Building Artificial Neural Networks with Arduinos

A Curriculum Unit for High School Biology

and AP Biology Classes

CENTER for NEUROTECHNOLOGY

a National Science Foundation Engineering Research Center

Research Experience for Teachers (RET) Program

October 2017

(Revised November 2019)

Table of Contents

Teacher Resource 4.2a: Input Layer Neuron Sketch

Teacher Resource 4.2b: Hidden Layer Neuron Sketch

Teacher Resource 4.2c: Output Layer Neuron Sketch

Teacher Resource 4.3: “Programming” an ANN—Teacher Answer Key

About the RET Program & the CNT

About the Research Experience for Teachers (RET) Program
The Research Experience for Teachers (RET) program is a seven week research experience for middle and high school STEM teachers, hosted by the Center for Neurotechnology (CNT) on the University of Washington's Seattle campus. Each summer cohort is selected through a competitive application process. Accepted teachers work in a CNT lab alongside a team of researchers conducting cutting-edge neural engineering research. They enhance their understanding of lab safety, bioethics, engineering education, and curriculum design. Together, the teachers work to develop innovative neural engineering curriculum materials, which are then pilot-tested in their own classrooms the following academic year.

About the Center for Neurotechnology (CNT)
The CNT's mission is to develop innovative neural devices and methods for directing engineered neuroplasticity in the brain and spinal cord, which will improve sensory and motor function for people with spinal cord injury, stroke and other neurological disorders. Engineered neuroplasticity is a new form of rehabilitation that uses engineered devices to restore lost or injured connections in the brain, spinal cord and other areas of the nervous system.

Neural Engineering Skill Sets
The CNT has identified the following skill sets as essential for students to achieve neural engineering competency. All education activities supported by the CNT are designed to teach one or more of these skills.

1. **Fundamentals of neuroscience, neural engineering, and neuroethics research:** Knowledge of core concepts in neuroscience and neural engineering, designing and conducting experiments, analysis and interpretation of results, problem solving, understanding primary scientific literature, building scientific knowledge, and ethical and responsible conduct of research.
2. **Neural engineering best practices:** Oral and written communication of neural engineering knowledge and research, confidence, working independently, working on a team, participating in a learning community, innovation, and persistence.
3. **Connections to neural engineering industry and careers:** Awareness of career options in neural engineering and pathways

Funding
The Research Experience for Teachers program is supported by National Science Foundation Award EEC-1028725.

Contact Information & Credits

Program Contact Information:
Kristen Bergsman, Ph.C.
Engineering Education Research Manager, CNT
University of Washington
Phone: 206-221-1494
Email: bergsman@uw.edu

Eric H. Chudler, Ph.D.
CNT Executive Director & Education Co-Director
University of Washington
Phone: 206-616-6899
Email: chudler@uw.edu

CNT Address: Bill & Melinda Gates Center for Computer Science & Engineering; 3800 E Stevens Way NE, Seattle, WA 98195

CNT Website: http://centerforneurotech.org

Credits:
Benjamin Hart, Science Teacher, Redmond High School, Redmond, WA
Lawrence Bencivengo Jr., Science Teacher, Mercer Island High School, Mercer Island, WA
Kristen Bergsman, Center for Neurotechnology

Acknowledgements:
We acknowledge the support of the following individuals: Rajesh Rao, PhD; Eric Chudler, PhD; Chet Moritz, PhD; Howard Chizeck, PhD; Lise Johnson, PhD; Sara Goering, PhD; Laura Specker Sullivan, PhD; Hale Soloff; Deb Harper; Josh Patrick; Mary Guiden; and Daphne Struck.

Unit Description

Artificial Neural Network Project Summary

Artificial Neural Networks (ANN) belong to a branch of Artificial Intelligence (AI) which seeks to create computers capable of learning by simulating how biological brains work. The thought is that brains have evolved to be excellent at learning new things, so why reinvent the wheel? Brains are especially good at recognizing patterns, such as faces in a crowd, which is something that has proven notoriously difficult to teach computers to do using traditional computer programming techniques.

In a living organism, neurons are the cells which process information in the brain. Each living neuron constantly receives signals form hundreds or thousands of other neurons. Some signals are excitatory, pushing the neuron to "fire" and generate its own signal which it sends on to other nerve cells in the brain. Some signals are inhibitory, preventing the cell from firing. All these incoming signals add together and will either make the neuron fire or not. The collection of cells that are firing at any given moment in our brains causes our thoughts, memories, feelings and actions!

Our ANN simulates this firing activity using microcontrollers called Arduinos to simulate neurons. Each simulated neuron receives signals from other neurons and adds up all the incoming signals. If the total is larger than the neuron's "threshold", it will "fire" and send a signal to other simulated neurons, otherwise it remains quiet. The ANN is capable of processing information in a manner very similar to the biological neural networks in our brains.

Real brains can learn by changing the number and strength of synapses - the structures where one neuron physically communicates with another neuron. Our ANN can simulate this, too, because each input is multiplied by a weight which can be changed, thus changing the likelihood that a simulated neuron will fire. ANNs can be trained to learn to recognize different inputs by presenting the network with Learning Examples which should produce a specific known output by the network. Scientists can adjust the weights that each simulated neuron uses in order to gradually adjust the network's output and correct any mistakes that it makes on the training examples. By presenting each training example hundreds or thousands of times to the ANN, and adjusting each of the many weights by a small amount using statistical techniques, the ANN gradually learns to correctly identify each input pattern.

Real ANNs, containing thousands or even tens of thousands of simulated neurons, are capable of learning quite complex tasks, and are currently being employed in face recognition software, Google's self-driving cars, and even in search engines that can sift through the vast amount of data contained in people's search queries. AP Bio's simple ANN will obviously not be capable of such feats, but it will operate using the same basic principles and hopefully help students to develop not only a basic understanding of ANNs, but also a deeper appreciation for how their own brains work!

In this unit, students will explore the applications of artificial neural networks, especially in the field of artificial intelligence. Students will learn about the history of artificial intelligence, explore the concept of neural networks through activities and computer simulation, and then construct a simple, three-level artificial neural network using Arduinos to simulate neurons. After building the network, they will be challenged to discover how altering the connections or programming of the “neurons” alters the behavior of the network. Finally, students will explore the ethical implications of building artificially intelligent machines.

Learning Outcomes

Unit-level learning outcomes are presented in this section. Each lesson plan highlights the learning outcomes aligned to the particular activities incorporated into that lesson.

Big Ideas & Enduring Understandings (aligned to AP Biology)

- **Living systems store, retrieve, transmit, and respond to information essential to life processes:** Cells communicate by generating, transmitting and receiving chemical signals. Transmission of information results in changes within and between biological systems
- **Biological systems interact, and these systems and their interactions possess complex properties:** Interactions within biological systems lead to complex properties.

Essential Questions:

- How do complex systems give rise to emergent properties?
- What is intelligence? Is human intelligence measurable by a single “Intelligence Quotient”?
- How do we *see*? How do the eye and brain process visual information?
- How intelligent are other creatures? Do any other animals possess a human-like level of intelligence?
- Can machines “think”? Is it possible to build an artificially intelligent machine? Would it be conscious in the way a person is? What are the ethical implications of building artificially intelligent machines?

Knowledge and Skills (Outcomes):

Students will know…

- Intelligence is the ability to learn and adapt one's behavior so that it is better able to accomplish specific goals.
- All living things and many artificial systems possess "intelligence" in this limited sense – some may possess intelligence that rivals or is equivalent to that of humans.
- In the field of artificial intelligence, neural networks have been used to explore the nature of intelligence and machine learning, as well as to simulate and study the way biological organisms process information.
- "Knowledge" in an artificial neural network is represented by the overall behavior of the network, *not* by individual neurons.

Students will be able to…

- Analyze the connectivity of a system, identifying nodes, edges, hubs and loops.
- Explain about the different layers (Input, Hidden and Output) of an artificial neural network, how they are connected and how the system is able to "learn."
- Program and assemble simple circuits controlled by Arduinos.
- Assemble several Arduinos programmed to simulate artificial neurons into a functional artificial neural network capable of demonstrating simple behaviors that simulate the function of the retina.
- Draw a flowchart of the programming logic.

Alignment to AP Biology Learning Outcomes

- **LO 3.33** The student is able to use representation(s) and appropriate models to describe features of a cell signaling pathway.
- **LO 3.34** The student is able to construct explanations of cell communication through cell-to-cell direct contact or through chemical signaling.
- **LO 3.35** The student is able to create representation(s) that depict how cell-to-cell communication occurs by direct contact or from a distance through chemical signaling.
- **LO 3.36** The student is able to describe a model that expresses the key elements of signal transduction pathways by which a signal is converted to a cellular response.
- **LO 3.38** The student is able to describe a model that expresses key elements to show how change in signal transduction can alter cellular response.
- **LO 3.41** The student is able to create a representation that describes how organisms exchange information in response to internal changes and external cues, and which can result in changes in behavior
- **LO 3.43** The student is able to construct an explanation, based on scientific theories and models, about how nervous systems detect external and internal signals, transmit and integrate information, and produce responses.

- **LO 3.44** The student is able to describe how nervous systems detect external and internal signals.
- **LO 3.45** The student is able to describe how nervous systems transmit information.
- **LO 3.46** The student is able to describe how the vertebrate brain integrates information to produce a response.
- **LO 3.47** The student is able to create a visual representation of complex nervous systems to describe/explain how these systems detect external and internal signals, transmit and integrate information, and produce responses.
- **LO 3.49** The student is able to create a visual representation to describe how nervous systems transmit information.
- **LO 4.8** The student is able to evaluate scientific questions concerning organisms that exhibit complex properties due to the interaction of their constituent parts.
- **LO 4.9** The student is able to predict the effects of a change in a component(s) of a biological system on the functionality of an organism(s).

Knowledge and Skills (Prerequisite):

Note: It is highly recommended that prior to teaching this unit, instructors facilitate the unit *Introduction to Neural Engineering & Ethical Implications* from the 2015 Research Experience for Teachers program at the Center for Neurotechnology. This unit provides an introduction to a broad range of neural engineering topics, including the human nervous system, electrophysiology, history of neural engineering, medical devices, and ethical implications of these emerging technologies. It can be downloaded for free from http://centerforneurotech.org.

Helpful prerequisite knowledge includes:

- The structure and function of a neuron, including general neuron types (sensory, motor and interneurons), dendrites, axons, cell bodies, and synapses.
- Neurons communicate with each other using a combination of electrical impulses and chemical signals. A synapse is the junction between two neurons where electrical signals are transformed into a chemical signal received by the post-synaptic cell.
- Electrical signals in the nervous system are produced by changes in concentrations of ions between the inside and outside of the cell membrane.
- The nervous system takes in information about the body and the environment through various senses, processes, and stores that information, and then generates responses in the form of behaviors or altered bodily functions.
- The retina of the eye contains light-sensitive cells called photoreceptors that detect the visual image. The retina develops from an extension of the brain, and contains several layers of neurons that process incoming visual stimuli before sending the results to the brain for further analysis.
- Various parts of the human brain are devoted to processing visual information, especially the primary visual cortex (V1) in the back of the brain and nearby secondary

areas that separately process different aspects of vision, such as form, location, color and motion. No one knows how the brain puts all that information back together to create the visual images we "see."

Helpful prerequisite skills include:

- General knowledge of computers including turning on, opening programs, and typing with a keyboard.

Key Vocabulary:

- Intelligence and Artificial Intelligence (AI)
- Machine Learning
- Neuron, synapse, sensory neuron, motor neuron, interneuron, neural network
- Artificial Neural Network (ANN)
- Node, edge, hub, loop
- Arduino
- Voltage, amperage, resistance, breadboard
- Resistor, photoresistor, pin, jumper wire, digital, analog, Light Emitting Diode (LED)
- Programing language: variable identification, void setup (), void manual tune (), void autotune (), digitalWrite, analogWrite, If … else… statements, void loop (), comments (// or /* and */), curly brackets, and Boolean logic.

Alignment to National Learning Standards

This unit is aligned to the Next Generation Science Standards (NGSS) and the Common Core State Standards (CCSS) in English Language Arts. Alignment to NGSS Performance Expectations and the three dimensions of science and engineering education (Disciplinary Core Ideas, Crosscutting Concepts, and Practices) are outlined in the tables below.

Next Generation Science Standards: High School (Grades 9-12)

	1: Introduction to AI	2: Introducing ANNs	3: Introducing Arduinos	4: Building Neural Networks
Engineering Design				
HS-ETS1-1 Engineering Design: Analyze a major global challenge to specify qualitative and quantitative criteria and constraints for solutions that account for societal needs and wants.				
HS-ETS1-2 Engineering Design: Design a solution to a complex real-world problem by breaking it down into smaller, more manageable problems that can be solved through engineering.				
HS-ETS1-3 Engineering Design: Evaluate a solution to a complex real-world problem based on prioritized criteria and trade-offs that account for a range of constraints, including cost, safety, reliability, and aesthetics as well as possible social, cultural, and environmental impacts.				
HS-ETS1-4 Engineering Design: Use a computer simulation to model the impact of proposed solutions to a complex real-world problem with numerous criteria and constraints on interactions within and between systems relevant to the problem.				
Life Sciences				
HS-LS1-1 Structure and Function: Construct an explanation based on evidence for how the structure of DNA determines the structure of proteins which carry out the essential functions of life through systems of specialized cells.				
HS-LS1-2 Structure and Function: Develop and use a model to illustrate the hierarchical organization of interacting systems that provide specific functions within multicellular organisms.				

Physical Sciences				
HS-PS3-3 Energy: Design, build, and refine a device that works within given constraints to convert one form of energy into another form of energy.				

NGSS Lead States. (2013). *Next Generation Science Standards: For States, By States*. Achieve, Inc. on behalf of the twenty-six states and partners that collaborated on the NGSS.

NGSS Cross–Cutting Concepts:

	1: Introduction to AI	2: Introducing ANNs	3: Introducing Arduinos	4: Building Neural Networks
Patterns: Observed patterns of forms and events guide organization and classification, and they prompt questions about relationships and the factors that influence them.				
Cause and Effect—Mechanism and Explanation: Events have causes, sometimes simple, sometimes multifaceted. A major activity of science is investigating and explaining causal relationships and the mechanisms by which they are mediated. Such mechanisms can then be tested across given contexts and used to predict and explain events in new contexts.				
Systems and System Models: Defining the system under study—specifying its boundaries and making explicit a model of that system—provides tools for understanding and testing ideas that are applicable throughout science and engineering.				
Structure and Function: The way in which an object or living thing is shaped and its substructure determine many of its properties and functions.				
Stability and Change: For natural and built systems alike, conditions of stability and determinants of rates of change or evolution of a system are critical elements of study.				

NGSS Lead States. (2013). *Next Generation Science Standards: For States, By States*. Achieve, Inc. on behalf of the twenty-six states and partners that collaborated on the NGSS.

Lesson One: Introduction to AI

Center for Neurotechnology
Lesson Plan Authors: Benjamin Hart, Redmond High School and Lawrence Bencivengo Jr., Mercer Island High School

LESSON OVERVIEW

Activity Time: One 50 minute class period. Additional time for homework.

Lesson Plan Summary:
In this lesson, students will be introduced Artificial Intelligence (AI) and neural networks by watching a short movie clip from *Transcendence.* After showing the movie clip, students will discuss questions in groups and as a class. Finally, students will read an article about the history of AI and neural networks.

STUDENT UNDERSTANDINGS

Big Idea & Enduring Understanding:

- **Artificial Intelligence and Neural Networks:** Students gain the basic background knowledge of AI and neural networks.

Essential Question:

- What is intelligence?

Learning Objectives:
Students will know…

- Intelligence is the ability to learn and adapt one's behavior so that it is better able to accomplish specific goals.
- All living things and many artificial systems possess "intelligence" in this limited sense – some may possess intelligence that rivals or is equivalent to that of humans.
- AI can be utilized to solve real world problems.

Students will be able to…

- Describe the development of AI and neural networks.
- Identify several applications of AI.

Vocabulary:

- Artificial Intelligence (AI)
- Brain
- Engineering
- Intelligence
- Nervous system
- Neural engineering
- Neural network
- Neuron

Standards Alignment: This lesson addresses the following high school Next Generation Science Standards (NGSS).

NGSS High School Disciplinary Core Ideas

- **HS-ETS1-1 Engineering Design:** Analyze a major global challenge to specify qualitative and quantitative criteria and constraints for solutions that account for societal needs and wants.

NGSS Cross-Cutting Concepts

- **Patterns:** Observed patterns of forms and events guide organization and classification, and they prompt questions about relationships and the factors that influence them.

NGSS Science & Engineering Practices

- Obtaining, evaluating and communicating information.

MATERIALS

Material	Description	Quantity
Science Notebooks	Students' science notebooks or journals	1 per student
Media Center	Classroom media equipment including computer with internet connection, speakers, and projector, for showing video clips.	1 media center

Transcendence (or *Her)* video clip	Introduces AI and neural networks. See *Teacher Preparation* section for URLs.	2 short video clips
Student Handout 1.1	*The Three Breakthroughs That Have Finally Unleashed AI on the World—Reading Questions*	1 copy per student
AI article from Wired.com	"The Three Breakthroughs That Have Finally Unleashed AI on the World". Access and print from www.wired.com/2014/10/future-of-artificial-intelligence/	1 copy per student
Teacher Resource 1.1	*The Three Breakthroughs That Have Finally Unleashed AI on the World—Reading Questions Answer Key*	1

TEACHER PREPARATION

1. Watch the clips of the 2014 film *Transcendence.* This film is rated PG-13: Please note that these clips do not depict violence. However, other clips of the movie do contain violence.

    a. Clip #1 Artificial Intelligence (0:59 minutes) https://www.youtube.com/watch?v=h38ihmHnXWs&index=7&list=PLwC4e-jE85Bq5ObC8O_chs41H104D4LFn
    b. Clip #2 Neural Networks (0:49 minutes) https://www.youtube.com/watch?v=l_QOR0fq4SQ&index=2&list=PLwC4e-jE85Bq5ObC8O_chs41H104D4LF

2. Read "The Three Breakthroughs that have Finally Unleashed AI on the World" from Wired.com that provides background information on neural networks and AI.

3. Brainstorm answers to questions that will be asked of students.

4. Make copies of *Student Handout 1.1* and the Wired.com article, one of each per student.

5. Preview *Teacher Resource 1.1*.

PROCEDURE

Engage: Artificial Intelligence and neural networks in popular culture (5 minutes)

1. Ask students "What is intelligence? What is artificial intelligence?" Students will record notes by defining these terms in their own words.

2. Show the first movie clip from *Transcendence* on Artificial Intelligence (AI).
   a. This clip reveals to students the potential power of AI.
   b. Students turn-and-talk with their neighbor to discuss how one could construct AI.

3. Introduce the second clip by saying to create AI, many scientists use a neural network.
   a. Show the second movie clip from *Transcendence* on neural networks.
   b. This clip shows the interrelationship between neural networks and AI.

Explore: Artificial intelligence (20 minutes)

4. Assign students a partner to work with.

5. Ask students to "Identify applications of Artificial Intelligence." Give time for students to first record their answers in their science notebooks and then discuss their ideas with their partners. Ask students to share out their responses to the two questions above and record their answers at the front of the classroom on a whiteboard to create a public record.

6. Ask, "Are there any devices that currently exist which exhibit artificial intelligence?" Students first record their thoughts in their science notebooks and then talk with their neighbor to discuss the above question. Record answers of a few group pairs on the whiteboard.

7. Discuss the following questions as a class: "What is the nature of consciousness and can a machine be truly intelligent without being conscious?" Again, allow time for students to write in their science notebooks and then record student answers as a public record.

Explain: Artificial intelligence (5 minutes)

8. As a class, ask students to define the neuron and nervous system. Then ask students to define engineering. Use student definitions to define neural engineering. Note that neural engineering is an engineering discipline that uses concepts from math and science to connect to and interact with the nervous system.

9. Brainstorm as a class the applications of artificial intelligence to the field of neural engineering. Ask students, “How could AI help those with sensory or motor deficits?”

Elaborate: Students begin homework in class (10 minutes)

10. Hand out copies of the Wired.com article and *Student Handout 1.1*.

11. Homework: Read the Wired.com article, “The Three Breakthroughs That Have Finally Unleashed AI on the World,” about the history of artificial intelligence (AI) and neural networks. Answer the questions on *Student Handout 1.1.*

Evaluate (carried over to *Lesson Two*):

12. After reading the article, students will return to class to review key points from the homework (see *Lesson Two*).

STUDENT ASSESSMENT

Assessment Opportunities: Student knowledge, skills, and concepts for this lesson will be assessed in a number of ways.

- First, students will participate in both small group and class discussions. During these discussions, the teacher can evaluate student thinking.
- Second, science notebooks can be reviewed to gain insight into students’ initial understandings at the beginning of the unit.
- Third, the homework assignment allows further assessment of student knowledge of artificial intelligence and neural networks and skills related to Common Core State Standards in English Language Arts (see *Standards Alignment* above).

Student Metacognition:

- At the start of the lesson, students will be asked “What is ‘intelligence’?” and “What is artificial intelligence?” Later, as students complete the worksheet, they will be asked “Do you agree with the author’s definition of intelligence?” (4b). Therefore, students will have structured opportunities for metacognition.

Scoring Guide:

- If students are successful, they will meet learning objectives stated above. Evidence needed would be from student answers during the group and class discussions and from homework assignment. See *Teacher Resource 1.1* for an answer key to *Student Handout 1.1*.

EXTENSION ACTIVITIES

Extension Activities:

- Students could further research neural networks and how these networks give rise to learning.

Adaptations:

- For ELL students, a list of keywords from the article and their definitions could be provided.
- For SPED or younger students, the teacher can assign select paragraphs of the article rather than the full reading.
- For gifted or older students, the teacher could ask students to read an article that is more technical in nature explaining how AI works. Recommended reading:
  - "The Rise of the Machines: Artificial Intelligence Scares People Excessively So" from *The Economist:* http://www.economist.com/news/briefing/21650526-artificial-intelligence-scares-peopleexcessively-so-rise-machines

TEACHER BACKGROUND & RESOURCES

Background Information:

See the following article: "The Rise of the Machines: Artificial Intelligence Scares People Excessively So" from *The Economist.* Available at http://www.economist.com/news/briefing/21650526-artificial-intelligence-scares-peopleexcessively-so-rise-machines

Resources:

It is important to have a strong understanding of the nervous system. Therefore, review college-level biology textbook description of the nervous system and intelligence that arises from the nervous system before teaching the lesson.

Citations:

Kelly, K. (n.d.). The Three Breakthroughs That Have Finally Unleashed AI on the World. *Wired*. Retrieved from http://www.wired.com/2014/10/future-of-artificial-intelligence/

Pfister, W., Depp, J., Freeman, M., Hall, R., & Warner Bros. Entertainment Canada (Firm). (2014). *Transcendence*.

(2015, May 9). “The Rise of the Machines: Artificial Intelligence Scares People--Excessively So.” *The Economist*. Retrieved from http://www.economist.com/news/briefing/21650526-artificial-intelligence-scares-peopleexcessively-so-rise-machines

Student Handout 1.1: The Three Breakthroughs That Have Finally Unleashed AI on the World—Reading Questions

Name:_________________________________ Date:__________________ Period:__________

1. List the three recent breakthroughs that have unleashed the long awaited arrival of artificial intelligence:

____

____

____

2. How is a neural network like a brain? Be specific.

3. According to the article list two applications of AI.

____

____

4. The author states "In the next 10 years, 99 percent of the artificial intelligence that you will interact with, directly or indirectly, will be nerdily autistic, supersmart specialists. In fact, this won't really be intelligence, at least not as we've come to think of it."

a. How does the author then define intelligence?

b. Do you agree with the author's definition of intelligence? Why?

5. What device would you like to put AI into? How would this help you?

Teacher Resource 1.1: The Three Breakthroughs That Have Finally Unleashed AI on the World—Reading Questions Answer Key

1. List the three recent breakthroughs that have unleashed the long awaited arrival of artificial intelligence:
    1. Cheap parallel computation

    2. Big data

    3. Better algorithms

2. How is a neural network like a brain? Be specific.
A neural network is like a brain because they both do processing of information in parallel. To accomplish this parallel processing, each node of a neural network is like a neuron and can process and communicate data to other nodes, similar to how neurons function in the brain.

3. According to the article list two applications of AI.
Any of the following are possible: Self-driving car, body tracker, personal photo archive, universal translator, smarter newsfeed, games, diagnosing patients, law, education, sorting videos.

4. The author states "In the next 10 years, 99 percent of the artificial intelligence that you will interact with, directly or indirectly, will be nerdily autistic, supersmart specialists. In fact, this won't really be intelligence, at least not as we've come to think of it."
   a. How does the author then define intelligence at this point in the article?
   The author defines intelligence as "our peculiar self-awareness, all our frantic loops of introspection and messy currents of self-consciousness."

   b. Do you agree with the author's definition of intelligence?
   Answers will vary.

5. What device would you like to put AI into? How would this help you?
Students should identify at least one device and explain why AI would be a helpful addition to them personally.

Lesson Two: Introducing Artificial Neural Networks (ANNs)

Center for Neurotechnology

Lesson Plan Authors: Benjamin Hart, Redmond High School and Lawrence Bencivengo Jr., Mercer Island High School

(Adapted from Teaching Engineering's "It's a Connected World" and 2013 Research Experience for Teachers Curriculum Unit "Traumatic Brain Injury: A Neural Network Journey")

LESSON OVERVIEW

Activity Time: One 50 minute class period.

Lesson Plan Summary: In this lesson, students will build upon their learning in *Lesson One*. First, students will review the material from the reading about artificial neural networks (ANN) and artificial intelligence. Second, student volunteers will model a simple neural network. Third, students will use paper and pencil models to analyze the behavior of simple ANNs. Fourth, discuss as a class the similarities between these three-layer ANNs and the organization of neurons in the retina.

STUDENT UNDERSTANDINGS

Big Idea & Enduring Understanding:

- **Artificial Intelligence (AI) and Neural Networks:** Students gain the foundational knowledge of AI and neural networks.

Essential Question:

- How do artificial neural networks and the nervous system process and transmit information?

Learning Objectives:

Students will know…

- A conceptual understanding of artificial neural networks.
- How both an ANN and the nervous system process information. This processing of information can lead to intelligence.

Students will be able to…

- Define "intelligence."
- Identify components of a network, such as a node, edge, and degrees of a node.

- Model a simple neural network.
- Construct analogies between an ANN and the retina.

Vocabulary:
- Artificial neural network (ANN)
- Complex network
- Degree of a node
- Edge
- Node

Standards Alignment: This lesson addresses the following high school Next Generation Science Standards (NGSS).

NGSS High School Disciplinary Core Ideas
- **HS-ETS1-1 Engineering Design:** Analyze a major global challenge to specify qualitative and quantitative criteria and constraints for solutions that account for societal needs and wants.
- **HS-LS1-1 Structure and Function:** Construct an explanation based on evidence for how the structure of DNA determines the structure of proteins which carry out the essential functions of life through systems of specialized cells.
- **HS-LS1-2 Structure and Function:** Develop and use a model to illustrate the hierarchical organization of interacting systems that provide specific functions within multicellular organisms.

NGSS Cross-Cutting Concepts
- **Structure and Function:** The way in which an object or living thing is shaped and its substructure determine many of its properties and functions.

NGSS Science & Engineering Practices
- Developing and using models.

MATERIALS

Material	Description	Quantity
Science Notebooks	Students' science notebooks or journals	1 per student
Media Center	Classroom media equipment including computer with internet connection, speakers, and projector, if you will be showing a PowerPoint presentation.	1 media center
Lesson One's AI article from Wired Magazine	Article: *"*The Three Breakthroughs That Have Finally Unleashed AI on the World"	1 copy per student from *Lesson One*
Lesson One's ***Teacher Resource 1.1***	Answer Key for Student Questions on the Wired magazine article.	1
Student Handout 2.1: *Complex Networks*	Worksheet to introduce concepts related to complex networks.	1 per student
Teacher Resource 2.1: *Complex Networks Teacher Answer Key*	Answer key for scoring ***Student Handout 2.1***.	1

TEACHER PREPARATION

1. Review ***Teacher Resource 1.1*** from *Lesson One.*

2. Make copies of ***Student Handout 2.1***, one per student.

PROCEDURE

Engage: Complex Networks All Around Us (15 minutes)

1. Discuss with the class:

   "We all live in a connected world. Many, if not all, of us have cell phones capable of sending signals to nearby cellular towers, which can be bounced all over the world to potentially reach other people with phones. Likewise, we have access to computers, which can connect to other computers all over the world using the internet. While these networked systems of interconnected components are triumphs of modern engineering, nature has been producing large and complex networks for millions of years."

2. Ask if anyone can think of other examples of complex networks besides cellular network and the internet. You may consider recording students' ideas on the whiteboard or poster paper. Some possible answers are provided in the table below, with the nodes and edges defined to help understand how the network is structured. If 'neural network' isn't brought up by the students, be sure to add this to the list.

Network	Nodes	Edges
Cellular network	Phones	Cellular towers
Internet network	Computers	Satellites, routers, Ethernet cables
Biochemical network	Molecules	Chemical reactions
Epidemiological network	Healthy and infected individuals	Infectious contacts
Trophic network	Predators and prey	Predation interactions
Power grid	Electrical generators and end users	Power lines, substations, and transformers
Professional network	Colleagues and collaborators	Collaborations
Social network	People	Friendships
Neural network	Neurons	Synaptic connections

Explore: Modeling an Artificial Neural Network (5 minutes)

3. Create a physical model of an Artificial Neural Network using student volunteers. Have nine student volunteers line up in three rows of three. Tell students the three rows comprise the three layers of a neural network. The first row is the **input layer**, the second row is the **hidden layer**, and the third row is the **output layer**. How will the input layer send a message to the output layer?

4. Instruct students to place their hands on the shoulders of the people in other rows.

5. Explain that in this simulation, neural signals are simulated through the squeezing of the shoulder of one student by another.
    - The **input layer** can only send a signal to the hidden layer.
    - The **hidden layer** can only send a signal to the output layer.
    - The **output layer** can only receive messages; they cannot send messages. When someone at the hidden layer squeezes the shoulder of someone at the output layer, that person will then raise their hand to show that the signal was received.

6. Start the simulation by squeezing the shoulders of one or more of the students at the input layer, so that the signal gets sent through the hidden layer to be received by students at the output layer.

7. After completing the simulation, ask students to share their thinking about the model. Ask students to draw a diagram of the simulation in their science notebooks.

Explore and Elaborate: Network Nodes and Edges (15 minutes)

8. Introduce students to the concept of a network. It is important that students become familiar with the following terms and definitions:

    - **Node:** A point of intersection, a connection point.
    - **Edge:** Lines that connect nodes.
    - **Degree of a node:** The number of edges connecting to the node.

9. Distribute copies of ***Student Handout 2.1***, one per student. Allow time for students to work individually or in pairs to complete the handout.

10. Review Examples #1, #2, and #3 from ***Student Handout 2.1*** with students. ***Teacher Resource 2.1*** provides an answer key for the handout.

Evaluate: Review of AI Reading (5 minutes)

11. Lead students through a review of ***Student Handout 1.1:*** *The Three Breakthroughs That Have Finally Unleashed AI on the World*. See *Lesson One's* ***Teacher Handout 1.1*** for an answer key.

STUDENT ASSESSMENT

Assessment Opportunities:

- Students will demonstrate their knowledge of artificial intelligence through the homework assignment from *Lesson One*.
- Students should be able to construct models of ANNs and logically predict an outcome given rules and inputs. In addition, students should be able to determine inputs given outputs and rules. Finally, given inputs and outputs, students should be able to develop rules that would enable a functional network. Questions of students may be asked during the lesson that assesses these logic skills.
- The conceptual understanding of artificial networks will be assessed through questioning of inputs, outputs, and rules that could be used to regulate the flow of information. This conceptual understanding can be assessed during the class volunteer modeling of an ANN.

- ***Teacher Resources 1.1*** and ***2.1*** provide answer keys to be used for scoring ***Student Handouts 1.1*** and ***2.1.***

Scoring Guide:

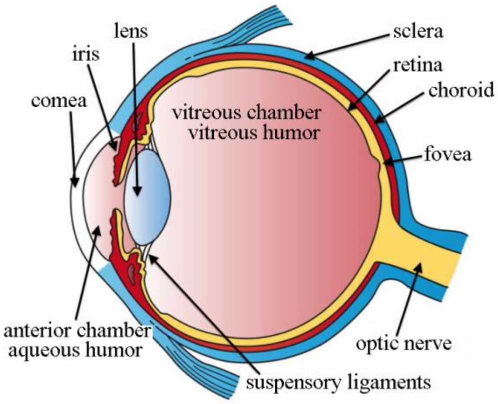
- ***Teacher Resource 1.1*** provides an answer key to be used for scoring ***Student Handout 1.1.***
- ***Teacher Resource 2.1*** provides an answer key to be used for scoring ***Student Handout 2.1.***

EXTENSION ACTIVITIES

Extension Activities:

- This activity could be extended by giving students more network problems. These problems could involve more complex systems that have a greater number of nodes organized into more layers. The teacher could then ask questions about inputs, outputs, or rules that govern how information is processed.

- Teacher asks students to draw analogies between the neural network and the layers of neurons in the retina. In the retina, the rods and cones receive light and send this light information to bipolar, amacrine, and ganglion cells. Eventually this information is sent to the optic nerve.



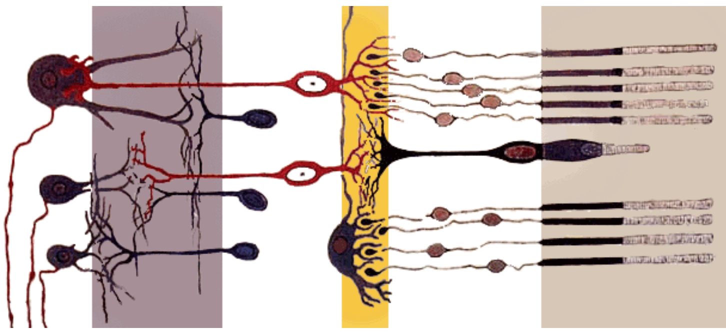


Figure on the **left** shows the anatomy of the human eye, including the location of the retina and optic nerve. Figure on the **right** shows the rods, cones, and nerve layers in the retina. The front (anterior) of the eye is on the left. Light (from the left) passes through several transparent nerve layers to reach the rods and cones (far right). A chemical change in the rods and cones send a signal back to the nerves. The signal goes first to the bipolar and horizontal cells (yellow layer), then to the amacrine cells and ganglion cells (purple layer), then to the optic nerve fibers. The signals are processed in these layers. First, the signals start as raw outputs of points in the rod and cone cells. Then the nerve layers identify simple shapes, such as bright points surrounded by dark points, edges, and movement. (Based on a drawing by Ramón y Cajal, 1911.) Source: Wikipedia.

Adaptations:

- To meet the needs of ELL and SPED students, the vocabulary terms could be presented with physical objects or pictures. In addition, pairing of students such that one student could help another understand the concepts and develop the necessary skills to be successful.
- To meet the needs of gifted students, challenge problems could be presented to them (see *Extension Activities* above).

TEACHER BACKGROUND & RESOURCES

Background Information:

To teach this unit, the teacher needs to familiarize themselves with the following terms: (1) artificial neural networks, (2) retina, (3) node, and (4) edge. The teacher should also draw the network students will demonstrate. Then the teacher can examine how various inputs and rules will lead to a certain output. Furthermore, the teacher should review all answers to *Student Handout 2.1*. Finally, if the teacher chooses to continue with extension suggestions, then the teacher should prepare by reviewing the layers of the retina and compare those layers to a neural network.

Resources:

- ***It's a Connected World*** **Curriculum Unit**
  Teaching Engineering
  https://www.teachengineering.org/view_curricularunit.php?url=collection/jhu_/curricular_units/jhu_cnetworks_unit.xml

- ***Traumatic Brain Injury: A Neural Network Journey*** **Curriculum Unit**
  2013 Research Experience for Teachers Program
  Center for Neurotechnology
  http://www.centerforneurotech.org/content/lesson-plans

Citations:

Shaw, M., & Jephson-Hernandez, S. (2013). "Lesson 5: Complex Networks and Graphs." In *Traumatic Brain Injury: A Neural Network Journey*, 2013 Research Experience for Teachers Neural Engineering Lessons and Resources. Retrieved from http://csne-erc.org/sites/default/files/Networks%202013%20lessons.pdf

Teach Engineering Curriculum for K-12 Teachers. Retrieved from https://www.teachengineering.org/index.phphttps://www.teachengineering.org/index.php

"Three Main Layers of the Eye" by Artwork by Holly Fischer - http://open.umich.edu/education/med/resources/second-look-series/materials - Eye Slide 3. Licensed under CC BY 3.0 via Commons - https://commons.wikimedia.org/wiki/File:Three_Main_Layers_of_the_Eye.png#/media/File:Three_Main_Layers_of_the_Eye.png

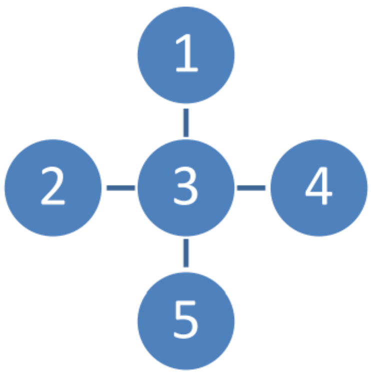
"Retina-diagram" by Fig_retine.png: Cajalderivative work Fig retine bended.png: Anka Friedrich (talk)derivative work: vectorisation by chris 論 - Fig_retine.pngFig retine bended.png. Licensed under CC BY-SA 3.0 via Commons - https://commons.wikimedia.org/wiki/File:Retina-diagram.svg#/media/File:Retina-diagram.svg

Student Handout 2.1: Complex Networks

Name:________________________________ Date:__________________ Period:__________

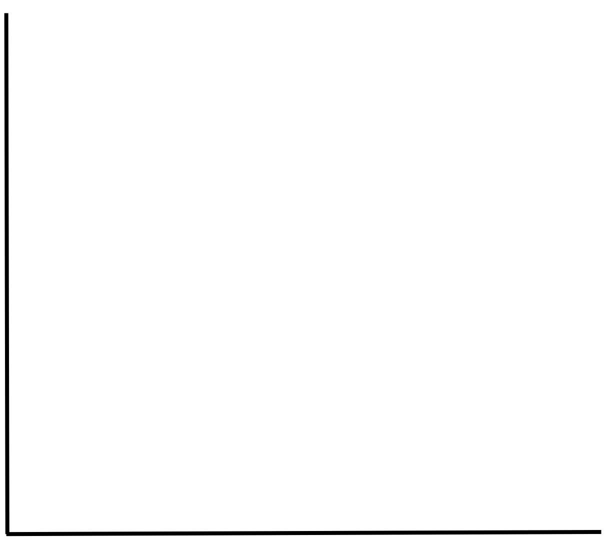
Example 1

1. Examine the network diagram below.



Node	# of Edges
1	
2	
3	
4	
5	

2. Draw and label a bar graph of the Node vs Edges. Which is the dependent variable? Which is the independent variable? (Remember, the dependent variable goes on the y-axis and the independent variable goes on the x-axis.)



Example 1 Questions

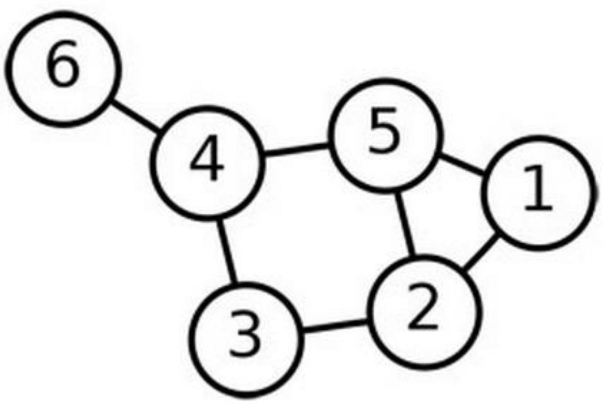
3. How many nodes are there in the diagram?

4. Which node(s) has the highest degree (degree= # of edges)?

5. Which node(s) has the lowest degree?

Example 2

6. Examine the network diagram below.



Node	Degrees of a Node
1	
2	
3	
4	
5	
6	

7. Draw and label a bar graph of the Node vs Edges.

Example 2 Questions:

8. Which node is the most important? Why?

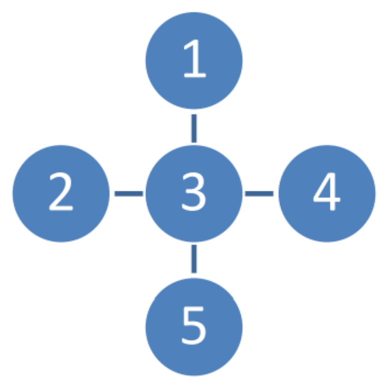
9. Which node is the second most important? Why?

10. Which node is the least important? Why?

Example 3

S.I.R. model of neural propagation: Remember signals are sent from the cell body of one neuron, through the axon. Neurotransmitters are released from the axon terminal and these transmitters bind to receptors on the dendrites or cell bodies of the neighboring, postsynaptic neuron. The binding of these neurotransmitters can lead to the postsynaptic neuron depolarizing and sending another message (excitatory). Conversely, binding of neurotransmitters can lead to the postsynaptic neuron becoming hyperpolarized, preventing a signal from being sent (inhibitory).

In the diagram below, nodes are analogous to the cell bodies of neurons and edges from a node are analogous to an axon. Also note S means Susceptible to Firing, I means In process of Firing, and R means Recovering (i.e. cannot fire while recovering).



Choose a neuron (in a specific region) with a single edge as a start. At time, t=0, Starting neuron is I, in process of firing. All other neurons are susceptible to firing. The interaction between neurons is excitatory.

For Example:

	Time (seconds)						
Node	**t = 0**	**t = 1**	**t = 2**	**t = 3**	**t = 4**	**t = 5**	**t = 6**
1	**S**	**I**	**R**	**S**	**S**	**S**	**S**
2	**S**	**S**	**S**	**I**	**I**	**R**	**S**
3	**S**	**S**	**I**	**R**	**S**	**S**	**S**
4	**S**	**S**	**S**	**I**	**I**	**R**	**S**
5	**S**	**S**	**S**	**I**	**I**	**R**	**S**

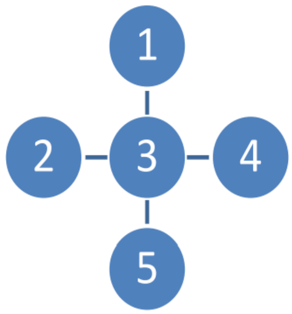
11. If Node 3 was excited first, which neurons would be excited next?

12. Would an Action Potential fire if a second impulse excited Node 3 at Time=3 seconds? Explain.

Teacher Resource 2.1: Complex Networks—Teacher Answer Key

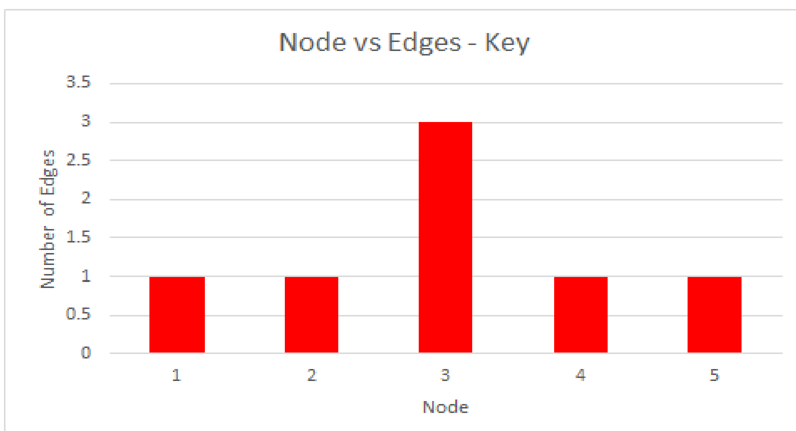
Example 1

1. Examine the network diagram below.



Node	# of Edges
1	1
2	1
3	4
4	1
5	1

2. Draw and label a bar graph of the Node vs Edges. Which is the dependent variable? Which is the independent variable? (Remember, the dependent variable goes on the y-axis and the independent variable goes on the x-axis.)



Example 1 Questions

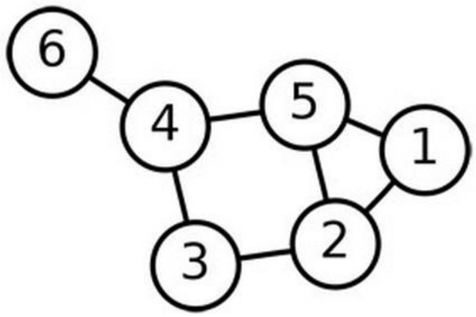
3. How many nodes are there in the diagram?
   There are five nodes in the diagram.

4. Which node(s) has the highest degree (degree=# of edges)?
   Node 3 has the highest degree.

5. Which node(s) has the lowest degree?
   Nodes 1, 2, 4, and 5 all have one degree.

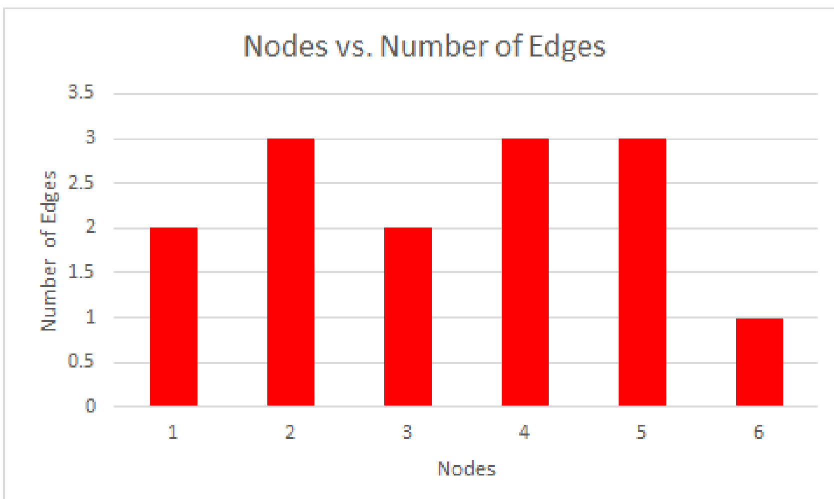
Example 2

6. Examine the network diagram below.



Node	Degrees of a Node
1	2
2	3
3	2
4	3
5	3
6	1

7. Draw and label a bar graph of the Node vs Edges.



Example 2 Questions:

8. Which node is the most important? Why?
   Arguably Node 4 since it has three edges and is the only one to connect to Node 6; However, answers may vary.

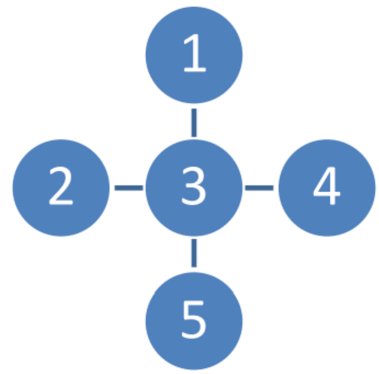
9. Which node is the second most important? Why?
   Answers will vary.

10. Which node is the least important? Why?
    Arguably Node 6 is least important because it has only one edge.

Example 3

S.I.R. model of neural propagation: Remember signals are sent from the cell body of one neuron, through the axon. Neurotransmitters are released from the axon terminal and these transmitters bind to receptors on the dendrites or cell bodies of the neighboring, postsynaptic neuron. The binding of these neurotransmitters can lead to the postsynaptic neuron depolarizing and sending another message (excitatory). Conversely, binding of neurotransmitters can lead to the postsynaptic neuron becoming hyperpolarized, preventing a signal from being sent (inhibitory).

In the diagram below, nodes are analogous to the cell bodies of neurons and edges from a node are analogous to an axon. Also note S means Susceptible to Firing, I means In process of Firing, and R means Recovering (i.e. cannot fire while recovering).



Choose a neuron (in a specific region) with a single edge as a start. At time, t=0, Starting neuron is I, in process of firing. All other neurons are susceptible to firing. The interaction between neurons is excitatory.

For Example:

	Time (seconds)						
Node	**t = 0**	**t = 1**	**t = 2**	**t = 3**	**t = 4**	**t = 5**	**t = 6**
1	**S**	**I**	**R**	**S**	**S**	**S**	**S**
2	**S**	**S**	**S**	**I**	**I**	**R**	**S**
3	**S**	**S**	**I**	**R**	**S**	**S**	**S**
4	**S**	**S**	**S**	**I**	**I**	**R**	**S**
5	**S**	**S**	**S**	**I**	**I**	**R**	**S**

11. If Node 3 was excited first, which neurons would be excited next?
    Node 3 would excite Nodes 1, 2, 4, and 5 next.

12. Would an Action Potential fire if a second impulse excited Node 3 at Time=3 seconds? Explain.
    No. At 3 seconds, Node 3 cannot fire because it is recovering. (The neuron is hyperpolarized.)

Lesson Three: Introducing Arduinos

Center for Neurotechnology

Lesson Plan Authors: Benjamin Hart, Redmond High School and Lawrence Bencivengo Jr., Mercer Island High School

(This lesson plan is a direct adaptation of "Getting to Know the Arduino Uno" from the 2014 Research Experience for Teachers Curriculum Unit, *Introduction to Neural Engineering: Neuroprosthetics & Brain-Computer Interfaces*. The lesson plan was written by Angelica Sauceda, Science Teacher at TAF Academy and Steve Pratt, Science Teacher at Cleveland High School)

LESSON OVERVIEW

Activity Time: Two 50 minute class periods.

Lesson Plan Summary: In this lesson, students will learn how to construct simple circuits with breadboards and controlling these circuits with Arduinos. More specifically, in this lesson students will be introduced the challenge of constructing an actual artificial neural network (ANN) using Arduino controllers and simple circuits built on breadboards. Students will receive direct instruction and watch a video clip introducing the Arduino controllers and electric circuitry. In groups, students will then construct simple breadboard circuits and program the Arduino controller.

STUDENT UNDERSTANDINGS

Big Idea & Enduring Understanding:

- **Systems:** A system is a network of sub-systems that contains inputs, outputs, and feedback mechanisms in order to control/regulate a specific outcome. Disturbances affect the outcome of the system.
- **Circuit:** A circuit consists of a power source, wires, and a resistor/s.

Essential Question:

- How can electronics be utilized to model complex connections between neurons?

Learning Objectives:

Students will know…

- The parts that make up a breadboard.
- A system consists of inputs, outputs, and feedback.
- The Arduino Uno programming language consists of declaration of variables, void setup, and void loop.

Students will be able to…

- Construct simple circuits with the breadboard.
- Run given programs on the Arduino.
- Design and evaluate a systems model for a neural prosthetic that includes a description of the role of the sensors, output mechanism, and possible feedback loops.
- Write and make sense of programming sketches.
- Describe the input, output, and explain if feedback is established.

Vocabulary:

- Amperage
- Arduino Uno
- Breadboard
- Circuit
- Input
- LED
- Motor
- Output
- Pin
- Resistor
- Photoresistor
- Resistance
- Sketch
- System
- Voltage

Standards Alignment: This lesson addresses the following high school Next Generation Science Standards (NGSS).

NGSS High School Disciplinary Core Ideas

- **HS-PS3-3 Energy:** Design, build, and refine a device that works within given constraints to convert one form of energy into another form of energy.
- **HS-ETS1-1 Engineering Design:** Analyze a major global challenge to specify qualitative and quantitative criteria and constraints for solutions that account for societal needs and wants.
- **HS-ETS1-2 Engineering Design:** Design a solution to a complex real-world problem by breaking it down into smaller, more manageable problems that can be solved through engineering.

- **HS-ETS1-3 Engineering Design:** Evaluate a solution to a complex real-world problem based on prioritized criteria and trade-offs that account for a range of constraints, including cost, safety, reliability, and aesthetics, as well as possible social, cultural, and environmental impacts.
- **HS-ETS1-4 Engineering Design:** Use a computer simulation to model the impact of proposed solutions to a complex real-world problem with numerous criteria and constraints on interactions within and between systems relevant to the problem.

NGSS Crosscutting Concepts

- **Patterns:** Observed patterns of forms and events guide organization and classification, and they prompt questions about relationships and the factors that influence them.
- **Systems and System Models:** Defining the system under study—specifying its boundaries and making explicit a model of that system—provides tools for understanding and testing ideas that are applicable throughout science and engineering.
- **Structure and Function:** The way in which an object or living thing is shaped and its substructure determine many of its properties and functions.

NGSS Science & Engineering Practices

- Developing and Using Models

MATERIALS

Material	Description	Quantity
Laptop or computer	Will be used for programming the Arduino Uno software. Arduino Uno software must be installed on each computer. Available from http://www.ArduinoUno.cc/en/Main/Software	1 per group
Classroom computer	Must have with speakers and internet connection, connected to projector.	1 per class
Arduino Uno Ultimate Starter Kit	Includes Arduino, electrical materials, and breadboard. Available from http://www.amazon.com/Arduino-Ultimate-Starter-page-Instruction/dp/B00BT0NDB8/ref=sr_1_2?s=pc&ie=UTF8&qid=1439317216&sr=1-2&keywords=arduino+kit	1 per group

Materials for light bulb challenge	Wire, 1.5 V battery, and a mini-light bulb incandescent or LED	1 set per group or pair of students
Battery	9V Battery	1 per group
USB cable	AmazonBasics Hi-Speed USB 2.0 A-Male to B-Male Cable, 6 feet. Available from http://www.amazon.com/AmazonBasics-Hi-Speed--Male-B-Male-Cable/dp/B001TH7GUA/ref=sr_1_1?s=electronics&ie=UTF8&qid=1407518205&sr=1-1&keywords=usb+cable	1 per group
Wire cutters/strippers.	Optional, only needed if students need to cut and strip their wires.	As needed
Resistor	Resistor, anything between 1 k – 10 k Ohms. Available from any electronics supplier.	1 per group
Mini push button switch	Available from https://www.sparkfun.com/products/97	1 per group
Student Handout 3.1	*Day One Exit Ticket*	1 per student
Student Handout 3.2	*Day Two Exit Ticket*	1 per student
Teacher Resource 3.1	*Arduino Sketches*	1
Teacher Resource 3.2: Circuits and Arduino Uno Slide Deck	Download from http://centerforneurotech.org/building-artificial-neural-networks	1 PPT file

TEACHER PREPARATION

1. Make copies of ***Student Handouts 3.1*** and ***3.2***, one per student.

2. Prepare a bin for each group containing all of the materials needed for each part of this lesson (this will help manage each part of the lesson smoothly). Each group will need: 1 Arduino Uno, USB cable, a LED, a resistor, a breadboard, a button, 3 wires, along with a laptop or desktop computer.

3. Familiarize yourself with all the materials listed above.

4. Download the Arduino Uno software on all computers that will be used by students.
    a. Go to https://www.arduino.cc/en/main/software
    b. Select the installer for your computer's operating system.

5. Preview the videos included in the PowerPoint presentation.

6. Familiarize yourself with the pHet simulation. It can be used during the PPT presentation by clicking on the Circuit Construction link on the slide for students who do not have any background with circuits.

    Circuit Construction Kit (DC Only) Interactive Simulation
    http://phet.colorado.edu/en/simulation/circuit-construction-kit-dc

7. **To be best prepared for work with the Arduino Uno, it is recommended that the teacher either take time to work through the "Getting Started with Arduino Uno" pdf guide OR watch a few of Jeremy Blum's Arduino Uno tutorials on YouTube (see the *Teacher Background & Resources* section for the links).** Both are helpful for learning the basics of connecting the hardware as well as programming within the Arduino Uno environment.

PROCEDURE

DAY ONE

Engage/Explore (15 minutes)

1. Display the PPT Presentation. Introduce Arduinos with *Slide #1* and present the question "How can electronics be utilized to model complex questions between neurons?" Then briefly show *Slide #2.*

2. Show challenge *Slide #3* as a warm-up that asks students to illuminate the light bulb using a piece of wire, 1.5 V battery, and a mini-light bulb incandescent or LED—just be careful to watch the voltage requirements for certain light bulbs. Hand out the required materials and allow time for students to work on the challenge in small groups. If students complete the challenge, ask them to make observations and explain why their design is a closed circuit.

Explain (15 minutes)

3. Teach the components of a closed circuit using *Slide #4* and referencing the designs students did at the beginning of class.

4. Use *Slides #5* and *#6* to check for understanding.

5. Use *Slides #7-9* to teach students about current. *Slide #7* will give you the definition and *Slides #8* and *#9* gives context to the definitions.

6. Show *Slide #10*. Ask the class the following questions:
    a. What would happen if we removed the light bulb from your circuit and connected just the wires to both terminals of the battery?
    b. What would happen if a wire was connected between the battery and the bulb?

7. Follow the hyperlink on *Slide #10* to show the "Introduction to Breadboard" video. Before showing the video, preview the video discussion questions on *Slide #11*. The purpose of this is to show primarily the setup of the breadboard so that students can create closed circuits on a breadboard.

    Introduction to Breadboard (Protoboards)
    Electroninstructor, September 2007, (8:08 minutes)
    https://www.youtube.com/watch?v=oiqNaSPTI7

8. Have students discuss the answers to the Discussion Questions on *Slide #11* in their groups, and then share out the results

9. Because the purpose of understanding **I vs V vs R** is to understand how to use a circuit board correctly, review the two diagrams on *Slides #13* and *#14* that relate these three concepts to water. Then show how you would use them in a circuit drawing.

10. ***OPTIONAL SLIDES (Slides #14-16):*** These slides offer more detailed definitions of current, resistance, and voltage.

Elaborate (20 minutes)

11. Introduce the Breadboard Challenge on *Slide #17*. Students are to create three functioning closed circuits with the listed materials. The challenge asks them both take a photo for each of the three closed circuits they will create, as well as to draw a circuit diagram for each one. **Tell students that they should *not* hook up an LED to a power source without a resistor because a LED is not a strong enough resistor.**

Evaluate (5 minutes)

12. Check off student work as they finish. Then pass out copies of ***Student Handout 3.1: Day One Exit Ticket*** to assess student knowledge of circuits.

DAY TWO

Note: Students will need their laptops with the Arduino Uno software installed for the Day Two activities.

Explain (25 minutes)

1. Use *Slide #18* to introduce Day Two and the Arduino Uno. Explain that now that students understand the basics of circuits, it is now time to learn how to program the Arduino Uno. The purpose of today's lesson is to learn the basics of programming, practice some code manipulation, and show how systems can be "programmed" with different variables, inputs, and outputs.

2. Distribute the bins with the Arduino Uno kits (Arduino Uno, USB cable, a LED, a resistor, a breadboard, a button, and 3 wires), one bin per group.

3. Show students *Slide #19* which describes each part of the Arduino Uno. Ask students to place a piece of paper underneath their Arduino Uno. Then, they can draw arrows and write labels on the paper for the major components of the Arduino Uno.

4. Have students open up the Arduino Uno program. They should open up the same sketch called "Blink" as shown on *Slide #20* (File → Examples → Basics → Blink). **Note:** Arduino Uno sketches can be copied and pasted from ***Teacher Resource 3.1.***

5. Use *Slide #20* to show students the three main sections of a sketch: declaration, setup, and loop. Remind the students that computers are essentially "stupid" and need to be told explicitly what things are and what they need to do with them. Be sure to point out the different types of declarations (int vs float vs char), the relevant methods (pinMode, etc.) and the point of a loop function. Also show students how a program is read from top-down, with semi-colons serving as "periods" that tell the computer to go to the next line to receive the next direction.

6. Then, use the same *Slide #20* to explain the step-by-step mechanics of the code using the // on the right hand side. Students can write their own comments to describe what each line means. Be sure to tell the students that they will need to do this on future sketches, even though you are modeling what to do on the first one.

7. Have students set up the LED and USB cable connected to the Arduino Uno according to the diagram on *Slide #21*. (Basically, put the short lead of the LED into the GND, the long into pin 13). This diagram is based off the "Getting Started with Arduino Uno" pdf (See *Resources* section below).

Explore/Elaborate (20 minutes)

8. Get students to start on Arduino Challenge #1 using *Slide #22*. Make sure that they identify the system inputs, outputs, and any subsystems, and also state whether there is feedback.

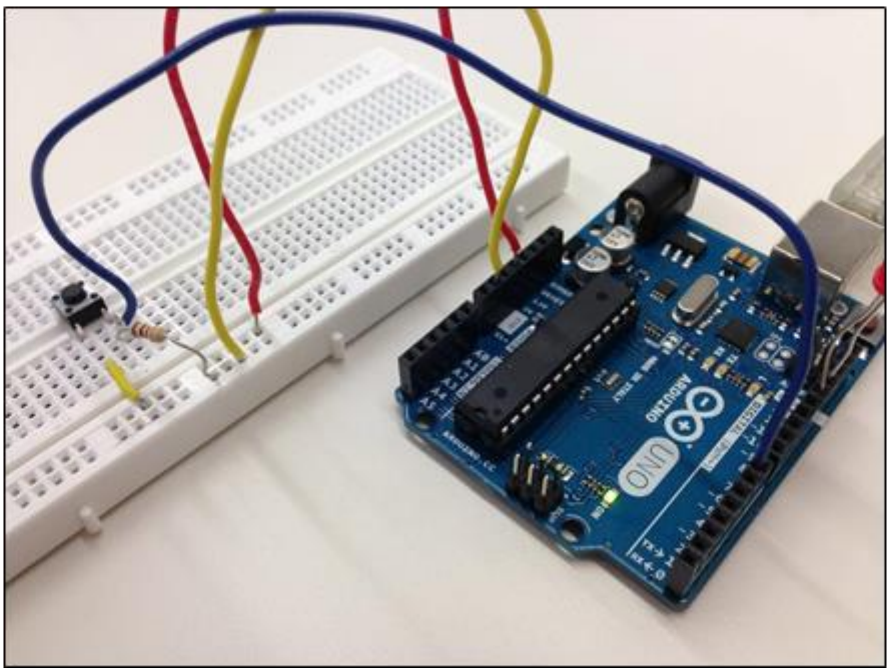
9. When students feel like they are ready to test out their sketch, have them set up a breadboard circuit, upload the sketch to the Arduino Uno, and test it out.

10. Next, students are challenged to change the code in two completely different ways to affect the outcome of the blinking LED. (**Hints:** Time delay duration, on vs off, adding extra lines to make a pattern, etc.) Ask students to document the different ways that they changed the output.

11. When students have completed Arduino Uno Challenge #1, ask them to start Arduino Challenge Uno #2 on *Slide #23* by asking students to copy and paste the “button” sketch into their module (the sketch is included in the ***Teacher Resource 3.1***.

12. Explain the serial.println function, and how a serial monitor works by relaying information that is outputted to a serial screen on their computer. It can output text, numbers, or any kind of information that is inputted into the Arduino Uno.

13. Have students add //comments to each line of code. Then, have students complete Arduino Uno Challenge #2 by setting up a breadboard circuit, uploading the sketch to the Arduino, and testing it out. A photo of the correct connections is included below.



Correct set-up for Arduino Uno Challenge #2

14. Students will need to paste the “fading light” sketch into their module to see the sketch function. Introduce the “for” function, which will cause a light bulb to increase in brightness in small increments and then decrease similarly, like the light on a sleeping Apple computer.

Evaluate (5 minutes)

15. Have students complete ***Student Handout 3.2: Day Two Exit Ticket*** for today's lesson.

EXTENSION ACTIVITIES

Extension Activities:

- For students who need more challenge during the BUTTON sketch or want to go farther, direct them towards the tutorial listed below, specifically (5:49-16:00), where they can save the inputs/outputs associated with the button.

  Tutorial 2 for Arduino: Buttons, PWM, and Functions
  Jeremy Blum, January 2011 (5:49 – 16:00 minutes)
  https://www.youtube.com/watch?v=_LCCGFSMOr4&list=PLA567CE235D39FA84

- Students who need more challenge can also complete Arduino Uno Challenge #3 on *Slide #24*. **Note:** The setup is the *same* as the above photo...the only thing that changes is that one of the LED pins should be in the 9 pin (because 13 is not capable of pulse width modulation signals). See if the students can notice that the LED pin has changed based on the code...don't tell them right away...see if they can see the change!

Adaptations:

- For those lacking background with basic circuits, the pHet simulation will provide an excellent introduction to this information (see *Teacher Preparation*).
- If a student is not comfortable with the programming language, you can provide all the sketches. In addition they can watch Jeremy Blum's Tutorials, provided in the *Background Information* section, for further assistance.
- If a student needs more of a challenge then you can ask them to come up with their own sketches.
- Another way to engage gifted students would be to have them help other groups of students that are struggling through the Arduino challenges.

TEACHER BACKGROUND & RESOURCES

Background Information:

- It will be helpful to familiarize yourself with the Arduino Uno (http://www.arduino.cc/), the Arduino Uno software, and the Arduino Uno programming language by doing one of two things:

  Watch **Jeremy Blum's Arduino Uno Tutorials 1 - 4**
  https://www.youtube.com/results?search_query=jeremy+blum+arduino+tutorial

  Work through the "Getting Started with Arduino, 2nd edition" (Download the free e-book using the following URL http://www.it-ebooks.info/book/1338/ to understand the hardware and programming language.

Resources:

Getting Started with Arduino Uno, 2nd edition
Massimo Banzi, free downloadable e-book
http://it-ebooks.info/book/1338/

Getting Started with Arduino
Step-by-step guide from the Arduino company
http://arduino.cc/en/Guide/HomePage

Arduino Uno Tutorials
Jeremy Blum offers a series of five tutorials on his YouTube channel
https://www.youtube.com/channel/UC4KXPjmKwPutGjwFZsEXB5g

Citations:

Content Credits

Sauceda, A., and Pratt, S. (2014). "Getting to Know the Arduino Uno." In the 2014 Research Experience for Teachers Curriculum Unit, *Introduction to Neural Engineering: Neuroprosthetics & Brain-Computer Interfaces*. Center for Sensorimotor Neural Engineering, University of Washington. http://centerforneurotech.org/education-k-12-lesson-plans/introduction-neural-engineering

PhET Interactive Simulations. PhET Circuit Construction Kit (DC Only). Retrieved from http://phet.colorado.edu/en/simulation/circuit-construction-kit-dc

Banzi, M. (2011). *Getting Started with Arduino, 2nd edition* [PDF]. O'Reilly Media. Retrieved from http://it-ebooks.info/book/1338/

Image Credits

Breadboard Picture on ***Student Handout 9.1*** is from http://www.backward-workshop.com/electronics/breadboard-curriculum/breadboard/

Voltage vs Current vs Resistance Water Diagram is from Sparkfun, https://cdn.sparkfun.com/assets/learn_tutorials/1/9/3/water-analogy.png

Voltage vs Current vs Resistance Diagram is from Wikimedia Commons, http://upload.wikimedia.org/wikipedia/commons/thumb/d/de/OhmsLaw.svg/220px-OhmsLaw.svg.png

Arduino Uno Intro Photo is from Arduino, http://ardunio.cc

Parts of an Arduino Uno Diagram is from Adafruit (original author Nick Gammon), http://www.adafruit.com/blog/2012/05/25/handy-arduino-r3-pinout-diagram/

Student Handout 3.1: Day One Exit Ticket

Name:_________________________________ Date:__________________ Period:__________

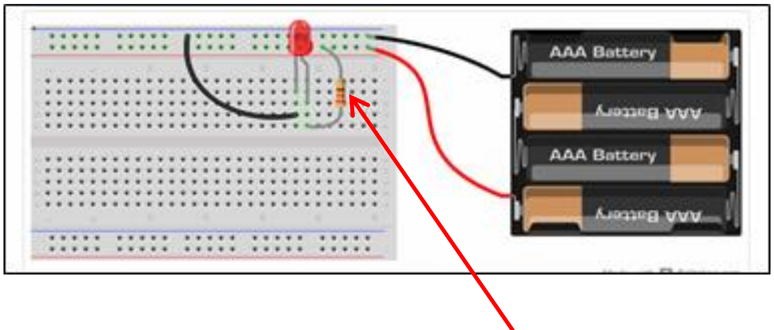
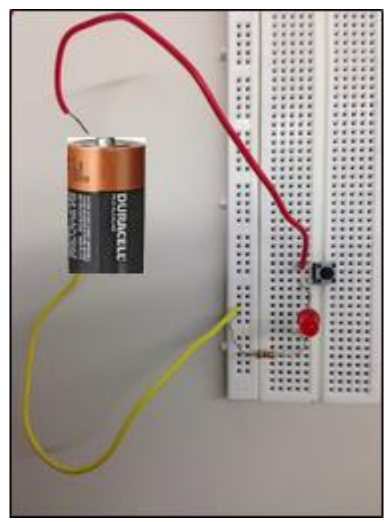


Image credit: www.Backward-workshop.com

1. Name and describe the item indicated by the red arrow in the picture above. What is its role within the circuit and what do you think would happen if it were to be removed?



2. Using the picture to the right, indicate which component represents the ***input*** function. How do you know?

3. Using the picture to the right, indicate which component represents the ***output*** function. How do you know?

Student Handout 3.2: Day Two Exit Ticket

Name:________________________________ Date:__________________ Period:__________

1. Describe the three main sections of an Arduino sketch.

2. How confident would you feel if you were asked to write your own Arduino sketch. Explain.

Teacher Resource 3.1: Sketches

Blinking Light, Challenge #1 (copy and paste into Arduino Uno)

https://www.arduino.cc/en/tutorial/blink

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);              // wait for a second
  digitalWrite(13, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);              // wait for a second
}
```

do nothing for 1000 milliseconds, or one second. When you use the delay() command, nothing else happens for that amount of time. */

Button Sketch, Challenge #2 (copy and paste into Arduino Uno)

```
 //Button Sketch

//Declarations

int buttonPin = 7;
int ledPin = 13;

//Setup to Determine which Pins are Input vs Output

void setup()
{
 Serial.begin(9600);
 pinMode (buttonPin, INPUT);
 pinMode (ledPin, OUTPUT);
}

//Sketch that will keep looping and looping
void loop()
{

 if(digitalRead(buttonPin) == HIGH)
 {
  digitalWrite(ledPin, HIGH);
 }
 else
 {
```

```
  digitalWrite(ledPin, LOW);
 }
}
```

Fading Light Sketch, Challenge #3 (copy and paste into Arduino Uno) - Optional

```
//Fading Light Sketch

//Declarations

int button = 7;
int LED = 9;
int i = 0;


//Setup

void setup()
{
 pinMode(button, INPUT);
 pinMode(LED, OUTPUT);
}

//Loop function

void loop ()
{
 if (digitalRead(button) == HIGH)
 {

  for(i = 0; i < 100; i++)
  {
  analogWrite(LED, i);
  delay(10);
  }

 for (i = 100; i > 0; i--)
  {
  analogWrite(LED, i);
  delay(10);
  }
 }

 else
 {
  digitalWrite(LED, LOW);
 }

}
```

Lesson Three

Circuits, Systems, & the Arduino

Challenge: Make the bulb light up!

-Make observations and explain why it has the setup it does?

- What are the components of a closed circuit and do they need to be connected? How do you know?

Components of a Circuit

- A simple circuit consists of a battery connected to wires and a….
  - Light bulb
  - Motor
  - Buzzer
  - Etc...
- A complex circuit can consist of 100’s to 1000’s of components!

Check for understanding

Which is the correct way to light the lightbulb with the battery?

a) all are correct

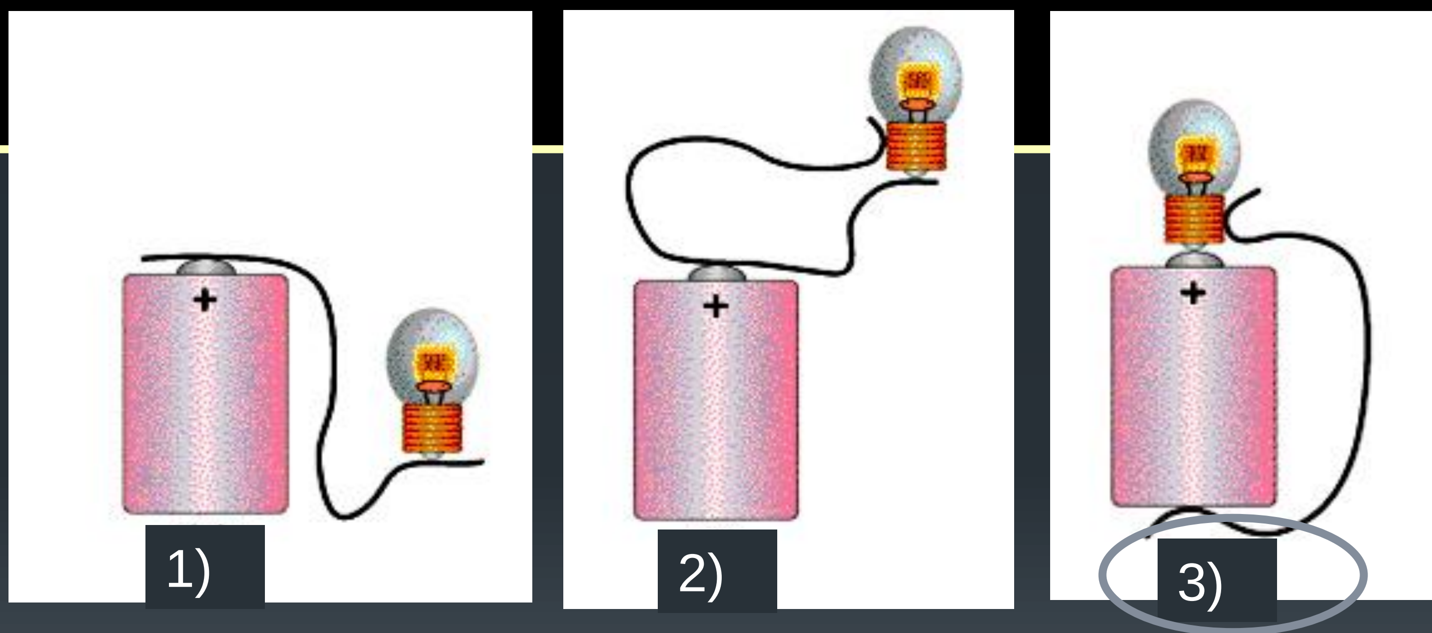
b) none are correct

Connect the Battery

Which is the correct way to light the lightbulb with the battery?

a) all are correct

b) none are correct



Current can only flow if there is a **continuous connection** from the negative terminal through the bulb to the positive terminal. This is only the case for Fig. (3).

<u>Electric Current (I)</u> is the flow of electric charge. More specifically, it is defined as the _rate_ at which charge flows.

Current's Effect on the Human Body (according to OSHA… for a 1 sec. duration)

From: http://www.osha.gov/SLTC/etools/construction/electrical_incidents/eleccurrent.html

Current in Amps	Effect on Human Body
1 mA (0.001 A)	**Perception level. Slight tingling sensation.**
5 mA (0.005 A)	**Slight shock felt; not painful but disturbing. Average individual can let go.**
6-16 mA	**Painful shock, begin to lose muscular control. Commonly referred to as the "let-go" range.**
17-99 mA	**Extreme pain, respiratory arrest, severe muscular contractions. Individual cannot let go. Death is possible.**
100-2000 mA	**Ventricular fibrillation (uneven, uncoordinated pumping of the heart.) Muscular contraction and nerve damage begins to occur. Death is likely.**
> 2000 mA (2 A)	**Cardiac arrest, internal organ damage, and severe burns. Death is probable.**

Other Interesting Currents…

- **Defibrillator**

  35 A!!

- **Police Taser**

  162 mA

In the previous activity, you created a closed circuit.

- What would happen if we removed the light bulb from your circuit and connected just the wires to both terminals of the battery?
- Circuit Construction Kit
- Turn and talk why did this happen?

In the previous activity, you created a circuit.

- What would happen if I added a wire that separates the bulb from the battery?

- [Circuit Construction Kit]

- Turn and talk why did this happen?

Introduction to a Breadboard

- Breadboard Introduction

Discussion Questions

1. How can you ensure that you have created a closed circuit on the breadboard?

2. Be specific, how does the woman on the video connect the components on the breadboard?

3. Why do you think she uses a resistor in her circuit?

Current vs Voltage vs Resistance

A metaphor using water

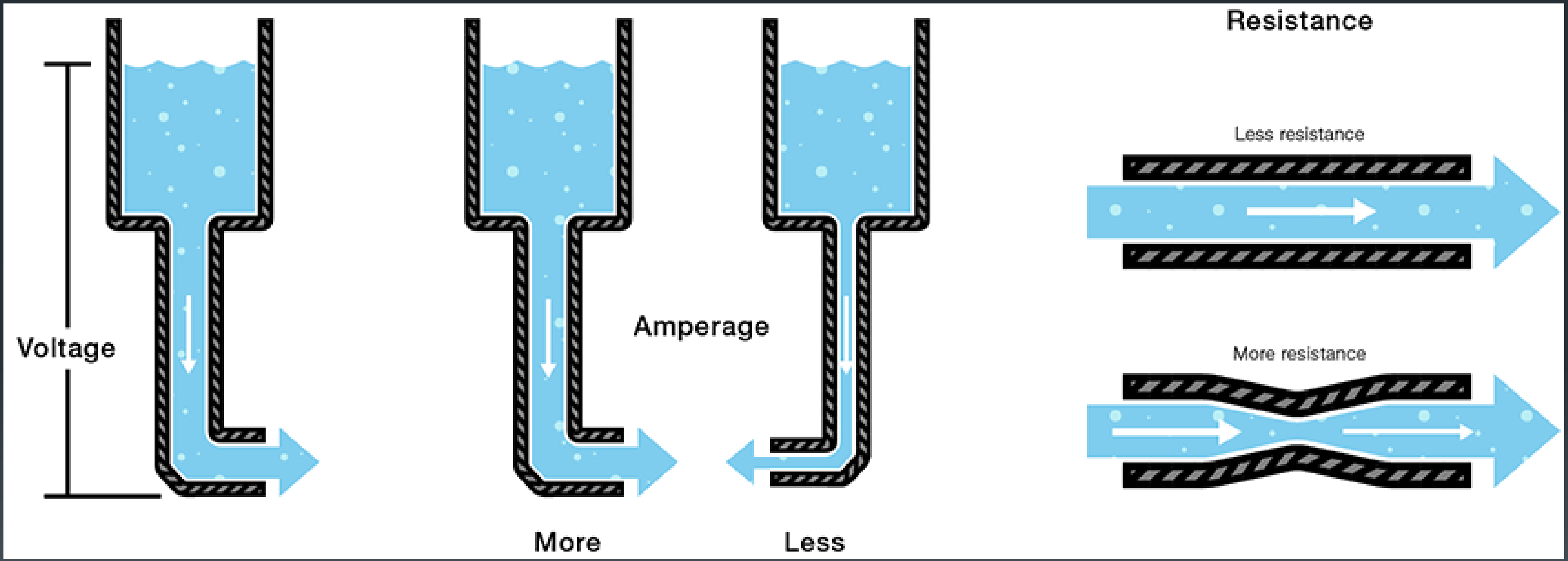


Image credit: Sparkfun, https://cdn.sparkfun.com/assets/learn_tutorials/1/9/3/water-analogy.png

Current vs Voltage vs Resistance

How to think about a circuit using a **CIRCUIT DIAGRAM**

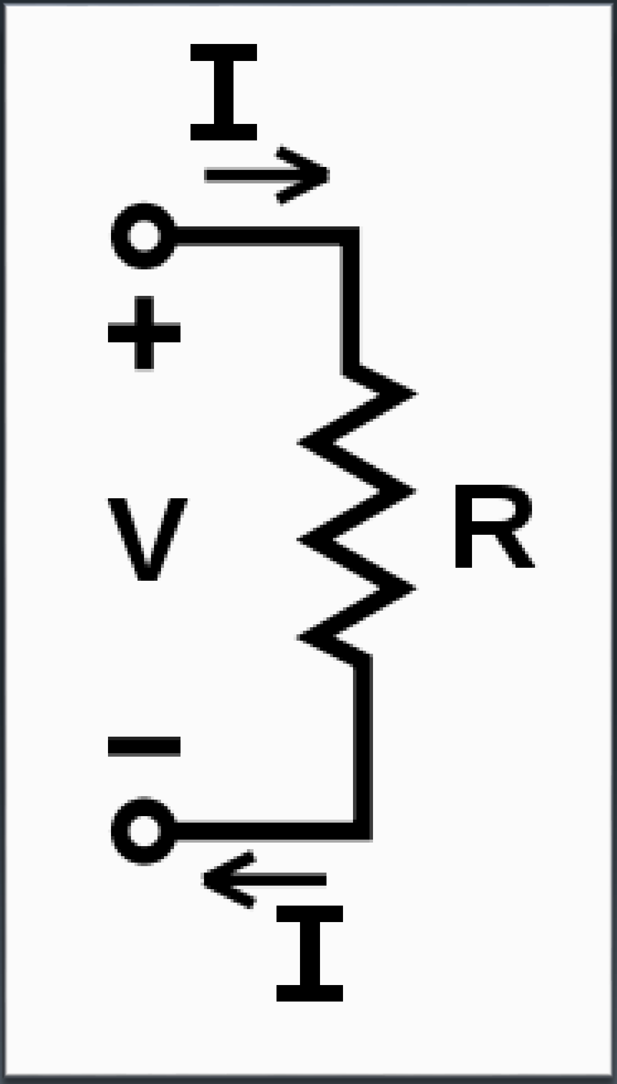


Image credit: Wikimedia Commons

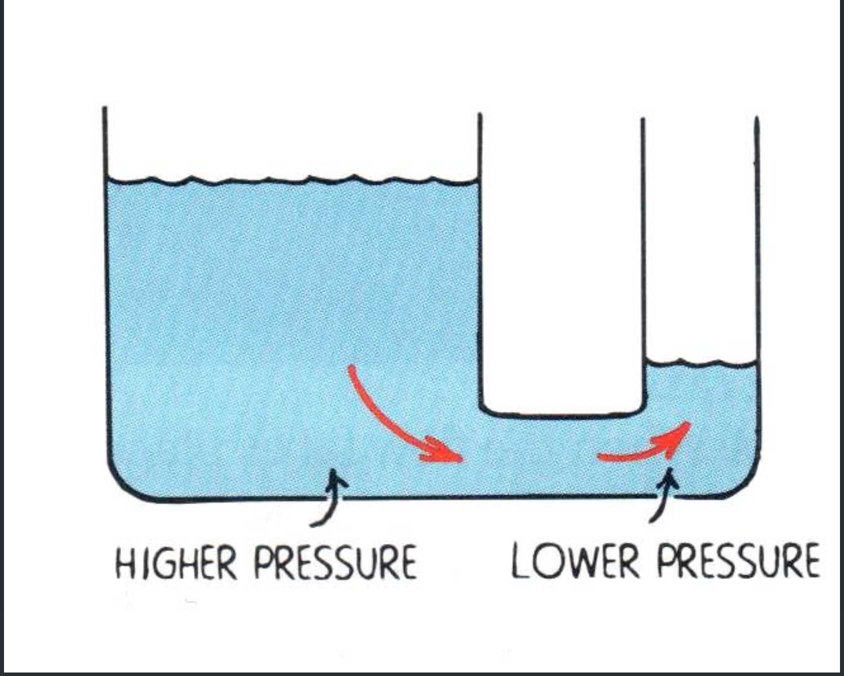
The Breadboard Challenge

- Using the materials in your bin, you are to build 3 separate closed circuits. Each circuit grows in complexity, but your task is to learn how a breadboard can be used to connect these terms together.
- After you complete each circuit, take a **photo** of it with a camera phone AND draw a **CIRCUIT DIAGRAM**
  - Circuit 1: Breadboard, LED, resistor, wires
  - Circuit 2: Breadboard, LED, resistor, wires, and a button
  - Circuit 3: Breadboard, 2 LEDs, resistor, wires, and a button

OPTIONAL SLIDES

- The next four slides are more “in-depth” as to the physics of circuits and can be omitted or added depending on your teaching context.

What Causes Current?



Water flow is similar to the flow of electric __charge__ (typically __electrons__)…

Charge will flow (a current) as long as there is a __difference__ in "electric __pressure__".

What Causes Current?

This difference in "electric pressure" is called a difference in __electric__ __potential__ or a __voltage__ difference and is measured in units of __Volts__ (__V__).

A Cause and Effect Relationship:

__Voltage (difference)__ is the cause, and __Current__ is the effect.

A Voltage difference can be thought of as the __"push"__ that causes __current__.

Check out what happens when a tree branch falls on two power lines at different voltages…….

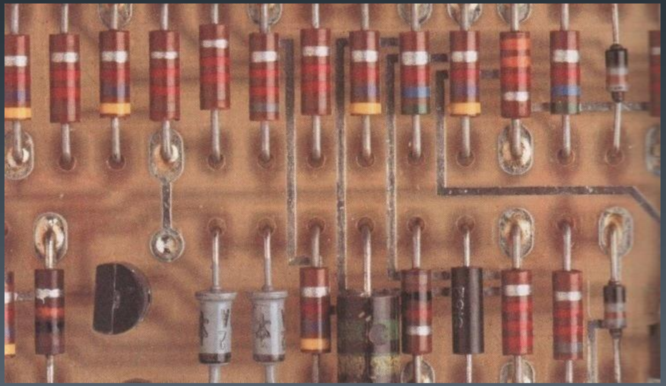
[VIDEO]

Resistance (R)

Current produced depends on the electrical **resistance** that the conductor offers to the flow of charge.

The components of a circuit (such as a light bulb filament, an electric motor, etc.), as well as the **wires** in a circuit, have electrical resistance.

A __resistor__ is a very common device found in circuits of electronic equipment. A color code is used to describe how much resistance to the flow of current that the resistor offers.



Color-coded Resistors

Getting to Know the Arduino

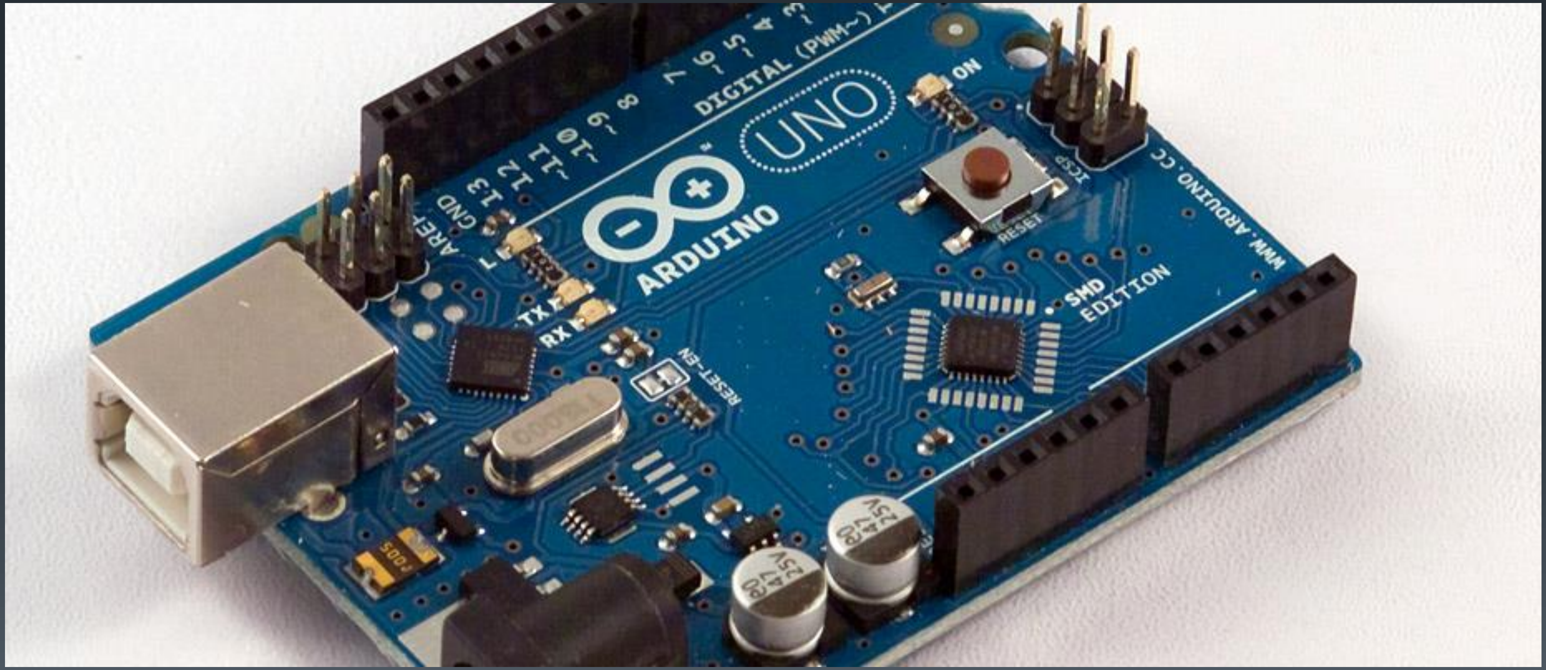


Image credit: Ardunio, http://ardunio.cc

Parts of an Arduino

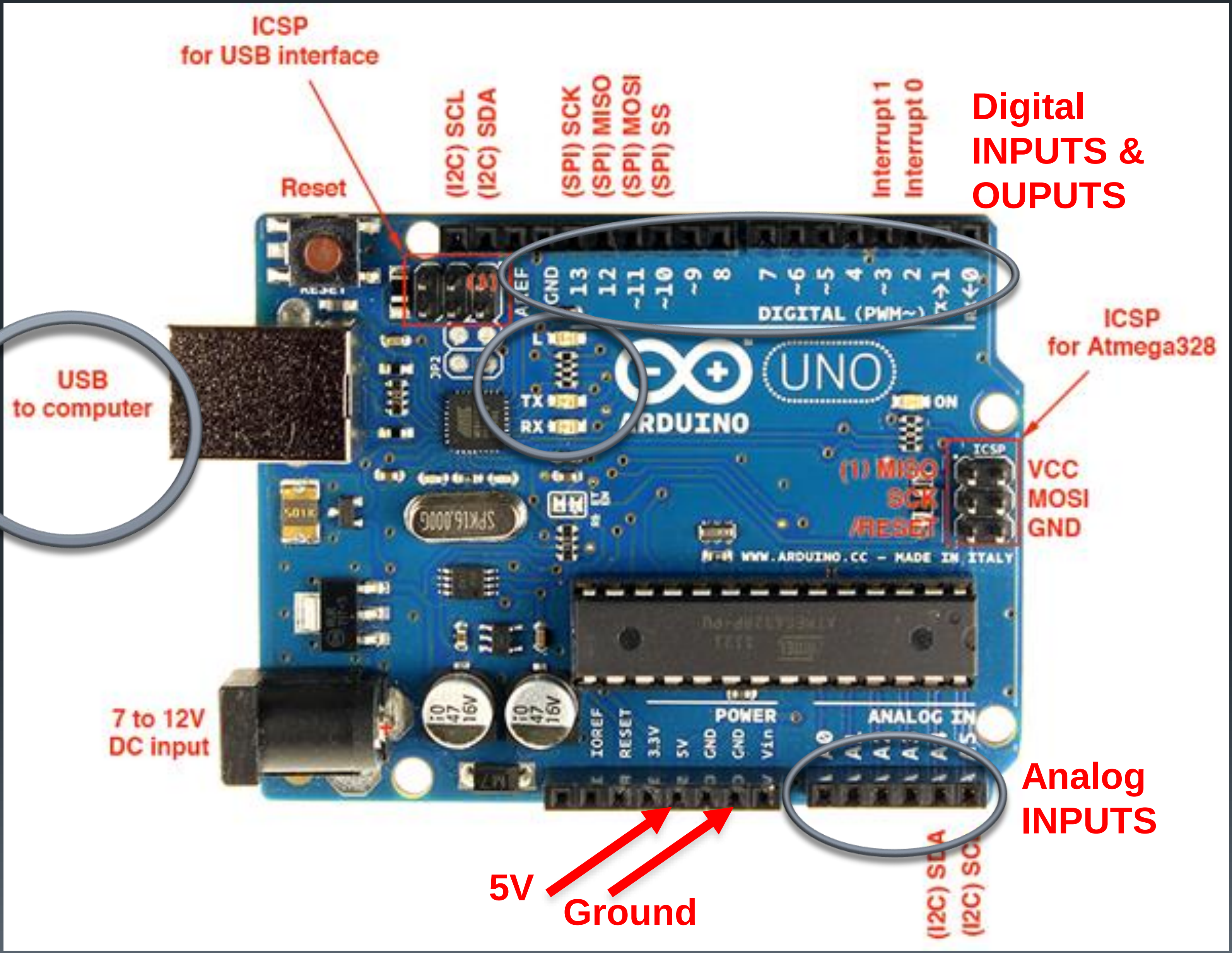
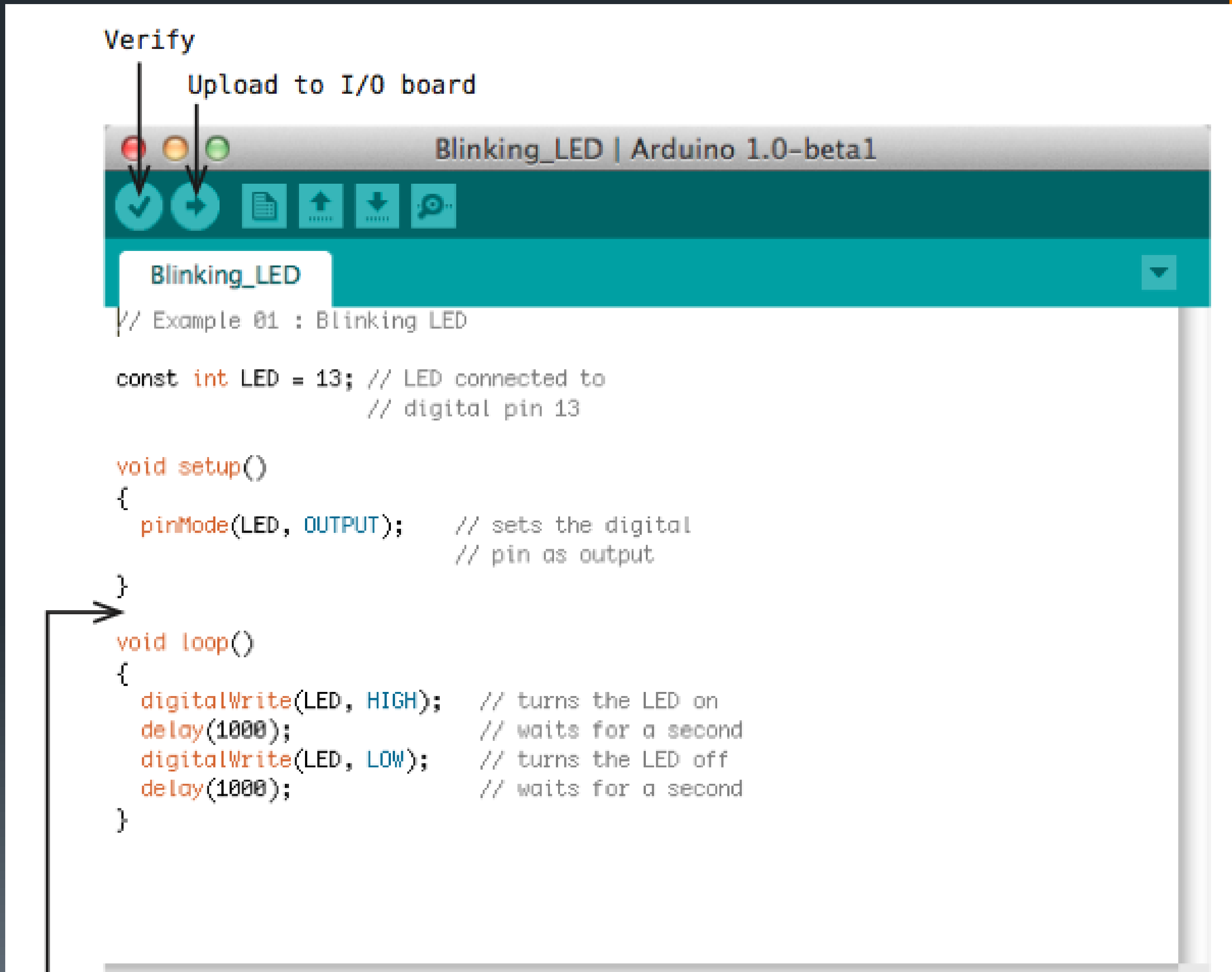
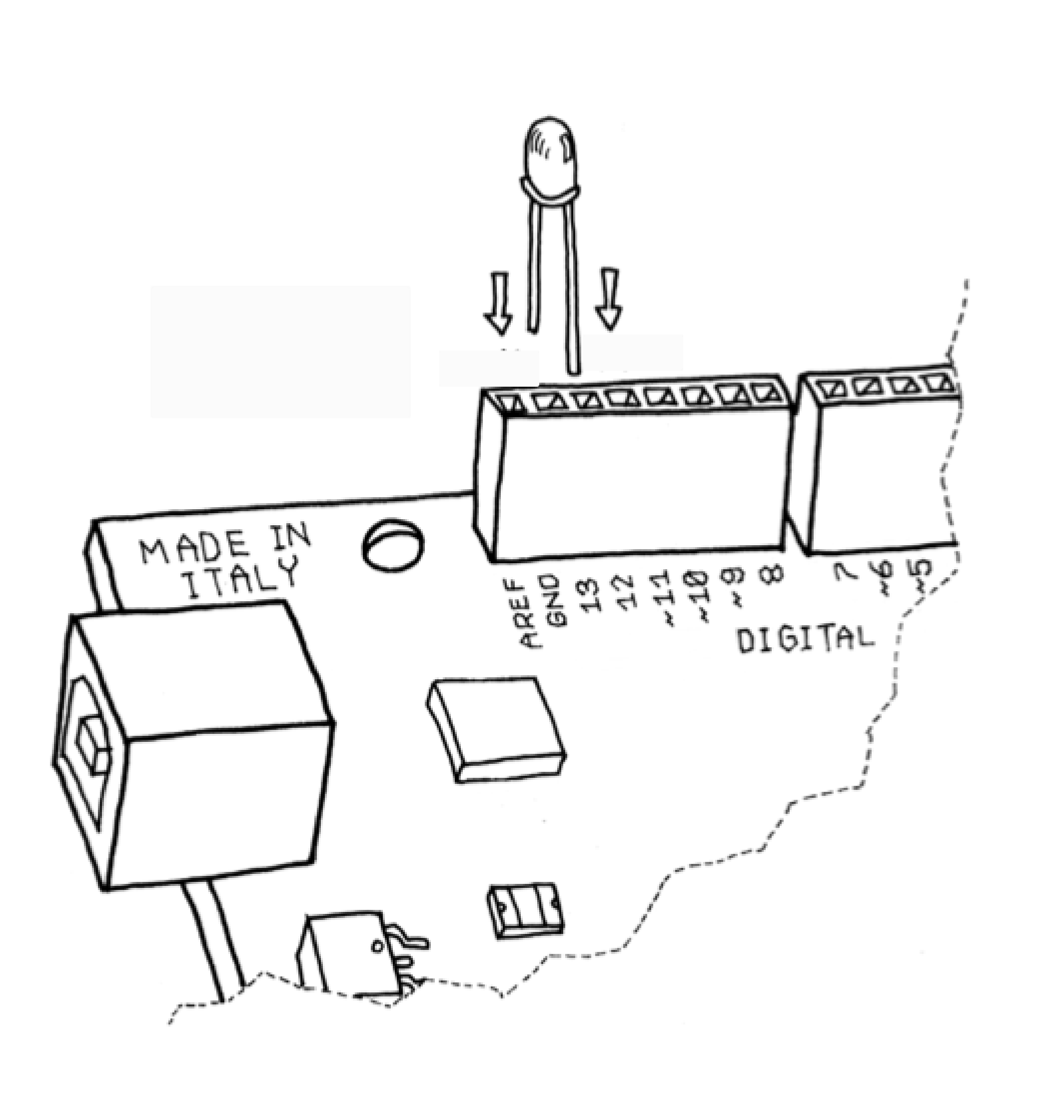


Image credit:
Adafruit / Nick
Gammon

The Parts of a Sketch



LED & Arduino Diagram



Arduino Challenge #1

- Get the BLINK sketch to work.
  - Identify the system INPUTS, OUTPUTS, any sub-systems, and if there is feedback.
  - Then, modify the sketch in two completely different ways to change the output of the system. Raise your hand once you have completed both changes. Be prepared to share out your change!

Arduino Challenge #2

- Get the BUTTON sketch to work.
- Identify the system INPUTS, OUTPUTS, any sub-systems, and if there is feedback.
- Then, modify the sketch in two completely different ways to change the output of the system. Raise your hand once you have completed both changes. Be prepared to share out your change!

Arduino Challenge #3

- Get the FADING LIGHT sketch to work!
  - Identify the system INPUTS, OUTPUTS, any sub-systems, and if there is feedback.
  - Then, modify the sketch in two completely different ways to change the output of the system. Raise your hand once you have completed both changes. Be prepared to share out your change!

Lesson Four: Building an Artificial Neural Network with Arduinos

Center for Neurotechnology

Lesson Plan Authors: Benjamin Hart, Redmond High School and Lawrence Bencivengo Jr., Mercer Island High School

LESSON OVERVIEW

Activity Time: Two 50 minute class periods.

Lesson Plan Summary: In this lesson, students will work in groups to build model neurons using Arduino controllers and breadboard circuits, and then they will assemble their neurons into a simple Artificial Neural Network (ANN). Once the ANN is functional, the class will explore how changing properties of individual neurons alters the behavior of the ANN.

STUDENT UNDERSTANDINGS

Big Idea & Enduring Understanding:

- **Artificial Neural Networks (ANNs):** ANNs are one attempt to build machines with Artificial Intelligence (AI) by mimicking the nervous systems of biological organisms. An ANN links many artificial neuron models together in such a way that the network can "learn" to categorize inputs by "mapping" various inputs to the desired outputs during training sessions. In an ANN, a model neuron receives inputs from other neurons, and then produces an output based on a weighted sum of all its inputs. The output from this neuron becomes, in turn, the input to other neurons, culminating in the final output of the network. ANNs can "learn" using algorithms which gradually modify the weights based on how closely the network's outputs match the correct patterns during training sessions.

Essential Questions:

- What are the potential applications for Artificial Neural Networks (ANN)? How might they be helpful for Brain-Computer Interfaces?
- How can an ANN "learn"?
- How does the behavior of an ANN embody "knowledge" about the input patterns that it is categorizing?
- What does it mean to "learn," and how does this relate to the concept of intelligence?

Learning Objectives:

Students will know…

- ANNs may have useful applications to Brain-Computer Interfaces in order to interpret complex data quickly and produce appropriate and safe outputs.
- The components of an Artificial Neural Network (Input Layer, Hidden Layer(s), Output Layer, weights, thresholds [or biases]).
- ANNs can learn to correctly categorize input patterns using Training Trials and learning algorithms which adjust the weights of the network until its outputs match the correct patterns.

Students will be able to…

- Assemble simple Arduino control circuits that will model an artificial neuron.
- Assemble model neurons into a functional Artificial Neural Network.
- Alter the behavior of the ANN by changing its physical structure and/or modifying the scripts that the Arduinos use to model neuron behavior.

Vocabulary:

- Artificial Neural Network (ANN)
- Hidden Layer
- Input Layer
- Inputs
- Learning Algorithm
- Output Layer
- Outputs
- Perceptron
- Sigmoid Function
- Threshold (or Bias)
- Training Data
- Training Trials
- Weights

Standards Alignment: This lesson addresses the following high school Next Generation Science Standards (NGSS).

NGSS High School Disciplinary Core Ideas

- **HS-ETS1-1 Engineering Design:** Analyze a major global challenge to specify qualitative and quantitative criteria and constraints for solutions that account for societal needs and wants.
- **HS-ETS1-2 Engineering Design:** Design a solution to a complex real-world problem by breaking it down into smaller, more manageable problems that can be solved through engineering.
- **HS-ETS1-3 Engineering Design:** Evaluate a solution to a complex real-world problem based on prioritized criteria and trade-offs that account for a range of constraints,

including cost, safety, reliability, and aesthetics as well as possible social, cultural, and environmental impacts.

NGSS Crosscutting Concepts

- **Patterns:** Observed patterns of forms and events guide organization and classification, and they prompt questions about relationships and the factors that influence them.
- **Cause and Effect—Mechanism and Explanation:** Events have causes, sometimes simple, sometimes multifaceted. A major activity of science is investigating and explaining causal relationships and the mechanisms by which they are mediated. Such mechanisms can then be tested across given contexts and used to predict and explain events in new contexts.
- **Structure and Function:** The way in which an object or living thing is shaped and its substructure determine many of its properties and functions.
- **Stability and Change:** For natural and built systems alike, conditions of stability and determinants of rates of change or evolution of a system are critical elements of study.

NGSS Science & Engineering Practices:

- Developing and using models.

MATERIALS

Material	Description	Quantity
Arduino Starter Kit	Includes Arduino UNO microcontroller, breadboard and basic electronics components. Available from Arduino.com or Amazon.com	8 - 10 per class
Student Handout 4.1	*Assembling an Artificial Neural Network*	1 per student
Student Handout 4.2a, b, c	*4.2a Assembling the Input Layer Neuron, 4.2b Assembling the Hidden Layer Neuron, 4.2c Assembling the Output Layer Neuron*	3 copies of each handout per class
Student Handout 4.3	*"Programming" an Artificial Neural Network*	1 per student
Teacher Resource 4.1	*Assembling an Artificial Neural Network--Teacher Answer Key*	1

Teacher Resources 4.2 a, b, c	*Input Layer Neuron Sketch,* *Hidden Layer Neuron Sketch,* *Output Layer Neuron Sketch*	Provide to students electronically so they can copy and paste
Teacher Resource 4.3	*"Programming" an Artificial Neural Network—Teacher Answer Key*	1

TEACHER PREPARATION

1. Make copies of ***Student Handouts 4.1*** and **4.3**, one per student.

2. Make copies of ***Student Handouts 4.2 a, b,*** and ***c***. You will need three copies of each handout per class (plus a few extras). It is helpful if these can either be printed in color or made available to students in electronic form, as the schematics for assembling the artificial neurons are color-coded. It does not actually matter what color wires are used to assemble the circuits; the color-coding is used simply to make the diagrams easier to read.

3. Divide the class into nine groups.

4. ***Optional***: To familiarize yourself with the structure of the neural network, construct an input layer, hidden layer, and output layer neuron.

PROCEDURE

Engage: How could an Arduino simulate a Neuron?

1. Distribute ***Student Handout 4.1: Assembling an Artificial Neural Network***.

2. Have the class list the parts and functions of biological neurons and simulated neurons.

3. In groups, have students discuss what minimum functions and components would be required for an Arduino to behave like a neuron in an Artificial Neural Network.

4. Discuss group suggestions and pare down to the minimum functionality required:
    a. one or more inputs

b. one or more outputs
c. a sketch to decide what outputs to generate depending upon the inputs
d. it is also helpful to have LEDs or some other external indicator of the neuron's internal state

Explore:

4. Divide the class into 9 groups. Each group will be assigned to build either:

- Input Neuron (3 groups - use ***Student Handout 4.2a***)
- Hidden Level Neuron (3 groups - use ***Student Handout 4.2b***)
- Output Neuron (3 groups - use ***Student Handout 4.2c***)

Note: If some students have experience with electronics, it is recommended that they be placed into one of the Hidden Level or Input Level groups, as these assemblies are slightly more challenging.

5. Provide each group with the materials they will need to assemble their artificial neuron models, along with copies of ***Student Handouts 4.1*** and ***4.2 a, b*** or ***c*** (according to which neuron they are building).

6. Working in their groups, students should attempt to assemble the components provided as shown in Part II of ***Student Handout 4.1*** and in ***Student Handout 4.2 a, b*** or ***c*** (according to which neuron they are building).

7. After building the circuit, students should upload the correct sketch to their Arduino. Use ***Teacher Resources 4.2 a, b, c***. These sketches are provided as a teacher resource rather than as student handouts so that you can make them available to students electronically; this will make it easier to copy and paste each sketch.

8. Each group will test its neuron to make sure it is functioning correctly.

Explain:

9. If they haven't already completed it, challenge students to fill in the diagram of a biological neuron on page 1 of the handout and respond to Questions # 1-3 in Part I. As you circulate among the groups, check students' explanations of how each artificial neuron receives inputs and produces outputs, how the neurons can communicate with each other, and how they will be assembled into an ANN.

Elaborate:

10. Explain the class challenge: the 9 student groups are challenged to assemble their artificial neurons into a ***single*** ANN with 9 neurons according to the diagram in Part III of Student Handout 4.1.

    Note: It is essential that the neurons be grounded together, otherwise the behavior of the network will be unpredictable! It is recommended that jumper wires be used to connect the - (ground) columns of each breadboard circuit to at least two (2) other neurons to insure the network is properly grounded.

8. Next, guide students through the process of testing the ANN to make sure it is assembled correctly and functioning, finding and fixing any nodes or connections which have been incorrectly assembled. Students can use Part IV of ***Student Handout 4.1*** to guide them through this process.

9. Allow time for students to freely explore how the behavior of the network changes as various parameters are altered, such as by changing the weights each neuron uses to determine its output.

10. Once the network is connected and grounded, it is time to figure out how to "program" it so that it will do something interesting—in this case, count inputs! Distribute ***Student Handout 4.3: "Programming" an Artificial Neural Network***. These activities will help students determine a set of weights and thresholds that will allow the ANN to count how many inputs (from 0 to 3) are currently on.

11. Once students have completed and understood Student Handout 4.3, they can agree as a class on what weights and thresholds each neuron of the ANN should set to. Students will need to alter the Hidden Layer and Output Layer sketches by changing the synapse Weights and Thresholds, then uploading the new sketches separately to each Hidden Layer and Output Layer neuron (the sketches of the Input Layer neurons will not require any changes). Once this is done, the ANN should indicate how many Input Layer neurons are on, which is easily demonstrated by covering the photoresistors to switch various Input Layer Neurons off.

Evaluate:

12. Through repeated testing and modifications, the class will evaluate how well the ANN accomplishes the designated task of categorizing different Input Patterns by mapping them onto specific Output Patterns.

13. Ask students to complete *Part V: Analysis and Conclusions* of ***Student Handout 4.1***.

14. Review students' answers to Part V of the handout. Discuss how the ability of the ANN to correctly categorize Input Patterns could be improved by systematically altering the weights and/or other parameters of the network. Also discuss the concept of a Learning Algorithm.

EXTENSION ACTIVITIES

Adaptations:

- If some students have experience with electronics, it is recommended that they be placed into one of the Hidden Level or Input Level groups, as these assemblies are slightly more challenging.
- Visuals and diagrams can help struggling students better understand the connectivity of the Arduino.

TEACHER BACKGROUND & RESOURCES

Resources:

- Neural Networks and Deep Learning
  http://neuralnetworksanddeeplearning.com/

- A Neural Network for Arduino
  http://robotics.hobbizine.com/arduinoann.html

Citations:

Fritzing.org.

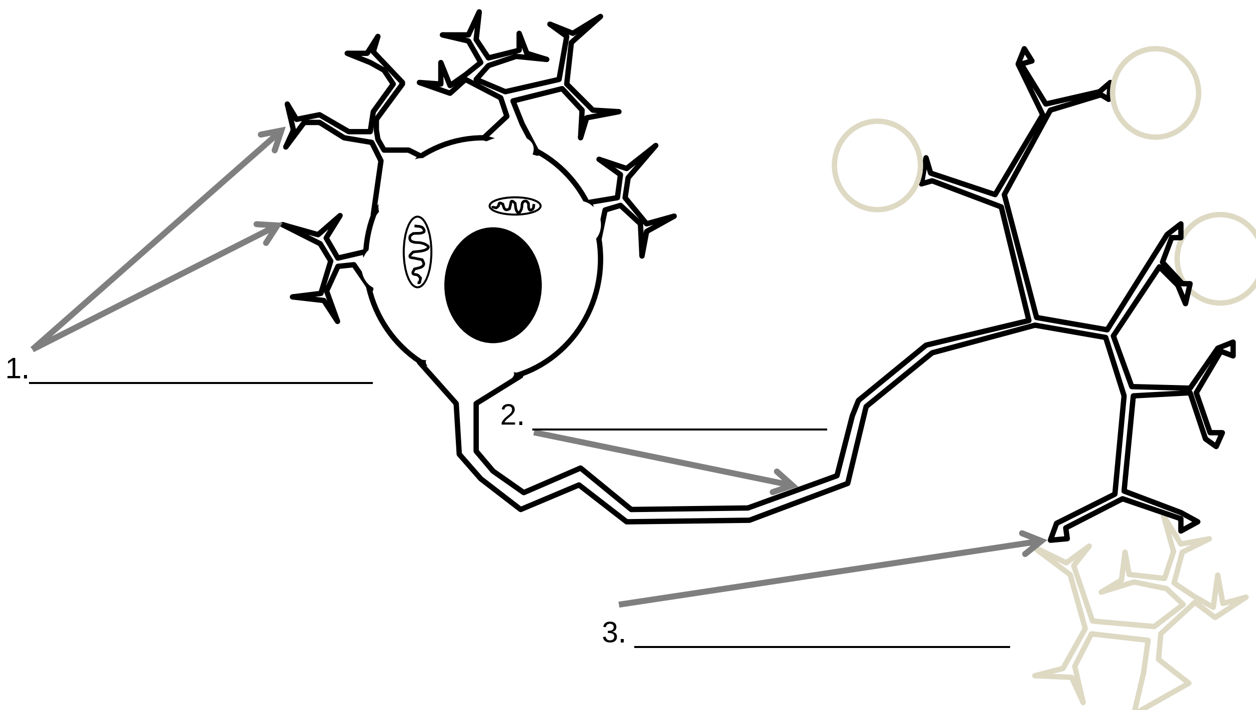
Nielsen, Michael. 2015. Neural Networks and Deep Learning. Determination Press. http://neuralnetworksanddeeplearning.com/

A Neural Network for Arduino. [n.a.] Hobbizine. http://robotics.hobbizine.com/arduinoann.html

Student Handout 4.1: Assembling an Artificial Neural Network

Name:_________________________________ Date:_______________ Period:__________

Part I: Understanding Artificial Neural Networks



1. Label the parts of a biological neuron indicated by the arrows in the diagram above. In the space below, *briefly* describe the function of each of these parts.

2. *Briefly* explain how an Arduino can be used to create an artificial neuron.

3. Briefly describe the purpose of the Input, Hidden, and Output layers of an Artificial Neural Network.

Part II: Assembling the Artificial Neurons

The pictures and diagrams below show you how to assemble artificial neurons using Arduinos and simple breadboard circuit components (see also ***Handouts 4.2 a, b,*** or ***c***). Your teacher will assign your group one of these artificial neuron types to assemble. Work together with your group members to assemble your artificial neuron, upload the correct sketch to the Arduino, and test your neuron to make sure it is assembled and functioning correctly. Then wait for further instructions about how to add your group's neuron to the Artificial Neural Network.

Input Layer Assembly

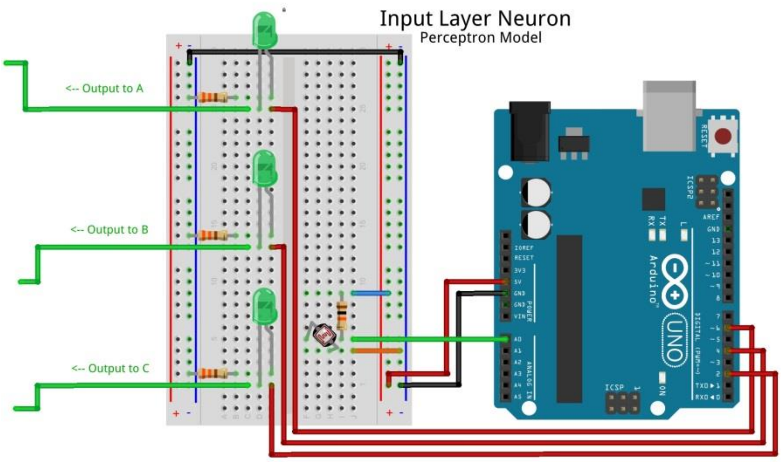
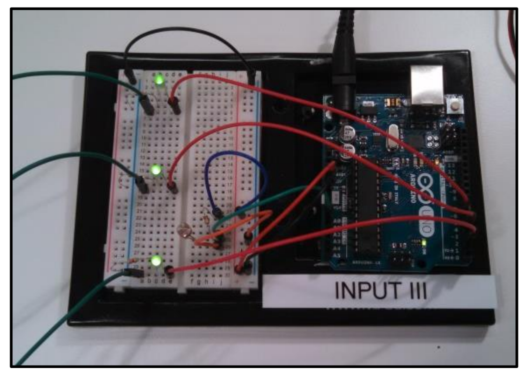


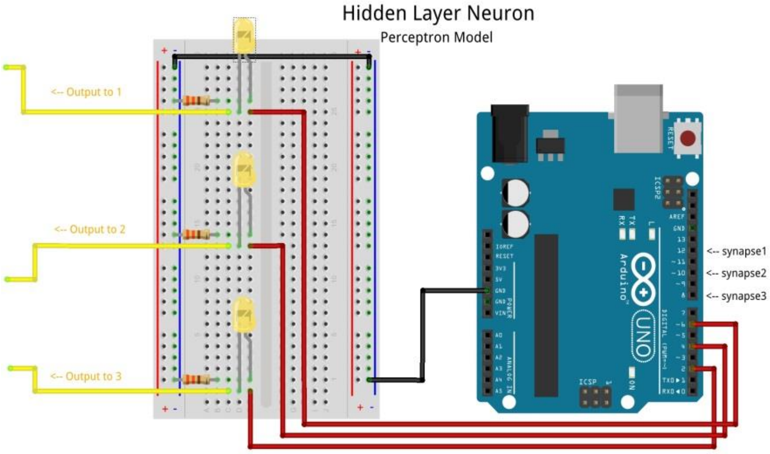
Image created with fritzing: fritzing.org



A fully assembled Input Layer Neuron

Three (3) groups will assemble **<u>Input Layer Neurons</u>**. These neurons have the most complex circuits because they receive their input from a light detector (see the lower right corner of the breadboard). The green LEDs are indicators to show that the Input Layer neuron is sending signals to Hidden Layer neurons. While the lights are on, the neuron is firing. When the lights are off, the neuron is "resting." See ***Handout 4.2a*** for additional information.

Hidden Layer Assembly



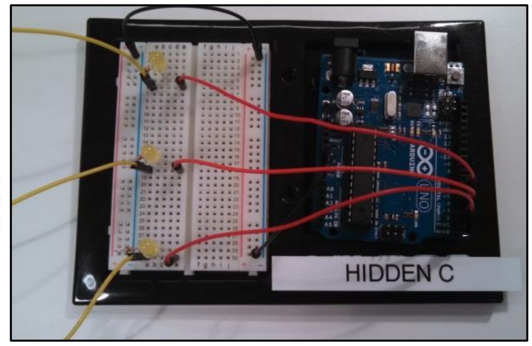
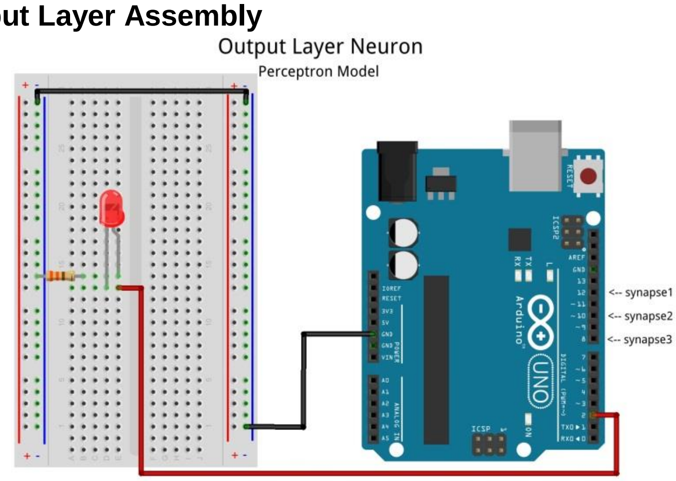


Image created with fritzing: fritzing.org

A fully assembled Hidden Layer Neuron

Three (3) groups will assemble **<u>Hidden Layer Neurons</u>**. These neurons receive their inputs from the Input Layer neurons, with pins # 8, 10 & 12 serving as the ‘synapses’ for these inputs (see the diagram above). The yellow LEDs are indicators to show that the Hidden Layer neuron is sending signals to Output Layer neurons. While the lights are on, the neuron is firing. When the lights are off, the neuron is “resting.” See ***Handout 4.2b*** for additional information.

Output Layer Assembly



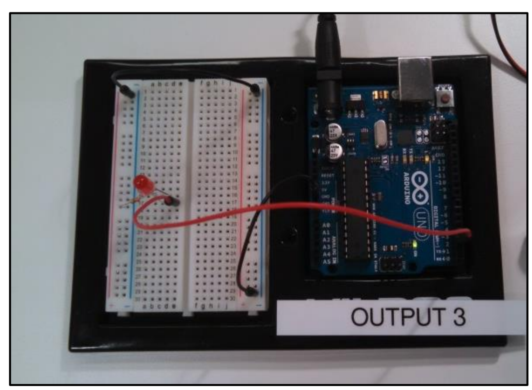


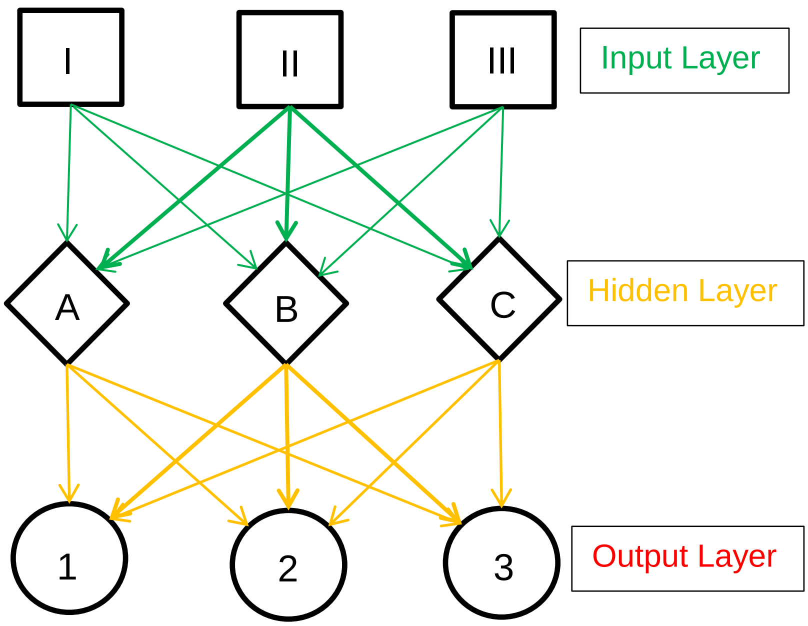
Image created with fritzing: fritzing.org

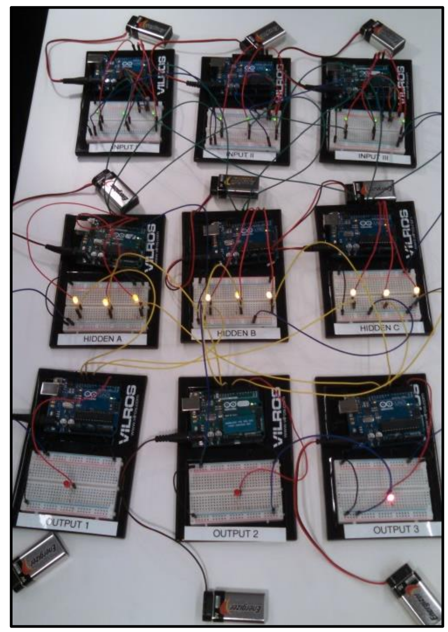
A fully assembled Output Layer Neuron

Three (3) groups will assemble **<u>Output Layer Neurons</u>**. These neurons receive their inputs from the Hidden Layer neurons, with pins # 8, 10 & 12 serving as the ‘synapses’ for these inputs (see the diagram above). The Output Layer neurons have the simplest circuits to assemble because they do not send outputs to other neurons. There is only a single red LED which serves as an indicator to show that the neuron is firing. See ***Handout 4.2c*** for additional information.

Part III: Assembling the Artificial Neural Network

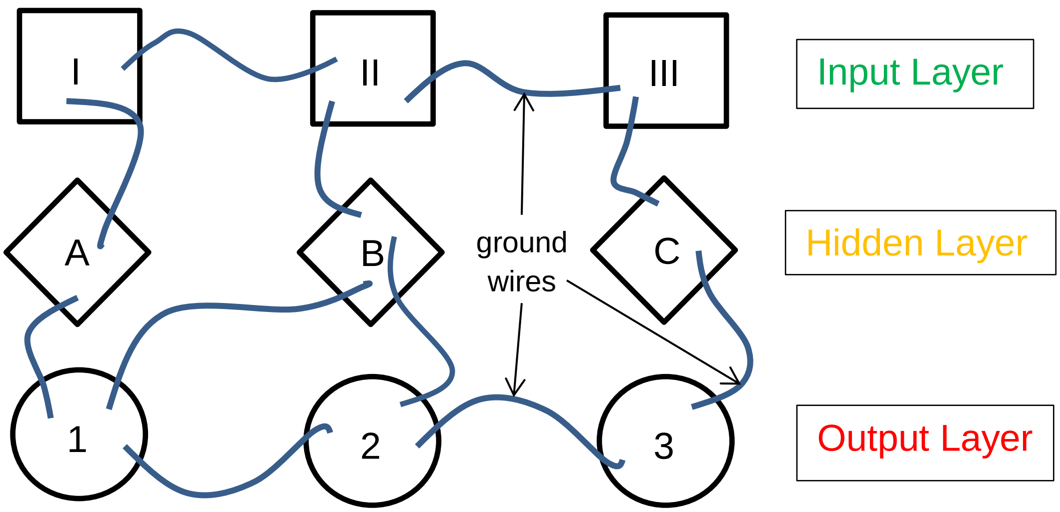
Once all 9 groups have assembled their neurons, uploaded the correct sketch and tested them, it is time to assemble the Artificial Neural Network. The schematic for assembling the network is shown on the left below, with an assembled ANN pictured on the right:





Each neuron **receives inputs** from other neurons through pins # 8, 10 and 12. Neurons **send outputs** through pins # 2, 4 and 6 to the LEDs on their breadboards. Jumper wires transmit these outputs from the breadboard to other neurons as indicated in the schematic above.

In addition to wiring the neurons together, the network must also be “**grounded**” together. This can be easily accomplished by using jumper wires to connect the ground lines (blue ‘-‘columns on the bread boards) together, similar to the schematic below. Each neuron should be grounded to at least 2 other neurons to insure that the network will function properly.



Part IV: Understanding How an Artificial Neural Network Functions

Once the ANN is assembled with all the neurons “synapsed” and grounded together, all nine neurons should be “Firing” (i.e., their LEDs should be lit). Follow your teacher’s instructions for setting all of the artificial neuron’s weights and thresholds to 0.5 and exploring the behavior of the ANN. Record your observations below and answer the questions that follow:

<u>Observations:</u>

1. What happens when *exactly one* of the Input neurons is switched off by covering the photoresistor? That is, how does the behavior of the network as a whole change? Explain why this happens.

2. What happens when two or more Input neurons are switched off? Explain.

3. Does it matter which *specific* Input neurons are on or off (or just how many)? Explain.

4. At this point, our ANN can only “categorize” a few different Input Patterns. Describe the “mapping” of Input Patterns to Output Patterns of the current network configuration. By “mapping”, we mean what is the pattern of Outputs from the network for each pattern of Inputs? For example, when all of the Input neurons are off, which (if any) Output neurons are switched on? What if only Input I is on, etc.?

Inputs	Outputs
000	
100	
010	
001	
110	
101	
011	
111	

5. What happens if we gradually “degrade” the connectivity of the network (e.g., if we disconnect just *one* of the inputs to a single neuron)? How does each small change alter the behavior of the network? Does the behavior of the network change dramatically or only slightly with each small change? Explain.

<u>Complete the Network:</u> Before continuing your exploration, follow your instructor’s directions for “programming” the ANN to categorize a specific mapping of inputs to outputs. You and your classmates will determine the weights and thresholds needed to accomplish the task, then alter each neuron’s sketch accordingly and study the behavior of the network. See ***Student Handout 4.3: “Programming” An Artificial Neural Network***. When you have completed the exploration, go on to finish the Analysis and Conclusion questions below.

Part V: Analysis & Conclusions:

1. What kind of signal is sent down the axon of a biological neuron? How is this similar to and different from the signals generated by the artificial neurons in our ANN?

2. What kind of signal is sent across the synapses of biological neurons? How is this similar to and different from the way signals are received by the artificial neurons in our ANN?

3. In a biological nervous system, some synapses are excitatory and some are inhibitory. How does an ANN create excitatory and inhibitory synapses?

4. How do the weights assigned to each neuron's inputs affect the mapping you described in # 4 of the Observations above (i.e., how does the behavior of the individual neurons collectively 'add up' to the overall behavior of the ANN)?

5. Based on the results of Observation # 5 above, how is “information” or “knowledge” about the Input Patterns stored in the neural network? For example, do individual neurons store information about specific Input Patterns, or is knowledge “distributed” across the entire network? Is it a little bit of both? How can you tell?

6. How could an ANN “learn” to generate the correct outputs without us “programing” in the weights (i.e. is there some way the network could teach *itself* how to adjust its own weights)?

Student Handout 4.2a: Assembling the Input Layer Neuron

Name:________________________________ Date:______________ Period:_____________

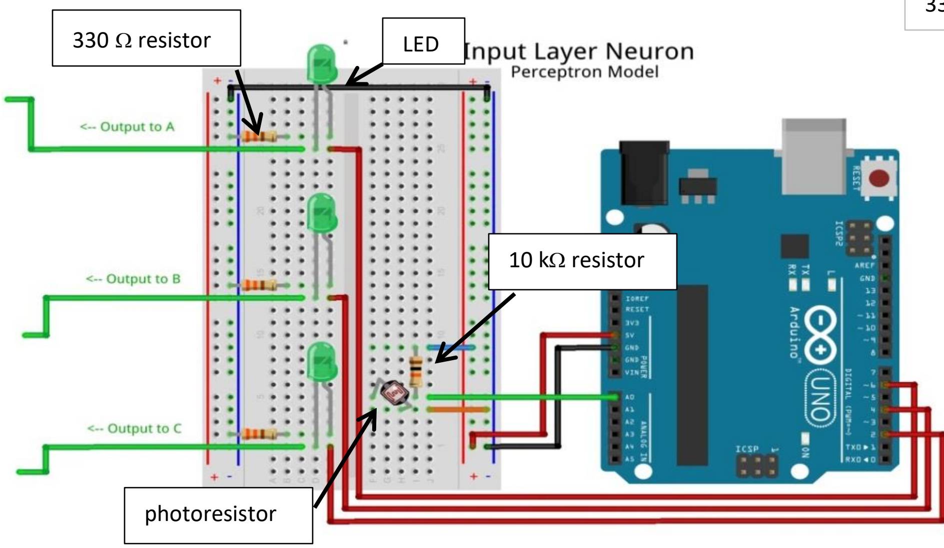
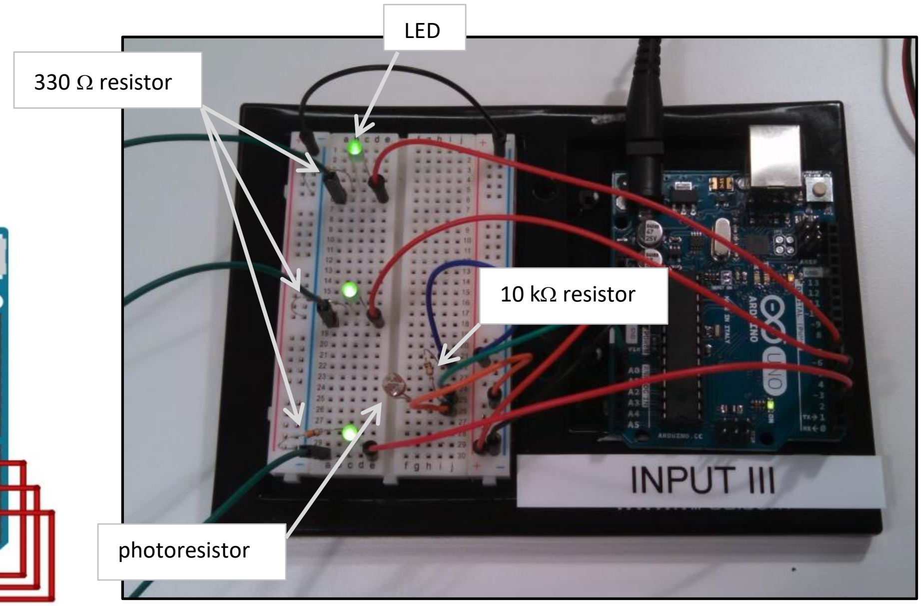


Image created with fritzing: fritzing.org



A fully assembled Input Layer Neuron

Input Layer neurons use a photoresistor to detect light levels. This provides the Input to the Artificial Neural Network. See the Comments in the Input Neuron sketch to adjust the value of 'Threshold' so that the LEDs light up and turn off properly.

Assemble the circuit as shown above. You will need the following components:

- 3x Green LEDs
- 3x 330Ω resistors
- 3x 12" green jumper wires
- 1x breadboard
- 1x 9V battery and connector
- 1x photoresistor
- 1x 10kΩ resistor
- 1x short orange jumper wire
- 1x short blue jumper wire
- 1x USB connector (not shown)
- 3x medium length red jumpers
- 1x short black jumper wire
- 1x short red jumper wire
- 1x Arduino UNO controller

Student Handout 4.2a: Assembling the Input Layer Neuron

Assembling the Input Layer Neuron

1. Load the Input Layer Neuron sketch onto the Arduino using the USB connector. All 3 LEDs should light up.

2. When you cover the photoresistor with your finger, the LED's should turn off.

3. You may need to adjust the value of the 'Threshold' variable in the sketch as described in the comments.

4. In general, if the LEDs do not light up, *increase* the value. If they do not turn off, decrease the value.

Student Handout 4.2b: Assembling the Hidden Layer Neuron

Name:________________________________ Date:___________ Period:_____________

CENTER for NEUROTECHNOLOGY

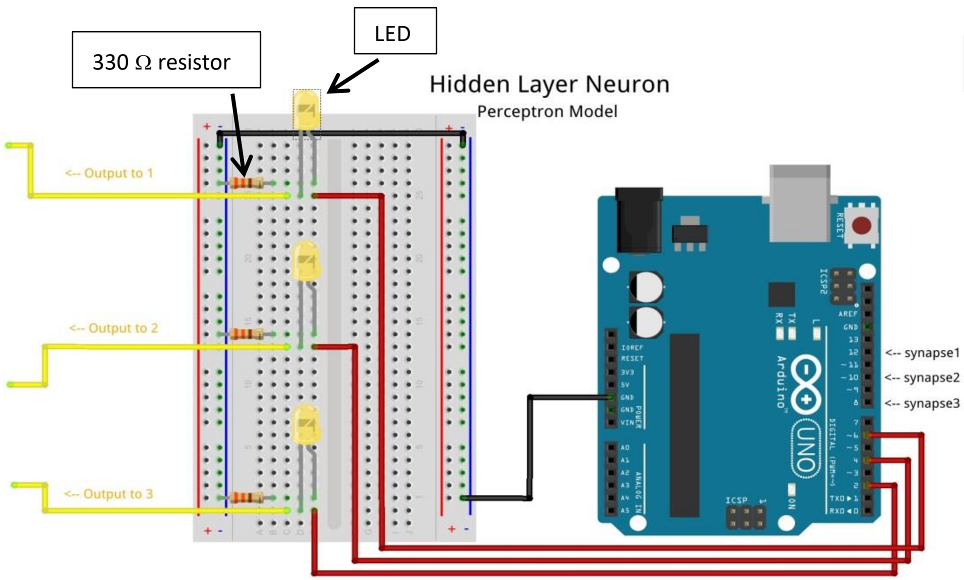
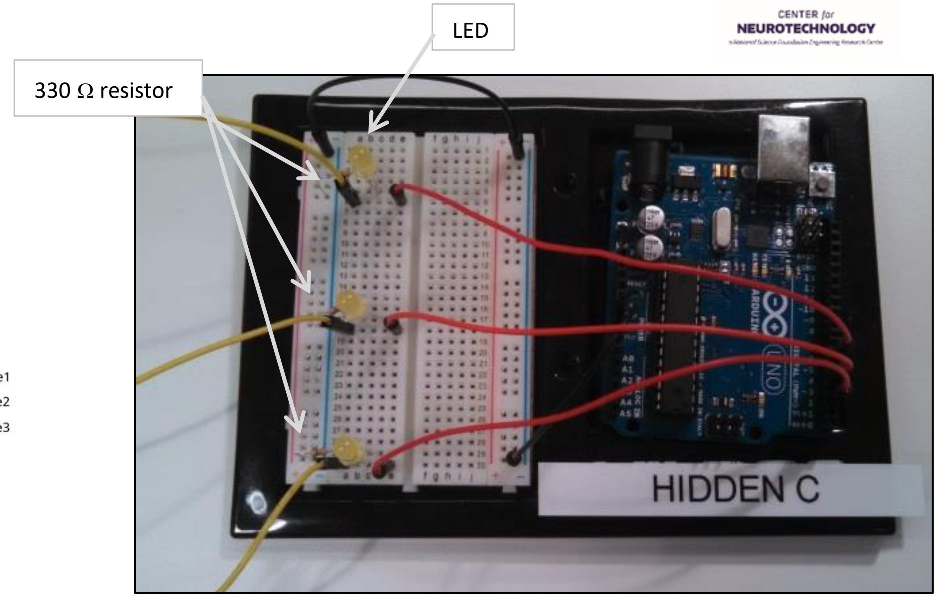


Image created with fritzing: fritzing.org



A fully assembled Hidden Layer Neuron

Hidden Layer neurons receive inputs from the Input Layer neurons (one input from each Input Layer neuron). Use pins # 8, 10 & 12 to connect the long green jumper wires from the Input neurons to each Hidden Layer neuron.

Assemble the circuit as shown above. You will need the following components:

3x Yellow LEDs	3x medium length red jumper wires	1x breadboard
3x 330Ω resistors	1x short black jumper wire	1x Arduino UNO controller
3x 12” green jumper wires	1x 9V battery and connector	1x USB connector (not shown)

Student Handout 4.2b: Assembling the Hidden Layer Neuron

Assemble the Hidden Layer Neuron

1. Load the Hidden Layer Neuron sketch onto the Arduino using the USB connector. Make sure that the 'Testing' variable is set to 'true' ("boolean Testing = true;" on Line #62). All 3 LEDs should blink on and off.

2. Once you have tested the neuron to make sure it is functioning, you will need to change the value of 'Testing' to 'false' on Line #62.

3. Follow your teacher's instructions to make other changes to the sketch so that your Hidden Layer neuron can function as part of an Artificial Neural Network (ANN). Then recompile and reload the sketch to your neuron before adding it to the ANN.

Student Handout 4.2c: Assembling the Output Layer Neuron

Name:_________________________________ Date:_____________ Period:______________

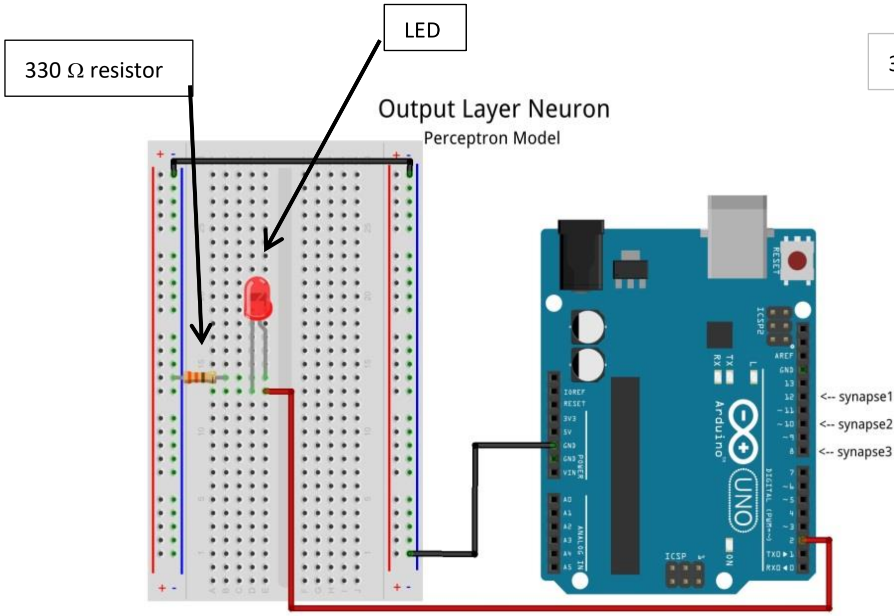
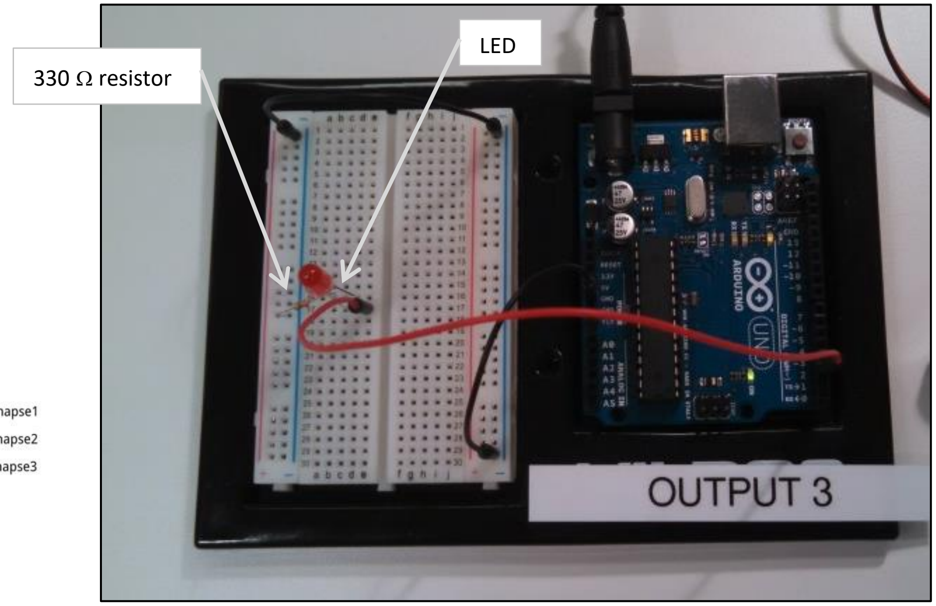


Image created with fritzing: fritzing.org



A fully assembled Output Layer Neuron

Output Layer neurons receive inputs from the Hidden Layer neurons (one input from each Hidden Layer neuron). Use pins # 8, 10 & 12 to connect the long yellow jumper wires from the Hidden neurons to each Output Layer neuron.

Assemble the circuit as shown above. You will need the following components:

1x Red LED	1x medium length red jumper wire	1x breadboard
1x 330Ω resistor	1x short black jumper wire	1x Arduino UNO controller
1x 9V battery and connector	1x USB connector (not shown)	

Student Handout 4.2c: Assembling the Output Layer Neuron

Assembling the Output Layer Neuron

1. Load the Output Layer Neuron sketch onto the Arduino using the USB connector. Make sure that the ‘Testing’ variable is set to ‘true’ (“boolean Testing = true;” on Line #60). The LED should blink on and off.

2. Once you have tested the neuron to make sure it is functioning, you will need to change the value of ‘Testing’ to ‘false’ on Line #60.

3. Follow your teacher’s instructions to make other changes to the sketch so that your Output Layer neuron can function as part of an Artificial Neural Network (ANN). Then recompile and reload the sketch to your neuron before adding it to the ANN.

Name:________________________ Date:___________________ Period:__________

Student Handout 4.3: "Programming" an Artificial Neural Network

Artificial Neural Networks (ANNs) have many current and potential applications. They are especially useful for identifying or "categorizing" complex patterns (e.g., face recognition software, search engines, or "data mining" on the internet). In these activities, you will learn how ANNs work by figuring out how to "program" our network to accomplish a specific task.

The Goal: We want our ANN to identify whether 0, 1, 2 or 3 of the input layer neurons is on, indicating this by lighting up a different output layer neuron in each case (none, #1 only, #2 only, or #3 only, respectively).

Table 1
Input Pattern | Output Pattern

1	000	000
2	100	100
3	010	100
4	001	100
5	110	010
6	101	010
7	011	010
8	111	001

Inputs and Outputs: We can formally represent this mapping as follows (in the Input and Output Patterns shown below, an entry such as '100' indicates that the *first* neuron of that layer has an output of 1 while the other two neurons have output 0; '010' would indicate that the *second* neuron has output 1 while the first and third neurons have output 0, etc.).

Weights and Thresholds: Remember that each neuron's output is either 1 or 0, and that each neuron multiplies each of its inputs by a different weight, adds the weighted inputs together, then compares the sum to its threshold. If the total is *above* the threshold, it sets its output to 1, otherwise its output is 0. This is summarized by the following equations:

a. If $\sum_{i=1}^{3} weight_i \times input_i > threshold$, set output = 1

b. If $\sum_{i=1}^{3} weight_i \times input_i <= threshold$, set output = 0

For example, if we were using Input Pattern # 5 (110), Hidden Neuron A would receive inputs of 1 from Input Neurons I & II, and 0 from Input Neuron III. If Hidden Neuron A's weights were w_1 = 0.25, w_2 = - 0.33 and w_3 = 0.47, the total input to Hidden Neuron A would be (0.25 x 1.0) + (- 0.33 x 1.0) + (0.47 x 0) = 0.25 – 0.33 + 0 = -0.08. If Hidden Neuron A had a threshold of 0.5, its output would be set to 0 because the total input is less than the threshold. Therefore Hidden Neuron A will not send messages to any of the output layer neurons.

Procedure: Our ultimate goal is to work out a set of weights for each neuron that will result in the mapping of Input Patterns to Output Patterns described above. However, this would be a very challenging task without a bit of practice. The following set of exercises will help you to develop several important insights into how ANNs work that should it make it possible for you to "program" our ANN with a set of synaptic weights that will allow the ANN to correctly categorize inputs as described above.

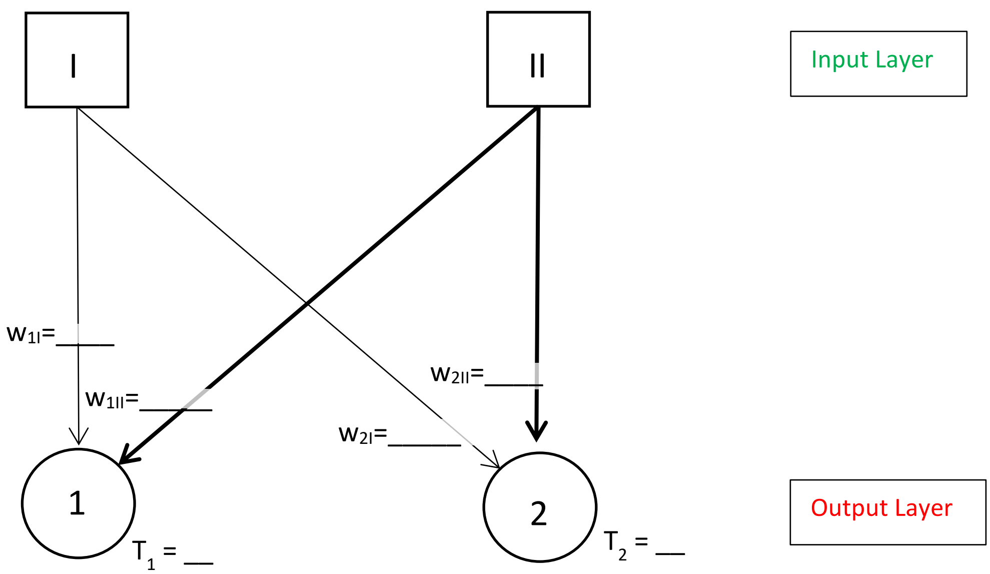
Name:______________________ Date:_________________ Period:_________

Warmup 1: Programming a 2 x 2 ANN

Directions: This activity is a warmup exercise for programming the full network (see the last page). Work in groups of two or three to find a set of weights (W_{xi}) and thresholds (T_x) that will enable the set of inputs to produce the corresponding outputs shown in Table 2 below. Notice that this simple ANN has no Hidden Layer. Once your group finds a solution, move on to Warmup 2.

Table 2

	Input Pattern	Output Pattern
1	00	00
2	10	10
3	01	01
4	11	11



For some students, programming this Input → Output mapping into the 2 x 2 network will seem simple; for others it may be very challenging. Students who complete this task easily should give *hints* to other students without simply revealing the answer.

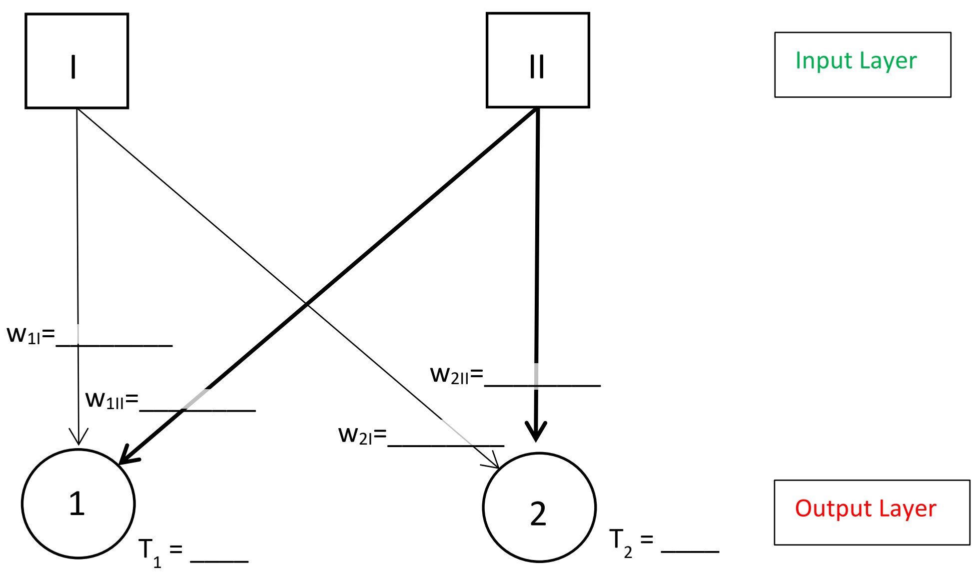
Name:______________________ Date:_________________ Period:_________

<u>**Warmup 2: Programming a 2 x 2 ANN**</u>

Directions: This second warmup exercise is much more challenging. Work in groups of two or three to find a set of weights (W_{xi}) and thresholds (T_x) that will enable the set of inputs to produce the corresponding outputs shown in Table 3 below. Is it possible?

Table 3

	Input Pattern	**Output Pattern**
1	00	00
2	10	10
3	01	10
4	11	01



This type of simple ANN with only two layers of neurons is limited in its ability to learn certain patterns. The type of pattern required to create the correct Input → Output mapping is called the "Exclusive Or" (XOR) function in Boolean logic (The Boolean XOR function receives two inputs. Its output is 1 if either of the individual inputs is 1, but the output is 0 if both inputs are 0 ***or*** if both inputs are 1). It has been shown mathematically that 2-layer ANNs are incapable of learning the XOR function! Our 3 x 3 ANN may not seem very complex, but it *is* capable of learning the XOR function, which is very useful for categorizing input patterns.

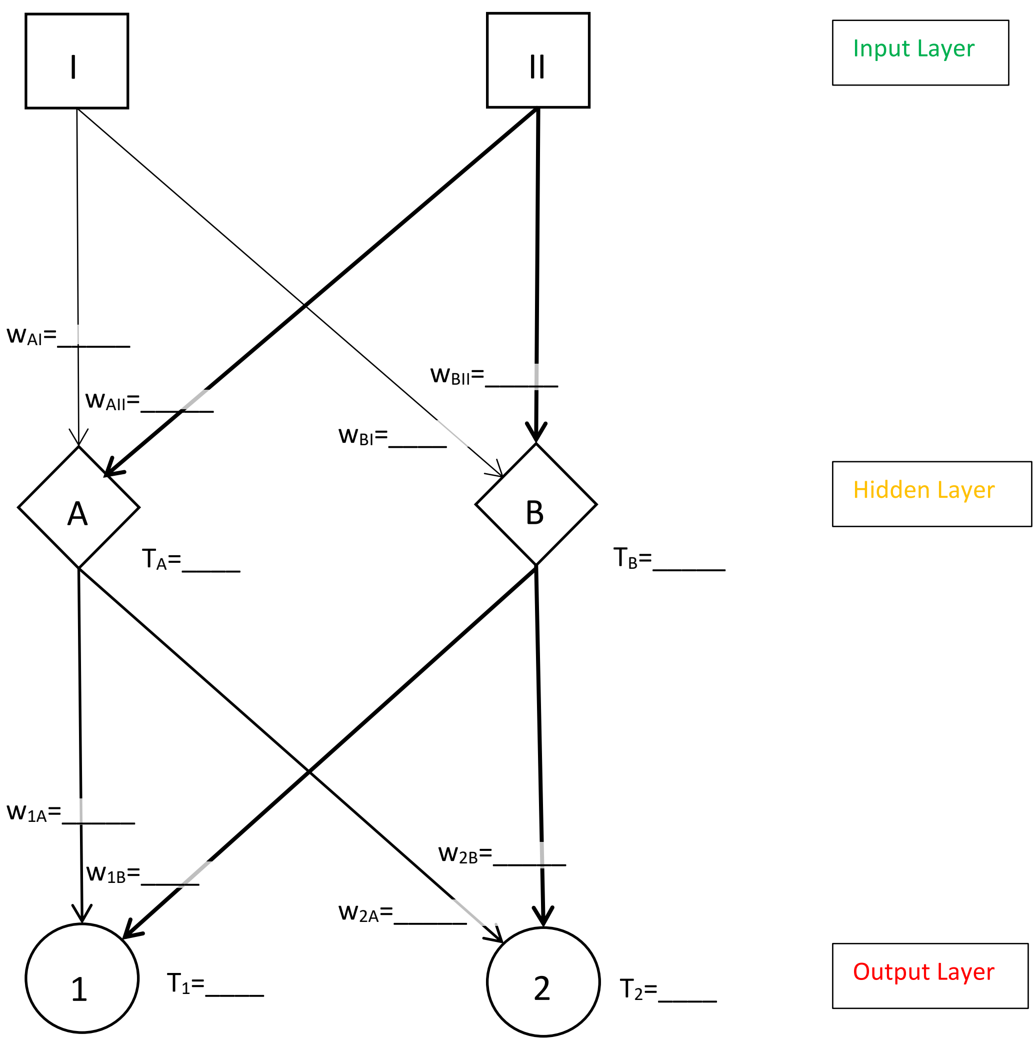
Name:______________________ Date:_________________ Period:_________

Warmup #3: Programming a 3 x 2 ANN

Directions: Work in groups of two or three to find a set of weights (W_{xi}) and thresholds (T_x) that will enable the set of inputs to produce the corresponding outputs from Table 4. Once your group finds a solution, move on the final problem of programming the complete 3 x 3 ANN.

Table 4

	Input Pattern	**Output Pattern**
1	00	00
2	10	10
3	01	10
4	11	01



Some groups may find this exercise easy after the challenge of Warmup #2, but completing this exercise will make the final task much easier. If you have made it this far, congratulations! Good luck on the Final Task!

Student Handout 4.3

Name:_______________________ Date:__________________ Period:__________

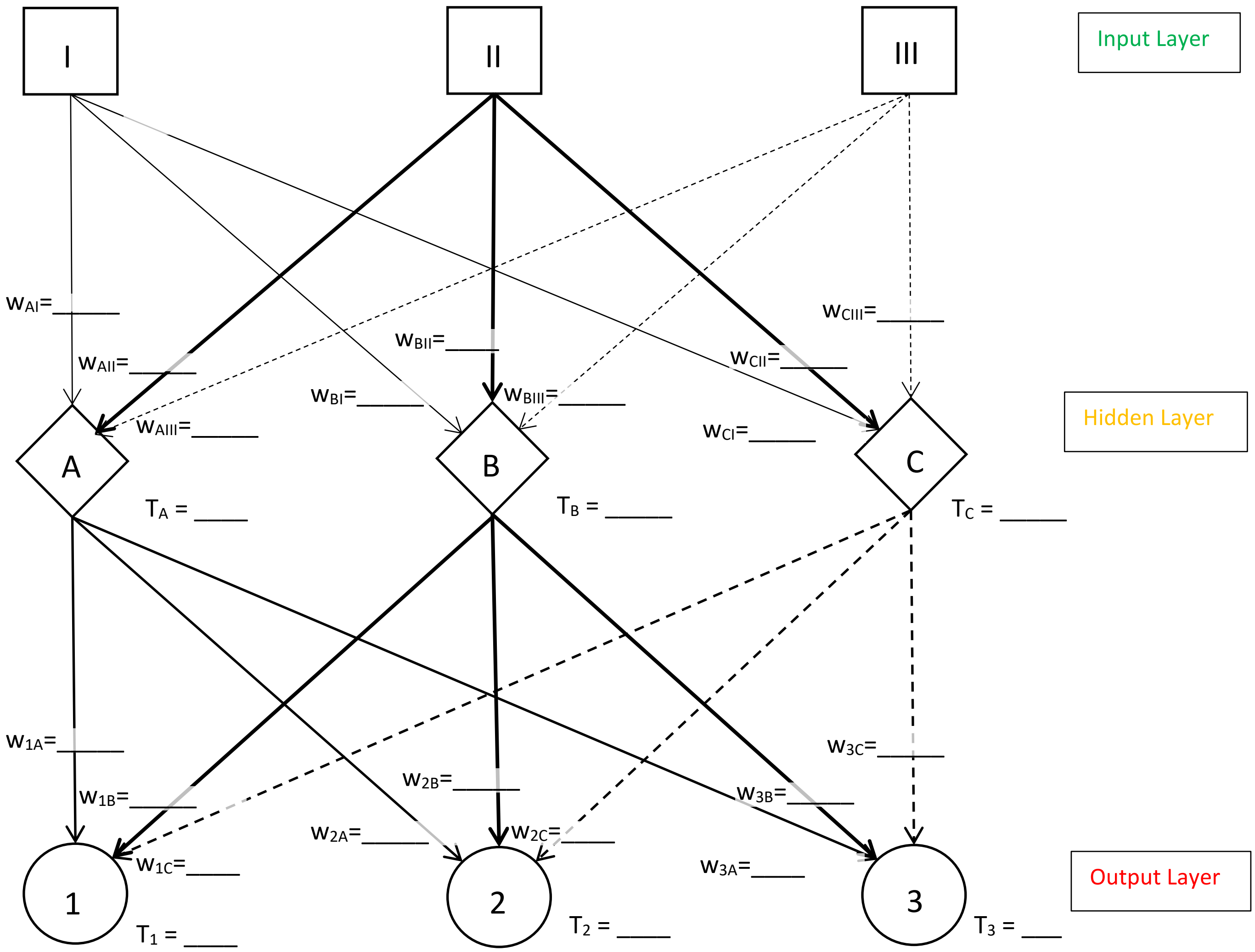
Programming Our 3 x 3 ANN

Directions: Work in groups of two or three to find a set of weights (W_{xi}) and thresholds (T_x) that will enable the set of inputs to produce the corresponding outputs from Table 1 (repeated below). Once your group finds a solution, we can program the ANN and see if it works! Can you solve the problem with a *different* set of weights? How many possible solutions do you think there are?

Table 1
Input Pattern | Output Pattern

1	000	000
2	100	100
3	010	100
4	001	100
5	110	010
6	101	010
7	011	010
8	111	001

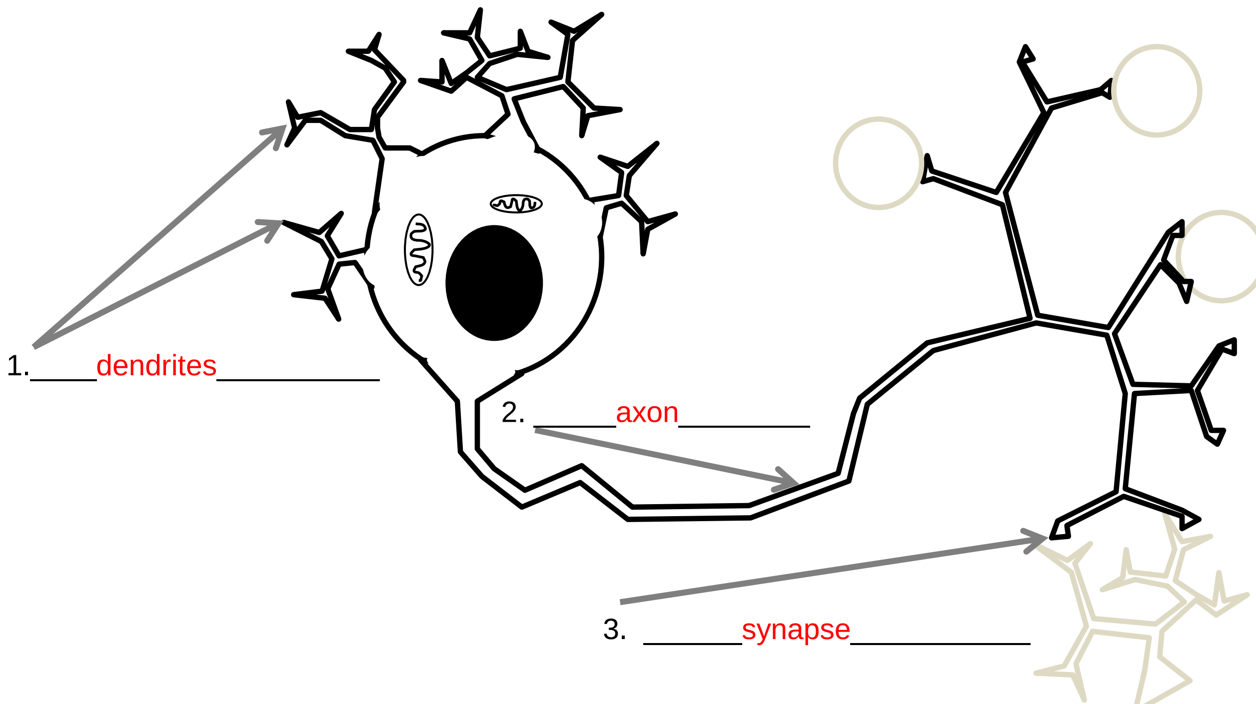
Name:_________________________ Date:____________________ Period:___________



**Teacher Resource 4.1: Assembling an Artificial Neural Network—
Teacher Answer Key**

CENTER for NEUROTECHNOLOGY

Part I: Understanding Artificial Neural Networks



1. Label the parts of a biological neuron indicated by the arrows in the diagram above. In the space below, *briefly* describe the function of each of these parts.

Dendrites receive signals from other neurons.
Axons send signals to other neurons.
Synapses are the structures where the electrical signals traveling down the axon are converted into the chemical signals that the receiving cell detects using receptor molecules in the dendrite.

2. *Briefly* explain how an Arduino can be used to create an artificial neuron.

Arduinos can simulate neurons by wiring them up to send and receive signals through their Input and Output pins, then programming them to multiply each input by a weight, add all the weighted inputs together, then compare the total to a “threshold” value to determine whether to “fire”. A neuron that is firing sets its outputs to a HIGH voltage; one that is silent keeps its outputs LOW.

3. Briefly describe the purpose of the Input, Hidden, and Output layers of an Artificial Neural Network.

Input Layer: Receives signals from outside the network and translates them into electrical signals.

Hidden Layer: Receives and processes signals from Input Layer, sending signals to Output layer.

Output Layer: Generates output signals that identify a particular pattern in the Inputs (e.g., our ANN could count how *many* Inputs there are, or some other pattern determined by the programmers).

Part II: Assembling the Artificial Neurons

The pictures and diagrams below show you how to assemble artificial neurons using Arduinos and simple breadboard circuit components (see also ***Handouts 4.2 a, b,*** or ***c***). Your teacher will assign your group one of these artificial neuron types to assemble. Work together with your group members to assemble your artificial neuron, upload the correct sketch to the Arduino, and test your neuron to make sure it is assembled and functioning correctly. Then wait for further instructions about how to add your group's neuron to the Artificial Neural Network.

Input Layer Assembly

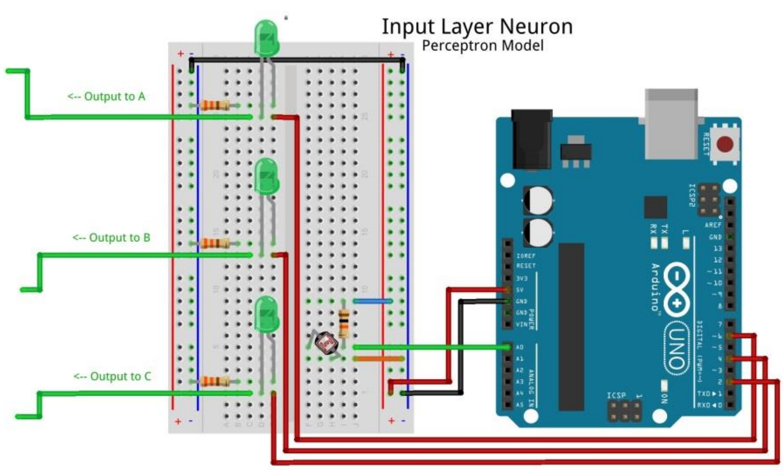
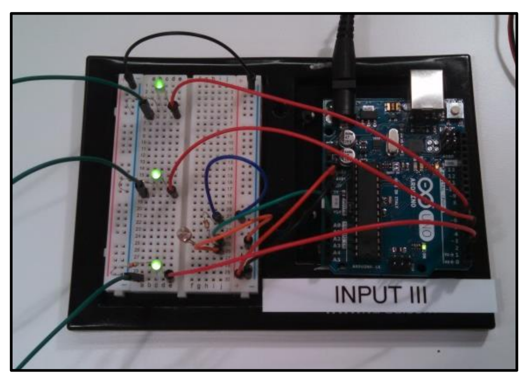


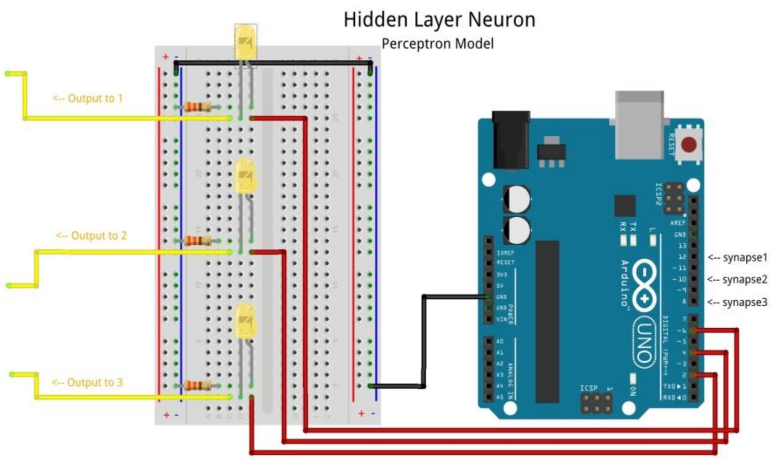
Image created with fritzing: fritzing.org



A fully assembled Input Layer Neuron

Three (3) groups will assemble **<u>Input Layer Neurons</u>**. These neurons have the most complex circuits because they receive their input from a light detector (see the lower right corner of the breadboard). The green LEDs are indicators to show that the Input Layer neuron is sending signals to Hidden Layer neurons. While the lights are on, the neuron is firing. When the lights are off, the neuron is "resting." See ***Handout 4.2a*** for additional information.

Hidden Layer Assembly



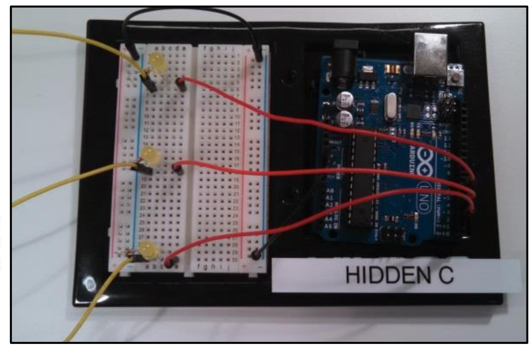
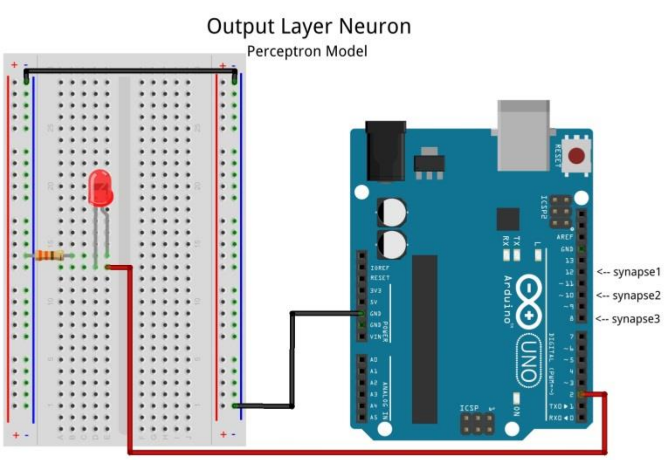


Image created with fritzing: fritzing.org

A fully assembled Hidden Layer Neuron

Three (3) groups will assemble **<u>Hidden Layer Neurons</u>**. These neurons receive their inputs from the Input Layer neurons, with pins # 8, 10 & 12 serving as the 'synapses' for these inputs (see the diagram above). The yellow LEDs are indicators to show that the Hidden Layer neuron is sending signals to Output Layer neurons. While the lights are on, the neuron is firing. When the lights are off, the neuron is "resting." See ***Handout 4.2b*** for additional information.

Output Layer Assembly



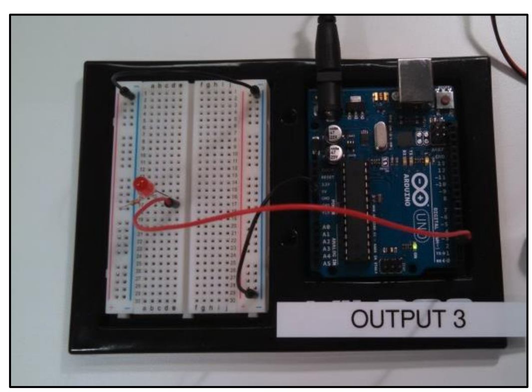


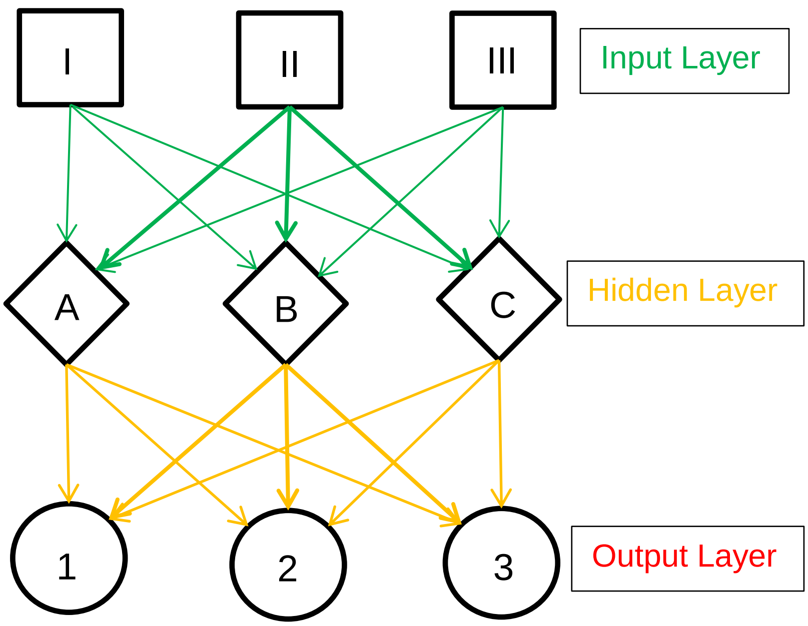
Image created with fritzing: fritzing.org

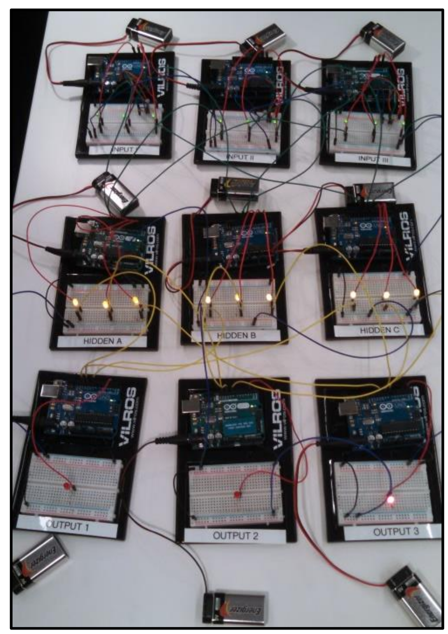
A fully assembled Output Layer Neuron

Three (3) groups will assemble **__Output Layer Neurons__**. These neurons receive their inputs from the Hidden Layer neurons, with pins # 8, 10 & 12 serving as the 'synapses' for these inputs (see the diagram above). The Output Layer neurons have the simplest circuits to assemble because they do not send outputs to other neurons. There is only a single red LED which serves as an indicator to show that the neuron is firing. See ***Handout 4.2c*** for additional information.

Part III: Assembling the Artificial Neural Network

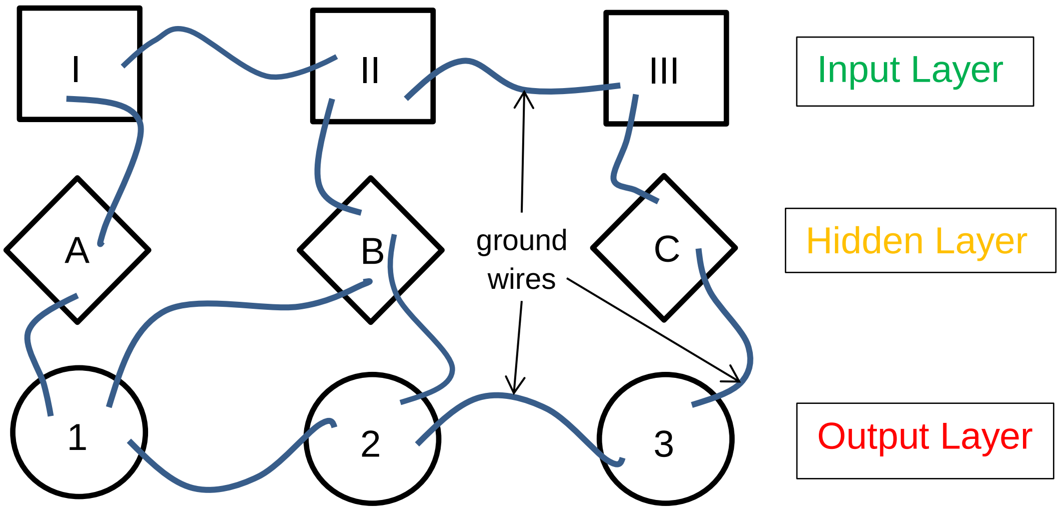
Once all 9 groups have assembled their neurons, uploaded the correct sketch and tested them, it is time to assemble the Artificial Neural Network. The schematic for assembling the network is shown on the left below, with an assembled ANN pictured on the right:





Each neuron **receives inputs** from other neurons through pins # 8, 10 and 12. Neurons **send outputs** through pins # 2, 4 and 6 to the LEDs on their breadboards. Jumper wires transmit these outputs from the breadboard to other neurons as indicated in the schematic above.

In addition to wiring the neurons together, the network must also be "**grounded**" together. This can be easily accomplished by using jumper wires to connect the ground lines (blue '-'columns on the bread boards) together, similar to the schematic below. Each neuron should be grounded to at least 2 other neurons to insure that the network will function properly.



Part IV: Understanding How an Artificial Neural Network Functions

Once the ANN is assembled with all the neurons "synapsed" and grounded together, all nine neurons should be "Firing" (i.e. their LEDs should be lit). Follow your teacher's instructions for setting all of the artificial neuron's weights and thresholds to 0.5 and exploring the behavior of the ANN. Record your observations below and answer the questions that follow:

Observations:

1. What happens when *exactly one* of the Input neurons is switched off (by covering the photoresistor)? That is, how does the behavior of the network as a whole change? Explain why this happens.

Nothing changes when only one Input is switched off. The weights and thresholds of the Hidden Layer neurons are such that it only requires two Inputs to cause them to fire.

2. What happens when two or more Input neurons are switched off? Explain.

When two or more Inputs are switched off, the entire ANN switches off. The Hidden Layer neurons require two Inputs to fire; once the Hidden Layer neurons stop firing, the Output Layer neurons also switch off.

3. Does it matter which *specific* Input neurons are on or off (or just how many)? Explain.

It does not matter which specific Inputs are on or off, only the number. This is because every neuron in each layer is programmed with identical weights and threshold, so they all behave identically.

4. At this point, our ANN can only "categorize" a few different Input Patterns. Describe the "mapping" of Input Patterns to Output Patterns of the current network configuration. By "mapping", we mean what is the pattern of Outputs from the network for each pattern of Inputs? For example, when all of the Input neurons are off, which (if any) Output neurons are switched on? What if only Input I is on, etc.?

On possible representation (an Input Pattern of '**011'** means the first Input neuron is OFF while the 2nd & 3rd are ON):

Input Pattern	Output Pattern
000	000
100	000
010	000
001	000
110	111
101	111
011	111
111	111

5. What happens if we gradually "degrade" the connectivity of the network (e.g. if we disconnect just *one* of the inputs to a single neuron)? How does each small change alter the behavior of the network? Does the behavior of the network change dramatically or only slightly with each small change? Explain.

Most small changes in the network connections have little to no effect on the ANN's behavior; this is because the connections are, in a sense, redundant in this simple network. Any two Inputs are sufficient to turn a Hidden Layer or Output Layer neuron ON, so eliminating one connection has little effect. As more connections are degraded, the behavior of the ANN changes more and more until finally it stops working altogether once a "critical number" of connections is eliminated.

Complete the Network: Before continuing your exploration, follow your instructor's directions for "programming" the ANN to categorize a specific mapping of inputs to outputs. You and your classmates will determine the weights and thresholds needed to accomplish the task, then alter each neuron's sketch accordingly and study the behavior of the network. See ***Student Handout 4.3: "Programming" An Artificial Neural Network***. When you have completed the exploration, go on to finish the Analysis and Conclusion questions below.

Part V: Analysis & Conclusions:

1. What kind of signal is sent down the axon of a biological neuron? How is this similar to and different from the signals generated by the artificial neurons in our ANN?

Neurons generate electrical signals (Action Potentials) in their axons. The signals sent by Arduinos are also electrical, but they are generated very differently. Neurons generate action potentials in bursts (or pulses) that last only a few milliseconds. Each action potential is more or less the same as every other one (in terms of its magnitude and duration), and so are essentially binary in nature (either ON or OFF). Arduinos generate analog signals which are constant – the pins are set to a voltage that ranges from 0 (ground) to +5 Volts, then remains at that level until changed by the sketch.

2. What kind of signal is sent across the synapses of biological neurons? How is this similar to and different from the way signals are received by the artificial neurons in our ANN?

Biological neurons (usually) convert electrical signals to *chemical* signals at the synapse by the release of neurotransmitter molecules into the synaptic gap. The Arduino artificial neurons use only electrical signals. However, one similarity to biological neurons is the fact that the Arduino signals can be made analog, so that the level of input can vary over a broad range of values. In biological synapses, the *amount* of neurotransmitter released with each action potential can vary in a similar manner, so that the strength of the signal received by the post-synaptic neuron is essentially analog rather than digital.

3. In a biological nervous system, some synapses are excitatory and some are inhibitory. How does an ANN create excitatory and inhibitory synapses?

Differences in the types of neurotransmitters released at synapses, and especially in the *receptors* for those neurotransmitters, can cause the post-synaptic cell to either move closer to its firing threshold (excitatory) or further from its threshold (inhibitory). The Arduino neurons achieve a similar effect using the weights by which each input is multiplied – positive weights have an excitatory effect by increasing the activation level of the artificial neuron receiving the signal, while negative weights have an inhibitory effect because they *reduce* the activation level of the neuron.

4. How do the weights assigned to each neuron's inputs affect the mapping you described in # 4 of the Observations above (i.e., how does the behavior of the individual neurons collectively 'add up' to the overall behavior of the ANN)?

Each Hidden Layer or Output neuron will only "fire" (i.e., light up) if the total of its weighted inputs exceeds its Threshold. The weights are crucial for creating the mapping of inputs to outputs. For example, if all of the weights for a given neuron's inputs were small positive values, such as 0.25, and the Threshold were 0.6, that neuron would only fire if all its inputs were set to 1. If the weights were larger positive values (or if the Threshold were lower), the neuron might fire with only one or two inputs set to 1. A single negative weight would make it harder or that neuron to fire, but only when the corresponding input is set to 1. In this way, the combinations of inputs, multiplied by various weights, creating new outputs from each neuron propagate through the network to create a specific mapping of inputs to outputs. Changing even one weight could create a “ripple effect” in the network, altering its behavior.

5. Based on the results of Observation # 5 above, how is “information” or “knowledge” about the Input Patterns stored in the neural network? For example, do individual neurons store information about specific Input Patterns, or is knowledge “distributed” across the entire network? Is it a little bit of both? How can you tell?

The manner in which the network’s behavior degrades gradually is an indication that information is stored in a distributed manner – it is not necessarily the case that one connection has a specific “meaning” (if it did, then eliminating any single connection would alter the network’s behavior significantly). At the same time, it is also true that individual neurons or groups of neurons are capable of detecting certain features of the Input pattern, and thus representing specific information not contained elsewhere in the network. For example, one way to create the mapping that counts inputs (0, 1, 2 or 3) is to have one Hidden Neuron detect when at least one Input is firing, a different Hidden neuron detect when two inputs are firing, and the last Hidden Neuron detect when all three are firing. In this way, the Hidden neurons can excite or inhibit the Output neurons in the appropriate pattern to indicate how many Inputs are active.

6. How could an ANN “learn” to generate the correct outputs without us “programing” in the weights (i.e., is there some way the network could teach *itself* how to adjust its own weights)?

ANNs can be programmed to teach themselves by first randomizing all the weights and thresholds, then running a series of Training Patterns and comparing the network’s current output to the *correct* pattern as determined by the desired mapping of inputs to outputs. All of the weights can be updated to an algorithm known as “Back Propagation” where the weights of the Output neurons are adjusted to bring their outputs closer to the correct values (e.g., decreasing the weights if the output is too high, increasing them if it is too low). These “errors” are then propagated back through the network layer by layer to determine how each neuron’s weights should be adjusted. For this purpose, neurons that use a sigmoid (or similar) activation function and a *bias* instead of a threshold work much better than the simple binary neurons used in our ANN. The reason is that the values of a sigmoid output change gradually from 0 to 1 as the weighted inputs increase. This prevents sudden changes in network behavior due to small changes in a single weight which just happens to push a neuron beyond its threshold – small changes in weights reliable cause only small changes in network behavior so that the network can gradually “converge” on a set of weights and biases that produce the desired mapping function.

Teacher Resource 4.2a: Input Layer Neuron—Arduino Sketch

// University of Washington Center for NeuroTechnology (CNT) Program
// Larry Bencivengo
// updated 1/14/19
//   - please cite CNT and the University of Washington if you use material from this sketch
//   - Copy and paste this material into a blank sketch, then save

// Input Layer Neuron
// **
//
// This sketch will allow you to program an Arduino to behave as a "Sensory Neuron" in an
// Artificial Neural Network. This "Sensory Neuron" receives input from a light sensor
// and creates outputs to other Arduinos which also serve as artificial "neurons."
//

// declare names to refer to the various pins we are using for inputs and outputs
//

const int sensorPin = 0;
const int outputPin1 = 6;
const int outputPin2 = 4;
const int outputPin3 = 2;

// "lightLevel" is the variable we will use to measure the input from the light sensor
// which is actually a device callled a photoresistor. The more light falling on the
// photoresistor, the higher its resistance (and the LOWER the level of current in the circuit).

int lightLevel, high = 0, low = 1023;

// "threshHold" is a value that you may have to determine through trial and error. The actual
// amount of current flowing through the light sensor may be slightly different for each
// "Sensory Neuron" in your Network. The LOWER the value of "threshHold", the MORE light is
// required to make our "Sensory Neuron" <FIRE> and send a signal to other "neurons" in the Network.
//
// **
// If the LEDs indicating that the "Sensory Neuron" is Firing do not turn on when you load the script,
// you may need to INCREASE the value of "threshHold." (try 700 or even 1000).
// If the LEDs do not turn off when you cover the light sensor, try decreasing the value of "threshHold"

```
// to 300 or 200 (but not less than 0).
// ****************************************************************************************************

int threshHold = 500;


void setup()
{
  // We'll set up the outputPins to be outputs.
  // (We don't need to do anything special to use the analog input.)

  pinMode(outputPin1, OUTPUT);
  pinMode(outputPin2, OUTPUT);
  pinMode(outputPin3, OUTPUT);
}


void loop()
{
  lightLevel = analogRead(sensorPin);

// Read the lightLevel from the light sensor.  If it is below the threshHold, the neuron should
<FIRE>,
// i.e. send signals to other neurons through the output pins.  Otherwise, the neuron should
remain quiet.

if (lightLevel < threshHold)
{
  digitalWrite(outputPin1, HIGH);
  digitalWrite(outputPin2, HIGH);
  digitalWrite(outputPin3, HIGH);
}
else
{
  digitalWrite(outputPin1, LOW);
  digitalWrite(outputPin2, LOW);
  digitalWrite(outputPin3, LOW);
}

}
```

Teacher Resource 4.2b: Hidden Layer Neuron—Arduino Sketch

```
//  University of Washington Center for NeuroTechnology (CNT) Program
//  Larry Bencivengo
//  updated 1/14/19
//     - please cite CNT and the University of Washington if you use material from this sketch
//     - Copy and paste this material into a blank sketch, then save


// HIDDEN LAYER NEURON - Perceptron Model
// ************************************************************************************************
//
// This sketch will allow you to program an Arduino to behave as an "Interneuron" in an
// artificial Neural Network. This "Interneuron" receives inputs from other Arduinos serving as
// artifical "Sensory Neurons" and creates outputs to other Arduinos serving as "Motor Neurons".
// These "Interneurons" form what is called the Hidden Layer of an artificial Neural Network.
//
// This script uses the "Perceptron" model for artificial neurons, where each neuron's output is
// binary (that is, either 0 or 1).  A Perceptron multiplies each of its inputs by a different weight,
// then sums the results to determine its Activation level.  If the neuron's Activation exceeds its
// Threshold, the neuron <FIRES>. Perceptrons mimic the "all or none" nature of the Action
Potentials
// of real neurons.  However, networks of Perceptrons are difficult to "train" to accomplish
specific
// tasks, so neural networks more commonly use the Sigmoid model for neurons.  See the
Sigmoid Model
// scripts to build a trainable artificial Neural Network.
//


// Declare Variables
// ************************************************************************************************

// Declare names to refer to the pins receiving inputs from other Arduinos.
// These pins serve as artificial "synapses" on our "Interneuron", so we will call them "synapse1",
// "synapse2", etc.

const int synapse1 = 12;
const int synapse2 = 10;
const int synapse3 = 8;

// Declare names to refer to the pins we are using for the outputs of our "Interneuron".
```

```
const int outputPin1 = 6;
const int outputPin2 = 4;
const int outputPin3 = 2;

// Declare names to refer to the values of the inputs at each "synapse"

int inputLevel1;
int inputLevel2;
int inputLevel3;

// Declare variables  activationLevel, threshold and the weights for each synapse
float activationLevel;
float threshold;
float synapse1Weight;
float synapse2Weight;
float synapse3Weight;

// *********************************************************************************************
//                                   TEST MODE
// *********************************************************************************************
//
// Your Hidden Layer Neuron will automatically be in "Test Mode" unless you set the Boolean
variable
// Testing to FALSE.
// Test Mode allows you to check that you have hooked up your neuron's circuit correctly and
that
// the Sketch has loaded properly.
// In Test Mode, all 3 of the neuron's Outputs should blink.  Once your Hidden Layer Neuron is
// functioning properly, change Testing to FALSE and re-upload the Sketch to your Arduino.
//

boolean Testing = true;
boolean ON = false; // this variable is used to make the LEDs blink on and off
//
// *********************************************************************************************


void setup()
{
  // set the various pins as inputs or outputs
  pinMode(synapse1, INPUT);
  pinMode(synapse2, INPUT);
  pinMode(synapse3, INPUT);
  pinMode(outputPin1, OUTPUT);
```

```
  pinMode(outputPin2, OUTPUT);
  pinMode(outputPin3, OUTPUT);
}

void loop()
{

  // *********************************************************************************************
  //                              Set Threshold and SynapseWeights
  // *********************************************************************************************
  //
  threshold = 0.5f;
  synapse1Weight = 1.0f;
  synapse2Weight = 1.0f;
  synapse3Weight = 1.0f;

  //
  // *********************************************************************************************
  //

  // reset the activationLevel to zero each time through the loop
  activationLevel = 0;

  // check for Test Mode
  if (Testing)
  {
    //if Testing, make LEDs blink.
    delay (1000);

    if (ON)
    {
      // turn LEDs off
      inputLevel1 = LOW;
      inputLevel2 = LOW;
      inputLevel3 = LOW;
      ON = false;
    }
    else
    {
      // turn LEDs on
      inputLevel1 = HIGH;
      inputLevel2 = HIGH;
      inputLevel3 = HIGH;
      ON = true;
```

```
    }
  }
  else
  {
    // if NOT Testing, read inputs from synapses
    inputLevel1 = digitalRead(synapse1);
    inputLevel2 = digitalRead(synapse2);
    inputLevel3 = digitalRead(synapse3);
  }

  // ****************************************************************************************
  //
  // determine the new activationLevel by multiplying the input at each synapse by its
  // synapseWeight and summing the results
  activationLevel =
    (inputLevel1 * synapse1Weight) +
    (inputLevel2 * synapse2Weight) +
    (inputLevel3 * synapse3Weight);

  // determine whether the neuron <FIRES> by comparing the activationLevel to the threshold
  // A Perceptron's output is digital (either 0 or 1)
  if (activationLevel > threshold)
  {
    // if activationLevel exceeds threshold, neuron <FIRES>
    // Set all outputs to 1
    digitalWrite(outputPin1, HIGH);
    digitalWrite(outputPin2, HIGH);
    digitalWrite(outputPin3, HIGH);
  }
  else
  {
    // if activationLevel <= threshold, its ouput is zero (at rest)
    // Set all outputs to 0
    digitalWrite(outputPin1, LOW);
    digitalWrite(outputPin2, LOW);
    digitalWrite(outputPin3, LOW);
  }

}
```

Teacher Resource 4.2c: Output Layer Neuron—Arduino Sketch

// University of Washington Center for NeuroTechnology (CNT) Program
// Larry Bencivengo
// updated 1/14/19
//   - please cite CNT and the University of Washington if you use material from this sketch
//   - Copy and paste this material into a blank sketch, then save

// OUTPUT LAYER NEURON - Perceptron Model
// ***
//
// This sketch will allow you to program an Arduino to behave as a "Motor Neuron" in an
// artificial Neural Network. This "Motor Neuron" receives inputs from other Arduinos serving as
// artificial "neurons" and creates an output to an LED (or other device) to show that it has
<FIRED>.
//
//
// This script uses the "Perceptron" model for artificial neurons, where each neuron's output is
// binary (that is, either 0 or 1). A Perceptron multiplies each of its inputs by a different weight,
// then sums the results to determine its Activation level. If the neuron's Activation exceeds its
// Threshold, the neuron <FIRES>. Perceptrons mimic the "all or none" nature of the Action
Potentials
// of real neurons. However, networks of Perceptrons are difficult to "train' to accomplish
specific
// tasks, so neural networks more commonly use the Sigmoid model for neurons. See the
Sigmoid Model
// scripts to build a trainable artificial Neural Network.
//

// Declare Variables
// ***

// Declare names to refer to the pins receiving inputs from other Arduinos.
// These pins serve as artificial "synapses" on our "Motor Neuron", so we will call them
"synapse1",
// "synapse2", etc.

```
const int synapse1 = 12;
const int synapse2 = 10;
const int synapse3 = 8;
```

```
// Declare name to refer to the pin we are using for the output of our "Motor Neuron".

const int outputPin = 2;

// Declare names to refer to the values of the inputs at each "synapse"

int inputLevel1;
int inputLevel2;
int inputLevel3;

// Declare variables  activationLevel, threshold and the weights for each synapse
float activationLevel;
float threshold;
float synapse1Weight;
float synapse2Weight;
float synapse3Weight;


// **********************************************************************************************************
//                                   TEST MODE
// **********************************************************************************************************
//
// Your Output Layer Neuron will automatically be in "Test Mode" unless you set the Boolean
variable
// Testing to FALSE.
// Test Mode allows you to check that you have hooked up your neuron's circuit correctly and
that
// the Sketch has loaded properly.
// In Test Mode, the neuron's Output LED should blink.  Once your Output Layer Neuron is
// functioning properly, change Testing to FALSE and re-upload the Sketch to your Arduino.
//

boolean Testing = true;
boolean ON = false; // this variable is used to make the LEDs blink on and off
//
// **********************************************************************************************************



void setup()
{
  // set the various pins as inputs or outputs
  // our Motor Neuron only has one output (it is either <FIRING> or resting)
  pinMode(synapse1, INPUT);
```

```
  pinMode(synapse2, INPUT);
  pinMode(synapse3, INPUT);
  pinMode(outputPin, OUTPUT);

}


void loop()
{

  // ********************************************************************************************
  //                              Set Threshold and SynapseWeights
  // ********************************************************************************************
  //
  threshold = 0.5f;
  synapse1Weight = 1.0f;
  synapse2Weight = 1.0f;
  synapse3Weight = 1.0f;
  //
  // ********************************************************************************************


  // reset the activationLevel to zero each time through the loop
  activationLevel = 0;

  // check for Test Mode
  if (Testing)
  {
    //if Testing, make LED blink.
    delay (1000);

    if (ON)
    {
      // turn LED off
      inputLevel1 = LOW;
      inputLevel2 = LOW;
      inputLevel3 = LOW;
      ON = false;
    }
    else
    {
      // turn LED on
      inputLevel1 = HIGH;
      inputLevel2 = HIGH;
```

```
      inputLevel3 = HIGH;
      ON = true;
    }
  }
  else
  {
    // if NOT Testing, read inputs from synapses
    inputLevel1 = digitalRead(synapse1);
    inputLevel2 = digitalRead(synapse2);
    inputLevel3 = digitalRead(synapse3);
  }

  // determine the new activationLevel by multiplying the input at each synapse by its
  // synapseWeight and summing the results
  activationLevel =
    (inputLevel1 * synapse1Weight) +
    (inputLevel2 * synapse2Weight) +
    (inputLevel3 * synapse3Weight);

  // determine whether the neuron <FIRES> by comparing the activationLevel to the threshold
  // A Perceptron's output is digital (either 0 or 1)
  if (activationLevel > threshold)
  {
    // if activationLevel exceeds threshold, neuron <FIRES>
    // Set output to 1
    digitalWrite(outputPin, HIGH);
  }
  else
  {
    // if activationLevel <= threshold, its ouput is zero (at rest)
    // Set all outputs to 0
    digitalWrite(outputPin, LOW);
  }

}
```

Teacher Resource 4.3: "Programming" an Artificial Neural Network—Teacher Answer Key

Artificial Neural Networks (ANNs) have many current and potential applications. They are especially useful for identifying or "categorizing" complex patterns (e.g., face recognition software, search engines, or "data mining" on the internet). In these activities, you will learn how ANNs work by figuring out how to "program" our network to accomplish a specific task.

The Goal: We want our ANN to identify whether 0, 1, 2 or 3 of the input layer neurons is on, indicating this by lighting up a different output layer neuron in each case (none, #1 only, #2 only, or #3 only, respectively).

Table 1
Input Pattern | Output Pattern

1	000	000
2	100	100
3	010	100
4	001	100
5	110	010
6	101	010
7	011	010
8	111	001

Inputs and Outputs: We can formally represent this mapping as follows (in the Input and Output Patterns shown below, an entry such as '100' indicates that the *first* neuron of that layer has an output of 1 while the other two neurons have output 0; '010' would indicate that the *second* neuron has output 1 while the first and third neurons have output 0, etc.).

Weights and Thresholds: Remember that each neuron's output is either 1 or 0, and that each neuron multiplies each of its inputs by a different weight, adds the weighted inputs together, then compares the sum to its threshold. If the total is *above* the threshold, it sets its output to 1, otherwise its output is 0. This is summarized by the following equations:

a. If $\sum_{i=1}^{3} weight_i \times input_i > threshold$, set output = 1

b. If $\sum_{i=1}^{3} weight_i \times input_i <= threshold$, set output = 0

For example, if we were using Input Pattern # 5 (110), Hidden Neuron A would receive inputs of 1 from Input Neurons I & II, and 0 from Input Neuron III. If Hidden Neuron A's weights were w_1 = 0.25, w_2 = - 0.33 and w_3 = 0.47, the total input to Hidden Neuron A would be (0.25 x 1.0) + (- 0.33 x 1.0) + (0.47 x 0) = 0.25 – 0.33 + 0 = - 0.08. If Hidden Neuron A had a threshold of 0.5, its output would be set to 0 because the total input is less than the threshold. Therefore Hidden Neuron A will not send messages to any of the output layer neurons.

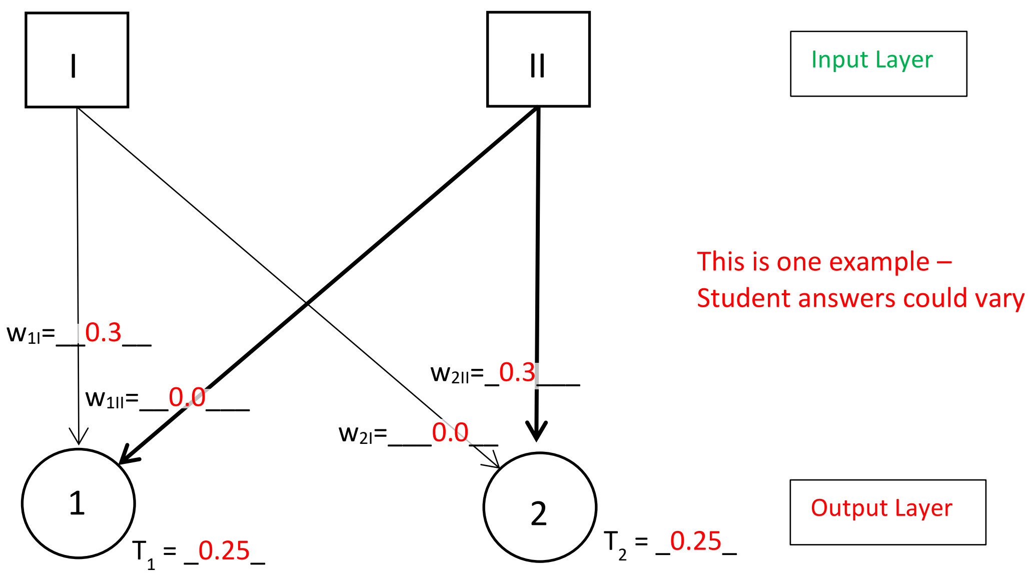
Procedure: Our ultimate goal is to work out a set of weights for each neuron that will result in the mapping of Input Patterns to Output Patterns described above. However, this would be a very challenging task without a bit of practice. The following set of exercises will help you to develop several important insights into how ANNs work that should it make it possible for you to "program" our ANN with a set of synaptic weights that will allow the ANN to correctly categorize inputs as described above.

Warmup 1: Programming a 2 x 2 ANN

Directions: This activity is a warmup exercise for programming the full network (see the last page). Work in groups of two or three to find a set of weights (W_{xi}) and thresholds (T_x) that will enable the set of inputs to produce the corresponding outputs shown in Table 2 below. Notice that this simple ANN has no Hidden Layer. Once your group finds a solution, move on to Warmup 2.

Table 2

	Input Pattern	Output Pattern
1	00	00
2	10	10
3	01	01
4	11	11



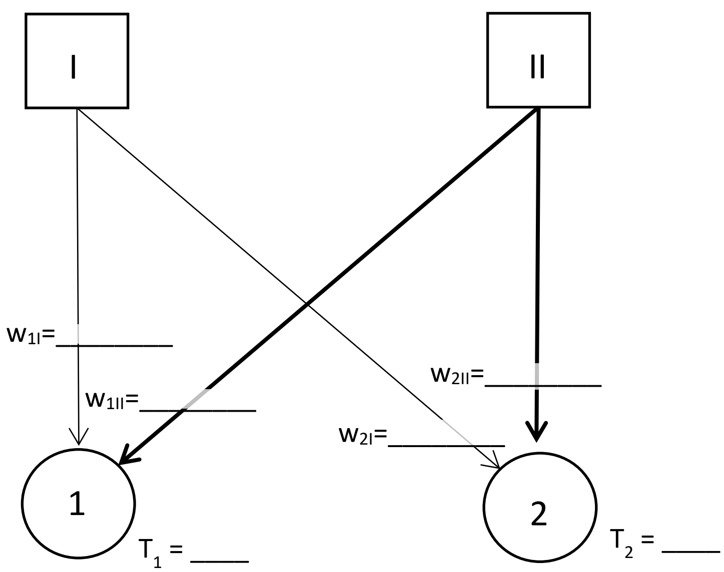
For some students, programming this Input → Output mapping into the 2 x 2 network will seem simple; for others it may be very challenging. Students who complete this task easily should give *hints* to other students without simply revealing the answer.

Warmup 2: Programming a 2 x 2 ANN

Directions: This second warmup exercise is much more challenging. Work in groups of two or three to find a set of weights (W_{xi}) and thresholds (T_x) that will enable the set of inputs to produce the corresponding outputs shown in Table 3 below. Is it possible?

Table 3

	Input Pattern	Output Pattern
1	00	00
2	10	10
3	01	10
4	11	01



It is impossible to complete this task – There is no combination of weights and Thresholds that will produce the desired mapping! See explanation below.

This type of simple ANN with only two layers of neurons is limited in its ability to learn certain patterns. The type of pattern required to create the correct Input → Output mapping is called the "Exclusive Or" (XOR) function in Boolean logic (The Boolean XOR function receives two inputs. Its output is 1 if either of the individual inputs is 1, but the output is 0 if both inputs are 0 ***or*** if both inputs are 1). It has been shown mathematically that 2-layer ANNs are incapable of learning the XOR function! Our 3 x 3 ANN may not seem very complex, but it *is* capable of learning the XOR function, which is very useful for categorizing input patterns.

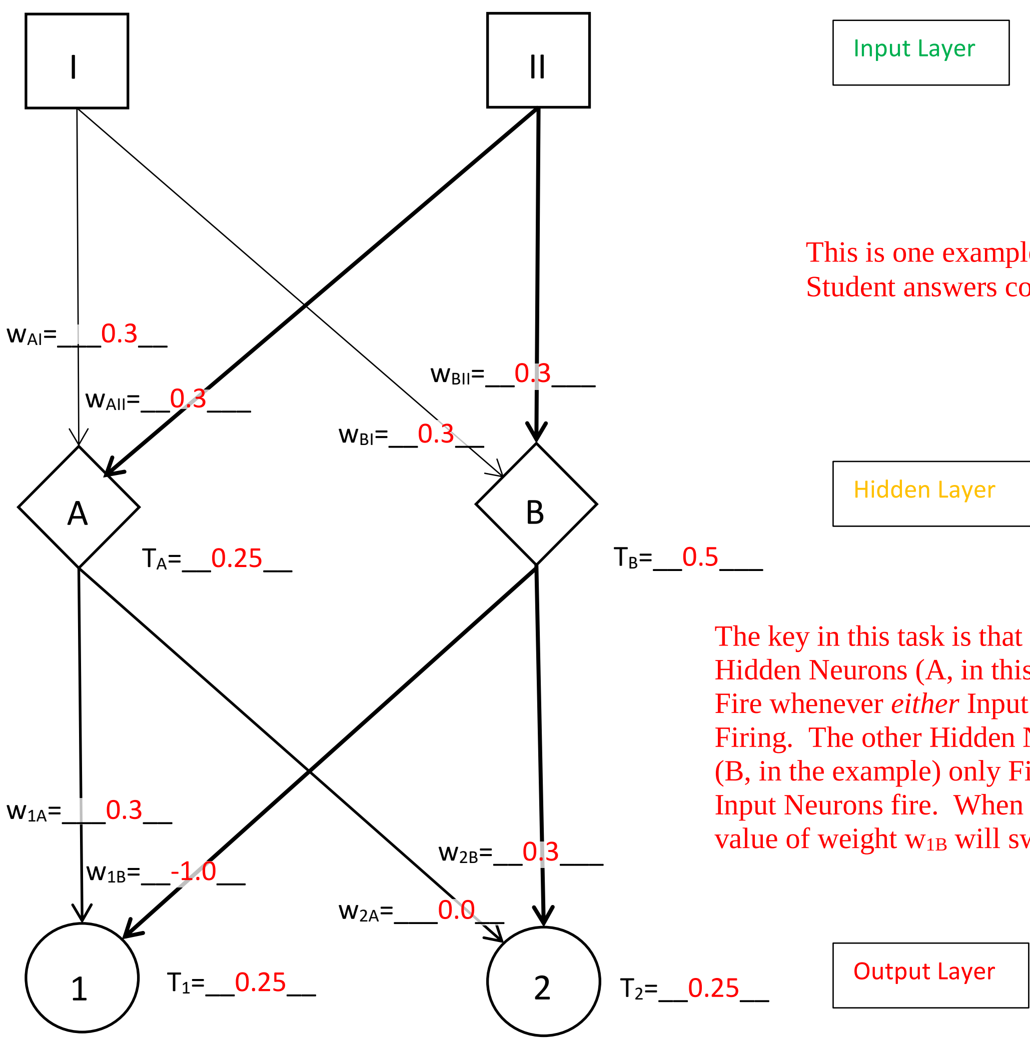
CENTER for NEUROTECHNOLOGY

Warmup #3: Programming a 3 x 2 ANN

Directions: Work in groups of two or three to find a set of weights (W_{xi}) and thresholds (T_x) that will enable the set of inputs to produce the corresponding outputs from Table 4. Once your group finds a solution, move on the final problem of programming the complete 3 x 3 ANN.

Table 4

	Input Pattern	**Output Pattern**
1	00	00
2	10	10
3	01	10
4	11	01



This is one example –
Student answers could vary.

The key in this task is that one of the Hidden Neurons (A, in this case) will Fire whenever *either* Input neuron is Firing. The other Hidden Neuron (B, in the example) only Fires when *both* Input Neurons fire. When B Fires, the negative value of weight w_{1B} will switch Output 1 off.

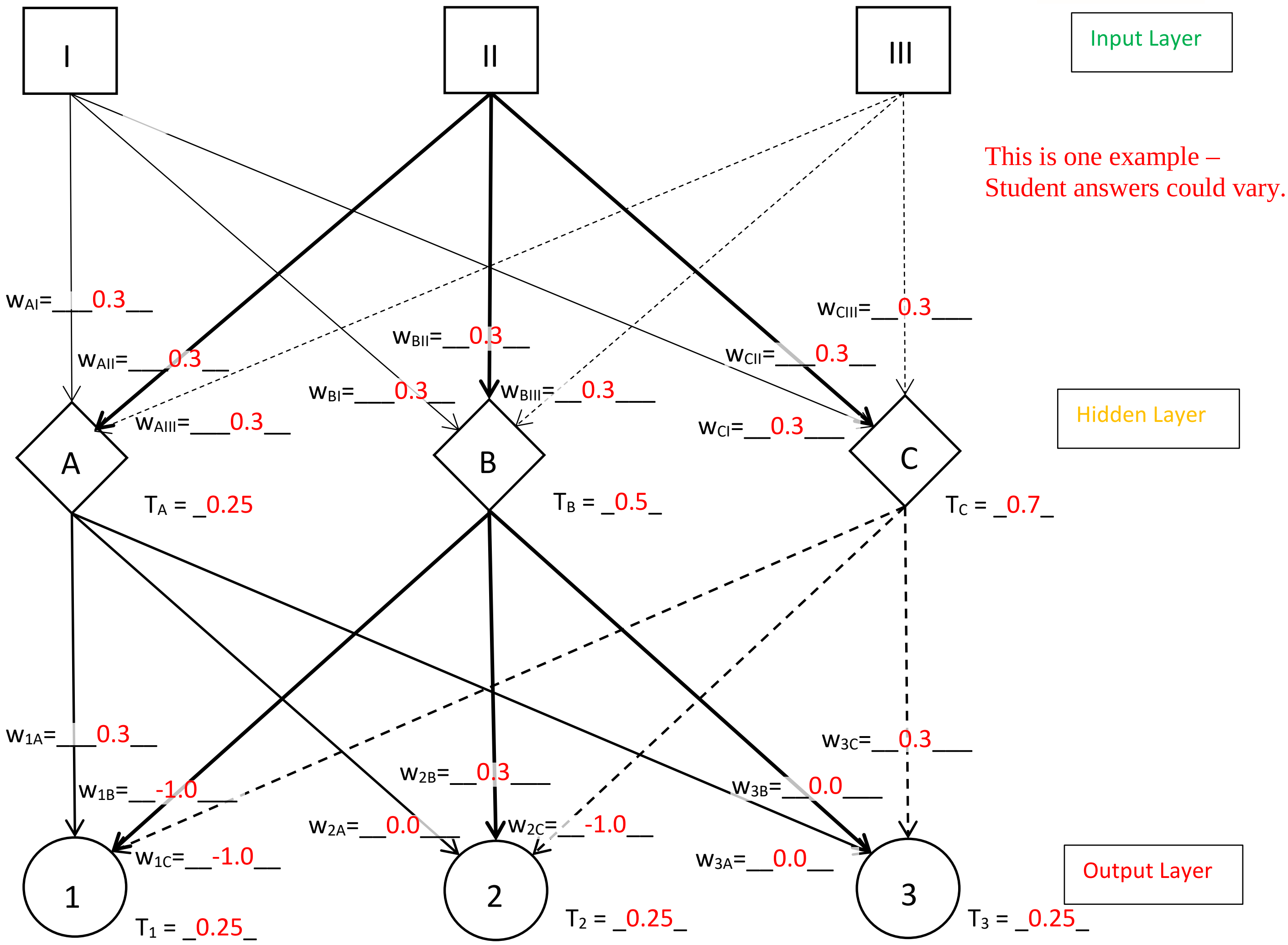
Some groups may find this exercise easy after the challenge of Warmup #2, but completing this exercise will make the final task much easier. If you have made it this far, congratulations! Good luck on the Final Task!

Programming Our 3 x 3 ANN

Directions: Work in groups of two or three to find a set of weights (W_{xi}) and thresholds (T_x) that will enable the set of inputs to produce the corresponding outputs from Table 1 (repeated below). Once your group finds a solution, we can program the ANN and see if it works! Can you solve the problem with a *different* set of weights? How many possible solutions do you think there are?

Table 1
Input Pattern | Output Pattern

1	000	000
2	100	100
3	010	100
4	001	100
5	110	010
6	101	010
7	011	010
8	111	001


I
II
III
Input Layer
This is one example – Student answers could vary.
w_{AI}=___0.3__
w_{AII}=___0.3__
w_{AIII}=___0.3__
w_{BI}=___0.3__
w_{BII}=__0.3__
w_{BIII}=__0.3___
w_{CI}=__0.3___
w_{CII}=___0.3__
w_{CIII}=__0.3___
Hidden Layer
A
B
C
T_A = _0.25
T_B = _0.5_
T_C = _0.7_
w_{1A}=___0.3__
w_{1B}=__-1.0___
w_{1C}=__-1.0__
w_{2A}=__0.0___
w_{2B}=__0.3___
w_{2C}=__-1.0__
w_{3A}=__0.0__
w_{3B}=__0.0___
w_{3C}=__0.3___
1
2
3
Output Layer
T_1 = _0.25_
T_2 = _0.25_
T_3 = _0.25_