Deep Sequence Modelling

B.Tech. Data Science, NMIMS
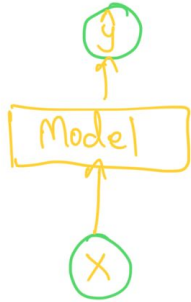
By,

Bilal Hungund, Data Scientist, Halliburton
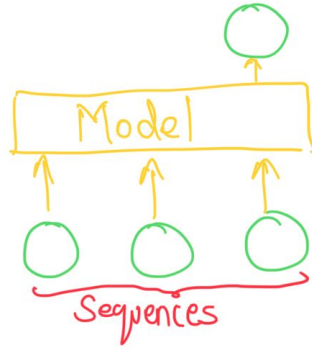
Inconsistent efforts

Consistent efforts
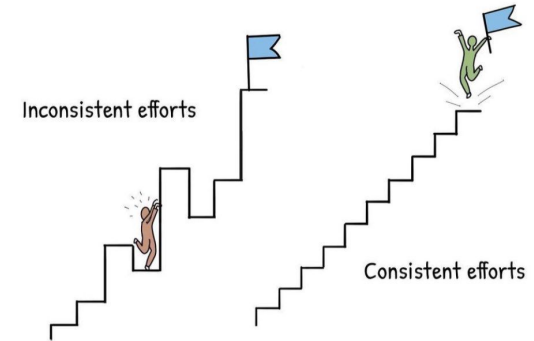
Sequencing Modelling Applications

One to one
(Classification)
Regression

$$y$$
Model
$$X$$

Many to One
(Sentimental Analysis)

Model

Sequences

One to Many
(Image Captioning)
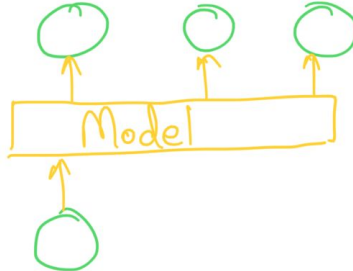
Model

Cat is sleeping

Many to Many
(Language Modelling)
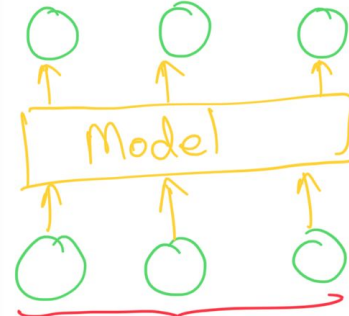
Model

Temporal Space

English to Hindi
Text

Perceptrons Revisited

Feed Forward Models

1

w_0

x_1 w_1

z $\xrightarrow{g(z)}$ $\hat{y}$

x_2 w_2

x_m w_3

(No notion of time sequence yet)

x_1

x_2

x_m

$\hat{y}_1$

$\hat{y}_2$

$\hat{y}_n$

$x \in \mathbb{R}^m$

$\hat{y} \in \mathbb{R}^n$
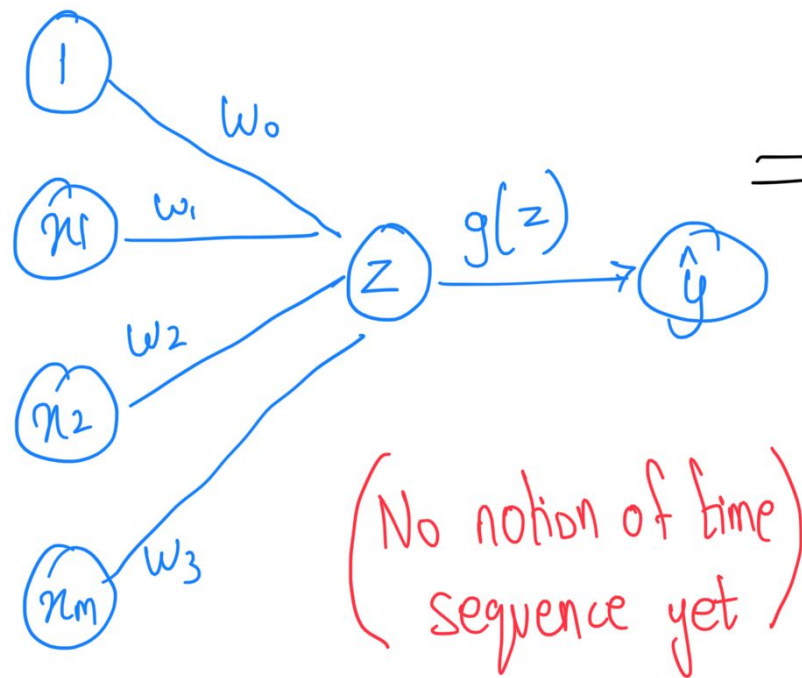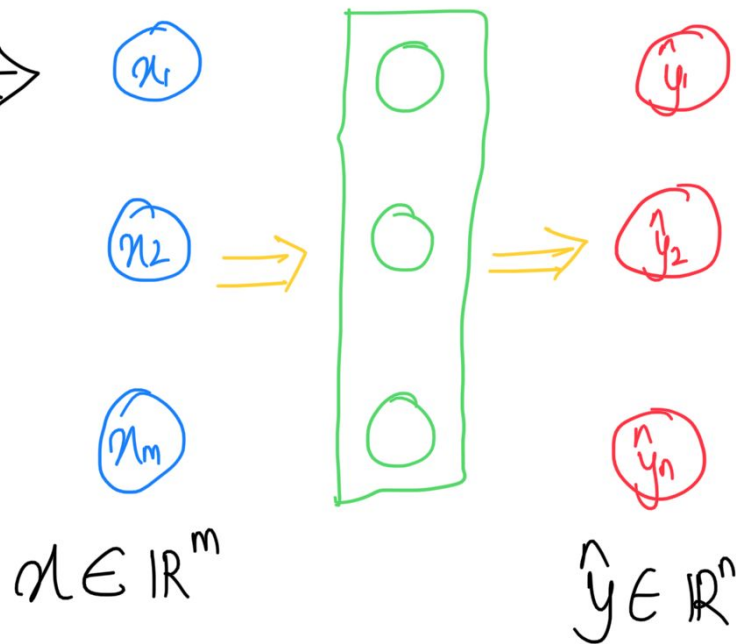
Handling with time steps

(How to relate network computations?)
Need some prior history

Output Vector: $\hat{y}_t$

Input Vector: x_t

Splitting t

$\hat{y}_0$ $\hat{y}_1$ $\hat{y}_t$

x_0 x_1 ..... x_t

$$\hat{y}_t = f(x_t)$$
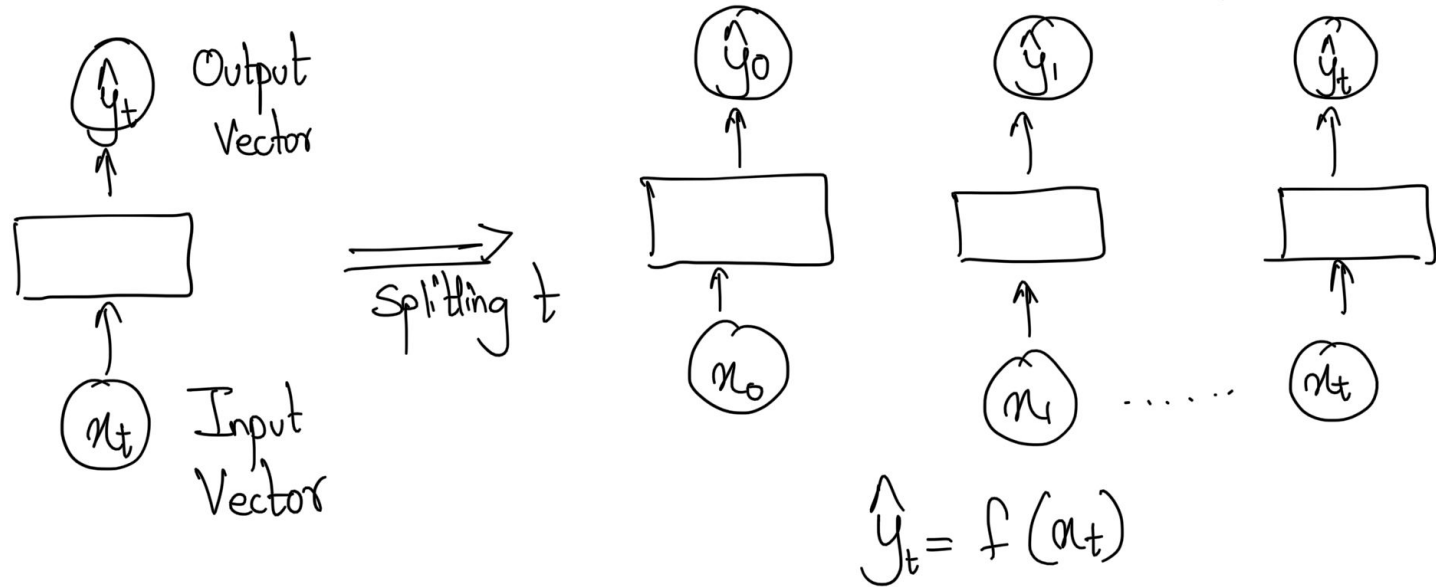
No interdependence or interconnectedness added yet

Neurons with Recurrence

Trainable weights

Linking the information and Computation of network

$\hat{y}_t$

$\hat{y}_0$ $\hat{y}_1$ $\hat{y}_2$ $\hat{y}_b$

Recurrent Cell

$h_t \Rightarrow$

h_1 h_2 h_3

x_t

x_0 x_1 x_2 x_t

Recurrence relation
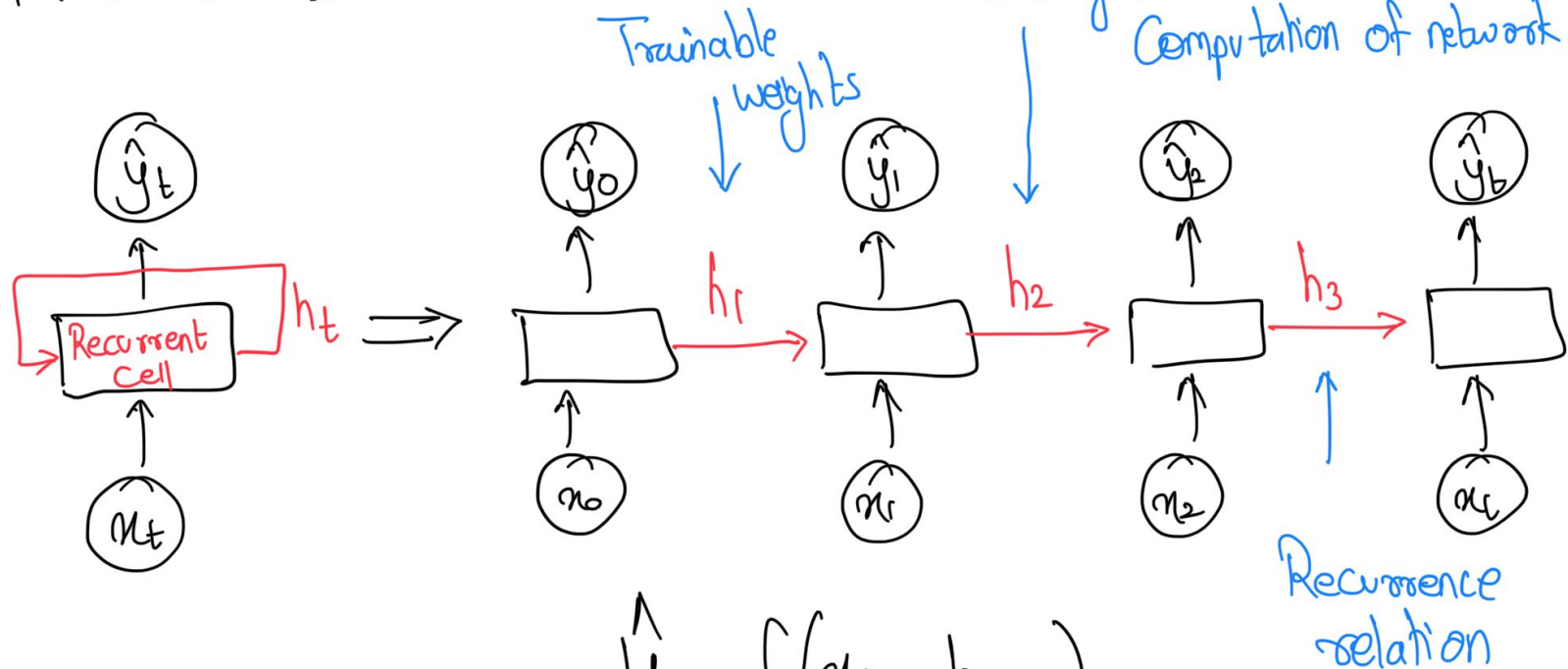
$$\hat{y} = f(x_t, h_{t-1})$$

Output   Input   past Memory

Recurrence Neural Networks (RNNs)

$\longrightarrow$ RNNs have a state h_t, that is updated at each time steps as a sequence

i.e.,

$$h_t = f_w \left(\alpha_t , h_{t-1} \right)$$

cell state   weight   inputs   old state

Input Vector	Update Hidden State	Output Vector
α_t	$h_t = \tanh \left(W_{hh}^T h_{t-1} + W_{\alpha h}^T \alpha_t \right)$	$\hat{y}_t = W_{hy}^T h_t$

Simplifying RNNs

Losses

$$L$$

Backpropagation
through
time

$$L_0 \qquad L_1 \qquad L_2 \qquad L_3$$

$\hat{y}_t$

$\hat{y}_0 \qquad \hat{y}_1 \qquad \hat{y}_2 \qquad \hat{y}_t$

$$\boxed{RNN} \quad = \quad h_0 \xrightarrow{} \boxed{} \xrightarrow{W_{hh}} \boxed{} \xrightarrow{W_{hh}} \boxed{} \xrightarrow{W_{hh}} \boxed{}$$

W_{hy} W_{hy} W_{hy} W_{hy}

$h_1 \qquad h_2 \qquad h_3$

W_{xh} W_{xh} W_{xh} W_{xh}

x_t

$x_0 \qquad x_1 \qquad x_2 \qquad x_t$

- - -> Backward Pass   ——> Forward pass
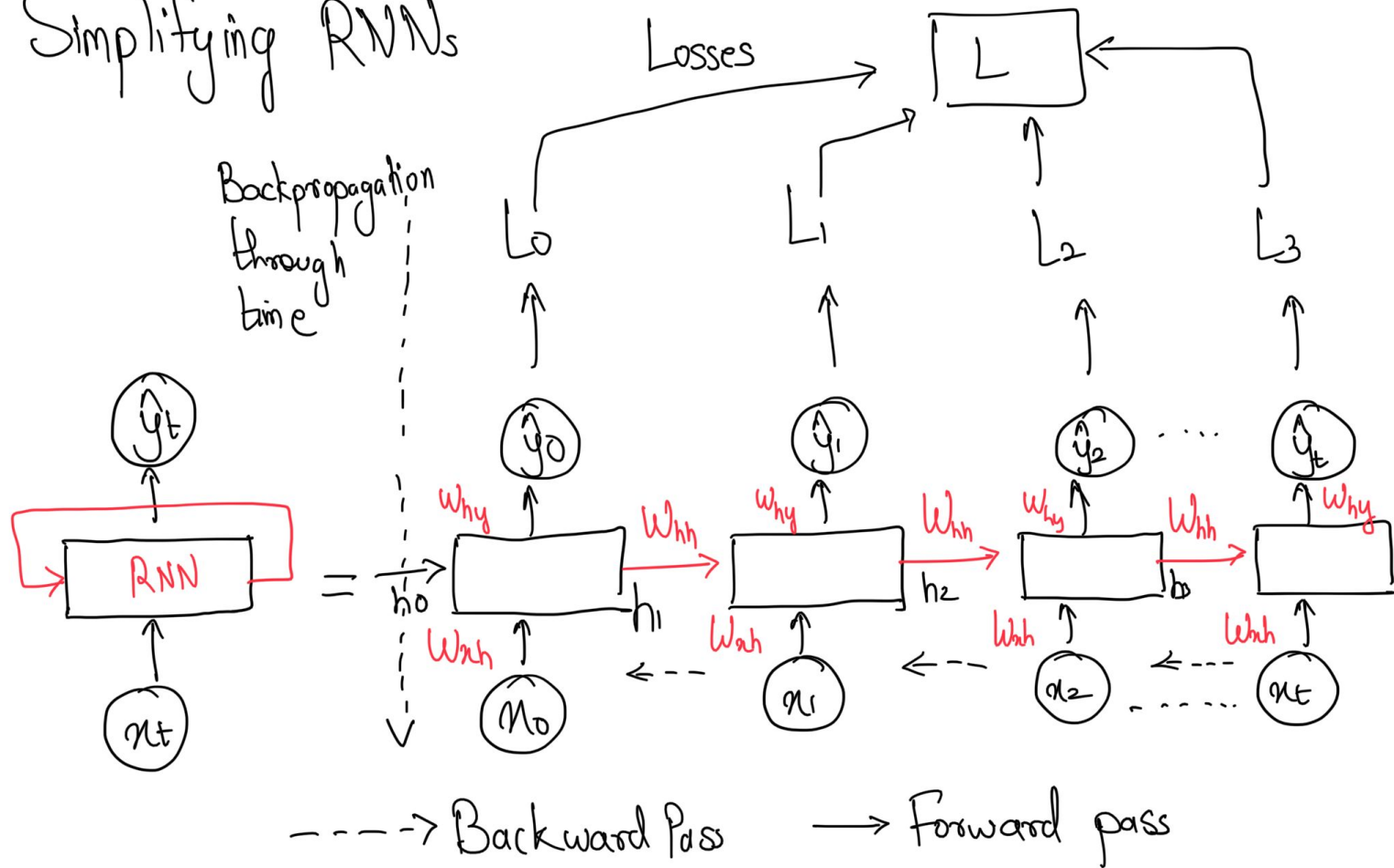
Gradient issues
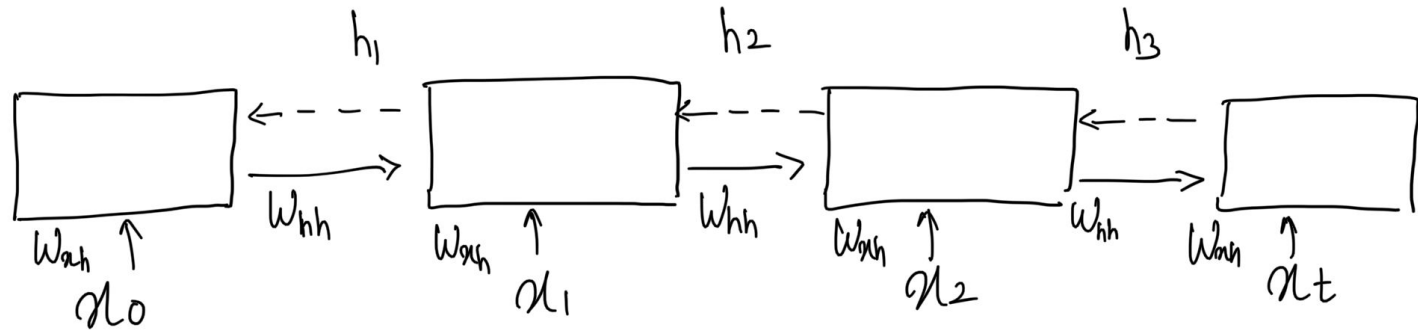
$\longrightarrow$ Exploding gradients (values > 1)

$\longrightarrow$ Gradient Clipping : Scale big gradients

$\longrightarrow$ Vanishing gradients (values < 1)

$\longrightarrow$ Activation Function, Weight initialization, Network architecture

Vanishing gradients

$\longrightarrow$ Focusing on short-term dependencies and ignoring long term dependencies

Short term dependencies

I love neural $\underline{\quad ? \quad}$
(network)

$\xrightarrow{\text{Ignorance}}$

I studied data science and
I am fluent in $\underline{\quad ? \quad}$
(ML, DL, ...)

To alleviate the problem

$\longrightarrow$ Use of ReLU function as it prevents shrinking the gradients when $x > 0$

$\longrightarrow$ Weight Initialization:

Initialize bias to zero and weights to identity matrix

$\underline{\text{It}}$ prevents the weights from shrinking to zero

$$I_n = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

$\longrightarrow$ Gated Cells : (LSTM)

Network Architecture

Use gates to add or remove information within each recurrent unit and to track information

Design Criteria of RNNs

- Handle variable-length sequences
- Track long term dependencies
- Maintain order of information
- Sharing parameters

Goals of Sequence Modelling

- Continuous stream
- Parallelization
- Long Memory

Limitations of RNNs

- Encoding bottleneck
- Slow, no parallelization
- Not long memory

NLP using RNN

Tokenization , pad Sequences and Embeddings

F L O W
001 002 003 004
Character Tokens

W O L F
004 003 002 001

Corpus/vocabulary

I   love   neural   network
001   002   003   004

I   love   deep   learning
001   002   005   006

Sequences
[[001, 002, 003, 004],
 [001, 002, 005, 006]]'

Tokenization , pad Sequences and Embeddings

building vocabulary

I love neural network
I love deep learning
You love neural network
Do you think deep learning is good?

Tokens →

love-1, i-2, you-3,
neural-4, network-5,
deep-6, learning-7,
do-8, think-9, good-10
is-11

Sequences

[2 1 4 5]
[2 1 6 7]
[3 1 4 5]
[8 3 9 6 7 11 10]

pad Sequences

$\begin{bmatrix} 0 & 0 & 0 & 2 & 1 & 4 & 5 \\ 0 & 0 & 0 & 2 & 1 & 6 & 7 \\ 0 & 0 & 0 & 3 & 1 & 4 & 5 \\ 8 & 3 & 9 & 6 & 7 & 11 & 10 \end{bmatrix}$

Pass to Embeddings