Lecture - 8 NLP topics 14/03/2026.

Model.

'This is a cat.'

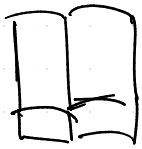
vocabulary.

Basic representation : 'T', 'h', 'i', 's' ... → characters.

Word. 'This', 'is', 'a', 'cat'.

↓

numbers.



word, Sentence, paragraph, Chapters. → book.

DNN → numbers.

{0 - 26}. few punctuation etc.

lower case.

256

Character representation →

Memory consuming → in case of # of cycles required for parsing a sentence.

Word : If we encounter a new type of word in the test sentence, then (if) model will be unable to predict that word.

Vocabulary → create.

- Byte pair encoding
- Tokenization.

} Balance tokenizer. } current models.

- Sparse representation → One hot vectors.

Let's consider for now

atomic element of a sentence is 'word'.

{ 'kilometers', 'kilograms' }

↓

$\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$

nx1 vector.

each of the word are unrelated to each other.

Hence we can have different meanings for each of the individual words.

'queen', 'lady', 'girl' → not completely unrelated.

'king', 'man', 'boy' → not completely unrelated text.

$\begin{pmatrix}1\\0\\0\end{pmatrix}$ $\begin{pmatrix}0\\1\\0\end{pmatrix}$ $\begin{pmatrix}0\\0\\1\end{pmatrix}$.

↳ which is not practical / useful.

- Since we want similar words to have similar representations.

Hence cosine similarity should be higher for similar words.

Special kinds of tokens. :-

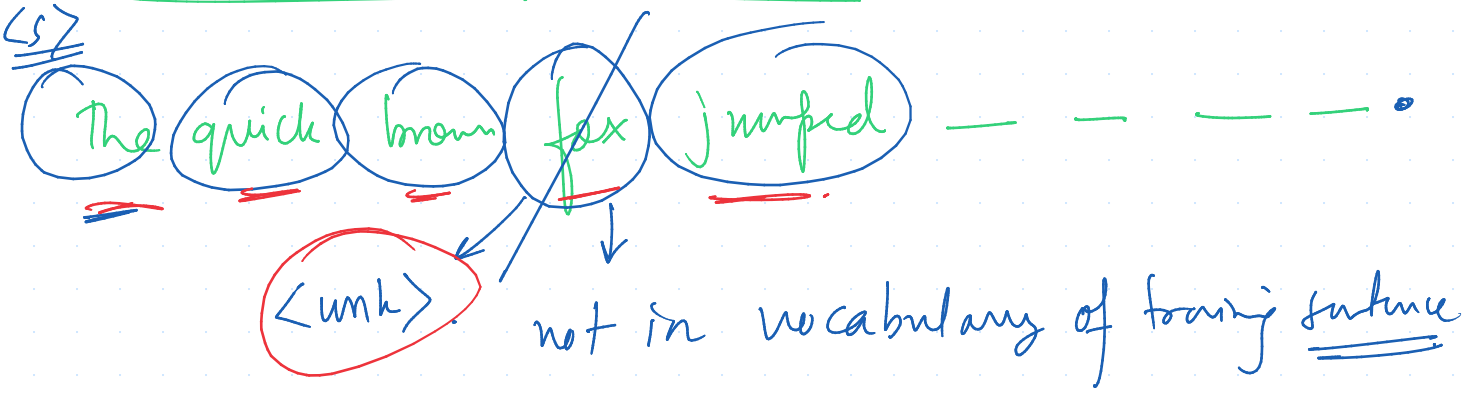
<s> , <bos>. → start of the sentence.

</s> , <eos> → end of sentence.

<unk> → unknown. : Most of the cases, we will see that a new kind of token is encountered during inference.

<think> </think> → reasoning of ChatGPT.

Sentence completion task.



Task:

Learn meaningful representations for word.

Learn word embeddings. (Token embeddings)

2013

Tomas Mikolov. Jeff Dean.

unsupervised algo → Google news dataset.

automatically learns meaningful representations.

No annotation required from human side.

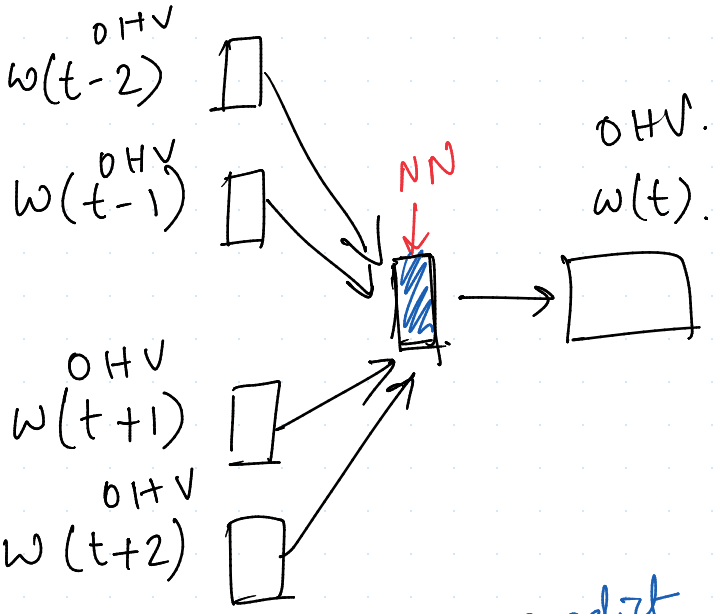
'A word is known by the company it keeps'

Similar words will be closer to each other.

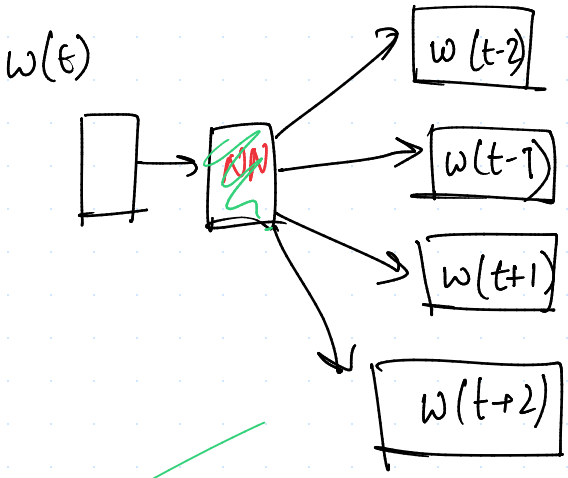
Word2Vec Models.

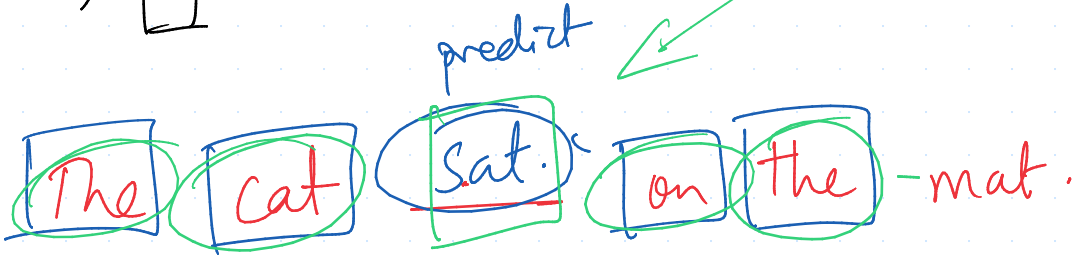
CBOW

Continuous Bag of Words.



Skip-gram Models.





sliding window.

CBOW → predicts a single target based on its surrounding context words.

Skip-gram

Predicts the surrounding context words given a single target word.

Vocabulary → (1, 0, 0, ... 0) One hot rep.

Categorical Cross Entropy loss → Training.

Pytorch ; OHV ⇒ Sparse

embedding vector.

function.

10,000

(0 1 0 0 ... 0)
(0 0 1 ... 0)
(0 ... ...)

10,000

Sparse rep → dense.

vocab size → V >> D → dense representation

OHV ×.

300 → D-dim-vector.

(2.3, 0.7, 0.6, 1.3, ...)

300

$\sqrt{(2.3)^2 + (0.7)^2 + (0.6)^2 + (1.3)^2 + \dots}$

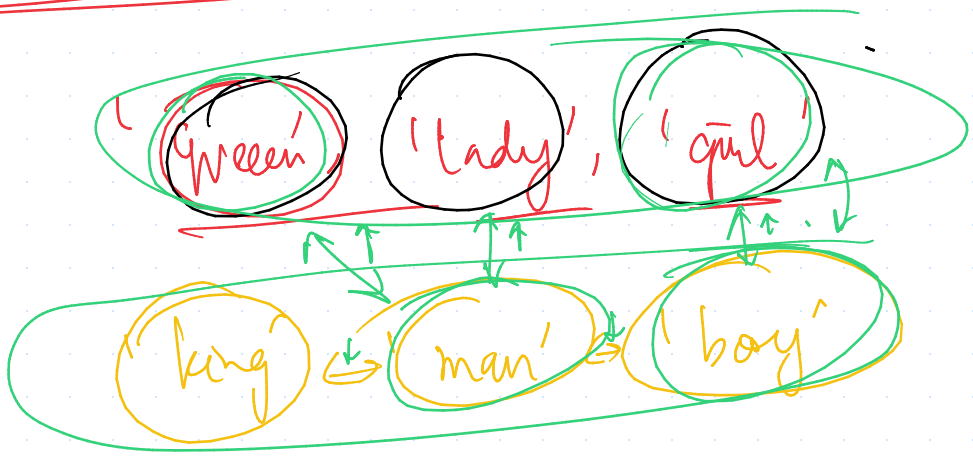
cosine similarity b/w any two words = 0, if it is sparse.

not sparse

doesn't have much info.

helpful when computing the cosine similarity b/w word vectors.

Dense representation



meaningful representations are learnt.

'boy' → tokenization → 20 → Embedding → (. . . . .) → Model.

Dense representation

(token . . .)

word predict

dot-product b/w normalized vectors, then it already calculates cosine similarity

(1, 0, 0) king ⟷ (0, 1, 0) man (0, 0, 1) boy

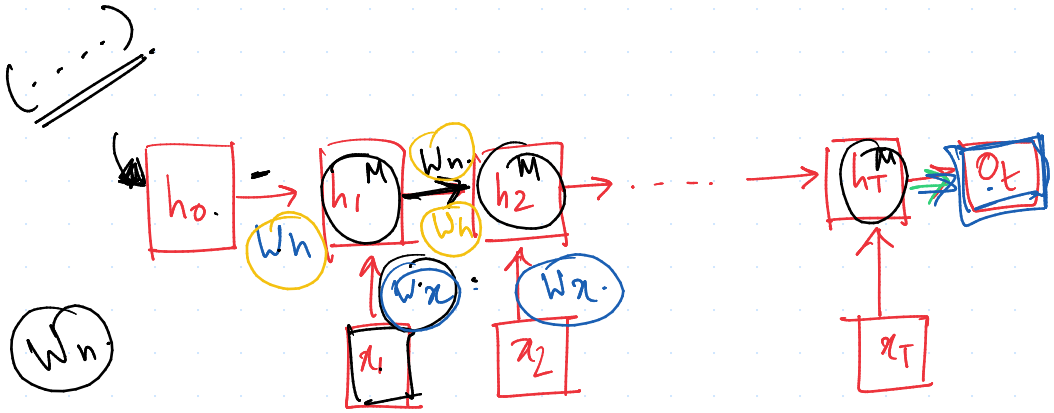
1·0 + 0·1 + 0·0 = 0

(0.1, 0.9) (0.9, 0.1) (0.3, 0.6)

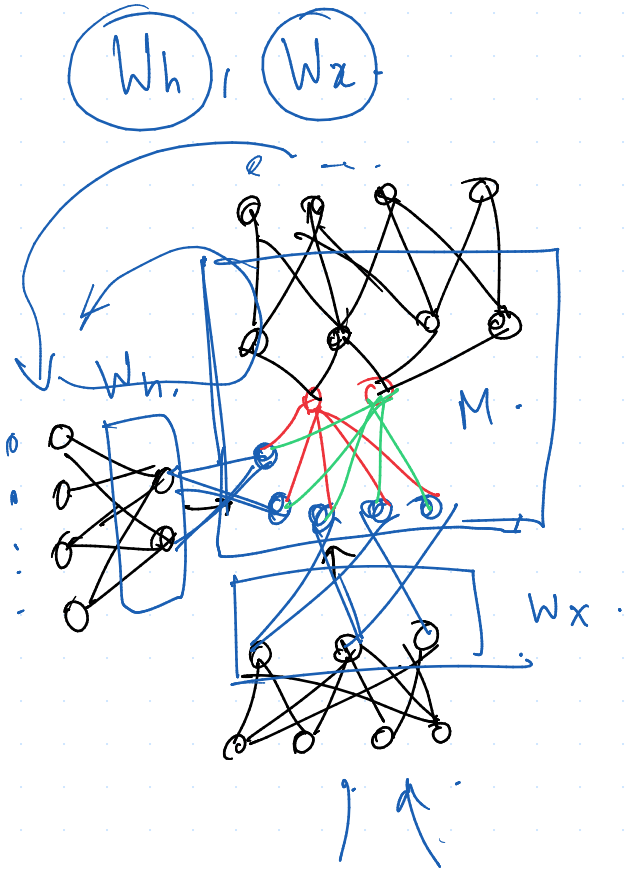
0.1 × 0.2 + 0.9 × 0.7

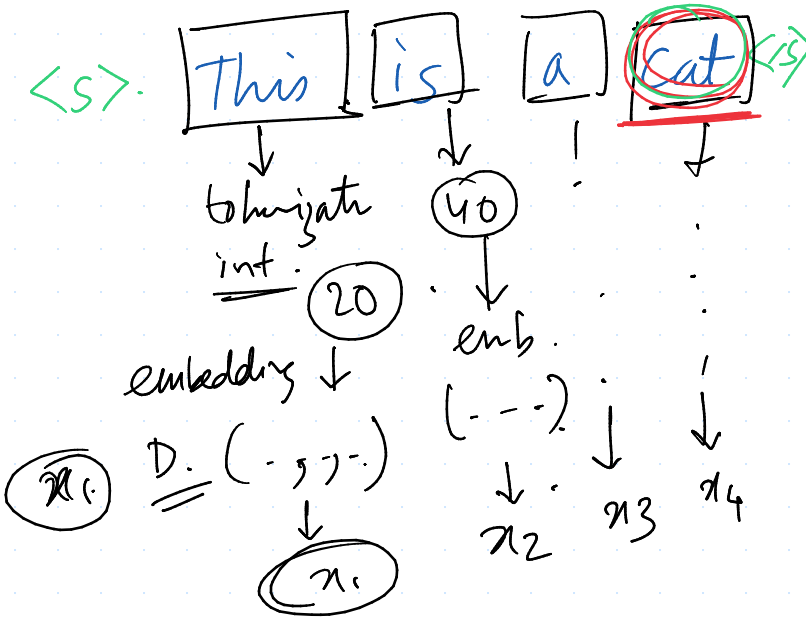
0.1 × 0.9 + 0.9 × 0.1 ≠ 0

Recurrent Neural Networks. (RNNs)



Not considering o/p for the time being.





$x_1, x_2, \cdots, x_T$ → sequence of word vectors.

g → non-linear function.

$h_i = g(W_h h_{i-1} + W_x x_i)$.

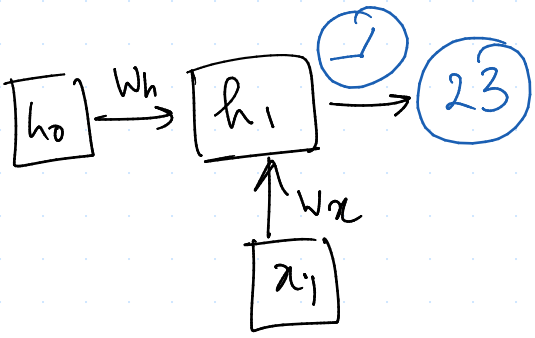
$y = \text{argmax}(\text{Softmax}(g(\circ)))$.

<s> hello </s>.

$W_x = [1 \quad 5 \quad 3]$. $W_h = [7]$.

$h_0 = [1]$.

$x_1 \rightarrow [3 \quad 2 \quad 1]$.



$\text{ReLU}(W_h \cdot h_0 + W_x \text{ embd}(\text{<s>}))$.

$\text{ReLU}([7] \times [1] + [1 \quad 5 \quad 3] \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix})$

$\Rightarrow \text{ReLU}(7 + 3 + 10 + 3)$.

$\Rightarrow \text{ReLU}(23) \Rightarrow 23$.

Shortcomings of RNNs.

- Vanishing & exploding gradients.
- History gets forgotten at time.

Temporary solution

- Gradient clipping

Normalize the gradient of a parameter vector when its L2 norm reaches a certain value.