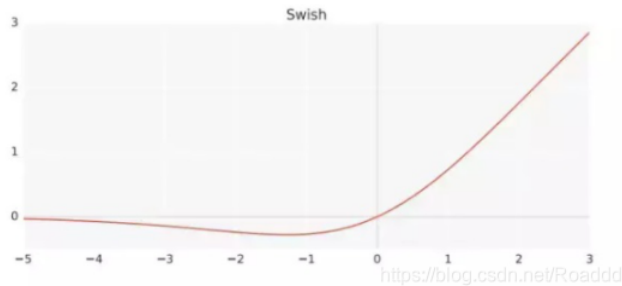


【激活函数】Swish激活函数详解

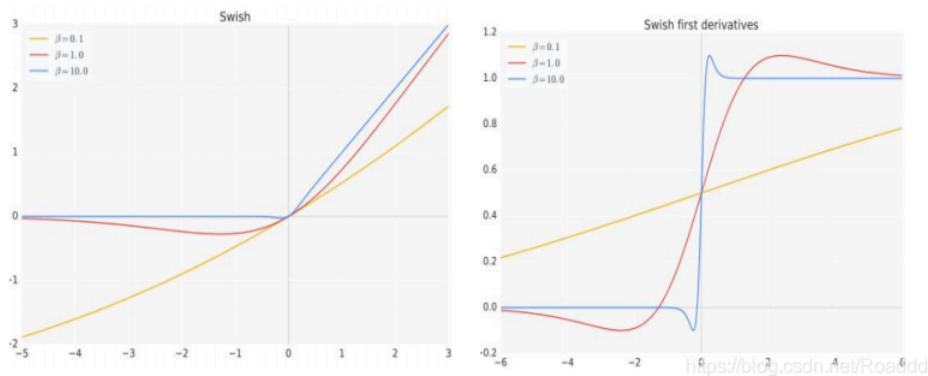
Swish: Self-gated activation function (2017)

• Swish图像:



• Swish公式:

$$f(x) = x * \text{sigmoid}(\beta x)$$



当 $\beta=1$ 时:

$$f(x) = x \cdot \sigma(x)$$

• Swish导数:

$$\begin{aligned} f'(x) &= \sigma(x) + x \cdot \sigma(x) \cdot (1 - \sigma(x)) \\ &= x \cdot \sigma(x) + \sigma(x)(1 - x \cdot \sigma(x)) \\ &= f(x) + \sigma(x) \cdot (1 - f(x)) \end{aligned}$$

• Swish特点:

- Unbounded above: Avoid saturation**
和 ReLU 一样，没有上边界，因此不会出现梯度饱和的现象。
- Bounded below: Strong regularization(especially for large negative numbers)**
有下边界，可以产生更强的正则化效果。(x 左半轴慢慢趋近于 0)
- Non-monotonicity: No died neuron for small negative**
非单调。
- Smoothness: Easier to train:less sensitive to init. & LR**
处处连续且可导，更容易训练。

注意：不稳定，在不同的任务当中可能产生不同的效果，不一定强于 ReLU。

• 关于正则化效果

x轴越靠近左半轴，纵坐标的值越小，甚至接近于0，如果x值是-10，那么经过激活之后的值接近于0，那么就可以一定程度上过滤掉一部分信息，起到 **正则化** 的效果。

代码实现:

```
1 def swish(x):  
2     return x * F.sigmoid(x)  
3  
4 # used as class:  
5 class Swish(nn.Module):  
6     def __init__(self):  
7         super().__init__()  
8  
9     def forward(self, x):  
10        return x * F.sigmoid(x)
```