Title
User Experience Evaluation of the Simbrain Neural Network Simulator

Author
Gamino, Stephanie K

Publication Date
2022-08-15

UNIVERSITY OF CALIFORNIA, MERCED

User Experience Evaluation of the Simbrain Neural Network Simulator

A Thesis submitted in partial satisfaction of the requirements for the degree of Master of Science

in

Cognitive and Information Sciences

by

Stephanie Kristin Gamino

Committee in charge:

Professor Jeff Yoshimi, Chair
Professor Heather Bortfeld
Professor Michael Spivey

2022

The Thesis of Stephanie Kristin Gamino is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Heather Bortfeld

Michael Spivey

Jeff Yoshimi, Chair

University of California, Merced

2022

I dedicate this thesis and everything that I've been able to accomplish to my parents, Grecia Corea and Carlos Argueta. Your support and love are the reason I'm here today. Soy muy afortunada de tener padres como ustedes.

También me gustaría dedicarle esto a mi angel Mercedes Pérez. No sería la persona que soy hoy en día sin ti. Siempre te amare.

I couldn't have don't this without the support of my close friends and family. I would also like to dedicate this to: Brianna Gamino, Roy and Yaret Corea, Jose and Eva Corea, Jacqueline & Ricky Cifuentes, Jose Perez, Alvaro Gamino & Suzanne Sotelo, Carlos Lopez, Atziry Madrigal, Citlali Perez, Jennifer Mendez, Amy Pimentel, Giselle Barajas, Taylor Moua, Juliana Ayala, and Ruben Sanchez. Forever thankful for you and your endless love.

Table of Contents

List of Figures

Acknowledgements

I would like to acknowledge my committee members, Professor Jeff Yoshimi, Professor Heather Bortfeld, and Professor Michael Spivey for your continued support throughout my graduate school education and experience. Throughout my undergraduate education, I have taken at least one class from you all and I am honored to have worked with you and even more honored to have you as part of my committee.

I would also like to acknowledge the entirety of the Cognitive and Informational Science department. I have never met such influential and kind people. Although we come from different walks of life, I appreciate the guidance and laughs we shared. I would like to also thank my cohort. I couldn't have made to get through this program without our collaborative efforts. You all are a huge inspiration and I cannot thank you enough for molding me into someone I didn't think I could become.

I would like to specifically acknowledge my Lacanian brothers: Adam Holm, Cody Moser, Chanuwas Aswamenakul and Dennis Perez. You all are my besties and I'm so glad our paths crossed. Forever thankful for the bond we have and the many memories we've created.

Lastly, I would like to thank my students who partook in the UNC Frontiers of Science Program. You all are so smart and I admire your passion and drive. We will forever be THE DENDRITES!

Abstract

User Experience Evaluation of the Simbrain Neural Network Simulator

by Stephanie Gamino for the partial satisfaction of the requirements for the degree of Master of Science in Cognitive and Information Sciences, University of California, Merced, 2022

Dr. Jeff Yoshimi, Chair

Don Norman, the author of *The Design of Everyday Things*, once said, “everything is designed.” Our world is surrounded by many physical and digital products, each with its own purpose. The study of design emphasizes making advancements to products and keeping in mind the ease of users' experience. User experience design provides a way to improve products while also creating better usability. This article will provide an overview of user experience and its different applications, along with a review of an educational application called Simbrain. Simbrain was used to teach a summer program called Frontier of Science at the University of Northern Colorado in Summer 2022. Experiences from this program will be described here.

Introduction

When a user fails to use a device properly, the blame is sometimes placed on the machine for not working, but in reality, failure also follows from the user not understanding the machine. In this scenario, the user is not at fault but rather the design itself. User experience design aims to create simple, understandable, intuitive products for users.

The first section of this article will provide a general background on user experience design. It defines what user experience is and how it can be broken down into subsections of interaction design and its heuristic rules of thumb. Section two summarizes the purpose and use of the interactive educational tool called Simbrain (Tosi & Yoshimi 2016). The third section, Classroom Experience Teaching with Simbrain, briefly overviews topics covered when teaching high school students neural networks. Simbrain was used to further their understanding of neural network concepts. Section four presents the students' review of Simbrain's user design features and recommendations to improve Simbrain. Lastly, section five gives a specific user experience analysis of one user interface component of Simbrain called, The Smile classifier. This section will describe how to create this simulation in Simbrain and what changes can be made to make the component more intuitive for users.

Background

User Experience

User experience is the process of creating products that are not only easy to use but also facilitate an enjoyable interaction between the product and the user. Two main components contribute to a good design: discoverability and understanding (Norman

2013). These two concepts allow for the critical assessment and improvement of a product’s inherent design qualities. Discoverability poses questions on how to use the product. Specifically, if it's possible to figure out how to use the product and consider what actions are needed for its use. On the other hand, understanding the product results in users being able to decipher what is being presented to them, what the different controls mean, and ultimately how the product is supposed to be used. Design applies not just to technological products, but to all physical entities such as clothing, furniture, doors, and many others. Design doesn’t happen just once, it takes a lot of iterations to improve a product. Good design is re-designing.

There are three specialization in design: experience, industrial, and interaction (Norman 2013). Experience design emphasizes emotional impact. The main goal of experience design is to create a product that will create an overall enjoyable experience for the user. Industrial design emphasizes the form and material of actual design to optimize functionality. Lastly, interaction design focuses on usability. Usability within interaction design depends on how a user will connect with the product. This means ensuring users understand what can be done to use the product. This specialization of design relies on other topics of psychology, design, and art to create an enjoyable experience for users.

Interaction Design

Within interactive design, 5 dimensions contribute to making the interaction between the product and user an effortless and holistic experience. The first four dimensions were introduced as “interaction design language” by Professor Gillian

Crampton Smith (Smith 2007), and interaction designer Kevin Silver added the last dimension, behavior (Moggridge 2007). The first dimension has visible *words* that can help users navigate and give them the correct information. *Visual representation* is the second dimension that includes having images or graphical elements. The third dimension has a physical object or space that reflects the medium through which the user will interact with the product. An example of this is using programming software applications via a desktop computer. The fourth dimension, *time*, relates to certain media changing over time within the product. This temporal dimension is critical in digital products as animations or sound design in computer applications are used to communicate intent and proper working status. Lastly, *behavior* reflects how the other dimension plays a role in how the user will interact. This fifth dimension mainly corresponds to user behavior when using products and how the products will provide feedback based on their input.

Interaction design is sometimes referred to as user experience design. User experience, or UX, focuses on shaping the experience of users, while interaction design uses interactions with design to advance the experience further. UX's primary goal is to improve the overall experience of the user's entire journey. Interaction design dictates how one will use the product and what improvements are needed for their interactive experience. Interaction design is a more direct aspect of how a user can interact with a product.

Heuristics

The Nielsen Norman Group created a set of engineering heuristics or possible rules that can be used to evaluate user interfaces and identify a user interface's design problems (Nielsen 2020). These 10 general rule principles offer support when creating interactive designs. Although these 10 principles are an excellent way to evaluate a design, not all 10 can be applied to every interaction design. For this article, only six heuristics are relevant for Simbrain.

One heuristic principle is the *visibility* of a system's status, where the design is meant to give feedback to the user. Users are able to learn the outcome of their prior interaction and, from there, can decide how to move forward. For example, consider the Apple iPhone. Apple designs products such as the iPhone that permit users to type on a glass surface. Within their design, Apple added multi-sensory feedback while a user is typing by providing clicking sounds and having the touched key grow larger. This provides the user with feedback by showing them that the key has been touched. Another example is unlocking the iPhone. It will let users know when the passcode is incorrect by vibrating but also visually shaking the box where the passcode was inputted. *Matching* focuses on the close comparison of a system and the real world. With this heuristic, the design follows real-world concepts such as words and phrases. *User control and freedom* focus on allowing users to exit, undo, and back out of an action. This can also include showing a straightforward way to exit their current path and return to another. There are a variety of apps an iPhone can hold, and although design varies per application, the default apps made by Apple all follow similar designs of arrows and exits. There are two types of *errors* a user can create, slips and mistakes. Slips tend to be unconscious errors caused by the user's inattention, while mistakes are errors based on the user's incomplete or incorrect information for a particular task or the initial design. *Flexibility and efficiency* emphasize allowing users to tailor content through keyboard shortcuts. Circling back to the iPhone example, users can add shortcuts to their iPhone to help with different tasks. An example of this is a text message shortcut, where a user schedules a message to be sent to a specific recipient based on their settings of time, date, and how often to send those scheduled messages. Many users prefer and benefit from an *aesthetic and minimalist design*. This design feature does not promote a flat or dull design but rather moves away from a busy design and focuses on the essential components. These usability heuristics are not meant to be used as specific guidelines but as a resource for maximum efficiency when creating interactive designs. This article will focus on a user experience review of a computational, interactive tool called Simbrain, as well as my experience using Simbrain as a teaching tool for high school students.

Simbrain

Simbrain is a tool for creating and analyzing artificial neural networks through computational simulations. Artificial neural networks are computer models of biological neural networks in the brain. Studies in artificial neural networks have recently been prominent in machine learning and psychology. Simbrain is an educational tool for building and creating different networks. This tool offers a variety of neuron models ranging from *decaying activation* to *Sigmoidal function*, as well as many models of synaptic plasticity and synaptic transmission.

There are different way a neural network can interact with an environment. Simbrain provides a two-dimensional virtual environment that can contain simulated organisms such as a mouse, cow, or a human. These organisms can have different sensors, such as smell and hearing, and the sensors have a specific location relative to the agent. Figure 1 shows a virtual environment in Simbrain called odor world.

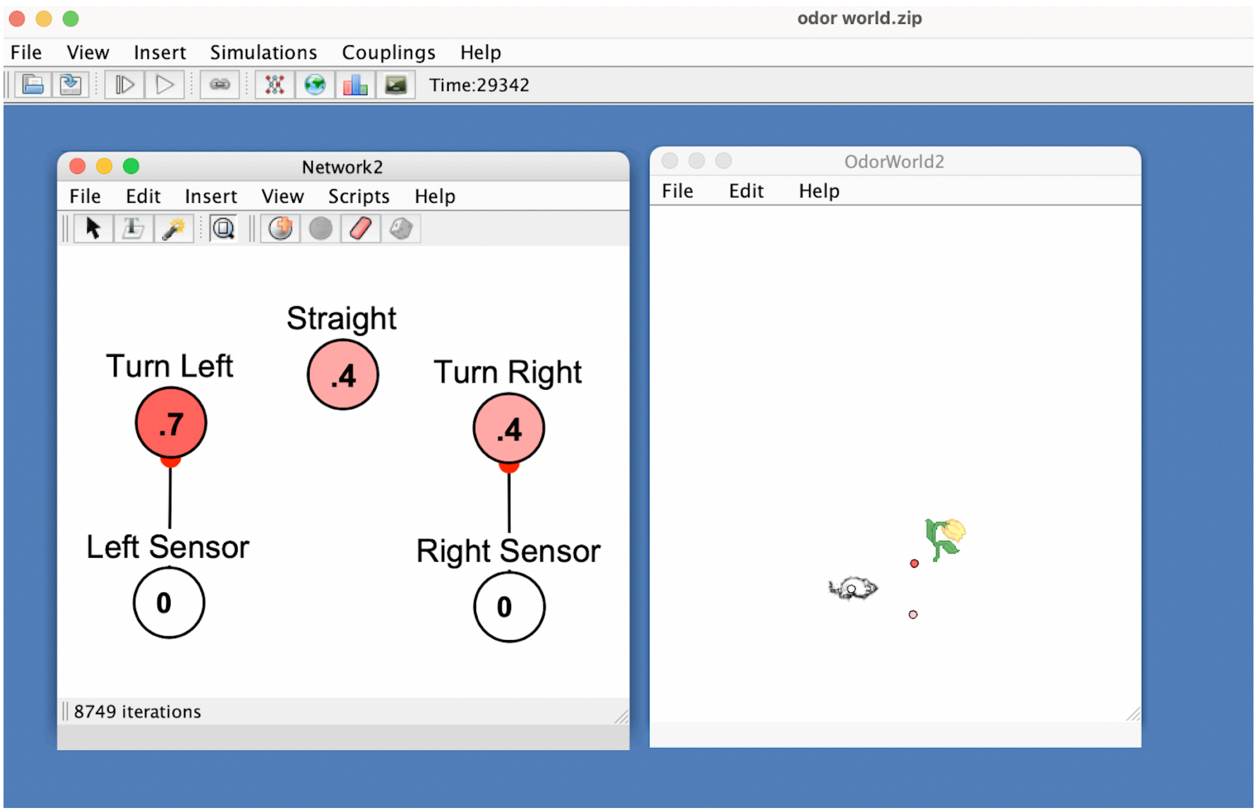

Figure 1: This is an example of Simbrains' workspace. Within the workspace is a preview of a virtual environment simulation called odor world.

There are many different simulations a user can make from scratch on Simbrain, but there are also preprogrammed simulations. These preprogrammed simulations can help first-time users understand the functions of the network.

Internship: Classroom Experience with Simbrain

In the summer of 2022, I was a mentor for the Frontiers of Science program at the University of Northern Colorado. I taught high school students an introduction to neural networks while implementing Simbrain. These advanced students were familiar with the anatomy of neurons and their fundamental role. With this in mind, they quickly grasped the concepts of artificial neural networks and their connection to biological networks. Of the students in my group, two incoming juniors were both 15 years old, and one incoming senior was 16 years old. One design goal of Simbrain is to be able to teach younger students. Throughout the summer, there was a variety of topics covered, but the main topics the students learned were behaviorism, history of neural networks, activation functions along with linear algebra, supervised learning, and unsupervised learning.

Behaviorism

Before diving into neural networks, students learned about the evolution of behaviorism, how behaviorism began, and how behaviorism influenced others and their work. Some of the precursors of behaviorism that were taught were psychologists Ivan Pavlov and Edward Thorndike. Students learned about classical conditioning and how unconscious learning happens when an unconditioned stimulus is paired with a neutral stimulus. With enough association, the neutral stimulus becomes the conditioned stimulus that triggers a conditioned response (Pavlov 1927). They also learned about operant conditions and how specific consequences, either positive or negative, are associated with voluntary behavior (Thorndike 1905). Students also learned about John B. Watson, who aimed to move away from introspection (Watson 1913). In addition, later psychologists such as B. F. Skinner, Clark Hull, Daniel Berlyne, and Edward Tolman made further contributions to behaviorism. Students used Simbrain's preprogrammed simulation of 'Classical Conditioning' to visualize their understanding of concepts of behaviorism. Additionally, they also made an operant conditioning simulator. In this simulation, they were able to train an agent to behave a certain way in the presence of certain stimuli. For example, students could train a mouse to spin in the presence of a flower and wiggle otherwise.

History of Neural Networks

The introduction of neural networks started with a brief overview of their history. This went over advancements from the beginning of the 20th century to the current stages of research. Psychologists like Clark Hull created diagrams that made suggestions for neural network connections. Other influential contributions were threshold units made by Warren McCulloch and Walter Pitts (McCulloch & Pitts 1943). The history portion also went over the cognitive revolution in the 1950s, the "Dark Ages of AI" in the '70s, and the Parallel Distributed Process (PDP) (Rumelhart et al 1987) times around the '80s. Students were also provided with different examples of neural network research. These research topics went over using neural networks to solve a real-world task, modeling biological networks, and connectionism or studying mental phenomena. With this background, the students made their first neural network simulation of an Interactive Activation and Competition (IAC) Network to simulate a connectionist model. Using Simbrain, they were able to represent how the information within semantic memory can flow throughout different neurons through the spread of activation. One student made an IAC network to simulate the properties of cars. Their model had four pools of nodes that showed the understanding of cars. They had one pool of specific cars, an instance pool, a pool of nodes that described their cylinder type, and a pool of different countries that cars are manufactured from.

Neuroscience

Neuroscience was another topic the students were taught. I went over how the cerebral cortex was divided into four lobes: frontal, parietal, occipital, and temporal. Each lobe specializes in a specific set of subprocesses. For instance, the frontal lobe is engaged during high-level processing such as decision making. However, it should be noted that inferior frontal gyrus—also known as Broca's area—is responsible for speech planning and production.

In addition to basic neuroanatomy, the students learned about different neuroscience techniques and reviewed sample studies in which those different techniques were employed. The first technique reviewed was electroencephalography (EEG), in which a cap containing a set of electrodes records electrical activity in different brain areas. The next technique reviewed was functional magnetic resonance imaging (fMRI), which measures changes in oxygenated and deoxygenated hemoglobin in different regions of the brain over time. Lastly, transcranial magnetic stimulation (TMS) was discussed. In this technique, a coil is placed on a subject's scalp to stimulate a magnetic pulse, which can interfere with or enhance neural processing in localized cortex. One outcome of TMS is depressed or heightened motor evoked potentials in the relevant limb controlled by the targeted cortex. There was no Simbrain lesson for this topic.

Activation Function

Additional topics were covered, such as activation functions. Students were taught that neural nodes are areas of concentrated activation. When one node is connected to another node, their collective weights reflect their additive strength. Activation functions are used to determine a node's activation. To calculate this, a node will compute its weighted inputs and add a bias. There are three different activation functions: threshold, linear, and sigmoidal.

Simbrain simulation for this topic was based on Braitenberg vehicles, which are agent-like vehicles where simple internal structures produce complex behaviors (Braitenberg 1986). Students had to create a simulation with an agent pursuing a cheese of their choice while avoiding another stimulus. They students used the three types of activation function to control the forward movement of the Braitenberg vehicles.

Unsupervised and Supervised Learning

The bulk of the summer focused on supervised and unsupervised learning. Unsupervised learning was thought of as learning without a teacher, being unsupervised. This type of learning reflects the changing of weights by only giving a network just inputs. These types of networks are given input data and are meant to learn from those inputs to produce some output. There are many variations of unsupervised learning, such as Hebb's Rule, Oja's Rule, and competitive learning. Students were taught a form of competitive learning called self-organizing maps (SOM), which are helpful in representing high-dimensional data structures in a two-dimensional array. Within a SOM network, clusters are created, representing input data distribution. Unlike unsupervised learning, supervised learning relies on weights changing based on how one wants the network to behave. For instance, supervised learning networks are given input data along with target data which are desired outputs one wants the network to produce. Backpropagation is best known for supervised learning (Rumelhart, Hinton, & Williams 1986). Within backpropagation, there are hidden layers that allow a transformation of inputs into different outputs. One way to transform inputs is through a classification task here, a model can classify inputs into a specific category. They used these topics to create their final project.

Final Project

One of the outcomes the students were required to produce by the end of their program was a final project. The students used what they learned from the course, primarily focusing on unsupervised and supervised learning to create their final project. They created a Self-Organizing Map (SOM) (Kohonen 1990) to represent unsupervised learning and a backpropagation network to reflect supervised learning. Below is an example of the simulation a student created. To do this project, students picked a topic they were interested in and needed different prototypes that could be categorized in input space. For example, a student was very interested in the effectiveness of cancer treatments for the five most common cancers: skin, lung, prostate, breast, and colorectal. These five cancer types were their prototypes in a five-dimension input space of different cancer treatments, including surgery, chemotherapy, immunotherapy, targeted therapy, and radiation therapy. The student found data on this information and was able to create an average rating of the effectiveness of each treatment per cancer type. Given this information, they used Google Collab to use those averages as centers and generated

1000 data points which served as input data that was added to Simbrain for both their SOM and backpropagation networks. Additionally, they also created target data using those same centers and data points to be used for their backpropagation network.

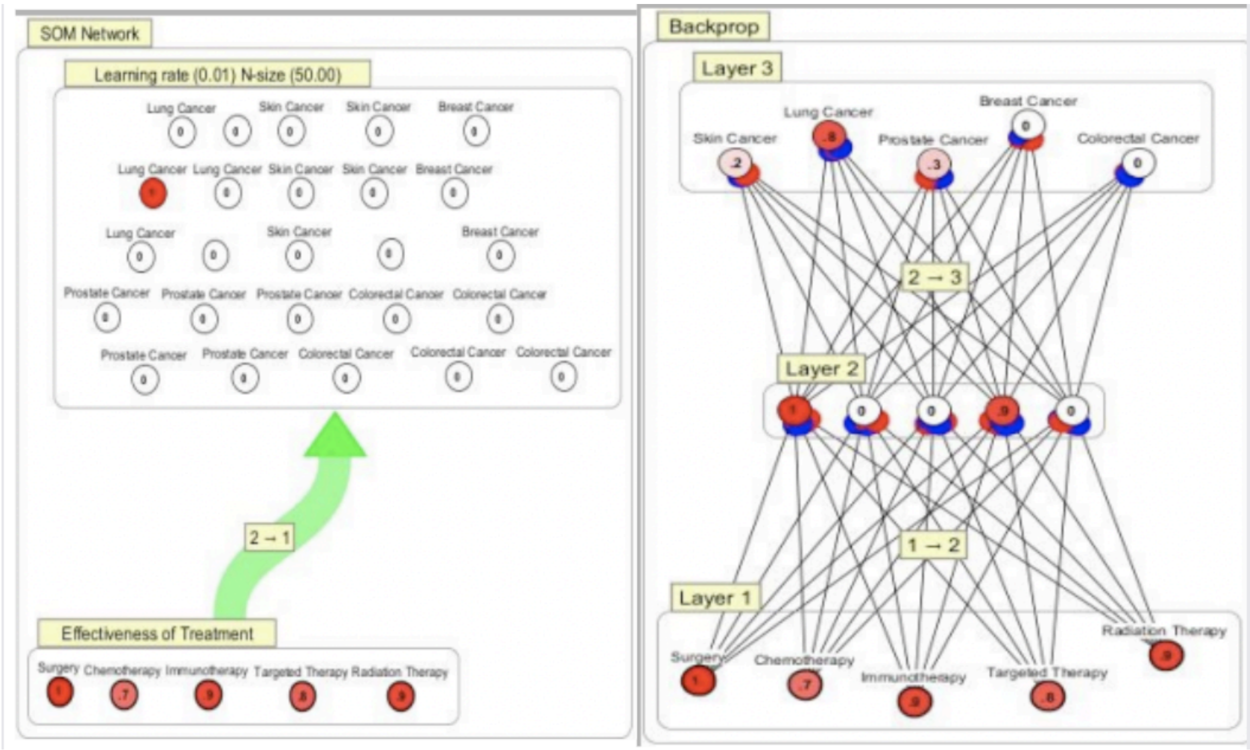

Figure 2: A screenshot of a student's SOM and the Backpropagation network they created for their final project.

Figure 2 shows the SOM network on the left and a backpropagation network. The SOM network has two layers; the first layer is the input layer which has 5 different treatments and the second layer is the SOM nodes. Within the SOM network, there are a variety of clusters of the other cancer types. For example, there is a cluster of prostate cancer in the bottom left-hand corner. The network on the right is their three-layer backpropagation network. The first layer is the inputs of different cancer treatments, the second layer is the hidden layer with 5 nodes, and lastly the output layer. In the image, it is apparent that based on the activation of the inputs, the backpropagation network is classifying lung cancer. All five treatments seem to have high effectiveness for lung cancer, but surgery is the most effective. Other students did similar final projects and used neural networks to simulate a classification task in their topics of interest. One student was very passionate about the Supreme Court. Their final project used average voting pattern data from 1994 to 2005 to determine if justices voted more liberal or conservative based on 6 different categories.

Students' Suggestions for Simbrain

During the students introduction week, they were introduced to an overview of user experience. They were briefed on the topics covered in the background section (pages 1-4). This was one of the first lectures introduced so they could keep those user experience features in mind as they used Simbrain. After six weeks of new knowledge of neural networks and using Simbrain, the students were given a survey to assess their thoughts and opinions on Simbrain functionality. There were general questions on Simbrain's usability, such as how likely someone is to continue using Simbrain, what specific features in Simbrain were and weren't helpful, and rating certain aspects like shortcuts and visual representations. There were three students surveyed.

Students were asked to describe Simbrain in a few words, and one student said it was "a user interface for intuitively creating neural networks," while another student said Simbrain is "a visualization tool," When rating Simbrain's user-friendliness, a student stated that more basic concepts are more accessible to grasp, such as connecting nodes and changing activations within in. What becomes more challenging for users is deciding what neuron type to use. This led to students rating how much background of neural networks is needed to use Simbrain. One student said little to no knowledge of neural networks would be helpful, while another student suggested a heavier background. One student stated, "One easy aspect of Simbrain is its pre-made algorithm.” This student enjoyed Simbrain's customizability by adding the number of nodes, connections, and files that can be added in a backprop network. Another student agreed that the pre-loaded simulations offered intuitive learning within that topic. This student stated that the pre-made simulations helped with learning the simulation functionality.

When asked what was most frustrating about Simbrain, the most common response was the lack of an undo button or function. Students reported having to start specific networks over due to the lack of an undo button. Since these students were first-time users of Simbrain, they were asked how well they understood Simbrain. One student reported, “Simbrain is intuitive due to its visual interface and many beginner tutorials found on the Simbrain website and YouTube channel. It only becomes unintuitive the further you go into the program because there are fewer resources out there to teach you the functions of different neural network components inside of Simbrain.” This student emphasized the need for some background to truly understand what's happening in Simbrain.

At the end of the survey, students were asked what changes could be made to further improve Simbrain. One student suggested creating more comprehensive tutorials for the more advanced components of Simbrain. Additionally, students suggested making those resources easy to find and possibly creating a drop-down menu within Simbrain that leads to those resources. Another student suggested an undo button but stated it would make the interaction more manageable, but not having it didn't take away from the primary purposes of Simbrain. Students were also asked to rate Simbrain on a scale of ten. Two students rated Simbrain a 9/10 on usability. One student said, “It is a widely accessible tool which enables people who might not have an extensive background in computer science or cognitive science to work with and create neural networks in a non-trivial manner.” Another student stated, “its graphical user interface lets users create and tamper with neural networks with ease.” The last student rated Simbrain an 8/10 and explained that small details such as an undo button could make a difference for first-time users. Given that this program was only 6 weeks long and students could only get a brief overview of neural networks, it seems that they appreciated the versatility of Simbrain and could make the connection between concepts learned and visually represent them.

UX Analysis of Simbrain Classifier Component

The current version available for Simbrain is 3.05. Researchers are constantly adding new features and simulations to improve and further advancements in Simbrain. The Smile classifier simulation is a current beta simulation that has not been published.

The Smile classifier is made by first creating a neuron collection. This is done by adding in the number of neurons needed and selecting them. After choosing the neurons, one can use either the key command, shift g, or click edit and select “Add neurons to the collection.” After the collection has been made, the Smile classifier feature can be added. This is done by right-clicking in the workspace and selecting “Add Smile Classifier” from the drop menu. To connect them, one must start by clicking the collection of neurons first, pressing key command 1, then selecting the Smile classifier, and pressing key command 2. After the connection is made, an arrow with a diagonal matrix appears. A diagonal matrix is a square matrix in which the entries off the diagonal are 0. In this case, it was a 4x4 identity matrix (a diagonal matrix whose diagonal entries are 1) which essentially passes activation through unchanged. The primary purpose of a diagonal matrix is to pass activation from neuron collection directly to the input section of the Smile classifier. An example of this is shown in Figure 3. Figure 3 shows the simulation before training. Within the image, it is apparent that the activation from the neuron collection directly passes through the matrix.

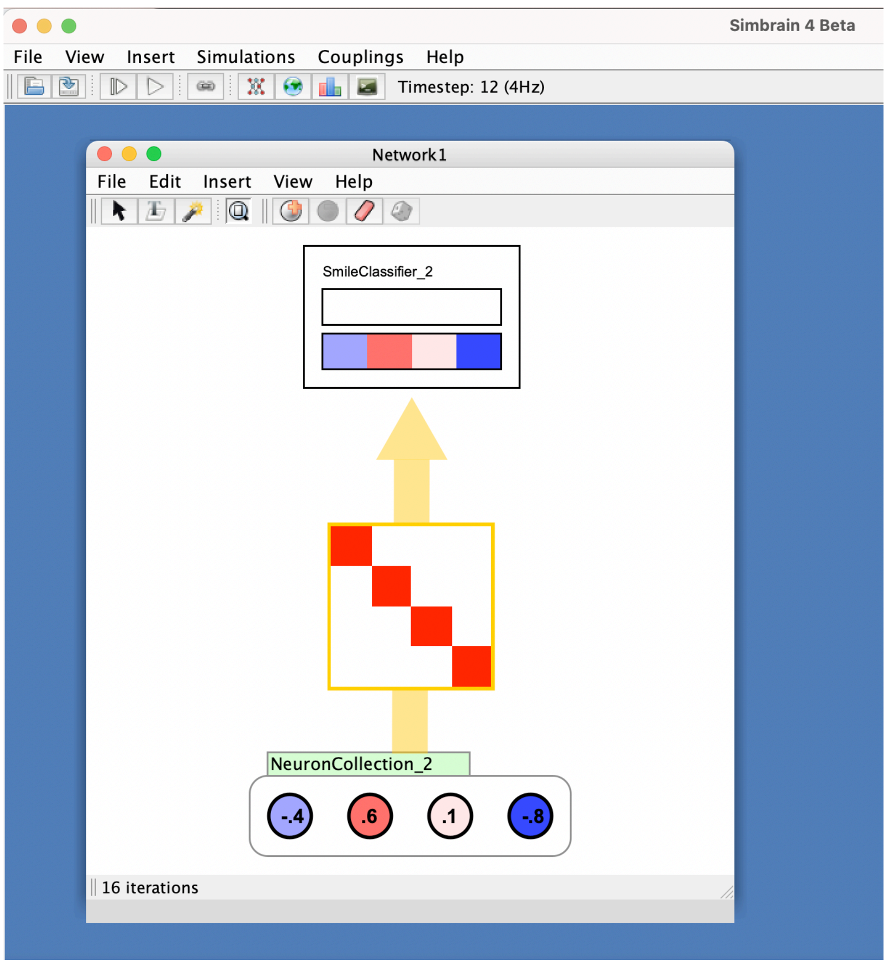

Figure 3: Image of the Smile classifier simulation before training. Visually showing activation in neuron collections is directly passed through the diagonal matrix.

There are two portions within the Smile classifier that represent different things. The bottom bar is an input space that reproduces the weights from the neuron collection, and the top bar reflects the group the trained classifier belongs to. The classification of the top portion of the Smile classifier represents the binary classification of a neuron being on or off. The Smile classifier must be trained first for the overall neuron to classify as on or off. To train, one must randomize the inputs and iterate the training button, which trains almost immediately. If the neuron on the right turns red, the classification for the set is (0,1) 'on,' and if it turns red on the left-hand side, the classification is (-1,0) or 'off'.

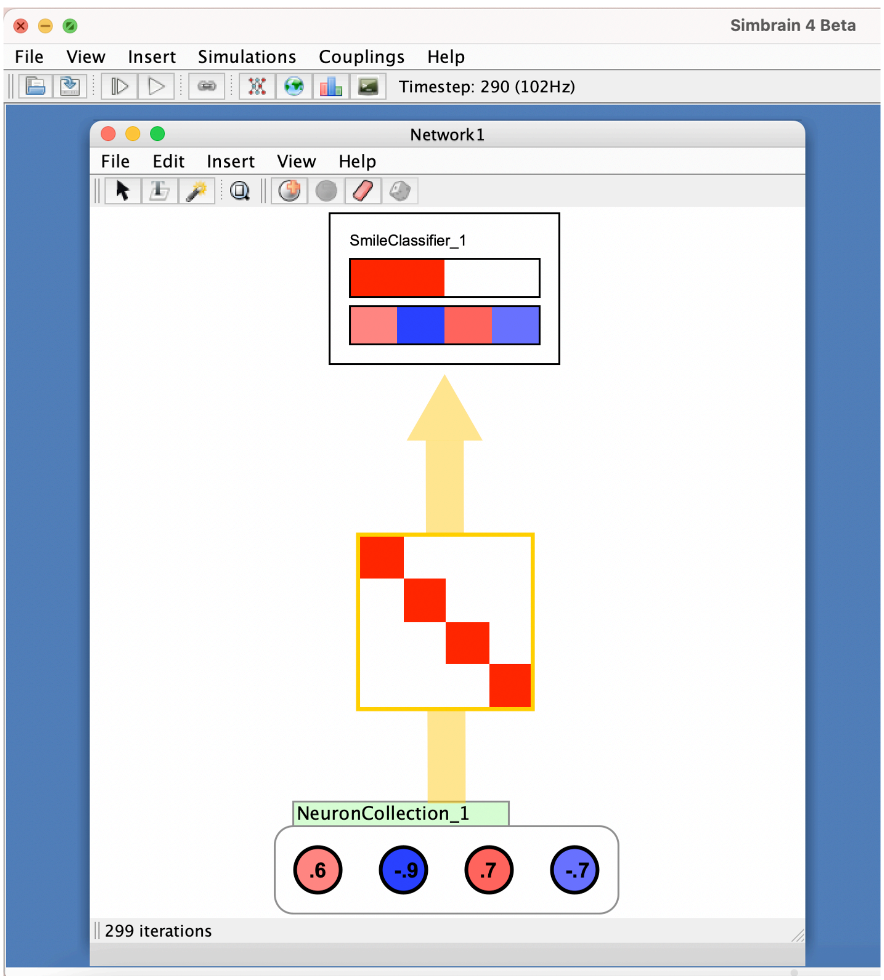

Figure 4: This image reflects the Smile classifier after training, and based on input activation, the Smile classifier is classifying the neuron as off.

Since the Smile classifier is in the developing stages, user experience features can be implemented to promote intuitive learning. These recommendations will be made in the next section below.

User Experience Improvements in Smile Classifier

While assessing the Smile classifier in Figure 3, it is apparent that the flow or direction of the simulation starts with the neuron collection, then to the diagonal matrix, and settles in the actual classifier. In Figure 3, the classifier hasn't been trained; therefore, there is no meaningful activation in the top bar of the classifier. In Figure 4, there is activation on the right side, signifying that based on the neuron collection, the neuron is classified as off, or (-1,0). Without any prior knowledge, a user can potentially make the association between the activation of each node in the neuron collections directly being passed through. The significance of that may not be entirely clear, as well as what the two bars inside the Smile classifier actually mean. A few user interface changes can be made to make the Smile classifier easier to interpret.

First, to make the overall experience more accessible, adding some labels to different bars inside the Smile classifier can provide clarity. I have made a mockup of what that design could potentially look like below with added labels.

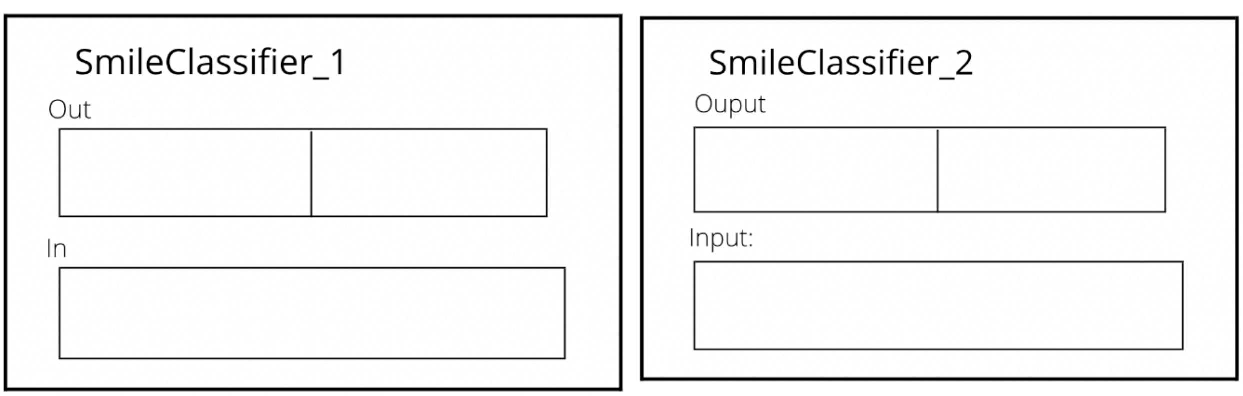

Figure 5: This figure shows two different suggested designs that include labels within the Smile classifier.

Adding labels would help the *visibility* and *status* of the interaction. Even if the Smile classifier hasn't been trained, users wouldn't have to guess or assume, given the neuron collection's activation, what the Smile classifier's bottom bar signifies. Figure 5 shows two different mock-up designs when adding labels. The two images are essentially the same, but the difference is the length of the labels. Having a short label is consistent with a minimal design, while a longer label supports clarity for users.

Figure 5 displays these two options and the tradeoffs between them.

Secondly, shapes within the Smile classifier were considered. Within Simbrain, a node is represented by a circle. Inside, the node holds a numeric value associated with that node's activation. An example of this is neuron collection. The neuron collection in Figure 4 has nodes that all have different activations. Those activations are passed through the diagonal matrix and represented in the input bar.

One can propose that since the input layer in the Smile classifier represents the nodes in the neuron collection, they should also be represented as circles in the Smile classifier to be consistent with other features of Simbrain.

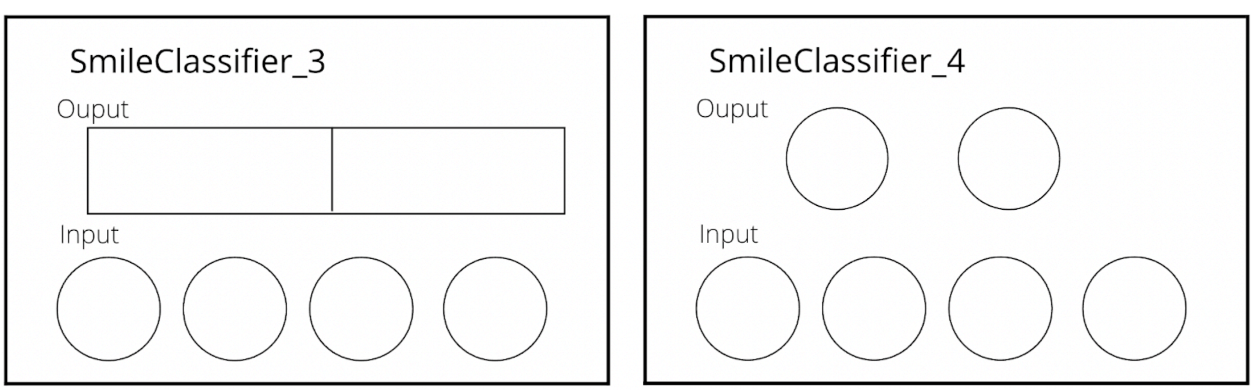

Figure 6: Two possible changes that can be made, which represent activation as circles, which is consistent with other parts of Simbrain.

Figure 6 shows two possible changes that can be made. In both images, the input layer contains circles instead of a bar. Since the input represents the nodes from the neuron collection, having circles can paint a clearer picture for the user of how the activation is traveling to the input layer. Having circles as the input for the Smile classifier may be more intuitive for lower-level inputs. In both images, there are only four nodes in the input space. As more nodes are added, they must be made smaller to fit in a constrained space. If there were 15 nodes or more in the input space, having circles might not be as effective. There would be 15 smaller circles which might be difficult for users to interpret. I recommend that if the input space is less than 15 nodes, having circles, but if there are more than 15 nodes in the input space, keeping the original bar.

Both images have different shapes for the output layer on the classifier. The left image keeps the original bar as the output, while the right has circular nodes for the output. The advantage of using a bar for the output is visually separating the output from all other component within the Smile classifier. The classifier will show one node that is on and one that if off by color affordances already instilled in Simbrain. If the output were circular nodes, a user could assume that activation is just being passed through from the inputs. Having the bar demarcates the end result of the classification.
Ideally, the left Smile classifier output might help users differentiate input and output layers and determine the classification.

Third, color, an affordance for activation, was considered. A node with positive activation is shown as red. The stronger the activation, the more saturated the shade of red. Similarly, a node with negative activation is shown as blue, and more negative activations correspond to more saturated blues. One specific decision associated with the color affordance is how to represent outputs using the Smile classifier. Outputs are generally one-hot encodings for class labels, where all but one of the outputs is “off” and one is “on”. However off can be represented in two ways, either using -1 or 0. Using Simbrain’s established color conventions, -1 would be colored blue and 0 white. We consider the case of two class labels, which might correspond to “apple” and “pear”. Suppose an input is classified as “pear”. That could either be shown as (-1,1) or as (0,1). The first case is shown in Figure 7 left**,** the second case in Figure 7 right.

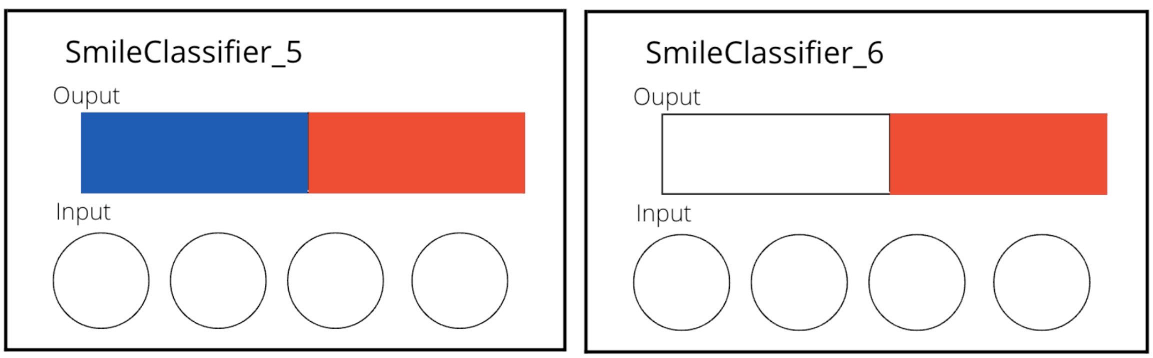

Figure 7: Two ways of representing the right node corresponding to a classification in the output layer. The left case corresponds to (-1,1) and the right to (0.1).

Although users will understand the relationship between the colors and activations, the left image might not be as intuitive when it comes to classification. In the left image, since both nodes are have a highly-saturated color, in a sense both feel like they are lit up, or “on”, and so users might not understand what is being classified. The left image does represent (-1,1) and users might make that connection, but it still might not be clear to users what is being classified. On the other hand, the right image shows only one node with a saturated color, only one node “on”, via (0,1). So, the right image could be more beneficial for users in terms of understanding the classification. Having a clear color association supports the visibility heuristic. Users would be able to quickly grasp which node is on or off. Additionally, having less color in the output could prevent errors and thus support the error prevention heuristic.

To summarize, adding labels can help users identify which is the input and output layers within the Smile classifier. Also, having nodes as the input later for the Smile classifier might help users make a clear connection of the activation from the neuron collection. Having circular nodes would be most beneficial in interpreting lower-level input spaces. If there are more than 15 nodes in the input space, having a bar would help users visually. It would be too much of a cognitive load if users were looking at 15 smaller circular nodes. Lastly, continuing the color affordance throughout Simbrain can reassure users on the association of color scheme and value of activation.

Conclusion

The intention of user experience design is to make easy and straightforward products that can be used by a wide range of users while also providing users with an enjoyable experience. Many different aspects make any design a good design. Minor changes to an interface can vastly improve a user's understanding of the product and their interaction with that product.

Simbrain was used to teach students learned about many different topics, including behaviorism, neuroscience, and even the history of neural networks. They could visually see connections of neural networks and produce a final project through Simbrain. After using Simbrain for five weeks, students of the Frontier of Science summer program were able to provide their own user experience review of Simbrain. Although students from the summer program suggested having some background in neural networks can help with the interaction users have with Simbrain, making those small interface changes can improve the user experience

The user experience review of the Smile classifier suggests minor interface changes that can further help users understand what is happening in the simulation. These changes included adding labels, using circle shape input nodes when there are 15 inputs or less, and preferring minimal color in the output to make classifications easier to interpret.

In this article, I have demonstrated a user experience review on an interactive tool, Simbrain, and based on the present possibilities, I recommend implementing user-friendly design choices. I am eager to see how Simbrain continues to grow and advance.

Bibliography

Braitenberg, V. (1984). *Vehicles, experiments in synthetic psychology*. Cambridge, Mass: MIT Press.

Norman, D. A. (2013). The design of everyday things. MIT Press.

Kohonen, T. (1990) The Self-Organizing Map. Proceedings of the IEEE.

McCulloch, W. & Pitts, W. (1943). A Logical Calculus of Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics*, 5, 127--147.

Moggridge, B. (2007). *Designing interactions*. Cambridge, Mass: MIT Press.

Nielsen, J. (2020). *10 usability heuristics for user interface design*. Nielsen Norman Group. Retrieved August 6, 2022, from https://www.nngroup.com/articles/ten-usability-heuristics/

Pavlov, I. P. (1927). *Conditioned reflexes: an investigation of the physiological activity of the cerebral cortex.* Oxford Univ. Press.

Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1986). Learning Representations by Back-propagating Errors. *Nature*, 323, 533--536. doi: 10.1038/323533a0

Thorndike, E.L. (1905) Elements of psychology. A. G. Seiler, New York. http://dx.doi.org/10.1037/10881-000

Tosi, Z. & Yoshimi, J. (2016). Simbrain 3.0: A flexible, visually-oriented neural network simulator. *Neural Networks*, 83, 1-10.

Watson, J. B. (1913). Psychology as the behaviorist views it. *Psychological Review, 20*(2), 158–177. https://doi.org/10.1037/h0074428