PAT 463/563 (Fall 2025)

Music & AI

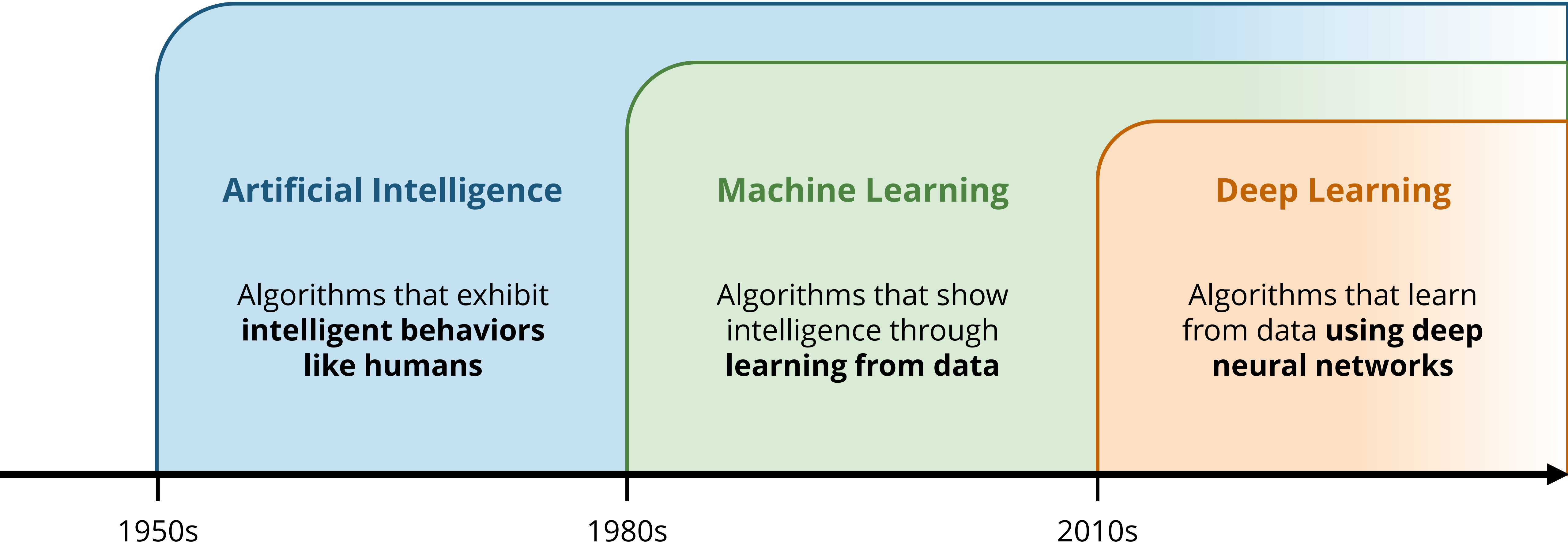
Lecture 6: Deep Learning Fundamentals

Instructor: Hao-Wen Dong

UNIVERSITY OF MICHIGAN

What is Deep Learning?

AI vs ML vs DL



Artificial Intelligence
Algorithms that exhibit **intelligent behaviors like humans**
Machine Learning
Algorithms that show intelligence through **learning from data**
Deep Learning
Algorithms that learn from data **using deep neural networks**
1950s
1980s
2010s

Components of a Machine Learning Model

Optimization

Defining inputs & outputs

Improve on task T,

with respect to performance metric P,

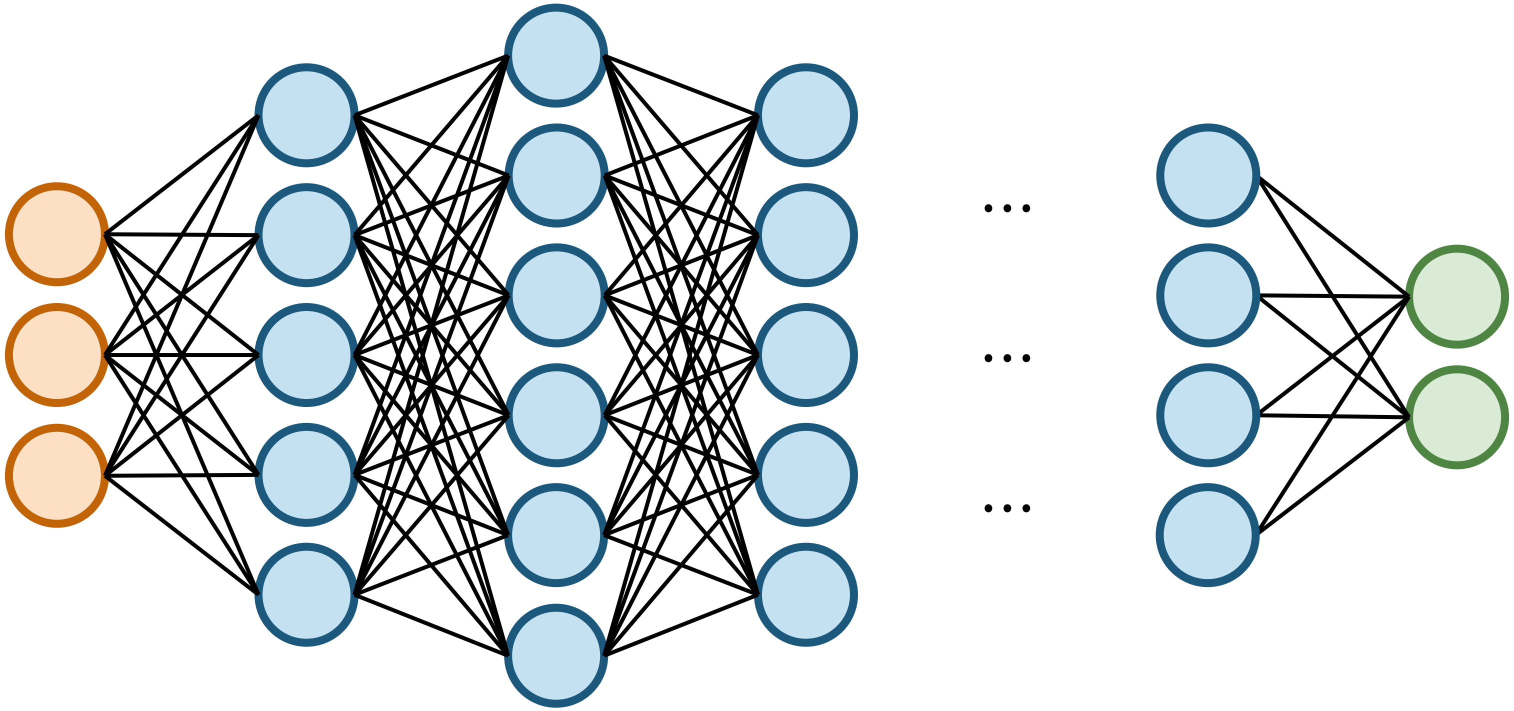
Objective function (loss function)

based on experience E

Training data

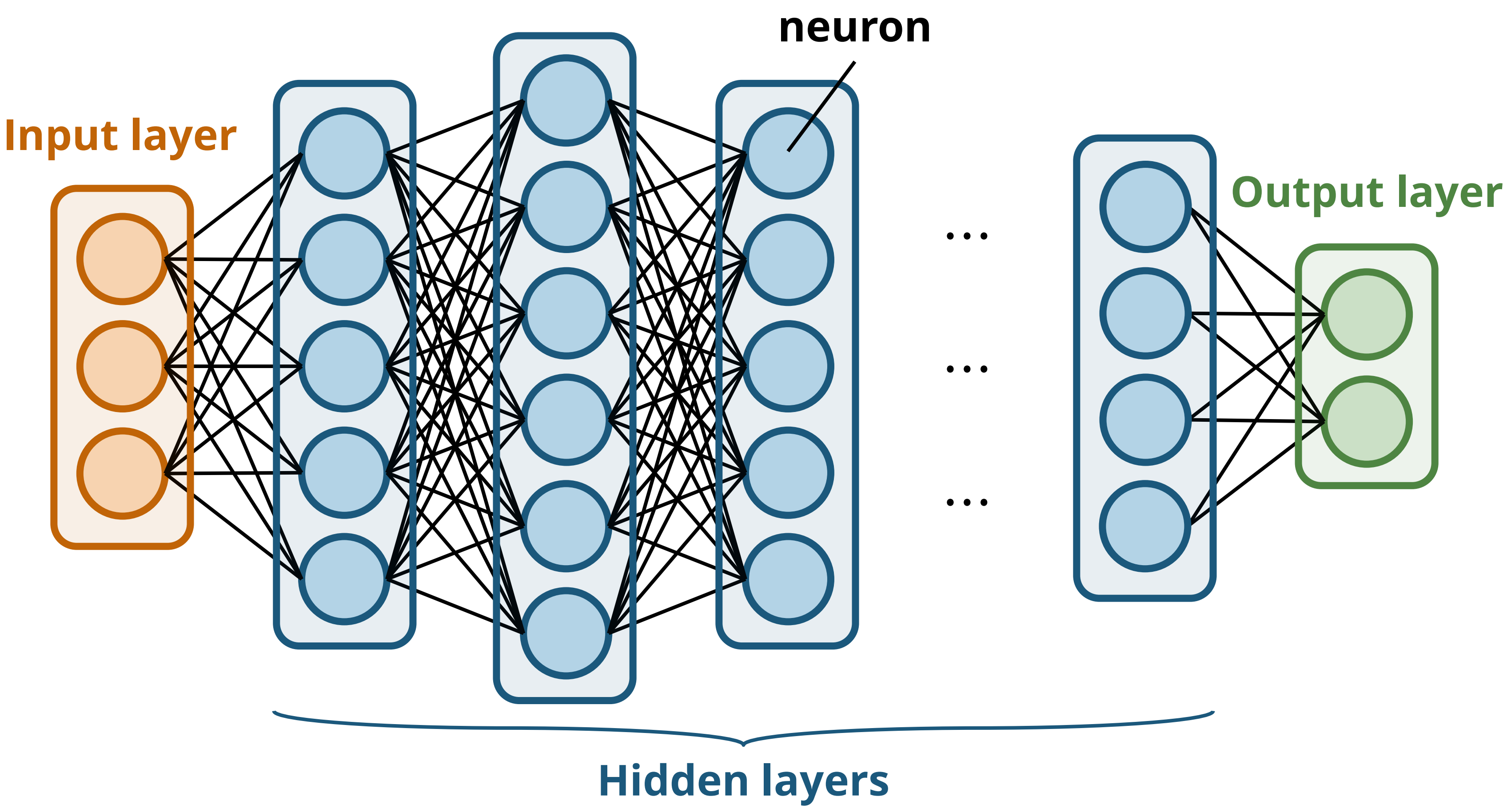
What is Deep Learning?

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



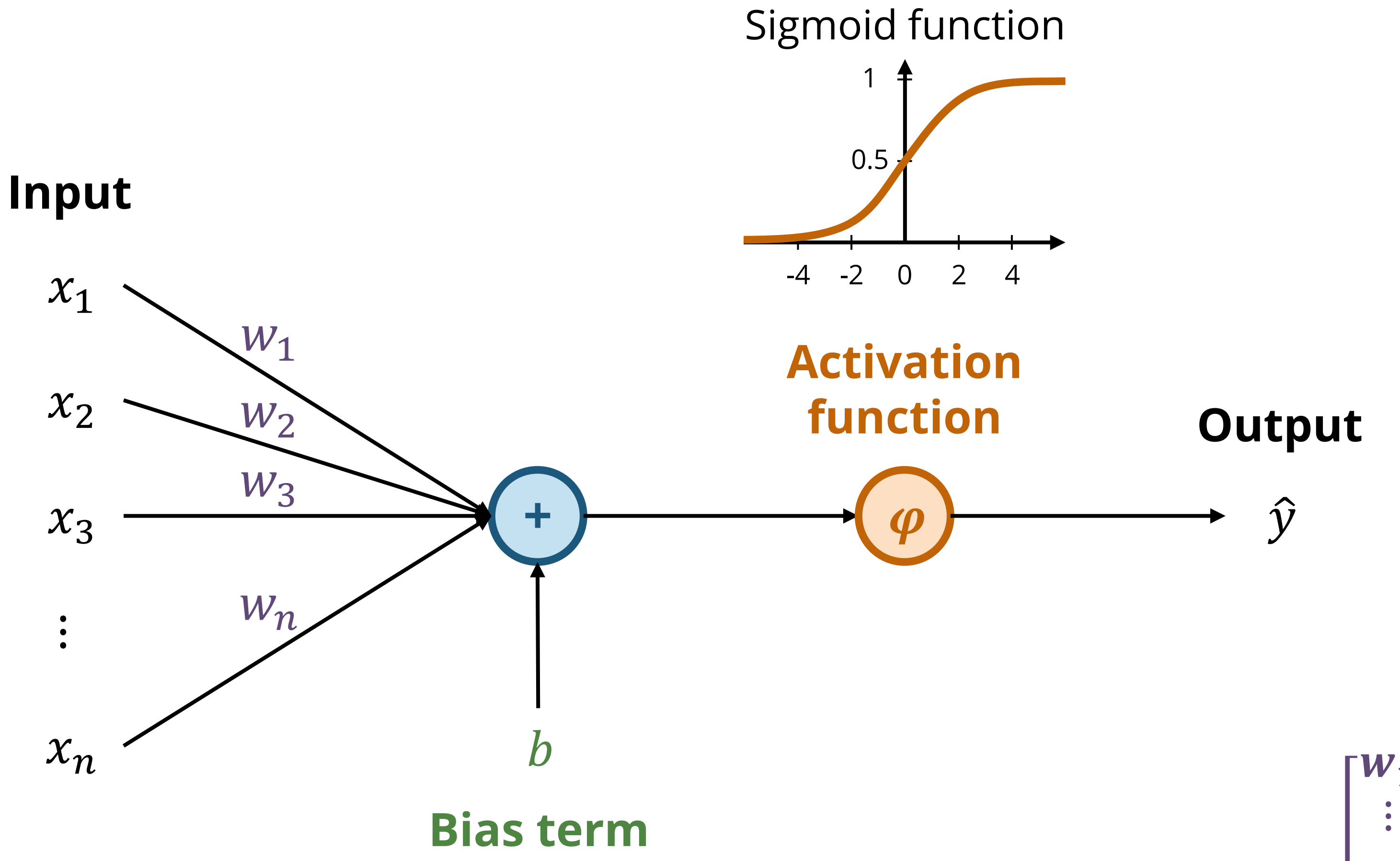
What is Deep Learning?

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



Neural Networks

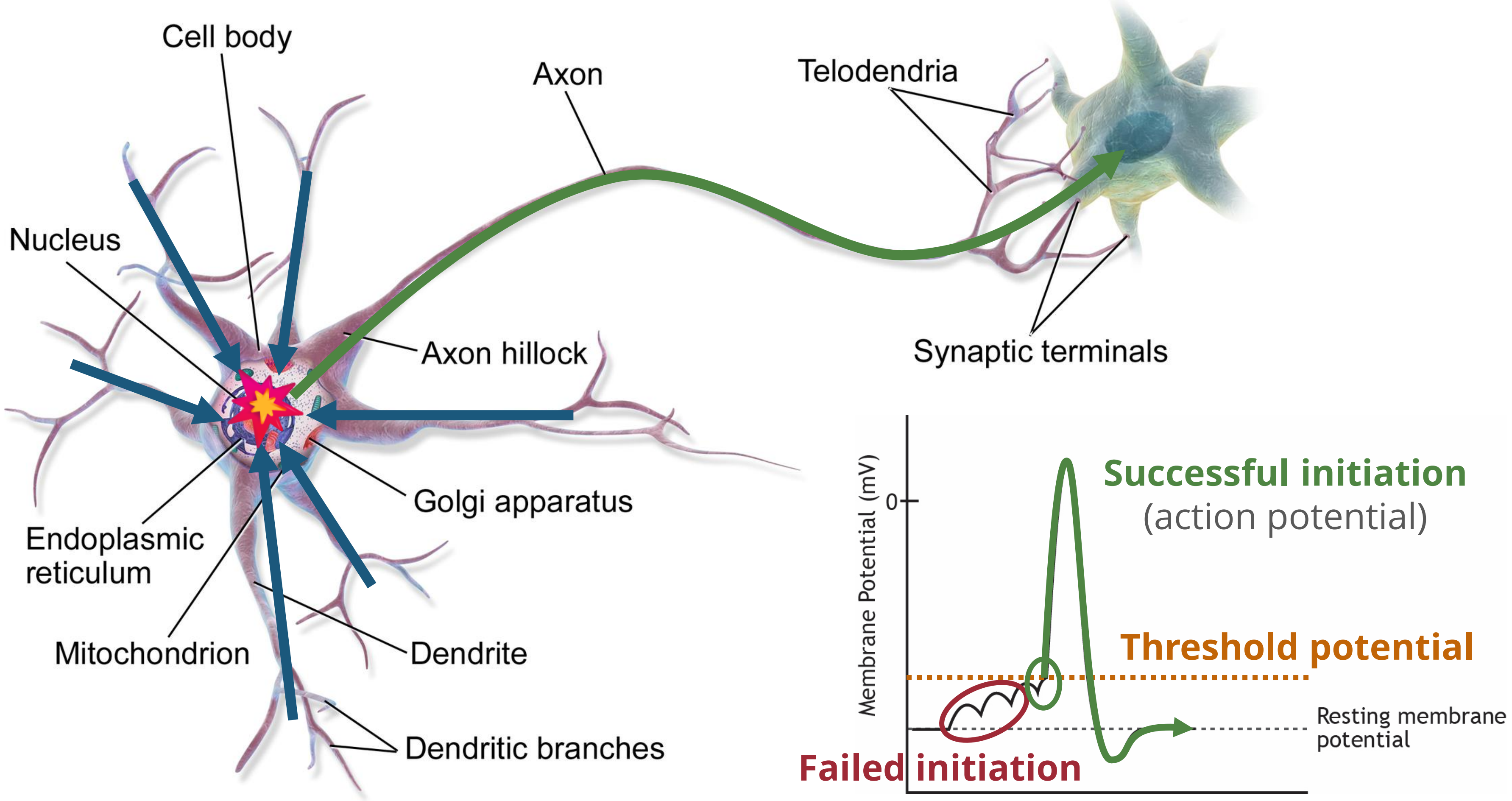
Inside a Neuron



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

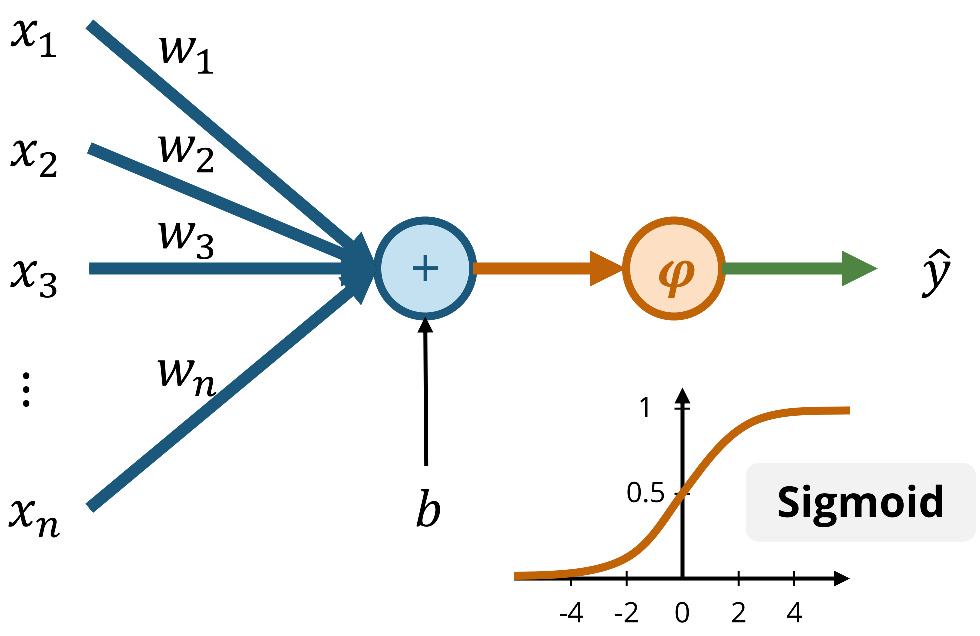
$$\mathbf{w} = \begin{bmatrix} \boldsymbol{w_1} \\ \vdots \\ \boldsymbol{w_n} \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} \boldsymbol{x_1} \\ \vdots \\ \boldsymbol{x_n} \end{bmatrix}$$

Human Neuron

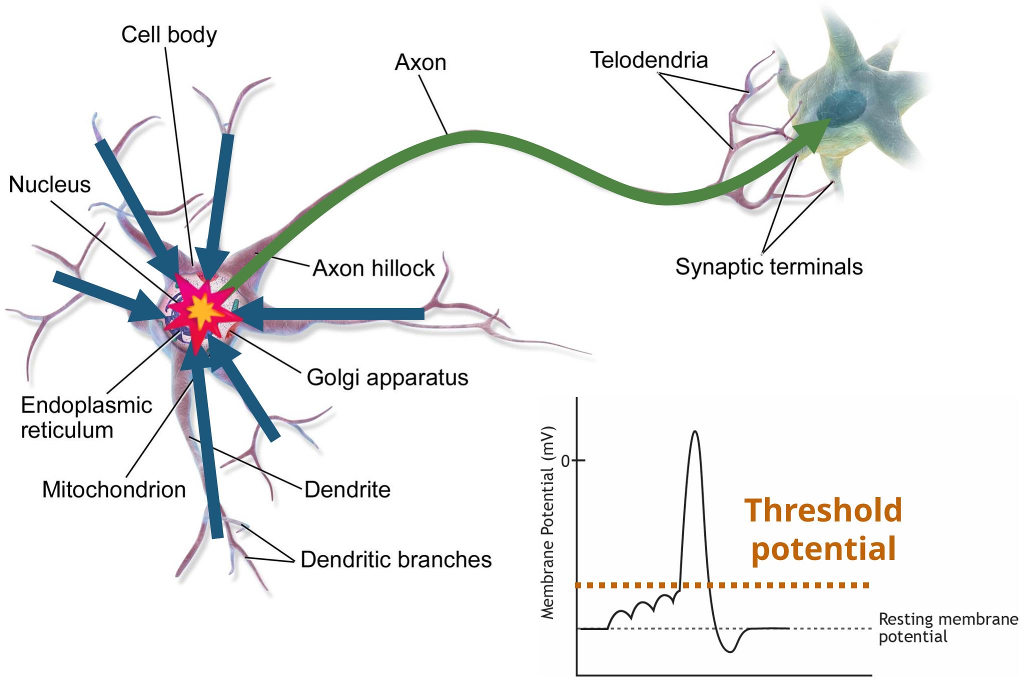


Artificial vs Human Neuron

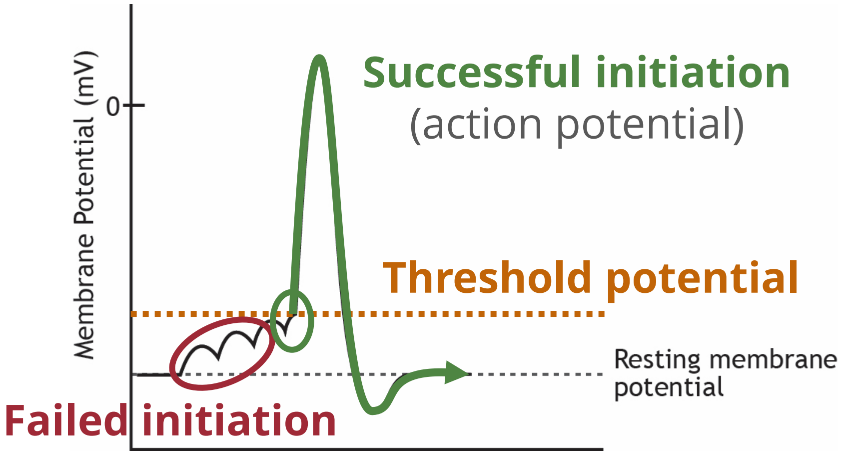
Artificial neuron



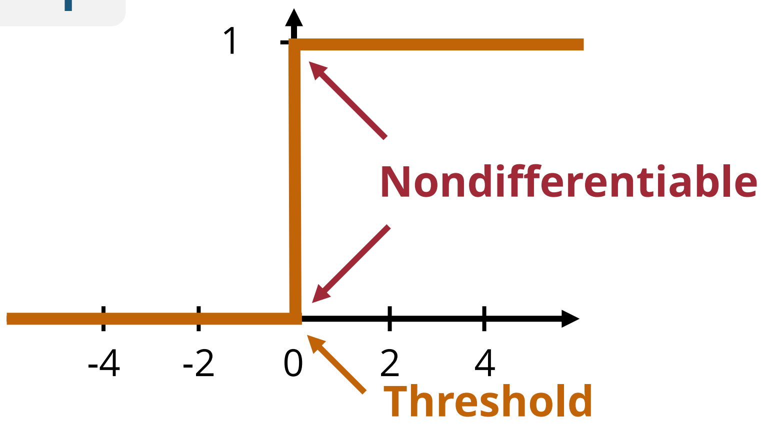
Human neuron



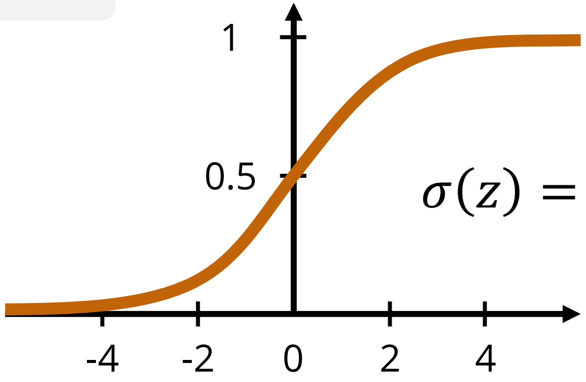
Why Sigmoid?



Unit step



Sigmoid



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

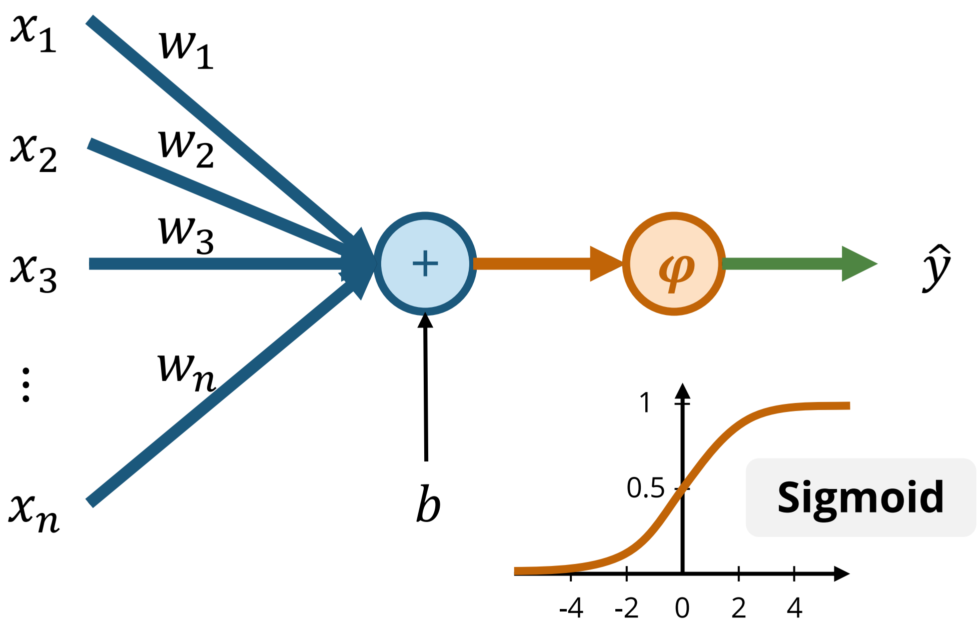
Why Bias Term?

- Allow nonzero outputs when all inputs are zero

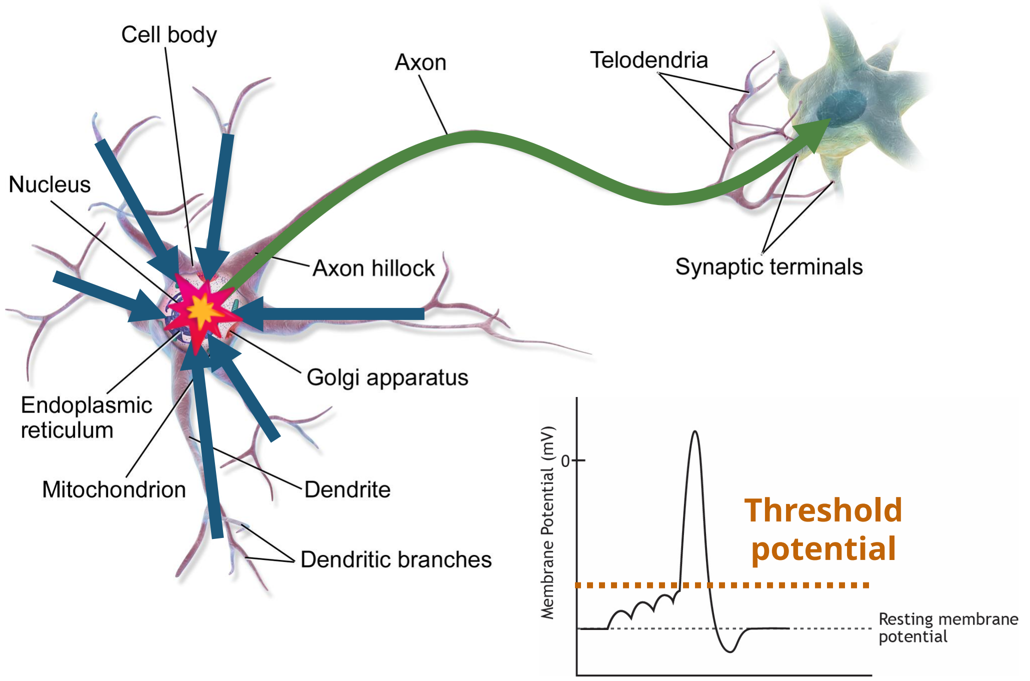
$$\hat{\boldsymbol{y}} = \boldsymbol{\varphi}(w_1 \cancelto{0}{x_1} + w_2 \cancelto{0}{x_2} + \cdots + w_n \cancelto{0}{x_n} + b) = \boldsymbol{\varphi}(b)$$

Artificial vs Human Neuron

Artificial neuron

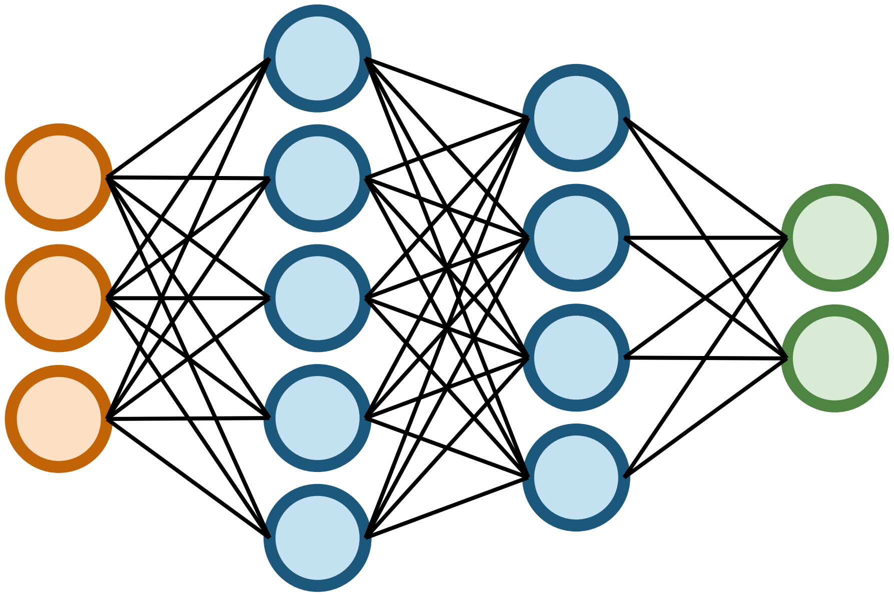


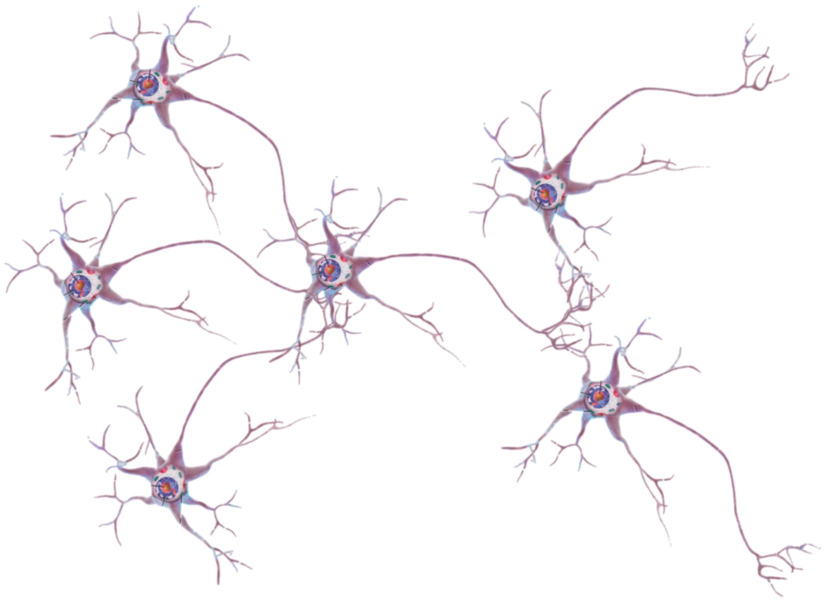
Human neuron



Artificial Neural Networks

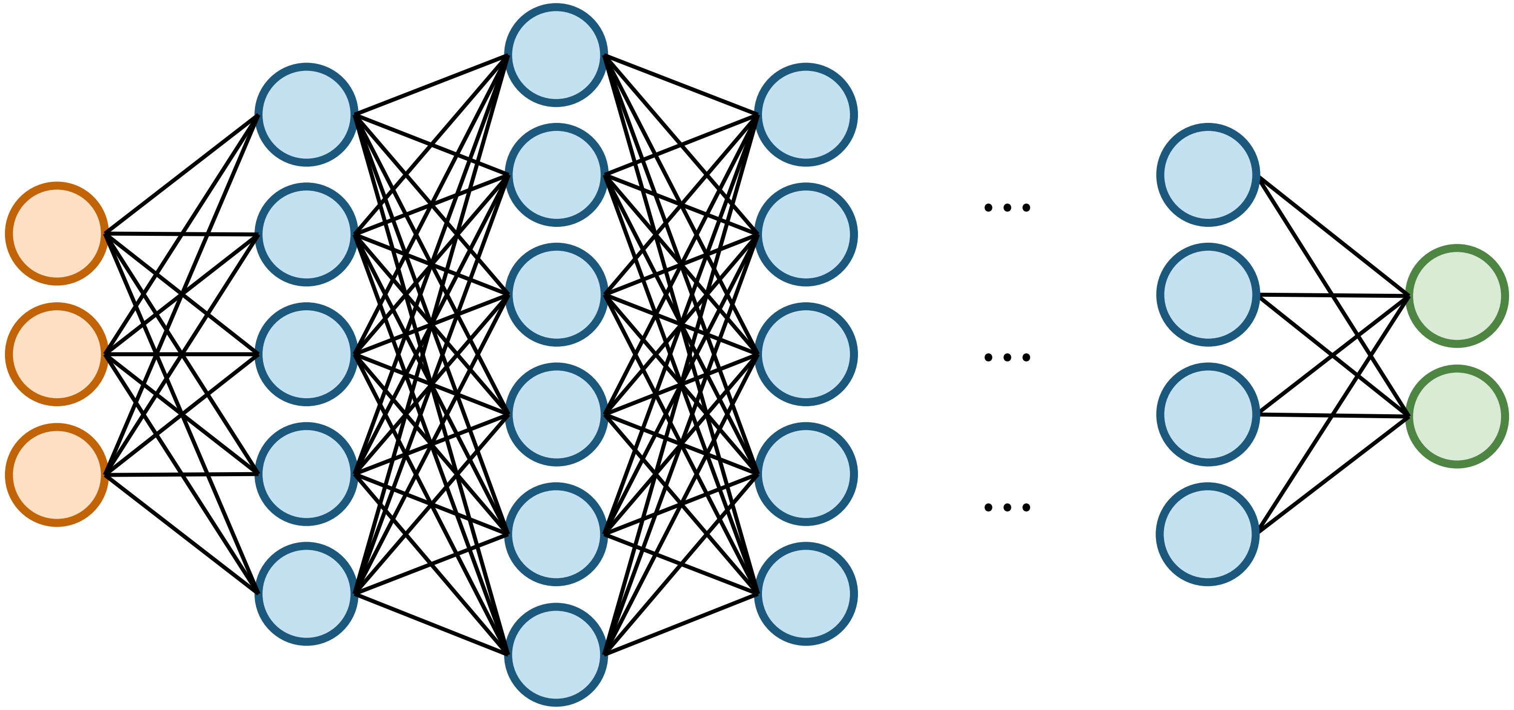
- Although inspired by human neural networks, artificial neural networks nowadays *do not work like human brains*
  - Lacking **functional hierarchy**, **high-level feedback loops**, **memory module**, etc.
  - Human brains work more like **spiking neural networks** → Efficiency!





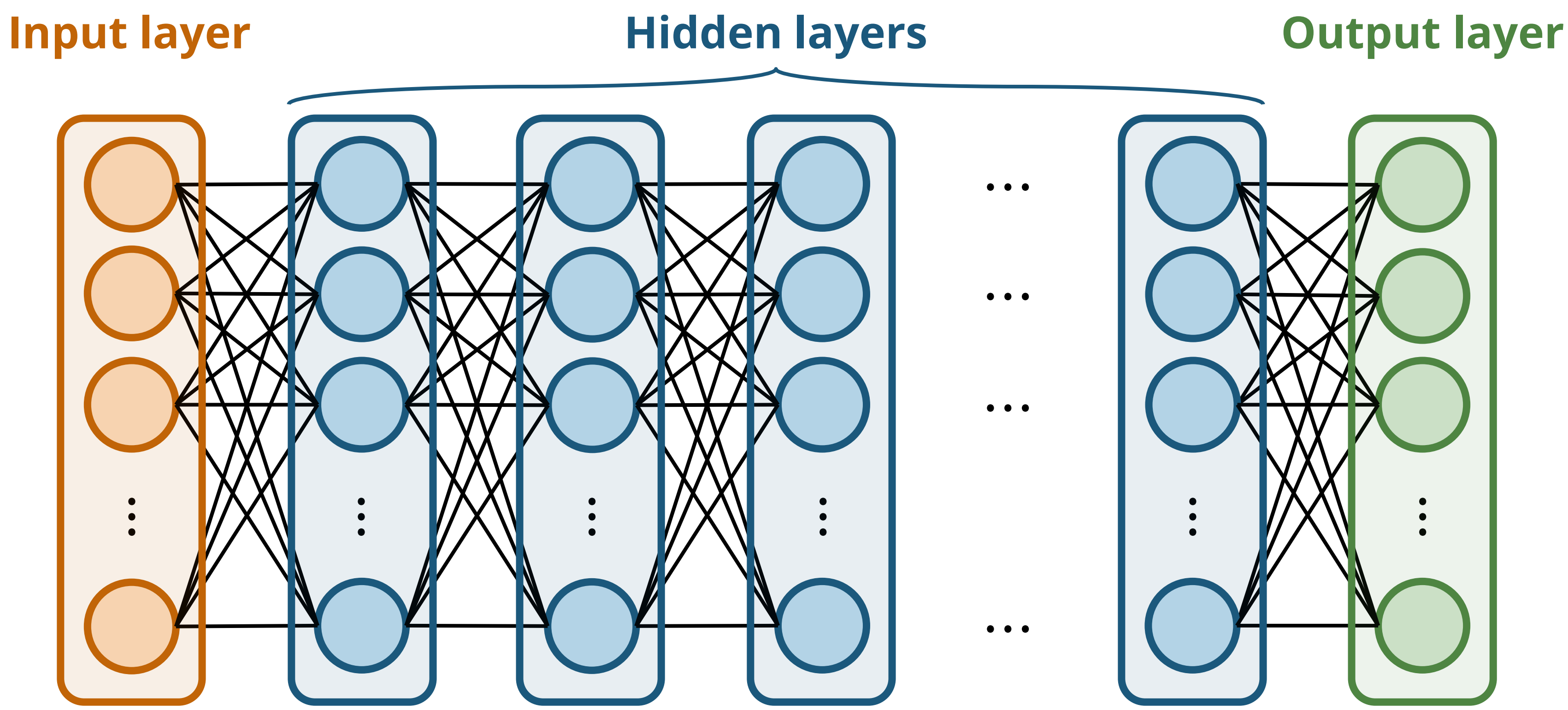
Fully Connected Feedforward Network

- Most basic form of deep neural networks



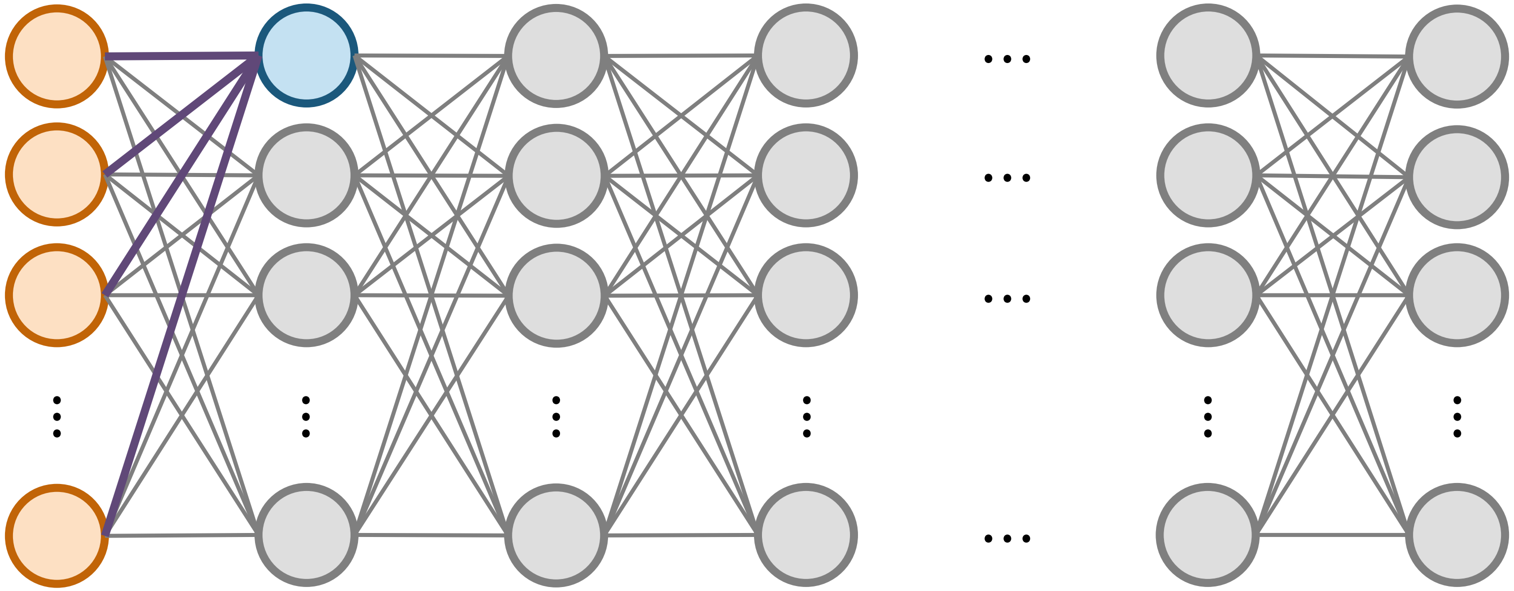
Math Formulation

Math Formulation



Math Formulation

$$h_1 = \varphi(\mathbf{w}_1 \cdot \mathbf{x} + b_1)$$

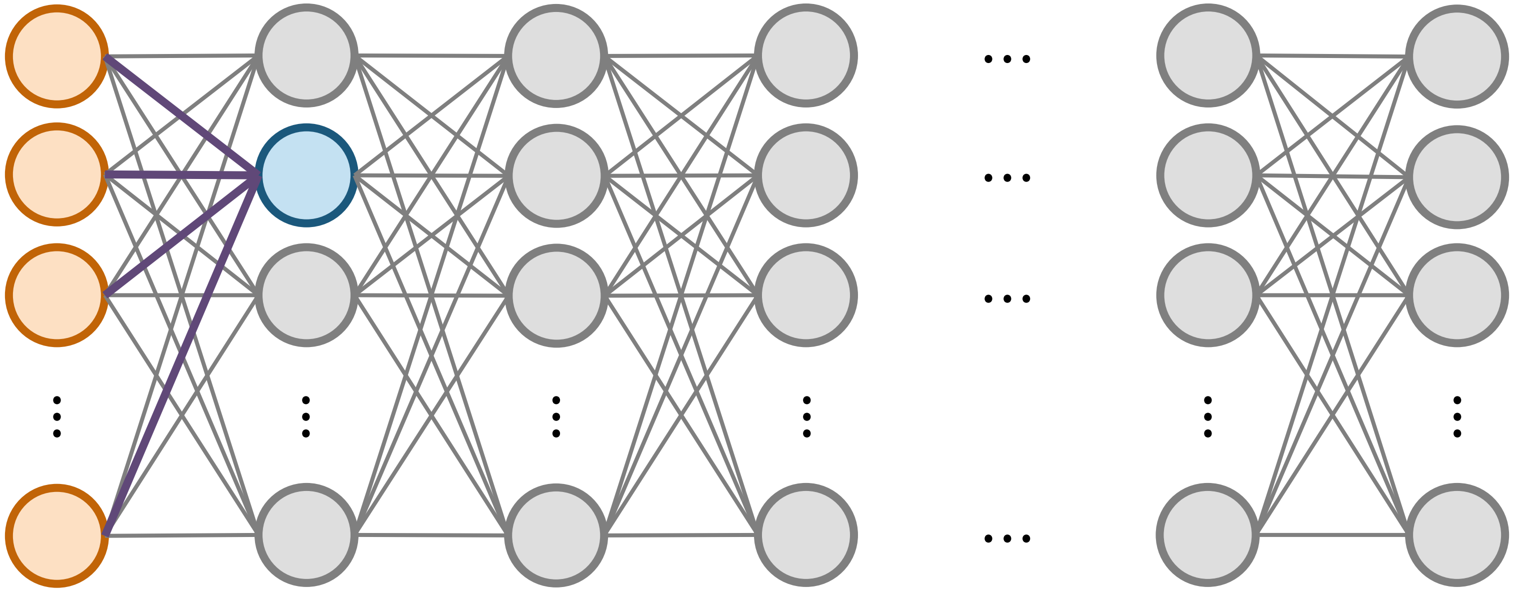


$$\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \begin{bmatrix} w_{1,1} \\ \vdots \\ w_{1,n} \end{bmatrix}$$

$$\mathbf{x} \quad \mathbf{w}_1$$

Math Formulation

$$h_2 = \varphi(\mathbf{w}_2 \cdot \mathbf{x} + b_2)$$

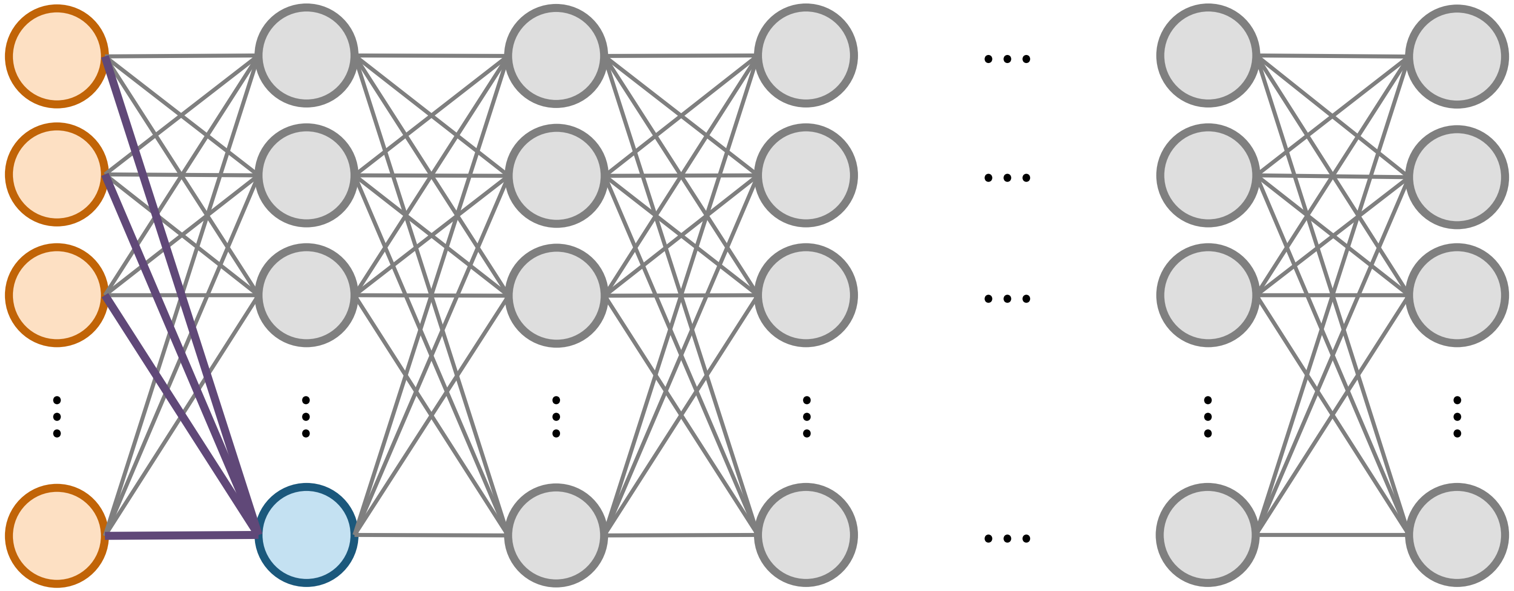


$$\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \begin{bmatrix} w_{2,1} \\ \vdots \\ w_{2,n} \end{bmatrix}$$

$\mathbf{x}$ $\mathbf{w}_2$

Math Formulation

$$h_n = \varphi(\mathbf{w_n} \cdot \mathbf{x} + b_n)$$

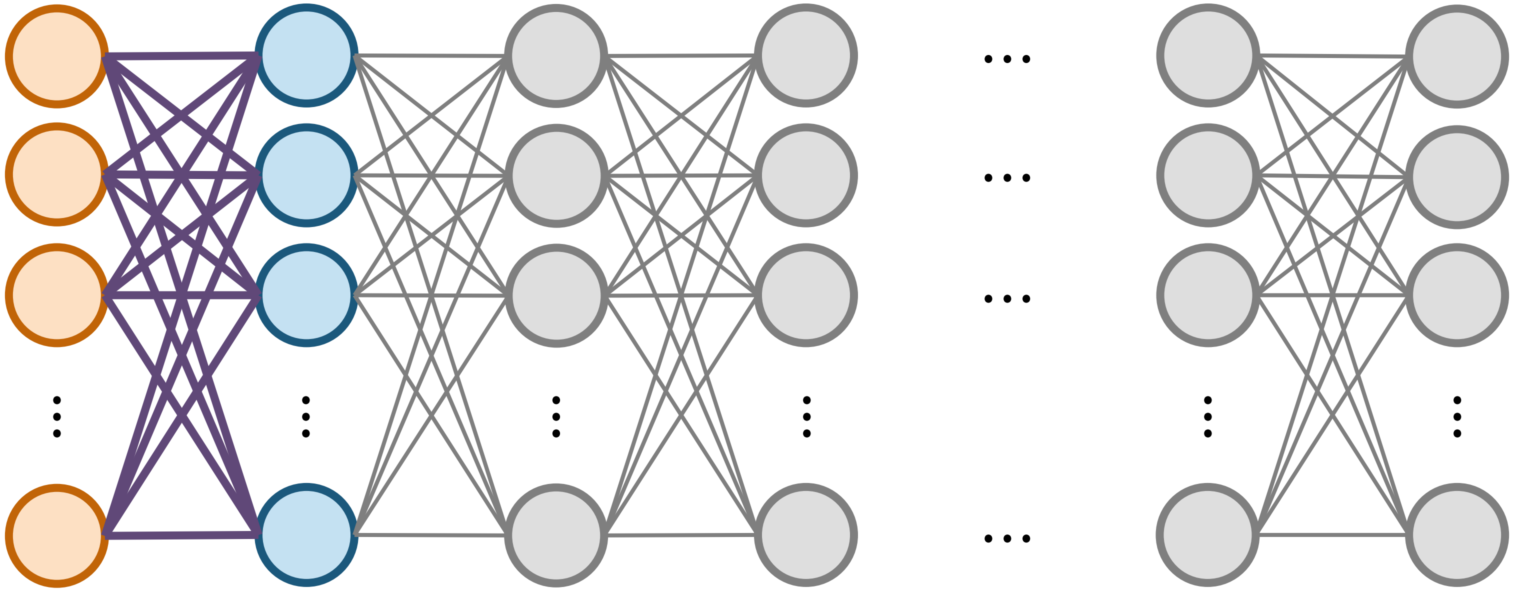


$$\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \begin{bmatrix} w_{n,1} \\ \vdots \\ w_{n,n} \end{bmatrix}$$

$\mathbf{x}$ $\mathbf{w_n}$

Math Formulation

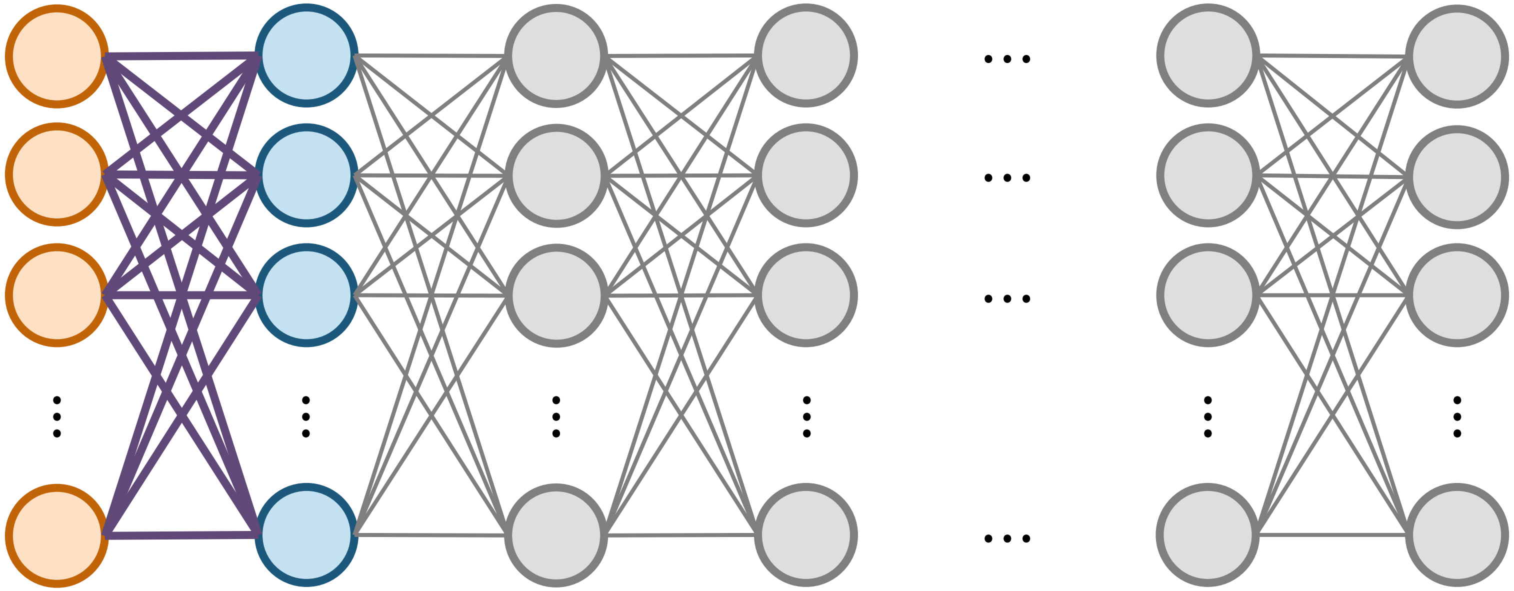
$$\mathbf{h} = \varphi(W\mathbf{x} + \mathbf{b})$$



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

Math Formulation

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

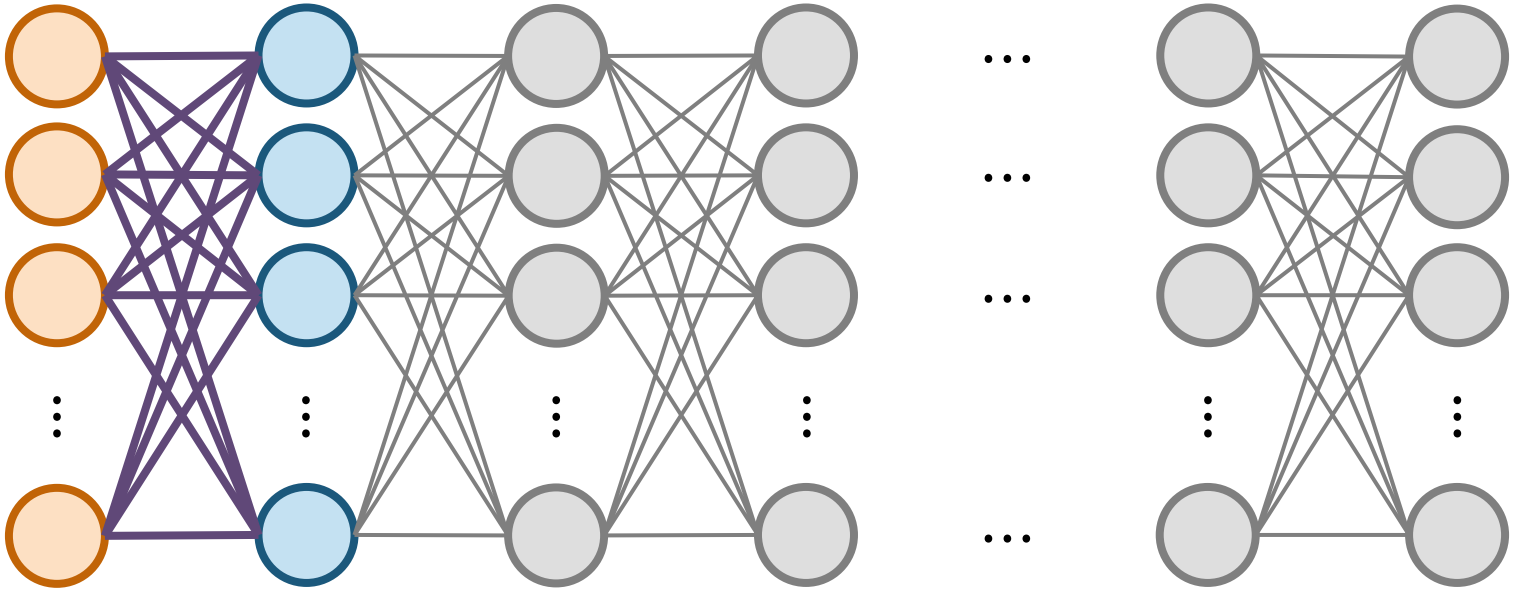


$$\underbrace{\begin{bmatrix} w_{1,1} & \cdots & w_{1,n} \\ \vdots & \ddots & \vdots \\ w_{n,1} & \cdots & w_{n,n} \end{bmatrix}}_{W} \underbrace{\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}}_{\mathbf{b}}$$

$$h_1 = \varphi(\mathbf{w_1} \cdot \mathbf{x} + b_1)$$

Math Formulation

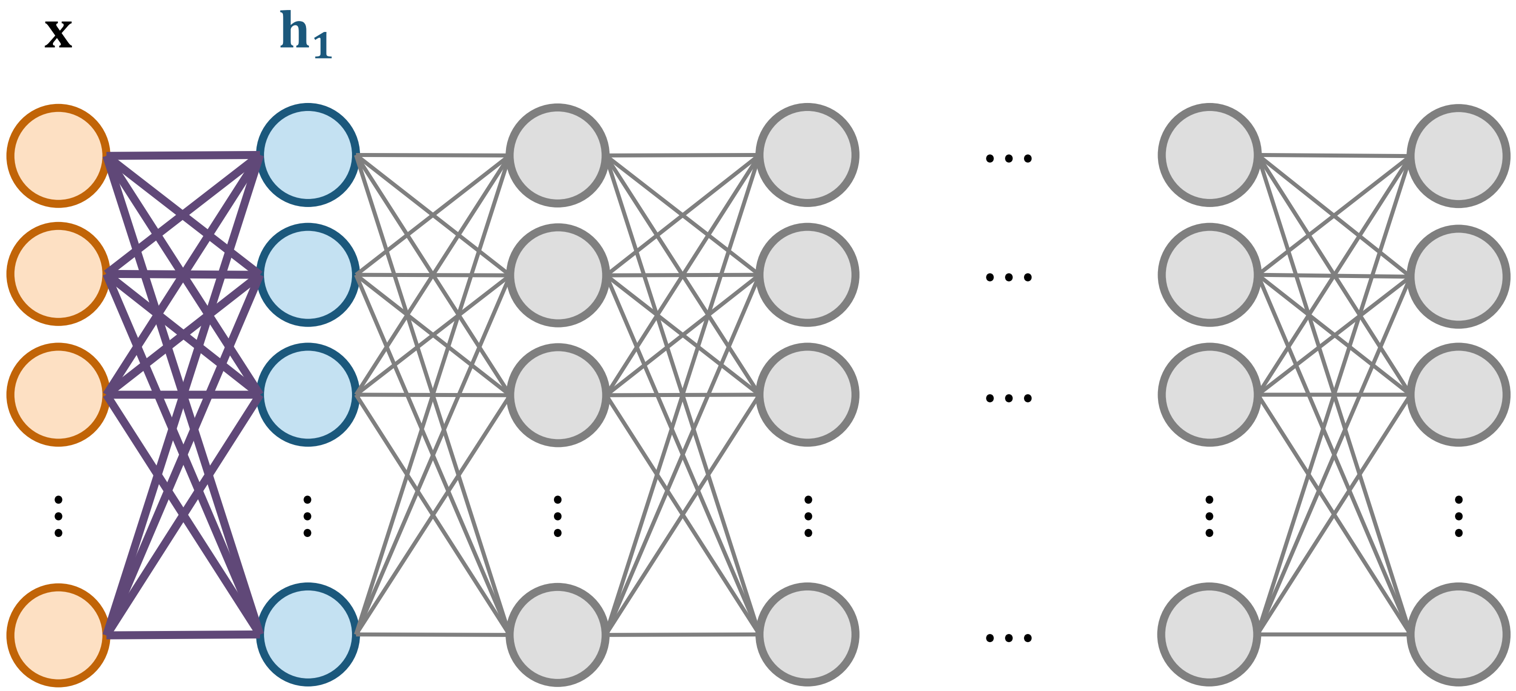
$$\mathbf{h} = \varphi(W\mathbf{x} + \mathbf{b})$$



$$\underbrace{\begin{bmatrix} w_{1,1} & \cdots & w_{1,n} \\ \vdots & \ddots & \vdots \\ w_{n,1} & \cdots & w_{n,n} \end{bmatrix}}_{W} \underbrace{\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} b_1 \\ \vdots \\ \boxed{b_n} \end{bmatrix}}_{\mathbf{b}}$$

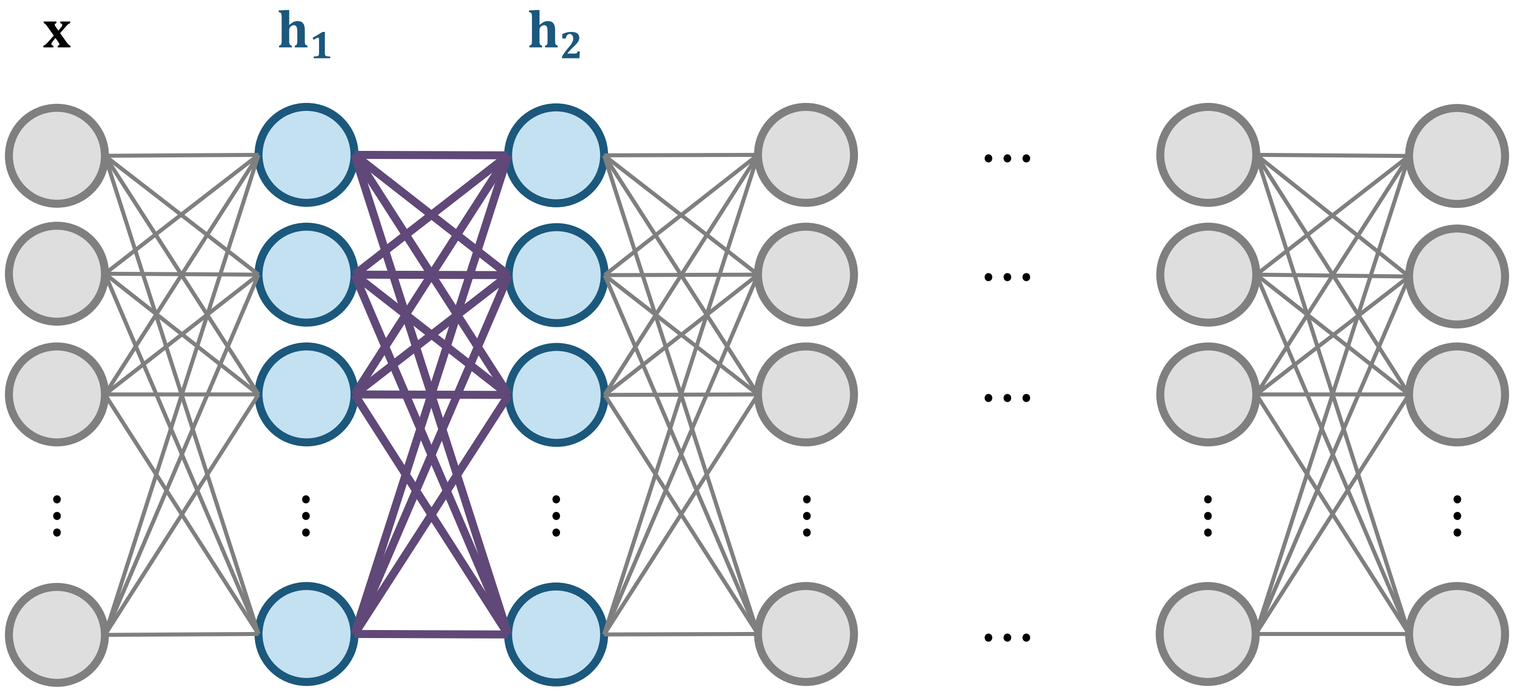
$$h_n = \varphi(\mathbf{w}_n \cdot \mathbf{x} + b_n)$$

Math Formulation



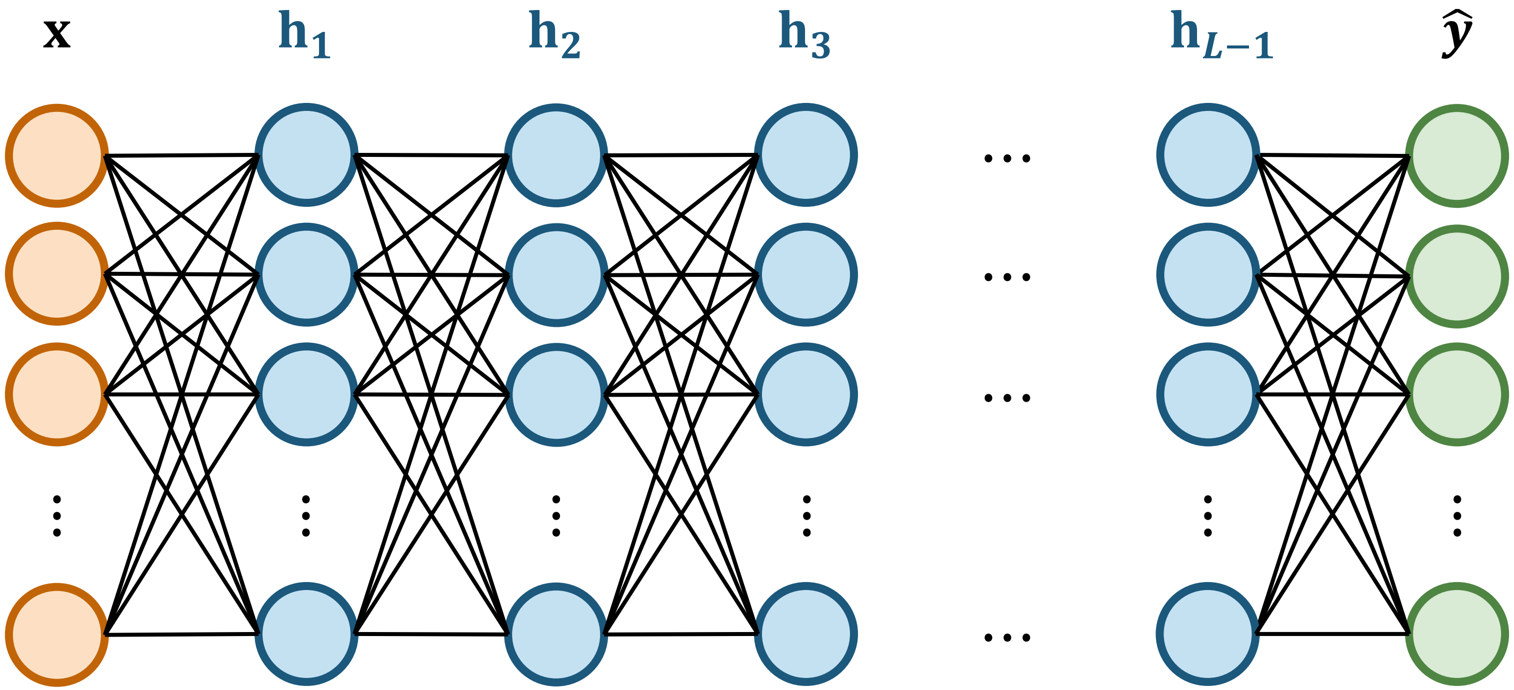
$$\mathbf{h}_1 = \varphi(W_1\mathbf{x} + \mathbf{b}_1)$$

Math Formulation



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

Math Formulation



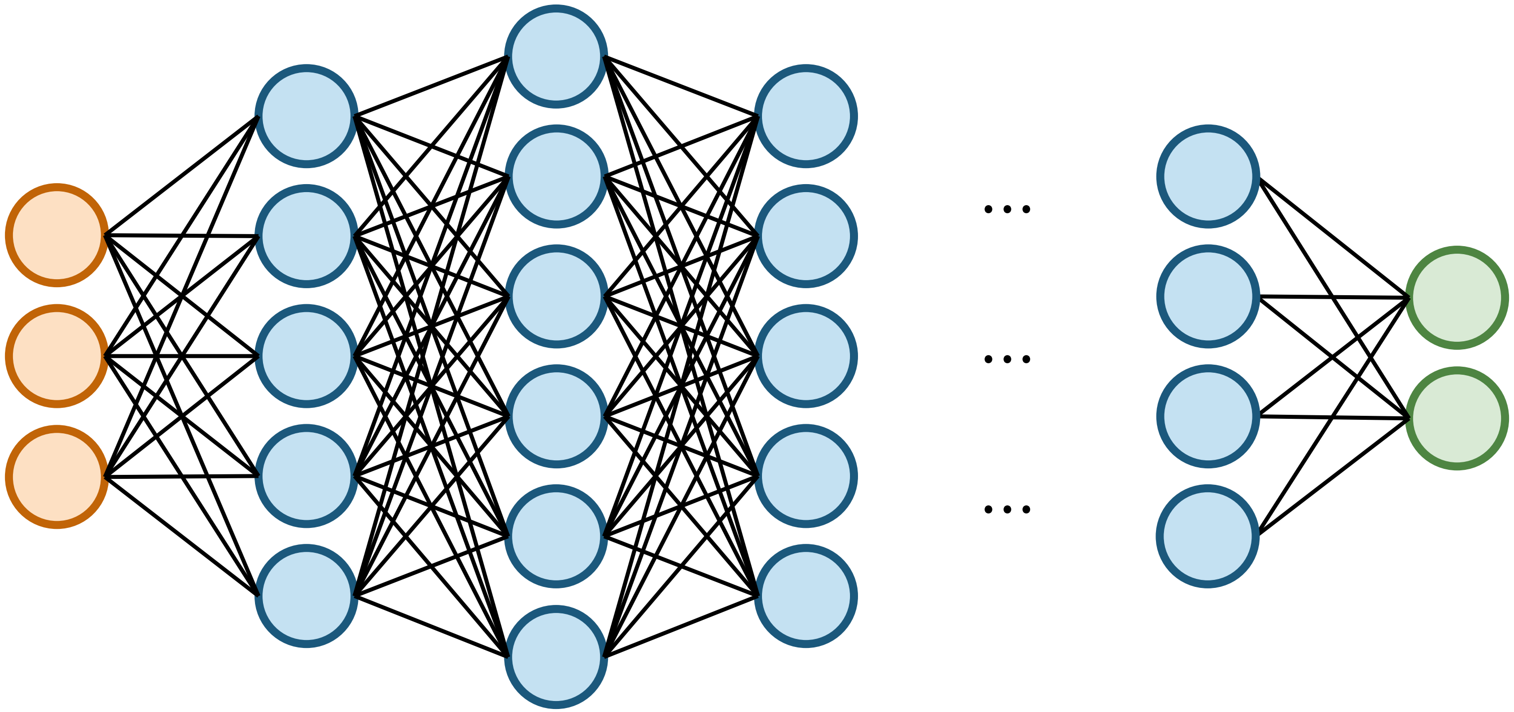
$$\mathbf{h}_1 = \varphi(W_1\mathbf{x} + \mathbf{b}_1)$$

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

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

$$\mathbf{h}_3 = \varphi(W_2\mathbf{h}_2 + \mathbf{b}_2)$$

Fully Connected Feedforward Network



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

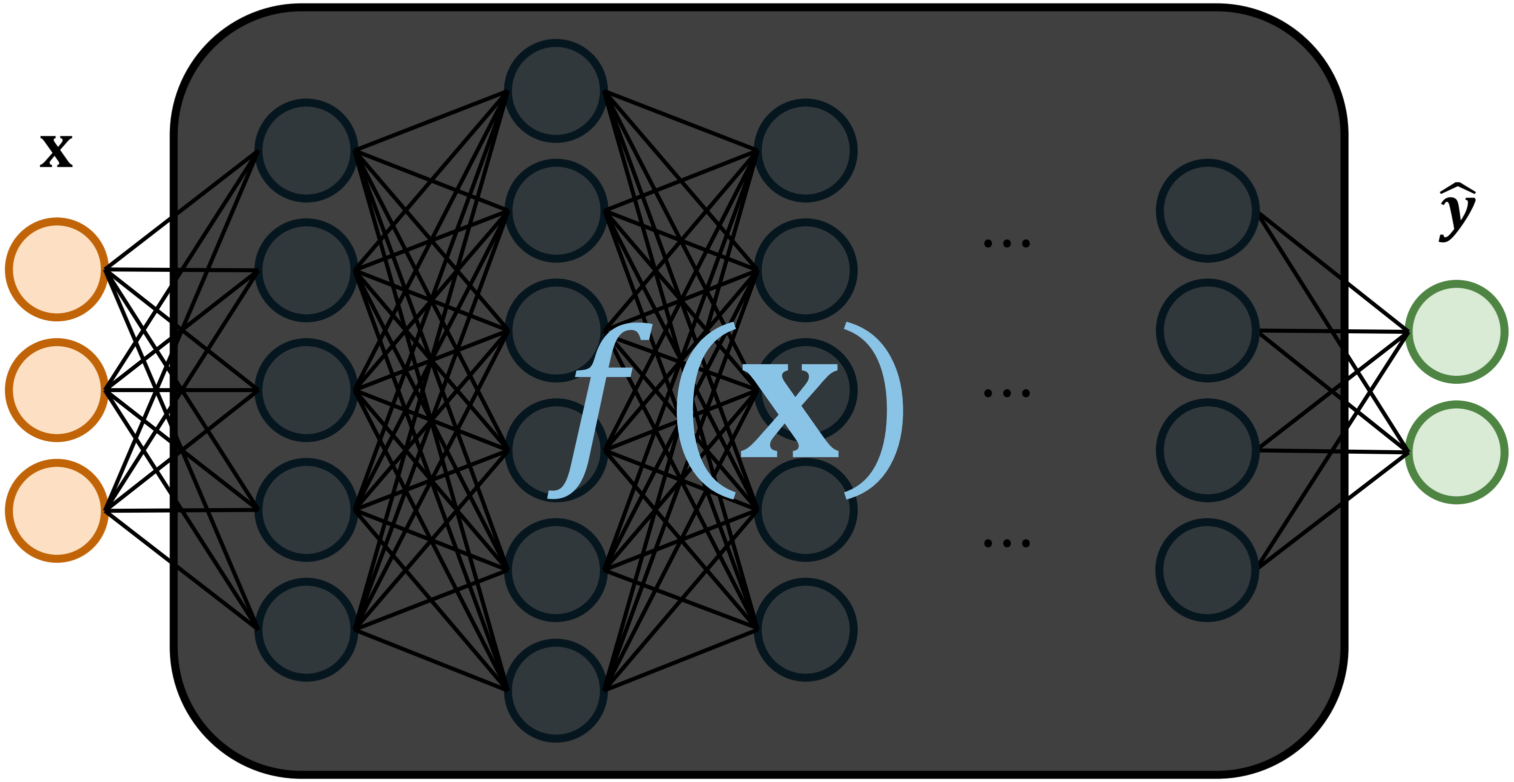
$$\mathbf{h}_2 = \varphi(W_2\mathbf{h}_1 + \mathbf{b}_2)$$

$$\mathbf{h}_3 = \varphi(W_2\mathbf{h}_2 + \mathbf{b}_2)$$

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

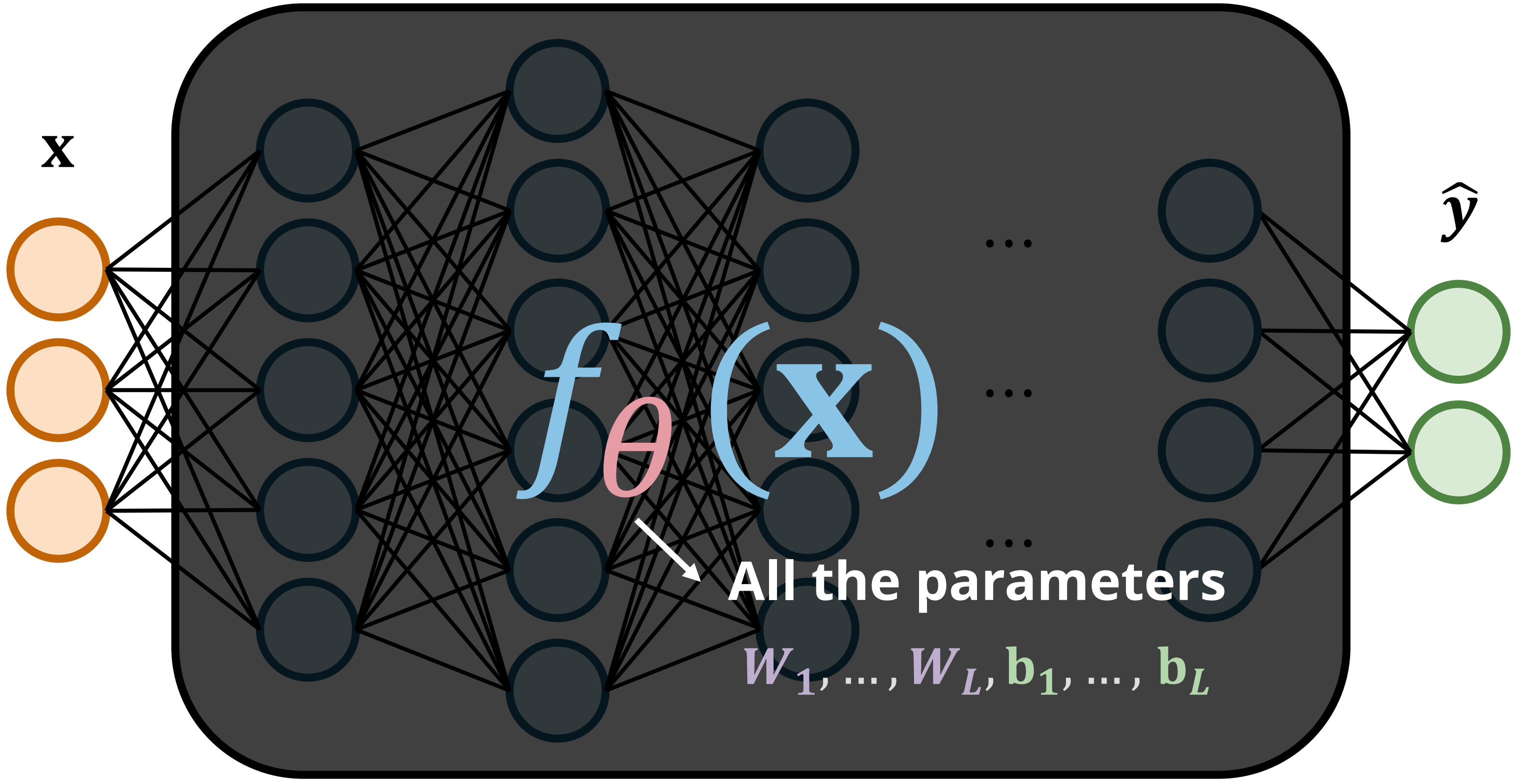
Neural Networks are Parameterized Functions

- 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

Build a neural network
(which defines a set of functions)

↓

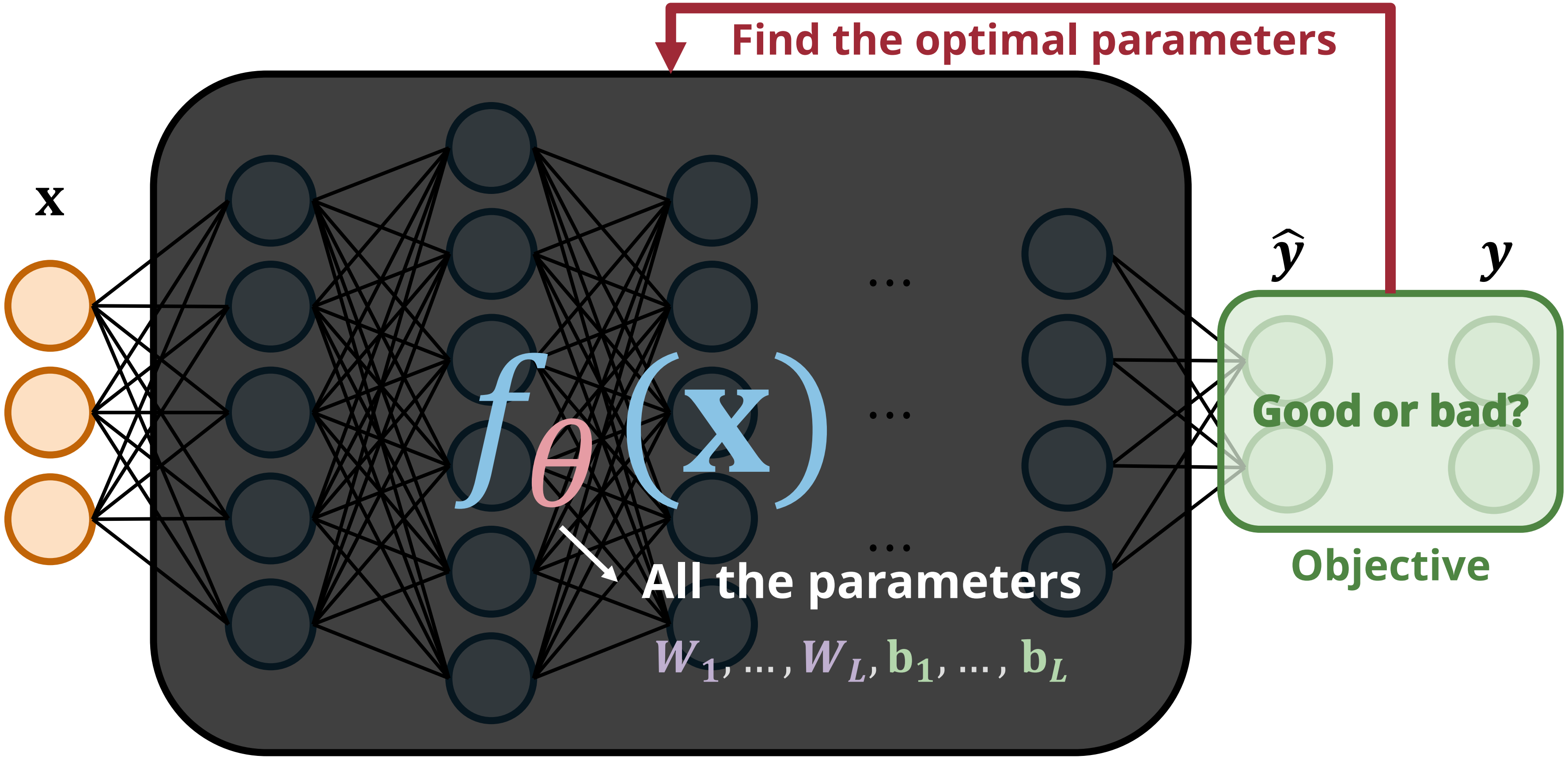
Define the objective
(i.e., what is good for a function)

↓

Find the optimal parameters
(which leads to the best function)

Neural Networks are Parameterized Functions

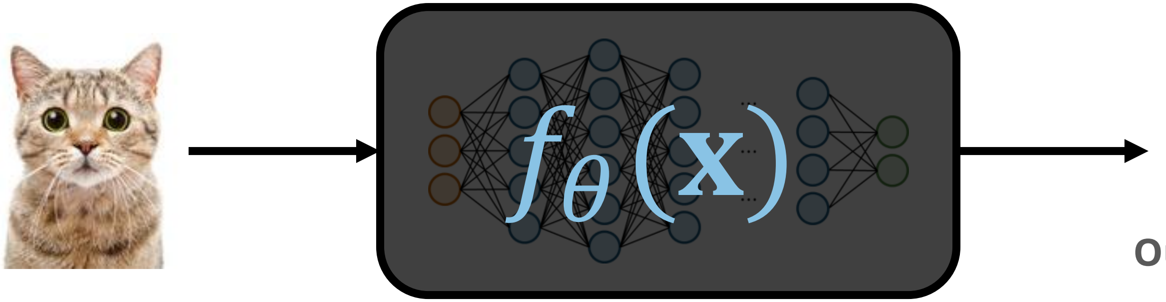
- A neural network represents **a set of functions**



Regression vs Classification

Regression vs Classification

Regression

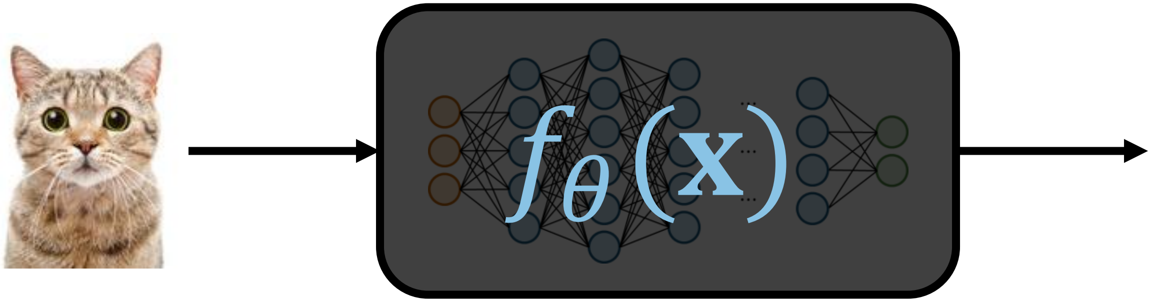


Age

5

Output a number

Classification



Is human?

Yes / No

Output a label

Regression Example: Stock Price Prediction

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

$$f\Bigg($$ 

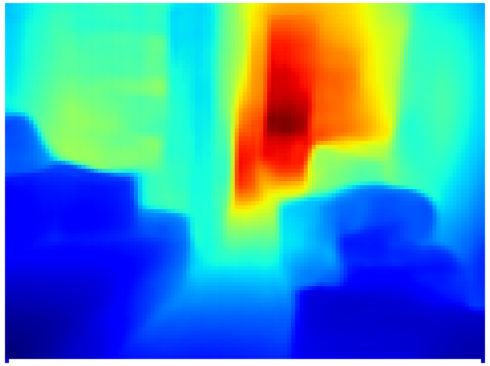
$$\Bigg) = 108.15$$

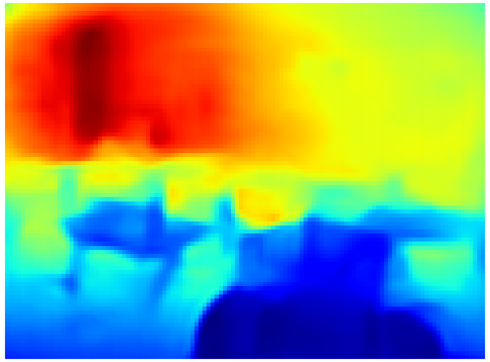
$$f\Bigg($$ 

$$\Bigg) = 18.95$$

Regression Example: Depth Estimation

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

$f\Big($  $\Big) =$ 

$f\Big($  $\Big) =$ 

medium.com/@omarbarakat1995/depth-estimation-with-deep-neural-networks-part-1-5fa6d2237d0d

Classification Example: Image Recognition

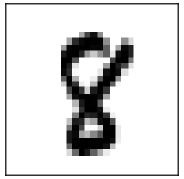
$y \in \{\text{cat}, \text{dog}, \text{bear}, \text{bird}\}$

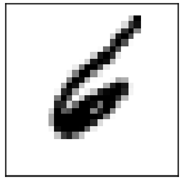
$f($  $) = \text{cat}$

$f($  $) = \text{dog}$

$f($  $) = \text{bear}$

$y \in \{0, 1, 2, \dots, 9\}$

$f($  $) = 8$

$f($  $) = 6$

Classification Example: Spam Filter

$f\Big($

POWERBALL POWERPLAY

CONGRATULATIONS!!

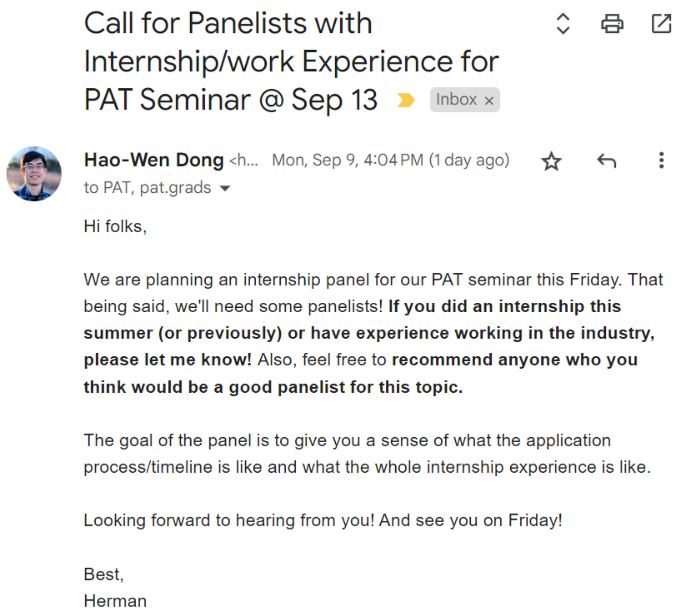
Your Email was selected in Powerball Lottery Draw with the sum of 1.5million dollars. Kindly send your Full Name, Address and Phone Number for claims.

Yours Sincerely
Mr. James Hodges
Head Of Operations

$\Big) = \text{spam}$

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

$f\Big($



Call for Panelists with Internship/work Experience for PAT Seminar @ Sep 13 Inbox ×

Hao-Wen Dong <h... Mon, Sep 9, 4:04 PM (1 day ago)
to PAT, pat.grads

Hi folks,

We are planning an internship panel for our PAT seminar this Friday. That being said, we'll need some panelists! **If you did an internship this summer (or previously) or have experience working in the industry, please let me know!** Also, feel free to **recommend anyone who you think would be a good panelist for this topic.**

The goal of the panel is to give you a sense of what the application process/timeline is like and what the whole internship experience is like.

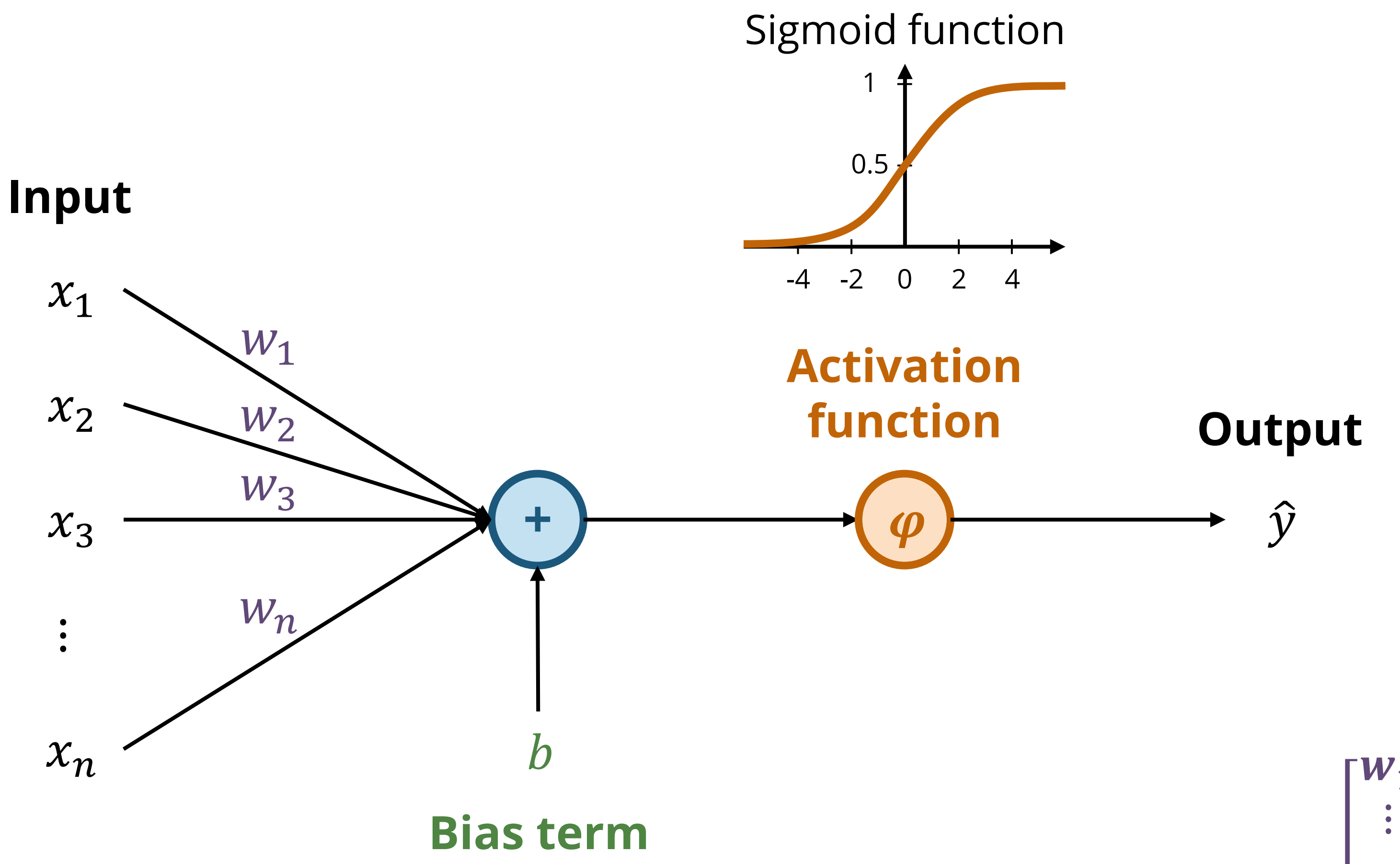
Looking forward to hearing from you! And see you on Friday!

Best,
Herman

$\Big) = \text{not spam}$

Activation Functions

Inside a Neuron



$$\hat{\boldsymbol{y}} = \boldsymbol{\varphi}(w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b) = \boldsymbol{\varphi}\left(\sum_{i=1}^{n} w_i x_i + b\right) = \boldsymbol{\varphi}(\overset{\begin{bmatrix} \boldsymbol{w_1} \\ \vdots \\ \boldsymbol{w_n} \end{bmatrix}}{\mathbf{w}} \cdot \overset{\begin{bmatrix} \boldsymbol{x_1} \\ \vdots \\ \boldsymbol{x_n} \end{bmatrix}}{\mathbf{x}} + b)$$

Why do We Need Activation Functions?

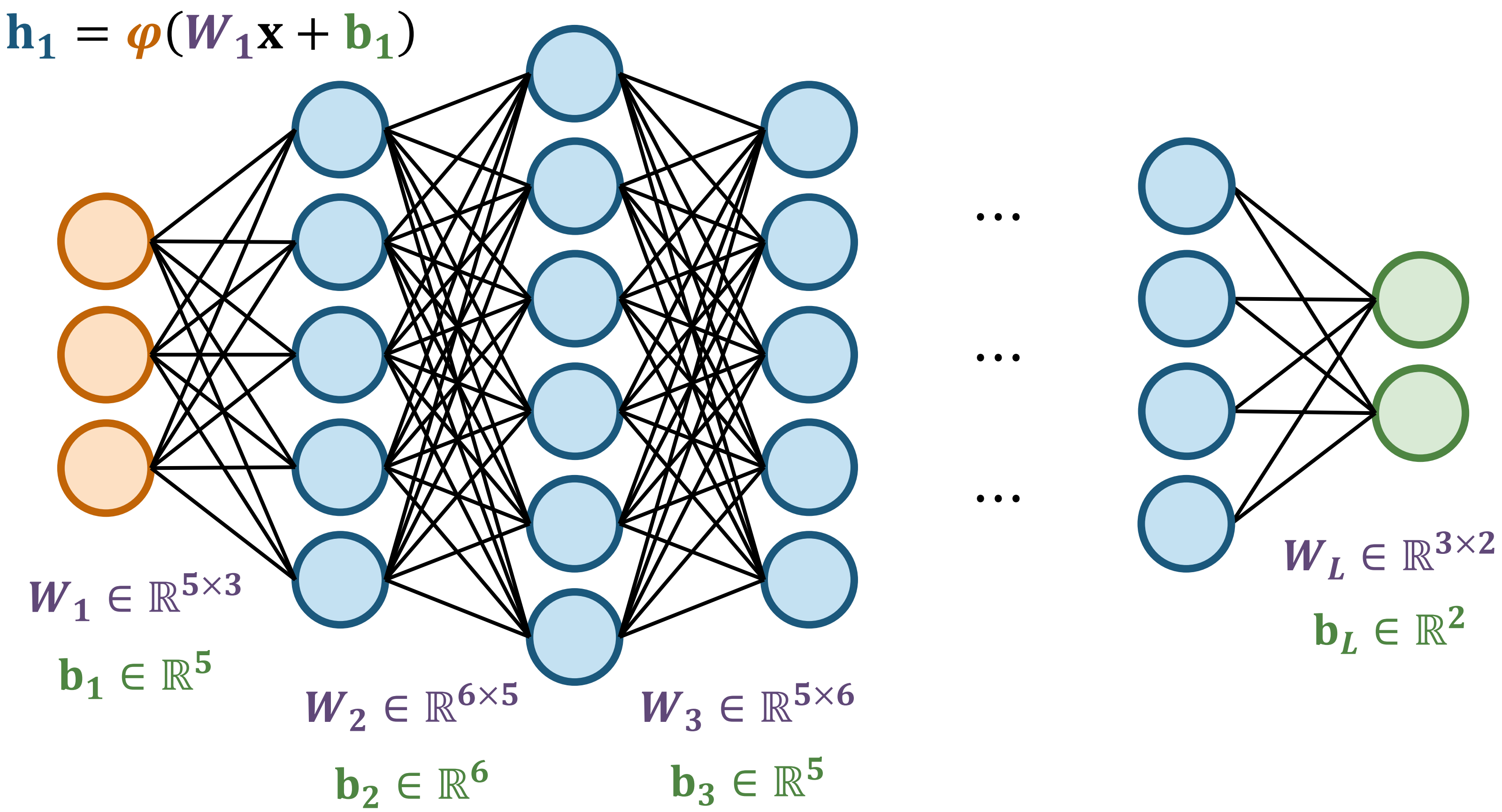
- Activation functions introduce **nonlinearity** to a neural network
- A linear function is a **weighted sum of the inputs** (plus a bias term)

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

- 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?



Why do We Need Activation Functions?

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

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

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

$$\vdots$$

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

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

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

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

$$\vdots$$

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

Why do We Need Activation Functions?

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

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

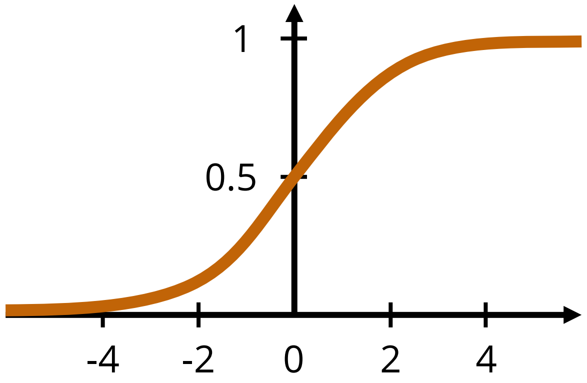
$$\downarrow$$

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

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

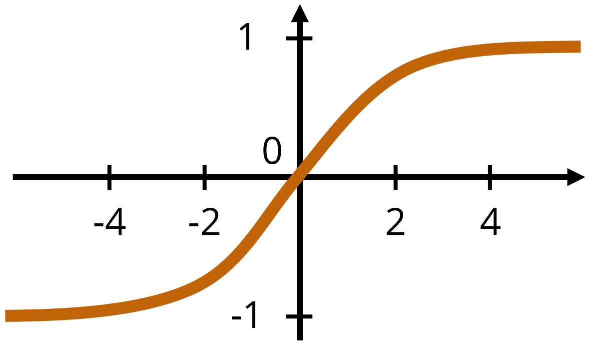
Commonly Used Activation Functions

Sigmoid



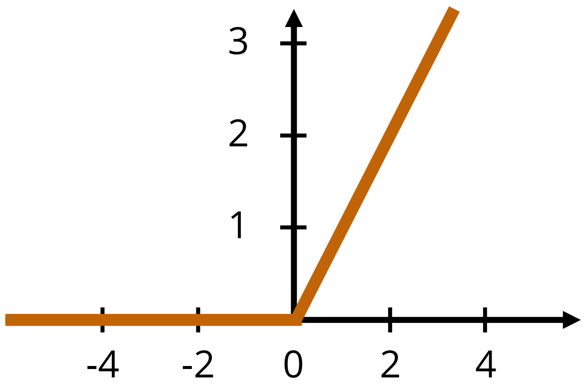
$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

tanh



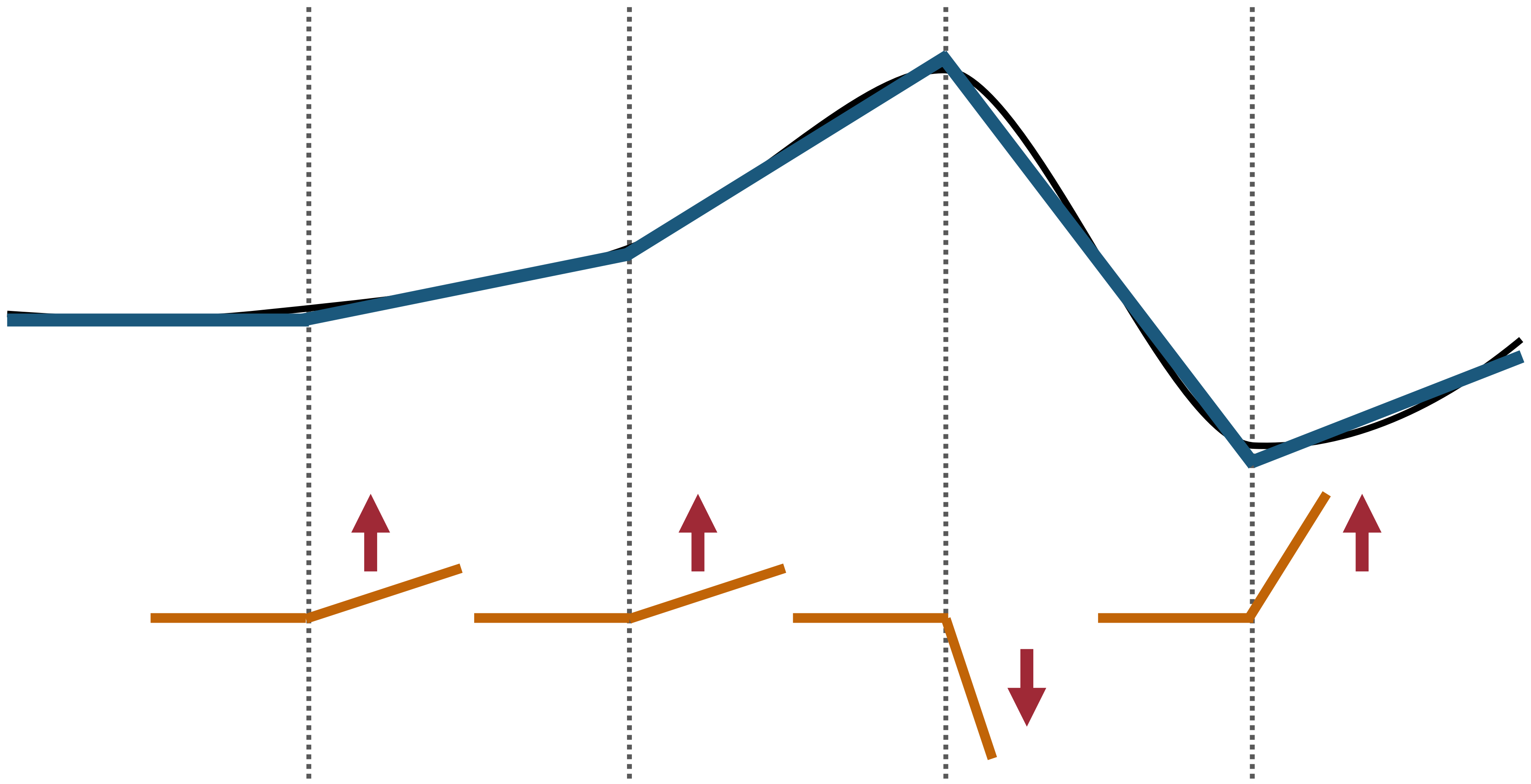
$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

ReLU



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

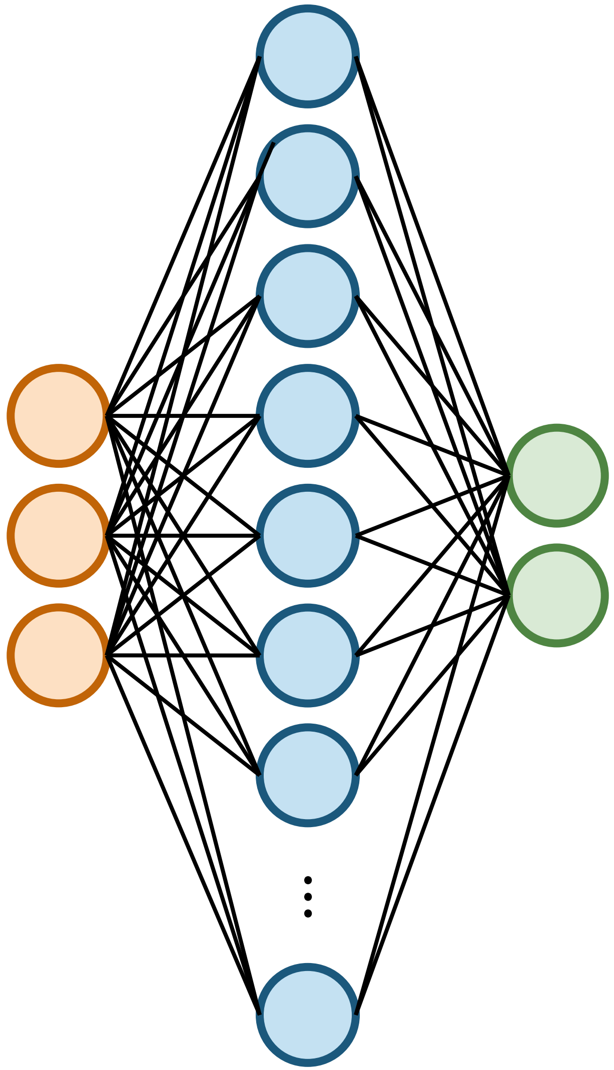
ReLUs & Piecewise Linear Functions



Expressiveness of Neural Networks

Universal Approximation Theorem

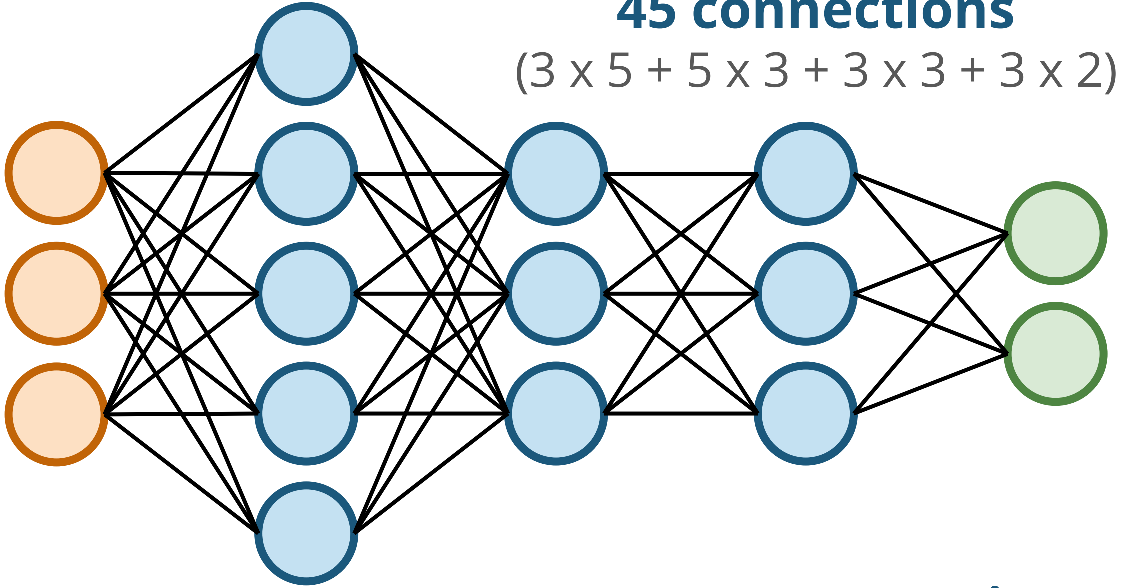
- A neural network with **one hidden layer** can **approximate any continuous function** given **sufficient hidden neurons** and **appropriate activation functions**
  - Sigmoid, ReLUs are good activation functions



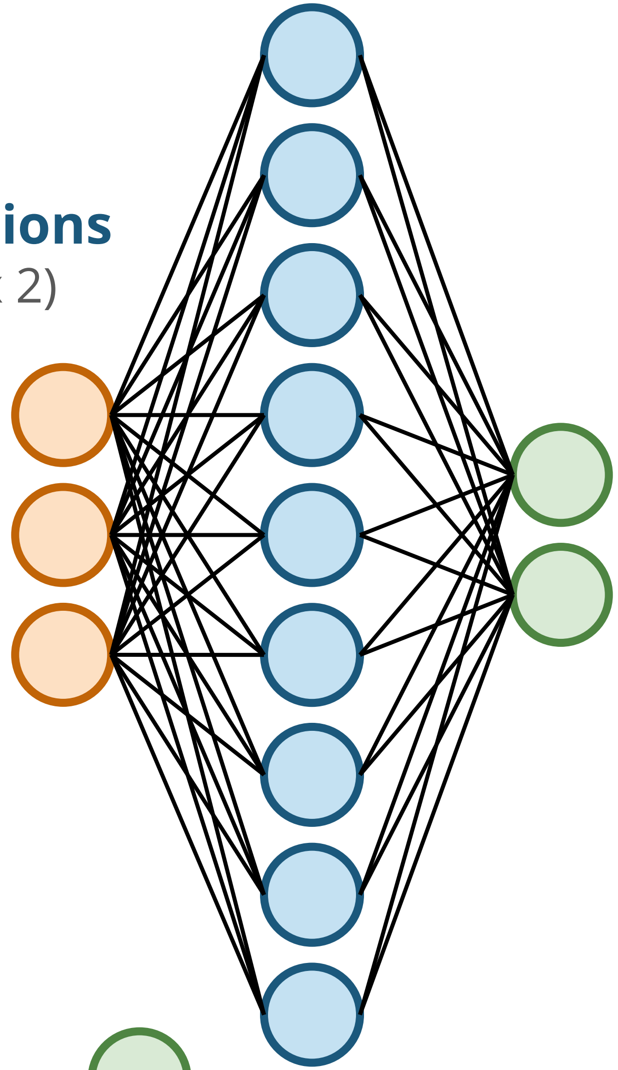
Then why do we want to go deep?

Shallow vs Deep Neural Networks

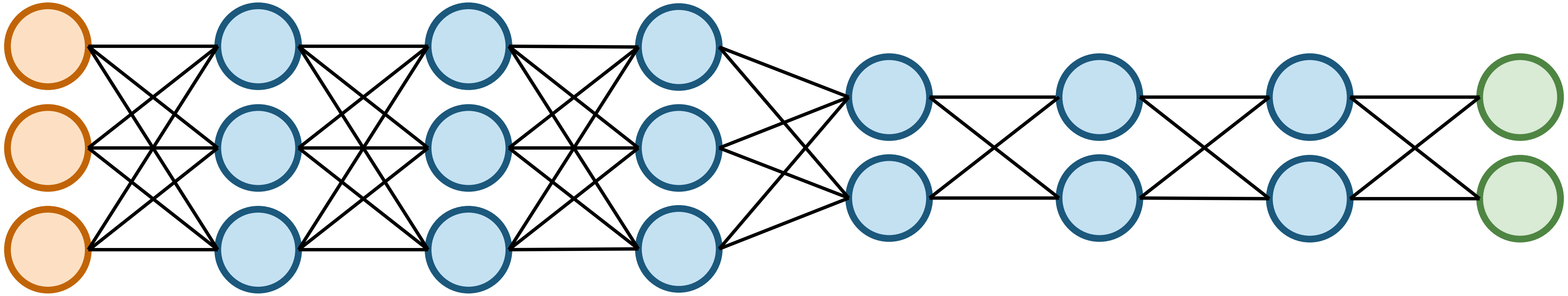
45 connections
(3 x 5 + 5 x 3 + 3 x 3 + 3 x 2)



45 connections
(3 x 9 + 9 x 2)

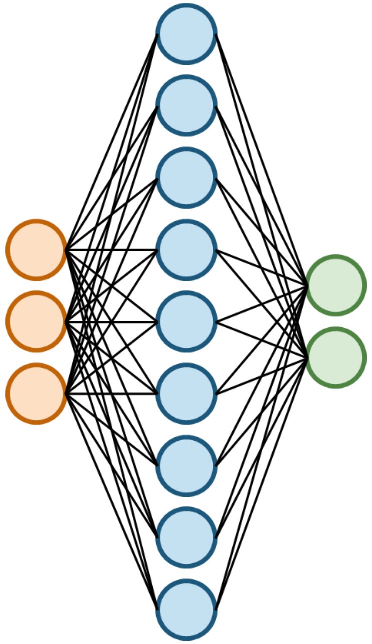


45 connections
(3 x 3 + 3 x 3 + 3 x 3 + 3 x 2 + 2 x 2 + 2 x 2 + 2 x 2)



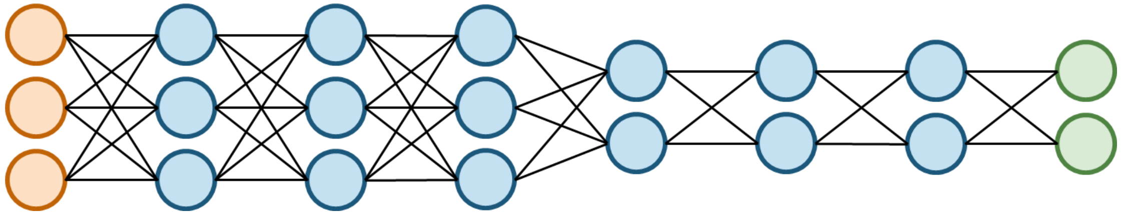
Shallow vs Deep Neural Networks – In Practice

Shallow neural nets



Less expressive
(less parameter efficient)

Deep neural nets

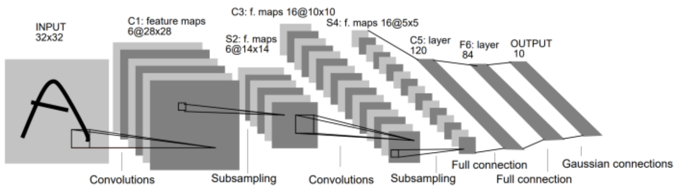


More expressive
(more parameter efficient)

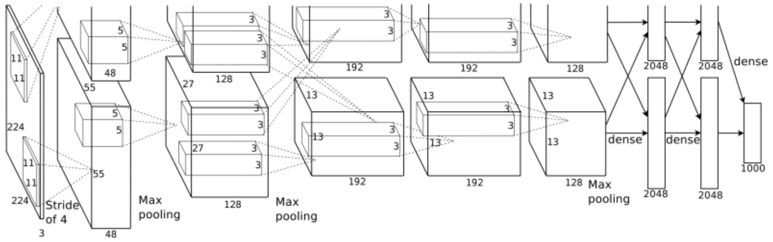
How Deep is Deep Enough?

- **Deeper is not always better**
  - Actual number of parameters
  - Optimization difficulties
  - Data size
  - Inductive bias of the model

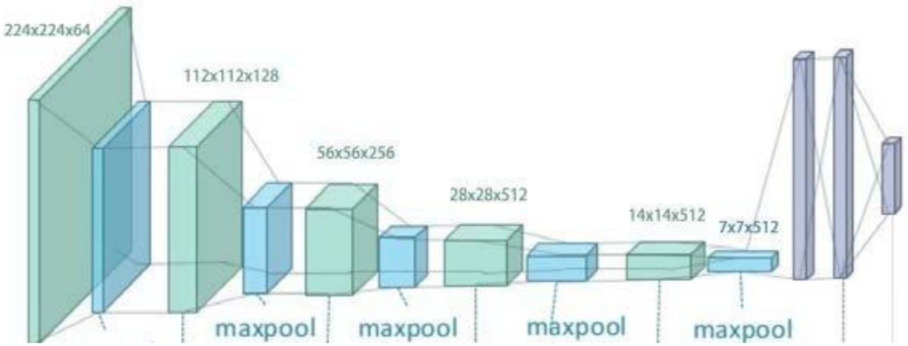
LeNet
(1998)



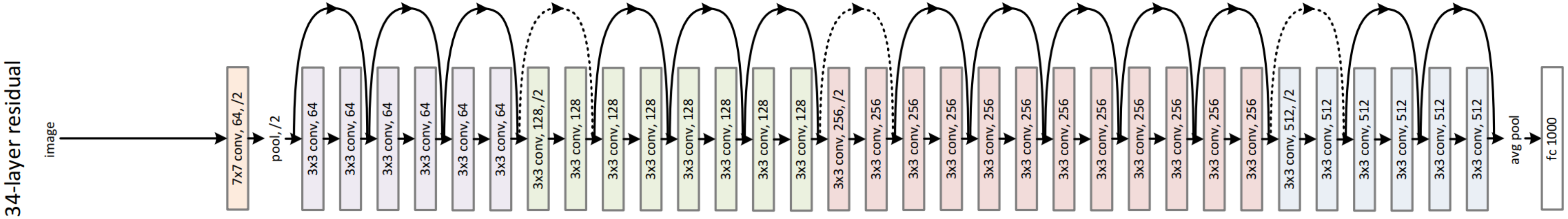
AlexNet
(2012)



VGG-19
(2015)

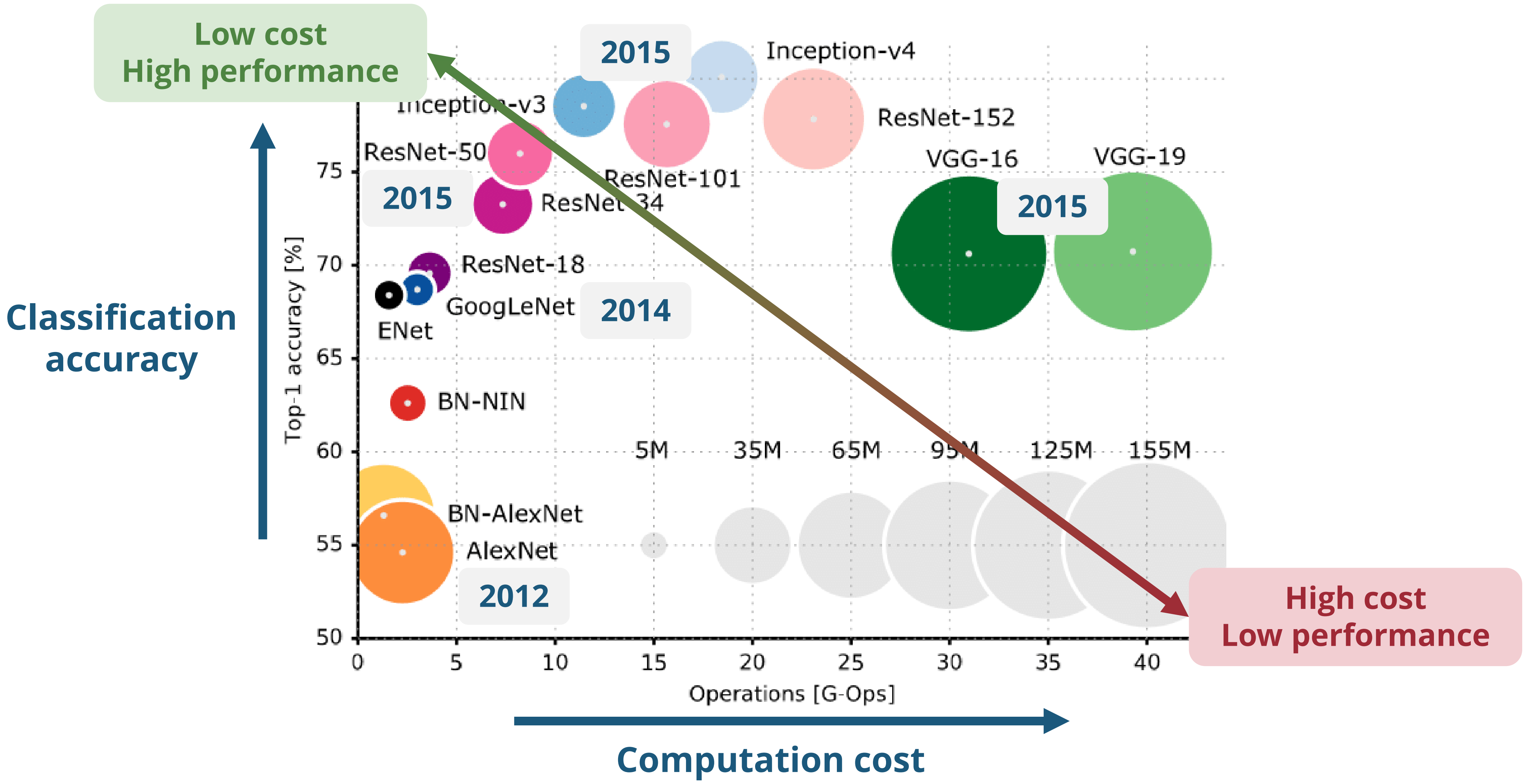


ResNet
(2015)



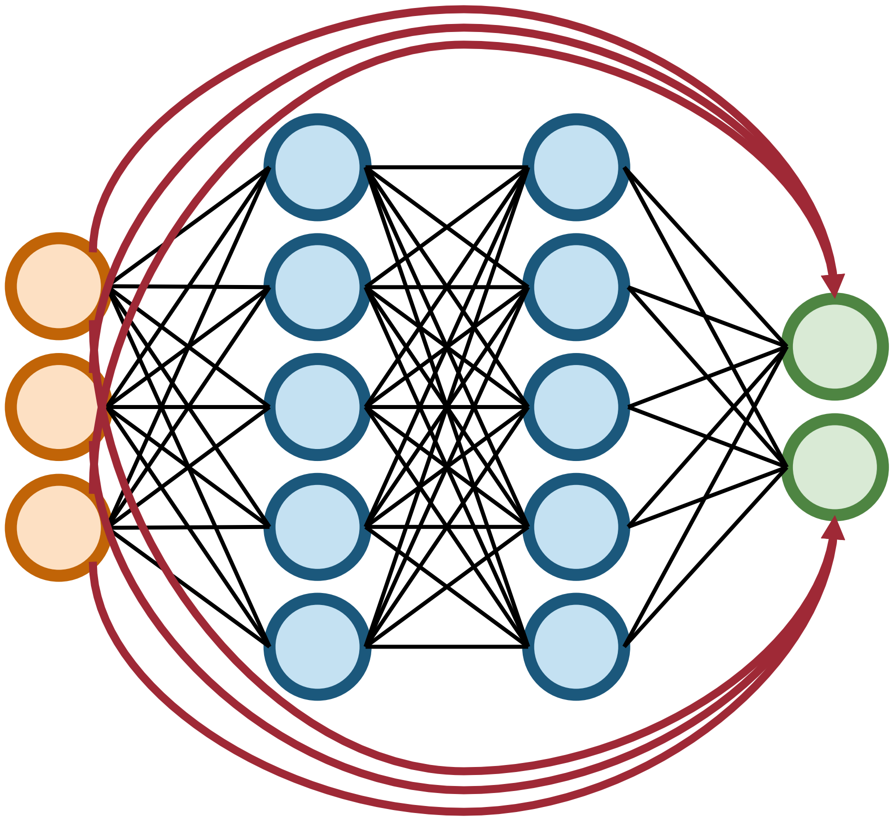
Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, 1998.
Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *NeurIPS*, 2012.
Karen Simonyan and Andrew 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.

Computation Cost vs Classification Accuracy



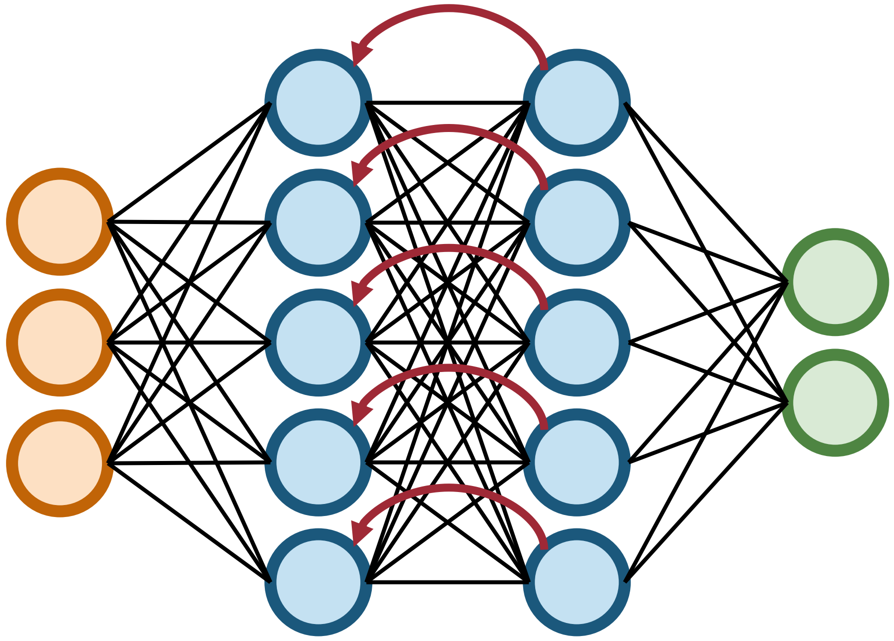
Neural Networks are NOT always Layer-by-Layer

Skip connections



Used in ResNets, U-Nets, diffusion models

Feedback loops

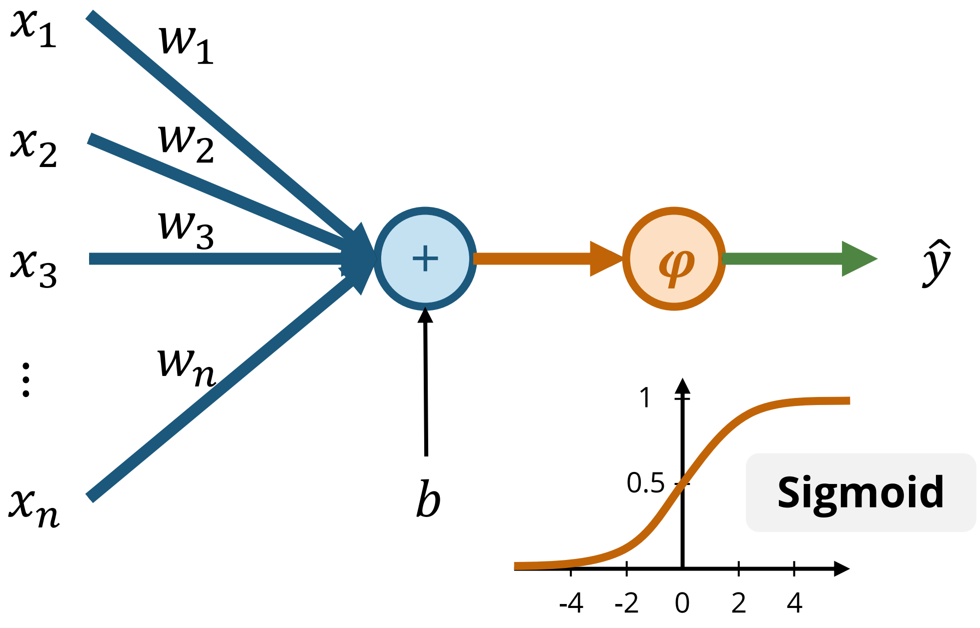


Used in RNNs, LSTMs, GRUs

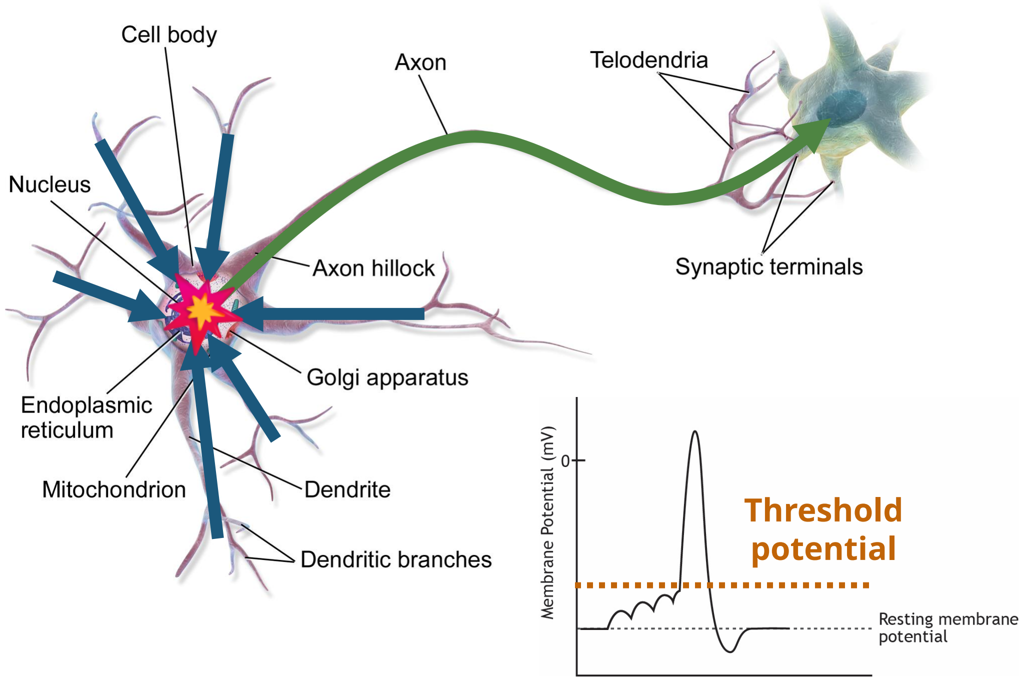
Recap

Artificial vs Human Neuron

Artificial neuron

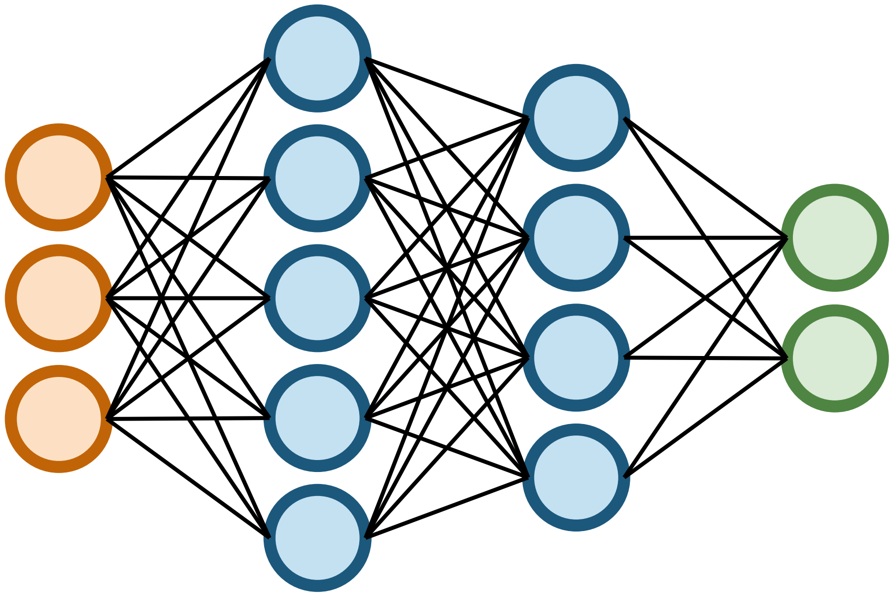


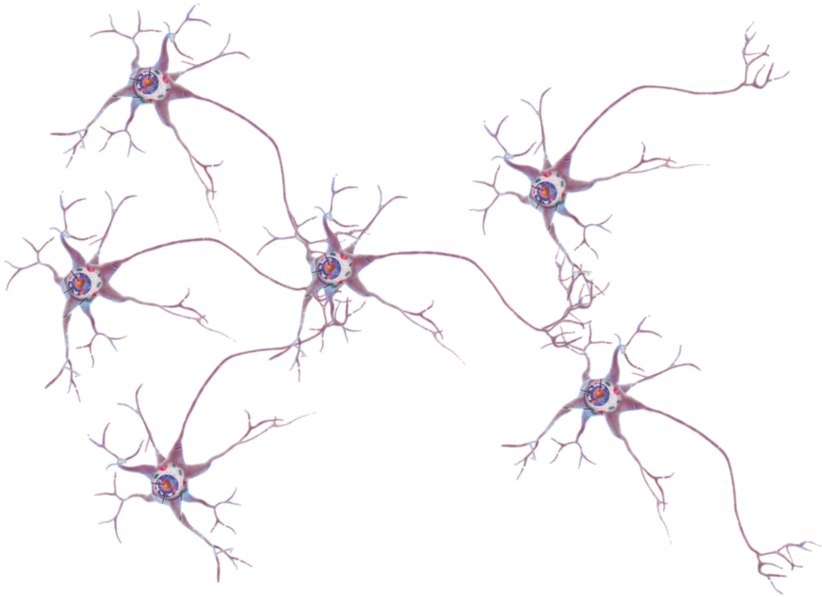
Human neuron



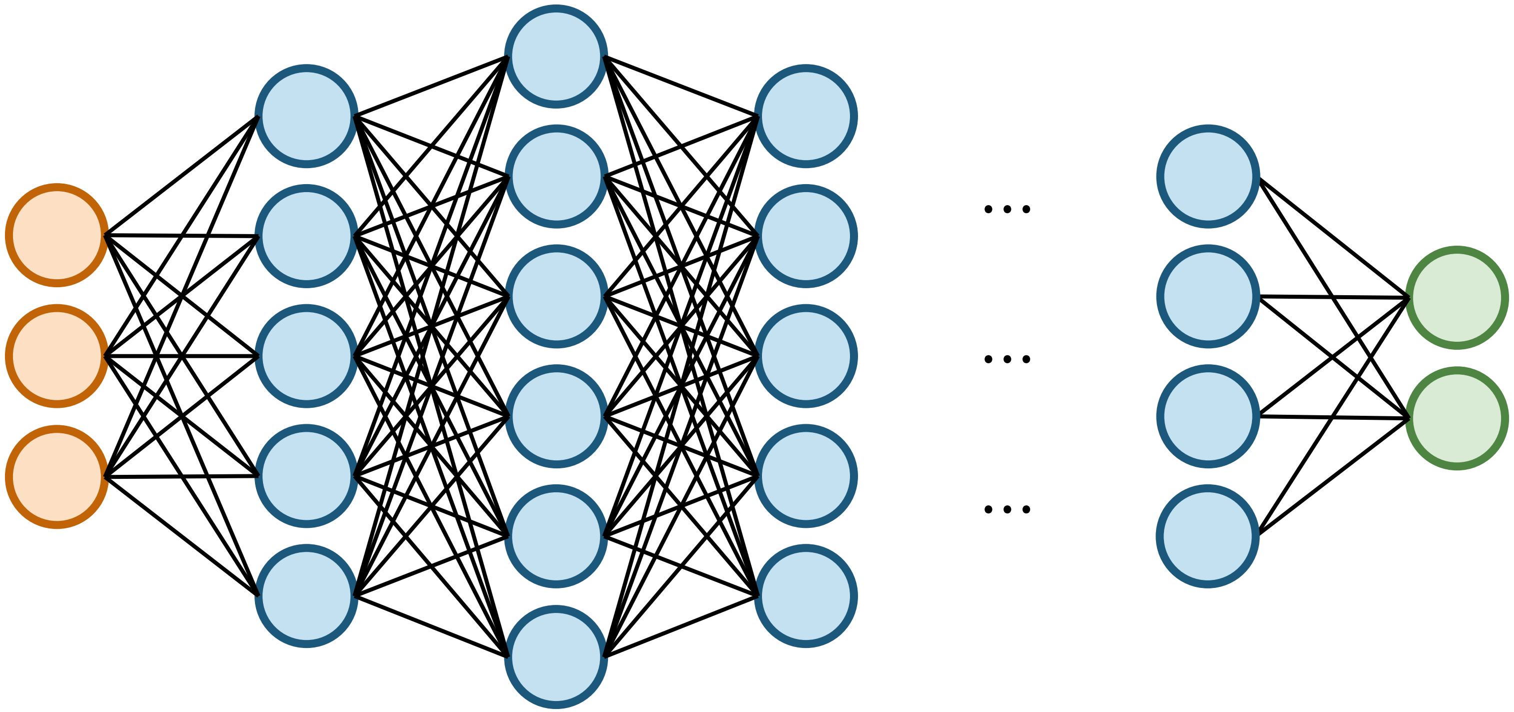
Artificial Neural Networks

- 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



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

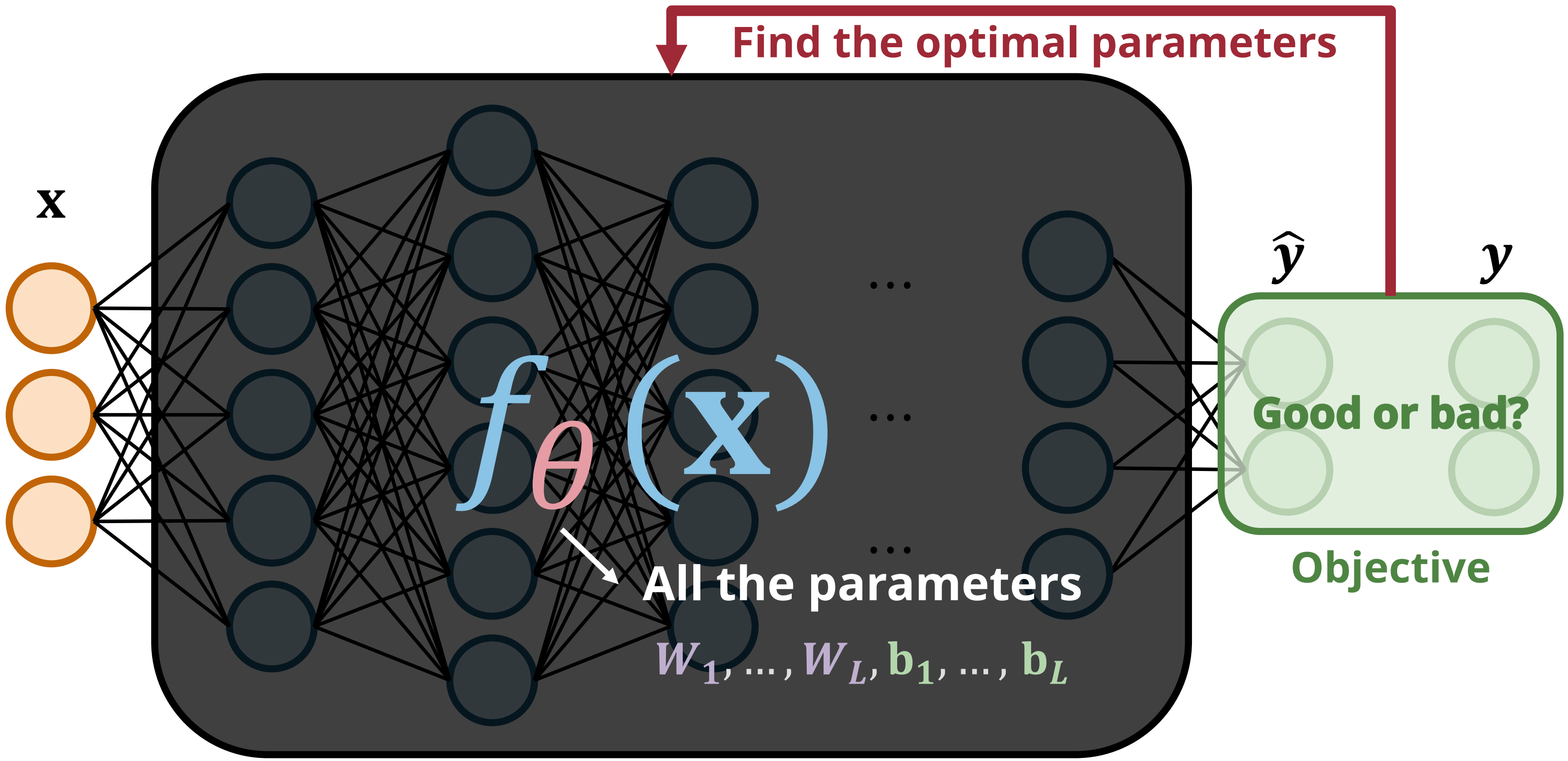
$$\mathbf{h}_2 = \varphi(W_2\mathbf{h}_1 + \mathbf{b}_2)$$

$$\mathbf{h}_3 = \varphi(W_2\mathbf{h}_2 + \mathbf{b}_2)$$

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

Neural Networks are Parameterized Functions

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



Training a Neural Network

Build a neural network
(which defines a set of functions)

↓

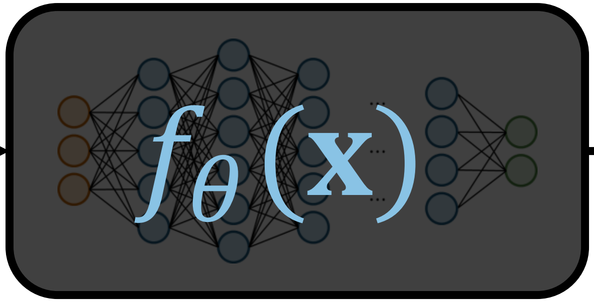
Define the objective
(i.e., what is good for a function)

↓

Find the optimal parameters
(which leads to the best function)

Regression vs Classification

Regression

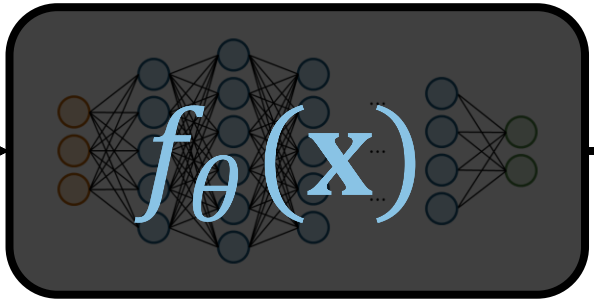


Age

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**

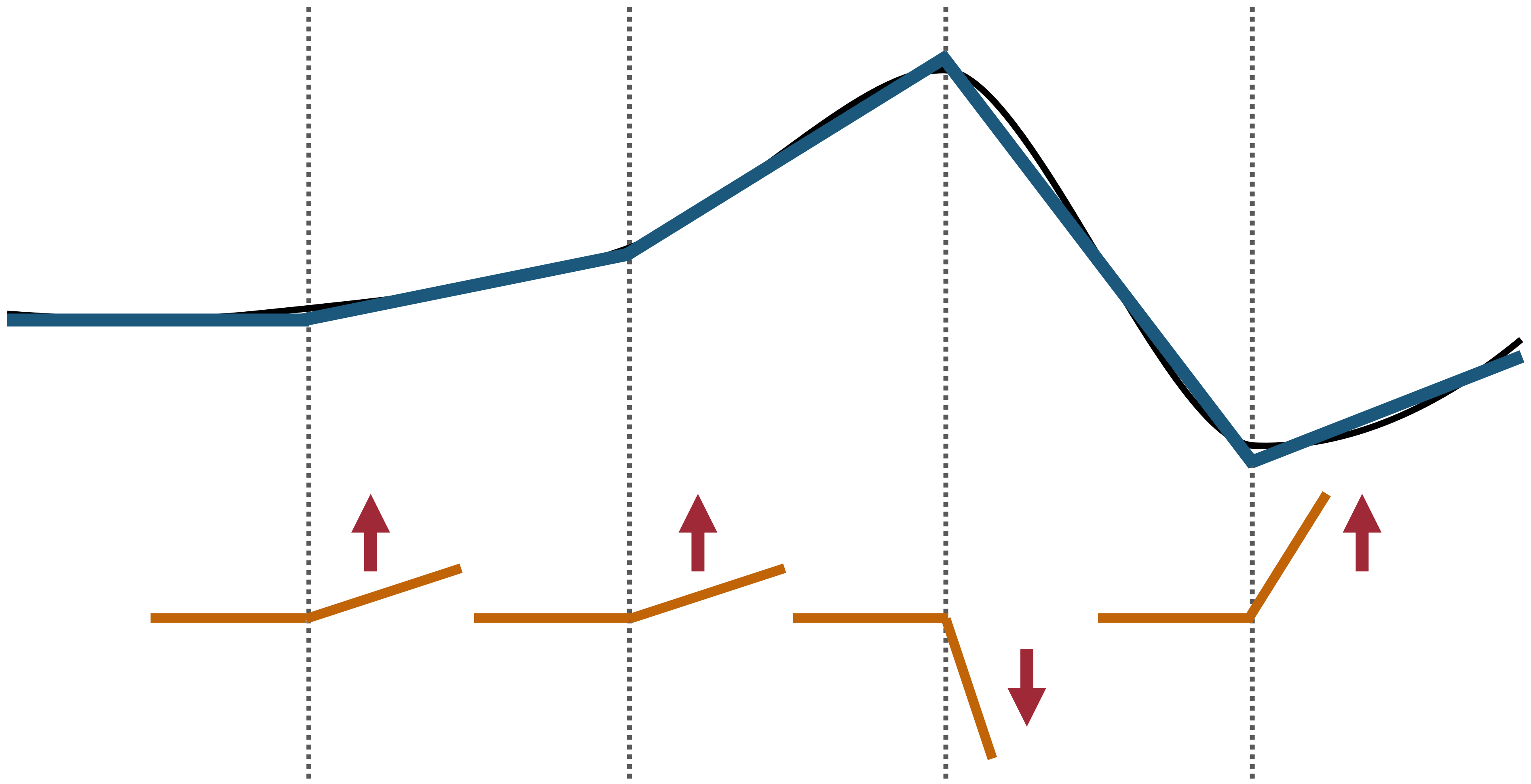
$$\hat{\boldsymbol{y}} = \boldsymbol{\varphi}(\boldsymbol{W}_L\ \boldsymbol{\varphi}(\boldsymbol{W}_{L-1}\ \boldsymbol{\varphi}(\boldsymbol{W}_{L-2}\ \boldsymbol{\varphi}(\cdots \mathbf{x} \cdots) + \mathbf{b}_{L-2})\ + \mathbf{b}_{L-1})\ + \mathbf{b}_L)$$

$$\downarrow$$

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

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

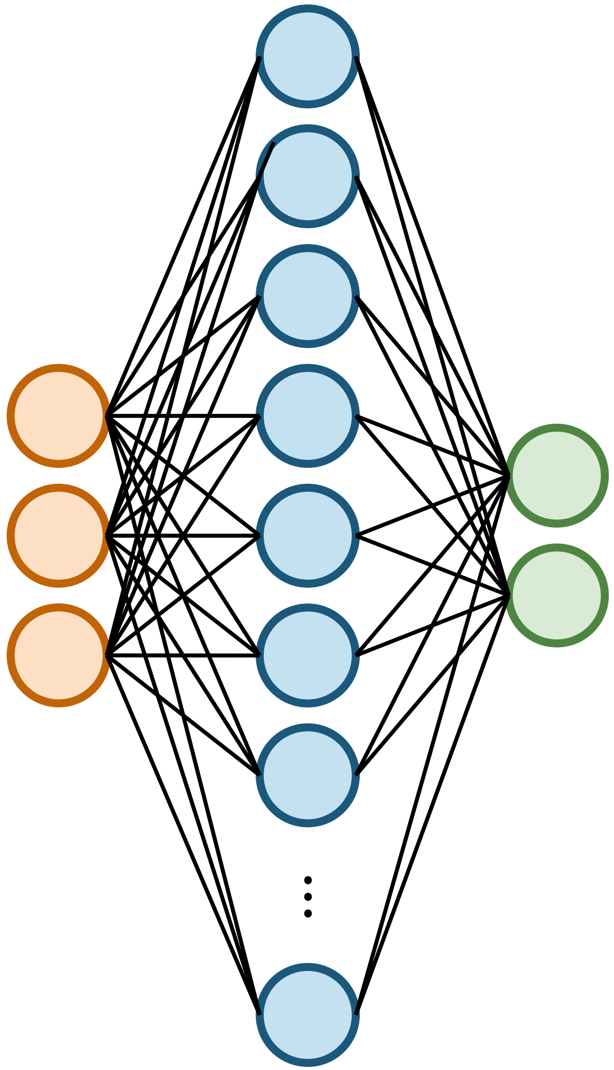
ReLUs & Piecewise Linear Functions



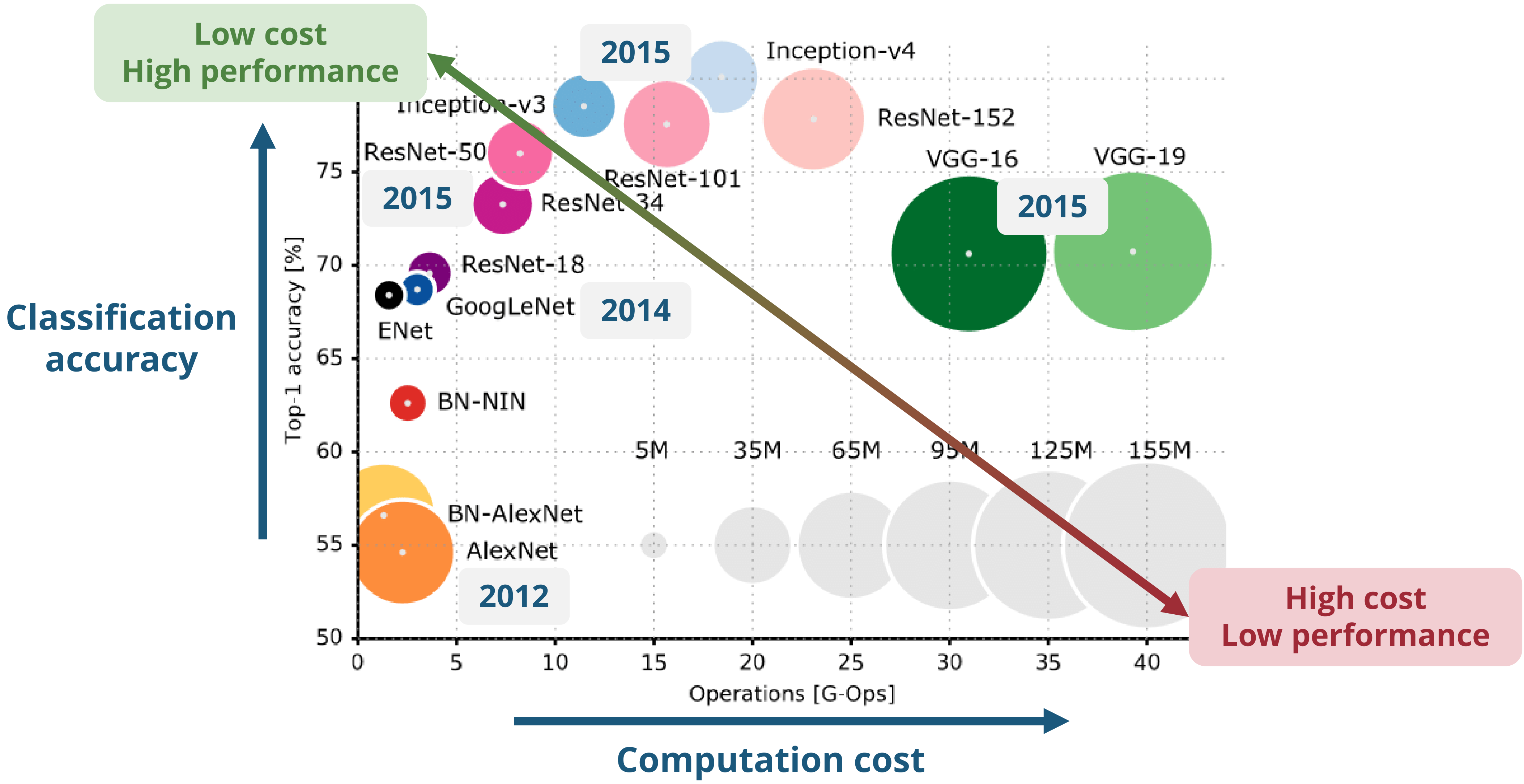
Universal Approximation Theorem

- 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



Next Lecture

Deep Learning Fundamentals II