Sequence Model

1° Recurrent Neural Networks

2° Word Embedding : Word2Vec & GloVe

3° Sequence models & Attention mechanism

RNN

Notation • intro

e.g: $x^{<1>}$ Harry $x^{<2>}$ Potter $x^{<3>}$ and $x^{<4>}$ Jie $x^{<5>}$ invented a new spell $x^{<t>}$

$T_x = 9$ x : ↓

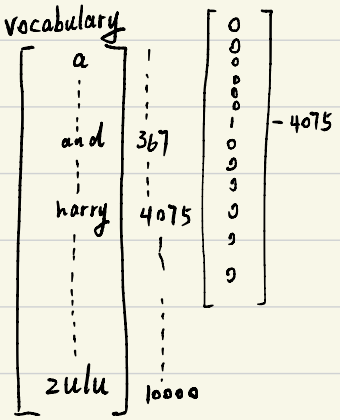
expected y : 1 1 0 1 0 0 0 0

$T_y = 9$ $y^{<1>}$ $y^{<2>}$ .. .. $y^{<9>}$

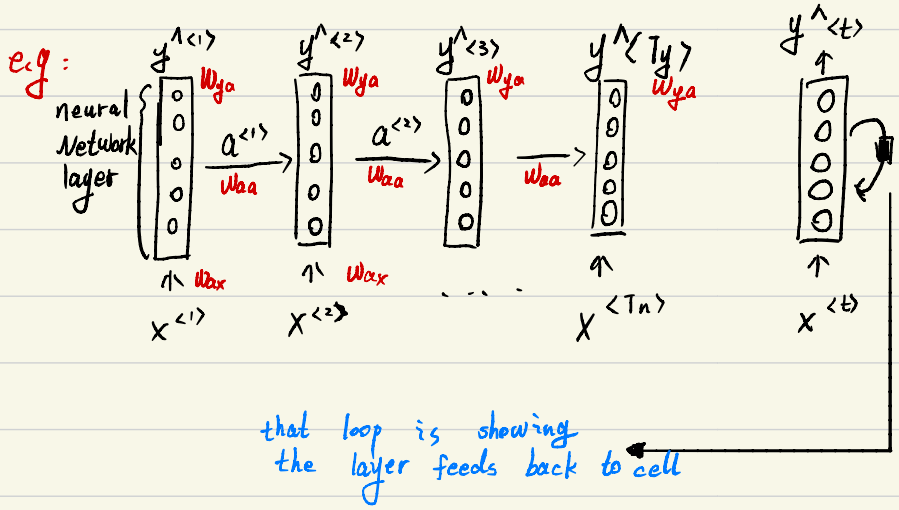
$T_x^{(i)}$: the length of training list

• representing words (one-hot-representation)

e.g: Harry ↘ $x^{<1>}$ and $x^{<2>}$ Jie .. invented .. a .. spell ..



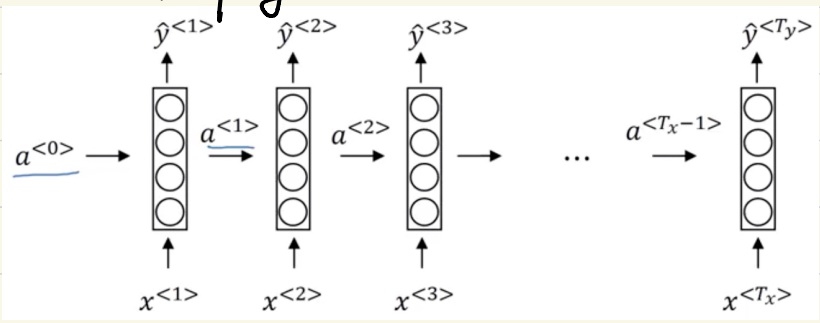
Recurrent Neural Network Model



Limitation for RNN :

the prediction for inputs uses from input earlier not input later in sequence

• Forward Propagation



$a^{<0>} = \vec{0}$

$a^{<1>} = g(W_{aa}\, a^{<0>} + W_{ax}\, x^{<1>} + \boxed{b_a})$ ← constant in linearly regression

$\hat{y}^{<1>} = g(W_{ya}\, a^{<1>} + b_y)$

$a^{<t>} = g(W_{aa}\, a^{<t-1>} + W_{ax}\, x^{<t>} + b_a)$

$\hat{y}^{<t>} = g(W_{ya}\, a^{<t>} + b_y)$

↓ simplified RNN Notation

$a^{<t>} = g(W_a [a^{<t-1>}, x^{<t>}] + b_a)$

$[a^{<t-1>}, x^{<t>}] = \begin{bmatrix} a^{<t-1>} \\ x^{<t>} \end{bmatrix}$ (100 / 10000 → 1100)

$\hat{y}^{<t>} = g(W_y\, a^{<t>} + b_y)$

$\begin{bmatrix} W_{aa} \mid W_{ax} \end{bmatrix} = W_a$ (100 × 1100)

Backpropagation through time

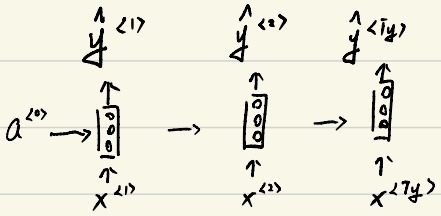
Loss function

$$\mathcal{L}^{<t>}(\hat{y}^{<t>}, y^{<t>}) = -y^{(t)} \log \hat{y}^{<t>} - (1-y^{<t>}) \log (1-\hat{y}^{<t>})$$

$$\mathcal{L}(\hat{y}, y) = \sum \mathcal{L}^{<t>}(\hat{y}^{<t>}, y^{<t>})$$

reverse of Forward propagation

Different type of RNN

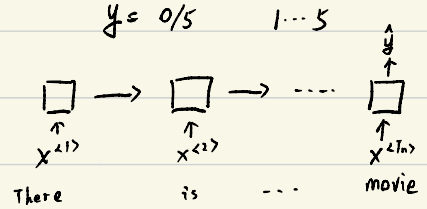


many - to - many

e.g: Setimant classification

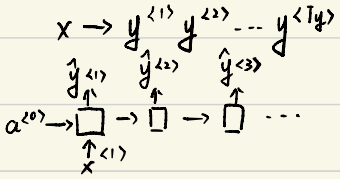
x = text

y = 0/5 1 ··· 5



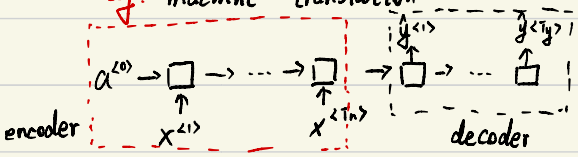
many - to - one

e.g: music generation



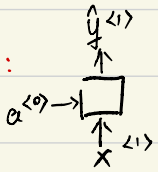
one - to - many

e.g: machine translation



many - to - many

e.g:



one to one

Languge modeling with RNN

- training set : large corpus of english text

Tokenize (标记化)

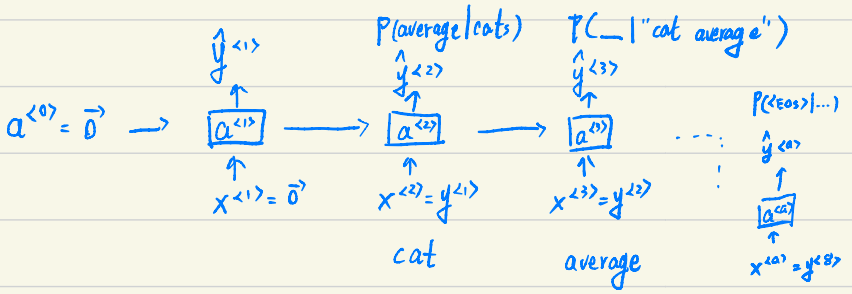
e.g: - Cat average 15 hours of sleep a day. <eos>

$y^{<1>}$ $y^{<2>}$ $y^{<3>}$ ... $y^{<8>}$ $y^{<9>}$

The Egyptian ~~Mau~~ is a bread of cat <eos>

<unk>

→ RNN model



$P(y^{<1>}, y^{<2>}, y^{<3>}) = P(y^{<1>}) P(y^{<2>}|y^{<1>}) P(y^{<3>}|y^{<1>}, y^{<2>})$

Sampling novel sequences

character-level language model

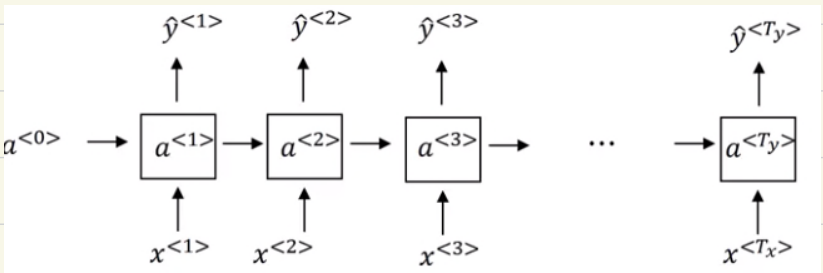
no worry with <UNK>; too long

Sequence generation

Vanishing gradients with RNN (梯度消失)

The cat, which ate ... , was full

The cats, which ate ... , were full



exploding gradients → sort address by gradient clipping

Gated Recurrent Unit (GRU) solve vanishing gradient with RNN

C = memory cell

$c^{<t>} = a^{<t>}$

$\tilde{c}^{<t>} = g(W_c[c^{<t-1>}, x^{<t>}] + b_a)$

$\Gamma_u = \sigma(W_u[c^{<t-1>}, x^{<t>}] + b_a)$

↑ update

$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + (1-\Gamma_u) * c^{<t-1>}$

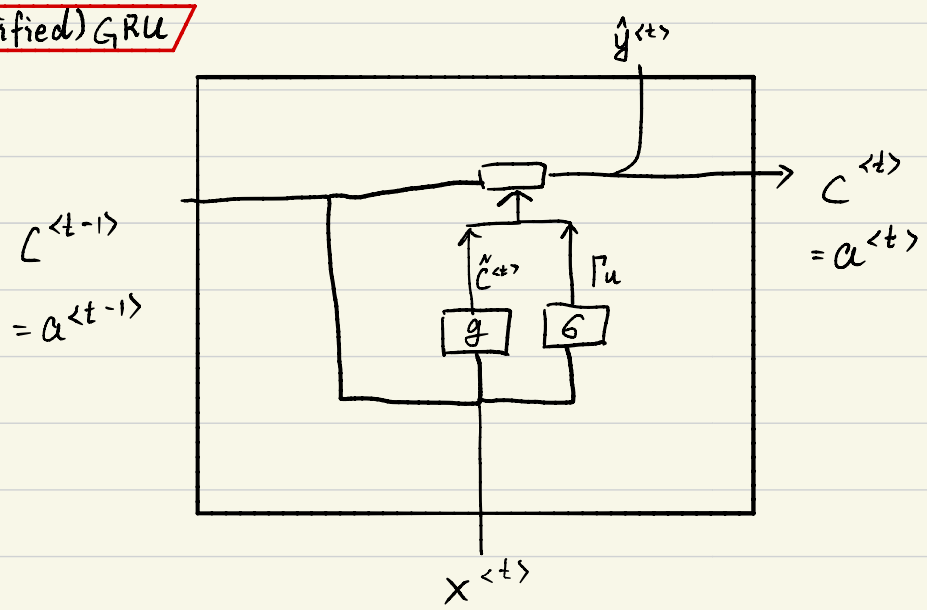
e.g.:

The cat, which ..., was full

$\rightarrow c^{<t>} = 1$ ---- $= 1$

$\Gamma_u = 1$ (determine if update to 1)

(simplified) GRU



full GRU

$\tilde{c}^{<t>} = g(W_c[\Gamma_r * c^{<t-1>}, x^{<t>}] + b_c)$

$\Gamma_r = \sigma(W_r[c^{<t-1>}, x^{<t>}] + b_c)$

LSTM long short Term Memory

$$\tilde{C}^{\langle t \rangle} = \tanh(W_c [a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_c)$$

update → $\Gamma_u = \sigma(W_u [a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_u)$

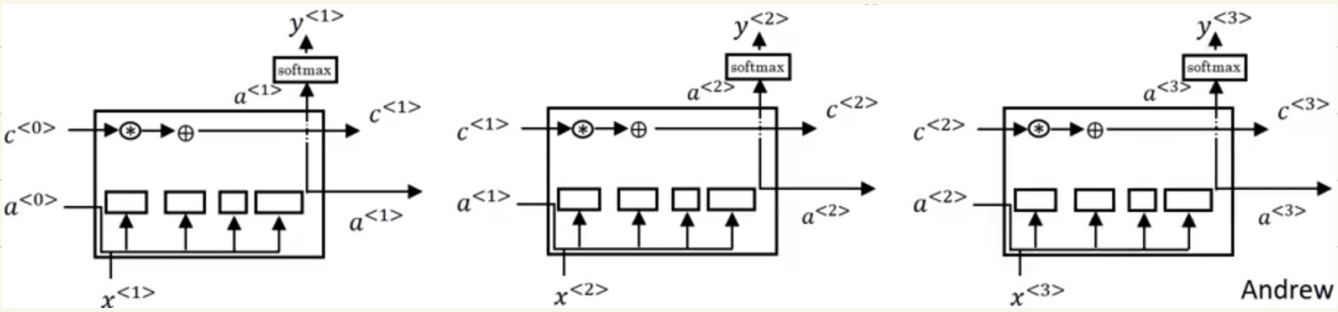
forget → $\Gamma_f = \sigma(W_f [a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_f)$

$\Gamma_o = \sigma(W_o [a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_o)$

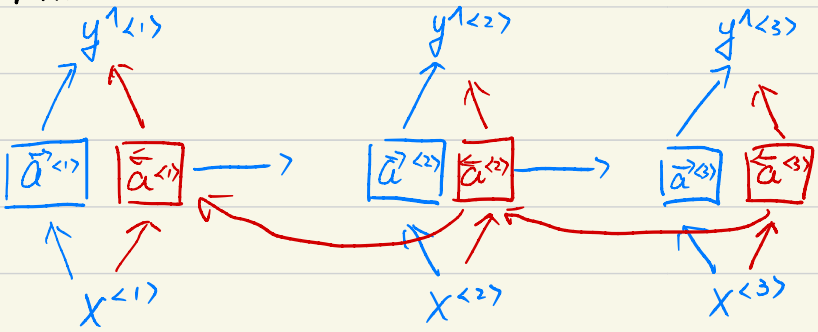
↑ output

$$c^{\langle t \rangle} = \Gamma_u * \tilde{C}^{\langle t \rangle} + \Gamma_f * c^{\langle t-1 \rangle}$$

$$a^{\langle t \rangle} = \Gamma_o * \tanh c^{\langle t \rangle}$$



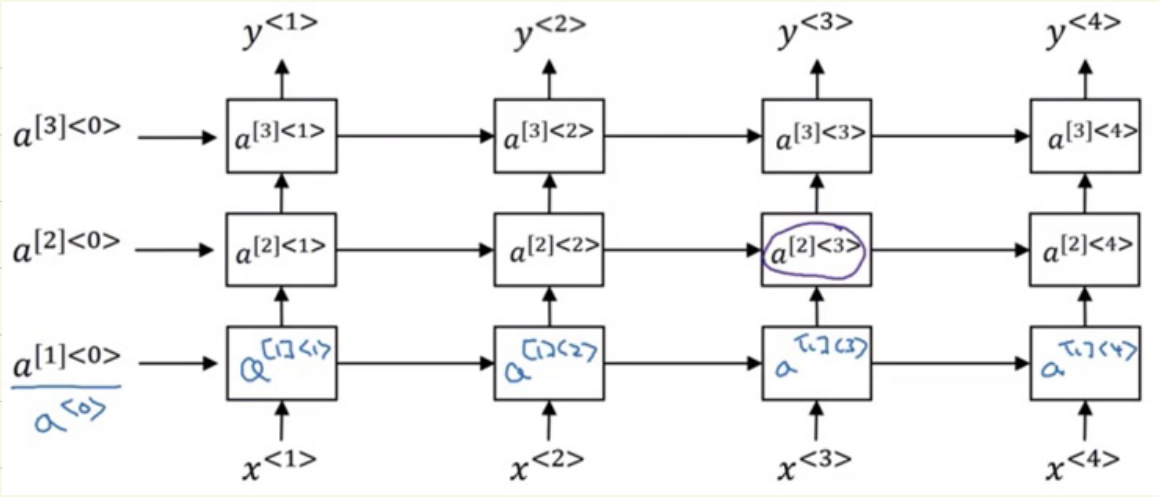
Bidirectional RNN



allow $a^{\langle 2 \rangle}$ to take front and back information

need all data (when speech, need complete and then can translate)

Deep RNN



$$a^{[2]<3>} = g\left(W_a^{[2]}\left[a^{[2]<2>}, a^{[1]<3>}\right] + b_a^{[2]}\right)$$

Word embedding 词向量

1- hot representation $|V| = 10\,000$

Man	Woman	king	queen	apple	Orange
(5391)	(9853)	(4914)	(7157)	(456)	(6257)

5391 → $\begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$ 9853 → $\begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$

Featurized representation

	e_{5391}	e_{9853}				
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

e.g: I want a glass of orange ___
apple ___

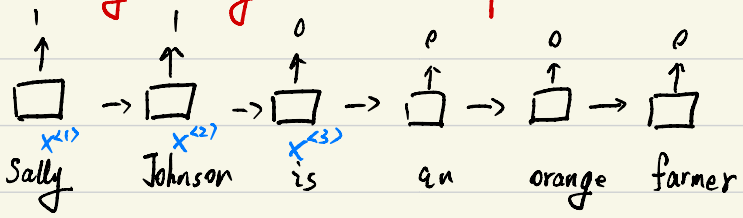
most features are similar

Visualizing word embeddings

t-SNE

Using word embeddings

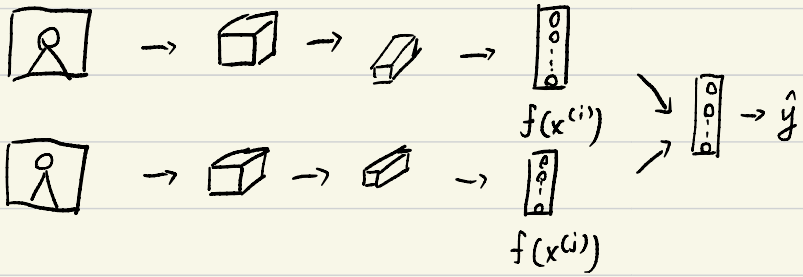
e.g: Named entity recognition example



transfer learning and word embedding

1. Learn word embedding from large text corpus (1-100B words)
2. transfer embedding to new task with smaller training set
3. Optional: Continue to finetune the word embeddings with new data.

relation to face encoding (embedding)



Properties of word embeddings

analogies

	Man	Women	king	queen	apple	orange
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
	e_{5391}	e_{9853}				

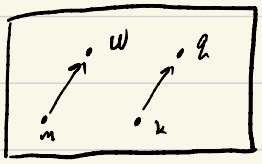
How to get

man -> woman => king -> ?

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

So $e_{man} - e_{woman} \approx e_{king} - e_{queen}$

analogies using word vectors



300 D

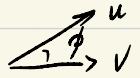
$e_{man} - e_{woman} \approx e_{king} - e_{w}$

$\arg\max_{w} \; sim(e_w, e_{king} - e_{man} + e_{woman})$

Cosine similarity

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

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



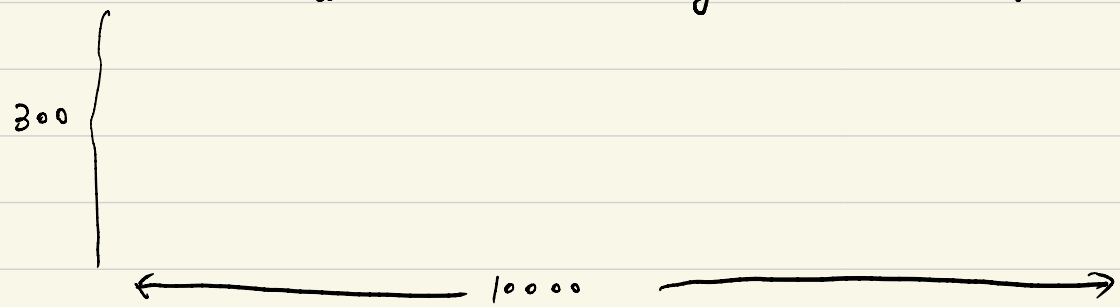
e.g: Ottawa : Canada as Nairobi : kenya

Big : Bigger as Tall , Taller

Embedding matrix

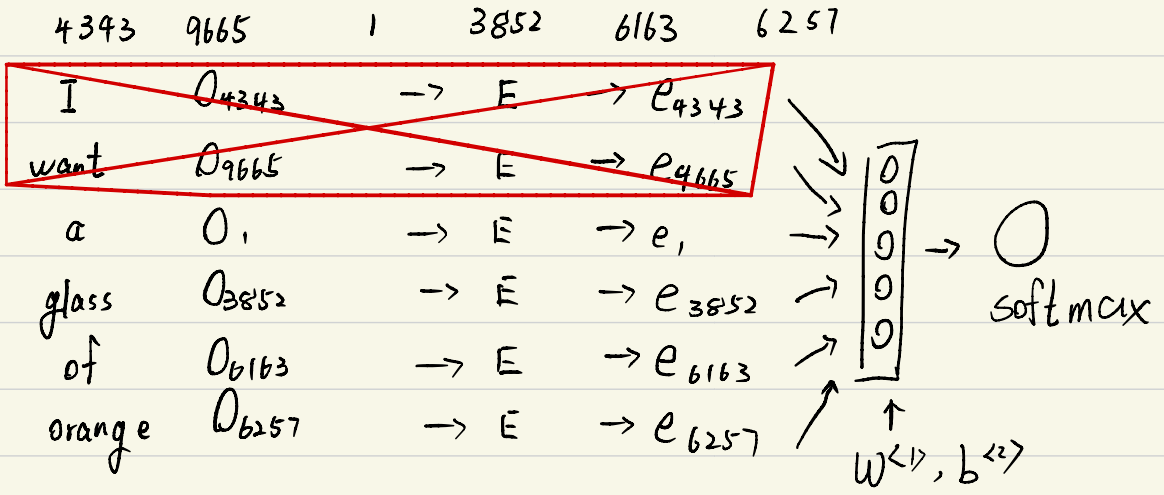
6757

a aaron ... orange ... zulu <UNK>



Learning word embeddings

e.g: I want a glass of orange ____.



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

context: Last 4 words

4 words or left & right

a glass of orange ? to go along with

Last 1 word orange ?

Nearby 1 word (skip gram) glass ?

Word2Vec

skip gram Model

Vocab size = 10 000 k

Context c ("orange") 6257 → Target t ("juice") 4834

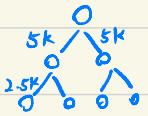
$O_c \rightarrow E \rightarrow e_c \rightarrow O \rightarrow \hat{y}$

O: softmax

softmax : $P(t|c) = \frac{e^{\theta_t^T e_c}}{\sum_{j=1}^{10000} e^{\theta_j^T e_c}}$ (need long computation time)

↑ solve that

Hierarchical softmax



log lvl

Negative Sampling

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

k = 5 - 20 smaller dataset

k = 2 - 5 larger dataset

model

$$\text{softmax} : P(t|c) = \frac{e^{\theta_t^T e_c}}{\sum_{j=1}^{10000} e^{\theta_j^T e_c}}$$

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

selecting negative example

$$P(w_i) = \frac{f(w_i)^{3/4}}{\sum_{j=1}^{10000} f(w_j)^{3/4}}$$

GloVe word vectors (global vectors for word representation)

e.g: I want a glass of orange juice to go along

with my cereal

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

minimize $\sum_{i=1}^{10000} \sum_{j=1}^{10000} f(X_{ij}) \left(\theta_i^T e_j + b_i + b_j' - \log X_{ij}\right)^2$

$f(X_{ij})$: weighting term

sentiment classification

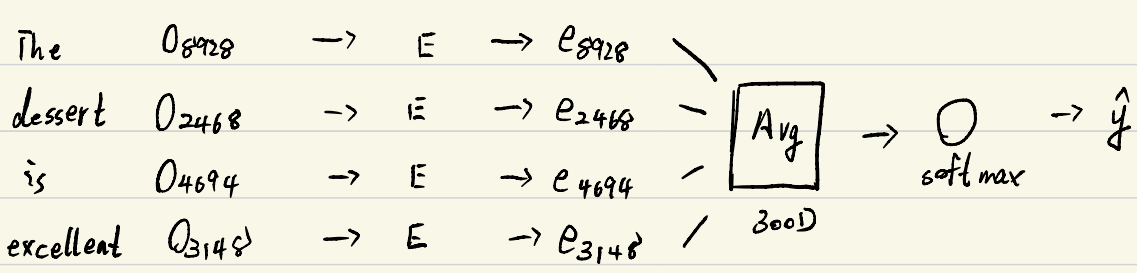
x	→	y
The dessert is excellent		☺
Service was quite slow		☹

challenge: × have a huge label data set

Simple sentiment classification model

The dessert is excellent

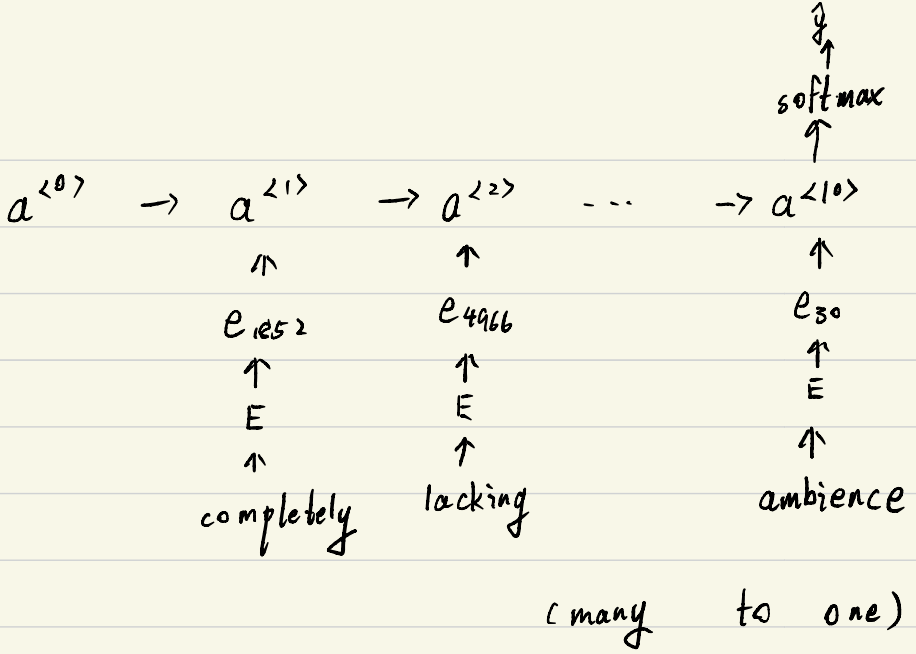
8928 2468 4694 3148



But not good at this example

"Completely lacking in good taste, good service"

RNN for sentiment classification



(many to one)

Debiasing word embedding

the problem of bias in word embedding

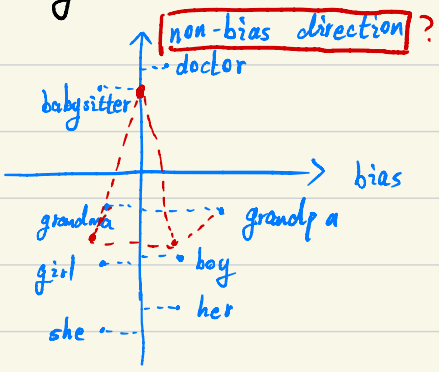
Man:Woman as King:Queen

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.

addressing bias in word embeddings



1. identify bias direction
   - $e_{he} - e_{she}$
   - $e_{male} - e_{female}$
2. neutralize: For every word that is not definitional, project to get rid of bias
3. equalize pairs