Date

LSTM

Gates

controls what is kept vs forgotten from prev cell state

$$f(t) = \sigma(W_f \cdot H(t-1), x_t + B_f)$$

what parts of new cell content are written to cell.

$$i(t) = \sigma(W_i \cdot H(t-1), x_t + B_i)$$

what parts of cell are output to hidden state.

$$o(t) = \sigma(W_o \cdot H(t-1), x_t, + B_o)$$

New Cell Content

$$\tilde{c_t} = \tanh(W_c \cdot H(t-1), x_t + B_c)$$

States

erase "forget" some content from last cell, write "input" some new cell content.

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c_t}$$

$$h_t = o_t + \tanh(c_t)$$

read "output"

We've thoroughly practiced employing LSTM (Long Short-Term Memory) in our previous assignment to forecast upcoming work tasks. The current objective involves computing values for the below given tasks.

1. Compute embedding from the given target weight matrix based on One Hot vector: [0 1 0 0]
2. Compute value for forget gate from the data given below.
3. Compute C_t & h_t value from all supporting values given below.
4. Write Equations for finding C_t & h_t.

Target Weight Matrix:

1	1	3	4
2	3	3	4
3	1	1	0
4	0	2	4

Weight Matrix for Input Gate:

6	4	3	1	5	6	2	0
0	0	6	6	1	3	3	5
6	4	2	3	4	5	1	2
6	6	2	6	1	4	0	0

Bias for Input Gate:

4
4
2
1

Forget Gate	Input Gate	Output Gate	h_{t-1}	c_{t-1}	$\tilde{c_t}$
1		1	0.76	1	1
1		1	0.76	1	1
1		1	0.76	1	1
1		1	0.76	0.99	1

Solution: (Show Steps)

Date

LSTM

Question No: 1

One Hot vector = [0 1 0 0]

Target weight matrix

$$\begin{bmatrix} 1 & 1 & 3 & 4 \\ 2 & 3 & 3 & 4 \\ 4 & 1 & 1 & 0 \\ 2 & 0 & 2 & 4 \end{bmatrix}$$

Weight matrix for input gate

$$\begin{bmatrix} 6 & 4 & 3 & 1 & 5 & 6 & 2 & 0 \\ 0 & 0 & 6 & 6 & 1 & 3 & 3 & 5 \\ 6 & 4 & 2 & 3 & 4 & 5 & 1 & 2 \\ 6 & 6 & 2 & 6 & 1 & 4 & 0 & 0 \end{bmatrix}$$

$$b_i = \begin{bmatrix} 4 \\ 4 \\ 2 \\ 1 \end{bmatrix}, \quad f_t = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad I_t = \begin{bmatrix} ? \end{bmatrix}, \quad O_t = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$h_{t-1} = \begin{bmatrix} .76 \\ .76 \\ .76 \\ .76 \end{bmatrix}, \quad C_{t-1} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0.99 \end{bmatrix}, \quad \hat{C}_t = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Solution

$$\text{Embeddings} \rightarrow [0\ 1\ 0\ 0] \begin{bmatrix} 1 & 1 & 3 & 4 \\ 2 & 3 & 3 & 4 \\ 4 & 1 & 1 & 0 \\ 2 & 0 & 2 & 4 \end{bmatrix}$$

$$\Rightarrow [2 \quad 3 \quad 3 \quad 4]$$

Date

$$I_t = \sigma(W_i \cdot H_{t-1}, x_t + b_i)$$

$$= \sigma \begin{bmatrix} 6 & 4 & 3 & 1 & 5 & 6 & 2 & 0 \\ 0 & 0 & 6 & 6 & 1 & 3 & 3 & 5 \\ 6 & 4 & 2 & 3 & 4 & 5 & 1 & 2 \\ 6 & 6 & 2 & 6 & 1 & 4 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0.76 \\ 0.76 \\ 0.76 \\ 0.76 \\ 2 \\ 3 \\ 3 \\ 4 \end{bmatrix}$$

$$+ \begin{bmatrix} 4 \\ 4 \\ 2 \\ 1 \end{bmatrix}$$

$$I_t = \begin{bmatrix} 0.017 \\ 0.980 \\ 0.003 \\ 0 \end{bmatrix}$$

$$C_t = f_t \odot C_{t-1} + I_t \odot \tilde{C_t}$$

$$C_t = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \odot \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0.99 \end{bmatrix} + \begin{bmatrix} 0.017 \\ 0.98 \\ 0.003 \\ 0 \end{bmatrix} \odot \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$C_t = \begin{bmatrix} 1.017 \\ 1.98 \\ 1.003 \\ 0.99 \end{bmatrix}$$

Date

$$ht = Ot \odot \tanh(ct)$$

$$ht = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \odot \tanh \begin{bmatrix} 1.017 \\ 1.98 \\ 1.003 \\ 0.99 \end{bmatrix}$$

$$ht = \begin{bmatrix} 0.769 \\ 0.963 \\ 0.763 \\ 0.757 \end{bmatrix}$$

Q) We've thoroughly practiced employing LSTM (Long Short-Term Memory) in our previous assignment to forecast forthcoming work tasks. The current objective involves computing values for the candidate cell state ($\tilde{c_t}$) and hidden state (h_t) and cell unit (c_t) at the next timestamp, using the provided prior information. [10]

```
Weight Matrix Values:
Weights and Bais for Forget Gate
[[1 6 0 0]
 [3 3 6 0]]
[[1]
 [0]]
Weights and Bais for Input Gate
[[5 1 2 2]
 [6 3 2 1]]
[[0]
 [3]]
Weights and Bais for Update Gate
[[0 4 0 5]
 [2 6 2 5]]
[[1]
 [1]]
Weights and Bais for Output Gate
[[4 3 4 4]
 [2 2 0 4]]
[[illegible]
 [illegible]]
```

```
Time =  2
Privous Hidden State (ht):
 [[0.4165792 ]
 [0.32134238]]
Privous Cell State (ct):
 [[0.44354576]
 [1.99999732]]
Input :
 [[0]
 [2]]
Values for Forget Gate:
 [[0.33209835]
 [0.9014788 ]]
Values for Input Gate:
 [[0.99999057]
 [0.333263  ]]
Values for Output Gate:
 [[0.33333224]
 [0.99998962]]
```

Question No. 2

$$W_c = \begin{bmatrix} 3 & 4 & 0 & 5 \\ 2 & 6 & 2 & 5 \end{bmatrix} \qquad W_f = \begin{bmatrix} 4 & 6 & 0 & 0 \\ 3 & 3 & 6 & 0 \end{bmatrix}$$

$$I = \begin{bmatrix} 0 \\ 2 \end{bmatrix}, \quad b_c = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad C_{t-1} = \begin{bmatrix} 0.4435 \\ 1.9999 \end{bmatrix}$$

$$\cancel{h_{t-1} =} \; I_t = \begin{bmatrix} 0.9999 \\ 0.33333 \end{bmatrix}, \quad O_t = \begin{bmatrix} 0.333 \\ 0.999 \end{bmatrix}$$

$$h_{t-1} = \begin{bmatrix} 0.4166 \\ 0.3213 \end{bmatrix}, \quad f_t = \begin{bmatrix} 0.3321 \\ 0.9015 \end{bmatrix}$$

Solution

$$\tilde{C}_t = \tanh(W_c \cdot H_{t-1}, x_t + b_c)$$

$$= \tanh \begin{bmatrix} 3 & 4 & 0 & 5 \\ 2 & 6 & 2 & 5 \end{bmatrix} \cdot \begin{bmatrix} 0.4166 \\ 0.3213 \\ 0 \\ 2 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

CamScanner

Date

$$\hat{Ct} = \tanh\begin{bmatrix}13.53\\13.75\end{bmatrix}$$

$$\hat{Ct} = \begin{bmatrix}1\\1\end{bmatrix}$$

$$Ct = ft \odot Ct_{-1} + It \odot \hat{Ct}$$

$$= \begin{bmatrix}0.3321\\0.9015\end{bmatrix}\begin{bmatrix}0.4435\\1.9999\end{bmatrix} + \begin{bmatrix}0.9999\\0.3333\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix}$$

$$= \begin{bmatrix}1.1469\\2.1363\end{bmatrix}$$

$$ht = Ot \odot \tanh(Ct)$$

$$ht = \begin{bmatrix}0.3321\\0.9015\end{bmatrix} \odot \tanh\begin{bmatrix}1.1469\\2.1363\end{bmatrix}$$

$$ht = \begin{bmatrix}0.2723\\0.9719\end{bmatrix}$$

Date

GRU

$$R_t = \sigma(X_t W_{xr} + H_{t-1} W_{hr} + b_r)$$

$$Z_t = \sigma(X_t W_{xz} + H_{t-1} W_{hz} + b_z)$$

$$\hat{H}_t = \tanh(X_t W_{xh} + (R \odot H_{t-1}) W_{hh} + b_h)$$

$$H_t = Z_t \odot H_{t-1} + (1 - Z_t) \odot \hat{H}_t$$

Why is vanishing gradient a problem?

If the gradient becomes vanishingly small over long distances, then we can't tell :

→ there's no dependency b/w step t & $t+n$ in data

→ we have wrong parameters to capture true dependency b/w t and $t+n$.

Effect of vanishing gradient on RNN-LM?

Due to vanishing gradient RNN-LM are better at learning from sequential recency than syntactic recency, so they make this type of error more often than we'd like.

if gradient is small
↳ model is unable to predict similar-long distance dependencies at test time.

Date

Why is exploiding gradient a problem ?

if gradient becomes too big SGD update step becomes too big :

$$\theta^{new} = \theta^{old} - \alpha \nabla_\theta J\theta$$

→ causes bad updates with large loss.
→ in worst case result is in Inf or NAN in network.

Solution → gradient clipping

if norm of gradient is > then some threshold scale it down before applying SGD update

intitution → take step in same direction but smaller step.

How to fix vanishing gradient ?

GRU
LSTM

RNN in terms of x, y, h

$$a_t = \underset{d \times d}{W_h} \, \underset{d \times 1}{H_{t-1}} + \underset{d \times k}{W_x} \, \underset{k \times 1}{x_t}$$

$$h_t = \tanh(a_t)$$

$$y = \text{softmax}(W_y h_t)$$

Date

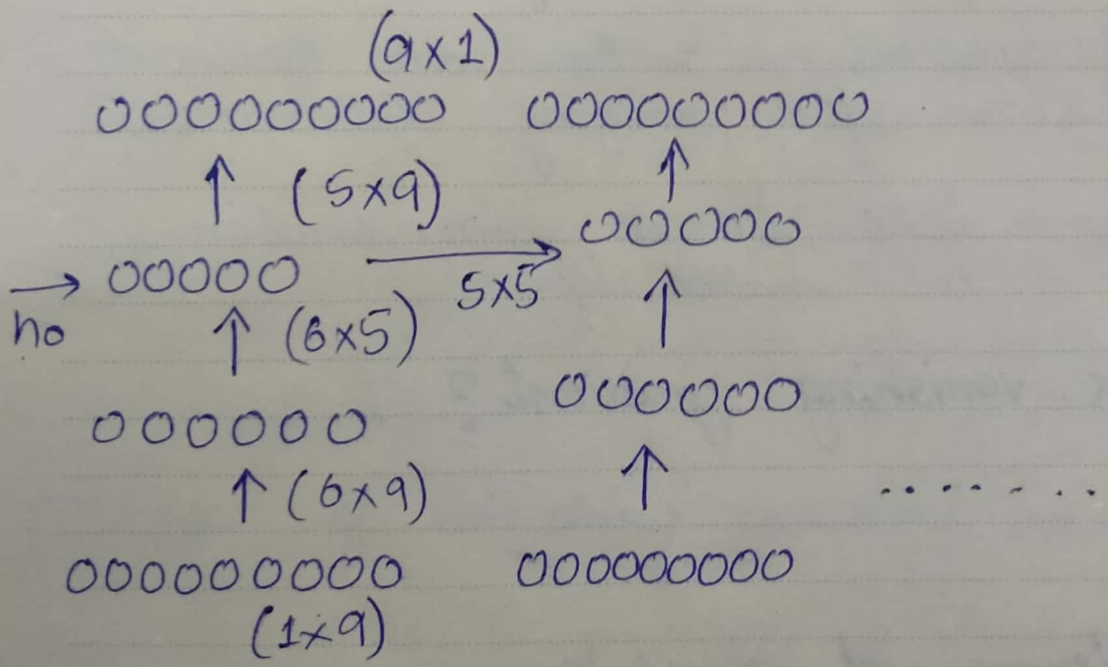
RNN in terms of embeddings

$$e^{(t)} = E(x^t)$$

$$h^{(t)} = \sigma(W_h h^{(t-1)} + W_e e^{(t)} + b_1)$$

$$\hat{y}^{(t)} = \text{softmax}(U h^{(t)} + b_2).$$

Example $V = 9$, $E = 6$, $h = 5$, $seq = 4$



Total parameters = $(6\times9) + (6\times5) + (5\times9) + (5\times5)$

$= 154$

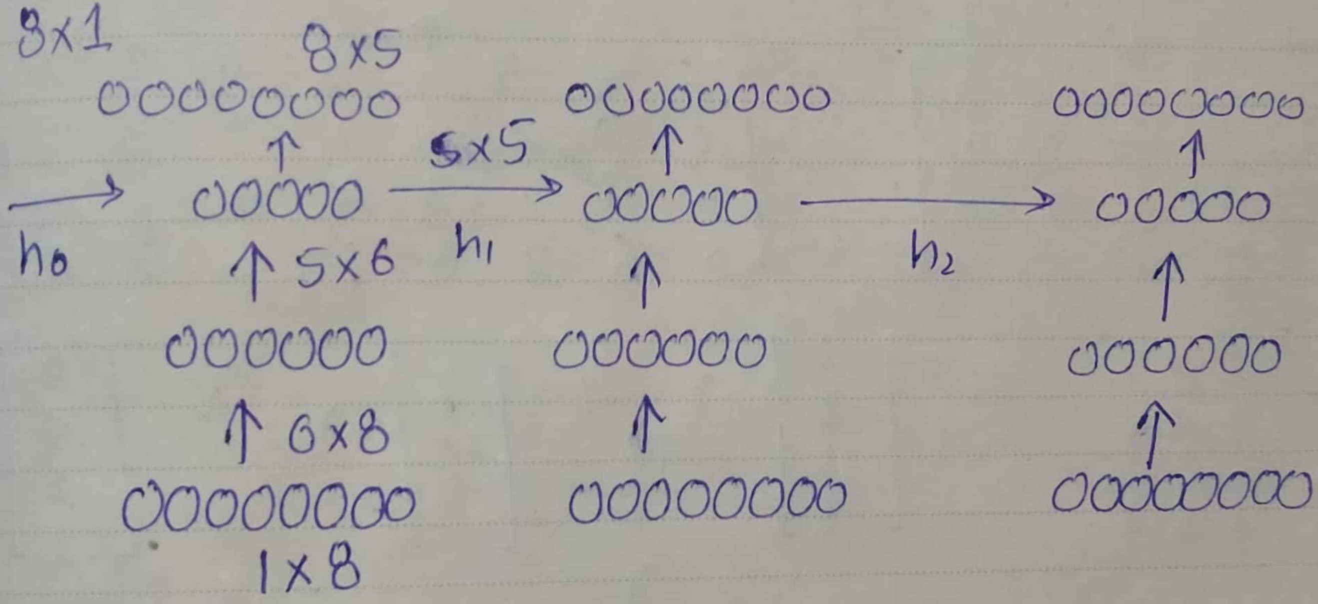
Date

Example seq = 3, V = 8, E = 6, h = 5

$$a_t = (5\times5)(5\times1) + (5\times8)(8\times1) = 5\times1$$

$$h_t = \tanh(5\times1)$$

$$y = \text{softmax}((8\times5)(5\times1)) = s(8\times1)$$



$$\text{Total parameters} = (6\times8) + (5\times6) + (5\times5) + (8\times5)$$

$$= 143$$