- One of the advantage of the skip-gram architecture is that it can handle rare word more effectively compare to CBOW.

Since - the skip-Gram predict multiple context words for a single target word, it can better capture the diverse context in which a word appears.

This property allow skip-Gram to generate more accurate word embedding, particularly for infrequent words. or words with limited occurence in the training data.

Part of Speech tagging (POS)

POS tagging is a task of lebelling each word in a sentence with its appropriate part of speech.

why	not	tell	someone	?
adverb	adverb	verb	noun	puntuation mark, sentence closer

POS is a preprocessing task or step

Uses

- Text Analysis
- machine Translation
- Named Entity Recognition (NER)
- Information Retrieval
- Speech Recognition
- Chatboat
- word sense disambiguation

I left the room

left of the room

Hidden Markov Models (HMMs): it is a sim

Ex

<S> $\overset{N}{\text{nitish}}$ $\overset{V}{\text{loves}}$ $\overset{N}{\text{campusx}}$ <E>

<S> $\overset{M}{\text{can}}$ $\overset{N}{\text{nitish}}$ $\overset{V}{\text{google}}$ $\overset{N}{\text{campusx}}$ <E>

<S> $\overset{M}{\text{will}}$ $\overset{N}{\text{Ankita}}$ $\overset{V}{\text{google}}$ $\overset{N}{\text{campusx}}$ <E>

<S> $\overset{N}{\text{Ankita}}$ $\overset{V}{\text{loves}}$ $\overset{N}{\text{will}}$ <E>

<S> $\overset{N}{\text{will}}$ $\overset{V}{\text{loves}}$ $\overset{N}{\text{google}}$ <E>

N - noun
V → verb
M - model

vocabulary	N	M	V
nitish	2/10	0	0
loves	0	0	3/5
campusx	3/10	0	0
can	0	1/2	0
google	1/10	0	2/5
will	0/10	1/2	0
ankita	2/10	0	0

← Emission probability →

① nitish as a noun do baar aaya hai

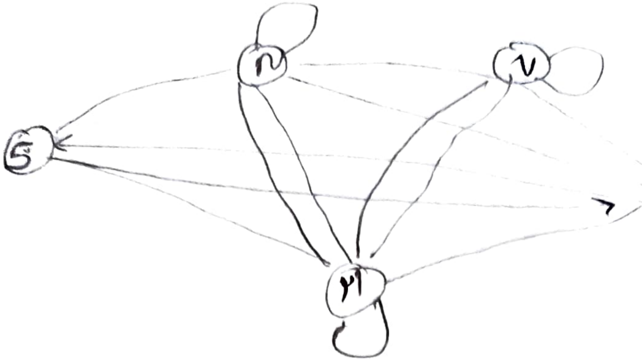
② loves 3 baar aaya hai, three baar as a verbs

google aik baar as a noun and 2 time as a verb aaya hai

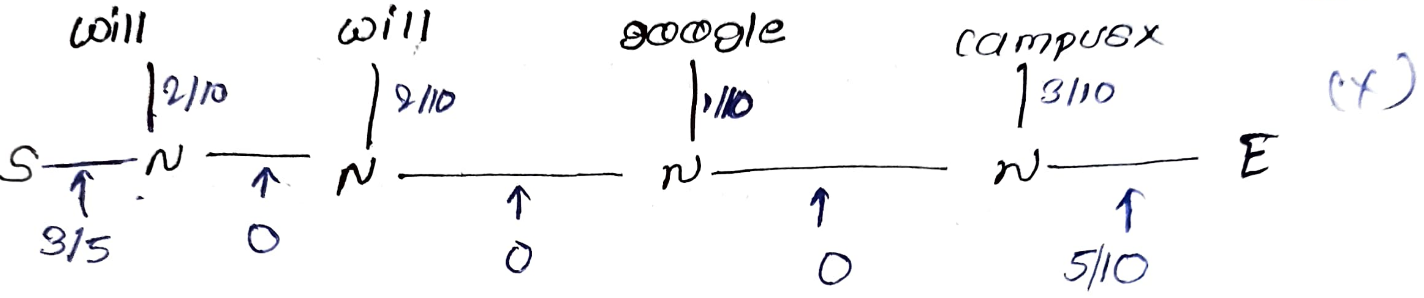
Total noun = 10
model = 2
verb = 5

— Transition Table —

	N	M	V	E	
S	3/5	2/5	0	0	→ 5 sum
N	0	0	5/10	5/10	→ 10 sum
M	2/2	0	0	0	→ 2
V	5/5	0	0	0	→ 5

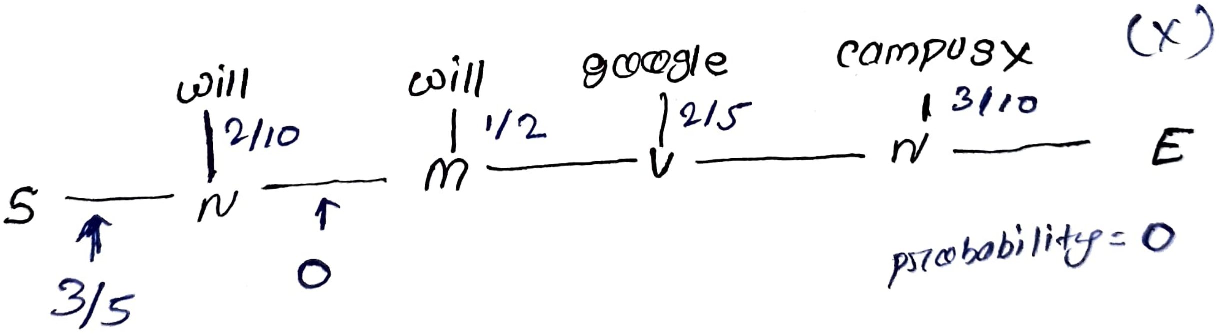


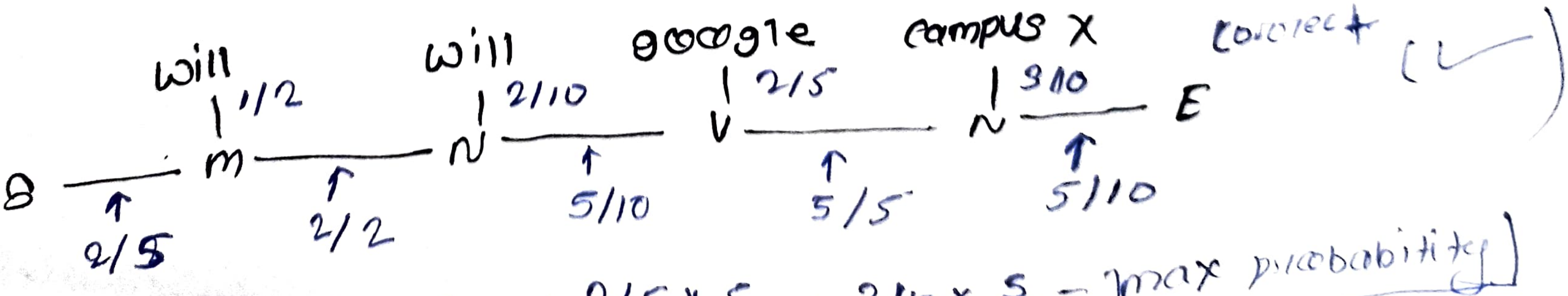
Let assum all word are noun



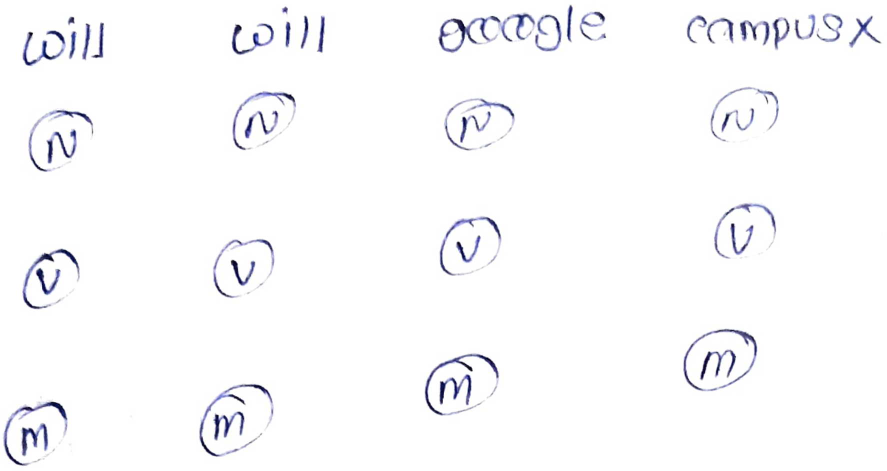
now multipy all and look for hiestest (bigest number amoung all posibility)

$= \frac{2}{10} \times \frac{3}{5} \times \frac{2}{10} \times 0 * \frac{1}{10} \times 0 * \frac{3}{10} \times \frac{5}{10} = 0$ (means not possible)



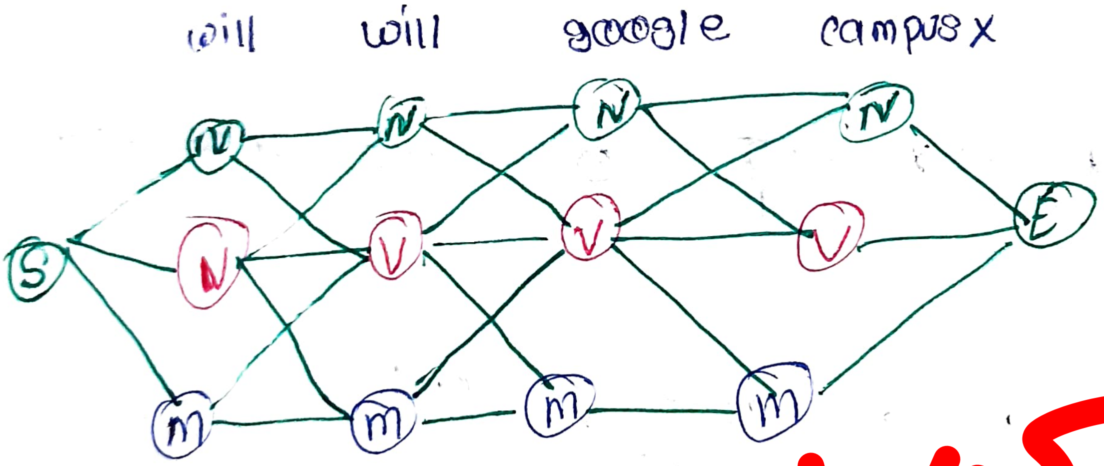


$= \frac{1}{2} \times \frac{2}{5} \times \frac{2}{10} \times \frac{2}{2} \times 2/5 \times \frac{5}{10} \times 3/10 \times \frac{5}{10} =$ [max probability]

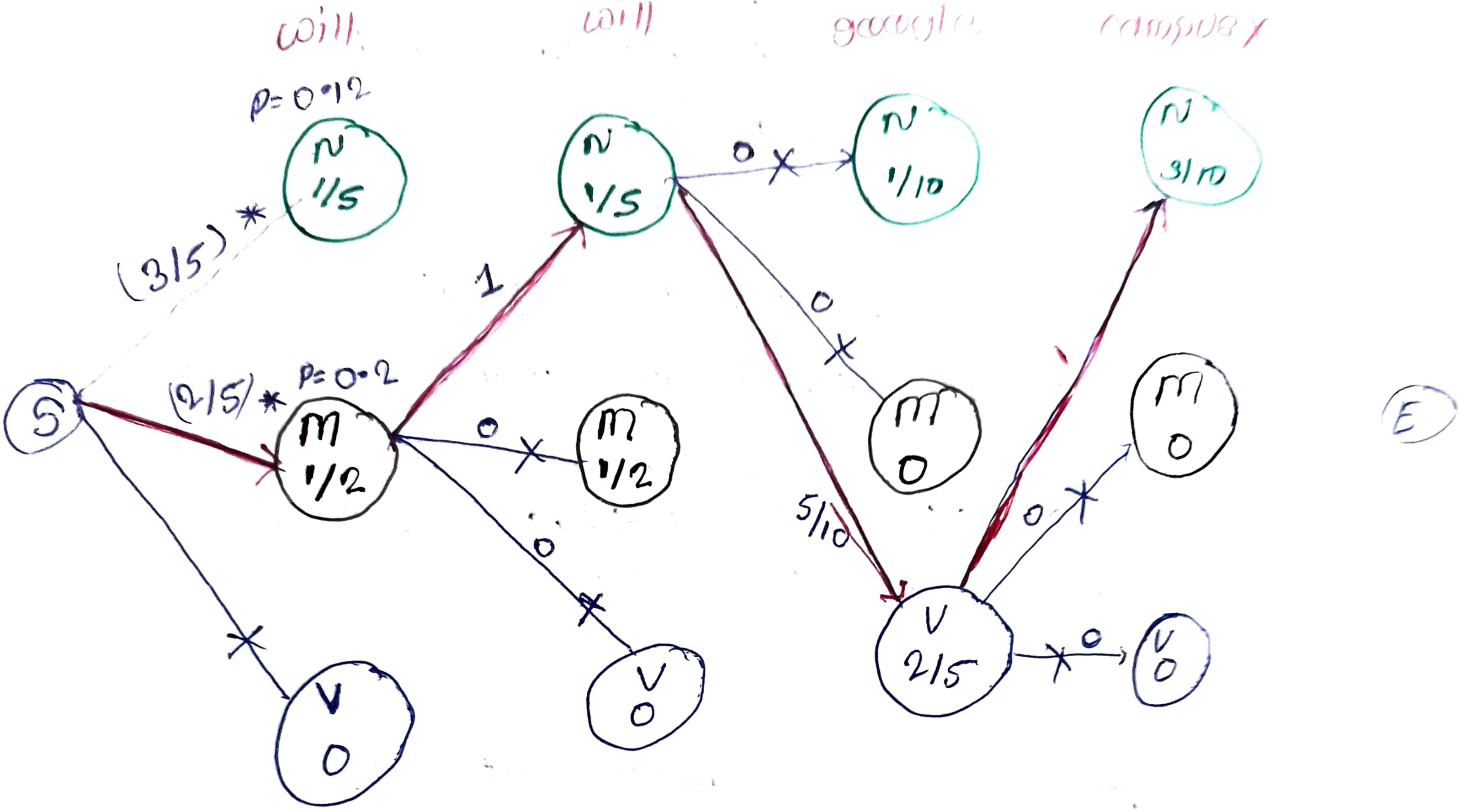


Total combination = 3^4 (length of sentance) part of speech used on sentance

So, it is brutforce method not good for large problem



Viterbi Algorithm



So,

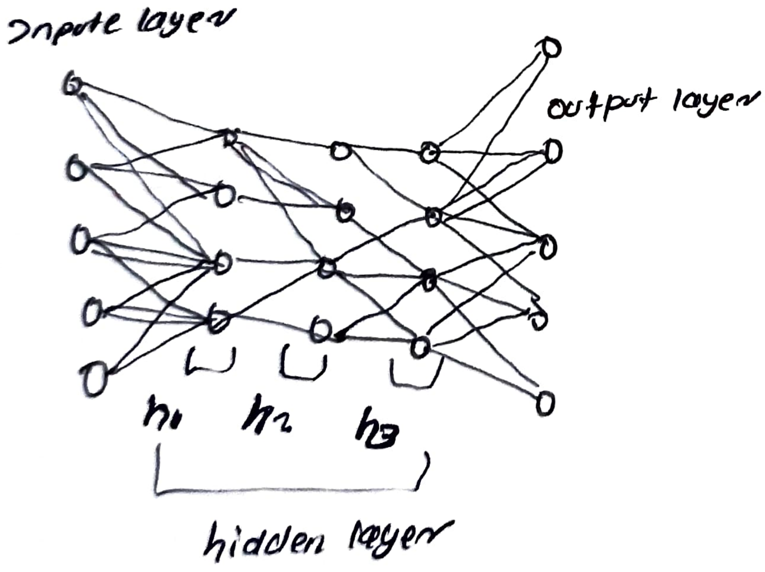
will	will	google	campusX
M	N	V	N

[deep Learing vs neural network]

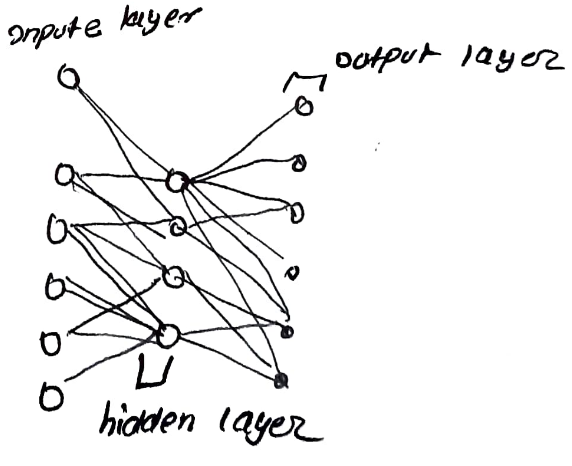
neural Network : A neural network is a computational model inspired by the structure and functionaling of biological neural network , such as human brain .

Deep learning : Deep learning refers to a specific approch within machine learning that involves training deep neural network . Deep learning focuses on neural network with multiple hidded layer (hence term deep) , allowing them to learn hierarchical representation of data .

deep learning



neural network



TRAX

Trax is an open-source deep learning library developed by Google Brain's team. It is primarily designed for training and evaluating neural networks, particularly sequence model.

Trax aims to provide a simple and efficient framework for building, training and deploying machine learning mode

Trax is build on JAX and Tensorflow.

Trax cambine the power of two deep learning libraries: JAX and Tensorflow. Jax provide the computational backend and high numerical operation, while tensorflow is used for various funtionality and utilities.

Trax take the advantage of GPU and TPU Hardware accelerator.

Trax focus on sequence model, such as recurrent network (RNNs), transformer, and other variants.

[advantage:]

- Runs fast on CPU, GPU and TPUs
- parallel computing
- Record algebraic computation for gradient evaluation

Recurrent Neural Networks

- It is a type of neural network architecture specifically designed to process sequential model data.

- It is a type of neural network where the output from the previous step is fed as input to the current step.

Rita was supposed to study with me. I called her but she did not ________

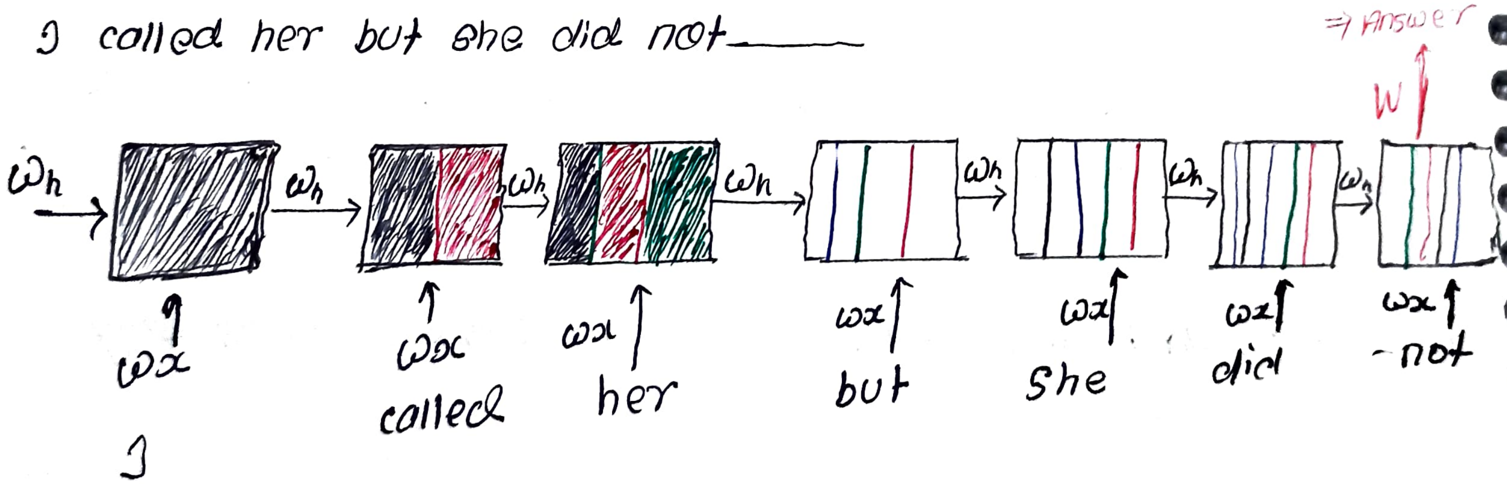
RNN look at every previous word

want
have
choose
ask
answer

→ Similar probabilities with trigram

answer ↓ according to RNN

I called her but she did not ________



#way to implement an RNN model

one to one: Given score of chap, you can predict the winner

one to many: given an image, we can predict the caption is going to be

~~many to many: Given tweet, we can predict the cap senti~~

many to one: Given tweet, we can predict sentiment

many to many: given an english sentence, we can translate in another

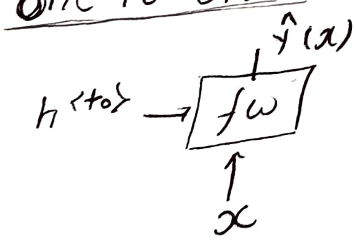
advantage of RNNs

- sequential model
- contextual understanding
- Flexible in input/output Length
- parameter sharing

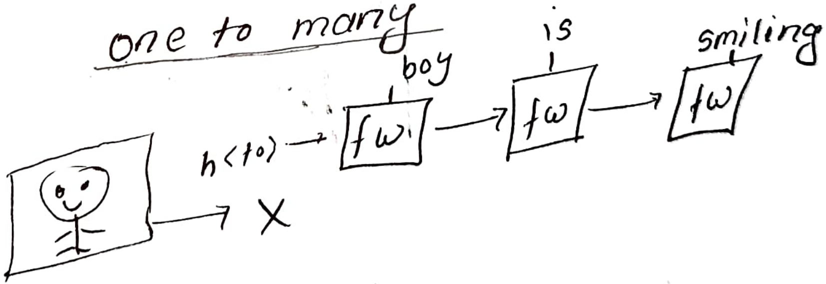
Application of RNNs

- NLP
- Speech Recognition
- Time series Analysis/Forecasting
- image caption: RNN can combine with CNN
- Music Generation

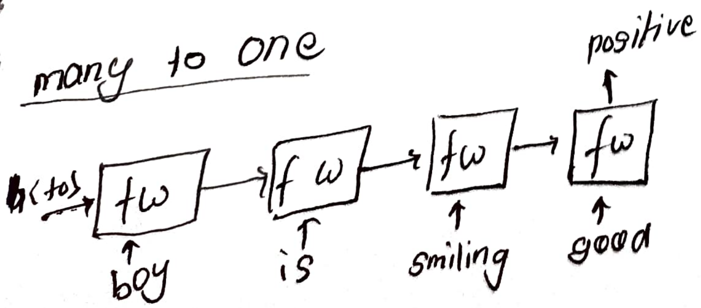
one to one



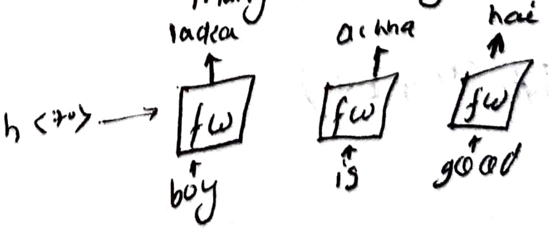
one to many



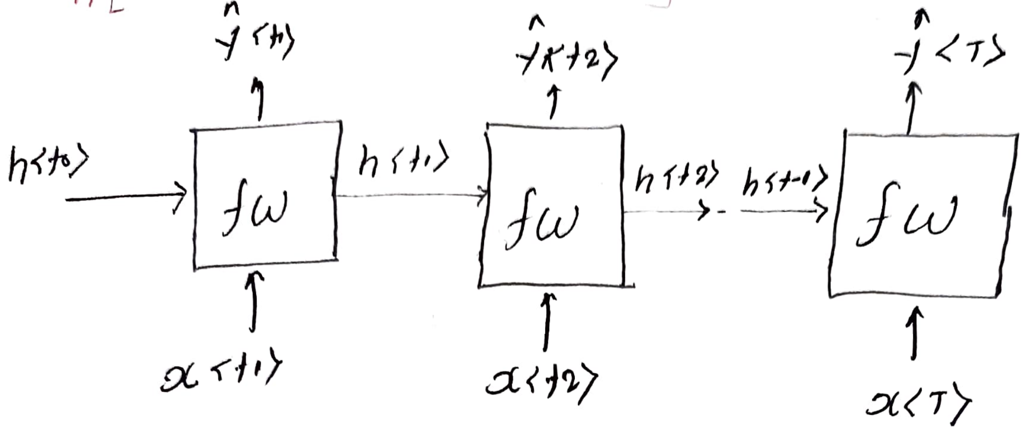
many to one



many to many



[Math in simple RNNs] [vanilla RNNs]



$$h^{<t>} = g(W_h[h^{<t-1>}; x^{<t>}] + b_h)$$

$$h^{<t>} = g(W_{hh} h^{<t-1>} \oplus W_{hx} x^{<t>} + b_h)$$

Whh → weight of recurrent neuron

Wxh → weight of inpute neuron

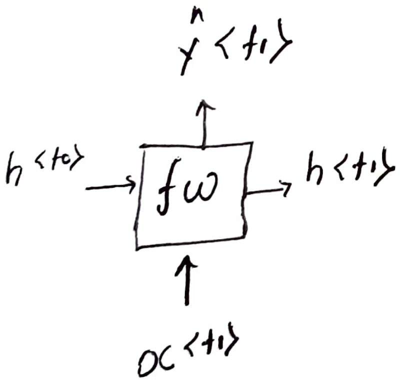
Formula for calculating current state

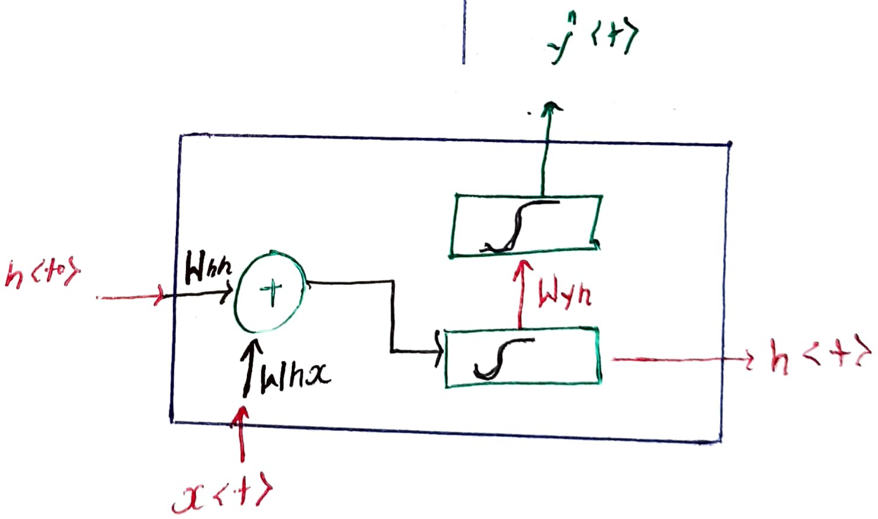
$$h_t = f(h_{t-1}, x_t)$$

h_t → current state

h_{t-1} → previous state

x_t → inpute state





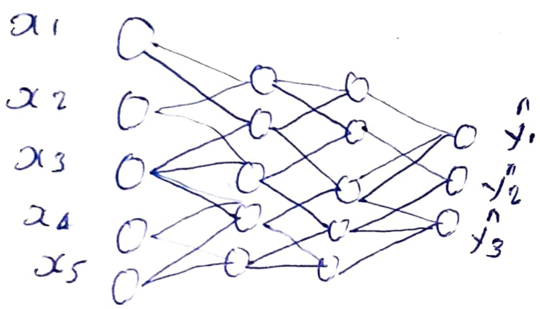
$$h^{<t>} = g(W_{hh} h^{<t-1>} \oplus W_{hx} x^{<t>} + b_h)$$

$$\hat{y}^{<t>} = g(W_{yh} h^{<t>} + b_y)$$

ould be the size of matrix Wh. if h^{<t>} had size 4x1 and x^<t> 10x1

a) 4x14 ✓
b) 14x14
c) 4x10

Cost Function for RNNs



The cost function used in RNN is the cross entropy loss

K - classes or possibilities

Either 1 or 0

$$Loss(J) = -\sum_{j=1}^{K} \boxed{y_j} \log \hat{y}_j$$

Looking at a single example (x,y)

Computing loss over several steps

$$\boxed{J = -\frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{K} y_j^{<t>} \log \hat{y}_j^{<t>}}$$

Vanilla RNN: Also know as simple RNN, refer to the basic and original form of the recurrent layer in deep learning.

- It suffer from vanishing problem

To address this vanishing gradient problem we use LSTM and GRU.

GRU (Gate Reccurent unit)

The Gate Reccurent unit (GRU) is a type of recurrent neural network (RNN) architecture that addresses the limitation of the traditional RNN, such as the vanishing gradient problem and difficulties in capturing long-term dependencies.

Three main component of GRU

1) update Gate (U): It decide how much of the previous hidden state should be kept and how much of the new information should be added to the current hidden state.

2) Reset Gate (r): It control how much of the previous hidden state should be forgotten.

3) candidate Hidden state (h~): It represents the new information that will be added to the hidden state.

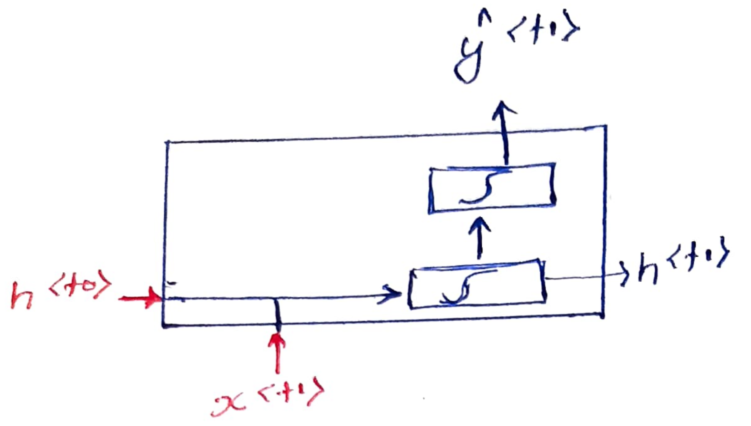
4) current Hidden state (h): It is the output of the GRU layer and serves as the hidden state for the next time step.

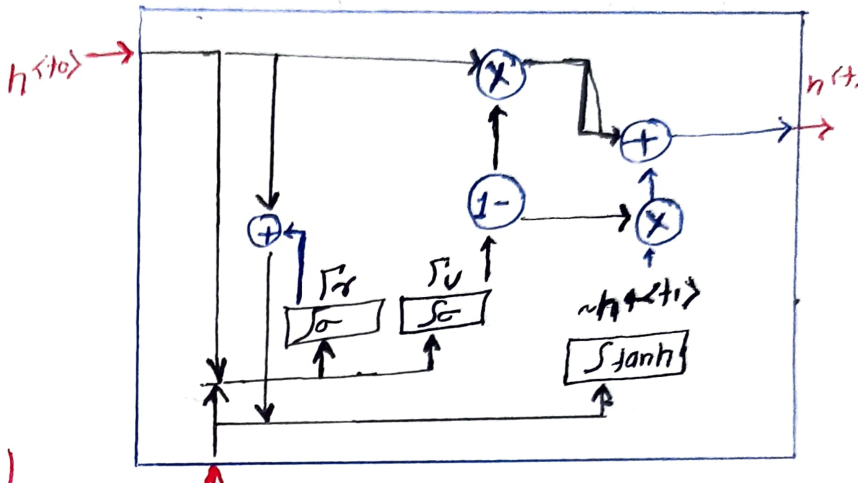
Gates to keep/update relevant information in the hidden state

$$\Gamma_r = \sigma(W_r[h^{<t0>}, x^{<t>}] + b_r)$$

$$\Gamma_u = \sigma(W_u[h^{<t0>}, x^{<t>}] + b_u)$$

Vanilla RNN vs GRU's





GRU

$$h^{<t>} = g(W_h[h^{<t-1>}, x^{<t>}] + b_h)$$

$$\hat{y}^{<t>} = g(W_y h^{<t>} + b_y)$$

Gates to keep/update relavent information ↑ in the hidden state

Forget gate → $\Gamma_r = \sigma(W_r[h^{<t_0>}, x^{<t>}] + b_r)$

Update gate → $\Gamma_u = \sigma(W_u[h^{<t_0>}, x^{<t>}] + b_u)$

=

hidden state candidate → $\tilde{h}^{<t>} = \tanh(W_h[\Gamma_r * h^{<t_0>}, x^{<t>}] + b_n)$

~~update gate~~ current hidden state $h^{<t>} = (1 - \Gamma_u) * h^{<t_0>} + \Gamma_u * \tilde{h}^{<t>}$

$$\hat{y}^{<t>} = g(W_y h^{<t>} + b_y)$$

LSTM (Long Short-Term Memory)

Long Short-Term memory is a type of recurrent neural network architecture that address the limitation of traditional RNNs in capturing and preserving long-term dependencies.

LSTM are designed to overcome vanishing and exploding problem.

i) Memory cell: The memory cell is responsible for storing and carrying information across time steps.

ii) Input Gate
iii) Forget gate
iv) output gate

Similar to GRU

application
- Next-character prediction
- Chatbots
- Image captioning
- Speech recognition
- music composition

GRU + Memory cell ⇒ LSTM (we can say that)

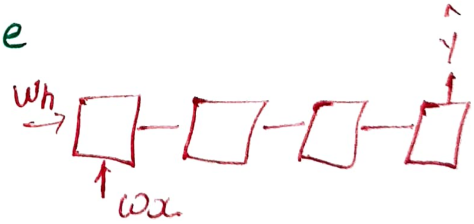
How vanishing and Exploding Gradient happen

→ Back propagation through time

$$\frac{\partial L}{\partial W_h} \propto \boxed{\sum_{1 \le k \le t} \left(\boxed{\prod_{t \ge i > k} \frac{\partial h_i}{\partial h_{i-1}}} \right) \frac{\partial h_k}{\partial W_h}}$$

→ contribution of hidden state k

Length of the product proportional to how far k is from t



Partial derivative < 1 contribution goes to 0 → vanishing Gradient

---

partial derivative > 1 contribution goes to infinity → Exploding Gradient

Solution to vanishing Gradient problems

1) Identity RNN with Relu activation

$$\begin{bmatrix} 1 & -1.0 & -0.01 \\ -0.1 & 1 & -0.1 \\ 0 & 0 & -0.2 \\ 1.1 & 1 & 1 \end{bmatrix} \xrightarrow{\text{ReLU activation}} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad -1 \to 0$$

$$\text{ReLU}(x) = \max(0, x)$$

2) Gradient clipping: Gradient clipping is a techique when the gradient are clipped or capped at a certain theresold during back propogation. This prevent the gradient from exploding and help to stabilize training.

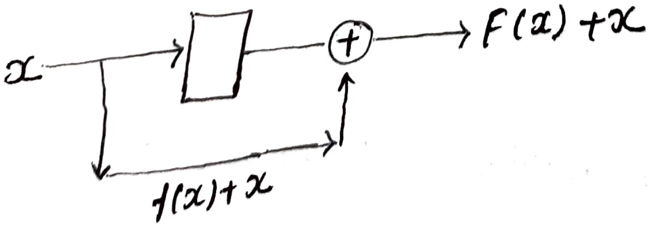
ex, theresold set = 25

then 32 → 25

46 → 25

6 → 6

3) Skip connections (Skipp activation function)



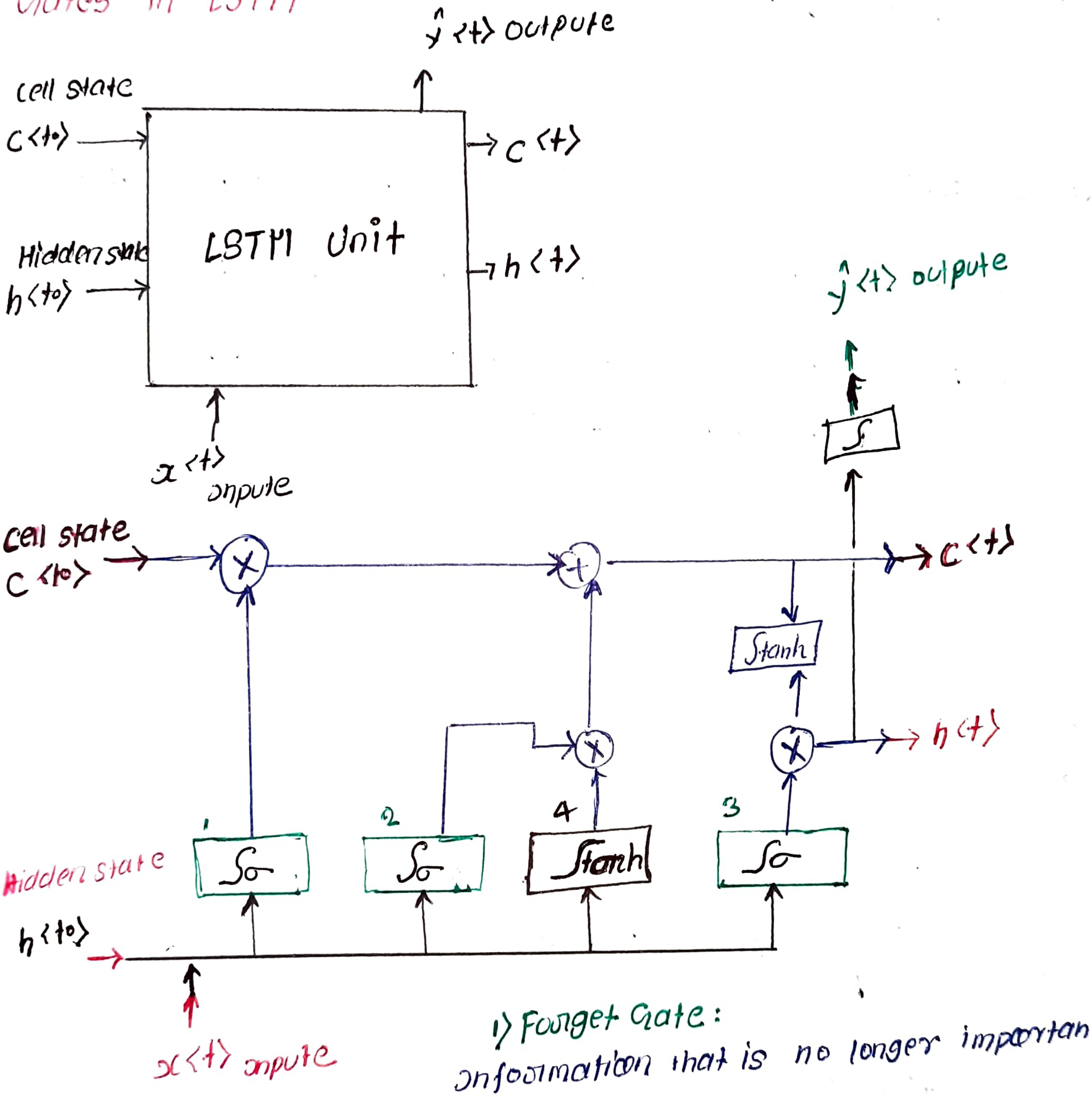
(Q) order of gate that information flows through in an LSTM unit.

a> Forget → outpute → inpute gate

b> inpute → Forget → outpute gate

c> Forget → inpute → output gate ✓

Gates in LSTM



1> Forget Gate:
information that is no longer important

2> inpute Gate
information to be stored

3> outpute gate
information to use at current step

Sigmoid output between 0 and 1

0 → close gate (information does not get through)

1 → Information get through freely

4

Stanh → **candidate cell state**

Information from the previous hidden state and current inpute

tanh shrinks argument to be between -1 and 1

New cell state

add information from the candidate cell state using forget and inpute Gate.

Summary

LSTM use a series of gates to decide which information to keep:

- Forget gate decide what to keep and what to forget
- Inpute gate decides what to add
- outpute gate decides what the next hidden state will be

Name Entity Recognition (NER)

is a NLP task that involves identifying and classify name entities in text into predefined categories such as person name, location, date, time etc.

Ex $\underbrace{\text{pawan}}_{\text{person}}$ is going To see $\underbrace{\text{Taj Mahan}}_{\text{location}}$ $\underbrace{\text{tomorrow}}\ \underbrace{\text{Sunday}}_{\text{date}}$ at $\underbrace{\text{8 pm}}_{\text{time}}$

Application

- Search engine efficiency
- Recommendation engine

- customer service
- Automatic trading

Training NERFS : Data processing

- convert word and Entity classes into arrays :
- pad with tokens
- create a data generator

Training on NER

- create a tensor for each inpute
- put them into bach (32, 64, 128 --)
- Feed into LSTM unit
- Run the outpute through a dense layer
- predict using log softmax over k classes

—END of week—

Siamese network

A siamese network is a type of neural network architecture that are designed to compare and measure similarity or, dissimilarity between two inpute sample.

Ex, How old ary you = what is your age

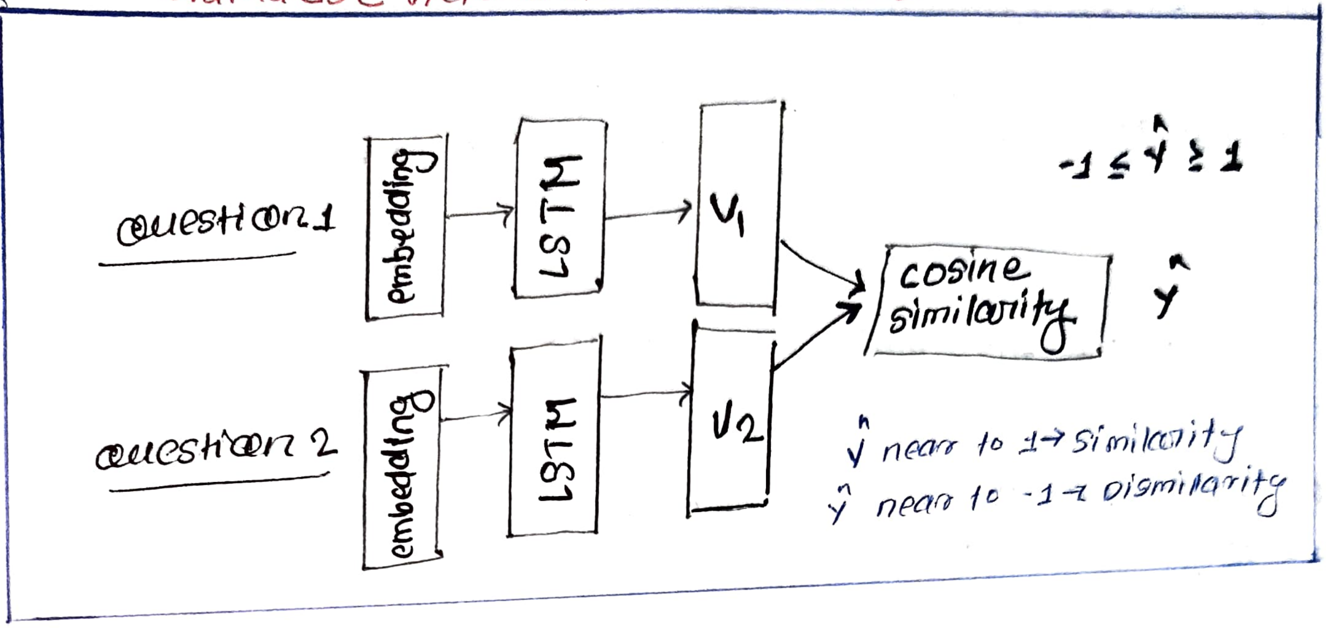
where are you from ≠ where are you going

Application

- Handwritten checks
- question duplicates

  How old are you ?
  what is your age ?
- queries

Siamaese network Architecture



1) Embedding
2) LSTM
3) vectors
4) cosine similarity

The architecture of a siamese network consists of two or more identical subnetworks, referred as 'twin' or 'siamese twin' share the same set of weight and parameters.

$$\cos(V_1, V_2) = \frac{V_1 \cdot V_2}{|V_1| \times |V_2|}$$

cosine similarity

Cost Function

How old are you? Anchor

What is your age? positive

Where are you from? negative

$\cos(\overset{\text{Anchor}}{A}, P) \approx 1$

$\cos(A, \underset{\text{negative}}{N}) \approx -1$

for good model

$$\boxed{\text{Loss} = \cos(A, N) - \cos(A, P) \leq 0}$$

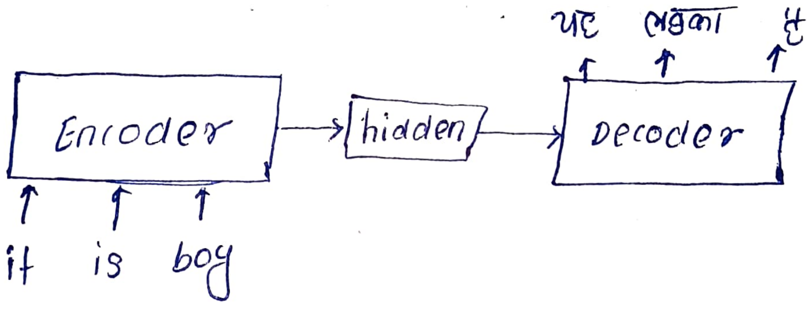
one short learning

The key reason for one short learning is to be able to classify new classes without retraining any models.

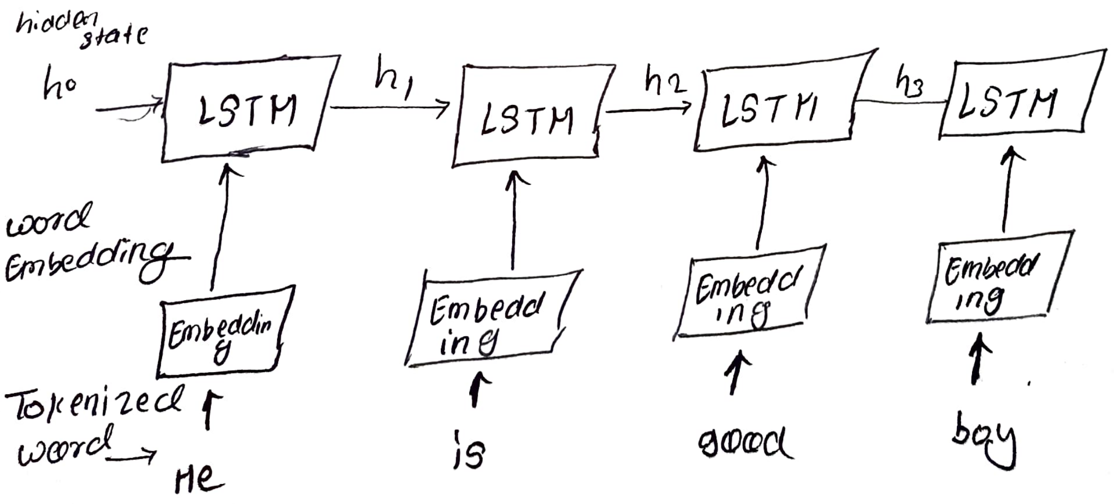
Sec2Sec Model

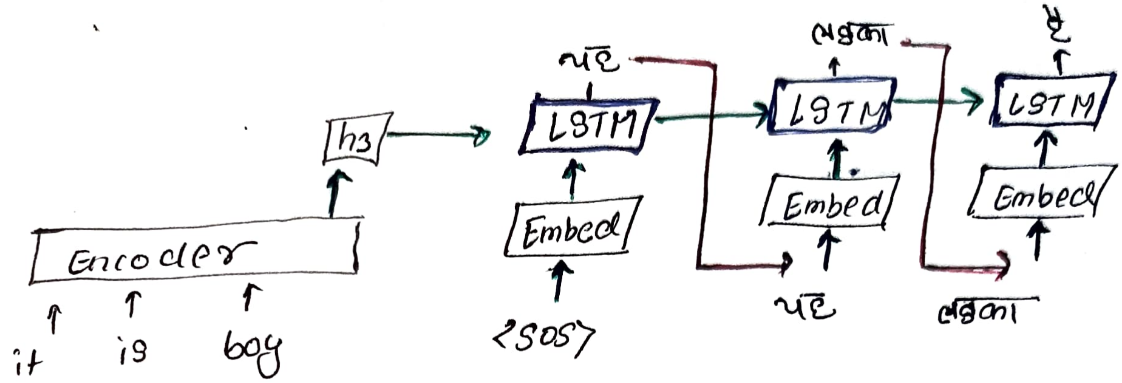
Sequence to sequence model commonly used in task like machine traslation.

- The model consist of two main component ; an encoder and decoder.
- The **encoder** processes the inpute sequence and convert it into fixed length vector called context vector
- The **decoder** takes the context vector as inpute and generate the outpute sequence step by step



seq2sec encoder

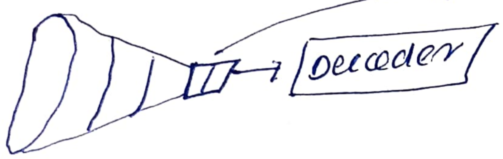




problem

information is bottle neck

→ Fixed hidden state size



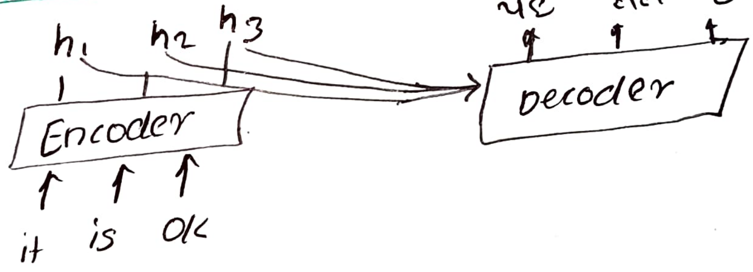
- A fixed amount of information goes to the decoder

(Q) why are longer sequences problematic for traditional seq2seq model

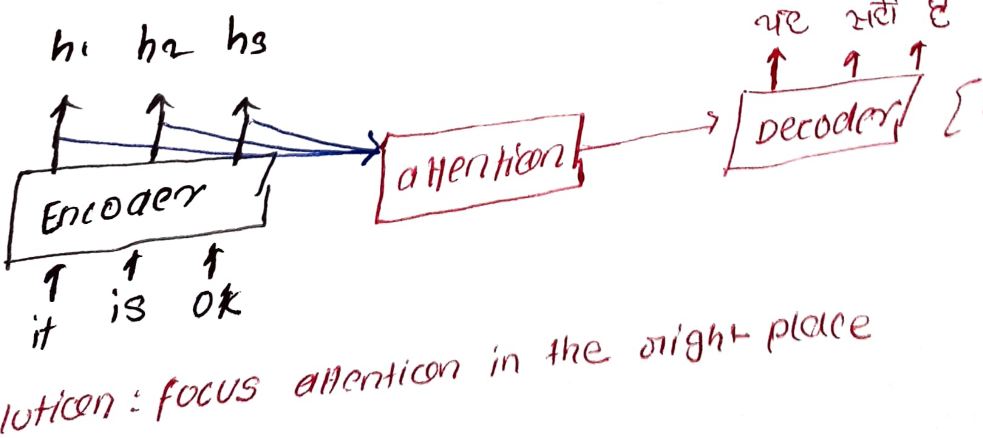
Ans Because sec2eq relies on fixed-length memory

- As sequence size increase, model performance decreases

Solution



[Traditional]

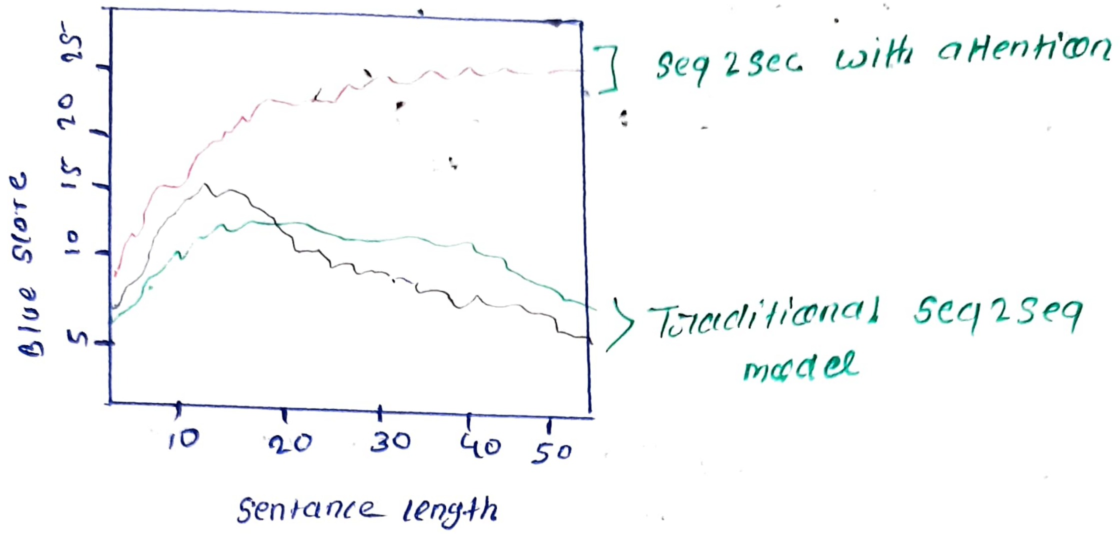


[modern solution]

or
Solution : focus attention in the right place

Performance

0503



In traditional model performace is decrease as word increasing.

- Attention is a layer that lets a model focus on what important
- Queries, values, and key are used for information retrivel inside the Attention layer.

Blue score (preision)

Blue score (Bilingual Evaluation understanding) is a matrix commonly used to evaluate of the quality of machine translation output by comparing one or more references translation.

The Blue score measure the similarity between the machine-generated translation and the reference translation, assigning a score between 0 and 1.

A higher BLUE score indicate a better match between the machine translation and the reference translation.

candidate = [I I am I] — 4

Reference 1 = Younes said I am hungry

reference 2 = He said I am hungry

First = I present in both sen (1)
second = I " " " (1)
third = am " " " (1)
Fourth = I " " " (1)

Count - Total = 4

Bule score = $\frac{1+1+1+1}{4} = 1$

Blue score (modified)

candidate I I am I

reference 1 Younes said ~~I~~ ~~am~~ hungry

reference 2 He said ~~I~~ ~~am~~ hungry

Bule score = $\frac{1+1}{4} = 0.5$

First: I present in both count (1), then delete I from both

seccon = I not present in both (0)

third : am present in both count (1) then delete it

forth = I not present

$= \frac{2}{4}$ ←