## **What is an RNN?**

**Recurrent Neural Networks (RNNs)** are a class of neural networks designed to handle **sequential data** — where the order of inputs matters.  
 Examples include:

* Time series (stock prices, sensor readings
* Text (sentences, paragraphs)
* Speech/audio sequences
* Video frame sequences

The **core idea** is that RNNs have a **“memory”**, they remember past information and use it to influence current output.

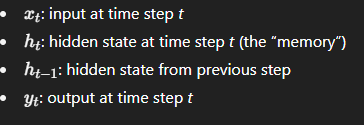
## **Why Traditional Neural Networks Fail on Sequences**

A standard **feedforward neural network** assumes that all inputs are **independent**.  
 For example, if you feed the words: I love cats

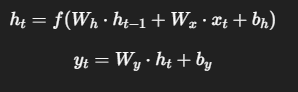
A normal NN will treat “I”, “love”, and “cats” as three unrelated words, losing the contextual dependency. But in sequences, **each word depends on the previous one.**Hence, we need something that can **retain the state** of what has come before - **RNN**.

## **How RNN Works (Step-by-Step)**

Let’s define:

**

The RNN updates are:



Here:

* fff is typically **tanh** or **ReLU** — a nonlinear activation.
* The same weights Wx,Wh,WyW\_x, W\_h, W\_yWx​,Wh​,Wy​ are **shared across all time steps** (that’s what makes it “recurrent”).

## **Key Challenges in RNNs**

### **1. Vanishing Gradient Problem**

During backpropagation through many time steps, gradients can shrink exponentially → the network forgets long-term dependencies.

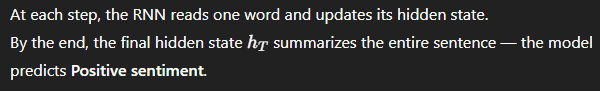
### **2. Exploding Gradient Problem**

Conversely, gradients can blow up, making training unstable.

### **Solutions:**

* **LSTM (Long Short-Term Memory)** — adds gates (input, forget, output) to control information flow.
* **GRU (Gated Recurrent Unit)** — a simplified version of LSTM.
* **Gradient clipping** — limits gradient size.
* **Sequence truncation** — limits backpropagation length.
* Real-World Example

**Sentiment Analysis-** Input sentence: I really enjoyed this movie.



## **Intuitive Analogy**

Think of an RNN as a **reader with short-term memory**:

* Each word you read updates your understanding of the sentence.
* You don’t remember every letter, but your mind updates the meaning based on context.
* At the end, you can summarize, similar to how an RNN produces the final output.

**RNNs introduced the idea of sequential dependency and temporal memory, but their limitation in capturing long-term dependencies led to LSTM and GRU architectures, which later evolved into Transformers that rely on attention instead of recurrence.**