14.0 NEXT WORD PREDICTOR USING PyTorch & LSTM USING PyTorch

24 August 2025 02:49 AM Avinash Yadav

IMPLIMENTATION OF LSTM USING PYTORCH

↓

UTILISING THIS CONCEPT TO BUILD NEXT WORD PREDICTOR

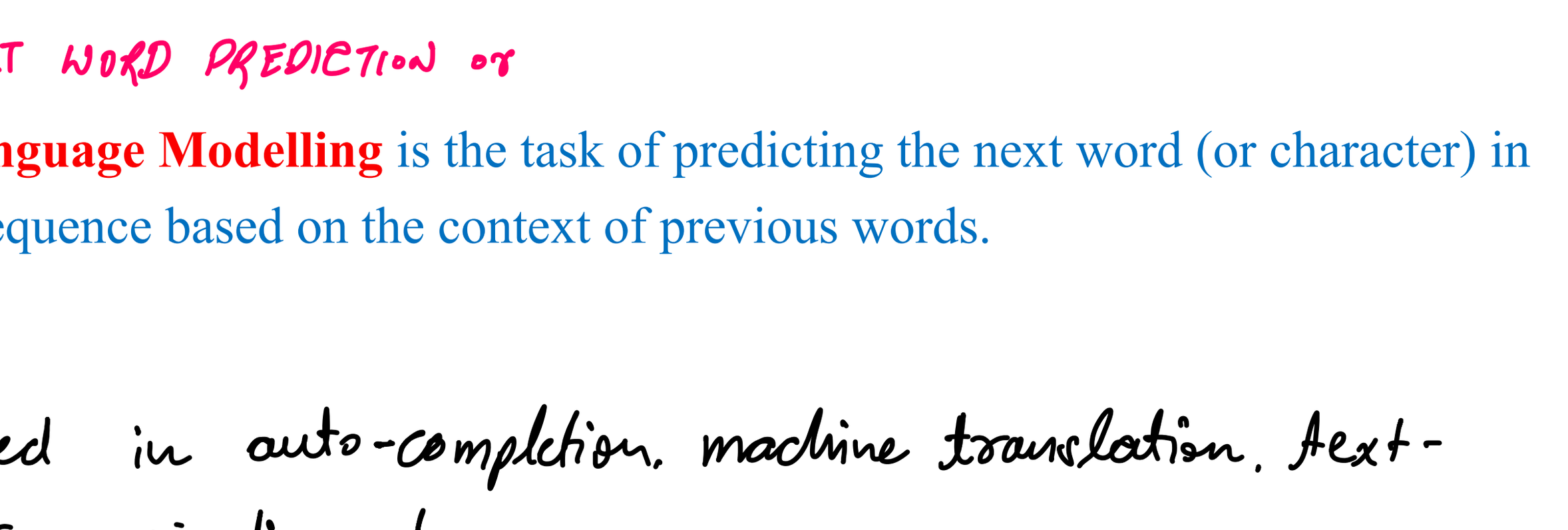
- USEFUL IN KEYBOARDS SMARTPHONES, SEARCH ENGINE FXNALITY, CHATBOTS ETC
- DOWN THE LINE WE'LL ALSO EXPLORE LANGUAGE MODELLING CONCEPT IN THIS PROCESS

STRATEGY :

NEXT WORD PREDICTION or

Language Modelling is the task of predicting the next word (or character) in a sequence based on the context of previous words.

Used in auto-completion, machine translation, text-summarization etc.


What is the course fee for Data Science Mentorship Program (DSMP 2023)
The course follows a monthly subscription model where you have to make monthly payments of Rs 799/month.
What is the total duration of the course?
The total duration of the course is 7 months. So the total course fee becomes 799*7 = Rs 5600(approx.)
What is the syllabus of the mentorship program?
We will be covering the following modules:
Python Fundamentals
Python libraries for Data Science
Data Analysis
SQL for Data Science
Maths for Machine Learning
ML Algorithms
Practical ML
MLOPs
Case studies

UNSUPERVISED DATASE → Means we do not have target labels here

So here we are given with some text, so in language modelling, first we need to convert the unsupervised data to supervised data

Means we will make input feature and target labels as well. So, first of all we'll split all the sentences from unsupervised dataset.

Example: Python libraries for data science

input	output
Python →	libraries
Python libraries →	for
Python libraries for →	data
Python libraries for data →	science

Now this same task would be repeated for all the other sentences in the dataset. So finally we'll have a dataset which will have input feature target label as well

The next task would be forming a VOCAB – a dictionary where each unique word is mapped with a number

```
VOCAB = { 'Python' : 1,
          'libraries' : 2,
          'for' : 3,
          'data' : 4,
          'Science' : 5
             ⋮
        }
```

Now Based on this VOCAB replace the word in new dataset with the values of the word, i.e. indexes

input	output
[1] →	[2]
[1, 2] →	[3]
[1, 2, 3] →	[4]
[1, 2, 3, 4] →	[5]

After this we'll convert every word into embeddings like [1, 2] ⇒ [[.....], [.....]]

↓

This will now go to our LSTM.

TOP LEVEL STEPS :

1. Data Pre-processing
   - 1.1) Unsupervised → supervised
   - 1.2) English VOCAB → NUMBER VECTOR
   - 1.3) EMBEDDINGS
2. Model Creation
3. Training
4. Prediction on trained model

4 LSTM MODEL ARCHITECTURE :

As of now { X: shape = 938, 62   Y: shape = 938 } for our taken documented dataset.

At each time we are taking a batch of size 32 so shape of [32, 62] batch, we are taking and sending it to our model. [32, 62] → 32 sentences, each sentence has 62 words

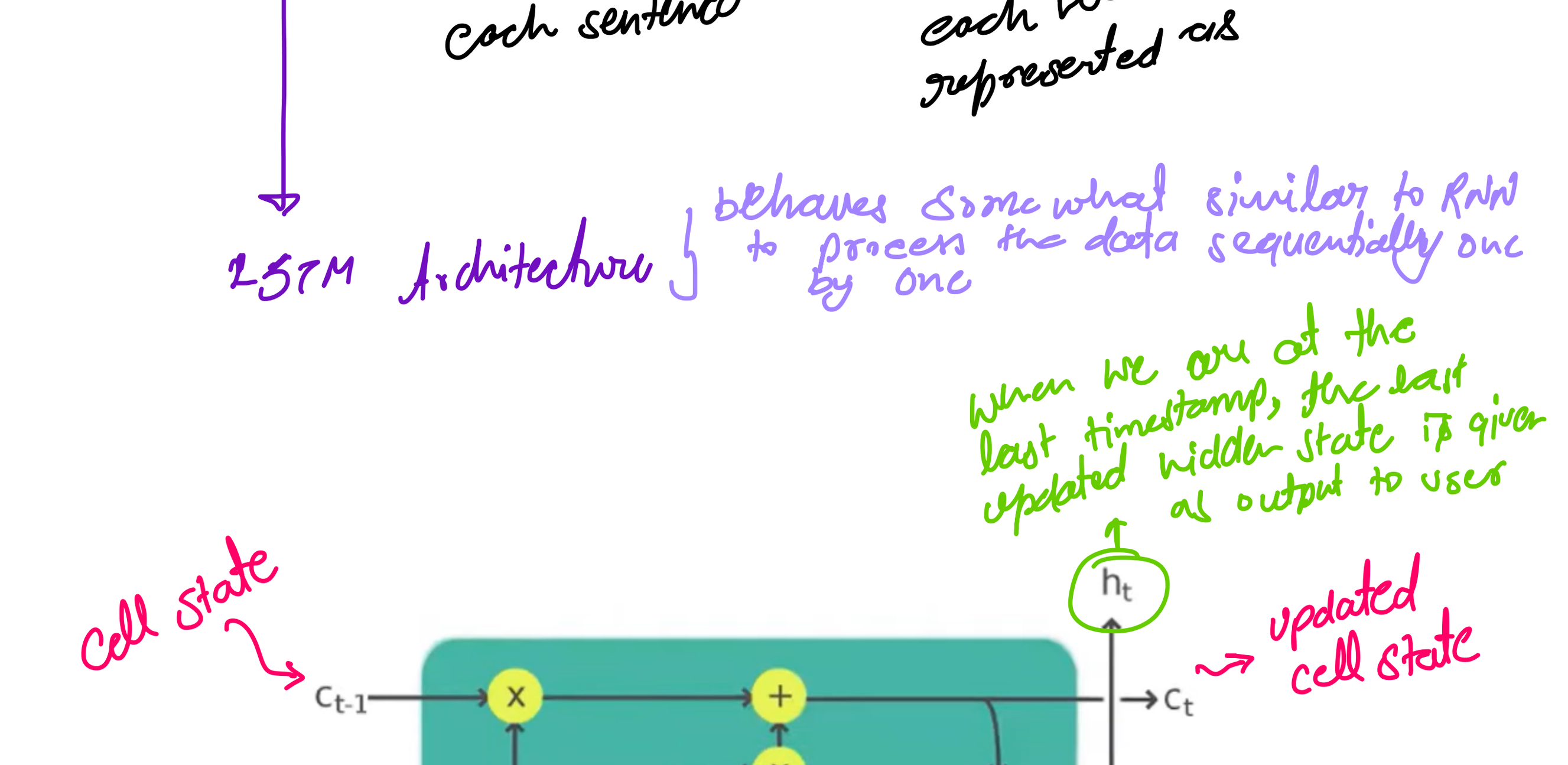
↓

embedding layer { 32 sentences / batch → 62 words / sentence → vector of dim = 100 for each word }

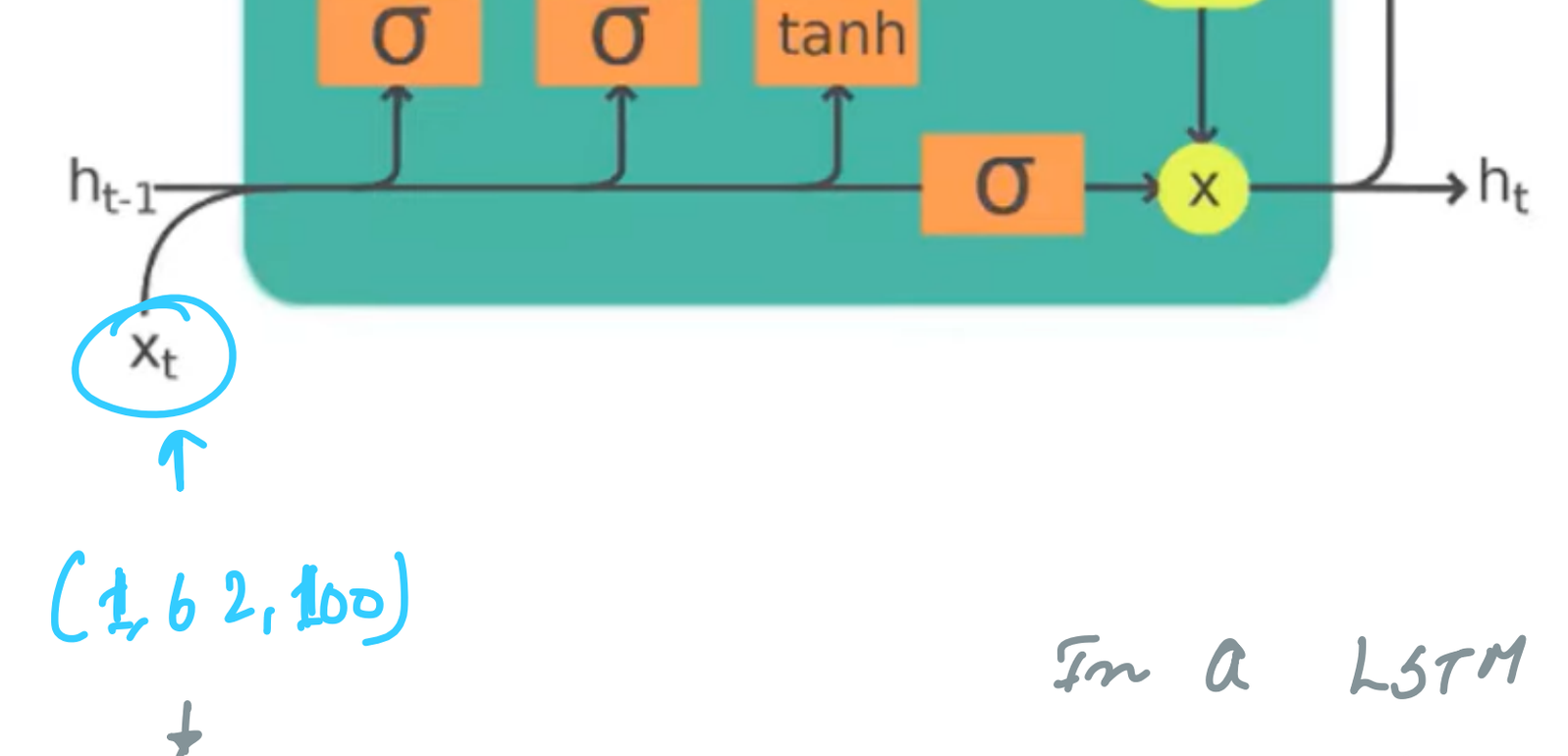
↓

Tensor ← Result ⇒ (32, 62, 100) → # sentences, # words, # vector dim = 100 (each sentence has # words; each word is represented as # vector dim)

Tensor goes to → LSTM Architecture { behaves somewhat similar to RNN to process the data sequentially one by one



Performing a Single sequence of data by LSTM i.e. (1, 62, 100)

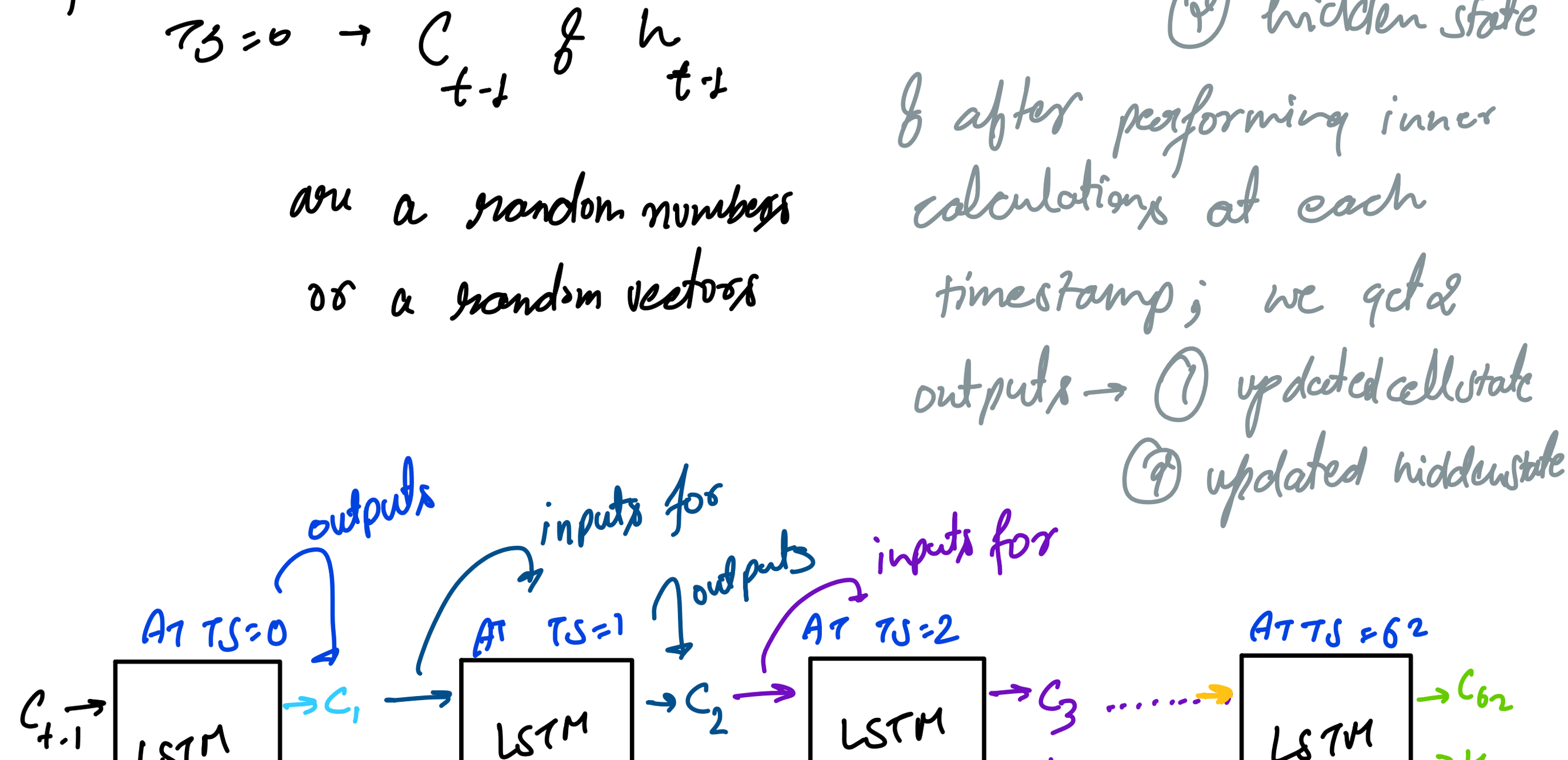


(1, 62, 100) ↓ in other words, we can say that, we have a sentence of 62 words where each word has 100 dimensional vector size.

In a LSTM cell (code), we send a single word or related embedding of it as input at one single timestamp and at each timestamp we get 2 more thing as input → (1) cell state (2) hidden state & after performing inner calculations at each timestamp; we get 2 outputs → (1) updated cell state (2) updated hidden state

for timestamp = 0

TS = 0 → C_{t-1} & h_{t-1} are a random numbers or a random vectors



Performing a Batch sequence of data by LSTM i.e. (32, 62, 100)

So the task we performed above for one sequence, the same task will be performed parallelly for the number of batches (# sentences)

i.e. parallelly 32 sequences will be executed at TS=0 and rest of all the calculations all together as that of above we did for one sequence.

The nn.LSTM gives 3 outputs :→ we get 3 things from LSTM after its processings :⇒