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

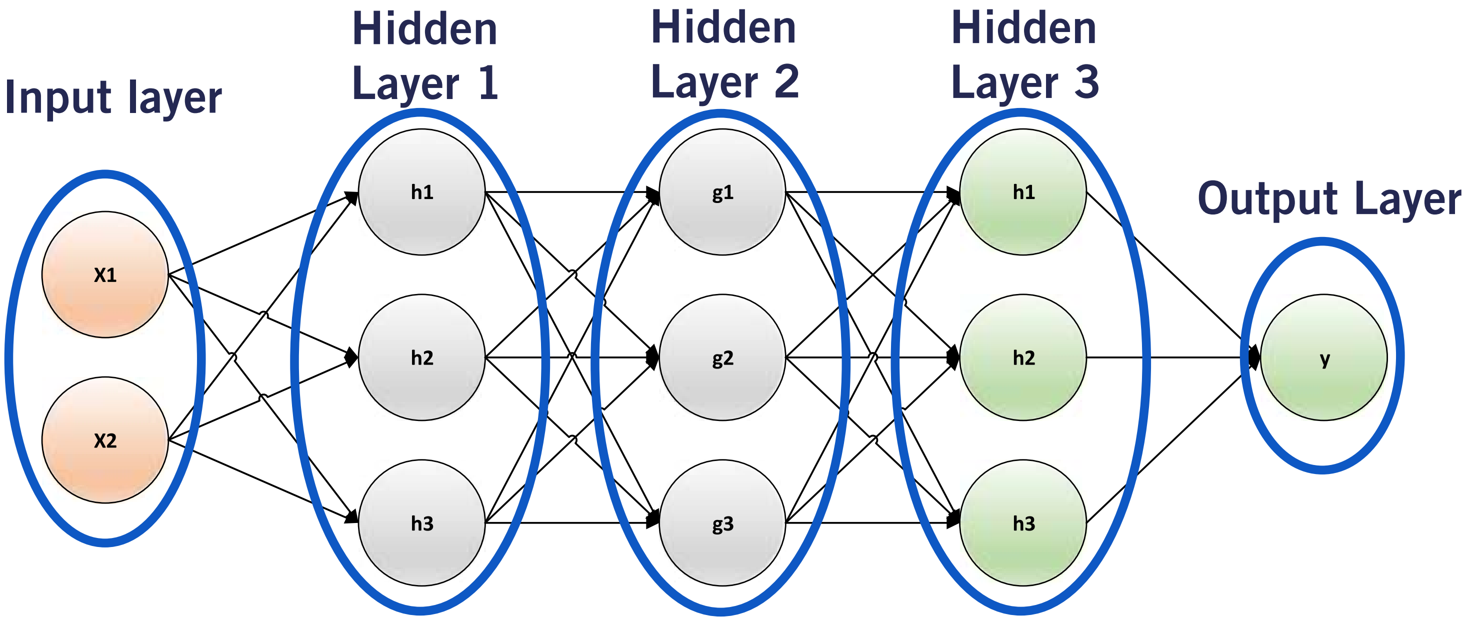
- A **Feedforward Neural Network** defines a **mapping** from **input** $\boldsymbol{x}$ to **output** $\boldsymbol{y}$ as:

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

- 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



Input layer

Hidden Layer 1

Hidden Layer 2

Hidden Layer 3

Output Layer

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

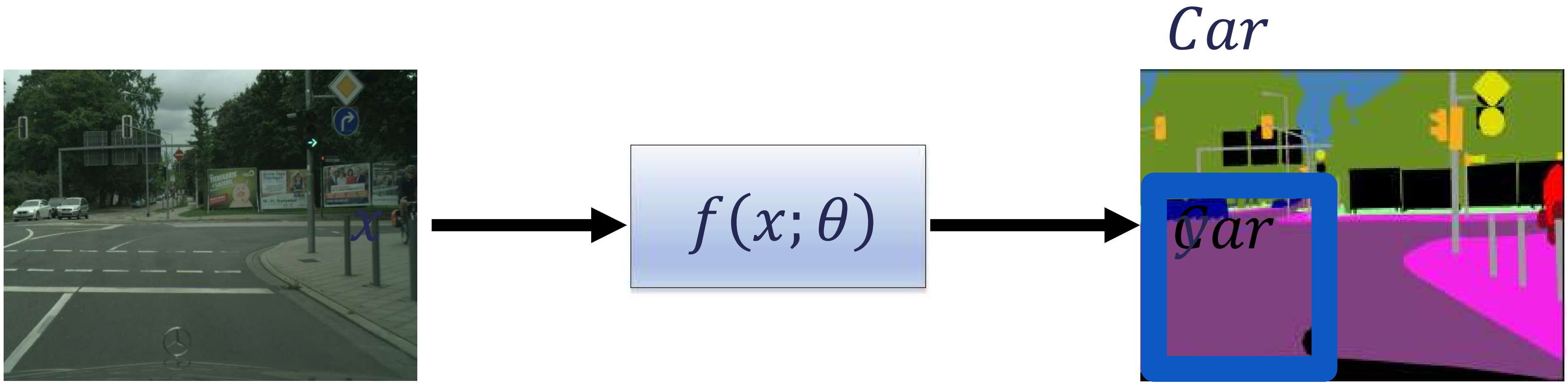
Feedforward Neural Networks

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

- $\boldsymbol{x}$ 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**

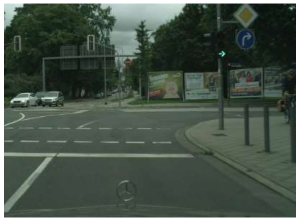
Feedforward Neural Networks

- 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

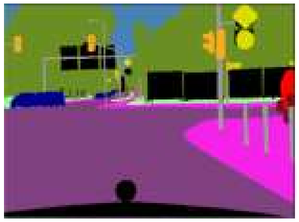


Mode Of Action Of Neural Networks

- **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

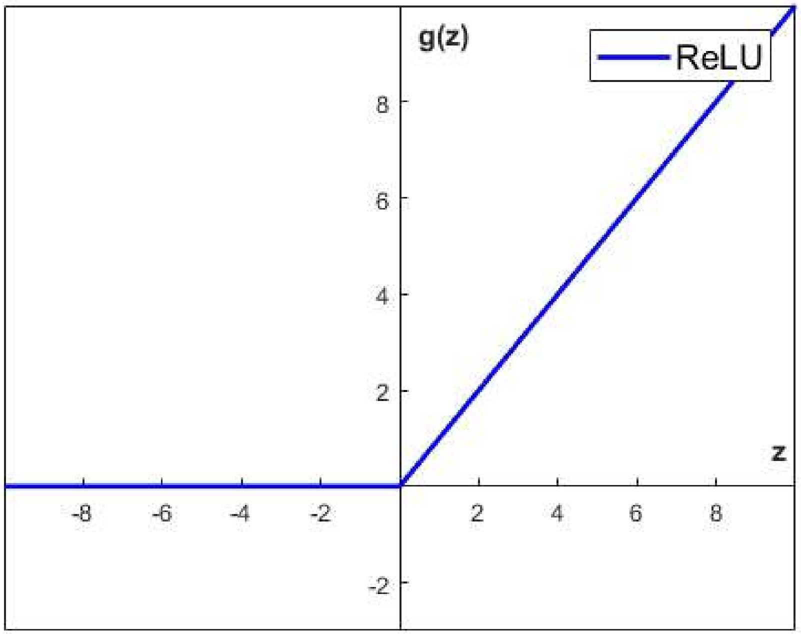
$$h_n = \boxed{g}\left(\boxed{\boxed{W^T}\boxed{h_{n-1}} + \boxed{b}}\right)$$

- 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

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

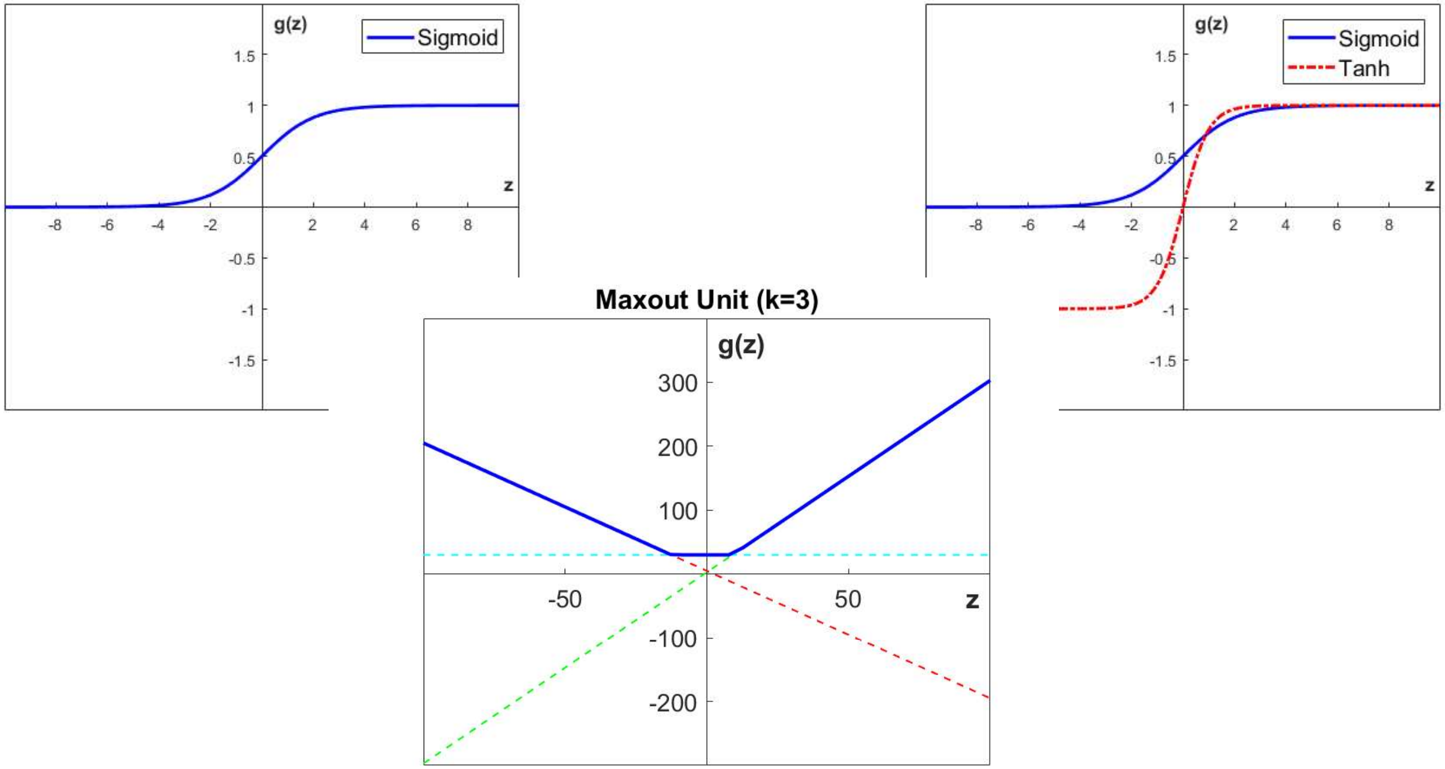
$$h_{n-1} = \underbrace{\begin{bmatrix} -17 & 12 & -6 \\ -14 & 2 & 7 \end{bmatrix}}_{\mathbf{2\times3}}, \qquad W = \underbrace{\begin{bmatrix} 9 & -2 & 9 & 10 & -3 \\ -3 & 10 & 4 & 6 & 3 \end{bmatrix}}_{\mathbf{2\times5}}, \qquad b = \begin{bmatrix} -2 & -18 & -12 \\ -6 & -3 & 2 \\ -7 & 4 & -9 \\ -6 & -12 & -10 \\ -8 & 4 & -15 \end{bmatrix}$$

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

$$\begin{bmatrix} 9 & -3 \\ -2 & 10 \\ 9 & 4 \\ 10 & 6 \\ -3 & 3 \end{bmatrix} \begin{bmatrix} -17 & 12 & -6 \\ -14 & 2 & 7 \end{bmatrix} + \begin{bmatrix} -2 & -18 & -12 \\ -6 & -3 & 2 \\ -7 & 4 & -9 \\ -6 & -12 & -10 \\ -8 & 4 & -15 \end{bmatrix} = \overbrace{\begin{bmatrix} -113 & 84 & -87 \\ -112 & -7 & 84 \\ -216 & 120 & -35 \\ -260 & 120 & -28 \\ 1 & -26 & 24 \end{bmatrix}}^{\mathbf{5\times3}}$$

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

Activation Functions



Maxout Unit (k=3)