# Neural Network Breakdown

can code communities

# Review Questions

- What makes RAG so valuable?
- What are the three core steps that make up the RAG pipeline?
- What is one way we can improve the user prompt for better retrieval?

# Review Take Home

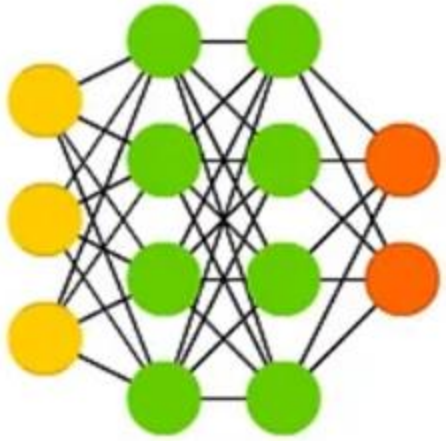
Create a RAG chain using LangChain and connect at least three documents within the same topic. Test the chain with some prompts asking for information you know can be retrieved in the documents to test.

# Understanding Neural Networks

- Neural networks are loosely inspired by the biological structure of the human brain.
  - Our brains consist of billions of interconnected neurons that transmit information through electrical signals.
  - These connections allow us to learn, recognize patterns, and make decisions.
- It is made up of neurons, also called nodes, arranged in layers.
  - These layers are connected by links that transmit signals (data) between them.

# Neurons

Neurons are the processing units of the network. Each receives input from other neurons, performs a simple calculation, and transmits a single output value (typically 0-1).



# Layers

Neurons are organized into layers. A typical network has three layers.

1. The input layer receives raw data from the external world.
2. The hidden layers perform the core computations and information processing.
    a. There can be one or more hidden layers.
    b. The computations done here differ between different styles of neural networks.
3. The output layer displays the probability of each possible outcome.
    a. The highest probability will be considered the predicted output from the neural network.

# Weights

Each layer’s neurons are connected to the layer’s neurons before and after it. These connections are called weights and they quantify the strength of connection between neurons in different layers.

# Biases

A bias term is a constant value added to the input of each neuron. This creates a better distinction between “inactive” neurons and neurons with low output values.

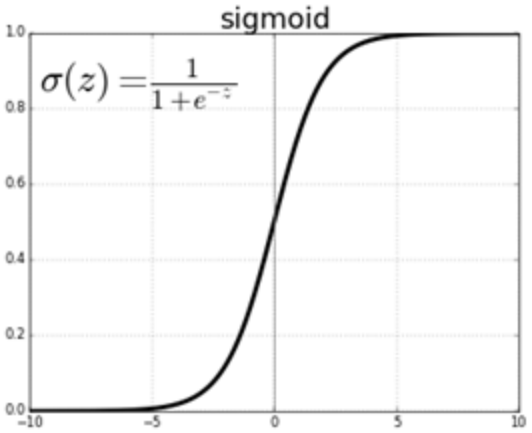
# Activation Function

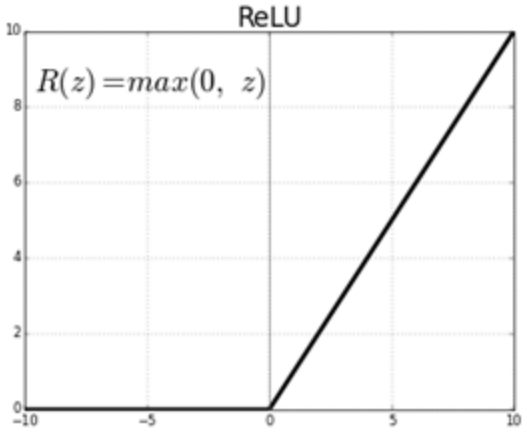
The activation function is the calculation made within the neuron. This function determines whether a neuron "fires" (sends a signal) based on the weighted sum of its inputs. Common activation functions include sigmoid and ReLU.

# Sigmoid vs. ReLU

Since neuron outputs typically range from 0 to 1 and weights aren't inherently bounded, the weighted sum of all the neurons of the previous layer may result in a value that is outside of the 0 to 1 range for this neuron's output value. We handle this by using a function which compresses values outside the range.

- Sigmoid
- ReLU



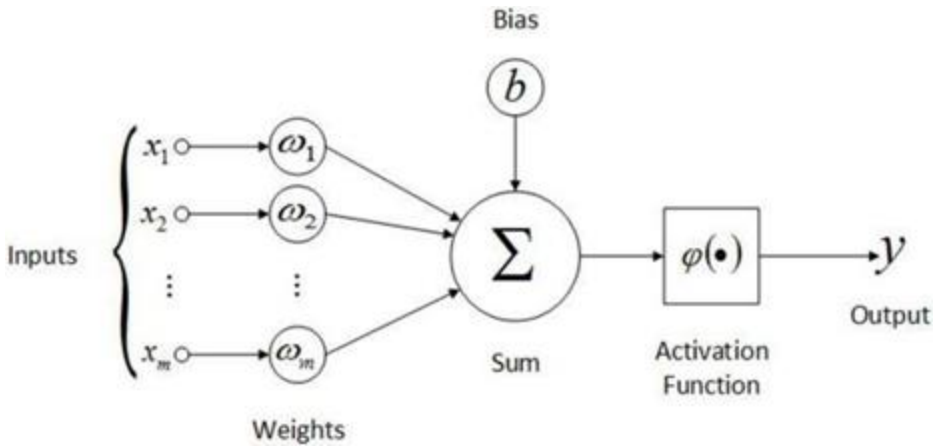



# Sigmoid vs. ReLU

- The sigmoid function has been used as the function to compress the range of the activation values for a while, but in the past few years, another function called ReLU has been found to be more efficient when training neural networks.
- The main difference between these functions is that a value below 0 for the ReLU will always be 0, and for the sigmoid function, the values past 0, sort of plateau right above zero.
- This simplification in the function helps there be less ambiguity between inactive values and lower activity values, which in turn makes the training process more efficient.
- Even though there are no theoretical positive bounds with the ReLU function, outputs are limited by other factors in practice.

# Parameter Count

These calculations are done for each neuron in every layer past the first layer, which is the reason for the popular language models to have over billions of these parameters (weights and biases).



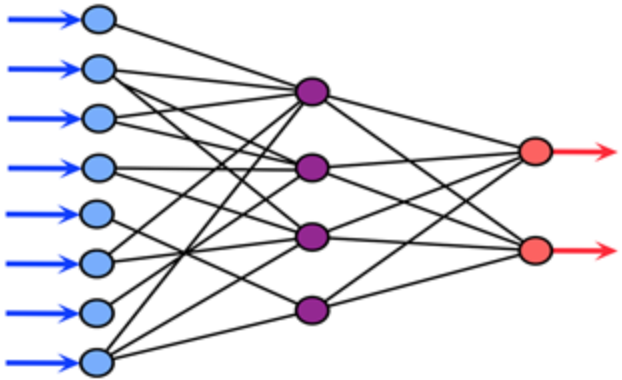
# Handling Neural Networks

For an example, let's say we are using a neural network for sentiment analysis.

- We're starting with text as the input, and looking to get either a positive or negative output.
- Tokenization is how we convert data into numerical representations so that the model can run its computations on the data.
- The input layer is a defined size that should be the maximum amount of tokens that we plan to process at a time.

# Handling Neural Networks

- The hidden layers perform all the computations.
- The output layer only consists of three nodes relating to the two possible outcomes (positive or negative).
    - Each of these nodes will contain a value that is regarded as the probability for the associated outcome.
    - The highest value of the three will be chosen as the predicted outcome.



# Take Home Exercise

Given the example we went over in class (where the neural network predicted the fourth value in a binary sequence), what could be three examples of what that binary data could represent that would make this model helpful in a real world scenario?