One hot Encoding

I had washed my house.

I had my house washed.

Exact same words.. but different meaning.

Why?

	had	house	I	my	washed
Index	0	1	2	3	4
Document #1	1	1	1	1	1
Document #2	1	1	1	1	1

Document #1

He is a good boy. She is also good.

Document #2

Radhika is a good person.

Vocabulary

a, also, boy, good, He, is, person, She, Radhika

	a	**also**	**boy**	**good**	**He**	**Is**	**person**	**She**	**Radhika**
Index	0	1	2	3	4	5	6	7	8

Assign index for each word in vocabulary

Document #1

He is a good boy. She is also good.

Document #2

Radhika is a good person.

Vocabulary

a, also, boy, good, He, is, person, She, Radhika

	a	also	boy	good	He	Is	person	She	Radhika
Index	0	1	2	3	4	5	6	7	8
Document #1	1	1	1	2	1	2	0	1	0
Document #2	1	0	0	1	0	1	1	0	1

Document as Vector

(CountVectorizer)

Word	Index	0	1	2	3	4	5	6	7	8
a	0	1	0	0	0	0	0	0	0	0
also	1									
boy	2									
good	3									
He	4									
is	5									
person	6									
She	7									
Radhika	8									

In one hot encoding ...each word is a as vector

Word	Index	0	1	2	3	4	5	6	7	8
a	0	1	0	0	0	0	0	0	0	0
also	1	0	1	0	0	0	0	0	0	0
boy	2									
good	3									
He	4									
is	5									
person	6									
She	7									
Radhika	8									

Words as a vector

Word	Index	0	1	2	3	4	5	6	7	8
a	0	1	0	0	0	0	0	0	0	0
also	1	0	1	0	0	0	0	0	0	0
boy	2	0	0	1	0	0	0	0	0	0
good	3	0	0	0	1	0	0	0	0	0
He	4	0	0	0	0	1	0	0	0	0
is	5	0	0	0	0	0	1	0	0	0
person	6	0	0	0	0	0	0	1	0	0
She	7	0	0	0	0	0	0	0	1	0
Radhika	8	0	0	0	0	0	0	0	0	1

One hot encoding

How do we use one hot encoding a document.

- Replace each word in the document with its one hot encoding.
- Keep word sequence intact i.e 1st word comes first, 2nd word second and so on.

Document #1

He is a good boy. She is also good.

Document#1	Word Index	0	1	2	3	4	5	6	7	8
He	4	0	0	0	0	**1**	0	0	0	0
is	5	0	0	0	0	0	**1**	0	0	0
a	0	**1**	0	0	0	0	0	0	0	0
good	3	0	0	0	**1**	0	0	0	0	0
boy	2	0	0	**1**	0	0	0	0	0	0
She	7	0	0	0	0	0	0	0	**1**	0
is	5	0	0	0	0	0	**1**	0	0	0
also	1	0	**1**	0	0	0	0	0	0	0
good	3	0	0	0	**1**	0	0	0	0	0

Document as matrix

Document #2

Radhika is a good person.

Document#1	Word Index	0	1	2	3	4	5	6	7	8
Radhika	8	0	0	0	0	0	0	0	0	**1**
is	5	0	0	0	0	0	**1**	0	0	0
a	0	**1**	0	0	0	0	0	0	0	0
good	3	0	0	0	**1**	0	0	0	0	0
person	6	0	0	0	0	0	0	**1**	0	0

Document → 5 x 9 matrix

What are the issues with One hot encoding.

- For large vocabulary size (common in NLP problems), document matrix becomes huge.
- Word vector do not provide any information on how words are related.

Discovering relationships between words

King + Man = Queen + ?

Can we use one hot encoding to solve this equation?

King + Man = Queen + Woman

How do we get numbers which show relationship between words?

King + Man = Queen + Woman

440	480	530	390
310	410	620	100
220	310	298	222
590	290	352	528
100	517	291	326

...something like this

Which is similar to ‘cat’

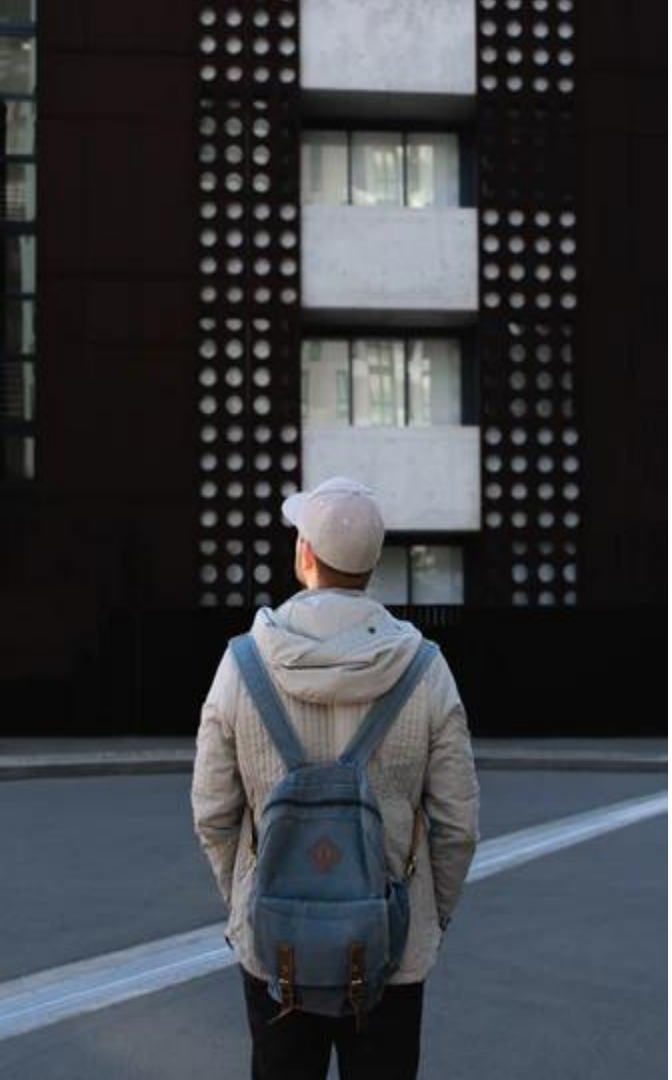
Plane

Bed

Dog

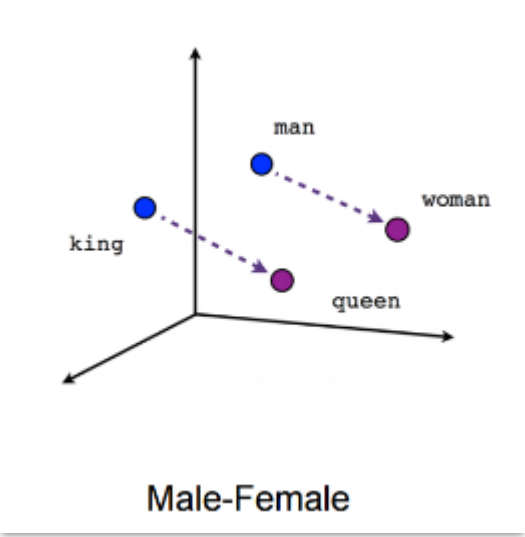
Boy

How can we get word vectors to answer such questions?



Discovering Semantic relationship using

Word2Vec

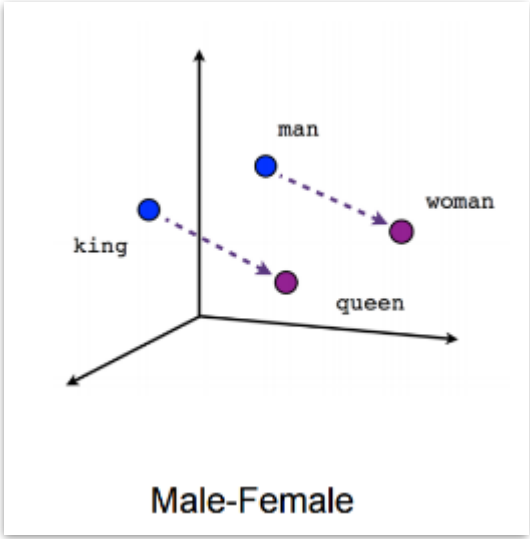


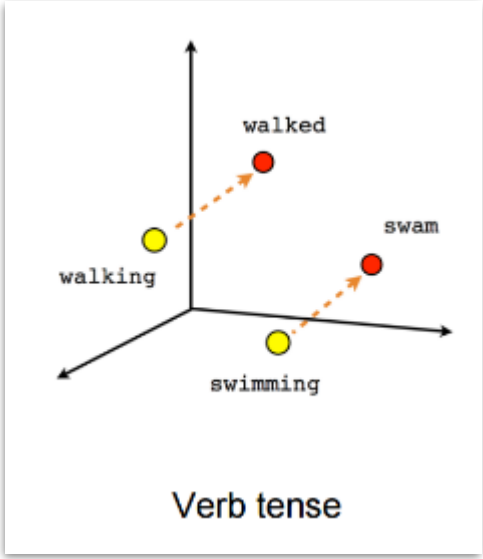
Word2vec provides a vector for each word which can help discover relationship between words

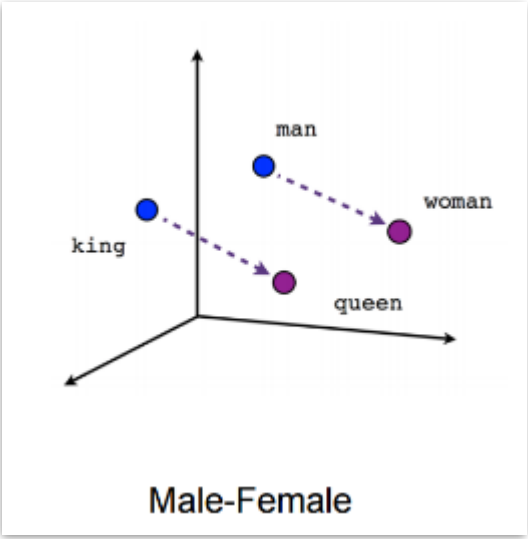
Distance between word vectors of
King & queen

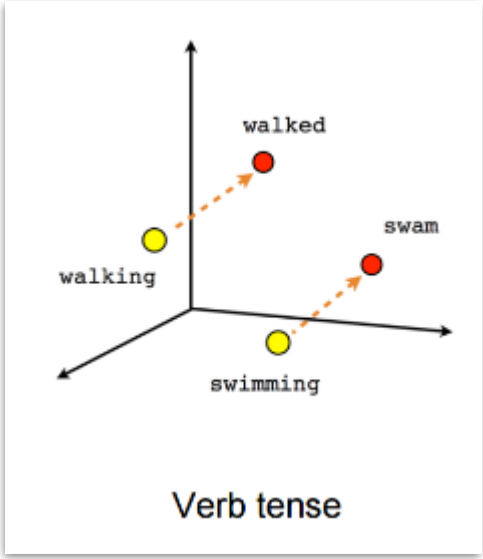
=

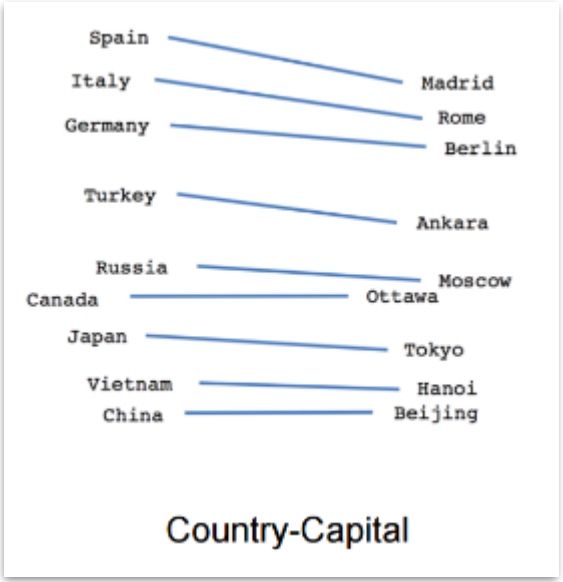
Distance between word vectors of
man and woman


man
woman
king
queen
Male-Female


walked
swam
walking
swimming
Verb tense


man
woman
king
queen
Male-Female


walked
swam
walking
swimming
Verb tense


Spain
Madrid
Italy
Rome
Germany
Berlin
Turkey
Ankara
Russia
Moscow
Canada
Ottawa
Japan
Tokyo
Vietnam
Hanoi
China
Beijing
Country-Capital

How to create Word2Vec vectors?

We use machine learning (ML) to create these vectors. There are two ways to build word2vec embeddings (vectors).

1. Supervised Learning
2. Unsupervised (or Self Supervised) Learning

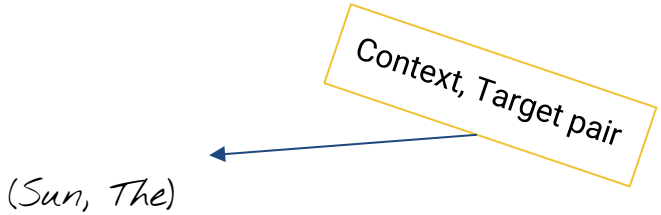
How does Word2Vec create embeddings using unsupervised learning?

It **learns** word embeddings by understanding word's **neighbours**.

The	Sun	rises	in	the	east

Given a word, what are the nearby words

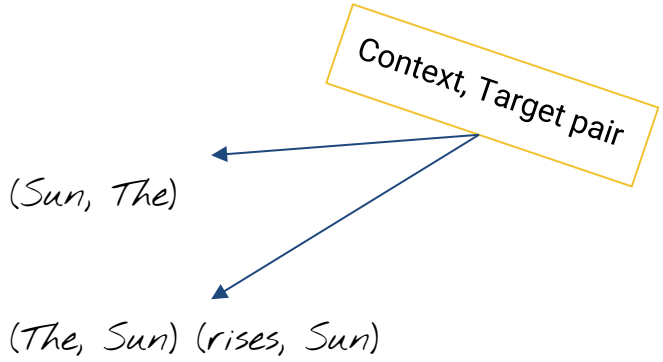
The	Sun	rises	in	the	east



For 1st word ‘The’, who are it’s neighbours

The	Sun	rises	in	the	east

The	Sun	rises	in	the	east

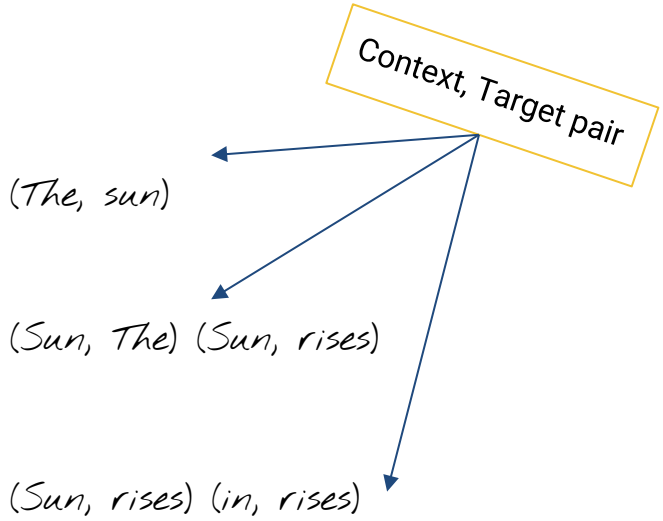


...neighbours of second word 'Sun'

The	Sun	rises	in	the	east

The	Sun	rises	in	the	east

The	Sun	rises	in	the	east



What is considered near?

We can decide on the same using a **window size**.

Window size = 1

(rise, in) (the, in)

One word to the left and one to the right are considered neighbours

Window Size

Window size = 1

(in, rises) (in, the)

Window size = 2

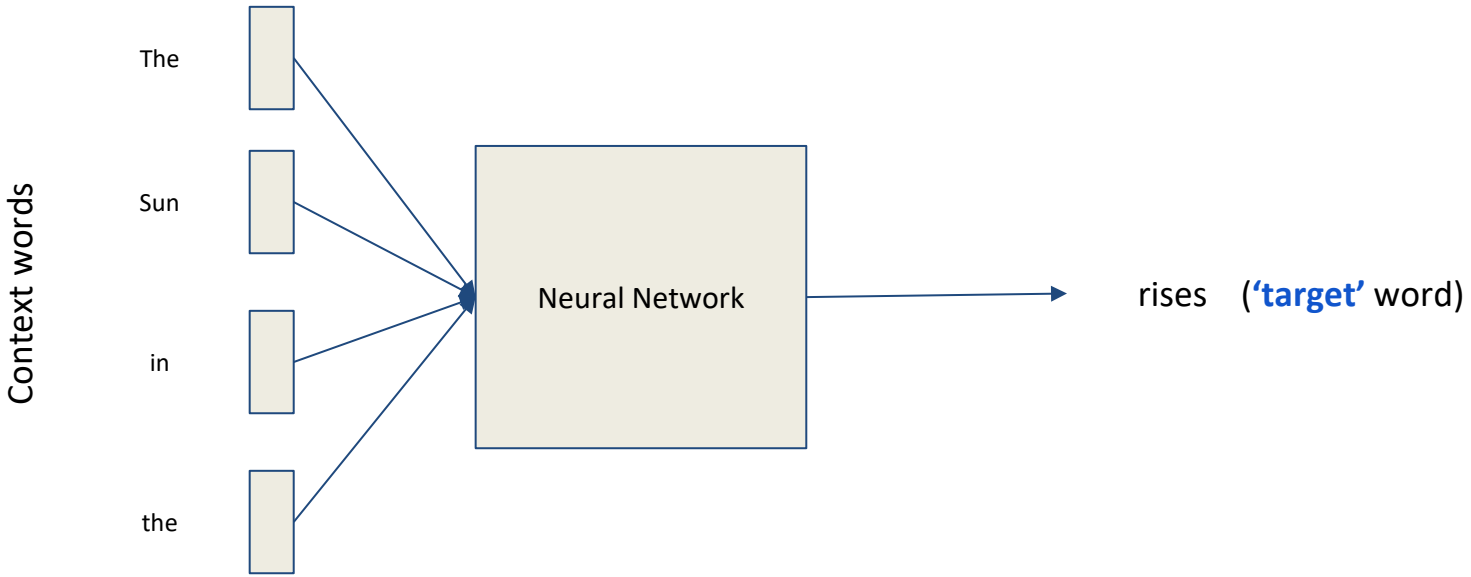
(rises, in) (Sun, in) (the, in) (east, in)

Two words to the left and two to the right are considered neighbours

Window Size

Building Word2Vec Embeddings

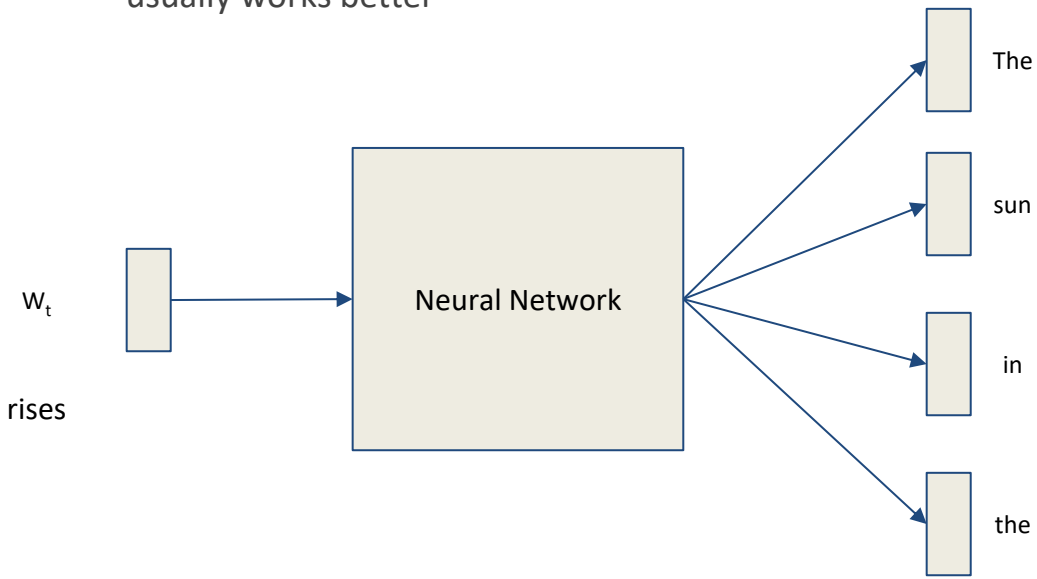
2 ways to get it


Context words
The
Sun
in
the
Neural Network
rises ('target' word)

CBOW model

Skip-Gram model

usually works better

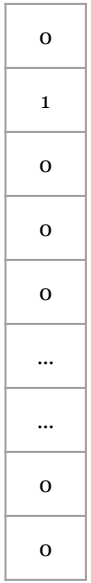


Predict the
'Context'
words

Word2vec model uses a simple Neural network

- 1 Input Layer
- 1 Hidden Layer
- 1 Output Layer

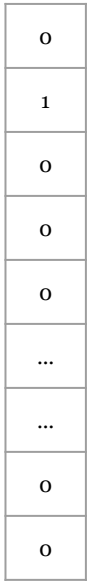
Input word as **one hot vector**



W_t

Input layer

Input word as **one hot vector**

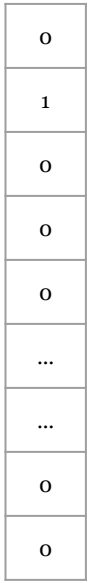


W_t

Size of the input vector?

Input layer

Input word as **one hot vector**

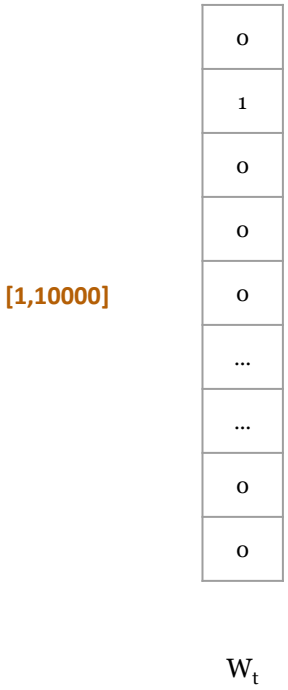


W_t

Same as
vocabulary size

Input layer

Input word as **one hot vector**

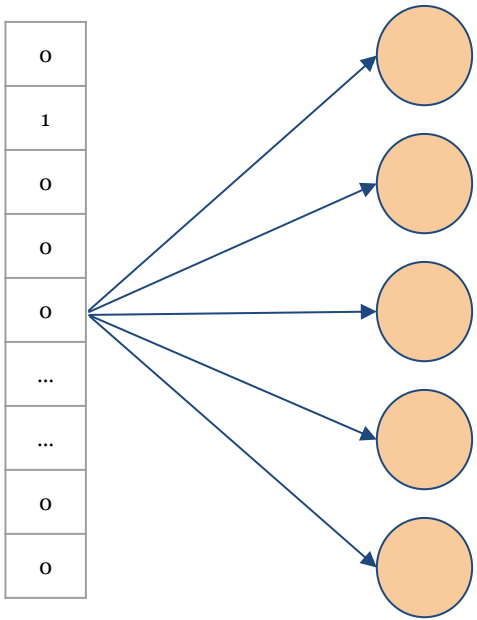


Assume we have 10000 words vocabulary

Input layer

Hidden Layer

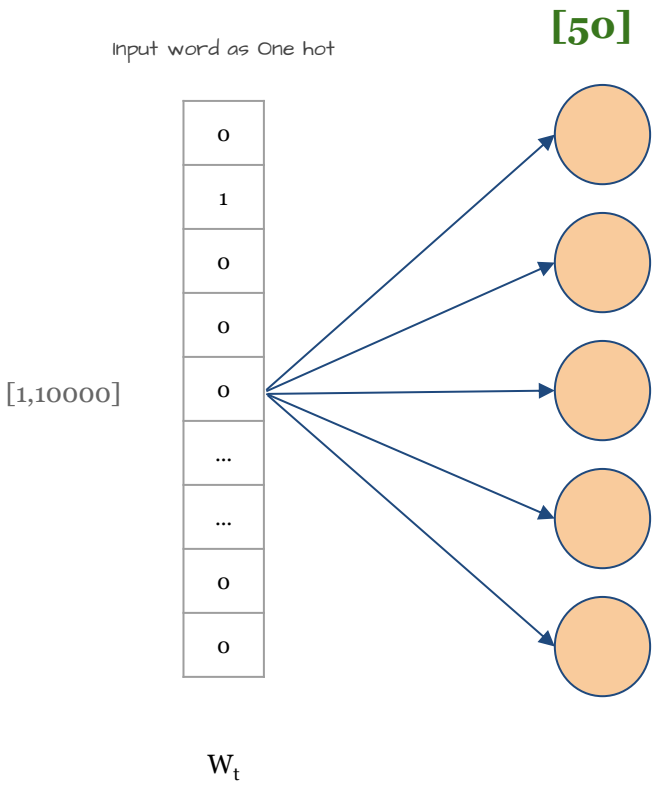
Input word as One hot



[1,10000]

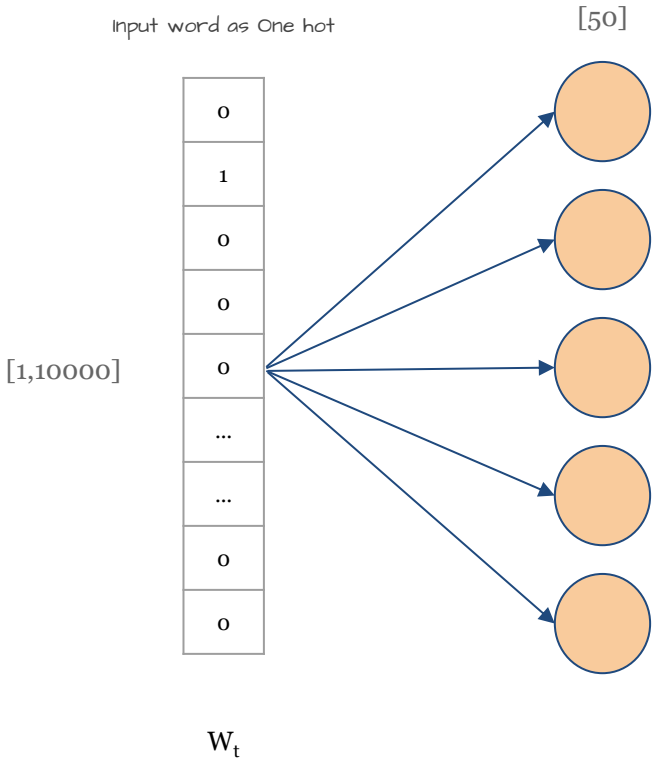
W_t

How many neurons in hidden layer?

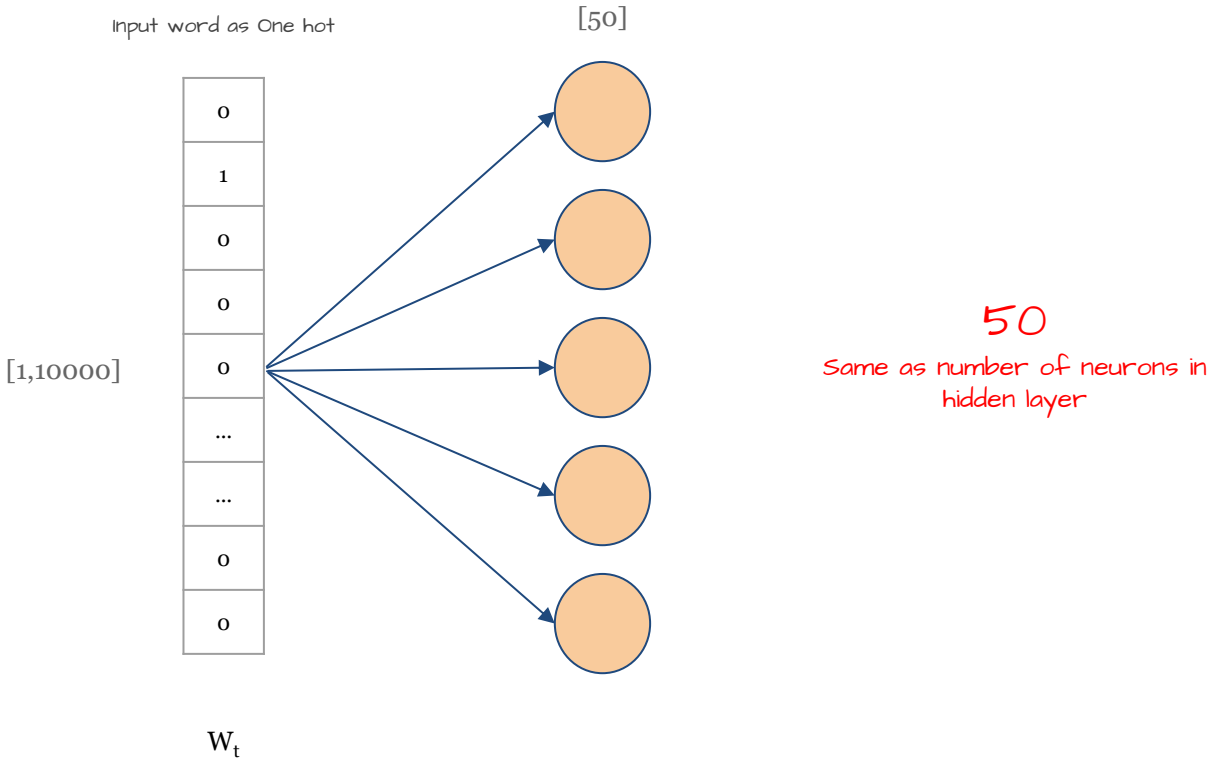


Let's have 50

(or whatever you like)



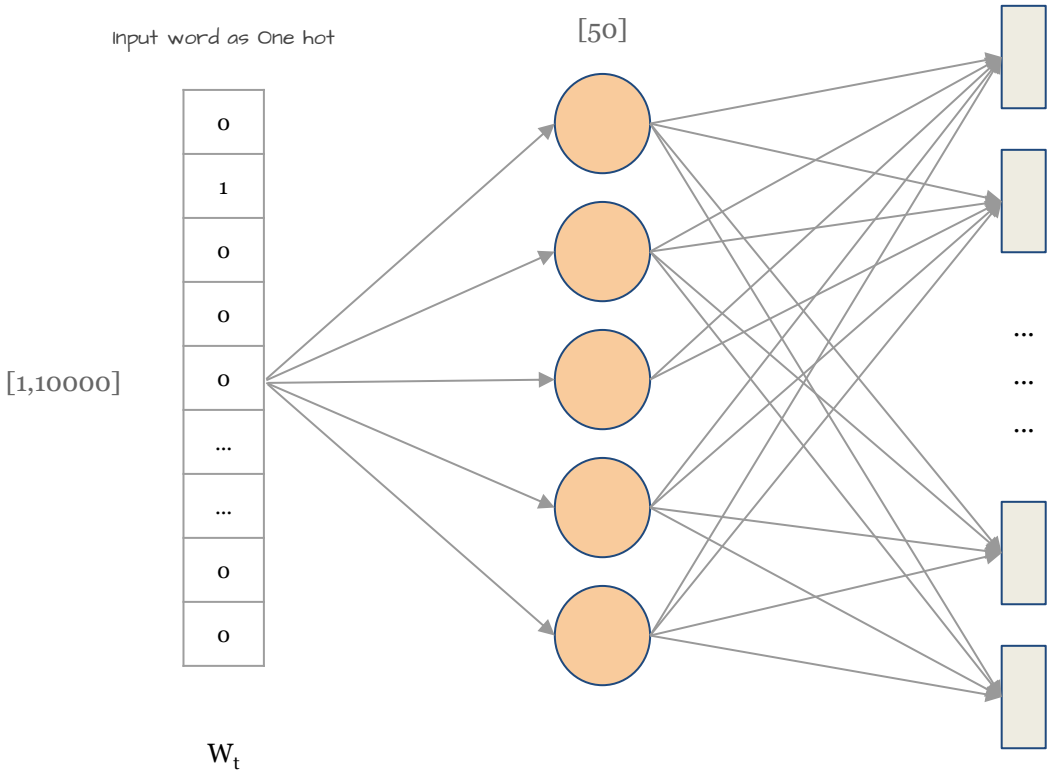
How many hidden layer outputs for each Word?



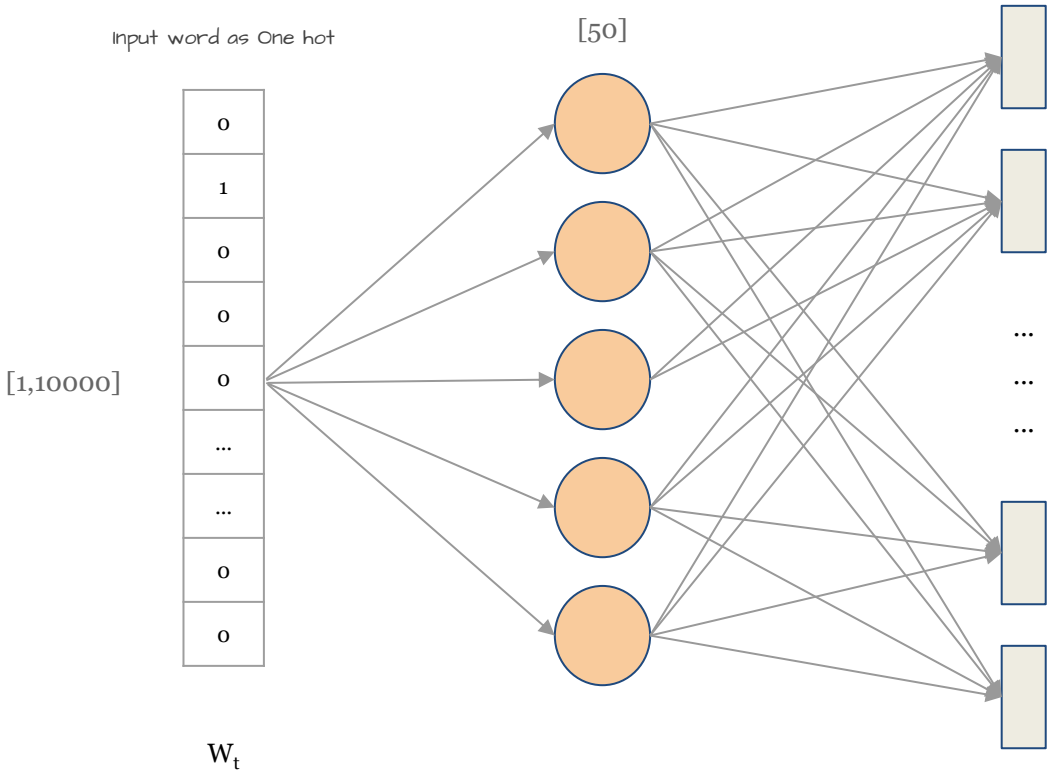
50

Same as number of neurons in hidden layer

The Output Layer

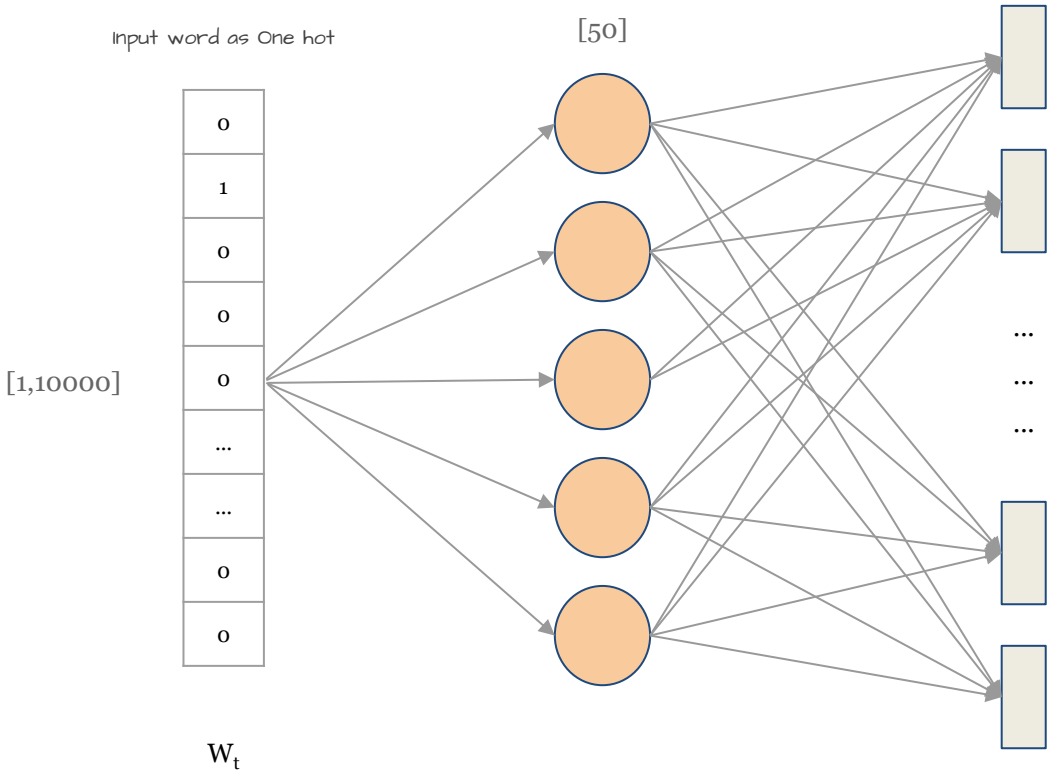


How many Outputs?



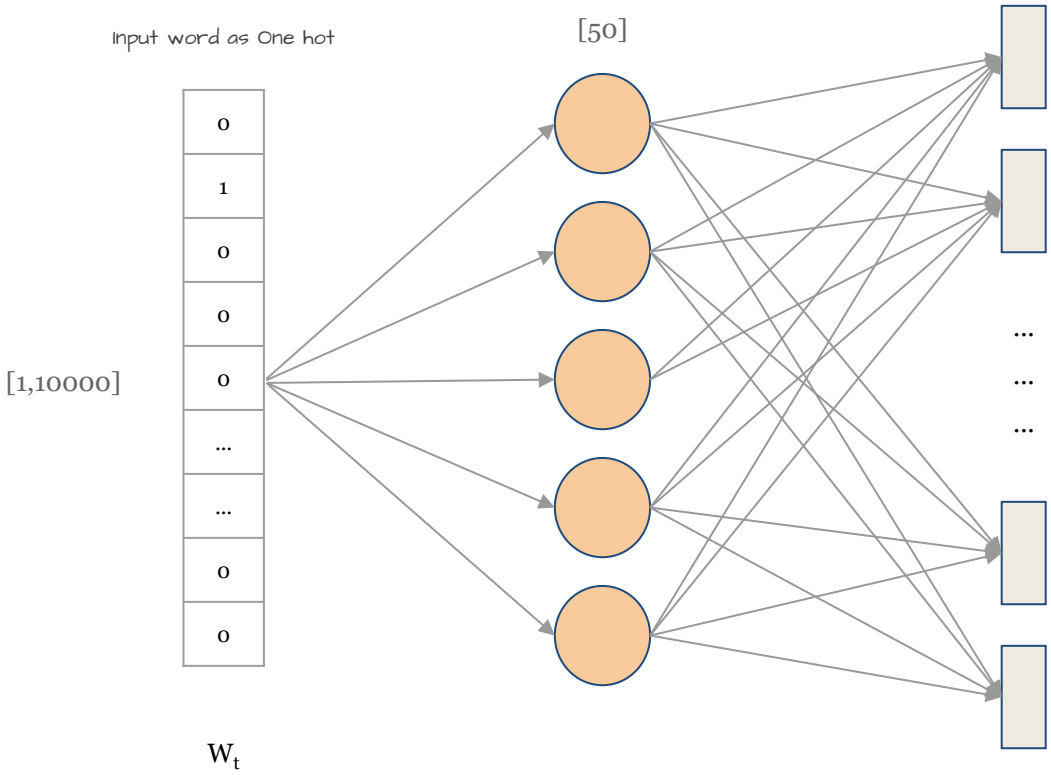
10,000

Same as Vocabulary size



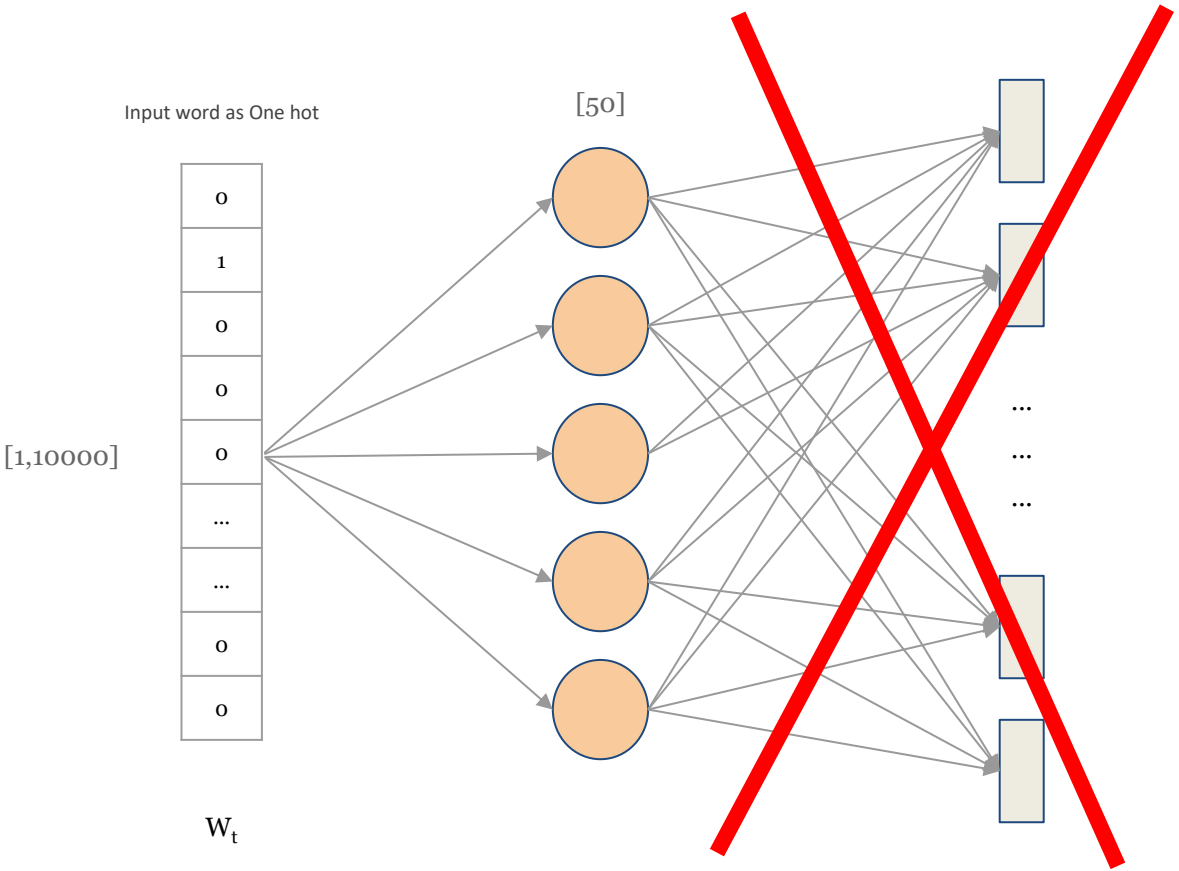
10,000

Predictions are a lot...how do we handle it better?

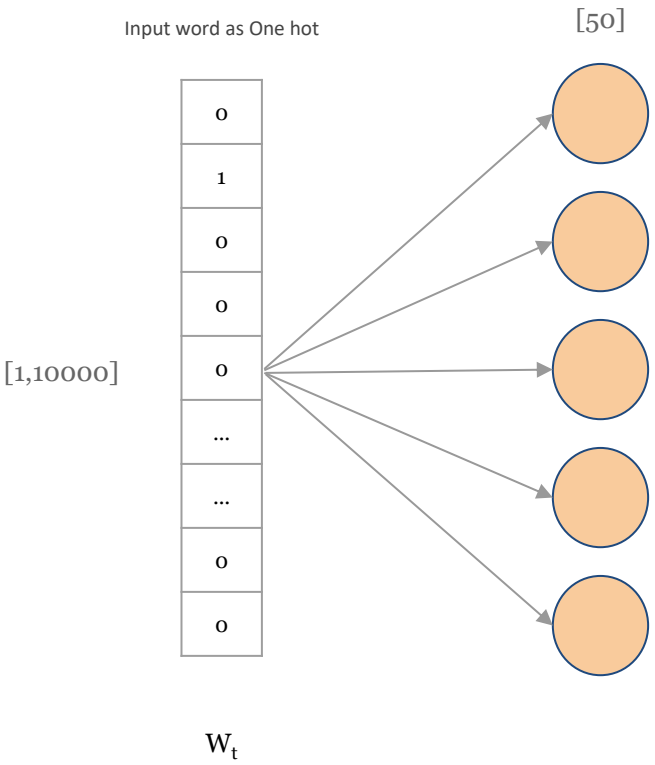


10,000
(same as Vocab size)

How do we get the word embeddings?

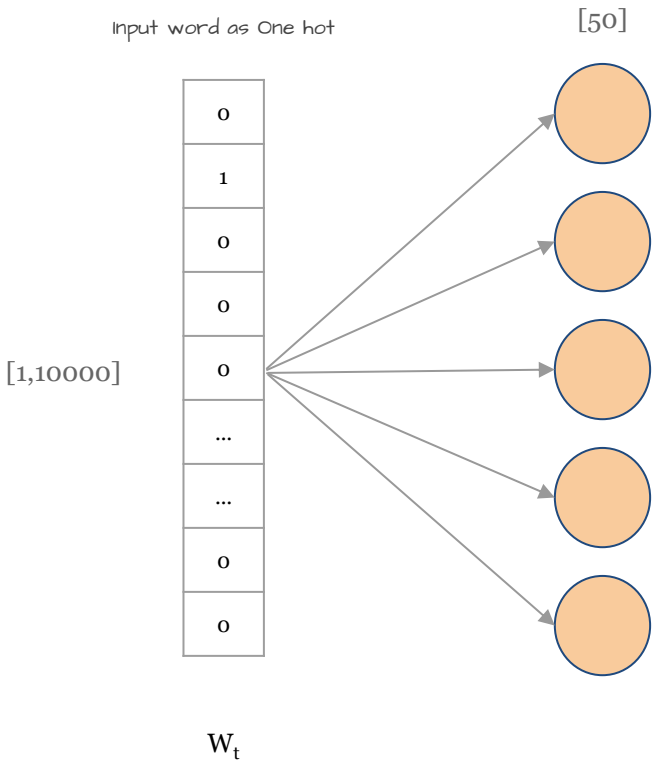


After training output layer is discarded



Output of hidden layer is the **word embeddings**

For each word in vocabulary, we get... **50 numbers**



For each word in vocabulary, we get...

50 numbers

Embedding Size

Building Word2Vec Model

Using gensim

Variations of Word2Vec model

Global **Ve**ctors (GloVe)

1. Similar to Word2Vec in looking at neighbours of a word
2. But also takes into account how many times two words **were neighbours**
3. Approach provided by Stanford University (2014)
4. Lot's of pre-trained models (with different embedding size) available

facebook research

FastText (by facebook)

1. Very similar to Word2Vec in looking at neighbours of a word
2. Word embedding not only at word level but at **subword** level
3. Subwords are created as character n-grams
4. Can handle **unseen or rare** words

Possible dictionary word/subwords for ‘awesome’
awesome → awe, wes, eso, som, ome

Pre-trained Word Embedding models

https://github.com/RaRe-Technologies/gensim-data

Using Pre-Trained Word2Vec model

An Example

Document #1

He is a good boy. She is also good.

Document #2

Radhika is a good person.

Vocabulary

a, also, boy, good, He, is, person, She, Radhika

	a	also	boy	good	He	is	person	She	Radhika
Index	1	2	3	4	5	6	7	8	9

Build vocabulary and assign index to each unique word

Document #1

He is a good boy. She is also good.

5, 6, 1, 4, 3, 8, 6, 2, 4

Document #2

Radhika is a good person.

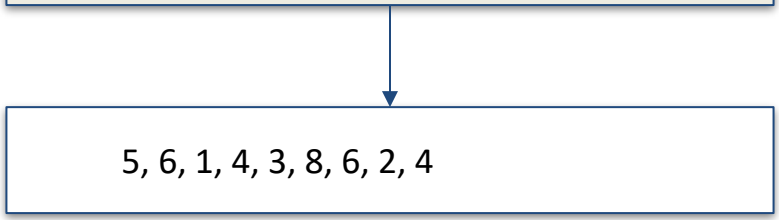
9, 6, 1, 4, 7

	a	also	boy	good	He	is	person	She	Radhika
Index	1	2	3	4	5	6	7	8	9

Create Document Vector - Replace each word by it's index

Document #1

He is a good boy. She is also good.

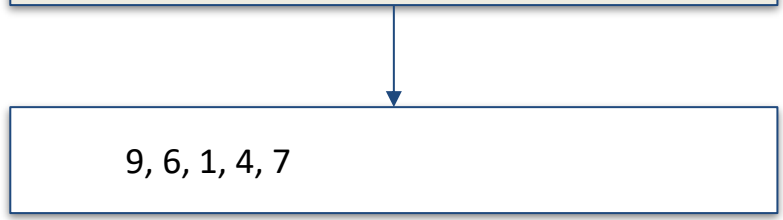


5, 6, 1, 4, 3, 8, 6, 2, 4

Vector length = 9

Document #2

Radhika is a good person.

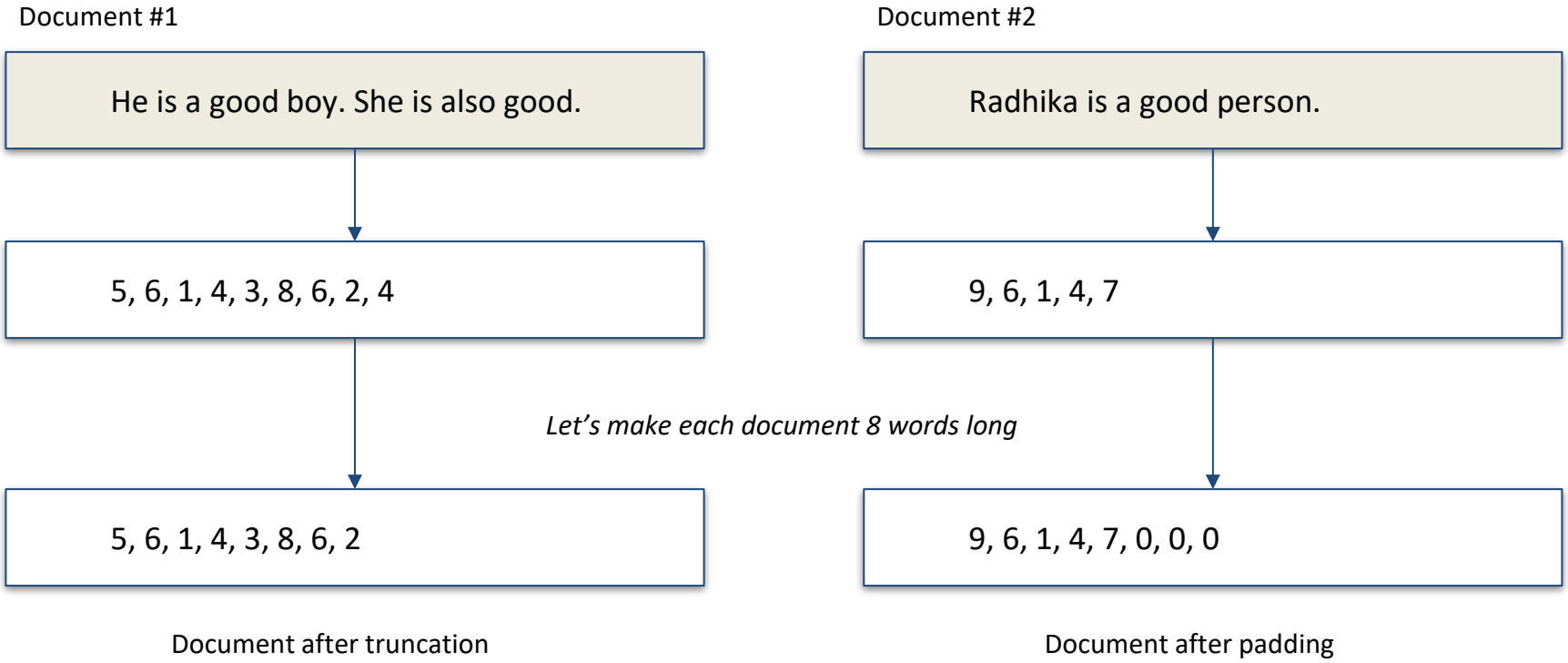


9, 6, 1, 4, 7

Vector length = 5

Make document vectors to be same length

1. Document need to be same size before they can be used in ML model
2. We have to first decide what size each document should be e.g 8 words long
3. If a document has more words than chosen size, we will truncate additional words
4. If document has less words than chosen size, we will pad the document with dummy words



Make document vectors to be same length

Building Document vector with Padding and Truncation

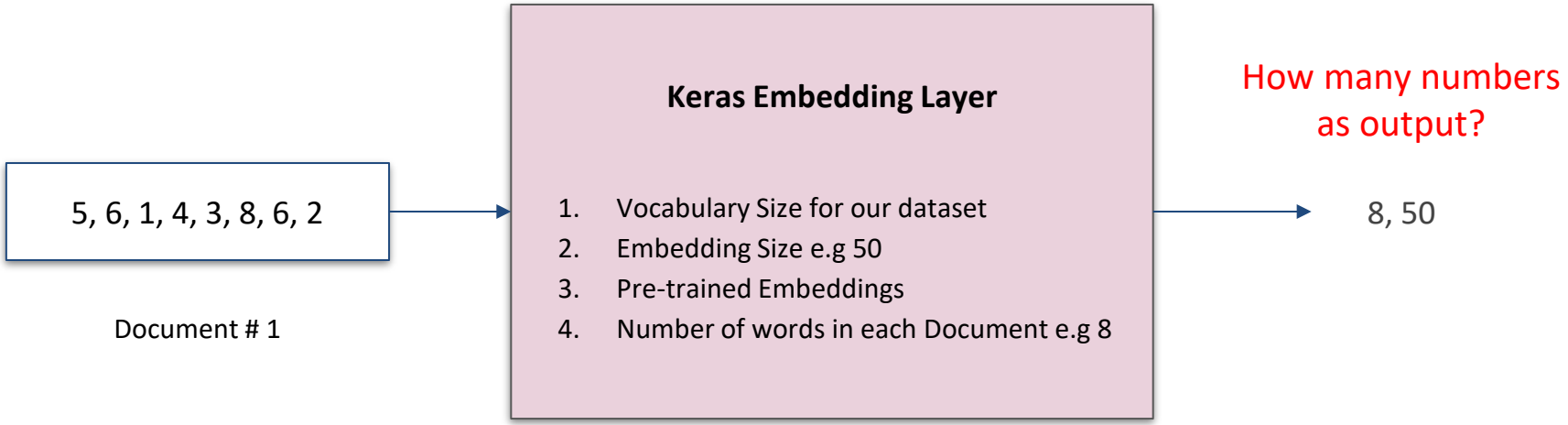
Using TensorFlow Keras

How do we get the word embeddings for our vocabulary words?

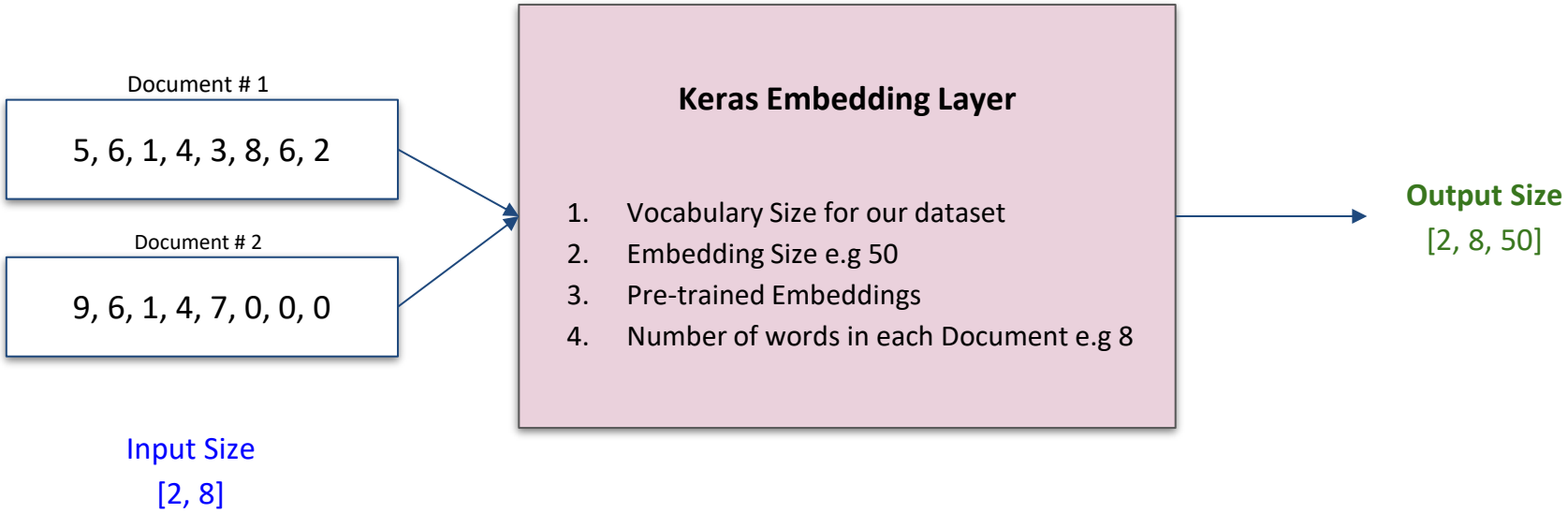
Word embeddings from Pre-trained model

1. Download a pre-trained word embedding model
    a. e.g Google Word2Vec or Glove or fasttext
    b. Check out https://github.com/RaRe-Technologies/gensim-data
    c. ‘Gensim-data’ provides an easy way to work with pre-trained word embedding models
2. Extract word embedding for your dataset vocabulary words
    a. We do not need embedding of all the words in the pre-trained model

How do we use word embeddings to train a model?

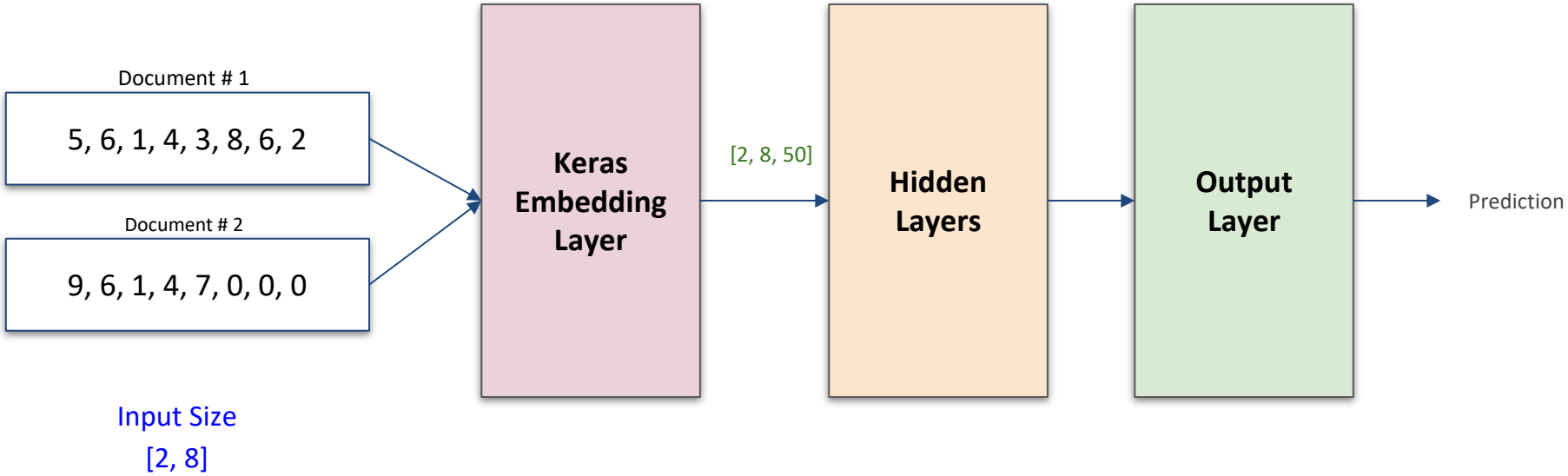


Using Keras Embedding layer

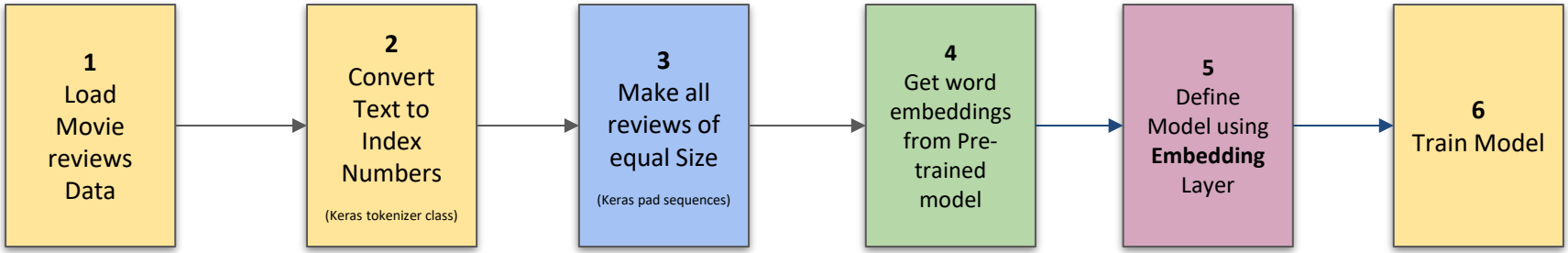


We can feed multiple documents at once *i.e* a batch of documents

What to do with Embedding layer output?



Feed it to next layer in the model



Building a model using pre-trained Word Embedding model