Q.

Deep Learning Assignment

Saideva Nathi
(Z23737856)

Q1) fig[1] shows the structure of an RNN cell vs. an LSTM cell

- Summarize the major difference between RNN cell vs LSTM cell interms of their neutral architectures

i) According to me the major difference between a standard RNN cell and an LSTM cell lies in their architectures and their ability to handle long-term dependencies.

RNN Cell:

- Therepeating module in a standard RNN consists of a single layer
- Standard RNNs have a simple structure and can be thought of as multiple copies of the same network, each passing a message to a successor.
- RNN's struggle with learning & maintaining information over long sequences.

LSTM cell:-

- It has more complex repeating module compared to RNN's
- The repeating module in LSTM consists of four interactive layers:

  i) input gate
  ii) forget gate
  iii) cell state
  iv) output gate

- LSTM are explicitly designed to address the long-term dependency problem, make them capable of learning & remembering info for long periods.

- why LSTM can achieve long-short term memory, where as RNN cannot.

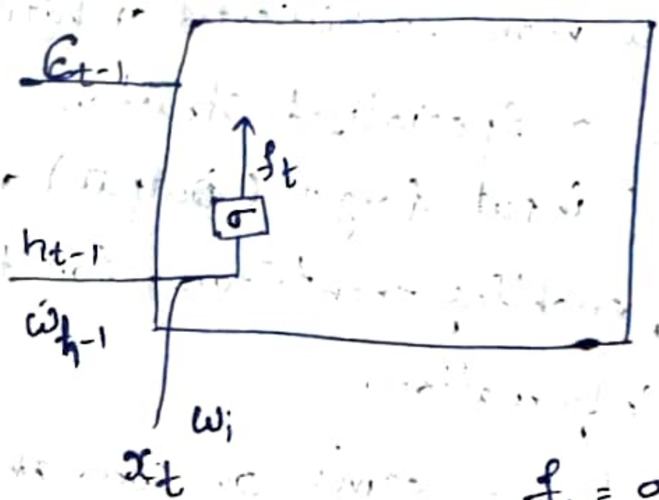
The main reason why LSTMs can capture long term dependencies while standard RNNS struggle is due to their more sophisticated architechture.

- LSTM's have a specialized structure with gates (input, forget, & output) and a cell state, enabling them to store, forget and retreive information.
- This design helps LSTM's overcome the vanishing gradient problem, making them more effective at learning

Q2)

- please Mark forget gate, input gate, output gate, and Candidate layer, respectively.

i) forget gate

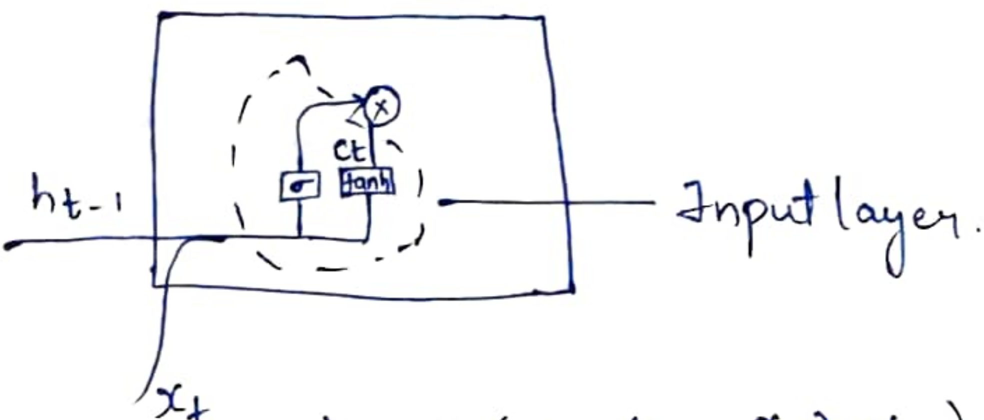


$W_f = [W_i, W_{h-1}]$

$\cancel{W_f = W_{h-1}}$,

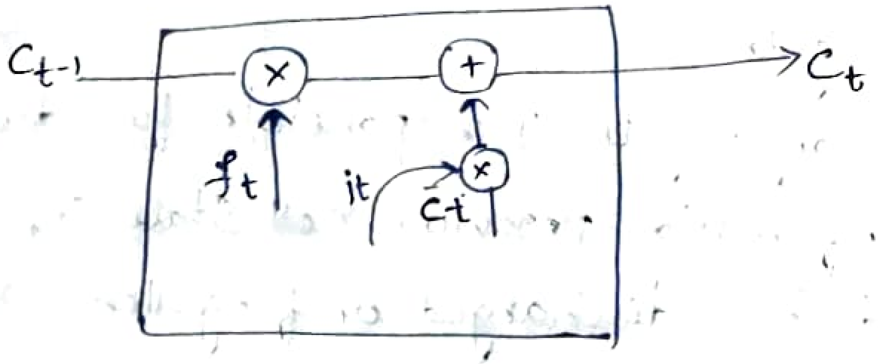
$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$

ii) Input Layer



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

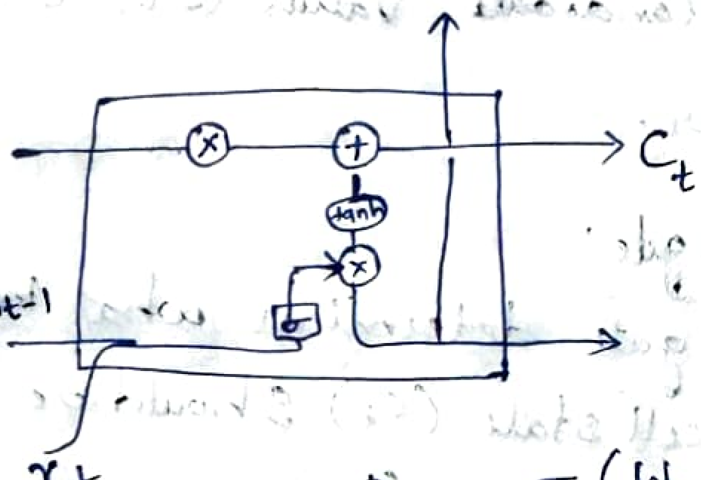
$\bar{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$



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

iii) Candidate layer

iv) output layer



$$O_t = \sigma(W_o(h_{t-1}, x_t) + b_o)$$

$$h_t = O_t * \tanh(C_t)$$

- Explain the main role/ functionality of forget gate, input gate, output gate, and Candidate layer, respectively.

i) forget gate:

- The forget gate is responsible for deciding what information perevious cell state (C_{t-1}) should be discharged or forgotten.

ii) Input gate:-

- The input gate has two parts: it ~~includes~~ decides which values from the current input (x_t) should be updated, and it generates a vector of a new candidate values ($\tilde{C}_t$) using a tanh layer.

iii) Output gate:-

The output gate determines what information from the cell state (C_t) should be used to generate the output (h_t)

iv) Candidate layer:- It represents a vector of a new candidate values (C_t) that could be added to the cell state. it is generated by the tanh layer in collaboration with input gate.

Q3) Figure shows Unfolded LSTM network with two Consequtive Cells. Using h_t and c_t to denote output and Cell memory of cell at a time point t. Use $\bar{c}_t$ to denote Candidate layer output at time point t.

- Relationship between cell memory at time t and cell memory / output at time $t-2$:

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \bar{c}_t$$

Cell memory (c_t) is updated based on the forget gate (f_t), the previous cell memory (c_{t-1}), the input gate (i_t), and the Candidate layer output ($\bar{c}_t$)

- Derive Relationship between δc_{t-2} & δc_t:

Using chain rule we can Express δc_t in interms of δc_{t-2}:

$$\delta c_t = \frac{\partial E}{\partial c_t} + \frac{\partial E}{\partial h_t} \cdot \frac{\partial h_t}{\partial c_t} \cdot \frac{\partial c_t}{\partial c_{t-1}} \cdot \frac{\partial c_{t-1}}{\partial c_{t-2}} \cdot \frac{\partial c_{t-2}}{\partial c_{t-2}} \cdot \delta c_{t-2}$$

This Expression demonstrates the relationship between the error gradient with respect to the cell State at that time t (δC_t) and the error gradient with respect to cell state at time $t-2$ (δC_{t-2}).

- Explain why LSTM cell can Alleviate weight Vanishing or Exploding:

- Due to their gating Mechanisms.

The forget gate (f_t), input gate (i_t), and output gate (O_t) provide Explicit Control over the flow of information, addressing challenges associated with gradient issues.

- In Summary, the gating in LSTMs enabled Controlled information flow, mitigating the problems of Vanishing and Exploding gradients Commonly encountered in deep Neural Network training.

Q4) The following Keras show a deeplearning for text classification

1) Embedding Layer:

model. add (embedding (1000, 16, input_length = 200.))

purpose of Embedding:

- The embedding is used to Convert integer-encoded words into dense vectors of fixed size.

- It is often the first layer in a text classification model and is crucial for learning representations of words.

Embedding Layer output Size (16):

- This means each word in the input will be represented by a dense vector of Size 16.

Number of weight parameters for embedding layer :-

- The number of parameter is determined by the vocabulary size (1,000) and the output size (16) formula:-

Vocabulary Size x Output Size

- 1000 x 16 = 16,000 weight palameters.

2) LSTM Layer!

Number of weight parameters for LSTM Layer:

- ~~The~~ it is determined by its input size, output size, the presence of bia terms.

- Formula: 4 x ((input Size + 1) x Output Size + Output Size^2)

- $4 \times ((16+1) \times 32 + 32^2) = 4 \times (544 + 1024)$

$= 4 \times (544 + 1024) = 4 \times 1568 = 6272$ weight

#

```
model.add(LSTM(32, dropout=0.1,
               recurrent_dropout=0.1))
```

3) Last Two Dense Layers:

```
# model.add(Dense(256, activation='Sigmoid'))
model.add(Dense(1, activation='Sigmoid'))
```

Total Number of parameters for Last two dense Layers:

- The first dense layer has $16 \times 256 + 256 =$ 4352 parameters
- The Second ~~ra~~ dense layer has $256 \times 1 + 1 = 257$ parameters.
- Total: $4352 + 257 = 4609$ weight parameters.

In Conclusion:-

1) Embedding Layer: 16,000 parameters
2) LSTM Layer: 6,272 parameters
3) Last two dense Layers: 4,609 parameters.