ELU-Networks: Fast and Accurate CNN Learning on ImageNet

Martin Heusel, Djork-Arné Clevert, Günter Klambauer, Andreas Mayr, Karin Schwarzbauer, Thomas Unterthiner, and Sepp Hochreiter

JOHANNES KEPLER UNIVERSITY LINZ

Abstract: We trained a CNN on the ImageNet dataset with a new activation function, called "exponential linear unit" (ELU) [1], to speed up learning.

Like rectified linear units (ReLUs) [2,3], leaky ReLUs (LReLUs) and parametrized ReLUs (PReLUs), ELUs also avoid a vanishing gradient via the identity for positive values. However ELUs have improved learning characteristics compared to the other activation functions. In contrast to ReLUs, ELUs have negative values which allows them to push mean unit activations closer to zero. Zero means speed up learning because they bring the gradient closer to the unit natural gradient. Like batch normalization, ELUs push the mean towards zero, but with a significantly smaller computational footprint. While other activation functions like LReLUs and PReLUs also have negative values, they do not ensure a noise-robust deactivation state. ELUs saturate to a negative value with smaller inputs and thereby decrease the propagated variation and information. Therefore ELUs code the degree of presence of particular phenomena in the input, while they do not quantitatively model the degree of their absence. Consequently dependencies between ELU units are much easier to model and distinct concepts are less likely to interfere.

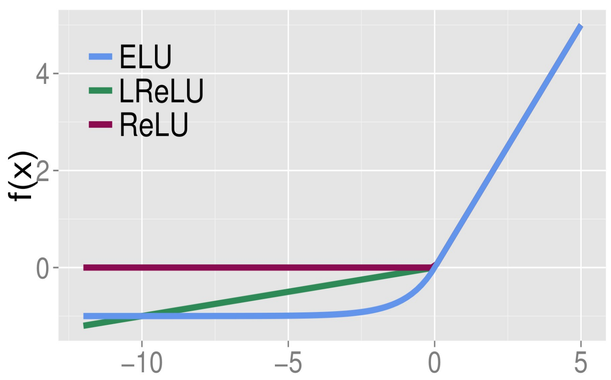
In the ImageNet challenge ELU networks considerably speed up learning compared to a ReLU network with similar classification performance (top 5 error below 10%).

Goal

- speed up learning in deep networks
- avoid bias shift (see below)
- bring gradient closer to naural gradient

Solution

- **exponential linear units (ELUs)**
- negative part serves as bias
- smaller mean → smaller bias shift

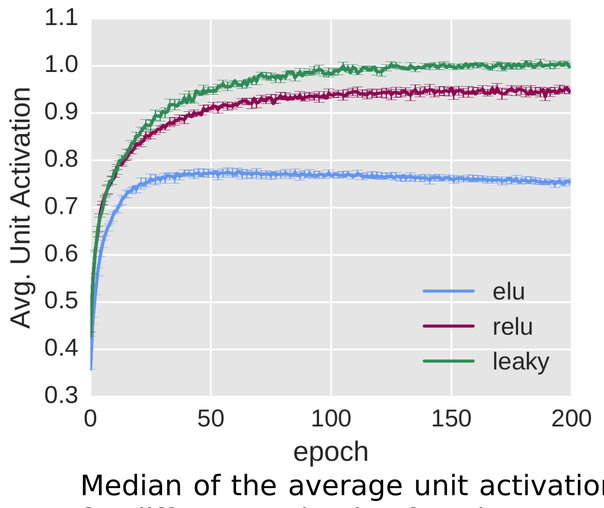


$$f(x) = \begin{cases} x & \text{if } x \geq 0 \\ \alpha\,(\exp(x) - 1) & \text{if } x < 0 \end{cases}$$

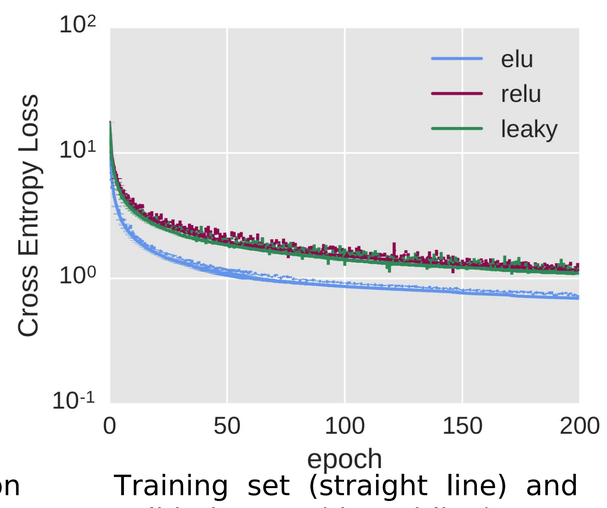
$$f'(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ f(x) + \alpha & \text{if } x < 0 \end{cases}$$

ELUs tested on MNIST

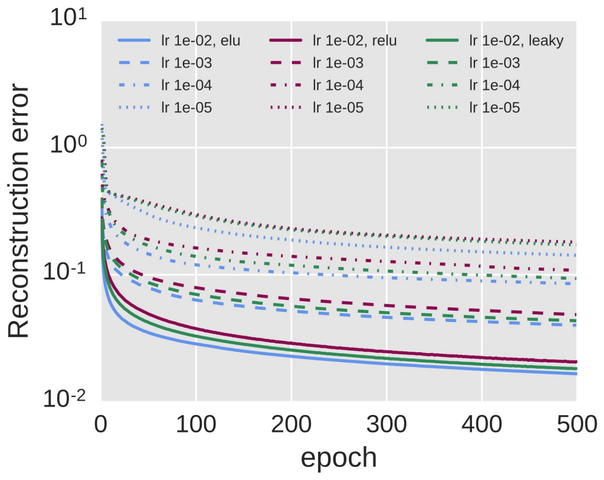
ELU networks evaluated at MNIST: mean activation and test/training loss. Average over five runs with different random initializations, error bars show standard deviation.



Median of the average unit activation for different activation functions.



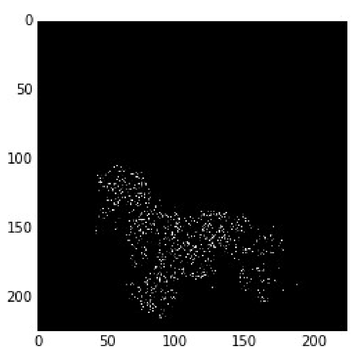
Training set (straight line) and validation set (dotted line) cross entropy loss.

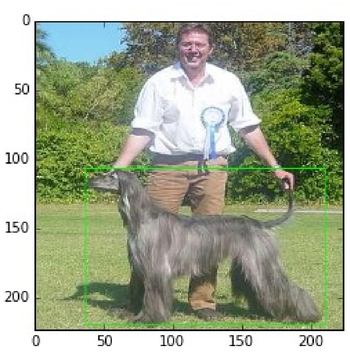


Autoencoder training on MNIST

Training set reconstruction error over epochs, using different activation functions and learning rates. The results are medians over several runs with different random initializations.

Backpropagation for Bounding Box





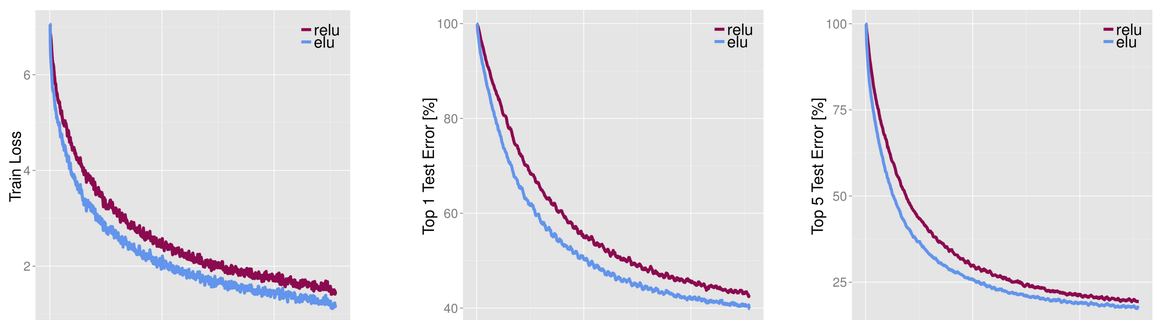
ELUs tested on CIFAR-10 and CIFAR-100

Network	CIFAR-10	CIFAR-100	augmented
AlexNet	18.04	45.80	-
DSN	7.94	34.57	√
NiN	8.81	35.68	√
Maxout	9.38	38.57	√
All-CNN	7.25	33.71	√
Highway Network	7.60	32.24	√
Fract Max-Pooling	**4.50**	**27.62**	√
ELU-Network	**6.55**	**24.28**	-

Test error in %. Comparison of ELU networks and other convolutional networks on CIFAR-10 and CIFAR-100. Reported is the test error in percent misclassification. Best results are in bold red, second best bold black.

ELUs tested on ImageNet

15 layer CNN with stacks of (1×96×6, 3×512×3, 5×768×3, 3×1024×3, 2×4096×FC, 1×1000×FC) layers×units×receptive fields or fully-connected (FC). 2×2 max-pooling with a stride of 2 after each stack, spatial pyramid pooling with 3 levels before the first FC layer.
L2-weight decay: 0.0005; 50% drop-out in FC layers; images resized to 256×256, subtracted per-pixel mean, training on 224 × 224 random crops with vertical flipping. No augmentation; single-model; single center crop.



ELUs used for the ImageNet classification task. The x-axis gives the number of iterations. The y-axis shows the training loss (left), top-5 error (middle) and the top-1 error (right) of 5,000 random validation samples, evaluated on the center crop. Both activation functions ELU (purple) and ReLU (blue) lead for convergence, but ELUs start reducing the error earlier and reach the 20% top-5 error after 160k iterations, while ReLUs need 200k iterations to reach the same error rate.

We submitted to ILSVRC 2015 achieving 9.18% test classification error rate.

Bias Shift and Unit Natural Gradient

- parametrized probabilistic model $p(\boldsymbol{x}; \boldsymbol{w})$ with parameter $\boldsymbol{w}$ and data $\boldsymbol{x}$
- training data $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N) \in \mathbb{R}^{(d+1)\times N}$ with $\boldsymbol{x}_n = (\boldsymbol{z}_n^T, y_n)^T \in \mathbb{R}^{d+1}$, where $\boldsymbol{z}_n$ is the input for example n and y_n is its label
- $\hat{\delta}$ error at unit i is $\hat{\delta} = \frac{\partial}{\partial \text{net}} \ln p\,(y \mid \boldsymbol{z}; \boldsymbol{w})$

Theorem 1. *Unit natural gradients correct weight updates* $(\Delta \boldsymbol{w}^T, \Delta w_0)^T$ *by*

$$\begin{pmatrix} \Delta \boldsymbol{w} \\ \Delta w_0 \end{pmatrix} = \begin{pmatrix} \boldsymbol{A}^{-1}\ (\boldsymbol{g} - \Delta w_0\ \boldsymbol{b}) \\ s\ \left(g_0 - \boldsymbol{b}^T\ \boldsymbol{A}^{-1} \boldsymbol{g}\right) \end{pmatrix} ,$$

where $\nabla_{(\boldsymbol{w}, w_0)} R_{\text{emp}} = (\boldsymbol{g}^T, g_0)^T$ *is the gradient and* $\boldsymbol{A}$ *is the unit Fisher information matrix without the bias weight. The vector* $\boldsymbol{b} = [\boldsymbol{F}(\boldsymbol{w})]_0$ *is the unit Fisher matrix column corresponding to the bias weight, and the scalar* s *is*

$$s = \mathrm{E}_{p(\boldsymbol{z})}^{-1}\left(\bar{\delta}^2\right) \left(1 + \mathrm{E}_{q(\boldsymbol{z})}^{T}(\boldsymbol{a})\, \mathrm{Var}_{q(\boldsymbol{z})}^{-1}(\boldsymbol{a})\ \mathrm{E}_{q(\boldsymbol{z})}(\boldsymbol{a})\right) ,$$

where $\boldsymbol{a}$ *is the vector of activations of units with weights to unit* i *and*

$$q(\boldsymbol{z}) = \bar{\delta}^2(\boldsymbol{z})\ p(\boldsymbol{z})\ \mathrm{E}_{p(\boldsymbol{z})}^{-1}\left(\bar{\delta}^2\right) \quad , \quad \bar{\delta}^2(\boldsymbol{z}) = \mathrm{E}_{p(y|\boldsymbol{z};\boldsymbol{w})}\left(\hat{\delta}^2\right) .$$

Theorem 2. *The bias shift correction of unit* i *by the unit natural gradient update is equivalent to following correction of the incoming mean* $\mathrm{E}_{p(\boldsymbol{z})}(\boldsymbol{a})$*:*

$$-\ (1 + k)\ \mathrm{E}_{q(\boldsymbol{z})}(\boldsymbol{a}) , \quad \text{where} \quad k = s\ \mathrm{E}_{p(\boldsymbol{z})}\left(\bar{\delta}^2\right)\ \mathrm{Cov}_{p(\boldsymbol{z})}^{T}\left(\bar{\delta}^2, \boldsymbol{a}\right) \boldsymbol{A}^{-1} \mathrm{E}_{q(\boldsymbol{z})}(\boldsymbol{a}) .$$

For $\mathrm{Cov}_{p(\boldsymbol{z})}\left(\bar{\delta}^2, \boldsymbol{a}\right) = \boldsymbol{0}$*, the incoming mean* $\mathrm{E}_{p(\boldsymbol{z})}(\boldsymbol{a})$ *is corrected to zero.*

References:
[1] Clevert et al, Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs), arxiv 2015
[2] Clevert et al, Rectified Factor Networks, NIPS 2015
[3] Mayr et al, DeepTox: Toxicity Prediction using Deep Learning, Frontiers in Environmental Science 2015
[4] Desjardins et al., Metric-free natural gradient for joint-training of Boltzmann machines, ICLR 2013
[5] Martens, Deep learning via Hessian-free optimization, ICML 2010