Sequence Models:

⊕ Recurrent Neural Networks (RNN)

* Seq. Models :

Speech recognition	$X \rightarrow y$	input X is a sequence
Music generation	$X \rightarrow y$.	
DNA sequence analysis	AGCCTCCAGCCTCC ...	
Machine Translation		
Video activity recognition		
Name entity recognition...		

* Notations

Ex:

x :	Harry	Potter	&	Hermoine	granger	invented	a	new	spell.
	$x^{<1>}$	$x^{<2>}$	$x^{<3>}$	- - - -		$x^{<t>}$	- - - -		$x^{<9>}$ $T_x = 9$
y :	1	1	0	1	1	0	0	0	0
(persons name)	$y^{<1>}$	$y^{<2>}$	- - - -			$y^{<t>}$	- - - -		$y^{<9>}$ $T_y = 9$

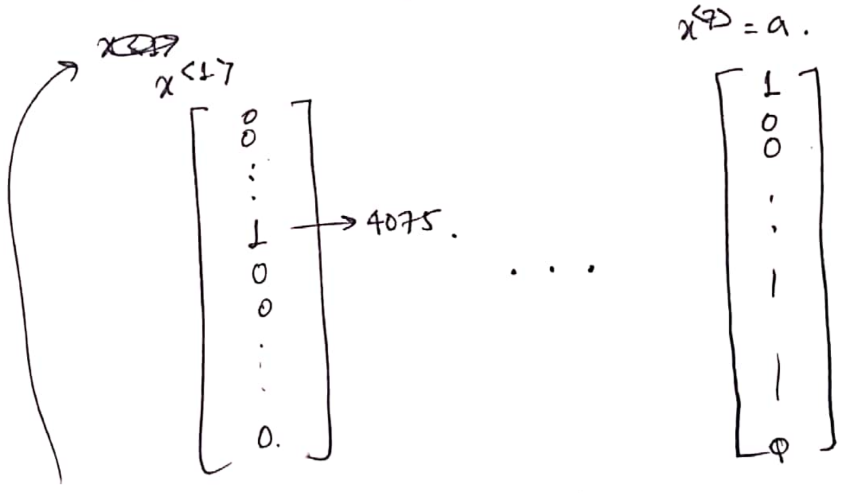
$X^{(i)<t>}$ → t^{th} element of ith Training ex

$T_x^{(i)}$ → length of i/p seq in i^{th} Traing ex.

Vocabulary

a	1
aaron	2
⋮	⋮
and	367
⋮	⋮
harry	4075
⋮	⋮
potter	6830
⋮	⋮
zulu	10,000

10,000 words ;



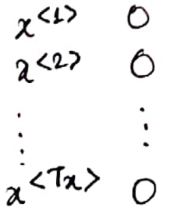
one hot vectors.

use one-hot representation

<unk> : for words not in vocabulary <unknown token>.

* Recurrent Neural N/w (RNN) model :

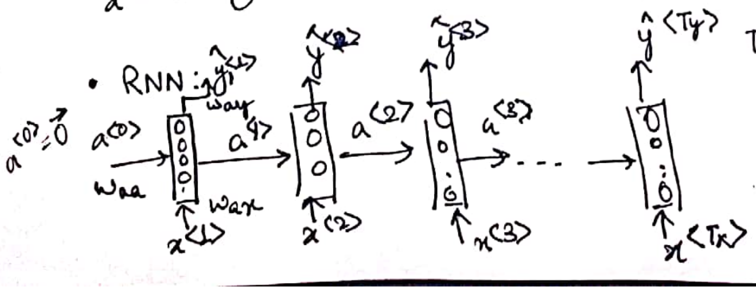
- why not a standard NN?



problems :

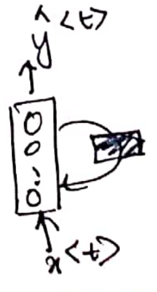
- i/p and o/p can be different lengths in diff examples
- Doesn't share features learned across diff posn of text.
- very large i/p layer.

- RNN :



$T_x = T_y$.

other notation :



- Scans left to right

- only earlier layers are used to predict : problematic as first few letters cant be names.
Ex : can confuse b/w $\underset{\text{object}}{\underline{\text{Teddy}}}$ & Teddy $\underset{\text{name}}{\underline{\text{Roosevelt}}}$. Sol^n : bidirectional RNN.

~~$a^{<1>} = g(w_{aa}a$~~ ~~$a^{<1>} = g$~~

$a^{<1>} = g_1(w_{aa}a^{<0>} + w_{ax}x^{<1>} + ba) \leftarrow$ tanh | Relu

$\hat{y}^{<1>} = g_2(w_{ya}a^{<1>} + by) \leftarrow$ sigmoid

$$\boxed{\begin{aligned} a^{<t>} &= g(\underline{w_{aa}a^{<t-1>} + w_{ax}x^{<t>}} + ba) \\ \hat{y}^{<t>} &= g(w_{ya}a^{<t>} + by) \end{aligned}}$$

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

Ex: $\underset{100,100}{w_{aa}}$ $\quad \underset{100}{a^{<t-1>}}$ $\quad \underset{(100,10000)}{w_{ax}}$

$$w_a = [\underbrace{w_{aa}}_{100} \vdots \underbrace{w_{ax}}_{10000}] \updownarrow 100$$
$$\underbrace{\qquad\qquad\qquad}_{10100}$$

$$[a^{<t-1>}, x^{<t>}] = \begin{bmatrix} a^{<t-1>} \\ \hline x^{<t>} \end{bmatrix}$$

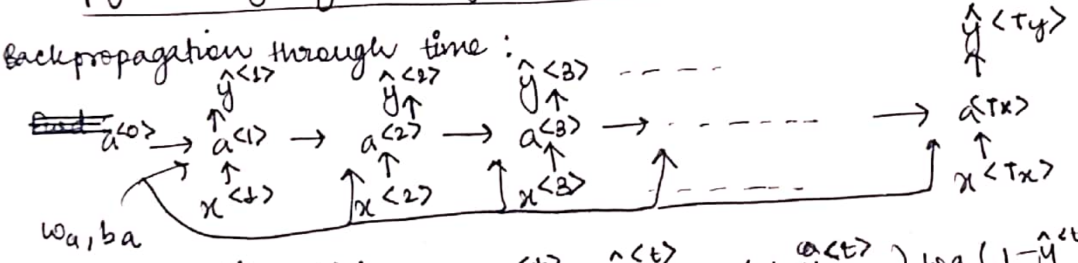
100 10000 10100

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

Thus,

$$\boxed{\begin{aligned} a^{<t>} &= g(W_a[a^{<t-1>}, x^{<t>}] + ba) \\ \hat{y}^{<t>} &= g(W_{ya}a^{<t>} + b_y) \end{aligned}}$$

* Backpropagation through time :



w_y, b_y given to all

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

$$L(\hat{y}, y) = \sum_{t=1}^{Ty} L^{<t>}(\hat{y}^{<t>}, y^{<t>}) \leftarrow$$

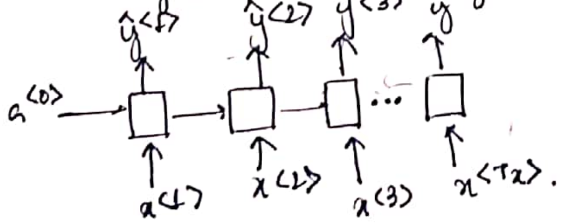
Here, it is Backpropagation through time.

* Different ~~[illegible]~~ types of RNN :

Tx and Ty might not always be same.

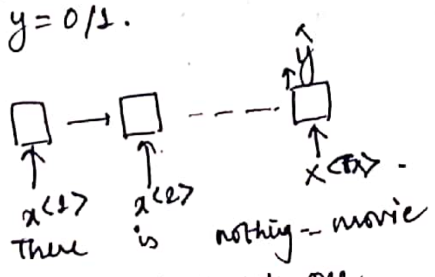
- Examples of RNN architectures :

Tx = Ty.



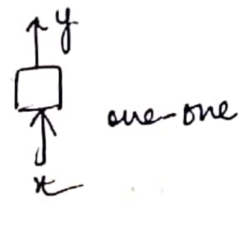
many to many.

Ex : Sentiment classification :
X = text
y = 0/1.



many to one

Ex :

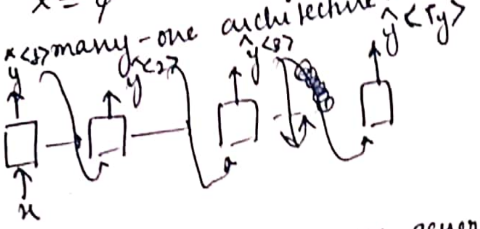


- Music generation:

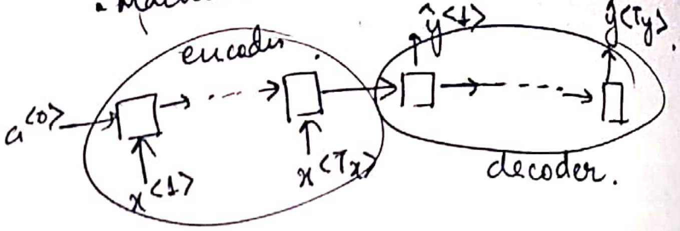
→ $y^{<1>} y^{<2>} \dots y^{<T_y>}$

$x = \phi$

many-one architecture



- Machine Translation.



* Language model and sequence generation:

- language modelling:

Speech recog'

- The apple & pair salad —(I) $P(1) = 3.2 \times 10^{-13}$
- The apple & pear salad. —(II) $P(2) = 5.7 \times 10^{-10}$

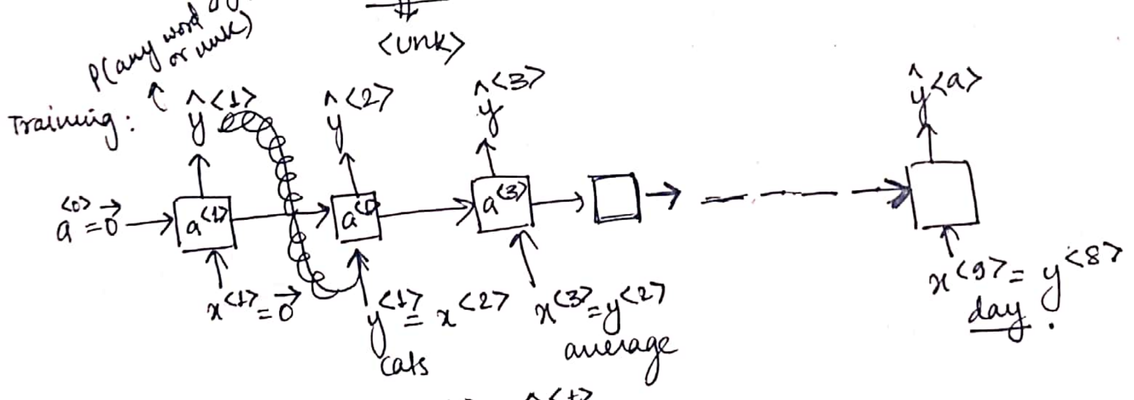
P(sentence) = ? $y^{<1>}, y^{<2>}, \dots, y^{<T_y>}$

Training set: large corpus of eng text.
$\Downarrow$ set

Cats average 15 hrs each day ~~<EOS>~~ Tokenise this + 1 extra token <EOS> end of sentence.

$y^{<1>}$ $y^{<2>}$, ---- $y^{<7>}$ $y^{<9>}$ $x^{<t>} = y^{<t-1>}$

The Egyptian Mau is beard of cat <EOS>
$\Downarrow$
<UNK>

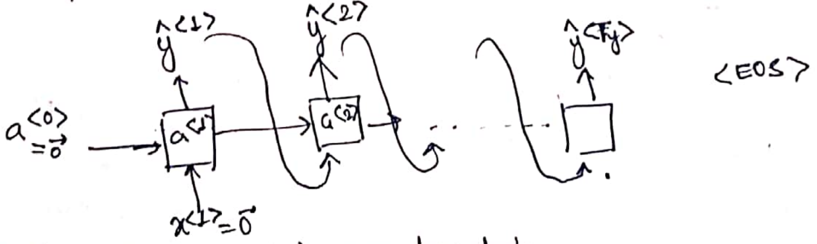


$$L(\hat{y}^{<t>}, y^{<t>}) = -\sum_i y_i^{<t>} \log \hat{y}_i^{<t>}$$

$$\boxed{L = \sum_t L^{<t>}(\hat{y}^{<t>}, y^{<t>})}$$

*Sampling novel sequences: P(sec word, given first word is the)
P(__ | the)

Sampling



→ P(a) P(aaron) P(cat) ... P(zulu) = np.random.choice.

character level RNN:
Vocabulary = [a, b, ..., z, 0, ..., 9, ., ;, ,, ' ', ', |, . — . A ... Z]

has own adv & disadv, much longer sequences.

much longer sequences.

*Vanishing gradients with RNNs :

The $\boxed{\text{cat}}$ which already ate ...., $\boxed{\text{was}}$ full

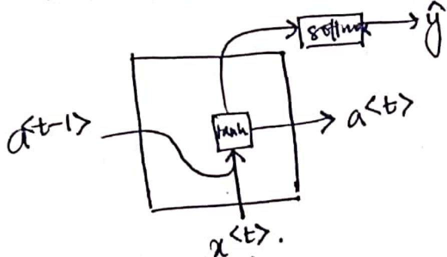
The $\boxed{\text{cats}}$ - - - - - - .... $\boxed{\text{were}}$ full.

derivative can ↑se or ↑se exponentially due to very deep RNN.

vanishing grads are problematic. can cause numerical overflow → sol^n: gradient clipping

*Gated Recurrent Units (GRU)

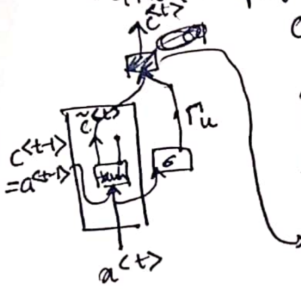
- makes RNN better at capturing long range connections.



papers : Cho et al, 2014 : on properties of neural M/C translation encoder-decoder approach.

Chung-et-al, 2014 : Empirical Eval. of gated RNN on seq. modelling

- GRU (simplified)



$c^{\langle t \rangle}$
$\tilde{c}^{\langle t \rangle}$
Γ_u
σ
tanh
$c^{\langle t-1 \rangle} = a^{\langle t-1 \rangle}$
$a^{\langle t \rangle}$

c = memory cell

$c^{\langle t \rangle} = a^{\langle t \rangle}$

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

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

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

(elementwise multipl^n) (as same dim^n)

$*$ → multiplic^n/ not convolution

$c^{\langle t \rangle} = \overset{1 \text{ (singular)}}{\underset{0 \text{ (plural)}}{}}$ - - - - - - = 1

The $\boxed{\text{cat}}$, which .... $\boxed{\text{was}}$ full

$\Gamma_u = 0$ $\Gamma_u = 0$ (when do you update). ———— $\Gamma_u = 0$ memorizes cat is similar.

Full GRU:

$\tilde{h}$ $\tilde{c}^{\langle t \rangle} = \tanh(w_c[\Gamma_r * c^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_c)$.

u $\Gamma_u = \sigma(W_u[c^{\langle t-1 \rangle}, \cancel{x^{\langle t \rangle}] + b_u})$

r $\Gamma_r = \sigma(w_r[c^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_c)$

h $c^{\langle t \rangle} = \Gamma_u * \tilde{c}^{\langle t \rangle} + (1 - \Gamma_u) + c^{\langle t-1 \rangle}$.

↳ alternate notations.

*Long short Term Memory : (LSTM)

- more powerful than GRU.

Paper : Hochreiter & Schmidhuber 1997. LSTM.

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

upd. gate $\Gamma_u = \sigma(W_u[\underline{a^{\langle t-1 \rangle}, x^{\langle t \rangle}}] + b_u)$

$\Gamma_f = \sigma(w_f[\underline{a^{\langle t-1 \rangle}, x^{\langle t \rangle}}] + b_f)$ variation → $c^{\langle t-1 \rangle}$ replaces $x^{\langle t \rangle}$. - peephole connection

forget gate

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

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

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

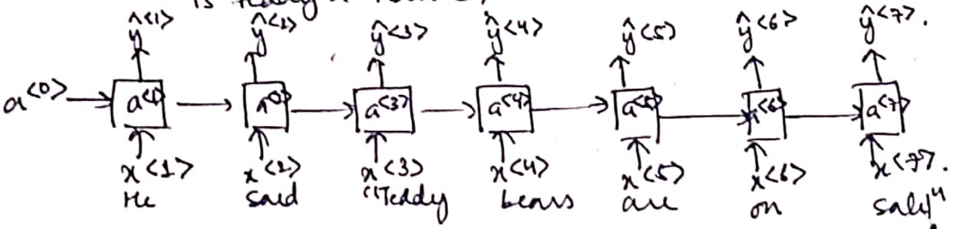
Scanned by CamScanner

* Bidirectional RNN

- Takes info from later & earlier inputs.

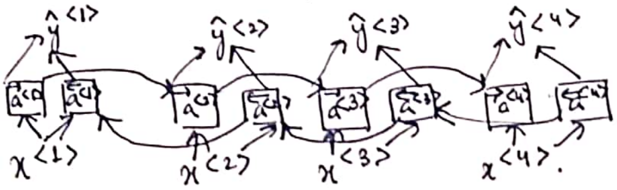
Ex: He said, "Teddy bears are on sale!"
He said, "Teddy Roosevelt was a great Prez!"

is teddy a name?



- BRNN.

also GRU / LSTM.

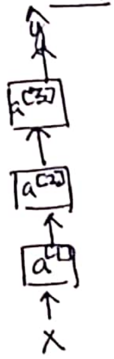


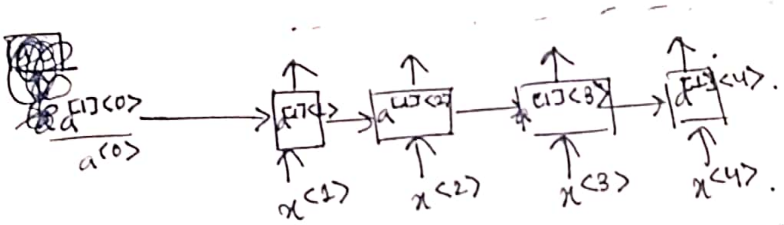
$$\hat{y}^{<t>} = g(W_y[\vec{a}^{<t>}, \overleftarrow{a}^{<t>}] + b_y)$$

* Deep RNNs :-

Ex:

multiple RNNs stacked together.





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

⊕ Natural language processing & word Embeddings :-

Word embeddings.

* Word Representation :

- V = [a, aaron, ..., zulu, <UNK>] |V| = 10,000

1-hot representation

weakness :

- Treats the word as object unto itself, and doesn't allow algo to easily generalise across words.

Featurised representation: word embedding

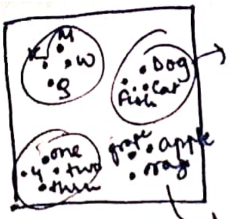
	Man (5391)	Woman (9853)	King (4914)	Queen (7157)	Apple (456)	Orange (6257)
gender	−1	1	−0.95	0.97	0.00	0.01
Royal	0.01	0.02	0.93	0.95	−0.01	0.00
Age	0.03	0.02	0.7	0.69	0.03	−0.02
Food.	0.04	0.01	0.02	0.01	0.95	0.97.
⋮						

300

e_{9853}

↳ 300D vector represents man e_{5391}

paper: van der Maaten and Hinton, 2008. Visualising data using t-SNE.

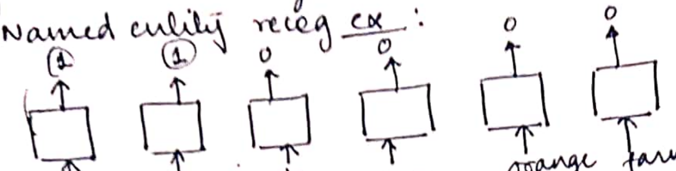


→ similar objects grouped together.

t-SNE. → objects are embeddings.

* using word embeddings :

Ex : Named entity recog ex :



Sally Johnson is an orange farmer } one hot encoding fails / word embedding succeeds.

Robert lin is an apple ~~farmer~~ / a durian cultivator. → to learn with word embedding

Steps : Transfer lear & word Embedding.

1. learn word embeddings from a large text corpus (1-100B words) (or download pre-trained embedding online)
2. Transfer embedding to new task with smaller Tr.S. (300D dense feature vector)
3. Optinal : Continue to fine tune word embeddings with new data.

- word embeddings generalise algorithms much better.

* Properties of word embeddings :

Analogies :

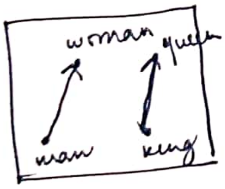
Man → woman , king → ? } ⇒ $e_{man} - e_{woman} \approx \begin{bmatrix} -2 \\ 0 \\ 0 \\ 0 \\ \vdots \end{bmatrix}$

e_{man} e_{woman} (embedding vectors).

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

Paper : Mikolov et. al. 2013 , linguistic Regularities in continuous space word representation

Analogies using word vectors :



300D

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

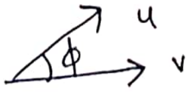
Find word w : $\arg\max_w$ similarity $(e_w , e_{king} - e_{man} + e_{woman})$

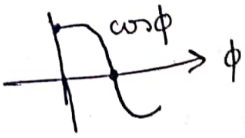
tSNE : 300D → 2D (parallelogram relationships may hold true (as above))

- Cosine similarity :

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

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

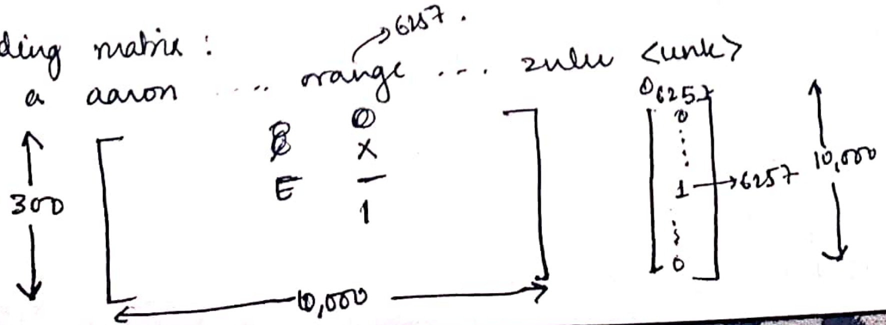




$\cos\phi = \phi$ $\phi = 0°$ similar

$\cos\phi = 0$ $\phi = 90°$ different.

* Embedding matrix :



$E_{(300,10K)} \cdot O_{10K,1} = \begin{bmatrix} 0 \\ X \\ 1 \end{bmatrix}$ ← 300,1

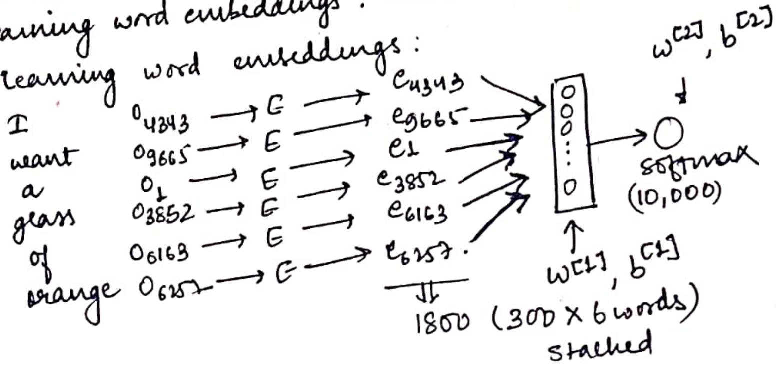
$= e_{6257}$

$E \cdot O_j = e_j$

= embedding for word j

learning word embeddings : word2vec & Glove

* Learning word embeddings :


I
want
a
glass
of
orange
$o_{4343} \rightarrow E \rightarrow e_{4343}$
$o_{9665} \rightarrow E \rightarrow e_{9665}$
$o_1 \rightarrow E \rightarrow e_1$
$o_{3852} \rightarrow E \rightarrow e_{3852}$
$o_{6163} \rightarrow E \rightarrow e_{6163}$
$o_{6257} \rightarrow E \rightarrow e_{6257}$
$w^{[2]}, b^{[2]}$
softmax
(10,000)
$w^{[1]}, b^{[1]}$
1800 (300 x 6 words)
stacked

paper : Bengio 2003
A neural probabilistic language model.

Other context/target pairs :

I want $\underbrace{\text{a glass of orange}}_{\text{context}}$ $\underbrace{\text{juice}}_{\text{target}}$. $\underbrace{\text{to go along with my}}_{\text{context}}$ cereal.

context : last 4 words or last 1 word or Nearby one word
orange ? ⇓ skip-gram ⇘ glass ?

* word2vec

- Skip grams (allows to create supervise learning task)

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

Context	Target
orange	Juice.
orange	glass
orange	my.

paper: Mikolov 2013
Efficient estim of word represent in vector space

Model

Vocab = 10,000 words

$$\text{context } \overset{x}{c}(\text{"orange"}) \xrightarrow{} \text{Target } \overset{y}{t} \ (\text{"juice"})$$

6257 4834

$$\boxed{o_c \rightarrow E \rightarrow e_c \longrightarrow \underset{\text{softmax}}{O} \longrightarrow \hat{y}}$$

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

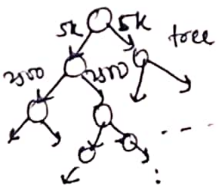
$\theta \rightarrow$ param associate with o/p t.

$$L(\hat{y}, y) = -\sum_{i=1}^{10000} y_i \log \hat{y}_i$$

$y_{10,000} \rightarrow$ one hot vector

problems :

→ computational speed.
soln : Hierarchical softmax



tree may be developed as non symmetrical, common words on top, whereas non common on lower level.

How to sample context c ?

the, of, a, and, to, ...
orange, apple, durian ...

* Negative sampling :

$\overset{x}{\text{Context ~~target~~ word}} \longrightarrow \overset{y}{\text{target ?}}$

c	t	y	
orange	juice	1	← true ex
orange	king	0	
orange	book	0	← zero (-ve) example
orange	the	0	
orange	of	0	

k

K = 5-20 for smaller D sets
K = 2-5 for larger Data sets.

paper : Mikolov 2013,
Distributed representation of words & phrases & their composition

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

instead of 10000 ∆ softmax, treat as 10 000 binary classification problem

* GloVe word vectors : (global vectors for word representation)

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

paper: pennington, 2014 global vectors for word representation

c, t.

$$x_{ij} = \# \text{ times } \underset{\uparrow t}{i} \text{ appears in context of } \underset{\uparrow c}{j}$$

x_{ij}: $i \leftarrow c$, $j \leftarrow j$

maybe: $x_{ij} = x_{ji}$ (symmetric case)

model : minimise $$\sum_{i=1}^{10000} \sum_{j=1}^{10000} f(x_{ij}) (\underset{t}{\theta_i}^T \underset{c}{e_j} + b_i - b_j' - \log x_{ij})^2$$

$\theta_t^T e_c$ → $\log x_{ij}$

"$0 \log 0$" $= 0$

waiting term
$f(x_{ij}) = 0$ if $x_{ij} = 0$.

θ_i, e_j are symmetric.

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

. A note on on the featurisation view of word embeddings.

Applications using word embeddings :

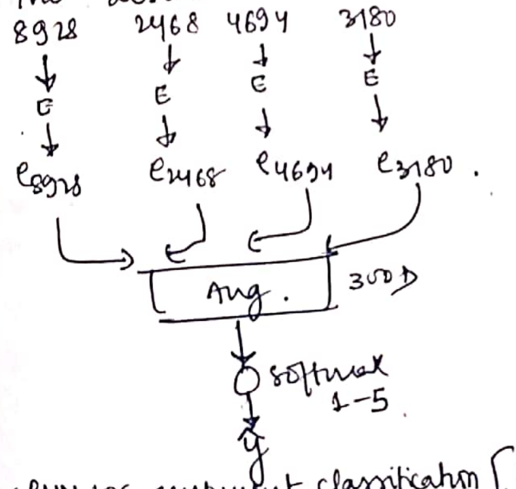
*Sentiment classification :-

x	y
The dessert is excellent	4 stars.
service was slow	2 stars
Good for a quick meal, nothing special	3 stars
Completely lacking in good state, good ...	1 star.

X → Y → can be used to monitor comments.

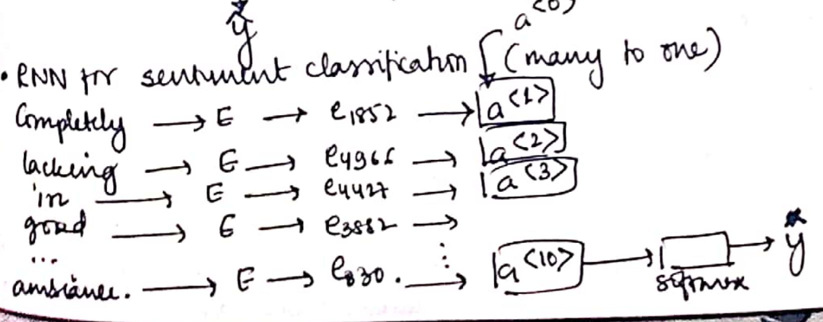
• Simple sentiment class. model :

The desert is excellent → 4 stars.



problem : if "good" appears many times in bad reviews, then classifier may think this is a good review

• RNN for sentiment classification (many to one)



* Debiasing word embeddings :
gender bias, etc . . .

paper : Bolukbasi 2016
Man is to computer programmer as woman is to homemaker ? Debiasing word embeddings

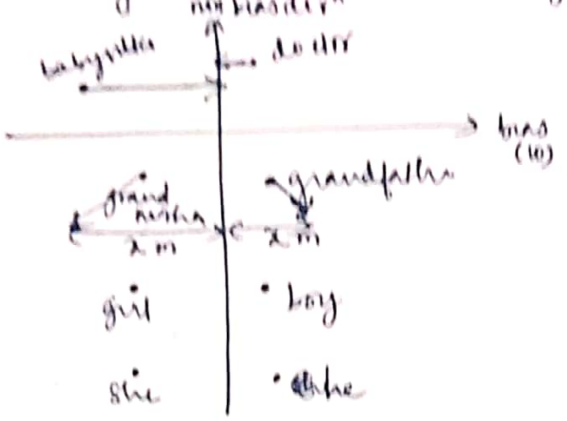
Man : Woman as King : Queen

Man : Computer Programmer as Woman : Homemaker ? X ↳ gender bias

Father : Doctor as Mother : Nurse X ↳ gender bias

- Word embeddings can ~~be used~~ reflect gender, ethnicity, age, sexual orientation, . .

• Addressing bias in word embeddings :



1. Identify bias direction
$e_{he} - e_{she}$
$e_{male} - e_{female}$
⋮
average

2. Neutralize : For every word that not definitional, project to get rid of bias
Ex: grandfather
grandmother
father
mother.

3. Equalization of pairs
grandmother — grandfather . (Ex : equidistant from babysitter)
girl — boy

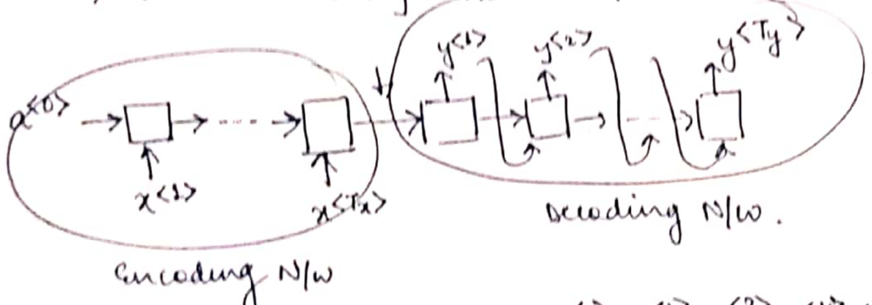
Ⓘ Sequence Models & Attention Mechanisms :

Various sequence to sequence architecture :

* Basic Models :

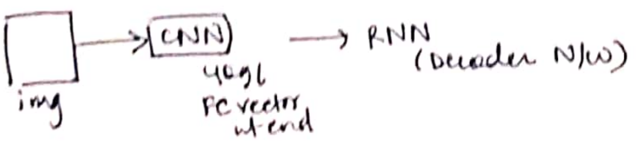
• Translation $\overset{x^{<1>}}{\text{Jane}}$ $\overset{x^{<2>}}{\text{visite}}$ $\overset{x^{<3>}}{\text{l'Afrique}}$ $\overset{x^{<4>}}{\text{en}}$ $\overset{x^{<5>}}{\text{septembre}}$
→ $\overset{y^{<1>}}{\text{Jane}}$ $\overset{y^{<2>}}{\text{is}}$ $\overset{y^{<3>}}{\text{visiting}}$ $\overset{y^{<4>}}{\text{Africa}}$ $\overset{y^{<5>}}{\text{in}}$ $\overset{y^{<6>}}{\text{september}}$

papers: • Sutskever, 2014 Sequence to sequence learning with Neural Networks
• Cho, 2014 Learning phrase rep. using RNN encoder-decoder for statistical m/c translation



• Img captioning :
$\overset{y^{<1>}}{\text{A}}$ $\overset{y^{<2>}}{\text{cat}}$ $\overset{y^{<3>}}{\text{sitting}}$ $\overset{y^{<4>}}{\text{on}}$ $\overset{y^{<5>}}{\text{a}}$ $\overset{y^{<6>}}{\text{chair.}}$

paper: Deep captioning with multimodal RNN .

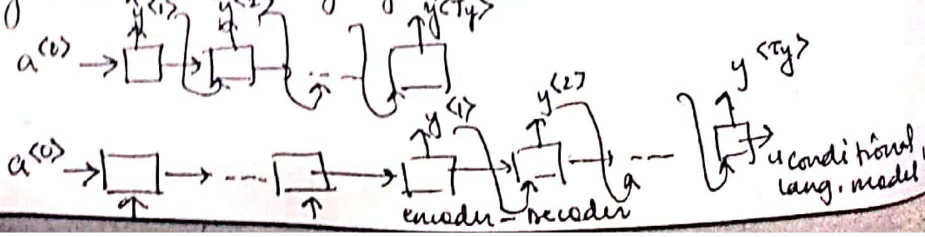


* Picking the most likely sentence :

• M/c translation as building a conditional language model .

language model

M/c translation:



$P(y^{<1>}, \dots, y^{<T_y>} \mid x^{<1>}, \dots, x^{<T_x>})$

Ex: $P(\underbrace{y^{<1>}, y^{<2>}, \dots\ y^{<T_y>}}_{\text{English possibility}} \mid x)$ ↳ french

$\therefore \arg\max_{y^{<1>}, \dots, y^{<T_y>}} P(y^{<1>}, \dots, y^{<T_y>} \mid x)$

↳ maximised by Beam search.

- why not greedy search?

Pick first words, then pick subsequently most likely words.

$P(\hat{y}^{<1>} \mid x)$

Ex: <u>Jane is</u> — going to visit —

going to be visiting.

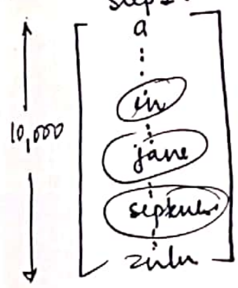
...

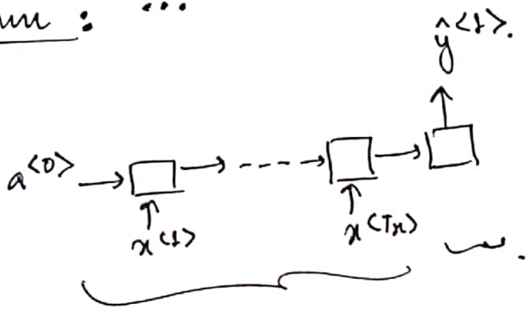
*Beam search Algorithm:

step 1.

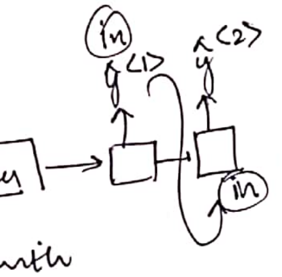
$B = 3$ (beam width)

→ $P(y^{<1>} \mid x)$





step 2 → evaluate 2nd word Att first word.



$P(y^{<2>} \mid x = \text{"in"})$

$\therefore P(y^{<1>}, y^{<2>} \mid x)$
$= P(y^{<1>} \mid x)$
$P(y^{<2>} \mid x, y^{<1>})$

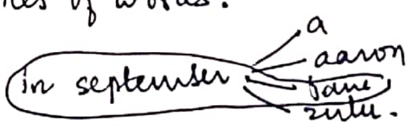
similarly with 'jane & september.

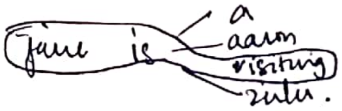
$\therefore$ 30,000 possibilities. ($B \times 10{,}000$).

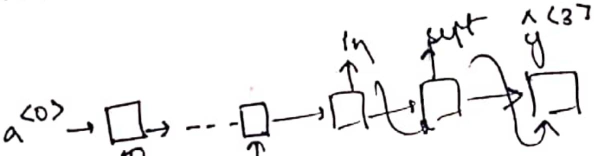
cut down possibilities

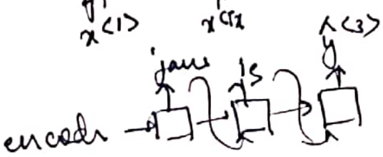
in sept ✓
Jane is ✓
'jane visits ✓ } → sept rejected with jane.

-3 copies of NWs for different choices of words.









* Refinements to beam search:

- length normalisation

$\arg\max_{y} \prod_{t=1}^{T_y} P(y^{<t>} \mid x, y^{<1>}, \dots, y^{<t-1>}) \Rightarrow < 1$

log

$\arg\max_{y} \sum_{y=1}^{T_y} \log P(y^{<t>} \mid x, y^{<1>}, \dots, y^{<t-1>})$

P log

$\log P(y \mid x)$ ↑sing

$P(y \mid x)$ ↑sing

$$\boxed{\frac{1}{T_y^{\alpha}} \sum_{t=1}^{T_y} \log P(y^{<t>} | x, y^{<1>}, \dots, y^{<t-1>})}$$ → Normalisation :

$\underline{\alpha = 0.7}$

↳ try different α.

$T_y = 1, 2, 3, \dots, 30$

Take sentences & compute this, highest value is kept.

- Beam width B ?

  large B : better result
  slower

  small B : worse result
  faster.

  on common system B = 10, 100
  research sometimes uses B = 1000, ...

→ unlike BFS or DFS, Beam search runs faster but not guaranteed to find exact maximum for $\arg\max_y P(y|x)$

* Error analysis in beam search :

Ex: Jane visite l'Afrique en septembre

Human : Jane visits Africa in september (y^*)

Algorithm : Jane visited Africa last september ($\hat{y}$)

→ RNN
→ Beam search. B↑
↓
for more data.

RNN ~~compltes~~ computes $P(y^*|x) \gtrless P(\hat{y}|x)$

Case 1 :

$P(y^*|x) > P(\hat{y}|x)$

Beam search chose $\hat{y}$, but y^* higher val.

∴ Beam search is at $\underline{\text{fault}}$.

Case 2 :

$P(y^*|x) < P(\hat{y}|x)$

y^* better trans than $\hat{y}$, but RNN predicted $P(y^*|x) < P(\hat{y}|x)$

RNN model at $\underline{\text{fault}}$.

- Process

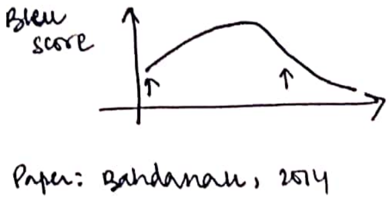
Human	Algo	$P(y^*\|x)$	$P(\hat{y}\|x)$	At fault ?
		2×10^{-10}	1×10^{-10}	B
				R
				R
				B
				R
				⋮

fraction of error due to beam search vs due to RNN

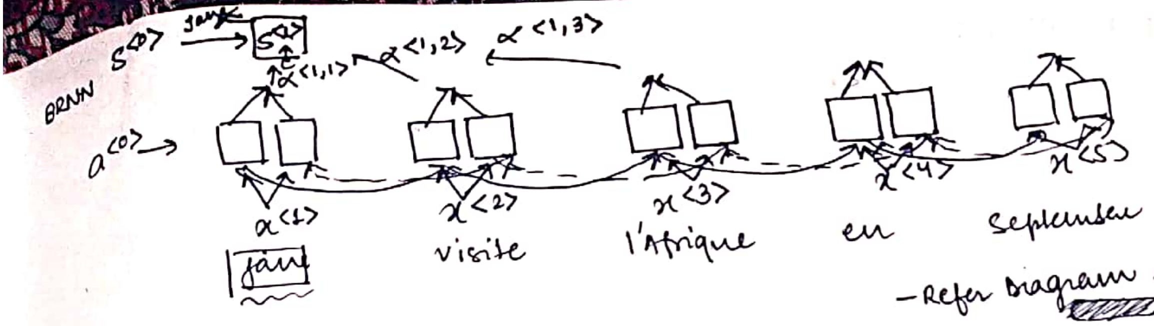
if Beam search resp, B↑es
else if RNN ~~[illegible]~~ is at fault, train ~~[illegible]~~ deeper N/w.

* Attention model intuition : → Translates part of a long sentence at a time.

- Problem of long sentences :



Paper: Bahdanau, 2014
Neural M/c translation by jointly learning to align & translate



-Refer Diagram to screenshot.

# Attention Model:

$(\overrightarrow{a}^{<t>}, \overleftarrow{a}^{<t>}) = a^{<t'>}$

context 'c' fed to upper RNN depends on attention params α

$\sum_{t'} \alpha^{<1,t'>} = 1.$

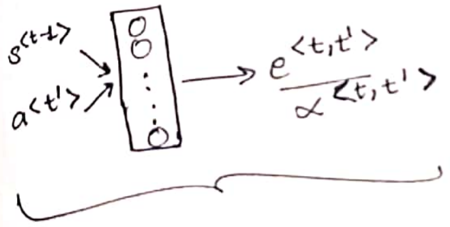
$c^{<1>} = \sum_{t'} \alpha^{<1,t'>} a^{<t'>}$

$\alpha^{<t,t'>}$ → amt of attention that $y^{<t>}$ must be paying to $a^{<t'>}$

$$\alpha^{<t,t'>} = \frac{\exp(e^{<t,t'>})}{\sum_{t'=1}^{T_x} \exp(e^{<t,t'>})}$$

paper: Bahdanau, 2014
Neural m/c translation by jointly learning to align & translate.

paper: Xu, 2015
Show attend & tell: Neural img caption generation with visual attention.



Ex

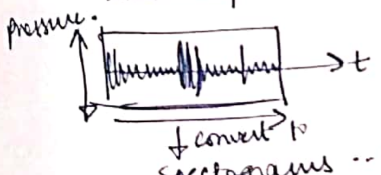
July 20th, 1969 → 1969-07-20

23 April, 1564 → 1564-04-23.

# Speech recognition - Audio data -

*Speech recognition:

x → y
audio clip transcript.



convert to spectograms ...

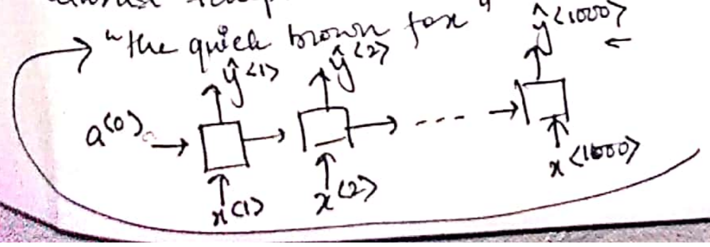
"the quick brown fox"

→ Hand engineered.

-once upon a time, speech recog. used to be built by "phonemes" (de kirk bran)

\#CTC cost for speech recog.
(Connectionist temporal classification)

paper: Graves, 2006
Connectionist temporal classification: labeling unsegmented sequence data with RNN

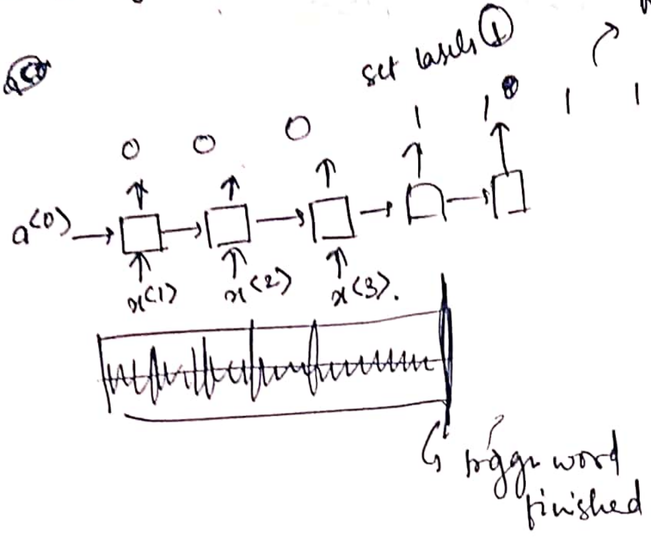


the⊔q

ttt_h_eee___⊔___qqq__ → blank
space

*. Trigger word detection :

Ex: Alexa, Hey siri, okay google, ...



→ multiple 1's slightly evens out ratio of 1's to 0's.