Learning to Think Like A Neuron in Middle School

David Touretzky[1], Christina Gardner-McCune[2], Will Hanna[3], Angela Chen[1], and Neel Pawar[1]

[1]Carnegie Mellon University, Pittsburgh, PA 15213
[2]University of Florida, Gainesville, FL 32611
[3]Thomas County Middle School, Thomasville, GA 31792
dst@cs.cmu.edu, gmccune@ufl.edu, whanna@tcjackets.net, angelac3@andrew.cmu.edu, neelpawar17@gmail.com

Abstract

Neuron Sandbox is a browser-based tool that helps middle school students grasp basic principles of neural computation. It simulates a linear threshold unit applied to binary decision problems, which students solve by adjusting the unit's threshold and/or weights. Although Neuron Sandbox provides extensive visualization aids, solving these problems is challenging for students who have not yet been exposed to algebra. We collected survey, video, and worksheet data from 21 seventh grade students in two sections of an AI elective, taught by the same teacher, that used Neuron Sandbox. We present a scaffolding strategy that proved effective at guiding these students to achieve mastery of these problems. While the amount of scaffolding required was more than we originally anticipated, by the end of the exercise students understood the computation that linear threshold units perform and were able to generalize their understanding of the worksheet's "solve for threshold" strategy to also solve for weights.

Website — https://www.cs.cmu.edu/~dst/NeuronSandbox

Introduction

Since neural networks are one of the key technologies powering the current AI revolution, it is sensible to introduce K-12 students to the basics of neural computation. In the case of middle school students we can use the linear threshold unit as our model of computation and explore how such units are used to solve binary decision problems. Although linear threshold units aren't the ReLUs or GELUs found in modern neural nets, they are the pedagogical equivalent of the Bohr model of the atom: historically important and an age-appropriate simplification (Touretzky, Chen, and Pawar 2024).

Learning to think like a neuron is challenging for middle school students because they have not yet taken algebra and are just beginning to develop mathematical problem solving skills. We created a browser-based tool called Neuron Sandbox (Chen, Pawar, and Touretzky 2024) that allows students to experiment with linear threshold units while solving a series of problems of increasing difficulty. We also created a detailed worksheet to scaffold student learning of a particular problem solving strategy. We used these tools with a group of 21 seventh grade students to investigate the following research questions:

RQ1 (Performance): To what extent are students able to solve for a threshold and/or solve for weights in binary decision problems?

RQ2 (Problem Solving): How are students solving these problems, and what do they know afterward about how a neuron computes?

RQ3 (Prerequisites): What knowledge and skills do students need to solve these problems?

In this paper we describe a tool that makes visually explicit how a single linear threshold neuron works. We explore how to support middle school students learning this content and the challenges they face in understanding the material. Our classroom observations provide insight into the skills students acquire as they learn to solve linear threshold unit problems.

Prior Work

The growth of K-12 AI education efforts in recent years has produced a wide variety of resources for introducing children to neural computation. Neural Networks for Babies (Ferrie and Kaiser 2019) is a picture book for toddlers that introduces the concepts of neurons, inputs, outputs, networks, and hidden units, but does not discuss weights. Code.org's video How Neural Networks Work gives an impression of how supervised learning can adjust a neuron's weights, but doesn't actually explain the operation of the neuron (Code.org 2020). The PBS video series Crash Course AI includes several videos on neural networks, the first of which discusses the history of the perceptron and explains its operation (Crash Course 2019). Numerous other introductions to neural networks can be found on YouTube that cover similar ground. There are also several unplugged activities in which students simulate the operation of small networks of neurons, as reviewed in (Touretzky, Chen, and Pawar 2024). None of these resources examine in depth how weight and threshold values influence the decision-making behavior of a linear threshold unit, or how to adjust these values to achieve a desired result. At the advanced high school level, Google's TensorFlow Playground (Sato 2016) is an excellent browser-based tool for exploring backpropagation learning in multilayer perceptrons, but it is too complex for middle school students. While tools and resources are avail-

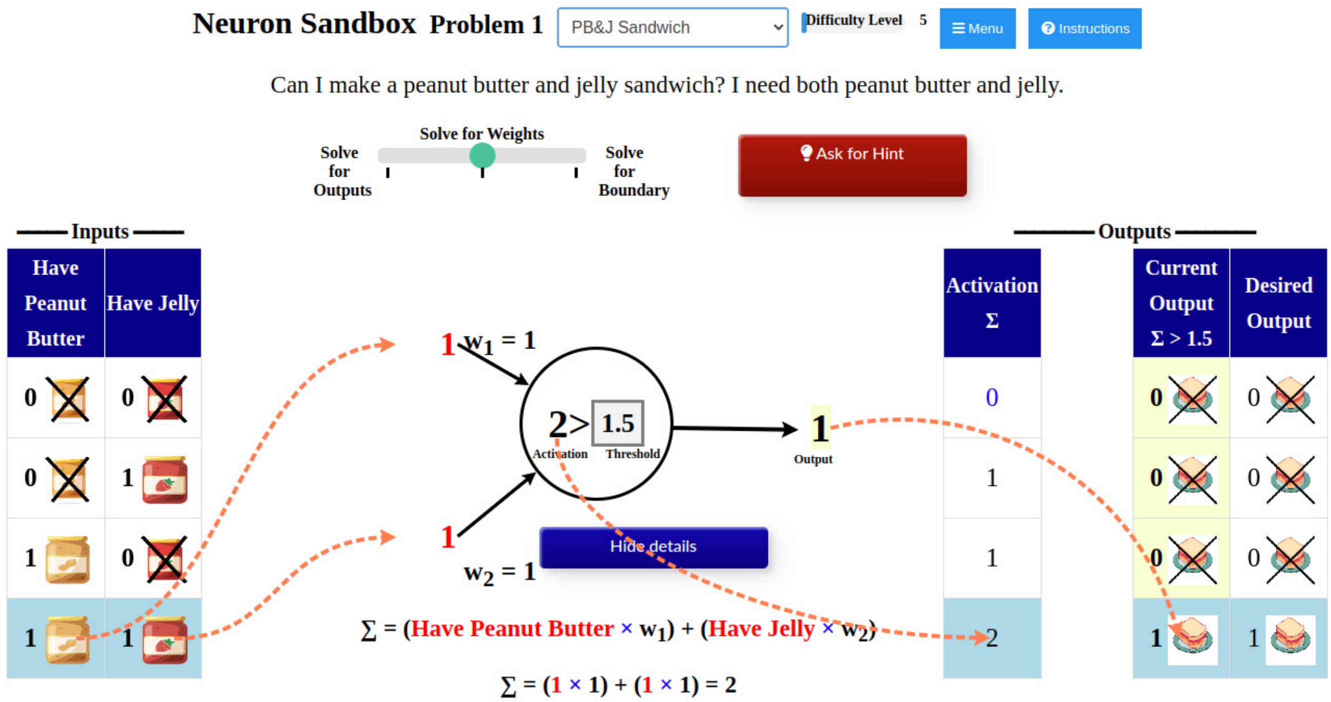


Figure 1: Neuron Sandbox display of the PB&J problem in "Solve for Weights" mode.

able to introduce students to neural networks, the majority of research on K-12 AI teaching and learning is focused on how to teach students to train neural network models (Tedre et al. 2021; Shamir and Levin 2021). There is a growing body of empirical research on student learning of machine learning concepts, but a lack of evidence of students understanding in depth how a single neuron works. In this paper we present a tool accessible to middle school students that helps them learn to reason about single-neuron computation.

Neuron Sandbox

Neuron Sandbox (Figure 1) is a browser-based application that supports experimentation and problem solving with linear threshold units. It has a built-in set of 17 problems of increasing difficulty, and an editor for creating new problems. When students solve a problem they can choose to advance to the next one or remain on the current problem for further experimentation. The first problem students are asked to solve is to make the neuron correctly answer the question "Can I make a peanut butter and jelly sandwich? I need both peanut butter and jelly" for all four cases.

Neuron Sandbox has three operating modes, the first of which, "Solve for Outputs", allows students to practice translating textual problem descriptions to truth tables. This mode was only used once in the study described here because the teacher usually modeled this skill in class as a group exercise instead.

The second mode, "Solve for Weights", presents a graphical depiction of the neuron with its weighted inputs and threshold. See Figure 1. To the left is a table of all the input cases. To the right is a corresponding table showing the neuron's activation value, its current output for that input (1 if the activation is above threshold, else 0), and the desired output. Any row where the current output doesn't match the desired output is highlighted in red. Thus it is immediately apparent whether the current weights and threshold constitute a solution to the problem. What's less evident is how to adjust these parameters when some inputs are not handled correctly. This is the primary skill we want students to learn.

Neuron Sandbox employs a variety of visual scaffolding techniques to help students grasp what is going on with the linear threshold unit. On the input side, icons appear beside each 0 or 1 to reinforce the meaning, e.g., for the PB&J problem we have an icon for a jar of peanut butter and one for a jar of jelly. In those cases where an input is zero, the jar is crossed out. Icons are also used on the output side for both the current and the desired output values, e.g., a sandwich on a plate if the value is 1, or a crossed-out sandwich if the value is zero.

To reinforce the importance of a number's sign, positive weights, thresholds, and activations are shown in black, while negative values are shown in red. A zero value is shown in blue. The arrows denoting the input connections to the neuron use the same colors as their respective weights. In addition, the magnitude of each weight is reflected in the thickness of the arrow.

To help students relate the tabular information on the left and right sides of the display to the graphical neuron in the middle, when they hover their mouse over a row of the table, the system draws leader lines connecting the cells of the input row to the input connections of the neuron, and additional leader lines connecting the neuron's activation and output values to the corresponding cells of the output row. The flow of information through the neuron is unmistakable. But we found this extensive visual scaffolding isn't enough for students to develop the skills required to solve these problems. Hence we developed the worksheet described later in this paper.

The third mode of Neuron Sandbox, "Solve for Boundary", provides for graphical display and manipulation of the neuron's decision boundary. It is intended for use at the high school level and was not available to students in this study.

Solve for Threshold Worksheets

We developed a worksheet template to scaffold students' problem solving process for "solve for threshold" problems. The worksheets used problems from Neuron Sandbox to ensure that students could use the tool to check their work when they were finished. The first worksheet, shown in Figure 2, presents Neuron Sandbox problem 1: Can I make a peanut butter and jelly sandwich? This problem requires students to solve for the threshold given the weights. Note that the column layout of the worksheet matches the layout of the Neuron Sandbox display, with inputs on the far left and desired outputs on the far right.

Our goal in designing the worksheet was to make explicit every step needed to solve a particular kind of linear threshold unit problem. We hoped that students would transfer that skill back to their use of Neuron Sandbox in order to solve other problems in the collection. The worksheet provides several types of assistance:

Process scaffolding:

- Column labels A-F guide students to perform the steps in the correct order.
- Column A, "desired output", provides students with a starting point for problem solving by prompting them to indicate the desired output for each case with a 0 (No) or 1 (Yes).
- Column F, "Is Activation greater than threshold?" encourages students to check their work. If correctly solved, columns A and F should match.

Computational scaffolding: Breaks down the computation into the simplest possible steps, requiring only a single multiplication, addition. or comparison.

- Column B, "Compute weighted inputs", has two subcolumns: one for each input and its corresponding weight.
- Column C, "Activation", has students sum the weighted inputs of column B.
- Column F, "Activation $>$ Threshold" prompts students to evaluate the greater than inequality ($>$) for each input case.

Problem Solving Strategy:

- Column D, "Do we want the activation to be greater than the threshold?" prompts students to think of the desired output as setting a constraint on the threshold, implicitly referring them to column A.
- Column E taught an explicit strategy for determining the threshold: "What decimal value is greater than your Ns (Nos) but less than your Ys (Yeses)?"

The worksheet also provides students with vocabulary to talk about the problems and their problem solving strategies.

Methods

Study Context

This study is part of a larger design and development research project (Touretzky et al. 2023; Gelder et al. in press) that co-designed a 9-week AI elective covering topics such as AI in self-driving cars and autonomous robots, how computers understand language, and machine learning (i.e., decision trees and neural networks). Teachers were recruited to assist in the co-design of the curriculum and gained experience rapidly adapting and creating materials to better help students understand as they were teaching. As such, lesson facilitation and materials used in the first section of students in the morning were adjusted for the second section in the afternoon. Similarly, weekly lesson plans were adapted as teachers evaluated the effectiveness of the materials day by day and made refinements to better address student needs and ensure student learning and comprehension of material.

Participants

Participant Recruitment: Students were recruited, consented, and assented as part of the overall research project. The study was approved as exempt in accordance with the IRB at our institution. All student names have been anonymized.

Participant Demographics: All students who participated were middle school students from a rural district in the Southeastern United States. We have demographic data from 18 of 21 students: 11 identified as female, 6 as male, and 1 did not indicate a gender. 16 students reported prior computing experience in the pre-survey. In-school formal computing courses: 12% had no prior computing courses, 25% had 1 prior computing course, and 63% had 2 or more prior computing courses. Informal coding activities: 56% indicated they participated in informal coding activities. Familiarity with coding environments included: 44% each for Scratch, App Inventor, and "other"; 31% each for Minecraft and VEX/LEGO, 12.5% for Hour of Code. We received one response for each of the following tools: Game Maker/Salad, Alice, Arduino, Lily pad, Raspberry Pi, Udoo.

Study Design

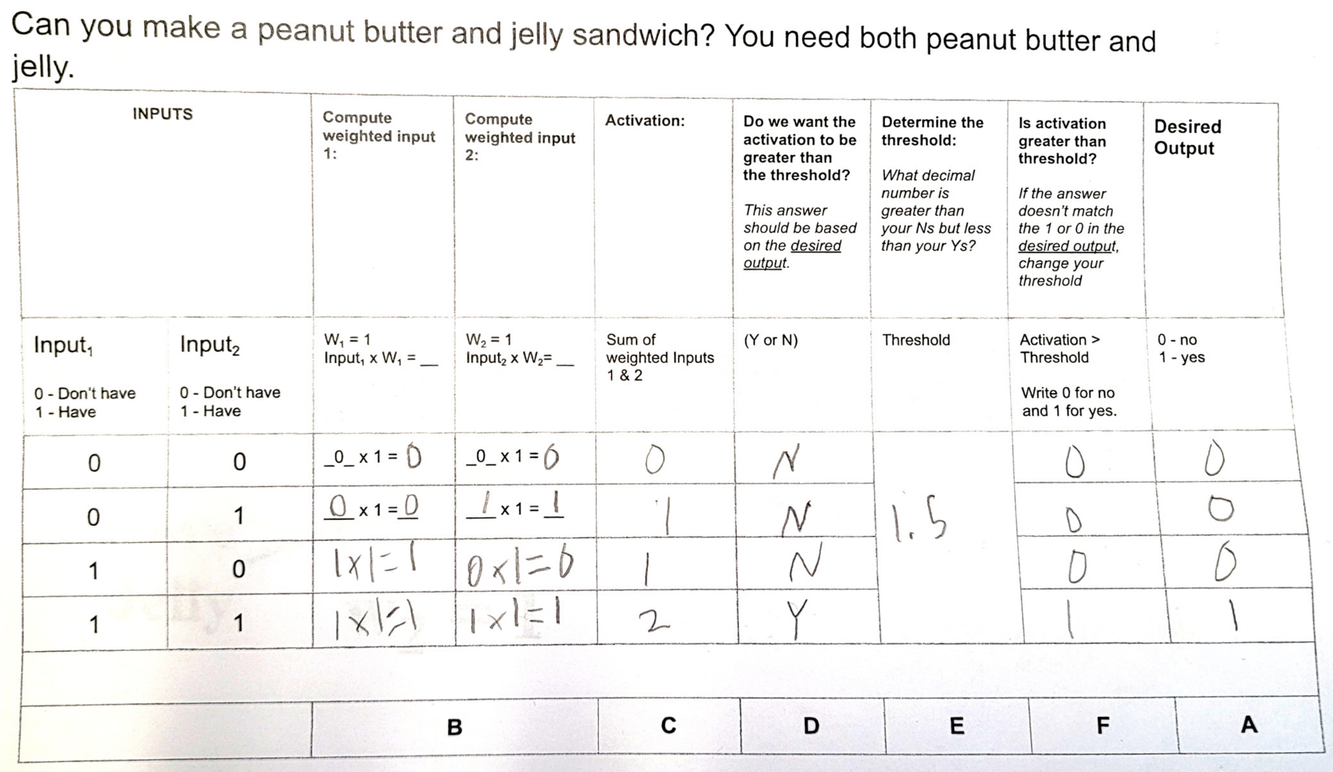
We conducted a two-day ethnographic study of one teacher teaching Neuron Sandbox to two course sections. This qualitative study explores three research questions:

RQ1: To what extent are students able to solve for threshold and solve for weights in linear threshold unit problems?

RQ2: How do students solve these problems and what do they know after solving them about how neurons compute?

RQ3: What knowledge and skills do students need to solve these kinds of problems?

Course Context. The lesson which is the focus of this paper takes place toward the end of the machine learning (ML) unit



Can you make a peanut butter and jelly sandwich? You need both peanut butter and jelly.

INPUTS		Compute weighted input 1:	Compute weighted input 2:	Activation:	**Do we want the activation to be greater than the threshold?** *This answer should be based on the desired output.*	**Determine the threshold:** *What decimal number is greater than your Ns but less than your Ys?*	**Is activation greater than threshold?** *If the answer doesn't match the 1 or 0 in the desired output, change your threshold*	**Desired Output**
$Input_1$ 0 - Don't have 1 - Have	$Input_2$ 0 - Don't have 1 - Have	$W_1 = 1$ $Input_1 \times W_1 =$ __	$W_2 = 1$ $Input_2 \times W_2 =$ __	Sum of weighted Inputs 1 & 2	(Y or N)	Threshold	Activation > Threshold Write 0 for no and 1 for yes.	0 - no 1 - yes
0	0	_0_ x 1 = 0	_0_ x 1 = 0	0	N		0	0
0	1	0 x 1 = 0	1 x 1 = 1	1	N	1.5	0	0
1	0	1x1=1	0x1=0	1	N		0	0
1	1	1x1=1	1x1=1	2	Y		1	1
		B		C	D	E	F	A

Figure 2: Student-completed worksheet for the PB&J problem.

and the 9-week curriculum. In the ML unit, students learned about different types of reasoners, decision tree learning, and neural networks (in this order). In the session prior to engaging with Neuron Sandbox, students had just started the neural network module and engaged in an interactive activity that models how neural networks work and how a neuron learns/adjusts its weights during training.

Observation. The two-day observation focused on how the teacher scaffolded his middle school students' learning how to solve binary decision problems with a linear threshold unit. To ensure authenticity of student learning, the observation was scheduled to coincide with the teacher's schedule for introducing neural networks to his students. The 2nd author (researcher) observed the 3rd author (teacher) while he was teaching. Initially this observation was scheduled for a single day. However, after discussing the difficulties students exhibited in grasping the concepts, the 2nd and 3rd authors decided that the lesson needed to be retaught with additional scaffolding, so they extended the observation and data collection to a second day.

Class Overview

Day 1: Brief introduction to Neuron Sandbox; the class solved the first problem (PB&J sandwich) together, then students worked on their own to solve the remaining 16 problems using the tool.

Day 2: Brief introduction to the newly-created worksheet: the class solved the peanut butter and jelly problem together using the worksheet. They then solved two more worksheet problems on their own. Those who completed these problems were offered an optional "bonus" problem of a different form ("solve for weight" instead of "solve for threshold") with a modified worksheet. Once students completed their worksheets, they were asked to try to solve the remaining Neuron Sandbox problems again using the online tool, while the investigator observed.

Data Collection

We collected both fieldnotes and videos of the instruction and student problem-solving. The fieldnotes included author 2's observations, the motivation and process of creating the student worksheet, and discussions with the teacher about how each of the class sessions went. We also collected student artifacts (worksheets) and pre and post course surveys. The student artifact was a six-page packet with three required problems (sample page in Figure 2). An additional bonus problem on a separate sheet was available for students to take when they completed the packet. Students used pencils to record their answers on the worksheets. As such, partially erased answers and overwritten values were sometimes visible upon inspection of the worksheet. The pre-post survey contained multiple choice, Likert scale, and open-ended response questions designed to obtain students' general perception of AI, the AI course, demographics, and prior computing experience.

	RQ1	RQ2	RQ3
Data Triangulation	Worksheets Observations Student interviews Member checking	Worksheets Observations Student interviews Member checking	Mistakes on worksheets Fieldnotes Observations Member checking
Codes	Types of mistakes Types of problems	Recognition of relationship between inputs and weights Recognition of relationship between activation and threshold Referencing worksheet in problem-solving Teacher modeling Discussion with peers	Reading and understanding the problem Mathematical knowledge Problem-solving skills
Themes	Performance on Problems	Explanations & Understanding of A Neuron's Computational Processes	Prerequisite Knowledge & Skills

Table 1: Codes and themes from data analysis.

Data Analysis

Our goal in analyzing the data was to gain a holistic understanding of how students learned to solve linear threshold unit problems. Our data analysis for this paper includes summary of participant demographics from the survey; evaluation of artifact correctness; and thematic analysis of artifacts, video, and field note data (Clarke and Braun 2014). We identified 10 high-level codes and grouped them into three main themes across our data sources: students' (1) performance on problems, (2) explanations and understanding of a neuron's computational processes, and (3) prerequisite knowledge and skills (see Table 1). We triangulated emerging themes from the analysis across artifacts, fieldnotes and videos (Thurmond 2001). We used member checking to validate the findings (Birt et al. 2016). The 2nd author presented the themes identified in the data to the 3rd author to review and comment on. Overall, the 3rd author agreed with the findings and provided clarity on student omissions, errors, and student challenges he observed, and the kinds of help he provided across the two days of instruction.

Findings

RQ1: Student Performance

After the introduction of the scaffolded worksheet, we found that 19 out of 21 students (90.5%) were able to correctly solve for the threshold in basic AND and OR problems (#1 & #3) on the worksheets. We also found that the majority of students (71%, n=15) correctly solved threshold problem #6 that had higher weights (weight = 2). In general, we observed that as the complexity of the problems increased, less students attempted the problem and more answered it incorrectly. For example, in the bonus problem that asked students to solve for weights (#4), we found that only 15 students attempted the problem and 86.6% (n=13) correctly solved for the missing weight. Our observational data suggest that students' understanding was still developing and they needed more time to solve the problems, resulting in decreased attempts at subsequent problems. The results for the four worksheet problems are summarized in Table 2.

Overall we observed that using the worksheet helped students solve for thresholds, and some could also use it to solve for weights even though, technically, a different worksheet is needed for that type of problem. When interviewed, students sometimes referred to the worksheet to scaffold their explanation of how they solved the problem and why their answers were correct. In a video recording of a pair of boys solving worksheet problem #4 (Table 3), one shared that prior to the introduction of the worksheets, he was just guessing — a sentiment shared by many of the students. But after the worksheets he understood the relationship between the inputs and weights and the activation and the threshold. While solving worksheet problem #4, he demonstrated his understanding of the relationships by proposing additional weights that would correctly solve the problem given the current threshold, and used the worksheet's reasoning steps to prove that his answers were correct.

After students completed the worksheets they went on to solve additional problems using just the software. In the interviews of students solving problems in Neuron Sandbox, we observed them correctly solving more complex "solve for threshold and weight" problems featuring 3-4 inputs, irrelevant inputs, negated inputs, and negative weights. Table 3 describes the types of problems that students solved in video-recorded interviews. Overall, we observed an increase in students' confidence, excitement, and ability to correctly solve and explain their answers.

RQ2: Student Problem Solving

In "solve for threshold" problems it is important for students to recognize three key relationships among the inputs, weights, weighted inputs, activation, threshold, and desired output. Recognizing these relationships is what allows them to move from merely following the worksheet to reasoning independently and taking on a wider variety of problems. The relationships are:

#	Name	Func.	Solve for	Notes	Student Performance
1	PB&J	AND	Threshold	Weights 1; threshold ∼1.5	20 correct; 1 incorrect
3	Wear boots	OR	Threshold	Weights 1; threshold ∼0.5	19 correct; 2 blank
6	Picnic	AND	Threshold	Higher weights (2 and 2) require higher threshold	15 correct; 4 incorrect; 2 blank
4	Play outside	AND	Weight	Separate sheet	13 correct; 0 incorrect; 2 interrupted before completing; 6 did not attempt

Table 2: Results from student worksheets.

#	Name	Problem Features and Content of Video
4	Play outside	Two students collaboratively using the worksheet to decide if using a value of 5 for weight 1 would still produce correct results.
5	Ham & cheese	Initial weight has the correct sign but is too large. One student solving it correctly.
11	School band	"a OR NOT b" problem: requires a negative weight and negative threshold. One student solving it correctly.
13	Missing the bus	Includes an irrelevant input. Teacher is reviewing the problem with the class, who all agree that the weight on the irrelevant input should be zero.
17	Garden supplies	Four inputs, two of which require negative weights. One student solving it correctly; another making an attempt but getting interrupted before correcting their error.

Table 3: Content of video recordings.

1. Relationship Between Inputs & Desired Outputs Based on the Problem Statement. We observed students reading the question, evaluating each input, and determining the desired output. This mirrored what they observed in the teacher's modeling for the whole group. In the videos we collected of students solving problems in Neuron Sandbox, each video starts with the students reading the problem, identifying what the questions is asking, e.g., "he only needs one of those [to be true]" – indicating it's an OR problem. Then they evaluate each row of the inputs. For example, in the school band problem ("Students can join the school band if they know an instrument or have never dropped music class"), the output should be 1 if either the first input is 1 or the second input is 0. Student 3 starts walking through each of the inputs in the truth table in the first row and says, "So this one is like zero [pointing to the first row of inputs where both values are zero and tapping on the first input] so he doesn't know an instrument so he can't join [pointing at the desired output column] and this one like he didn't drop a music class so he could join [pointing at the desired output column which indicates a 1 and a green check mark]. Notice in this explanation the student substitutes the current value of the input into the problem statement and evaluates the possibility of the answer based on the conditions of the problem. This is representative of what we observed for students who were confident in their problem solving. For students who were struggling to get started, we observed the teacher modeling this process for them for the first row of inputs and then asking them to do it for the other rows.

2. Relationship Between Weighted Inputs & Activation (Sum). Most students were able to correctly compute the weighted inputs and sum those values to compute the activation for each row of the worksheet. We observed several kinds of errors in students' calculations. For example when multiplying 1×0 we observed that several students wrote 1 for the product instead of the correct result of 0. Similarly, we observed several students multiplying 1×1 and writing the product as 2 rather than 1, suggesting they were adding the values instead of multiplying them. Sometimes students' miscalculations would affect their activation and this led to an error in the activation they calculated. Other times we observed that despite errors in their calculation of the weighted inputs, students correctly calculated the activation. This could be due to students working collaboratively and catching each other's errors. Another possible explanation is that students caught their own errors but did not record the corrected weighted input values. Erasures on some of the papers in these columns suggest that some students did correct their calculations.

3. Relationship between Activation & Threshold Based on the Desired Outputs. We found that students often repeated the rule for identifying an appropriate threshold: it needs to be "greater than the Nos but less than the Yeses". For early problems, most students chose thresholds at the midpoint between the largest No value and the smallest Yes value, e.g., for AND problems where the largest No activation is 1 and the smallest Yes is 2, they chose a value of 1.5. For later problems we observed more students choosing diverse values falling somewhere within the correct range, suggesting they understood the concept of threshold and how its relationship to the activation reflected the desired output.

We initially observed students verbally using the rule to identify a range of weights in large group discussions on Day 2. These threshold range discussions were sparked by

Student 16 asking if the threshold could only be a single number or if it could be a range of possible values between 1 and 2, such as 1.2. Other students also shouted out alternative values. Five students recorded ranges of numbers in the threshold column on the first worksheet problem (PB&J), e.g., "1.1 – 1.99", mirroring values shared during this discussion. This suggests that this was an important concept that students wanted to remember. In addition, one of these five students (#10) also wrote at the bottom of the page "Threshold greater than number" to help them remember this key takeaway for this activity. We observed many students using the rule to evaluate the rows of inputs in their table to determine the threshold. Student 16 demonstrated his mastery of the concept through proposing a threshold for an imaginary range of problems. Student 16: "What if there were different threshold numbers, like the No=3 and Yes=4? Could I pick 3.2 or 3.5?" Similarly, on the third worksheet problem students were asked to solve an AND problem where both the weights were 2. In this problem the students were solving for the threshold and the activations were 0, 2, 2, and 4. Fifteen out of 21 students answered this question correctly by choosing a threshold between 2 and 3.9. Two students used their understanding of inequalities to pick the lowest possible threshold, 2, based on the highest valued No which was 2. The diversity of answers for this question suggests that by the time they solved three questions students understood that thresholds aren't unique values and they just needed to pick a value within the correct range. These examples are representative of our observations of other students relying on the activation $>$ threshold rule to determine the threshold and explain how they chose their threshold value.

RQ3: Prerequisite Knowledge and Skills

Learning to solve these sorts of problems requires certain knowledge and skills at the start of the exercise. We have identified the following:

1. Reading and understanding the problem
   (a) Reading comprehension
      - What is the question asking?
      - Recognizing the problem type: AND, OR, or something else
   (b) Critical thinking
      - Elimination of irrelevant information
      - Recognizing the logical role of an input: excitor or inhibitor
2. Mathematical knowledge
   (a) Multiplication by zero
   (b) Testing inequalities (specifically "greater than")
   (c) Understanding decimal fractions, e.g., recognizing that 1.5 is greater than 1 and less than 2
   (d) Multiplication and addition of negative numbers
3. Problem solving skills
   (a) Decomposing a problem into steps
   (b) Keeping track of steps
   (c) Recognizing connections between values
   (d) Checking results for correctness

Discussion

Middle school students can learn to solve linear threshold unit problems, but they require more scaffolding than we originally anticipated. Neuron Sandbox provides memory assistance by keeping the entire truth table visible all the time, and computational assistance by computing the activations and actual outputs for all rows simultaneously. It also provides logical assistance by highlighting rows where the actual output does not match the desired output. But middle school students also need problem solving assistance: help with breaking a problem down into steps, performing steps in the proper order, keeping track of intermediate results, and combining pieces to assemble a solution. The worksheet provided this assistance, guiding them to successful solutions for several problems.

Once students properly understood the domain, they could reason more independently. We saw this in both their successes and their mistakes. Students successfully took on "solve for weight" problems that did not fit the "solve for threshold" worksheet structure. On the other hand, some students mistakenly used an "AND" pattern for desired outputs when they should have used an "OR" pattern. The fact that they knew these patterns is evidence of learning, despite occasionally choosing the wrong one.

Providing appropriate scaffolding through worksheets, teacher modeling, and peer support drastically changed students' attitudes toward the material. What began as a frustrating experience on day 1 evolved into an engaging puzzle-solving session on day 2. Students were pleased that they had achieved mastery of the problems. At the conclusion of day 2, one girl asked the investigator, "Why is this so fun?"

Limitations

Day 1 exposure to the tool without the worksheets is a confound whose effects are unknown. The modest sample size of 21 students with a single teacher, and the fact that these were mostly high-ability students, calls for additional studies to assess whether the results generalize to other student populations.

Conclusions

We see three benefits of students learning to solve linear threshold unit problems in middle school. Obviously they are learning about the fundamentals of neural computation, an important component of artificial intelligence technology. They are also strengthening their mathematical and problem solving skills. Finally, learning to express decision problems in terms of linear equations and thresholds gives students a taste of the numerical, non-symbolic approach to computing that characterizes much of modern AI, and contrasts with the discrete symbolic framework they encounter in a typical middle school computing class.

Acknowledgments

This study was funded by National Science Foundation awards DRL-2049029 and DRL-2048502. Development of Neuron Sandbox was funded by National Science Foundation awards DRL-2049029 and IIS-2112633.

References

Birt, L.; Scott, S.; Cavers, D.; Campbell, C.; and Walter, F. 2016. Member Checking: A Tool to Enhance Trustworthiness or Merely a Nod to Validation? *Qualitative health research*, 26(13): 1802–1811.

Chen, A.; Pawar, N.; and Touretzky, D. 2024. Neuron Sandbox. https://www.cs.cmu.edu/%7Edst/NeuronSandbox.

Clarke, V.; and Braun, V. 2014. Thematic Analysis. In Teo, T., ed., *Encyclopedia of Critical Psychology*, 1947–1952. New York, NY: Springer New York. ISBN 978-1-4614-5583-7.

Code.org. 2020. How Neural Networks Work. YouTube video available at https://code.org/curriculum/how-ai-works.

Crash Course. 2019. Supervised Learning: Crash Course AI #2. YouTube video available at https://thecrashcourse.com/topic/ai/.

Ferrie, C.; and Kaiser, S. 2019. *Neural Networks for Babies*. Naperville, IL: Sourcebooks Explore.

Gelder, W.; Yu, X.; Touretzky, D.; Gardner-McCune, C.; and Uchidiuno, J. in press. From Lecture Hall to Homeroom: Co-Designing an AI Elective with Middle School CS Teachers. *International Journal of Artificial Intelligence in Education*.

Sato, K. 2016. Understanding neural networks with TensorFlow Playground. *Google Cloud Blog*. July 26, 2016.

Shamir, G.; and Levin, I. 2021. Neural network construction practices in elementary school. *KI-Künstliche Intelligenz*, 35(2): 181–189.

Tedre, M.; Toivonen, T.; Kahila, J.; Vartiainen, H.; Valtonen, T.; Jormanainen, I.; and Pears, A. 2021. Teaching Machine Learning in K–12 Classroom: Pedagogical and Technological Trajectories for Artificial Intelligence Education. *IEEE Access*, 9: 110558–110572.

Thurmond, V. A. 2001. The point of triangulation. *Journal of nursing scholarship*, 33(3): 253–258.

Touretzky, D.; Gardner-McCune, C.; Cox, B.; Uchidiuno, J.; Kolodner, J.; and Stapleton, P. 2023. Lessons Learned From Teaching Artificial Intelligence to Middle School Students. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 2 (SIGCSE 2023)*, 1371. Association for Computing Machinery.

Touretzky, D. S.; Chen, A.; and Pawar, N. 2024. Neural Networks in Middle School. *ACM Inroads*, 15(3): 24–28.