Visualizing Information in Deep Neural Networks Receiving Competitive Stimuli

Henrique U. Gobbi[1], Marco A. P. Idiart[2]

Universidade Federal do Rio Grande do Sul

[1]Informatics Institute, [2]Physics Department

Introduction

Deep Neural Networks (DNNs) exhibit significant parallels with the hierarchical organization of representations in the primate visual system. However, their feed-forward architecture, where all information in a scene is processed simultaneously, is unlikely to accurately reflect reality [1]. To push towards more realistic models, we designed an experiment where a DNN is presented with two competing items, one more salient than the other, placed in each of the network's non-overlapping receptive fields. The items used are instances of the MNIST digit dataset, and the DNN's objective is to identify the more salient digit. We subsequently developed visualization tools capable of tracking the flow of informaiton across the layer, as well as comparing how the different stimuli are represented and interact with each other using cosine similarity. In this study, we introduce our novel tool designed for visualizing information flow within networks. Additionally, we present results obtained from networks with different architectures, subjected to the same training strategy.

Methods

DNN architecture and training

The DNNs used in this study follow a similar structure. Each of them have two inputs (receptive fields, or RFs), followed by a sequence of dense layers for each RF. The layers are eventually concatenated, and followed by more dense layers, now containing information about both RFs. For the output layer, all of them have a 10 neuron dense layer, one for each possible digit. The difference between the DNNs is the number of dense layers: each of them have n dense layers, of which $\lfloor n/2 \rfloor$ are placed before the concatenation and the remaining are placed after the concatenation.

For both training and testing sets, instances of the MNIST dataset were randomly paired, and a random visual field was chosen to be more salient than the other by increasing or decreasing the value of the pixels in the image. The instance was then labeled by the more salient digit. Figures 1 and 2 show the network's structure as well as input type.

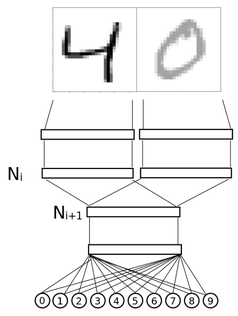

Figure 1: DNN basic architecture.

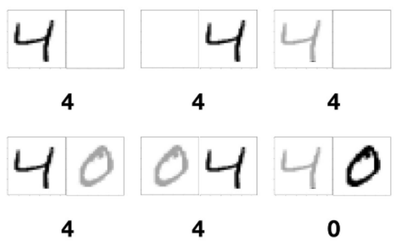

Figure 2: DNN input types.

Visualization

In order to visualize the flow of information through a trained DNN, the entire network was plotted: every neuron in each layer and the connections of these neurons between each layer. Given an input to the network, all the activations were computed and the color of each neuron is determined by the value of its activation, normalized with respect to all neurons in its layer. The connections of neurons between layers were colored by the value of their weight. If the weight was below a certain threshold, it wasn't displayed.

We also developed a technique to get an idea of the information in each dense layer. Called *output heads (OHs)*, they allow us to see the k digits that the network perceives in each layer, that is, the information contained in each layer. This was acheived by, for each dense layer, copying the DNN up to that layer and freezing its parameters. Then, an output layer was attached to this new network, with trainable weights. This model was then trained on the same dataset as the original one, making it possible to know the underlying digit contained in each dense layer of the original DNN.

Similarity calculations

Every network used has a set of single receptive field layers (SRFs), that is, the layers in each RF before the concatenation, carrying information about a single digit, and a set of double receptive fields (DRFs), which are the layers after, and including, concatenation, carrying information about two digits.

To compare the representation, in a layer L of the DNN, of a digit placed in one of the RFs with the representation of another digit placed on the same or the other RF, the following algorithm was developed:

- The DNN was presented with m instances of paired digits, obtaining every possible arrangement of inputs m times.
- The *prototype* of a digit pair $d1$, $d2$ in a layer L was computed taking the average of the m activation vectors those pairs produce in layer L.
- For SRF layers, we used the *prototypes* of the digits in L to compute the cosine similarity matrix CSM for the digits in that layer, comparing every combination of two digits.
- For DRF layers, we computed the CSM of the layer using the *prototypes* of the digits in that layer, comparing every combination of pairs of digits.

Orthogonality

Also, using the CSMs (C), we calculated the orthogonality for a given layer L, considering left RFs, right RFs and cross RFs (containing information about two RFs) using the following equations: $O_{L,left} = 1 - \bar{C}_{L,left}$, $O_{L,right} = 1 - \bar{C}_{L,right}$ and $O_{L,cross} = \bar{C}_{L,cross} - (\bar{C}_{L,left} + \bar{C}_{L,right})$

Results

Visualization tool

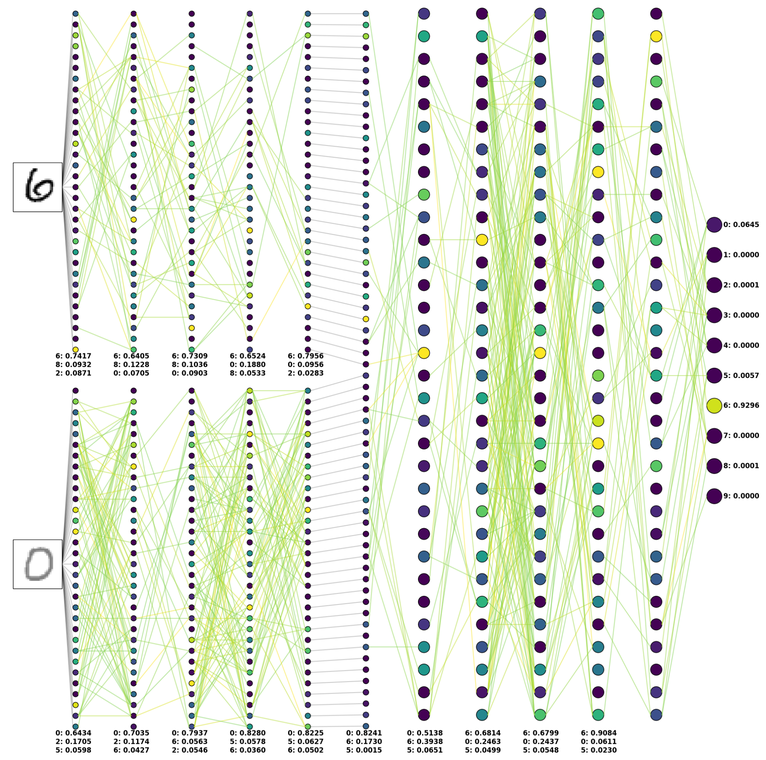

Figure 3: Visualization of a DNN with 10 dense layers, 32 neurons per layer, 3 digits in the *output heads*, showing the activation of each neuron given the input and the intensity of the weights in each connection. Note how the information about the digit zero is present across the entire network, even the output.

Cosine similarity and orthogonality

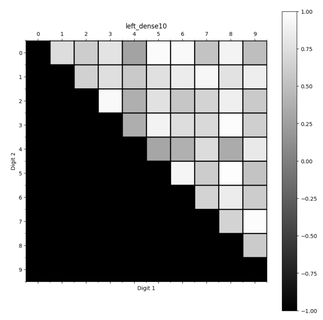

Figure 4: Cosine similarity matrix of a left SRF layer in a DRF DNN. It is a comparison of the representation of each digit with each other in this given layer. It is possible to see that the network hasn't yet learned the proper distinctions between digits, and is still confused by certain pairs of digits.

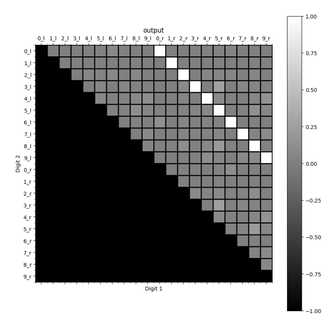

Figure 5: Cosine similarity matrix of a DRF layer in a DRF DNN, namely, its output layer. It is a comparison of the representation of each digit with each other in this given layer placed in each receptive field. We can see that the same digit placed in each RF is similar, also digits that look similar visually to us, e.g. 5 and 8, are slightly similar.

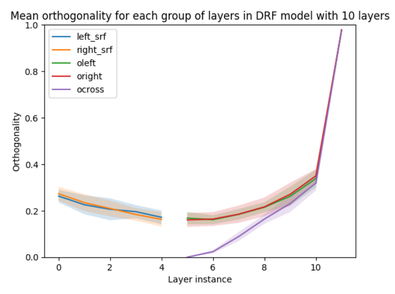

Figure 6: Graph of the mean and standard deviation orthogonality value for each group of layers in DRF network with 10 dense layers.

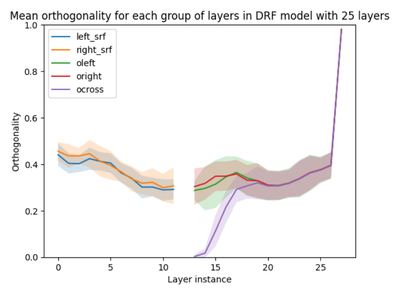

Figure 7: Graph of the mean and standard deviation orthogonality value for each group of layers in DRF network with 25 dense layers.

Conclusions

With the construction of the viewer, it was possible to observe some interesting behaviors of the network. For example, the orthogonality between representations does not necessarily increase with depth in the network, at least for dense models. The viewer will continue to be developed with the addition of other possible functionalities to monitor representations at each layer. For future work, we will explore other architectures beyond dense layers, such as CNNs, and we will work on introducing an oscillatory dynamic that allows the network to have representations of the content of both RFs being simultaneously represented in the DRF layers.

References

[1] Katharina Duecker, Marco Idiart, Marcel AJ van Gerven, and Ole Jensen. Oscillations in an artificial neural network convert competing inputs into a temporal code. *bioRxiv*, 2023.

Acknowledgements

This work has been funded by CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico.