ProbAct: A probabilistic Activation Function

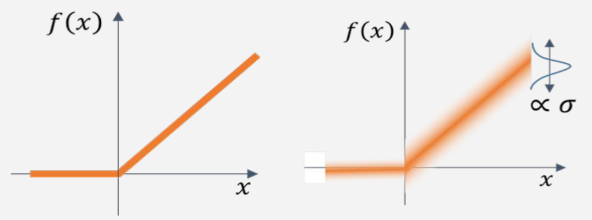
Kumar Shridhar *, Joonho Lee, Hideaki Hayashi, Purvanshi Mehta, Brian Kenji Iwana, Seokjun Kang, Seiichi Uchida, Sheraz Ahmed, Andreas Dengel

* shridhar.kumar@inf.ethz.ch

Abstract

ProbAct is a novel stochastic activation function composed of a mean and variance with the output value sampled from the formed distribution. Stochastic perturbation induced through ProbAct is shown to act as a viable generalization technique for feature augmentation. We compare ProbAct with well-known activation functions on classification tasks on Images and Text dataset and achieved performance improvements of +2-3%.

ProbAct: A Stochastic Activation Function



- Every layer of a neural network computes its output y for the given input x:

$$y = f(W^T x)$$

where *w* is the weight vector of the layer and f(·) is an activation function, such as ProbAct.

- ProbAct is defined as:

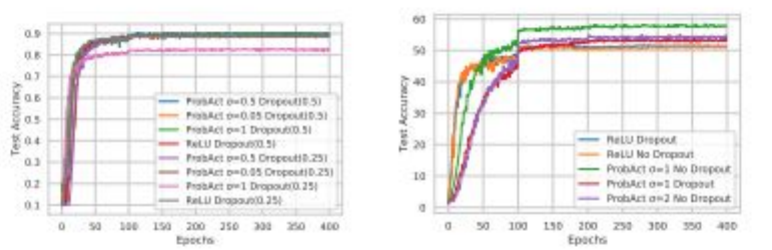
$$f(a) = \mu(a) + \sigma\epsilon$$

where *a* is the input to the activation function, μ(a) is a static or learnable mean, σ defines the range of stochastic perturbation (can be static or learnable) and ϵ is a random value sampled from a normal distribution N(0,1).

Experiments

- We evaluate ProbAct on the following image classification and sentiment analysis datasets :
  - CIFAR-10, CIFAR-100, and STL-10
  - Large Movie Review
- We compare ProbAct to the following activation functions and methods:
  - ReLU[1], Sigmoid, TanH, Leaky ReLU[2], PReLU[3], ELU[4], SELU[5], and Swish[6].
  - Bayesian VGG with Variational Inference[7]
- We used a 16 layer VGG for Image Classification and two-layered CNN for text classification task.
- No regularization, pre-training or data augmentation techniques were used.
- STL images were resized to 32 by 32.

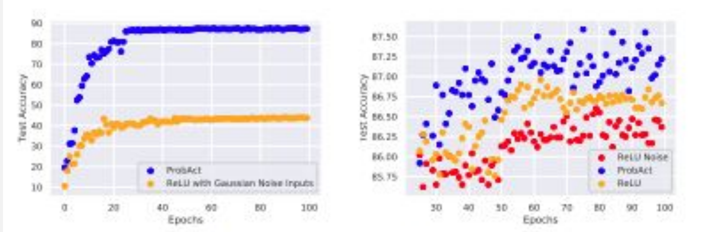
Accuracy Comparison between ReLU and ProbAct



(*a*) Test Acc CIFAR10 (*b*) Test Acc CIFAR100

Results and Discussion

Activation function	CIFAR-10	CIFAR-100	STL-10	IMDB	Train time (sec)	Test time (milli-sec)
Sigmoid	10.00	1.00	10.00	85.92	1.07	1.03
Tanh	10.00	1.00	10.00	85.88	1.08	1.03
ReLU	87.27	52.94	60.80	85.85	**1.00**	**1.00**
Leaky ReLU	86.49	49.44	59.16	85.47	1.04	1.08
PReLU	86.35	46.30	60.01	85.95	1.16	**1.00**
ELU	87.65	56.60	64.11	86.51	1.16	1.04
SELU	86.65	51.52	60.71	85.71	1.19	1.05
Swish	86.55	54.01	63.50	86.14	1.20	1.13
Bayesian VGG VI[1]	86.22	48.27	57.22	-	-	-
ProbAct						
Mean						
Element-wise μ	85.80	48.50	54.17	83.86	1.29	1.35
Sigma						
Fixed ($\sigma = 0.5$)	88.50	56.85	62.30	**87.31**	1.09	1.25
Fixed ($\sigma = 1.0$)	88.87	**58.45**	62.50	87.00	1.10	1.27
One Trainable σ	87.40	53.87	63.07	86.35	1.23	1.30
EW Trainable σ						
Unbound	86.40	54.10	61.70	86.64	1.25	1.31
Bound	**88.92**	55.83	**64.17**	85.86	1.26	1.33



(*c*) Noisy Input (*d*) Noisy Activation

References

1. Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In International Conference on Machine Learning, pages 807–814, 2010.
2. Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. arXiv preprint arXiv:1505.00853, 2015.
3. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In International Conference on Computer Vision, dec 2015. doi: 10.1109/iccv.2015.123.
4. Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). arXiv preprint arXiv:1511.07289, 2015
5. G Klambauer, T Unterthiner, A Mayr, and S Hochreiter. Self-normalizing neural networks. arxiv 2017. arXiv preprint arXiv:1706.02515, 2017.
6. Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. arXiv preprint arXiv:1710.05941, 2017
7. Kumar Shridhar, Felix Laumann, and Marcus Liwicki. Uncertainty estimations by softplus normalization in bayesian convolutional neural networks with variational inference. arXiv preprint arXiv:1806.05978, 2018.