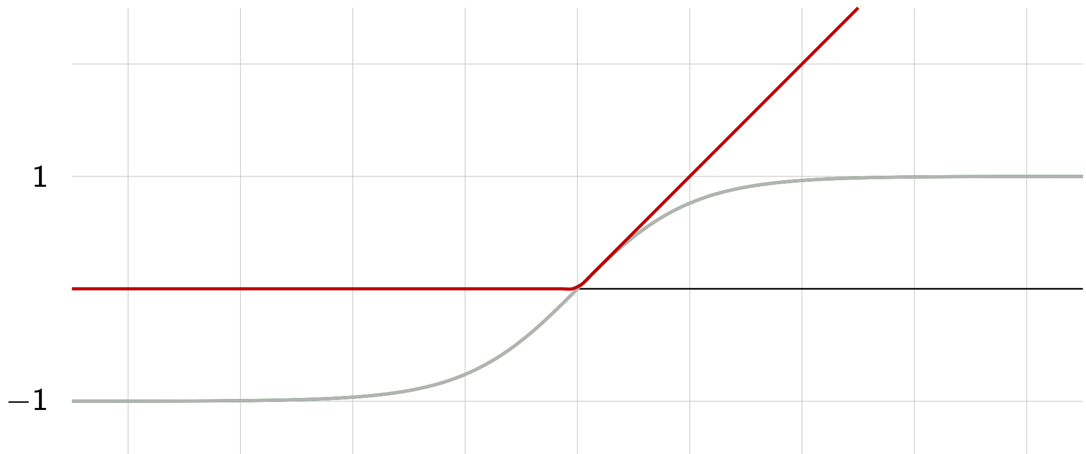
Deep learning

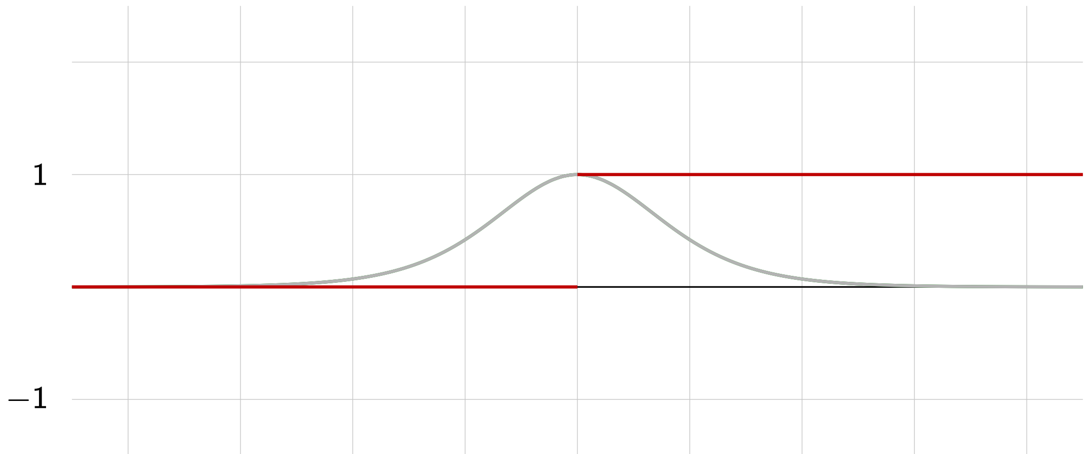
6.2. Rectifiers

François Fleuret
https://fleuret.org/dlc/
Jan 1, 2021

The use of the ReLU activation function was a great improvement compared to the historical tanh.



This can be explained by the derivative of ReLU itself not vanishing, and by the resulting coding being sparse (Glorot et al., 2011).



The steeper slope in the loss surface speeds up the training.

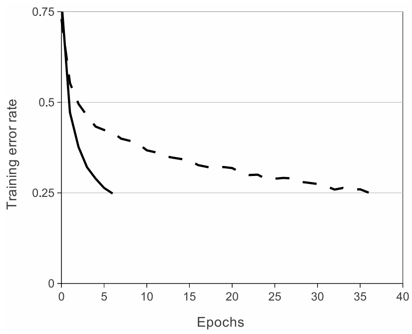


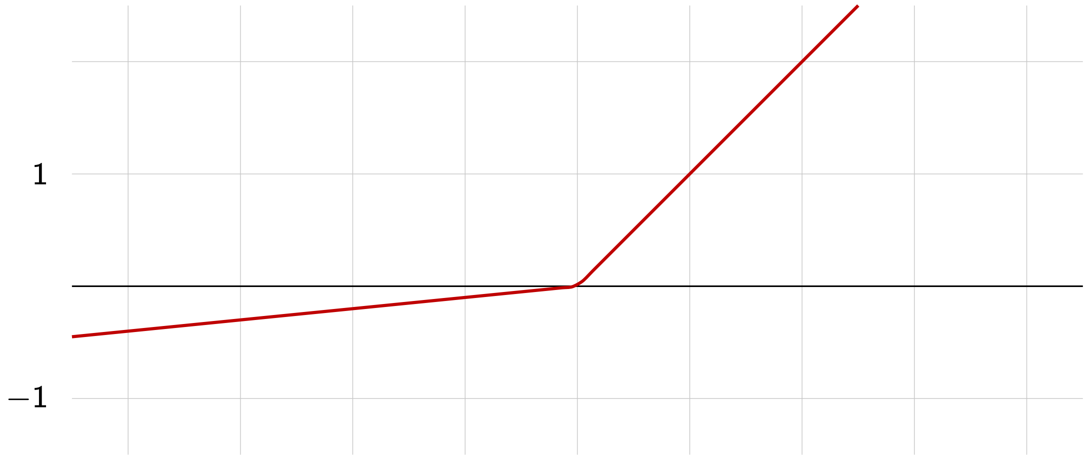
Figure 1: A four-layer convolutional neural network with ReLUs **(solid line)** reaches a 25% training error rate on CIFAR-10 six times faster than an equivalent network with tanh neurons **(dashed line)**. The learning rates for each network were chosen independently to make training as fast as possible. No regularization of any kind was employed. The magnitude of the effect demonstrated here varies with network architecture, but networks with ReLUs consistently learn several times faster than equivalents with saturating neurons.

(Krizhevsky et al., 2012)

A first variant of ReLU is Leaky-ReLU (Maas et al., 2013)

$$\begin{aligned}\mathbb{R} &\to \mathbb{R}\\ x &\mapsto \max(ax, x)\end{aligned}$$

with $0 \leq a < 1$ usually small.

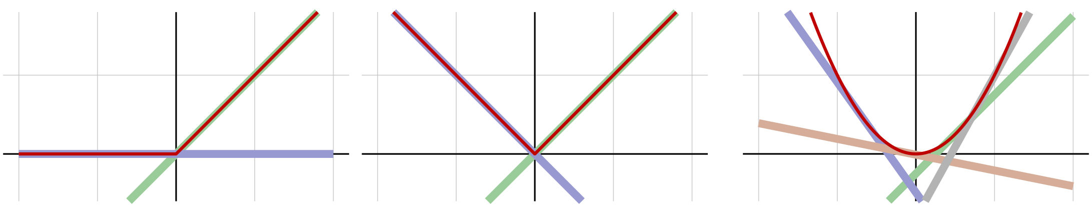


The parameter a can be optimized during training (PReLU, He et al., 2015), or randomized for every sample (RReLU, Xu et al., 2015).

The "maxout" layer proposed by Goodfellow et al. (2013) takes the max of several linear units. This is not an activation function in the usual sense, since it has trainable parameters.

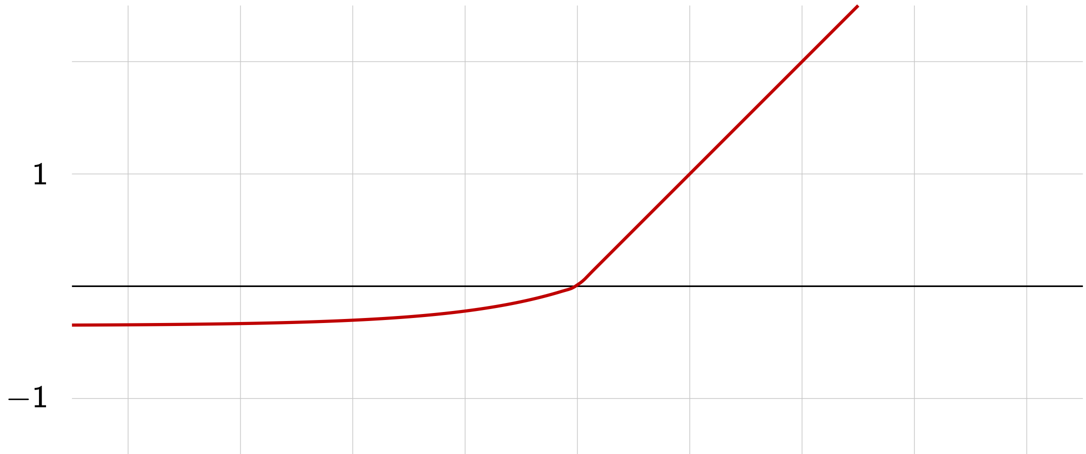
$$\begin{aligned}h : \mathbb{R}^D &\to \mathbb{R}^M\\ x &\mapsto \left(\max_{j=1}^{K} x^\top W_{1,j} + b_{1,j}, \dots, \max_{j=1}^{K} x^\top W_{M,j} + b_{M,j}\right)\end{aligned}$$

It can in particular encode ReLU and absolute value, but can also approximate any convex function.



Clevert et al. (2015) proposed the exponential linear unit (ELU), with an exponential saturation

$$x \mapsto \begin{cases} x & \text{if } x \geq 0 \\ \alpha \left(e^x - 1\right) & \text{otherwise.} \end{cases}$$



Another variant is the "Concatenated Rectified Linear Unit" (CReLU) proposed by Shang et al. (2016):

$$\mathbb{R} \to \mathbb{R}^2$$
$$x \mapsto (\max(0, x), \max(0, -x)),$$

which doubles the number of activations but keeps the norm of the signal intact during both the forward and the backward passes.

References

D. Clevert, T. Unterthiner, and S. Hochreiter. **Fast and accurate deep network learning by exponential linear units (elus)**. CoRR, abs/1511.07289, 2015.

X. Glorot, A. Bordes, and Y. Bengio. **Deep sparse rectifier neural networks**. In International Conference on Artificial Intelligence and Statistics (AISTATS), 2011.

I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio. **Maxout networks**. In International Conference on Machine Learning (ICML), 2013.

K. He, X. Zhang, S. Ren, and J. Sun. **Delving deep into rectifiers: Surpassing human-level performance on imagenet classification**. CoRR, abs/1502.01852, 2015.

A. Krizhevsky, I. Sutskever, and G. Hinton. **Imagenet classification with deep convolutional neural networks**. In Neural Information Processing Systems (NIPS), 2012.

A. L. Maas, A. Y. Hannun, and A. Y. Ng. **Rectifier nonlinearities improve neural network acoustic models**. In in ICML Workshop on Deep Learning for Audio, Speech and Language Processing, 2013.

W. Shang, K. Sohn, D. Almeida, and H. Lee. **Understanding and improving convolutional neural networks via concatenated rectified linear units**. CoRR, abs/1603.05201, 2016.

B. Xu, N. Wang, T. Chen, and M. Li. **Empirical evaluation of rectified activations in convolutional network**. CoRR, abs/1505.00853, 2015.