Neural Network Fundamentals

From Biological Inspiration to Computational Models

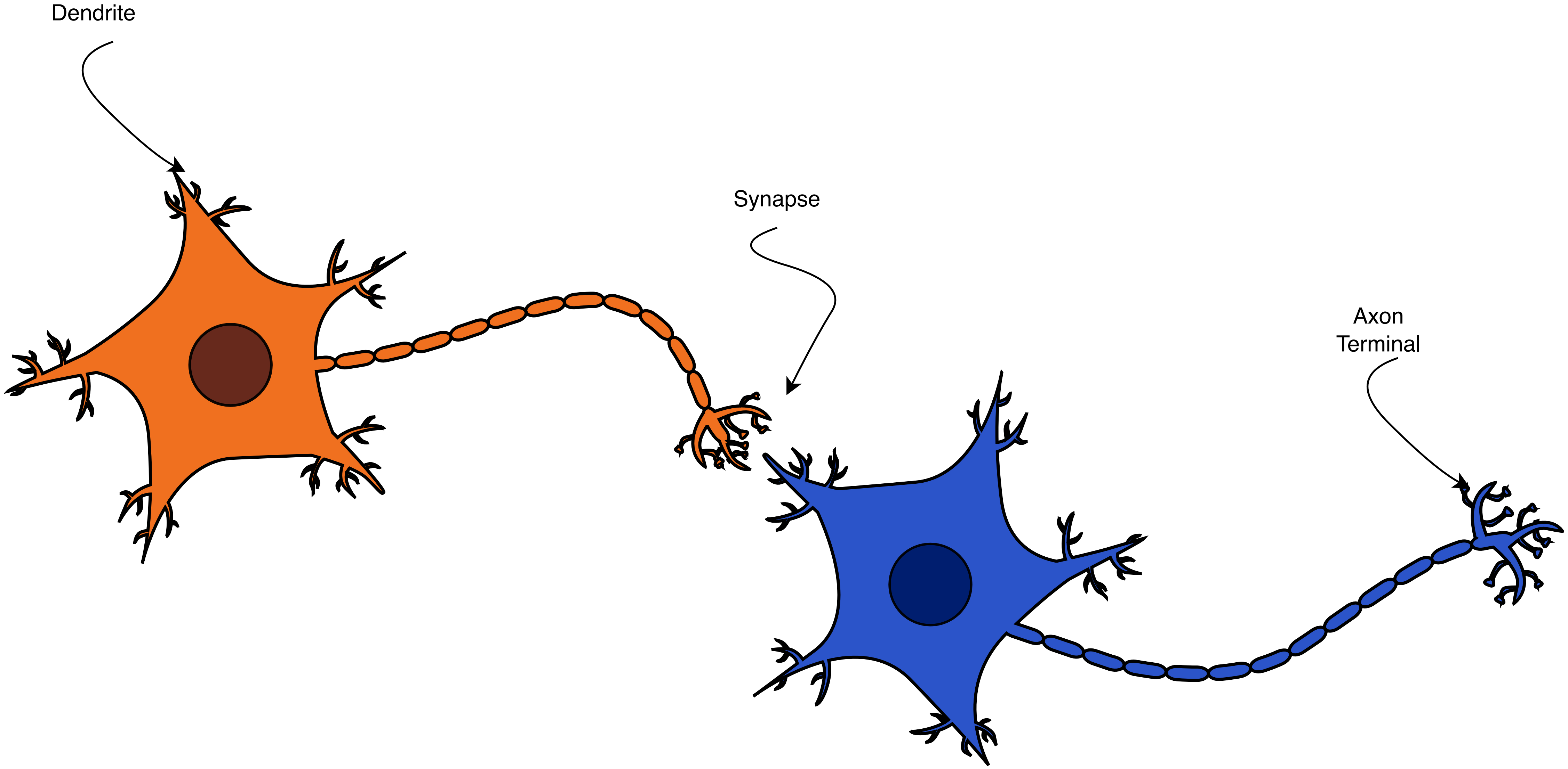
Sina Moghimi, Autumn 2024

МФТИ

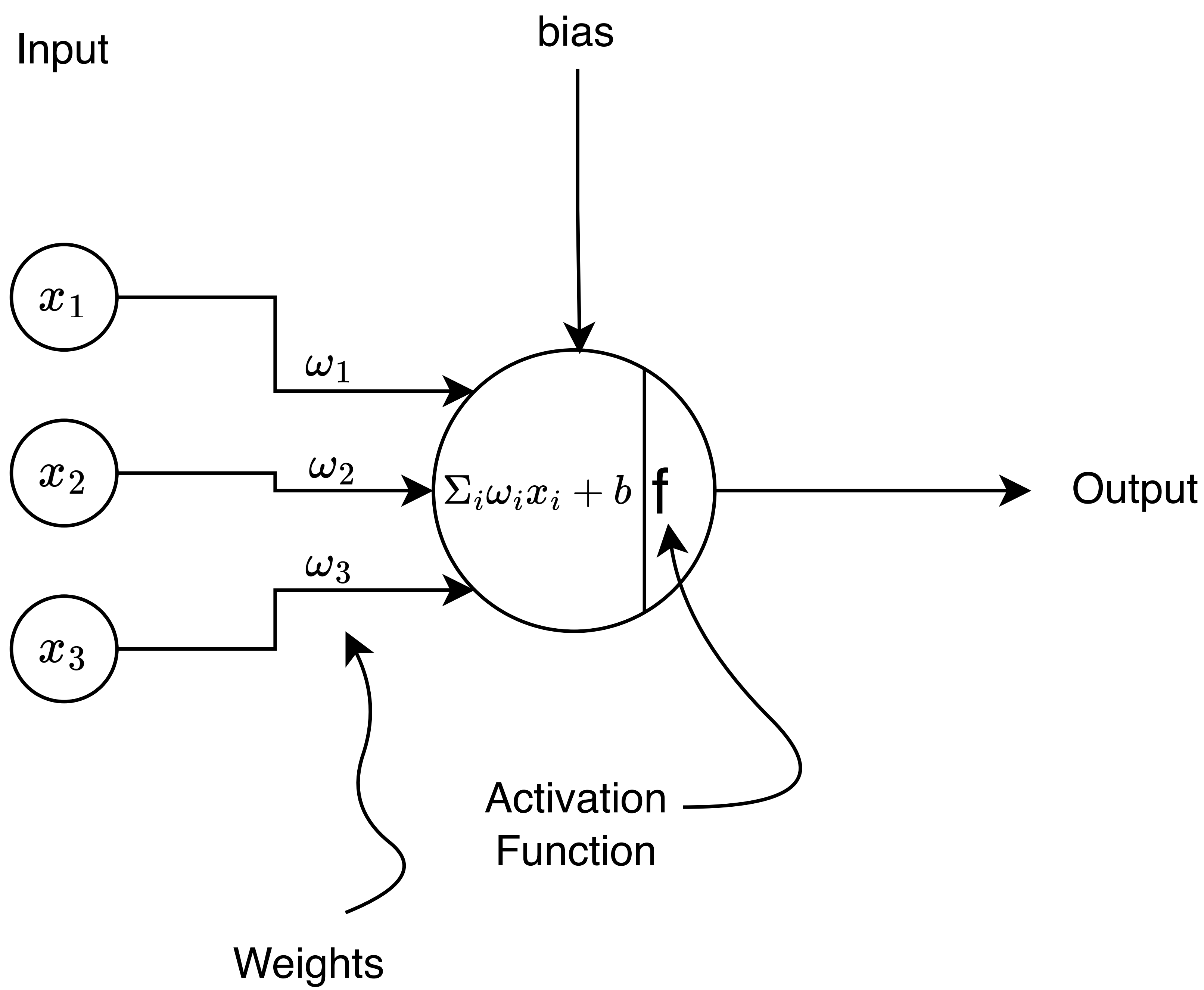
Outline

- Biological Neuron Structure
- Basic Artificial Neuron Model
- Neural Network Architecture
- Nodes and Connections
- Activation Functions
- Network Complexity
- Learning in Neural Networks

Biological Neuron Structure



Artificial Neuron Structure



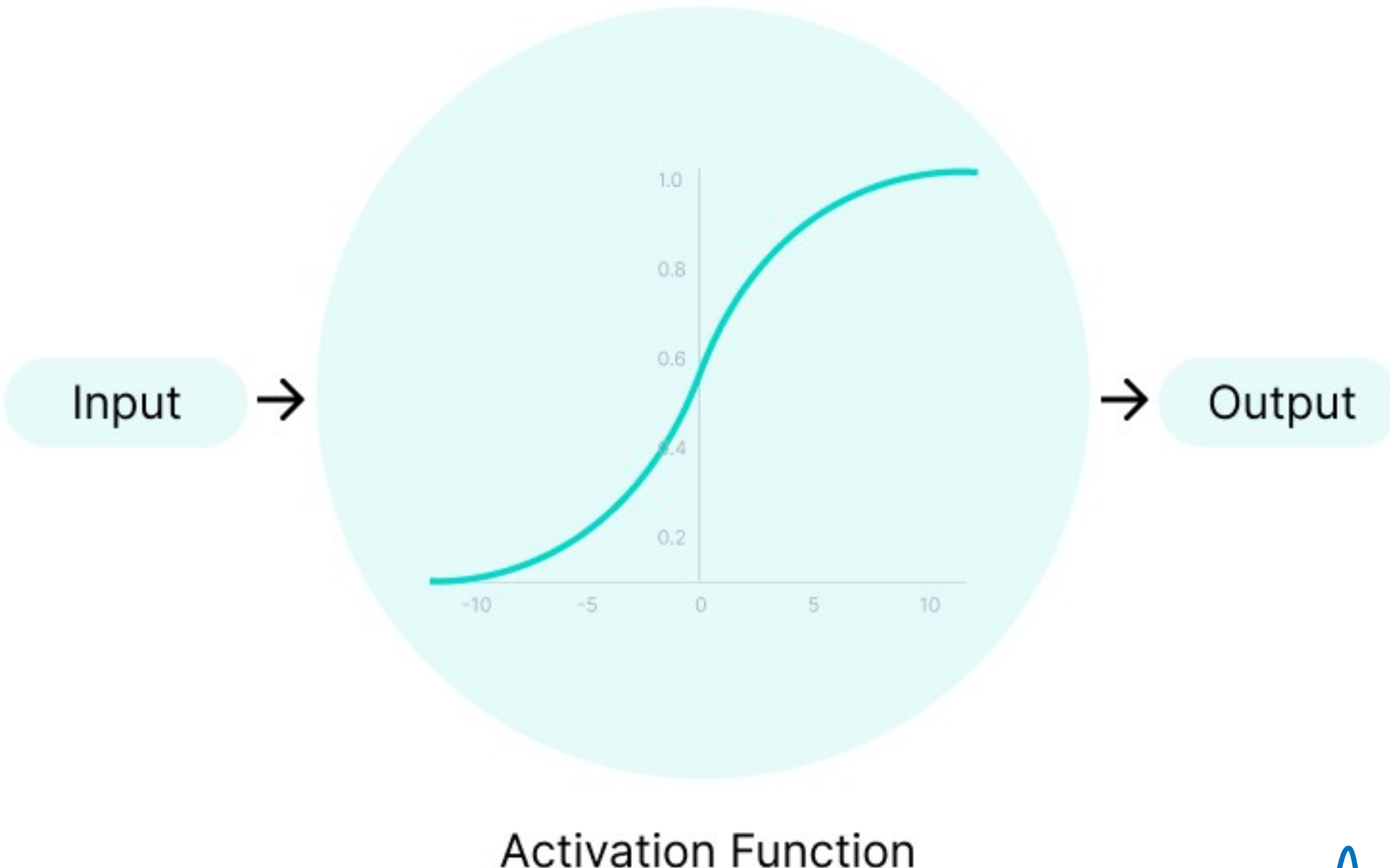
Artificial Neuron

- Single input - Single Output
- Multi input - Single Output
- Single input - Multi Output
- Multi input - Multi Output

- Aggregator ($\Sigma_i \omega_i x_i + b$)
- Activation Function (f)

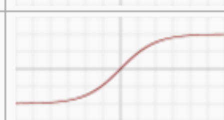
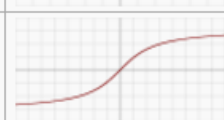
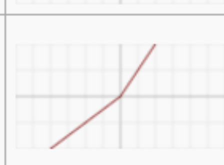
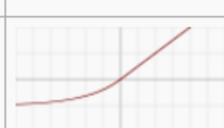
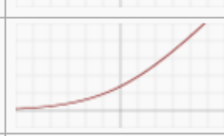
Activation Function

- Transforms the input data, why?



https://www.v7labs.com/blog/neural-networks-activation-functions

Activation Functions

Name	Plot	Equation	Derivative
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1 + e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
TanH		$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2 + 1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parameteric Rectified Linear Unit (PReLU) [2]		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Exponential Linear Unit (ELU) [3]		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1 + e^x)$	$f'(x) = \frac{1}{1 + e^{-x}}$

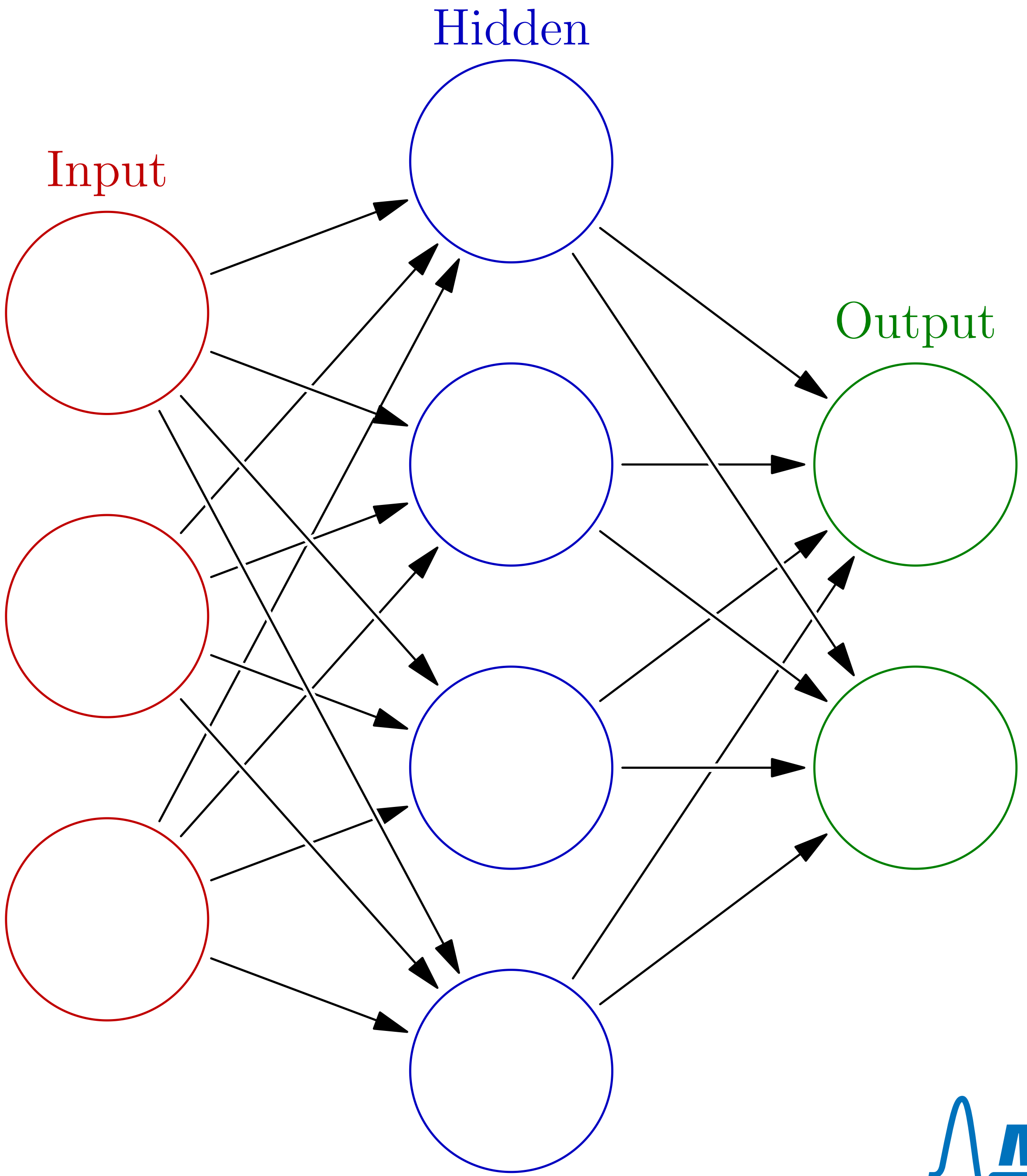
https://medium.com/the-modern-scientist/an-overview-of-activation-functions-in-deep-learning-97a85ac00460

Neural Network Architecture

Let’s build a network of neurons …

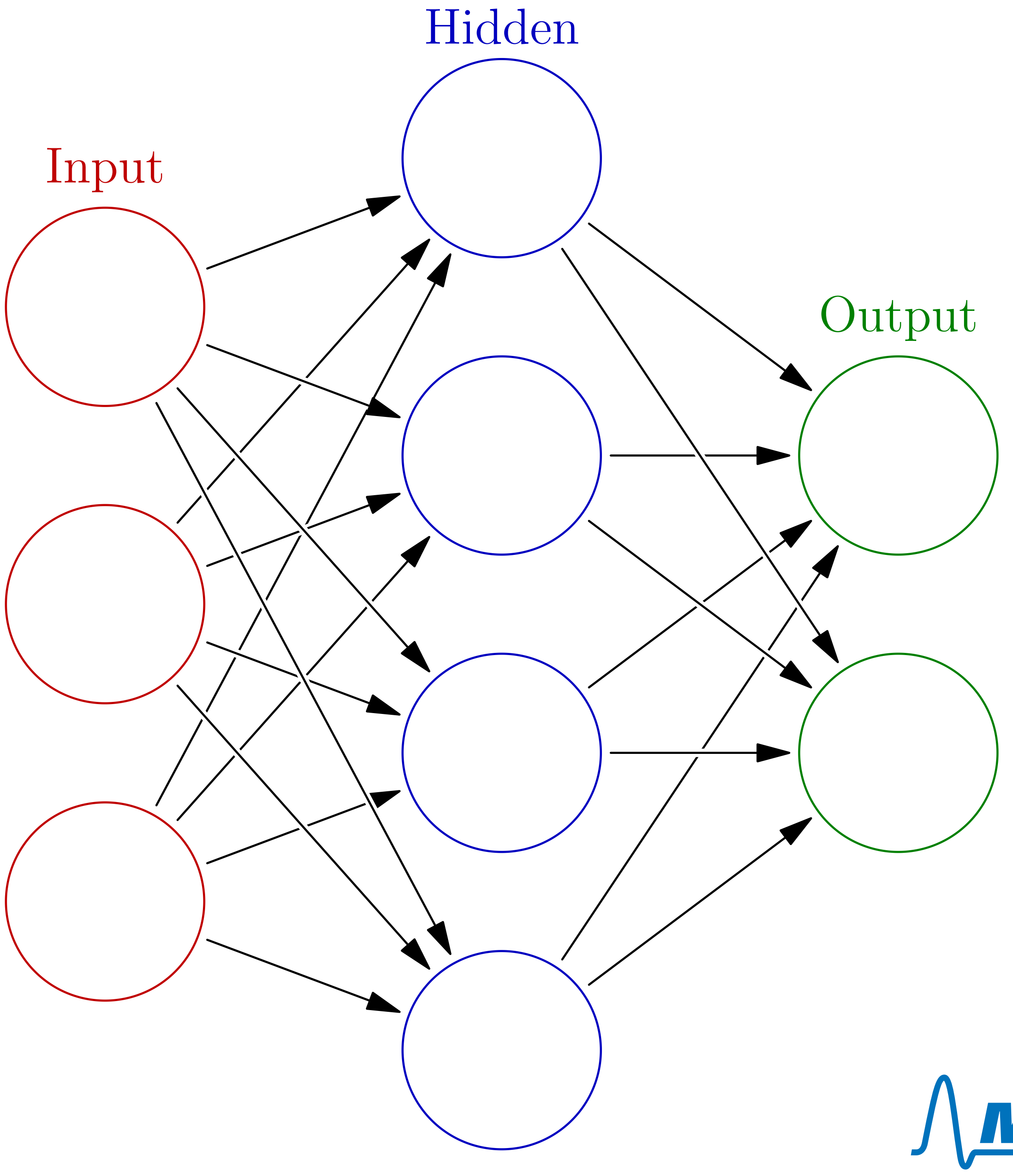
- Network Layers:
  - Input layer
  - Hidden layer
  - Output layer

- Connection Mechanism
  - Fully Connected
  - Sparse Connections



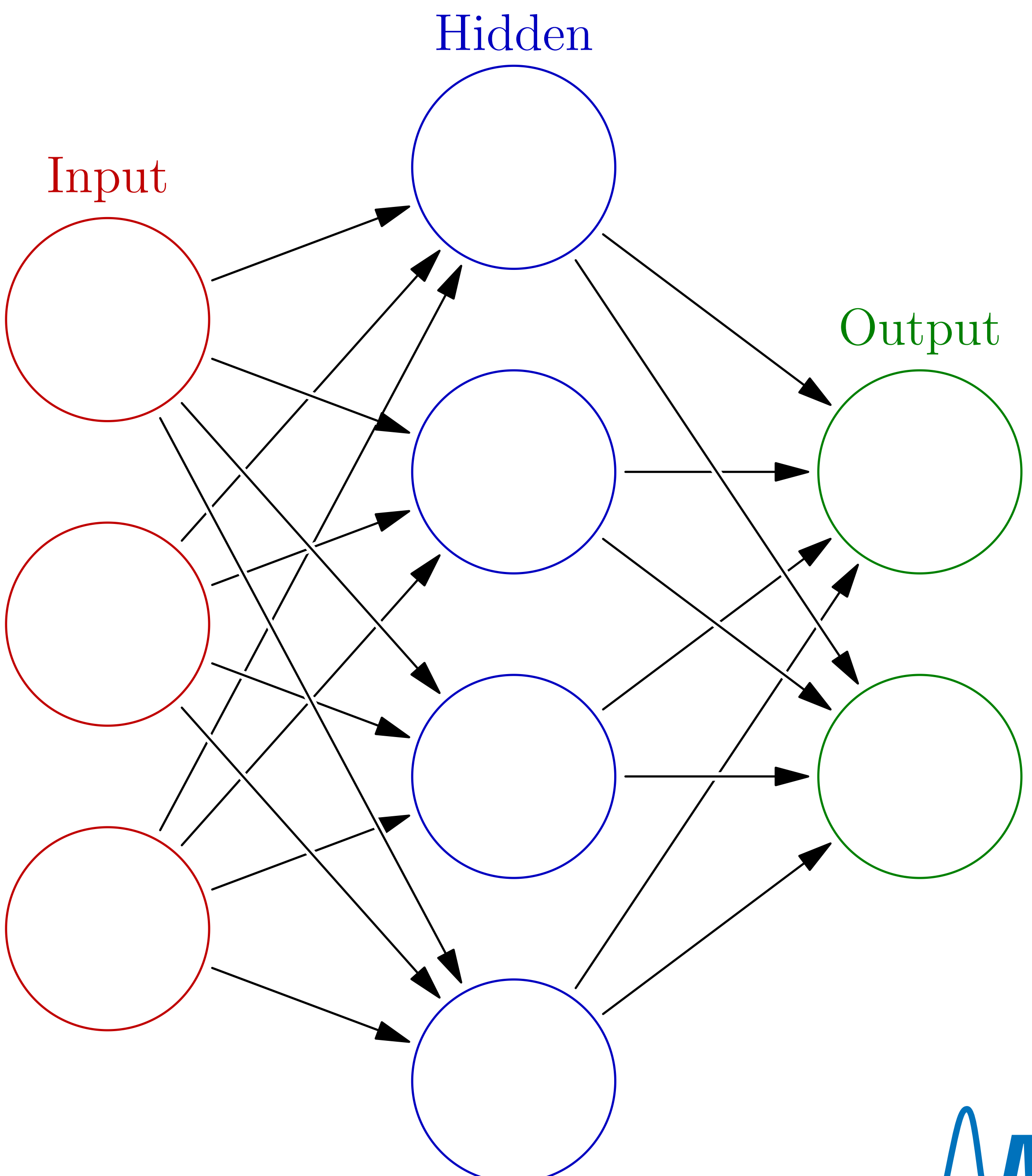
Network Complexity

- Depth vs. Width
- Complexity trade-offs

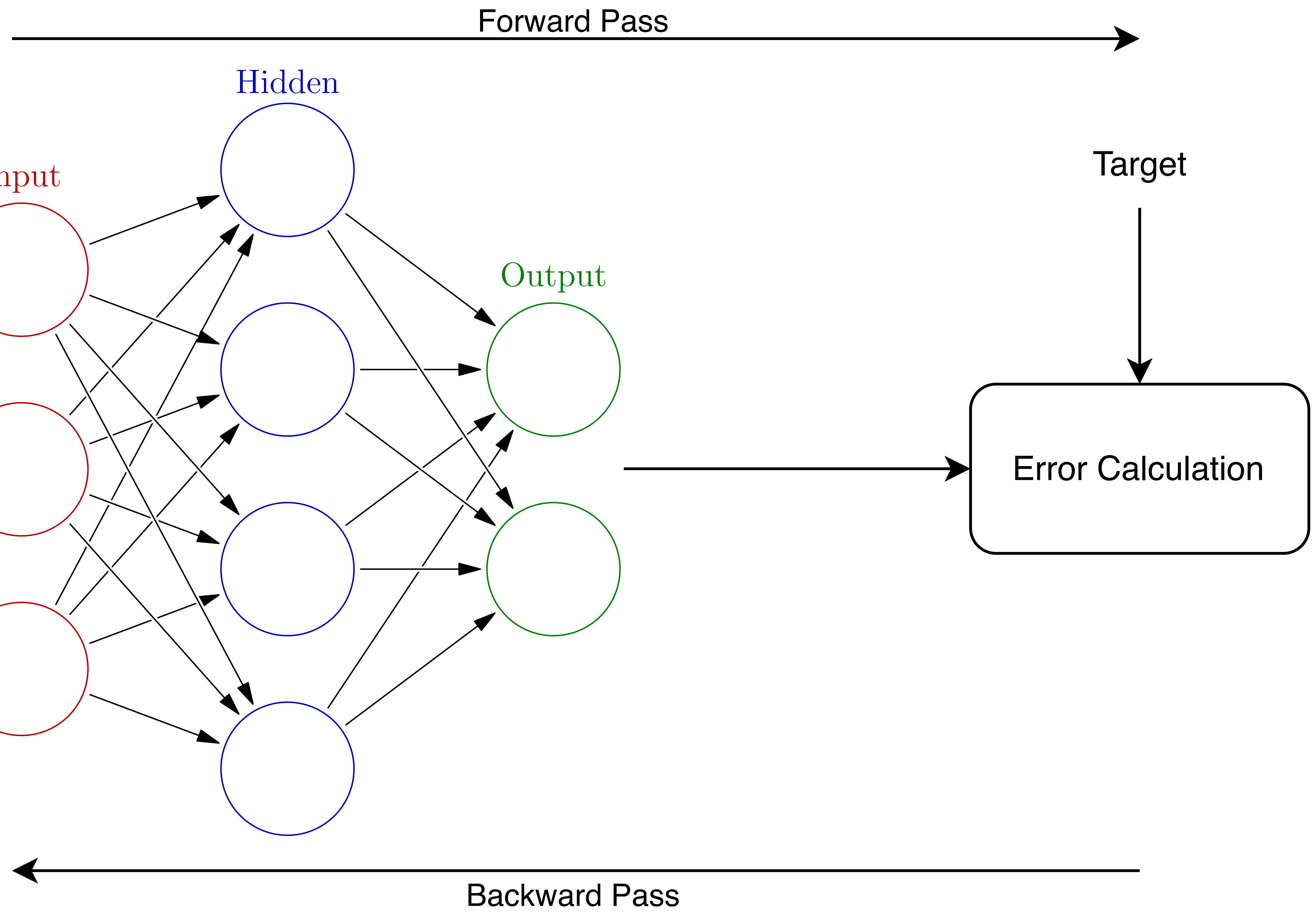


Learning in Neural Networks

1. Initialize the Weights randomly ?!
2. Forward Pass
3. Error Calculation
4. Backward Pass
    1. Error Back Propagation
    2. Gradient Descent
    3. Weight Adjustment



Pipeline



Forward Pass
Hidden
Input
Output
Target
Error Calculation
Backward Pass