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 sessions
2. Stats
3. Machine Learning Algo
4. ANN, Optimizers, Loss functions

Quiz : 5000 Rs

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

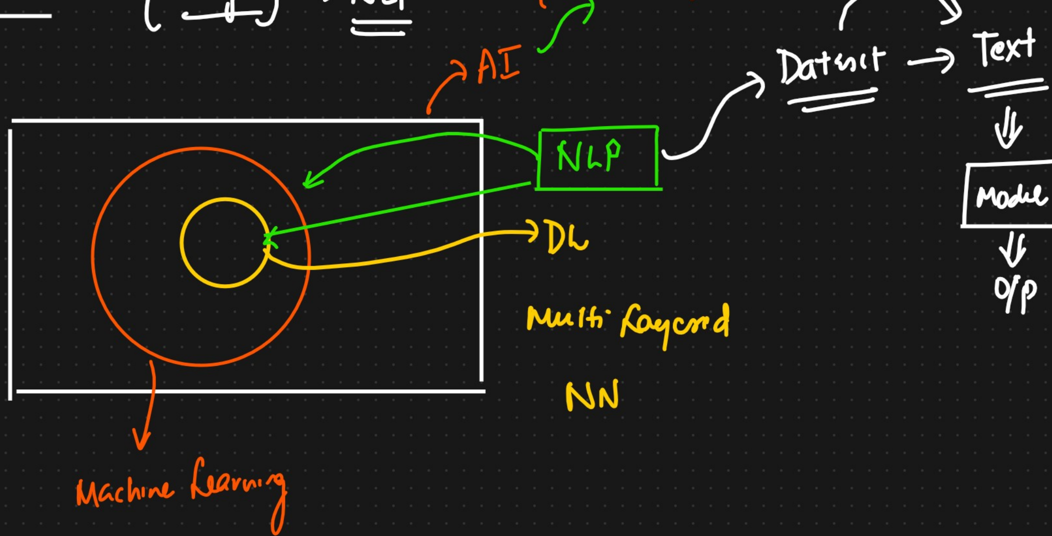
{KrishnaiK06}

SPAM CLASSIFICATION

{Machine Translation}

Why NLP ? [Google] → NLP

{Application}



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 }

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

You | won | 1000000 | $$

Text Preprocessing

① Tokenization : { Sentence into words }   Sentence → Converts

[Hey buddy I want ~~to~~ go ~~to~~ your house] → not

⇓

Stopwords → yes

[.]

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

historical, history ⟩ histori

⇓

Root word

or

Base form

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

Usecase

Stemming

① Spam classification
② Review classification

Disadvantage

① It is slow.

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

⇓

Term Frequency — Inverse Document Frequency.

Day 2 - NLP For ML And DL

Agenda

{NLTK}

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

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

⇒ CORPUS

size

9

food

A man eat Cat People watch KRISH YT

Out of vocabulary

⇒ CANNOT TRAIN THE MODEL

D1 → [[1 0 0 0],
[0 1 0 0],
[0 0 1 0],
[0 0 0 1]]

D2 - [[1 0 0]
[0 1 0],
[0 0 1]]

Advantages

① Simple to Implement
② Intuitive

Disadvantage

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

[Extra Test data]

② Bag of Words

Stop words

lower all the words case

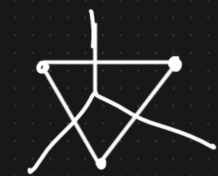
D1 → He is a good boy
D2 → She is a good girl ~~IT~~
D3 → Boy and girl are good

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



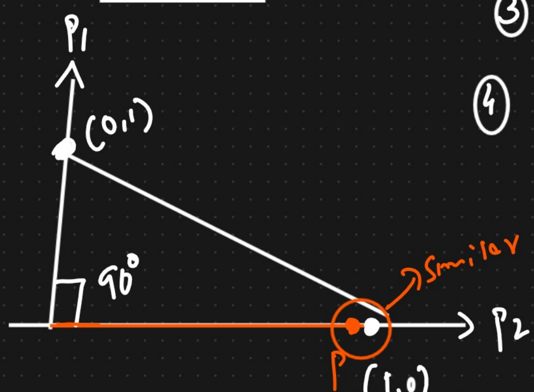
{ Euclidian Distance
Cosine Similarity.

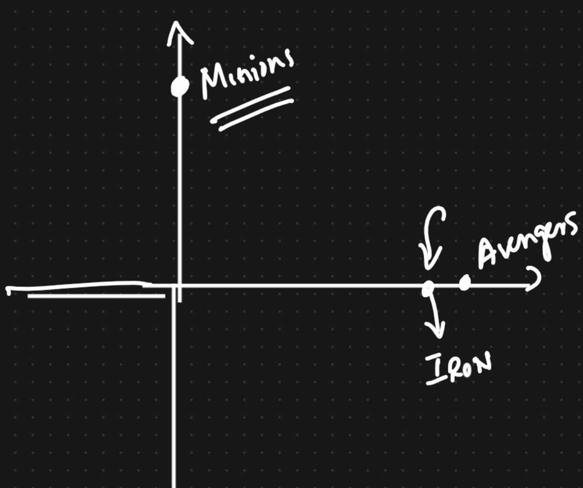
BoW ⇒ Binary BoW

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	f2	f3	f4	f5
	good	boy	girl	good boy	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

KRISH IS

f1	f2	f3	f4	FS
KRISH	IS	NOT	FEELING	WELL

Day 3 - NLP

Spam Classifier

Agenda

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

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	~~Cat~~
Sent 3	boy	girl	good

3 Frequency (Vocabulary)

	frequency
good	3
boy	2
girl	2

4

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

Not all similar

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 ✓

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

× ⇓ 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 captured

Disadvantage

① Sparsity

② Out of vocabulary

good ✓	10
boy ✓	7
girls ✓	6
toy	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, TFIDF → 2. Semantic Meaning { Honest, Good }

2. **Word2vec** : Feature Representation

Vocabulary

Semantic (related words)

1. Limited Dimension
2. Sparsity is reduced
3. Semantic Meaning

Feature Representation

	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 $[0.96 \quad 0.95]$ Man $[0.95 \quad 0.98]$

Queen $[-0.96, 0.95]$ Women $[-0.94 \quad -0.96]$

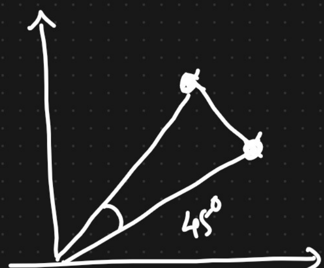
⇑

Matching

Distance = 1 − Cosine Similarity

$= 1 - 0.7071 \approx 0.29$

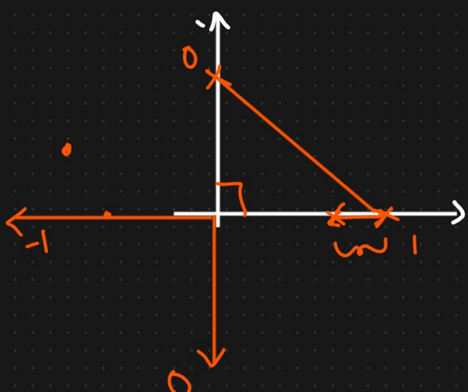
Cosine Similarity



Eucledian 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

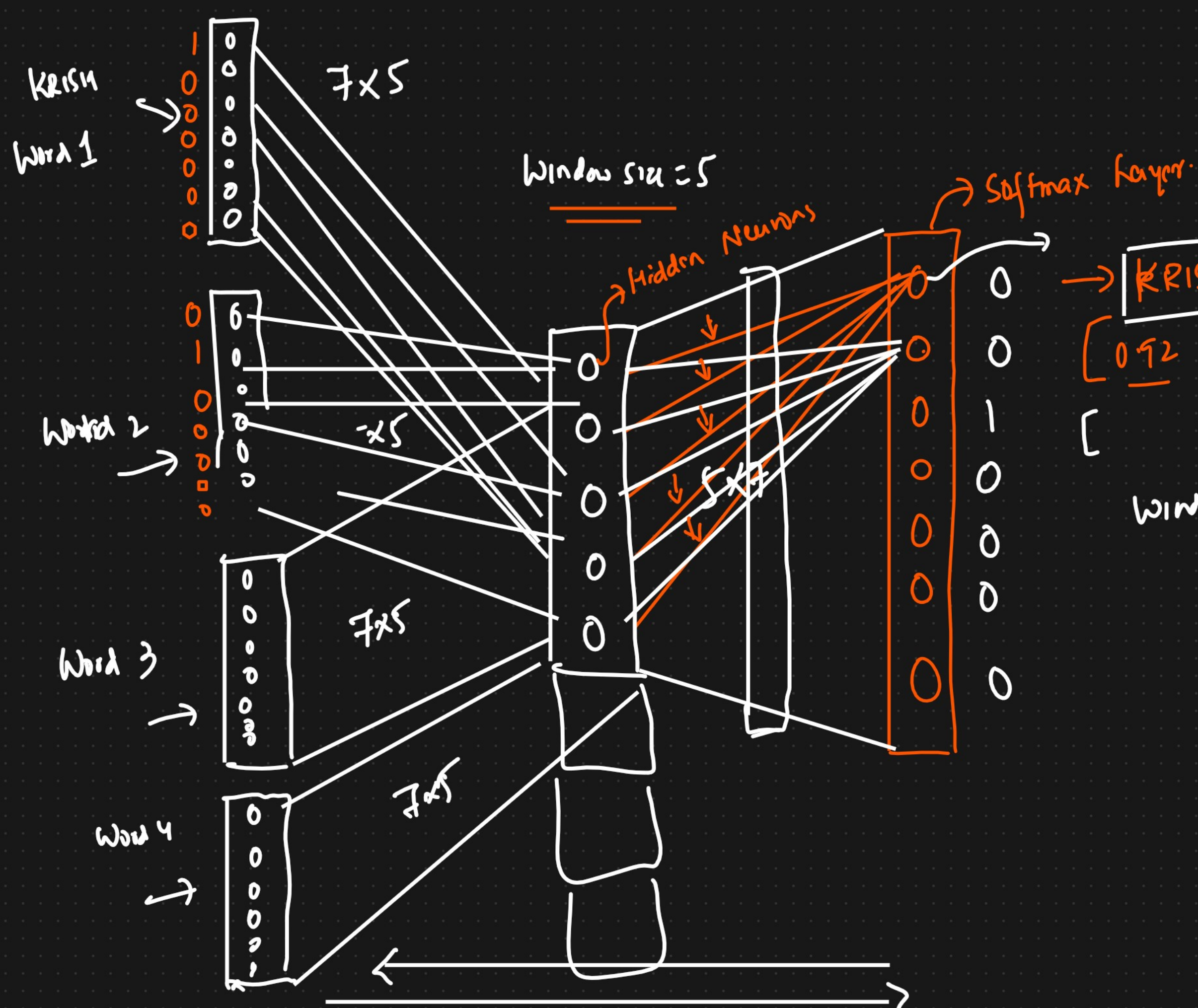
BOW

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



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

→ KRISH

Window size

[0.92 0.94 0.25 0.36 0.45]

⇓

Window size = 5

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

I/p

I want to eat pizza

↳ 300 dimension

O/p

Spam/ham

Word2vec

Skipgram CBOW

gensim

Word2vect

75%

Dataset

Word2vec

Word → vector

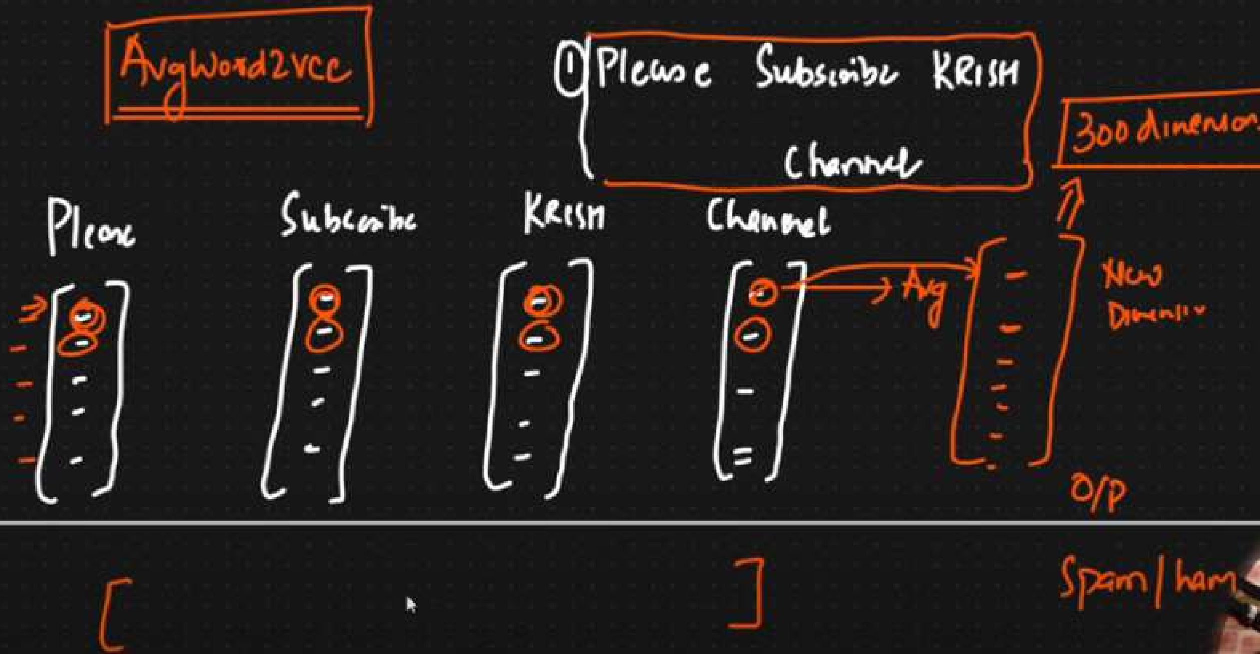
300 dimension →

Pretrained Model

{Google new 300}

Train this model from Scratch

KING Queen


AvgWord2vec
1 Please Subscribe KRISH Channel
300 dimension
Please
Subscribe
KRISH
Channel
Avg
New Dimension
O/P
Spam / ham

Day 6 - NLP {Recurrent Neural N/W}

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

Word2vec → CBoW, Skip Gram

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

Interview Question

① Train Vs Test Vs Validation →

Dataset → TRAINING, TEST

⇒ DATA Leakage.

TEST } → Model Ever Sees this

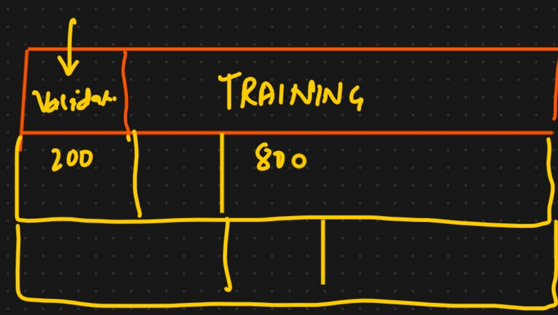
TRAINING ↓ Model Training → Train, Validation

Validation ↓ Hyperparameter Tuning { Tuning the Model }

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

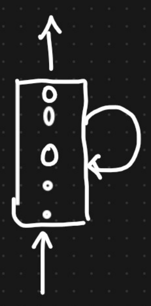
① RNN ② LSTM RNN ③ Transformers ④ BERT

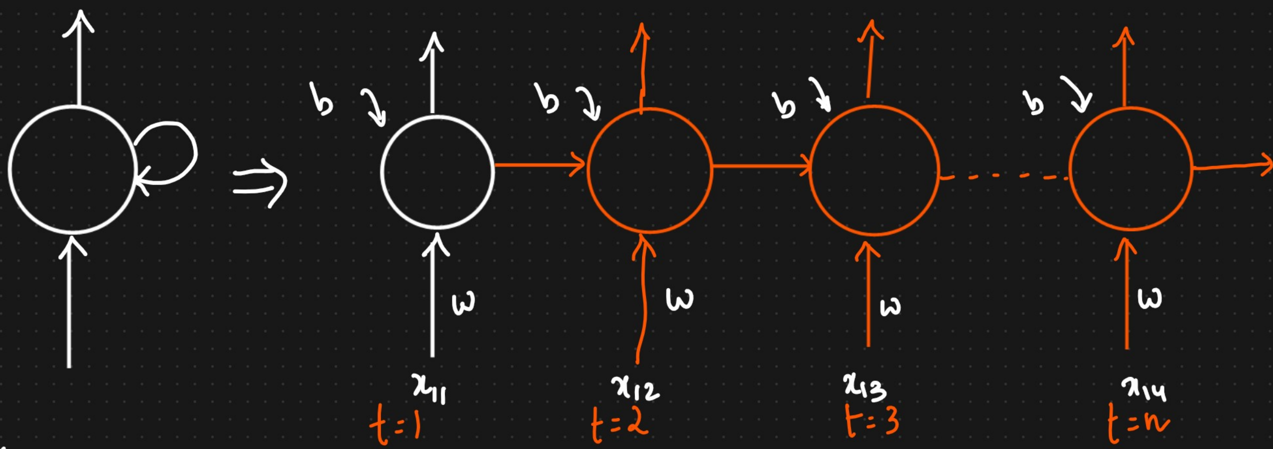
Word2vec

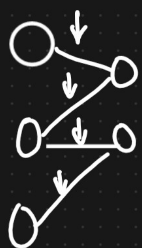
⇓ Deep learning

Words ⟹ Vec

① Recurrent Neural N/W







Sentiment Analysis → The food is good → Positive

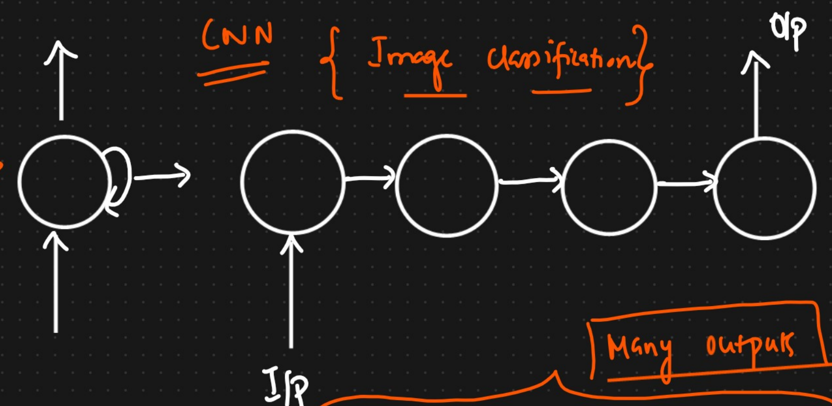
O/p

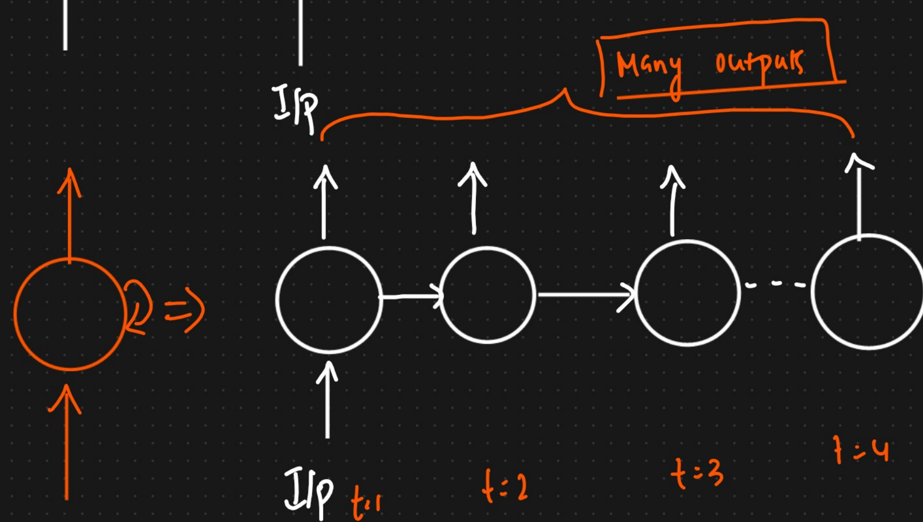
$< x_{11} \quad x_{12} \quad x_{13} \quad x_{14} >$

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





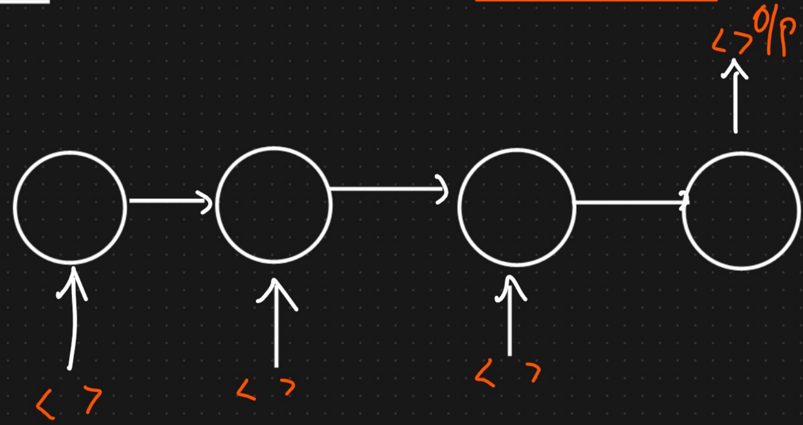
Eg: Music Generation, Text Generation

Google Search Suggestion, Movie Recommendation

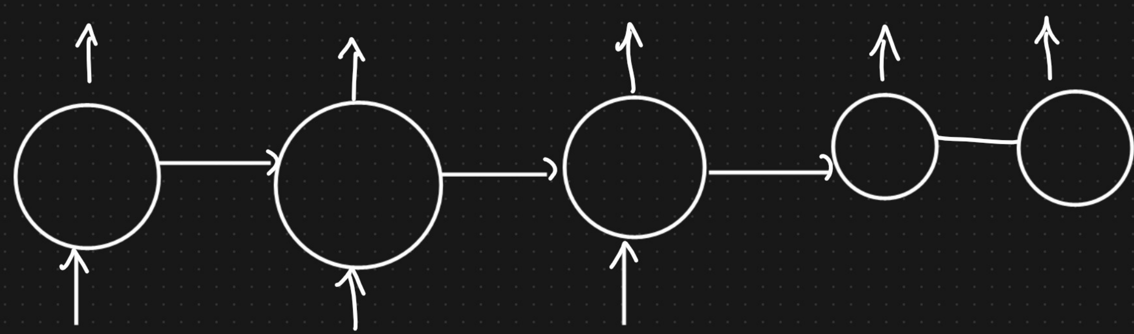
③ Many to One

Sentiment Analysis

{ Predict Next Day Sales }



④ Many to Many



① Language Translation
② Question answers
③ Chatbots.

{ ① Forward Propogation
② Backward Propogation }

Time Series Data

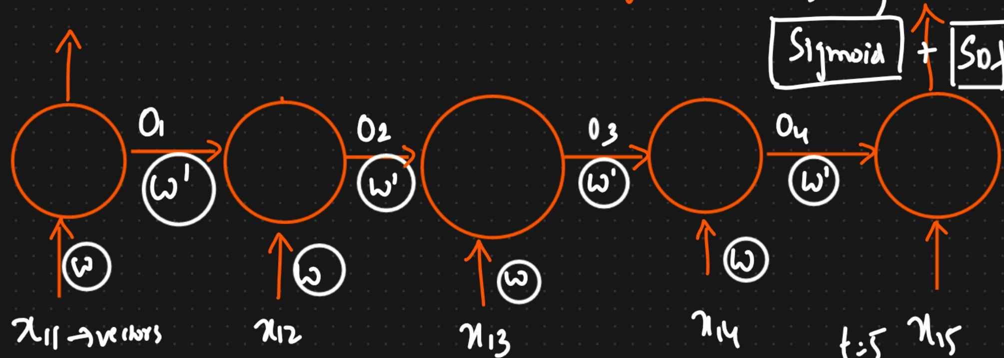
① Forward Propogation In RNN

Sentiment Analysis → Many to One

Activation

Binary y Multiclass → $\hat{y}$

Sigmoid + Softmax



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

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

The food is very good

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

O/P

Positive.

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

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

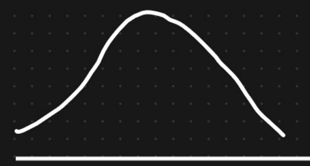
$$O_3 = f(x_{13} * w + O_2 * w')$$

Day 7 → Natural Language Processing

RNN ⇒ Forward Propogation

log normal

⇑



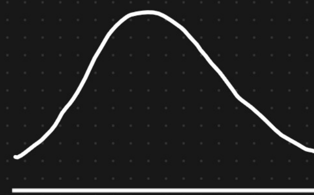
↓

Normal Distr

Stats : 1st Interview

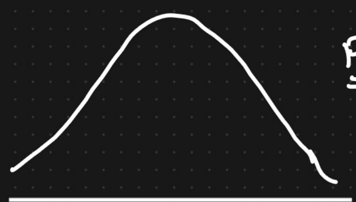
⇒ Pareto Distribution

⇒



Box Cox

Transform.



Follows

⇒ Normal or Gaussian Distri

⇓

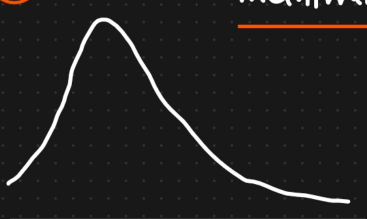
{ Q-Q plots }

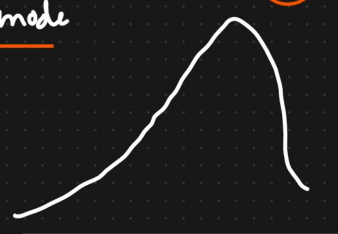
Standard Normal Distribution ??

A

mean, median & mode

B



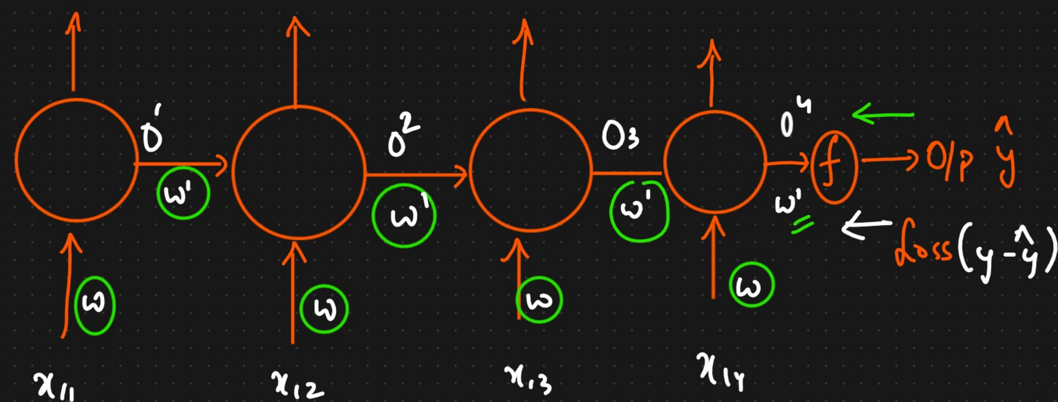


(*) Diff fit_transform and transform

fit_predict method ?

(*) Normalization And Standardization ??

RNN



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

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

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

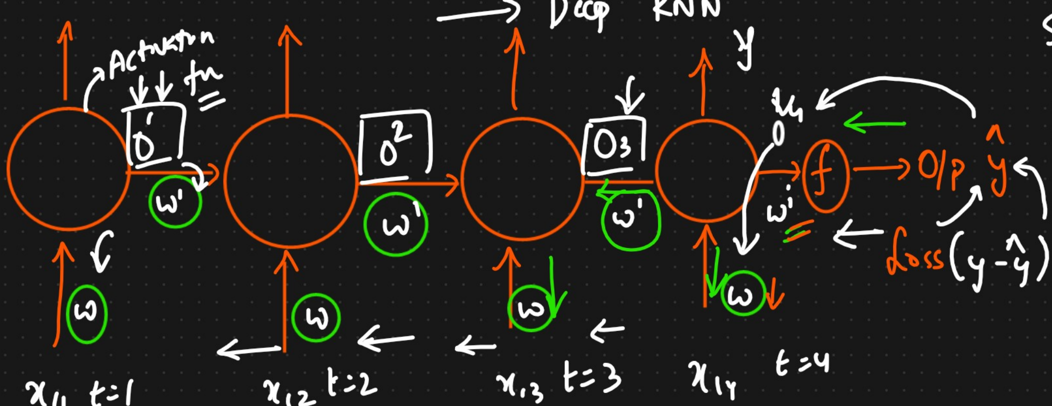
$O_3 = f[(x_{13} * w) + O_2 * w_1]$

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

Loss → stagnant
Epochs ⇒ Early Stopping

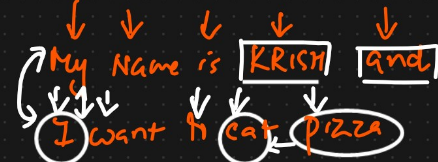
Backward Propogation

→ Deep RNN



Sigmoid ⇒ 0 to 1

Derivation(Sigmoid) ⇒ 0 to 0.25



Weight Updation formula

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

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

Chain Rule

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

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

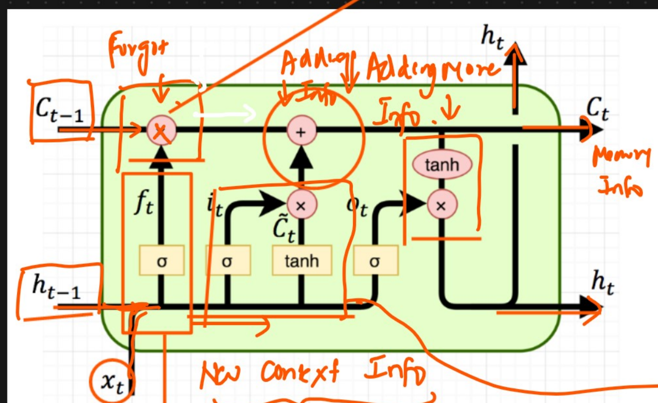
$$\boxed{\frac{\partial L}{\partial w'} = \frac{\partial L}{\partial \hat{y}} * \frac{\partial \hat{y}}{\partial O_4} * \frac{\partial O_4}{\partial w'}}$$

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

→ Reconstruction

{ ① Memory cell ② Forget cell ③ Input cell }

LSTM → Point Wise Operation



① Memory cell { Remember } { Forget }

KRISH Like DS

{ Yann Lecun Like CNN - -

→ Input Layer

{ 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

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)

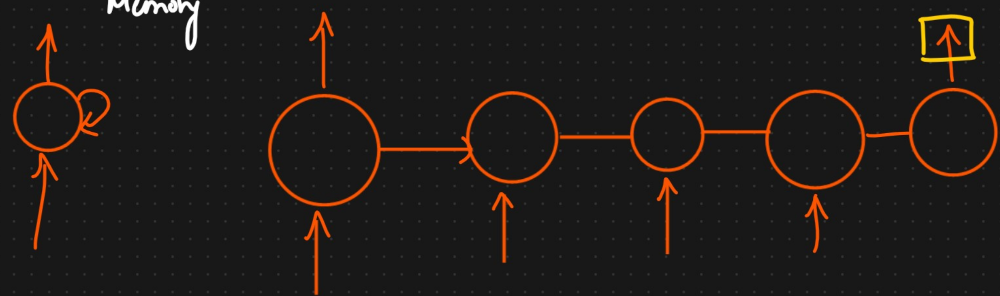
Many to One

(Unezni)

Short term Memory

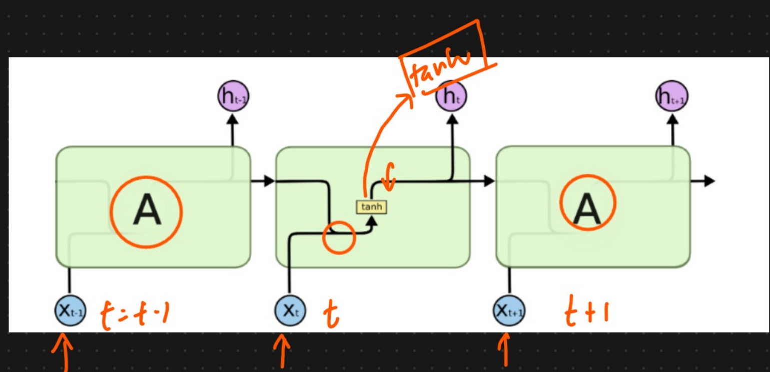
On Sunday I want to eat pizza,

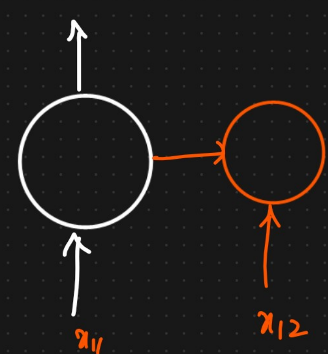
On Monday I want to eat ___



LSTM RNN (Long short term Memory).

Represent RNN ←





→ TRADITIONAL RNN

→ KRISH like pizza but my friend like Burger

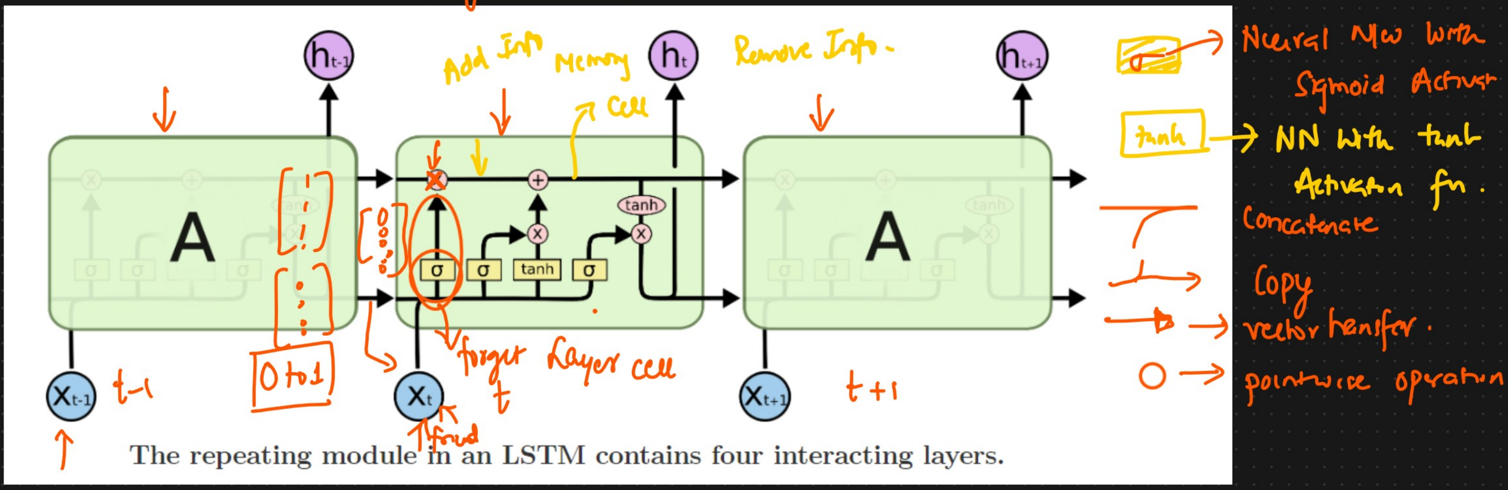
LSTM RNN

① Long sentences

t=1 t=2

My name is KRISH and my friend name is ___

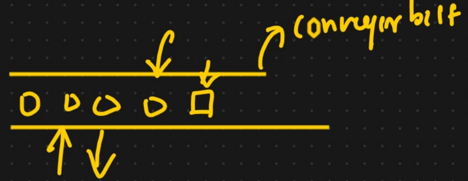
Long Short Term (LSTM)



The repeating module in an LSTM contains four interacting layers.

KRISH like pizza but his friend likes burger.

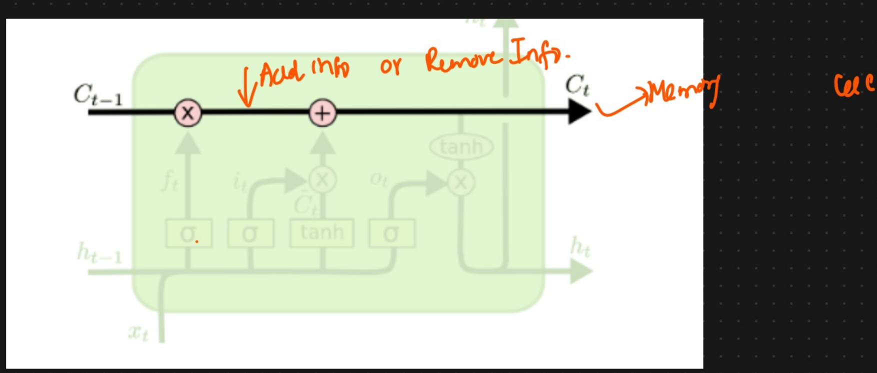
forget this info



Memory → ① Add Information
② Remove Information.

Forget Layer cell →

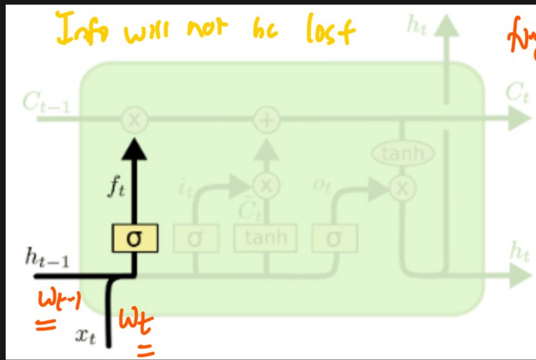
① Memory cell



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

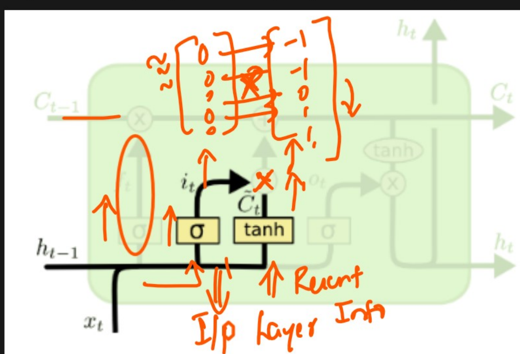
[W_t, W_{t-1}]

bias.

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

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

I/p Gate Layer



-1 to 1

[W_t, W_{t-1}]

Addition

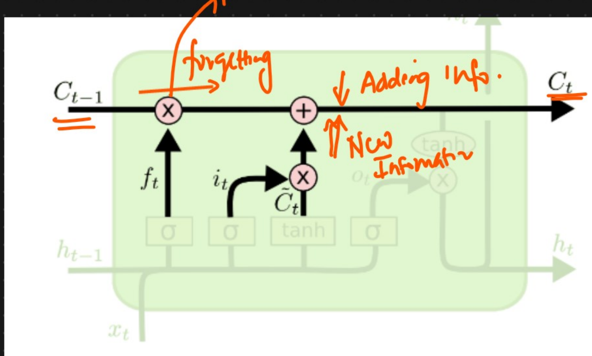
⊕ ⊗

dot product

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

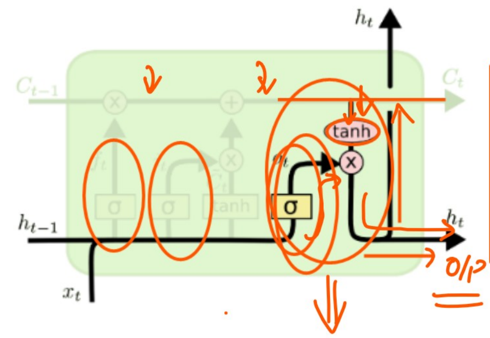
② ⟶ KRISH like pizza but $\boxed{\text{his friend}}$ like burger

point wise dot product



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

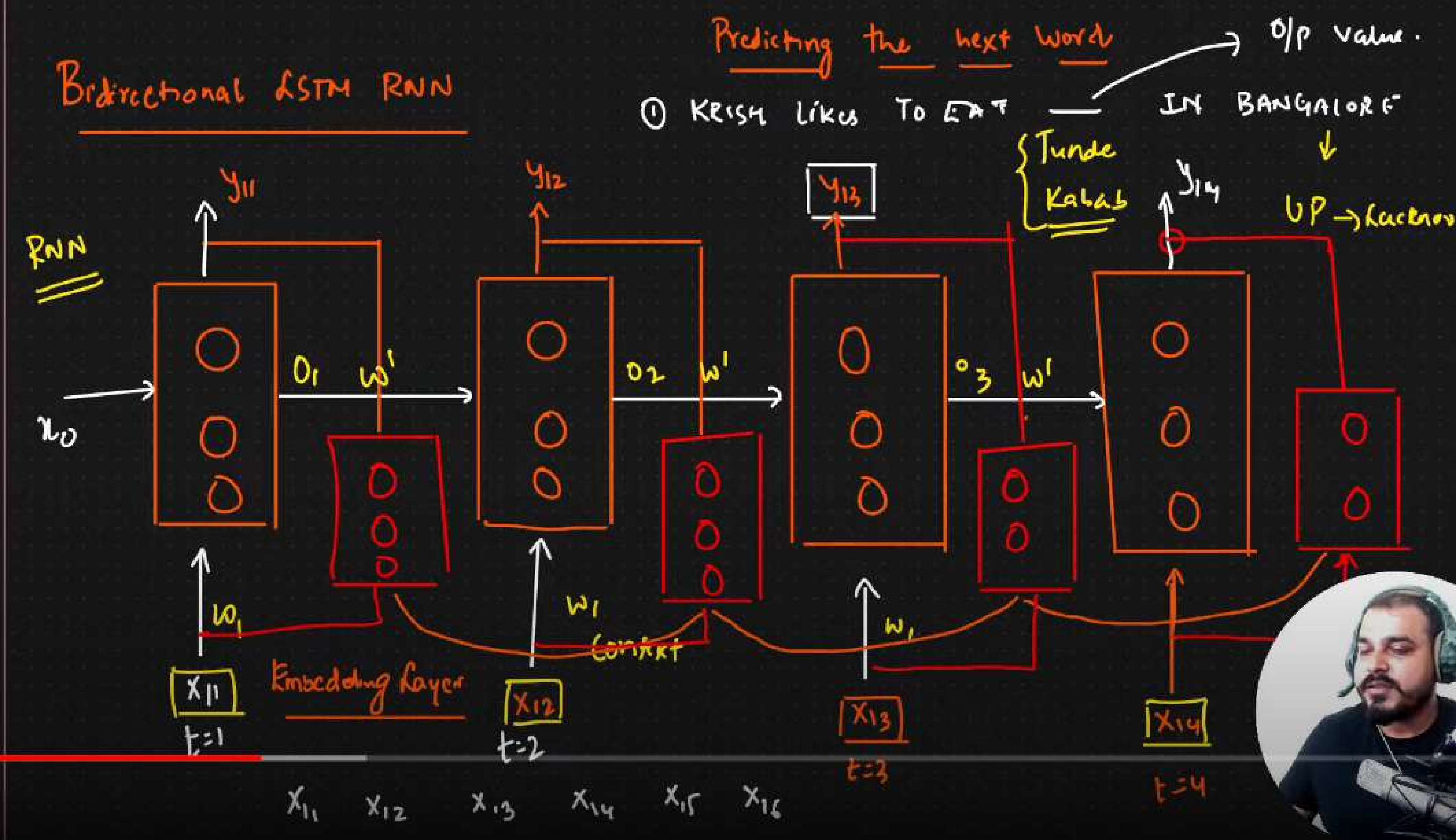
Output Gated Layer



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

useless data is remove

only important data.



Bidirectional LSTM RNN
Predicting the next word
O/P value.
① KRISH Likes TO EAT ___ IN BANGALORE
Tunde
Kabab
UP → Lucknow
RNN
y_{11}
y_{12}
y_{13}
y_{14}
x_0
O_1
w'
O_2
O_3
w_1
context
Embedding Layer
X_{11}
X_{12}
X_{13}
X_{14}
t=1
t=2
t=3
t=4
x_{11} x_{12} x_{13} x_{14} x_{15} x_{16}

Many to Many RNN