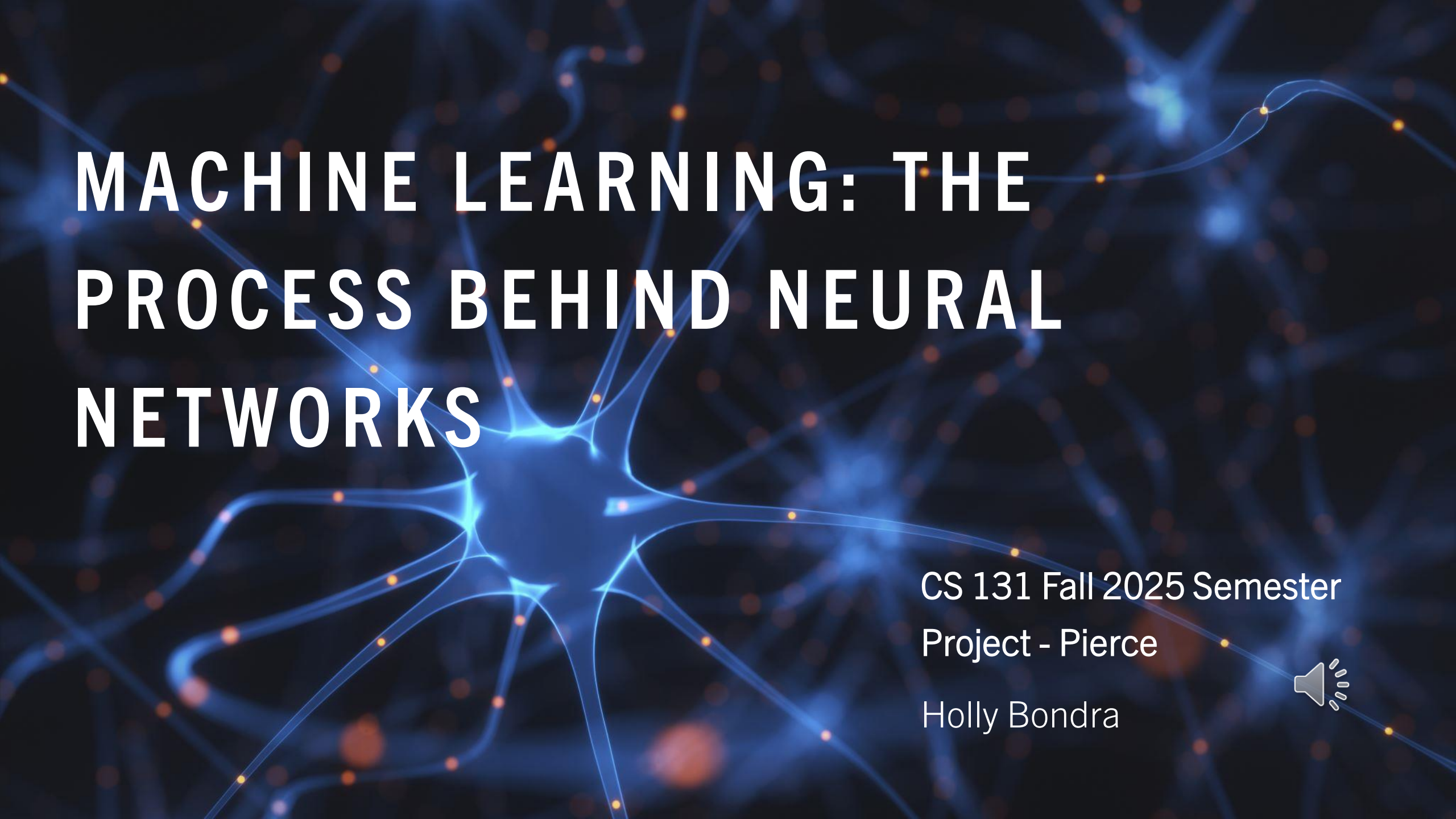


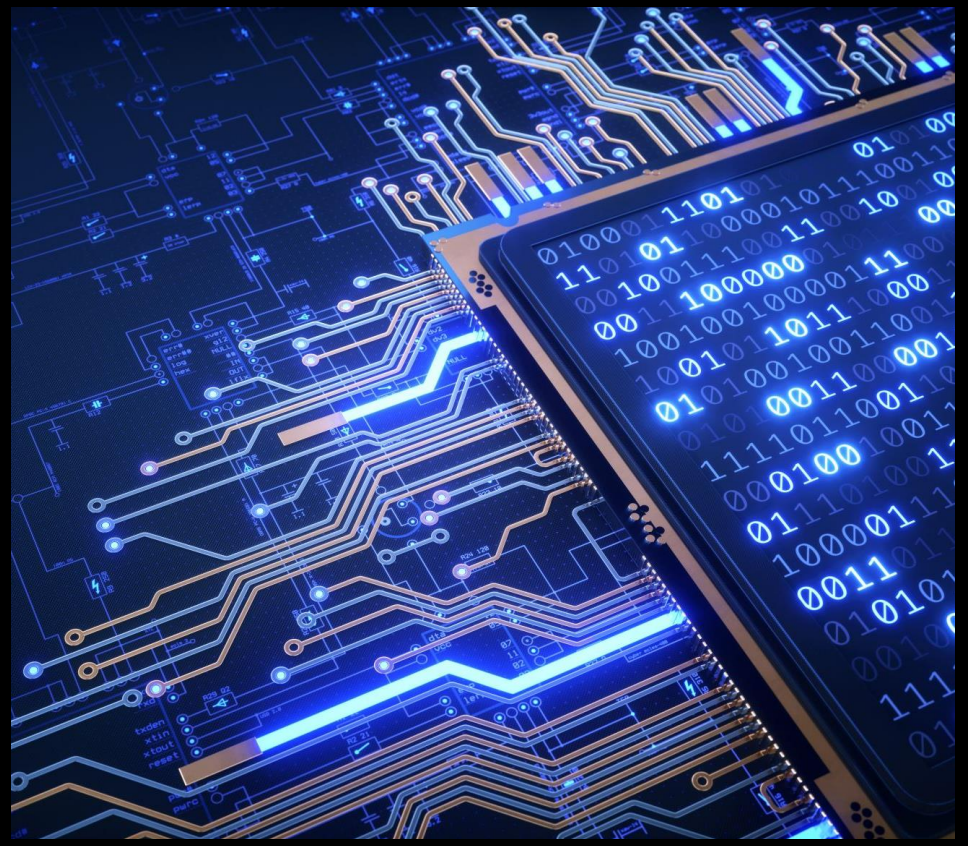
MACHINE LEARNING: THE PROCESS BEHIND NEURAL NETWORKS

CS 131 Fall 2025 Semester Project - Pierce

Holly Bondra

WHAT IS MACHINE LEARNING?

- A branch of Artificial Intelligence
- Learns from data rather than fixed rules
- Finds patterns that may be too complex for humans to code manually
- Over time, it improves



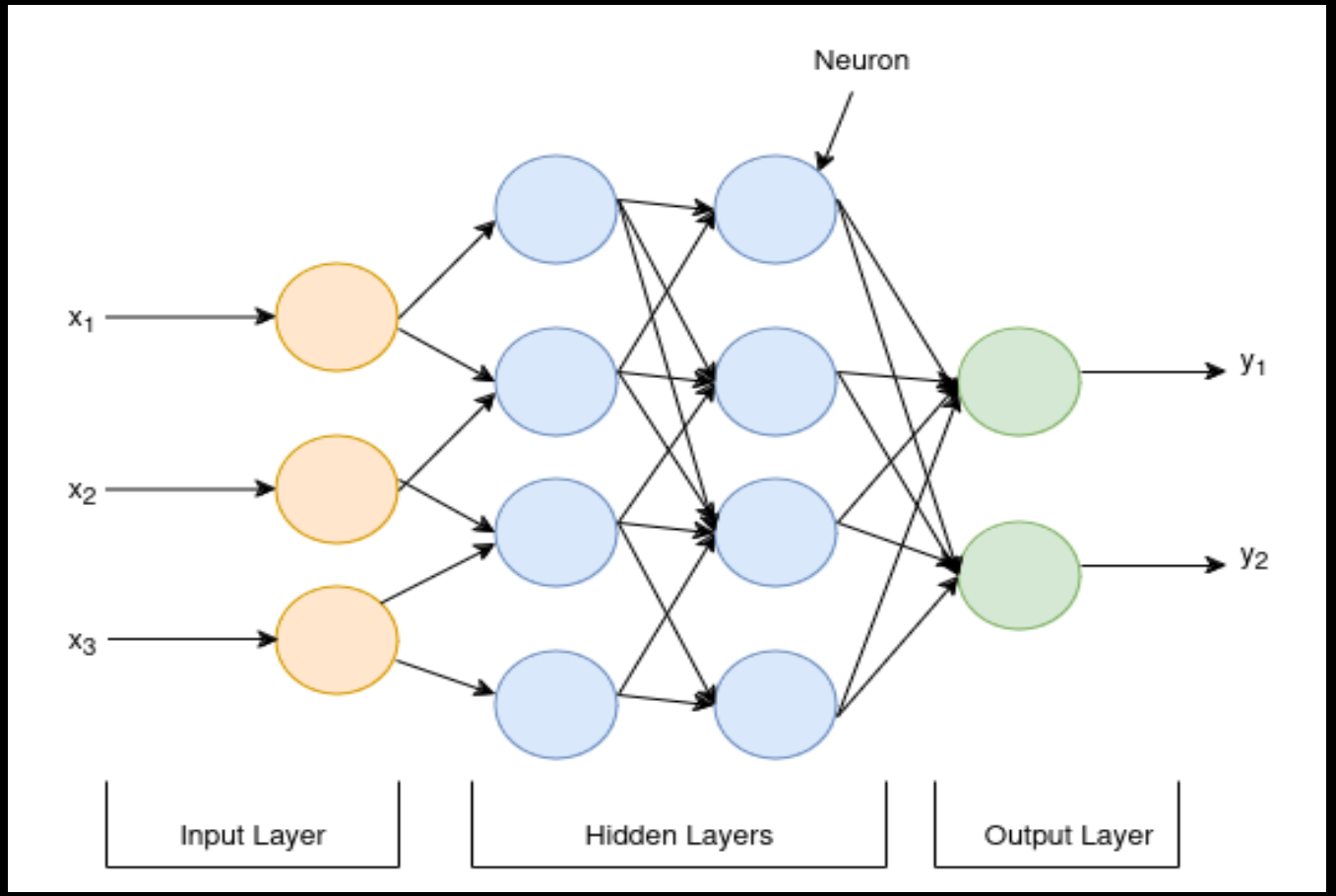
WHAT IS A NEURAL NETWORK?

- Inspired by human brains
- Contain "neurons" which are mathematical functions
- Neurons are arranged in layers
- Learns by adjusting its weights based on error
- As more layers are created, complex patterns are learned

STRUCTURE OF A NEURAL NETWORK

- Three layers
  - Input
  - Hidden
  - Output

Weighted sum = (input × weight) + bias



NEURONS

BIOLOGICAL NEURON

- Composed of dendrites, a soma, and axon
- Use electrical and chemical impulses to receive, process, and transmit signals
- Handles uncertain information efficiently

ARTIFICIAL NEURON

- Uses inputs, weights, biases, and activation functions
- Emulates the behavior of a biological neuron
- Perform best on specific, well-defined tasks

NEURONS

- The weighted sum:

$$z = (w_1 x_1 + w_2 x_2 + \cdots + w_n\, x_n) + \mathrm{b}$$

- The activation function:

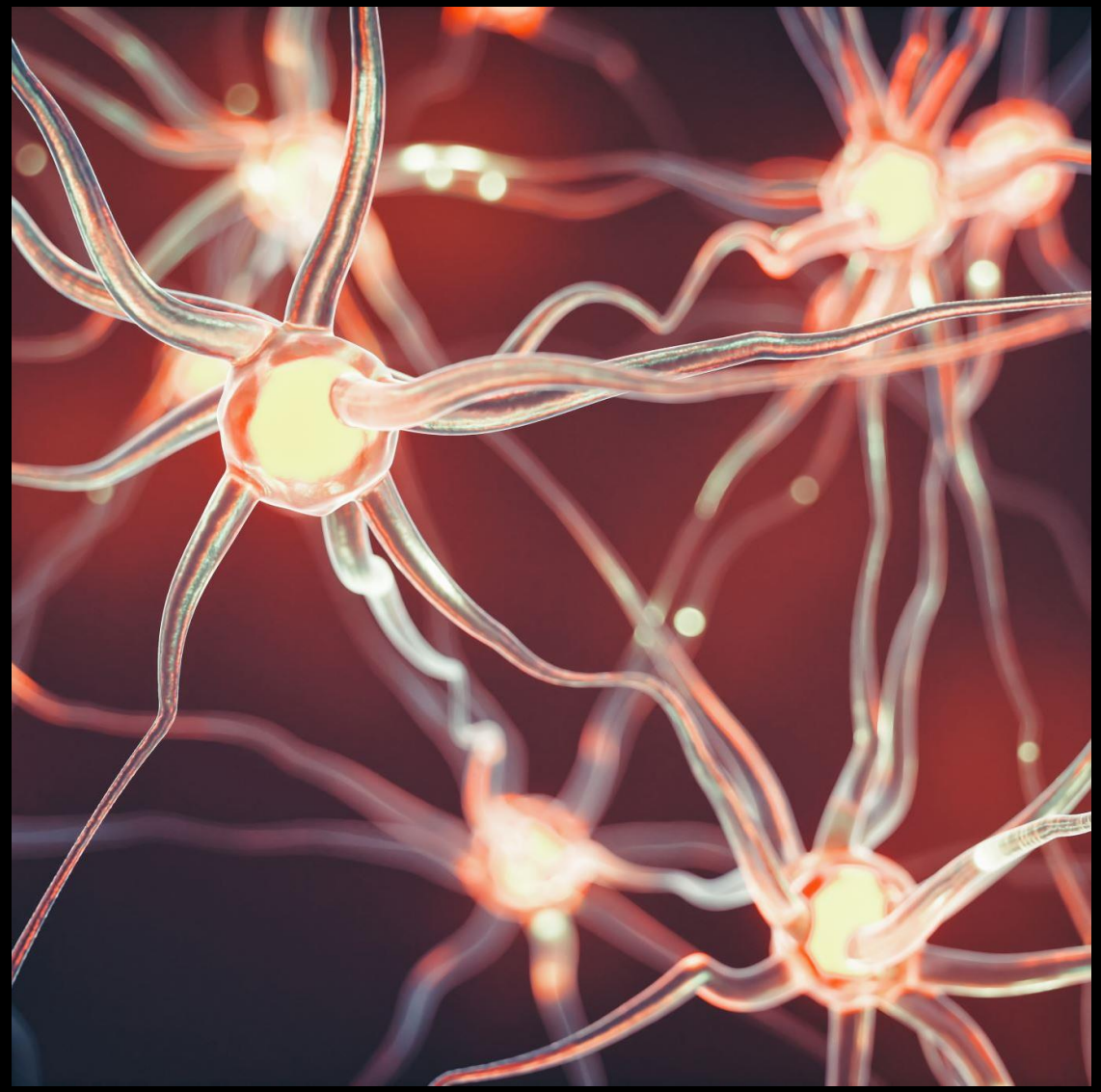
$$\boldsymbol{Output} = \boldsymbol{f}(\boldsymbol{z}) = \boldsymbol{f}(w_1 x_1 + \ w_2 x_2 + \cdots + w_n\, x_n + \mathrm{b})$$

ACTIVATION FUNCTIONS

What are activation functions?

What is their purpose?

What are some examples of activation functions?



ACTIVATION FUNCTIONS

RECTIFIED LINEAR UNIT (RELU)

- Outputs 0 for negative values, and keeps positive values
- Common choice for hidden layers
- Better at deep learning patterns as they are faster to train

$$\text{ReLU}(z) = \max(0,z)$$

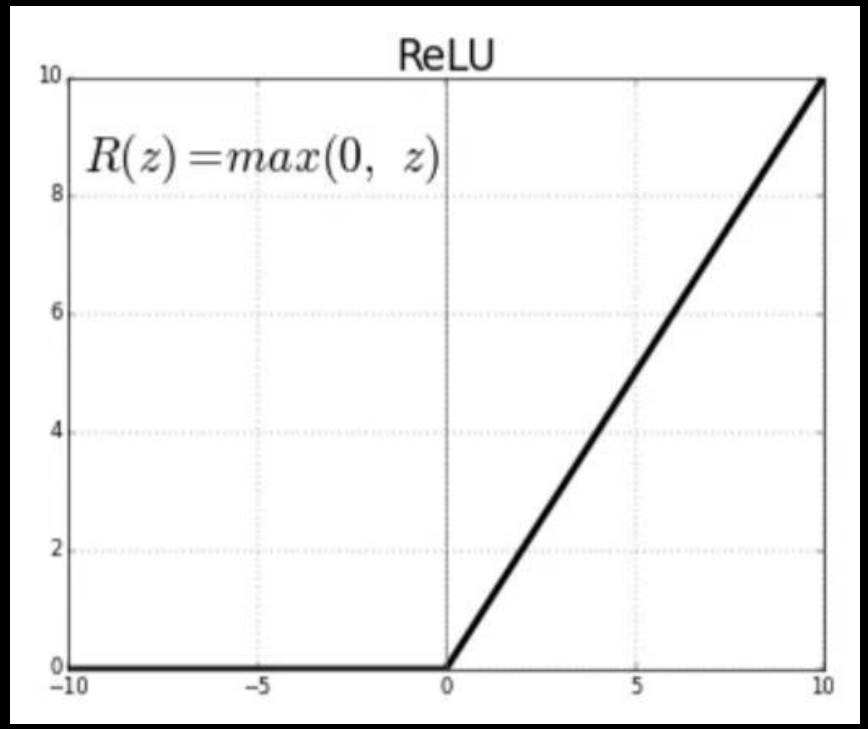
SIGMOID FUNCTION

- Maps any number within the range of 0 and 1
- Used in output layer for yes/no decisions
- Not commonly used in hidden layers, as they are slower

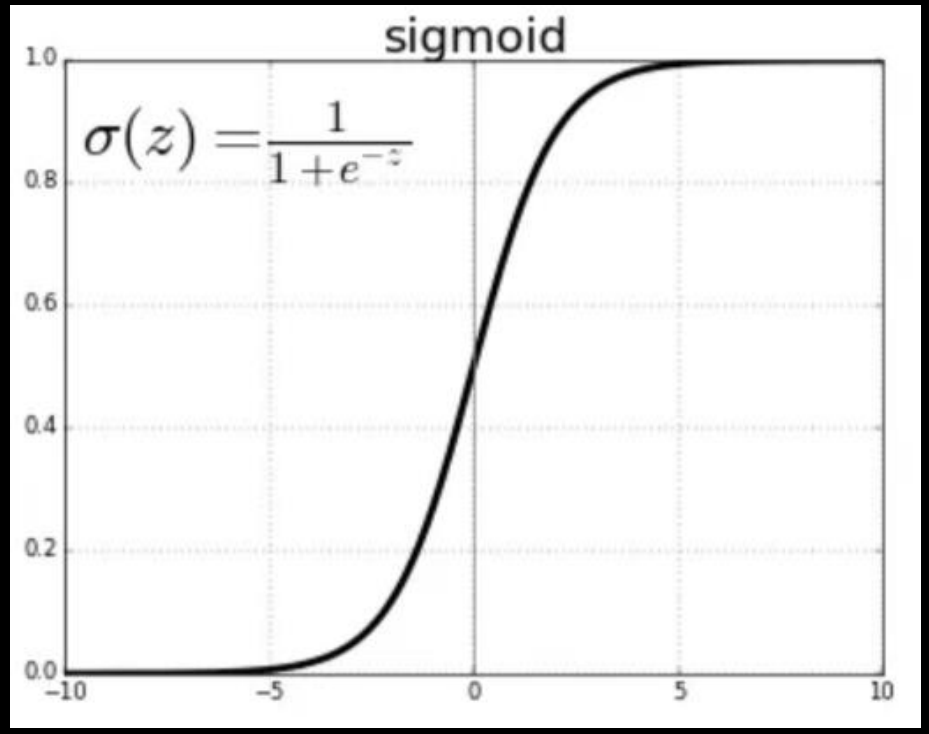
$$\sigma(z) = \frac{1}{1+e^{-z}}$$

ACTIVATION FUNCTIONS

RECTIFIED LINEAR UNIT (RELU)



SIGMOID FUNCTION



BACKPROPAGATION

- The algorithm used to train neural networks
- Measures the inaccuracy of the network’s prediction
- Determines which weight caused errors
- Adjusts the weight to reduce errors

BACKPROPAGATION

Step 1: Make a prediction $Output = f(z)$

Step 2: Measure the error $L = \frac{1}{2}(y_{true} - y_{pred})^2$

Step 3: Find what caused the error $\frac{\partial L}{\partial w_i}$

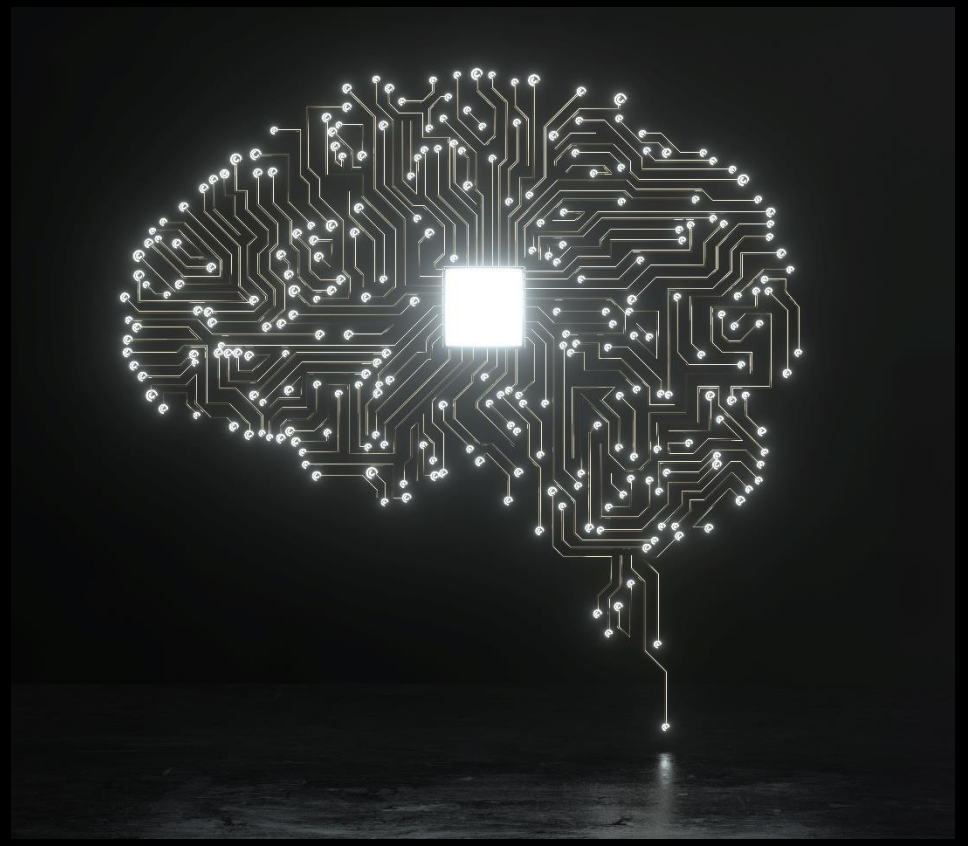
Step 4: Update the weights $w_i - \eta \frac{\partial L}{\partial w_i}$

WHY ARE NEURAL NETWORKS IMPORTANT?

- Allow machines to learn from data
- Solve problems too complex for traditional programming
- Lead breakthroughs for pattern recognition
- Enable automation
- Power modern artificial intelligence

THE FUTURE OF NEURAL NETWORKS

- More efficient models that learn faster and use less computing power
- Deeper integration into daily life
- Smarter automation in medicine, transportation, and robotics



CONCLUSION

- Understanding machine learning and neural networks
- Math and algorithms behind neural networks
- Why neural networks are important
- The future of neural networks