How do we handle words?

Last time, we saw how to improve on BOW representations of documents

- TF-IDF
- PMI
- LSA
- NMF

Now, we ask how should individual words be represented?

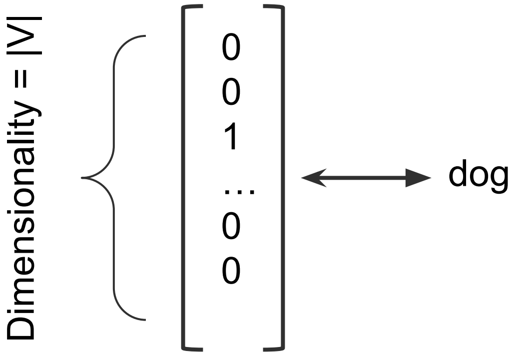
How do we handle words?

Statistical ML models expect numeric inputs. How do we represent words numerically?

How do we handle words?

Statistical ML models expect numeric inputs. How do we represent words numerically?

The simplest idea is to use a **one-hot encoding**



How do we handle words?

Statistical ML models expect numeric inputs. How do we represent words numerically?

The simplest idea is to use a **one-hot encoding**

$$\text{Dimensionality} = |V| \left\{ \begin{bmatrix} 0 \\ 0 \\ 1 \\ \dots \\ 0 \\ 0 \end{bmatrix} \right. \longleftrightarrow \text{dog} \qquad\qquad \begin{bmatrix} 0 \\ 0 \\ 0 \\ \dots \\ 1 \\ \dots \\ 0 \end{bmatrix} \longleftrightarrow \text{pomeranian}$$

How do we handle words?

Statistical ML models expect numeric inputs. How do we represent words numerically?

The simplest idea is to use a **one-hot encoding**

$$\text{Dimensionality} = |V| \left\{ \begin{bmatrix} 0 \\ 0 \\ 1 \\ \cdots \\ 0 \\ 0 \end{bmatrix} \right. \longleftrightarrow \text{dog} \qquad \begin{bmatrix} 0 \\ 0 \\ 0 \\ \cdots \\ 1 \\ \cdots \\ 0 \end{bmatrix} \longleftrightarrow \text{pomeranian}$$

What are some problems with this approach?

How do we handle words?

Statistical ML models expect numeric inputs. How do we represent words numerically?

The simplest idea is to use a **one-hot encoding**

$$\text{Dimensionality} = |V| \left\{ \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \right. \longleftrightarrow \text{dog} \qquad \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \longleftrightarrow \text{pomeranian}$$

Pixels have relative meaning

vector(dog) • vector(pomeranian) = 0 = vector(dog) • vector(bookshelf) = …

How do we handle words?

Statistical ML models expect numeric inputs. How do we represent words numerically?

The simplest idea is to use a **one-hot encoding**

Pixels have relative meaning

Dimensionality = |V|

$$\begin{bmatrix} 0 \\ 0 \\ 1 \\ \dots \\ 0 \\ 0 \end{bmatrix} \qquad \begin{bmatrix} 0 \\ 0 \\ \\ \\ \dots \\ 0 \end{bmatrix}$$

omeranian

What approaches let us represent word **meanings**?

vector(dog) • vector(pomeranian) = 0 = vector(dog) • vector(bookshelf) = …

Distributional Hypothesis

Statistical ML models expect numeric inputs. What is the **meaning** of a word?

Word to search for: happy Search WordNet

Display Options: (Select option to change) Change

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations
Display options for sense: (gloss) "an example sentence"

Adjective

- S: (adj) **happy** (enjoying or showing or marked by joy or pleasure) *"a happy smile"; "spent many happy days on the beach"; "a happy marriage"*
  - ***see also***
    - S: (adj) cheerful (being full of or promoting cheer; having or showing good spirits) *"her cheerful nature"; "a cheerful greeting"; "a cheerful room"; "as cheerful as anyone confined to a hospital bed could be"*
    - S: (adj) contented, content (satisfied or showing satisfaction with things as they are) *"a contented smile"*
    - S: (adj) glad (showing or causing joy and pleasure; especially made happy) *"glad you are here"; "glad that they succeeded"; "gave a glad shout"; "a glad smile"; "heard the glad news"; "a glad occasion"*
    - S: (adj) elated (exultantly proud and joyful; in high spirits) *"the elated winner"; "felt elated and excited"*
    - S: (adj) euphoric (exaggerated feeling of well-being or elation)
    - S: (adj) felicitous (exhibiting an agreeably appropriate manner or style) *"a felicitous speaker"*
    - S: (adj) joyful (full of or producing joy) *"make a joyful noise"; "a joyful occasion"*
    - S: (adj) joyous (full of or characterized by joy) *"felt a joyous abandon"; "joyous laughter"*

Distributional Hypothesis

Statistical ML models expect numeric inputs. What is the **meaning** of a word?

Word to search for: happy Search WordNet

Display Options: (Select option to change) Change

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Display options for sense: (gloss) "an example sentence"

Adjective

- S: (adj) **happy** (enjoying or showing or marked by joy or pleasure) *"a happy smile"; "spent many happy days on the beach"; "a happy marriage"*
  - ***see also***
    - S: (adj) cheerful (being full of or promoting cheer; having or showing good spirits) *"her cheerful nature"; "a cheerful greeting"; "a cheerful room"; "as cheerful as anyone confined to a hospital bed could be"*
    - S: (adj) contented, content (satisfied or showing satisfaction with things as they are) *"a contented smile"*
    - S: (adj) glad (showing or causing joy and pleasure; especially made happy) *"glad you are here"; "glad that they succeeded"; "gave a glad shout"; "a glad smile"; "heard the glad news"; "a glad occasion"*
    - S: (adj) elated (exultantly proud and joyful; in high spirits) *"the elated winner"; "felt elated and excited"*
    - S: (adj) euphoric (exaggerated feeling of well-being or elation)
    - S: (adj) felicitous (exhibiting an agreeably appropriate manner or style) *"a felicitous speaker"*
    - S: (adj) joyful (full of or producing joy) *"make a joyful noise"; "a joyful occasion"*
    - S: (adj) joyous (full of or characterized by joy) *"felt a joyous abandon"; "joyous laughter"*

Word to search for: inebriated Search WordNet

Display Options: (Select option to change) Change

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Display options for sense: (gloss) "an example sentence"

Verb

- **S: (v) exhilarate, tickle pink, inebriate, thrill, exalt, beatify (fill with sublime emotion) *"The children were thrilled at the prospect of going to the movies"; "He was inebriated by his phenomenal success"***

Distributional Hypothesis

Statistical ML models expect numeric inputs. What is the **meaning** of the word *bardiwac*

John enjoys sipping bardiwac in the warm weather

Distributional Hypothesis

Statistical ML models expect numeric inputs. What is the **meaning** of the word *bardiwac*

John enjoys sipping bardiwac in the warm weather

She knocked over the glass of bardiwac, and now there’s a stain on the carpet

Distributional Hypothesis

Statistical ML models expect numeric inputs. What is the **meaning** of the word *bardiwac*

John enjoys sipping bardiwac in the warm weather

She knocked over the glass of bardiwac, and now there's a stain on the carpet

He drank too much bardiwac, so he can't drive tonight

Distributional Hypothesis

Statistical ML models expect numeric inputs. What is the **meaning** of the word *bardiwac*

John enjoys sipping bardiwac in the warm weather

She knocked over the glass of bardiwac, and now there’s a stain on the carpet

He drank too much bardiwac, so he can’t drive tonight

The bardiwac grapes didn’t fare well in this summer’s heat

Distributional Hypothesis

Statistical ML models expect numeric inputs. What is the **meaning** of the word *bardiwac*

John enjoys sipping bardiwac in the warm weather

She knocked over

n the carpet

He drank too much

The bardiwac grap

Distributional Hypothesis

"You shall know a word by the company it keeps."

J.R. Firth (1957)

Distributional Hypothesis

Our goal is to describe a word's meaning using the contexts it appears in

Put another way, can we find **representations** so that words that appear in similar contexts are represented similarly and words that do not appear in similar contexts are represented dissimilarly

Harry and Sally sat <u>on the river</u> bank <u>and had a</u> picnic

The robbers <u>struck the largest</u> bank <u>in all of</u> New York

It doesn't matter <u>if you always</u> bank <u>the shot you</u> just need to make it in

Distributional Hypothesis

Our goal is to describe a word's meaning using the contexts it appears in

Put another way, can we find **representations** so that words that appear in similar contexts are represented similarly and words that do not appear in similar contexts are represented dissimilarly

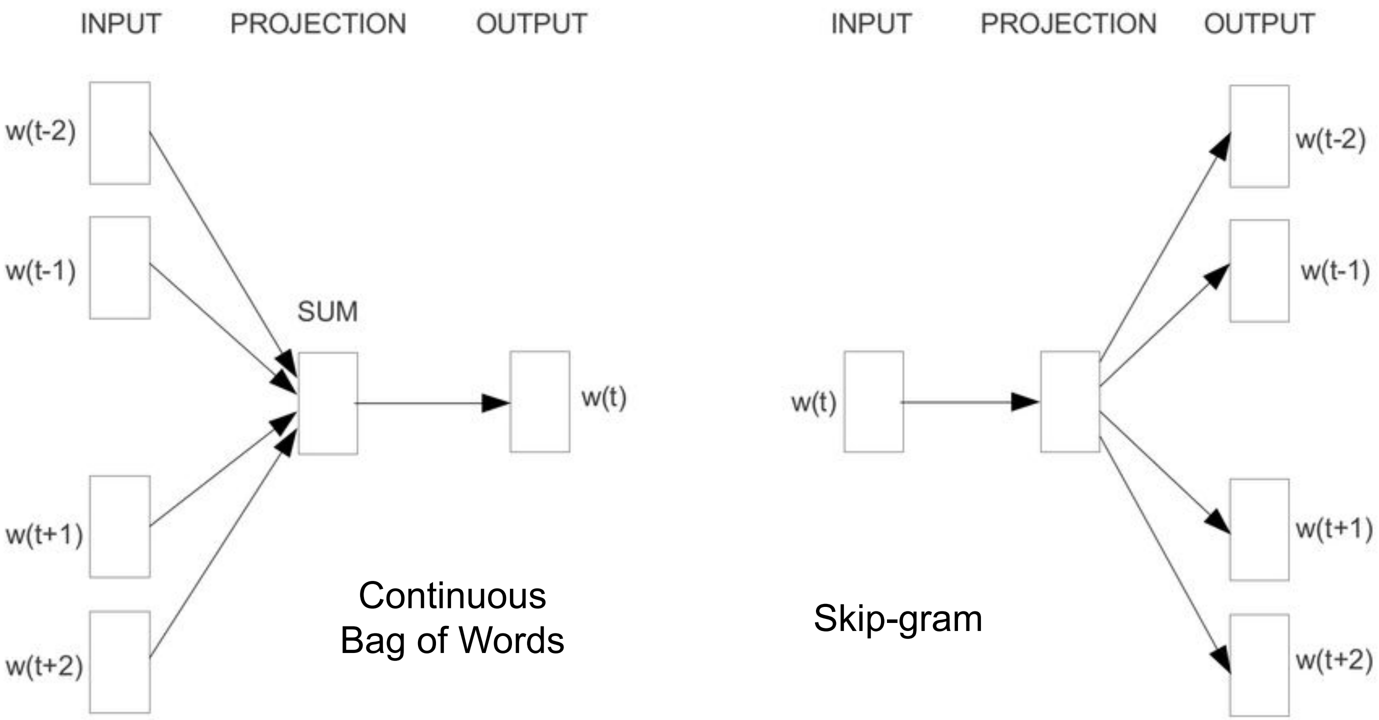
Harry and Sally sat on the river bank and had a picnic

The robbers struck the largest bank in all of New York

It doesn't matter if you always bank the shot you just need to make it in

Tasks: (1) predict center word from context (2) predict context from center word

Word2Vec

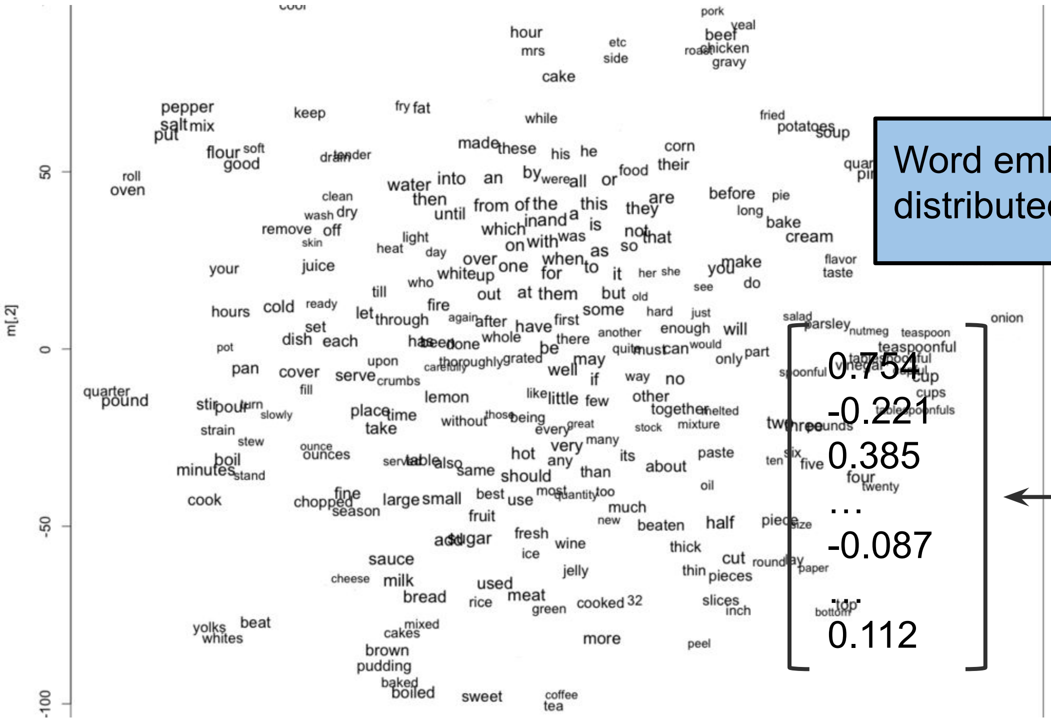


Continuous
Bag of Words

Skip-gram

Efficient Estimation of Word Representations in Vector Space, Mikolov et al., 2013a

Word2Vec



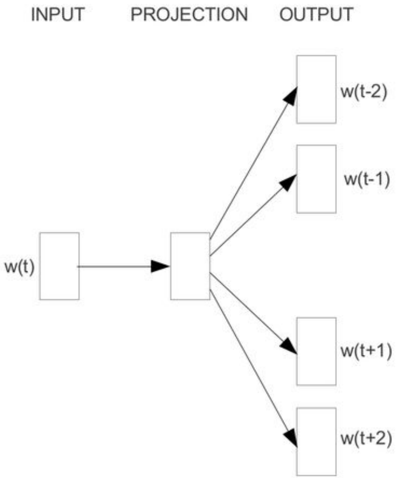
Word embeddings are distributed representations

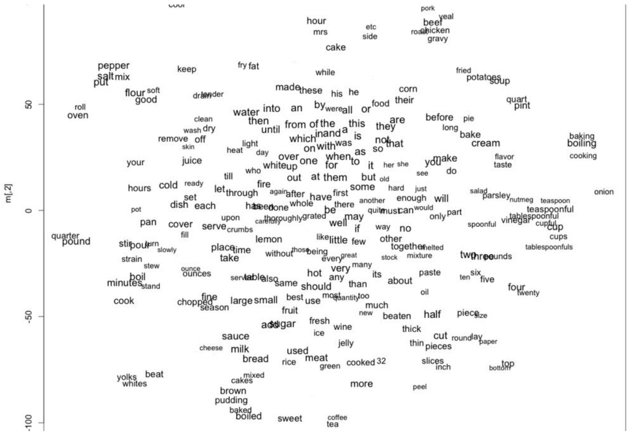
$$\begin{bmatrix} 0.754 \\ -0.221 \\ 0.385 \\ \dots \\ -0.087 \\ \dots \\ 0.112 \end{bmatrix} \longleftrightarrow \text{pomeranian}$$

Also see: A Neural Probabilistic Language Model, Bengio, et al., 2003

Skip-gram model

Harry and Sally sat on the river bank and had a picnic
The robbers struck the largest bank in all of New York
It doesn't matter if you always bank the shot you just need to make it in





Corpus

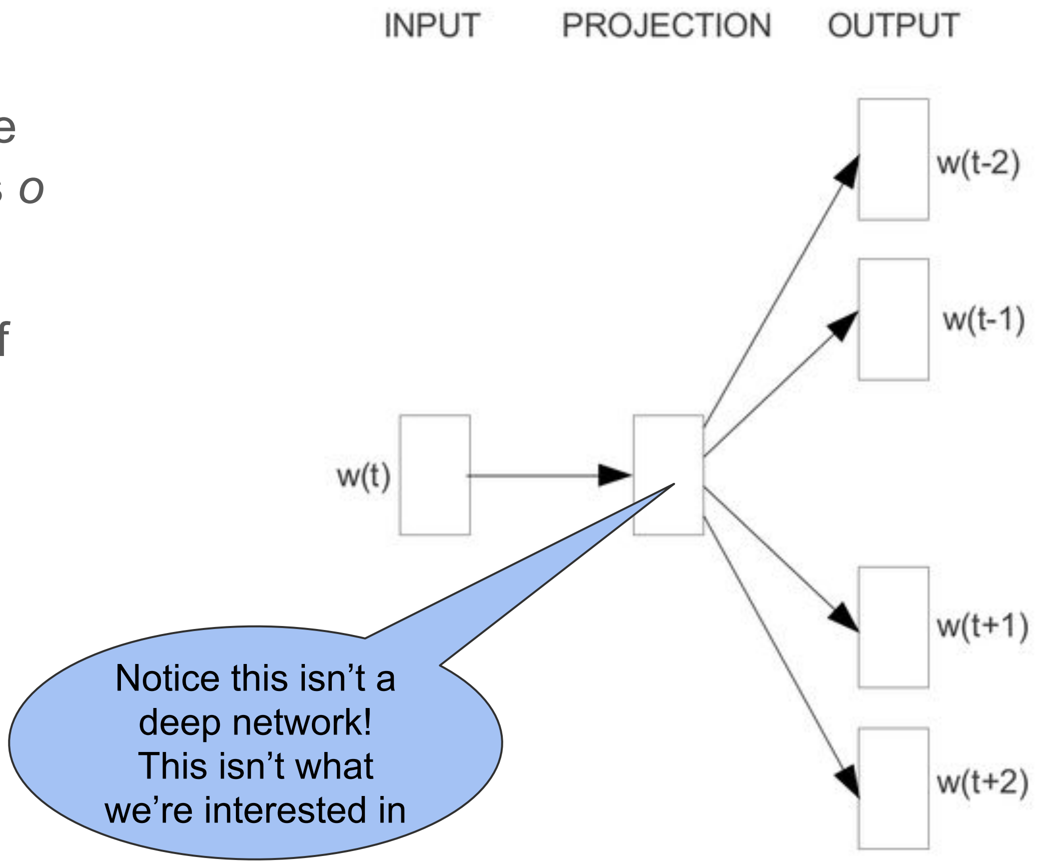
Model

2-d representation

Skip-gram model

The skip-gram model learns the probability of the context words *o* given the center word *c*

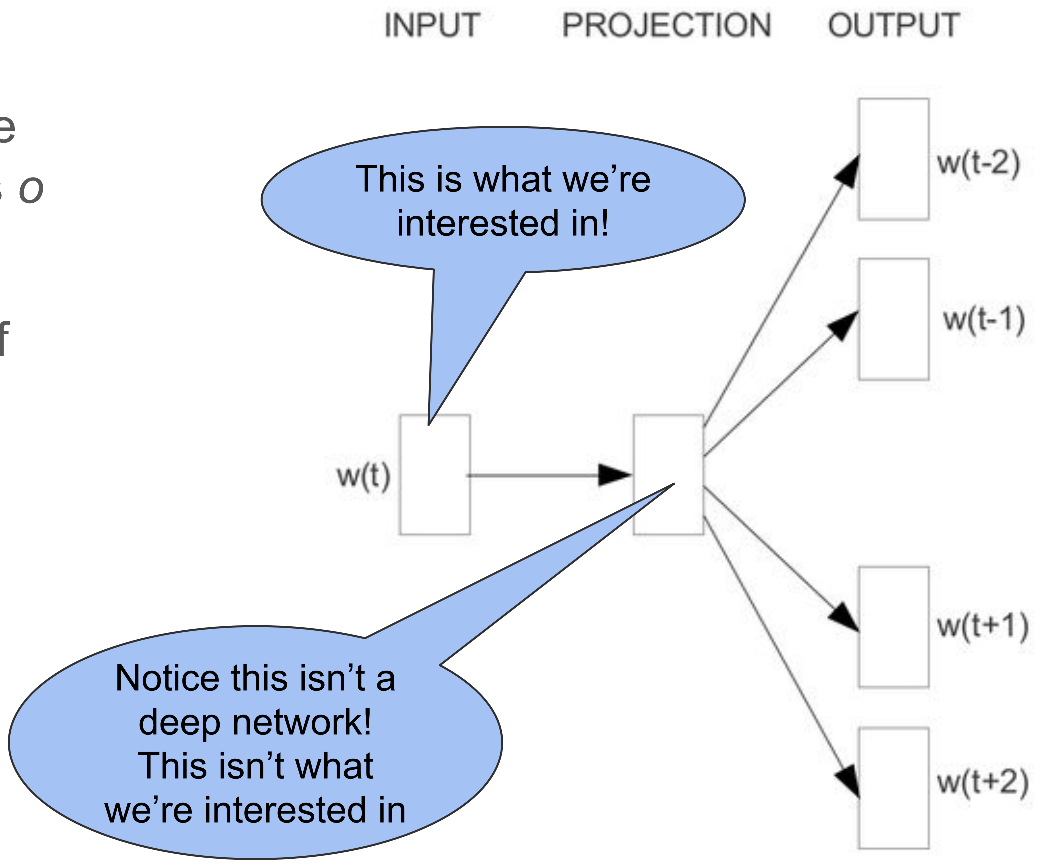
- Start with a large corpus of text
- Each word representation starts as a random vector
- Calculate p(o|c) using the word vectors
- Adjust word vectors to maximize the probabilities



Skip-gram model

The skip-gram model learns the probability of the context words *o* given the center word *c*

- Start with a large corpus of text
- Each word representation starts as a random vector
- Calculate p(o|c) using the word vectors
- Adjust word vectors to maximize the probabilities



Skip-gram model

The skip-gram model learns the probability of the context words *o* given the center word *c*

$p(w_{t-3}|w_t)$ $p(w_{t-2}|w_t)$ $p(w_{t-1}|w_t)$ $p(w_{t+1}|w_t)$ $p(w_{t+2}|w_t)$ $p(w_{t+3}|w_t)$

Harry and Sally sat <u>on the river</u> bank <u>and had a</u> picnic

Skip-gram model

The skip-gram model learns the probability of the context words *o* given the center word *c*

$p(w_{t-3}|w_t)$ $p(w_{t-2}|w_t)$ $p(w_{t-1}|w_t)$ $p(w_{t+1}|w_t)$ $p(w_{t+2}|w_t)$ $p(w_{t+3}|w_t)$

Harry and Sally sat <u>on the river</u> bank <u>and had a</u> picnic

But how do we train the model?

Skip-gram model

The skip-gram model learns the probability of the context words *o* given the center word *c*

For each position in the corpus t = 1, …, T, predict the context with window size *m* given the center word w_j. We train by maximizing a likelihood:

$$L(\theta) =$$

Skip-gram model

The skip-gram model learns the probability of the context words *o* given the center word *c*

For each position in the corpus t = 1, …, T, predict the context with window size *m* given the center word w_j. We train by maximizing a likelihood:

$$L(\theta) = \prod_{t=1}^{T} \prod_{-m \leq j \leq m, j \neq 0} p(w_{t+j} | w_t; \theta)$$

Skip-gram model

The skip-gram model learns the probability of the context words *o* given the center word *c*

For each position in the corpus t = 1, …, T, predict the context with window size *m* given the center word w_j. We train by maximizing a likelihood:

$$L(\theta) = \prod_{t=1}^{T} \prod_{-m \le j \le m, j \neq 0} p(w_{t+j} | w_t; \theta)$$

$$\Rightarrow \theta^\star = \operatorname*{argmin}_{\theta} -\frac{1}{T} \sum_{t=1}^{T} \sum_{-m \le j \le m, j \neq 0} \log p(w_{t+j} | w_t; \theta)$$

Skip-gram model

The skip-gram model learns the probability of the context words *o* given the center word *c*

For each position in the corpus t = 1, …, T, predict the context with window size *m* given the center word w_j. We train by maximizing a likelihood:

$$L(\theta) = \prod_{t=1}^{T} \prod_{-m \leq j \leq m, j \neq 0} p(w_{t+j} | w_t; \theta)$$

What is the probability model?

$$\Rightarrow \theta^\star = \underset{\theta}{\operatorname{argmin}} -\frac{1}{T} \sum_{t=1}^{T} \sum_{-m \leq j \leq m, j \neq 0} \log p(w_{t+j} | w_t; \theta)$$

Skip-gram model

The skip-gram model learns the probability of the context words *o* given the center word *c*

In fact, we use two vectors per word—one for when the word is a center word and one for when the word is a context word—which makes optimization easier

vector representation of *o*

vector representation of *c*

θ are all the u, v vectors for every word in our vocabulary!

$$p(o|c) = \frac{\exp\left(u_o^T v_c\right)}{\sum_{w \in V} \exp\left(u_w^T v_c\right)}$$

Computes **similarity**

sum over vocabulary (softmax regression)

Skip-gram model

The skip-gram model learns the probability of the context words *o* given the center word *c*

In fact, we use two vectors per word—one for when the word is a center word and one for when the word is a context word—which makes optimization easier

vector representation of *o*

vector representation of *c*

θ are all the u, v vectors for every word in our vocabulary!
So, how do we get a single word vector?

$$p(o|c) = \frac{\exp\left(u_o^T v_c\right)}{\sum_{w \in V} \exp\left(u_w^T v_c\right)}$$

Computes **similarity**

sum over vocabulary (softmax regression)

Skip-gram model

The skip-gr ... ability of the context words *o* given the center word *c*

Exercise: Write down the formula for the context word optimization: $\partial/\partial v_c \; p(o|c)$.

In fact, we use two vectors per word—one for when the word is a center word and one for when the word is a context word—which makes optimization easier

θ are all the u, v vectors for every word in our vocabulary!
So, how do we get a single word vector?

vector representation of *o*

vector representation of *c*

$$p(o|c) = \frac{\exp\left(u_o^T v_c\right)}{\sum_{w \in V} \exp\left(u_w^T v_c\right)}$$

Computes **similarity**

sum over vocabulary (softmax regression)

Skip-gram model

Practical considerations:

- The denominator is **expensive to compute when V is large (|V| > 10^5)**. So, we modify the model to perform binary classification so we don't have to sum over all vocabulary words ("what is the probability that this word is a context word for *c*" vs. "what is the probability that this word is right as a context word")

$$p(o \text{ is right word} | c) = \sigma(u_o^T v_c)$$

Only using this term pushes similar words together but doesn't discourage dissimilar words from being far apart—add in examples of *o* words that are not the right word for *c*

$$p(o \text{ is wrong word} | c) = \sigma(-u_o^T v_c)$$

Skip-gram model

Practical considerations:

- The denominator is **expensive to compute when V is large (|V| > 10^5)**. So, we modify the model to perform binary classification so we don't have to sum over all vocabulary words ("what is the probability that this word is a context word for *c*" vs. "what is the probability that this word is right as a context word")

$$p(o \text{ is right word}|c) = \sigma(u_o^T v_c)$$

Only using this term pushes similar words together but doesn't discourage dissimilar words from being far apart—add in examples of *o* words that are not the right word for *c*

$$p(o \text{ is wrong word}|c) = \sigma(-u_o^T v_c)$$

Skip-gram model

Practical considerations:

$$L(\theta) = -\frac{1}{T}\sum_{c,o}\left(\log\sigma(u_o^T v_c) + \sum_w \log\sigma(-u_w^T v_c)\right)$$

- T … So, w … sum o … text word for *c*" vs. "what is the probability that this word is right as a context word")

$$p(o \text{ is right word}|c) = \sigma(u_o^T v_c)$$

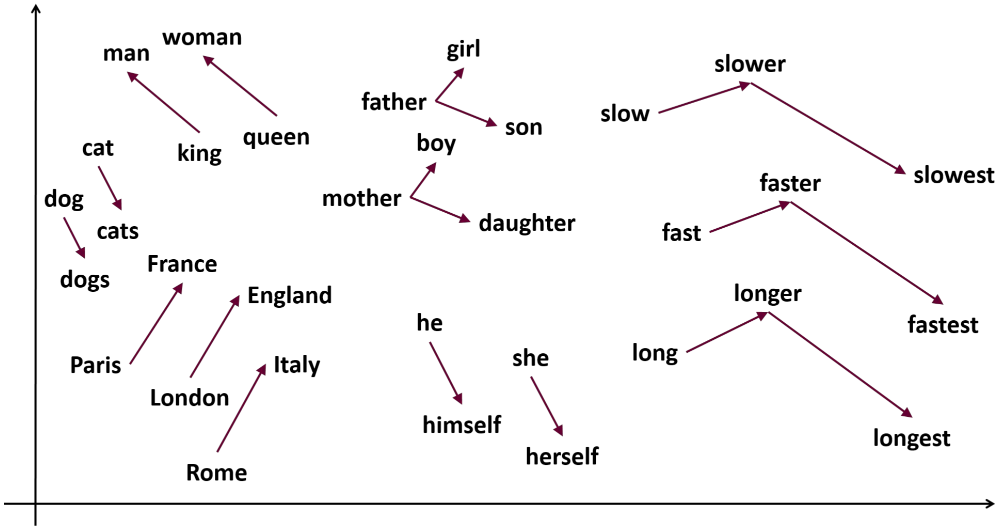
Only using this term pushes similar words together but doesn't discourage dissimilar words from being far apart—add in examples of *o* words that are not the right word for *c*

$$p(o \text{ is wrong word}|c) = \sigma(-u_o^T v_c)$$

Word2vec embeddings

Advantages of word2vec:

- Word vectors staged in a semantically-meaningful space
- Low-dimensional embedding compared to one-hot
- Common-sense operations are interpretable
- Labels come for free (self-supervised)



Disadvantages of word2vec:

- Relative distances alone are meaningful
- Embeddings are static
- Requires lots of data

Take-home message

- Working with language data is challenging
- Embeddings are the most profitable approach to representing individual words as input to a neural model
- We saw static embedding approaches and will later see how embeddings can be determined on the fly