# Sequence Modeling
# Introduction to RNNs

Deep Neural Network

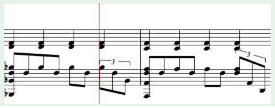
Session 18

Pramod Sharma
pramod.sharma@prasami.com

# Agenda

- Sequence Modeling
- Introduction to RNN
- Different Architectures
- Language Modelling
- Image Captioning

# Examples – Sequence Modelling

| Domain | Data Type | Output type |
|---|---|---|
| Speech Recognition | Audio | Words (text) |
| Music Creation | Nodes ( ∅ ) | Audio  |
| Sentiment classification | … an enjoyable one-time-watch for the funny punchlines, far-out characters and performances. But the unconvincing story and the temperate screenplay prevent it from reaching its full potential … | Integers ( Stars ratings from 1 to 5) |
| Machine Translation | डीएनएन व्याख्यानमाला आपले स्वागत आहे। | Welcome to DNN Lecture. |
| Named Entity Recognition | Mohan was driving a Maruti | Mohan was driving a Maruti |
| Video activity recognition | Sequence of Video Frames | Identify activity say running |

# Sequence Modeling – Named Entity Recognition

- x : <u>Mohan</u> was driving a <u>Maruti</u>
- y: 1 0 0 0 1

# Sequence Modeling – Named Entity Recognition

- x : <Mohan Sharma> was driving a <Maruti 800>

- y: 1 0 0 0 1

# Sequence Modeling – Named Entity Recognition

- x : <u>Mohan</u> was driving a <u>Maruti</u>

  $x_1$ $x_2$ $x_3$ $x_4$ $x_5$ $\Rightarrow x_t$

- y: 1 0 0 0 1

  $y_1$ $y_2$ $y_3$ $y_4$ $y_5$ $\Rightarrow y_t$

- $T_x$ is length of input and $T_y$ is length of output

# Representing Words

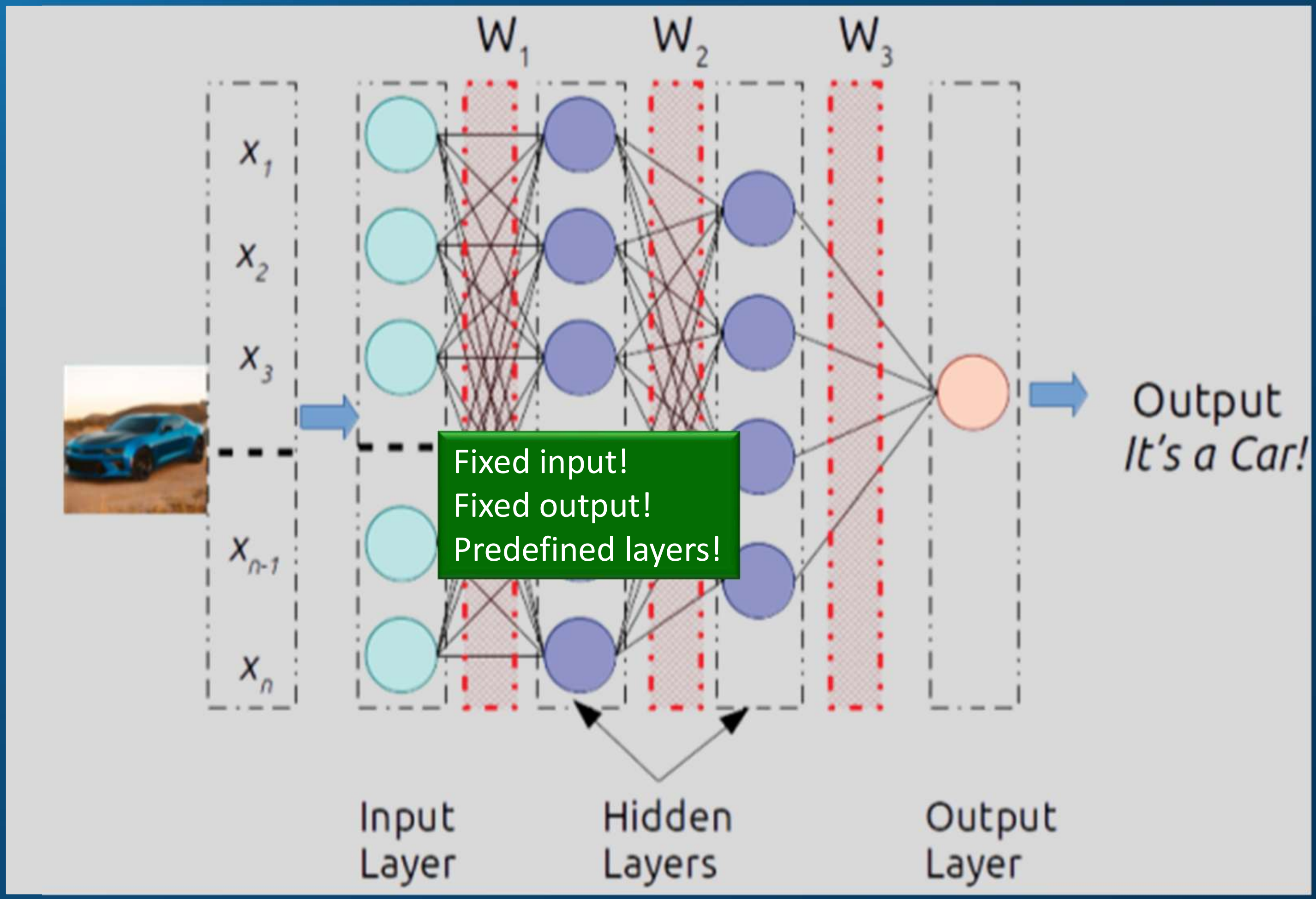
- Vocabulary = [a, aakash, aamaan… to zulu, zyzzogeton]
  - Also referred as corpus
  - Two more tokens <UNK> and <EOS>

- Can be converted to one hot encoding

- x : Mohan was driving a Maruti

- $$\begin{array}{ccccc} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ - & - & 1 & - & - \\ - & - & - & - & - \\ - & - & - & - & - \\ - & - & - & 0 & 1 \\ 1 & - & - & - & - \\ - & 1 & - & - & - \\ 0 & 0 & 0 & 0 & 0 \end{array}$$

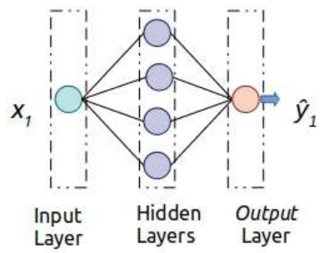
# Using Standard Architecture



# To Summarize….

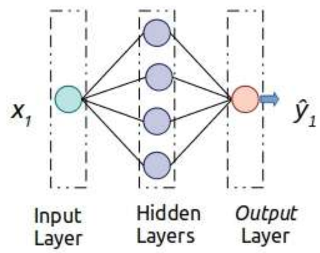
- Not all problems can be converted into one with fixed length inputs and outputs
- Problems such as Speech Recognition or Time-series Prediction require a system to store and use context information
- Hard/Impossible to choose a fixed context window
- There can always be a new sample longer than anything seen

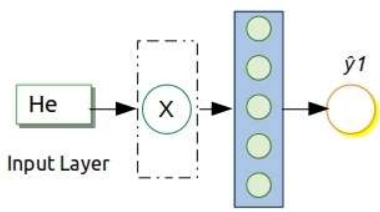
# What is Recurrent Neural Network…




- Remember our little Neural Network…
- Let's simplify the layout a little

# What is Recurrent Neural Network…

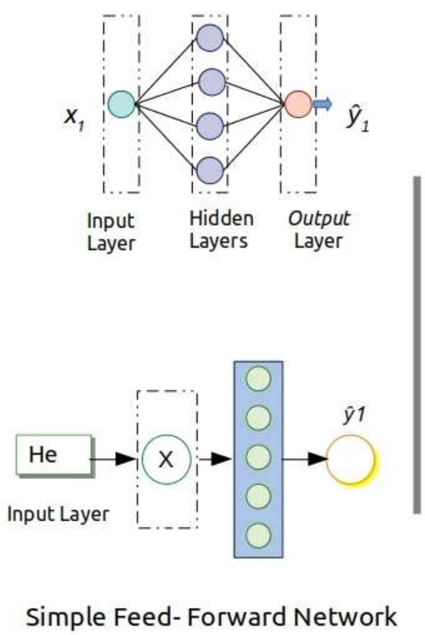


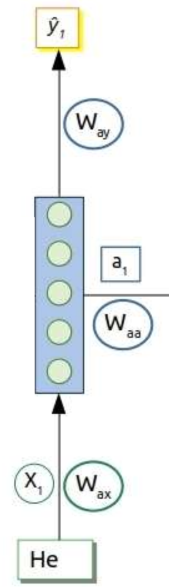




Simple Feed- Forward Network

- ❑ It takes one value and gives probability of it being a word or character or a value

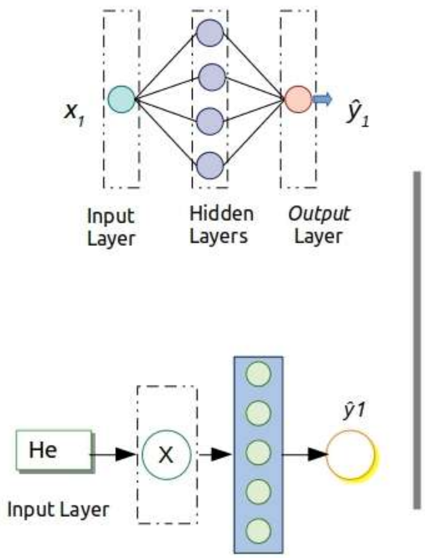
# What is Recurrent Neural Network…



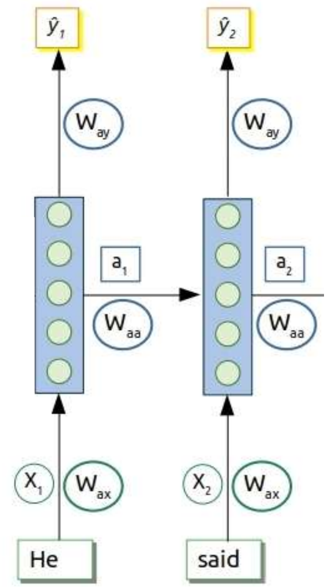




- Let's also calculate activations $a_1$ and weights $W_{aa}$
- Assume that we have some method of calculating them
- At the moment both $W_{ax}$ and $W_{aa}$ would seem to be same

# What is Recurrent Neural Network…

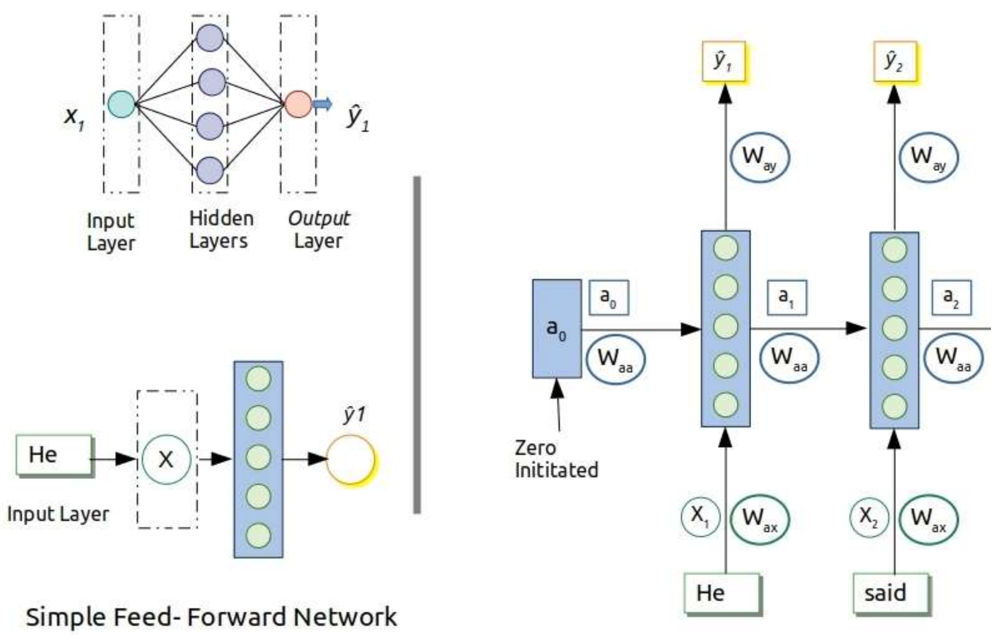



Simple Feed- Forward Network



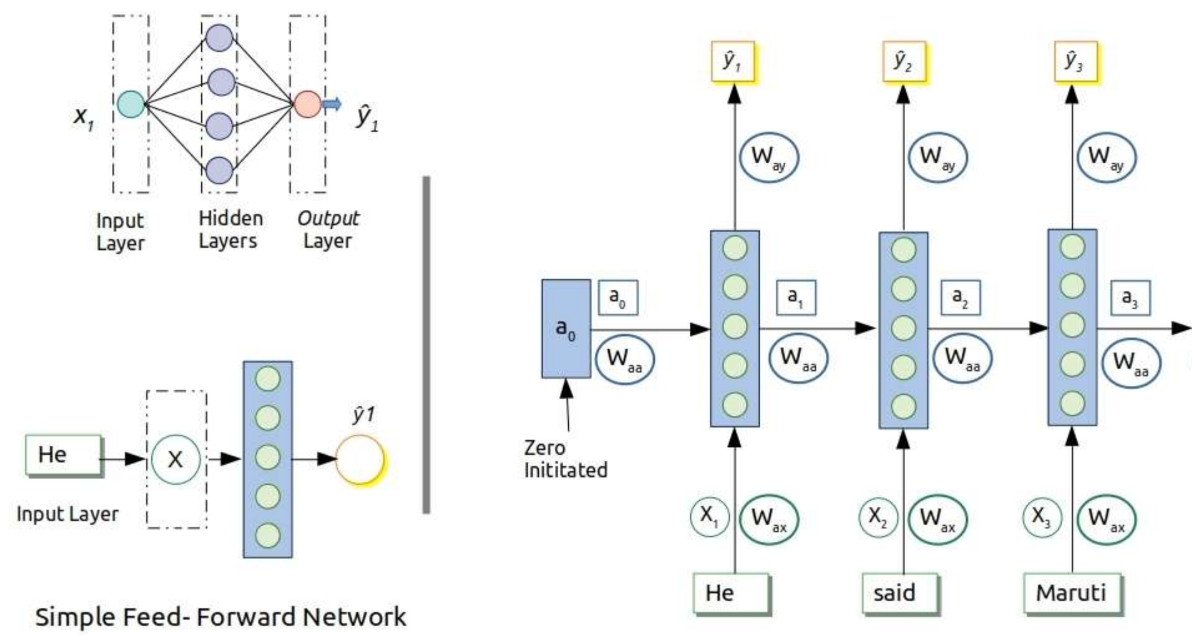

- Using the weights and activations, read $X_2$ and process it through the network
- Calculation of $\hat{y}_2$ will be based on $X_2$, $W_{ax}$, $W_{aa}$ and $a_1$,

# What is Recurrent Neural Network…




- But it makes two set of calculations
- Using different formulae
- To make it consistent let's initialize $a_0$ with weights $W_{aa}$

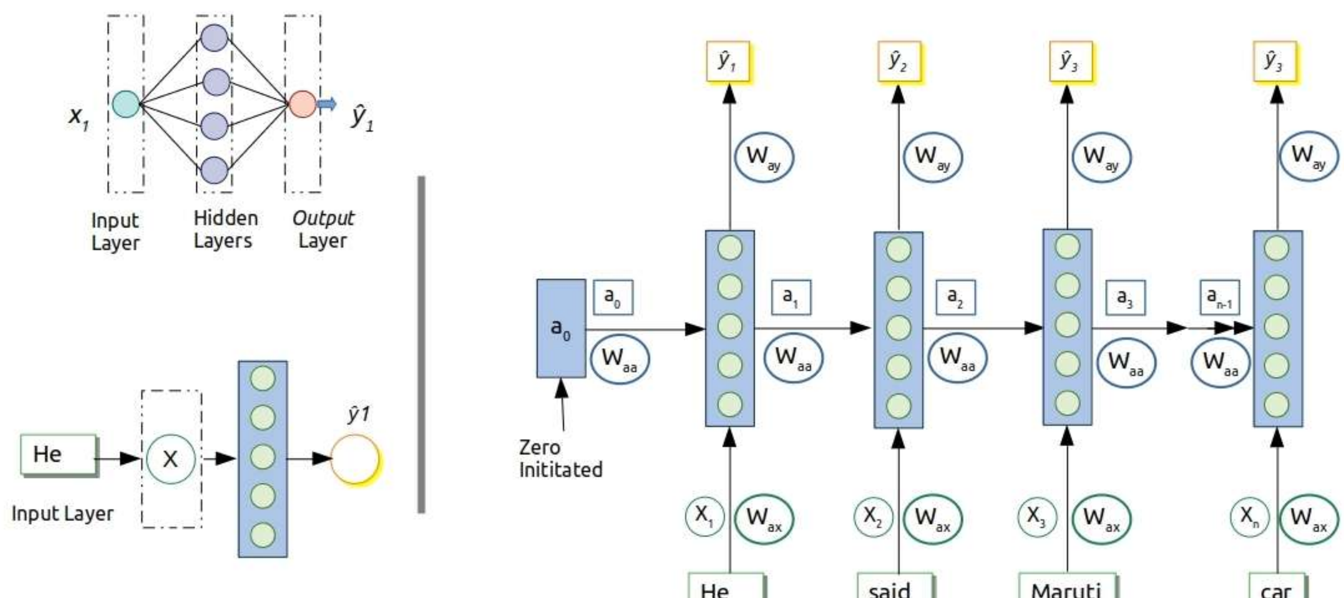
# What is Recurrent Neural Network…




- Similarly we can calculate $\hat{y}_3$

17

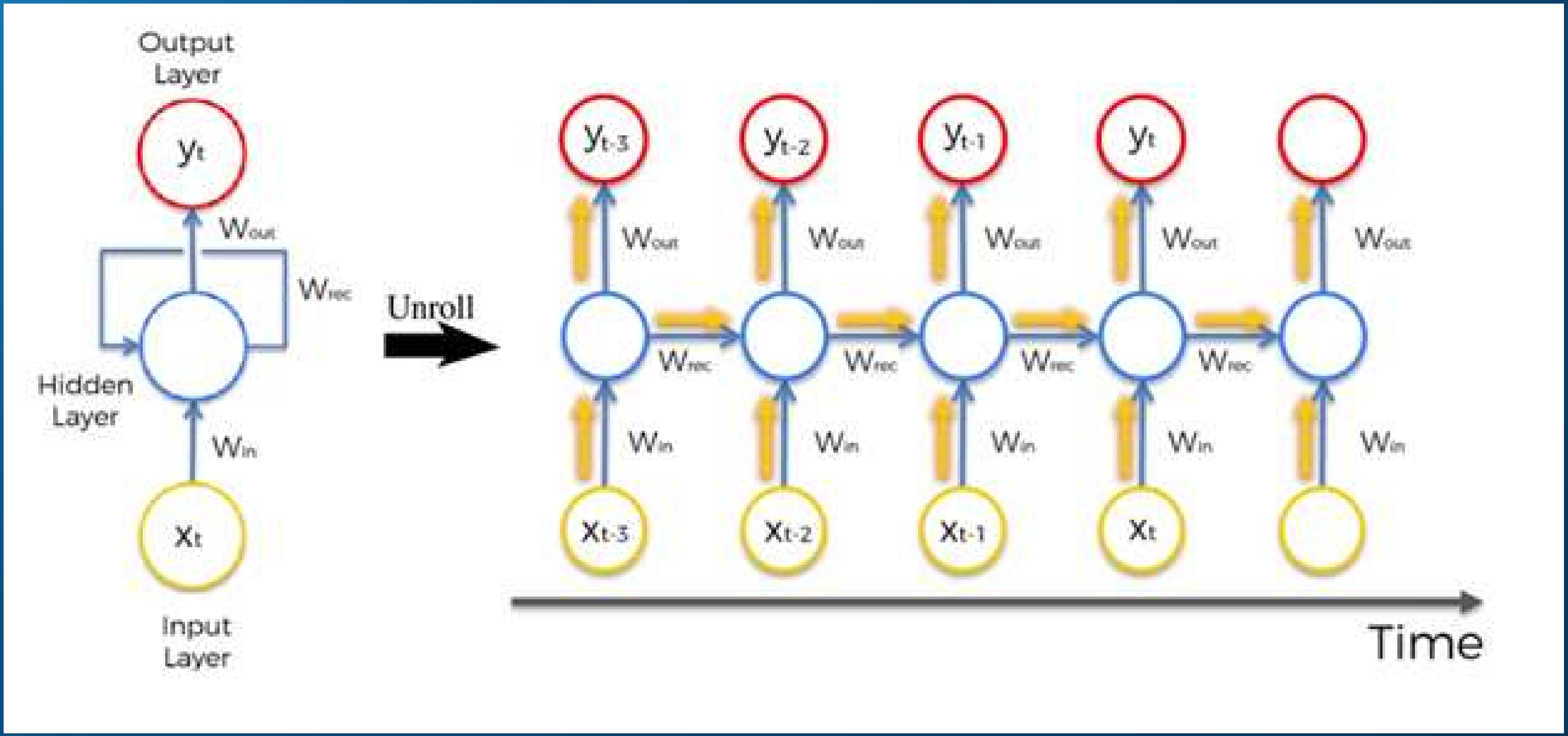
# What is Recurrent Neural Network…



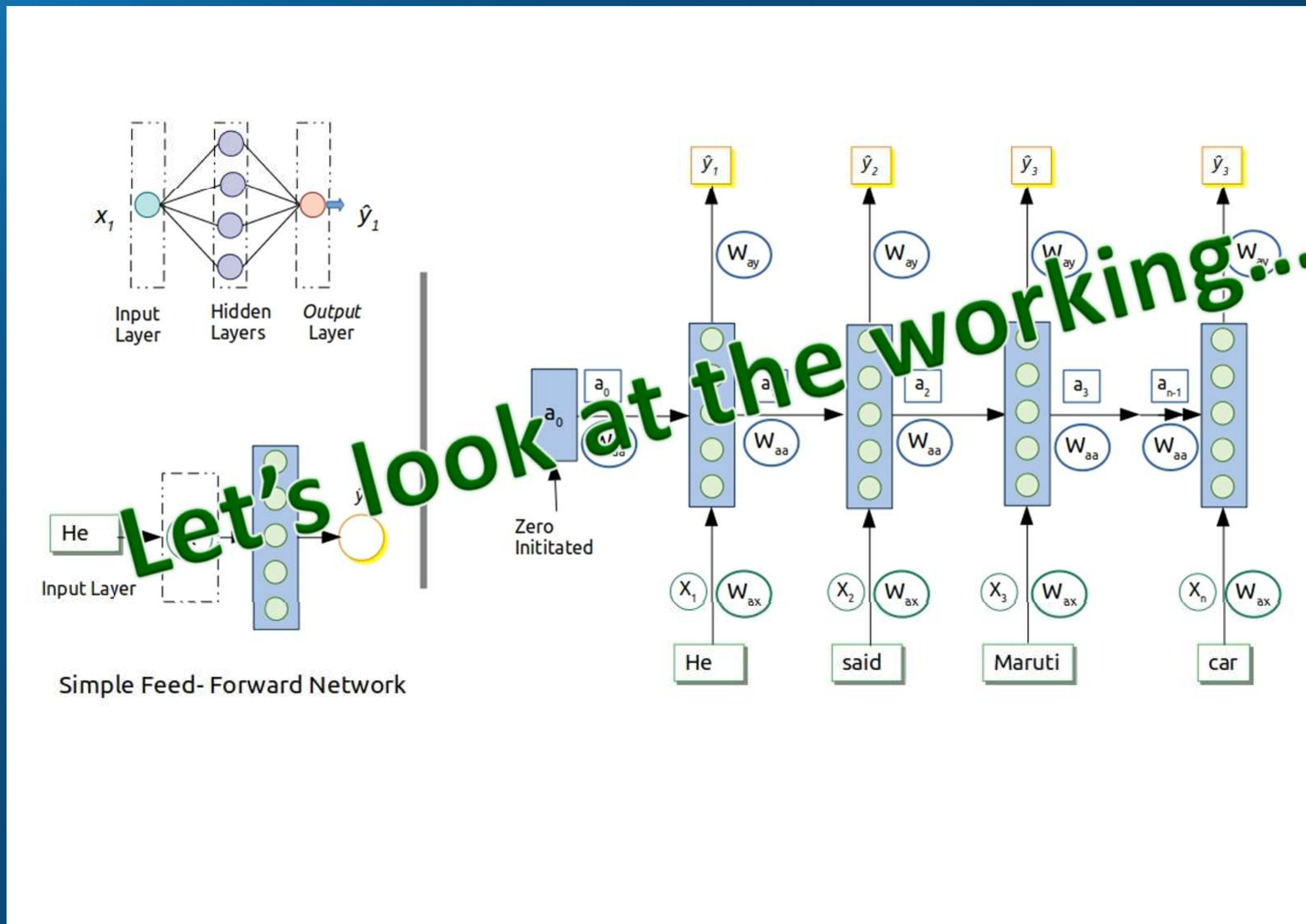

Simple Feed- Forward Network

- ❑ And continue till end,
- ❑ Some literatures represent it with a loop,

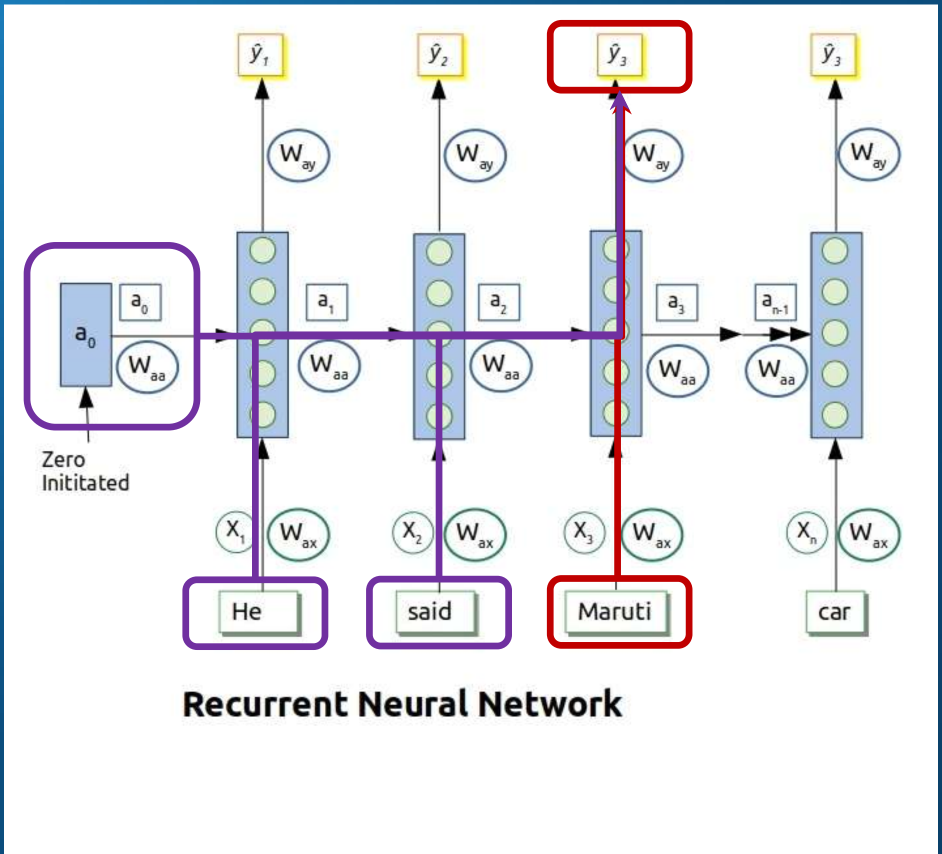
# Alternate Representations



# What is Recurrent Neural Network…

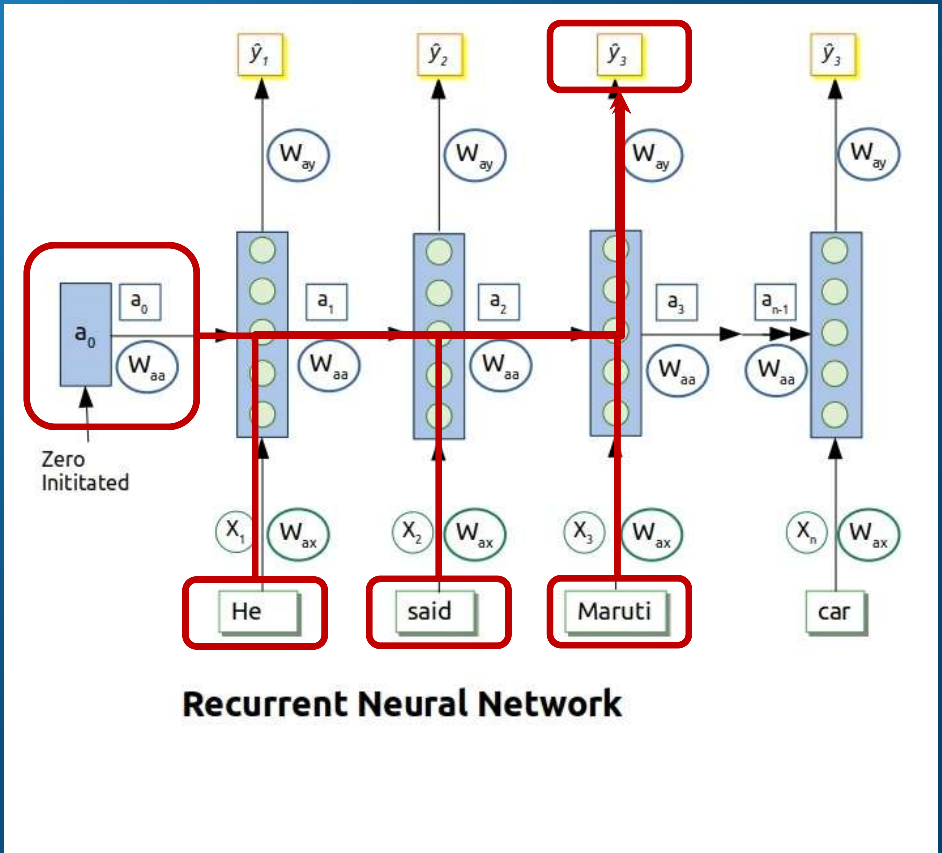


# What is Recurrent Neural Network…




- Taking activations from previous time step also
- The $W_{ax}$ and $W_{aa}$ are shared parameters across all time steps
- So, for calculation of $\hat{y}_3$ would be influenced by those for $\hat{y}_2$ and $\hat{y}_1$

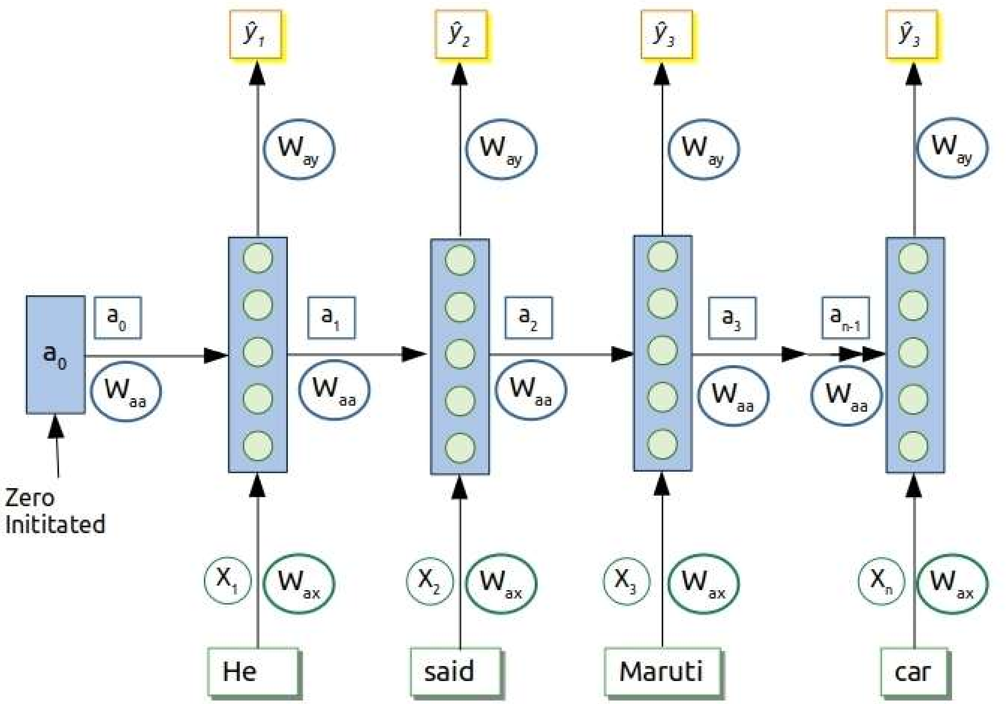
# What is Recurrent Neural Network…




- It is using the information till time step 3.
  - He said “*Maruti…*

- However, it has no clue what comes next!!!
  - He said “*Maruti is most fuel efficient car*”
  - He said “*Maruti is most expensive shop*”
  - He said “*Maruti is strongest*”

# That's is Recurrent Neural Network…

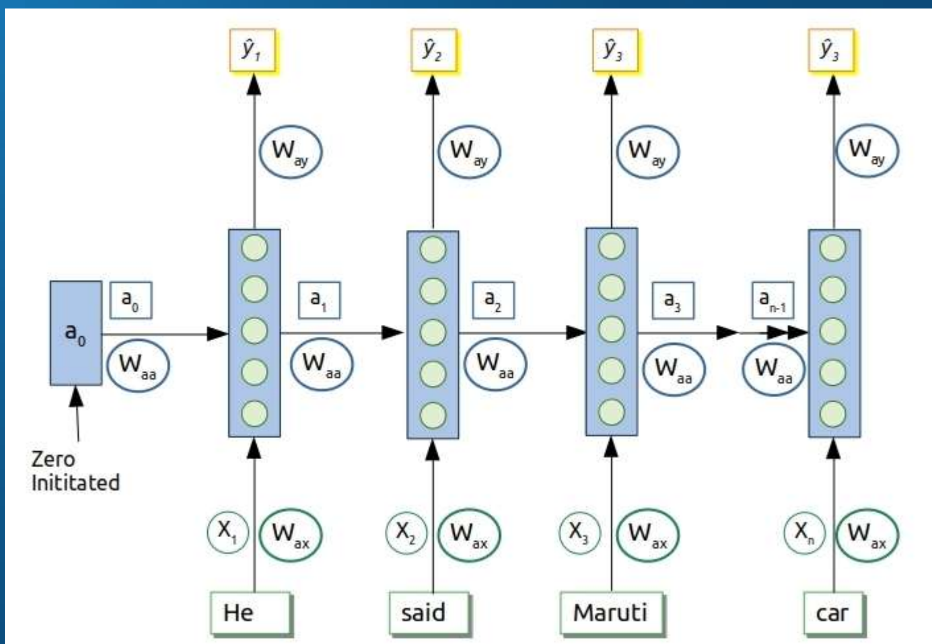



- ❑ Its it great!
- ❑ All done… sealed, signed, and delivered…
- ❑ Wait… let's do some math too….

# What We Know So Far….

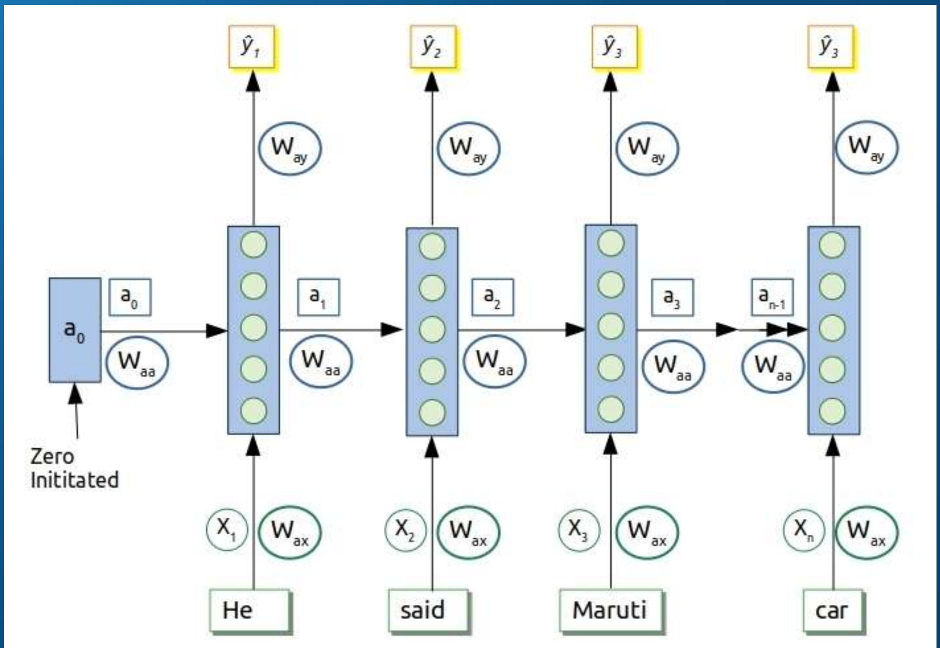
- Recurrent Neural Networks take the previous output or hidden states as inputs.
- The composite input at time ‘t’ has some historical information about the happenings at time ‘T’ < ‘t’.
- RNNs are useful as their intermediate values (state) can store information about past inputs for a time that is not fixed a priori
- Note that the weights are shared over time
- Essentially, stacks of the RNN cell are made over time (unrolling/unfolding), with different inputs at different time steps

# Forward Propagation



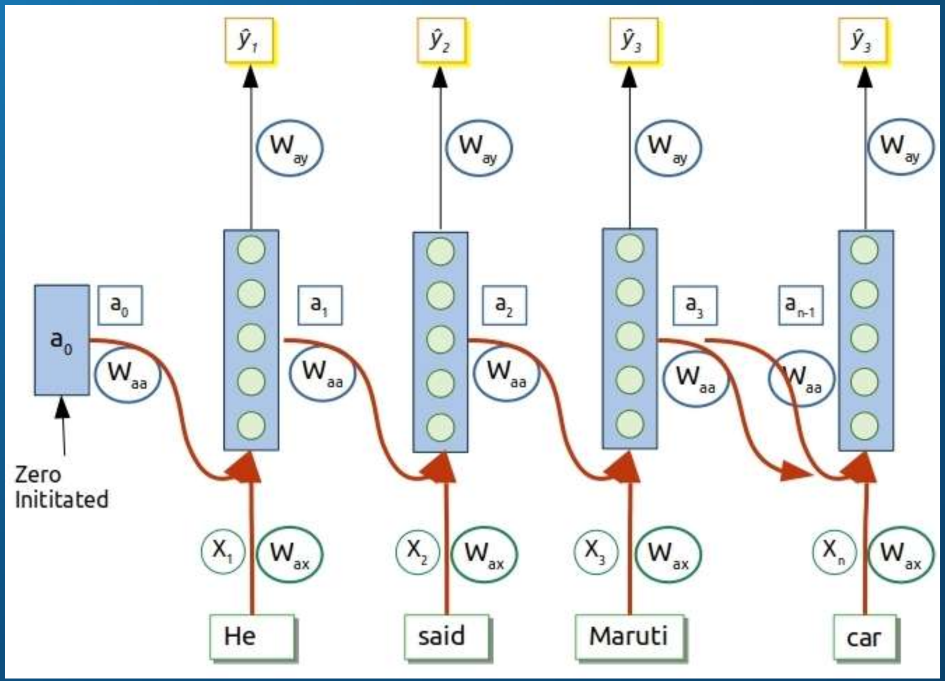

- Let's work on equations

# Forward Propagation




- To start with; $a_0$ is vector of all zeros
  - $a_1 = g_1 (a_0 \ . W_{aa} + X_1 . W_{ax} + b_a)$ ➔ Tanh / ReLU
  - $\hat{y}_1 = g_2 (a_1 . W_{ay} + b_y)$ ➔ Sigmoid/Softmax (for classification)
- Tanh Activation function is more prevalent in RNN
  - Sometime ReLU too is used
- For output layers, the activation function will depend on type of output
- Generally, at 't' we can write
  - $a_t = g_1 (a_{t-1} \ . W_{aa} + X_t . W_{ax} + b_a)$
  - $\hat{y}_t = g_2 (a_t . W_{ay} + b_y)$

# Forward Propagation

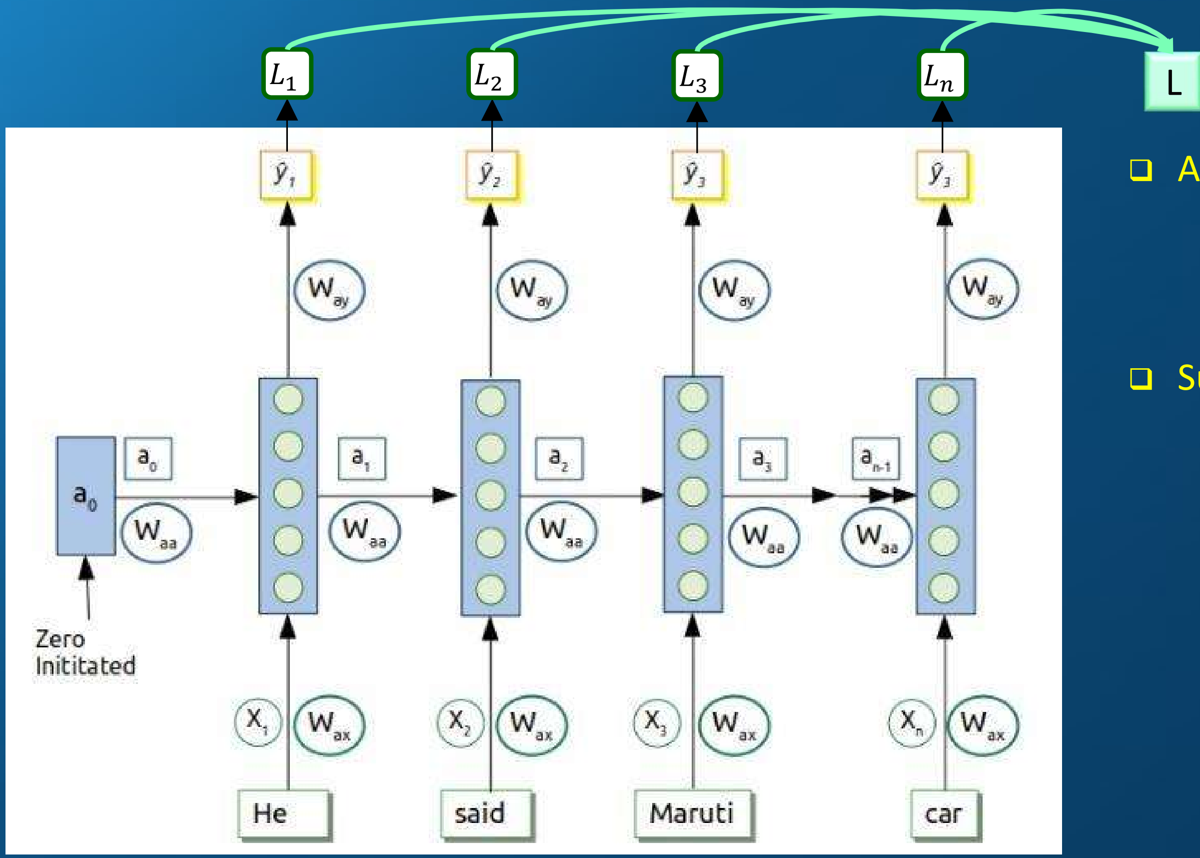



- Our equations
  - $a_t = g_1(a_{t-1}.W_{aa} + x_t.W_{ax} + b_a)$
  - $\hat{y}_t = g_2\left(a_t.W_{ya} + b_y\right)$

- Can be written as:
  - $a_t = g_1([a_{t-1}.\ x_t]\ W_a + b_a)$
  - $\hat{y}_t = g_2\left(a_t.W_y + b_y\right)$
  - where $W_a$ will be stacked matrix of $W_{aa}$ and $W_{ax}$
  - $W_a = \begin{bmatrix} W_{aa} \\ W_{ax} \end{bmatrix}$
  - Similarly ,
  - $[a_{t-1}.\ x_t]$ = $[a_{t-1}. | \ x_t]$

- We know that :
  - $[a_{t-1}. | \ x_t]$ . $\begin{bmatrix} W_{aa} \\ W_{ax} \end{bmatrix}$ = $a_{t-1}.W_{aa}$+ $x_t.\ W_{ax}$

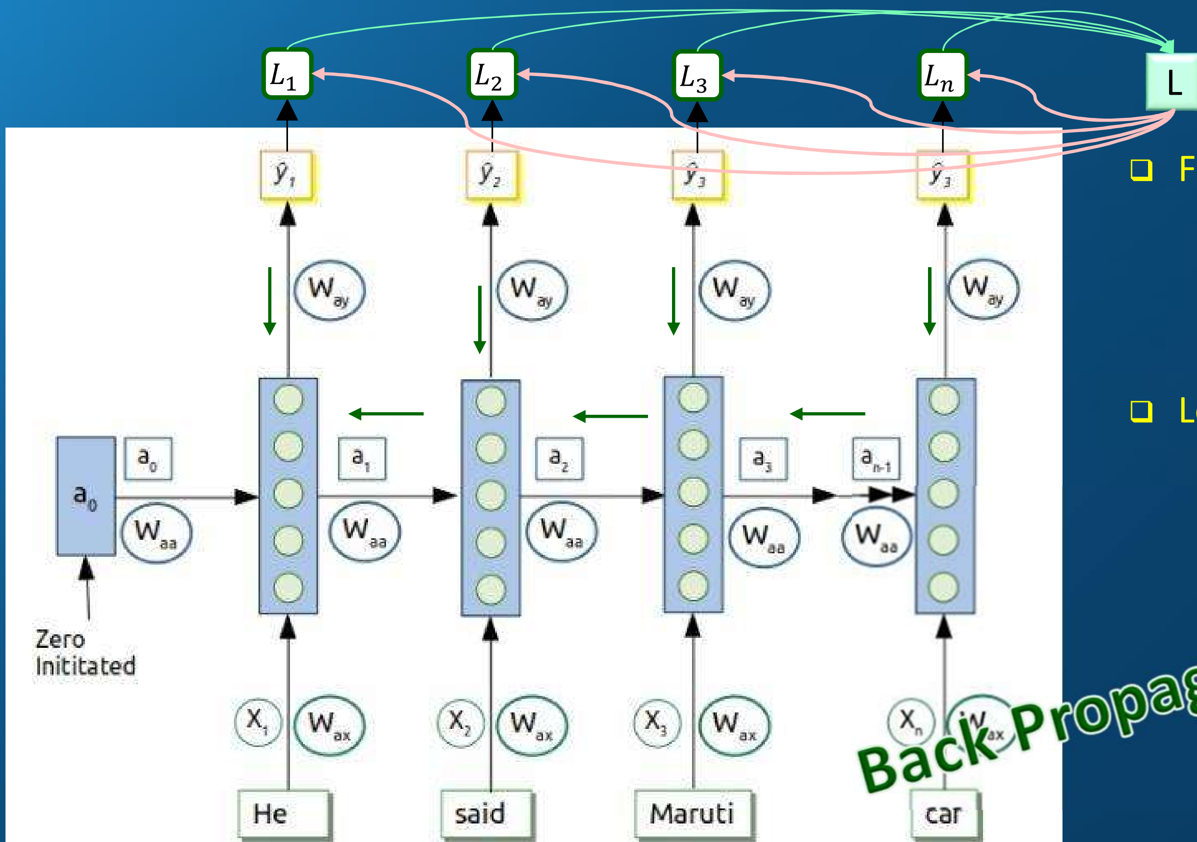
# Back Propagation




- At time step 't'; Loss Function for single prediction
  - $L_t(\hat{y}_t, y) = - y_t \, . \log(\hat{y}_t) - (1- y_t) \, . \log(1 - \hat{y}_t)$

- Sum of losses at all time steps:
  - $L(\hat{y}, y) = \sum_{t=1}^{T_x} L_t(\hat{y}_t, y)$

# Back Propagation




- Forword propagation:

$$a_t = g_1([a_{t-1}, x_t] \,.\, W_a + b_a)$$
$$\hat{y}_t = g_2(a_t . W_y + b_y)$$

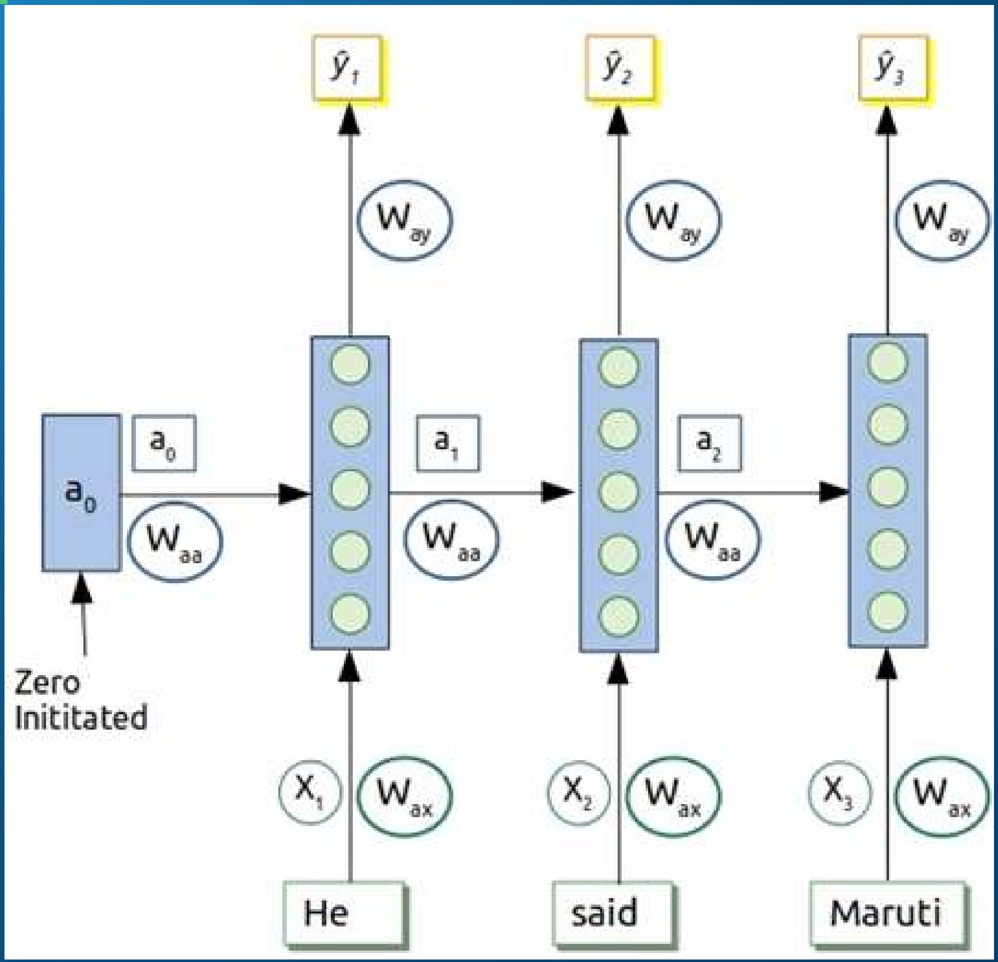
- Loss Function
  - $L_t(\hat{y}, y) = - y_t \,.\, \log(\hat{y}_t) - (1 - y_t) \,.\, \log(1 - \hat{y}_t)$

Back Propagation through Time.



pra-sâmi

# Back Propagation Through Time…




Forword propagation:

$$a_t = g_1([a_{t-1}, x_t].W_a + b_a)$$
$$\hat{y}_t = g_2(a_t.W_y + b_y)$$

Loss Function :

$$L_t(\hat{y}, y) = -y_t.\log(\widehat{y_t}) - (1 - y_t).\log(1 - \widehat{y_t})$$

Step 3:

$$\frac{dL_3}{dw_y} = \frac{dL_3}{d\hat{y}_3}.\frac{d\hat{y}_3}{dw_y}$$

$$\frac{dL_3}{dw_a} = \frac{dL_3}{d\hat{y}_3}.\frac{d\hat{y}_3}{da_3}.\frac{da_3}{dw_a} + \frac{dL_3}{d\hat{y}_3}.\frac{d\hat{y}_3}{da_3}.\frac{da_3}{da_2}.\frac{da_2}{dw_a} + \frac{dL_3}{d\hat{y}_3}.\frac{d\hat{y}_3}{da_3}.\frac{da_3}{da_2}.\frac{da_2}{da_1}.\frac{da_1}{dw_a}$$

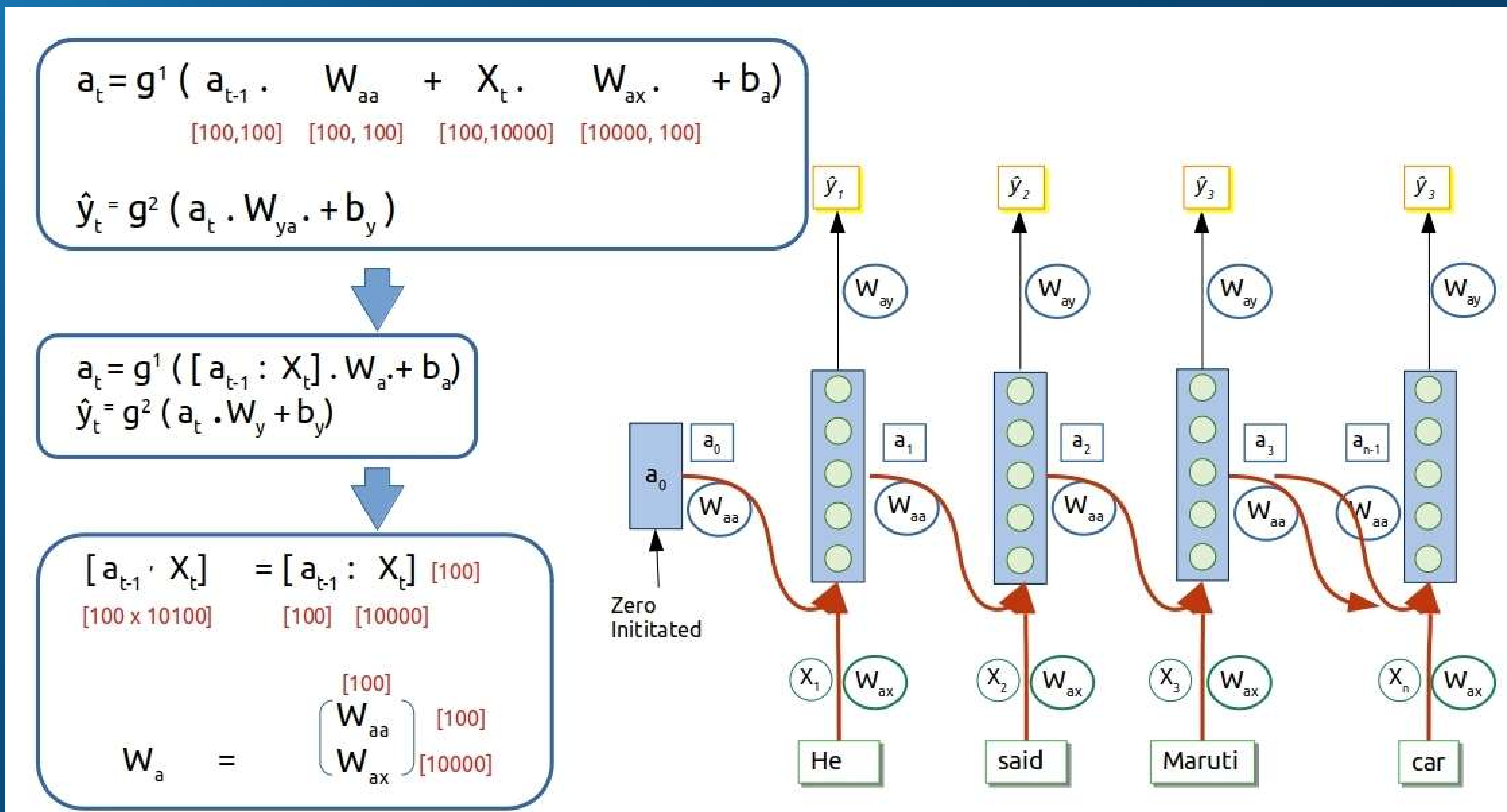
There is a pattern here!

$$\frac{dL_3}{dw_x} = \frac{dL_3}{d\hat{y}_3}.\frac{d\hat{y}_3}{da_3}.\frac{da_3}{dw_x} + \frac{dL_3}{d\hat{y}_3}.\frac{d\hat{y}_3}{da_3}.\frac{da_3}{da_2}.\frac{da_2}{dw_x} + \frac{dL_3}{d\hat{y}_3}.\frac{d\hat{y}_3}{da_3}.\frac{da_3}{da_2}.\frac{da_2}{da_1}.\frac{da_1}{dw_x}$$

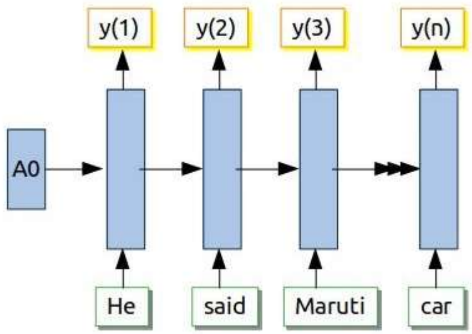


pra-sâmi

# Quickly check the dimension….



# Type of Architectures



Many-to-Many Architecture

Many to many mapping. $T_x$ input parameters are same as $T_y$ output parameters

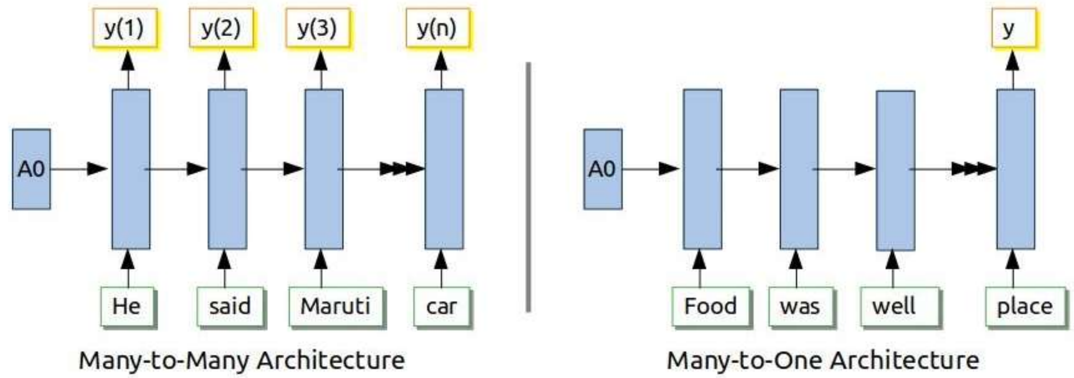
Named entity Recognition:

Mohan was driving a Maruti

➔ 1 0 0 0 1

The Unreasonable Effectiveness of Recurrent Neural Networks
- Andrej Karpathy

# Type of Architectures

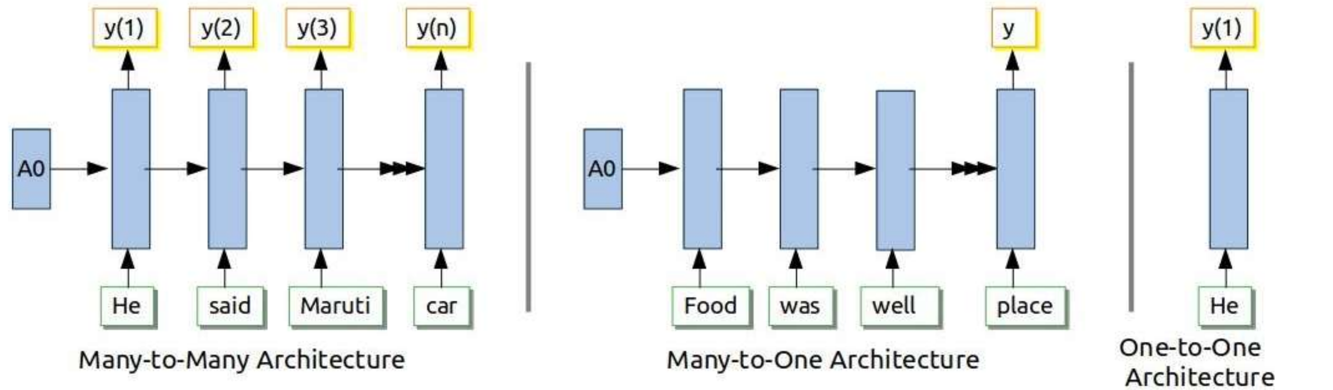



Many to one architecture.

Input is the ‘review’ written by a patron and output is an integer (star rating)

The Unreasonable Effectiveness of Recurrent Neural Networks
- Andrej Karpathy

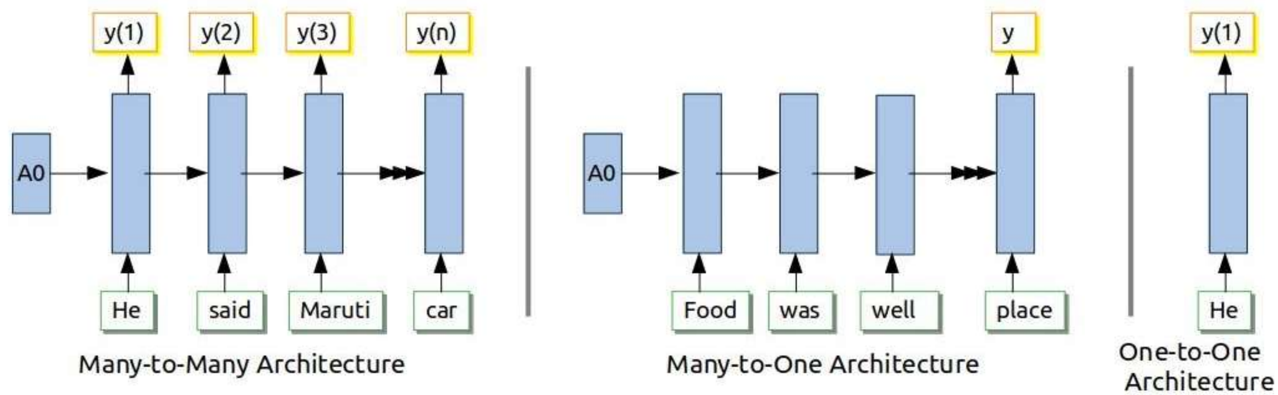
# Type of Architectures

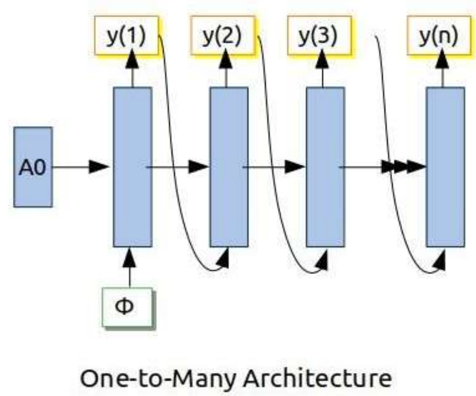



Of course there is one to one. i.e. Basic neural network…

The Unreasonable Effectiveness of Recurrent Neural Networks - Andrej Karpathy

# Type of Architectures

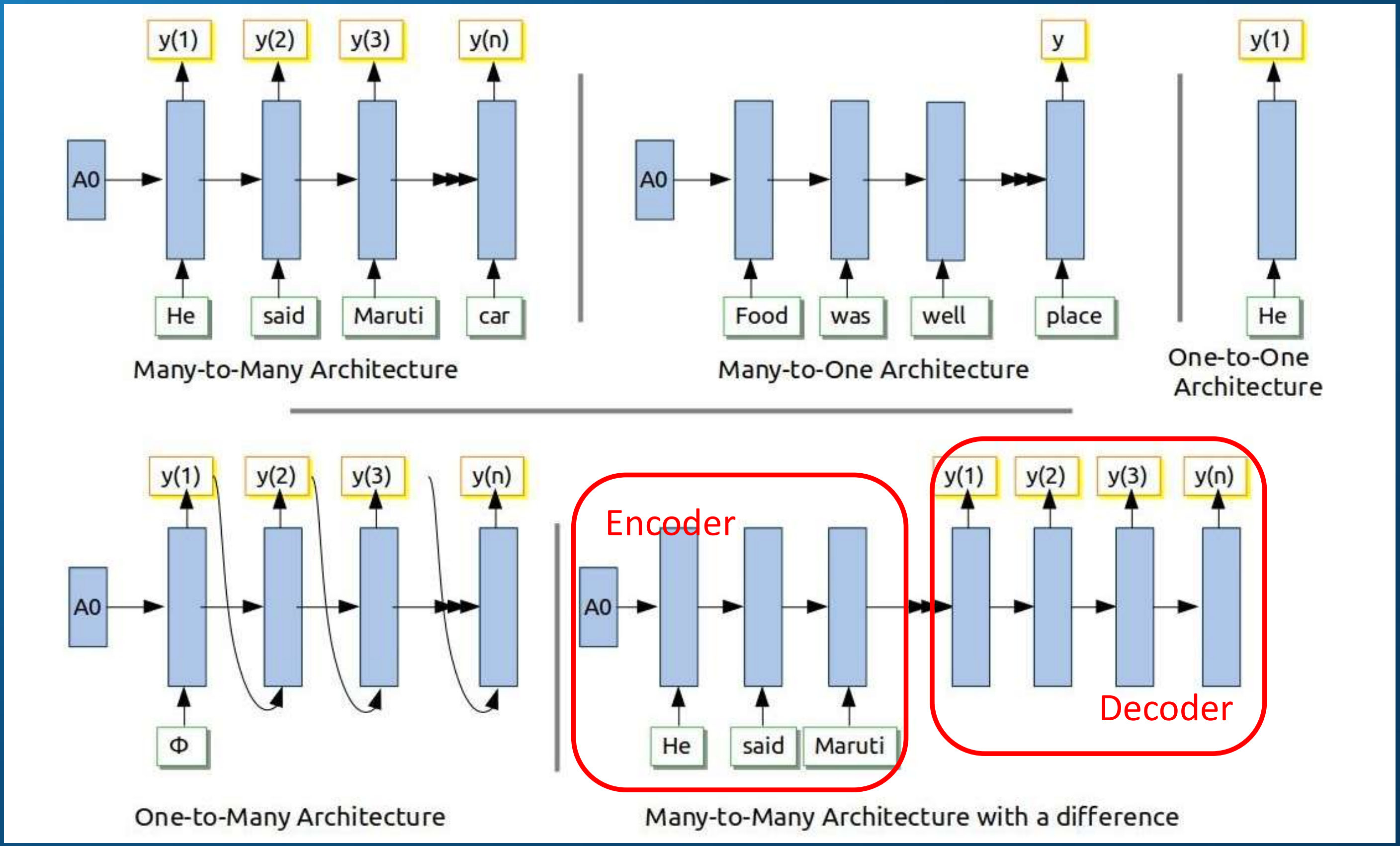



We also have one to many architecture, mapping one input to multiple outputs. Output from each layer becomes input for next time step.

The Unreasonable Effectiveness of Recurrent Neural Networks
- Andrej Karpathy

# Type of Architectures




डीएनएन व्याख्यानमाला आपले स्वागत आहे|

➔ Welcome to DNN Lecture

In this Architecture, we have two completely different parts. One side reading sentences in one language, and other side translating in different language.

We can have $T_x$ and $T_y$ different which is a case in machine translations

The Unreasonable Effectiveness of Recurrent Neural Networks
- Andrej Karpathy

# Language Modelling

Speech Recognition

- Toad met Pit….
- Todd met Pete…
- Given any sentence, what is the probability of that being a valid sentence
- So what language model would do is to calculate probability of a sentence with that combination of words
  - P(Toad met Pit) = $4.6\ x\ 10^{-15}$
  - P(Todd met Pete) = $9.3\ x\ 10^{-9}$
- Mathematically P(sentence) = P($y_1, y_2, y_3, \ldots\ y_n$)

# How to Model?

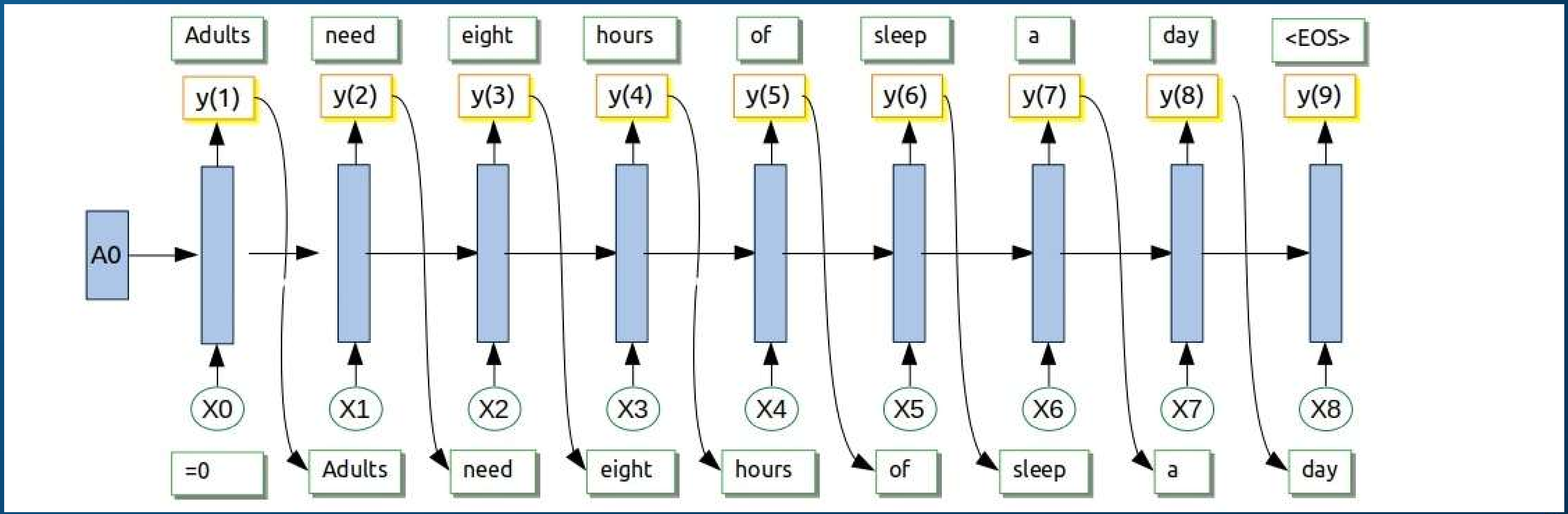
- Training set : Large corpus of English text
  - Adults need eight hours of sleep a day!

| Adults | need | eight | hours | of | sleep | a | day | ! | <EOS> |
|---|---|---|---|---|---|---|---|---|---|
| $y_1$ | $y_2$ | $y_3$ | $y_4$ | $y_5$ | $y_6$ | $y_7$ | $y_8$ | - | $y_9$ |

- First step is to tokenize the sentence
- Add a token at end and at the beginning <EOS> ($y_9$)
- Remember we have limited tokens (say we only have 10,000 tokens).
- Unknown words will be given a token <unk>

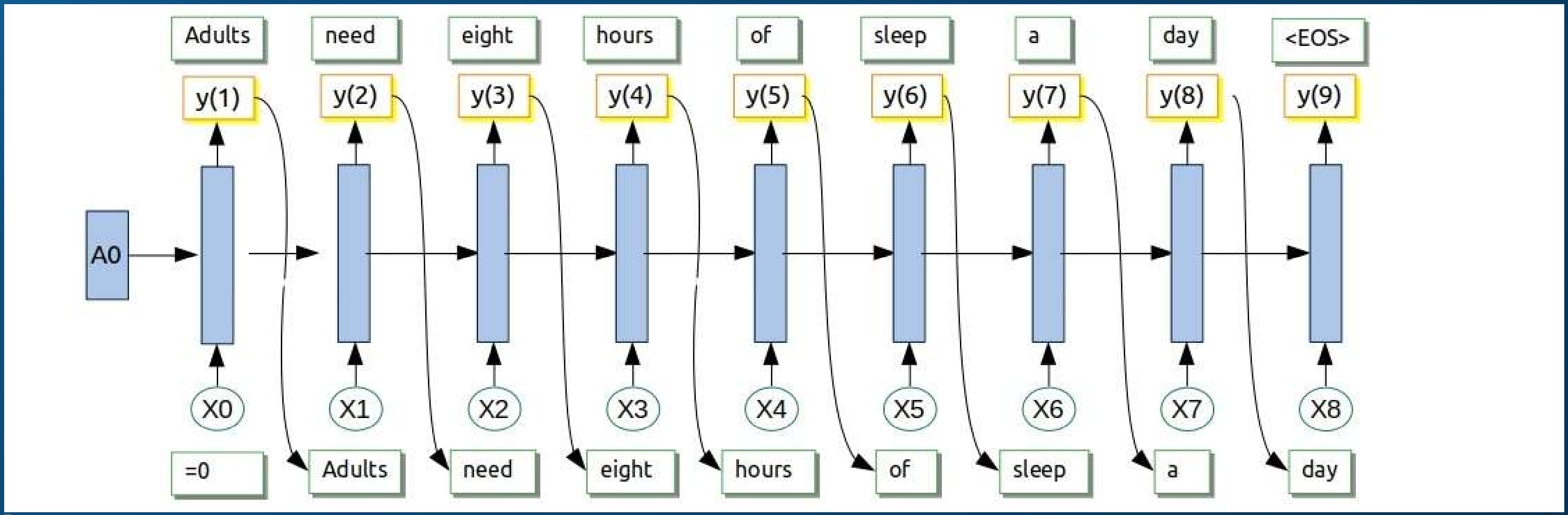
# RNN Model

- At the onset RNN tries to predict probabilities of each word in the corpus of being first word in this sentence.
- i.e. P[a], P[aakash], P[aamaan]… to P[zulu], P[zyzzogeton]
  - This would be an array of 10002 elements



# RNN Model

- Thus we can calculate error between $\hat{y}_1$ and "Adults"
- Given first word "Adults", again RNN predicts the probabilities for second word, thus combined probability, and it continues…
  - i.e. P[a|Adult], P[aakash|Adult], P[aamaan |Adult]… to P[zulu |Adult], P[zyzzogeton |Adult]
- Somewhere in that bunch there will be a probability P[need|Adult]

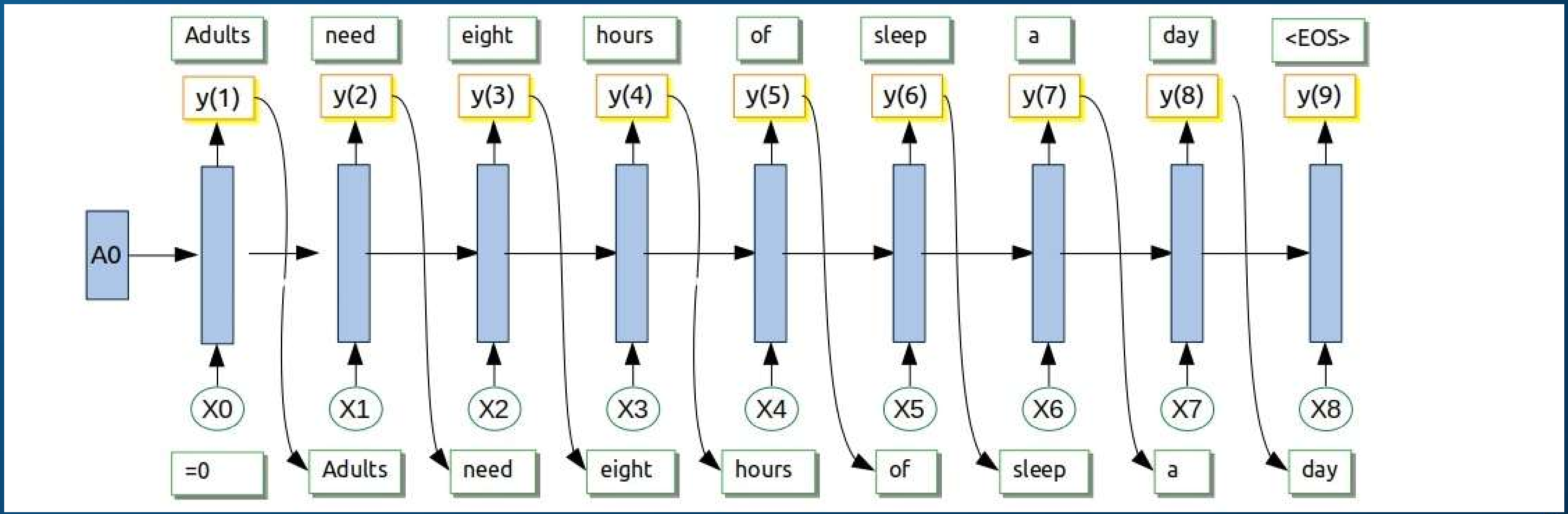





pra-sâmi

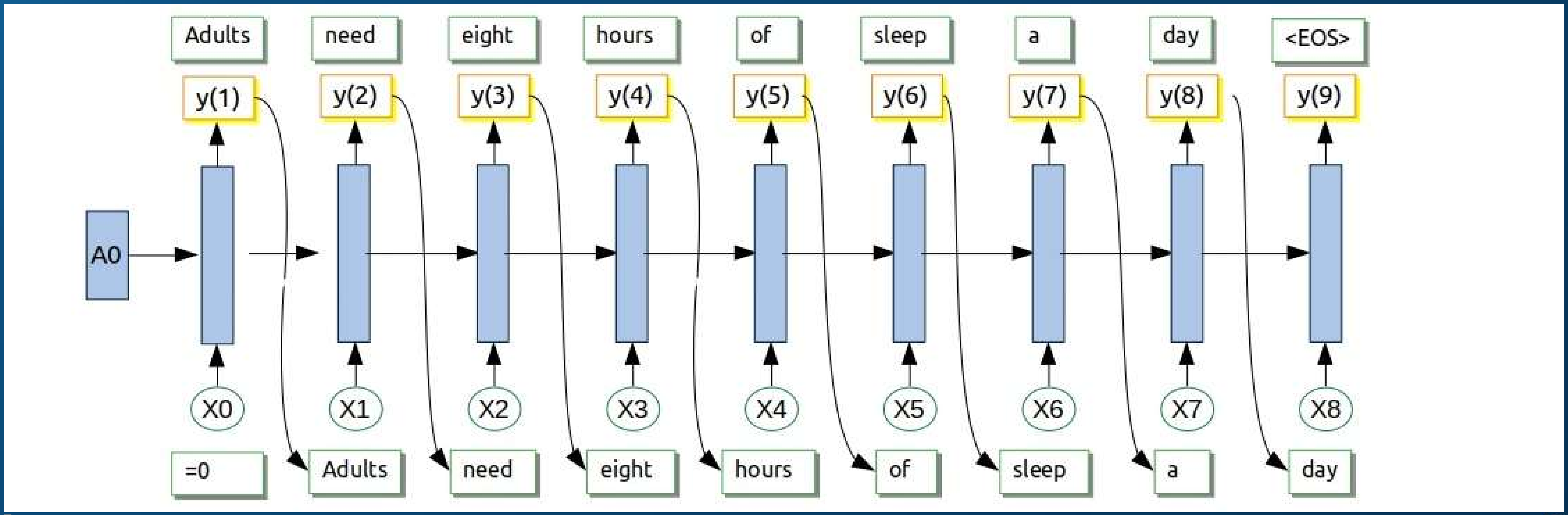
# RNN Model

- At third step we can calculate error between $\hat{y}_2$ and “need”.
- Given first word “Adults”, and second word as “need”, again RNN predicts the probabilities for third word
- i.e. P[a | Adult, need], P[aakash | Adult , need], P[aamaan | Adult , need]… to P[zulu |Adult , need], P[zyzzogeton |Adult , need]
- Somewhere in that bunch there will be a probability P[eight|Adult, need]



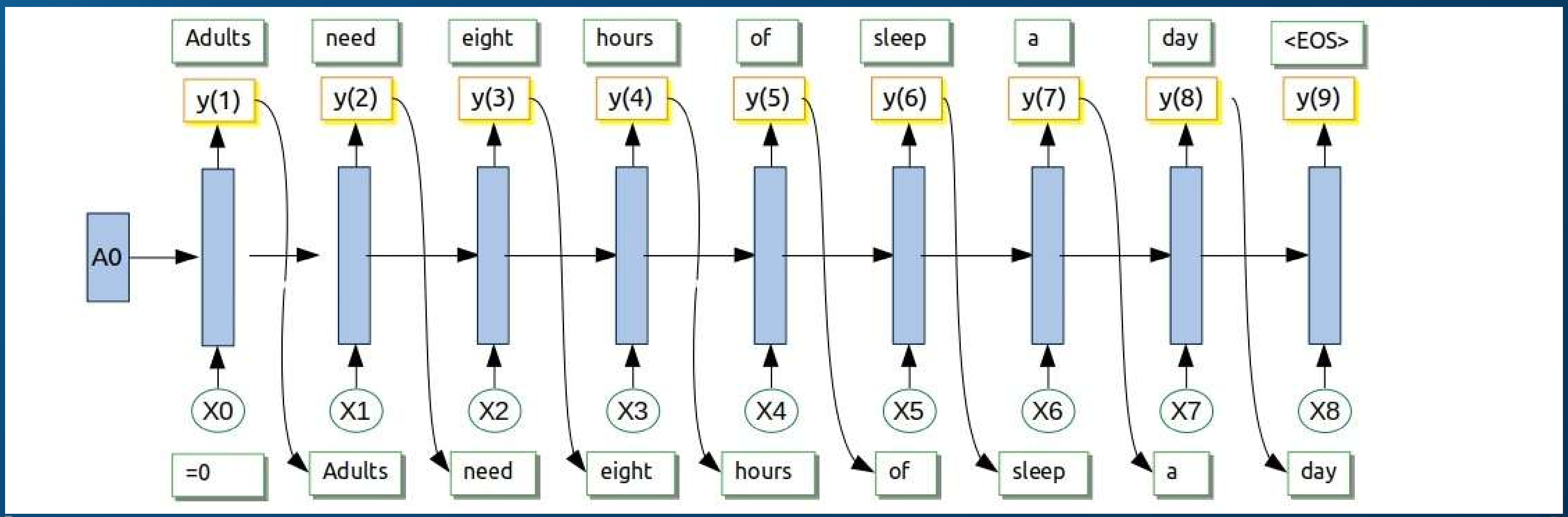
# RNN Model

- Thus we can calculate error between $\hat{y}_3$ and “eight”.
- It continues from left to right till end, $X_8$
- Given all previous words, what is the probability of this word being <EOS>.



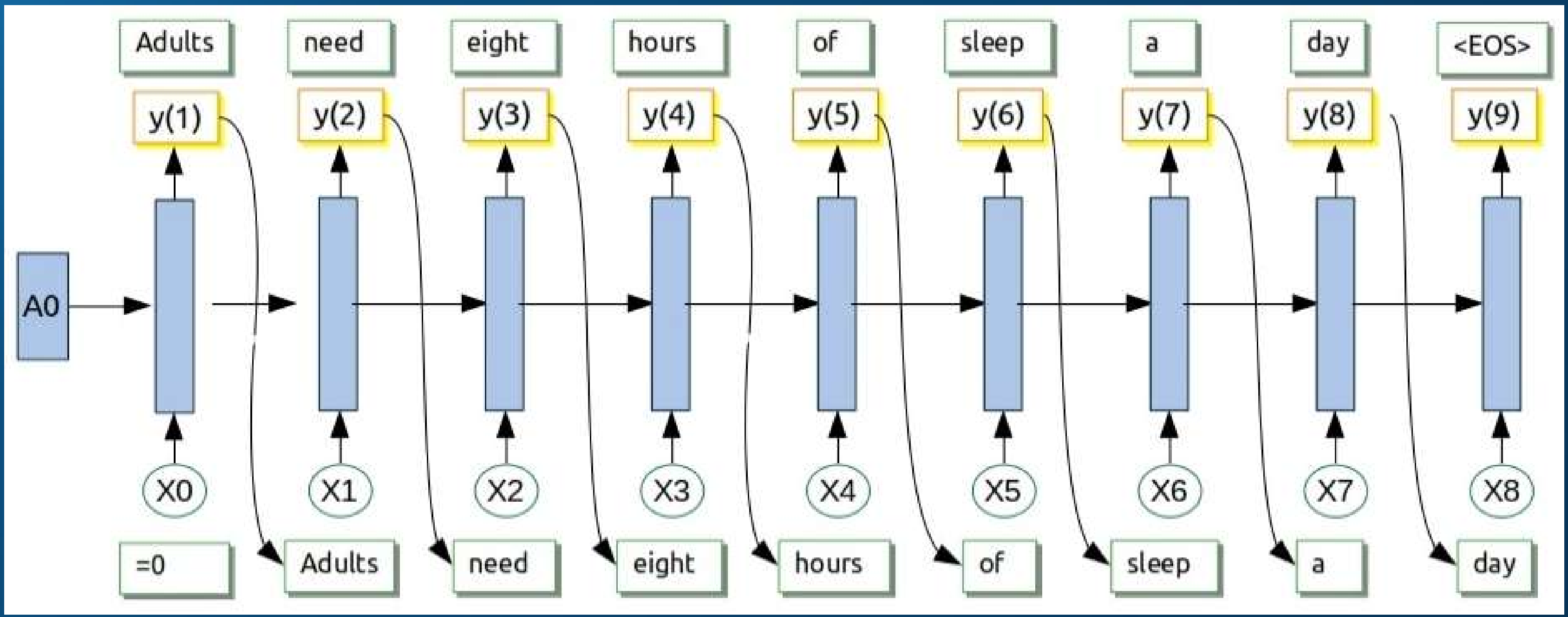
# RNN Model

- RNN is trying to predict one word at a time from left to right.
- Given that we are going to use logits and subsequently softmax for loss function, our loss function will be
- $\ell(\hat{y}, y) = -y * \log(\hat{y})$ as $\hat{y}$ is very close to 0 for all other words
  - since its remaining part $[ (1 - y) * \log(1 - \hat{y}) ]$ is insignificantly small we can ignore it.



# RNN Model

- Thus for overall sentence, Cost will be
  - $J(\hat{y}, y) = \Sigma\, \ell\, (\hat{y}, y)$
  - $J(\hat{y}, y) = -\frac{1}{m} \Sigma\, y * \log(\hat{y})$
  - Which we will be minimizing.





pra-sâmi

# RNN Model

- Suppose you have sentence with 3 words
- You want to know probability of it being a sentence
- Given a sentence $y_1, y_2, y_3$
- P($y_1, y_2, y_3$) = P[$y_1$] * P[$y_2$| $y_1$] * P[$y_3$ | $y_1, y_2$]

# Word representation

- Vocabulary = [a, aakash, aamaan… to zulu, zyzzogeton]
  - Also referred as corpus
  - Two more tokens <UNK> and <EOS>

- Can be converted to one hot encoding

| Man | Women | King | Queen | Apple | Oranges |
|---|---|---|---|---|---|
| (5468) | (8701) | (4823) | (7157) | (56) | (7259) |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 0 |
| – | – | 1 | – | 0 | – |
| – | – | – | – | – | – |
| 1 | – | – | – | – | – |
| – | – | – | 1 | – | 1 |
| – | 1 | – | – | – | – |
| – | – | – | – | – | – |
| 0 | 0 | 0 | 0 | 0 | 0 |

This representation is treating words independently….



pra-sâmi

# Featured Representation

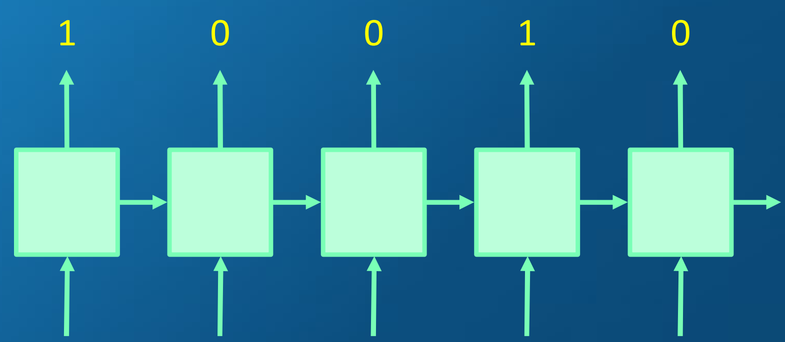
| | Man (5468) | Women (8701) | King (4823) | Queen (7157) | Apple (56) | Oranges (7259) |
|---|---|---|---|---|---|---|
| **Gender** | -1 | 1 | -0.95 | 0.97 | 0 | 0.001 |
| **Royal** | 0.01 | 0.02 | 0.90 | 0.98 | 0.05 | -0.01 |
| **Age** | 0.05 | 0.02 | 0.7 | 0.68 | 0.001 | -0.4 |
| **Food** | 0.001 | 0.002 | 0.0001 | 0.0002 | 0.95 | 0.90 |

Feature representing a huge corpus can drastically be reduced…

- Man → Women ≈ King → ????

- In terms of algorithm, we can use this using Similarity Coefficients
  - Find a word W : argmax ( $e_w, \; e_{king} - e_{man} + e_{women}$ )
  - Cosine sim ( u, v ) $= \frac{(u^T.v)}{||u||_2.||v||_2}$
  - Euclidian distances or Manhattan distances can also be used

# Named Entity and Word Embedding




| | | | | |
|---|---|---|---|---|
| Mohan | is | a | wheat | farmer |
| Sudesh | is | a | mango | ______ |
| Ravi | is | a | sarbati | cultivator |


Words → Embedding Layer → Dense Layer $W_d, b_d$ → Softmax Layer $W_o, b_0$

5 words 5 x 300 10000 probabilities

# Sampling a Sequence from a Well Trained Model

- Imagine we have super trained RNN network
- We ask it to predict first word,
  - which results in probability words in corpus to be first word,
- Pick a word from the probabilities to be first word (np.random.choice())
- Enter this word as input to timestamp ‘2’ to generate second word, again pick a word at random and pass it to third time stamp.
- and you will generate a sentence till you reach a <EOS>
- Alternatively, you can limit the sentence to say 20 words

- Voila!!!

- Remember 2016 US Election, someone fabricated how Trump would have answered questions during press conference
- Obviously it would not make exact sense. But in general it will be same.

# RNN Model

- In some cased, it is advantageous to have character based RNN instead of word based RNN.

- Both formats have their own advantages.

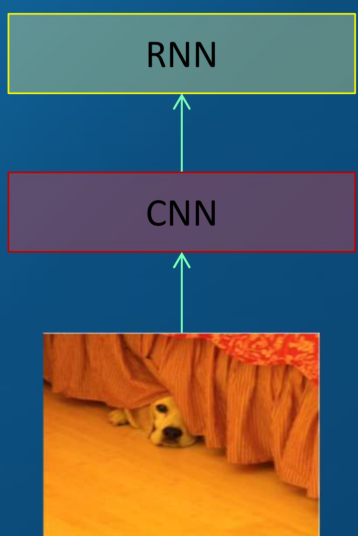
# Sequence to sequence : Image Captioning

- Given an image, produce a sentence describing its contents

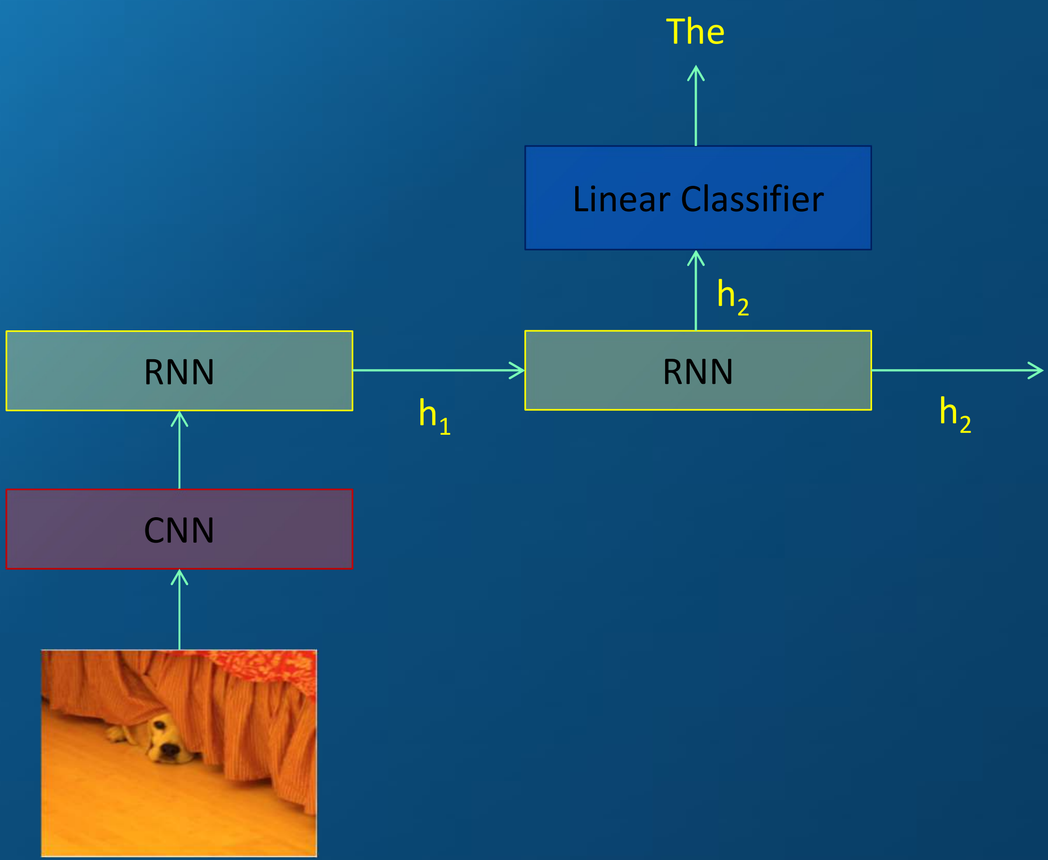
- Inputs: Image feature (from a CNN)
- Outputs: Multiple words (let’s consider one sentence)

: The dog is hiding

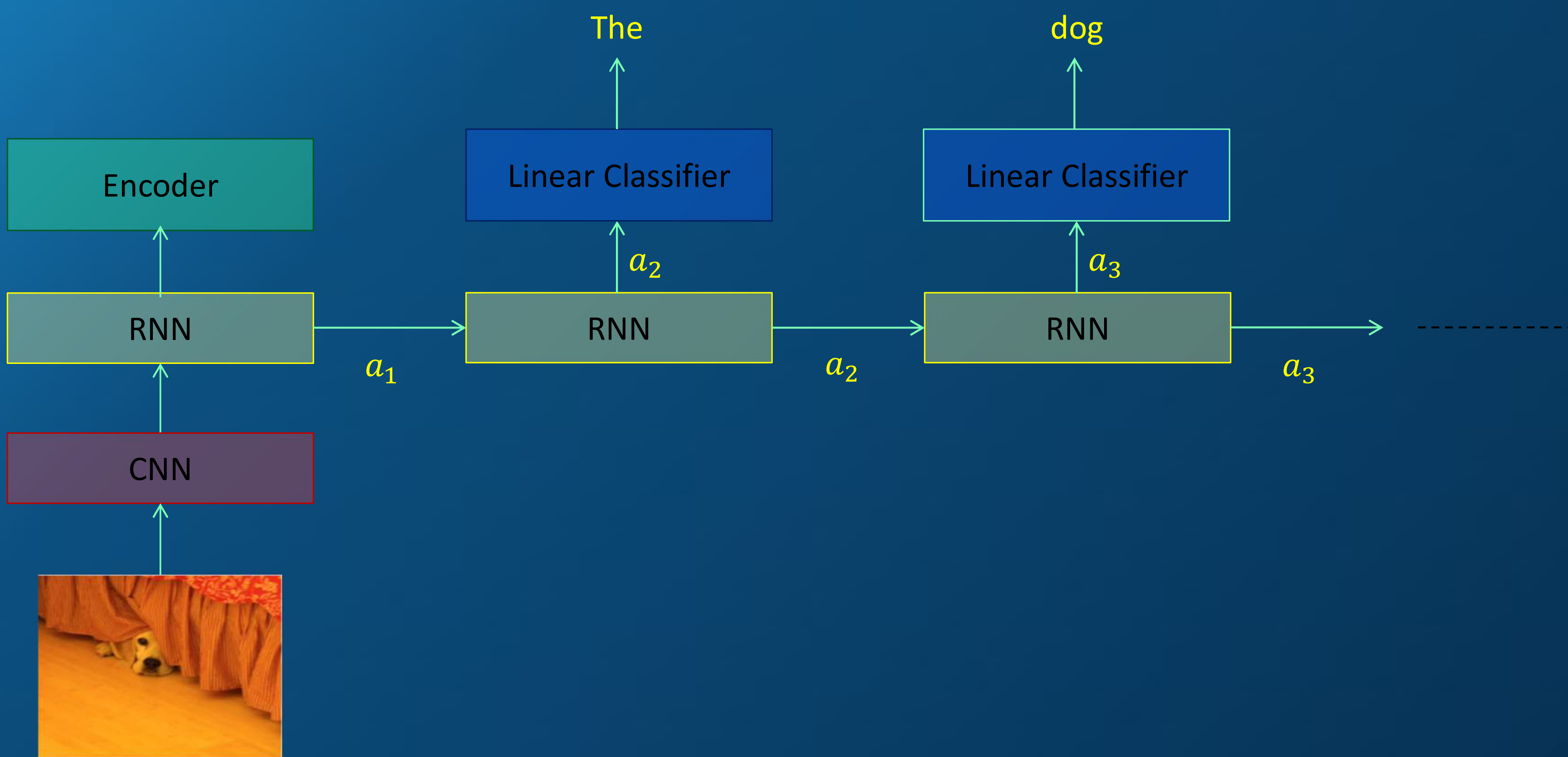
# Image Captioning



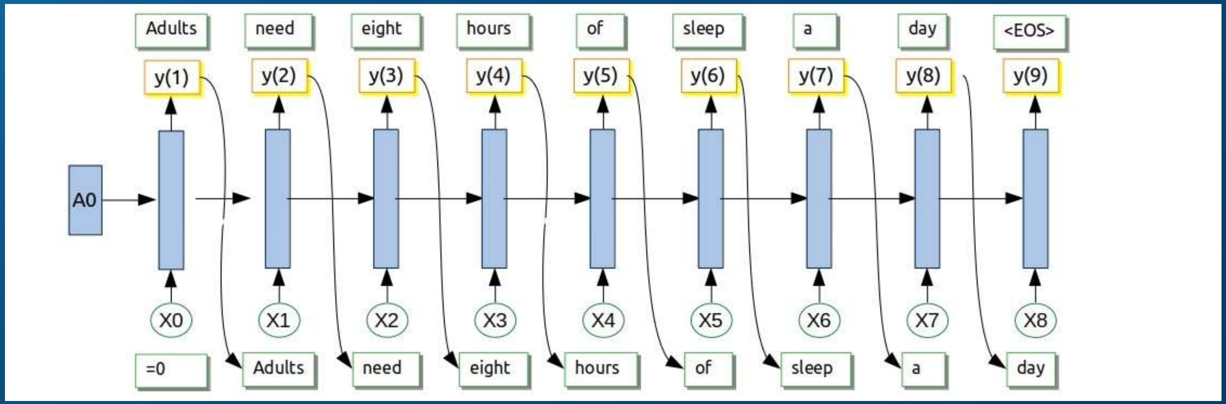
# Image Captioning



# Image Captioning

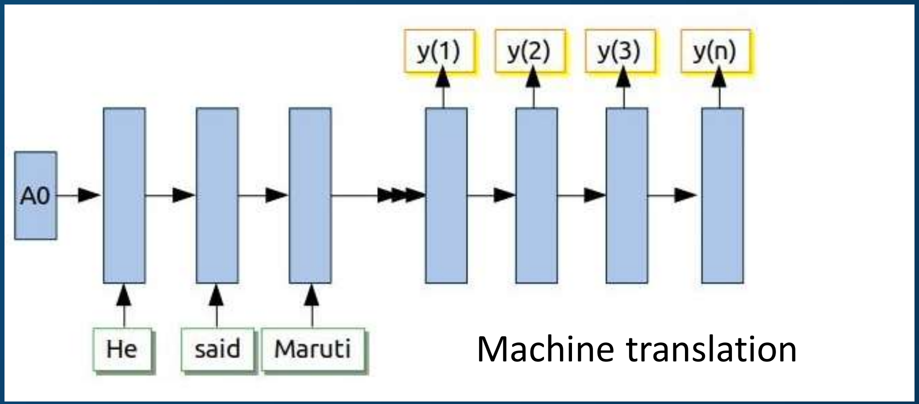


# Machine translation




Language Model

- Conditional language Model



# Sequence to sequence : Bleu Score

- ‘Dog’ , ‘bed’, ‘hiding’
- Le chien est sous le lit
- कुत्ता बिस्तर के नीचे है.
- कुत्रा पलंगाच्या खाली आहे.

: The dog is hiding

- Reference 1: The Dog is hiding under the bed
- Reference 2: There is a dog under the bed

- MT Output : The dog the dog hiding under the bed

“BLEU: a Method for Automatic Evaluation of Machine Translation” By Kishore Papineni, Salim Roukos, Todd Ward, Wei Jing Zhu.

# RNN Outputs: Image Captions


Show and Tell: A Neural Image Caption Generator, CVPR 15


THANK YOU