



贊同24



分享

速用筆記| Sigmoid/Tanh/ReLu/Softplus 激活函數的圖形、表達式、導數、適用條件



橘Oran

致力於用經濟學×數據科學解決問題

24 人贊同了該文章

目錄

收起

一、Sigmoid 激活函數

公式：★Sigmoid 激活函數...

值域：★由於值域剛好落在0...

導數：★致命缺點，在函數...

二、Tanh 激活函數

公式：★除了下面這個經典...

值域：★具有數據居中優勢...

導數：★可惜還是存在和Sig...

三、ReLU 激活函數

公式：★ReLU 是當前最常用...

值域：★簡單粗暴卻好用的R...

導數：★要么為0，要么為1

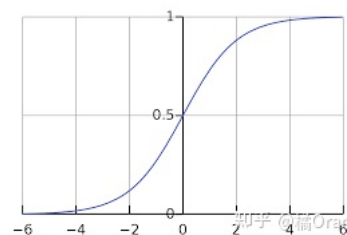
四、Leaky ReLU 激活函數

公式：★對於小於0的，前...

非線性激活函數是神經網絡捕捉數據間非線性關係的最基本前提，如果取消所有的激活函數，或者都換成線性函數，那麼整個神經網絡都將最終退化為一個由輸入到輸出的線性組合。激活函數有很多，論文中使用的更是令人眼花繚亂。網上能找到不少關於激活函數優缺點的總結梳理，但是如果沒有函數公式、圖形、導數這些基本信息，我們其實是很難理解和記住各自的優缺點的。

於是今天在這裡整理了常見的激活函數基本屬性，從屬性出發，去直觀理解各自的優缺點，從而在需要的時候做出最適合場景的選擇。

一、Sigmoid 激活函數



sigmoid 函數圖形

公式：★Si

▲ 贊同24 ▼

● 2 條評論

🔗 分享

❤ 喜歡

★ 收藏

📄 申請轉載

...

登錄即可查看 超5億 專業優質內容

超5 千萬創作者的優質提問、專業回答、深度文章和精彩視頻盡在知乎。

立即登錄/註冊



四、ELU 激活函數

值域：★由於值域剛好落在0~1之間，恰好是概率的範圍，因此最常用在二分類模型的輸出層

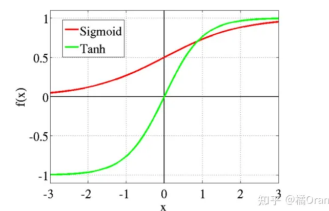
$(0, 1)$

導數：★致命缺點，在函數兩側當 x 很小或很大時，導數接近0（圖形呈水平），易出現梯度消失

$$f'(x) = f(x)(1 - f(x))$$

多說一句：導數大小可以通過上面的函數圖形看，如圖，當 x 落在兩端的時候，函數斜率幾乎為0（被稱作**函數飽和**），也就是導數（梯度）趨近於0，這使得反向傳播時很容易出現**梯度消失**。因此Ng 吳恩達老師建議，除了作為二分類中的輸出層，**不要用Sigmoid** 激活函數，目前工業界已幾乎沒人在隱層中用它了。

二、Tanh 激活函數



tanh 函數圖形（綠色）對比Sigmoid 函數圖形（紅色）

公式：★除了下面這個經典表達，tanh 還很多種寫法，有時一眼還不認識了，其實都等價的

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

值域：★具有數據居中優勢，隱藏層出來的激活值平均數接近0，且導數不為 0

$(-1, 1)$

導數：★可惜還是存在和Sigmoid 一樣的缺陷，函數兩端導數趨近於0，也容易出現梯度消失

$$f'(x) = 1 - f(x)^2$$

登錄即可查看 超5億 專業優質內容

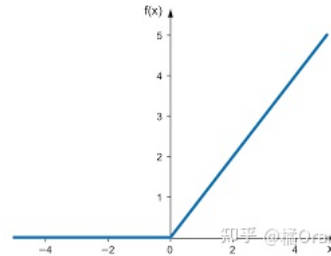
超5 千萬創作者的優質提問、專業回答、深度文章和精彩視頻盡在知乎。



嚴格優於sigmoid。不過指數計算還是比較耗時的。

為了緩解上述的梯度消失問題，並提高學習速度，2012年學術界提出了下面的ReLU 激活函數。

三、ReLU 激活函數



ReLU 函數圖形

公式：★ReLU 是當前最常用的激活函數，公式極為簡單，一般會縮寫成 $\max(0, x)$

$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$$

值域：★簡單粗暴卻好用的ReLU，小於0 的輸入就輸出0，大於0 的直接原樣輸出

$[0, \infty)$

導數：★要么為0，要么為1

$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$$

多說一句：在實踐中如果不知道如何選激活函數，一般默認選ReLU。因為相比於Sigmoid 和Tanh 等激活函數，ReLU 有兩個突出的優勢：

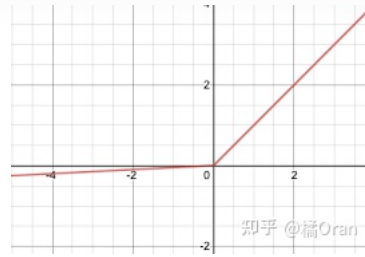
- 一是緩解了梯度消失的問題（ x 大於0的右側），只要 x 大於0，導數就為1；
- 二是計算效率高，模型學起來快得多。一個原因是避免了大量的指數運算。另一個原因是函數導數有約一半的概率為0，減少了很多非零計算。並且在訓練效果上，由於模型很大，即便很多隱藏單元的導數為0，網絡中也有足夠多的單位使 x 大於0。因此，對於大多數訓練例子來說，學習仍然是快且有效的。

不過ReLU 對於小於0 的 x 進行“一刀切”映射，讓它們前向輸出和反向梯度都是0，相當於這些神經元都沒用上（Dead ReLU問題）。的確有些粗暴了。因此後來出現了很多ReLU 的變種。下文列舉了其中三

登錄即可查看 超5億 專業優質內容

超5 千萬創作者的優質提問、專業回答、深度文章和精彩視頻盡在知乎。





Leaky ReLU 函數圖形

公式：★對於小於0 的 x ，前面乘了一個很小的係數 α ，例如取0.01

$$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$$

值域：★包含了負區間，解決Dead ReLU 問題

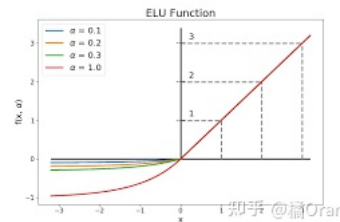
$(-\infty, +\infty)$

導數：★這樣對於小於0 的 α ，也有了一個很小的導數

$$f(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$$

多說一句：Leaky ReLU 相比於ReLU，針對小於0 的 x 做了更細緻的映射，使得它們的前向傳播值和反向傳播的梯度都不再為0，因而可以**利用到更多信息**。並且它的函數**雙側都不飽和**，對於所有的輸入，不管是**大於等於0**還是**小於0**，**都不會出現梯度消失的情況**。但它也有潛在的問題，參數 α 的選取需要經驗或調參。

四、ELU 激活函數

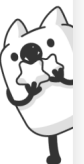


ELU 函數圖形（當參數取1時，函數趨近-1）

公式：★重

登錄即可查看 **超5億** 專業優質內容

超5 千萬創作者的優質提問、專業回答、
深度文章和精彩視頻盡在知乎。



值域：★解決了Dead ReLU 問題，且包含負區間，有利於數據居中，加速學習

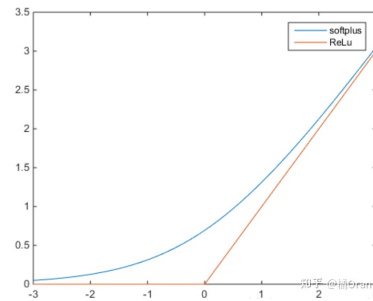
$$(-\alpha, +\infty)$$

導數：★對於小於0 的 α ，也有一個很小的導數，但左側存在軟飽和性

$$f(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$$

多說一句：ELU 相比於ReLU 的優勢：首先，左側數據不再為0，解決了Dead ReLU，問題；第二，包含負數區間，一定程度上讓數據居中，輸出均值接近於零，所以收斂速度更快。另外有研究認為，左側軟飽和能夠讓ELU對輸入變化或噪聲更魯棒。不過指數計算也讓計算量增大了很多。

五、SoftPlus 激活函數



SoftPlus 函數圖形（藍色）對比ReLU 函數（橙色）

公式：★也可以看作是ReLU 的平滑

$$f(x) = \log_e(1 + e^x)$$

值域：★不包含0，解決了Dead ReLU 問題，但不包含負區間，不能加速學習

$$(0, +\infty)$$

導數：★有沒有很眼熟？哈哈剛好是Sigmoid 函數公式！

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

多說一句：SoftPlus 相當於是對ReLU 的平滑，解決了Dead ReLU 問題。至於為什麼要整理 softplus 這個不管特別當目的激活函數呢？原因且這而在于恰好看了一篇論文（DeepAR）裡面用了這個函數，咋

登錄即可查看 超5億 專業優質內容

超5 千萬創作者的優質提問、專業回答、深度文章和精彩視頻盡在知乎。



以上就是對常用激活函數的整理，下面這張表是在網上冲浪時淘來的[2]，賞心悅目，有需要就保存吧：

Name	Plot	Equation	Derivative
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1 + e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
Tanh		$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2 + 1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parameteric Rectified Linear Unit (PReLU) [2]		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Exponential Linear Unit (ELU) [3]		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1 + e^x)$	$f'(x) = \frac{1}{1 + e^{-x}}$

A Good Take away, please download :)

參考資料

[1]神經網絡梯度消失與梯度爆炸 [神經網絡梯度消失與梯度爆炸_Mlss-Y的博客-CSDN博客_神經網絡梯度消失](#)

[2]Activation Functions in Neural Networks [towardsdatascience.com/ ...](#)

編輯於2022-12-02 00:06 · IP 屬地未知

深度學習 (Deep Learning) relu

寫下你的評論...

2 條評論

默认排序

最新

登錄即可查看 超5億 專業優質內容

超5 千萬創作者的優質提問、專業回答、深度文章和精彩視頻盡在知乎。

