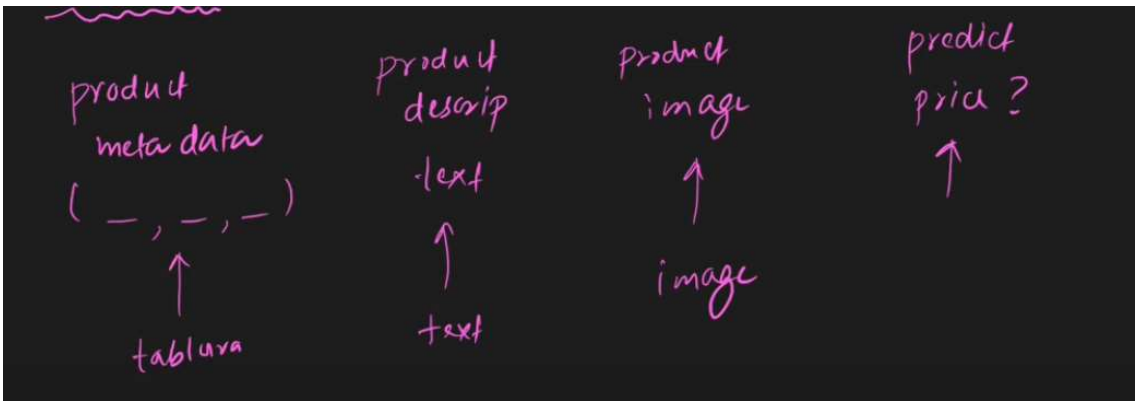
Non-Linear Topology

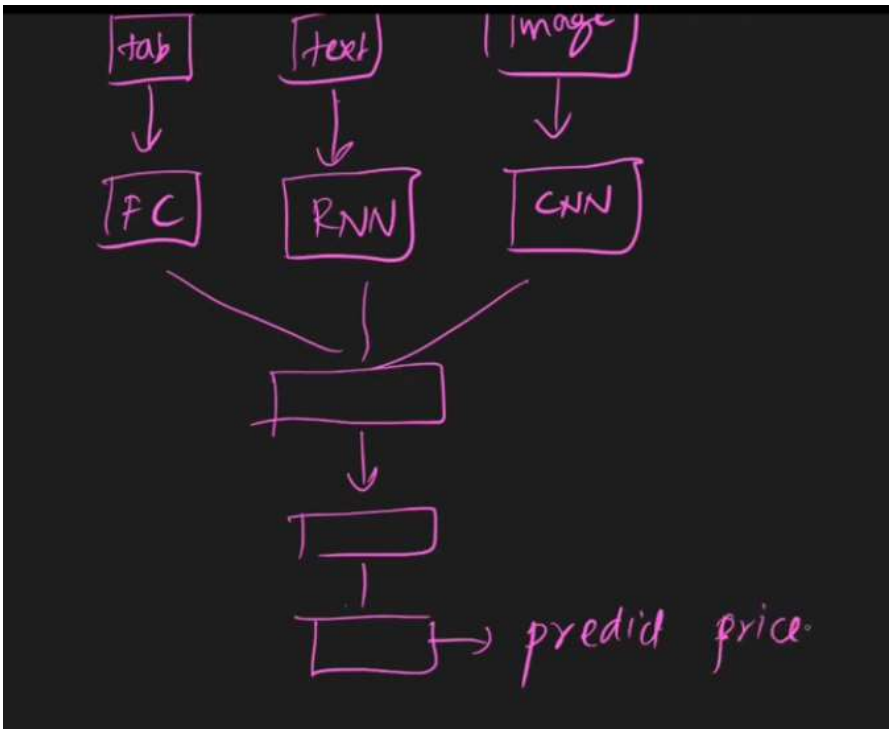
Example: E-commerce data in which there is product meta-deta (in tabular form), product description (in text form), product image (in image form) and tell its price?

- For tabular data – ANN model solve issue
- For image data – CNN model solve issue
- For text data – CNN fail (because in text data is sequential which CNN cannot handle – to solve this we go for **Recurrent Neural Network**)

This is resolved by Functional API's in keras.



Functional API's in keras



Key Point:

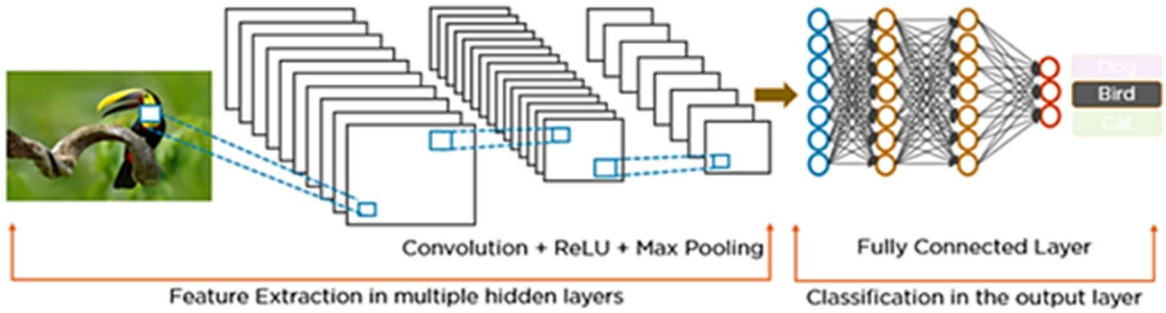
Sequential Topologies (Linear Topology) involve a straightforward, layer-by-layer approach to network design.

Functional Topologies (Non-Linear Topology) allow for more complex and flexible connections between layers, accommodating non-linear and advanced architectures.

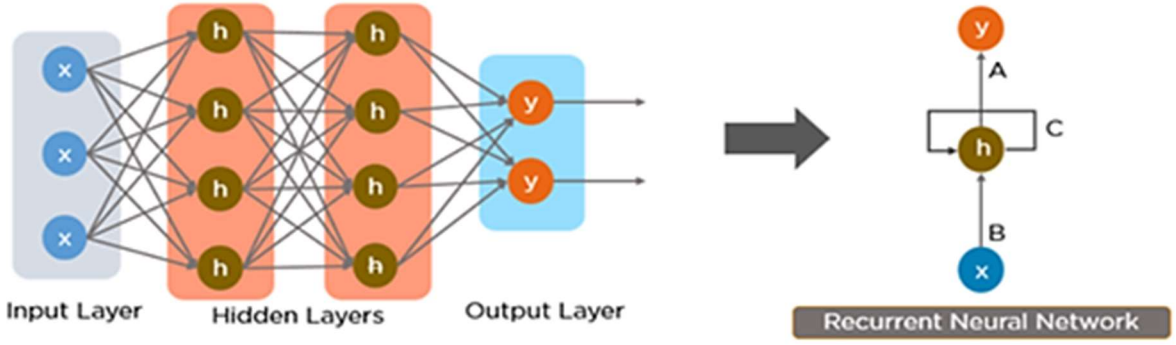
CNN Vs RNN:

- CNNs excel at spatial data and are widely used for image-related tasks.
- RNNs excel at sequential data, capturing temporal or ordered dependencies, making them ideal for tasks like text or time-series analysis.

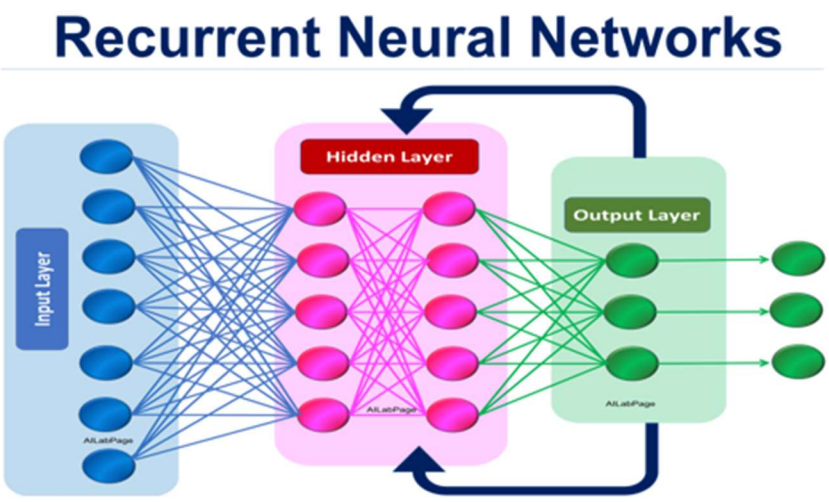
Convolutional Neural Network



Recurrent Neural Network



Note: In RNN there is informed processing. It also takes previous layers results.

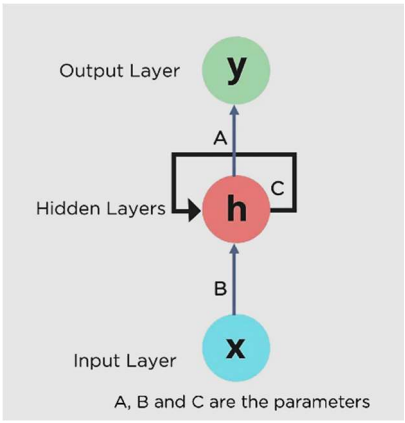


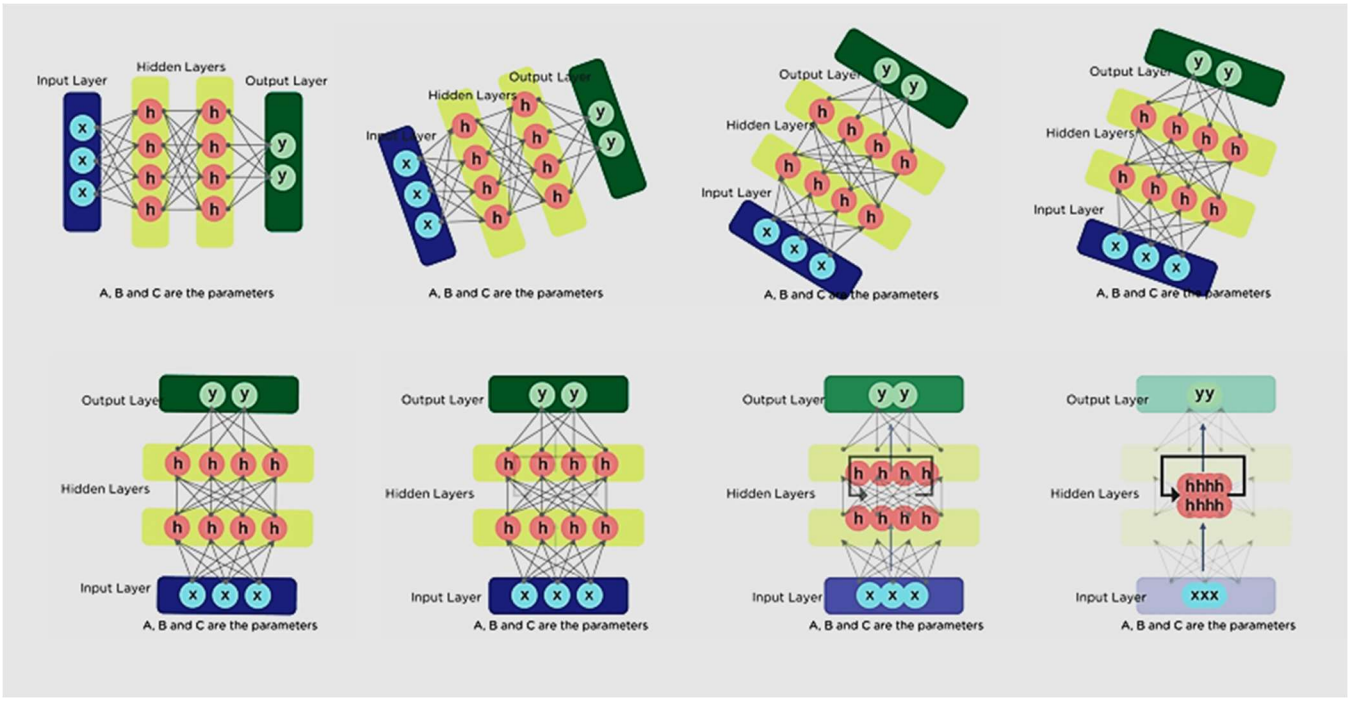
These kind of networks relay on a simple principle, feeding the output of a layer back to it's self, thus creating a feedback loop between different layers of the network. Working with sequential data, the order in which information appears is important, by feeding back the previous input to the current layer, the network keeps a record of the input signal and can find patterns between different parts of the input. More advanced model follow this type of architecture such as LSTMs and GRUs.

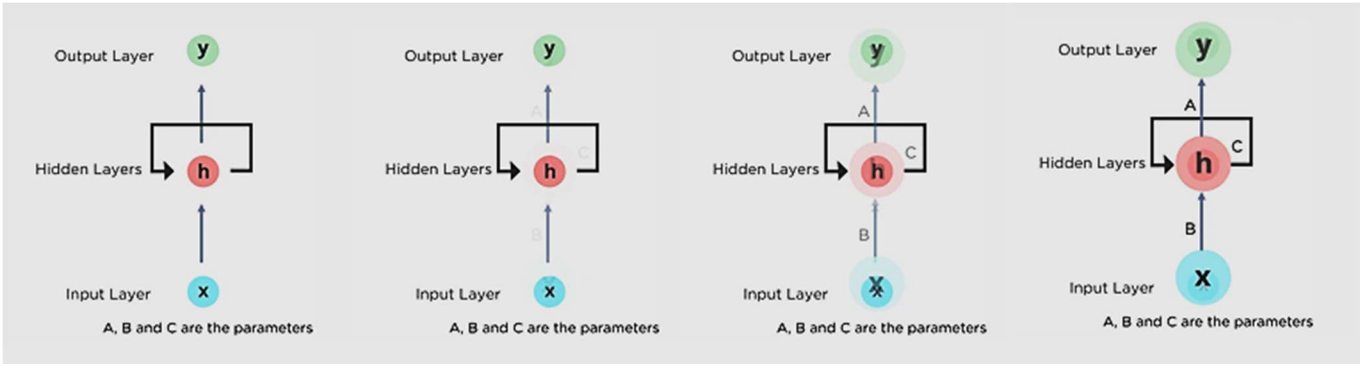
Comparision

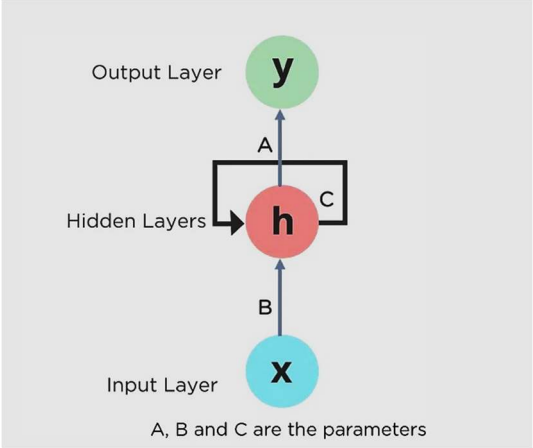
Aspect	Convolutional Neural Network (CNN)	Recurrent Neural Network (RNN)
Primary Use	Image and spatial data processing	Sequential data processing (e.g., text, time series)
Architecture	Convolutional layers that extract spatial features	Recurrent layers with loops to capture temporal dependencies
Input Type	Fixed-size inputs (e.g., images)	Variable-length input sequences (e.g., text, speech)
Data Handling	Captures local spatial patterns	Captures temporal dependencies and order of data
Key Components	Convolutional layers, pooling layers, fully connected layers	Recurrent layers, with variants like LSTM or GRU
Strengths	Image classification, object detection, spatial data	Language modeling, machine translation, time-series prediction
Information Flow	Feedforward, layer by layer	Sequential, with information passed through time steps
Example Applications	Image recognition, object detection, video analysis	Text generation, speech recognition, sentiment analysis

RNN Animation









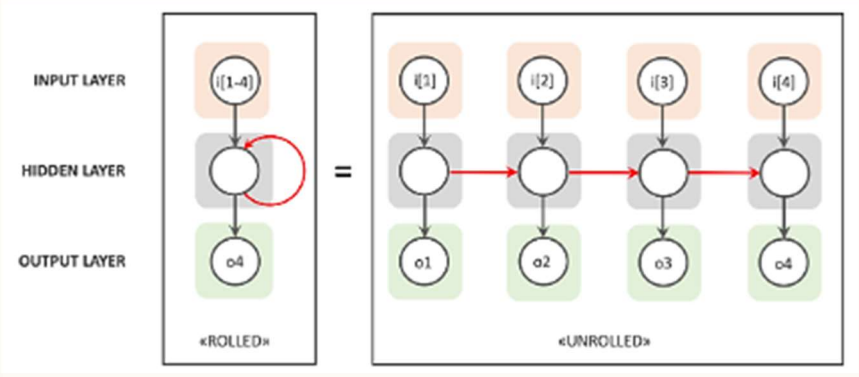
Basic attributes of RNNs

Looping through sequential data

RNNs support processing of sequential data by the addition of a loop. This loop allows the network to step through sequential input data whilst persisting the state of nodes in the Hidden Layer between steps - a sort of working memory.

In each iteration the Hidden Layer inherits the working memory from the previous iterations, as indicated by the red arrow.

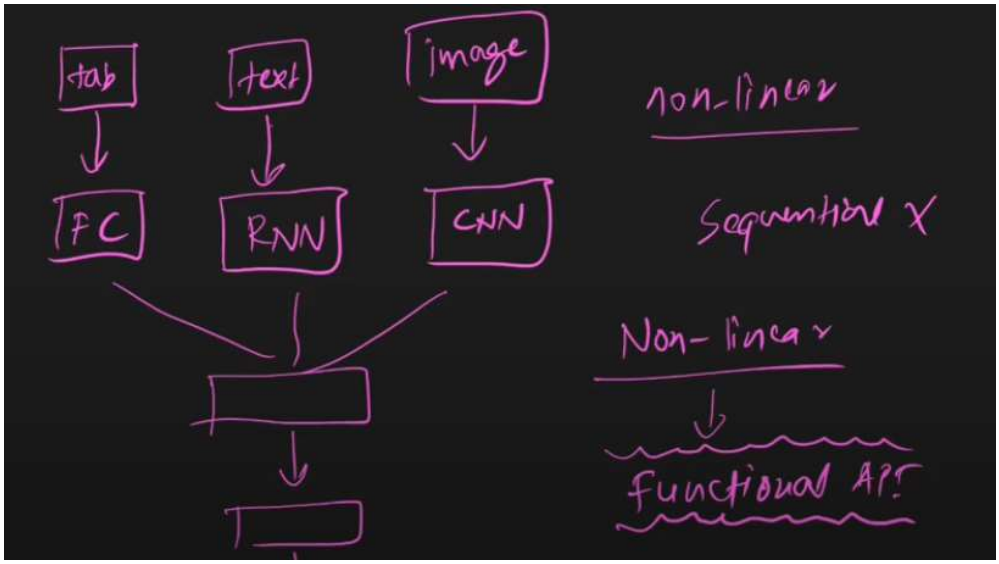
On the right hand side we see an “unrolled” representation of the same RNN, which presents the RNNs loops as a chain of identical Feed Forward ANNs.



A RNN can be viewed as many copies of a Feed Forward ANN executing in a chain

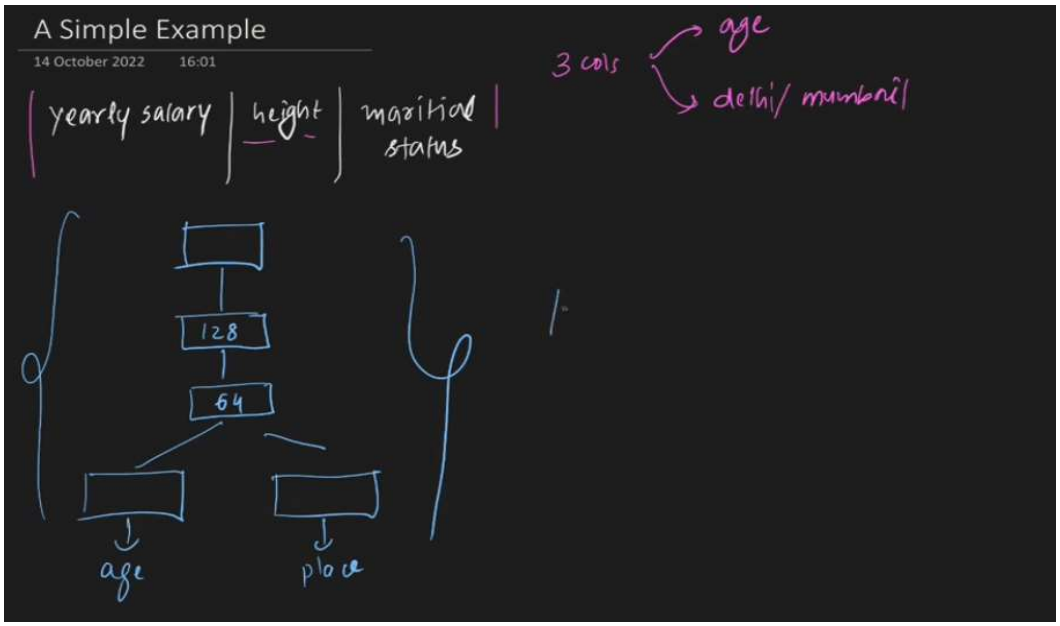
Reference: https://www.bouvet.no/bouvet-deler/explaining-recurrent-neural-networks

Functional API



Example: Given that: Salary, Height and marital status. Find its age and location.

Answer: Use Functional API



Note: Sequential API does not carry previous information. While Functional API carry these information.

Cosine similarity (embedding technique) is a metric used to measure how similar two vectors are, based on the angle between them. It's particularly useful for comparing the similarity between embedding vectors.