APAN PS5430
Applied Text & Natural Language Analytics
Week 6: Vector Space Modeling using Neural Networks

Javid Huseynov, Ph.D.
Thursday, February 27, 2020

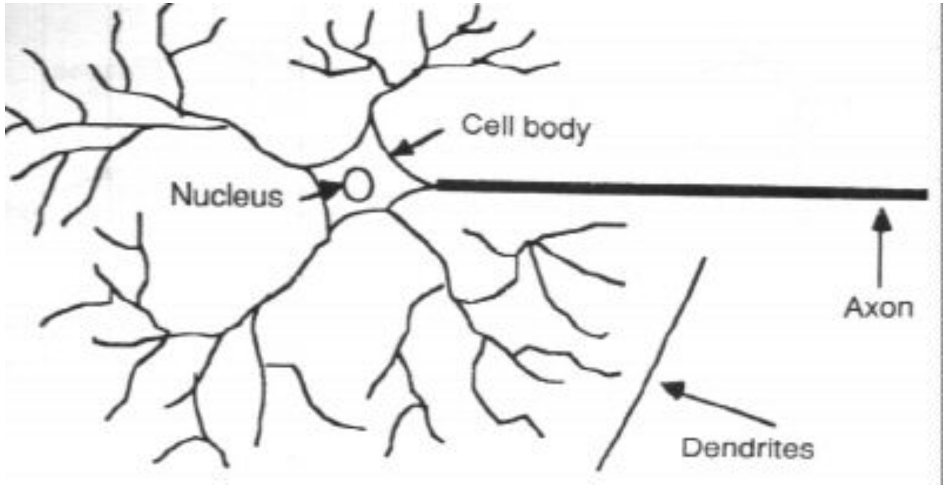
COLUMBIA UNIVERSITY

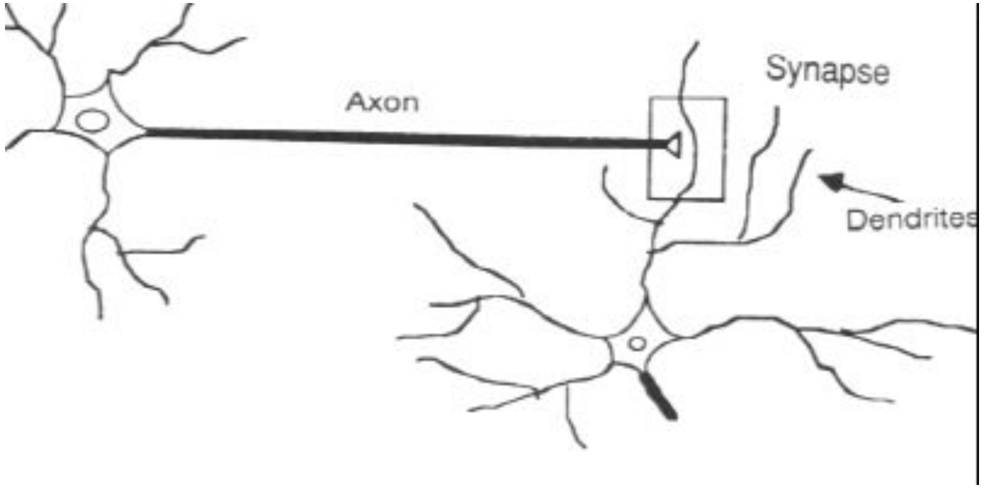
Week 6 Agenda

- Neural Networks Overview
- Vector-Space Model
- Sentence Vectors
- Word Vectors: Concept
- Co-occurrence Matrix
- Singular Value Decomposition (SVD)
- Word2Vec
- Class Exercises

Neural network: Bio inspiration

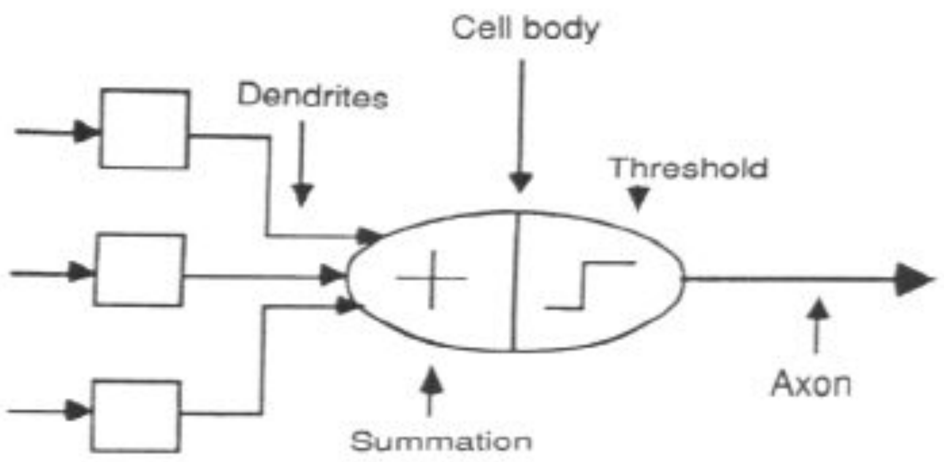
- Almost all living species can learn and react to changes in their environment using their nervous system

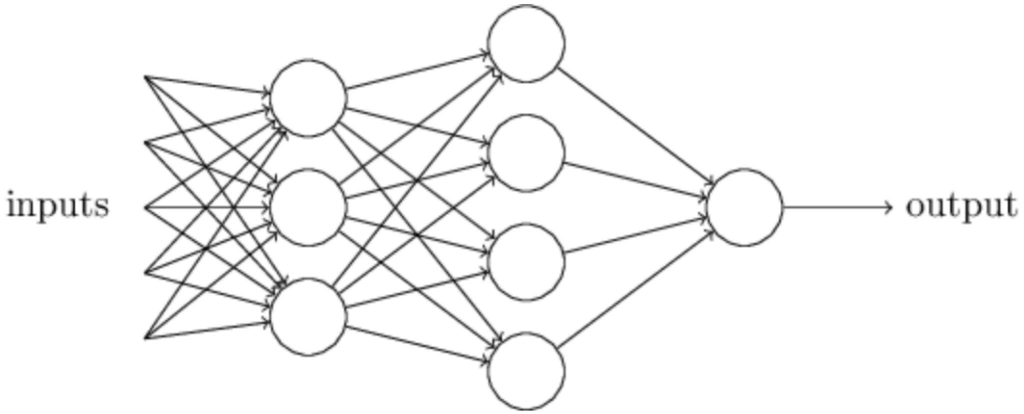




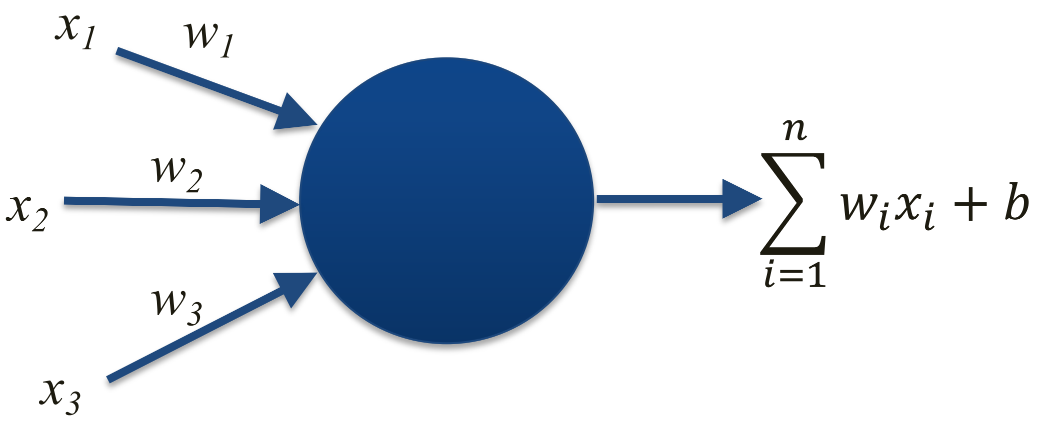
- An artificial mathematical model can reproduce the behavior of the central nervous system

From human to artificial neurons





- Simple Neuron – Perceptron Model
- (Rosenblatt 1958)

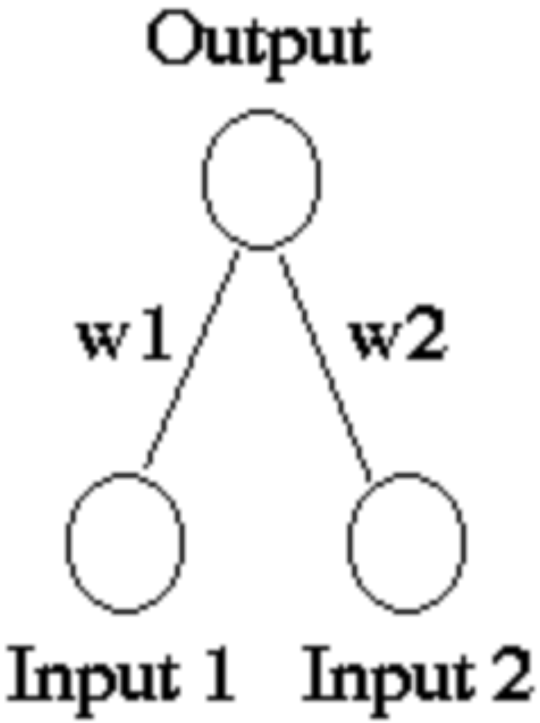


$$\text{output} = \begin{cases} 0 & \text{if } \sum_j w_j x_j \leq \text{ threshold} \\ 1 & \text{if } \sum_j w_j x_j > \text{ threshold} \end{cases}$$

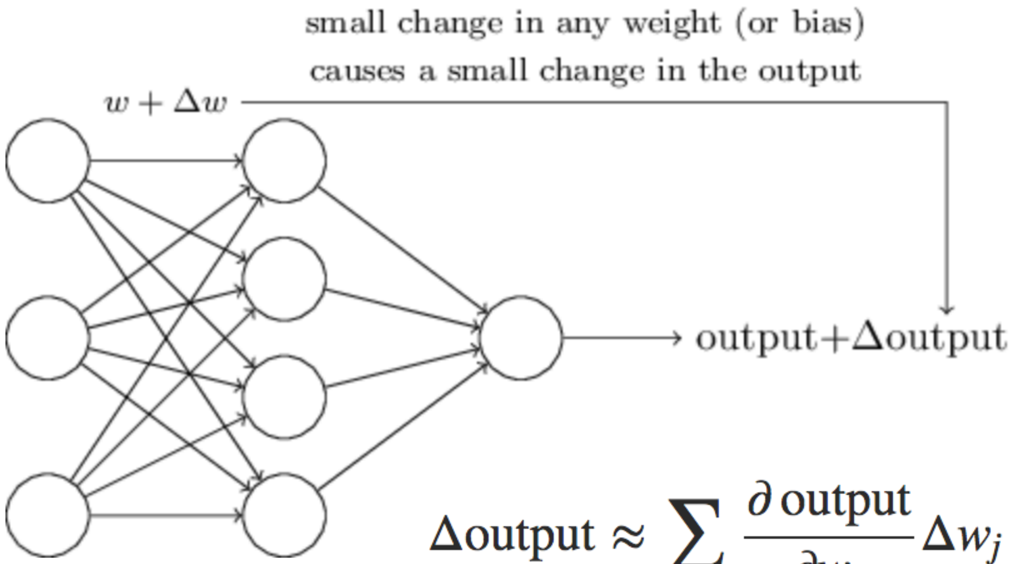
2-layer Perceptron Limitation: Exclusive-OR (XOR) problem

- Minsky & Pappert 1969

Input 1	Input 2	Output
1	1	0
1	0	1
0	1	1
0	0	0

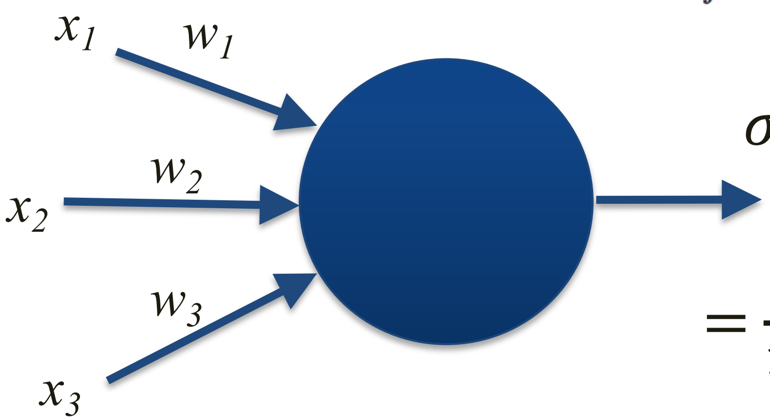


From Perceptron to Activation Function



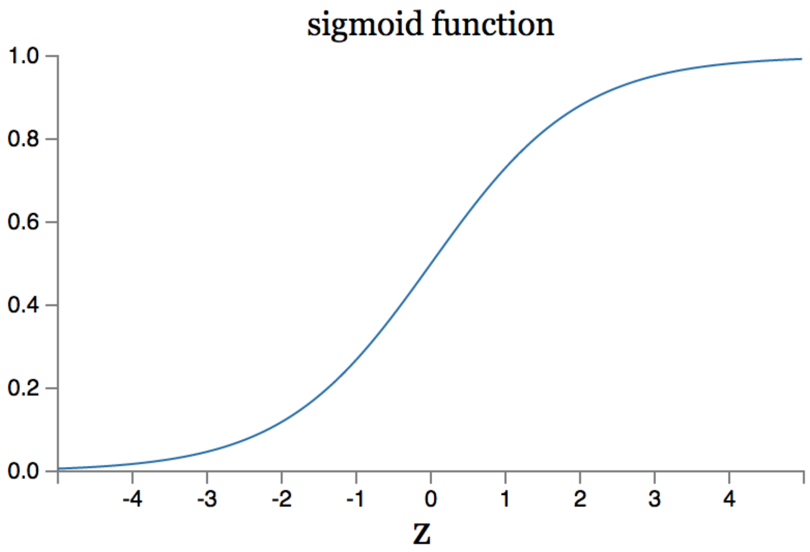
- Set of small changes in connection weights can alter the output significantly
- To smooth this impact, sigmoid activation function is used

$$\Delta \text{output} \approx \sum_j \frac{\partial\, \text{output}}{\partial w_j} \Delta w_j + \frac{\partial\, \text{output}}{\partial b} \Delta b,$$

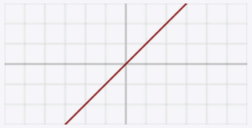
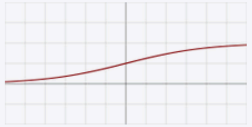
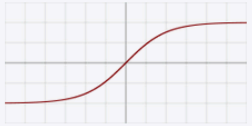
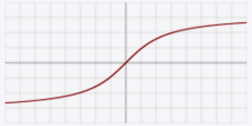
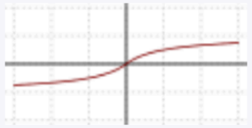


$$\sigma(z) = \frac{1}{1+e^{-z}} =$$

$$= \frac{1}{1+\exp(-\sum_{i=1}^{n} w_i x_i - b)}$$



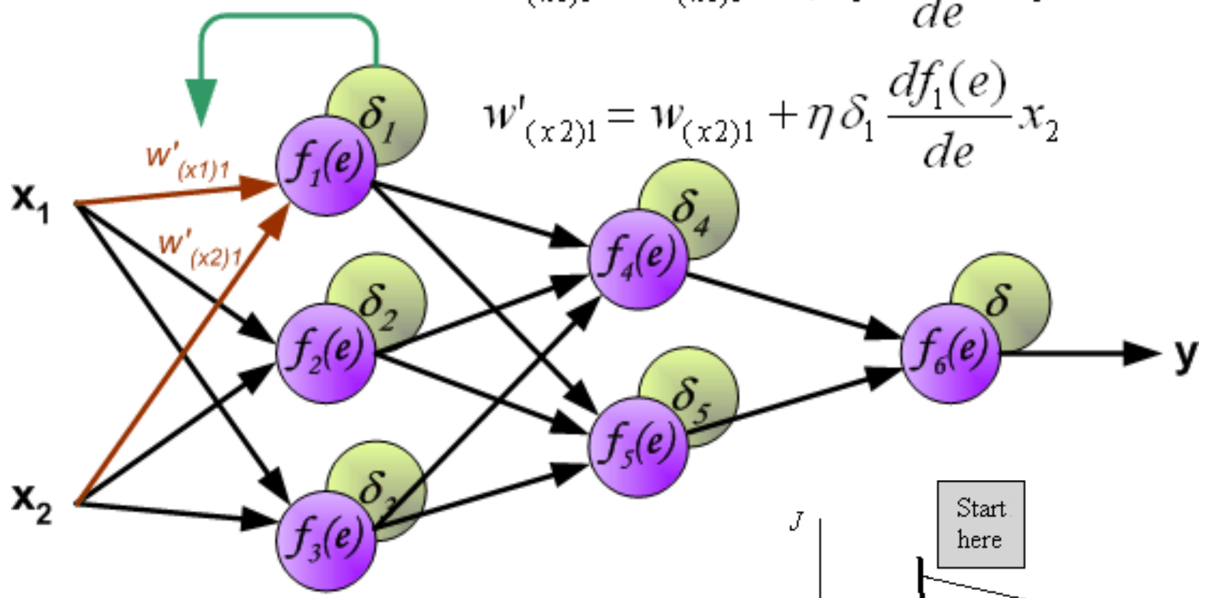
Types of Activation Functions

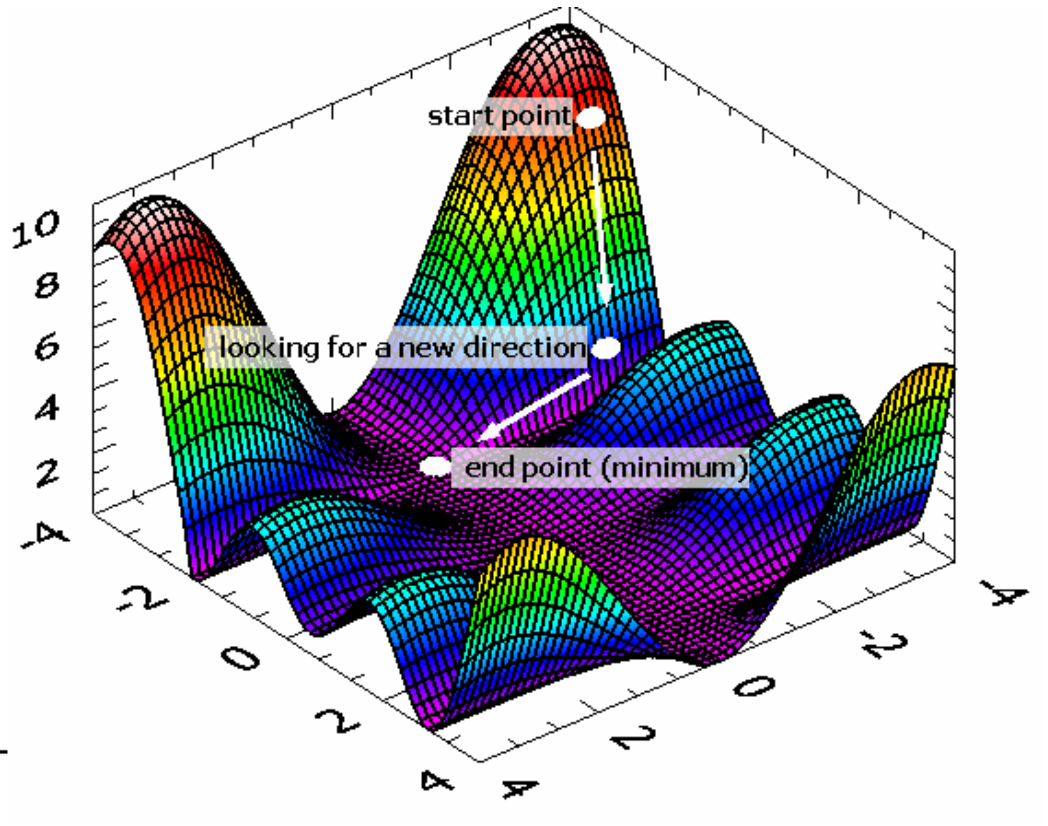
Name Sort ascending	Plot	Equation	Derivative (with respect to *x*)	Range
Identity		$f(x) = x$	$f'(x) = 1$	$(-\infty, \infty)$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$	$\{0, 1\}$
Logistic (a.k.a. Soft step)		$f(x) = \frac{1}{1 + e^{-x}}$	$f'(x) = f(x)(1 - f(x))$	$(0, 1)$
TanH		$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$	$(-1, 1)$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2 + 1}$	$\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$
Softsign [7][8]		$f(x) = \frac{x}{1 + \lvert x \rvert}$	$f'(x) = \frac{1}{(1 + \lvert x \rvert)^2}$	$(-1, 1)$
Rectified linear unit (ReLU)[9]		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$[0, \infty)$

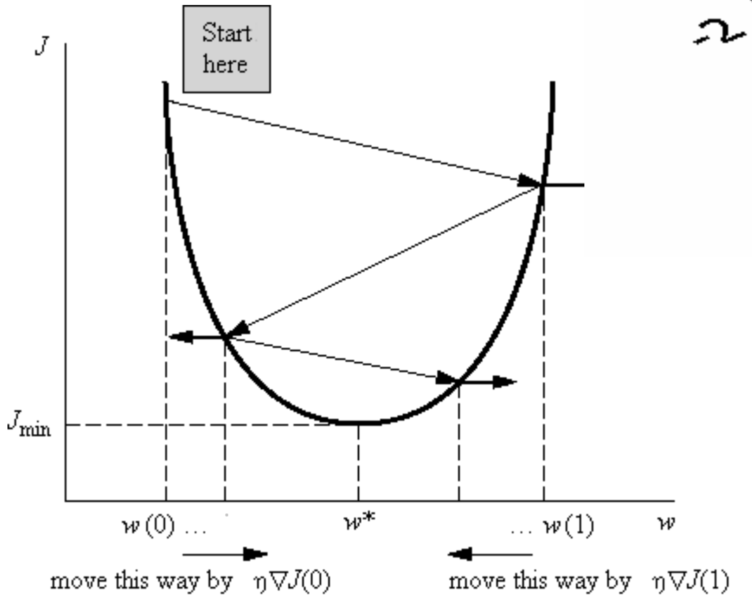
Backpropagation & conjugate gradient

$$w'_{(x1)1} = w_{(x1)1} + \eta \delta_1 \frac{df_1(e)}{de} x_1$$

$$w'_{(x2)1} = w_{(x2)1} + \eta \delta_1 \frac{df_1(e)}{de} x_2$$







Backpropagation Step-by-Step

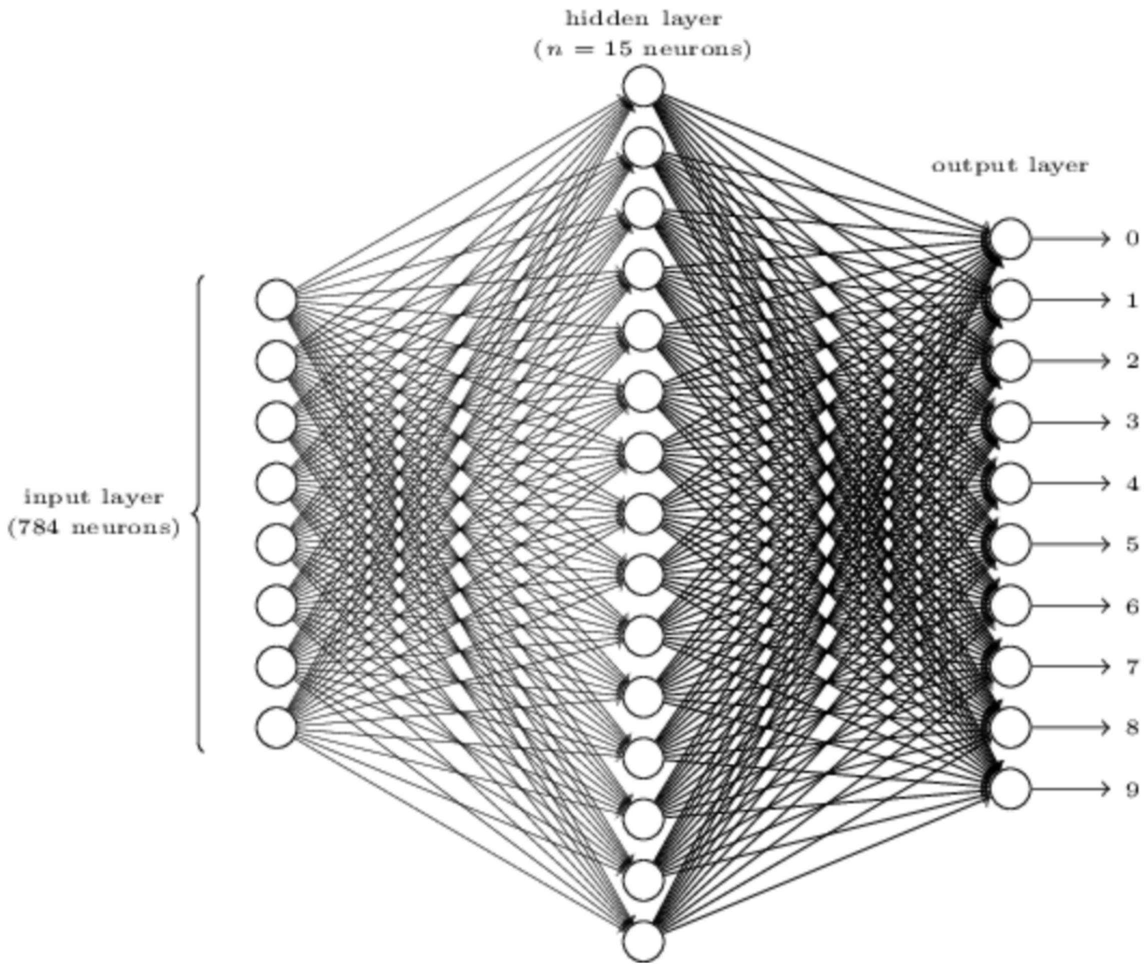
- Visit the following site to see a simple but great overview of the operation of the backpropagation algorithm with numbers:
- https://mattmazur.com/2015/03/17/a-step-by-step-backpropagation-example/

Neural Network Applications

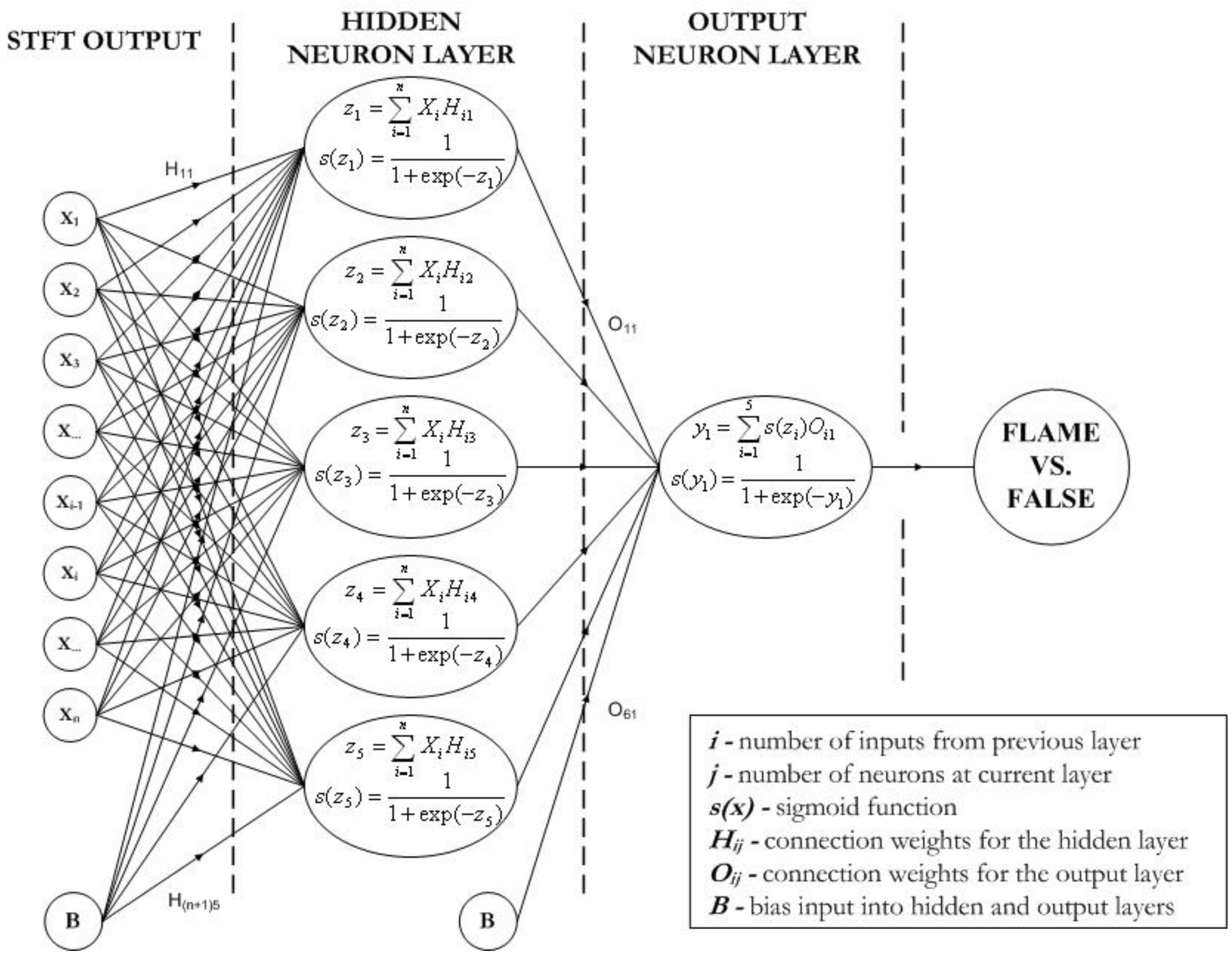
- Neural networks in pop culture

- Natural Language Processing
- Signal Processing | Time Series
- Stock Market Forecasting
- Credit Risk Assessment
- Image Processing | Character Recognition
- Speech Recognition
- Self-driving cars
- Sports Prediction
- Any other field where extra brain can be useful ☺

Use Case: Scanned Digit Recognition with Neural Networks



Another Use Case: Flame Detection using Neural Networks



VSM + Neural Networks

IR Vector-Space Model

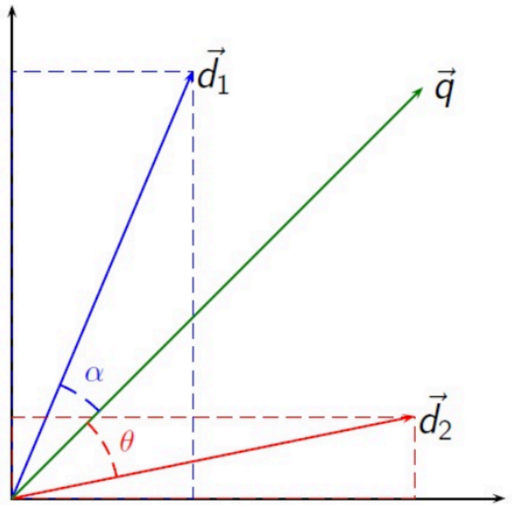
- An algebraic representation of text documents or queries as vectors

$$d_j = (w_{1,j}, w_{2,j}, \dots, w_{t,j})$$
$$q = (w_{1,q}, w_{2,q}, \dots, w_{n,q})$$

- Cosine similarity

$$\|\mathbf{q}\| = \sqrt{\sum_{i=1}^{n} q_i^2}$$

$$\cos\theta = \frac{\mathbf{d_2} \cdot \mathbf{q}}{\|\mathbf{d_2}\| \, \|\mathbf{q}\|}$$



$$\mathrm{sim}(d_j, q) = \frac{\mathbf{d_j} \cdot \mathbf{q}}{\|\mathbf{d_j}\| \, \|\mathbf{q}\|} = \frac{\sum_{i=1}^{N} w_{i,j} w_{i,q}}{\sqrt{\sum_{i=1}^{N} w_{i,j}^2} \sqrt{\sum_{i=1}^{N} w_{i,q}^2}}$$

One-Hot Encoding (Word Embedding)

- Given "`Can I eat the pizza`" of `N=5`
  1. Convert to lower case
  2. Sort the words in alphabetical order
  3. Give numerical labels to each word:
     `can:0, i:2, eat:1, the:4, pizza:3`
  4. Transform to binary vectors

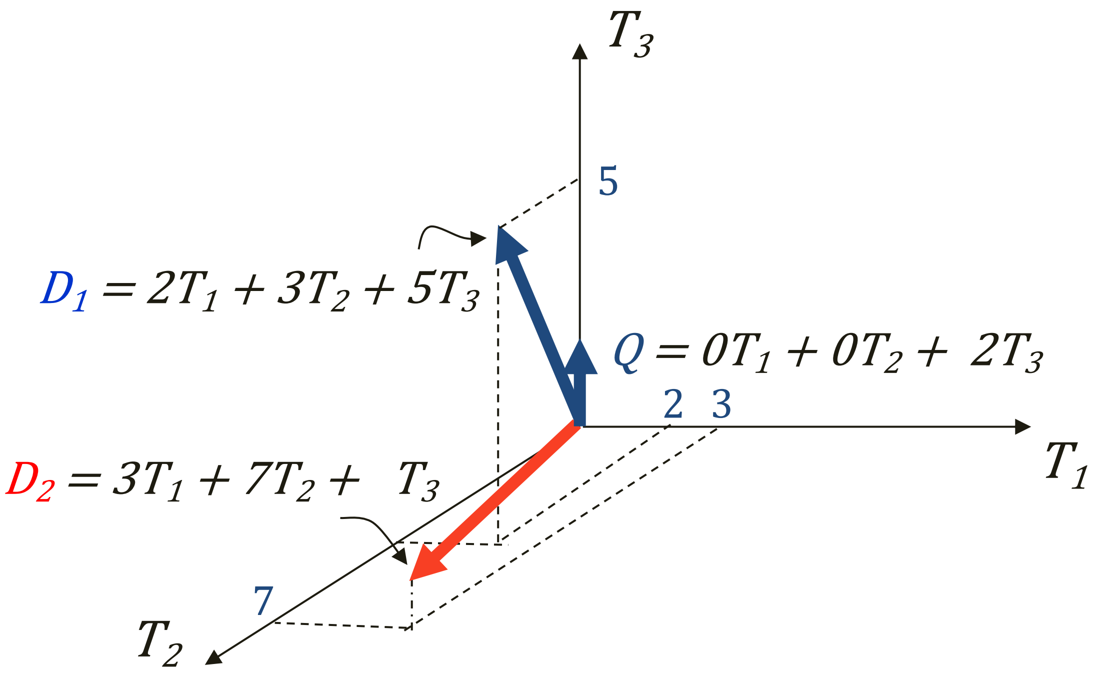
```
[[1. 0. 0. 0. 0.] #can
 [0. 0. 1. 0. 0.] #i
 [0. 1. 0. 0. 0.] #eat
 [0. 0. 0. 0. 1.] #the
 [0. 0. 0. 1. 0.]] #pizza
```

Sentence Vectors

- Collection of ***n*** documents with **m** terms can be represented in the vector space model by a term-document matrix.
- An entry in the matrix corresponds to the "weight" of a term in the document;
  - zero means the term has no significance or it simply doesn't exist in the document

$$
\begin{array}{c|cccc}
 & T_1 & T_2 & \cdots & T_m \\
\hline
D_1 & w_{11} & w_{21} & \cdots & w_{m1} \\
D_2 & w_{12} & w_{22} & \cdots & w_{m2} \\
\vdots & \vdots & \vdots & & \vdots \\
\vdots & \vdots & \vdots & & \vdots \\
D_n & w_{1n} & w_{2n} & \cdots & w_{mn}
\end{array}
$$

- *tf-idf* weighting: $\mathbf{w_{ij} = tf_{ij} * idf_i = tf_{ij} \log_2 (N / df_i)}$

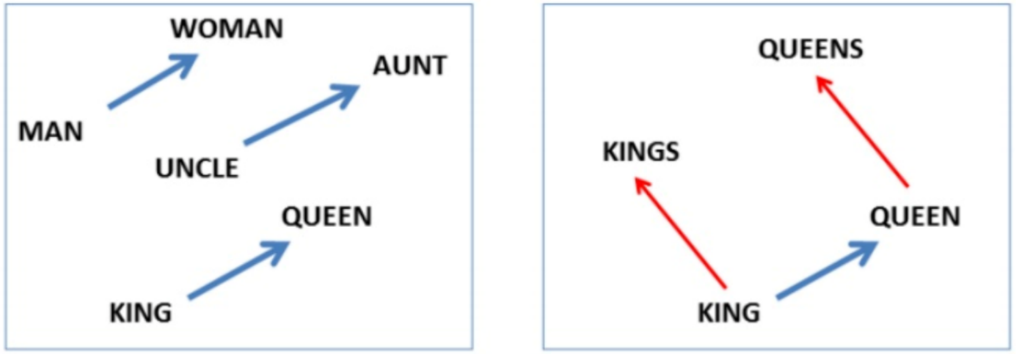


- Is D_1 or D_2 more similar to Q?
- How to measure the degree of similarity? Distance? Angle? Projection?

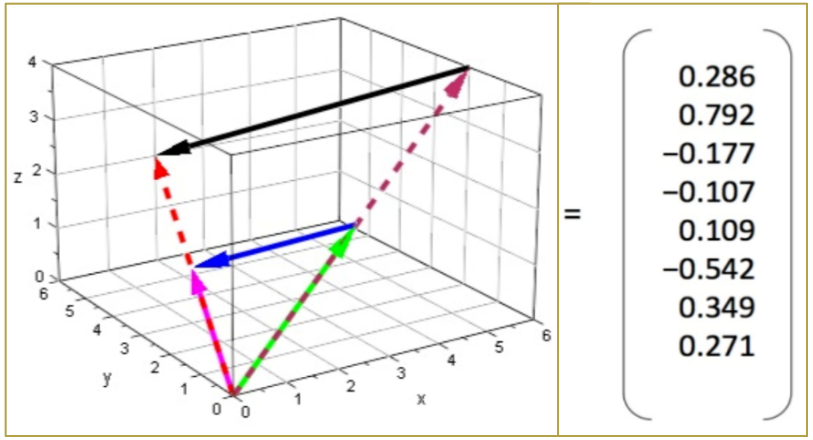
Word vectors: concept

"You shall know a word by the company it keeps"
~ J.R. Firth 1957

- Vectors are directions in space, which can also encode relationships:
  - e.g. **man** is to **woman** as **king** is to **queen**



(Mikolov et al., NAACL HLT, 2013)



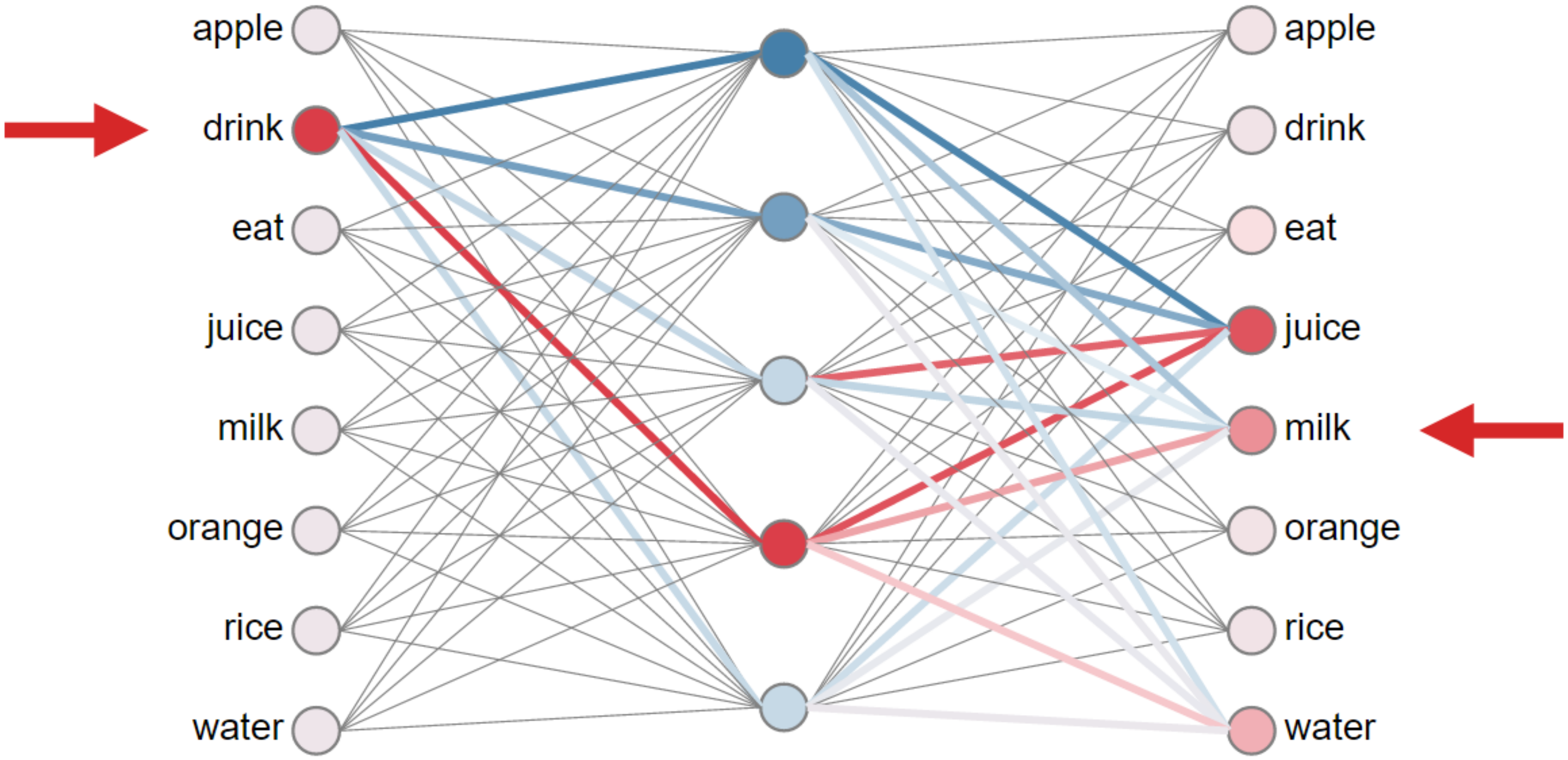
- One of the most successful ideas in modern statistical NLP
  - …government debt problem turning into banking crises as has happened in…
  - …saying that Europe needs unified banking regulation to replace the hodgepodge…
  - **surrounding words capture the context of the word *banking***

Word vectors: Learning with Neural Networks

Neural Network-based
Word Embedding
Models

Google
Word2vec

facebook
FastText



Co-occurrence Matrix

Corpus = {"I like deep learning"
"I like NLP"
"I enjoy flying"}

Context = previous word and next word

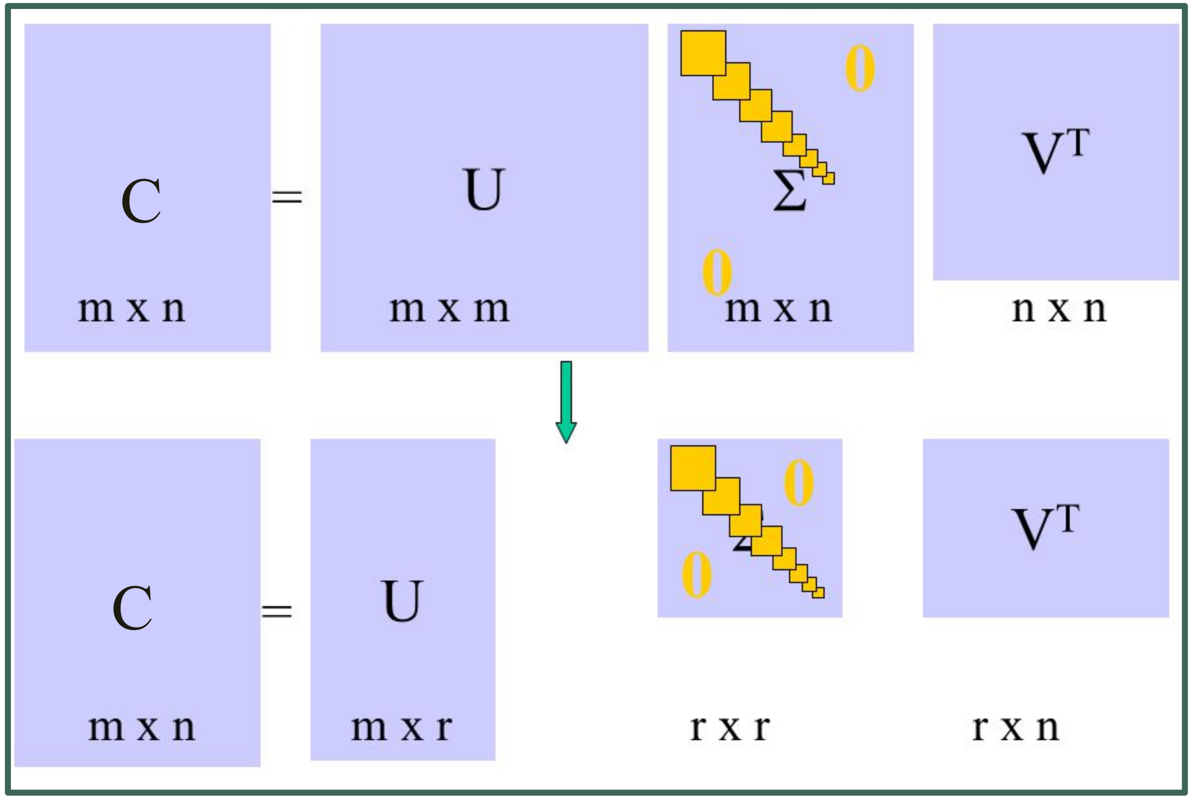
counts	I	like	enjoy	deep	learning	NLP	flying	.
I	0	2	1	0	0	0	0	0
like	2	0	0	1	0	1	0	0
enjoy	1	0	0	0	0	0	1	0
deep	0	1	0	0	1	0	0	0
learning	0	0	0	1	0	0	0	1
NLP	0	1	0	0	0	0	0	1
flying	0	0	1	0	0	0	0	1
.	0	0	0	0	1	1	1	0

Singular Value Decomposition (SVD)

- Any real ***m x n*** matrix ***C*** can be decomposed into:

  $$\mathbf{C = U\Sigma V^T}$$

- ***U*** is ***m x n,*** column orthonormal ($\mathbf{U^TU = I}$)
- $\mathbf{\Sigma}$ is ***n x n*** and diagonal:
  - $\boldsymbol{\Sigma = diag(\sigma_1, \sigma_2, \ldots, \sigma_n)}$
  - $\boldsymbol{\sigma_1}$ are called *singular* values of ***C***
  - $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0$
- *V* is ***n x n*** and orthonormal ($VV^T=V^TV=I$)



SVD Example (LSA)

- Term-Document Matrix $C = U\Sigma V^T$

C	d_1	d_2	d_3	d_4	d_5	d_6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
wood	1	0	0	1	1	0
tree	0	0	0	1	0	1

U	1	2	3	4	5
ship	−0.44	−0.30	0.57	0.58	0.25
boat	−0.13	−0.33	−0.59	0.00	0.73
ocean	−0.48	−0.51	−0.37	0.00	−0.61
wood	−0.70	0.35	0.15	−0.58	0.16
tree	−0.26	0.65	−0.41	0.58	−0.09

Σ	1	2	3	4	5
1	2.16	0.00	0.00	0.00	0.00
2	0.00	1.59	0.00	0.00	0.00
3	0.00	0.00	1.28	0.00	0.00
4	0.00	0.00	0.00	1.00	0.00
5	0.00	0.00	0.00	0.00	0.39

V^T	d_1	d_2	d_3	d_4	d_5	d_6
1	−0.75	−0.28	−0.20	−0.45	−0.33	−0.12
2	−0.29	−0.53	−0.19	0.63	0.22	0.41
3	0.28	−0.75	0.45	−0.20	0.12	−0.33
4	0.00	0.00	0.58	0.00	−0.58	0.58
5	−0.53	0.29	0.63	0.19	0.41	−0.22

SVD Example (LSA)

- Dimensionality reduction $C_2 = U\Sigma_2 V^T$

U	1	2	3	4	5
ship	−0.44	−0.30	0.00	0.00	0.00
boat	−0.13	−0.33	0.00	0.00	0.00
ocean	−0.48	−0.51	0.00	0.00	0.00
wood	−0.70	0.35	0.00	0.00	0.00
tree	−0.26	0.65	0.00	0.00	0.00

Σ_2	1	2	3	4	5
1	2.16	0.00	0.00	0.00	0.00
2	0.00	1.59	0.00	0.00	0.00
3	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00

V^T	d_1	d_2	d_3	d_4	d_5	d_6
1	−0.75	−0.28	−0.20	−0.45	−0.33	−0.12
2	−0.29	−0.53	−0.19	0.63	0.22	0.41
3	0.00	0.00	0.00	0.00	0.00	0.00
4	0.00	0.00	0.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	0.00	0.00

C_2	d_1	d_2	d_3	d_4	d_5	d_6
ship	0.85	0.52	0.28	0.13	0.21	−0.08
boat	0.36	0.36	0.16	−0.20	−0.02	−0.18
ocean	1.01	0.72	0.36	−0.04	0.16	−0.21
wood	0.97	0.12	0.20	1.03	0.62	0.41
tree	0.12	−0.39	−0.08	0.90	0.41	0.49

SVD Example (LSA)

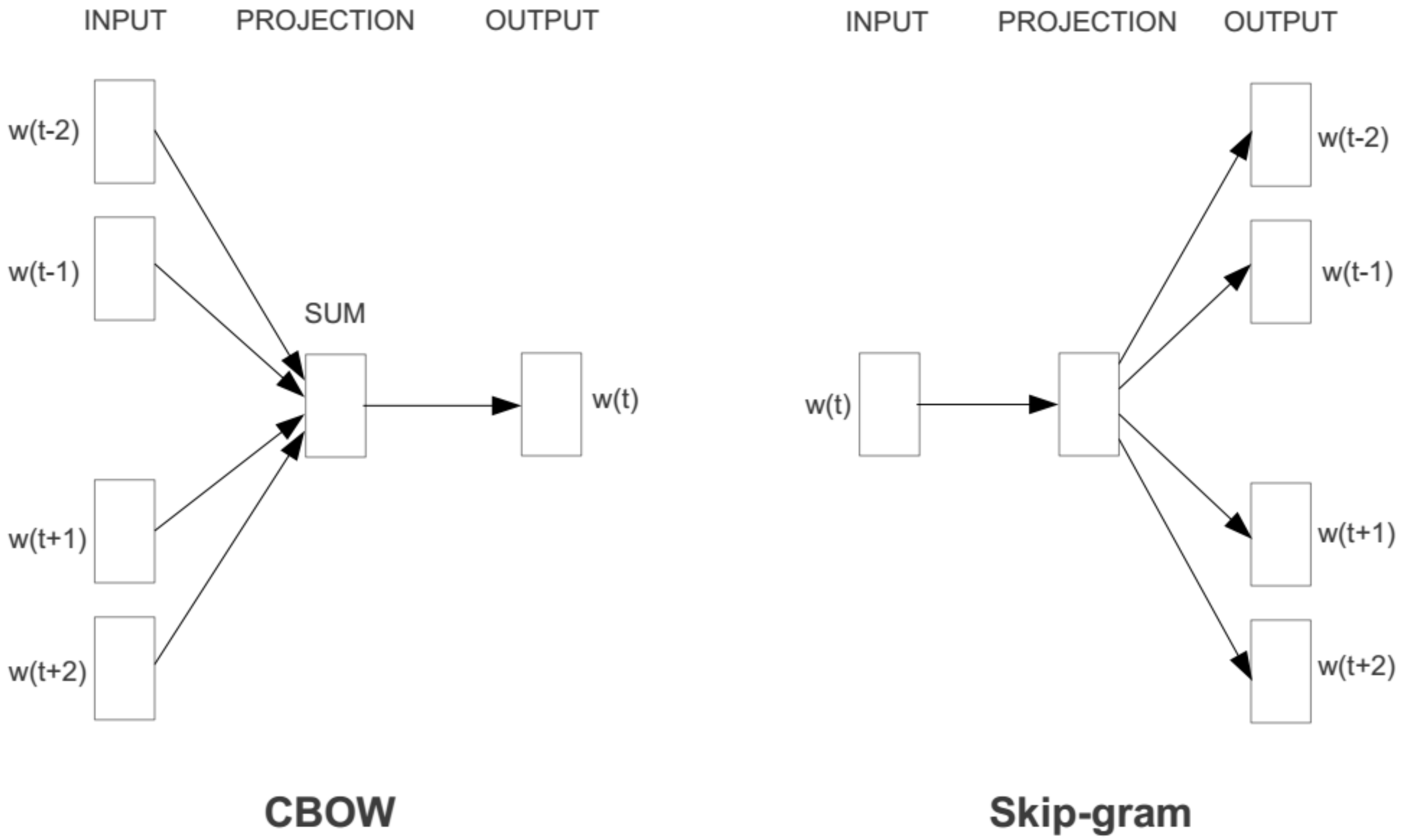
- Similarity of d_2 and d_3 in the original space: 0.
- Similarity of d_2 and d_3 in the reduced space: $0.52 * 0.28 + 0.36 * 0.16 + 0.72 * 0.36 + 0.12 * 0.20 + -0.39 * -0.08 \approx \mathbf{0.52}$

C	d_1	d_2	d_3	d_4	d_5	d_6
ship	1	0	1	0	0	0
boat	0	1	0	0	0	0
ocean	1	1	0	0	0	0
wood	1	0	0	1	1	0
tree	0	0	0	1	0	1

C_2	d_1	d_2	d_3	d_4	d_5	d_6
ship	0.85	0.52	0.28	0.13	0.21	−0.08
boat	0.36	0.36	0.16	−0.20	−0.02	−0.18
ocean	1.01	0.72	0.36	−0.04	0.16	−0.21
wood	0.97	0.12	0.20	1.03	0.62	0.41
tree	0.12	−0.39	−0.08	0.90	0.41	0.49

Word2vec

- Represent the meaning of the word by its context



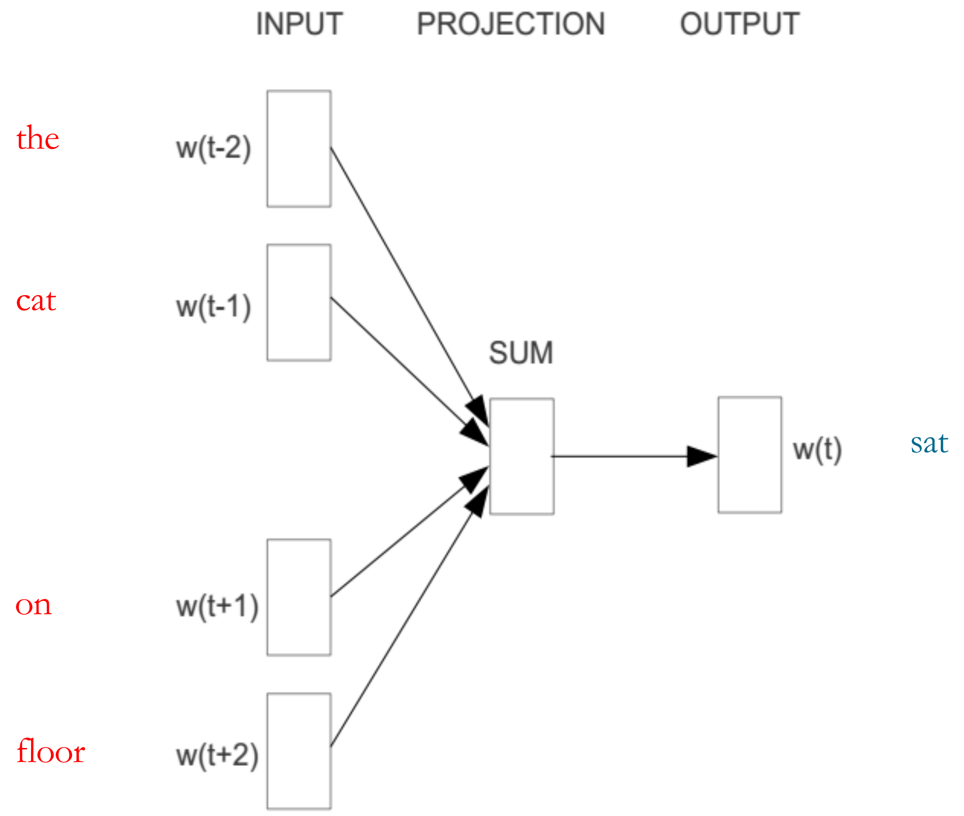
Word2Vec Learning models

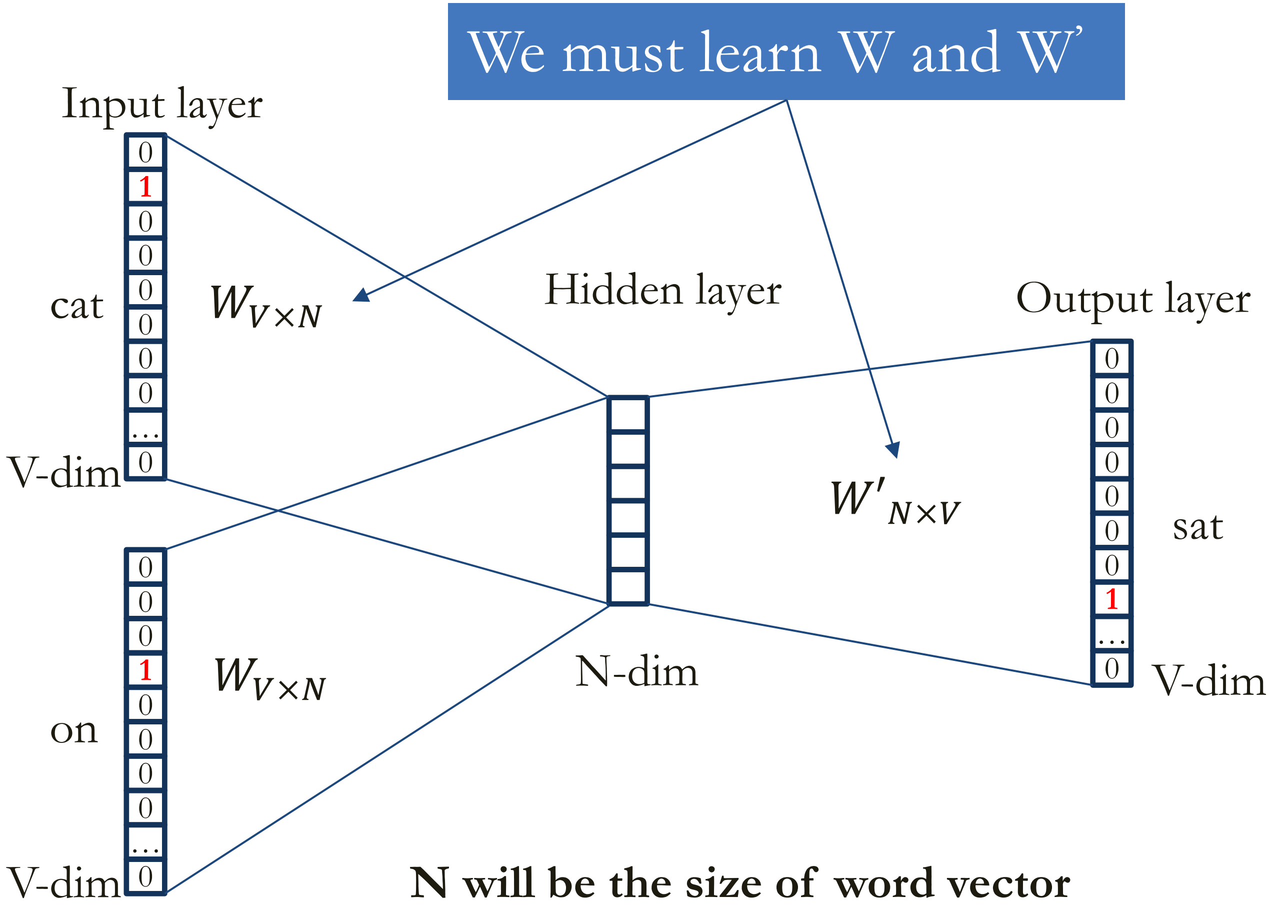
- Continuous Bag of Words (CBOW):
  - Given the context predict the word:
    - w_{i-2}, w_{i-1}, w_i, w_{i+1}, w_{i+2}
  - Example: **The cat ate _____.**
    - Fill in the blank, e.g. "food".
  - Faster to train
  - Works well for large amount of training data

- Continuous Skip-Gram:
  - Given the word predict the context:
    - w_{i-2}, w_{i-1}, w_i, w_{i+1}, w_{i+2}
  - Ex: ____ ____ ____ food.
    - Fill in the blank, e.g. "The cat ate"
  - Slower to train
  - Works better for infrequent words

Word2Vec: CBOW Example

- ”the cat ____ on floor”





Learning Connection Weights