CONSTELLATE

How does ChatGPT work?

CONSTELLATE

A review of the structure of the webinar series

ChatGPT

GPT: generative pre-trained $\boxed{\text{transformer}}$

Transformer

a multi-layer neural network that relies on the parallel multi-head attention mechanism.

ChatGPT

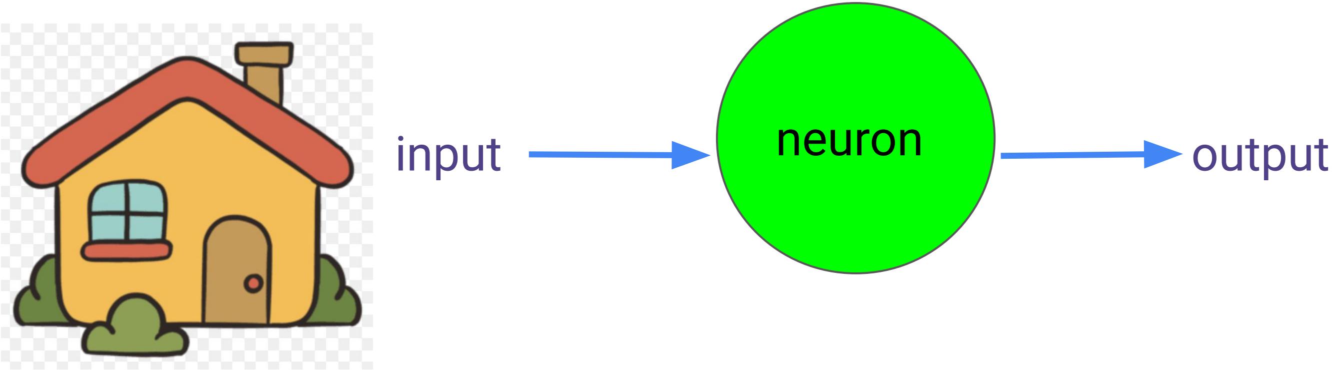
Part 1: multi-layer neural network

Part 2: multi-head attention mechanism

Last time

Part 1: multi-layer neural network

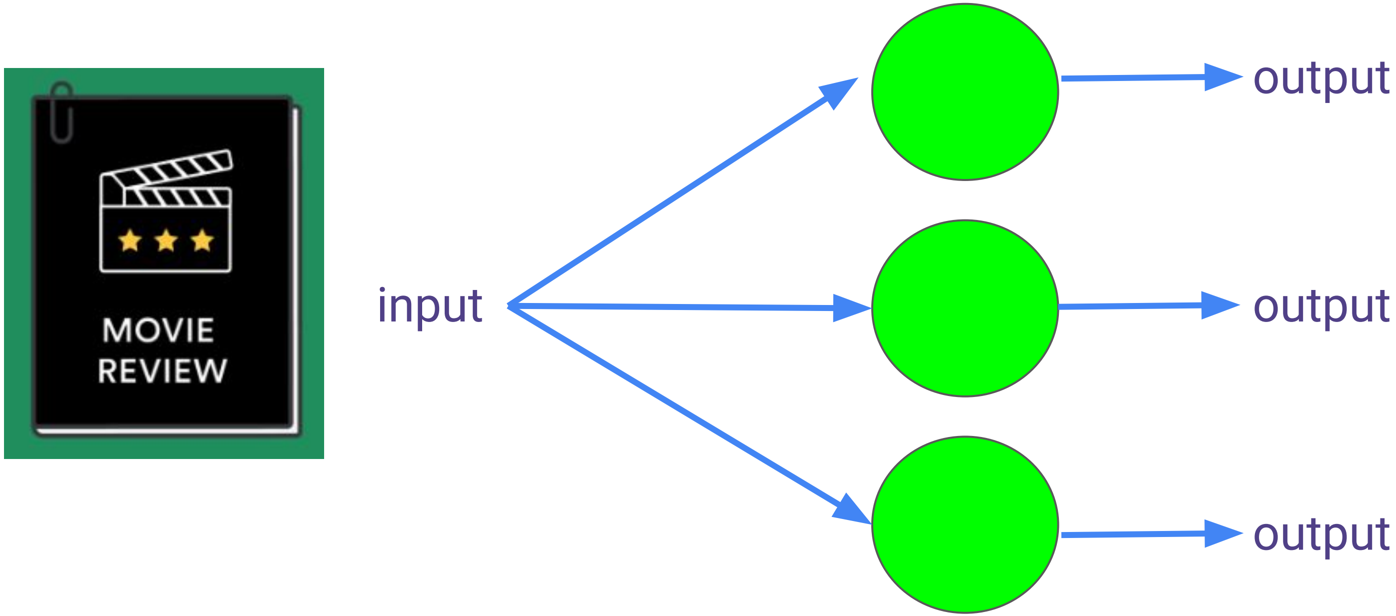
Last time



One neuron

One layer

Last time



many neurons

one layer

Today

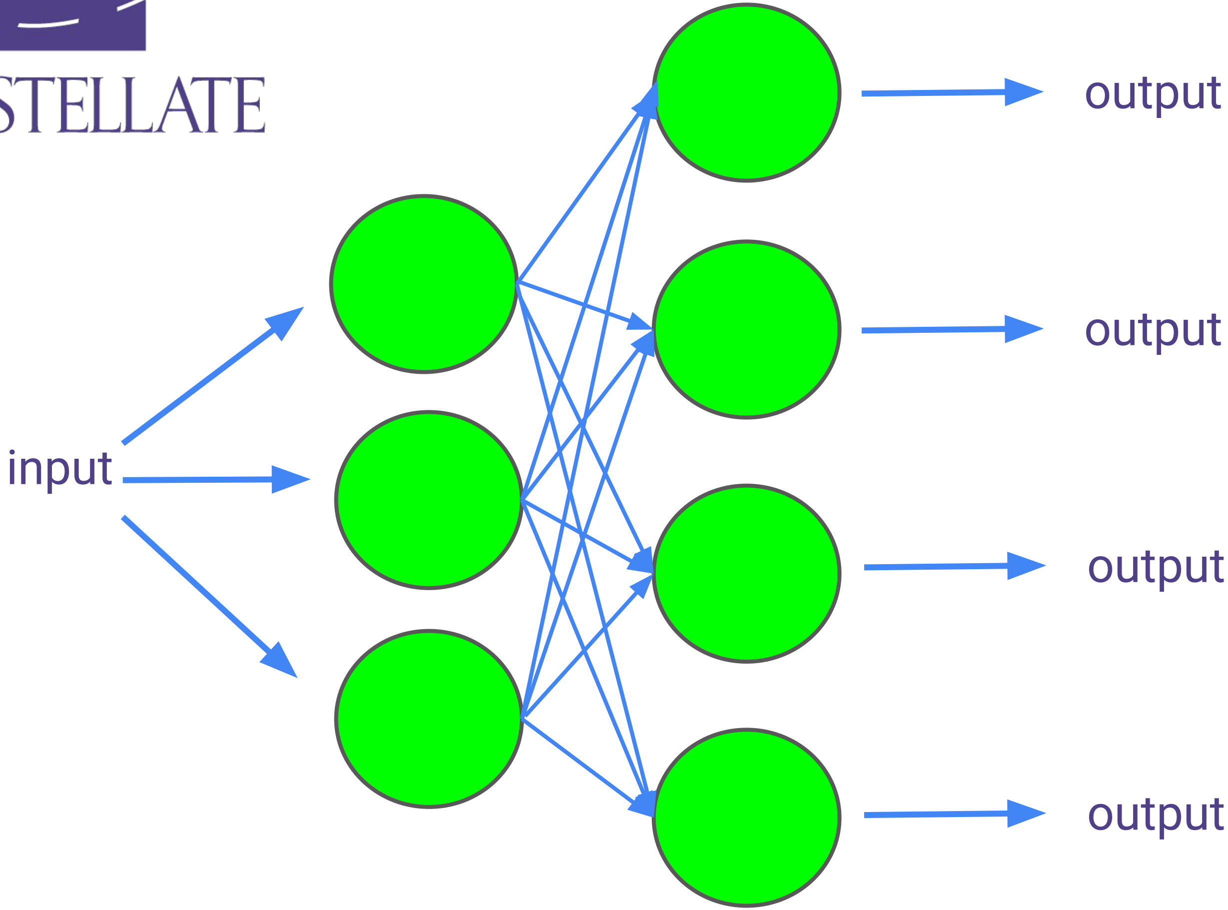
Part 1: multi-layer neural network

CONTINUED

CONSTELLATE

Many neurons, many layers

CONSTELLATE



many neurons

many layers

The task of ChatGPT

Given a sequence of words, what is the most likely word that appears next?

The quick brown fox jumps over the lazy dog.

- In the house classification example, we use a feature vector to represent a house and use it to predict the house being good or bad

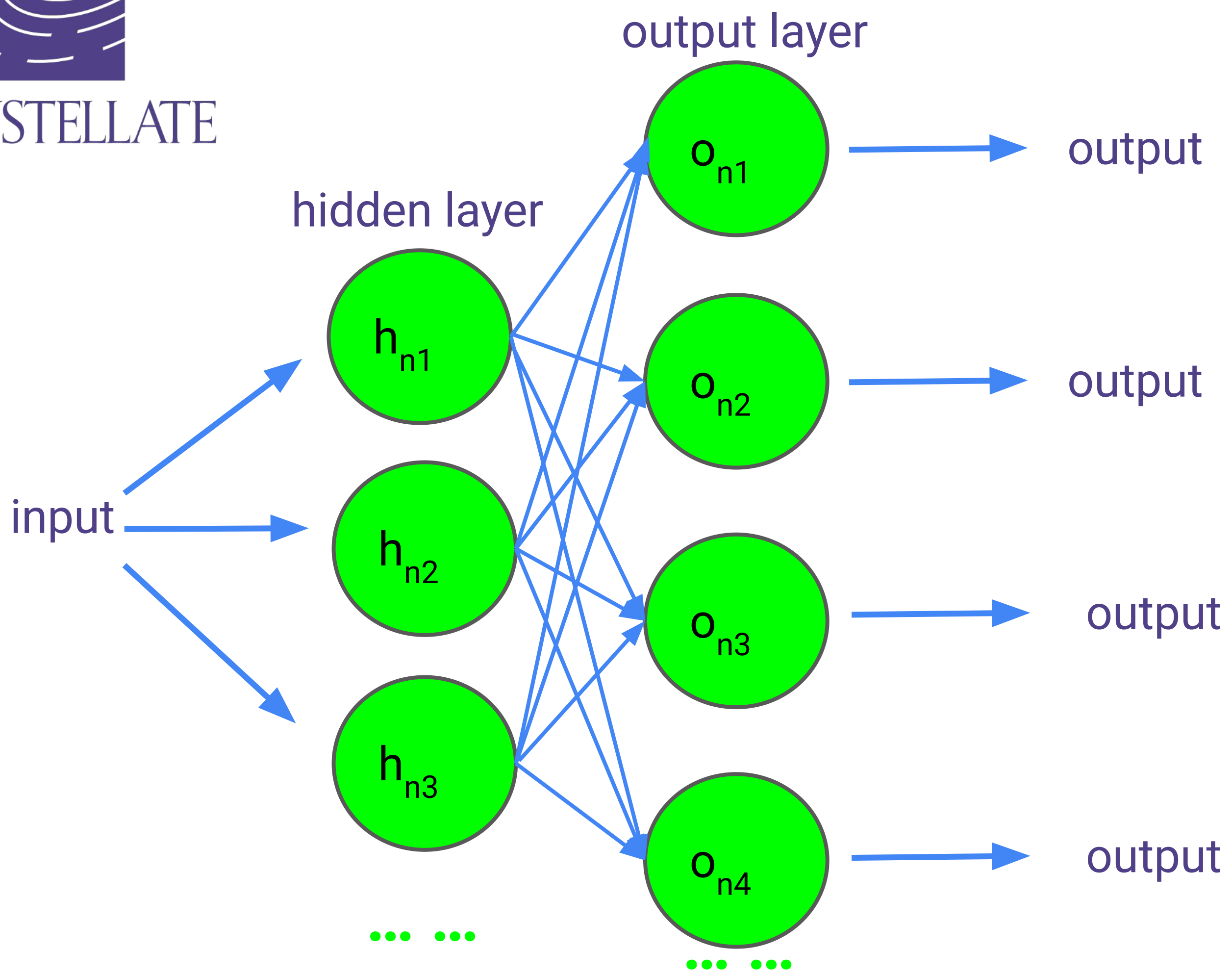
- By analogy, we will need to use feature vectors of words to predict the next possible word in a sequence.

How do we derive the feature vectors of words?

CONSTELLATE

We'll introduce a model that derives feature vectors of words called skip-gram, a neural network that looks like …

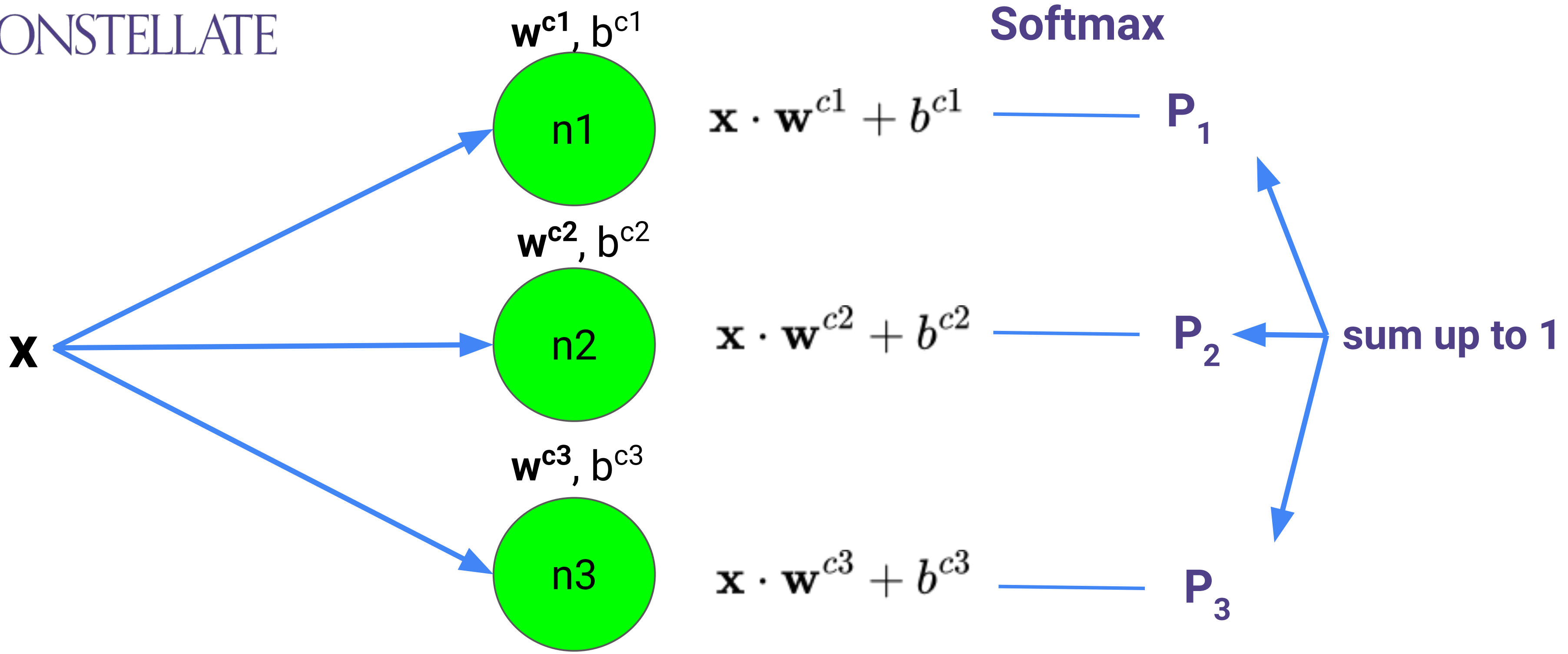
Skip gram



many neurons

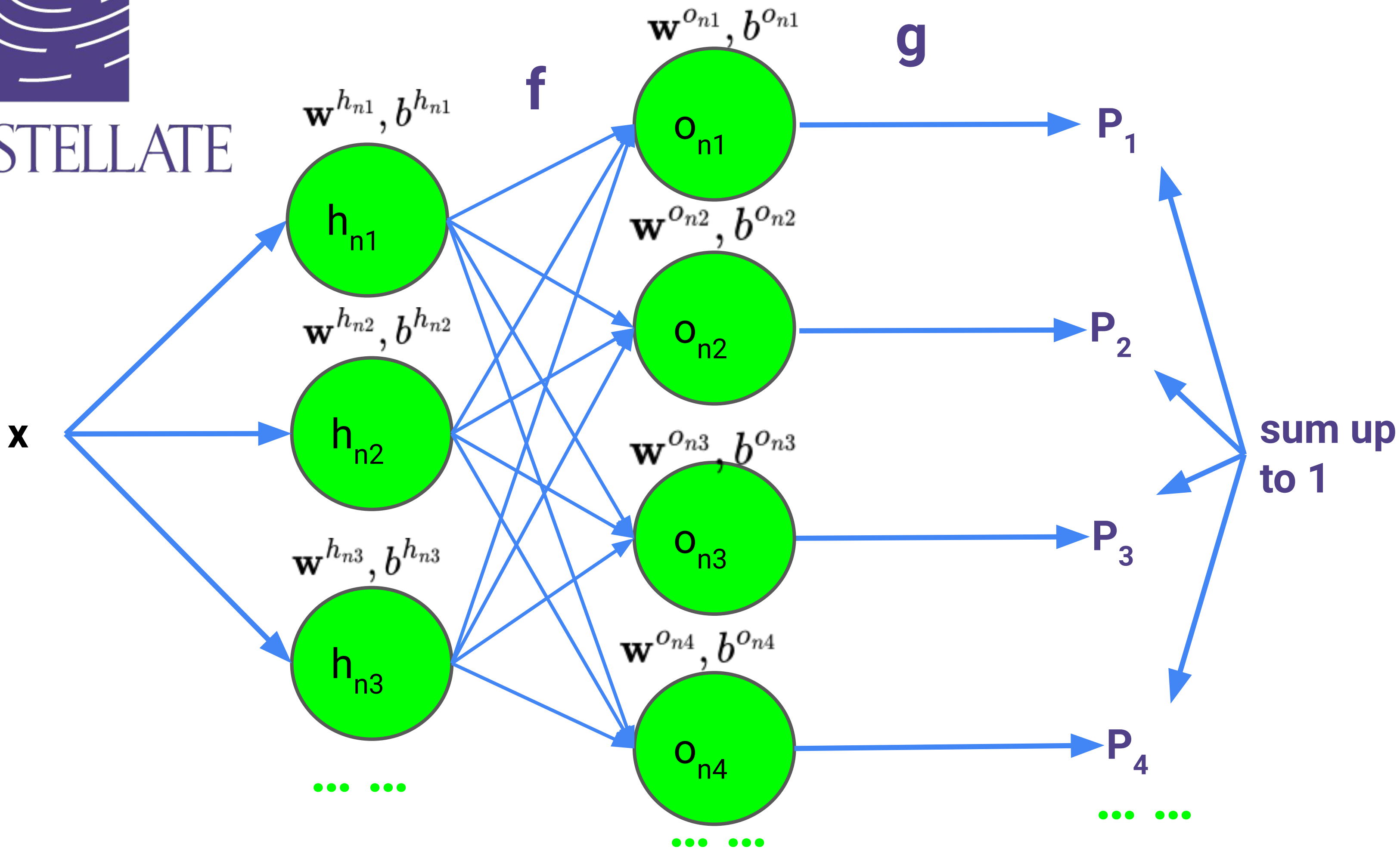
many layers

Recall that...



Each neuron has its own weight vector and bias term

Generalizing

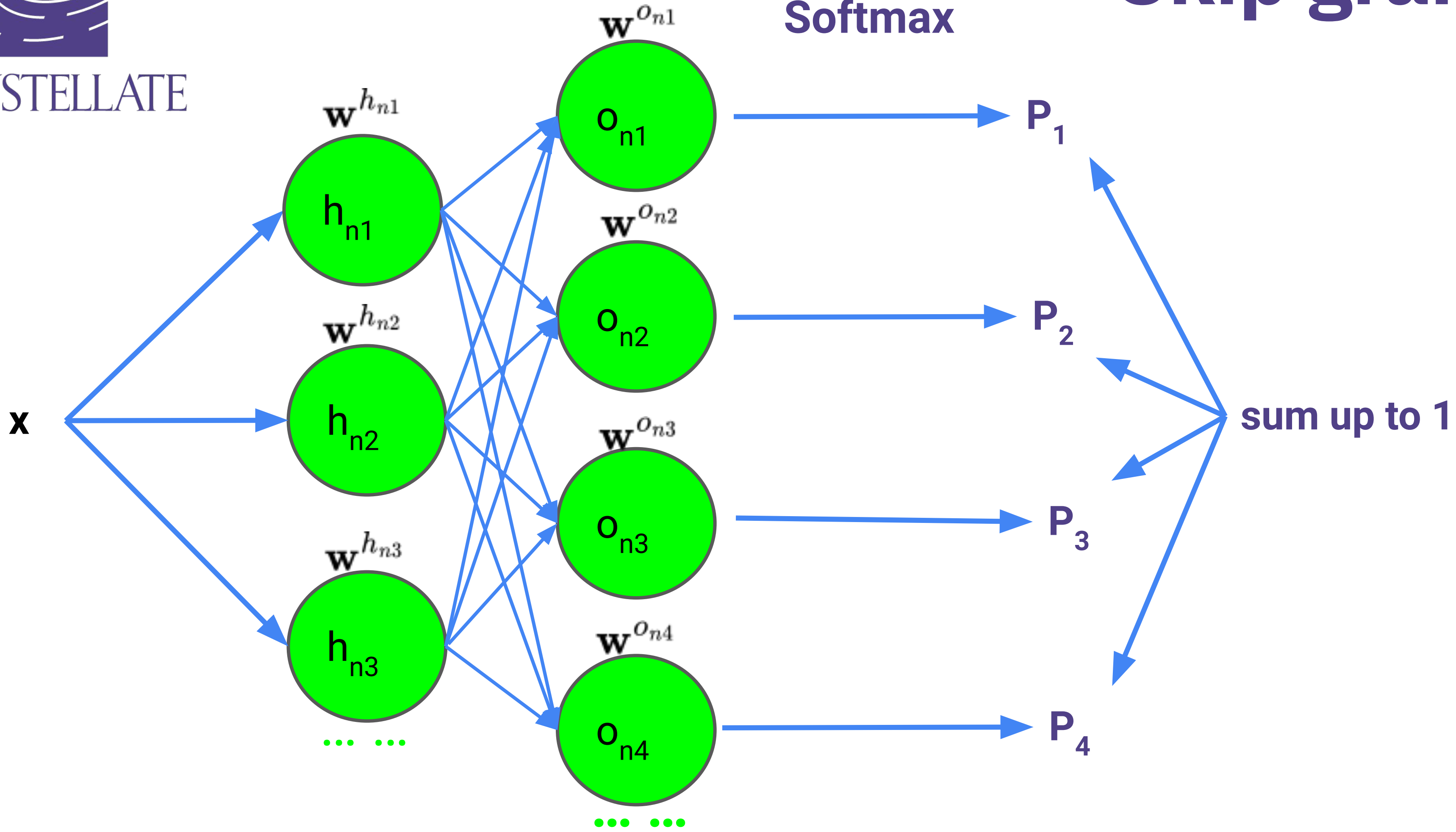


Skip-gram is much simpler!

Skip gram

- No bias term in either the hidden layer or the output layer
- No activation function in the hidden layer

Skip gram



Softmax
x
$\mathbf{w}^{h_{n1}}$
h_{n1}
$\mathbf{w}^{h_{n2}}$
h_{n2}
$\mathbf{w}^{h_{n3}}$
h_{n3}
$\mathbf{w}^{o_{n1}}$
o_{n1}
$\mathbf{w}^{o_{n2}}$
o_{n2}
$\mathbf{w}^{o_{n3}}$
o_{n3}
$\mathbf{w}^{o_{n4}}$
o_{n4}
P_1
P_2
P_3
P_4
sum up to 1

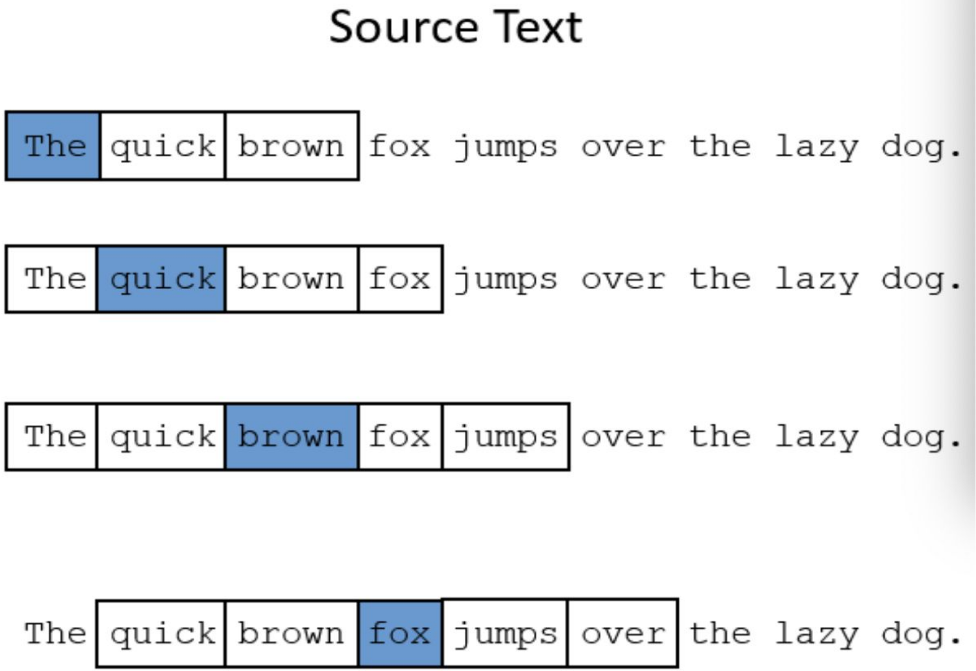
Back to the task of deriving feature vectors of words

How to derive word vectors?

The distributional hypothesis: words that have similar context will have similar meanings

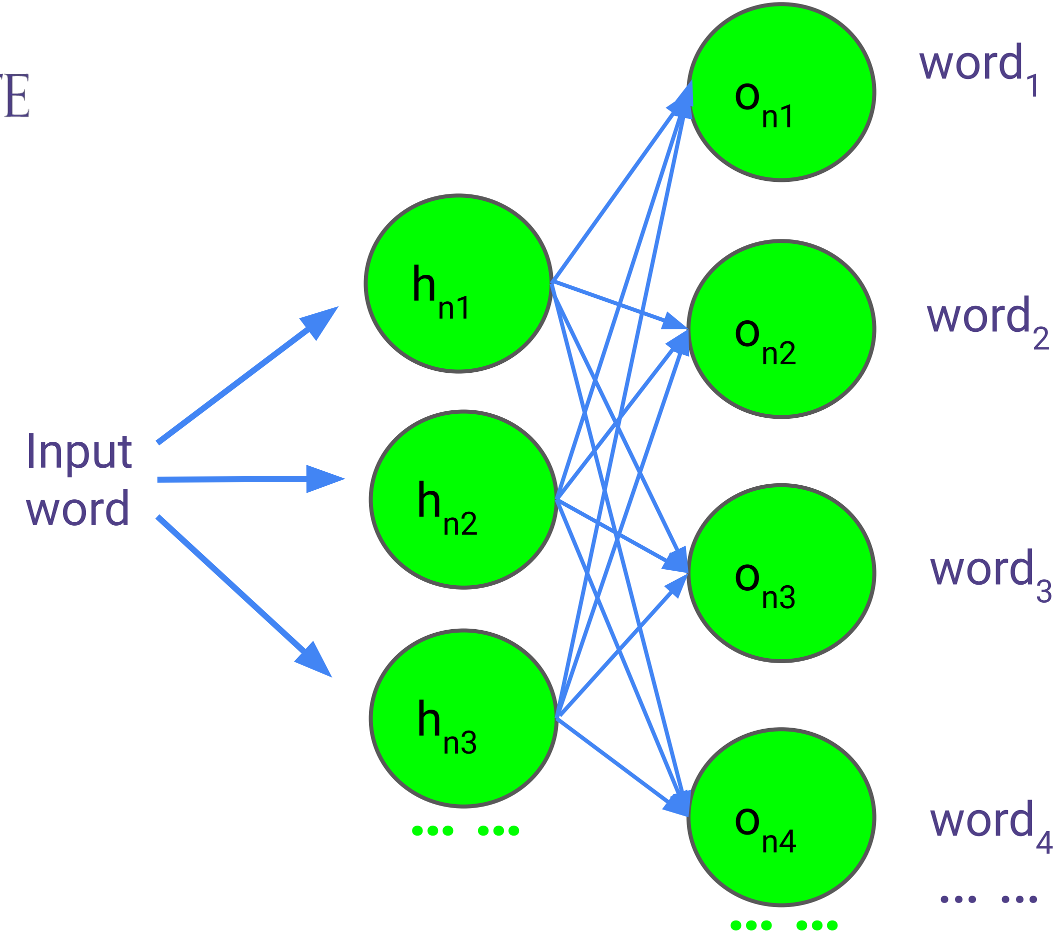
We'll derive the vector representations of words from their context words!

What is the context of a word?

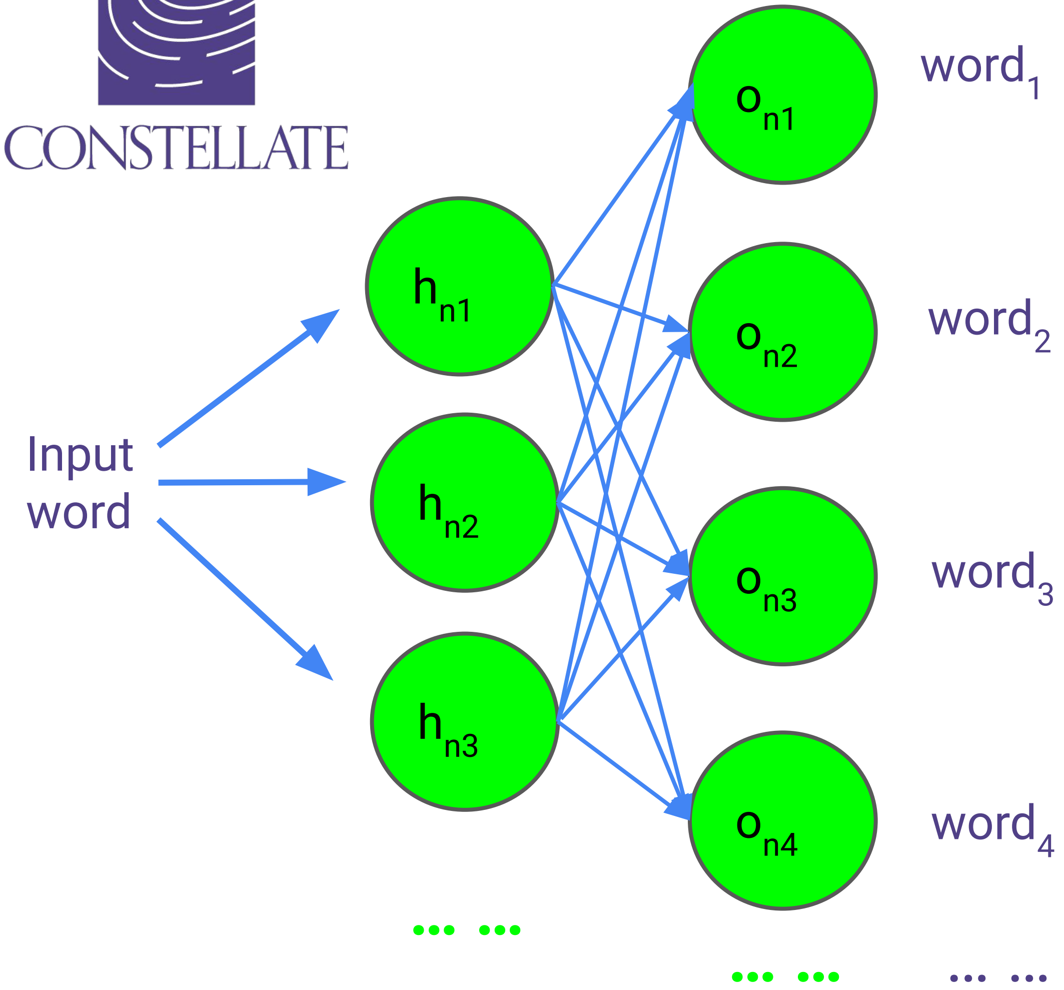


context window size = 2

(Fake) task of skip gram

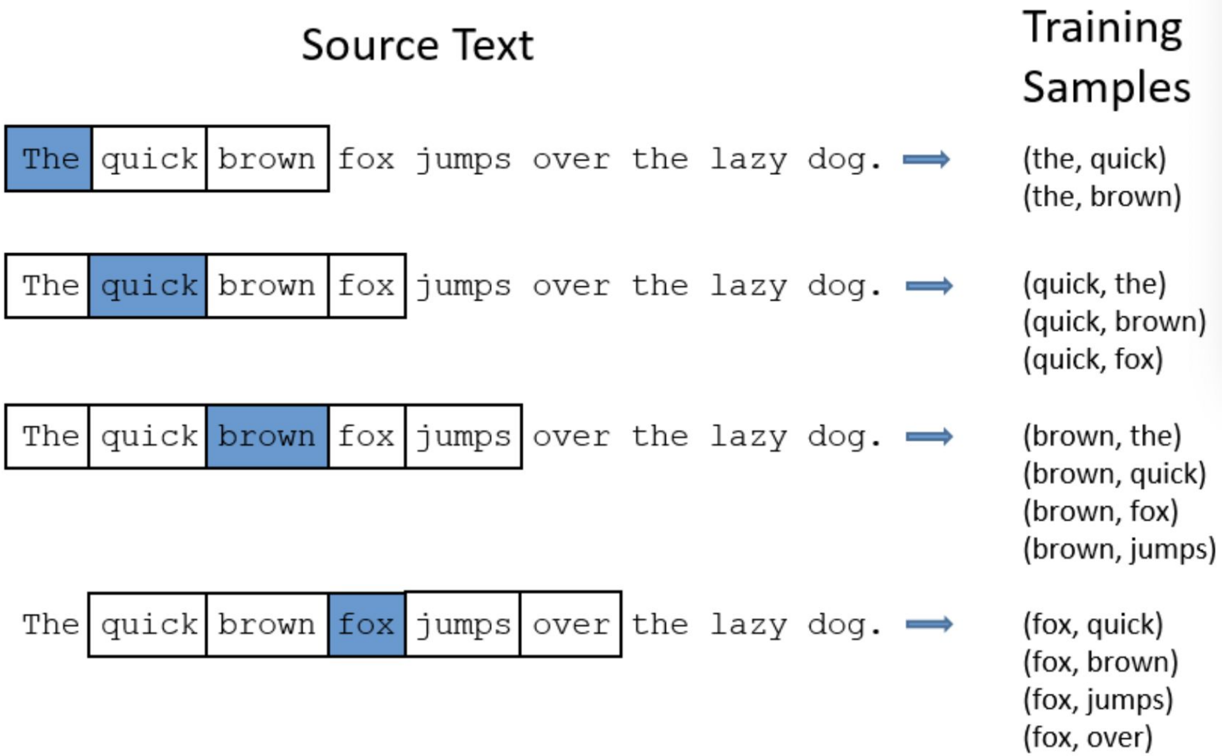


Real task of skip gram



Learn the weights of the hidden layer, and use these weights as the vector representations of the input words

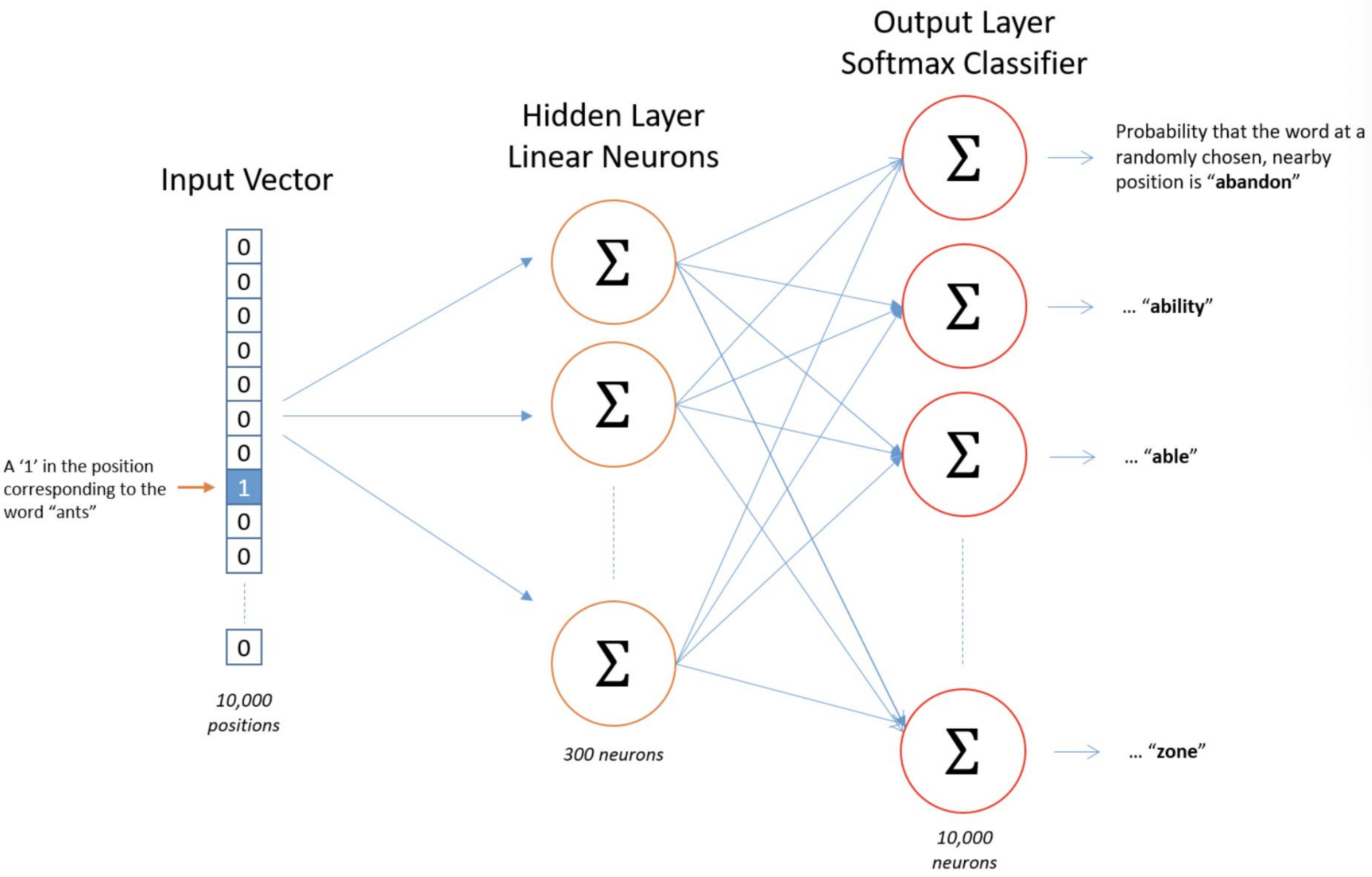
Skip gram


Source Text
Training Samples
The quick brown fox jumps over the lazy dog.
(the, quick)
(the, brown)
The quick brown fox jumps over the lazy dog.
(quick, the)
(quick, brown)
(quick, fox)
The quick brown fox jumps over the lazy dog.
(brown, the)
(brown, quick)
(brown, fox)
(brown, jumps)
The quick brown fox jumps over the lazy dog.
(fox, quick)
(fox, brown)
(fox, jumps)
(fox, over)

Task of skip-gram: a classification task

- Given a target word, say, ‘ants’, we train a classifier. For each word *w* in the vocabulary, is *w* likely to be a context word of ‘ants’?

Skip-gram



Input Vector
Hidden Layer
Linear Neurons
Output Layer
Softmax Classifier
0
0
0
0
0
0
0
1
0
0
0
A '1' in the position corresponding to the word "ants"
10,000 positions
300 neurons
10,000 neurons
Probability that the word at a randomly chosen, nearby position is "abandon"
... "ability"
... "able"
... "zone"

One-hot vector

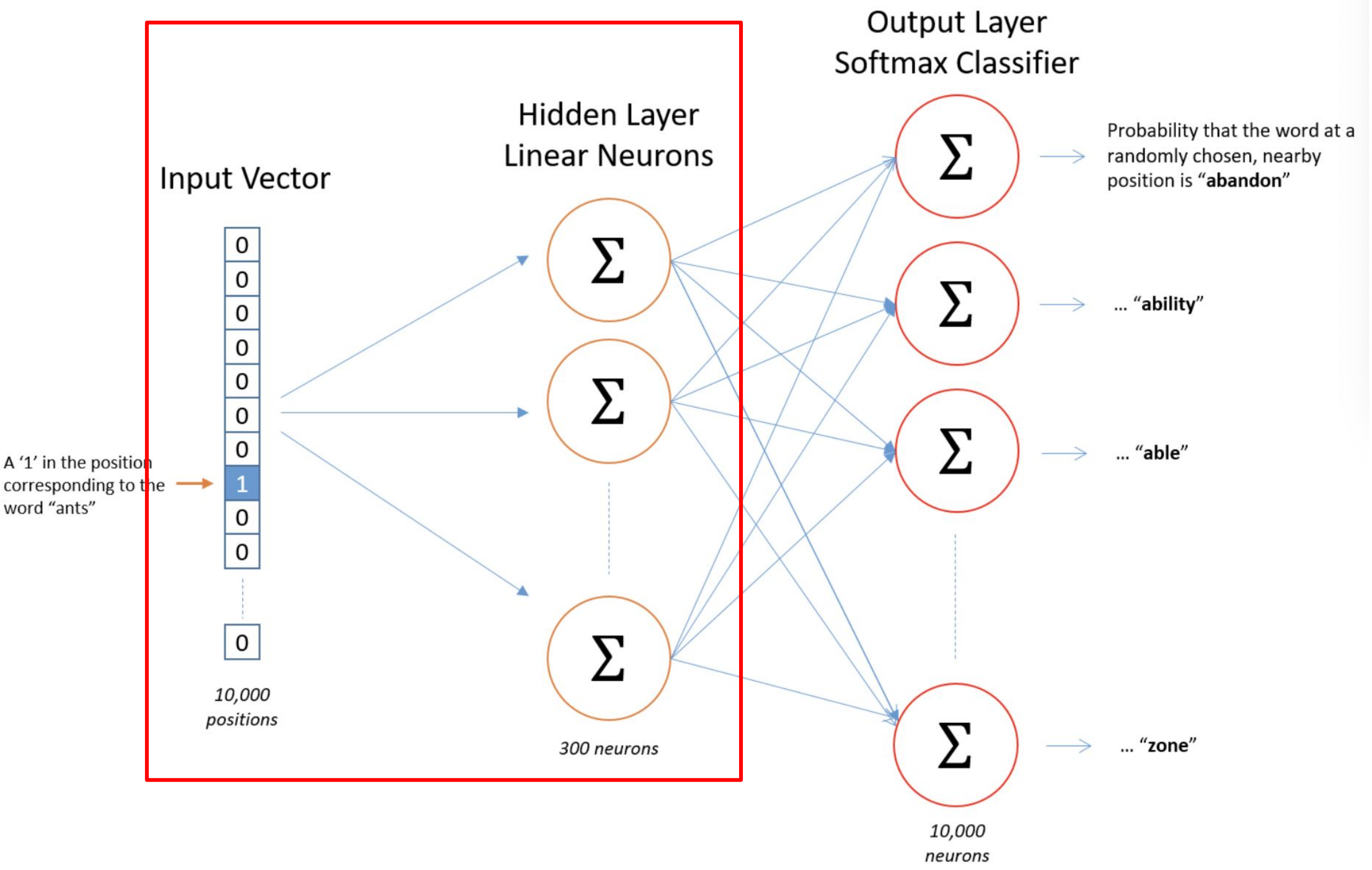
A vector with one value equal to 1 and all rest 0

For example, $\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \end{bmatrix}$

Skip-gram

Vocab size: 10000

300 neurons in the hidden layer



Given one input word

$$
\begin{bmatrix} 0 & 0 & 0 & 1 & \dots & 0 \end{bmatrix} \bullet \begin{bmatrix} w_1^{h_{n1}} & w_1^{h_{n2}} & \dots & w_1^{h_{n300}} \\ w_2^{h_{n1}} & w_2^{h_{n2}} & \dots & w_2^{h_{n300}} \\ w_3^{h_{n1}} & w_3^{h_{n2}} & \dots & w_3^{h_{n300}} \\ w_4^{h_{n1}} & w_4^{h_{n2}} & \dots & w_4^{h_{n300}} \\ \dots & \dots & \dots & \dots \\ w_{10000}^{h_{n1}} & w_{10000}^{h_{n2}} & \dots & w_{10000}^{h_{n300}} \end{bmatrix}
$$

$$1 \times 10000 \qquad\qquad 10000 \times 300$$

From input layer to hidden layer

CONSTELLATE

$$
\begin{bmatrix} 0 & 0 & 0 & 1 & \dots & 0 \end{bmatrix} \bullet
\begin{bmatrix}
w_1^{h_{n1}} & w_1^{h_{n2}} & \dots & w_1^{h_{n300}} \\
w_2^{h_{n1}} & w_2^{h_{n2}} & \dots & w_2^{h_{n300}} \\
w_3^{h_{n1}} & w_3^{h_{n2}} & \dots & w_3^{h_{n300}} \\
w_4^{h_{n1}} & w_4^{h_{n2}} & \dots & w_4^{h_{n300}} \\
\dots & \dots & \dots & \dots \\
w_{10000}^{h_{n1}} & w_{10000}^{h_{n2}} & \dots & w_{10000}^{h_{n300}}
\end{bmatrix}
= \begin{bmatrix} w_4^{h_{n1}} & w_4^{h_{n2}} & \dots & w_4^{h_{n300}} \end{bmatrix}
$$

$$1 \times 10000 \qquad\qquad 10000 \times 300 \qquad\qquad 1 \times 300$$

CONSTELLATE

Exercise

What's the effect of the one-hot vector?

Vocab size: 5

3 neurons in the hidden layer

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \end{bmatrix} \bullet \begin{bmatrix} 17 & 24 & 1 \\ 23 & 5 & 7 \\ 4 & 6 & 13 \\ 10 & 12 & 19 \\ 11 & 18 & 25 \end{bmatrix}$$

A look-up table

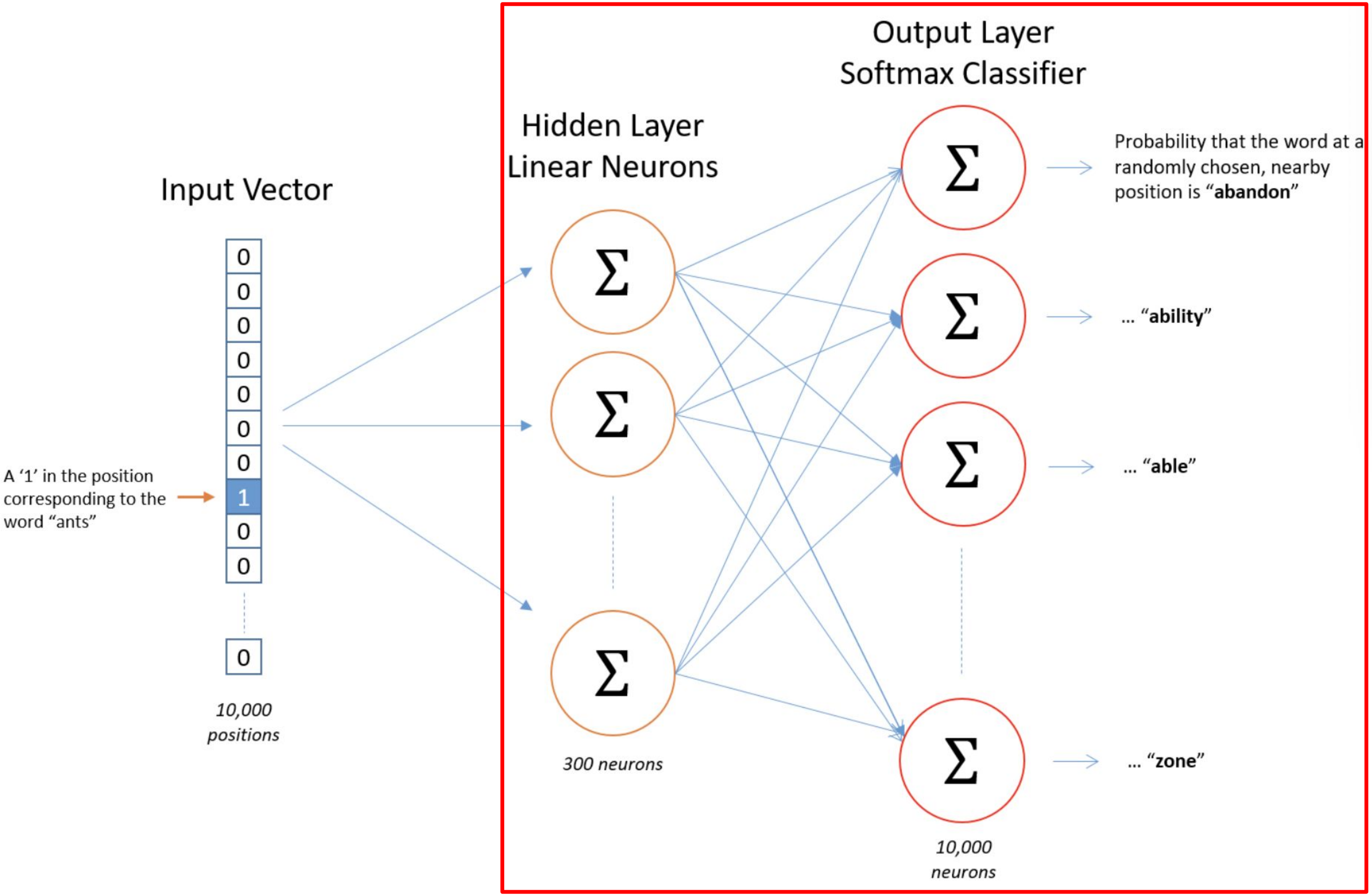
$$\begin{bmatrix} 0 & 0 & 0 & 1 & \dots & 0 \end{bmatrix} \bullet \begin{bmatrix} w_1^{h_{n1}} & w_1^{h_{n2}} & \dots & w_1^{h_{n300}} \\ w_2^{h_{n1}} & w_2^{h_{n2}} & \dots & w_2^{h_{n300}} \\ w_3^{h_{n1}} & w_3^{h_{n2}} & \dots & w_3^{h_{n300}} \\ w_4^{h_{n1}} & w_4^{h_{n2}} & \dots & w_4^{h_{n300}} \\ \dots & \dots & \dots & \dots \\ w_{10000}^{h_{n1}} & w_{10000}^{h_{n2}} & \dots & w_{10000}^{h_{n300}} \end{bmatrix} = \begin{bmatrix} w_4^{h_{n1}} & w_4^{h_{n2}} & \dots & w_4^{h_{n300}} \end{bmatrix}$$

CONSTELLATE

word2vec

$$[0 \quad 0 \quad 0 \quad 1 \quad \dots \quad 0] \bullet \begin{bmatrix} w_1^{h_{n1}} & w_1^{h_{n2}} & \dots & w_1^{h_{n300}} \\ w_2^{h_{n1}} & w_2^{h_{n2}} & \dots & w_2^{h_{n300}} \\ w_3^{h_{n1}} & w_3^{h_{n2}} & \dots & w_3^{h_{n300}} \\ \boxed{w_4^{h_{n1}} \quad w_4^{h_{n2}} \quad \dots \quad w_4^{h_{n300}}} \\ \dots & \dots & \dots & \dots \\ w_{10000}^{h_{n1}} & w_{10000}^{h_{n2}} & \dots & w_{10000}^{h_{n300}} \end{bmatrix} = [w_4^{h_{n1}} \quad w_4^{h_{n2}} \quad \dots \quad w_4^{h_{n300}}]$$

Skip-gram



Output Layer
Softmax Classifier
Hidden Layer
Linear Neurons
Input Vector
0
0
0
0
0
0
0
1
0
0
0
A '1' in the position corresponding to the word "ants"
10,000 positions
Σ
300 neurons
Probability that the word at a randomly chosen, nearby position is "abandon"
... "ability"
... "able"
... "zone"
10,000 neurons

From hidden layer to output layer

$$\begin{bmatrix} w_4^{h_{n1}} & w_4^{h_{n2}} & \dots & w_4^{h_{n300}} \end{bmatrix} \bullet \begin{bmatrix} w_1^{o_{n1}} & w_1^{o_{n2}} & \dots & w_1^{o_{n10000}} \\ w_2^{o_{n1}} & w_2^{o_{n2}} & \dots & w_2^{o_{n10000}} \\ \dots & \dots & & \\ w_{300}^{o_{n1}} & w_{300}^{o_{n2}} & \dots & w_{300}^{o_{n10000}} \end{bmatrix}$$

1×300

300×10000

From hidden layer to output layer

CONSTELLATE

$$\begin{bmatrix} w_4^{h_{n1}} & w_4^{h_{n2}} & \dots & w_4^{h_{n300}} \end{bmatrix} \bullet \begin{bmatrix} w_1^{o_{n1}} & w_1^{o_{n2}} & \dots & w_1^{o_{n10000}} \\ w_2^{o_{n1}} & w_2^{o_{n2}} & \dots & w_2^{o_{n10000}} \\ \dots & \dots & & \\ w_{300}^{o_{n1}} & w_{300}^{o_{n2}} & \dots & w_{300}^{o_{n10000}} \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{w}_4^h \cdot \mathbf{w}^{o_{n1}}, & \mathbf{w}_4^h \cdot \mathbf{w}^{o_{n2}} & ,\dots, & \mathbf{w}_4^h \cdot \mathbf{w}^{o_{n10000}} \end{bmatrix}$$

$$1 \times 10000$$

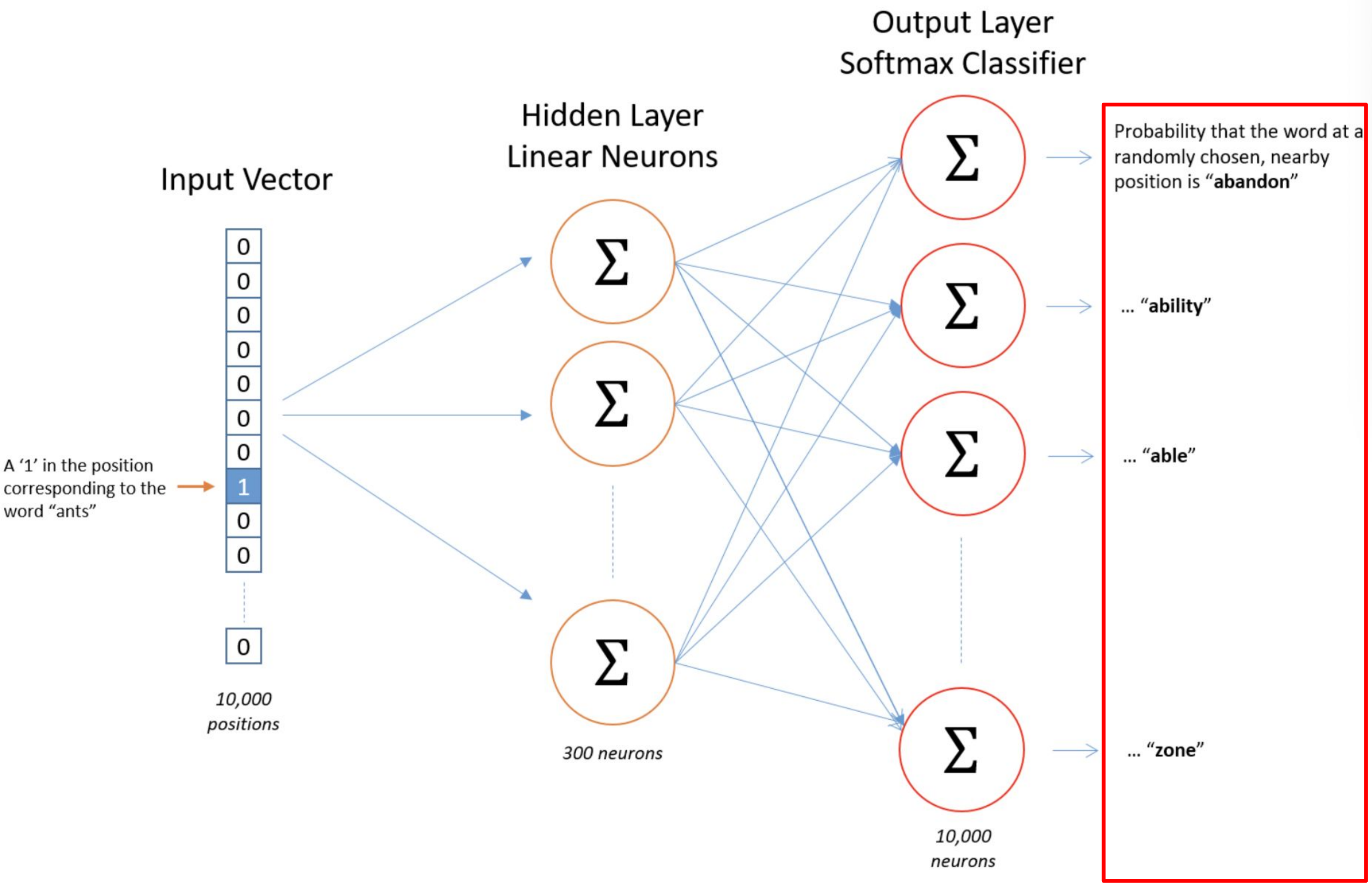
Softmax (activation function)

Softmax function

The softmax function takes a vector of K values $[z_1, z_2, \ldots z_K]$, and maps the values to a probability distribution where each value is in the range (0,1) and the probabilities sum up to one.

$$\text{softmax}(\mathbf{z}) = \left[\frac{e^{\mathbf{z}_1}}{\sum_{i=1}^{K} e^{\mathbf{z}_i}}, \quad \frac{e^{\mathbf{z}_2}}{\sum_{i=1}^{K} e^{\mathbf{z}_i}} \quad , \ldots, \frac{e^{\mathbf{z}_K}}{\sum_{i=1}^{K} e^{\mathbf{z}_i}} \right]$$

Skip-gram


Input Vector
0
0
0
0
0
0
0
1
0
0
0
A '1' in the position corresponding to the word "ants"
10,000 positions
Hidden Layer
Linear Neurons
300 neurons
Output Layer
Softmax Classifier
10,000 neurons
Probability that the word at a randomly chosen, nearby position is "abandon"
... "ability"
... "able"
... "zone"

Minimize the error

- For those words in the vocabulary which are the context words of the input word, we want the output probabilities for them to be high.

- For those words in the vocabulary which are not the context words of the input word, we want the output probabilities for them to be low.

Minimize the error

- When we calculate the outputs, we take in an input and go from the hidden layer to the output layer.

- When we calculate the error to minimize it, we go from the output layer back to the hidden layer. This is called backward propagation of errors (backpropagation).

Word similarity

The distributional hypothesis:
words that have similar context will have similar meanings

Word similarity

- If two words have very similar surrounding words, then they are very similar in meaning.
- This means, the two words will have very similar output probability distribution with regard to the words in the vocabulary.
- This also means, two words with a similar meaning will ultimately get very similar vector representations.

Word similarity visualized in a 2-dim space

Jurafsky, Daniel, and James H. Martin (2023)

CONSTELLATE

Any questions?

Learning objectives

- Concepts
  - Understand some basic concepts in neural networks
    - feature, weight, bias, vector, matrix……
    - neuron, activation function, hidden layer……
  - Understand a neuron as a computation unit
  - Understand the fundamental algorithms underlying NNs

- Hands-on computation
  - Know how to do matrix multiplication by hand

Calculate all observations at one time: Matrix multiplication

$$\begin{array}{c} \mathbf{x}^{(1)} \\ \mathbf{x}^{(2)} \\ \\ \mathbf{x}^{(m)} \end{array} \begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \dots & x_n^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \dots & x_n^{(2)} \\ \dots & \dots & \dots & \dots \\ x_1^{(m)} & x_2^{(m)} & \dots & x_n^{(m)} \end{bmatrix} \bullet \begin{array}{c} \begin{array}{cccc} \mathbf{w}^{c1} & \mathbf{w}^{c2} & & \mathbf{w}^{cK} \end{array} \\ \begin{bmatrix} w_1^{c1} & w_1^{c2} & \dots & w_1^{cK} \\ w_2^{c1} & w_2^{c2} & \dots & w_2^{cK} \\ \dots & \dots & \dots & \dots \\ w_n^{c1} & w_n^{c2} & \dots & w_n^{cK} \end{bmatrix} \end{array} + \begin{bmatrix} \boxed{b^{c1}} & b^{c2} & \dots & b^{cK} \end{bmatrix}$$

$$= \begin{bmatrix} \boxed{\mathbf{w}^{c1}\mathbf{x}^{(1)} + b^{c1}} & \mathbf{w}^{c2}\mathbf{x}^{(1)} + b^{c2} & \dots & \mathbf{w}^{cK}\mathbf{x}^{(1)} + b^{cK} \\ \mathbf{w}^{c1}\mathbf{x}^{(2)} + b^{c1} & \mathbf{w}^{c2}\mathbf{x}^{(2)} + b^{c2} & \dots & \mathbf{w}^{cK}\mathbf{x}^{(2)} + b^{cK} \\ \dots & \dots & \dots & \dots \\ \mathbf{w}^{c1}\mathbf{x}^{(m)} + b^{c1} & \mathbf{w}^{c2}\mathbf{x}^{(m)} + b^{c2} & \dots & \mathbf{w}^{cK}\mathbf{x}^{(m)} + b^{cK} \end{bmatrix}$$

Calculate all observations at one time: Matrix multiplication

$$
\begin{array}{c} \mathbf{x}^{(1)} \\ \mathbf{x}^{(2)} \\ \\ \mathbf{x}^{(m)} \end{array}
\begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \dots & x_n^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \dots & x_n^{(2)} \\ \dots & \dots & \dots & \dots \\ x_1^{(m)} & x_2^{(m)} & \dots & x_n^{(m)} \end{bmatrix}
\bullet
\begin{array}{c}
\begin{array}{cccc} \mathbf{w}^{c1} & \mathbf{w}^{c2} & & \mathbf{w}^{cK} \end{array} \\
\begin{bmatrix} w_1^{c1} & w_1^{c2} & \dots & w_1^{cK} \\ w_2^{c1} & w_2^{c2} & \dots & w_2^{cK} \\ \dots & \dots & \dots & \dots \\ w_n^{c1} & w_n^{c2} & \dots & w_n^{cK} \end{bmatrix}
\end{array}
+ \begin{bmatrix} b^{c1} & b^{c2} & \dots & \boxed{b^{cK}} \end{bmatrix}
$$

$$
= \begin{bmatrix} \mathbf{w}^{c1}\mathbf{x}^{(1)} + b^{c1} & \mathbf{w}^{c2}\mathbf{x}^{(1)} + b^{c2} & \dots & \boxed{\mathbf{w}^{cK}\mathbf{x}^{(1)} + b^{cK}} \\ \mathbf{w}^{c1}\mathbf{x}^{(2)} + b^{c1} & \mathbf{w}^{c2}\mathbf{x}^{(2)} + b^{c2} & \dots & \mathbf{w}^{cK}\mathbf{x}^{(2)} + b^{cK} \\ \dots & \dots & \dots & \dots \\ \mathbf{w}^{c1}\mathbf{x}^{(m)} + b^{c1} & \mathbf{w}^{c2}\mathbf{x}^{(m)} + b^{c2} & \dots & \mathbf{w}^{cK}\mathbf{x}^{(m)} + b^{cK} \end{bmatrix}
$$

CONSTELLATE

Calculate all observations at one time: Matrix multiplication

$$
\begin{array}{l}
\mathbf{x}^{(1)} \\
\mathbf{x}^{(2)} \\
 \\
\mathbf{x}^{(m)}
\end{array}
\begin{bmatrix}
x_1^{(1)} & x_2^{(1)} & \dots & x_n^{(1)} \\
\boxed{x_1^{(2)} & x_2^{(2)} & \dots & x_n^{(2)}} \\
\dots & \dots & \dots & \dots \\
x_1^{(m)} & x_2^{(m)} & \dots & x_n^{(m)}
\end{bmatrix}
\bullet
\begin{array}{c}
\begin{array}{cccc} \mathbf{w}^{c}_{1} & \mathbf{w}^{c}_{2} & & \mathbf{w}^{cK} \end{array} \\
\begin{bmatrix}
w_1^{c1} & w_1^{c2} & \dots & w_1^{cK} \\
w_2^{c1} & w_2^{c2} & \dots & w_2^{cK} \\
\dots & \dots & \dots & \dots \\
w_n^{c1} & w_n^{c2} & \dots & w_n^{cK}
\end{bmatrix}
\end{array}
+
\begin{bmatrix} \boxed{b^{c1}} & b^{c2} & \dots & b^{cK} \end{bmatrix}
$$

$$
=
\begin{bmatrix}
\mathbf{w}^{c1}\mathbf{x}^{(1)} + b^{c1} & \mathbf{w}^{c2}\mathbf{x}^{(1)} + b^{c2} & \dots & \mathbf{w}^{cK}\mathbf{x}^{(1)} + b^{cK} \\
\boxed{\mathbf{w}^{c1}\mathbf{x}^{(2)} + b^{c1}} & \mathbf{w}^{c2}\mathbf{x}^{(2)} + b^{c2} & \dots & \mathbf{w}^{cK}\mathbf{x}^{(2)} + b^{cK} \\
\dots & \dots & \dots & \dots \\
\mathbf{w}^{c1}\mathbf{x}^{(m)} + b^{c1} & \mathbf{w}^{c2}\mathbf{x}^{(m)} + b^{c2} & \dots & \mathbf{w}^{cK}\mathbf{x}^{(m)} + b^{cK}
\end{bmatrix}
$$

CONSTELLATE

Calculate all observations at one time: Matrix multiplication

$$
\begin{array}{c}
\mathbf{x}^{(1)} \\ \mathbf{x}^{(2)} \\ \\ \mathbf{x}^{(m)}
\end{array}
\begin{bmatrix}
x_1^{(1)} & x_2^{(1)} & \dots & x_n^{(1)} \\
x_1^{(2)} & x_2^{(2)} & \dots & x_n^{(2)} \\
\dots & \dots & \dots & \dots \\
x_1^{(m)} & x_2^{(m)} & \dots & x_n^{(m)}
\end{bmatrix}
\bullet
\begin{array}{c}
\begin{array}{cccc} \mathbf{w}^{c}_{1} & \mathbf{w}^{c}_{2} & & \mathbf{w}^{cK} \end{array} \\
\begin{bmatrix}
w_1^{c1} & w_1^{c2} & \dots & w_1^{cK} \\
w_2^{c1} & w_2^{c2} & \dots & w_2^{cK} \\
\dots & \dots & \dots & \dots \\
w_n^{c1} & w_n^{c2} & \dots & w_n^{cK}
\end{bmatrix}
\end{array}
+ [b^{c1} \quad b^{c2} \quad \dots \quad \boxed{b^{cK}}]
$$

$$
= \begin{bmatrix}
\mathbf{w}^{c1}\mathbf{x}^{(1)} + b^{c1} & \mathbf{w}^{c2}\mathbf{x}^{(1)} + b^{c2} & \dots & \mathbf{w}^{cK}\mathbf{x}^{(1)} + b^{cK} \\
\mathbf{w}^{c1}\mathbf{x}^{(2)} + b^{c1} & \mathbf{w}^{c2}\mathbf{x}^{(2)} + b^{c2} & \dots & \boxed{\mathbf{w}^{cK}\mathbf{x}^{(2)} + b^{cK}} \\
\dots & \dots & \dots & \dots \\
\mathbf{w}^{c1}\mathbf{x}^{(m)} + b^{c1} & \mathbf{w}^{c2}\mathbf{x}^{(m)} + b^{c2} & \dots & \mathbf{w}^{cK}\mathbf{x}^{(m)} + b^{cK}
\end{bmatrix}
$$

Calculate all observations at one time: Matrix multiplication

$$\begin{array}{c}\mathbf{x}^{(1)}\\[2ex] \mathbf{x}^{(2)}\\[2ex] \\[2ex] \mathbf{x}^{(m)}\end{array}
\begin{bmatrix} x_1^{(1)} & x_2^{(1)} & \dots & x_n^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \dots & x_n^{(2)} \\ \dots & \dots & \dots & \dots \\ x_1^{(m)} & x_2^{(m)} & \dots & x_n^{(m)} \end{bmatrix}
\bullet
\begin{array}{c}
\begin{array}{cccc} \mathbf{w}^{c}_{1} & \mathbf{w}^{c}_{2} & \quad & \mathbf{w}^{cK} \end{array} \\
\begin{bmatrix} w_1^{c1} & w_1^{c2} & \dots & w_1^{cK} \\ w_2^{c1} & w_2^{c2} & \dots & w_2^{cK} \\ \dots & \dots & \dots & \dots \\ w_n^{c1} & w_n^{c2} & \dots & w_n^{cK} \end{bmatrix}
\end{array}
+ \begin{bmatrix} b^{c1} & b^{c2} & \dots & b^{cK} \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{w}^{c1}\mathbf{x}^{(1)} + b^{c1} & \mathbf{w}^{c2}\mathbf{x}^{(1)} + b^{c2} & \dots & \mathbf{w}^{cK}\mathbf{x}^{(1)} + b^{cK} \\ \mathbf{w}^{c1}\mathbf{x}^{(2)} + b^{c1} & \mathbf{w}^{c2}\mathbf{x}^{(2)} + b^{c2} & \dots & \mathbf{w}^{cK}\mathbf{x}^{(2)} + b^{cK} \\ \dots & \dots & \dots & \dots \\ \mathbf{w}^{c1}\mathbf{x}^{(m)} + b^{c1} & \mathbf{w}^{c2}\mathbf{x}^{(m)} + b^{c2} & \dots & \mathbf{w}^{cK}\mathbf{x}^{(m)} + b^{cK} \end{bmatrix}$$

Calculate all observations at one time: Matrix multiplication

$$
\begin{array}{c}
\mathbf{x}^{(1)} \\ \mathbf{x}^{(2)} \\ \\ \mathbf{x}^{(m)}
\end{array}
\begin{bmatrix}
x_1^{(1)} & x_2^{(1)} & \dots & x_n^{(1)} \\
x_1^{(2)} & x_2^{(2)} & \dots & x_n^{(2)} \\
\dots & \dots & \dots & \dots \\
x_1^{(m)} & x_2^{(m)} & \dots & x_n^{(m)}
\end{bmatrix}
\bullet
\begin{array}{c}
\begin{array}{cccc} \mathbf{w}^{c}\,1 & \mathbf{w}^{c}\,2 & & \mathbf{w}^{cK} \end{array} \\
\begin{bmatrix}
w_1^{c1} & w_1^{c2} & \dots & w_1^{cK} \\
w_2^{c1} & w_2^{c2} & \dots & w_2^{cK} \\
\dots & \dots & \dots & \dots \\
w_n^{c1} & w_n^{c2} & \dots & w_n^{cK}
\end{bmatrix}
\end{array}
+ \begin{bmatrix} b^{c1} & b^{c2} & \dots & \boxed{b^{cK}} \end{bmatrix}
$$

$$
= \begin{bmatrix}
\mathbf{w}^{c1}\mathbf{x}^{(1)} + b^{c1} & \mathbf{w}^{c2}\mathbf{x}^{(1)} + b^{c2} & \dots & \mathbf{w}^{cK}\mathbf{x}^{(1)} + b^{cK} \\
\mathbf{w}^{c1}\mathbf{x}^{(2)} + b^{c1} & \mathbf{w}^{c2}\mathbf{x}^{(2)} + b^{c2} & \dots & \mathbf{w}^{cK}\mathbf{x}^{(2)} + b^{cK} \\
\dots & \dots & \dots & \dots \\
\mathbf{w}^{c1}\mathbf{x}^{(m)} + b^{c1} & \mathbf{w}^{c2}\mathbf{x}^{(m)} + b^{c2} & \dots & \boxed{\mathbf{w}^{cK}\mathbf{x}^{(m)} + b^{cK}}
\end{bmatrix}
$$

References

Jiang, L. (2020). A Visual Explanation of Gradient Descent Methods (Momentum, Ada-Grad, RMSProp, Adam). June-2020. [online]. *Available:* https://towardsdatascience.com/a-visual-explanation-of-gradient-descent-methods-momentum-adagrad-rmsprop-adam-f898b102325c

Jurafsky, Daniel, and James H. Martin. (2023). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition.*

McCormick, C. (2016). Word2vec tutorial-the skip-gram model. *Apr-2016.[Online]. Available:* http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, *26*.