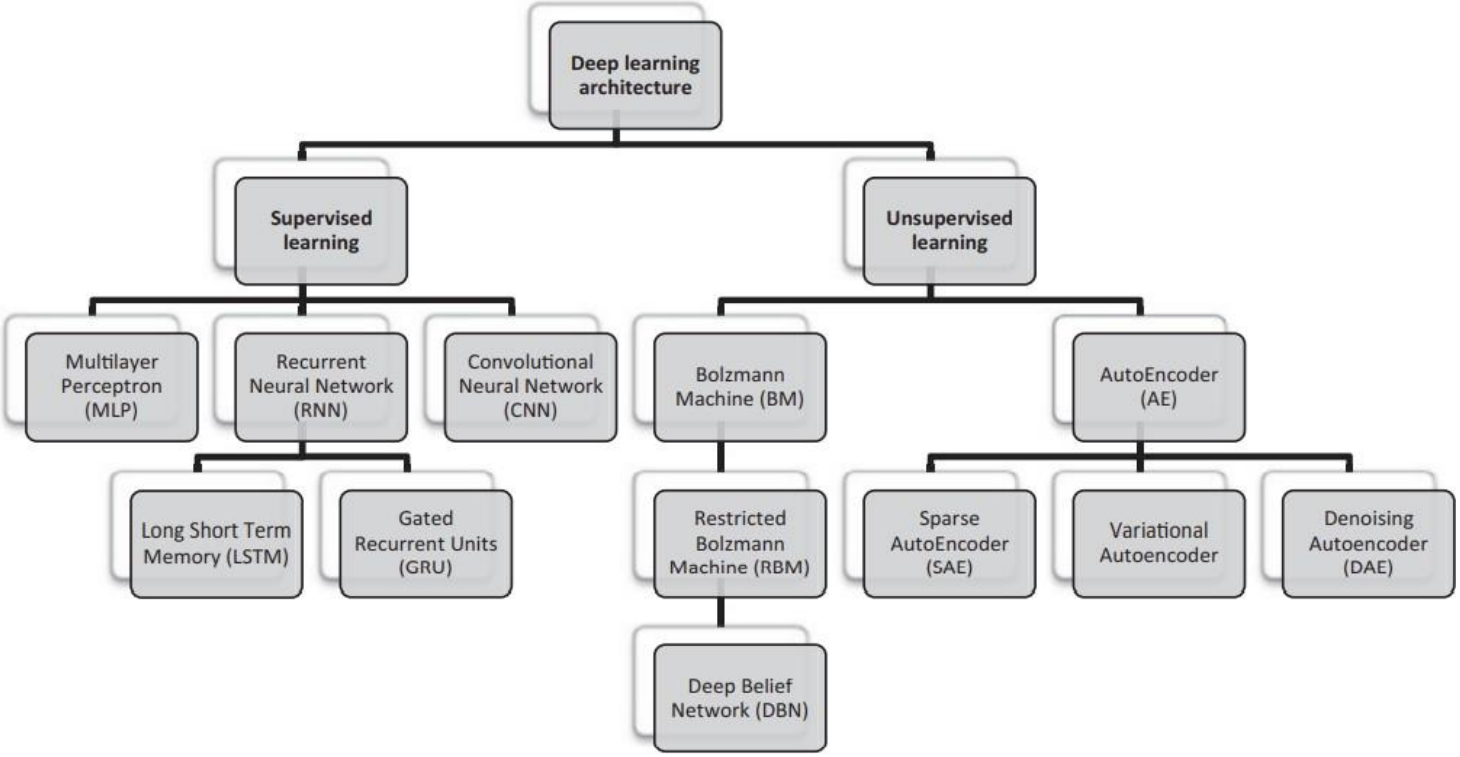
Recurrent Neural Networks (RNN)

Deep Learning Architectures

- Deep learning algorithms are categorized into supervised & unsupervised techniques

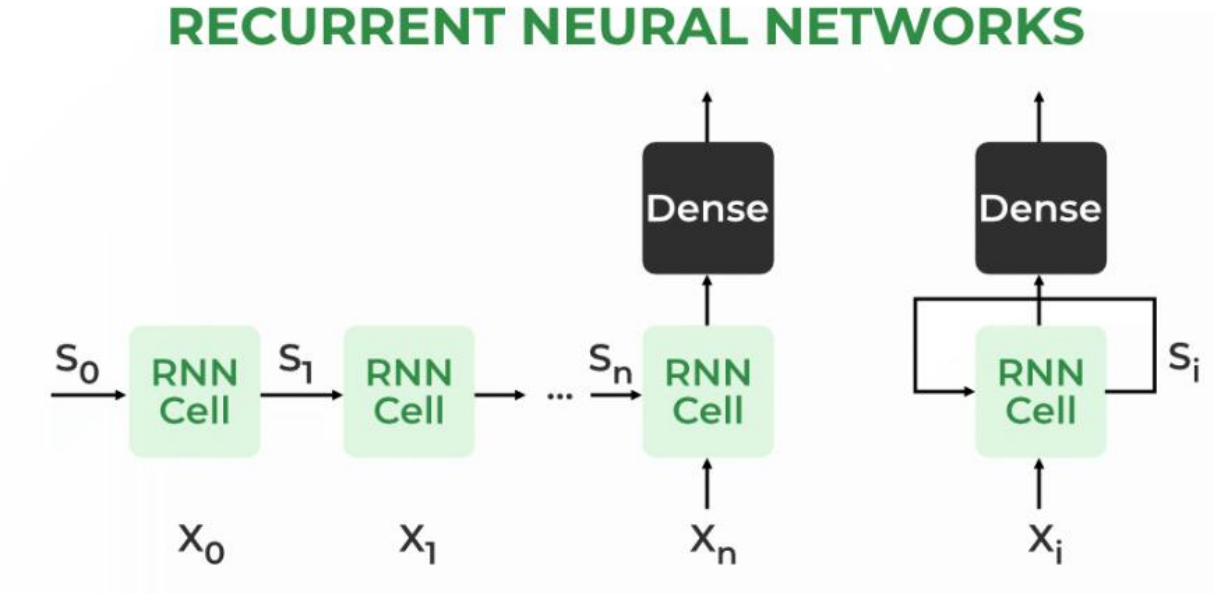


Recurrent Neural Networks

- In this session, we will introduce a new variation of neural network which is the Recurrent Neural Network also known as (RNN) that works better than a simple neural network when data is sequential like Time-Series data and text data.
- Recurrent Neural Network(RNN) is a type of Neural Network where the output from the previous step is fed as input to the current step.
- In traditional neural networks, all the inputs and outputs are independent of each other, but in cases when it is required to predict the next word of a sentence, the previous words are required and hence there is a need to remember the previous words.
- Thus RNN came into existence, which solved this issue with the help of a Hidden Layer.

Recurrent Neural Networks

- The main and most important feature of RNN is its Hidden state, which remembers some information about a sequence. The state is also referred to as Memory State since it remembers the previous input to the network. It uses the same parameters for each input as it performs the same task on all the inputs or hidden layers to produce the output. This reduces the complexity of parameters, unlike other neural networks.

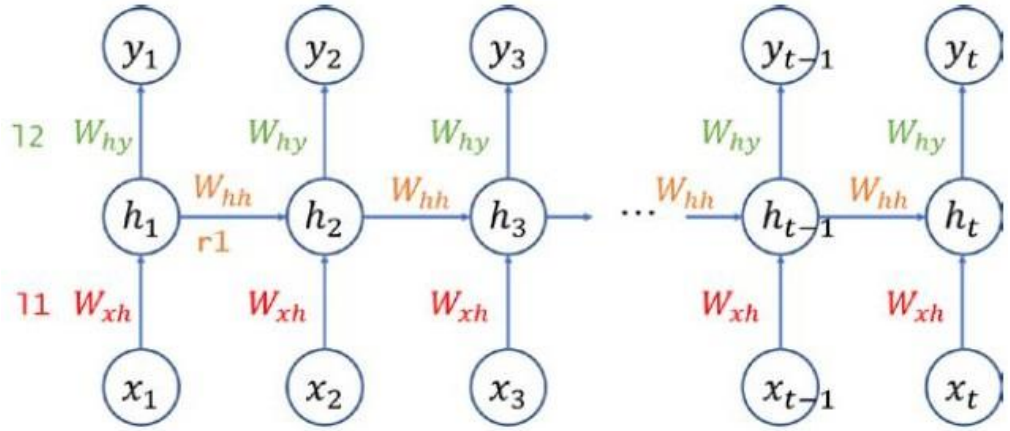


Recurrent Neural Networks

- RNN is a type of ANN that is best suited to recognizing patterns in sequences of data, e.g., text, video, speech & time-series data
- Powerful algorithm - can classify, cluster & make predictions from data
- Uses MLP network with addition of loops to the architecture, i.e., the nodes of the hidden layers are interconnected
- **Advantage:** Can use internal memory to process random inputs
- **Disadvantage:** Short memory & has scaling issues, i.e., cannot be trained for large # of inputs
- **Why use?** Can learn temporal patterns, used when data pattern changes over time

Recurrent Neural Networks

- The output depends on the sequence of data rather than a single piece of data
- i.e., output at t=n depends on the inputs at t=n, t=n-1, …, & t=1



$$h_t = \tanh(\, l1(x_t) + r1(h_{t-1})\,)$$

$$y_t = l2(h_t)$$

(Input1) → Output1

(Input2, Input1) → Output2

(Input3, Input2, Input1) → Output3

(Input4, Input3, Input2, Input1) → Output4

(Input1) → Hidden1

(Input2, Hidden1) → Hidden2

(Input3, Hidden2) → Hidden3

(Input4, Hidden3) → Hidden4

How RNN Works?

- The Recurrent Neural Network consists of multiple fixed activation function units, one for each time step. Each unit has an internal state which is called the hidden state of the unit. This hidden state signifies the past knowledge that the network currently holds at a given time step. This hidden state is updated at every time step to signify the change in the knowledge of the network about the past. The hidden state is updated using the following recurrence relation:-

The formula for calculating the current state:

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

How RNN Works?

- ht -> current state
- ht-1 -> previous state
- xt -> input state

Formula for applying Activation function(tanh):

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

- whh -> weight at recurrent neuron
- wxh -> weight at input neuron

How RNN Works?

- **Formula for the output layer:**

$$y_t = W_{hy} h_t$$

- Yt -> output
- Why -> weight at output layer

Advantages of Recurrent Neural Networks

- An RNN remembers each and every piece of information through time. It is useful in time series prediction only because of the feature to remember previous inputs as well. This is called Long Short Term Memory.
- Recurrent neural networks are even used with convolutional layers to extend the effective pixel neighborhood.

Disadvantages of Recurrent Neural Networks

- Gradient vanishing and exploding problems.
- Training an RNN is a very difficult task.
- It cannot process very long sequences if using tanh or relu as an activation function.

Applications of Recurrent Neural Networks

- Language Modeling and Generating Text
- Speech Recognition
- Machine Translation
- Image Recognition, Face detection
- Time series Forecasting

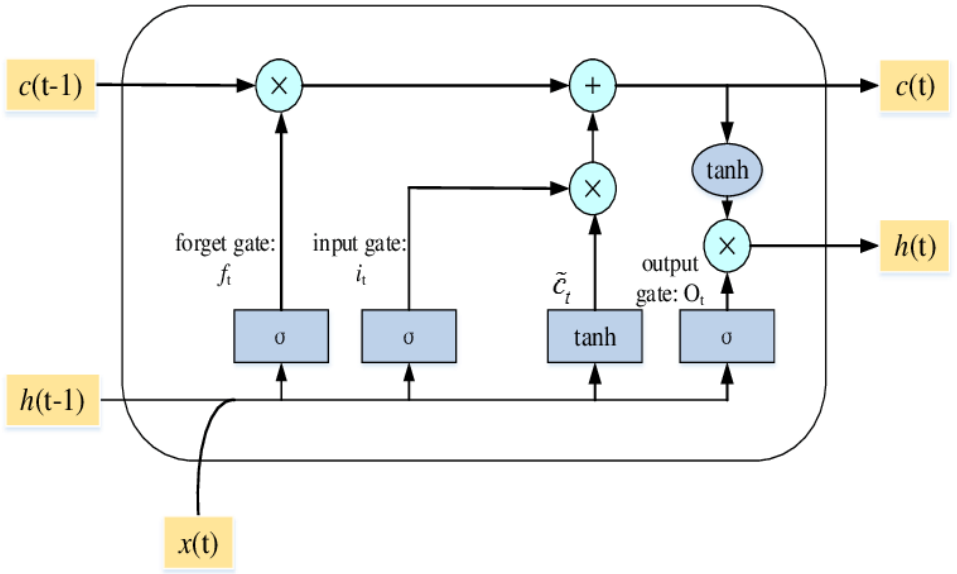
Recurrent Neural Networks

- **Gradients:**
  - Essential for tuning the weight & bias
  - Computed during backpropagation when sequences are quite long
  - Vanishing & exploding gradients cause the RNN model to train slowly
  - **Vanishing gradients:** Due to multiplication of many small values < 1
  - **Exploding gradients:** Multiplication of many large values > 1

- **LSTM:** A modified RNN architecture that **tackles the problem of vanishing & exploding gradients**

Long-Short Term Memory (LSTM)

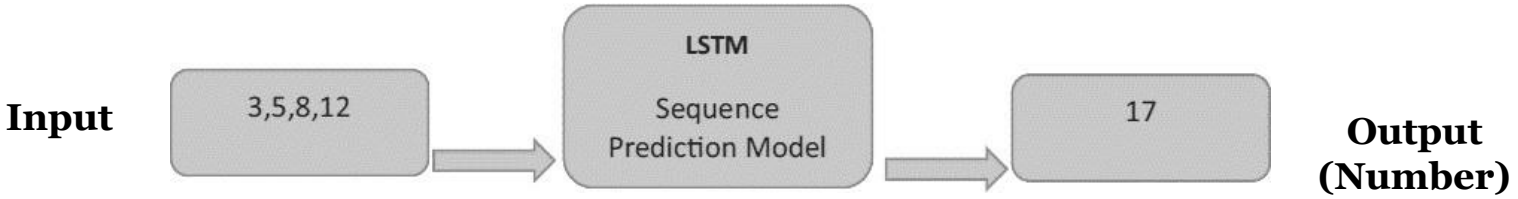
- A type of RNNs with better backpropagation
- LSTMs are modified to include a memory cell that can maintain information in memory for long periods of time
- Addresses the problem of training over long sequences & retaining memory



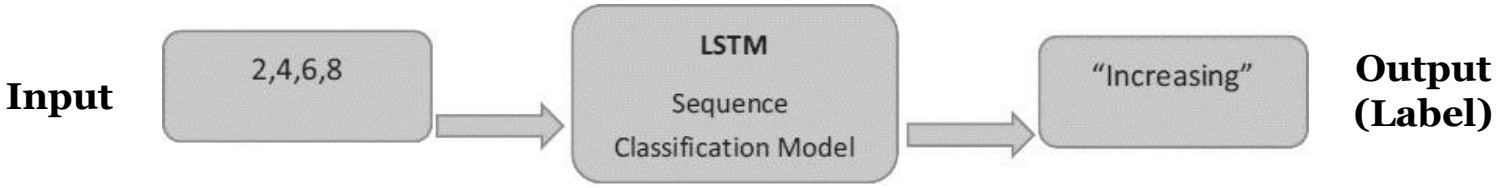
- **Gates:** Regulated structures that have the ability to remove or add information to the cell in LSTM
- **Types:** Input, Forget & Output gates

LSTM Sequence Prediction

- LSTM is best suited for sequence data
- **Types of sequence prediction:**
  1. **Sequence numeric prediction:** Involves predicting the next value for a given sequence, e.g., stock market forecasting & weather forecasting

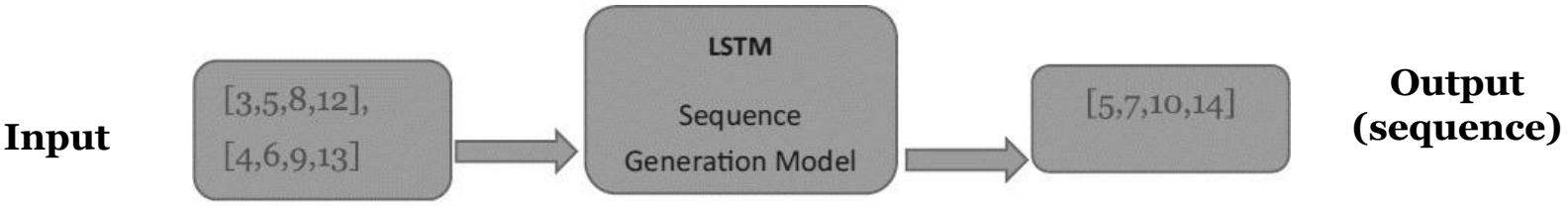


  2. **Sequence classification:** predicts the class label for a given sequence, e.g., fraud detection, student performance classification



LSTM Sequence Prediction

3. **Sequence generation:** Generate a new output sequence with the same properties as the input sequences, e.g., text generation (given 100 lines of a blog, generate the next line of the blog)



4. **Sequence-to-sequence prediction:** Predict the next sequence for a given sequence, e.g., document summarization