Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs), which were the brains behind early AI systems for language, speech, and time-series prediction before Transformers came along.

Let’s make this one clean, structured, and professional — perfect for your technical writing portfolio.

Overview

A **Recurrent Neural Network (RNN)** is a type of neural network designed to handle **sequential data** — data where order matters.
Unlike traditional networks, RNNs remember previous inputs through internal memory, allowing them to make predictions based on **context over time**.

They were the foundation for early **natural language processing (NLP)** systems, **speech recognition**, and **time-series forecasting**.

Key Concepts

1. Sequence and Memory

RNNs process data **one element at a time**, maintaining a **hidden state** that carries information from earlier steps in the sequence.

For example, in a sentence:

> “Leslie loves writing technical documentation.”

An RNN reads one word at a time and uses memory from previous words to predict what comes next — maintaining *context*.

---

2. Hidden State

The hidden state acts like the network’s “memory.”
At each step, it’s updated based on:

New hidden state = f(previous hidden state, current input)

This allows RNNs to make decisions not just from the current input, but from everything seen before.

---

3. Long Short-Term Memory (LSTM)

A special RNN architecture designed to overcome the **vanishing gradient problem** (where early information is forgotten).
LSTMs can remember information over **longer sequences** — e.g., whole paragraphs, not just sentences.

---

4. Gated Recurrent Unit (GRU)

A simplified version of LSTM that performs similarly but with fewer parameters, making it faster to train.

Visual Summary

Input (x_1) → Hidden (h_1) → Output (y_1) ↓ Input (x_2) → Hidden (h_2) → Output (y_2) ↓ Input (x_3) → Hidden (h_3) → Output (y_3)

The hidden state “flows” through the sequence, carrying past context forward.

Real-World Applications

- 💬 **Language Modelling:** Predicting the next word in a sentence.
- 🎙️ **Speech Recognition:** Understanding spoken commands or dictation.
- ✍️ **Text Generation:** Writing captions, lyrics, or stories.
- 💹 **Stock Market Forecasting:** Predicting trends over time.
- 🩺 **Healthcare:** Analysing patient time-series data for diagnosis.
- 📈 **Sensor Data Analysis:** Predictive maintenance in IoT systems.

Example

Suppose you want to predict the next temperature reading from a weather station.

An RNN receives:

Inputs: [22°C, 24°C, 23°C, 25°C] → predicts: 26°C

It learns the pattern of how temperatures evolve over time.

Why It Matters

RNNs introduced the ability for AI to “remember” — a key leap from static models.
They paved the way for voice assistants, translation systems, and chatbots.
While largely replaced by **Transformers**, RNNs remain useful for smaller, real-time applications.

Famous RNN Architectures

- **Elman Network (1990):** The original simple RNN.
- **LSTM (1997):** Solved long-term memory loss.
- **GRU (2014):** A lighter, faster alternative to LSTM.
- **Seq2Seq (2014):** Used in machine translation before Transformers.

Real-World Systems Using RNNs

- **Google Translate (early versions)**

- **Apple Siri (before Transformer updates)**
- **Predictive text keyboards**
- **Financial time-series forecasting systems**