A Gray Box Interpretable Visual Debugging Approach for Deep Sequence Learning Model

Anonymous Author(s)
Affiliation
Address
email

Abstract

Deep Learning algorithms are often used as black box type learning and they are too complex to understand. The widespread usability of Deep Learning algorithms to solve various machine learning problems demands deep and transparent understanding of the internal representation as well as decision making. Moreover, the learning models, trained on sequential data, such as audio and video data, have intricate internal reasoning process due to their complex distribution of features. Thus, a visual simulator might be helpful to trace the internal decision making mechanisms in response to adversarial input data, and it would help to debug and design appropriate deep learning models. However, interpreting the internal reasoning of deep learning model is not well studied in the literature. In this work, we have developed a visual interactive web application, namely d-DeVIS, which helps to visualize the internal reasoning of the learning model which is trained on the audio data. The proposed system allows to perceive the behavior as well as to debug the model by interactively generating adversarial audio data point. The web application of d-DeVIS is available at `ddevis.herokuapp.com`.

1 Introduction

Machine Learning(ML) algorithms have been pouring the blessings in a form of solving Artificial Intelligence(AI) problems, such as classification, clustering, genomics data visualization, etc. Deep Learning(DL), an influential extension of ML, has been evolving rapidly in recent years and successfully being applied in solving various real-world problems including machine translation, speech recognition, image classification, etc[1][2][3]. While traditional ML models require external domain knowledge, DL is mostly characterized for efficient learning of the non-linear complex feature representation without having domain expertise. Hence, the DL model remains as a black-box type learning for practitioners and researchers. In effect, the interpretability and transparency of DL models have been reduced significantly [4]. Although DL approaches have been studied widely, a few works address the interpretability issue of deep learning models in the literature.

With the increasing use of the DL methodologies in real-world systems, such as self driving car and medical imaging, it becomes a prime concern to have publicly understandable systems explaining the underlying reasoning. Although the linear systems can be easily demonstrated with simple examples having mathematical proofs, non-linear systems, such as Deep Neural Network(DNN), is complex to understand and visualize. Nonetheless, the general users as well as researchers need to understand the mechanism of the algorithms to debug and determine appropriate learning model. In addition, the teachers and the learners are interested to visualize the algorithms to develop the basic intuition of the algorithm. The researchers have been working to utilize the visualization approaches to teach the ML algorithms [5] while it has been proven that people can grasp the principles of an algorithm better when they are taught using visualization approaches [6][7].

Submitted to 32nd Conference on Neural Information Processing Systems (NIPS 2018). Do not distribute.

Visualization of internal operation details of a machine learning algorithm has been studied previously in [8], where the authors have surveyed several visualization techniques to understand the learning and decision-making processes of neural networks and also describe their work in knowledge-based neural networks. After the explosion of deep learning applications in computer vision and machine translations, researchers have been trying to visualize the interpretations of the specialized algorithms used for different kinds of unstructured data. In [9], authors have introduced a novel visualization technique that gives insight into the function of intermediate feature layers and the operations of Convolutional Neural Network(CNN). Nonetheless, it is rather black-box type visualization approach to reveal the model behavior, as such it can not interpret the internal reasoning. In [10], authors have developed an interactive system to enable users understand and explore the deep learning models and get an insight on the learning mechanisms of image classifiers. It introduces a gray-box type approach but does not demonstrate how classifiers work in response to sequence audio data.

In this paper, we have designed a deep Sequence Learning Model Debugger and Visual Interactive Simulator, namely d-DeVIS, that focuses on gray box concept, where outcome of an internal block is transparent to the users. More explicitly, we are interested to visualize the internal feature representation of a deep sequence learning model (i.e. CNN) in response to multi modal audio sequence data. The layer wise visualization of hidden features in d-DeVIS assists us to understand the interpretation of feature extraction methods of DL models. The main contributions of the paper are as follows:

- A web-based application, d-DeVIS, to visualize the representation of hidden layers' features and the behavior of the CNN model in response to the adversarial audio sequence data.
- d-DeVIS, allows user to interactively change the audio features, such as pitch, amplitude etc, and interpret the behavior of the learning model based on the modified data.
- We have designed a visually transparent debugging User Interface(UI), which demonstrates layer-wise features' representation and model hyper parameters. In so doing, it guides DL model's debugging.
- d-DeVIS enables users to hear and visualize the intermediary hidden layer results, layer-wise converted audio outputs and weight distributions, in order to interpret the final prediction. It also allows practitioners to compare the performance of the learning model in response to different adversarial audio input.

The rest of the paper is structured as follows. In Section 2, we discuss the related work. Thereafter, Section 3 is focused on the goals and features of the proposed system. Section 4 describes the use cases of d-DeVIS. Finally, Section 5 concludes with future plans.

2 Related Work

The recent widespread use of deep learning models in various artificial intelligence task attracts both the visualization and the deep learning communities to deal with the new challenge of improving the interpretability and explainability of these models [8]. It is worth mentioning that visualizing the Neural Network (NN) models is not a new research domain. To be precise, it has been studied well before the recent surge of deep learning models. For instance, N2Vis [5] visualizes the attributes of NN, such as hidden layers weights, weights' volatility, network structure and nodal activation levels. Nonetheless, most of the previous approaches utilize the static graphical visualization to describe hidden reasoning of the learning models.

In recent years, a number of works have been sought to address the explainability and transparency issue of the DL models and few others have been focused on designing interactive visualization models to illustrate underline reasoning. For example, Tensorflow Playground [11] designed an interactive interface, where users can change the parameters and structure of the NN models and examine their effect. Moreover, ShapeShop [10] enables the users to interactively change input image and visualize the behavior and feature's representation of the DL models. Similarly, in [12], authors designed an application, which allows an user to examine the behavior of a DL based image classifier.

Apart from these black-box visualization approaches, a number of works visualize the behavior of deep learning models. For instance, in [13], authors present a static visualization of hidden state representation and the prediction model behavior of Long-Short-Term-Memory(LSTM) based language model. Similar to the previous work, LSTMVis [14] designed an interactive visualization approach to visualize the hidden state representations of recurrent neural network and allows user to examine the internal behavior of LSTM model on different application scenarios. Additionally, in [15] and [9], authors visualize the Convolutional Neural Network (CNN) and provide visually explainable reasoning of internal feature representation. Furthermore, Seq2Seq [16] designed a visual debugging tools for the sequence-to-sequence learning model and enables users to interact with the model to develop an insight about the model.

Inspired from the previous works done by [10, 14, 16], we have designed an interactive visual DL models debugging system, d-DeVIS: Deep Sequence Learning Model's Debugger and Visually Interactive Simulator. Most of the previous works utilize the black box visualization approaches to help developing the basic intuition of the deep learning models. Surprisingly, visualizing the deep learning model behavior and features representation of the multimodal data, such as audio or video, is neglected in the literature. Moreover, visualizing the correlation between the hidden layer features representation and the model behavior is not properly studied for sequence models. d-DeVIS allows user to interactively change the multimodal audio data to generate adversarial data examples and enables users to examine the deep learning model behavior to visualize the features representation.

3 Design and Development of d-DeVIS

In this section, we present the key components and goals for designing our proposed interactive application to visualize DL model in response to the adversarial data input. We take into considerations the interactivity of the users and flexibility of the system. To do so, we have developed a web application that shows the gray box debugging method for deep neural network of sequence data. The prime goal of designing d-DeVIS is to make the learning and debugging DL model user friendly and also ensure that it should be able to visualize the internal reasoning of deep sequence model and features representation of hidden layers with the help of an interactive user interface. Table 1 lists a number of major design goals for designing an interpretable deep audio sequence learning model.

Table 1: Design Goals of d-DeVIS

Goals	Description
G1: Improve DL Models Interpretability and Transparency	An interpretable system of DL models depicts how deep sequence learning models work and how the hidden layer features can help to easily interpret the functionality of the learning model.
G2: Gray-Box Visual Debugging	A good grasp of the feature extraction method of deep neural networks is required for DL enthusiast and d-DeVIS provides a fluid gray box debugging experience which enables the users to understand how the features of the hidden layers affect the training.
G3: Interactively Examining the Deep Sequence Model Behavior	An interactive tool is required, where user can manipulate audio features(such as slicing, cross-fading, repetition, etc) to generate adversarial example data. Moreover, it allows user to examine the internal reasoning in response to the modified adversarial data.
G4: Comparison and exposure of the extracted features from audio data	The proposed system must enable users to listen the extracted audio data from different layer after applying CNN filters. Hence, users should be able to grasp the extracted hidden layer audio features.

3.1 Features of d-DeVIS

We have designed d-DeVIS as an interactive web application while considering the design goals listed in Table 1. The primary goal of our proposed system is to ease the interpretation of the intermediate reasoning and the deep audio sequence learning model. We divide the proposed d-DeVIS model into the following three major components.

- Model Visualization.
- Audio Feature Manipulation
- Adversarial Feature Comparison

3.1.1 Model Visualization

The primary purpose of our work is to interpret the internal reasoning of the deep sequence learning model in response to adversarial audio example data. For this reason, d-DeVIS provides an interactive web application interface, which depicts the intermediate layer wise visual features representation in the form of audio spectrogram. Moreover, we employed the inverse Fourier transformation to extract the audio features from the intermediate layer spectrogram. d-DeVIS allows user to not only visualize the features extracted by the hidden layer filters, but also it enables them to listen to the audio representation of the features of the input audio extracted by the CNN. The web interface to visualize the layer wise feature is depicted in Fig 1. Furthermore, d-DeVIS allows the users to examine the weight distributions of the internal hidden layer. To extract the intermediate features, we trained a baseline CNN model on audio sequence data. The details of the trained model and the backed system of d-DeVIS is presented in Section 3.2.

During any forward propagation step, the spectrogram feature data of the audio files are traversed through the hidden layer of the CNN. At each layer, the convolution filter tries to extract significant hidden features from the audio data input and optimizes itself during backward propagation in order to minimize the training loss. In our system, the users will be able to upload an audio file or record an audio of their own. After the processing of the input, our system will calculate the logarithmic spectrogram and feed it into the trained model to produce the prediction. At each convolution layer there are predefined tunned filters. The types of features CNN] extracts from the input data depends on the filters. In our trained model, the first layer and second layers have 16 filters, the third layer has 32 filters, each. So, our system visualizes the features corresponding to the filters and also the distributions of the trained weights.

A particular feature extracted by the 13th filter of first layer is visualized in Fig 1(a). When a user clicks on the image it zooms in to show the spectrogram clearly. Users can also listen to the hidden extracted feature by clicking on the play button, which is depicted in Fig. 1(c).

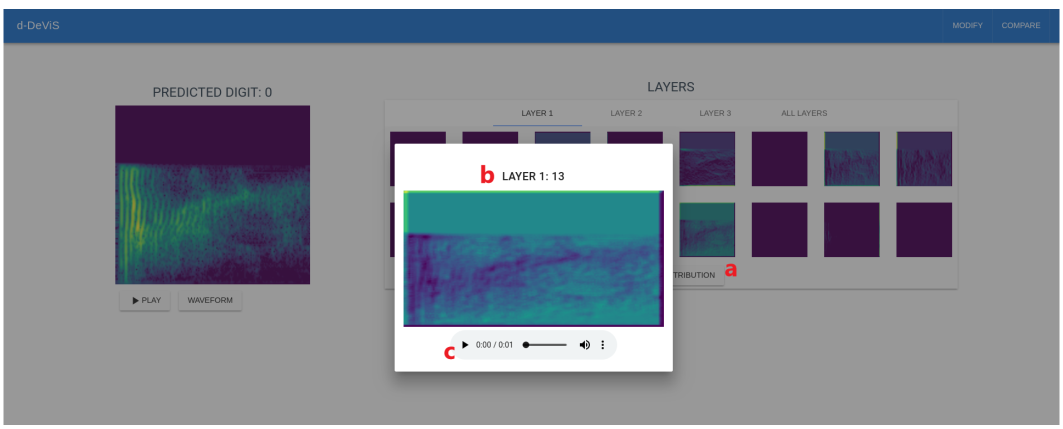


Figure 1: Visualization of audio features extracted by CNN layer filter.

3.1.2 Adversarial Feature Comparison

d-DeVIS allows users to interpret the DL model behavior by examining the intermediate feature representation based on the different audio data input. The adversarial behavior comparison is illustrated in Fig 2 and the different module of this feature is presented in red alphabets. The module a and b are the two spectrogram representation of the two audio inputs with their predictions by the trained deep sequence learning model. Users can observe the feature representation of different layers in module d and e. There are different spectrogram images of the extracted features and users can click on them to listen to the audio representation. Finally users can also see the weight distribution of each layers by clicking on the button marked by f.

3.1.3 Audio Feature Manipulation

Our proposed interactive system, d-DeVIS, enables the users to not only examine the behavior of the learning model in response to a sample file or recording of their voice but also it allows users to

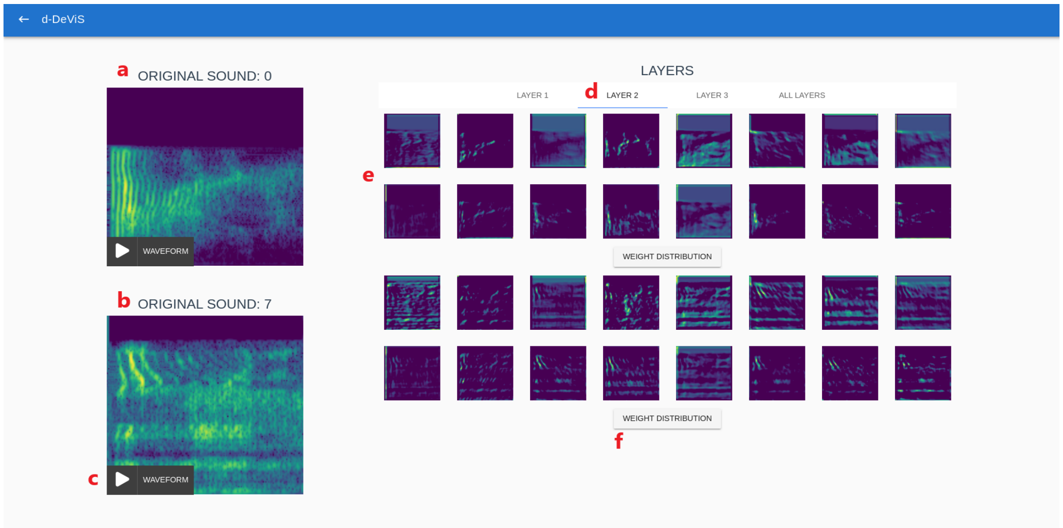


Figure 2: Comparison of different audio inputs.

manipulate the different properties of audio example data and thus enables to generate adversarial example data. Among the various characteristics of the audio, which can be changed, d-DeVIS allows to manipulate the following audio features.

- **Slicing** allows users to slice an audio.
- **Cross-fading** changes the amplitude of the sound waves.
- **Changing the loudness** option will make the beginning louder and the ending quieter.
- **Repeating** option repeats the sound twice.
- **Invert:** allows to invert the sound wave, i.e. inverted sound will be played from the ending.
- **Fade:** option fades in for a particular time and then fades out similarly.

A pictorial modification of audio feature is presented in Fig. 3. After manipulating and generating the audio example data, d-DeVIS allows users to examine the behavior of audio deep sequence learning model by observing behavior changes in response to the original and adversarial audio data input.

In Table 2, we discuss all the features of d-DeVIS and how the features meet the design goals.

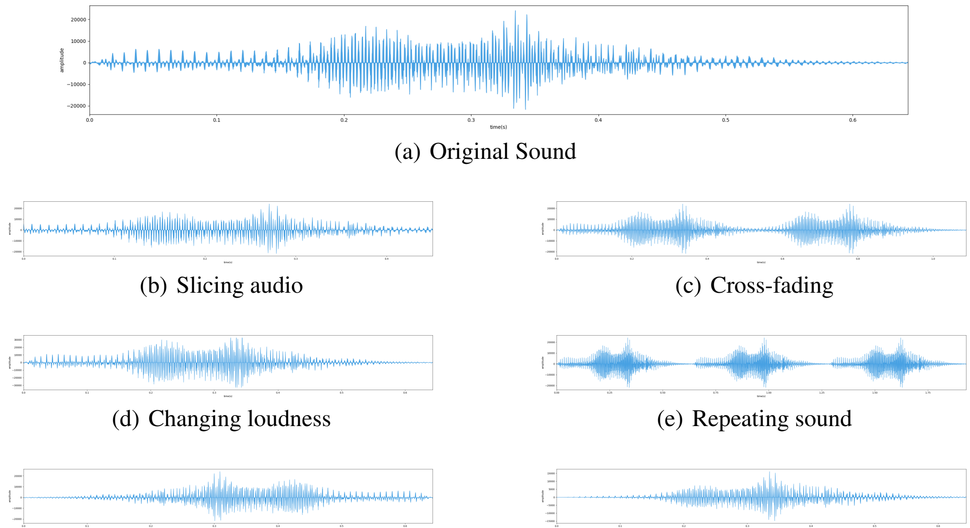
3.2 Implementation of d-DeVIS

d-DeVIS is developed as a web application so that users can seamlessly interact with the system to interpret the behavior of deep learning model by generating adversarial audio input data. In the following section we present the implementation details of d-DeVIS. The source code of our implementation is available at `https://github.com/anon-conf/d-DeVIS`.

3.2.1 Trained Deep Learning Model with Audio Data

We trained a CNN model on Speech Commands dataset [1], which is used to visualize the behavior of the model. The dataset consists of almost 30 speech classes but for the sake of the simplicity and reduction of training time we used 10 classes, which are the audio recordings of zero to nine digits in English language. All the clips are one second long. We calculated logarithmic spectrogram as features of the audio (.wav) data to feed into the training model. A three layer Convolutional Neural Network (CNN) is used as the spectrogram feature matrix represents an image and CNNs have proven to be decent at image classification. The Convolutional Architecture consisted of 3s set of filters with different square kernel of sizes [7,5,3]. We used max pooling after every filter to reduce the sizes of the output matrices and added necessary dropout to reduce overfitting. The complete architecture of

[1] `www.kaggle.com/c/tensorflow-speech-recognition-challenge`



(a) Original Sound

(b) Slicing audio

(c) Cross-fading

(d) Changing loudness

(e) Repeating sound

(f) Invert

(g) Fade

Figure 3: Visualize the modified audio features. (*time* vs *amplitude*)

Table 2: Mapping of d-DeVIS features and goals

Feature	Goal
Visualizing the Hidden Extracted Features of Convolutional neural networks: d-DeVIS provides visualization of the extracted features in each layer as image data and shows the various features of different filters of the deep Convolutional Neural Network. Users can also hear the audio representation of the hidden extracted features	G1 & G2 & G4
Interactive User Experience: For a fluid user experience,we provide an interactive platform for the users so that they will be able to focus on the productivity of the system without any unnecessary hassle.	G3 & G4
Visualizing the Audio Features as well as Modifying the Waveforms: Due to the complex structure of audio data, our system let's users modify various aspects of the sound property and visualize the updated waveform to provide a keen knowledge on audio data representation.	G2 & G3
Custom Audio Input for Testing and Feature Distribution Visualization: User can not only upload a default audio data but also they can record custom speech to test the trained model. Proper distribution of the weights is also visualized.	G1 & G3
Comparing different audio inputs and their hidden features: d-DeVIS also enables users to measure the differences of different audio inputs and check their extracted layer features.	G4

the training model is depicted in Fig 4. Our baseline model reached 95% validation accuracy with a minimal hyper parameter tuning. We have used Keras deep learning framework which is a wrapper library of Tensorflow to train our deep learning model. We have utilized the computation system of Google Colaboratory platform for the training purpose.

3.2.2 Front-end and Back-end of d-DeVIS

Audio files manipulations such as Slicing the audio, Changing Loudness, Cross-fading, Repeating the Sound, Invert and Fade are done using Numpy, Pydub and Scipy libraries. Matplotlib is utilized to visualize the audio features. After the training of the model, we have saved the data using Pickle python module. We have used HTML5, CSS, Javascript for designing the front-end of the application. We used the Vue.js framework to build an SPA and communicated with the server using REST API. The back-end is built in python using the Flask framework.

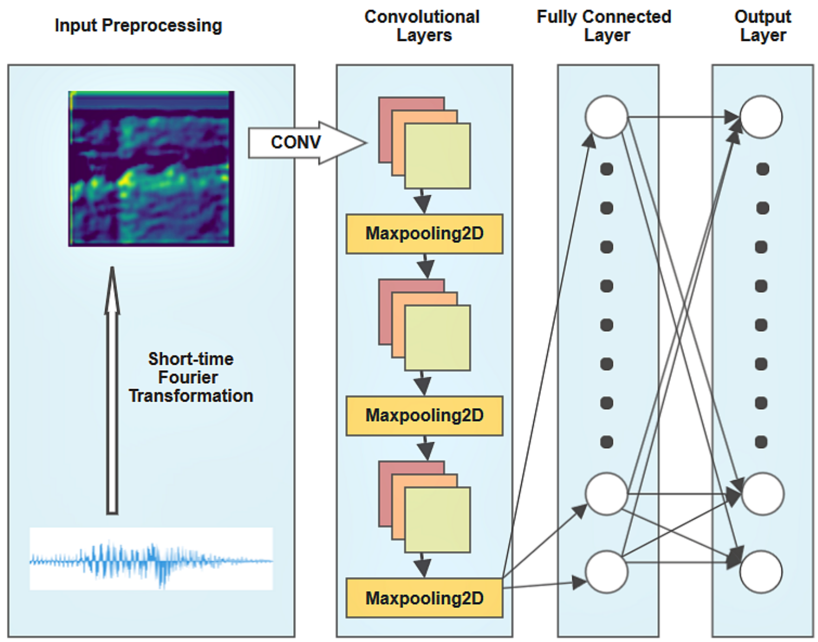


Figure 4: Convolutional Neural Network Architecture of d-DeVIS.

4 Use Cases

While experimenting with the system, we have applied several modifications to both the CNN trained model and the input audio file. Then, we have analyzed the obtained results by tuning with the system. In the remainder of this section, we present important use cases that demonstrate the general applicability of our system. A demo video of our proposed system d-DeVIS can be found at `http://bit.ly/ddevis-demo`.

- **Visualizing the Audio Features:** Speech is a sequence data which is hard to grasp just by looking at the amplitude vs time representation. In our system, a user can upload or record a customized audio file and tune with various aspects of the waveform. Therefore, predictions of the audio will change with accordance to the change in the waveforms and users can easily observe the changed results.
- **Learning Medium for the Academia:** Our system provides an interactive web application with which learners will be able to test various types of aspects of audio data and the deep learning model. By using d-DeVIS , academics can provide appropriate insights of the feature extraction method of neural networks to the students. Hence, it can be a great medium for learning.
- **Experimenting Platform for AI Enthusiasts:** We provide a platform for easy training and proper results of the feature extractions which are shown as a form of images. Users can test their own custom input and observe the decisive hidden features that make the distinctions between the inputs. Thus, the feature manipulation and interactivity of the system will inspire the deep learning enthusiasts and engineers to do various experiments on it.

5 Conclusion

d-DeVIS allowed the users to visualize how CNN recognizes digits from audio sequence data. It collected input from user and allowed them to interactively manipulate it. The tool easily allowed the comparison of the given input with other adversarial examples. Overall, this helped users to develop a better intuition of the underlying reasoning of the model which allowed them to make more learned decisions regarding learning model development.

In future extension of d-DeVIS, we have the plan to visualize other sequence deep learning models behavior and allow users to manipulate the input data representation interactively. Moreover, visualizing the hidden layer complex feature representations for multi-modal sequence data is a great avenue for future research work.

References

[1] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[2] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, *et al.*, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *International Conference on Machine Learning*, pp. 173–182, 2016.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[4] F. M. Hohman, M. Kahng, R. Pienta, and D. H. Chau, "Visual analytics in deep learning: An interrogative survey for the next frontiers," *IEEE Transactions on Visualization and Computer Graphics*, 2018.

[5] M. J. Streeter, M. O. Ward, and S. A. Alvarez, "Nvis: An interactive visualization tool for neural networks," in *Visual Data Exploration and Analysis VIII*, vol. 4302, pp. 234–242, International Society for Optics and Photonics, 2001.

[6] A. Robins, J. Rountree, and N. Rountree, "Learning and teaching programming: A review and discussion," *Computer science education*, vol. 13, no. 2, pp. 137–172, 2003.

[7] B. Du Boulay, "Some difficulties of learning to program," *Journal of Educational Computing Research*, vol. 2, no. 1, pp. 57–73, 1986.

[8] M. W. Craven and J. W. Shavlik, "Visualizing learning and computation in artificial neural networks," *International journal on artificial intelligence tools*, vol. 1, no. 03, pp. 399–425, 1992.

[9] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*, pp. 818–833, Springer, 2014.

[10] F. Hohman, N. Hodas, and D. H. Chau, "Shapeshop: Towards understanding deep learning representations via interactive experimentation," in *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, pp. 1694–1699, ACM, 2017.

[11] D. Smilkov, S. Carter, D. Sculley, F. B. Viégas, and M. Wattenberg, "Direct-manipulation visualization of deep networks," *arXiv preprint arXiv:1708.03788*, 2017.

[12] Á. Cabrera, F. Hohman, J. Lin, and D. H. Chau, "Interactive classification for deep learning interpretation," *arXiv preprint arXiv:1806.05660*, 2018.

[13] A. Karpathy, J. Johnson, and L. Fei-Fei, "Visualizing and understanding recurrent networks," *arXiv preprint arXiv:1506.02078*, 2015.

[14] H. Strobelt, S. Gehrmann, H. Pfister, and A. M. Rush, "Lstmvis: A tool for visual analysis of hidden state dynamics in recurrent neural networks," *IEEE transactions on visualization and computer graphics*, vol. 24, no. 1, pp. 667–676, 2018.

[15] M. Liu, J. Shi, Z. Li, C. Li, J. Zhu, and S. Liu, "Towards better analysis of deep convolutional neural networks," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 1, pp. 91–100, 2017.

[16] H. Strobelt, S. Gehrmann, M. Behrisch, A. Perer, H. Pfister, and A. M. Rush, "Seq2seq-vis: A visual debugging tool for sequence-to-sequence models," *arXiv preprint arXiv:1804.09299*, 2018.