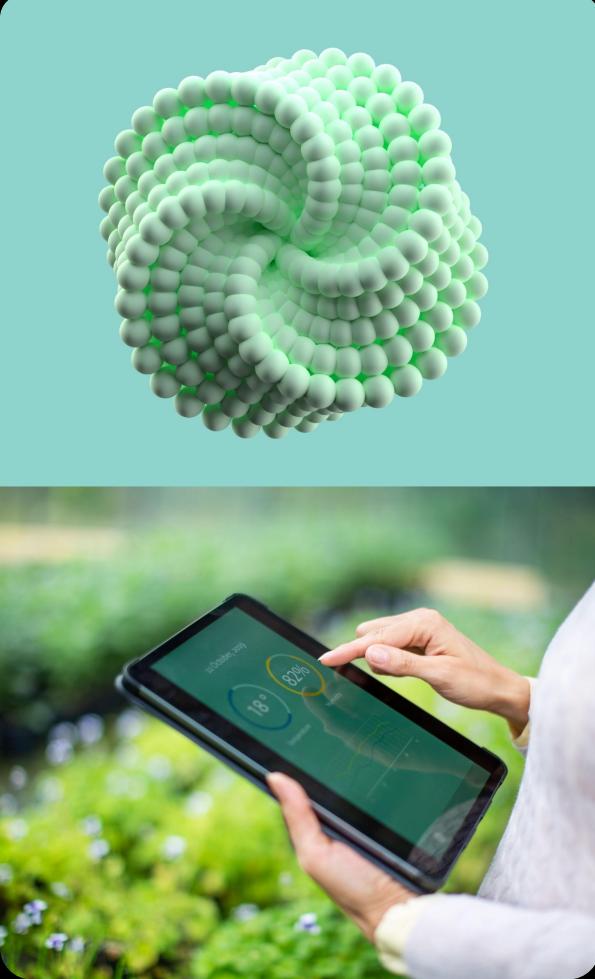




GROUP 2DEEP2LEARN – FEB 2025

Neural Networks in Nature



Contributors

Olugbenga Adegoroye
Jazmine Brown
Varit (Henry) Kobutra
Kimberly Navarrete
Phillip Torres

Prepared for

Prof. Patricia McManus
ITAI 2376



Agenda

01 Brief overview

02 The Neural Network Zoo

03 Individual Networks

04 RNN Deep Dive

05 Conclusion

06 Q&A

Brief overview of biological neurons



Brief Overview

Nature has been implementing neural networks long before computers existed.



Biological neural networks, found in brains and nervous systems, have inspired the artificial neural networks we use today. These networks are composed of interconnected neurons that process and transmit information through electrochemical signals, forming the basis for learning, memory, and adaptation in living organisms.

2Deep2Learn

Brief Overview



The Neural Network Zoo



Feedforward Neural Networks

Simple, organized in layers, works in colonies like neurons



RNN

Adaptable, processes information cyclically, has memory-like capabilities



GAN

Adapts and transforms, creates convincing imitations



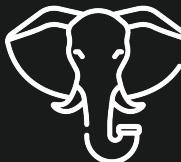
CNN

Exceptional at pattern recognition, processes visual information hierarchically



Transformers

Highly intelligent, processes complex sequences, excellent at communication



LSTM

Superior memory retention, processes long sequences of information



Self-Organizing Maps

Organizes data into patterns, creates structured representations



Hopfield Networks

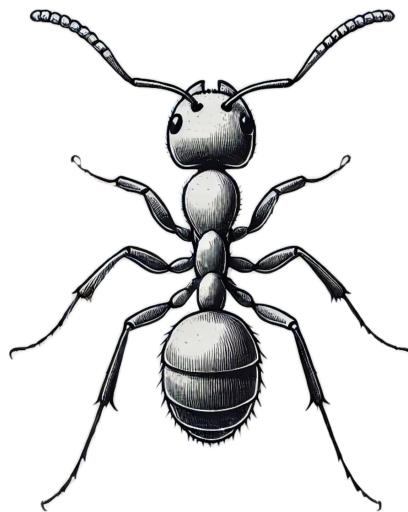
Creates intricate webs of connections, pattern completion abilities



Understanding Each Neural Network



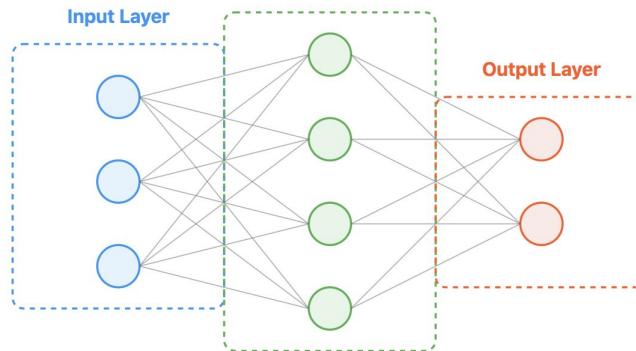
Ants



Feedforward Neural Networks

(The Building Block of AI)

Hidden Layer



How It Works: Pet Classification Example

Input Features

- Has Fur
- Size (inches)
- Weight (lbs)

Hidden Processing

Combines features to find patterns like:
"small + furry"
"medium + heavy"

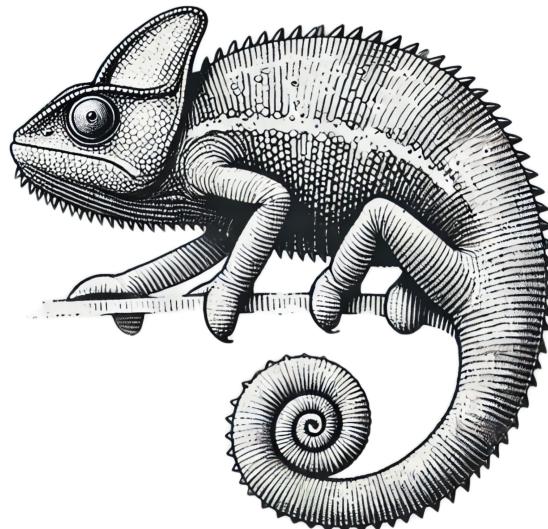
Output Decision

- Predicts:
- Cat
- Dog

Simple, organized in layers, works in colonies
like neurons



Chameleon



Adapts and transforms, creates convincing imitations

GAN

How GANs Work: The Art Creation Game

(Like the AI that creates images!)

The Creator (Generator)

Tries to create realistic content
Takes random noise as input
Learns to make better content
from feedback

The Judge (Discriminator)

Tries to spot fake content
Looks at real and generated
content to learn the
difference

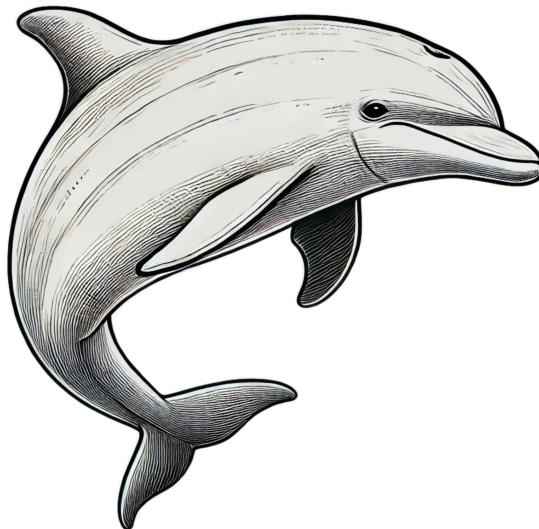
The Training Process



Just like an art student learning from a teacher's feedback,
the Creator gets better by learning from the Judge's critiques!



Dolphin



Highly intelligent, processes complex sequences, excellent at communication

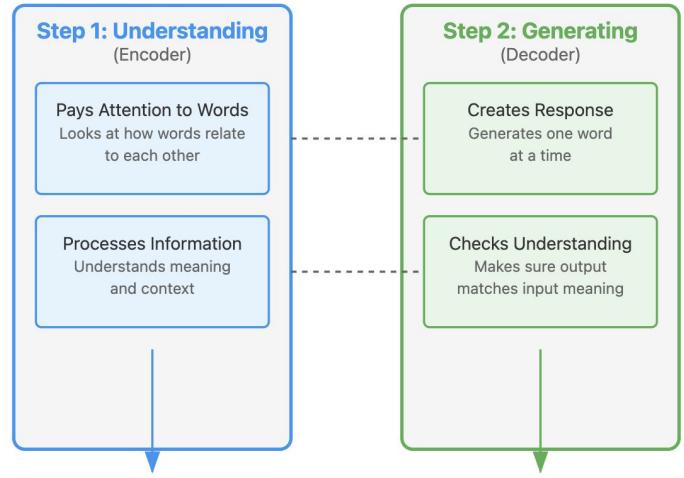
Transformers

How a Transformer Neural Network Works

(Like the one used in ChatGPT!)

Input: "Hello, how are you?"

Output: "Bonjour, comment allez-vous?"



Real-World Example:
When you input "Hello, how are you?" in English, Step 1 understands it's a friendly greeting asking about wellbeing. Step 2 generates the equivalent French greeting "Bonjour, comment allez-vous?"



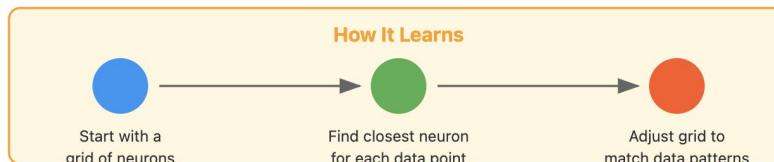
Honeybee



Organizes data into patterns, creates structured representations

Self-Organizing Maps

(How AI Organizes Data Like a Brain)

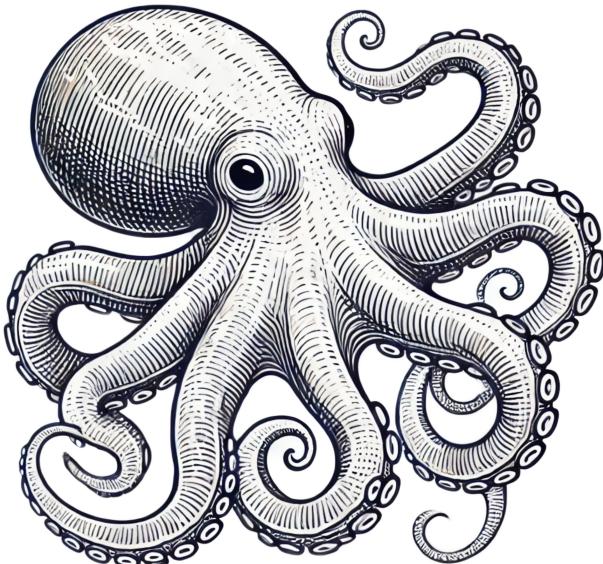


Real-World Example: Customer Segmentation

Imagine organizing customers based on their shopping habits:
Similar customers (like "luxury shoppers" or "bargain hunters")
naturally group together on the map!

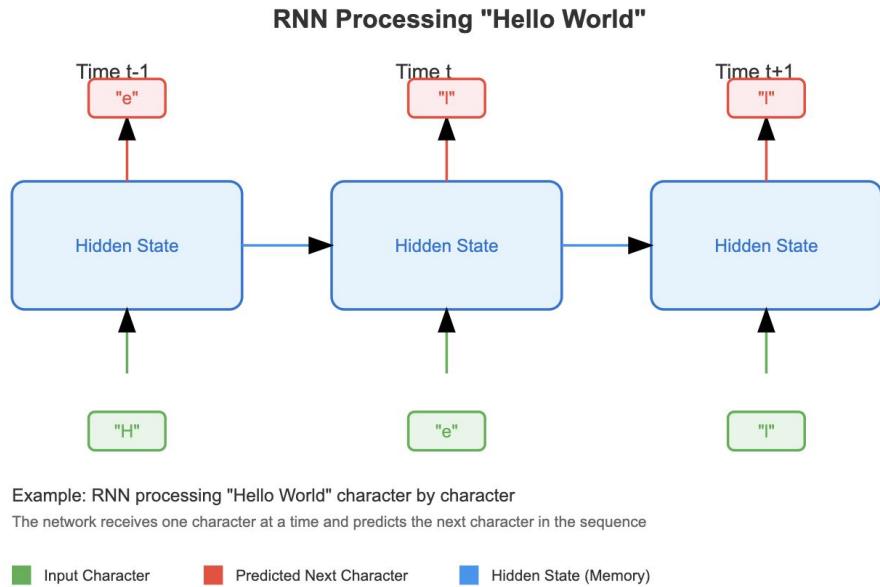


Octopus



Adaptable, processes information cyclically, has memory-like capabilities

RNN





Eagle

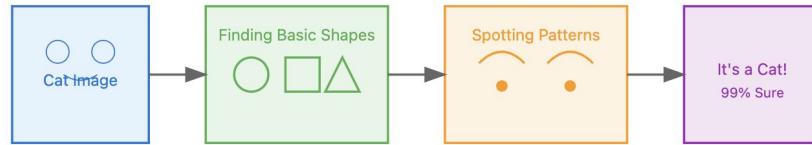


Exceptional at pattern recognition, processes visual information hierarchically

CNN

How CNNs Look at Images

(Like Having Super-Powered Eyes!)



How It Works:

1. Looks at Image

Like your eyes seeing a picture

2. Finds Simple Shapes

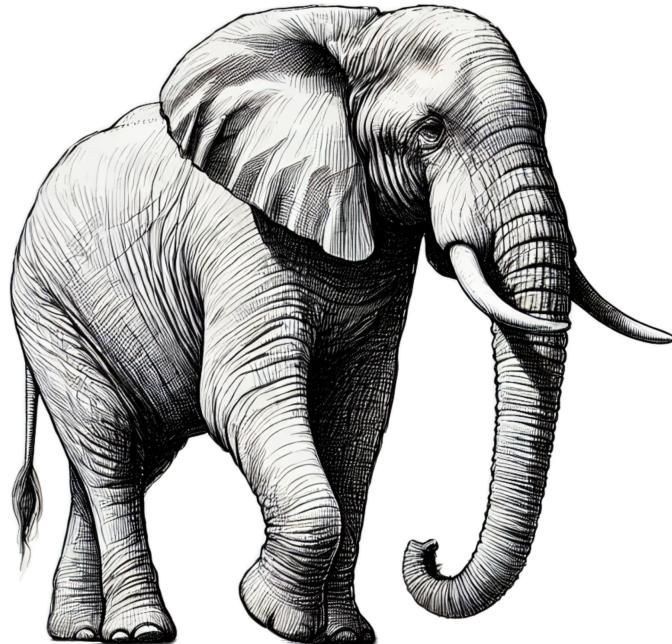
Spots edges, circles, and lines

3. Recognizes Patterns

Combines shapes into features (eyes, ears, whiskers)



Elephant



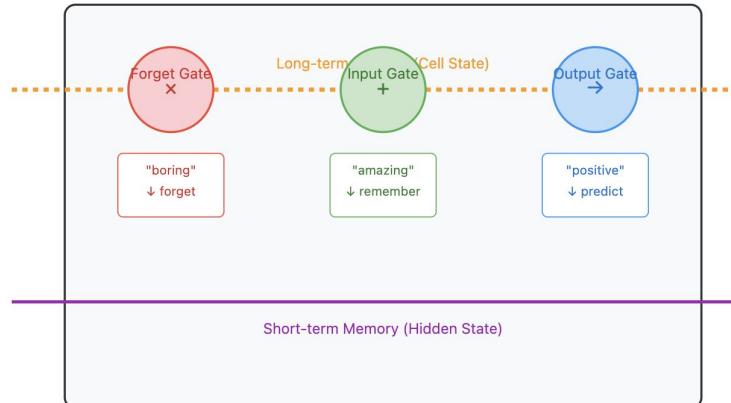
Superior memory retention, processes long sequences of information

LSTM

Understanding LSTM Memory

(Like a Smart Notepad that Remembers Important Things)

Example: Reading a Movie Review Word by Word

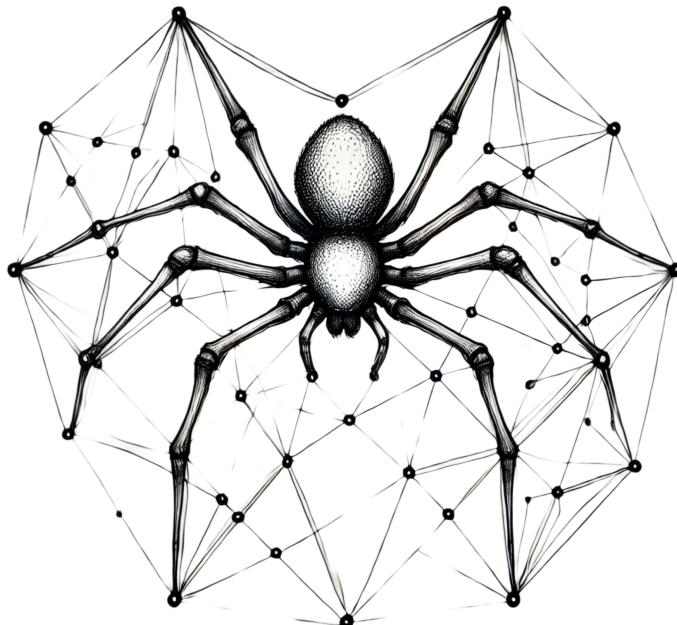


How LSTM Processes "This movie was boring but the ending was amazing!"

1. Forget Gate: Decides to forget "boring" as the ending changes the sentiment
2. Input Gate: Adds "amazing" to memory as it's more relevant for final opinion
3. Output Gate: Predicts "positive" based on remembering the amazing ending
→ The movie gets a positive rating despite mixed feelings!



Spider

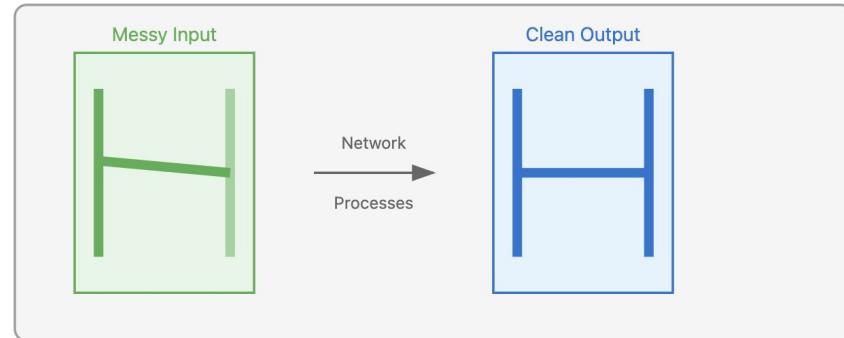


Organizes data into patterns, creates structured representations

Hopfield Networks

Like a spider's web capturing patterns, Hopfield Networks serve as content-addressable memory systems. They excel at pattern completion, taking noisy or incomplete inputs and reconstructing the complete learned pattern.

Hopfield Network: Pattern Memory





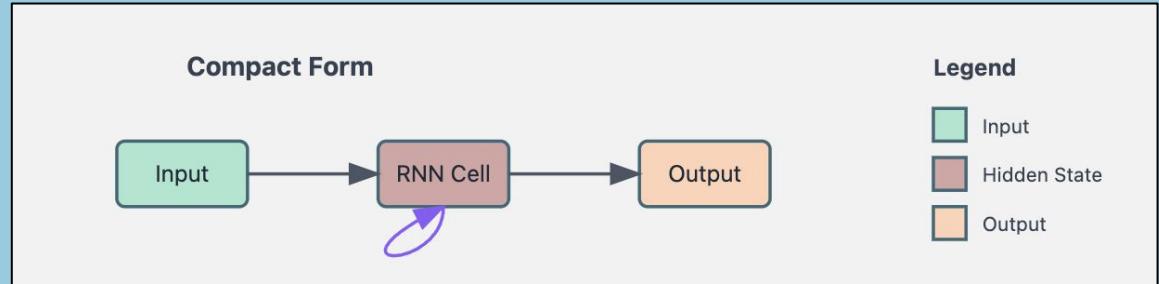
RNN Deep Dive



Recurrent Neural Networks



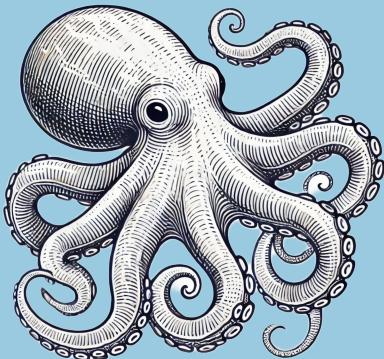
Like an octopus with its distributed neural system, RNNs process information in cycles, maintaining context through time.



RNNs excel at sequential data processing, making them ideal for tasks like natural language processing, time series analysis, and pattern recognition in temporal data. Their ability to maintain internal state and process sequences of varying lengths makes them particularly versatile among neural network architectures.



Sample (Simplified) Implementation



```
# Define RNN model class inheriting from PyTorch's Module
class SimpleRNN(nn.Module):
    def __init__(self, input_size, hidden_size, output_size):
        super(SimpleRNN, self).__init__()

        # Core RNN layer
        # input_size: dimension of input features
        # hidden_size: number of hidden units
        self.rnn = nn.RNN(input_size, hidden_size)

        # Output layer transforms RNN output
        # to desired output dimension
        self.fc = nn.Linear(hidden_size, output_size)

    def forward(self, x, hidden):
        # x shape: (seq_len, batch, input_size)
        # hidden shape: (1, batch, hidden_size)

        # Process input sequence through RNN
        out, hidden = self.rnn(x, hidden)

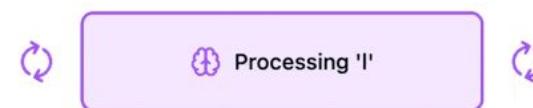
        # Transform RNN output to final output
        out = self.fc(out)

        # Return both output and final hidden state
        # for next sequence prediction
        return out, hidden
```



RNN Processing Sequential Data

Input: H e I I o



Output: e I I o ?

Current Input: 'I'
Predicted Next: 'o'



Conclusion

Neural networks in nature demonstrate remarkable efficiency and adaptability, having evolved over millions of years

By studying and emulating these biological systems, we continue to advance artificial neural network design and capabilities. The parallels between natural and artificial neural networks highlight the profound connection between biological intelligence and modern AI systems.



We'll now take questions