Natural Language Processing

One hot Encoding

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

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

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

Assign index for each word in Vocabulary

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									

Words 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 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

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

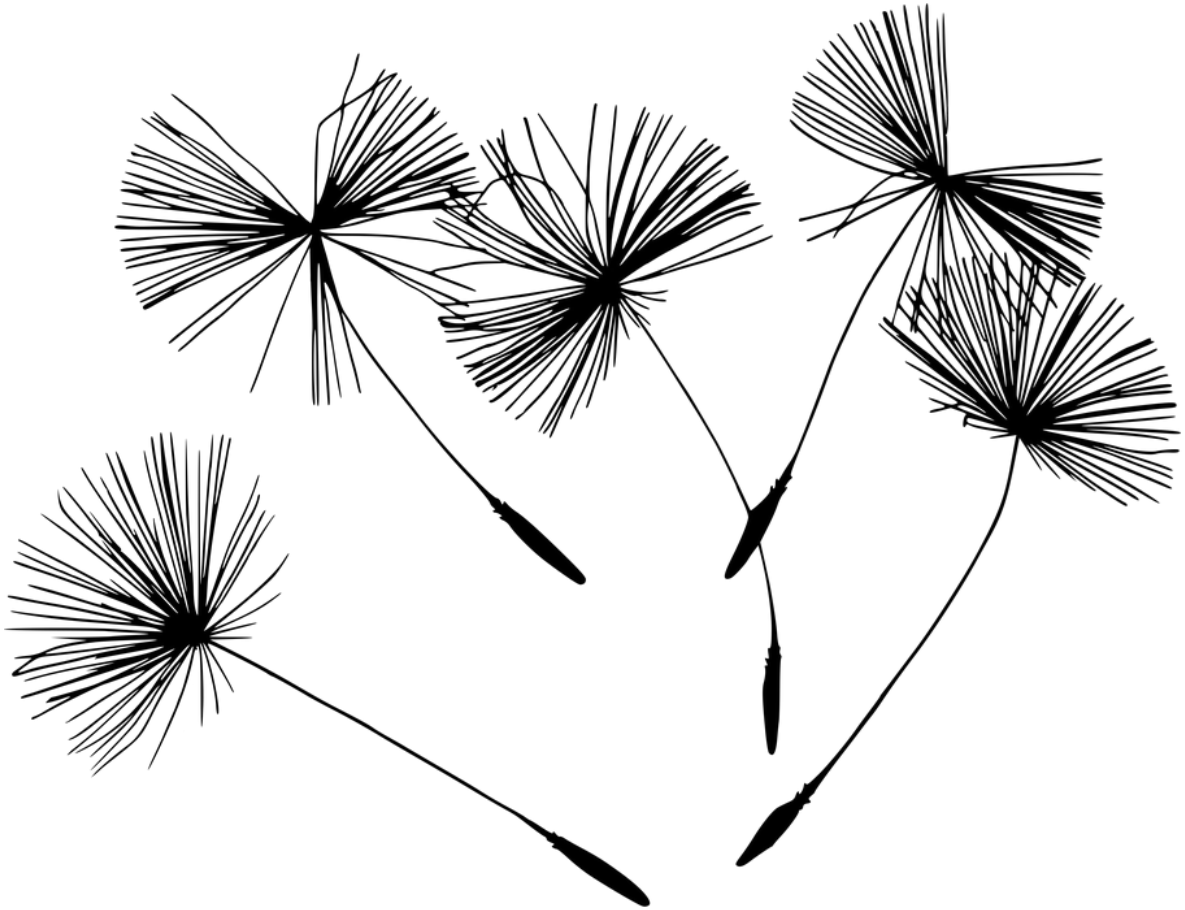
Document as Matrix

Document #2

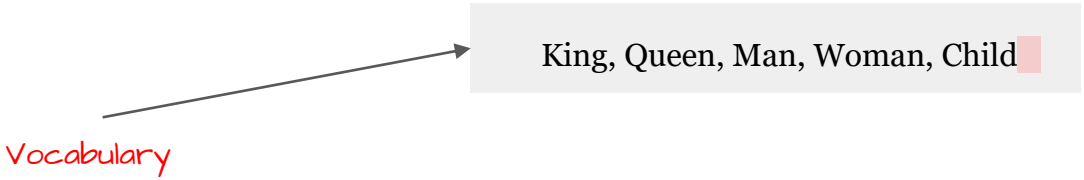
Radhika is a good person.

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



Discovering relationships between Words



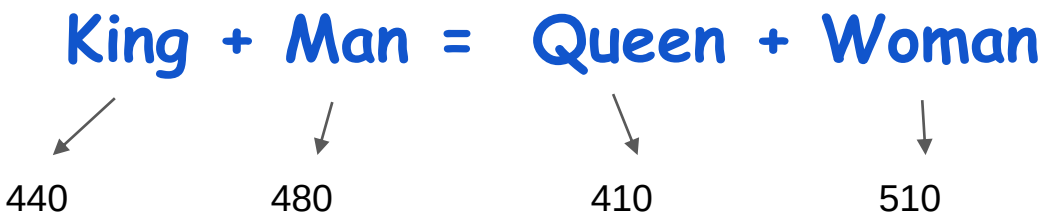
Vocabulary → King, Queen, Man, Woman, Child

One hot vector

King	1	0	0	0	0	
	0	1	0	0	0	Queen
Child	0	0	0	0	1	

King + Man = Queen + ?

Can we use TF-IDF or one hot vector to solve this equation?



920 = 920

Which is similar to ‘cat’

Plane

Bed

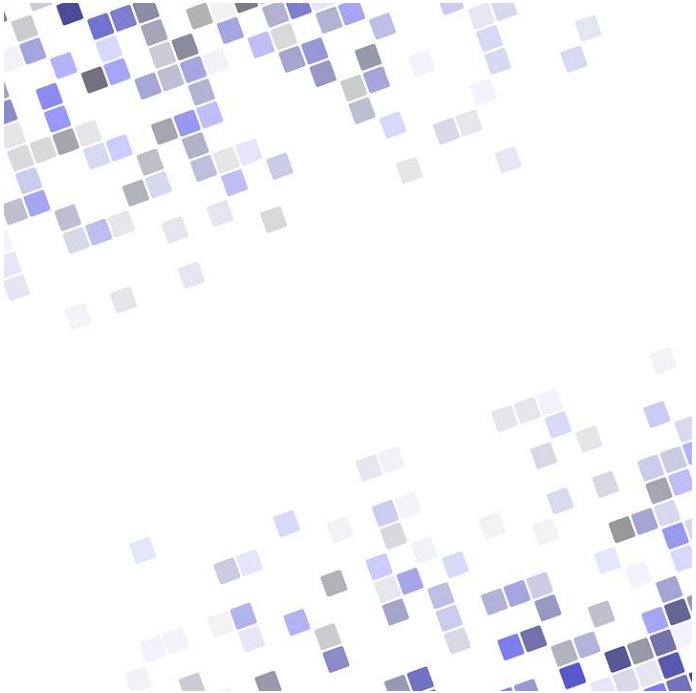
Dog

Boy

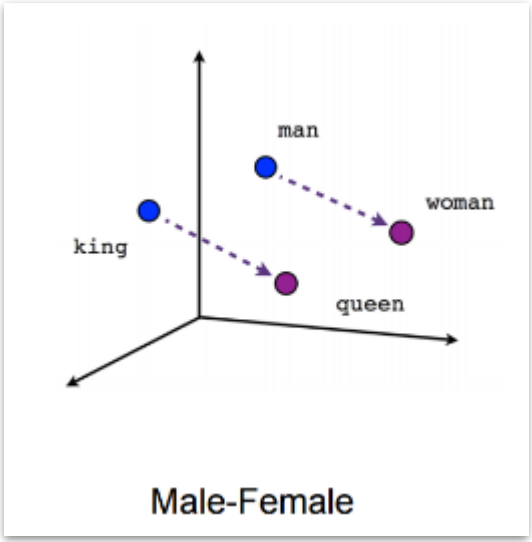
greatlearning
Learning for Life

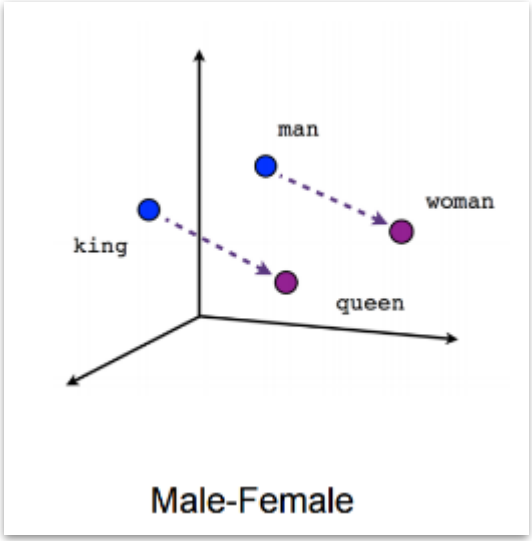
Discovering Semantic relationship using

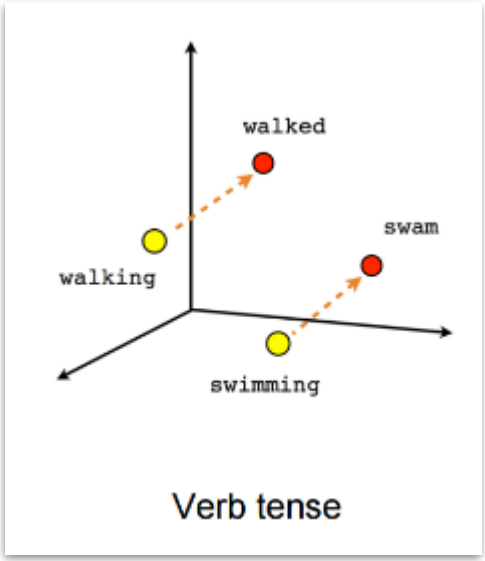
Word2Vec

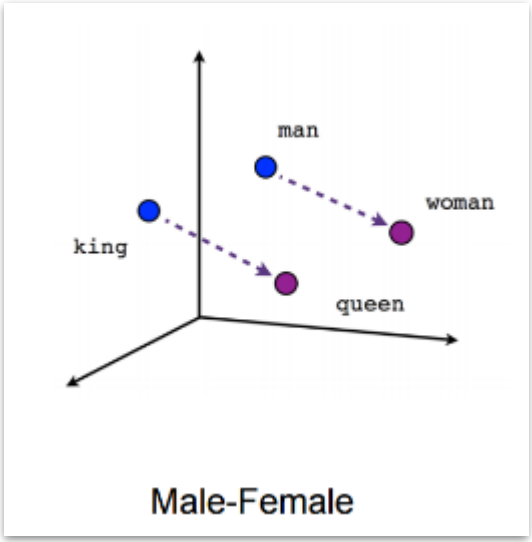


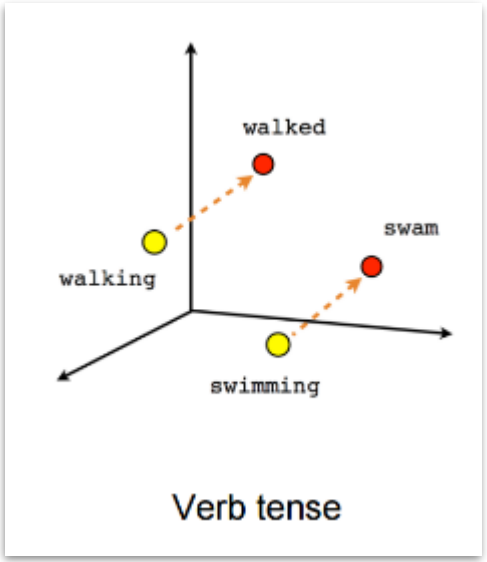
What does Word2Vec do?

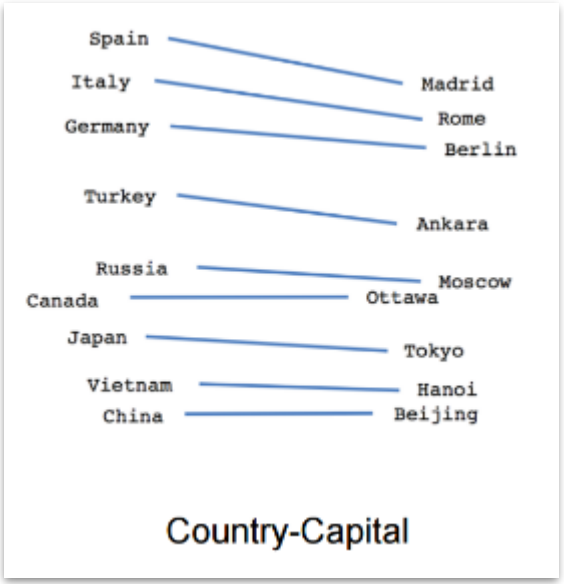












How does Word2Vec work?

The	Sun	rises	in	the	east

Given a word, what are the nearby word(s)

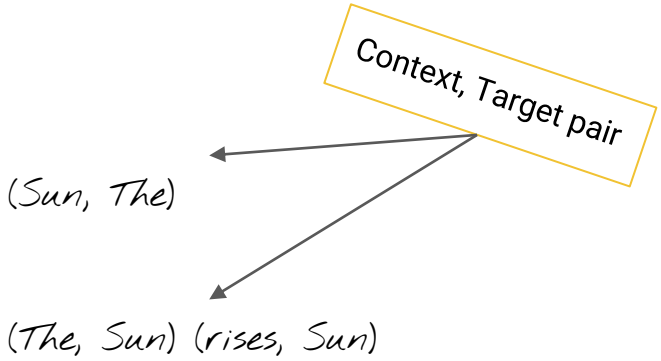
The	Sun	rises	in	the	east

(Sun, The)

Context, Target pair

The	Sun	rises	in	the	east

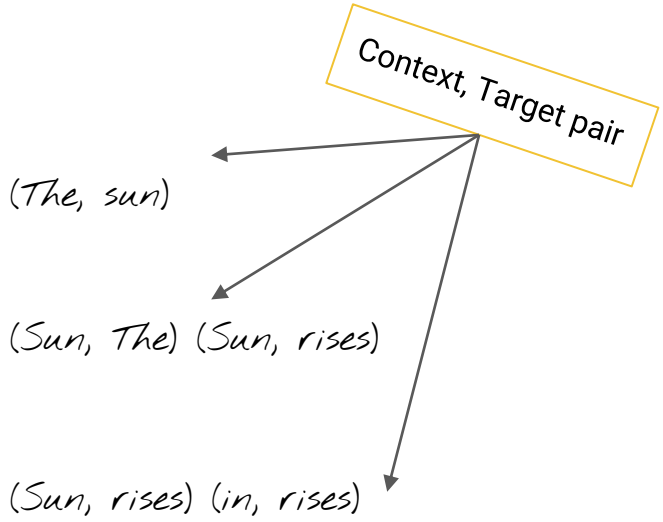
The	Sun	rises	in	the	east



The	Sun	rises	in	the	east

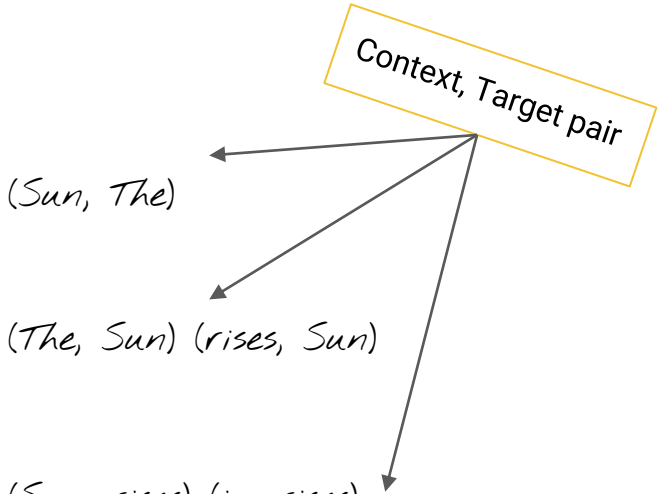
The	Sun	rises	in	the	east

The	Sun	rises	in	the	east



Find probability of a word being 'nearby word'

The	Sun	rises	in	the	east
The	Sun	rises	in	the	east
The	Sun	rises	in	the	east



What is considered near?

Window Size

The	Sun	rises	in	the	east

Window size = 1

(rise, in) (the, in)

Window Size

The	Sun	rises	in	the	east

Window size = 1

(in, rises) (in, the)

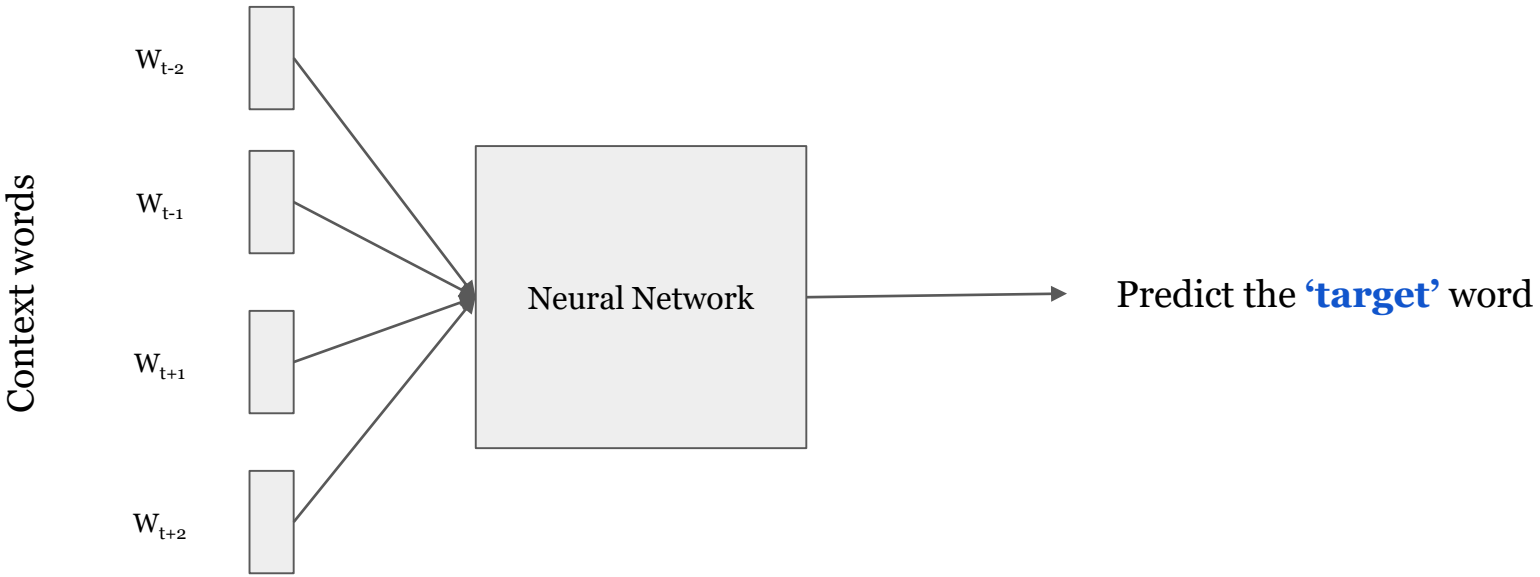
Window size = 2

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

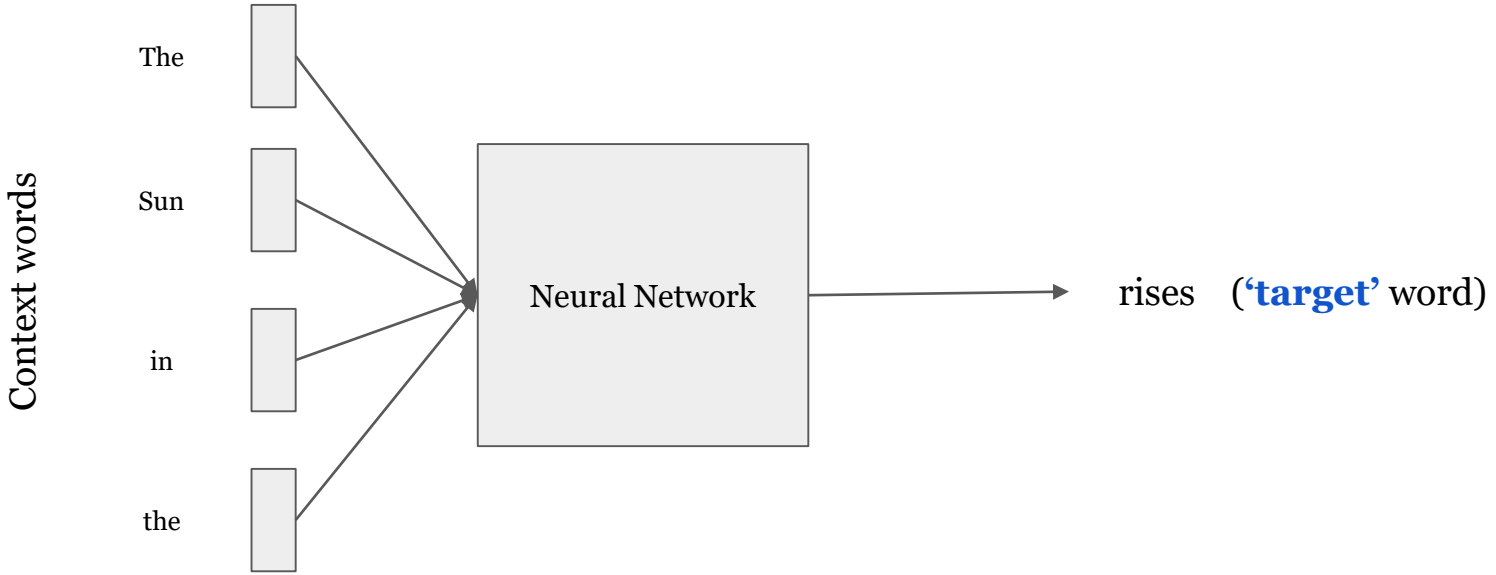
Word2Vec Embeddings

2 ways to get it

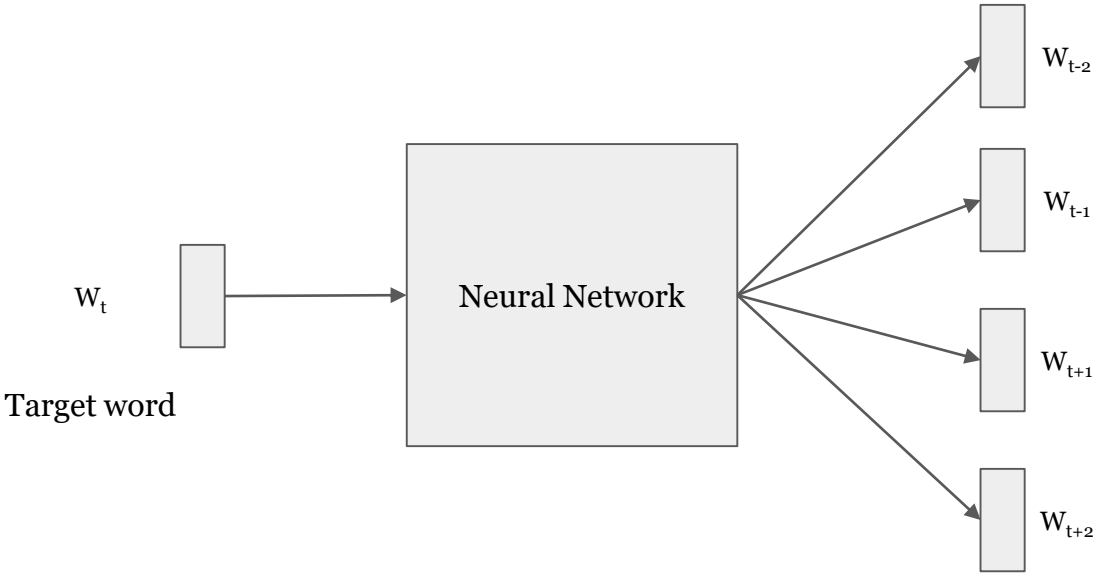
CBOW model



CBOW model

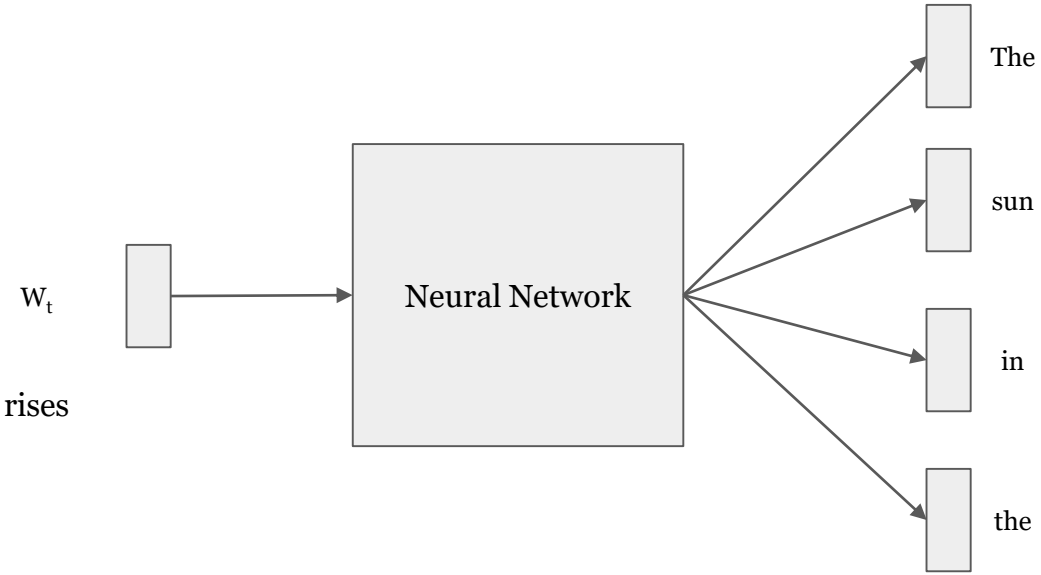


Skip-Gram model



Predict the **'Context'** words

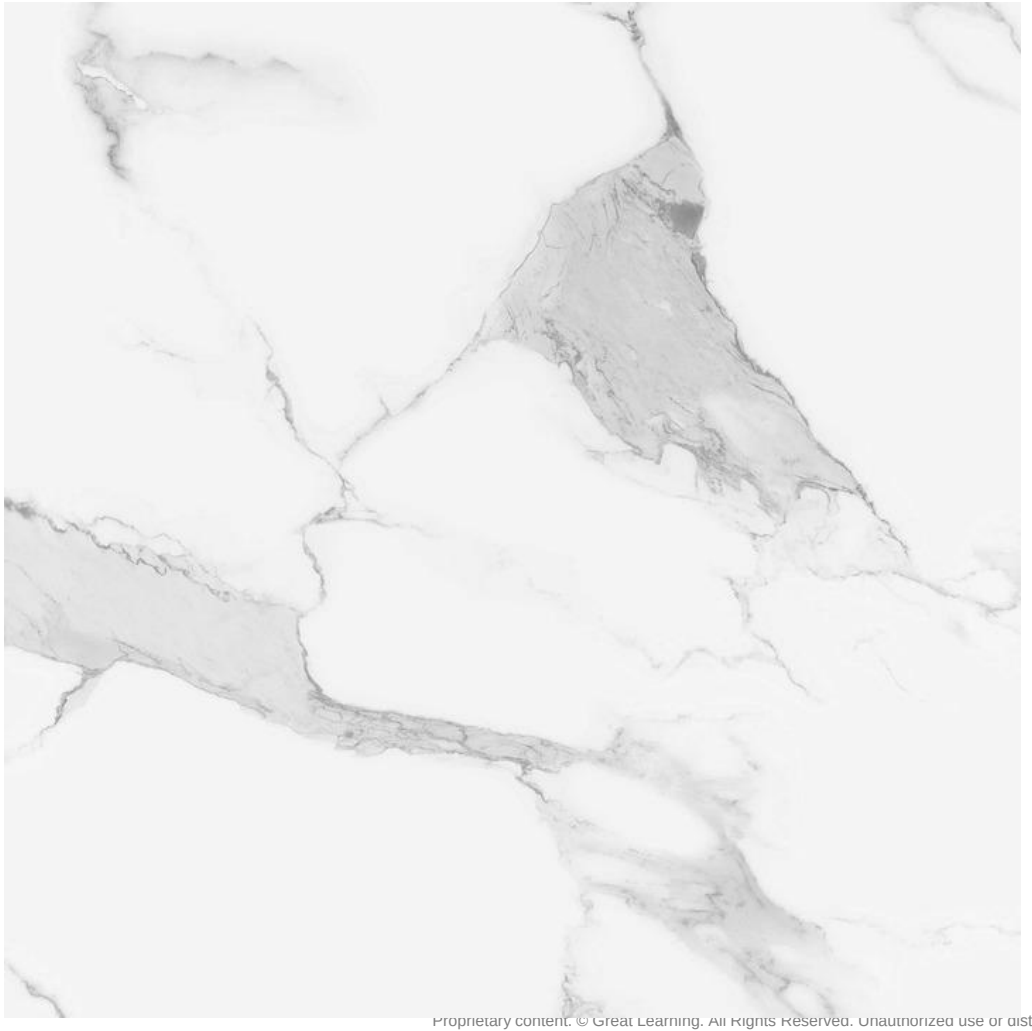
Skip-Gram model



Predict the **'Context'** words

Which approach will work better?

Skip-Gram



greatlearning
Learning for Life

Building

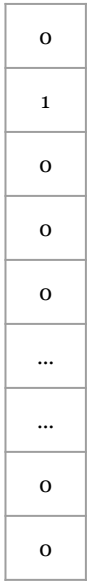
Skip-Gram model

Let’s build a very simple Neural network

1 Input Layer, 1 Hidden Layer, 1 Output Layer

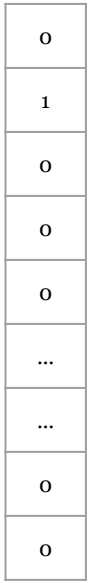
The Input Layer

Input word as One hot



W_t

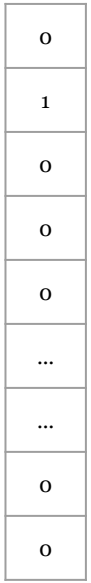
Input word as One hot



W_t

Size of the input vector?

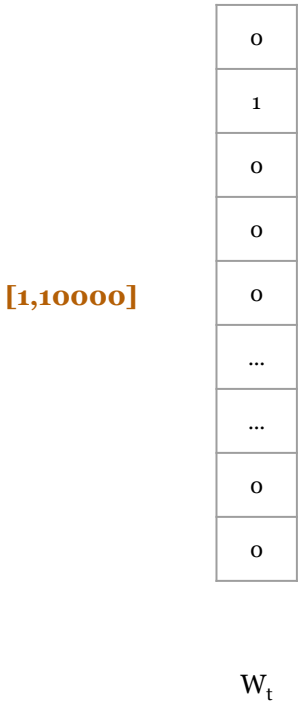
Input word as One hot



W_t

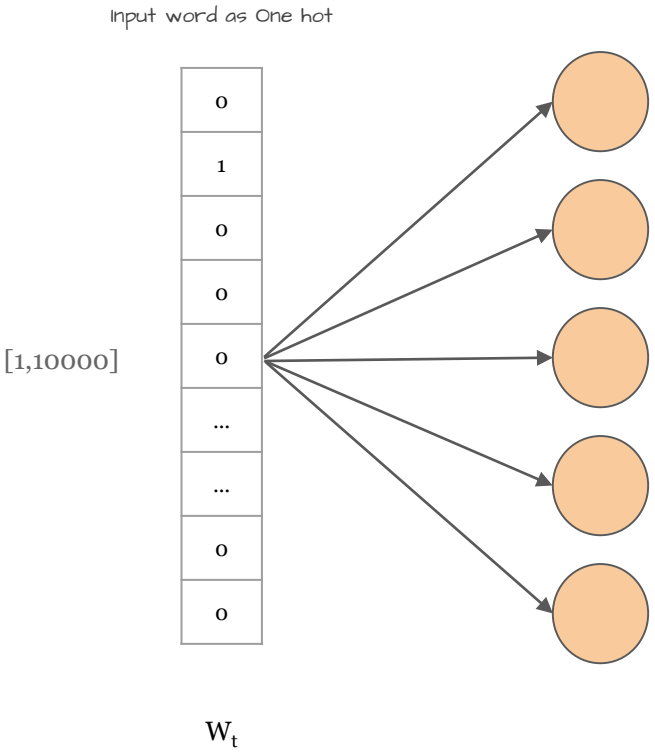
Same as vocabulary size

Input word as One hot

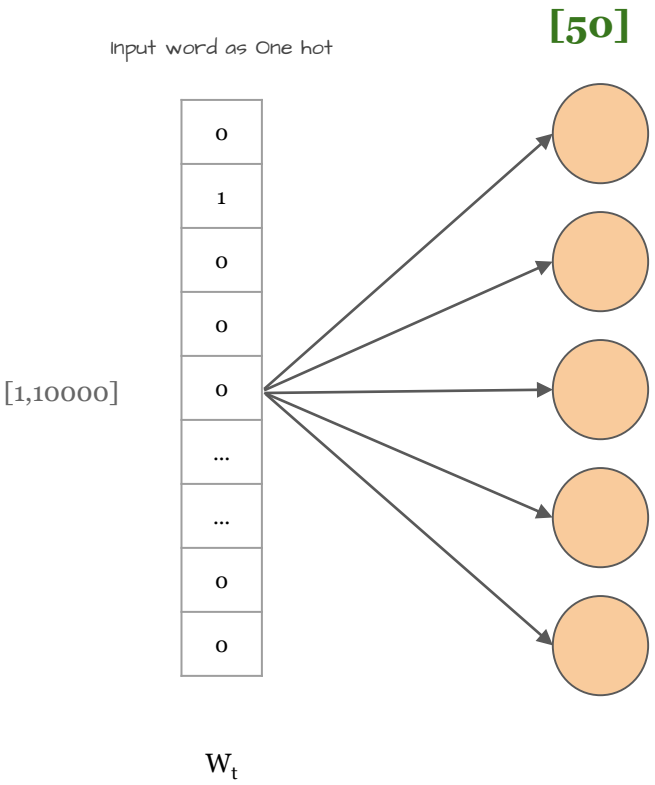


Assume we have 10000 words vocabulary

The Hidden Layer

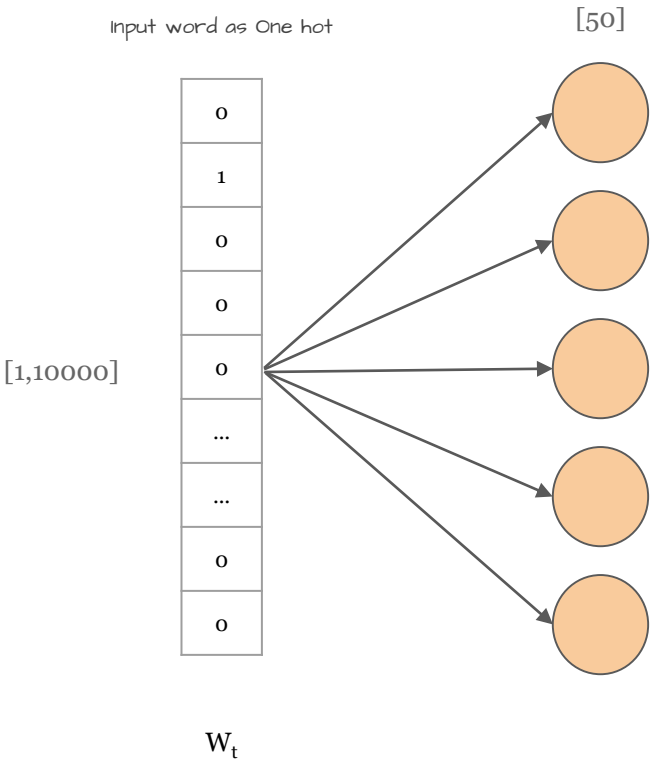


How many neurons in hidden layer?

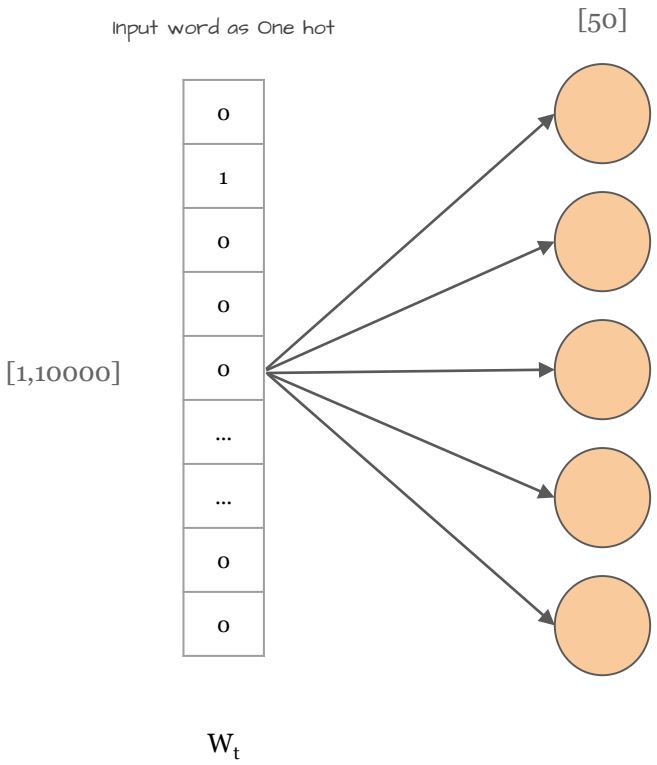


Let's have 50

(or whatever you like)



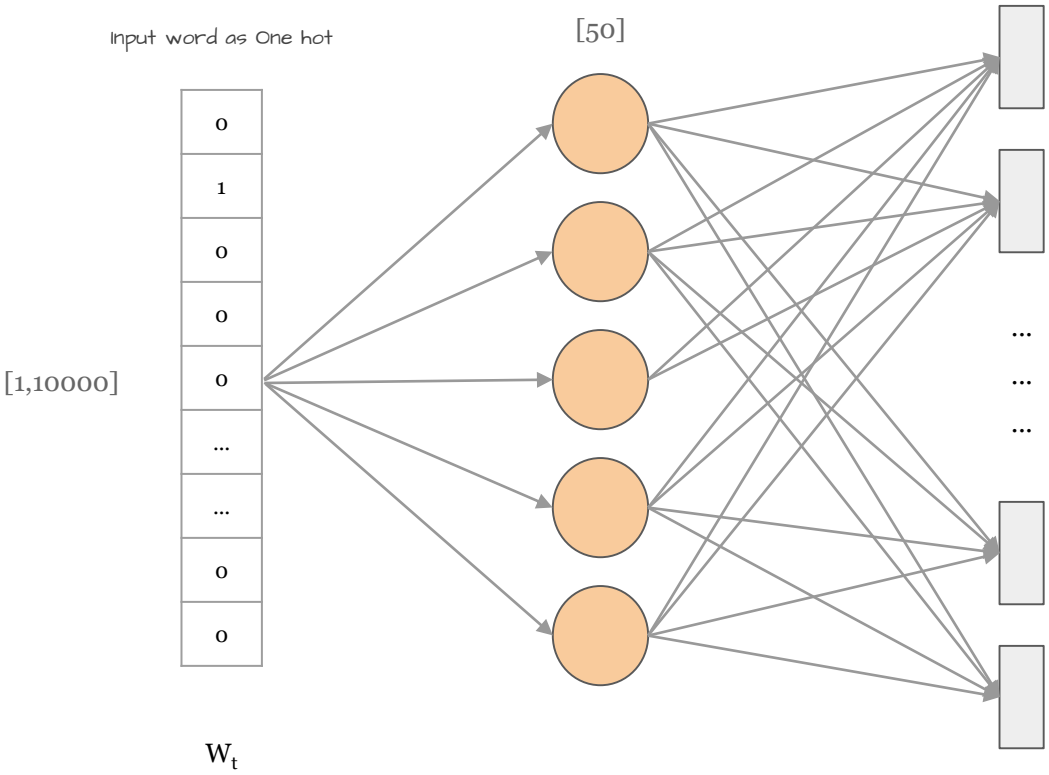
How many hidden layer outputs for each Word?



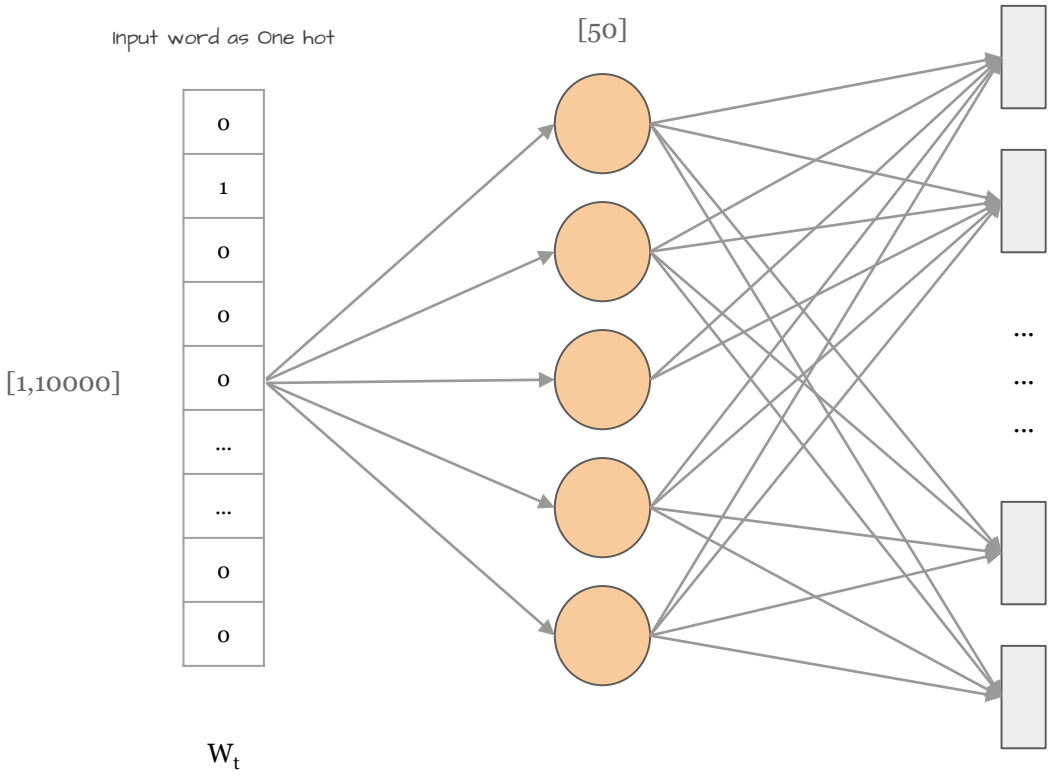
50

Same as number of neurons in hidden layer

The Output Layer

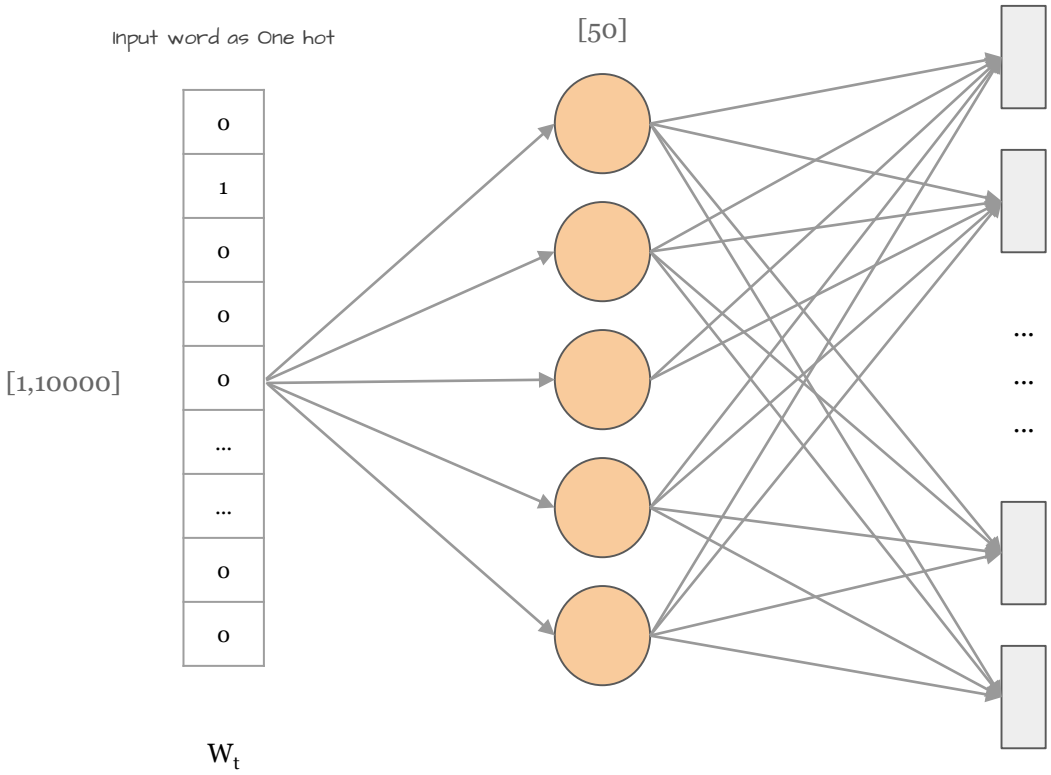


How many Outputs?



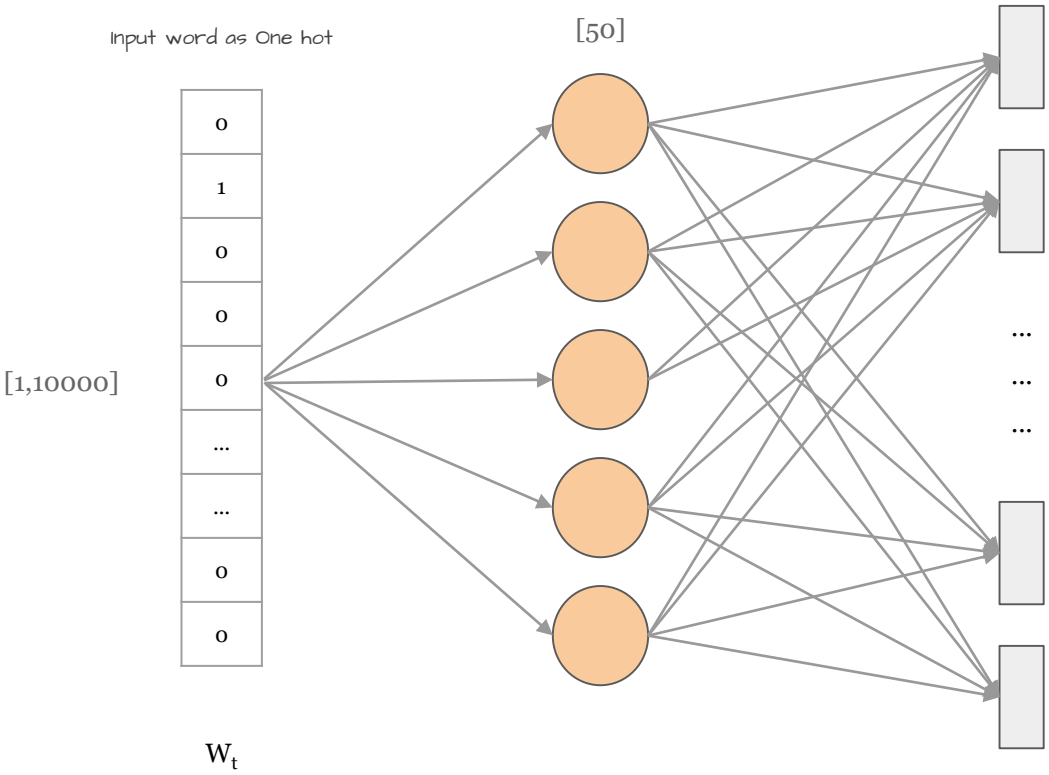
10,000

Same as Vocabulary size



10,000

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



10,000

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

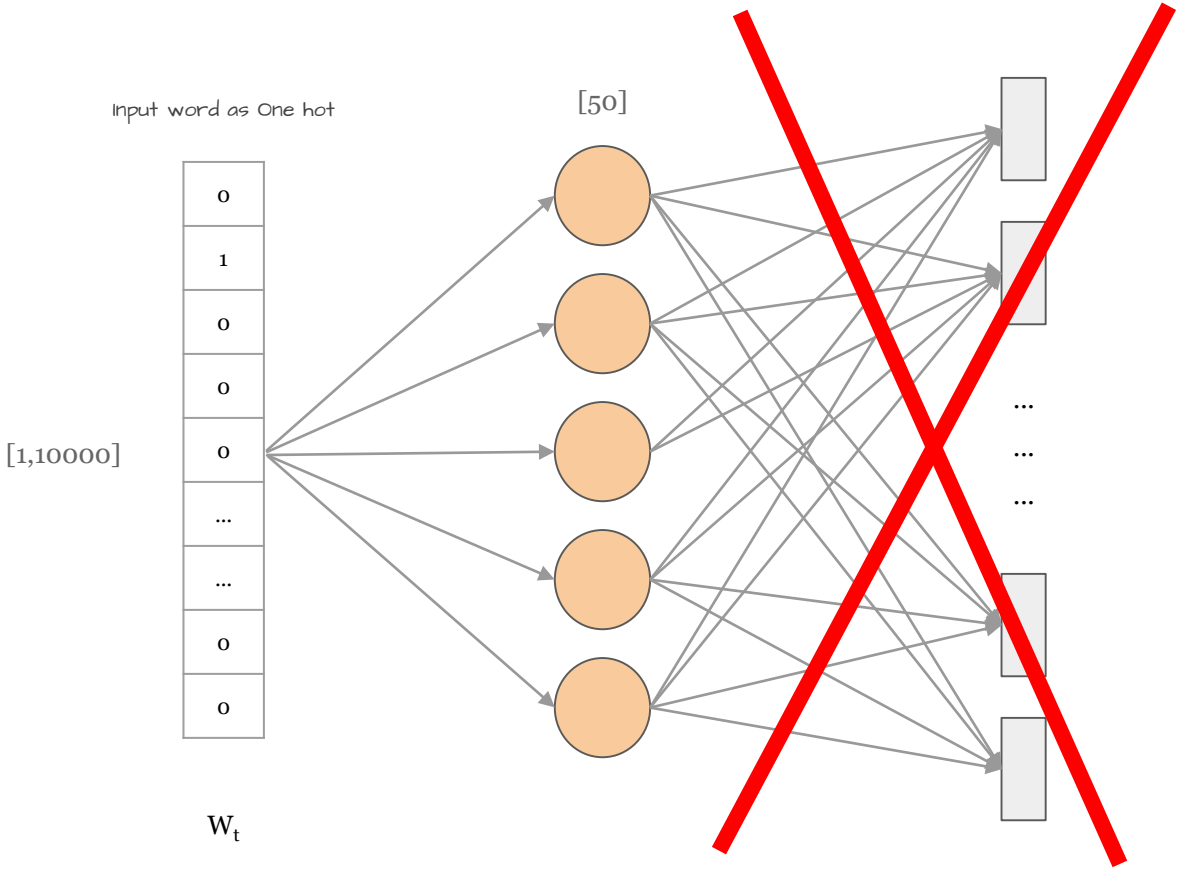
NEGATIVE SAMPLING

Negative Sampling

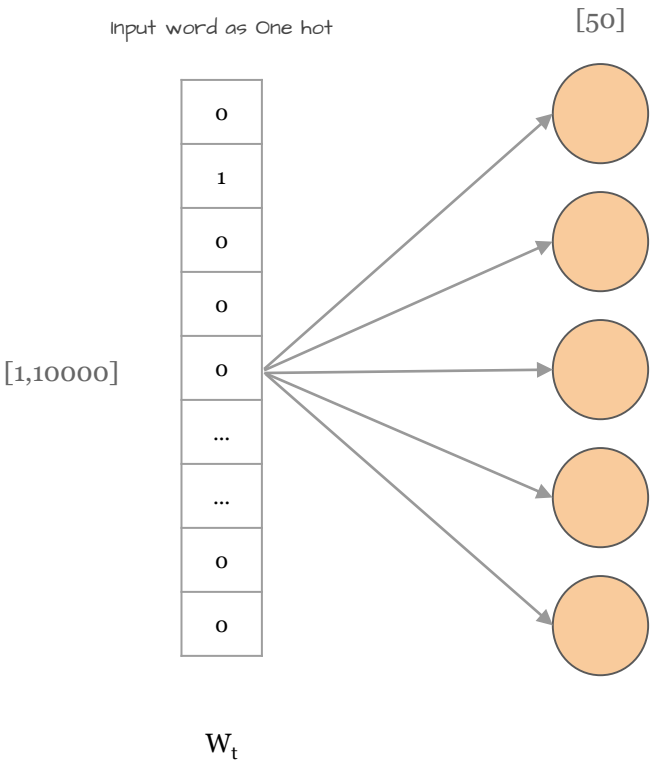
Only a few weights are updated

1. Weights corresponding to Positive outputs (Window Size)
2. Very small number of weights for Negative output
    a. 5-20 for small datasets
    b. 2-5 for Large dataset

What happens after training?

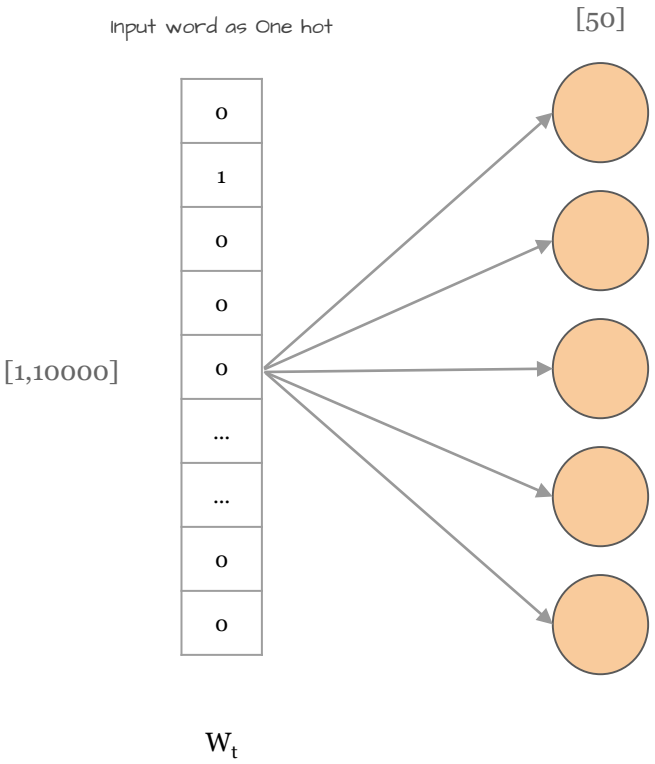


Output Layer is discarded



For each Word in vocabulary, we get...

50 numbers



For each Word in vocabulary, we get...

50 numbers

Embedding Size

This is how we convert words into numbers...

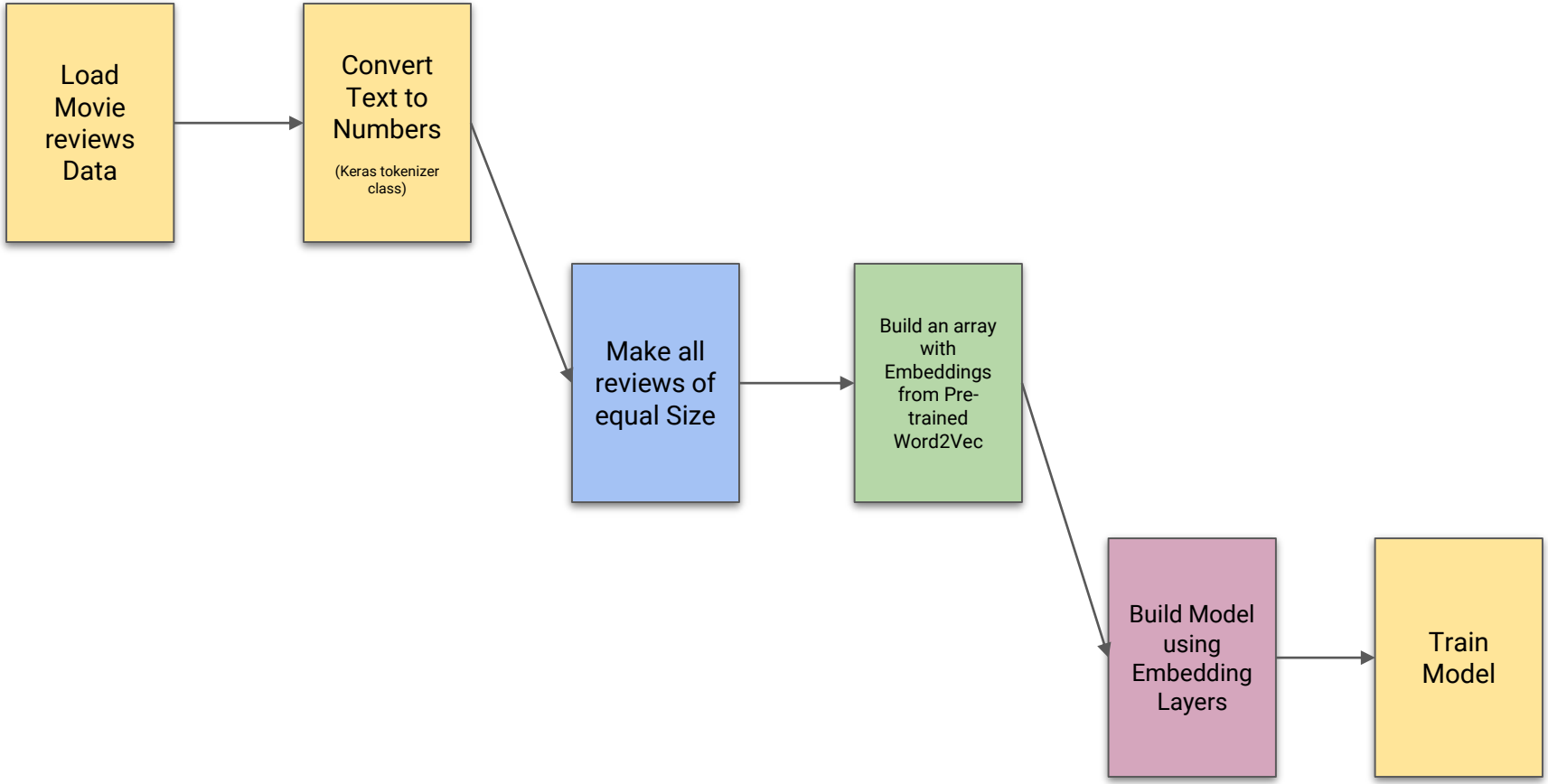
and discover semantic relationships between words

Building Word2Vec Model

Using gensim

Using Pre-Trained Word2Vec model

Sentiment Analysis



Keras Embedding Layer

Input_dim → Possible Input values

Output_dim → How many numbers for each Input value

Input_length → How many input numbers in each Example

Weights → Pre-trained Embeddings, if any