CBCS SCHEME

USN

BCS714A

Seventh Semester B.E./B.Tech. Degree Examination, Dec.2025/Jan.2026
Deep Learning

Time: 3 hrs. Max. Marks: 100

Note: 1. Answer any FIVE full questions, choosing ONE full question from each module.
2. M : Marks , L: Bloom's level , C: Course outcomes.

		Module – 1	M	L	C
Q.1	a.	Explain with a suitable diagram, how according to Hubel and Wiesels discovery a simple cell in the primary visual cortex of a cat fires at different rates depending upon the orientation of the line shown to the cat.	8	L2	CO1
	b.	Explain LeNet-5 hierarchical architecture and its use in automating the reading of postal service ZIP codes.	6	L2	CO1
	c.	Explain how the correctness of the sentence "The program impeccably translated the text" is determined using traditional learning and deep learning representations.	6	L2	CO1
		OR			
Q.2	a.	Explain one-hot representation of words for the vocabulary = {The, bat, sat, on, a, cat} and describe the process of finding whether a word in the vocabulary is an animal or not for a simple NLP task.	8	L2	CO1
	b.	Explain AlexNet's hierarchical architecture and its use in face detection.	6	L2	CO1
	c.	Describe word-vector representation of words. Given the vector for words as follows: King = [-0.9, 1.9, 2.2] Man = [-1.1, 2.4, 3.0] Woman = [-3.2, 2.5, 2.6] Queen = [-3.0, 2.1, 1.7] Explain the process for finding that, vector for queen will be closest to the relation King – Man + Woman = Queen.	6	L2	CO1
		Module – 2			
Q.3	a.	Illustrate with a suitable example, the need for regularization in deep learning.	8	L2	CO2
	b.	Write the mathematical formulation for the following parameter norm penalties: i) Limiting mode capacity ii) L^2 parameter regularization iii) L^1 regularization For each of the above, give the expression for the norm penalty term Ω (θ) and explain in each case, the effect of adding these penalties to the learning models. Also write the equations for the following giving explanation of the terms there-in. I. Closed form of Ridge regression II. Soft Thresholding rule for lasso slope solution.	8	L2	CO2

			M	L	C
	c.	Explain the need for data augmentation and how it is effective in object recognition.	4	L2	CO2
		OR			
Q.4	a.	Differentiate between Batch Gradient Descent and Minibatch Gradient Descent.	8	L2	CO2
	b.	Write the algorithm for stochastic gradient descent with momentum. Explain the core idea for introducing momentum and how it helps optimization.	8	L2	CO2
	c.	Describe the key mechanics of the Adam Optimization Algorithm.	4	L2	CO2
		Module – 3			
Q.5	a.	Perform full convolution (flip the Kernel) for the image $f(x, y) = \begin{bmatrix} ③ & 2 \\ 1 & 4 \end{bmatrix}$ and the Kernel $h(x, y) = \begin{bmatrix} ⑦ & 6 \\ 5 & 8 \end{bmatrix}$ where the circled values 3 and 7 in the image and kernel indicate their (0, 0) positions.	9	L3	CO3
	b.	The convolution result for an image $f(x, y) = \begin{bmatrix} ① & 2 \\ 3 & 4 \end{bmatrix}$ and Kernel $h(x, y) = \begin{bmatrix} ⑤ & 6 \\ 7 & 8 \end{bmatrix}$ is given as $\text{Gshared} = \begin{bmatrix} ⑤ & 16 & 12 \\ 22 & 60 & 40 \\ 21 & 52 & 32 \end{bmatrix}$ Perform unshared convolution using the above information.	8	L3	CO3
	c.	In the CNN architecture, the output of the convolution operation is a feature map which is further given as input to an activation function followed by pooling. Given the result of convolution as $g(x, y) = \begin{bmatrix} 2 & 3 & -2 & -3 \\ 3 & 10 & -5 & 2 \\ 1 & 5 & -5 & 1 \end{bmatrix}$ Perform the activation function operation using Rectified Linear Unit (ReLU) followed by Max Pooling (2 × 2, stride 1)	3	L3	CO3
		OR			
Q.6	a.	Compute the output of full cross-correlation (without kernel flipping) for an input image $f(x, y) = \begin{bmatrix} ④ & 2 \\ 3 & 1 \end{bmatrix}$ and a Kernel $h(x, y) = \begin{bmatrix} ⑧ & 6 \\ 7 & 5 \end{bmatrix}$ where the circled values 4 and 8 indicate their (0, 0) positions.	9	L3	CO3

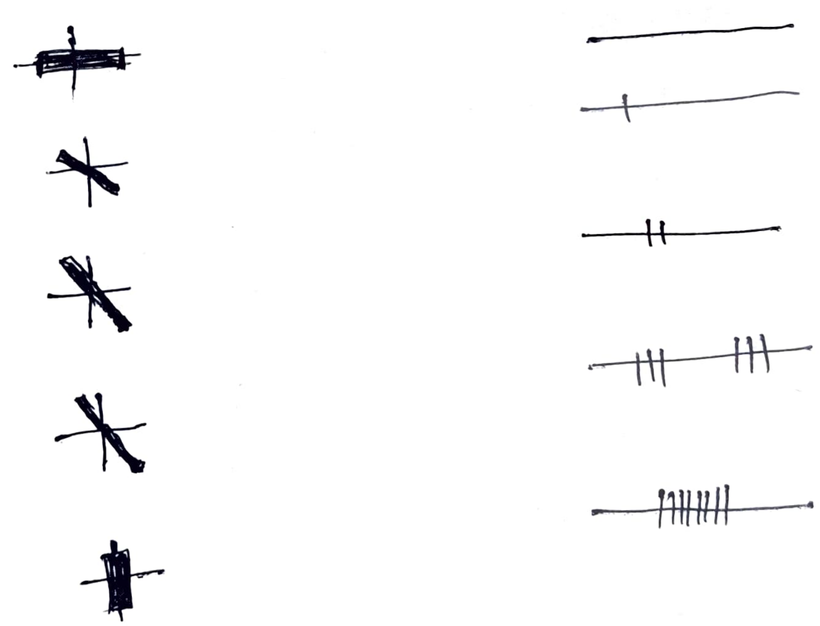
	b.	Compute the result of full tiled convolution (flip the kernels) for an input image $f(x, y) = \begin{array}{\|c\|c\|}\hline 4 & 3 \\ \hline 2 & 1 \\ \hline\end{array}$ and Kernel $h(x, y) = \begin{array}{\|c\|c\|}\hline 7 & 5 \\ \hline 6 & 8 \\ \hline\end{array}$ where the circled values 4 and 7 indicate their (0, 0) positions. Consider Tile_A = h(x, y) (flipped) and Tile_B = Tile_A + 10	8	L3	CO3
	c.	The result of convolution + ReLU in the CNN workflow is given as shown below: $G(\text{CONV} + \text{ReLU}) = \begin{array}{\|c\|c\|c\|c\|}\hline 4 & 17 & 21 & 18 \\ \hline 9 & 32 & 43 & 30 \\ \hline 2 & 10 & 16 & 8 \\ \hline\end{array}$ Perform the operations of Max Pooling (2 × 2, stride 1) followed by reshaping the output of Max Pooling into a 1D vector suitable to be given to the next stage of the CNN workflow.	3	L3	CO3
		Module – 4			
Q.7	a.	Draw a diagram to show how unfolding a computational graph helps in understanding a Recurrent Neural Network (RNN). For the sequence of three words "I play tennis", show the step by step computation of the basic RNN processing of sequential data using forward propagation. Sequence of three words : I play tennis Where I = 1, play = 2, tennis = 3 Use, i) Input – hidden weight $u_x = 0.5$ ii) Hidden – hidden weight $w_h = 0.8$ iii) Bias b = 0.1 iv) Initial state $h_0 = 0$	10	L3	CO4
	b.	Compare Bidirectional RNN with standard RNN. Compute the Backward pass (←) for the input sequence : "I am thrilled" where I = 1, am = 1, thrilled = 2 Use, i) Initial backward state: hB3 = 0 ii) Backward Input-Hidden weight : $u_B = 0.6$ iii) Backward Hidden-Hidden weight : $W_B = 0.8$ iv) Backward Bias = 0.2 = β	10	L3	CO4
		OR			
Q.8	a.	Describe the roles of encoder and decoder in RNN encoder-decoder architectures. Assume that the final encoder state h_3 for the input sentence : I play cricket = 0.937. If the classification weights per class for the classes [learning, sports, others] is given by $w_y = [w_1, w_2, w_3] = [0.5, 1.5, -0.5]$ and classification biases per class are given as by = $[b_1, b_2, b_3] = [0.0, -0.9, 0.0]$, then decode the input sequence to the class "sports".	10	L3	CO4
	b.	Compare Deep Recurrent Networks and Recursive Neural Networks. Represent the below provided input sentence: "I study Deep Learning" as a tree using Recursive Network concept.	10	L3	CO4

BCS714A

		Module – 5			
Q.9	a.	Explain the following common natural language preprocessing steps: i) Stop word removal ii) Stemming	10	L2	CO5
	b.	Explain how high dimensional word vectors are plotted to map them to two or three dimensions.	10	L2	CO5
		OR			
Q.10	a.	Explain the following performance metrics of deep learning NLP models: i) The area under the ROC curve ii) The confusion matrix.	10	L2	CO5
	b.	Explain with a toy example, how the ROC Auc metric is calculated.	10	L2	CO5

* * * * *

BCS714A - DEEP LEARNING

Q.NO	Solution	Marks
1a.	Hubel & Wiesel studied neurons in cat's visual cortex. → found simple cells respond selectively to lines or edges of specific orientation. → A simple cell in the primary visual cortex of a cat fires at different rates.	02
	Orientation of the line — Firing (Electrical Act)	03
		03
1b.	LeNet-5 : 7 layer CNN This uses local receptive fields, shared weights and subsampling. Hierarchical Feature learning :- Input, First convolution, First subsampling second convolution, second subsampling fully connected layers, output	04

Convolution → subsampling (pooling) → convolution → subsampling (pooling) → Full connection → Full connection → output 02

1c. sentence: Goal is to get a computer to understand that is a syntactically correct.

Traditional method: The "Grammar Rulebook"

Rule 1: sentence → Noun Phrase + Verb Phrase 03

Rule 2: Noun phrase → Article + Noun

Rule 3: Verb phrase → Verb + Noun phrase

Rule 4: Article → "the", "a", "an"

Rule 5: Noun → "program"

Rule 6: Verb → "translated", "is"..

Computer process:-

→ The (article) + program (noun) = Noun phrase

→ translated (verb) + the text (Noun Phrase) = Verb phrase.

Deep Learning process:-

It reads the sentences and asses the probability of this word based on training

→ It has seen the pattern [Noun] + [Adverb] + [verb] 03

2a. <u>One-hot representation of words</u> :-

It is represented by a binary vector vector length is equal to vocabulary size. 02

vocabulary = { The, bat, sat, on, a, cat }

→ word : The → [1, 0, 0, 0, 0, 0]

→ word : bat → [0, 1, 0, 0, 0, 0]

→ word : sat → [0, 0, 1, 0, 0, 0]

→ word : on → [0, 0, 0, 1, 0, 0] 03

→ word : a → [0, 0, 0, 0, 1, 0]

→ word : cat → [0, 0, 0, 0, 0, 1]

Perceptron setup as follows,

output = 1 for animal 01

output = 0 for Not animal

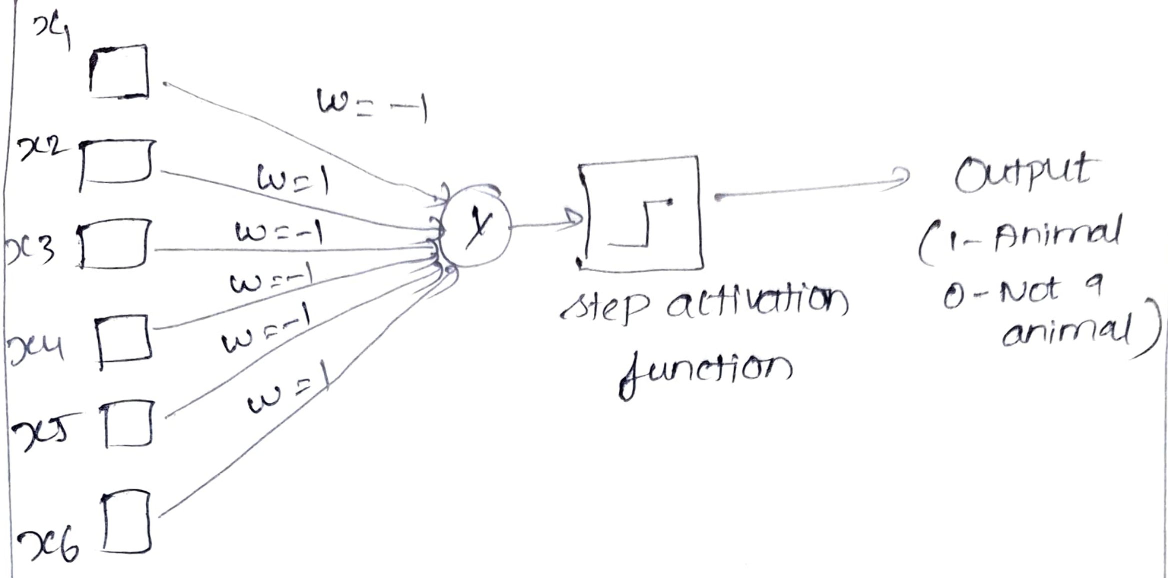
weight vector $w = [-1, 1, -1, -1, -1, 1]$

activation : Step activation function

For any word with vector x

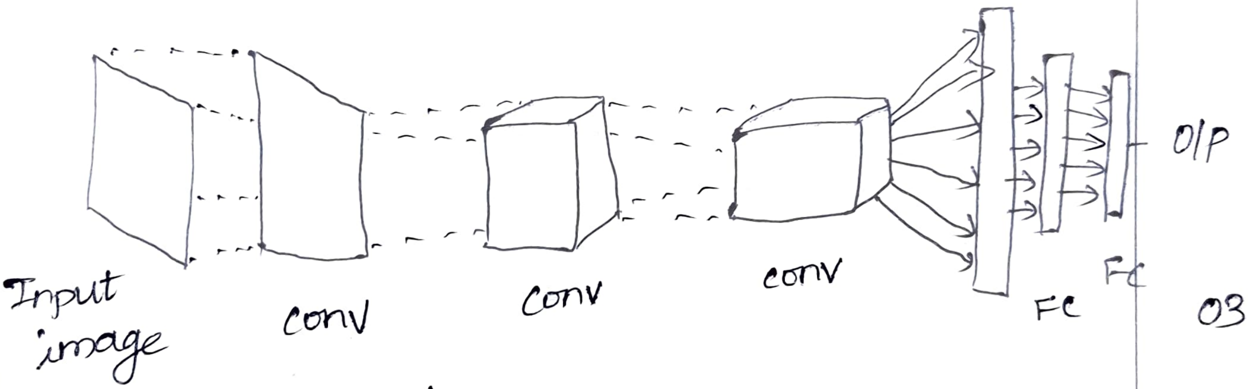
$y_{in} = w \cdot x$ 01

$$y_{out} = \begin{cases} 1, & \text{if } y_{in} \geq 0 \\ 0, & \text{otherwise} \end{cases}$$



01

2b. AlexNet



03

For detecting Face:-

Input layer, convolution & pooling layers, first, second, third convolution, Fully convolution layers, output (software/softmax layer) 03

2c A word vector is a way of representing a word as a list of numbers in a high dimensional space. 01

Man = [-1.1, 2.4, 3.0], woman = [-3.2, 2.5, 2.6]

king = [-0.9, 1.9, 2.2] 03

King - Man + women = Queen

king - man + woman = [(-0.9 + 1.1 - 3.2), (1.9 - 2.4 + 2.5), (2.2 - 3.0 + 2.6)] 03

king - man + woman = [-3.0, 2.0, 1.8]

$$\text{cos\_sim}[\text{Result}, \text{women}] = \frac{(-3)(-3.2) + (2)(2.5) + (1.8)(2.6)}{\sqrt{(-3)^2 + 2^2 + (1.8)^2}\sqrt{(-3.2)^2 + (2.5)^2 + 2.6^2}}$$

cos_sim [Result, woman] = 0.99221 02

cos_sim [Result, queen] = 0.99939

cos_sim [Result, king] = 0.87399

cos_sim [Result, man] = 0.83828

3a. Regularization is defined as any modication we make to a learning algorithm to reduce its generalization error 01

consider the example having the measure of temperature y over time x.

x	1	2	3	4	5
y	11	12	13	14	15

03

$f(x) = ax + b$

$$a = \frac{n\sum xy - (\sum x)(\sum y)}{n\sum x^2 - (\sum x)^2} \quad \& \quad b = \frac{\sum y - a\sum x}{n}$$

$n = 5$, $\sum xy = 205$, $a = 1$, $b = 10$ 03

$g(c) = 16 + 120k$

Regularization would push k towards 0, bringing g(x) back to the sensible lines x+10. 01

3b. $J_{regularized}(\theta) = J(\theta) + \alpha\,\Omega(\theta)$

where $J(\theta)$ = original loss function

$\Omega(\theta)$ = norm penality term 01

α = regularization

1. Limiting model capacity
2. L^2 parameter Regularization 01

$$\theta = J(\theta) + \alpha\left[\|w\|_2^2 + \|b\|_2^2\right)$$

3. L^1 Regularization: 01

$$J(\text{regularized}) = J(\theta) + \alpha\,\Omega(\theta)$$

Effects of adding penalities 03

Closed form of ridge regression 01

$$\theta = (X^T X + \alpha I)^{-1} X^T y$$

softutthresholding rule for Lasso slope solu:

$$a = \frac{1}{S_{xx}} \cdot \text{sign}(S_{xy}) \cdot \max(|S_{xy}| - \alpha, 0)$$ 01

3c. Need for data Augmentation :-

→ To train on more data.

→ Generally the amount of data is limited & this problem can be overcome with data augmentation by creating synthesized data 02

Effectiveness for object recognition 02

4a.

Batch Gradient descent	MiniBatch stochastic gradient descent
→ uses the entire training dataset	→ uses small, random subset of training dataset
→ High computational cost & memory usage per update	→ Low computational cost per update.
→ converges smoothly to local minimum	→ convergence is noisy & oscillatory due to variance.

08

4b. Stochastic Gradient Descent with momentum:
Learning rate E, momentum parameter α
Initial parameter ϕ, initial velocity V
while stopping criteria not met do
sample a minibatch $\{x^{(1)} \dots x^{(n)}\}$
Compute gradient estimate
Compute velocity update
Apply update
end while. 06

Past gradient helps to optimizes,
i) Build up speed in directions with consistent, persistent gradients 02
ii) Resist erratic changes in direction caused by noisy gradient

4c. Adam optimization Algorithm:-

1. $m_t = \beta_1 \cdot m_{t-1} + (1-\beta_1) \cdot g_t$

2. $U_t = \beta_2 \cdot V_{t-2} + (1-\beta_2) \cdot g_t^2$ 04

3. $\hat{m}_t = \frac{m_t}{1-\beta_1^t}$ & $\hat{U}_t = \frac{V_t}{1-\beta_2^t}$

4. $\theta_t = \theta_{t-1} - n \cdot \frac{\hat{m}_t}{\sqrt{\hat{V}_t + \epsilon}}$

5a. The convolution operation:

$$g(x,y) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f(m,n)\, h(x-m, y-n)$$

The first step,

$f(x,y) = f(m,n)$

$h(x,y) = h(m,n)$

$f(m,n) = \begin{array}{|c|c|} \hline 3 & 2 \\ \hline 1 & 4 \\ \hline \end{array}$, $h(m,n) = \begin{array}{|c|c|} \hline 7 & 6 \\ \hline 5 & 8 \\ \hline \end{array}$

the flipped kernel $h(-m,-n) = \begin{array}{|c|c|} \hline 8 & 5 \\ \hline 6 & 7 \\ \hline \end{array}$

$f\ x=0, y=0 \Rightarrow g(0,0) = 21$ 01

$x=0, y=1 \Rightarrow g(0,1) = 32$ 01

$x=0, y=2 \Rightarrow g(0,2) = 12$ 01

$x=1, y=0 \Rightarrow g(1,0) = 22$ 01

01

$x=1, y=1$, $g(1,1) = 68$ 01

$x=1, y=2 \Rightarrow g(1,2) = 40$

$x=2, y=0 \Rightarrow g(2,0) = 5$ $g(x,y) = \begin{array}{|c|c|c|} \hline 21 & 32 & 12 \\ \hline 22 & 68 & 40 \\ \hline 5 & 28 & 32 \\ \hline \end{array}$ 01

$x=2, y=1 \Rightarrow g(2,1) = 28$ 01

5b. Unshared convolution

$G_{unshared}(i,j) = G_{shared}(i,j) + C_{i,j} \times \begin{bmatrix} \text{sum of} \\ \text{image} \\ \text{pixels} \end{bmatrix}$

calculate the unshared convolution,

1. $G_{unshared}(i,j) =$

5	16	12
22	60	40
21	52	32

2. The position depend constants $(C_{i,j} = i+j)$ 02

$C = [C_{i,j}] = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 3 \\ 2 & 3 & 4 \end{bmatrix}$ 04

3. S = sum of image pixels that overlapped with kernel at (i,j) 01

4. compute $C \times S$

5. Add all $C \times S$ to $G_{shared}(i,j)$ rowwise

$G_{unshared}(i,j) =$

5	19	16
26	80	58
27	73	48

5C Activation Func. operation by ReLu

$ReLu(g(x,y)) = max(0, \text{value in each cell}(i,j))$ 02

Final result,

$g(x,y)$

2	3	-2	-3
3	10	5	2
1	5	5	1

$\xrightarrow{\text{Relu}}$ $g(x,y)$

2	3	0	0
3	10	5	2
1	5	5	1

$\xrightarrow{\text{Max Pooling}}$

10	10	5
10	10	5

2x2 stride 1

01

6a. Cross - correlation operation

$$g(x,y) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f(m,n)\, h(x+m, y+n)$$

$g(0,0) = 70$ $\quad x = -, y = 1 = 18$

$g(0,1) = 39$ $\quad g(-1,-1) = 8$

$g(0,-1) = 23$

$g(1,0) = 38$

$g(1,1) = 20$

$g(1,-1) = 14$

$g(-1,0) = 30$

$g(x,y) =$

8	30	18
23	70	39
14	38	20

6b The Titled convolution opertion:

$h(x,y)$(flipped) $= h(m,n)$ flipped) $= h(-m,-n) =$

8	6
5	7

$$T(x,y) = \begin{cases} \text{Tile\_A}, & \text{if } (x+y) \bmod 2 = \text{even} \\ \text{Tile-B}, & \text{if } (x+y) \bmod 2 = \text{odd} \end{cases}$$ 01

$\therefore$ Tile-A $= h(-m,-n) =$

8	6
5	7

01

Tile_B = Tile_A + 10 =

8 + 10	6 + 10
5 + 7	7 + 10

=

18	16
15	17

$g(0,0) = 28$ $\quad g(1,2) = 69$ 06

$g(0,1) = 111$ $\quad g(2,0) = 12$

$g(0,2) = 15$ $\quad g(2,1) = 52$

$g(1,0) = 98$ $\quad g(3,2) = 8$

$g(1,1) = 67$

$(x,y) =$

28	111	15
98	67	69
12	52	8

6c. max pooling on G (conv + ReLu) 02

max pooling result

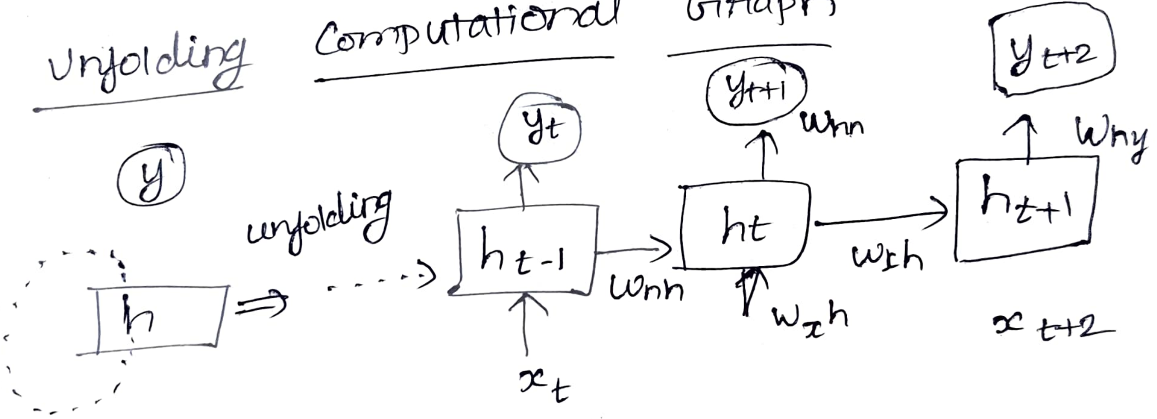
32	43	43
32	43	43

Reshaping the max pooling

32	43	43
32	43	43

flatten the 2×3 pooled map to 1D vector → [32, 43, 43, 32, 43, 43] 01

7a. Unfolding Computational Graph 04



$h_3 = 0.9817$ → captures the full meaning of the sentence. 06

7b 05

standard RNN	Bidirectional RNN
Process in forward direction.	Process the sequence in both the direction
considers only the past info.	considers both pasts & future context.
Employs single RNN layers	Employs two RNN's
O/P depends 1→t (1 to t)	1→ end (full sequence

$hB2 = 0.885$

$hB1 = 0.906$

$hB0 = 0.910$

8a. Role of Encoder- Decoder in RNN Archi :-

Encoder : Encoder RNN reads the entire input sequence

Decoder : The Decoder RNN takes the context vector as input & generats the output sequence. 04

Ex: I/P : I play cricket

O/P : sports

Classification :

$$P_i = \frac{e^{(w_i . h + b_i)}}{\sum_j e^{(w_j . h + b_j)}}$$ 06

Softmax = [0.412, 0.427, 0.161]

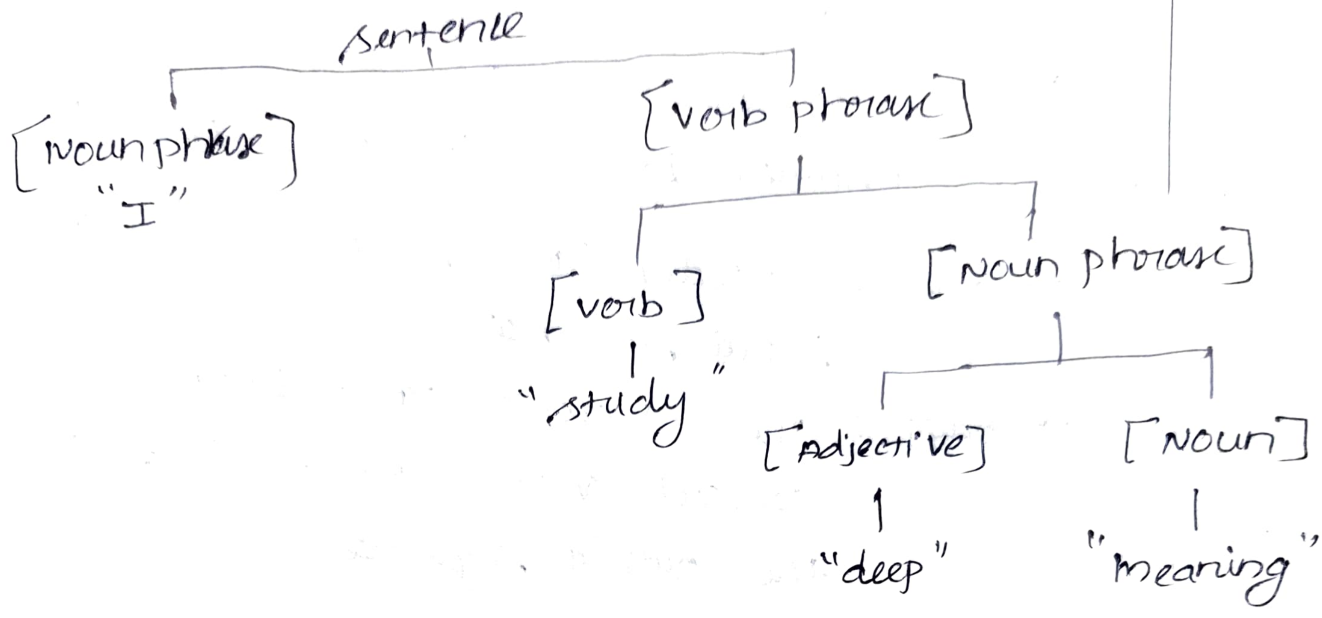
→ class 2 ⇒ sports

8b.

Deep Recurrent Networks	Recurrent Neural N/W
→ Multiple RNN Layers are stacked vertically	→ Nodes are connected tree like structure
→ sequential data.	→ structured data.
→ Inf. flows thro' time	→ Inf. flows bottom-up
→ captures the temporal dependencies	→ captures compositional or hierarchical meaning

05

Recursive Network.



9a. stop word removal 05

stemming 05

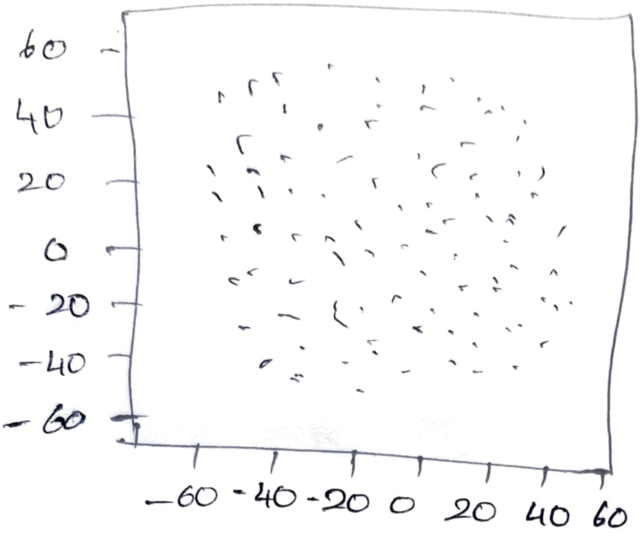
It is the truncation of words down to their stem.

Ex: house and housing

both have the stem hous

9b. Human brains are not well suited to visualizing anything greater than three dimensions.

We use dimensionality reduction techniques to map the words from high dimensions to two or three dimensions



Static 2D word-vector scatter plot — 10

10a. Area Under the ROC curve :-

ROC curve plotted between 2 classes, True positive rate (TPR) and Negative False positive rate (FPR) for different threshold values. — 05

ii) Confusion Matrix :

It is a performance measurement tool for classification models.

It summarizes predictions vs actuals. — 05

True positive (TP)

True Negative (TN)

False positive (FP)

False Negative (FN)

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

$$\text{Precision} = \frac{TP}{(TP+FP)}$$

$$\text{Recall} = TP/(TP+FN)$$

$$\text{F1 score} = \frac{2 \times \text{Prec} \times \text{Recall}}{\text{Prec} + \text{Recall}}$$

10b AUC = area under the ROC curve plotted with TPR (y axis) vs FPR (x axis)

$$TPR\ (Recall) = \frac{TP}{TP+FN}$$ 02

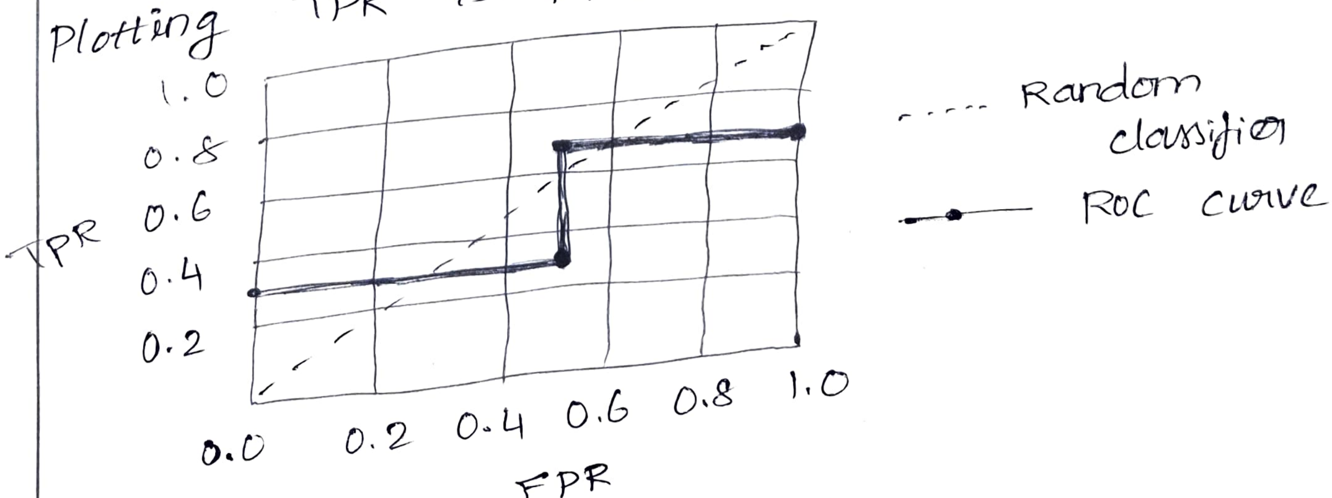
$$FPR = \frac{FP}{FP+TN}$$

Compute TPR & FPR at each threshold

Threshold ≥	Predicted positives	TP	FP	FN	TN	TPR	FPR
0.9	A	1	0	2	2	0.33	0.00
0.8	A, B	1	1	2	1	0.33	0.5
0.7	A, B, C	2	1	1	1	0.67	0.5
0.4	A, B, C, D	2	2	1	0	0.67	1.0
0.3	A, B, C, D, E	3	2	0	0	1.0	1.00

03

Plotting TPR VS FPR 0.5



02

$$AUC = \sum \frac{(TPR_i + TPR_{i-1})}{2} \times FPR_i - FPR_{i-1}$$

AUC = 0.503