

PAT 463/563 (Fall 2026)

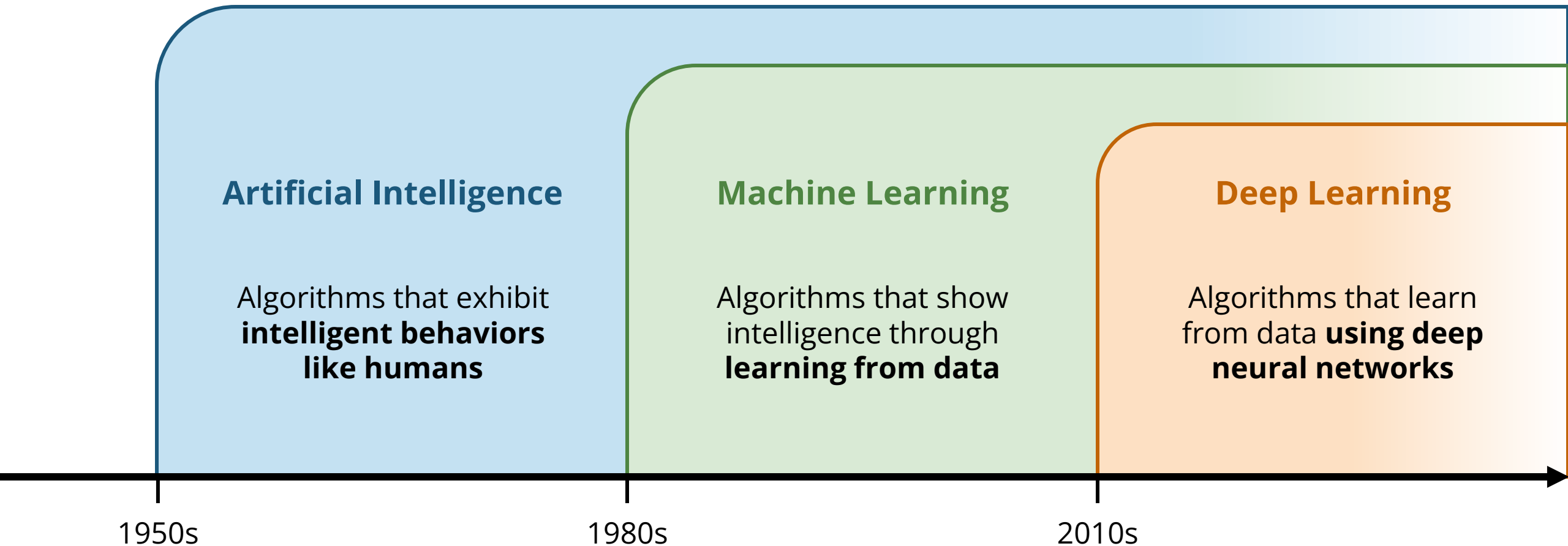
Music & AI

Lecture 6: Deep Learning Fundamentals

Instructor: Hao-Wen (Herman) Dong

What is Deep Learning?

| AI vs. ML vs. DL



Machine Learning

Traditional

Input



Algorithm



Output

Machine learning

Training

Example
inputs

Example
outputs



Machine learning



Model

Inference (test)

Input

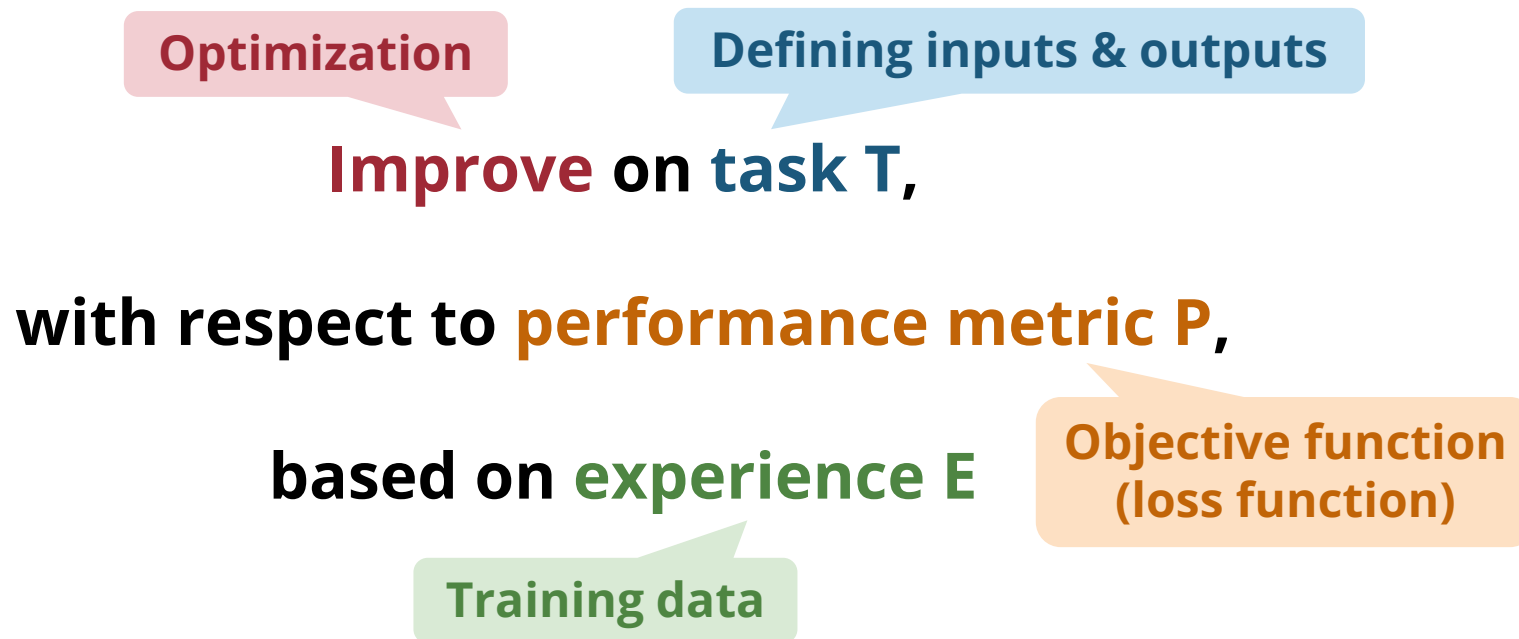


Model



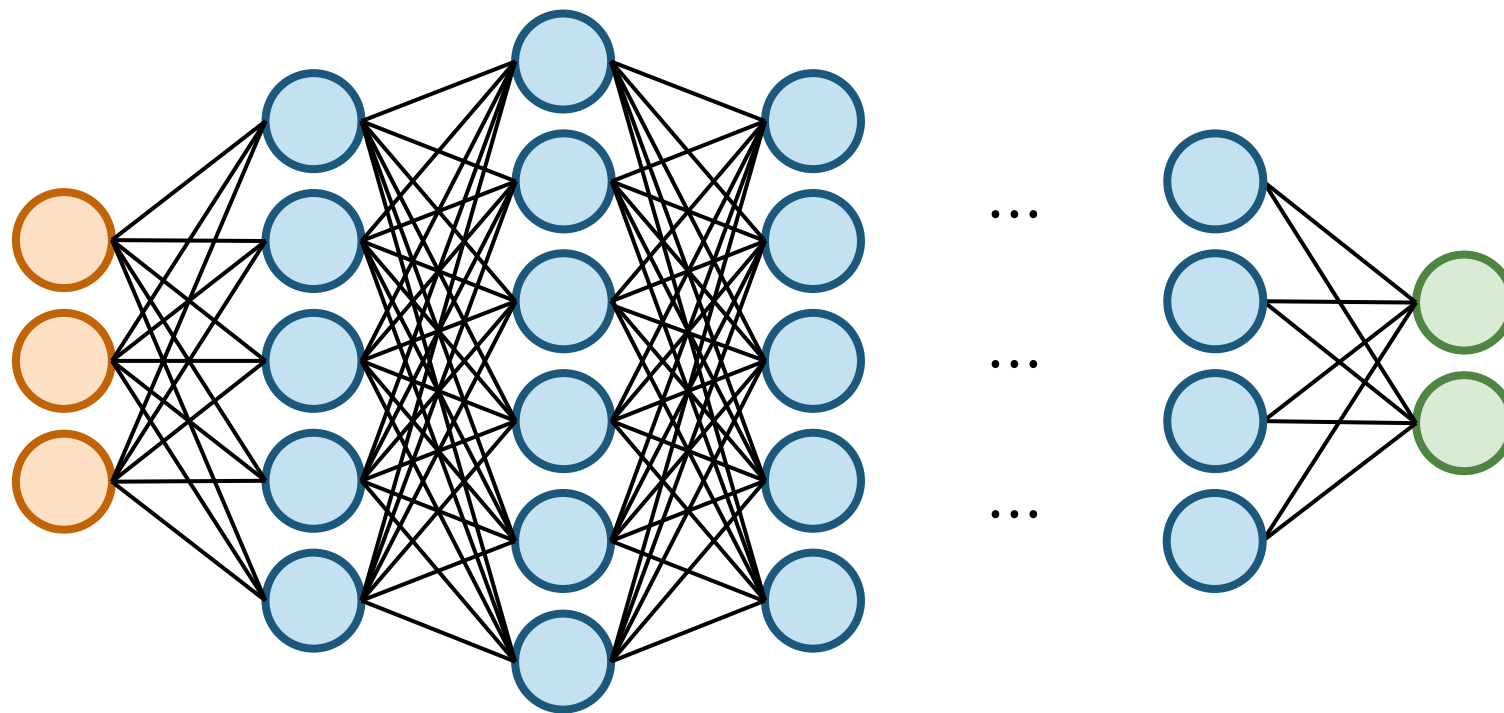
Output

Components of a Machine Learning Model



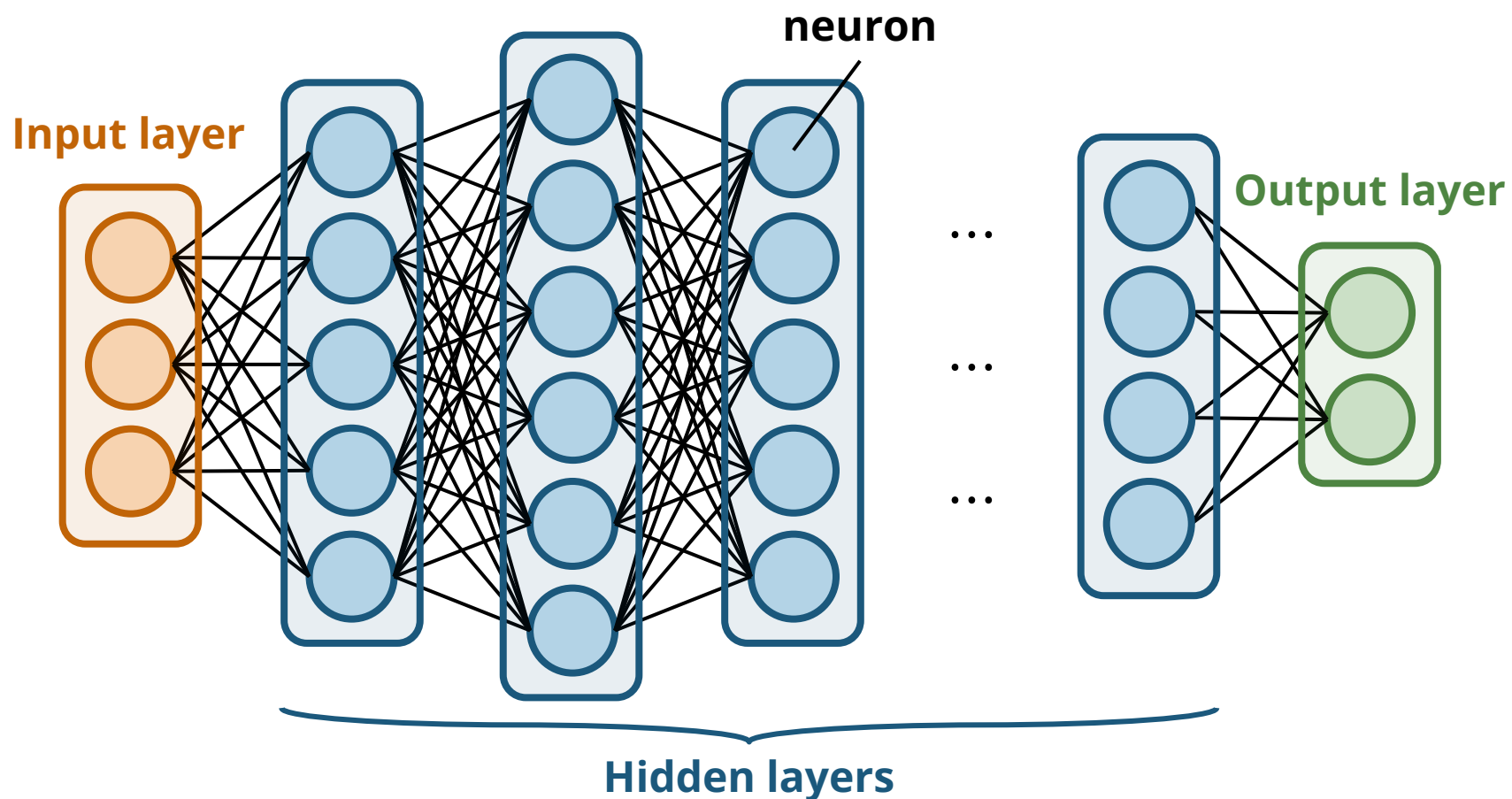
| What is Deep Learning?

- A type of machine learning that uses **deep neural networks**



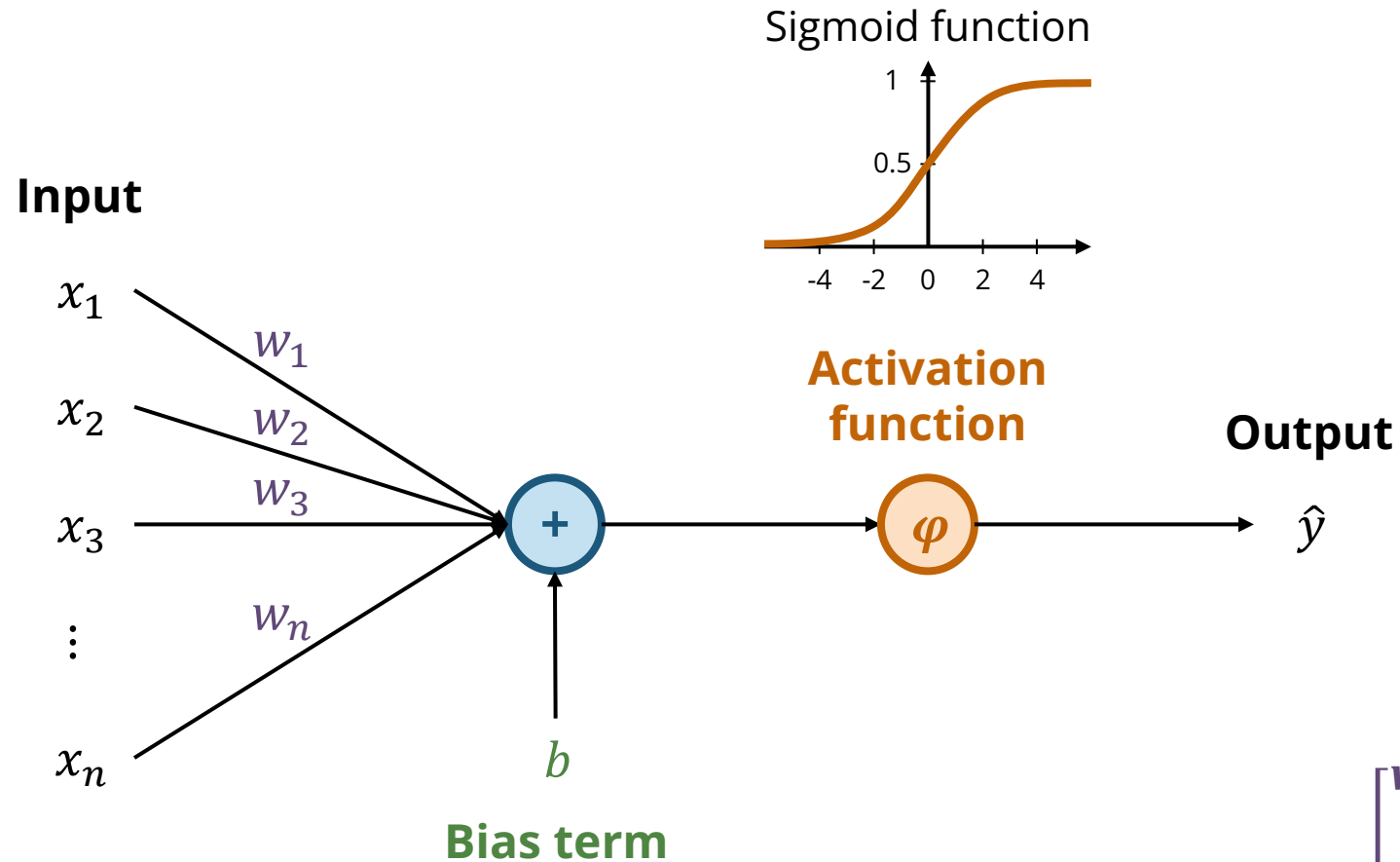
| What is Deep Learning?

- A type of machine learning that uses **deep neural networks**



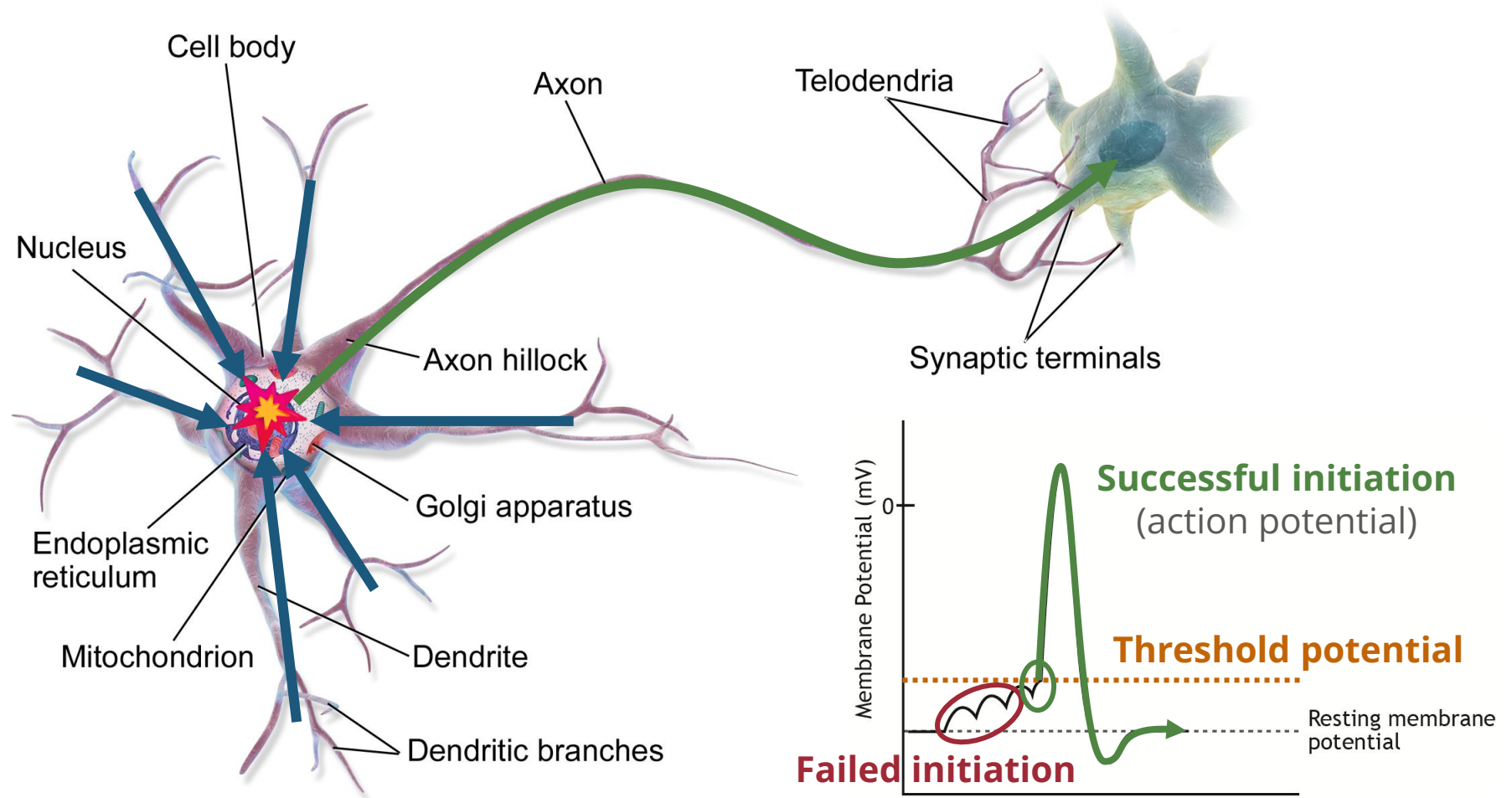
Neural Networks

Inside a Neuron



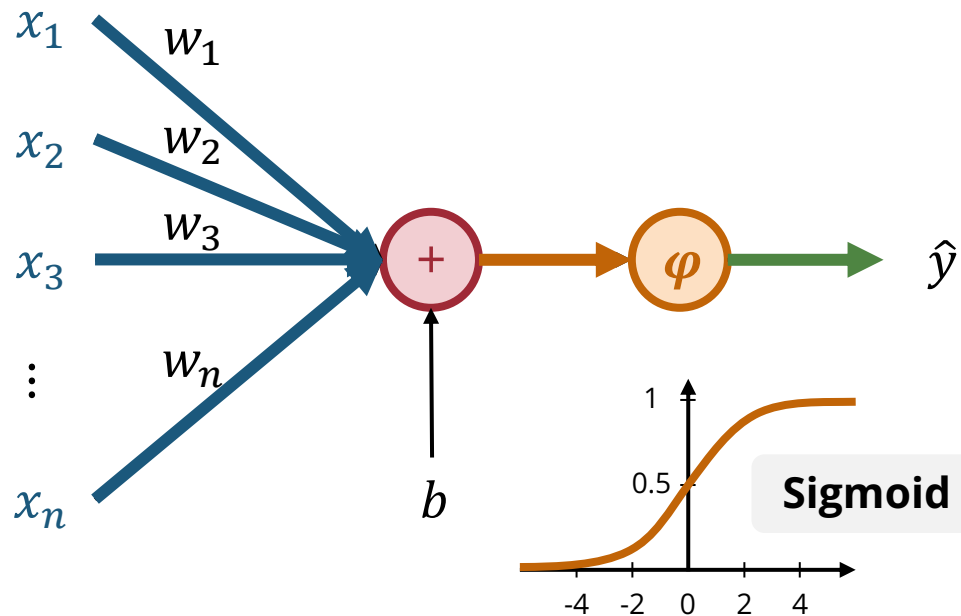
$$\hat{y} = \varphi(w_1x_1 + w_2x_2 + \dots + w_nx_n + b) = \varphi\left(\sum_{i=1}^n w_i x_i + b\right) = \varphi(\mathbf{w} \cdot \mathbf{x} + b)$$

Human Neuron

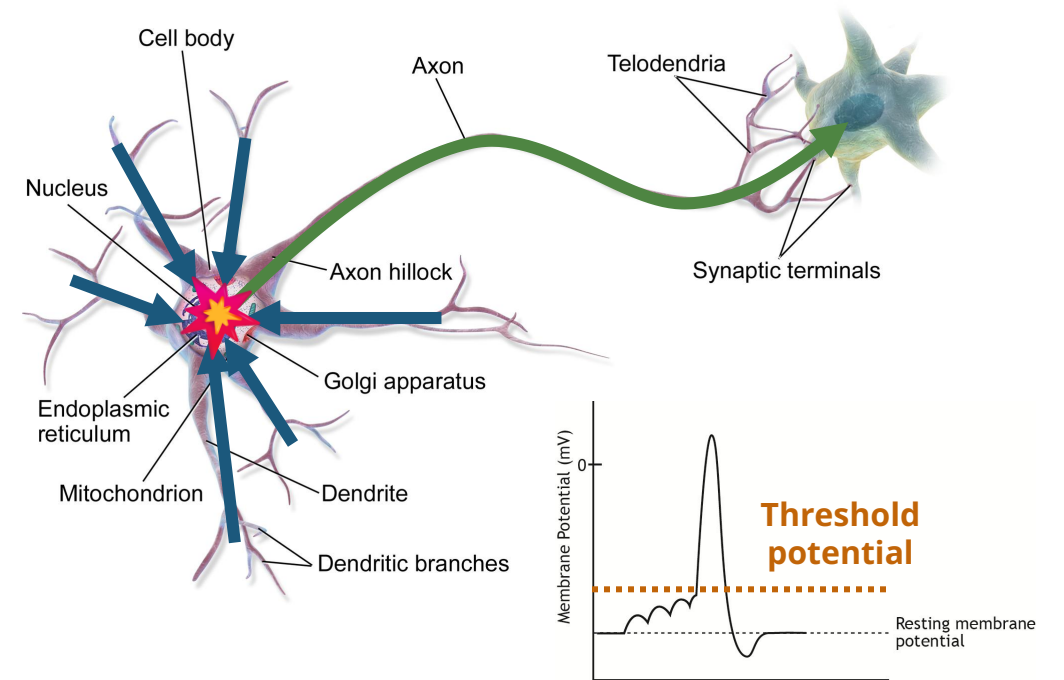


Artificial vs. Human Neuron

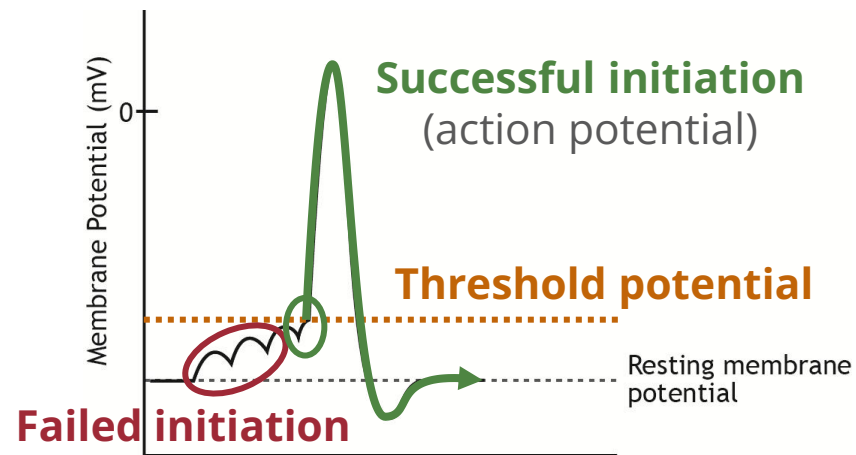
Artificial neuron



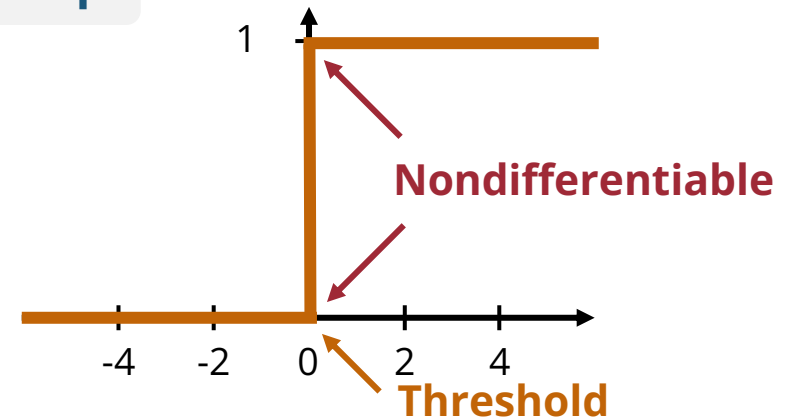
Human neuron



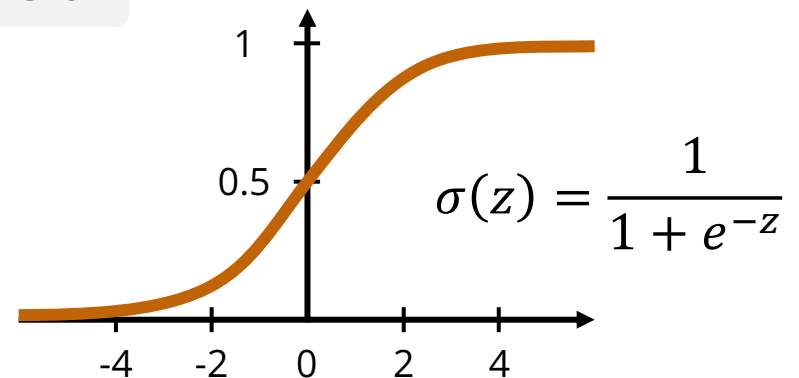
Why Sigmoid?



Unit step



Sigmoid



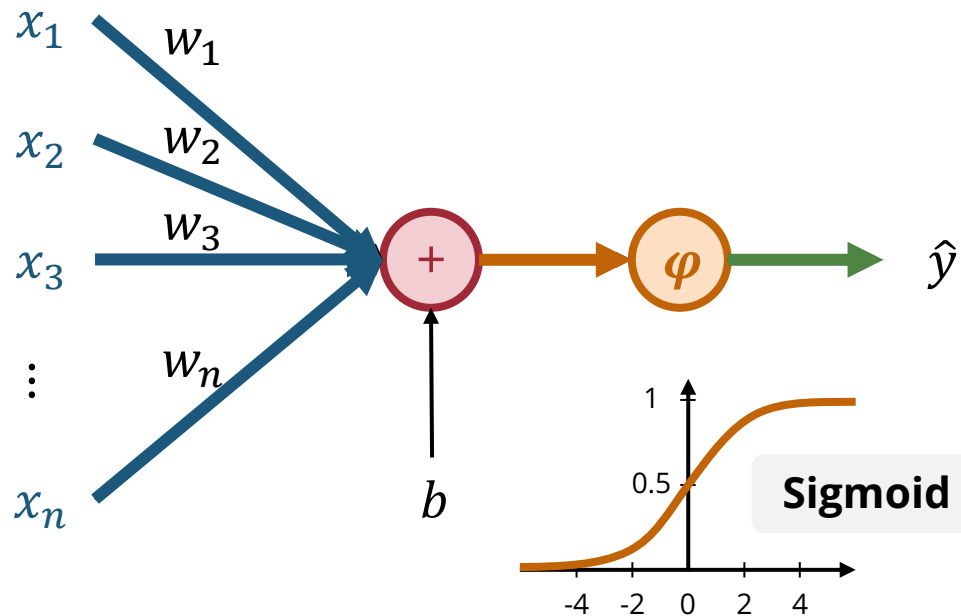
| Why Bias Term?

- Allow nonzero outputs when **all** inputs are zero

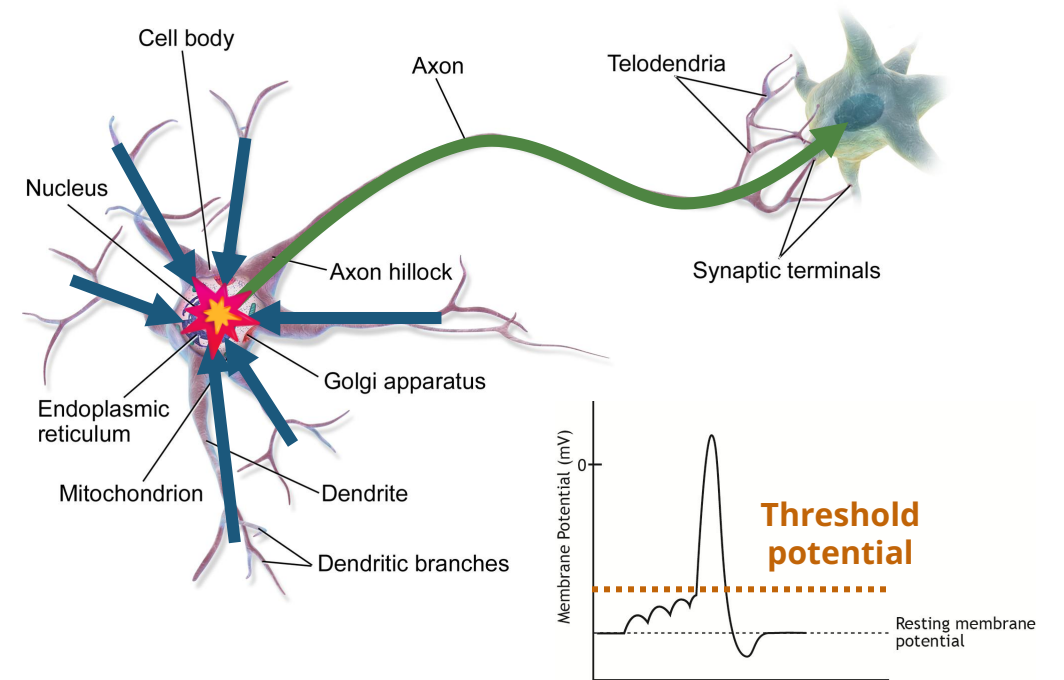
$$\hat{y} = \varphi(w_1 \cancel{x_1}^0 + w_2 \cancel{x_2}^0 + \cdots + w_n \cancel{x_n}^0 + b) = \varphi(b)$$

Artificial vs. Human Neuron

Artificial neuron

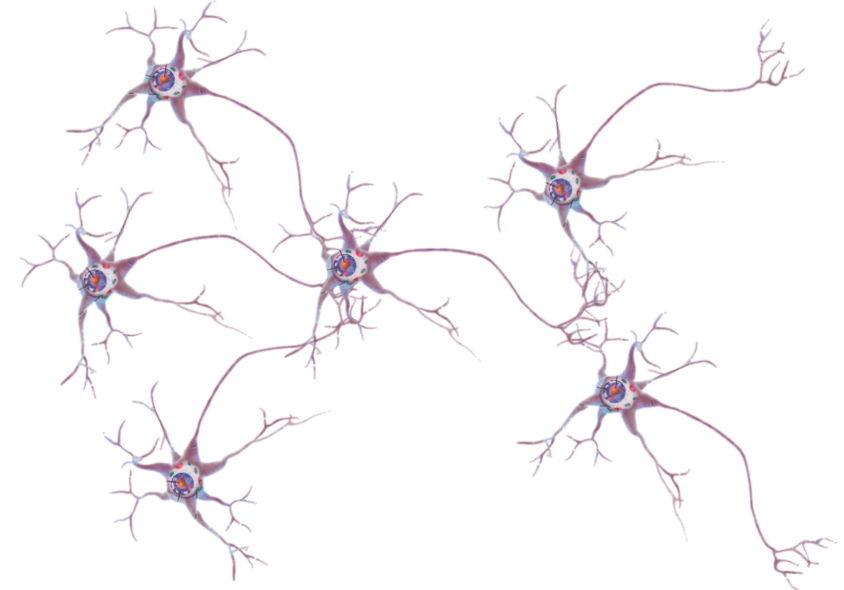
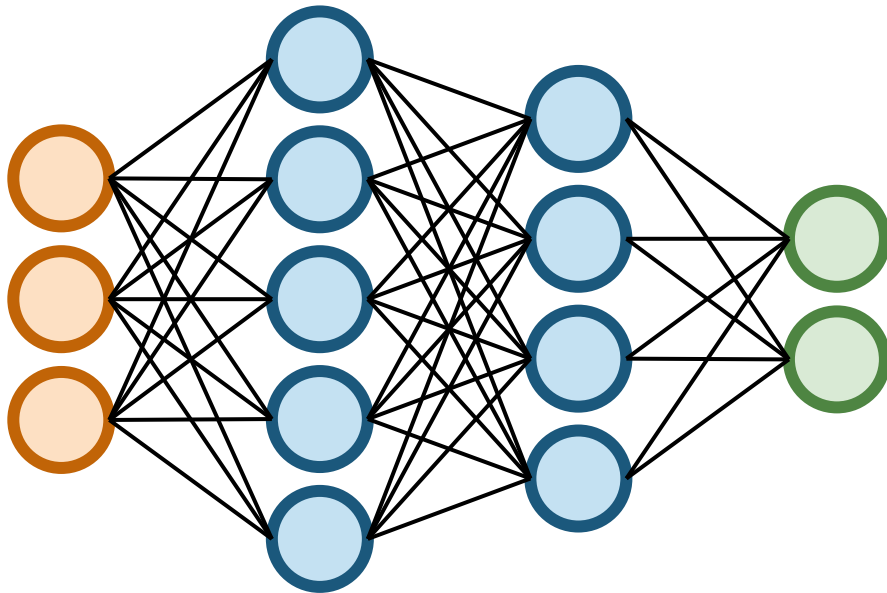


Human neuron



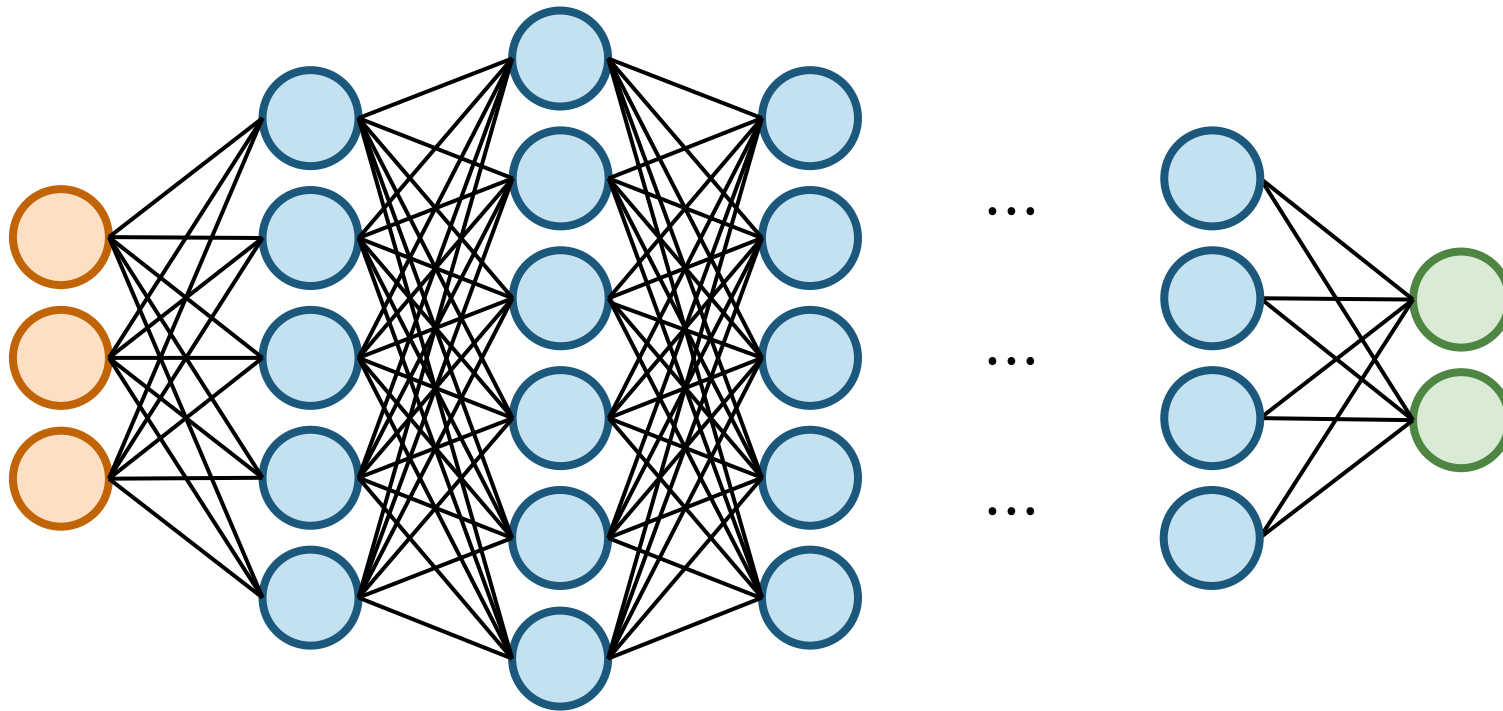
Artificial Neural Networks

- Although inspired by human neural networks, artificial neural networks nowadays *do not work like human brains*
 - Lacking **functional hierarchy, high-level feedback loops, memory module**, etc.
 - Human brains work more like **spiking neural networks** → Efficiency!



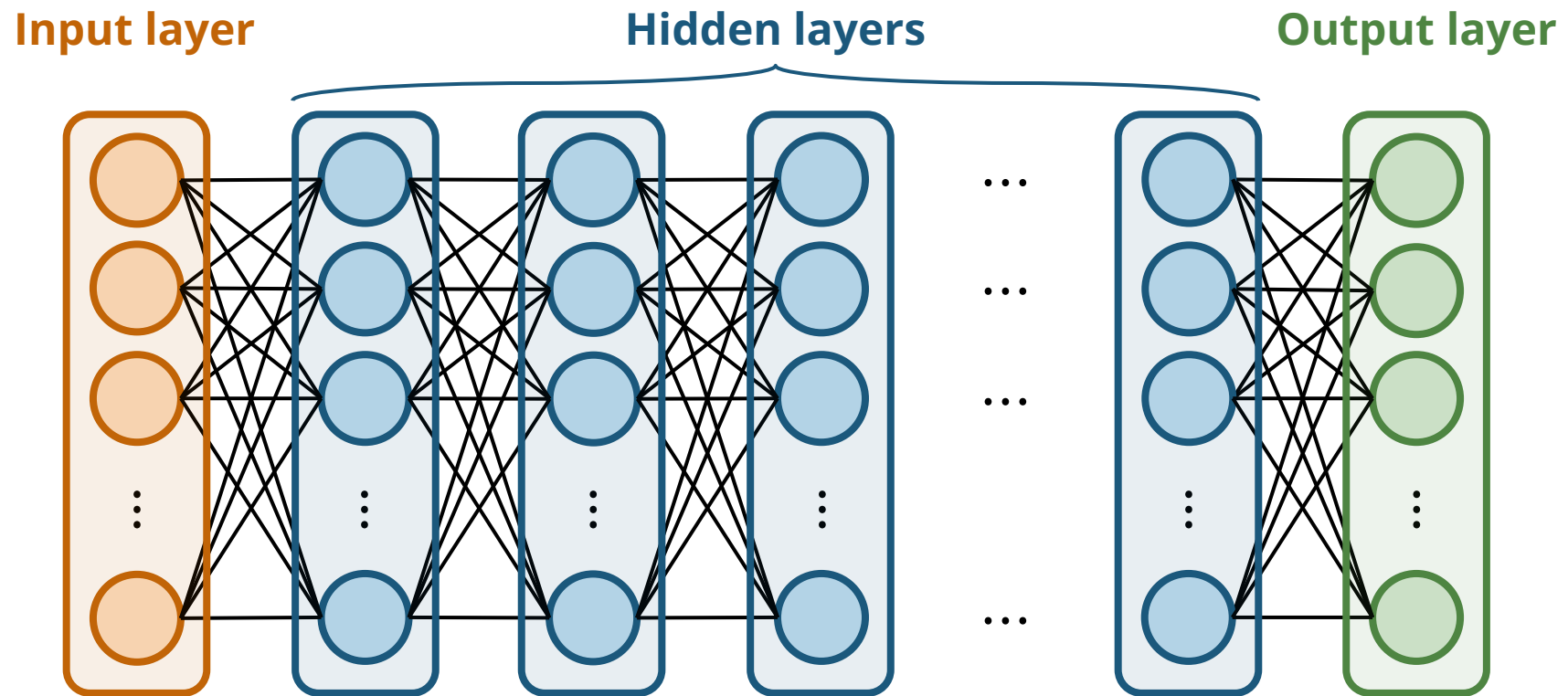
Fully Connected Feedforward Network

- Most basic form of deep neural networks

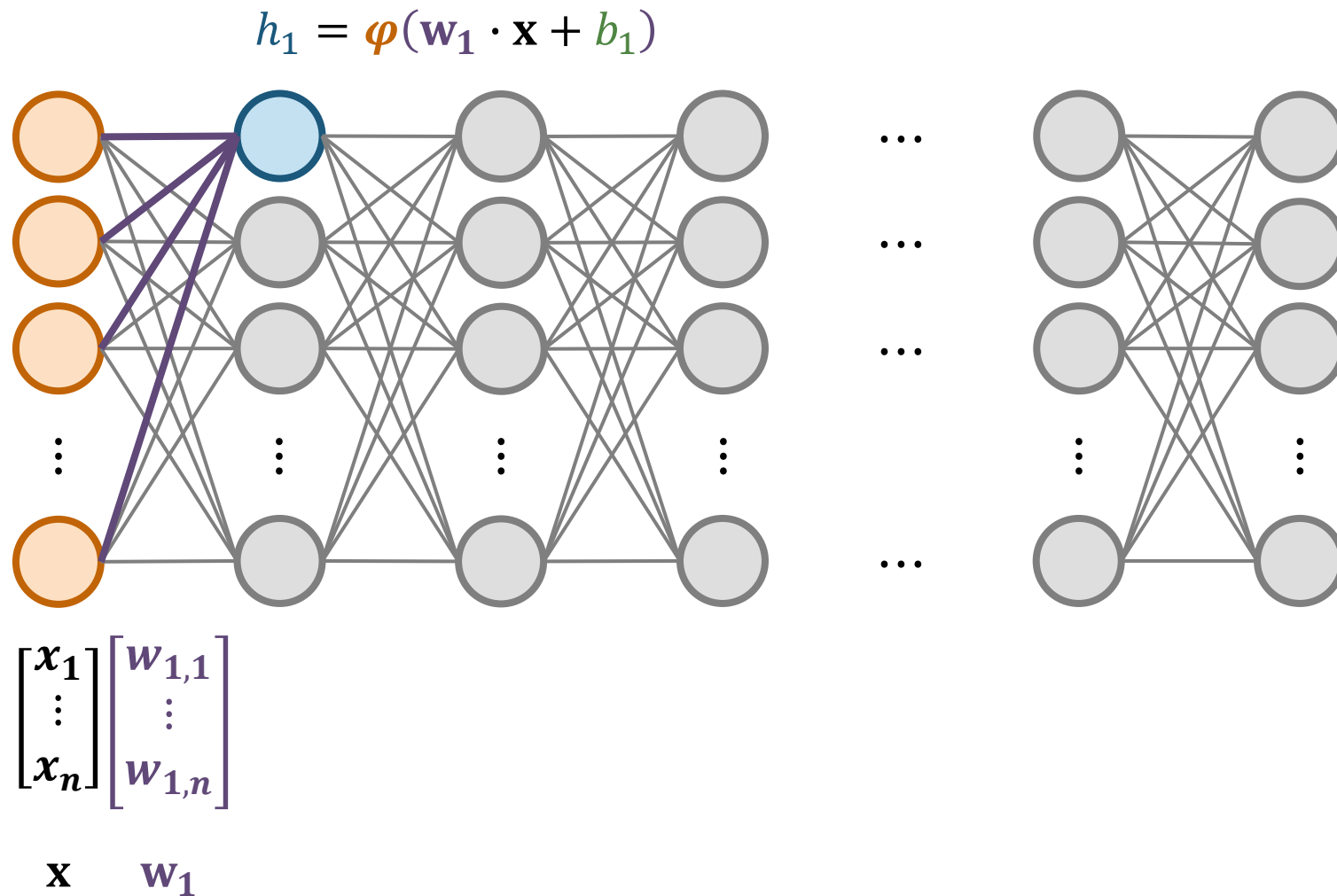


Math Formulation

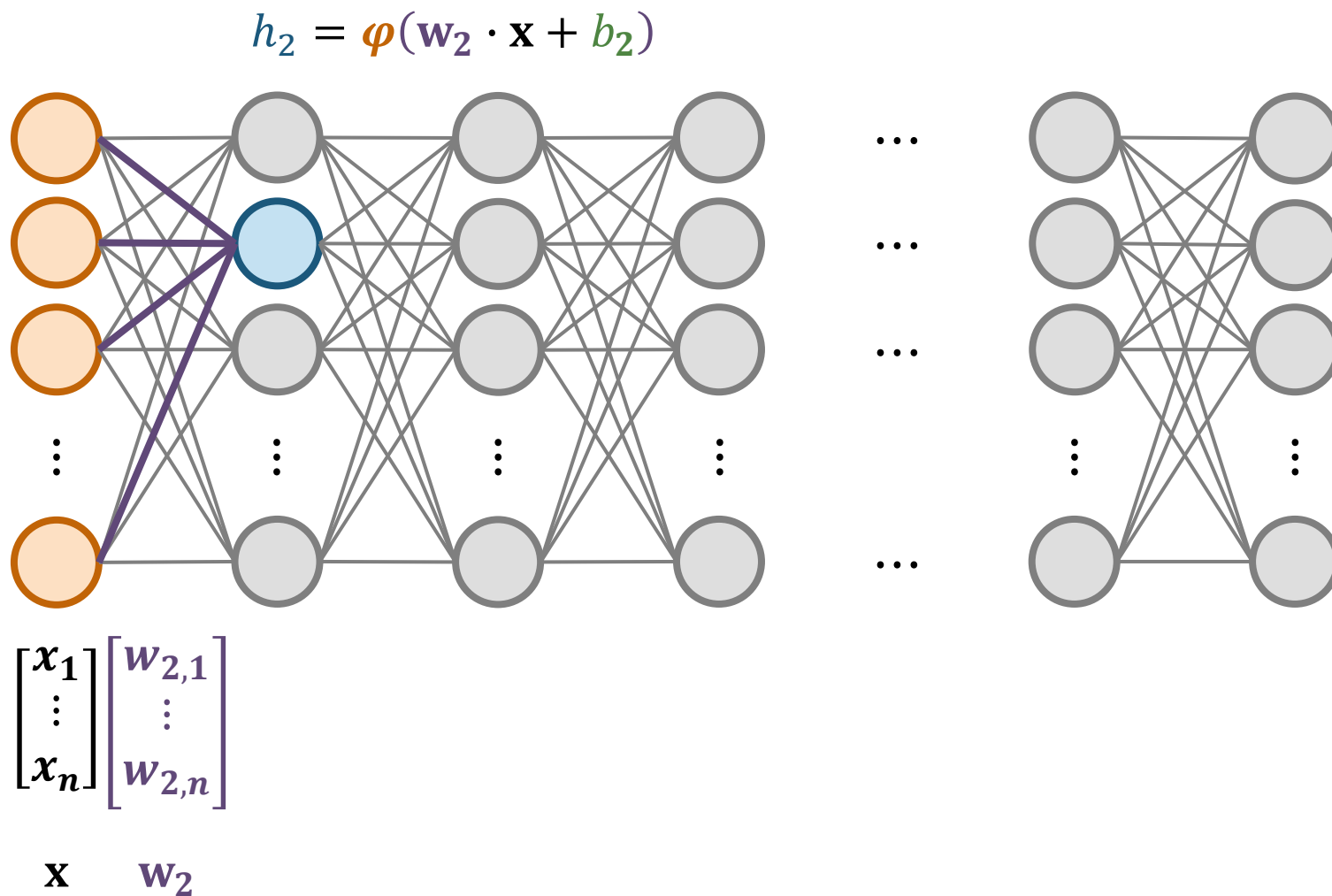
| Math Formulation



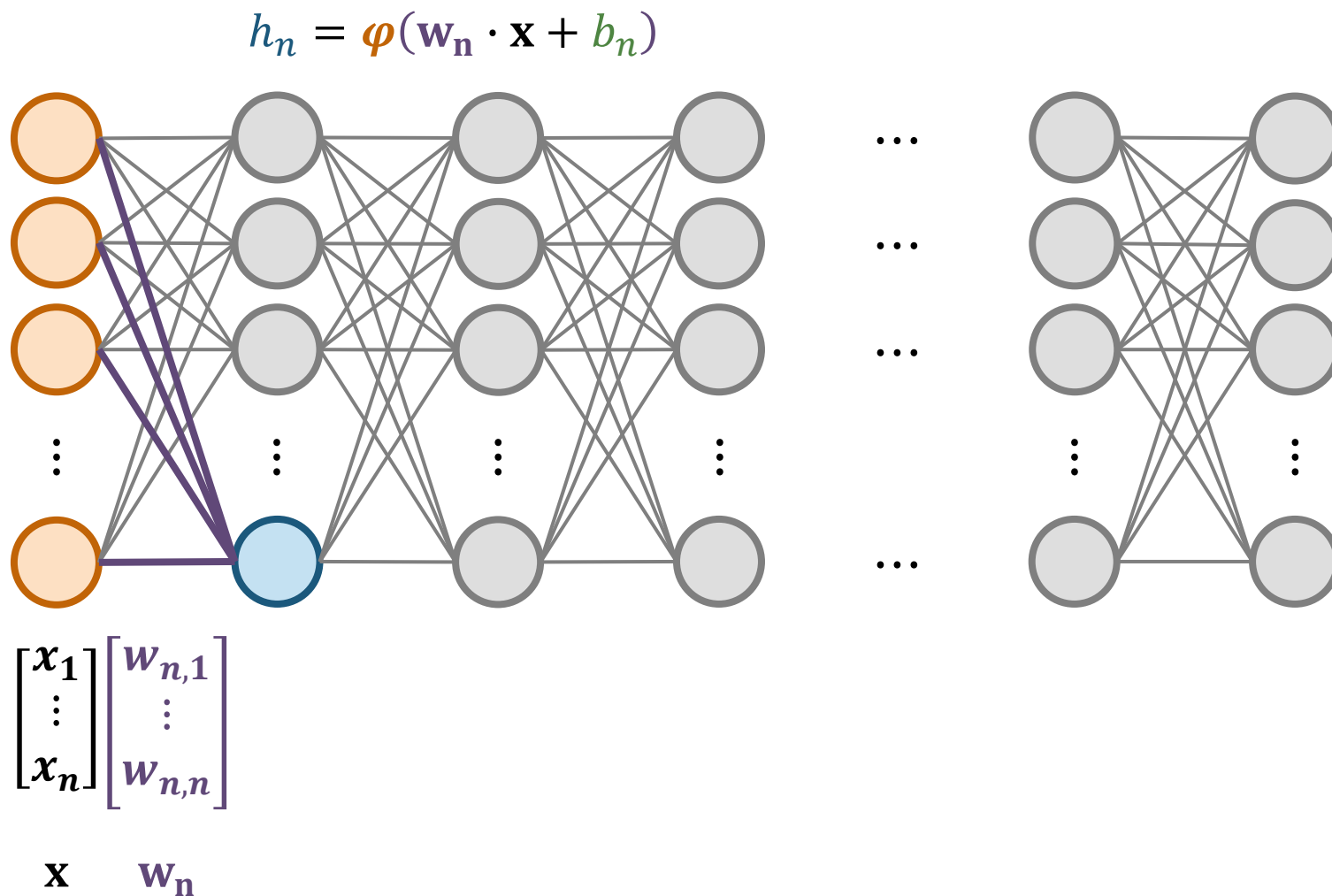
Math Formulation



Math Formulation

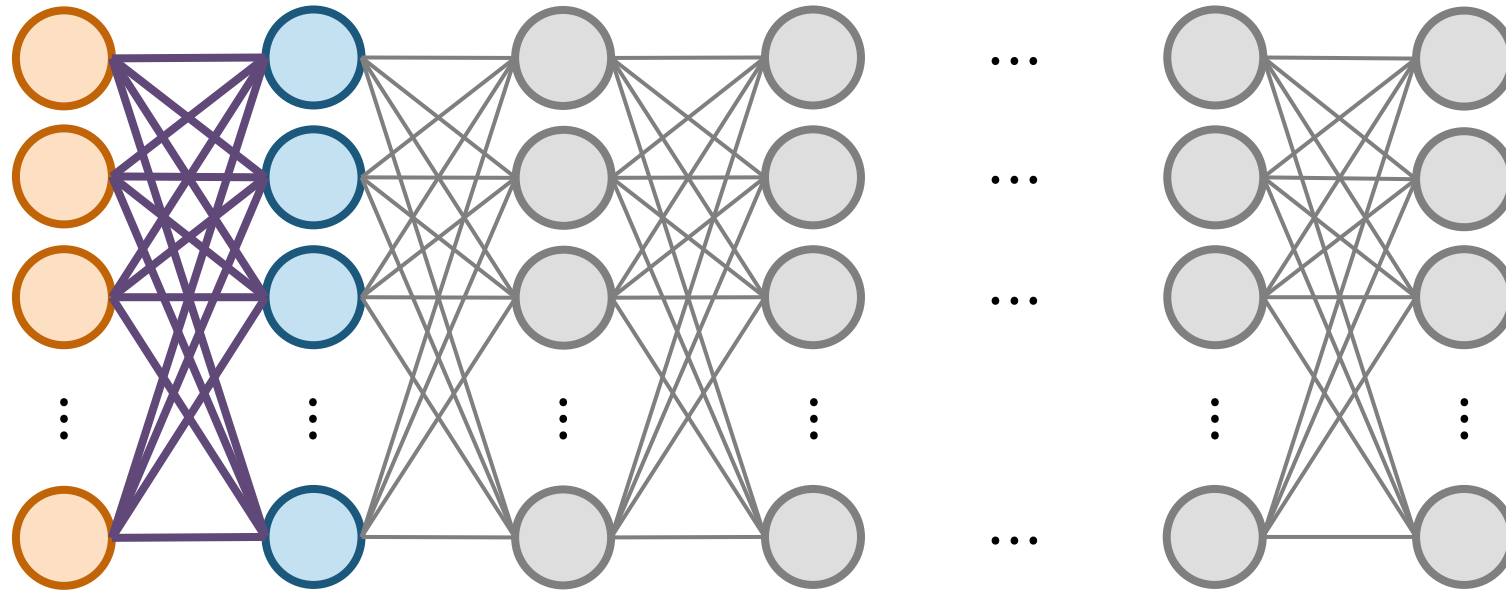


Math Formulation



Math Formulation

$$\mathbf{h} = \varphi(W\mathbf{x} + \mathbf{b})$$

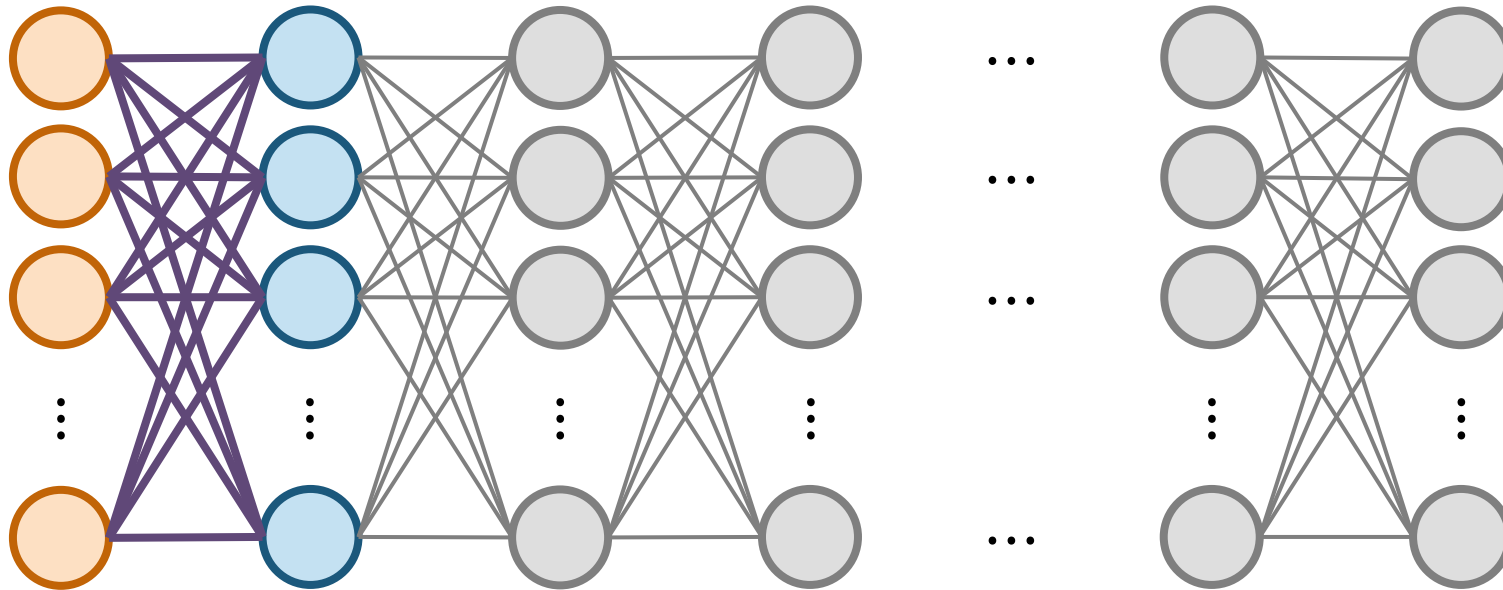


$$\begin{bmatrix} w_{1,1} & \cdots & w_{1,n} \\ \vdots & \ddots & \vdots \\ w_{n,1} & \cdots & w_{n,n} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}$$

$W \qquad \mathbf{x} \qquad \mathbf{b}$

Math Formulation

$$\mathbf{h} = \varphi(W\mathbf{x} + \mathbf{b})$$

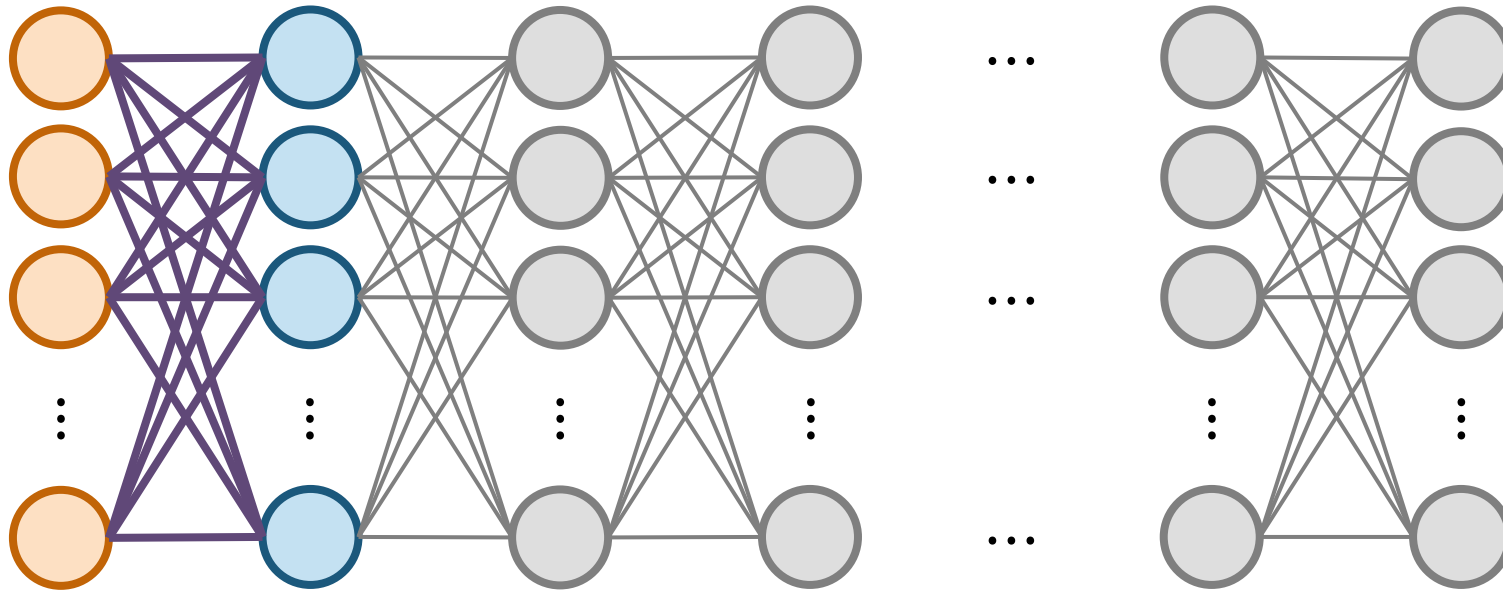


$$\begin{bmatrix} w_{1,1} & \cdots & w_{1,n} \\ \vdots & \ddots & \vdots \\ w_{n,1} & \cdots & w_{n,n} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} \quad h_1 = \varphi(\mathbf{w}_1 \cdot \mathbf{x} + b_1)$$

$W \qquad \mathbf{x} \qquad \mathbf{b}$

Math Formulation

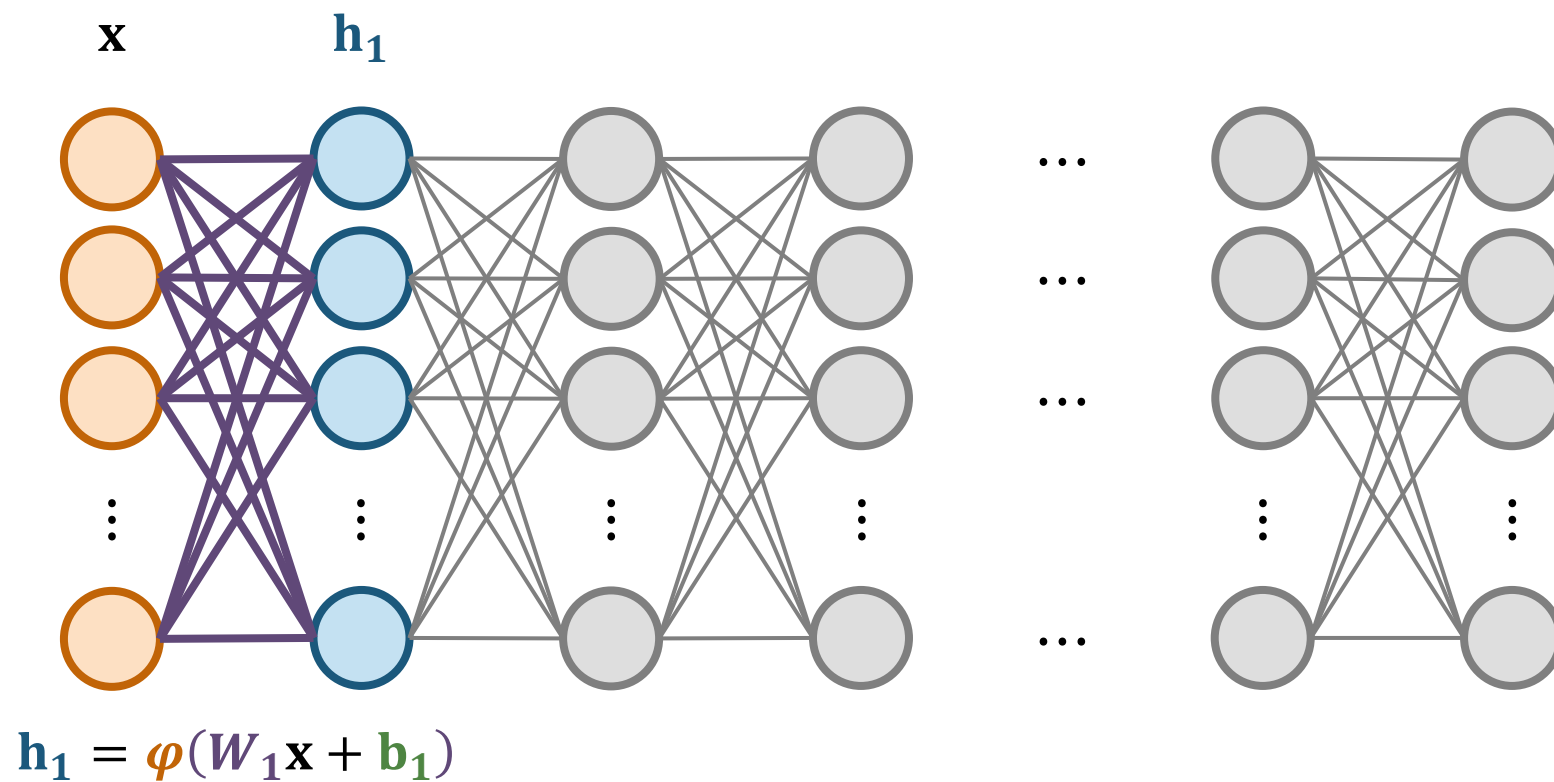
$$\mathbf{h} = \varphi(W\mathbf{x} + \mathbf{b})$$



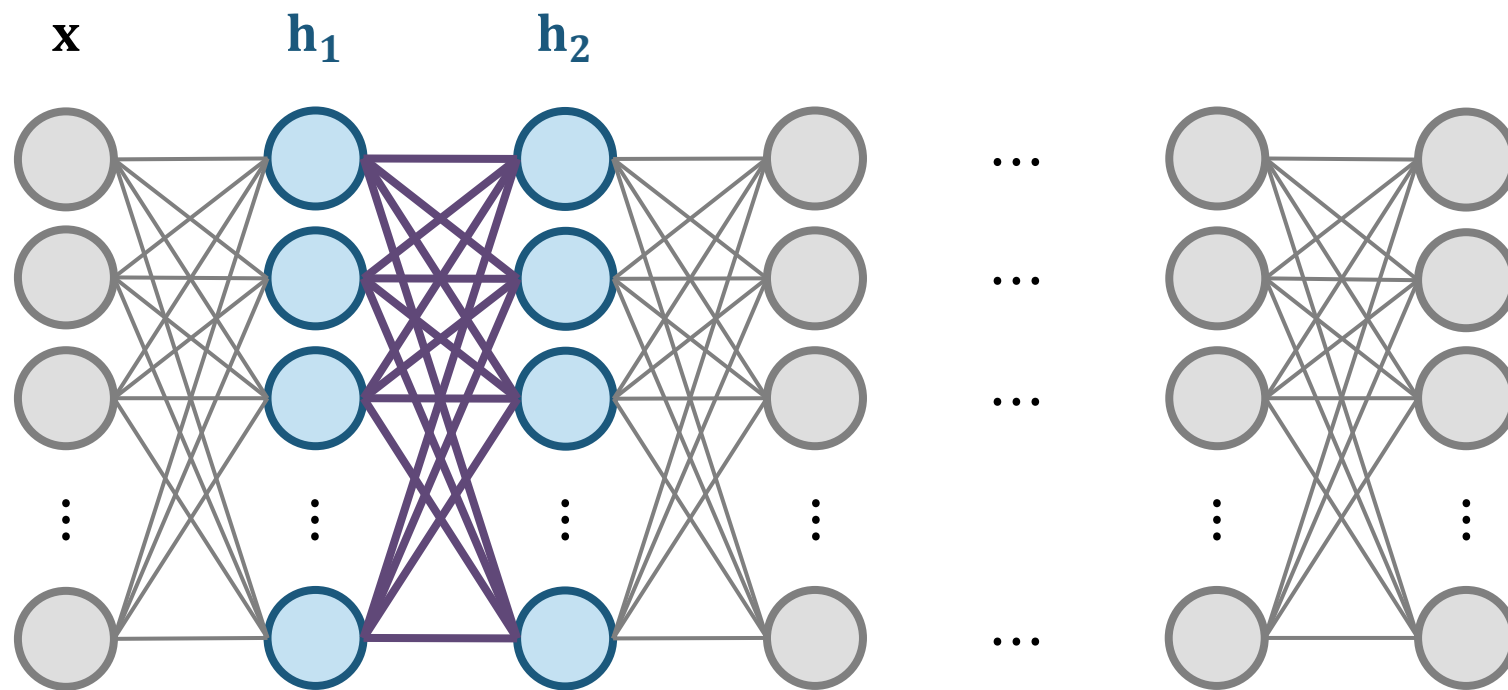
$$\begin{bmatrix} w_{1,1} & \cdots & w_{1,n} \\ \vdots & \ddots & \vdots \\ w_{n,1} & \cdots & w_{n,n} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} \quad h_n = \varphi(w_n \cdot \mathbf{x} + b_n)$$

$W \qquad \mathbf{x} \qquad \mathbf{b}$

| Math Formulation

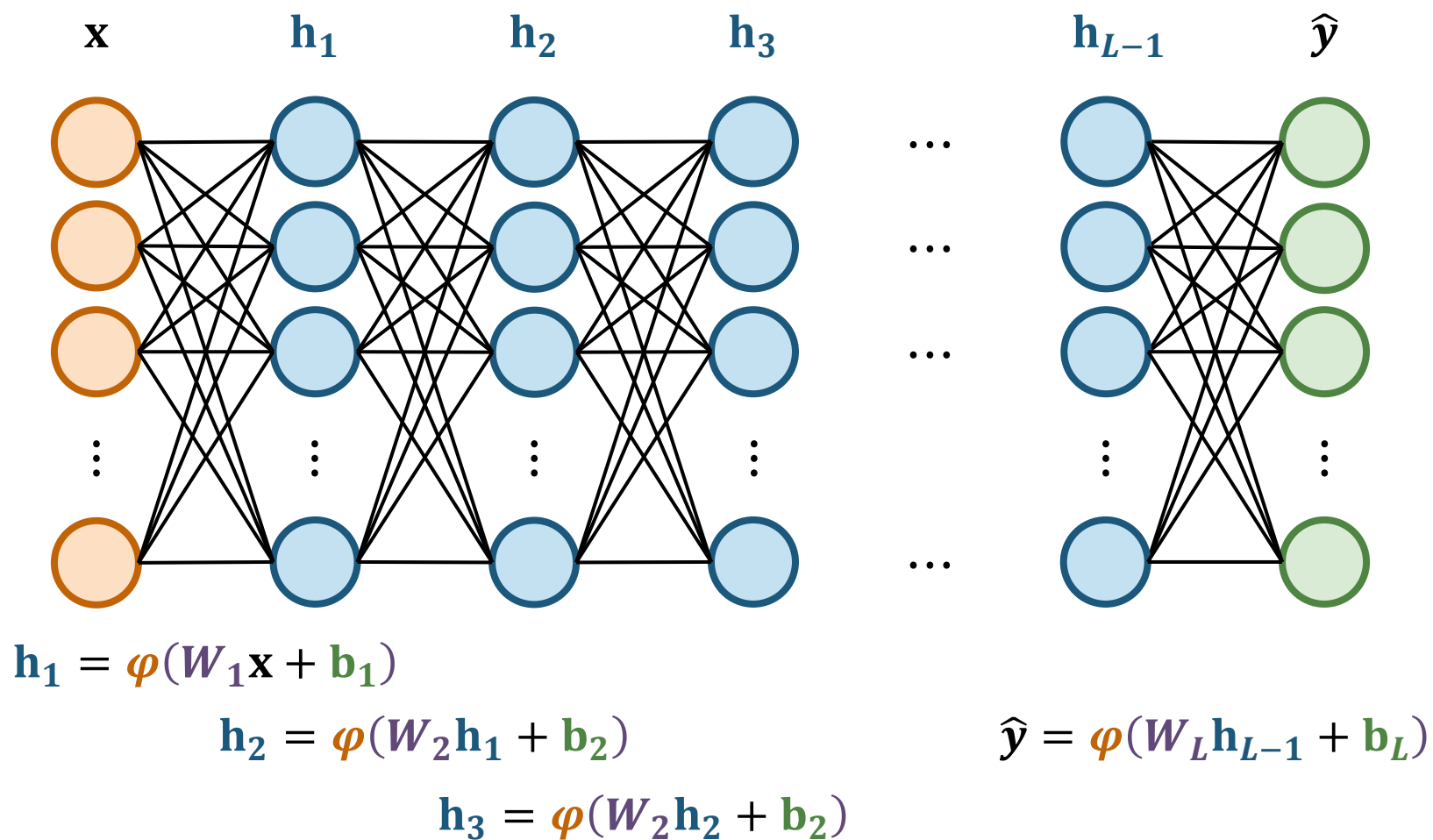


| Math Formulation

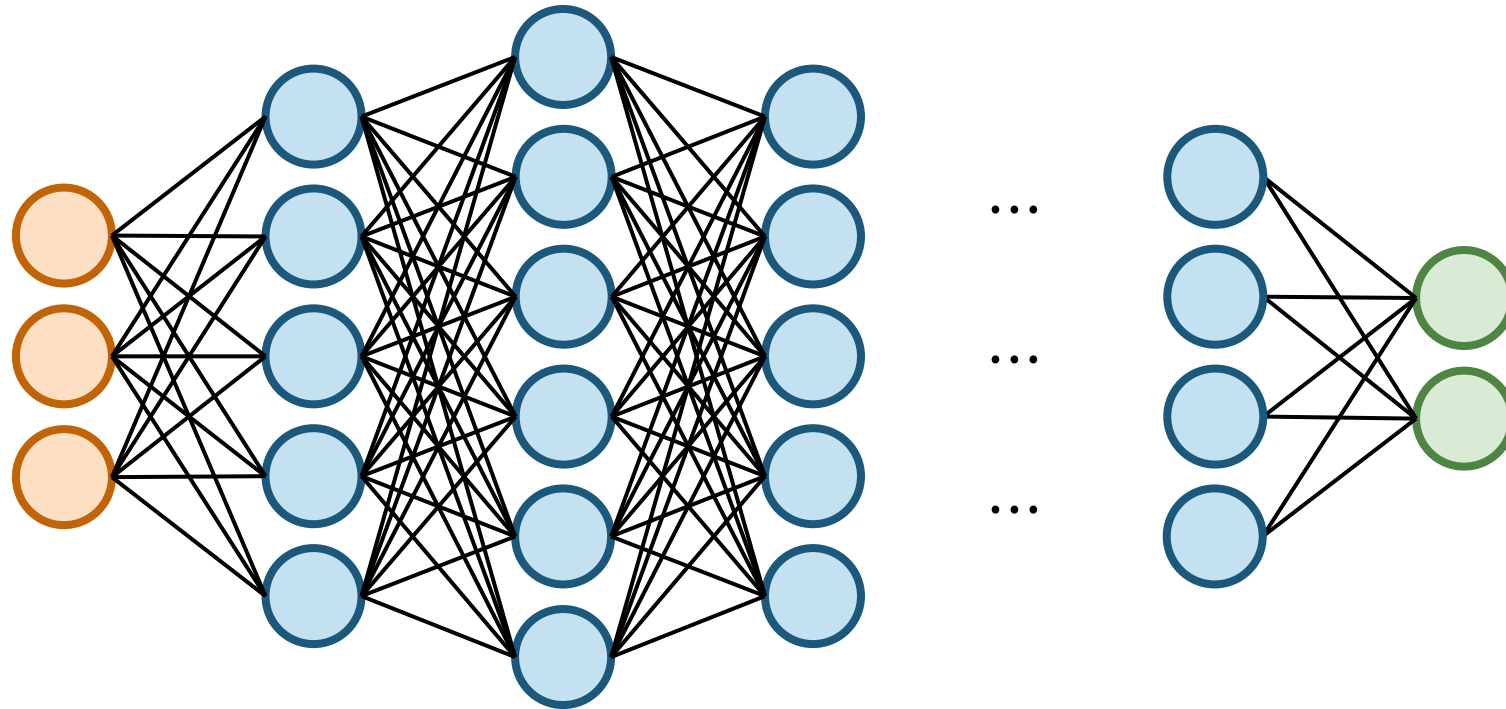


$$\mathbf{h}_2 = \varphi(W_2 \mathbf{h}_1 + \mathbf{b}_2)$$

Math Formulation



Fully Connected Feedforward Network



$$\mathbf{h}_1 = \varphi(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1)$$

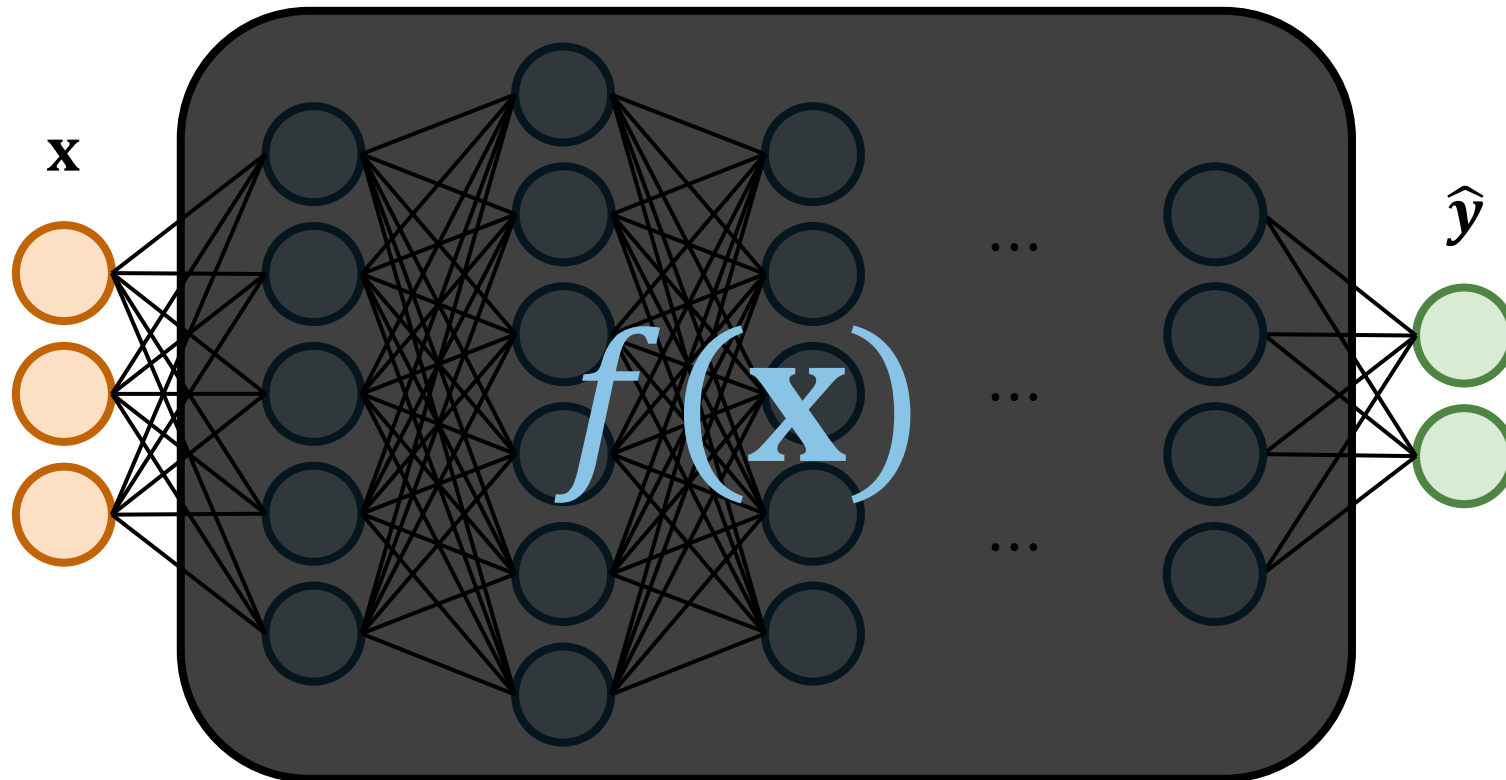
$$\mathbf{h}_2 = \varphi(\mathbf{W}_2 \mathbf{h}_1 + \mathbf{b}_2)$$

$$\mathbf{h}_3 = \varphi(\mathbf{W}_2 \mathbf{h}_2 + \mathbf{b}_2)$$

$$\hat{\mathbf{y}} = \varphi(\mathbf{W}_L \mathbf{h}_{L-1} + \mathbf{b}_L)$$

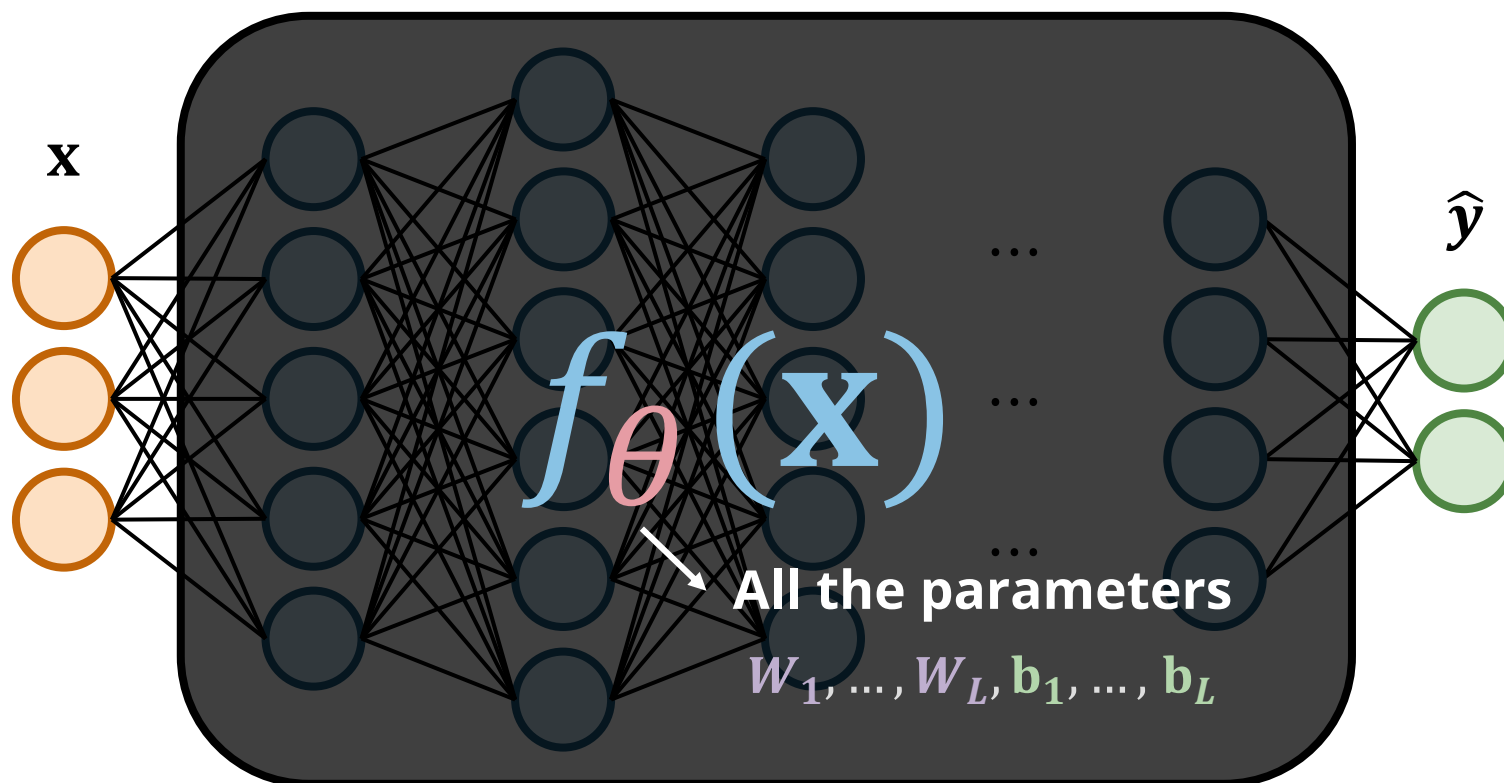
Neural Networks are Parameterized Functions

- A neural network represents **a set of functions**

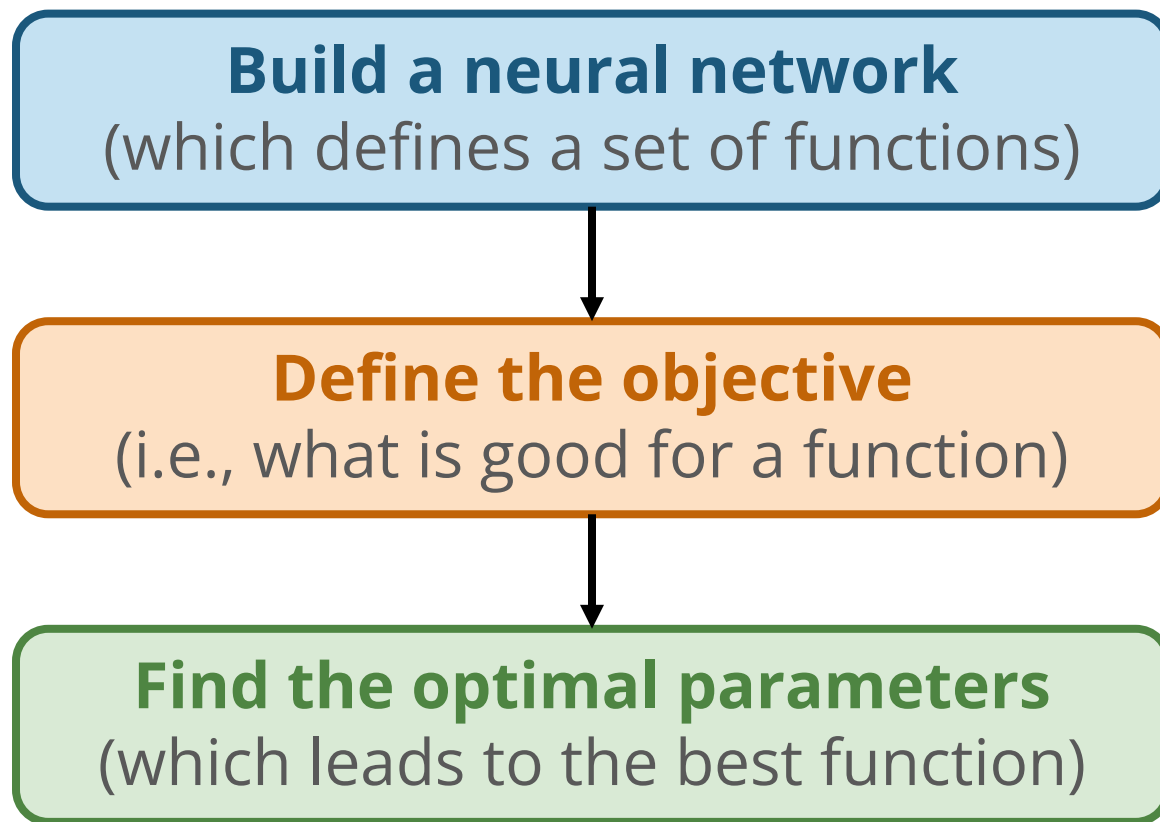


Neural Networks are Parameterized Functions

- A neural network represents **a set of functions**

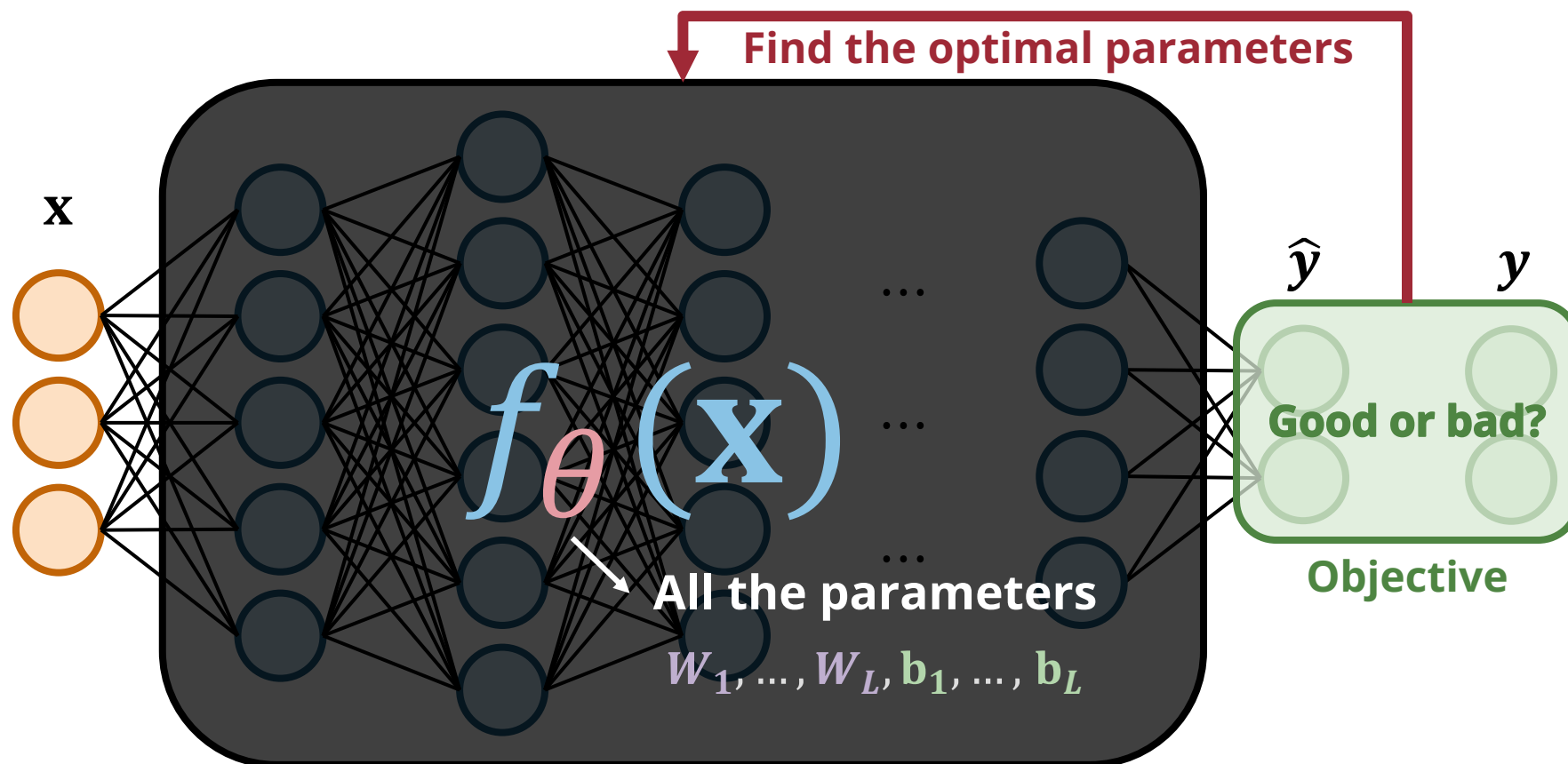


| Training a Neural Network



Neural Networks are Parameterized Functions

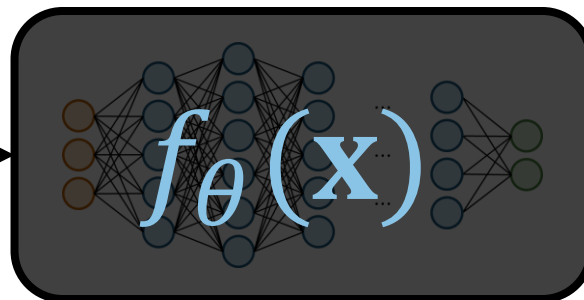
- A neural network represents **a set of functions**



Regression vs. Classification

Regression vs. Classification

Regression

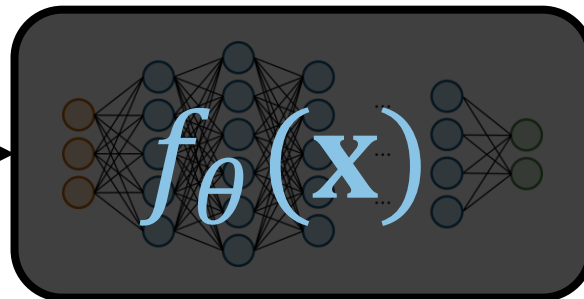


Age

5

Output a number

Classification



Is human?

Yes / No

Output a label



Regression or Classification? Stock Price Prediction



Input

$$\mathbf{x} \in [0, \infty)^T$$

Output

$$y \in [0, \infty)$$

| 🤔 Regression or Classification? Animal Recognition

$$f\left(\text{cat image}\right) = \text{cat}$$

$$f\left(\text{dog image}\right) = \text{dog}$$

$$f\left(\text{bear image}\right) = \text{bear}$$

Input

$$\mathbf{x} \in [0, 1]^{W \times H}$$

Output

$$y \in \{\text{cat, dog, bear}\}$$

🤔 Regression or Classification? Spam Filter

$$f \left(\begin{array}{c} \text{POWERBALL} \\ \text{POWERPLAY} \\ \text{CONGRATULATIONS!!} \\ \text{Your Email was selected in Powerball Lottery} \\ \text{Draw with the sum of 1.5million dollars.} \\ \text{Kindly send your Full Name, Address and} \\ \text{Phone Number for claims.} \\ \text{Yours Sincerely} \\ \text{Mr. James Hodges} \\ \text{Head Of Operations} \end{array} \right) = \text{spam}$$

$$f \left(\begin{array}{c} \text{Hao-Wen Dong <h...> Mon, Sep 9, 4:04 PM (1 day ago)} \\ \text{to PAT, pat.grads} \\ \text{Hi folks,} \\ \text{We are planning an internship panel for our PAT seminar this Friday. That} \\ \text{being said, we'll need some panelists! If you did an internship this} \\ \text{summer (or previously) or have experience working in the industry,} \\ \text{please let me know! Also, feel free to recommend anyone who you} \\ \text{think would be a good panelist for this topic.} \\ \text{The goal of the panel is to give you a sense of what the application} \\ \text{process/timeline is like and what the whole internship experience is like.} \\ \text{Looking forward to hearing from you! And see you on Friday!} \end{array} \right) = \text{not spam}$$

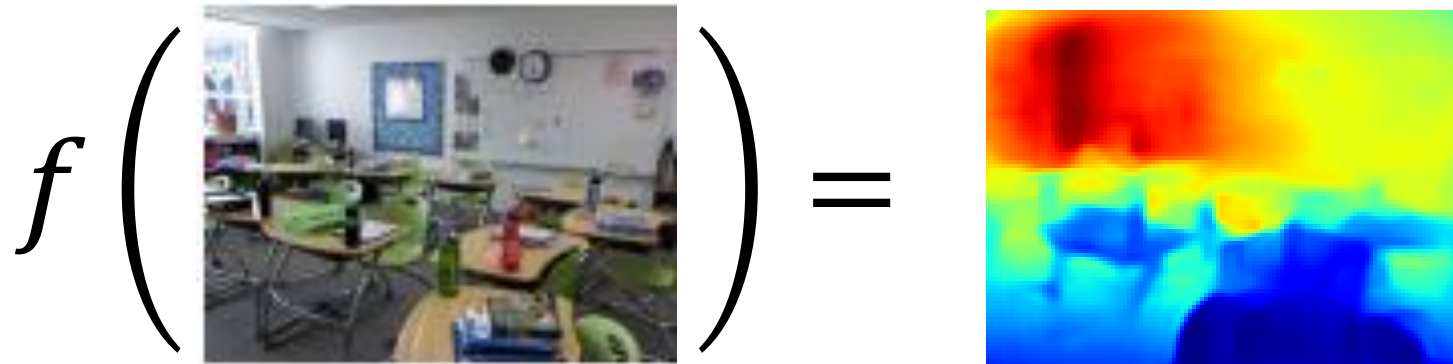
Input

$$\mathbf{x} \in \mathcal{V}^N$$

Output

$$y \in \{\text{spam, not spam}\}$$

🤔 Regression or Classification? Depth Estimation



Input

$$\mathbf{x} \in [0, 1]^{W \times H \times 3}$$

Output

$$\mathbf{y} \in [0, \infty)^{W \times H}$$

| 🤔 Regression or Classification? Digit Recognition

$$f\left(\begin{array}{c} \text{8} \end{array}\right) = 8$$

$$f\left(\begin{array}{c} \text{6} \end{array}\right) = 6$$

Input

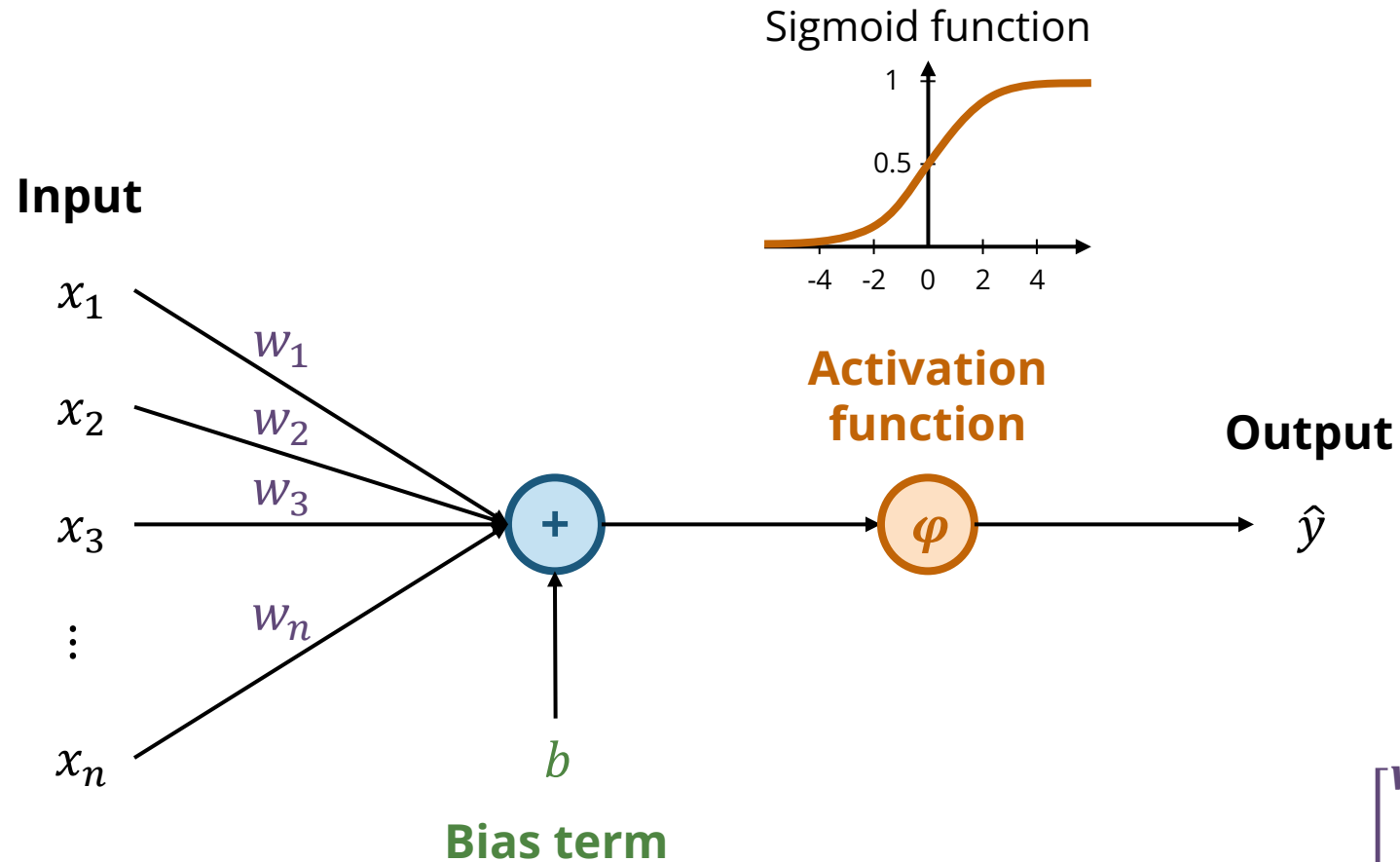
$$\mathbf{x} \in [0, 1]^{W \times H}$$

Output

$$y \in \{0, 1, \dots, 9\}$$

Activation Functions

Inside a Neuron



$$\hat{y} = \varphi(w_1x_1 + w_2x_2 + \dots + w_nx_n + b) = \varphi\left(\sum_{i=1}^n w_i x_i + b\right) = \varphi(\mathbf{w} \cdot \mathbf{x} + b)$$

Why do We Need Activation Functions?

- Activation functions introduce **nonlinearity** to a neural network
- A linear function is a **weighted sum of the inputs** (plus a bias term)

$$f(x_1, x_2, \dots, x_n) = a_1x_1 + a_2x_2 + a_3x_3 + \dots + a_nx_n + b$$

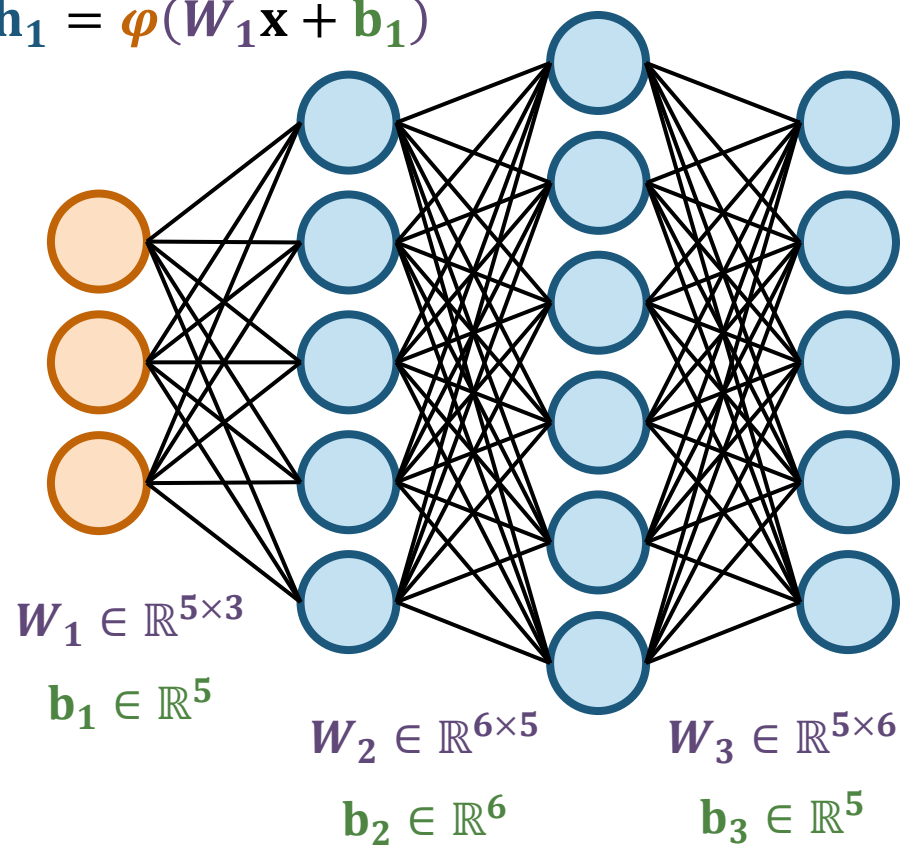
- Examples of nonlinear functions:

- $f(x_1) = \frac{1}{x_1}$
- $f(x_1) = x_1^2$
- $f(x_1) = e^x$
- $f(x_1, x_2) = x_1x_2$

**Nonlinear functions are hard to model and approximate.
That's where deep neural networks shine!**

Why do We Need Activation Functions?

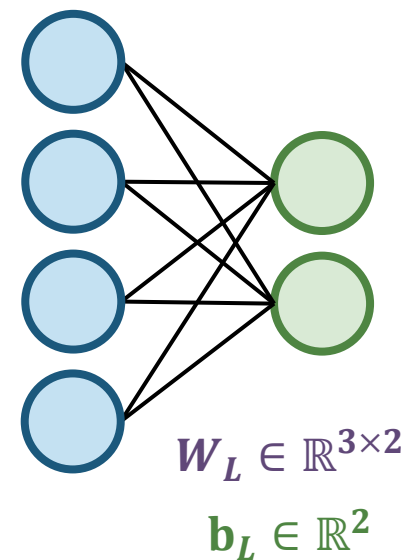
$$\mathbf{h}_1 = \varphi(W_1 \mathbf{x} + \mathbf{b}_1)$$



...

...

...



| Why do We Need Activation Functions?

$$\mathbf{h}_1 = \varphi(W_1 \mathbf{x} + \mathbf{b}_1)$$

$$\hat{\mathbf{y}} = \varphi(W_L \mathbf{h}_{L-1} + \mathbf{b}_L)$$

$$\mathbf{h}_2 = \varphi(W_2 \mathbf{h}_1 + \mathbf{b}_2)$$

$$\hat{\mathbf{y}} = \varphi(W_L \varphi(W_{L-1} \mathbf{h}_{L-2} + \mathbf{b}_{L-1}) + \mathbf{b}_L)$$

$$\mathbf{h}_3 = \varphi(W_3 \mathbf{h}_2 + \mathbf{b}_3)$$

$$\hat{\mathbf{y}} = \varphi(W_L \varphi(W_{L-1} \varphi(W_{L-2} \mathbf{h}_{L-3} + \mathbf{b}_{L-2}) + \mathbf{b}_{L-1}) + \mathbf{b}_L)$$

⋮

⋮

$$\hat{\mathbf{y}} = \varphi(W_L \mathbf{h}_{L-1} + \mathbf{b}_L)$$

$$\hat{\mathbf{y}} = \varphi(W_L \varphi(W_{L-1} \varphi(W_{L-2} \varphi(\cdots \mathbf{x} \cdots) + \mathbf{b}_{L-2}) + \mathbf{b}_{L-1}) + \mathbf{b}_L)$$

| Why do We Need Activation Functions?

With activation functions, a neural network can represent **nonlinear functions**

$$\hat{y} = \varphi(W_L \varphi(W_{L-1} \varphi(W_{L-2} \varphi(\cdots \mathbf{x} \cdots) + \mathbf{b}_{L-2}) + \mathbf{b}_{L-1}) + \mathbf{b}_L)$$

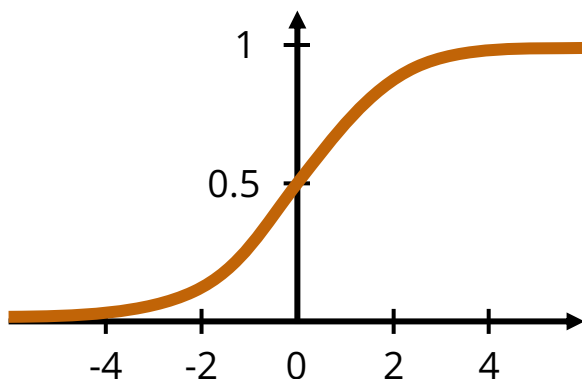


$$\hat{y} = W_L(W_{L-1}(W_{L-2}(\cdots \mathbf{x} \cdots) + \mathbf{b}_{L-2}) + \mathbf{b}_{L-1}) + \mathbf{b}_L$$

Without activation functions, a neural network can only represent **linear functions**

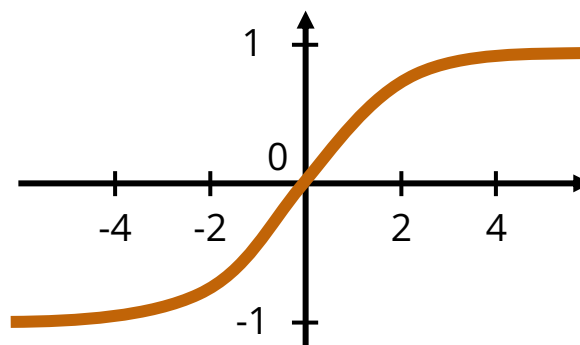
Commonly Used Activation Functions

Sigmoid



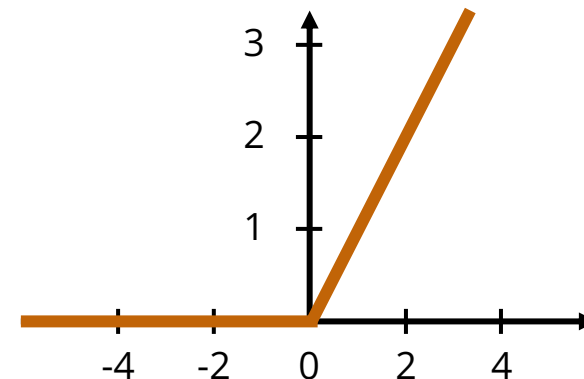
$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

tanh



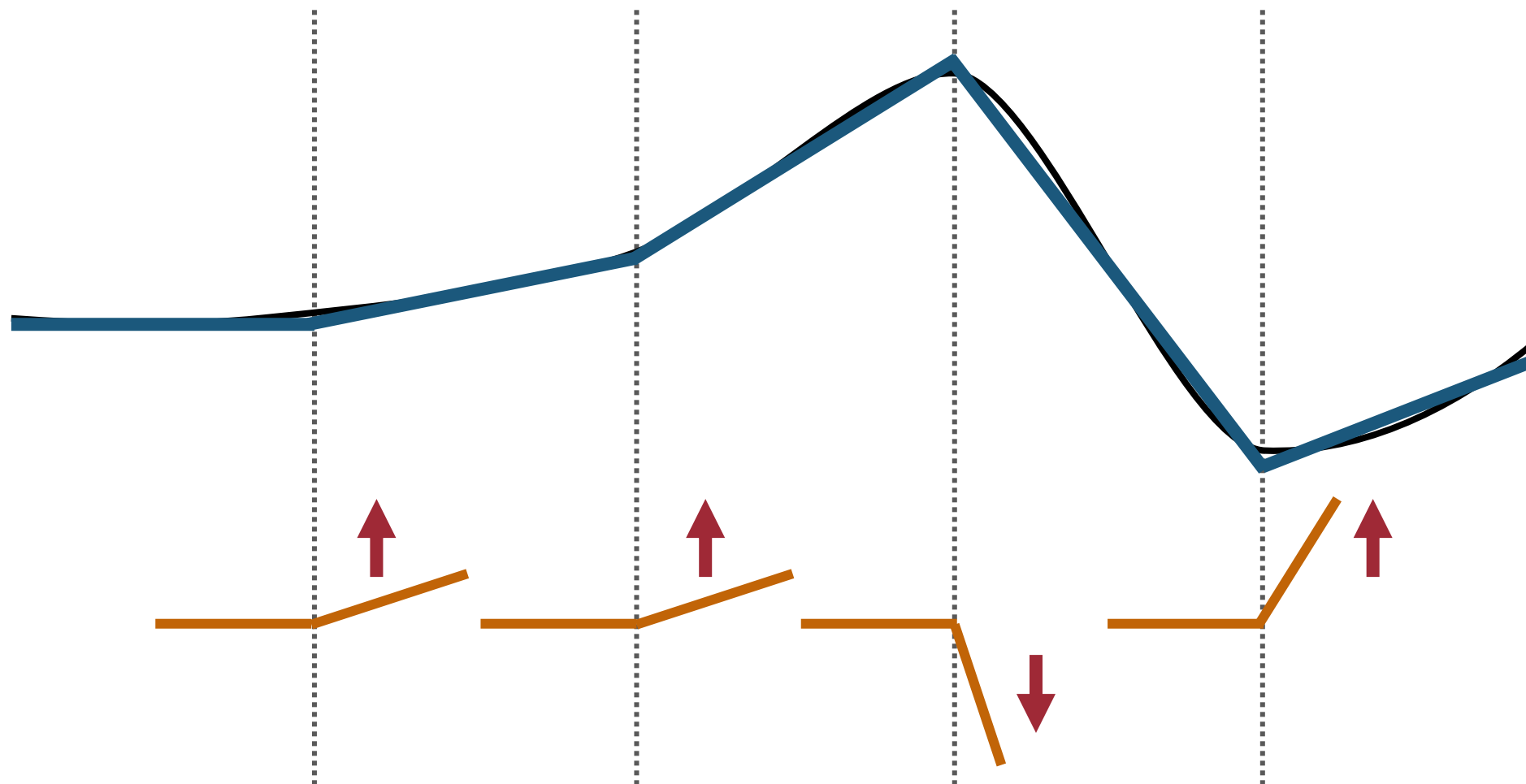
$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

ReLU



$$\text{ReLU}(z) = \begin{cases} z, & \text{if } z \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

ReLUs & Piecewise Linear Functions

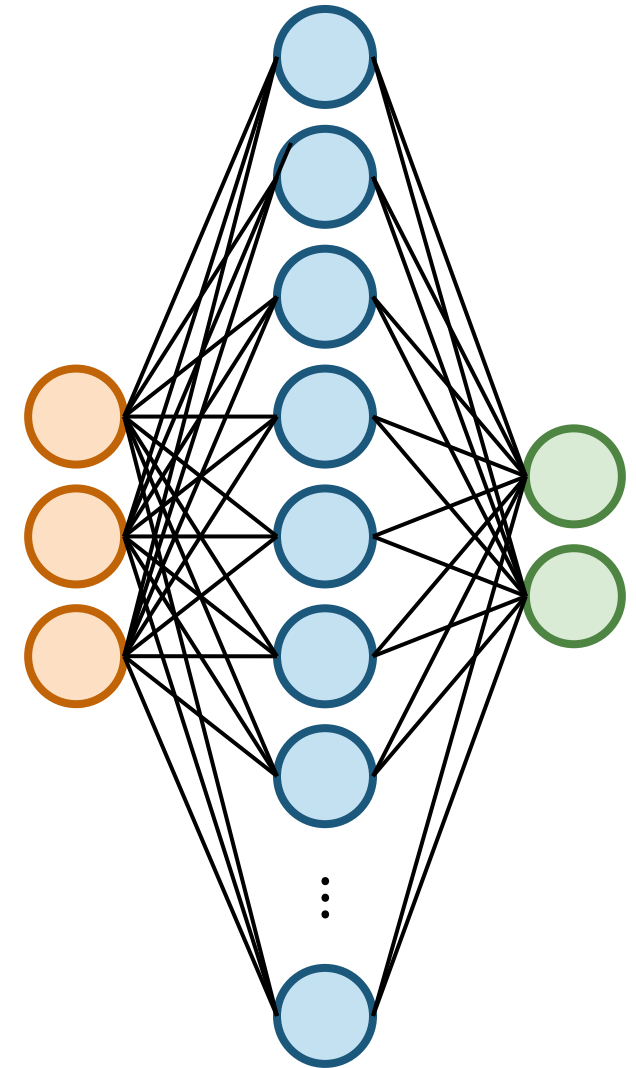


Expressiveness of Neural Networks

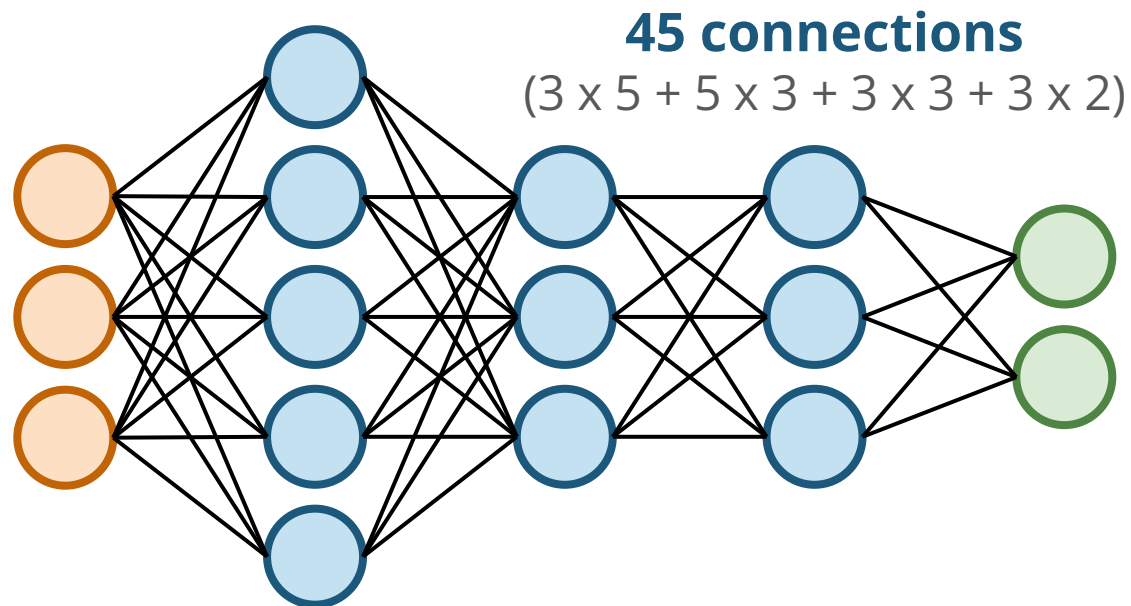
Universal Approximation Theorem

- A neural network with **one hidden layer** can **approximate any continuous function** given **sufficient hidden neurons** and **appropriate activation functions**
 - Sigmoid, ReLUs are good activation functions

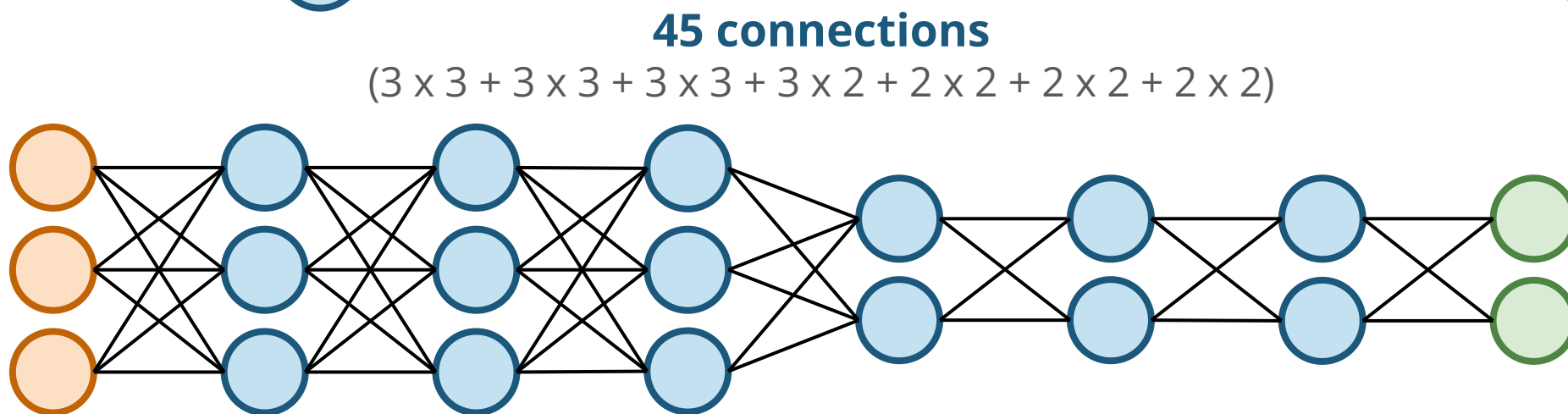
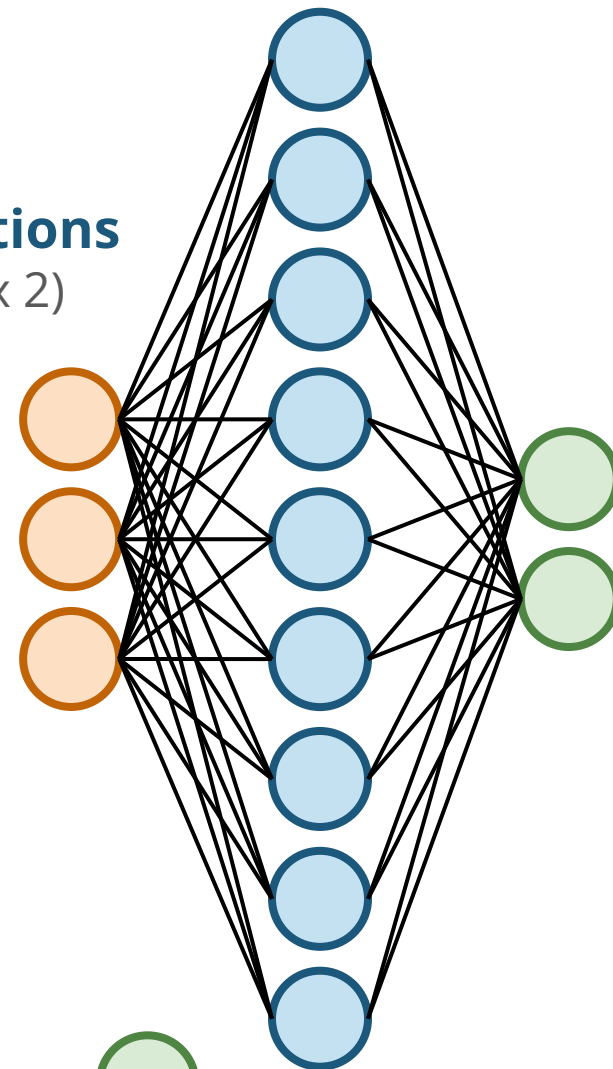
Then why do we want to go deep?



Shallow vs. Deep Neural Networks



45 connections
 $(3 \times 9 + 9 \times 2)$



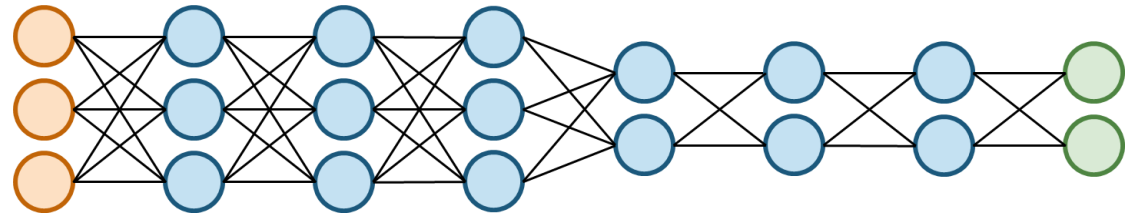
Shallow vs. Deep Neural Networks – In Practice

Shallow neural nets



Less expressive
(less parameter efficient)

Deep neural nets



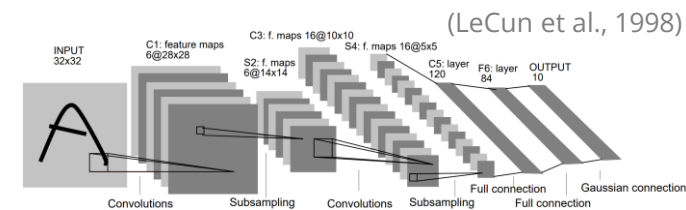
More expressive
(more parameter efficient)

How Deep is Deep Enough?

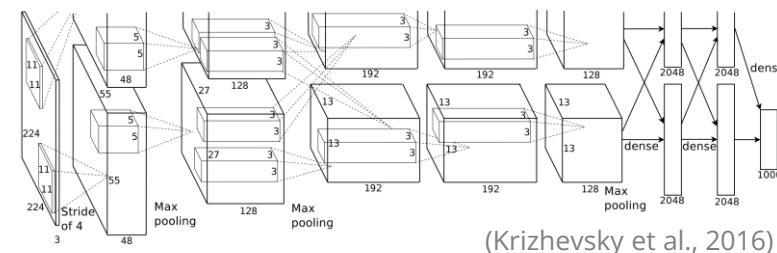
- **Deeper is not always better**

- Actual number of parameters
- Optimization difficulties
- Data size
- Inductive bias of the model

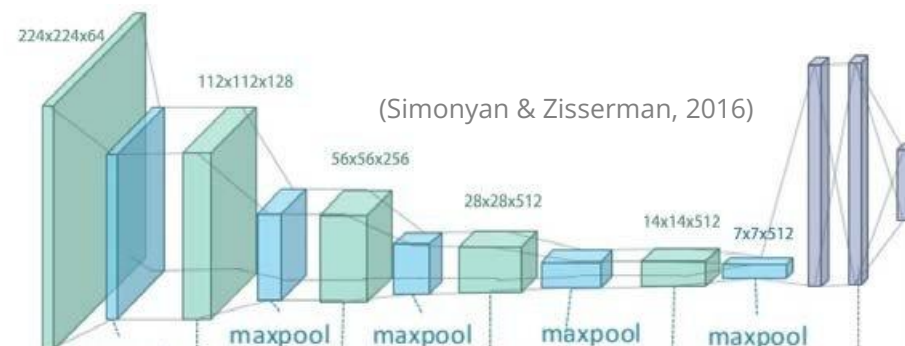
LeNet
(1998)



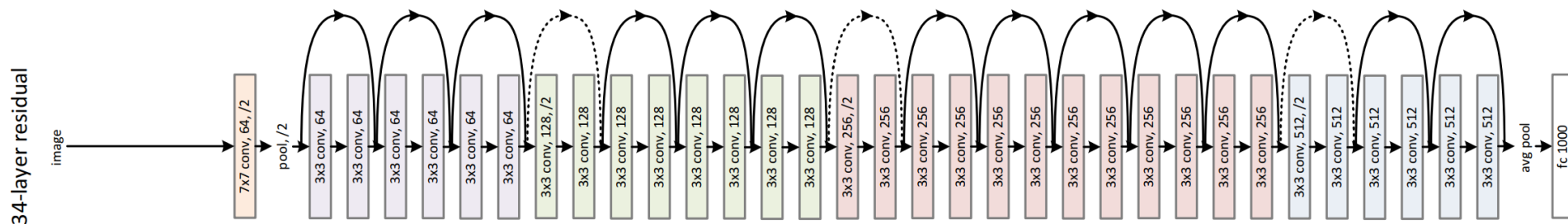
AlexNet
(2012)



VGG-19
(2015)



ResNet
(2015)



Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, 1998.

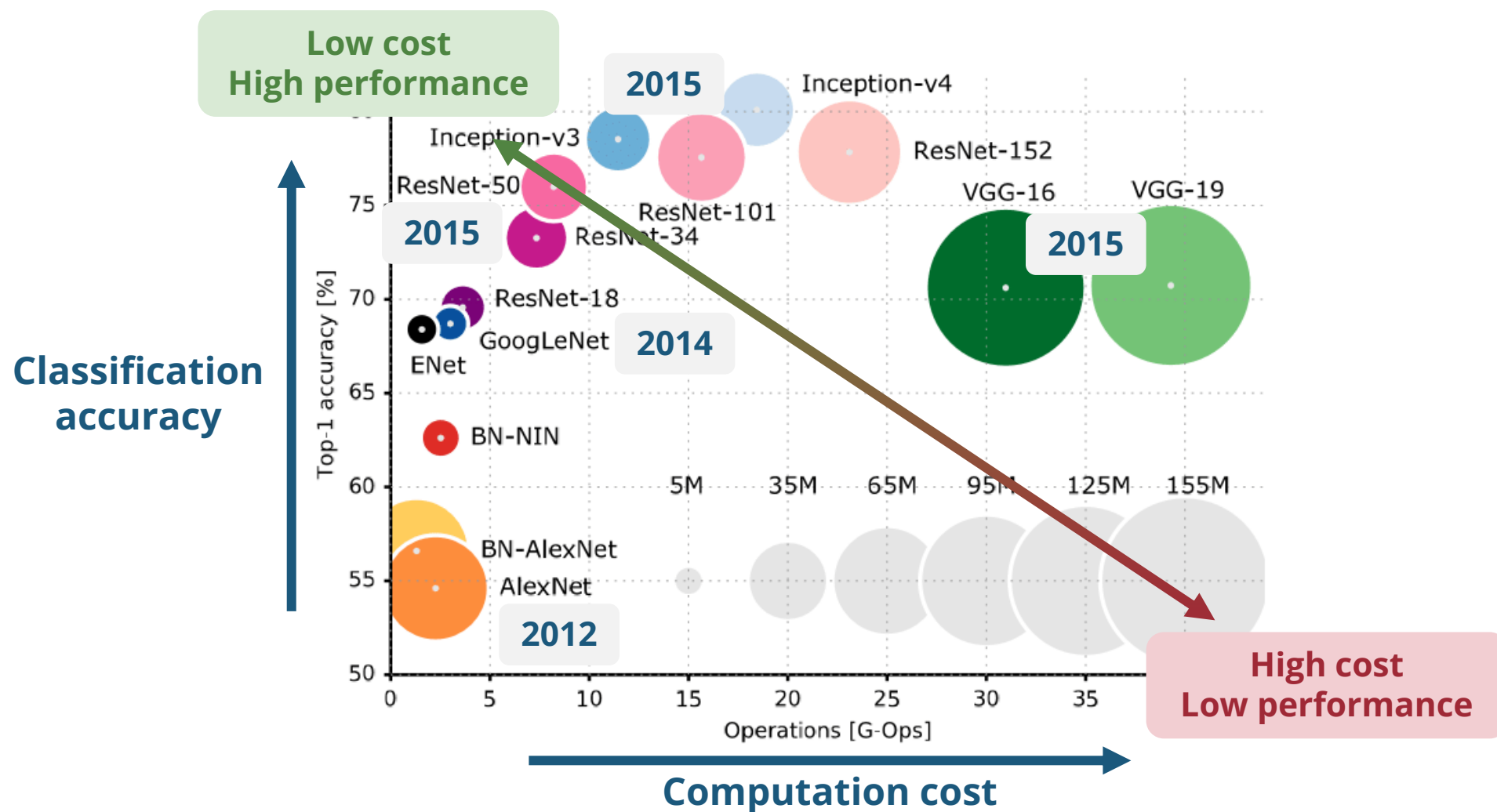
Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *NeurIPS*, 2012.

Karen Simonyan and Andrew Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *ICLR*, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep Residual Learning for Image Recognition," *CVPR*, 2016.

(He et al., 2016)

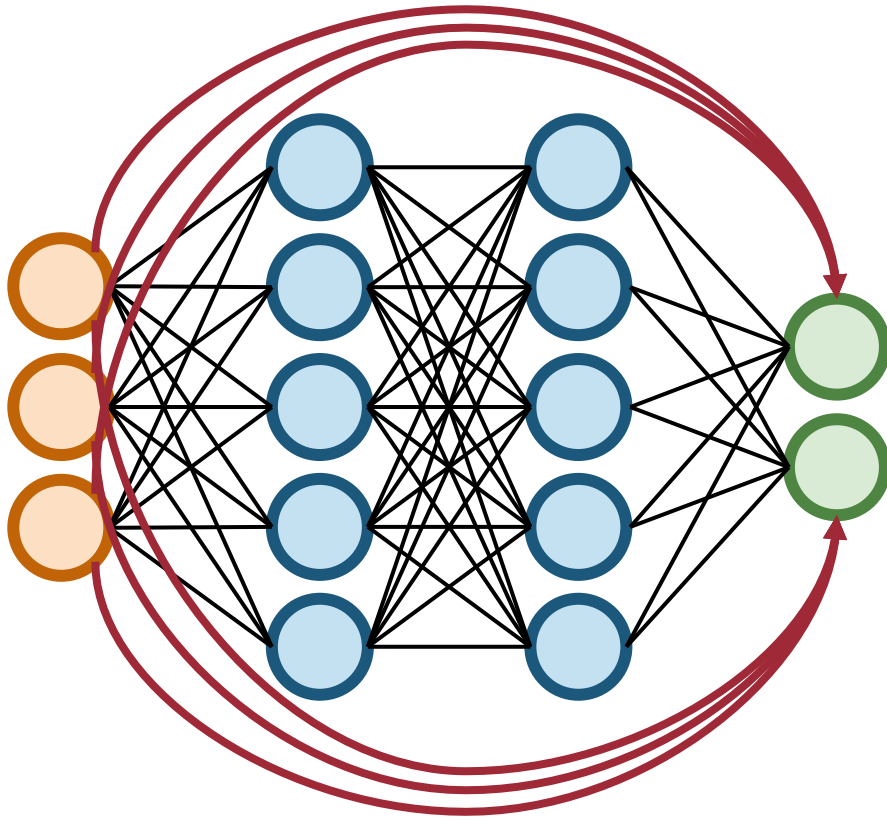
Computation Cost vs. Classification Accuracy



(Source: Canziani et al., 2016)

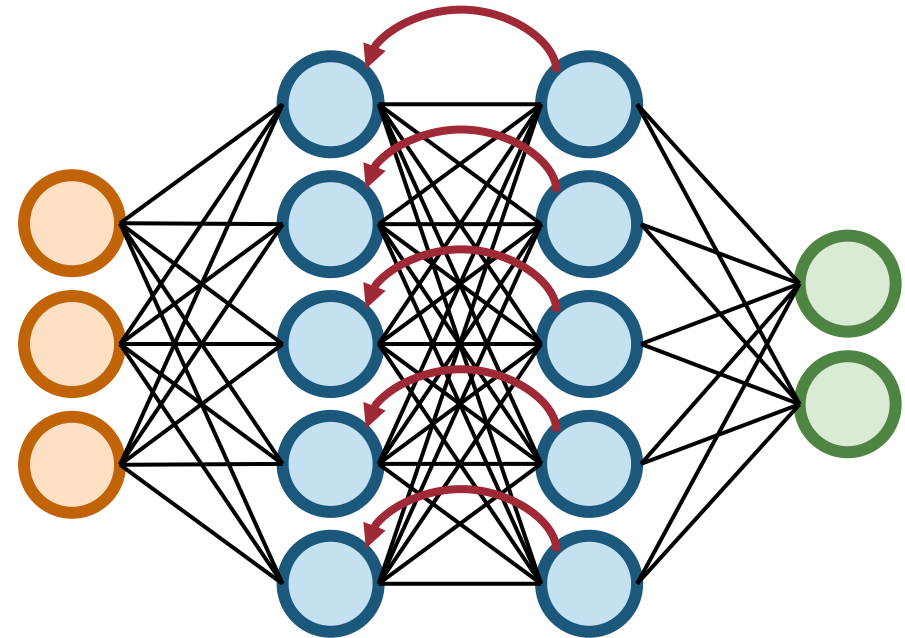
Neural Networks are NOT always Layer-by-Layer

Skip connections



Used in ResNets, U-Nets, diffusion models

Feedback loops

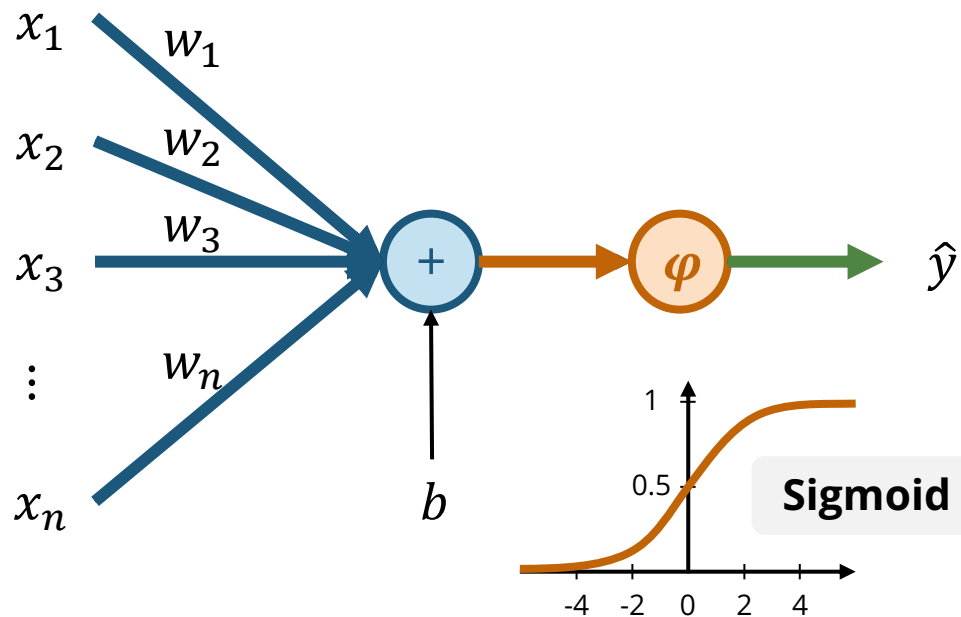


Used in RNNs, LSTMs, GRUs

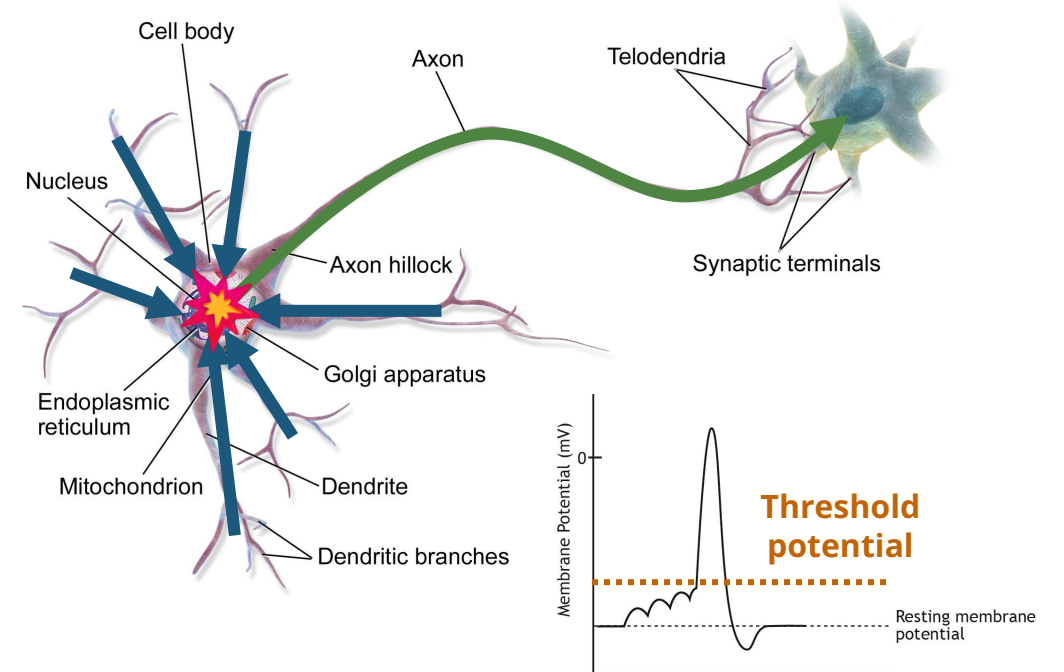
Recap

Artificial vs. Human Neuron

Artificial neuron

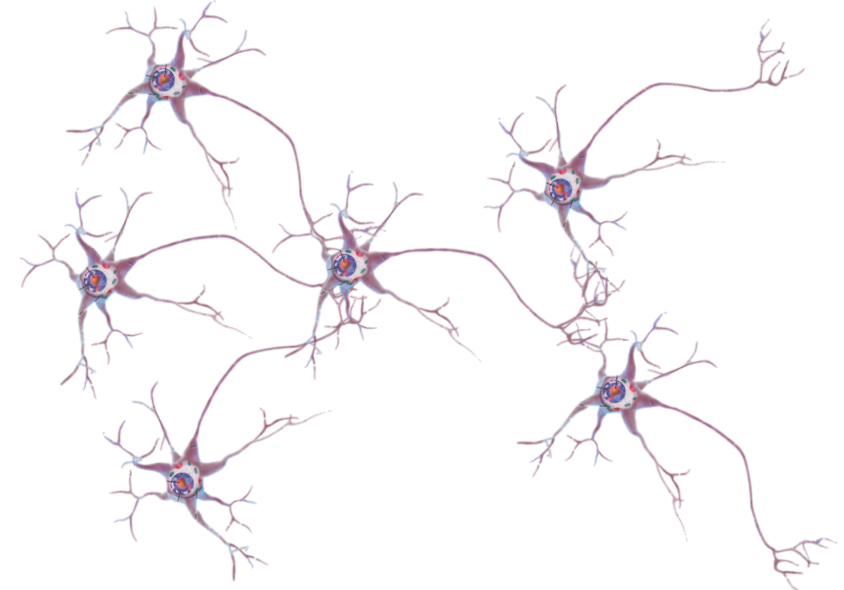
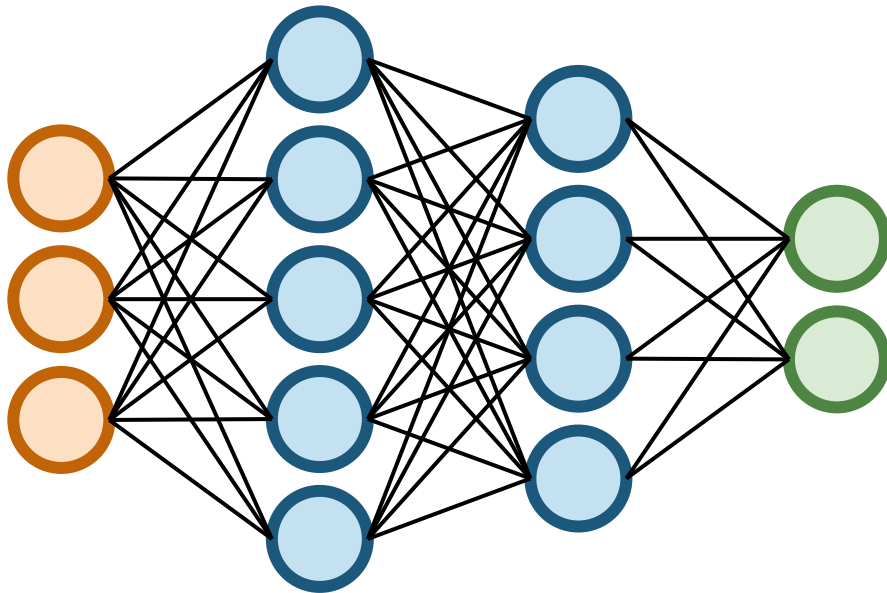


Human neuron

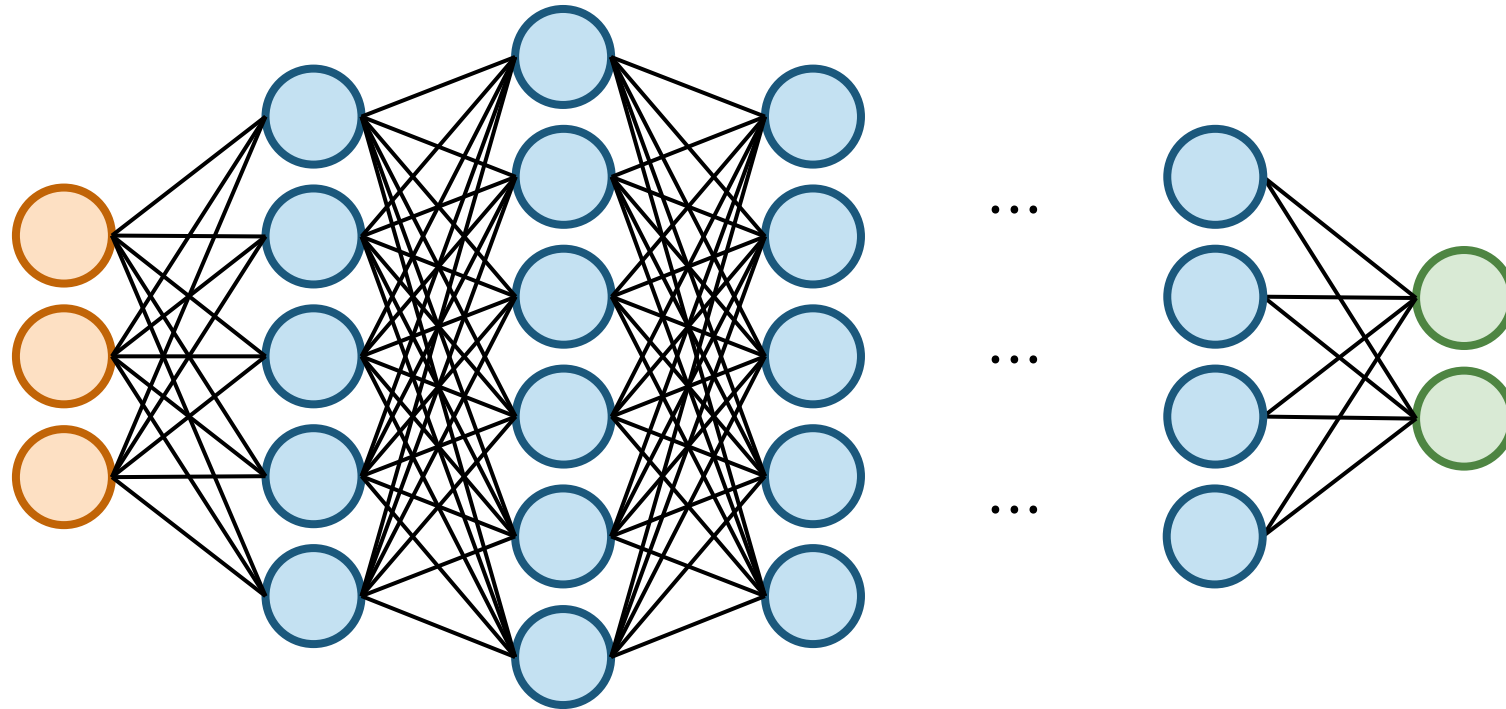


Artificial Neural Networks

- Although inspired by human neural networks, artificial neural networks nowadays *do not work like human brains*
 - Lacking **functional hierarchy, high-level feedback loops, memory module**, etc.
 - Human brains work more like **spiking neural networks** → Efficiency!



Fully Connected Feedforward Network



$$\mathbf{h}_1 = \varphi(W_1 \mathbf{x} + \mathbf{b}_1)$$

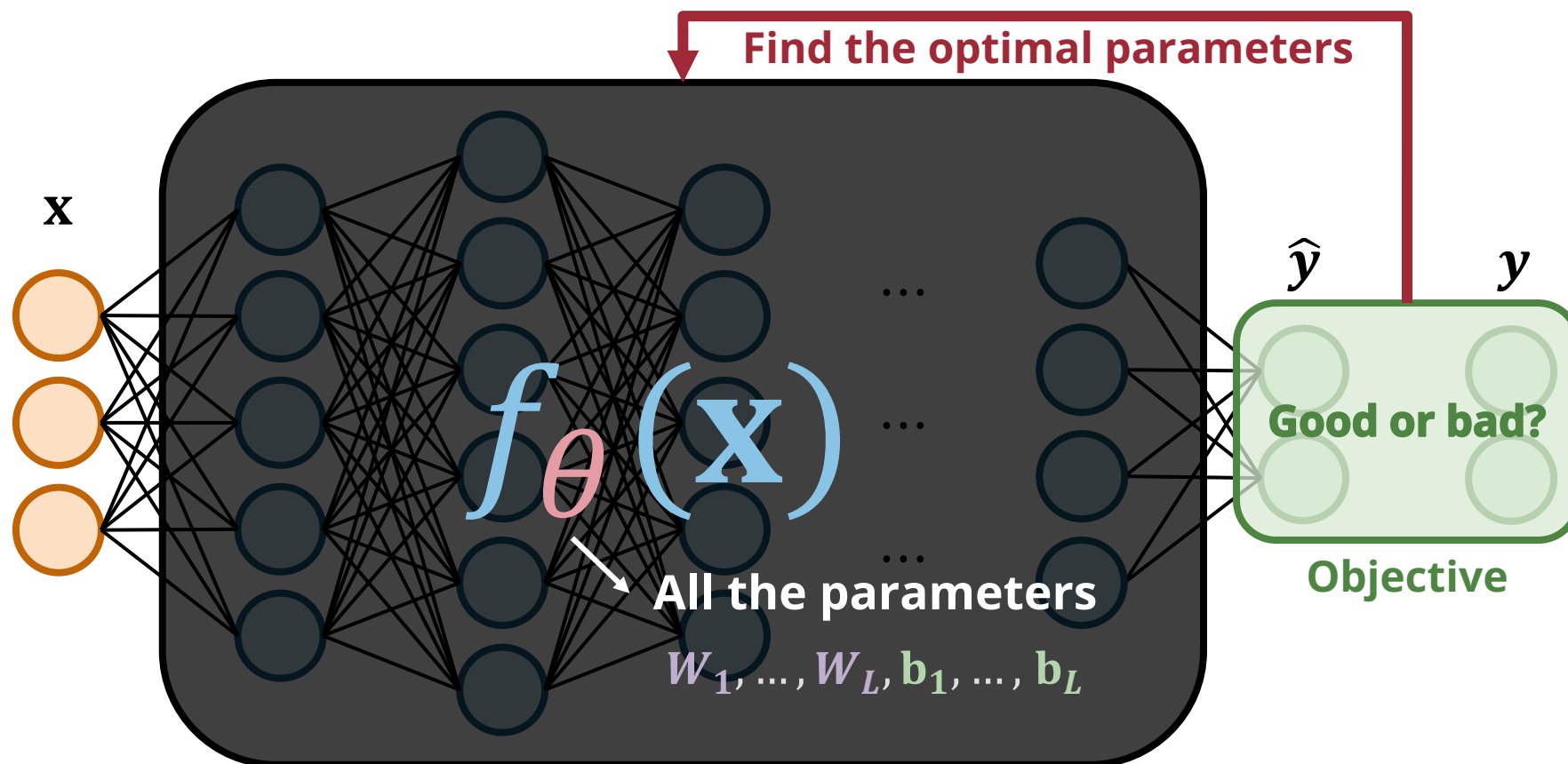
$$\mathbf{h}_2 = \varphi(W_2 \mathbf{h}_1 + \mathbf{b}_2)$$

$$\mathbf{h}_3 = \varphi(W_3 \mathbf{h}_2 + \mathbf{b}_3)$$

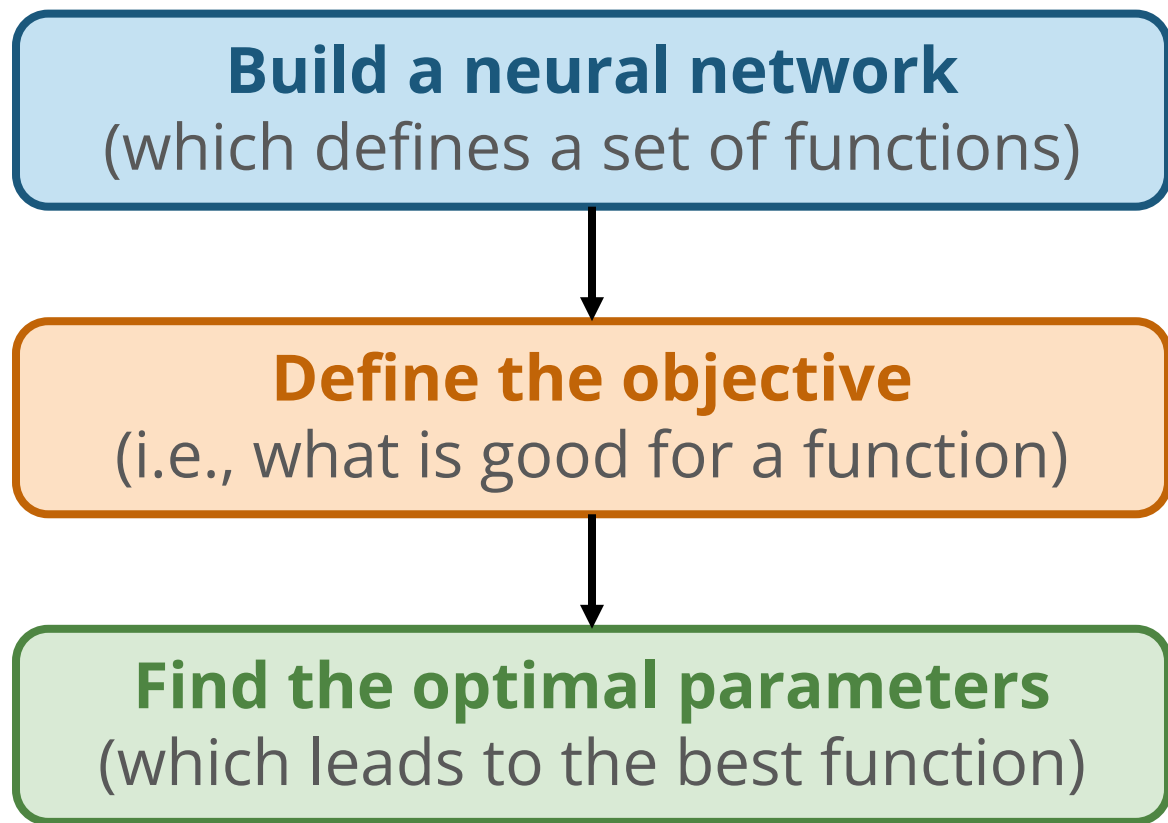
$$\hat{\mathbf{y}} = \varphi(W_L \mathbf{h}_{L-1} + \mathbf{b}_L)$$

Neural Networks are Parameterized Functions

- A neural network represents **a set of functions**

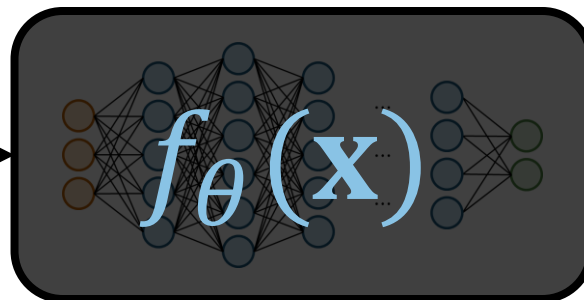


| Training a Neural Network



Regression vs. Classification

Regression

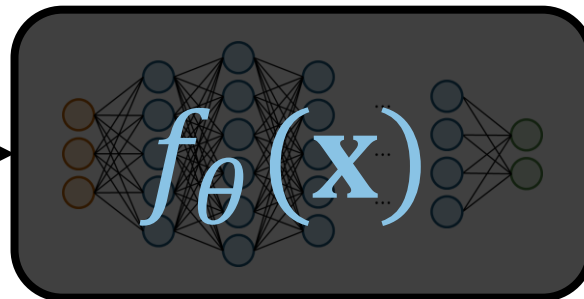


Age

5

Output a number

Classification



Is human?

Yes / No

Output a label

Why do We Need Activation Functions?

With activation functions, a neural network can represent **nonlinear functions**

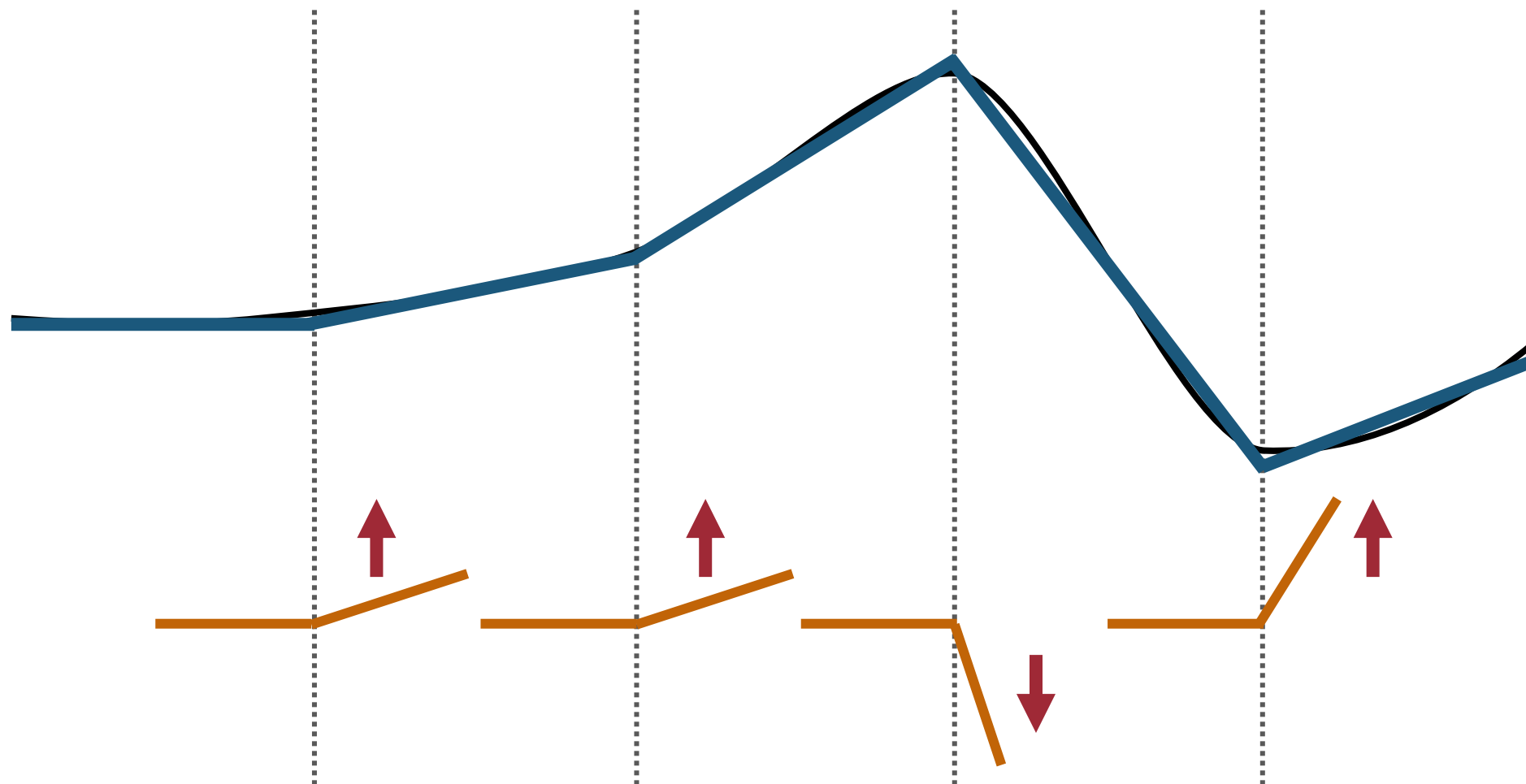
$$\hat{y} = \varphi(W_L (\varphi(W_{L-1} (\varphi(W_{L-2} (\varphi(\cdots \mathbf{x} \cdots)) + \mathbf{b}_{L-2})) + \mathbf{b}_{L-1})) + \mathbf{b}_L)$$



$$\hat{y} = W_L(W_{L-1}(W_{L-2}(\cdots \mathbf{x} \cdots) + \mathbf{b}_{L-2}) + \mathbf{b}_{L-1}) + \mathbf{b}_L$$

Without activation functions, a neural network can only represent **linear functions**

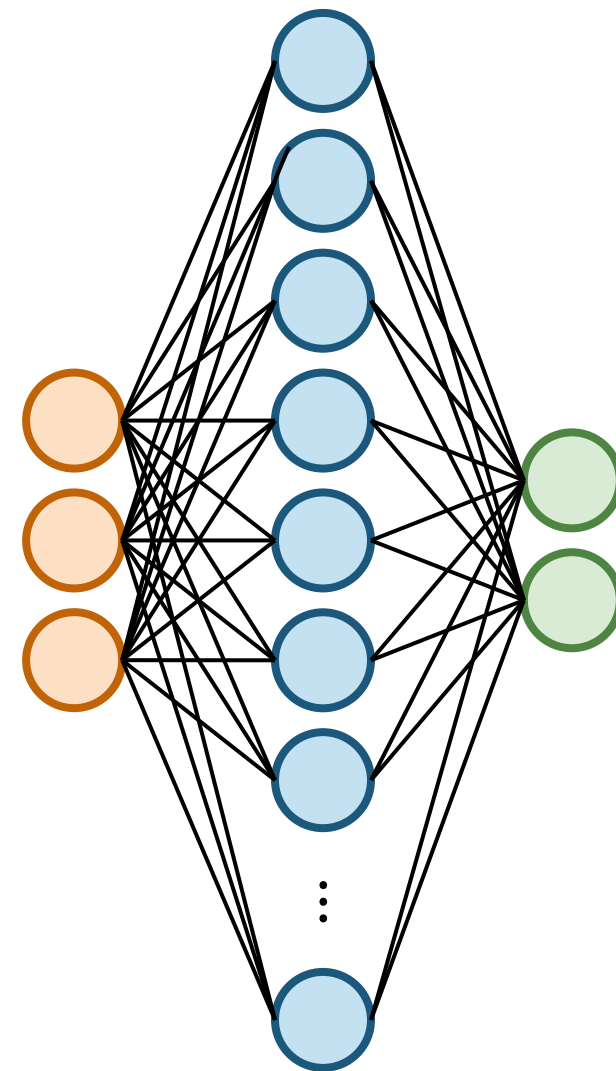
ReLUs & Piecewise Linear Functions



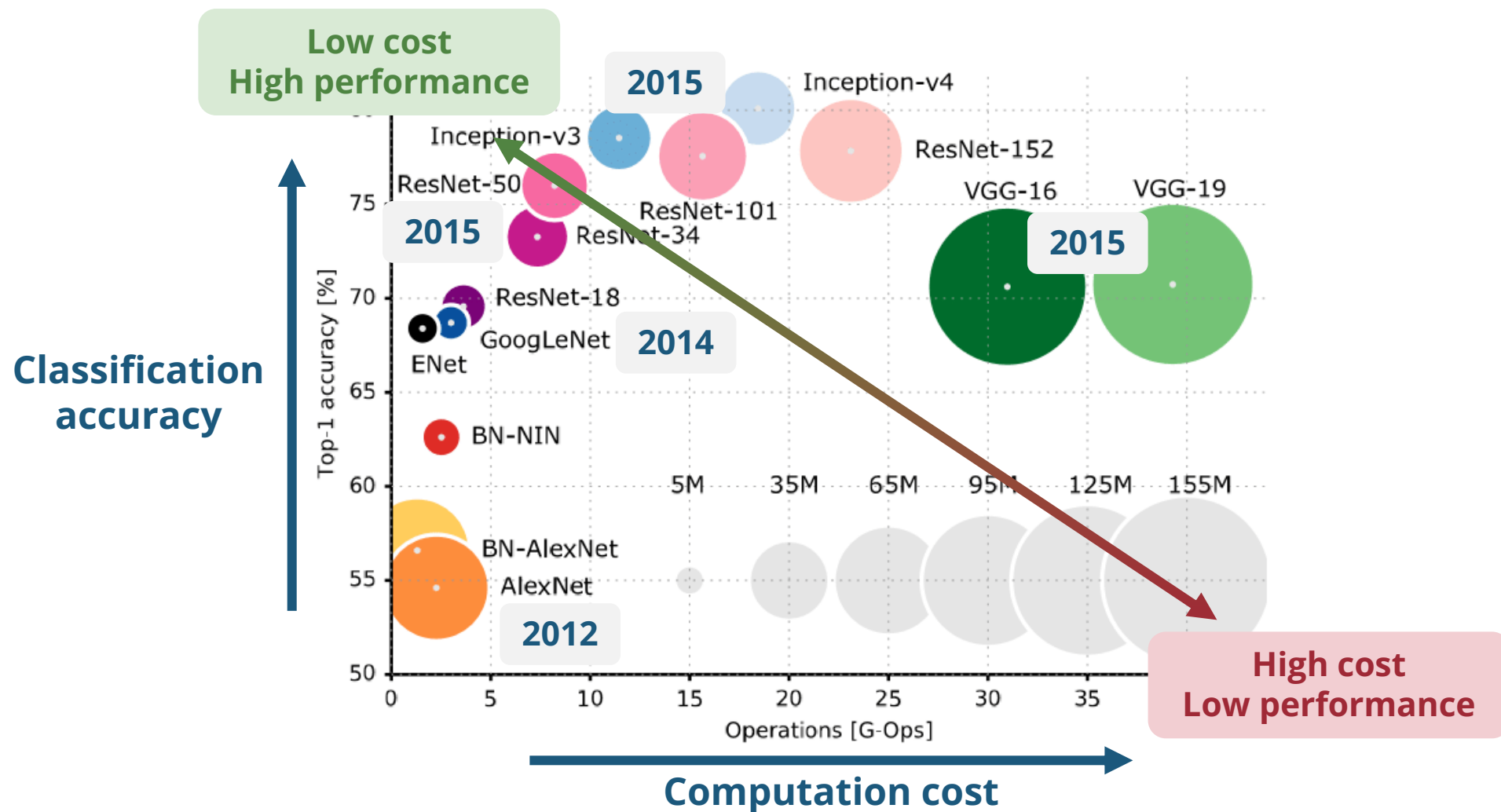
Universal Approximation Theorem

- A neural network with **one hidden layer** can **approximate any continuous function** given **sufficient hidden neurons** and **appropriate activation functions**
 - Sigmoid, ReLUs are good activation functions

Then why do we want to go deep?



Computation Cost vs. Classification Accuracy



(Source: Canziani et al., 2016)

Next Lecture

Deep Learning Fundamentals II

