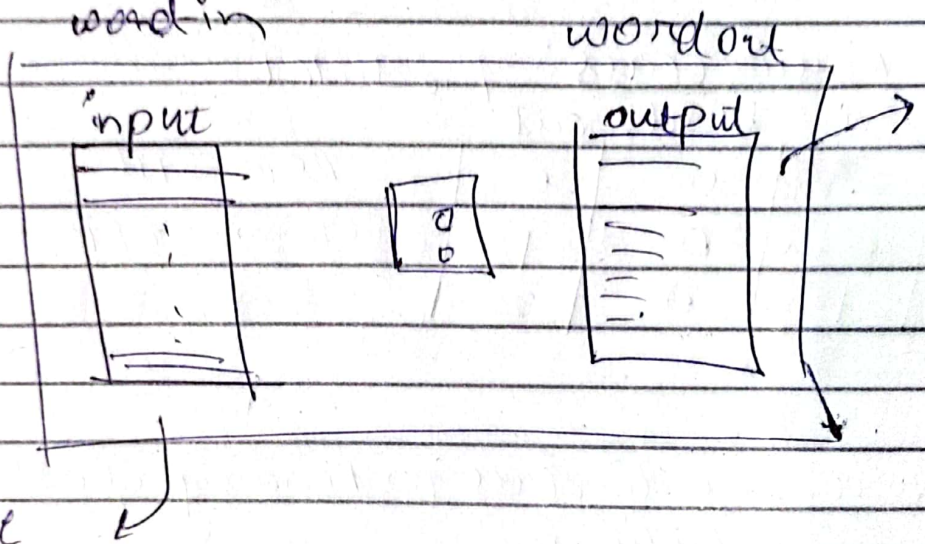
Lecture NO.8 (TF-IDF)

Lecture NO.9

→ word representation
↳ word embedding
↳ TF-IDF
↳ Sparse Vector

→ Embedding dimension are dimension controlled.



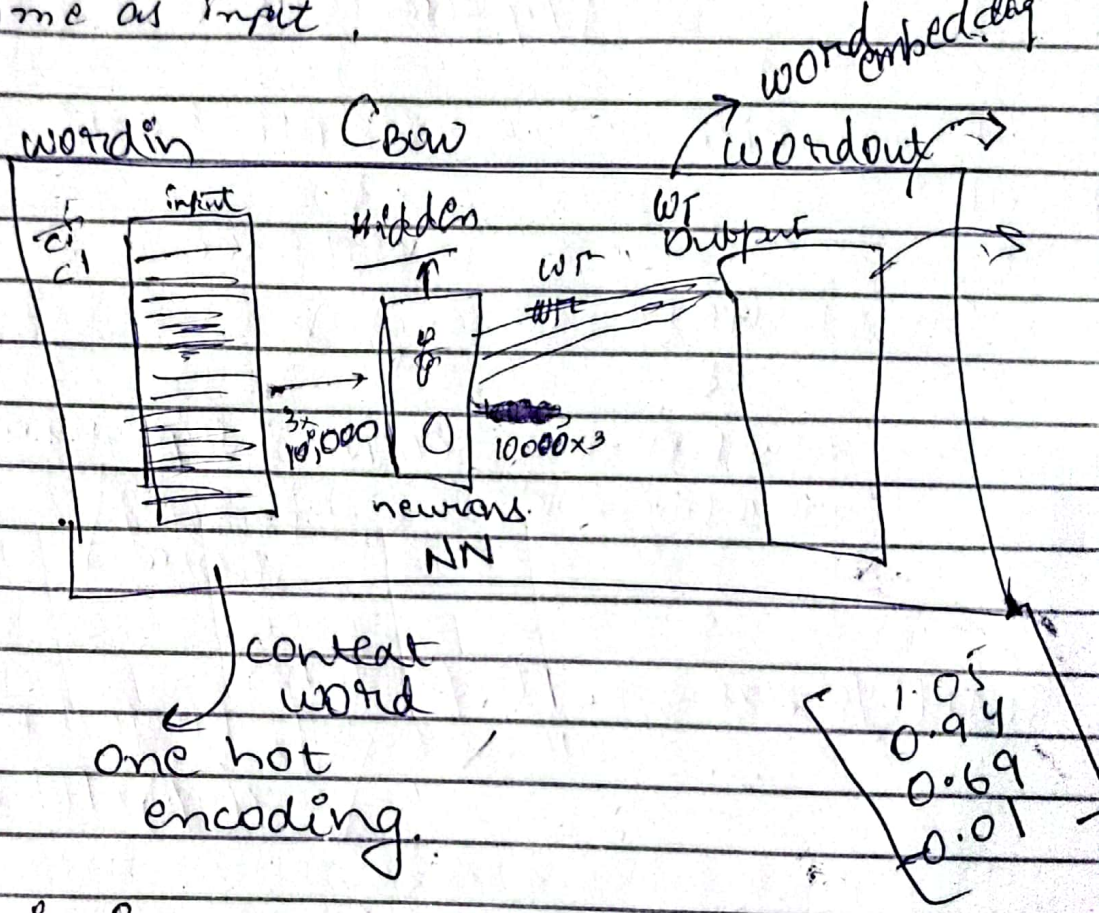
one hot encoding

→ Types of Words
-) Context word
-) Target word.

→ Types of word2Vec.
-) CBOW
-) ~~Skip-graph~~ Skip-gram

Self-Supervised!

•) In case of CBOW, context will come as input.



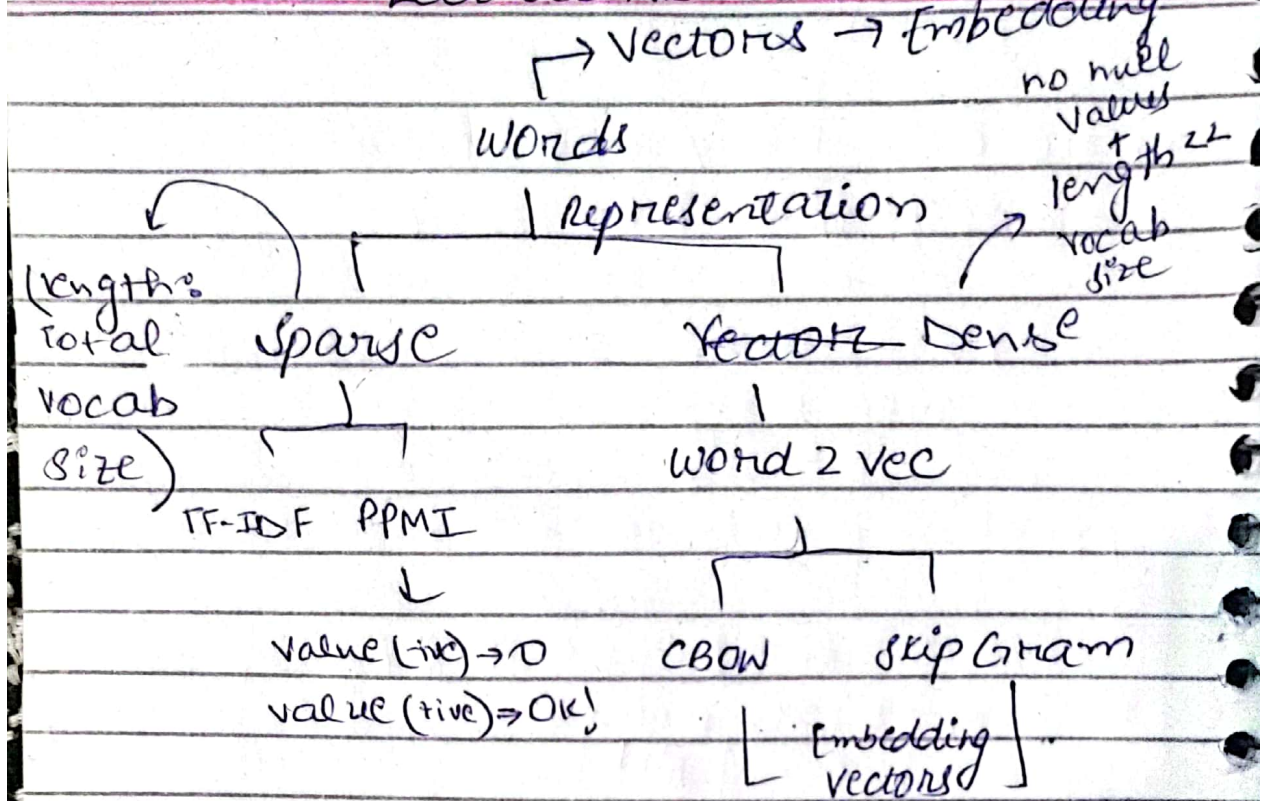
→ training the model based on neighbouring words.

→ learning → self supervision → no human involvement.

→ No. of hidden units = size of embedding

→ [0.7 0.2 0.1 ...] such that, SVM learning.

Lecture NO: 10



Ex:

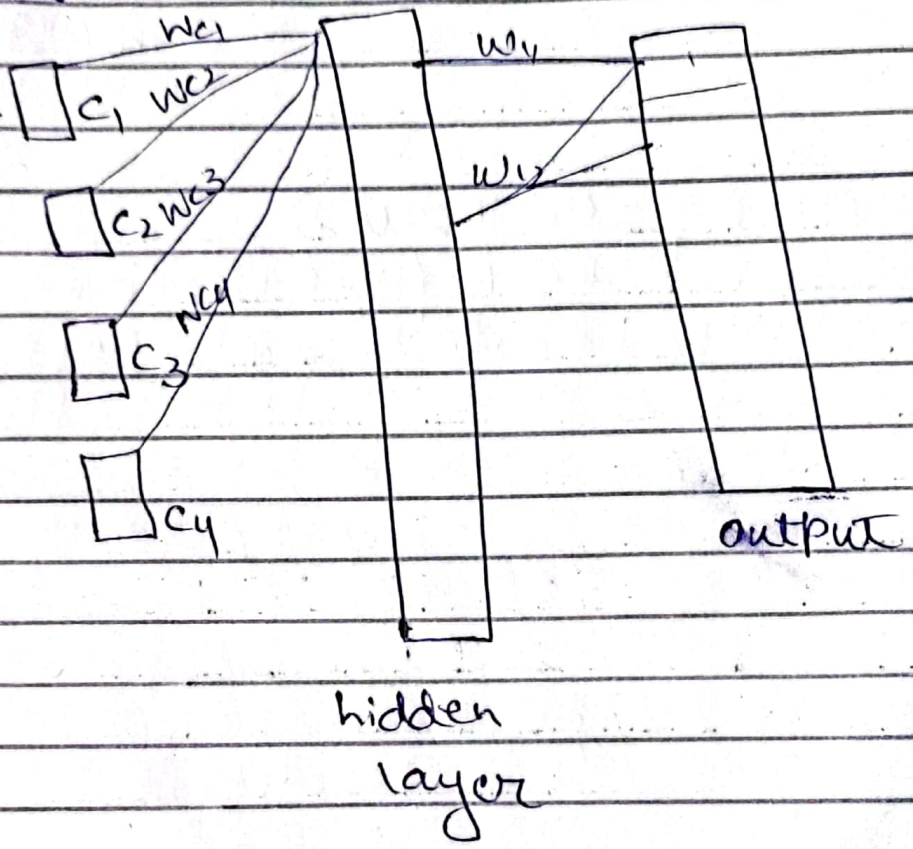
$$PPMI(info, data) = P(info, data)$$
$$= \text{to max}\left[\log_2 \frac{P(info, data)}{P(info) \times P(data)} > 0\right]$$

→ CBOW: (used when we have large corpus)
- → input: context words
- → output: target
- → Single hidden layer is used.
- → weights (hidden → output)

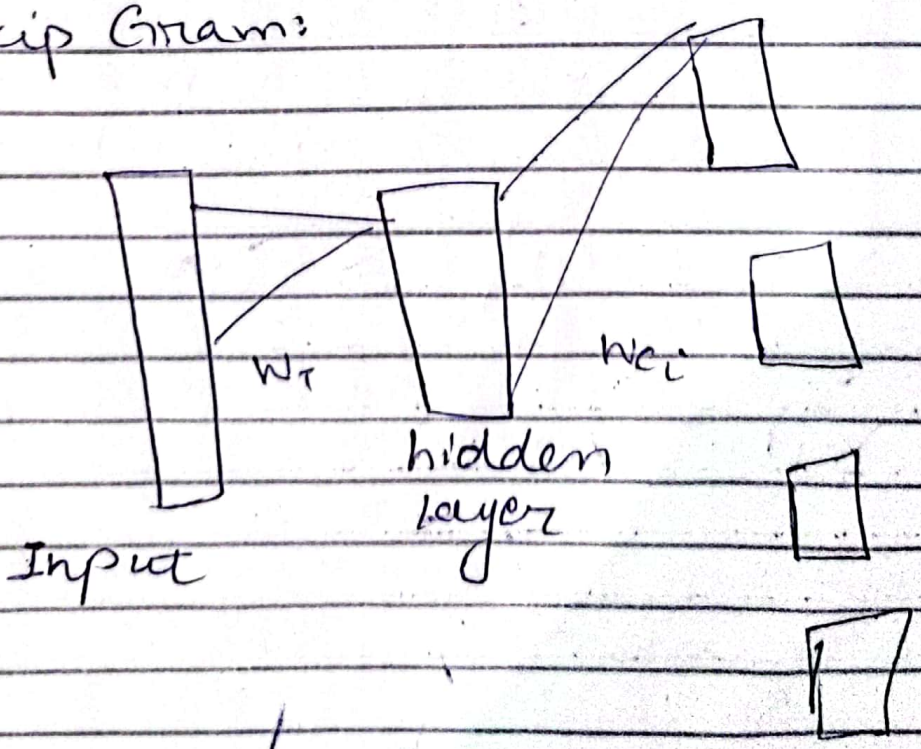
→ Skip Gram (Reverse of CBW):
- → input: target
- → output: context word.
- → used when we have corpus of small size
- → weights (input → hidden)

Wr * Wone hot encoding

• CBOW :



• Skip Gram:



→ Method used for training: Negative & samples

Word Best Representation
↳ Semantic Representation
↳ Dense

Lecture NO. 11

! Language Models !

Problems:

- Large Corpus → occurence of a certain word is less (Sparse)

In this case, Backoff is used.

- LM → looks into positional information by encoding / ~~information~~

NLM doesn't do this.

↓

Neural Language Model.

Lecture NO.12

NN (Neural Network)

Ex:

One-hot

V = { The, students, opened, their, Laptop, book, exam. }

|V| = 7

1	0	0	0	0	0	0
0	1	0	0	0	0	0
0	0	1	0	0	0	0
0	0	0	1	0	0	0
0	0	0	0	1	0	0
0	0	0	0	0	1	0
0	0	0	0	0	0	1

{The student opened their}

e_1 e_2 e_3 e_4

Embedding of 1st four: W_H 7×4

3	0	4	4
7	7	0	0
-8	0	-8	0
2	5	2	5

$E = [e_1\ e_2\ e_3\ e_4]$ (4 4 4 4)

$w_1 = 3 \times 16$

$b = 3 \times 1$

3×1

$w_2 = 7 \times 3$

$B = 7 \times 1$

16×1

7×1

a_1 | h_1

$a_1 = (W_1 \cdot I) + B$ $\quad h_1 = \sigma(a_1)$

$3\times16 \cdot 16\times1 + 3\times1$

3×1

16×1 input

$$\begin{bmatrix} 3 \\ 0 \\ 4 \\ 4 \\ 7 \\ 7 \\ 0 \\ 0 \\ -8 \\ 0 \\ -8 \\ 0 \\ 2 \\ 5 \\ 2 \\ 5 \end{bmatrix}$$

$$W_1 = \begin{bmatrix} 1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1 \\ 2\,2\,2\,2\,2\,2\,2\,2\,2\,2\,2\,2\,2\,2\,2\,2 \\ 5\,5\,5\,5\,5\,5\,5\,5\,-5\,-5\,-5\,-5\,-5\,-5\,-5\,-5 \end{bmatrix}_{3\times16}$$

$$B_1 = \begin{bmatrix} 7 \\ 0 \\ 0 \end{bmatrix}_{3\times1}$$

$a_1 = W_1 \cdot I + B_1$

$$\begin{bmatrix} 23 \\ -54 \\ -135 \end{bmatrix} + \begin{bmatrix} 7 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 30 \\ -54 \\ -135 \end{bmatrix}$$

$$h_1 = \sigma\left(\begin{bmatrix} 30 \\ -54 \\ -135 \end{bmatrix}\right) = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$a_2 = W_2 \cdot h_1 + B_2$ $\quad \hat{y} = \text{softmax}(a_2)$

$7\times3 \quad 3\times1 \quad 7\times1$

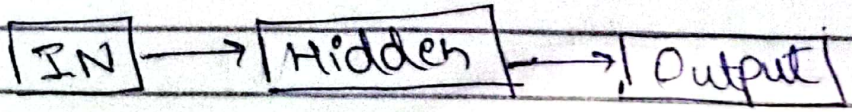
7×1

$$W_2 = \begin{bmatrix} 1 & 1 & 3 \\ 2 & 2 & 1 \\ 2 & 3 & 1 \\ 1 & 4 & 2 \\ 2 & 4 & 4 \\ 7 & 5 & 4 \\ 2 & 5 & 5 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

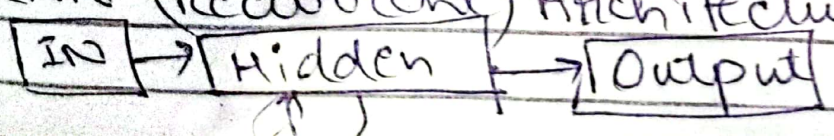
$$a_2 = \begin{bmatrix} 1 \\ 2 \\ 2 \\ 1 \\ 2 \\ 7 \\ 2 \end{bmatrix} \Rightarrow \hat{y} : \text{Softmax} \begin{bmatrix} 1 \\ 2 \\ 2 \\ 1 \\ 2 \\ 7 \\ 2 \end{bmatrix}$$

$$= \begin{bmatrix} 0.0024 \\ 0.0065 \\ 0.0065 \\ 0.0024 \\ 0.0065 \\ 0.9891 \\ 0.0065 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \Rightarrow \text{BOOKS}$$

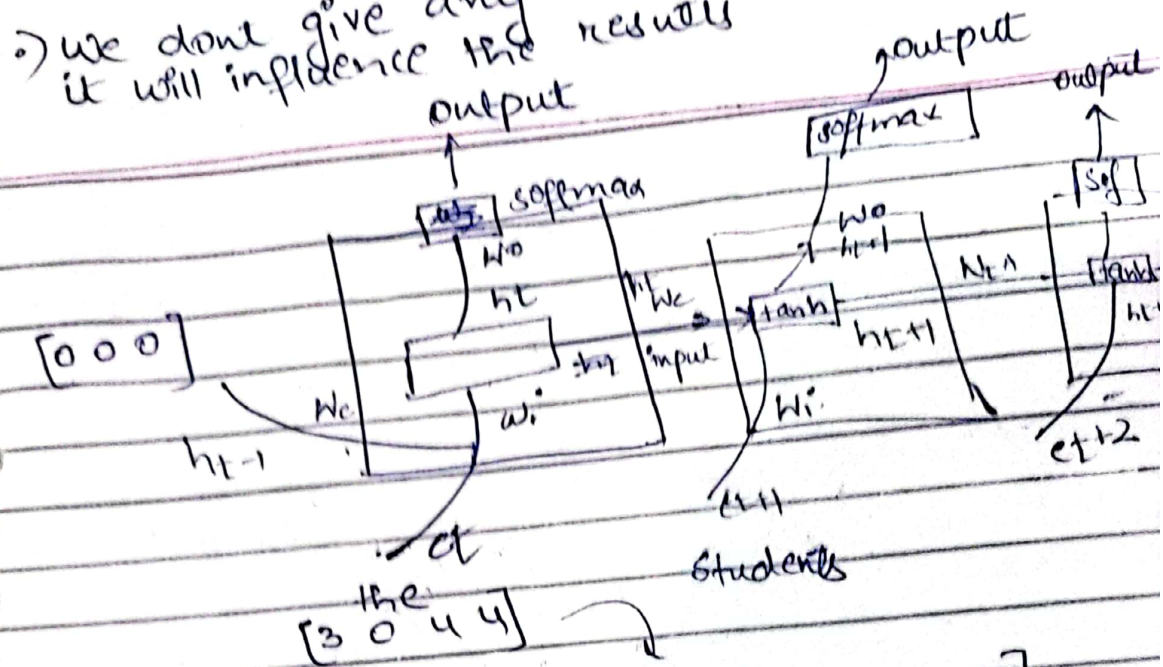
Simple NN Architecture



RNN (Recurrent) Architecture



*) we dont give any random value bcz it will influence the results



7×1

$I = \begin{bmatrix} 0\,0\,0\,0\,0\,0\,0 \\ h \quad \text{neurons from embedding} \end{bmatrix}$ $I = \begin{bmatrix} 0\,0\,0\,3\,0\,4\,4 \\ h \quad e \end{bmatrix}$

$I = [h_{t-1}, c_t]$

→ RNN types:
- one-to-one
- many-to-one
- one-to-many
- many-to-many

Formulizing

$$a_t = W_c \times h_{t-1} + W_i \times e_t + b_i$$

$$= W_i \cdot \begin{bmatrix} h_{t-1} \\ e_t \end{bmatrix} + b_i$$

W_i: 3×7 (h : 3×1, e : 4×1 → 7×1), b_i: 3×1

$h_t = \tanh(a_t)$ → 'The'

$a_{t+1} = W_i \cdot \begin{bmatrix} h_t \\ \text{The} \\ \text{student} \end{bmatrix} \cdot c_{t+1} + b_i$

$h_{t+1} = \tanh(a_{t+1})$ → 'student'

The student

* RNN gives a sequence
* computation is stored in hidden units. and is used further afterwards

$a_{t+2} = W_i \cdot [h_{t+1}, c_{t+2}] + b_i$

$h_{t+2} = \tanh(a_{t+2})$

↓

'The students opened',

$a_{t+3} = W_i \cdot [h_{t+2}, c_{t+3}] + b_i$

$h_{t+3} = \tanh(a_{t+3})$

↓

'The students opened their'

$c = \text{output} = [\text{softmax}\{W_o * h_{t+3}\} + b_o]$

vector [ ]

Disadvantage:

- More the words, More the context is disturbed

Lecture NO-13

RNN

V = { the, students, opened, their, laptops, books, exams }

One Hot

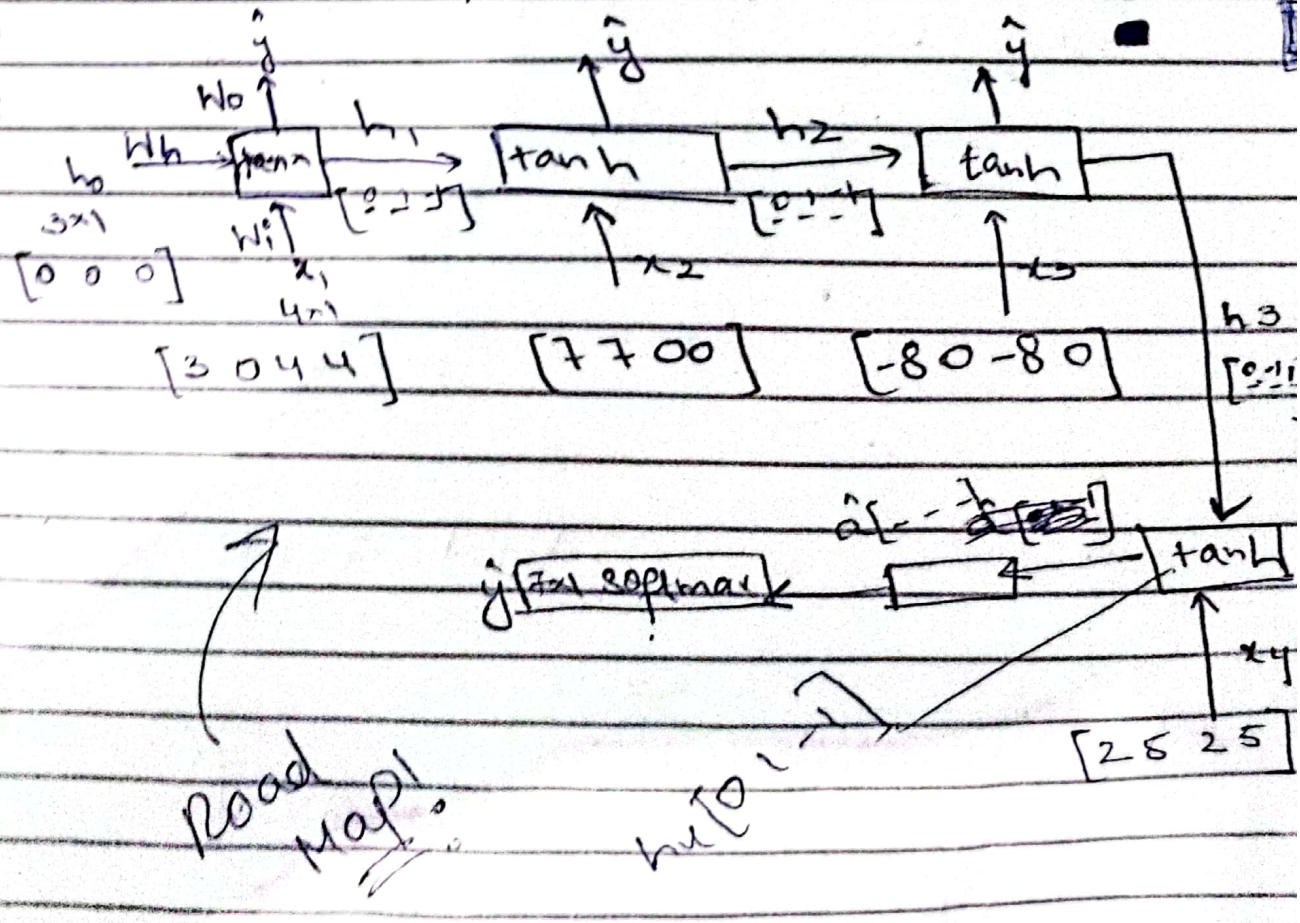
"the students opened their laptops."

Embedding of 'the' [3 0 4 4] x_1

" " 'students' [7 7 -8 0] x_2

" " 'their' [2 5 2 5] x_4

" " 'opened' [-8 0 -8 0] x_3



Road map!

ex:

$$I_1 = [h_0, x_1] = [0\ 0\ 0\ 3\ 0\ 4\ 4]^T \quad (7\times 1)$$

$T = 0$

$h_1 = [0\ 0\ 0]$ — Fixed.

$T = 1$

$$h_1 = \tanh(W \times I_1 + B) \quad (3\times 7,\ 7\times 1,\ 3\times 1)$$

$$= \tanh\left(\begin{bmatrix} 111.0000 \\ -3\ -3\ -3\ 1111 \\ 222 \quad -2\ -2\ -2\ -2 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 3 \\ 0 \\ 4 \\ 4 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}\right)$$

$$= \begin{bmatrix} 0 \\ 11 \\ -24 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \tanh\left(\begin{bmatrix} 0 \\ 11 \\ -15 \end{bmatrix}\right) \quad (3\times 1)$$

$$= \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}$$

$$I_2 = [h_1, x_2] = [0\ 1\ -1\ 7\ 7\ 0\ 0]$$

$T = 2$

$$h_2 = \tanh(W \times I_2 + B)$$

$$= \tanh\left(\begin{bmatrix} 111\ 0000 \\ -3\ -3\ -3\ 1111 \\ 222 \quad -2\ -2\ -2\ -2 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \\ -1 \\ 7 \\ 7 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}\right)$$

$$\tanh x = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

$$\cdot \tanh\left(\begin{bmatrix} 0 \\ 111 \\ -2 \end{bmatrix}\right) \cdot \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}$$

$t = 3$

$$I_3 = [h_2, x_3] = [0\ 1\ -1\ -8\ 0\ -8\ 0]$$

$$h_3 = \tanh\left(\begin{bmatrix} 111 & 0000 \\ -3\ -3\ -3 & 1111 \\ 222 & -2\ -2\ -2\ -2 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \\ -1 \\ -8 \\ 0 \\ -8 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 7 \end{bmatrix}\right)$$

$$\cdot \tanh\left(\begin{bmatrix} 0 \\ -16 \\ 32 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 7 \end{bmatrix}\right) = \tanh\left(\begin{bmatrix} 0 \\ -16 \\ 39 \end{bmatrix}\right)$$

$$= \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix}_{3\times 1}$$

$t = 4$

$$I_4 = [h_3, x_4] = [0\ -1\ 1\ 2\ 5\ 25]$$

$$h_4 = \tanh\left(\begin{bmatrix} 111 & 0000 \\ -3\ -6\ -3 & 1111 \\ 222 & -2\ -2\ -2\ -2 \end{bmatrix}\begin{bmatrix} 0 \\ -1 \\ 1 \\ 2 \\ -5 \\ 2 \\ 5 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 7 \end{bmatrix}\right)_{8\times 1}$$

$$\cdot \tanh\left(\begin{bmatrix} 0 \\ 111 \\ 28 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 7 \end{bmatrix}\right) = \tanh\left(\begin{bmatrix} 0 \\ 14 \\ -21 \end{bmatrix}\right) = \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix}$$

NLP

: Bias dependent on neurons.

$\hat{Y}_{7\times1} = W_1 \cdot B$ softmax$(W_0 \cdot h_i + B_0)$

B_0 (7×1)

$$= \text{softmax}\left(\begin{bmatrix} 1 & -2 & -3 \\ 1 & -2 & 3 \\ 1 & 1 & 1 \\ 1 & 2 & 2 \\ 1 & 3 & -5 \\ 1 & 2 & 2 \\ 1 & -2 & -2 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ -1 \end{bmatrix} + \begin{bmatrix} 2 \\ 0 \\ 0 \\ 0 \\ 2 \\ 0 \\ 0 \end{bmatrix}\right)$$

~~$$= \text{softmax}\left(\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 8 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 2 \\ 0 \\ 0 \end{bmatrix}\right)$$~~

$$= \text{softmax}\begin{pmatrix} 3 \\ -5 \\ 0 \\ 0 \\ 10 \\ 0 \\ 0 \end{pmatrix} = \begin{bmatrix} 9.1\times10^{-4} \\ \text{~~3.00×100~~} \\ 0 \\ 0. \\ 0.99 \\ 0 \\ 0. \end{bmatrix}$$

argmax → 1

7×1

[0 0 0 0 1 0 0] laptops

MSE cost

$\frac{1}{2}(Y-\hat{Y})^2$

W_{23}

Lecture NO:14

1x1

change the direction

'-' indicates the network/neuron to local optima

Back Propogation (Backward)

NN Architecture:

Loss func: MSE, Cross-Entropy

$\frac{\partial C}{\partial W_{23}} = \frac{\partial C}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial x} \cdot \frac{\partial x}{\partial H} \cdot \frac{\partial H}{\partial A} \cdot \frac{\partial A}{\partial W_{12}}$

$\rightarrow$ (1) $\xrightarrow{W_{12}}$ (2) $\xrightarrow{W_{23}}$ (3) (4) Actual

I — A|H — x|$\hat{Y}$ — e^3

A|H

Predicted

$\eta \rightarrow$ LR/learning rate

$\downarrow$ jumps/steps

Activation unit = sigmoid

Forward:

$A = I \cdot W_{12}$

$H = F(A) = \sigma(A)$

$X = H * W_{23}$

$\hat{Y} = F(X) = \sigma(X)$

Backward:

$W_{23}^{new} = W_{23}^{old} - \eta \frac{\partial \mathcal{L}(Y,\hat{Y})}{\partial W_{23}}$

$\frac{\partial C}{\partial W_{23}} = \frac{\partial C}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial x} \cdot \frac{\partial x}{\partial W_{23}}$ (C → cost)

For $\frac{\partial \mathcal{L}(Y,\hat{Y})}{\partial W_{23}}$, using chain Rule

$\hookrightarrow \frac{\partial a}{\partial c} = \frac{\partial a}{\partial b} \cdot \frac{\partial b}{\partial c}$

C = cost = cross-entropy

$\hookrightarrow$ loss $\rightarrow -y \log \hat{y} - (1-y)\log(1-\hat{y})$

$\rightarrow \frac{\partial C}{\partial \hat{y}} = \frac{\partial -Y \text{ loss}(\hat{Y}) - (1-Y)\log(1-\hat{Y})}{\partial \hat{Y}}$

$= -Y \times \frac{1}{Y} \times (+1) - (1-Y) \cdot \frac{1}{(1-\hat{Y})} \times (-1)$

$$= \frac{-Y}{\hat{Y}} + \frac{(1-Y)}{(1-\hat{Y})}$$

$$= \left[\frac{(1-Y)}{(1-\hat{Y})} \quad \frac{-Y}{\hat{Y}}\right]$$

$$\rightarrow \frac{\partial \hat{Y}}{\partial x} = \hat{Y} \times (1-\hat{Y})$$

↓

Sigmoid

$$\frac{\partial C}{\partial \hat{Y}} \cdot \frac{\partial \hat{Y}}{\partial x} = \left[\frac{(1-Y)}{(1-\hat{Y})} \quad \frac{-Y}{\hat{Y}}\right] \cdot \left[\hat{Y} \times (1-\hat{Y})\right]$$

$$= \frac{(1-Y)}{(1-\hat{Y})} \cdot \hat{Y}(1-\hat{Y}) - Y \cdot \hat{Y}(1-\hat{Y})$$

$$= (1-Y)\hat{Y} - Y(1-\hat{Y})$$

$$= \hat{Y} - Y\hat{Y} - Y + Y\hat{Y}$$

$$= \hat{Y} - Y$$

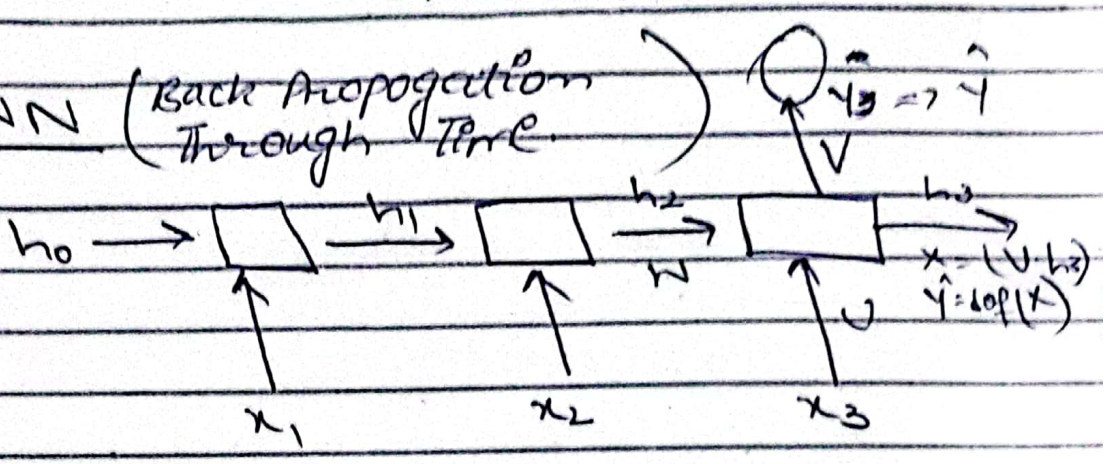
$$\rightarrow \frac{\partial x}{\partial W_{23}} = \frac{\partial(-H \cdot W_{23})}{\partial(W_{23})} = H$$

$$\rightarrow \frac{\partial L(Y,\hat{Y})}{\partial W_{23}} = (Y-\hat{Y}) \cdot H$$

$$\rightarrow \frac{\partial C}{\partial W_{12}} = \frac{\partial C}{\partial \hat{Y}} \cdot \frac{\partial \hat{Y}}{\partial x} \cdot \frac{\partial x}{\partial H} \cdot \frac{\partial H}{\partial A} \cdot \frac{\partial A}{\partial W_{12}}$$

$$= (\hat{Y} - Y) \cdot \underset{\text{(updated)}}{W_{23}} \cdot H(1-H) \cdot I$$

RNN (Back Propogation Through Time)



$$\frac{\partial C}{\partial V} = \frac{\partial C}{\partial y_3} \times \frac{\partial y_3}{\partial x} \times \frac{\partial x}{\partial V}$$

- Difference b/w NN & RNN is that RNN has the concept of shared weights

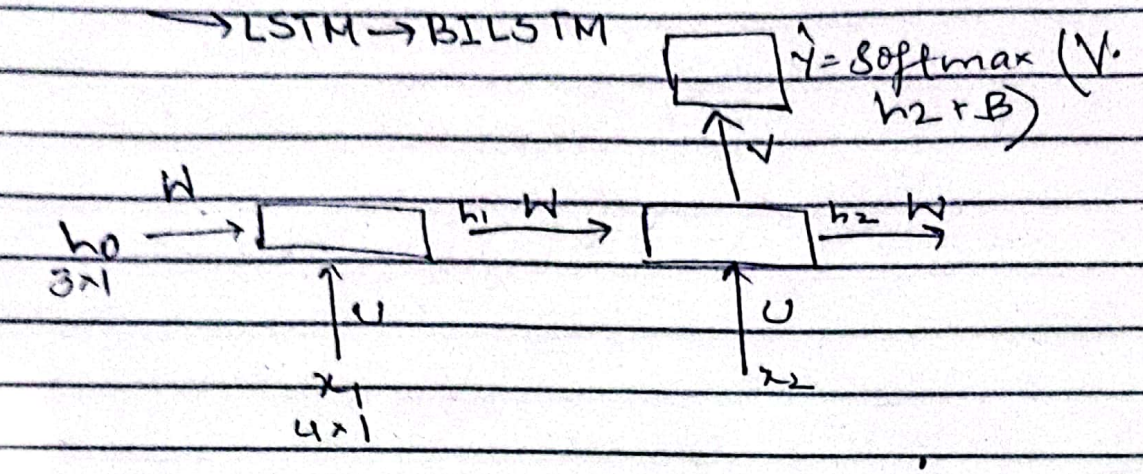
$$\frac{\partial C}{\partial W} = \frac{\partial C}{\partial y_3} \cdot \frac{\partial y^3}{\partial x} \cdot \frac{\partial x}{\partial h_2} \cdot \frac{\partial h_2}{\partial W} + \frac{\partial C}{\partial y_3} \cdot \frac{\partial y^3}{\partial x} \cdot \frac{\partial x}{\partial h_2} \cdot \frac{\partial h_2}{\partial h_1} \cdot \frac{\partial h_1}{\partial W}$$

- Misla of RNN → 1) vanishing gradient problem
2) Exploding Gradient

Lecture NO.15

RNN → GRU

→ LSTM → BILSTM



$W' = [W \cdot U]$ (3×7)

$I = [h_0, x_1]$ (7×1; 3×1, 4×1)

$h_1 = \tanh(W' \cdot I + B)$ (3×1)

$h_1 = \tanh([Wh_0 + Ux_1 + B])$ (3×3, 3×1, 4×1 → 3×4)

1) $\frac{\partial C}{\partial V} = \frac{\partial C}{\partial \hat{y}} * \frac{\partial \hat{y}}{\partial V}$

2) $\frac{\partial C}{\partial W} = \frac{\partial C}{\partial \hat{y}} * \frac{\partial \hat{y}}{\partial h_2} * \frac{\partial h_2}{\partial W} + \frac{\partial C}{\partial \hat{y}} * \frac{\partial \hat{y}}{\partial h_2} * \frac{\partial h_2}{\partial h_1} * \frac{\partial h_1}{\partial W} +$

3) Too much multiplication may cause the resultant value to be small causing following problems

- Vanishing Gradient Descent /
- Exploding Gradient

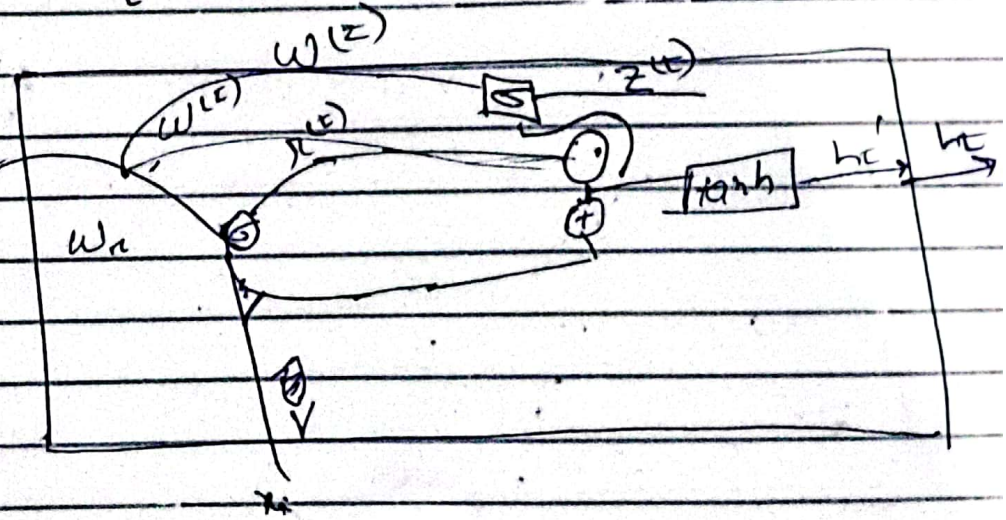
GRU:

Gates:
- update → to update prior context
- Reset → prior context to forget

similar to forget gate

how much

$$r^{(t)} = \sigma\left[W^{(r)} h_{t-1} + U^{(r)} x_t\right]$$



$$h'_t = \tanh\left[W^{(h)} h_{t-1} + U^{(h)} x_t\right] \odot r_t$$

Ex:

$h_{t-1} = [\,7\,|\,7\,|\,7\,]$

$r_t = [\,1\,|\,0\,|\,0\,]$

same size

result: $[\,7\,|\,0\,|\,0\,]$

↓

preserving 7 forgetting rest

•) weight matrix of reset, update & final output will be different.

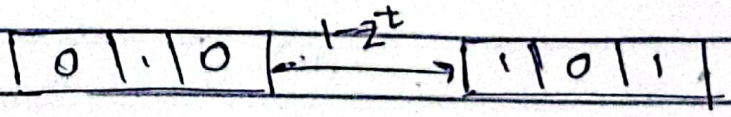
•) update

$$z^{(t)} = \sigma\left(W^{(z)} h_{t-1} + U^{(z)} x_t\right)$$

sigmoid

$$h_t = z^t + h_{t-1} + (1 - z^t) * h_t'$$

Ex:



$$\underset{3\times1}{r^{(t)}} = \sigma\left(\underset{3\times3}{W^{(r)}}\ \underset{3\times1}{h_{t-1}} + \underset{3\times4}{U^{(r)}}\ \underset{4\times1}{x_t}\right) \Rightarrow \boxed{0\ 0\ 0}$$

$$\underset{3\times1}{z^{(t)}} = \sigma\left(\underset{3\times3}{W^{(z)}}\ \underset{3\times1}{h_{t-1}} + \underset{3\times4}{U^{(z)}}\ \underset{4\times1}{x_t}\right)$$

GRU

$$h_t' = \tanh\left(\underset{3\times1}{U^{(t)}}\ \underset{4\times1}{x_t} + r \odot \underset{3\times3}{W^t}\ \underset{3\times1}{h_{t-1}}\right)$$

$$\underset{3\times1}{h_t} = \underset{3\times1}{z^{(t)}} \odot h_{t-1} + \left(1 - \underset{3\times1}{z^{(t)}}\right) \odot h_{t-1}$$

FOR

Equations GRU

Dimensions according to example.

$h_0 = \boxed{7}\boxed{7}\boxed{7}$

$x_1 = \boxed{3}\boxed{0}\boxed{4}\boxed{4}$

$r^{(t)} = \boxed{0}\boxed{0}\boxed{0}$

$z^{(t)} = \boxed{x_0}\boxed{x_0}\boxed{x_0}$ $\quad 1 - z^{(t)} = \boxed{1}\boxed{1}\boxed{1}$

GRU came after LSTM
↳ improved version of LSTM

LECTURE NO. 16

LSTM
↓
solved the problems of RNN

Gates:
1) Forget → removing all the prev values
2) Input
3) Output

NATURAL LANGUAGE PROCESSING