* Unit 5: Convolutional Neural Network.

- Convolutional Neural Network
- Recursive Neural Network
- Recurrent Neural Network
- long-Short Term Memory
- Gradient Descent Optimization.

* Convolutional Neural Network:

-

* Recurrent Neural Network:

- RNN is a type of Neural Network where the output from the previous step are fed as an input to current step.

- Need:
- In traditional Neural Networks, all the inputs and outputs are independent of each other.
- In cases, like when it is required to predict the next word of a sentence, the previous words are required. Hence, there is a need to remember the previous words.
- Here RNN come into existence, which solved the issue with the help of hidden layer.

- The most important feature of RNN is a hidden state, which remembers some info about sequence.

- RNN has a memory which remembers all info. about what has been calculated.

*working of RNN:

- Here, "x" is input layer, "h" is hidden layer, "y" is the output layer.

- At any given time t, the current input is a combination of input at x(t) and x(t-1).

- The output at any given time is fetched back to network to improve on the output.

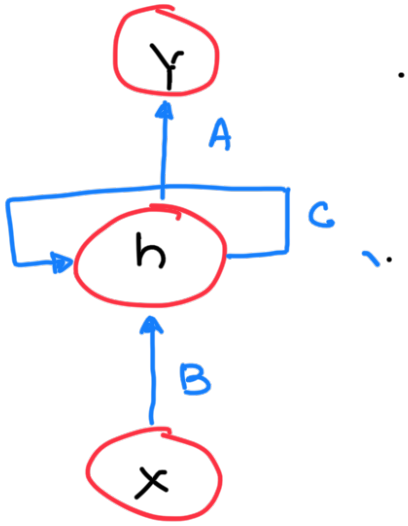


Fig. 1.

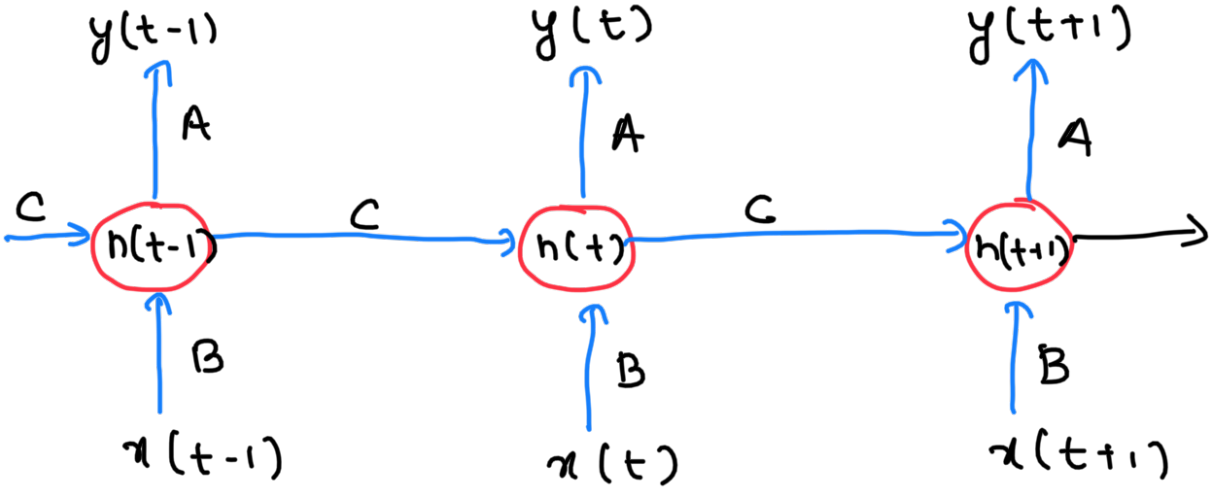


Fig. 2.

- RNN converts the independent activations into dependent activations by providing the same weights and biases to all layers, reducing complexity of increasing parameters and memorizing each previous output by giving it to input of next hidden layer.

- Fig. 2 shows that h(t) is dependent of h(t-1) and h(t+1) is dependent on the h(t).

- Formula for calculating current state:

$$h_t = f(h_{t-1}, x_t)$$

Here,

h_t = Hidden layer of current state.
h_{t-1} = Hidden layer of previous state
x_t = input to current hidden layer

- Formula for applying Activation function:

$$h_t = \tanh(w_{hh} h_{t-1} + w_{xh} x_t)$$

w_{hh} = weight for recurrent neuron
w_{xh} = weight at input neuron.

- for output :

$$Y_t = w_{hy} h_t$$

w_{hy} = weight at output layer.

* Training :

- First, the input is provided to the network.

- Then calculate its current state using current input and its previous state.

- The current h_t becomes h_{t-1} for next step.

- One can go as many times and join the information from all the previous states.

- Once all steps are completed, the final current state is used to calculate the output.

- Then output is compared to actual output and error is generated.

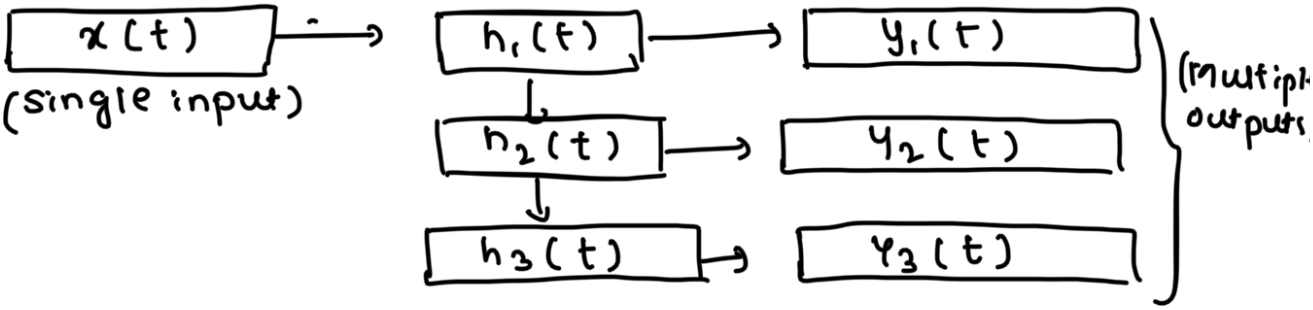
- Error is then back propagated to network and update the weights and hence the network is trained.

* Types :

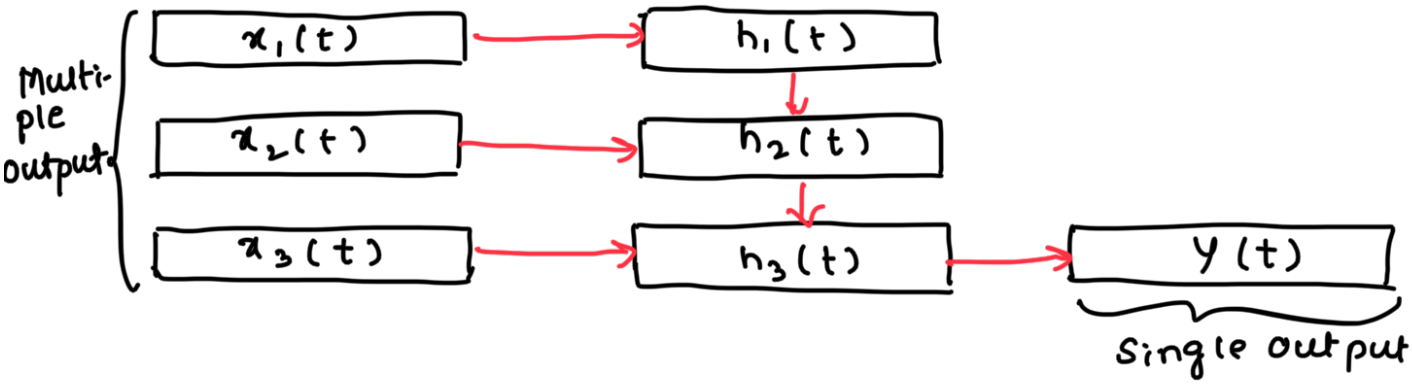
1. One to one: Type of netural network is known as Vanilla Neural Network. It's used for general machine learning problems, which has a single input and single output.

RNN

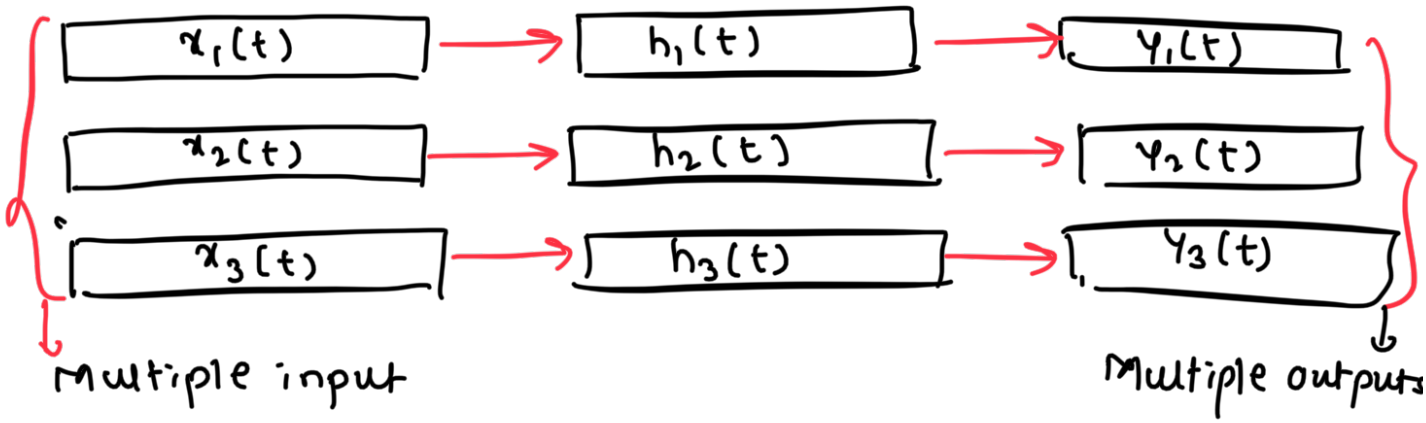
2. One to Many RNN: Type of neural network has a single input and multiple outputs. An example is image caption.



3. Many to one RNN : It takes sequence of input and generates a single output. Sentiment analysis is a example of this network where a given sentence can be classified as expressing positive or negative sentiment.



4. Many to Many RNN : Takes sequence of inputs and generates sequence of outputs.



* Issues with RNN :

1. Vanishing Gradient Problem:

- RNN is hard to train because of the gradient problem.
- The gradients carry information used in

RNN, and when the gradient becomes too small, the parameter updates becomes insignificant.

2. Exploding Gradient Problem:

- While training a neural network, if the slope tends to grow exponentially instead of decaying, this called Exploding Gradient.

- This problem arises when large error gradients accumulate.
- Long Training Time, poor performance and bad accuracy are major issues in gradient problems.

* Advantage :

1. Remembers each and every piece of info through time.
2. Used with convolutional layers.

* Disadvantage :

1. Gradient Vanishing and Exploding problems.
2. Training RNN is very difficult.
3. Cannot process very long sequences.

* Applications :

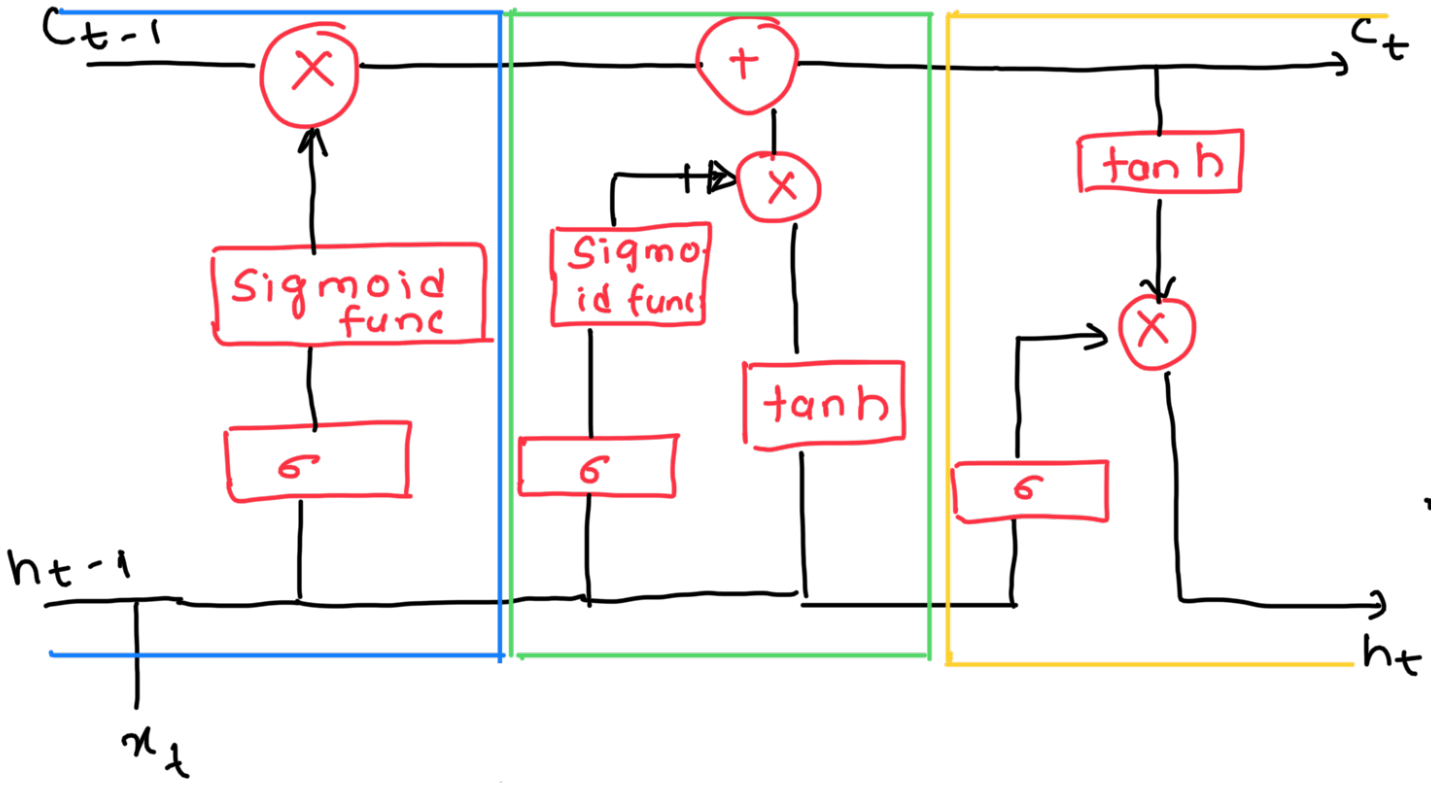
1. Language modelling and Generating Text
2. Speech Recognition.
3. Machine Translation.
4. Image Recognition.
5. Time series Forecasting.

* Long-Short Term Memory :

- Kind of recurrent neural network.
- It provides solution to the problems of RNN i.e. vanishing and exploding gradients.

* Working:

- A LSTM unit is composed of a cell, an input gate, an output gate and a forget gate.



- The C_{t-1} to C_t line that runs all over the top of unit represents the Long Term Memory. Although the long term memory can be modified by multiplication (X) and sum (+), their are no weights and bias which can modify it directly. The lack of weights allows Long-Term Memories to flow through the units without causing the gradient to explode or vanish.

- The h_{t-1} to h_t line that runs over the lower of unit represents the Short Term Memory. Short term memory is directly connected to weights that can modify them.

1. Forget Gate(■) :

- The blue box part is Forget Gate.

- Two inputs h_{t-1} and x_t are fed to the gate and multiplied with weights followed by the addition of Bias.

- The resultant is passed through an sigmoid function which gives binary output.

- If the cell state output is 0, then the piece of info is forgotten.

- If it's 1, the information is retained for future use.

2. Input Gate: (■)

- The Green box is input gate part.

- First, the short term memory (h_{t-1}) and the input (x_t) is multiplied by weights and added the bias(σ) to it.

- Then, it is given to sigmoid function which gives binary output.

- Again, the h_{t-1} and x_t is multiplied by weights and added to bias and given to tanh activation function.

- The tanh activation function returns value from -1 to 1.

- The binary output from sigmoid and tanh

function output is multiplied and at last added to long term memory.

3. Output Gate : ( )

- The input gate has updated the Long Term Memory.
- In output gate, the updated New Long Term memory is used as input to Tanh activation function.
- The short term memory is then added to bias and given to sigmoid function which generates the binary output.
- The binary output and the output generated by tanh function is then multiplied and given as the new short term memory i.e. the output of LSTM.
- The output of short term memory is given as input to next cell.

* Applications:

1. Language Modelling.
2. Machine Translation
3. Image Captioning.