NLP is all about?

Natural Language Processing or NLP is a field of Artificial Intelligence that gives the machines the ability to read, understand and derive meaning from human languages.

Components of NLP

1. Natural Language Processing
2. Natural Language Understanding
3. Natural language Generation

"Hey Google, how did the stock market do today?"

Challenges

1. treating the word “board” as noun or verb?

1. “He lifted the beetle with red cap.” − Did he use cap to lift the beetle or he lifted a beetle that had red cap?

1. Rima went to Gauri. She said, “I am tired.” − Exactly who is tired?

1. Unstructured data

Key Terms

1. **Syntactic features:** Focus on arrangement of words matter (e.g. computer virus)
2. **Semantic features:** Focus on meaning of words (e.r. Strong coffee, strong muscles)
3. **Corpus:** Collection of text data
4. **Vocabulary:** Set of unique tokens in the corpus
5. **Lexicon:** Vocabulary which include words and expression
6. **Morphological Analysis:** Analysis, identification, description of structure of words

Libraries

1. NLTK
2. Gensim
3. CoreNLP
4. Spacy

Roadmap for learning NLP

1 Understand text data

a. Linguistics, technical terms
b. Linguistics databases

2 Text pre-processing

a. Parsing the data (web scraping)
b. Tokenization
c. Stemming, Lemmatization
d. Regex expression
e. Stop words removal

3 Feature Extraction

a. Vocabulary
b. Feature extraction based on frequency
c. Dimensionality reduction
d. POS tagging, NER
e. Word embedding

4 Domain based concepts

a. Spelling similarity
b. Semantic similarity
c. Topic modelling
d. Latent drichilet allocation

5 Machine Learning

a. Classification based approach
b. Clustering based approach

6 Deep Learning

a. Deep Learning basics
b. RNN, LSTM, GRU
c. Seq2seq modelling

7 Advanced models

a. Encoder, Decoder,
b. Attention models,
c. Transformers, BERT

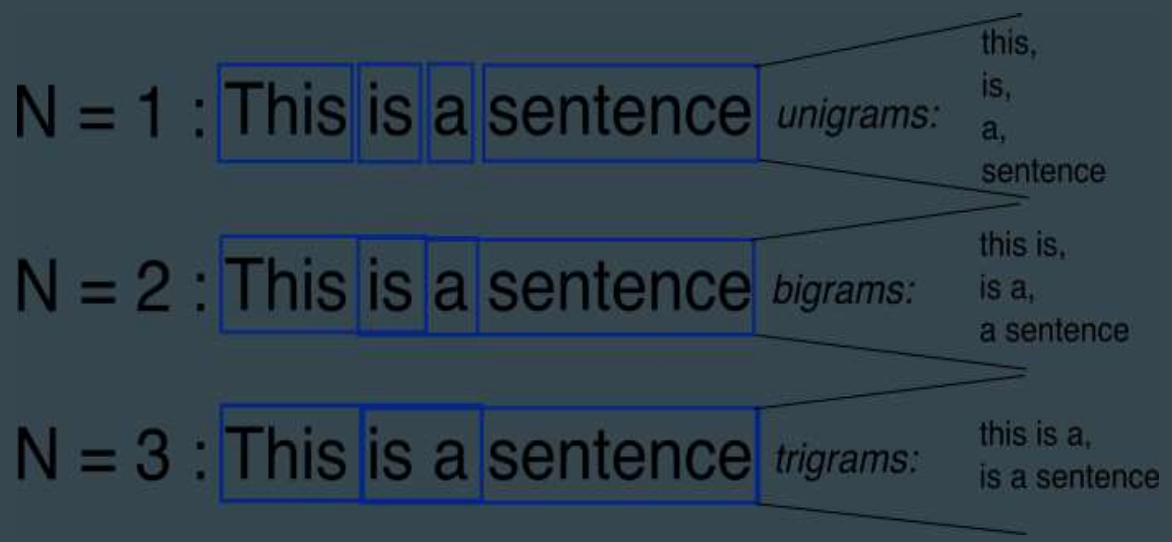
https://www.letthedataconfess.com/

Step 1: Data Normalization

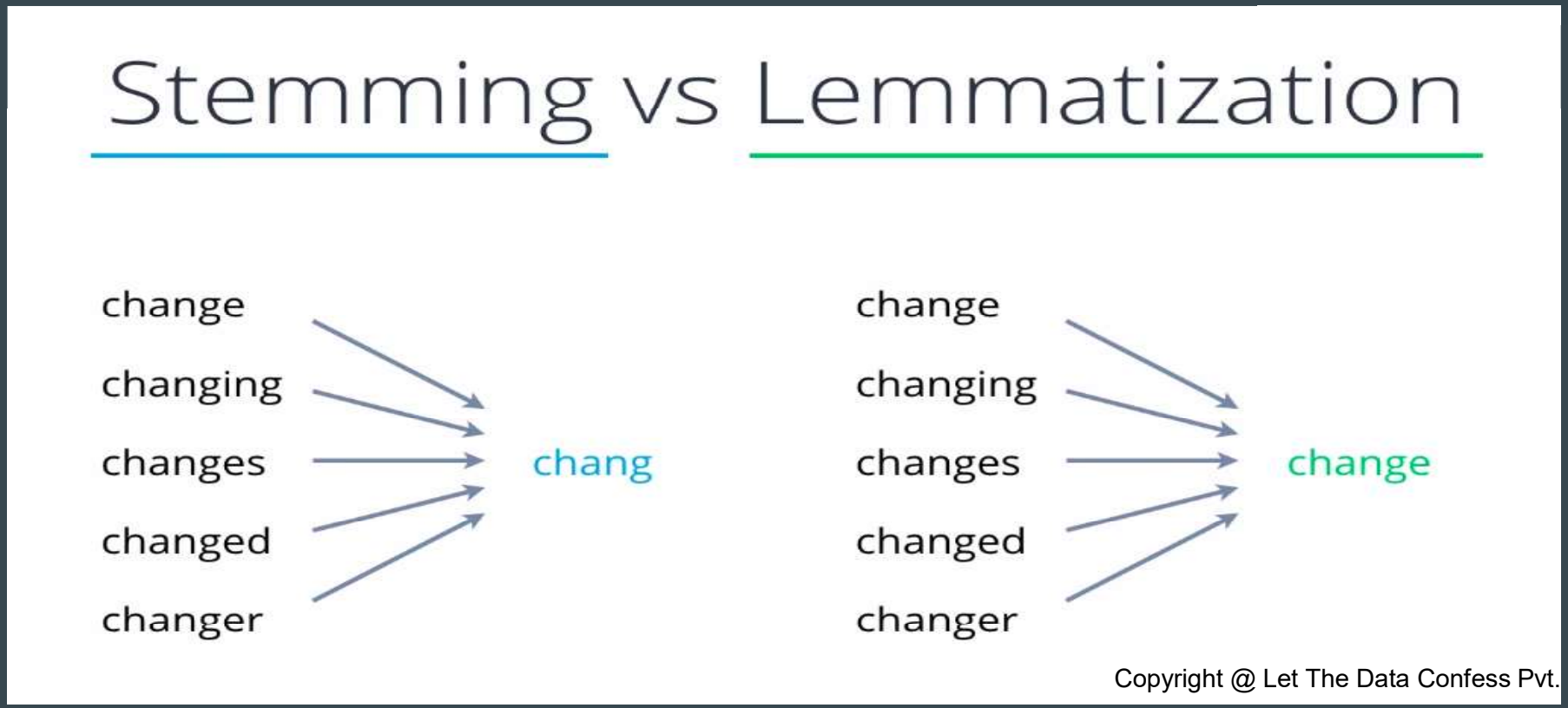
1. Punctuation removal
2. Stop words removal
3. Convert into lower case
4. Contraction removal
5. Spelling correction
6. Special character removal
7. Emoji removal

Step2: Tokenization

1. Bag of words
2. Bag of N-grams



Step3: Feature Selection



Step4: Word Embedding (Feature Extraction)

Learnt representation of words into Vectors

Why do we need it?

How to generate Word embedding

1. Frequency Based
   a. Count based
   b. Tf-Idf Based
2. Prediction based
   a. Word2vec
   b. Glove
   c. Elmo
   d. BERT

Count Based

Document1 = ‘He is a lazy boy. She is also lazy.’

Document2 = ‘Neeraj is a lazy person.’

Dictionary: [‘He’,’She’,’lazy’,’boy’,’Neeraj’,’person’]

He	She	lazy	boy	Neeraj	person
1	1	2	1	0	0
0	0	1	0	1	1

Tf-idf

- term frequency(TF) of word ‘this’ in Document 1 is ⅛
- TF of word ‘This in Document 2 is ⅕

- IDF = log(N/n), where, N is the total number of documents and n is the number of documents a term t has appeared in.

- IDF(This) = log(2/2) = 0.
  If a word appears in each document of given corpus then idf of that word = 0
- IDF(Messi) = log(2/1) = 0.301.

  Combine TF and IDF together:

- TF-IDF(This,Document1) = (1/8) * (0) = 0
- TF-IDF(This, Document2) = (1/5) * (0) = 0
- TF-IDF(Messi, Document1) = (4/8)*0.301 = 0.15

Document 1

Term	Count
This	1
is	1
about	2
Messi	4

Document 2

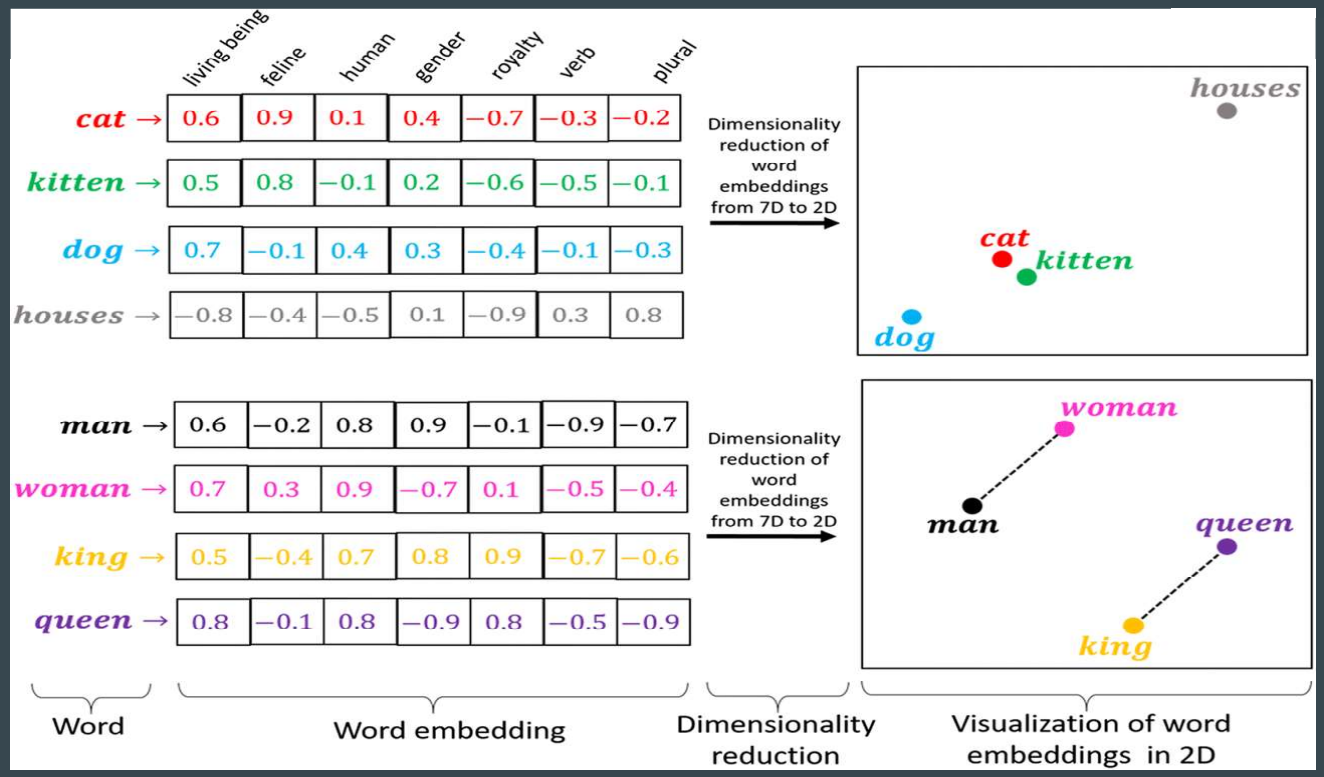
Term	Count
This	1
is	2
about	1
Tf-idf	1

King -man +woman = Queen

How will you solve such problems??

Using Prediction Based Embedding

How does it work?



LET THE DATA CONFESS

Understand | Learn | Code | Implement

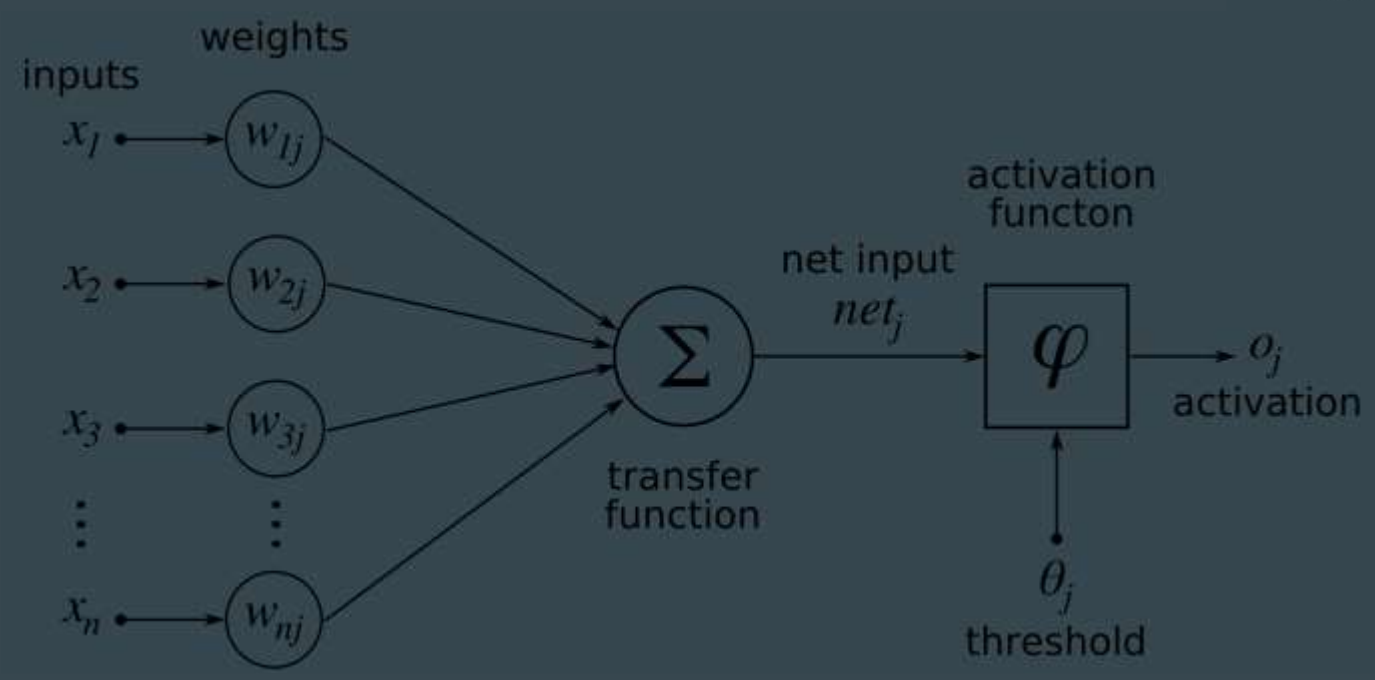
Still is there any problem?

Step 5: Building the model

Questions to be asked?

1. Why Deep Learning?
2. Which model to use?

Artificial Neural Network

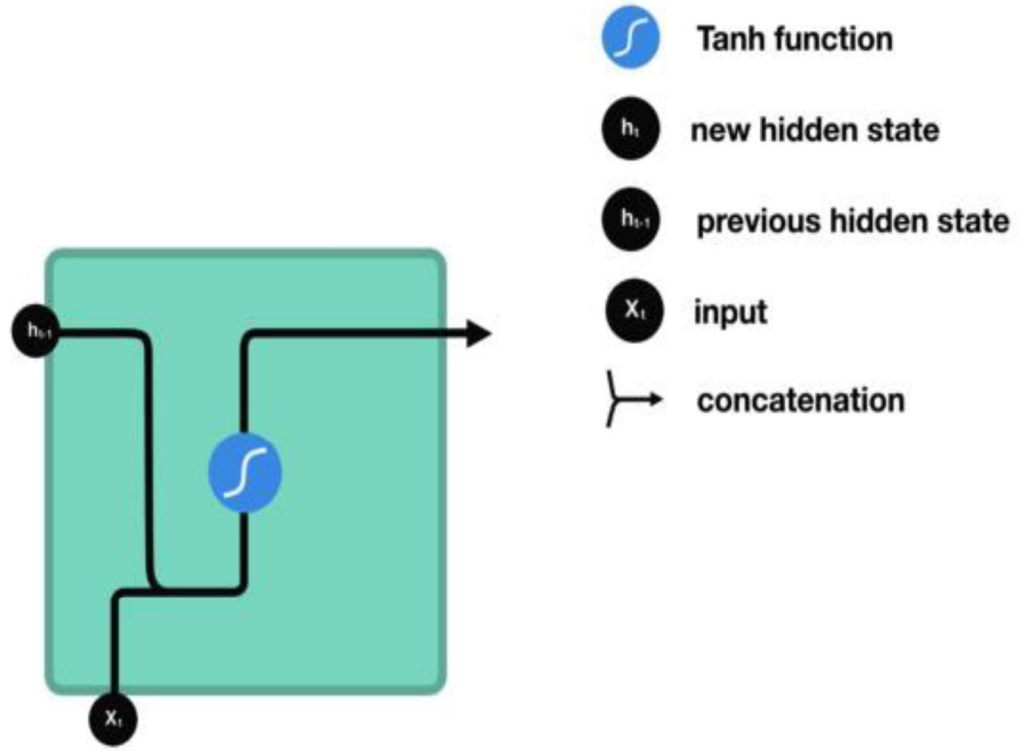


LET THE DATA CONFESS

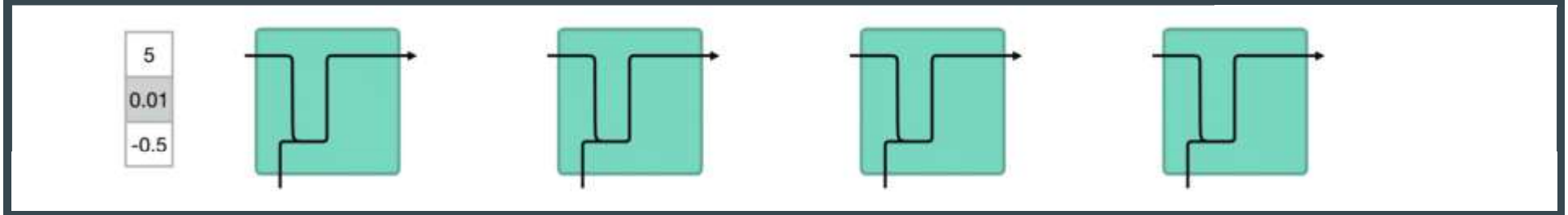
Understand | Learn | Code | Implement

Isn’t it good enough?

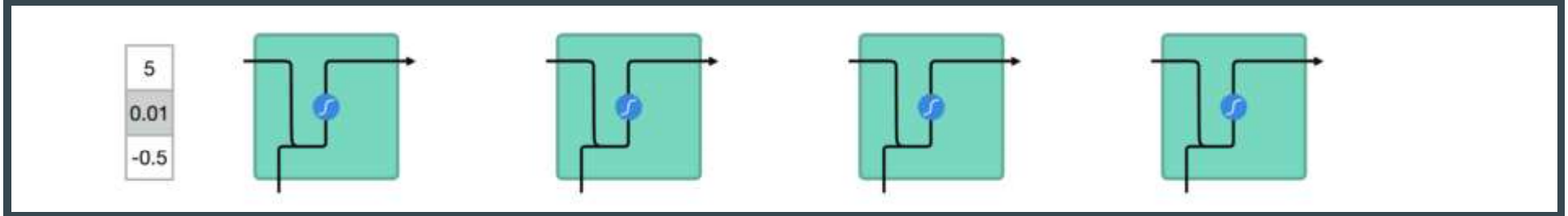
Recurrent Neural Network



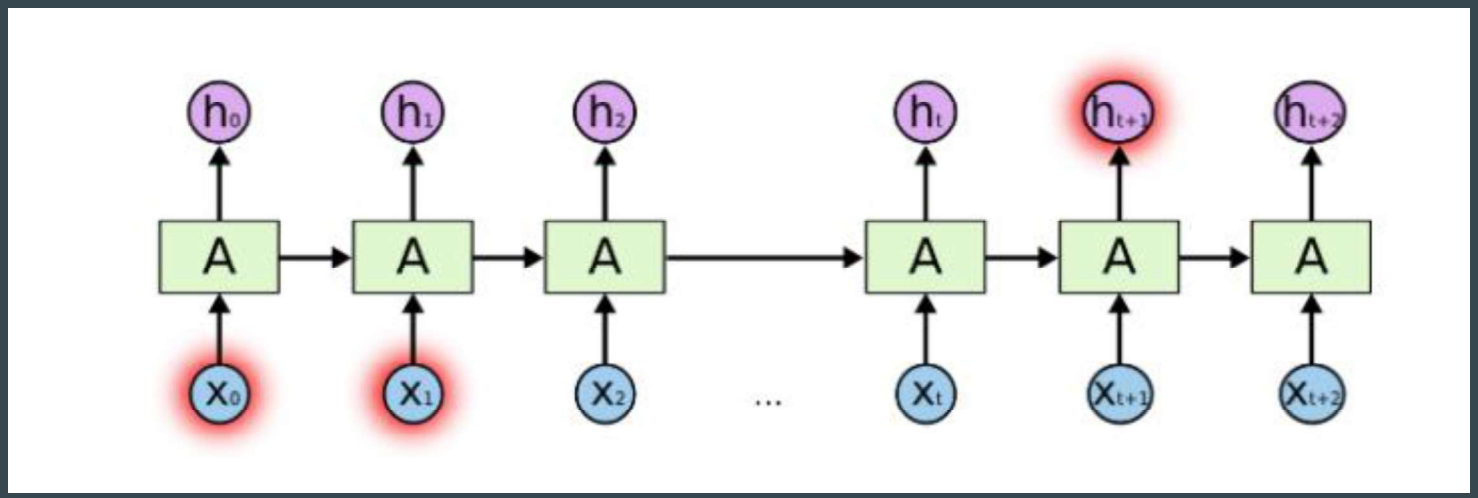
Without tanh function



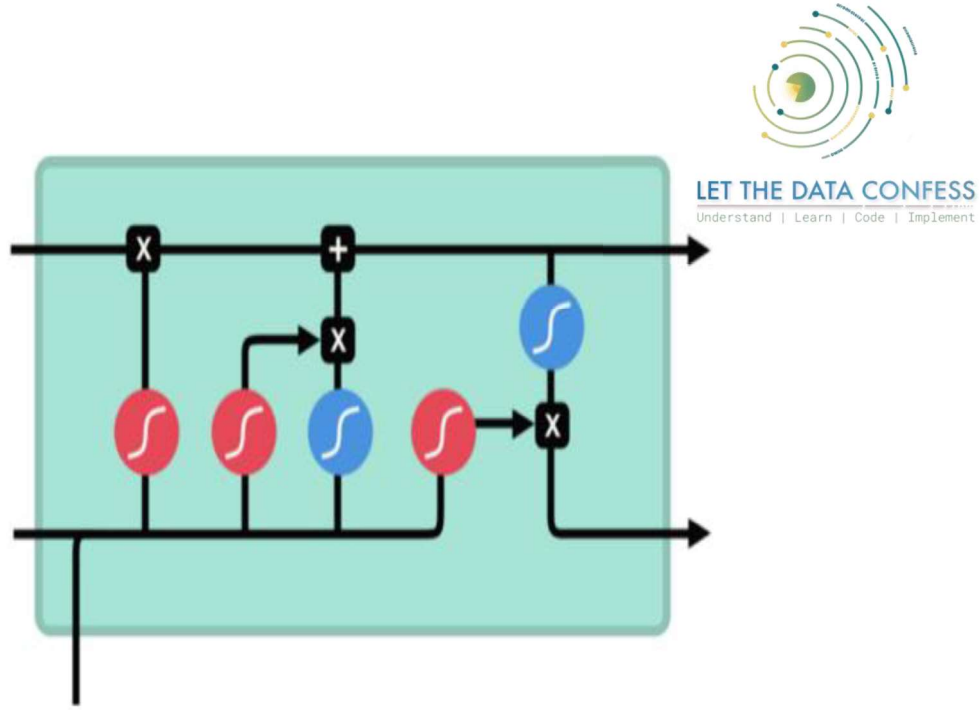
With tanh function



What is the problem now?

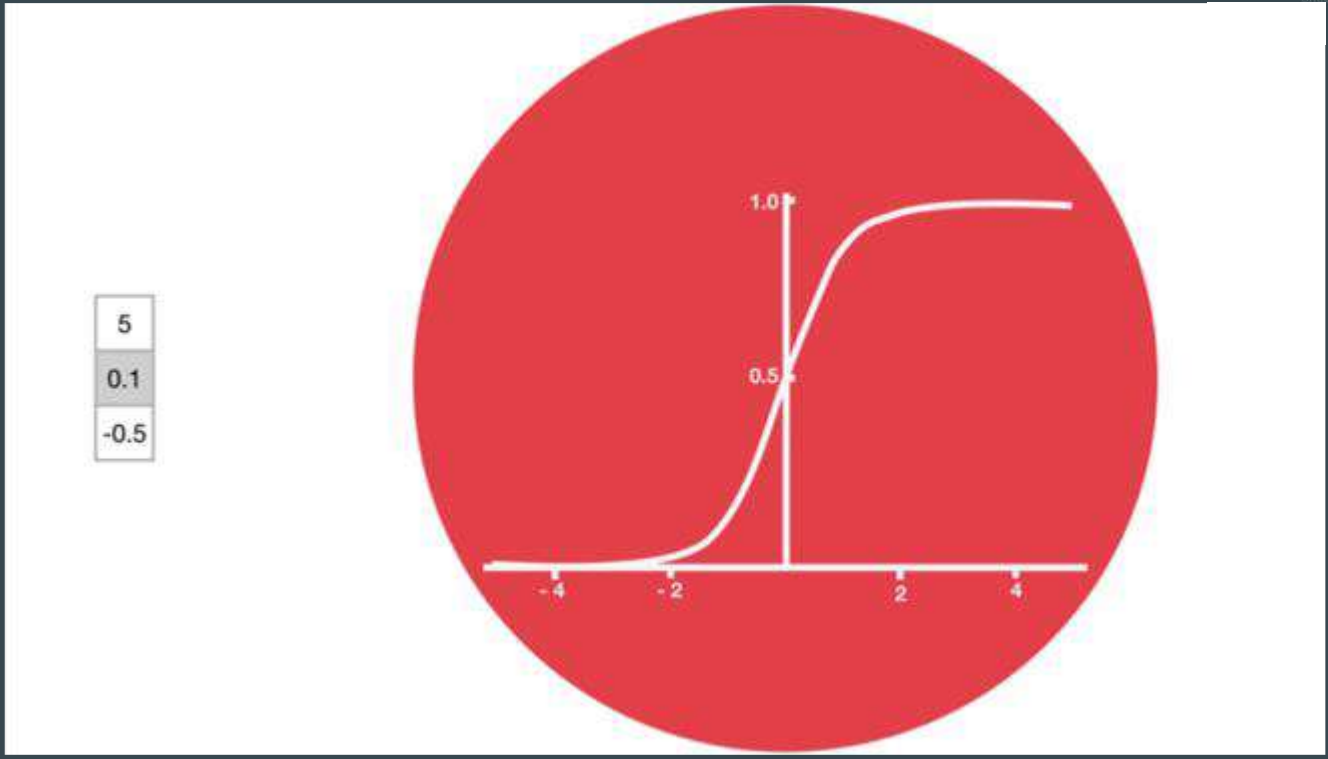


LSTM Cell

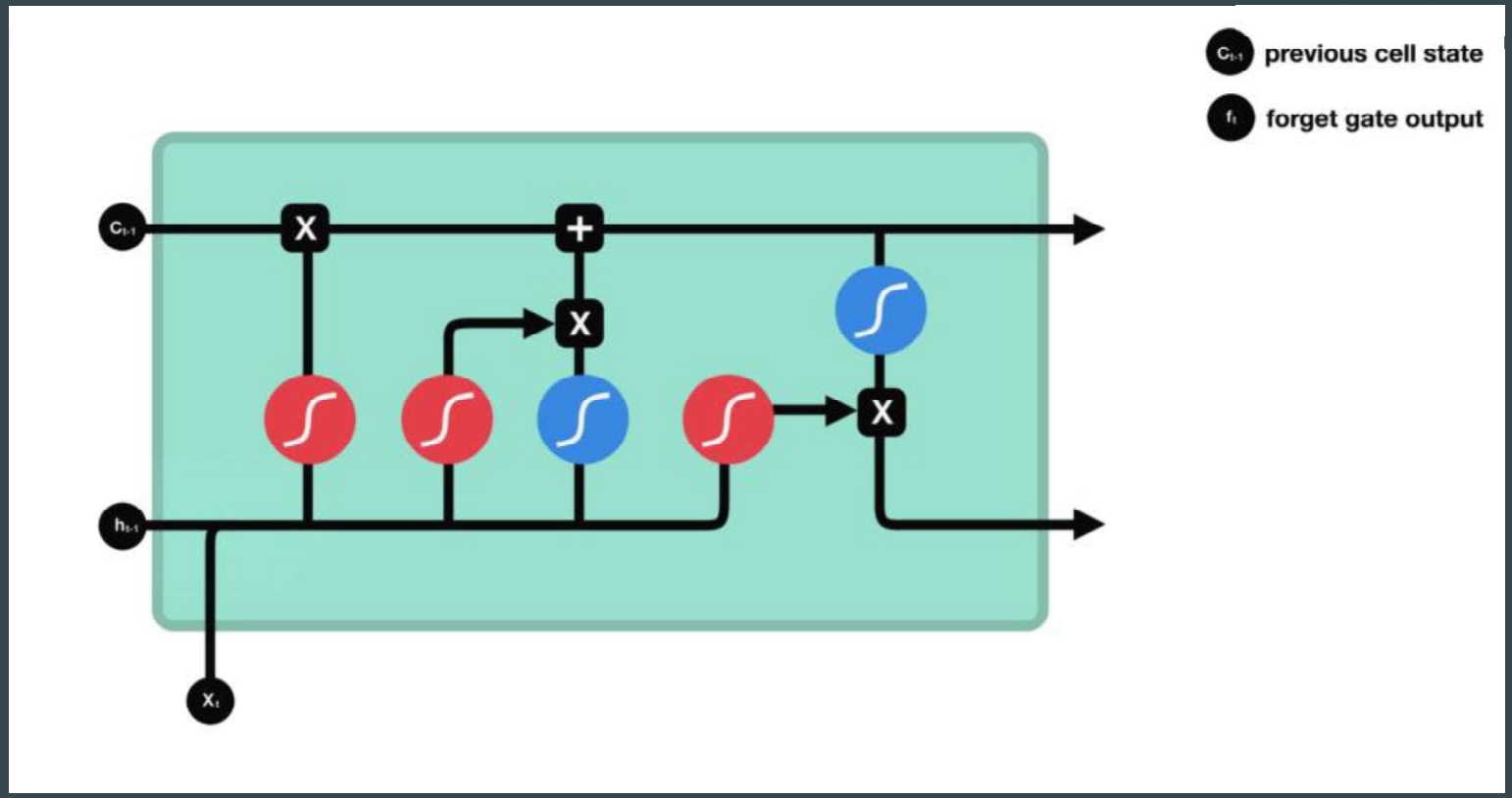

LET THE DATA CONFESS
Understand | Learn | Code | Implement


sigmoid
tanh
pointwise multiplication
pointwise addition
vector concatenation

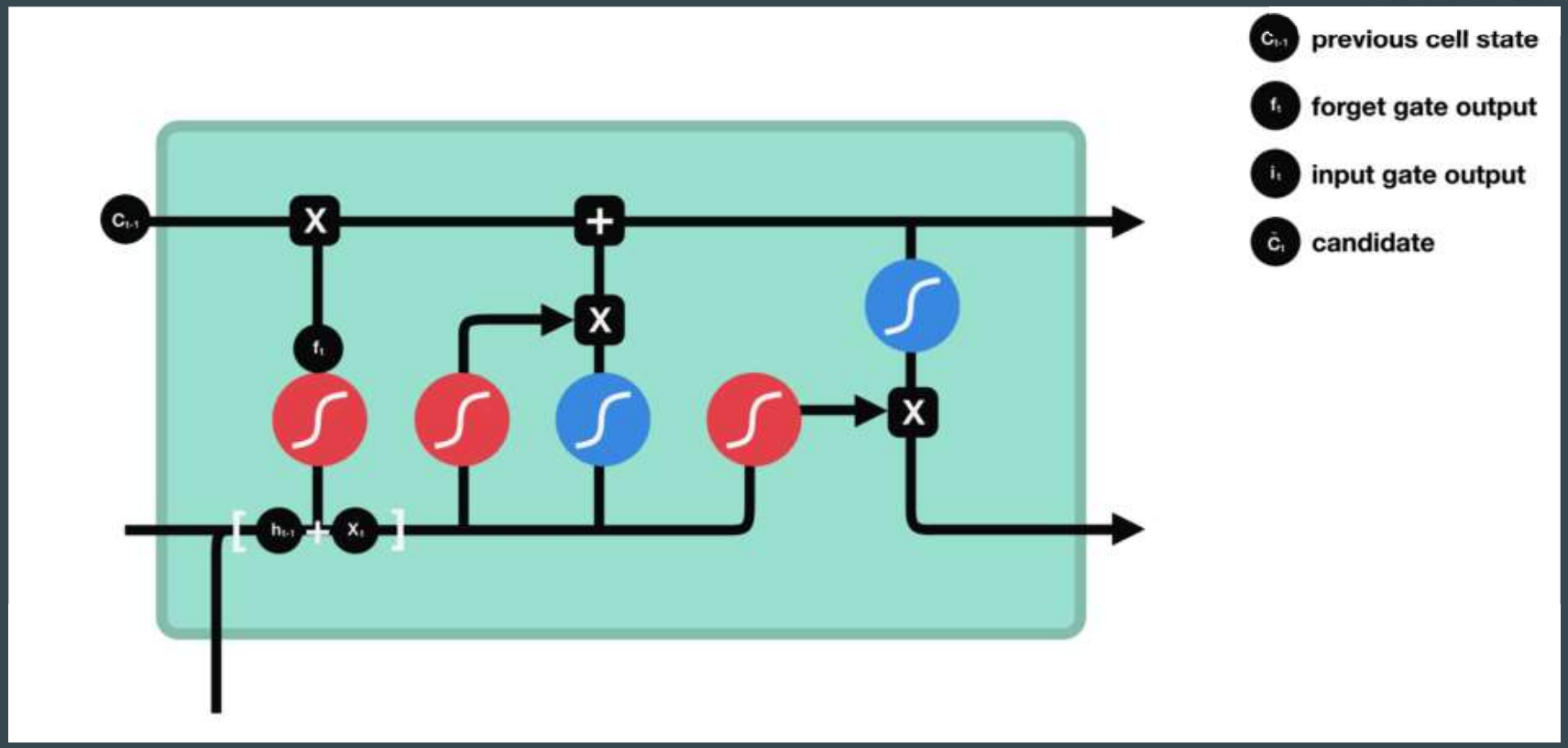
Sigmoid Function



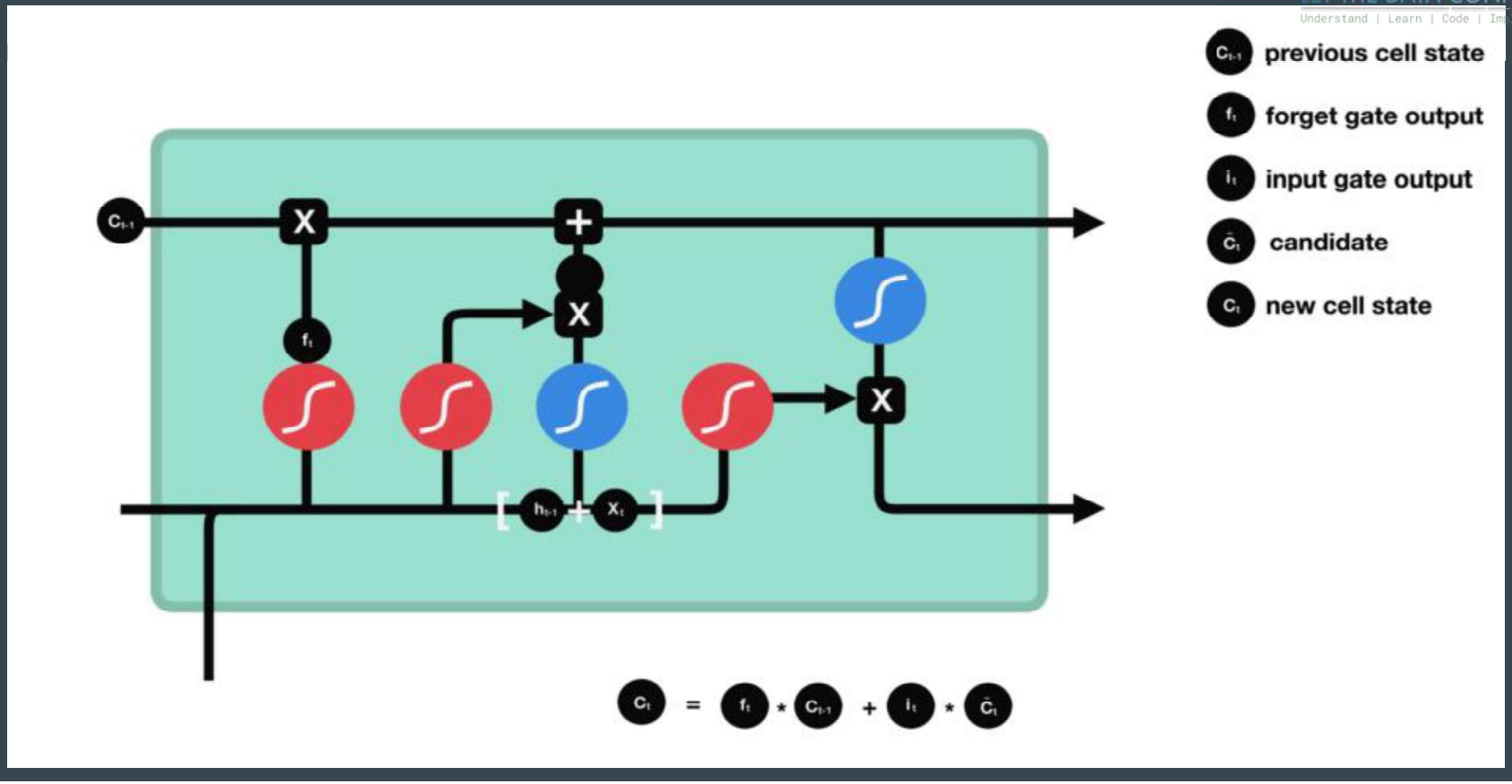
Forget gate



Input Gate


c_{t-1}
f_t
h_{t-1}
x_t
previous cell state
forget gate output
input gate output
candidate

Cell State


C_{t-1}
f_t
$[h_{t-1} + X_t]$
previous cell state
forget gate output
input gate output
candidate
new cell state
$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$