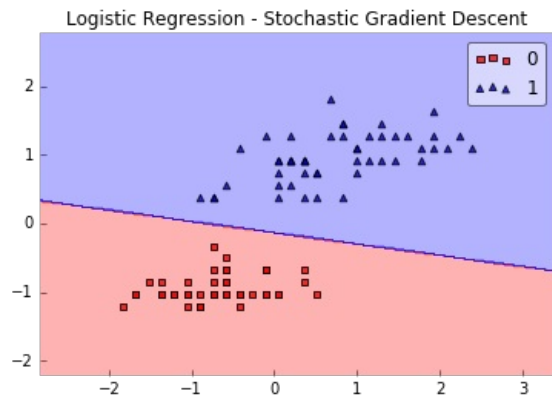


## 浅谈深度学习中的激活函数 - The Activation Function in Deep Learning

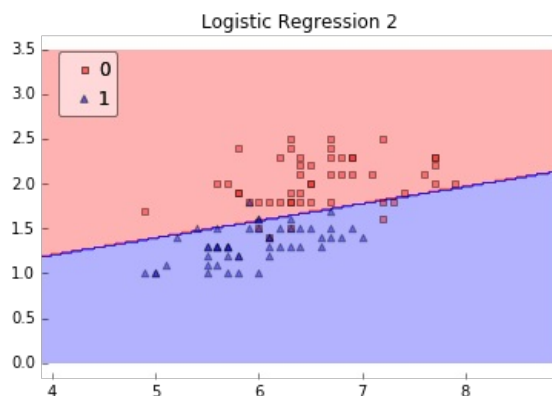
### 激活函数的作用

首先，激活函数不是真的要去激活什么。在神经网络中，激活函数的作用是能够给神经网络加入一些非线性因素，使得神经网络可以更好地解决较为复杂的问题。

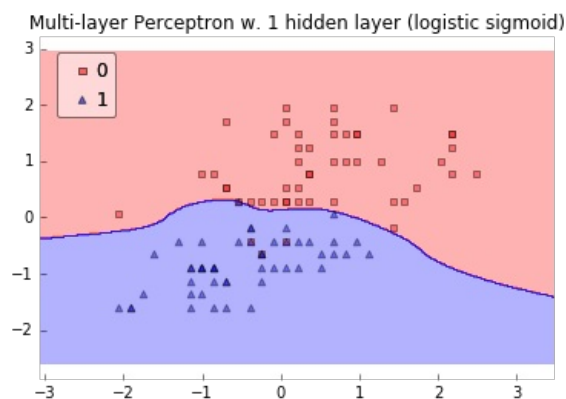
比如在下面的这个问题中：



如上图(图片来源), 在最简单的情况下，数据是线性可分的，只需要一条直线就已经能够对样本进行很好地分类。



但如果情况变得复杂了一点呢？在上图中(图片来源), 数据就变成了线性不可分的情况。在这种情况下，简单的一条直线就已经不能够对样本进行很好地分类了。



于是我们尝试引入非线性的因素，对样本进行分类。

在神经网络中也类似，我们需要引入一些非线性的因素，来更好地解决复杂的问题。而激活函数恰好就是那个能够帮助我们引入非线性因素的存在，使得我们的神经网络能够更好地解决较为复杂的问题。

### 公告

Math + Computer Science = ♥

-

My interests focus on Machine Learning, Data Structure & Algorithm.

昵称: [rgvb178](#)

园龄: [2个月](#)

粉丝: [7](#)

关注: [3](#)

[+加关注](#)

### 搜索

### 随笔分类

[Data Structure & Algorithm\(13\)](#)

[Deep Learning\(3\)](#)

[LeetCode\(8\)](#)

[Machine Learning\(6\)](#)

[MATLAB\(1\)](#)

[Online Judge\(18\)](#)

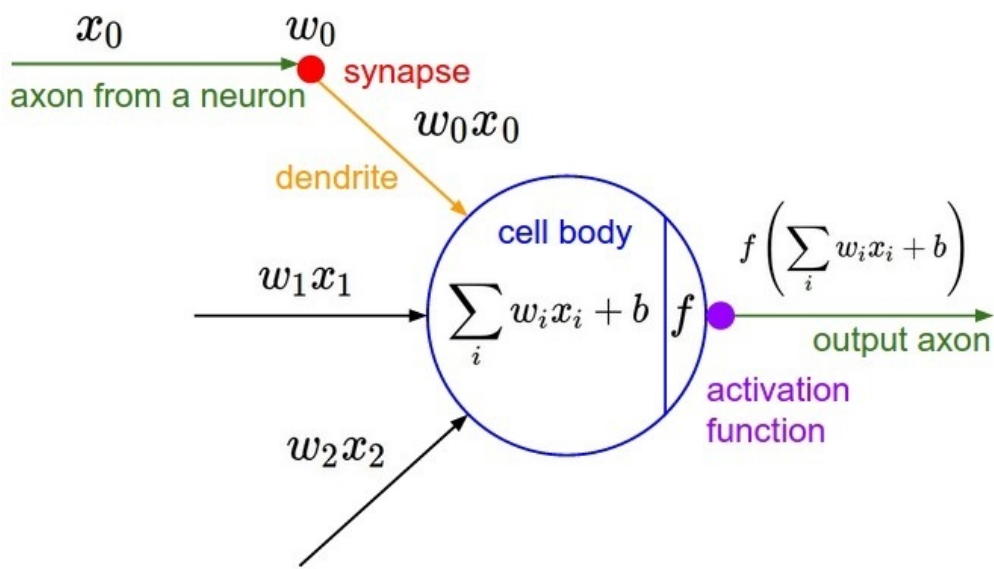
[UVa\(10\)](#)

### 阅读排行榜

1. [UVa 294 - Divisors 解题报告 c语言实现 素数筛法\(304\)](#)
2. [机器学习 \(一\) 单变量线性回归 Linear Regression with One Variable\(297\)](#)
3. [MATLAB中矢量场图的绘制 \(quiver/quiver3/dfield/ppplane\) Plot the vector field with MATLAB\(285\)](#)
4. [深度学习笔记 \(二\) 在TensorFlow上训练一个多层卷积神经网络\(223\)](#)
5. [机器学习 \(三\) 逻辑回归 Logistic Regression\(186\)](#)
6. [浅谈深度学习中的激活函数 - The Activation Function in Deep Learning\(164\)](#)
7. [LeetCode - 412. Fizz Buzz - \( C++ \) - 解题报告 - to\\_string\(105\)](#)
8. [深度学习笔记 \(一\) 卷积神经网络基础 \(Foundation of Convolutional Neural Networks\)\(84\)](#)
9. [机器学习 \(二\) 多变量线性回归 Linear Regression with Multiple Variables\(80\)](#)
10. [UVa 401 - Palindromes 解题报告 - C语言\(66\)](#)
11. [UVa 10082 - WERTYU 解题报告 - C语言\(62\)](#)
12. [LeetCode - 413. Arithmetic Slices - 含中文题意解释 - O\(n\) - \( C++ \) - 解题报告\(60\)](#)
13. [UVa 1225 - Digit Counting - ACM/ICPC Danang 2007 解题报告 - C语言\(57\)](#)
14. [UVa 455 - Periodic Strings - \( C++ \) - 解题报告\(39\)](#)
15. [UVa 1586 - Molar Mass - ACM/ICPC Seoul 2007 - C语言\(37\)](#)
16. [UVa 1583 - Digit Generator 解题报告 - C语言\(36\)](#)
17. [UVa 340 - Master-Mind Hints 解题报告 - C语言\(33\)](#)
18. [UVa 1585 - Score - ACM/ICPC](#)

激活函数的定义及其相关概念

在ICML2016的一篇论文Noisy Activation Functions中，作者将激活函数定义为一个几乎处处可微的  $h : \mathbb{R} \rightarrow \mathbb{R}$ 。



在实际应用中，我们还会涉及到以下的一些概念：

a.饱和

当一个激活函数h(x)满足

$$\lim_{n \rightarrow +\infty} h'(x) = 0$$

时我们称之为**左饱和**。当一个激活函数，既满足左饱和又满足又饱和时，我们称之为**饱和**。

b.硬饱和与软饱和

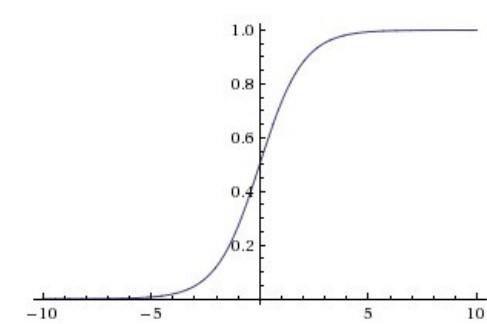
对任意的 $x$ ，如果存在常数 $c$ ，当 $x > c$ 时恒有 $h'(x) = 0$ 则称其为**右硬饱和**，当 $x < c$ 时恒有 $h'(x) = 0$ 则称其为**左硬饱和**。若既满足左硬饱和，又满足右硬饱和，则称这种激活函数为**硬饱和**。但如果只有在极限状态下偏导数等于0的函数，称之为**软饱和**。

Sigmoid函数

Sigmoid函数曾被广泛地应用，但由于其自身的一些缺陷，现在很少被使用了。Sigmoid函数被定义为：

$$f(x) = \frac{1}{1 + e^{-x}}$$

函数对应的图像是：



优点：

- 1.Sigmoid函数的输出映射在(0, 1)之间，单调连续，输出范围有限，优化稳定，可以用作输出层。
- 2.求导容易。

缺点：

- 1.由于其软饱和性，容易产生梯度消失，导致训练出现问题。
- 2.其输出并不是以0为中心的。

Seoul 2005 解题报告 - C语言(32)  
19. UVa -1584 Circular Sequence 解  
题报告 - C语言(27)  
20. LeetCode - 389. Find the  
Difference - 三种不同解法 - ( C++ ) -  
解题报告(27)  
21. LeetCode - 136. Single Number -  
( C++ ) - 解题报告 - 位运算思路  
xor(23)  
22. LeetCode - 231. Power of Two -  
判断一个数是否2的n次幂 - 位运算应  
用实例 - ( C++ )(21)  
23. LeetCode - 167. Two Sum II -  
Input array is sorted - O(n) - ( C++ ) -  
解题报告(20)  
24. LeetCode - 13. Roman to Integer  
- 思考if-else与switch的比较 - ( C++ ) -  
解题报告(20)  
25. LeetCode - 268. Missing Number  
- stable\_sort应用实例 - ( C++ ) - 解题  
报告(19)

评论排行榜

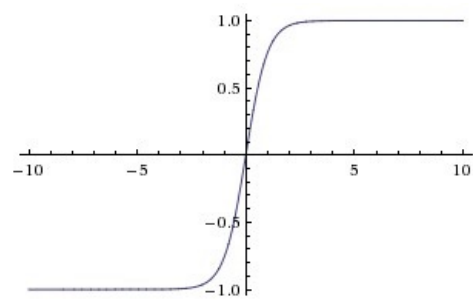
1. 深度学习笔记 (一) 卷积神经网络基  
础 (Foundation of Convolutional  
Neural Networks)(3)  
2. 机器学习 (三) 逻辑回归 Logistic  
Regression(1)

# tanh函数

现在，比起Sigmoid函数我们通常更倾向于tanh函数。tanh函数被定义为

$$\tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}}$$

函数位于[-1, 1]区间上，对应的图像是：



**优点：**

- 1.比Sigmoid函数收敛速度更快。
- 2.相比Sigmoid函数，其输出以0为中心。

**缺点：**

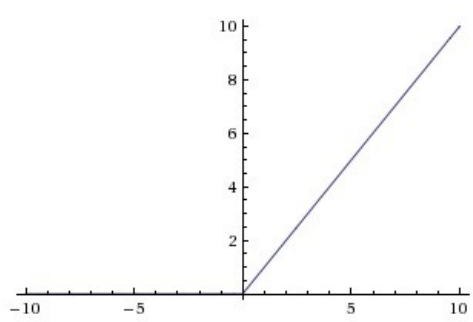
还是没有改变Sigmoid函数的最大问题——由于饱和性产生的梯度消失。

## ReLU

ReLU是最近几年非常受欢迎的激活函数。被定义为

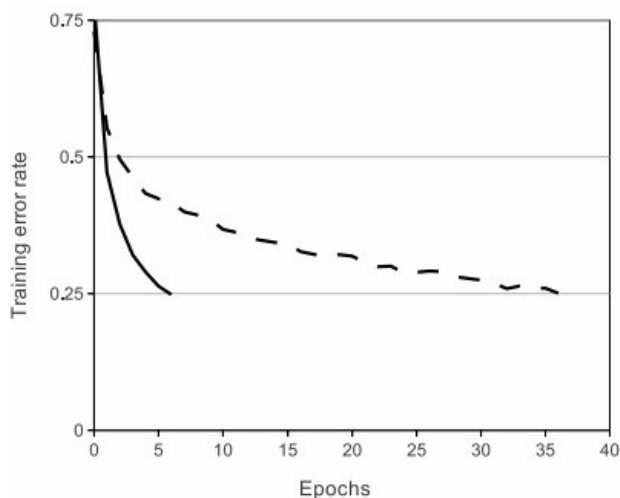
$$y = \begin{cases} 0 & (x \geq 0) \\ x & (x < 0) \end{cases}$$

对应的图像是：



**优点：**

- 1.相比起Sigmoid和tanh，ReLU(e.g. a factor of 6 in Krizhevsky et al.)在SGD中能够快速收敛。例如在下图的实验中，在一个四层的卷积神经网络中，实线代表了ReLU，虚线代表了tanh，ReLU比起tanh更快地到达了错误率0.25处。据称，这是因为它线性、非饱和的形式。



2.Sigmoid和tanh涉及了很多很expensive的操作（比如指数），ReLU可以更加简单的实现。

3.有效缓解了梯度消失的问题。

4.在没有无监督预训练的时候也能有较好的表现。

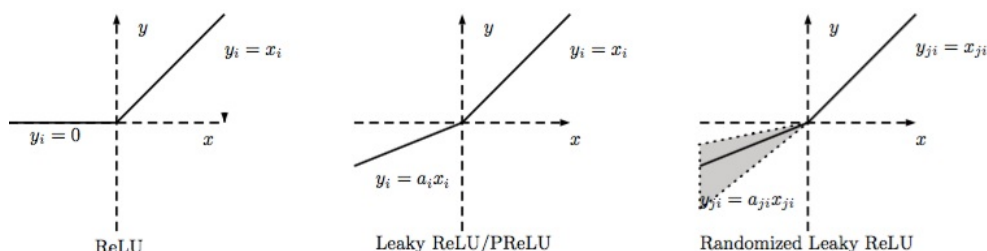
Neuron	MNIST	CIFAR10	NISTP	NORB
<b><i>With unsupervised pre-training</i></b>				
Rectifier	<b>1.20%</b>	<b>49.96%</b>	<b>32.86%</b>	<b>16.46%</b>
Tanh	<b>1.16%</b>	<b>50.79%</b>	35.89%	17.66%
Softplus	<b>1.17%</b>	<b>49.52%</b>	<b>33.27%</b>	19.19%
<b><i>Without unsupervised pre-training</i></b>				
Rectifier	<b>1.43%</b>	<b>50.86%</b>	<b>32.64%</b>	<b>16.40%</b>
Tanh	1.57%	52.62%	36.46%	19.29%
Softplus	1.77%	53.20%	35.48%	17.68%

5.提供了神经网络的稀疏表达能力。

#### 缺点：

随着训练的进行，可能会出现神经元死亡，权重无法更新的情况。如果发生这种情况，那么流经神经元的梯度从这一点开始将永远是0。也就是说，ReLU神经元在训练中不可逆地死亡了。

## LReLU、PReLU与RReLU



通常在LReLU和PReLU中，我们定义一个激活函数为

$$f(y_i) = \begin{cases} y_i & \text{if } (y_i > 0) \\ a_i y_i & \text{if } (y_i \leq 0) \end{cases}$$

#### • LReLU

当 $a_i$ 比较小而且固定的时候，我们称之为LReLU。LReLU最初的目的是为了避免梯度消失。但在一些实验中，我们发现LReLU对准确率并没有太大的影响。很多时候，当我们想要应用LReLU时，我们必须非常小心谨慎地重复训练，选取出合适的 $a$ ，LReLU的表现出的结果才比ReLU好。因此有人提出了一种自适应地从数据中学习参数的PReLU。

- PReLU

PReLU是LReLU的改进，可以自适应地从数据中学习参数。PReLU具有收敛速度快、错误率低的特点。

PReLU可以用于反向传播的训练，可以与其他层同时优化。

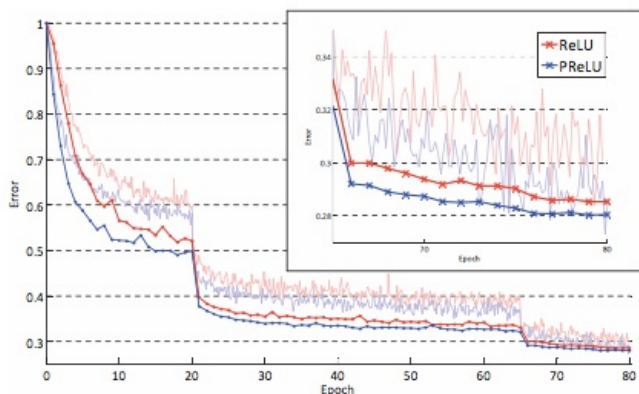


Figure 4. Convergence of ReLU (red) vs. PReLU (blue) of model A on ImageNet. Light lines denote the training error of the current mini-batch, and dark lines denote validation error of the center crops. In the zoom-in is the last few epochs. Learning rates are switched at 20 and 65 epochs.

在论文Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification中，作者就对比了PReLU和ReLU在ImageNet model A的训练效果。

在RReLU中，我们有

$$y_{ji} = \begin{cases} x_{ji} & \text{if } (x_{ji} > 0) \\ a_{ji} y_{ji} & \text{if } (x_{ji} \leq 0) \end{cases}$$

$$a_{ji} \sim U(l, u), l < u \text{ and } l, u \in [0, 1)$$

- RReLU

其中， $a_{ji}$ 是一个保持在给定范围内取样的随机变量，在测试中是固定的。RReLU在一定程度上能起到正则效果。

Activation	Training Error	Test Error
ReLU	0.00318	0.1245
Leaky ReLU, $a = 100$	0.0031	0.1266
Leaky ReLU, $a = 5.5$	0.00362	<b>0.1120</b>
PReLU	0.00178	0.1179
RReLU ( $y_{ji} = x_{ji} / \frac{l+u}{2}$ )	0.00550	<b>0.1119</b>

Table 3. Error rate of CIFAR-10 Network in Network with different activation function

Activation	Training Error	Test Error
ReLU	0.1356	0.429
Leaky ReLU, $a = 100$	0.11552	0.4205
Leaky ReLU, $a = 5.5$	0.08536	<b>0.4042</b>
PReLU	0.0633	0.4163
RReLU ( $y_{ji} = x_{ji} / \frac{l+u}{2}$ )	0.1141	<b>0.4025</b>

Table 4. Error rate of CIFAR-100 Network in Network with different activation function

Activation	Train Log-Loss	Val Log-Loss
ReLU	0.8092	0.7727
Leaky ReLU, $a = 100$	0.7846	0.7601
Leaky ReLU, $a = 5.5$	0.7831	0.7391
PReLU	0.7187	0.7454
RReLU ( $y_{ji} = x_{ji} / \frac{l+u}{2}$ )	0.8090	<b>0.7292</b>

Table 5. Multi-classes Log-Loss of NDSB Network with different activation function

在论文Empirical Evaluation of Rectified Activations in Convolution Network中，作者对比了RReLU、LReLU、PReLU、ReLU 在CIFAR-10、CIFAR-100、NDSB网络中的效果。

## 总结

关于激活函数的选取，目前还不存在统一的定论，实践过程中更多还是需要结合实际情况，考虑不同激活函数的优缺点综合使用。同时，也期待越来越多的新想法，改进目前存在的不足。



文章部分图片内容参考自：

[CS231n Convolutional Neural Networks for Visual Recognition](#)

[Quora - What is the role of the activation function in a neural network?](#)

[深度学习中的激活函数导引](#)

[Noisy Activation Functions—ICML2016](#)

本文为作者的个人学习笔记，转载请先声明。如有疏漏，欢迎指出，不胜感谢。

分类: [Deep Learning](#),[Machine Learning](#)

好文要顶

关注我

收藏该文



rgvb178

关注 - 3

粉丝 - 7

0

推荐

0

反对

+加关注

« 上一篇: [深度学习笔记 \(二\) 在TensorFlow上训练一个多层卷积神经网络](#)

posted @ 2016-11-11 23:07 [rgvb178](#) 阅读(164) 评论(0) [编辑](#) [收藏](#)

[刷新评论](#) [刷新页面](#) [返回顶部](#)



注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库

【推荐】用1%的研发投入，搭载3倍性能的网易视频云技术

【推荐】融云发布 App 社交化白皮书 IM 提升活跃超 8 倍



**你提项目需求  
Demo原型我来做**

帮你突破开发困境

[免费验证](#)

### 最新IT新闻:

- 京东发布双11战报：12时交易额同比增长78%
  - Facebook利用毫米波技术实现20 Gbps传输率
  - 中国液态金属逆天：逆重力攀爬
  - 用照片讲故事 Instagram用户可在照片上添加备注和链接
  - 自动驾驶工程师都去哪儿了？通用、谷歌抢得最欢
- » [更多新闻...](#)

极光 智能推送全面升级 更快、更稳定、更成熟

[了解更多](#)

最新知识库文章:

- [循序渐进地代码重构](#)
- [技术的正宗与野路子](#)
- [陈皓：什么是工程师文化？](#)
- [没那么难，谈CSS的设计模式](#)
- [程序猿媳妇儿注意事项](#)
- » [更多知识库文章...](#)

Copyright ©2016 rgvb178