# Challenges in Interpretability of Neural Networks for Eye Movement Data

Ayush Kumar
Stony Brook University
aykumar@cs.stonybrook.edu

Prantik Howlader
Stony Brook University
phowlader@cs.stonybrook.edu

Rafael Garcia
University of Stuttgart
Rafael.Garcia@visus.uni-stuttgart.de

Daniel Weiskopf
University of Stuttgart
Daniel.Weiskopf@visus.uni-stuttgart.de

Klaus Mueller
Stony Brook University
mueller@cs.stonybrook.edu

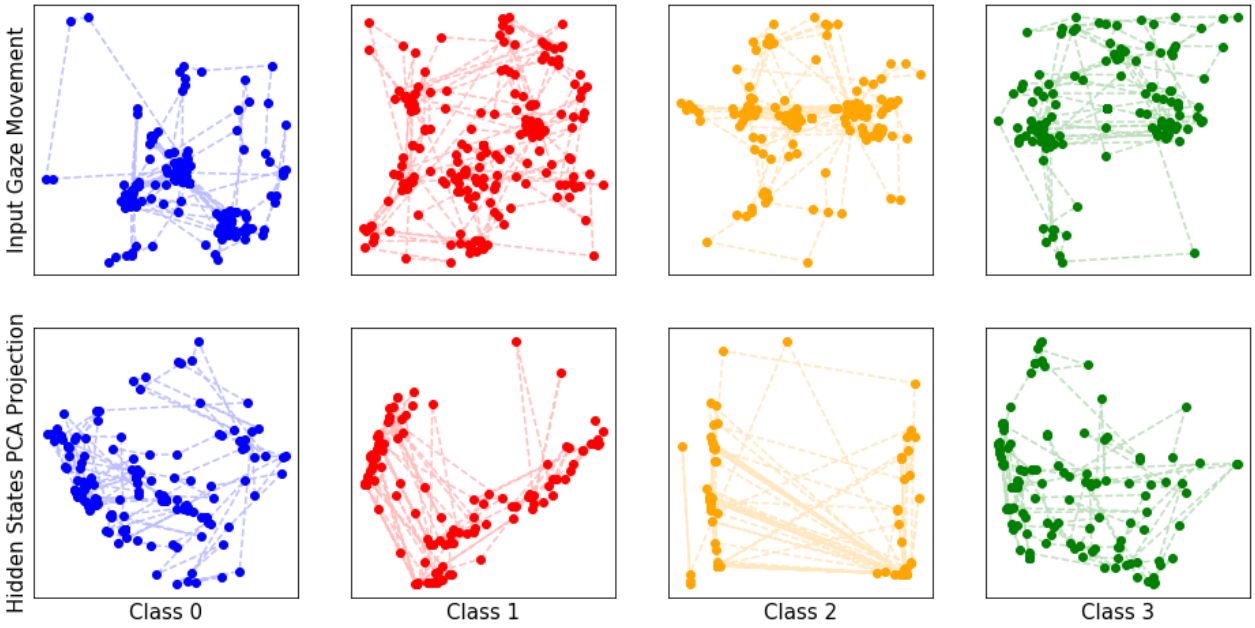


**Figure 1: Gaze plots and neural network visualization to illustrate interpretability challenges for machine learning. The four columns show example sequences of four different classes of tasks performed by participants. Top: Plot of x and y coordinates of scanpaths (sequences of fixations). Bottom: PCA projection of the hidden states produced by an LSTM layer for these sequences. While we can qualitatively see differences both in the gaze plots and LSTM visualizations, it is still challenging to fully understand the difference between the machine learning models.**

## ABSTRACT

Many applications in eye tracking have been increasingly employing neural networks to solve machine learning tasks. In general, neural networks have achieved impressive results in many problems over the past few years, but they still suffer from the lack of interpretability due to their black-box behavior. While previous research on explainable AI has been able to provide high levels of interpretability for models in image classification and natural language processing tasks, little effort has been put into interpreting and understanding networks trained with eye movement datasets. This paper discusses the importance of developing interpretability methods specifically for these models. We characterize the main problems for interpreting neural networks with this type of data, how they differ from the problems faced in other domains, and why existing techniques are not sufficient to address all of these issues. We present preliminary experiments showing the limitations that current techniques have and how we can improve upon them. Finally, based on the evaluation of our experiments, we suggest


*ETRA '20 Short Papers, June 2–5, 2020, Stuttgart, Germany*

ACM ISBN 978-1-4503-7134-6/20/06…$15.00
https://doi.org/10.1145/3379156.3391361

future research directions that might lead to more interpretable and explainable neural networks for eye tracking.

## CCS CONCEPTS

• **Human-centered computing** → *Visualization design and evaluation methods*; *Visual analytics*; *Visualization techniques.*

## KEYWORDS

Eye tracking, visualization, deep learning, explainable AI

**ACM Reference Format:**
Ayush Kumar, Prantik Howlader, Rafael Garcia, Daniel Weiskopf, and Klaus Mueller. 2020. Challenges in Interpretability of Neural Networks for Eye Movement Data. In *Symposium on Eye Tracking Research and Applications (ETRA '20 Short Papers), June 2–5, 2020, Stuttgart, Germany*. ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3379156.3391361

## 1 INTRODUCTION

Deep neural networks are one of the most effective ways to achieve good performances on difficult pattern recognition tasks [LeCun et al. 2015]. Over the past few years, they have been used in a vast number of applications, such as computer vision and natural language processing (NLP) [LeCun et al. 2015]. It also proved to be an efficient way to perform machine learning (ML) for eye tracking, such as event detection [Zemblys et al. 2019, 2018], classification of eye movement data [Dalrymple et al. 2019; Komogortsev and Karpov 2013; Kumar et al. 2019; Startsev et al. 2019; Tafaj et al. 2013], encoding of gaze data [Fuhl et al. 2019], and pupil detection [Fuhl et al. 2017].

However, despite their outstanding performance, neural networks are essentially black-box models, which impairs the user's ability to interpret the model and understand the reasoning behind the predictions. This lack of interpretability leads to trust issues—where the user is not confident about whether the reasoning learned by the model makes sense or not—and makes it much more difficult to identify and fix problems in the development of the model. When not properly addressed, such issues can severely hinder the usability of the model.

To address this problem, the ML community has focused on building visualization tools to provide more interpretability to neural networks, giving insights into the model's learned features and reasoning process [Choo and Liu 2018; Garcia et al. 2018; Liu et al. 2017]. However, most approaches are restricted to models trained for a specific application, such as computer vision or NLP. None of the existing contributions can, to the best of our knowledge, address the specific issues faced when modeling eye movement datasets.

Existing interpretability techniques resort in the analysis of the input data and how the input affects the features and output produced by the model. However, eye movement datasets are essentially spatio-temporal and often contain attributes from multiple data domains. For this reason, they are not straightforward to interpret when compared to image or text datasets. Consequently, techniques developed with the latter kind of datasets in mind cannot be directly applied to eye tracking models.

In this paper, we address this problem by discussing the characteristics of eye movement datasets, how neural networks transform their features to perform classification, which interpretability issues appear in the analysis of such models, and how visualization techniques can open these black boxes to improve interpretability for the field of eye tracking.

We also present an example case that illustrates the potential and challenges of intepretable ML for eye tracking. Figure 1 shows a screenshot from this example.

## 2 BACKGROUND AND RELATED WORK

One of the first approaches to achieve interpretability was to measure and visualize the importance each input feature has in the prediction output. This approach was successfully employed in models trained for image classification [Kahng et al. 2017; Yosinski et al. 2015] and natural language processing [Strobelt et al. 2017]. However, while effective in giving insights into which features were used in the classification process, it does not allow the analyst to visualize the model's reasoning process and thus answer questions such as whether the model is correctly learning distinguishable features or how following layers use their input data to build more abstract features.

Such problems can be addressed by exploring internal feature vectors and hidden states produced by the network. These internal representations contain all the information extracted from the original input by the model, and thus, they have to contain information meaningful to the prediction task. Rauber et al. [2016] address this problem by employing dimensionality reduction techniques to visualize the separability of feature vectors in different hidden layers of a neural network. By doing so, the user can find out whether the model learned distinguishable features for each class.

LSTMVis [Strobelt et al. 2017] employs temporal visualization to identify sequences of input producing similar hidden states, thus giving insights into how recurrent networks learn to represent sentences with similar meaning in NLP. In a related approach, Giurgiu and Schumann [2019] explain the prediction of RNN classifiers by returning the key events that significantly modified the model's hidden states. A more comprehensive list of papers on the topic can be found in survey articles by Garcia et al. [2018] and Hohman et al. [2018].

Although such contributions can handle many interpretability problems, they do not address particular issues found in eye tracking tasks. Due to the temporal nature of eye movement data, many of the visualization techniques employed in NLP can be generalized to handle eye tracking to some extent. However, there are important differences as well. First, eye movement data also has a spatial component. Second, text-based inputs do not have any kind of time span between the elements of the input sequence. Eye movement data are time series in which the time between two events in the sequence carries important information that may be a required feature for the classification task.

To the best of our knowledge, no previous work addresses issues involving such spatio-temporal—and particularly eye movement—data. The paper closest to our discussion is by Giurgiu and Schumann [2019]. They explain the predictions of a long short-term memory (LSTM) model by converting time series into events based on a threshold and identifying which events led to the prediction output. However, the event-based analysis employed by them and

the lack of spatial component creates a limitation on how it can support models trained with eye movement datasets.

## 3 PROBLEM CHARACTERIZATION

Interpretability of neural networks is by itself a recent and highly relevant topic, with most papers dating back from the last six years [Garcia et al. 2018]. In this section, we characterize the specificities of interpretable neural networks for eye tracking.

### 3.1 Data Model

We assume that the eye tracking data can be described as a temporal sequence of data tuples $d_i$, where $i$ describes the temporal index and $d_i$ is a vector of data attributes. Typically, the data attributes contain at least $x$ and $y$ coordinates of gaze. However, there might be other attributes like pupil diameter or information about the stimulus (background image or video) around the gaze position, or many more. Furthermore, the temporal index $i$ may correspond to a regular, equidistant sampling of time (for example, in the form of the typical output of an eye tracker), but it could also relate to an ordered list of events (for example, a sequence of fixations produced from applying a fixation filter to raw gaze data). Therefore, our data model applies to a wide range of eye tracking data, including the raw data from eye tracking devices all the way to processed scanpaths. However, we do not consider typical image analysis problems that, for example, play a critical role in gaze estimation from video [Zhang et al. 2019].

Therefore, we have to address datasets that are spatio-temporal (to include gaze positions and temporal sequence information) along with potentially further data attributes.

### 3.2 Relationship to Other Domains

Therefore, existing approaches to interpretability of neural networks in computer vision and NLP do not necessarily carry over to eye tracking without adaptation. However, there are some important commonalities shared with NLP. Machine learning for eye tracking typically uses recurrent networks; for example, LSTMs [Hochreiter and Schmidhuber 1997] are a popular choice. These network are particularly suited to handle temporal datasets, as the network's layers maintain an internal memory—called hidden state—that are updated every time a new input element is fed into the network. NLP also employs this type of neural networks, which makes visualization tools such as LSTMVis [Strobelt et al. 2017]—which aims to visualize patterns in the hidden state sequences that can explain the reasoning learned by the model—a good starting point to build effective tools for eye tracking tasks.

However, in contrast to text-based datasets, a sequence of eye movement data points contains a temporal component in between two data points that, depending on the application, might be important for the classification tasks. For instance, two data points located in the same spatial position may represent different meaning if the time passed from the previous gazing position is different. Text datasets do not have such characteristics, as there is no temporal relationship between the words—only their order matters. Existing tools for interpretability of recurrent networks do not address this problem and, thus, are not fully suited to eye tracking.

Therefore, some aspects of interpretability for computer vision networks have to be considered too. For instance, in image classification, a single pixel value only has information value when aggregated with the values of neighboring pixels. Such a behavior also appears in eye movement datasets, e.g., an individual gaze location may only produce useful information for the classification tasks when combined with subsequent gaze locations.

Many existing tools in computer vision use heatmaps to display which pixels in the input image impacted prediction the most, and thus give better insights into what features the model is looking for to classify the input as belonging to a particular class. A similar approach can be used in eye tracking models to identify which gazing points impacted classification the most. However, it does not answer questions such as if that gaze point is important for the classification due to its spatial location, or to the temporal aspect of it, or some other component of data which could be combination of both spatial and temporal.

Interpretability for neural networks for eye movement data will likely draw from existing work in NLP and computer vision but will also have to merge currently separated approaches from those areas and include further adaptations specific to eye tracking. The next section describes an example case that targets interpretability for a typical eye movement problem, where some existing approaches are adopted.

## 4 EXAMPLE CASE

To give an example on how interpretability is important for the training of neural networks employed in eye tracking applications, we developed an experiment using an LSTM trained with the dataset used by Greene et al. [2012]. This dataset comprises a classification task with four classes. Each class represents the action performed by the person at the moment of the eye movement sequence capture: $Class1$ represents the action of determining the *Decade* in which an image was taken; $Class2$ represents the action of *memorizing* a picture; $Class3$ represents the action of determining how well you know the *people* on a picture; and $Class4$ represents the action of determining the *wealth* of the people on a picture.

To model the task, we developed a small LSTM model with a single hidden layer. We chose this architecture because our goal is to visualize how the model learns to represent the features from the input data in its hidden state. Since the hidden states learned by the model represent the features of the input sequences, these hidden states are supposed to contain significant information for the model's decision making. Visualizing hidden states that are high-dimensional in nature can be challenging, as humans can only imagine up to three dimensions with ease. To ease the understanding of hidden states, we project high-dimensional states into a lower-dimensional representation while preserving its underlying structure as much as possible using principal component analysis (PCA) [Wold et al. 1987].

Figure 1 displays the gaze sequence of four inputs (top)—one for each class—and the PCA projection of the hidden state sequence produced by the model for the respective input (bottom). A similar approach is used to analyze convolutional networks in work by Rauber et al. [2016] and Kahng et al. [2017]. However, it is not easy to distinguish what is learnt and how it is learnt, mostly because

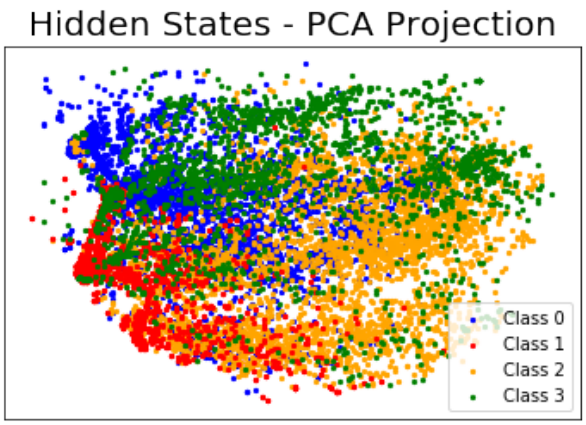


**Figure 2: PCA projection of the hidden states produced by the network's LSTM layer for all inputs in the test set. Colored by the predicted class.**

such an approach is unable to give all the insights needed to build hypotheses regarding spatio-temporal datasets.

The insights above are corroborated by Figure 2. Here, we adapt the strategy used by Rauber et al. [2016] to LSTMs. For each test set sequence, we extract the hidden states produced by the network at every time step. We then join all hidden state vectors in a single dataset and project them to a 2D space using PCA. Data points are colored by the model's final output, i.e., the label produced by the model if that hidden state were the last one produced by the sequence. Ideally, we would expect to see the data points of each class converging toward distinguishable regions, as this would mean that the model learned to create different representations for each class. This is not the case for this model, as there is a significant overlap between elements of different classes. This is expected, as the model only reaches a 75% accuracy on the test set. Nonetheless, it showcases how such visualizations can be useful to identify issues in the modeling process and guide the user in making possible changes.

The hidden states that are learnt for a classification model are expected to be distinguishable in the feature space, on the basis of their classes. But in most of the classification models, there is the possibility of interactivity among the features learnt for the different classes [Wold et al. 1987]. Explainable system can be used for greater understanding of the accuracy of the deep network, based on the events that caused a specific classification. It can also be used to see any overlap in the events, between the different classes, this can augur a new set of research on the intersecting events and also class distinctive events.

## 5 RESEARCH CHALLENGES

Based on literature on interpretable machine learning in NLP and computer vision, and our previous experience with the analysis of eye movement data, and own experiments with interpretability for concrete examples from eye tracking, we have identified challenges that can be better solved using explainability.

*Explain the correlation between the data points (with their features) and the hidden states or feature vectors produced by the network at their hidden layers.* Interpretability tools should focus on understanding the dynamics of the hidden states, allowing the user to inspect how these vectors learn to abstract the input's features.

*Explainability beyond events.* Recurrent networks often interpret temporal data as events. However, in eye movement datasets, there is not always a clear definition of what an event is. For instance, if we define attention on a gaze point in spatial aspect as an event, we may end up losing the significance of the order in which the point was achieved as well as the temporal aspect associated with it.

*Interactive exploration.* Complex neural networks often have dozens of hidden layers and are trained with datasets having millions of data points. To interpret such a model, existing interaction approaches need to be extended to allow for better scalability with network complexity.

Apart from generic challenges associated with interpretability, we also found that there are challenges associated with eye tracking dataset too in general.

*Interpret which data point components (spatial, temporal, etc.) the model is taking into account.* Tools aiming to explain what features impacted classification should be able to effectively measure which components (spatial, temporal or both) of the data influenced the prediction and to what extent they did so.

*Performance analysis for spatio-temporal data.* One of the main issues when training neural networks is to identify the reasons why the training process did not achieved the expected performance. This is even harder when dealing with spatio-temporal data as recurrent networks are very prone to problems such as vanishing gradient.

*Annotation support.* Eye movement data often lacks sufficient labeled data, which is the prerequisite to exploit deep learning. Therefore, there is need for visual and interactive tools that allow users to quickly annotate data for learning [Kumar et al. 2020].

## 6 CONCLUSION

Both eye tracking and machine learning are growing research fields that are constantly opening the doors to novel and interesting topics. As neural networks become more and more popular in both fields, it is important to invest time into finding new ways to interpret and analyze these models. That said, we argue for an increase in the collaboration between the eye tracking, machine learning, and visualization community in order to develop more tools able to address the issues raised in this paper. With such collaboration, advances in eye tracking applications involving complex machine learning tasks will occur more often, and will spread faster both to scientific and industrial applications.

## ACKNOWLEDGMENTS

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 251654672 – TRR 161 (Project B01), SUNY Korea's ICTCCP (IITP-2020-2011-1-00783) supervised by the IITP and NSF grant IIS 1527200.

# REFERENCES


Jaegul Choo and Shixia Liu. 2018. Visual analytics for explainable deep learning. *IEEE Computer Graphics and Applications* 38, 4 (2018), 84–92.

Kirsten A Dalrymple, Ming Jiang, Qi Zhao, and Jed T Elison. 2019. Machine learning accurately classifies age of toddlers based on eye tracking. *Scientific Reports* 9 (2019), 1–10.

Wolfgang Fuhl, Efe Bozkir, Benedikt Hosp, Nora Castner, David Geisler, Thiago C Santini, and Enkelejda Kasneci. 2019. Encodji: encoding gaze data into emoji space for an amusing scanpath classification approach. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*. Article 64, 4 pages.

Wolfgang Fuhl, Thiago Santini, Gjergji Kasneci, Wolfgang Rosenstiel, and Enkelejda Kasneci. 2017. PupilNet v2.0: Convolutional neural networks for CPU based real time robust pupil detection. *arXiv preprint 1711.00112* (2017).

Rafael Garcia, Alexandru C Telea, Bruno Castro da Silva, Jim Tørresen, and João Luiz Dihl Comba. 2018. A task-and-technique centered survey on visual analytics for deep learning model engineering. *Computers & Graphics* 77 (2018), 30–49.

Ioana Giurgiu and Anika Schumann. 2019. Explainable failure predictions with RNN classifiers based on time series data. *arXiv preprint 1901.08554* (2019).

Michelle R Greene, Tommy Liu, and Jeremy M Wolfe. 2012. Reconsidering Yarbus: A failure to predict observers' task from eye movement patterns. *Vision Research* 62 (2012), 1–8.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9 (1997), 1735–1780.

Fred Hohman, Minsuk Kahng, Robert Pienta, and Duen Horng Chau. 2018. Visual analytics in deep learning: An interrogative survey for the next frontiers. *IEEE Transactions on Visualization and Computer Graphics* 25, 8 (2018), 2674–2693.

Minsuk Kahng, Pierre Y Andrews, Aditya Kalro, and Duen Horng Polo Chau. 2017. ActiVis: Visual exploration of industry-scale deep neural network models. *IEEE Transactions on Visualization and Computer Graphics* 24, 1 (2017), 88–97.

Oleg V Komogortsev and Alex Karpov. 2013. Automated classification and scoring of smooth pursuit eye movements in the presence of fixations and saccades. *Behavior Research Methods* 45 (2013), 203–215.

Ayush Kumar, Debesh Mohanty, Kuno Kurzhals, Fabian Beck, Daniel Weiskopf, and Klaus Mueller. 2020. Demo of the EyeSAC System for Visual Synchronization, Cleaning, and Annotation of Eye Movement Data. In *Proceedings of the 12th ACM Symposium on Eye Tracking Research & Applications (ETRA '20 Adjunct)*. 3.

Ayush Kumar, Anjul Tyagi, Michael Burch, Daniel Weiskopf, and Klaus Mueller. 2019. Task classification model for visual fixation, exploration, and search. In *Proceedings of the 11th ACM Symposium on Eye Tracking Research & Applications*. Article 65, 4 pages.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444.

Shixia Liu, Xiting Wang, Mengchen Liu, and Jun Zhu. 2017. Towards better analysis of machine learning models: A visual analytics perspective. *Visual Informatics* 1, 1 (2017), 48–56.

Paulo E Rauber, Samuel G Fadel, Alexandre X Falcao, and Alexandru C Telea. 2016. Visualizing the hidden activity of artificial neural networks. *IEEE Transactions on Visualization and Computer Graphics* 23 (2016), 101–110.

Mikhail Startsev, Ioannis Agtzidis, and Michael Dorr. 2019. 1D CNN with BLSTM for automated classification of fixations, saccades, and smooth pursuits. *Behavior Research Methods* 51 (2019), 556–572.

Hendrik Strobelt, Sebastian Gehrmann, Hanspeter Pfister, and Alexander M Rush. 2017. LSTMVis: A tool for visual analysis of hidden state dynamics in recurrent neural networks. *IEEE Transactions on Visualization and Computer Graphics* 24 (2017), 667–676.

Enkelejda Tafaj, Thomas C Kübler, Gjergji Kasneci, Wolfgang Rosenstiel, and Martin Bogdan. 2013. Online classification of eye tracking data for automated analysis of traffic hazard perception. In *International Conference on Artificial Neural Networks*. Springer, 442–450.

Svante Wold, Kim Esbensen, and Paul Geladi. 1987. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems* 2 (1987), 37–52.

Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. 2015. Understanding Neural Networks Through Deep Visualization. arXiv:cs.CV/1506.06579

Raimondas Zemblys, Diederick C Niehorster, and Kenneth Holmqvist. 2019. gazeNet: End-to-end eye-movement event detection with deep neural networks. *Behavior Research Methods* 51 (2019), 840–864.

Raimondas Zemblys, Diederick C Niehorster, Oleg Komogortsev, and Kenneth Holmqvist. 2018. Using machine learning to detect events in eye-tracking data. *Behavior Research Methods* 50, 1 (2018), 160–181.

Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. 2019. MPIIGaze: Real-world dataset and deep appearance-based gaze estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41, 1 (2019), 162–175.