


常用激活函数比较

 不会停的蜗牛 (/u/7b67af2e61b3) [+ 关注](#)

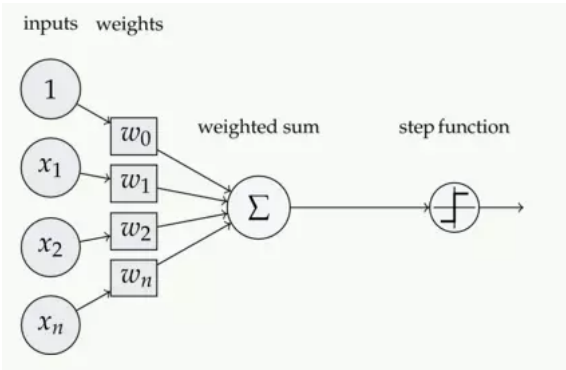
2017.03.14 12:46* 字数 1589 阅读 20138 评论 9 喜欢 51 (/u/7b67af2e61b3)

本文结构:

- 1. 什么是激活函数
- 2. 为什么要用
- 3. 都有什么
- 4. sigmoid , ReLU, softmax 的比较
- 5. 如何选择

1. 什么是激活函数

如下图，在神经元中，输入的 inputs 通过加权，求和后，还被作用了一个函数，这个函数就是激活函数 Activation Function。



2. 为什么要用

如果不用激励函数，每一层输出都是上层输入的线性函数，无论神经网络有多少层，输出都是输入的线性组合。

如果使用的的话，激活函数给神经元引入了非线性因素，使得神经网络可以任意逼近任何非线性函数，这样神经网络就可以应用到众多的非线性模型中。

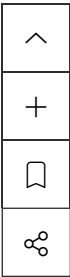
3. 都有什么

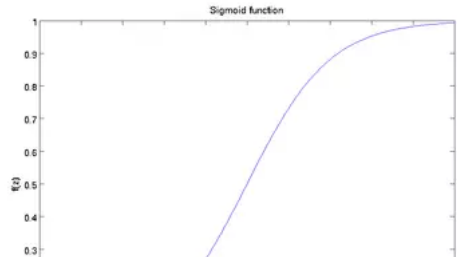
(1) sigmoid函数

公式:

$$f(z) = \frac{1}{1 + \exp(-z)}.$$

曲线:





也叫 Logistic 函数，用于隐层神经元输出
取值范围为(0,1)
它可以将一个实数映射到(0,1)的区间，可以用来做二分类。
在特征相差比较复杂或是相差不是特别大时效果比较好。

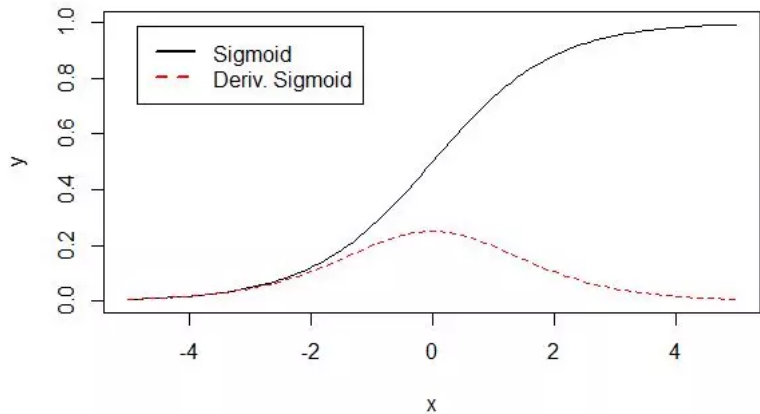
sigmoid缺点：
激活函数计算量大，反向传播求误差梯度时，求导涉及除法
反向传播时，很容易就会出现梯度消失的情况，从而无法完成深层网络的训练

下面解释为何会出现梯度消失：

反向传播算法中，要对激活函数求导，sigmoid 的导数表达式为：

$$\phi'(x) = \phi(x)(1 - \phi(x))$$

sigmoid 原函数及导数图形如下：



由图可知，导数从 0 开始很快就又趋近于 0 了，易造成“梯度消失”现象

(2) Tanh函数

公式

$$f(z) = \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}},$$

$$\tanh(x) = 2sigmoid(2x) - 1$$

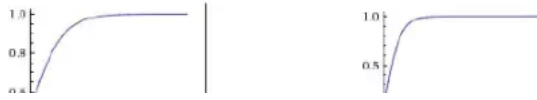
曲线

^

+

🔖

🔗



也称为双切正切函数

取值范围为 $[-1, 1]$ 。

tanh在特征相差明显时的效果会很好，在循环过程中会不断扩大特征效果。

与 sigmoid 的区别是，tanh 是 0 均值的，因此实际应用中 tanh 会比 sigmoid 更好

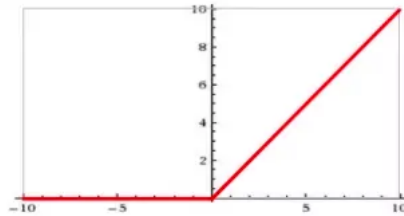
(3) ReLU

Rectified Linear Unit(ReLU) - 用于隐层神经元输出

公式

$$\phi(x) = \max(0, x)$$

曲线



输入信号 < 0 时，输出都是0， > 0 的情况下，输出等于输入

ReLU 的优点：

Krizhevsky et al. (<https://link.jianshu.com?>

[t=http://www.cs.toronto.edu/~fritz/absps/imagenet.pdf](http://www.cs.toronto.edu/~fritz/absps/imagenet.pdf)) 发现使用 ReLU 得到的 SGD 的收敛速度会比 sigmoid/tanh 快很多

ReLU 的缺点：

训练的时候很“脆弱”，很容易就“die”了

例如，一个非常大的梯度流过一个 ReLU 神经元，更新过参数之后，这个神经元再也不会对任何数据有激活现象了，那么这个神经元的梯度就永远都会是 0。

如果 learning rate 很大，那么很有可能网络中的 40% 的神经元都“dead”了。

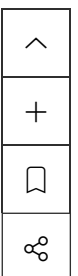
(4) softmax函数

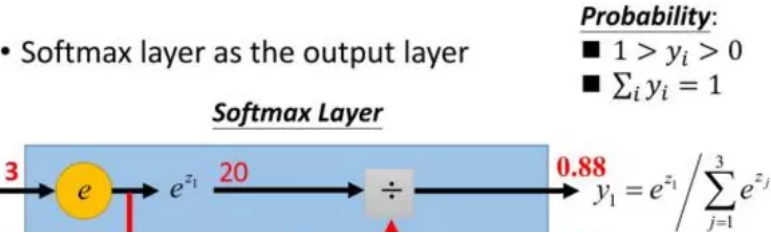
Softmax - 用于多分类神经网络输出

公式

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}}$$

举个例子来看公式的意思：





就是如果某一个 z_j 大过其他 z , 那这个映射的分量就逼近于 1,其他就逼近于 0, 主要应用就是多分类。

为什么要取指数, 第一个原因是要模拟 \max 的行为, 所以要让大的更大。
第二个原因是需要一个可导的函数。

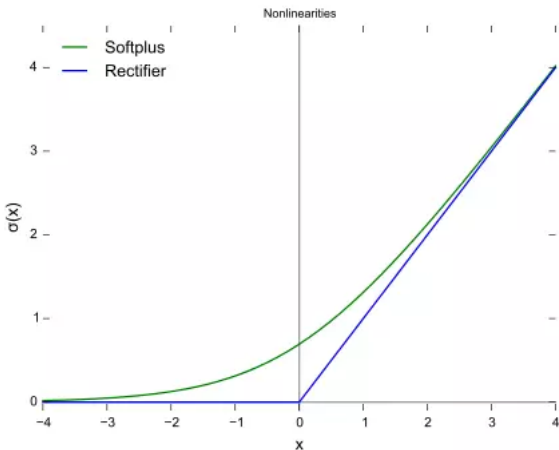
4. sigmoid , ReLU , softmax 的比较

Sigmoid 和 ReLU 比较:

sigmoid 的梯度消失问题, ReLU 的导数就不存在这样的问题, 它的导数表达式如下:

$$\phi'(x) = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases}$$

曲线如图



对比sigmoid类函数主要变化是:

- 1) 单侧抑制
- 2) 相对宽阔的兴奋边界
- 3) 稀疏激活性。

Sigmoid 和 Softmax 区别:

softmax is a generalization of logistic function that “squashes”(maps) a K-dimensional vector z of arbitrary real values to a K-dimensional vector $\sigma(z)$ of real values in the range $(0, 1)$ that add up to 1.

sigmoid将一个real value映射到 $(0,1)$ 的区间, 用来做二分类。

而 softmax 把一个 k 维的real value向量 $(a_1,a_2,a_3,a_4....)$ 映射成一个 $(b_1,b_2,b_3,b_4....)$ 其中 b_i 是一个 $0 \sim 1$ 的常数, 输出神经元之和为 1.0, 所以相当于概率值, 然后可以根据 b_i 的概率大小来进行多分类的任务。

二分类问题时 sigmoid 和 softmax 是一样的, 求的都是 cross entropy loss, 而 softmax 可以用于多分类问题

^

+

🔖

🔗

softmax是sigmoid的扩展，因为，当类别数 $k = 2$ 时，softmax 回归退化为 logistic 回归。具体地说，当 $k = 2$ 时，softmax 回归的假设函数为：

$$h_{\theta}(x) = \frac{1}{e^{\theta_1^T x} + e^{\theta_2^T x}} \begin{bmatrix} e^{\theta_1^T x} \\ e^{\theta_2^T x} \end{bmatrix}$$

利用softmax回归参数冗余的特点，从两个参数向量中都减去向量 θ_1 ，得到：

$$\begin{aligned} h(x) &= \frac{1}{e^{\tilde{\theta}^T x} + e^{(\theta_2 - \theta_1)^T x}} \begin{bmatrix} e^{\tilde{\theta}^T x} \\ e^{(\theta_2 - \theta_1)^T x} \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{1 + e^{(\theta_2 - \theta_1)^T x}} \\ \frac{e^{(\theta_2 - \theta_1)^T x}}{1 + e^{(\theta_2 - \theta_1)^T x}} \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{1 + e^{(\theta_2 - \theta_1)^T x}} \\ 1 - \frac{1}{1 + e^{(\theta_2 - \theta_1)^T x}} \end{bmatrix} \end{aligned}$$

最后，用 θ' 来表示 $\theta_2 - \theta_1$ ，上述公式可以表示为 softmax 回归器预测其中一个类别的概率为

$$\frac{1}{1 + e^{(\theta')^T x}}$$

另一个类别概率的为

$$1 - \frac{1}{1 + e^{(\theta')^T x}}$$

这与 logistic回归是一致的。

softmax建模使用的分布是多项式分布，而logistic则基于伯努利分布

多个logistic回归通过叠加也同样可以实现多分类的效果，但是 softmax回归进行的多分类，类与类之间是互斥的，即一个输入只能被归为一类；多个logistic回归进行多分类，输出的类别并不是互斥的，即"苹果"这个词语既属于"水果"类也属于"3C"类别。

5. 如何选择

选择的时候，就是根据各个函数的优缺点来配置，例如：

如果使用 ReLU，要小心设置 learning rate，注意不要让网络出现很多“dead”神经元，如果不好解决，可以试试 Leaky ReLU、PReLU 或者 Maxout。

参考资料：

<http://feisky.xyz/machine-learning/neural-networks/active.html> (<https://link.jianshu.com?t=http://feisky.xyz/machine-learning/neural-networks/active.html>)

http://jishu.y5y.com.cn/qq_17754181/article/details/56495406 (https://link.jianshu.com?t=http://jishu.y5y.com.cn/qq_17754181/article/details/56495406)

<https://www.zhihu.com/question/29021768> (<https://link.jianshu.com?t=https://www.zhihu.com/question/29021768>)

http://blog.csdn.net/cyh_24/article/details/50593400 (https://link.jianshu.com?t=http://blog.csdn.net/cyh_24/article/details/50593400)

<http://www.itdadao.com/articles/c15a1000722p0.html> (<https://link.jianshu.com?t=http://www.itdadao.com/articles/c15a1000722p0.html>)

<https://www.zhihu.com/question/23765351> (<https://link.jianshu.com?t=https://www.zhihu.com/question/23765351>)

