Corso di Biblioteche Digitali

- Vittore Casarosa
  - casarosa@isti.cnr.it
  - tel. 050-621 3115
  - cell. 348-397 2168
  - Skype vittore1201
- Ricevimento dopo la lezione o per appuntamento
- Valutazione finale
  - 70% esame orale
  - 30% progetto (una piccola biblioteca digitale)
- Materiale di riferimento:
  - Ian Witten, David Bainbridge, David Nichols, How to build a Digital Library, Morgan Kaufmann, 2010, ISBN 978-0-12-374857-7 (Second edition)
  - Materiale fornito dal Professore
- **http://cloudone.isti.cnr.it/casarosa/BDG/**

Modules

- Computer Fundamentals and Networking
- A conceptual model for Digital Libraries
- Bibliographic records and metadata
- Information Retrieval and Search Engines
- Knowledge representation
- Digital Libraries and the Web
- Hands-on laboratory: the Greenstone system

Representation of words (and documents)

- In "traditional" Information Retrieval, documents are represented as "Bag of Words"
  - It means that no information is retained about the "context" of the word
- In a given corpus, each word can be represented as a "one-hot vector", i.e. a vector as long as the lexicon with just one 1 in the position of the word and zeros in all other positions
  - Very long vectors (hundred of thousands of elements, all of them zeros except one)
- With Word embedding we move to short and dense vectors (hundreds of elements and no zeros), capturing the information provided by the context of the word

Document as vectors of term frequency

lexicon

documents

d	col	day	eat	hot	lot	nin	old	pea	por	pot
	Document vectors $\langle w_{d,t} \rangle$									
1	1	0	0	1	0	0	0	2	2	0
2	0	0	0	0	0	0	0	1	1	1
3	0	1	0	0	0	1	1	0	0	0
4	1	0	0	1	0	0	0	0	0	2
5	0	0	0	0	0	0	0	2	2	0
6	0	0	1	0	1	0	0	0	0	0
eat	0	0	1	0	0	0	0	0	0	0
hot porridge	0	0	0	1	0	0	0	0	1	0

each document (and each query) is now represented as a sequence (a vector) of zeros and numbers, i.e. each number is the number of occurrences of the term in the document

As before, the number of components of the vectors is equal to the size of the lexicon

Final weight: tf x idf (or tf.idf)

- In conclusion, the weight of each term *i* in each document *d* ($w_{i,d}$) is usually given by the following formula (or very similar variations), called the *tf.idf* weight

$$w_{i,d} = tf_{i,d} \times \log(n / df_i)$$

$$tf_{i,d} = \text{frequency of term } i \text{ in document } d$$
$$n = \text{total number of documents}$$
$$df_i = \text{the number of documents that contain term } i$$

- Increases with the number of occurrences *within* a doc
- Increases with the rarity of the term *across* the whole corpus

Word embedding

- In “traditional” Information Retrieval, documents are represented as “Bag of Words”
  - It means that no information is retained about the “context” of the word
- In a given corpus, each word can be represented as a “one-hot vector”, i.e. a vector as long as the lexicon with just one 1 in the position of the word and zeros in all other positions
  - Very long vectors (hundred of thousands of elements, all of them zeros except one)
- With Word embedding we move to short and dense vectors (hundreds of elements and no zeros), capturing the information provided by the context of the word

How to learn such Embedding ?

- Use context information

…he curtains open and the moon shining in on the barely…

…ars and the cold , close moon " . And neither of the w…

…rough the night with the moon shining so brightly , it…

…made in the light of the moon . It all boils down , wr…

…surely under a crescent moon , thrilled by ice-white…

…sun , the seasons of the moon ? Home , alone , Jay pla…

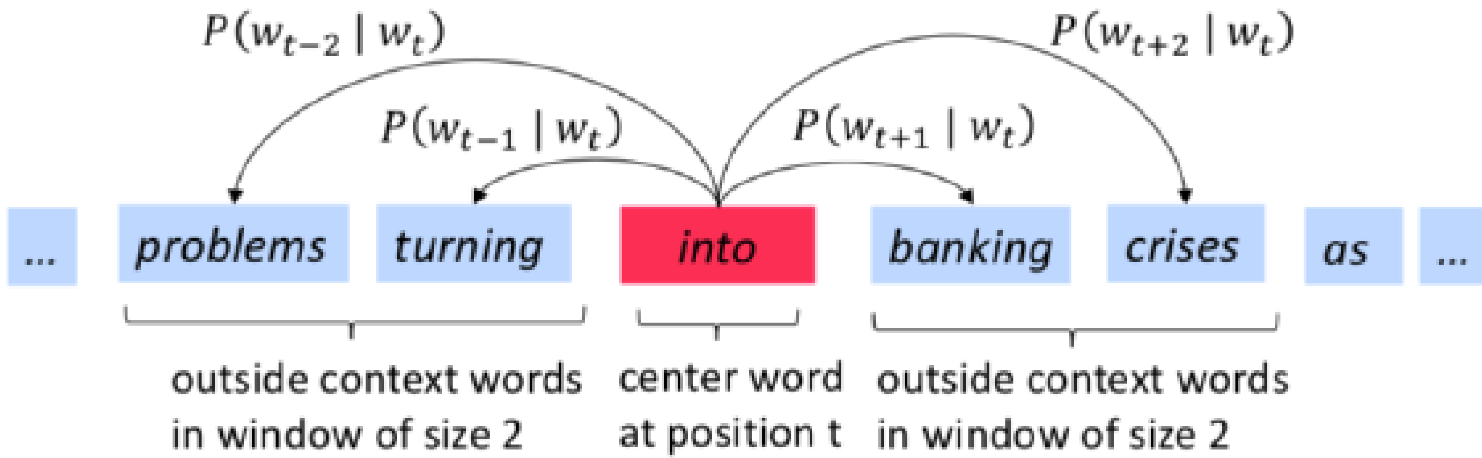
…m is dazzling snow , the moon has risen full and cold…

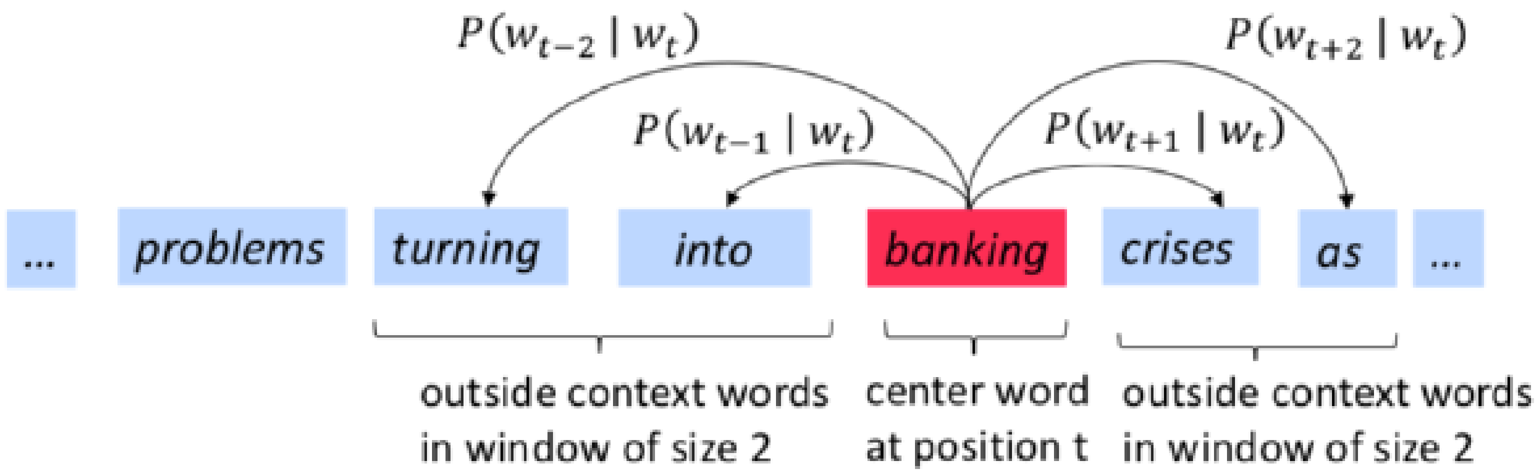
…un and the temple of the moon , driving out of the hug…

…in the dark and now the moon rises , full and amber a…

…bird on the shape of the moon over the trees in front…

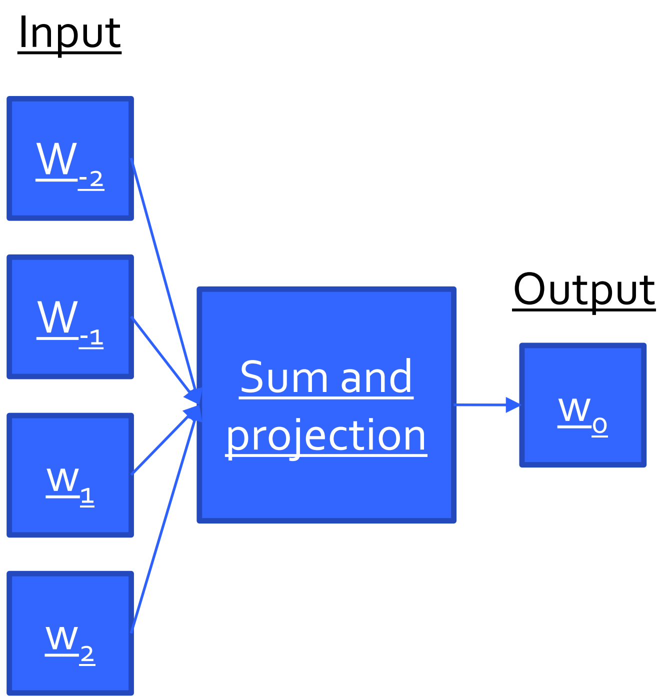
Words in context



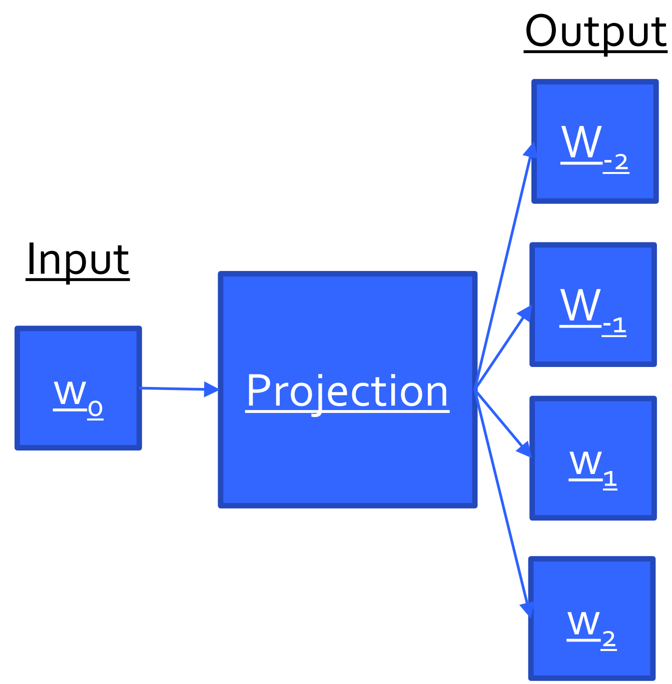


Word2Vec
main context representation models

Continuous Bag of Words (CBOW)

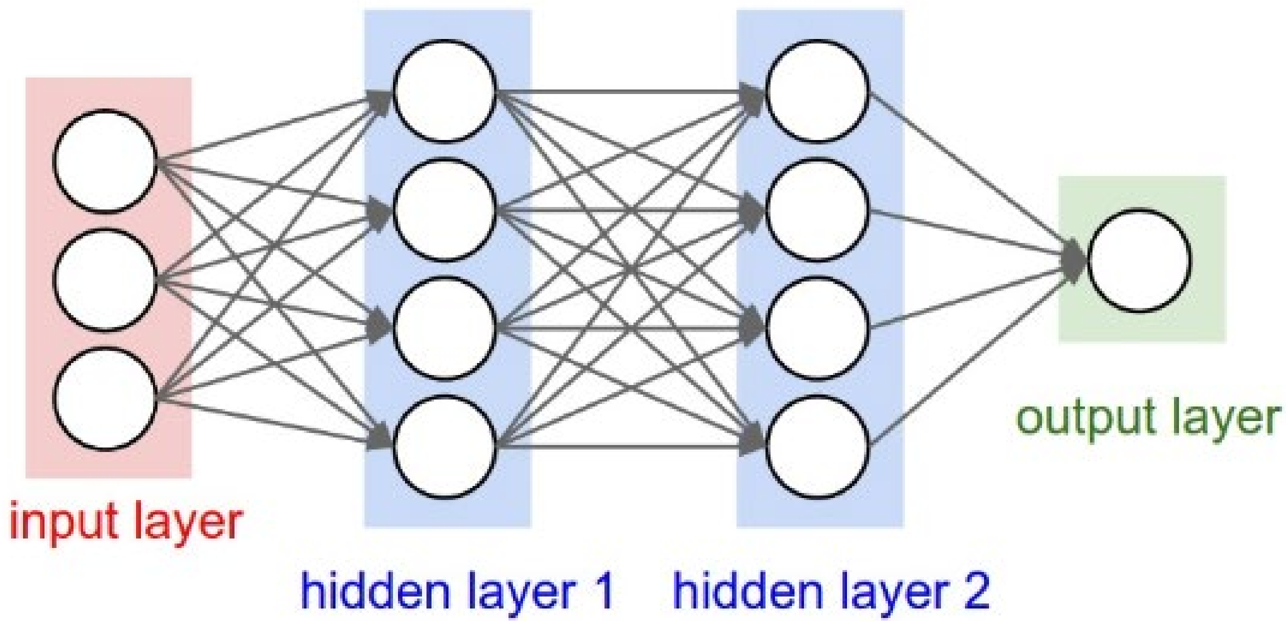


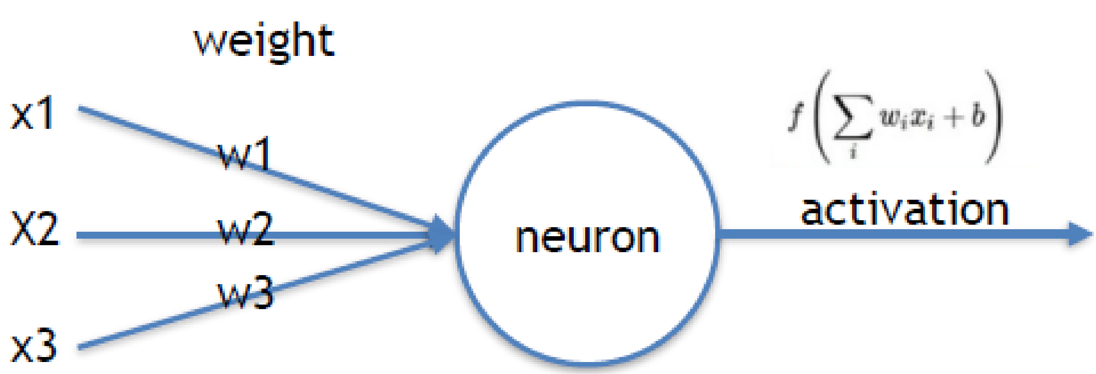
Skip-Ngram



Distributed Representations of Words and Phrases and their Compositionality, Mikolov et al, 2013

Neural networks

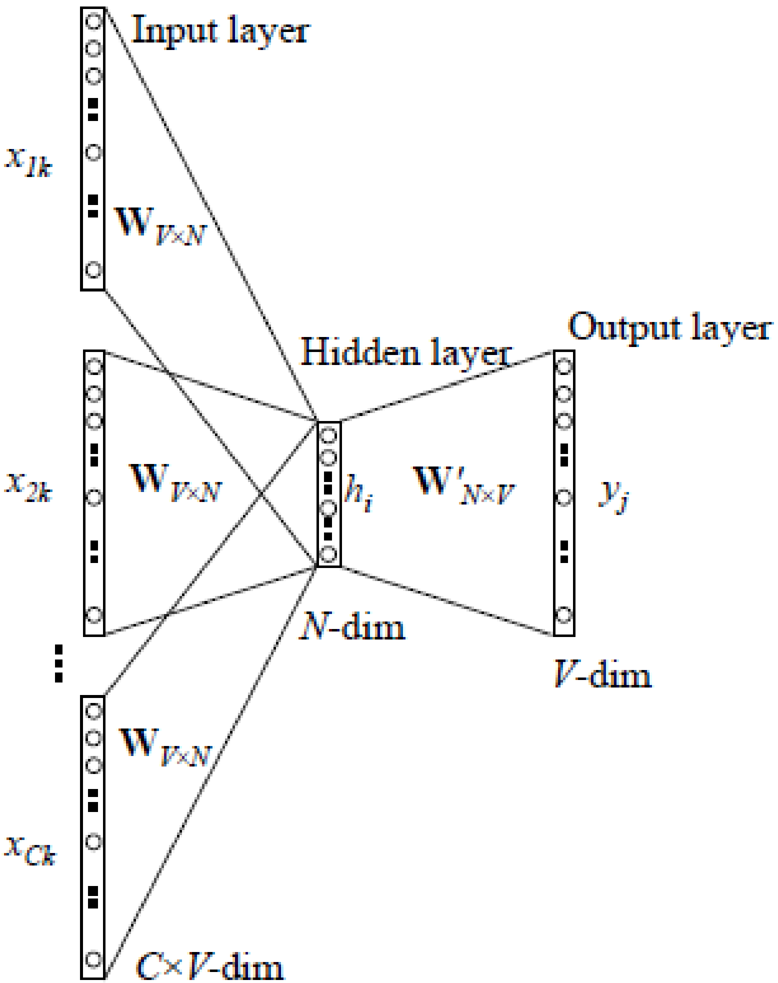


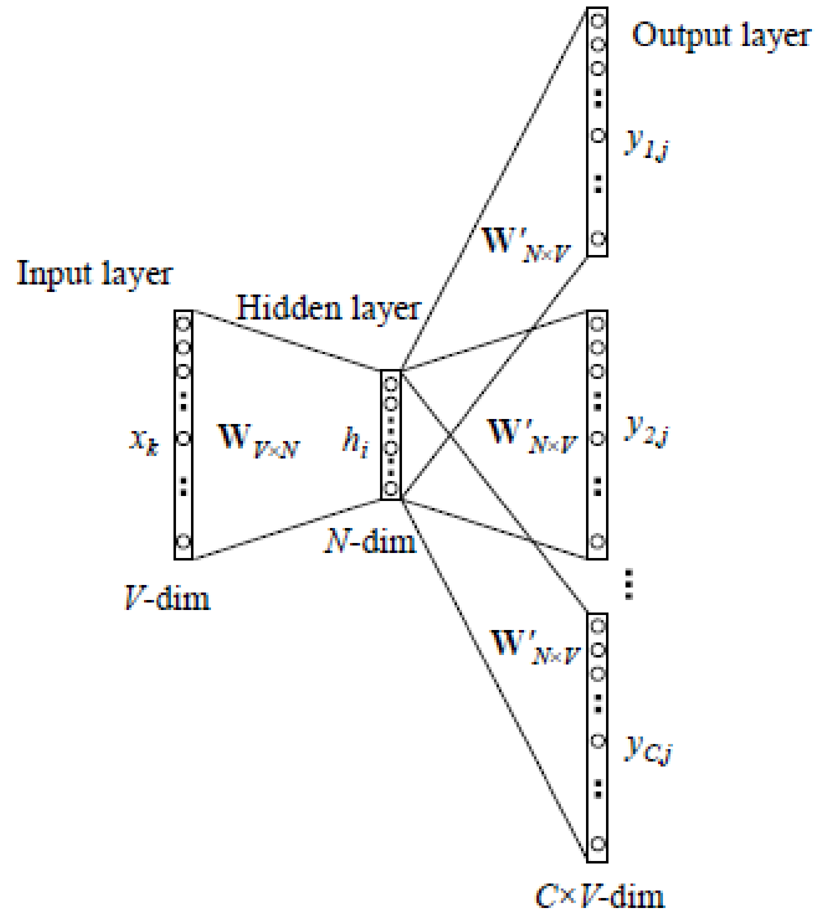


Word2Vec neural network

- Input layer one input for each word in the lexicon (one-hot vector)
- Output layer one node for each word in the lexicon, giving the probability for that word to be part of the context of the input
- Only one hidden layer
- The number of nodes of the hidden layer is the dimension of the vectors representing the words

CBOW and Skipgram



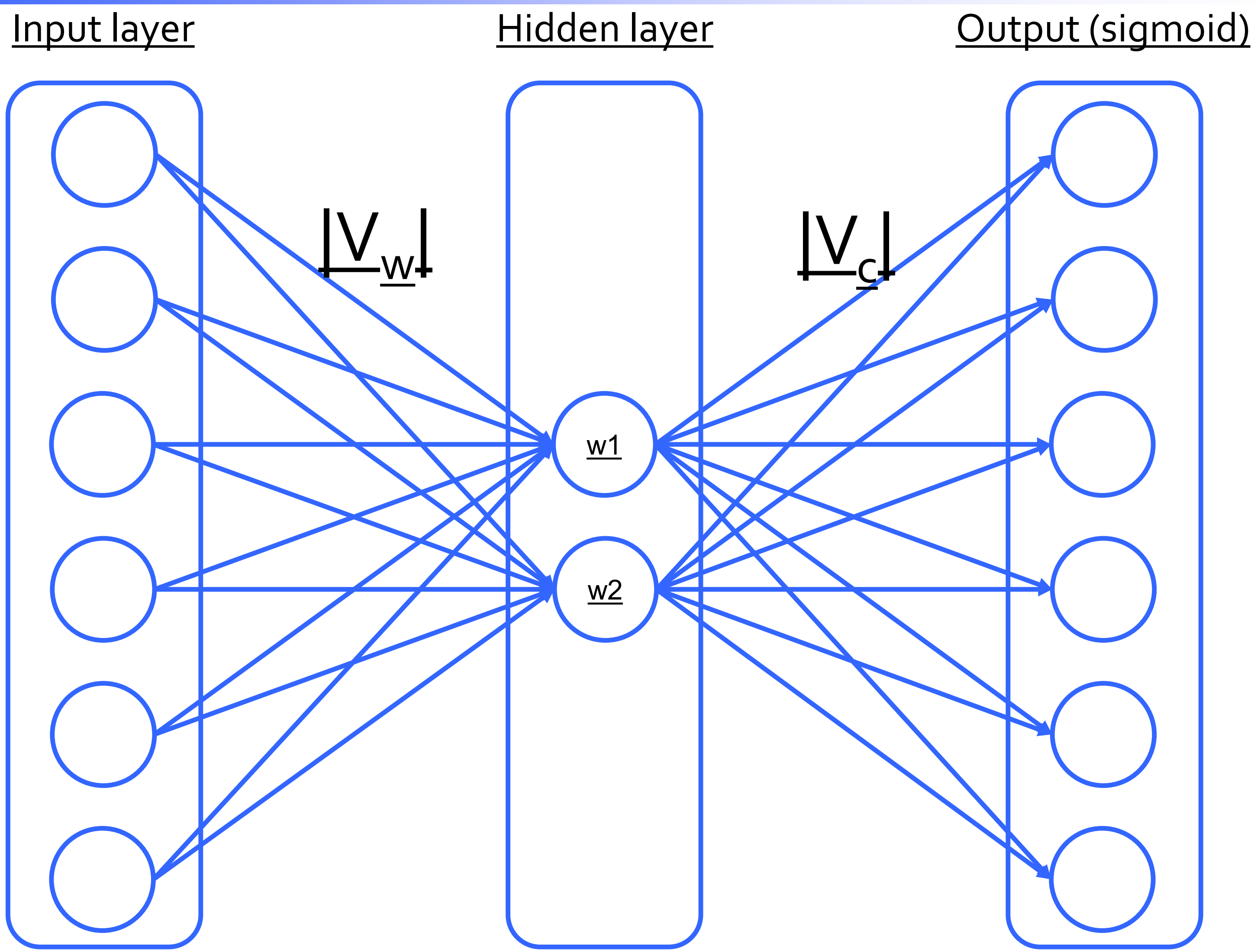


Word2Vec: Data generation (window size = 2)

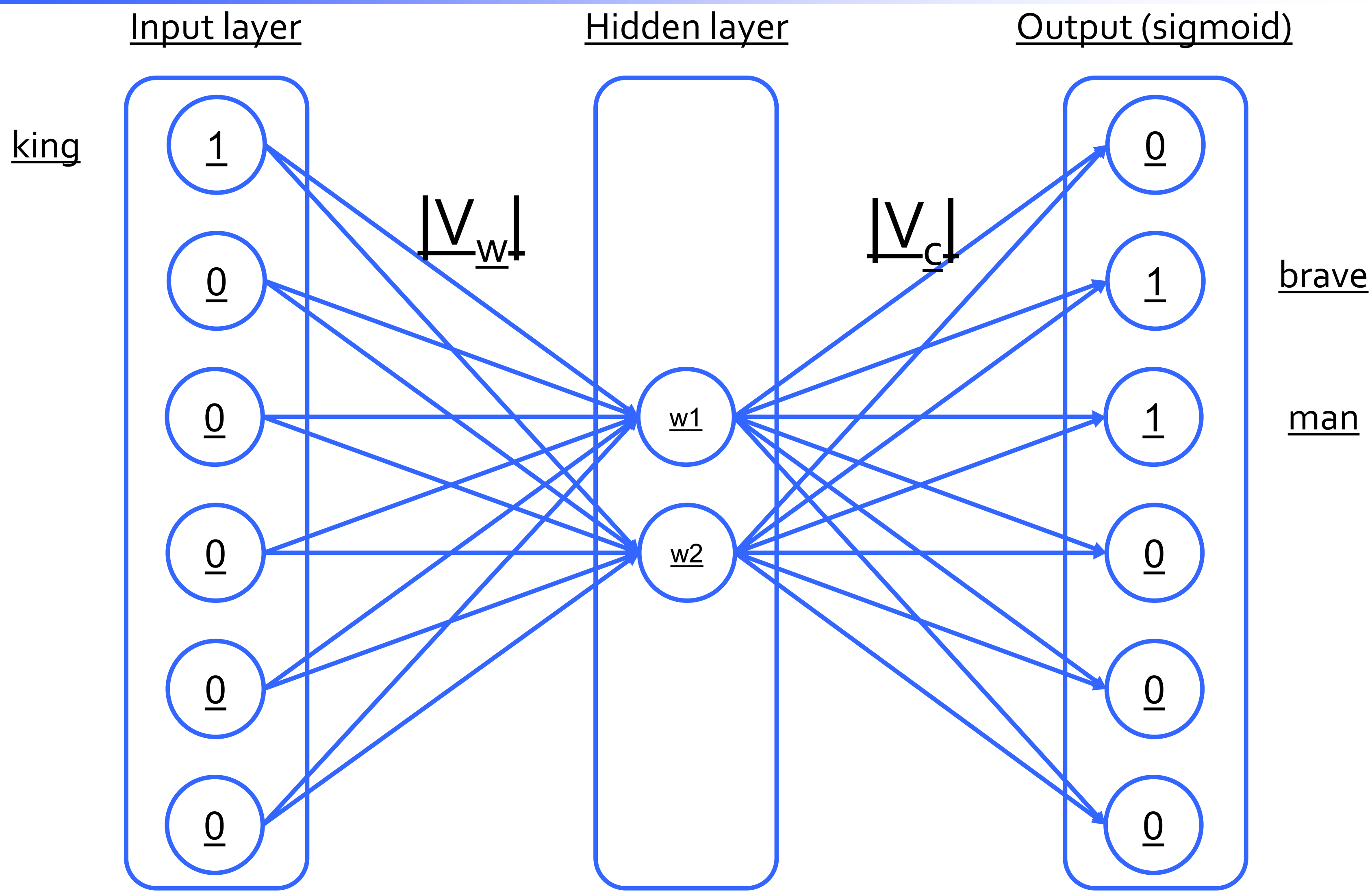
Example: d1 = “king brave man” , d2 = “queen beautiful women”

word	Word one hot encoding	neighbor	Neighbor one hot encoding
king	[1,0,0,0,0,0]	brave man	[0,1,1,0,0,0]
brave	[0,1,0,0,0,0]	king man	[1,0,1,0,0,0]
man	[0,0,1,0,0,0]	king brave	[1,1,0,0,0,0]
queen	[0,0,0,1,0,0]	beautiful women	[0,0,0,0,1,1]
beautiful	[0,0,0,0,1,0]	queen women	[0,0,0,1,0,1]
woman	[0,0,0,0,0,1]	queen beautiful	[0,0,0,1,1,0]

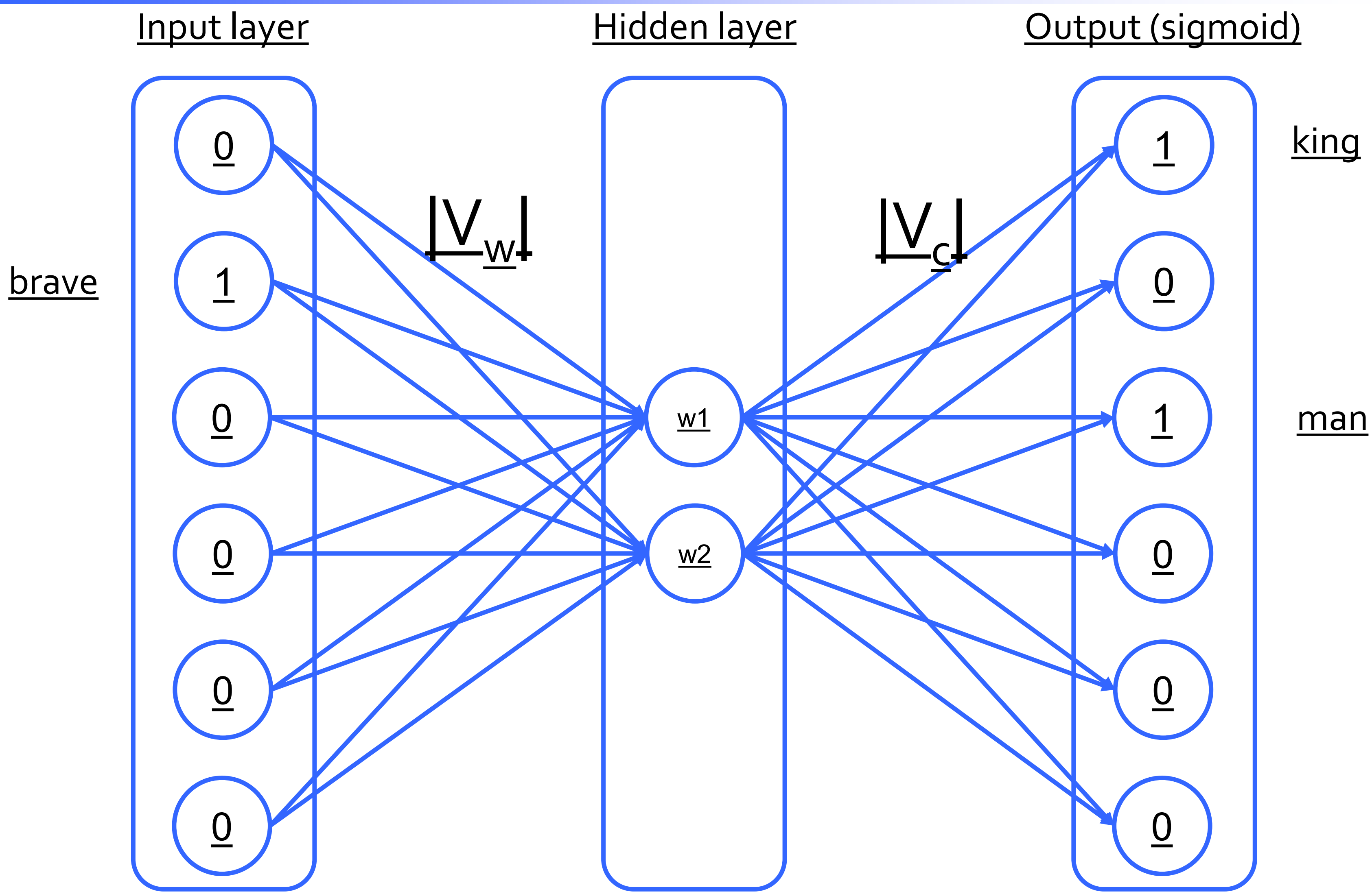
Word2Vec : Neural Network representation



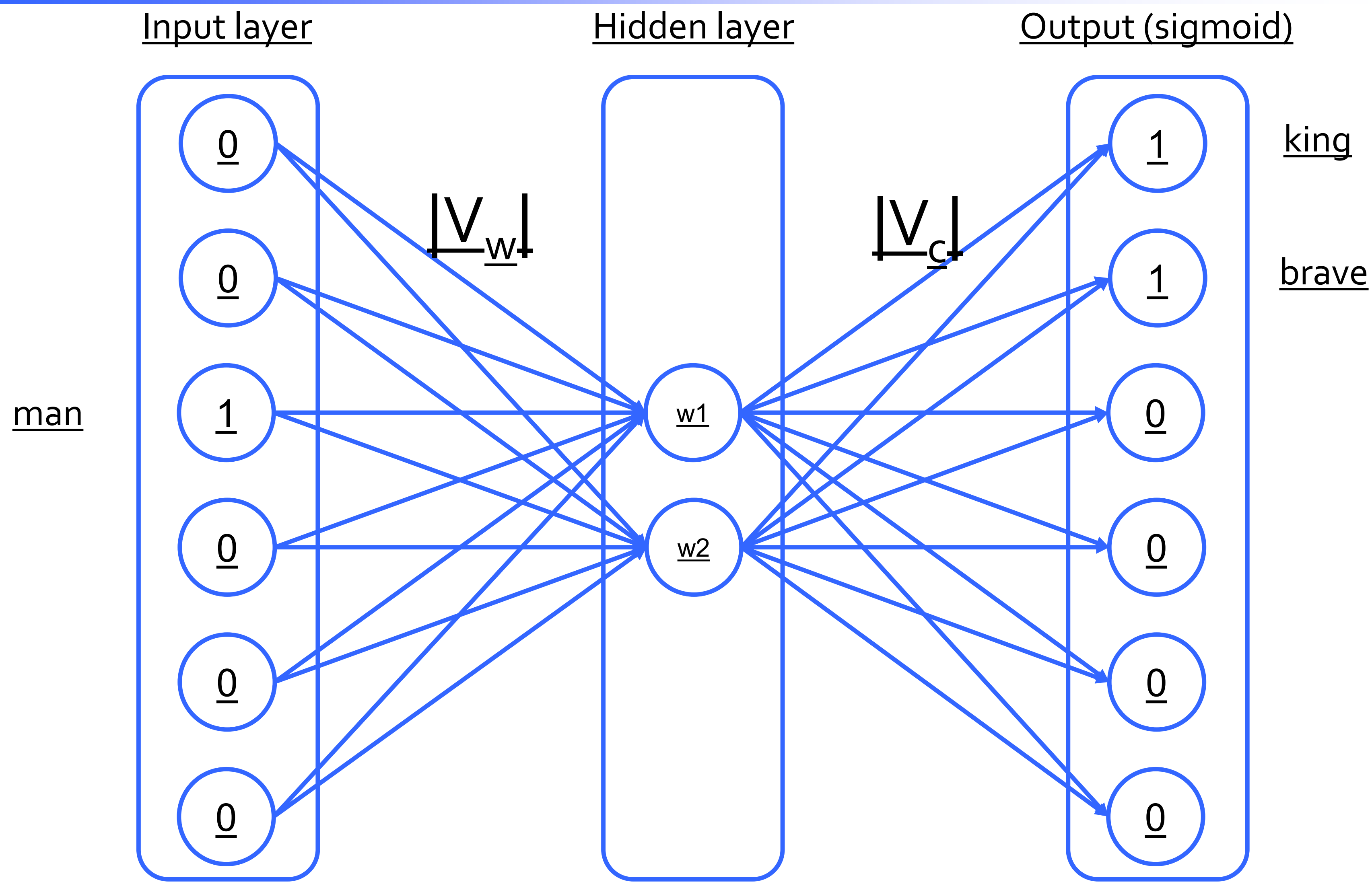
Word2Vec : Neural Network representation



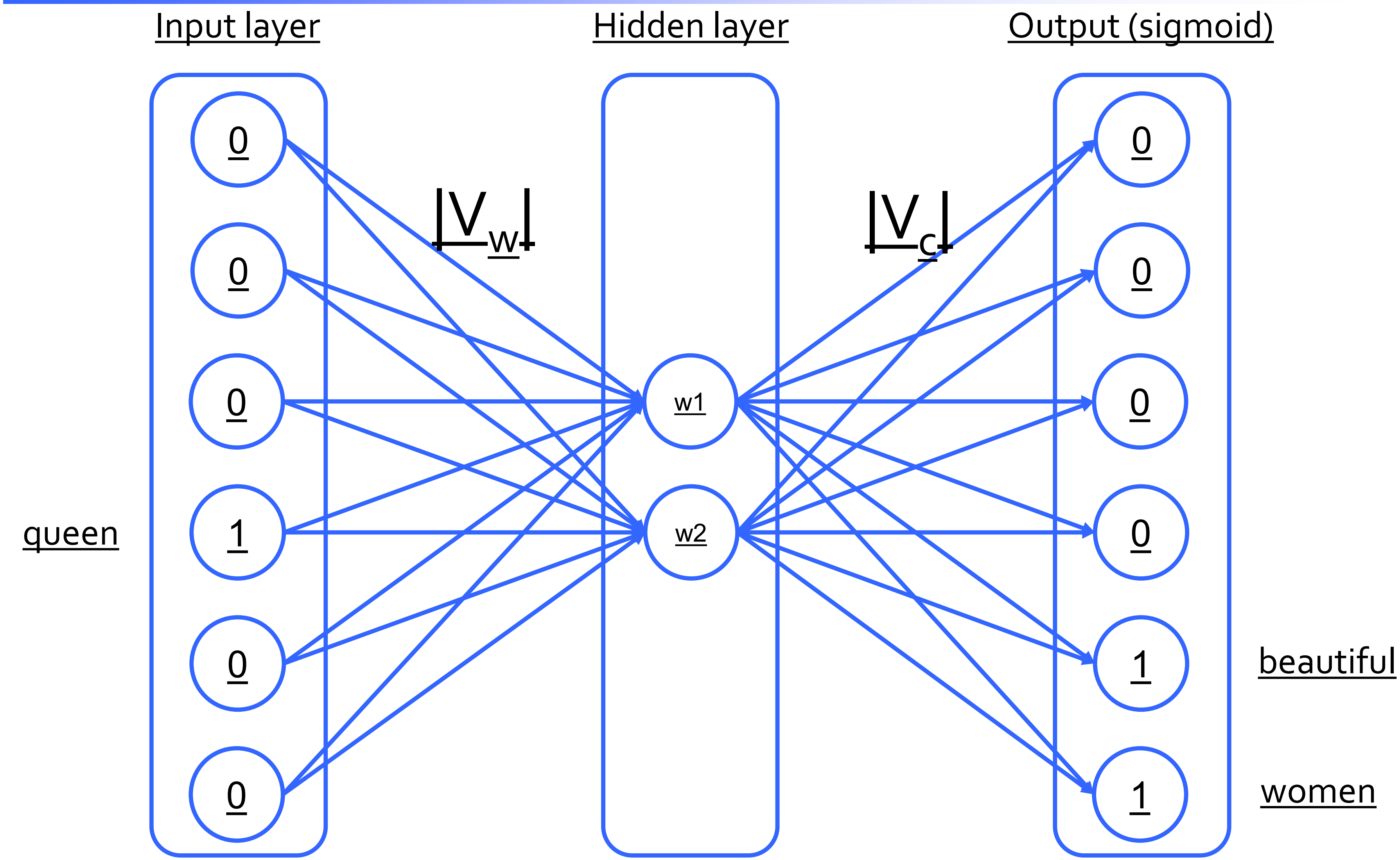
Word2Vec : Neural Network representation



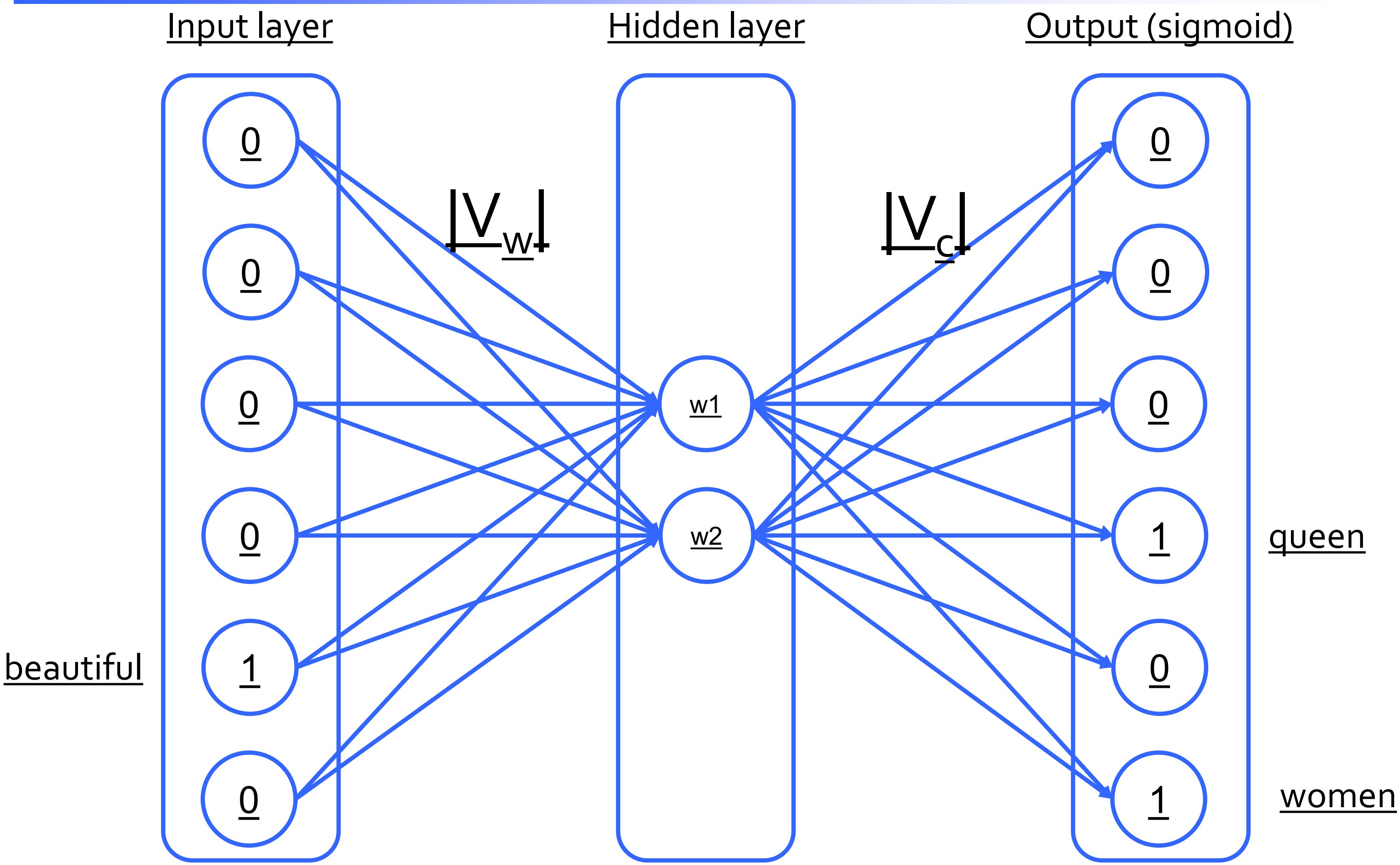
Word2Vec : Neural Network representation



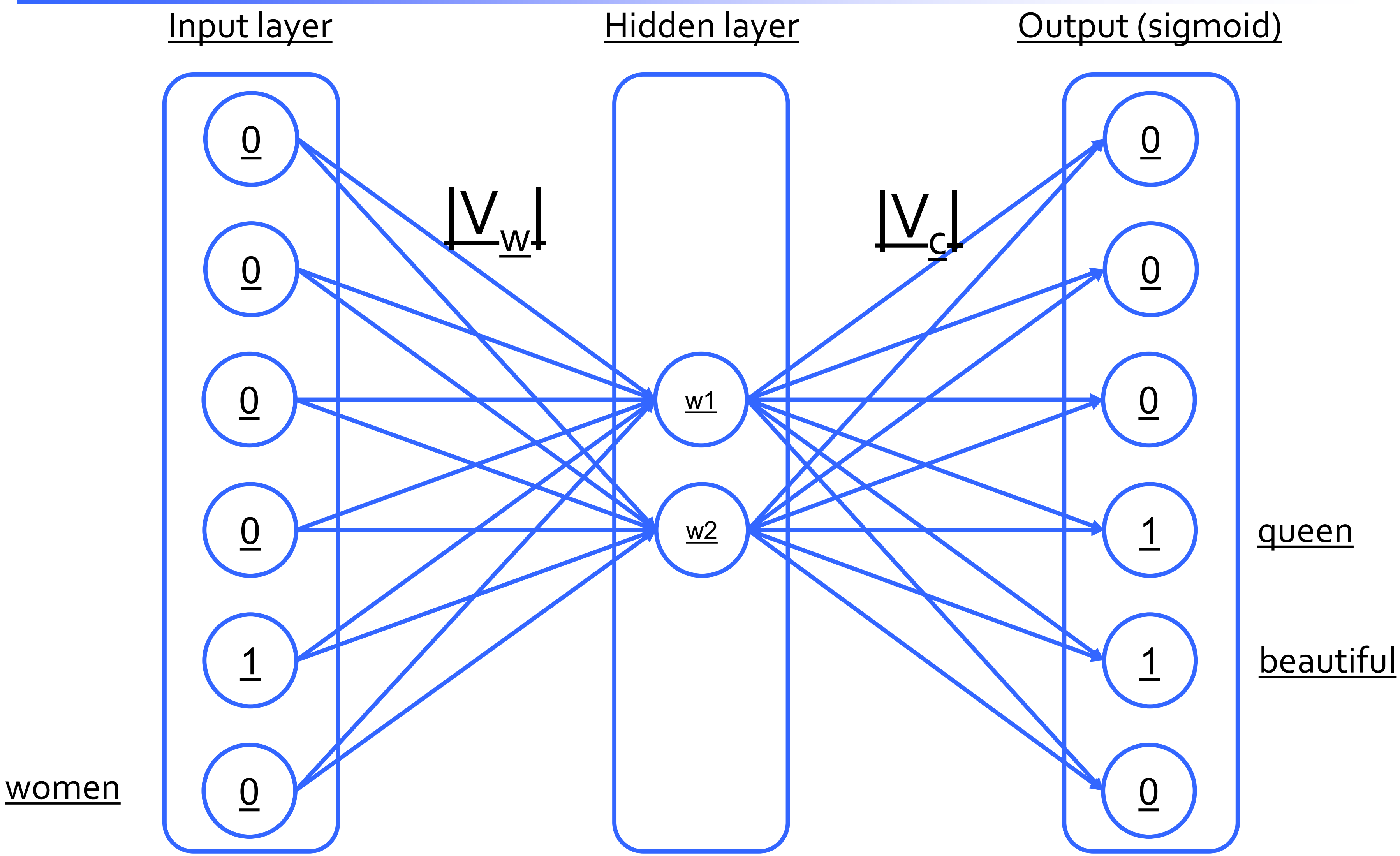
Word2Vec : Neural Network representation



Word2Vec : Neural Network representation



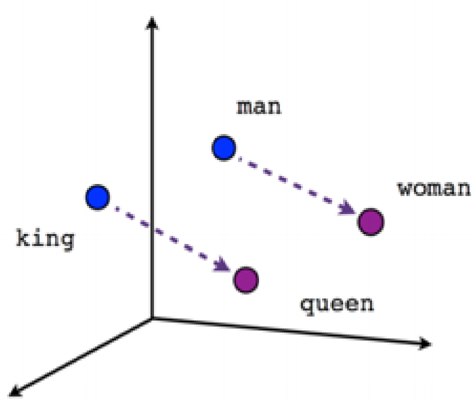
Word2Vec : Neural Network representation



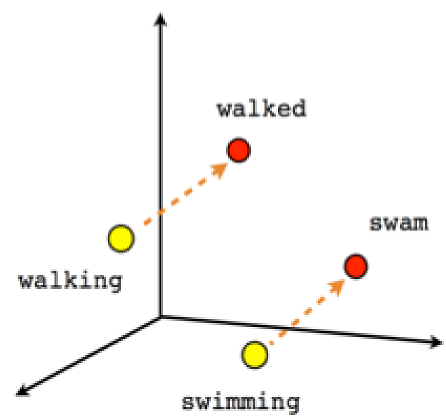
Word2Vec vector representation

- The outout node should give the probability for each word in the lexicon to be part of the context of the word(s) provided in input
- During learning, the parameters of the neural network are adjusted in order to increase the probability of the words in the context
- The parameters being adjusted are the weights of the input layer and/or the weights of the hidden layer
- At the end of the learning process, those weights represent the vector representation of the words in the lexicon
- This process clusters the words with similar meaning in the multimensional space defined by the number of components of the vectors representing the words

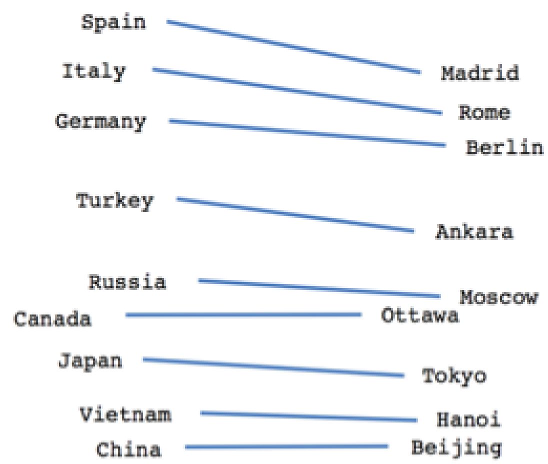
Example



Male-Female



Verb tense



Country-Capital

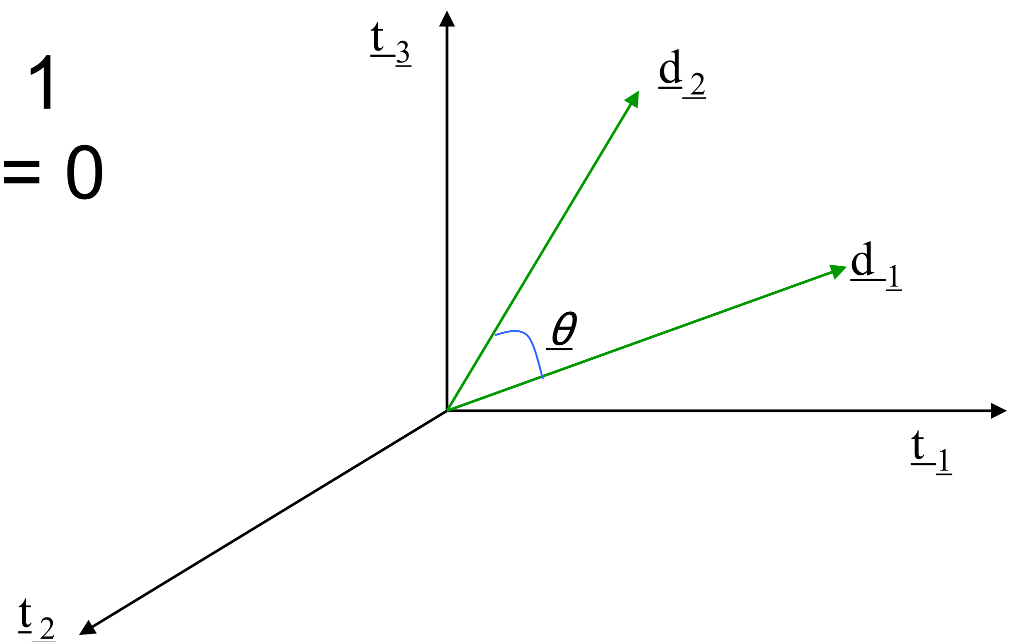
- vector[Queen] ≈ vector[King] - vector[Man] + vector[Woman]
- vector[Italy] ≈ vector[France] - vector[Paris] + vector[Rome]
- vector[Paris] ≈ vector[France] - vector[Italy] + vector[Rome]
  - This can be interpreted as “France is to Paris as Italy is to Rome”.

Similarity in vector space

- Similarity between vectors d_1 and d_2 is *captured* by the **cosine** of the angle *x* between them.
- Note – this is *similarity*, not distance

$\cos 0° = 1$

$\cos 90° = 0$



Relations Learned by Word2Vec

- A relation is defined by the vector displacement in the first column. For each start word in the other column, the closest displaced word is shown.

Relationship	Example 1	Example 2	Example 3
France - Paris	Italy: Rome	Japan: Tokyo	Florida: Tallahassee
big - bigger	small: larger	cold: colder	quick: quicker
Miami - Florida	Baltimore: Maryland	Dallas: Texas	Kona: Hawaii
Einstein - scientist	Messi: midfielder	Mozart: violinist	Picasso: painter
Sarkozy - France	Berlusconi: Italy	Merkel: Germany	Koizumi: Japan
copper - Cu	zinc: Zn	gold: Au	uranium: plutonium
Berlusconi - Silvio	Sarkozy: Nicolas	Putin: Medvedev	Obama: Barack
Microsoft - Windows	Google: Android	IBM: Linux	Apple: iPhone
Microsoft - Ballmer	Google: Yahoo	IBM: McNealy	Apple: Jobs
Japan - sushi	Germany: bratwurst	France: tapas	USA: pizza