Introduction to Neural Networks

Understanding the Basics

DATA ANALYTICS | IRONHACK

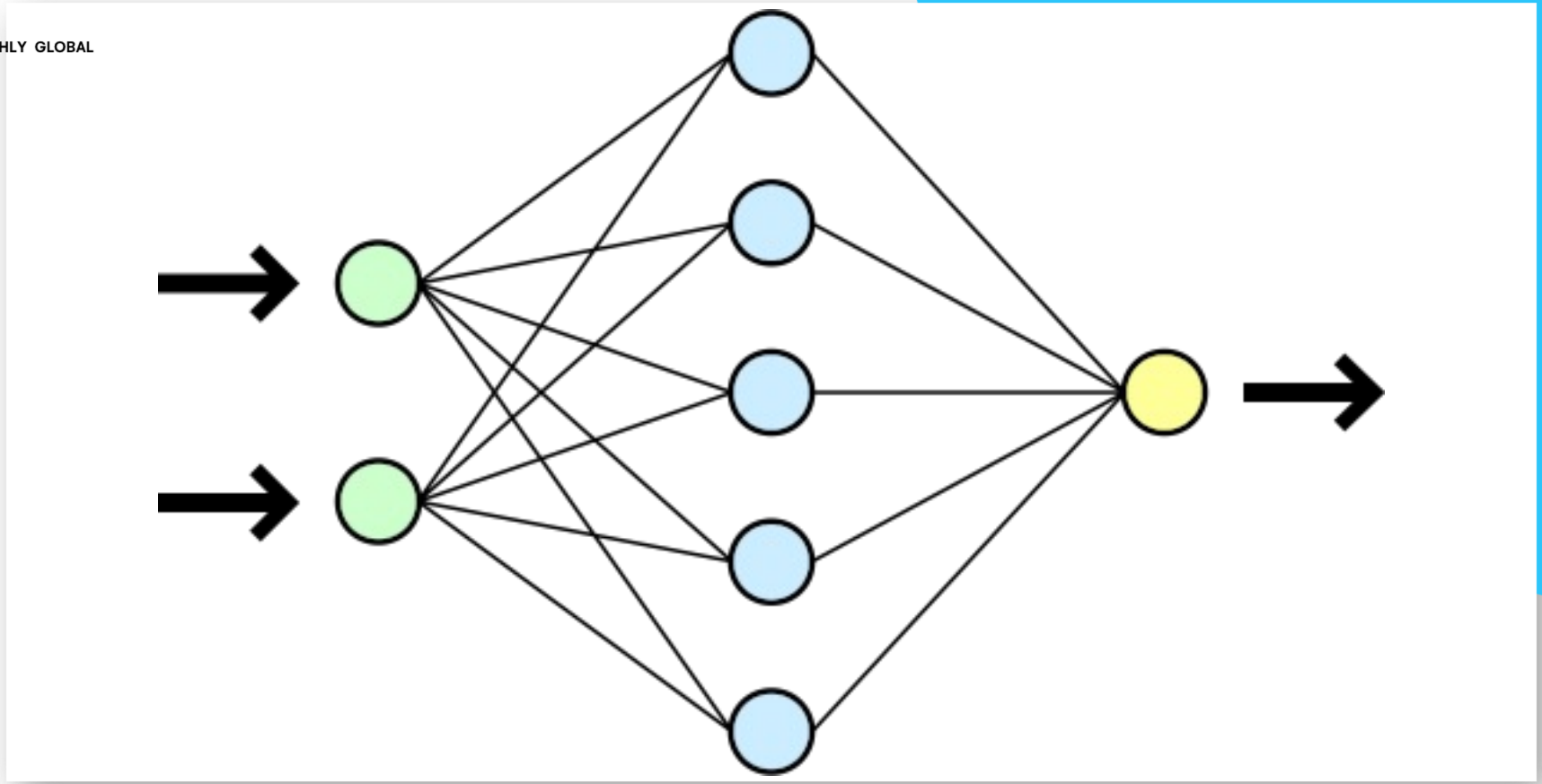
What is a Neural Network?

- A **neural network** is a series of algorithms that attempts to recognize underlying relationships in a set of data through a process that mimics the human brain.
- It is composed of layers of nodes (neurons), where each layer transforms the input data into more abstract representations.

Basic Structure

- **Input Layer**: Receives input data.
- **Hidden Layers**: Intermediate layers that process inputs.
- **Output Layer:** Produces the final output.

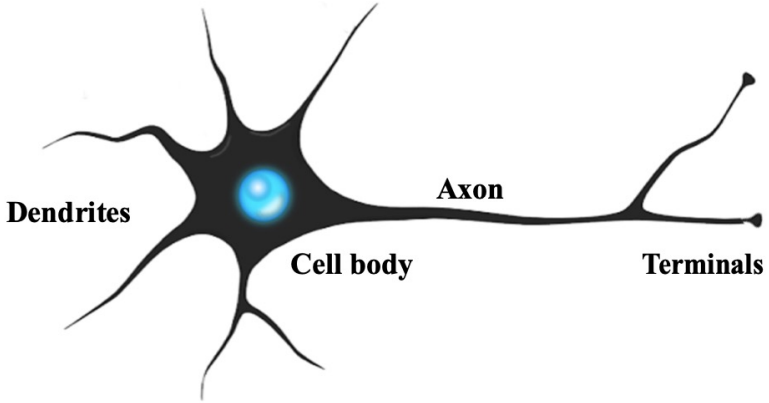
.

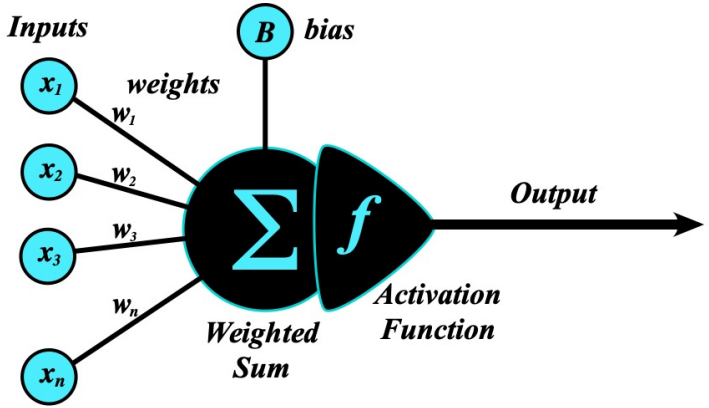


Neurons and Activation Functions

- **Neuron**: The basic unit, processes inputs by applying weights, adding bias, and passing through an activation function.

- **Activation Function:** Determines whether a neuron should be activated (e.g., Sigmoid, ReLU, Tanh).


Dendrites
Axon
Cell body
Terminals


Inputs
B
bias
x_1
weights
w_1
x_2
w_2
x_3
w_3
w_n
x_n
Σ
f
Output
Activation Function
Weighted Sum

Example

We want to build a model that predicts if an IronHack student graduates or not (binary variable, fail or pass) , based on the hours they've studied and their number of absences (2 inputs). A simple neural network structure might look like this:

Example

- **Input layer**: Has 2 neurons, one for each input (Hours studied and absences)
- **Hidden layer**: We will have 3 neurons in this layer. Each input neuron is connected to each hidden neuron, and weights are assigned to each connection.
- **Output Layer**: This layer will have 1 neuron, connected to to all 3 neurons in the hidden layer. It will have a threshold (using a sigmoid activation function, for example) to classify the result (The Ironhacker passed or failed).

Training a Neural Network

- **Forward Propagation:** Passing inputs through the network to get the output.
- **Loss Function:** Measures the difference between predicted and actual outputs.
- **Backpropagation:** Adjusts weights using the gradient of the loss function to minimize errors.
- **Optimization:** Gradient Descent or Adam, for example.

Common Types of Neural Networks

- **Feedforward Neural Networks (FNN):** The simplest form, where connections do not form cycles.
- **Convolutional Neural Networks (CNN):** Specializes in processing grid-like data such as images.
- **Recurrent Neural Networks (RNN):** Designed to handle sequential data, with feedback loops.

Real-World Applications

- **Image Recognition:** Face detection, object classification.
- **Natural Language Processing:** Sentiment analysis, translation.
- **Finance:** Stock price prediction, fraud detection.
- **Healthcare:** Disease diagnosis, personalized medicine.

Challenges and Limitations of Neural Networks

- **Overfitting**
- **Data Requirements:** Requires large amounts of data for training.
- **Computational Resources**: High demand for processing power and time.
- **Interpretability:** Neural networks are often seen as black boxes.

The Future of Neural Networks

- **Advancements:** Continued research in deep learning, reinforcement learning.
- **Ethical Considerations:** AI ethics, bias in algorithms.
- **Integration:** Increasing use in various industries, IoT, automation.

Summary

- Neural networks are a powerful tool for machine learning and AI.
- Applications are vast, but challenges remain in interpretability and data requirements.
- Computational resources are a big challenge

Q&A

THANKS !