

[Get started](#)[Open in app](#)

## Aditi Mittal

328 Followers

[About](#)[Follow](#)

You have **1** free member-only story left this month. [Sign up for Medium and get an extra one](#)

# Understanding RNN and LSTM



Aditi Mittal · Oct 12, 2019 · 4 min read ★

## What is Neural Network?

Neural Networks are set of algorithms which closely resemble the human brain and are designed to recognize patterns. They interpret sensory data through a machine perception, labelling or clustering raw input. They can recognize numerical patterns, contained in vectors, into which all real-world data ( images, sound, text or time series), must be translated. Artificial neural networks are composed of a large number of highly interconnected processing elements (neuron) working together to solve a problem.

An ANN usually involves a large number of processors operating in parallel and arranged in tiers. The first tier receives the raw input information — analogous to optic nerves in human visual processing. Each successive tier receives the output from the tier preceding it, rather than from the raw input — in the same way neurons further from the optic nerve receive signals from those closer to it. The last tier produces the output of the system..

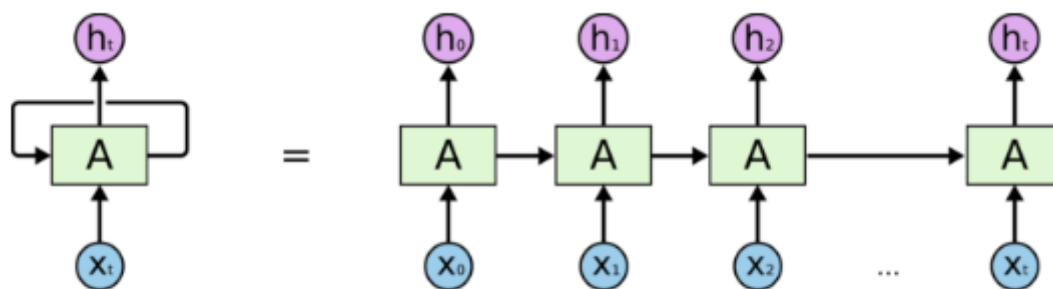
## What is Recurrent Neural Network (RNN)?

Recurrent Neural Network is a generalization of feedforward neural network that has an internal memory. RNN is recurrent in nature as it performs the same function for every input of data while the output of the current input depends on the past one



has learned from the previous input.

Unlike feedforward neural networks, RNNs can use their internal state (memory) to process sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition. In other neural networks, all the inputs are independent of each other. But in RNN, all the inputs are related to each other.



**An unrolled recurrent neural network.**

First, it takes the  $X(0)$  from the sequence of input and then it outputs  $h(0)$  which together with  $X(1)$  is the input for the next step. So, the  $h(0)$  and  $X(1)$  is the input for the next step. Similarly,  $h(1)$  from the next is the input with  $X(2)$  for the next step and so on. This way, it keeps remembering the context while training.

The formula for the current state is

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

**Applying Activation Function:**

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

**W** is weight, **h** is the single hidden vector, **W<sub>hh</sub>** is the weight at previous hidden state, **W<sub>xh</sub>** is the weight at current input state, **tanh** is the Activation function, that implements a Non-linearity that squashes the activations to the range  $[-1, 1]$



$$y_t = W_{hy}h_t$$

$y_t$  is the *output state*.  $W_{hy}$  is the *weight at the output state*.

## Advantages of Recurrent Neural Network

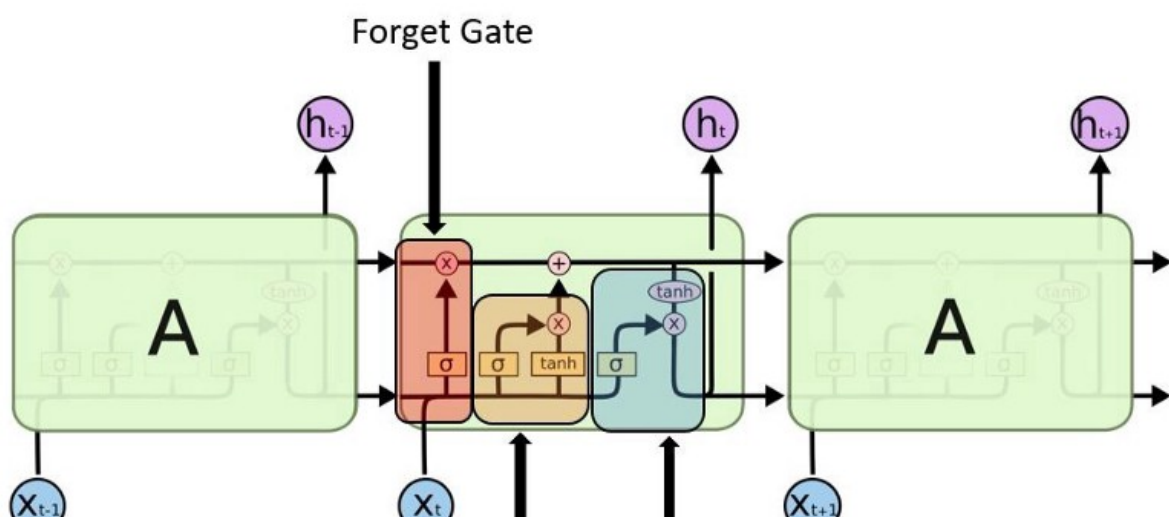
1. RNN can model sequence of data so that each sample can be assumed to be dependent on previous ones
2. Recurrent neural network are even used with convolutional layers to extend the effective pixel neighbourhood.

## Disadvantages of Recurrent Neural Network

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

## What is Long Short Term Memory (LSTM)?

Long Short-Term Memory (LSTM) networks are a modified version of recurrent neural networks, which makes it easier to remember past data in memory. The vanishing gradient problem of RNN is resolved here. LSTM is well-suited to classify, process and predict time series given time lags of unknown duration. It trains the model by using back-propagation. In an LSTM network, three gates are present:





## Input Gate    Output Gate

### LSTM gates

1. **Input gate** — discover which value from input should be used to modify the memory. **Sigmoid** function decides which values to let through **0,1**. and **tanh** function gives weightage to the values which are passed deciding their level of importance ranging from **-1** to **1**.

$$i_t = \sigma (W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

Input gate

2. **Forget gate** — discover what details to be discarded from the block. It is decided by the **sigmoid function**. it looks at the previous state(**ht-1**) and the content input(**Xt**) and outputs a number between **0(omit this)** and **1(keep this)** for each number in the cell state **Ct-1**.

$$f_t = \sigma (W_f \cdot [h_{t-1}, x_t] + b_f)$$

Forget gate

3. **Output gate** — the input and the memory of the block is used to decide the output. **Sigmoid** function decides which values to let through **0,1**. and **tanh** function gives weightage to the values which are passed deciding their level of importance ranging from **-1** to **1** and multiplied with output of **Sigmoid**.

Get started

Open in app



$$h_t = o_t * \tanh(C_t)$$

Output gate

*Thanks for reading! You can checkout my other articles:*

- [Introduction to U-Net and Res-Net for Image Segmentation](#)
- [Instance Segmentation Using Mask R-CNN](#)

Machine Learning

Rnn

Lstm

[About](#) [Write](#) [Help](#) [Legal](#)

Get the Medium app

