Precept 2: Word Embeddings

COS 484

Sijia Liu (slides borrowed from Tianyu Gao, Spring 2022 and Simon Park, Spring 2025)
2/6/2026

Today’s Plan

1. Overview
2. Predict-based methods (count-based methods skipped)
3. Evaluation
4. Matrix Calculus
5. Exercises
6. Additional topic: Cursor

Overview

Overview - Word Embeddings

- Represent words as vectors
  - e.g., apple -> [0.1, 0.2, 0.5]
  - Encode semantic information
  - Useful for downstream NLP tasks

QUESTION: **how can we get good word vectors**

Overview - Distributional Hypothesis

- Words that occur in similar contexts have similar meaning

- EXAMPLE
  - A is the capital of …
  - B is the capital of …
- A, B should have similar meaning
- Word vectors for A, B should be “similar”

Overview - Different Approaches

- **Count-based** methods: PMI, …
  - Use statistics
  - Covered in great details in lectures, we will skip today

- **Predict-based** methods: word2vec, GloVe, …
  - Use ML

Predict-based Methods

Overview

- Learn an ML model

- Input: **corpus,** dictionary **V**, and desired dimension **d**
- Output: learned model parameters
  - **embedding vector** of dimension d for each word

Word2vec / skip-gram

- Learn an ML model

- Input: corpus, dictionary V, and desired dimension d
- Output: learned model parameters
  - **two embedding vectors** of dimension d for each word
    - **u** when the word is a target word
    - **v** when the word is a context word

Word2vec / skip-gram

- Learn an ML model

- Input: corpus, dictionary V, and desired dimension d
- Output: learned model parameters
  - **two embedding vectors** of dimension d for each word
    - **u** when the word is a target word
    - **v** when the word is a context word
  - **Quick poll: How many model parameters total?**

Word2vec / skip-gram

- Learn an ML model

- Input: corpus, dictionary V, and desired dimension d
- Output: learned model parameters
  - **two embedding vectors** of dimension d for each word
    - **u** when the word is a target word
    - **v** when the word is a context word
  - **Quick poll: How many model parameters total?** $2d\left|V\right|$

Word2vec / skip-gram: Learning Objective Part 1

- Input: target word **t**, context word **c**
- Current model parameters: **u, v** for each word in dictionary
- Model's behavior:
  - Predict $\mathbb{P}[\mathbf{c} \mid \mathbf{t}]$
  - Probability that **c** (out of all possible words) is in context of **t**
  - How? Compute logits $\mathbf{u_t} \cdot \mathbf{v_{c'}}$ for all possible context words
  - Normalize with softmax function
  - $$\mathbb{P}[\mathbf{c} \mid \mathbf{t}] = \frac{\exp(\mathbf{u_t} \cdot \mathbf{v_c})}{\sum_{\mathbf{c'}} \exp(\mathbf{u_t} \cdot \mathbf{v_{c'}})}$$

Word2vec / skip-gram: Learning Objective Part 1

- Input: target word **t**, context word **c**
- Current model parameters: **u, v** for each word in dictionary
- Model's behavior:
  - $$\mathbb{P}[\mathbf{c} \mid \mathbf{t}] = \frac{\exp(\mathbf{u_t} \cdot \mathbf{v_c})}{\sum_{\mathbf{c}'} \exp(\mathbf{u_t} \cdot \mathbf{v_{c'}})}$$
  - Loss = cross-entropy / negative log likelihood
  - $$L_{\mathbf{t},\mathbf{c}} = -\log \mathbb{P}[\mathbf{c} \mid \mathbf{t}]$$

Word2vec / skip-gram: Learning Objective Part 2

- Input: a sequence of words $w_1, w_2, \cdots, w_T$
- Model's behavior:
  - Choose a context window size m
  - Choose a word w_t and consider it a target word
  - For each word $w_{t-m}, \cdots, w_{t-1}, w_{t+1}, \cdots, w_{t+m}$ consider it a context word
  - Compute loss $L_{w_t, w_{t+j}}$ for all $-m \leq j \leq m, j \neq 0$
  - Sum it all up $L_t = \sum_{-m \leq j \leq m, j \neq 0} L_{w_t, w_{t+j}}$
  - Model's loss for predicting context words of a particular word w_t

Word2vec / skip-gram: Learning Objective Part 3

- Input: a sequence of words $w_1, w_2, \cdots, w_T$
- Model's behavior:
  - $L_t = \sum_{-m \leq j \leq m, j \neq 0} L_{w_t, w_{t+j}}$
    - Model's loss for predicting context words of a particular word w_t
  - Take the average to get the final loss $L = \frac{1}{T} \sum_{t=1}^{T} L_t$

Word2vec / skip-gram: Learning Objective Part 3

- Input: a sequence of words $w_1, w_2, \cdots, w_T$
- Model's behavior:
- $$L = -\frac{1}{T}\sum_{t=1}^{T}\sum_{-m\leq j\leq m, j\neq 0} \log \frac{\exp(\mathbf{u}_{w_t} \cdot \mathbf{v}_{w_{t+j}})}{\sum_{\mathbf{c}\in V}\exp(\mathbf{u}_{w_t} \cdot \mathbf{v}_{\mathbf{c}})}$$
  - Note $\mathbf{c} \in V$ takes the sum over all possible words in the dictionary
  - Not just the words that appear in the corpus / sequence of words

Word2vec / skip-gram: How to Train

- Input: a sequence of words $w_1, w_2, \cdots, w_T$
- Compute loss:
  - $$L = -\frac{1}{T}\sum_{t=1}^{T}\sum_{-m\leq j\leq m, j\neq 0}\log\frac{\exp(\mathbf{u}_{w_t}\cdot\mathbf{v}_{w_{t+j}})}{\sum_{\mathbf{c}\in V}\exp(\mathbf{u}_{w_t}\cdot\mathbf{v}_{\mathbf{c}})}$$
- Compute gradient of the loss with respect to all $\mathbf{u}, \mathbf{v}$: $\frac{\partial L}{\partial \mathbf{u}}, \frac{\partial L}{\partial \mathbf{v}}$
- Update model parameters via Gradient Descent
- Problem?

Word2vec / skip-gram: How to Train

- Input: a sequence of words $w_1, w_2, \cdots, w_T$
- Compute loss:
- $$L = -\frac{1}{T}\sum_{t=1}^{T}\sum_{-m\leq j\leq m, j\neq 0} \log \frac{\exp(\mathbf{u}_{w_t} \cdot \mathbf{v}_{w_{t+j}})}{\sum_{\mathbf{c}\in V}\exp(\mathbf{u}_{w_t} \cdot \mathbf{v}_{\mathbf{c}})}$$
- Compute gradient of the loss with respect to all $\mathbf{u}, \mathbf{v}$: $\frac{\partial L}{\partial \mathbf{u}}, \frac{\partial L}{\partial \mathbf{v}}$
- Update model parameters via Gradient Descent
- Problem? For every pair (**t, c**), you need to update $2d\left|V\right|$ parameters
- ONE SOLUTION: instead of $\mathbf{c} \in V$, only sample K (5-20) alternatives

Word2vec / skip-gram: Learning Objective Part 1 (recap)

- Input: target word **t**, context word **c**
- Current model parameters: **u, v** for each word in dictionary
- Model's behavior:
  - Predict $\mathbb{P}[\mathbf{c} \mid \mathbf{t}]$
  - Probability that **c** (out of all possible words) is in context of **t**
  - How? Compute logits $\mathbf{u_t} \cdot \mathbf{v_{c'}}$ for all possible context words
  - Normalize with softmax function
  - $$\mathbb{P}[\mathbf{c} \mid \mathbf{t}] = \frac{\exp(\mathbf{u_t} \cdot \mathbf{v_c})}{\sum_{\mathbf{c'}} \exp(\mathbf{u_t} \cdot \mathbf{v_{c'}})}$$

Word2vec / skip-gram: Negative Sampling Part 1

- Input: target word **t**, context word **c**, **alternative context words** $\mathbf{c_1}, \cdots, \mathbf{c_K}$
- Current model parameters: **u, v** for each word in dictionary
- Model's behavior:
  - Predict $\mathbb{P}[+|\mathbf{t}, \mathbf{c}] \cdot \prod_{i=1}^{K} \mathbb{P}[-|\mathbf{t}, \mathbf{c_i}]$
  - Probability that **c** is in context of **t AND** $\mathbf{c_1}, \cdots, \mathbf{c_K}$ are not in context of **t**
  - How? Consider these **independent** binary logistic regression
  - $\mathbb{P}[+|\mathbf{t}, \mathbf{c}] = \sigma(\mathbf{u_t} \cdot \mathbf{v_c})$
  - $\mathbb{P}[-|\mathbf{t}, \mathbf{c_i}] = \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})$

Word2vec / skip-gram: Negative Sampling Part 1

- Input: target word **t**, context word **c**, **alternative context words** $\mathbf{c_1}, \cdots, \mathbf{c_K}$
- Current model parameters: **u, v** for each word in dictionary
- Model's behavior:
  - $\mathbb{P}[+|\mathbf{t}, \mathbf{c}] \cdot \prod_{i=1}^{K} \mathbb{P}[-|\mathbf{t}, \mathbf{c_i}] = \sigma(\mathbf{u_t} \cdot \mathbf{v_c}) \cdot \prod_{i=1}^{K} \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})$
  - Loss = cross-entropy / negative log likelihood
  - $L_{\mathbf{t},\mathbf{c}} = -\log(\sigma(\mathbf{u_t} \cdot \mathbf{v_c})) - \sum_{i=1}^{K} \log(\sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}}))$
  - Recall that $\mathbf{c_1}, \cdots, \mathbf{c_K}$ randomly sampled

Word2vec / skip-gram: Negative Sampling Part 1

- Input: target word **t**, context word **c**, **alternative context words** $\mathbf{c_1}, \cdots, \mathbf{c_K}$
- Current model parameters: **u, v** for each word in dictionary
- Model's behavior:
  - $\mathbb{P}[+|\mathbf{t}, \mathbf{c}] \cdot \prod_{i=1}^{K} \mathbb{P}[-|\mathbf{t}, \mathbf{c_i}] = \sigma(\mathbf{u_t} \cdot \mathbf{v_c}) \cdot \prod_{i=1}^{K} \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})$
  - Loss = cross-entropy / negative log likelihood
  - $L_{\mathbf{t},\mathbf{c}} = -\log(\sigma(\mathbf{u_t} \cdot \mathbf{v_c})) - \sum_{i=1}^{K} \mathop{\mathbb{E}}_{\mathbf{c_i} \sim V} \log(\sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}}))$
  - **Q: why do we need to take the expectation?**

Word2vec / skip-gram: Negative Sampling Part 1

- Input: target word **t**, context word **c**, **alternative context words** $\mathbf{c_1}, \cdots, \mathbf{c_K}$
- Current model parameters: **u, v** for each word in dictionary
- Model's behavior:
  - $\mathbb{P}[+|\mathbf{t}, \mathbf{c}] \cdot \prod_{i=1}^{K} \mathbb{P}[-|\mathbf{t}, \mathbf{c_i}] = \sigma(\mathbf{u_t} \cdot \mathbf{v_c}) \cdot \prod_{i=1}^{K} \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})$
  - Loss = cross-entropy / negative log likelihood
  - $L_{\mathbf{t},\mathbf{c}} = -\log(\sigma(\mathbf{u_t} \cdot \mathbf{v_c})) - \sum_{i=1}^{K} \mathop{\mathbb{E}}_{\mathbf{c_i} \sim V} \log(\sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}}))$
  - **Q: why do we need to take the expectation?**
  - $\mathbf{c_1}, \cdots, \mathbf{c_K}$ are random variables so the second term is a Monte-Carlo estimate using K alternative samples rather than the ground truth negative set.
    - In practice, you can just compute with K random samples without bootstrapping

Word2vec / skip-gram: Negative Sampling Part 1

- Input: target word **t**, context word **c**, **alternative context words** $\mathbf{c_1}, \cdots, \mathbf{c_K}$
- Current model parameters: **u, v** for each word in dictionary
- Model's behavior:
  - $\mathbb{P}[+|\mathbf{t}, \mathbf{c}] \cdot \prod_{i=1}^{K} \mathbb{P}[-|\mathbf{t}, \mathbf{c_i}] = \sigma(\mathbf{u_t} \cdot \mathbf{v_c}) \cdot \prod_{i=1}^{K} \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})$
  - Loss = cross-entropy / negative log likelihood
  - $L_{\mathbf{t},\mathbf{c}} = -\log(\sigma(\mathbf{u_t} \cdot \mathbf{v_c})) - \sum_{i=1}^{K} \mathop{\mathbb{E}}_{\mathbf{c_i} \sim V} \log(\sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}}))$
  - **Quick Poll: How many parameters contribute to this loss value?**

Word2vec / skip-gram: Negative Sampling Part 1

- Input: target word **t**, context word **c**, **alternative context words** $\mathbf{c_1}, \cdots, \mathbf{c_K}$
- Current model parameters: **u, v** for each word in dictionary
- Model's behavior:
  - $\mathbb{P}[+|\mathbf{t}, \mathbf{c}] \cdot \prod_{i=1}^{K} \mathbb{P}[-|\mathbf{t}, \mathbf{c_i}] = \sigma(\mathbf{u_t} \cdot \mathbf{v_c}) \cdot \prod_{i=1}^{K} \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})$
  - Loss = cross-entropy / negative log likelihood
  - $L_{\mathbf{t},\mathbf{c}} = -\log \sigma(\mathbf{u_t} \cdot \mathbf{v_c}) - \sum_{i=1}^{K} \mathop{\mathbb{E}}_{\mathbf{c_i} \sim V} \log \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})$
  - **Quick Poll: How many parameters contribute to this loss value?**
  - $(K+2)d$ (d for each of $\mathbf{u_t}, \mathbf{v_c}, \mathbf{v_{c_1}}, \mathbf{v_{c_2}}, \cdots \mathbf{v_{c_K}}$) (much more efficient compared to $2d\left|V\right|$ w/o negative sampling)

Word2vec / skip-gram: Train with Negative Sampling

- Input: a sequence of words $w_1, w_2, \cdots, w_T$
- Compute loss:
  - $$L = \frac{1}{T} \sum_{t=1}^{T} \sum_{-m \leq j \leq m, j \neq 0} L_{w_t, w_{t+j}}$$
- Compute gradient of the loss with respect to all $\mathbf{u}, \mathbf{v}$: $\frac{\partial L}{\partial \mathbf{u}}, \frac{\partial L}{\partial \mathbf{v}}$
- Update model parameters via Gradient Descent

Evaluation

Overview

- Recall earlier QUESTION: **how can we get good word vectors**
- How do we know if we got **good** word vectors?

- Intrinsic Evaluation
  - evaluate word vectors directly
  - “similarity” based tasks

- Extrinsic Evaluation:
  - evaluate ML model built on top of the word vectors
  - other tasks

Matrix Calculus

Matrix Calculus / Vectorized Gradients

- Go through this note:
- http://web.stanford.edu/class/cs224n/readings/gradient-notes.pdf
- Make sure that you can understand all the cases in Section 2, 3

Basic Definition

$$f : \mathbb{R}^n \longrightarrow \mathbb{R}^m$$

$$f(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_m(\mathbf{x}))$$

$$\frac{\partial f}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

Gradient w.r.t vector - matrix * vector

Can directly take gradients of a vector with respect to a vector

$\mathbf{x} \in \mathbb{R}^n \quad \mathbf{z} \in \mathbb{R}^m \quad \mathbf{W} \in \mathbb{R}^{m \times n}$

$\mathbf{z} = \mathbf{W}\mathbf{x}$

$$\frac{\partial \mathbf{z}}{\partial \mathbf{x}} = \mathbf{W}$$

Gradient w.r.t vector - vector * matrix

Can directly take gradients of a vector with respect to a vector

$\mathbf{x} \in \mathbb{R}^n \quad \mathbf{z} \in \mathbb{R}^m \quad \mathbf{W} \in \mathbb{R}^{n \times m}$

$\mathbf{z} = \mathbf{x}\mathbf{W}$

$$\frac{\partial \mathbf{z}}{\partial \mathbf{x}} = \mathbf{W}^T$$

Gradient w.r.t vector - elementwise operation

Can directly take gradients of a vector with respect to a vector

$$\mathbf{x} \in \mathbb{R}^n \quad \mathbf{z} \in \mathbb{R}^n \quad \mathbf{f} \in \mathbb{R}^n \longrightarrow \mathbb{R}^n$$

$$\mathbf{z} = f(\mathbf{x}) = (f(x_1), \cdots, f(x_n))$$

$$\frac{\partial \mathbf{z}}{\partial \mathbf{x}} = diag(f'(\mathbf{x})) = \begin{bmatrix} f'(x_1) & 0 & \cdots & 0 \\ 0 & f'(x_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & f'(x_n) \end{bmatrix}$$

Gradient w.r.t matrix - matrix * vector

Cannot directly take gradients of a vector with respect to a matrix

Need to take gradient of the final loss value (scalar) with respect to matrix

$\mathbf{x} \in \mathbb{R}^n \quad \mathbf{z} \in \mathbb{R}^m \quad \mathbf{W} \in \mathbb{R}^{m \times n} \quad L \in \mathbb{R}$

$\mathbf{z} = \mathbf{W}\mathbf{x} \quad L = f(\mathbf{z})$

$$\frac{\partial L}{\partial \mathbf{W}} = \left(\frac{\partial L}{\partial \mathbf{z}}\right)^T \mathbf{x}^T$$

Gradient w.r.t matrix - vector * matrix

Cannot directly take gradients of a vector with respect to a matrix

Need to take gradient of the final loss value (scalar) with respect to matrix

$\mathbf{x} \in \mathbb{R}^n \quad \mathbf{z} \in \mathbb{R}^m \quad \mathbf{W} \in \mathbb{R}^{n \times m} \quad L \in \mathbb{R}$

$\mathbf{z} = \mathbf{x}\mathbf{W} \quad L = f(\mathbf{z})$

$$\frac{\partial L}{\partial \mathbf{W}} = \mathbf{x}^T \left(\frac{\partial L}{\partial \mathbf{z}} \right)$$

Exercises

Q: Gradients for Skip-gram with Negative Sampling

Recall: loss for target **t**, context **c**, **alternative context $\mathbf{c_1}, \cdots, \mathbf{c_K}$**

$$L_{\mathbf{t},\mathbf{c}} = -\log \sigma(\mathbf{u_t} \cdot \mathbf{v_c}) - \sum_{i=1}^{K} \log \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})$$

(a) $\dfrac{\partial L_{\mathbf{t},\mathbf{c}}}{\partial \mathbf{u_t}} = ?$

(b) $\dfrac{\partial L_{\mathbf{t},\mathbf{c}}}{\partial \mathbf{v_c}} = ?$

(c) $\dfrac{\partial L_{\mathbf{t},\mathbf{c}}}{\partial \mathbf{v_{c_j}}} = ?$

A particular index j (not the same index i in the sum)

Q: (a)

$$L_{\mathbf{t},\mathbf{c}} = -\log \sigma(\mathbf{u_t} \cdot \mathbf{v_c}) - \sum_{i=1}^{K} \log \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})$$

$$\frac{\partial L_{\mathbf{t},\mathbf{c}}}{\partial \mathbf{u_t}} =$$

Q: (a)

$$L_{\mathbf{t},\mathbf{c}} = -\log \sigma(\mathbf{u_t} \cdot \mathbf{v_c}) - \sum_{i=1}^{K} \log \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})$$

$$\frac{\partial L_{\mathbf{t},\mathbf{c}}}{\partial \mathbf{u_t}} = -\frac{\frac{\partial \sigma(\mathbf{u_t} \cdot \mathbf{v_c})}{\partial \mathbf{u_t}}}{\sigma(\mathbf{u_t} \cdot \mathbf{v_c})} - \sum_{i=1}^{K} \frac{\frac{\partial \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})}{\partial \mathbf{u_t}}}{\sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})}$$

1. d log(x) / dx = 1/x
2. chain rule

Q: (a)

$$L_{\mathbf{t},\mathbf{c}} = -\log \sigma(\mathbf{u_t} \cdot \mathbf{v_c}) - \sum_{i=1}^{K} \log \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})$$

$$\frac{\partial L_{\mathbf{t},\mathbf{c}}}{\partial \mathbf{u_t}} = -\frac{\frac{\partial \sigma(\mathbf{u_t} \cdot \mathbf{v_c})}{\partial \mathbf{u_t}}}{\sigma(\mathbf{u_t} \cdot \mathbf{v_c})} - \sum_{i=1}^{K} \frac{\frac{\partial \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})}{\partial \mathbf{u_t}}}{\sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})}$$

1. d sigmoid(x) / dx = sig(x) * (1 - sig(x))
2. chain rule

$$= -\frac{\sigma(\mathbf{u_t} \cdot \mathbf{v_c})(1 - \sigma(\mathbf{u_t} \cdot \mathbf{v_c}))\frac{\partial(\mathbf{u_t} \cdot \mathbf{v_c})}{\partial \mathbf{u_t}}}{\sigma(\mathbf{u_t} \cdot \mathbf{v_c})} - \sum_{i=1}^{K} \frac{\sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})(1 - \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}}))\frac{\partial(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})}{\partial \mathbf{u_t}}}{\sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})}$$

Q: (a)

$$L_{\mathbf{t},\mathbf{c}} = -\log \sigma(\mathbf{u_t} \cdot \mathbf{v_c}) - \sum_{i=1}^{K} \log \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})$$

$$\frac{\partial L_{\mathbf{t},\mathbf{c}}}{\partial \mathbf{u_t}} = -\frac{\frac{\partial \sigma(\mathbf{u_t} \cdot \mathbf{v_c})}{\partial \mathbf{u_t}}}{\sigma(\mathbf{u_t} \cdot \mathbf{v_c})} - \sum_{i=1}^{K} \frac{\frac{\partial \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})}{\partial \mathbf{u_t}}}{\sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})}$$

d (x * y) / dx = y

$$= -\frac{\sigma(\mathbf{u_t} \cdot \mathbf{v_c})(1 - \sigma(\mathbf{u_t} \cdot \mathbf{v_c}))\frac{\partial(\mathbf{u_t} \cdot \mathbf{v_c})}{\partial \mathbf{u_t}}}{\sigma(\mathbf{u_t} \cdot \mathbf{v_c})} - \sum_{i=1}^{K} \frac{\sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})(1 - \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}}))\frac{\partial(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})}{\partial \mathbf{u_t}}}{\sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})}$$

$$= -\frac{\sigma(\mathbf{u_t} \cdot \mathbf{v_c})(1 - \sigma(\mathbf{u_t} \cdot \mathbf{v_c}))\mathbf{v_c}}{\sigma(\mathbf{u_t} \cdot \mathbf{v_c})} - \sum_{i=1}^{K} \frac{\sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})(1 - \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}}))(-\mathbf{v_{c_i}})}{\sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})}$$

Q: (a)

$$L_{\mathbf{t},\mathbf{c}} = -\log \sigma(\mathbf{u_t} \cdot \mathbf{v_c}) - \sum_{i=1}^{K} \log \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})$$

$$\frac{\partial L_{\mathbf{t},\mathbf{c}}}{\partial \mathbf{u_t}} = -\frac{\frac{\partial \sigma(\mathbf{u_t} \cdot \mathbf{v_c})}{\partial \mathbf{u_t}}}{\sigma(\mathbf{u_t} \cdot \mathbf{v_c})} - \sum_{i=1}^{K} \frac{\frac{\partial \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})}{\partial \mathbf{u_t}}}{\sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})}$$

$$= -\frac{\sigma(\mathbf{u_t} \cdot \mathbf{v_c})(1 - \sigma(\mathbf{u_t} \cdot \mathbf{v_c}))\frac{\partial(\mathbf{u_t} \cdot \mathbf{v_c})}{\partial \mathbf{u_t}}}{\sigma(\mathbf{u_t} \cdot \mathbf{v_c})} - \sum_{i=1}^{K} \frac{\sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})(1 - \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}}))\frac{\partial(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})}{\partial \mathbf{u_t}}}{\sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})}$$

$$= -\frac{\sigma(\mathbf{u_t} \cdot \mathbf{v_c})(1 - \sigma(\mathbf{u_t} \cdot \mathbf{v_c}))\mathbf{v_c}}{\sigma(\mathbf{u_t} \cdot \mathbf{v_c})} - \sum_{i=1}^{K} \frac{\sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})(1 - \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}}))(-\mathbf{v_{c_i}})}{\sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})}$$

Q: (a)

$$L_{\mathbf{t},\mathbf{c}} = -\log \sigma(\mathbf{u_t} \cdot \mathbf{v_c}) - \sum_{i=1}^{K} \log \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})$$

$$\frac{\partial L_{\mathbf{t},\mathbf{c}}}{\partial \mathbf{u_t}} = -\frac{\frac{\partial \sigma(\mathbf{u_t} \cdot \mathbf{v_c})}{\partial \mathbf{u_t}}}{\sigma(\mathbf{u_t} \cdot \mathbf{v_c})} - \sum_{i=1}^{K} \frac{\frac{\partial \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})}{\partial \mathbf{u_t}}}{\sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})}$$

$$= -\frac{\sigma(\mathbf{u_t} \cdot \mathbf{v_c})(1 - \sigma(\mathbf{u_t} \cdot \mathbf{v_c}))\frac{\partial(\mathbf{u_t} \cdot \mathbf{v_c})}{\partial \mathbf{u_t}}}{\sigma(\mathbf{u_t} \cdot \mathbf{v_c})} - \sum_{i=1}^{K} \frac{\sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})(1 - \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}}))\frac{\partial(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})}{\partial \mathbf{u_t}}}{\sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})}$$

$$= -(1 - \sigma(\mathbf{u_t} \cdot \mathbf{v_c}))\mathbf{v_c} - \sum_{i=1}^{K} (1 - \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}}))(-\mathbf{v_{c_i}})$$

Q: (a)

$$L_{\mathbf{t},\mathbf{c}} = -\log \sigma(\mathbf{u_t} \cdot \mathbf{v_c}) - \sum_{i=1}^{K} \log \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})$$

$$\frac{\partial L_{\mathbf{t},\mathbf{c}}}{\partial \mathbf{u_t}} = -\frac{\frac{\partial \sigma(\mathbf{u_t} \cdot \mathbf{v_c})}{\partial \mathbf{u_t}}}{\sigma(\mathbf{u_t} \cdot \mathbf{v_c})} - \sum_{i=1}^{K} \frac{\frac{\partial \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})}{\partial \mathbf{u_t}}}{\sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})}$$

$$= -\frac{\sigma(\mathbf{u_t} \cdot \mathbf{v_c})(1 - \sigma(\mathbf{u_t} \cdot \mathbf{v_c}))\frac{\partial(\mathbf{u_t} \cdot \mathbf{v_c})}{\partial \mathbf{u_t}}}{\sigma(\mathbf{u_t} \cdot \mathbf{v_c})} - \sum_{i=1}^{K} \frac{\sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})(1 - \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}}))\frac{\partial(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})}{\partial \mathbf{u_t}}}{\sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})}$$

$$= -(1 - \sigma(\mathbf{u_t} \cdot \mathbf{v_c}))\mathbf{v_c} - \sum_{i=1}^{K} (1 - \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}}))(-\mathbf{v_{c_i}})$$

$$= (\sigma(\mathbf{u_t} \cdot \mathbf{v_c}) - 1)\mathbf{v_c} + \sum_{i=1}^{K} \sigma(\mathbf{u_t} \cdot \mathbf{v_{c_i}})\mathbf{v_{c_i}}$$

sigmoid(-x) = 1 - sigmoid(x)

Q: (a) FINAL VERSION

$$\frac{\partial L_{\mathbf{t,c}}}{\partial \mathbf{u_t}} = (\sigma(\mathbf{u_t} \cdot \mathbf{v_c}) - 1)\mathbf{v_c} + \sum_{i=1}^{K} \sigma(\mathbf{u_t} \cdot \mathbf{v_{c_i}})\mathbf{v_{c_i}}$$

Q: (b)

$$L_{\mathbf{t},\mathbf{c}} = -\log \sigma(\mathbf{u_t} \cdot \mathbf{v_c}) - \sum_{i=1}^{K} \log \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})$$

$$\frac{\partial L_{\mathbf{t},\mathbf{c}}}{\partial \mathbf{v_c}} =$$

Q: (b)

$$L_{\mathbf{t},\mathbf{c}} = -\log \sigma(\mathbf{u_t} \cdot \mathbf{v_c}) - \sum_{i=1}^{K} \log \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})$$

$$\frac{\partial L_{\mathbf{t},\mathbf{c}}}{\partial \mathbf{v_c}} = -\frac{\frac{\partial \sigma(\mathbf{u_t} \cdot \mathbf{v_c})}{\partial \mathbf{v_c}}}{\sigma(\mathbf{u_t} \cdot \mathbf{v_c})}$$

Summands do not depend on v_c

Q: (b)

$$L_{\mathbf{t},\mathbf{c}} = -\log \sigma(\mathbf{u_t} \cdot \mathbf{v_c}) - \sum_{i=1}^{K} \log \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})$$

$$\frac{\partial L_{\mathbf{t},\mathbf{c}}}{\partial \mathbf{v_c}} = -\frac{\frac{\partial \sigma(\mathbf{u_t} \cdot \mathbf{v_c})}{\partial \mathbf{v_c}}}{\sigma(\mathbf{u_t} \cdot \mathbf{v_c})}$$

$$= -(1 - \sigma(\mathbf{u_t} \cdot \mathbf{v_c}))\mathbf{u_t}$$

Same as before

$$= (\sigma(\mathbf{u_t} \cdot \mathbf{v_c}) - 1)\mathbf{u_t}$$

Q: (b) FINAL VERSION

$$\frac{\partial L_{\mathbf{t},\mathbf{c}}}{\partial \mathbf{v_c}} = (\sigma(\mathbf{u_t} \cdot \mathbf{v_c}) - 1)\mathbf{u_t}$$

Q: (c)

$$L_{\mathbf{t},\mathbf{c}} = -\log \sigma(\mathbf{u_t} \cdot \mathbf{v_c}) - \sum_{i=1}^{K} \log \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})$$

$$\frac{\partial L_{\mathbf{t},\mathbf{c}}}{\partial \mathbf{v_{c_j}}} =$$

Q: (c)

$$L_{\mathbf{t},\mathbf{c}} = -\log \sigma(\mathbf{u_t} \cdot \mathbf{v_c}) - \sum_{i=1}^{K} \log \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})$$

$$\frac{\partial L_{\mathbf{t},\mathbf{c}}}{\partial \mathbf{v_{c_j}}} = -\frac{\frac{\partial \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_j}})}{\partial \mathbf{v_{c_j}}}}{\sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_j}})}$$

All other summands do not depend on j

Q: (c)

$$L_{\mathbf{t},\mathbf{c}} = -\log \sigma(\mathbf{u_t} \cdot \mathbf{v_c}) - \sum_{i=1}^{K} \log \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})$$

$$\frac{\partial L_{\mathbf{t},\mathbf{c}}}{\partial \mathbf{v_{c_j}}} = -\frac{\frac{\partial \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_j}})}{\partial \mathbf{v_{c_j}}}}{\sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_j}})}$$

$$= -(1 - \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_j}}))(-\mathbf{u_t})$$

Same as before

$$= \sigma(\mathbf{u_t} \cdot \mathbf{v_{c_j}})\mathbf{u_t}$$

Q: (c) FINAL VERSION

$$\frac{\partial L_{\mathbf{t},\mathbf{c}}}{\partial \mathbf{v}_{\mathbf{c}_j}} = \sigma(\mathbf{u}_\mathbf{t} \cdot \mathbf{v}_{\mathbf{c}_j})\mathbf{u}_\mathbf{t}$$

Q: FINAL VERSION

Recall: loss for target **t**, context **c**, **alternative context $\mathbf{c_1}, \cdots, \mathbf{c_K}$**

$$L_{\mathbf{t},\mathbf{c}} = -\log \sigma(\mathbf{u_t} \cdot \mathbf{v_c}) - \sum_{i=1}^{K} \log \sigma(-\mathbf{u_t} \cdot \mathbf{v_{c_i}})$$

(a) $$\frac{\partial L_{\mathbf{t},\mathbf{c}}}{\partial \mathbf{u_t}} = (\sigma(\mathbf{u_t} \cdot \mathbf{v_c}) - 1)\mathbf{v_c} + \sum_{i=1}^{K} \sigma(\mathbf{u_t} \cdot \mathbf{v_{c_i}})\mathbf{v_{c_i}}$$

(b) $$\frac{\partial L_{\mathbf{t},\mathbf{c}}}{\partial \mathbf{v_c}} = (\sigma(\mathbf{u_t} \cdot \mathbf{v_c}) - 1)\mathbf{u_t}$$

(c) $$\frac{\partial L_{\mathbf{t},\mathbf{c}}}{\partial \mathbf{v_{c_j}}} = \sigma(\mathbf{u_t} \cdot \mathbf{v_{c_j}})\mathbf{u_t}$$

Additional topic: Cursor Demo

- What’s cursor?
- A very powerful coding agent!

- Now it’s free for students for 1 year: https://cursor.com/students

Additional topic: Cursor Demo

- **Key concepts:**
  - Tab
  - Agent (cmd + I)
  - Inline Edit (cmd + K)
  - Chat (cmd + N)
  - Rules
  - Semantic search
  - MCP
  - Context
  - Models

Additional topic: Cursor Demo

- **Best practices with cursor agents**
  - **Use plan mode before coding**
  - **Let the agent find the context**
  - **Start with tests and use as feedback loop**
  - **Customize with rules and skills**
    - Tip: Start simple! Add rules only when you notice the agent making the same mistake repeatedly.
  - **Use debug mode for tricky bugs**
- See more in https://cursor.com/blog/agent-best-practices