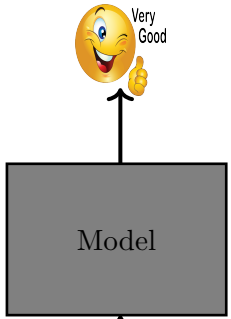
Module 10.1: One-hot representations of words

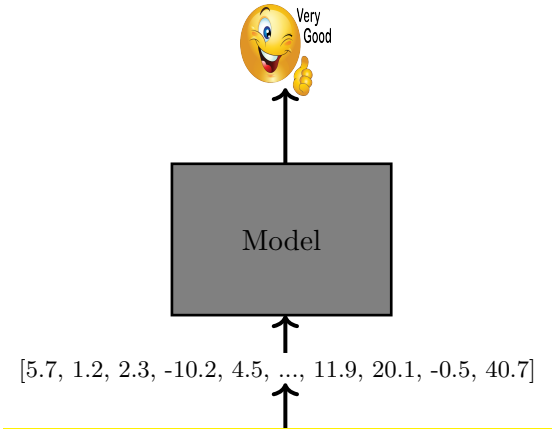
- Let us start with a very simple motivation for why we are interested in vectorial representations of words

- Let us start with a very simple motivation for why we are interested in vectorial representations of words
- Suppose we are given an input stream of words (sentence, document, etc.) and we are interested in learning some function of it (say, $\hat{y} = sentiments(words)$)

This is by far AAMIR KHAN's best one. Finest casting and terrific acting by all.



- Let us start with a very simple motivation for why we are interested in vectorial representations of words
- Suppose we are given an input stream of words (sentence, document, etc.) and we are interested in learning some function of it (say, $\hat{y} = sentiments(words)$)
- Say, we employ a machine learning algorithm (some mathematical model) for learning such a function ($\hat{y} = f(\mathbf{x})$)



- Let us start with a very simple motivation for why we are interested in vectorial representations of words
- Suppose we are given an input stream of words (sentence, document, etc.) and we are interested in learning some function of it (say, $\hat{y} = sentiments(words)$)
- Say, we employ a machine learning algorithm (some mathematical model) for learning such a function ($\hat{y} = f(\mathbf{x})$)
- We first need a way of converting the input stream (or each word in the stream) to a vector $\mathbf{x}$ (a mathematical quantity)

- Given a corpus,

Corpus:

- Human machine interface for computer applications
- User opinion of computer system response time
- User interface management system
- System engineering for improved response time

- Given a corpus,

Corpus:

- Human machine interface for computer applications
- User opinion of computer system response time
- User interface management system
- System engineering for improved response time

- Given a corpus, consider the set V of all unique words across all input streams (*i.e.*, all sentences or documents)

Corpus:

- Human machine interface for computer applications
- User opinion of computer system response time
- User interface management system
- System engineering for improved response time

V = [human,machine, interface, for, computer, applications, user, opinion, of, system, response, time, interface, management, engineering, improved]

- Given a corpus, consider the set V of all unique words across all input streams (*i.e.*, all sentences or documents)
- V is called the **vocabulary** of the corpus (*i.e.*, all sentences or documents)

Corpus:

- Human machine interface for computer applications
- User opinion of computer system response time
- User interface management system
- System engineering for improved response time

V = [human,machine, interface, for, computer, applications, user, opinion, of, system, response, time, interface, management, engineering, improved]

- Given a corpus, consider the set V of all unique words across all input streams (*i.e.*, all sentences or documents)
- V is called the **vocabulary** of the corpus (*i.e.*, all sentences or documents)
- We need a representation for every word in V

Corpus:

- Human machine interface for computer applications
- User opinion of computer system response time
- User interface management system
- System engineering for improved response time

V = [human,machine, interface, for, computer, applications, user, opinion, of, system, response, time, interface, management, engineering, improved]

machine:

0	1	0	...	0	0	0

- Given a corpus, consider the set V of all unique words across all input streams (*i.e.*, all sentences or documents)
- V is called the **vocabulary** of the corpus (*i.e.*, all sentences or documents)
- We need a representation for every word in V
- One very simple way of doing this is to use one-hot vectors of size $|V|$

Corpus:

- Human machine interface for computer applications
- User opinion of computer system response time
- User interface management system
- System engineering for improved response time

V = [human,machine, interface, for, computer, applications, user, opinion, of, system, response, time, interface, management, engineering, improved]

machine:

0	1	0	...	0	0	0

- Given a corpus, consider the set V of all unique words across all input streams (*i.e.*, all sentences or documents)
- V is called the **vocabulary** of the corpus (*i.e.*, all sentences or documents)
- We need a representation for every word in V
- One very simple way of doing this is to use one-hot vectors of size $|V|$
- The representation of the i-th word will have a 1 in the i-th position and a 0 in the remaining $|V|-1$ positions

cat:	0	0	0	0	0	1	0
dog:	0	1	0	0	0	0	0
truck:	0	0	0	1	0	0	0

Problems:

- V tends to be very large (for example, 50K for PTB, 13M for Google 1T corpus)

cat:	0	0	0	0	0	1	0
dog:	0	1	0	0	0	0	0
truck:	0	0	0	1	0	0	0

Problems:

- V tends to be very large (for example, 50K for PTB, 13M for Google 1T corpus)
- These representations do not capture any notion of similarity

cat:	0	0	0	0	0	1	0
dog:	0	1	0	0	0	0	0
truck:	0	0	0	1	0	0	0

Problems:

- V tends to be very large (for example, 50K for PTB, 13M for Google 1T corpus)
- These representations do not capture any notion of similarity
- Ideally, we would want the representations of cat and dog (both domestic animals) to be closer to each other than the representations of cat and truck

cat:	0	0	0	0	0	1	0
dog:	0	1	0	0	0	0	0
truck:	0	0	0	1	0	0	0

$$euclid\_dist(\textbf{cat}, \textbf{dog}) = \sqrt{2}$$
$$euclid\_dist(\textbf{dog}, \textbf{truck}) = \sqrt{2}$$

Problems:

- V tends to be very large (for example, 50K for PTB, 13M for Google 1T corpus)
- These representations do not capture any notion of similarity
- Ideally, we would want the representations of cat and dog (both domestic animals) to be closer to each other than the representations of cat and truck
- However, with 1-hot representations, the Euclidean distance between **any two words** in the vocabulary in $\sqrt{2}$

cat:	0	0	0	0	0	1	0
dog:	0	1	0	0	0	0	0
truck:	0	0	0	1	0	0	0

$$euclid\_dist(\textbf{cat}, \textbf{dog}) = \sqrt{2}$$
$$euclid\_dist(\textbf{dog}, \textbf{truck}) = \sqrt{2}$$
$$cosine\_sim(\textbf{cat}, \textbf{dog}) = 0$$
$$cosine\_sim(\textbf{dog}, \textbf{truck}) = 0$$

Problems:

- V tends to be very large (for example, 50K for PTB, 13M for Google 1T corpus)
- These representations do not capture any notion of similarity
- Ideally, we would want the representations of cat and dog (both domestic animals) to be closer to each other than the representations of cat and truck
- However, with 1-hot representations, the Euclidean distance between **any two words** in the vocabulary in $\sqrt{2}$
- And the cosine similarity between **any two words** in the vocabulary is 0