# RNN Theory

Suppose I have a sequential data, , how can program a code to learn the pattern in the data?

We can use a Timestep of the Data, of length , to predict the next values in the dataset

There should also be a “memory element” to the equation, such that the previous data in the sequence affects the future predictions. Let us call that the hidden vector .

In some way, this “memory element” should also store some information about the prediction . So we can compute as some linear combination of .

# Basic Flow

Using an arbitrary activation function , we can construct a simple flow as follows:

Some Linear combination

Some Linear combination

# Equations

With this flow in mind, we can construct the recursive equation for the hidden vector and the prediction as such:

Where we define the following parameters and vectors



Dimensional Parameters

Activation Function: Sigma function

# RNN Architecture

…

**Sequential Data**

**Timestep Data**

**Hidden Layer**

**Prediction**



# Back Propagation Through Time (BPTT)

Based on the prediction , we can compute the loss function from the actual values of   
 as follows:

In finding the partial derivatives, it is worth noting the derivative of the sigma function can be simplifies as:

From the loss function, we can thus compute the following partial derivatives of the weighting parameters, , , :

# Algorithm

Initialize Parameters

Iterate for N epochs

From t = 1 to n-T+1 :

Get

Find hidden vector from previous hidden vector

Compute predictions

Compute Partial Derivatives and update Parameters