TensorViz: Visualizing the Training of Convolutional Neural Network Using Paraview

Extended Abstract, LA-UR-17-26660

Xinyu Chen
University of New Mexico
Albuquerque, New Mexico
xychen@cs.unm.edu

Qiang Guan
Los Alamos National Laboratory
Los Alamos, New Mexico
qguang@lanl.gov

Xin Liang
University of California, Riverside
Riverside, California
xlian007@ucr.edu

Li-Ta Lo
Los Alamos National Laboratory
Los Alamos, New Mexico

Simon Su
US Army Research Laboratory
Adelphi, Maryland

Trilce Estrada
University of New Mexico
Albuquerque, New Mexico

James Ahrens
Los Alamos National Laboratory
Los Alamos, New Mexico

ABSTRACT

Deep Convolutional Networks have been very successful in visual recognition tasks recently. Previous works visualize learned features at different layers o help people to understand how CNNs learn visual recognition tasks. However they do not help to accelerate the training process. We use Paraview to provides both qualitative and quantitative visualization that help understand the learning procedure, tune the learning parameters and direct merging and pruning of neural networks.

CCS CONCEPTS

• **Human-centered computing → Information visualization**;
• **Computing methodologies → Neural networks**; *Online learning settings*;

KEYWORDS

Visualization, Convolutional Networks, Paraview

ACM Reference format:
Xinyu Chen, Qiang Guan, Xin Liang, Li-Ta Lo, Simon Su, Trilce Estrada, and James Ahrens. 2017. TensorViz: Visualizing the Training of Convolutional Neural Network Using Paraview. In *Proceedings of Super Computing, Denver, Colorado USA, November 2017 (SC17),* 2 pages.
https://doi.org/10.475/123_4

1 INTRODUCTION

Deep Convolutional Networks (AlexNet [2] etc.)have been very successful in visual recognition and natural language processing tasks. Lots of previous works aimed to help people to understand why those biology-inspired networks achieved such good performances. Deconvnet[5], Guided propagation[3] and a comprehensive visualization tool box[4] aimed to help people to understand the learned features at different layers of the networks. These works greatly support the biology origin and provided understanding of how convolutional networks emulate visual recognition tasks.

Post-pruning the networks have been proposed to address energy and memory concerns Denton et al. [1]. Tensorboard provides a lot of visualization tools for networks built from tensorflow framework. But most of the visualizations and analytics it provides are post-training statistics. The on-line analysis and tuning during training still largely remains in black-boxes.

We us Paraview and Matplotlib to study the network as a dynamic system and treat its learning process as the evolution of parameters. By visualization of the evolution of network's parameters, we find out similarities between convolution filters. We then merge and prune the redundant filters during the training. This is an attempt to use visualization as a facility to find optimal hyper-parameters of deep convolutional networks.

2 THE METHOD AND DATASET

We study networks as dynamic systems. We use Paraview to visualize weights,gradients,and activations in different layers.

Paraview is an open-source data analysis and visualization application which is more often used in large scientific simulation. It has the capabilities of provide interactive and in-situ analysis on extremely large datasets using distributed memory computing resources. For the purpose of proof of concept, we use a network similar to the LeNet-5 to learn the MNIST dataset(Fig. 1).

3 EXPERIMENT RESULTS

We use Paraview 5.2.0, Python Matplotlib for the visualization. The convolution network is built in tensorflow. All experiments were carried out on Darwin cluster in the Los Alamos National Laboratory.

[1]The publication has been assigned the LANL identifier LA-UR-17-26748.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
SC17, November 2017, Denver, Colorado USA
© 2017 Copyright held by the owner/author(s).
ACM ISBN 123-4567-24-567/08/06.
https://doi.org/10.475/123_4

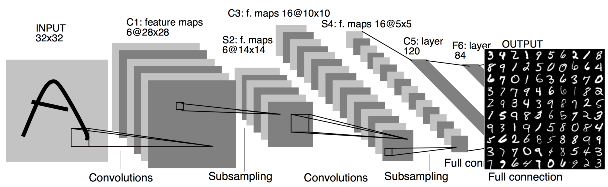

Figure 1: (Left)The LeNet-5. (Right)The MNIST dataset: images of hand written digits.

3.1 Visualize Weights

Paraview support the display of time-series and in-situ image rendering(Fig. 2). The evolution of weights can also be visualized in animations. In cases the learning rate is too high, it is easy to see weights get stuck within a short period of time.

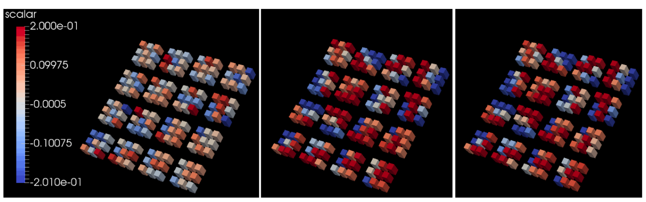

Figure 2: Convolution filters in layer-1 at 0, 4k and 8k steps. Colored boxes represent weights. Red is more positive, blue is more negative.

3.2 Visualize L2norms

We concatenate weights into long *weight vectors* in each layer and compute their L2norms to see the distance between initial positions and local optimal. It looks like a high dimensional random walk.

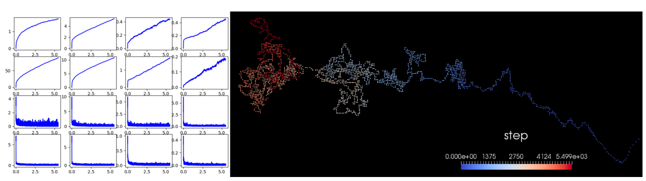

Figure 3: Left: convolution weights,fully connected weights and their gradients. Right: Visualization of first 2 dimensions of convolution weights.

3.3 Visualize Activations

Fig.4 shows similar activations in both convolution layers. This indicates redundancies of convolution filters. The heatmaps illustrates Pearsons' correlation between convolution filters.

3.4 Filter Pruning

Fig.5 shows pruning of similar filters. The accuracy of pruned networks remains the same as the original network(99.2%).

4 CONCLUSIONS

We use Paraview to visualize and analyze the training of deep convolution networks. This helps us to reduce parameters during training.

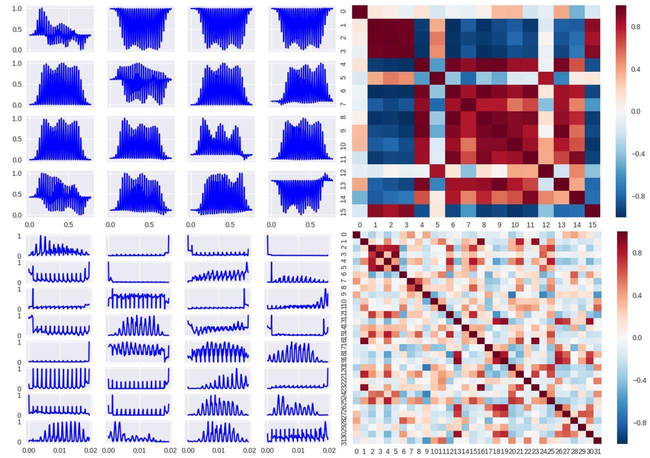

Figure 4: Left: Activation layer-1 and 2. Right: Heatmap of activations' correlation in layer-1 and 2.

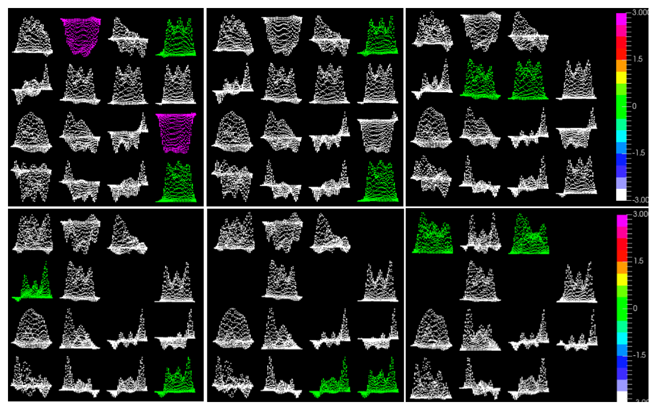

Figure 5: We merged similar filters during training. Colors indicate similarities. After 20 epochs, 16 filters reduced to 12.

It is beneficial to build an interactive mechanism to facilitate the training of more complex networks in the future.

ACKNOWLEDGMENTS

The authors would like to thank the Data Science at Scale summer school in the Los Alamos National Laboratory.

REFERENCES

[1] Emily L Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus. 2014. Exploiting linear structure within convolutional networks for efficient evaluation. In *Advances in Neural Information Processing Systems*. 1269–1277.
[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.
[3] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. 2014. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806* (2014).
[4] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. 2015. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579* (2015).
[5] Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*. Springer, 818–833.