

# FeedForward Neural Networks

Course 3, Module 3, Lesson 1



UNIVERSITY OF TORONTO  
FACULTY OF APPLIED SCIENCE & ENGINEERING

# Learning Objectives

- Learn the mode of action of feedforward neural networks
- Learn the mathematical equations of the **hidden layers**, a building block that makes neural networks special among machine learning models

# Feedforward Neural Networks

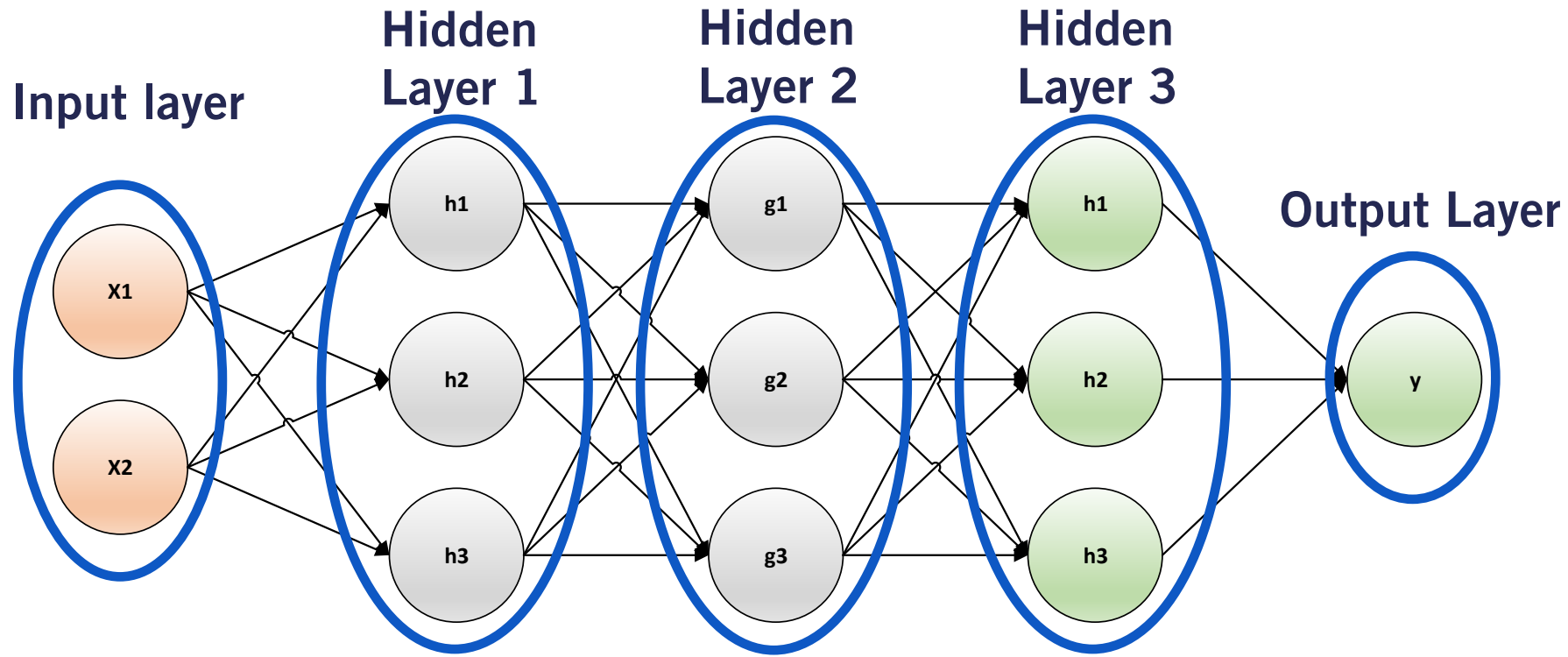
- A **Feedforward Neural Network** defines a **mapping** from **input**  $x$  to **output**  $y$  as:

$$y = f(x; \theta) \quad \text{learned parameters}$$

- An **N layer FNN** is represented as the function composition:

$$f(x; \theta) = f^{(N)} \left( f^{(N-1)} \left( \dots f^{(2)} \left( f^{(1)}(x) \right) \right) \right)$$

# Feedforward Neural Networks: Example



$$f^{(1)}(x) \quad f^{(2)} f^{(1)}(x) \quad f^{(3)} f^{(2)} f^{(1)}(x) \quad f^{(4)} f^{(3)} f^{(2)} f^{(1)}(x) \quad f^{(5)} f^{(4)} f^{(3)} f^{(2)} f^{(1)}(x)$$



# Feedforward Neural Networks

$$f(x; \theta) = f^{(N)} \left( f^{(N-1)} \left( \dots f^{(2)} \left( f^{(1)}(x) \right) \right) \right)$$

- $x$  is called the **input layer**
- The final function  $f^{(N)}$  is called the **output layer**
- The functions  $f^{(1)}$  to  $f^{(N-1)}$  are called the **hidden layers**

# Feedforward Neural Networks

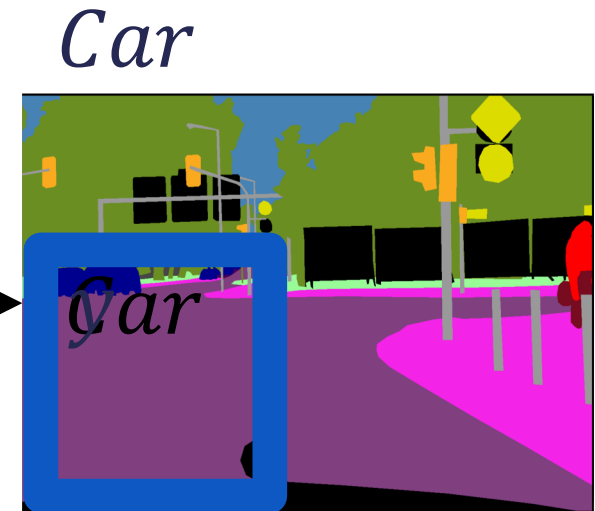
*hard to model processes*

- Functions to estimate:

- **Object Classification:** Image  $\rightarrow$  Label
- **Object Detection:** Image  $\rightarrow$  Label + Location
- **Depth Estimation:** Image  $\rightarrow$  Depth for every pixel
- **Semantic Segmentation:** Image  $\rightarrow$  Label for every pixel



$$f(x; \theta)$$

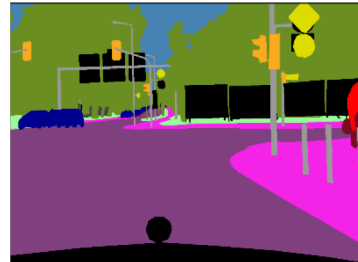


# Mode Of Action Of Neural Networks

- **Training:** Give neural network **examples of  $f^*(x)$** . for a wide variation of the input  $x$ . Then, optimize its parameters  $\theta$  to force  $f(x; \theta) \cong f^*(x)$
- Pairs of  $x$  and  $f^*(x)$  are called **training data**



$x$



$f^*(x)$

- Only **output** is specified by training data! Network is free to do anything with its **hidden layers**

# Hidden Units

$$h_n = g(W^T h_{n-1} + b)$$

*an affine transformation*

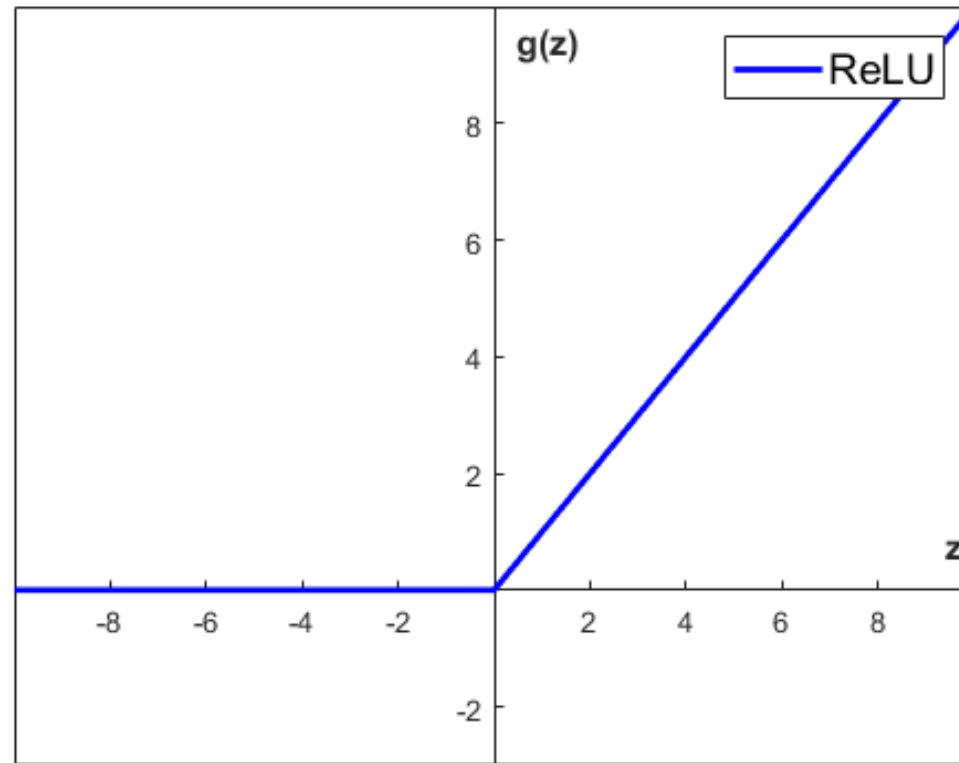
- Activation function  $g$
- Input  $h_{n-1}$
- Weight matrix  $W$
- Bias  $b$
- Parameters  $\theta$  are the weights and biases of all the layers of the network
- Transformed parameters passed through activation function  $g$  *nonlinear function*



# The Rectified Linear Unit: ReLU

- The **ReLU** hidden unit is currently the **default** choice of activation function for Feedforward Neural Networks

$$g(z) = \max(0, z)$$



# Example: Hidden Layer With RELU Activation Function

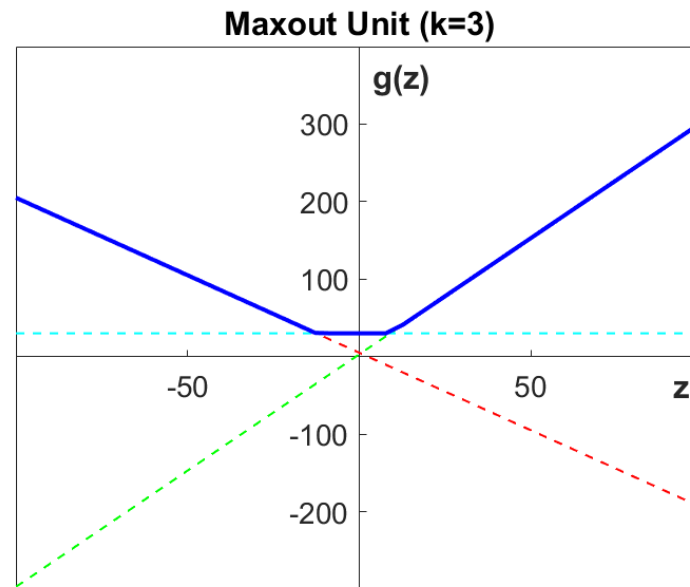
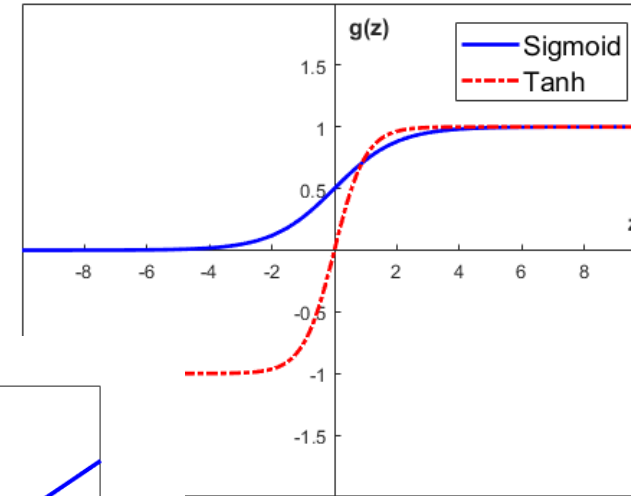
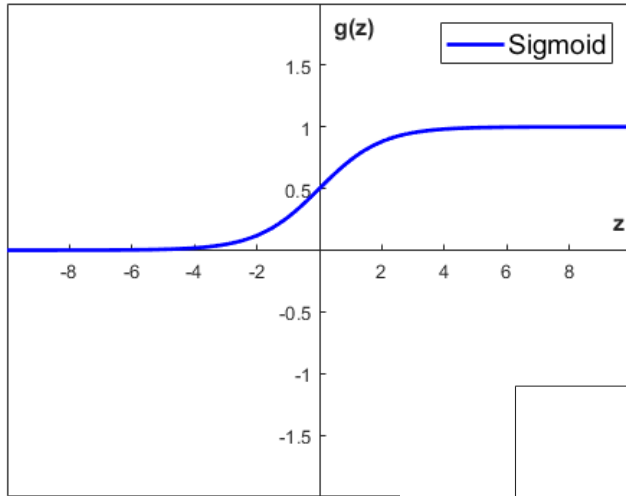
$$h_{n-1} = \underbrace{\begin{bmatrix} -17 & 12 & -6 \\ -14 & 2 & 7 \end{bmatrix}}_{2 \times 3}, \quad W = \underbrace{\begin{bmatrix} 9 & -2 & 9 & 10 & -3 \\ -3 & 10 & 4 & 6 & 3 \end{bmatrix}}_{2 \times 5}, \quad b = \begin{bmatrix} -2 & -18 & -12 \\ -6 & -3 & 2 \\ -7 & 4 & -9 \\ -6 & -12 & -10 \\ -8 & 4 & -15 \end{bmatrix}$$

Evaluate affine transformation  $W^T h_{n-1} + b$ :

$$\underbrace{\begin{bmatrix} 9 & -3 \\ -2 & 10 \\ 9 & 4 \\ 10 & 6 \\ -3 & 3 \end{bmatrix}}_{5 \times 2} \begin{bmatrix} -17 & 12 & -6 \\ -14 & 2 & 7 \end{bmatrix} + \begin{bmatrix} -2 & -18 & -12 \\ -6 & -3 & 2 \\ -7 & 4 & -9 \\ -6 & -12 & -10 \\ -8 & 4 & -15 \end{bmatrix} = \begin{bmatrix} -113 & 84 & -87 \\ -112 & -7 & 84 \\ -216 & 120 & -35 \\ -260 & 120 & -28 \\ 1 & -26 & 24 \end{bmatrix}$$

$$h_n = \max(0, W^T h_{n-1} + b) = \begin{bmatrix} 0 & 84 & 0 \\ 0 & 0 & 84 \\ 0 & 120 & 0 \\ 0 & 120 & 0 \\ 1 & 0 & 24 \end{bmatrix}$$

# Activation Functions



# Summary

- Feedforward neural networks can be used for a variety of perception tasks related to self-driving cars
- Feedforward neural networks rely on **hidden layers** to implement a good approximation of a target function
- Various activation functions are available to use within hidden layers. **RELU** is the default baseline
- **Next: Output Layers and Loss Functions**