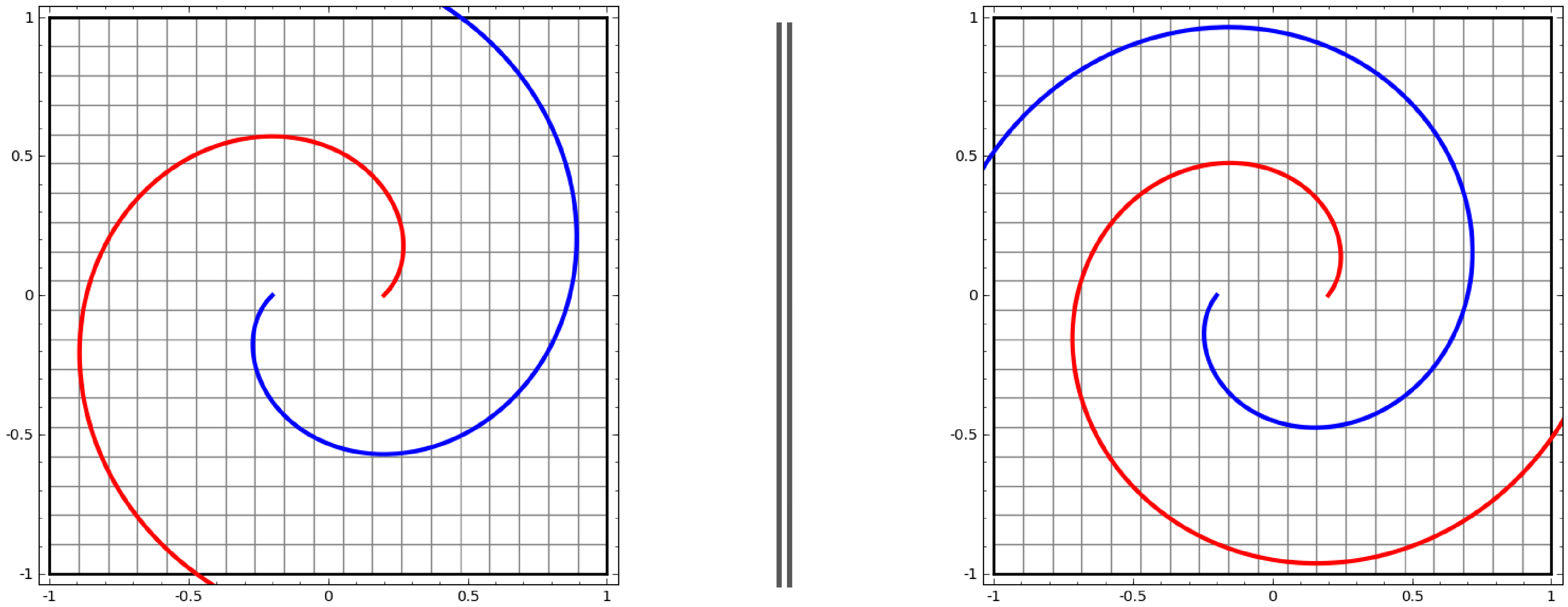
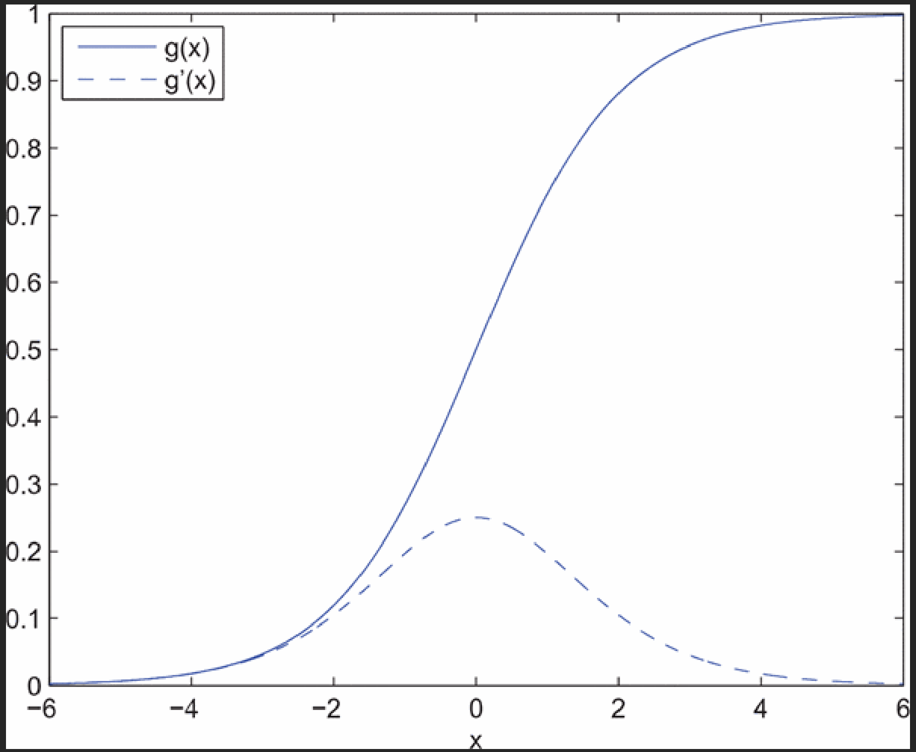
On the Robustness of Activation Functions

By Zachary Harrison

Visualizing Hidden Layers



Sigmoid



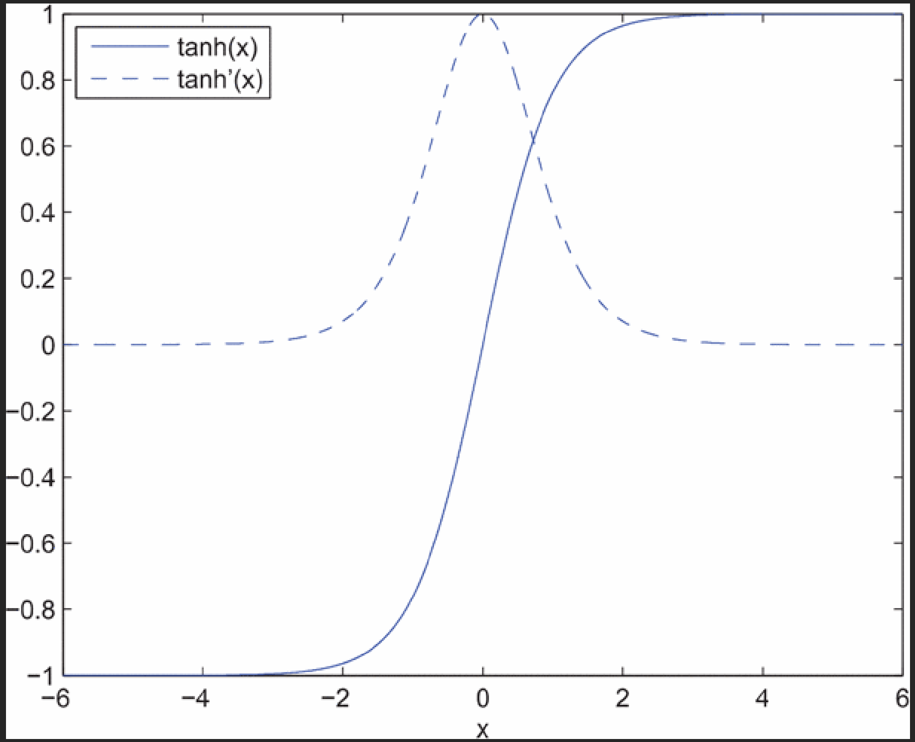
- One of the first activation functions
- Very susceptible to vanishing gradients

$$g(x) = \frac{1}{1 + e^{-x}}$$

$$g'(x) = \frac{e^{-x}}{(1 + e^{-x})^2}$$

Tanh

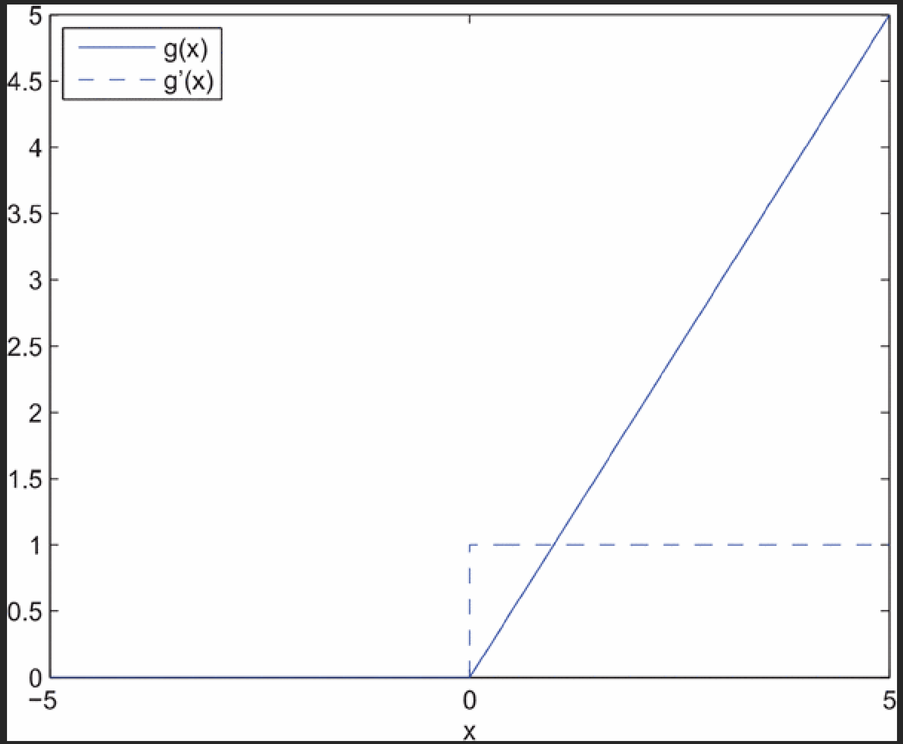
- Similar to Sigmoid
- Also susceptible to vanishing gradients



$$tanh(x) = \frac{sinh(x)}{cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

$$tanh'(x) = 2sigmoid'(2x) - 1 = \frac{4e^{-2x}}{(1+e^{-2x})^2}$$

ReLU

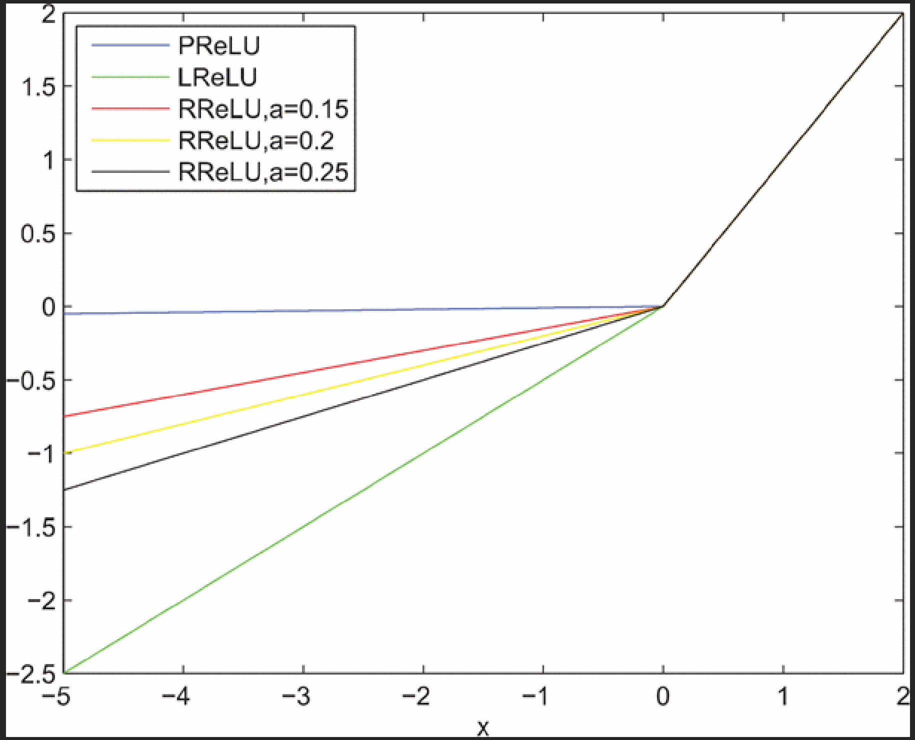


- **Extremely** fast computation
- Much less susceptible to vanishing gradients

$$g(x) = max(0, x) = \begin{cases} x & if\, x \geq 0 \\ 0 & if\, x < 0 \end{cases}$$

$$g'(x) = \begin{cases} 1 & if\, x \geq 0 \\ 0 & if\, x < 0 \end{cases}$$

ReLU's Wacky Family

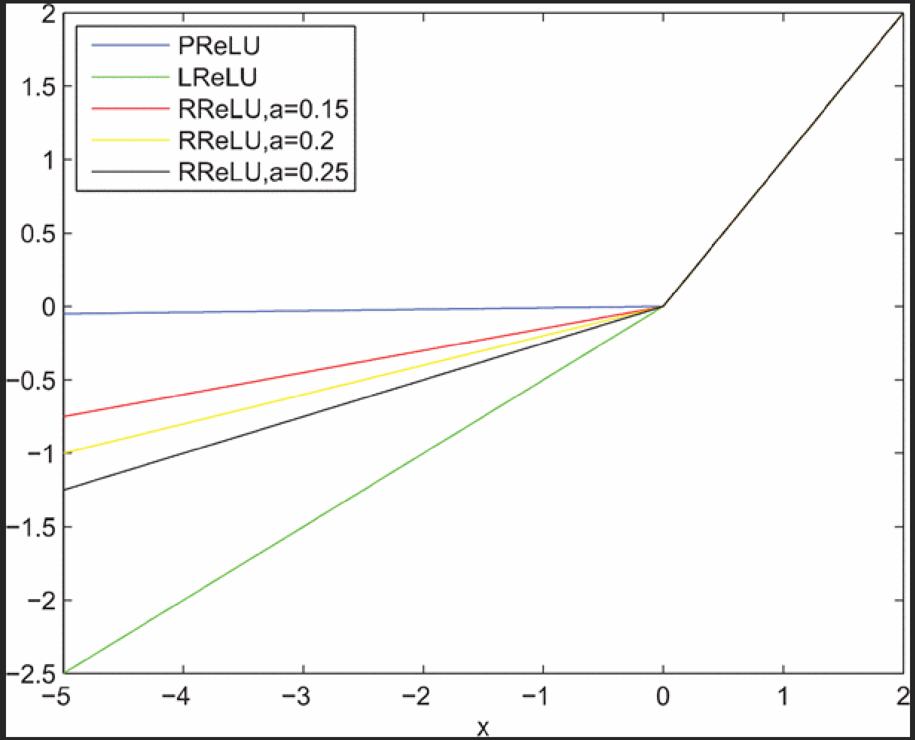


- LReLU or Leaky ReLU allows for small negative values
- PReLU is like Leaky ReLU, but has a learnable parameter ***a*** instead of a constant 0.01.

$$g(x) = \begin{cases} x & if \;\; x \geq 0 \\ ax & if \;\; x < 0 \end{cases}$$

$$g'(x) = \frac{e^{-x}}{(1 + e^{-x})^2}$$

ReLU's Wacky Family



- LReLU allows for small negative values.
- PReLU is like Leaky ReLU, but has a learnable parameter ***a*** instead of a constant 0.01.
- In RReLU, the slopes are randomized during training and fixed during testing.

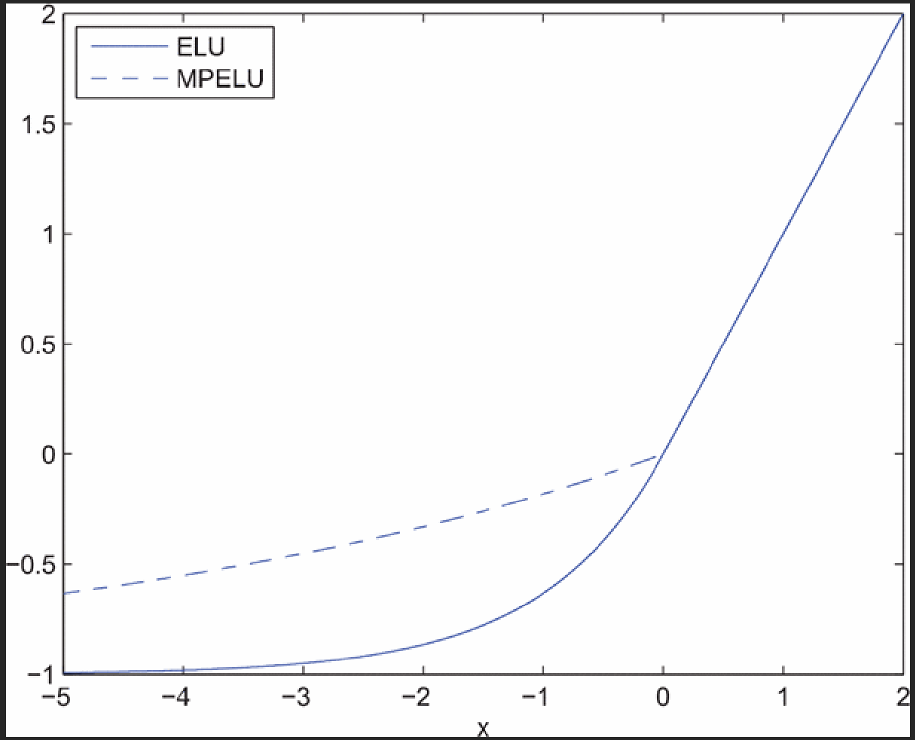
LReLU (Leaky ReLU)

$$g(x) = \begin{cases} x & if\ x \geq 0 \\ 0.01x & if\ x < 0 \end{cases}$$

PReLU (Parametric Rectified Linear Unit)

$$g(x) = \begin{cases} x & if\ x \geq 0 \\ ax & if\ x < 0 \end{cases}$$

ELU and MPELU



- ELU is more robust to the input perturbation or noise due to its convergence property as x approaches negative infinity.
- Because MPELU can become ReLU, LReLU, PReLU, or ELU through training, it is usually considered better.

ELU (Exponential Linear Unit)

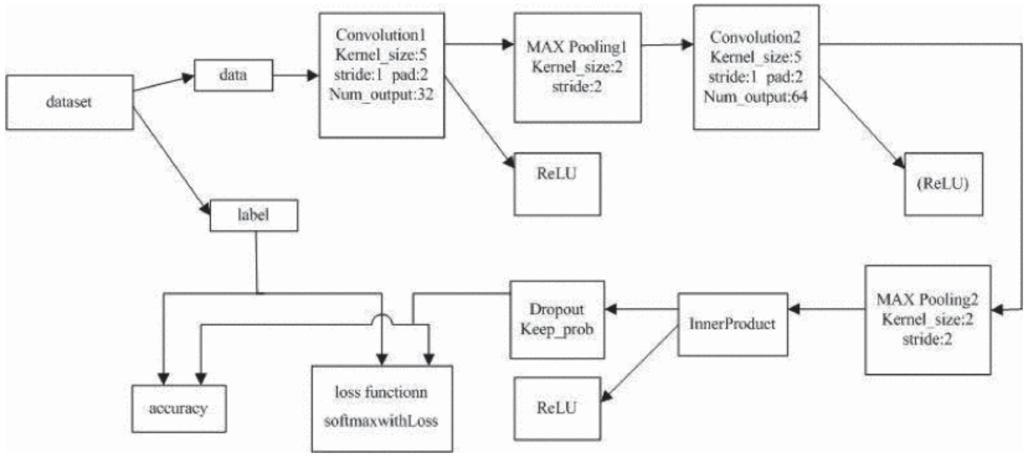
$$g(x) = \begin{cases} x & if \;\; x > 0 \\ \alpha(e^{x} - 1) & if \;\; x \leq 0 \end{cases}$$

MPELU (Multiple Parametric Exponential Linear Unit

$$g(x) = \begin{cases} x & if \;\; x > 0 \\ \alpha(e^{\beta x} - 1) & if \;\; x \leq 0 \end{cases}$$

Results from DCNN

- Using the model architecture defined in the top right, each activation type's performance was recorded.
- It's important to note that models only had 20,000 iterations on their training set of 1,000 28x28 grayscale images.



Activation function	Parameter	Error (%)
Sigmoid	-	1.15
Tanh	-	1.12
ReLU	-	0.8
RReLU	$a = 0.5$	0.99
ELU	$\alpha = 1$	1.1

Sigmoid revisited

- A huge innovation that ELU brought was this convergence property as x approaches negative infinity. The reason this increases robustness is because it reduces outliers' ability to change the model during the training process.
- Both Sigmoid and Tanh have this property, but they are widely considered inferior to all ReLU's family of activation functions. Why is that?
- That's what I'm going to research!

References

1. Activation functions and their characteristics in deep neural networks
2. The Power of Approximating: a Comparison of Activation Functions
3. How to Choose an Activation Function
4. Small nonlinearities in activation functions create bad local minima in neural networks
5. Neural Networks, Manifolds, and Topology

Thank you for listening!