Individual Activity 5

IA5008 Sistemas neuronales

Nisim Hurst Tarrab, A01012491

Monday 12 February 2018

Due Date: 14th of February 2018

Abstract

There are similarities between models used by neuroscientists and machine learning researchers even though in appearance they have opposed objectives. The features learned in one layer resembles to areas of the visual cortex of biological models. However, activations functions like logistic sigmoid and hyperbolic tangent fail to cope with sparse data, a domain in which a *rectifying linear unit* (ReLU) activation function excels without the need of unsupervised pre-training on deep networks (even though it can benefit from it).

The paper shows that ReLU activation function yields better performance for naturally sparse data in spite of non linearity at zero. This networks can be trained without unsupervised pre-training for deep networks (Glorot, Bordes, and Bengio 2011). Thus, the paper has demystified the notion that hard zeros are detrimental to gradient descent.

I think this paper is a cornerstone in understanding why deep networks have had a huge impact in nowadays machine learning.

Background

Neuroscience observations are:

1. Biological neurons encode information in sparse and distributed way, a trade-off between energy expenditure and richness of representation. All neurons that use sigmoid of *tanh* have a constant saturation regime of $\frac{1}{2}$ and this hurts gradient based optimization.
2. *Leaky integrate and Fire* (LIF) biological neuron model (Dayan and Abbott 2001), is given by:

$$f(I) = \begin{cases} \left[\tau \log\left(\frac{E+RI-V_r}{E+RI-V_{th}}\right) + t_{ref}\right]^{-1}, & \text{if } E + RI > V_{th} \\ 0, & \text{otherwise} \end{cases}$$

Figure 1 shows the traditional sigmoidal functions that are defined on zero. The *tanh* is preferable than *sigmoid* from an optimization standpoint but it forces antisymmetry around 0. Both functions are dissimilar to the LIF function shown in Figure 2. However, the ReLU function which is an approximation of the LIF function, can provide different activations for each neuron in sparse data.

It is convenient to use sparse representations because we get a better understanding of the principal components of the model. In concrete, the advantages are:

1. **Information disentangling** small changes in the input doesn't change the set of non-zero features.
2. **Efficient variable-size representations** it can vary the number of active neurons without greatly modifying the others.
3. **Linear separability** due to the high dimensionality, the information is more likely to be linearly separable.
4. **Distributed but sparse.** Dense distributed representations potentially encode more rich information.

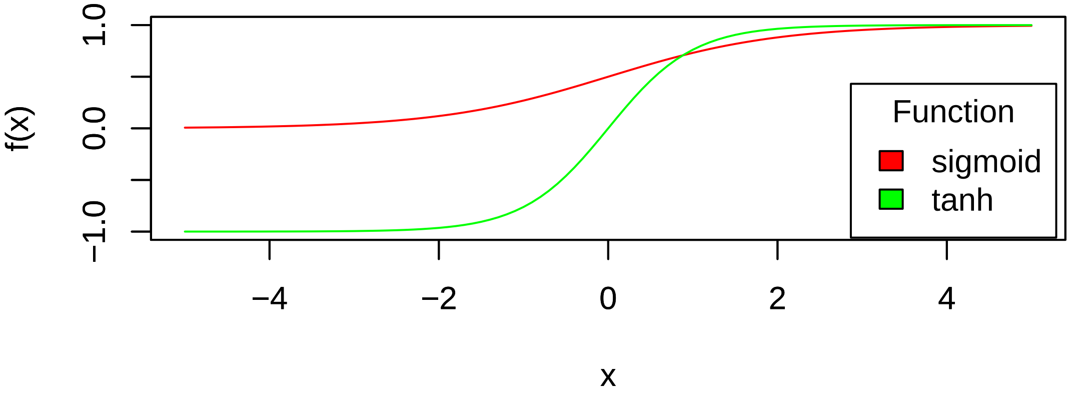

Figure 1: Traditional activation functions

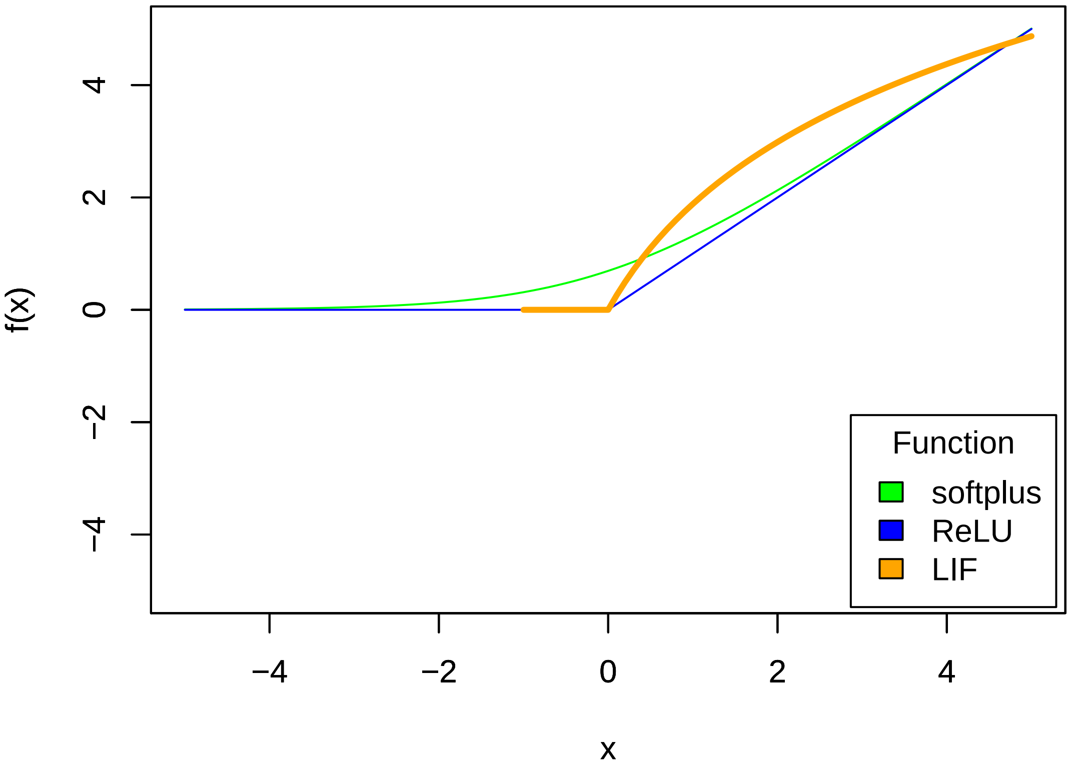

Figure 2: Proposed activation functions

Deep Rectified Networks

Apart from being more biologically similar (see Figure 2 where ReLU is more similar to LIF) , rectifier activation function has the advantage that maintains the data sparsity of 50% on hidden units and can increase through regularization. This sparsity allow us to see the model as an exponential number of linear models that share parameters reducing the vanishing gradient effect. Linear computations are cheaper than exponentials and sparsity can be used to find the singular values and reduce dimensionality.

However, the hard angle at 0 may hurt gradient back-propagation in comparison to the softplus function (Figure 2) in the hope of reduce computations. In spite of this, experimental results actually show hard zeros help supervise training maybe because the gradient may propagate by other non-zero paths.

Another problem is the unbounded nature of the function that can be easily solved with L1 regularization . More hidden units are needed to solve antisymmetric behavior. Finally one must take care of initialize the weights on the same scale.

Unsupervised Pre-trainning

Deep learning benefit from this activation function (Y. Bengio and Yoshua 2009), especially unsupervised (Erhan et al. 2010). However, the same disadvantages previously mentioned, i.e. hard 0 hurts gradient and the unbounded behavior. There are several solutions:

1. Use softplus in the reconstruction layer (along with cost).
2. Manually bound the values between 0 and 1 and use sigmoid for the reconstruction layer (along with cost).
3. Use linear activations before of after ReLU (along with the cost).
4. Use ReLU for the reconstruction layer, (allong with cost).

Experimental Study

This experimental study presents only the first two strategies because those where the ones that yield better results. ReLU is compared to $tanh$ and $softplus$ then is applied to sentiment analysis. The results on 4 datasets were (test error):

Table 1: With unsupervised pre-training

Neuron	MNIST	CIFAR10	NISTP	NORB
ReLU	1.2	49.96	32.86	16.46
Tanh	1.16	50.79	35.89	17.66
Softplus	1.17	49.52	33.27	19.19

Table 2: Without unsupervised pre-training

Neuron	MNIST	CIFAR10	NISTP	NORB
ReLU	1.43	50.86	32.64	16.40
Tanh	1.57	52.62	36.46	19.29
Softplus	1.77	53.20	35.48	17.68

The main observations were:

1. ReLU found the local minima equally well than softplus.
2. The greater improvement was found without unsupervised pre-training.
3. ReLU performed better in sparse datasets.
4. In the NORB dataset, ReLU perform better with more supervised examples for pre-training.

Sentiment Analysis

The words are treated like a sparse bag of words to predict *stars* ($\star\star\star\star\star$) restaurant rate (OpenTable). The model are stacked denoising auto-encoderswith 1 or 3 hidden layers of 5000 hidden units trained greedy layer by layer (Hinton, Osindero, and Teh 2006). Due to the sparse behavior of the deep network, the best result (less RMSE 0.746 ± 0.004) was obtained using 3 hidden layers of ReLU. Also testing with Amazon products 78.95% was achieved vs 73.72% obtained by Zhou et al. (2010).

Conclusion

We saw that this article written by Glorot, Bordes, and Bengio (2011) show a more plausible biological model using *ReLU*. The function was tested both on image and text sparse data with good results on both unsupervised pre-trained networks and not pre-trained. There are two main problems, zeros in the gradient and ill weight initialization but simple strategies to solve them were presented.

I think this break through is fundamental to grasp the way efficient biological natural beings learn and to close the gap applied to sparse datasets and networks such as deep networks and animal brains.

References

Bengio, Y., and Yoshua. 2009. "Learning Deep Architectures for AI." *Foundations and Trends in Machine Learning* 2 (1). Now Publishers Inc.: 1–127. doi:10.1561/2200000006.

Dayan, Peter., and L. F. Abbott. 2001. *Theoretical neuroscience : computational and mathematical modeling of neural systems*. Massachusetts Institute of Technology Press. https://dl.acm.org/citation.cfm?id=1205781.

Erhan, Dumitru, Yoshua Bengio, Aaron Courville, Pierre-Antoine Manzagol, Pascal Vincent, and Samy Bengio. 2010. "Why does Unsupervised Pre-training Help Deep Learning?" https://research.google.com/pubs/pub35536.html.

Glorot, Xavier, Antoine Bordes, and Yoshua Bengio. 2011. "Deep sparse rectifier neural networks." *AISTATS '11: Proceedings of the 14th International Conference on Artificial Intelligence and Statistics* 15: 315–23. doi:10.1.1.208.6449.

Hinton, Geoffrey E., Simon Osindero, and Yee Whye Teh. 2006. "A fast learning algorithm for deep belief nets." *Neural Computation* 18 (7): 1527–54. doi:10.1162/neco.2006.18.7.1527.