Next word Generator

I - Input - Output pairs Determination

1. LLMs predict next ~~word~~ token based on a set of previous tokens
2. Total number of tokens taken as the basis is known as Context window.
3. LLMs are ~~auto~~ regressive, i.e. They predict the next Token based on the training set which is nothing but previous tokens predicted
4. Example :

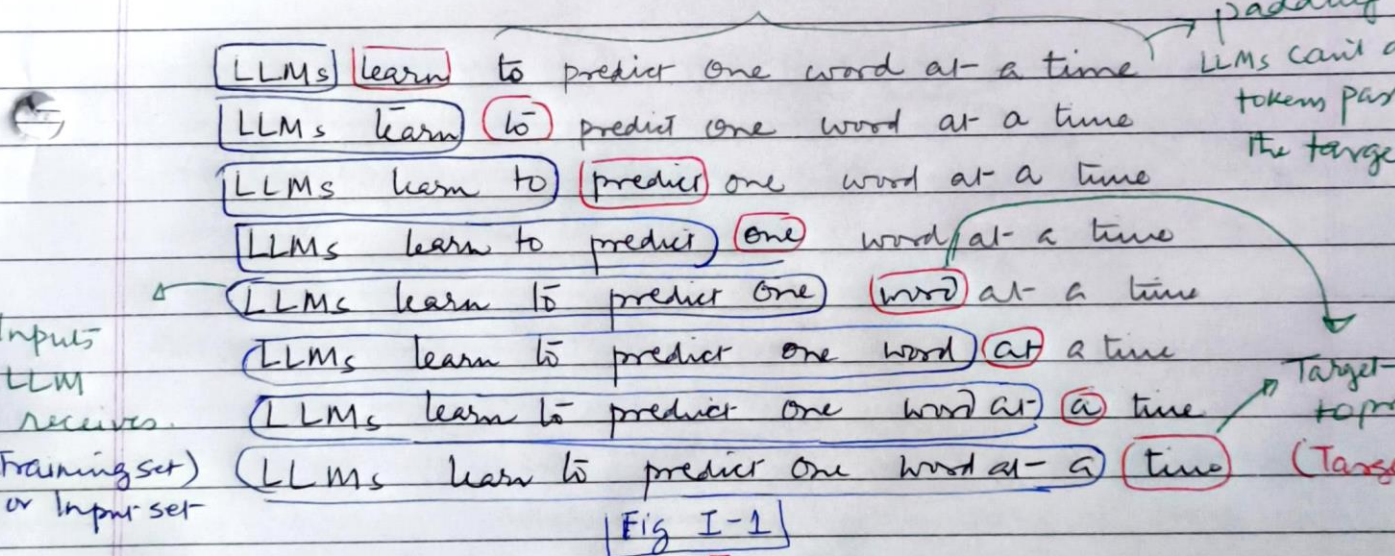


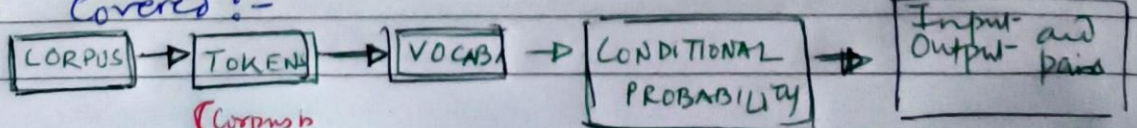
Fig I-1

5. Tokens are converted into Token Ids and then processed. Token Ids are numerical representation of each token.
6. ~~For computation efficiency~~ Each Token that is predicted above it via joint probability

$$P(w_1, w_2, w_3, w_4) = P(w_4 \mid w_1, w_2, w_3) \times \text{probability of a given Sentence}$$
$$= P(w_1) \times P(w_2 \mid w_1) \times P(w_3 \mid w_1, w_2) \times P(w_4 \mid w_1, w_2, w_3)$$

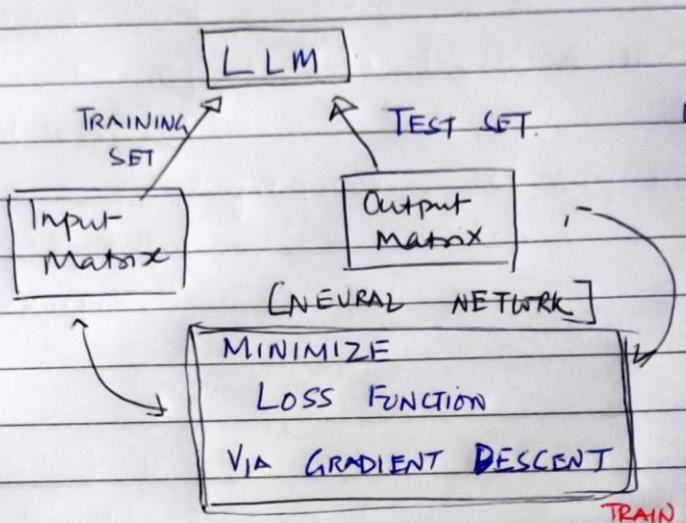
~~x P~~

7. This probability is formed based on the ~~Corpus~~ Vocabulary that is formed that is used to make the vocabulary in the 1st stage out of the Corpus.
8. NO machine learning has happened till now!
9. Steps Covered :-



II - TRAINING THE LLM -

1.



The LLM is trained using the Input/Output pairs determined in Step I-9.

2. For Computational Efficiency the input and output sets sent to LLM IS NOT as shown in I-9's figure Fig I-1.
Rather it is loaded in BATCHES as follows :-

TRAIN TEST

TRAINING SET (Diff from Corpus that was used to Build Vocab)

"In the heart of the city stood the old library, A relic from a bygone era. Its stone walls bore the marks of time, and ivy clung tightly to its facade ..."

Input Matrix: x = ([["In", "the", "heart"],
["of", "the", "city"],
["stod", "the", "old"],
[...]])

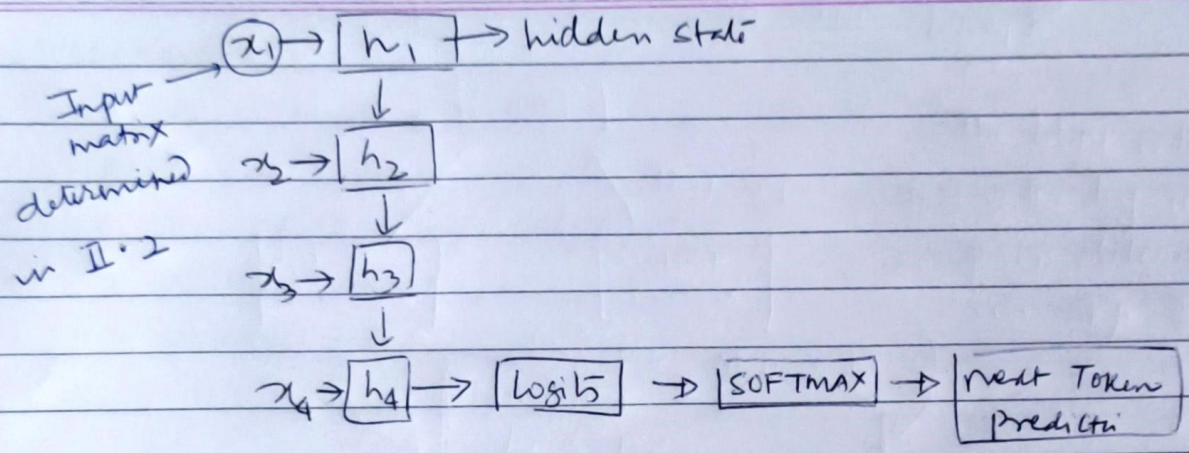
Output Matrix: y = ([["the", "heart", "of"],
["the", "city", "stood"],
[...]])

all Tokens are not characters but token Ids!

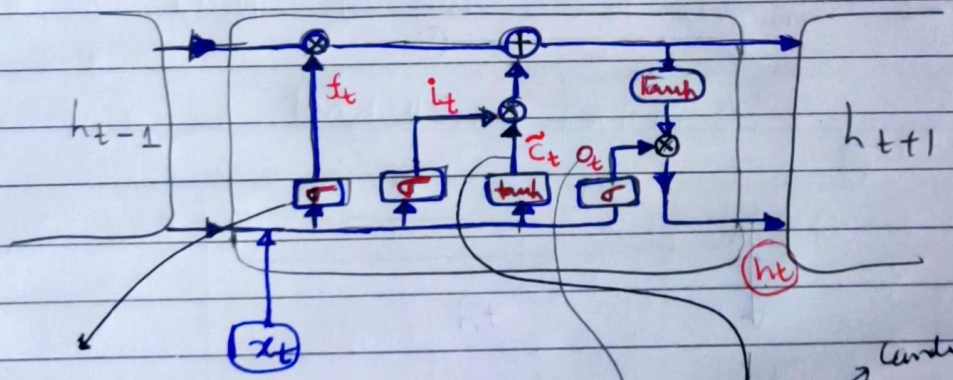
3. Using these Test- and Train matrices the neural network is trained.

III LSTM as the Neural Net (LLM) THAT IS TRAINED

1.



2. Making of the hidden state!



Forget gate

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

$W_f = [\ ]$ hidden size × Embedd size

Candidate cell state

$$\tilde{C}_t = \tanh\left(W_c \cdot [h_{t-1}, x_t] + b_c\right)$$

$[\ ]$ h×e

Cell state

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

Output state

$$o_t = \sigma\left(W_o \cdot [h_{t-1}, x_t] + b_o\right)$$

$W_o = [\ ]$ hidden size × Embedd size

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

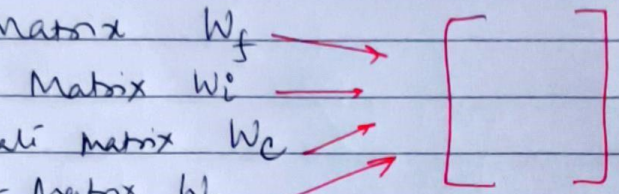
$W_i = [\ ]$ hidden size × embed size

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

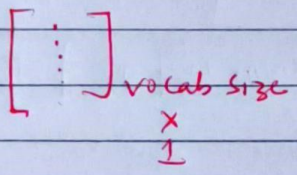
hidden state

3. MATRICES :

I. Forget Matrix W_f
II. Input Matrix W_i
III. Candidate matrix W_c
IV. Output Matrix W_o



V. Logits Vector →



4. Recap :