Day 2 - NLP For ML And DL

{NLTK}

Agenda

① Text preprocessing → Words → Vectors

a) OHE (One hot Encoding)
b) Bag of Words (BOW)
c) TF-IDF (Term Frequency - Inverse Document Frequency)
d) Word2Vec

⇒ practical Implementation
⇒ ngrams

② Quiz → Live → 5000Rs INR

1st Prize → 2000 Rs INR
2nd Prize → 1500 Rs INR
3rd Prize → 1500Rs INR

Basic Terminologies Used In NLP

① CORPUS ✓ → Paragraph → [D1, D2, D3, D4]
② Documents ✓ → Sentence
③ Vocabulary ✓ ⇒ 10K unique words
④ Word → 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

⇑
DATASET

→ Dictionary Book
10K unique words

DATASET → Text Preprocessing -1 → Text Preprocessing -2 → Words ↳ vectors

Tokenization,

① STEMMING

① BOW 2.

lowering the case of words ② Lemmatization ① TFIDF ⇒
③ STOPWORDS ③ Word2Vec

① One hot Encoding

Paragraph

Vocabulary

9 ⇓

→ A man eat food
→ Cat Eat food
→ People watch KRISH YT

⇒ CORPUS

→ Size

A man eat Cat People Watch KRISH YT

⇑ food

Out of vocabulary

⇒ CANNOT TRAIN THE MODEL

$$D1 \rightarrow \begin{bmatrix} [1 & 0 & 0 & 0], \\ [0 & 1 & 0 & 0], \\ [0 & 0 & 1 & 0], \\ [0 & 0 & 0 & 1] \end{bmatrix}$$

$$D2 - \begin{bmatrix} [1 & 0 & 0] \\ [0 & 1 & 0], \\ [0 & 0 & 1] \end{bmatrix}$$

Advantages

① Simple to Implement
② Intuitive

Disadvantage

① Sparse Matrix ✓ [Extra Test data]
② OOV {out of Vocabulary} ✓
③ Not fixed size ✓
④ Semantic meaning between word is not captured

② Bag of Words

Stop words
lower all the words
case

D1 → He is a good boy
D2 → She is a good girl ~~ST~~
D3 → Boy and girl are good

D1 → good boy good
D2 → good girl
D3 → Boy girl good

Vocabulary	Frequency
good	3
boy	2
girl	2

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

O/p ⇒ Assumptions

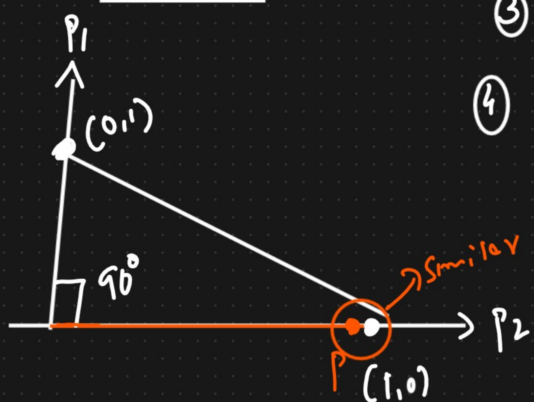
{ Euclidian Distance
Cosine Similarity.

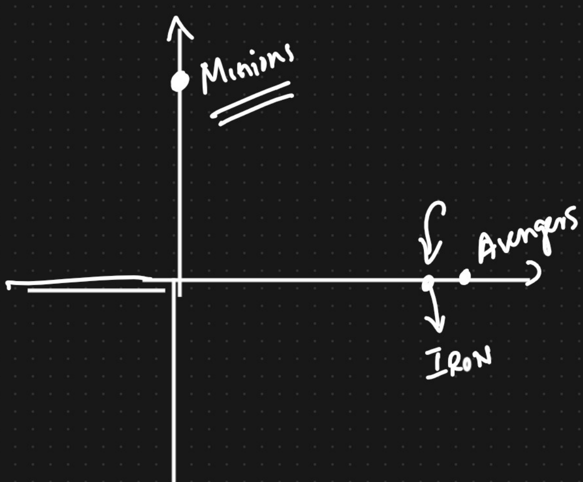
BOW ⇒ Binary BOW

Advantages

① Simple and Intuitive

Cosine Similarity





Capture the semantic Info

Disadvantages

① Sparsity
② OOV
③ Ordering of the words.
④ Semantic meaning Not able to capture }.

$\cos 45 = 0.53$

$1 - 0.53 = \text{Cos-Similarity}$

0.47

$\cos 90 = 0$

$1 - 0 = \boxed{1}$

$\cos 0 = 1$

$1 - 1 = 0$

Ngrams ⇒ Bigrams, Trigrams, — N grams

	f1 good	f2 boy	f3 girl	f4 good boy	f5 good girl
Sent 1	1	1	0	1	0
Sent 2	1	0	1	0	1
Sent 3	1	1	1	0	0

KRISH EATS FOOD

BI-GRAMS ? 2 Bigrams

⇓

practically

KRISH EATS   EATS FOOD

TRIGRAMS ⇒ 3 Trigrams

I am not feeling well

I am not   Am Not feeling   Not feeling well

ngrams

(1, 3)

KRISH IS NOT FEELING WELL

f1	f2	f3	f4	FS
KRISH	IS	NOT	FEELING	WELL

KRISH IS

Day 3 - NLP

Agenda

1. BOW {Bag of Words}
2. Tf Idf
3. Practical Implementation
4. Quiz

Spam Classifier

Word2vec

1. Embedding Layer
2. Word2vec → CBOW ✓ / Skip gram ✓
3. Architecture
4. Practical Problem
5. Glove

Bag of Words → Text → Vectors

① Good

Sent 1 → He is a good boy

Sent 2 → She is a good girl

Sent3 → Boy and girl are good

Stopwords ⇒ Lowering

②

Sent1	good	boy	
Sent 2	good	girl	~~C4~~
Sent 3	boy	girl	good

③ Frequency (Vocabulary)

	frequency
good	3
boy	2
girl	2

Not all Similar

④

	good f_1	boy f_2	girl f_3
Sent 1	1	1	0
Sent 2	1 ✓	0	1 ✓
Sent3	1	1	1

opposite

The food is good

The food is not good

food	good	~~is~~	not	~~The~~
1	1	1	0	1
1	1	1	1	1

⇓ { TF-IDF }

⇓ Similar

Term Frequency – Inverse Document Frequency   TF - IDF

Sent 1 : good boy ✓

Sent 2 : good girl ✓

Sent 3 : boy girl good ✓

$$\text{Term Frequency} = \frac{\text{No. of rep of words in sentence}}{\text{No. of words in sentence}} \Downarrow \text{Sentences}$$

×

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

⇓ words

Term Frequency * Inverse Document Frequency

	Sent1	Sent 2	Sent 3
good	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{3}$
boy	$\frac{1}{2}$	0	$\frac{1}{3}$
girl	0	$\frac{1}{2}$	$\frac{1}{3}$

*

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

⇓

	f1 good	f2 boy	f3 girl
Sent 1	0 ✓	$\frac{1}{2} \times \log_e(3/2)$ ✓	0
Sent 2	0 ✓	0	$\frac{1}{2} \times \log_e(3/2)$ ✓
Sent 3	0 ✓	$\frac{1}{3}(\log_e(3/2))$ ✓	$\frac{1}{3}\log_e(3/2)$ ✓

O/p

Advantages

1. Intuitive
2. Word Importance is getting capture

Disadvantage

1. Sparsity
2. Out of vocabulary

good ✓	10
boy ✓	7
girls ✓	6
try	5
1	1
⋮	1
⋮	1
⋮	2

< 3

max.f = 3

good boy girls

[ ]

Assignment

Take any Text Dataset

{ Appy BOW, TF-IDF, ngram } ✓ ✓

Krishnaik06@gmail.com

Day 4 → NLP

① Word Embeddings

② Word2vec → CBOW (Continuous Bag of words) → Skipgram

③ Practical Implementation Using Python

① **Word Embeddings** → It is a techniques which converts words into vectors

Embedding Layer

Prerequisite : ANN

Word Embedding
- Count or Frequency
  - ① BOW
  - ② TF-IDF
  - ③ OHE
- Deep Learning Trained Model
  - ① Word2vec
    - CBOW
    - SKIP GRAMS

Sparse matrix [Huge dimension]

0 or 1s

0.23

BOW, TFID → ② Semantic Meaning {Honest, Good}

Vocabulary

② **Word2vec** : Feature Representation

① Limited Dimension
② Sparsity is reduced
③ Semantic Meaning

Semantic (related words)

Feature Represen	f1	f2	f3	f4	f5	f6
	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	0.01
Age	0.03	0.02	0.7	0.6	0.95	0.96
Food						
|						

300 dimension

KING − Man + Women = Queen

King [0.96 0.95] Man [0.95 0.98]

Queen [−0.96, 0.95] Women [−0.94 −0.96]

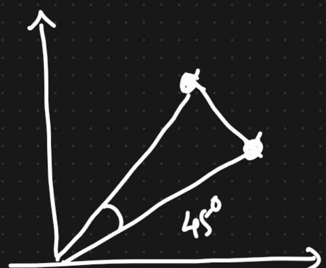
⇑

Matching

Distance = 1 − Cosine Similarity

$= 1 - 0.7071 \approx 0.29$

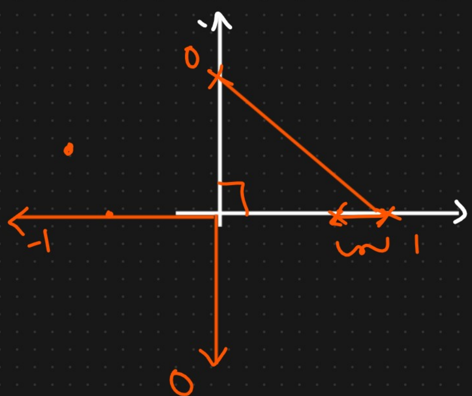
Cosine Similarity



Euclidean Distance

Cosine-Sim = $\cos\theta$

$\cos 45° = 0.7071 = \frac{1}{\sqrt{2}}$



Cos-Sim = $\cos 90 = 0$

Dist = $1 - 0$

$= 1$

(#) Word2vec

(i) CBOW {Continuous Bag of Words}

CORPUS ÷ DATASET TRAINING

O/P Target Word2vec

[KRISH CHANNEL $\boxed{\text{IS}}$ RELATED $\boxed{\text{TO}}$ DATA SCIENCE]

Context Window Size = 5 Context

TRAINING DATA

Word2vec

Text ⟶ Vectors {Semantic}

Independent feature ✓ O/P	O/P ✓ I/P
→ KRISH, CHANNEL, Related, To	IS
→ CHANNEL, IS, TO, DATA	Related
→ IS, Related, DATA, SCIENCE	To

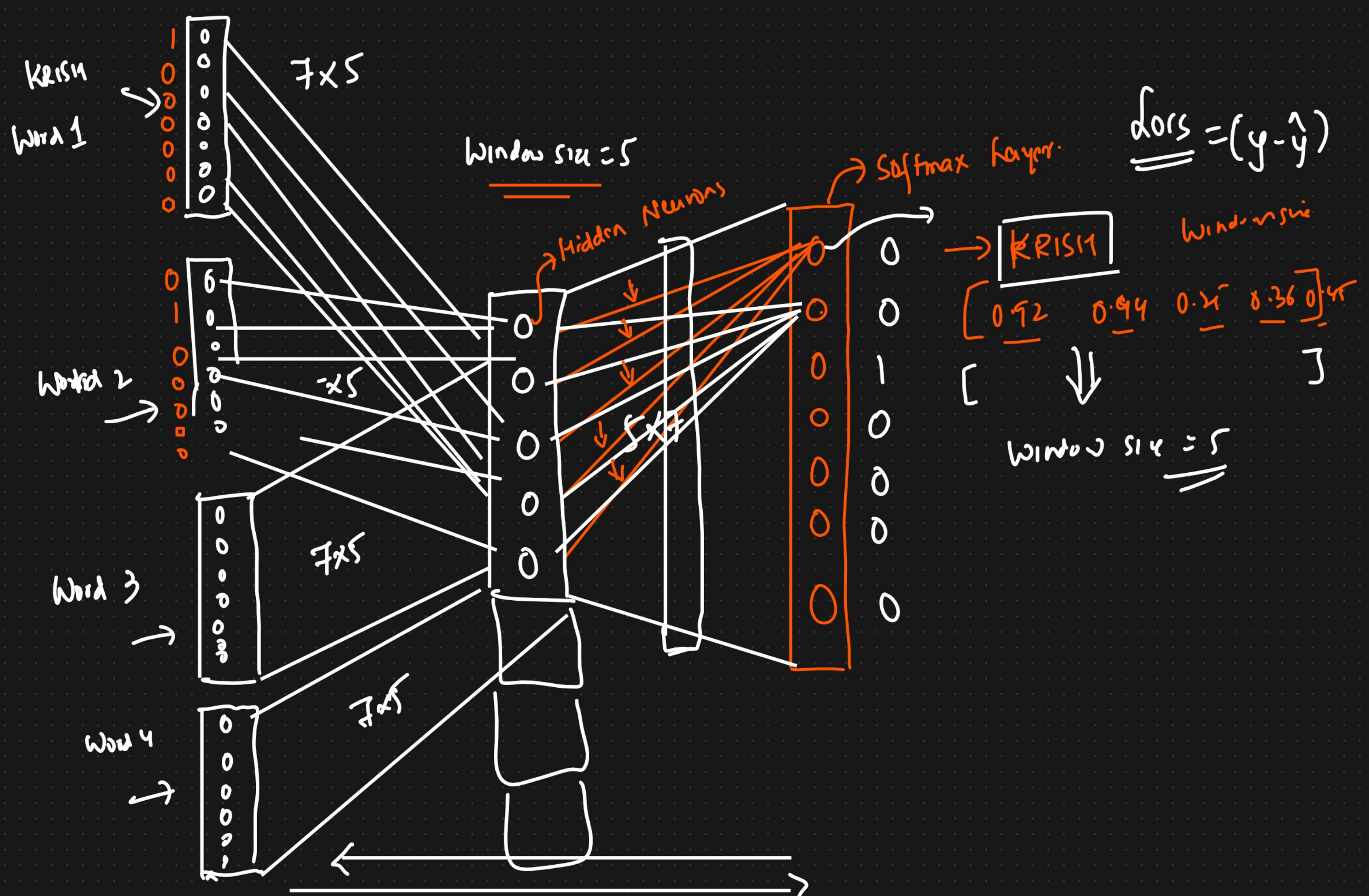
BOW

KRISH	1	0	0	0	0	0	0
Channel	0	1	0	0	0	0	0
Is	0	0	1	0	0	0	0
Related	0	0	0	1	0	0	0

⇓

CBow

(ANN) Fully Connected Layer



$\text{Loss} = (y - \hat{y})$

Window size = 300

⇓

$\boxed{300}$

② Skip GRAM

I/P	O/P
IS	KRISH, CHANNEL, Related, TO
Related	Channel, Is, To, Data.
To	IS, Related, Data, Science

Day 6 - NLP {Recurrent Neural N/W}

① BOW, TFIDF, Word2vec, Avg Word2Vec {Python Practical Implementation}

Word2vec → CBOW, Skip Gram

② <u>Deep Learning</u> → ① RNN ② LSTM RNN ③ GRU RNN ④ Bidirectional LSTM RNN
⑤ Encoders - Decoders ⑥ TRANSFORMERS ⑦ BERT

<u>Interview Question</u>

① Train vs Test Vs Validation →

Dataset

⇒ DATA Leakage.

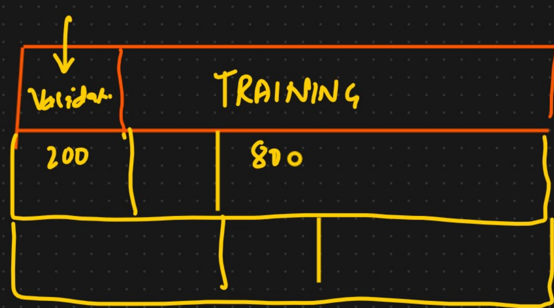
TRAINING → Model Training → Train, $\boxed{\text{Validation}}$ → Hyperparameter Tuning {Tuning the Model}

TEST } → Model Ever Sees this

TEST → Model Testing

$\frac{1000}{5} = 200$

CV=5



CROSS - VALIDATION
⇓
GRIDSEARCHCV
RANDOMIZED SEARCH CV

② Why RandomForest instead of Decision Tree {Answer}.

DT → {Low Bias High Variance}
↓

{ Low Bias   Low Variance } → Random Forest }.

① **Recurrent Neural N/W** ⇒ Text → Vectors

⇓

Machine Learning

Word Embedding ← Word2vec, AvgWord2vec

⇓

Chatbot : Question and Answer   Sequence of Words

Language Translation → Hindi → English

Text generation → A sentence $\xrightarrow{\text{Suggestion}}$ Completion of Sentences

⇑ Gramatically

GMAIL   AutoSuggestion

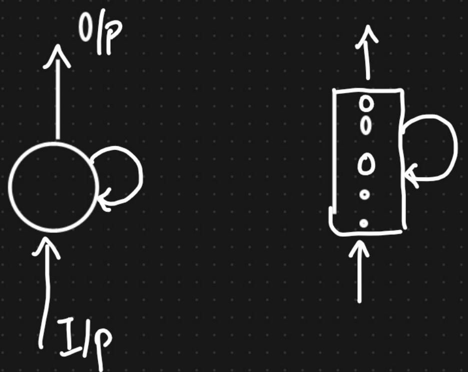
Deep Learning

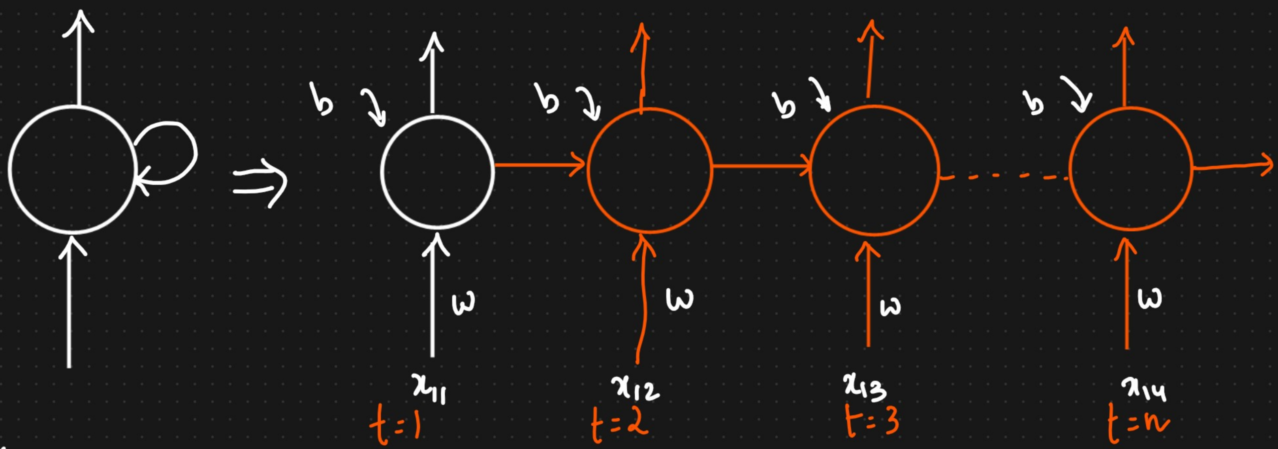
① RNN ② LSTM RNN ③ Transformers ④ BERT

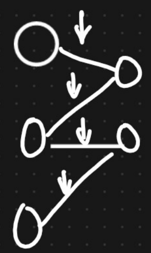
Word2vec
⇓ Deep Learning

Words ⇒ Vec

① **Recurrent Neural N/W**







Sentiment Analysis → The food is good → Positive

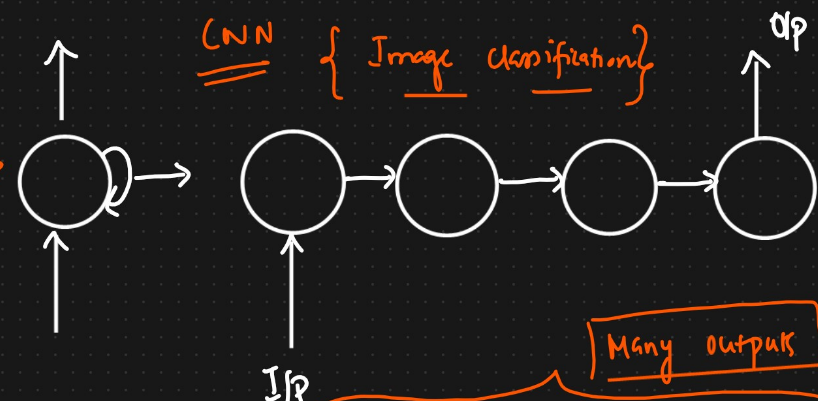
O/P

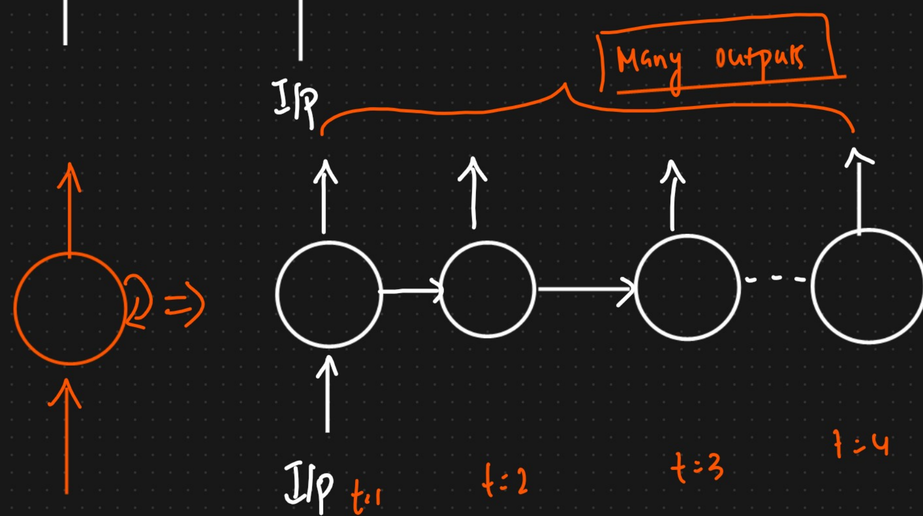
< x_{11} x_{12} x_{13} x_{14} >

x_{11} → Word2vec → Vectors → d = 300

Types of RNN

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





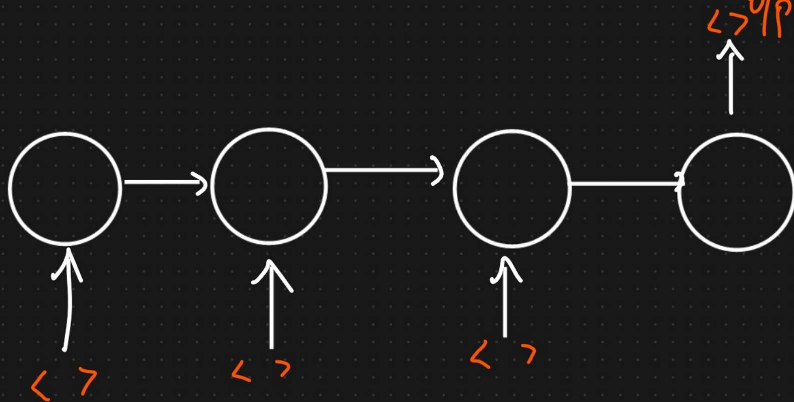
Eg: Music Generation, Text Generation

Google Search Suggestion, Music Recommendation

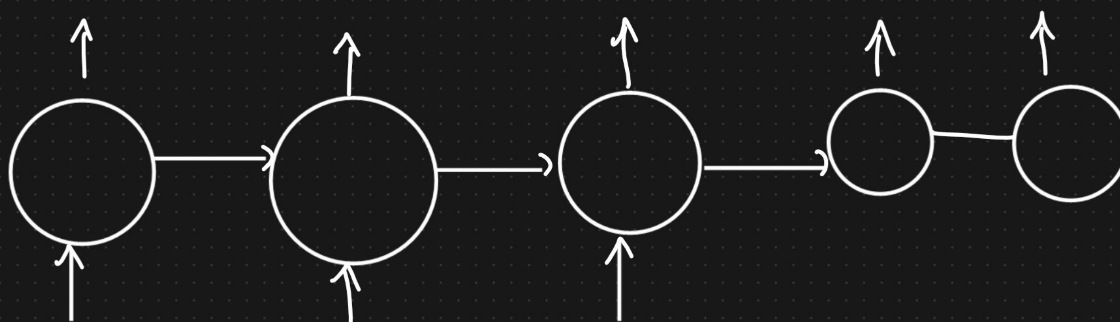
③ Many to One

Sentiment Analysis

{ Predict Next Day Sales }



④ Many to Many



① Language Translation
② Question answers
③ Chatbots.

{ ① Forward Propogation
② Backward Propogation }

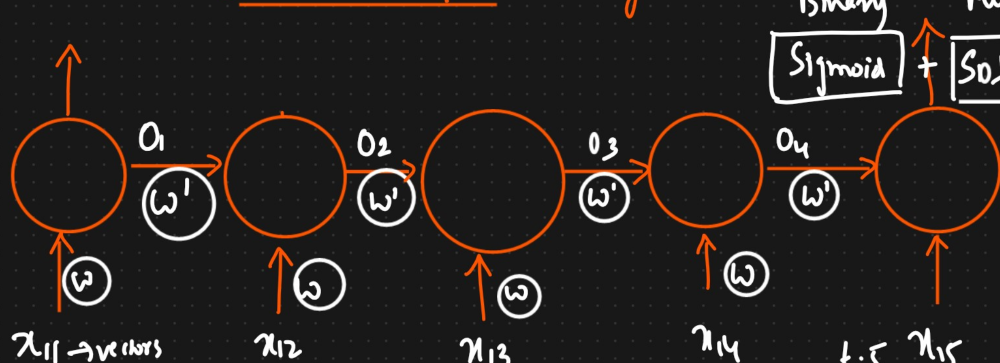
$\boxed{\text{Time Series Data}}$

① Forward Propogation In RNN

Sentiment Analysis → Many to One

Activation

Binary $\boxed{\text{Sigmoid}}$ + $\boxed{\text{Softmax}}$ Multiclass → $\hat{y}$



$Loss = (y - \hat{y})$

$t=1$ $t=2$ $t=3$ $t=4$

O/P

The food is very good — Positive.

x_{11} x_{12} x_{13} x_{14} x_{15}

$O_1 = f(x_{11} * w)$

$O_2 = f(x_{12} * w + O_1 * w_i)$

$O_3 = f(x_{13} * w + O_2 * w')$

Day 8 → Natural Language Processing

Agenda

① LSTM Recurrent Neural N/w {Indepth Architecture}

Problems with RNN

[Predict the next word]

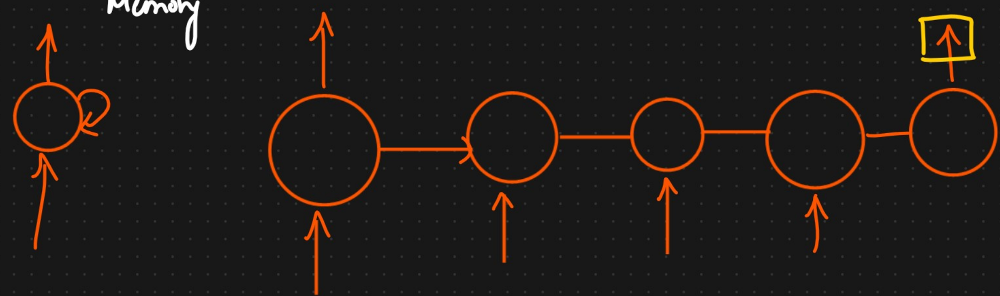
(Why kind of RNN)

Many to One

(Lhsezni)

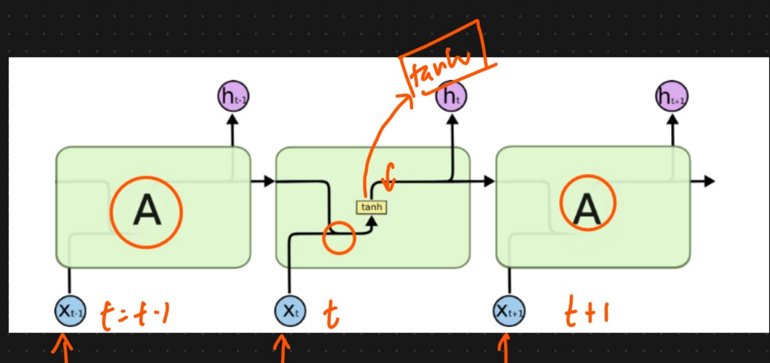
Short term Memory

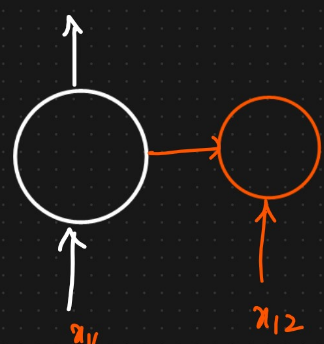
{ On $\boxed{\text{Sunday}}$ I want to eat pizza,
{ On $\boxed{\text{Monday}}$ I want to eat ___



LSTM RNN (Long short term Memory).

Represent RNN ←





→ TRADITIONAL RNN

→ $\boxed{\text{KRISH}}$ like pizza but my $\boxed{\text{friend}}$ like Burger

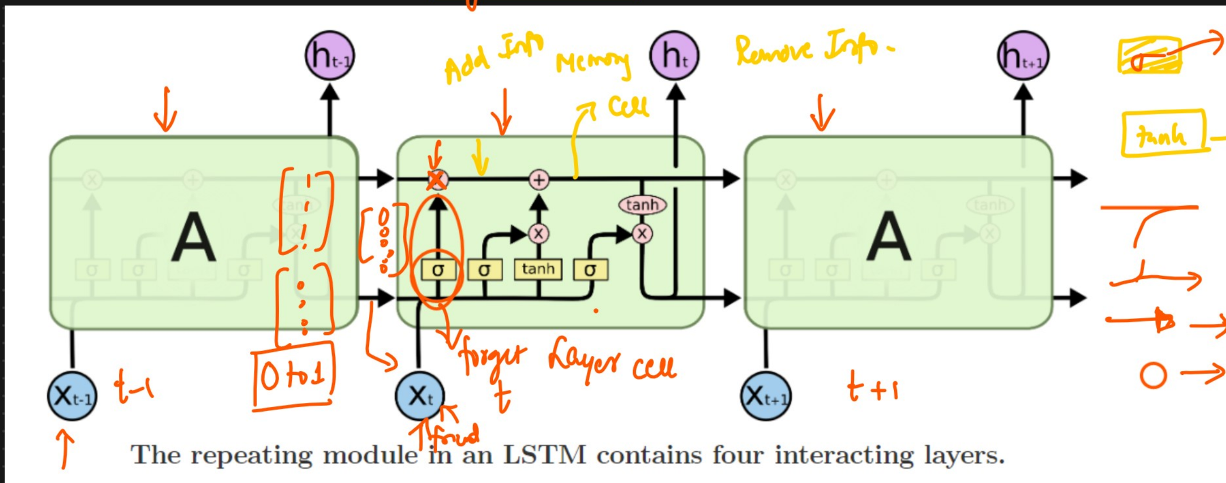
LSTM RNN

t=1 t=2

My name is $\boxed{\text{KRISH}}$ and my $\boxed{\text{friend}}$ $\boxed{\text{name}}$ is ___

① Long sentences

Long Short Term (LSTM)



The repeating module in an LSTM contains four interacting layers.

→ Neural N/w with Sigmoid Activation

tanh → NN with tanh Activation fn.

Concatenate

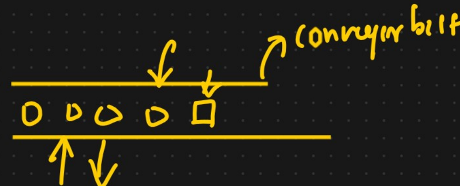
Copy

→ vector transfer.

O → pointwise operation

KRISH like pizza but his friend likes burger.

forget this info

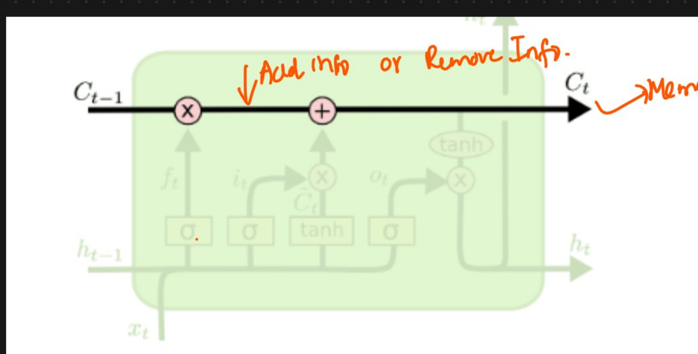


Memory → ① Add Information

② Remove Information.

Forgot Layer cell →

① Memory cell

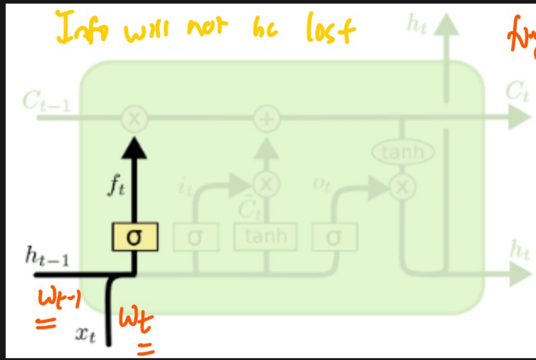


cell

② Forget Gate Layer

① → KRISH like pizza but he doesn't like burger

② → KRISH like pizza but his friend like burger



forgotten

0 to 1

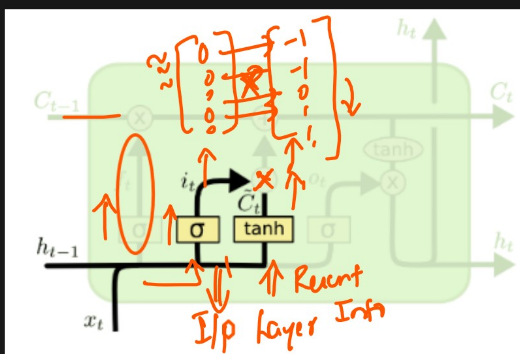
$[W_t, W_{t-1}]$

bias.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

$$\begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ \vdots \end{bmatrix} \quad \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix}$$

I/p Gate Layer



-1 to 1

Addition

$[W_t, W_{t-1}]$

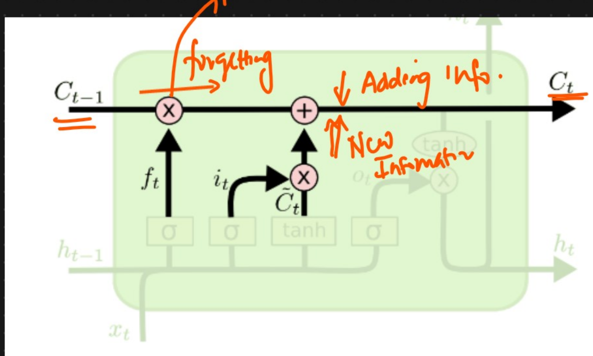
dot product

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

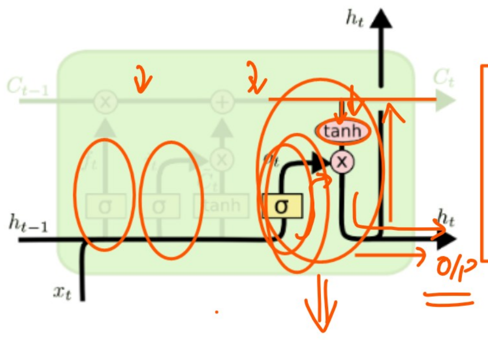
② → KRISH like pizza but his friend like burger

point wise dot product



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Output Gated Layer



$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

useless data is remove

only important dar.

Day 1 - Natural Language Processing (ML & Deep Learning)

Agenda

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

Prerequisites

1. Python ⟹ community Sunnons
2. Stats
3. Machine Learning Algo
4. ANN, Optimizers, Loss functions

Quiz : 5000 Rs

1. 2000 Rs INR
2. 1500 INR
3. 1500 INR

{KrishnaiK06}

SPAM CLASSIFICATION

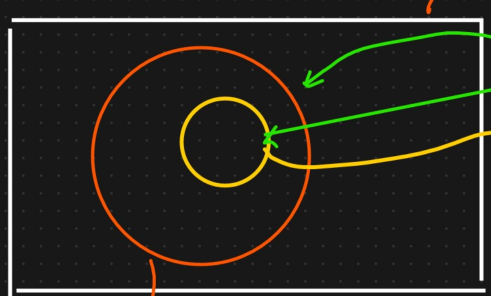
{Machine Translation}

Why NLP ?

[Google] → NLP

{Application}

AI



NLP

DL

Multi Layered NN

Machine Learning

NLP → Dataset → Text ⇓ Model ⇓ O/P

Roadmap of NLP

- BERT
- TRANSFORMERS
- Bidirectional LSTM, Encoders, Decoders, Attention Models
- Text Preprocessing – Word Embedding → Word2vec
- RNN, LSTM RNN, GRU RNN → Deep Learning
- ML Usecases
- Text Preprocessing 3 → Word2vec, Avg word2vec
- Text Preprocessing 2 → BOW, TFIDF, Unigrams, Bigrams
- Text Preprocessing 1 → Tokenization, Lemmatization, StopWords, Stemming

{ Tensorflow ✓
Pytorch ✓ }

{ NLTK
SPACY
TextBlob }

Huggingface

NLP

① Tokenization

ML Usecase

Mail Spam classifier

I/p features: Email body, Email Subject

Spam/ham

{ ① Tokenization → ② Stemming → ③ Lemmatization
↓ Stopwords }

You | won | 1000000 | $$

Dataset

	f1 Email body	f2 Email Subject	O/P
1)	You won 1000000 $$	Billionaire	Spam
2)	Hey KRISH, How ARE YOU	Hello	HAM
③	Credit Cards worth	Winner	Spam

Text Preprocessing

① Tokenization : { Sentence into words }   Sentence → Converts

[Hey buddy I want ~~to~~ go ~~to~~ your ~~of~~ house] → $\boxed{\text{not}}$ ←

⇓

Stopwords → yes

[.]

③ **Stemming** { Not have any meaning } Processing of reducing words to their Base word Stem

⇓

Root word or Baseform

historical
history
⇒ histori

⇑ Not have any meaning

finally
final
finalized
⇒ fina

⇑ Meaning is gone

going
goes
gone
⇒ go

⇑ { Meaningful word }

Advantages

① Stemming is really fast

Disadvantage

① It is removing the meaning of the word

④ **Lemmatization**

history
historical
⇒ history

finally
final
finalized
⇒ final

Advantages

1. Meaningful world

Disadvantage

1. It is slow.

Usecase

Stemming

1. Spam classification
2. Review classification

Lemmatization

1. Text Summarization
2. Language Translation
3. Chatbot

Step 1

Text Preprocessing

1. Tokenization
2. Stopwords
3. Stemming
4. Lemmatization

Step 2: Words → Vectors

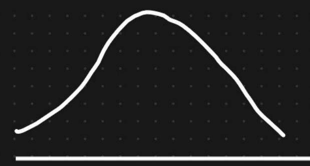
1. Bag of Words
2. TF-IDF
3. Word2Vec

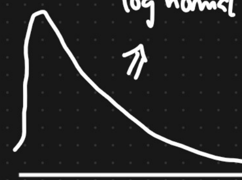
TF-IDF ⇓

Term Frequency – Inverse Document Frequency.

Day 7 → Natural Language Processing

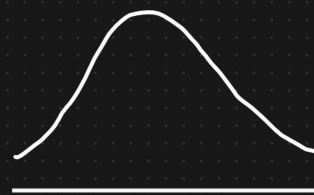
RNN ⇒ Forward Propogation



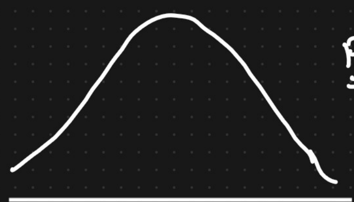


Stats : 1st Interview

⇒ Pareto Distribution ⇒



Box Cox Transform.



Follows ⇒ Normal or Gaussian Distri

⇓

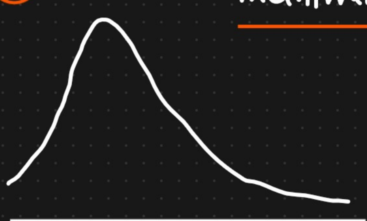
{ Q-Q plots }

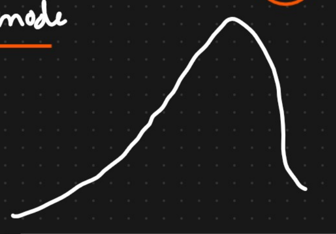
Standard Normal Distribution ??

A

mean, median & mode

B



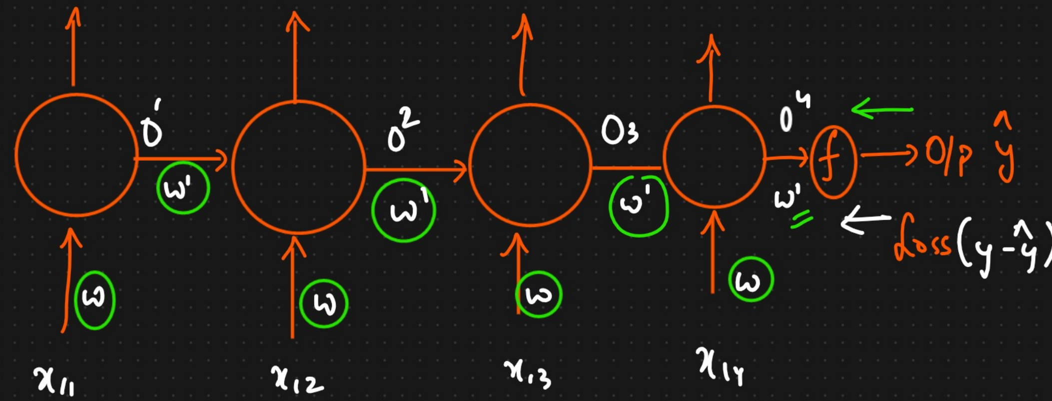


* Diff fit_transform and transform

fit_predict method ?

* Normalization And Standardization ??

RNN



$\langle x_{11}, x_{12}, x_{13}, x_{14} \rangle$ 0

$O_1 = f(x_{11} * w)$

$O_2 = f[(x_{12} * w) + O_1 w']$

$O_3 = f[(x_{13} * w) + O_2 * w_1]$

$O_4 = f[(x_{14} * w) + O_3 * w_1]$

Loss → stagnant

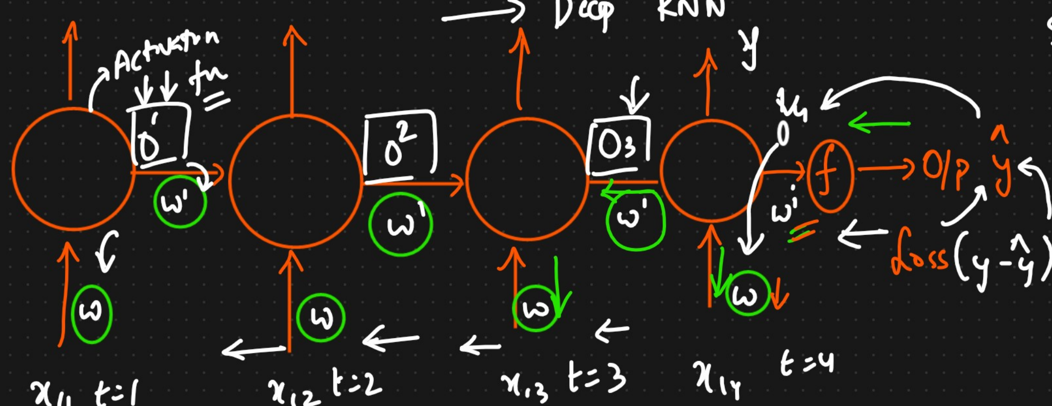
Epochs ⇒ Early Stopping

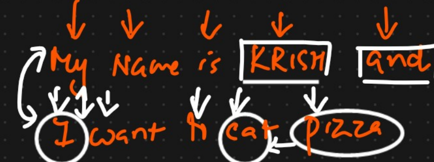
Backward Propogation

→ Deep RNN

Sigmoid ⇒ 0 to 1

Derivation(Sigmoid) ⇒ 0 to 0.25





Weight Updation formula

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

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

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

Chain Rule

$$\boxed{\frac{\partial L}{\partial w} = \frac{\partial L}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial O_4} \cdot \frac{\partial O_4}{\partial w}}$$

$$\boxed{\frac{\partial L}{\partial w'} = \frac{\partial L}{\partial \hat{y}} * \frac{\partial \hat{y}}{\partial O_4} * \frac{\partial O_4}{\partial w'}}$$

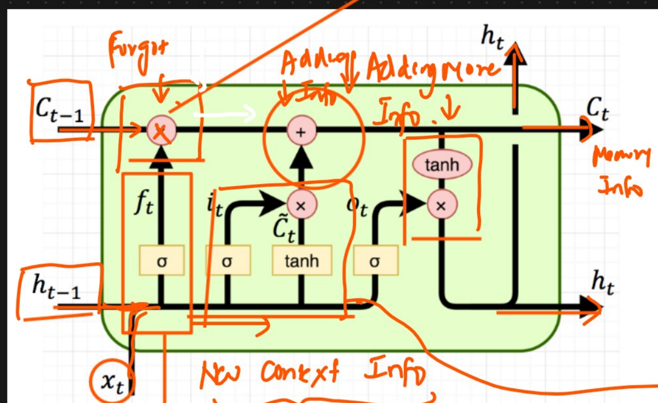
LSTM RNN ⇒ { Long Short term Memory Recurrent Neural N/W }

construction

{ ① Memory cell ② Forget cell ③ Input cell }

LSTM

→ Point Wise Operation



→ Input Layer

① Memory cell {Remember} {Forget}

KRISH Like DS

{ Yann Lecun like CNN - -

{ forget cell, memory cell, Input Layer }

Practical Appl Python} Forget cell

⇓ Detailed Architecture

Why LSTM RNN instead of RNN ??

① Vanishing Gradient or Dead Neuron

② Context Info. Deep RNN {Huge} ⇒ Gap