2020

University of North Carolina Wilmington

Master of Science in

Computer Science and Information Systems

Proceedings

https://csbapp.uncw.edu/mscsis

VISUALIZING THE MATURATION OF AN
ARTIFICIAL NEURAL NETWORK

Cody Polera

A Capstone Project Submitted to the
University of North Carolina Wilmington in Partial Fulfillment
of the Requirements for the Degree of
Master of Science

Department of Computer Science
Department of Information Systems and Operations Management

University of North Carolina Wilmington

2020

Approved by

Advisory Committee

Jeffrey Cummings

Elham Ebrahimi

Toni Pence, Chair

Accepted By

Dean, Graduate School

Contents

ABSTRACT

Visualizing the Maturation of an Artificial Neural Network
Polera, Cody, 2019
Capstone Paper, University of North Carolina Wilmington

The objective of this project is to create a visualization application capable of demonstrating in three-dimensional virtual space the abstract learning processes an artificial neural network (ANN) undergoes. This application will consist of three main components: an ANN implementation, a framework to store the states of the network, and the graphical visualization that can ingest the stored states. This application will have an example implementation to leverage this visualization as an educational tool. The example tool is intended for use by post-secondary students learning about ANN to improve their understanding of the general concepts of ANN.

An ANN undergoes numerous changes and adaptations to the various nodes that make up the network via the learning process. As a network learns, aspects such as the weights given to neurons in a layer change along with their values due to a complex series of formulas occurring recursively over time. As the size and scope of the network increases, it becomes more difficult to tune the performance or troubleshoot issues. By visualizing the process over time as a time-lapse, it is proposed that it would be simpler to comprehend the changes. This would be further assisted by allowing the user to explore the network in a virtual space where they can gain an even better perspective of the changes.

It was originally proposed that the project leverage virtual reality to enhance the experience, but this was ultimately removed from the project. Virtual reality capable devices are not readily available which provides a mechanical obstacle to later usability

testing with individuals. This obstacle is further compounded by the limitations that the global Covid-19 pandemic responses have generated. These responses limit the ability to interact with other individuals as well as provide them with devices that are handled by numerous people. The scope of the application was reduced to a desktop application which was accessible on a wide range of machines and did not require atypical devices or hardware. The scope change permitted the experiment to be safely undertaken by removing the risks associated with face-to-face interactions during the pandemic. The removal of XR also removed the risks of contamination on the devices that would be required to be shared to experience the tool in XR.

The efficacy of the tool was evaluated using a small usability study. The procedure of the usability study was having selected users download the application, answer a survey, navigate the example learning tool implementation, and then respond to a final survey. Metrics were then gathered from these survey results which indicated that while the tool was helpful overall, it is unclear if a better implementation would have improved the outcomes. Furthermore, it would be necessary to compare the results between this tool and more traditional learning mediums through further studies before any other conclusions could be drawn.

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

Artificial intelligence (AI) paradigms are in use across numerous industries and fields. Tesla, an automaker, and energy innovator leverages AI to improve their production vehicle features and safety measures. Online based retail provider Amazon utilizes AI to better predict what products website visitors might be interested in. Scientific fields from medical to environmental are all leveraging this toolset to better predict future outcomes and to prepare for what might come. Today, the use case for AI is even now extending to domestic and international policies, both as the subject of and the producer of said legislation. As AI continues to permeate every aspect of society, it will be important that these paradigms are understood beyond simply seeing them as black boxes that ingest and output data.

This project is to develop a set of frameworks that can be utilized to build artificial neural network (ANN) visualization tools. To demonstrate the usefulness of these frameworks, the project will include implementing them into an educational tool for post-secondary students in learning the concepts of an ANN. An ANN is a network of nodes with corresponding values and biases along with connections that all play a role in determining the ultimate state of the network. Through the example implementation, a learner will be able to learn and explore a visualized ANN. Learners will monitor the changes in the nodes and other values through a virtual dashboard. This dashboard will allow them to interact with the network, control playback, and choose what data output should be shown. As an outside observer of the network, the learner will be able to monitor the network through its training process as a predesigned network. If time and resources permit the learner will also be able to visualize a network custom built by them.

The extended reality (XR)[1] industry is growing at an incredibly fast rate with large investments being made by large corporations and financial firms [10]. With this intensive expansion, numerous industries are implementing the technology to improve processes, workflows, and outcomes. CNBC reported that several Fortune 500 companies utilize XR for training and education programs. Walmart for example utilizes XR training within 220 internal academies inside the United States as of October 2018 [12].

Functional XR use cases at a professional level are not viewed as limited to training or gaming only. Numerous companies have begun to branch out their preexisting data visualization tools into the virtual reality space. Looker.com, for example, has released LookVR, a tool to import data from their professional visualization tools and show in a three-dimensional space allowing the user to conceptually understand better their data [1]. By using XR, individuals can understand more complex concepts via this visualization that is more natural to them. Statically viewing an object on a screen only gives a rudimentary idea of scale that may or may not be accurate and does not allow for much interaction. By transposing the object into the virtual environment, the scale is easier to conceptualize in relation to the individual's own perspective. This transposition also allows the individual to interact with the object through any interfaces that might be provided. Research suggests that virtual reality applications improved outcomes when compared to traditional learning methods [7] as well as a promising platform for comprehension and analysis.

---

[1] For the purposes of this paper, the term XR indicates any extended reality technology including immersive virtual reality, immersive augmented reality, and three-dimensional virtual experiences with computer monitors. Regarding VR, the immersion is assumed to be achieved using a head mounted display (HMD) system typically used in conjunction with a device such as a personal computer, personal smartphone/tablet, or gaming console. Immersive headsets provide the user with a circular view roughly between 60 and 180 degrees of a virtual environment and work in conjunction with inputs that mimic the physical world interactions.

The software industry is increasingly calling for developers with experience in XR and AI. This demand is expected to continue growing year over year [3]. Statista.com estimated that growth in virtual reality usage would top 171 million users by the end of 2018, with 114 million of those being users outside the gaming cohort. By completing this project, I will gain valuable experience in VR technology, implementing ANNs, and Unity that can help me enter the VR industry. This project will also serve to improve knowledge of Java while also adding C#, JavaScript, and Python experience to my skill set. By necessity, this project will also include experience with git (GitHub), build management systems (Gradle), data transfers between systems, and industry research. This project is also anticipated to help solidify the concepts of data visualization.

CHAPTER 2: BACKGROUND RESEARCH

2.1 Motivation

Neural networks are part of the artificial intelligence (AI) paradigm which is being pushed to the forefront of industries as a tool to solve complex problems and to pursue better outcomes. Neural networks are utilized across industries and sectors with subject matter varying from bankruptcy predictions [14] to polymerization processes [15]. With AI's usefulness spanning industries and professions beyond computer science, it is evident that a conceptual understanding of AI would be appropriate for individuals within those cohorts. It would also be beneficial to have the ability to better demonstrate visually the underlying processes.

ANN visually lend themselves well to this project because of the layered structure and constantly changing values. It is feasible that with the rendered visualization of the network, observations can be made that would not be possible otherwise. The observations that can be made of an ANN relate to how the combination of network inputs and the network parameters affect the output characteristics. These metrics guide the developer in adjusting parameters such as the learning rate or how long to allow the network to train. Further observation would require developing frameworks to store specific data from within the network itself over time, evaluate differences and interpret results. Developing these targeted metrics could be very time consuming and possibly fruitless. This tool is intended to innovatively permit users to visually observe the network holistically and gain better insight into the mechanisms operating within the network. The example learning tool implemented as part of this project is expected to leverage that insight into positive outcomes for individuals learning about ANN.

2.2 XR as a Learning Tool

As XR becomes more widely accepted across industries and education facilities, it is important to identify what strengths and weaknesses this technology provides. The pursuit of this knowledge has provided an environment for a field of study known as design-science-research [4]. Eileen O'Connor and Jelia Domingo focused their research and suggestions on utilizing open-source environments and established designs for general education applications while still allowing for custom and personalized additions [4]. O'Connor et al. suggest that the design phase of a learning environment should also include considerations for natural interactions as though the learners were in a physical environment [4]. Conversely, the design phase can ignore certain restraints such as location (far away locations) or time (seed growth experiments). Utilizing these concepts, the creator can focus on the strengths of XR and improve the learning outcomes of students. This project also takes into consideration that an entirely realistic environment or high-fidelity feedback is not necessary or preferable [5]. Fowler posits that when given a learning goal, photo realism might be less important than behavioral realism. By focusing on the learning objectives, considerations for this project were placed on clarity of changes in the network, performance, and usability. The design will avoid superfluous additions and distracting elements.

XR as a learning tool has been debated in various publications with some researchers finding that while presence was improved, learning outcomes were not necessarily improved as well [6]. O'Connor et al. discuss the findings of Martin (2014) as indicative that outcomes may be dependent on learner's prior exposure to learning mediums [4]. This is to say that while a "fun factor" can be distracting on its own merits, it could also be due to the novelty of the technology and therefore may not be as inhibitory for learners that are already familiar with the technology.

In reference to the above paragraph, Makransky [6] suggests that "cutting-edge high-immersion VR can create an increase in processing demands on working memory and a decrease in knowledge acquisition, as compared to conventional media." Makransky also suggested that potential improvements on outcomes could result from reducing intrinsic leads on learners by utilizing a less complex subject in the presence of users with little to no experience with immersive VR. To take advantage of this suggestion, the application will focus as well on simplifying the ANN visually such that a learner is not overwhelmed by nonessential details. XR efficacy as a technology from an environment perspective is dependent on three core concepts: Immersion, interaction, and involvement [13].

2.3 Design Principles

Ensuring that the experience of this application is effective specifically requires special consideration to be given to the ANN visualization itself and how the changes are demonstrated. This application intends to reflect changes in the network utilizing changes to characteristics of 3D objects such as size and color. Further details of this can be found in later sections. It is therefore paramount that colors and objects follow good design principles for communicating data. While standards are still being researched for efficacy, we can look to well understood concepts such as pre-attentive attributes and Gestalt principles. Although this project will be focusing on animation of changes in the ANN, it is effectively replacing charts such as the one in Figure 3 and representing that data with visual characteristics. This project also intends to offer users a new perspective on how the ANN is performing and identify patterns based on the underlying data to improve the network's underlying structure. For these reasons, the concepts listed above are being adhered to for learner retention.

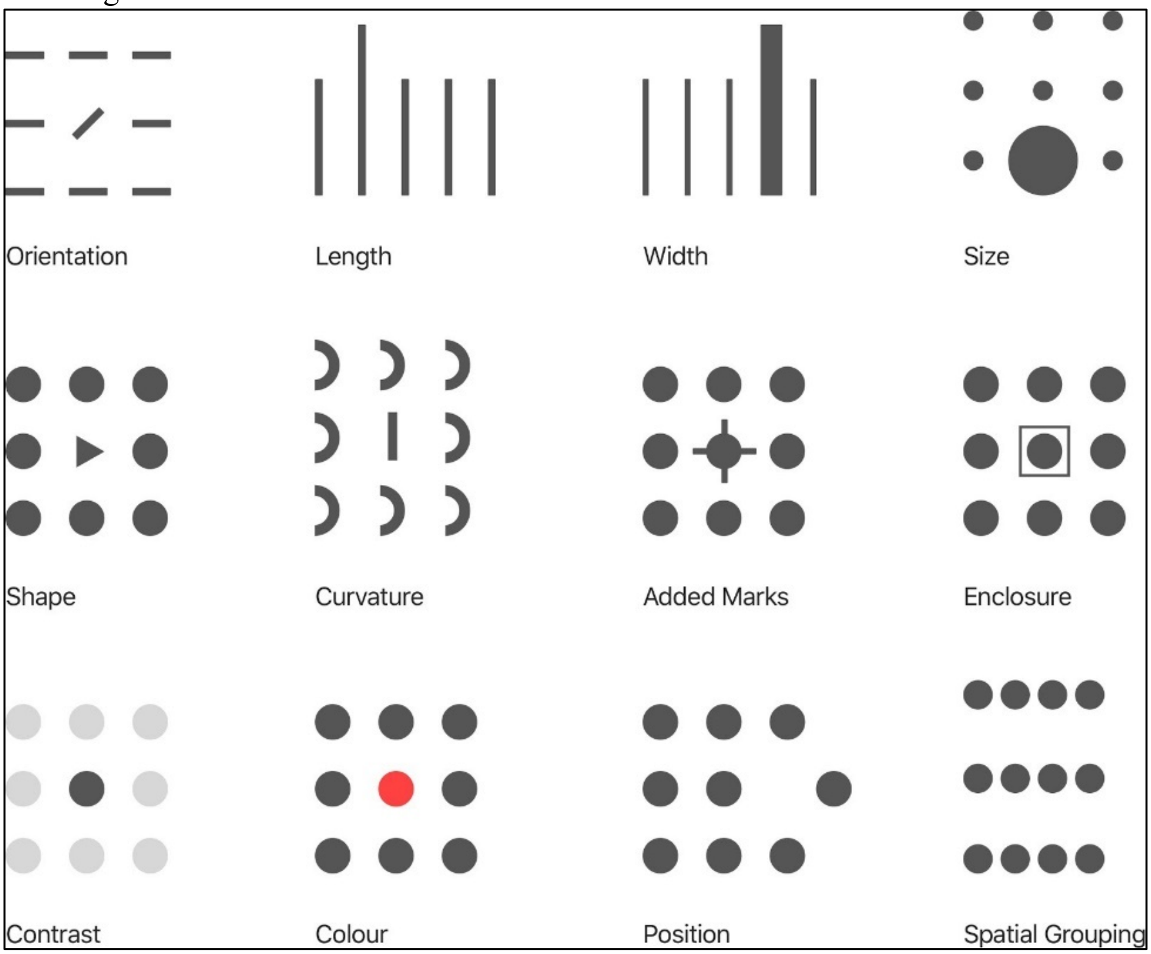

Figure 1 Preattentive attributes [16]

Vision makes up nearly 70% of our sensory input for humans. It is important that visual data be communicated with considerations of several visual design principles [8]. These pre-attentive principles focus on leveraging the pre-attentive processing that occurs based on visual attributes. By encoding the information under these principles, viewers can better process and understand the visualizations occurring. The attributes are broken into form, color, spatial position, and motion [9].

Form refers to aspects such as line width, size, and spatial grouping [8]. Primarily this concerns the ability to quickly process visual inputs and selectively do so based on unique aspects. A group of parallel lines can draw the eye to a specific line through the addition of a box around that line. However, the concept also holds that it is better to add rather than take away. **Color** is considered to refer to hue and intensity. Good design therefore will consider complementary colors for reduced distractions or difficulty processing. In XR, **spatial position** and **motion** are heavily related and refer to stereoscopic depth, convex/concave shapes, direction of motion and flickering.

Few [8] warns against 3D charts and diagrams for the purposes of presenting data on a flat surface in a static manner. Presenting data in a virtual space that can be manipulated should avoid many of the concerns that Few posits. Few states that illusory cues are difficult to present from a 3D surface such as a monitor due to how we perceive shapes and depth given lighting and shadow. This is particularly true of instances where the data presentation is static. Humans process spatial data through numerous inputs as their perspective of the object changes even minutely over a short period of time. While a 2D static graphic prevents this type of processing, a 3D object with interaction enabled allows for processing the space more accurately.

Further consideration given to the design of this application is based on Gestalt principles. These are the result of research extending back to 1912 on how we perceive and process visual inputs [8]. Two of these principles focused on are the *principle of proximity* and the *principle of connection*. The principle of proximity states that by perceiving objects in proximity indicates a grouping of those objects. By placing nodes of specific layers near one another spatially, the viewer will intrinsically process them as groups together. By combining this principle with the principle of similarity through the usage of similar shapes for a given layer node set, we can reduce the preprocessing load on the viewer.

The principle of proximity (Figure 2) also has the characteristic of being able to influence the natural flow of the viewer's perception. By organizing the group in specific ways, the viewer will perceive the inputs in a specific order via direction. This can be in a 2D or 3D plane, however 3D is better suited to a XR environment as discussed earlier.

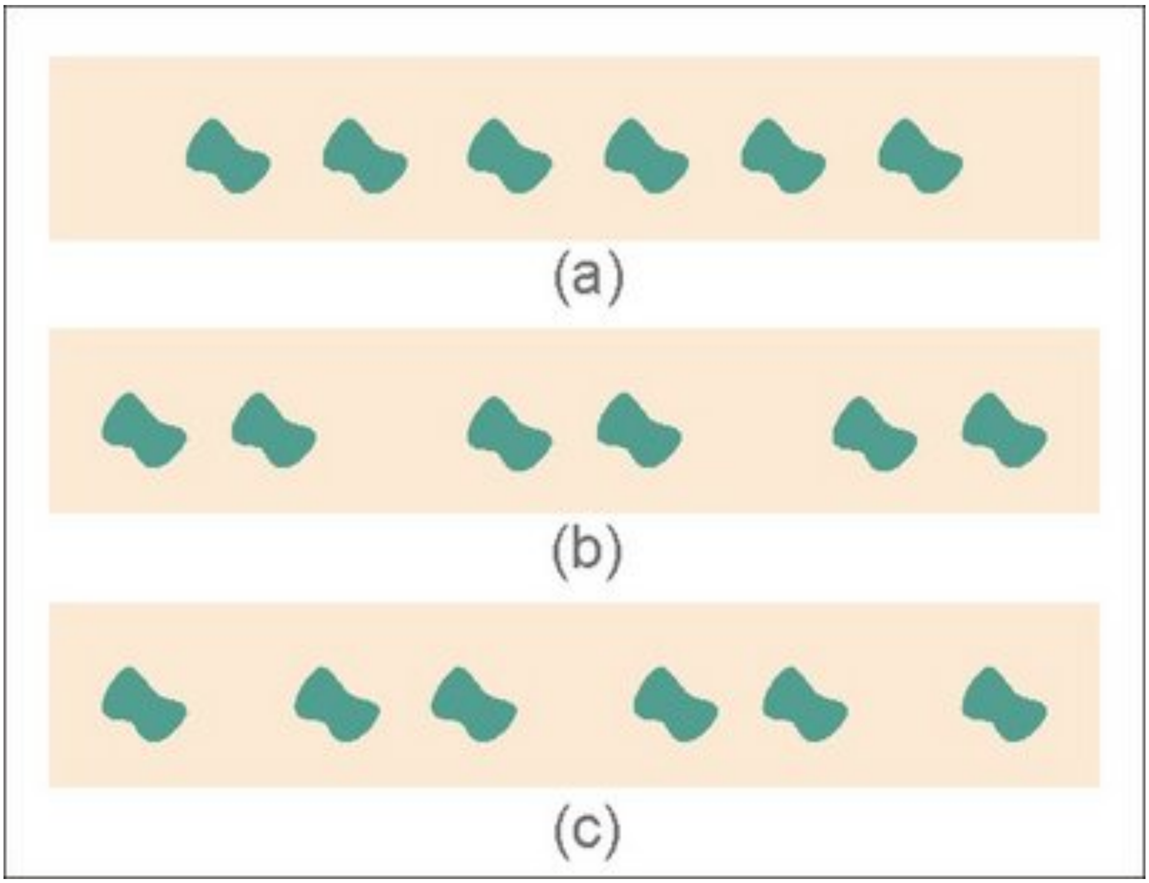

Figure 2: Gestalt Proximity Principle [17]

The principle of connection can be leveraged to help the viewer more easily process relationships between objects that may or may not be easily processed as a group. With a line connection through a set of dissimilar objects, the viewer can understand that there is a relationship between those objects as part of the pre-attentive process which can facilitate comprehending the full relationship more quickly. By including these connections, we can also give shape to a dispersed group of objects to clearly visualize the object being processed.

2.4 Artificial Neural Network

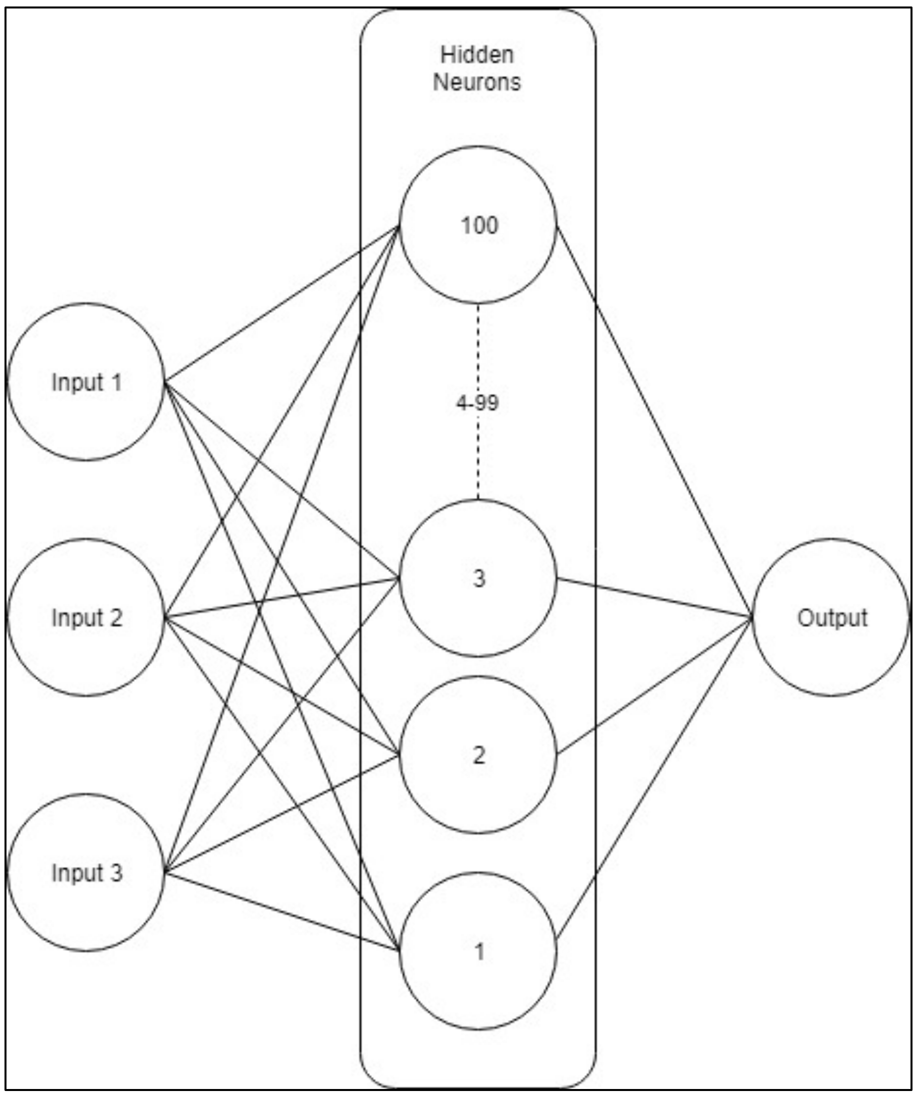

Figure 3: Neural Network as typically diagramed.

Artificial neural networks (ANN) are useful for ingesting related data and predicting a categorization for the inputs. In a *feedforward ANN* with *backpropagation*, a learning data set is given to the network via input nodes. Each hidden layer neuron receives the inputs and passes on calculated values which eventually arrive at the output neuron(s). The neuron generates the output after processing the input through an activation function. This output prediction is compared to an expected result and then iteratively updates the earlier neuron(s). This is done with calculations utilizing the error signal calculated at the output neuron, which is the propagated back through the layers, also known as backpropagation. With each data input, the network optimizes each neuron to recognize and interpret aspects of the input and the impact of that aspect. Each node contains parameters used in these calculations that will be utilized in the project's visual representations.

A depiction of how a neural network can be diagramed is shown in Figure 3. There can be n inputs, j hidden neuron layers, k hidden neurons in a layer, and i output nodes, all of which relate to every node in the adjacent layer. For example, input node 1 is connected directly to hidden neurons 1-100. Each connection also contains a weight to indicate the influence of the node on the upstream node. With each iteration of the feed forward/backpropagation learning process, these connection weights are updated along with node biases and node values. As each set of training data are ingested, the nodes change in their influence over the output given a certain input set.

2.5 ANN Use Cases

ANN are interesting implementations of machine learning algorithms, however like any tool, they are first and foremost tools to be used to solve real world problems. They are particularly useful as a predictive tool across various disciplines. In the late 1990's a team in Turkey used ANN to research and help predict algal blooms in three water reservoirs (19). Algal blooms can have a very real impact on human quality of life by affecting limited water sources through toxins, decreased fishery activity, and water purification processes. The team in Turkey utilized input parameters such as phosphorus levels, dissolved oxygen levels, water temperature, and pH to predict the levels of chlorophyll-a. Chlorophyll-a is present in specific species of algae such as Microcystis which can produce toxins that affect human and animal life.

Another popular use case for ANN is to optimize stock portfolios (20). Stocks lend themselves to this sort of analysis very well due to the well documented historical values and patterns via time-series data. Additionally, there is a large financial incentive to be able to predict future prices and trends. Research has even been conducted to use news stories to predict future values and trends with stock prices (21). Heston and Sinha found using an ANN that positive news had an effect on stock price for a relatively short period of time (1 week) versus negative news that could affect it up to a quarter (3 months).

CHAPTER 3: DEVELOPMENT GOALS AND OVERVIEW

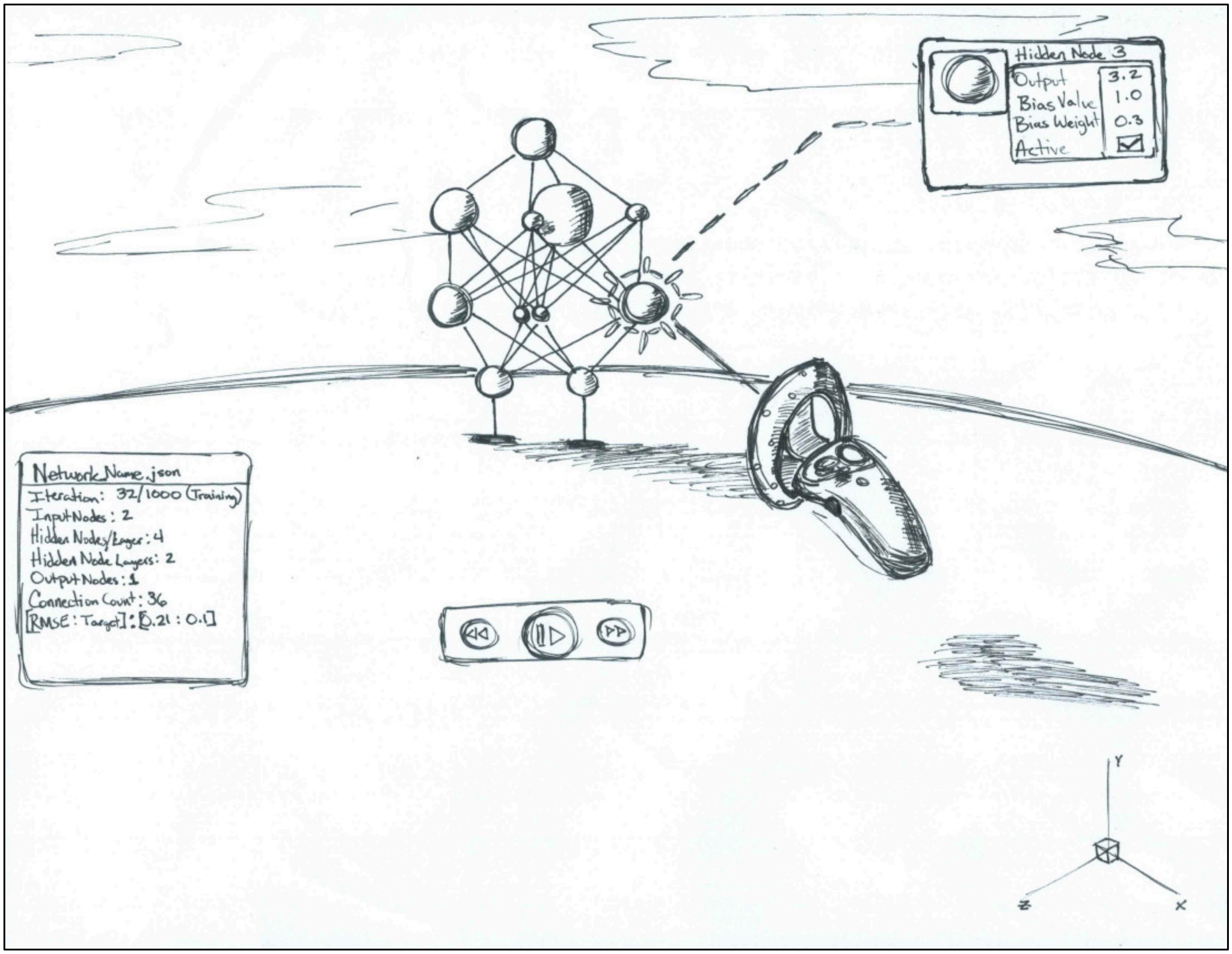

Figure 4: Concept artwork of the application to be built.

3.1 ANN Implementation

The implementation of the ANN utilizes a backpropagation process to update the neurons as part of the learning process. The implementation is broken out into layers for the adaptation of data into usable values for the network. There is a network layer to establish the neural network with supporting structures for connections and nodes. The network can process training data as well as extract a test set randomly from the input data for validation. A layer for output handling will be developed with the focus being a

JSON file for exporting. The most important aspect of this will be to have usable data to be imported and acted upon.

The ANN is currently written in Java 8 (Java) and will persist as such for this project. Performance and readability in Java are sufficient for this project. Java also provides a strong incentive to implement separation of concern design as well as providing a powerful debugging environment. Java also has the benefit of creating applications that can be executed on most systems due to the Java virtual machine (JVM). Additionally, there exist methods of executing Java Archive (JAR) within Unity projects.

.Java 8 is well supported in the development community and many solutions to implementations exist in libraries thereby reducing the scope of work that was completed. Java also has access to several graphical libraries such as Swing to create visuals and user interfaces (UI). Development itself has been completed through IntelliJ Community edition and the code will be hosted on GitHub.

3.2.1 ANN Visualization - Nodes

A node can be thought of as an abstract object that takes in an input and processes it through an activation function to produce some given output. This output is affected by the function itself as well as external factors that are represented by the bias. An easy way to think of this would be to consider a decision to vote for a given politician. There are several factors such as party affiliation, one's own corresponding morals, and who the opponent might be. Each of these factors can be looked at as inputs that predict the choice. If the voter leans heavily towards a particular party, then the politician's party affiliation will more heavily influence the decision. ANN randomize these node bias weights initially and over the training and testing phases will automatically incorporate those biases onto various hidden neurons. If a voter chooses candidates from a specific party 95% of the time, then the network could take that and convert that information into a bias.

In this project, the node bias value and bias weight inputs for a given neuron will be represented as the neuron hue. The hue will range from blue representing a small impact bias and will progress through green and yellow up to red to indicate increasing bias value applying on that neuron. The bias weight will determine the tint and shade of the neuron with a high tint indicating a low weight and conversely a high shade value will indicate a high weight. Each hidden neuron object size will be representing the output value. If a neuron is determined to not be impactful on the output during the training process, it will be pruned from the network. The pruned node will be desaturated, and all characteristics of this node will remain static. The corresponding connections will also be deactivated in the same manner.

Component	***Characteristic***
Node Output	Size
Node Bias Weight	Hue
Connection Weight	Luminance
Output Node Result	Hue (red/green)

Figure 5: Visual Guide for ANN Changes

3.2.2 ANN Visualization - Connection

In addition to the neurons, a key part of the ANN are the node connections. Connections also contain a bias value that affects the output between the neurons. These connections and their bias value being represented by a thin luminescent rod object. This object's luminance was to be indicative of the weight with a high intensity corresponding to a heavy weight while low intensity would indicate a low weight. In the case of a pruned node, the object will be changed to no longer be luminescent and will no longer updated. While the underlying framework was developed and tested, the actual visualization portion of this object was not completed.

3.2.3 ANN Visualization – Overview

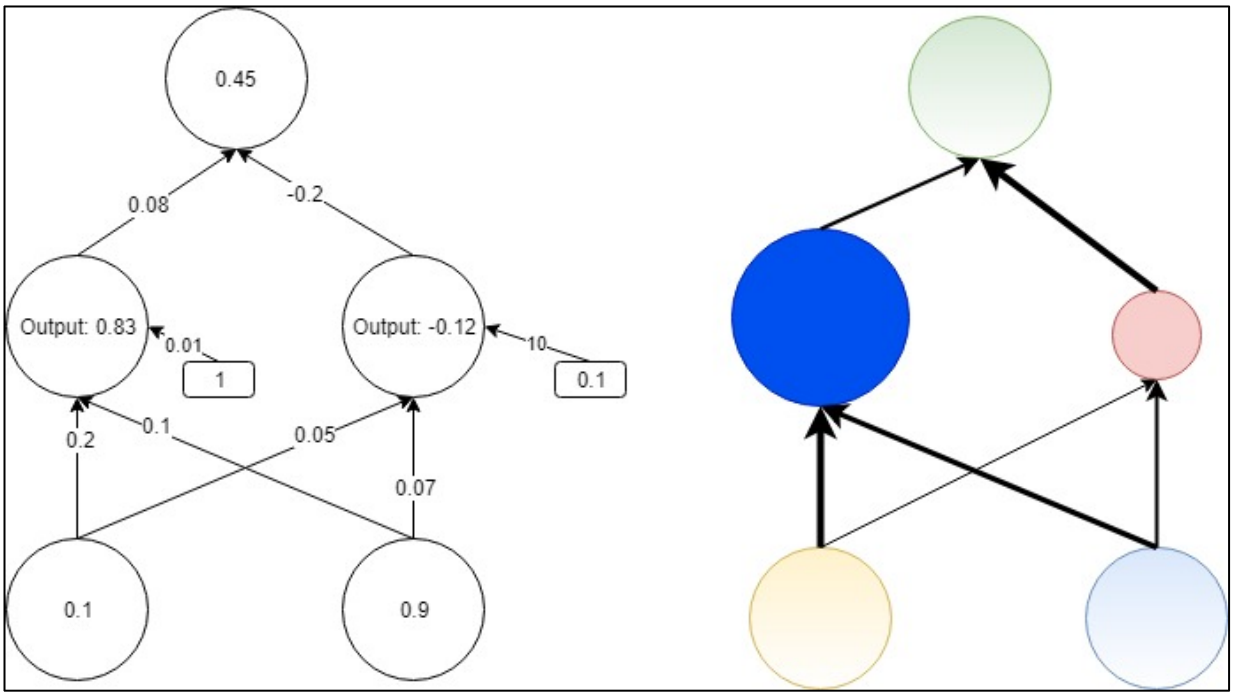

Figure 6: Comparison of visualizing a neural network state. These represent the same states, but in different manners. Given the two inputs(0.1,0.9)/(Yellow, Blue) the output is expected to be 0.45/Green.

In Figure 6 a demonstration of the previously mentioned concepts are shown. This image includes visualizations for the hidden neuron layer as well as the connections between neurons. Given two inputs 0.1(yellow) and 0.9(blue) there is an expected output of 0.45(green). The left hidden node is larger due to its larger output, but the color is blue due to the bias not being as impactful. The shade is on the darker side of the spectrum however to indicate the bias value is large in comparison to other bias values. Conversely, the hidden node on the right is small due to a smaller output value and is light red, indicating a low bias value that is also heavily weighted against the neuron.

Each connection is given a thickness based on the absolute difference of the weight from zero. As noted, this is a single state of the network. The depiction on the left of Figure 6 will show a change in the numbers over time only. The diagram on the right however will depict visual changes using color shifts and size modification. It is expected that the visualization will provide a better conceptual understanding of the changes and be easier to observe the changes when compared to attempting to process raw numbers as they change.

Output neurons will also take on a greener hue if the result is within the expected target parameters, while it will be redder if the value is outside of those parameters for a single output node during the training. If a single node has more than a binary state, further colors could be utilized in addition to the prescribed green and red. However, due to the constraints on time and resources, exploring options beyond binary states will not be undertaken. For the purposes of this project, the output neuron hue will reflect the RMSE with changes to the color occurring in increments.

3.4 Virtual Reality Component

This project aimed to create an application that is usable not only as a desktop application, but also in the Microsoft virtual reality platform. The Unity platform, which is currently one of two platforms supported by Microsoft for their VR devices, was used for the development of the visualization and example implementation of this project.

Unreal Engine is in the process of being supported, but due to time constraints was not considered for this project. The underlying code was developed in Visual Studio using C# as required by working with Microsoft projects. The initial stages of the VR portion were tested in the Microsoft VR emulator environment. Early usable versions of the application were tested with the Samsung Odyssey Plus, a WMR based device.

Later in this paper, the virtual reality component is discussed more in depth. For the purposes of this section, it is necessary to relate that while VR was initially included, its implementation was removed. Many of the proposed features were designed with XR intended as the manner of interacting and viewing the application. The corresponding concept depictions and descriptions reflect this intention. Much of the application as designed was compatible with a traditional desktop application user experience and as such did not require much modification. The main differences stemmed from the input control structure and the way the user would interact with the application. Some proposed features were not completed due in part to this change. These proposed features can be viewed in the appendix for reference.

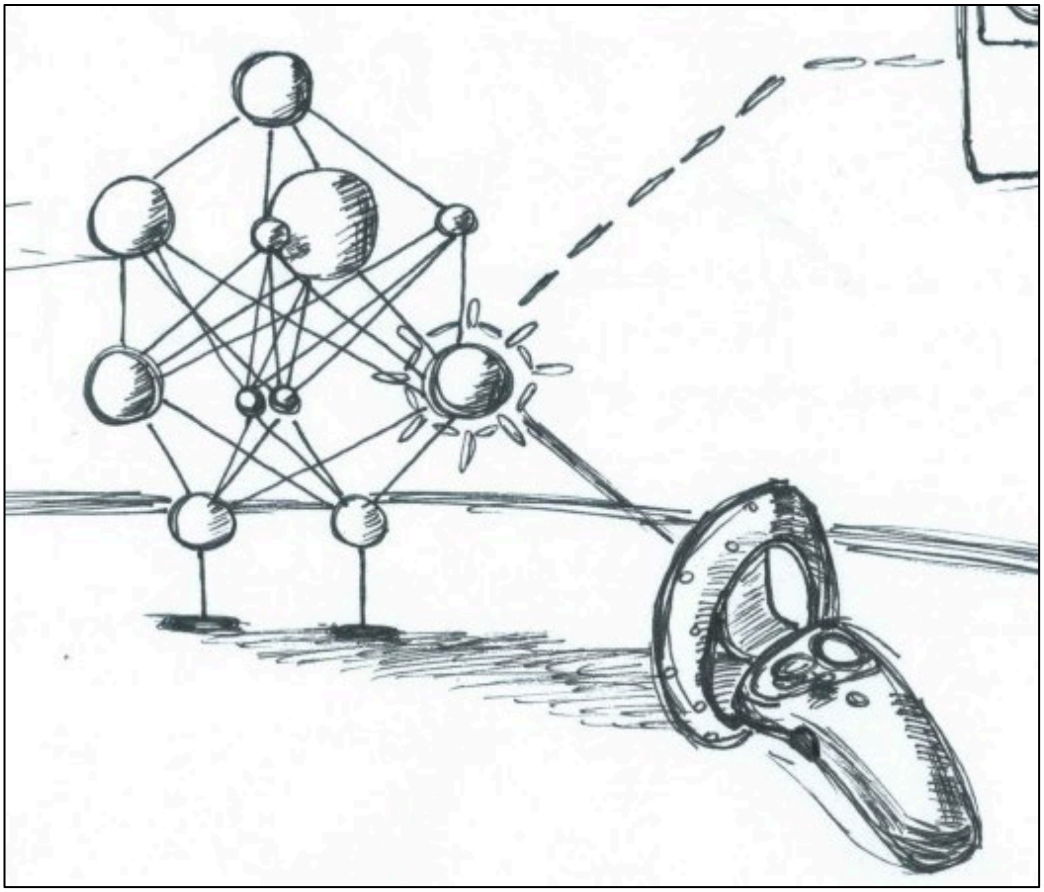

Figure 7: The user will be able to select a network object to learn more about it

3.5.1 User Interactions

Three dimensional applications benefit from presence and immersion which necessitates interactions. As described earlier, there were to be several opportunities for the user to interact with the application. This section will cover the details of the implemented interactions. To see proposed features that were not included, refer to the appendix. Motion controllers as shown in Figure 9 were used to manage these functions and all other user actions when XR mode was supported. The desktop application version uses the traditional mouse and keyboard combination to control the user interactions. A click action is when a user focuses on an object and clicks a button on the controller (VR) or mouse (Desktop). The object of the action is determined by the controller ray-cast (VR) or the center of the screen (Desktop).

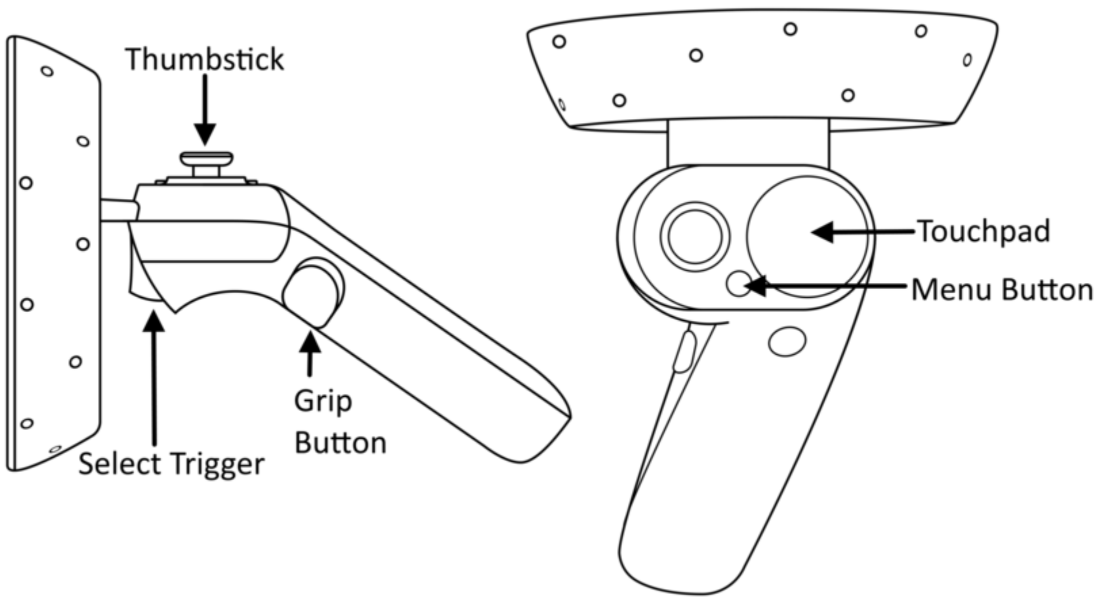

Figure 8: Control layout for Windows Mixed Reality (WMR) controllers (18)

3.5.2 Object Selection:

The user is able to select a network object to learn more about the object or to monitor changes to it (see Figure 8). The actual data output is discussed in a later section. Once an object is selected, the object's data is shown to the user. To facilitate object selection, users have a cursor/crosshair in their view to designate where they are currently pointing. Additionally, the object currently being ray-cast at is outlined with a faint light effect to better differentiate the object from similar ones in the vicinity. Once selected, a similar effect with a brighter shade replaces the lighted effect and permits the user to maintain visual contact more easily during the training process. The user can deselect the object by using a different input command.

3.5.3 Selected Object Data:

When an object is selected, the data corresponding to that object is shown with a GUI implementation. The top of the display area will show the name of the object and its type (e.g., Hidden Node 31). Below this designation will be a list of descriptors of that object. For a hidden neuron, this will consist of the output value, bias value, bias weight, and its active status. A connection will show its weight, input node, and output node along with its own active status.

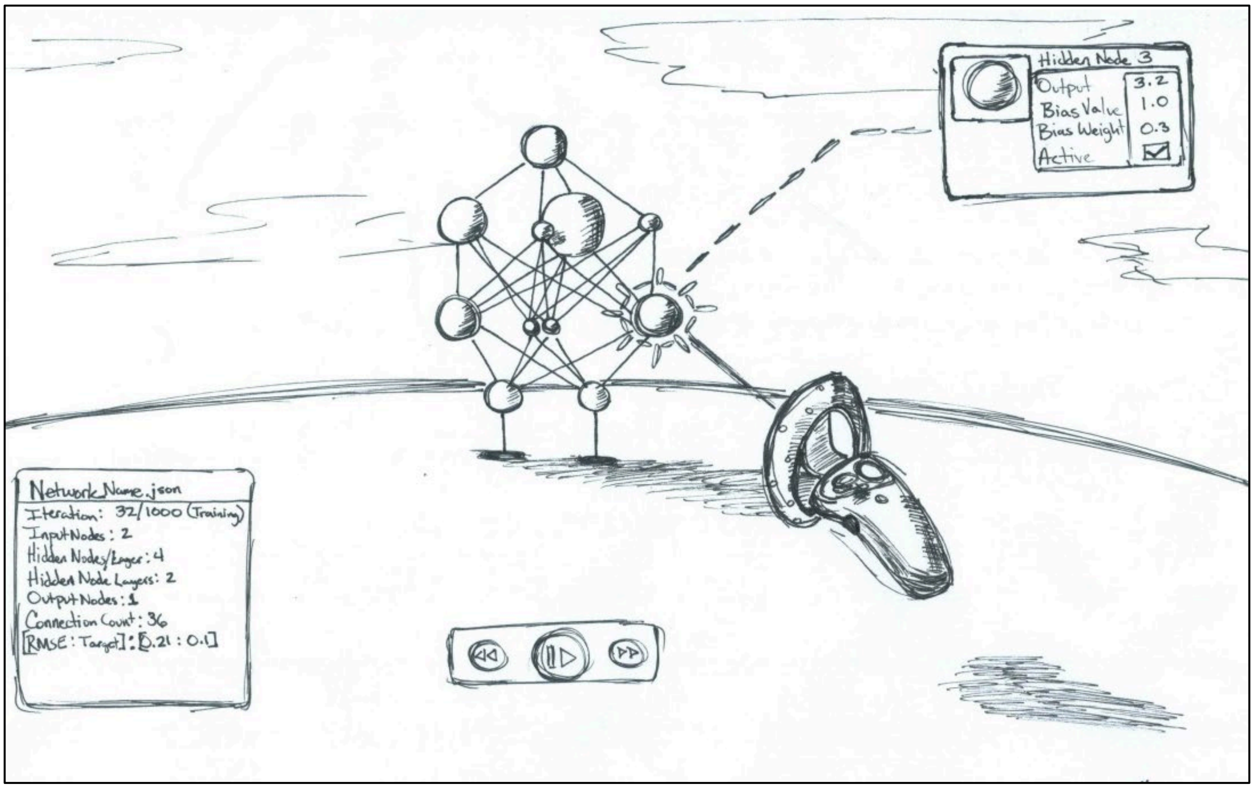

Figure 9: Output data display areas for selected network object (top left) and the network (top right), respectively

3.5.4 Network Information:

There is a display area for the information corresponding to the network itself. This section displays the name of the network, the number of input nodes, the number of output nodes, total node count, total connection count, pruned objects count, the status (testing or training), and the current RMSE value.

3.9.1 Scenarios

The first example scenario that could be given to a user is based on a very simple dataset where there are two inputs into the network and a single expected output. The user will be given control over the playback of the scenario and control over the monitored aspects of the network. The user will not be permitted to modify the number of hidden layers or hidden nodes. The target RMSE will also be locked for the duration of the scenario to 0.1. This dataset was chosen for its simplicity and common usage. It is a variant of an XOR dataset. The scenario will have one hidden layer, two hidden neurons per layer and one output.

Input Values	Expected Output Value
0.1 : 0.1	0.1
0.1 : 0.9	0.9
0.9 : 0.1	0.9
0.9 : 0.9	0.1

Figure 10: Simple XOR scenario

3.9.2 Scenario 1.1

The second version of scenario 1 will contain a second hidden layer and each hidden layer will contain three hidden neurons.

3.9.3 Scenario 2

The second scenario available to the user will be the numerical digit recognition dataset scenario. Each input set will consist of 25 input nodes that represent each pixel of a 5x5 input image.

An example of the input image transformed to bits would be:

0	1	1	1	0
0	1	0	0	0
0	1	1	1	0
0	0	0	1	0
0	1	1	1	0

In this example, you can see the positive bits which are represented with 1's makes the shape of a number 5. As the network is processing the learning set, the user will again have control over the playback and monitored aspects. At the beginning of the scenario, the user will have the ability to identify their desired number of hidden layers as well as the number of hidden nodes to utilize. Layers will be limited to a maximum of 3 hidden layers while hidden nodes will be limited to 100 in increments of 5.

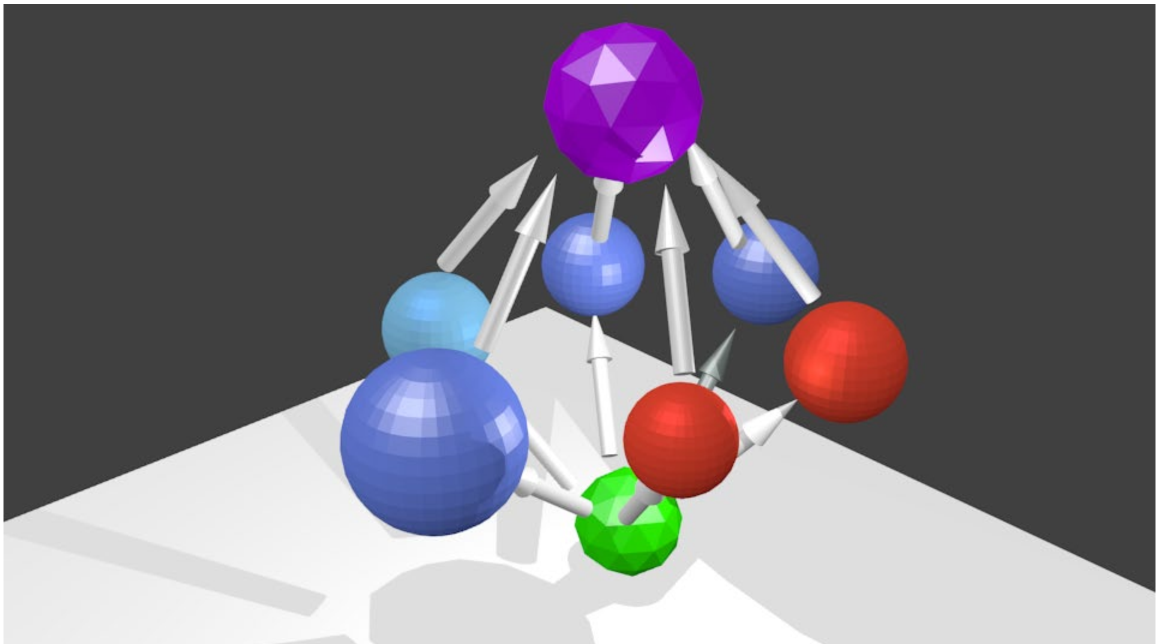

Figure 11: Rudimentary depiction of the visualization of the ANN in 3D. These objects are meant to represent the basic premise of the project.

CHAPTER 4: IMPLEMENTATION

4.1.1 Implementation Introduction

Throughout the process of completing this capstone it was necessary to make several changes and adaptations. These updates along with the reasoning behind each change will be found in the sections below. The first section covers the ANN implementation and what challenges arose from using the previously designed network. The following section details how the data from the network was logged and why specific choices were made to handle the overwhelming amount of data generated. Finally, the last section demonstrates each iteration of the visualization and how it evolved over time based on feedback, new requirements, and other inputs.

4.1.2 Network

The first task to be completed with this project was by necessity, the ANN. The first subtask was to refactor the existing ANN project to be repurposed for this capstone. Much of the prior development on the ANN did not adhere closely to OOP principles such as inheritance or abstraction. For example, previously nodes were stored as values in simple arrays. While this was effective for the purposes of a simple ANN sample network, it made it difficult to maintain. This also made storing data and exporting the network states nearly impossible. By establishing a class structure and abstracting various components, it became possible to track changes to the network and to also store the states during the process.

All nodes in the network are represented by the Node parent class. Each input node is represented by the InputNode object which is a child of the Node class. OutputNeuron followed suite with its own inheritance structure. This structure was necessary to allow for custom code to be developed for each layer, particularly when it came to the data storage of each object.

Another important task as part of enhancing this original code was to develop a process to facilitate acceptance of data for training and testing. The network expects an array of input values (*doubles*) along with an array of values corresponding to the correct output. An example would be (in code) :

```
int[] plusSymbolIntoVals = new int[]{
      0, 0, 1, 0, 0,
      0, 0, 1, 0, 0,
      1, 1, 1, 1, 1,
      0, 0, 1, 0, 0,
      0, 0, 1, 0, 0
};
```

And the expected output for a three-output node network could be (in code):

```
int[]{0, 0, 1}, "PLUS"
```

While the network requires these sorts of inputs, it is not something a typical user would ever enter. The example *plusSymbolIntoVals* shown above is a representation of a plus symbol with 1's for the plus itself and 0's for the whitespace around the character. For the network to be able to accept data that is not already in this array format, this data construct was abstracted out to a parent Symbol object which is a child of NNObj class. By using this structure, when the user inputs some sort of data set to train on, it can easily be parsed out with custom logic built for the different expected file types.

4.1.3 Logging

Once the ANN was successfully implemented, it was necessary to build the framework to log the states that the network would go through. A network state in this case is a snapshot of all the values in the network after the backpropagation process is complete following a single training event. Multiple training events constitute a training run. By storing all the values in an appropriate and ordered manner, it is possible to recreate the entire maturation. To see a sample of the stored data, please refer to Appendix 1.

Several forms of data storage were tested for this project including adding listeners and reporting classes along with custom built text file writers. The complexity of these was evident after little testing. In addition to the highly custom aspect and the time it would involve outputting the data like this, it would also necessitate creating and maintaining a way to import the data on the Unity side. The JSON file format was investigated as an option to transfer the data out of the network. JSON is one of the most common forms of storing data to pass between frameworks currently. It is a very structured format that is highly supported in many different programming languages and frameworks. This makes it an ideal candidate for this project to be able to move data from a Java 8 application over to a Unity/C# project without creating a strict dependency structure.

There are numerous libraries and built-in tools for both Java and C# to manage the data stored in JSON files. In the case of this project on the Java side, the decision was made to use a GSON library to facilitate converting Java beans into strings of data that could be stored as JSON. The initial file structure stored a new object every time as a copy of the network along with all its related data and objects. This ultimately turned out to be problematic on two levels. The first was that many of the classes did not easily convert if at all over to a usable data structure due to how the GSON library handles object data. The second issue was that due to the amount of data and metadata being stored, file sizes were more than 10GB for a small network and more than 30GB for a large network.

Storing the network state is a complex task that involves managing the entire network over time and appropriately storing the data snapshots. Consider that each network contains an array of input nodes, the hidden nodes per hidden layer, and output nodes. If this network uses the example above, we can assume there are 25 input nodes, thus an array of 25 values. With 10 hidden nodes per layer and two layers, we can say we have 20 hidden nodes. Each hidden node has a bias value, a bias weight, and an output value. At this point for the hidden layers, we are storing 60 values.

In this same example, the output is judged on three values, so we have another array of three nodes, each with a bias value, weight, and output. Let's say this is nine values for the output layer. Total values stored are 94 values for a single network state. Remembering that each node is connected to each node of the adjacent layer via a Connection object, and each connection object has a bias weight, this brings the total connections related values to 25 x 10 x 10 x 3 = 750 values. This does not include any other necessary data that might be passed along such as identifiers, other fields, or field names. For a network that undergoes a small training round of 1000 iterations, the output file contains 750,000 values. If we include testing results as well for 100 tests, then we are over 800,000 values stored in the file. It is worth noting as well that this is a small network with little training being run. When viewed in this way, it is obvious that file sizes can be difficult to minimize and can make porting the data difficult.

Resolving these issues required a new approach to how the data was being stored. Firstly, the application would no longer store the raw network along with its objects each time. Certain data does not change between iterations, such as meta data or static network values. Any data such as this that was static over the course of the network lifespan was stored in a separate section of the output file and was only stored once. This reduced the size of the file by nearly a 1GB (Roughly 10%) for a small network due to how much data was stored as well as the field names.

A second aspect was to take the input objects for training and testing and only store them once in a separate section as well. Previously the file was storing a copy of the input object arrays for each iteration. For example, if there were four characters converted to arrays of 25 integers, this would result in 100 values stored plus metadata for each iteration. This was not necessary and by storing these with references to them later, it also reduced scope and complexity of the file.

The third portion of the approach is tied with the first two as the method by which they were accomplished. A custom-built process was created to selectively choose what to save at a given time. More specifically it was a combination of what should be kept as well as what should be excluded from the file. By ensuring that only the changing data was stored, space was not wasted on static values. This process also allowed the application to create the file with these three sections which allowed parsing it later to be much simpler.

4.2.1 Prototype Visualization

As explained earlier in the paper, visual representation can increase comprehension of complex concepts. The third portion of this project focused on providing a framework that could be used to graphically demonstrate how an ANN develops over time. There are two primary options to create this demonstration in three-dimensional space. The first is via a 3D computer graphics software such as Blender or Maya. This software allows the designer to create three dimensional objects along with creating animations. Software like this is used to create renderings with high fidelity graphics and lighting. This option would have allowed producing the demonstration but would have not allowed any kind of interaction with the user, much like any other video. A new network would have to be imported, processed, rendered, post processed, and encoded before being viewed.

The second option is to use a game engine which can render objects in 3D space in real time and allow interactions by the user. In addition to having the ability to import 3D computer graphic objects, it can modify them in real time, execute animations, react to user inputs, and store data regarding the session for future use. Of the two major game engines, it was elected to utilize Unity over Unreal Engine. Initially this decision was due to Microsoft only supporting Unity for their Windows Mixed Reality platform. During the time in which this project was developed, WMR support has been added for Unreal Engine. There were no anticipated benefits from changing game engines and the project persisted with Unity since much work had already been completed there.

4.2.2 Prototype Visualization Iterations

Temp:

1. spheres: https://assetstore.unity.com/packages/vfx/shaders/gem-shader-3
2. Panels and frames: https://assetstore.unity.com/packages/2d/gui/icons/sci-fi-interface-frames-10747
3. Outline tool: https://assetstore.unity.com/packages/tools/particles-effects/quick-outline-115488
4. Building/Lights: https://assetstore.unity.com/packages/3d/environments/3d-free-modular-kit-85732

5. Skybox: https://assetstore.unity.com/packages/2d/textures-materials/sky/skybox-volume-2-nebula-3392

4.2.3 First Iteration Vertical Orientation

The earliest iteration of the graphic visualization was built to utilize the WMR tool kit with the intention that the project would be a strictly VR project. By using the toolkit, controls and camerawork was easily implemented by using the built-in library. Doing so reduced the scope of the project which allowed more time to develop the rest of the project. This decision later introduced challenges that are discussed later in this paper.

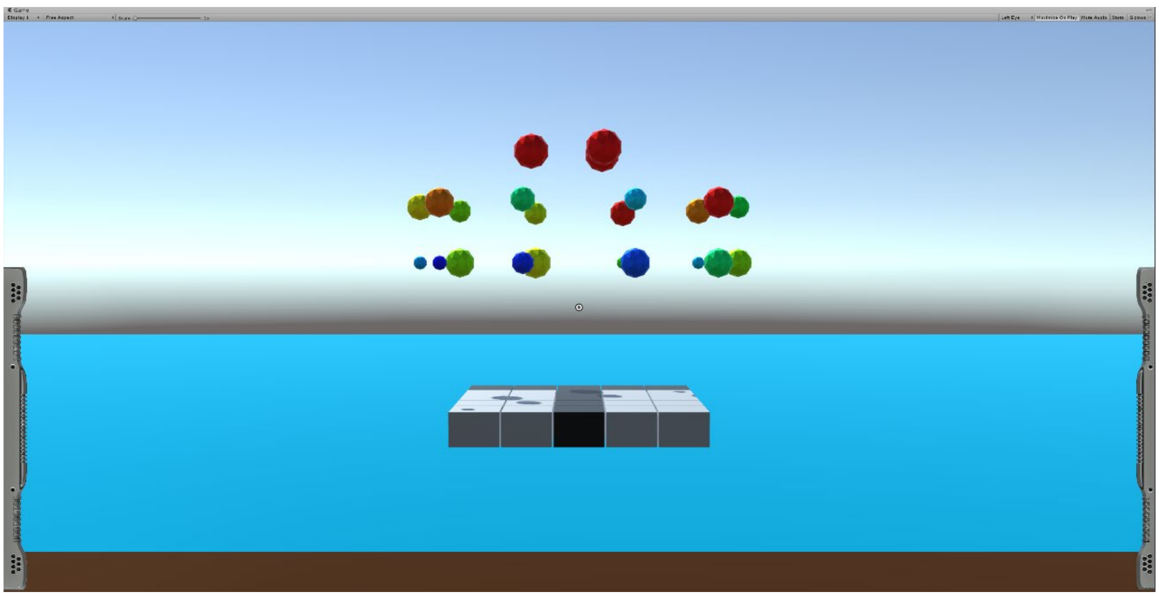

Figure 12: Early implementation of the ANN Visualization with Mixed Reality

The network was graphically shown in a vertical orientation with the layers laying out the nodes horizontally (Figure 14). This put the input layer at the bottom, the hidden layers above it at regular intervals, and the output layer above the hidden layers. All nodes were represented with low poly spheres that were spaced at equal distances from each other. This was an interesting structure that demonstrated how changes bubbled up from the lower layers up to the output layer. As described earlier in the paper, each value that changed was represented by a graphical change in the node. Size, color, and saturation were all constantly adapting to the latest data from the network.

This first iteration, while conforming to the initial illustrations proposed, was unfortunately fairly lacking in features and did not adequately demonstrate the network. This was due to several factors. A major factor was the lack of differentiation between the input layer nodes and the other nodes. A graphical demonstration of any concept should convey as much information as possible within a few seconds of glancing at the visualization. Because the differentiation was not present, an explanation would be needed to show what the user was looking at. This made for an inefficient visualization.

A second factor was that the node changes were not conforming to any sort of smoothing algorithm or limit. This led to drastic and rapid changes in the size and colors of the objects which would be confusing to a user. While this would accurately represent the network in the strictest sense, it would not permit the user to observe the trends of changes within the network.

The issue caused by the drastic shifts was that nodes would grow or shrink to extreme sizes that were not useful. For example, a node that was 1000 times larger than the smallest node will convey the scope of the difference beyond what is necessary and would only serve to distract. Another portion of this issue was that because each value reflected the output value exactly, the size would pulse. This is because as the node is refined, the value will overshoot or undershoot the eventual final value.

This was especially evident earlier in the playback when the network is beginning to establish itself. This pulsing was highly distracting and made it difficult to make observations about the network which is counterproductive to the efforts of this project.

A similar issue arose with the color changes where certain ranges of the color spectrum chosen were overutilized and overshadowed any potential observations of trends or patterns to learn from. A core component of this issue is that this first iteration relied on a color to value scale that was linear. This applied a static scale of measurement to the network that did not accurately reflect the weight of the difference in values between the node values which is what the network utilizes. This weighted difference is part of the true observable pattern to be visualized. This first iteration also lacked any environment to adequately house the visualization and focus the users' concentration. The network simply existed in an undefined 3D space.

4.2.4 Second Iteration

Improvements to the first iteration were made with a focus on improving the information that a user received visually from the demonstration. This was primarily accomplished by redesigning the input layer structure and visuals. There were three parts to this redesign: building a framework to handle specific inputs based on categories, new shapes, and a new graphical change scheme.

Remembering that comprehension can be improved with better visualizations and that *people can comprehend something visually much faster than audibly/written, the framework was updated to allow categorizing inputs. The sample network in this case used letters and symbols as inputs to learn and be able to identify them later. The network converted those symbols from a simulated pixel representation into an array of values to be used as inputs. Since it essentially began with a visual input, it was decided that it could be replicated on the graphical side to help facilitate the observations of the network. Under this concept the framework was updated to handle inputs categorized as graphical. By doing so, the framework can create an input layer that graphically resembles the original object.

To facilitate the new framework changes, the low poly spheres were replaced with three dimensional cubes. Doing this allowed them to be placed close together to create a more cohesive image representation while also maintaining a distinction between each node. This is similar to any digitally rendered image made up of pixels. From a distance, lines and shapes can seem to be cohesive as single objects. However, upon viewing the same image at the same resolution much closer shows distinct edges of each pixel that can easily be discerned. By changing the nodes to cubes, it was intended to replicate the pixel structure of images while achieving a balance between making the image discernable while still focusing on the idea each “pixel” was an input value.

To further strengthen the idea that each cube was a representation of a pixel, each cube was given a color based on the input value. For the given example, there were only two possible values: 0 or 1. Thus for each node with an input value of 0, the cube was shaded white while each with a value of 1 was shaded black. By doing this, it was possible to make the input layer show a recognizable letter or shape while still being abstract enough to not remove focus from the network nodes themselves (Figure 15).

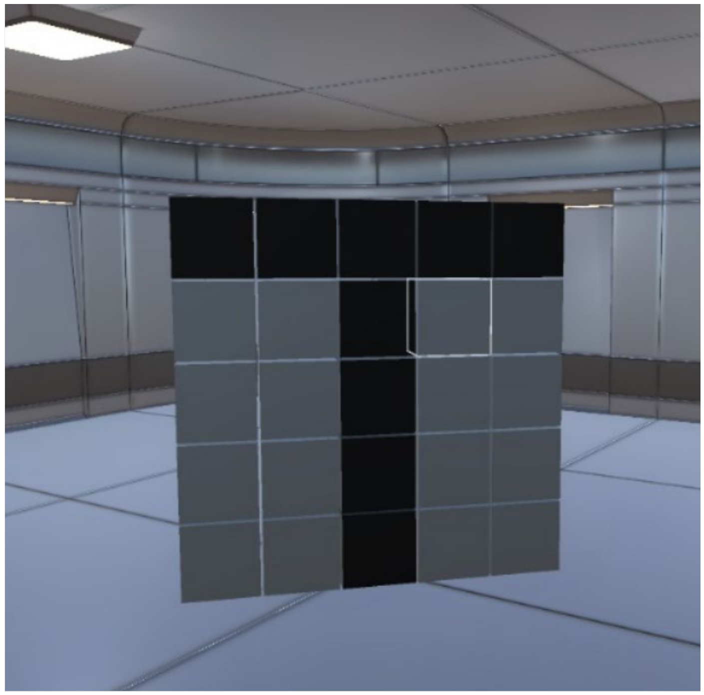

Figure 13: Letter Representation 'T'

The final update of this iteration was to mitigate the observed issues with the first regarding the colors and sizes changing at inconsistent rates. Colors were previously being changed based on a simple linear formula:

Node Value/Max Value * 360

This resultant value would then be used to choose the color via the HSV scale as the *hue*. Because the HSV scale is a circle, the quotient of node value and the max value was multiplied by 360 and rounded to the nearest whole number.

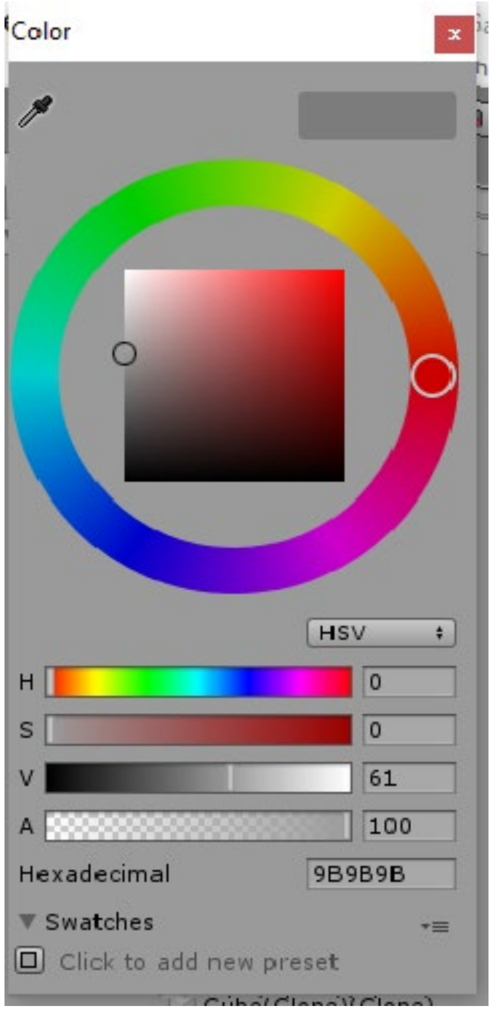

Figure 14: Sample HSV Color Wheel

The formula was changed instead to only encompass a certain range of values. This meant that colors would range between red and blue. Between the two endpoints were green and yellow as well as cyan. By constraining the range of colors, the visualization was stabilized a bit by reducing the potential color variation representing values.

Size was similarly limited by setting a floor for the values based on the overall maximum and minimum output values in the network. This was then multiplied against a scaling factor to smooth the noise that might occur from the value changing rapidly. While not a true smoothing function, it achieved its goal of removing the pulsing that was seen before as well as the extremes that were shown. By preventing the change from being excessive between frames, the fidelity of the visualization was maintained. The floor was set to make sure that the minimum size was always at least 10% of the maximum output value. This maintained a reasonable scale that was not seen in the previous iteration.

Some quality-of-life changes were also implemented to assist the observation process. A pause feature was added to allow the user to explore the network without being pressed for time to see a specific portion of the playback. An escape menu and underlying frameworks were also added at this time to support future changes, as necessary. To help with the previous issues of pulsing and distractingly quick changes, the playback was updated to slow down the changes between frames.

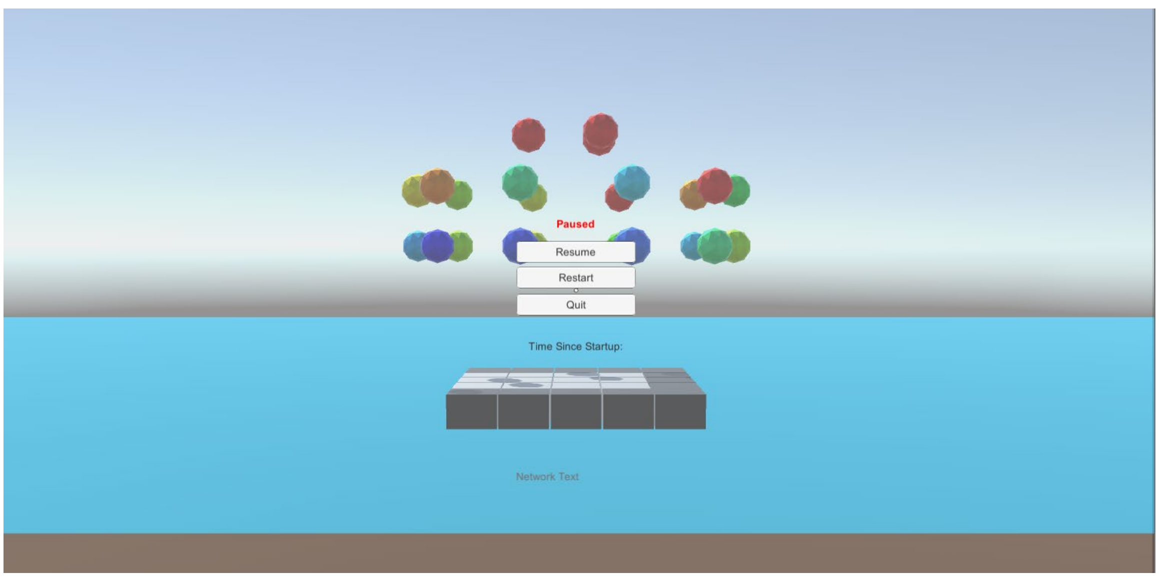

Figure 15: Early pause menu implementation

Though there were numerous improvements made as part of this iteration there were still issues that remained unresolved or were not completely resolved by the changes. Notably there was still no appropriate environment to encompass the visualization which would most likely reduce the efficacy of the visualization. Another issue noted was that the visualization still lacked clarity with regards to the input layer. While the improvements were made by moving to a pixel inspired representation, the positioning of the layer underneath, as seen in Figure 14, the rest of the network made it difficult to truly observe.

Another major consideration made at this time was that the user interactions were limited to moving around the visualization and pausing it. The project plan was updated to include more user interactions which can be read about earlier in the paper. The next iteration executed numerous updates for this particular purpose.

4.2.5 Third Iteration

As mentioned previously, the prior iterations lacked any meaningful user interactions which would be a major obstacle to the goal of maximum engagement by users. Additionally, any learning from the model would require outside instructional input. This third iteration involved enhancing the overall look and feel of the application by improving the environment, adjusting the model presentation, and adding UI components to the visualization.

The initial change for this iteration began with the introduction of an environment as seen in Figure 18 and Figure 19. ANN are part of the machine learning which is often viewed as the tool of choice for future advances in the world, so it was decided to implement a slightly more futuristic style environment to highlight this. The skybox background was set to be a representation of viewing a galaxy from space. This skybox however is only intended to be viewed incidentally and serves as a finite bound within the application. The player object along with the ANN model are housed inside a futuristic building created from prefabs found on the Unity Store. While this building does offer some incidental views of the skybox, it is not the focus for the player at any time.

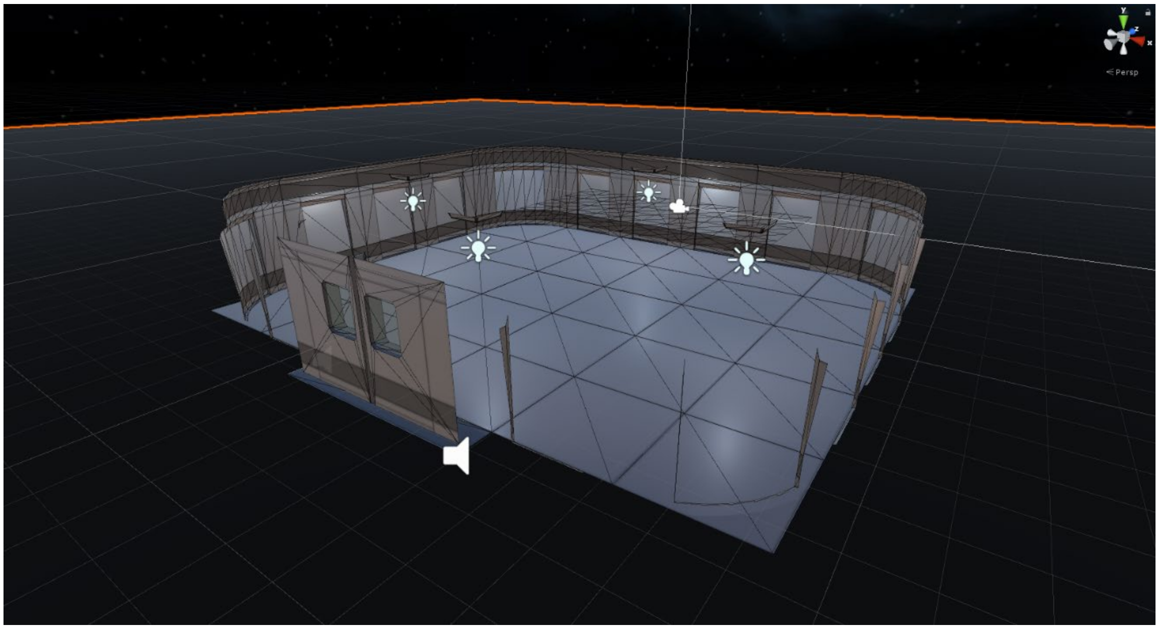

Figure 16: Environment exterior

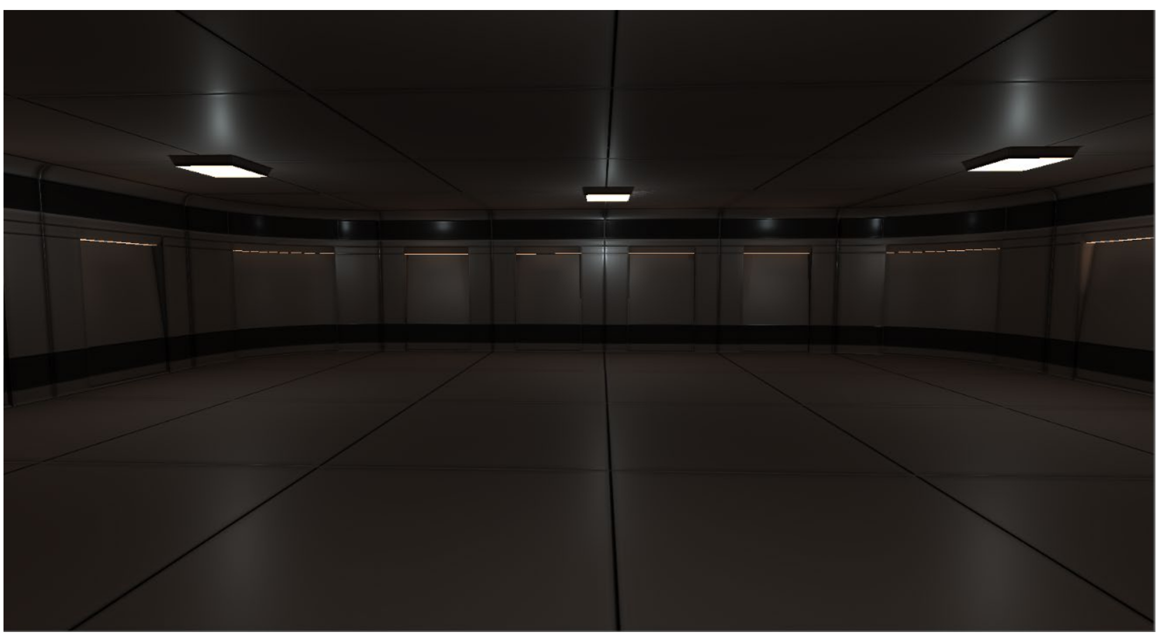

Figure 17: Environment interior

With the implementation of an environment and a bounded area for the user to navigate, it became necessary to further adjust the model and how it is generated. Prior to this change, the network model was generated at the global Unity scale with the only reference being to itself and the size of the nodes or the spacing of the layers. In order to make the experience more intuitive to a user, the model was scaled down to a smaller size. Changes were also made to the structure of the model so that each layer was encapsulated as children of a parent network object.

During the testing phase of this iteration and the subsequent feedback from the capstone committee members, it was determined the model itself also needed to be slightly reworked from the bottom-up visualization it was currently using. By rotating the model 90 degrees the layers were now vertical layers and the model flowed from left to right with the inputs on the left and the outputs on the right. This is a standard orientation for two dimensional depictions of ANN in traditional learning environments. Furthermore, the input layer was moved further to the left of the model and turned so as to give the user a better view of the patterns represented by the inputs.

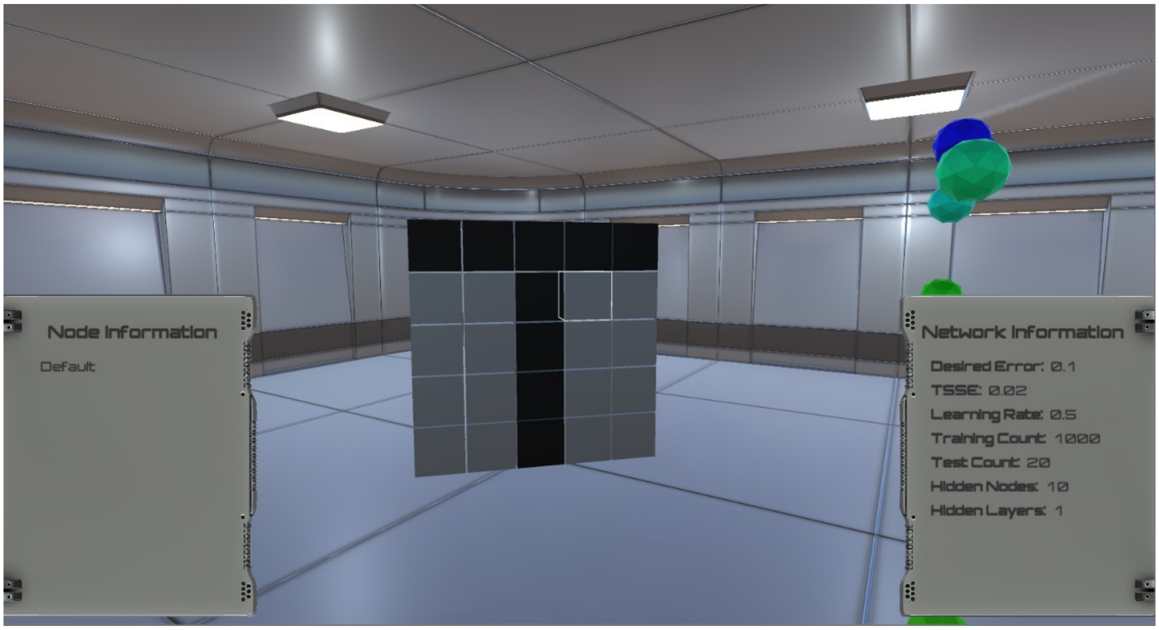

Figure 18: Input layer visualization show the letter 'T'

Without a UI or a way to convey information to the user, the model would require some supplemental source of instruction. To rectify this oversight, the UI as proposed earlier in this paper was implemented to allow the user the ability to better understand the network. The user was given two informational panels that would slide in from the sides of the viewport. By making these interactable, the user could choose to hide one or both as desired to be able to better observe the network as well as engaging them in the demonstration. The left panel was designated as the panel to show information about a selected node. This selection process is detailed further in this paper. The right panel is used to show more information about the network itself and what values were used to create it.

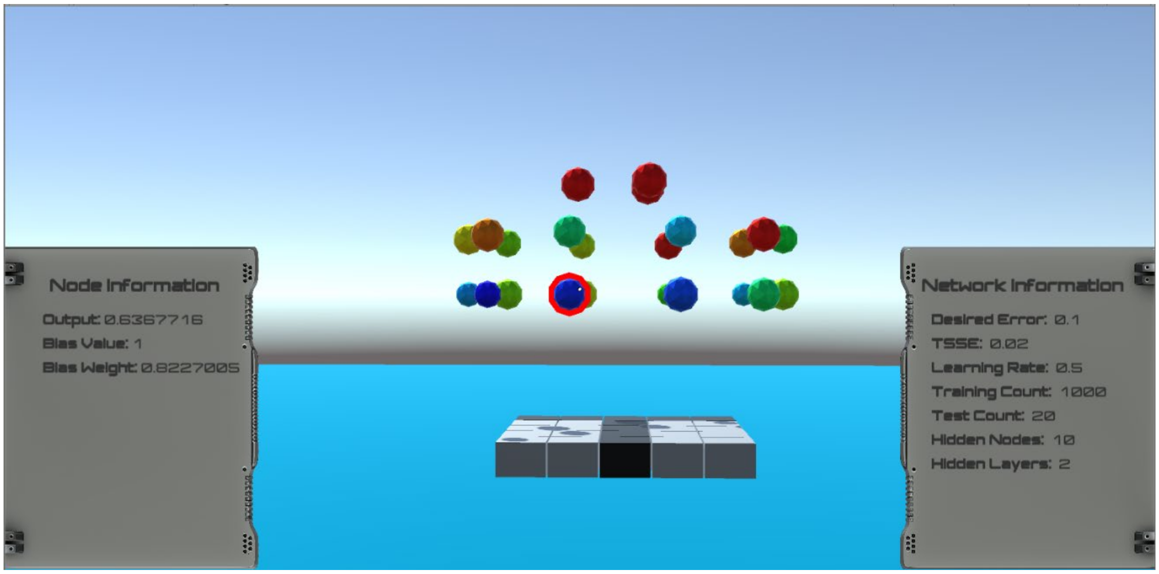

Figure 19: Sliding informational panels with a highlighted node corresponding to the left panel

Information about nodes is a key factor in this portion of the project. In order to populate the information panel and allow the user the freedom to delve into the data, an outline tool was found, implemented and adapted for use with this project. When the center of the user view port intersects with a node, it is outlined in a red bold outline. This makes it simple for the user to notice exactly which node object they were selecting. If the user were to click the left mouse button while a node was highlighted, the outline will turn white and the information for that node will be shown in the panel on the left. Furthermore, if the user has already selected a node and were to focus on a different node, the selected one will remain highlighted in white while the newly targeted node will outline in red. This makes it very simple for a user to be able to navigate the network to view each node as they desire.

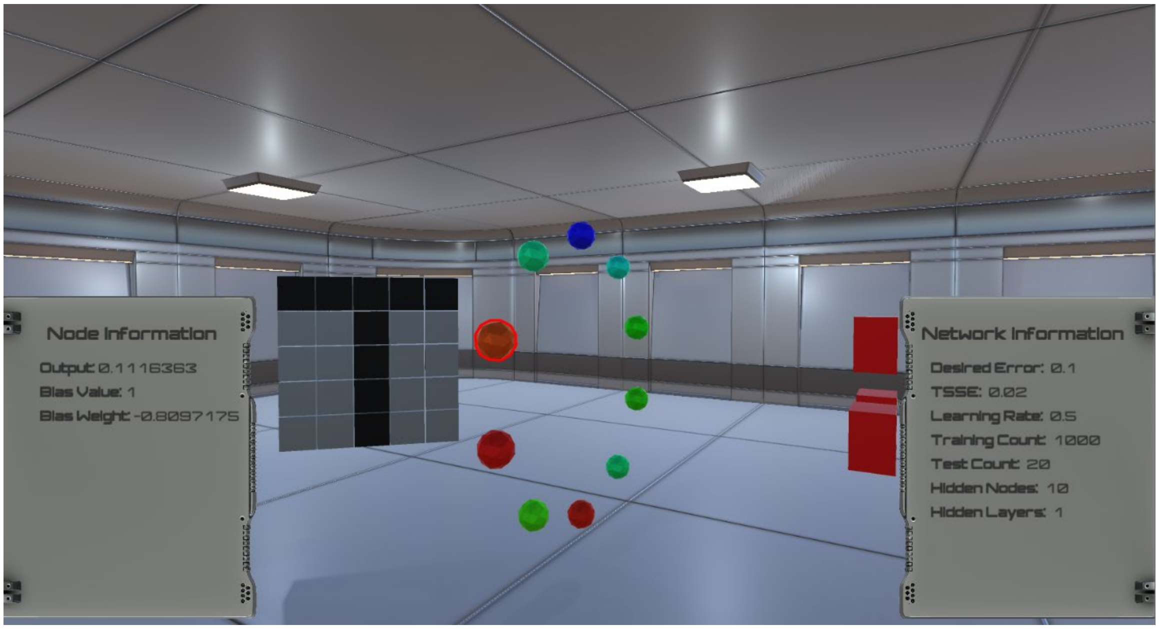

Figure 20: Finalized version of the second iteration.

4.3.1 Final Iteration

For the final iteration, it was necessary to implement numerous enhancements to better improve the user experience and provide an educational experience. These enhancements fell into four groupings: usability enhancements, educational modules, ANN model updates, and the removal of the VR framework.

4.3.2 Usability changes

The primary purpose of the usability enhancements was to facilitate the user experience by providing a simple, standard workflow. This was accomplished with the implementation of a main menu system, establishing a learning module path, and also allowing the user to select the stored network file they wish to view. When the app is first initialized, the user is taken to a main menu where they are presented with three options: Learn, Load, and Exit.

Exit is simply used to exit the application. Clicking on Learn will take the user to a menu with available modules that can be selected. The user can then click on the module they wish to learn. This will take the user directly to the visualization and provide them a guided experience. If the user clicks on Load, then they will be shown a menu with the available stored networks that they can choose from to explore freely.

Figure 21: Main menu

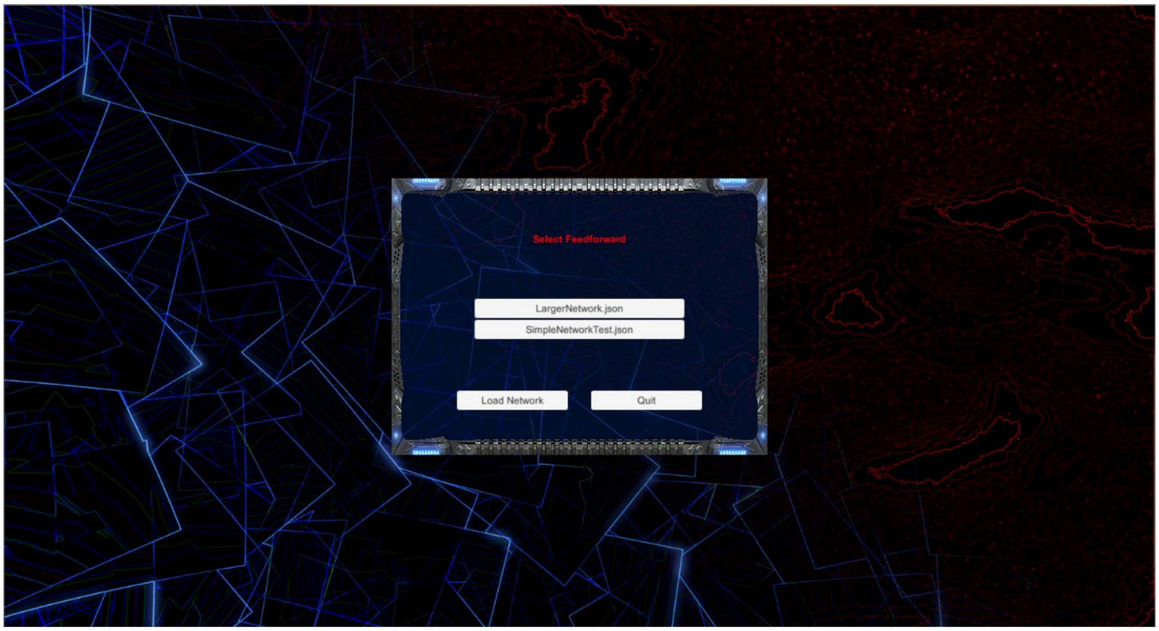

Figure 22: Menu to select a .JSON file to load

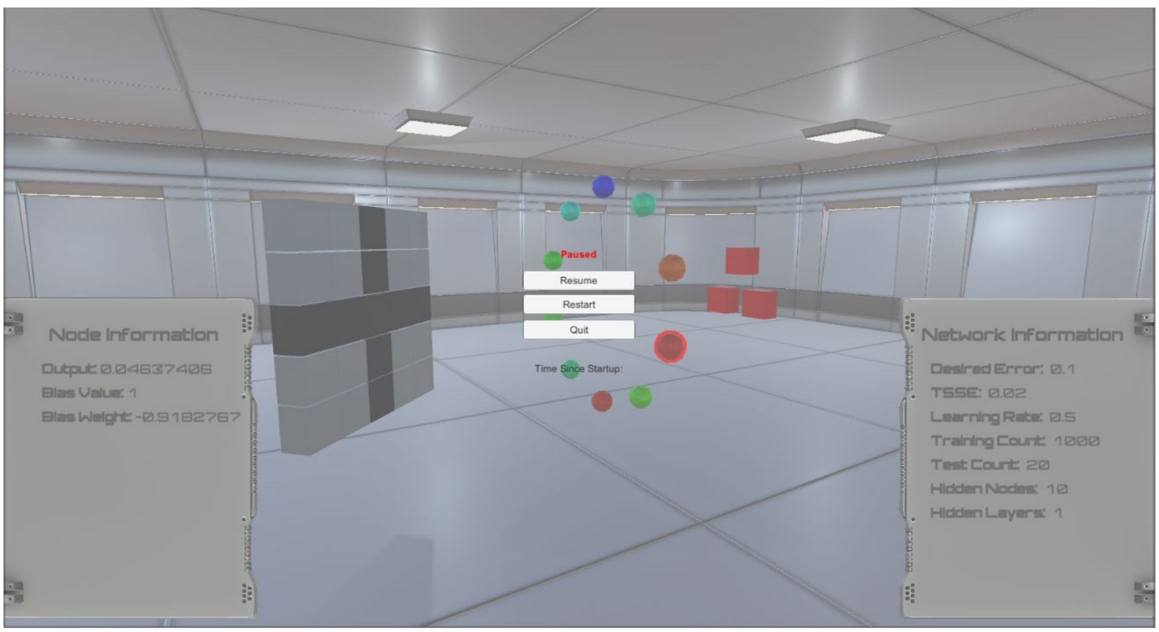

Figure 23: The pause menu as seen in the third iteration

4.3.4 Teaching panels

When the user clicks on a module in the Learn menu, they are taken directly to a visualization of an ANN. This experience is a guided experience with limited control allowed to the user. Upon loading the scenario and network, the user is shown a panel on the bottom right side of the screen which will provide the text associated with the module. In this situation, the informational panels that were implemented previously are not shown so as to not distract or confuse the user. The user is not able to freely move in this mode; instead, they will learn about the network through the text in the panel and observing the network visualization. Within the panel, the user will be able to page through various topics about the network. When doing so, there will be certain topics that will cause the camera to focus on specific parts of the network. This is intended to better engage the user and improve their comprehension of the lesson. The user can page both forwards and backwards.

The given learning module is a walk-through and explanation of the ANN and how it works. The user is guided through a series of learning objectives covering the concept of the network, what each layer is, how the nodes interact, what the data structure looks like, along with calculations and other details.

Figure 24: A teaching panel with information regarding the ANN

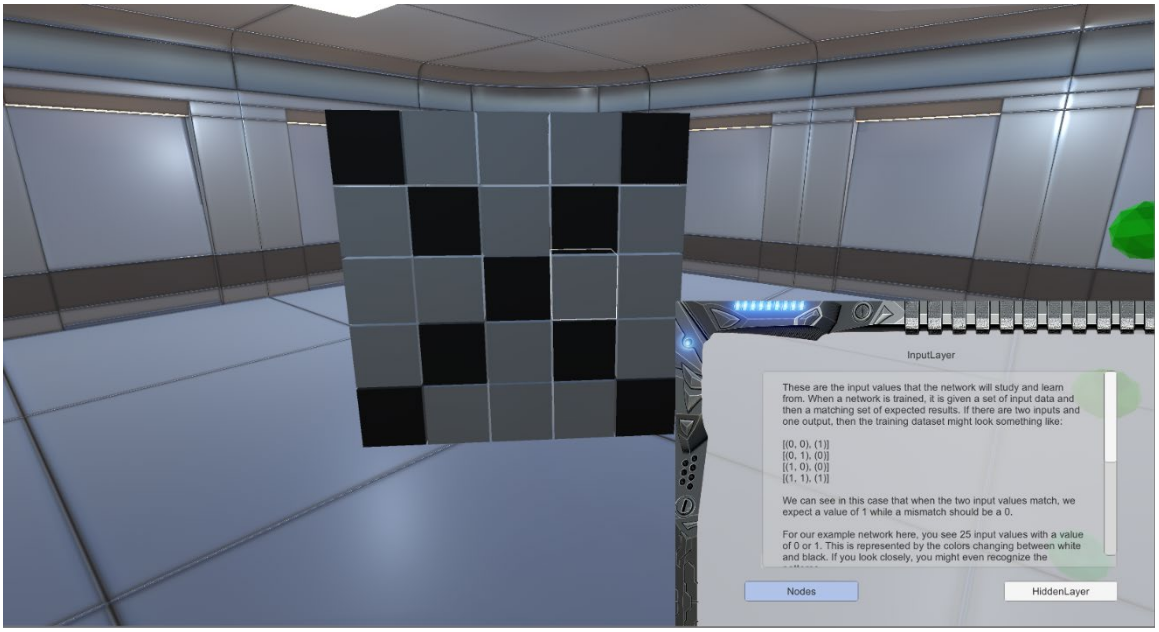

Figure 25: Specific panels will place the user view near the object described; the Input Layer in this case

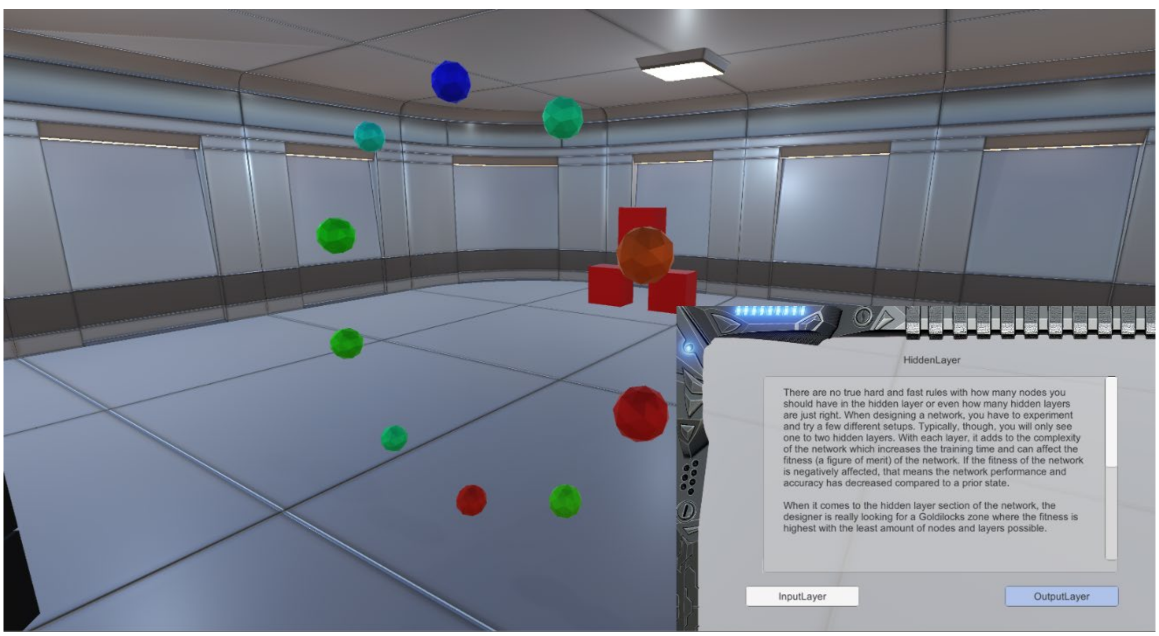

Figure 26: The teaching panel and view for learning about the hidden layer in an ANN

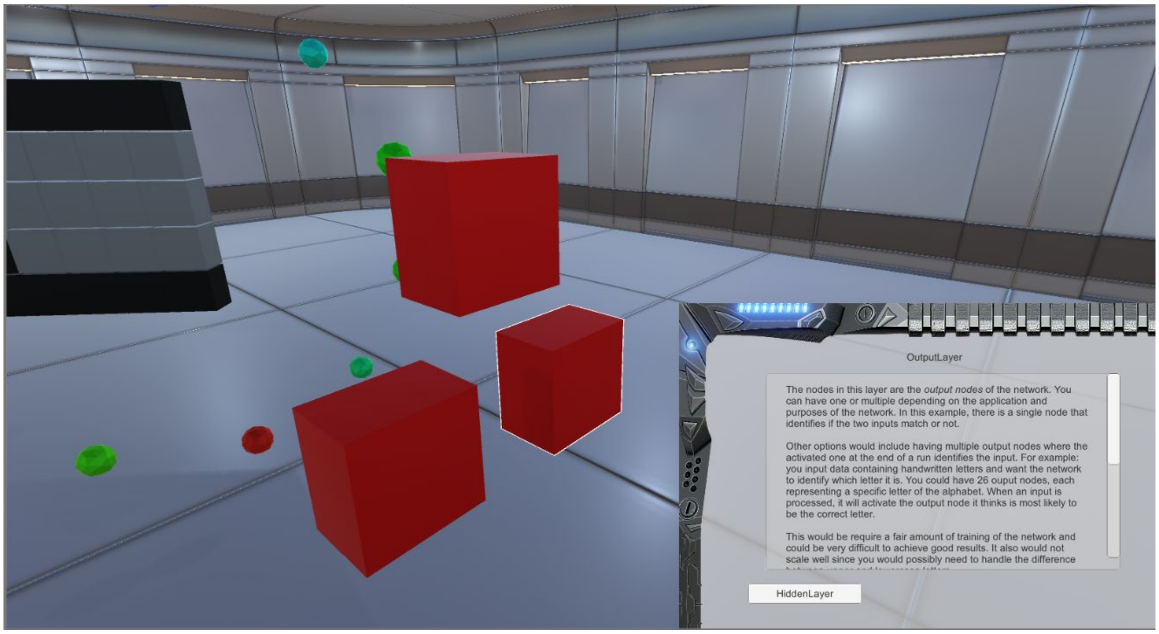

Figure 27: The teaching panel and view for learning about the Output Layer

4.3.5 Model changes

The network visualization model was further refined in this iteration by updating the underlying framework to utilize specific graphical assets based on the network metadata. This was achieved by updating the ANN logging code to store additional data about the input data by classifying it. For example, if the inputs are images, then the metadata will reflect this and the visualization can use the appropriate 3D assets when building the network to simulate the input image. This framework change also makes it possible to facilitate specific color schemes based on the classification.

4.3.6 Removal of VR

While making the adjustments and enhancements during this iteration, it was determined that the current VR implementation was problematic. The two primary issues were that the implementation utilized an early and no longer supported version of Windows Mixed Reality Toolkit (MRKT) for Unity, and the priority to provide a compelling graphical user experience superseded the VR aspect. Specifically, it was determined that much of the same benefits expected of a VR implementation could still be realized in a more traditional 3D video game setting while still allowing for future adaptation to VR. To update to a supported version of the MRKT for Unity would only be feasible with the complete extraction of the current implementation. Due to issues arising from the version currently implemented while also not having the time to upgrade to a new implementation, the MRKT framework was removed.

Another major consideration that influenced the decision to remove the VR aspect involved the fact that VR is still not quite commonplace. Initially users were to be invited to a location such as campus to demo the application by providing them with a PC and VR headset that they could use on site. Unfortunately, with the global Covid-19 pandemic and the challenge to safely allow multiple people to use the same device it was simply not feasible to provide users with the devices to test in VR. With laptops and desktop PCs being commonplace, it was far more feasible to be able to send the application to users that could run it on their own local machine than it would have been to find users that already owned a VR device compatible with the application. By focusing on the application for PC use, it would no longer be necessary to continue supporting VR.

Although it was determined that it would be necessary to remove the MRKT framework due to time and resource constraints, this also meant it would be necessary to create a user control structure that could be used to navigate the environment. Previously the user was able to navigate the environment due to the built-in control structure from the MRKT packages. It was necessary that a user control implementation was designed to still permit the user to be able to navigate the virtual room.

CHAPTER 5: USER TESTING AND SURVEYING

5. 1 Introduction

After discussion and consideration with the capstone committee, it was determined that it would be necessary to include a small-scale study as part of the project. The primary objective of this study was to determine how the user viewed the experience and how effective the tool was at teaching.

5.2 Study Design

To allow users to be able to test the application, it was necessary to devise the manner and method of distribution of the learning tool. As mentioned earlier, the Covid-19 pandemic made it necessary to utilize tools to allow users to test remotely. Using Unity's project build tool, the Windows 32-bit and Macintosh builds were generated. These builds were zipped using WinZip and hosted on Microsoft OneDrive for participants to be able to access via a shared link.

The data gathering portion of this study consisted of two surveys. The first survey was designed to get data about the participant and assess their current knowledge of machine learning concepts including about ANN. Participants were asked to answer questions related to their age, profession, and education level. This survey was designed to be completed just prior to demoing the learning tool.

The second survey assessed what knowledge the participant retained after completing the module as well as inquired about their overall experience with the application. At the end of the survey, participants could provide any additional feedback they desired to share. Both surveys were designed and administered through Qualtrics.

This tool made it very simple to design the surveys, distribute them, notify participants of changes, remind participants if they have not completed the surveys, and report on the results. (A copy of both surveys will be stored below in the appendix).

The surveys and file share location were sent to participants through the Qualtrics email service. In this email, instructions for Windows and Macintosh users were provided to ensure the process went as smoothly as possible regardless of the participants operating system of choice.

5.3 Pre-Evaluation Survey Results and Analysis

At the time of this writing, of the 16 participants invited to the study, 13 participants completed the first survey and 11 of those completed the learning module and responded to the follow up survey. Demographic data was gathered as part of the first survey and showed the following: 53% were between the ages of 20 and 29, all had some college or more with 30% of the participants holding a post graduate degree and 46% holding an undergraduate degree as their highest level of education. 50% of the individuals with a bachelor's degree or higher held degrees in natural sciences while 20% were in information systems.

Survey questions were also prepared to ascertain the participants experience with interactive learning tools and their knowledge of neural networks prior to the module. All participants had previous experience with video games and learning with the aid of computer software applications. 92% stated that they were slightly comfortable or better with using software to learn classroom concepts with final 8% stating they were neither comfortable nor uncomfortable with using software to learn. These results alleviate concerns that the data gathered might be impacted by users' inexperience with technology in general. It should also avoid the video game-like experience of the application distracting the participant due to the novelty of the game environment.

23% of the participants stated that they had never heard of neural networks prior to the study. In addition, these same participants along with one other stated they had never heard of machine learning. Over 50% of participants stated they would not be able to explain the concepts of a neural network adequately. This data was supported by the following section which tested their current knowledge about neural networks. This same section is utilized in the post-evaluation survey for comparison later.

The first question is a terminology question asking them to identify which of the choices given were typically part of a neural network. This question had three correct and two incorrect choices as well as the option to not select any. 15% of the participants correctly selected the three valid terms related to neural networks. 46% of all the participants were able to select at least two, although some also included a third incorrect choice. 77% selected at least one correct term. In this case the same term was chosen by all participants in this 77%.

The final six survey questions in the final section permitted the participants to either select an answer they believed was correct or to state they did not know the answer. 66% of the responses in this section utilized the choice to state they did not know the answer. Overall, only 11% of responses in this section selected the correct choice. Only one participant was able to answer more than 50% of the questions correctly in this portion.

Based on this pre-evaluation survey, it can be concluded that this cohort overall is not familiar with the concepts of neural networks and that much of what will be taught will be new material.

	Question	Choices (Bold is Correct)	Explanation
Q1	Which of the following are typically part of a neural network? (Select all that apply or none if none apply)	**Node**, **Connection**, Pipeline, **Bias**, Locus	Considered correct if two or more correct values are chosen.
Q2	How many input layers does a neural network have?	0, **1**, 2+, "I do not know", "Does not apply"	There is only one input layer in a neural network as taught in the module
Q3	How many hidden layers can a neural network have?	0, **1**, **2+**, "I do not know", "Does not apply"	As taught in the module, the network must have one hidden layer or more. Either the answer '1' or '2+' was accepted as a correct answer
Q4	How many test layers can a neural network have?	**0**, 1, 2+, **"I do not know"**, **"Does not apply"**	'Test Layer' is not a concept in neural networks. Therefore, any answer other than '1' or '2+' was considered correct
Q5	How many output layers does a neural network have?	0, **1**, 2+, "I do not know", "Does not apply"	There is only one output layer in a neural network as taught in the module
Q6	A neuron in the network can only have a bias or a weight, but not both.	True, **False**, "I do not know", "Does not apply"	A neuron can have both a bias value and weight
Q7	Connections in a neural network have a weight and output value.	True, **False**, "I do not know", "Does not apply"	Connections do not have an output value as taught in the module

Figure 28: Pre-evaluation Survey Knowledge-Based Questions

5.4 Post-Evaluation Survey Results and Analysis

The post-evaluation survey included the seven knowledge-based questions from the initial survey as seen in Figure 30. This survey also has two additional questions (Figure 31) that are similar to Q6 and Q7 and are meant to confirm the knowledge that was taught. By including the same questions as the earlier survey, it will be possible to see what knowledge was gained as well as where weaknesses in the teaching experience may exist. After analyzing the data there were several interesting results that will be expounded up in this section.

	Question	Choices (Bold is Correct)	Explanation
Q8	Do connections have an output value and a bias?	Yes, **No**, 'I do not know', 'Does not apply'	Connections were not taught as having output values or biases

Q9	Do hidden nodes have an output value and a bias?	**Yes**, No, 'I do not know', 'Does not apply'	They have both an output value and a bias

Figure 29: Post-evaluation Survey Additional Knowledge-Based Questions

The first comparison to be looked at will be the change between the pre- and post-evaluations for Q1. In the first survey, 77% of participants selected at least one correct choice. After going through the learning module, 100% of participants selected at least one correct choice. Only one participant selected the incorrect values during the post-evaluation survey, although they did still select one correct value. In Figure 32, the data show that the rate of the correct answers being chosen for this question increased across all correct answer. Meanwhile the rate of choosing incorrect ones decreased. It is worth noting at this time that of the three correct answers, 'Connection' was selected at the lowest rate with only 63% of participants selecting it. Later in this section will be an exploratory analysis of why 'Connection' was problematic for participants not only in this question, but also Q7 and Q8.

Value	Selection Rate - Pre	Selection Rate - Post
Node	77%	91%
Connection	38%	63%
Bias	8%	91%
Pipeline[2]	16%	9%
Locus[2]	16%	9%

Figure 30: Table demonstrating how many participants selected each value for the Pre-evaluation and Post-evaluation.

The pre-evaluation survey only saw 33% of participants on average attempting to answer any given question between Q2 and Q7 (inclusive). As expected, after the learning module, the participants rate of attempting to select a correct answer for any given question increased to an average rate of 93% from the earlier 33%. It appears that

[2] Incorrect choices that were provided. A lower selection rate is preferred

after going through the learning module, participants overall gained enough confidence in their knowledge to be able to select an answer for each question. However, as seen in Figure 33, the rate of correct responses being chosen varies between 18% and 82% with an average of 50% for Q2 through Q7. The only question (Q7) specific to connections was again the lowest value at a meager 18%. Q8 and Q9 had a response rate of 100% with a correct response rate of 36% and 72% respectively. Once again, the question with the lowest correct response rate is specific to connections.

	Attempt Rate – Pre	Attempt Rate – Post	% of Correct Responses – Post
Q2	38%	100%	45%
Q3	31%	91%	82%
Q4	15%	(See footnote)	45%
Q5	38%	91%	36%
Q6	23%	82%	73%
Q7	54%	100%	18%
Q8	n/a	100%	36%
Q9	n/a	100%	73%

Figure 31: Q2-Q7 Answer Attempt Rate - 'I do not know' was treated as a non-attempt.

The results of the comparison between the pre- and post-evaluation surveys clearly show that all participants increased in their knowledge of neural networks. This alone is not evidence enough that the visualization produced a net positive gain in knowledge acquisition and retainment as opposed to traditional text-based instruction accompanied by 2D illustrations of neural networks. This is partly due to the lack of a control group that might have learned through the more traditional path and participated in the same surveys. Had this control group been established, comparisons could have been made between the two groups to better determine the effectiveness of the learning tool.

To still provide insight into the effectiveness of the tool, survey questions were included that established the participants perceived comfort level with the material and how effective they felt the visualization was in improving their comprehension. The data in Figure 34 compares the perceived effectiveness of the model with the participants rate of correct responses. It also includes their *net improvement* which in this context is described as the change in score because of changing a previous answer. Interestingly the participants that selected the lowest value in the range of responses and the participants that selected the highest value in the range of selected responses, all scored 56% or less on the knowledge evaluation section during the post-survey. It is also worth noting that 91% of participants felt they could competently explain the concept of a neural network as opposed to just 46% before going through the learning module. This along with the increased response rate could signify that the participants overall felt confident in their knowledge and retention.

Participants	How much did the associated visualization impact your comprehension of neural networking concepts?	% of Correct Responses	Net Improvement*
1	Somewhat positively	89	4
2	Somewhat positively	78	2
3	Somewhat positively	78	1
4	Somewhat positively	67	1
5	Somewhat positively	56	4
6	None at all	56	-3
7	Positively	44	-2
8	None at all	44	1
9	None at all	44	-1
10	Positively	33	-2
11	Somewhat positively	33	0

Figure 32: Comparing the perceived effectiveness of the model with the knowledge evaluation. Net improvement describes the change from a previous response.

The sentiment that participants felt they could explain the concept of a neural network along with the increased response rate could signify that the participants overall felt confident in their knowledge and retention. Based on the figure 34, there appear to be three groups that can be analyzed. The first group are the top 4 scorers that make up the majority of users that felt the visualization was somewhat positive in its effect. This group also benefited the most from the visualization tool. The second group are the users that felt the effect was positive. The third are the users that felt it did not improve their comprehension at all. It is notable that the two extremes of the selected responses also resulted in an average negative result when answering questions as compared to the pre-evaluation survey.

Before drawing a conclusion, it is necessary to also look at the issue with questions regarding connections. As explained previously, only 63% of participants selected connection as a term related to neural network while the other two correct terms were selected by 91%. Only 18% and 36% of participants correctly answered Q7 and Q8 respectively, both of which also relate to connections. The only major difference between how connection was taught versus the other components of a neural network with this module is that the connections were ***not*** visualized. The teaching of connections is strictly limited to the text within the teaching panels.

Only two participants (Participant 1 and 3) correctly answered both the Q7 and Q8 connection questions. These participants also correctly selected the term for the terminology question, thus also being the only participants to answer correctly all three questions that include connections. Based on the data, it can be concluded that most participants (68%) recognize the term to be related to the network based on the reading portion, but without the visualization the concept of the connection was not well retained at a rate of less than 50% of the other terms on average.

It is hypothesized that for the group that felt the visualization was maximally positive for them, the lack of the connection visualization limited their retainment of knowledge. For the group that saw no perceived benefit of the visualization, it is not clear if the visualization of the connection would have assisted their knowledge retention. There is the potential that the lack of perceived benefit was in part due to the implementation lacking the connection visualization or that the implementation was lacking overall to meet their own personal threshold for effectiveness. This reason could also equally contribute to the prior group's poor performance despite the perceived positive effect. Additionally, because the group that most improved and scored the highest overall thought the impact of the visualization was moderate indicates that the tool was effective with the visualization and the text provided combined.

If the visualization were exceptionally suited to its purpose along with the text, it would be expected that the data would show high scores and better knowledge retention. Conversely if the visualization were exceptional while the text portion lackluster, it would be expected to see higher ratings for the perceived impact while scores for participants 7 and 10 would increase. The scores for participants 6,8, and 9 would only increase if their low perception were due to a lack of quality in implementation and not a lack of competency in visual learning. If the visualization were to have been implemented at a subpar level or simply not be an effective tool, it would be expected that there would be lower ratings for user perception overall with the scores as shown. Furthermore, no responses show that participants thought that the visualization had a negative impact of any level. It is therefore assumed that the implementation is at least adequate for the purposes of this small-scale study.

In conclusion, based on the data received, the visualization does not negatively impact learning. The visualization had an overall positive perceived effect on users and with some improvements could increase the magnitude of that positive effect, particularly with users that more highly relied on the visualization.

CHAPTER 6: CONCLUSIONS AND FUTURE WORK

6.1 Conclusion

At the outset of this project, there were three main development objectives: build an artificial neural network, design, and implement a framework to store all states of a neural network, and to design an application that can ingest that stored network and play it back in three-dimensional space. The amalgamation of these three objectives provided a proof-of-concept learning tool for teaching about neural networks. The tool's goal was to effectively teach individuals with at least some college level education about these networks.

With online surveys and gaining the participation of numerous individuals, it was possible to complete an initiatory small-scale study on the effectiveness of the learning tool. The results showed an overall positive trend in the retention of knowledge as well as an overall positive perception of the effectiveness of the tool. It also demonstrated that concepts which were only explained in the text were not well retained when compared to those which also had a corresponding visualization. Due to the small-scale nature of this study, a more in-depth statistical analysis was not possible. The conclusions drawn from this small study contain enough evidence to not preclude further research and study into this project and the effectiveness of the tool. With a larger study, control groups and general improvements to the tool, such as minor bug fixes, it will be possible to gain a better understanding of the impact the tool can have on learning outcomes.

6.2 Future Work

One of the major components of an ANN is the concept of connections. These connections are how each layer interacts with the adjacent layers and is therefore a key part of the ANN. Unfortunately, this implementation of the ANN visualization did not include the connections as originally intended. If this project were to continue, the inclusion on the connections in the visualization would be a priority. By including this aspect of the ANN in the visualization, the user will be better able to visualize the way each layer interacts with the adjacent layer over time.

Another aspect of the application that would be ideal to improve upon would be the playback control structure. The basic pause functionality is present and the framework for complete control is coded, but the front-end integration with this was incomplete due to time constraints. This would be a boon to the learning process where a user could speed up, slow down, or even reverse the changes in the network to better observe how the network changes and learns.

Although the framework was finished on the Java side to be able to input parameters and receive a file containing all the network states, the unity project framework was never integrated with it. In future work with this project, the ideal path to complete this aspect would be to setup up an API interface between the Unity project and the Java project to be able to seamlessly enable this process. However, this would still have the limitation of only being able to use a stored and recorded version of the network. If the Java framework were to be reworked, or if the network was changed to run in C# within the unity project, the user could have more control over the network and be able to make changes while the network is running. This would open further avenues for the user to learn and for potential professional use cases with this tool.

REFERENCES

1.) Akpan, Ikpe Justice, and Murali Shanker. “A Comparative Evaluation of the Effectiveness of Virtual Reality, 3D Visualization and 2D Visual Interactive Simulation: An Exploratory Meta-Analysis.” SIMULATION, vol. 95, no. 2, Feb. 2019, pp. 145–170, doi:10.1177/0037549718757039.

2.) Elinda Ai-Lim Lee, Kok Wai Wong, “Learning with desktop virtual reality: Low spatial ability learners are more positively affected”, Computers & Education, Volume 79, 2014, Pages 49-58,
ISSN 0360-1315, https://doi.org/10.1016/j.compedu.2014.07.010.

3.) “Active Virtual Reality Users Worldwide 2014-2018 | Statistic.” Statista, www.statista.com/statistics/426469/active-virtual-reality-users-worldwide/

4.) O’Connor, Eileen A., and Jelia Domingo. “A Practical Guide, With Theoretical Underpinnings, for Creating Effective Virtual Reality Learning Environments.” Journal of Educational Technology Systems, vol. 45, no. 3, Mar. 2017, pp. 343–364, doi:10.1177/0047239516673361.

5.) Fowler, Chris. “Virtual Reality and Learning: Where is the Pedagogy?” British Journal of Educational Technology, vol. 46, no. 2, 2015, pp. 412-422, https://doi.org/10.1111/bjet.12135

6.) Guido Makransky, Thomas S. Terkildsen, Richard E. Mayer, “Adding immersive virtual reality to a science lab simulation causes more presence but less learning”, Learning and Instruction, Volume 60, 2019, Pages 225-236, ISSN 0959-4752.

7.) Rolando, J., Wijekumar, K. (., & Barnes, R. (2018). VR TRAINING SOFTWARE: Research shows strong results for learners. Professional Safety, 63(12), 35-38.

8.) Few, Stephen. Show Me the Numbers. GAZELLE Book SERVICES, 2004.

9.) Ware, Colin. Information Visualization: Perception for Design. Elsevier/Morgan Kaufman, 2004

10.) King, R. (2019). Facebook's latest multi-billion buy: 3D gaming company Oculus | ZDNet. [online] ZDNet. Available at: https://www.zdnet.com/article/facebooks-latest-multi-billion-buy-3d-gaming-company-oculus/

11.) The Untold Story of Magic Leap, t. (2019). The Untold Story of Magic Leap, the World's Most Secretive Startup. [online] WIRED. Available at: https://www.wired.com/2016/04/magic-leap-vr/

12.) CNBC. (2019). Why Walmart and other F500 companies are using virtual reality to train the next generation of American workers. [online] Available at: https://www.cnbc.com/2018/10/29/why-f500-companies-use-virtual-reality-to-train-workers-of-the-future.html

13.) Chen, Y. (2016). The effects of virtual reality learning environment on student cognitive and linguistic development. The Asia - Pacific Education Researcher, 25(4), 637-646. doi:http://dx.doi.org.liblink.uncw.edu/10.1007/s40299-016-0293-2

14.) Nikunj Chauhan, V. Ravi, D. Karthik Chandra, Differential evolution trained wavelet neural networks: Application to bankruptcy prediction in banks, Expert Systems with Applications, Volume 36, Issue 4, 2009, Pages 7659-7665, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2008.09.019.

15.) Fernandes, F. A. N., & Lona, L. M.F.. (2005). Neural network applications in polymerization processes. Brazilian Journal of Chemical Engineering, 22(3), 401-418. https://dx.doi.org/10.1590/S0104-66322005000300009

16.) Medium. (2019). Building Purposeful UI Using Pre-attentive Attributes. [online] Available at: https://medium.com/design-at-zoopla/building-purposeful-ui-using-pre-attentive-attributes-9c5ee5dcc25c

17.) http://www.scholarpedia.org/article/Gestalt_principles

18.) https://docs.unity3d.com/Manual/Windows-Mixed-Reality-Input.html

19.) Cüneyt Karul, Selçuk Soyupak, Ahmet F. Çilesiz, Nihat Akbay, Emin Germen, (2000).

Case studies on the use of neural networks in eutrophication modeling. Ecological Modelling, Volume 134, Issues 2–3, 145-152,

http://www.sciencedirect.com/science/article/pii/S0304380000003604

20.) Kalani, E. , Elhami, A. , Kazem-Zadeh, R. and Kamrani, E. (2018) Selection of Investment Basis Using Neural Networks in Stock Exchange. *American Journal of Industrial and Business Management*, **8**, 548-562.

doi: 10.4236/ajibm.2018.83036.

21.) Heston, S. L., & Sinha, N. R. (2017). News vs. sentiment: Predicting stock returns from news stories. *Financial Analysts Journal, 73*(3), 67-83. Retrieved from https://search-proquest-com.liblink.uncw.edu/docview/1953804675?accountid=14606

Appendix 1 – Unimplemented proposed items

3.6.1 Monitored Characteristics:

Users will be able to determine which of the ANN changes they wish to see. They will have the ability to turn off functionality such as the bias value colorizing or the size changes of a given connection. It is expected that most users would get the best experience by not being given all the functions initially and therefore should be able to progress through versions of the visualization. This interaction will be managed via four options in a menu with the following labels: Bias Value, Bias Weight, Output, and Connection. If active, the button will be lit up to indicate that the characteristic will be visually adapted during the training process. Inactivity will be indicated by the lack of a back light for that button. The user will be able to select all, some, or none of the buttons. In order to access this menu, the user will need to press the *grip button* as shown in

3.6.2 Network Adjustment Controls:

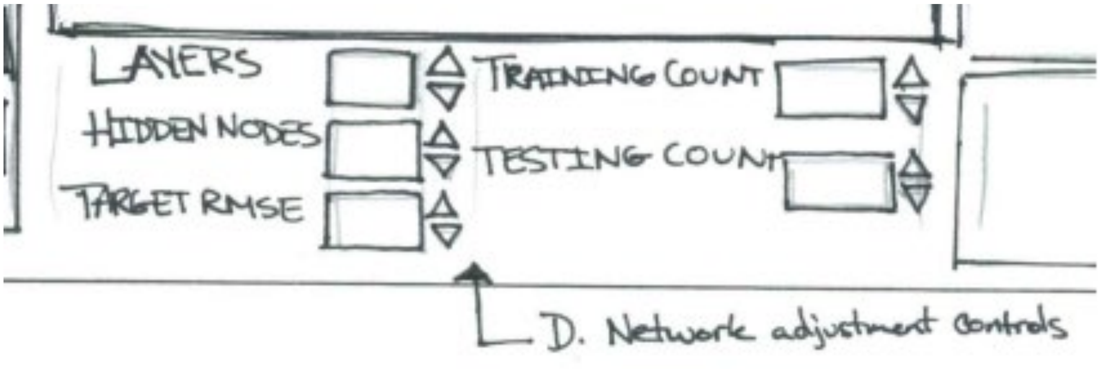

Figure 33: Network adjustment menu options

These controls will show the number of hidden layers, number of hidden nodes, target RMSE, training iteration count, and testing iteration counts. This set of controls will be part of a menu for new scenarios only and cannot be utilized with recorded networks. The buttons will also not be active during a scenario playback regardless of whether it is recorded or not prior to the playback. If a current playback is underway, the user will need to click the stop button in the playback controls. Upon doing so, the adjuster controls will be unlocked.

In order for the user to be able to change the values the user will need to point their controller at the field for a given input and select it with the trigger. The user will then be able to use the d-pad up and down buttons to increase or decrease the values. Each field will be given a hardcoded limit that cannot be exceeded due to the limitations of this project.

3.6.3 Playback:

Users will also have control over the playback much like a video player. Users can pause the animation to relocate their current position and focus on certain sections of the network. The user will have access to a play/resume function as well as a stop function.

The speed functions along with the play/pause functionality will not be available for scenarios which are not utilizing recorded assets. If the scenario is a new version, then the user will only have access to the Stop button for playback. In a recorded scenario, the stop button will halt the current network scenario playback. In a new scenario the stop button will halt the current process and any changes to the network will not be saved or recorded.

In order for the user to be aware of how the recording is currently running, there will be a part of the user interface (UI) that will always be visible with a play/pause icon, fast forward icon, and a rewind icon. Whichever function is active will have the corresponding icon highlighted. This set of icons can be seen in the bottom middle half of Figure 4 .

Appendix 1

```
[
    [
        [
            {
                "desiredError": 0.1,
                "TSSE": 0.02,
                "RMSE": 0.1,
                "acceptablePassRate": 0.95,
                "learningRate": 0.5,
                "trainingCount": 1000,
                "testCount": 20,
                "hiddenNodeLayerSize": 3,
                "hiddenNodeLayerCount": 1,
                "dataType": "NNObj"
            }
        ],
        [
            {
                "trainingObjs": "[{\"desc\":\"Low Low\",\"dVals\":[0.1,0.1],\"dVals_targets\":[0.9],\"id\":0},{\"desc\":\"Low High\"
            },
            {
                "testingObjs": "[{\"desc\":\"Low Low_t\",\"dVals\":[0.1,0.1],\"dVals_targets\":[0.9],\"id\":0},{\"desc\":\"Low High_
            }
        ],
        [
            {
                "1":
                "{\"passCountTotal\":0,\"failCountTotal\":0,\"testCountTotal\":0,\"trainCountTotal\":1,\"totalCountTraining\":0,\"in
                0291,\"inputConnections\":[{\"weight\":1.080843191402366,\"id\":2},{\"weight\":0.42774373382190495,\"id\":8}],\"outp
                9067624354867,\"inputConnections\":[{\"weight\":-0.40745443080060145,\"id\":4},{\"weight\":0.961702352591519l,\"id\"
                tVal\":0.5025972428206659,\"inputConnections\":[{\"weight\":0.847914235751029l,\"id\":6},{\"weight\":0.559436532316
                0.6313489192281015,\"outputVal\":0.6527952625840135,\"inputConnections\":[{\"weight\":1.0136391924976305,\"id\":3},{
            },
            {
                "2":
                "{\"passCountTotal\":0,\"failCountTotal\":0,\"testCountTotal\":0,\"trainCountTotal\":2,\"totalCountTraining\":0,\"in
                29144,\"inputConnections\":[{\"weight\":1.0794491854392154,\"id\":2},{\"weight\":0.4151976801535492,\"id\":8}],\"out
                045640588245,\"inputConnections\":[{\"weight\":-0.40753426796827036,\"id\":4},{\"weight\":0.9609838180824989,\"id\":
                al\":0.6118097996562 6,\"inputConnections\":[{\"weight\":0.8485247201406728,\"id\":6},{\"weight\":0.5649308918237178,
                2087909092605,\"outputVal\":0.6744120810686548,\"inputConnections\":[{\"weight\":0.9708375805740165,\"id\":3},{\"wei
            },
            {
                "3":
                "{\"passCountTotal\":0,\"failCountTotal\":0,\"testCountTotal\":0,\"trainCountTotal\":3,\"totalCountTraining\":0,\"in
                7196,\"inputConnections\":[{\"weight\":1.069970381833431,\"id\":2},{\"weight\":0.41414447975290647,\"id\":8}],\"outp
                0816967705297,\"inputConnections\":[{\"weight\":-0.4077217804986571,\"id\":4},{\"weight\":0.960962983356 9003,\"id\":
                tVal\":0.6666143803356834,\"inputConnections\":[{\"weight\":0.8541292871932123,\"id\":6},{\"weight\":0.5655536214962
                657474905401 6304,\"outputVal\":0.6586929330770075,\"inputConnections\":[{\"weight\":0.9219805639351157,\"id\":3},{\"
            },
```

Appendix 2 Pre-evaluation Survey

Neural Network Visualization Tool Pre-evaluation

Thank you for agreeing to be an early beta tester for this virtual learning tool.

Please answer the following survey questions as seen below, and then proceed to walk through the learning tool application. A follow-up survey will be provided after you have completed the walk-through.

Please select your age group

- 18-29
- 30-39
- 40-49
- 50+

Please choose your level of education

- No formal education
- Some formal education
- High School
- Some College
- College Graduate
- Post Graduate

If you completed a college degree, what was your major?

Please list your area(s) of professional expertise

How comfortable are you with using technology such as computer programs to learn classroom concepts?

- Extremely comfortable
- Moderately comfortable
- Slightly comfortable
- Neither comfortable nor uncomfortable
- Slightly uncomfortable
- Moderately uncomfortable
- Extremely uncomfortable

Have you ever played video games?

- Yes
- No

How often do you play video games?

- Always
- Most of the time
- Occasionally
- Rarely
- Never

Have you ever heard of machine learning before?

- Yes
- No

Have you ever heard of neural networks before?

- Yes
- No

How well do you feel you would be able to explain the concept of a neural network?

- Extremely well
- Very well
- Moderately well
- Slightly well
- Not well at all
- Not at all

Which of the following are typically part of a neural network? (Select all that apply or none if none apply)

- Node
- Pipeline
- Connection
- Bias
- Locus

How many input layers does a neural network have?

- 0
- 1
- 2+
- I do not know
- Does not apply

How many hidden layers can a neural network have?

- 0
- 1
- 2+
- I do not know
- Does not apply

How many test layers can a neural network have?

- 0
- 1
- 2+
- I do not know
- Does not apply

How many output layers does a neural network have?

- 0
- 1
- 2+
- I do not know
- Does not apply

A neuron in the network can only have a bias or a weight, but not both.

- True
- False
- I do not know
- Does not apply

Connections in a neural network have a weight and output value.

- True
- False
- I do not know
- Does not apply

Appendix 3 Post-evaluation Survey

Neural Network Learning Tool Post-evaluation

Do you feel the tool helped improve your overall knowledge of neural networks compared to what you knew beforehand?

- Extremely positively
- Somewhat positively
- Neither positively nor negatively
- Somewhat negatively
- Extremely negatively

How well would you be able to conceptually explain neural networks after going through the module?

- Extremely well
- Very well
- Moderately well
- Slightly well
- Not well at all

How easy was it to navigate the tool?

- Extremely easy
- Moderately easy
- Slightly easy
- Neither easy nor difficult
- Slightly difficult
- Moderately difficult
- Extremely difficult

Do connections have an output value and a bias?

- Yes
- No

Do hidden nodes have an output value and a bias?

- Yes
- No

How well do you feel you would be able to explain the concept of a neural network?

- Extremely well
- Very well
- Moderately well
- Slightly well
- Not well at all
- Not at all

Which of the following are typically part of a neural network? (Select all that apply or none if none apply)

- Node
- Pipeline
- Connection
- Bias
- Locus

How many input layers does a neural network have?

- 0
- 1
- 2+
- I do not know
- Does not apply

How many hidden layers can a neural network have?

- ○ 0
- ○ 1
- ○ 2+
- ○ I do not know
- ○ Does not apply

How many test layers can a neural network have?

- ○ 0
- ○ 1
- ○ 2+
- ○ I do not know
- ○ Does not apply

How many output layers does a neural network have?

- ○ 0
- ○ 1
- ○ 2+
- ○ I do not know
- ○ Does not apply

A neuron in the network can only have a bias or a weight, but not both.

- ○ True
- ○ False
- ○ I do not know
- ○ Does not apply

Connections in a neural network have a weight and output value.

- True
- False
- I do not know
- Does not apply

How much did the associated visualization impact your comprehension of neural networking concepts?

- Positively
- Somewhat positively
- None at all
- Somewhat negatively
- Negatively

Please share any additional feedback you have about the application:

____