RECURRENT NEURAL NETWORKS

* RNN’s are special kind of Neural Networks that were specifically designed to work with sequential data.
* They designed to overcome the limitations of ANN’s and CNN’s on sequential type of data.
* Example of Sequential Data:

1. Text:

“Hi, my name is Shreyas”

In the above sentence each word possess some meaning and by combining them we get a meaningful sentence. If we change the order of the words in which they appear, the meaning of the sentence is lost.

1. Time series:

It may be a graph depicting the stock prices of a company over the years, where we can observe how the price changes over the years.

LAYERS IN RNN

* RNN are quite similar to ANN in terms of its architecture, but they differ in how both the models work to produce an output.

1. Input layer:  
   -As ANN, RNN also contains an input layer where we will feed the model some sequential data, but in terms of ‘timesteps’. It can be given as (timesteps, input\_features)

-Timesteps represent the element in the data or its position in the data.

-input\_features represents total number of features in the data.

Example:

‘I love RNN’

* I want to load the above data as input in my RNN model.
* Since it’s a sequential data, we will pass each element in the data one by one and the model will then process it.
* So, if I send the first element ‘I’ into the model then its timestep can be given as t=1.

1. Hidden Layer:

-RNN also contains a hidden layer, but this layer works very differently from the one in ANN.

-Here the output obtained after each timestep is taken as input along with the new input for the current timestep.

1. Output Layer:

* This layer gives us the output after each timestep, which means we will get Ot after each timestep ‘t’.

WHY USE RNN INSTEAD OF ANN OR CNN

-RNN’s are best suited for Sequential Data, but what if we use ANN instead of RNN.

For eg:

We take three Inputs(X) and predict their output(Y)

|  |  |
| --- | --- |
| Input(X) | Output(Y) |
| My name is Shreyas | 1 |
| Yesterday was Sunday | 0 |
| I am pursuing my Masters | 0 |

We load this data in an ANN model and predict its output,

-Firstly, we cannot load this data as it is, we’ll need to convert it into numbers or vectors.

-If we calculate, our data contains 12 unique words. So each word in the data can be represented in terms of 12 numbered vectors as,

My - [1,0,0,0,0….]

Name – [0,1,0,0,0…]

Is – [0,0,1,0,0…]

Shreyas – [0,0,0,1,0,…]

-The remaining words can also be represented in the same format

-Then we load the data in our ANN model (4 units), suppose we load our first sentence (X1) its input size will be 12\*4 = 48 and the weights will be 12\*4\*4 = 176.

-After successfully loading our 1st sentence in the ANN model, you may notice a problem that the length of our 2nd and 3rd sentences vary from the 1st.

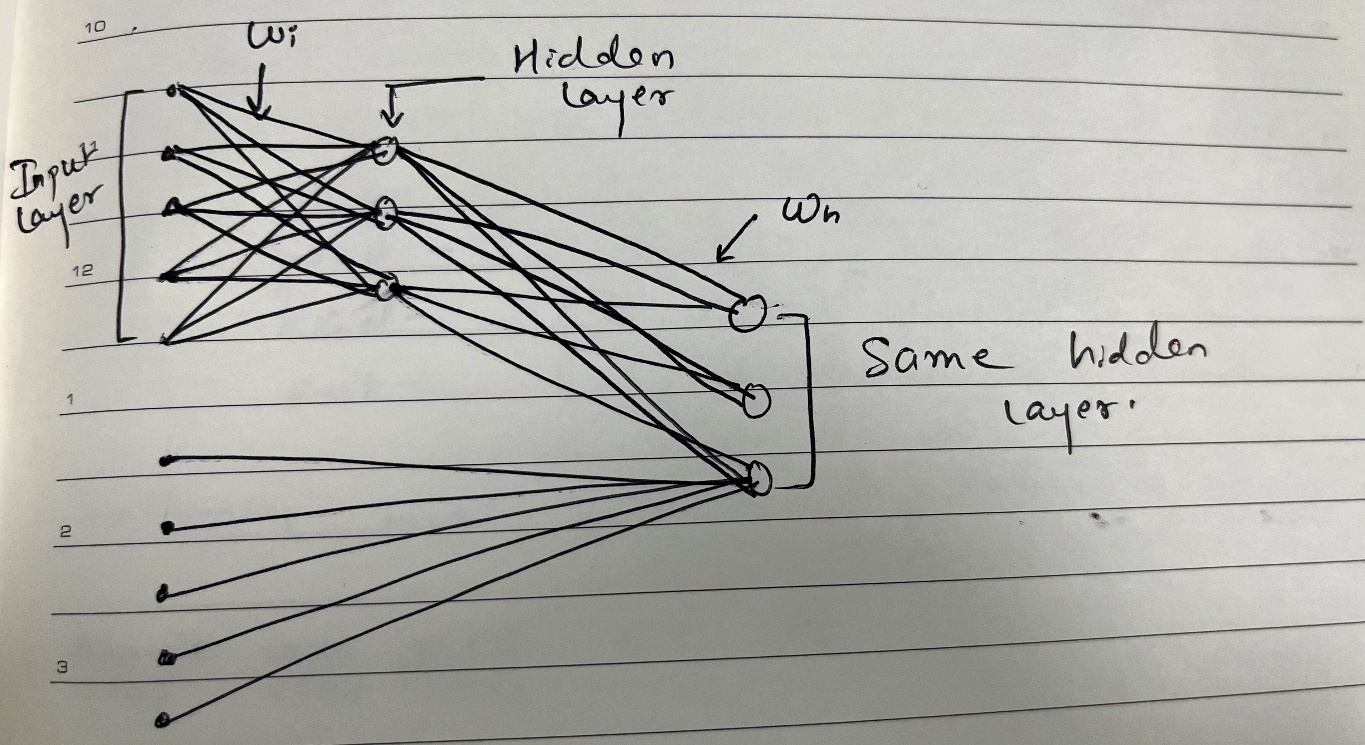
-So their input sizes also varies for X2 its 12\*3 = 36 and for X3 its 12\*5 = 60.

-Each input size of the data varies from the other and ANN’s cannot deal with inputs with varying sizes. This is the biggest limitation of using ANN on sequential data.

-Just like ANN’s , CNN’s also cannot preserve the order of the data, CNN’s treat data spatially and applies filters on the data simultaneously and not sequentially.

- Along with ANN’s, CNN’s don’t have the ability to handle or deal with varying input\_sizes.

ARCHITECTURE OF RNN



WORKING OF RNN

Lets take an example of Sentiment Analysis

|  |  |
| --- | --- |
| Review | Sentiment |
| Movie was good | 1 |
| Movie was bad | 0 |
| Movie was not good | 0 |

-For RNN to take input we transform the textual data in vector format using OneHotEncoder. Since our data contains 5 unique words , so a five numbered vector will be formed , which will be given as

Movie – [1,0,0,0,0]

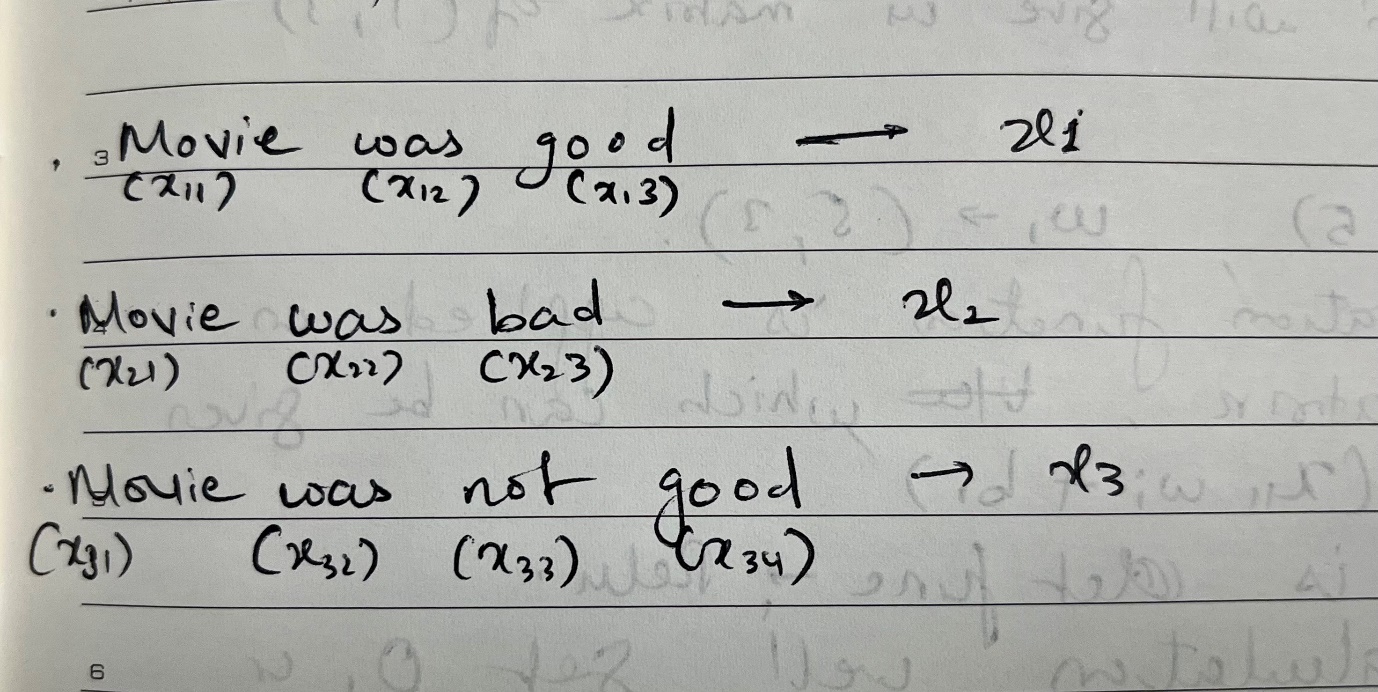
was – [0,1,0,0,0]

good – [0,0,1,0,0]

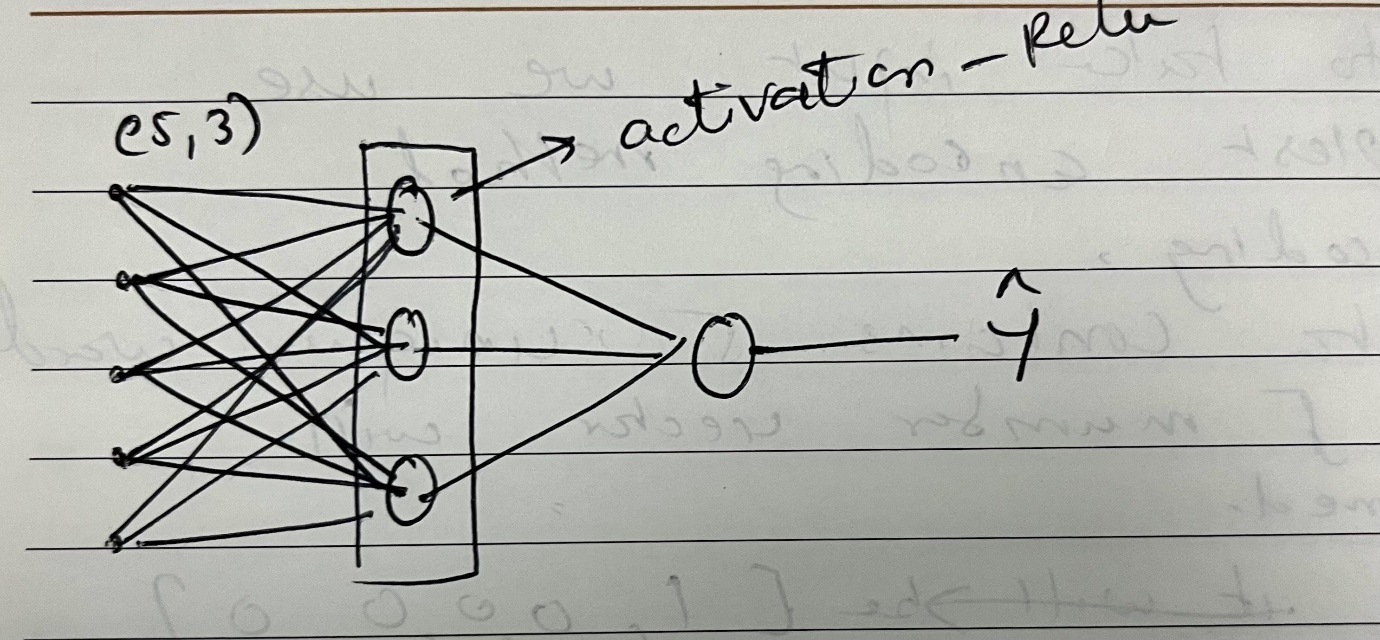
bad – [0,0,0,1,0]

not – [0,0,0,0,1]

-Following are some notations,



-Now we load the input data in our RNN model.



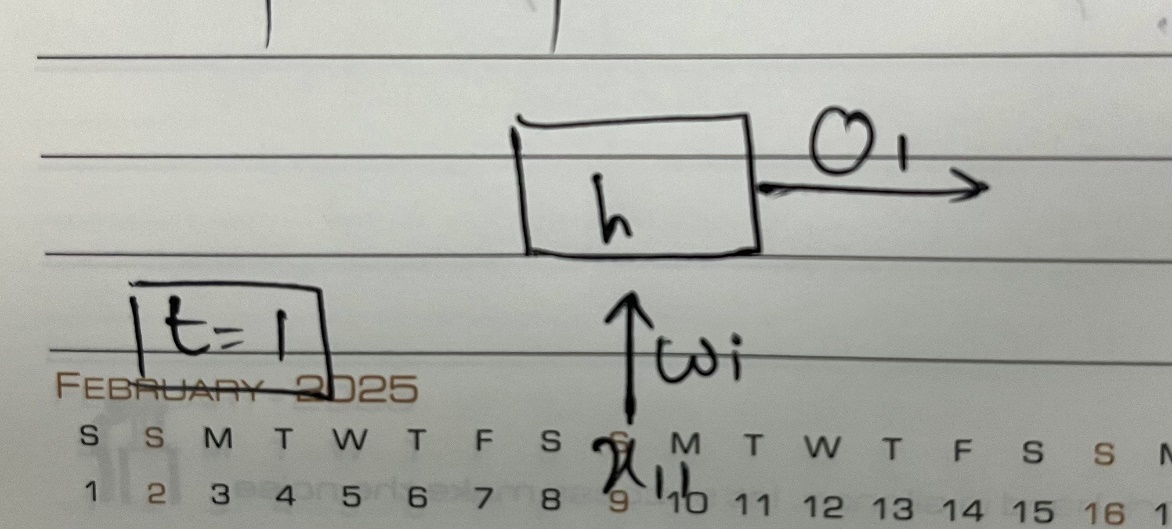
-At t=1, we pass x11 i.e the first word from review1, it pass to the hidden layer through weight (wi).

-A dot product of x11wi will be formed which will give us matrix of (1,3), since x11 – (1,5) and wi – (5.3)

-An Activation function (Relu) is applied on the (1,3) matrix , which can be given as f(x11wi + b1). Here f is the activation function.

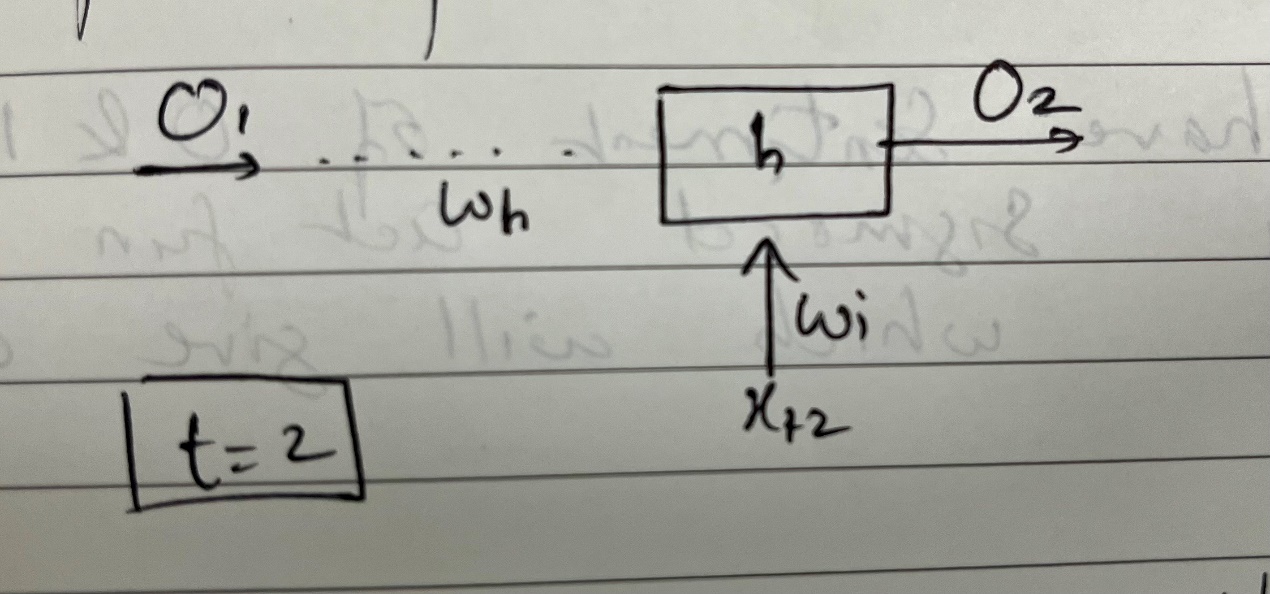
-After this calculation we will get output O1.

Simple Representation:



-At t=2, we pass x12 through weight (wi). Along with x12 ,O1 is also taken as an input and here O1 will pass through a weighted output (wh).

-Here calculation will be carried out this time at t=2 it will be given as f(x12wi+O1wh+b1) and we’ll get O2.

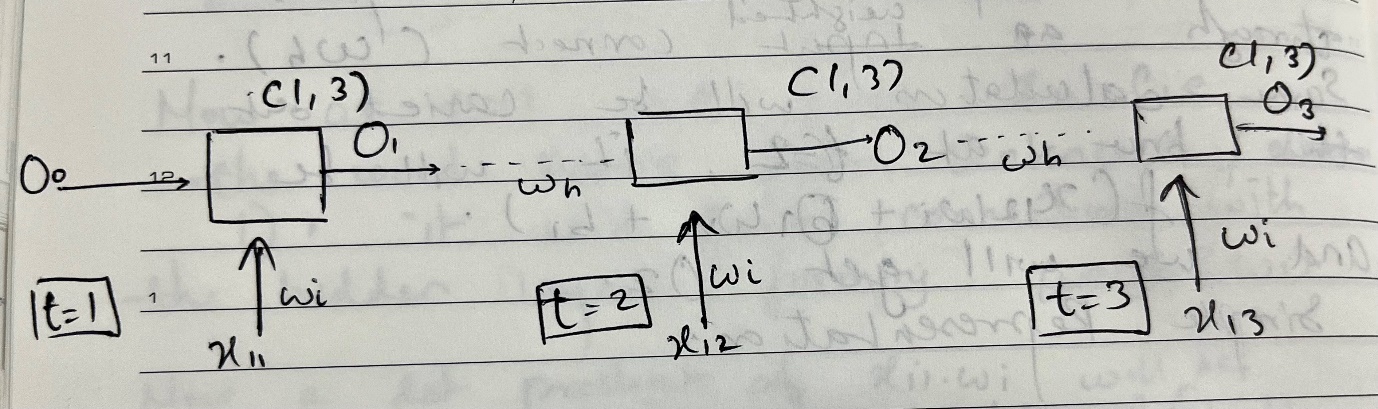


-Similarly, at t=3 we use the same weights to x13 and a weighted connection to pass O2 this time. So we will get f(x13wi+O2wh+b1)

After calculations we get O3.

-And with this we reach end of Review1 which consisted of three words.

-So, the whole RNN working can be given as ,



-Since we have to predict the sentiment 0 or 1, we apply sigmoid activation function at y(cap) which will be our final output