THE CORRELATION

Recurrent Neural Networks

By

thecorrelation.in

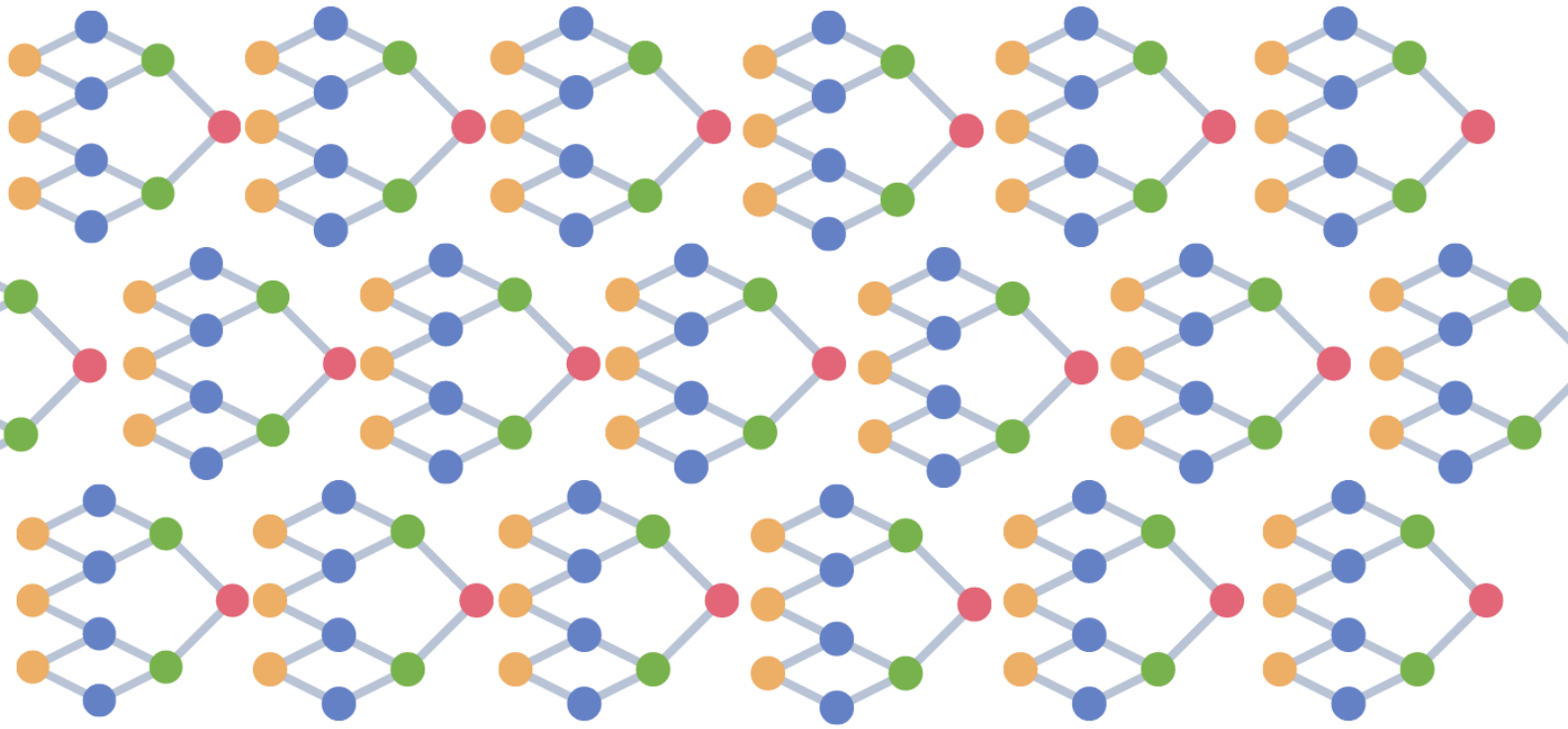
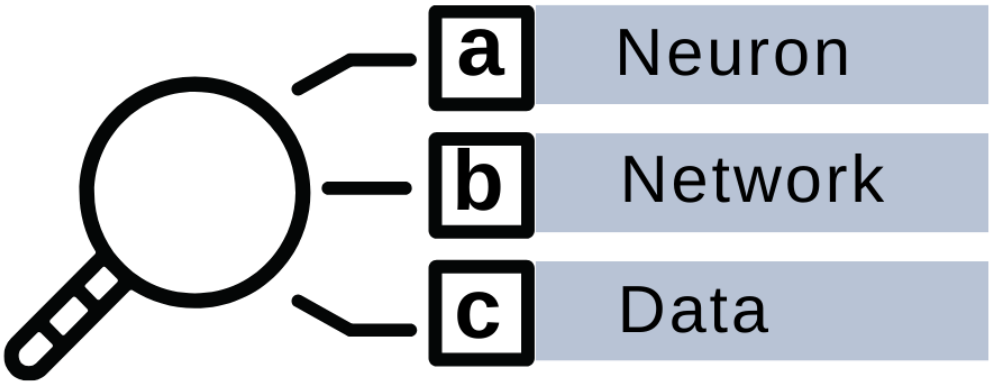


Table of Contents

Introduction to Recurrent Neural Network

- **Read this Sentence and try to complete it from the following options.**

We are revising Recurrent Neural _____.

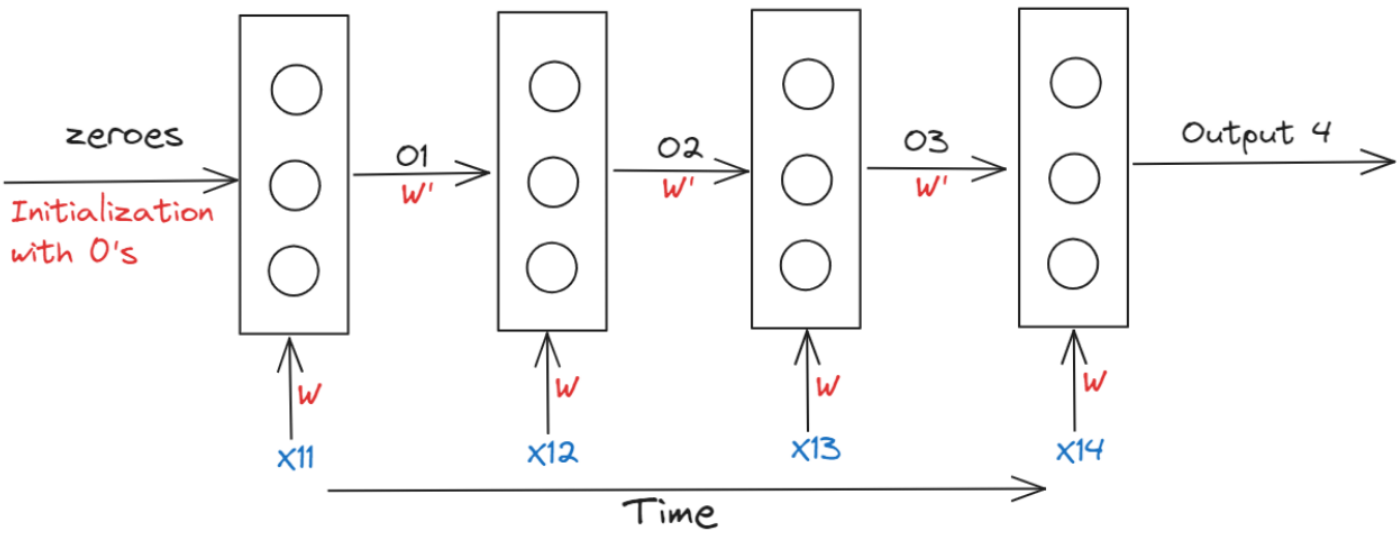


Were you able to think about next word ?

Now, Imagine doing the same with computer. That;s where RNN comes into play.

Recurrent Neural Network

An RNN **(Recurrent Neural Network)** is a type of artificial intelligence model used in machine learning and natural language processing. It's designed to process **sequential data**, like sentences or time-series data, where the **order of the information matters**.



where,

- O1, O2, O3 - Outputs
- W - Weights of Inputs
- W' - Weights of Outputs
- X11, X12.. - Words Input

RNN Use cases

We use RNN (Recurrent Neural Network), when we have a **Sequence in Data.**

Text Data

Stock Market Prediction

Machine translation

Speech Recognition

Image Caption

Alexa, Siri etc

Training RNN

The Training of RNN happens in four simple steps:

1. In the first step it does **forward pass** and makes prediction

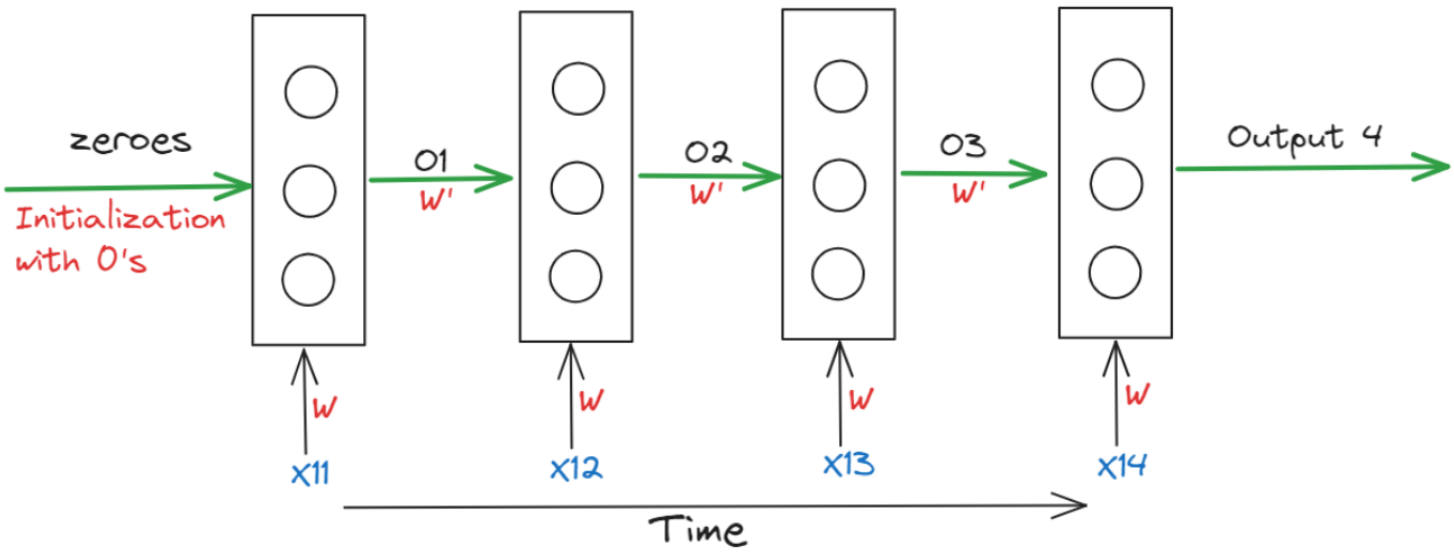
2. In the second step , it compares ground truth to prediction using **loss function** like log –loss or RMSE. and the loss function **outputs an error** value which is an estimate of how poorly your network performed.

3. In the final step it uses that error to do a **back propagation** which calculates the gradients for each node in network.

Training RNN

1 Forward Propagation

Forward propagation is the process in neural networks where input data is passed through the layers of the model, from the input to the output, to make predictions.



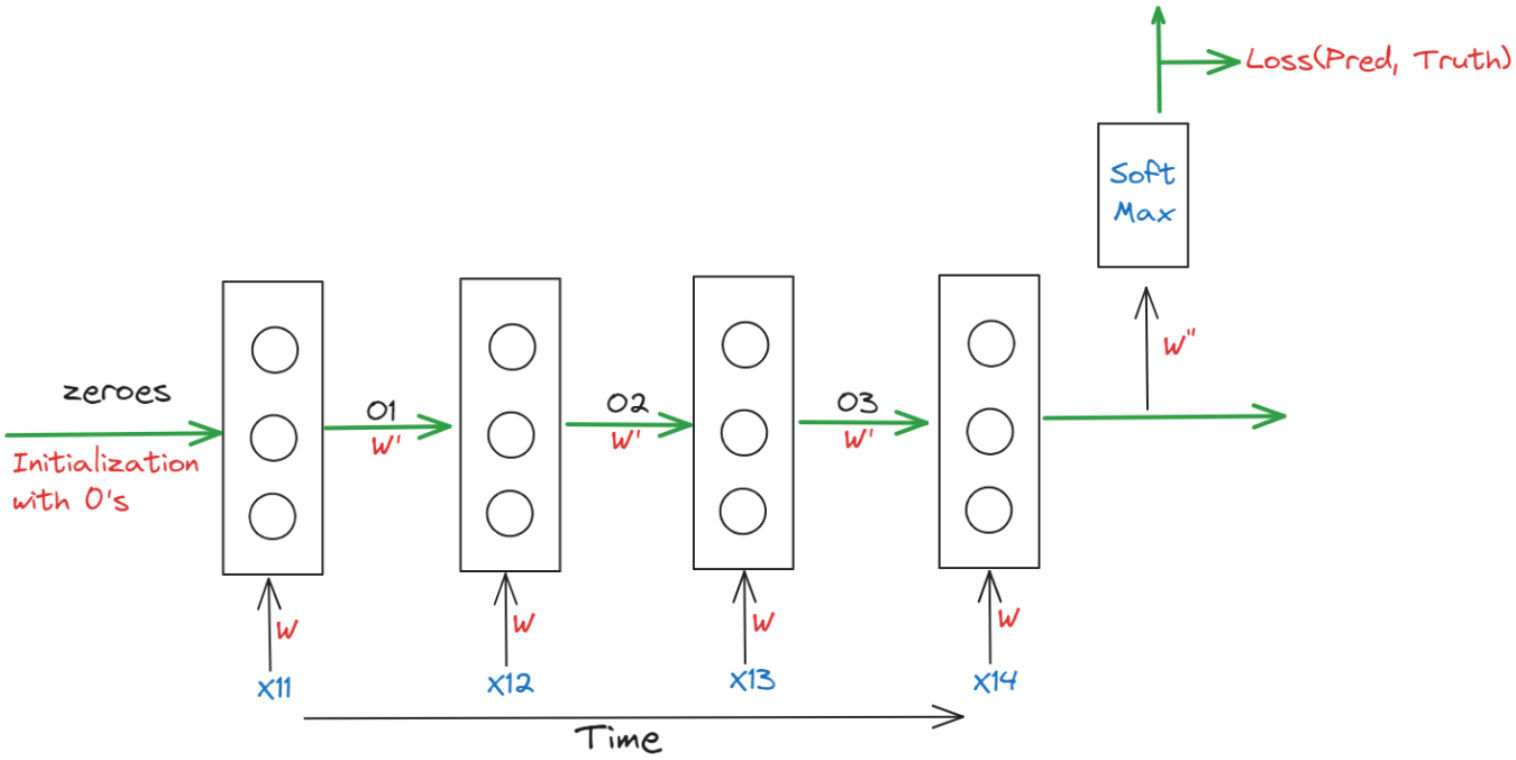
where,

- O1, O2, O3 - Outputs
- W - Weights of Inputs
- W' - Weights of Outputs
- X11, X12.. - Words Input

Training RNN

2 Loss Function & Error

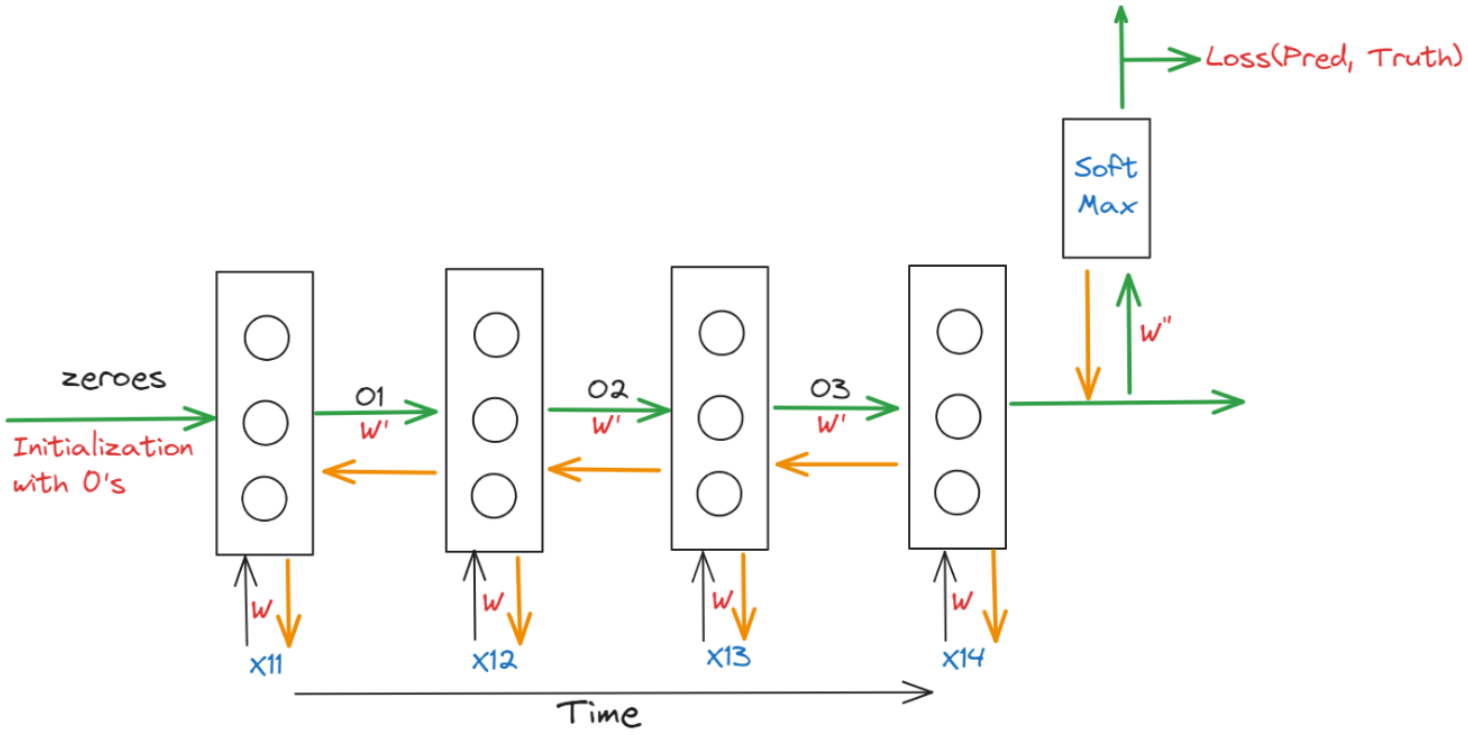
Loss function measures the difference between model predictions and actual values, guiding the model during training to minimize errors and improve performance.



Training RNN

3 Backward Propagation

Backward propagation, also known as backpropagation, is like learning from mistakes. The model calculates the error in its predictions, then adjusts its parameters backward through the layers to minimize errors and improve accuracy during training.



Advantages of RNN

ADVANTAGES

Ability to process sequential data: RNNs can handle sequences like text, audio, and time-series data where order matters.

Memory of past information: RNNs maintain a hidden state, enabling them to remember previous inputs, useful for context-based tasks.

Disadvantages of RNN

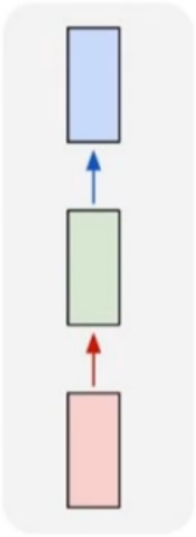
Vanishing and Exploding Gradients: In long sequences, gradients can diminish or explode, leading to difficulties in learning and slower convergence.

Limited Memory: Simple RNNs struggle to retain information for very long sequences, affecting their ability to capture long-term dependencies.

Less Robust for Long Sequences: Their performance degrades as the sequence length increases, especially compared to more advanced RNN variants like LSTM and GRU.

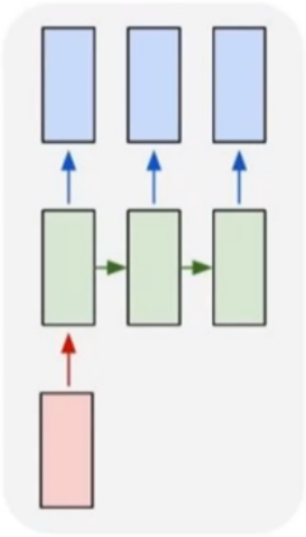
Types of RNN

one to one



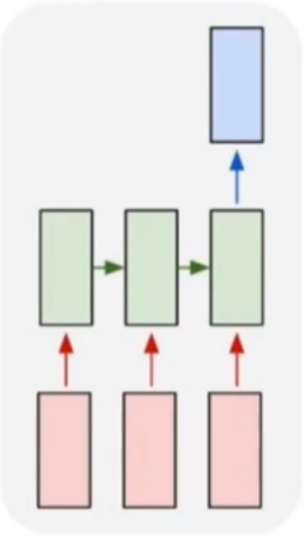
1

one to many



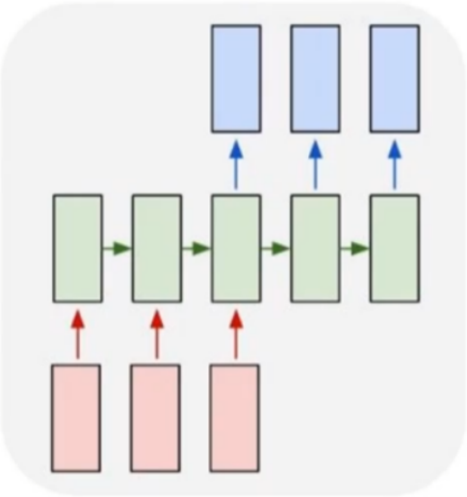
2

many to one



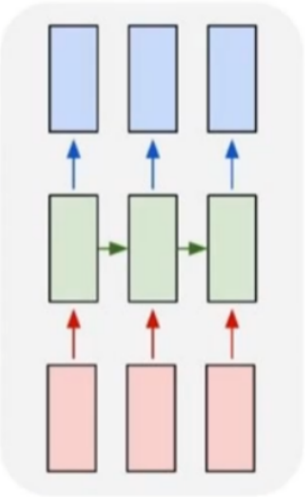
3

many to many



4

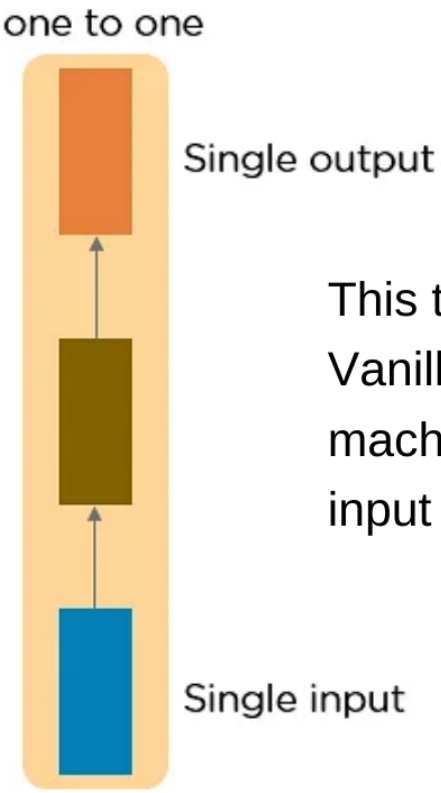
many to many



5

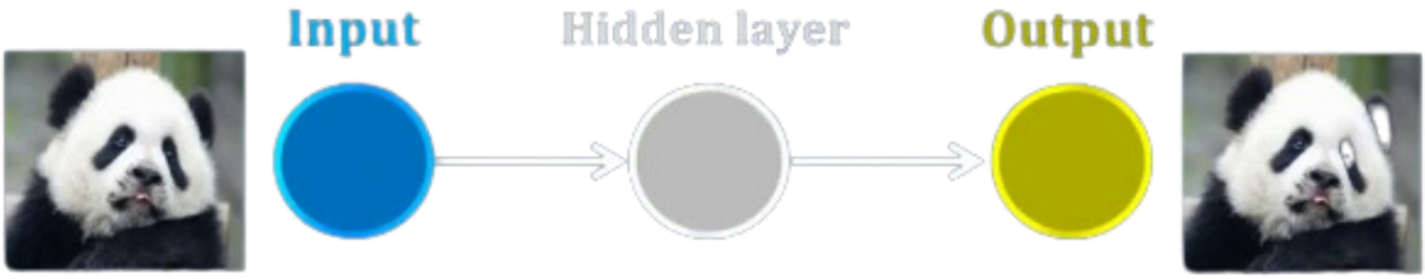
Types of RNN

One-to-one



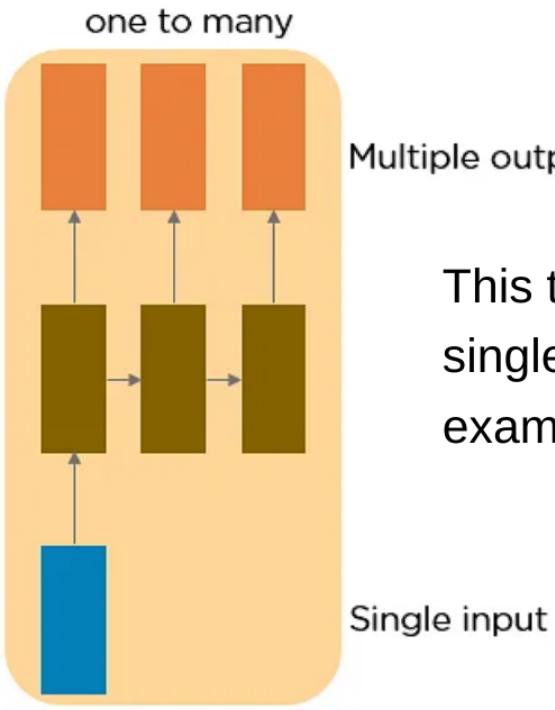
This type of neural network is known as the Vanilla Neural Network. It's used for general machine learning problems, which has a single input and a single output.

Example: classification of images.



Types of RNN

One-to-many

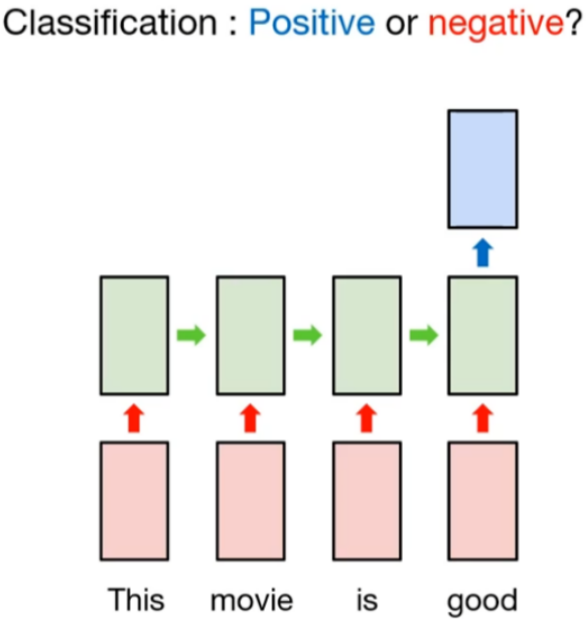


This type of neural network has a single input and multiple outputs. An example of this is the image caption.

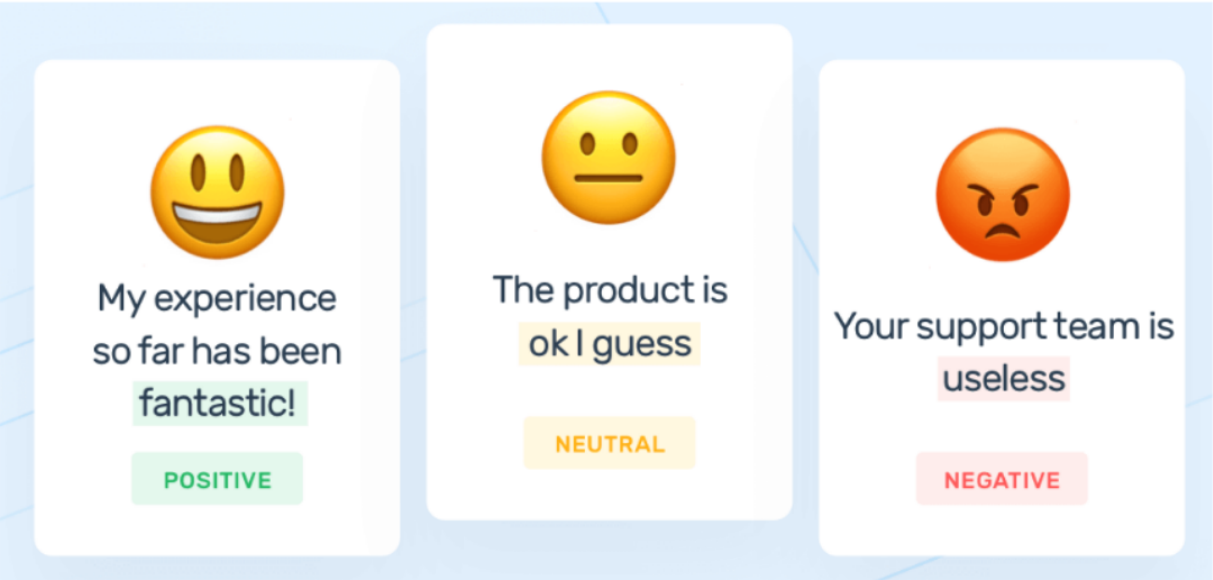
Example: Image Captioning

Types of RNN

Many-to-one

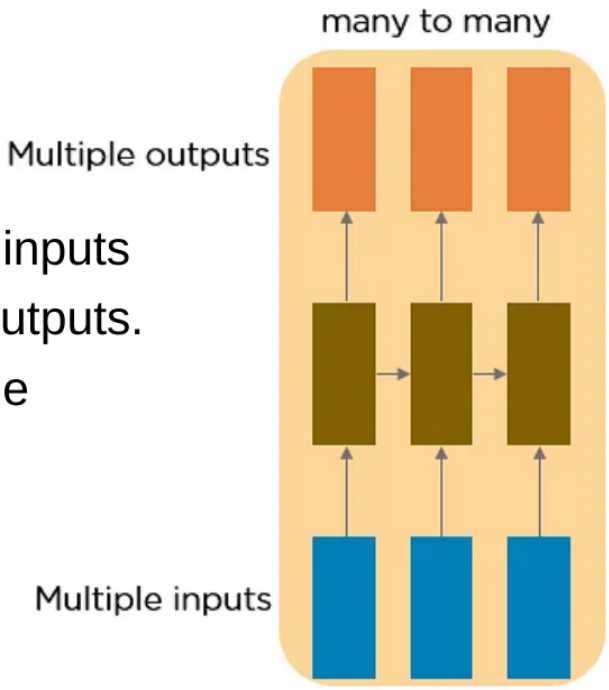


An example of a "Many-to-one" RNN could be a sentiment analysis task.
In this case, the RNN takes multiple sentences as input, and the goal is to predict the sentiment (positive or negative) of each sentence.



Types of RNN

Many-to-many



This RNN takes a sequence of inputs and generates a sequence of outputs. Machine translation is one of the examples.

Example: Machine Translation

Conclusion

Recurrent Neural Networks (RNNs) provide a powerful framework for processing *sequential data*. They excel in capturing dependencies over time and have found applications in various domains such as natural language processing, speech recognition, and time series analysis.

Vanishing and exploding gradients can impact the training of RNNs. Techniques like *gradient clipping and weight initialization* help mitigate these issues. Understanding the mathematical formulation of RNNs provides insights into their functioning and optimization.

THE CORRELATION

School of Applied Data Science
And Artificial Intelligence

thecorrelation.in

To Know more
visit our learning portal