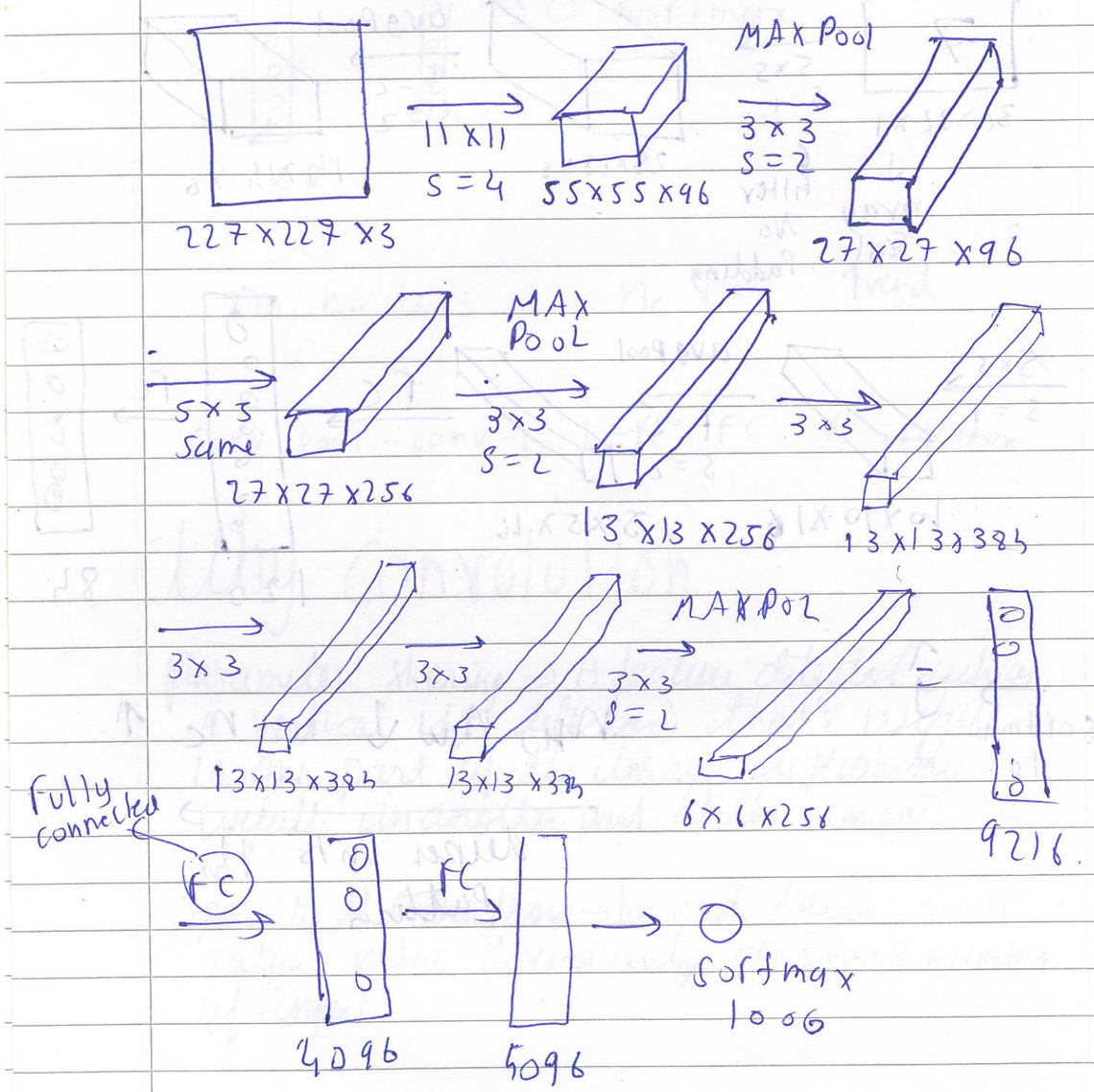
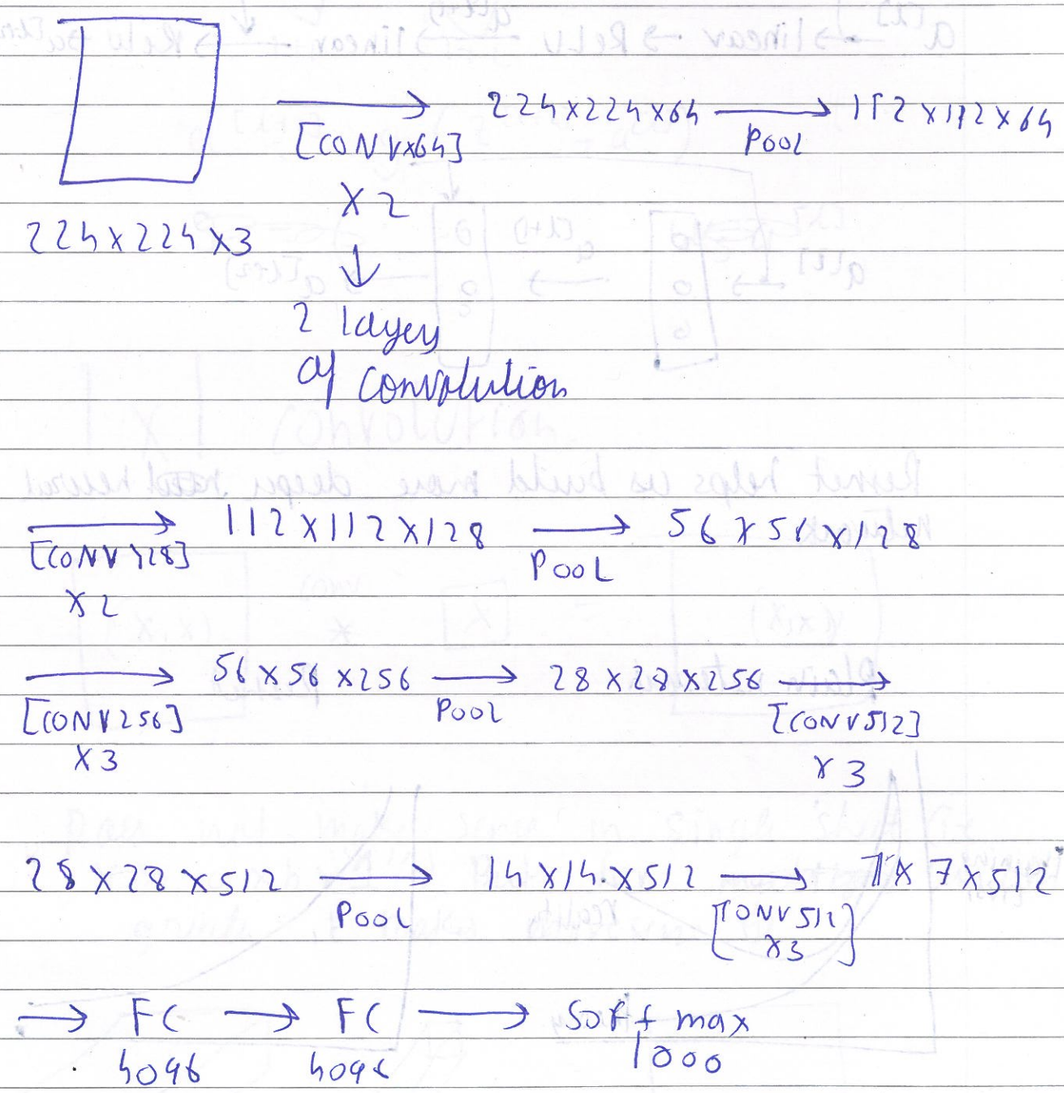
AlexNet.

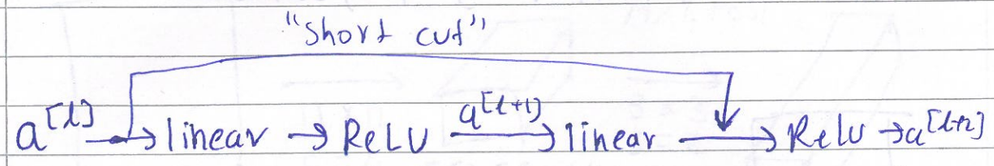


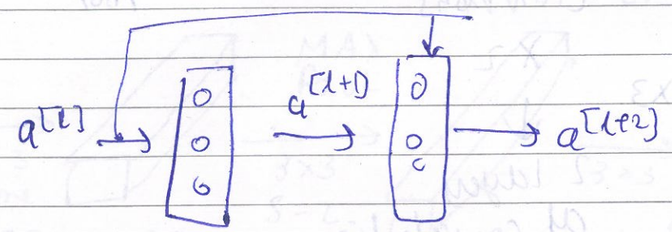
VGG-16.

CONV = 3x3 Filter S = 1, same



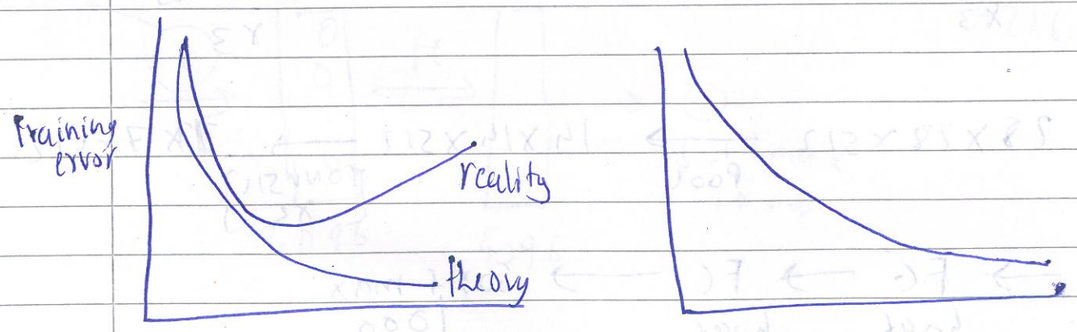
Resnet

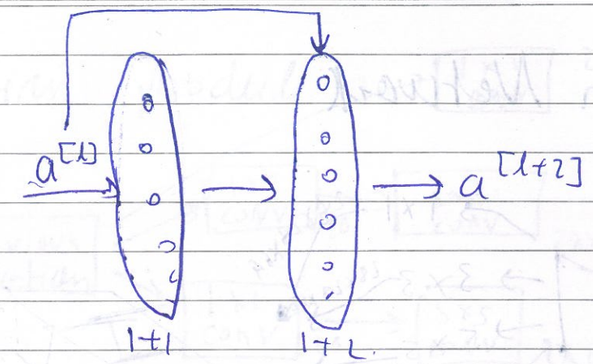




Resnet helps us build more deeper ~~neu~~ neural network

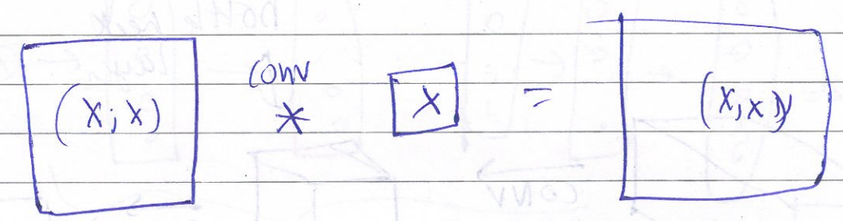
Plain network Resnet



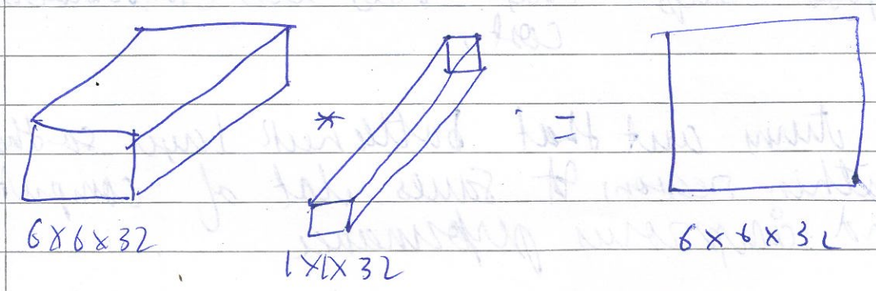


$a^{[l+2]} = g(z^{[l+2]} + a^{[l]})$

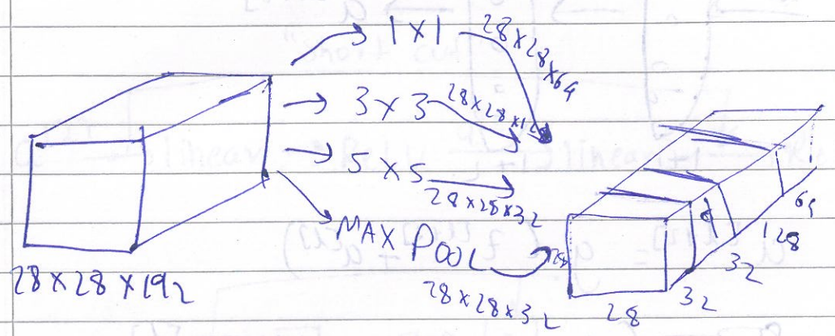
1x1 convolution



Does not make sense in single sheet (ie ~~xx~~ "a x b x 1") But for ~~multiple~~ width growth it makes more sense. ☺.

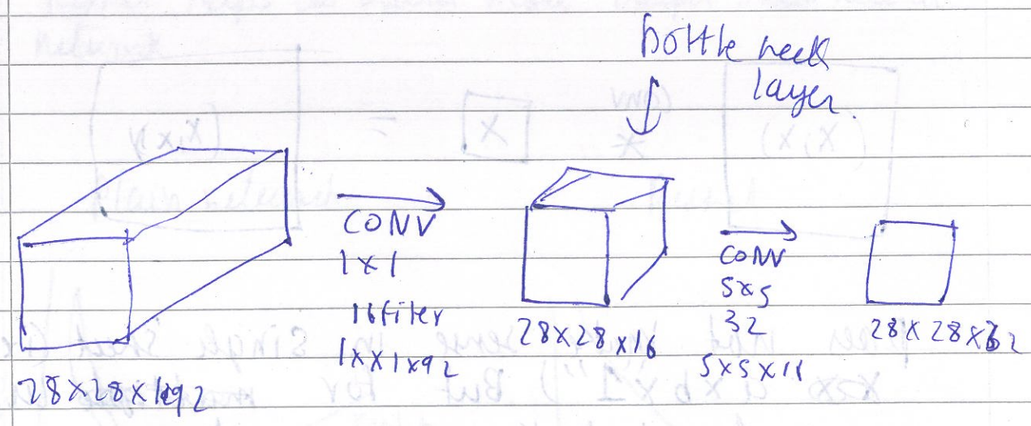


Inception Network



⇒ It has high computation cost.

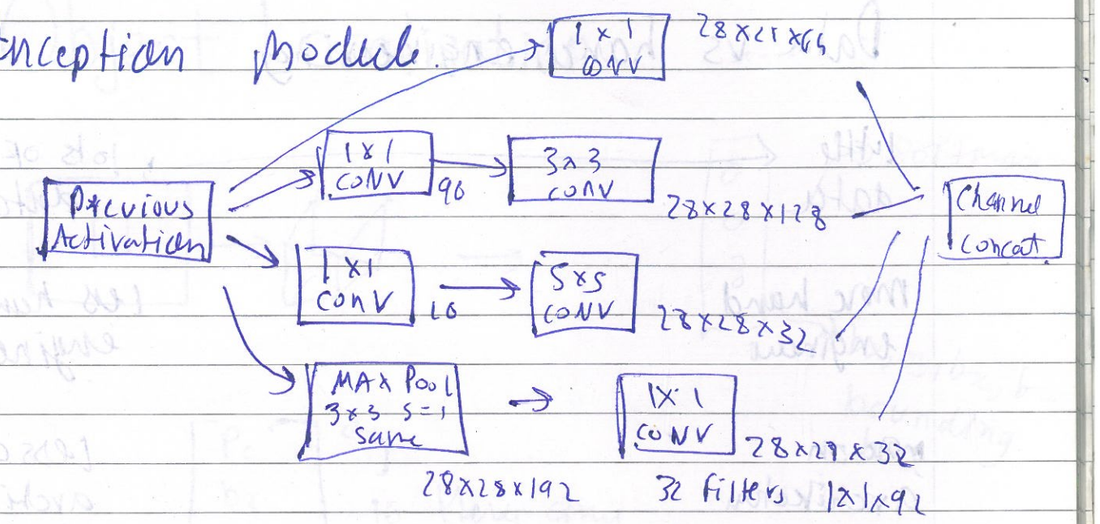
Alternate archetecture



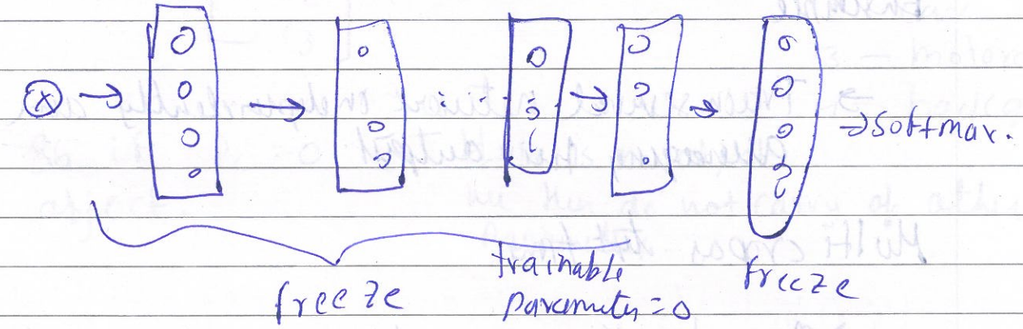
~~It just says~~ This has less convolution cost.

It turns out that bottleneck layer so that within reason, it saves lot of computation and compromise performance

Inception module



Transfer learning



- when you have small dataset.

Page :
Date :

Data vs hand engineering

little data ⟷ lots of data

more hand engineering — less hand engineering

more on architecture — less on architecture

Tips. for doing good in benchmarks.

Ensemble

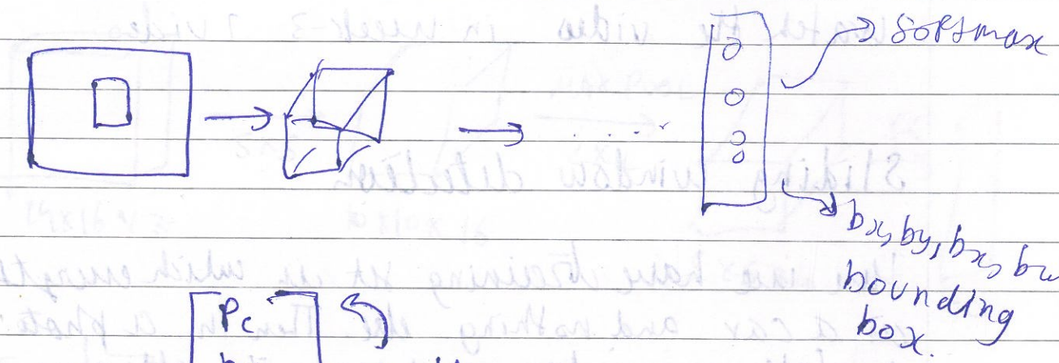
→ Train several network independently and averaging their output.

Multi crops test time.

→ Run classifier on multiple version of test image and average results.

Page :
Date :

Object Localisation



$$y = \begin{bmatrix} P_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

P_c → is there any object (

c_1, c_2, c_3 ← types of object i.e 1 - Pedistrian, 2 - car, 3 - motorcycle, 4 - background

So if $P_c = 0$ ∴ their is no object object. we then do not carry of other Parameters.

$$\text{Loss fn} = L(\hat{y}, y) = \begin{cases} (\hat{y}_1 - y_1)^2 + (\hat{y}_2 - y_2)^2 + \dots \\ \quad \dots (\hat{y}_8 - y_8)^2 & \text{if } y_1 = 1 \\ (\hat{y}_1 - y_1)^2 & \text{if } y_1 = 0 \end{cases}$$

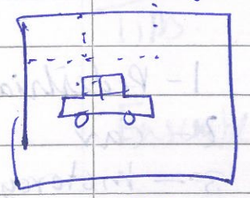
Page :
Date :

landmark detection.

watch the video in week-3 2 video.

Sliding window detection

Here we have training set in which everything is a car and nothing else. Then in a photo we take a window and slide it through out the entire image. Then we will increase ~~the slide with~~ size of the window and continue doing it.

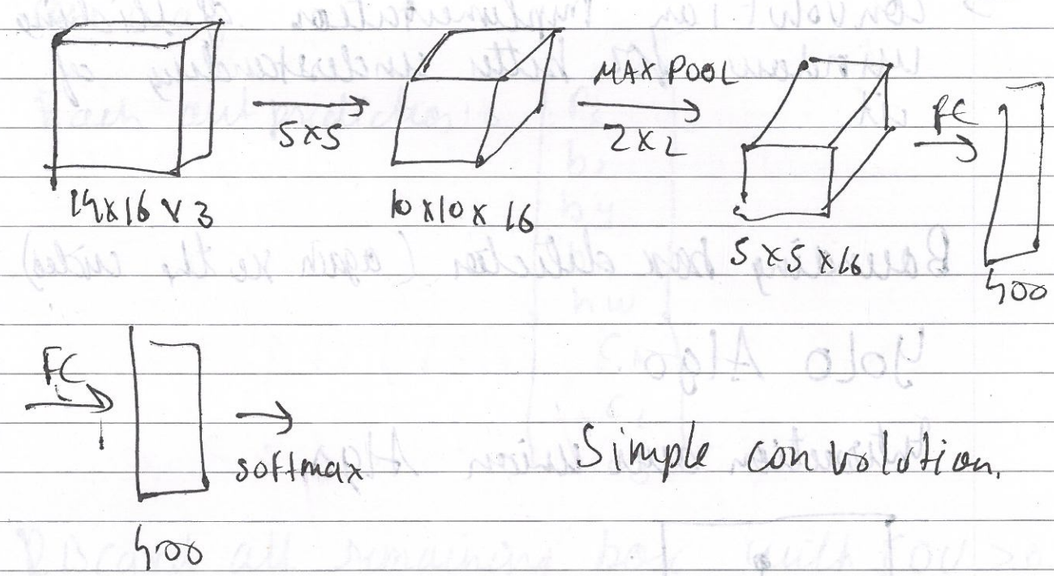


But sliding window can be implemented much effectively as the the above method is ~~co~~ costly.

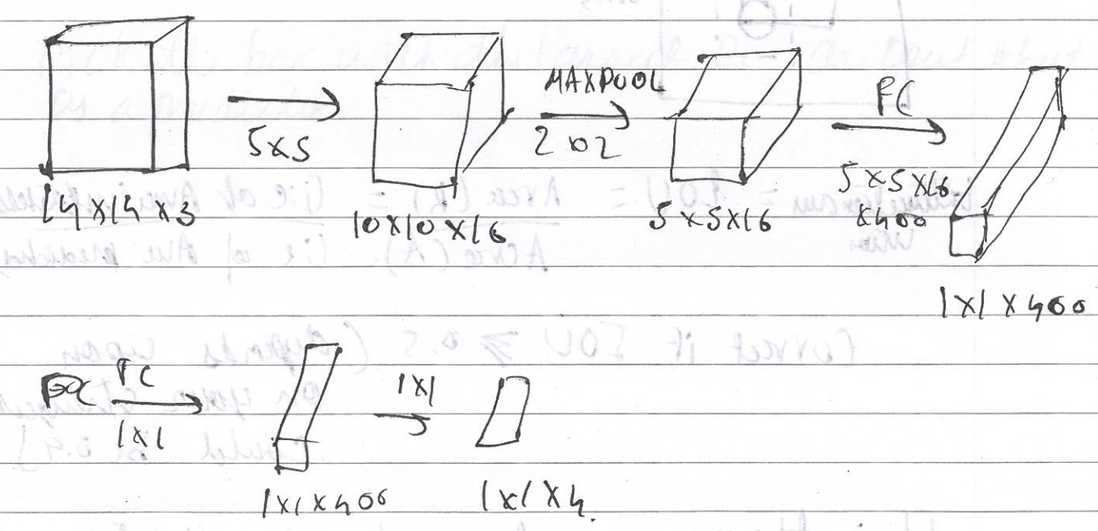
Convolution implementation of Sliding Window.

Page :
Date :

Turning FC layer into convolutional layer.



Simple convolution.



Page :
Date :

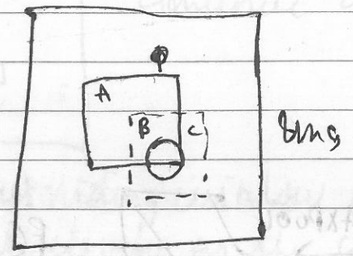
Please ~~get~~ see video

→ Convolution implementation of sliding window for better understanding of it.

Bounding box detection (again see the video).

Yolo Algo.

Intersection over union Algo.



$$\frac{\text{Intersection area}}{\text{Union}} = IOU = \frac{\text{Area}(B)}{\text{Area}(A)} = \frac{\text{(i.e of Area intersection)}}{\text{(i.e of Area predicted)}}$$

Correct if IOU $\geq$ 0.5 (depends upon on your stringent could be 0.9).

It is the measure of overlap of predicted and actual bounding box.

Page :
Date :

Non Max suppression.

One way.

Each out prediction is $\begin{bmatrix} P_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$

- Discard all remaining box with IOU > 0.5 with ~~box~~ the box output in the previous step.

- Pick the box with the largest P_c output that as a prediction.

Face Recognition vs face-Verification

Input	Input image name/ID	Has a database of K person
	output whether the input image is that of claimed person	Get an input image
		Output ID of a person in any K person.

One short learning

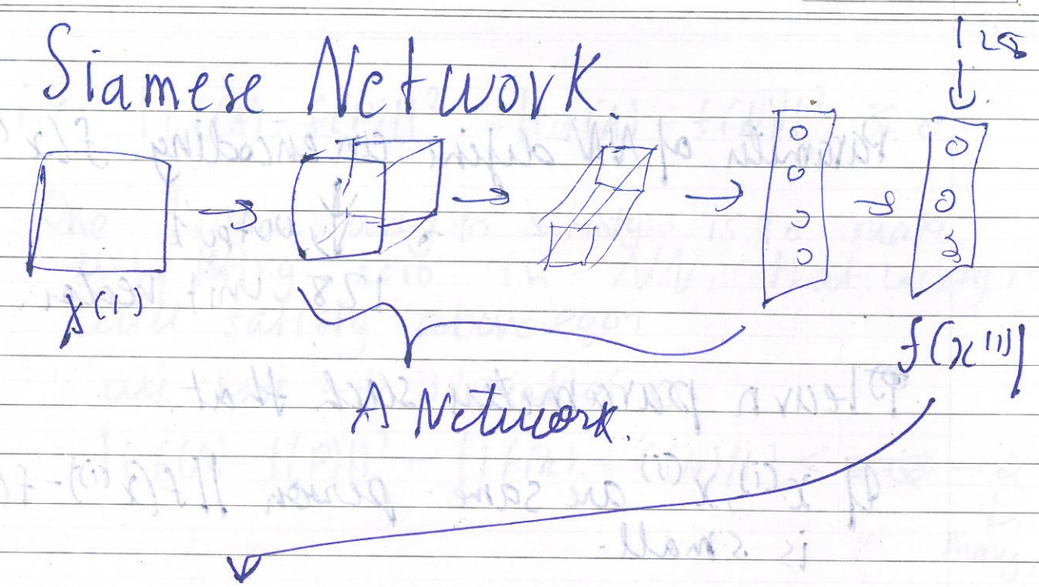
→ To identify with only 1 example.

Similarity F^n.

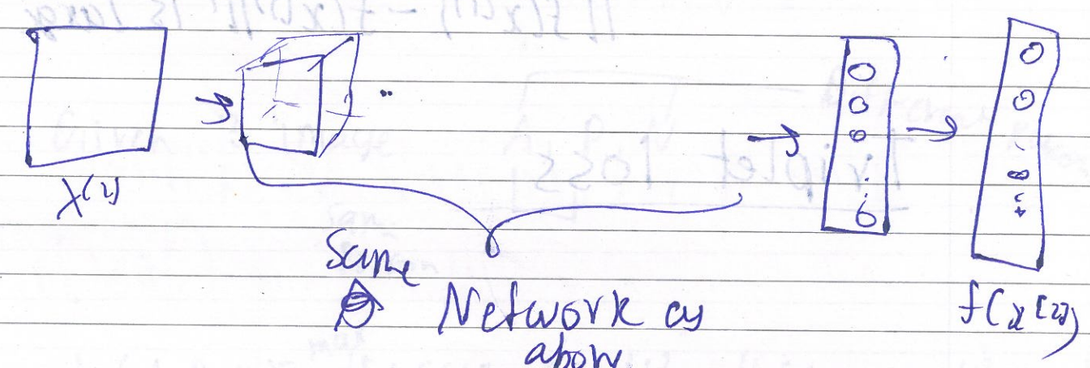
$d(img1, img2)$ = degree of diff bet^n 2 images.

if $d(img1, img2) \leq T$ "same"
$> T$ "diff" } verification

Siamese Network



It may be called encoding of $x^{(1)}$



$$d(x^{[1]}, x^{[2]}) = \| f(x^{[1]}) - f(x^{[2]}) \|_2^2$$

~~Siam~~ Siamese NN architecture.

Parameter of NN define an encoding $f(x^{(i)})$

$\downarrow$ output

128 unit vector.

learn parameters such that.

If $x^{(i)}, x^{(j)}$ are same person $\|f(x^{(i)}) - f(x^{(j)})\|^2$ is small.

If $x^{(i)}, x^{(j)}$ are different person

$\|f(x^{(i)}) - f(x^{(j)})\|^2$ is large

Triplet loss

Anchor image — +ve image — Anchor img — & -ve img

A — P — A — N.

$\|f(A) - f(P)\|^2 \leq \|f(A) - f(N)\|^2$

$\therefore$ We have Anchor, +ve, -ve image

$d(A,P)$ $\qquad$ $d(A,N)$

i.e $\|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 \leq 0$

One trivial way to satisfy is to make everything zero in NN. that way it will satisfy above equn

$\therefore$ we use -2 instead of 0.

$\|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 \leq -2$

$\downarrow$ margin

$\|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + 2 \leq 0$

Given 3 image A, P, N

A, P — Same Person

A, N — Different Person

$$\mathcal{L}(A,P,N) = \max\left(\|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + 2, 0\right)$$

$\downarrow$ loss f^n.

$$J = \sum_{i=1}^{m} \mathcal{L}(A^{(i)}, P^{(i)}, N^{(i)})$$

Page :
Date :

Choosing the triplet A, P, N

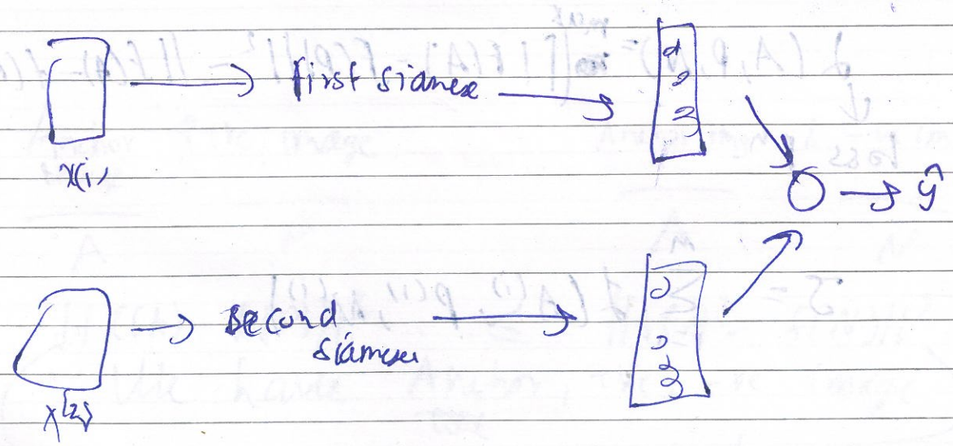
During training, if A, P, N are chosen randomly

$d(A,P) + \alpha \leq d(A,N)$ is easily satisfied

Choose triplet that're "hard" to train on

$d(A,P) + \alpha \leq d(A,N)$
$d(A,P) \approx d(A,N)$

Face verification vs Binary Classification



~~It's after~~ We have final logistic regn unit.

Page :
Date :

$$\hat{y} = \sigma\left(\sum_{k=1}^{128} w_k \underbrace{\left| f(x^{(i)})_k - f(x^{(j)})_k \right|}_{\text{else}} + b\right)$$

$$\frac{\left(f(x^{(i)})_k - f(x^{(j)})_k\right)^2}{f(x^{(i)})_k + f(x^{(j)})_k}$$

We can precompute the siamese value of second NN which can be directly used with 1st encoding.

Neural style Transfer

Content Image (C)

Style Image (S)

Generated Image (G)

What are deep ConvNet learning

watch the video.

Cost F^n.

Neural style Transfer

$$J(G) = \alpha J_{content}(C, G) + \beta J_{style}(S, G)$$

Step

① Initiate G randomly.

$G : 100 \times 100 \times 3$.

② Use gradient descent to minimize (G)

$$G \equiv G - \frac{\alpha}{\partial G} J(G)$$

Content cost F^n.

$J_{content}(C, G)$.

- Say you use hidden layer l to compute content cost
  Not ~~ste~~ shallow not too deep
- Use pre trained ConvNet
- let $a^{[l](C)}$ and $a^{[l](G)}$ are similar, both image have similar content.

$$J_{content}(C, G) = \frac{1}{2} \| a^{[l](C)} - a^{[l](G)} \|^2$$

Style of an Image

Say you are using l's activation to measure style.

Define ~~style~~ style as correlation bet^n activation across channel.

lets $a^{[l]}_{i,j,k}$ = activation at (i, j, k) $G^{[l]}$ is $n_c^{[l]} \times n_c^{[l]}$

$$G^{[l]}_{kk'} = \sum_i \sum_j a^{[l](S)}_{ijk} a^{[l](S)}_{ijk'}$$

$$G^{[l]}_{kk'} = \sum_{i=1}^{n_H^{[l]}} \sum_{j=1}^{n_W^{[l]}} a_{ijk}^{[l][G]} a_{ijk'}^{[l][G]}$$

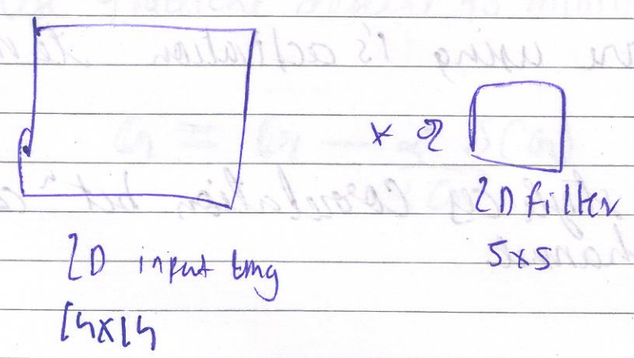
$$J^{[l]}_{style}(S,G) = \| G^{[l][S]} - G^{[l][G]} \|_F^2$$

$$= \frac{1}{(2 n_H n_W n_C)^2} \left(G^{[l][S]}_{kk'} - G^{[l][G]}_{kk'} \right)^2$$

$$J_{style}(S,G) = \sum_l \lambda^{[l]} J_{style}^{[l]}(S,G)$$

$$\cancel{J(G) = \alpha J_{content}(C,G) + \beta J_{style}(S,G)}$$

1D and 3D Generalization



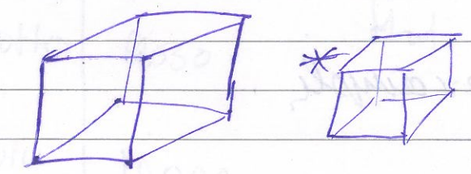
2D

$14 \times 14 \times 3 * 5 \times 5 \times 3 \rightarrow 10 \times 10 \times 16$

1D.

$14 \times 1 * 5 \times 1 \rightarrow 10$

for 3D



RNN

X: Harry potter and Hermoin Granger invented a new spell

$x^{<1>}$ $x^{<2>}$ $x^{<3>}$

T_x = length of sentence = 9

y 1 1 0 1 1 0 0 0 0 0

$y^{<1>}$ $y^{<2>}$ $y^{<3>}$

$T_y = 9$.

$X^{(i)}$ → training example

To refer t^{th} element of ith example we use

$X^{(i)<t>}$

Representing words.

X: Harry potter and Hermoin . . . .

Vocabulary

a	1
aaron	2
and	367
harry	4075.
Potter	6830
⋮	⋮
zulu	10000

How to choose 10000 words

→ To look through your training set find top 10000 occuring words. Or look online dictionary that also look through.

10000 words vocabulary

(It's relatively small)

We than use one hot representation

i.e $x^{<1>}$

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \leftarrow 4075$$

$x^{<2>}$

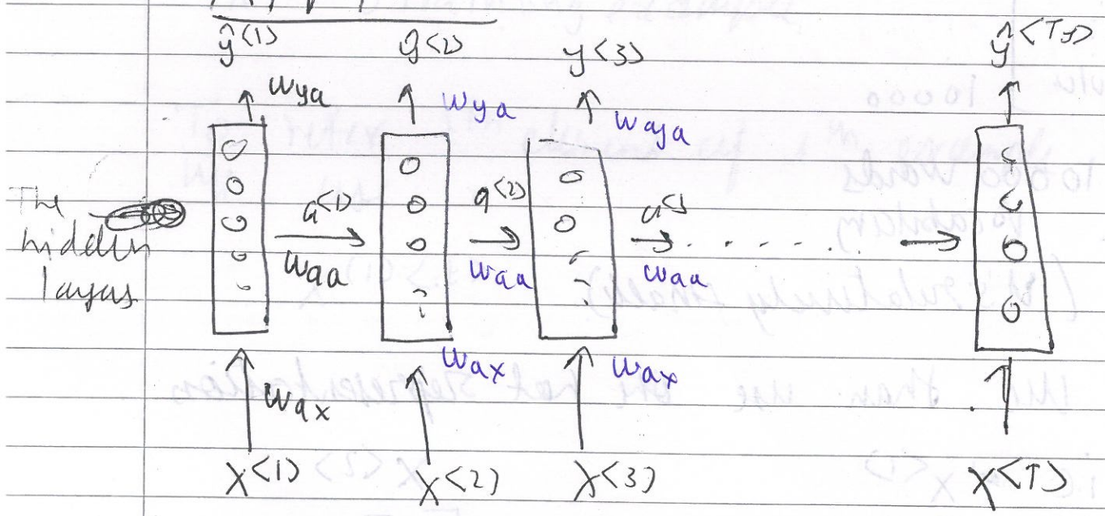
$$\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \leftarrow 6830$$

What if we don't have word in your vocabulary we a <UNKS> a unknown fake word.

Why not standar network?

It hard ~~and~~ write and simple to understan so just go to 3 video of week 1 to understand it.

RNN



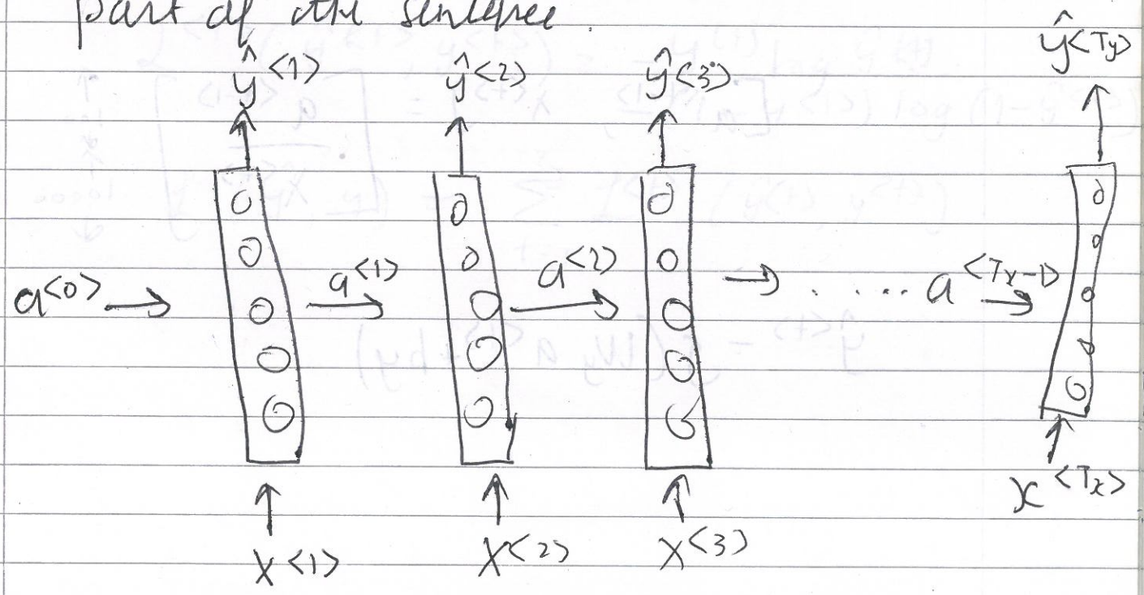
Each time a input comes it uses the previous ~~[illegible]~~ input activation to do it.

The above network has a problem. To output a result you may sometimes need the other part of sentence

i.e

He said "Teddy Roosevelt was a great president".

Here to know word Teddy is a person or thing we ~~[illegible]~~ would the other part of the sentence.



$$a^{<1>} = g\left(W_{aa}a^{<0>} + W_{ax}X^{<1>} + b_a\right) \leftarrow \text{tanh / Relu}$$

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

Page :
Date :

$$a^{\langle t\rangle} = g(\underbrace{W_{aa}}_{(100,100)} a^{\langle t-1\rangle} + \underbrace{W_{ax}}_{(100,10000)} x^{\langle t\rangle} + b_a)$$

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

we would write it as

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

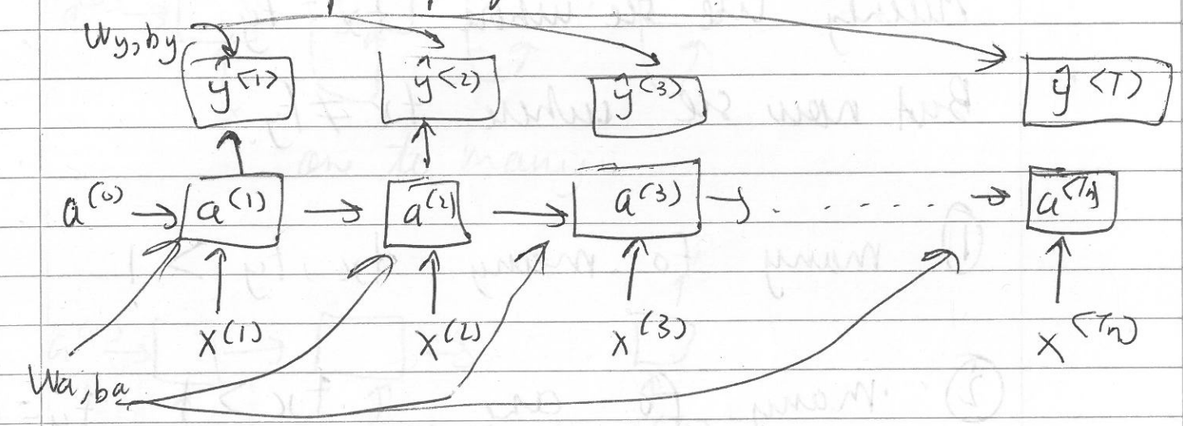
$$100 \updownarrow \underbrace{[W_{aa} \,;\, W_{ax}]}_{\langle 100 \rightarrow \langle 10000 \rightarrow} = W_a \quad (100, 10100)$$

$$[a^{\langle t-1\rangle}, x^{\langle t\rangle}] = \begin{bmatrix} a^{\langle t-1\rangle} \\ \hline x^{\langle t\rangle} \end{bmatrix} \begin{matrix} \uparrow \\ 100 \\ \updownarrow \\ 10000 \\ \downarrow \end{matrix}$$

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

Page :
Date :

Back propagation through time



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

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

Page :
Date :

Different types of RNN

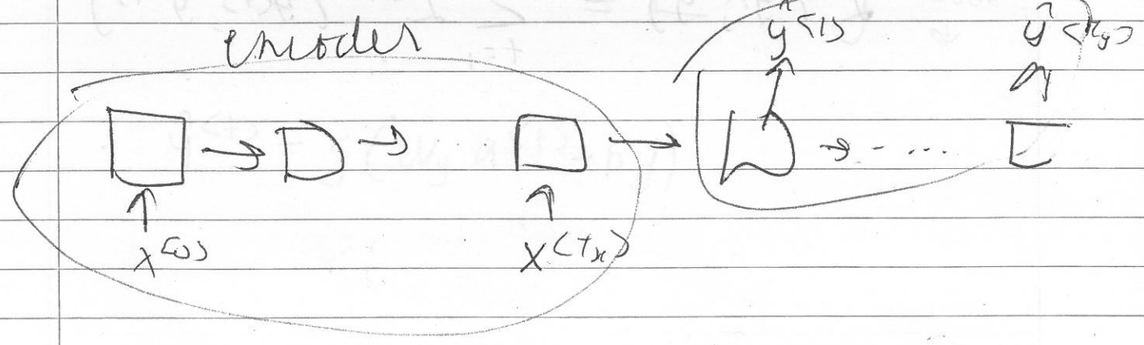
Mainly we see where $t_x = t_y$.

But now see where $t_x \neq t_y$.

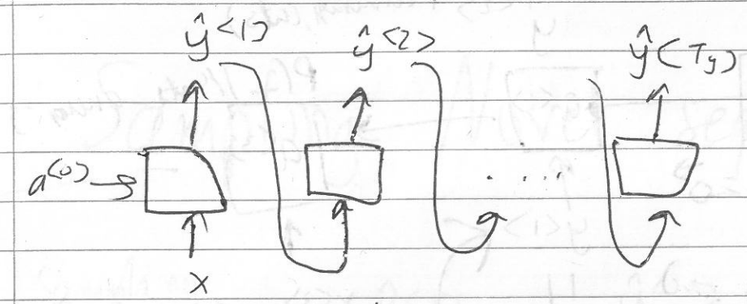
1. many to many $t_x, t_y > 1$
2. many to one $t_x > 1$ $t_y = 1$
3. one to many

Machine translation, decoder

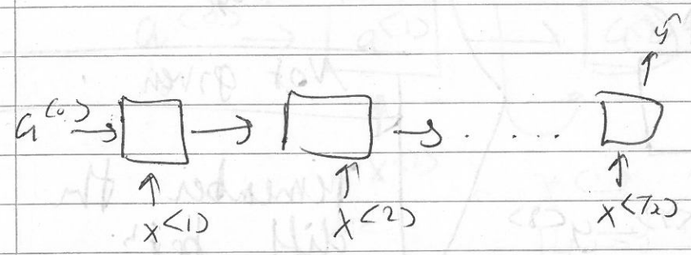
encoder



Page :
Date :



one to many



language modeling

Cats <EOS>

End of sentence

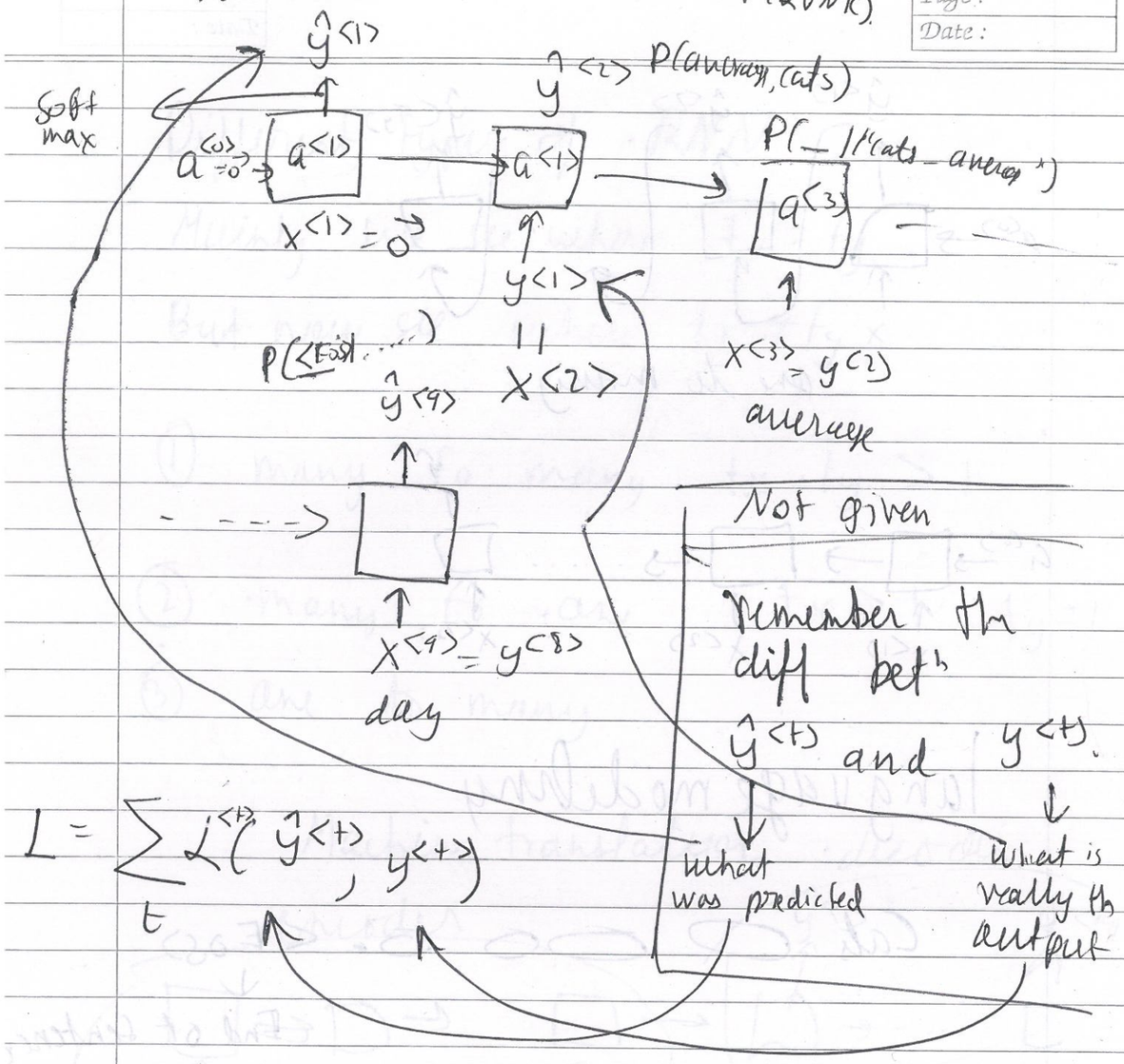
unknown word should be change (UNK)

maybe used sometimes.

The Egyptian ~~Mau~~ is a breed of cat.

<UNK>

$P(a)\ P(\cdots) \ldots P(cat) \ldots P(zulu)\ P(\langle UNK \rangle)$



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

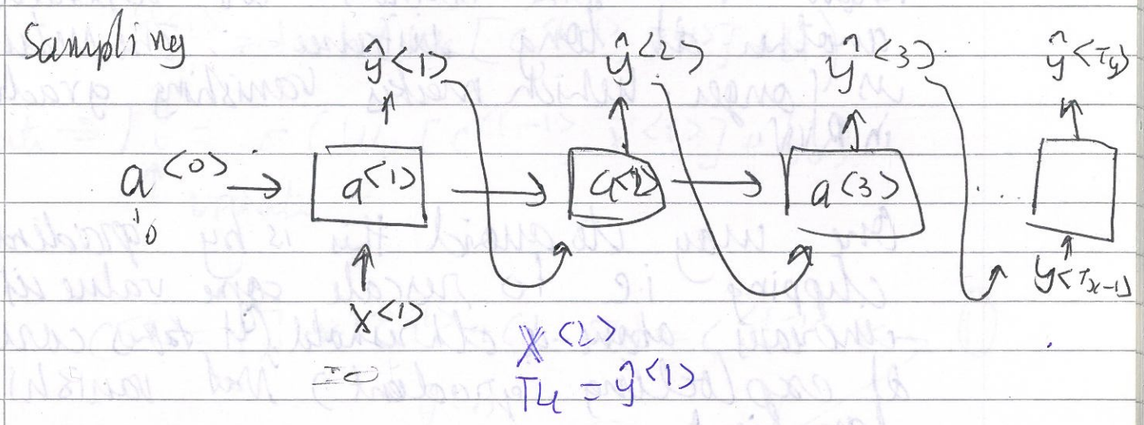
Also. $P(y^{<1>}, y^{<2>}, y^{<3>})$

$= P(y^{<1>})\ P(y^{<2>} / y^{<1>})$

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

Sampling Novel sequence.

Sampling



some parts in ~~vid~~ video left.

in

Character level token

vocabulary = [a, aaron, ... zulu, <UNK>]

This is when we make data dictionary ~~wo~~ with letters

Main Disadvantage → much longer sequences.

Not as good.

Not good in dependency.

Page :
Date :

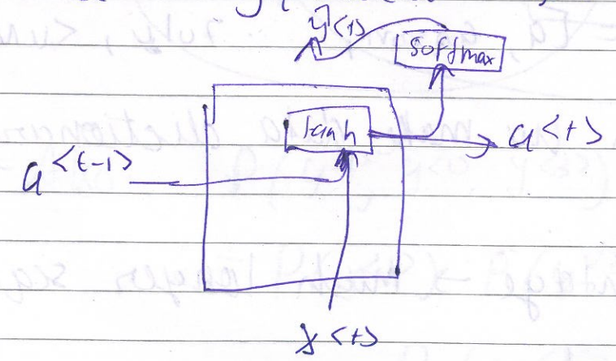
Vanishing Gradient in RNN

Sometimes while training RNN. As we know that some words are dependent on another at long distance ∴ The network is longer which makes vanishing gradient in RNN.

One way to avoid this is by gradient clipping i.e to rescale some value if it increases above a threshold. (It takes care of exploding gradient) Not vanishing gradient.

Gated Recurrent Unit (GRU)

$$a^{<t>} = g\left[W_a\left[a^{<t-1>}, x^{<t>}\right] + b_a\right]$$



Page :
Date :

c = memory cell.

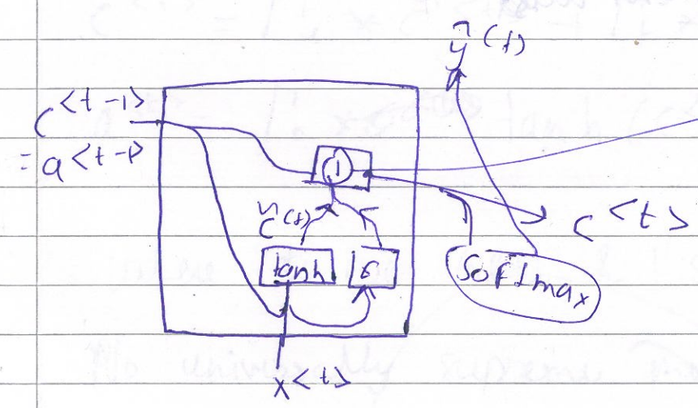
$$c^{<t>} = a^{<t>}$$

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

Update Gate → $\Gamma_u = \sigma\left(W_u\left[c^{<t-1>}, x^{<t>}\right] + b_u\right)$

↑ update

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



$c^{<t>}$ = can be a vector.
so would be Γ_u & $c^{<t>}$.
& * in (1) is element wise.

alternative notation

Full GRU

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

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

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

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

element wise

LSTM

9723456629

मनीकान्त

Page :
Date :

LSTM

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

update $\quad \Gamma_u = \sigma(W_u [a^{<t-1>}, x^{<t>}] + b_u)$

forget $\quad \Gamma_f = \sigma(W_f [a^{<t-1>}, x^{<t>}] + b_f)$

output $\quad \Gamma_o = \sigma(W_o [a^{<t-1>}, x^{<t>}] + b_o]$

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

$a^{<t>} = \Gamma_o * \tanh(c^{<t>})$

where to use GRU & LSTM

No universally supreme model.

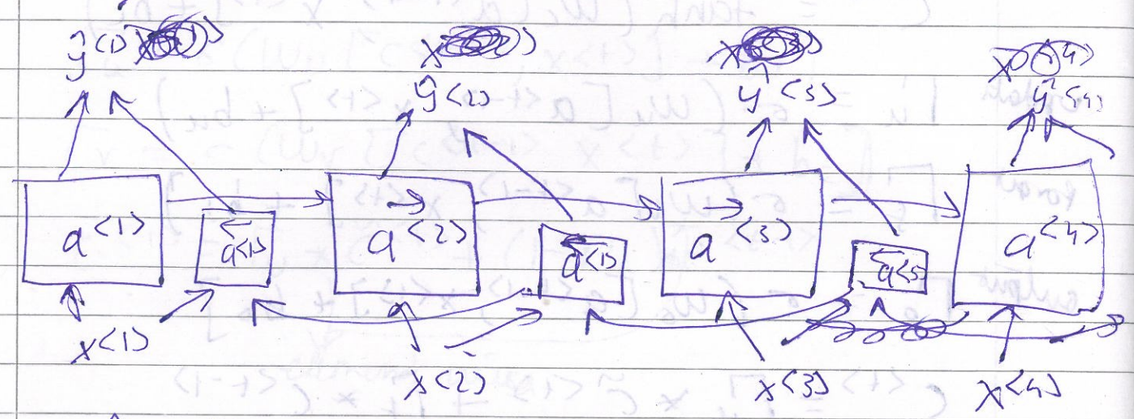
But according to author GRU is simpler and more quick. Able scale them for bigger problem

But LSTM is more powerfull

But LSTM is more proven choice.

Bidirectional RNN

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



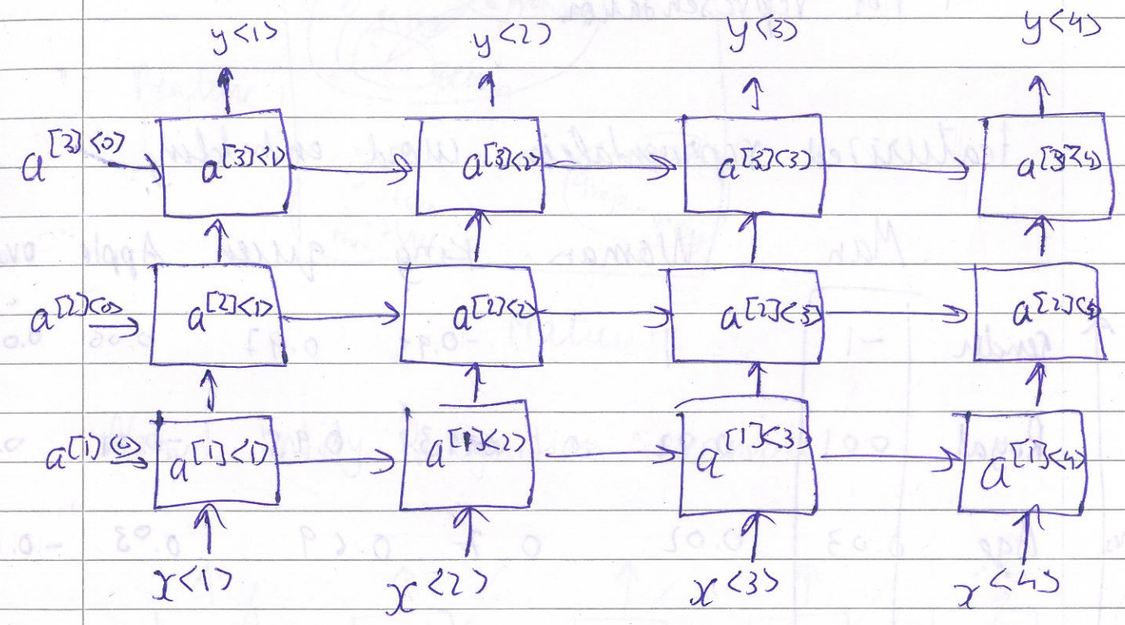
Acyclic graph.

Each Blocks can be GRU or LSTM.

Please review this in 2nd last video of 1 week.

Deep RNN

Example



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

Again each of this can be RNN, GRU, LSTM.

Word representation

V = [a, aaron, . . . zulu, <UNK>]

1 hot representation

Featurized representation: word embedding

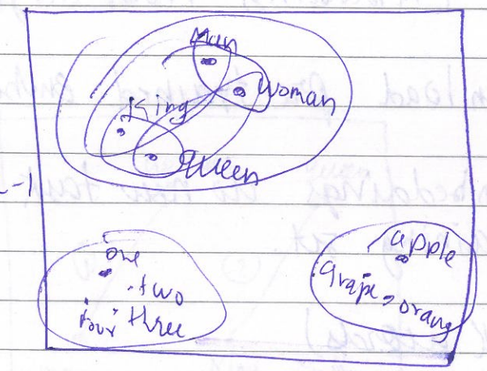
	Man	Woman	King	queen	Apple	orange
gender	-1	1	-0.95	0.97	0.00	0.01
Royal	0.01	0.02	0.93	0.95	-0.01	0.00
Age	0.03	0.02	0.7	0.69	0.03	-0.02
⋮						
lost size alive						

300 features

e_{5391}

This allows to generalize. (word embedding)

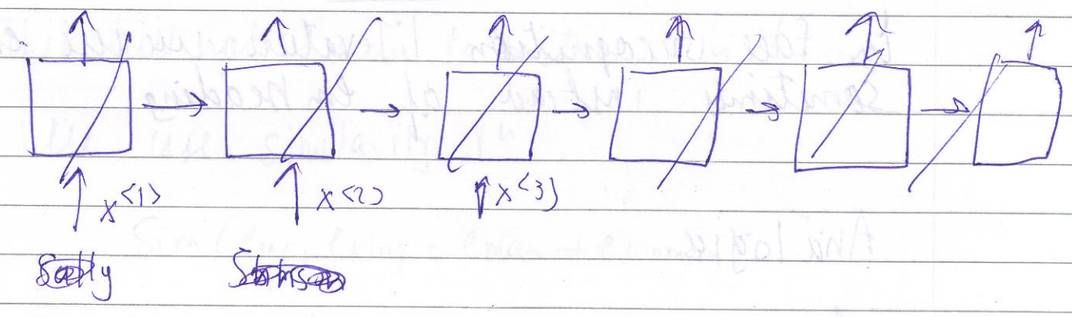
Visualizing word embeddings.



Feature-1

Feature-2

Named Entity Recognition example.



Page :
Date :

Transfer learning and word embeddings.

① learn word embeddings from large text corpus.

(or download pre-trained embedding online)

② Transfer embeddings to new task with smaller training set.

(say 100k words).

③ optional: continue to finetune ~~data~~ the word embeddings with new data.

Relation to face recognition

In face recognition literature we use encoding sometimes instead of embedding.

Analogies.

~~Man~~ ~~Woman~~

Using table in last page. We want to find

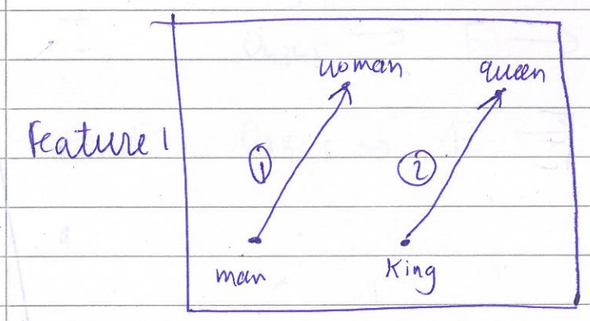
if man : woman :: King : ?

So we can do this by

Page :
Date :

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

Visualization



↑ is a vector.

$\vec{①} \approx \vec{②}$

This will not shown to be true in t-sne.

We use similarity f^n.

$Sim(e_w, e_{King} - e_{man} + e_{woman})$

It is called Cosine similarity

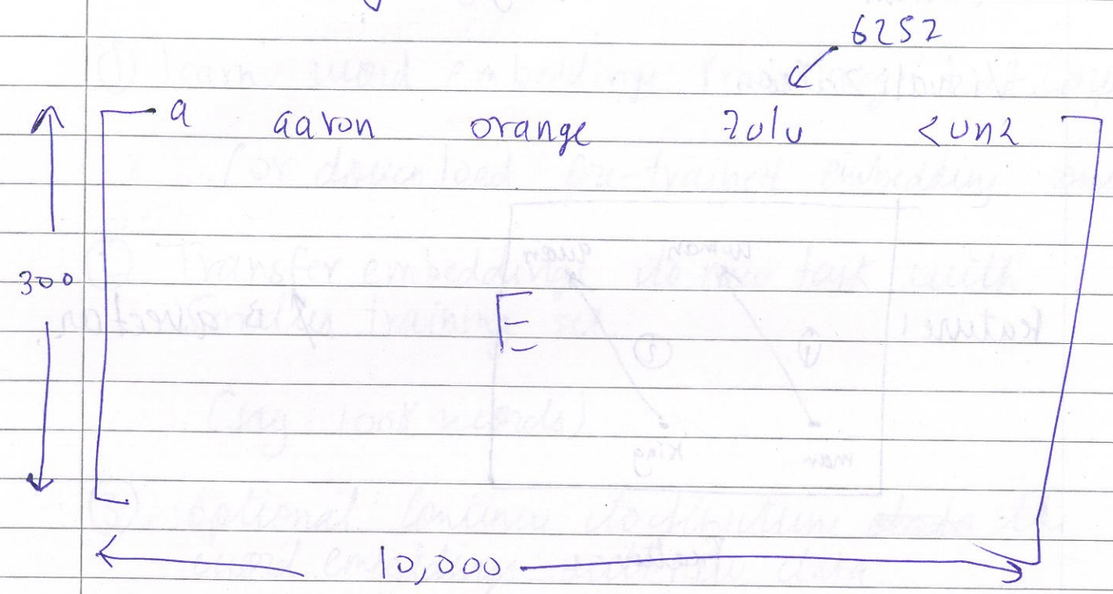
$$sim(u, v) = \frac{u^T v}{\|u\|_2 \|v\|_2}$$

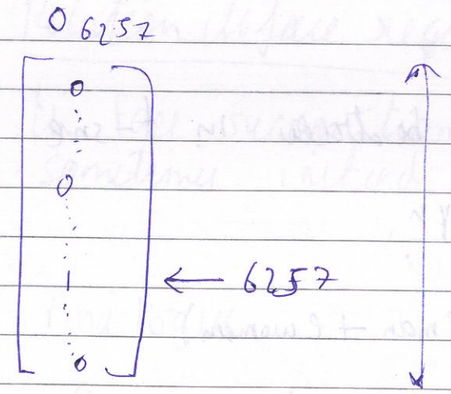
Euclidian distance.

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

Page :
Date :

Embedding Matrix.





$$\underset{(300,\ 10K)}{E} \cdot \underset{(10K,\ 1)}{O_{6257}} = \underset{(300,\ 1)}{\begin{bmatrix} \\ \\ \\ \end{bmatrix}} = e_{6257}$$

e_w = embedding for word w

In practice, use specialized fn to look up an embedding

Page :
Date :

learning word embeddings

I want a glass of orange ____

4343 9665 1 3852 6163 6257

This step many

I $O_{4343} \rightarrow E \rightarrow e_{4343}$ $e_{4343} = E \cdot O_{4343}$

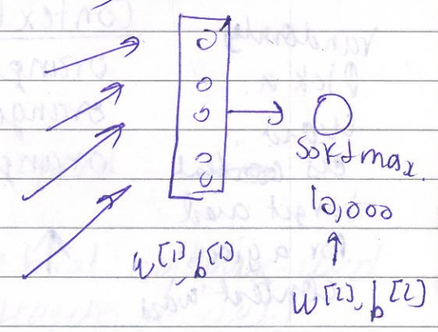
want $O_{9665} \rightarrow E \rightarrow e_{9665}$

a

glass

of

water



The above model selected a target word from the content before it.

i.e.

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

Context (for last 4 words only) — target

We could also select a 4 words of left and right and select a target word in middle.

Page :
Date :

Content example's.
- 4 words on left & right
- last 1 word
- Nearby 1 word.

Word 2Vec.

Skip Gram

Context	Target	
orange	juice	last example
orange	glass	
orange	my	

randomly Pick a word. as ~~context~~ target and For a given context word

So the target words are chosen ~~word~~ randomly near context word on training set.

Model

Vocab size = 10000

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

NN to learn embeddings

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

$e_c = E O_c$

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

Page :
Date :

θ_t = Parameters associated with output.

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

$$y = \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \leftarrow 4834.$$

Problem - computational speed.
↓
To solve this Hierachical softmax used. (search on web).

Negative Sampling

Defining a new learning problem

The eg is the ~~last~~ statement on previous to previous page.

Given 2 word a context and a word.

Context	Word	Target ?	
orange	juice	1	← +ve example
orange	King	0	
orange	of	0	

K

And identify that can it be target word. We generate a dataset like that.

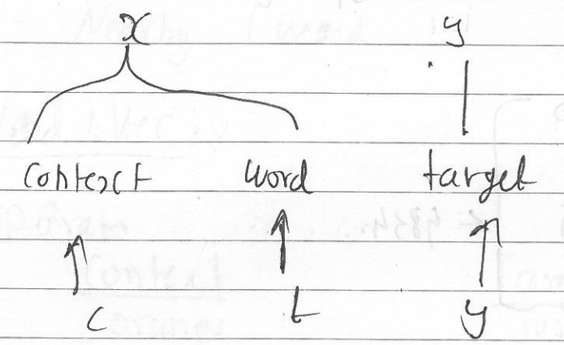
~~is that~~ is

K here is no of this target identification ~~from~~ of a given content word. excluding where target = 1

Please refer neg sampling in week 2

Page :
Date :

K = 5 – 20 smaller dataset.
K = 2 – 5 larger dataset



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

So while making the dataset. First we will select a content word. And than ±5 words ~~of each~~ can be selected for an of the → a +ve example.
for K – ve example we selected content word can be paired with any word in dictionary
In last page content orange would of but still its 0. in target even though its nearby content word. The reason being that it was selected randomly and blindly assigned target value of 0.

Orange
6257

$O_{6257} \rightarrow E \rightarrow e_{6257}$ → 0, 0, 0 → 10,000

Page :
Date :

Now in last fig last layer which has 10000 nodes we are only going to train on (K+1) of them (i.e one +ve target value & K –ve values)

∴ having one giant 10000 softmax we convert it into (K+1) logistic binary classification.

How to sample –ve example for training set.

So probability of a –ve target word for a particular context is given by. (

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

$f(w_i)$ = observed frequency of that ith word in english language

Glove word Vectors (global vector for word representation)

c, t.

X_{ij} = # times j appears in context of i
(c, t) (t) (c)

$X_{ij} = X_{ji}$ (mostly, here define as wheather i & j word are in close proximity of each other)

$$\text{Minimize} \sum_{i=1}^{10000} \sum_{j=1}^{10000} f(X_{ij}) \left(\underbrace{\theta_i^T e_j + b_i + b_j}_{\text{"}\theta_t^T e_c\text{"}} - \log X_{ij} \right)^2$$

(Not to sum over terms which has $X_{ij} = 0$.)

weighted term

$f(X_{ij}) = 0$ of $X_{ij} = 0$ as $\log 0 = \infty$ ∴ to avoid

Here θ_j e_j are symmetric

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

Now training Algo of word embedings we can't guarantee that axis represent the features will be well aligned with what might be easily humanly interpretable axis

Sentiment classification problem

x sentence $\longrightarrow$ y rating.

Simple sentiment classification model.

The dessert is excellent.

The $O_{8928} \rightarrow E \rightarrow e_{8928}$

dessert $O_{2468} \rightarrow E \rightarrow e_{2468}$

is $O_{4694} \rightarrow E \rightarrow e_{4694}$

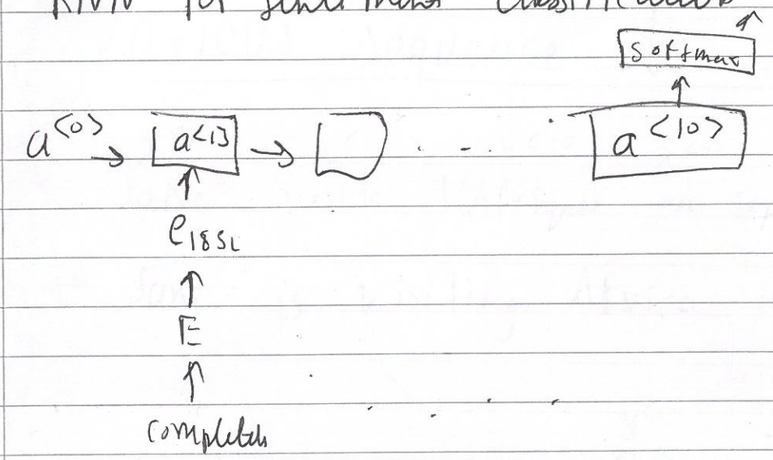
excellent $O_{3180} \rightarrow E \rightarrow e_{3180}$

$\rightarrow$ $\boxed{AVG}$ 300D $\rightarrow o \rightarrow \hat{y}$ softmax 1-5

disadvantage

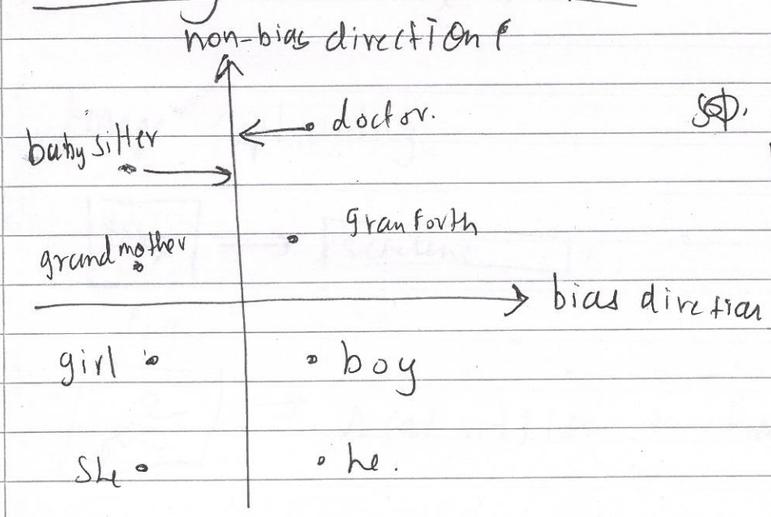
- ignores word order.

RNN for sentiment classification



many to one, architecture

Debasing word embedings



Steps -1

1. Identify the bias direction

$e_{he} - e_{she}$
$e_{male} - e_{female}$
$\vdots$
$\rightarrow$ average

2. Neutralization for every word that is not definational Project to get rid of bias.

3. Equalize pairs

watch the video to get intuition

Page :
Date :

Lore

Page :
Date :

Various Sequence to sequence Architecture

$x^{<1>}$ $x^{<2>}$ $x^{<3>}$ $x^{<4>}$ $x^{<5>}$

→ Jane visite l'Afrique en septembre

→ Jane is visiting Africa in septembre

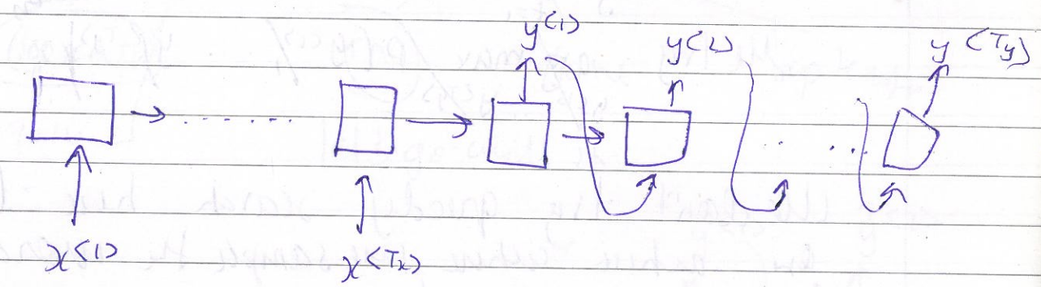
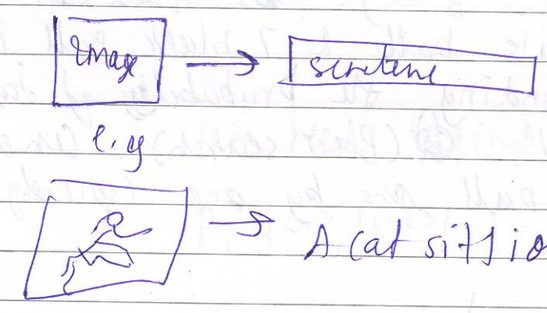


Image captioning.



Machine translation as building a conditional language model.

In last fig we try to find $\underbrace{P(y^{<1>}, \ldots, y^{<T_y>}}_{\text{English translation}} | \underset{\text{french}}{x^{<1>} \ldots x^{<T_x>}})$

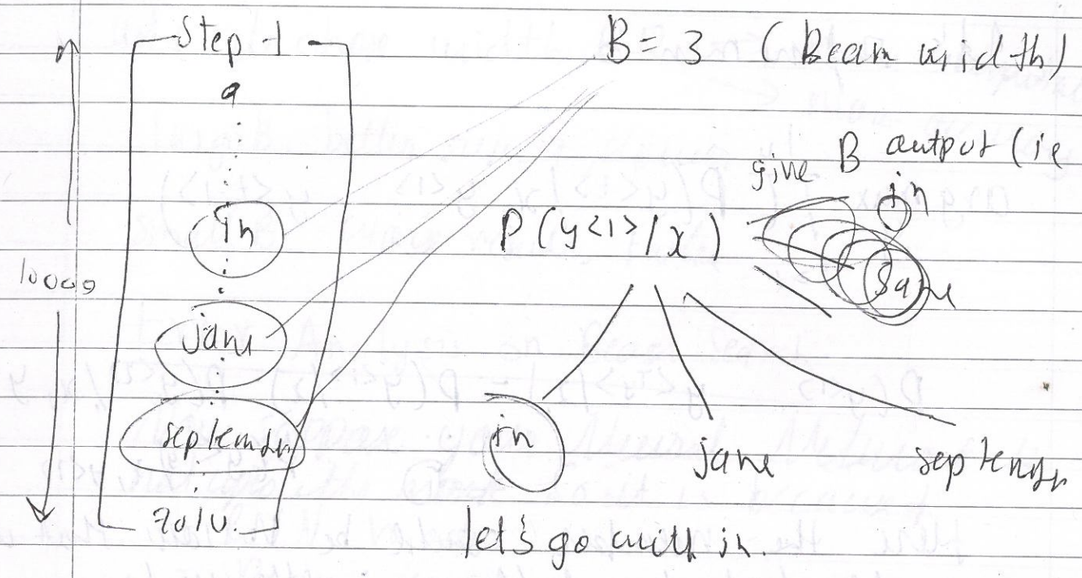
$$\arg\max_{y} \prod_{t=1}^{T_y} P(y^{<t>} | \ldots)$$

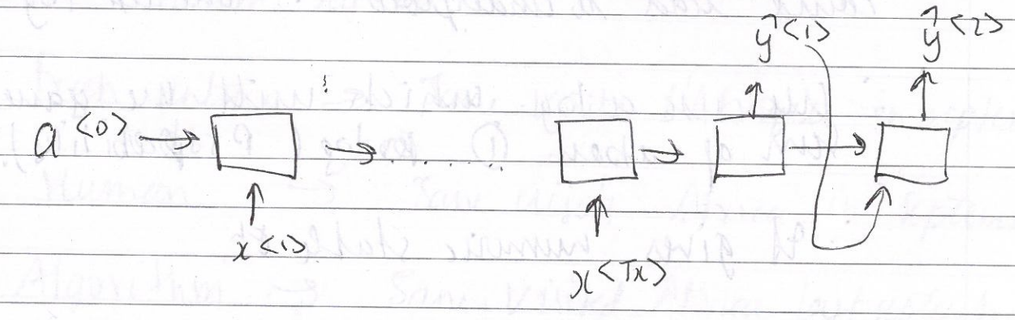
We try $\arg\max_{y^{<1>} \ldots y^{<T_y>}} P(y^{<1>}, \ldots y^{<T_y>} | x)$

We don't try greedy search here (i.e the one where we sample the words in the previous weeks.

But the problem is that greedy search gives $P(y^{<n>} | y^{<n-1>} \ldots y^{<1>})$ ~~that~~ It's like the 2 white balls & 2 black ball in a bag. It's like finding the probability of taking out 2 black ball's. (Bleu search). and taking out 2 black ball one by one. (greedy search)

Beam Search





We need to find.

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

Please see beam search video as it difficult to explain it.

remember if $b=1$ it becomes greedy search

Page :
Date :

It's refinement

$$\arg\max_{y} \prod_{t=1}^{T_y} \left(P(y^{<t>} / x, y^{<1>} \ldots y^{<t-1>}) \right)$$

$$P(y^{<1>} \ldots y^{<T_y>} / x) = P(y^{<1>}/x) \cdot P(y^{<2>}/x, y^{<1>}) \cdots P(y^{<T_y>} / x, y^{<1>} \ldots y^{<T_y-1>}) \quad \text{— (1)}$$

Here the number could be to low that it could lead to underflow. ∴ We use log

∴ We use a log which will give us sum of above (1) log (Propability).

∴ It gives numeric stable sh.

$$\arg\max \sum_{y=1}^{T_y} \log P(y^{<t>} / x, y^{<1>} \ldots y^{<t-1>})$$

Here we see that ~~it can always~~ it would prefer short items (as in (1) multiplication used always < 1).

∴ We use

$$\frac{1}{T_y^{\alpha}} \sum_{t=1}^{T_y} \log P(y^{<t>} / x, y^{<1>} \ldots, y^{<t-1>})$$

Page :
Date :

Who to chose width B ? → more computation
→ more accuracy

large B: better result, slower

small B: more result, faster

Error Analysis on Beam Search

Now suppose your Neural Network is not upto the mark. So it is because of
- RNN model
- Beam search

x French sentence → Jane visite l'Afrique en septembre

y^* Human → Jane visits Africa in September.

$\hat{y}$ Algorithm → Jane visited Africa last september.

RNN computes $P(\hat{y} / x)$

RNN

Now get $P(y^* / x)$

length Normalization not taken into consideration.

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

then increase the size of beam search

else

work on RNN

Page :
Date :

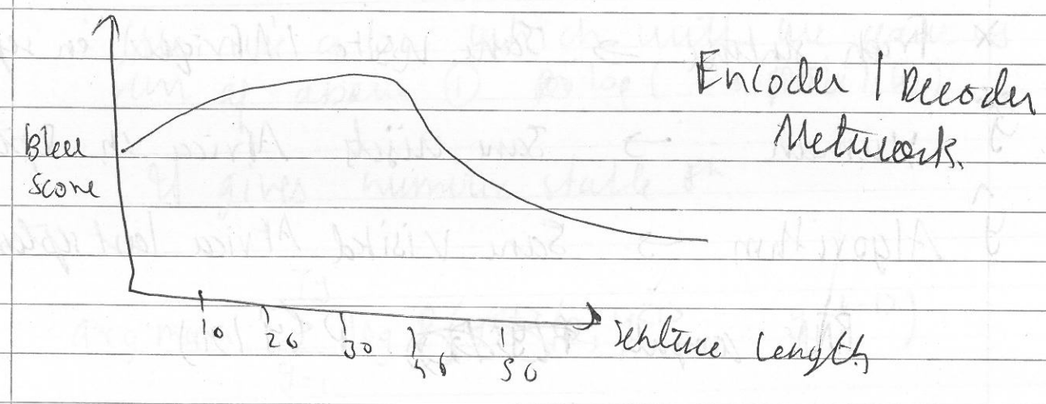
Bleu Score

~~How to con~~ what to do if their are multiple great answer. How measure Accuracy

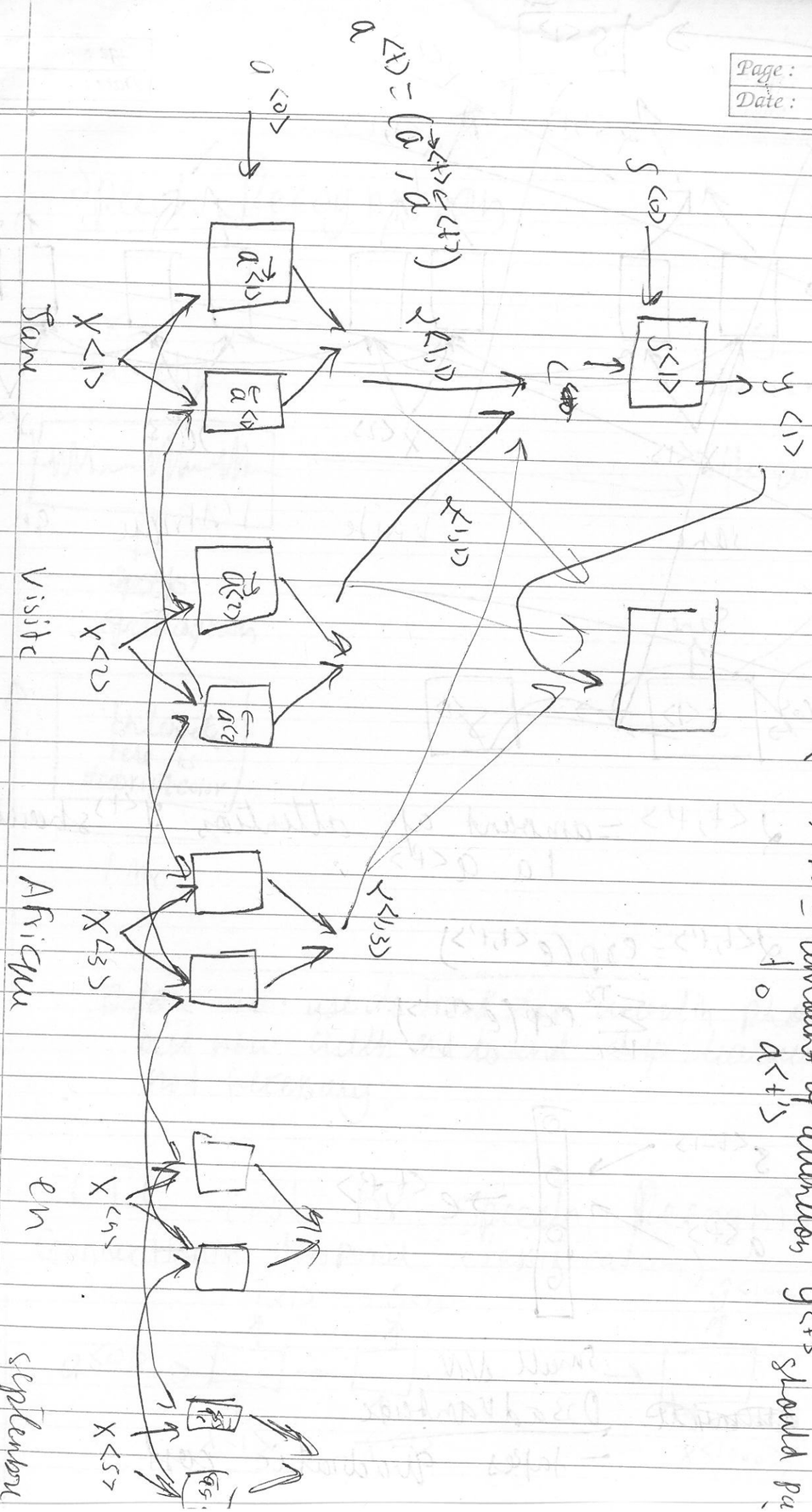
Bleu → Bilingual evalution understudy.

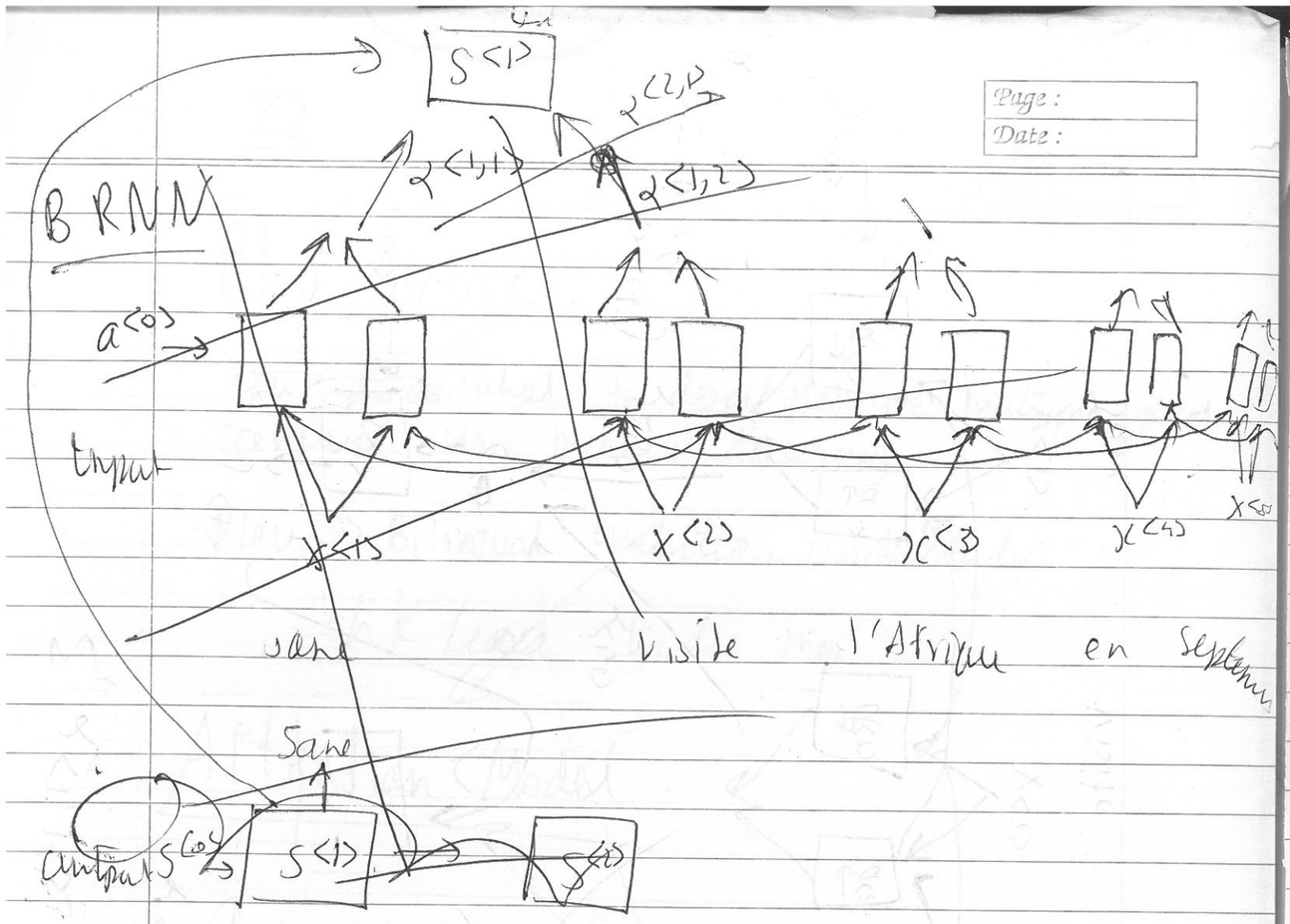
Lehk Learn. Video mai

Attention Model



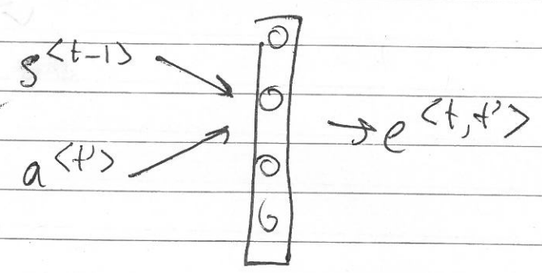
Page :
Date :





$\alpha^{<t,t'>}$ = amount of attention $y^{<t>}$ should pay to $a^{<t'>}$

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



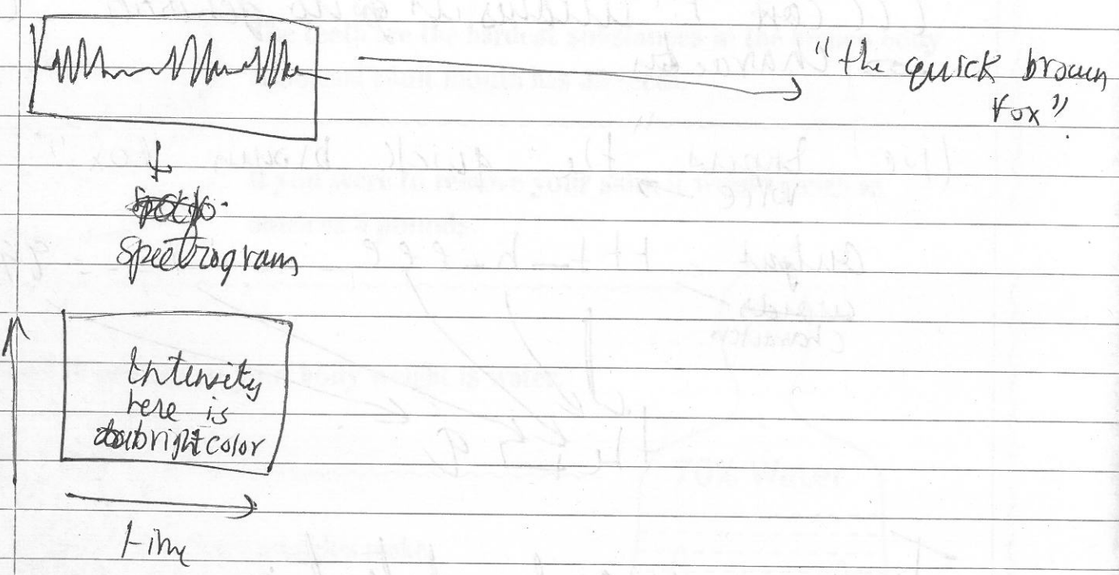
Small NN.

~~Disadvantage~~ Disadvantage
- takes quadratic cost.

Speech Recognition

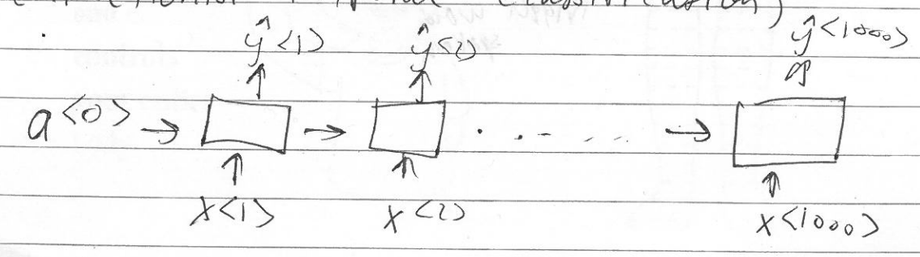
x
audio clip

y
transcript



Before we use to break the word in phonemes but now with end to end deep learning its not necessary.

CTC cost for Speech Recognition.
(Connectionist temporal classification)



Page :
Date :

Like we may have 1000 input and just 10 output. (i.e 1000 notes & 10 output is 10 words).
input.

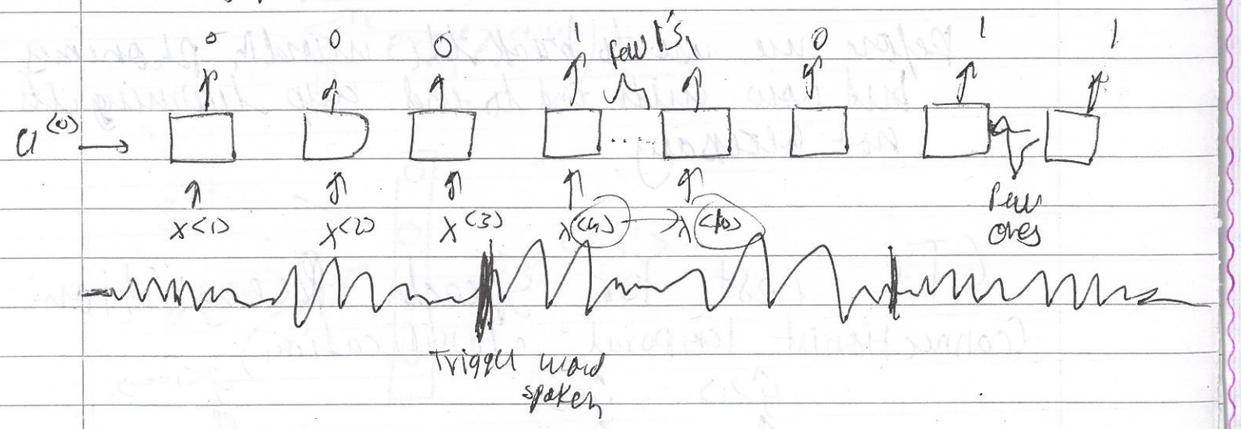
CTC cost F^n allows it go to generate — (Blank) ∞ characters.

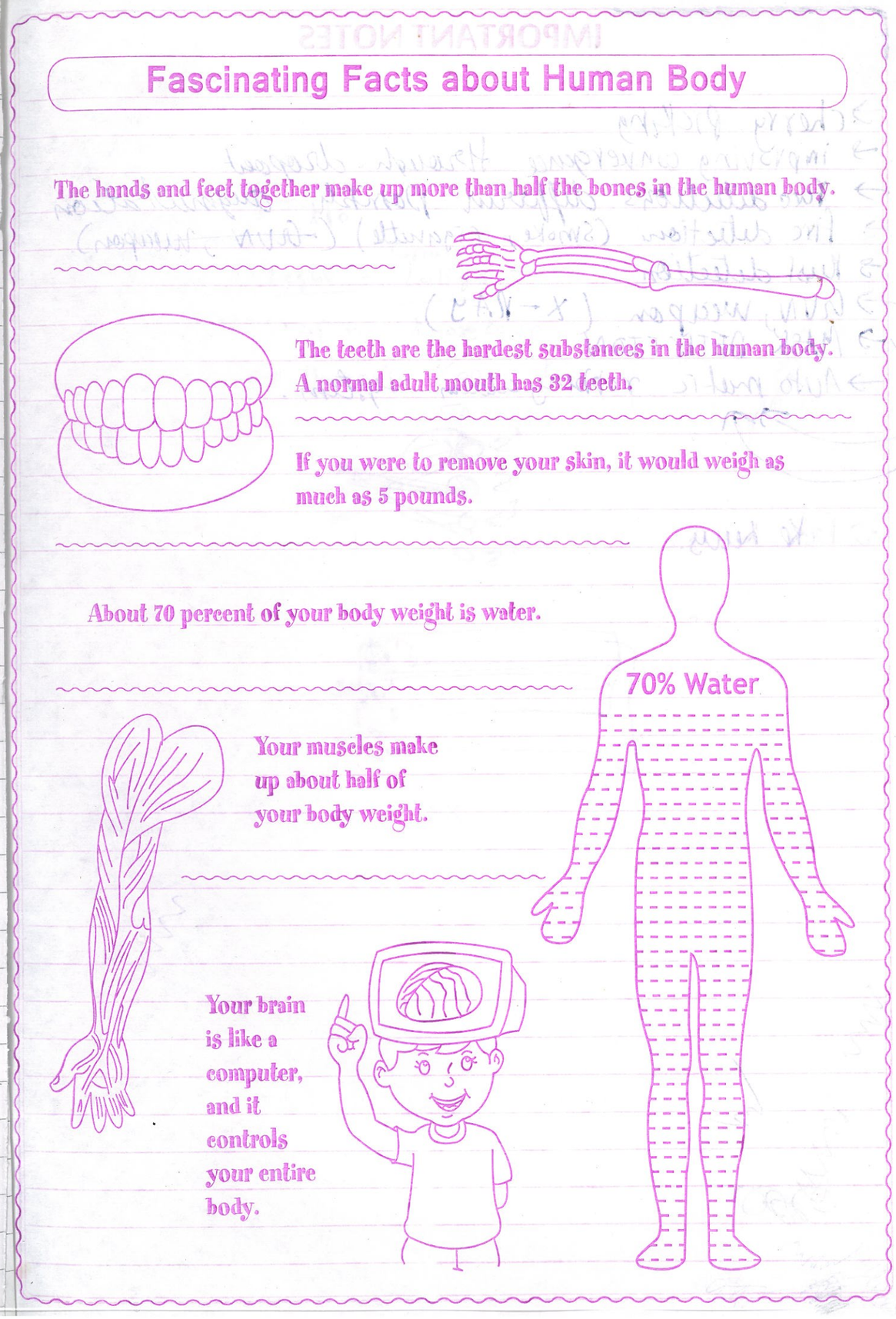
(i.e input voice → "the quick brown fox"

output ~~words~~ characters — t t t _ h _ e e e _ _ _ _ _ q q q _ _ _

t h e _ q

Trigger word detection





IMPORTANT NOTES

- → cherry picking
- → improving convergence through dropout
- → face detection's different possibly augmentation
- → fire detection (smoke, cigarette) (~~GUN, weapon~~).
- → Rust detection
- → GUN, weapon (X-RAY)
- → MASK DETECTION
- → Automatic robbery alarm system.

- → Fake News.

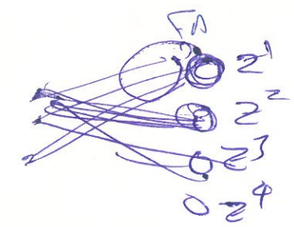
Title

Top

1 Legue — Spain Laligs —

→ 1 Chapin Knowkown

Go Domesti Copa K



$$\begin{bmatrix} w_1 \\ \\ w_{n_1} \end{bmatrix} \qquad \begin{bmatrix} w^1 \\ w^2 \\ w^3 \\ w^4 \end{bmatrix}$$

$w^T X$ $\qquad$ wX