

Visualizing Machine Learning in 3D

Diego Rivera
drivera13@gaels.iona.edu
Iona College
New Rochelle, NY, USA

ABSTRACT

Understanding machine learning models can be difficult when the models at hand have many parts to them. Having a visual model can help aid in understanding how the model functions. A way to visualize these models is to use a 3D (three-dimensional) game development application. An application that will have an interactive element allowing the users to interact with the model (rotating and scaling it) and see changes at run-time. An interactive element will keep the users engaged, understanding, and seeing how a machine learning model looks and behaves. This paper describes the process of visualizing a machine learning model in a 3D application.

CCS CONCEPTS

• **Human-centered computing** → *Virtual reality*.

KEYWORDS

Neural Networks, Transformers, Interactive

ACM Reference Format:

Diego Rivera. 2022. Visualizing Machine Learning in 3D. In *28th ACM Symposium on Virtual Reality Software and Technology (VRST '22)*, November 29-December 1, 2022, Tsukuba, Japan. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3562939.3565688>

1 INTRODUCTION

Machine learning can be a scary topic for some people as many see it as hard and advanced code that takes years to understand in order to implement it. Machine learning is not as hard nowadays, with many tutorials and applications dedicated to helping to create machine and deep learning models. Virtual Reality is something that has been in development for years, and many users seem to enjoy the virtual applications that were created. Machine and deep learning can be an application that will work well in virtual reality. Creating these models and making them interactive will allow machine learning to be open to the public and hopefully make create a new generation of students willing to work with machine learning. A virtual experience can provide knowledge and experience better than textbook work. A virtual experience can provide fun interactions to keep the user engaged.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

VRST '22, November 29-December 1, 2022, Tsukuba, Japan

© 2022 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-9889-3/22/11.

<https://doi.org/10.1145/3562939.3565688>

2 RELATED WORK

There was a three-dimensional interactive neural network that was developed back in 2018. The program illustrated had a feature that allowed the user to see changes in real-time and have limited movement around the model. The program Stefsietz [Sietzen [n. d.], 2022] created was a CNN model that had to change parameters and the physical appearance of the model. The model was very well built, and the application worked as it intended. Stefsietz created the application via Unity and Json files which store the parameters and weights for the model that gets created in Unity.

3 METHODOLOGY

In developing the application, Unity is the main application. In order for the project to be compatible with virtual reality headsets, downloading packages and tool kits are necessary to have the game running properly. In the XR toolkit, inside toolkit, one needs to enable XR windows and Oculus compatibility in order to have a three-dimensional environment. In this case, the Android extension has to be downloaded from Unity's built-in modules. The reason for using an Android extension in this project is for the Meta Quest 2, which is classified as an Android in Unity. Barracuda is another package necessary in order for the ONNX file to work in this application. Once the necessary applications are downloaded properly, some settings need to be changed and modified for the application to run fine in the Meta Quest.

3.1 Convolutional Neural Network Scene

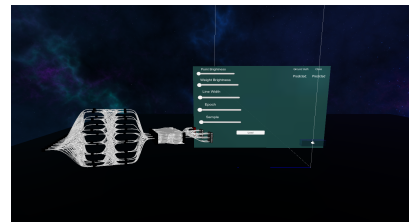


Figure 1: Demo of CNN visuals

Developing the CNN scene in Unity was built with the help of Stefsietz, who, with the help of his documentation and code, I was able to replicate and modify to fit the current version of Unity compared to the version of Unity Stefsietz used. Once the assets and scripts were debugged and re-scripted to fit in the current version of Unity, developing the environment was next. Using an action-based and a VR camera controller was needed to have the project running. At the start of the level, the model shown is the base model without any inputs, weights, or desired outputs. On the right of the player, the user interface elements will be placed that are responsive to the controllers. The user interface elements have a separate tab where

