It was six men of Indostan
To learning much inclined,
Who went to see the Elephant
(Though all of them were blind),
That each by observation
Might satisfy his mind.

J.G. Saxe

All images from Wikimedia Commons unless specified.

UNSW SYDNEY

1

Natural Language Processing (NLP)

COMP6713 – 2025 Term 1

Convener
Dr. Aditya Joshi
aditya.joshi@unsw.edu.au

Week 2
Representation learning

Schedule
2025 Term 1

UNSW SYDNEY

2

Announcements

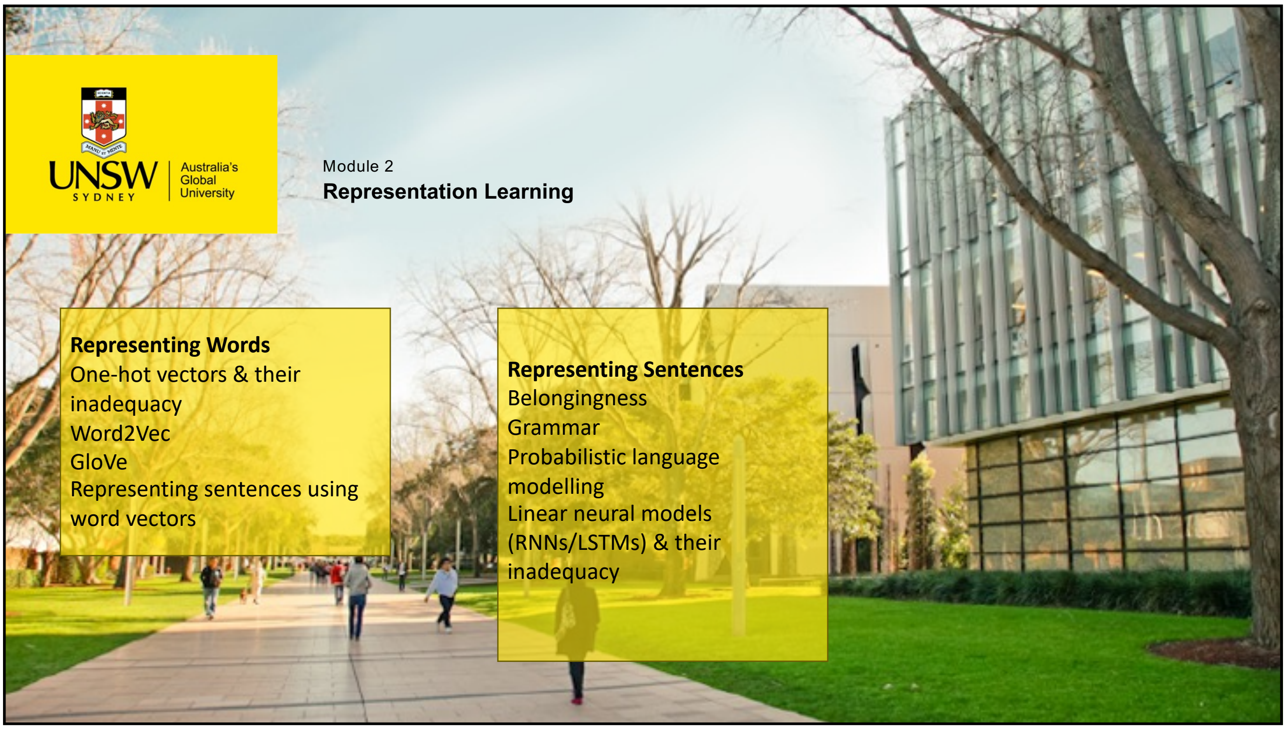
First quiz:

- Moodle
- Opens on Friday at 5:25pm
- Available till next Wednesday, 5:25pm
- One attempt only: Once you submit, you cannot change your answers.

Tutorials start this week. Do attend! You will complete the assignment faster, if you attend the tutorial.

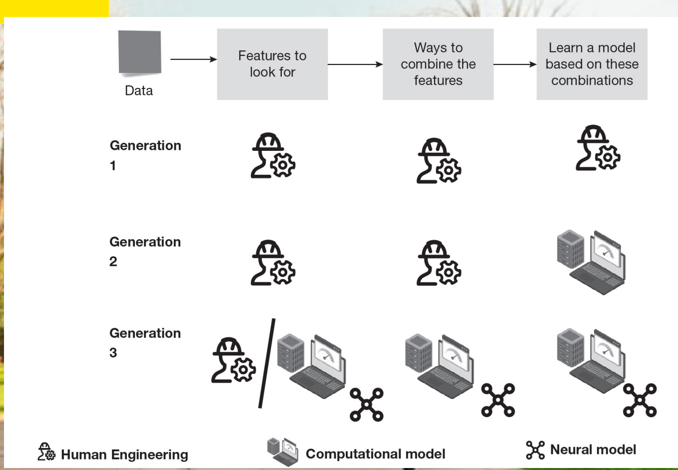
Consultation: Thursdays 10-11. Online or in-person (217B in Building K17). Best to drop an email.

3



4

Topics this week



Computational grammar

One-hot vectors, probabilistic language modeling

word2vec, GloVe, LSTM-based language modeling

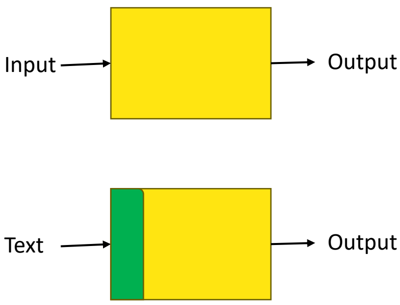
5

Part 0

Representation Matters!

6

Let’s unpack the NLP black-box



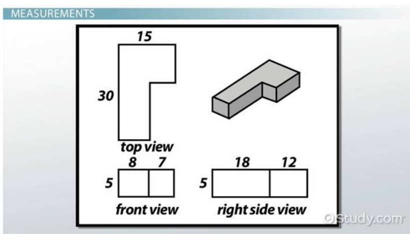
“dog” → ?

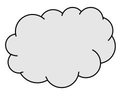
“the dog is cuddly” → ?

7

Unpacking the intuition behind text representations

“the description or portrayal of someone or something in a particular way.”





There are multiple incomplete ways to represent an entity

A representation is an abstraction that captures the essence of the idea.

Representations describe entities (things/people/concepts..).

Source: (Left) Wikimedia & (right) a picture from my window

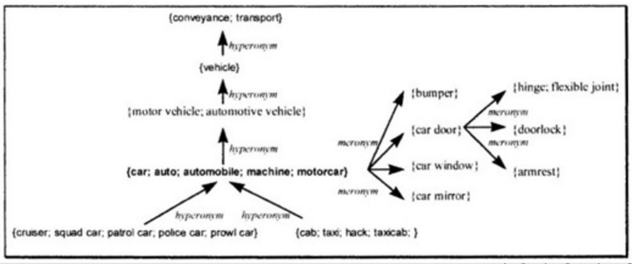
8

Representing concepts as words

Entities in the real world

Representations of entities as English words

dog
cat
chair
table



https://www.cairn.info/revue-francaise-de-linguistique-appliquee-2002-1-page-27.htm UNSW SYDNEY

Representation via combination of words

Meaning of a Word --> Meaning of a sentence

table 

dog 

The dog is on the table.

What about this emoji?

An emoticon is a pictorial **representation** of a facial expression using characters to express a person's feelings, mood or reaction, or as a time-saving method.

Representations describe something to communicate an idea.

Phare, Darin M., Ning Gu, and Michael Ostwald. "Representation in design communication: meaning-making in a collective context." *Frontiers in Built Environment* 4 (2018): 36.

UNSW SYDNEY

How do we communicate text to machines?

A word itself is a representation of an idea/entity

Representations in NLP: Converting text to a form that can be understood by and useful for a machine learning algorithm

Dog table brown *word*

The dog is on the table. *sentence*

The dog is on the table. He is cuddly. *discourse*

11

Describing concepts as words.. to a ML algorithm

Representations of entities as English words

c1: dog
c2: cat
c3: chair
c4: table

Representation of words as vectors

#columns = |Vocabulary|

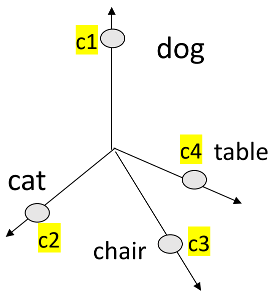
	dog	cat	chair	table
C1	1	0	0	0
C2	0	1	0	0
C3	0	0	1	0
C4	0	0	0	1

Machine Learning Algorithm (e.g. Logistic Regression/SVM)

Each word becomes a random variable. The goal of the learning algorithm is to learn to predict the output variable given values of input variables (words).

12

ONE-HOT VECTORS (A.K.A. UNIGRAM VECTORS)



Representation of words as vectors

	dog	cat	chair	table
c1	1	0	0	0
c2	0	1	0	0
c3	0	0	1	0
c4	0	0	0	1

These are known as **one-hot vectors.**

The ruling paradigm in statistical NLP.

How would we represent the concept below?

What information is lost? When would it be reasonable to lose this information?

13

What is hot about one-hot?

"One-hot": Electronics engineering: Hot represents that there is a current at the specific terminal

Decimal	Binary	One-hot representation
0	00	1000
1	01	0100
2	10	0010
3	11	0001

Word	One-hot representation
dog	1000
cat	0100
chair	0010
table	0001

Questions to ponder: (a) Does it mean there are one-cold vectors too?, (b) Why use one-hot representations for words, why not binary? Wouldn't they require lower number of bits aka vectors of shorter lengths?

14

Scikit-learn

Popular data science library

Also provides vectorization tools for NLP

You will use it when you train statistical classifiers for sentiment analysis (week 6)

Demo time!

15

Scikit-learn: Recap

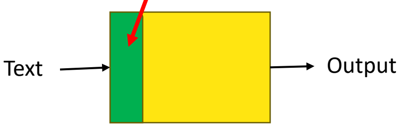
Popular data science library

Also provides vectorization tools for NLP

- Count vectorizer (similar to one-hot vector)
- TF-IDF vectorizer (TF: Term Frequency; IDF: Inverse-document frequency)

What can vectorization be used for?

- It essentially represents the input text in a numeric format.
- Can be used for classification tasks (using classical ML algorithms such as logistic regression, Naïve Bayes, and so on)
- We will see how it can be used for sentiment analysis in Week 6



16

Issues with one-hot vectors: Similarity

Word	One-hot representation
dog	1000
cat	0100
chair	0010
table	0001

Difference between 'dog' and 'cat': 2 bits

Difference between 'dog' and 'chair': 2 bits

Difference between 'chair' and 'table': 2 bits

Are the words really equally dissimilar?

Word	One-hot representation
trousers	1000
pant	0100
shirt	0010
boot	0001

Synonyms are different variables. 'Trousers' and 'pant' can be considered synonyms - but still represent different terms.

What impact does it have on the machine learning algorithm?

17

Let's add dimensions to the representation

Think of column names here that will allow a better representation than one-hot

Word						
dog						
cat						
chair						
Table						
trousers						
pant						
shirt						
boot						

18

Let's add dimensions to the representation

Word	Animal	Bird	Furniture	Garment	Bottom-garment	Top-garment	Can it be put on a table?
dog							
cat							
chair							
Table							
trousers							
pant							
shirt							
boot							

19

Let's add dimensions to the representation

Word	Animal	Bird	Furniture	Garment	Bottom-garment	Top-garment	***Can it be placed on a table?***
dog	1	0	0	0	0	0	1
cat	1	0	0	0	0	0	1
chair	0	0	1	0	0	0	1
Table	0	0	1	0	0	0	1
trousers	0	0	0	1	1	0	1
pant	0	0	0	1	1	0	1
shirt	0	0	0	1	0	1	1
boot	0	0	0	1	***0.5***	0	1

Hand-crafted dimensions are neither complete nor accurate.

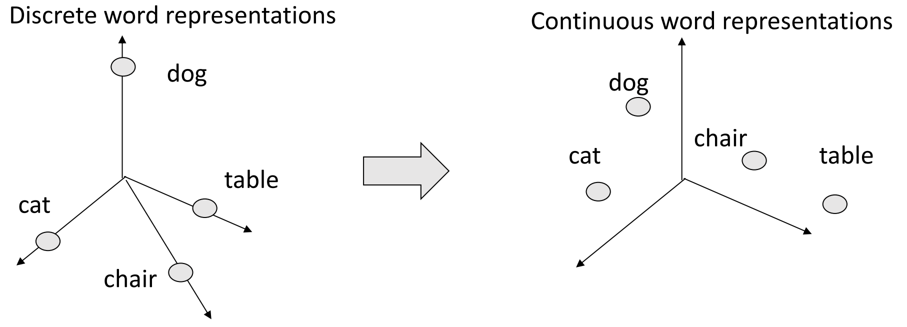
20

Therefore…. Continuous representations

Word	D1	D2	D3	…	…	..	***DN***
dog	.5	.3	.6	.6			.3
cat	.7	.8	.5	.3			.9
chair	.43	.63	.37	.82			.25
Table	0.11	0.24	0.4	0.4	0.2	0.35	0.1
trousers	0.6	0.24	.7	.8	0.74	0.24	0.4
pant	0.53	0.6	.43	.63	.9	.43	0.54
shirt	.5	.3	.43	0.64	.25	0.11	0.54
boot	.37	.82	0.11	0.5	0.32	0.3	0.1

21

Visualisation of continuous representations



Word embeddings are continuous vectors that represent the semantics of a word in a k-dimensional space.

22

Learning representations of words

Don't map words to pre-defined vector hashmaps

"Learn" representations of their meanings

Representation of words

Word vectors; word embeddings; word representations; semantic vector space models

https://projector.tensorflow.org/ ←Click here.

23

How can vectors for words help?

Similarity (dog, cat) = 0.76

Similarity (dog, chair) = 0.08

Similarity(chair, table) = 0.3

What is this similarity?

$$similarity(A,B) = \frac{A \cdot B}{\|A\| \times \|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}}$$

24

Semantic vector representations of words

Semantic vector space models of language represent each word with a real-valued vector (Pennington et al, 2014)

Can words be represented as dense, real-valued vectors that capture the meaning of these words?

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. .
Goldberg, Yoav, and Omer Levy. "word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method." *arXiv preprint arXiv:1402.3722* (2014).
Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. 2013b. Distributed representations of words and phrases and their compositionality. *NIPS* 3111–3119.

25

Word meaning in terms of distributional similarity

Similar words have overlapping words in their corresponding context.

… I have a pet dog named Tommy …
… I have a pet cat named Lucy …
.. I took my dog to the park ..
.. I took my cat to the park ..

.. I took my chair to the park..?

For example:

I went to the **bank** to withdraw money.

I went to the **bank** to catch fish.

A word is known by the company it keeps.

26

What is distributional about the similarity?

A word is represented in terms of the distribution of words **around** it

It's not the size of the **dog** in the fight, it's the size of the fight in the **dog**.
If you pick up a starving **dog** and make him prosperous he will not bite you. This is the principal difference between a **dog** and man.
The more I learn about people, the more I like my **dog**.
(Quotes by Mark Twain)

What are we trying to learn?

Representation learning algorithm

Word	Vector
dog	[0.8, 0.9 ..]
cat	[0.2,-1.4,...]

UNSW SYDNEY

27

UNSW SYDNEY Australia's Global University

Part 1

word2vec & GloVe

Suggested Reading:
Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of NIPS, 2013.

Pennington, J., Socher, R. and Manning, C.D., 2014, October. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

28

Algorithms to learn word representations

Word2Vec:

Linguistics: Paradigmatic Similarity

"Similar words are substitutable" (e.g. dog and cat)

GloVe:

Linguistics: Syntagmatic Similarity

"Similar words frequently co-occur" (e.g. dog and bone)

29

Word2Vec

Word2Vec

Word	Vector
dog	[0.8, 0.9 ..]
cat	[0.2,-1.4,...]

Convert words to vectors

Limitations: Does not capture multiple meanings of a word

Where do we get 'labels' from?

Self-supervised task: "Fill in the gaps" over spans of words

*i see two boats with nets, lying off the shore of paumanok, quite still**

Span = 3

Skip-gram^

Input	Output
see	I, two
two	see, boats
boats	two, with

OR

Input	Output
I, two	see
see, boats	two
two, with	boats

Continuous bag of words

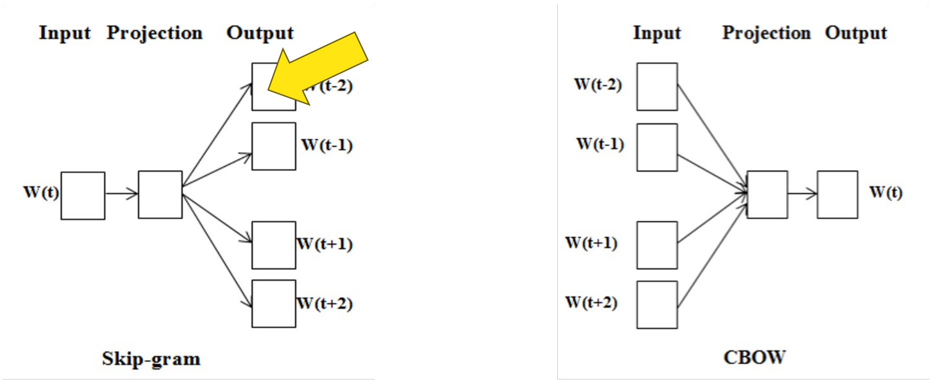
^Remember n-grams?

* Poem by Walt Whitman

Suggested Reading: https://d2l.ai/chapter_natural-language-processing-pretraining/word2vec.html

30

… and as a diagram



Suggested Reading: https://d2l.ai/chapter_natural-language-processing-pretraining/word2vec.html

UNSW SYDNEY

31

Softmax: Every NLP person's trusted friend

Soft Max

Allows mapping vector similarity to a normalized score

Has interesting differentiable properties

Common in neural NLP

Indicates a choice: 'ambiguity resolution'

For a vector z of K real numbers, the standard (unit) softmax function $\sigma : \mathbb{R}^K \mapsto (0,1)^K$, where $K \geq 1$, is defined by the formula

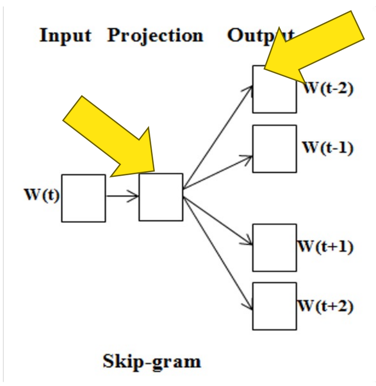
$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \quad \text{for } i = 1, \ldots, K \text{ and } \mathbf{z} = (z_1, \ldots, z_K) \in \mathbb{R}^K.$$

Image: MS Stock Photos

UNSW SYDNEY

32

A Tale of Two Vectors



softmax helps to obtain prediction as a word. **How?**

$$P(w_o \mid w_c) = \frac{\exp(\mathbf{u}_o^\top \mathbf{v}_c)}{\sum_{i\in\mathcal{V}} \exp(\mathbf{u}_i^\top \mathbf{v}_c)}$$

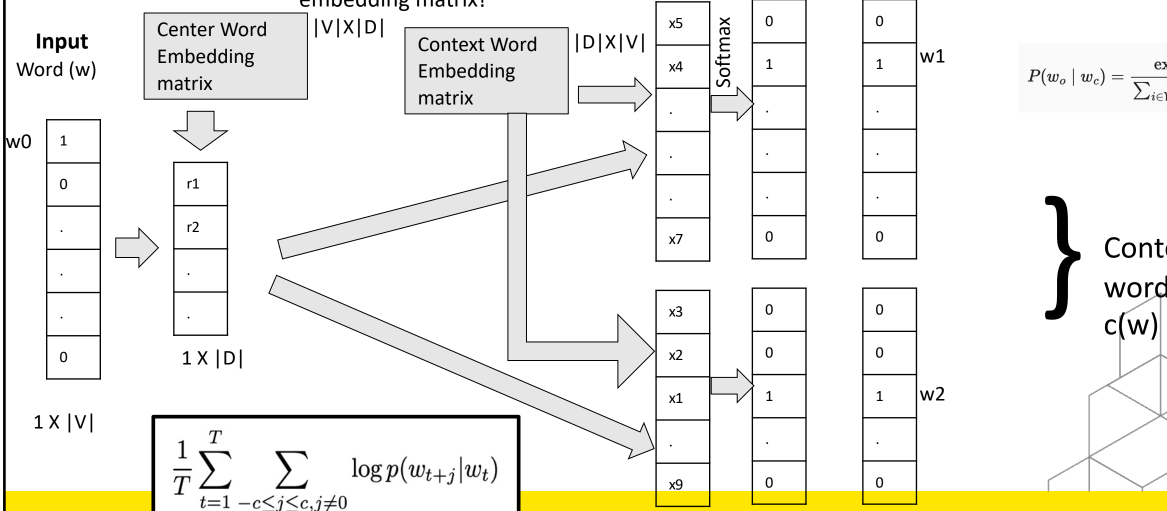
Word	Center Vector	Context Vector
dog	[0.4, 0.9 ..]	[0.8, 0.9 ..]
cat	[0.45, 0.18..]	[0.2,-1.4,...]
.....		

Note: One-hot Vector → Two Vectors (of which one will be used as a word embedding) -→ Next week: THREE vectors

33

Skip-gram: Architectural view

At the end of learning this model, the word embedding matrix becomes a useful embedding matrix!

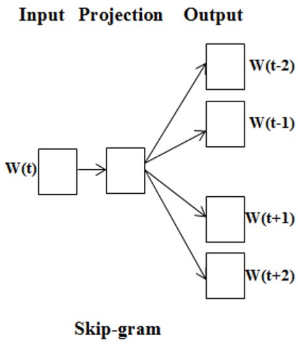


$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le j\le c, j\neq 0} \log p(w_{t+j}|w_t)$$

$$P(w_o \mid w_c) = \frac{\exp(\mathbf{u}_o^\top \mathbf{v}_c)}{\sum_{i\in\mathcal{V}} \exp(\mathbf{u}_i^\top \mathbf{v}_c)}$$

34

Mathematical formulation of skip-gram



$$P(w_o \mid w_c) = \frac{\exp(\mathbf{u}_o^\top \mathbf{v}_c)}{\sum_{i \in \mathcal{V}} \exp(\mathbf{u}_i^\top \mathbf{v}_c)}$$

Pen-and-paper time!

Suggested Reading: https://d2l.ai/chapter_natural-language-processing-pretraining/word2vec.html

35

Word embedding matrix

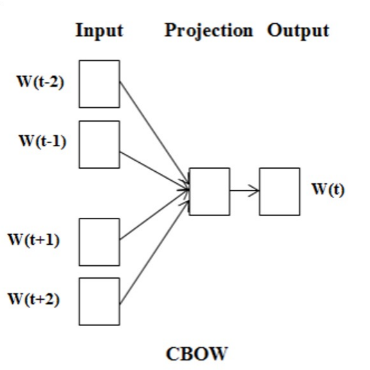
Word Embedding matrix

Word2Vec is our first application of self-supervision

Unlabeled data -> Modified to set up a context prediction task -> Learned word representations

36

What about CBOW?



Exercise: Adapt Skip-gram architecture to CBOW. Draw a corresponding diagram.

37

Staring at the softmax

$$P(w_o \mid w_c) = \frac{\exp(\mathbf{u}_o^\top \mathbf{v}_c)}{\sum_{i \in \mathcal{V}} \exp(\mathbf{u}_i^\top \mathbf{v}_c)}$$


Soft Max

What the V?

What are the implications?

The value of V could be too large making computation difficult.
Two alternatives to address this problem

38

Methods to optimise (1/2): Hierarchical Softmax

Softmax can be computed as a hierarchy of nodes in a tree

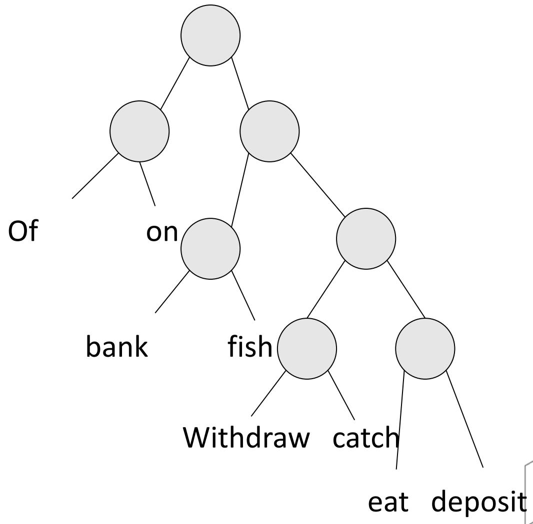
What kind of tree? Huffman tree

1)Leaf nodes are words

2)Words that are frequent are closer to the root

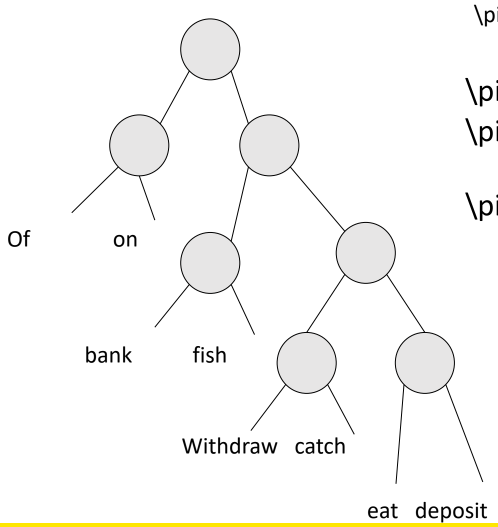
3)Unique path from root to leaf

Why are these properties desirable? (Hint: “Hierarchical softmax”)



39

Huffman tree over vocabulary



\pi: Huffman code

\pi (of) = [0, 0, 0, …]

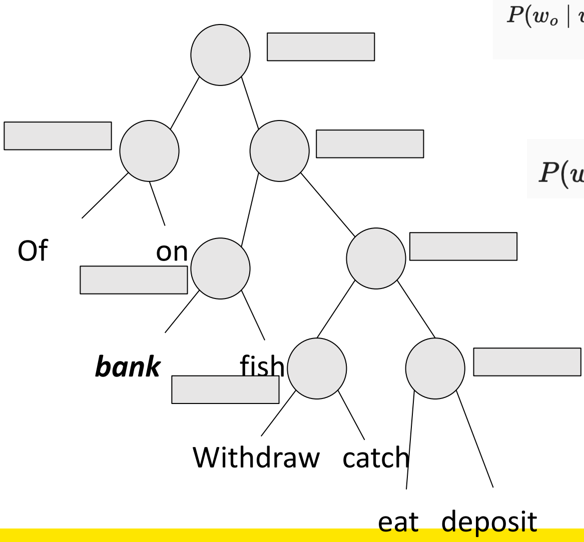
\pi (on) = [0, 1, …]

\pi (deposit) = ??

40

Hierarchy of softmax'es makes it easy

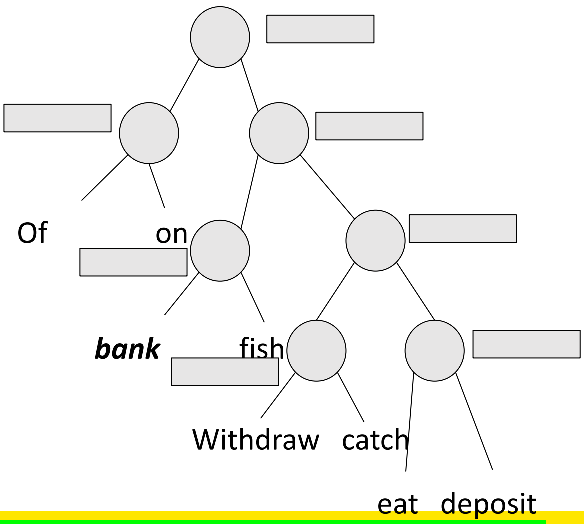
$$P(w_o \mid w_c) = \prod_{j=1}^{L(w_o)-1} \sigma\left(\llbracket n(w_o, j+1) = \mathrm{leftChild}(n(w_o, j)) \rrbracket \cdot \mathbf{u}_{n(w_o,j)}^{\top} \mathbf{v}_c\right);$$



$$P(w_3 \mid w_c) = \sigma(\mathbf{u}_{n(w_3,1)}^{\top} \mathbf{v}_c) \cdot \sigma(-\mathbf{u}_{n(w_3,2)}^{\top} \mathbf{v}_c) \cdot \sigma(\mathbf{u}_{n(w_3,3)}^{\top} \mathbf{v}_c)$$

Let's try the example with w_3 as "bank" and w_c as "deposit".

Hierarchy of softmax'es makes it easy



The representations of the intermediate nodes help to compute values of leaf nodes in their subtrees.

Softmax are computed as dot products of intermediate vectors along the tree, instead of V.

Challenge: Would a binary tree work instead of a Huffman tree?

Methods to optimise (2/2): Negative sampling

Multinomial classification converted to Binary classification: How?

Given a context word and a target word, predict if it is a valid combination.

Withdraw money -> True
Deposit money -> True
Eat money -> False
Catch money -> False

Withdraw fish -> False
Deposit fish -> False
Eat fish -> True
Catch fish -> True

But you only have the 'seen' corpus

What about the 'unseen/invalid' combinations?
'Sample' dummy instances

Hence, the name.

43

How do negative samples help?

Eat fish: 1

Eat money: 0

Eat if: 0

Eat table: 0

…

..

Negative samples

How many?

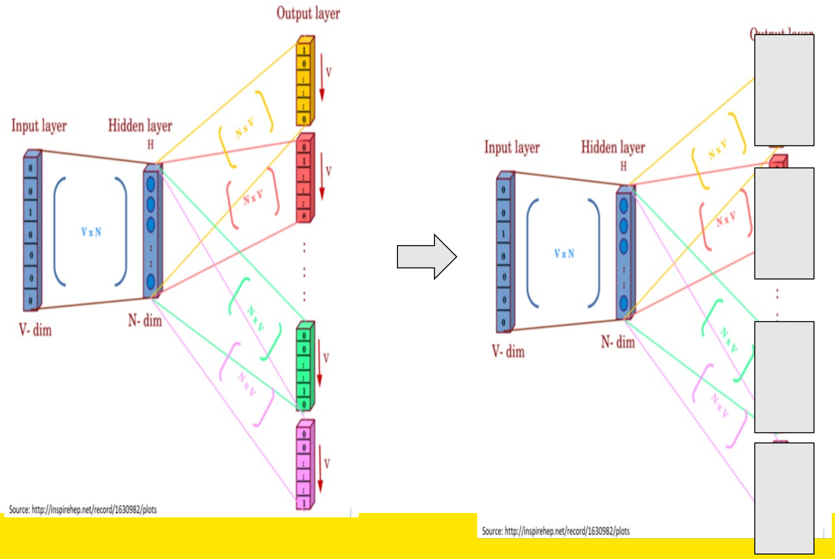
Should words be sampled randomly?

$$\arg\max_\theta \prod_{(w,c)\in D} p(D=1|c,w;\theta) \prod_{(w,c)\in D'} p(D=0|c,w;\theta)$$
$$= \arg\max_\theta \prod_{(w,c)\in D} p(D=1|c,w;\theta) \prod_{(w,c)\in D'} (1-p(D=1|c,w;\theta))$$
$$= \arg\max_\theta \sum_{(w,c)\in D} \log p(D=1|c,w;\theta) + \sum_{(w,c)\in D'} \log(1-p(D=1|w,c;\theta))$$
$$= \arg\max_\theta \sum_{(w,c)\in D} \log \frac{1}{1+e^{-v_c\cdot v_w}} + \sum_{(w,c)\in D'} \log(1-\frac{1}{1+e^{-v_c\cdot v_w}})$$
$$= \arg\max_\theta \sum_{(w,c)\in D} \log \frac{1}{1+e^{-v_c\cdot v_w}} + \sum_{(w,c)\in D'} \log(\frac{1}{1+e^{v_c\cdot v_w}})$$

Note the notational variation:
c: context word
w: center word

44

Negative sampling: Modifying the architecture



Each is now a logistic regression,
One for each word in the vocabulary.

Only a subset of them are trained per training instance.
Why?

45

Word2Vec Recap

- Dense representations of words
- Learn to predict word given context or vice versa
- Two kinds of models: skip-gram and CBOW
- Two potential optimisations:
  - Hierarchical softmax: Use a Huffman tree and compute softmax as product over vector representations of non-leaf nodes
  - Negative sampling: Create negative samples; use logistic regression to train a binary classification task

- https://radimrehurek.com/gensim/models/word2vec.html <-- Click here.

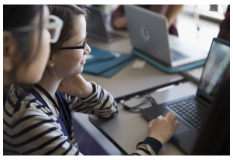
46

Other extensions of word2vec

Wang2vec: Order of words in a context (Liu et al, 2015)

Sense2vec: Sense embeddings in addition to word embeddings. Predict the word and the sense given the context and their sense.

Task-specific word vectors: Learning word vectors for sentiment analysis (Maas et al, 2011)



Demo time!

Liu et al, Two/Too Simple Adaptations of Word2Vec for Syntax Problems, NAACL 2015
Maas et al, Learning Word Vectors for Sentiment Analysis, ACL 2011

UNSW SYDNEY

47

Breather:

What phenomenon connects the following words? (Non-exhaustive list)

Dust

Screen

Rent

Sanction

UNSW SYDNEY

48

What phenomenon connects the following words? (Non-exhaustive list)

They are contronyms.

Dust: **Dust** the bread with some cinnamon ; **Dust** the shelves

Screen: **Screen** a film; **Screen**

Rent: **Rent** an apartment (as a tenant or an owner?)

Sanction: imposed a **sanction;** the project received a **sanction**

Word embeddings will assign the same embedding for the word.

GloVe: Global Vectors

A weighted least squares model that trains on global word-word co-occurrence counts

	...	dog	cheese	...
..		100	44	
pizza	43	12	327	
bone	324	324	.	
...	..	..	..	

X_{ij} : the number of times word j occurs in the context of word i.
X_{j}: \sum k X_{ik}
P_{ij} = P(j|i) = X_{ij}/X_{i}

Why Global? Co-occurrence matrix is global to the corpus

Goal: Learn word vectors **wi** such that:
The dot product of a word's vector with that of another word correlates with the co-occurrence of the two words

Note how this differs from word2vec!

Learning GloVe (1/2)

Let wi and wk be the word vectors

wi . ŵk = log P(k|i)
= log X_{ki} - log X_{i}

wk. ŵi = log P(i|k)
= log X_{ki} - log X_{k}

2.wi. ŵk = 2. log X_{ki} - log X_{i} - log X_{k}

wi. ŵk = log X_{ki} - 0.5 * log X_{i} - 0.5 * log X_{k}

51

Learning GloVe (2/2)

wi. ŵk = log X_{ki} - 0.5 * log X_{i} - 0.5 * log X_{k}

wi. ŵk = log X_{ki} - a -b

What you want to learn

What you know

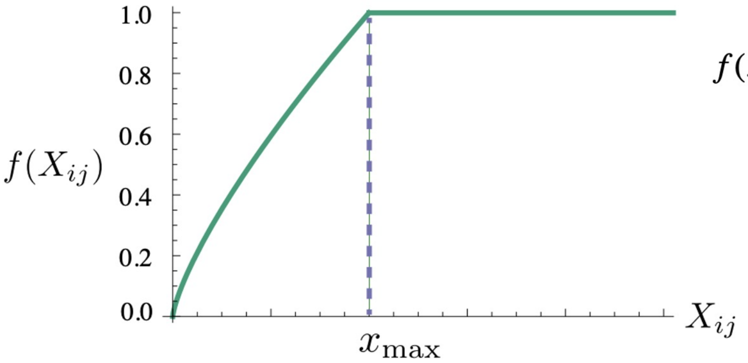
$$f(x) = \begin{cases} (x/x_{\max})^{\alpha} & \text{if } x < x_{\max} \\ 1 & \text{otherwise} \end{cases}.$$

Min. \sum (i,k) (wi.qk + a + b - log X_{ki})^2

Min. \sum (i,k) f(X_{ki}) (wi.qk + a + b - log X_{ki})^2

52

WHAT'S WITH THE DISTANCE FORMULA?



$$f(x) = \begin{cases} (x/x_{\max})^{\alpha} & \text{if } x < x_{\max} \\ 1 & \text{otherwise} \end{cases} .$$

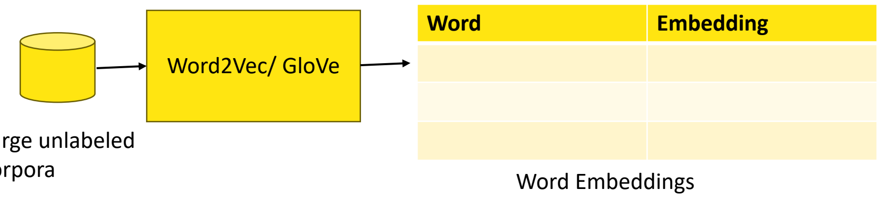
53

Comparison of word2vec and GloVe

Word2Vec	**GloVe**
Window-based method; Prediction-based	Hybrid of count-based and window-based method; Count-based
Learn vectors such that context word vectors can be predicted using a center word vector	Learn vectors such that co-occurrence can be predicted for a pair of words
Distance between center word and context words is a step function	Distance between center word and context words is modeled as a long-tailed function
The goal of the model is its predictive nature	The goal of the model is dimensionality reduction
The 'learning' is for each context window	The 'learning' is for pair of words
Negative sampling or hierarchical softmax may be used to approximate optimisation.	Distance-weighted to account for similarity

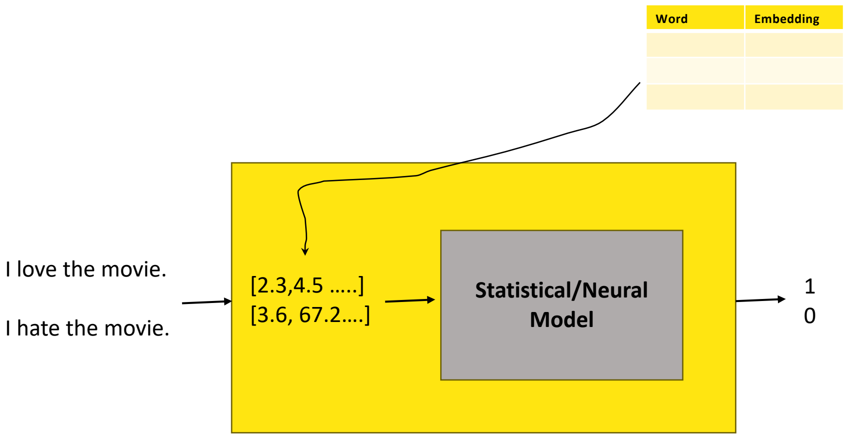
54

So, how can word embeddings be used? (1/2)



55

So, how can word embeddings be used? (2/2)



Word embeddings can be averaged to get sentence embeddings.

(Transformers **offer a better way** to do that – but we are learning word representations for now.)

Schnabel, T., Labutov, I., Mimno, D. and Joachims, T., 2015, September. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 conference on empirical methods in natural language processing* (pp. 298-307).

56

Part 2

Probabilistic Language Modeling

Suggested Reading: https://web.stanford.edu/~jurafsky/slp3/3.pdf

57

Representing sentences

- Sentences in a language were represented as computational grammar in the first generation of NLP
- Derives from compilers in programming languages
- First-generation of NLP
- The focus is 'belongingness':
  Can the grammar represent the set of sentences that belong to a 'language'
- 'Language': Set of valid strings

58

Grammar

Non-terminals : Capital letters
Terminals : Small-case words
Epsilon (ep): End of string

Grammar written in the form of rules

S -> K A B
K -> a | the | ep
A -> happy | sad
B -> man | woman | person

Valid strings: the happy woman, sad person, happy person, ...

59

Let's construct the grammar for this 'language'

Assume that a language has exactly the following valid sentences:

The boy eats rice
The girl eats rice
The boy drinks milk
The girl drinks milk

60

Let's construct the grammar for this 'language'

Assume that a language has exactly the following valid sentences:

The boy eats rice
The girl eats rice
The boy drinks milk
The girl drinks milk

S -> The P A
P -> boy | girl
A -> E | D
E -> eats ED
ED -> rice
D -> drinks DD
DD -> milk

61

Let's grow the grammar…

Assume that a language has exactly the following valid sentences:

The boy eats rice
The girl eats rice
The boy drinks milk
The girl drinks milk
The boy eats pizza
The girl eats pizza
The boy eats
The girl eats
The boy occasionally eats rice
The girl occasionally eats rice
The boy occasionally drinks milk
The girl occasionally drinks milk

S -> The P T A
P -> boy | girl
A -> E | D
T -> occasionally | ep
E -> eats ED | eats
ED -> rice | pizza
D -> drinks DD
DD -> milk

62

What are the limitations of rule-based grammar for languages?

S -> The P T A
P -> boy | girl
A -> E | D
T -> occasionally | ep
E -> eats ED | eats
ED -> rice | pizza
D -> drinks DD
DD -> milk

The list of sentences needs to be known in advance.

Accommodating new sentences may be cumbersome.

Can test 'belongingness' but not so much 'generation'.

63

Probabilistic Language Modeling

Belongingness -> Likelihood

How 'likely' is the sentence?

-> What is the probability of this sentence?

Sentence: w1, w2, w3…..

Likelihood: $P(w_1, w_2, \ldots w_n) = P(w_n \mid w_1, w_2 \ldots, w_{n-1}).P(w_{n-1} \mid w_1, w_2, \ldots w_{n-2}) \ldots .P(w_2 \mid w_1).P(w_1)$

Sentence: "The girl eats rice"

Likelihood: P("The girl eats rice") = P("rice" | "The girl eats"). P("eats" | "The girl"). P("girl" | "The"). P("The" | ep)

64

Use a dataset to compute the probabilities

P("The girl eats rice") = P("rice" | "The girl eats"). P("eats" | "The girl"). P("girl" | "The"). P("The" | ep)

The boy eats rice
The girl eats rice
The boy drinks milk
The girl drinks milk

P("rice" | "The girl eats") = 1/1 = 1
P("eats" | "The girl") = 1/ 2 = 0.5
P("girl" | "The") = 2/4 = 0.5
P("The" | ep) = 4/4 = 1

P("The girl eats rice") = 0.25

65

N-gram assumption

Assume that a word only depends on the word before it: bi-gram assumption (i.e., N= 2)

$$P(w_1, w_2, \dots w_n) = P(w_n \mid w_1, w_2 \dots, w_{n-1}).P(w_{n-1} \mid w_1, w_2, \dots w_{n-2})\dots.P(w_2 \mid w_1).P(w_1)$$

$$\Downarrow$$

$$P(w_1, w_2, \dots w_n) = P(w_n \mid w_{n-1}).P(w_{n-1} \mid w_{n-2})\dots P(w_2 \mid w_1).P(w_1)$$

Bigram probability is easier to compute (especially with large datasets)
Provides variability to the generated language model

66

Let's understand the connection between probability and sentence completion

Select the most likely word to fill the gap.

1. three ______ : (a) apples, (b) bottles, (C) happy
2. ate three _____ : (a) apples, (b) bottles, (C) happy
3. drank three _____ : (a) apples, (b) bottles, (C) happy

UNSW SYDNEY

67

Let's understand the connection between probability and sentence completion

Select the most likely word to fill the gap.

1. three ______ : (a) apples, (b) bottles, (C) happy P(w1 = "apples" | w0 = "three") > P(w1 = "happy" | w0 = "three")
2. ate three _____ : (a) apples, (b) bottles, (C) happy P(w2 = "apples" | w1 = "three", w0="ate") > P(w2 = "happy" | w1 = "three", w0 = "ate")
3. drank three _____ : (a) apples, (b) bottles, (C) happy P(w2 = "bottles" | w1 = "three", w0="drank") > P(w2 = "apples" | w1 = "three", w0 = "ate")

UNSW SYDNEY

68

But.. what happens if your dataset had not seen the pattern?

P("The girl eats rice") = P("rice" | "eats"). P("eats" | " girl"). P("girl" | "The"). P("The" | ep)

The boy eats rice
The boy drinks milk
The girl drinks milk

P("rice" | "eats") = 1/1 = 1
P("eats" | " girl") = 1/ 2 = 0
P("girl" | "The") = 2/4 = 0.5
P("The" | ep) = 4/4 = 1

P("The girl eats rice") = 0

UNSW SYDNEY

69

A simpler example: How likely is "great joke"?

"great day"
"great story"
"great achievement"

N-grams	Count
#day	1
#story	1
#achievement	1
#"great day"	1
#"great story"	1
#"great achievement"	1
#"great"	3

Prob.	Value
P("day" | "great")	#("great day")/#("great") = 1/3 = 0.33
P("story" | "great")	#("great story")/#("great") = 1/3 = 0.33
P("achievement" | "great")	#("great achievement")/#("great") = 1/3 = 0.33
P("joke" | "great")	#("great joke")/#("great") = 0/3 = 0

UNSW SYDNEY

70

Smoothing: Making impossible possible

'Bump up' the zero probabilities by a small amount so that no n-grams have zero probability

Smoothing: Statistical technique to modify the probabilities so that differences in probabilities are reduced.

Prob.	Value	Smoothed values
P("day" | "great")	#("great day")/#("great") = 1/3 = 0.33	#("great day")+1/#("great")+4 = 2/7 = 0.28
P("story" | "great")	#("great story")/#("great") = 1/3 = 0.33	2/7 = 0.28
P("achievement" | "great")	#("great achievement")/#("great") = 1/3 = 0.33	2/7= 0.28
P("joke"|"great")	0/1 = 0	0+1/1+4 = 1/ 4 = 0.25

$$P(w_i) = \frac{c_i}{N} \Rightarrow P_{\text{Laplace}}(w_i) = \frac{c_i+1}{N+V}$$

https://web.stanford.edu/~jurafsky/slp3/3.pdf

71

Smoothing Techniques

Add-one smoothing (Laplace smoothing)

$$P_{\text{Laplace}}(w_i) = \frac{c_i+1}{N+V}$$

Interpolation-based smoothing

$$\begin{aligned}\hat{P}(w_n|w_{n-2}w_{n-1}) &= \lambda_1 P(w_n) \\ &+\lambda_2 P(w_n|w_{n-1}) \\ &+\lambda_3 P(w_n|w_{n-2}w_{n-1})\end{aligned}$$

72

Using probabilistic language model for generation

$	The	boy

Language generation becomes a prediction problem.

Input: Sequence so far
Output: Next word (among all words in the vocabulary)

Prob.	Value
P("is"|"boy")	0.01
P("goes"|"boy")	0.02
P("pizza"|"boy")	0.00001
…	…
….	…
…	[V words]

Sampling may also be used. Why? How?

Demo time!

73

What is the problem with this method?

$	The	boy

Prob.	Value
P("is"|"boy")	0.01
P("goes"|"boy")	0.02
P("pizza"|"boy")	0.00001
…	…
….	…
…	[V words]

Context is limited to the length of the n-gram.
A sentence may not make sense.

Sentences require long-term context.
"The students in a class …" : learn/learns?

We need a better way to do this!!

74

Metric: Perplexity

Perplexity of a language model is the inverse probability of a test set

$$\begin{aligned}\text{perplexity}(W) &= P(w_1 w_2 \ldots w_N)^{-\frac{1}{N}} \\ &= \sqrt[N]{\frac{1}{P(w_1 w_2 \ldots w_N)}}\end{aligned}$$

Estimate of belongingness
"How well does the language model capture the sentences in the test set"
Lower the better
The language model is not 'perplexed' to accept the test sentences as valid sentences
Where can you use perplexity?

75

Australia's Global University

Part 3

Sequential neural language modeling

Suggested Reading: https://web.stanford.edu/~jurafsky/slp3/9.pdf

76

Let's progress from words to longer text.

For sale: Baby shoes. Never worn.

Trigger Warning: Morbid.

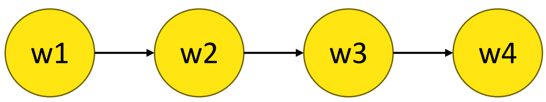
https://en.wikipedia.org/wiki/For_sale:_baby_shoes,_never_worn

77

Prob -> Recurrent

Auto-regressive models: Why?

$$P(w_1, \ldots, w_T) = \prod_{i=1}^{i=T} P(w_i | w_1, \ldots, w_{i-1}) \approx \prod_{i=1}^{i=T} P(w_i | w_{i-(n-1)}, \ldots, w_{i-1})$$



What. if …..?

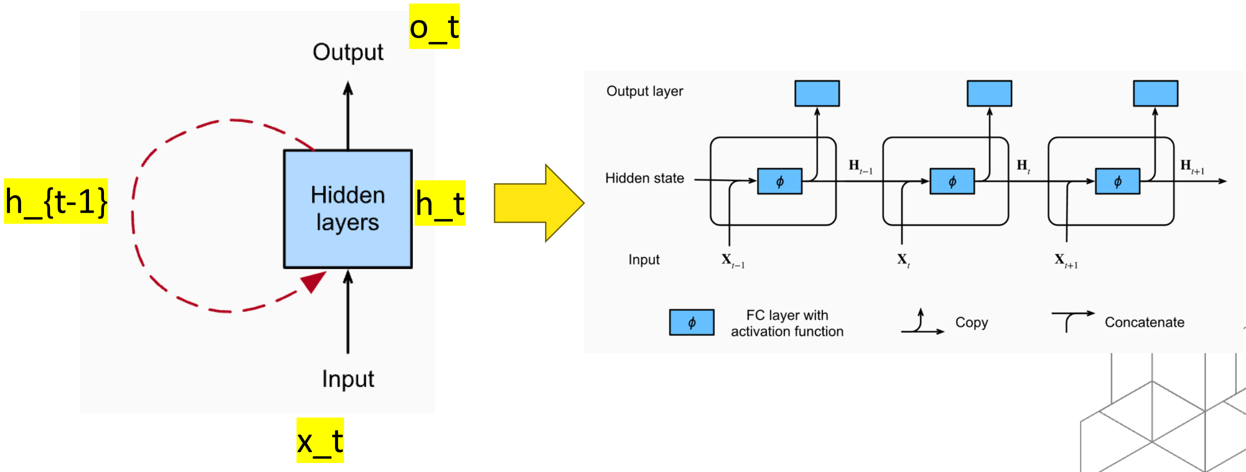
$$P(x_t \mid x_{t-1}, \ldots, x_1) \approx P(x_t \mid h_{t-1})$$

$$h_t = f(x_t, h_{t-1})$$

https://d2l.ai/chapter_recurrent-neural-networks/rnn.html

78

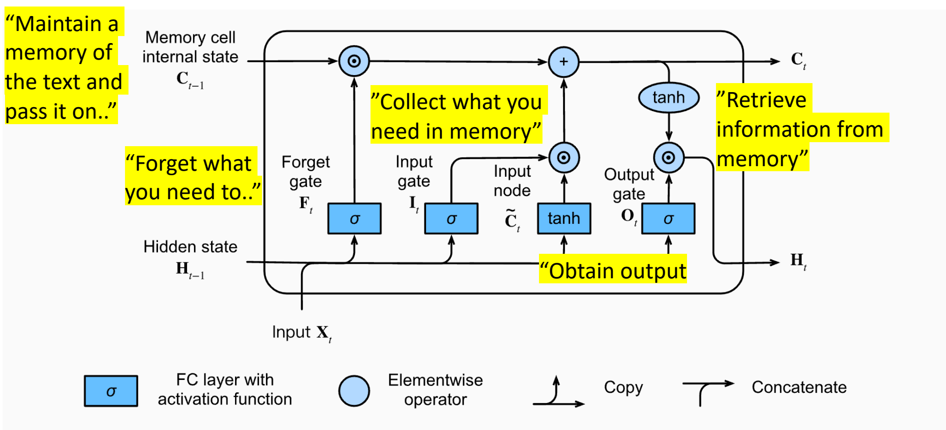
"Unrolling an RNN"



79

Specialised RNNs: LSTM

LSTMs: Long Short-term Memory (Can we increase the **distant memory** of an RNN?)



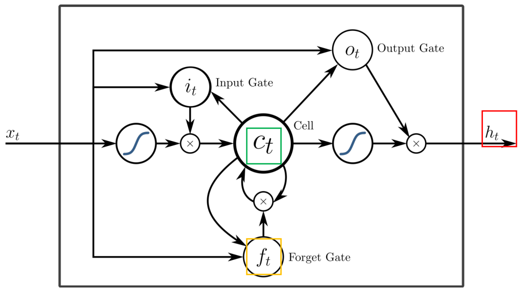
80

LSTM

An extension of recurrent neural network (RNN)

Unit: LSTM cell

"Act Now" | "Forget what is not important" | "Remember what is important for the future"

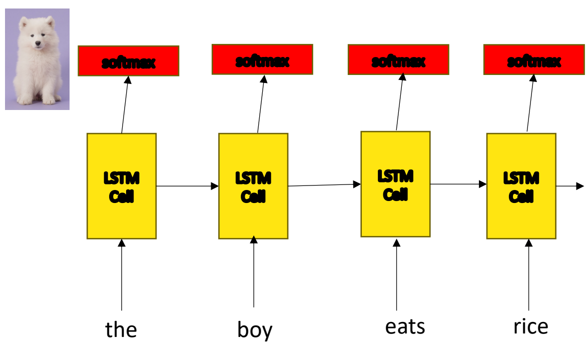


https://web.stanford.edu/~jurafsky/slp3/9.pdf

UNSW SYDNEY

81

LSTM as a chain: Train a language model

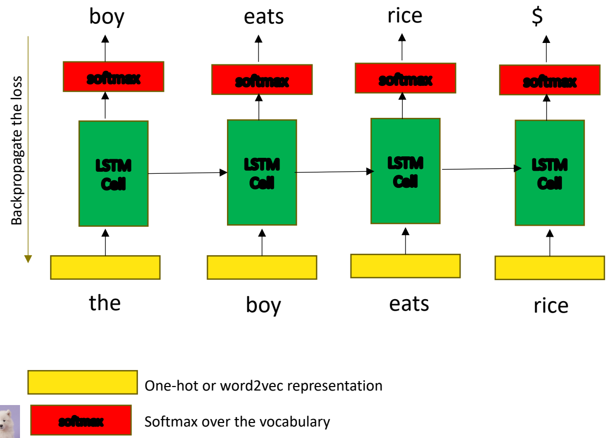


$$L_{CE}(\hat{\mathbf{y}}_t, \mathbf{y}_t) = -\log \hat{\mathbf{y}}_t[w_{t+1}]$$

UNSW SYDNEY

82

How does the learning proceed?



83

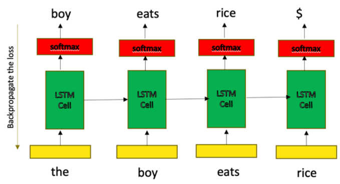
Let’s make a language model with LSTMs

We will be using Keras, an open-source Python library
Keras is a wrapper on the top of TensorFlow.
… not as popular since Transformers.

Demo time!

84

Challenges with LSTMs



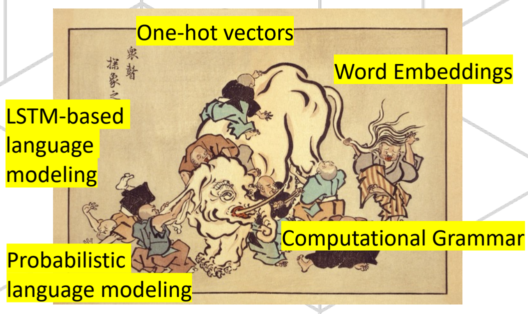
-Relies on linear information-passing (as in the case of probabilistic language modeling)
-Language has long-distance dependencies

Can we have a mechanism to pass information between non-consecutive hidden states?

UNSW SYDNEY

85

UNSW SYDNEY Australia's Global University



The story of the six blind men and the elephant

They interpreted the elephant differently based on which part of the elephant's body they touched.

86

Summary

Part	Key Idea	Demos
Representation matters	One-hot vectors and their limitations	Vectorizer
Word2vec & GloVe	Word representation using context prediction or co-occurrence estimation	Word2Vec using gensim
Probabilistic language modeling	Language generation as conditional probability; smoothing helps.	Probabilistic language modeling using NLTK primitives
Sequential neural language modeling	RNNs/LSTMs can help mitigate the problem in probabilistic language modeling. However, linear structures limit the capability of the models.	Simple LSTM-based language model using Keras

UNSW SYDNEY

87

Suggested Reading

Chapter 2: Representation Learning - Probabilistic Language Modeling: https://web.stanford.edu/~jurafsky/slp3/3.pdf

LSTM/RNN: https://web.stanford.edu/~jurafsky/slp3/9.pdf

https://jalammar.github.io/illustrated-word2vec/

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed Representations of Words and Phrases and their Compositionality. In Proceedings of NIPS, 2013.

Pennington, J., Socher, R. and Manning, C.D., 2014, October. Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).
https://radimrehurek.com/gensim/auto_examples/tutorials/run_word2vec.html
https://github.com/stanfordnlp/GloVe
https://spacy.io/usage/vectors-similarity

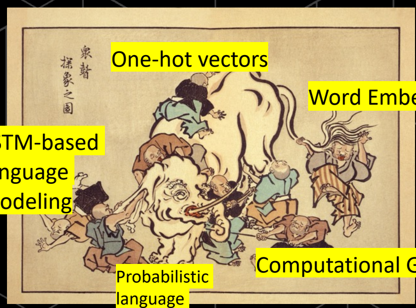
Advanced Reading: https://arxiv.org/pdf/1411.2738

UNSW SYDNEY

88

Can we have a mechanism to pass information between non-consecutive hidden states?

➔ Attention
(Coming up next!)



89