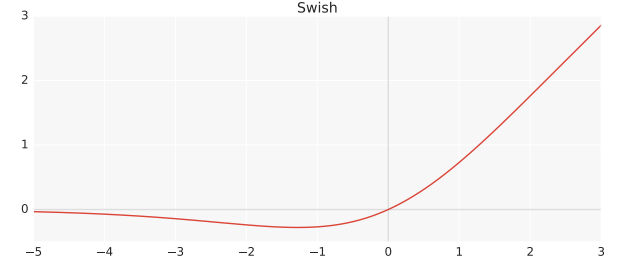
**Swish Activation Function by Google**

<https://medium.com/@neuralnets/swish-activation-function-by-google-53e1ea86f820>

 Currently, the most successful and widely-used activation function is the **Rectified Linear Unit(ReLU)**, which is **f(x)=max(0,x)**. Although various alternatives to ReLU have been proposed, none have managed to replace it due to inconsistent gains. So Google Brain Team has proposed a new activation function, named **Swish**, which is simply **f(x) = x · sigmoid(x)**. Their experiments show that **Swish tends to work better than ReLU on deeper models across a number of challenging data sets**. For example, simply replacing ReLUs with Swish units improves top-1 classification accuracy on [ImageNet](http://www.image-net.org/) by 0.9% for Mobile NASNetA and 0.6% for [Inception-ResNet-v2](https://github.com/Trangle/mxnet-inception-v4/blob/master/inception-resnet-v2.pdf). **The simplicity of Swish and its similarity to ReLU make it easy for practitioners to replace ReLUs with Swish units in any neural network.**



Swish Activation Function Image [Source](https://lazyprogrammer.me/wp-content/uploads/2017/10/Screen-Shot-2017-10-18-at-2.39.55-PM.png)

With ReLU, the consistent problem is that its derivative is **0** for half of the values of the input ***x*** in [**ramp Function**](https://en.wikipedia.org/wiki/Ramp_function), i.e. **f(x)=max(0,x)**. As their parameter update algorithm, they have used **Stochastic Gradient Descent** and if the parameter itself is **0**, then that parameter will never be updated as it just assigns the parameter back to itself, leading close to [**40% Dead Neurons in the Neural network environment**](https://www.theseus.fi/bitstream/handle/10024/123282/Naumetc_Daniil.pdf?sequence=1) when θ=θ. Various substitutes like [**Leaky ReLU**](https://en.wikipedia.org/wiki/Rectifier_(neural_networks)#Leaky_ReLUs) or [**SELU (Self-Normalizing Neural Networks)**](https://arxiv.org/pdf/1706.02515.pdf) have unsuccessfully tried to devoid it of this issue but now there seems to be a revolution for good.

**Swish is a smooth, non-monotonic function** that consistently matches or outperforms ReLU on deep networks applied to a variety of challenging domains such as Image classification and Machine translation. It is **unbounded above and bounded below** & it is the non-monotonic attribute that actually creates the difference. **With self-gating, it requires just a scalar input**whereas in multi-gating scenario, it would require multiple two-scalar input. It has been **inspired by the use of Sigmoid function in LSTM (Hochreiter & Schmidhuber, 1997) and**[**Highway networks (Srivastava et al., 2015)**](http://people.idsia.ch/~rupesh/very_deep_learning/)**where** ‘*self-gated*’ means that the gate is actually the ‘*sigmoid*’ of activation itself.

We can train **deeper Swish networks than ReLU networks** when using [BatchNorm (Ioffe & Szegedy, 2015)](http://people.ee.duke.edu/~lcarin/Zhao12.17.2015.pdf) despite having gradient squishing property. With [MNIST data set](http://yann.lecun.com/exdb/mnist/), when Swish and ReLU are compared, both activation functions achieve similar performances up to 40 layers. However, **Swish outperforms ReLU by a large margin in the range between 40 and 50 layers**when optimization becomes difficult. In very deep networks, Swish achieves higher test accuracy than ReLU. **In terms of batch size**, the performance of both activation functions decrease as batch size increases, potentially due to [sharp minima (Keskar et al., 2017)](https://arxiv.org/abs/1703.04933). However, **Swish outperforms ReLU on every batch size, suggesting that the performance difference between the two activation functions remains even when varying the batch size**.