An Immersive Node-Link Visualization of Artificial Neural Networks for Machine Learning Experts

Martin Bellgardt
Visual Computing Institute
RWTH Aachen University
Germany

Christian Scheiderer
Institute for Information Management
in Mechanical Engineering
RWTH Aachen University, Germany

Torsten W. Kuhlen
Visual Computing Institute
RWTH Aachen University
Germany

***Abstract*—The black box problem of artificial neural networks (ANNs) is still a very relevant issue. When communicating basic concepts of ANNs, they are often depicted as node-link diagrams. Despite this being a straight forward way to visualize them, it is rarely used outside an educational context. However, we hypothesize that large-scale node-link diagrams of full ANNs could be useful even to machine learning experts. Hence, we present a visualization tool that depicts convolutional ANNs as node-link diagrams using immersive virtual reality. We applied our tool to a use-case in the field of machine learning research and adapted it to the specific challenges. Finally, we performed an expert review to evaluate the usefulness of our visualization. We found that our node-link visualization of ANNs was perceived as helpful in this professional context.**

***Index Terms*—Artificial Neural Networks, Immersive Analytics, Node-Link Diagram**

I. Introduction

During the past decade, ANNs have become a standard tool within the larger topic of artificial intelligence (AI). They form an integral part of many state-of-the-art machine learning approaches in the areas of, e.g., image classification, object detection or semantic image segmentation. As these approaches rapidly advance, our understanding of their means to achieve their outstanding results stagnates; a circumstance, commonly called *the black box problem* [3]. However, the behavior and especially flaws of ANNs have to be properly explained for them to be acceptable in safety-critical applications. As a result, the field of explainable artificial intelligence (XAI) has recently gained traction [2], [4], with many possible directions to address the black box problem. We refer the reader to the survey on methods for explaining black box models presented in [5] for an extensive overview.

A common way to depict neural networks are node-link diagrams [7]. In a node-link diagram, the core concepts of neurons and synapses are clearly depicted as nodes and edges. This makes the representation close to the physical appearance of ANNs biological counterparts. Despite this, advanced visualizations for complex ANNs rarely make use of node-link diagrams. Instead, they are mainly used in literature to illustrate concepts on small scale examples, like, e.g., Figure 11 in [11] or many Figures in [12]. There are also interactive visualizations making use of node-link diagrams, such as the tensorflow playground [14]. Their interactive website allows users to set the hyperparameters of a small neural network and observe their behavior on a few two dimensional toy datasets. While it is a powerful tool to teach the basics of ANNs to beginners, the networks depicted are just as small and exemplary. To the best of our knowledge, it has not been attempted before to depict a large ANN, that is subject to active research, as a node-link diagram in this way.

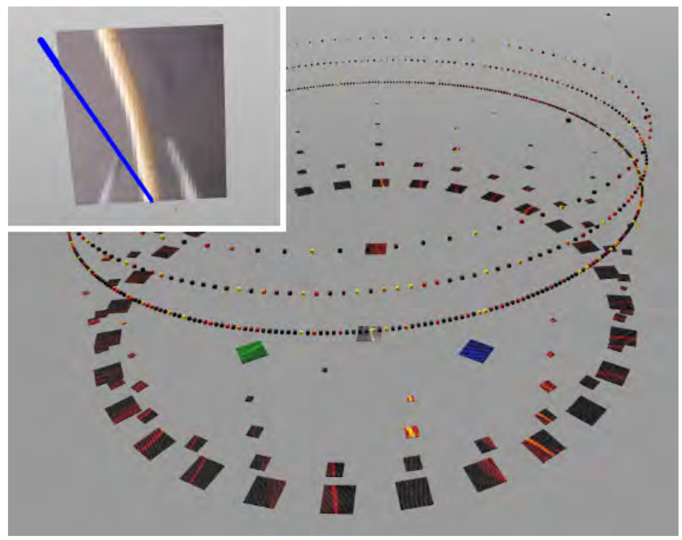

Fig. 1. An overview of our visualization without anything selected. The layers of the ANN are represented by circles that are stacked on top of each other.

Another way node-link diagrams are used to depict ANNs is in computation graphs, as described, e.g., in [8]. In a computation graph, the nodes do not depict individual neurons but computations and the edges depict data flowing between these computations. The data, in this case, is usually high-dimensional tensors which correspond to activation values, weights, biases, gradients, etc. For this reason, these node-link diagrams do not represent the networks themselves, but the computations involved. Although this kind of node-link diagram has been popularized for large, productively used ANNs by the TensorBoard visualization [15], it does not convey the scale of the network and its topology very well.

One may not forget to mention that there is one notable related work that comes very close to our vision. The web-based visualization described in [6] depicts a convolutional ANN trained on the MNIST dataset of hand written digits as a three-dimensional node-link diagram. Users are able to feed their own digits into the network and immediately observe the responses of any neuron as their color is mapped to their activation value. The weights of all synapses are also depicted using colored lines and the learned filter kernels can be displayed. While their approach depicts a working, full-scale ANN, the authors state that it is targeted at an educational use case. In this work, we aim to take their idea one step further and evaluate if a similar depiction of ANNs can also be useful to machine learning experts.

One major challenge, and probably the reason why large scale node-link visualizations have not been attempted more often, is the increasing size and complexity of ANNs. Modern ANNs contain thousands of neurons and millions of synapses. Naively drawing a network of that size on a two-dimensional plane would result in a lot of visual clutter and be far from interpretable. A three-dimensional visualization as described in [6] alleviates the problem but still suffers the same issues if displayed on a two-dimensional screen. This is why, for our visualization, we decided to make use of immersive virtual reality (VR) technology. Due to recent economical advancements, VR hardware has become much cheaper and is now feasible for use in everyday work [1]. This is handy for our task, since the ability to directly display our visualization in three-dimensions lets us make better use of all three spatial dimensions. In addition to that, there have been strong hints that higher degrees of immersion lead to better performance in some data analysis tasks [9], [13].

To show that our system can be useful to machine learning experts, using example datasets like MNIST is not enough. Instead, we need to apply our system to a use case that is subject to active research. In the field of production technologies, machine learning is currently an area of increasing interest. Therefore, we decided to apply our visualization to a use case in that area. The following section describes this use case in more detail.

II. Use Case

We demonstrate our visualization approach on a convolutional ANN that controls a robot arm. This section will give a short overview of the project. A more detailed description is presented in [10].

In essence, the project demonstrates how the movements of a robotic arm can be generated via reinforcement learning. For this, the robot is given the task of playing the classic wire loop game, where a fork, carried by the robot, has to be moved along a path defined by a bendable wire without touching it. Touching the wire results in an electrical circuit being closed, signaling that the robot has made a mistake. Based on this information and the images of a camera that is mounted to the robot's head, a reinforcement learning approach trains a convolutional ANN. This ANN uses (downsampled) camera images as inputs and generates robot trajectories as outputs. To put it simply, the process works as follows: First, an image is taken of the next part of the wire. Then, the ANN calculates a trajectory based on this image. Finally, the robot moves along the trajectory and the process repeats. If the robot touches the wire, it has to reset its position and receives a negative reward during the training process. The details of this training process are not relevant to this work and are thus omitted.

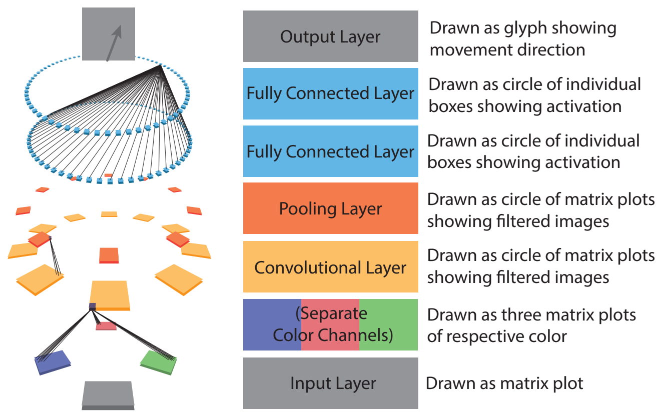

Fig. 2. A schematic depiction of our visualization.

The project has already demonstrated that it is possible to learn a control-policy for a robotic arm to play the wire loop game, solely based on camera images. Nevertheless, there are still many ways to improve this process that are actively being worked on. This makes this project an ideal use case for our visualization, since the experts working on it can likely benefit from a better understanding of their ANNs. Additionally, the system uses images as inputs and a convolutional ANN, similar to [6] which inspired this work. However, there are a few major differences. The network controlling the robot is a lot larger and more complicated. Furthermore, it uses color images as inputs and it outputs continuous values that encode a trajectory. Finally, it is not updated based on user input but every step the robot takes, which is about once every second. These aspects pose additional challenges, which had to be addressed to make the visualization applicable to this use case. The way we designed it around these aspects is described in the next section.

III. Concept

An overview of our visualization can be seen in Figure 1. As described earlier, it is greatly inspired by the system described in [6]. Just like them, we visualize the ANN as a node-link diagram where the nodes represent individual activations and the edges represent the weights between them. Likewise, we show only those edges connecting to the node that is selected, to avoid visual clutter (see Figure 3). However, due to the larger networks from our use case, using a linear layout and showing the visualization on a 2D screen, as it is done in [6], is not feasible. This is why we decided to use immersive VR, since it allows us to make better use of the third dimension. The decisions made regarding our VR implementation are described in Section III-A. Section III-B explains the spatial layout of our visualization, especially regarding the additional freedom gained from VR. Finally, Section III-C describes additional aspects that needed to be addressed in our application to adapt it to our use case.

A. Immersive Virtual Reality

Our goal was to design an application that machine learning expert could integrate into their workflow. The first decision, when designing an immersive VR application for use in everyday work, should be the usage scenario [1]. ANNs are abstract objects that do not exist in the physical world, so the user's sense of their own body will likely not effect their immersion. Hence, a seated scenario would be feasible for this application. However, the experts in our use case usually work standing in front of the robot and walk around frequently. The location of the robot inside their workshop even allows for a tracking space that is large enough to walk around in. Therefore, we decided to design our application to be used while standing and allow the users to walk around if they wish. Additionally, we included a virtual travel technique (flying) to give the users more freedom to position themselves.

Naturally, since we wanted the availability of our visualization as high as possible, we decided to use consumer grade head mounted displays (HMDs) as our target display devices. These devices are cheap, easy to use and don't require a complicated setup. To enable virtual travel and all other interactive elements of the visualization, we can use the standard spatial input devices that are included with the HMD. Our application is built on the OpenVR framework, so a variety of consumer HMDs can be easily supported. We designed our application mainly around Oculus Rift and HTC Vive HMDs and respective controllers. Using different kinds of controllers is possible but might require a redesign of the control scheme, since the button layout on VR controllers is not yet standardized.

B. Spatial Layout

The baseline approach for the spatial layout of the nodes in our visualization is the linear approach used in [6]. In this case, the third spatial dimension is only used to group activations of the same convolutional neuron, while all neurons are placed along a straight line. This layout works well for smaller networks and is especially useful when the target platform supports only 2D display and interaction. Another positive aspect of this layout is the way users can zoom in to easily view any part of the network in isolation. For our networks, however, this layout results in a very long line, where most of the information is distributed along one spatial dimension. This is why, we needed to find a layout that is more compact but doesn't introduce too much visual clutter.

We wanted to preserve the idea of encoding the layer structure of the networks along one spatial dimension. This leaves us with two spatial dimensions to layout the neurons. The neurons in the hidden layers of an ANN do not have any ordering or structure that would be implied from the calculations. Hence, there is a lot of freedom for their placement. We considered placing them on a regular grid, since this would result in a very compact layout. Unfortunately, this caused many of them to be occluded. In the end, we settled on arranging the neurons in circles. This makes the visualization compact enough to easily fit the users' field of view. Still, it preserves the good properties of the linear layout, as users can look at parts of the network from within the circle to view them in isolation without clutter. To decrease the amount of movement the users have to do and give them the ability to compare neurons of adjacent layers, the neurons can also be rotated along the circle. Additionally, the fact that there is no inherent ordering between the neurons is visually communicated better than in the linear layout. Overall, we see this layout, shown schematically in Figure 2, as a good compromise between compactness, clutter and simplicity. However, we do not claim that it is the best possible layout, as it makes little use of the additional dimension. Other layouts might use the spatial arrangement of neurons to encode properies of them or their relationship. We plan to investigate this in future work.

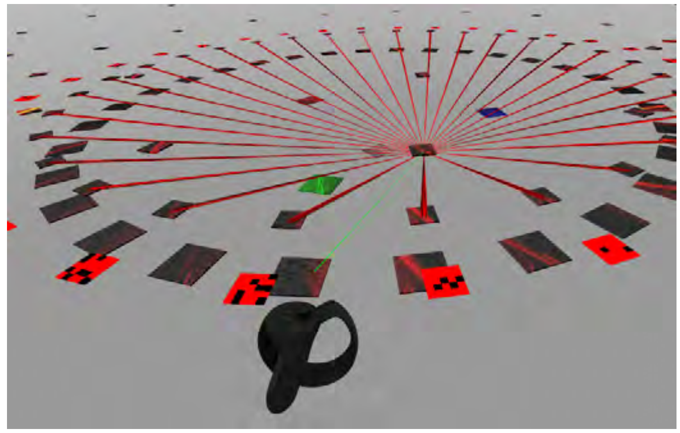

Fig. 3. Pointing at any activation shows all edges that connect to it. For convolutional layers, the filter kernels are shown as well.

C. Use Case Specific Features

There were a few other challenges we had to address, to make our visualization feasible to be used by the experts in our use case. First of all, the output of the network controlling the robot is not nearly as interpretable as the output of the MNIST network in [6]. Instead of encoding class probabilities, the output layer of our networks encodes a path that the robot will follow. Hence, we added a glyph to our visualization that contains the input image and plots the path that the robot will take as a blue line. An example of this can be seen at the top of Figure 1. The experts wished to be notified when the robot touched the wire, so we made the background of our visualization flash red for a short time, if a respective network package is received. Additionally, we added a function to save and load the state of the visualization. We did this because switching from and to VR frequently can cause accidental inputs which might cause the user to fly far away from the visualization or change the parameters in an undesired fashion.

IV. Expert Review

To test our hypothesis that a node-link visualization of a complete network is helpful to machine learning experts, we performed an expert review. Overall, we interviewed four experts. Two of them were actively working in our use case and had also worked with us and our visualization before. The other two had heard of the project but were not directly involved and had never seen the visualization.

Every person we interviewed started with an exhaustive tour of our visualization and an explanation of all its features. After that, they were given the opportunity to look around freely as long as they wanted, while being told to think aloud as much as possible. When they were done, we conducted a structured interview with them to gather qualitative feedback.

All four experts agreed that our visualization is helpful. One even stated that they developed multiple visualizations of their own before but that our visualization is more helpful than all of them. We also asked the participants which part of their work could benefit from our visualization and received a variety of answers. Two of the experts stated that our visualization could help them find out if more layers or neurons are needed in the network. While they did not elaborate on how exactly they would do this, we suspect that this is because they get a more intuitive understanding of the quantities controlled by hyperparameters. Since hyperparameters are usually only defined numerically, actually seeing the scale of the resulting network might help the experts to determine where they are too high or too low. One expert even stated explicitly that the visualization is helpful to get a feeling for the size of the network. Additionally the experts mentioned they could determine similar feature maps, which filters were important, which neurons got more updates, which neurons were activated more frequently or generally if the patterns learned by the network made sense to them. One of the experts who was working on the use case even spotted a potential error in their implementation using the visualization. This happened when they were looking at a pooling layer and noticed that it appeared very dark in comparison to the convolutional layer below. Since their pooling was supposed to work using the maximum operation, directly comparing the values showed that they did not make sense which hinted at a potential implementation error.

We also asked the experts if they would like to see more networks visualized in a similar fashion. Two participants answered that they would definitely like to see even more complicated networks like, e.g., recurrent ANNs. One said they would like to try it but was unsure if it would provide a benefit. The last expert said that the tool is useful for scientific purposes but in practical settings they would not have the time to analyze the network in that much detail.

We also received criticism and suggestions from the experts. For example, all but one expert suggested to show only the most important edges instead of all of them. The same experts also requested a way to keep one node selected without pointing at it, to look at the edges from different directions. Two even suggested that they wanted to follow the edges through multiple layers, like a tree. Another point that two experts mentioned was that the fully-connected layers were too complicated to really gain information from them. Other points of criticism we received were that it was too difficult to turn the individual layers, that the flashing when the wire was touched felt too aggressive and that the selection ray is sometimes difficult to see in front of a red background.

V. Conclusion

We have presented an interactive node-link visualization for ANNs based on immersive virtual reality. Unlike previous work, our system is targeted explicitly at machine learning experts. We then performed an expert review to test our hypothesis that a node-link visualization can help them in their everyday work. We perceived the feedback from the expert review as generally positive and see it as a strong hint that our hypothesis holds. Obviously more evaluation needs to be performed to show that node-link visualizations can actually improve the workflow of machine learning experts. However, we are optimistic that a future revision of our visualization could be actively used by these experts on a regular basis.

VI. Acknowledgments

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2023 Internet of Production – 390621612.

References

[1] M. Bellgardt, S. Pick, D. Zielasko, T. Vierjahn, B. Weyers, and T. W. Kuhlen. Utilizing Immersive Virtual Reality in Everyday Work. In *IEEE Workshop on Everyday Virtual Reality, WEVR*, 2017.

[2] M. Berrada. Peeking Inside the Black-Box : A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access*, 6:52138–52160, 2018.

[3] D. Castelvecchi. Can we open the black box of AI? *Nature News*, 538(7623):20–23, 2016.

[4] D. Doran, S. Schulz, and T. R. Besold. What does explainable AI really mean? A new conceptualization of perspectives. *CoRR*, abs/1710.0, 2017.

[5] R. Guidotti, A. Monreale, F. Turini, D. Pedreschi, and F. Giannotti. A Survey of Methods for Explaining Black Box Models. *CoRR*, abs/1802.0, 2018.

[6] A. W. Harley. An Interactive Node-Link Visualization of Convolutional Neural Networks. In *International Symposium on Visual Computing*, pages 867–877, 2015.

[7] F. Hohman, M. Kahng, R. Pienta, and D. H. Chau. Visual Analytics in Deep Learning: An Interrogative Survey for the Next Frontiers. *CoRR*, abs/1801.0, 2018.

[8] M. Kahng, P. Y. Andrews, A. Kalro, and H. D. Chau. ActiVis: Visual Exploration of Industry-Scale Deep Neural Network Models. *CoRR*, abs/1704.0, 2017.

[9] B. Laha, K. Sensharma, J. D. Schiffbauer, and D. A. Bowman. Effects of Immersion on Visual Analysis of Volume Data. *TVCG*, 18(4):597–606, 2012.

[10] R. Meyes, C. Scheiderer, and T. Meisen. Continuous Motion Planning for Industrial Robots based on Direct Sensory Input. *Procedia CIRP*, 72:291–296, 2018.

[11] G. Montavon, W. Samek, and K. R. Müller. Methods for interpreting and understanding deep neural networks. *Digital Signal Processing: A Review Journal*, 73:1–15, 2018.

[12] C. Olah and S. Carter. Attention and Augmented Recurrent Neural Networks. *Distill*, 2016.

[13] D. Raja, D. A. Bowman, J. Lucas, and C. North. Exploring the Benefits of Immersion in Abstract Information Visualization. In *Proc. of the Immersive Projection Technology Workshop*, pages 61–69, 2004.

[14] D. Smilkov, S. Carter, D. Sculley, F. B. Viégas, and M. Wattenberg. Direct-Manipulation Visualization of Deep Networks. *CoRR*, abs/1708.0, 2017.

[15] K. Wongsuphasawat, D. Smilkov, J. Wexler, J. Wilson, D. Mané, D. Fritz, D. Krishnan, F. B. Viégas, and M. Wattenberg. Visualizing Dataflow Graphs of Deep Learning Models in TensorFlow. *TVCG*, 24, 2017.