COS 484

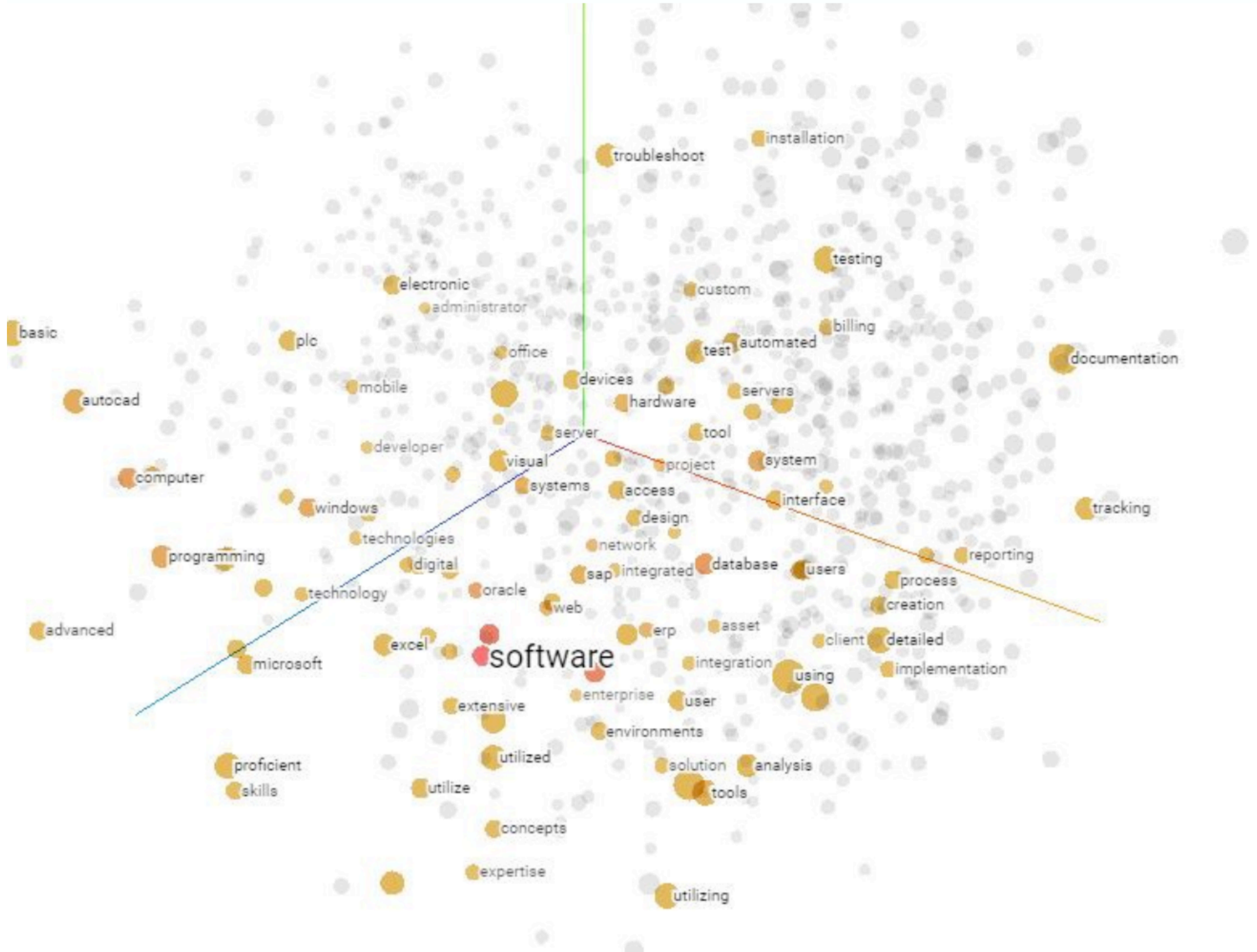
Natural Language Processing

Precept #3

Preceptor: Howard Chen

Word embeddings

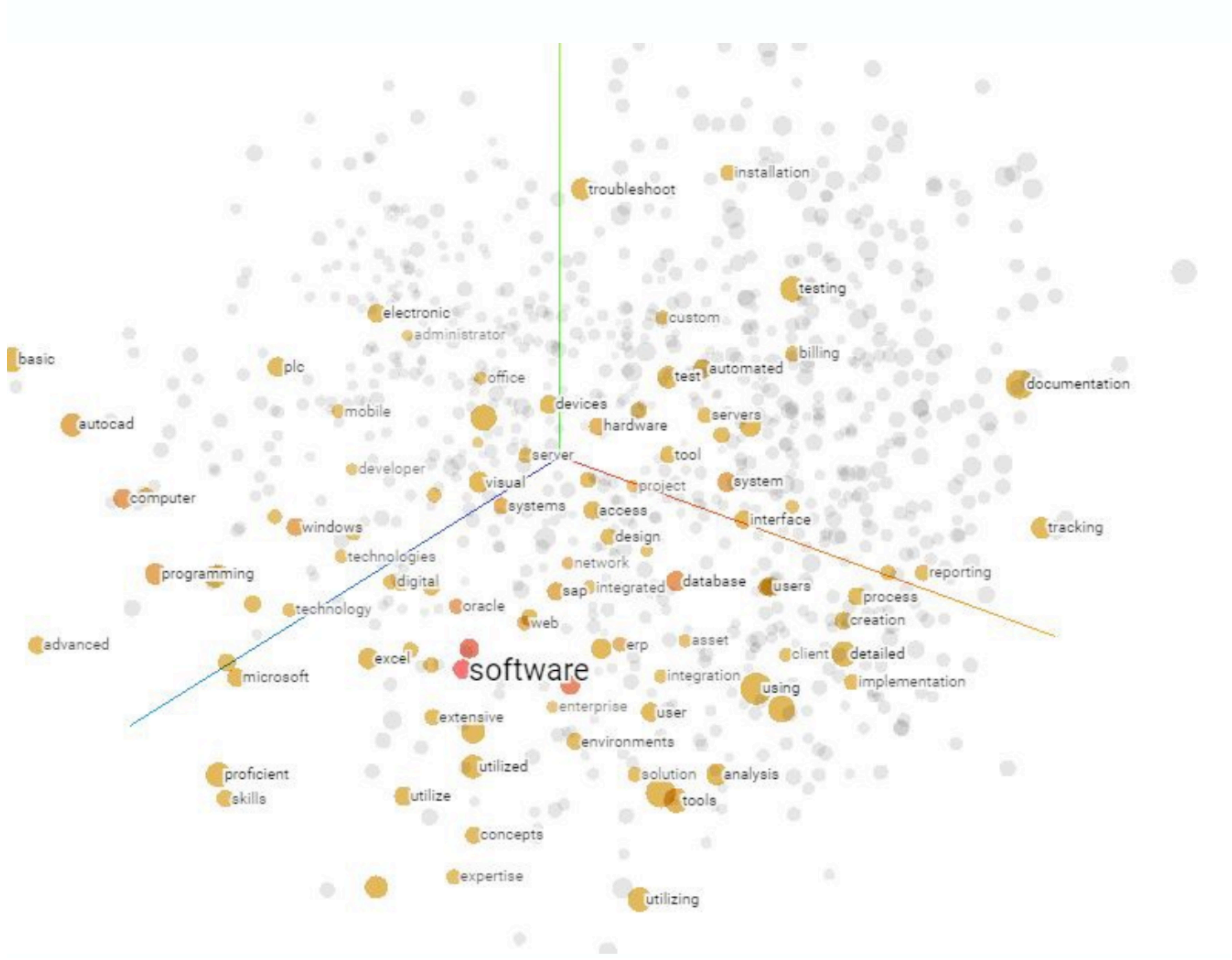
- Represent words as vectors: apple -> [0.1, 0.2, 0.3, 0.5]
  - Encode the semantic information in the word vector
  - Use for downstream NLP tasks



Word embeddings

- Represent words as vectors: apple -> [0.1, 0.2, 0.3, 0.5]
  - Encode the semantic information in the word vector
  - Use for downstream NLP task

  - How can we get high-quality word vectors.



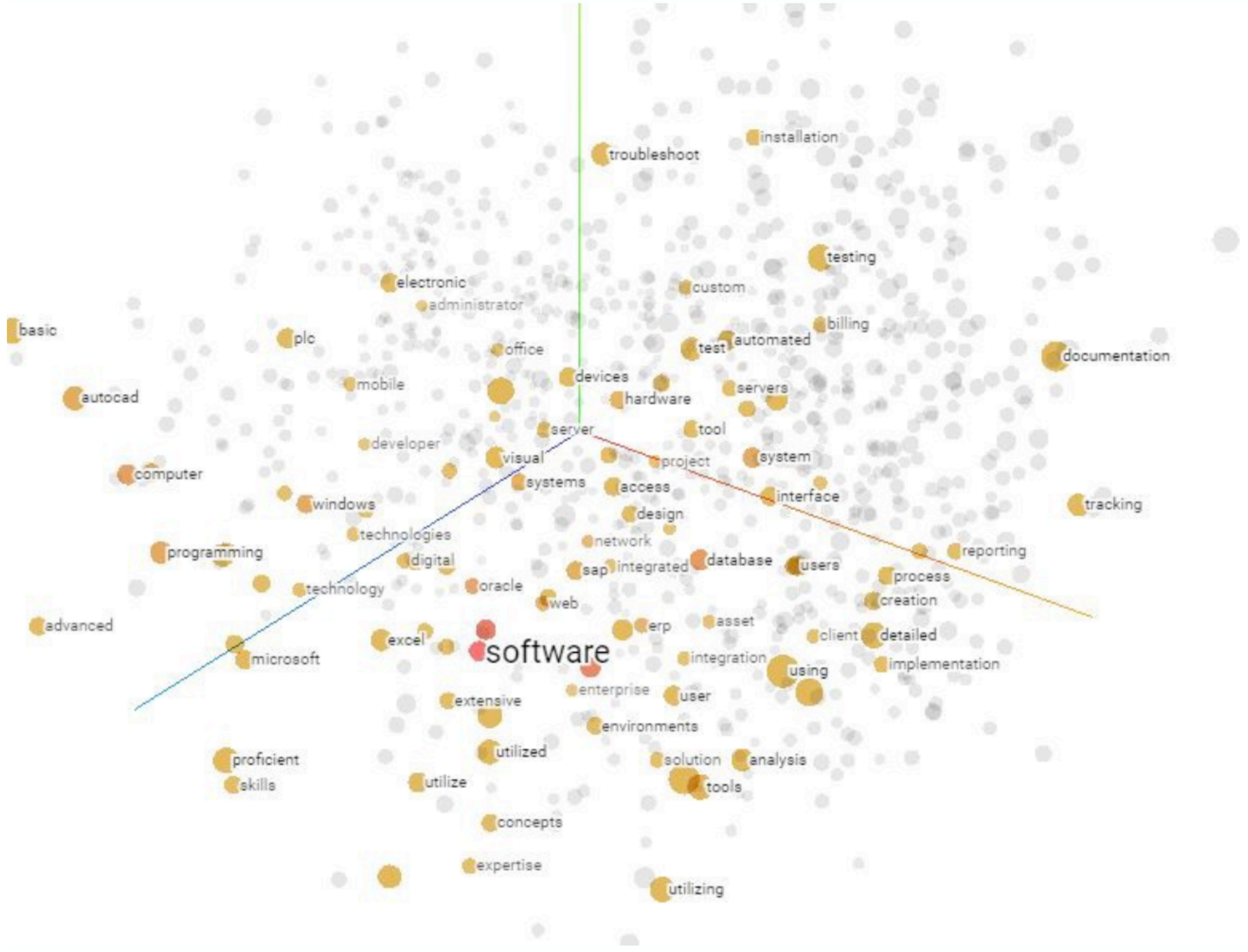
Word embeddings

- Represent words as vectors: apple -> [0.1, 0.2, 0.3, 0.5]
  - Encode the semantic information in the word vector
  - Use for downstream NLP task

- Distributional hypothesis
  - words that occur in similar contexts
    tend to have similar meanings

  - A is the capital of …
  - B is the capital of …

A and B are both the name of capital cities.



Word embeddings

- Distributional hypothesis
  - words that occur in similar contexts tend to have similar meanings
  - How can we get high-quality word vectors following the intuition of distributional hypothesis?

Word embeddings

- Distributional hypothesis
  - words that occur in similar contexts tend to have similar meanings

  - How can we get high-quality word vectors following the intuition of distributional hypothesis?
    - Count-based methods: PMI, PPMI …
    - Predict-based methods: word2vec, GloVe, Fasttext …

Word embeddings

- Distributional hypothesis
  - words that occur in similar contexts tend to have similar meanings

  - How can we get high-quality word vectors following the intuition of distributional hypothesis?
    - Count-based methods: PMI, PPMI … (statistics)
    - Predict-based methods: word2vec, GloVe, Fasttext …

Word embeddings

- Distributional hypothesis
  - words that occur in similar contexts tend to have similar meanings
  - How can we get high-quality word vectors following the intuition of distributional hypothesis?
    - Count-based methods: PMI, PPMI … (statistics)
    - Predict-based methods: word2vec, GloVe, Fasttext … (learning)
      - Task: predict the context word given the target word.

Count-based word vectors

- Word-word co-occurrence matrix W
  - W[t, c] = count(t, c)

(the counts that word c occurs in the context of word t)

context words:
4 words to the left,
4 words to the right

is traditionally followed by	**cherry**	pie, a traditional dessert
often mixed, such as	**strawberry**	rhubarb pie. Apple pie
computer peripherals and personal	**digital**	assistants. These devices usually
a computer. This includes	**information**	available on the internet

	aardvark	...	**computer**	**data**	**result**	**pie**	**sugar**	...
cherry	0	...	2	8	9	442	25	...
strawberry	0	...	0	0	1	60	19	...
digital	0	...	1670	1683	85	5	4	...
information	0	...	3325	3982	378	5	13	...

Most entries are 0s $\Longrightarrow$ sparse vectors

Count-based word vectors

- Word-word co-occurrence matrix W
  - W[t, c] = count(t, c)

    (the counts that word c occurs in the context of word t)

  - Weakness: overly frequent words like “the”, “it", or “they” appear a lot near other words
    - W[the, apple] >> W[apple, pie]

Count-based word vectors

- Pointwise mutual information (PMI)
  - From text, extract a lot of word pairs: (target, context).
  - p(t, c) = the probability that target is t and context is c.
  - p(t) = the probability that target is t.
  - p(c) = the probability that context is.
  - PMI[t, c] = $\log \frac{p(t,c)}{p(t)p(c)}$

Count-based word vectors

- Pointwise mutual information (PMI): PMI[t,c] ∈ (-inf, inf)

  $\log \frac{p(t,c)}{p(t)p(c)}$ = 0

  - p(t, c) = the probability that target is t and context is c.
  - p(t) = the probability that target is t.
  - p(c) = the probability that context is.

Count-based word vectors

- Pointwise mutual information (PMI): PMI[t,c] ∈ (-inf, inf)

  $\log \frac{p(t,c)}{p(t)p(c)} = 0 \Leftrightarrow \frac{p(t,c)}{p(t)p(c)} = 1$

  - p(t, c) = the probability that target is t and context is c.
  - p(t) = the probability that target is t.
  - p(c) = the probability that context is.

Count-based word vectors

- Pointwise mutual information (PMI): PMI[t,c] ∈ (-inf, inf)

$$\log \frac{p(t,c)}{p(t)p(c)} = 0 \Leftrightarrow \frac{p(t,c)}{p(t)p(c)} = 1 \Leftrightarrow \frac{p(t,c)}{p(t)} = p(c)$$

  - p(t, c) = the probability that target is t and context is c.
  - p(t) = the probability that target is t.
  - p(c) = the probability that context is.

Count-based word vectors

- Pointwise mutual information (PMI): PMI[t,c] ∈ (-inf, inf)

$$\log \frac{p(t,c)}{p(t)p(c)} = 0 \Leftrightarrow \frac{p(t,c)}{p(t)p(c)} = 1 \Leftrightarrow \frac{p(t,c)}{p(t)} = p(c) \Leftrightarrow p(c|t) = p(c)$$

  - p(t, c) = the probability that target is t and context is c.
  - p(t) = the probability that target is t.
  - p(c) = the probability that context is.

Count-based word vectors

- Pointwise mutual information (PMI): PMI[t,c] ∈ (-inf, inf)

  $\log \frac{p(t,c)}{p(t)p(c)}$ = 0 ⇔ $\frac{p(t,c)}{p(t)p(c)}$ = 1 ⇔ $\frac{p(t,c)}{p(t)} = p(c)$ ⇔ p(c|t) = p(c)

  ⇔ p(t) and p(c) is independent.

  - p(t, c) = the probability that target is t and context is c.
  - p(t) = the probability that target is t.
  - p(c) = the probability that context is.

Count-based word vectors

- Pointwise mutual information (PMI): PMI[t,c] ∈ (-inf, inf)

  $\log \frac{p(t,c)}{p(t)p(c)}$ = 0 ⇔ $\frac{p(t,c)}{p(t)p(c)}$ = 1 ⇔ $\frac{p(t,c)}{p(t)} = p(c)$ ⇔ p(c|t) = p(c)

  ⇔ p(t) and p(c) is independent.

  ⇔ Knowing that the target is t doesn't affect the probability of context is c.

  - p(t, c) = the probability that target is t and context is c.
  - p(t) = the probability that target is t.
  - p(c) = the probability that context is.

Count-based word vectors

- Pointwise mutual information (PMI): PMI[t,c] ∈ (-inf, inf)

  $\log \frac{p(t,c)}{p(t)p(c)} > 0$

  - p(t, c) = the probability that target is t and context is c.
  - p(t) = the probability that target is t.
  - p(c) = the probability that context is.

Count-based word vectors

- Pointwise mutual information (PMI): PMI[t,c] ∈ (-inf, inf)

$$\log \frac{p(t,c)}{p(t)p(c)} > 0 \Leftrightarrow \frac{p(t,c)}{p(t)p(c)} > 1$$

  - p(t, c) = the probability that target is t and context is c.
  - p(t) = the probability that target is t.
  - p(c) = the probability that context is.

Count-based word vectors

- Pointwise mutual information (PMI): PMI[t,c] ∈ (-inf, inf)

$$\log \frac{p(t,c)}{p(t)p(c)} > 0 \Leftrightarrow \frac{p(t,c)}{p(t)p(c)} > 1 \Leftrightarrow \frac{p(t,c)}{p(t)} > p(c)$$

  - p(t, c) = the probability that target is t and context is c.
  - p(t) = the probability that target is t.
  - p(c) = the probability that context is.

Count-based word vectors

- Pointwise mutual information (PMI): PMI[t,c] ∈ (-inf, inf)

$$\log \frac{p(t,c)}{p(t)p(c)} > 0 \Leftrightarrow \frac{p(t,c)}{p(t)p(c)} > 1 \Leftrightarrow \frac{p(t,c)}{p(t)} > p(c) \Leftrightarrow p(c|t) > p(c)$$

  - p(t, c) = the probability that target is t and context is c.
  - p(t) = the probability that target is t.
  - p(c) = the probability that context is.

Count-based word vectors

- Pointwise mutual information (PMI): PMI[t,c] ∈ (-inf, inf)

  $\log \frac{p(t,c)}{p(t)p(c)} > 0 \Leftrightarrow \frac{p(t,c)}{p(t)p(c)} > 1 \Leftrightarrow \frac{p(t,c)}{p(t)} > p(c) \Leftrightarrow p(c|t) > p(c)$

  $\Leftrightarrow$ Knowing that the target is t, context is more likely to be c.

  - p(t, c) = the probability that target is t and context is c.
  - p(t) = the probability that target is t.
  - p(c) = the probability that context is.

Count-based word vectors

- Pointwise mutual information (PMI): PMI[t,c] ∈ (-inf, inf)

  $\log \frac{p(t,c)}{p(t)p(c)}$ < 0

  - p(t, c) = the probability that target is t and context is c.
  - p(t) = the probability that target is t.
  - p(c) = the probability that context is.

Count-based word vectors

- Pointwise mutual information (PMI): PMI[t,c] ∈ (-inf, inf)

$$\log \frac{p(t,c)}{p(t)p(c)} < 0 \Leftrightarrow \frac{p(t,c)}{p(t)p(c)} < 1$$

  - p(t, c) = the probability that target is t and context is c.
  - p(t) = the probability that target is t.
  - p(c) = the probability that context is.

Count-based word vectors

- Pointwise mutual information (PMI): PMI[t,c] ∈ (-inf, inf)

$$\log \frac{p(t,c)}{p(t)p(c)} < 0 \Leftrightarrow \frac{p(t,c)}{p(t)p(c)} < 1 \Leftrightarrow \frac{p(t,c)}{p(t)} < p(c)$$

  - p(t, c) = the probability that target is t and context is c.
  - p(t) = the probability that target is t.
  - p(c) = the probability that context is.

Count-based word vectors

- Pointwise mutual information (PMI): PMI[t,c] ∈ (-inf, inf)

$$\log \frac{p(t,c)}{p(t)p(c)} < 0 \Leftrightarrow \frac{p(t,c)}{p(t)p(c)} < 1 \Leftrightarrow \frac{p(t,c)}{p(t)} < p(c) \Leftrightarrow p(c|t) < p(c)$$

  - p(t, c) = the probability that target is t and context is c.
  - p(t) = the probability that target is t.
  - p(c) = the probability that context is.

Count-based word vectors

- Pointwise mutual information (PMI): PMI[t,c] ∈ (-inf, inf)

  $\log \frac{p(t,c)}{p(t)p(c)} < 0 \Leftrightarrow \frac{p(t,c)}{p(t)p(c)} < 1 \Leftrightarrow \frac{p(t,c)}{p(t)} < p(c) \Leftrightarrow p(c|t) < p(c)$

  $\Leftrightarrow$ Knowing that the target is t, context is less likely to be c.

  - p(t, c) = the probability that target is t and context is c.
  - p(t) = the probability that target is t.
  - p(c) = the probability that context is.

Count-based word vectors

- Pointwise mutual information (PMI): PMI[t,c] ∈ (-inf, inf)

$$\log \frac{p(t,c)}{p(t)p(c)} < 0 \Leftrightarrow \frac{p(t,c)}{p(t)p(c)} < 1 \Leftrightarrow \frac{p(t,c)}{p(t)} < p(c) \Leftrightarrow p(c|t) < p(c)$$

⇔ Knowing that the target is t, context is less likely to be c.

- p(t, c) = the probability that target is t and context is c.
- p(t) = the probability that target is t.
- p(c) = the probability that context is.

PMI(x, y): Do events *x* and *y* co-occur more or less than if they were independent?

Count-based word vectors

- PMI(x, y): Do events *x* and *y* co-occur more or less than if they were independent?
  - Negative PMI values tend to be unreliable without enormous corpora.
  - Why? (will study this problem in the exercise)

- Positive Pointwise Mutual Information (PPMI)
  - PPMI[t, c] = max(0, PMI[t, c])

Count-based word vectors

- Positive Pointwise Mutual Information (PPMI)
  - PPMI[t, c] = max(0, PMI[t, c])

- PPMI is in the shape of |V|x|V|.
  - The dimensionality is too big. (curse of dimensionality)
  - Dimensionality reduction by SVD
    - Transform into low dimension space but retain meaningful information.

Singular value decomposition

SVD: $A = U\Sigma V^T$

A is an m x n matrix.

U is an m x m orthogonal matrix. $U^T U = I$.

V is an n x n orthogonal matrix. $V^T V = I$.

Σ is an m x n nonnegative diagonal matrix.

The diagonal entries of Σ are called singular values of A.

The columns of U and V are called left/right singular vectors.

Singular value decomposition

SVD: $A = U\Sigma V^T$

A is a m-by-n matrix. PPMI matrix. m target word, n context word

U is a m-by-m orthogonal matrix. $U^T U = I$. Consider rows of U as word vectors.

V is a n-by-n orthogonal matrix. $V^T V = I$. Consider rows of V as context word vectors.

Σ is a m-by-n nonnegative diagonal matrix.

The diagonal entries of Σ are called singular values of A.

The columns of U and V are called left/right singular vectors.

Singular value decomposition

SVD: $A = U\Sigma V^T$

Low-rank matrix approximation:

Find a p-rank matrix B to approximate A based on minimizing $\sum(A[i,j] - B[i,j])^2$.

The solution is $B = U\hat{\Sigma}V^T$, where $\hat{\Sigma}$ is the same as Σ except it contains only the p largest singular values.

Only the p columns of U that have nonzero singular values contribute to B.

We can throw away other columns and the rest m-by-p matrix $\widehat{\mathrm{U}}$ still contain necessary information to restore B.

The row vector of $\widehat{\mathrm{U}}$ is the low-rank word vectors.

word2vec

- Learn word vectors by solving a machine learning task
  - Use the target words to predict their context words (skip-gram).
  - Build a learning objective for this task.
  - Optimize the word vectors to minimize the learning objective.
  - Input: a large text corpora, *V*, *d*
  - Output: $f: word \rightarrow R^d$

Skip-gram

- Learning objective
  - Use the target words to predict their context words: P(c|t)
  - A |V|-way classification problem: |V| potential context word.

Skip-gram

- Learning objective
  - Use the target words to predict their context words: P(c|t)
  - A |V|-way classification problem: |V| potential context word.
  - Get |V| scores, one for each context word.

Skip-gram

- Learning objective
  - Use the target words to predict their context words: P(c|t)
  - A |V|-way classification problem: |V| potential context word.

  - F: input x labels → scores ([0.1, 0.2, 0.3]).
  - G: scores -> prediction ([0, 0, 1]).
  - Minimize the difference between prediction and true labels ([0,1,0]).

Skip-gram

- Learning objective
  - Use the target words to predict their context words: P(c|t)
  - A |V|-way classification problem: |V| potential context word.

  - F: input x labels → scores ([0.1, 0.2, 0.3]).
  - G: scores -> prediction ([0, 0, 1]). Discrete values! Non-differentiable!
  - Minimize the difference between prediction and true labels ([0,1,0]).

Skip-gram

- Learning objective
  - Use the target words to predict their context words: P(c|t)
  - A |V|-way classification problem: |V| potential context word.

  - F: input x labels → scores ([0.1, 0.2, 0.3]).
  - G: scores -> prediction ([0, 0, 1]). Discrete values! Non-differentiable!
  - Minimize the difference between prediction and true labels ([0,1,0]).

  - Continuous approximation: prediction -> probability distribution

Skip-gram

- Learning objective
  - Use the target words to predict their context words
  - A |V|-way classification problem: |V| potential context word.

  - F: input x labels → scores ([0.1, 0.2, 0.3]).
  - G: scores -> prediction ([0, 0, 1]). Discrete values! Non-differentiable!
  - Minimize the difference between prediction and true labels ([0,1,0]).

  - Continuous approximation: prediction -> probability distribution P(c|t)

    G -> softmax function

Skip-gram

- Softmax:
  - Softmax(s1, s2, …, sk) = $[\frac{\exp(s1)}{\sum_j \exp(sj)}, \frac{\exp(s2)}{\sum_j \exp(sj)}, \ldots, \frac{\exp(sk)}{\sum_j \exp(sj)}]$
    - If s1 is the largest one, $\frac{\exp(s1)}{\sum_j \exp(sj)}$ is close to but smaller than 1.
    - Otherwise, $\frac{\exp(s1)}{\sum_j \exp(sj)}$ is close to but larger than 0.
    - The output sums to 1.
    - A perfect continues approximation of argmax function G.

Skip-gram

- Learning objective
  - Use the target words to predict their context words
  - A |V|-way classification problem: |V| potential context word.

  - F: input x labels → score s ([0.1, 0.2, 0.3]).
  - Softmax: score s-> p(c|t)
  - Minimize the difference between p(c|t) and labels
    - cross entropy: $-log \frac{\exp(s_{c|t})}{\sum_a \exp(s_{a|t})}$

Skip-gram

- Learning objective
  - Use the target words to predict their context words
  - A |V|-way classification problem: |V| potential context word.

  - F: input x labels → score s ([0.1, 0.2, 0.3]).
  - Softmax: score s-> p(c|t)
  - Minimize the difference between p(c|t) and labels
    - cross entropy: $-log\frac{\exp(s_{c|t})}{\sum_a \exp(s_{a|t})}$

$$F(t,c) = s_{c|t} = \boldsymbol{u}_t \cdot \boldsymbol{v}_c$$

Skip-gram

- Learning objective
  - Use the target words to predict their context words
  - A |V|-way classification problem: |V| potential context word.

  - F: input x labels → score s ([0.1, 0.2, 0.3]).
  - Softmax: score s-> p(c|t)
  - Minimize the difference between p(c|t) and labels
    - cross entropy: $-log\,\frac{\exp(\boldsymbol{u}_t \cdot \boldsymbol{v}_c)}{\sum_a \exp(\boldsymbol{u}_t \cdot \boldsymbol{v}_a)}$
    - u: word embedding.
    - v: context word embedding.

Skip-gram

- Learning objective
  - Use the target words to predict their context words
  - A |V|-way classification problem: |V| potential context word.

  - F: input x labels → score s ([0.1, 0.2, 0.3]).
  - Softmax: score s-> p(c|t)
  - Minimize the difference between p(c|t) and labels
    - cross entropy: $-log \frac{\exp(\boldsymbol{u}_t \cdot \boldsymbol{v}_c)}{\sum_a \exp(\boldsymbol{u}_t \cdot \boldsymbol{v}_a)}$ → Learning objective for one target-context pair.
    - u: word embedding.
    - v: context word embedding.

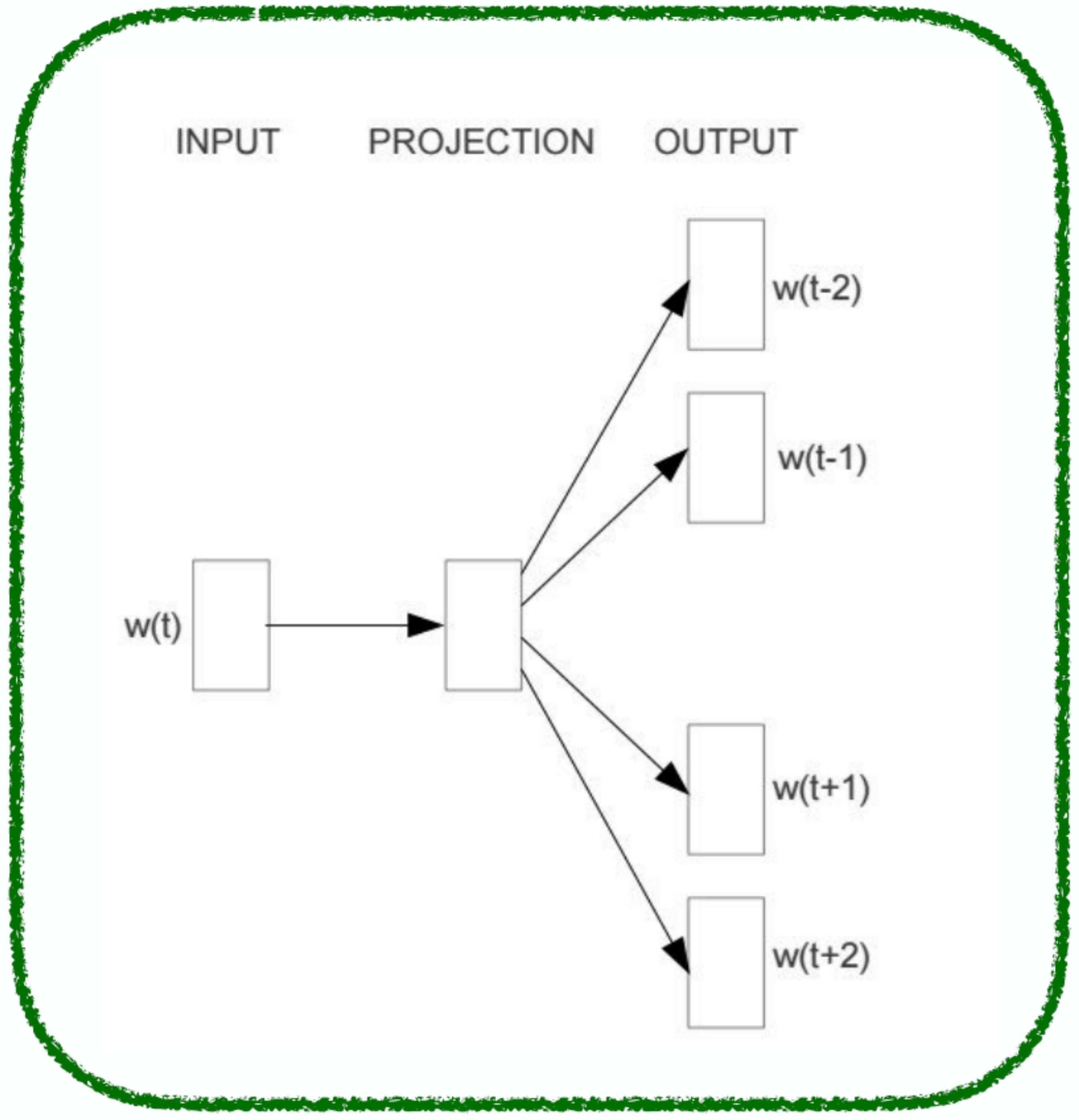
Skip-gram

- Learning objective
  - Use the target words to predict their context words
  - A |V|-way classification problem: |V| potential context word.

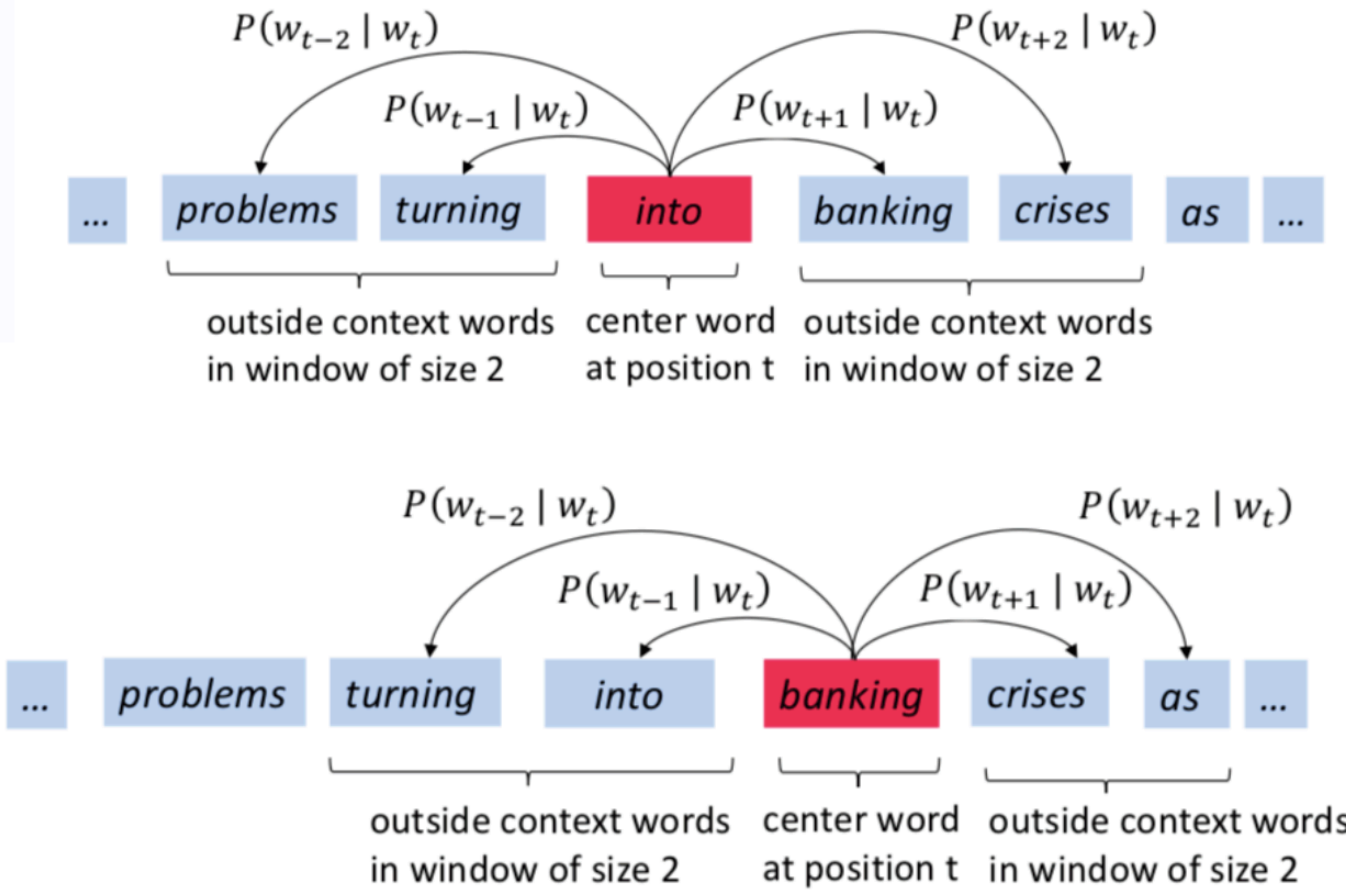
  - F: input x labels → score s ([0.1, 0.2, 0.3]).
  - Softmax: score s-> p(c|t)
  - Minimize the difference between p(c|t) and labels
    - cross entropy: $-log\frac{\exp(\boldsymbol{u}_t \cdot \boldsymbol{v}_c)}{\sum_a \exp(\boldsymbol{u}_t \cdot \boldsymbol{v}_a)}$ → Learning objective for one target-context pair. Follow the training corpora, sum over all target-context pairs.
    - u: word embedding.
    - v: context word embedding.

Skip-gram

$$J(\theta) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{-m\le j\le m, j\neq 0}\log\frac{\exp(\mathbf{u}_{w_t}\cdot\mathbf{v}_{w_{t+j}})}{\sum_{k\in V}\exp(\mathbf{u}_{w_t}\cdot\mathbf{v}_k)}$$



Skip-gram



Skip-gram

$$J(\theta) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{-m\leq j\leq m, j\neq 0}\log\frac{\exp(\mathbf{u}_{w_t}\cdot\mathbf{v}_{w_{t+j}})}{\sum_{k\in V}\exp(\mathbf{u}_{w_t}\cdot\mathbf{v}_k)}$$

- Optimization
  - Non-convex
  - Gradient descent.
  - Too slow to update all context word embeddings v_k at every step.

Skip-gram

$$J(\theta) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{-m\leq j\leq m, j\neq 0}\log\frac{\exp(\mathbf{u}_{w_t}\cdot\mathbf{v}_{w_{t+j}})}{\sum_{k\in V}\exp(\mathbf{u}_{w_t}\cdot\mathbf{v}_k)}$$

- Optimization
  - Non-convex
  - Gradient descent.
  - Too slow to update all context word embeddings v_k at every step.

  - Use negative sampling

Matrix calculus to compute 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 and section 3.

  - Today, we will look at
    - Section 2
    - Section 3 (5)
    - Section 3 (7)

Vectorized gradients

Next, we are going to compute gradients with respect to many variables together and write them in vector/matrix notations.

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

$$\boldsymbol{f}(\boldsymbol{x}) = [f_1(x_1, ..., x_n), f_2(x_1, ..., x_n), ..., f_m(x_1, ..., x_n)]$$

$$f(\mathbf{x}) = \mathbf{x} \in \mathbb{R}^n$$

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

Vectorized gradients

Next, we are going to compute gradients with respect to many variables together and write them in vector/matrix notations.

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

$$\boldsymbol{f}(\boldsymbol{x}) = [f_1(x_1, ..., x_n), f_2(x_1, ..., x_n), ..., f_m(x_1, ..., x_n)]$$

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

$$f(\mathbf{x}) = \mathbf{x} \in \mathbb{R}^n$$

$$\frac{\partial f}{\partial \mathbf{x}} = I_n$$

Vectorized gradients

Next, we are going to compute gradients with respect to many variables together and write them in vector/matrix notations.

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

$$\boldsymbol{f}(\boldsymbol{x}) = [f_1(x_1, ..., x_n), f_2(x_1, ..., x_n), ..., f_m(x_1, ..., x_n)]$$

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

$$f(\mathbf{x}) = \mathbf{x} \in \mathbb{R}^n$$

$$\frac{\partial f}{\partial \mathbf{x}} = I_n \qquad \frac{\partial f_i}{\partial x_j} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

If m = 1 (loss), the shape of gradients is the same as the shape of input.

Let's compute gradients for word2vec

$$J(\theta) = -\frac{1}{T}\sum_{t=1}^{T}\sum_{-m\leq j\leq m, j\neq 0}\log\frac{\exp(\mathbf{u}_{w_t}\cdot\mathbf{v}_{w_{t+j}})}{\sum_{k\in V}\exp(\mathbf{u}_{w_t}\cdot\mathbf{v}_k)}$$

Consider one pair of center/context words (t, c):

$$y = -\log\left(\frac{\exp(\mathbf{u}_t\cdot\mathbf{v}_c)}{\sum_{k\in V}\exp(\mathbf{u}_t\cdot\mathbf{v}_k)}\right)$$

We need to compute the gradient of y with respect to

$\mathbf{u}_t$ and $\mathbf{v}_k, \forall k \in V$

Let's compute gradients for word2vec

$$y = -\log \left(\frac{\exp(\mathbf{u}_t \cdot \mathbf{v}_c)}{\sum_{k \in V} \exp(\mathbf{u}_t \cdot \mathbf{v}_k)} \right)$$

$$y = -\log(\exp(\mathbf{u}_t \cdot \mathbf{v}_c)) + \log(\sum_{k \in V} \exp(\mathbf{u}_t \cdot \mathbf{v}_k))$$

$$= -\mathbf{u}_t \cdot \mathbf{v}_c + \log(\sum_{k \in V} \exp(\mathbf{u}_t \cdot \mathbf{v}_k))$$

Recall that

$$P(w_{t+j} \mid w_t) = \frac{\exp(\mathbf{u}_{w_t} \cdot \mathbf{v}_{w_{t+j}})}{\sum_{k \in V} \exp(\mathbf{u}_{w_t} \cdot \mathbf{v}_k)}$$

$$\frac{\partial y}{\partial \mathbf{u}_t} = \frac{\partial(-\mathbf{u}_t \cdot \mathbf{v}_c)}{\partial \mathbf{u}_t} + \frac{\partial(\log \sum_{k \in V} \exp(\mathbf{u}_t \cdot \mathbf{v}_k))}{\partial \mathbf{u}_t}$$

$$= -\mathbf{v}_c + \frac{\frac{\partial \sum_{k \in V} \exp(\mathbf{u}_t \cdot \mathbf{v}_k)}{\partial \mathbf{u}_t}}{\sum_{k \in V} \exp(\mathbf{u}_t \cdot \mathbf{v}_k)}$$

$$= -\mathbf{v}_c + \frac{\sum_{k \in V} \exp(\mathbf{u}_t \cdot \mathbf{v}_k) \cdot \mathbf{v}_k}{\sum_{k \in V} \exp(\mathbf{u}_t \cdot \mathbf{v}_k)}$$

$$= -\mathbf{v}_c + \sum_{k \in V} \frac{\exp(\mathbf{u}_t \cdot \mathbf{v}_k)}{\sum_{k' \in V} \exp(\mathbf{u}_t \cdot \mathbf{v}_{k'})} \mathbf{v}_k$$

$$= -\mathbf{v}_c + \sum_{k \in V} P(k \mid t) \mathbf{v}_k$$

Gradients for word2vec

What about context vectors?

$$\frac{\partial y}{\partial \mathbf{v}_k} = \begin{cases} (P(k \mid t) - 1)\,\mathbf{u}_t & k = c \\ P(k \mid t)\mathbf{u}_t & k \neq c \end{cases}$$

$$y = -\log\left(\frac{\exp(\mathbf{u}_t \cdot \mathbf{v}_c)}{\sum_{k \in V} \exp(\mathbf{u}_t \cdot \mathbf{v}_k)}\right)$$

See assignment 1 :)

Overall algorithm

- Input: text corpus, embedding size d, vocabulary V, **context size m**
- Initialize $\mathbf{u}_i, \mathbf{v}_i$ randomly $\forall i \in V$
- Run through the training corpus and for each training instance (*t*, *c*):
  - Update $\mathbf{u}_t \leftarrow \mathbf{u}_t - \eta \frac{\partial y}{\partial \mathbf{u_t}}$ $\qquad \frac{\partial y}{\partial \mathbf{u}_t} = -\mathbf{v}_c + \sum_{k \in V} P(k \mid t)\mathbf{v}_k$
  - Update $\mathbf{v}_k \leftarrow \mathbf{v}_k - \eta \frac{\partial y}{\partial \mathbf{v_k}}, \forall k \in V$ $\qquad \frac{\partial y}{\partial \mathbf{v}_k} = \begin{cases} (P(k \mid t) - 1)\,\mathbf{u}_t & k = c \\ P(k \mid t)\mathbf{u}_t & k \neq c \end{cases}$

Convert the training data into:
(into, problems)
(into, turning)
(into, banking)
(into, crises)
(banking, turning)
(banking, into)
(banking, crises)
(banking, as)
…

Q: Can you think of any issues with this algorithm?

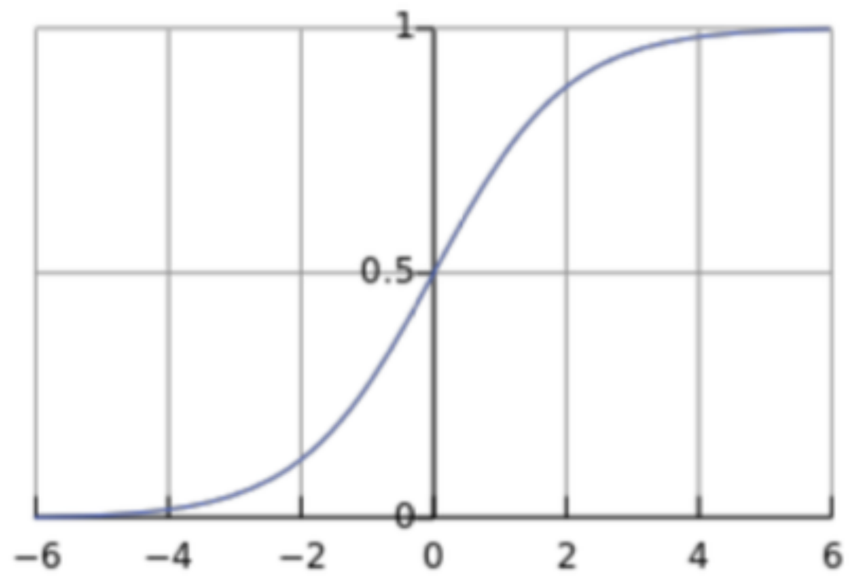
Skip-gram with negative sampling (SGNS)

Problem: every time you get one pair of (t, c), you need to update $\mathbf{v}_k$ with all the words in the vocabulary! This is very expensive computationally.

$$\frac{\partial y}{\partial \mathbf{u}_t} = -\mathbf{v}_c + \sum_{k \in V} P(k \mid t)\mathbf{v}_k \qquad \frac{\partial y}{\partial \mathbf{v}_k} = \begin{cases} (P(k \mid t) - 1)\,\mathbf{u}_t & k = c \\ P(k \mid t)\mathbf{u}_t & k \neq c \end{cases}$$

Negative sampling: instead of considering all the words in V, let's randomly sample K (5-20) negative examples.

$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$



softmax: $$y = -\log\left(\frac{\exp(\mathbf{u}_t \cdot \mathbf{v}_c)}{\sum_{k \in V} \exp(\mathbf{u}_t \cdot \mathbf{v}_k)}\right)$$

Negative sampling: $$y = -\log(\sigma(\mathbf{u}_t \cdot \mathbf{v}_c)) - \sum_{i=1}^{K} \mathbb{E}_{j \sim P(w)} \log(\sigma(-\mathbf{u}_t \cdot \mathbf{v}_j))$$

Skip-gram with negative sampling (SGNS)

Key idea: Convert the $|V|$-way classification into a set of binary classification tasks.

Every time we get a pair of words (*t*, *c*), we don't predict *c* among all the words in the vocabulary. Instead, we predict (*t*, *c*) is a positive pair, and (*t*, *c'*) is a negative pair for a small number of sampled *c'*.

positive examples +

t	c
apricot	tablespoon
apricot	of
apricot	jam
apricot	a

negative examples -

t	c	t	c
apricot	aardvark	apricot	seven
apricot	my	apricot	forever
apricot	where	apricot	dear
apricot	coaxial	apricot	if

$$y = -\log(\sigma(\mathbf{u}_t \cdot \mathbf{v}_c)) - \sum_{i=1}^{K} \mathbb{E}_{j \sim P(w)} \log(\sigma(-\mathbf{u}_t \cdot \mathbf{v}_j))$$

P(w): sampling according to the frequency of words

Similar to **binary logistic regression**, but we need to optimize $\mathbf{u}$ and $\mathbf{v}$ together.

$$P(y = 1 \mid t, c) = \sigma(\mathbf{u}_t \cdot \mathbf{v}_c) \qquad p(y = 0 \mid t, c') = 1 - \sigma(\mathbf{u}_t \cdot \mathbf{v}_{c'}) = \sigma(-\mathbf{u}_t \cdot \mathbf{v}_{c'})$$

Skip-gram with Negative Sampling

Recall the loss for a particular (word, context word) pair in the Skip-gram with Negative Sampling model:

$$J(w, c_{pos}, \mathbf{U}, \mathbf{V}) = -\log(\sigma(\mathbf{u}_w^\intercal \cdot \mathbf{v}_{c_{pos}})) - \sum_{c_{neg} \in W_{neg}} \log(\sigma(-\mathbf{u}_w^\intercal \cdot \mathbf{v}_{c_{neg}}))$$

σ is the sigmoid function

c_{pos} is the positive context word

w is the center word

$\mathbf{u}_w$ is the center word vector for word w

$\mathbf{v}_{c_{pos}}$ is the context word vector for context word c_{pos}

W_{neg} are the K negative context word samples

Calculate:

(a) $\frac{\partial J}{\partial \mathbf{u}_w}$ (b) $\frac{\partial J}{\partial \mathbf{v}_{c_{pos}}}$

(c) $\frac{\partial J}{\partial \mathbf{v}_{c_{neg}}}$

Skip-gram with Negative Sampling (a)

$$J(w, c_{pos}, \mathbf{U}, \mathbf{V}) = -\log(\sigma(\mathbf{u}_w^\intercal \cdot \mathbf{v}_{c_{pos}})) - \sum_{c_{neg} \in W_{neg}} \log(\sigma(-\mathbf{u}_w^\intercal \cdot \mathbf{v}_{c_{neg}}))$$

$$\frac{\partial J}{\partial \mathbf{u}_w} =$$

Skip-gram with Negative Sampling (a)

$$J(w, c_{pos}, \mathbf{U}, \mathbf{V}) = -\log(\sigma(\mathbf{u}_w^\intercal \cdot \mathbf{v}_{c_{pos}})) - \sum_{c_{neg} \in W_{neg}} \log(\sigma(-\mathbf{u}_w^\intercal \cdot \mathbf{v}_{c_{neg}}))$$

$$\frac{\partial J}{\partial \mathbf{u}_w} = -\frac{\sigma(\mathbf{u}_w^\intercal \mathbf{v}_{c_{pos}})(1 - \sigma(\mathbf{u}_w^\intercal \mathbf{v}_{c_{pos}})) \cdot \mathbf{v}_{c_{pos}}}{\sigma(\mathbf{u}_w^\intercal \mathbf{v}_{c_{pos}})} - \sum_{c_{neg} \in W_{neg}} \frac{\sigma(-\mathbf{u}_w^\intercal \mathbf{v}_{c_{neg}})(1 - \sigma(-\mathbf{u}_w^\intercal \mathbf{v}_{c_{neg}})) \cdot -\mathbf{v}_{c_{neg}}}{\sigma(-\mathbf{u}_w^\intercal \mathbf{v}_{c_{neg}})}$$

$$\sigma'(x) = \sigma(x)(1 - \sigma(x))$$

Skip-gram with Negative Sampling (a)

$$J(w, c_{pos}, \mathbf{U}, \mathbf{V}) = -\log(\sigma(\mathbf{u}_w^\intercal \cdot \mathbf{v}_{c_{pos}})) - \sum_{c_{neg} \in W_{neg}} \log(\sigma(-\mathbf{u}_w^\intercal \cdot \mathbf{v}_{c_{neg}}))$$

$$\frac{\partial J}{\partial \mathbf{u}_w} = -\frac{\sigma(\mathbf{u}_w^\intercal \mathbf{v}_{c_{pos}})(1 - \sigma(\mathbf{u}_w^\intercal \mathbf{v}_{c_{pos}})) \cdot \mathbf{v}_{c_{pos}}}{\sigma(\mathbf{u}_w^\intercal \mathbf{v}_{c_{pos}})} - \sum_{c_{neg} \in W_{neg}} \frac{\sigma(-\mathbf{u}_w^\intercal \mathbf{v}_{c_{neg}})(1 - \sigma(-\mathbf{u}_w^\intercal \mathbf{v}_{c_{neg}})) \cdot -\mathbf{v}_{c_{neg}}}{\sigma(-\mathbf{u}_w^\intercal \mathbf{v}_{c_{neg}})}$$

Skip-gram with Negative Sampling (a)

$$J(w, c_{pos}, \mathbf{U}, \mathbf{V}) = -\log(\sigma(\mathbf{u}_w^\intercal \cdot \mathbf{v}_{c_{pos}})) - \sum_{c_{neg} \in W_{neg}} \log(\sigma(-\mathbf{u}_w^\intercal \cdot \mathbf{v}_{c_{neg}}))$$

$$\frac{\partial J}{\partial \mathbf{u}_w} = -\frac{\cancel{\sigma(\mathbf{u}_w^\intercal \mathbf{v}_{c_{pos}})}(1 - \sigma(\mathbf{u}_w^\intercal \mathbf{v}_{c_{pos}})) \cdot \mathbf{v}_{c_{pos}}}{\cancel{\sigma(\mathbf{u}_w^\intercal \mathbf{v}_{c_{pos}})}} - \sum_{c_{neg} \in W_{neg}} \frac{\cancel{\sigma(-\mathbf{u}_w^\intercal \mathbf{v}_{c_{neg}})}(1 - \sigma(-\mathbf{u}_w^\intercal \mathbf{v}_{c_{neg}})) \cdot -\mathbf{v}_{c_{neg}}}{\cancel{\sigma(-\mathbf{u}_w^\intercal \mathbf{v}_{c_{neg}})}}$$

$$\frac{\partial J}{\partial \mathbf{u}_w} = -(1 - \sigma(\mathbf{u}_w^\intercal \mathbf{v}_{c_{pos}})) \cdot \mathbf{v}_{c_{pos}} + \sum_{c_{neg} \in W_{neg}} (1 - \sigma(-\mathbf{u}_w^\intercal \mathbf{v}_{c_{neg}})) \cdot \mathbf{v}_{c_{neg}}$$

$$\sigma(-x) = (1 - \sigma(x))$$

Skip-gram with Negative Sampling (a)

$$J(w, c_{pos}, \mathbf{U}, \mathbf{V}) = -\log(\sigma(\mathbf{u}_w^\intercal \cdot \mathbf{v}_{c_{pos}})) - \sum_{c_{neg} \in W_{neg}} \log(\sigma(-\mathbf{u}_w^\intercal \cdot \mathbf{v}_{c_{neg}}))$$

$$\frac{\partial J}{\partial \mathbf{u}_w} = -\frac{\cancel{\sigma(\mathbf{u}_w^\intercal \mathbf{v}_{c_{pos}})}(1 - \sigma(\mathbf{u}_w^\intercal \mathbf{v}_{c_{pos}})) \cdot \mathbf{v}_{c_{pos}}}{\cancel{\sigma(\mathbf{u}_w^\intercal \mathbf{v}_{c_{pos}})}} - \sum_{c_{neg} \in W_{neg}} \frac{\cancel{\sigma(-\mathbf{u}_w^\intercal \mathbf{v}_{c_{neg}})}(1 - \sigma(-\mathbf{u}_w^\intercal \mathbf{v}_{c_{neg}})) \cdot -\mathbf{v}_{c_{neg}}}{\cancel{\sigma(-\mathbf{u}_w^\intercal \mathbf{v}_{c_{neg}})}}$$

$$\frac{\partial J}{\partial \mathbf{u}_w} = -(1 - \sigma(\mathbf{u}_w^\intercal \mathbf{v}_{c_{pos}})) \cdot \mathbf{v}_{c_{pos}} + \sum_{c_{neg} \in W_{neg}} (1 - \sigma(-\mathbf{u}_w^\intercal \mathbf{v}_{c_{neg}})) \cdot \mathbf{v}_{c_{neg}}$$

$$\frac{\partial J}{\partial \mathbf{u}_w} = (\sigma(\mathbf{u}_w^\intercal \mathbf{v}_{c_{pos}}) - 1) \cdot \mathbf{v}_{c_{pos}} + \sum_{c_{neg} \in W_{neg}} \sigma(\mathbf{u}_w^\intercal \mathbf{v}_{c_{neg}}) \cdot \mathbf{v}_{c_{neg}}$$

Skip-gram with Negative Sampling (b)

$$J(w, c_{pos}, \mathbf{U}, \mathbf{V}) = -\log(\sigma(\mathbf{u}_w^\intercal \cdot \mathbf{v}_{c_{pos}})) - \sum_{c_{neg} \in W_{neg}} \log(\sigma(-\mathbf{u}_w^\intercal \cdot \mathbf{v}_{c_{neg}}))$$

$$\frac{\partial J}{\partial \mathbf{v}_{c_{pos}}} =$$

Skip-gram with Negative Sampling (b)

$$J(w, c_{pos}, \mathbf{U}, \mathbf{V}) = -\log(\sigma(\mathbf{u}_w^\top \cdot \mathbf{v}_{c_{pos}})) - \sum_{c_{neg} \in W_{neg}} \log(\sigma(-\mathbf{u}_w^\top \cdot \mathbf{v}_{c_{neg}}))$$

↑ Constant! (Not in terms of $\mathrm{v}_{c_{pos}}$)

$$\frac{\partial J}{\partial \mathbf{v}_{c_{pos}}} =$$

Skip-gram with Negative Sampling (b)

$$J(w, c_{pos}, \mathbf{U}, \mathbf{V}) = -\log(\sigma(\mathbf{u}_w^\intercal \cdot \mathbf{v}_{c_{pos}})) - \sum_{c_{neg} \in W_{neg}} \log(\sigma(-\mathbf{u}_w^\intercal \cdot \mathbf{v}_{c_{neg}}))$$

$$\frac{\partial J}{\partial \mathbf{v}_{c_{pos}}} = \frac{-\sigma(\mathbf{u}_w^\intercal \mathbf{v}_{c_{pos}})(1 - \sigma(\mathbf{u}_w^\intercal \mathbf{v}_{c_{pos}})) \cdot \mathbf{u}_w}{\sigma(\mathbf{u}_w^\intercal \mathbf{v}_{c_{pos}})} + 0$$

Skip-gram with Negative Sampling (b)

$$J(w, c_{pos}, \mathbf{U}, \mathbf{V}) = -\log(\sigma(\mathbf{u}_w^\intercal \cdot \mathbf{v}_{c_{pos}})) - \sum_{c_{neg} \in W_{neg}} \log(\sigma(-\mathbf{u}_w^\intercal \cdot \mathbf{v}_{c_{neg}}))$$

$$\frac{\partial J}{\partial \mathbf{v}_{c_{pos}}} = \frac{-\sigma(\mathbf{u}_w^\intercal \mathbf{v}_{c_{pos}})(1 - \sigma(\mathbf{u}_w^\intercal \mathbf{v}_{c_{pos}})) \cdot \mathbf{u}_w}{\sigma(\mathbf{u}_w^\intercal \mathbf{v}_{c_{pos}})} + 0$$

Skip-gram with Negative Sampling (b)

$$J(w, c_{pos}, \mathbf{U}, \mathbf{V}) = -\log(\sigma(\mathbf{u}_w^\intercal \cdot \mathbf{v}_{c_{pos}})) - \sum_{c_{neg} \in W_{neg}} \log(\sigma(-\mathbf{u}_w^\intercal \cdot \mathbf{v}_{c_{neg}}))$$

$$\frac{\partial J}{\partial \mathbf{v}_{c_{pos}}} = \frac{-\sigma(\mathbf{u}_w^\intercal \mathbf{v}_{c_{pos}})(1 - \sigma(\mathbf{u}_w^\intercal \mathbf{v}_{c_{pos}})) \cdot \mathbf{u}_w}{\sigma(\mathbf{u}_w^\intercal \mathbf{v}_{c_{pos}})} + 0$$

$$\frac{\partial J}{\partial \mathbf{v}_{c_{pos}}} = (\sigma(\mathbf{u}_w^\intercal \mathbf{v}_{c_{pos}}) - 1) \cdot \mathbf{u}_w$$

Skip-gram with Negative Sampling (c)

$$J(w, c_{pos}, \mathbf{U}, \mathbf{V}) = -\log(\sigma(\mathbf{u}_w^\intercal \cdot \mathbf{v}_{c_{pos}})) - \sum_{c_{neg} \in W_{neg}} \log(\sigma(-\mathbf{u}_w^\intercal \cdot \mathbf{v}_{c_{neg}}))$$

$$\frac{\partial J}{\partial \mathbf{v}_{c_{neg}}} =$$

Skip-gram with Negative Sampling (c)

$$J(w, c_{pos}, \mathbf{U}, \mathbf{V}) = -\log(\sigma(\mathbf{u}_w^\intercal \cdot \mathbf{v}_{c_{pos}})) - \sum_{c_{neg} \in W_{neg}} \log(\sigma(-\mathbf{u}_w^\intercal \cdot \mathbf{v}_{c_{neg}}))$$

$$\frac{\partial J}{\partial \mathbf{v}_{c_{neg}}} =$$

Constant!
(Not in terms of $v_{c_{neg}}$)

(Mostly) constant (not in terms of $v_{c_{neg}}$)
with the exception of 1
sampled negative!

Skip-gram with Negative Sampling (c)

$$J(w, c_{pos}, \mathbf{U}, \mathbf{V}) = -\log(\sigma(\mathbf{u}_w^\intercal \cdot \mathbf{v}_{c_{pos}})) - \sum_{c_{neg} \in W_{neg}} \log(\sigma(-\mathbf{u}_w^\intercal \cdot \mathbf{v}_{c_{neg}}))$$

$$\frac{\partial J}{\partial \mathbf{v}_{c_{neg}}} = -\frac{\sigma(-\mathbf{u}_w^\intercal \mathbf{v}_{c_{neg}})(1 - \sigma(-\mathbf{u}_w^\intercal \mathbf{v}_{c_{neg}})) \cdot -\mathbf{u}_w}{\sigma(-\mathbf{u}_w^\intercal \mathbf{v}_{c_{neg}})}$$

Skip-gram with Negative Sampling (c)

$$J(w, c_{pos}, \mathbf{U}, \mathbf{V}) = -\log(\sigma(\mathbf{u}_w^\intercal \cdot \mathbf{v}_{c_{pos}})) - \sum_{c_{neg} \in W_{neg}} \log(\sigma(-\mathbf{u}_w^\intercal \cdot \mathbf{v}_{c_{neg}}))$$

$$\frac{\partial J}{\partial \mathbf{v}_{c_{neg}}} = -\frac{\sigma(-\mathbf{u}_w^\intercal \mathbf{v}_{c_{neg}})(1 - \sigma(-\mathbf{u}_w^\intercal \mathbf{v}_{c_{neg}})) \cdot -\mathbf{u}_w}{\sigma(-\mathbf{u}_w^\intercal \mathbf{v}_{c_{neg}})}$$

Skip-gram with Negative Sampling (c)

$$J(w, c_{pos}, \mathbf{U}, \mathbf{V}) = -\log(\sigma(\mathbf{u}_w^\intercal \cdot \mathbf{v}_{c_{pos}})) - \sum_{c_{neg} \in W_{neg}} \log(\sigma(-\mathbf{u}_w^\intercal \cdot \mathbf{v}_{c_{neg}}))$$

$$\frac{\partial J}{\partial \mathbf{v}_{c_{neg}}} = -\frac{\sigma(-\mathbf{u}_w^\intercal \mathbf{v}_{c_{neg}})(1 - \sigma(-\mathbf{u}_w^\intercal \mathbf{v}_{c_{neg}})) \cdot -\mathbf{u}_w}{\sigma(-\mathbf{u}_w^\intercal \mathbf{v}_{c_{neg}})}$$

$$\frac{\partial J}{\partial \mathbf{v}_{c_{neg}}} = \sigma(\mathbf{u}_w^\intercal \mathbf{v}_{c_{neg}}) \cdot \mathbf{u}_w$$