AIL721: Deep Learning

Instructor: James Arambam

ScAI | Yardi School of Artificial Intelligence
Indian Institute of Technology Delhi

Class Announcements

- Please use your name and official email IDs in the piazza.
  - Enrollments without actual names and official email IDs will be removed.
- Guidelines regarding the project topic.
  - Applications of deep learning (or neural networks) in problems related to your respective branches.
  - Pick an application that interests you, and explore how best to apply learning algorithms to solve it.
  - Computer Vision, Natural Language Processing, Speech Recognition, Reinforcement Learning, Healthcare etc.
- Guest Lecture (online) on Training LLMs - Confirmed!

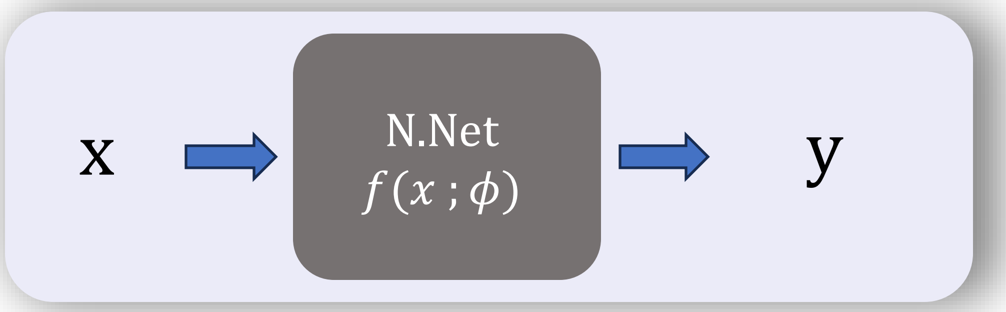
Dr. Maksim Tkachenko
Research Scientist,
Rakuten Institute of Technology, Singapore.

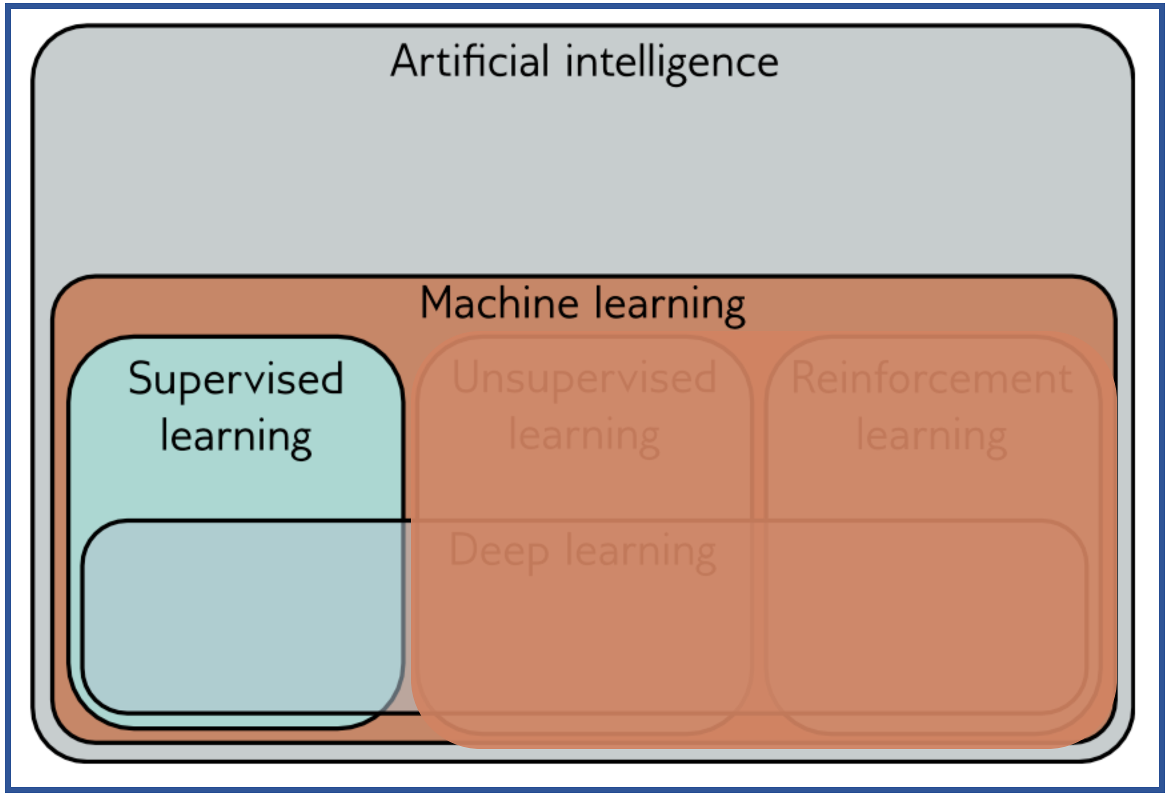
- IIT Delhi HPC Credit
  - How much for each student?


START

What is Deep Learning?

- Deep learning is a branch of **machine learning.**
- A general-purpose framework for **learning from data**.
- Based on computational models called **neural networks.**

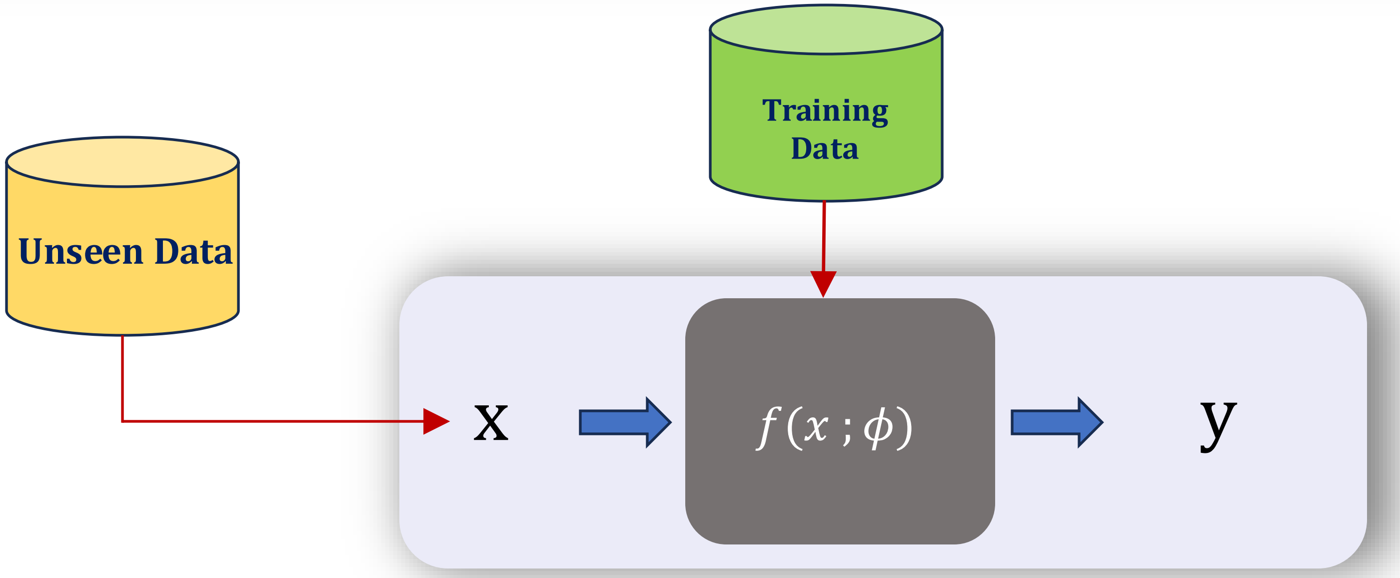




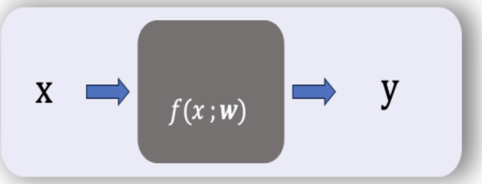
[Image from J.D Prince 2023 DL Book]

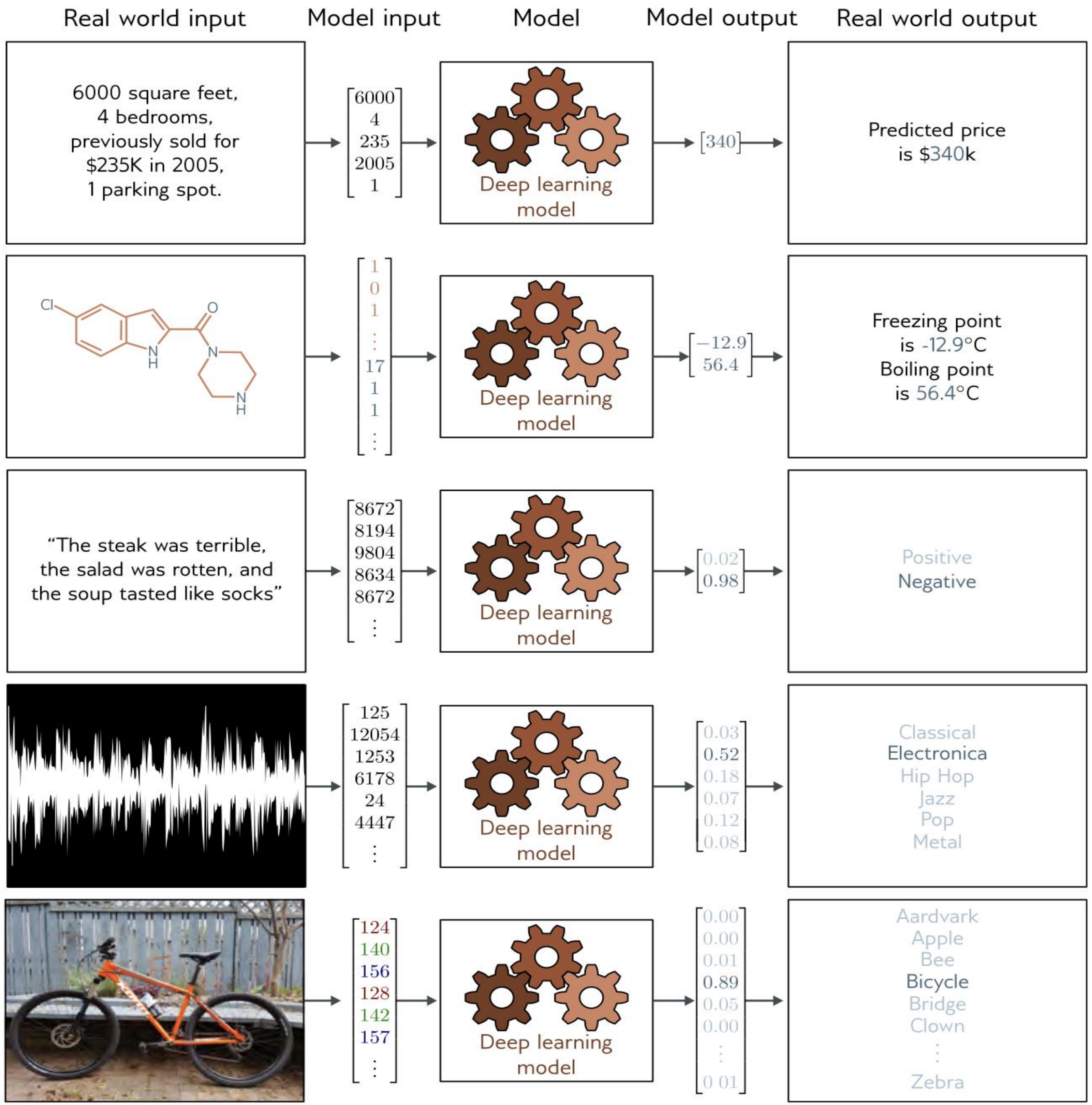
Supervised Learning

A type of **machine learning paradigm** that uses **labeled datasets** to train algorithms to **predict outcomes** and **recognize patterns**.



Examples





Regression

Multivariate Regression

Binary Classification

Multiclass Classification

[Image from J.D Prince 2023 DL Book]

Supervised Learning

Types of supervised learning:

- Regression problem: Model predicts **real values**.
- Classification problem: Model predicts **discrete values.**

$$x \rightarrow f(x\,;\phi) \rightarrow y$$

Regression

$$\mathbf{x} \Rightarrow f(x\,;\phi) \Rightarrow \mathbf{y}$$

$$< x_1, x_2, \ldots, x_d > \qquad\qquad < y_1, y_2, \ldots, y_m >$$

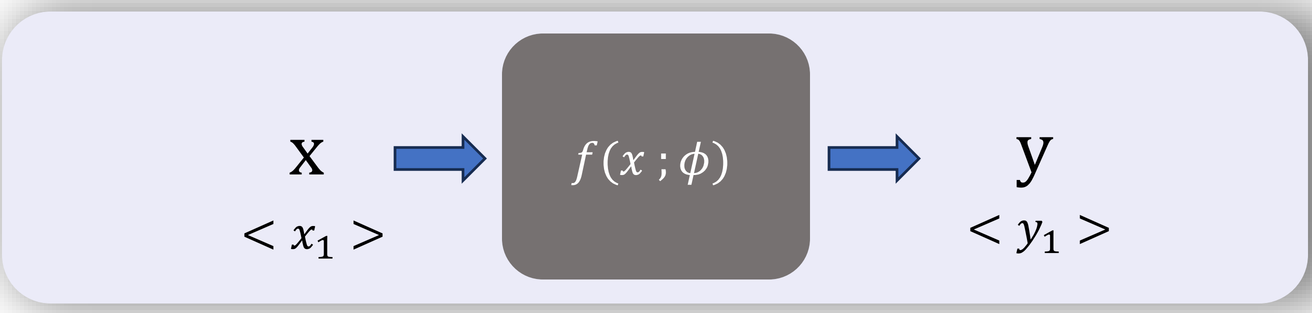
Regression

$$\mathbf{x} \Rightarrow f(x\,;\phi) \Rightarrow \mathbf{y}$$

$$< x_1 > \qquad\qquad < y_1 >$$

A Simple Regression Problem

Linear Regression



A Simple Regression Problem

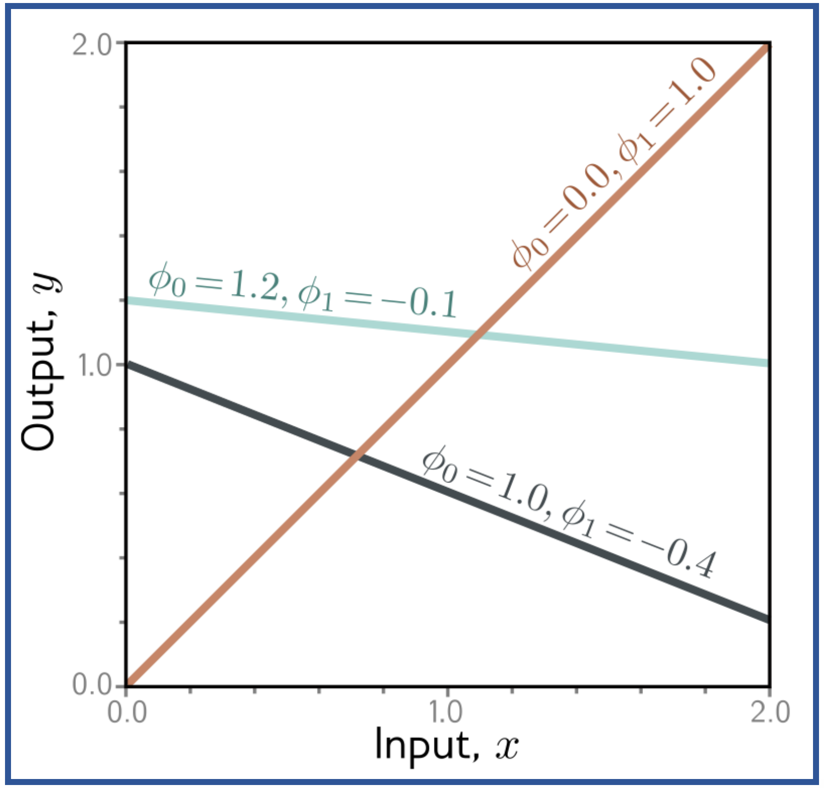
What is the simplest mathematical model to represent the function $f(x\,;\phi)$?

$$\boxed{f(x;\boldsymbol{\phi}) = \phi_0 + \phi_1 \cdot x}$$

$$\boxed{f(x;\boldsymbol{w}) = w_0 + w_1 \cdot x}$$

$$\boxed{f(x;\boldsymbol{\theta}) = \theta_0 + \theta_1 \cdot x}$$

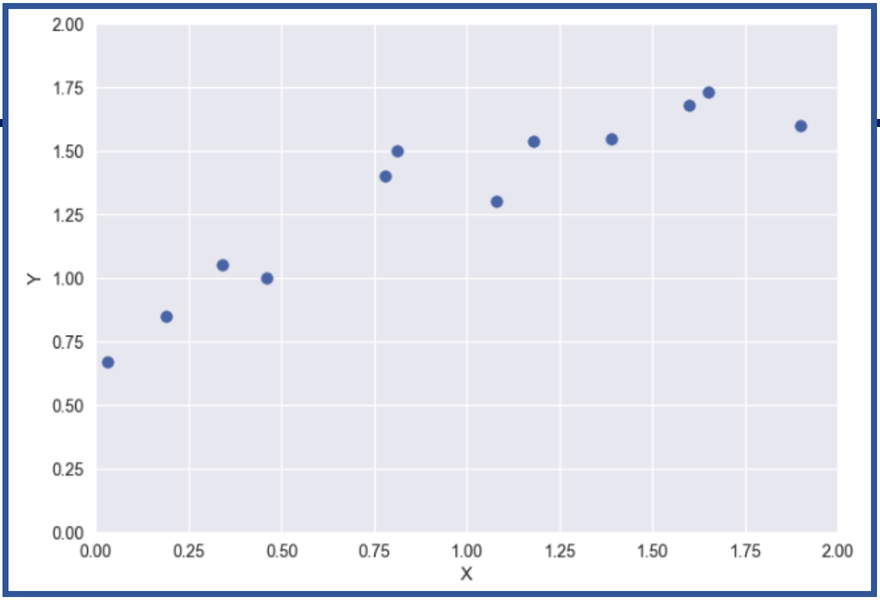
Linear Model

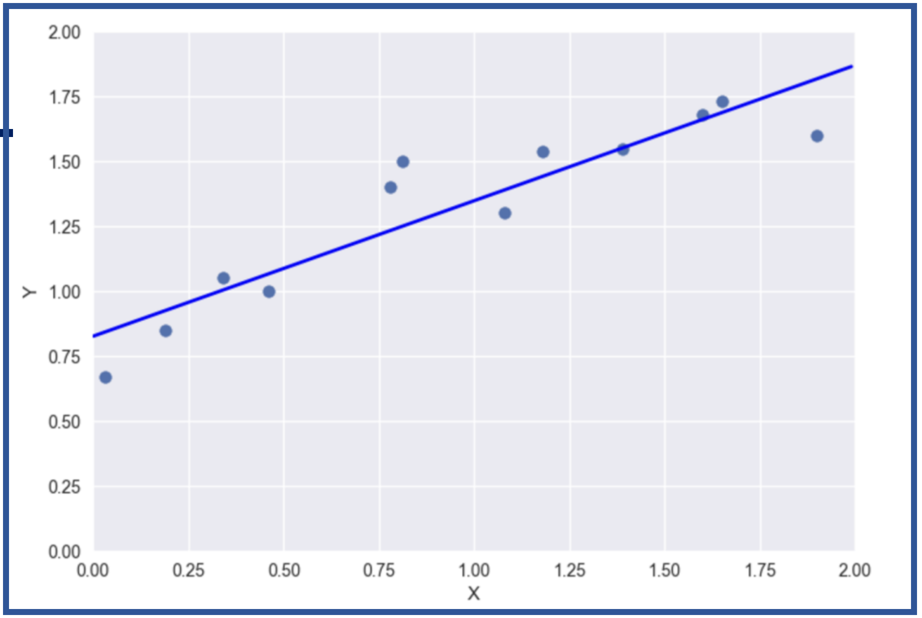


What's the limitation of such a linear model?

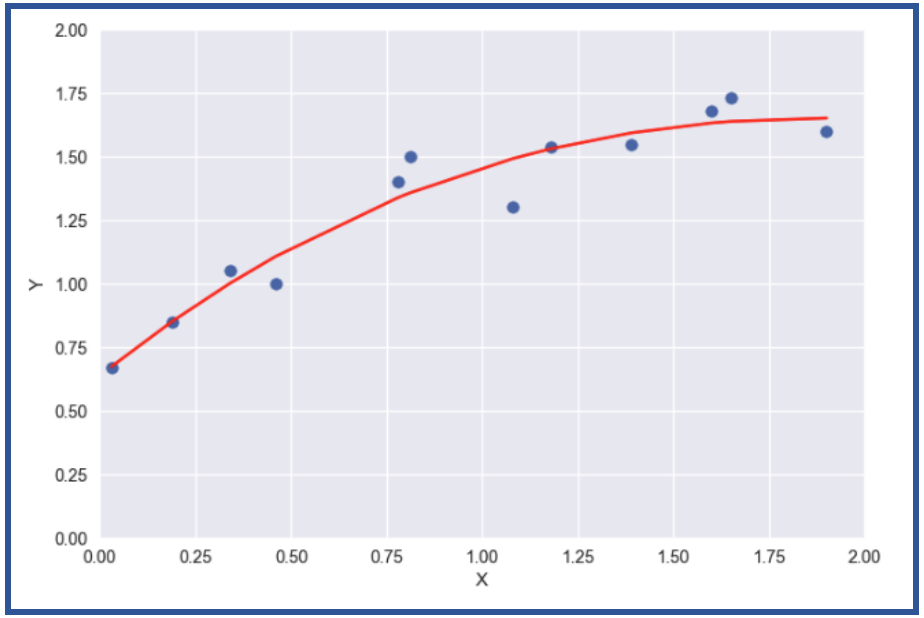
Linear models can only describe the input/output relationship as a line.

[Image from J.D Prince 2023 DL Book]

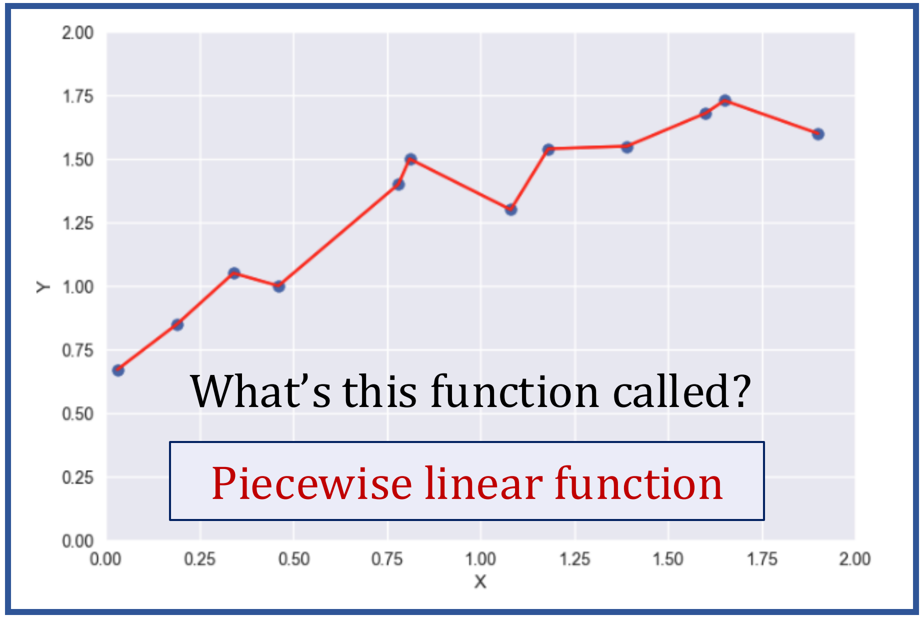




Linear function

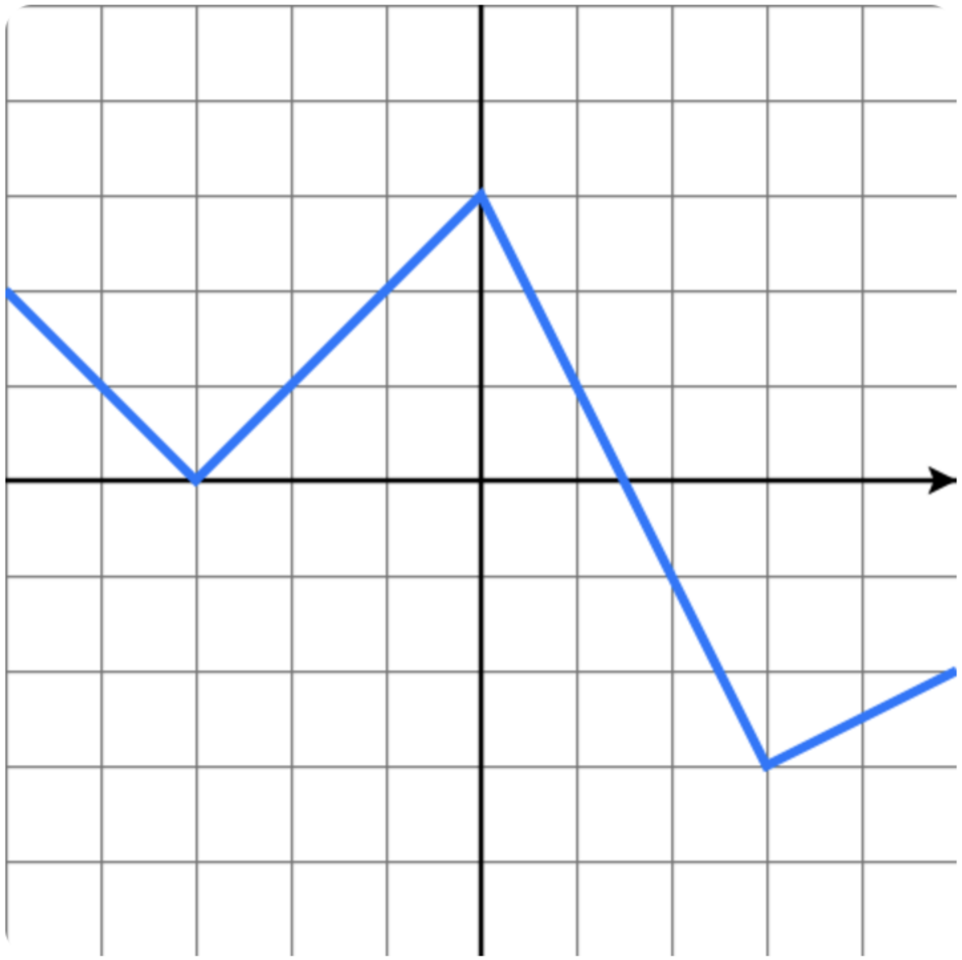


Smooth function



Piecewise Linear Function

$$f(x) = \begin{cases} -x-3 & \text{if } x \leq -3 \\ x+3 & \text{if } -3 < x < 0 \\ -2x+3 & \text{if } 0 \leq x < 3 \\ 0.5x - 4.5 & \text{if } x \geq 3 \end{cases}$$



A continuous piecewise linear function

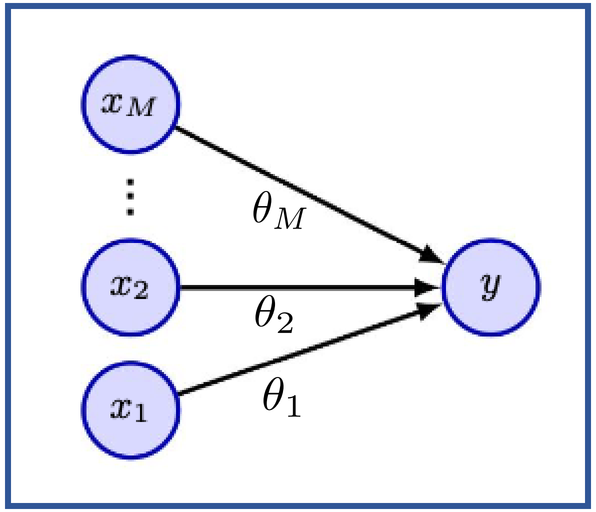
How piecewise linear function relates to neural networks?

Neural Networks

- Mathematically:

$$h = \sum_{i=1}^{M} \theta_i \cdot x_i \qquad (1)$$

$$y = a[h] \qquad (2)$$



$a[\cdot]$: **Activation Function**

[Image is from Bishop 2024 DL Book]

Activation Function

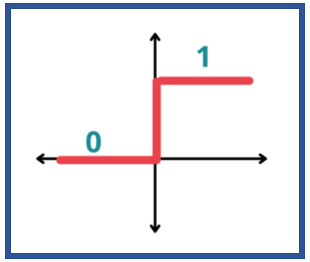
ScAI

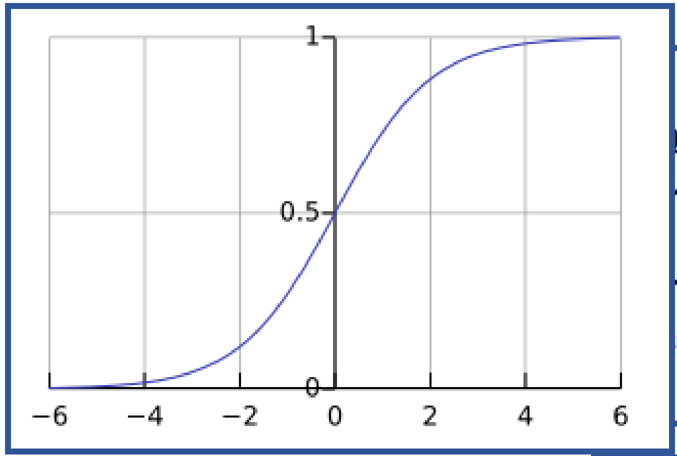
Earlier motivation:

- **Firing of neurons** depends on the **strength of the synapses**.
- Rosenblatt's Perceptron, 1962

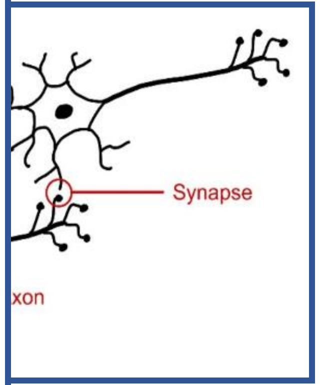
Activation function a[h] is a **step function**:

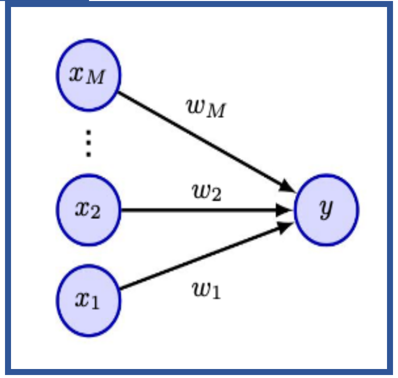
$$a[h] = \begin{cases} 0, & \text{if } h \leq 0, \\ 1, & \text{if } h > 0 \end{cases}$$





Sigmoid Function



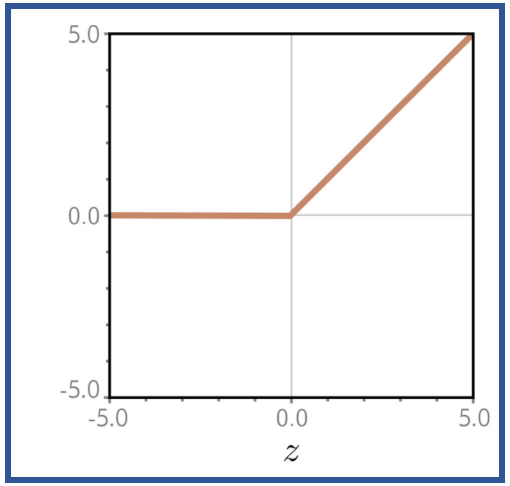


$$h = \sum_{i=1}^{M} \theta_i \cdot x_i$$

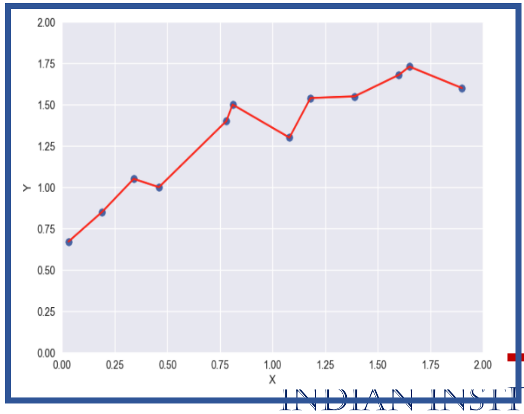
$$y = a[h]$$

- Depending on the type of a[h], it can introduce **non-linearity (a key property of NN).**

$$\text{a}[z] = \text{ReLU}[z] = \begin{cases} 0 & z < 0 \\ z & z \geq 0 \end{cases}$$



Piecewise linear function

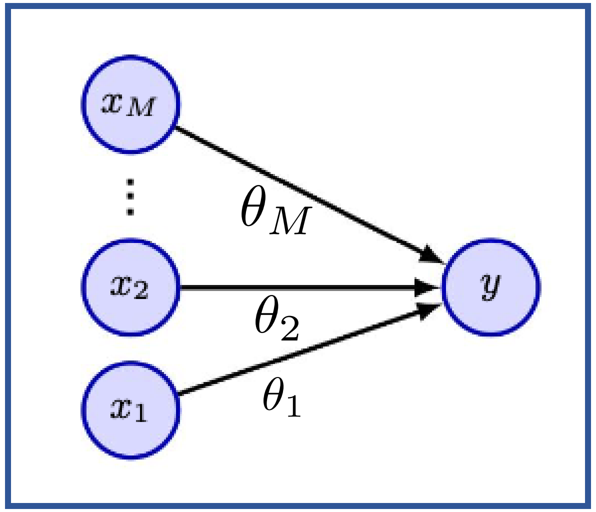


Neural Networks

❑ Mathematically:

$$h = \sum_{i=1}^{M} \theta_i \cdot x_i \qquad (1)$$

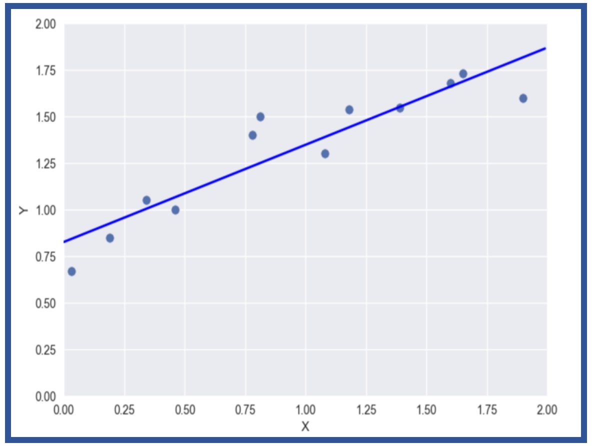
$$y = a[h] \qquad (2)$$



$$h = \theta_0 + \theta_1 \cdot x$$

$$y = \boxed{a[h] = h}$$

Linear or Nonlinear activation?



[Image is from Bishop 2024 DL Book]

$$h = \theta_0 + \theta_1 \cdot x$$

Linear or Nonlinear Function?

$$y = \phi_0 + \phi_1 \cdot a[h]$$

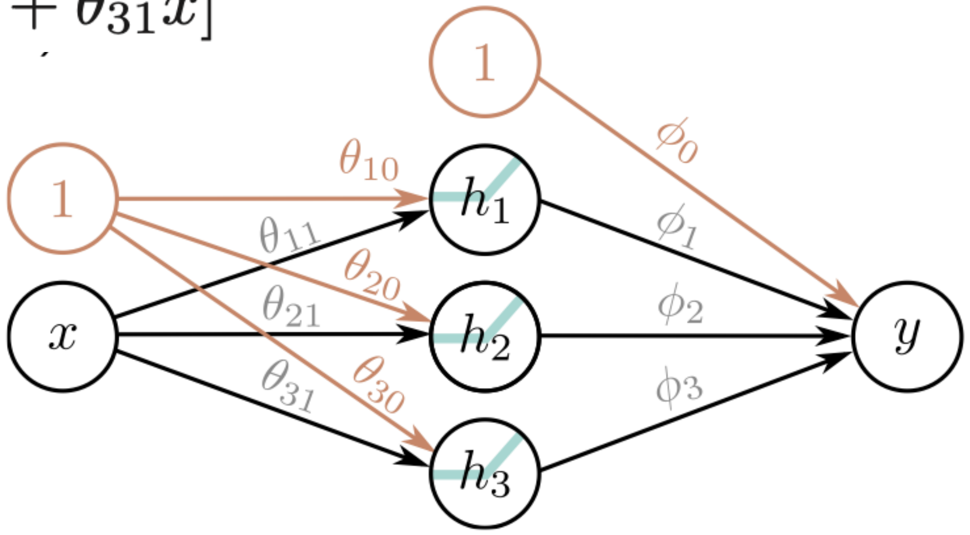
$$y = \phi_0 + \phi_1 \cdot a[\theta_{10} + \theta_{11} \cdot x]$$

Neural Networks

How many hidden nodes and hidden layers?

$$y = \phi_0 + \phi_1 \cdot a[\theta_{10} + \theta_{11} \cdot x]$$

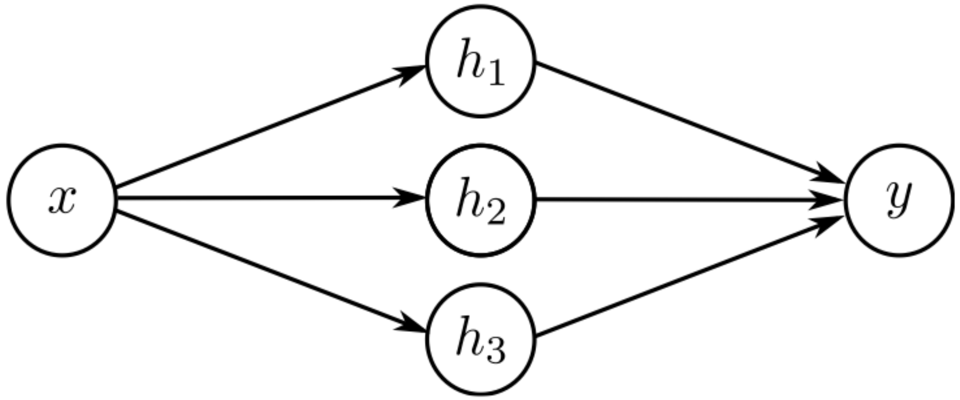
$$y = \phi_0 + \phi_1 \mathrm{a}[\theta_{10} + \theta_{11}x] + \phi_2 \mathrm{a}[\theta_{20} + \theta_{21}x] + \phi_3 \mathrm{a}[\theta_{30} + \theta_{31}x]$$



$$\begin{aligned} h_1 &= \mathrm{a}[\theta_{10} + \theta_{11}x] \\ h_2 &= \mathrm{a}[\theta_{20} + \theta_{21}x] \\ h_3 &= \mathrm{a}[\theta_{30} + \theta_{31}x]. \end{aligned}$$

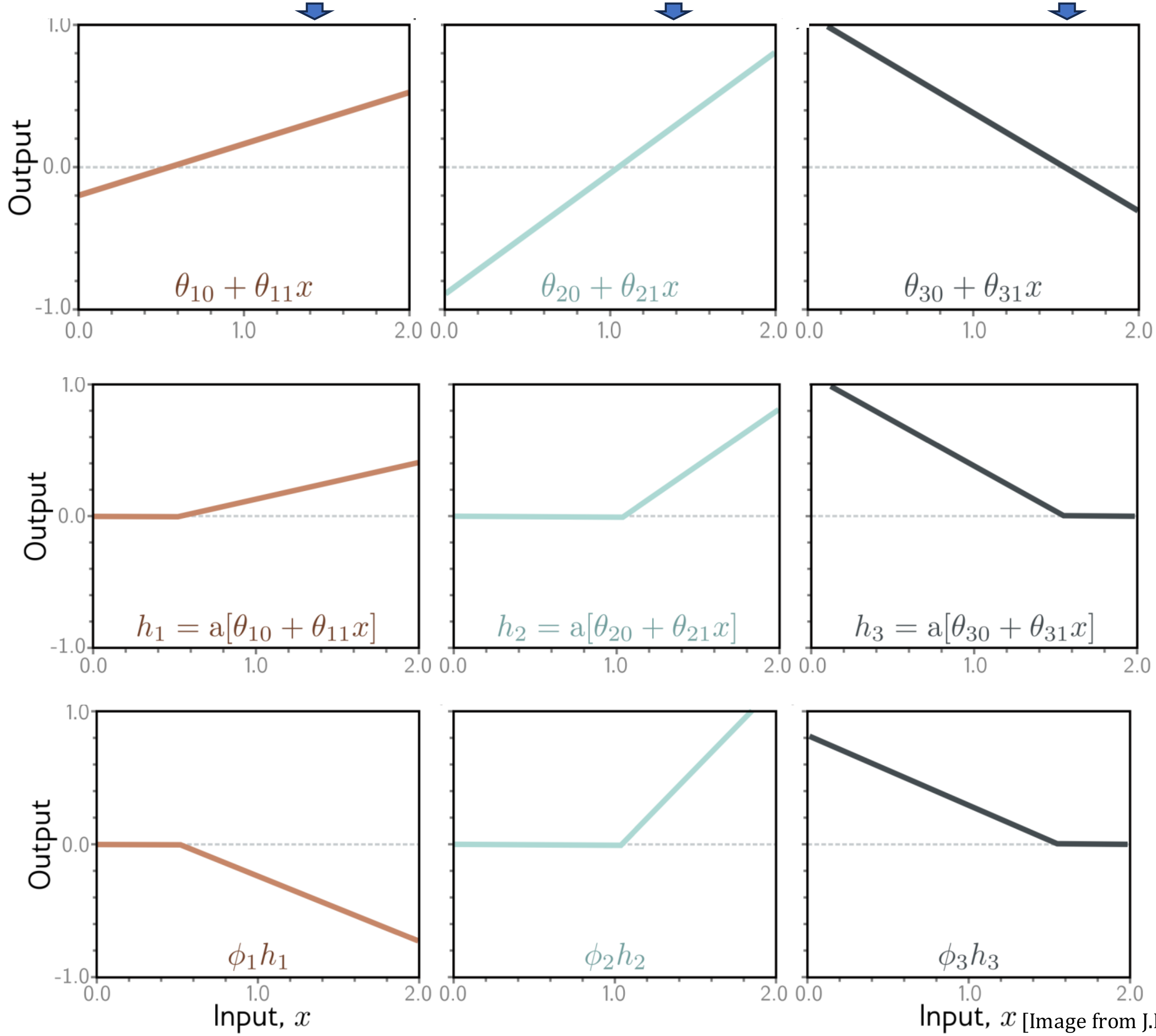
Hidden Units

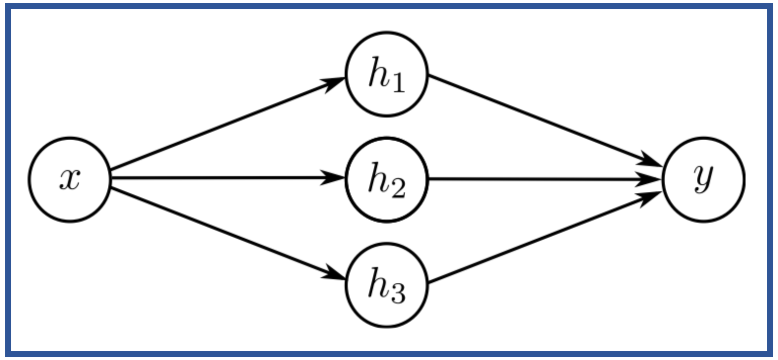
$$y = \phi_0 + \phi_1 h_1 + \phi_2 h_2 + \phi_3 h_3$$



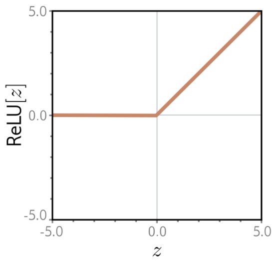
[Image from J.D Prince 2023 DL Book]

$$y = \phi_0 + \phi_1 \mathrm{a}[\theta_{10} + \theta_{11}x] + \phi_2 \mathrm{a}[\theta_{20} + \theta_{21}x] + \phi_3 \mathrm{a}[\theta_{30} + \theta_{31}x]$$

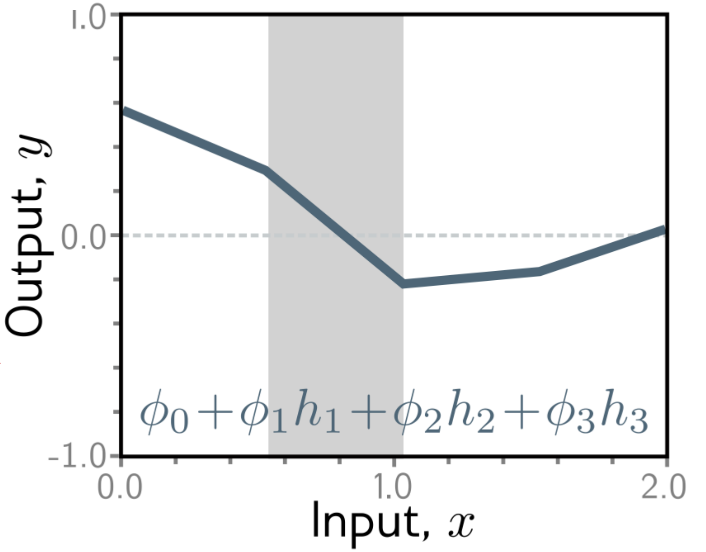




$\boldsymbol{a}[\cdot]$



ReLU Activation

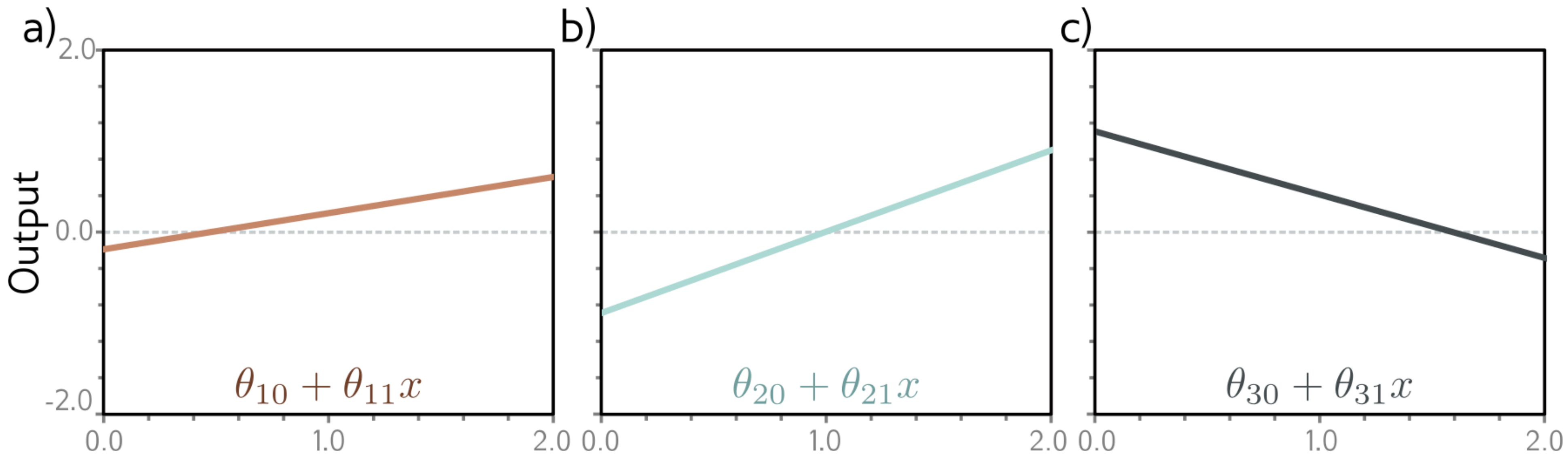


$$y = \phi_0 + \phi_1 h_1 + \phi_2 h_2 + \phi_3 h_3$$

[Image from J.D Prince 2023 DL Book]

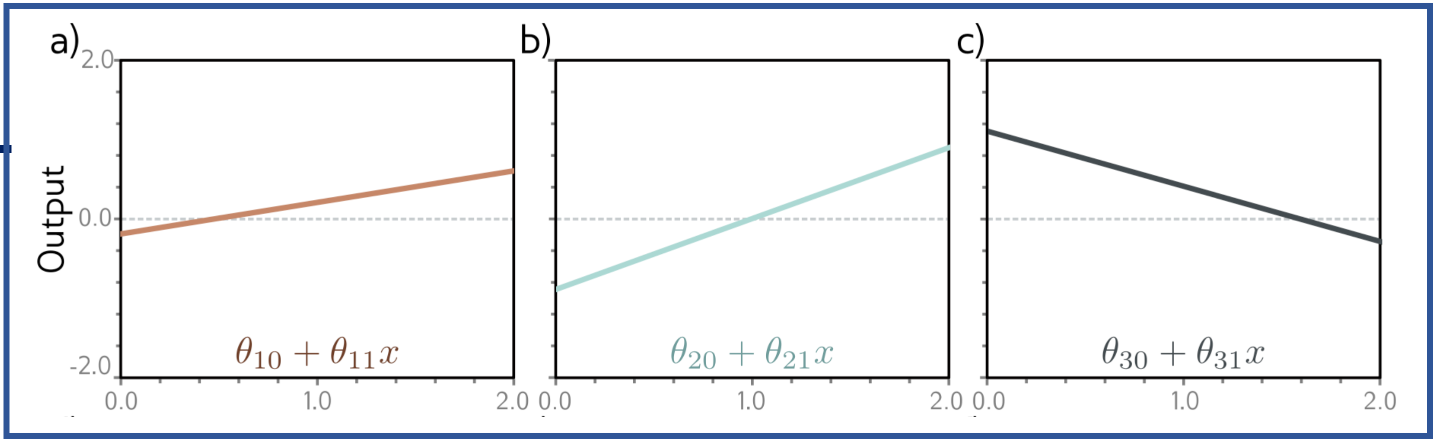
Exercise

$$y = \phi_0 + \phi_1 \mathrm{a}[\theta_{10} + \theta_{11}x] + \phi_2 \mathrm{a}[\theta_{20} + \theta_{21}x] + \phi_3 \mathrm{a}[\theta_{30} + \theta_{31}x]$$



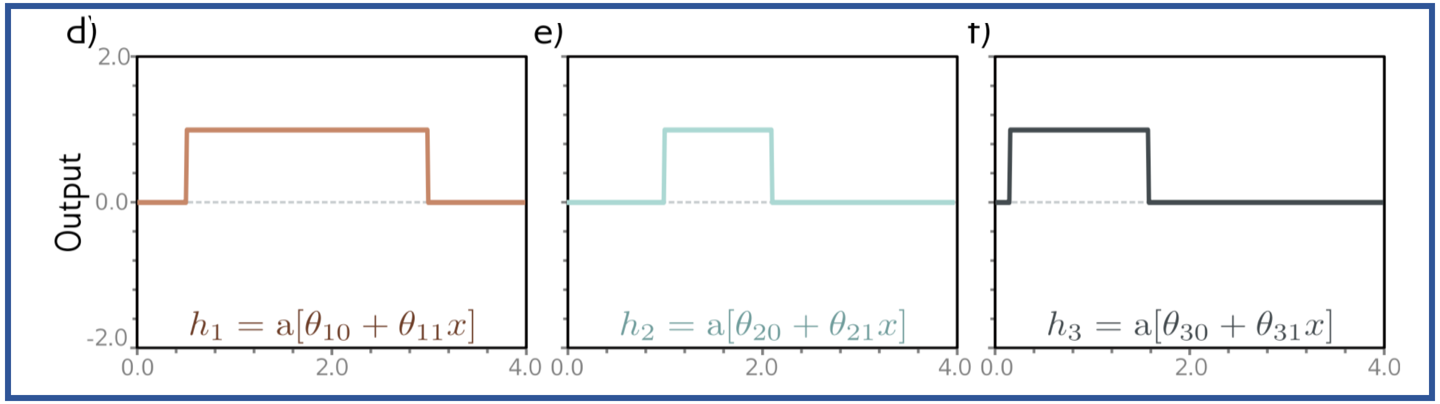
Activation Fn:

$$\mathrm{rect}[z] = \begin{cases} 0 & z < 0 \\ 1 & 0 \leq z \leq 1 \\ 0 & z > 1 \end{cases}$$

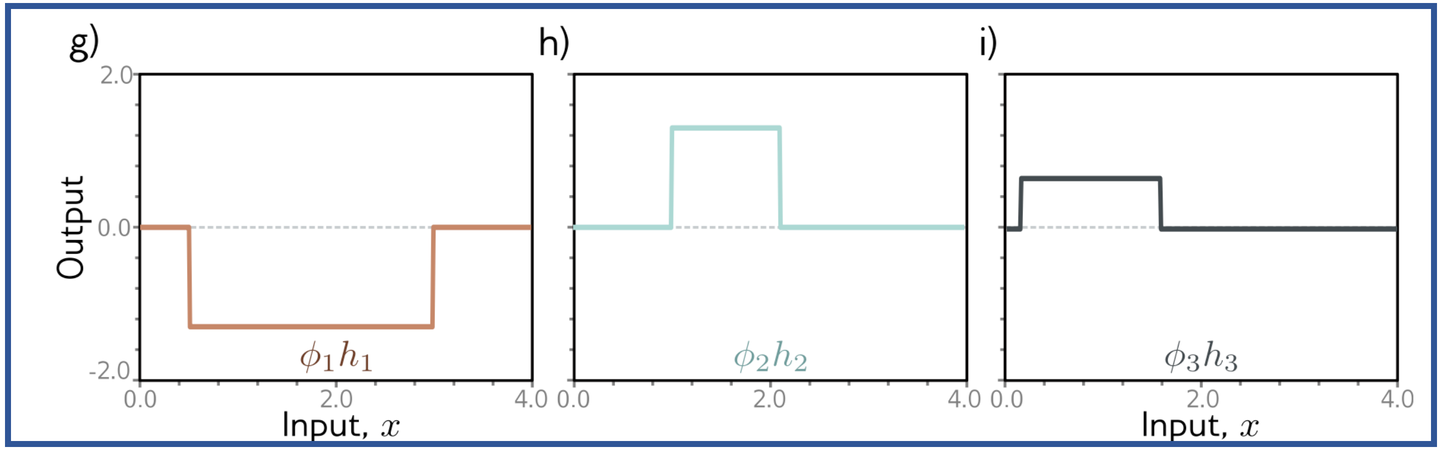


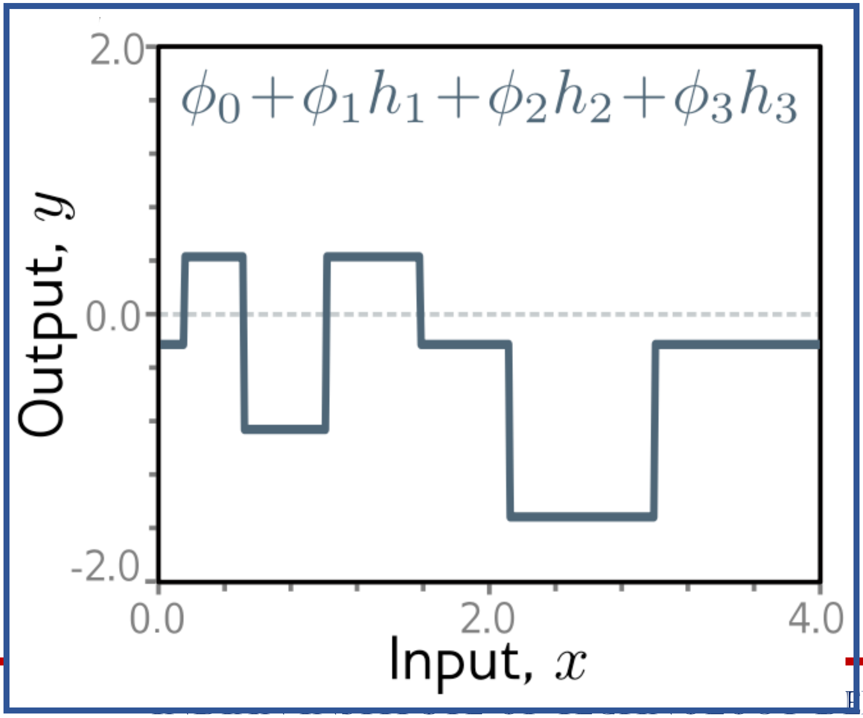
Activation Fn:

$$\mathrm{rect}[z] = \begin{cases} 0 & z < 0 \\ 1 & 0 \leq z \leq 1 \\ 0 & z > 1 \end{cases}$$

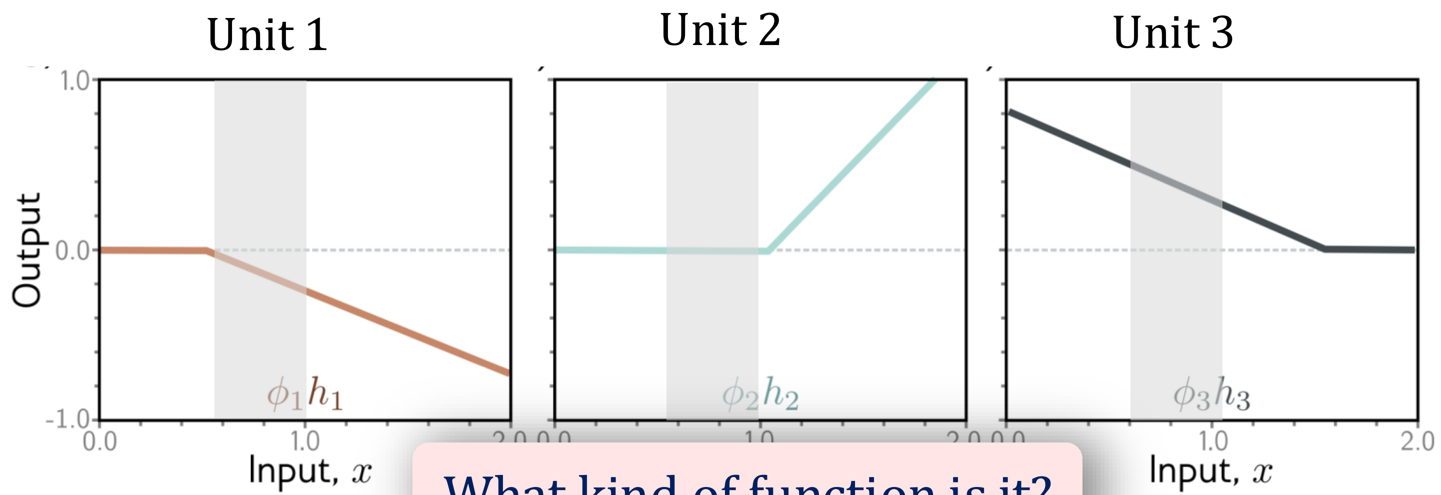


$$y = \phi_0 + \phi_1 h_1 + \phi_2 h_2 + \phi_3 h_3$$

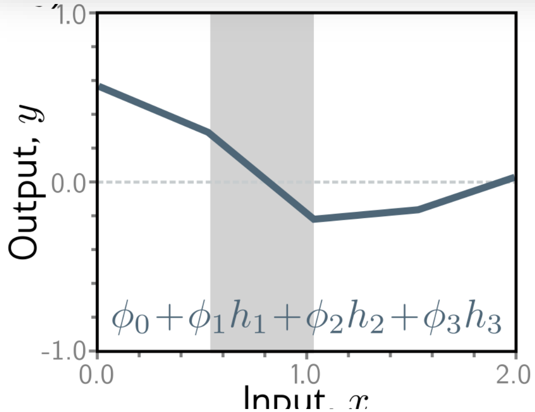


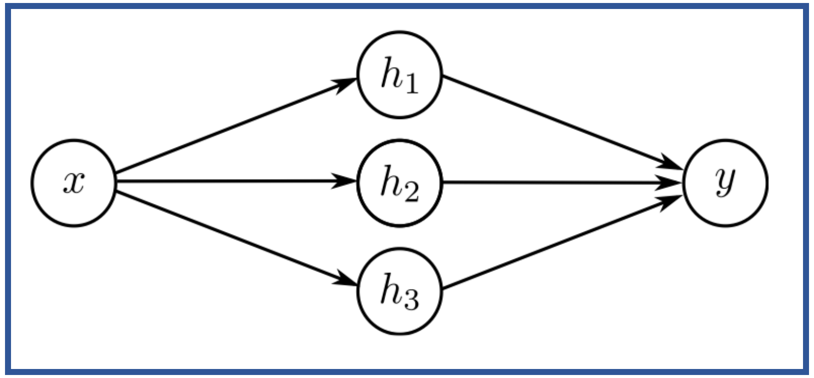


Activation Pattern



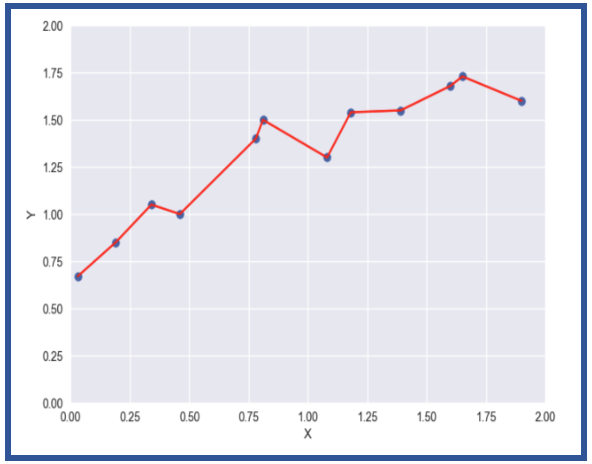
What kind of function is it?





Shaded region:

- Unit 1 active
- Unit 2 inactive
- Unit 3 active



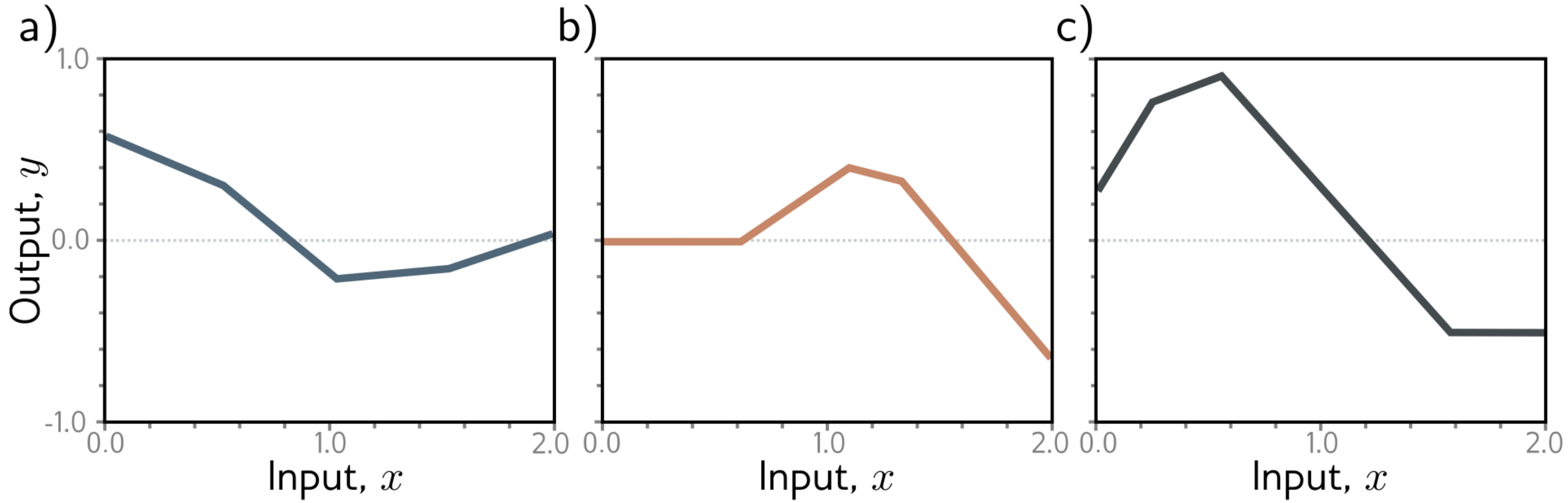
Piecewise linear function

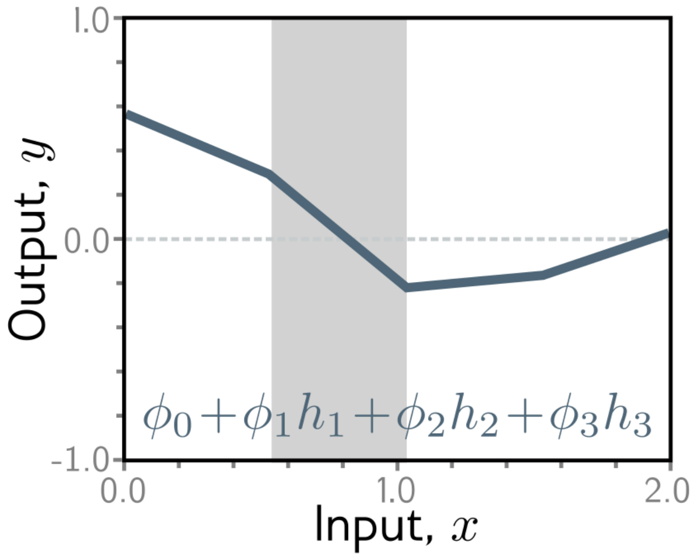
How piecewise linear function relates to neural networks?

[Image from J.D Prince 2023 DL Book]

Neural Networks

$$y = \phi_0 + \phi_1 \mathrm{a}[\theta_{10} + \theta_{11}x] + \phi_2 \mathrm{a}[\theta_{20} + \theta_{21}x] + \phi_3 \mathrm{a}[\theta_{30} + \theta_{31}x]$$





What factors influence the curve's shape?

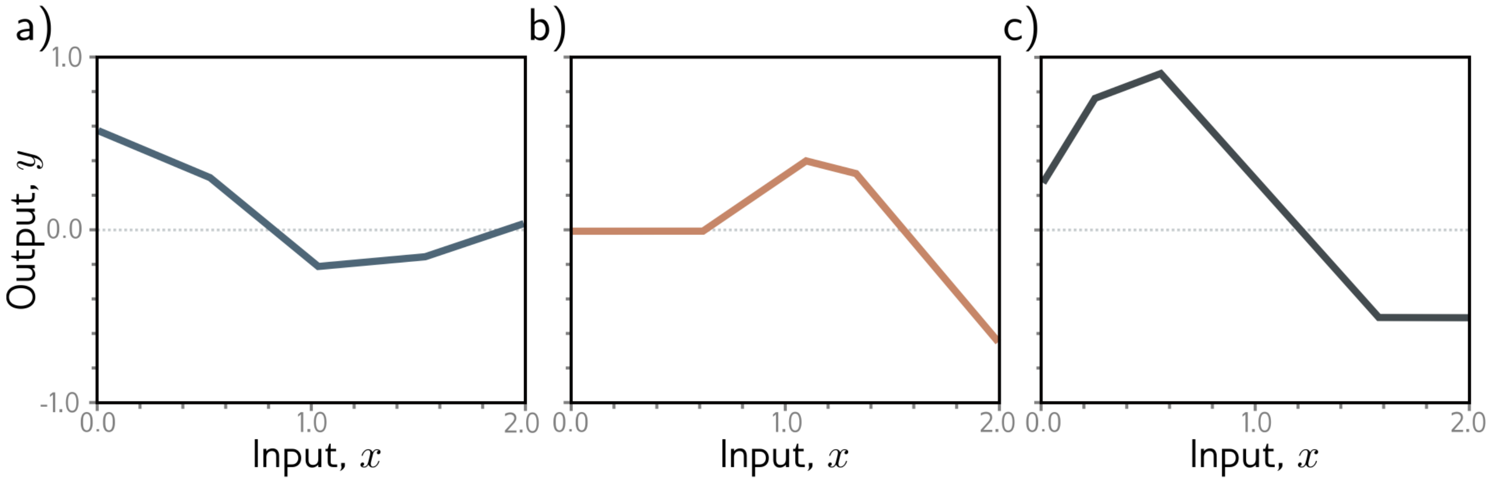
- **Number of model parameters/weights:** $\phi_0, \phi_1, \phi_2, \phi_3, \theta_{10}, \theta_{11}, \theta_{20}, \theta_{21}, \theta_{30}, \theta_{31}$
- **Type of activation function:** ReLU, Sigmoid, tanh, etc.

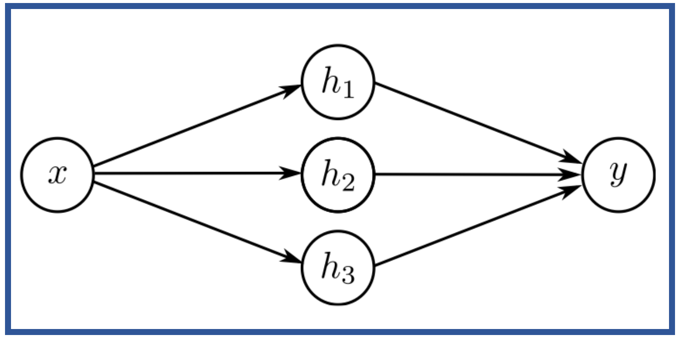
Data

One more key factor remaining. Anyone?

[Image from J.D Prince 2023 DL Book]

Neural Networks





How many linear regions (or segments) in the above piecewise linear function?

[Image from J.D Prince 2023 DL Book]

General Form of Neural Networks

Neural Network with three hidden units:

$$y = \phi_0 + \phi_1 \mathrm{a}[\theta_{10} + \theta_{11}x] + \phi_2 \mathrm{a}[\theta_{20} + \theta_{21}x] + \phi_3 \mathrm{a}[\theta_{30} + \theta_{31}x]$$

Neural Network with "D" hidden units:

$$h_d = \mathrm{a}[\theta_{d0} + \theta_{d1}x] \qquad y = \phi_0 + \sum_{d=1}^{D} \phi_d h_d$$

How many linear regions (or segments) with "D" hidden units under ReLU activation?

D+1

General Form of Neural Networks

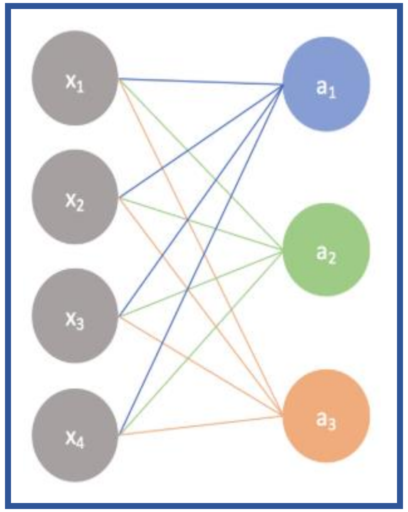
Neural Network with three hidden units:

$$y = \phi_0 + \phi_1 \mathrm{a}[\theta_{10} + \theta_{11}x] + \phi_2 \mathrm{a}[\theta_{20} + \theta_{21}x] + \phi_3 \mathrm{a}[\theta_{30} + \theta_{31}x]$$

Neural Network with “D” hidden units:

$$h_d = \mathrm{a}[\theta_{d0} + \theta_{d1}x] \qquad y = \phi_0 + \sum_{d=1}^{D} \phi_d h_d$$

Matrix Form:

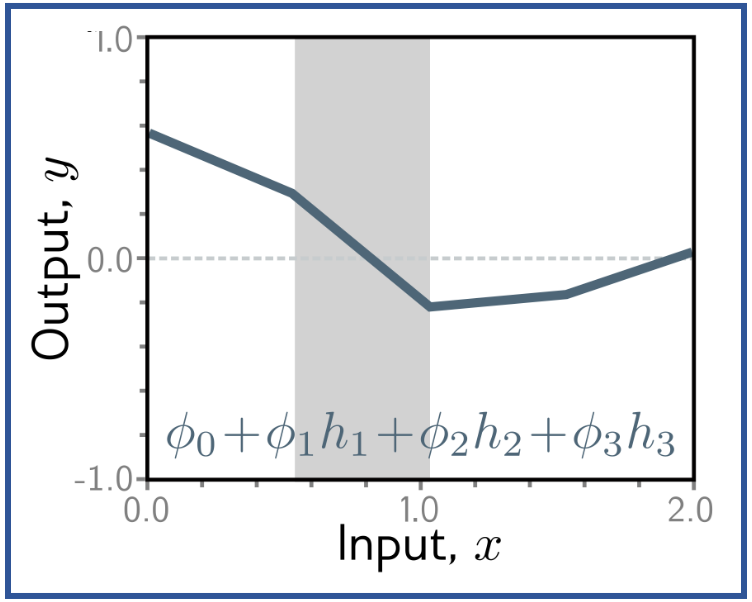


$$\begin{bmatrix} w_1 & w_2 & w_3 & w_4 \\ w_1 & w_2 & w_3 & w_4 \\ w_1 & w_2 & w_3 & w_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} b \\ b \\ b \end{bmatrix} = \begin{bmatrix} w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + b \\ w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + b \\ w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + b \end{bmatrix} \xrightarrow[activation]{} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

Neural Networks

❑ Neural Network with three hidden units:

$$y = \phi_0 + \phi_1 \mathrm{a}[\theta_{10} + \theta_{11}x] + \phi_2 \mathrm{a}[\theta_{20} + \theta_{21}x] + \phi_3 \mathrm{a}[\theta_{30} + \theta_{31}x]$$

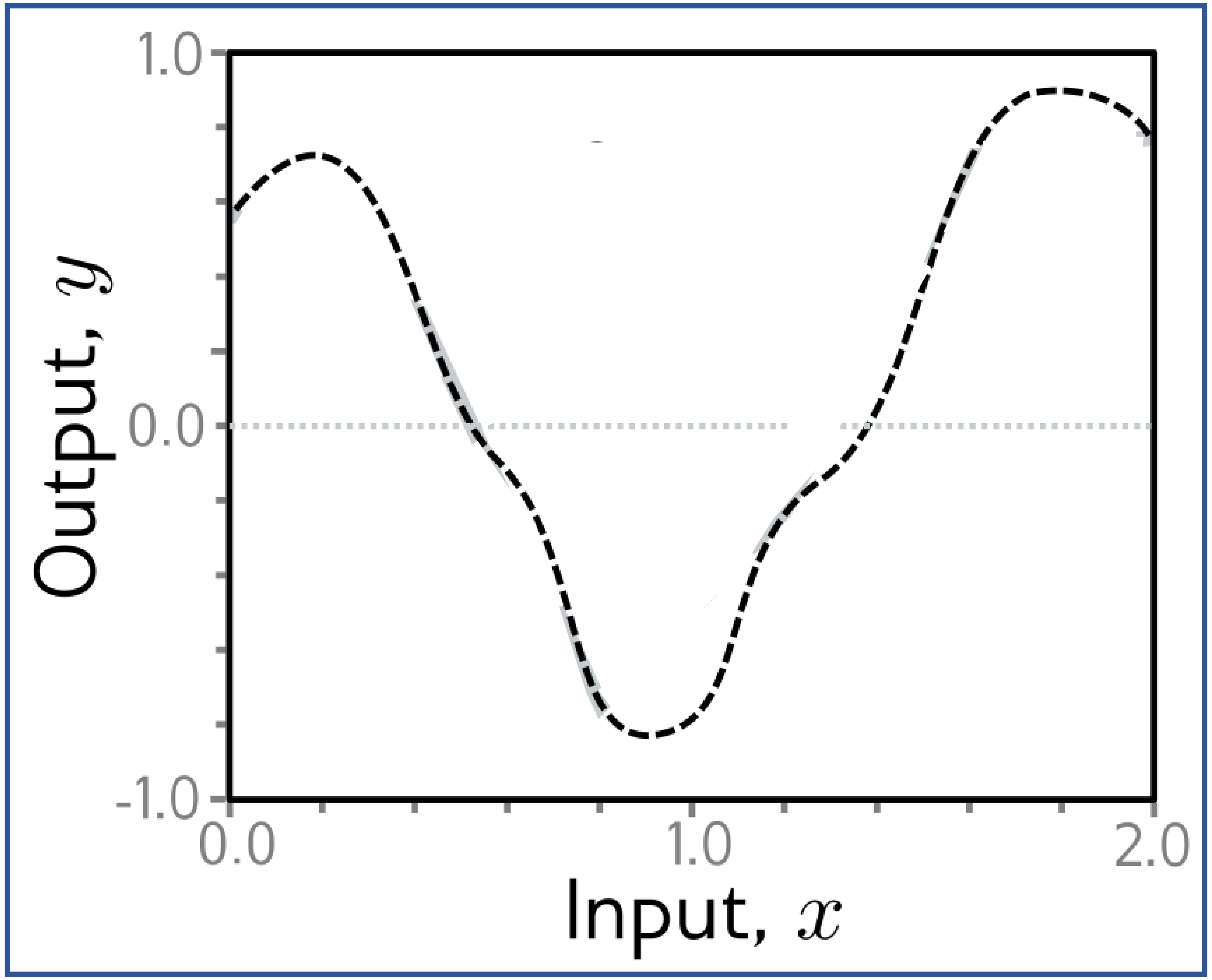


What the function be like if a[z] = $\psi_0 + \psi_1 \cdot z$?

[Image from J.D Prince 2023 DL Book]

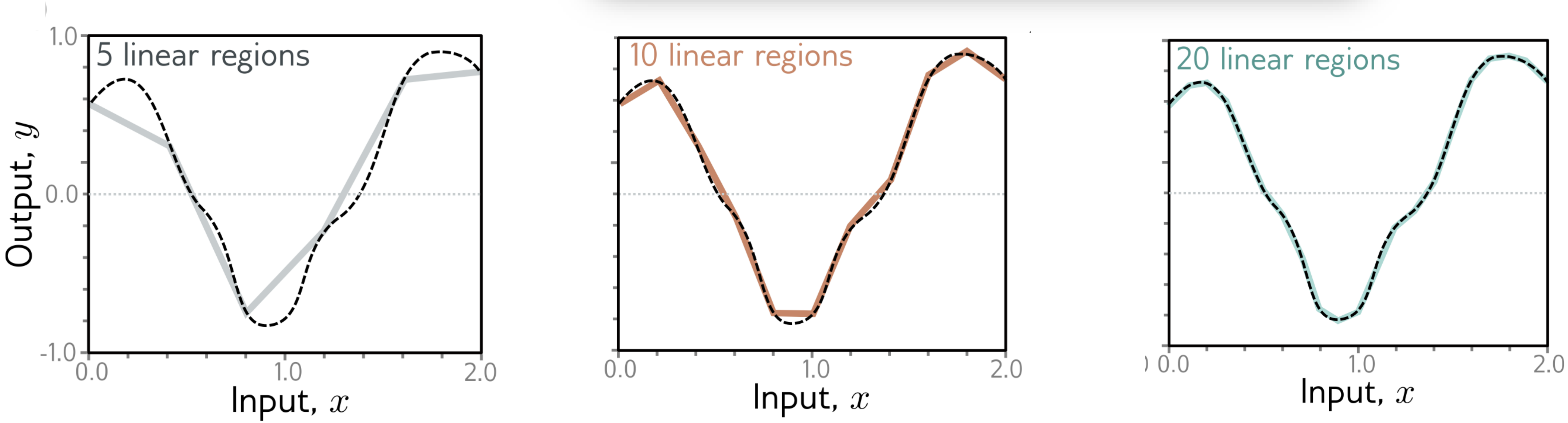
Neural Networks

Any arbitrary continuous function.



[Image from J.D Prince 2023 DL Book]

Power of Neural Network



With enough hidden units (linear regions), we can describe any 1D function with arbitrary accuracy.

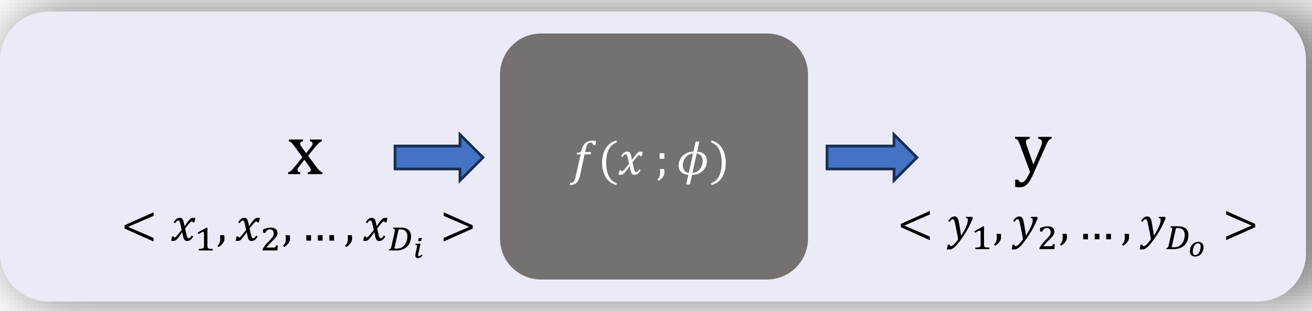
[Image from J.D Prince 2023 DL Book]

SCAI

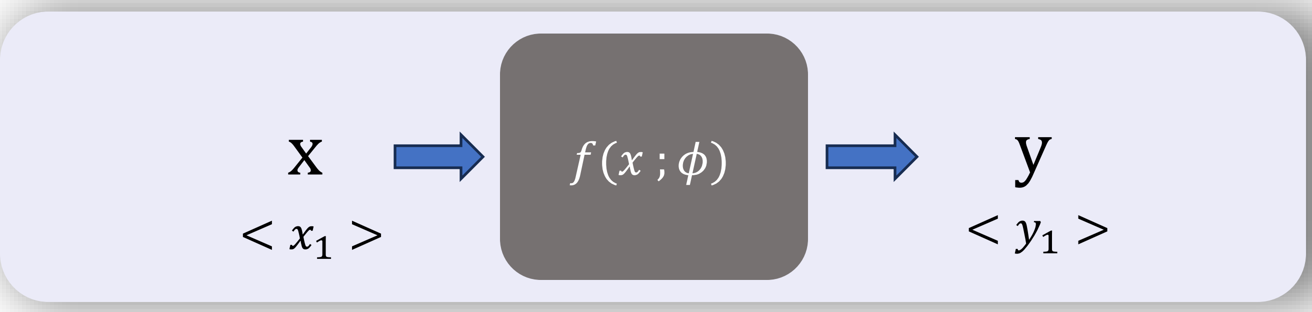
Universal Approximation Theorem

"a formal proof that, with enough hidden units, a shallow neural network can describe any continuous function"

Neural Networks



Di-input/Do-output

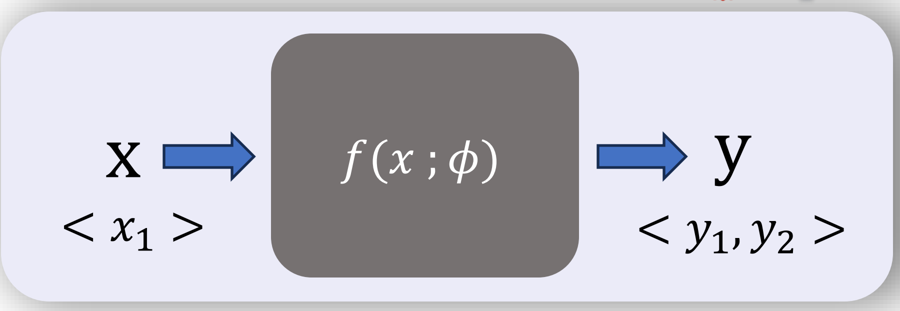


1-input/ 1-output

Neural Networks

❑ Two Outputs

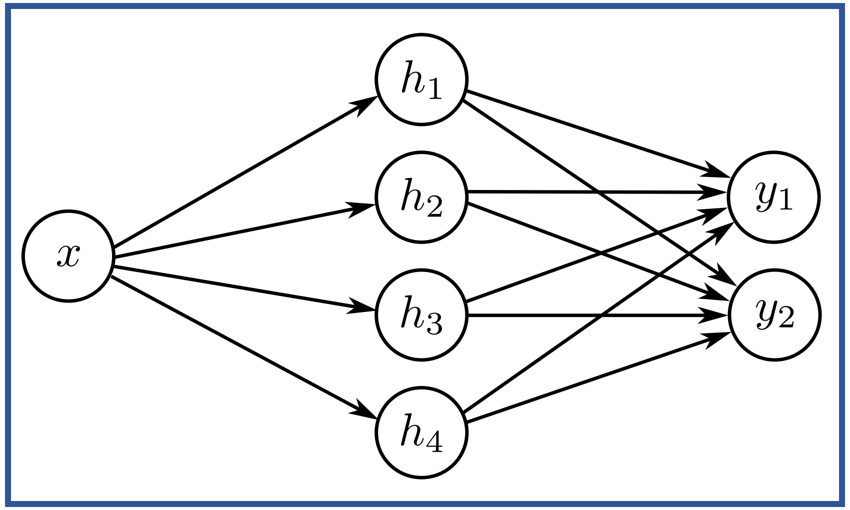
- 1 input, 4 hidden units, 2 outputs

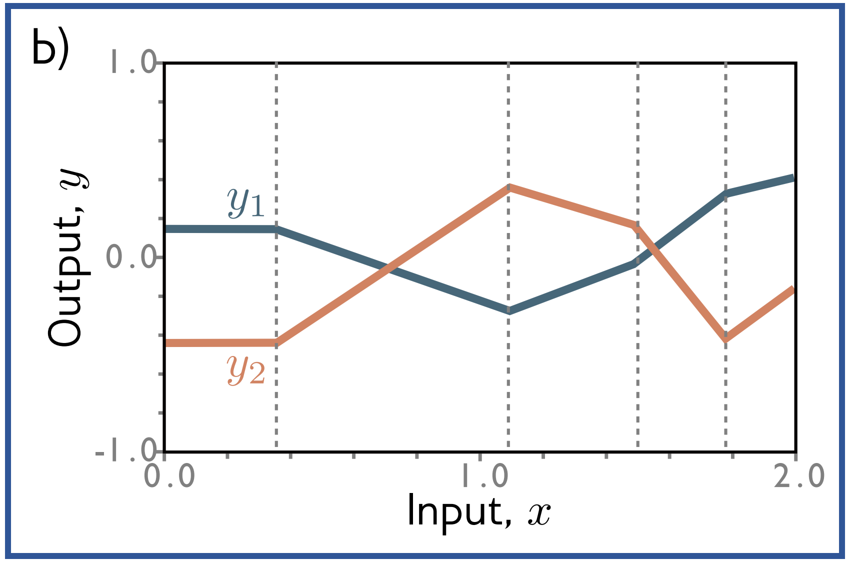


1-input/ 2-output

$$h_1 = \mathrm{a}[\theta_{10} + \theta_{11}x]$$
$$h_2 = \mathrm{a}[\theta_{20} + \theta_{21}x]$$
$$h_3 = \mathrm{a}[\theta_{30} + \theta_{31}x]$$
$$h_4 = \mathrm{a}[\theta_{40} + \theta_{41}x]$$

$$y_1 = \phi_{10} + \phi_{11}h_1 + \phi_{12}h_2 + \phi_{13}h_3 + \phi_{14}h_4$$
$$y_2 = \phi_{20} + \phi_{21}h_1 + \phi_{22}h_2 + \phi_{23}h_3 + \phi_{24}h_4$$



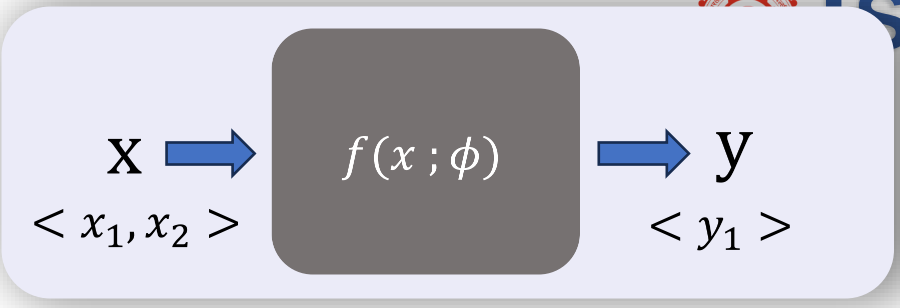


[Image from J.D Prince 2023 DL Book]

Neural Networks

❑ Two Inputs

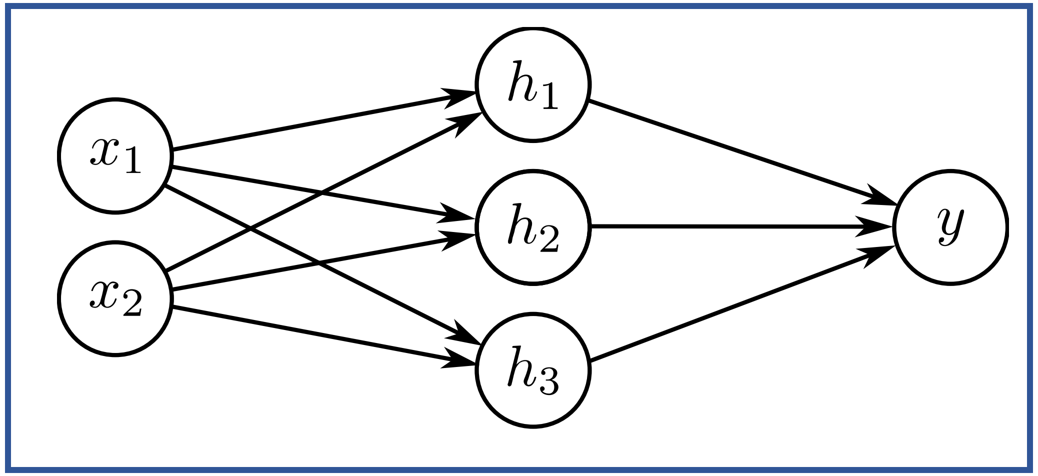
- 2 inputs, 3 hidden units, 1 outputs

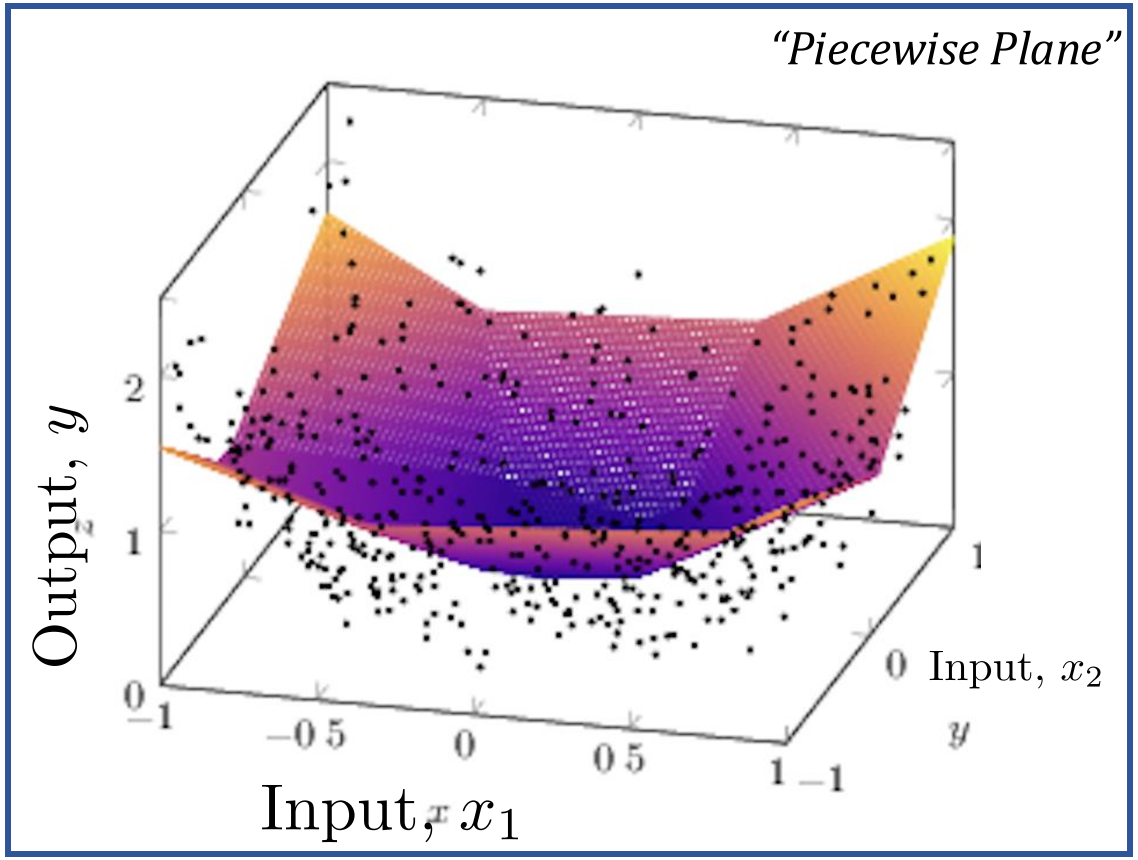


2-input/ 1-output

$$h_1 = \mathrm{a}[\theta_{10} + \theta_{11}x_1 + \theta_{12}x_2]$$
$$h_2 = \mathrm{a}[\theta_{20} + \theta_{21}x_1 + \theta_{22}x_2]$$
$$h_3 = \mathrm{a}[\theta_{30} + \theta_{31}x_1 + \theta_{32}x_2]$$

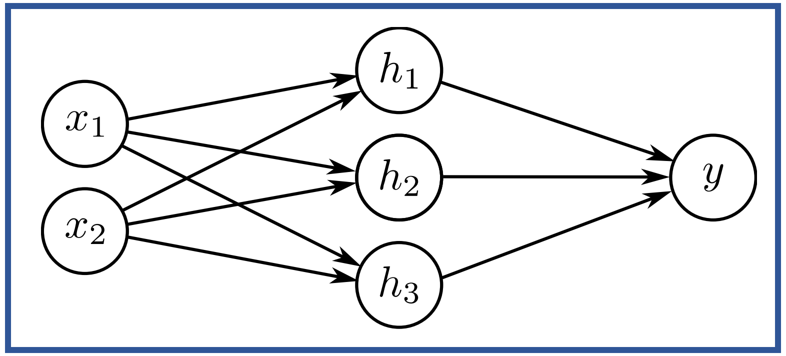
$$y = \phi_0 + \phi_1 h_1 + \phi_2 h_2 + \phi_3 h_3$$

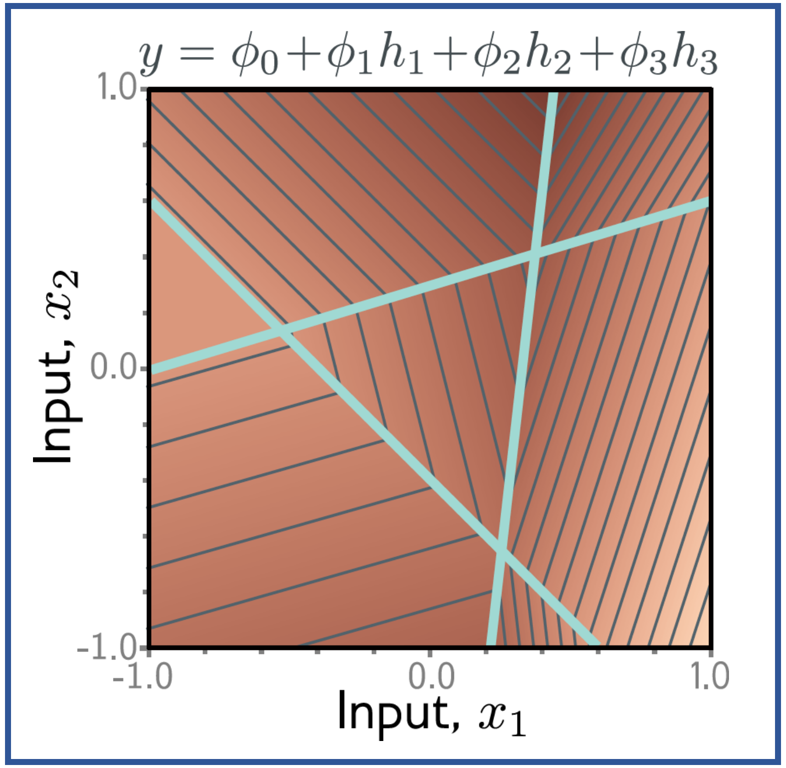


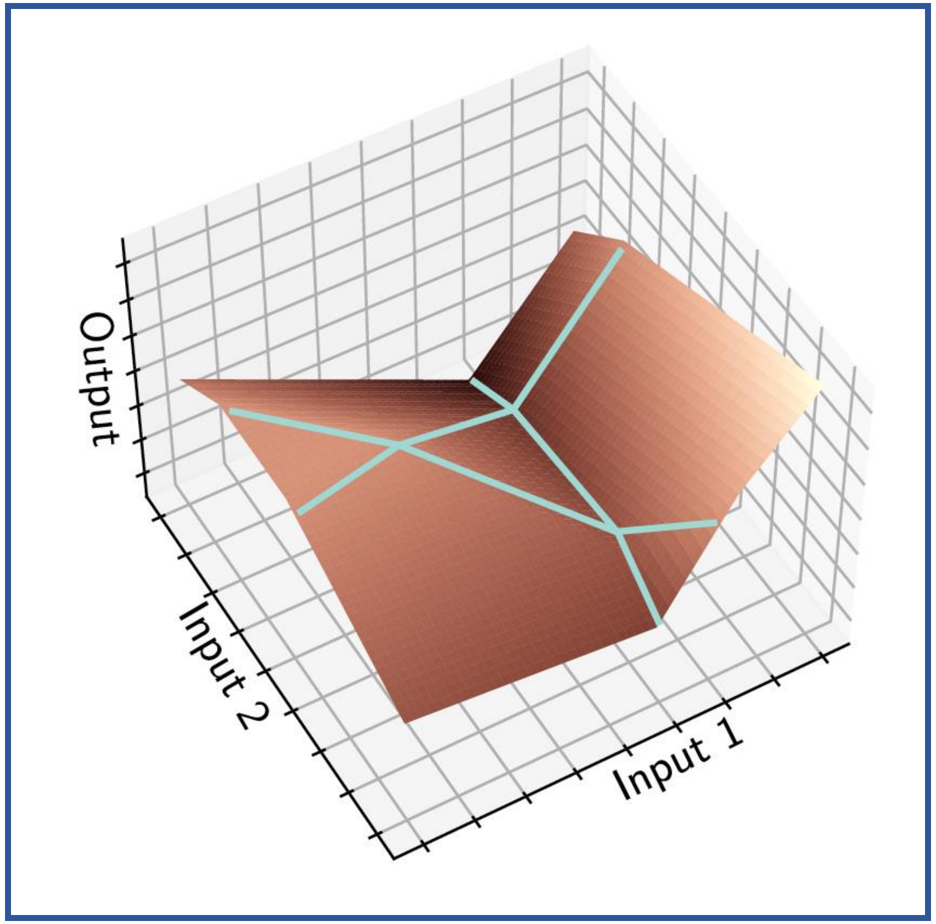


[Image from J.D Prince 2023 DL Book]

Neural Network

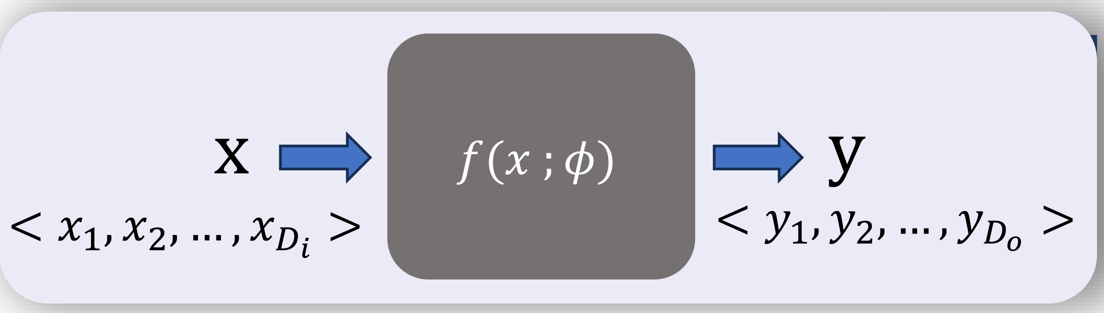






[Image from J.D Prince 2023 DL Book]

Neural Networks



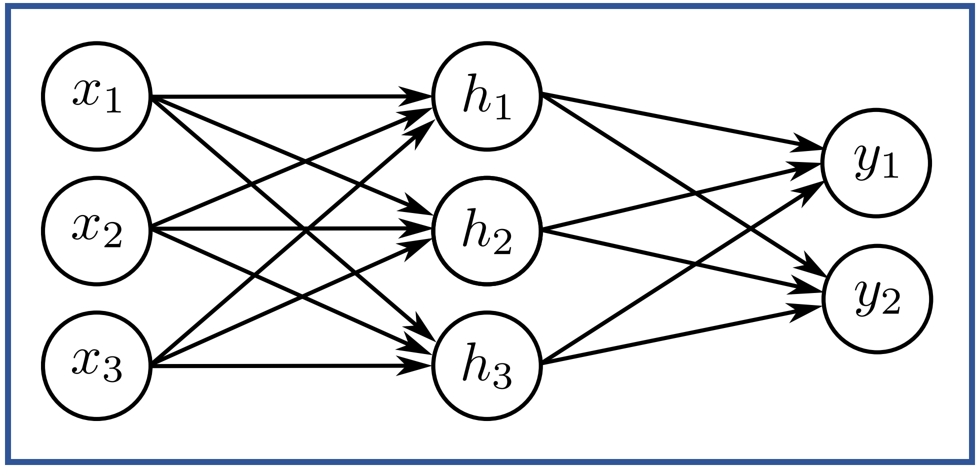
Di-input/Do-output

- Di-Inputs, D hidden units, Do-Outputs:

$$h_d = \mathrm{a}\left[\theta_{d0} + \sum_{i=1}^{D_i} \theta_{di} x_i\right]$$

$$y_j = \phi_{j0} + \sum_{d=1}^{D} \phi_{jd} h_d$$

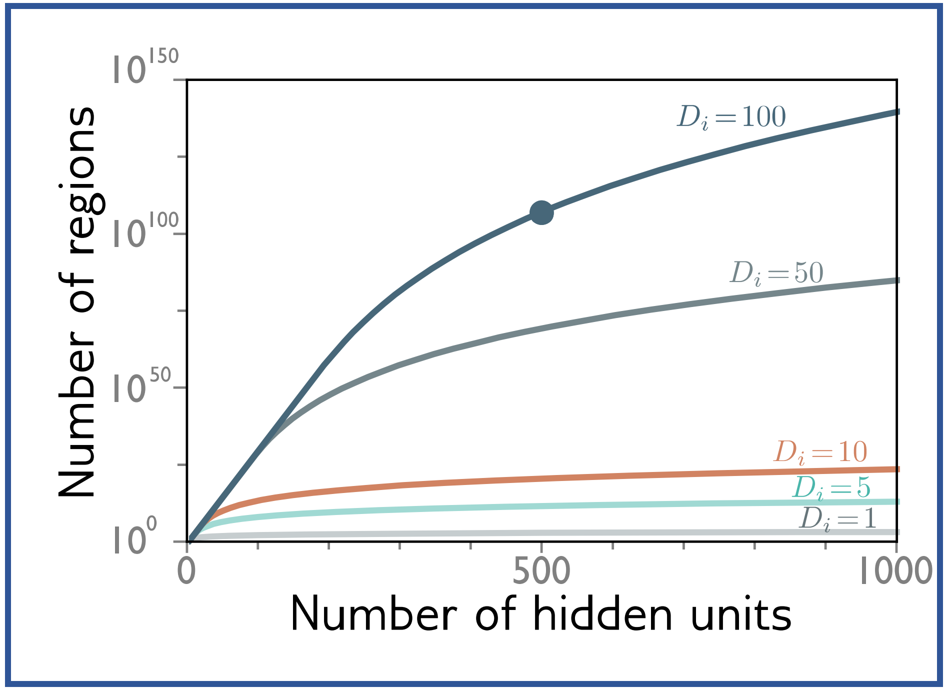
- e.g., Three inputs, three hidden units, two outputs

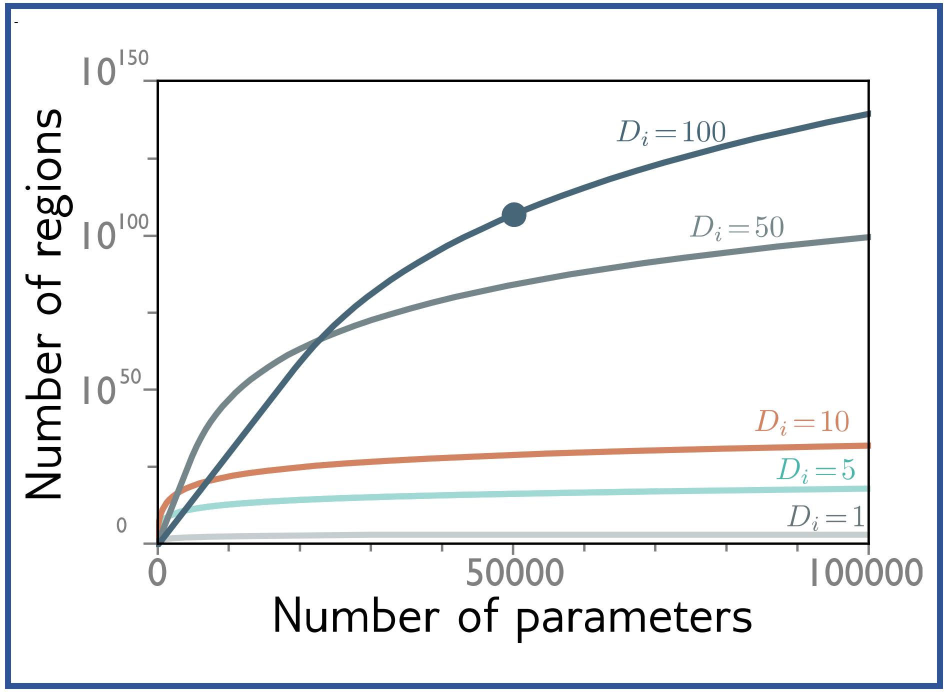


[Image from J.D Prince 2023 DL Book]

Neural Networks

- #output regions vs #hidden units vs #parameters:





[Image from J.D Prince 2023 DL Book]

Neural Networks

❑ Nomenclature:

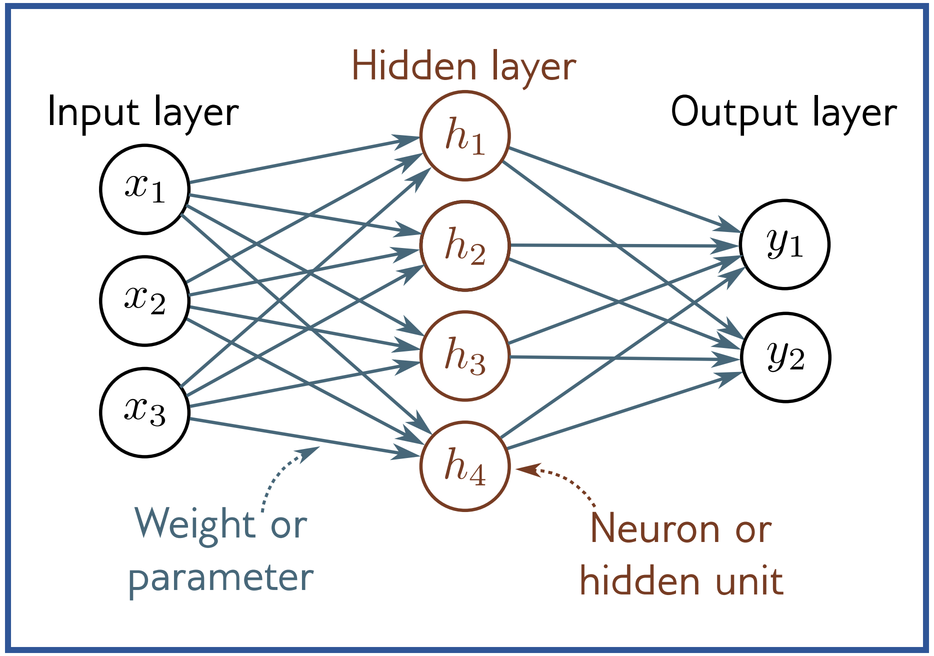
What is this neural network called?

1. Single-layered neural network.

2. Two-layered neural network.

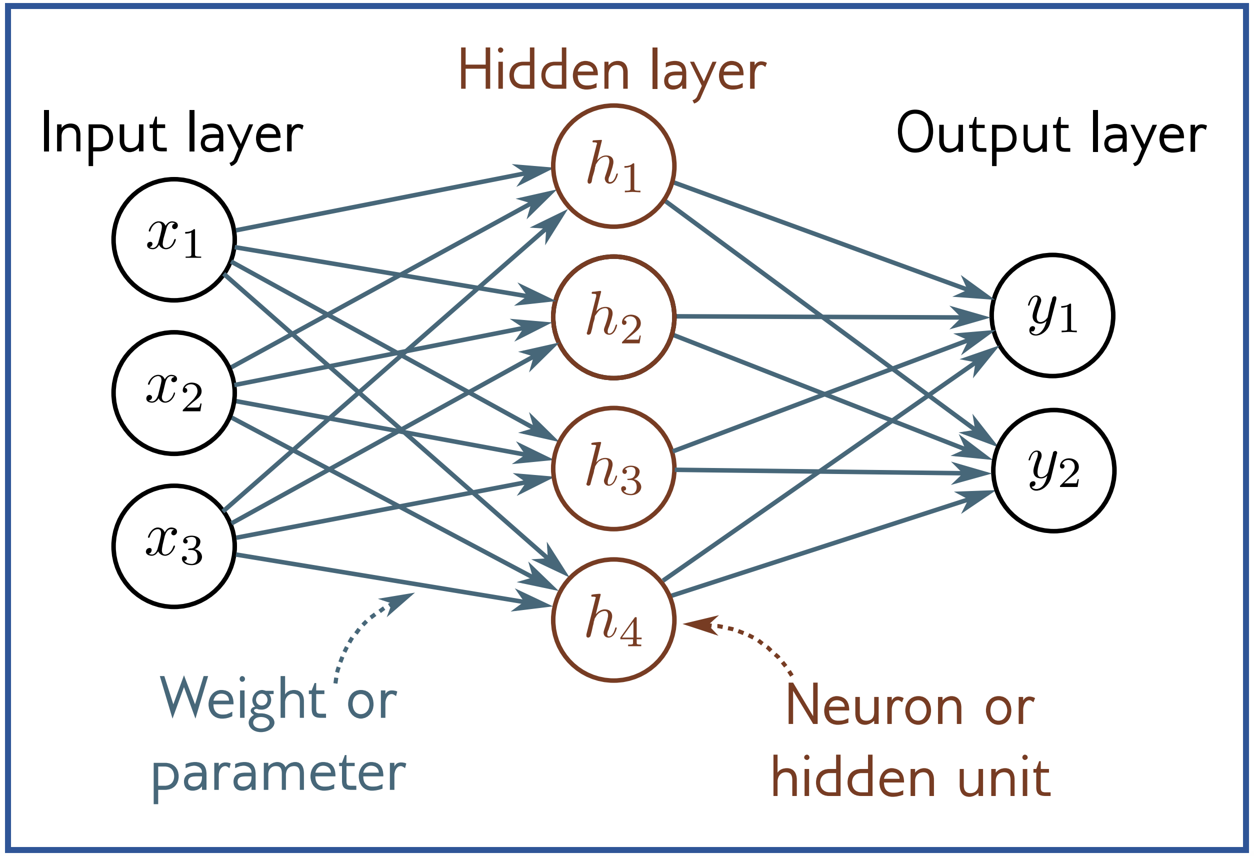
Two layers of learnable parameters (as per the Bishop 2024 DL Book)

3. Three-layered neural network.



[Image from J.D Prince 2023 DL Book]

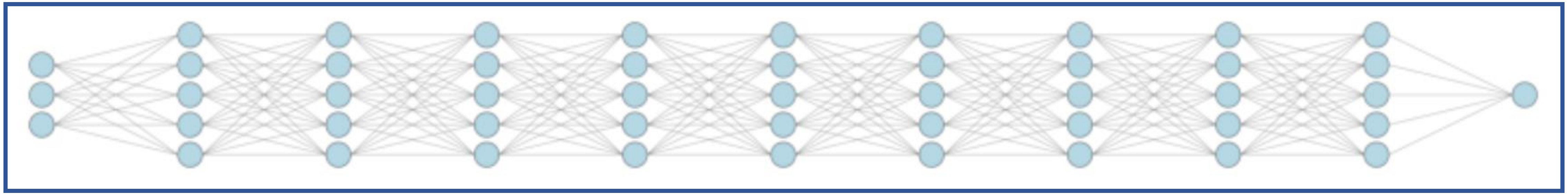
Shallow Neural Networks

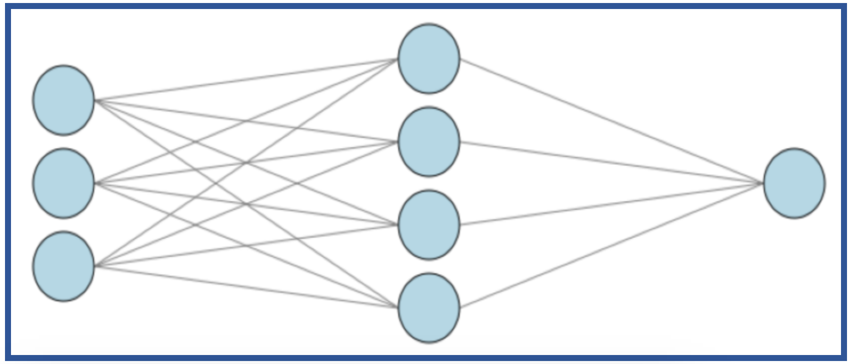


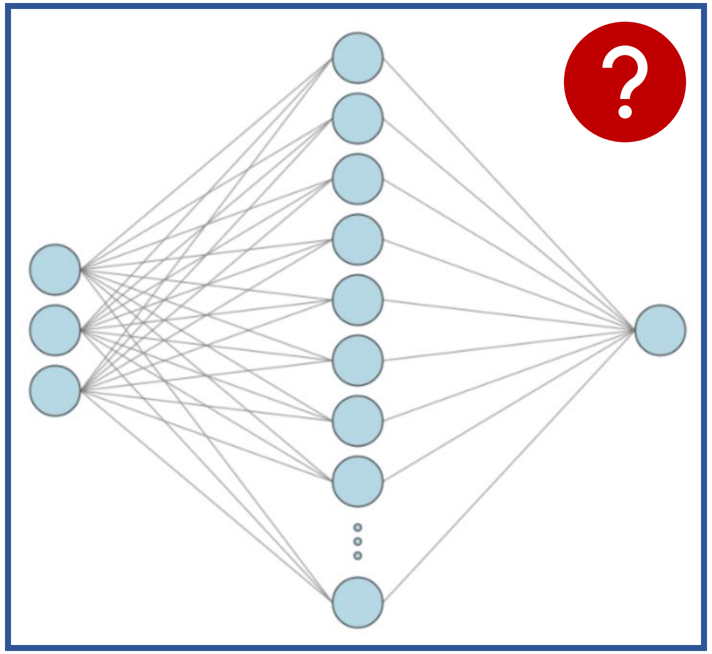
What happens if we add more layers?

[Image from J.D Prince 2023 DL Book]

Deep Neural Network



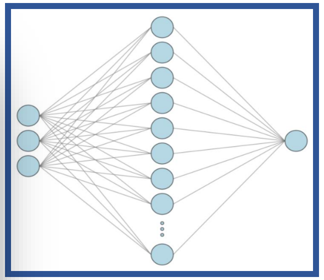




What’s different?

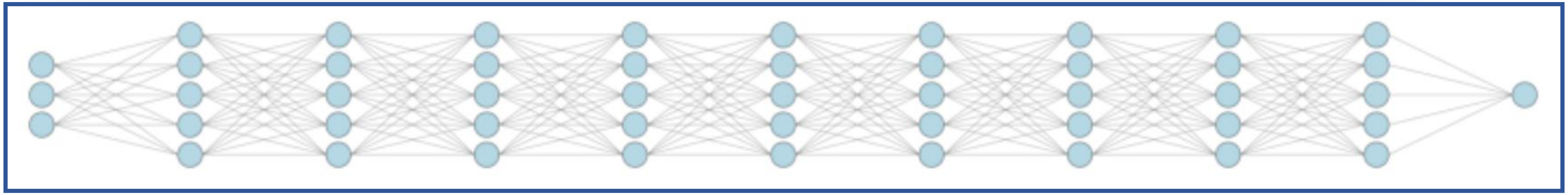
Number of parameters

Deep Neural Network



- A network with **two layers of learnable parameters - universal approximation** capabilities.

- A network with **more than two layers** – can represent a given function with **far fewer parameters**.



Paper: On the Number of Linear Regions of Deep Neural Networks. Mont´ufar et al. NeurIPS-2014