Activation function

Reference

https://www.zhihu.com/question/22334626

https://en.wikipedia.org/wiki/Activation_function

Definition

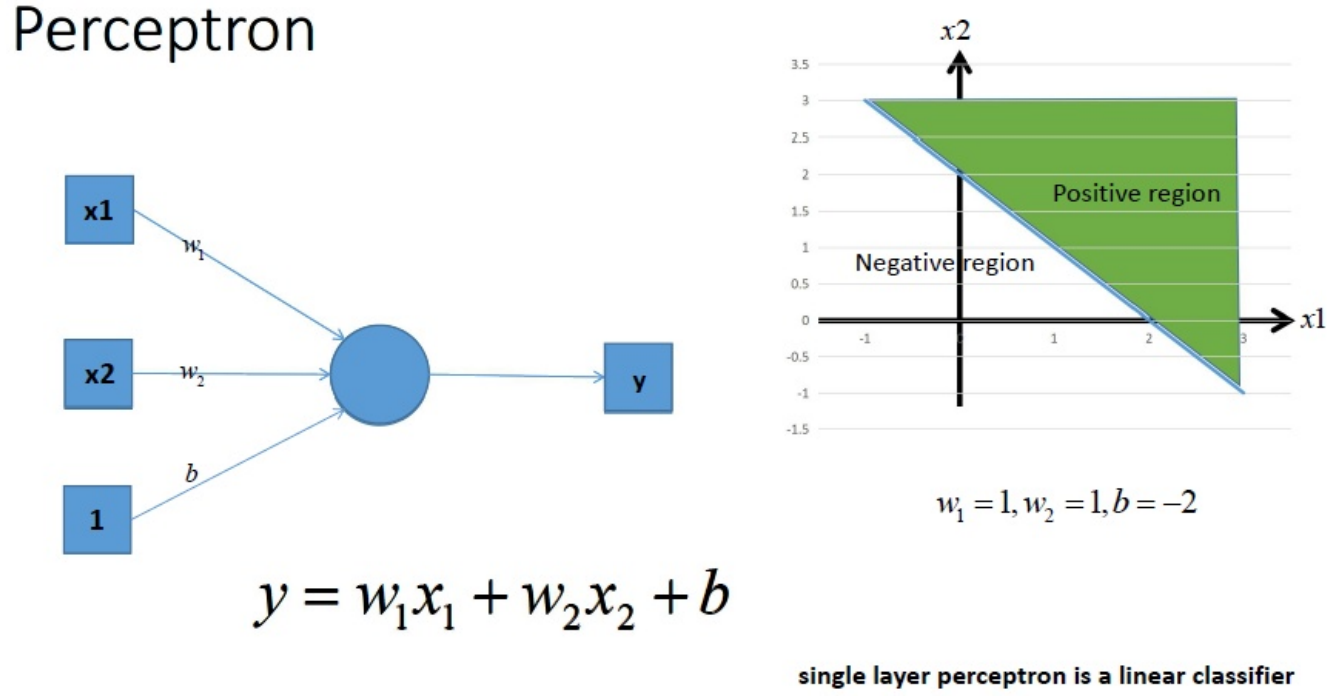
The activation function of a node defines the output of that node given an input or set of inputs.

Why do we need it?

为什么说activation function能实现非线性的分类？

没有hidden layer的神经网络 --- 线性划分

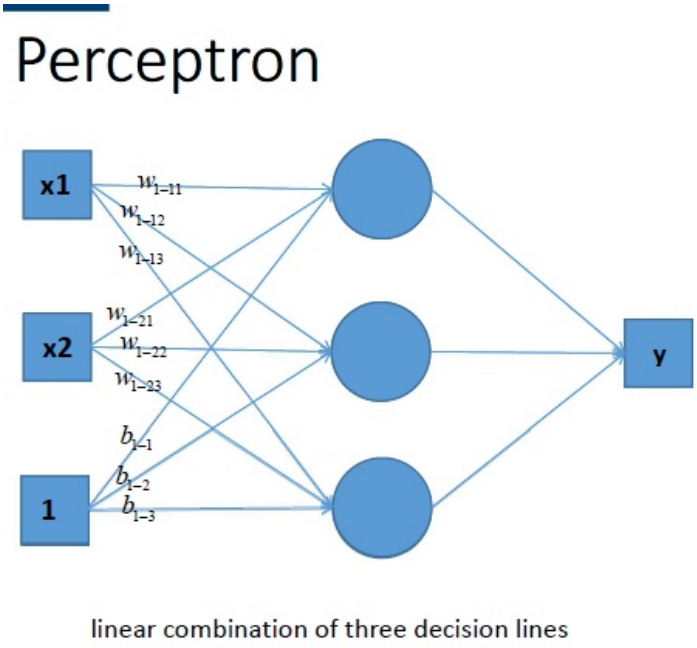
这是一个单层的感知机, 也是我们最常用的神经网络组成单元啦. 用它可以划出一条线, 把平面分割开。



single layer perceptron is a linear classifier

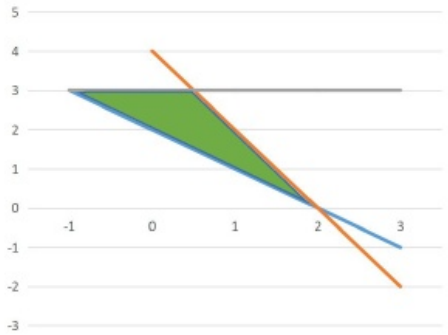
推广到全连接

那么很容易地我们就会想用多个感知机来进行组合, 获得更强的分类能力:



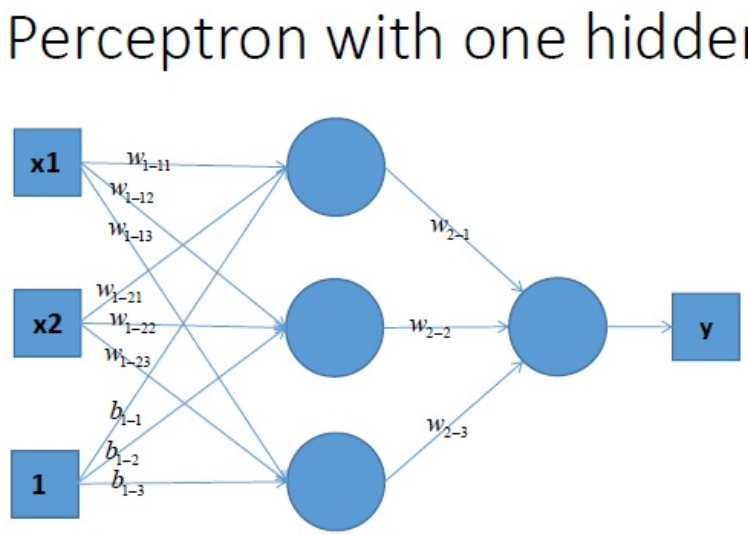
linear combination of three decision lines

single layer perceptron is a linear classifier



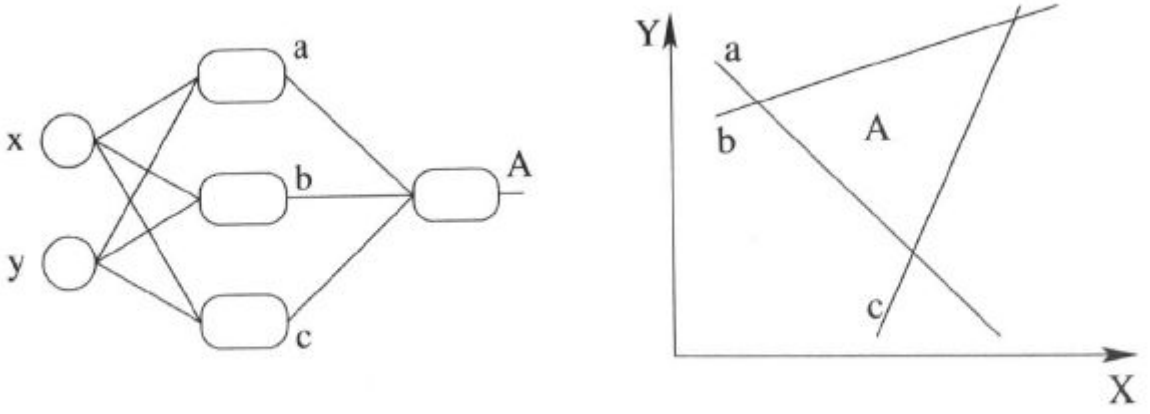
$$w_{1-11}=1, w_{1-12}=1, b_{1-1}=-2$$
$$w_{1-21}=2, w_{1-22}=1, b_{1-2}=4$$
$$w_{1-31}=0, w_{1-32}=1, b_{1-3}=3$$

如果加上隐藏层(不管加上多少层)，分类器还是线性的



$$y = w_{2\text{-}1}(w_{1\text{-}11}x_1 + w_{1-21}x_2 + b_{1\text{-}1}) + w_{2\text{-}2}(w_{1\text{-}12}x_1 + w_{1-22}x_2 + b_{1\text{-}2}) + w_{2\text{-}3}(w_{1\text{-}13}x_1 + w_{1-23}x_2 + b_{1\text{-}3})$$

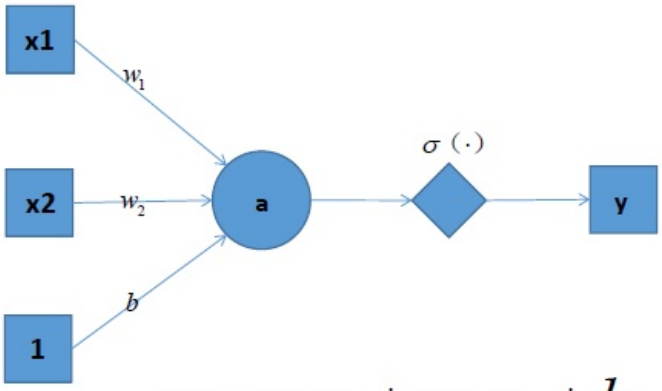
当然你可以说我们可以用无限条直线去逼近一条曲线啊！额,当然可以,不过比起用non-linear的activation function来说就太萌了！



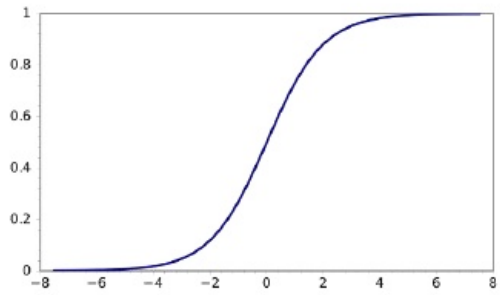
with step activation function

每一层叠加完了以后, 加一个激活函数

Perceptron with non-linear activation function



$$a = w_1 x_1 + w_2 x_2 + b$$
$$y = \sigma(a)$$

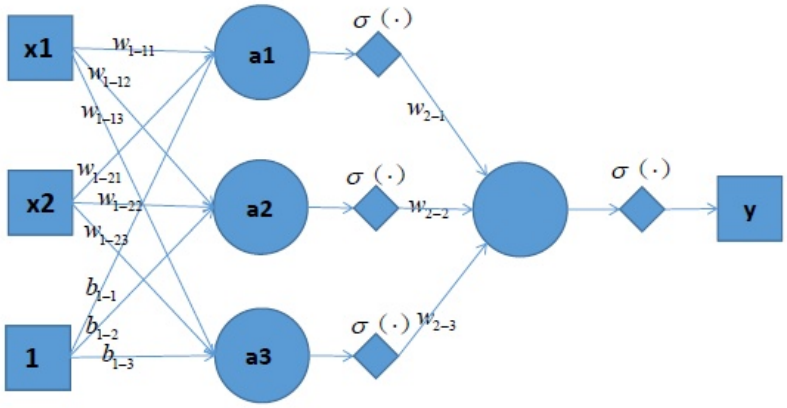


$\sigma(\cdot)$ is a non-linear activation function, sigmoid was the most popular one,

$$\sigma(y) = \frac{1}{1+e^{-y}}$$

复杂一点

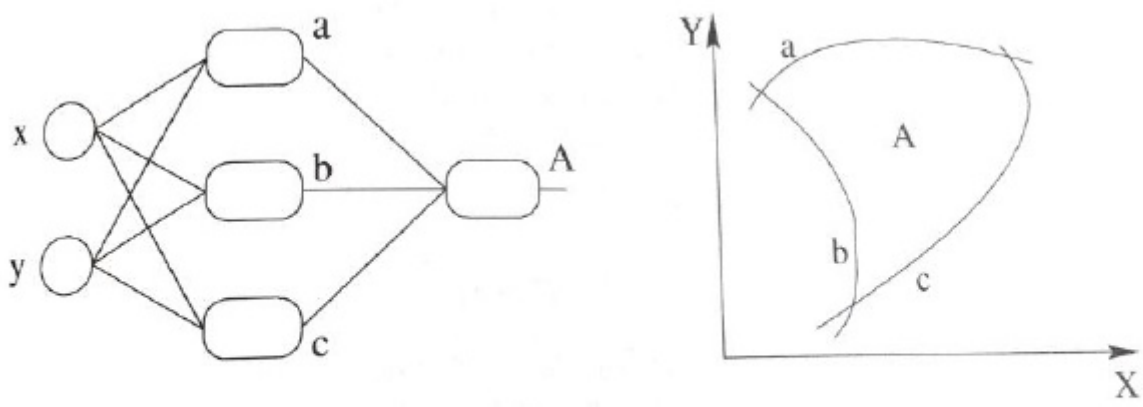
Perceptron with non-linear activation function



$$a1 = w_{1\text{-}11}x_1 + w_{1-21}x_2 + b_{1\text{-}1}$$
$$a2 = w_{1\text{-}12}x_1 + w_{1-22}x_2 + b_{1\text{-}2}$$
$$a3 = w_{1\text{-}13}x_1 + w_{1-23}x_2 + b_{1\text{-}3}$$

$$y = \sigma(w_{2-1}\sigma(a1) + w_{2-2}\sigma(a2) + w_{2-3}\sigma(a3))$$

加上非线性激活函数之后, 我们就有可能学习到这样的平滑分类平面。



with sigmoid activation function

Most common use activation functions

sigmoid

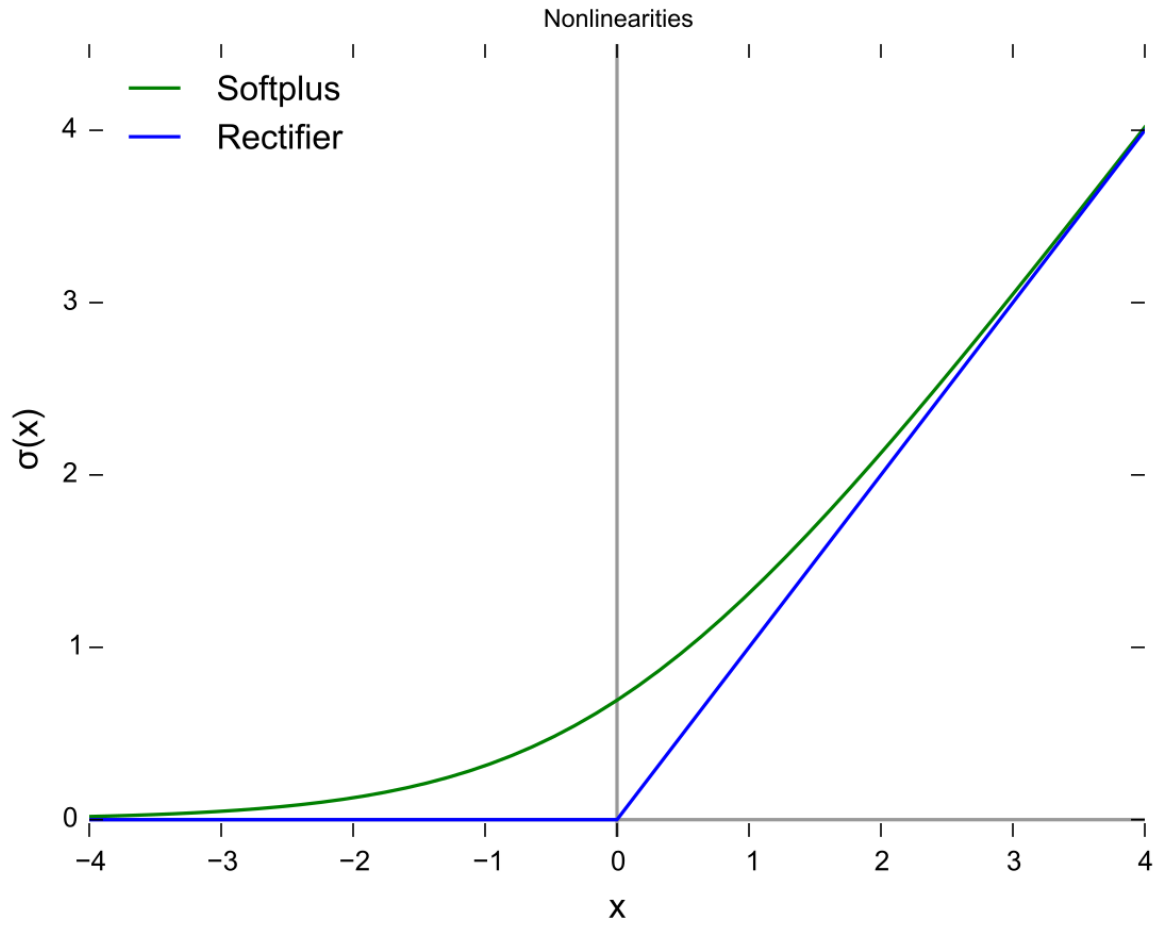
See in 02-Classification.md Sigmoid Function.

ReLu 整流线性单位函数（Rectified Linear Unit, ReLU）

The rectifier is an activation function defined as the positive part of its argument:

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

where x is the input to a neuron.



When $x=0$, $f(x)$ is not differentiable.

Thus, we have this **softplus or SmoothReLU function**.

$$f(x) = ln(1 + e^x)$$

The Derivative is $f'(x) = \frac{1}{1+e^{-x}}$, logistic function.

TanH (hyperbolic tangent)

$$tanh(x) = sinh(x)/cosh(x) = \frac{e^x - e^{-x}}{e^x + e-x}$$