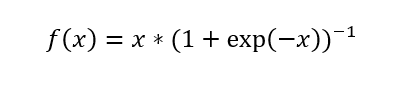
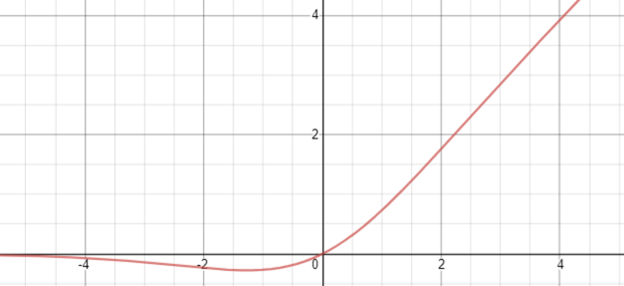
**Swish: Booting ReLU from the Activation Function Throne**

<https://towardsdatascience.com/swish-booting-relu-from-the-activation-function-throne-78f87e5ab6eb>

Activation functions have a long history. First, the sigmoid function was chosen for its easy derivative, range between 0 and 1, and smooth probabilistic shape. The tanh function was also considered as being an alternative to the sigmoid function, fitted on a scale between -1 and 1, but these classical activation functions have been replaced with ReLU. The Rectified Linear Unit (ReLU) is currently the most popular activation function because the gradient can flow when the input to the ReLU function is positive. Its simplicity and effectiveness has pushed ReLU and branching methods like Leaky ReLU and Parametrized ReLU past the sigmoid and tanh units.

Formally stated, the Swish activation function is…



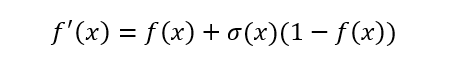


The Swish activation function.

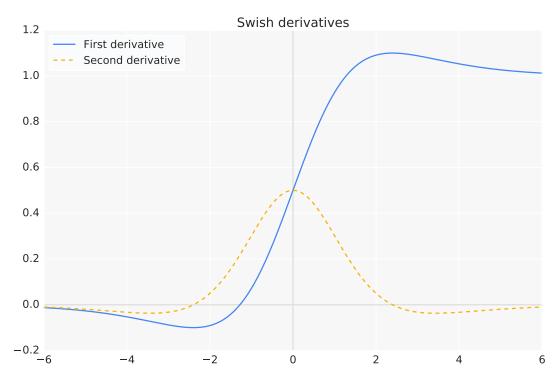
Like ReLU, Swish is bounded below (meaning as *x* approaches negative infinity, *y* approaches some constant value) but unbounded above (meaning as *x*approaches positive infinity, *y* approaches infinity). However, unlike ReLU, Swish is *smooth* (it does not have sudden changes of motion or a vertex):

Additionally, Swish is non-monotonic, meaning that there is not always a singularly and continually positive (or negative) derivative throughout the entire function. (Restated, the Swish function has a negative derivative at certain points and a positive derivative at other points, instead of only a positive derivative at all points, like Softplus or Sigmoid.

The derivative of the Swish function is…



The first and second derivatives of Swish, plotted:



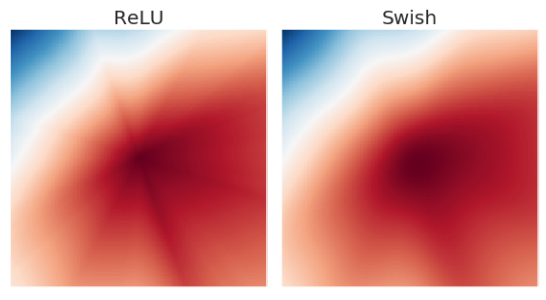
**Properties of Swish**

Unboundedness is desirable for activation functions because it avoids a slow training time during near-zero gradients — functions like sigmoid or tanh are bounded above and below, so the network needs to be carefully initialized to stay within the limitations of these functions.

The ReLU function is an improvement over tanh because it is bounded above — this property is so important that every successful activation function after ReLU is unbounded above.

Being bounded below may be advantageous because of strong regularization — functions that approach zero in a limit to negative infinity are great at regularization because large negative inputs are discarded. This is important at the beginning of training when large negative activation inputs are common.

Additionally, smoothness helps optimize and generalize the neural network. In the output landscapes below, it is obvious that ReLU’s output landscape is sharp and jarring because of its non-smooth nature, whereas the Swish network landscape is much smoother.



The output landscape smoothness directly correlates with the error landscape. A smoother error space is easier to traverse and find a minima — consider it like walking in the jarring range of altitudes of the Himalayan range versus walking in the smooth, rolling hills of the English countryside.

# ****Swish Performance****

The authors of the Swish paper compare Swish to the following other activation functions:

* Leaky ReLU, where f(x) = x if x ≥ 0, and ax if x< 0, where a = 0.01. This allows for a small amount of information to flow when x < 0, and is considered to be an improvement over ReLU.
* Parametric ReLU is the same as Leaky Relu, but a is a learnable parameter, initialized to 0.25.
* Softplus, defined by f(x) = log(1 + exp(x)), is a smooth function with properties like Swish, but is strictly positive and monotonic.
* Exponential Linear Unit (ELU), defined by f(x) = x if x ≥ 0 and a(exp(x) — 1) if x < 0 where a = 1.
* Scaled Exponential Linear Unit (SELU), identical to ELU but with the output multiplied by a value s.