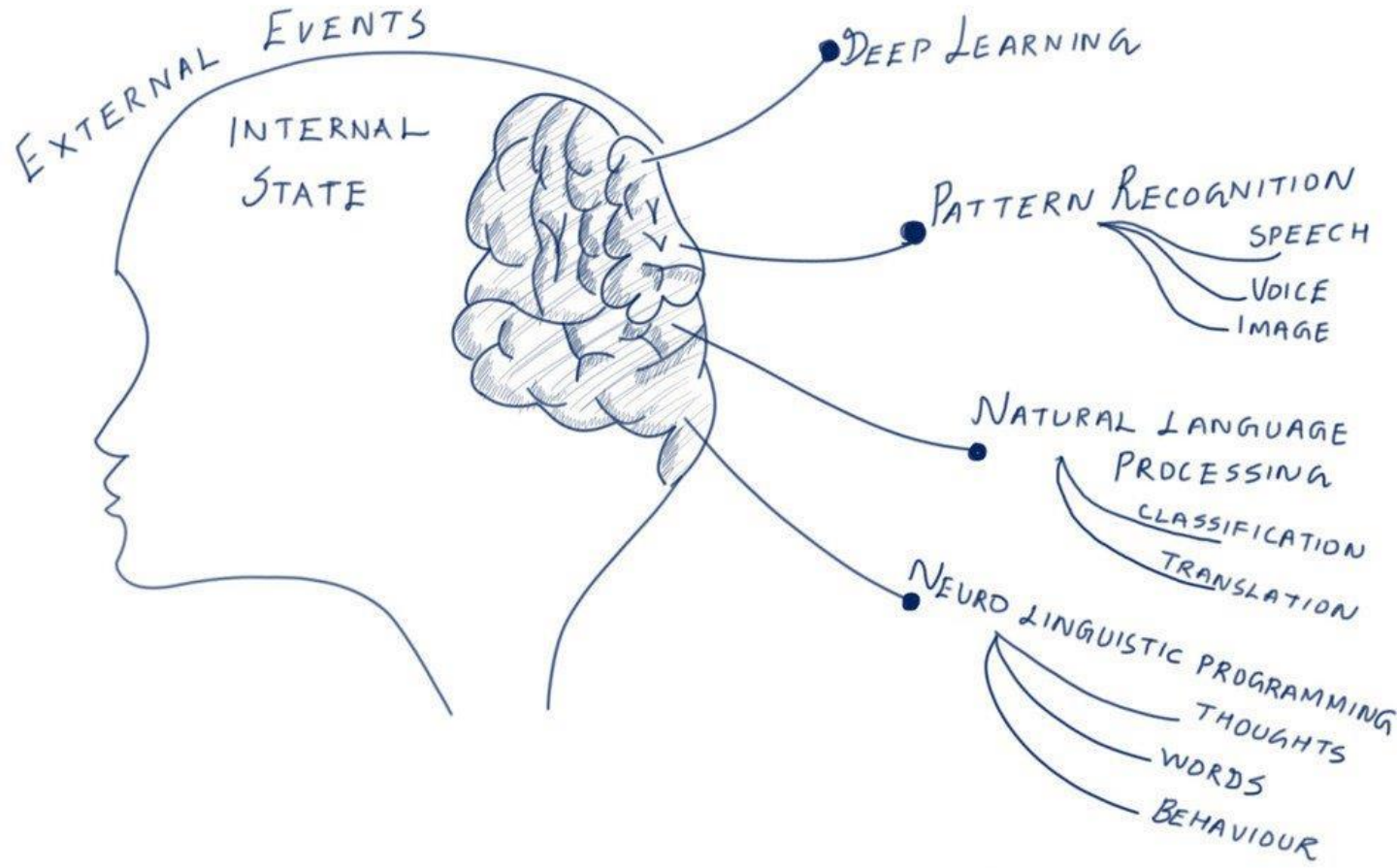
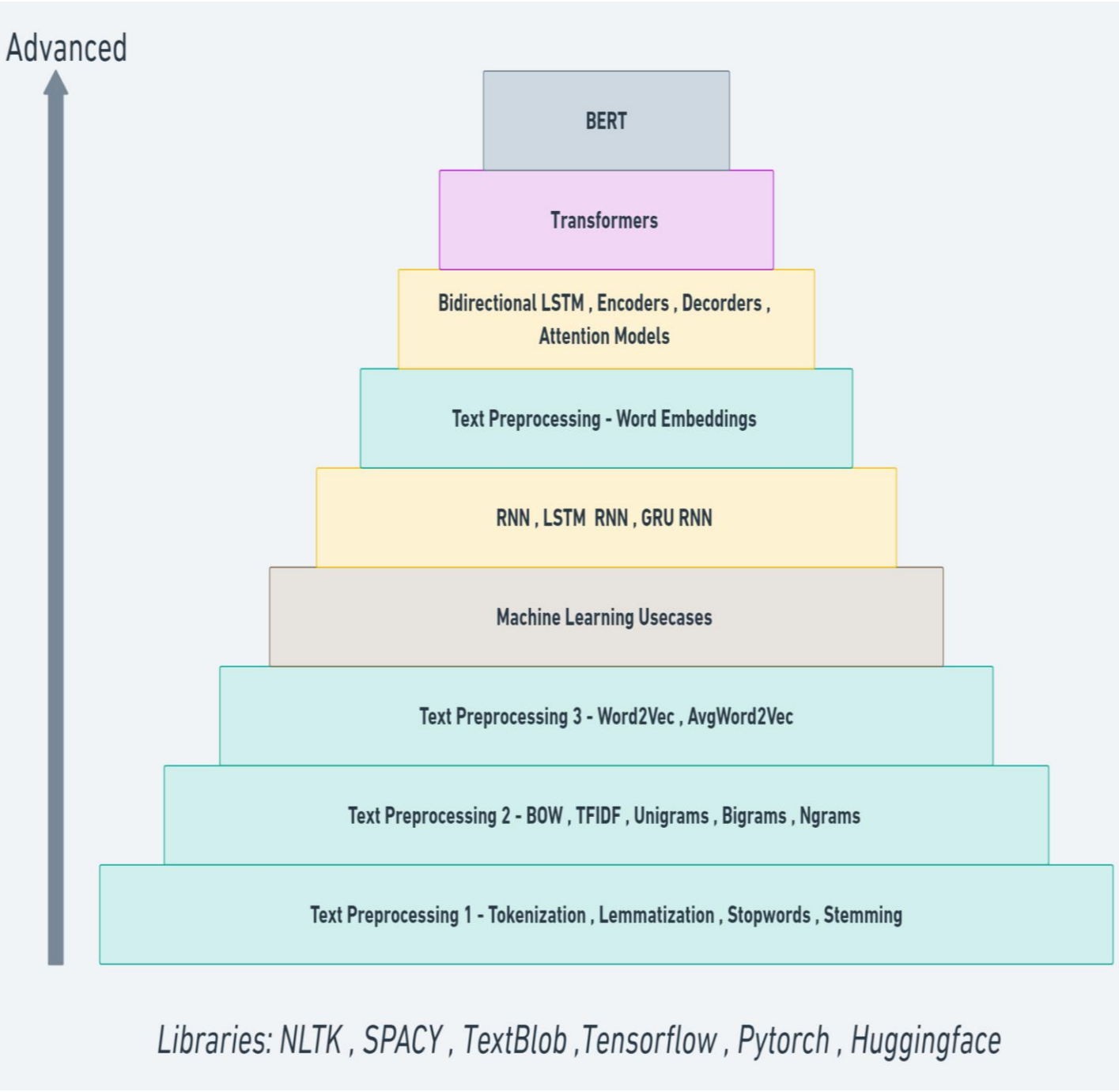


NATURAL LANGUAGE PROCESSING

ANNA ELSA LUIZ


Advanced
BERT
Transformers
Bidirectional LSTM , Encoders , Decorders , Attention Models
Text Preprocessing - Word Embeddings
RNN , LSTM RNN , GRU RNN
Machine Learning Usecases
Text Preprocessing 3 - Word2Vec , AvgWord2Vec
Text Preprocessing 2 - BOW , TFIDF , Unigrams , Bigrams , Ngrams
Text Preprocessing 1 - Tokenization , Lemmatization , Stopwords , Stemming
Libraries: NLTK , SPACY , TextBlob ,Tensorflow , Pytorch , Huggingface

NLP

Day 1

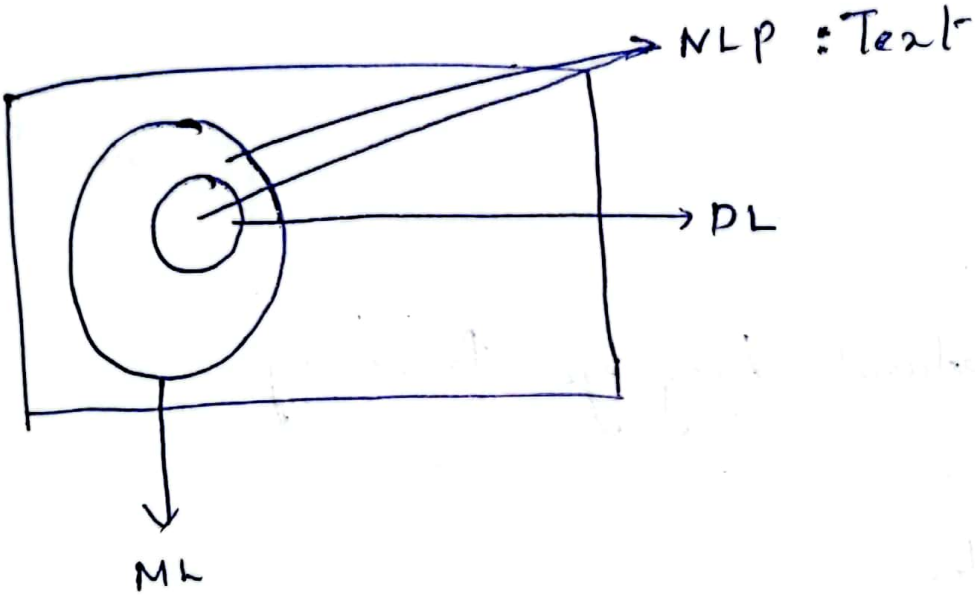
Agenda:

1. Roadmap of Natural Language Processing
2. Why NLP
3. Lots of Examples
4. Tokenization, Stemming, Lemmatization
5. Bag of Words.

Prerequisites

1) Python.
2) Stats
3) ML
4) ANN, optimisers, Loss functions

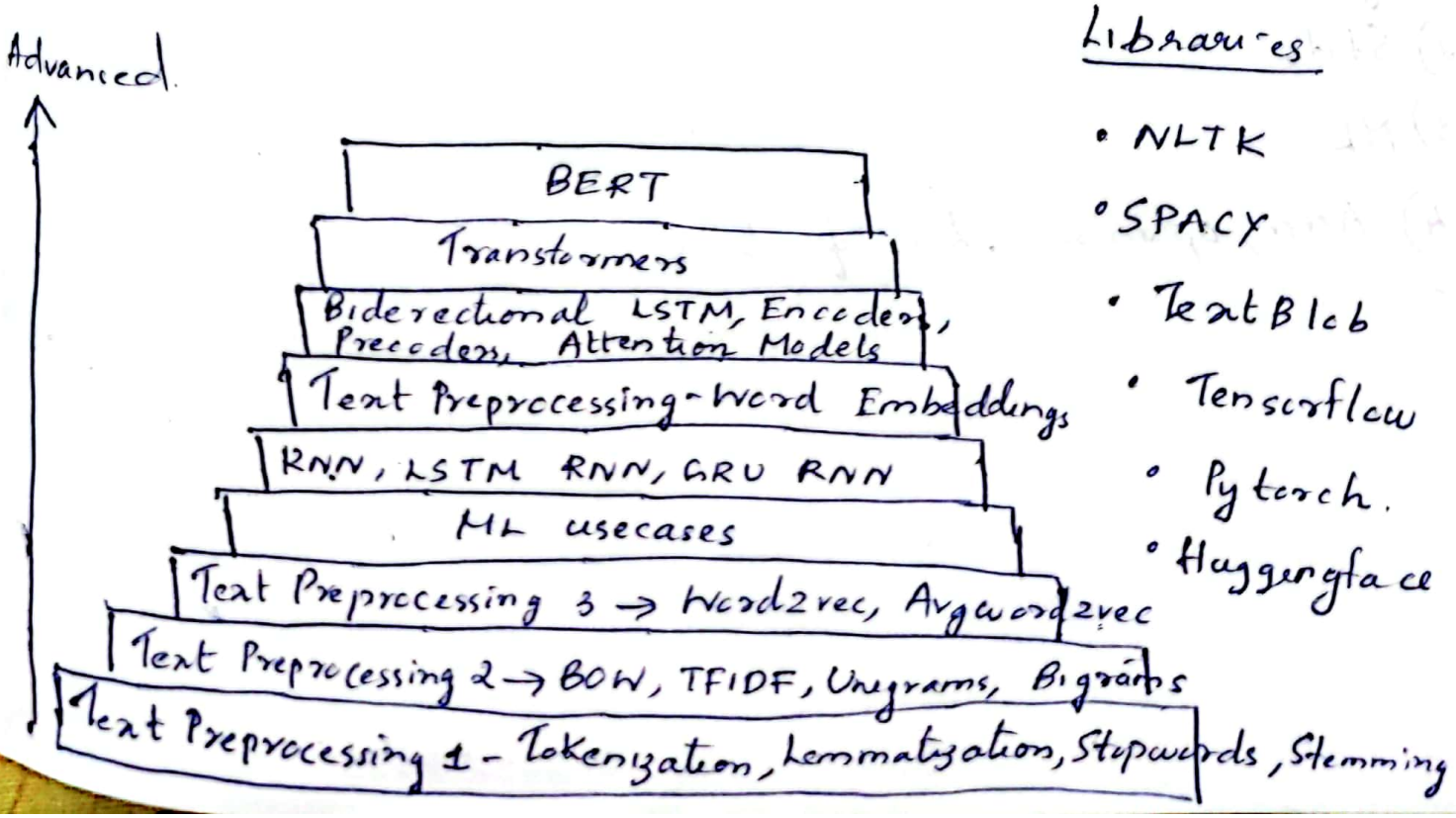
Why NLP?



How machine gonna understand our language?

So we need NLP.

RoadMap of NLP



Libraries

- NLTK
- SPACY
- TextBlob
- Tensorflow
- Pytorch.
- Huggingface.

Text Preprocessing

Text may be in the form of Paragraphs, sentences, we should convert this into specific format specifically called as vectors.

1. Tokenization

ML Usecase; Spam classifier.

Dataset.

~~Dat~~ I/p f_1	I/p f_2	Spam/ham
Email-body	E-mail Subject	O/p
1) You won 1000000 $$	Billionaire	Spam
2) Hey Anna, How are you?	Hello	Ham
3) Credit Cards Worth	Winner	Spam.

① Tokenization ⟶ Stopwords ② Stemming ⟶ ③ Lemmatization

Tokenization: Converting sentence into words.

eg: Hey Buddy I want ~~to~~ go ~~to~~ your house

Not Important

"Not" Keyword can be important.

In order to remove the words we apply stopwords.

We can use our own stopwords.

Stemming: We try to find out the base stem of the word.

historical
history → Historic Word Stem/Root word/Base form.

It is a process of reducing words to their base word stem. It has a disadvantage that it has no meaning.

finally
final
finalized. } fina (meaningless)

going
goes
gone } go (meaningful)

advantages

1. It is really fast

Disadvantages

1. It is removing the meaning of the word.

Lemmatization

Text pre-processing technique in ~~natural~~ NLP that breaks down a word into it's root meaning, also known as Lemma.

history
historical > history ...

finally
final
finalized } final ..

Advantages

1. Meaningfull words

Disadvantages

1. It's slow.

Usecases of Stemming

1. Spam Classification
2. Review Classification
3.

Use cases of Lemmatization

1. Text Summarization
2. Language Translation
3. Chatbot.

Text Preprocessing

Step 1

1) Tokenization ② Stopwords ③ Stemming ④ Lemmatization

Step 2

Now we should convert the words into vectors

1. Bag of Words 2. TF IDF 3. Word2vec.

Day 2

Agenda

1) Text Preprocessing → words → vectors

a) One Hot Encoding (OHE)

b) Bag of words (BOW)

c) TF-IDF (Term Frequency - Inverse Document Frequency)

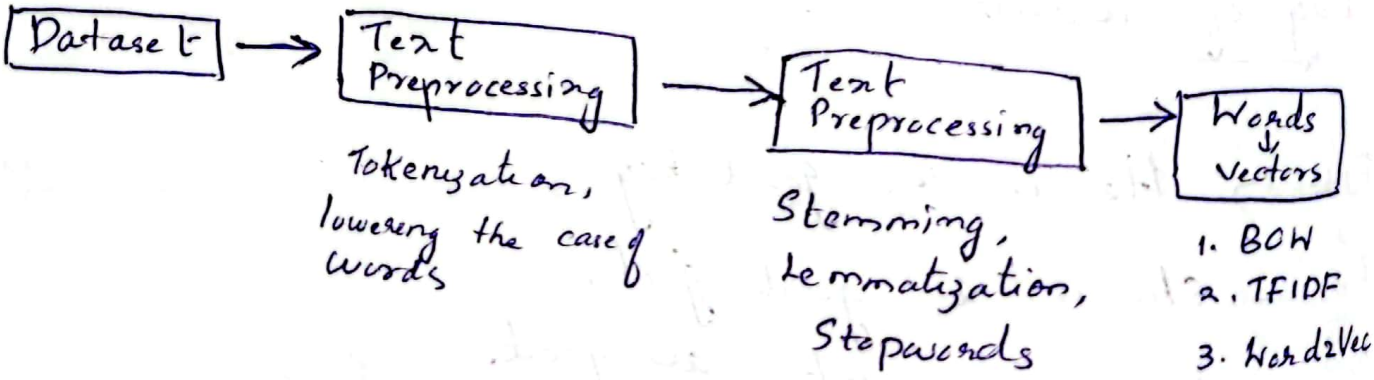
d) Word2Vec.

Basic Terminologies in NLP

1. CORPUS → Paragraph → [D1, D2, D3, D4]
2. Documents → Sentence → D1, D2, D3, D4
3. Vocabulary → Total number of unique words
4. Words → word

Sentiment Analysis

	Text	o/p
D1	The food is good	1
D2	The food is bad	0
D3	Pizza is amazing	1
D4	Burger is bad.	0



One Hot Encoding

[A man eat food
Cat eat food
People watch Krish Youtube] CORPUS

vocabulary = 9

D_1 [[1 0 0 0 0 0 0 0 0]
[0 1 0 0 0 0 0 0 0]
[0 0 1 0 0 0 0 0 0]
[0 0 0 1 0 0 0 0 0]]

→ OHE ←

D_2 = [[1 0 0]
[0 1 0]
[0 0 1]]

Advantages

1. Simple to implement
2. Intuitive

Disadvantages

1. It creates sparse matrix.
2. Out of Vocabulary (OOV)

Extra words that may coming in the test data cannot be handled.

3. Not fixed size.
4. Semantic meaning b/w words not captured.

Bag of Words

D_1 → He is a good boy
D_2 → she is a good girl.
D_3 → Boys and girls are good.

After applying Stop words

D_1 → Good boy
D_2 → Good Girl
D_3 → Boys Girls Good.

Vocabulary (3)	Frequency
Good	3
Boy	2
girl	2

f_1 f_2 f_3 (Order it based on the frequency in descending order)

Good boy girl

→ Whenever we use stop words make sure that lower all the words case (lower case)

	f_1	f_2	f_3	~~(Assumption)~~
	Good	boy	girl	o/p
Doc1	1	1	0	—
Doc 2	1	0	1	—
Doc 3	1	1	1	—

In BOW, we have an option to make it binary BOW.

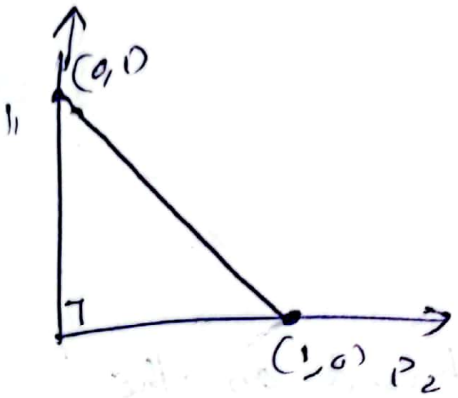
Advantages

1. Simple and Intuitive

Disadvantages

1. Sparsity
2. OOV
3. Ordering of the words has completely changed Imp
4. Semantic meaning is lost

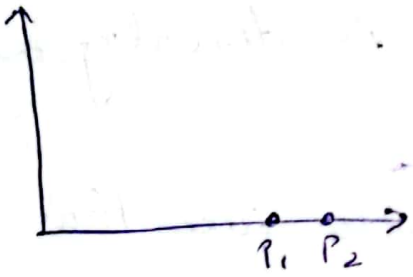
Cosine Similarity



$\cos 90 = 0$

$1 - 0 = 1$

Points are not Similar



$\cos 0 = 1$

$1 - 1 = 0$

Points are almost Similar.

To Capture the Semantic Information

We use Ngrams : Bigrams, Trigrams, Ngrams

	f_1 Good	f_2 boy	f_3 Girl	f_4 Good boy	f_5 Good Girl
Sent 1	1	1	0	1	0
Sent 2	0	0	1	0	1
Sent 3	1	1	1	0	0

Anna Eats Food

No. of Bigrams ? 2.

- Anna Eats
- Eats Food

I am not feeling well.

Trigrams ? 3.

- I am not
- am not feeling
- not feeling well.

- Bag of Words in text Processing is a Feature Extraction technique.

Day 3

Bag of Words is a technique to convert text into vectors.

(1) Sent 1 → He is a good boy
Sent 2 → She is a good girl
Sent 3 → Boy and girl are good.

Applying Stopwords

Sent 1 → good boy
Sent 2 → good girl
Sent 3 → boy girl good

③ Frequency (Vocabulary)

	frequency
good	3
boy	2
girl	2

④

	f_1 good	f_2 boy	f_3 girl
Sent 1	1	1	0
Sent 2	0	1	1
Sent 3	1	1	1

The food is good
The food is not good

food good is not the
1 1 1 0 1
1 1 1 1 1

Even though the sentences are opposite → Here it shows they are always similar

⇒ So this issue can be solved by TF-IDF

TF-IDF : Term Frequency - Inverse Document Frequency

After cleaning
Sent 1 : good boy
Sent 2 : good girl
Sent 3 : boy girl good

⇒ Whichever words are present rarely in the sentence we should give more weightage.

⇒ The rare words are captured by Term frequency and
Common words are captured by Inverse Document frequency.

⇒ How to calculate Term frequency?

$$\text{Term Frequency} = \frac{\text{No. of repetition of words in Sentence}}{\text{No. of words in Sentence}}$$

⇒ How to calcute IDF?

$$\text{IDF} = \log_e \left(\frac{\text{No of Sentences}}{\text{No. of Sentences Containing the word}} \right)$$

$$\text{TF-IDF} = \text{TF} \times \text{IDF}$$

Term Frequency

	Sent 1	Sent 2	Sent 3
god	1/2	1/2	1/3
boy	1/2	0	1/3
girl	0	1/2	1/3

Inverse Document Frequency

Words	IDF
good	$\log_e(3/3) = 0$
boy	$\log_e(3/2) = 0.65$
girl	$\log_e(3/2) = 0.65$

TF-IDF

	f_1	f_1	f_3
	good	boy	girl
Sent1			
Sent2			
Sent3			

	Sent 1	Sent 2	Sent 3
good	0	0	0
boy	0.325	0	0.217
girl	0	0.325	0.217

Advantages

1) Inductive
2) Word Imp are getting Captured.

Disadvantages

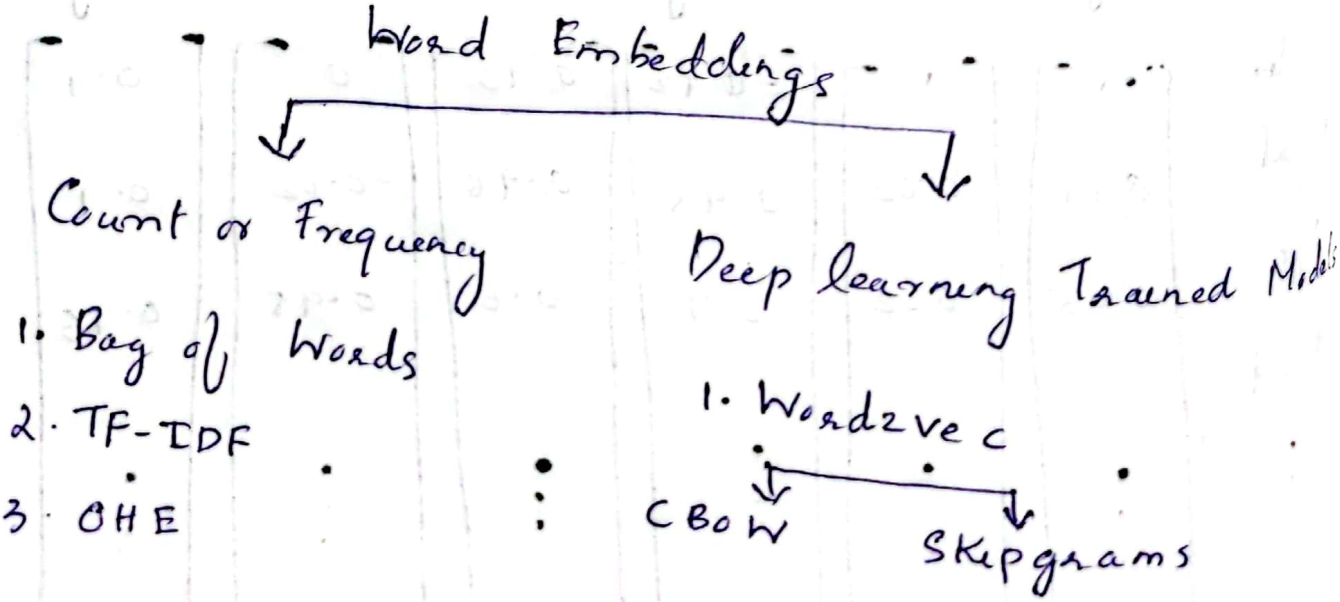
1. Sparsity (< bag of Words)
2. Out of Vocabulary.

Day 11

Imp 1. Word Embeddings
2. Word2vec → CBOW (Continuous Bag of Words)
→ Skipgram
3. Practical implementation using Python.

Word Embeddings

Technique to converts words into vectors



Word2vec : Feature Representation

f_1	f_2	f_3	f_4	f_5	f_6
Boy	Girl	King	Queen	Apple	Mango

→ For every word it is going to create vectors but within the limited dimension.

1. Limited Dimension
2. Sparsity is reduced.
3. Semantic meaning is maintained. (related info)

Wherever they are related the vectors will be almost same.

	f_1	f_2	f_3	f_4	f_5	f_6
	Boy	Girl	King	Queen	Apple	Mango
Gender	-1	1	-0.92	0.93	0	0.1
Royal	0.01	0.02	0.95	0.96	-0.02	0.1
Age	0.03	0.02	0.7	0.6	0.95	0.96
Food						
⋮	⋮	⋮	⋮	⋮	⋮	⋮

300 dimensions

This is created by the model.

King [0.96 0.95] Man [0.95 0.98]

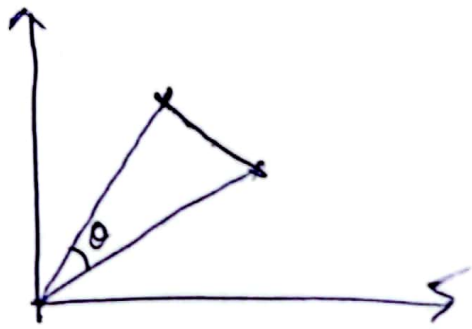
Queen [-0.96 0.95] Women [-0.94 -0.96]

King - Man + Women = Queen

We can find the distance using Euclidean distance, Manhattan Distance or Cosine Similarity

Distance = 1 - Cosine Similarity.

Cosine Similarity



Cosine Similarity = Cos θ.

∴ distance = 1 - Cos θ

if Cosine-Similarity = 1 ⟹ More Similar.

" " = 0 ⟹ No Similarity.

Word2vec

1. CBOW

CORPUS : TRAINING DATASET

[It is a Software that tries to think and learn]

a → o/p/target

It is ← Context

that tries to think and learn ← Context

Window Size = 5

Training Data.

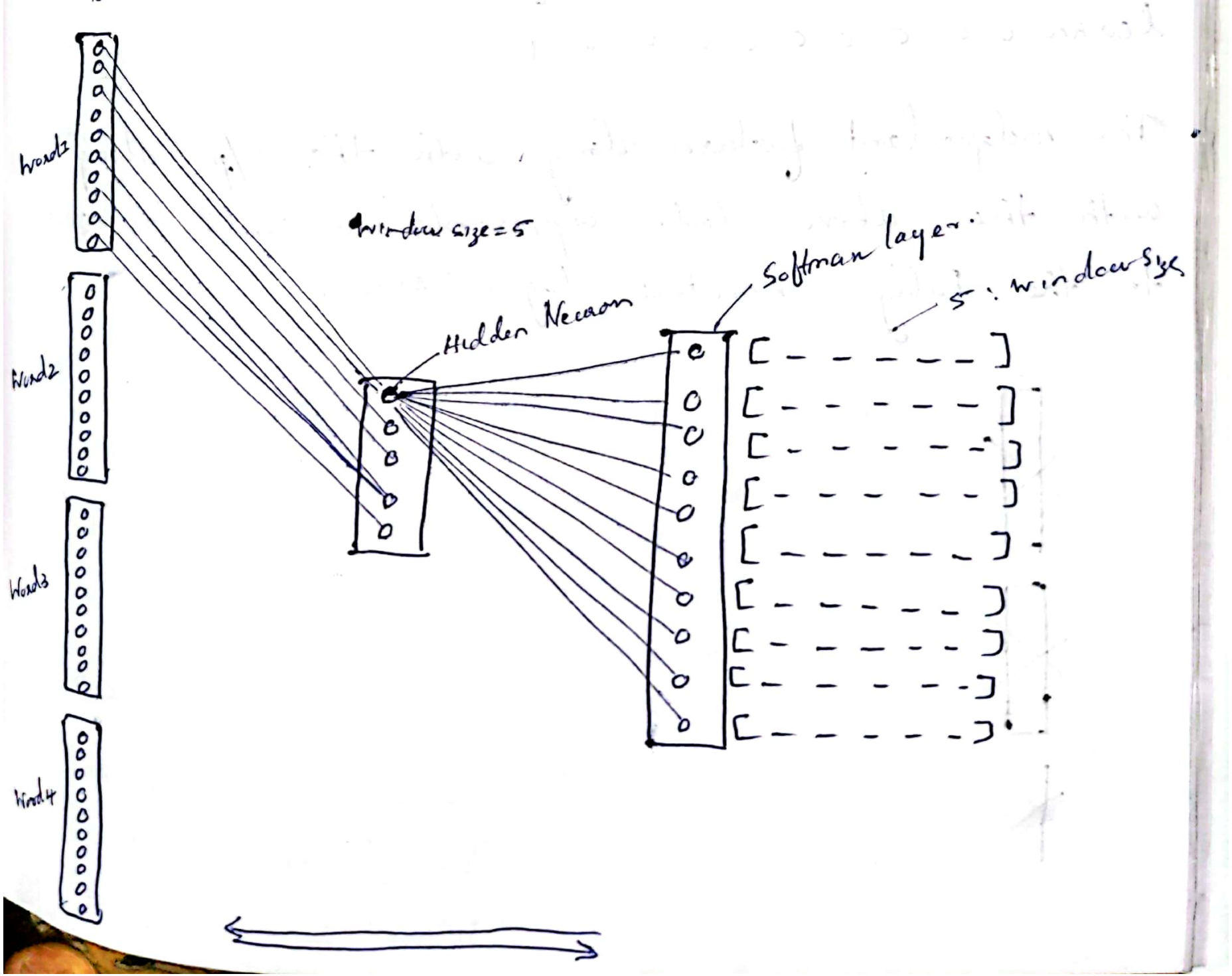
Independant Feature	o/p.
It, is, Software, that	a

[It is a. Sctware that tries to think and learn]

i/p	o/p
is, a, that, true	Software.
a, software, tries, to	that
Software, that, to, think.	true
that, true, think, and	to
tries, to, and, learn	think.

It	1	0	0	0	0	0	0	0	0	0
Is	0	1	0	0	0	0	0	0	0	0
a	0	0	1	0	0	0	0	0	0	0
Software	0	0	0	1	0	0	0	0	0	0
that	0	0	0	0	1	0	0	0	0	0
tries	0	0	0	0	0	1	0	0	0	0
to	0	0	0	0	0	0	1	0	0	0
think	0	0	0	0	0	0	0	1	0	0
and	0	0	0	0	0	0	0	0	1	0
learn	0	0	0	0	0	0	0	0	0	1

- The endependant features along with the o/p along with the above data representation will be given to the Fully Connected layer (ANN).


Word1
Word2
Word3
Word4
window size = 5
Hidden Neuron
Softmax layer
5 : window size

2. Skip Gram

i/p and o/p are interchanged

i/p	o/p
Software	is, a, that, true
that	a, software, true, to
true	Software, that, to, think
to	that, true, think, and
think	true, to, and, learn

Day 5

1. Practical Implementation

Spam Classification

① Text Preprocessing → Tokenization, Stopwords, Stemming, Lemmatization, NLTK

② Text → Vectors : BOW, TF-IDF, Word2Vec, Avg word2vec.

Word2vec

Skip grams CBOW

Word2vec

Pretrained model — Train the model from scratch.

Q) Should we always train the model or use the pretrained model?

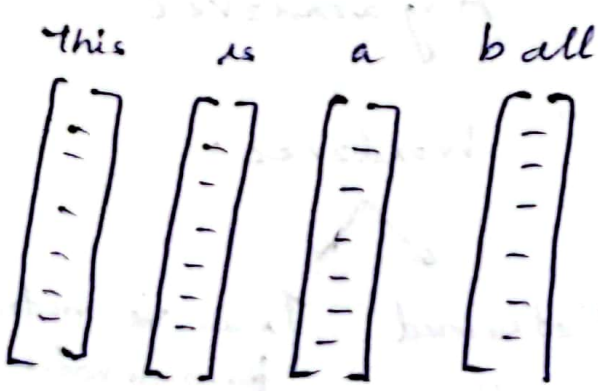
- If in the new dataset if already captured the 75% of the words in pretrained model then opt the pretrained Model
- Otherways, train your own model.

Avgword2vec

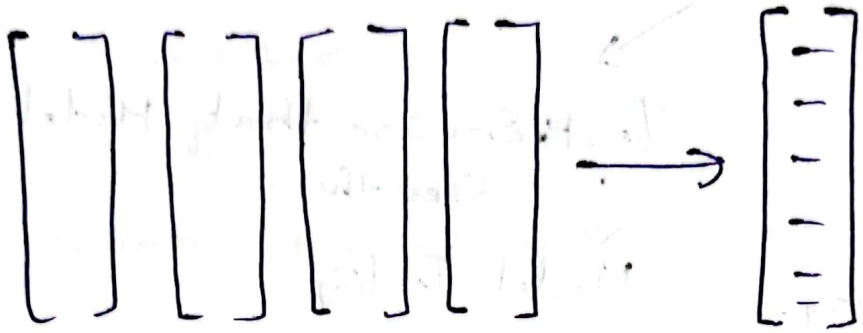
It is an extension of the Word2vec model that generates vector representations for sentences or documents instead of individual words.

Sent: This is a ball

Word2vec converts each word into specific dimension say 300



But Avg Word2vec find out the average.



Day 6

NLP - (Recurrent Neural N/w)

Agenda

1) RNN 2) LSTM RNN 3) GRU RNN

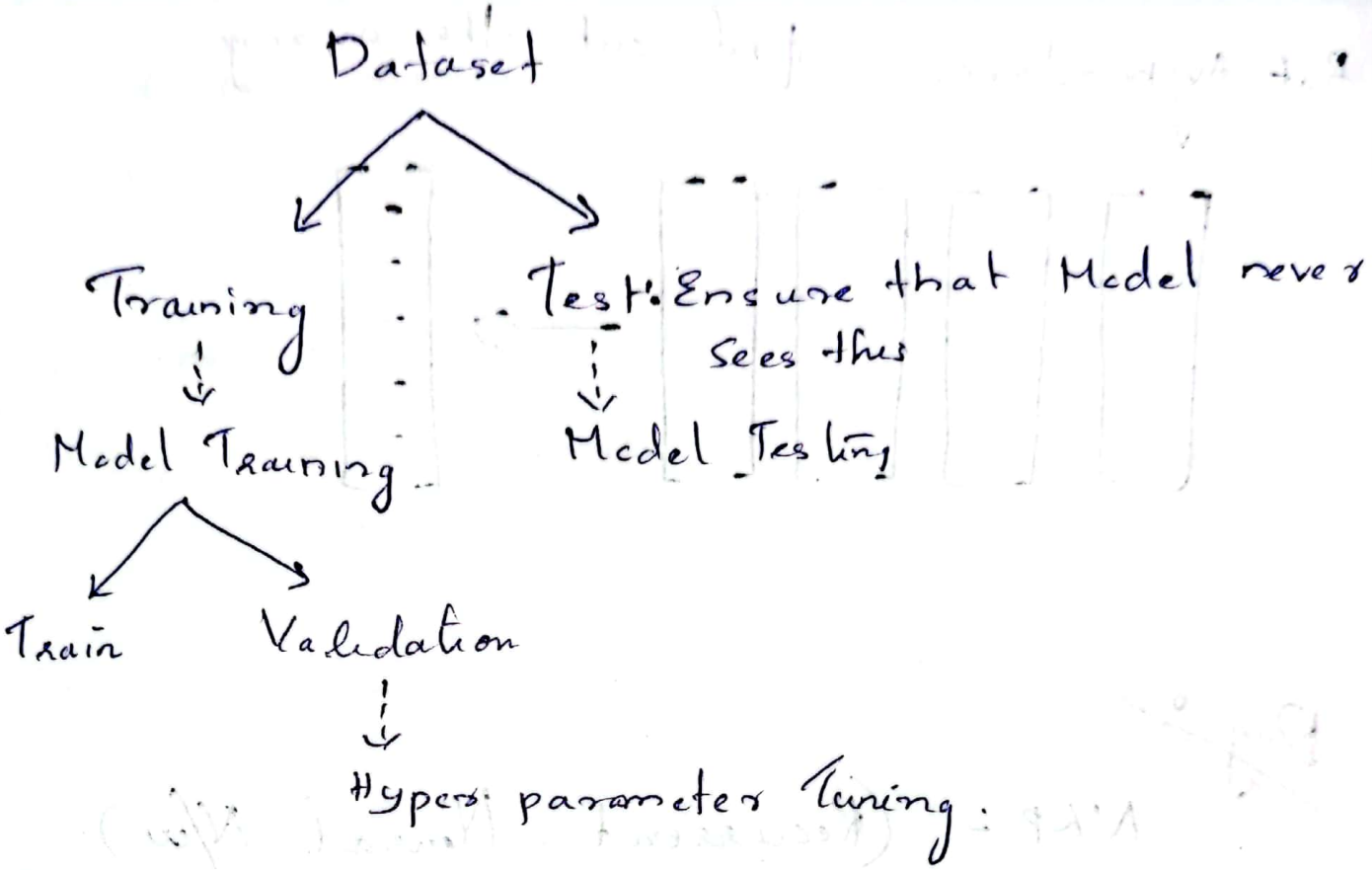
4) Bidirectional LSTM RNN

5) Encoders - Decoders

6) TRANSFORMERS 7) BERT.

Interview Questions

1. Train V/s Test V/s Validation ?



2. Why people use RandomForest instead of Decision Tree.

DT ⟶ Low Bias and high Variance.

Inorder to reduce the Variance keeping the bias low we use Random Forest instead of Decision Tree.

1. Recurrent Neural N/w.

Text → Vectors.

⇓

Word2vec, Avgword2vec } Word Embeddings.

Application :

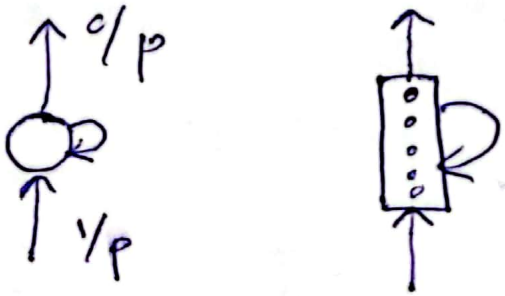
1) Chatbot.
2) Language Translation
3) Text Generation

ML techniques will not be able to give us good accuracy.

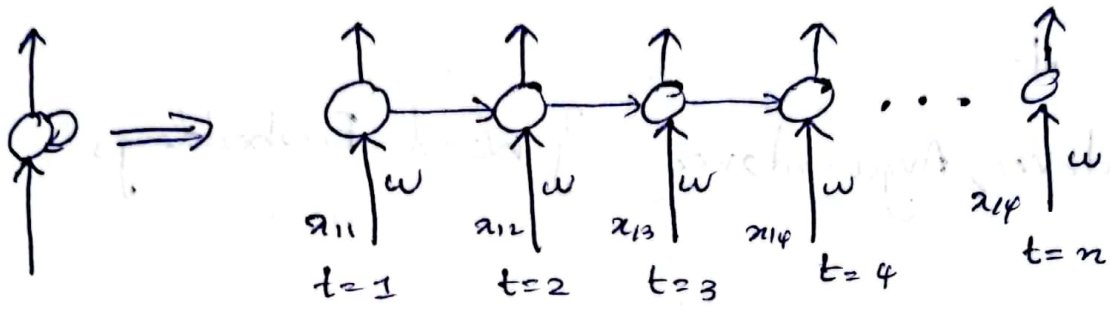
Deep Learning Techniques.

1) RNN 2) LSTM RNN 3) Transformer 4) BERT

Recurrent Neural N/w



~~Sentiment~~



Every o/p is given to the same neural n/w.

~~each~~

Sentiment Analysis → The food is so good.

o/p

Positive

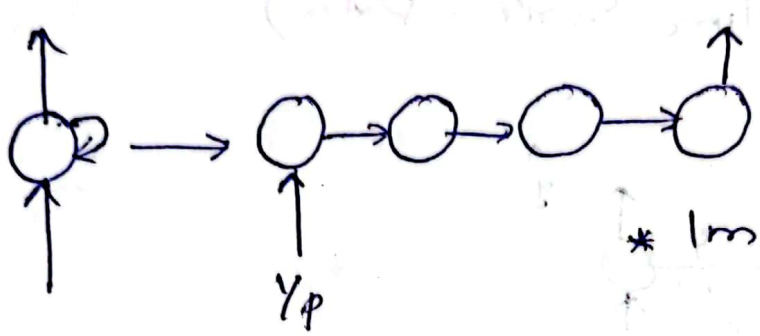
$< x_{11}\ x_{12}\ x_{13}\ x_{14}\ x_{1f} >$

x_{11} → word2vec → vectors → d = 300 (no. of features)

Types of RNN

1. One to One RNN
2. One to Many RNN
3. Many to one RNN
4. Many to Many RNN

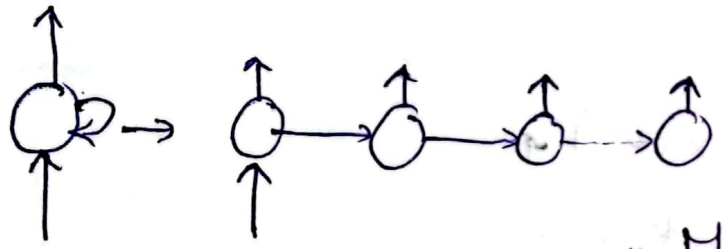
1. One to One RNN



1 i/p and 1 o/p.

* Image classification.

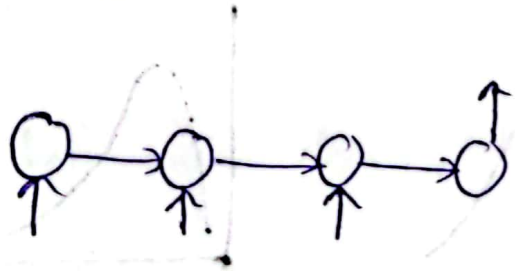
2. One to Many RNN



1 i/p multiple o/p

* Music generation
* Text generation
* Google search suggestion
* Movie Recommendation

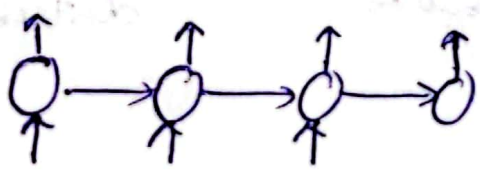
3. Many to One RNN



multiple i/p 1 o/p

* Sentiment Analysis
* Predict Next Day Sales

4. Many to Many RNN.

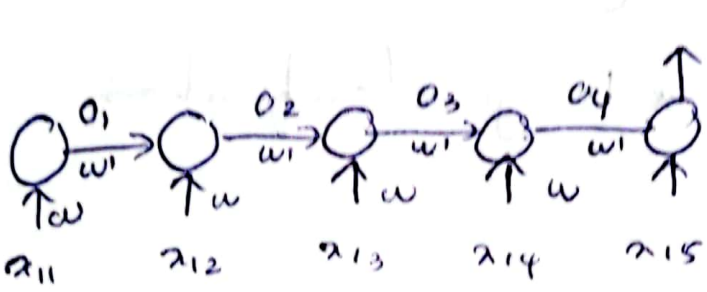


multiple i/p multipe o/p.

* Language Translation
* Question Answers
* Chatbots.

1. Forward Propagation in RNN:

(RNN is used for Time Series Data)



The food is very good. o/p Positive

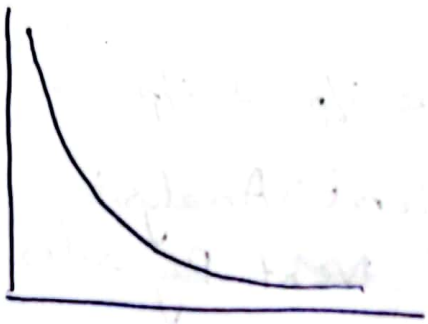
$$o_1 = f(x_{11} * w)$$

$$o_2 = f(x_{12} \times w + o_1 \times w')$$

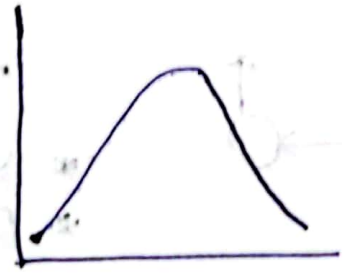
$$o_3 = f(x_{13} * w + o_2 * w')$$

Day 7

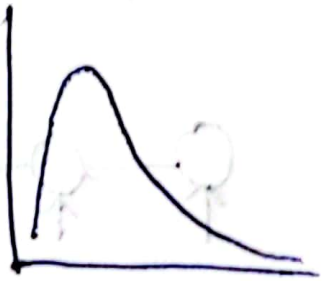
Q)



Power law distribution or Pareto distribution

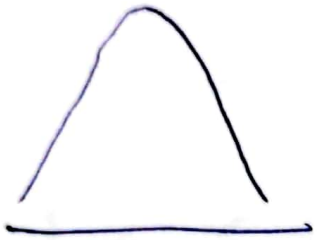


Normal distribution



log normal distribution

- Boxcox transformation is used to transform Pareto distribution → Normal.

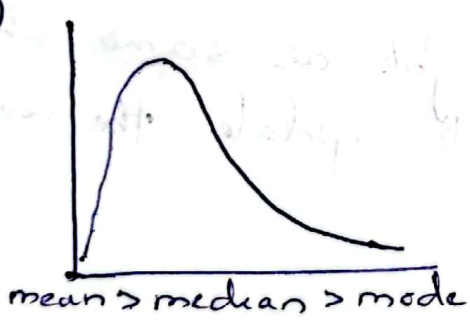


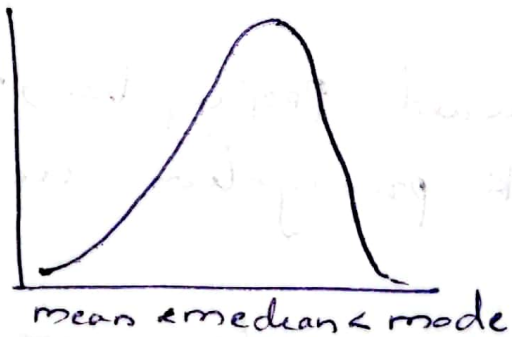
Q) What techniques should we apply to check whether this distribution follows normal

ans: Q-Q plots

Q) What is SND?

Q)



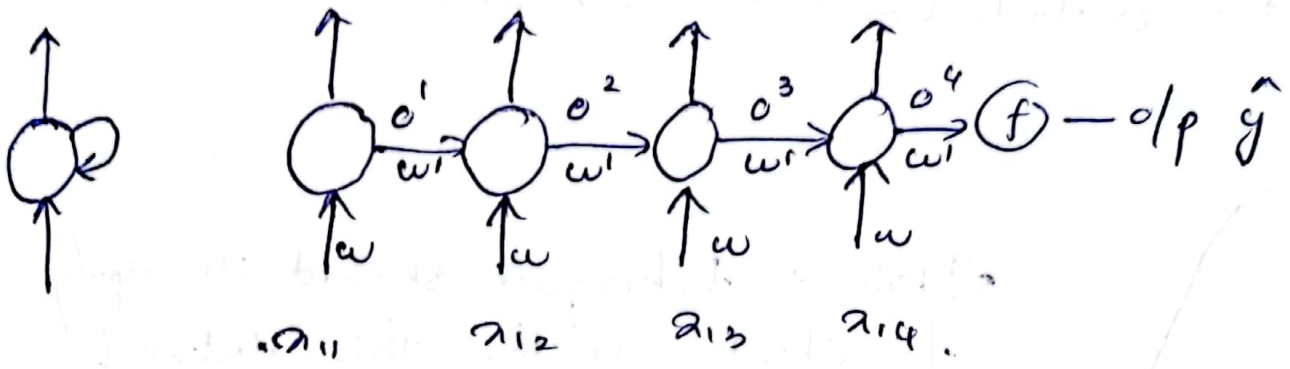


What is the relation b/w mean, median, mode in the above diagram?

Q) Difference b/w fit-transforms and transforms?

Q) Difference b/w Standardization and Normalization

RNN



$o_1 = f(x_{11} \times w)$

$o_2 = f(x_{12} \times w + o^1 \times w^1)$

$o_3 = f(x_{13} \times w + o^2 \times w^1)$

In Forward propagation, the weights are same while in back propagation we need to update the weights

Weight Updation formula

$$w'_{new} = w'_{old} - \eta \frac{\partial L}{\partial w'}$$

$$\rightarrow \frac{\partial L}{\partial w'} = \frac{\partial L}{\partial \hat{y}} * \frac{\partial \hat{y}}{\partial w'}$$

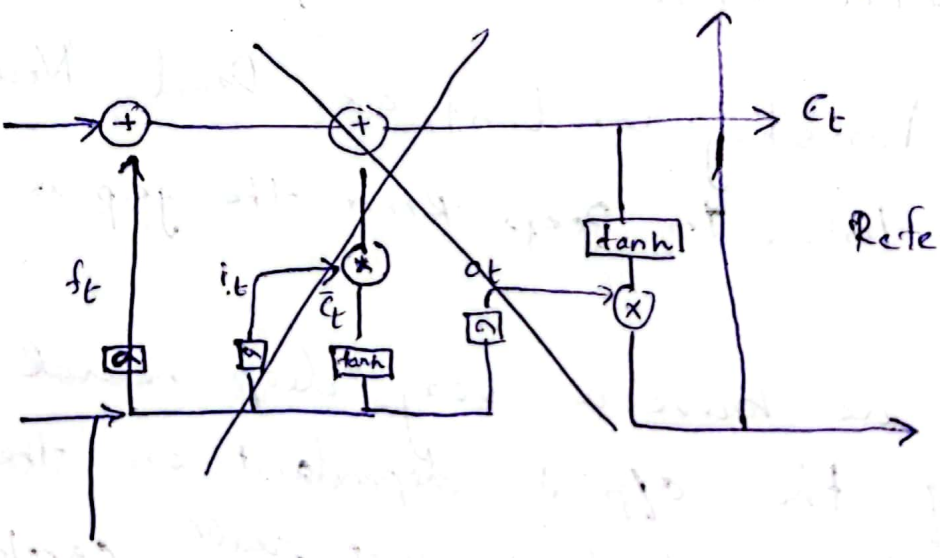
$$w_{new} = w_{old} - \eta \frac{\partial L}{\partial w_{old}}.$$

$$\rightarrow \frac{\partial L}{\partial w} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial o_{y4}} \cdot \frac{\partial o_4}{\partial w'}$$

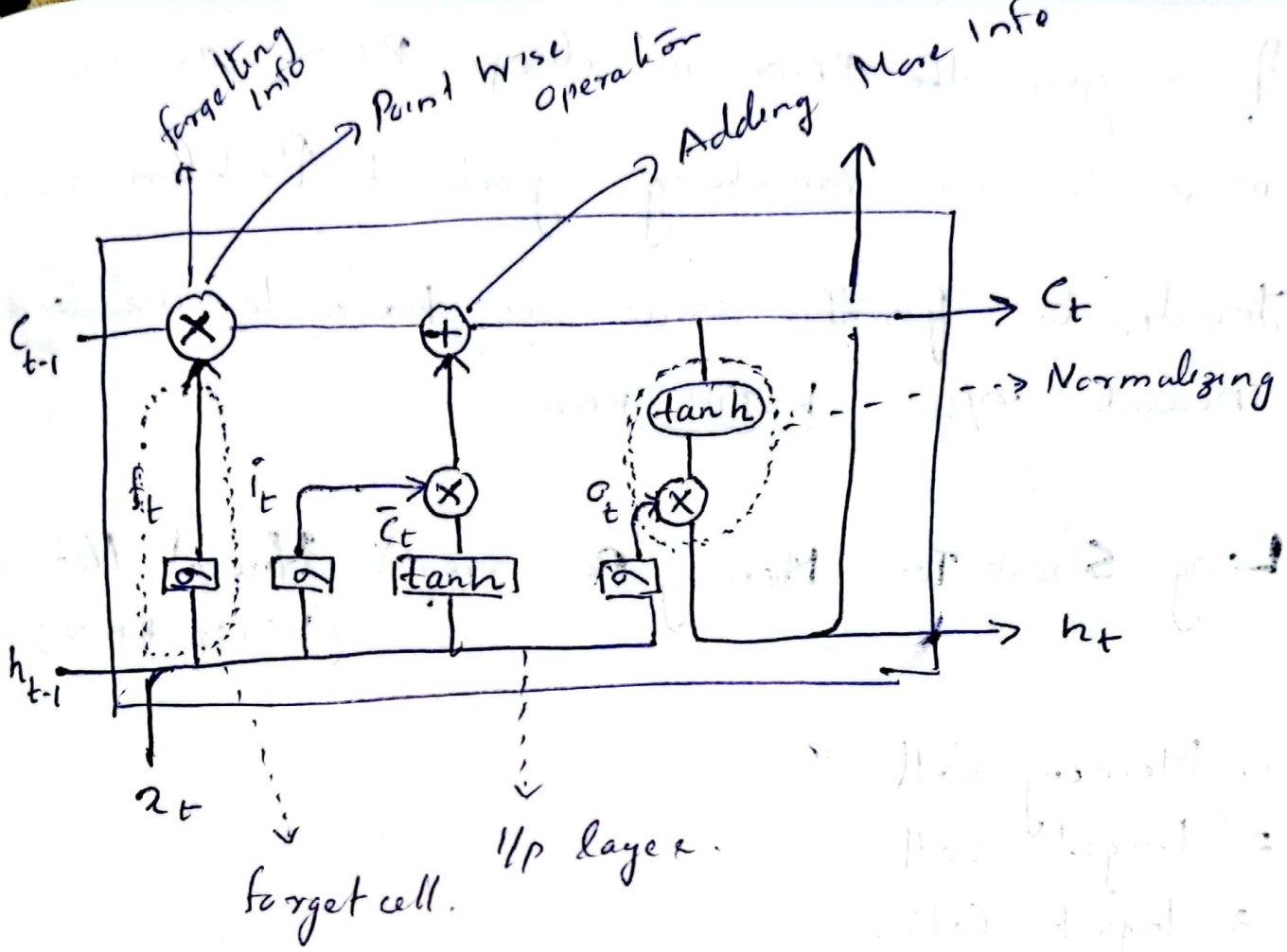
If suppose the RNN is deep RNN as we move forward, vanishing gradient Problem occurs. Inorder to fix this issue we have to use another neural n/w LSTM RNN

Long Short Term Memory Recurrent Neural Network (LSTM RNN)

1. Memory Cell
2. Forget Cell
3. Input Cell.



Refer Google.



Imp Q) Why LSTM RNN ~~not~~ not RNN?

1. Solves Vanishing Gradient or Dead Neuron.
2. Context Info. In Deep RNN the gap is huge.

Let's say we have a very deep neural n/w and some of the o/p is dependant on the first i/p, but this context is not really captured with the help of RNN.

Day 8

Agenda

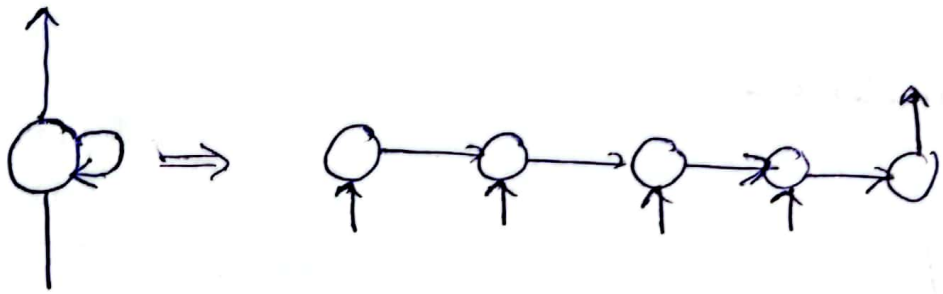
1. LSTM RNN (indepth Architecture).

Problem with RNN

Suppose I want to predict the next Word,

On Sunday I want to eat Pizza,

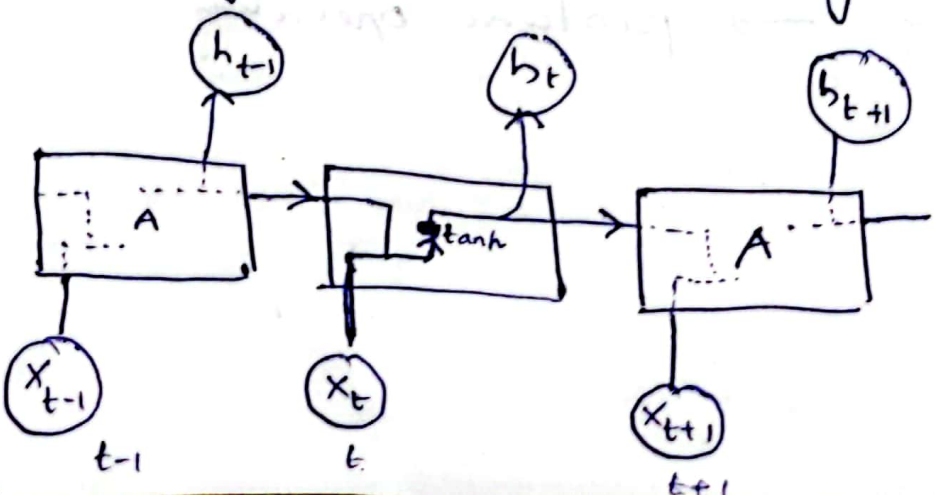
On Monday I want to eat ____



RNN can have only short term memory.

Represent RNN

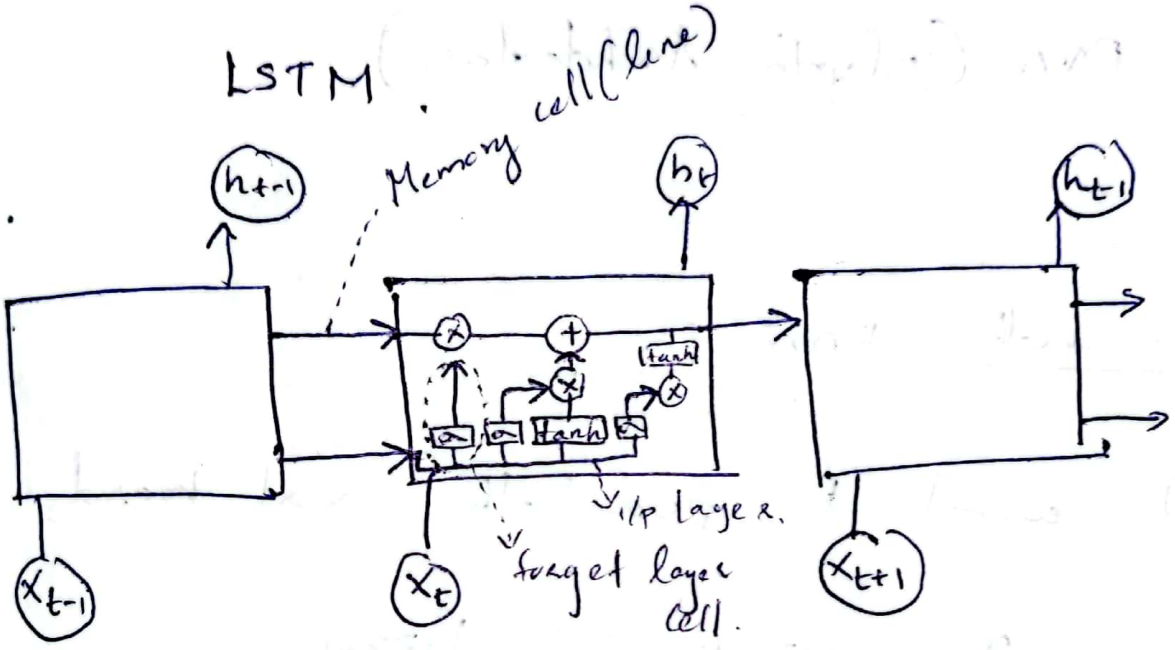
Traditional RNN



Why we Specifically use LSTM RNN?

i) Long Sentences → Context will be switching.

LSTM.



Refer google.

σ → Neural N/w with Sigmoid Activation fun.

tanh → Neural N/w with tanh Activation fun

→ Concatenate

→ Copy

→ → Vector transform

→ pointwise operation

Memomy cell

It acts like a conveyer belt in airport
- Similarly, you can add and remove information.

Forget layer cell

- It is specifically used for one critical information
- The Sigmoid layer will be trained in such a way that, for most of the information we get zeros. ie, we are making the memory cell forgetting the information because, forward we have a point wise operation of cross! ⊗

Say 2 Sentences

① Anna like pizza but She doesn't like Burger
↑
Context Switch is not happening

②. Anna like pizza but her friend like Burger.
Context Switch is happening.

In ①, the information is not lost.
In ②, the ↑previous information is lost.

I/p Gate layer

Consider, "Anna like pizza but her friend like Burger".

This info should be forget

This information should be stored, or add it in the memory cell.

And it will be passed through o activation funct and tanh activation funct;

And when you combine both the new context will be added to the memory cell.

Output Gated layer

From the memory cell we get the recent information on top of that tanh is applied. And it is combined with ~~activation~~ the previous forgotten information. & Then it is passed to the o/p to the next cell.

Similar things will go on wit different time stamps.

& all useless layer is removed, only imp data is passed to the o/p.

Day 9

Agenda

1. Word Embeddings Layer: Tensorflow ~~and Python~~.
2. LSTM Sentiment Analysis.

Word Embedding Layer.

1. Sentence.
2. OHE : One Hot Encoding (Vocabulary Size)
3. Padding : Post Padding and Pre Padding
4. OHE → vectors.

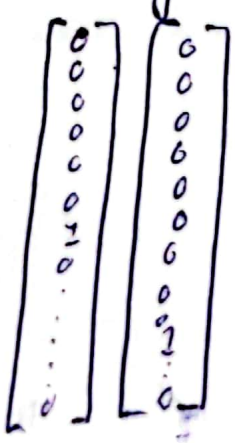
Sent

[Anna Likes Pizza]

[Ram Likes Burgers]

Say Vocabulary size = 500

OHE



By considering this OHE we have take that entire words and probably create an embedding layer and convert this into vectors → which is done by Embedding Layer in DL.

Practical: Word Embeddings.

Day 10

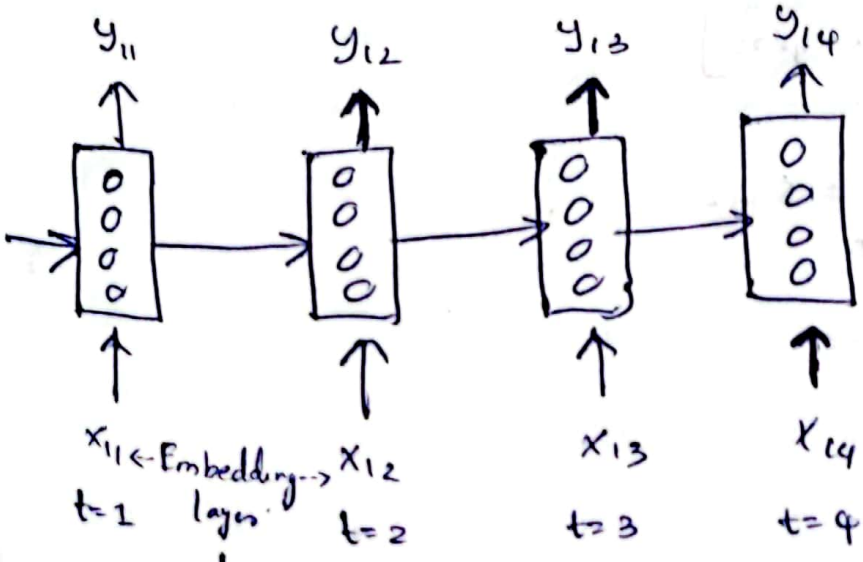
Fake-news-classification-using:- LSTM.

Day 11

Bidirectional LSTM RNN.

Anna likes to eat → o/p. in Kochi

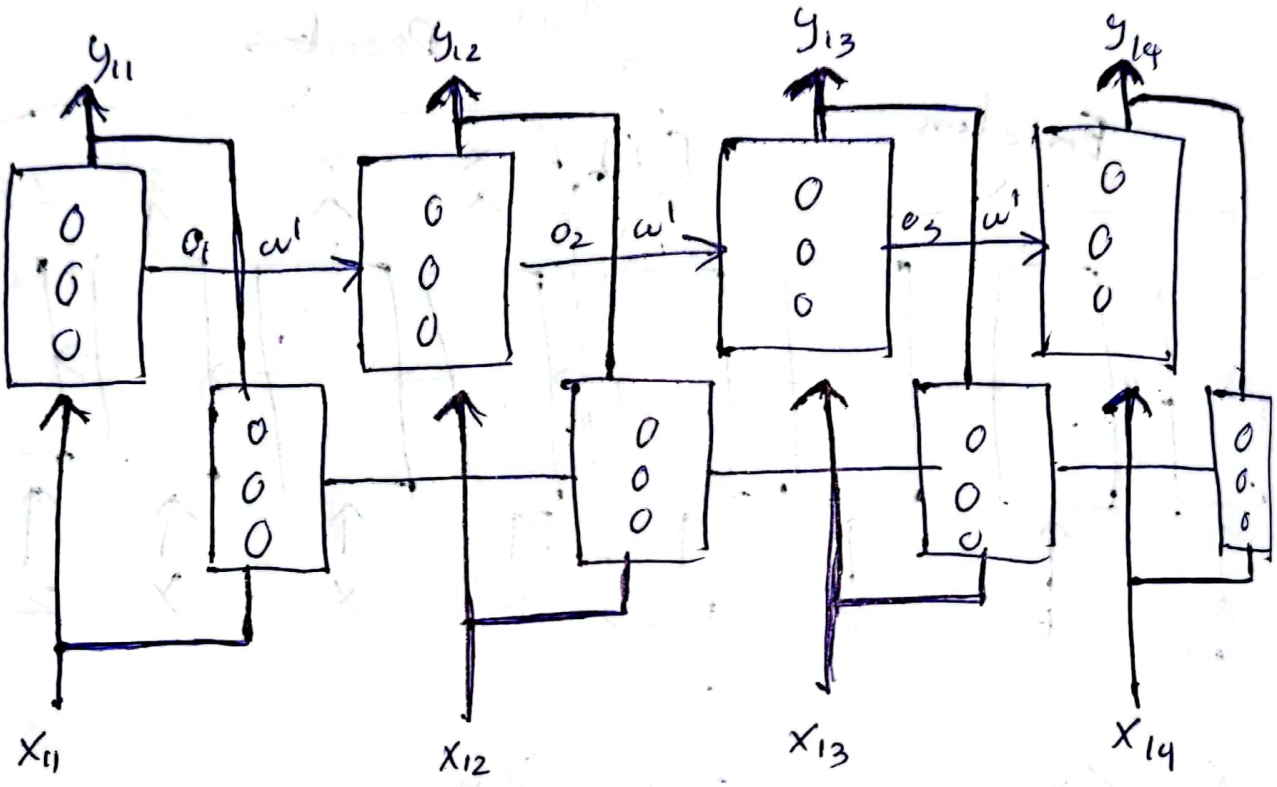
Many-Many RNN



which makes sure that it converts the text into

vectors

y_{13} have the information of x_{13}, x_{12}, x_{11}

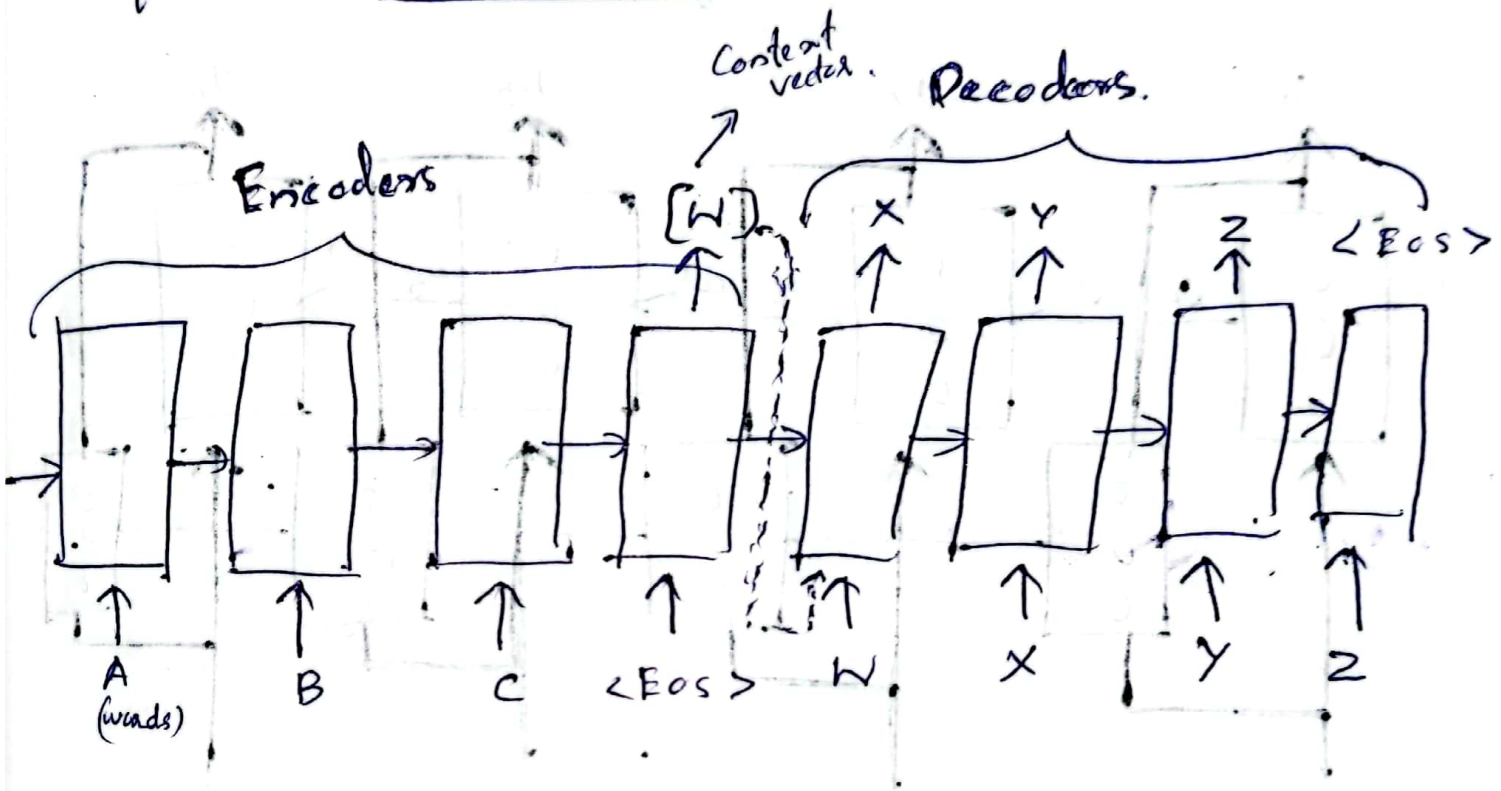


In the forward direction you are passing all the weights and in the backward direction you are passing the words.

Practical

Fake-news-classification using Bidirectional LSTM.

Sequence to Sequence with Neural N/w.



$X = \langle x_1, x_2, x_3 \rangle$ $\quad Y = \langle y_1, y_2, y_3 \rangle$

Vector conversion ⟶ Embedding layer.
One Hot Representation
Word2vec

- Once we are able to get the context vector, then passed to the Decoder.
- The i/p and o/p Sequence may be completly different.
- When we are doing the prediction for new test data, we will be creating $\hat{y_1}$, $\hat{y_2}$, $\hat{y_3}$ and so on.

- $P(\hat{y} / \langle x_1, x_2, x_3 \ldots x_n \rangle)$
- $\hat{y}$ is a Combination of $\langle \hat{y_1}, \hat{y_2}, \hat{y_3}, \ldots \hat{y_t} \rangle$.
- Once we find out $\hat{y}$, we try to find out the loss. $L(y, \hat{y})$. And we can use an optimizer to reduce the loss through back Propagation.
- In case of Image captioning, we use advanced CNN like VGG16, RESNET instead of LSTM as an Encoder.

Disadvantages

When you have long sentence the accuracy is not good.

So we are going to use another model called Attention Model (Bidirection LSTM as Encoders)

Problems with Encoders and Decoders

BLEU (Bilingual Evaluation Understudy is a metric used to measure the quality of Predicted text in NLP models by comparing

it do a set of references.
Ranges from 0-1, higher values indicating better predictions.

0 → Machine translated o/p has no overlap with the reference translation.

1 → perfect overlap.

Problems with Encoders and Decoders

1. It usually do not perform with longer sentence
2. w, lower BLEU score.

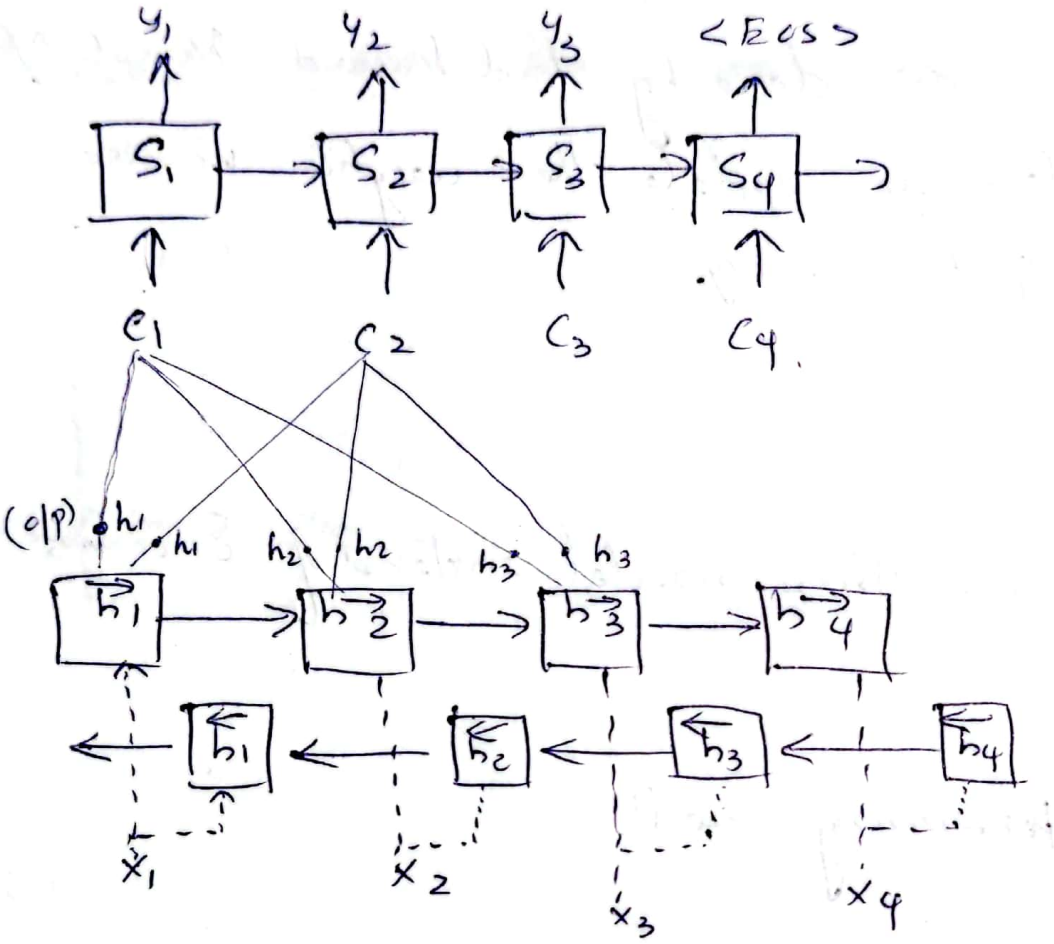
* Window-Size

Lets take an example of language translator.

FRENCH → ENGLISH.

Translator will hearing some sentence then he/she will be translating.
Similarly we choose 'Window size': hyperparameter

In Attention Model : Encoder → Bidirectional LSTM



- If window size = 3.
- ie, you are giving attention to some words (window size)
- LSTM RNN works on probability.
- h based on Tx will generate some values like $\alpha_1, \alpha_2, \alpha_3$ (weights)

These are some condition,

For eg: $T = 3$

$\alpha_1, \alpha_2, \alpha_3$

(1) $\sum_{i=1}^{T_x} \alpha_i = 1$

② Context vector $c_i^o = \sum_{i=1}^{T_x} \alpha_{ij} h_i^o$

these operations are done by feed forward Neural N/w. Until and unless we update the weights, we are not gonna get good accuracy.

Interview

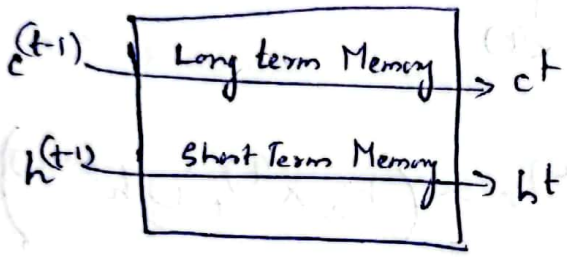
Why do you use these model instead of Encoders and decoders?

- It was not performing well.
- (Problems)

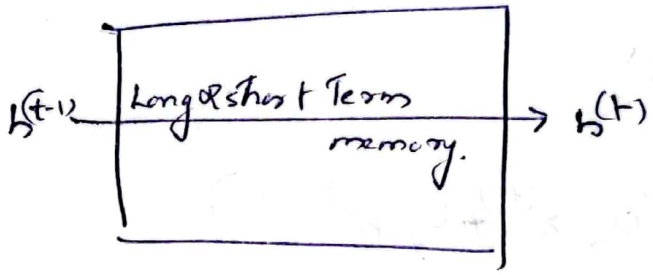
GRU (Gated Recurrent Units)

- Traditional RNN have short term Memory.

LSTM

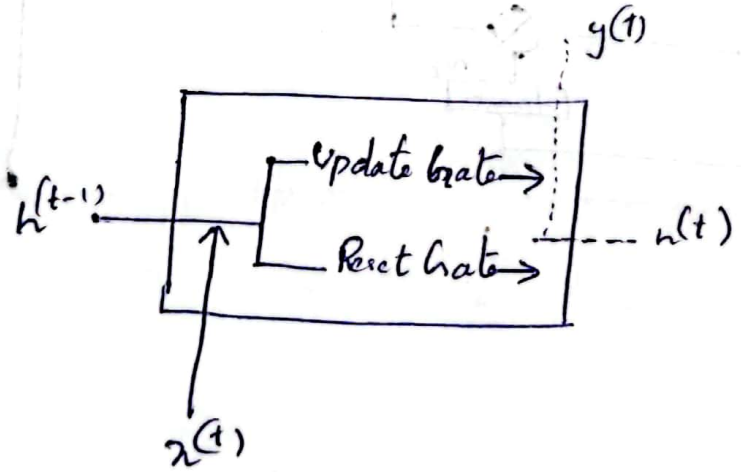


GRU



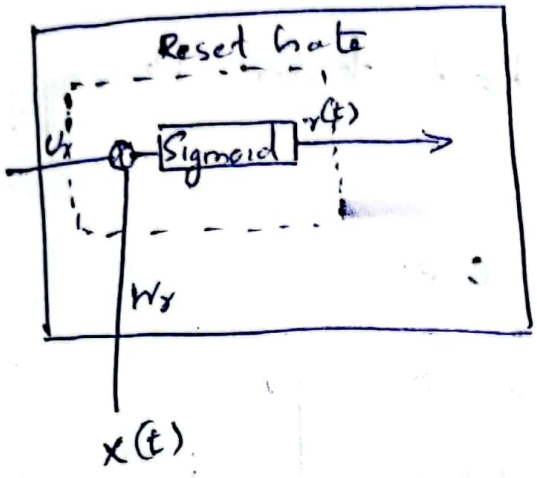
LSTM has 3 gates → i/p, o/p, forget

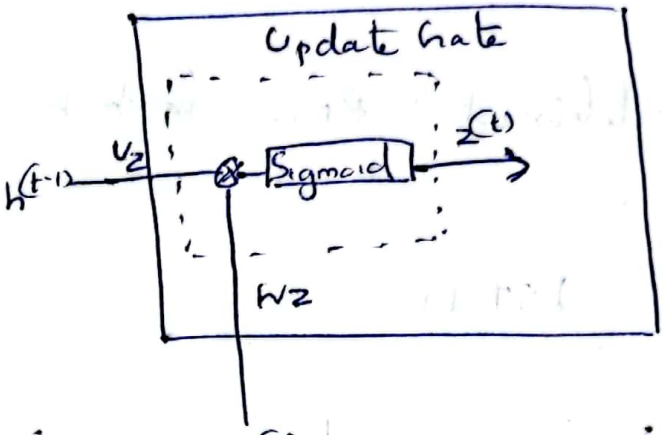
GRU has 2 gates → update and Reset.



Update Gate helps : How much of Past memory to retain

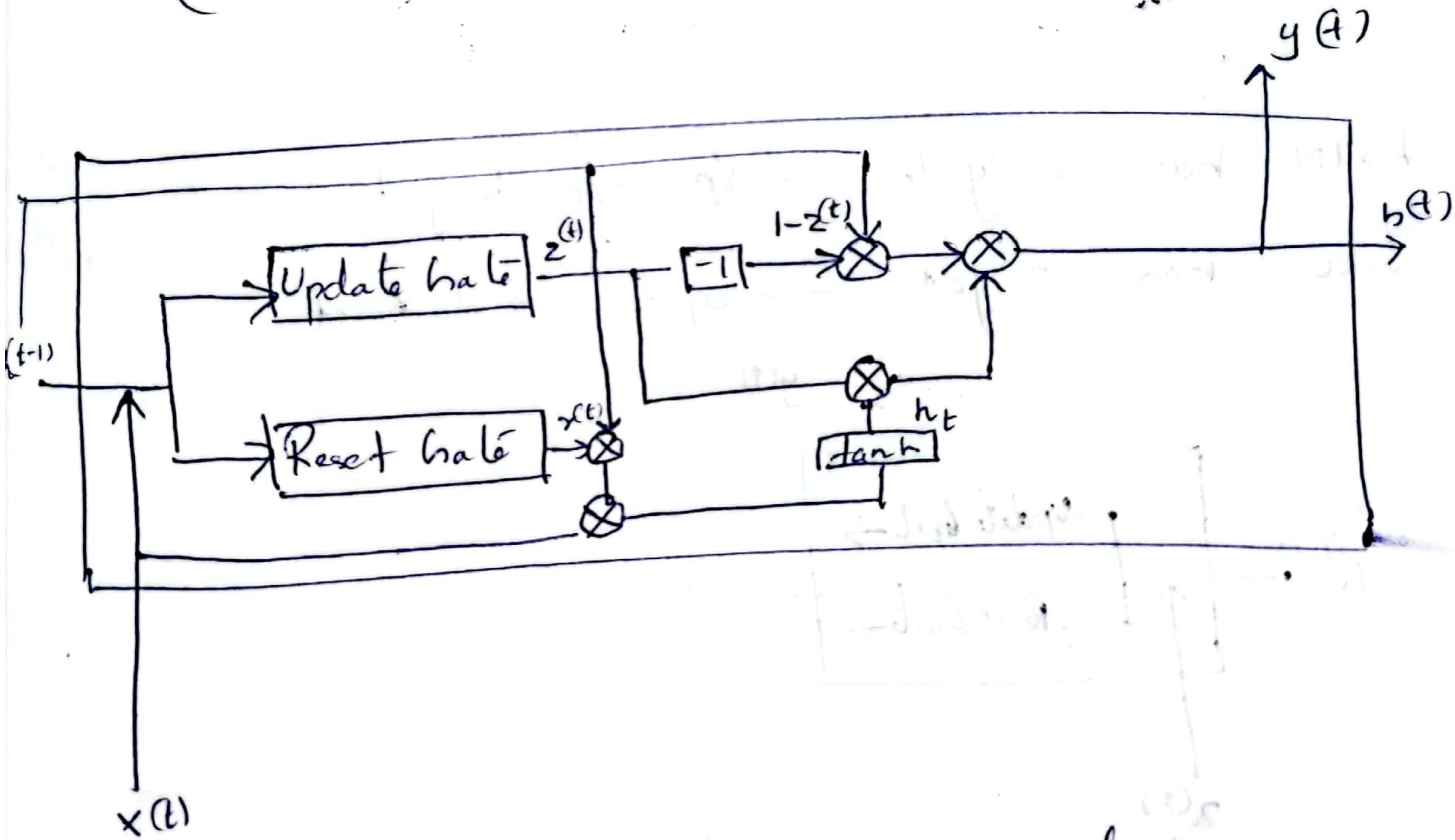
Reset Gate : How much of Past memory to forget.





$$r^{(t)} = \sigma\left(U_r x^{(t)} + U_r h^{(t-1)}\right)$$

$$z^{(t)} = \sigma\left(W_z x^{(t)} + U_z h^{(t-1)}\right)$$



Difference b/w LSTM and GRU

LSTM

1. 3 gates: I/P, O/P, forget
2. More accurate on longer sequence, less efficient

GRU

1. 2 Gates: reset, update
2. More efficient computation wise, getting more popular.

Transformers

- Transformers are a type of neural n/w archte.
- Are that transforms or changes an i/p int an o/p sequence.
  - It enable large scale models
  - Enable faster customization
  - Facilitate multi-modal AI system
  - AI research and industry innovation.

Use Cases of Transformers.

1) NLP 2) Machine Translation
3) DNA Sequence Analysis
4) Protein Structure Analysis.

- Self attention Mechanism in Transformers.

Components of transformer architecture.

for image refer. AWS : transformers
and blog : the illustrated Transformer by Jay Alammar.

Components

- I/p embeddings
- Positional embeddings
- Transformer block
- Linear and Softmax blocks

Different Types of transformer models?

1) ~~Bidirec~~ Bidirectional Transformers - BERT.
2) Generative pretrained Transformers
3) Bidirectional and autoregressive transformers
4) Transformers for multimodal Tasks.
5) Vision Transformers

Difference b/w transformers and CNN ?
" " " RNN ?

BERT (Bidirectional Encoder Representation from Transformers)

- Refer Blog by Jay Alammar.

- BERT can generate contextualized embeddings
  - BERT is a model with absolute position embeddings so it's usually advised to pad the i/p on the right rather than the left.
  - BERT was trained with the masked language Modelling (MLM) and next Sequence prediction (NSP) objectives. It is efficient at predicting masked tokens and at NLU in general, but it is not optimal for text generation.