Plush Neuron: AI for 8th Grade

Learning to Think Like a Neuron

Cody Soska | Carnegie Mellon University | November 4th, 2025

Introduction

Young learners today will not know a world untouched by *artificial intelligence*. By the time they reach middle school, many have been using AI assistants, games, and smart devices for years. As current technology users and future decision-makers, students must develop AI literacy - an understanding of how AI works - so they can engage meaningfully with AI in their lives [Touretzky et al., 2019]. Hands-on experiences are especially powerful for building this literacy. Interactive modeling activities allow learners to construct foundational knowledge by embodying steps in the process. Education research shows that when students design and experiment with models of complex systems, their understanding deepens [Touretzky et al., 2022] and the "black box" these systems inhabit is demystified. Allowing middle schoolers to play and learn with a simple neuron model can indeed make AI less mysterious: even young students can develop useful mental models of AI concepts when given the chance to interact and create with AI representations [Touretzky et al., 2022].

One such tool, the **Neuron Sandbox**, provides a browser-based simulation of a simple, individual neuron, and has demonstrated the effectiveness of interactive neuron models for this age group [Touretzky et al., 2025]. This curriculum extends and builds on the Neuron Sandbox by taking the same concept into the physical world in the form of a single *linear threshold unit* (LTU). The Plush Neuron offers a tangible, interactive model of an LTU that complements the digital model. As Touretzky (2024) notes in advocating a hands-on approach to neural networks in middle school, giving learners an active role in constructing and experimenting with AI models greatly enhances understanding. By providing a physical counterpart to the virtual neuron in Neuron Sandbox, the Plush Neuron furthers those educational goals - students can literally hold and manipulate a "neuron," (LTU) making the learning experience concrete and multisensory in ways that deepen engagement.

In this curriculum, 8th grade students explore basic neural computation and decision-making using the Plush Neuron device. This plush neuron is a physical simulator of a simple artificial neuron (LTU) with three inputs and one output. Through a series of fun, relatable scenarios, students will use the Plush Neuron to model decision problems (e.g. “Can we make a PB&J

sandwich?” or “Should I wear boots today?”) and see how an artificial neuron *“decides”* based on inputs. This approach connects abstract AI concepts to everyday experiences, making AI less like “magic” and more like a reasoning process students can understand. By adjusting the neuron’s weights and net activation (the parameters that determine its output), students essentially “train” their mini AI model, gaining insight into how larger AI systems learn from data. By understanding the logic behind neural computation at the simplest levels, students will gain an understanding of the types of questions these tools can be helpful in solving. Gradually, students will build fluency in knowing how to frame their questions to get the most out of what AI can offer as it grows and changes.

Just as importantly, these activities are designed with student engagement and empowerment in mind. When learners get to experiment, create, and solve problems with technology, they become active participants rather than passive consumers. This boosts their interest and confidence. Providing youth with accessible, hands-on AI tools can inspire them to express themselves, sustain their curiosity, and build a sense of identity as creators of technology [Touretzky et al., 2022]. In short, this curriculum uses a constructionist, learning-by-doing approach to make AI concepts approachable. Eighth graders will physically model a simple "brain cell" decision, reflect on how it works, and connect it to the broader world of AI - laying a foundation for informed and empowered engagement with AI in years to come.

Standards Alignment: AI4K12 Five Big Ideas

This curriculum is aligned with the AI4K12 Initiative’s "Five Big Ideas in AI" for the 6-8 grade band [Touretzky and Gardner-McCune, 2021]. Each lesson explicitly ties into one or more of these big ideas, ensuring students build understandings that map to emerging national AI education guidelines. Below is an overview of how the Plush Neuron activities connect to each of the Five Big Ideas:

1. **Perception:** *AI systems use sensors to perceive the world.* In our lessons, the Plush Neuron’s manual inputs represent simplified "sensors" or conditions (e.g. *Is it raining? Do we need boots? Do we have peanut butter?*). Students experience how an AI agent (the neuron) takes in signals from the environment and evaluates these against a pre-set threshold, concretely addressing how AI perceives input factors from its world.

2. **Representation & Reasoning:** *AI agents maintain internal representations of the world and use them for reasoning.* Students set the neuron's weights and threshold and observe how the AI (the Plush Neuron) uses those parameters to reason through input data. *Do the inputs forming the net activation excede the thresholding function?* In doing so, they see how an AI "represents" a rule (through weights/threshold) and applies logic to reach a

decision.

3. **Learning:** *Computers can learn from data.* In these activities, students iteratively adjust neuron parameters to "train" it to produce desired outcomes, mirroring a basic machine learning process. Although we are manually setting weights (rather than using an algorithm), the concept that the neuron's behavior can be improved by tweaking parameters is akin to learning from experience (data/student observations).

4. **Natural Interaction:** *Intelligent agents interact with humans in natural ways.* In our context, students practice explaining the neuron's decisions using everyday language and relate the neuron's simple decision-making to human scenarios. This alignment of machine reasoning with human understanding touches on the idea of natural interaction - for example, in the family voting scenario, the AI mediates a group decision in a human-friendly way.

5. **Societal Impact:** *AI can impact society in positive and negative ways.* Throughout the curriculum, activities encourage reflection on fairness, responsibility, and the impact of decision rules. For instance, when students compare a majority-rule vs. unanimous rule in a family decision model, they discuss what seems "fairest" - highlighting that even a simple algorithm's settings (like a threshold) can change outcomes and affect people who may have been excluded from the start. These discussions connect to broader issues of how AI decisions might mirror or challenge human values and real-world issues.

Learning Goals and Objectives

Learning Goals: By engaging with both digital simulations (like the Neuron Sandbox) and the physical Plush Neuron, students demystify AI and understand it as logical, not magical. They learn how inputs, weights, and thresholds work together to create outputs in a neural model, bridging abstract concepts and real-world reasoning.

By the end of this curriculum, students will be able to:

- **Identify and describe neuron components and their functions.** For example, students can point out the Plush Neuron's inputs (pushbuttons), weights (dendrite input dials), threshold (body input dial), and output (axon lights), and explain what each does in the model.
- **Translate real-world scenarios into truth tables or logical rules.** Given a scenario, students can enumerate possible input combinations and desired outcomes in a table or list. For instance, for the PB&J sandwich problem, they list all ingredient present/absent

combinations and determine for each whether a sandwich is possible. This aligns with the Neuron Sandbox's *Expanded Worksheet*
- **Configure neuron parameters to match decision rules and execute the model.** Students will set the Plush Neuron's weights and threshold to implement specific logic functions for each scenario (such as an AND function requiring both inputs, or a majority OR function requiring 2 out of 3 inputs).
- **Predict and explain the neuron's outputs for given inputs (and vice versa).** Students practice anticipating what the neuron will do with certain inputs and justify why. They can also determine what inputs would be needed to get a certain output, just like in the Neuron Sandbox. In essence, they learn to read the neuron's "mind" and explain its reasoning in everyday terms.

These objectives align with the educational goals of the Neuron Sandbox software [Touretzky et al., 2025] while taking advantage of a physical model. Through both virtual and tangible experimentation, students gain a robust understanding of how a simple LTU operates.

Formative Assessment Strategies

To ensure students are meeting the objectives throughout the unit, each lesson includes formative assessments. These are informal checks - questions, observations, or short tasks - that let the teacher gauge understanding in real time and provide feedback. Each objective above has a corresponding assessment strategy. For example:

- **Assessing component identification:** Early in the unit, as students handle the Plush Neuron, the teacher can ask them to point out each part and describe its function. The teacher might hold up the device and say, "Show me where the inputs go in... What is this knob for?" (indicating the threshold dial). Listen for correct identifications (e.g., distinguishing inputs vs. output, weight knobs vs. threshold) and simple explanations ("Inputs are like the neuron's questions" or "This dial sets how many yeses it needs to light up."). Misconceptions (like confusing weight and threshold) can be noted and clarified on the spot.

- **Assessing scenario translation to a truth table:** During scenario introductions, have students fill out a quick chart or worksheet. For instance, for the PB&J sandwich problem, provide a blank table of all input combos (peanut butter present or not, jelly present or not) and ask students to predict "Can we make the sandwich?" for each combo. Circulate and check if they correctly mark the outputs (only the case with both PB=1 AND J=1 should get a "Yes"). If some are incorrect, use probing questions: "If we have jelly but no peanut butter, can we really make it?" to prompt corrections. This written artifact lets you see if they grasp the all-or-none logic *before* using the neuron.

- **Assessing neuron configuration skills:** This can be done through observation and quick performance checks. As students work in pairs with the Plush Neuron, the teacher listens and watches: Are they adjusting the threshold knob appropriately? Are they testing multiple input combinations methodically? One specific formative check is to ask each group to demonstrate their neuron's output for a given test case. For example: "Show me what your neuron does if we have peanut butter (Input1 = 1) but no jelly (Input2 = 0). What output do you predict? Let's see if it lights up." If their prediction or their configuration is wrong, it opens a discussion to troubleshoot and debug the model using reinforcement learning through correcting mistakes.

- **Assessing prediction of outcomes:** Use questioning during activities to probe student understanding of the model's logic. After a neuron is set up (or before testing it), present a hypothetical situation and have students privately vote or write down what they think the neuron's output will be *before* they check it. For instance: "Suppose it's snowing (Input1) and muddy (Input2) but *not* raining (Input3) - will our boot-deciding neuron say YES or NO to wearing boots?" Students commit to an answer (perhaps via a show of hands or a mini whiteboard session), then they test the neuron. If their predictions match the neuron's behavior, it indicates they understand the rule. If not, discuss which input combination tripped them up. This quick predict-and-check approach makes student thinking visible.

- **Assessing explanation of reasoning:** After each activity, have students articulate *why* the neuron gave the outputs it did. This could be a turn-and-talk with a partner or a quick written reflection. For example, ask: "Why did the neuron decide that we should order pizza in scenario X, but not in scenario Y?" Listen to a few students share or collect their written responses. You are looking for explanations that reference the inputs and the threshold rule (e.g., "In scenario X, two conditions were met, exceding the threshold, so it said yes. In scenario Y, only one condition was true, so it wasn't enough."). These explanations demonstrate whether students can connect the neuron's internal weight and function to the outcome in context.

Instructional Activities

The following activities guide the instructor through facilitating exercises using the Plush Neuron in progressively richer decision-making scenarios. Each example scenario introduces a different logical function or rule (AND, OR, majority vote, unanimous decision, etc.), building on the previous ones. Students work in small groups with a Plush Neuron to set up the inputs and parameters, test outcomes, and discuss the results. Throughout, the teacher facilitates questions and highlights connections to AI concepts and the AI4K12 big ideas.

Example 1: PB&J Sandwich Decision (AND Logic)

Scenario: *"Can we make a peanut butter and jelly sandwich?"* This first activity introduces the neuron model in a familiar context to that of the Neuron Sandbox. The decision depends on having the necessary ingredients. Students will model the situation where you need certain ingredients to successfully make a PB&J sandwich using the **AND** function.

- **Basic Version:** In the simplest scenario, assume the sandwich requires two key ingredients: peanut butter and jelly. (We'll ignore bread for now to keep it simple, assuming bread is always on hand.) We use 2 of the Plush Neuron's inputs for this: one input represents "Have Peanut Butter?" (yes = 1, no = 0) and the second represents "Have Jelly?" (yes = 1, no = 0). The desired outcome is that the neuron outputs 1 (YES - we *can* make the sandwich) *only if both* peanut butter and jelly are available. In all other cases (one or both ingredients missing), the output should be 0 (NO sandwich). This effectively encodes a logical **AND**. To configure the Plush Neuron for this function: set the weight of each of these two inputs to 1 and **set the threshold to 1.5** (1 fully-lit, green LED and 1 half-lit, green LED). That way, the neuron requires a total input sum of 2 (1+1) to fire. If only one ingredient is present (input sum = 1) or none (sum = 0), it will not reach the threshold, and the neuron's output will remain 0.

- **Setup:** Begin by setting the scene. Ask the class, "Who likes PB&J sandwiches? What do you need to make one?" As students list ingredients, emphasize peanut butter and jelly as crucial. Explain that today the "decider" will be our Plush Neuron, which will light up if and only if we *can* make the sandwich. Divide students into small groups (2-3 students per Plush Neuron is ideal). Give each group a Plush Neuron. Have them label **Input 1** as "Peanut Butter" (you might provide a small PB jar icon or sticker for that input) and **Input 2** as "Jelly". (Input 3 on the plush can be left unused in this activity and should be set to zero/blue to avoid accidental summation.) Ensure everyone understands that an input can be either 1 (we have that item) or 0 (we don't have it).

  Note: *The dials on the dendrites and the threshold dial on the cell body use rotary switches. These do not "reset" themselves and will preserve whatever position they are in, even after turning off the device. Instructors should remind students to return all knobs to zero position (single blue LED) and double check this before beginning a new exercise; this is the neutral state for all activities (all dendrites and threshold lights should be 0, blue LED).*

- **Activity Steps (Basic):**

    1. **Truth table brainstorm:** Before using the neuron, have groups list all possible situations of having/not having PB or J. They should come up with four cases for the two ingredients: (0,0), (0,1), (1,0), (1,1) - where, for example, (1,0) means PB = yes (1), J = no (0). For each case, ask them to predict if a sandwich is possible. Circulate and listen to their reasoning. They should conclude that only (1,1) (having both PB and J) yields a "yes, we can make it." All other combinations yield "no." This discussion frames the target behavior for the neuron before they even configure it.

    2. **Configure the neuron:** Guide students in setting up the Plush Neuron according to the function. Explain that we want it to output "yes" only in the case where both PB *and* J are present (the (1,1) case). Ask, "How many 'yes' inputs should the neuron require to light up for a sandwich?" Students typically realize it should require 2 (both ingredients). Show them how to adjust the threshold dial to 1.5 if necessary; be sure to point out how the sound the neuron makes increases in accordance with the weight of the input. The beginning state for dendrite weights are 0 (single blue LED) and should be reset before each exercise. Turning the dial clockwise will set a 1 (single green LED); ensure students understand this before testing. Emphasize that now the neuron is configured to test input combinations.

    3. **Test all cases:** Now students systematically test the neuron by squeezing the end of each dendrite simultaneously (may require teamwork) for each combination of ingredient availability to verify it behaves as expected:

        - PB = 0, J = 0 (no peanut butter, no jelly) → Check the neuron's light.
            - **Expected:** Off (0 output), since 0 + 0 = 0 which is below the threshold 2.

        - PB = 1, J = 0 (only peanut butter)
            - **Expected:** Off (0 output), since 1 < 2.

        - PB = 0, J = 1 (only jelly)
            - **Expected:** Off (0 output), since 1 < 2.

        - PB = 1, J = 1 (both present)

- **Expected:** On (1 output - the neuron's light turns on), since 1+1 = 2 which excedes the threshold.

4. As they do this, the students are essentially checking that the neuron's "truth table" matches the real logic of the problem.

5. **Discussion during testing:** Ask groups, "Why did the neuron only light up for that one combination (PB and J)?" Have them articulate that both ingredients needed to be present because of how we set the threshold to 1.5. If a group's neuron doesn't behave as expected, troubleshoot with them. Common issues might be: the threshold was accidentally set to 2 or <1 (resulting in no firing or continuous firing), or an unused input was mistakenly left on. Use these moments to reinforce how each parameter matters. For example, if someone had threshold = 0.5 by mistake, the neuron would incorrectly fire with only one ingredient - highlighting why the threshold must be set correctly. If an unused third input is accidentally set to 1, it might inadvertently contribute to the sum - so ensure any unused inputs are effectively set to 0/off/blue and that students are not squeezing the tips of unused dendrites. These troubleshooting moments help solidify understanding that the neuron's behavior strictly follows the logic encoded by weights and threshold.

6. **Connect to neural terminology:** Introduce or reinforce proper vocabulary in context. For instance, say: "We set the *threshold* to 1.5 because we needed an input sum of two ones, which is greater than 1.5. We gave equal *weight* to PB and J, so each ingredient counts the same (1). This neuron is basically computing **PB AND J**." Connect this to *Big Idea 2 (Representation & Reasoning): the neuron's internal reasoning was "if PB AND J, then output yes."* Students have just implemented that reasoning in a tiny AI model.

- **Wrap-up:** Bring the class back together and have groups share observations. Ask, "What did you notice about how the neuron decided on the sandwich?" Expect responses like, "It only said yes when *both* things were there." Highlight that this is analogous to how a rule-based AI might work for a similar decision. For instance, you can mention a simple security system analogy: "This is like a door that only unlocks if two keys are turned at once - both conditions must be met." Emphasize that the students essentially *programmed* the Plush Neuron by setting its rule (through the threshold and weights). To foreshadow the idea of learning (Big Idea 3), note that if the neuron had been *trained* on examples (rather than explicitly programmed), it would have eventually learned the same rule: *if PB AND J then yes*. In this activity, **the students were acting as the learning algorithm**, adjusting the neuron's parameters to get the desired behavior. Be sure to clarify while explaining that these plush neurons cannot learn and require inputs to be set.

Bonus! Version: "PB&J 2.0: Don't Forget the Bread!": In this extension, we add a third input to represent having bread. Now the question becomes: Can we make a proper PB&J sandwich with bread? Most students will quickly agree that bread is indeed required (a PB&J without bread is just a mess!).

So the updated rule is: the neuron should output YES only if peanut butter **AND** jelly **AND** bread are all present, a 3-out-of-3 rule. This is a perfect use of the Plush Neuron's third input. Students will configure the neuron with three inputs **(PB, J, Bread)**, each weight = 1, and set the threshold to 2.5 (requiring all three inputs to be 1 for the neuron to fire; 2 fully-lit green LEDs and 1 half-lit green LED on the threshold light bar). Essentially, this makes the neuron perform a 3-input **AND** function.

Bonus! Setup: Tell students: "Now we're introducing bread into the equation!" Hand out a third ingredient card or token labeled "Bread" to each group. Have them attach or assign it to the Plush Neuron's third input. **Reset the neuron** (reset all input to 0 to start) and make sure everyone knows the new goal: **the neuron should only light up if PB = 1, J = 1, and Bread = 1**. Emphasize that this is an even stricter condition than before, since now all three must be present.

Activity Steps (Bonus):

1. **Predict with 3 Inputs:** Ask groups to think about the possible combinations now (with 3 binary inputs, there are 8 combinations). Which combinations should result in YES? **They should conclude that only the case [PB=1, J=1, Bread=1] should give a yes** if all three ingredients are truly mandatory. Some may wonder about edge cases like "What if there is no jelly but we have honey as a substitute?" Encourage them to stick to the defined inputs for now (or note it as a creative thought for later). This prediction step reinforces logical reasoning with an extra input.
2. **Configure Neuron for 3 Inputs:** Students now set the Plush Neuron's threshold to 2.5. All three weights remain at 1. Walk around to verify each group has threshold = 2.5. They have created a neuron state that is an **AND** with three conditions (similar structure to the previous **AND**, just with one more requirement).
3. **Test Examples:** Have students test a few scenarios:
    - "Try one with something missing: PB=1, J=1, Bread=0 - does it light up?" (It should not light, output stays 0, since one ingredient is missing.)
    - "Now try with all three at 1 - what happens?" (It should light up, output 1, as expected.)

   They will see the neuron is even pickier now. If any group proposed a 2-out-of-3 rule (threshold 2.5) for an "acceptable sandwich", let them share that outcome and perhaps demonstrate it with their neuron if they set threshold to 2.5 and one of the dendrites with a weight of 2. It's interesting for the class to see a different rule in the same model and discuss which rule better fits the definition of a PB&J.

4. **Faded Scaffolding:** Consider not directly telling students the threshold this time. Instead, pose the question: "Previously, a threshold of 1.5 worked for two ingredients. What do you think the threshold should be now for three ingredients?" Let groups discuss and reason it out. Most will arrive at 2.5 (some might say "you need all three, so 3"; remember **the dendrites' weight must be greater than the threshold**). This gentle removal of scaffolding gives them a taste of figuring out the model themselves using logic and pre-trains for more complex activities.
5. **Wrap-Up (Bonus):** Congratulate students on mastering the PB&J neuron, now with three inputs. They have essentially created a mini three-input neuron that models a real decision scenario. Take a moment to tie this to *Big Idea 3: Learning. Again, noting that in a real AI situation, an algorithm could learn the rule "need PB, J, and bread" from data (examples of successful vs. failed sandwich attempts).* In this activity, the students acted like the learning algorithm, adjusting the neuron's inputs until its output matched the rule - a hands-on introduction to how machines can learn rules from requirements.
   Ask a reflective question to solidify understanding: "How did adding a third ingredient change the difficulty or the way we set the neuron?"
   Expected answers: it made the condition stricter (we had to raise the threshold), and there were more combinations to consider. Highlight that while the number of factors grew, the fundamental principle - summing inputs and comparing to a threshold - stayed the same. This foreshadows that more complex AI decisions (with many inputs) might require networks of neurons or more intricate setups, but each neuron still does a similar, simple job. You can whimsically ask, "Would a 4th ingredient be easy to handle with our single plush neuron device?" (Answer: not without another device or some trick, which hints at needing multiple neurons or a network for more complexity). This naturally leads into trying a scenario with a different logical structure next time, not just a strict AND.

Example 2: Wearing Boots (OR function)

Scenario: *"Should I wear boots outside today?"* In this activity, students use the Plush Neuron to decide if conditions warrant wearing boots. This scenario introduces an **OR** function (in the basic version, any one of several conditions suffices to trigger the outcome). It can later be extended to explore a "majority needed" logic. The decision will be based on weather conditions - for simplicity, we'll use three possible conditions as inputs.

- **Basic Version:** We consider three factors that might each justify wearing boots:
  - **Is it raining?**

- **Is it snowing?**
- **Is it muddy?**

- If **any one** of these is true, we will wear boots (output = YES). In other words, even one adverse condition is enough to break out the boots. This is effectively a logical **OR** among the inputs. To implement this with the Plush Neuron's three dendrites: assign each input to one of the conditions (Rain, Snow, Mud). We want the neuron to fire (output 1) if *at least one* input is 1. The simplest configuration is to set each weight to 1 and set the threshold to 0.5. That means if the sum of active conditions is > 0.5, the neuron outputs "yes." Any single input being true will exceed the threshold. If none of the conditions are true (sum = 0), the output will be 0 (NO boots needed). Obviously, any combination of inputs set to 1 will yield a net activation greater than the threshold of 0.5.

- **Setup:** Start by asking students about times they wear boots. "Why do we sometimes choose boots instead of regular shoes?" List their answers - likely weather-related reasons: rain, snow, lots of mud, etc. Lead them to the idea that *any* of these conditions alone is a good reason to wear boots. Announce that the Plush Neuron will help us decide each day whether we need boots, based on the weather conditions. Have students imagine they are feeding the neuron three bits of information about the day: whether it's raining, snowing, or muddy. Form groups (if not already formed) and distribute the Plush Neurons (or have them reuse their device from the previous activity). This time, all three inputs will be in play from the start. Students label Input1 = "Raining?", Input2 = "Snowing?", Input3 = "Muddy ground?". Ensure they interpret 1 as "yes, this condition is present today" and 0 as "no, it's not present."

- **Activity Steps (Basic):**
  - **Define desired outputs (OR function):** Begin with a discussion to make sure students understand the decision rule. Pose a few example scenarios and ask if boots should be worn:
    - "If it's *raining* (yes) but not snowing (no) or muddy, should we wear boots?" Students should answer *Yes* (because rain alone is enough).
    - "If it's dry and sunny (no rain, no snow, ground is dry), boots or no boots?" Answer: *No* (no adverse conditions).
  - After a couple of examples, generalize: "So basically, if **any** of these conditions is true, we want a yes (wear boots)." Ensure students grasp the concept of an OR

condition - there are multiple triggers, but only one is needed to prompt the action.

- **Initial neuron setup:** Challenge the class: “How can we set our neuron so that it lights up even if just one input is on?” Let them propose ideas for the threshold. Someone will likely suggest threshold = 0.5 (since even one input of value 1 will meet it). Confirm that this is correct. Have each group set the threshold dial to 0.5. (Weights remain at 1 for all inputs.) Explain that this means “the neuron will say YES if the sum of the active inputs is 1 or more” - i.e., if one or more conditions are true.

- **Test cases systematically:** Have each group test a variety of weather combinations to make sure the neuron behaves as intended:

  - Exactly one condition is true (e.g., Rain=1, Snow=0, Mud=0 - it's only raining):
    - **Expected output:** YES (neuron lights up) because one input (1) > threshold 0.5.

  - Try a different single condition: Only snowing (Snow=1, others 0)
    - **Expected:** YES (lights up, since one input is on).

  - Only muddy (Mud=1, others 0)
    - **Expected:** YES.

  - No bad conditions (Rain=0, Snow=0, Mud=0)
    - **Expected:** NO light, since 0 < 1.

  - If two conditions or all three conditions happen (e.g., a really nasty day with both rain and mud = inputs 1,1,0):
    - **Expected:** YES (the sum will be 2 or 3, which is > 0.5, so the neuron definitely lights up).

- Students will observe that it doesn’t matter if multiple conditions are true - as long as *at least one* is true, the neuron outputs yes. This confirms the **OR** behavior.

- **Class discussion - noticing the OR pattern:** Ask, “Did anyone find a case where the neuron did *not* tell you to wear boots even though maybe you think you should?” Students should agree that as configured, it covers all cases where one or more conditions are present. (If any group is unsure, discuss their reasoning.) This discussion highlights that the neuron currently implements an inclusive "any condition triggers boots" OR rule.

- **Bonus Variation (Threshold = 1, 'Majority' boots rule):** After the basic **OR**, you can introduce a twist to illustrate how changing the threshold alters the decision policy. Some people might only wear boots if things are *really* bad. Ask, "What if our rule was that at least two of these things have to be happening to justify boots? For example, it's both raining *and* muddy." See if students can articulate that this would be a stricter rule. Quickly adjust the neuron together: have groups change the **threshold to 1.5**, keeping dendrite weights at 1. Remember, the weight of the net activation must **exceed** 1.5 (threshold); at least two conditions must be present.

    - Now discuss why someone might choose this threshold=1.5 rule. Ensure students understand the rationale: maybe they don't mind just a little rain, but if it's rain *and* mud together, then boots are necessary. In other words, requiring two conditions avoids overreacting to mild conditions; it sets a higher bar (*"only when the weather is quite bad do we wear boots"*).

    - **Test the new rule:** Have groups test combinations again under the threshold=1 setting:

        - One condition (e.g., just raining): Now the neuron's output should be NO (because 1 = 1, not greater, one adverse condition is not enough).

        - Two conditions (e.g., raining and muddy): Output YES (2 > 1, neuron lights up).

        - All three conditions: YES (3 > 1, definitely meets the requirement).

        - None: NO (0 < 1, as before).

    - Students will see that now a single factor alone is *not* enough; you need a combination of at least two. This illustrates a **more strict policy** encoded by a higher threshold.

    - **Reflect on the change:** Ask, "Why might someone choose the threshold=1.5 rule instead of the threshold=0.5 rule?" They might respond with ideas like: to avoid wearing boots when it's only a little bad outside; to only wear boots when it's *really* necessary because they slow you down, etc.. Highlight that the **threshold is essentially a policy choice** - it represents how strict or lenient our decision criteria are.

  - **Interpretation:** Have students explain in plain language what the neuron's output means now, under the new rule. For example: "When the neuron lights up now, what does that mean about the weather?" They should say something like, "It means it's bad enough outside (at least two bad conditions) that we should wear boots." And when it stays off: "It means it's not that bad - fewer than two issues - so regular shoes are fine." Getting them to translate the neuron's output back into the scenario context ensures they understand how the parameter change affected the decision criteria.

  - **Wrap-up discussion (OR vs. stricter condition):** Summarize that with the same three inputs, we experimented with two different decision thresholds: a simple OR function (threshold 0.5) vs. a "majority" rule (threshold 1). Highlight to students that **changing the threshold changes the decision policy** even though the inputs are the same. This is a key concept: even a simple AI model's behavior can be *tuned* by adjusting parameters. Relate this to *Big Idea 5 (Societal Impact) on a simple level - for instance: "If an AI were deciding something important, changing a threshold like this could make it easier or harder to get a positive result. We should always think about what threshold is fair or appropriate."* You can make it concrete by asking, "Which rule do you think is better or more fair for deciding on boots?" (There is no single right answer, but it gets them thinking about criteria and fairness.) For fun and reinforcement, you could do a quick class poll on it. This discussion connects the technical tweak back to human values and preferences, showing how an AI decision can be calibrated.

- After this activity, emphasize the takeaway: the same neuron can implement different decision policies just by changing one number (the threshold). That's a powerful insight - the technology is flexible, and it's up to humans (or a training algorithm designed by humans) to set it appropriately. Many students will have grasped that intuitively by seeing the difference between the two boot rules. This also segues into the next scenario: "We've now seen an **AND** (all conditions needed) and an **OR** (any one needed, with even a majority variation). Next, let's consider a decision where some factors might be more important than others... like deciding on getting a pet!"

Example 3: Getting a Pet (All Requirements AND Considering Trade-offs)

Scenario: *"Should our family get a pet?"* In this activity, students model a decision that has multiple requirements and potential trade-offs. It's a richer scenario that can spark discussion about which factors are non-negotiable in a decision. The Plush Neuron will simulate the family’s decision-making process for adopting a pet (e.g., a dog). The basic model uses an **AND** function (all key conditions must be met) but this scenario also invites students to consider exceptions or additional factors (like an inhibitory factor/negative input weight). It reinforces the concept of strict requirements and introduces the idea of a veto or critical factor that can override the others.

- **Basic Version:** Assume the family has three key requirements that all need to be true before saying "YES" to getting a pet:
    - **Parental Approval** - Do the parents/guardians agree to get a pet?
    - **Time to Care** - Does the family have enough time to take care of a pet (e.g., someone can walk the dog, feed the parrot, etc.)?
    - **Financial Resources** - Can they afford the pet (food, vet bills, etc.)?
- In the basic model, we treat all three as equally important and *all must be “yes”* (1) to get a pet. So, Input1 = Parent Permission, Input2 = Time Available, Input3 = Sufficient Money. The neuron should output 1 (Yes, get a pet) **only if all three inputs are 1**. This is essentially an **AND** of three conditions (very similar in structure to the PB&J scenario, but now with three factors like the bonus version). We set each weight to 1 and the threshold to 2.5 (requiring 3 out of 3 inputs to be true). If any input is 0 (meaning a requirement is not met), the sum will be less than 2.5 and the output will be 0 (No pet).
- **Setup:** Introduce the scenario with a question to the class: “Who here has a pet, or wants a pet? What kinds of things do your parents consider when deciding if you can have one?” Let students share ideas. As they mention factors, guide the discussion toward the three we’ve identified (permission, time, money). For example, if someone says “We need to have space for the pet,” acknowledge that space is a valid factor but note that we'll focus on these common three for now. Ensure everyone understands each of our factors: parents must say yes (if they forbid it, no pet), the family must have time to care for the pet (if everyone is super busy, it wouldn’t be feasible), and money matters (pets cost money for food, vet care, etc.). Summarize the family’s rule: “In our simplified model, if *any one* of these is missing, the answer will be NO to getting a pet.” Now have students form their groups (if not already in them) and label their Plush Neuron inputs accordingly: Input1 = "Parents' Approval (Yes/No)", Input2 = "Time Available (Yes/No)", Input3 = "Enough Money (Yes/No)". This sets the stage that all three need to be yes for a positive decision.
- **Activity Steps (Basic):**

- **All-or-nothing logic check:** Start by posing hypothetical situations to ensure students agree that *all* conditions must be satisfied for the family to get a pet:
    - "If your parents agree and you have time, but you have no money to feed the pet, should you get one?" Students will answer *No* (money is missing).
    - "If you have money saved and plenty of time, but your parents said no?" Definitely *No* (parental approval is missing).
    - "If parents and money are okay but no one has time to take care of the pet?" Also *No* (time is missing).
- Through these questions, students realize that any single missing factor is a deal-breaker. This reinforces that the function is an AND - every condition must be true for a yes.
- **Set neuron parameters (Threshold = 2.5):** Once everyone agrees on the logic, have students predict what threshold the neuron will need. They will likely say 2.5 (since there are 3 factors and all must be yes). Confirm this and instruct each group to set the neuron's threshold dial to 2.5 (weights remain 1). Explain that this configuration means "the neuron will only fire if the sum of its inputs is greater than 2.5 - which corresponds to parents yes, time yes, money yes."
- **Test a few scenarios:** Now have each group test some realistic combinations to verify the neuron works as intended:
    - **All three inputs = 1** (Parents yes, Time yes, Money yes) → The neuron *should light up* (output = 1). This represents the family meeting all requirements and deciding *Yes, we can get a pet*.
    - **Any scenario where one of the three inputs is 0** (e.g., parents say no, but time and money are yes) → The neuron should *not* light up (output = 0), because 2 < 2.5. Try a couple of different ones: no time, or no money, etc. In each case, with only two yeses and one no, the output should be No. This confirms that if any requirement isn't met, the neuron's output stays at 0.
- Students should quickly see that the neuron only gives a "yes" when *all three* conditions are satisfied, as expected.

- **Extensions & Discussion:** This scenario naturally leads to a richer discussion about decision criteria. Ask students if they think all factors are equally important in real life or if there could be exceptions. Two interesting extension ideas can be explored if time permits: an **inhibitory factor (veto)** and considering **trade-offs (relaxing the strict requirement)**.

  - **Extension 1: Modeling an inhibitory factor (veto)** - Introduce a "what if" scenario: "What if someone in the family is allergic to pets? If anyone is allergic, that one factor might override all others - even if time, money, and willingness are there, an allergy could veto the decision." Discuss: if we had a fourth input for **"Allergic? (1 = yes allergic)"**, what should the neuron output be if that input is 1? Students will unanimously say the decision should be *NO pet* if someone is allergic, regardless of the other inputs. This is a chance to talk about how we could model a **veto** or inhibitory input in a neural model. The Plush Neuron hardware only has 3 inputs, so we can't physically add a fourth in this case, but we can do a thought experiment. In a neural network, a veto could be modeled by a **negative weight** with a large magnitude. For example, if we had an allergy input, we might assign it a weight = -3, so that if allergy = 1, it subtracts enough from the net activation to always keep it below threshold (1 + 1 + 1 - 3 = 0, in a scenario where all other conditions are yes but allergy is yes as well, yielding a total of 0 which is < 2.5, therefore output 0).

    This can be modeled by "networking" a second neuron whose combined output = -3 (1 dendrite = -3; -1 value set on all dendrites; etc.) and plugging it into the jack of one of the dendrites, but this may take too long for a bonus activity. You could consider removing one of the requirements (say, money is no issue) and introducing this "veto" condition in its place. Students should be explicitly told to model this condition by turning the input dial down to -3 until 3 fully-lit red LEDs are displayed. Even if this functionality is not used, the concept can still be discussed or even tried on paper. This introduces the idea that some conditions can *negatively* influence a decision - connecting to more advanced neural network concepts (excitatory vs. inhibitory inputs). It shows that our model can be tweaked not just by thresholds, but by weights as well (some inputs might carry a "negative" impact).

  - **Extension 2: Considering trade-offs (relaxing the strict AND)** - Alternatively, pose a scenario focusing on possible trade-offs between factors. Ask: "If you could have at least 2 of these 3 factors, would that ever be enough for a pet?" Some students might argue, for example, *"If I have a lot of time and money, I could convince my parents eventually,"* suggesting that perhaps Parents' approval could come later or be won over if the other factors are strong. Another student might insist *"No way, parents must say yes no matter what"*. This can lead groups to come up with their own modified rules: some might stick to requiring all 3, while others propose *"maybe 2 out of 3 could be enough in some circumstances."* If groups have different ideas, let them adjust their neuron threshold accordingly just to experiment (e.g., try threshold = 1.5 for a scenario where they think any two conditions might suffice). Then they can test what that model would do: for instance, threshold 1.5 would allow a pet as long as any two conditions are met. Have them critique whether that makes sense. For example, if the neuron with threshold 1.5 outputs yes when Time and Money are 1 but Parents are 0, does that match reality? Likely not - it would mean the model says "yes, get a pet" even without parental approval, which conflicts with the real-world constraint that parents must agree. This usually leads students to realize that some factors (like Parent approval) are *non-negotiable*. The exercise of trying a threshold=1.5 model and seeing its implications helps reinforce why the original rule required all 3. It shows how a seemingly reasonable relaxation (2 of 3) can produce an outcome that doesn't fit the real situation (a yes without parent approval).

    ○ **You can explore either or both of these extensions** as time permits. The key is to prompt deeper thinking about the model's limitations and how real-life decisions might not always be a simple straight **AND** of factors. Students see that our neuron model is flexible and can represent different policies, but they must decide which matches reality or desired criteria.

- After discussing these extensions conceptually (and possibly trying the threshold=1.5 experiment), ask students what they learned. Emphasize how we can *tune* the model: we could, in theory, integrate a special "no-go" signal (allergy) with a negative weight, or adjust the threshold to allow some trade-offs. This ties back to the idea that AI designers (or family decision-makers, in this analogy) must choose what rules and weights to encode, reflecting their values or priorities.

- **Wrap-up:** Summarize what the class has done with the pet decision neuron. "This pet decision was an example of a strict **AND** function - all conditions had to be met. But we also saw that in reality, some decisions might have deal-breakers (like allergies) and some might have negotiables or ways to compromise (like maybe if enough other factors are strong)." Connect this to how AI systems often have to incorporate both types of factors: some rules are absolute, others are weighted. Tie this discussion to *Big Idea 5 (Societal Impact) briefly: for example, mention that an AI deciding something important (say, a loan approval AI) might have certain rules that absolutely deny if one condition is met (e.g., if the applicant is under 18, loan = no by rule). It's crucial to think about those* *rules and who sets them.* Our classroom example is benign, but it mirrors the idea of policy encoded in AI. By playing with thresholds and even the concept of negative weights, the students saw how values and priorities can be translated into the neuron's numbers. That's a deep connection between the *math* and the *meaning* of the model. Encourage students by pointing out: it's not just about getting the right number on the light indicator - it's about understanding the story those numbers tell. At this point, they've tackled strict **AND** and **OR** functions and even thought about majority and veto ideas. Now they're ready to tackle decisions involving multiple people's preferences, which adds another layer of complexity (one they've actually touched on with the idea of majority vs. unanimity).

Example 4: Family Dinner Decision - Voting on Burgers (Majority Vote Logic)

Scenario: *"Should we have burgers for dinner tonight?"* This scenario frames the neuron as a model of a group decision made by voting. Imagine a family of three (for example, two parents and one child) voting on whether to have burgers for dinner. The basic version uses a **majority rules** logic (at least 2 out of 3 family members want burgers), and the extension explores a unanimous decision or a veto-power concept. This is the first scenario where the neuron's output represents a **collective preference**, showcasing how AI might aggregate inputs from multiple people.

- **Basic Version (Majority Wins):** We have three inputs corresponding to the three family members' desires:
    - **Input1: Mom wants burgers? (Yes=1, No=0)**
    - **Input2: Dad wants burgers? (Yes=1, No=0)**
    - **Input3: Kid wants burgers? (Yes=1, No=0)**
- We assume in the basic scenario that the decision rule is **majority vote**: if two or more family members want burgers, the decision is YES (we will have burgers). If fewer than two (i.e., zero or only one person) want it, then the decision is NO (we won't have burgers tonight). To implement this, set each weight = 1 and threshold = 1.5 (since we need at least 2 votes of 1 to trigger a yes). This is very similar to the "boots with majority" **OR** function we already explored, but now it's interpreted in a social context (votes).
- **Setup (Basic):** Introduce the scenario in a relatable way: "Have you ever tried to decide as a family what to eat, and not everyone agrees?" Many students will relate to situations of voting on movies, restaurants, etc., with family or friends. Explain that in our

simplified family of three, we'll assume they go with whatever the majority wants. If at least two people want burgers, that carries the decision. If only one or none want it, they'll choose something else. Have students label the neuron inputs with the family members' names (or, generally, Person A, B, C if they prefer). They can even imagine *themselves* as each person in a role-play sense. Emphasize the meaning of inputs: a 1 means that person *wants* burgers tonight, a 0 means they *do not* (they'd rather eat something else).

- **Activity Steps (Basic):**
    - **Discuss majority logic:** Ask the class, "Why might a family choose at least 2 out of 3 as the rule, instead of requiring all 3?" Students will likely say things like: to be fair and not let one person's preference always dictate for everyone; so one person can be outvoted if most of the family wants something. Connect this to how groups often make decisions by majority vote, which usually feels fair in group settings.
    - **Set threshold for majority:** Students can infer the neuron's threshold easily here. Have them set threshold = 1.5 on the Plush Neuron (with all weights = 1). Explain: "We're saying the neuron will light up (yes to burgers) if it gets 2 or 3 yes votes. That's our majority rule."
    - **Simulate votes:** This part can be fun. Perhaps assign each student in a group to pretend to be one family member and set their input accordingly (1 if *their* person wants burgers, 0 if not). Let them try different vote combinations and see what the neuron outputs:
        1. **Only one yes** (e.g., just Mom wants burgers, Dad and Kid do not) - Neuron output? It should be *No* (0 output) because 1 vote < 1.5; the one vote is outvoted by two no's.
        2. **Two yes, one no** - Output should be *Yes* (the neuron lights up, since 2 > 1.5, majority wants burgers).
        3. **Three yes** -> Obviously *Yes* (3 > 1.5, everyone wants burgers).
        4. **Zero yes** (nobody wants burgers) - *No*.
    - They can act out scenarios: for example, "Mom and Dad want burgers, kid doesn't - what does the neuron say?" The neuron will light up (Yes) because the

majority (2 out of 3) is in favor. This makes the idea of majority rule concrete as they see the neuron's response.

  - **Verify logic:** Ensure each group tests enough combinations to convince themselves it's working as a majority vote decider. This mapping is straightforward (it's basically what they expect), so they should quickly find it logical: whenever at least 2 inputs are 1, the output is 1; otherwise 0.

- **Wrap-up (Majority rule):** Highlight that this is the first scenario where the neuron's output represents a *group* decision. It's modeling a social process inside the neuron. This showcases *Big Idea 4 (Natural Interaction) in a different way: here AI is mediating group preferences, or you could say it's doing a simple recommendation/decision based on multiple people's inputs.* For example, an AI might one day suggest a restaurant for a family by weighing everyone's votes or preferences. What the students built is essentially a tiny automated voting machine for dinner. Check for understanding by asking a few questions: "If only one person wanted burgers, why did the neuron say no burgers?" (Expected: because the majority needed 2 votes - one vote wasn't enough.) "If two people wanted them?" (Majority wins, so yes.) These quick questions confirm they grasp how the majority threshold works.

- **Bonus Version: "Unanimous or Veto?"** For an extension, explore altering the decision logic to examine a **unanimous decision** scenario or introducing a scenario where one person's strong objection can veto the decision. This teaches about stricter thresholds and touches on negative inputs (weights).

  - **Unanimous rule:** Change the threshold to 2.5 so that the neuron only outputs yes if **all three** inputs are 1. This means if even one family member isn't on board, burgers are off the table. Explain that this represents a family that only goes with an option if nobody objects. Students implement this simply by raising the threshold to 2.5 (weights remain 1). Have them test it: if two say yes and one says no, what happens now? The neuron stays off - *No burgers* - because one person's "no" prevented a unanimous agreement ($2 < 2.5$). Only the case of three yes votes will light it now. Ask, "In what situations might a family use unanimity instead of majority?" Students might say to make sure nobody is unhappy, especially if the one person's happiness is important. Or they might mention that for serious decisions, sometimes everyone's agreement is required.

  - **Veto power (single person override):** Alternatively, consider if one family member really hates burgers (maybe they're vegetarian or just strongly opposed). How could we model a situation where if *that specific person* votes no, the decision is automatically no - even if the other two want burgers? In a single neuron model, this is akin to giving that person veto power. There are a couple ways to think about implementing a veto. One way is essentially the unanimous rule we just did: requiring all 3 to agree effectively gives everyone equal veto power (any single "no" blocks the decision). If we wanted specifically one person to have veto power but not the others, that would involve weighting that person's "no" more heavily (a negative weight of larger magnitude for that person's input, perhaps). That gets complex for students at this level, so we must carefully frame it. You can explain: in a neural network, we could give that person's vote a strong negative weight to simulate a veto.

- **Discuss these possibilities:** Present them to the class: "Sometimes families won't do something unless *everyone* agrees - that's unanimous decision-making. Or maybe in your family, if one person really *dislikes* an idea, you don't do it, even if the others do - that's giving a sort of veto power to that person." See if students have heard of these concepts or can relate (maybe they mention needing everyone on board for certain activities, or a scenario where one parent's "no" can override everyone else). You can decide which to explore (or do both quickly, since they're just threshold tweaks for the neuron).

- **Activity Steps (Bonus):**

  1. **Unanimous requirement (Threshold = 2.5):** Have each group adjust the threshold to 2.5 on their neuron (weights still 1). Now the neuron will only light up if 3 out of 3 inputs are 1 (i.e., everyone votes yes). Ask them to predict: "If two say yes and one says no, what will the neuron do now?" They should predict *no light*. Let them test a scenario: two yes, one no → indeed, the neuron stays off (no burgers) because $2 < 2.5$. If all three say yes, it lights (yes burgers). Discuss: "Is this fair? When might a family use a unanimous rule instead of majority?" Students might answer that unanimous decisions ensure nobody is forced into something they don't want - perhaps important if the one dissenter would be really unhappy. They might also note it can be inefficient or overly strict for casual decisions.

  2. **Veto discussion (conceptual):** If time and curiosity allow, discuss how one person could have a veto. For example, maybe "Dad really hates burgers, so if Dad says no, it's an automatic no." In our model, the unanimous rule already covers the case where anyone's no is decisive. To specifically highlight one person, you could imagine weighting that

person's input differently (e.g., Dad's input might carry more weight or a negative influence when it's 0). This gently introduces the concept of input weights and is easily demonstrated. Have students choose a dendrite to represent this "veto" and set that dendrite's input weight to -3 (3 red lights). Then have them run a simple test - no, the neuron does not fire in the presence of such negative weight. They will have just seen conditions that result in the neuron firing, so this is a quick pivot. After, students should recognize this pattern: giving a single input the power to override the others.

  - **Wrap-up (Majority vs. Unanimous):** Bring the discussion back to what these variations mean. Ask, "Which method seems fairest - majority or unanimous - in the family dinner scenario?" Students may have different opinions. Some will say the majority is fairer because the group can't be held back by one person every time; others might say unanimous is nicer because it ensures everyone is happy. This highlights that how you set the threshold changes the decision policy, and there isn't a single "right" answer - it depends on context and what the group values (fairness vs. harmony, etc.). Relate this to *AI fairness (Big Idea 5) in simple terms: even a simple algorithm's setting (like our threshold) can change outcomes for people and what we consider a fair decision.*

- You might even have students role-play a quick vote on something fun (like voting on a class snack or activity) and see if the neuron's output (with a given rule) matches what they consider the fair decision. This drives home that the neuron was modeling a human decision process.

  By the end of the "Burgers" activity, students have seen that not only can the neuron model strict **AND** and **OR** functions, but it can also be tuned for majority rules and even conceptual vetoes. They've now tackled decisions with multiple people involved, and understood how changing a model's parameters (threshold or weights) can mirror different decision-making rules.

Example 5: Pizza Night Decision (Combining Conditions - Threshold Logic)

Scenario: *"Should we order pizza tonight?"* This scenario involves multiple **reasons** that could justify ordering pizza, illustrating another threshold-based logic. It reinforces the idea of needing at least two signals, but in the context of *reasons* rather than people's votes. In many ways, it's similar to the boots example (multiple conditions), but framed as needing a **combination of reasons**.

- **Basic Version:** Consider a family deciding whether to order pizza for dinner. Possible factors might include:
    - **It's Friday or a special occasion (they treat themselves on weekends or special events).**
    - **The family is too tired to cook (long day, no one has energy to cook).**
    - **They have no leftovers or alternative food at home (nothing else easy to eat).**
- We will assume that if **at least two** of these conditions are true, the family will order pizza. One reason alone might not be enough - for example, if it's Friday but the fridge is full of food and everyone isn't that tired, maybe they won't order; or if it's a Tuesday and they're very tired but there is food available, they might push through and cook. But two reasons combined would tip the decision to "Yes, let's get pizza tonight."

  To model this, we use the Plush Neuron's three inputs for these factors:

  Input1 = "Special Night (Friday?)"

  Input2 = "Too Tired to Cook?"

  Input3 = "No Food at Home?"

  We set each weight = 1 (if present) and threshold = 1.5, requiring two out of three conditions to be true for the neuron to output yes.

- **Setup:** Introduce this by asking, "When do your families decide to order pizza instead of cooking?" Students will likely mention things like weekends (Friday night treat), being busy or exhausted, nothing in the fridge, etc. List the three reasons above as a summary of common answers. State the rule clearly: "This family needs at least two of those reasons to say 'Alright, pizza tonight!'" Have each group label their inputs accordingly (Input1 = "Friday/Special night?", Input2 = "Nobody wants to cook (tired)?", Input3 = "No food at home?").
- **Activity Steps:**
    - **Predict outcomes:** Have students consider a few scenarios and predict the decision:

        - "It's a plain Tuesday (not special), everyone's a bit tired, and the fridge is empty - that's 2 reasons (tired + no food). According to our rule, would they order pizza?" Students should say *Yes* (threshold 1.5 surpassed).

        - "It's Friday but you did meal prep and aren't tired - that's 1 reason (it's Friday only). Order pizza?" According to the rule, no, since only one condition).

        - Perhaps another: "It's Friday and you're tired, but there *are* leftovers in the fridge (so two conditions: special + tired, with food available) - that's 2 reasons, so yes pizza." Or "It's Friday, not tired, no food - also 2 reasons (Friday + no food) - yes."

    - Go through a couple examples like these. This helps learners see why requiring at least two conditions makes sense - one alone doesn't justify it, but when two happen, it feels like a good idea.

    - **Configure the neuron:** Have groups set the threshold = 1.5, weights = 1 for all present inputs. Now each group tests different combinations of the three conditions to verify the neuron's behavior:

        - Whenever *two or more* of the conditions are true (any two, or all three), the neuron should light up (output yes, order pizza).

        - If fewer than two (i.e., zero or exactly one) are true, the neuron stays off (no pizza).

    - They should test, for example: (1,1,0) - yes; (1,0,1) - yes; (0,1,1) - yes; (1,0,0) - no; (0,1,0) - no; (0,0,1) - no; (1,1,1) - yes obviously; (0,0,0) - no. Through these tests, they verify the model aligns with the intended rule.

- **Discussion & Extensions:** Once the basic understanding is confirmed, discuss with students: "Do you think this logic might be different for different families?" Likely yes - some families might order pizza at the drop of a hat (with even one reason), others only on very special occasions (might need all three conditions!). This leads to a brief exploration of threshold extremes again, similar to what we did with the boots scenario:

    - **Family A (very lenient):** *"Even one reason and they order pizza."* Ask students what the pros/cons of that policy are. Pros might be: quick to treat themselves, more fun; cons: might order pizza too often, even when not necessary. In neuron terms, that family's threshold would be 0.5 (an **OR** function) - which is exactly what we had in the boot's basic scenario. If time permits, let groups quickly set threshold to 0.5 and see how the outputs change for this pizza model: basically it becomes an **OR** condition where any single reason triggers a yes (the neuron would light up even if only one condition is true).

  - **Family B (very strict):** *"They require all three reasons to order pizza."* Pros: very disciplined (they almost never order pizza unless it's really justified); cons: they rarely get pizza (maybe too strict). In neuron terms, that would be threshold 2.5 with our three inputs. Again, groups could try setting threshold to 3 and testing: now only the case of all three reasons being true lights the neuron. Any missing reason (even if two are true) means no pizza.

- Have students reflect: "Which threshold do you think matches your family's style?" This might be expected to generate some fun discussion and anecdotes. The point to drive home is that different threshold settings = different decision policies, and neither is universally "right" or "wrong" - it depends on context or preference. This echoes a theme in AI about tuning models to desired behavior: the designer (or in this case, the family) has to decide what criteria should be in place for a positive decision.

  If they experimented with threshold 0.5 or 2.5, discuss what they observed. Likely, threshold 0.5 made the neuron say yes for any single reason (maybe surprisingly frequent pizza nights), while threshold 2.5 was so strict that it required a perfect storm of conditions to get pizza. Both are extremes; threshold 1.5 was a balanced choice. This mirrors real-world AI tuning: too lenient vs. too strict, and finding the right balance.

- **Wrap-up:** Conclude that this scenario showed how one AI model (our LTU) can be adjusted to fit different situations or preferences. The family had to decide what their criteria were, and then they implemented that in the neuron. This reflection helps students see AI as configurable and not one-size-fits-all - aligning with the idea that ***humans set the goals/criteria for AI*** *(Big Idea 5: Societal Impact, in the sense that human choices affect outcomes).* They realize that they, as the model designers, had control over how the neuron makes decisions, and that choice can be made differently by different people. This is a powerful insight as they finish the guided scenarios.

At this stage, students have completed a series of scaffolded activities covering various function and decision types (AND, OR, majority, unanimous, etc.) through fun scenarios. They have seen how the Plush Neuron can model decisions that involve requirements, conditions, votes, and combinations of factors. They are now ready to apply all these concepts in a creative way in the culminating project.

Culminating Collaborative Project

To solidify understanding, students will engage in a summative collaborative project where they **design and solve their own neural decision problem**. This project serves as a capstone, encouraging creativity, teamwork, and the application of all five Big Ideas in a student-generated context. Students can expand upon problems covered in the instructional activities, choose a completely new one, or combine elements from multiple scenarios.

(For example, a group might propose: "Can we have PB&Js for dinner? We must have the ingredients to make PB&Js ***and*** *meet the same criteria from the burger voting problem." They might realize this would actually require two neurons to solve - one to decide if PB&J is possible, and another to incorporate that into a family vote - an interesting complexity!)*

Note: Some student-designed activities may require more than one neuron. This necessitates staggered/scheduled demonstrations (if 2 neurons are available) or students could mock up an LTU using crafting material and simulate the presence of a "pre-decided" LTU; e.g. "*Our group is going to demonstrate the decision to get a classroom pet and what that pet should be. We've used construction paper cutouts with icons indicating the needed inputs and weights to surpass a threshold leading to the decision of getting a turtle for the classroom. We will use the plush neuron to "sense" this information by setting a weight of 1 on dendrite 1. This sets a positive weight for the first input of the pet question: 'the class will only get a pet if the class can agree on which pet to get.'"*

Project Overview: "Design Your Own Neural Decision Problem"

Students as AI Designers: In small teams of 3-4, students will create an original scenario (a yes/no or multi-option decision) that can be modeled with one or more Plush Neurons working together. They will then build the neural model and demonstrate how it solves the problem. In essence, each group becomes the author of a new AI problem: they *decide the context*, *define the inputs*, *determine the desired output logi*c, and then *use the Plush Neuron(s)* to implement and test their weight inputs, just like they did in the guided examples throughout the curriculum.

Guidelines for Students:

- **Choose a Scenario:** Think of a relatable decision or classification problem in your life or imagination. It could be practical or playful. For example: "Should our class have an outdoor field day tomorrow?", "Which pet should we adopt - a cat or a dog?", or "Can a robot vacuum this room now or later?" The scenario should have a clear outcome (yes/no, or perhaps A/B choice) that depends on multiple conditions or factors. Encourage fun and creativity! The more it connects to their interests, the better. **Teachers might suggest students revisit the Neuron Sandbox or provide some prompts or examples if students struggle to come up with ideas**.

- **Identify Important Inputs & Desired Outcomes:** Once a scenario is chosen, students decide on the key factors (inputs and weights) that will affect the decision. They should pick approximately 3 significant inputs (since one Plush Neuron handles three easily, though they can have more by combining neurons for a bonus, if they feel ready). For each input, they define what 1 and 0 mean (e.g., "Homework all done? 1=yes, 0=no"). Next, they should articulate how these factors combine to determine the outcome. Essentially, they outline the decision rule in words or a truth table: **for each combination of inputs, what should the decision (output) be?** This step is like writing the "specification" for their mini AI.
- For example, in the field day scenario:
  - Inputs: Weather Good?, All Permission Slips In?, Everyone Finished Homework? and they might decide "Yes to field day" happens only if at least two of those are yes, or maybe "only if weather is good **AND** slips are in" regardless of homework, etc. They should discuss and agree on the logic that makes sense for their problem.
- **Map the Scenario to the Neuron(s):** Students then translate their rule into the LTU model. If their scenario can be handled by a single neuron (3 inputs or fewer, and a single output logic), great - they proceed with one Plush Neuron. If not - for instance, perhaps they thought of 4 or 5 factors, or a decision that isn't simply yes/no - they can consider using two neurons in a simple network.
  - **This is an advanced option:** for example, if deciding between two different outcomes (like cat vs. dog in a pet choice), they might use one neuron to decide "Should we get a pet at all?" and a second neuron to decide "if yes, cat or dog" based on different inputs. Or they might split factors between two neurons whose outputs feed into a final decision (this starts to introduce the idea of multiple neurons working together, aligning with Big Idea 2 & 3 on a slightly higher level). They won't be making anything like a complex multi-layer network - just a simple combination, given their grade level. Teachers should guide them to keep it simple (e.g., at most two neurons chained or parallel).
- **Build and Test the Model:** Using their Plush Neuron device(s), teams configure the weights and threshold according to their plan. They should create a truth table or list of test cases (input combinations) that covers the important situations for their scenario. Then, they run those tests: set the inputs for a test case, observe the neuron's output, and check if it matches the intended outcome in their design; if only one plush neuron is available, students can demo one part of the scenario and then quickly reset to demo the other. If something doesn't match, they engage in a bit of debugging: maybe the threshold needs adjusting, or perhaps they realize one factor should carry more weight. They refine their model until the neuron's behavior aligns with their expected decision rule for all (or most) cases. This is an exercise in iterative design and mirrors how AI models are tuned. **Instructors should ensure their learners understand the purpose and use of the**

fractional threshold. Only the threshold dial increments in half-number values. This ensures that whole-number dendrite inputs will excede the threshold if intended. Remember, the inputs must be GREATER THAN the threshold, not greater than or equal to.

- **Present and Reflect:** Each group will prepare a short presentation of their scenario and solution. They should explain the context of their decision problem, what inputs they chose and why, how these were weighted, and describe the function they implemented (e.g., "Our neuron outputs yes for field day in all cases except when both weather is bad **AND** not everyone did homework"). They will then demonstrate the Plush Neuron in action with a couple of example input combinations to show it working. During the presentation (or as a written summary), they should also reflect on how their model relates to real-world AI: For instance, *"This neuron is like an AI that helps make school event decisions based on conditions"* or *"We gave more weight to safety (weather) than fun (homework done) - similar to how a real AI might prioritize safety factors."* They should mention any challenges they encountered and how they solved them (e.g., "We initially set threshold to 2, but realized that allowed an unsafe field day, so we changed it to 2.5."). The instructor should check with each group to ensure they have all elements necessary for a good presentation. If time allows, students should lead their peers through their scenarios as if they were the instructor; instructors should be available to supervise this.

After all groups have presented, facilitate a brief class discussion to connect their projects to the broader themes:

- **Big Idea 1 (Perception):** their neurons took in "perceptions" (be it weather, personal preferences, etc.) as inputs set on the dendrites.
- **Big Idea 2 (Representation & Reasoning):** They created representations of logic with weights/thresholds and had the neuron reason with them.
- **Big Idea 3 (Learning):** Though they directly set the weight and threshold, they effectively trained their neuron by tweaking it until it behaved as desired, analogous to how an AI learns.
- **Big Idea 4 (Natural Interaction):** Many projects likely involved human-oriented decisions (family, school, etc.), which is AI interacting in human contexts. They also practiced explaining the AI's decision, which is part of making AI's interaction understandable.
- **Big Idea 5 (Societal Impact):** Ask if any group thought about fairness or consequences (e.g., "Was your rule fair to everyone involved?" "What if your AI made a wrong call - what could happen?"). This encourages reflection on the impact of automated decisions.

If any time remains, you might encourage students to rapidly write a narrative or story that combines the decisions from all Plush Neurons and then act out this narrative using props.

This culminating project not only reinforces the technical concepts learned but also empowers students to be creators of AI models, not just users. They get to apply creativity and personal relevance, which increases engagement and retention of the material.

Conclusion

Through these relatable, hands-on activities with the Plush Neuron, 8th grade students explore how simple AI models make decisions. By physically configuring a *Linear Threshold Unit* to solve everyday problems, AI becomes less mysterious and more tangible. This constructionist approach builds conceptual understanding of AI fundamentals and supports the AI4K12 learning goals - students engage with perception (dials as inputs), reasoning (logic through weights and thresholds), learning (iteratively adjusting models), natural interaction (explaining AI decisions in human terms), and societal impact (considering fairness and rules). Students finish the unit with excitement and confidence, having just played with AI. They have reflected on what it means to "teach" an AI a rule, experienced how parameter changes alter outcomes, and drawn parallels between their mini-models and real-world AI systems. The hope is that this curriculum not only imparts knowledge about neural networks and decision-making, but also fosters an AI mindset - a curiosity and critical thinking about intelligent systems. Equipped with this foundational experience, students will be better prepared to engage with the AI-driven world around them as informed creators, users, and citizen-technologists.

References

Touretzky, D. S., Gardner-McCune, C., Martin, F., & Seehorn, D. (2019). Envisioning AI for K-12: What should every child know about AI? The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)

Touretzky, D. S., & Gardner-McCune, C. (2021). Chapter 8: Artificial intelligence thinking in K–12. In S.-C. Kong & H. Abelson (Eds.), *Computational thinking education in K–12:* Artificial intelligence literacy and physical computing (pp. 153–180). MIT Press.

Velander, J., Mohammed, T. A., Otero, N., & Milrad, M. (2024). Artificial intelligence in K–12 education: Eliciting and reflecting on Swedish teachers' understanding of AI and its implications for teaching and learning. *Education and Information Technologies, 29*(5), 4085–4105. https://doi.org/10.1007/s10639-023-11990-4

Touretzky, David & Chen, Angela & Pawar, Neel. (2024). Neural Networks in Middle School. ACM Inroads. 15. 24-28. 10.1145/3677610.

Touretzky, D. S., Gardner-McCune, C., Cox, B., Uchidiuno, J., Yu, X., Gelder, W., McKlin, T., Brown, T., Kareem, B., Chung, W., Jones, A., & Kolodner, J. (2025). Artificial intelligence education in Georgia middle schools. *AI Magazine, 46*(2). https://doi.org/10.1002/aaai.70013