Word Embeddings

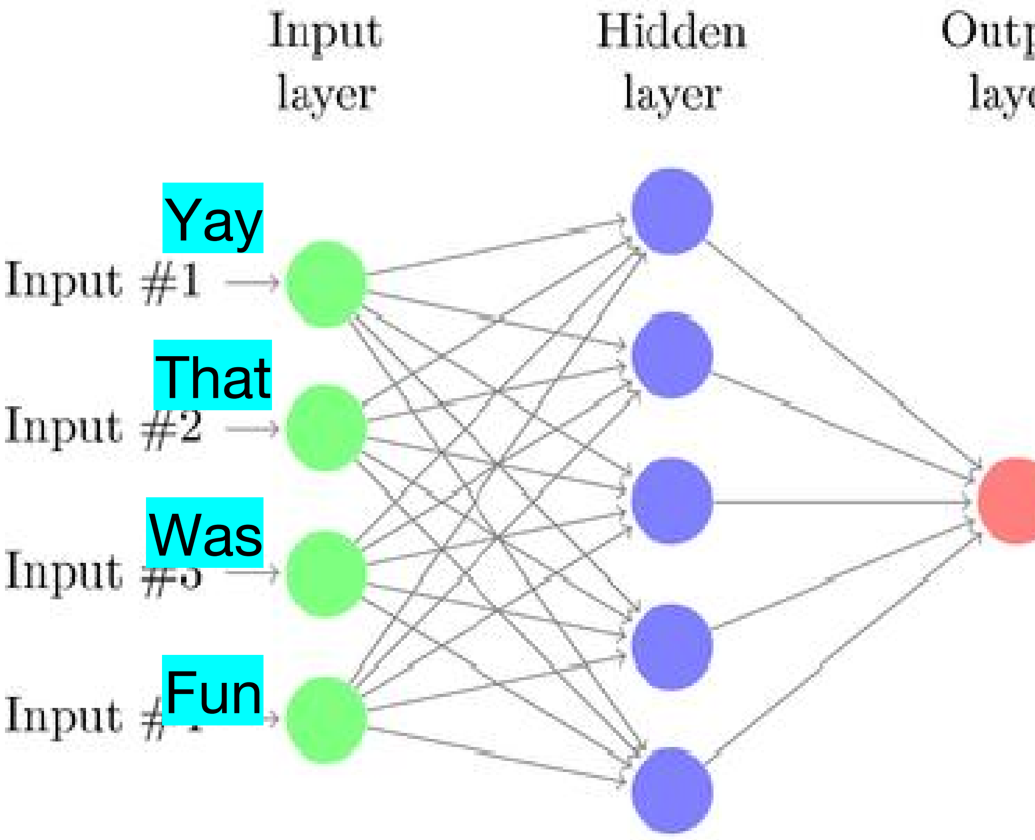
Michael Shao

Problem: encoding words

- Recall the example from last class:

- emotion detection from sentence

How can we input the words into the NN?



Problem: encoding words

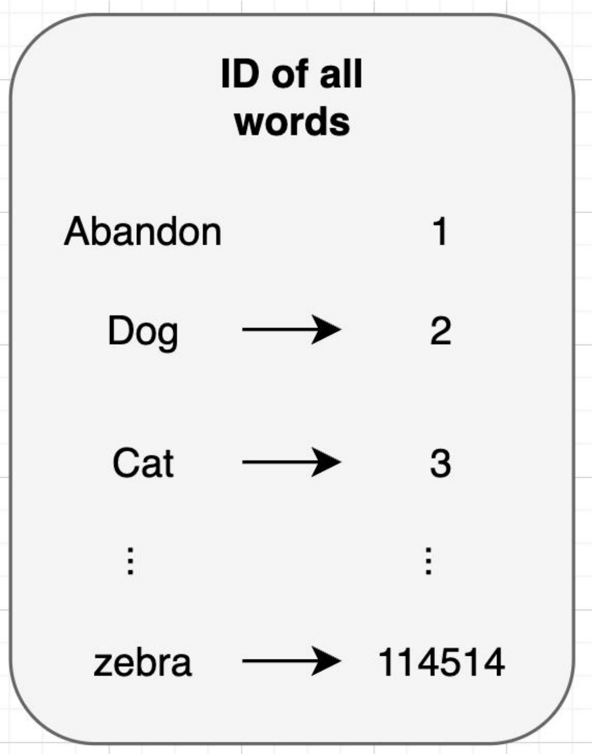
- We need a way of turning words into numbers so that our NNs can process them

Solution: Word embeddings

- (May remove later)

1st Solution: ID Assignment

- The most simple way we can think of is just assigning a number to each word and using that for the input.



ID of all words

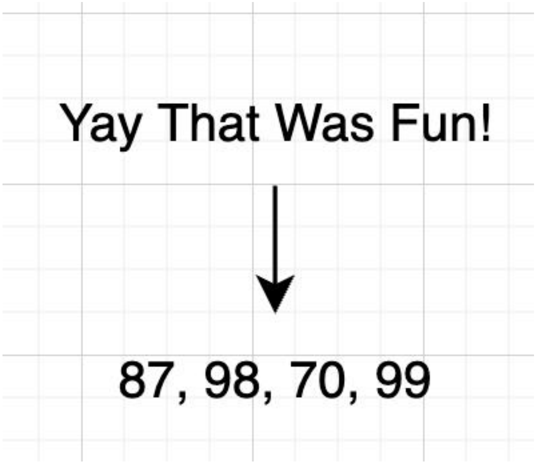
Abandon 1

Dog → 2

Cat → 3

⋮ ⋮

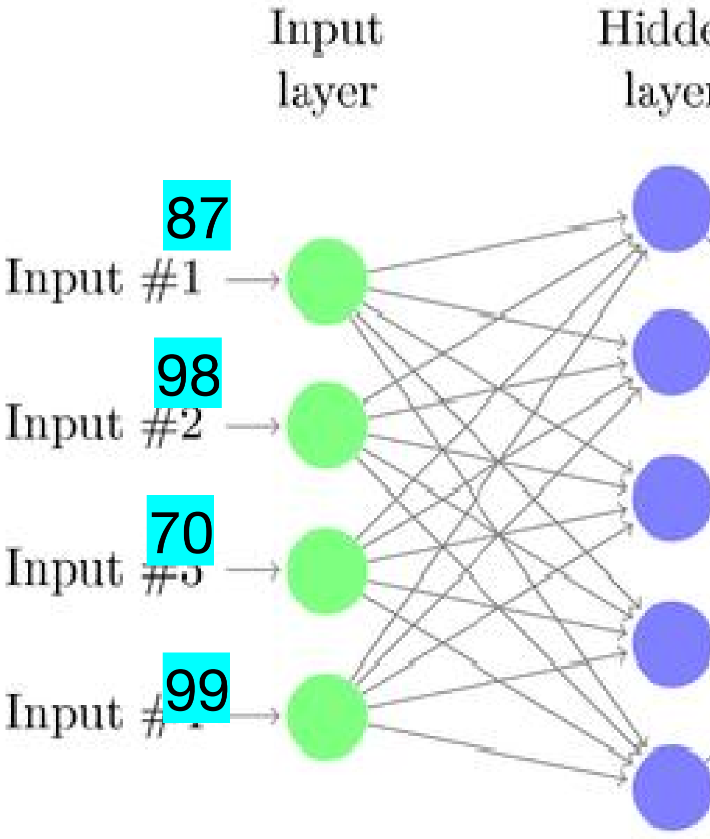
zebra → 114514



Yay That Was Fun!

↓

87, 98, 70, 99



2nd Solution: Bag of words(BoW)

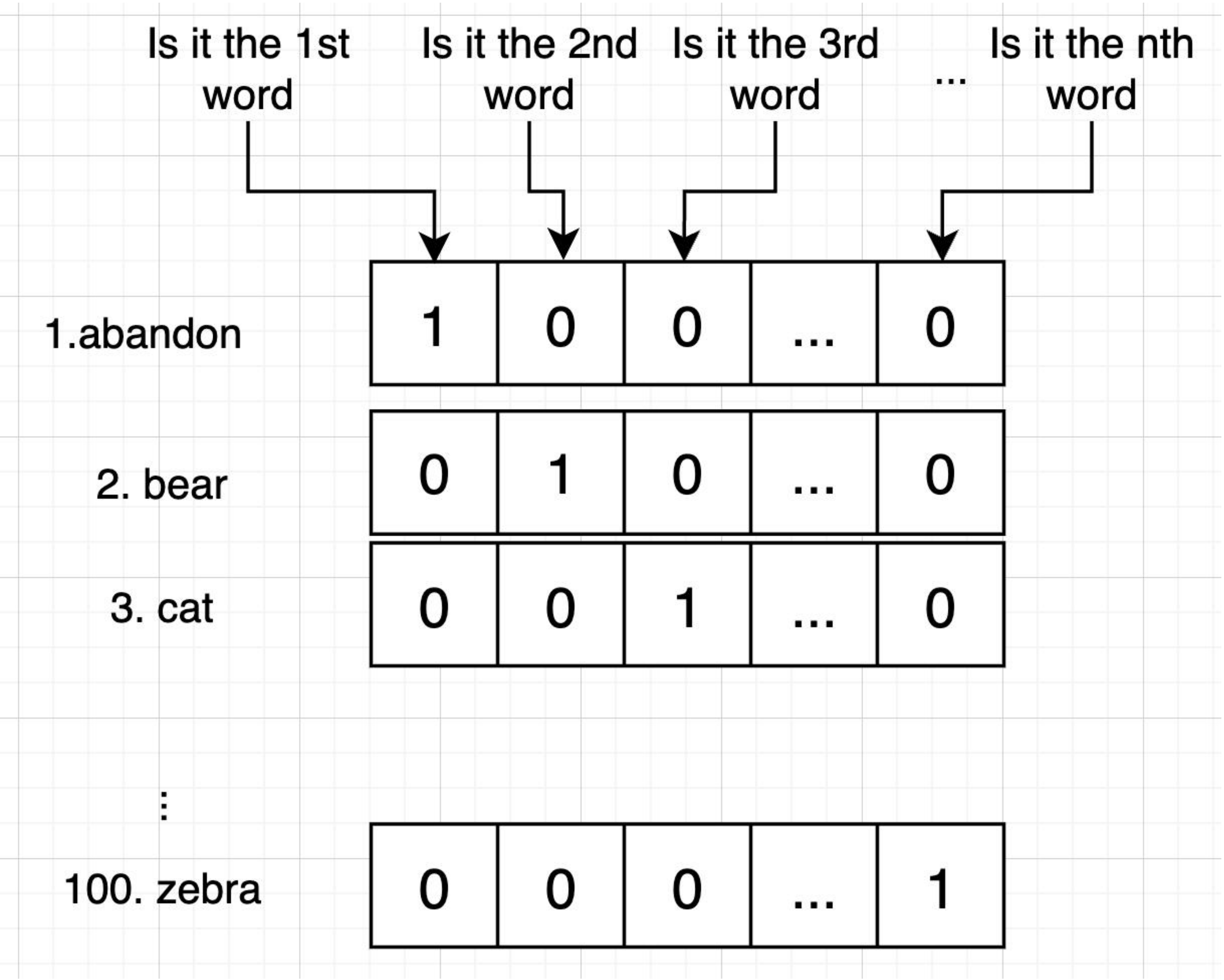
- (show how we can encode a sentence using bag of words)
- (not exactly a vector representation of the word, does this for the whole sentence)
- (Might remove)

Problem with 1st Solution

- Suppose we have these two words and their IDs:
- Yay: 87
- Yes: 10000
- The problem here is the ID for words are not even in the same magnitude, which is really bad for the NN.

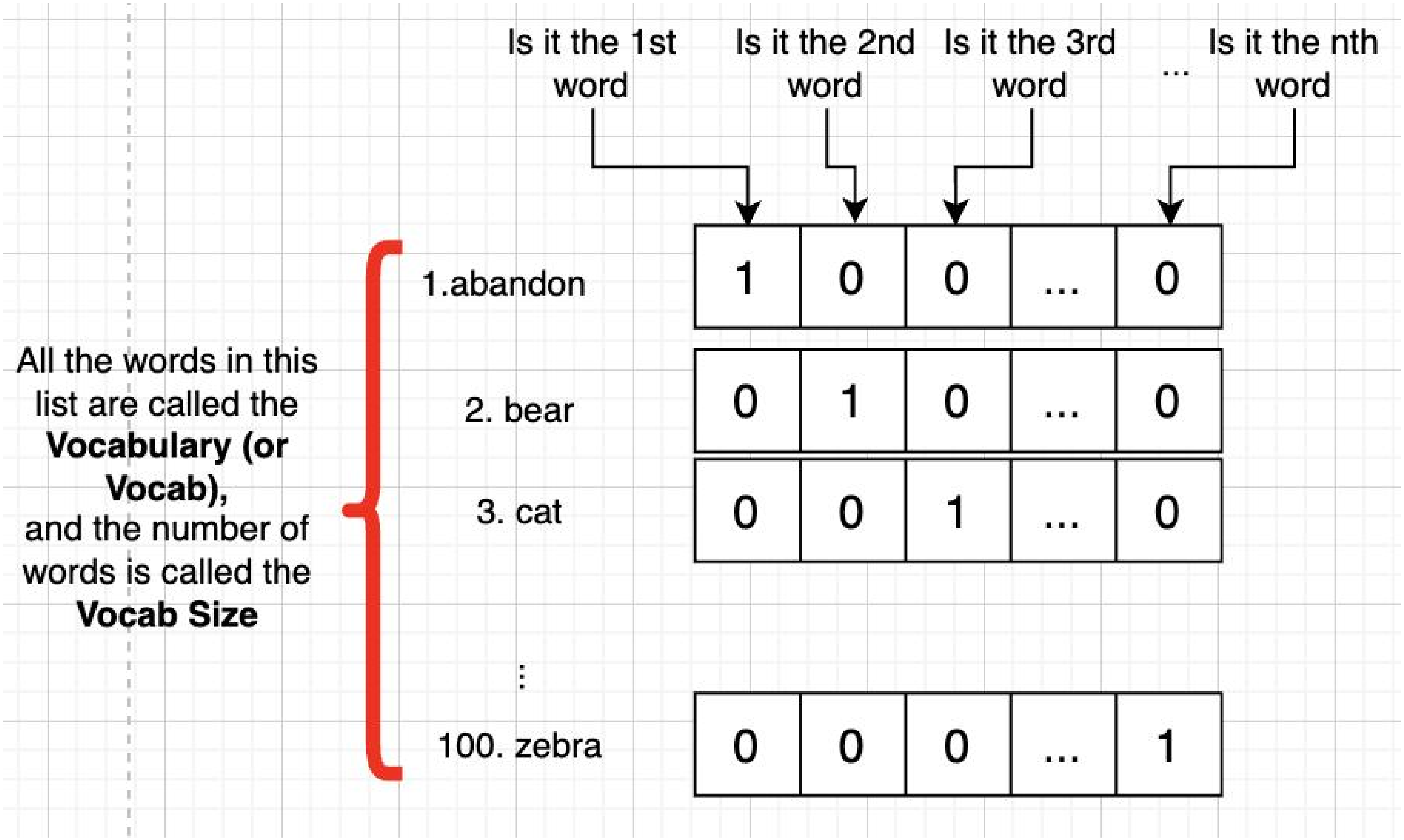
2nd Solution: One-hot encoding

- One hot encoding uses a vector, which is a list of numbers, to represent a word.



	Is it the 1st word	Is it the 2nd word	Is it the 3rd word	...	Is it the nth word
1.abandon	1	0	0	...	0
2. bear	0	1	0	...	0
3. cat	0	0	1	...	0
⋮					
100. zebra	0	0	0	...	1

2nd Solution: One-hot encoding



All the words in this list are called the **Vocabulary (or Vocab),** and the number of words is called the **Vocab Size**

	Is it the 1st word	Is it the 2nd word	Is it the 3rd word	...	Is it the nth word
1.abandon	1	0	0	...	0
2. bear	0	1	0	...	0
3. cat	0	0	1	...	0
⋮					
100. zebra	0	0	0	...	1

2nd Solution: One-hot encoding

- Problem with One-Hot Encoding:
- Suppose we have these two word vectors:
- [0, 0, 0, 1, 0] and [1, 0, 0, 0, 0]
- can you tell what they mean?

2nd Solution: One-hot encoding

- One-hot encoding offers a good way of indexing and inputting the words, but it does not carry the semantic meaning of the words.

Encoding Semantic Meaning of Words

- Look at this word encoded with one hot encoding:

- Hello: [0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]

- It seems a bit wasteful using such a big vector to only encode so little information.

Encoding Semantic Meaning of Words

- Instead, we can use the different dimensions to represent different properties of the word.

- https://www.cs.cmu.edu/~dst/WordEmbeddingDemo/tutorial.html

Encoding Semantic Meaning of Words

- Instead, we can use the different dimensions to represent different properties of the word.

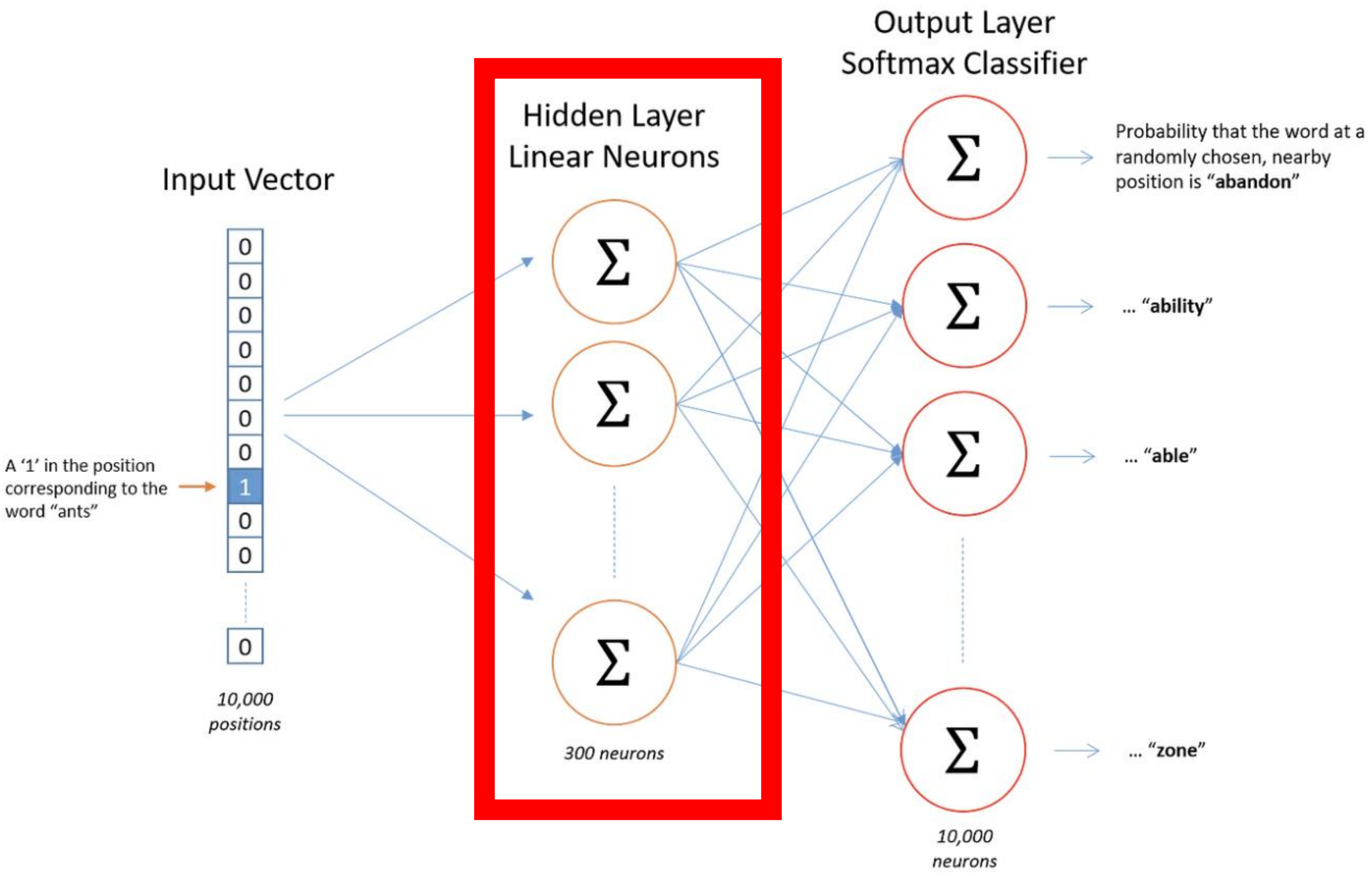
- https://www.cs.cmu.edu/~dst/WordEmbeddingDemo/tutorial.html

How to we get these vectors: Word2Vec

- But setting these values by hand is next to impossible when the vector gets bigger.

- We need a way to automatically find these embedding vectors.

Basic Idea of Word2Vec

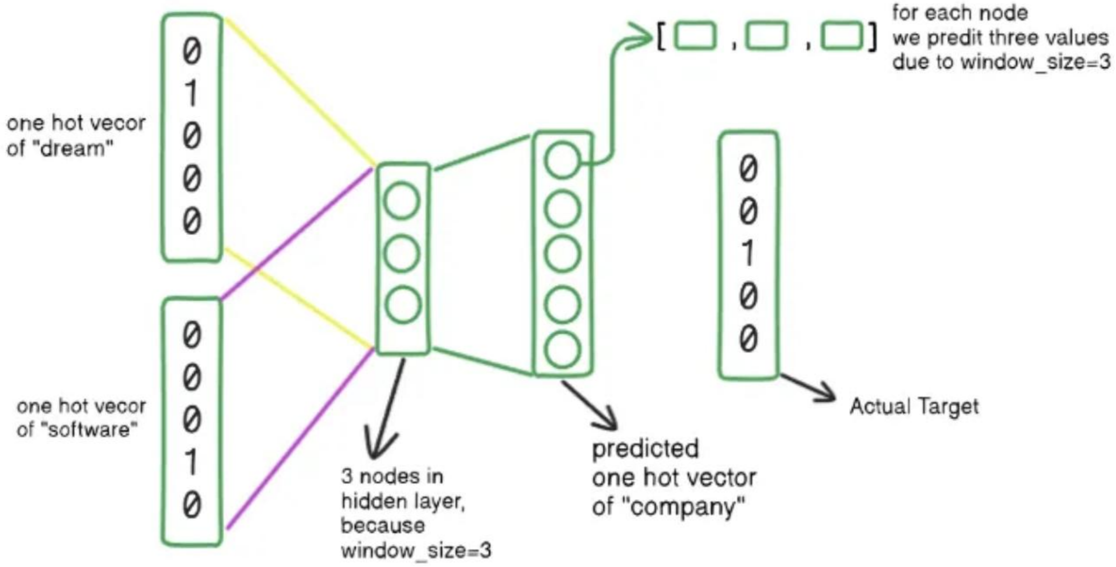


Word2Vec: CBOW

"google dream company software engineer"

Independent Features		o/p
dream	software	company

google	dream	company	software	engineer
[0 ,	1,	0,	0,	0]
[0 ,	0,	0,	1,	0]

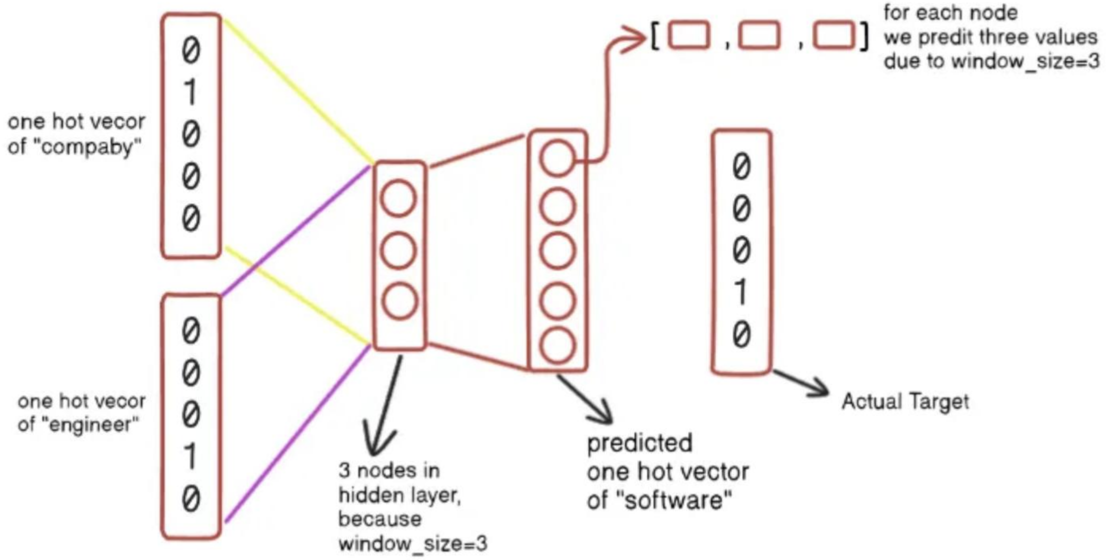


[□ , □ , □]
for each node we predit three values due to window_size=3
one hot vecor of "dream"
0 1 0 0 0
one hot vecor of "software"
0 0 0 1 0
3 nodes in hidden layer, because window_size=3
predicted one hot vector of "company"
0 0 1 0 0
Actual Target

"google dream company software engineer"

Independent Features		o/p
company	engineer	software

google	dream	company	software	engineer
[0 ,	0,	1,	0,	0]
[0 ,	0,	0,	0,	1]



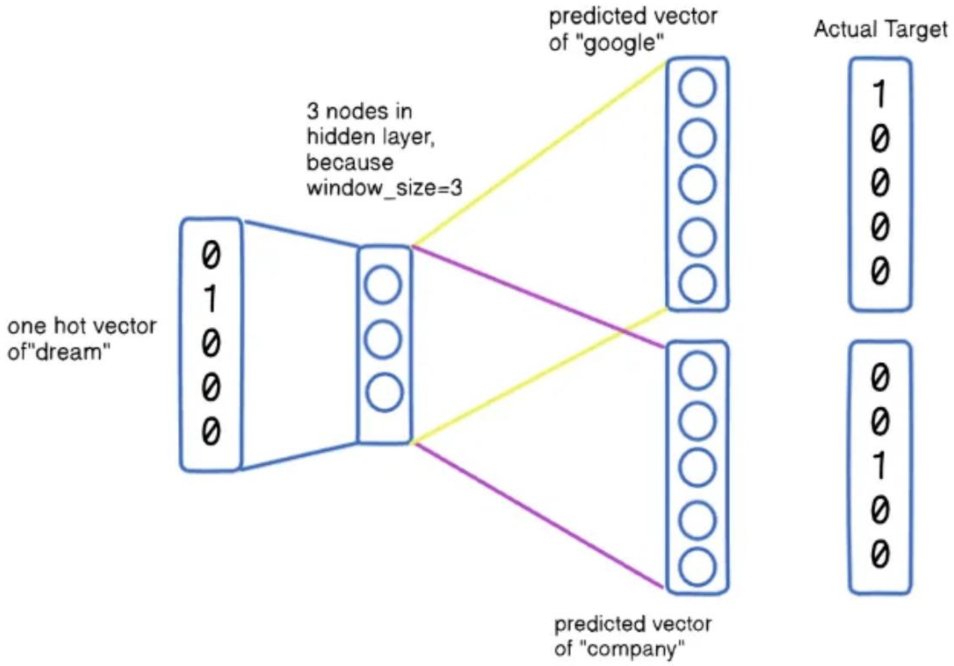
[□ , □ , □]
for each node we predit three values due to window_size=3
one hot vecor of "compaby"
0 1 0 0 0
one hot vecor of "engineer"
0 0 0 1 0
3 nodes in hidden layer, because window_size=3
predicted one hot vector of "software"
0 0 0 1 0
Actual Target

Word2Vec: Skipgram

"google dream company software engineer"

Independent Features		o/p
google	company	dream

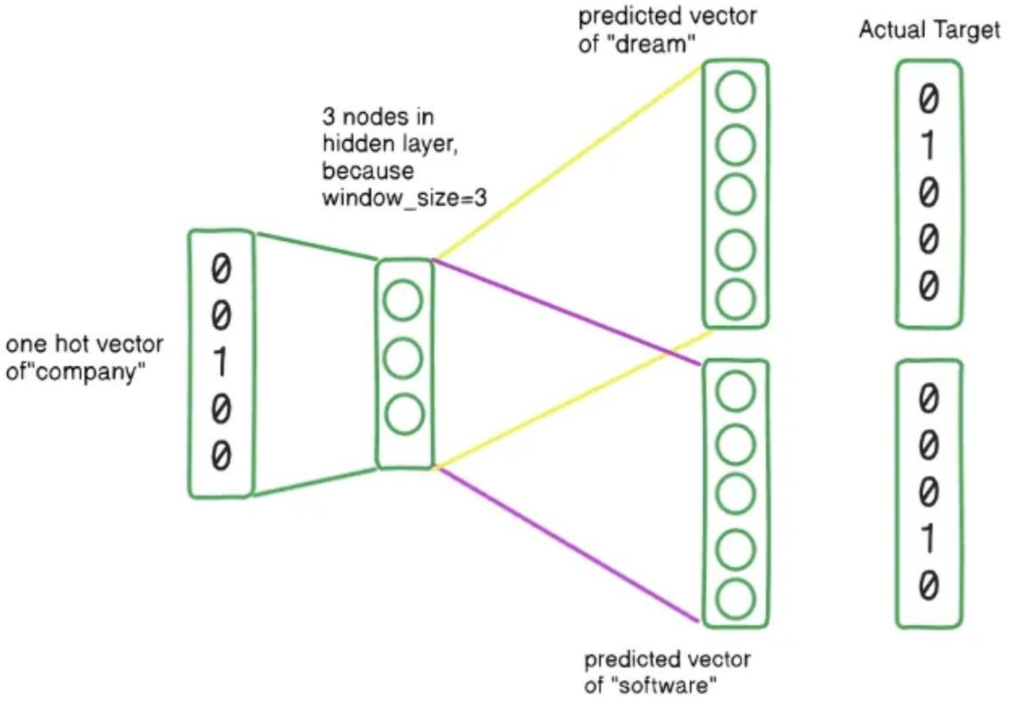
google	dream	company	software	engineer
[1,	0,	0,	0,	0]
[0,	0,	1,	0,	0]


predicted vector of "google"
Actual Target
3 nodes in hidden layer, because window_size=3
one hot vector of"dream"
0 1 0 0 0
1 0 0 0 0
0 0 1 0 0
predicted vector of "company"

"google dream company software engineer"

Independent Features		o/p
dream	software	company

google	dream	company	software	engineer
[0,	1,	0,	0,	0]
[0,	0,	0,	1,	0]


predicted vector of "dream"
Actual Target
3 nodes in hidden layer, because window_size=3
one hot vector of"company"
0 0 1 0 0
0 1 0 0 0
0 0 0 1 0
predicted vector of "software"