Deep Neural Network Layer Visualization

Michael L. Iuzzolino*

ABSTRACT

Despite the burgeoning success of Deep Neural Networks (DNNs) and their pervasion into myriad domains, ranging from art and music to science and medicine, their deep and complex architectures, innumerable parameters, and nonlinear activations render their inner workings nearly incomprehensible. This 'black box' nature of DNNs prohibits not only a robust intellectual understanding for academic endeavors and optimization engineering, but it also restricts these models and their exceptional pattern-recognition power to expert-users. Importantly, the lack of understanding how and why a DNN makes any particular decision also leads to significant legal liability for self-driving automobile companies and medical institutions. Interactive visualizations of DNNs may provide both expert and non-expert users the insight and intuition required to more optimally architect DNNs, as well as foster more confidence in their inner workings or reveal pitfalls to which solutions may be engineered. Based upon a review of current deep neural network visualization techniques, we have developed an interactive comparative visualization of DNN layer parameters and their distributions as they change with time-steps (epochs) in the DNN training sequence.

1 INTRODUCTION

There are 37 moves on average in a game of chess, and the maximum number of moves can reach 277. In this sequence of decisions, to which move or set of moves can the final outcome, a victory or loss, be attributed? In a more complex context, of systems biology for example, there are indiscernibly many sequences of cascading metabolic events that culminate into a final outcome. To which event in the process is the final outcome attributed? These are different forms of the long-standing Credit Assignment Problem, formally broached by Marvin Minksy in his 1961 paper, "Steps toward Artificial Intelligence" [12].

The credit assignment problem can be formulated as the basis of any complex phenomenon requiring pattern recognition, and adaptive learning systems have long been pursued as potential solutions to this problem. In 1966, only five years after Minksy's publication, the first deep, feed-forward multilayer perceptron model was developed, initiating the long march toward a solution to the credit assignment problem, cristening the dawn of Deep Neural Networks (DNNs) [8].

DNNs have since expanded beyond the confines of academic theoretic and into myriad, every-day applications for expert and non-expert users alike. These applications include automatic speech recognition and natural language processing, image recognition and automated captioning, drug discovery and toxicology, biomedical informatics, intelligent gaming systems, recommender systems, cyber security, postal systems, and self-driving automobiles. Despite their wide-spread use and burgeoning success, DNNs are not without significant issues, which includes overfitting and prohibitive computational complexity. We will briefly survey these issues below, along with suggesting a potential visualization solution that may foster insight. The proliferation of DNNs is due to their exceptional ability to model complex, non-linear relationships. In essence, they

*e-mail: michael.iuzzolino@colorado.edu

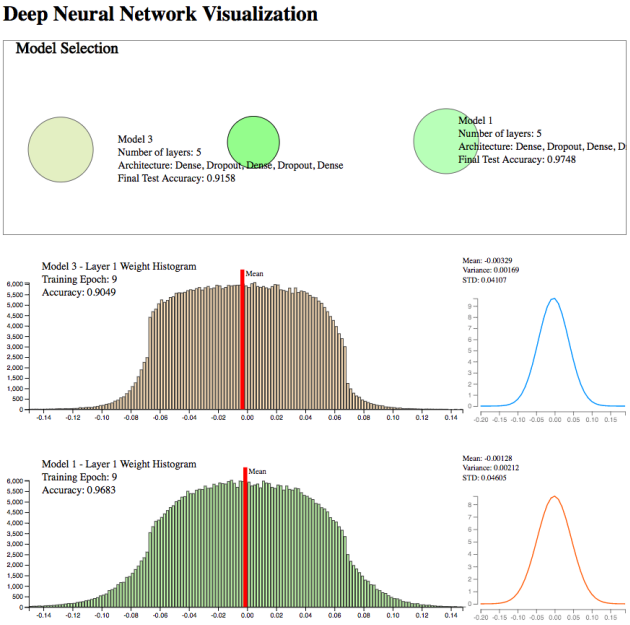

Figure 1: DNN Visualization interface design

are making great headway in the credit assignment problem. Their ability to accomplish this complex modeling task is due to the "deep" part of the neural network (NN). The architecture of a vanilla DNN is predicated on that of a "shallow" Artificial Neural Network, with large expansions to the number of hidden layers and number of nodes within each hidden layer. Consequently, the number of parameters can easily extend into the millions. This results in two primary problems that DNNs face: 1. overfitting, and, 2. computational complexity. The problem of overfitting has been largely addressed by regularization techniques such as Dropout [15], and the problem of computational complexity has been attenuated via the utilization of GPUs and parallel processing. Novel system architectures that diverge from von Neumann architecture, such as IMB's Brain-inspired chip, TrueNorth - also referred to as neuromorphic architectures - hold some promise for overcoming the computational complexity bottlenecks [3, 7].

Although these deep architectures are precisely what allow for complex, non-linear modeling, they are also the primary cause of DNNs being described as "black boxes." In other words, their inner workings lack transparency and are difficult, if not impossible, to interpret, understand, and optimize. This results in intellectual dissatisfaction, but also in legal liabilities, inaccessibility to non-expert users and a consequent restriction on domain usages, a lengthy and inefficient trial-and-error design process, and

2 RELATED WORK

Modern DNN architectures are as diverse as their applications. These architectures include Convolutional Neural Networks (CNNs), recurrent neural networks (RNNs), Long Term Short Memory (LTSM), Deep Belief Networks (DBNs) and Convolutional DBNs, Deep Boltzman Machines, and Deep Q-networks. DNNs are dif-

ficult to visualize for a number of reasons. First, as their name suggests, they are deep. The term 'deep' means that the architecture consists of many layers of nodes (hidden layers) between the input and output layers of the network. For example, a common toy CNN that is trained on CIFAR-10 [10], a dataset consisting of 60,000 32x32 color images in 10 mutually exclusive classes, typically has 6 convolution layers resulting in a total number of parameters on the order of 700,000. Put another way, the dimensionality of the datasets are extraordinary and not readily distilled into comprehensible visualizations. Secondly, once the information propagates beyond the first layer of the network and into the hidden layers with non-linear activation functions - typically RELU - a non-linear model is established along with the accompanying complexity and lack of interpretability. This non-linearity and high dimensionality of these models, along with massive data sets and computing power, is what imbues them with such staggering power as approximators and pattern detectors.

2.1 Heat Maps

2.1.1 Layer-wise Relevance Propagation

A number of innovative approaches have been taken to visualize deep neural networks (DNNs) with some modicum of success. One recent approach is a method called Layer-wise Relevance Propagation (LRP) [1]. LRP, stemming from a method called Deep Taylor Decomposition [13], computes scores for an image's pixels and groupings of pixels that most strongly contribute to the classification and reconstructs the input image as a heat map based upon the scores. This yields a visualization in terms of the original image and the features that are being captured by the network and utilized most heavily for classification. Although this sheds some transparency on the network's functionality and it being quickly picked up by the cognitive science community for applications to EEG analysis, it does not reveal the full extent to how the network is making decisions and at what level of the information processing it is occurring. In addition, the method's results are highly sensitive to the algorithm's parameters, ϵ and β, and an optimal tradeoff between numerical stability of the decomposition and meaning of the heatmap has yet to be achieved.

2.1.2 Hinton Diagrams

Similar to the LRP heatmaps, the Hinton diagram visualizes the parameters of a neural network layer as a grid of squares in various sizes and colors [2, 6]. The parameters of a neural network are the weights of the links (also known as edges or synapses) connecting the nodes of the layers; these weights take on real-valued numbers and can consequently be encoded into a visualization that takes advantage of both color, size, and spatial channels. A Hinton diagram maps the magnitude of the parameter onto the size channel, the sign of the parameter onto the color channel - white for positive values and black for negative, and spatially arranges the marks such that the rows correspond to the nodes of the sending layer and the columns correspond to the nodes of the receiving layer of the connection matrix. Although this is a first attempt at visualizing the internal layers of the network, better approaches have been developed that we will talk about in the next section.

2.2 Intermediate Layer Visualization

An alternative approach to heat mapping is a tool that dynamically visualizes the activations produced at each layer of a trained CNN as it processes an image or video [18]. Accompanying this visualization is a second visualization tool that visualizes features at each layer of the DNN. These interactive tools have shed additional transparency on CNNs, revealing that deeper CNN layers tend towards detection of 'local', specific features of an image; for example, the wheel of a car of the face of a person. Although the details are beyond the scope of this paper, these locality revelations shed additional insight onto two other opaque characteristics of neural networks - transfer learning and the 'hacking' of discriminative networks via generative models that produce structured noise [4, 5, 16, 17].

Other approaches have visualized the intermediate feature layers and the operation of the classifier, but were used with slightly different intentions. For example, Zeiler et al. used their intermediate feature layer visualization tools in a diagnostic role to make comparisons between models, as well as ablation studies to obtain performance measures from different model layers [19]. The visual comparison of models may prove highly effective in fostering another aspect of CNN intuition and understanding via intrinsically strong human learning mechanisms that utilize comparisons and contrasts as a way of cultivating knowledge about the world and it's myriad objects. Yet another group approached deep network visualizations from another angle: hierarchical rectangle packing algorithms and a matrix reordering algorithm to show the derived features of a neuron cluster [11]. As alleged by the authors, this approach facilitates a better understanding of CNNs and therefore, it supports the machine learning experts during the refinement of the CNN architecture towards the improvement of performance – as opposed to only the trial-and-error strategy.

2.3 Interactive Neural Network Visualization

Lastly, Daniel Smilkov and Shan Carter, inspired by Andrew Karpathy's ConvnetJS demo [9], have approached the visualization of deep neural networks pedagogically with their TensorFlow (TF) Playground [14]. TF Playground enables a user to adjust the hyper parameters of customizable DNN in-browser, loading various datasets, apply noise, set activation functions and learning rates, and a number of other features, all whilst enabling the user to then train their customized network and view the results in real-time. This approach provides users, expert and non-expert alike, with valuable insight into the various architectures and parameters and how the functionality of the model is effected. Using TF Playground as conceptual inspiration, we have developed a tool that enables a user to explore CNNs in a similar manner to how TF Playground explores DNNs, which we will describe in detail below.

3 Approach

Three DNN models were trained on the MNIST dataset for 15 epochs each to establish the backend of the visualization. The visualization interface is segmented into two parts. First, the top frame consists of a selection panel in which the available models are loaded. The models are represented by circle marks with a coloring that corresponds to the final training accuracy achieved after the 15 epochs. Hovering over a model will present a user with that DNNs layer numbers, its architecture, and its final test accuracy. The user may click any model, selecting up to a total of two for comparison.

Once a user selects a model, the second frame appears along with a slider. The frame contains the parameter distribution of the selected DNN's first layer weights. The color of the distribution corresponds to the model's test accuracy at the selected epoch. The slider enables the user to dynamically shift through the weight distributions across the range of training epochs. The mean of the distribution is represented by the vertical red line. Further, to the right of the bar plot is a statistical description and visualization of the layer distribution, which includes mean, variances, and standard deviation.

If a user selects a second model, it is aligned below the first model and will offer a visualization configuration as described above. This enables the user to investigate and compare various DNN models with different architectures, hyper-parameter settings, and training paradigms by visually comparing the layer distributions dynamically across training epochs and statistical characteristics.

ACKNOWLEDGMENTS

The author wishes to thank Dr. Danielle Szafir.

REFERENCES

[1] A. Binder, S. Bach, G. Montavon, K.-R. Müller, and W. Samek. Layer-wise relevance propagation for deep neural network architectures. In *Information Science and Applications (ICISA) 2016*, pp. 913–922. Springer, 2016.

[2] F. J. Bremner, S. J. Gotts, and D. L. Denham. Hinton diagrams: Viewing connection strengths in neural networks. *Behavior Research Methods*, 26(2):215–218, 1994.

[3] P. U. Diehl, B. U. Pedroni, A. Cassidy, P. Merolla, E. Neftci, and G. Zarrella. Truehappiness: Neuromorphic emotion recognition on truenorth. In *Neural Networks (IJCNN), 2016 International Joint Conference on*, pp. 4278–4285. IEEE, 2016.

[4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

[5] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[6] G. E. Hinton and T. Shallice. Lesioning an attractor network: Investigations of acquired dyslexia. *Psychological review*, 98(1):74, 1991.

[7] J. Hsu. Ibm's new brain [news]. *IEEE Spectrum*, 51(10):17–19, 2014.

[8] A. G. Ivakhnenko and V. G. Lapa. *Cybernetic predicting devices*, vol. No. TR-EE66-5. PURDUE UNIV LAFAYETTE IND SCHOOL OF ELECTRICAL ENGINEERING, 1966.

[9] A. Karpathy. Convnetjs, 2012.

[10] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images, 2009.

[11] 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*, 23(1):91–100, 2017.

[12] M. Minsky. Steps toward artificial intelligence. *Proceedings of the IRE*, 49(1):8–30, 1961.

[13] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, and K.-R. Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65:211–222, 2017.

[14] D. Smilkov and S. Carter. Tensorflow playground, 2016.

[15] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[16] C. Szegedy, A. Toshev, and D. Erhan. Deep neural networks for object detection. In *Advances in Neural Information Processing Systems*, pp. 2553–2561, 2013.

[17] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[18] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015.

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