

# Recurrent Neural Network Model

Partha Ghosh

August 24, 2020

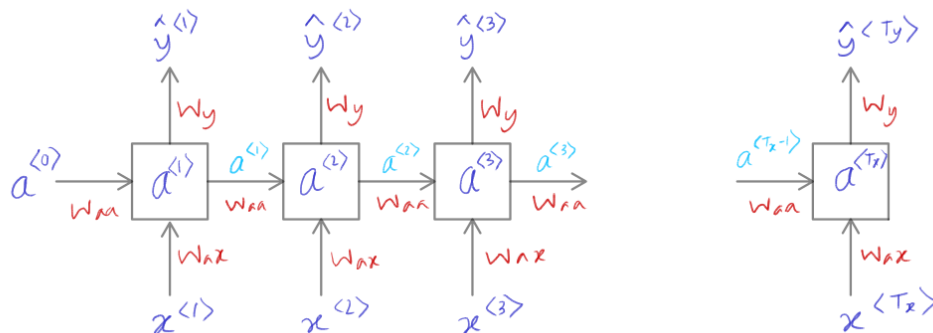
## Contents

1	Problems in using standard NN	1
2	Forward propagation in RNN	2
3	Backpropagation through time	2
4	Different Types of RNN	2
4.1	Many-to-many . . . . .	2
4.1.1	Input and output lengths are same . . . . .	2
4.1.2	Input and output lengths are different . . . . .	2
4.2	Many-to-one . . . . .	3
4.3	One-to-many . . . . .	3

## 1 Problems in using standard NN

- Inputs, outputs can be different lengths in different examples.
- Doesn't share features learned across different positions of text. (For example, 'Harry' appearing as the first word in the sentence, 'he' appearing afterward, standard NN wouldn't be able to learn that 'Harry' is a male character.)

RNN solves these problems. Below is the RNN architecture:



## 2 Forward propagation in RNN

We start with  $\mathbf{a}^{(0)} = \mathbf{0}$  or a vector with random initialization and at  $t$ -timestamp we have

$$\begin{aligned}\mathbf{a}^{(t)} &= g_1(W_{aa}\mathbf{a}^{(t-1)} + W_{ax}\mathbf{x}^{(t)} + b_a) = g_1(W_a[\mathbf{a}^{(t-1)}, \mathbf{x}^{(t)}]^t + b_a) \\ \hat{\mathbf{y}}^{(t)} &= g_2(W_{ya}\mathbf{a}^{(t)} + b_y)\end{aligned}$$

Here  $W_a = [W_{aa}|W_{ax}]$ . Generally we use tanh or ReLu activations for  $g_1$  and depending on what output  $y$  is, for example if it a classification task we use softmax for  $g_2$ .

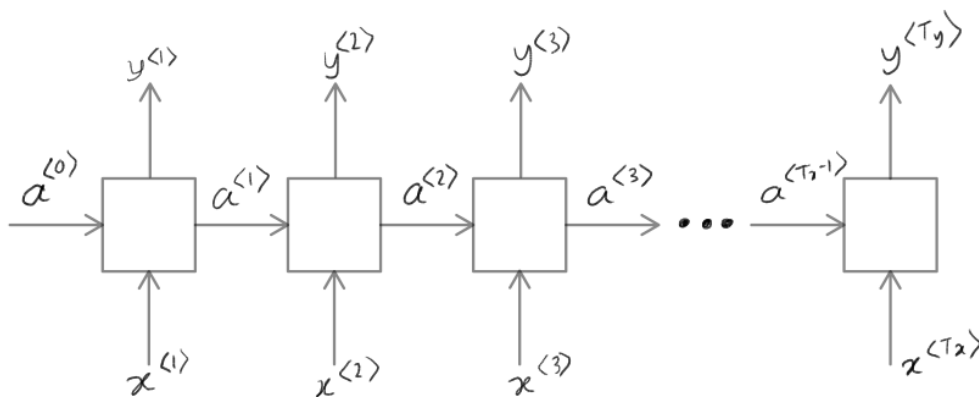
## 3 Backpropagation through time

$$\mathcal{L}(\hat{\mathbf{y}}, \mathbf{y}) = \sum_{t=1}^{T_y} \mathcal{L}^{(t)}(\hat{\mathbf{y}}^{(t)}, \mathbf{y}^{(t)})$$

## 4 Different Types of RNN

### 4.1 Many-to-many

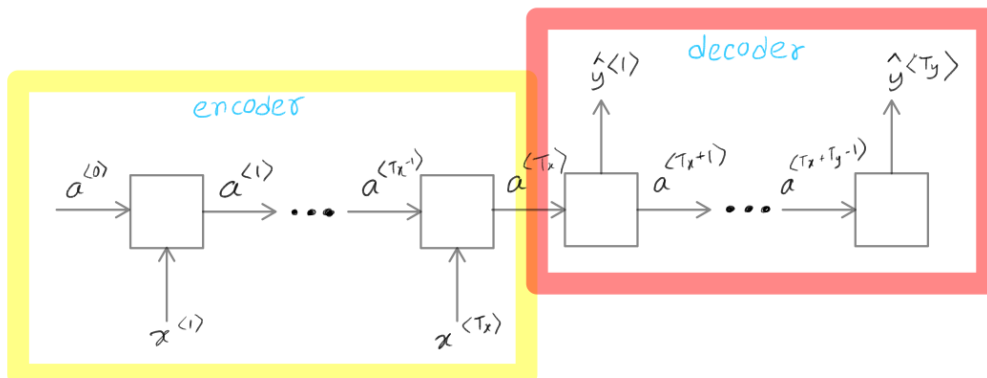
#### 4.1.1 Input and output lengths are same



Applications:

- Name entity recognition

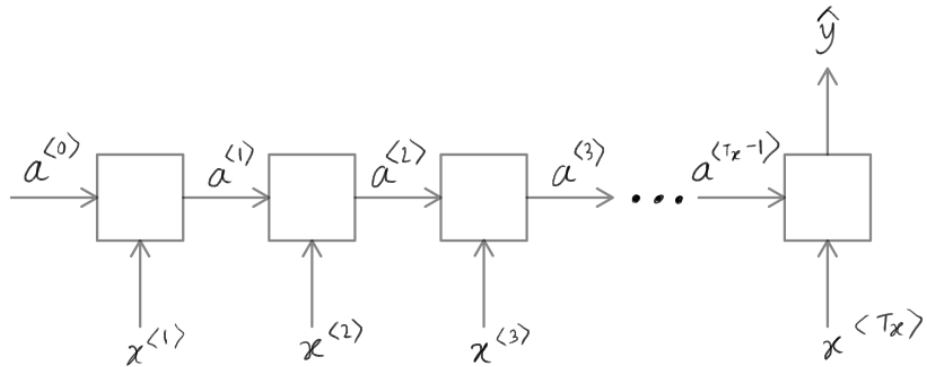
#### 4.1.2 Input and output lengths are different



**Applications:**

- Machine translation

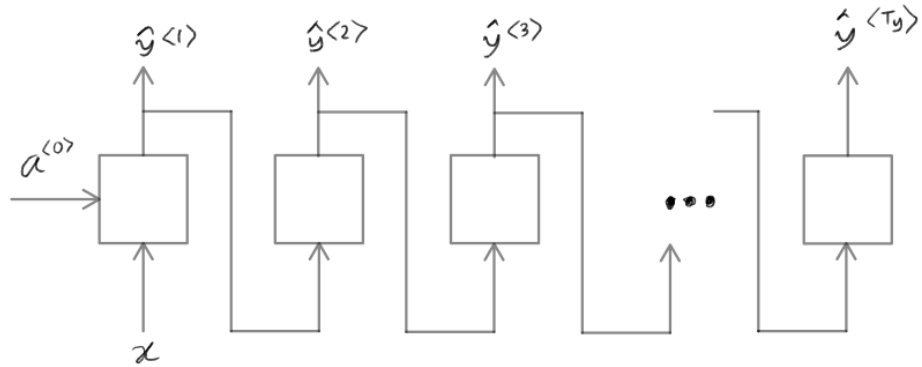
## 4.2 Many-to-one



**Applications:**

- Sentiment Analysis

## 4.3 One-to-many



**Applications:**

- Music generation