(1) Course 5:- Week 1

PAGE :
DATE : / /

Examples of Sequence data.

(1) Speech Recognition
(2) Music generation
(3) Sentiment classification
DNA analysis
Machine translation
Video Activity Recognition
Name Entity recognition.

Another Example:-

Name Recognition in a sentense

x: Harry Potter and Hermione Granger invented a new spell

$x^{<1>}$ $x^{<2>}$ $x^{<3>}$ $x^{<4>}$ .... $x^{<9>}$

$T_x = 9$ bcz 9 words.

y

1 1 0 1 1 0 0 0 0

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

Representing words.

Vocabulary Ex:- Harry Potter a

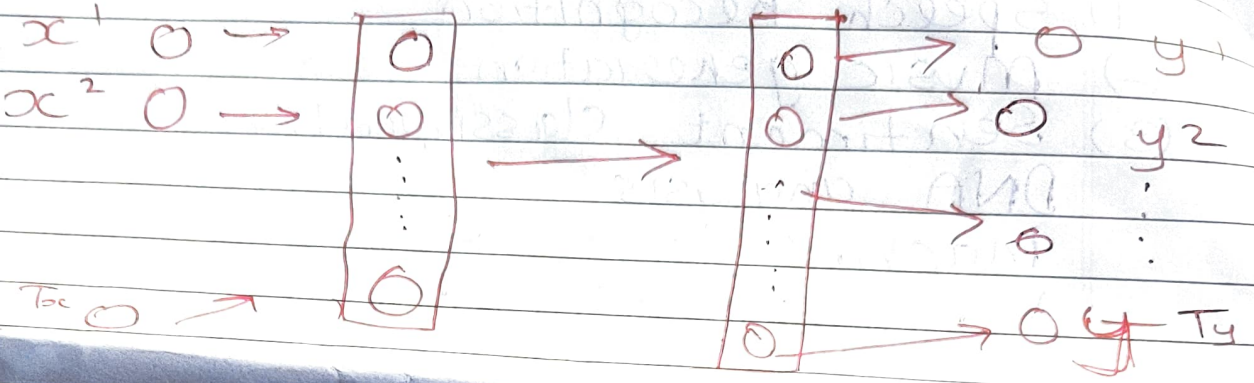
x^1 x^2 x^7

x

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

10,000

Why don't we use a standard neural Network.

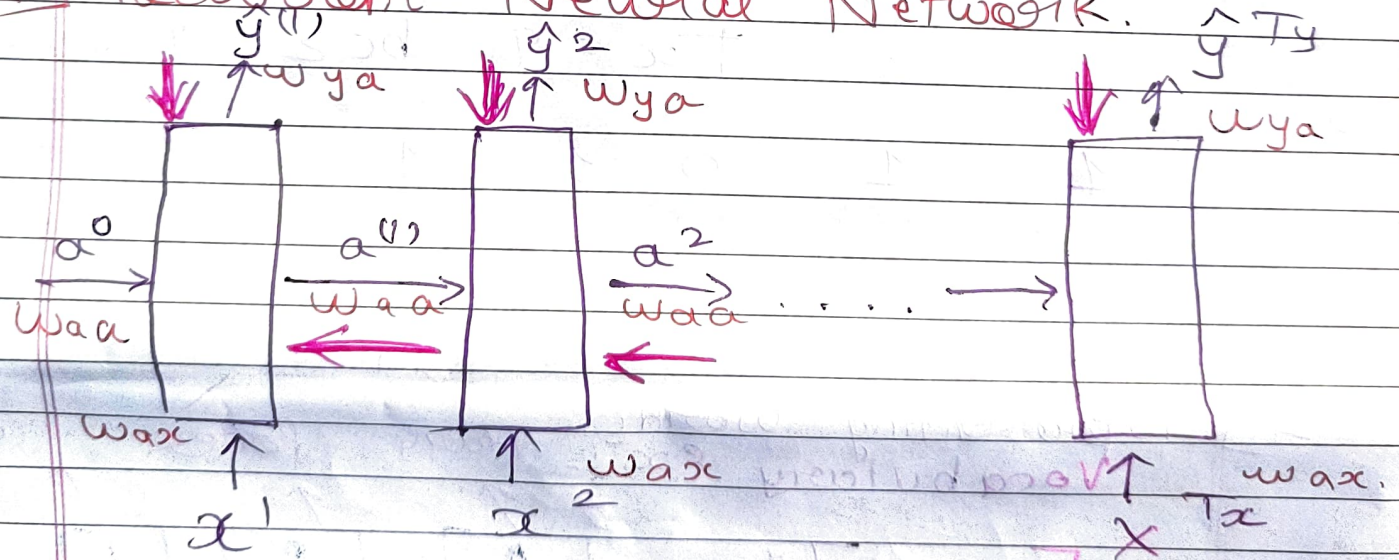


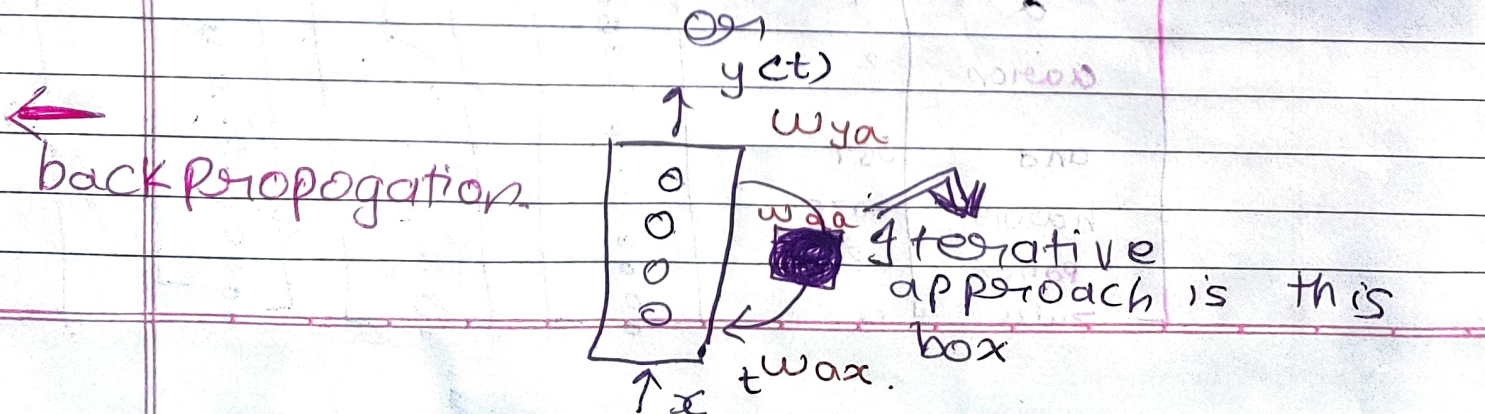
Problems:-

(1) Inputs and outputs can be different length in different examples

(2) Doesn't share features learned across different position of text

Recurrent Neural Network.





PAGE :
DATE : / /

In above diagram

tanh / Relu

$a^0 = \vec{0}$ $\quad a^1 = g(W_{aa} a^0 + W_{ax} x^1 + b_a)$

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

Sigmoid

General Notation.

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

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

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

$W_a = \begin{bmatrix} W_{aa} \vdots W_{ax} \end{bmatrix}$

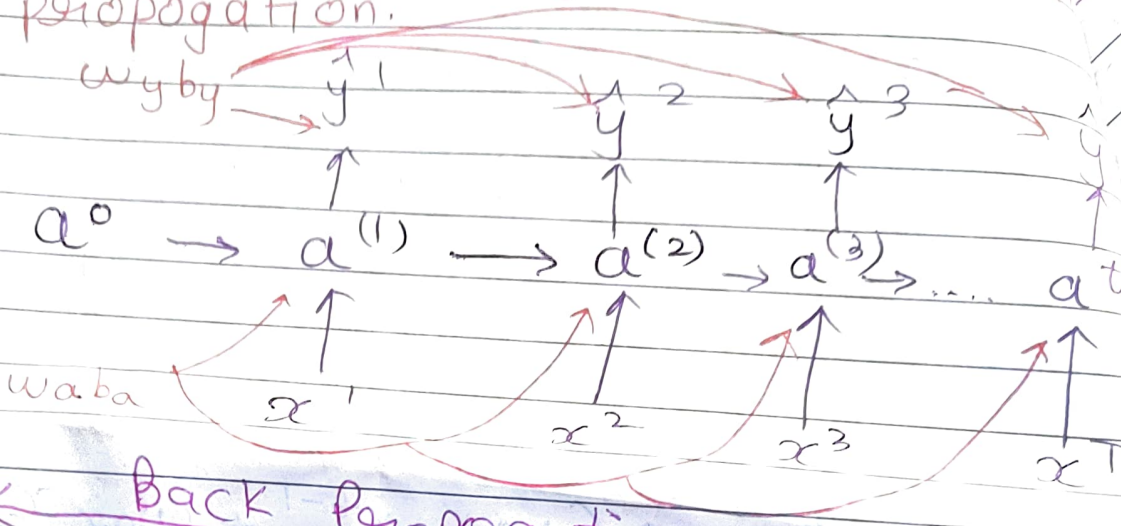
(100, 10100)

100 $\quad$ 10000

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

(10100, 1)

Forward Propogation and back propogation.



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

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

Note:- It's not necessary that T_x and T_y are similar in length or data type.

T_x can be voice.

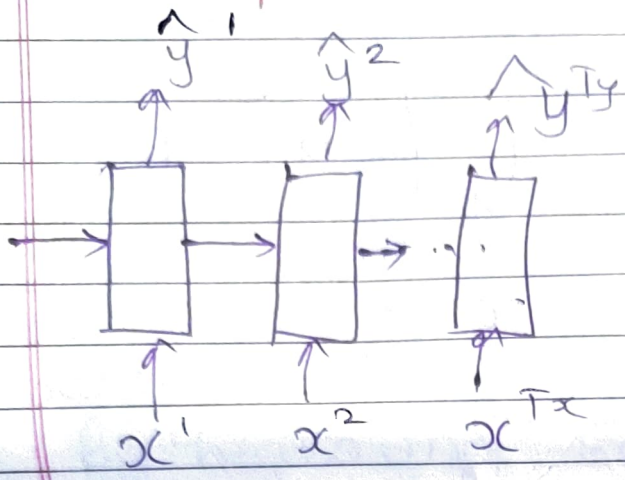
T_y can be speech recognition.

Let's see On Next Page

PAGE :
DATE : / /

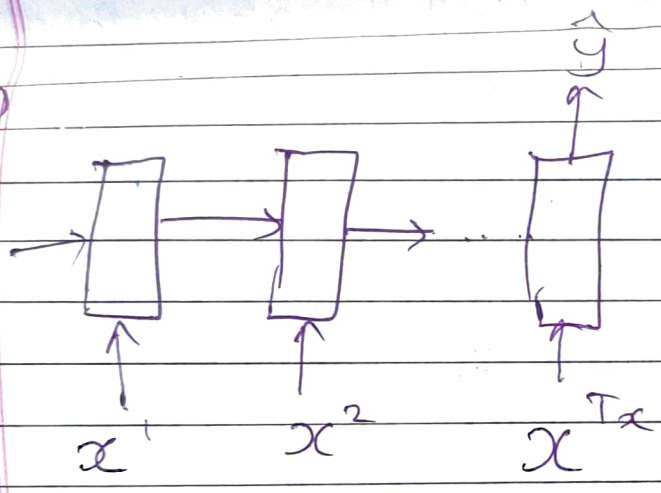
Example of RNN architecture

(1)



Many to Many

(2)



Many to One

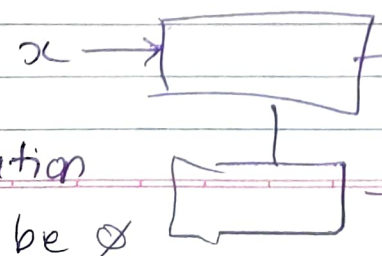
example you take input from user what was his opinion on movie. Based on his language you give 1 or 0
1 → Good experie
0 → Bad

(3)

Normal Neural Network

One to One

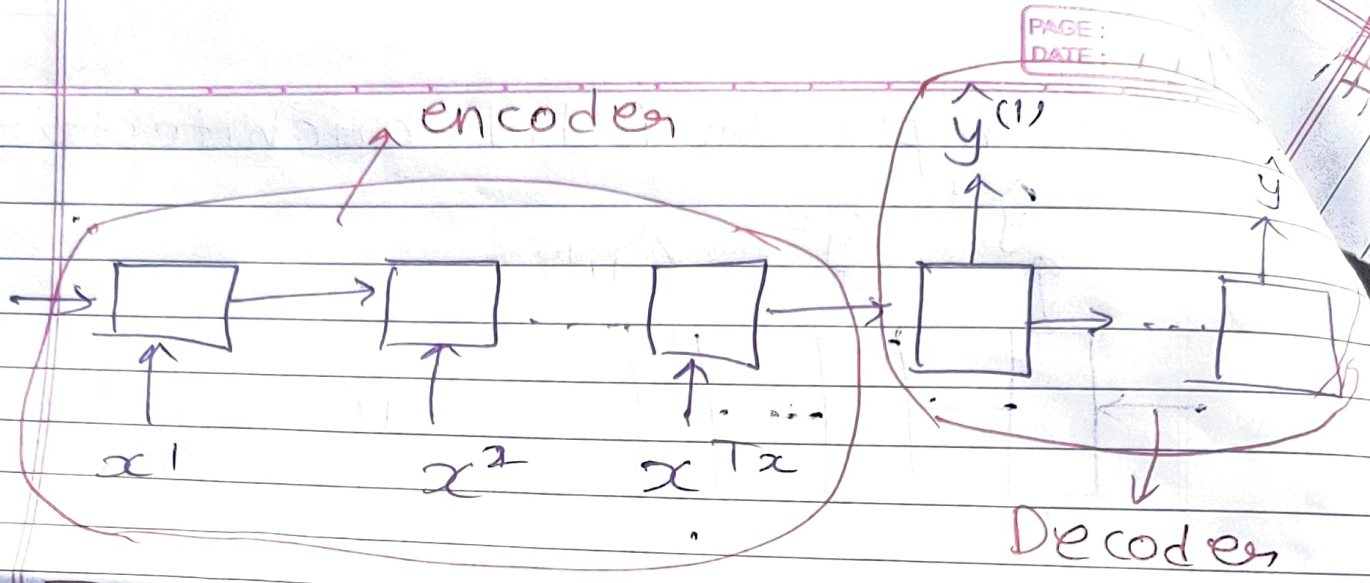
(4)



One to Many

Music generation

(5) x can be $\varnothing$



This is popularly used in Machine Translation

PAGE :
DATE : / /

What is language Modelling.

(1) Ex :- In speech Recognition.

1st sent The Apple and pair salad

2nd sent The apple and pear salad.

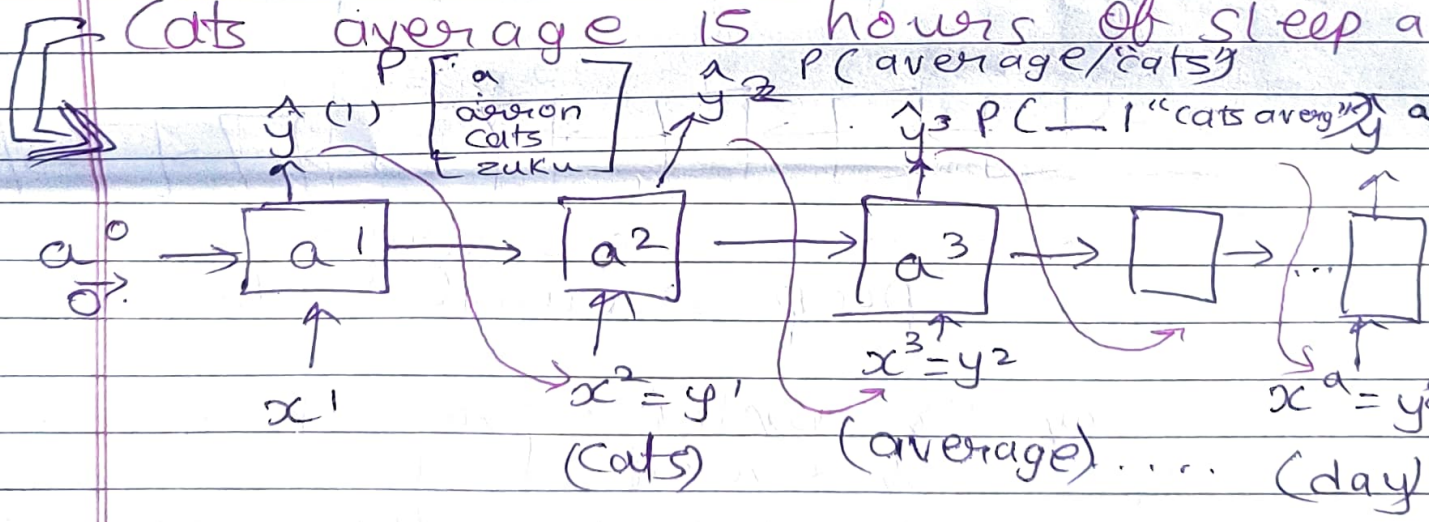
Both sound same however 2nd sentence is printed bcz it has more probability and more logical sense in it.

Language modelling mainly deals with Probability of words in a sentence.

Ex :- P (1st sentence) = 3.2×10
P (2nd sentence) = 5.7×10^{-10}

⇒ This sentence is first Broken into token

Cats average is hours of sleep a



$P_y^{(1)} \rightarrow P(y^2/y^1) \rightarrow P(y^3 | y^1 y^2)$

$\ldots\ldots \leftarrow P(y^4 | y^1 y^2 y^3)$

<UNK> = Unknown token

<EOS> = End of Sentence

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

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

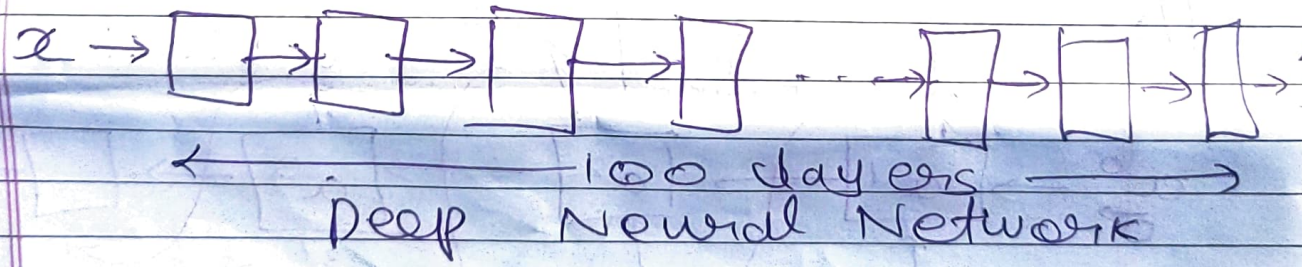
$y^{(1)}$ = cats

$y^{(2)}$ = cats Average

y^3 = cats Average is

y^8 = cats Average is hours of steep day

Problem with RNNs is vanishing gradient problem

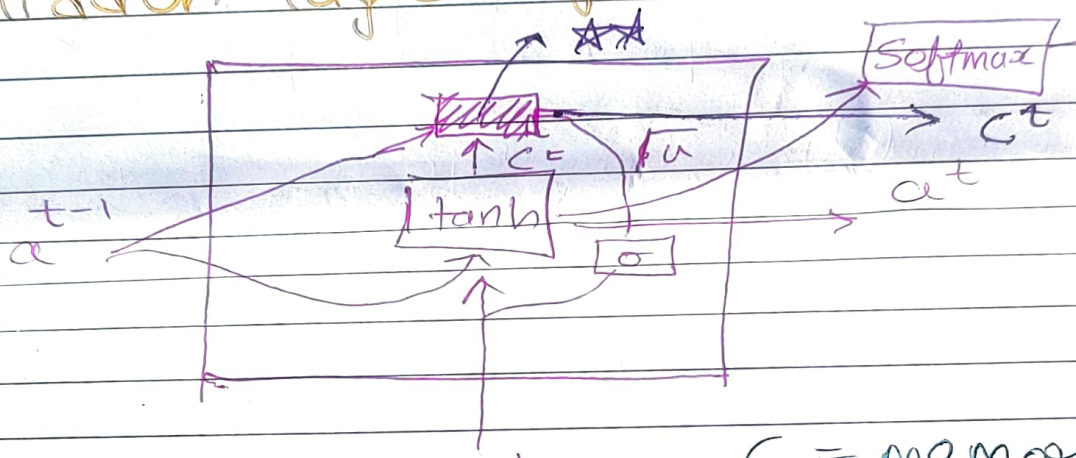


When the sentence proceeds further it is observed that new characters produced are less dependent and often overloop the meaning of words used previously thereby creating errors in sentence gramatical or logical

PAGE :
DATE : / /

When we train Deep Neural Network we often develop Vanishing and exploding gradient problem

Hidden layer of RNN unit.



Gated Recurrent Unit (GRU)

C = memory cell which remember that a^t is singular or plural so rules like that which are gramatically correct.

$$c^t = a^t$$

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

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

Gamma
u = for update

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

$$\{[(1+1)+2]+3\}\{[(0+1)*2]+3\}$$

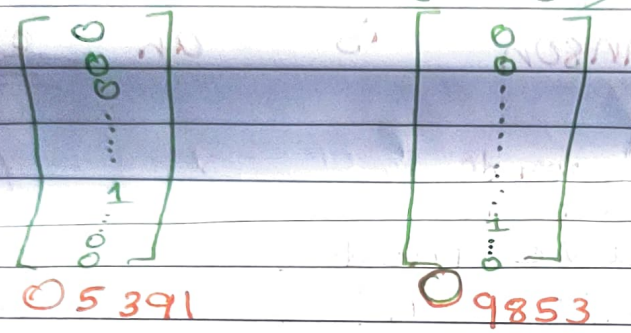
Course 5:-
Week 2:-

WORLD STAR™ Date: Page:

Word representation.

1-hot representation

Man (5391) — Woman (9853)

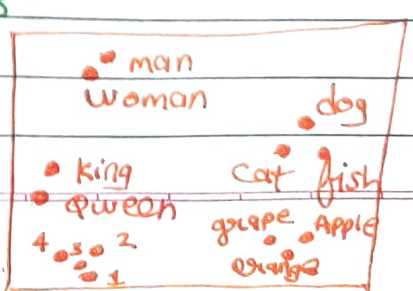


Ex:-

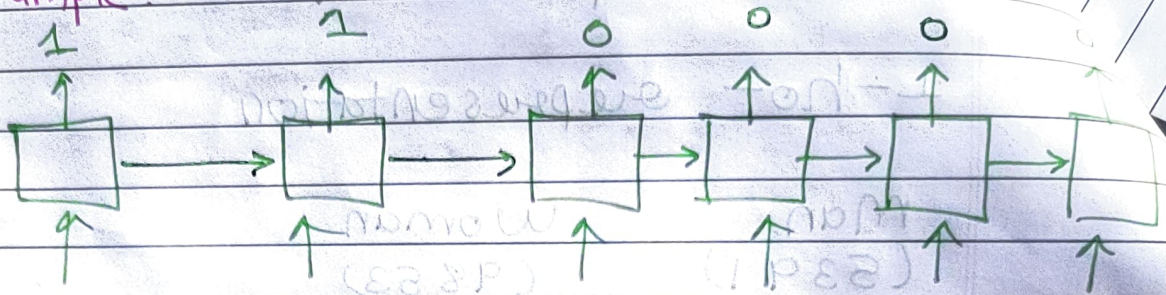
	Man	Woman	King	Queen	Apple	e
	5391	9853	4914	7157	456	
Gender	-1	1	-0.95	0.97	0	
Royal	0.01	0.02	0.93	0.95	-0.01	0.
Age	0.03	0.02	0.7	0.69	0.03	-0
Food	0.09	0.01	0.02	0.01	0.95	0.97
:	:	:	:	:	:	:
	300 parameters					

e_{5391} e_{9853}

Visual word Embeddings



Example of Named entity recognition example



1B to 100B words of data set to train above model.

Based on this sentence we can understand that orange can be replaced by another fruit.

Learning and word embeddings.

(1) Learning words from large text corpus (1-100B) words.

(2) Transfer embedding to new task with smaller training set ex 100k words

We Reduce the words set and other things even more based on need and also

ANALOGIES

Man and woman
King and ___ ?

(1) Analogy using matrix

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

-2	→ Gender
0	→ Royal
0	→ Age
0	→ Food

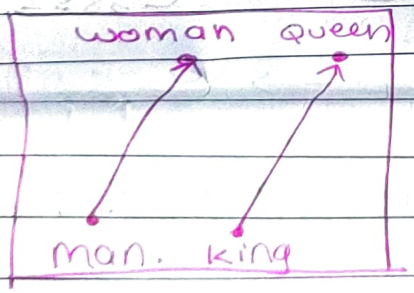
$e_{king} - e_{queen} \approx$

-2	→ Gender
0	→ Royal
0	→ Age
6	→ Food.

$\therefore e_{man} - e_{woman} \approx e_{king} - e_i$

$\therefore i \approx$ Queen based on predictio

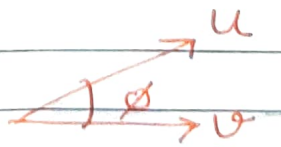
(2) Analogy using word vectors.



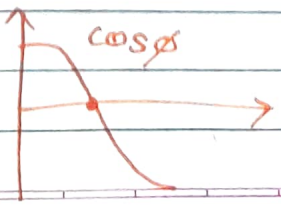
Sim $(e_i, e_{king} - e_{man} + e_{woman})$

$i \rightarrow$ queen

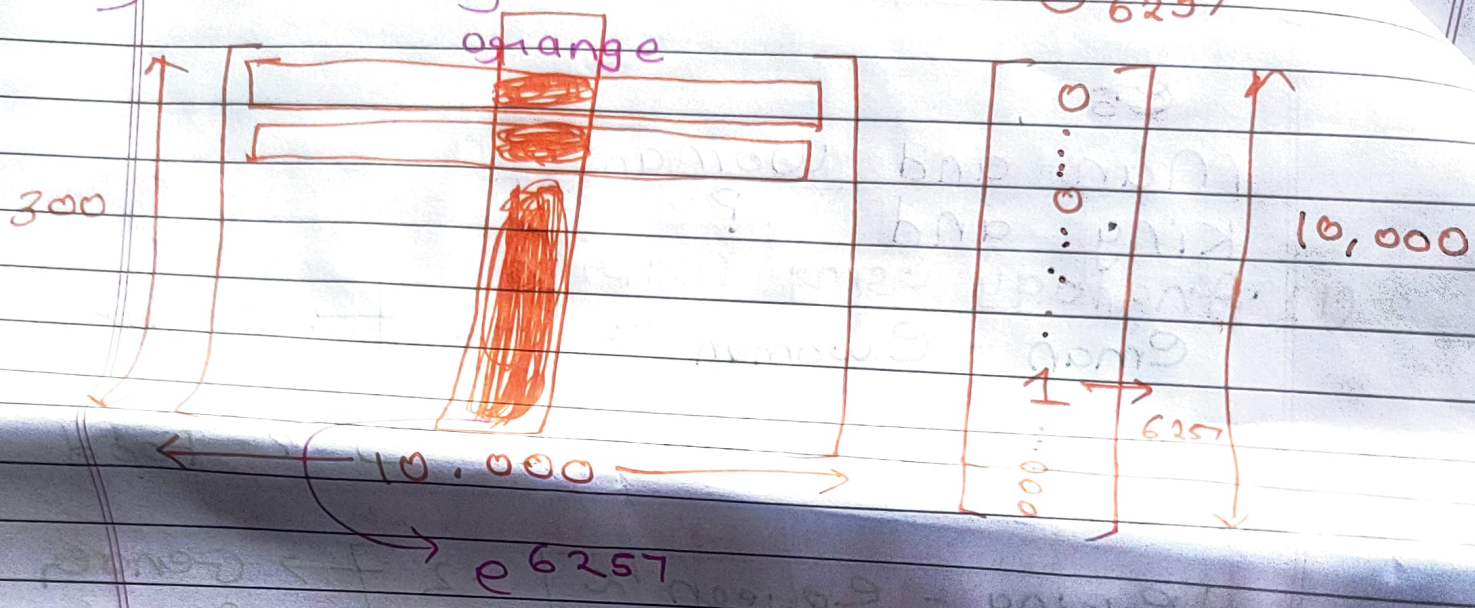
$Sim(u, v) = \frac{u^T v}{\|u\|_2 \|v\|_2}$



$\|u - v\|^2$



Embedding Matrix



Neutral Language modelling

4 $O_{4343} \rightarrow E \rightarrow e_{4343}$

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

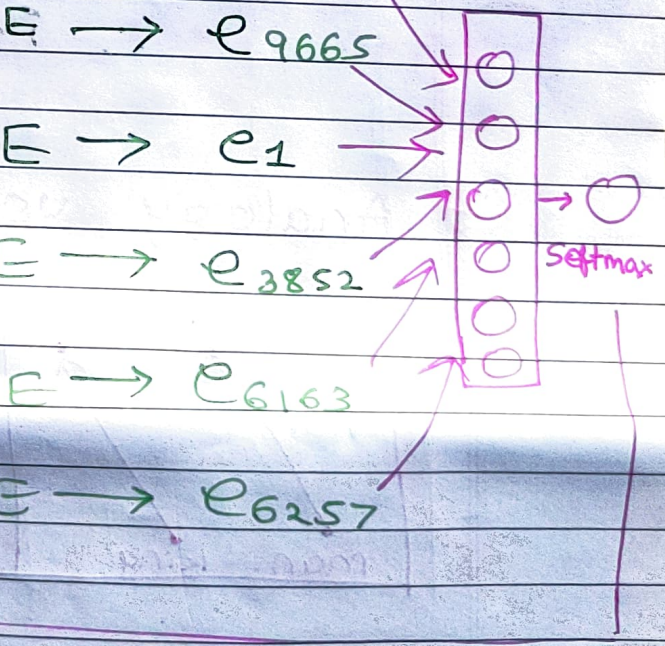
a $O_1 \rightarrow E \rightarrow e_1$

glass $O_{3852} \rightarrow E \rightarrow e_{3852}$

of $O_{6163} \rightarrow E \rightarrow e_{6163}$

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

Juice



WORLD STAR™
Date :
Page :

Other context / Target Pairs

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

Context last 4 words.
a glass of orange ____ to go alg with.

last 1 word.
orange ____

Model

Vocab size = 10000

x → y
context target
(c) t

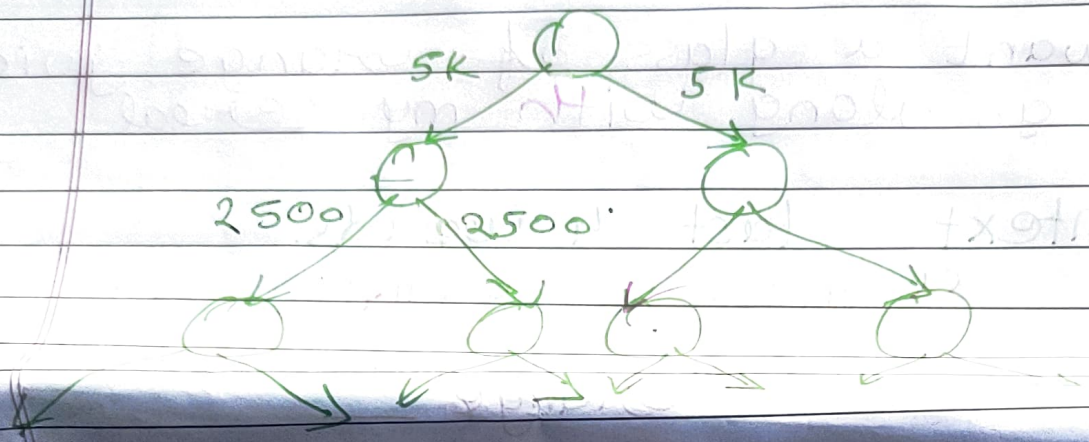
orange	glass
orange	juice
orange	my

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

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

WORLD STAR™
Date :
Page :

Hierchial Softmax



How to sample the context 'c'
→ The, of, a, and to, ...
→ orange, apple, division

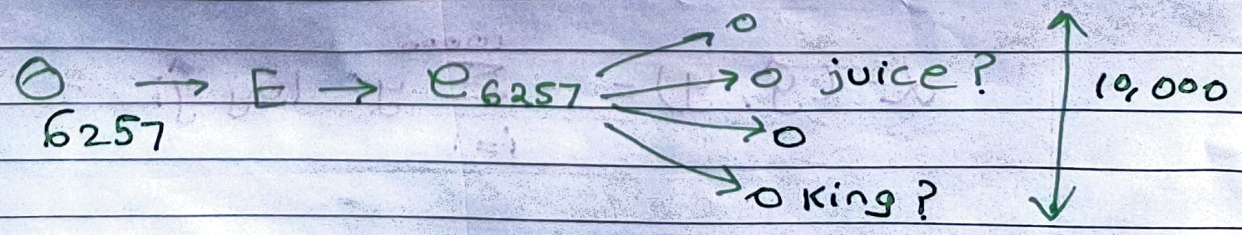
Negative Sampling

Context	word	target
orange	juice	1
orange	king	0
orange	book	0
orange	the	0

K

c t y

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



Glove (Global Vectors for Word Representation)

X_{ij} = i appears in context of j

c t t c

$X_{ij} = X_{ji}$

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

Problem of bias in word Embeddings

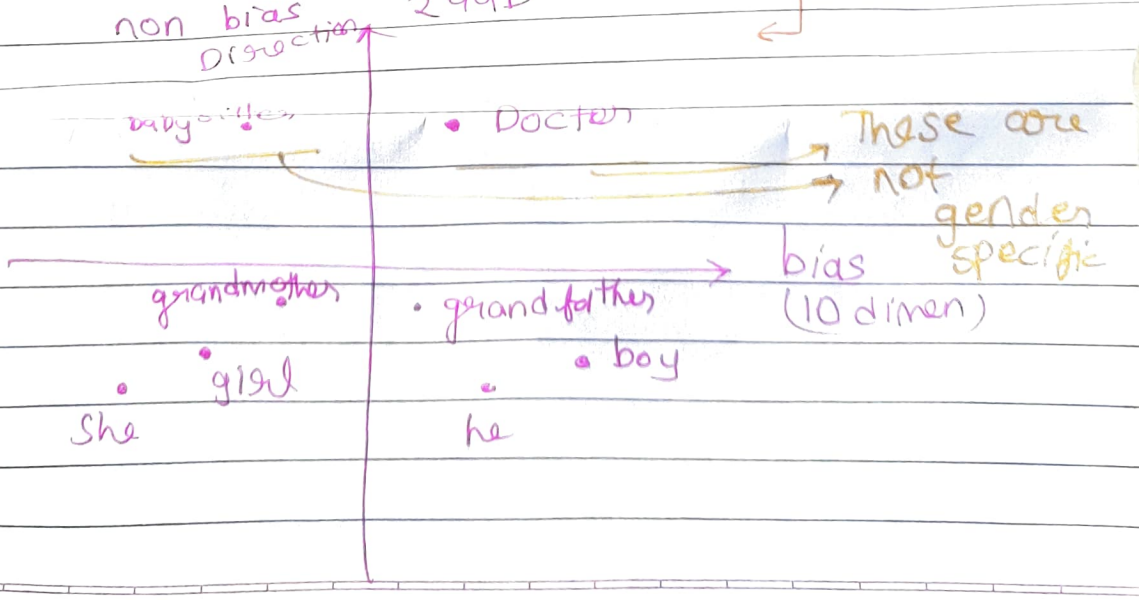
Man x woman as King x Queen ✓

Father x Doctor as Mother x Nurse ⇒ ✗

⇓

This problem is arising here

To avoid this we use



① Course 5 Week 3:-

Why not Greedy Search

Greed Search means printing the words in a line which are most connected to each other i.e with highest probability however this is not at all useful bcz we need to make sentence based on conditions, constraits and input.

Beam search

Length Normalization

$$\underset{y}{\operatorname{argmax}} \prod_{t=1}^{T_y} P(y^t \mid x, y^1, \ldots, y^{t-1})$$

$$\arg\max_y \sum_{t=1}^{T_y} \log P(y^t \mid x, y^1, \ldots, y^{t-1})$$

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

Beam width B

large B: better result, slower

smaller B: worse result, faster

Beam search runs faster but is not guaranteed to find exact

minimum for $\arg\max P(y|x)$.

Error Analysis Process.

Human	Algorithm	$P(y^*\|x)$	$P(\hat{y}\|x)$	Fault ?
Jane visits Africa in September	Jane visited Africa last september	2×10^{-10}	1×10^{-10}	Beam at fault

$\therefore y^* > \hat{y}$

If $y^* < \hat{y}$ then RNN at fault.

Attention Model Intution.

Course 5 Week 4

In RNNs GRUs, LSTMs the complexity of ~~Transform~~ model keeps on increasing

Hence we developed Transformers. which works on
- Attention + CNN
  - Self-Attention
  - Multi-Head Attention

A(q, K, V) = Attention based Vector representation of a word
- Calculate for each word.

$A^1, A^2 \ldots A^n \Rightarrow n$ words have 1 Attention each

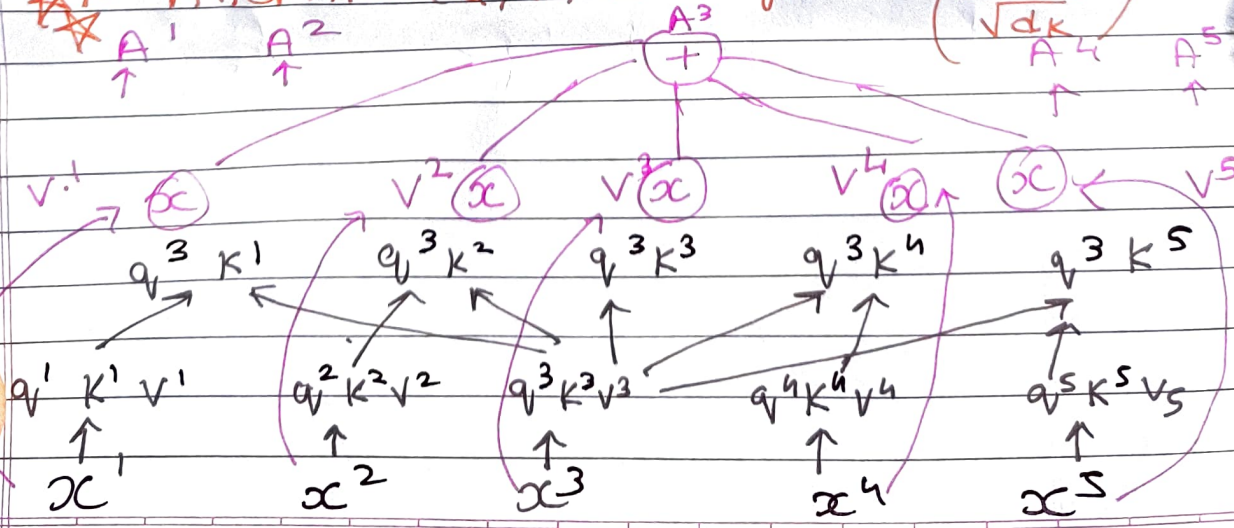
RNN Attention.

$$a^{t\,t'} = \frac{\exp(e^{t,t'})}{\sum_{t'=1}^{Tx} \exp(e^{t\,t'})}$$

Transformers Attention

$$A(q, K, V) = \frac{\sum_i \exp(q \cdot k^i)}{\sum_j \exp(q \cdot k^j)} \times v^i$$

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$



Jane visite l'Afrique en septembre

WORLD STAR™
Date :
Page :

Attention → queries, Key and Values

- queries $q_1, q_2 \ldots q_T$ $\quad q_i \in R^d$
- Keys $K_1, K_2 \ldots K_T$ $\quad K_i \in R^d$
- values $V_1, V_2 \ldots V_T$ $\quad V_i \in R^d$

Vectors in a given dimension.

Number of ~~Keys~~ queries can differ from the number of keys and values in practice.

In self-Attention q, K, V are drawn from same source.

The output of previous layer is $x_1 \ldots, x_T$ (One vector per word)

we can let $q_i = K_i = V_i = x_i$ [Can use same vectors for then]

$e_{ij} = q_i^T K_j$

$T \times T$ matrix with scalar values not bounded by size

Compute Key - query affinities

$$a_{ij} = \frac{\exp(e_{ij})}{\sum_{j'} \exp(e_{ij'})}$$

Normalisation constant

[Compute attention weights from affinities (Softmax)]

WORLD STAR™
Date :
Page :

output $i = \sum_j a_{ij} V_j$ → going to weigh out things importance

Compute outputs as weighted sum of values.

Attention treats each words representation as a query to access and incorporate information from set of values.

* Why cannot we have Self-Attention itself as a building block i.e. a replacement of RNNs and LSTM why need of Transformer.

↓

we don't use bcz the self Attention model does not accept input in particular order like

The Chef makes food

and the food Chef makes

is the same for it even though there is lot of difference. it does not give incidies to q_i, k_i, v_i no impertance of value of i

To solve above problem we use

$P_i \in R^d$ for $i \{1, 2 \ldots T\}$

↳ position vectors.

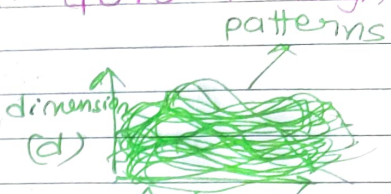
- $\tilde{q}_i, \tilde{v}_i, \tilde{k}_i$ are our old values

$$q_i = \tilde{q}_i + P_i$$
$$v_i = \tilde{v}_i + P_i$$
$$k_i = \tilde{k}_i + P_i$$

} you can do many other things.

vectors.

* Position representation done through sinosudial.



$$P_i = \begin{bmatrix} \sin(i/10000^{2*1/d}) \\ \cos(i/10000^{2*1/d}) \\ \vdots \\ \sin(i/10000^{(2*\frac{d}{2})/d}) \\ \cos(i/10000^{(2*\frac{d}{2})/d}) \end{bmatrix}$$

Pros:-

(1) Periodicity indicates that maybe absolute position isn't as important

(2) Can explolate to longer sequences as period starts.

Cons :-

(1) Not Learnable and the extrapolation doesn't really work.

We have modified P_i to make it learnable

$P \in R^{d \times T}$ ← Most systems use this one

let each P_i be a column of that matrix.

Pros:-

(1) gets to be learned to fit the data

2

Cons:-

(2) Cannot extrapolate to indices outside $1, \ldots, T$

WORLD STAR™
Date :
Page :

Problem 2 :-

Note :- In self Attention - there are no element-wise non-linearities in self-Attention; we have more self-Attention layers just to re-averages value vectors.

This can be easily resolved by a feed-forward network to post process each output vector.

Problem 3 :-

We cannot make sentence based on the knowledge what we are going to produce in future. This is hypothetical or impossible This happens then no need to train network.

→ so to inhibit it from predicting future based on present.

i.e to enable parellization we mask out attention to future words by seeting attention scores to $-\infty$

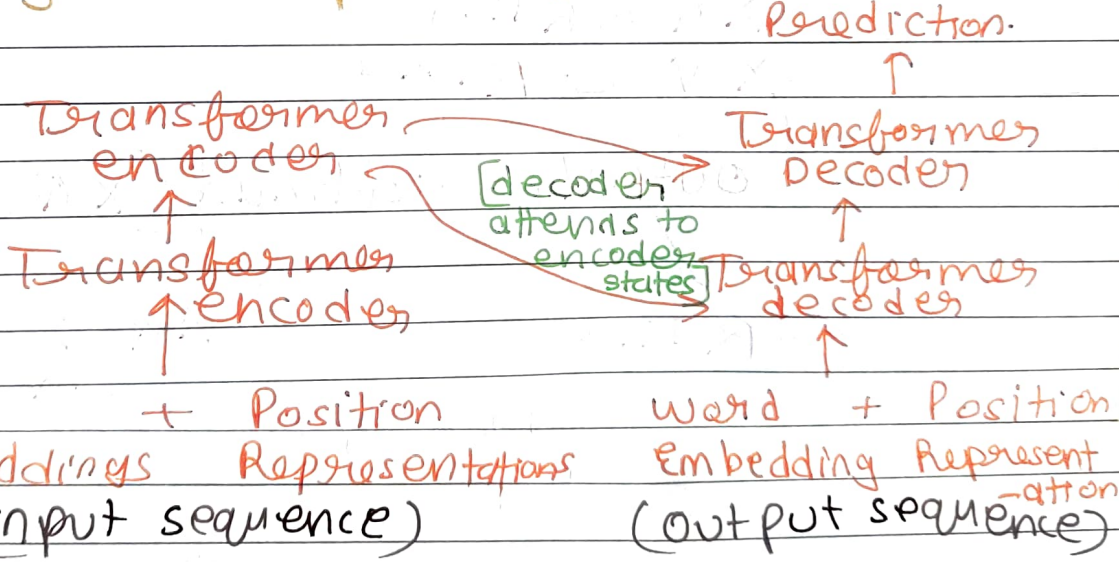
$$e_{ij} = \begin{cases} q_i^T k_j, & k < j \\ -\infty & k > j \end{cases}$$

	start	The	chef	who
start	$-\infty$	$-\infty$	$-\infty$	$-\infty$
The		$-\infty$	$-\infty$	$-\infty$
chef			$-\infty$	$-\infty$
who				$-\infty$

Necessities for building a self-Attention building block.

(1) Self-Attention.
(2) Position Representations
(3) Non-linearities with simple fed forward network. (will be at output of self attention block)
(4) Masking
   - On order to parallelize operations while not looking at the future.
   - Keeps the information about the future from "Leaking" to the past.

Above model is not a transformer but just a part of that.



K, q, v come from same source in transformer
Let $x_1, \ldots x_T$ be input vecters to the transformer encoder $x_i \in R^d$

$k_i = K x_i$ where $K \in R^{d \times d}$ is Key matrix

$q_i = Q x_i$ $Q, V \in R^{d \times d}$ is query and

$v_i = V x_i$ value matrix.

Transformer Encoders: Key - Query Value Attention

Let's see how q, v attention is computed in matrices.

Let $X = [x_1 \ldots . x_T] \in R^{T \times d}$ be the concatenation of input vectors.

$XK \in R^{T \times d}$

$XQ \in R^{T \times d}$

$XV \in R^{T \times d}$

$$\text{output} = \text{Softmax}(XQ(XK)^T) * XV$$

$\in R^{T \times d}$

$XQ K^T X^T \in R^{T \times T}$