 版权声明：转载请注明出处 <https://blog.csdn.net/JNingWei/article/details/79210904>

Introduction

激活函数 (activation function) 层又称 非线性映射 (non-linearity mapping) 层，作用是 增加整个网络的非线性 (即 表达能力 或 抽象能力) 。

深度学习之所以拥有 强大的表示能力，法门便在于 激活函数的 非线性。

然而物极必反。由于 非线性设计 所带来的一系列 副作用 (如 期望均值不为0、死区)，迫使炼丹师们设计出种类繁多的激活函数来 约束 非线性的 合理范围。

由于激活函数接在bn之后，所以激活函数的输入被限制在了 (-1, 1) 之间。因此，即使是relu这种简易的激活函数，也能很好地发挥作用。

激活函数类型

激活函数中，常用的有 Sigmoid、tanh(x)、Relu、Relu6、Leaky Relu、参数化Relu、随机化Relu、ELU。

其中，最经典的莫过于 **Sigmoid函数** 和 **Relu函数**。

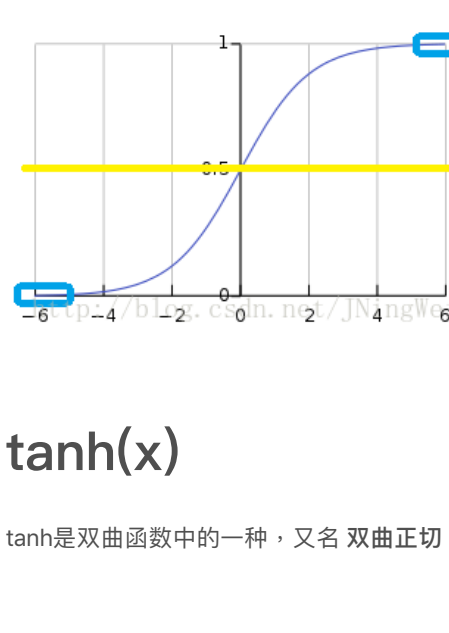
点赞3

评论

收藏

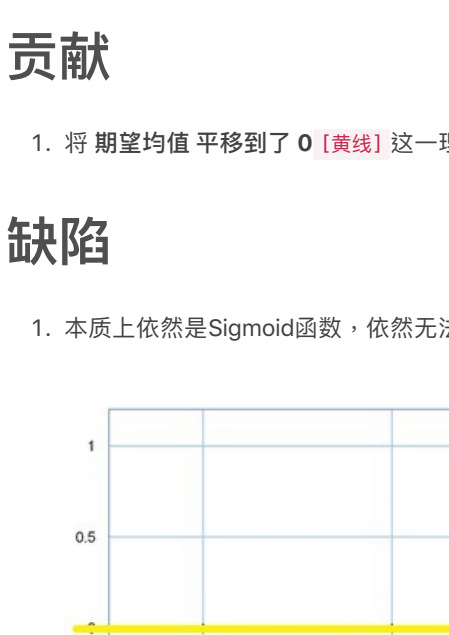
被用作神经网络的阈值函数，将变量映射到 (0，1) 之间：

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



缺陷：

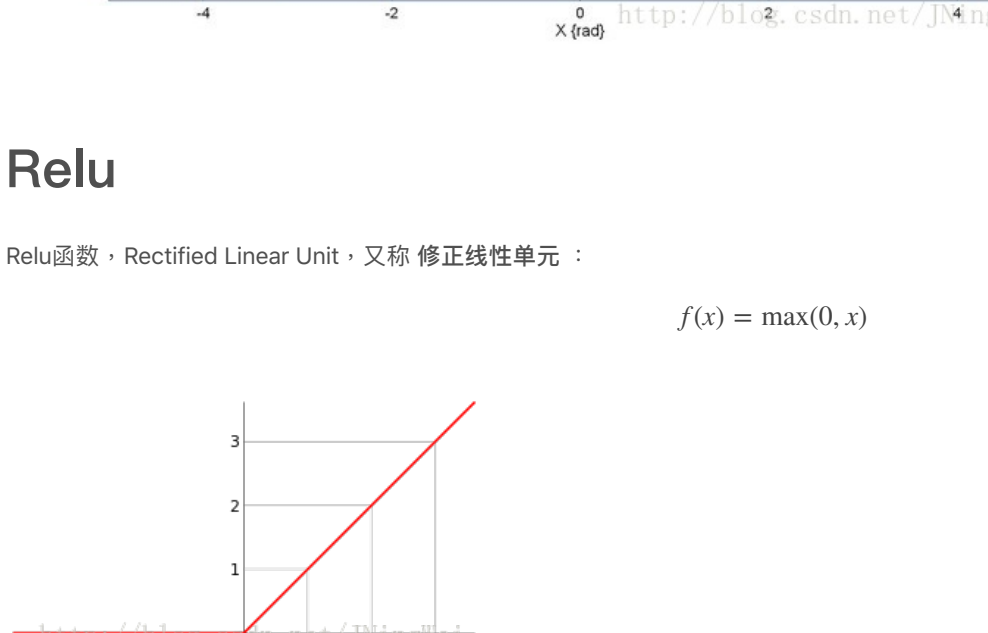
- 输出值落在 (0, 1) 之间，期望均值为 0.5 **[黄线]**，不符合 均值为 0 的理想状态。
- 发现有的梯度下降算法所限 (严重依赖逐层的梯度计算值)，Sigmoid函数对落入 $(-\infty, -5) \cup (5, +\infty)$ 的输入值，梯度 计算为 0，发生 梯度弥散。因此该函数存在一正一负 两块“死区” **[蓝框区域]**：



tanh(x)

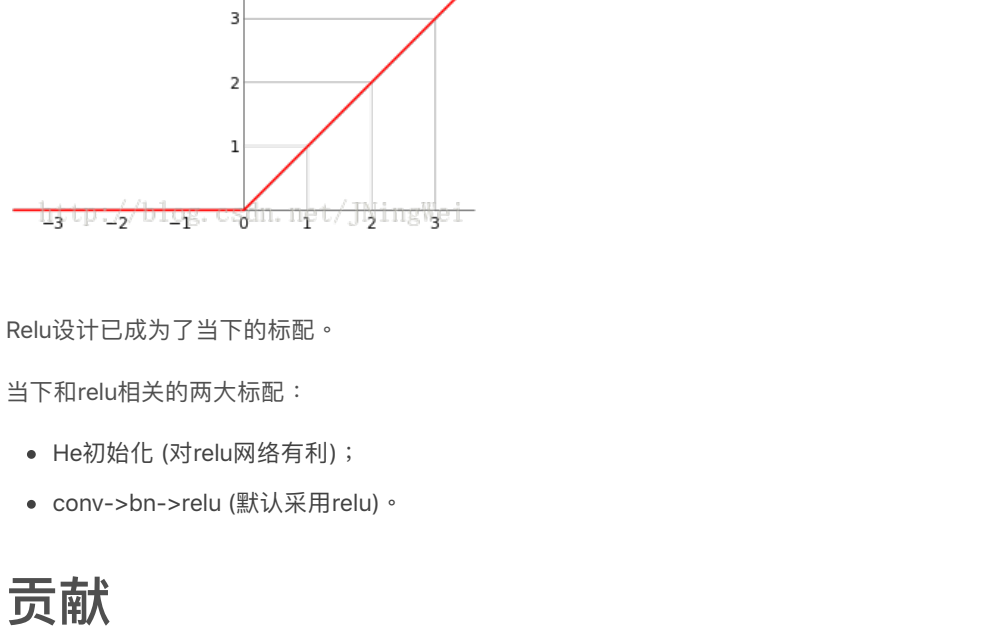
tanh是双曲函数中的一种，又名 双曲正切：

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



贡献

- 将 期望均值 平移到了 0 **[黄线]** 这一理想状态。



缺陷

- 本质上依然是 Sigmoid函数，依然无法回避一左一右两块 “死区” **[蓝框区域]** (此时“死区”甚至还扩张了区间范围)：



Relu

Relu函数，Rectified Linear Unit，又称 修正线性单元：

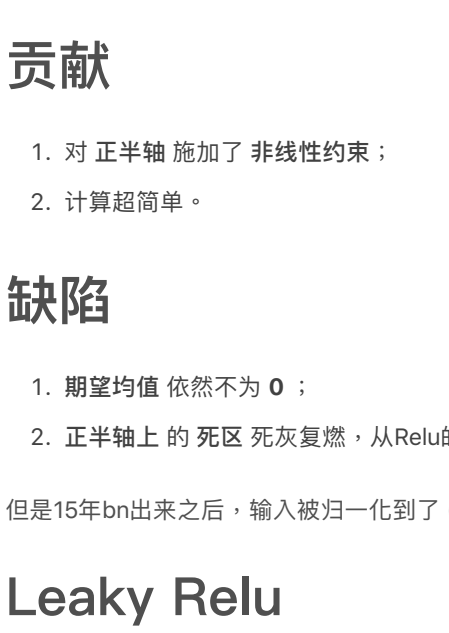
$$f(x) = \max(0, x)$$



Relu设计已成为了当下的标配。

当下和relu相关的两大标配：

- He初始化 (对relu网络有利)；
- conv->bn->relu (默认采用relu)。



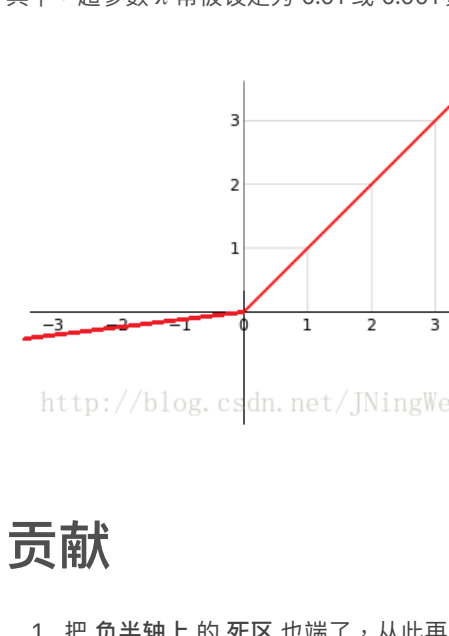
贡献

- 彻底 消灭了 正半轴上的 死区；
- 计算超简单；
- 正是因为AlexNet中提出了Relu，在当时很好地缓解了梯度弥散，使得网络深度的天花板第一次被打破；
- 该设计有助于使模型参数稀疏。



缺陷

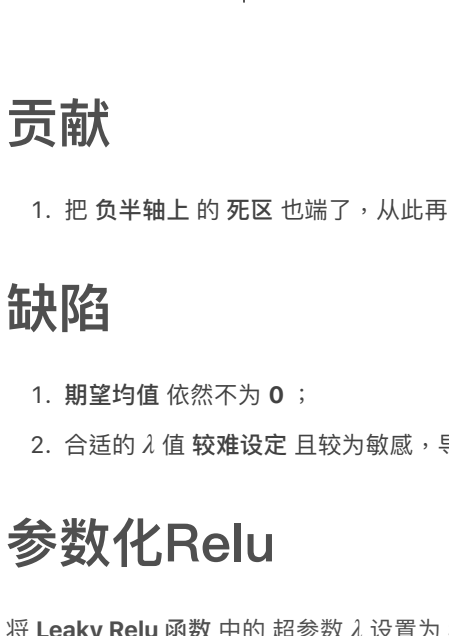
- 期望均值 跑得离 0 更远了；
- 负半轴上的 死区 **[蓝框区域]** 直接蚕食到了 0 点。



Relu6

由于 **Relu函数** 的 正半轴 不能加任何非线性约束，因此当输入为 正大数时，易引起 正半轴上的 梯度爆炸。因此，Relu6 应运而生：

$$f(x) = \min(\max(0, x), 6)$$




贡献

- 对 正半轴 施加了 非线性约束；
- 计算超简单。



缺陷

- 期望均值 依然不为 0；
- 正半轴上的 死区 死灰复燃，从Relu的 $(-\infty, 0)$ 蚕食至 Relu6的 $(-\infty, 0) \cup (6, +\infty)$ 。

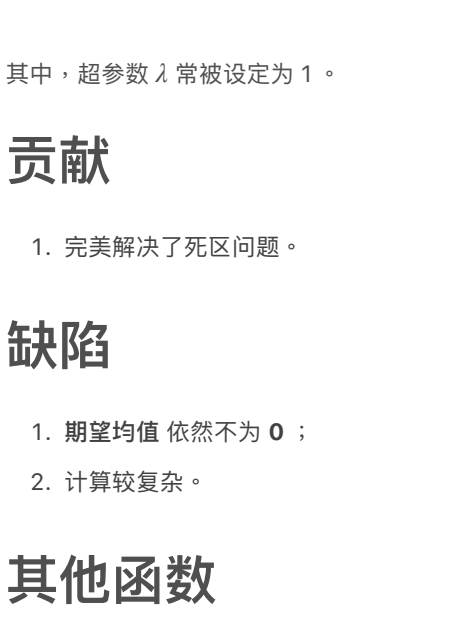


但是15年bn出来之后，输入被归一化到了 (-1, 1) 之间。因此，relu6的设计就显得没有必要了。

Leaky Relu

对 **Relu函数** 新增一 超参数 λ ，以解决负半轴的死区问题：

$$f(x) = \begin{cases} x & \text{当} x \geq 0 \text{时;} \\ \lambda x & \text{当} x < 0 \text{时.} \end{cases}$$



其中，超参数 λ 常被设定为 0.01 或 0.001 数量级的 较小正数。

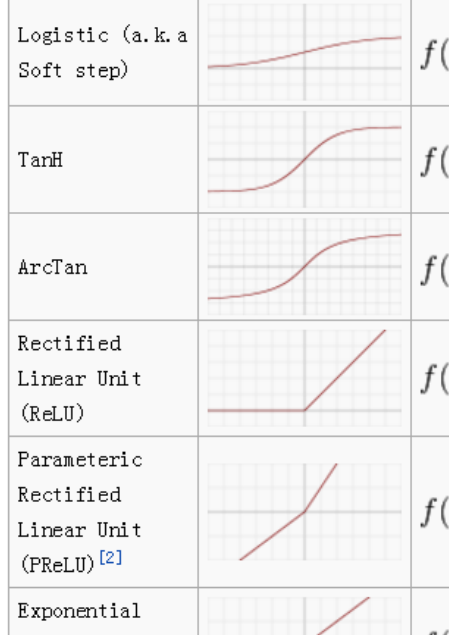
贡献

- 把 负半轴上的 死区 也端了，从此再无死区；



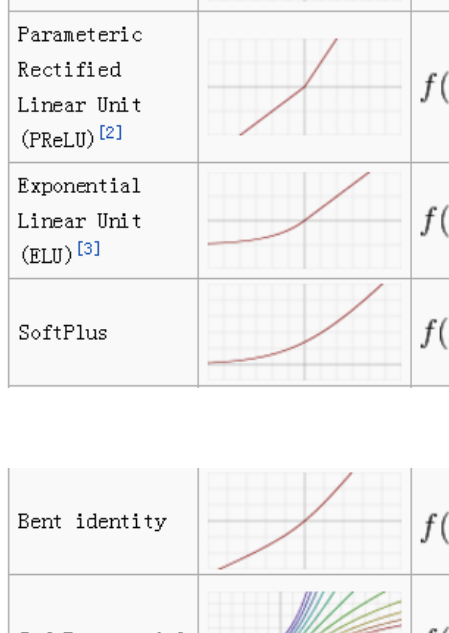
缺陷

- 期望均值 依然不为 0；
- 合适的 λ 值 较难设定 且较为敏感，导致在实际使用中 性能不稳定。



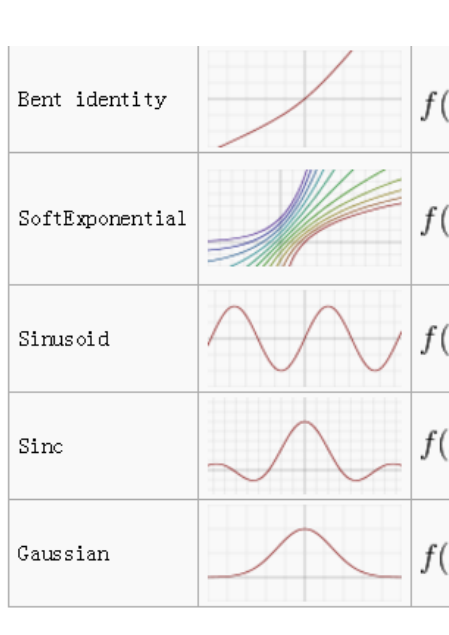
参数化Relu

将 **Leaky Relu函数** 中的 超参数 λ 设置为 和模型一起 被训练到的 变量，以解决 λ 值 较难设定 的问题。




贡献

- 更大自由度。



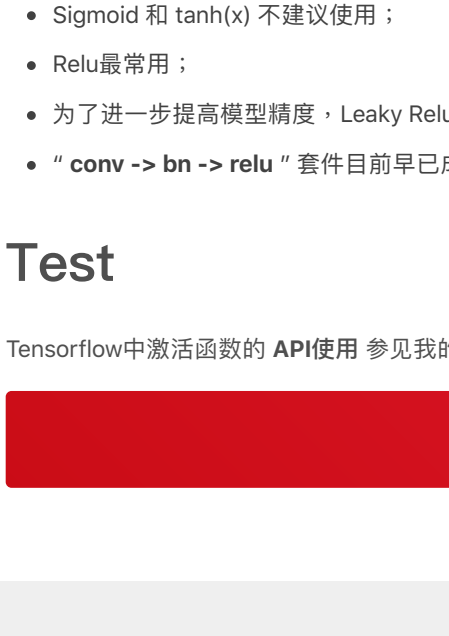
缺陷

- 更大的过拟合风险；
- 较为麻烦。



随机化Relu

将 **Relu Relu函数** 中的 超参数 λ 随机设置。



ELU

ELU函数，Exponential Linear Unit，又称 指数化线性单元，于2016年提出。

$$f(x) = \begin{cases} x & \text{当} x \geq 0 \text{时;} \\ \lambda(\exp(x) - 1) & \text{当} x < 0 \text{时.} \end{cases}$$



其中，超参数 λ 常被设定为 1。

贡献

- 完美解决了死区问题。



缺陷

- 期望均值 依然不为 0；
- 计算较复杂。



其他函数

下图摘自：【机器学习】神经网络-激活函数-面面观(Activation Function)

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


Bent identity

$$f(x) = \frac{\sqrt{x^2 + 1} - 1}{2} + x$$
$$f'(x) = \frac{x}{2\sqrt{x^2 + 1}} + 1$$



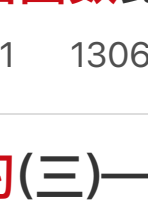
SoftExponential

$$f(\alpha, x) = \begin{cases} -\frac{\log_e(1 - \alpha(x + \alpha))}{\alpha} & \text{for } \alpha < 0 \\ x & \text{for } \alpha = 0 \\ \frac{e^{\alpha x} - 1}{\alpha} + \alpha & \text{for } \alpha > 0 \end{cases}$$
$$f'(\alpha, x) = \begin{cases} \frac{1}{1 - \alpha(x + \alpha)} & \text{for } \alpha < 0 \\ e^{\alpha x} & \text{for } \alpha \geq 0 \end{cases}$$




SineSoid

$$f(x) = \sin(x)$$
$$f'(x) = \cos(x)$$



Sinc

$$f(x) = \begin{cases} 1 & \text{for } x = 0 \\ \frac{\sin(x)}{x} & \text{for } x \neq 0 \end{cases}$$
$$f'(x) = \begin{cases} \cos(x) & \text{for } x = 0 \\ \frac{\sin(x)}{x^2} & \text{for } x \neq 0 \end{cases}$$



Gaussian

$$f(x) = e^{-x^2}$$
$$f'(x) = -2xe^{-x^2}$$



Summary

- Every coin has two sides.
- Sigmoid 和 tanh(x) 不建议使用；
- Relu最常用；
- 为了进一步提高模型精度，Leaky Relu、参数化Relu、随机化Relu 和 ELU 均可尝试 (但四者之间无绝对的高下之分)。
- "conv->bn->relu" 套件目前早已成为了CNN标配module。

Test

Tensorflow中激活函数的API使用 参见我的另一篇文章：[tensorflow: 激活函数\(Activation Functions\) 探究](#)

APP打开，阅读更顺畅

当前没有评论 [点击发表评论](#)

深度学习使用到的激活函数种类和优缺点解释！

yangdashi888 5610次阅读 09-18 APP打开

深度学习笔记六：常见激活函数总结

xierhacker 5828次阅读 05-10 APP打开

[深度学习] 激活函数(Activation function) – 小黑_BUP..._CSDN博客

2018-10-21

...学习中的激活函数 – The Activation Function in De..._CSDN博客

2018-10-12

孩子注意力不集中，作业写得慢怎么办？台湾家长都在看！
上课走神怎么办

常用激活函数（激励函数）理解与总结

tyhj_sf 39682次阅读 05-13 APP打开

...学习笔记-06]激活函数(Activation Function) – caic..._CSDN博客

2019-5-14

...学习中的激活函数 – The Activation Function in De..._CSDN博客

2018-8-16

聊一聊深度学习的activation function

omnispace 2230次阅读 10-07 APP打开

常用激活函数比较

u011584941 13065次阅读 05-10 APP打开

深度学习(三)——激活函数(active function)的作用和类..._CSDN博客

2018-11-12

[深度学习] 激活函数 – 四月晴 – CSDN博客

2018-11-14

45K! 刚 面完AI 岗这几分分享给你！

58937次阅读 05-09

激活函数

Ddreaming 8725次阅读 11-18 APP打开

机器学习(一):激活函数(Activation Function) – tiny_g..._CSDN博客

2019-5-7

CNN入门讲解:什么是激活函数(Activation Function) – b..._CSDN博客

2019-5-15

DL学习—AF: 理解机器学习中常用的激活函数(sigmoid、softmax等)简介、应用、计算图实现、代码实现详细攻略

qq_41185868 3364次阅读 08-30 APP打开

tensorflow: 激活函数(Activation_Functions) 探究 – P..._CSDN博客

2019-1-24

深度学习:神经网络中的激活函数 – 皮皮blog – CSDN博客

2019-5-10

《Noisy Activation Function》噪声激活函数（一）

YhL_Leo 6989次阅读 06-22 APP打开

[深度学习] 神经网络中的激活函数（Activation function）

weixin_41028208 1830次阅读 09-30 APP打开

神经网络的激活函数(Activation Function)

chenxaioxue 603次阅读 09-02 APP打开

本地100万红包限时领 快来抢

从零开始编写深度学习库（三）ActivationLayer网络层CPU实现

hijmce 1611次阅读 05-11 APP打开

深度学习中激活函数

u013989576 7008次阅读 04-15 APP打开

【深度学习】深度学习中常用的激活函数

shwan_ma 1300次阅读 07-28 APP打开

激活函数总结.md

zuolunqiang 2332次阅读 12-22 APP打开

汽车美容五大盈利系统

机器学习笔记-神经网络中激活函数（activation function）对比--Sigmoid、ReLu, tanh

lilu916 7389次阅读 09-03 APP打开

【机器学习】神经网络-激活函数-面面观(Activation Function)

cyh24 60678次阅读 01-27 APP打开

神经网络之激活函数(Activation Function)

memray 19945次阅读 05-18 APP打开

深度学习：神经网络中的激活函数

pipisorry 20888次阅读 05-04 APP打开

烘焙糕点 哪里好

几种常见的激活函数

alwaystry 1387次阅读 03-05 APP打开

Deep Learning–TensorFlow (13) CNN卷积神经网络_ GoogLeNet之Inception(V1–V4)

u013751160 11190次阅读 04-24 APP打开

激活函数的作用

bojackhosreman 25665次阅读 04-06 APP打开

深度神经网络及TensorFlow实现1–激活函数（Activation Function）

hongxue8888 3329次阅读 08-11 APP打开

帕萨特1.4 超底价

机器学习09--神经网络的激活函数(Activation Function)及python代码实现

carmelcarmen 1352次阅读 02-22 APP打开

深度学习之常用激活函数

liuy9803 693次阅读 08-10 APP打开

激活函数之sigmoid激活（1）

zbbz1000 15739次阅读 03-24 APP打开

机器学习理论篇之激活函数激活函数