B and C, Q1). The procedure is the same as exercise 2, but the students add an extra one to the original input pattern (from exercise 1), instead of taking one away as they did in exercise 2. Note that this exercise will only work if the thresholding function (see Eqs. 3-4) is followed correctly.

(4) At this point in the exercises, the students should have learned the basic procedures required to experiment with pencil-and-paper neural networks. They are ready to work with larger networks (6x6) and able to teach the same network two different associations. When training their networks with a second association, they must be careful not to train the same input pattern with two different output patterns. They also must take care not to convert ones back to zeros during the second training, or the network will forget the first association. They should be able to show that their networks can represent two different associations by storing information in the form of strong and weak synapses. When prompted with the input from

the first association, they should get the first output from the pairing they used to train the network, and likewise with the second association.

(5) Hamming distance is a way of describing the distance between two binary vectors. It is calculated by lining two vectors up and counting the number of bits (ones and zeros) that are different between the two patterns. The Hamming distance is greatest between two orthogonal binary sequences (for which the dot product is zero). For example, the Hamming distance between (0,1,0,1,0,1) and (1,0,1,0,1,0) is 6. The Hamming distance is lowest for patterns that differ by only one bit. When a trained network has been presented with an input it has not seen before, it should treat that input as though it were the trained input pattern with the shortest Hamming distance to the novel pattern that it is seeing for the first time. If, however, the novel pattern is at an equal Hamming distance from two trained input patterns, it may treat the novel input as a hybridized version of the two trained inputs, and thus recall a nonsensical, hybridized output. Previously learned patterns can be conceived of as "attractors" or "basins of attraction", similar to well-worn ruts on a dirt bike trail; bike wheels tend to gravitate into these ruts, and once in one stay there. Similarly, networks in novel states of activity tend to fall into the nearest state of activation. Specifically, they tend to develop the state of activation with the lowest Hamming distance to the novel state. (Of course, if the bike is equally distant from two ruts in this analogy, one tire gets stuck in one rut, the other in the opposite rut, and the biker crashes down somewhere in the middle!)

(6-7) The load parameter α is a value that relates to the degree of saturation of the network. It is calculated by dividing the number of patterns stored by the number of neurons that make synapses in the network (the number of neurons in the input layer). Another way to measure saturation is to determine the fraction of synapses that can be strengthened before the networks begins making errors in recall. For the last two exercises, the students study the degree of saturation of the network at the point where it first begins to make mistakes. They should be encouraged to think about their own learning experiences while completing this exercise. For example, they may notice that less distinct patterns (with shorter Hamming distances) lead to mistakes at a lower level of saturation, which is one explanation of why it is hard to memorize large numbers of very similar stimuli (see Q3 in Appendices B and C). Also, they should note that these types of associative networks are quite characteristic of hippocampal connectivity. Because the hippocampus is necessary for short-term memory and has a limited load factor, we rely on our ability to consolidate memories into long-term representations stored elsewhere. This is one possible explanation of the effectiveness of paced learning.

Assessment of Learning: To assess learning due to this pencil-and-paper neural network activity, students were given two ten-minute writing exercises. The pre-exercise assessment (Appendix B) was conducted in class the day prior to the lab activity, immediately after a 10-minute introduction to the concept of the McCulloch-Pitts neuron model. The post- exercise assessment (Appendix C) was conducted in class on the day following the lab activity. Students' responses were digitalized for quantitative analysis of word counts and frequency of term use, as well as for qualitative assessment of learning. St. Olaf College IRB approval was obtained for all assessment included within this study.

RESULTS

The activity took most students 90 minutes to two hours to complete when working in groups. Each lab section had 16 to 18 students, one instructor and one teaching assistant. The students were diverse in their academic majors, with psychology (39%), biology (33%), and chemistry (15%) having the highest representation. Other majors represented included nursing, physics, religion, philosophy, exercise science, French, mathematics and an individualized major. 76% of the students were enrolled in the neuroscience concentration program. Their class years were as follows: 15% seniors, 48% juniors, 33% sophomores, and 3% first-years. There were 13 men and 20 women in the class. The students worked in groups of two to four to complete the activity. Although they worked in groups, each student performed the activity independently, with their own arbitrary input and output patterns applied to each of the exercises in Appendix A.

Research on artificial neural networks suggests that redundant synaptic connections permit a network to respond to a novel stimulus as it would to a closely related, familiar input. On the other hand, the ability to learn multiple patterns is greater when the patterns are distinctive than if they are similar due to overlapping pools of synaptic connections. It was our hope that by carrying out the activities described here (see Methods and Appendix A), students would develop a mechanistic understanding of how their brains may cope with noisy, incomplete and closely related information. The assessment writing activities were designed in parallel around three questions: (1) How do neural networks deal with noisy/fuzzy input? (2) How do neural networks deal with incomplete input (partial cue)? (3) How do neural networks deal with confusingly similar inputs (distinctiveness)? Analysis of answers to open-ended questions is difficult. As a start, we asked whether the students had more to say in response to the prompts after the lab activity (word count) and at what grade level (complexity) were their answers written from a readability analysis perspective. The results of the quantitative analysis of the responses to these questions are shown in Figure 5.

With respect to how neural networks deal with noisy data (Q1; see Appendices B and C), the students wrote the same length answers on the post-exercise assessment (42.5 ± 2.3 [standard error of the mean] words; n = 32) as they did on the pre-exercise assessment (40.7 ± 2.3 words; n = 31). Paired-samples t-tests were conducted to compare words per answer in the pre- and post-exercise answers to Q1. There was no difference between the pre- and post-scores for Q1 ($t_{30} = 0.56$). Readability statistics indicate how suitable a text is for a hypothetical audience. The Flesch-Kincaid grade levels are commonly used to predict the appropriateness of texts to readers of different

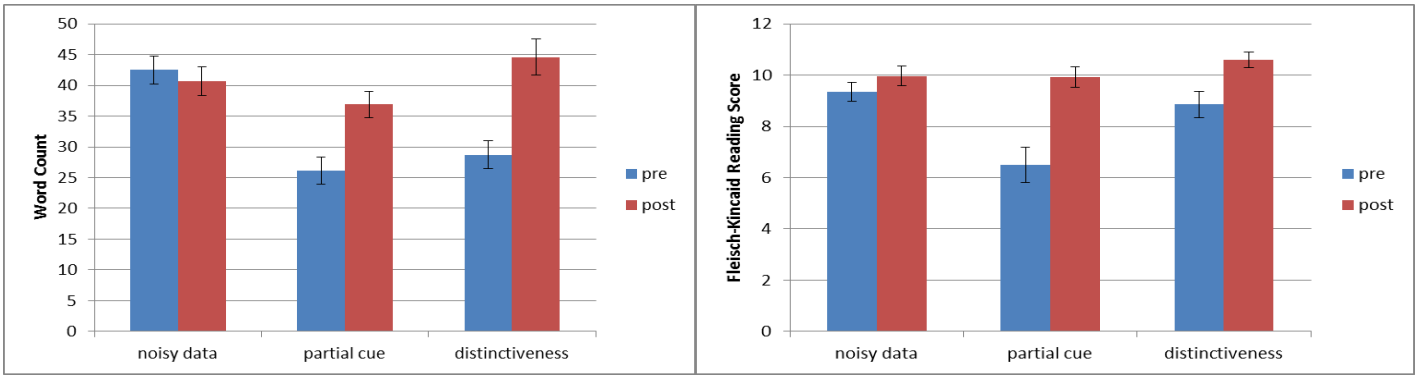


Figure 5. Comparison of students' written response length (left) and sophistication (right) from the three pre- and post-exercise assessment questions (see Appendices B and C).

experience levels, as well as to provide feedback to writers about the suitability of their work for the intended audience. Grade level is calculated by a formula that takes into account words per sentences and syllabi per word (Kincaid et al., 1975). Such methods have been shown to strongly correlate comprehension measured by standardized reading tests (DuBay, 2006). The complexity of the students' writing (sentence length and grammatical complexity) did not change (t_{30} = 1.24) after the lab activity in their answers to Q1 (from 9.34 ± 0.36 to 9.97 ± 0.39 on the Flesch-Kincaid reading scores as calculated by Microsoft Word; n = 31).

Qualitatively, their answers to Q1 on the pre-exercise assessment focused on phenomenological explanations: the most common answers they gave were that the brain figures things out from context (10 instances), and fills in missing information (7 instances). After the activity, answers that were based on filling in the gaps and using context decreased, and answers related to neural network mechanisms increased. The following neural network related terms increased in the post-exercise assessment: "filter" (from 0 to 5 instances), "input" (from 6 to 24 instances), "output" (from 0 to 16 instances), "recall" (from 0 to 6 instances), "train-" (from 0 to 7 instances), and "noise" (from 1 to 30 instances). Thus, while the total number of words used and complexity of answers in answers to Q1 did not change, the terminology used showed a shift to the inclusion of concepts relevant to neural networks.

Changes in student answers for Q2 and Q3 were less subtle. Student answers to Q2 increased in length from 26.1 ± 2.2 words (n = 31) to 36.9 ± 2.1 words (n = 32), while answers to Q3 increased from 28.7 ± 2.3 words (n = 31) to 44.6 ± 2.9 words (n = 32). Similarly, the sophistication (Flesch-Kincaid reading score) of the answers changed from 6.5 ± 0.7 (n = 31) to 9.93 ± 0.4 (n=32) on Q2, and 8.85 ± 0.5 (n = 31) to 10.6 ± 0.3 (n = 32) on Q3. These differences in the changes of individual students' response lengths concerning partial cue (Q2; t_{30} = 3.35; $p < 0.01$) and distinctiveness (Q3; t_{30} = 6.03; $p < 0.001$) were statistically significant. Paired-samples t-tests comparing Flesch-Kincaid reading scores also revealed significant difference in the changes of individual students' response lengths concerning partial cue (t_{30} = 3.71; $p < 0.01$) and distinctiveness (t_{30} = 3.24; $p < 0.01$).

Terms relating to neural networks also increased in post-exercise assessment responses to Q2 compared to the pre-exercise assessment, including "network" (from 1 to 13 instances), "incomplete" (from 0 to 17 instances), "input" (from 0 to 20 instances), and "output" (from 1 to 17 instances). Neural network related terms that were more prevalent in the post-exercise assessment responses to Q3 included "network" (from 1 to 8 instances), "input" (from 0 to 12 instances), "output" (from 1 to11 instances), "pattern" (from 11 to17 instances), and "saturat-" (from 0 to 7 instances). Note that the stem "saturat-" was counted because it appeared in several forms (e.g., saturate, saturation, etc.). We did this with other stems, too, such as "neu-" and "nerv-."

Across all three questions, several terms of relevance to neural networks increased in frequency of use from the pre-exercise assessments to the post-exercise assessments (see Fig. 6). The term "brain" increased from 42 to 68 instances, "input" from 6 to 55 instances, "output" from 2 to 41 instances, "network" from 5 to 29 instances and "pattern" from 15 to 31 instances. Words with the stems "neu-" and "nerv-" increased from 6 to 28 instances.

A few examples of student answers before and after the lab exercises are shown in Table 1. Our impression from reading the raw answers that were turned in during the assessment exercises were that the students were learning to think about mechanisms behind these phenomena. We feel that our quantitative analysis, although simple, largely support our qualitative impressions.

DISCUSSION

Since we first introduced these exercises five years ago, they have been used for in-class and in-lab exercises at St. Olaf College in the Cellular and Molecular Neuroscience class, the Bio-Math seminar, an upper division seminar called "The Neuron," and the Science Conversation seminar series. Because these exercises appeared to be useful in helping the students to comprehend high-level concepts such as distributed representation of memories and the role of synaptic plasticity in memory formation, we endeavored to take a more rigorous approach to assessing

Pre-Lab Sample Answers	Post-Lab Sample Answers
"Brain[s] can recognize words/patterns better because they can also infer meaning from context" (Q1)	"Even with noise in the image, there is enough similarity within the patterns, causing the closest matching pattern to 'fire.'" (Q1)
"Characteristic patterns and cues… connections to previous examples…" (Q2)	"Our neural networks yesterday showed that we can have an omission in the input and still have the same output due to the connections made by other associations in the synapses." (Q2)
"They are relatively similar… Nerves that fire based on similar patterns are all going at once." (Q3)	"Because the stimuli are so similar the Hamming distance must be very small and it's easy to get one pattern instead of another. The similarity between patterns… makes it very easy for the brain to become confused and have trouble learning…" (Q3)

Table 1. Comparison of students' written responses to assessment questions before and after the lab activity. Pre- and post-lab sample answers are provided from the same student, with a different student represented for each pair.

specific learning outcomes. Quantifying learning gains can be challenging, especially in interdisciplinary fields where a fundamental outcome is that the students gain an appreciation of and an ability to use content and practices from multiple disciplines to solve a problem (Crisp and Muir, 2012).

We settled on the following learning outcomes: after the activity, students should be able to use terms and concepts from neural network theory to discuss (1) how the brain extracts meaningful information from noisy inputs, (2) how the brain uses prior learning to complete information when presented with a partial cue, and (3) why the brain struggles to form correct associations among stimuli that are highly similar (low distinctiveness).

Thus, the pre- and post-exercise assessment activities were designed in parallel, such that the students wrote to each learning outcome both before and after the activity, although the specific examples presented differed in the pre- and post-exercise assessments. One problem with the assessment questions was that several students were not able to figure out what the image was in question 2 (Q2) of the pre-exercise assessment likely due to a decreasing familiarity with soda cans (see Appendix B). Because of this, it is not clear whether they had more to say about the answer to Q2 after the assessment, or if they did not understand the pre-exercise assessment Q2 sufficiently to say much about it. However, frequency of use of neural network-related terms in post-exercise assessment Q2 still suggests a better understanding of the concept of an incomplete input. Gains in sentence length and response sophistication were also seen in Q3 after assessment, and the students seemed to have no difficulty in recognizing or understanding the examples used for Q3.

Little difference in how much they wrote or in the sophistication of their writing was observed after the activity with respect to how the brain deals with noisy data. However, the content analysis revealed increases in the frequency of use of terms related to neural networks (such as "input," "output," "training," "recall" and "noise) after the activity compared to student responses during the pre-assessment. This change in term use was present across all three assessment questions. Overall, the change in language use seemed to suggest a transition from describing the phenomena illustrated by the example to explanations of a more mechanistic nature. For example, one student wrote before the pencil-and-paper neural network activity in response to Q2 about an incomplete input: "We recognize different aspects of the picture despite changes in lighting/shading and connect it to something we have seen before. The brain likes to find patterns." After the lab activity, the same student wrote in response to how the brain can recognize a phrase with letters missing: "It's easy to fill in the missing letters because this represents an incomplete input (stimulus)

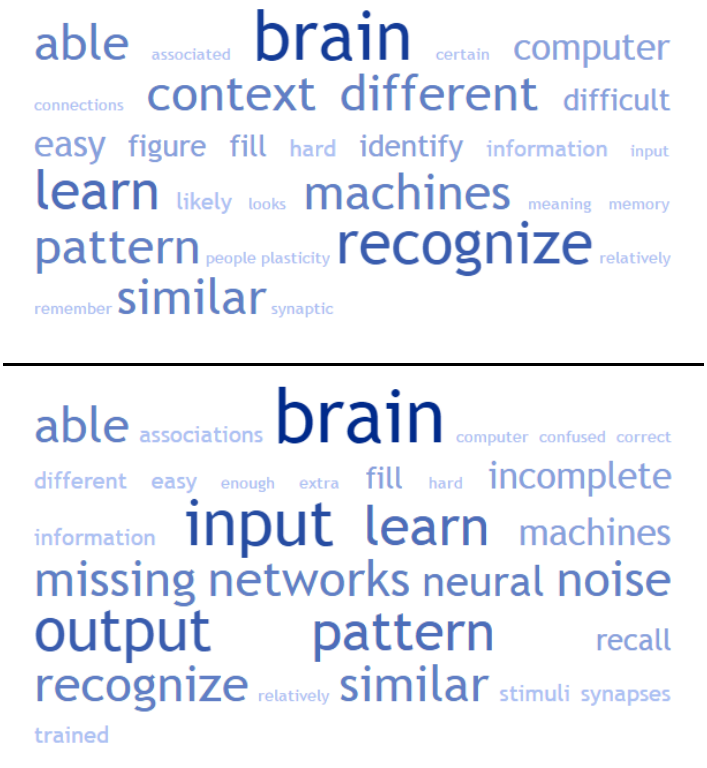


Figure 6. Word clouds of the students' responses to the assessment questions. Pre- (top) and post-exercise assessment (bottom) responses were pooled across the three questions. Common words specific to the stimuli (e.g., "Morse code", "Molecule") were omitted. Images generated using TagCrowd.com.

and, given an already strengthened synapse, the correct output (complete phrase) can be recalled even though the input is incomplete."

It was the general impression of instructor and the teaching assistants that the activity was well received. One student wrote a note on their assessment piece that read "Continue to do this lab in the future. It is quick, simple, fun and informative." Some students disliked the exercise because it was a "dry lab" experience. They preferred labs that involved microscopy and electrophysiology (e.g., a later lab where we recorded action potentials intracellularly from leech ganglia). Other students seem to connect very well with the pencil-and-paper neural network exercises. One group of students from the prior year used the model as the basis for an independent project where they tried to produce a nerve net capable of playing tic-tac-toe. Another group from the assessed cohort adopted the model as the basis of an independent project in a software design course.

The students struggled with concepts such as Hamming distance and load parameter and how these related to neural network function. They also tended to make errors at first when training a network with multiple patterns in which they changed ones to zeros and seemed puzzled when the network failed to recall what it had previously learned. Some did not understand the training and testing states, and changed synaptic weights during testing with novel patterns. A few students were troubled by the thresholding function and what it meant biologically for the threshold to be different when the network was presented with an incomplete pattern. The explanation of this is non-trivial. In a winner-take-all network, the output units with the highest dendritic sums become active. This is perhaps the simplest way of setting a threshold. However, we are recommending here a variant of the Wilshaw strategy, in which the threshold is equal to the number of ones in the input pattern. The Wilshaw strategy has been shown to improve storage capacity and reduce error rates compared to other winner-take-all networks. (Wilshaw et al., 1969; Graham and Wilshaw, 1994). In our variant, it is equal to the lesser of the number of ones in the test pattern or the number of ones in the original input pattern. This is still a fairly simple rule to execute, but affords networks that are more forgiving of ambiguous inputs. In conversations with students about the biological meaning of threshold, they seemed to make the assumption that the threshold must be exactly equal to the net dendritic sum, when in reality many synapses employ a safety factor to ensure signaling fidelity. For example, far more neurotransmitter is released at the neuromuscular junction than is required to bring the sarcolemma to threshold.

We have not yet tried using software to augment the lab experience. When the students do the activities by hand, they seem to develop a clear concept of how the model works. Using a simulator of these types of networks might obscure the inner workings. However, when used alongside the pencil-and-paper neural network exercises, simulation software may allow the students to work with larger, more complex patterns, and might make phenomena such as noise reduction and pattern completion easier to interpret and more exciting to observe.

Simple artificial neural networks have a great deal of potential as teaching tools to help students build connections between cellular and network level phenomena and higher-order processes like thought and behavior. We have used variations of such networks to construct oscillatory networks in simulations of rhythmic animal behaviors, to demonstrate lateral inhibition in visual-processing systems such as the *Limulus* eye, and to build artificial intelligence systems capable of playing Tic-Tac-Toe. We hope that the demonstrations herein described inspire faculty to incorporate artificial neural networks into their teaching in new and exciting ways.

REFERENCES

Crisp KM (2012) A structured-inquiry approach to teaching neurophysiology using computer simulation. J Undergrad Neurosci Educ 11:A132-138.

Crisp KM, Muir GM (2012) Assessing development of an interdisciplinary perspective in an undergraduate neuroscience course. J Undergrad Neurosci Educ 10:A88-A95.

DuBay WH (2006) Smart language: readers, readability and the grading of text. Costa Mesa, CA: Impact Information.

Graham DJ (1987) An associative retrieval model of arithmetic memory: how children learn to multiply. In: Cognitive processes in mathematics (Sloboda JA, Rogers D, eds). Oxford: Oxford University Press.

Graham D, Wilshaw D (1994) Capacity and information efficiency of a brain-like associative net. In: Advances in neural information processing systems 7 (Tesauro G, Touretzky DS, Leen TK, eds). NIPS Proceedings.

Hopfield JJ (1982) Neural networks and physical systems with emergent collective computational abilities. PNAS 79:2554-2558.

Kincaid JP, Fishburne RP, Rogers RL, Chissom BS (1975) Derivation of new readability formulas (automated readability index, fog count, and flesch reading ease formula) for Navy enlisted personnel. Research Branch Report 8-75. Chief of Naval Technical Training: Naval Air Station Memphis.

McCulloch WS, Pitts WH (1943) A logical calculus of the ideas immanent in nervous activities. Bull Math Biophys 5:115-133.

McNaughton BL, Morris RGM (1987) Hippocampal synaptic enhancement and information storage within a distributed memory system. TINS 10:408-415.

Viscuso SR, Anderson JA, Spoehr KT (1989) Representing simple arithmetic in neural networks. In: Advances in cogntivie science: theory and applications (Tiberghien, ed). Cambridge: Horwoods.

Wilshaw DJ, Buneman OP, Longuet-Higgins HC (1969) Non-holographic associative memory. Nature 2222:960-962.

Received May 11, 2015; revised July 03, 2015; accepted August 07, 2015.

The authors thank the St. Olaf College Neuroscience Faculty for guidance and support in the development and execution of this project, especially Professors Steve McKelvey, Anne Walter, Shelly Dickinson and James Demas.

Address correspondence to: Dr. Kevin Crisp, Biology Department, St. Olaf College, 1520 St. Olaf Avenue, Northfield, MN 55024.
Email: crisp@stolaf.edu

APPENDIX A. List of in-lab exercises.

1. Teach a 16-synapse network (with 4 input neurons and 4 output neurons) to pair an input pattern with an output pattern. Then test the network with the same input pattern. Does it recall the correct output pattern?
2. Using the trained network you generated in (1), test the network with an incomplete version of the input pattern you trained it with. Is it still capable of recalling the correct output pattern, even with an incomplete "cue"?
3. Test the trained network you generated in (1) with a noisy version of the original training input pattern, in which at least one of the zeros in the original pattern is now a one. Can it recall the correct output pattern even when there is interference in the input cue?
4. Teach a 36-synapse network (with 6 input units and 6 output neurons) to associate an input pattern with an output pattern. Then, teach it a second association between a new input pattern and a new output pattern, being careful only to strengthen and never to weaken synapses. Test this network with both input patterns. Can it faithfully recall the correct output patterns?
5. Test the network with a distorted version of one of the two training patterns (e.g., a version with an extra one or an extra zero). Does it recall the correct output pattern? Calculate the Hamming distance (number of different bits) between the novel input and the two trained inputs. Does it treat the novel pattern as though it were the learned pattern with the smallest or largest Hamming distance from the novel pattern? What happens if you give it a novel input pattern which is the *same* Hamming distance from both of the learned patterns?
6. How many paired associations can you teach the network you developed in (4) before it starts making mistakes in recall? What fraction of the synapses are strengthened when it starts making mistakes? What's the load parameter (α) of the network? (The load parameter is calculated by dividing the number of stored patterns by the number of neurons in the input layer.)
7. Calculate the maximum load parameter for the network you developed in (1). Do the two load parameters differ? What does this suggest about the information-storing capacity of the two networks?

APPENDIX B. Pre-Lab Assessment

Using what you already may know about neural networks, write to each of the following:

Original Text:

State which so nobly has borne herself thus far in the conflict.

At the close of Col. Wooster's address, the elegant flag was handed to the tall color sergeant, who took his position in the line with evident pride and resolute purpose. The men of the Twenty-ninth, many of them well known and worthy citizens of New Haven, seemed thoroughly to appreciate the high and sacred emblem which is entrusted to their keeping, and will do all that men can do to achieve new renown beneath its inspiring folds. Thus ended the presentation.

After optical character recognition:

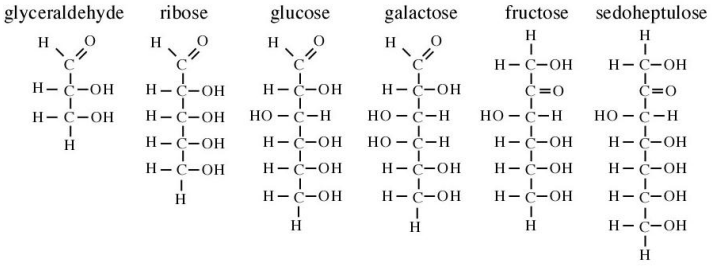
Stiitc wliii,li so nobly lia? borne herself tluis far in the coiitlii't.

At tiie elose of Col. "NVooster's a.l(lre?», tlie elegant flip was hanileil to the tall eolor sergeant, who took his po-itioii in tlio line with evident pride and resolute juirposc. The men of the Twenty-ninth, many of tln ni well known and worthy citizens of New Haven, seemed thorouchly to appreciate the hii;h nnd sacred emblem wliich is entrnsted to their keeping, and will do all that men can do to achieve new renown beneath its insjiiring folds. Thii.s ended the presentation.

Q1: Why is it relatively easy for the brain and harder for machines to recognize scanned text?

(http://floorsix.blogspot.com/)

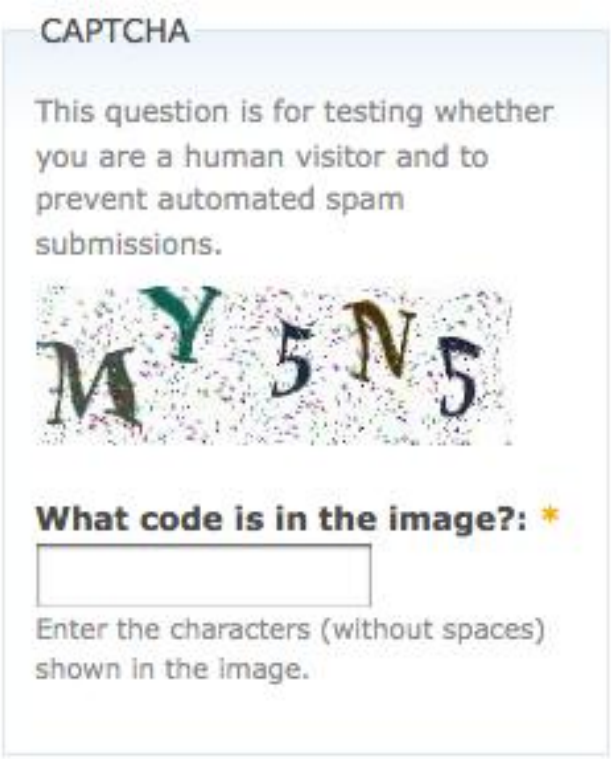
Q2: Why is it relatively easy to figure out what this is a picture of?



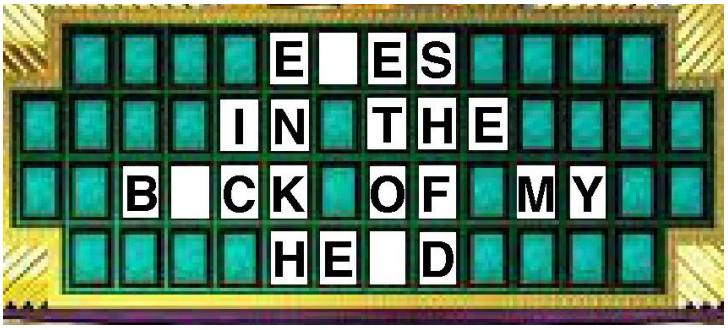
Q3: Why is it relatively hard to learn the names of organic molecules?

APPENDIX C. Post-Lab Assessment

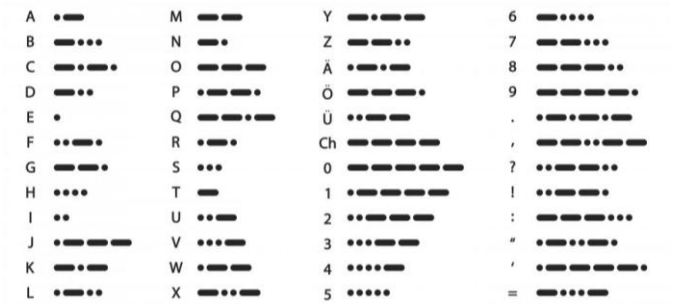
Using what you learned about neural networks in yesterday's lab, write to each of the following:



Q1: Why is it relatively easy for the brain and harder for machines to recognize CAPTCHAs?



Q2: Why is it relatively easy to fill in the missing letters?



Q3: Why is it relatively hard to learn the international Morse code?