SEQUENCE MODELS

Why Sequence Models

Examples of seq data
- Speech recognition
- music generation
- sentiment classification
- DNA seq analysis

.

* Notation

example

x: Harry Potter and Hermione Granger invented a potion

y: 1 1 0 1 ... 0 0 0

indexing: $x^{<1>}$ $x^{<2>}$ $x^{<3>}$ ... $x^{<t>}$ ... $x^{<8>}$

$y^{<1>}$ $y^{<2>}$ - - - - - $y^{<t>}$ ... $y^{<8>}$

$T_y = 9$

$x^{(i)<t>}$: t^{th} element of i^{th} training example

$T_x^{(i)}$: length of i^{th} training example

Vocabulary: [a, aaron, ... harry ... potter ... Zulu]

all the possible ~~words~~ words. 10,000

are represented as one-hot form

x = [0, 0, ..., 1 → 4075, 0, 0] (harry)

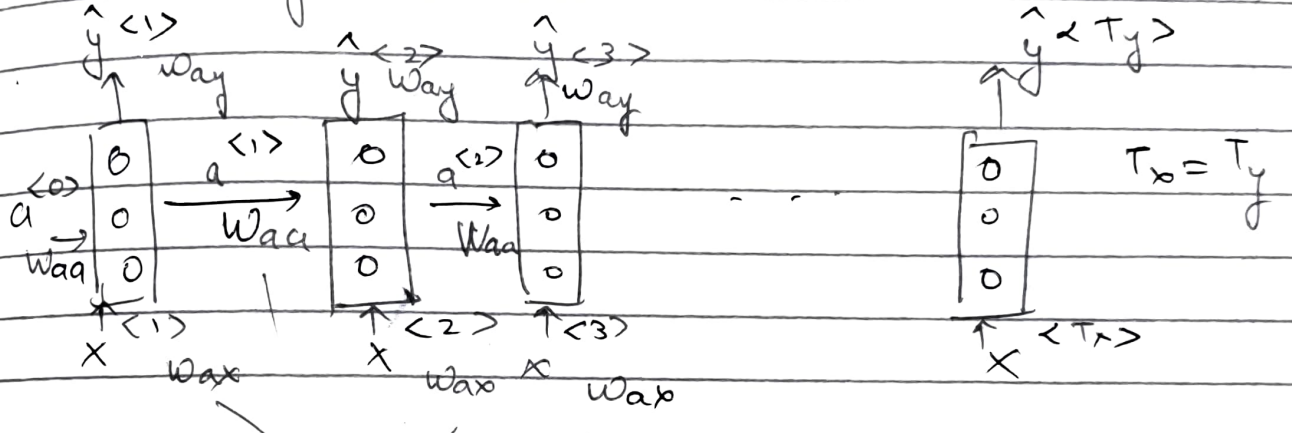
[0, 0, ..., 1 → 6830, 0] (potter)

Recurrent Neural Network Model

Problems in a Std network

- Input, outputs can be of diff lengths in diff examples
- doesnt share features learned across diff positions of text

RNN general arch:



- forward propagation

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

$a^{<1>} = g(W_{aa} a^{<0>} + W_{ax} x^{<1>} + b_a)$ ← tanh/ReLU

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

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

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

$$a^{\langle t \rangle} = g(W_a [a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_a)$$

Shape

$W_{aa} = (100, 100)$

$W_{ax} = (100, 10000)$

$W_a [a^{\langle t-1 \rangle}, x^{\langle t \rangle}] = \begin{bmatrix} a^{\langle t-1 \rangle} \\ x^{\langle t \rangle} \end{bmatrix}$ (100 / 10000)

stack vertically

$[W_{aa} \mid W_{ax}]$ (100 / 10000)

stack horizontally

$$[W_{aa} \mid W_{ax}] \begin{bmatrix} a^{\langle t-1 \rangle} \\ x^{\langle t \rangle} \end{bmatrix} = W_{aa} a^{\langle t-1 \rangle} + W_{ax} x^{\langle t \rangle}$$

* Back prop

$$L^{\langle t \rangle}(\hat{y}^{\langle t \rangle}, y^{\langle t \rangle})$$

$$L(\hat{y}, y) = \sum_{1}^{t} L^{\langle t \rangle}(\hat{y}^{\langle t \rangle}, y^{\langle t \rangle})$$

* RNN Types

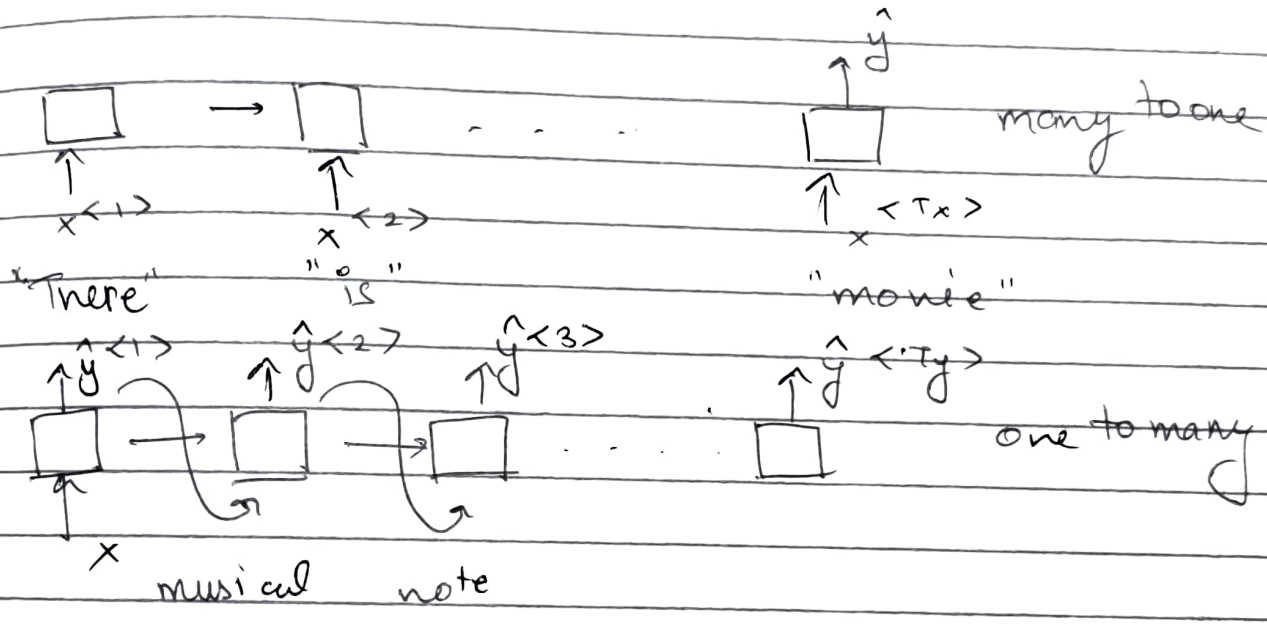
Tx does not always equals to Ty

eg: sentiment classification:

"There is nothing to like in the movie"

converted to a star rating.

- generating music from initial notes.



* Language Model & Seq generation

Training set: large corpus of english text

Tokenize

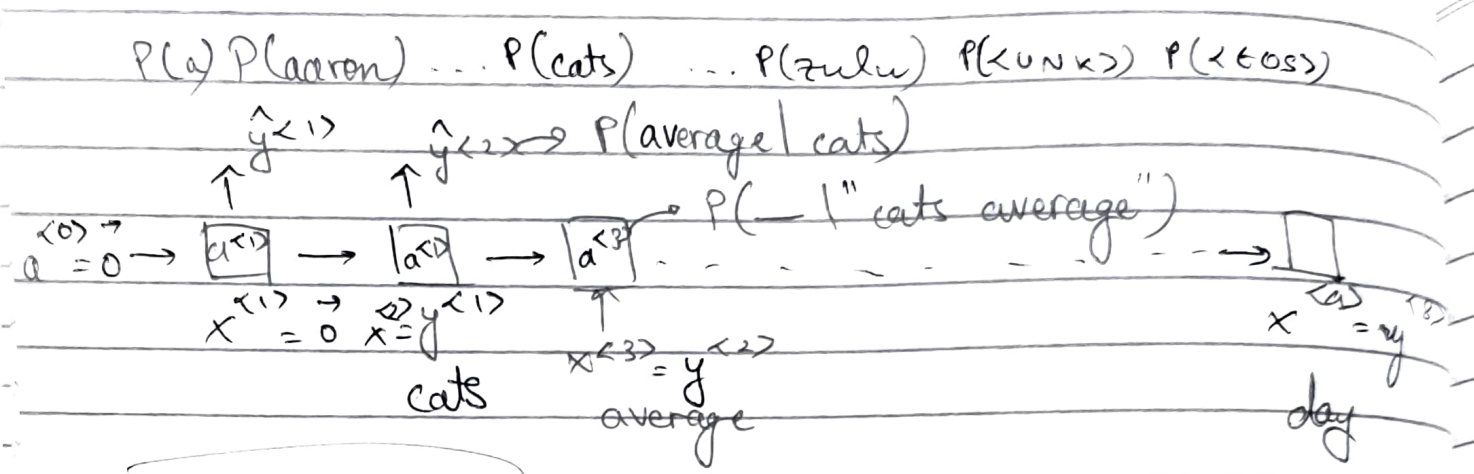
"Cats average 15 hours of sleep a day." <EOS>

onehot tokens $y^{<1>}$ $y^{<2>}$ ...

"The Egyption Mau is a breed of cat"

bisnot in the vocabulary

replace with token <UNK>



⇒ (Cats average 15) hours of sleep a day <EOS>

$$L(\hat{y}^{<t>}, y^{<t>}) = -\sum_i y_i^{<t>} \log \hat{y}_i^{<t>}$$

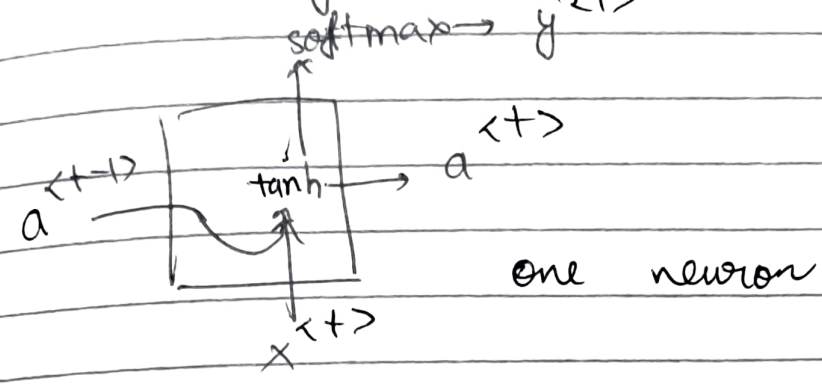
* Sampling Novel Sequences
  - Word
  - character

* Vanishing Gradients in RNNs

In case of deep RNNs the backpropagation gradients obtained from the later layers may not significantly affect the earlier layers.

*GRU - Grated Recurrent Unit

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



The cat, which already ate..., was full
The cats, which already ate..., were full

Lets introduce c : memory cell
which will store cat/cats

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

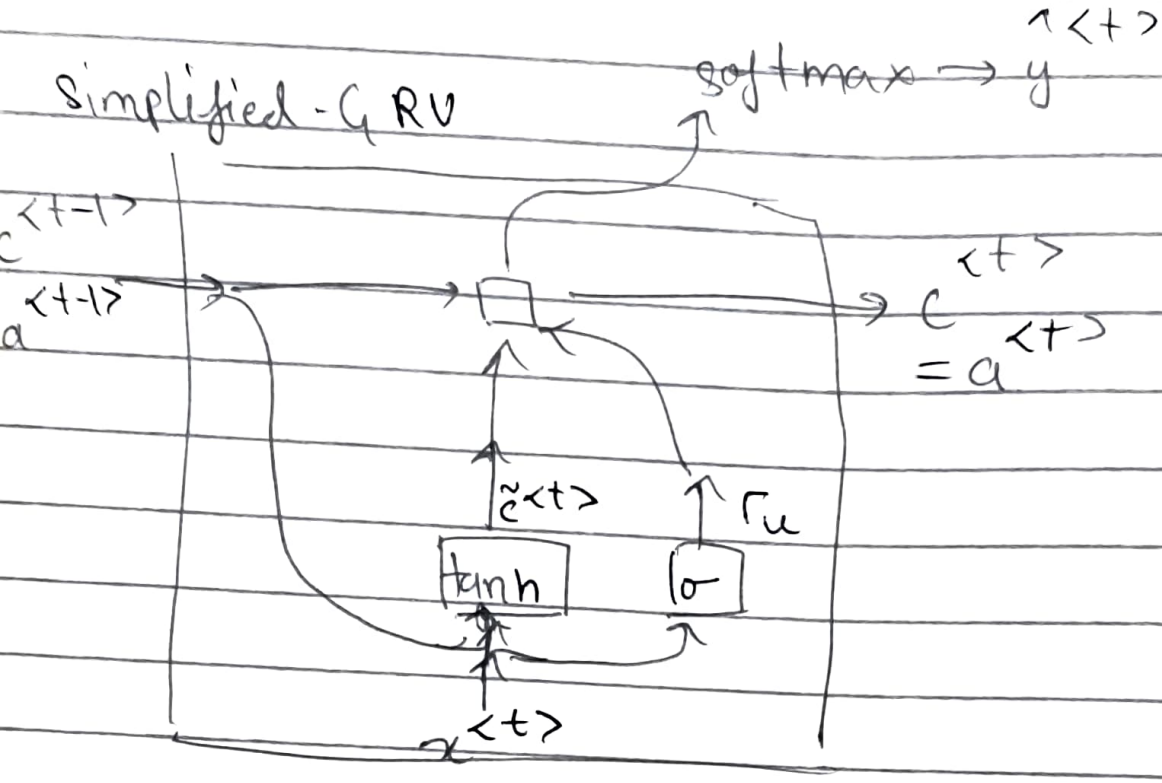
$$\tilde{c}^{<t>} = \tanh(W_c [c^{<t-1>}, x^{<t>}], b_c)$$

gamma $\Gamma_u = \sigma(W_u [c^{<t-1>}, x^{<t>}], b_u)$
↑ update

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

element wise multiplication

Simplified - GRU



* Long Short Term Memory (LSTM)

$$\tilde{c}^{\langle t \rangle} = \tanh(w_c [a^{\langle t-1 \rangle}, x^{\langle t \rangle}]_0 + b_c)$$

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

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

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

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

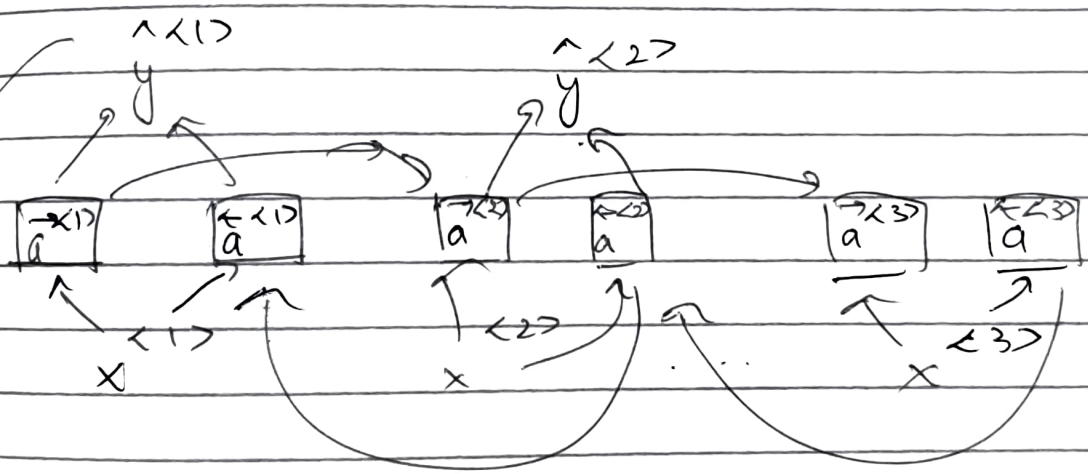
$$a^{\langle t \rangle} = \Gamma_o * \tanh(c^{\langle t \rangle})$$

* Bidirectional RNN

He said "Teddy bears are on sale!"
↳ not person

He said "Teddy Roosevelt was a great pr resident"
person

We need more info than the previous words (in both cases - same) to determine if Teddy is a person or not. It will be obtained from the later part of the sentance.



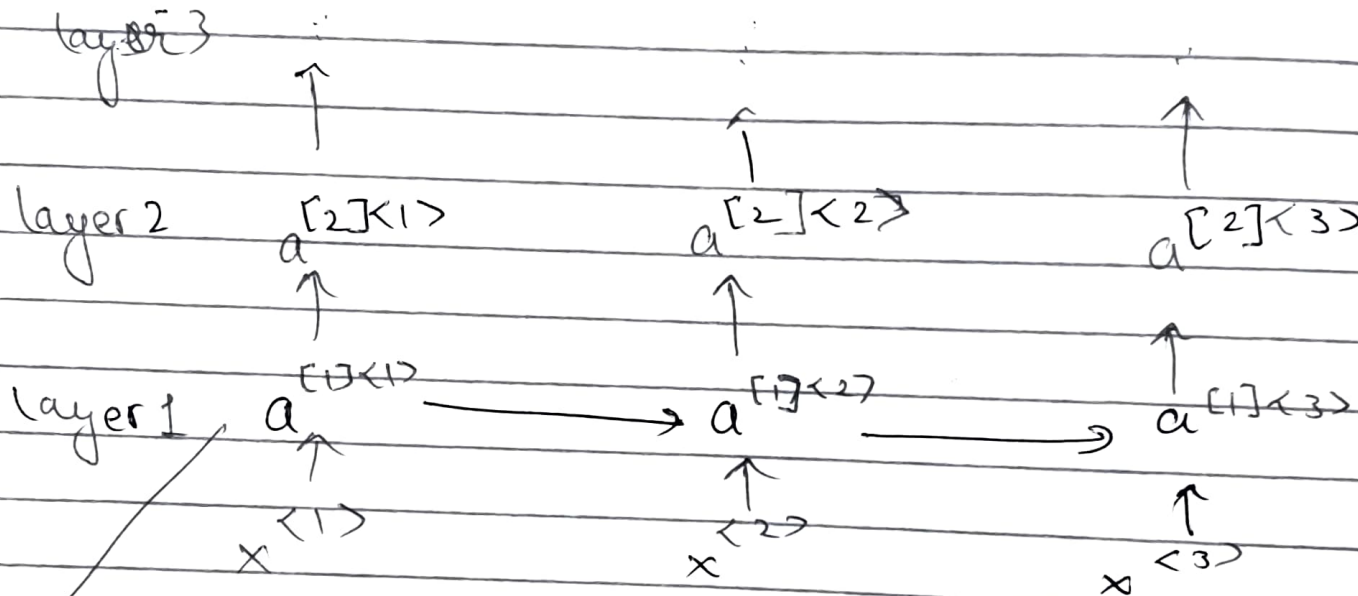
$$\hat{y}^{<t>} = g(W_y [\overrightarrow{a}^{<t>}, \overleftarrow{a}^{<t>}] + b_y)$$

It is often used with LSTM for NLP

You need to process entire data seq before making any prediction

$\hat{y}^{\langle 2 \rangle}$

* Deep RNNs



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

can be RNN, GRU, LSTM

* Word Embeddings

problem with one hot representation

'apple' = 456 $\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$ 'orange' = 6257 $\begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$

I want a glass of apple juice

I want a glass of orange ____

there is a ~~tot~~ large distance (less similarity) in apple & orange according to one hot representation.

Featurized representations

	Man 5391	Woman 9853	King 4914	Queen 7157	Apple 456	Orange 6257
Gender	-1	1	-0.95	0.97	0.0	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						
:						
:						

300 feature word embedding

embedding ≈ encoding

These 300 dimensions can be reduced to 2D using to-SNE

* Using Word Embeddings

Joe David is an apple farmer
Don Pablo is an orange farmer
↳ it is easy to learn that Don Pablo is a person

Michael Scott is a kiwi cultivator
↳ never seen before words

What to do now?
Use transfer learning

1. Learn word embeddings from a larger text corpus (1-100B words)
or download pre trained embedding

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

Properties of word Embeddings: Analogies

Man : Woman :: King : ?

from the previous table

$e_{man} = \begin{bmatrix} -1 \\ 0.01 \\ 0.03 \\ 0.04 \end{bmatrix}$ $e_{woman} = \begin{bmatrix} 1 \\ 0.02 \\ 0.02 \\ 0.01 \end{bmatrix}$

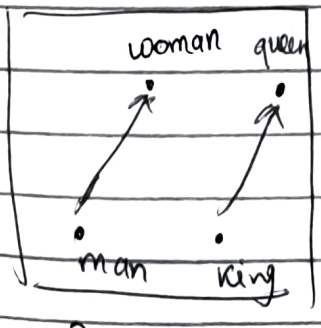
$e_{man} - e_{woman} \approx \begin{bmatrix} -2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

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

find e_w such that

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

$e_w = e_{queen}$



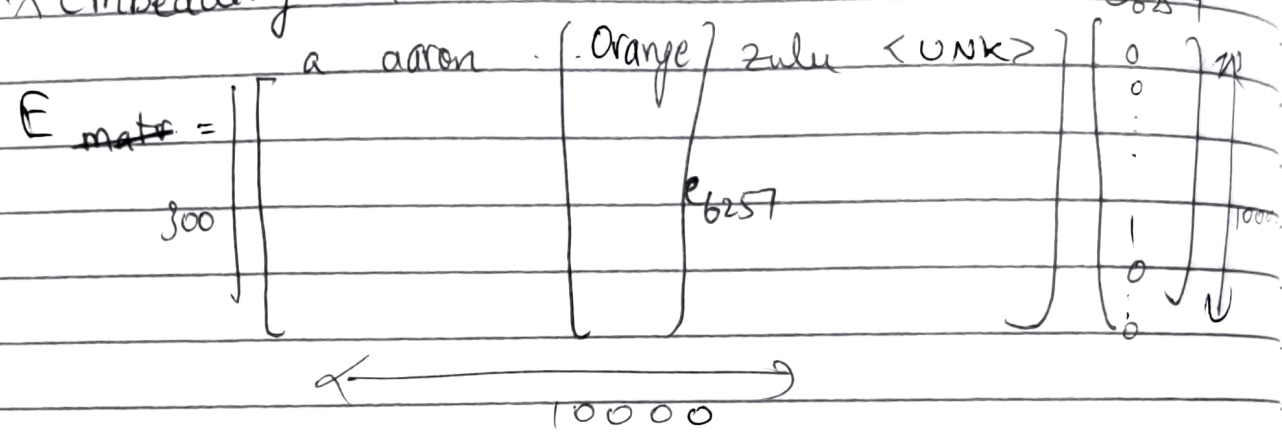
find word w:

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

Cosine similarity

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

* Embedding Matrix



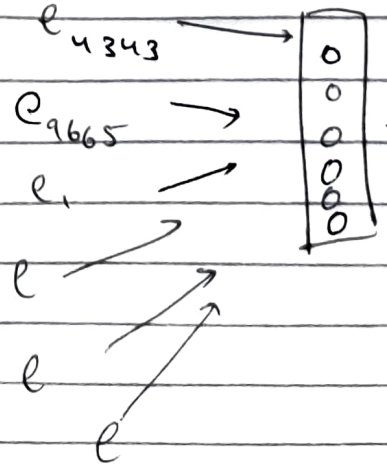
$E \cdot O_{6257} = e_{6257}$

$(300, 10k) \times (10k, 1)$

* Neural Language Model

I want a glass of orange juice ?

4343 9665 1 3852 6163 6257



*Word 2 Vec

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

context	target
orange	glass
orange	juice
orange	my

context c ("orange") → target t ("juice")

6257 4834

$O_c \rightarrow E \rightarrow e_c \rightarrow O \xrightarrow{\text{softmax}} \hat{y}$

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

θ_t : parameter associated with t

$$L(\hat{y}, y) = -\sum_{i=1}^{10K} y_i \log \hat{y}_i$$

*Negative Sampling

context	word target	target
orange	juice	1
orange	king	0
orange	of	0