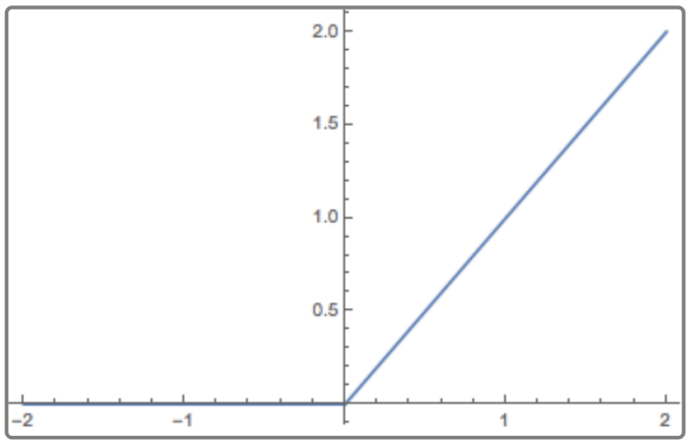
Project ML

Piazza • Ask. Answer. Explore. Whenever.

一文概览深度学习中的激活函数 | 机器之心
浅谈神经网络中激活函数的设计 - 科学空间|Scientific Spaces

utstat.toronto.edu/?page_id=2269
Swish in depth: A comparison of Swish & ReLU on CIFAR-10
https://medium.com/@jaiyamsharma/experiments-with-swish-activation-function-on-mnist-dataset-fc89a8c79ff7
https://medium.com/@shahariarrabby/mnist-kaggle-submission-with-cnn-keras-switch-activation-62108f9463df
Experiments with SWISH activation function on MNIST dataset
http://www.jianshu.com/p/95e3630ad9e2

其图像是

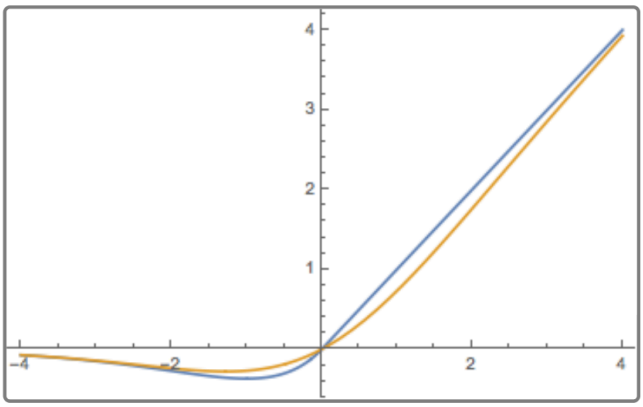


这是个分段线性函数，显然其导数在正半轴为1，负半轴为0，这样它在整个实数域上有一半的空间是不饱和的。相比之下，sigmoid函数几乎全部区域都是饱和的（饱和区间占比趋于1，饱和的定义是导数很接近0）。

ReLu是分段线性函数，它的非线性性很弱，因此网络一般要做得很深。但这正好迎合了我们的需求，因为在同样效果的前提下，往往深度比宽度更重要，更深的模型泛化能力更好。所以自从有了Relu激活函数，各种很深的模型都被提出来了，一个标志性的事件是应该是VGG模型和它在ImageNet上取得的成功，至于后来的发展就不详细说了。

其实样子跟Swish差不多，思路大概是正半轴维持x，负半轴想一个先降后升还趋于0的函数，我想到了xe^{-x}，稍微调整就得到了这个函数了。在我的一些模型中，它的效果甚至比Swish要好些（在我的问答模型上）。当然我只做了一点实验，就不可能有那么多精力和算力去做对比实验了。

与Swish的比较，橙色是Swish。



要提醒的是，如果要用这个函数，不能直接用这个形式写，因为e^x的计算可能溢出，一种不会溢出的写法是

$$\max(x, x \cdot e^{-|x|}) \tag{6}$$

或者用ReLu函数写成

$$x + \text{relu}(x \cdot e^{-|x|} - x) \tag{7}$$

们作业上的勾（√）？

IV. 改进思路

Swish函数惹来了一些争议，有些人认为Google大脑小题大作了，简单改进一个激活函数，小团队就可以玩了，Google大脑这些大团队应该往更高端的方向去做。但不过怎样，Google大脑做了很多实验，结果都表明Swish优于ReLu。那么我们就需要思考一下，背后的原因是什么呢？

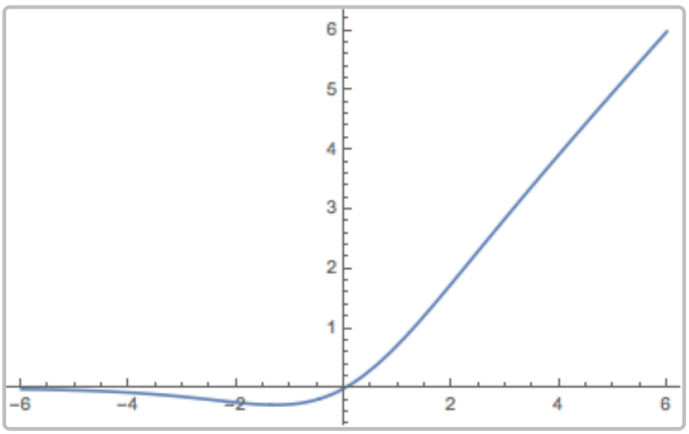
下面的分析纯属博主的主观臆测，目前没有理论或实验上的证明，请读者斟酌阅读。我觉得，Swish优于ReLu的一个很重要的原因是跟初始化有关。

Swish在原点附近不是饱和的，只有负半轴远离原点区域才是饱和的，而ReLu在原点附近也有一半的空间是饱和的。而我们在训练模型时，一般采用的初始化参数是均匀初始化或者正态分布初始化，不管是哪种初始化，其均值一般都是0，也就是说，初始化的参数有一半处于ReLu的饱和区域，这使得刚开始时就有一半的参数没有利用上。特别是由于诸如BN之类的策略，输出都自动近似满足均值为0的正态分布，因此这些情况都有一半的参数位于ReLu的饱和区。相比之下，Swish好一点，因为它在负半轴也有一定的不饱和区，所以参数的利用率更大。

前面说到，就连笔者都曾思考过Swish激活函数，但没有深入研究，原因之一是它不够简洁漂亮，甚至我觉得它有点丑～～看到Swish的实验结果那么好，我想有没有类似的、更加好看的激活函数呢？我想到了一个

$$x \cdot \min(1, e^x) \tag{5}$$

其图像如下



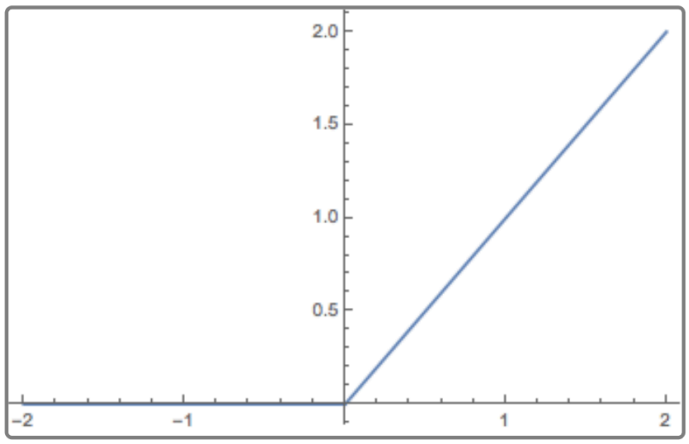
团队的测试结果表明该函数在很多模型都优于ReLu。

从图像上来看，Swish函数跟ReLu差不多，唯一区别较大的是接近于0的负半轴区域。马后炮说一句，其实这个激活函数就连笔者也思考过，因为这跟facebook提出的GLU激活函数是类似的，GLU激活函数为

$$(\boldsymbol{W}_1\boldsymbol{x}+\boldsymbol{b}_1)\otimes\sigma(\boldsymbol{W}_2\boldsymbol{x}+\boldsymbol{b}_2) \tag{4}$$

也就是说，分别训练两组参数，其中一组用sigmoid激活，然后乘上另一组，这里的$\sigma(\boldsymbol{W}_2\boldsymbol{x}+\boldsymbol{b}_2)$就称为“门”，也就是GLU中的G的意思（gate）。而Swish函数则相当于两组参数都取同样的，只训练一组参数。

其图像是



这是个分段线性函数，显然其导数在正半轴为1，负半轴为0，这样它在整个实数域上有一半的空间是不饱和的。相比之下，sigmoid函数几乎全部区域都是饱和的（饱和区间占比趋于1，饱和的定义是导数很接近0）。

ReLu是分段线性函数，它的非线性性很弱，因此网络一般要做得很深。但这正好迎合了我们的需求，因为在同样效果的前提下，往往深度比宽度更重要，更深的模型泛化能力更好。所以自从有了Relu激活函数，各种很深的模型都被提出来了，一个标志性的事件是应该是VGG模型和它在ImageNet上取得的成功，至于后来的发展就不详细说了。