School of Informatics, Computing and Engineering

Activation Function
CSCI-B 659

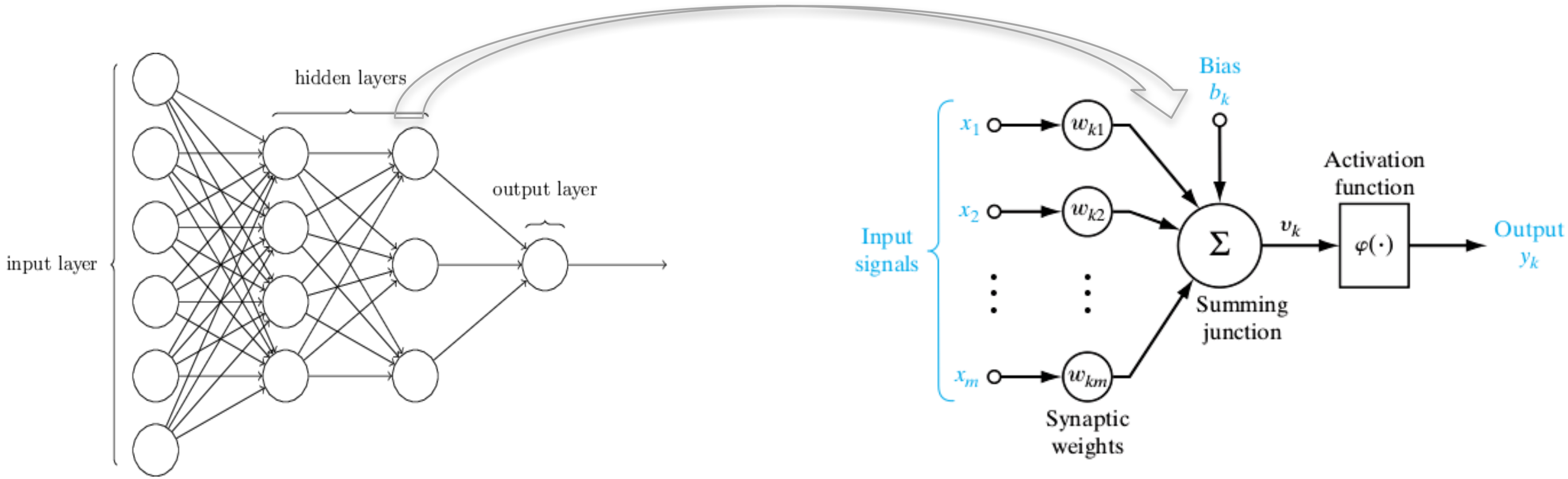
Khandokar Md. Nayem

INDIANA UNIVERSITY BLOOMINGTON

SECTION 1

What is AF?

Activation Function

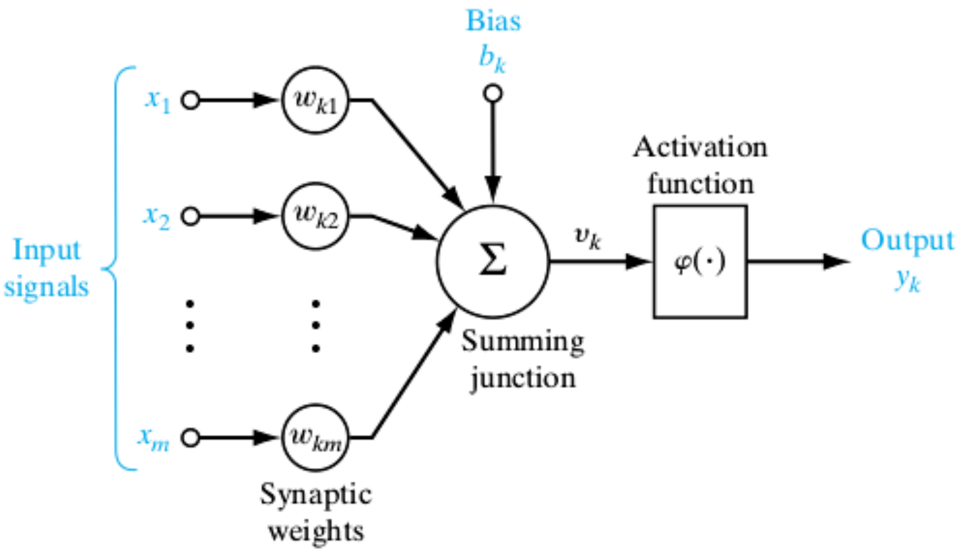


Definition

1. **Activation function** of a node defines the output of that node given an input or set of inputs.
2. In Biologically inspired the activation function is usually an abstraction representing the rate of action potential firing in the cell.
3. In its simplest form, this function is binary (Step Function) that is, either the neuron is firing or not.
4. A line of +ve slope may be used to reflect the increase in firing rate that occurs as input current increases.

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

Activation Function Equation



$$v_k = \sum_{i=0}^{m} w_{ki} x_i + b_k$$

$$y_k = \varphi(v_k)$$

SECTION 2

Why AF is important?

Nonlinearity in Neural Networks

1. A Neural Network without Activation function would simply be a **Linear regression Model.**

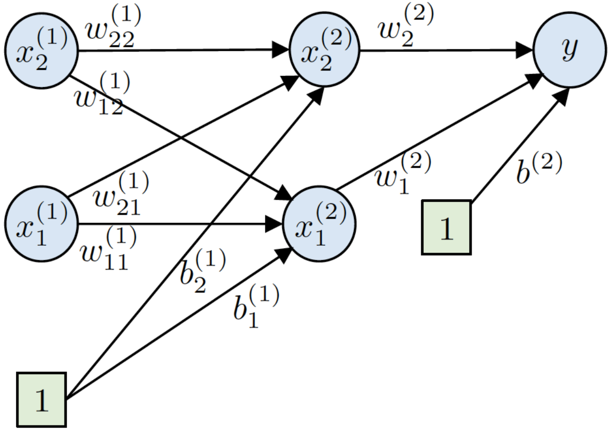
2. Stacked linear models form another linear model.

3. Non-linear functions are those which have degree more than one and they have a curvature when we plot a Non-Linear function.

4. We need to apply activation function so the network is more powerful and add ability to it to learn complex and complicated from data and represent non-linear complex arbitrary functional mappings between inputs and outputs.

XOR problem

$$x_1^{(2)} = w_{11}^{(1)} x_1^{(1)} + w_{12}^{(1)} x_2^{(1)} + b_1^{(1)}$$
$$x_2^{(2)} = w_{21}^{(1)} x_1^{(1)} + w_{22}^{(1)} x_2^{(1)} + b_2^{(1)}$$
$$y = w_1^{(2)} x_1^{(2)} + w_2^{(2)} x_2^{(2)} + b^{(2)}$$
$$= w_1^{(2)} w_{11}^{(1)} x_1^{(1)} + w_1^{(2)} w_{12}^{(1)} x_2^{(1)} + w_1^{(2)} b_1^{(1)}$$
$$+ w_2^{(2)} w_{21}^{(1)} x_1^{(1)} + w_2^{(2)} w_{22}^{(1)} x_2^{(1)} + w_2^{(2)} b_2^{(1)} + b^{(2)}$$

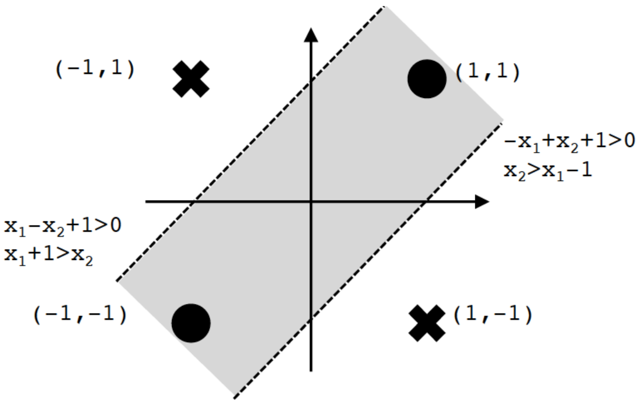


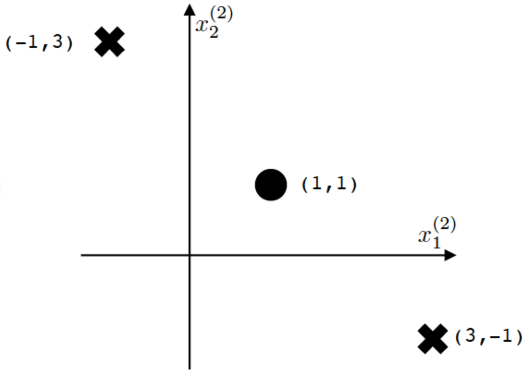
$$= (w_1^{(2)} w_{11}^{(1)} + w_2^{(2)} w_{21}^{(1)}) x_1^{(1)} + (w_1^{(2)} w_{12}^{(1)} + w_2^{(2)} w_{22}^{(1)}) x_2^{(1)} + w_1^{(2)} b_1^{(1)} + w_2^{(2)} b_2^{(1)} + b^{(2)}$$
$$= \boldsymbol{w}^{(2)} (\boldsymbol{W}^{(1)} \boldsymbol{x} + \boldsymbol{b}^{(1)}) + b^{(2)}$$
$$= \boldsymbol{w}^{(2)} \boldsymbol{W}^{(1)} \boldsymbol{x} + \boldsymbol{w}^{(2)} \boldsymbol{b}^{(1)} + b^{(2)}$$

Courtesy: Prof. Minje Kim, course ENGR-E 533

XOR problem

A nonlinear problem that needs two hyperplanes.



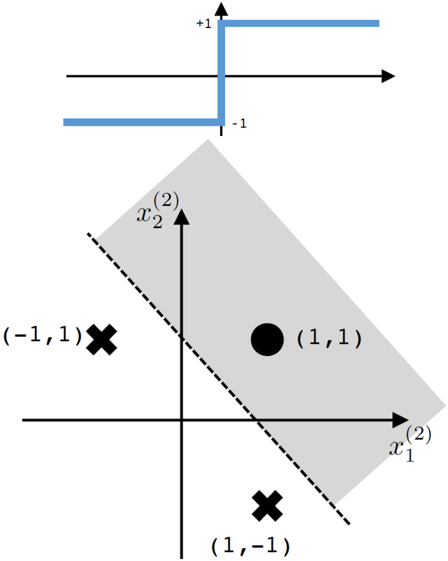


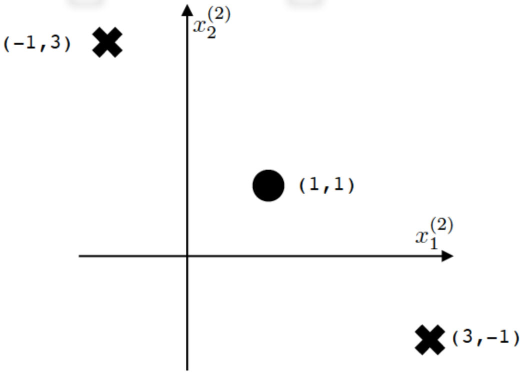
$$\boldsymbol{W}^{(1)} = \begin{bmatrix} +1 & -1 & 1 \\ -1 & +1 & 1 \end{bmatrix}$$

Still NOT linearly solvable!

XOR problem Solved

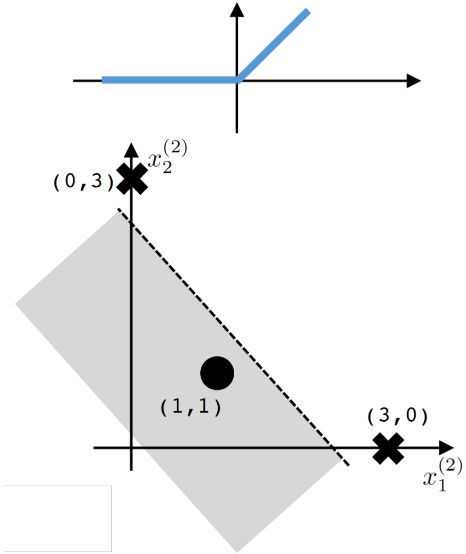
$$x_i^{(2)} = \text{sign}(\boldsymbol{W}_{i,:}^{(1)}\boldsymbol{x} + b_i^{(1)})$$





$$\boldsymbol{W}^{(1)} = \begin{bmatrix} +1 & -1 & 1 \\ -1 & +1 & 1 \end{bmatrix}$$

$$x_i^{(2)} = \max(\boldsymbol{W}_{i,:}^{(1)}\boldsymbol{x} + b_i^{(1)}, 0)$$



Courtesy: Prof. Minje Kim, course ENGR-E 533

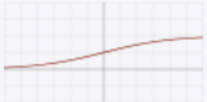
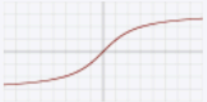
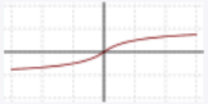
SECTION 3

Different types of AF

Linear Activation Functions

Name	Plot	Equation	Derivative (with respect to *x*)	Range
Identity		$f(x) = x$	$f'(x) = 1$	$(-\infty, \infty)$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$	$\{0, 1\}$

Nonlinear Activation Functions

Name	Plot	Equation	Derivative (with respect to *x*)	Range
Logistic (a.k.a. Soft step)		$f(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1 - f(x))$	$(0, 1)$
TanH		$f(x) = \tanh(x) = \frac{(e^{x} - e^{-x})}{(e^{x} + e^{-x})}$	$f'(x) = 1 - f(x)^2$	$[-1, 1]$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2+1}$	$\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$
Softsign [7][8]		$f(x) = \frac{x}{1+\lvert x\rvert}$	$f'(x) = \frac{1}{(1+\lvert x\rvert)^2}$	$(-1, 1)$

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

Nonlinear Activation Functions

Name	Plot	Equation	Derivative (with respect to *x*)	Range
Rectified linear unit (ReLU)[10]		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$[0, \infty)$
Leaky rectified linear unit (Leaky ReLU)[11]		$f(x) = \begin{cases} 0.01x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0.01 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$(-\infty, \infty)$
Parametric rectified linear unit (PReLU)[12]		$f(\alpha, x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(\alpha, x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$(-\infty, \infty)$

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

Nonlinear Activation Functions

Name	Plot	Equation	Derivative (with respect to *x*)	Range
SoftPlus[18]		$f(x) = \ln(1 + e^x)$	$f'(x) = \frac{1}{1 + e^{-x}}$	$(0, \infty)$
Softmax		$f_i(\vec{x}) = \frac{e^{x_i}}{\sum_{j=1}^{J} e^{x_j}}$	$\frac{d f_j(\vec{x})}{d x_j} = f_i(\vec{x})(\delta_{ij} - f_j(\vec{x}))$	(0,1)

SECTION 4

General Practice of AF

General Practice

1. Sigmoid functions and their combinations generally work better in the case of classifiers.
2. Sigmoids and tanh functions are avoided due to the vanishing gradient problem.
3. ReLU function is a general activation function and is used in most cases these days.
4. Leaky ReLU function is the best choice in case of dead neurons problem.
5. ReLU function should only be used in the hidden layers.
6. As a rule of thumb, begin with using ReLU function and then move over to other activation functions, in case ReLU doesn't provide with optimum results.

https://www.analyticsvidhya.com/blog/2017/10/fundamentals-deep-learning-activation-functions-when-to-use-them/

Thank you!