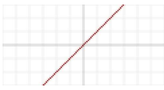
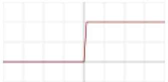
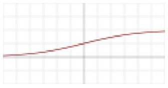
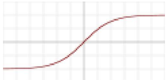
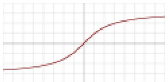
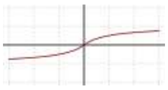
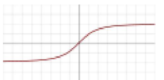
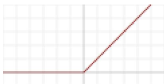
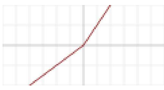
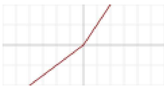
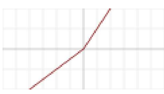
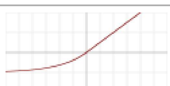
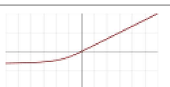
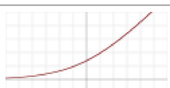
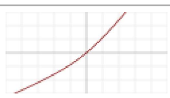
# 激活函数

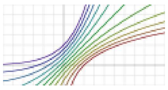
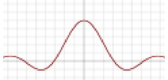
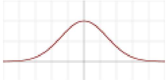
维基百科，自由的百科全书

在计算网络中， 一个节点的激活函数(Activation Function)定义了该节点在给定的输入或输入的集合下的输出。标准的计算机芯片电路可以看作是根据输入得到开（1）或关（0）输出的数字电路激活函数。这与神经网络中的线性感知机的行为类似。然而，只有非线性激活函数才允许这种网络仅使用少量节点来计算非平凡问题。 在人工神经网络中，这个功能也被称为传递函数。

## 函数

下表列出了几个激活函数，它们的输入为单一变量。

| 名称 | 函数图形 | 方程式 | 导数 | 区间 | Order of continuity |
|---|---|---|---|---|---|
| 恒等函数 |  | $f(x)=x$ | $f'(x)=1$ | $(-\infty,\infty)$ | $C^\infty$ |
| 单位阶跃函数 |  | $f(x)=\begin{cases}0 & \text{for } x<0\\ 1 & \text{for } x\ge 0\end{cases}$ | $f'(x)=\begin{cases}0 & \text{for } x\ne 0\\ ? & \text{for } x=0\end{cases}$ | $\{0,1\}$ | $C^{-1}$ |
| 逻辑函数 (也被称为S函数) |  | $f(x)=\sigma(x)=\frac{1}{1+e^{-x}}$ [1] (https://zh.wikipedia.org/wiki/%E6%BF%80%E6%B4%BB%E5%87%BD%E6%95%B0#endnote_logistic1) | $f'(x)=f(x)(1-f(x))$ | $(0,1)$ | $C^\infty$ |
| 双曲正切函数 |  | $f(x)=\tanh(x)=\frac{(e^x-e^{-x})}{(e^x+e^{-x})}$ | $f'(x)=1-f(x)^2$ | $(-1,1)$ | $C^\infty$ |
| 反正切函数 |  | $f(x)=\tan^{-1}(x)$ | $f'(x)=\frac{1}{x^2+1}$ | $\left(-\frac{\pi}{2},\frac{\pi}{2}\right)$ | $C^\infty$ |
| Softsign函数[1][2] |  | $f(x)=\frac{x}{1+\lvert x\rvert}$ | $f'(x)=\frac{1}{(1+\lvert x\rvert)^2}$ | $(-1,1)$ | $C^1$ |
| 反平方根函数 (ISRU)[3] |  | $f(x)=\frac{x}{\sqrt{1+\alpha x^2}}$ | $f'(x)=\left(\frac{1}{\sqrt{1+\alpha x^2}}\right)^3$ | $\left(-\frac{1}{\sqrt{\alpha}},\frac{1}{\sqrt{\alpha}}\right)$ | $C^\infty$ |
| 线性整流函数 (ReLU) |  | $f(x)=\begin{cases}0 & \text{for } x<0\\ x & \text{for } x\ge 0\end{cases}$ | $f'(x)=\begin{cases}0 & \text{for } x<0\\ 1 & \text{for } x\ge 0\end{cases}$ | $[0,\infty)$ | $C^0$ |
| 带泄露线性整流函数(Leaky ReLU) |  | $f(x)=\begin{cases}0.01x & \text{for } x<0\\ x & \text{for } x\ge 0\end{cases}$ | $f'(x)=\begin{cases}0.01 & \text{for } x<0\\ 1 & \text{for } x\ge 0\end{cases}$ | $(-\infty,\infty)$ | $C^0$ |
| 参数化线性整流函数(PReLU)[4] |  | $f(\alpha,x)=\begin{cases}\alpha x & \text{for } x<0\\ x & \text{for } x\ge 0\end{cases}$ | $f'(\alpha,x)=\begin{cases}\alpha & \text{for } x<0\\ 1 & \text{for } x\ge 0\end{cases}$ | $(-\infty,\infty)$ | $C^0$ |
| 带泄露随机线性整流函数 (RReLU)[5] |  | $f(\alpha,x)=\begin{cases}\alpha x & \text{for } x<0\\ x & \text{for } x\ge 0\end{cases}$ [2] (https://zh.wikipedia.org/wiki/%E6%BF%80%E6%B4%BB%E5%87%BD%E6%95%B0#endnote_alpha_random) | $f'(\alpha,x)=\begin{cases}\alpha & \text{for } x<0\\ 1 & \text{for } x\ge 0\end{cases}$ | $(-\infty,\infty)$ | $C^0$ |
| 指数线性函数 (ELU)[6] |  | $f(\alpha,x)=\begin{cases}\alpha(e^x-1) & \text{for } x<0\\ x & \text{for } x\ge 0\end{cases}$ | $f'(\alpha,x)=\begin{cases}f(\alpha,x)+\alpha & \text{for } x<0\\ 1 & \text{for } x\ge 0\end{cases}$ | $(-\alpha,\infty)$ | $\begin{cases}C_1 & \text{when } \alpha=\\ C_0 & \text{otherwise}\end{cases}$ |
| 扩展指数线性函数(SELU)[7] | | $f(\alpha,x)=\lambda\begin{cases}\alpha(e^x-1) & \text{for } x<0\\ x & \text{for } x\ge 0\end{cases}$ with $\lambda=1.0507$ and $\alpha=1.67326$ | $f'(\alpha,x)=\lambda\begin{cases}\alpha(e^x) & \text{for } x<0\\ 1 & \text{for } x\ge 0\end{cases}$ | $(-\lambda\alpha,\infty)$ | $C^0$ |
| S 型线性整流激活函数 (SReLU)[8] | | $f_{t_l,a_l,t_r,a_r}(x)=\begin{cases}t_l+a_l(x-t_l) & \text{for } x\le t_l\\ x & \text{for } t_l<x<t_r\\ t_r+a_r(x-t_r) & \text{for } x\ge t_r\end{cases}$ $t_l,a_l,t_r,a_r$ are parameters. | $f'_{t_l,a_l,t_r,a_r}(x)=\begin{cases}a_l & \text{for } x\le t_l\\ 1 & \text{for } t_l<x<t_r\\ a_r & \text{for } x\ge t_r\end{cases}$ | $(-\infty,\infty)$ | $C^0$ |
| 反平方根线性函数(ISRLU)[3] |  | $f(x)=\begin{cases}\frac{x}{\sqrt{1+\alpha x^2}} & \text{for } x<0\\ x & \text{for } x\ge 0\end{cases}$ | $f'(x)=\begin{cases}\left(\frac{1}{\sqrt{1+\alpha x^2}}\right)^3 & \text{for } x<0\\ 1 & \text{for } x\ge 0\end{cases}$ | $\left(-\frac{1}{\sqrt{\alpha}},\infty\right)$ | $C^2$ |
| 自适应分段线性函数(APL)[9] | | $f(x)=\max(0,x)+\sum_{s=1}^{S}a_i^s\max(0,-x+b_i^s)$ | $f'(x)=H(x)-\sum_{s=1}^{S}a_i^sH(-x+b_i^s)$ [3] (https://zh.wikipedia.org/wiki/%E6%BF%80%E6%B4%BB%E5%87%BD%E6%95%B0#endnote_heaviside) | $(-\infty,\infty)$ | $C^0$ |
| SoftPlus函数[10] |  | $f(x)=\ln(1+e^x)$ | $f'(x)=\frac{1}{1+e^{-x}}$ | $(0,\infty)$ | $C^\infty$ |
| 弯曲恒等函数 |  | $f(x)=\frac{\sqrt{x^2+1}-1}{2}+x$ | $f'(x)=\frac{x}{2\sqrt{x^2+1}}+1$ | $(-\infty,\infty)$ | $C^\infty$ |
| Sigmoid-weighted linear unit (SiLU)[11] (也被称为Swish[12]) | | $f(x)=x\cdot\sigma(x)$ [4] (https://zh.wikipedia.org/wiki/%E6%BF%80%E6%B4%BB%E5%87%BD%E6%95%B0#endnote_logistic2) | $f'(x)=f(x)+\sigma(x)(1-f(x))$ [5] (https://zh.wikipedia.org/wiki/%E6%BF%80%E6%B4%BB%E5%87%BD%E6%95%B0#endnote_logistic3) | $[\approx -0.28,\infty)$ | $C^\infty$ |

| 名称 | 函数图形 | 方程式 | 导数 | 区间 | Order of continuity |
|---|---|---|---|---|---|
| SoftExponential函数[13] |  | $f(\alpha,x)=\begin{cases}-\frac{\ln(1-\alpha(x+\alpha))}{\alpha} & \text{for } \alpha<0\\ x & \text{for } \alpha=0\\ \frac{e^{\alpha x}-1}{\alpha}+\alpha & \text{for } \alpha>0\end{cases}$ | $f'(\alpha,x)=\begin{cases}\frac{1}{1-\alpha(\alpha+x)} & \text{for } \alpha<0\\ e^{\alpha x} & \text{for } \alpha\geq 0\end{cases}$ | $(-\infty,\infty)$ | $C^\infty$ |
| 正弦函数 |  | $f(x)=\sin(x)$ | $f'(x)=\cos(x)$ | $[-1,1]$ | $C^\infty$ |
| Sinc函数 |  | $f(x)=\begin{cases}1 & \text{for } x=0\\ \frac{\sin(x)}{x} & \text{for } x\neq 0\end{cases}$ | $f'(x)=\begin{cases}0 & \text{for } x=0\\ \frac{\cos(x)}{x}-\frac{\sin(x)}{x^2} & \text{for } x\neq 0\end{cases}$ | $[\approx -.217234,1]$ | $C^\infty$ |
| 高斯函数 |  | $f(x)=e^{-x^2}$ | $f'(x)=-2xe^{-x^2}$ | $(0,1]$ | $C^\infty$ |

^ 此处$H$是单位阶跃函数。
^ $\alpha$是在训练时间从均匀分布中抽取的随机变量，并且在测试时间固定为分布的期望值。
^ ^ ^ 此处$\sigma$是逻辑函数。

下表列出了几个激活函数，它们的输入为多个变量。

| 名称 | 方程式 | 导数 | 区间 | Order of continuity |
|---|---|---|---|---|
| Softmax函数 | $f_i(\vec{x})=\frac{e^{x_i}}{\sum_{j=1}^J e^{x_j}}$ for $i$ = 1, …, $J$ | $\frac{\partial f_i(\vec{x})}{\partial x_j}=f_i(\vec{x})(\delta_{ij}-f_j(\vec{x}))$[6] (https://zh.wikipedia.org/wiki/%E6%BF%80%E6%B4%BB%E5%87%BD%E6%95%B0#endnote_kronecker_delta) | $(0,1)$ | $C^\infty$ |
| Maxout函数[14] | $f(\vec{x})=\max_i x_i$ | $\frac{\partial f}{\partial x_j}=\begin{cases}1 & \text{for } j=\underset{i}{\operatorname{argmax}}\, x_i\\ 0 & \text{for } j\neq\underset{i}{\operatorname{argmax}}\, x_i\end{cases}$ | $(-\infty,\infty)$ | $C^0$ |

^ 此处$\delta$是克罗内克δ函数。

## 参见

- 逻辑函数
- 线性整流函数
- Softmax函数
- 人工神经网络
- 深度学习

## 参考资料


1. Bergstra, James; Desjardins, Guillaume; Lamblin, Pascal; Bengio, Yoshua. Quadratic polynomials learn better image features". Technical Report 1337. Département d' Informatique et de Recherche Opérationnelle, Université de Montréal. 2009. （原始内容存档于2018-09-25）.
2. Glorot, Xavier; Bengio, Yoshua, Understanding the difficulty of training deep feedforward neural networks (PDF), International Conference on Artificial Intelligence and Statistics (AISTATS' 10), Society for Artificial Intelligence and Statistics, 2010
3. Carlile, Brad; Delamarter, Guy; Kinney, Paul; Marti, Akiko; Whitney, Brian. Improving Deep Learning by Inverse Square Root Linear Units (ISRLUs). 2017-11-09. arXiv:1710.09967 [cs.LG].
4. He, Kaiming; Zhang, Xiangyu; Ren, Shaoqing; Sun, Jian. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. 2015-02-06. arXiv:1502.01852 [cs.CV].
5. Xu, Bing; Wang, Naiyan; Chen, Tianqi; Li, Mu. Empirical Evaluation of Rectified Activations in Convolutional Network. 2015-05-04. arXiv:1505.00853 [cs.LG].
6. Clevert, Djork-Arné; Unterthiner, Thomas; Hochreiter, Sepp. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). 2015-11-23. arXiv:1511.07289 [cs.LG].
7. Klambauer, Günter; Unterthiner, Thomas; Mayr, Andreas; Hochreiter, Sepp. Self-Normalizing Neural Networks. 2017-06-08. arXiv:1706.02515 [cs.LG].
8. Jin, Xiaojie; Xu, Chunyan; Feng, Jiashi; Wei, Yunchao; Xiong, Junjun; Yan, Shuicheng. Deep Learning with S-shaped Rectified Linear Activation Units. 2015-12-22. arXiv:1512.07030 [cs.CV].
9. Forest Agostinelli; Matthew Hoffman; Peter Sadowski; Pierre Baldi. Learning Activation Functions to Improve Deep Neural Networks. 21 Dec 2014. arXiv:1412.6830 [cs.NE].
10. Glorot, Xavier; Bordes, Antoine; Bengio, Yoshua. Deep sparse rectifier neural networks (PDF). International Conference on Artificial Intelligence and Statistics. 2011.
11. Sigmoid-Weighted Linear Units for Neural Network Function Approximation in Reinforcement Learning (https://arxiv.org/abs/1702.03118)
12. Searching for Activation Functions (https://arxiv.org/abs/1710.05941)
13. Godfrey, Luke B.; Gashler, Michael S. A continuum among logarithmic, linear, and exponential functions, and its potential to improve generalization in neural networks. 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and

Knowledge Management: KDIR. 2016-02-03, **1602**: 481–486. Bibcode:2016arXiv160201321G. arXiv:1602.01321.

14. Goodfellow, Ian J.; Warde-Farley, David; Mirza, Mehdi; Courville, Aaron; Bengio, Yoshua. Maxout Networks. JMLR WCP. 2013-02-18, **28** (3): 1319–1327. Bibcode:2013arXiv1302.4389G. arXiv:1302.4389.