deeplearning.ai

NLP and Word Embeddings

Word representation

Word representation

words with similar meanings have similar representations

$V = [a, aaron, \ldots, zulu, <UNK>]$

$|V| = 10{,}000$

1-hot representation

inner product shall be zero, when words are related

Man	Woman	King	Queen	Apple	Orange
(5391)	(9853)	(4914)	(7157)	(456)	(6257)

O_{5391} O_{9853}

I want a glass of orange juice.

I want a glass of apple ?.

we aim to replace one-hot, which is computationally expensive in higher dimensions, with something more agile & dense

Andrew Ng

Featurized representation: word embedding

Feature list	Man (5391)	Woman (9853)	King (4914)	Queen (7157)	Apple (456)	Orange (6257)
Gender	-1	1	-0.95	0.97	0.00	0.01
Royal	0.01	0.02	0.93	0.95	-0.01	0.00
Age	0.03	0.02	0.7	0.69	0.03	-0.02
Food	0.04	0.01	0.02	0.01	0.95	0.97
⋮	⋮	⋮				
Size						
Cost						
Alive						
Verb						

300

e_{5391} e_{9853}

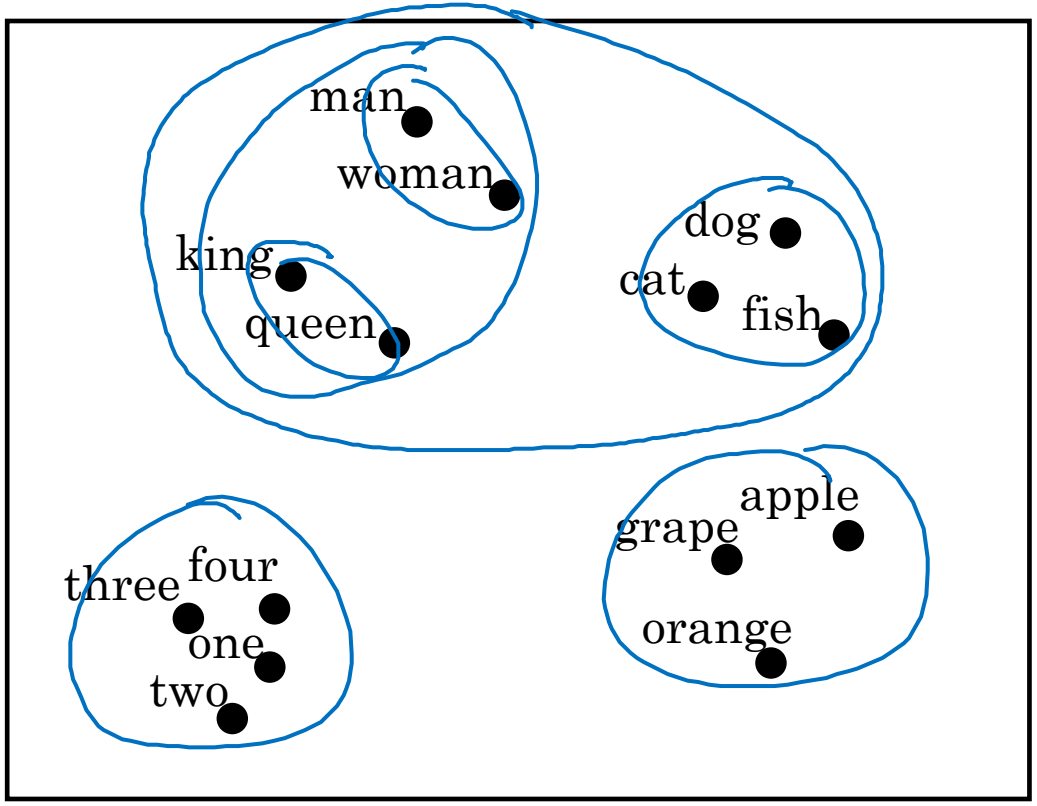
I want a glass of orange juice.

I want a glass of apple juice.

Andrew Ng

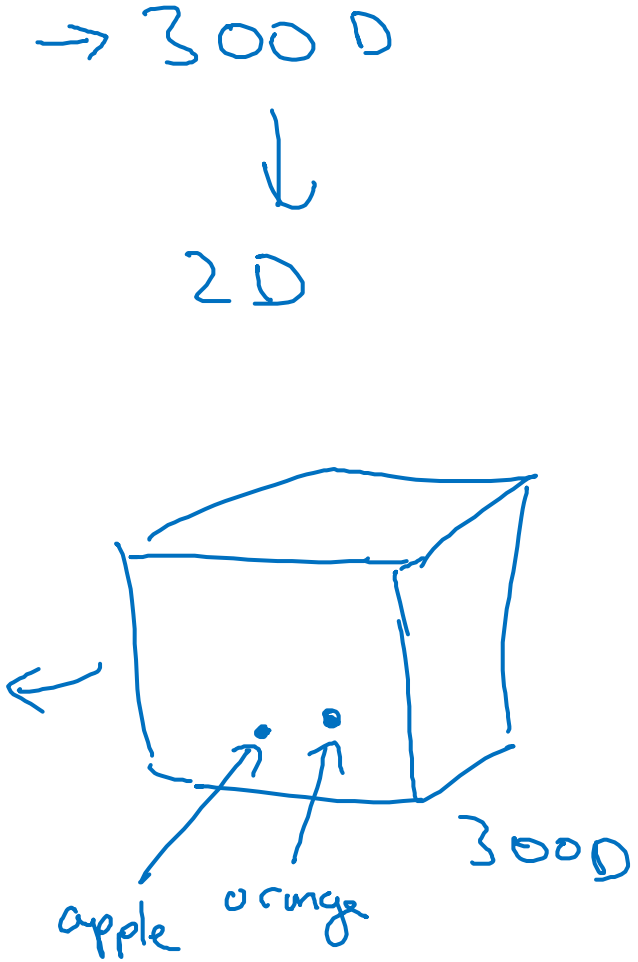
Visualizing word embeddings

Dimension of word vectors is usually smaller than the size of the vocabulary.
most common sizes for word vec ranges bet 50 & 400



t-SNE

non-linear dimensionality reduction technique



[van der Maaten and Hinton., 2008. Visualizing data using t-SNE]

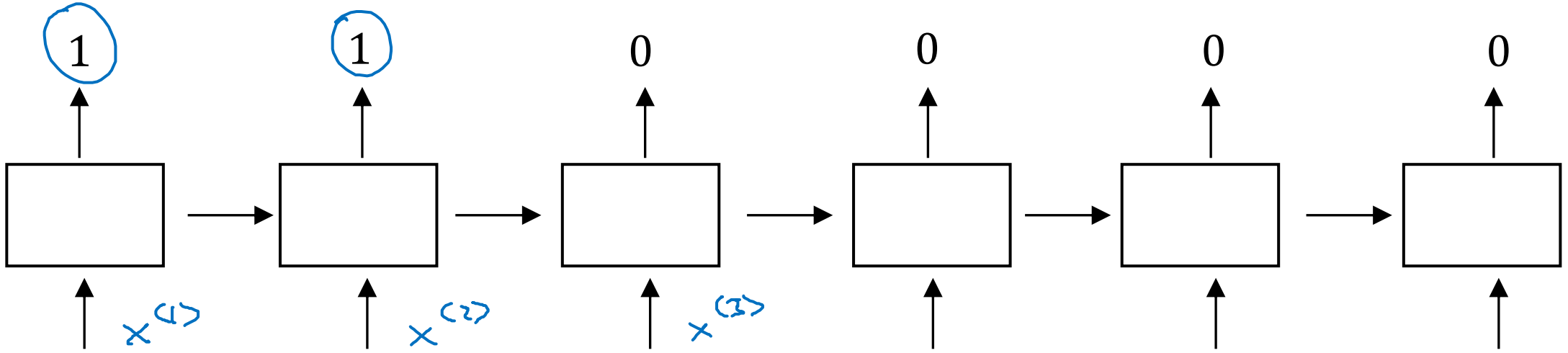
Andrew Ng

deeplearning.ai

NLP and Word Embeddings

Using word embeddings

Named entity recognition example



Robert Lin is ~~an~~ ~~apple~~ ~~farmer~~

durian

cultivator.

training readily available online corpus → 1B words – 100B words

transfer learning to smaller selected dataset → 100k words

a

RNN

less common, yet can be identified with embedding

Andrew Ng

Transfer learning and word embeddings

makes huge difference when trainingset is relatively small

A

1. Learn word embeddings from large text corpus. (1-100B words)

   (Or download pre-trained embedding online.)

B

2. Transfer embedding to new task with smaller training set. (say, 100k words)

instead of using → 10,000 Dim → 300 Dim

3. Optional: Continue to finetune the word embeddings with new data.

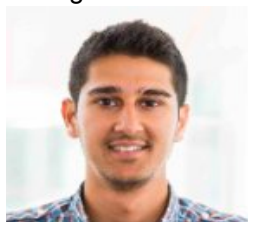
Relation to face encoding

synonymous to each other

(embedding)

128D

could be a totally new image



$x^{(i)}$

$x^{(j)}$

$f(x^{(i)})$

$f(x^{(j)})$

$\hat{y}$

fixed # words to train

$|V| = 10{,}000$

$e_1, \ldots, e_{10{,}000}$

[Taigman et. al., 2014. DeepFace: Closing the gap to human level performance]

deeplearning.ai

NLP and Word Embeddings

Properties of word embeddings

Analogies

A good word embedding: e_boy - e_girl = e_brother - e_sister
e_boy - e_brother = e_girl - e_sister

assuming 4-dim embedding	Man (5391)	Woman (9853)	King (4914)	Queen (7157)	Apple (456)	Orange (6257)
Gender	−1	1	-0.95	0.97	0.00	0.01
Royal	0.01	0.02	0.93	0.95	-0.01	0.00
Age	0.03	0.02	0.70	0.69	0.03	-0.02
Food	0.09	0.01	0.02	0.01	0.95	0.97

e_{5391} e_{man} e_{woman}

assuming..

Man → Woman ≈ King → ? Queen

required to compute $e_{man} - e_{woman} \approx e_{king} - e_{?}$

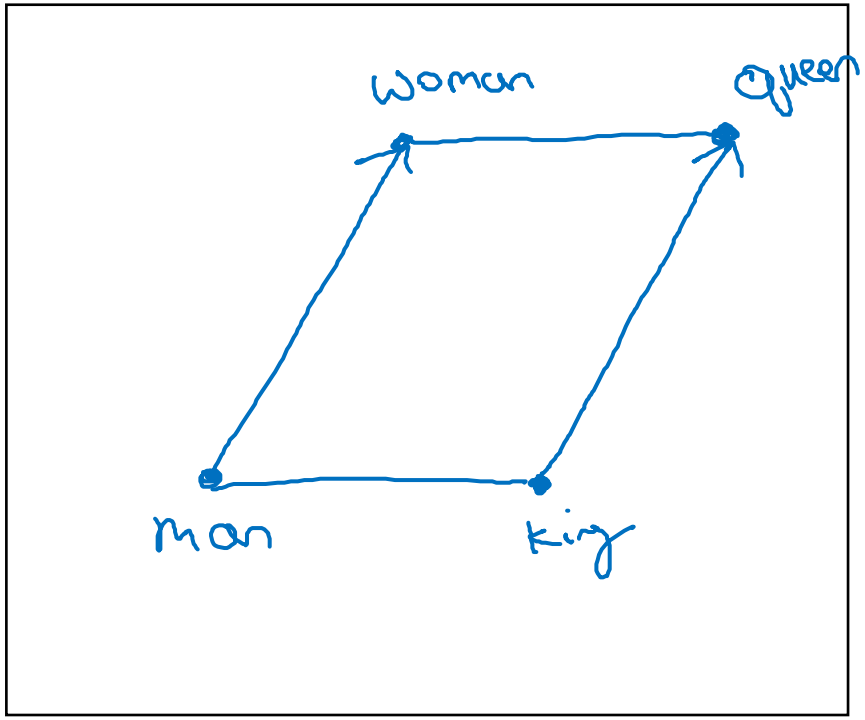
$?$ ≈ Queen

vec difference

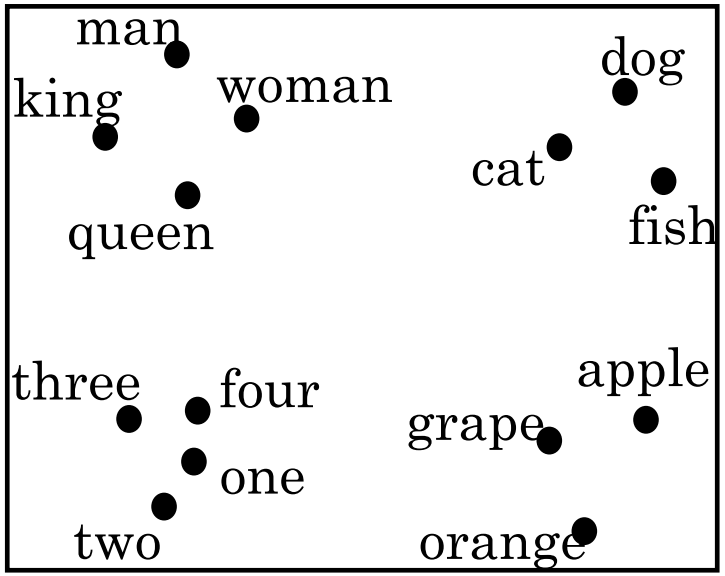
- $e_{man} - e_{woman} \approx \begin{bmatrix} -2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$
- $e_{king} - e_{queen} \approx \begin{bmatrix} -2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

[Mikolov et. al., 2013, Linguistic regularities in continuous space word representations]

Analogies using word vectors



300D



300 → 2D

t-SNE

$$e_{man} - e_{woman} \approx e_{king} - e_?$$

Find word w: $\arg\max_w$ $\text{sim}(e_w, e_{king} - e_{man} + e_{woman})$

similarity function

30 - 75%

research papers acc

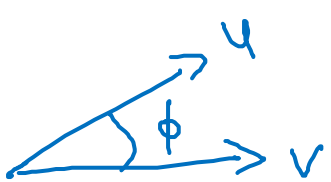
Andrew Ng

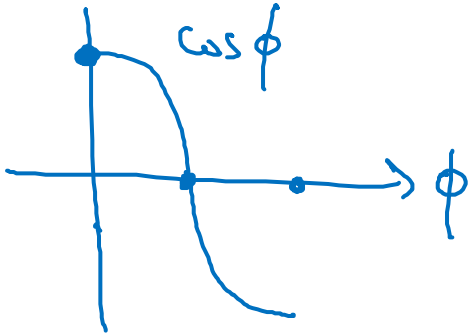
Cosine similarity

$$sim(e_w, e_{king} - e_{man} + e_{woman})$$

$$sim(u, v) = \frac{u^T v}{\|u\|_2 \|v\|_2}$$

it's a judgement of orientation, not magnitude.
i.e 2 vec with same orientation have a cosine similarity of 1





$$\|u - v\|^2$$

another functions: squared difference, measures the dissimilarities though

Man:Woman as Boy:Girl

Ottawa:Canada as Nairobi:Kenya

Big:Bigger as Tall:Taller

Yen:Japan as Ruble:Russia

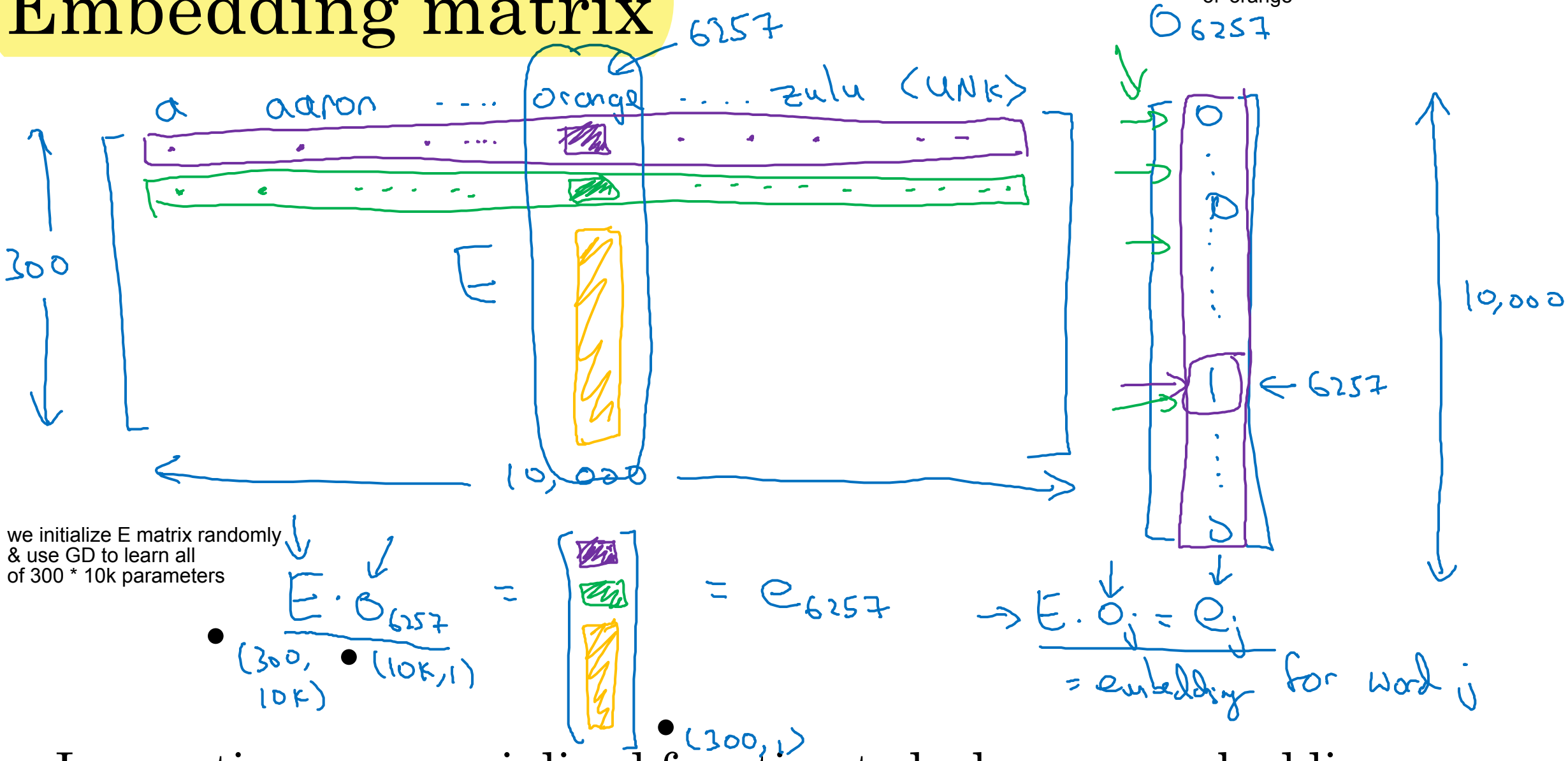
Andrew Ng

deeplearning.ai

NLP and Word Embeddings

Embedding matrix

Embedding matrix



In practice, use specialized function to look up an embedding.

keras → Embedding

more efficient to be used than doing mat mul.
because of such high dim in one-hot

Andrew Ng

deeplearning.ai

NLP and Word Embeddings

Learning word embeddings

Neural language model

earlier complex models

this is supervised learninig

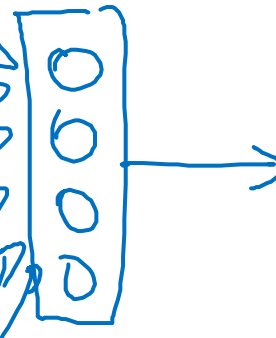
I	want	a	glass	of	orange	juice	.
4343	9665	1	3852	6163	6257		

$e_{4343} = E\, o_{4343}$

apple juice .

I	o_{4343}	E	e_{4343}	
want	o_{9665}	E	e_{9665}	
a	o_1	E	e_1	
glass	o_{3852}	E	e_{3852}	
of	o_{6163}	E	e_{6163}	
orange	o_{6257}	E	e_{6257}	

same matrix E



Softmax

10,000

$W^{[1]}, b^{[1]}$

$W^{[2]}, b^{[2]}$

6 embedding * 300 dim

1800 1200

choosing the last 4 words before the word to be predicted

[Bengio et. al., 2003, A neural probabilistic language model]

Andrew Ng

Other context/target pairs

I want a glass of orange juice to go along with my cereal.

context

target

- Context: Last 4 words.

multiple options available

- 4 words on left & right
- Last 1 word
- Nearby 1 word

a glass of orange ? to go along with

orange ?

glass ?

skip gram

* building lanuage model, use last few words
* learning word embedding, use other contexts

Important Note: It is okay if we do poorly on P(t given c), the more important by-product of this task is that we learn a useful set of word embeddings.

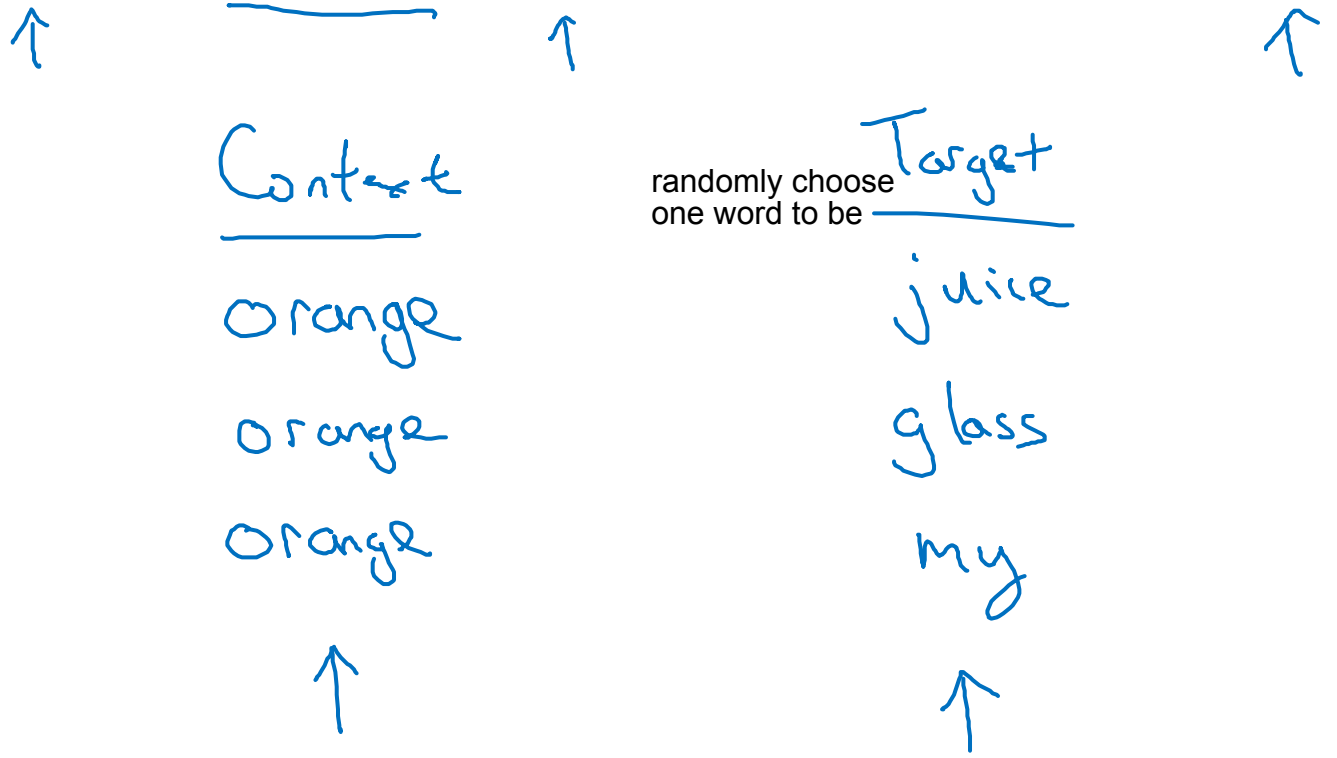
deeplearning.ai

NLP and Word Embeddings

Word2Vec

Skip-grams

I want a glass of orange juice to go along with my cereal.

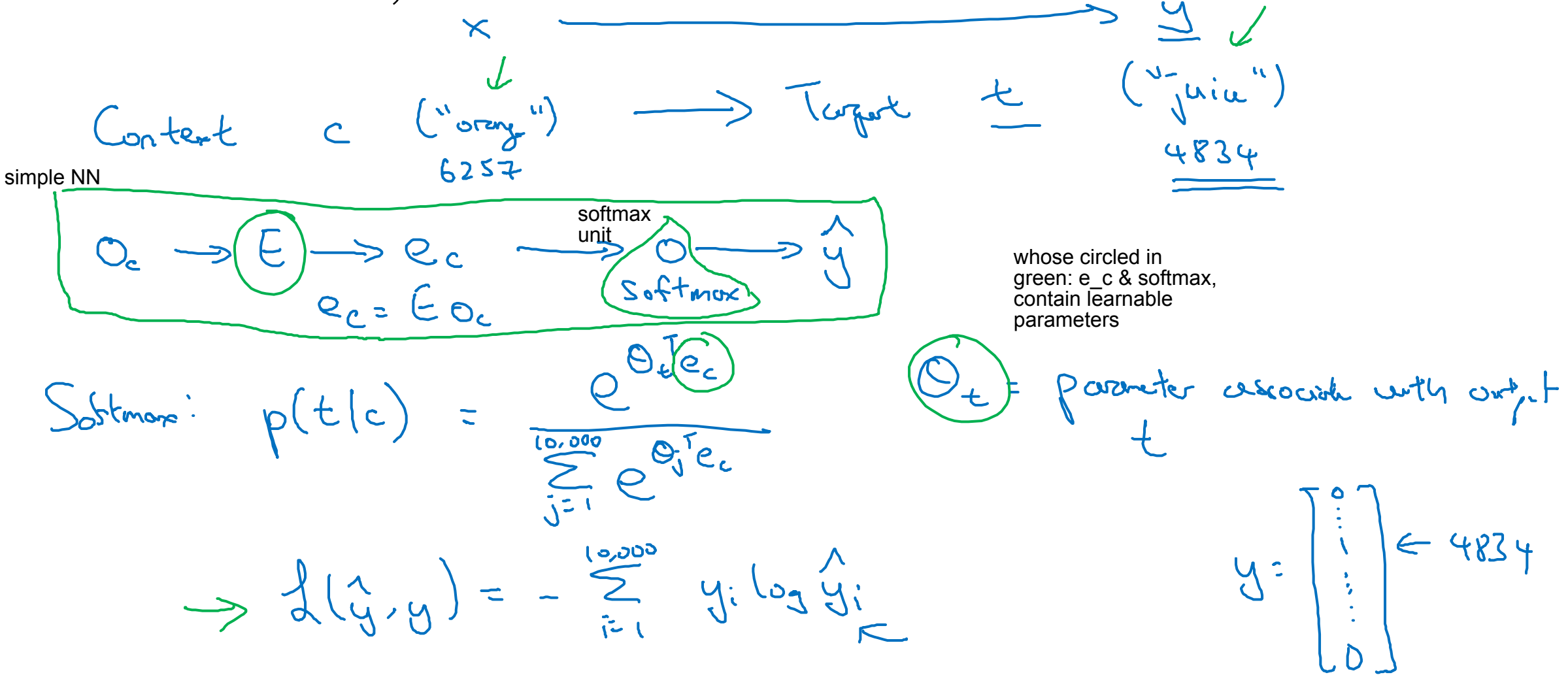


[Mikolov et. al., 2013. Efficient estimation of word representations in vector space.]

Skip Gram Model

Model

Vocab size = 10,000k



Andrew Ng

Problems with softmax classification

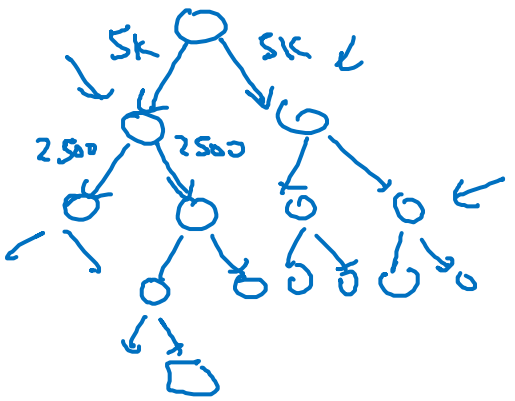
computational speed of softmax is one main problem

$$p(t|c) = \frac{e^{\theta_t^T e_c}}{\sum_{j=1}^{10,000} e^{\theta_j^T e_c}}$$

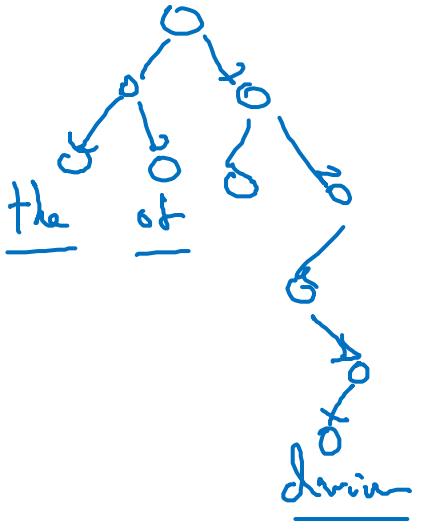
summing over all 10k, is computationally expensive

one solution: Hierarchical softmax.

$\log |V|$



in practice, often-words appear in earlier layers of graph



How to sample the context c?

not entirely random, some heurstics are used

c & t shall be chosen to be nearby words

t

c → t

frequently occuring words → the, of, a, and, to, ...

→ orange, apple, durian

P(c)

durian

- theta_t & e_c will have the same # dimensional vector as the word embedding
- they are both trained with optimization

Andrew Ng

deeplearning.ai

NLP and Word Embeddings

Negative sampling

Defining a new learning problem

- this is an update of Word2Vec model; to address the problem of having a huge NN, requiring lots of training data

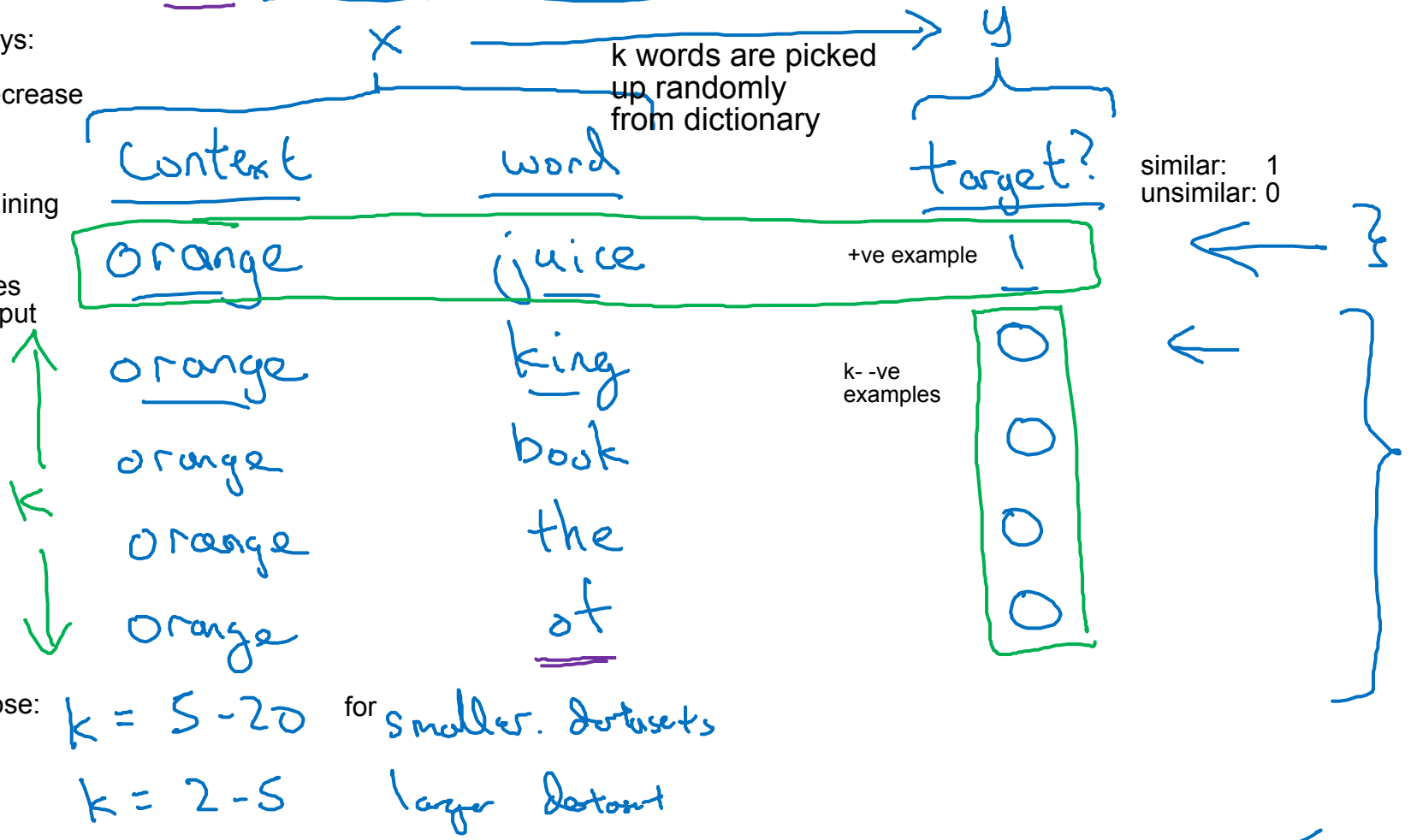
context ↓ target ↓

I want a glass of orange juice to go along with my cereal.

Authors tackled this problem in 2 ways:

1. subsampling frequent words to decrease # training examples.

2. modifying optimizationo through -ve sampling, which causes each training sample to update only small % of the huge model weights.
10,000 words in corpus * 300 features
a total of 3M in each of hidden & output layers



Check:
Word2Vec_McCormick notes

[Mikolov et. al., 2013. Distributed representation of words and phrases and their compositionality]

Model

Negative Sampling

we deliberately generate -ve examples

Softmax: $p(t|c) = \frac{e^{\theta_t^T e_c}}{\sum_{j=1}^{10,000} e^{\theta_j^T e_c}}$

● instead of

10,000-way softmax

possible target word

possible embedding word

logistic regression model

$P(y=1 \mid c, t) = \sigma(\theta_t^T e_c)$

context-target pair

context	word	target?
orange	juice	1
orange	king	0
orange	book	0
orange	the	0
orange	of	0

k

c t y

input: Orange 6257

$O_{6257} \rightarrow E \rightarrow e_{6257}$

one-hot embedding matrix embedding

juice?

king

10,000

● we update

10,000 binary classification problem

-ve examples k+1 +ve example

Andrew Ng

Selecting negative examples

context	word	target?
orange	juice	1
orange	king	0
orange	book	0
orange	the	0
orange	of	0

high rep. of such words

the, of, and, ...

one option: sampling according to empirical frequency of words in corpus

$P(w_i)$

best option:

$$\frac{f(w_i)^{3/4}}{\sum_{j=1}^{10,000} f(w_j)^{3/4}}$$

another option, yet not very representative

$$\frac{1}{|V|}$$

$|V|$ vocab size

deeplearning.ai

NLP and Word Embeddings

GloVe word vectors

GloVe (global vectors for word representation)

I want a glass of orange juice to go along with my cereal.

previously we used: c, t

X_{ij} = # times i appears in context of j.

(i = t, j = c; X_{ij} = c t)

$X_{ij} = X_{ji}$

It's a count capturing how many times i & j appear close to each other i.e +/- 10 words

[Pennington et. al., 2014. GloVe: Global vectors for word representation]

Model

- theta_i & e_j shall be initialized randomly at the beginning of training
- X_i_j : # times word j appears in the context of word i

optimizing by GD

Minimize

$$\sum_{i=1}^{10{,}000} \sum_{j=1}^{10{,}000} f(X_{ij}) \left(\theta_i^T e_j + b_i - b_j' - \log X_{ij} \right)^2$$

$\theta_i^T e_j$: θ is t, e is c

"$\theta_t^T e_c$"

$\log X_{ij}$: what if $X_{ij} = 0$? ($\log 0 = -\infty$)

how related i & j, as measured by how often they occur with each other

weighting term

heuristics can be used to define weighting term; giving weight to both.
more frequent weights shall be given more weight though

$f(X_{ij}) = 0$ if $X_{ij} = 0$.

"$0 \log 0$" = 0

- weighting fn f(.) must satisfy f(0)=0

it helps prevent learning only from extremely common word pairs.
It is not necessary that it satisfies this function

stop words: this, is, of, a, ...

more frequent

duration

less frequent

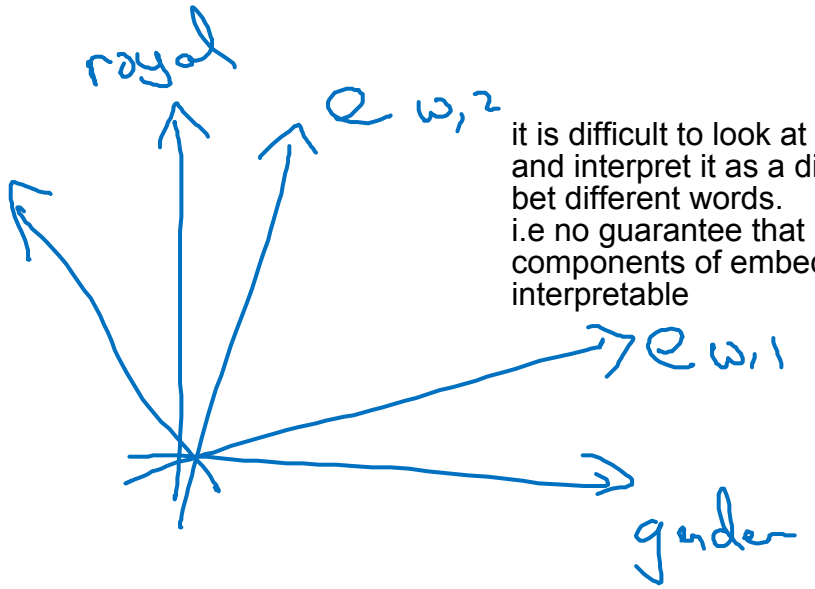
- θ_i, e_i are symmetric

$$e_w^{(final)} = \frac{e_w + \theta_w}{2}$$

Andrew Ng

A note on the featurization view of word embeddings

	Man (5391)	Woman (9853)	King (4914)	Queen (7157)	
Gender	−1	1	-0.95	0.97	←
Royal	0.01	0.02	0.93	0.95	←
Age	0.03	0.02	0.70	0.69	←
Food	0.09	0.01	0.02	0.01	←



it is difficult to look at a single row and interpret it as a direct relation bet different words.
i.e no guarantee that individual components of embeddings are interpretable

$$\text{minimize} \sum_{i=1}^{10,000} \sum_{j=1}^{10,000} f(X_{ij})\left(\theta_i^T e_j + b_i - b_j' - \log X_{ij}\right)^2$$

$\theta_i^T e_j \rightarrow (A\theta_i)^T (A^{-T} e_j) = \theta_i^T A^T A^{-T} e_j$

Andrew Ng

deeplearning.ai

NLP and Word Embeddings

Sentiment classification

Sentiment classification problem

x → y

The dessert is excellent. ★★★★☆

Service was quite slow. ★★☆☆☆

Good for a quick meal, but nothing special. ★★★☆☆

Completely lacking in good taste, good service, and good ambience. ★☆☆☆☆

- one challenge: no huge datasets

10,000 → 100,000 words

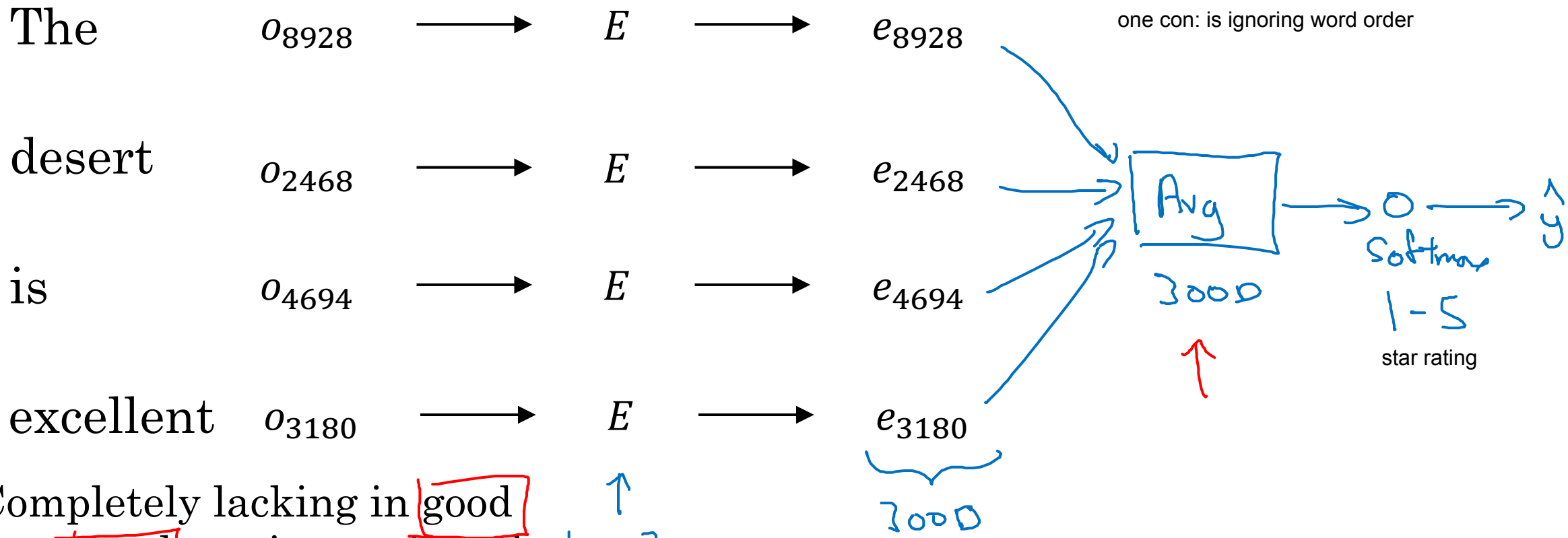
Simple sentiment classification model

The dessert is excellent

8928 2468 4694 3180

It's a process of summing all words in the example.

one con: is ignoring word order

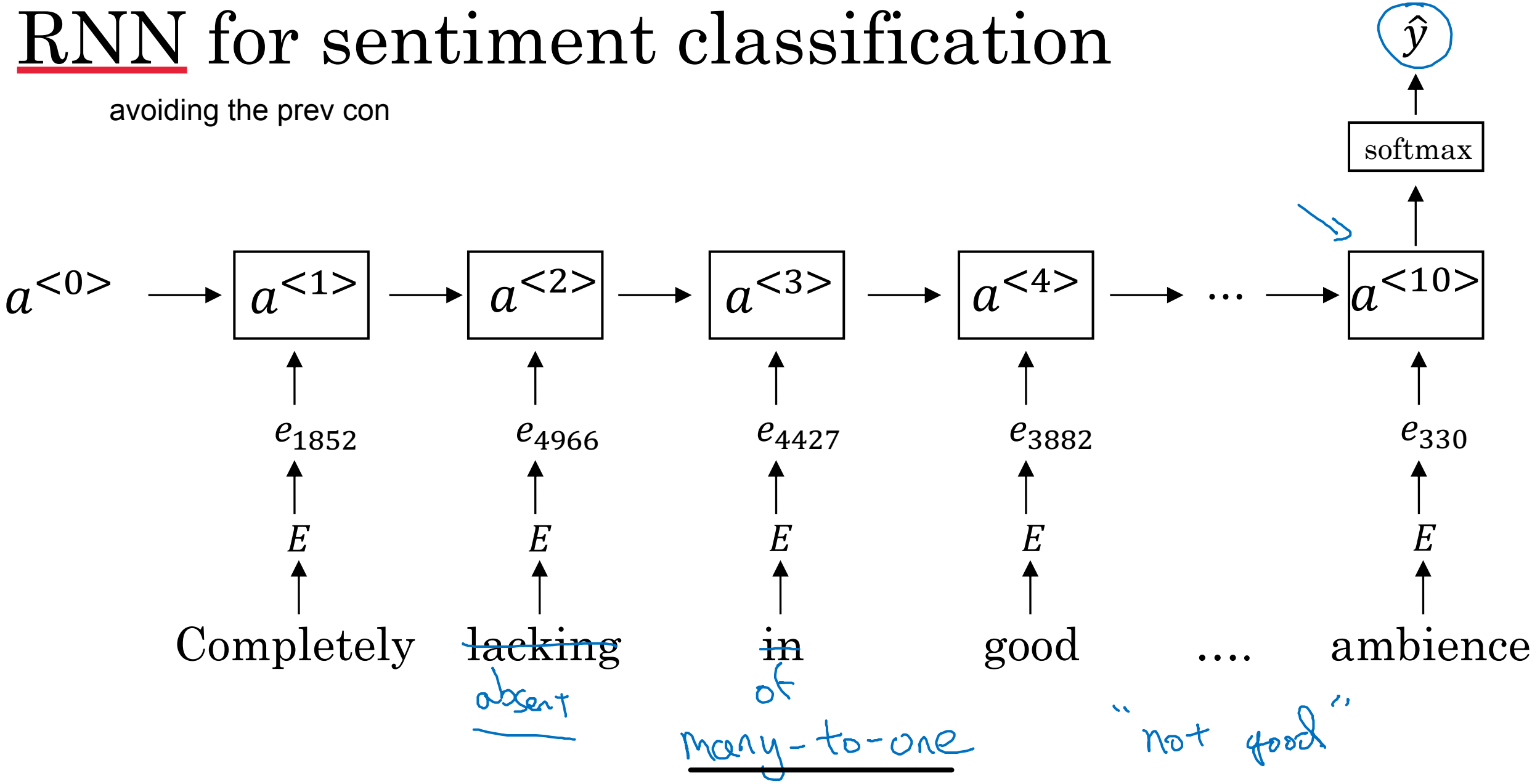


"Completely lacking in good taste, good service, and good ambience."

100 B words

RNN for sentiment classification

avoiding the prev con



Andrew Ng

deeplearning.ai

NLP and Word Embeddings

Debiasing word embeddings

The problem of bias in word embeddings

Man:Woman as King:Queen

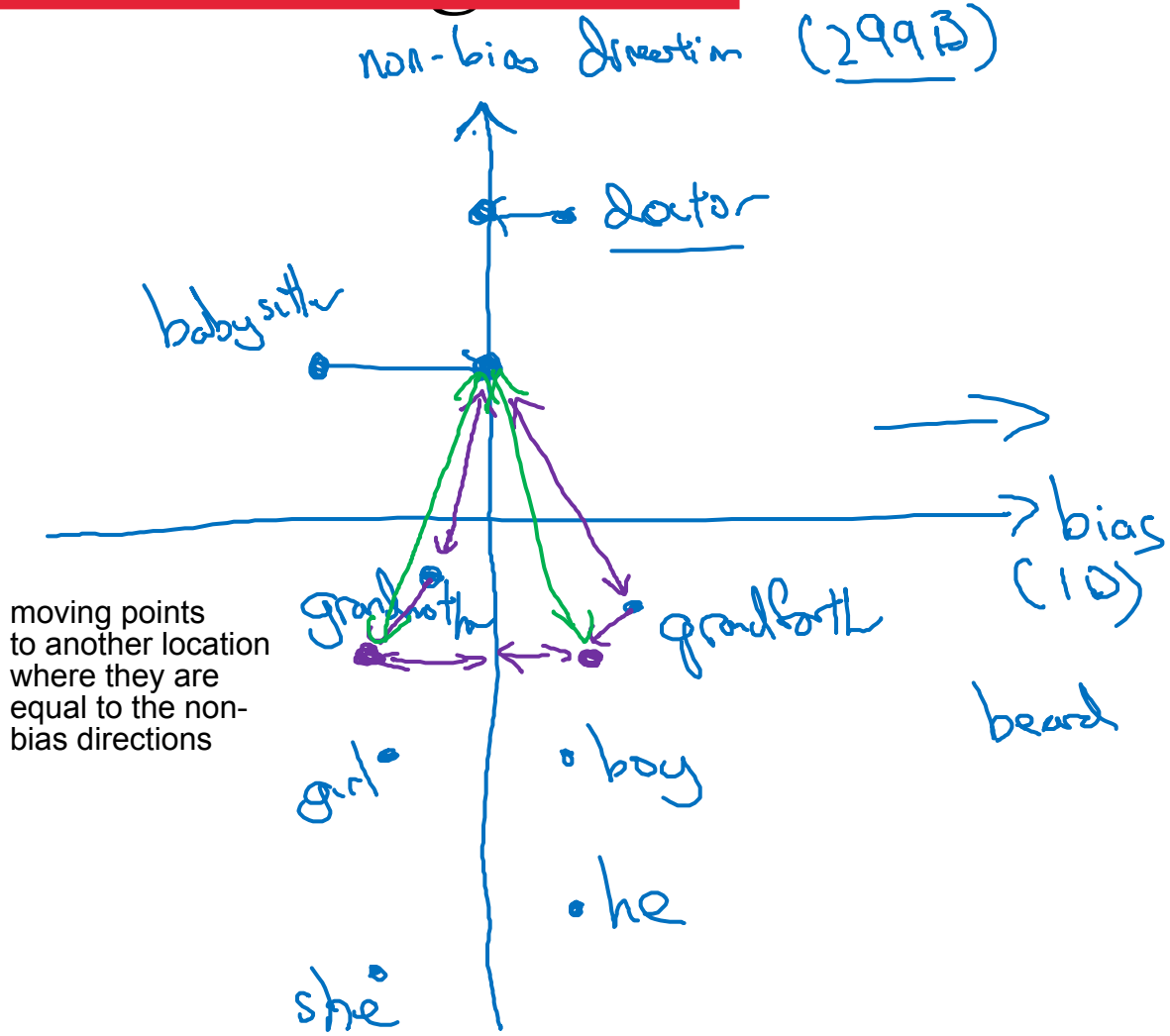
Man:Computer_Programmer as Woman:Homemaker ✗

Father:Doctor as Mother:Nurse ✗

✓ Word embeddings can reflect gender, ethnicity, age, sexual orientation, and other biases of the text used to train the model.

[Bolukbasi et. al., 2016. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings]

Addressing bias in word embeddings



moving points to another location where they are equal to the non-bias directions

1. Identify bias direction. e.g gender

$e_{he} - e_{she}$

$e_{male} - e_{female}$

⋮

average

words like grandpa & grandma are by construction gender included, unlike doctor or babysitter

2. Neutralize: For every word that is not definitional, project to get rid of bias.

such equalized pairs (grandma, grandpa) shall have the same similarity (distance) with gender neutral words (doctor)

3. Equalize pairs.

→ grandmother − grandfather
girl boy

[Bolukbasi et. al., 2016. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings]