Deep ReLU Networks Have Surprisingly Few Activation Patterns

Boris Hanin, Facebook AI Research & David Rolnick, University of Pennsylvania

Summary by Sri Datta, 11th October 2019 (budaraju@kth.se)

Abstract

The authors of the paper suggest that realizing the full expressivity of deep neural networks may not be possible with the current methods in practice. The expressivity of a deep network is its ability to approximate a rich class of functions. The number of activation patterns of a ReLU network is one way to measure its expressivity. And the number of patterns or the expressivity of a network is believed to exponentially grow with depth. But the authors of this paper show that this is in fact proportional to the total number of neurons raised to the power of input dimensions and not to that of the depth. This holds both at the initialization and training of these deep networks.

Introduction/Main contribution

For a ReLU network *N,* given a vector *θ* of its parameters, computes a continuous and piecewise linear function *x → N (x; θ).* Each *θ* is associated with a partition of input space $\mathbb{R}^{n(in)}$ into activation regions. These activation regions are polytopes on which *N (x; θ)* computes a single linear function corresponding to a fixed activation pattern in the neurons of *N.* One of the main contributions of this paper is the upper bound for the expected number of activation regions in *N* as in the equation(1) below (Theorem 5)*.*

$$\frac{\#\text{activation regions of } \mathcal{N} \text{ that intersect } \mathcal{C}}{\mathrm{vol}(\mathcal{C})} \leq \frac{(T\#\text{neurons})^{n_{\text{in}}}}{n_{\text{in}}!}, \quad T > 0.$$

Where n_{in} is the input dimension and C is a cube in the input space $\mathbb{R}^{n(in)}$. When a neuron *z* in *N,* turns on and off frequently so that the value of pre-activation *z(x)* crosses the level of bias b_z many times, many activation patterns can be created if there is significant overlap between the range of pre-activations on the different activation regions and b_z to lie within that overlap. Figure 1 shows that the number of activation regions starts at around *(#neurons)*^{2}*/2* and decreases a bit before coming back instead of increasing exponentially.

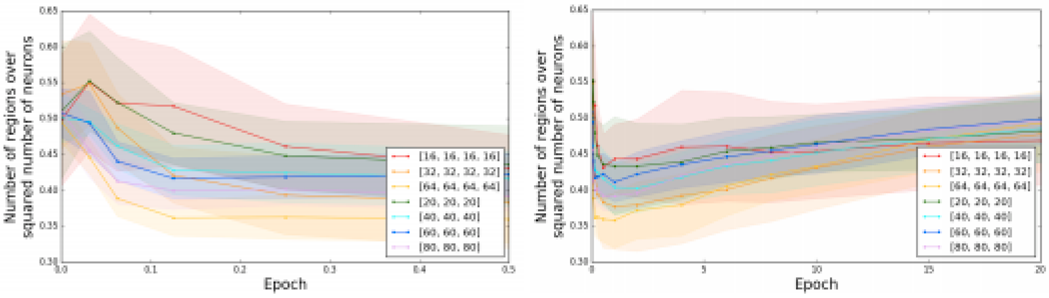

Figure 1: The average number of activation regions in a 2D cross-section of input space. [20,20,20] indicated 2 layer network each of width 20. The curves are averaged over 10 independent runs and regions are averaged over different 2D cross-sections for each run. Source: Paper

How to Think about Activation Regions

I cannot find a better way than just quoting the lemmas.

Lemma 1: “For every activation pattern *A* and any vector *θ* of trainable parameters for *N* each activation region *R(A; θ)* is convex”. Where linear regions are not necessarily convex.

Where “An activation pattern for N is an assignment to each neuron of a sign:

$A := \{a_z,\ z$ a neuron in $N\} \in \{-1, 1\}\ \#neurons$”

Lemma 2: Activation regions as connected components

activation regions (N , θ) = connected components ($\mathbb{R}^{n(\text{in})}$ / $\cup_{\text{neurons } z}$ $H_z(\theta)$).

$H_z(\theta)$ explained in (Lemma 4)

Lemma 3: More activation regions than linear regions.

“The number of linear regions in *N* at *θ* is always bounded above by the number of activation regions in *N* at *θ*.”

Lemma 4: $H_z(\theta)$ *as* bent hyperplanes

$H_z(\theta) := \{x \in \mathbb{R}^{n(in)} \mid z(x; \theta) = b_z\}$

For hyperplanes in general position, the total number of connected components from an arrangement of m hyperplanes in input space $\mathbb{R}^{n(in)}$ is constant.

(**Lemma 5** is theorem 5, the main contribution of the paper)

Lemma 6: “Rescaling all biases by the same constant, therefore, does not change the total number of activation regions.”

Maximizing the Number of Activation Regions

Memorization - The authors try to investigate if make a network learn more complex tasks will increase the number of regions during the training phase. They observe a slight increase in the regions with an increase in the amount of noise to be memorized as illustrated in figure 2. One way they added noise was to randomly assign labels to a fraction of MNIST samples. The authors also train a network to memorize random labels o 2D points. The regions increase slightly learning the hard task before it fails completely as shown in figure 3. However, despite many experiments, they could not observe a substantial increase in the regions other than by a constant factor. From the series of such experiments, hyperparameter choice does not seem to be a faction increasing the activation regions.

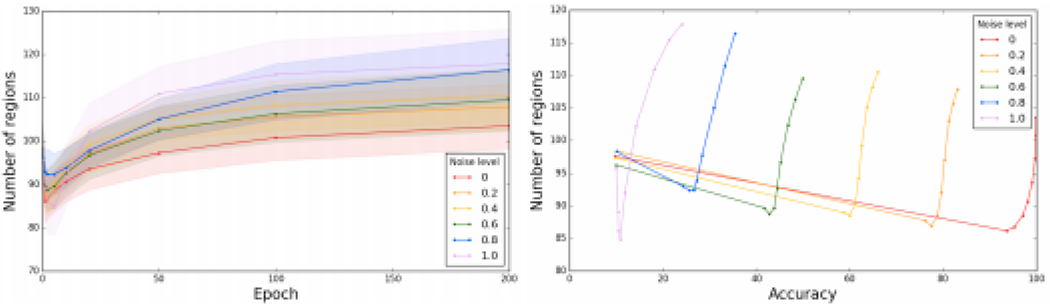

Figure 2: Depth 3, width 32 network trained on MNIST with varying levels of label corruption | Source: Paper

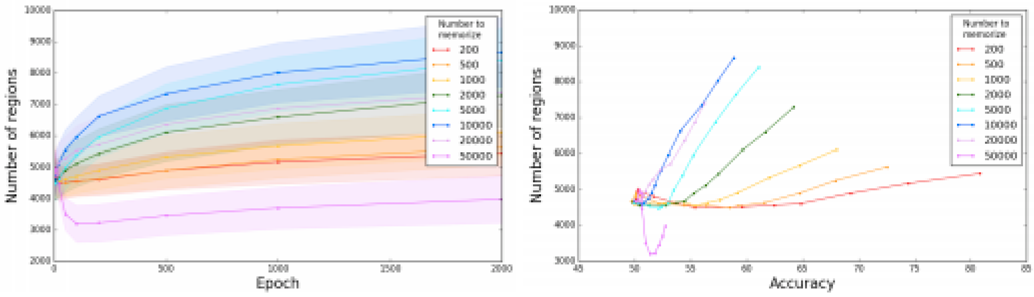

Figure 3: Depth 3, width 32 fully connected ReLU net to memorize random 2D points with binary label | Source: Paper

The effect of initialization - Here the authors try to investigate if the initialization of weights and biases affects the number of regions. We know by theorem 5 the upper bound varies with scaling the biases. However, using theorem 5 we only compute the local number of regions that are independent of biases scaling, except when they are initialized to 0. Maithra et al. 2017, suggest that there is an increase in the number of activation regions when the weights of a network are initialized with higher variance. Boris et al. 2018, show that an increase in weight variance increases the gradient norms. And for local activation regions, from theorem 5 the density of activation regions increases as gradient norms increase. Also scaling weights uniformly over the network is the same as scaling biases across each layer. And from lemma 6 that scaling biases uniformly do not affect the number of activation regions globally. Hence scaling weights should not affect the global region count.

TakeAway

One of the main take-aways is current deep ReLU networks learn a lot fewer activation regions than they theoretically can and depth does not contribute to the number of activation regions as expected by the total number of neurons does. Additionally, there is no exponential increase in the regions during training. There is an upper bound for the number of regions for a network following theorem 5.

Strong Points

A very detailed explanation of the idea. Extensive experimentation, averaging multiple independent training runs to provide high-quality results is very impressive.

Weak Points

A major part of the paper was for introducing various concepts and lemma to prove the upper bound. It would be more engaging to see more insights on maximizing the number of activation regions or giving more intuition of their main results.

References

Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl-Dickstein. On the expressive power of deep neural networks. In ICML, pages 2847–2854, 2017.

Boris Hanin and Mihai Nica. Products of many large random matrices and gradients in deep neural networks. Preprint arXiv:1812.05994, 2018.