Theoretical and computational aspects of expressive power of deep neural networks

Summer At ICERM 2020

The success of Deep Neural Network (DNN) originates from the richness of the learning function

$$F(v) = F_L(F_{L-1}(\cdots F_2(F_1(v)))) \tag{1}$$

associated with the network. Here $v \in \mathbb{R}^p$ is the input and F_i denotes the operation at the i^{th} (hidden) layer. DNN creates complex functions from simple ones by composition. We focus on DNNs that have piecewise linear activation functions, e.g. $\text{ReLU}(v) = v_+$. In this case, each layer of nodes slices the input space by a number of hyperplanes, thereby enriching the learning function by allowing it to be linear on each of the sliced pieces that are increasing in number and shrinking in size. The optimization procedure employed when we train the network is nothing but a systematic approach for determining this slicing so that the learning function matches with the data. That is, given data $\{v^i, y^i\}$, we aim to find function F^* in some function class such that

$$F^*(v^i) \approx y^i.$$

Back propagation executes chain rule for differentiating the network learning function $F(\cdot)$ (1) so we can use gradient-based approaches during the optimization for locating $F^*(\cdot)$.

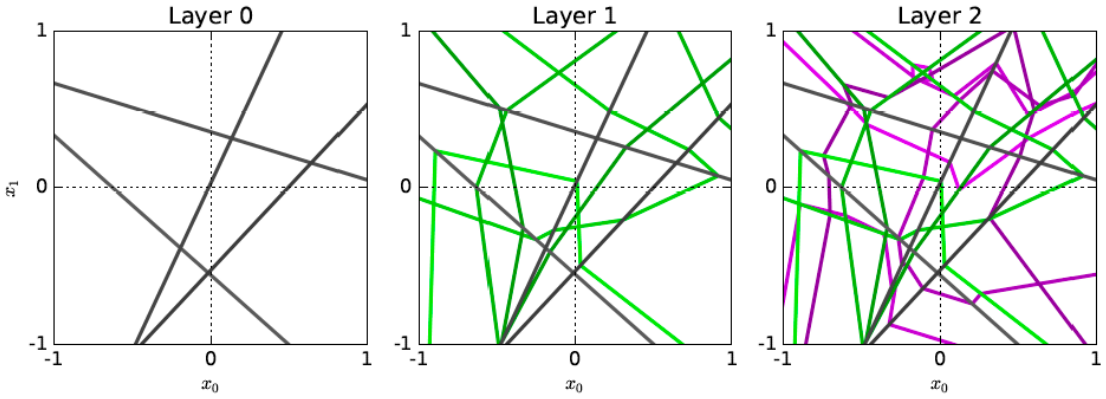

Figure 1: Picture from [5]. Here is a three hidden layer ReLU network, with $p = 2$ and four units in each layer. The left pane shows activations for the first layer. The center pane shows activation boundary lines corresponding to second hidden layer neurons, in green, bending at the boundaries at the left. The right pane adds the on/off boundaries for neurons in the third hidden layer, in purple, bending at the two sets of boundaries. This final set of convex polytopes corresponds to all activation patterns for this network (with its current set of weights) over the unit square, with each polytope representing a different linear function.

Figure 1 shows the slicing by a 3-layer network to a two-dimensional input space. It is noticeable that the resulting number of "linear regions" quickly, in fact, exponentially, increases as we add layers. Moreover, these regions are often irregular. This is drastically different from the way traditional numerical algorithms, e.g. Finite Element/difference Methods, discretize the input space, see Figure 2. These traditional methods, e.g. linear finite elements, are also adopting piecewise linear functions.

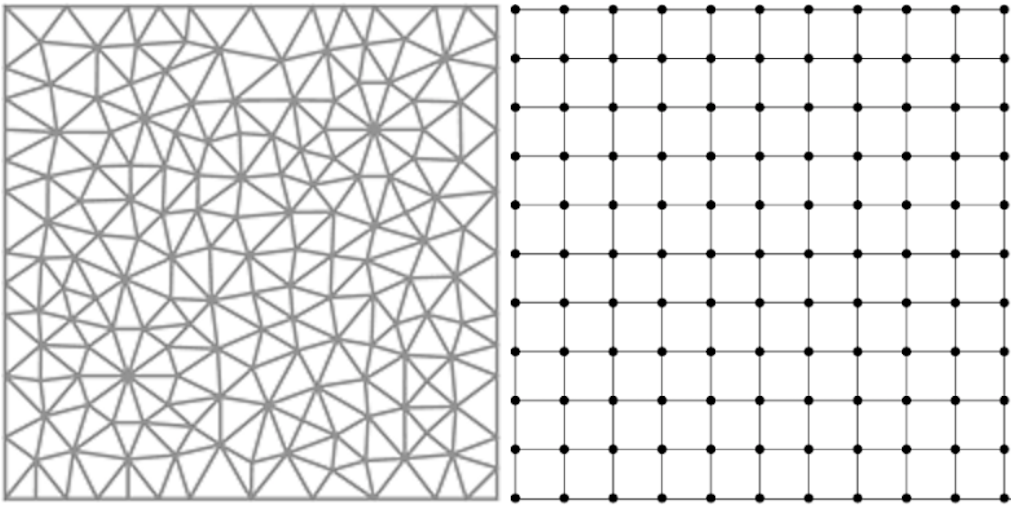

Figure 2: Possible grids for finite element (Left) or finite difference (right) methods.

Directions of project

The goals are theoretical investigation, practical verification and visualization of the "expressivity" of deep neural network function $F(v)$, and the comparison between a shallow ReLU DNN and linear finite elements. These are the **possible** steps/deliverables:

- A thorough combinatorial study of the number of linear regions for shallow and deep neural networks. Part VII of the book [6] and papers [3] [4] are good starting points.
- A visualization of the slicing, including when the training of a simple network is in action.
- A study of the activation patterns, trajectory lengths, and network stability. [5] is a good starting point.
- An investigation of the relationship between linear finite elements and a shallow ReLU neural network. Start with the relevant sections of [1].
- Study and extension of Section 3 of the paper [2] toward *A Neural Network Based Approach Towards Matrix Inversion.*

References

[1] Juncai He, Lin Li, Jinchao Xu, and Chunyue Zheng, *Relu deep neural networks and linear finite elements*, arXiv preprint arXiv:1807.03973 (2018).

[2] Gitta Kutyniok, Philipp Petersen, Mones Raslan, and Reinhold Schneider, *A theoretical analysis of deep neural networks and parametric pdes*, arXiv preprint arXiv:1904.00377 (2019).

[3] Guido F Montufar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio, *On the number of linear regions of deep neural networks*, Advances in neural information processing systems, 2014, pp. 2924–2932.

[4] Tomaso Poggio, Hrushikesh Mhaskar, Lorenzo Rosasco, Brando Miranda, and Qianli Liao, *Why and when can deep-but not shallow-networks avoid the curse of dimensionality: a review*, International Journal of Automation and Computing **14** (2017), no. 5, 503–519.

[5] Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl Dickstein, *On the expressive power of deep neural networks*, Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR. org, 2017, pp. 2847–2854.

[6] Gilbert Strang, *Linear algebra and learning from data*, Wellesley-Cambridge Press, 2019.