Deeper Neural Networks with Non-Vanishing Logistic Hidden Units: *NoVa* vs. *ReLU* Neurons

Olaoluwa Adigun
Signal and Image Processing Institute
Department of Electrical and Computer Engineering
University of Southern California
Los Angeles, CA 90089-2564.
adigun@usc.edu

Bart Kosko
Signal and Image Processing Institute
Department of Electrical and Computer Engineering
University of Southern California
Los Angeles, CA 90089-2564.
kosko@usc.edu

***Abstract*—The new *NoVa* (nonvanishing) logistic neuron activation allows deeper neural networks because its derivative is positive. So it helps mitigate the problem of vanishing gradients in deep networks. Deep neural classifiers with NoVa hidden units had better classification accuracy on the CIFAR-10, CIFAR-100, and Caltech-256 image databases compared with threshold-linear ReLU hidden units. Still simpler identity hidden units also outperformed ReLU hidden units in deep classifiers but usually had less classification accuracy than NoVa networks. NoVa hidden neurons also outperformed ReLU hidden neurons in deep convolutional neural networks.**

I. The Search for a Better Hidden Neuron

What it is the best neural activation function $a : R \to R$ to use in the hidden layers of a deep neural network? This paper presents the new hybrid *NoVa* or nonvanishing logistic neuron as a candidate.

The current answer appears to be the threshold-linear neuron [10] that dates from the 1960s (see Equation (2) in both Fukushima papers [3] and [4]):

$$a(x) = \text{ReLU}(x) = \max\{0, x\} = \begin{cases} 0 & x \leq 0 \\ x & x > 0 \end{cases} \tag{1}$$

now called the *ReLU* (rectified linear unit) neuron [5], [13]. Goodfellow [6] even states that the ReLU is the "default recommendation" for the choice of hidden neuron in a deep feedforward classifier. Figure 1(a) shows the ramp-shaped graph of a ReLU neuron.

The default answer in the 1980s and 1990s was the logistic sigmoid σ in Figure 1(b):

$$a(x) = \sigma(bx) = \frac{1}{1 + \exp^{-bx}} \tag{2}$$

for some steepness constant $b > 0$. A large b turns the logistic's soft threshold into a hard threshold as in a classical on-off or threshold neuron. So the logistic activation seemed to have it both ways: It could easily model a threshold neuron and yet it was smoothly differentiable as all modern learning algorithms required. The logistic is also bounded because it lies in the unit interval $[0, 1]$. So it naturally defines a probability and indeed describes the posterior for two-class Bayesian decisions. A *bipolar* logistic results from the scaled and translated binary logistic $2\sigma(bx) - 1$ and lies in the bipolar interval $[-1, 1]$.

The logistic activation is differentiable. It has a simple closed-form and nonnegative derivative $a' \geq 0$:

$$a'(x) = \frac{da}{dx} = b\sigma(bx)[1 - \sigma(bx)] \tag{3}$$

for steepness parameter $b > 0$.

It is just this product form (3) that has led so many neural engineers to abandon the logistic neuron as a viable hidden unit in deep networks. The derivative quickly approaches zero as the neuron saturates to either its upper bound 1 or its lower bound 0.

Steep logistic sigmoid quickly saturate because they approximate on-off thresholds so well. This saturation leads to a "vanishing gradient" in the many learning algorithms that use the chain rule in computing the gradient of an error function. The learning gradient tends to vanish quickly as the number of hidden logistic layers grows.

The ReLU neuron in (1) thresholds a linear or identity activation and so does not saturate for large inputs x. The neuron is unbounded to the right and has a constant derivative for positive values. It does not have a derivative at the origin. Users often take this missing value as zero.

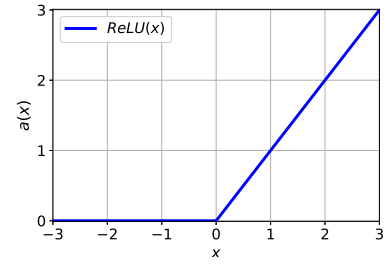
The ReLU neuron thus avoids saturation at the expense of an asymmetric threshold. Its linear portion also sacrifices the proven function-approximation power of nonlinear logistic hidden neurons [2]. And yet the ReLU still "dies" in large networks [1], [14], [15]. Our large-scale simulations show not only that it dies but that often the simpler and symmetric identity neuron $a(x) = x$ "lives" and gives better classification accuracy. Yet neither hidden neuron tends to outperform the new NoVa neuron in deep networks for large-K image datasets.

II. A New Activation: The NoVa neuron

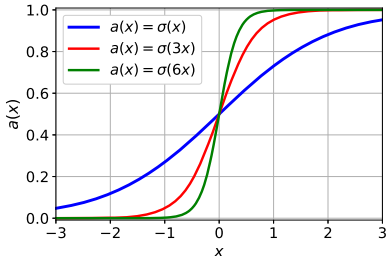
We now introduce the nonvanishing logistic or *NoVa* activation as a generalization of both the logistic and ReLU neurons. The NoVa neuron is a family of parametrized neurons. A simple example would be any scaled sum of a ReLU and a logistic. We instead focus on the NoVa neuron as a sum of a scaled linear or identity neuron and a logistic:

$$a(x) = cx + \sigma(bx) \tag{4}$$

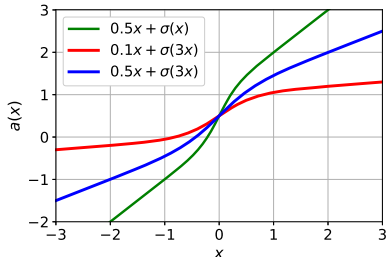
978-1-6654-4337-1/21/$31.00 ©2021 IEEE
DOI 10.1109/ICMLA52953.2021.00227



(a) Rectified Linear Unit (ReLU)



(b) Logistic sigmoid units



(c) Nonvanishing (NoVa) logistic sigmoids

Fig. 1: Graphs of hidden-neuron activation functions: threshold-linear or ReLU, logistic, and the new NoVa logistic neuron. The NoVa neuron $a(x) = 0.5x + \sigma(3x)$ performed best overall in simulations where σ denotes a logistic sigmoid function.

for scalar input x and where σ is the logistic sigmoid in (2). We assume that $c > 0$ and again that $b > 0$. Figure 1(c) shows the hybrid nature of three NoVa curves. Simulations found that the particular choice $a(x) = 0.5x + \sigma(3x)$ gave the best classification accuracy on deep multilayer classifiers trained on a large number K of patterns.

The crucial property of the NoVa neuron is that its derivative cannot vanish if $c > 0$:

$$a'(x) = c + b\,\sigma(bx)[1 - \sigma(bx)] \tag{5}$$

$$\geq c \quad > \quad 0. \tag{6}$$

This property justifies the name "nonvanishing." It also shows that the NoVa neuron is a type of *perturbed* logistic neuron where the constant $c > 0$ controls the degree of perturbation. An ordinary logistic σ results when there is no perturbation and thus when $c = 0$.

III. Simulation Comparison of Deep Classifiers

We compared the three types of hidden neurons on deep classifiers that trained on the CIFAR-10 image set and the much larger ("big K") image sets CIFAR-100 and Caltech-256. The findings support using NoVa hidden neurons in at least very deep networks.

NoVa networks tended to perform best in classification accuracy while logistic networks performed worst for very deep networks trained on large-K image sets. The logistic networks suffered quickly from the predicted vanishing gradient [8] for neural classifiers with only a few hidden layers. ReLU networks did better but also died [1], [14], [15]. A surprising finding was that hidden layers with simple identity hidden neurons $a(x) = x$ often performed quite well and tended to easily outperform ReLU networks in deeper classifiers. NoVa networks performed best in the deepest networks.

The neural classifiers consisted of several layers. All input layers used identity neurons as data registers. All output layers used K softmax classifier neurons. The hidden neurons were either all ReLU (or identity) or all logistic or all NoVa. We then reran the simulations on deep convolutional classifiers. NoVa networks still performed best for the deeper classifiers. The next section describes the dataset for the experiments.

A. Datasets for the Deep Classifiers

The simulations used three image datasets. The first was the usual CIFAR-10 dataset. The second was its extension to CIFAR-100. The third was the Caltech-256 image dataset.

1) CIFAR-10: The popular CIFAR-10 image set is a small-K test set that consists of 60,000 color images from 10 categories ($K = 10$). Each image has size $32 \times 32 \times 3$. The 10 pattern categories are airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck [12]. Each class consists of 5,000 training samples and 1,000 testing samples. Figure 2 shows 10 sample images from the CIFAR-10 dataset with one image per pattern class.

Fig. 2: CIFAR-10 sample images: The figure shows 10 samples from the CIFAR-10 dataset that contains 10 image pattern classes and a total of 60,000 sample images.

2) CIFAR-100: CIFAR-100 is a large-K set of 60,000 color images with image size $32 \times 32 \times 3$. The images come from 100 pattern classes ($K = 100$) with 600 images per class. Each class consists of 500 training images and 100 test images. Figure 3 shows 100 sample images from the CIFAR-100 dataset with one image per pattern class.

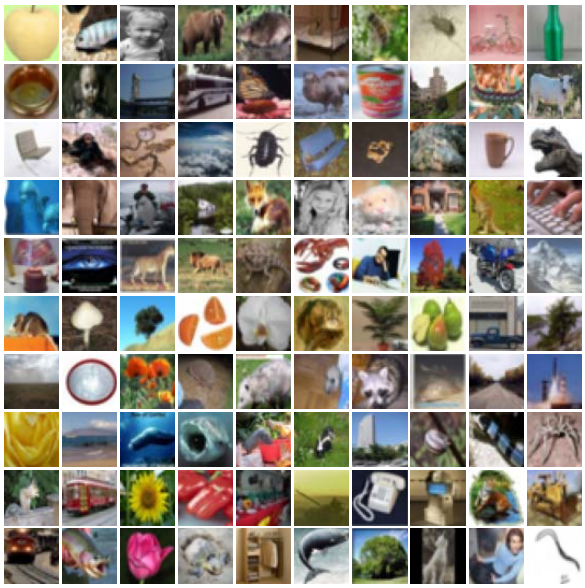


Fig. 3: CIFAR-100 sample images: The figure shows 100 samples from the CIFAR-100 dataset that contains 100 pattern classes with 600 images per class. CIFAR-100 consists of 20 super-classes with 5 classes per super-class.

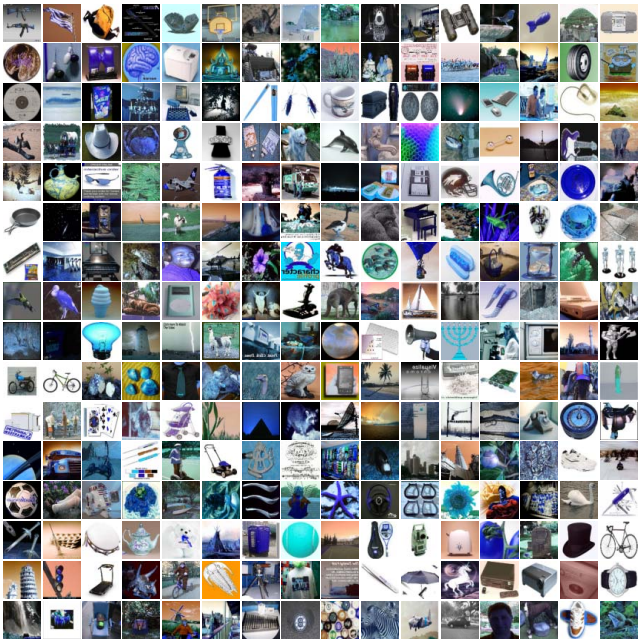


Fig. 4: Caltech-256 sample images: The figure shows 256 samples from the Caltech-256 dataset that consists of 256 pattern classes.

3) Caltech-256: This large-K dataset had 30,607 images from 256 pattern classes. So $K = 256$. Each class had between 31 and 80 images. The 256 classes consisted of the two superclasses $animate$ and $inanimate$. The animate superclass contained 69 pattern classes. The inanimate superclass contained 187 pattern classes [7]. We removed the $cluttered$ images and reduced the size of the dataset to 29,780 images. We resized each image to 100×100×3. Figure 4 shows 256 sample images from the Caltech-256 dataset with one image per pattern class.

B. Network Description and Training Parameters

We trained the deep neural classifiers on the CIFAR and Caltech-256 datasets. We varies the number of neurons per hidden layer. The classifiers used 100, 500, or 1,000 neurons per hidden layer. The convolutional neural networks used either ReLU or identity activations in their convolutional layers and then used either ReLU, identity, logistic, or NoVa hidden neurons in their fully connected layers. We also varied the number of hidden layers.

All the deep neural classifier models used softmax activations for their classifier outputs with 1-in-K coding. Here K denotes the total number of pattern classes in the dataset that the classifier recognizes. The trained classifier itself partitions the input pattern space into K decision classes. This trained K-partition further opens the door to XAI or explainable AI proxy systems that can estimate or absorb the partition result into a rule-based or other structured system.

The deep classifier networks trained over 100 epochs with the ordinary unidirectional backpropagation algorithm [9], [12] of iterative maximum likelihood [11]. So the algorithm iteratively climbed the nearest hill of log-likelihood to find the total network parameters Θ^*:

$$\Theta^* = \underset{\Theta}{\operatorname{argmax}} \ln p(\mathbf{y}|\mathbf{x}, \Theta). \tag{7}$$

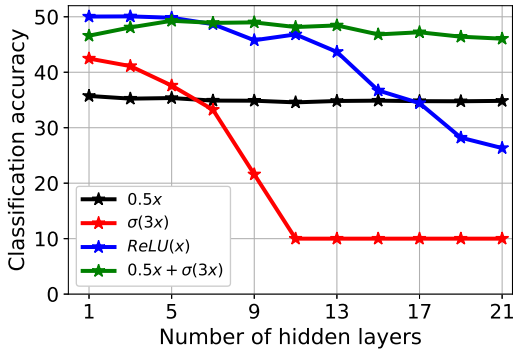
Then the gradient

$$\nabla_\Theta L = \nabla_\Theta \ln p(\mathbf{y}|\mathbf{x}, \Theta) \tag{8}$$

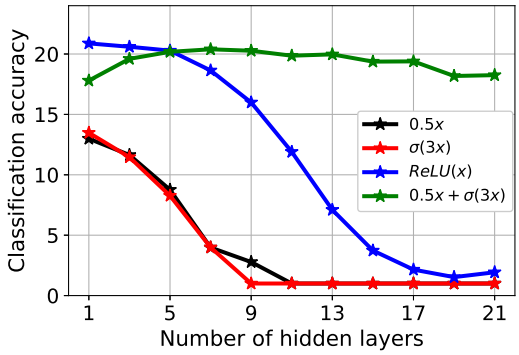
gives the backpropagation algorithm for the total network log-likelihood L so long as backpropagation invariance holds at the terminal or output layer of nodes.

Backpropagation invariance does hold here for a classifier because the K output softmax neurons define a one-shot multinomial probability density or one roll of a K-sided die. Then the negative log-likelihood of that multinomial gives the error function as the usual cross-entropy. So its gradient gives back the usual signal-times-error form of the backpropagation algorithm as it iteratively maximizes the layer likelihood. This is just a special case of the general Expectation-Maximization algorithm (because at root Shannon entropy minimizes cross-entropy) [11].

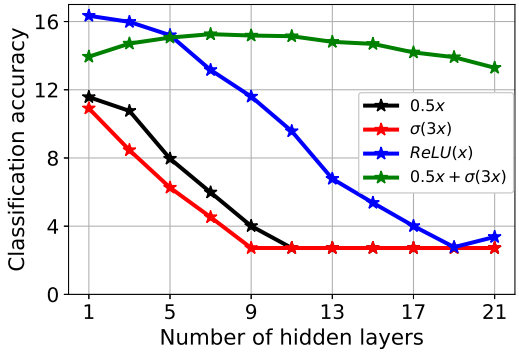
This backpropagation (maximum-likelihood) training used stochastic gradient descent with a momentum value of 0.0 and learning rate $\alpha = 0.001$. A dropout value of 0.1 for some of the hidden layers reduced the overfitting.



(a) Networks trained on the CIFAR-10 dataset



(b) Networks trained on the CIFAR-100 dataset



(c) Networks trained on the Caltech-256 dataset

Fig. 5: The NoVa classifiers (in green) had far greater classifier accuracy than the other classifiers as the number of hidden layers increased.

Figure 5 shows the classification-accuracy curves for the simulations on all three image datasets. The NoVa classifiers markedly outperformed the ReLU and other classifiers as the number of hidden layers increased. The benefit of NoVa hidden units grew more pronounced on the big-K image datasets CIFAR-100 and Caltech-256 with respective pattern classes $K = 100$ and $K = 256$.

Tables I–V confirm this NoVa performance in greater detail for both ordinary and convolutional deep classifiers. The tables also reveal that the simple identity hidden unit always outperformed the ReLU unit in the convolutional comparisons and sometimes outperformed the NoVa neuron there as well. So users should consider experimenting with both identity and NoVa units in very deep large-K networks.

IV. CONCLUSIONS

The new nonvanishing logistic NoVa hidden neuron outperformed ReLU hidden neurons for very deep networks on the CIFAR-10 dataset and especially on the much larger datasets CIFAR-100 and Caltech-256. Further simulations showed that the NoVa hidden neurons outperformed leaky-ReLU hidden neurons as well. The ordinary identity activation also often outperformed ReLU hidden neurons for very deep networks although the classification accuracy was less than with NoVa hidden units (except in some cases of convolutional classifiers). The NoVa neuron itself generalizes in many directions. Future simulations need to explore these variants on different and still larger datasets.

REFERENCES

[1] A. F. Agarap, "Deep learning using rectified linear units (relu)," *arXiv preprint arXiv:1803.08375*, 2018.

[2] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.

[3] K. Fukushima, "Visual feature extraction by a multilayered network of analog threshold elements," *IEEE Transactions on Systems Science and Cybernetics*, vol. 5, no. 4, pp. 322–333, 1969.

[4] ——, "Cognitron: A self-organizing multilayered neural network," *Biological cybernetics*, vol. 20, no. 3, pp. 121–136, 1975.

[5] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 315–323.

[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[7] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," 2007.

[8] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber *et al.*, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," 2001.

[9] W. P. J., "Beyond regression: New tools for prediction and analysis in the behavioral sciences." *Doctoral Dissertation, Applied Mathematics, Harvard University, MA*, 1974.

[10] B. Kosko, *Neural Networks and Fuzzy Systems*. Prentice-Hall, 1991.

[11] B. Kosko, K. Audhkhasi, and O. Osoba, "Noise can speed backpropagation learning and deep bidirectional pretraining," *Neural Networks*, vol. 129, pp. 359–384, 2020.

[12] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," Citeseer, Tech. Rep., 2009.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[14] L. Lu, "Dying relu and initialization: Theory and numerical examples," *Communications in Computational Physics*, vol. 28, no. 5, pp. 1671–1706, 2020.

[15] L. Trottier, P. Giguere, and B. Chaib-Draa, "Parametric exponential linear unit for deep convolutional neural networks," in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2017, pp. 207–214.

Hidden Activation	100 Neurons per Hidden Layer			500 Neurons per Hidden Layer		
	1 Hidden layer	*7 Hidden layers*	*13 Hidden layers*	*1 Hidden Layer*	*7 Hidden layers*	*13 Hidden layers*
x	35.70%	34.73%	34.53%	35.73%	34.89%	34.82%
$0.5x$	35.02%	28.21%	**9.39%**	35.08%	29.92%	**9.39%**
$\sigma(x)$	32.60%	**10.00%**	**10.00%**	33.86%	**10.00%**	**10.00%**
$\sigma(3x)$	39.28%	23.45%	**10.00%**	42.47%	33.24%	**10.00%**
ReLU(x)	**47.54%**	**43.39%**	32.67%	**50.04%**	48.87%	43.67%
$x + \sigma(x)$	38.00%	39.27%	37.04%	40.10%	43.43%	41.02%
$0.5x + \sigma(x)$	37.59%	34.88%	31.07%	39.89%	37.85%	34.81%
$0.5x + \sigma(3x)$	41.83%	42.73%	**40.10%**	46.59%	**48.90%**	**48.44%**

TABLE I: Performance of NoVa and other hidden units on the CIFAR-10 dataset for non-convolutional deep classifiers. The classifiers used 3,072 identity neurons at the input and 10 softmax neurons at the output layer ($K = 10$). We varied the number of hidden layers (1,7, or 13) and the number of neurons per hidden layer (100 or 500). NoVa hidden neurons performed best in classification accuracy for the deepest classifier with the most hidden neurons per layer.

Hidden Activation	100 Neurons per Hidden Layer			500 Neurons per Hidden Layer		
	1 Hidden layer	*7 Hidden layers*	*13 Hidden layers*	*1 Hidden Layer*	*7 Hidden layers*	*13 Hidden layers*
x	13.22%	13.15%	12.25%	13.58%	14.21%	13.70%
$0.5x$	12.44%	2.49%	1.00%	3.99%	3.97%	**1.06%**
$\sigma(x)$	5.47%	**1.00%**	**1.00%**	8.36%	**1.00%**	**1.00%**
$\sigma(3x)$	9.54%	10.50%	**1.00%**	13.48%	3.99%	**1.00%**
ReLU(x)	**16.88%**	10.79%	2.28%	**20.88%**	15.43%	7.10%
$x + \sigma(x)$	13.67%	13.73%	**13.58%**	16.42%	16.42%	16.83%
$0.5x + \sigma(x)$	13.61%	11.70%	6.13%	14.43%	11.19%	11.42%
$0.5x + \sigma(3x)$	13.78%	**14.80%**	13.26%	17.80%	**20.40%**	**19.97%**

TABLE II: Performance of NoVa and other hidden units on the CIFAR-100 dataset for non-convolutional deep classifiers. The classifiers used 3,072 identity neurons at the input and 100 softmax neurons at the output layer ($K = 100$). We varied the number of hidden layers (1,7, or 13) and the number of neurons per hidden layer (100 or 500). Identity hidden neurons did better than ReLU neurons for the deepest classifier with 13 hidden layers NoVa hidden neurons performed best for the deep classifiers with 7 or with 13 hidden layers.

Hidden Activation	100 Neurons per Hidden Layer			500 Neurons per Hidden Layer		
	1 Hidden layer	*7 Hidden layers*	*13 Hidden layers*	*1 Hidden Layer*	*7 Hidden layers*	*13 Hidden layers*
x	11.38%	11.30%	10.70%	11.18%	12.39%	**12.19%**
$0.5x$	10.73%	5.64%	2.72%	11.57%	5.98%	**2.72%**
$\sigma(x)$	6.13%	**2.72%**	**2.72%**	7.01%	**2.72%**	**2.72%**
$\sigma(3x)$	7.15%	4.13%	**2.72%**	10.91%	4.53%	**2.72%**
ReLU(x)	**13.36%**	8.65%	3.16%	**16.37%**	13.16%	6.85%
$x + \sigma(x)$	12.21%	12.56%	**11.72%**	12.19%	13.38%	7.11%
$0.5x + \sigma(x)$	12.06%	9.00%	6.23%	12.96%	12.34%	9.10%
$0.5x + \sigma(3x)$	12.72%	**11.11%**	9.65%	13.94%	**15.26%**	**14.52%**

TABLE III: Performance of NoVa and other hidden units on the Caltech-256 dataset for non-convolutional deep classifiers. The classifiers used 30,000 identity neurons at the input and 256 softmax neurons at the output layer ($K = 256$). We varied the number of hidden layers (1,7, or 13) and the number of neurons per hidden layer (100 or 500). Identity hidden neurons did better than ReLU neurons for the deepest classifier with 13 hidden layers. NoVa hidden neurons performed best for the deep classifiers with 7 or with 13 hidden layers.

Hidden Activation	Dataset		
	CIFAR-10	*CIFAR-100*	*Caltech-256*
x	34.81%	13.52%	12.00%
$0.5x$	10.00%	1.00%	2.72%
$\sigma(x)$	10.00%	1.00%	2.72%
$\sigma(3x)$	10.00%	1.00%	2.72%
ReLU(x)	26.86%	1.87%	2.72%
$x + \sigma(x)$	39.27%	15.38%	2.72%
$0.5x + \sigma(x)$	24.21%	4.70%	5.07%
$0.5x + \sigma(3x)$	**46.53%**	**18.86%**	**13.45%**

TABLE IV: Hidden activations affect very deep neural classifiers. The classifiers used 20 hidden layers with 500 identical neurons per hidden layer for all three image datasets. The ReLU and logistic sigmoid failed as the layer depth increased. Identity and NoVa classifiers performed better while the NoVa networks performed best for all three image datasets.

Dataset	Activation		Accuracy	
	Convolutional Layer	*Fully Connected (FC) Layer*	*2 FC Layers*	*5 FC Layers*
CIFAR-10	ReLU	ReLU	68.43%	31.12%
		Sigmoid	59.12%	9.81%
		Identity	68.89%	**66.73%**
		NoVa	**70.94%**	43.49%
	Identity	ReLU	49.34%	27.75%
		Sigmoid	37.40%	10.34%
		Identity	54.60%	**54.00%**
		NoVa	**55.96%**	49.81%
CIFAR-100	ReLU	ReLU	38.97%	19.69%
		Sigmoid	21.69%	7.65%
		Identity	39.10%	**32.91%**
		NoVa	**45.63%**	29.86%
	Identity	ReLU	19.97%	12.85%
		Sigmoid	11.60%	0.67%
		Identity	24.49%	18.73%
		NoVa	**25.61%**	**19.99%**
Caltech-256	ReLU	ReLU	20.99%	7.42%
		Sigmoid	9.36%	2.69%
		Identity	28.74%	**16.36%**
		NoVa	**34.79%**	14.66%
	Identity	ReLU	10.22%	7.75%
		Sigmoid	9.22%	2.75%
		Identity	16.28%	12.75%
		NoVa	**16.58%**	**13.29%**

TABLE V: Fully connected NoVa hidden layers outperformed others in deep convolutional neural networks (CNNs). The CNNs used 3 convolutional layers (either with ReLU or with identity activations $a(x) = x$) and varied the number of fully connected hidden layers (2 or 5). NoVa hidden neurons at the fully connected layers gave better classification accuracy than did ReLU, identity, and ordinary logistic neurons for both types of convolutional layers. Identity neurons also outperformed ReLU at the fully connected hidden layers.