Title 1: Predictive Healthcare Chatbot: AI's Role in Forecasting Infectious Diseases

Title 2: Infection Forecasting with AI Chatbot: A Medical Breakthrough

Title 3: Infectious Disease Prophecy: AI Chatbot for Proactive Medical Insights

Title 4: Forecasting Infections: An AI-Powered Medical Chatbot Approach

Title 5: Proactive Infection Prediction: The AI Medical Chatbot Paradigm

***An AI-Powered Medical Chatbot using NLP for Covid-19***

**BACKGROUNG WORK**

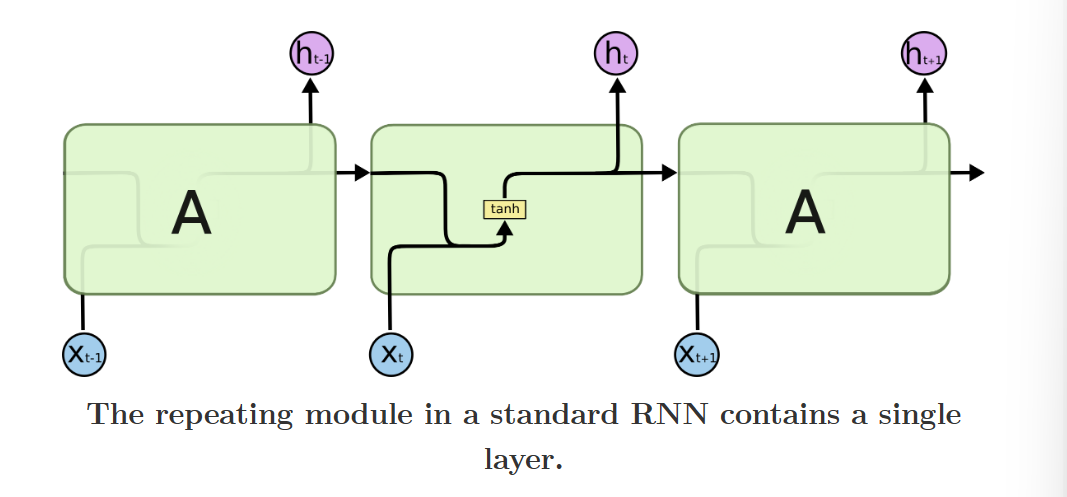
**Understanding LSTM Networks:**

Ref: [Understanding LSTM Networks -- colah's blog](http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

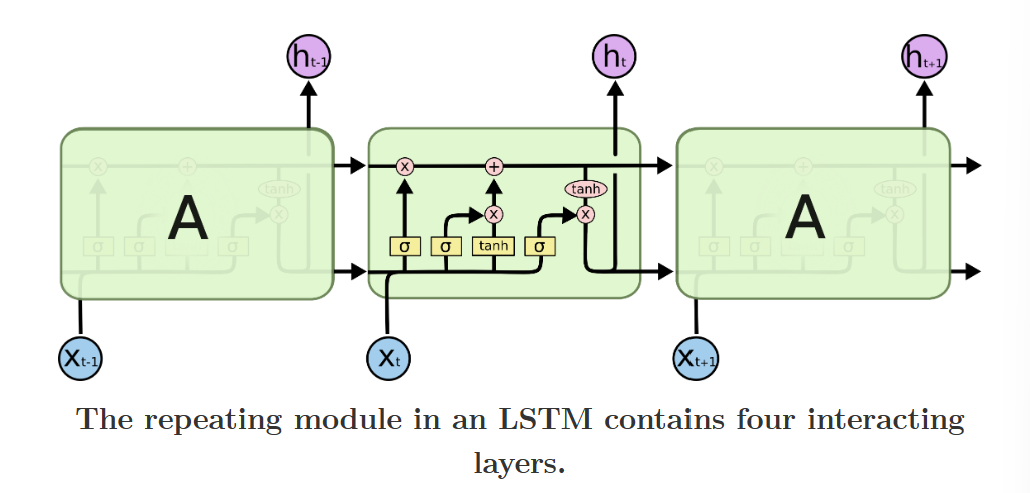
Long Short Term Memory networks – usually just called “LSTMs” – are a special kind of RNN, capable of learning long-term dependencies. They were introduced by Hochreiter & Schmidhuber (1997), and were refined and popularized by many people in following work.1 They work tremendously well on a large variety of problems, and are now widely used.

LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn!

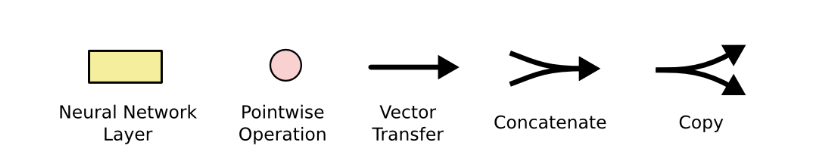
All recurrent neural networks have the form of a chain of repeating modules of neural network. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer.



LSTMs also have this chain like structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four, interacting in a very special way.



Don’t worry about the details of what’s going on. We’ll walk through the LSTM diagram step by step later. For now, let’s just try to get comfortable with the notation we’ll be using.

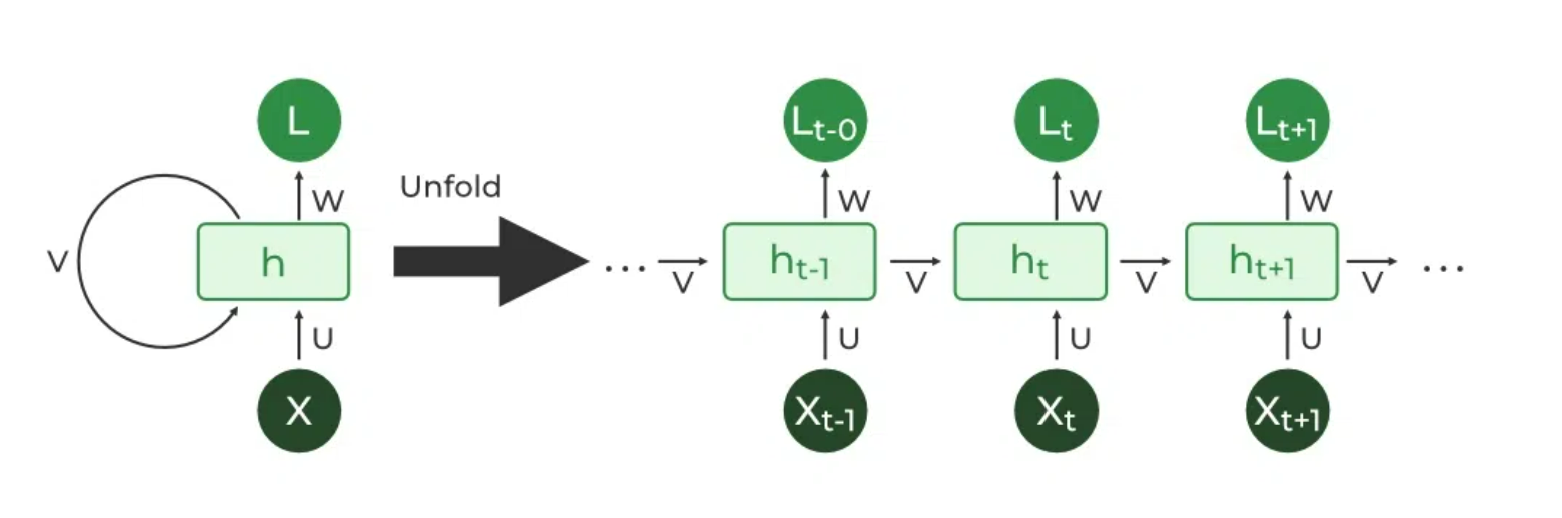


In the above diagram, each line carries an entire vector, from the output of one node to the inputs of others. The pink circles represent pointwise operations, like vector addition, while the yellow boxes are learned neural network layers. Lines merging denote concatenation, while a line forking denote its content being copied and the copies going to different locations.

**Recurrent Neural Network (RNN):**

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. 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 network :*



Architecture Of Recurrent Neural Network

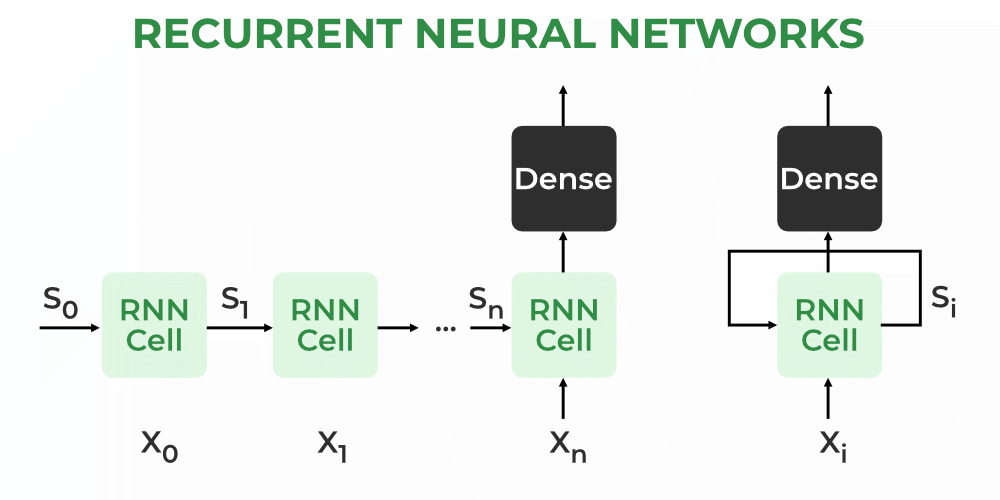
RNNs have the same input and output architecture as any other deep neural architecture. However, differences arise in the way information flows from input to output. Unlike Deep neural networks where we have different weight matrices for each Dense network in RNN, the weight across the network remains the same. It calculates state hidden state Hi for every input Xi . By using the following formulas:

***h= σ(UX + Wh-1 + B)***

***Y = O(Vh + C) Hence***

***Y = f (X, h , W, U, V, B, C)***

*Here S is the State matrix which has element si as the state of the network at timestep i  
The parameters in the network are W, U, V, c, b which are shared across timestep*



Working of RNN:

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:**



where:

ht -> current state

ht-1 -> previous state

xt -> input state

**Formula for applying Activation function(tanh):**

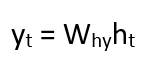


where:

whh -> weight at recurrent neuron

wxh -> weight at input neuron

**The formula for calculating output:**



Yt -> output

Why -> weight at output layer

These parameters are updated using Backpropagation. However, since RNN works on sequential data here we use an updated backpropagation which is known as Backpropagation through time.

**Training through RNN:**

1. A single-time step of the input is provided to the network.
2. Then calculate its current state using a set of current input and the previous state.
3. The current ht becomes ht-1 for the next time step.
4. One can go as many time steps according to the problem and join the information from all the previous states.
5. Once all the time steps are completed the final current state is used to calculate the output.
6. The output is then compared to the actual output i.e the target output and the error is generated.
7. The error is then back-propagated to the network to update the weights and hence the network (RNN) is trained using Backpropagation through time.

**Advantages of Recurrent Neural Network:**

1. 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.
2. Recurrent neural networks are even used with convolutional layers to extend the effective pixel neighborhood.

**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.

**Applications of Recurrent Neural Network:**

1. Language Modelling and Generating Text
2. Speech Recognition
3. Machine Translation
4. Image Recognition, Face detection
5. Time series Forecasting

**Types Of RNN:**

There are four types of RNNs based on the number of inputs and outputs in the network.

1. One to One
2. One to Many
3. Many to One
4. Many to Many

**Decision Tree:**

A decision tree is a flowchart-like [tree structure](https://www.geeksforgeeks.org/introduction-to-tree-data-structure-and-algorithm-tutorials/) where each internal node denotes the feature, branches denote the rules and the leaf nodes denote the result of the algorithm. It is a versatile [supervised machine-learning](https://www.geeksforgeeks.org/ml-types-learning-supervised-learning/) algorithm, which is used for both classification and regression problems. It is one of the very powerful algorithms. And it is also used in Random Forest to train on different subsets of training data, which makes random forest one of the most powerful algorithms in [machine learning](https://www.geeksforgeeks.org/machine-learning/).

Working on Dialog Flow instead of Tkinter library: