FeedForward Neural Networks

Course 3, Module 3, Lesson 1

UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE & ENGINEERING

Feedforward Neural Networks

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

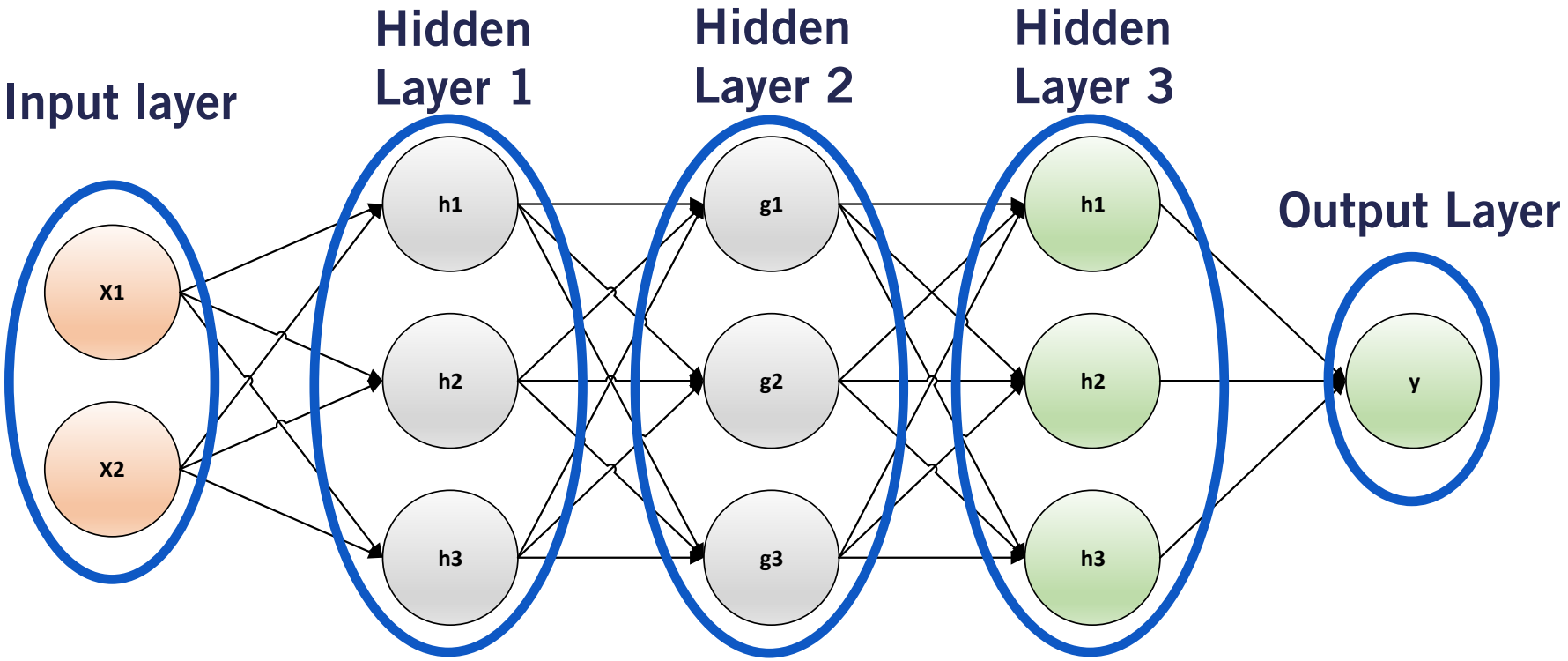
$$y = f(x; \theta)$$

learned parameters

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

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

Feedforward Neural Networks: Example



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

Feedforward Neural Networks

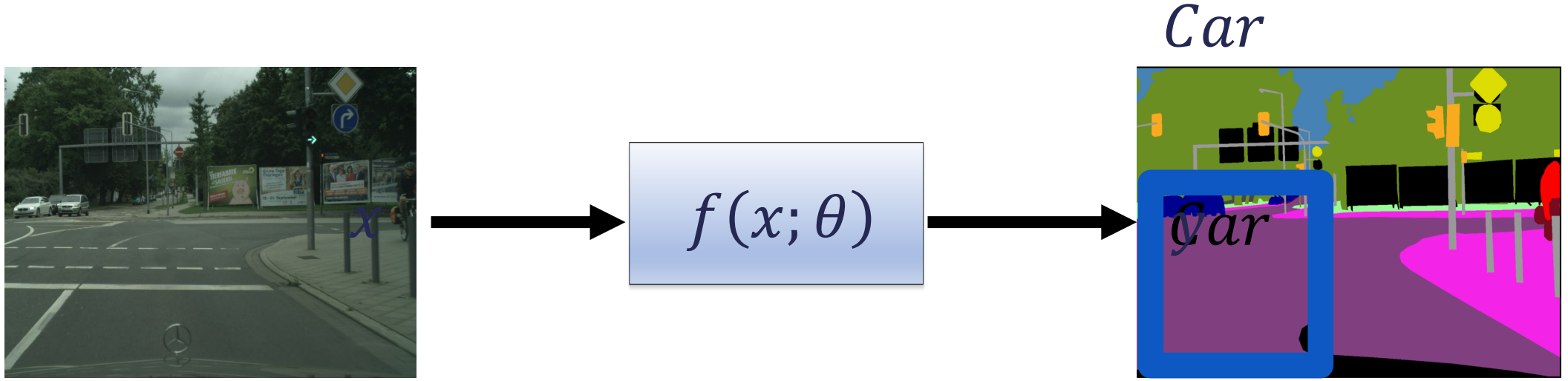
$$f(x;\theta) = f^{(N)}\left(f^{(N-1)}\left(\dots f^{(2)}\left(f^{(1)}(x)\right)\right)\right)$$

- $\boldsymbol{x}$ 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 process

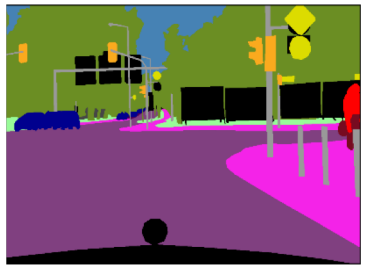
- Functions to estimate:
  - **Object Classification**: Image → Label
  - **Object Detection**: Image → Label + Location
  - **Depth Estimation**: Image → Depth for every pixel
  - **Semantic Segmentation**: Image → Label for every pixel



Mode Of Action Of Neural Networks

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

$\boldsymbol{x}$



$f^*(\boldsymbol{x})$

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

Hidden Units

an affine transformation

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

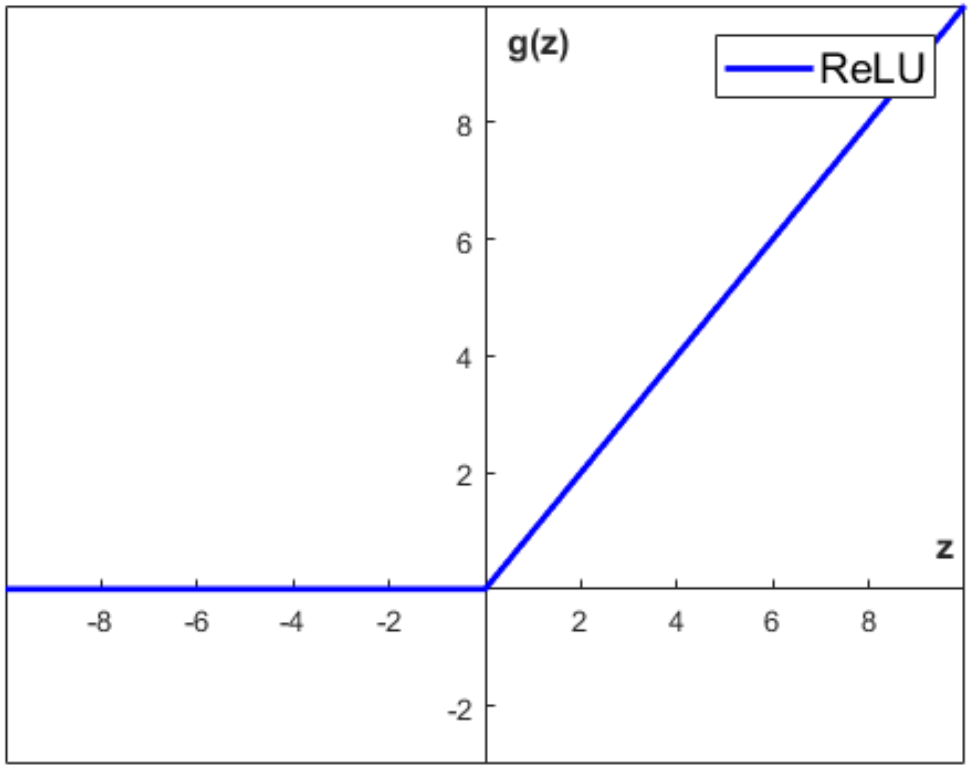
- Activation function g
- Input h_{n-1}
- Weight matrix $\boldsymbol{W}$
- Bias $\boldsymbol{b}$
- Parameters $\boldsymbol{\theta}$ are the weights and biases of all the layers of the network
- Transformed parameters passed through activation function g

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

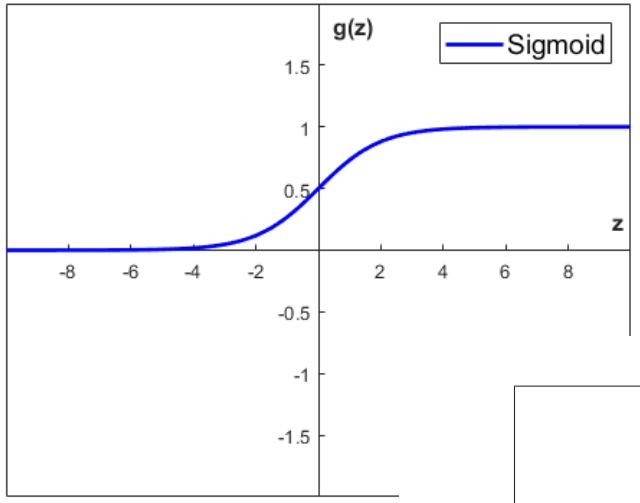
$$h_{n-1} = \underbrace{\begin{bmatrix} -17 & 12 & -6 \\ -14 & 2 & 7 \end{bmatrix}}_{2\times 3}, \quad W = \underbrace{\begin{bmatrix} 9 & -2 & 9 & 10 & -3 \\ -3 & 10 & 4 & 6 & 3 \end{bmatrix}}_{2\times 5}, \quad b = \begin{bmatrix} -2 & -18 & -12 \\ -6 & -3 & 2 \\ -7 & 4 & -9 \\ -6 & -12 & -10 \\ -8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

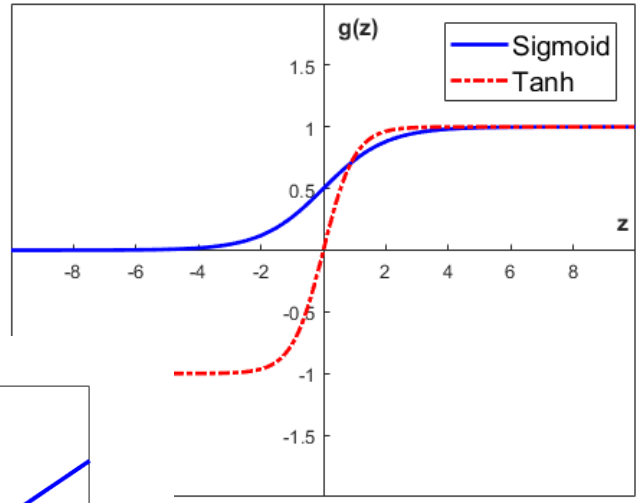
Evaluate affine transformation $W^T h_{n-1} + b$:

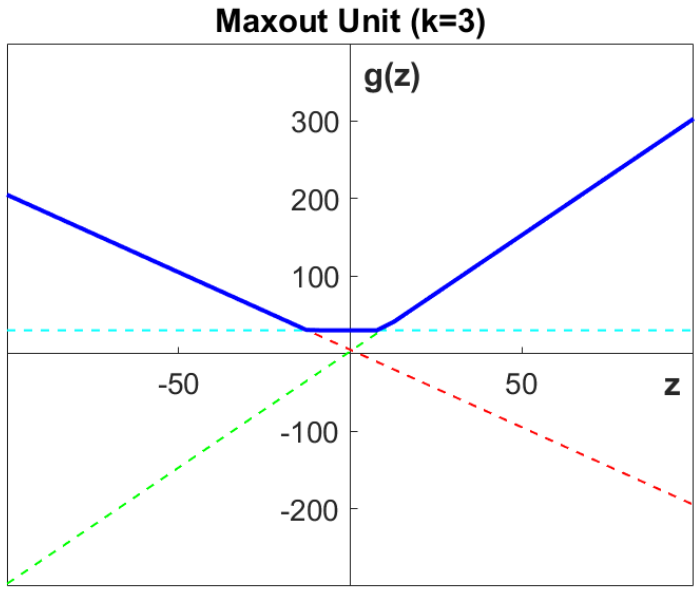
$$\begin{bmatrix} 9 & -3 \\ -2 & 10 \\ 9 & 4 \\ 10 & 6 \\ -3 & 3 \end{bmatrix} \begin{bmatrix} -17 & 12 & -6 \\ -14 & 2 & 7 \end{bmatrix} + \begin{bmatrix} -2 & -18 & -12 \\ -6 & -3 & 2 \\ -7 & 4 & -9 \\ -6 & -12 & -10 \\ -8 & 4 & -15 \end{bmatrix} = \overbrace{\begin{bmatrix} -113 & 84 & -87 \\ -112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

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

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**