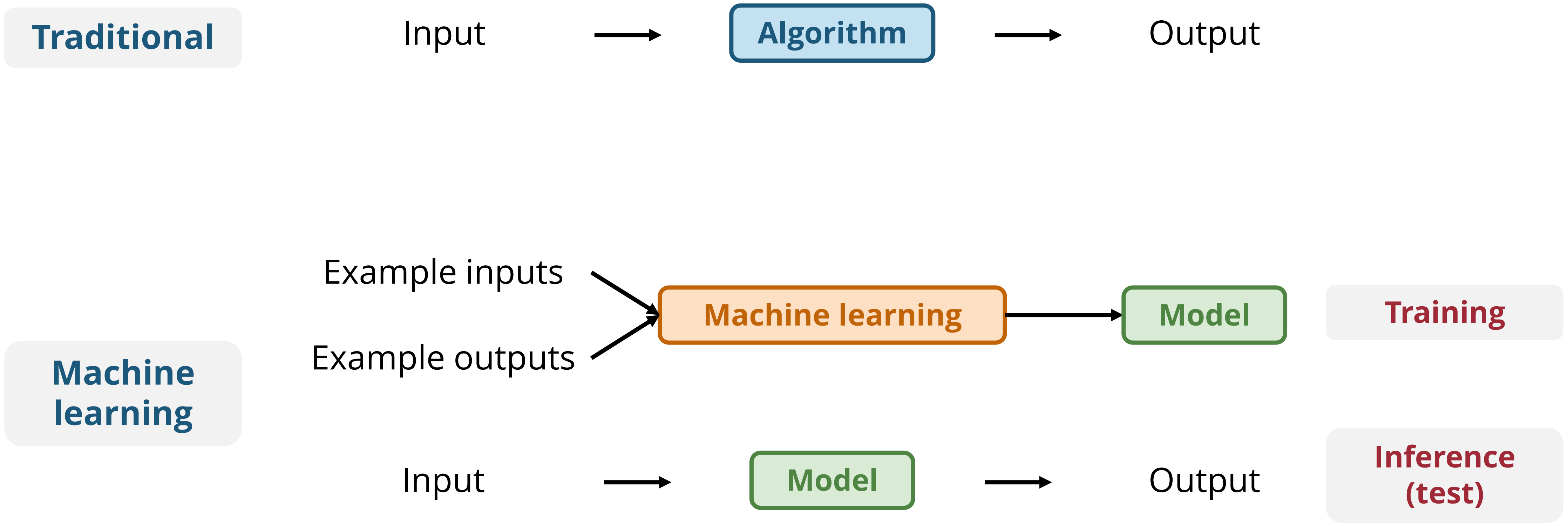
PAT 498/598 (Fall 2024)

Special Topics:
Generative AI for Music and Audio Creation

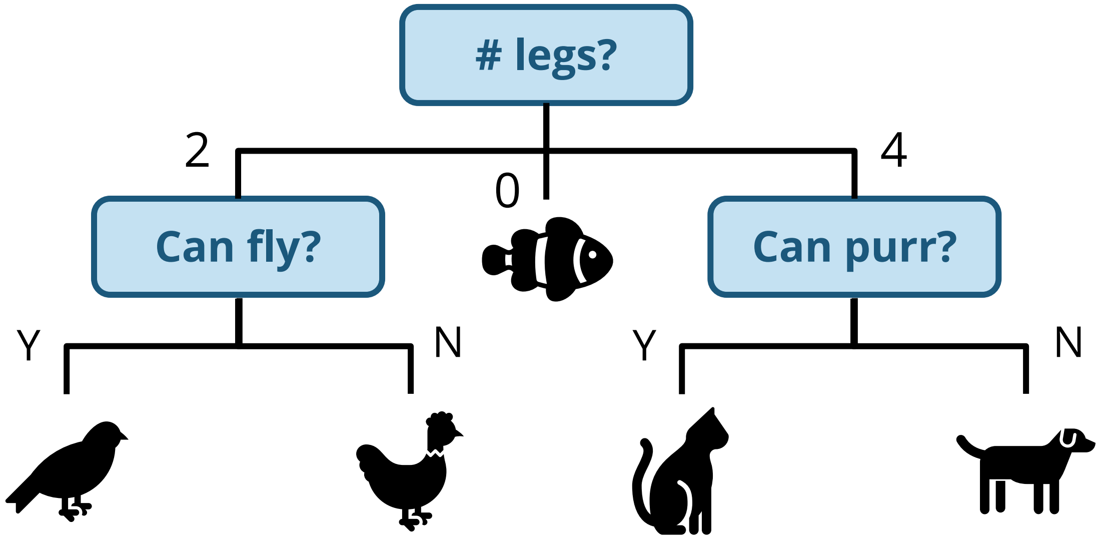
Lecture 4: Deep Learning Fundamentals

Instructor: Hao-Wen Dong

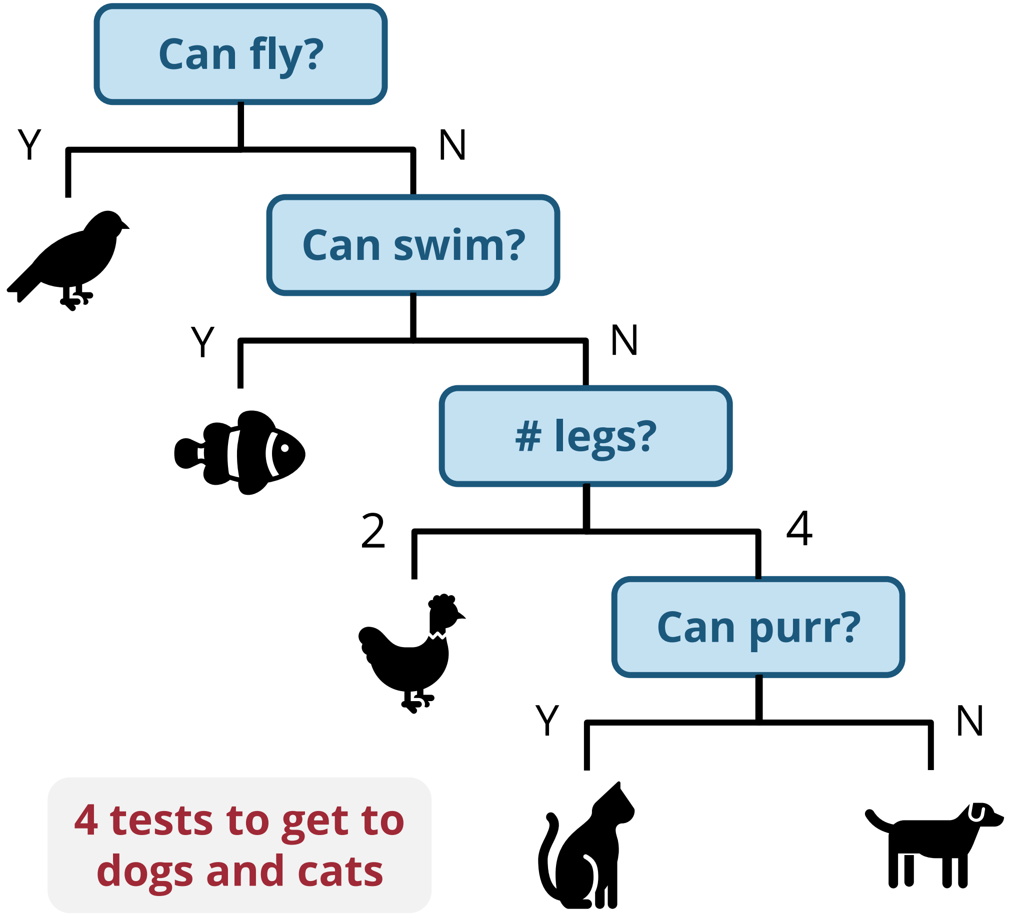
(Recap) Machine Learning



(Recap) How to Determine the Order of Features to Test?



2 tests maximum to get to any animal type



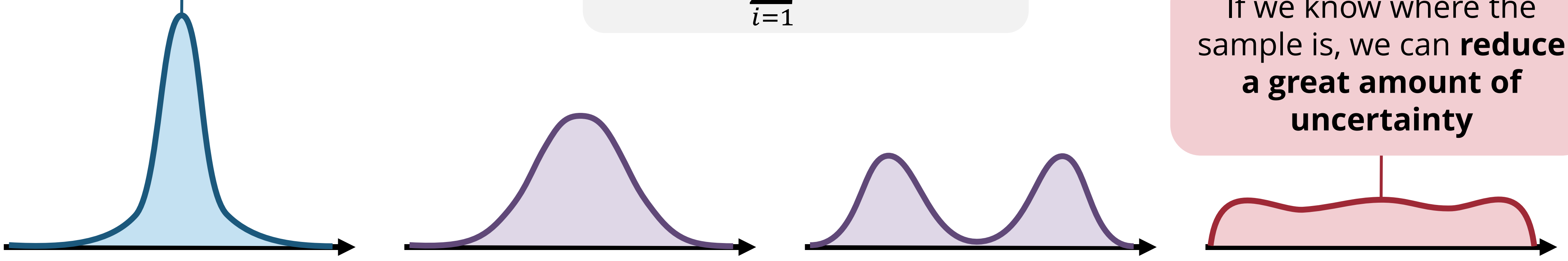
4 tests to get to dogs and cats

(Recap) Entropy of a Distribution

Even without knowing anything, we know **the sample is very likely to be around the peak**

$$H(S) = -\sum_{i=1}^{C} P(c_i) \log P(c_i)$$

If we know where the sample is, we can **reduce a great amount of uncertainty**



Low entropy ⟷ **High entropy**

Low uncertainty ⟷ High uncertainty

Provide less information if known ⟷ Provide more information if known

(Recap) Components of a Machine Learning Model

Improve on task T,
with respect to performance metric P,
based on experience E

• **Task T**	**Animal classification**
• **Performance metric P**	Percentage of correct predictions
• **Experience E**	Examples of animals with their features

(Recap) Types of Machine Learning

- **Supervised learning** — Given **pairs of example inputs and outputs**
  - **Classification**: *discrete* outputs
  - **Regression**: *continuous* outputs
- **Unsupervised learning** — Given ***only* example inputs**
  - **Self-supervised learning**
- **Semi-supervised learning** — Given **example inputs** and **a few example outputs**
- **Reinforcement learning** — Given **scalar rewards** for **a sequence of actions**

Many generative AI models based on self-supervised learning!

Intro to Deep Learning

Components of a Machine Learning Model

Optimization

Defining inputs & outputs

Improve on task T,

with respect to performance metric P,

Loss function (objective function)

based on experience E

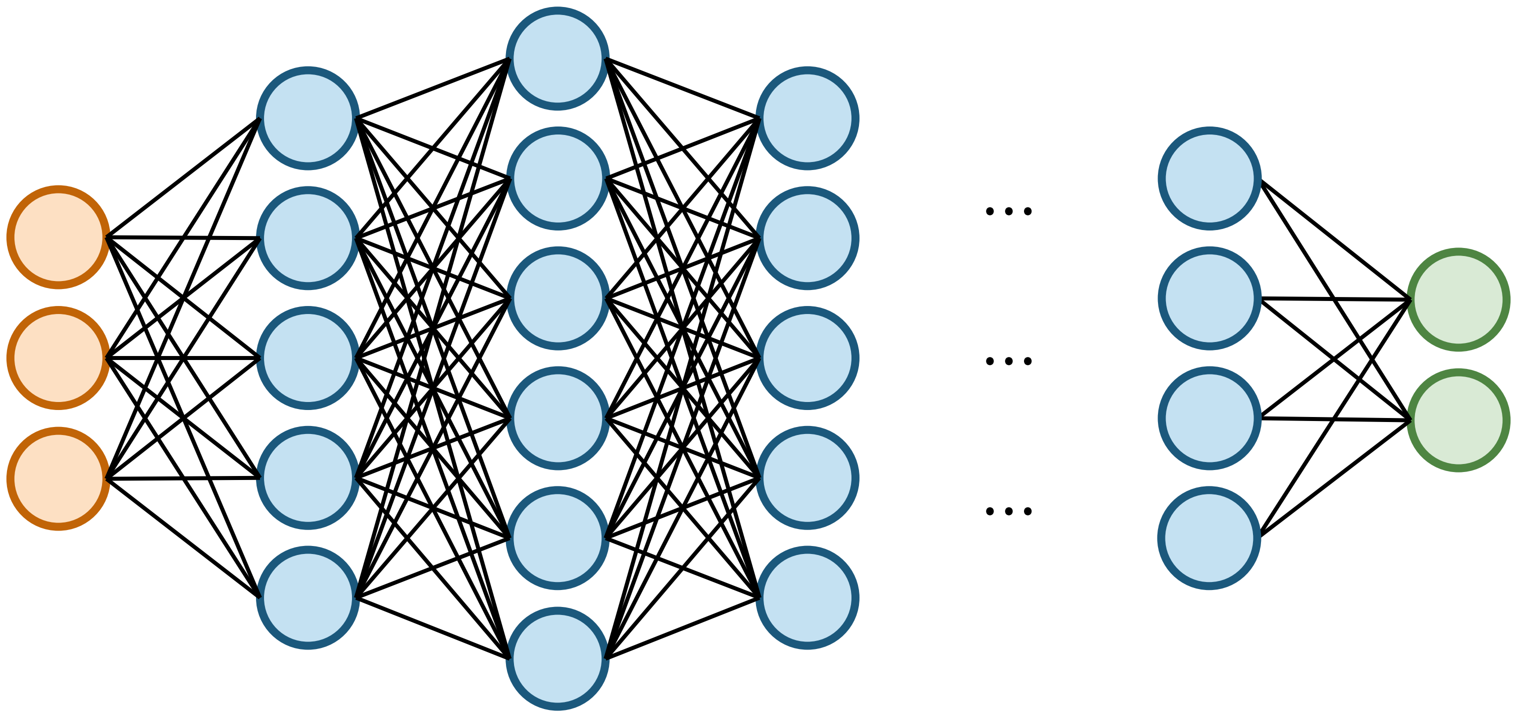
Training data

Deep learning is almost the same as machine learning by this definition!

What's special about deep learning?

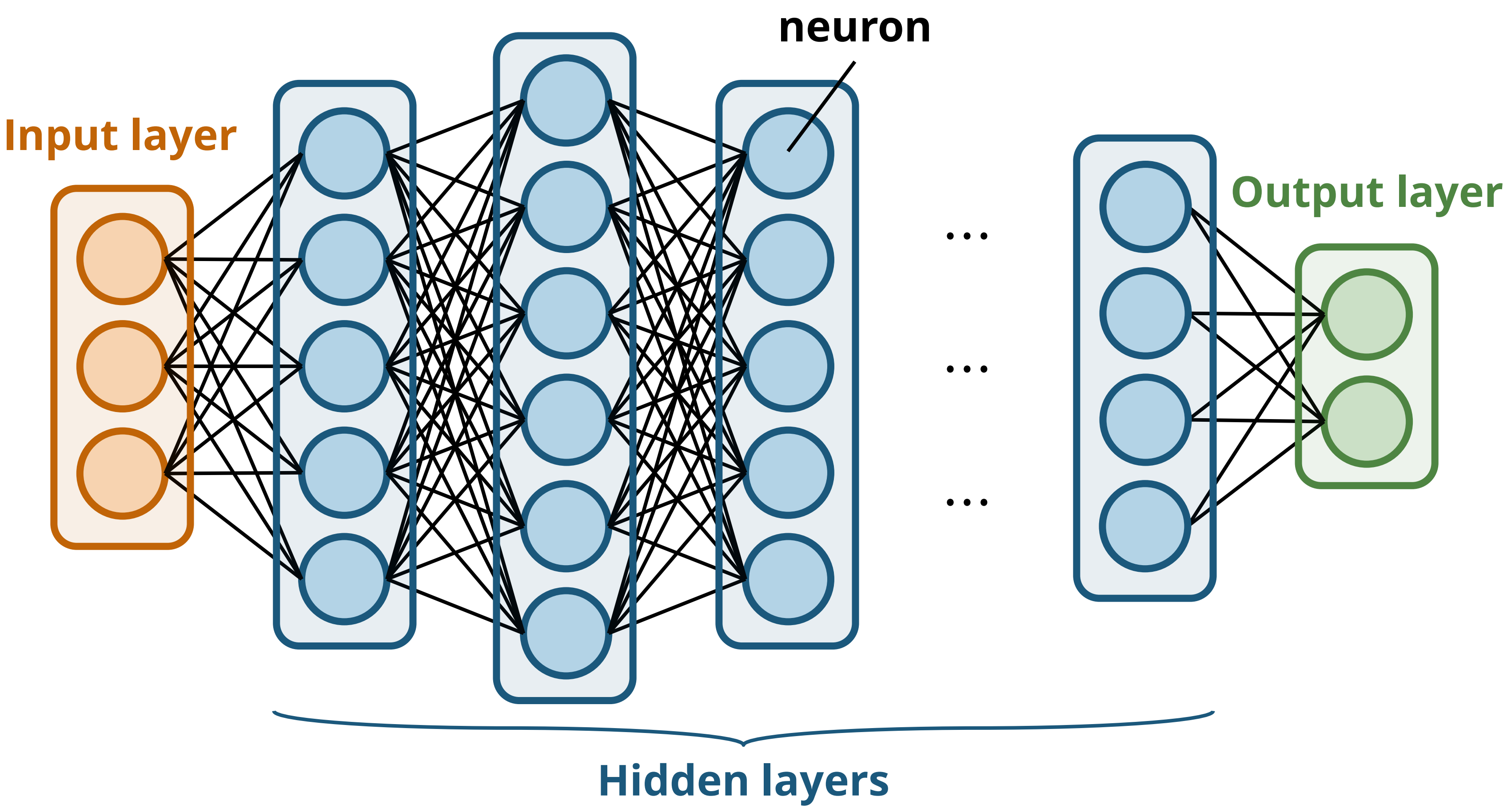
What is Deep Learning?

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



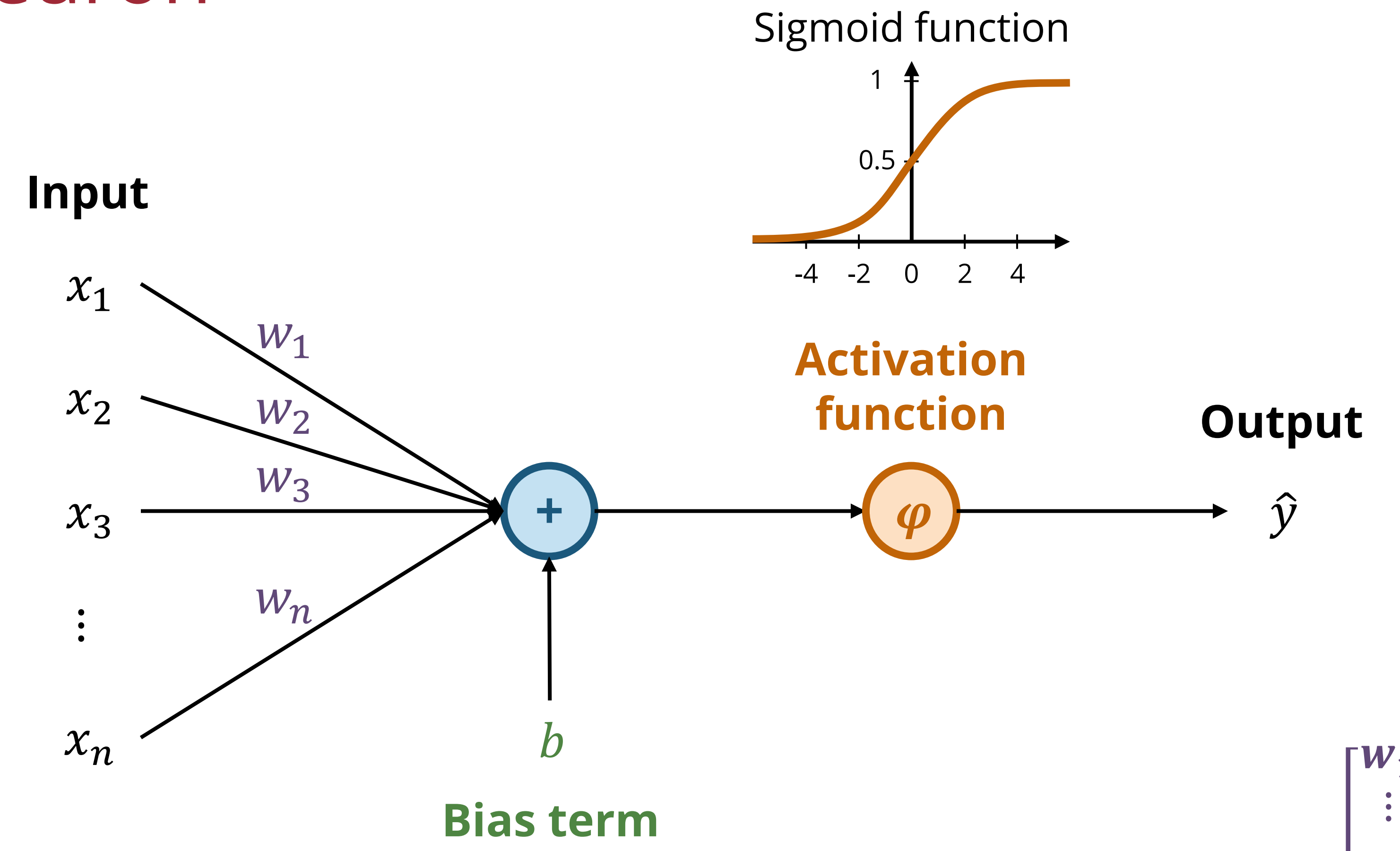
What is Deep Learning?

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



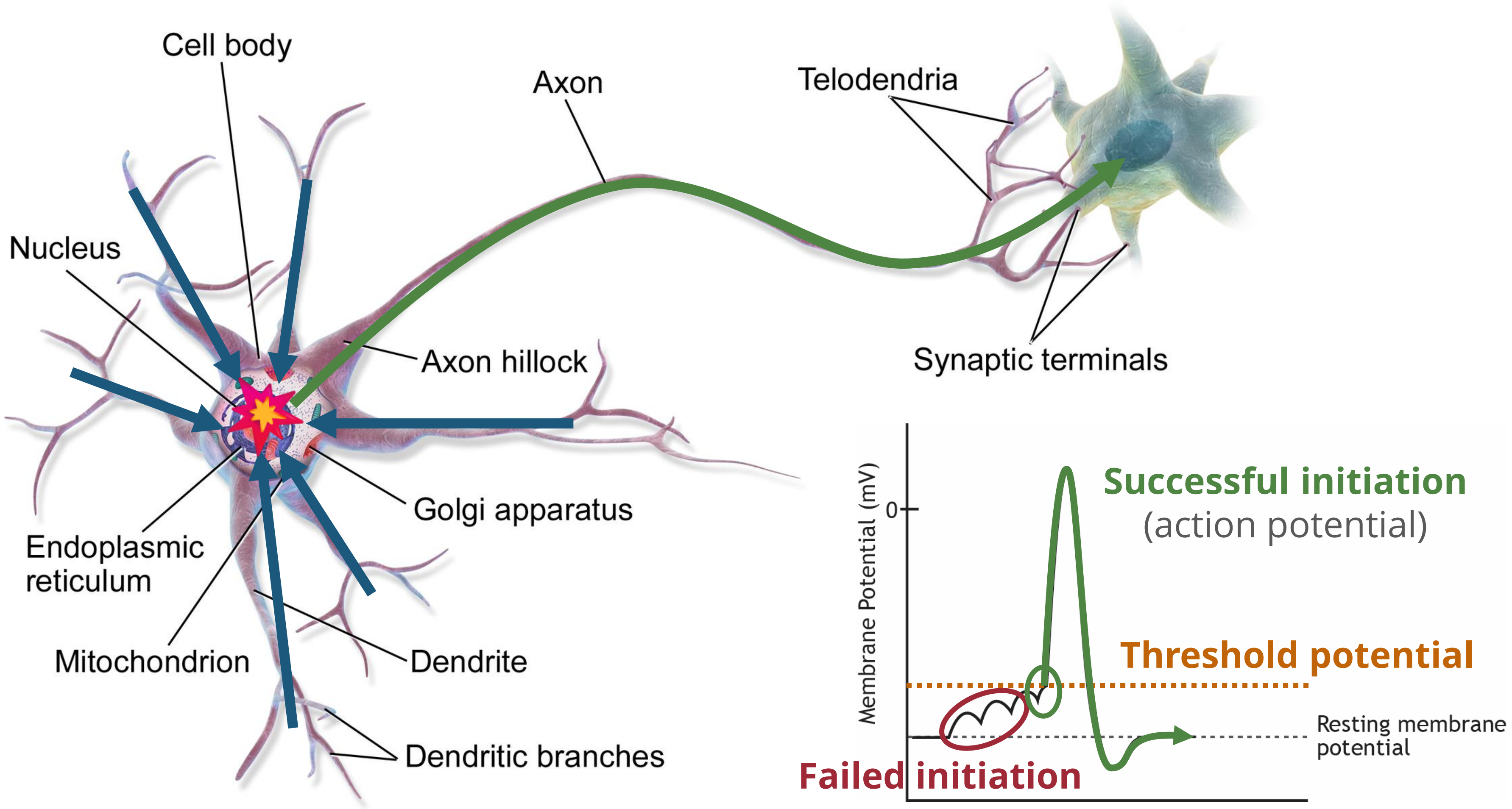
Neural Networks

Inside a Neuron

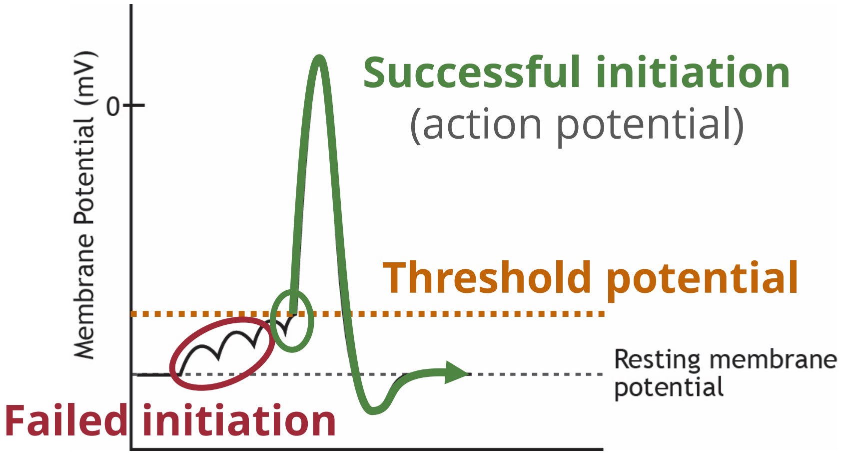


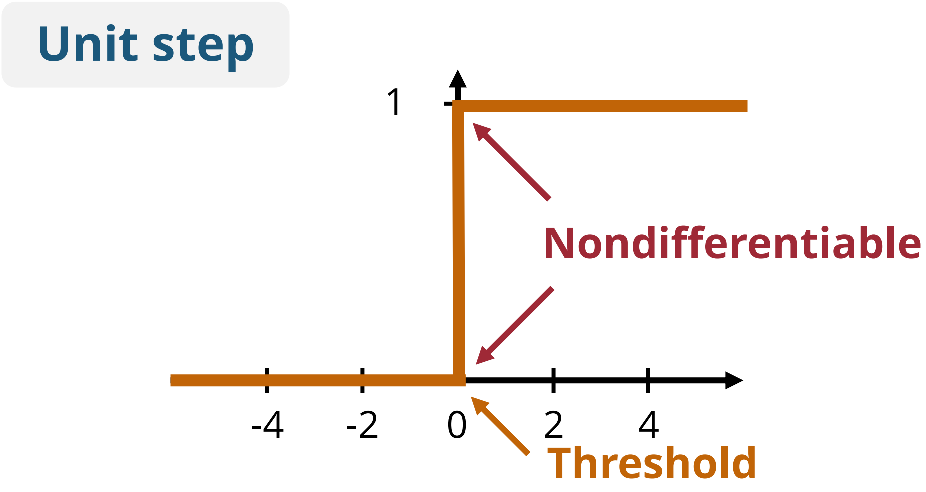
$$\hat{\boldsymbol{y}} = \boldsymbol{\varphi}(w_1x_1 + w_2x_2 + \cdots + w_nx_n + b) = \boldsymbol{\varphi}\left(\sum_{i=1}^{n} w_ix_i + b\right) = \boldsymbol{\varphi}(\underbrace{\mathbf{w}}_{\begin{bmatrix} \boldsymbol{w_1} \\ \vdots \\ \boldsymbol{w_n} \end{bmatrix}} \cdot \underbrace{\mathbf{x}}_{\begin{bmatrix} \boldsymbol{x_1} \\ \vdots \\ \boldsymbol{x_n} \end{bmatrix}} + b)$$

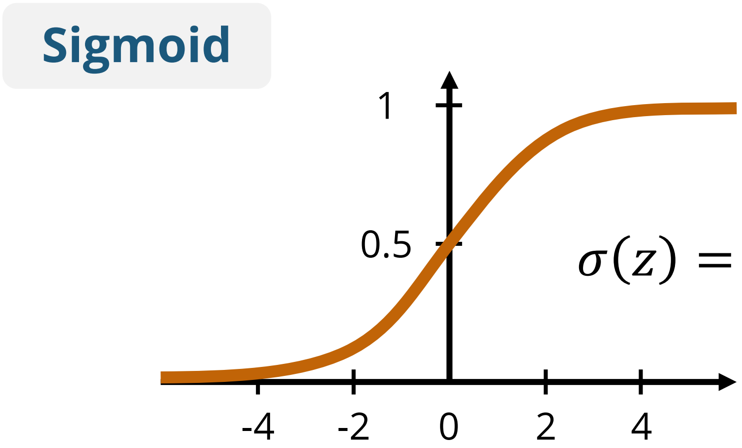
Human Neuron



Why Sigmoid?







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

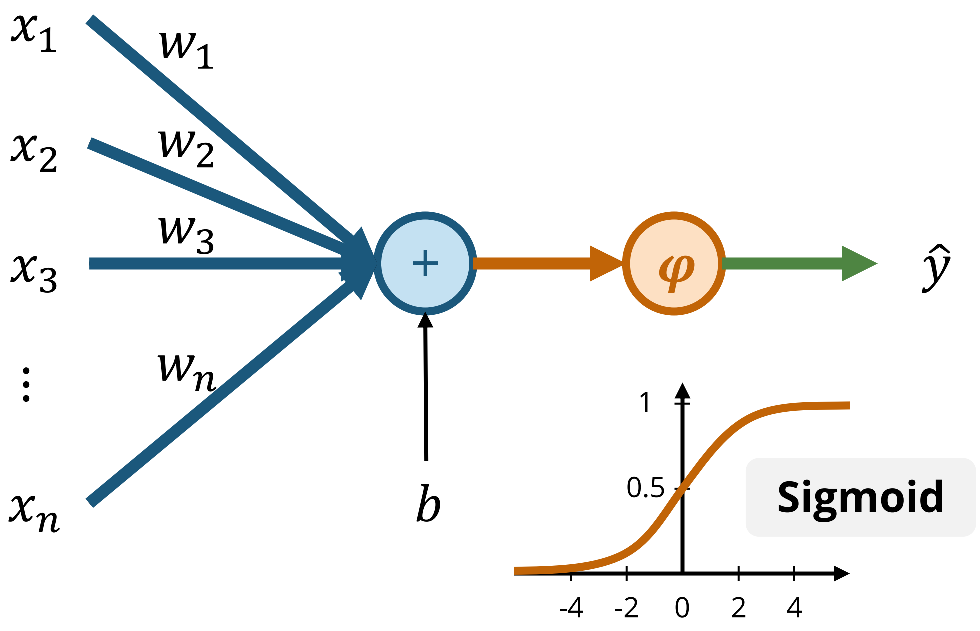
Why Bias Term?

- Allow nonzero outputs when all inputs are zero

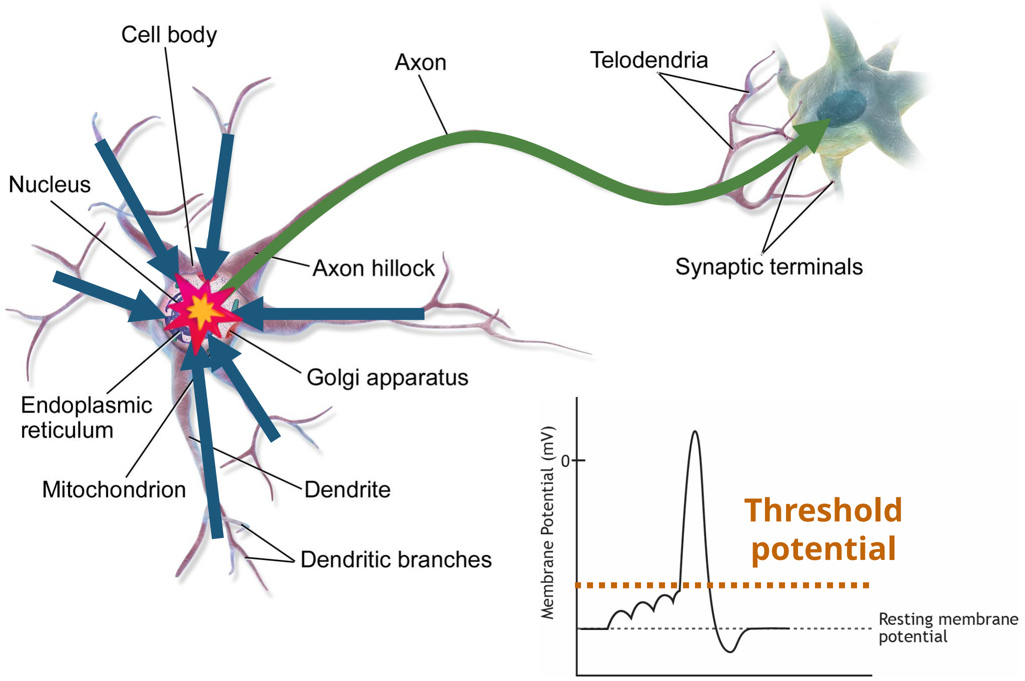
$$\hat{\boldsymbol{y}} = \boldsymbol{\varphi}(w_1 \cancelto{0}{x_1} + w_2 \cancelto{0}{x_2} + \cdots + w_n \cancelto{0}{x_n} + b) = \boldsymbol{\varphi}(b)$$

Artificial vs Human Neuron

Artificial neuron

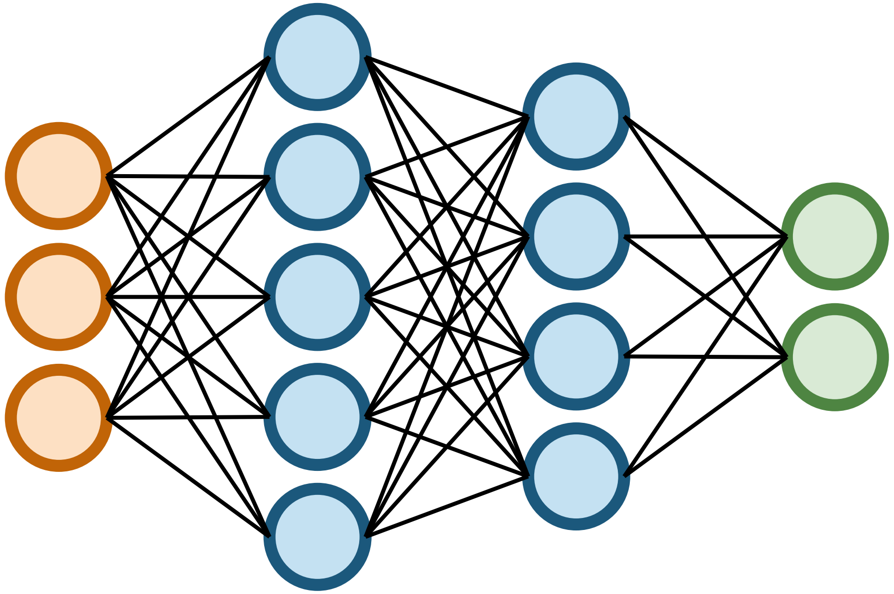


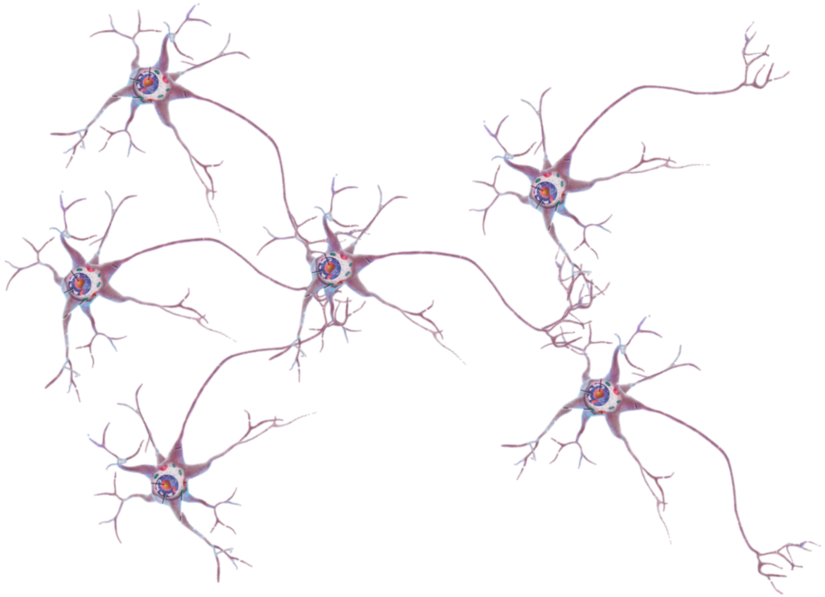
Human neuron



Artificial Neural Networks

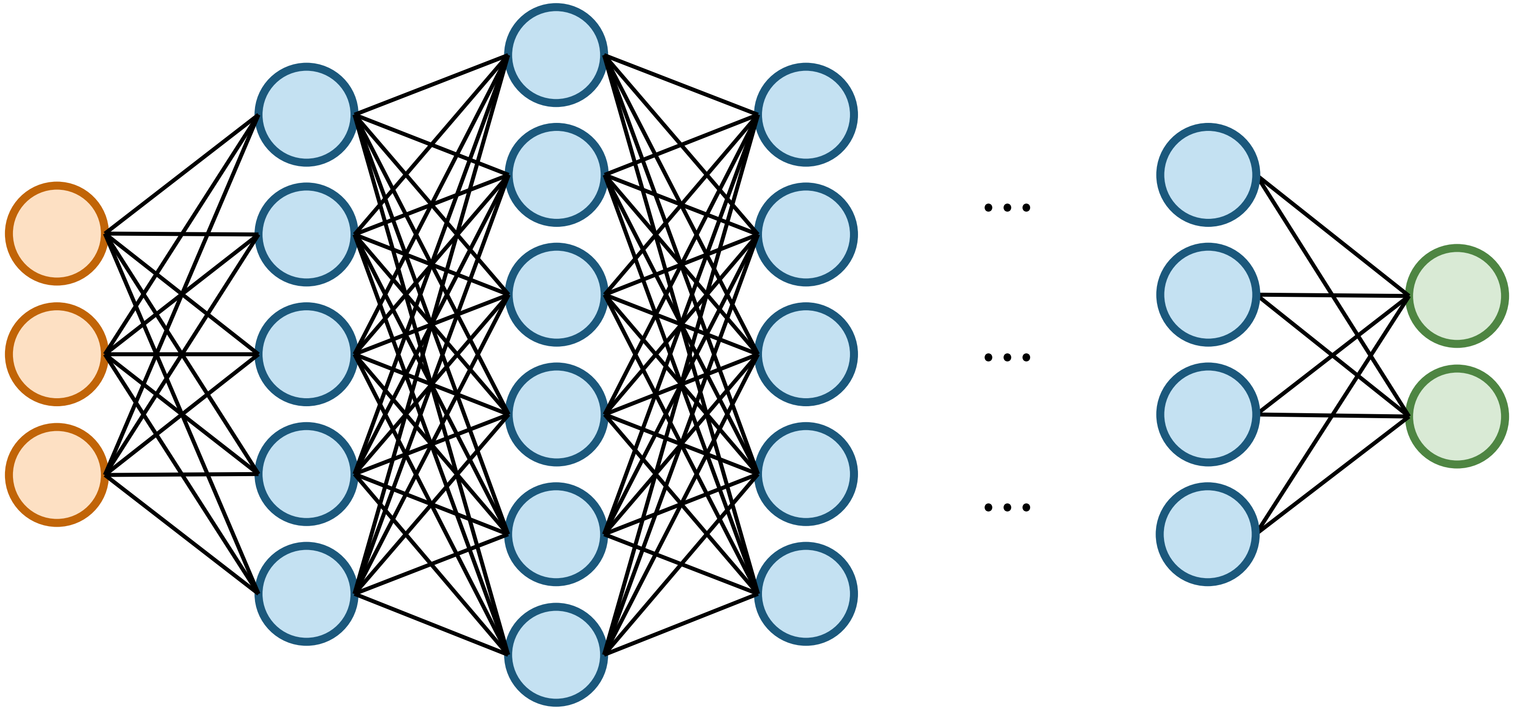
- Although inspired by human neural networks, artificial neural networks nowadays *do not work like human brains*
  - Lacking **functional hierarchy**, **high-level feedback loops**, **memory module**, etc.
  - Human brains work more like **spiking neural networks** → Efficiency!





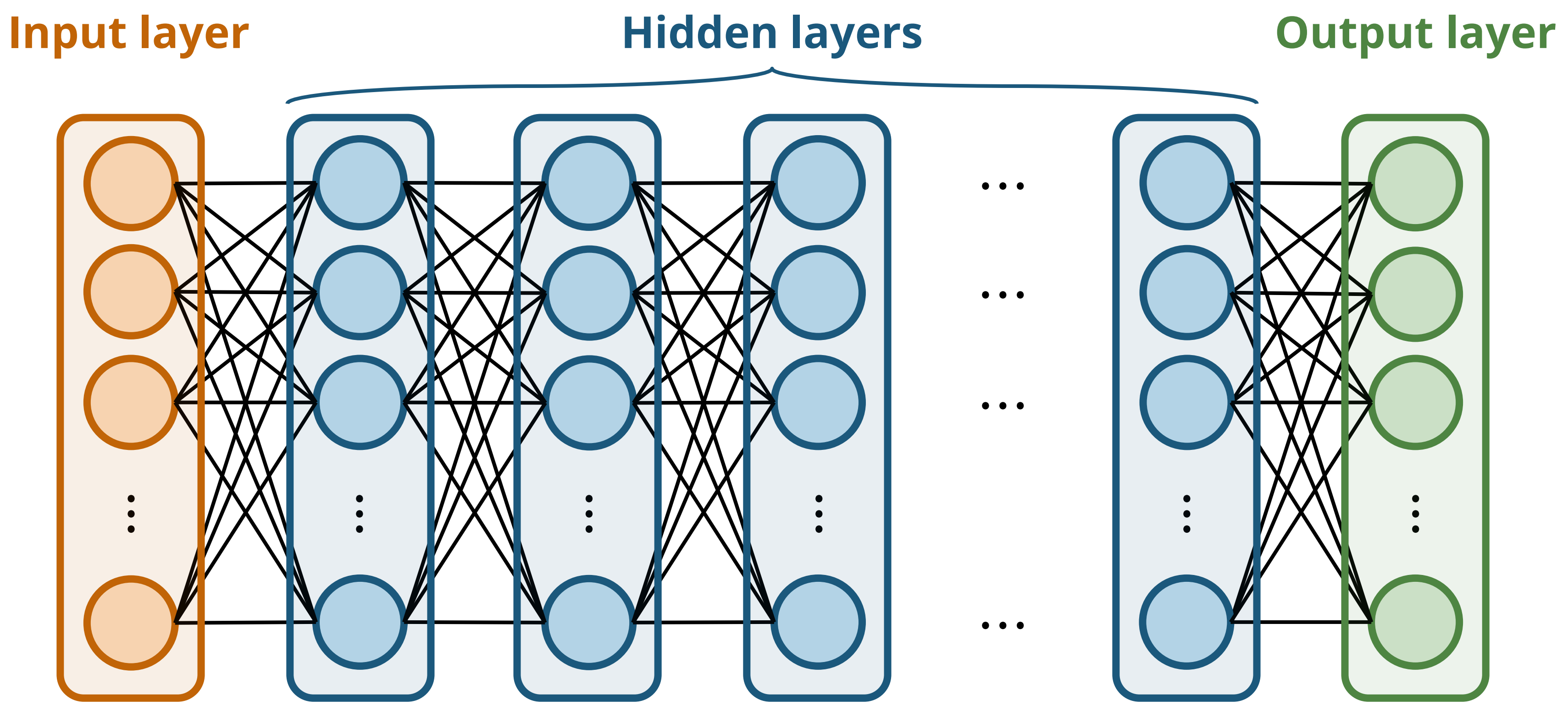
Fully Connected Feedforward Network

- Most basic form of deep neural networks



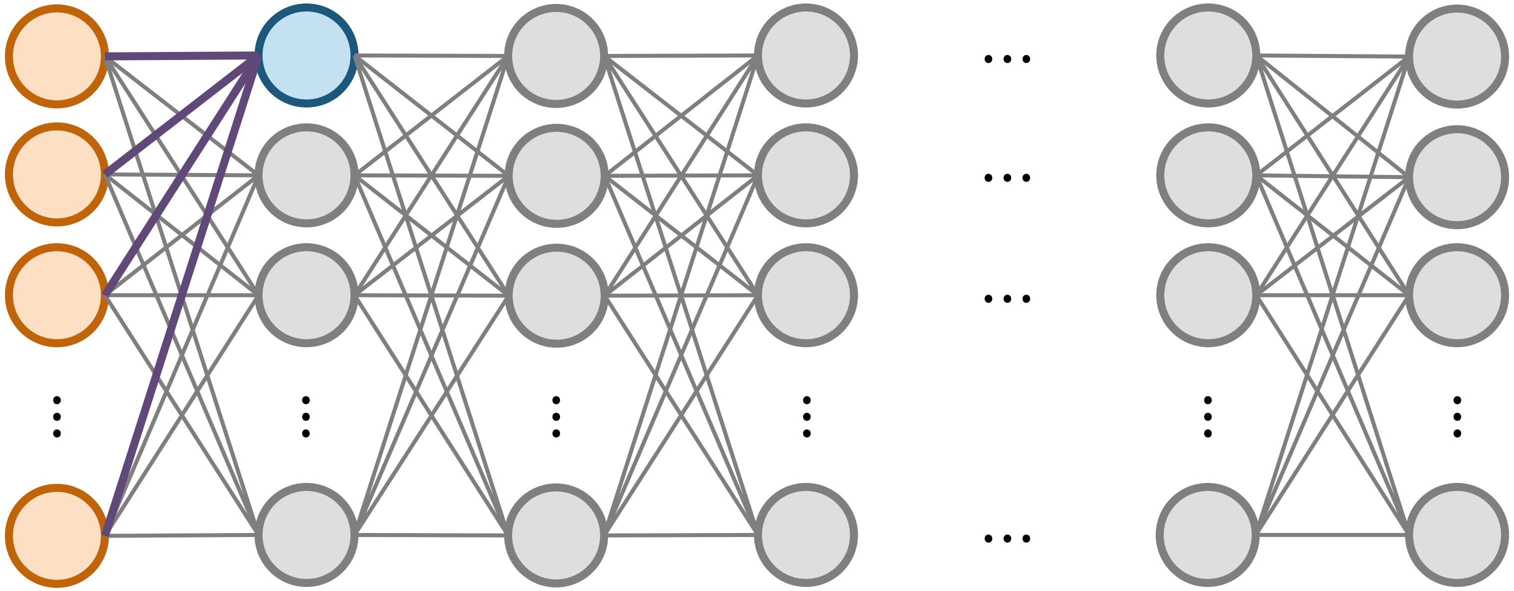
Math Formulation

Math Formulation



Math Formulation

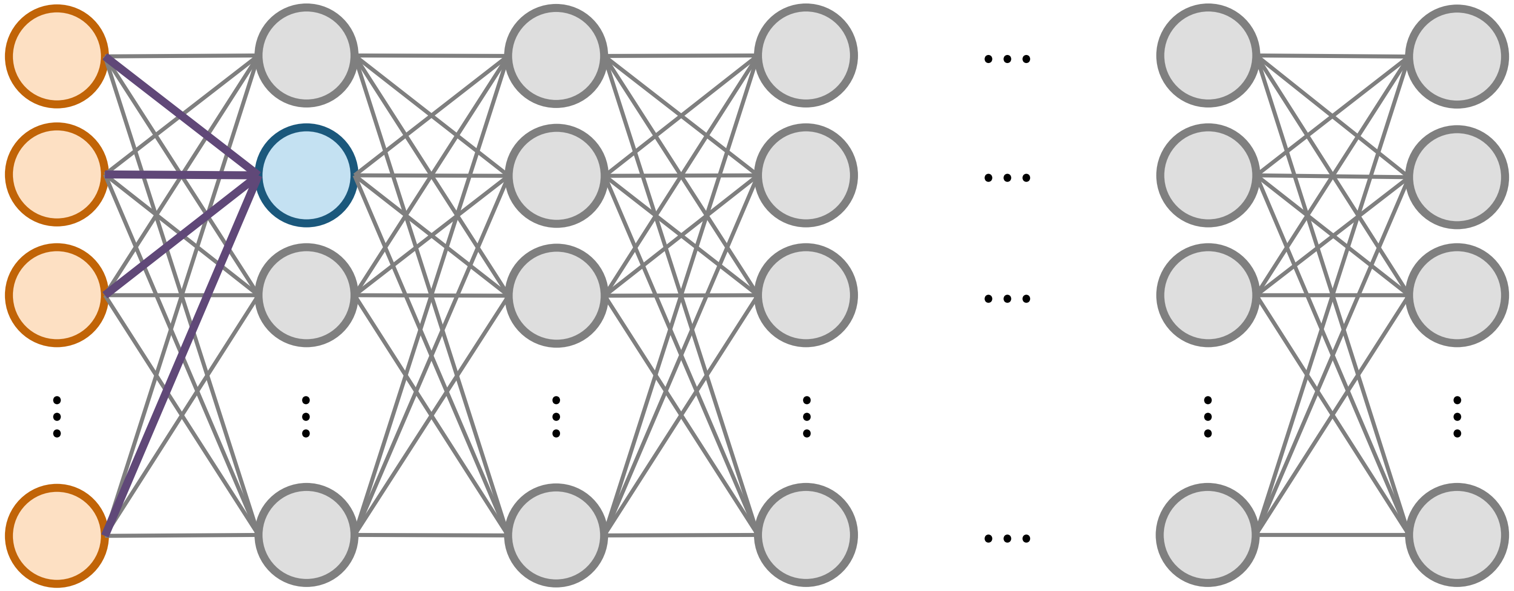
$$h_1 = \varphi(\mathbf{w}_1 \cdot \mathbf{x} + b_1)$$



$$\underbrace{\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}}_{\mathbf{x}} \quad \underbrace{\begin{bmatrix} w_{1,1} \\ \vdots \\ w_{1,n} \end{bmatrix}}_{\mathbf{w}_1}$$

Math Formulation

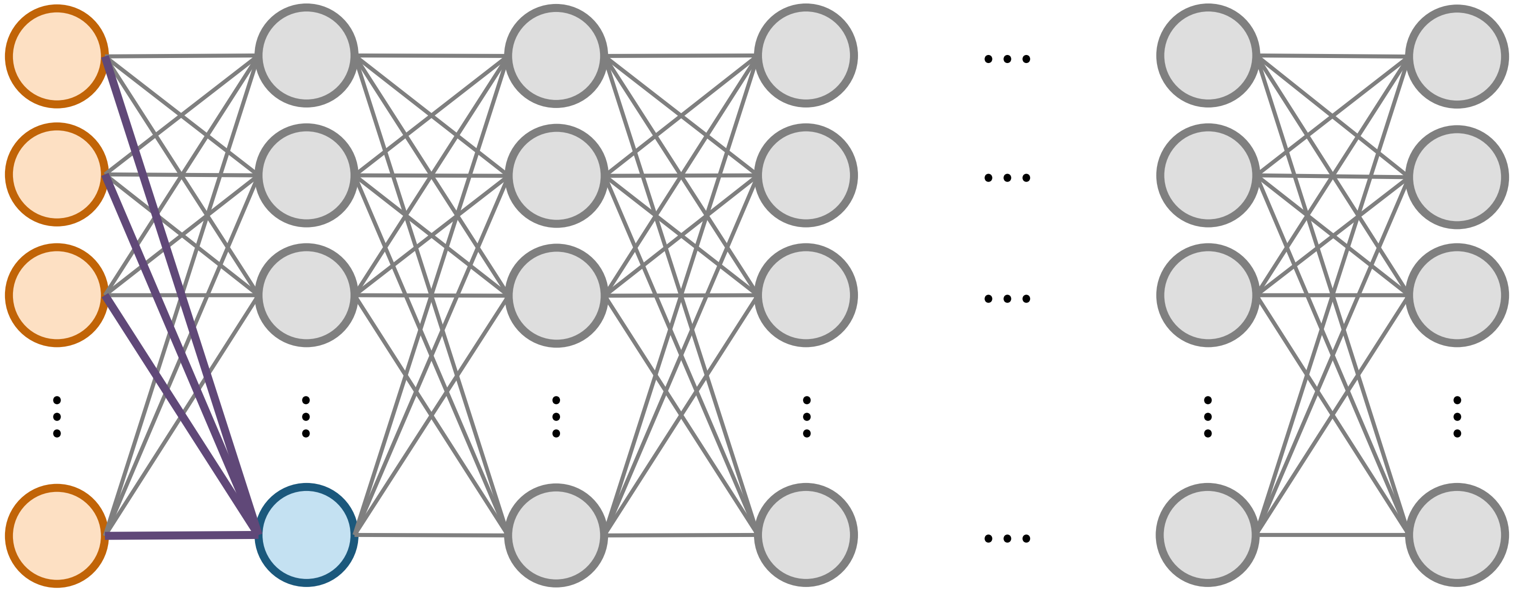
$$h_2 = \varphi(\mathbf{w}_2 \cdot \mathbf{x} + b_2)$$



$$\underbrace{\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}}_{\mathbf{x}} \quad \underbrace{\begin{bmatrix} w_{2,1} \\ \vdots \\ w_{2,n} \end{bmatrix}}_{\mathbf{w}_2}$$

Math Formulation

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

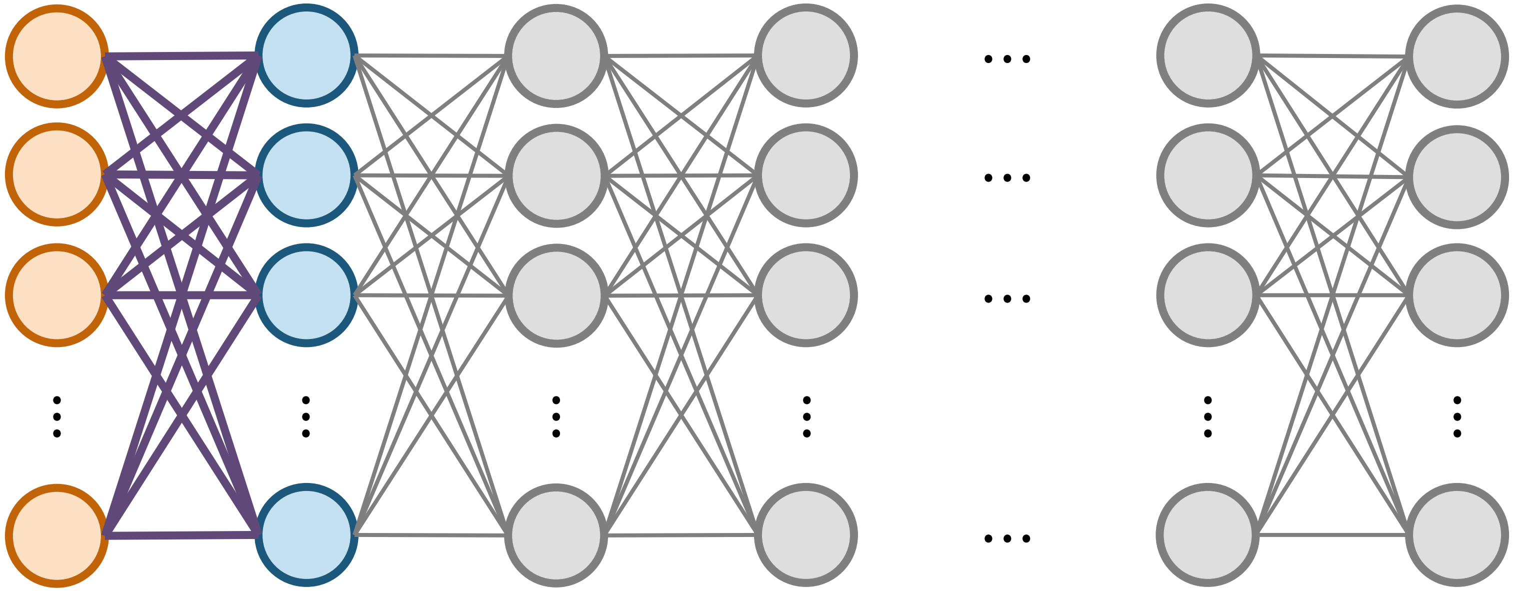


$$\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \begin{bmatrix} w_{n,1} \\ \vdots \\ w_{n,n} \end{bmatrix}$$

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

Math Formulation

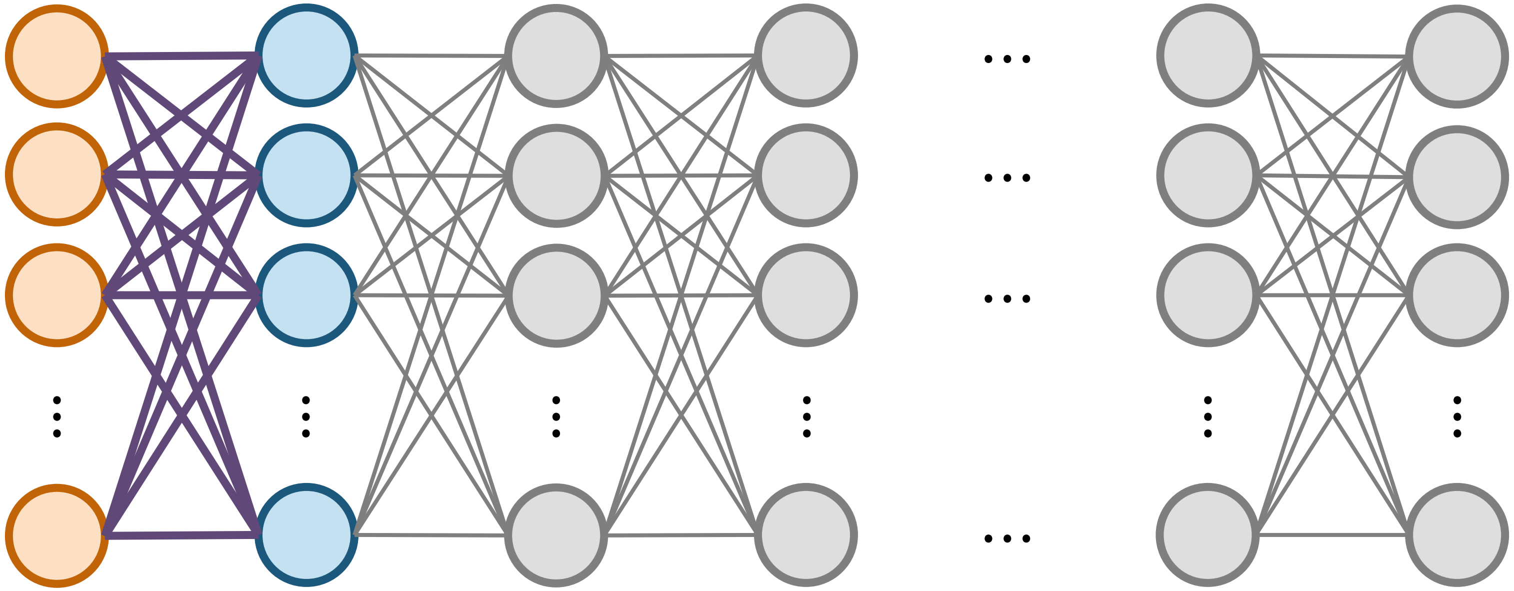
$$\mathbf{h} = \varphi(W\mathbf{x} + \mathbf{b})$$



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

Math Formulation

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

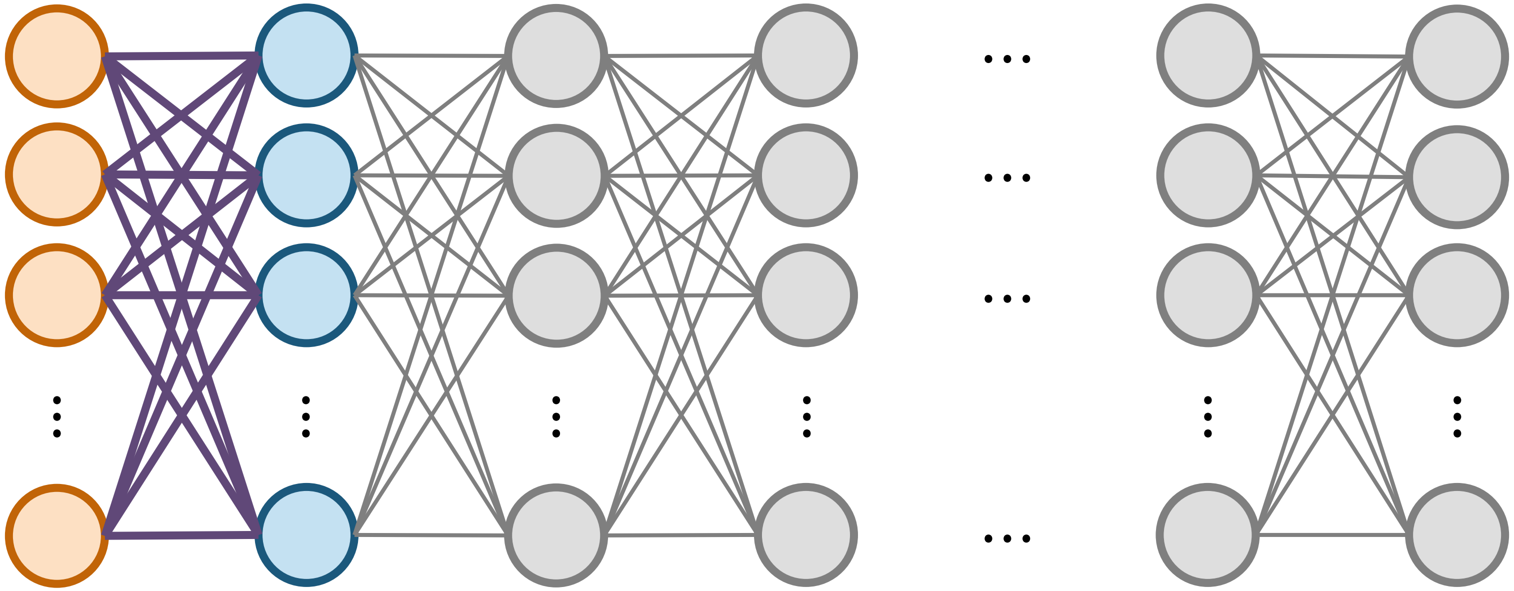


$$\underbrace{\begin{bmatrix} \boxed{w_{1,1} \quad \cdots \quad w_{1,n}} \\ \vdots \quad \ddots \quad \vdots \\ w_{n,1} \quad \cdots \quad w_{n,n} \end{bmatrix}}_{W} \underbrace{\boxed{\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} \boxed{b_1} \\ \vdots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

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

Math Formulation

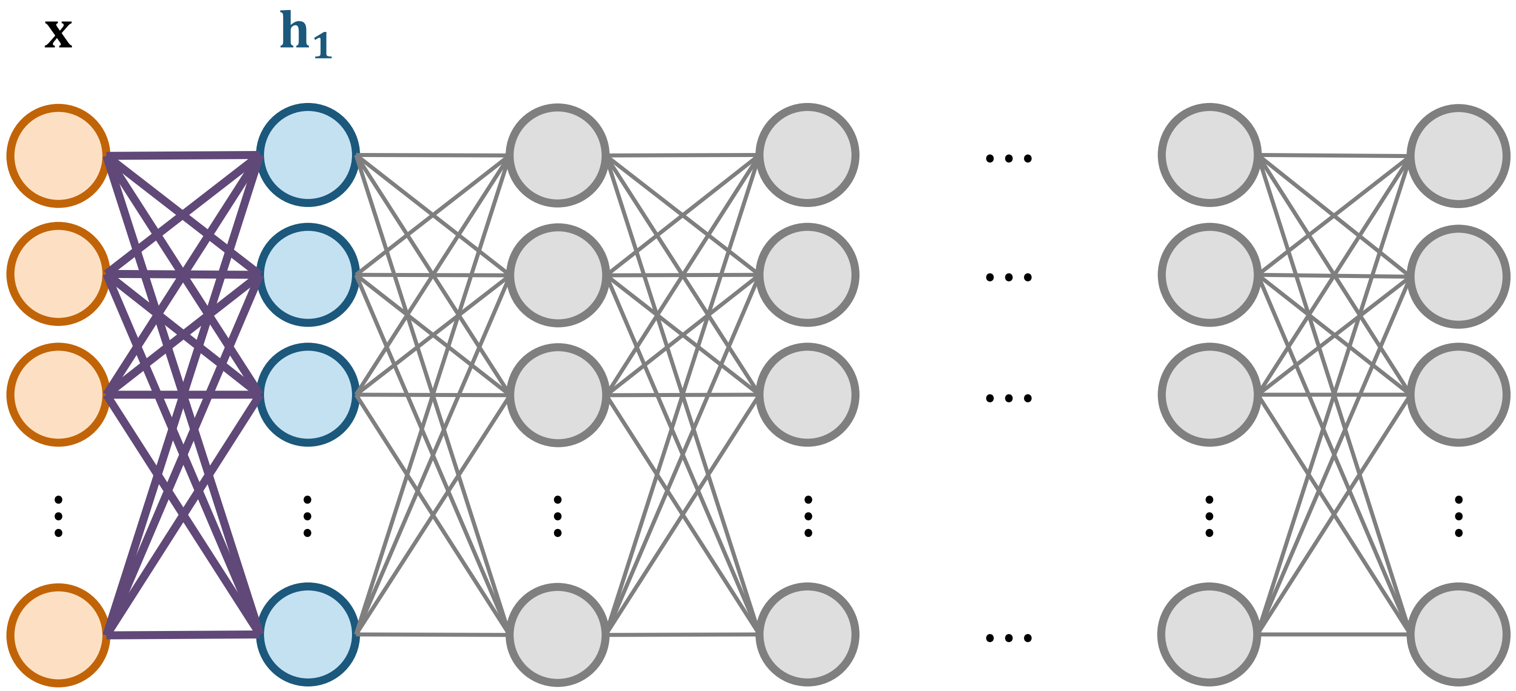
$$\mathbf{h} = \varphi(W\mathbf{x} + \mathbf{b})$$



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

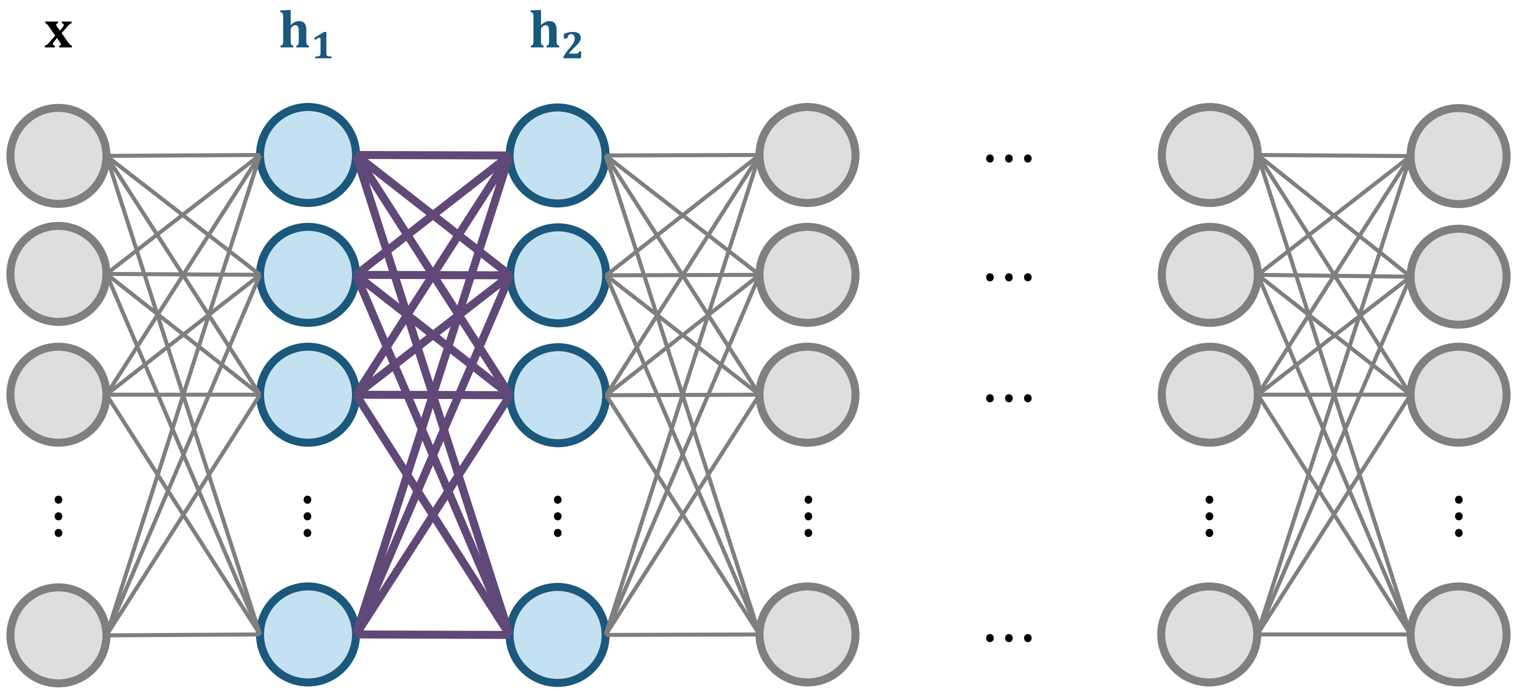
$$h_n = \varphi(\mathbf{w}_n \cdot \mathbf{x} + b_n)$$

Math Formulation



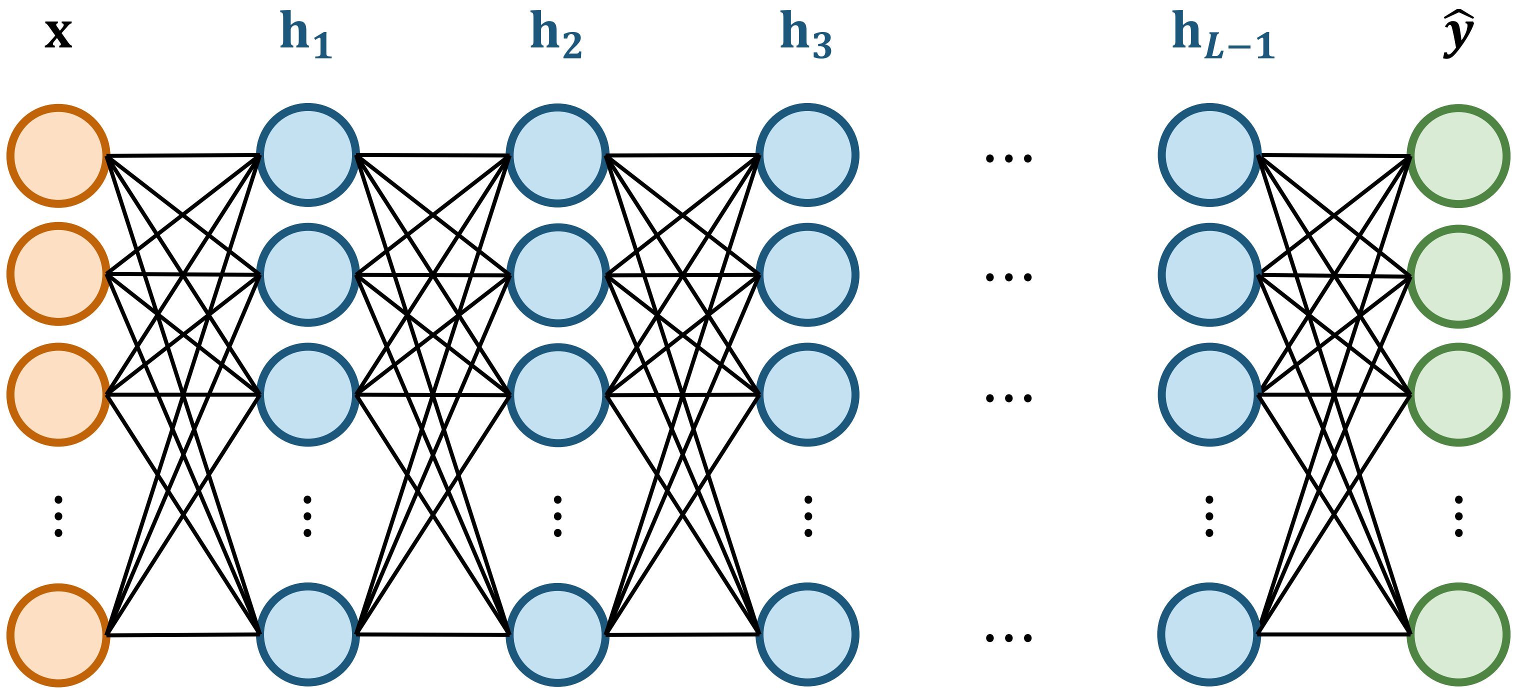
$$\mathbf{h}_1 = \varphi(W_1\mathbf{x} + \mathbf{b}_1)$$

Math Formulation



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

Math Formulation



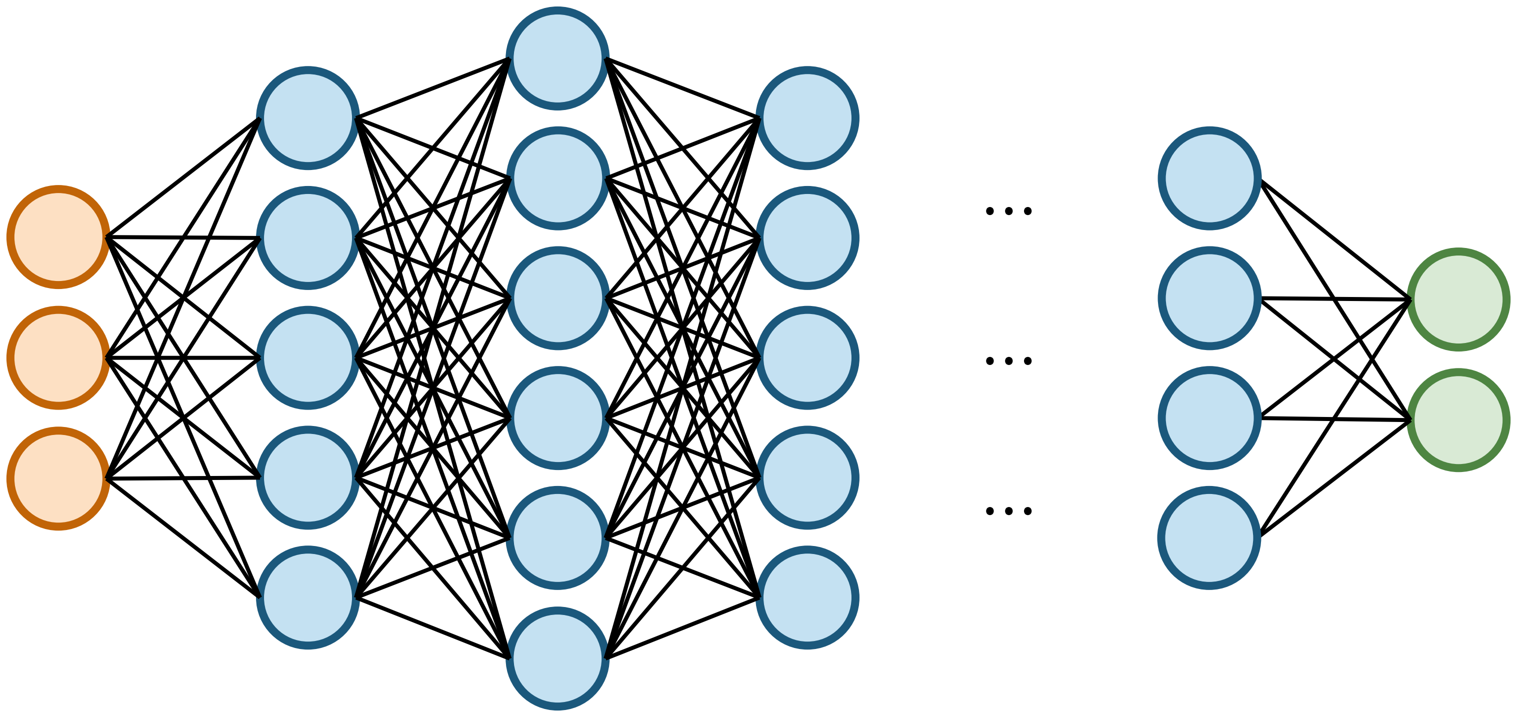
$$\mathbf{h}_1 = \varphi(W_1\mathbf{x} + \mathbf{b}_1)$$

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

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

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

Fully Connected Feedforward Network



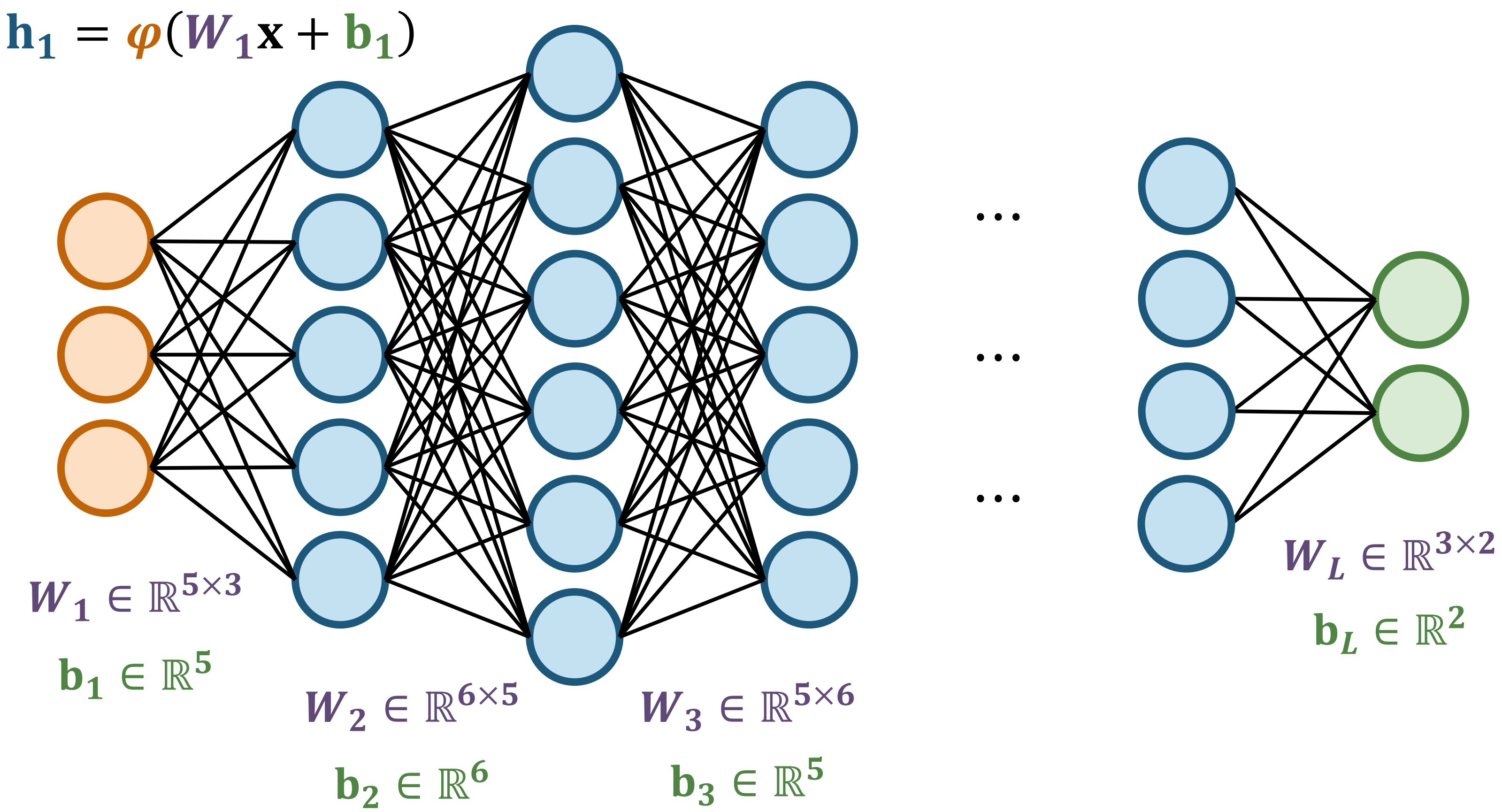
$\mathbf{h}_1 = \varphi(W_1 \mathbf{x} + \mathbf{b}_1)$

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

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

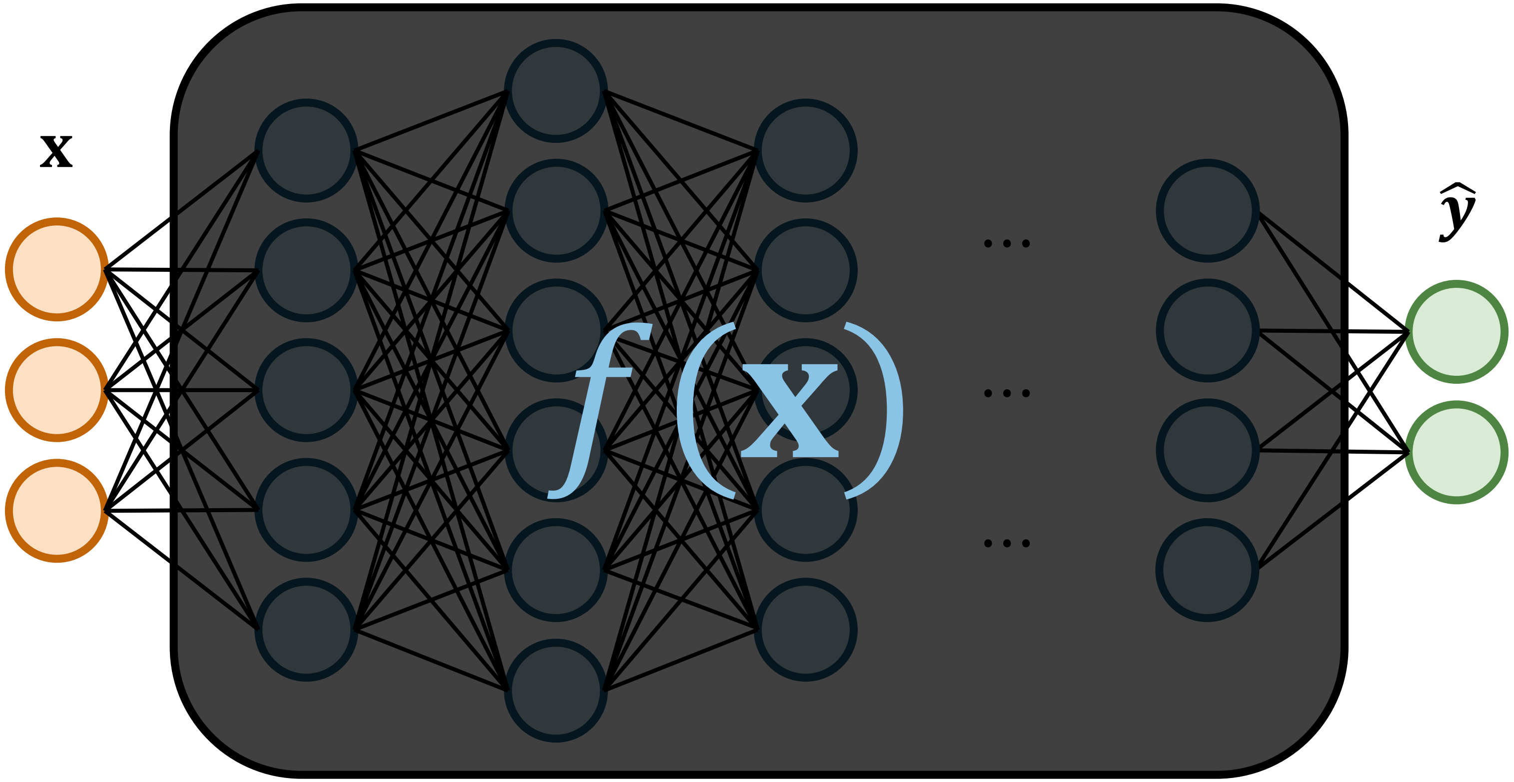
$\mathbf{h}_3 = \varphi(W_2 \mathbf{h}_2 + \mathbf{b}_2)$

Fully Connected Feedforward Network



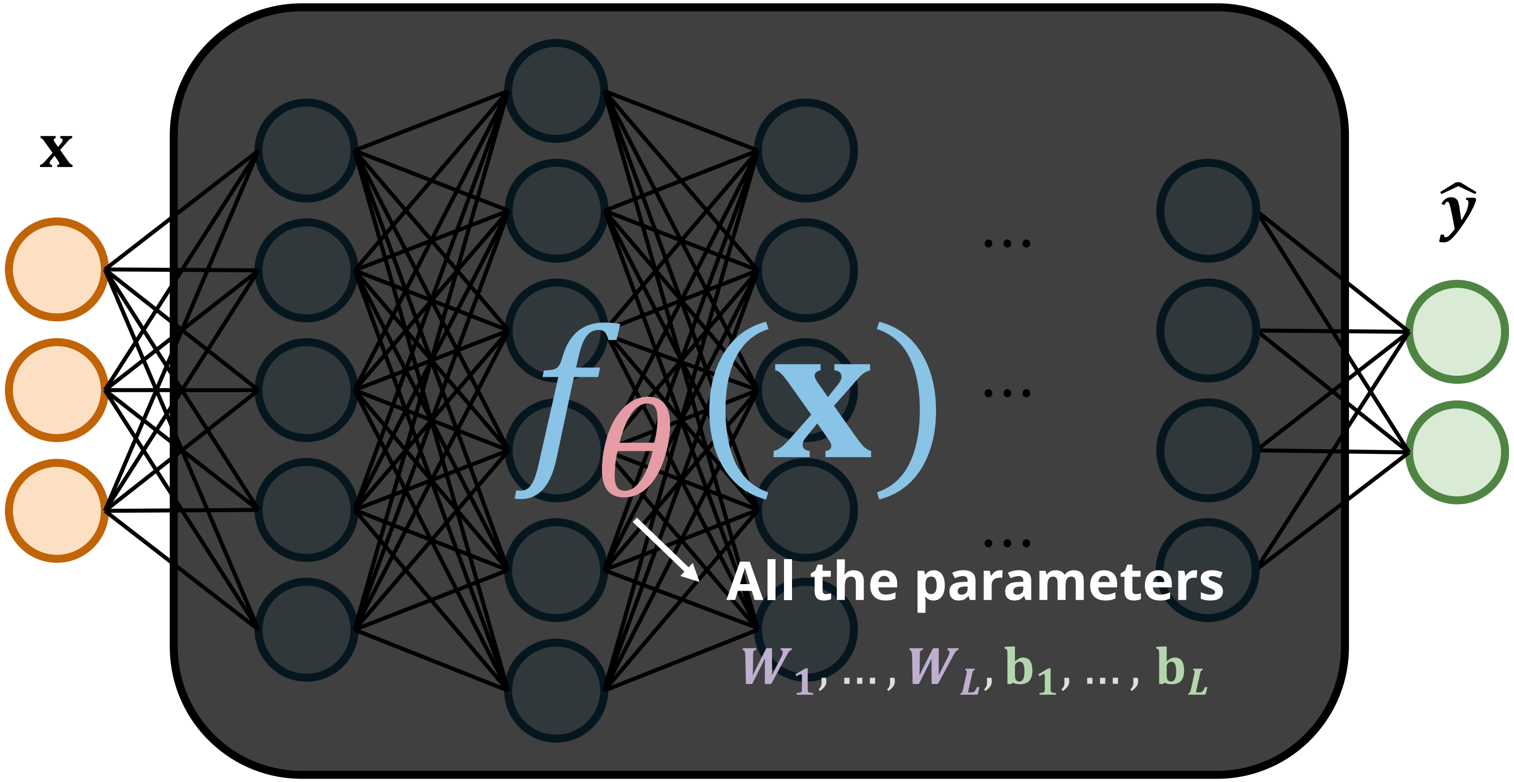
Neural Networks are Parameterized Functions

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

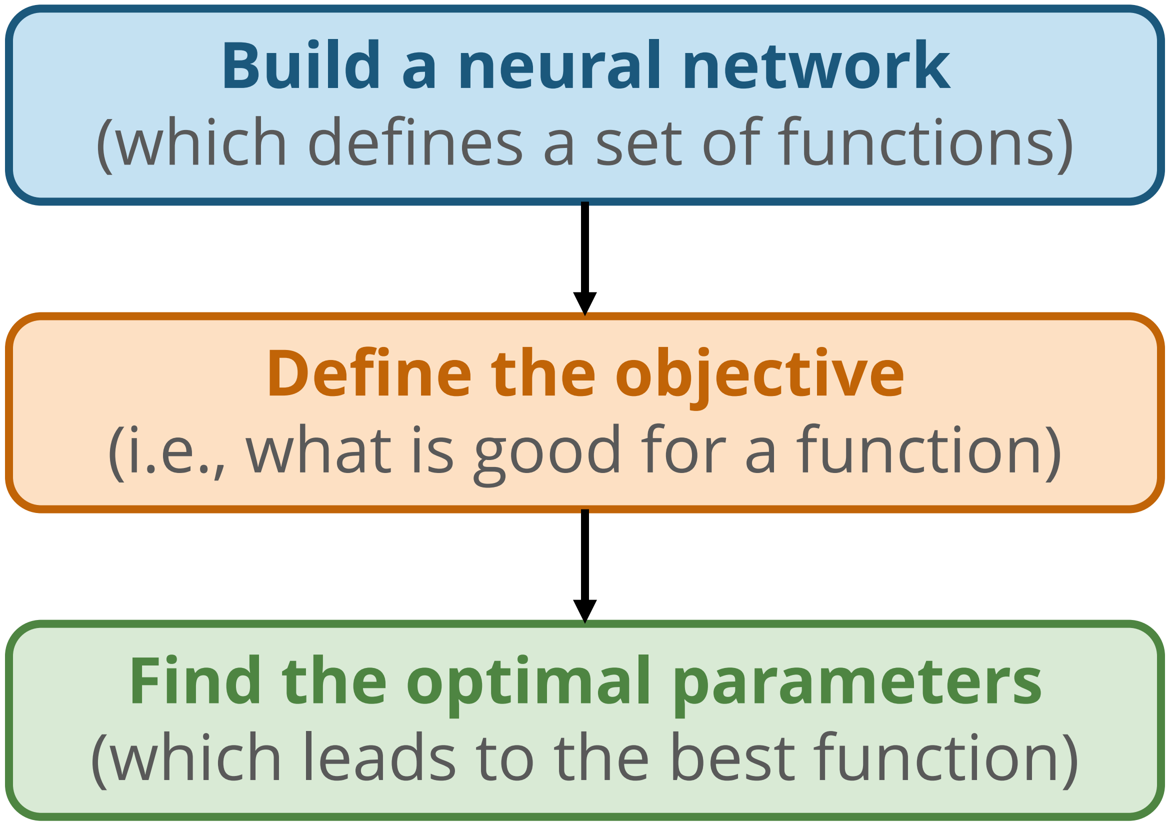


Neural Networks are Parameterized Functions

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

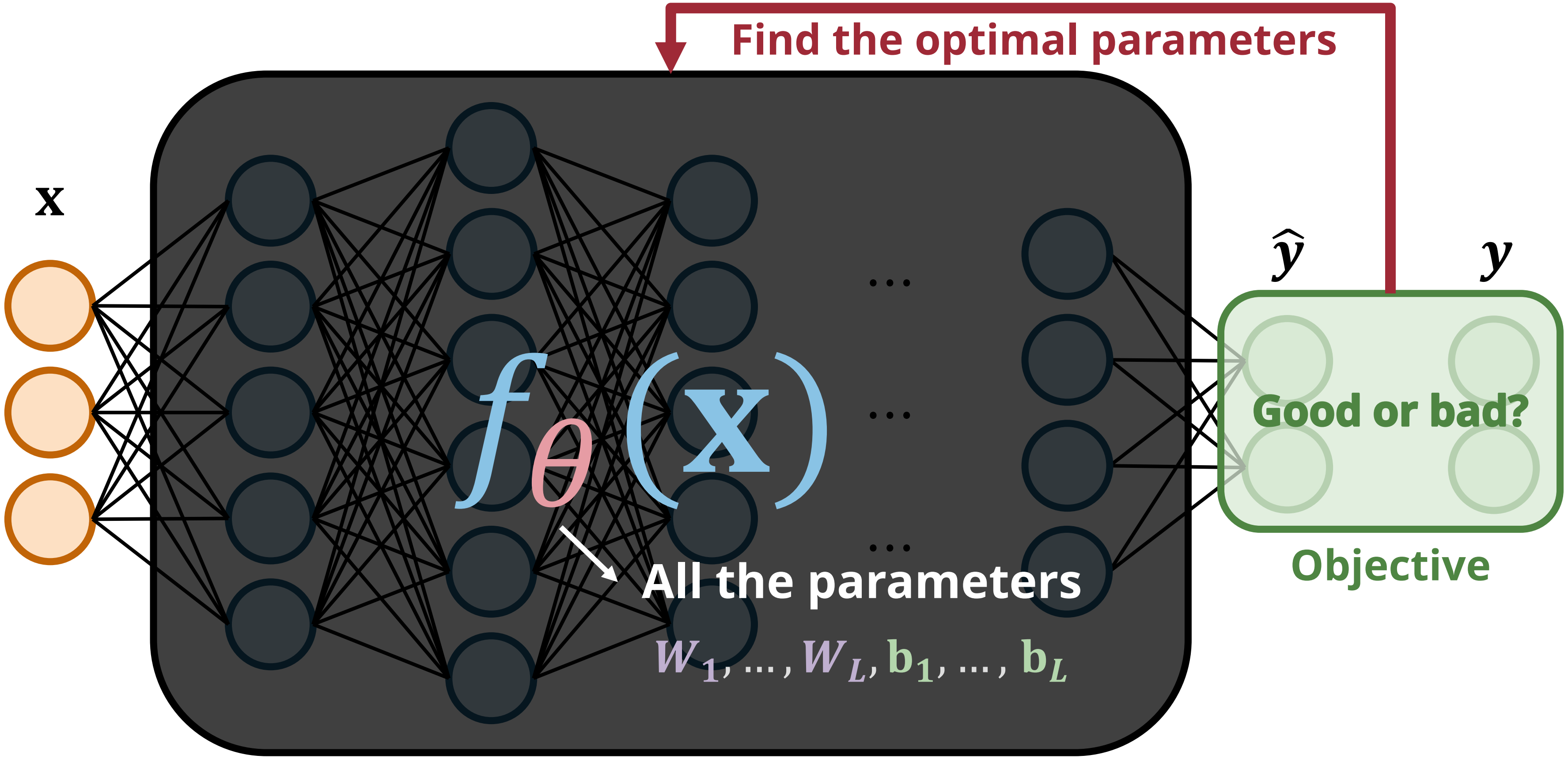


(Preview) Training a Neural Network



Neural Networks are Parameterized Functions

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



Activation Functions

Why do We Need Activation Functions?

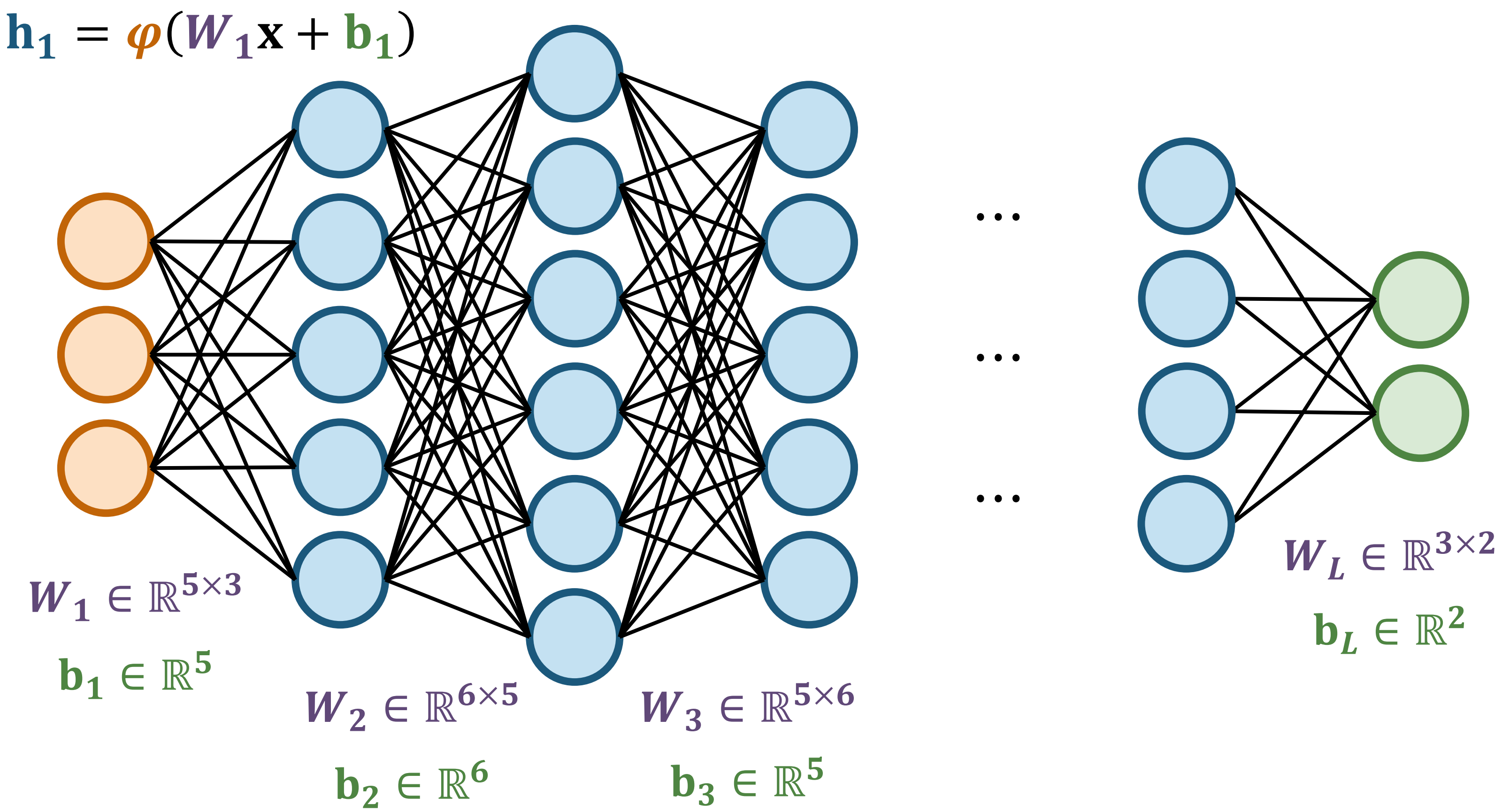
- Activation functions introduce **nonlinearity** to a neural network
- A linear function is a **weighted sum of the inputs** (plus a bias term)

$$f(x_1, x_2, \dots, x_n) = a_1 x_1 + a_2 x_2 + a_3 x_3 + \cdots + a_n x_n + b$$

- Examples of nonlinear functions:
  - $f(x_1) = \frac{1}{x_1}$
  - $f(x_1) = x_1^2$
  - $f(x_1) = e^x$
  - $f(x_1, x_2) = x_1 x_2$

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**

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

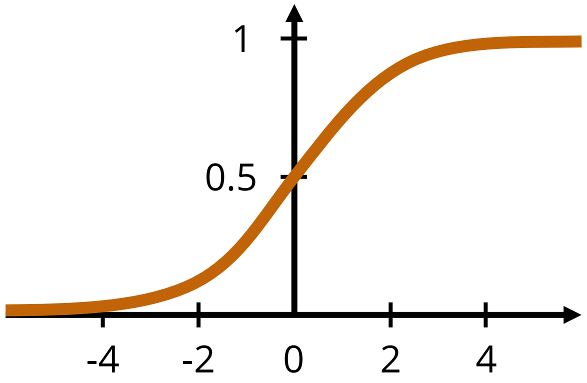
$$\downarrow$$

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

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

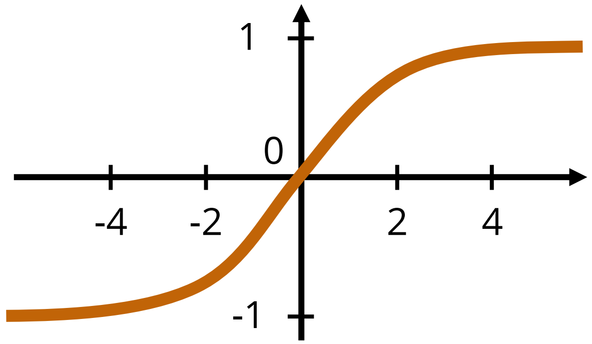
Commonly Used Activation Functions

Sigmoid



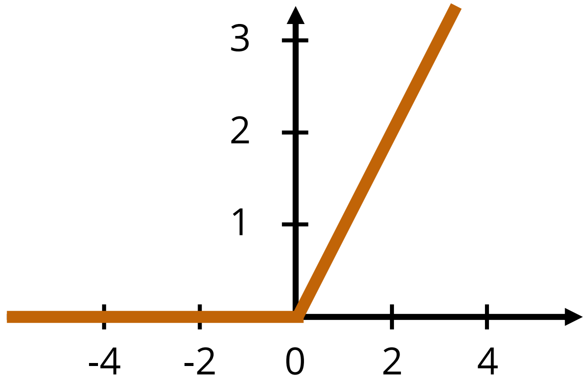
$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

tanh



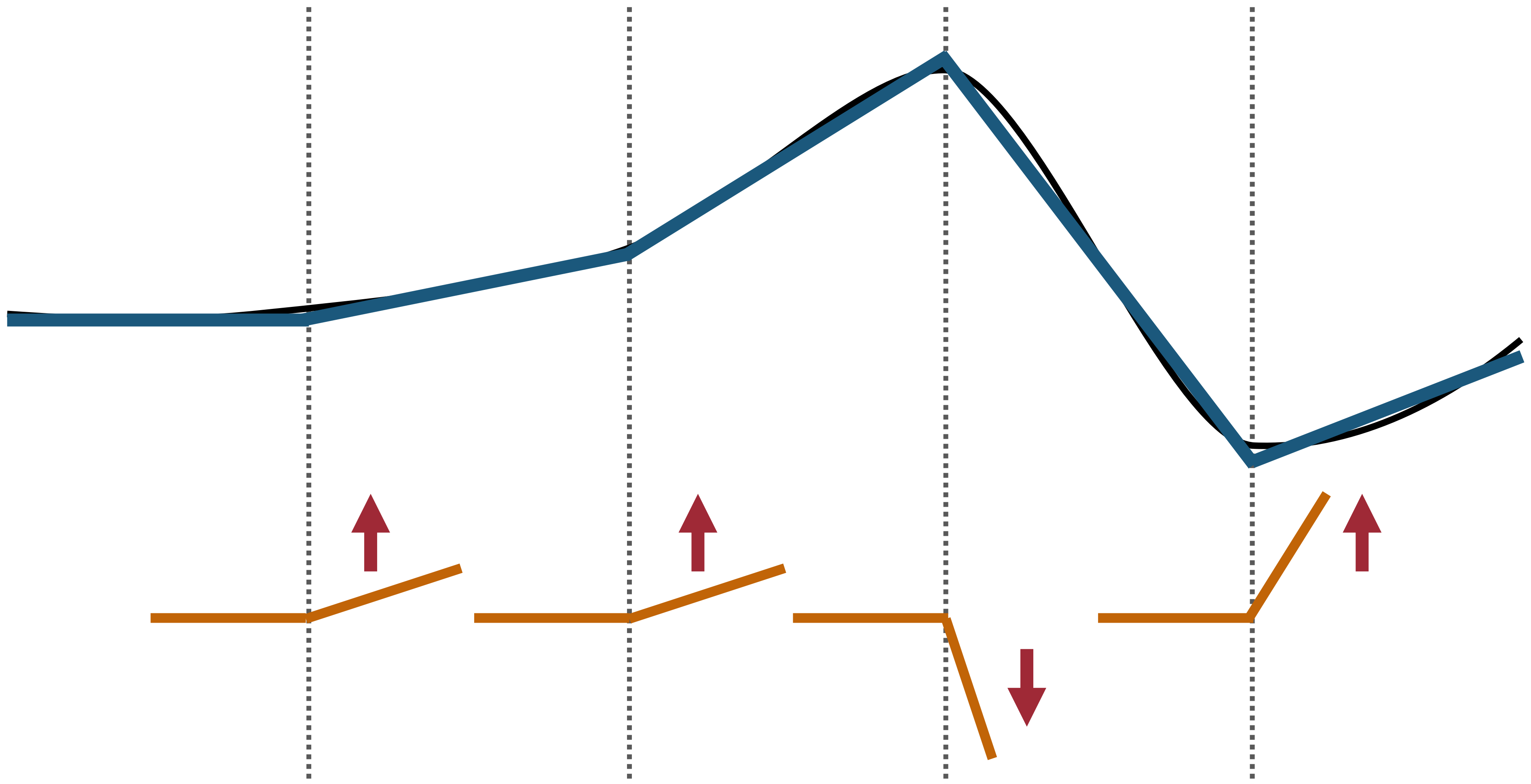
$$\tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$

ReLU



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

ReLUs & Piecewise Linear Functions

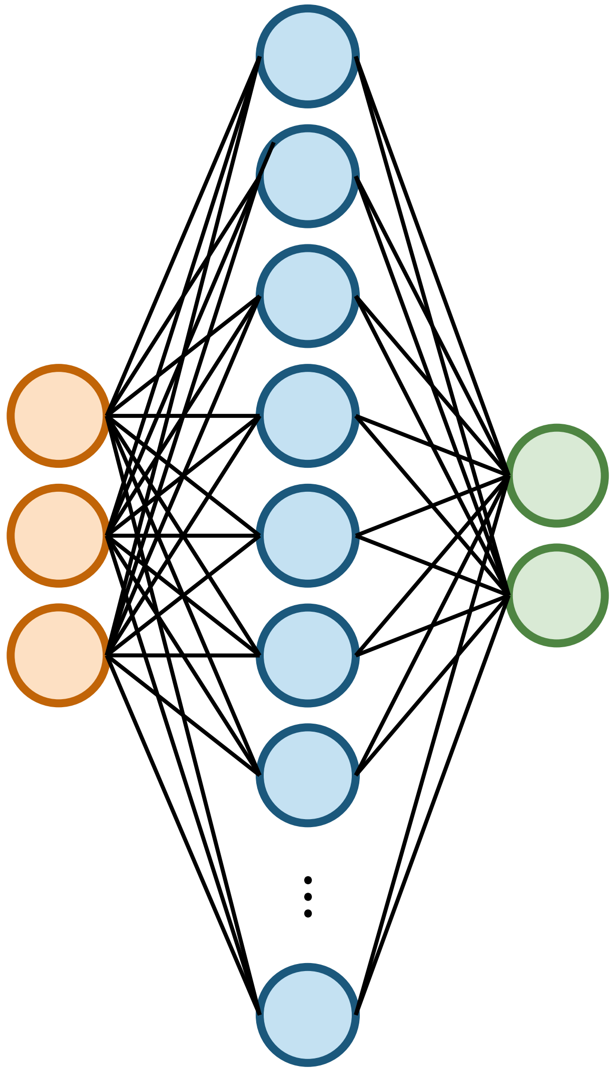


Expressiveness of Neural Networks

Universal Approximation Theorem

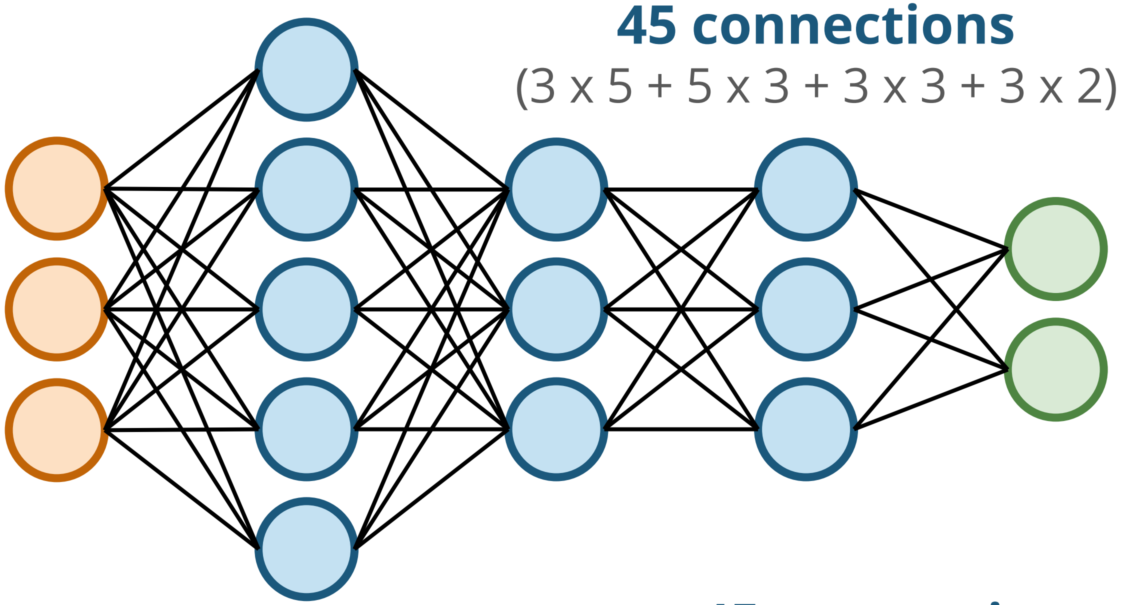
- A neural network with **one hidden layer** can **approximate any continuous function** given **sufficient hidden neurons** and **appropriate activation functions**
  - Sigmoid, ReLUs are good activation functions

Then why do we want to go deep?

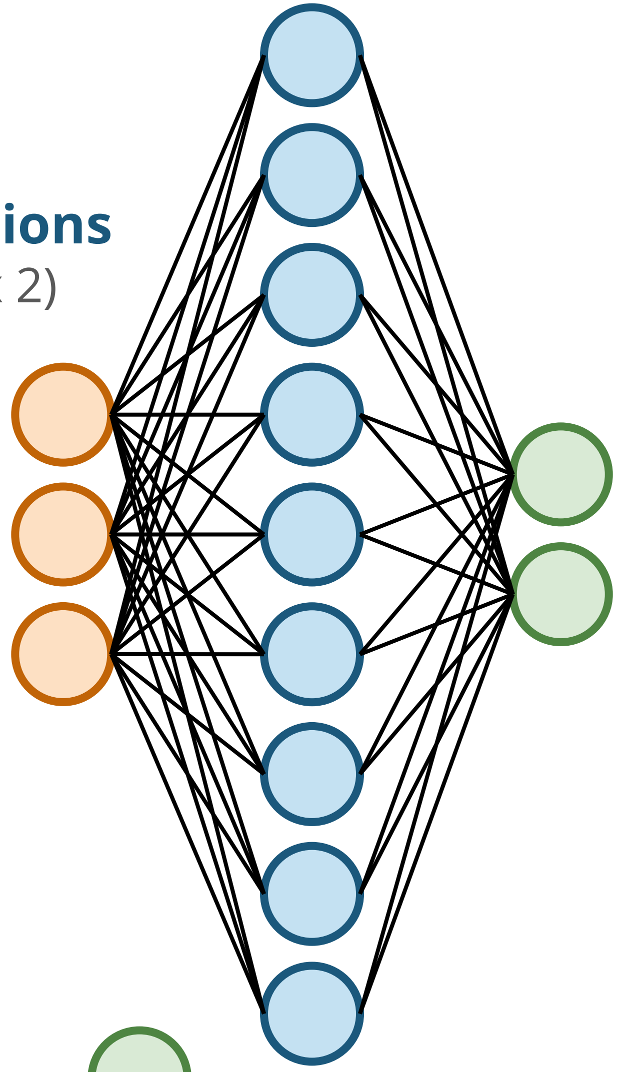


Shallow vs Deep Neural Networks

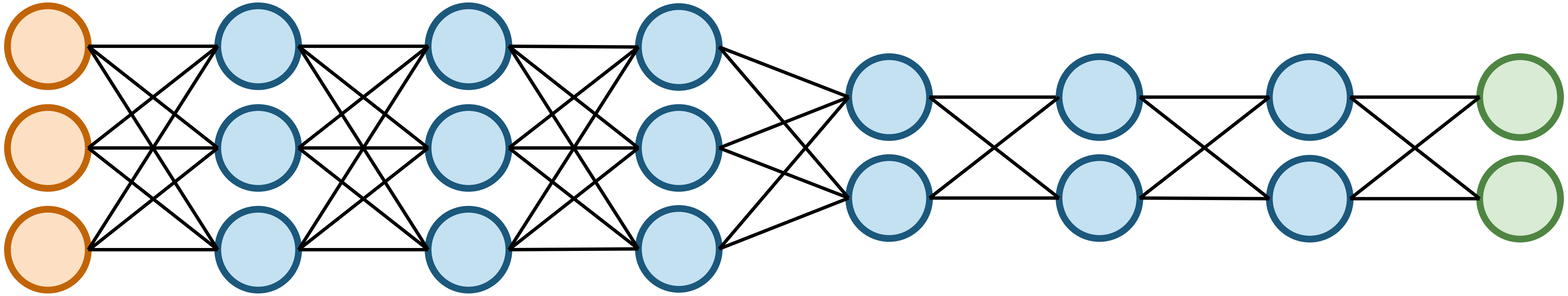
45 connections
(3 x 5 + 5 x 3 + 3 x 3 + 3 x 2)



45 connections
(3 x 9 + 9 x 2)

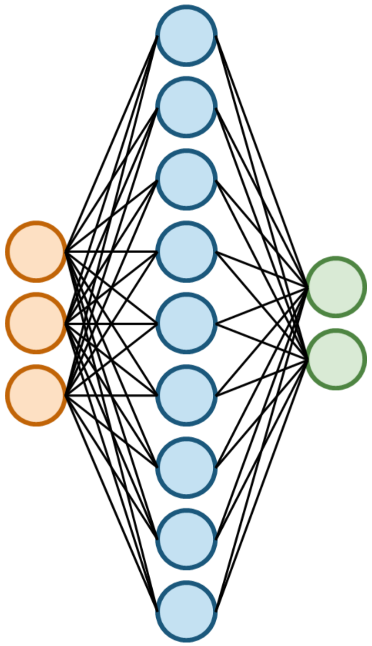


45 connections
(3 x 3 + 3 x 3 + 3 x 3 + 3 x 2 + 2 x 2 + 2 x 2 + 2 x 2)



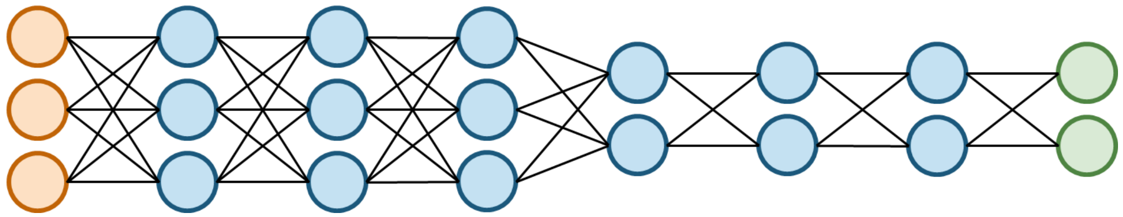
Shallow vs Deep Neural Networks – In Practice

Shallow neural nets



Less expressive
(less parameter efficient)

Deep neural nets

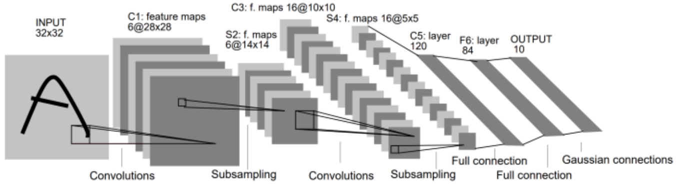


More expressive
(more parameter efficient)

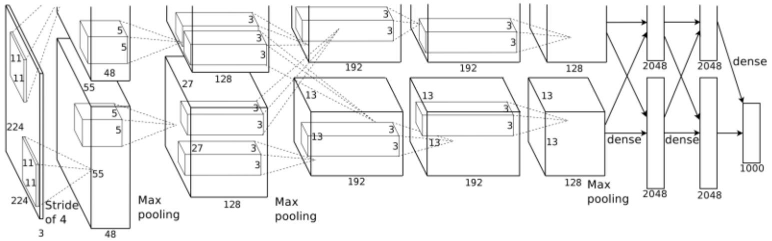
How Deep is Deep Enough?

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

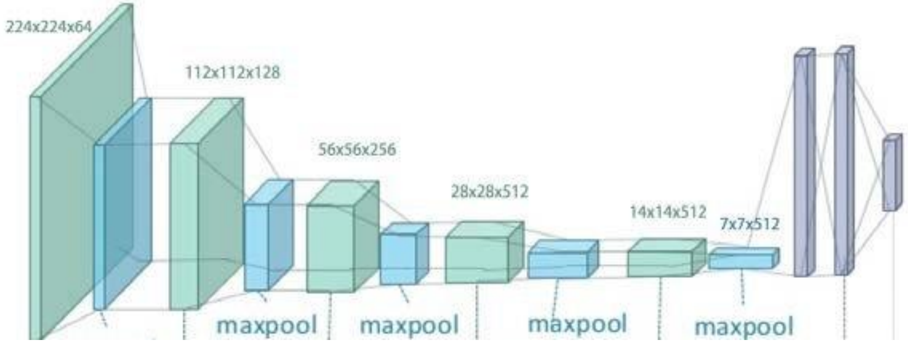
LeNet
(1998)



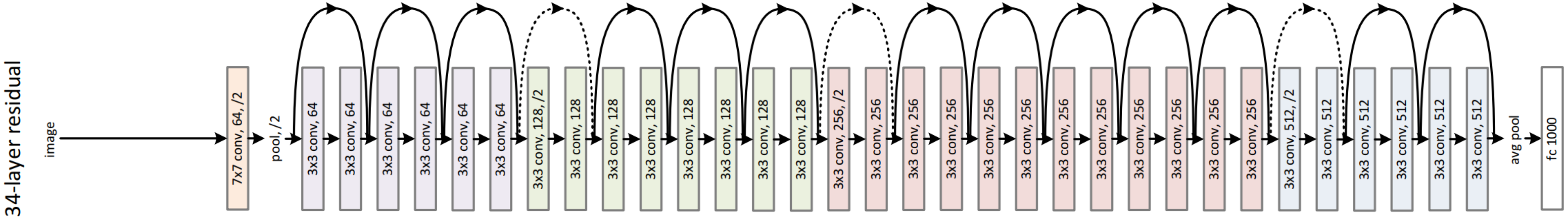
AlexNet
(2012)



VGG-19
(2015)

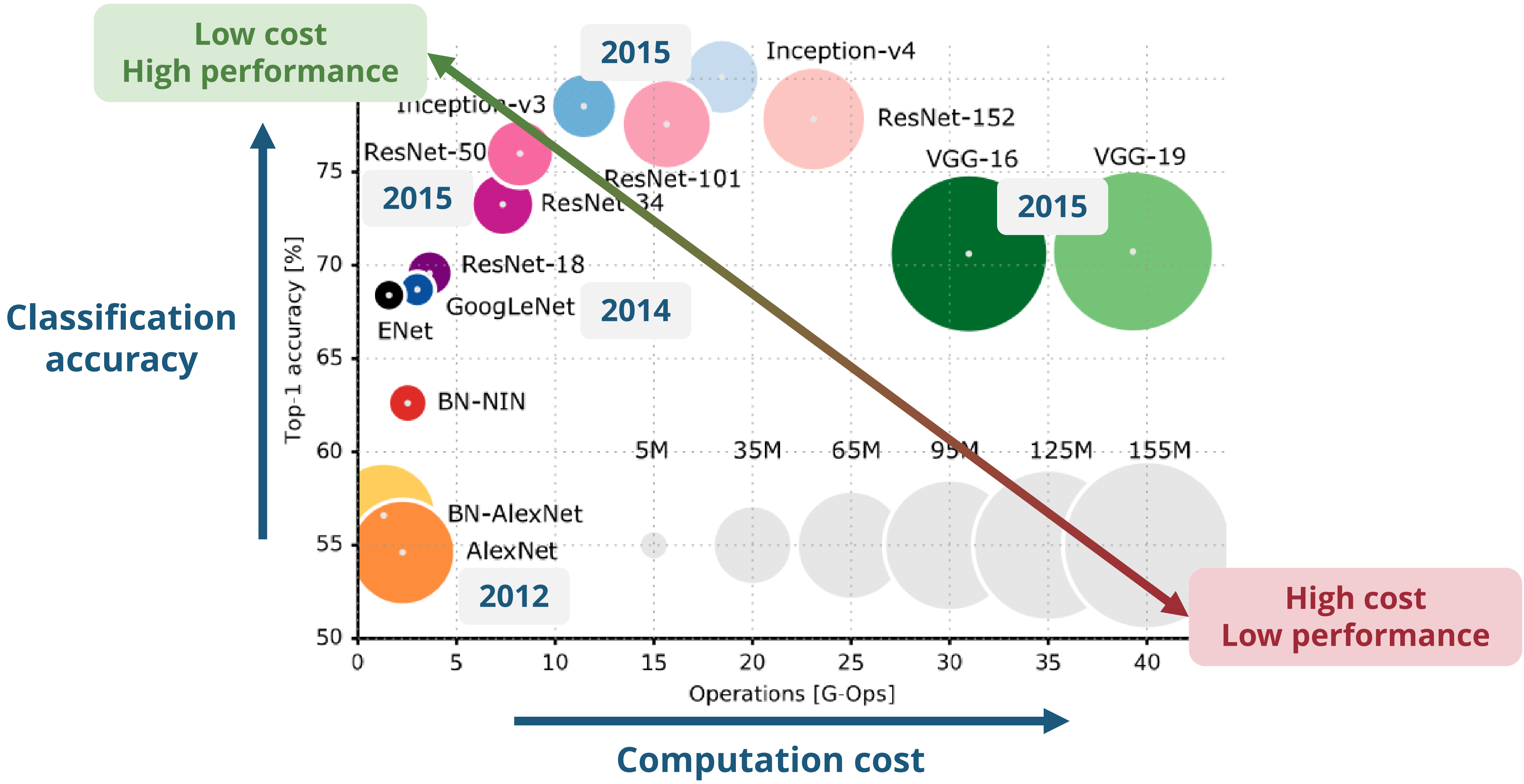


ResNet
(2015)



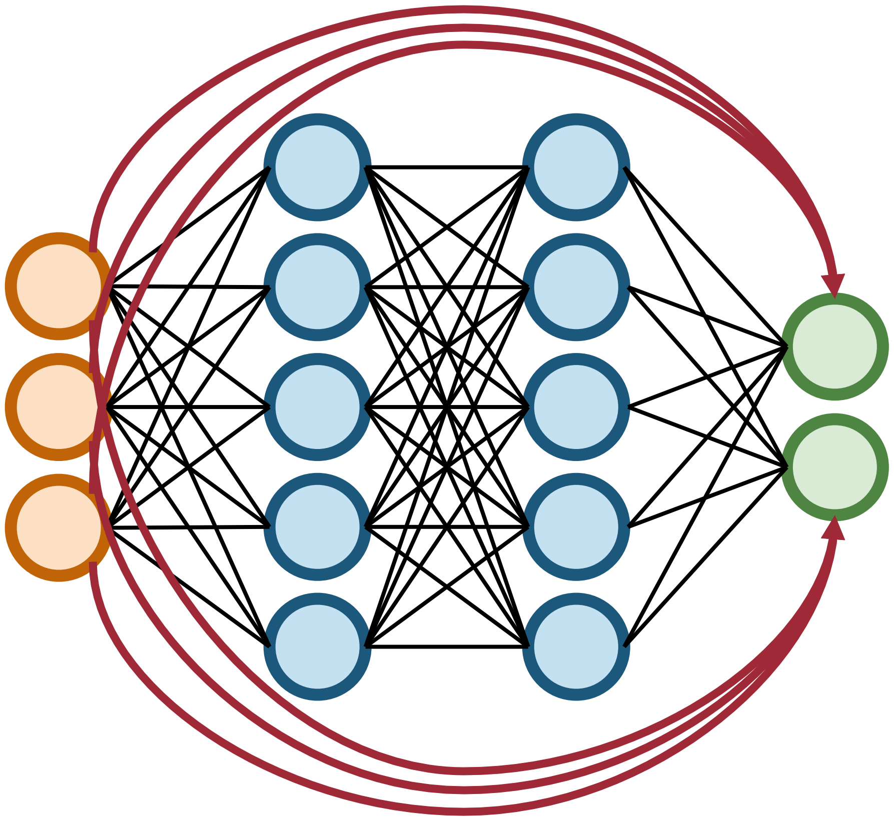
Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, 1998
Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” *NeurIPS*, 2012
Karen Simonyan and Andrew Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *ICLR*, 2015
Kaiming He Xiangyu Zhang Shaoqing Ren Jian Sun

Computation Cost vs Classification Accuracy



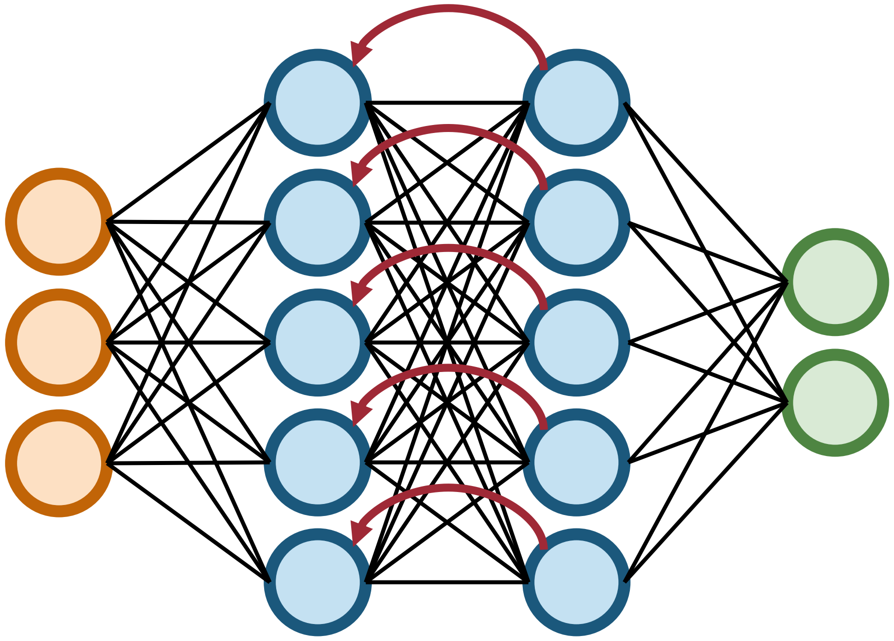
Neural Networks are NOT always Layer-by-Layer

Skip connections



Used in ResNets, U-Nets, diffusion models

Feedback loops



Used in RNNs, LSTMs, GRUs