Visualizing a Convolutional Neural Network

Mahdi Ghodsi, m.ghodsi1990@gmail.com

Hooman Shariati, hooman.shariati@alumni.ubc.ca

Recently, Deep Neural Network (DNN) has gained much attention due to its success in improving tasks such as image classification and speech recognition. Amongst different DNN approaches Convolution Neural Networks (CNNs) is extremely popular in particular due to their outstanding capacity to utilize spatial information. We have taken a course (EECE592) that covers DNNs. Currently, there has been a shift on applying bioinformatics data to CNNs[1]. While much success have been achieved in the biomedical imaging domain [2] using CNNs, there is currently ongoing research in other bioinformatics domains such as have been successful in many application and have been applied to many fields such as genomic sequence [3] motifs and EEG[4].

Despite the encouraging success of CNNs, many still see CNNs as a promising black box with little insight into the behavior of internal components of CNNs. This fact leaves many researchers to relying on try-and error to achieve better performance and fine-tune the hyper parameters involved. Large modern neural networks are even harder to study because of their size; for example, understanding the widely used AlexNet DNN involves making sense of the values taken by the 60 million trained network parameters [5,6].

Recently, visualizing DNNs in particular CNNs has resulted in many publications. The most cited and basis of recent approaches are based on a paper by Zeiler and Ferfus et al. in which they propose a multi-layered deconvolutional Network (deconvnet) to project the feature activations back to the input pixel space for a trained network. This technique reveals the input stimuli that excite individual feature maps at any layer in the model [5]. The second most cited paper by Yosinski et al. improves this method by proposing several new regularization methods that combine to produce more human interpretable visualizations [6]. Recently, Google's Tensorflow (an open source software library for machine learning) is complemented by a visualization tool called Tensorboard [7]. In addition to the above tools and many more scholarly published papers, there are many publicly available online tools that are worth investigating [8].

This project's main goal is to combine the previous works in a complete interactive visualization tool that focuses on visualizing the operation of internal components of a given neural network. This project as requested by Mahdi's supervisor, focuses on delivering a tool that resembles the familiar architecture that is shown in figure 1. The requirements for this tool are derived from a study that surveyed researchers and identified the main tasks for a visualization tool [9]. A few of these tasks combined with what Mahdi's supervisor's requests are selected and we aim to deliver them within our final solution. These tasks are:

1. Overall view of the architecture and network depth
2. Overview of learned features of neurons
3. Interactively exploring multiple facets of neurons (learned features, activation, numerical values)
4. The input to the network and the effect of applying convolution through out the system

In order to deliver a tool that focuses on the above tasks, we are hoping to use a CNN, possibly fed by MNIST hand written digital database, that is publically available and produce the required data by modifying/adding to the available code. The data produced will be a set of weights for each neuron within each layer of the network. The network is consisted of different layers and depending on the layer and the neuron the number of learnable weights differ. For instance a fully connected layer for a typical MNIST fed CNN has a total approximately 128 neurons. Each of these neurons takes the output of previous layer that in this case are 1764 values. These values are then processed and a single output is produced by each neuron. Our task will be investigating previous publications to realize the state of art methods in producing a meaningful visualization.

This is slightly different for visualizing convolutional layers, as each layer will consist of a set of convolutional filters (for instance 5x5 filters) that allows us to visualize a set of weights as one filter. For instance a convolutional layer may consists of 16 filters of 5x5 sized, allowing us to visualize 16 boxes rather than 16x5x5=375 values.

Below is a scenario for user "X", who has just trained a CNN and wants to visualize the network:

X has to run a forward propagation of one input sample and include the instructions given by us for how to do so and then the tool will be ready to use. The tool aggregates all the detailed information and illustrates an overview of the architecture such as the one given in figure 1. Then X can choose to explore the filters in first convolutional layer. X needs to click on the appropriate layer to see more details. The details could be filters shown. If the X wants to see the effect of that filter on the input image, there will be a button/slider to change the detail shown on demand. The connection between that filter and all the other nodes/filters are shown. For any other information about each layer X needs to follow the same logic of click to view more details.

This tool will help researchers to get some insight about how each individual neuron is activated or information in regards to the output of a set of neurons that construct one full layer of the Neural Network architecture. The scope of this project is limited to a trained CNN to avoid complexity of running dynamic analysis of the network which requires heavy computation and on the other hand it is not as easy to interpret for a human user since each training set contains thousands of data points . Visualizing a trained CNN may give a domain expert enough intuition to analyze and improve the performance of original network.

The final product will be delivered by the end of April as a tool that could visualize a trained CNN of our choice. However, it will be complemented with enough documentation usable for other researchers who whish to visualize a given CNN.

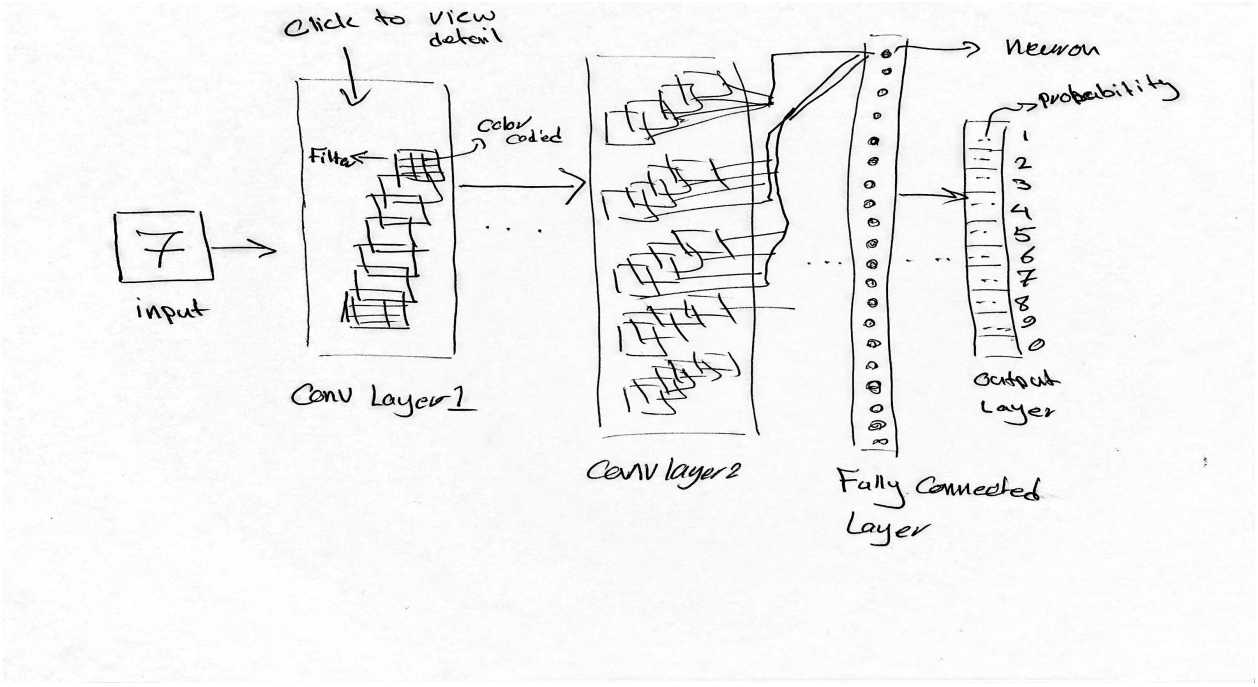

Figure 1- Overview of CNN to visualize

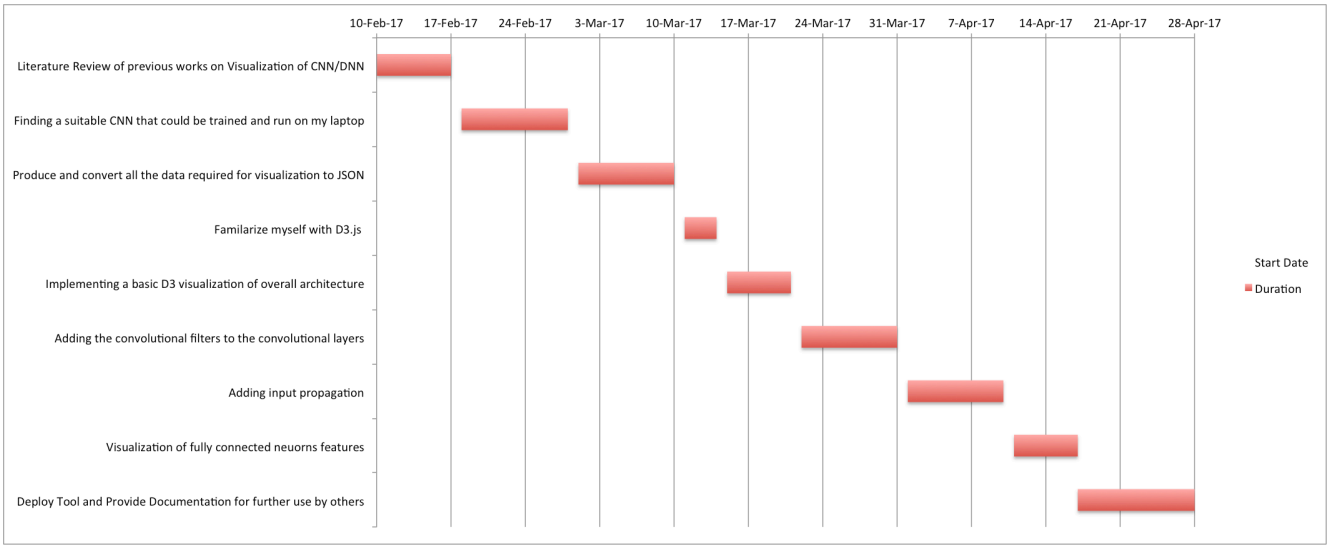

Figure 2- Project Milestone

References:

[1] S. Min, B. Lee, and S. Yoon, “Deep learning in bioinformatics,” Briefings in Bioinformatics, 2016.

[2] Roth, Holger R., Le Lu, Amal Farag, Hoo-Chang Shin, Jiamin Liu, Evrim B. Turkbey, and Ronald M. Summers. "DeepOrgan: Multi-level Deep Convolutional Networks for Automated Pancreas Segmentation." Lecture Notes in Computer Science Medical Image Computing and Computer-Assisted Intervention -- MICCAI 2015 (2015): 556-64. Web.

[3] J. Lanchantin, R. Singh, B. Wang, and Y. Qi, “Deep Motif Dashboard: Visualizing And Understanding Genomic Sequences Using Deep Neural Networks,” Biocomputing 2017, 2016.

[4] P. Bashivan, I. Rish , M. Yeasin, N. Codella, “Learning Representation From EEG With Deep Recurrent-Convolution Neural Netwokrs,” Preceding 5th International Conference on Learning Representations(ICLR), 2016.

[5] Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." *European conference on computer vision*. Springer International Publishing, 2014.

[6] Yosinski, Jason, et al. "Understanding neural networks through deep visualization." *arXiv preprint arXiv:1506.06579* (2015).

[7] https://www.tensorflow.org/get_started/summaries_and_tensorboard

[8] https://handong1587.github.io/deep_learning/2015/10/09/visulize-cnn.html

[9] Liu, Mengchen, et al. "Towards better analysis of deep convolutional neural networks." *IEEE Transactions on Visualization and Computer Graphics* 23.1 (2017): 91-100.