CS772: Deep Learning for Natural Language Processing (DL-NLP)

Word2vec and Glove

Pushpak Bhattacharyya
Computer Science and Engineering Department
IIT Bombay

Week 5 of 30th Jan, 2023

Re-cap

Deriving the word vector: setting

$$W^s : w_0^s, w_1^s, w_2^s, \ldots w_i^s, \ldots w_m^s$$

$$V_{w_i} : [v_0^i, v_1^i, v_2^i, \ldots v_k^i, \ldots v_d^i]$$

$$J = P(w_j \mid w_i)$$

$$L = -P(w_j \mid w_i)$$

$$P(w_j \mid w_i) = \frac{e^{V_{w_i} . V_{w_j}}}{\sum_{j'=1}^{|V|} e^{V_{w_i} . V_{w_{j'}}}}$$

$$LL = -V_{w_i} . V_{w_j} + \ln\left(\sum_{j'=1}^{|V|} e^{V_{w_i} . V_{w_{j'}}} \right)$$

W^S: word sequence in the s^{th} Sentence

V_{wi}: word vector of w_i

Deriving the word vector: Optimization (1/2)

$$V_{w_i} : [v_0^i, v_1^i, v_2^i, \ldots v_k^i, \ldots v_d^i] = [u_0, u_1, u_2, \ldots u_k, \ldots u_d]$$

$$V_{w_j} : [v_0^j, v_1^j, v_2^j, \ldots v_k^j, \ldots v_d^j] = [v_0, v_1, v_2, \ldots v_k, \ldots v_d]$$

$$V_{w_{j'}} : [v'_0, v'_1, v'_2, \ldots v'_k, \ldots v'_d]$$

$$V_{w_i} . V_{w_j} = \sum_{k=0}^{d} u_k v_k$$

$$\frac{\partial LL}{\partial u_k} = -v_k + \frac{\frac{\partial}{\partial u_k}\left(\sum_{j'=1}^{|V|} e^{\sum_{k=0}^{d} u_k v_k^{'}} \right)}{\sum_{j'=1}^{|V|} e^{\sum_{k=0}^{d} u_k v_k^{'}}}$$

Deriving the word vector: Optimization

$$= -v_k + \frac{\sum_{j'=1}^{|V|} \frac{\partial}{\partial u_k}\left(e^{\sum_{k=0}^{d} u_k v_k^{'}} \right)}{\sum_{j'=1}^{|V|} e^{\sum_{k=0}^{d} u_k v_k^{'}}} = -v_k + \frac{\sum_{j'=1}^{|V|} e^{\sum_{k=0}^{d} u_k v_k^{'}} \frac{\partial}{\partial u_k}\left(\sum_{k=0}^{d} u_k v_k^{'} \right)}{\sum_{j'=1}^{|V|} e^{\sum_{k=0}^{d} u_k v_k^{'}}}$$

$$= -v_k + \frac{\sum_{j'=1}^{|V|} e^{\sum_{k=0}^{d} u_k v_k^{'}} v_k^{'}}{\sum_{j'=1}^{|V|} e^{\sum_{k=0}^{d} u_k v_k^{'}}} = -v_k + \sum_{j'=1}^{|V|} P(w_{j'} \mid w_i).v_{k'} = -v_k + E(v_{k'})$$

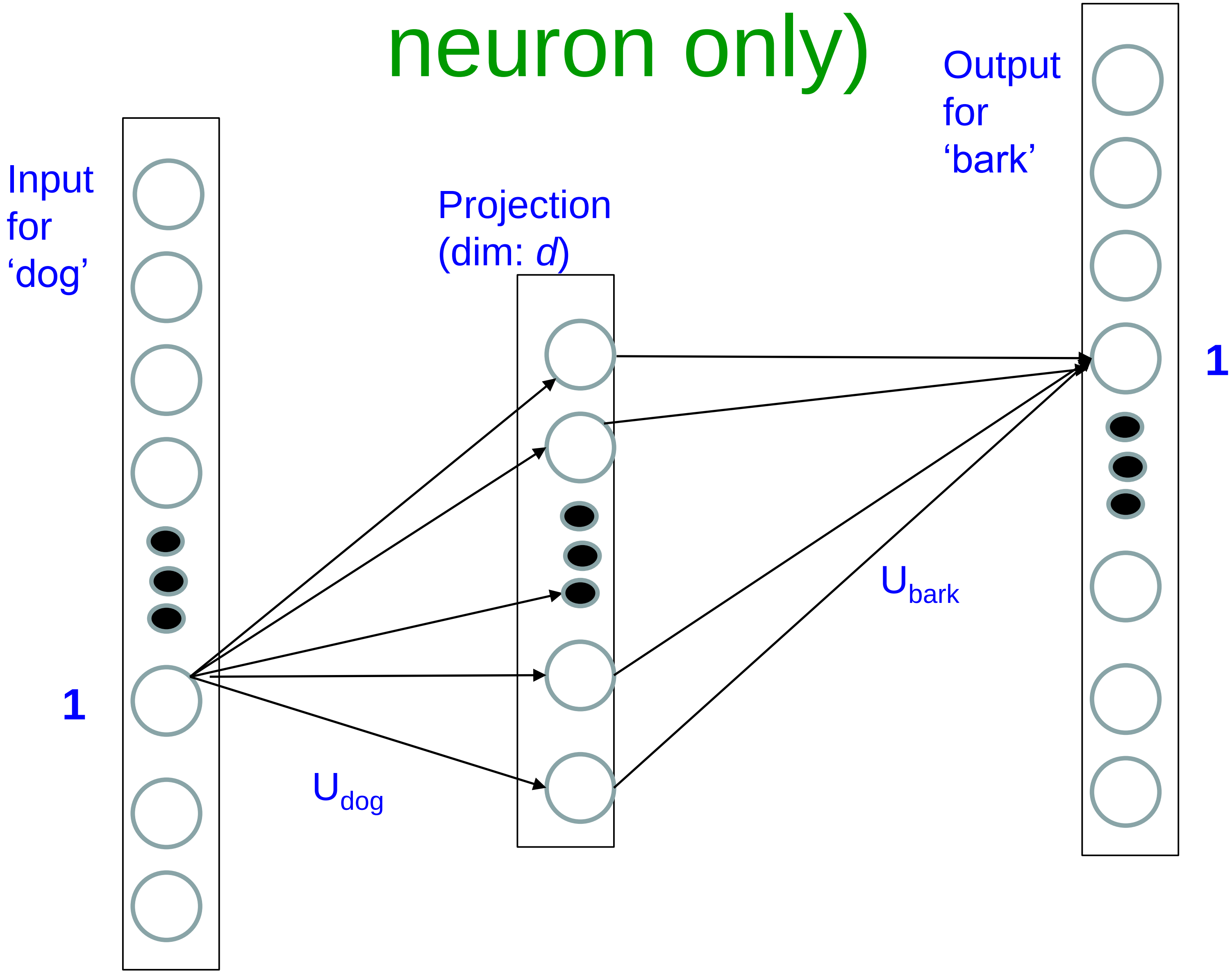
Deriving the word vector, Gradient Descent: Δu_k

$$\Delta u_k = -\eta \frac{\partial LL}{\partial u_k} = \eta[v_k - E(v_{k'})]$$

Example

- We want, say, *P(‘bark’|’dog’)*
- Take the weight vector **FROM** ‘dog’ neuron **TO** projection layer (call this U_{dog})
- Take the weight vector **TO** ‘bark’ neuron **FROM** projection layer (call this U_{bark})
- When initialized, U_{dog} and U_{bark} give the initial estimates of word vectors of ‘dog’ and ‘bark’
- The weights and therefore the word vectors get fixed by back propagation

Input to Projection (shown for one neuron only)



Modelling *P(context word|input word)* *(2/2)*

- To model the probability, first compute dot product of u_{dog} and v_{bark}
- Exponentiate the dot product
- Take softmax over all dot products over the whole vocabulary

$$P('bark'|'dog') = \frac{\exp(U_{dog}^{T} U_{bark})}{\sum_{R \varepsilon Vocabulary} \exp(U_{dog}^{T} U_{R})}$$

P('bark'|'dog') (1/2)

$$P('bark'|'dog') = \frac{\exp(U_{dog}^{T} U_{bark})}{\sum_{R \varepsilon Vocabulary} \exp(U_{dog}^{T} U_{R})}$$

$$\log(P('bark'|'dog')) = U_{dog}^{T} U_{bark} - \log(\sum_{R \varepsilon Vocabulary} \exp(U_{dog}^{T} U_{R}))$$

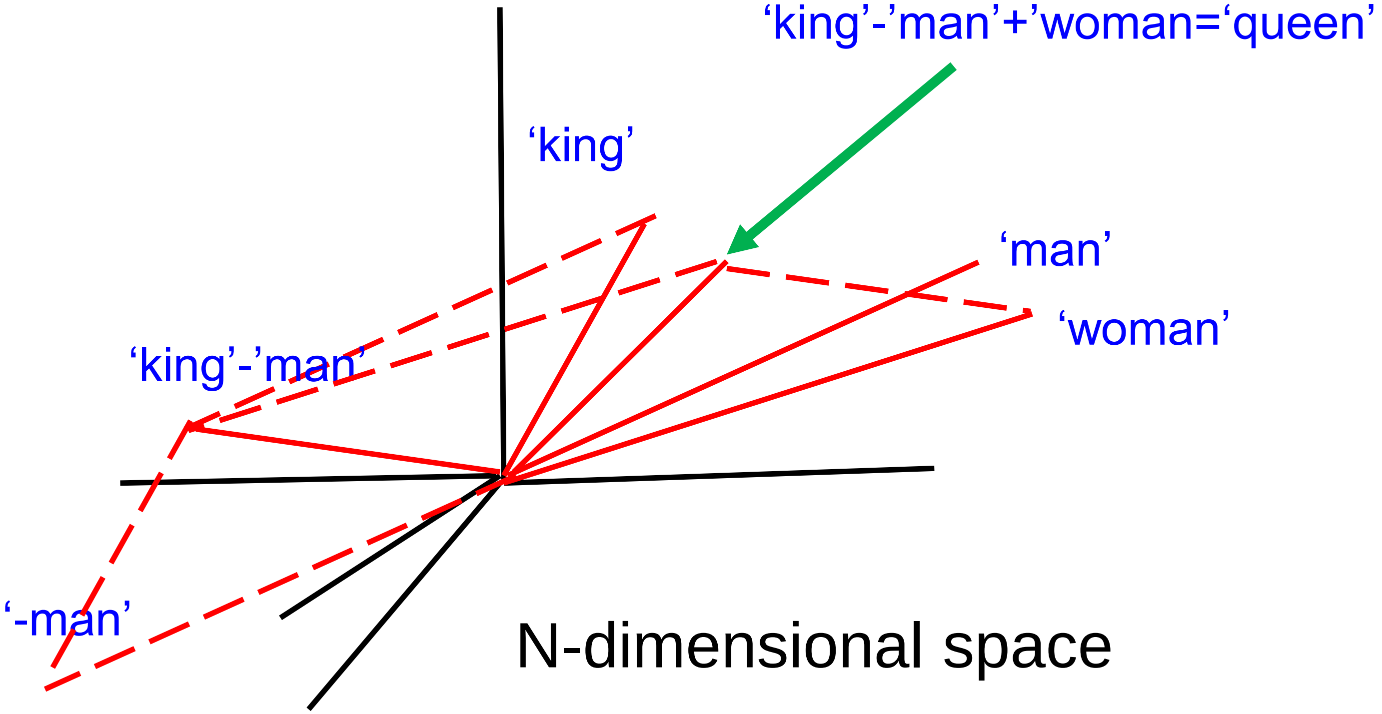
Word2vec architectures

Mikolov 2013

Classic work

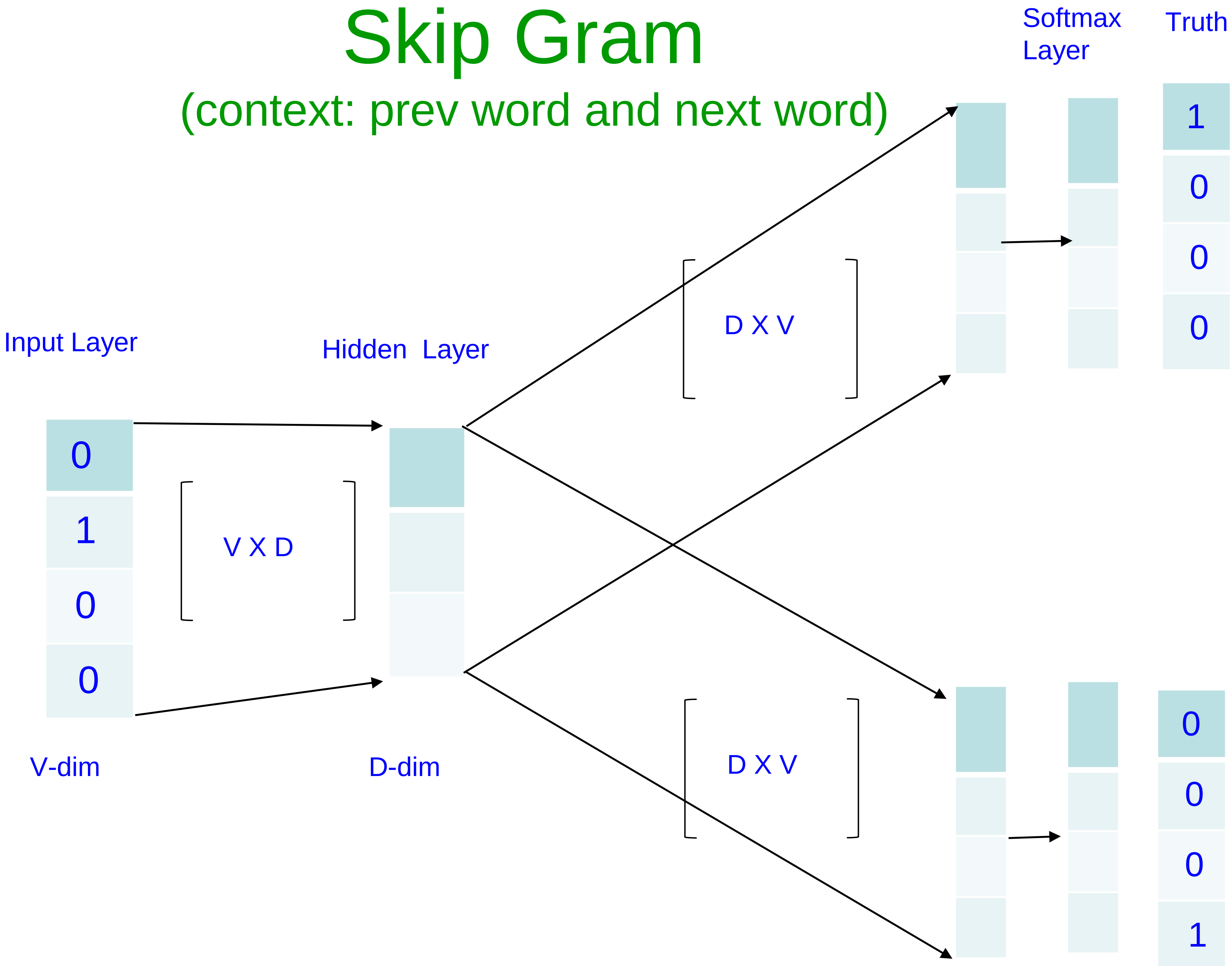
- Caught the attention of the world by equations like

'king'-'man'+'woman'='queen'

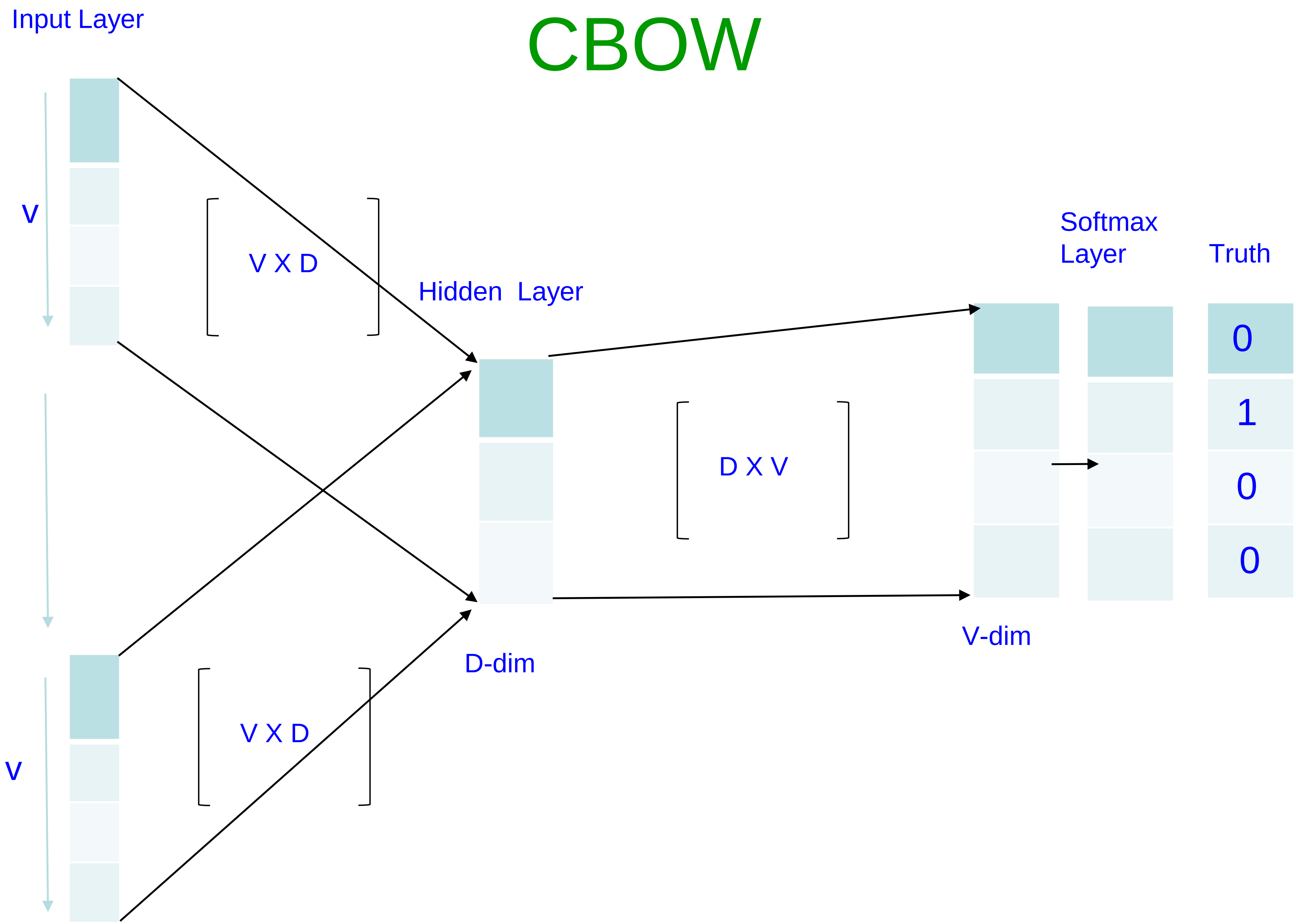


Skip Gram

(context: prev word and next word)



CBOW



Input Layer
V
V X D
Hidden Layer
Softmax
Layer
Truth
0
1
0
0
D X V
V-dim
D-dim
V
V X D

Symbolic approach to representing word meaning

Syntagmatic and Paradigmatic Relations

- Syntagmatic and paradigmatic relations
  - Lexico-semantic relations: synonymy, antonymy, hypernymy, mernymy, troponymy etc. **CAT is-a ANIMAL**
  - Coccurence: **CATS MEW**
- Resources to capture semantics:
  - Wordnet: primarily paradigmatic relations
  - ConceptNet: primarily Syntagmatic Relations

Syntagmatic and Paradigmatic Relations cntd.

- There are interesting studies for English on the syntagmatic and paradigmatic association
- The study finds that when a subject hears a word the words that come on hearing, are 50% syntagmatic and 50% paradigmatic
- Thus on hearing ‘dog’, the words ‘animal’, ‘mammal’, ‘tail’ etc. are pulled as paradigmatic and ‘bark’, ‘friend’, ‘police’ etc. as syntagmatic
- In particular, word vectors capture syntagmatic relations

Fundamental Device- Lexical Matrix (with examples)

Word Meanings	**Word Forms**				
	F_1	F_2	F_3	…	F_n
M_1	(*depend*) $E_{1,1}$	(*bank*) $E_{1,2}$	(rely) $E_{1,3}$		
M_2		(*bank*) $E_{2,2}$		(*embankment*) $E_{2,\ldots}$	
M_3		(*bank*) $E_{3,2}$	$E_{3,3}$		
…				…	
M_m					$E_{m,n}$

Wordnet

- **Princeton Wordnet** for English developed over 15 years. Released 1992.

- **Eurowordne**t- linked structure of European language wordnets built in 1998 over 3 years.

- **IndoWordnet** completed in 2010; effort of 10 years.

Basic Principle

- Words in natural languages are polysemous-meaning has many (‘poly’) meanings (‘sems’)

- However, when synonymous words are put together, a unique meaning often emerges.

- Use is made of *Relational Semantics.*
- Competing scheme: *Componential Semantics*, where a word is represented by features, e.g.,
  - Features: <Large?, Domesticable?, carnivorous?, furry?>
  - Tiger: <1, 0, 1, 1>, Cat: <0, 1, 1, 1>, Cow: <1, 1, 0, 0>

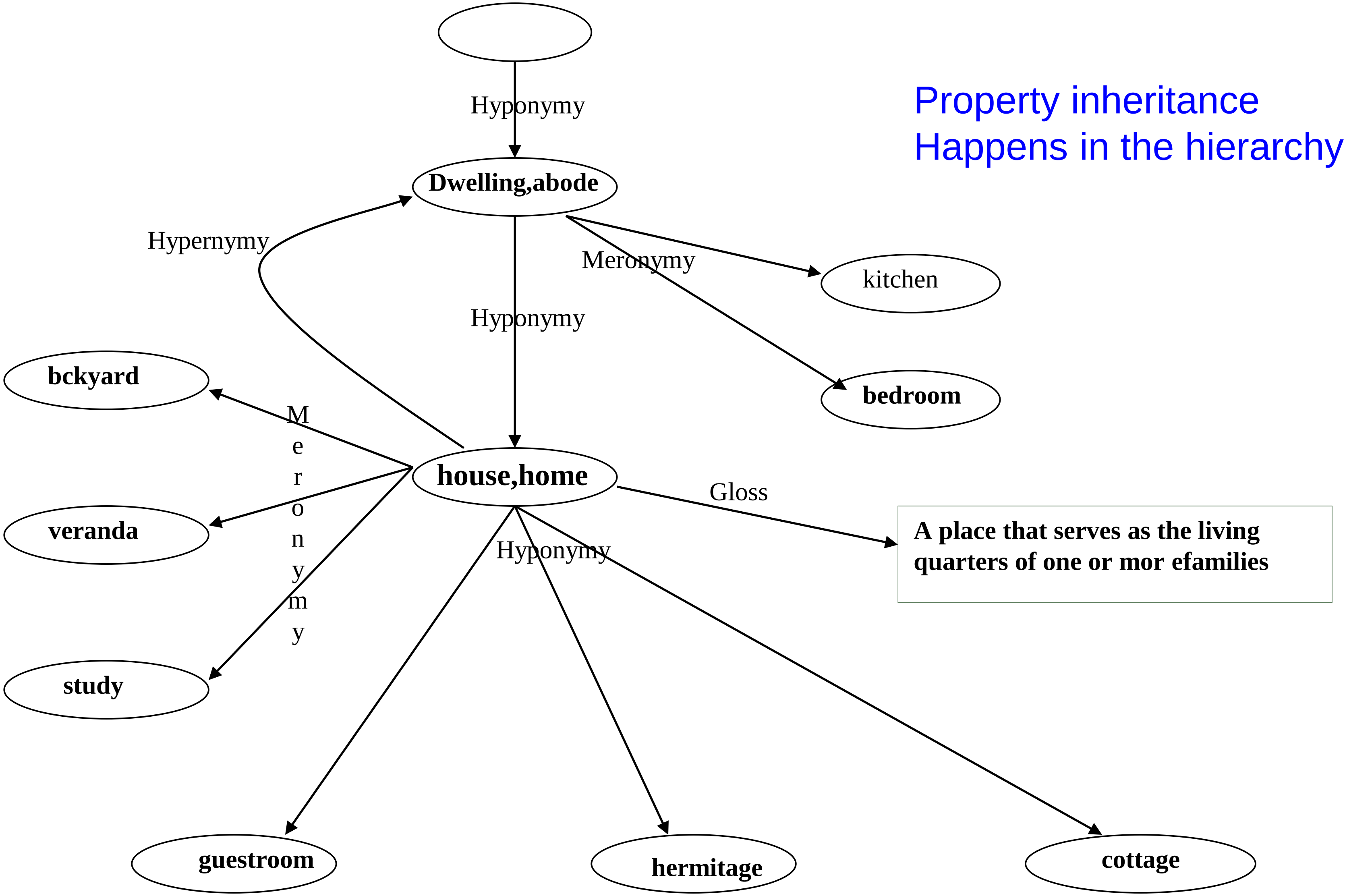
Lexical and Semantic relations in wordnet

1. Synonymy
2. Hypernymy / Hyponymy *(kind-of*)
3. Antonymy
4. Meronymy / Holonymy *(part of)*
5. Gradation
6. Entailment
7. Troponymy (*manner of*)

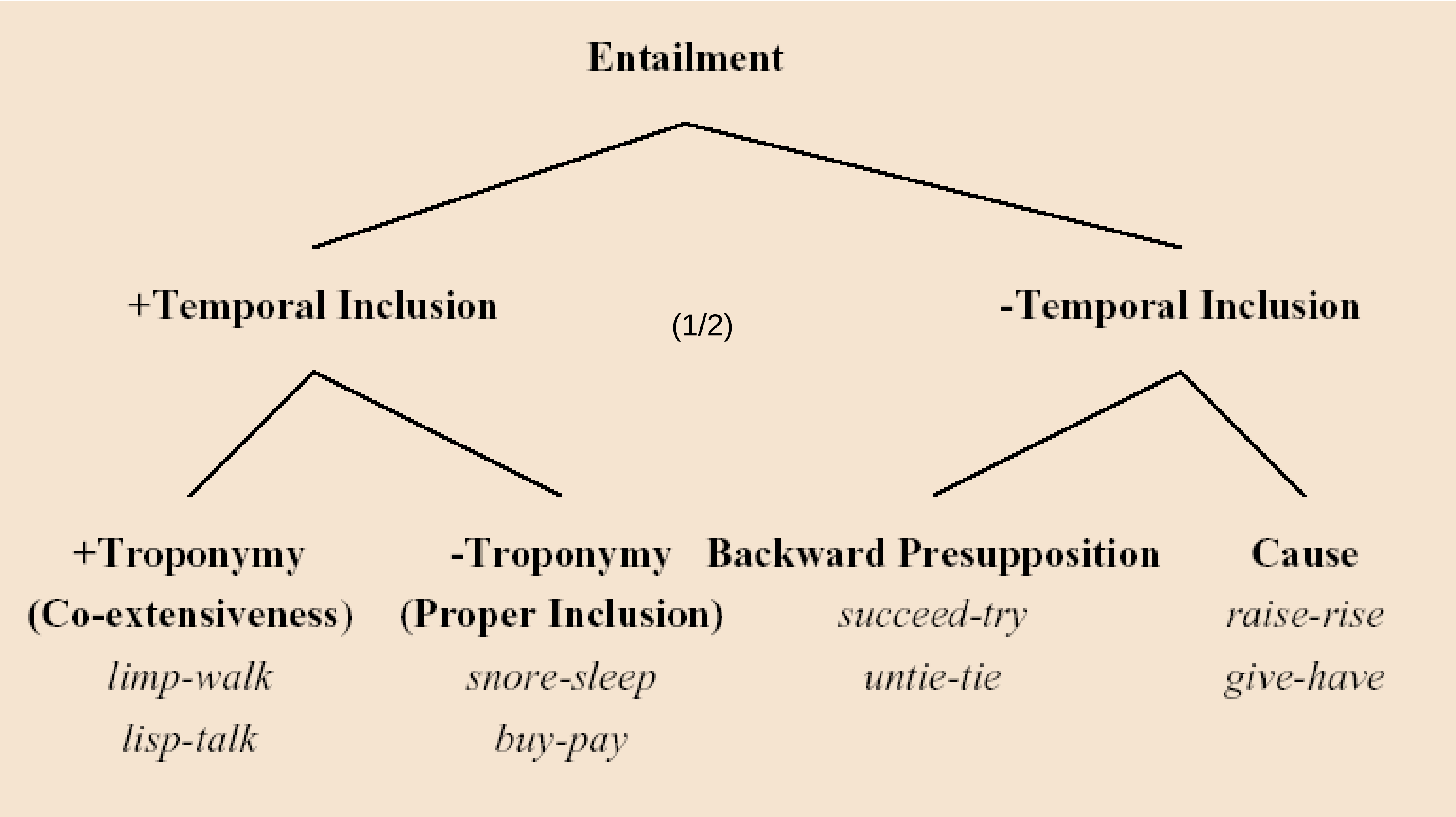
1, 3 and 5 are lexical (*word to word)*, rest are semantic (*synset to synset).*

WordNet Sub-Graph

Property inheritance
Happens in the hierarchy



Entailment: fundamental meaning relation linking verbs



Principles behind creation of Synsets

Three principles:

Minimality: (first decide the exact synonyms that are minimally needed to make the meaning unique)

Coverage: for that sense include ALL the words in the synset

Replacability: at least the first few words should be able to replace one anothere

Synset creation: example

Home

John's home was decorated with lights on the occasion of Christmas.

Having worked for many years abroad, John Returned home.

House

John's house was decorated with lights on the occasion of Christmas.

Mercury is situated in the eighth house of John's horoscope.

Synsets (continued)

{house} is ambiguous.

{house, home} has the sense of *a social unit living together;*

Is this the minimal unit?

{family, house} will make the unit completely unambiguous.

For coverage:

{family, household, house} ordered according to frequency.

Replacability of the most frequent words is a requirement which is satisfied

Representation using syntagmatic relations: Co-occurrence Matrix

Corpora: I enjoy cricket. I like music. I like deep learning

	I	**enjoy**	**cricket**	**like**	**music**	**deep**	**learning**
I	-	1	1	2	1	1	1
enjoy	1	-	1	0	0	0	0
cricket	1	1	-	0	0	0	0
like	2	0	0	-	1	1	1
music	1	0	0	1	-	0	0
deep	1	0	0	1	0	-	1
learning	1	0	0	1	0	1	-

Collocation and Co-occurrence

- Collocation: Two or more words that tend to appear frequently together.
  - Heavy rain
  - Scenic view
- Co-occurrence: A relation between two or more phenomena such that they tend to occur together.
  - Thunder co-occurs with lightning
  - Bread and butter.

Project Idea

- Detect oxymorons given a piece of text.
- Oxymoron: A figure of speech in which apparently contradictory terms appear in conjunction.
  - *Original copy*
  - *Awfully good*
  - *Silent scream*

Co-occurence Matrix

Fundamental to NLP

Also called **Lexical Semantic Association (LSA)**

Very sparse, many 0s in each row

Apply Principal Component Analysis (PCA) or Singular Value Decomposition (SVD)

Do Dimensionality Reduction; merge columns with high internal affinity (e.g., *cricket* and *bat*)

Compression achieves better semantics capture

GLOVE

Pennigton et al, 2014

Two main models for learning word vectors

- 1) global matrix factorization methods, such as latent semantic analysis (LSA) (Deerwester et al., 1990) and
- 2) local context window methods, such as the skip-gram model of Mikolov et al. (2013)
- Currently, both families suffer significant drawbacks.

Drawbacks

- Methods like LSA efficiently leverage **statistical information**, but they do relatively poorly on the word analogy task,
  - indicating a sub-optimal vector space structure.
- Skip-gram may do better on the **analogy task**, but they poorly utilize the statistics of the corpus
  - since they train on separate local context windows instead of on global co-occurrence counts

Matrix Factorization Methods

- LSA: “term-document” matrix
  - Rows→ words or terms, and columns→ documents in the corpus.
- Hyperspace Analogue to Language (HAL) (Lund and Burgess, 1996): “term-term” matrix
  - rows and columns→ words and
  - entries → the number of times a given word occurs in the context of another given word

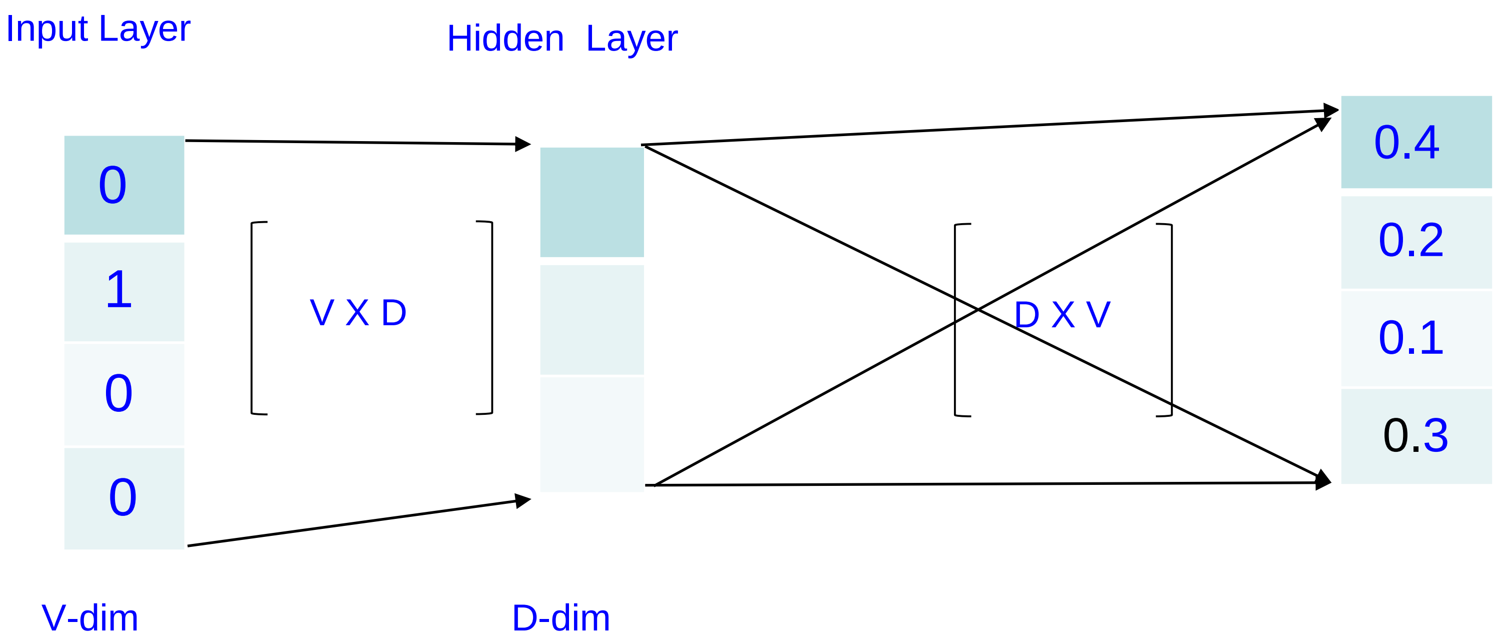
Matrix Factorization: drawback

- “most frequent words contribute a disproportionate amount to the similarity measure: the number of times two words co-occur with *the* or *and*, for example, will have a large effect on their similarity despite conveying relatively little about their semantic relatedness.”

Skip Gram & CBOW: drawback

- “shallow window-based methods suffer from the disadvantage that they do not operate directly on the co-occurrence statistics of the corpus. Instead,these models scan context windows across the entire corpus, which fails to take advantage of the vast amount of repetition in the data”

chitecture for Glove work?



Representation using syntagmatic relations: Co-occurrence Matrix

Corpora: I enjoy cricket. I like music. I like deep learning

	I	enjoy	cricket	like	music	deep	learning
I	-	1	1	2	1	1	1
enjoy	1	-	1	0	0	0	0
cricket	1	1	-	0	0	0	0
like	2	0	0	-	1	1	1
music	1	0	0	1	-	0	0
deep	1	0	0	1	0	-	1
learning	1	0	0	1	0	1	-

Solution: uses co-occurences

$$J = \sum_{i,j=1}^{V} f\left(X_{ij}\right)\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij}\right)^2$$

Working out a simple case of word2vec

Example (1/3)

- 4 words: *heavy, light, rain, shower*
  - *Heavy:* U_0 *<0,0,0,1>*
  - *light:* U_1*: <0,0,1,0>*
  - *rain:* U_2*: <0,1,0,0>*
  - *shower:* U_3*: <1,0,0,0>*
- We want to predict as follows:
  - *Heavy* → *rain*
  - *Light* → *shower*

Note

- Any bigram is theoretically possible, but actual probability differs

- E.g., heavy-heavy, heavy-light are possible, but unlikely to occur
- Language imposes constraints on what bigrams are possible
- Domain and corpus impose further restriction

Example (2/3)

- We will call input as U and output as V
  - *Heavy: U_0 <0,0,0,1>, light: U_1: <0,0,1,0>, rain: U_2: <0,1,0,0>, shower: U_3: <1,0,0,0>*

- *Heavy: V_0 <0,0,0,1>, light: V_1: <0,0,1,0>, rain: V_2: <0,1,0,0>, shower: V_3: <1,0,0,0>*

Example (3/3)

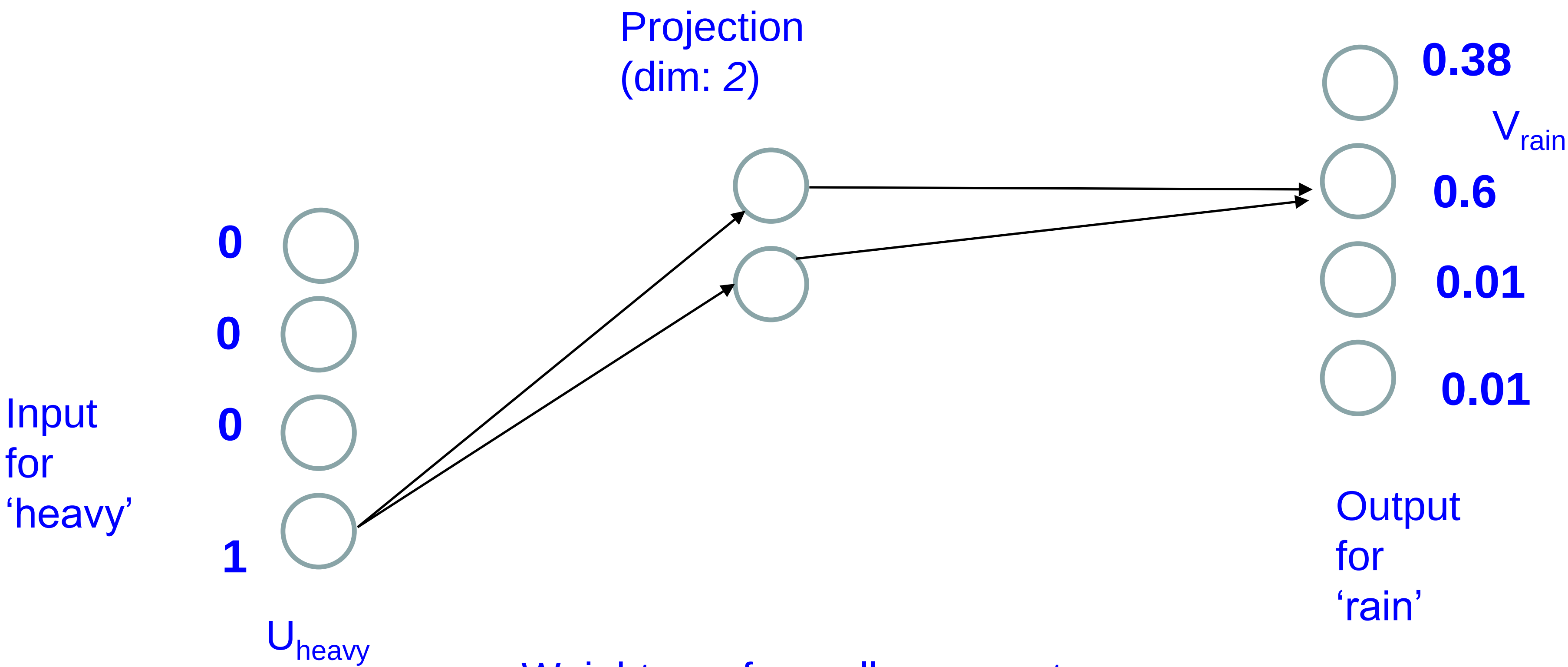
- *heavy→ rain*
  - *heavy: U_0 <0,0,0,1>*

    →

  - *rain: V_2: <0,1,0,0>*

- *light→ shower*
  - *light: U_1: <0,0,1,0>, → shower: V_3: <1,0,0,0>*

Word2vec n/w



Weights go from all neurons to
all neurons in the next layer; shown
For only one input and output

Chain of thinking

- P(rain|heavy) should be the highest
- So the output from V2 should be the highest because of softmax
- This way of converting an English statement into probability in insightful