An Interactive Graphical Visualization Approach to CNNs and RNNs

Akhil Vyas*, Prasad Calyam**

*University of Illinois Urbana-Champaign, **University of Missouri
akhilv4@illinois.edu, calyamp@missouri.edu

Abstract—Deep Learning models such as Convolutional Neural Nets (CNNs) and Recurrent Neural Nets (RNNs) are routinely used due to their utility and benefits in various practical applications such as e.g., natural language processing and image recognition. However, despite such adoption, these techniques typically present opaqueness of their internal workings to users. The current black box approach to deep learning makes models difficult to understand and fine-tune, and the related lack of information directly influences productivity and hinders innovations in reliable and consistent model development. In this paper, we present a novel web-based, graphical approach to visualizing CNNs and RNNs to address the above adoption challenges. Our approach features an interactive graphical user interface, where the user can view the overarching network architecture and data flow, the weights and corresponding input processing at each layer, and some interpretable aspects of the model as a whole. We show the effectiveness of our visualization techniques on the MNIST dataset corresponding to an image recognition application. Our work contributes to the effective techniques on the MNIST dataset corresponding to an image recognition application. Our work contributes to the effective

Index Terms—CNN Visualization, RNN Visualization, Explainable AI, Deep Learning Visualization

I. INTRODUCTION

In recent years, deep learning techniques have gained widespread industry usage due to their ability to process a large number of features. They also have been effective in solving complex problems involving classification, regression, and policy recommendation. In particular, Convolutional Neural Nets (CNNs) and Recurrent Neural Networks (RNNs) have become routine in computer vision and natural language processing applications, respectively. However, a continued problem of these models is the opaqueness of their internal workings. With the high complexity of these models, it is difficult to gauge the underlying mechanism through which models make decisions. This causes a black box approach to deep learning which makes it difficult to understand and fine-tune the behavior of these models. Consequently, users experience an elongated process to come up with reliable and consistent models. As these models continue to increase in complexity with usage in high-risk domains such as the medical field, it becomes imperative to be able to effectively understand and interpret the decisions of these models so that their behavior can be determined and regulated effectively.

The underlying weaknesses of current deep learning techniques has led to research initiatives around creating Explainable AI (XAI) [1]. Explainable AI seeks to add dimensions to the existing deep learning architectures so that they display sufficient information about the internal workings and prediction rationale. They also help to identify failure points within the network, or are able to incorporate expert domain knowledge into the training process, or all of the above [2]. Interactive visualizations are a powerful form of XAI which give practical intuition about the network architecture. They do so by enabling an understanding about the role of individual layers and nodes through direct experimentation. Exemplar prior works such as Tensorflow Playground [3] allows users to change hyperparameters of simple neural networks and showcases the derived features of all neurons in the network with a heatmap of the encompassing decision boundary for the whole network.

In this paper, we present a novel web-based, graphical approach that integrates visualization techniques relevant to CNN and LSTM architectures. In our visualization approach, users can manipulate parameters of both networks and pass in inputs of their choice and view the corresponding input processing that occurs at each layer and at individual nodes. For CNNs, the parameters that can be viewed include the weights at the convolutional, pooling and fully-connected layers, while for LSTMs, the parameters include all the components of an LSTM cell. On top of the model architecture, we also visualize interpretable features of the model prediction process. We extend on the work done in [4] for CNNs, where images are constructed that maximize the class score to showcase model representations of different classes, by extracting the relevant features at each layer. For RNNs, we display the activation patterns for user-selected nodes based on sequential input. To evaluate the effectiveness of our visualization approach, we consider four different metrics: the goodness of the explanation, user satisfaction, user understanding of the system, and performance.

Our paper contributions can be summarized as follows:

(1) We develop visualization techniques for both CNNs and RNNs considering the data flow, layer and neuron weights, and allow user manipulation of network parameters.

(2) For CNNs, we generate images that maximize class score
to showcase representative images and display features of these images at each layer.

(3) For RNNs, we visualize activation patterns of LSTM layer for user selected images.

The remainder of this paper is organized as follows: Section II discusses related works corresponding to Explainable AI and existing visualization frameworks and techniques for CNNs and RNNs. Section III gives a formal description and methodology of our CNNs and RNN visualization methodology. Section IV details the effectiveness of our visualization in terms of usability and performance. Section V concludes the paper with a discussion on our findings and possible future research directions.

II. RELATED WORKS

A. Explainable AI

Many prior works have conducted comprehensive reviews of the field of XAI and discussed the need for related systems. The authors in [5] discuss various notions of XAI including interpretable systems (show mappings between layers) and comprehensible systems (show symbols with output that users can infer). Similarly, a survey in [6] explores these two notions of XAI, further categorizing them into transparent (to see internal workings in action) and post-hoc (to see information from learned model) approaches. Furthermore, survey work done in [7] focused on creation of interpretable AI that analyzed research gaps and opportunities in creating more interpretable models in terms of model transparency and functionality. In our work, we attempt to strike a balance between these various notions of XAI and aim to show the internal workings while displaying interpretable properties of the model. Other works, such as [8], discuss the need and role of XAI in verifying and improving models, and explain how learning insightful patterns from these improved models can be used to build up domain knowledge. Metrics for comparing and evaluating XAI systems are detailed in [9].

Prior works also explore theoretical approaches and metrics for creating and evaluating XAI systems. In [8], authors explain sensitivity analysis and layer-wise relevance propagation methodologies for analyzing and visualizing deep learning models. In [10], authors propose a model-agnostic approach that uses Bayesian teachings to give probabilistic interpretations of any machine learning model. Finally, a survey paper in [11] conducts a comprehensive study of various methodologies for explainable machine learning and classifies them based on notions of XAI.

B. Visualization for CNNs

Much work has been done in visualizing features of trained models in convolutional neural networks. In [12], authors developed two tools, one that visualized activations of the trained network, and one that visualized layer features via regularized optimization. Two visualization approaches were proposed in [13], which visualized the internal representation of images and the representation spaces constructed by internal layers. Different methodologies for feature visualization were reviewed in [14] including input reconstruction, input modification, and deconvolutional methods.

Other works take a different approach to visualizing CNNs. Authors in [15] proposed a prediction difference analysis methodology that highlighted areas of an image that provided evidence for and against a class. Work done in [16] aimed to use a distillation process to convert trained CNN models to easily-readable decision trees. Authors in [4] proposed two techniques, one that generated images that maximized class score, and one that generated a saliency map based on image and class. As previously mentioned, in our work we extend upon this methodology and generate visualized features for the generated images. A survey done on visual analytics for deep learning methods [2] discusses other state-of-the-art visualization tools for CNNs fulfilling various purposes such as model debugging, interpretation, and explanation. Another survey in [17] explores visualization methods and offers insights into the focus areas and interpretation perspective of these techniques. Layer-wise propagation, sensitivity analysis, and deconvolutional methods are evaluated in [18] and related findings are reported on how layer-wise propagation methods offer better explanations.

C. Visualization for RNNs

Many visualization works on RNNs focus on LSTM and RNN architectures for NLP applications. Authors in [19] present an interactive glyph-based visualization tool for LSTM and GRU architectures for text-based applications that displays relation between input and hidden-state units as bipartite graphs. Authors in [20] use character-level language models to analyze representations, predictions, and error types of LSTM networks. A study done in [21] applies and evaluates input derivative, occlusion, masking, and temporal visualization methodologies for an electrocardiogram application. Their experiments find input masking to be the best visualization technique in terms of reduction. The work in [2] also offers insights into LSTM visualization tools that visualize activation patterns of hidden nodes. We use a similar approach to display activation patterns for user-selected neurons (individual or grouped). Lastly, a survey conducted in [22] presents different visualization techniques in terms of different notions of XAI for RNNs.

III. METHODOLOGY FOR GRAPHICAL VISUALIZATION

A. CNN Implementation

The main interface of our CNN implementation displays the network architecture and layer information as shown in Fig. 1. The architecture visualizes the dataflow and the input and output shapes of the data at different layers. Users can use a slider to view layer information specific to the type of layer. For convolution layers, we display the filter weights and normalize the weight to display a grayscale image of the weights. Fig. 1 shows the interface for convolution layers. Users can interact with the interface and view or manipulate the weights for different layers, and different input and output channels. Furthermore, we reconstruct the outputs of the
Fig. 1: The visualization interface for CNNs showing a convolution layer; Left: shows the encompassing network architecture, Right: shows layer specific information; Users can interact with a slider to view information specific to a type of layer.

Fig. 2: Visualization of dense layer weights distribution given a node of the layer; helps to understand which parts of input are more important than other convoluted layers and display a grayscale image for each of the outputs. Users can pass their own images into the network and view the corresponding input processing. For the pooling layer, we display the functionality of the layer, visualizing the window size and the stride values. Finally, for the fully-connected layers, we display a plot of the weights given a selected node as shown in Fig. 2.

We also generate images that maximize the activations of classes. We do so by using gradient ascent and maximizing the loss the function characterized by a particular output class. We take the gradient of the loss function with respect to the pixel values and manually update the values iteratively. The images can be seen in Fig. 3 for the MNIST dataset [23].

**B. LSTM Implementation**

Similar to our CNN implementation, our LSTM implementation also displays the network architecture and allow users to move between layers, viewing different kinds of information based on the type of layer.

For our LSTM implementation, we visualize various aspects about the LSTM layer and its cells. We use a heatmap for the three different kinds of weights in the LSTM layers: the kernel, recurrent and bias weights. Furthermore, we further categorize the weights based on the type of gate which include the input, forget, cell state, and output gates. The users are able to interact with the interface to see the different gates and types of weights as shown in Fig. 4. We also visualize the input processing occurring at LSTM layers through a plot of the activations generated by the layers. The plot helps users understand which particular nodes are activated for different kinds of inputs and input classes. An example activation plot is shown in Fig. 4.

**IV. Performance Evaluation**

In this section, we first present the evaluation metrics for measuring the effectiveness and usability of our interface. We conducted a two-part study with 15 experts and non-experts of deep learning methods. In the first iteration, we aimed to...
Fig. 3: Class representative images generated through 50 iterations of gradient ascent maximizing the class output scores; the images capture the notion of a class, and show what a model is looking for when classifying images as a particular class.

Fig. 4: Visualization interface for RNNs showing a LSTM layer; the input image is shown in the center with the Activation Plot for the image on the bottom. On the right, a heatmap for four type of weights and three types of gates is displayed.

TABLE I: Evaluation Process and Summary of Iterations

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Objective</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Measure effectiveness of visualization and get user feedback</td>
</tr>
<tr>
<td>2</td>
<td>Make changes and get final measurements for effectiveness along with usability (SUS)</td>
</tr>
</tbody>
</table>

TABLE II: Demographics of Survey Participants

<table>
<thead>
<tr>
<th>Participant Type</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Male</td>
<td>10</td>
</tr>
<tr>
<td>Female</td>
<td>5</td>
</tr>
<tr>
<td>Expert</td>
<td>8</td>
</tr>
<tr>
<td>Non-Expert</td>
<td>7</td>
</tr>
<tr>
<td>Undergraduate</td>
<td>7</td>
</tr>
<tr>
<td>Graduate</td>
<td>8</td>
</tr>
<tr>
<td>Total</td>
<td>15</td>
</tr>
</tbody>
</table>

measure the response on how our interface was organized and the utility of the information that was provided to the users. Based on the user feedback from the first iteration survey experiments, we updated our interface to increase ease-of-use and clarity of the information provided to the users. Thereafter, we organized a second iteration where we additionally measured the usability of our interface with no feedback loop. Table I shows our evaluation process with objectives at various stages. In this two-part study, we aimed to garner feedback from users with varying deep-learning backgrounds and education levels. Table II lists the demographics of our survey participants.

A. Evaluation Metrics

We measured the effectiveness of our visualization methodology through user satisfaction surveys that gauged the benefit and value of various aspects of our implementation. The aspects considered were: (i) goodness and clarity of explanation for individual decisions and layer functionality, (ii) understanding of the model architecture as a whole, and (iii) completeness of the information provided. These aspects were measured on a 5-point scale as shown in Table III. Evaluation
of our visualization interface utility based on these metrics provides insights on the understanding that users gain at various levels of focus (individual node vs. individual layer vs. whole model) for a specified input image.

**TABLE IV: Iteration 1: User Satisfaction Survey Results**

<table>
<thead>
<tr>
<th>Question</th>
<th>CNN</th>
<th>RNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clarity of the explanation</td>
<td>4.47</td>
<td>3.8</td>
</tr>
<tr>
<td>Utility of the explanation</td>
<td>4.06</td>
<td>3.73</td>
</tr>
<tr>
<td>Understanding individual decisions</td>
<td>4.13</td>
<td>3.53</td>
</tr>
<tr>
<td>Understanding the overall model</td>
<td>4.33</td>
<td>4</td>
</tr>
<tr>
<td>The Explanation has sufficient detail</td>
<td>4.13</td>
<td>3.67</td>
</tr>
</tbody>
</table>

**TABLE V: Iteration 2: User Satisfaction Survey Results**

<table>
<thead>
<tr>
<th>Question</th>
<th>CNN</th>
<th>RNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clarity of the explanation</td>
<td>4.65</td>
<td>4.05</td>
</tr>
<tr>
<td>Utility of the explanation</td>
<td>4.16</td>
<td>3.87</td>
</tr>
<tr>
<td>Understanding individual decisions</td>
<td>4.2</td>
<td>3.8</td>
</tr>
<tr>
<td>Understanding the overall model</td>
<td>4.67</td>
<td>4.32</td>
</tr>
<tr>
<td>The Explanation has sufficient detail</td>
<td>4.43</td>
<td>3.95</td>
</tr>
</tbody>
</table>

**B. System Prototype 1 Evaluation**

The results of our preliminary survey are encapsulated in Table IV. The preliminary results suggested that our visualization approach with respect to the CNN architecture was effective in terms of clarity and understanding of the model and individual decisions, utility, and the amount of detail provided. In contrast, the RNN framework was found lacking for understanding individual decisions and did not offer sufficient detail. The following were a selection of problems that were mentioned from the first user survey:

"LSTM layers in RNN visualization are not clear... Can’t understand what the information means... Add more information”

**Context:** The heatmaps for our LSTM layers did not have labels which made it difficult to understand what the individual pixels were supposed to represent.

"More information about how to use interface. It was hard to follow the text... Took some time to figure out how everything worked”

**Context:** The description provided for usage information was lacking and contained only text.

"Improve the UI. Using sliders is a good option, however at the very beginning it is a little confusing, and it takes some time to understand that each step of the slider points a different layer”

**Context:** The design using a slider to navigate between layers was not intuitive. Furthermore, the interface was overly space consuming, such that the information was not effectively organized and the users had to keep scrolling to reach different parts of the interface. Fig. 5 shows our initial implementation with the highlighted issues raised by the users during the first iteration.

**C. Improved System Prototype 2 Evaluation**

In our second effectiveness survey, we aimed to resolve the issues that were raised in the previous iteration by the users. We changed the sliders to buttons so that navigation between layers was more intuitive. We added images, similar to Figs. 1 and 4 at the start of the interface to facilitate user navigation of the interface. Additionally, the interface was made more concise such that the user could view all the layer information, model architecture, and interpretive features with minimal movement. Fig. 6 shows the reworked interface with the highlighted changes that resolved the user issues.

The results of the second survey are encapsulated in Table V, where we saw an improvement in all metrics for RNNs. We also measured the effectiveness of our user interface using the System Usability Score (SUS) method [24]. Results are encapsulated in Table VI and shows that with an overall SUS of 91.25, our interface can be deemed to be user-friendly with a high ease-of-use.

**D. Discussion**

In this section, we present a discussion on the plausibility of web-based visualizations and areas needing improvement based on our iterative implementation efforts and survey results. Providing visualizations through web-based technologies provides convenience to users i.e., users only need to upload their model to the interface, with no installation or package requirements. However, a limitation that arises from such a convenience is that it is harder to accommodate larger models with higher number of layers because of processing times for calculating outputs of layers. Visualizing interpretative aspects such as the generation of class images utilizing gradient ascent is time-consuming. Currently, this process is done in parallel with our interface displaying the weights concurrently, but further investigation is needed on making this process run in a faster and optimized manner to improve user experience.

Our results showcased how the visualization of models fulfills both interpretative and transparent notions of XAI, and can be very effective in offering complete information about a model. Additionally, such visualizations are more accessible to users, and can be utilized and appreciated by both experts and novice users in the field. Enabling navigation between layers can enable greater vision into the data flow and the role of individual layers, giving better understanding of the overall
TABLE VI: System Usability Survey Results

<table>
<thead>
<tr>
<th>Questions (Strongly Disagree - Strongly Agree) 5 point scale</th>
<th>Mean Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>I think that I would like to use this system frequently</td>
<td>4.5</td>
</tr>
<tr>
<td>I found the system unnecessarily complex</td>
<td>1.25</td>
</tr>
<tr>
<td>I thought the system was easy to use</td>
<td>4.625</td>
</tr>
<tr>
<td>I thought the system was easy to use</td>
<td>5</td>
</tr>
<tr>
<td>I thought the system was easy to use</td>
<td>1.25</td>
</tr>
<tr>
<td>I would imagine that most people would learn to use this system very quickly</td>
<td>4.5</td>
</tr>
<tr>
<td>I found the various functions in this system were well integrated</td>
<td>4.625</td>
</tr>
<tr>
<td>I felt very confident using the system</td>
<td>1.75</td>
</tr>
<tr>
<td>I needed to learn a lot of things before I could get going with this system</td>
<td>91.25</td>
</tr>
</tbody>
</table>

input processing. From our results, we can also identify areas for future exploration. Having an activation plot for LSTM layers did not offer an adequate explanation for the internal workings and input processing. Additional investigation can be done on visualization of the LSTM layer that takes advantage of an image recognition application.

V. CONCLUSION AND FUTURE WORK

In this paper, we proposed an interactive visualization methodology for convolutional neural networks and the Long Short Term Memory (LSTM) architecture of RNNs. Using a web-based, graphical approach, we visualized various aspects of the model including the overall model architecture, the data flow, and present information related to specific layers. For convolution layers, we displayed the filters as a heatmap and displayed the output images of the layer in grayscale. For dense layers, we visualized the node activations as a bar chart. In regards to RNNs, we visualized the different weights (kernel, recurrent, and bias) for different types of gate (input, cell state, forget, output) as a heatmap. We also displayed interpretive features of models. For CNNs, we generated class representative images using a gradient ascent methodology to facilitate the understanding of a notion of a class with respect to the model. For RNNs, we visualized activation patterns of nodes in the LSTM layer.

For evaluating the performance of our implementation, we conducted a two-part survey for measuring effectiveness of the visualization framework and a System Usability Score (SUS) survey to measure the usability of our visualization user interface. Our findings suggested that our interactive visualizations are effective in aiding understanding of the internal workings for different layers and under different inputs.

Based on our development process and evaluation results, we discussed the potential of web-based technologies and interactive visualization, along with open issues for improvement. Using a web-based interface requires less work from users and provides convenience, but makes it harder to accommodate larger and complex models. Visualizations fulfilling transparent and interpretative notions give more complete information that can be utilized by both experts and novice users in the field. Future work can involve investigation of visualizations for LSTM layers that take advantage of an image recognition application.

REFERENCES


Fig. 6: Final RNN interface for iteration 2 with resolved problems