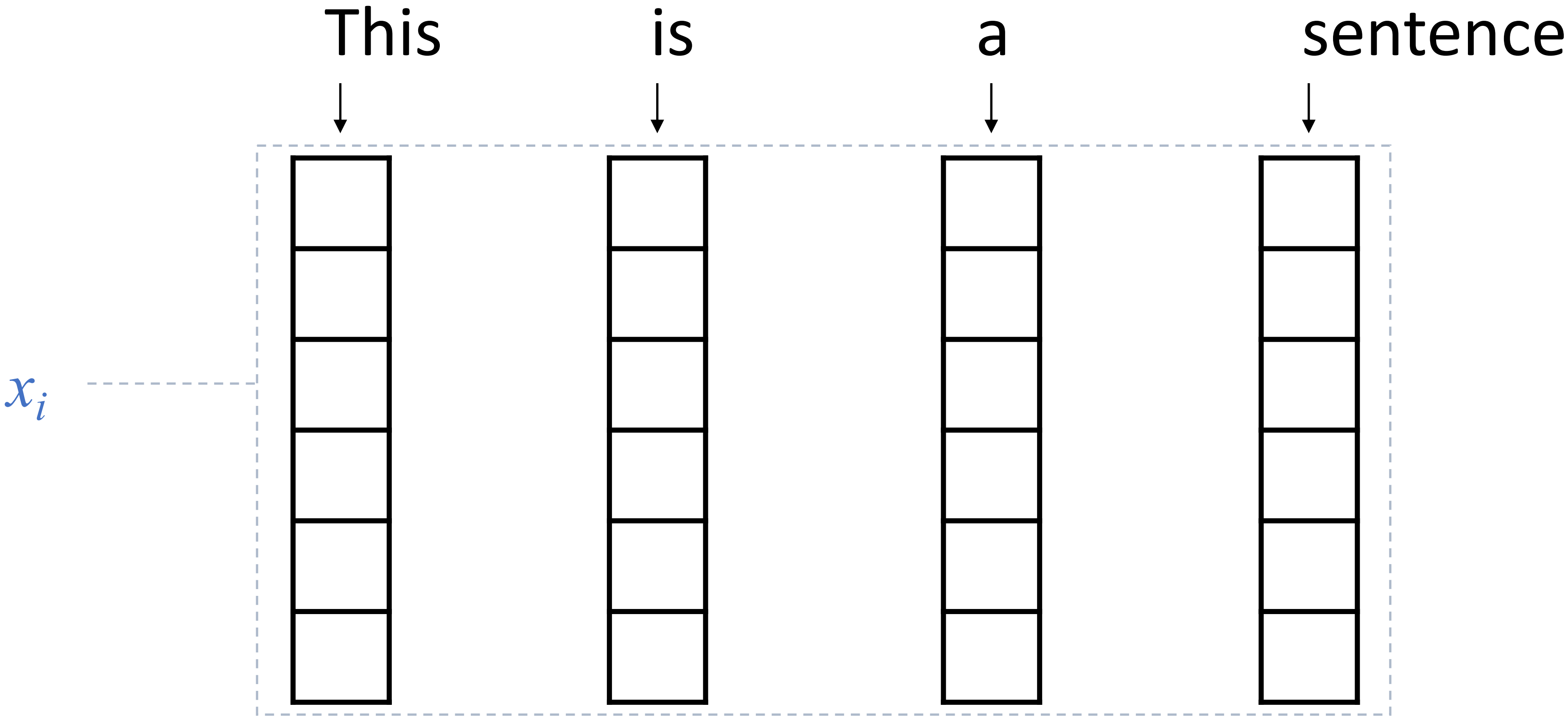
Learning Word Embeddings

MMCi Block 5

Matthew Engelhard

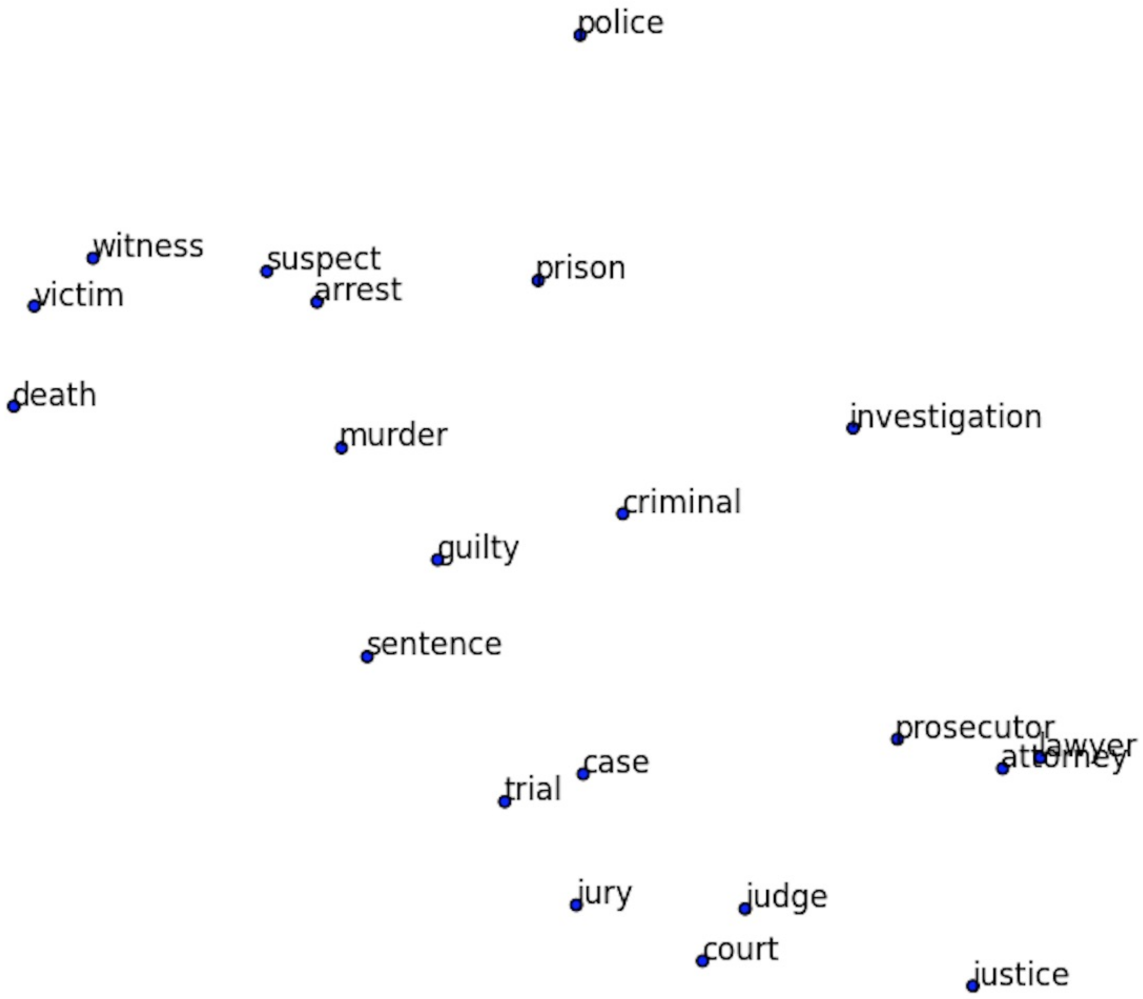
From sentence to sequence of vectors.

- Look up words individually to obtain their vectors
- But where do the vectors come from?



Visualizing Word Embeddings

If the meaning is similar, the vectors (i.e. locations) should be similar!



How are word embeddings learned?

KEY IDEA: words are *defined* by the context in which they appear

A man strolls down the street

A woman strolls down the street

A child strolls down the street

A crocodile strolls down the street

A banana strolls down the street

A concept strolls down the street

KEY IDEA: words are *defined* by the <u>context</u> in which they appear

-> if words are always exchangeable, they must have very similar meaning

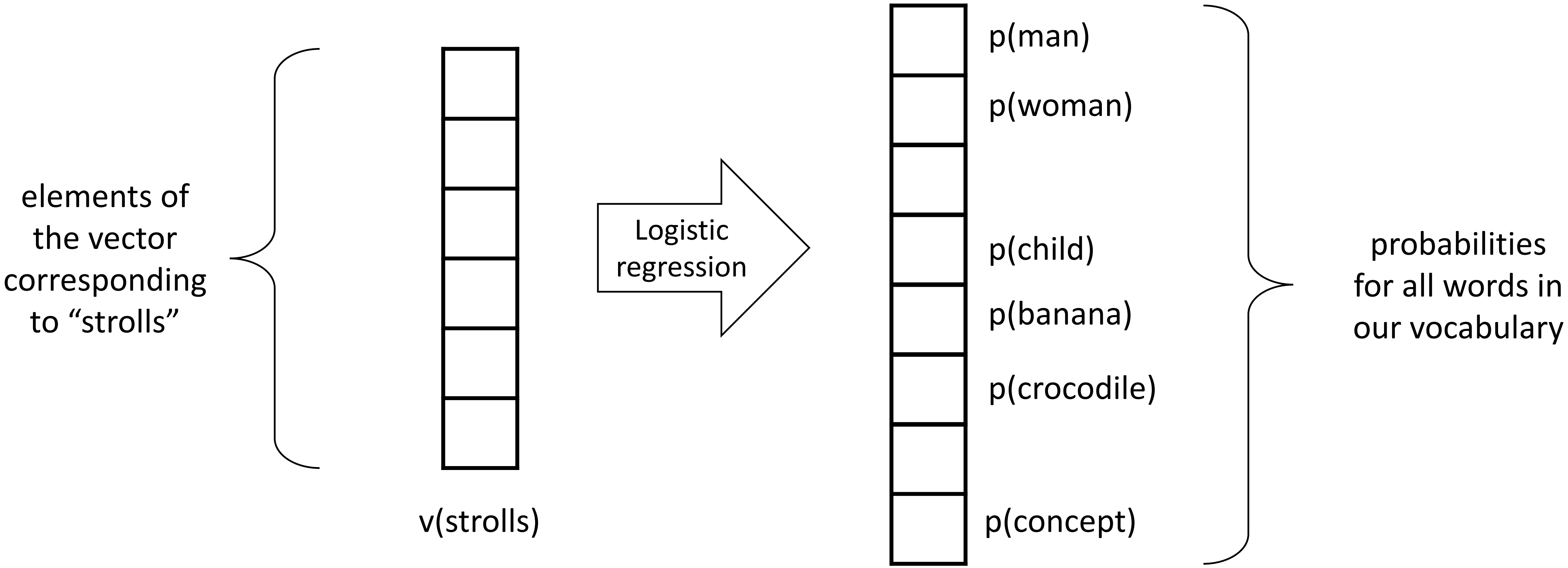
learn word meaning like an adult:
explicit definitions

https://www.parenting.com/activities/baby/teach-baby-to-talk/

learn word meaning like an child:
<u>implicit definitions from context</u>

So how do we learn spatial locations for each word?

We want: a vector for each word that allows us to predict its context (i.e. what other words are likely/unlikely to be around it)


elements of the vector corresponding to “strolls”
v(strolls)
Logistic regression
p(man)
p(woman)
p(child)
p(banana)
p(crocodile)
p(concept)
probabilities for all words in our vocabulary

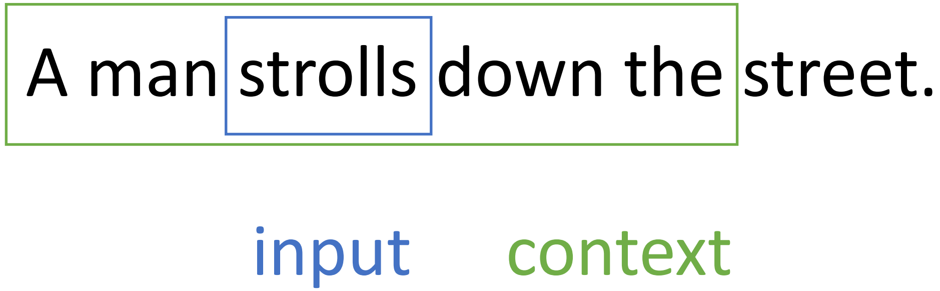
Predict Context Words from Input Words

{input word, context word}

{strolls, man}
{strolls, woman}
{swims, crocodile}
{swims, fish}
{flies, bird}
{flies, plane}

We define a context word as one that appears inside a fixed-length window around the input word in our training corpus.

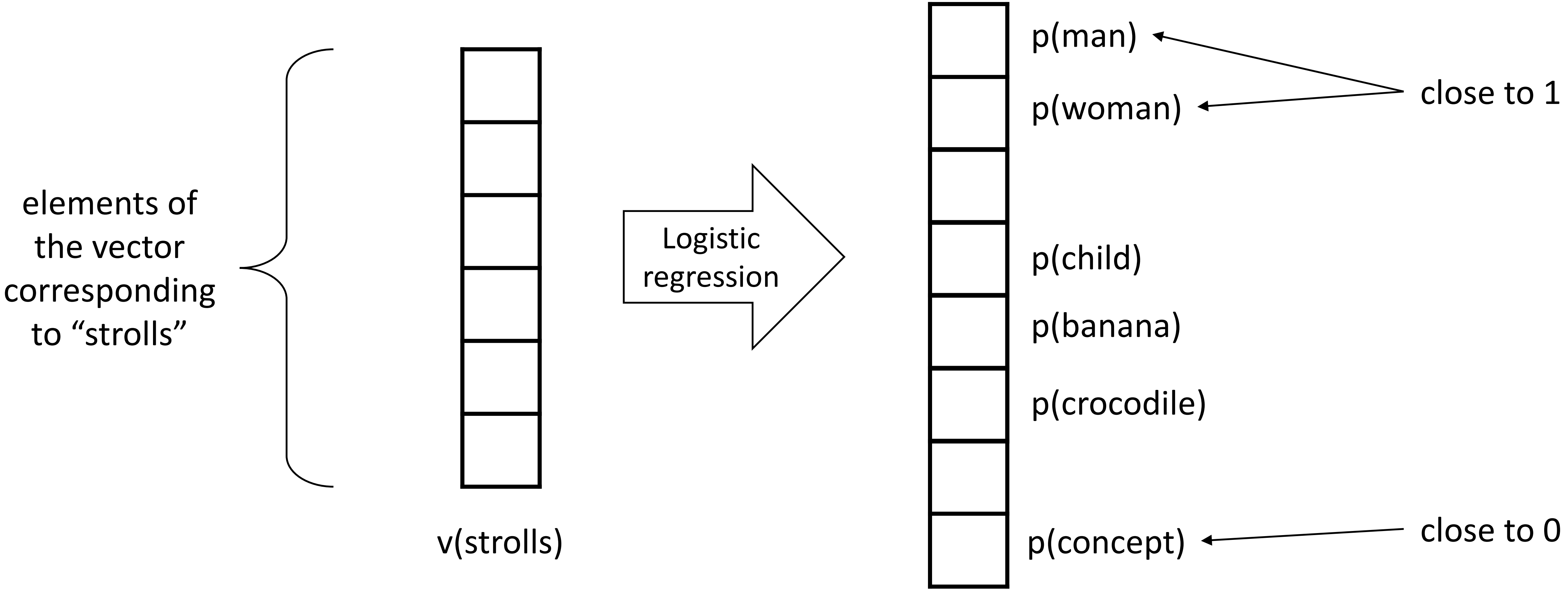
(e.g. Wikipedia)



A man strolls down the street.

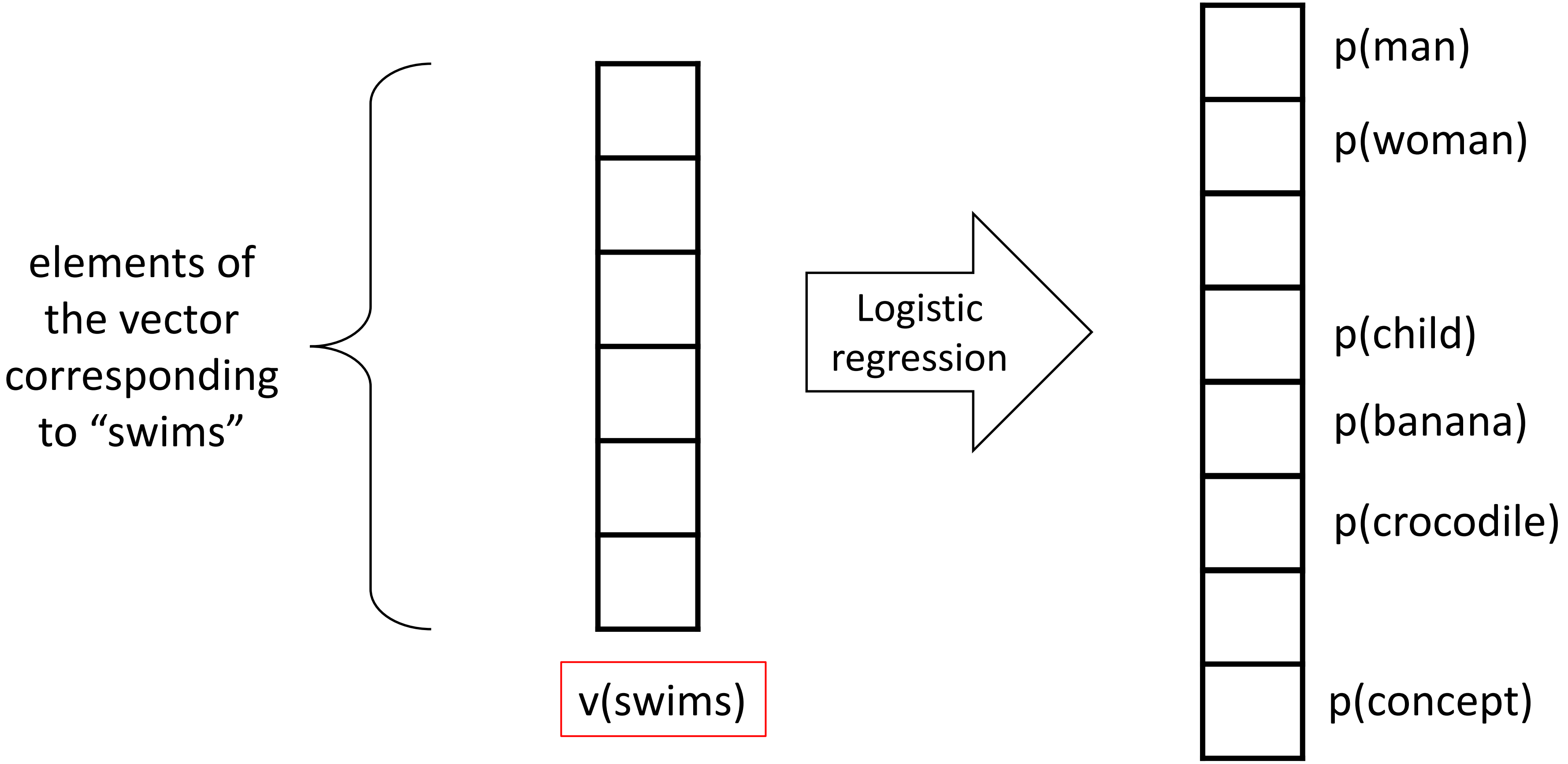
So how do we learn spatial locations for each word?

We want: a vector for each word that allows us to predict its context (i.e. what other words are likely/unlikely to be around it)



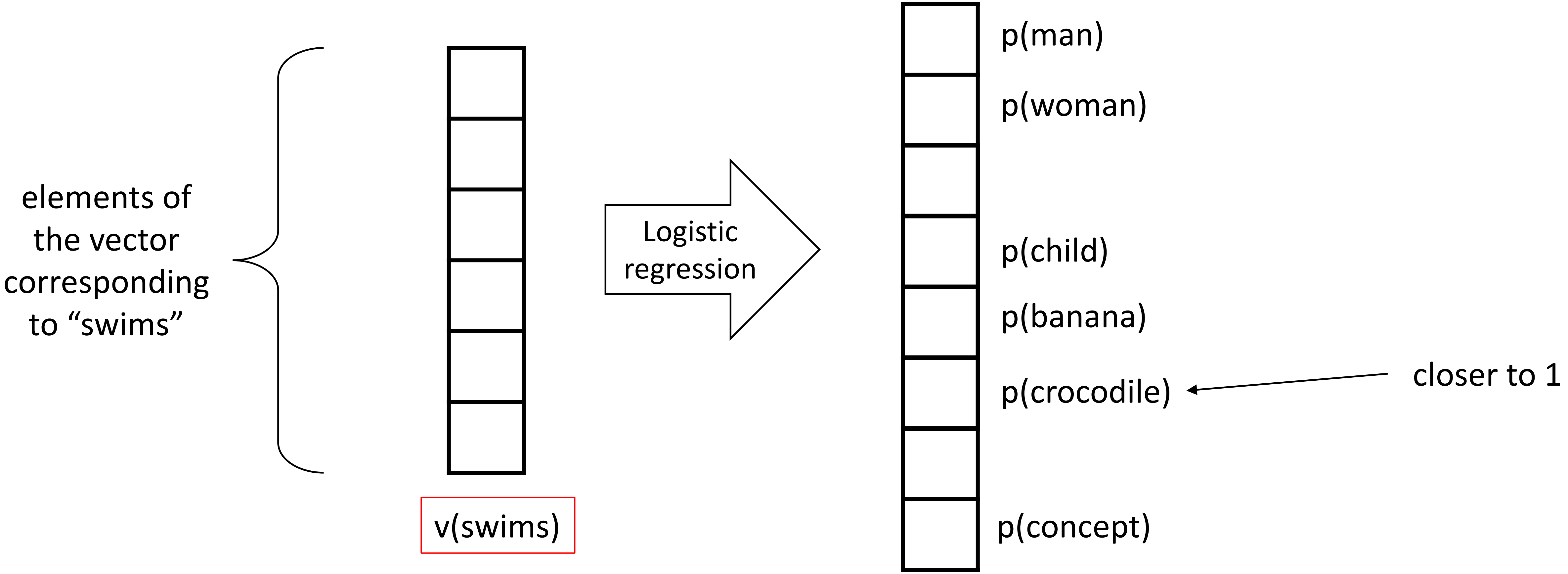
So how do we learn spatial locations for each word?

We want: a vector for each word that allows us to predict its context (i.e. what other words are likely/unlikely to be around it)



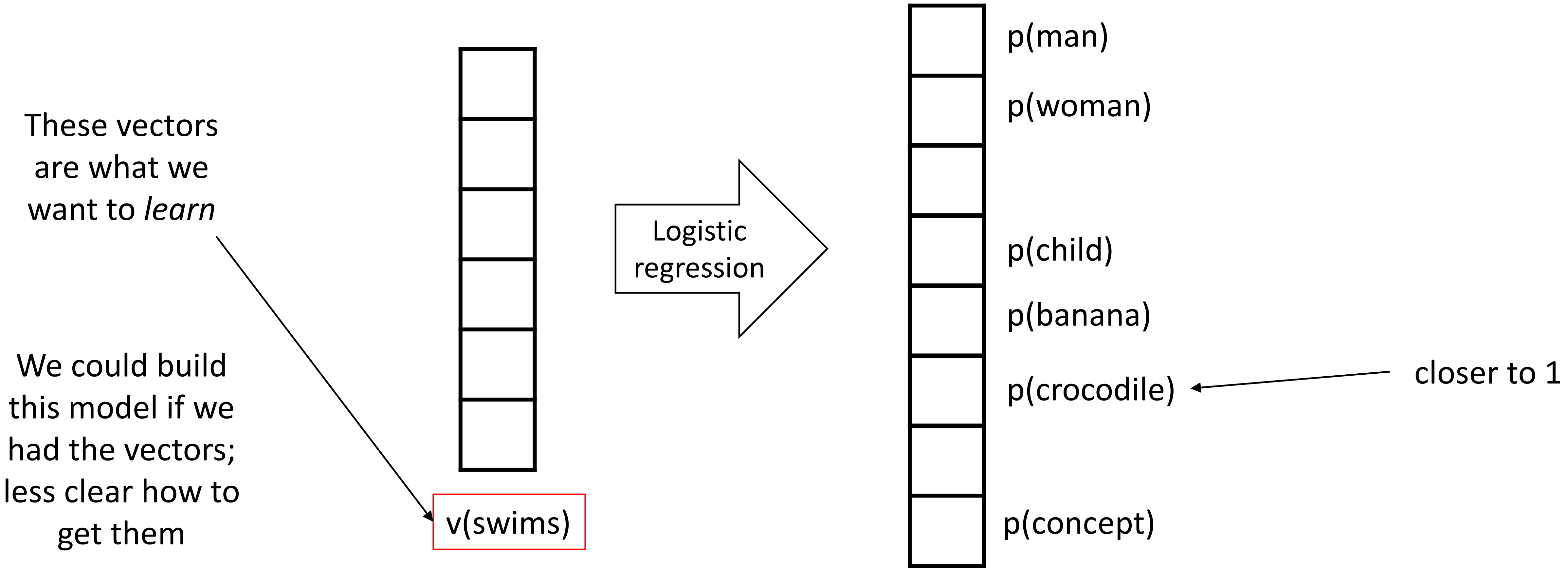
So how do we learn spatial locations for each word?

We want: a vector for each word that allows us to predict its context (i.e. what other words are likely/unlikely to be around it)



So how do we learn spatial locations for each word?

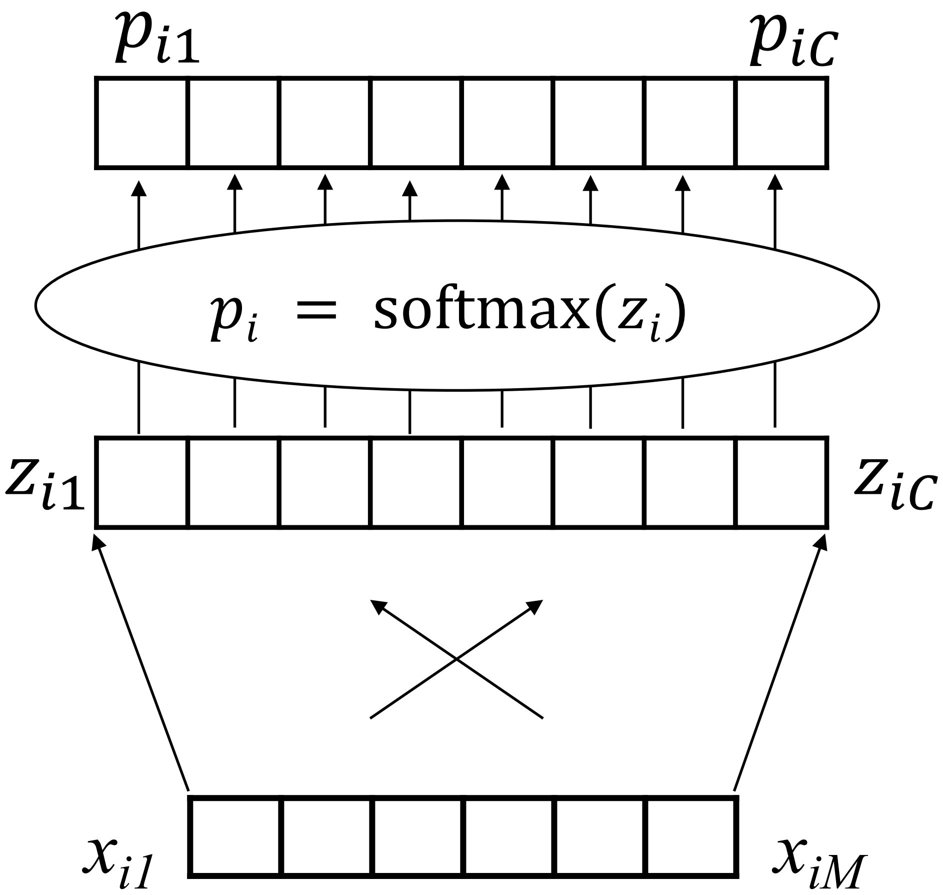
We want: a vector for each word that allows us to predict its context (i.e. what other words are likely/unlikely to be around it)



These vectors are what we want to *learn*

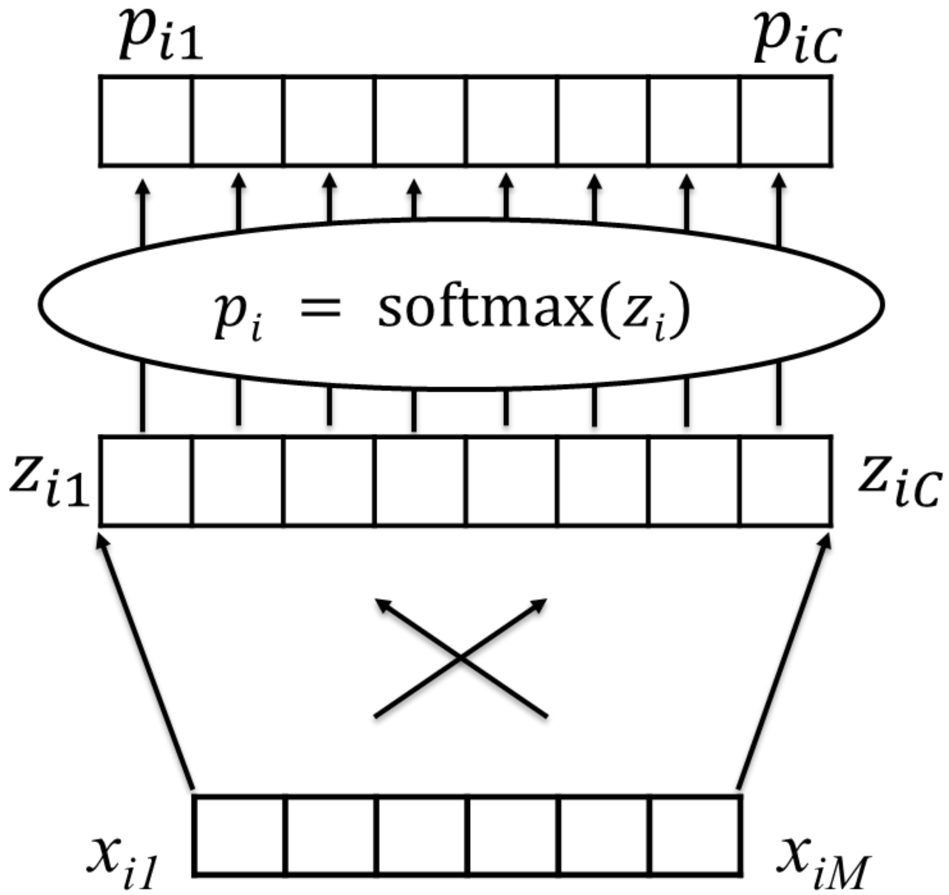
We could build this model if we had the vectors; less clear how to get them

Recall: Multi-Class Logistic Regression

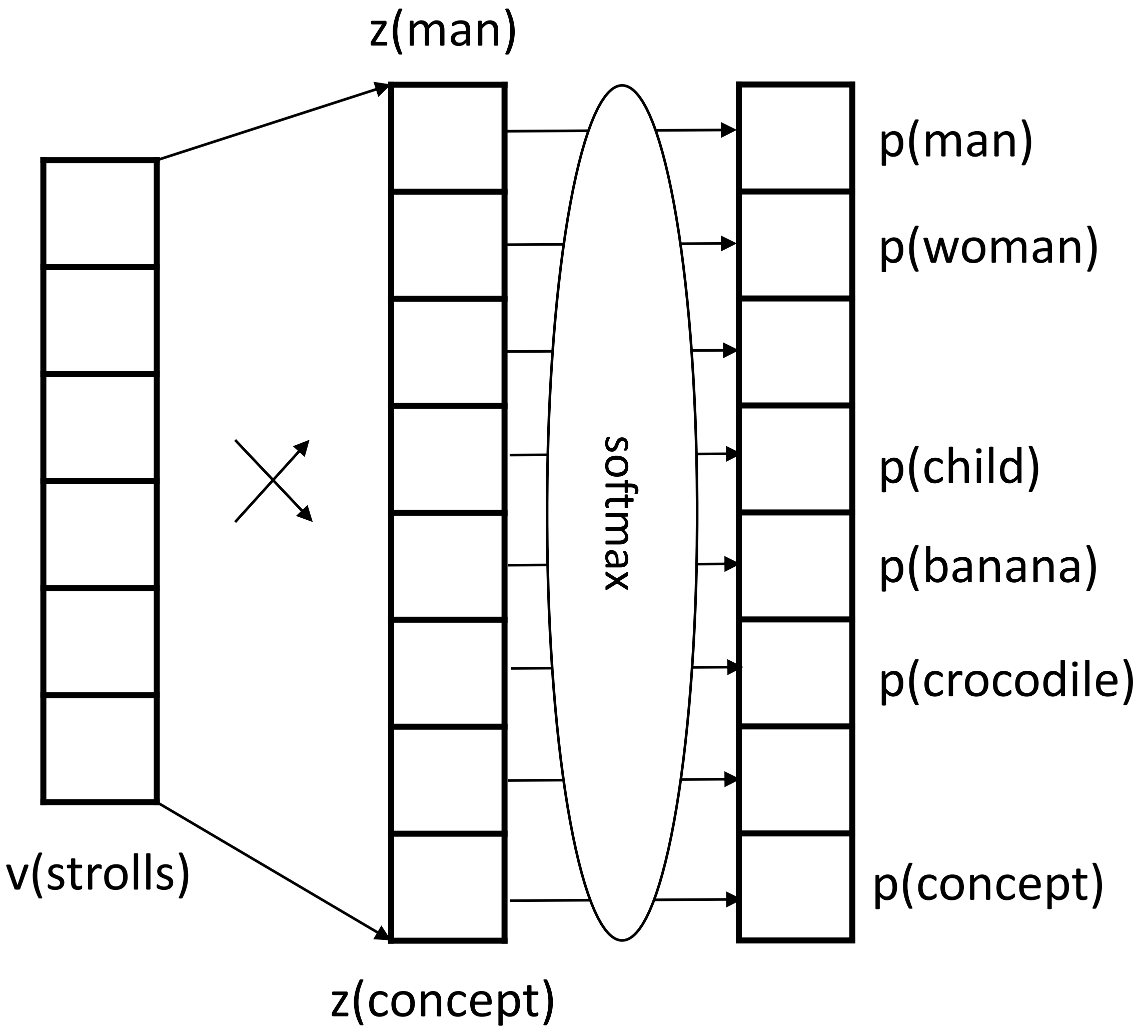


$$p_{ij} = \frac{e^{z_{ij}}}{\sum_{c=1}^{C} e^{z_{ic}}}$$

Recall: Multi-Class Logistic Regression



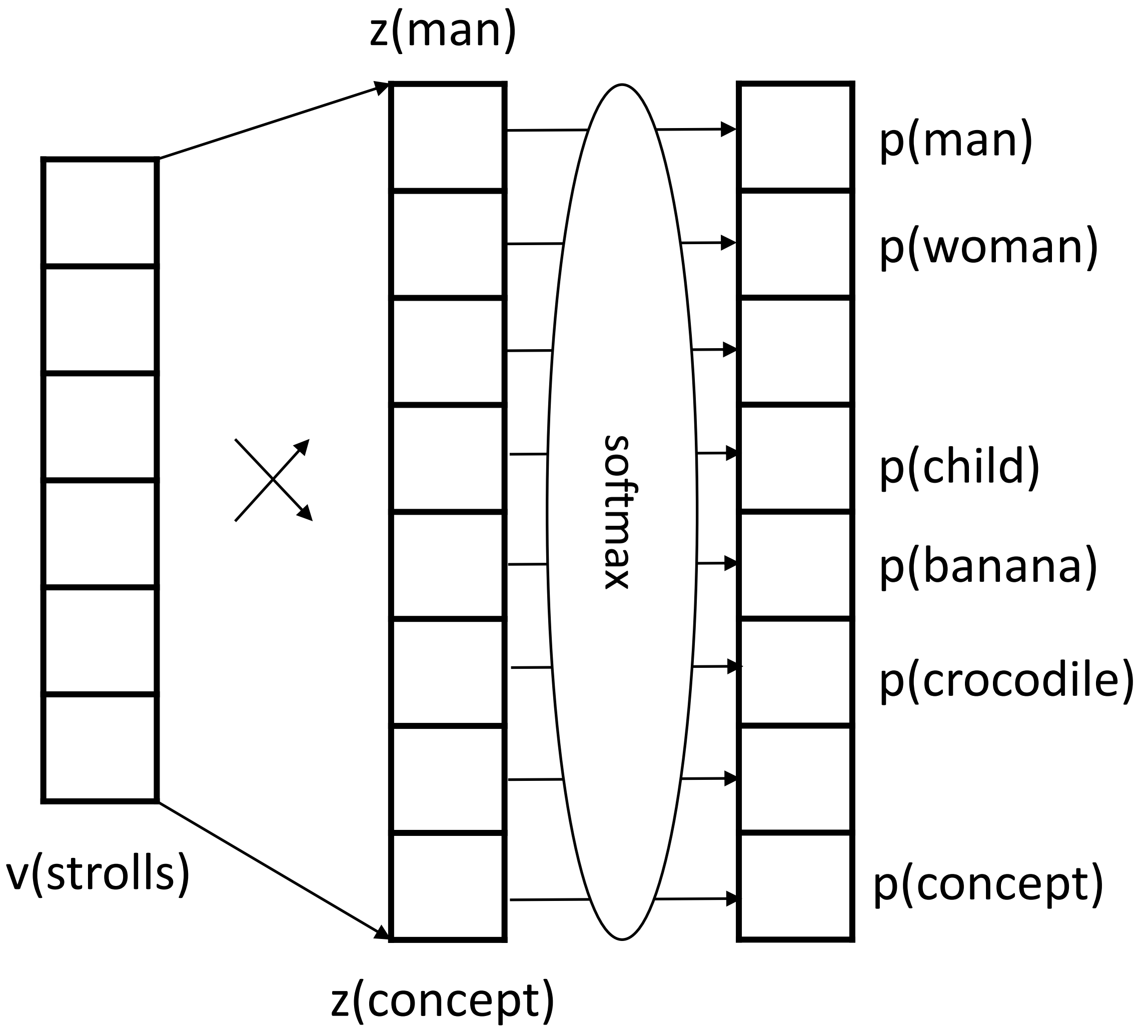
Recall: Multi-Class Logistic Regression



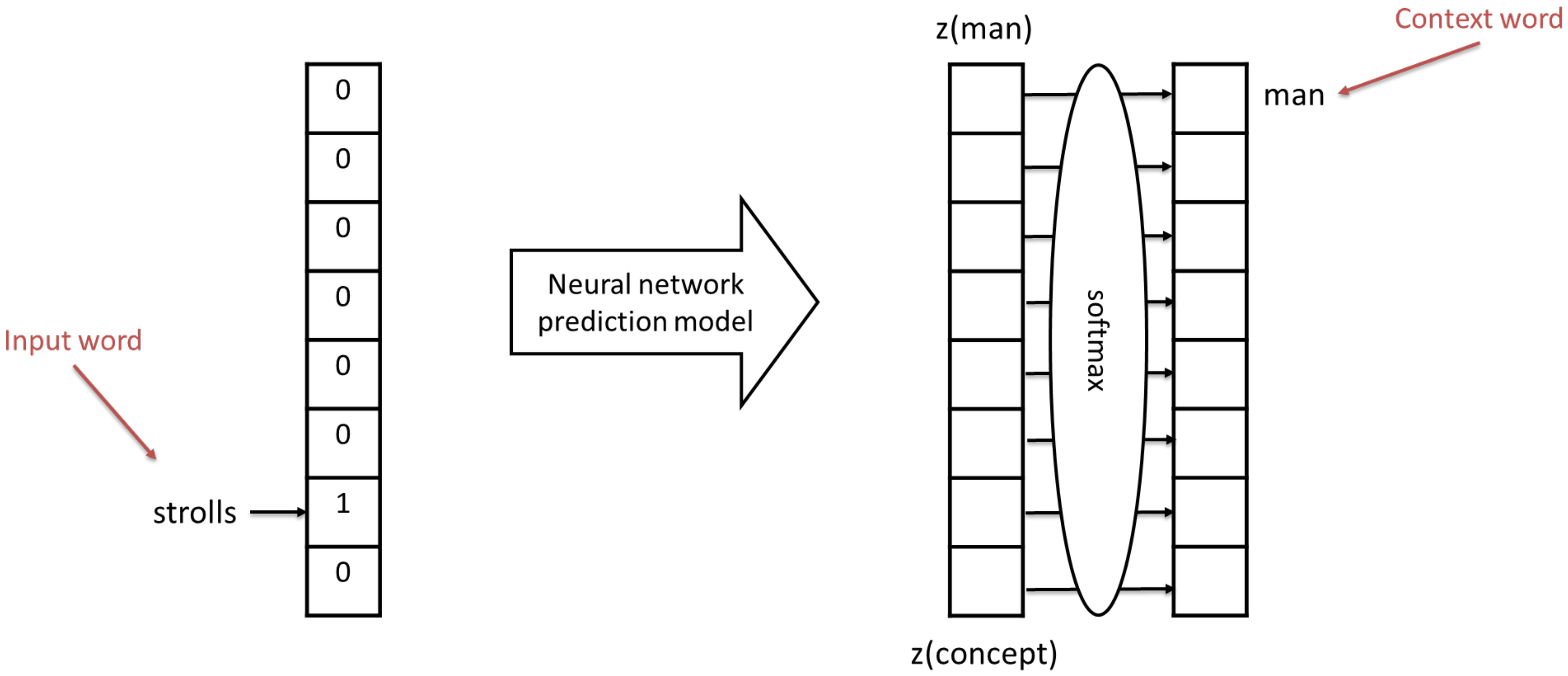
We want: word vectors that allow us to predict their likely context

But again, how do we *learn* these vectors?

Let’s take a step back: we’ll focus on understanding how we can predict context words based on input words



Predicting context words based on input words



Input words and context words are one-hot encoded
(similar to bag of words representation)

Predicting context words based on input words

Training Data:

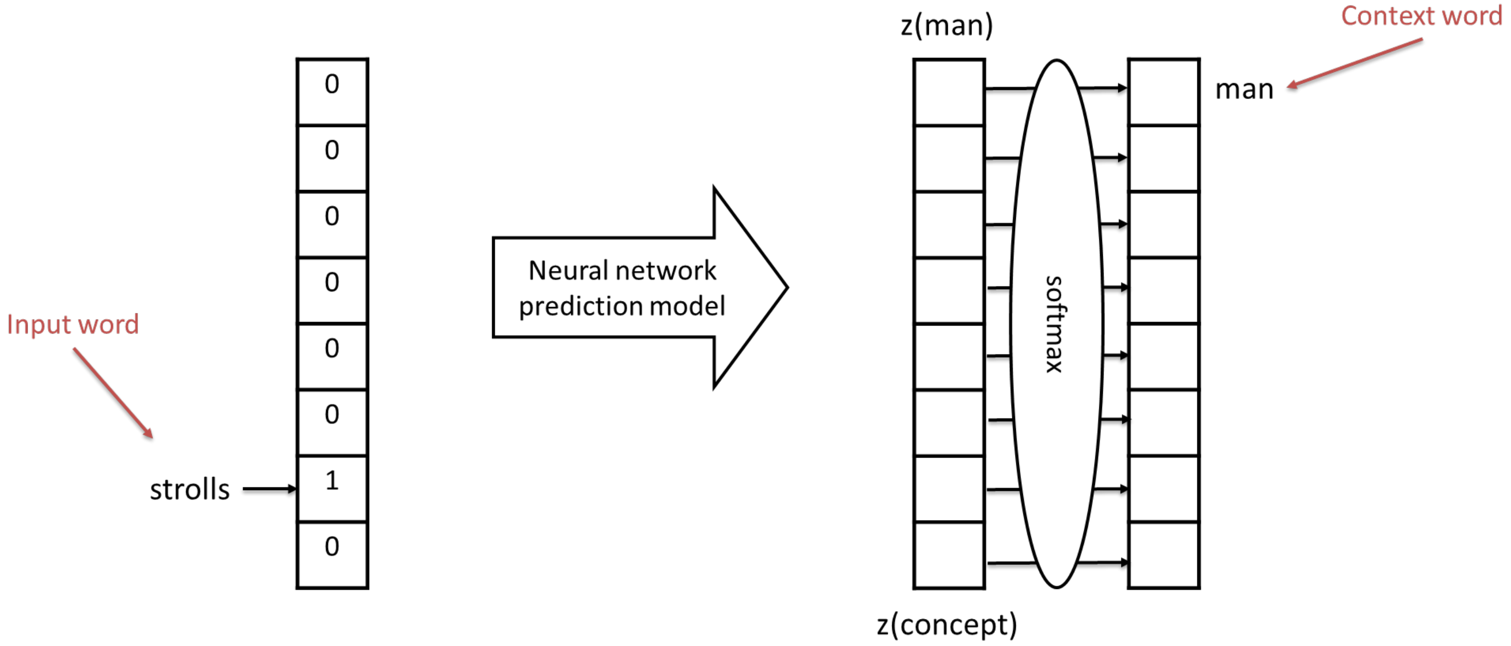
HUGE number of pairs of the following form:

{input word, context word}

e.g. from Wikipedia

Examples:

{strolls, man}
{strolls, woman}
{swims, crocodile}
{swims, fish}
{flies, bird}
{flies, plane}



Predicting context words based on input words

Training Data:

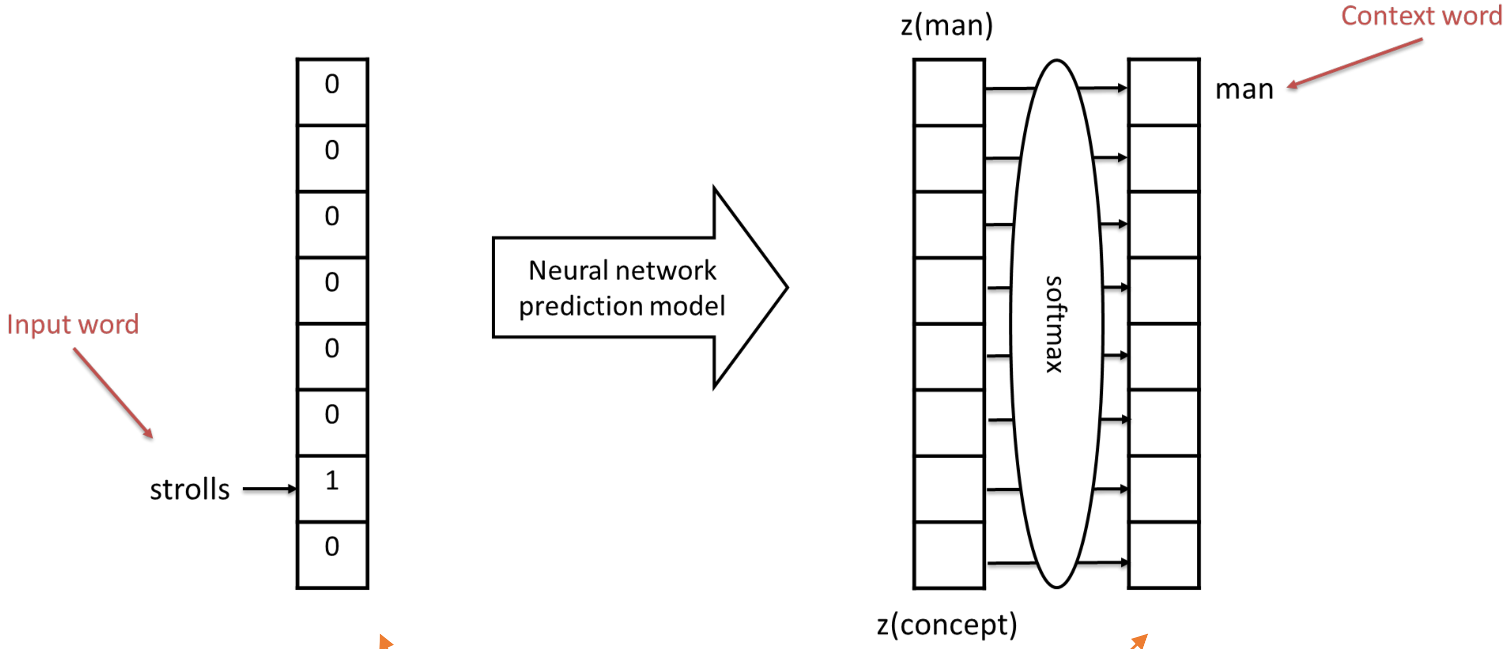
HUGE number of pairs of the following form:

{input word, context word}

e.g. from Wikipedia

Examples:

{strolls, man}
{strolls, woman}
{swims, crocodile}
{swims, fish}
{flies, bird}
{flies, plane}

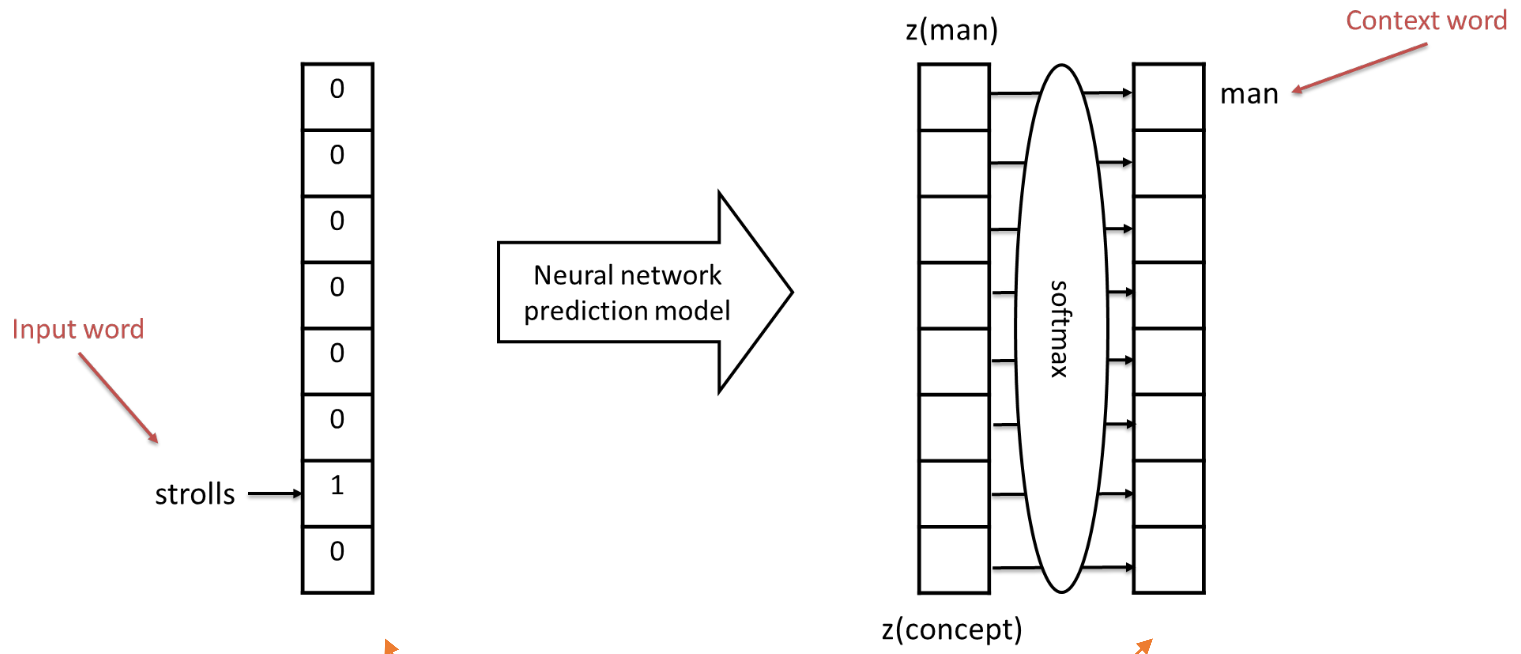


These vectors are huge!
They’re the size of our vocabulary

What’s the simplest model we can possibly use?

First idea:

Directly connect our input to the log-odds layer



These vectors are huge!
They’re the size of our vocabulary

What’s the simplest model we can possibly use?

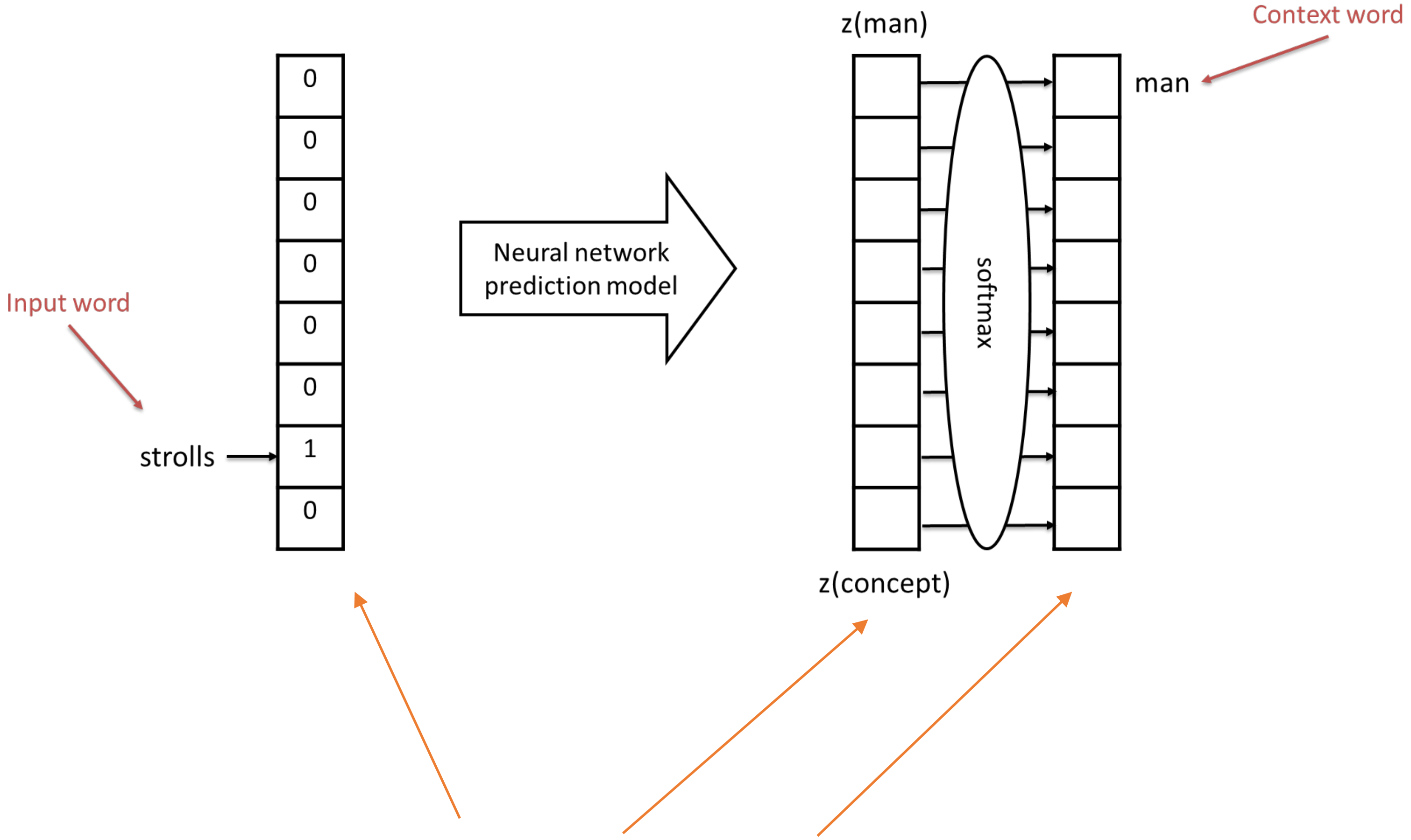
First idea:

~~Directly connect our input to the log-odds layer~~

~~How many connections?~~

~~V x V~~

~~Where V is our vocabulary size (approx. 6 billion)~~



These vectors are huge!
They’re the size of our vocabulary

What’s the next simplest?

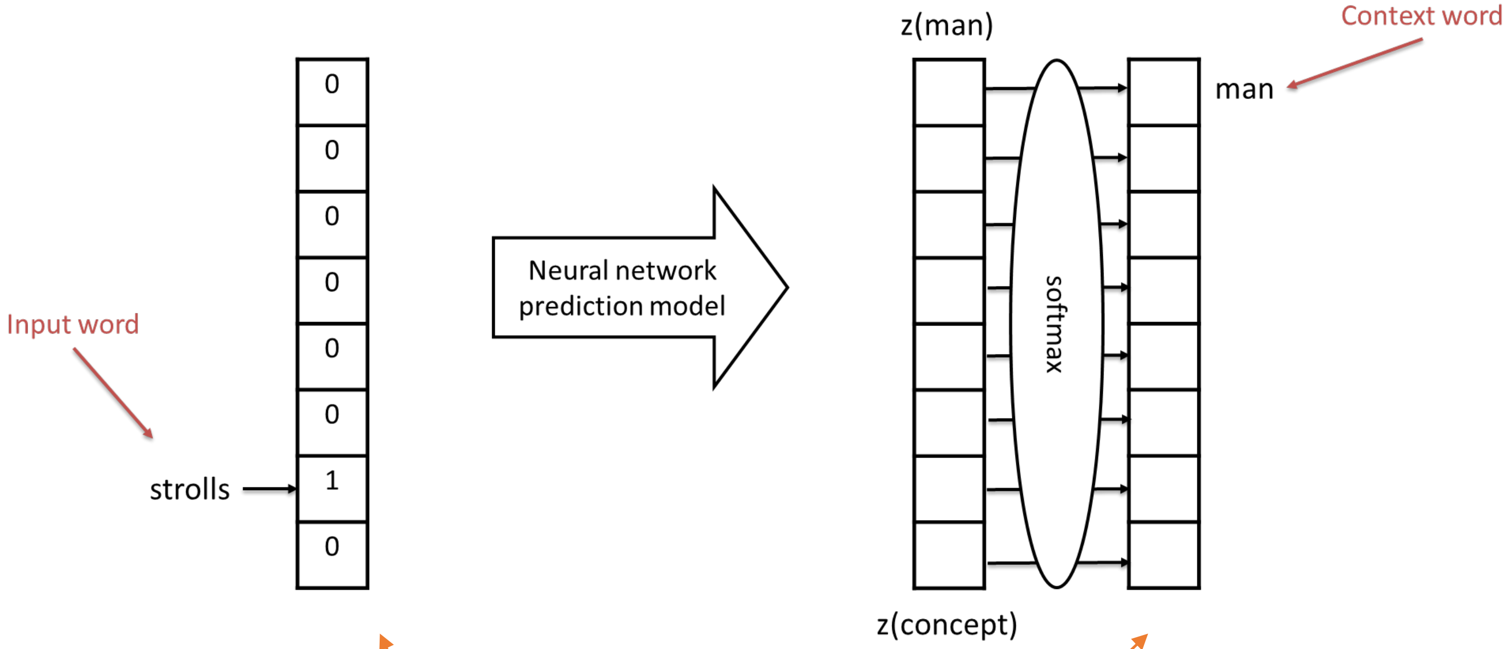
How about a single hidden layer?

How many connections?

V x H x 2

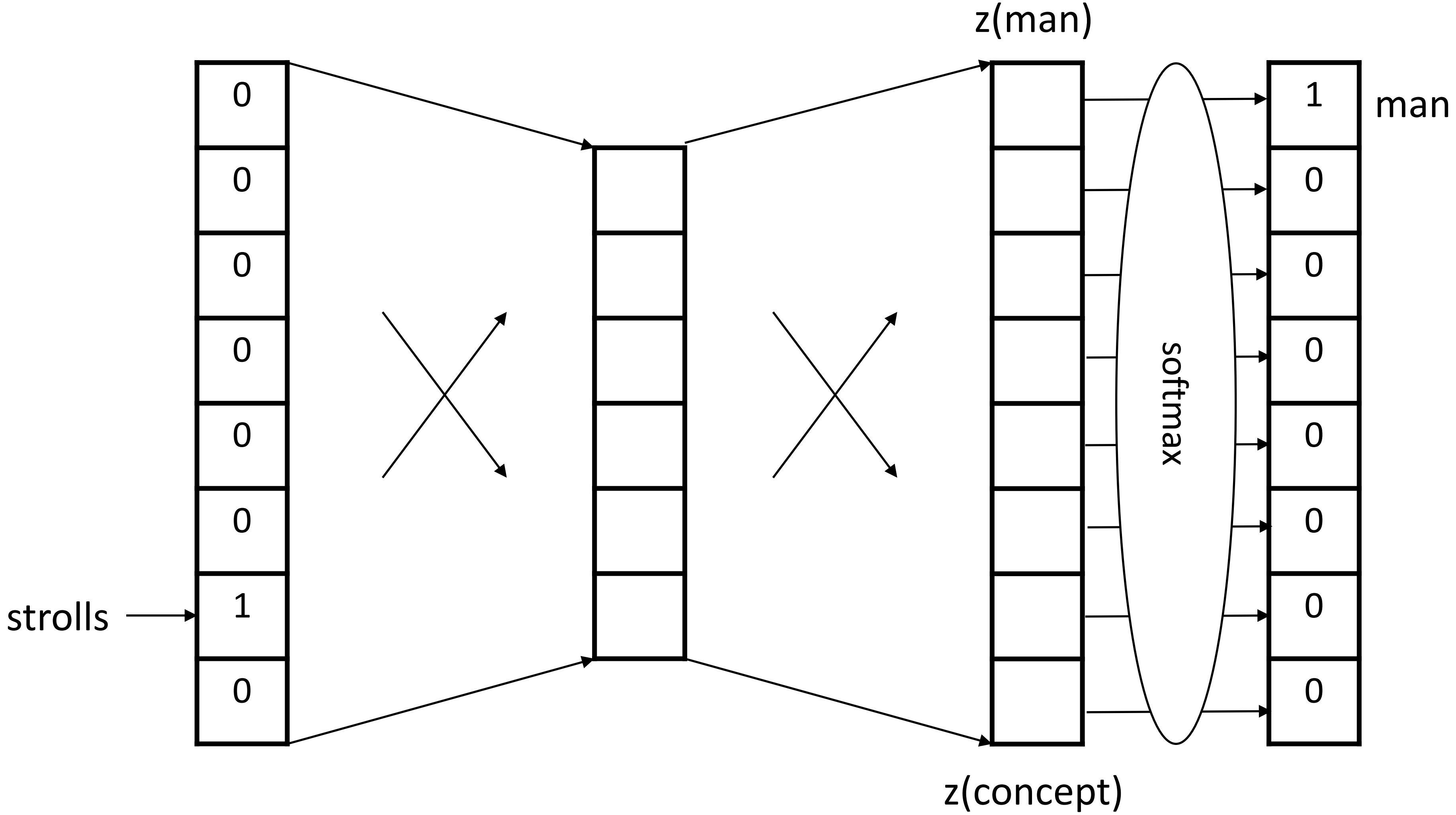
Where V is our vocabulary size (approx. 6 billion)

And H is our hidden layer size ($\ll V$)

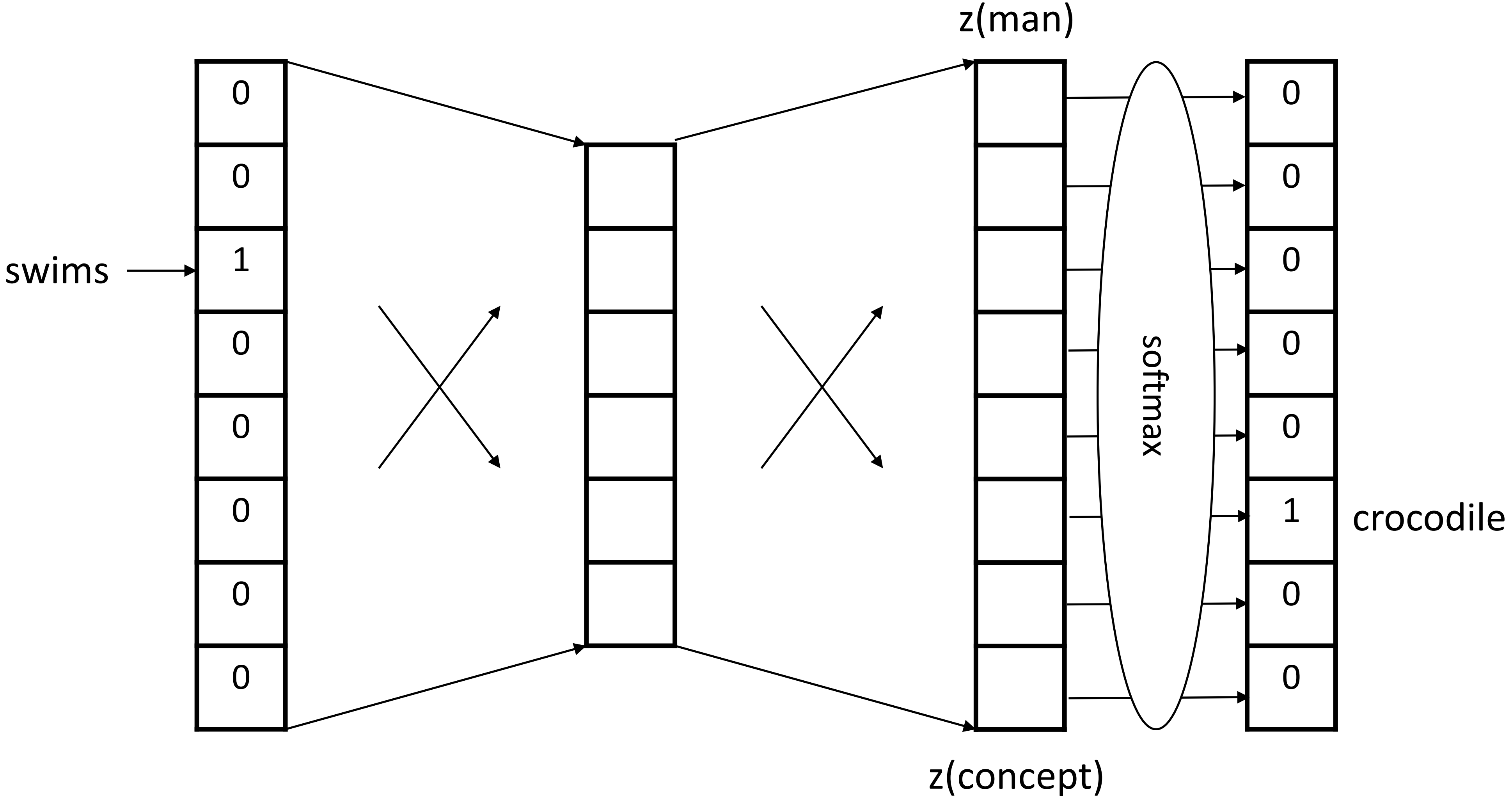


These vectors are huge!
They’re the size of our vocabulary

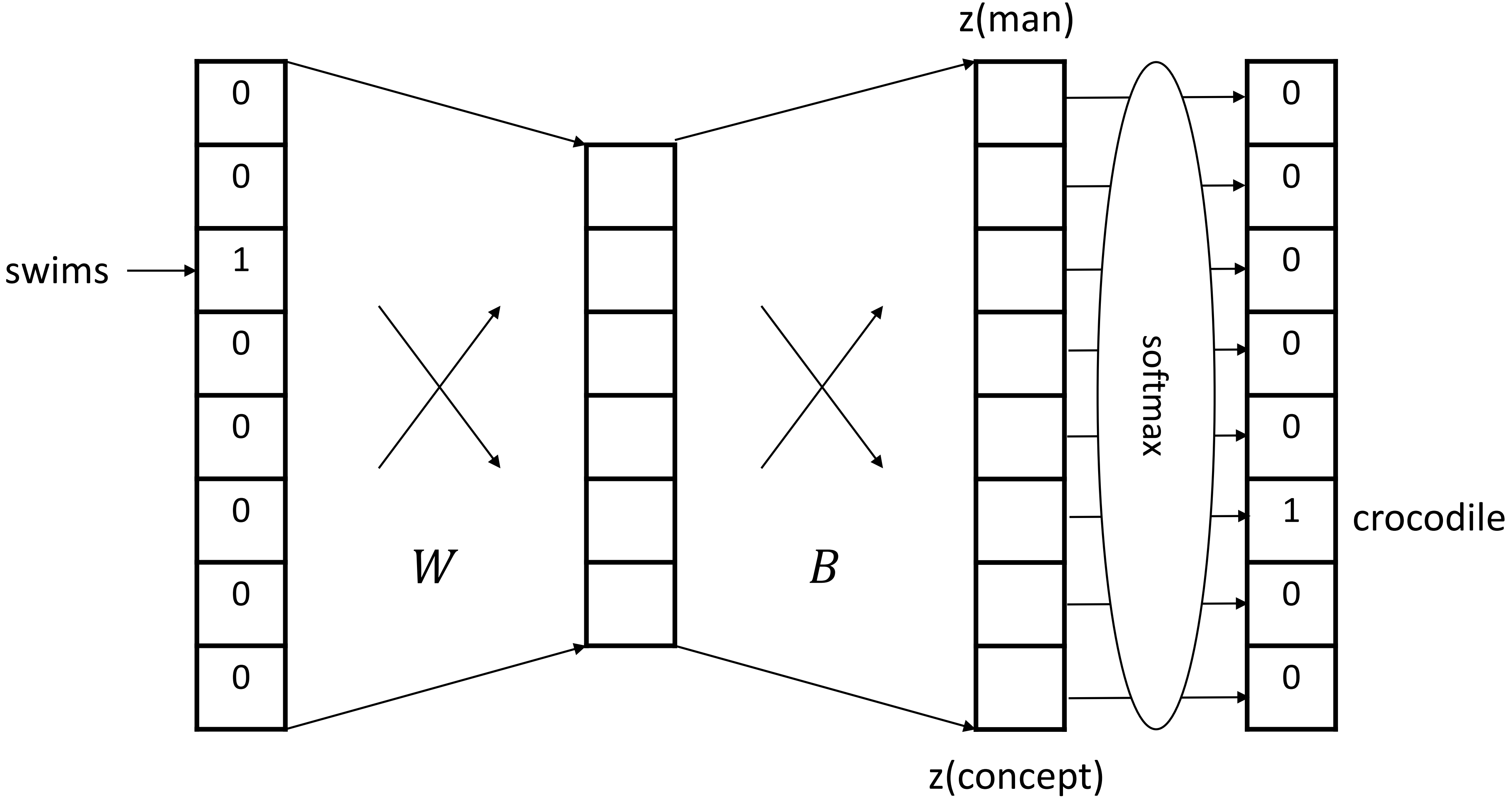
OK, let’s try it: use a single hidden layer



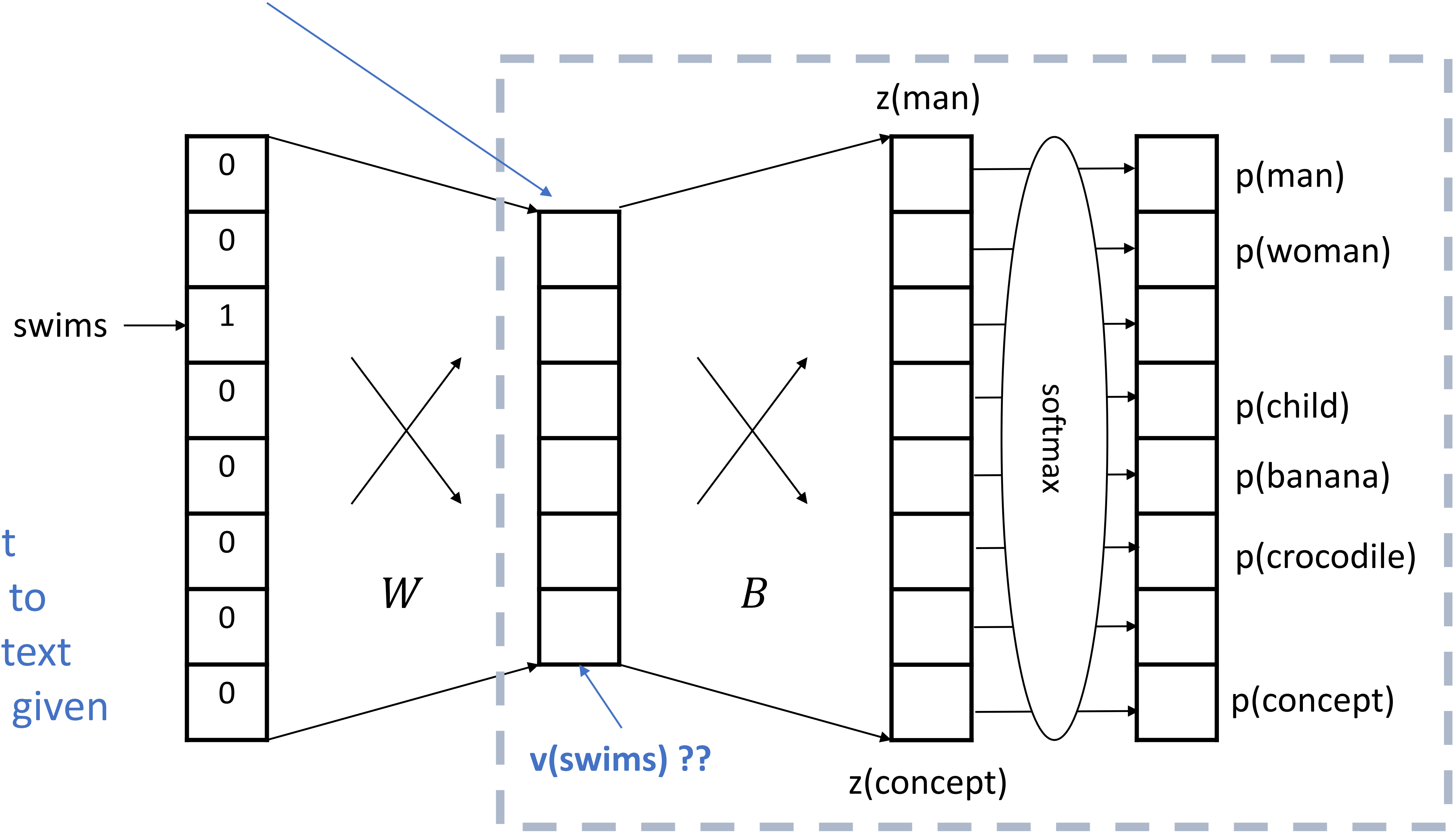
Use mini-batches of training examples; minimize cross-entropy loss



Learn our parameters: Weight Matrices W and B

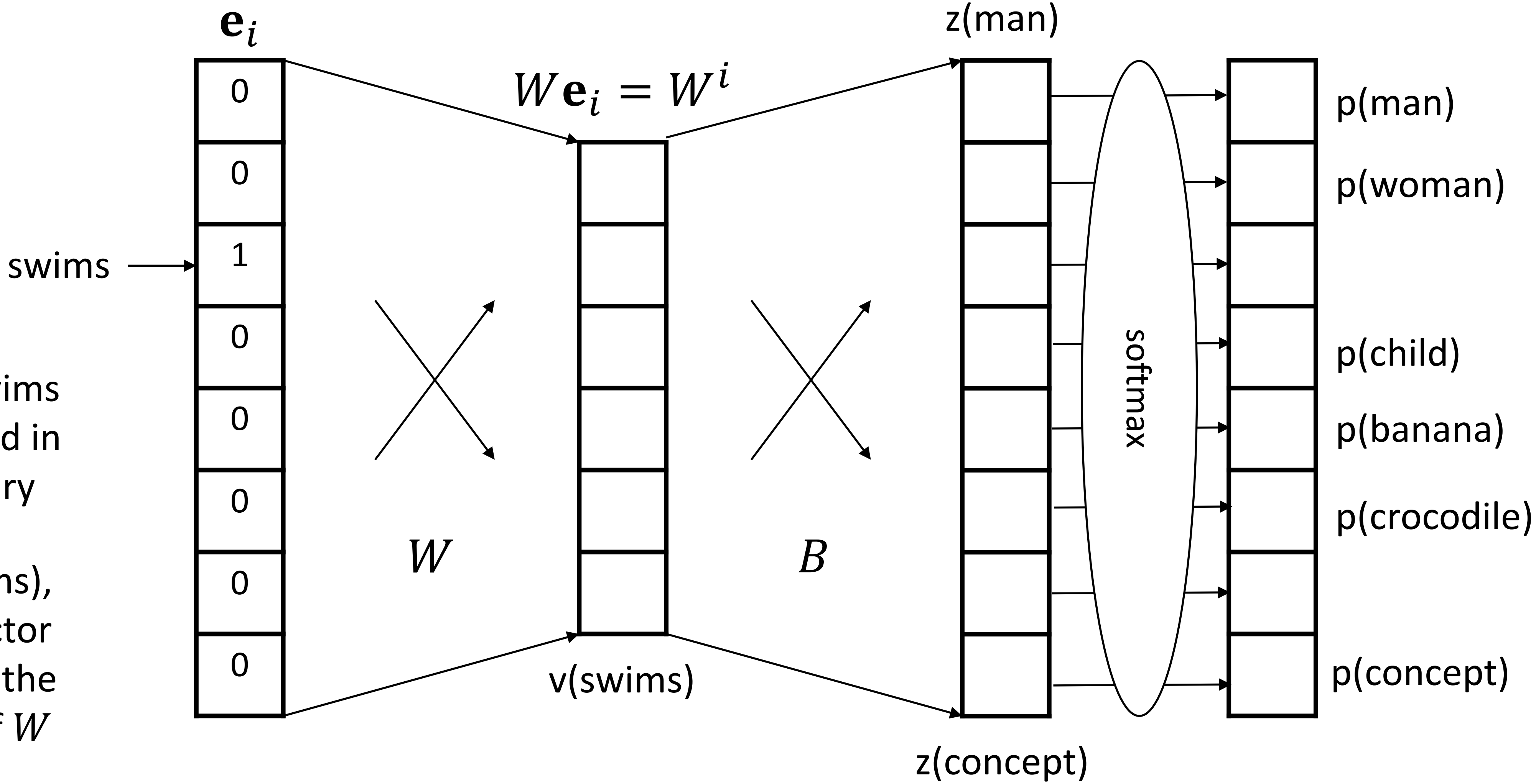


Isn't **this** the vector we were looking for?



- it's compact
- It allows us to predict context words for a given input word

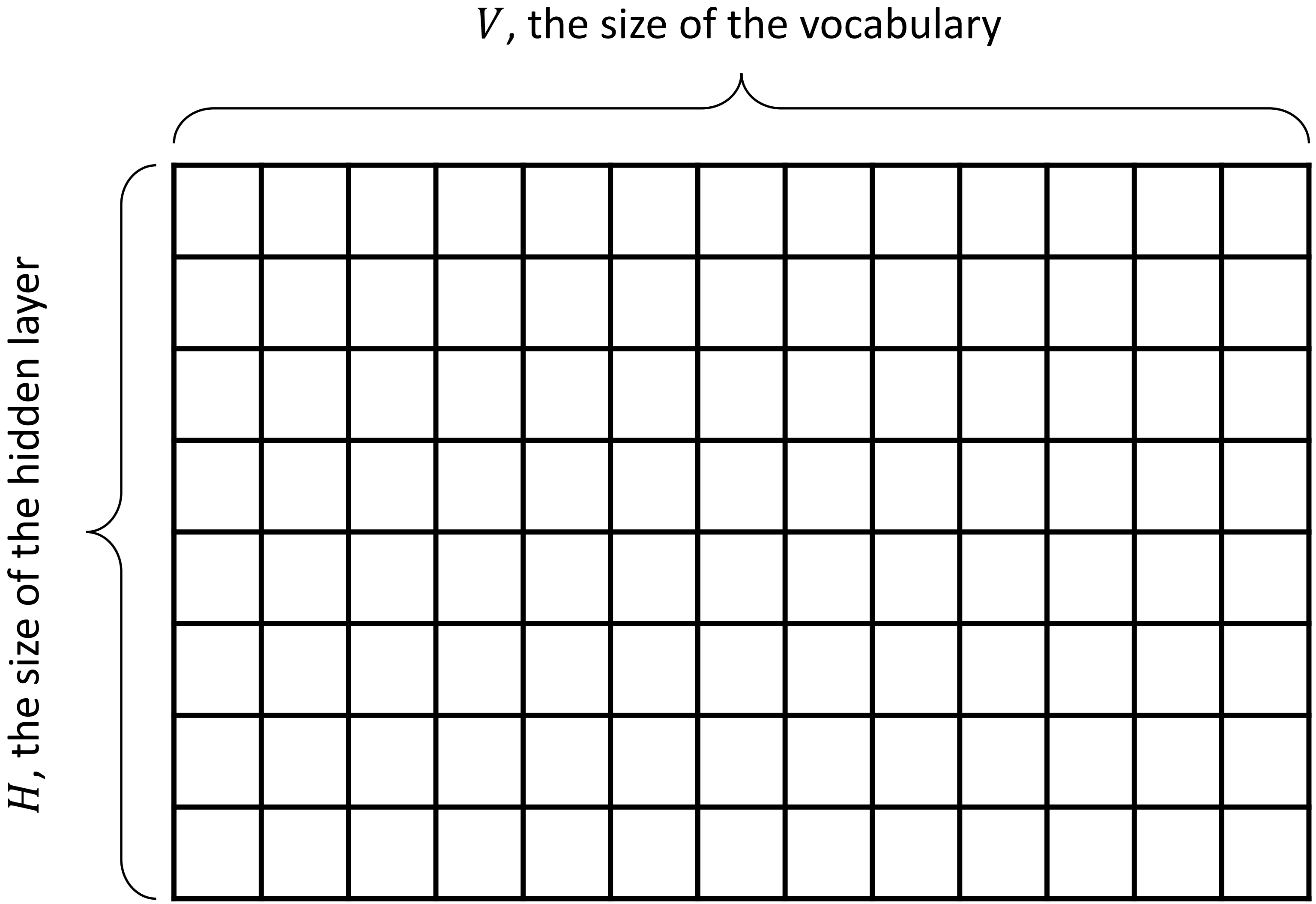
Let's take a closer look at W



- suppose swims is the i^{th} word in our vocabulary

- then v(swims), the word vector for swims, is the i^{th} column of W

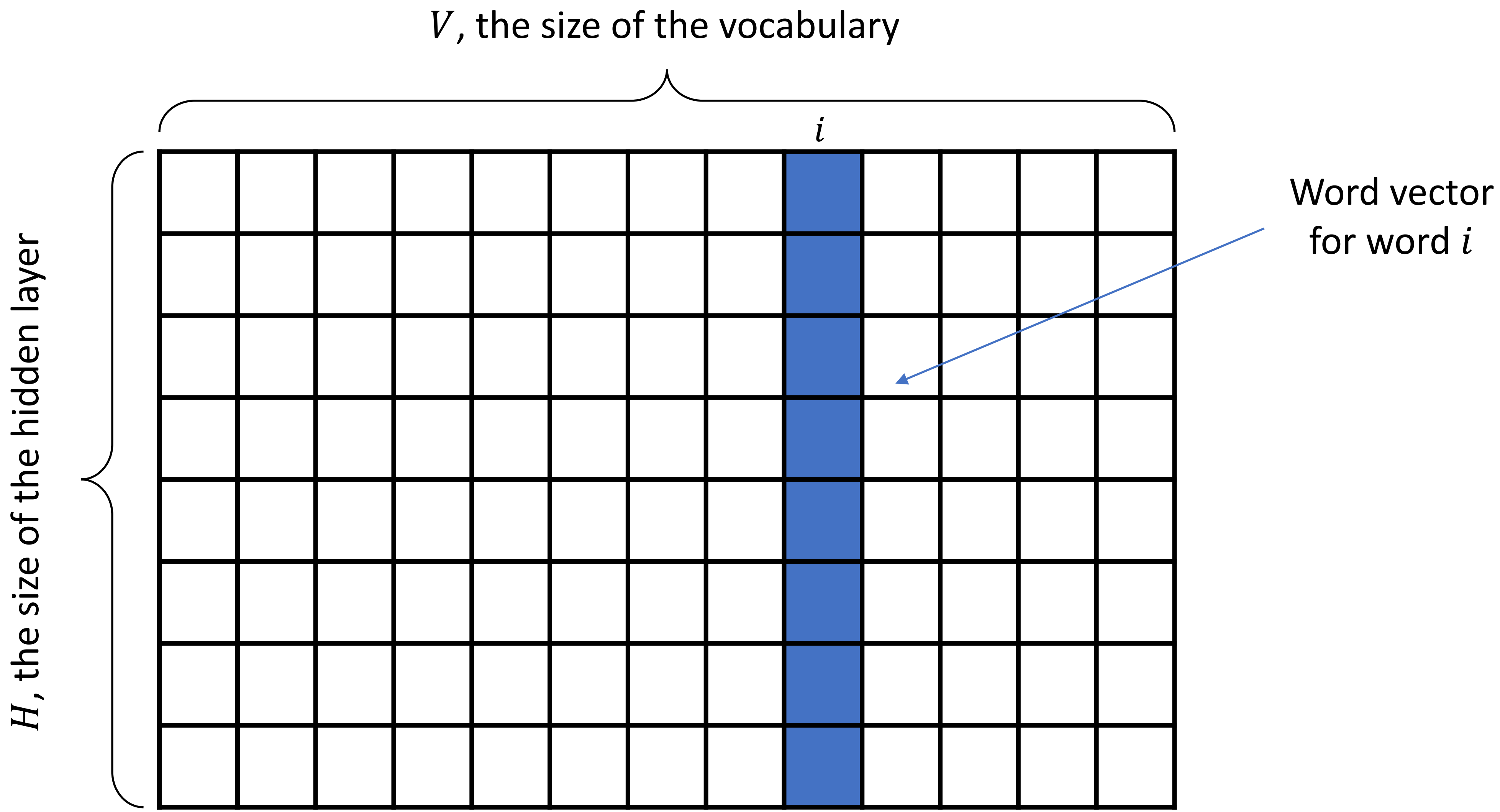
Let’s take a closer look at W



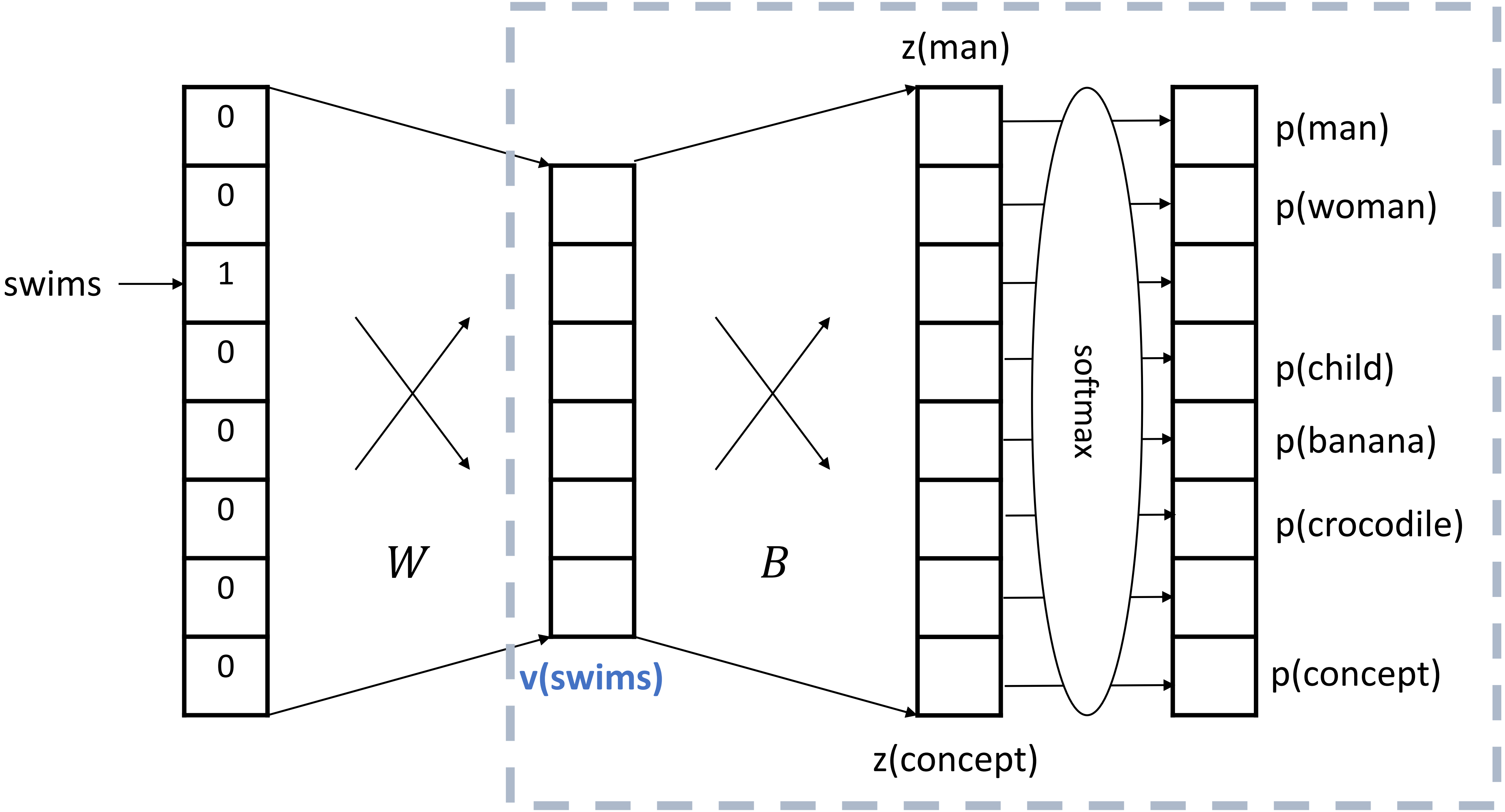
V, the size of the vocabulary

H, the size of the hidden layer

Let’s take a closer look at W


V, the size of the vocabulary
i
H, the size of the hidden layer
Word vector for word i

We now have a distributed representation of word *meaning* based on *context*



Important Takeaways:

- We are learning a vector representation for each word based on the contexts in which it appears
- training data: large number of pairs of nearby words from a large corpus
- These vectors give us much more flexibility when modeling: makes text sequences like other sequences