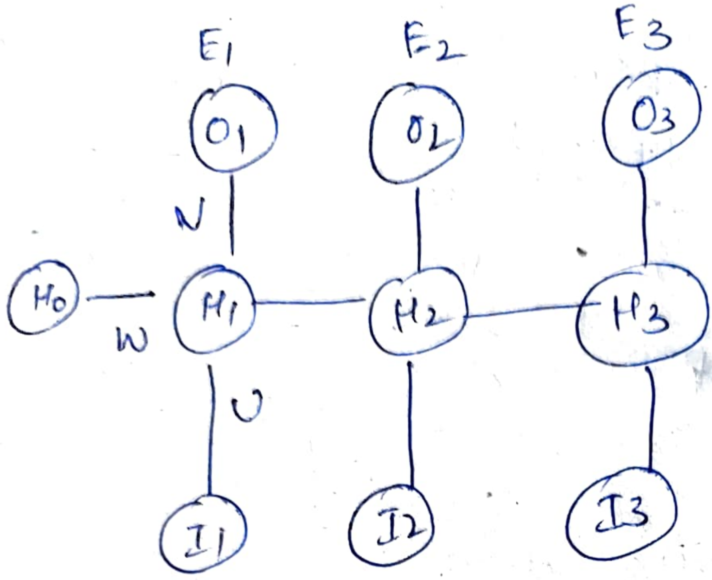
Assignment - 04

K. Surya Prakash

EE18BTECH11026

Q1)

Soln



** Assuming H0 is independent of W, U, V

* Assuming tanh activation : in hidden state
Softmax : in output state

& Cross entropy as loss fn.

Defn.

$$H_t = \tanh(U I_t + W H_{t-1})$$

$$O_t = \text{Softmax}(V H_t)$$

* We ~~can~~ know that:

$$\frac{\partial H_t}{\partial H_{t-1}} = (1 - H_t^2) \cdot W$$

$$E_t = CE(O_t, \overset{\text{actual}}{y_t})$$

$$\Rightarrow \frac{\partial E_t}{\partial H_t} = \underbrace{(O_t - y_t)}_{\text{combining both CE + Softmax derivatives}} V$$

$$\frac{\partial H_t}{\partial W} = (1-H_t^2)\cdot\left(H_{t-1} + W\cdot\frac{\partial H_{t-1}}{\partial W}\right)$$

$$\frac{\partial H_t}{\partial U} = (1-H_t^2)\,(I_t)$$

(a) Derivatives of total losses:

$$E = E_1 + E_2 + E_3$$

(i) $\frac{\partial E}{\partial W} = \frac{\partial E_1}{\partial W} + \frac{\partial E_2}{\partial W} + \frac{\partial E_3}{\partial W}$

(ii) $\frac{\partial E}{\partial U} = \frac{\partial E_1}{\partial U} + \frac{\partial E_2}{\partial U} + \frac{\partial E_3}{\partial U}$

(iii) $\frac{\partial E}{\partial V} = \frac{\partial E_1}{\partial V} + \frac{\partial E_2}{\partial V} + \frac{\partial E_3}{\partial V}$

Each component calculated seperatives

Derivative of E_1:

$$\frac{\partial E_1}{\partial W} = \frac{\partial E_1}{\partial H_1}\cdot\frac{\partial H_1}{\partial W}$$

$$= (O_1 - y_1) * V * (1-H_1^2)\cdot H_0$$

$$\frac{\partial E_1}{\partial U} = \frac{\partial E_1}{\partial H_1}\cdot\frac{\partial H_1}{\partial U}$$

$$= (O_1 - y_1) * V * (1-H_1^2)\, I_1$$

$$\frac{\partial E_1}{\partial V} = \frac{\partial E_1}{\partial (VH_1)}\,\frac{\partial (VH_1)}{\partial V} = (O_1 - y_1) \otimes H_1$$

* b) <u>Derivatives of E_2</u>:

$\Rightarrow$ (i) $\frac{\partial E_2}{\partial W} = \frac{\partial E_2}{\partial H_2} \cdot \left(\frac{\partial H_2}{\partial W} + \frac{\partial H_2}{\partial H_1} \cdot \frac{\partial H_1}{\partial W} \right)$

$= (O_2 - y_2) * V * \left[(1 - H_2^2) \right] \left(H_1 + W(1 - H_1^2) H_0 \right)$

$\Rightarrow$ (ii) $\frac{\partial E_2}{\partial U} = \frac{\partial E_2}{\partial H_2} \left(\frac{\partial H_2}{\partial U} + \frac{\partial H_2}{\partial H_1} \cdot \frac{\partial H_1}{\partial U} \right)$

$= (O_2 - y_2) * V * \left[(1 - H_2^2) I_2 + (1 - H_2^2) W \cdot (1 - H_1^2) I_1 \right]$

$= (O_2 - y_2) * V * (1 - H_2^2) \left(I_2 + W * (1 - H_1^2) I_1 \right)$

(iii) $\frac{\partial E_2}{\partial V} = \frac{\partial E_2}{\partial (VH_2)} \cdot \frac{\partial (VH_2)}{\partial V}$

$= (O_2 - y_2) * H_2$

i) Derivatives of E_3:

$$\frac{\partial E_3}{\partial W} = \frac{\partial E_3}{\partial H_3} \cdot \left(\frac{\partial H_3}{\partial W} + \frac{\partial H_3}{\partial H_2} \cdot \frac{\partial H_2}{\partial W} + \frac{\partial H_3}{\partial H_2} \cdot \frac{\partial H_2}{\partial H_1} \cdot \frac{\partial H_1}{\partial W} \right)$$

$$= (O_3 - y_3) * V \left((1-H_3^2) H_2 + (1-H_3^2) W (1-H_2^2) H_1 + (1-H_3^2) W (1-H_2^2) W \cdot (1-H_1^2) H_0 \right)$$

$$= (O_3 - y_3) * V * (1-H_3^2) \cdot \left(H_2 + W(1-H_2^2) \{ H_1 + W(1-H_1^2) H_0 \} \right)$$

ii) $$\frac{\partial E_3}{\partial U} = \frac{\partial E_3}{\partial H_3} \left(\frac{\partial H_3}{\partial U} + \frac{\partial H_3}{\partial H_2} \cdot \frac{\partial H_2}{\partial U} + \frac{\partial H_3}{\partial H_2} \cdot \frac{\partial H_2}{\partial H_1} \cdot \frac{\partial H_1}{\partial U} \right)$$

$$= (O_3 - y_3) * V \cdot \left((1-H_3^2) I_3 + (1-H_3^2) W \cdot (1-H_2^2) I_2 + (1-H_3^2) W (1-H_2^2) W (1-H_1^2) I_1 \right)$$

$$= (O_3 - y_3) * V * (1-H_3^2) \left(I_3 + W(1-H_2^2) \{ I_2 + W(1-H_1^2) I_1 \} \right)$$

//

* (iiii)

$$\frac{\partial E_3}{\partial V} = \frac{\partial E_3}{\partial (VH_3)} \cdot \frac{\partial (VH_3)}{\partial V}$$

$$= (O_3 - y_3) \otimes H_3$$ //

Q2)

From Question ① ; we can generalise that

$$\frac{\partial E_3}{\partial W} = \sum_{k=0}^{3} \frac{\partial E_3}{\partial O_3} \cdot \frac{\partial O_3}{\partial H_3} \underbrace{\left(\prod_{j=k+1}^{3} \frac{\partial h_j}{\partial h_{j-1}} \right)} \cdot \frac{\partial h_k}{\partial W}$$

* We can see that the error gradient wrt. 'W' has to go through chain-rule for different time steps. ; which involves multiplication of gradients of activation functions

As $\frac{\partial h_j}{\partial h_{j-1}} \propto \frac{d(\tanh(\cdot))}{d h_{j-1}}$

We know that grad. of tanh (:) is bounded (or sigmound is bouder by 1 (≤ 1)

This cascaded product will result in a very small number: $\Rightarrow$ resulting in a small gradient of E_3 wrt W.

→ This is called as vanishing gradient problem

→ Possible Solutions:

→ Use ReLU activation (whose derivative can be 1)

→ Reduce time sequences.

→ Using architectures like LSTM / GRU's

Q.b) (i) The data consists of repetitive words, & hence model relies on long term dependencies. { in terms of understanding how many times can a word repeat, or only unique words }

→ To learn such patterns, while back-propagation; it needs to go till the last time step.

→ As 10 time steps are present, (being a large sequence); this might suffer from vanishing gradient

Solution-

↳ Using LSTM:

⇒ LSTM is a class of RNN specially designed to tackle such tasks.

→ It has a cell state which stories to remember long-term dependencies

→ In case a word is repeated we can pass on the cell state without any change. → This holds on long-term dependency.

↳ We can also use ReLU activation instead of tanh / sigmoid a

Q3) Precision : $\frac{TP}{TP+FP}$: $\frac{(\text{True pred} \le \text{rank})}{\text{rank}}$

Recall : $\frac{TP}{TP+FN}$: $\frac{(\text{True pred} \le \text{rank})}{\text{Total true} = 5}$

Here, Total instances of goose class are 5

$\Rightarrow TP+FN = 5$ ||

Avg. Precision (AP) $= \frac{1}{11} \sum_{r \in \{0, 0.1, \dots 1\}} P_{inter}(r)$

where $P_{inter}(r) = \max_{\tilde{r} \geq r} p(\tilde{r})$

Rank	Pred	Precision	Recall
1	T	1	$1/5 = 0.2$
2	T	1	$2/5 = 0.4$
3	F	$2/3 = 0.67$	0.4
4	F	$2/4 = 0.5$	0.4
5	F	~~3~~ $2/5 = 0.4$	0.4
6	T	$3/6 = 0.5$	$3/5 = 0.6$
7	T	$4/7 = 0.57$	$4/5 = 0.8$
8	F	$4/8 = 0.5$	0.8
9	F	$4/9 = 0.44$	0.8
10	T	$5/10 = 0.5$	$5/5 = 1$

Precision: $\frac{TP}{TP+FP}$; Recall $= \frac{TP}{TP+FN}$

rank ← { TP+FP

Total no. of true rows : 5

TP: (True preds before ≤ Rank)

Interpolation of Precision & Recall

Recall	Pinter
0	1
0.1	1
0.2	1
0.3	1
0.4	1
0.5	0.57
0.6	0.57
0.7	0.57
0.8	0.57
0.9	0.5
1	0.5

$\Rightarrow AP = \frac{1}{11}(\Sigma \text{ Pinter})$

$= \frac{1}{11}(5 + 4(0.57) + 2(0.5))$

$= 0.752$

4Q)

Focal loss (FP) = $-(1-p)^{\gamma}(\log p)$

γ: hyperparameter.

p: prob. of labelled class (ground truth class)

CE (cross entropy):

$$CE(p) = -\log(p)$$

---

a) If $\gamma = 0$

$\Rightarrow$ $FL(p) = CE(p)$

Both losses will be same

b) i) For correctly classified point

$\Rightarrow$ p close to 1

$\Rightarrow$ $(1-p)$ close to 0

$\log p$ close to '0'

$$FL(p) = (1-p)^{\gamma}\log(p) \longrightarrow 0$$

This point contributes very small to the total loss.

(ii) For incorrect classification:

p close to 0 than 1

$(1-p)$ close to 1

$-\log p \rightarrow$ very high

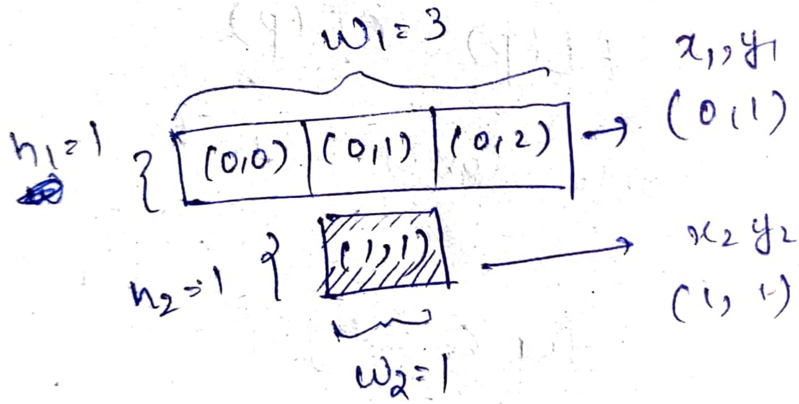
$FL(p) = -(1-p)^{\gamma} \log(p) \rightarrow$ high

∴ This point contributes more for total-loss.

Q5) Example:

Soln

a)



$$L2(\text{loss}) = \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2 + (h_2-h_1)^2 + (w_2-w_1)^2}$$

$$\Rightarrow L2(\text{loss}) := \sqrt{(1)^2 + (1-1)^2 + (1-1)^2 + (3-1)^2}$$

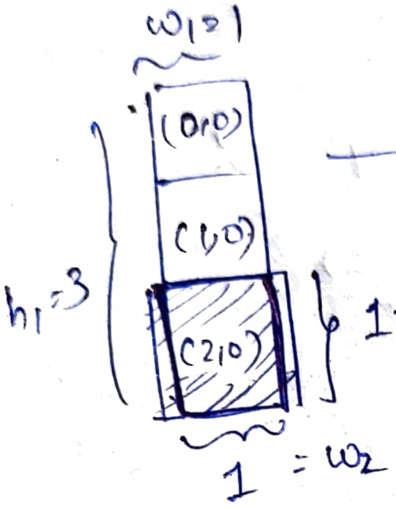
$$= \sqrt{1+4}$$

$$= \sqrt{5}$$

But here $IoU = 0/_{3+1} = \underline{0}$

$$\boxed{\therefore IoU = 0}$$

b)



(x_1, y_1)
$(1, 0)$

(x_2, y_2)
$(2, 0)$

$$L_2 (loss) = \sqrt{(2-1)^2 + (0-0)^2 + (3-1)^2 + (1-1)^2}$$

$$= \sqrt{(1)^2 + (2)^2} = \sqrt{5}$$

But here

$$IoU = 1/_{3+1-1} = 1/3$$

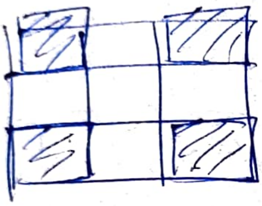
$\Rightarrow \therefore$ In both cases: L2 lou is same while IoU is different.

Intuitive Explanation:

↳ ~~IoU and loss function com~~

⇒ If Both bbox are squares

then same L_2 loss ⇒ same IoU loss



} same IoU for all 4 cases & same L_2 loss

→ Here, L2 loss is taken on bbox parameters $\{x_c, y_c, h, w\}$

x_c, y_c → center coords

h → height

w → width

⇒ when we fix the centres of both bboxes and try to vary $(h_2-h_1)^2$ & $(w_2-w_1)^2$

a) For Square bbox,

$(h_2-h_1)^2 = (w_2-w_1)^2$

∴ ~~In order to have~~

In such scenario ⇒ the parameters are

symmetric ⇒ Results in same IoU

b) when bbox are different shapes:

→ Here we can fix centers and tweak (h, w) of other bbox so that we can get the same IoU with different geometric shape ⇒ This leads to different IoU; which is shown in the example

L_2 (loss): Finds the euclidean dist ~~bt the~~ of parameters (vectors)

IoU: Find Overlap of areas.

Can be similar if both are squares. (symmetric)

Q6)

Input : (3x3) ones matrix [X,X]

kernel: (7x7) ~~E~~, [K,K]

↳ Stride : 01 (S) padding : 0 (P)

⇒ Output size : of transposed conv is

$$O = (X \cdot S) - S + K - 2P$$

$$O = ((3 \cdot 1) - 1 + 7 - 0)$$

$$= 3 + 7 - 1$$

$$= 9$$

∴ $\boxed{\text{Output size : } (9 \times 9)}$

6.b)

2D. 'Transposed Conv' in ~~2D~~ matrix multiplication form.

Assuming kernel: (2×2)
Input (2×2)

let $K = \begin{bmatrix} k_1 & k_2 \\ k_3 & k_4 \end{bmatrix}$;

$I = \begin{bmatrix} x_1 & x_2 \\ x_3 & x_4 \end{bmatrix}$; $\underbrace{I\text{-}f = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}}_{\text{flattened input}}$

$K\text{-}f$: Linear mapped version of kernel

$$K\text{-}f = \begin{bmatrix} k_1 & k_2 & 0 & k_3 & k_4 & 0 & 0 & 0 & 0 \\ 0 & k_1 & k_2 & 0 & k_3 & k_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & k_1 & k_2 & 0 & k_3 & k_4 & 0 \\ 0 & 0 & 0 & 0 & k_1 & k_2 & 0 & k_3 & k_4 \end{bmatrix}$$

$\underbrace{\underbrace{4}_{\text{Input size}} \times \underbrace{9}_{\text{output size}}}_{\text{flattened}}$

O-f: Flattened Output

$$\boxed{\underset{9\times1}{O\text{-}f} = \underset{9\times4}{(K\text{-}f)^T}\ \underset{4\times1}{(I\text{-}f)}}$$

$\therefore$ If $O\text{-}f = \begin{bmatrix} O_1 \\ O_2 \\ O_3 \\ O_4 \\ O_5 \\ O_6 \\ O_7 \\ O_8 \\ O_9 \end{bmatrix} = \begin{bmatrix} k_1x_1 \\ k_2x_1 + k_1x_2 \\ k_2x_2 \\ k_3x_1 + k_1x_3 \\ k_4x_1 + k_3x_2 + k_2x_3 + k_1x_4 \\ k_4x_2 + k_2x_4 \\ k_3x_3 \\ k_4x_3 + k_3x_4 \\ k_4x_4 \end{bmatrix}$

$\Rightarrow$ Output O: Reshaped version

$$O = \begin{bmatrix} O_1 & O_2 & O_3 \\ O_4 & O_5 & O_6 \\ O_7 & O_8 & O_9 \end{bmatrix}$$

$$
O = \begin{bmatrix} k_1 x_1 & k_1 x_2 + k_2 x_1 & k_2 x_2 \\ k_1 x_3 + k_3 x_1 & k_1 x_4 + k_2 x_3 + k_3 x_2 + k_4 x_1 & k_2 x_4 + k_4 x_2 \\ k_3 x_3 & k_3 x_4 + k_4 x_3 & k_4 x_4 \end{bmatrix} //
$$