DEEP LEARNING SPECIALIZATION
COURSE 5/5

SEQUENCE MODELS

→ EXAMPLES OF SEQUENCE DATA

* SPEECH RECOGNITION, MUSIC GENERATION, SENTIMENT CLASSIFICATION, DNA SEQUENCE ANALYSIS, MACHINE TRANSLATION, VIDEO ACTIVITY RECOGNITION, NAME ENTITY RECOGNITION

→ NOTATION

* MOTIVATION EXAMPLE: NAME-ENTITY RECOGNITION

X: HARRY | POTTER | AND | HERMIONE | GRANGER | INVENTED | A | NEW | SPELL

$x^{<1>}$ $x^{<2>}$ ..... $x^{<t>}$ ..... $x^{<9>}$

$T_x = 9$

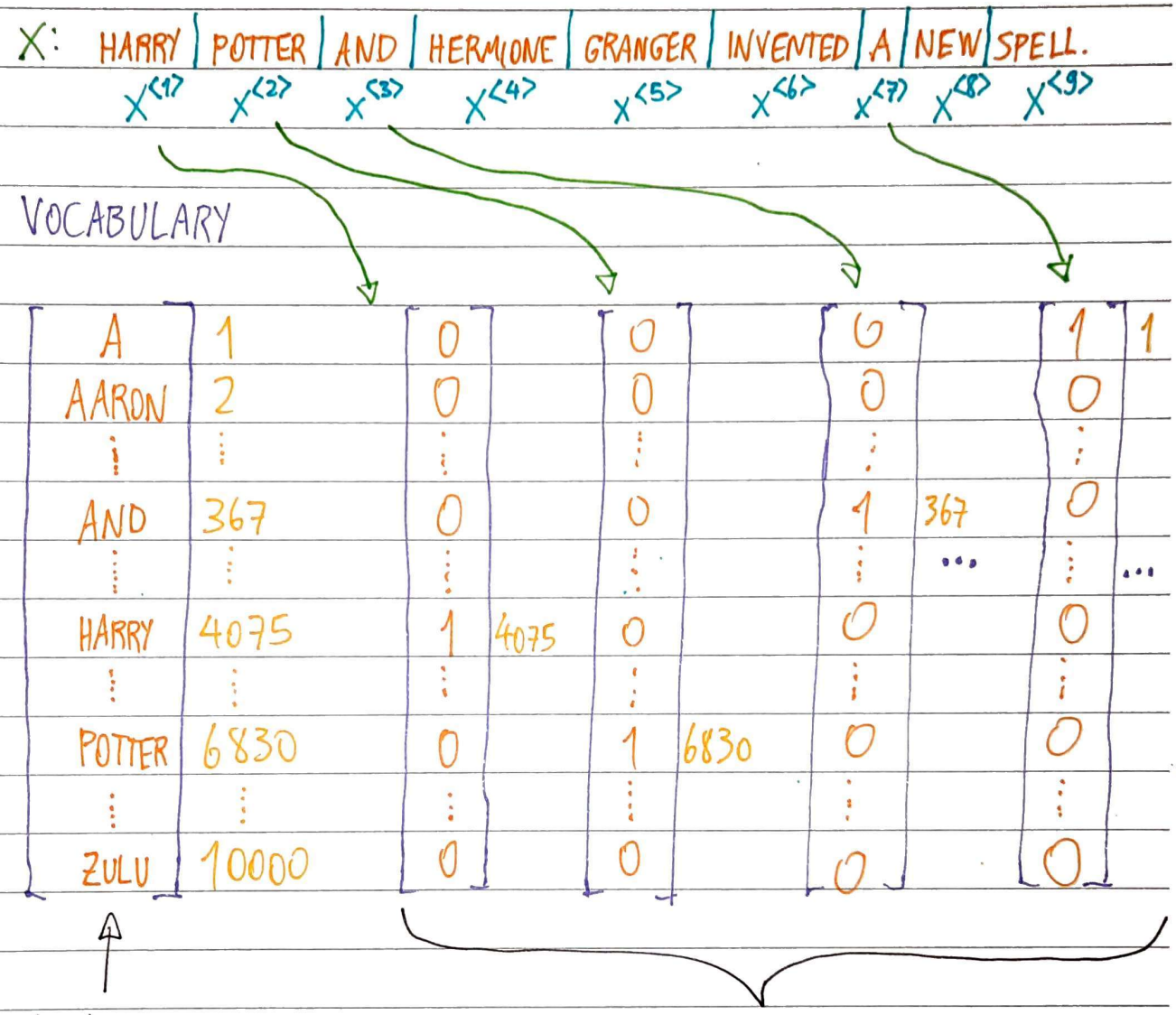
y: 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0

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

$T_y = 9$

$x^{(i)<t>}$ - t-th ELEMENT FROM i-th EXAMPLE
$T_x^{(i)}$ - SIZE (NO. OF ELEMENTS)
$y^{(i)<t>}$ - t-th ELEMENT FROM i-th OUTPUT
$T_y^{(i)}$ - SIZE (NO. OF ELEMENTS)

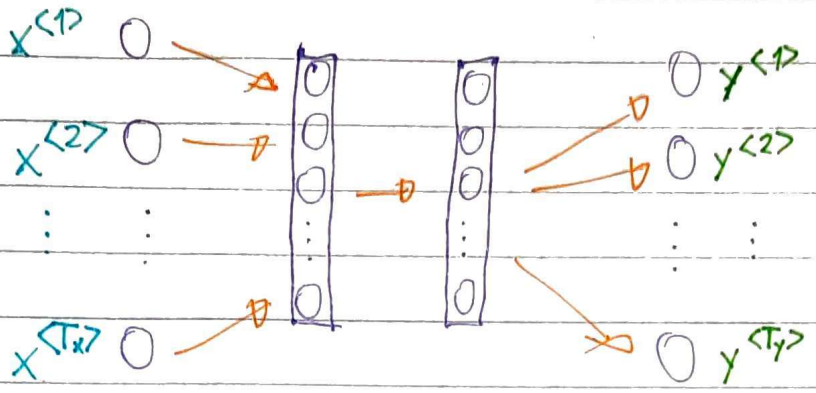
* VOCABULARY (DICTIONARY)



* WHAT IF A WORD IS NOT FOUND ON THE VOCABULARY?

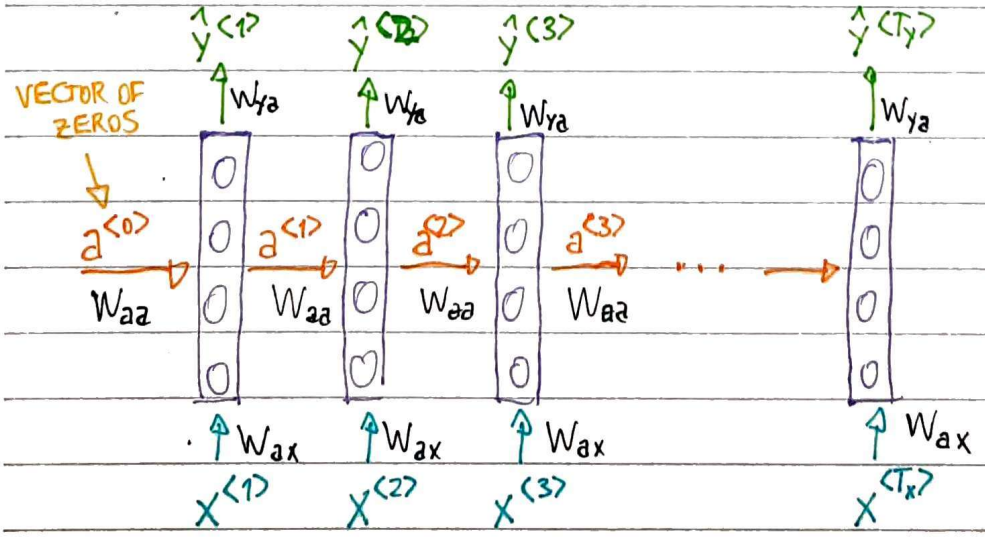
- CREATE NEW TOKEN FOR UNKNOWN WORDS. IF A WORD ISN'T ON DICTIONARY, RETURN THIS TOKEN.

→ WHY NOT A STANDARD NETWORK?



* PROBLEMS:
  - INPUTS AND OUTPUTS CAN HAVE DIFFERENT LENGTHS IN DIFFERENT EXAMPLES.
  - DOESN'T SHARE FEATURES LEARNED ACROSS DIFFERENT POSITIONS OF TEXT.

→ RECURRENT NEURAL NETWORKS (RNN)



* DISADVANTAGE: RNN's ONLY USE EARLIER INFORMATION
* TO USE LATER INFORMATION: BIDIRECTIONAL RNN (BRNN)

→ FORWARD PROPAGATION ON RNN'S

$$a^{\langle 0 \rangle} = \vec{0}$$

$$a^{\langle t \rangle} = g_1\left(W_{aa} \cdot a^{\langle t-1 \rangle} + W_{ax} \cdot x^{\langle t \rangle} + b_a\right) \leftarrow g_1: \text{TANH/ReLU}$$

$$\hat{y}^{\langle t \rangle} = g_2\left(W_{ya} \cdot a^{\langle t \rangle} + b_y\right) \leftarrow g_2: \text{SOFTMAX / SIGMOID / ETC.}$$

* SIMPLIFYING NOTATION:

$$W_a = \left[W_{aa} \vdots W_{ax}\right]$$

$$\left[a^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] = \begin{bmatrix} a^{\langle t-1 \rangle} \\ \cdots\cdots \\ x^{\langle t \rangle} \end{bmatrix} \rightarrow a^{\langle t \rangle} = g_1\left(W_a\left[a^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_a\right)$$

$$W_{ya} = W_y \rightarrow \hat{y}^{\langle t \rangle} = g_2\left(W_y \cdot a^{\langle t \rangle} + b_y\right)$$

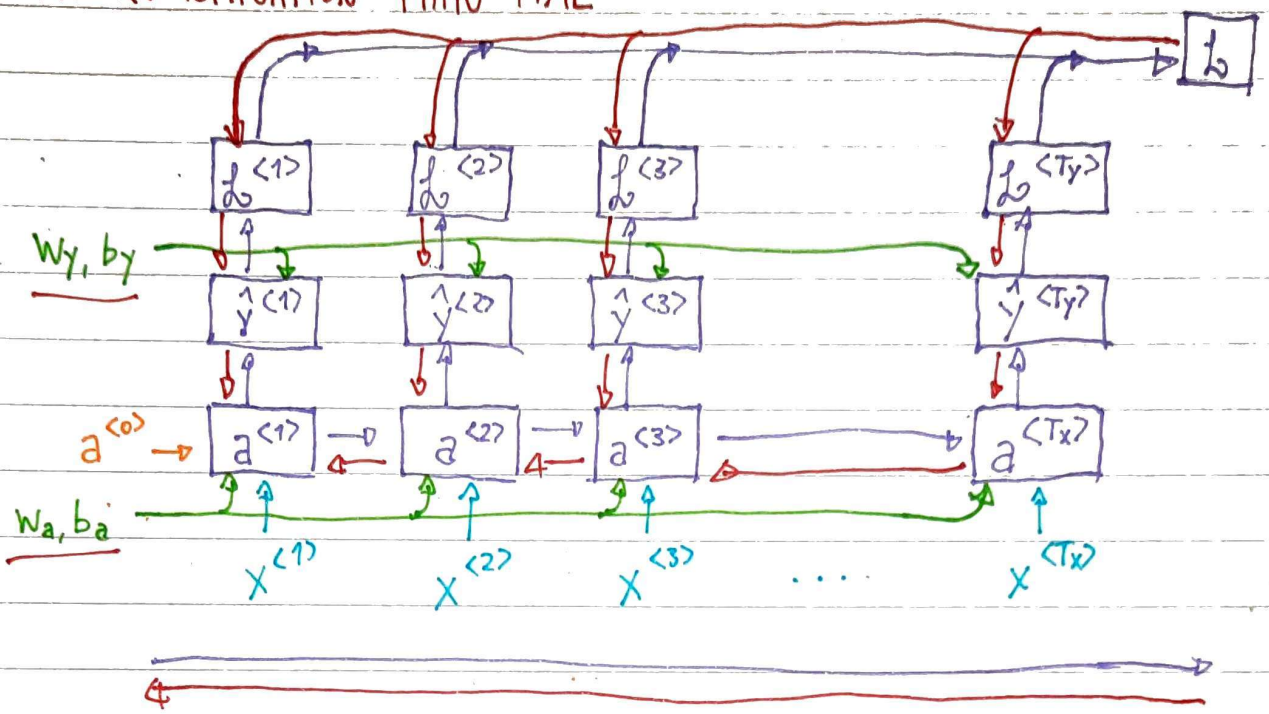
* FOR W_a:

- STACK MATRIXES HORIZONTALLY. THEY HAVE THE SAME NUMBER OF ROWS, BUT NUMBER OF COLUMNS CAN BE DIFFERENT.

* FOR $\left[a^{\langle t-1 \rangle}, x^{\langle t \rangle}\right]$:

- STACK MATRIXES VERTICALLY.

BACKPROPAGATION THRU TIME



* LOSS FUNCTION: CROSS-ENTROPY

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

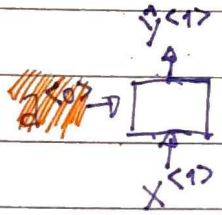
$$\mathcal{L}(\hat{y}, y) = \sum_{t=1}^{T_y} \mathcal{L}^{\langle t\rangle}(\hat{y}^{\langle t\rangle}, y^{\langle t\rangle})$$

- BACKPROPAGATE OVERALL LOSS TO PER-STEP LOSSES AND TO THE REST OF THE CHAIN

- MOST SIGNIFICANT RECURSIVE CALCULATION: ACTIVATIONS. BACKPROP OCCURS "FROM RIGHT TO LEFT" (LATER INFORMATION TO EARLIER INFORMATION)

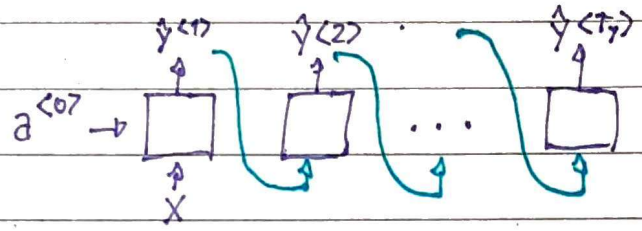
→ EXAMPLES OF RNN ARCHITECTURES

* ONE-TO-ONE



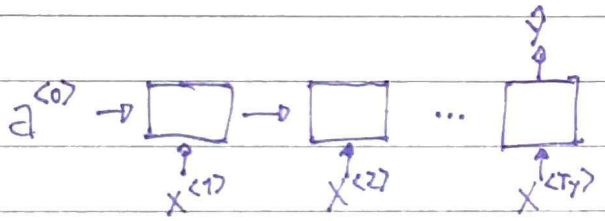
JUST A STANDARD N.N.
NOTHING SPECIAL.

* ONE-TO-MANY



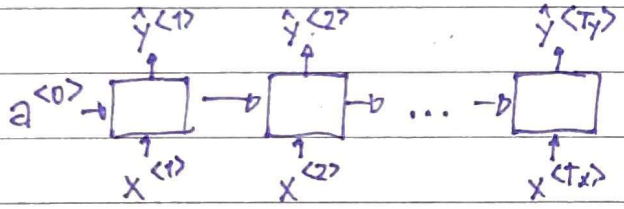
ONE INPUT, MANY OUTPUTS.

* MANY-TO-ONE



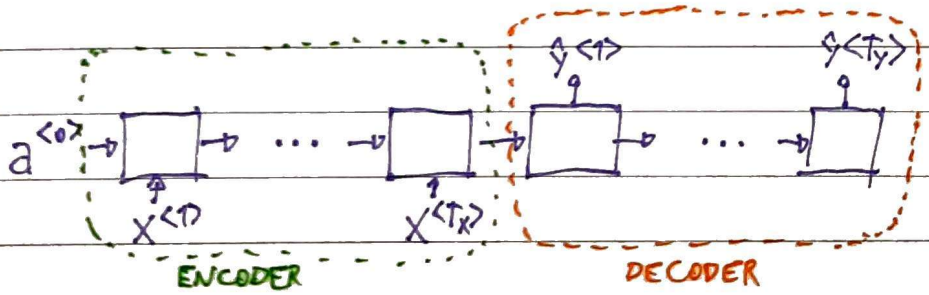
MANY INPUTS, ONE OUTPUT.

* MANY-TO-MANY ($T_x = T_y$)



INPUT AND OUTPUT HAVE THE SAME LENGTH.

* MANY-TO-MANY ($T_x \neq T_y$) (ENCODER/DECODER)



ENCODER/DECODER STRUCTURE. INPUT AND OUTPUT HAVE DIFFERENT LENGTHS.

→ LANGUAGE MODELLING

* CALCULATE PROBABILITY OF A SENTENCE TO AVOID ERRORS/AMBIGUITY
* EXAMPLE: SPEECH RECOGNITION
  - THE APPLE AND PAIR SALAD.
  - THE APPLE AND PEAR SALAD.

$P(\text{THE APPLE AND PAIR SALAD}) = 3.2 \times 10^{-13}$

$P(\text{THE APPLE AND PEAR SALAD}) = 5.7 \times 10^{-10}$

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

→ LANGUAGE MODELLING WITH AN RNN

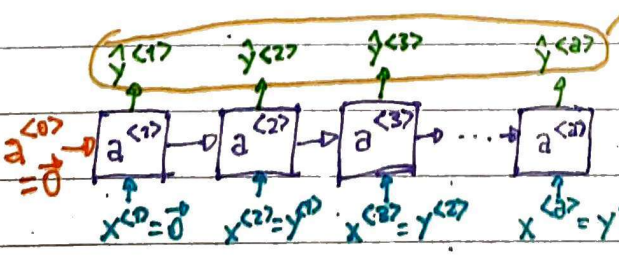
* TRAINING SET: LARGE CORPUS OF ENGLISH TEXT
* TOKENIZE WORDS BASED ON A VOCABULARY

THE EGYPTIAN <UNK> IS A BREAD OF CAT. <EOS>

$y^{<1>}$ $y^{<2>}$ $y^{<3>}$ $y^{<4>}$ $y^{<5>}$ $y^{<6>}$ $y^{<7>}$ $y^{<8>}$ $y^{<9>}$

- <EOS>: END OF SENTENCE TOKEN
- <UNK>: UNKNOWN WORD TOKEN

* MODEL



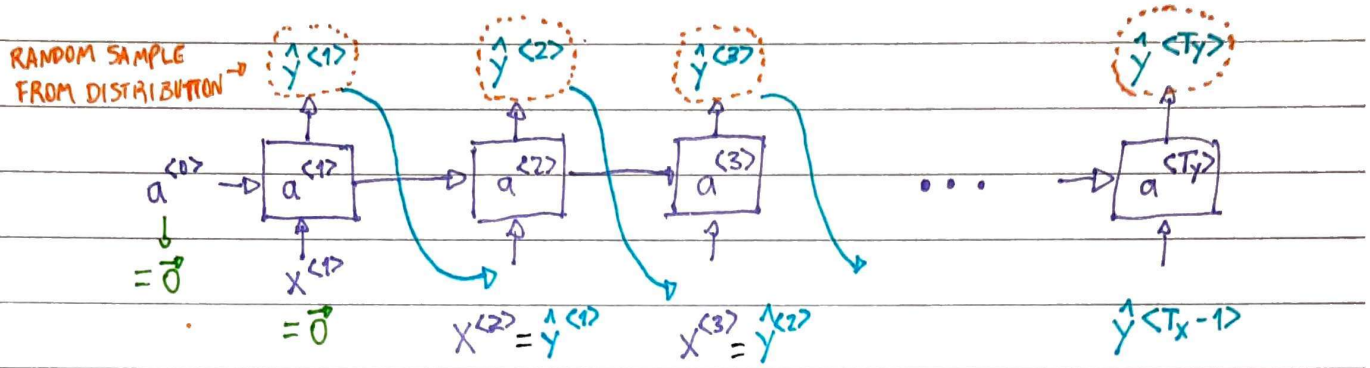
→ PREDICT NEXT WORD GIVEN PREVIOUS WORDS

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

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

→ SAMPLING A SEQUENCE FROM A TRAINED RNN

* TRAINED MODEL REPRESENTS A DISTRIBUTION OF PROBABILITY FOR SEQUENCES
* SAMPLING: TAKE RANDOM SAMPLES FROM THE PROBABILITY DISTRIBUTION
  - NUMPY: np.random.choice



* END OF PREDICTION WHEN <EOS> IS RETURNED
* IF <EOS> TOKEN ISN'T ON VOCABULARY, LIMIT Nº OF WORDS

* REJECTING <UNK> TOKENS OR NOT INCLUDING IT ON VOCABULARY

→ CHARACTER-LEVEL LANGUAGE MODEL

VOCABULARY = [a, b, c, ..., z, ␣, ., ,, ;, 0, 1, ..., 9, A, B, ..., Z]

* EVERY OUTPUT IS A CHARACTER
* LONGER SEQUENCIES
* MORE COMPUTATIONALLY EXPENSIVE TO TRAIN
* USE IN SPECIALIZED APPLICATIONS

→ SEQUENCE GENERATION

* DEPENDS ON THE DATA YOU USE TO TRAIN THE MODEL
* EXAMPLES: TRAIN A LANGUAGE MODEL WITH NEWS TEXTS OR SHAKESPEARE TEXTS

→ VANISHING GRADIENTS ON RNN'S

* EXAMPLE:

THE CAT, WHICH ALREADY ATE ..., WAS FULL.
THE CATS, WHICH ALREADY ATE ..., WERE FULL.

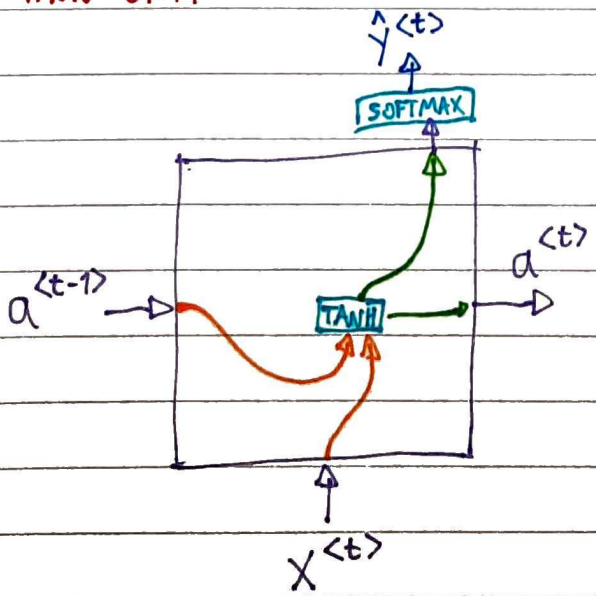
- RNN NEEDS PREVIOUS INFORMATION TO CHOOSE BETWEEN SINGULAR OR PLURAL NOUN

* RNN'S TEND NOT TO BE VERY GOOD AT CAPTURING LONG-RANGE DEPENDENCIES: VANISHING GRADIENTS ARE HARDER TO SOLVE

* EXPLODING GRADIENTS ARE EASIER TO DETECT AND SOLVE
  - NaN: NUMERICAL OVERFLOW
  - SOLUTION: GRADIENT CLIPPING

* PROPOSED SOLUTIONS
  - GATED RECURRENT UNIT (GRU)
  - LONG SHORT TERM MEMORY (LSTM)

→ RNN UNIT



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

g: TANH

GATED RECURRENT UNIT (GRU) SIMPLIFIED

* MEMORY CELL C $\Rightarrow$ $c^{\langle t \rangle} = a^{\langle t \rangle}$
* C CAN BE A VECTOR
* CANDIDATE TO REPLACE $c^{\langle t \rangle}$: $\tilde{c}^{\langle t \rangle} = \tanh(W_c[c^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_c)$

* GAMMA VALUE Γ_u BETWEEN 0 AND 1
  (Γ_u: UPDATE)

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

(σ: SIGMOID.)

* $c^{\langle t \rangle}$, $\tilde{c}^{\langle t \rangle}$ AND Γ_u HAVE THE SAME DIMENSION
* GAMMA IS A "GATE". IT DECIDES WHETHER OR NOT UPDATE THE MEMORY CELL

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

(* : ELEMENT-WISE)

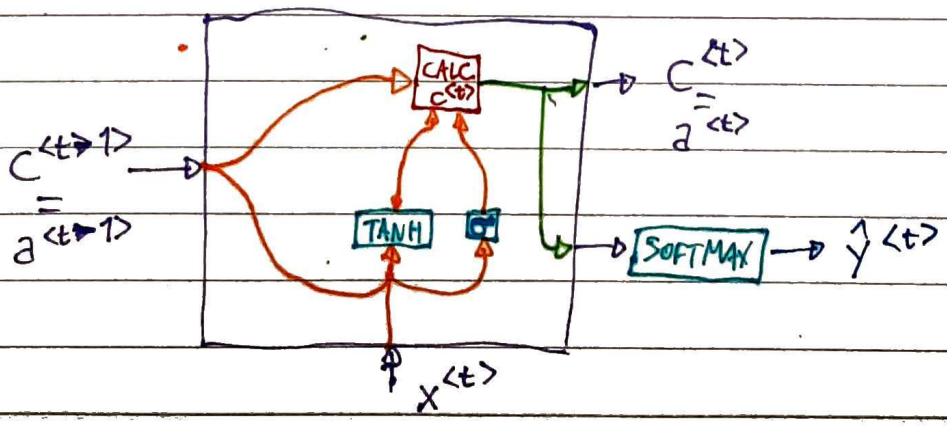
* EXAMPLE: C DETERMINES SINGULAR (1) OR PLURAL (0) NOUN

$\Gamma_u = 1$ $\quad \Gamma_u = 0$ $\quad \Gamma_u = 0$ $\quad \Gamma_u = 0, \ldots$ $\quad \Gamma_u = 1$

$c^{\langle t \rangle} = 1$

THE CAT, WHICH ALREADY ATE, WAS FULL

* SIGMOID KEEPS Γ_u CLOSE TO 0 OR CLOSE TO 1



FULL GATED RECURRENT UNIT (GRU)

* ADD RELEVANCE GATE Γ_r TO CALCULATE RELEVANCE OF $c^{\langle t-1 \rangle}$ IN THE CALCULUS OF $\tilde{c}^{\langle t \rangle}$

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

$$\Gamma_r = \sigma\left(W_r \left[c^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_r\right)$$

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

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

* WHY USING RELEVANCE GATE TOO?

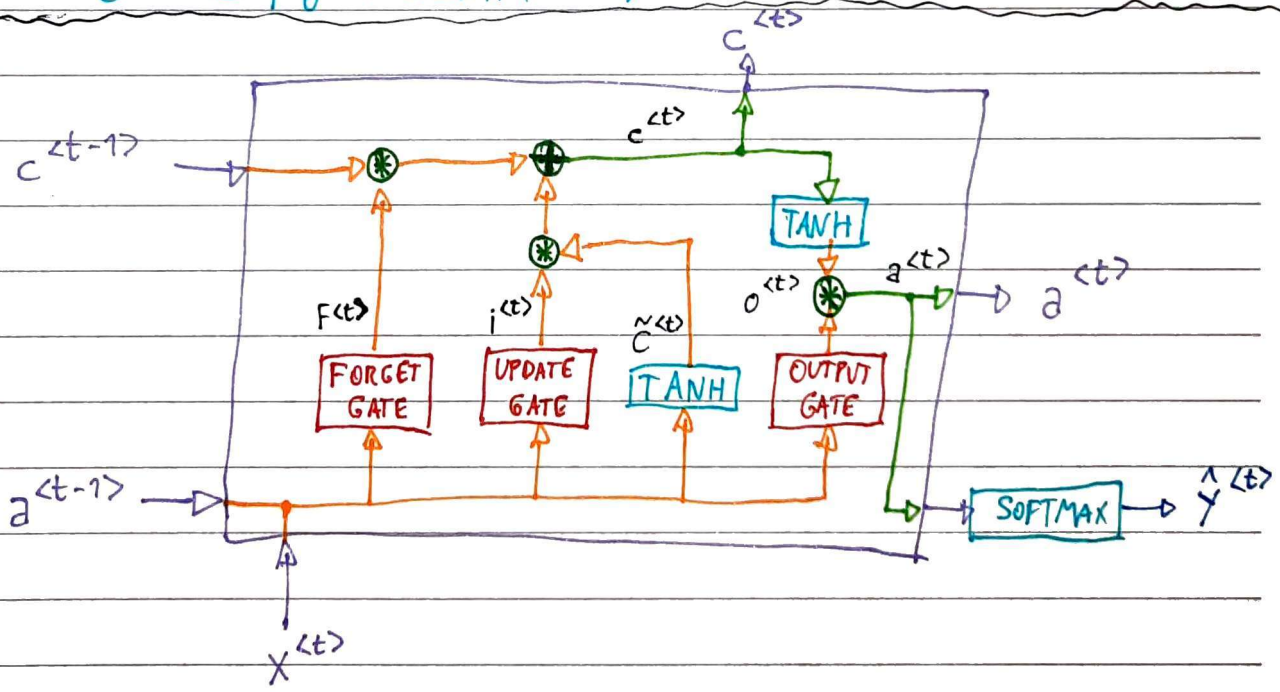
- MOST COMMONLY USED
- RELIABLE AND ROBUST

* MAYBE IN ACADEMIC LITERATURE, MEMORY CELLS, CANDIDATES AND GATES MAY HAVE A DIFFERENT NOTATION

→ LONG-SHORT-TERM MEMORY (LSTM)

* SIMILAR TO THE IDEA OF GRU's, BUT WITH THREE GAMMAS THAT WORK SLIGHTLY DIFFERENT
  - Γ_u (UPDATE), Γ_f (FORGET), Γ_o (OUTPUT)

$$\tilde{c}^{\langle t \rangle} = \tanh\left(W_c [a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_c\right)$$
$$\Gamma_u = \sigma\left(W_u [a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_u\right)$$
$$\Gamma_f = \sigma\left(W_f [a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_f\right)$$
$$\Gamma_o = \sigma\left(W_o [a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_o\right)$$
$$c^{\langle t \rangle} = \Gamma_u * \tilde{c}^{\langle t \rangle} + \Gamma_f * c^{\langle t-1 \rangle}$$
$$a^{\langle t \rangle} = \Gamma_o * \tanh(c^{\langle t \rangle})$$



* VARIATION: "PEEPHOLE CONNECTION"
  - Γ_o BEING DEPENDENT ON $c^{\langle t-1 \rangle}$ TOO
  - CAN BE APPLIED TO Γ_u AND Γ_f TOO

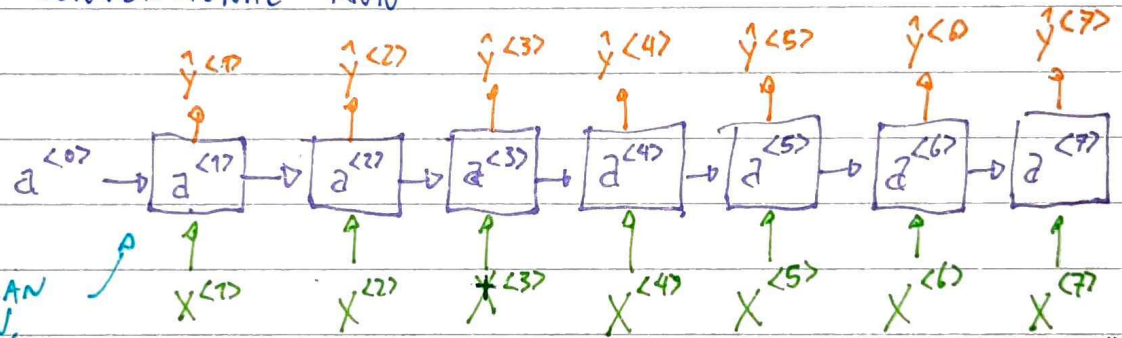
GETTING INFORMATION FROM THE FUTURE (BIDIRECTIONAL RNN)

* IMAGINE TWO PHRASES WITH DIFFERENT CONTEXTS:
  - HE SAID, "TEDDY | BEARS ARE ON SALE"!
  - HE SAID, "TEDDY | ROOSEVELT WAS A GREAT PRESIDENT!"

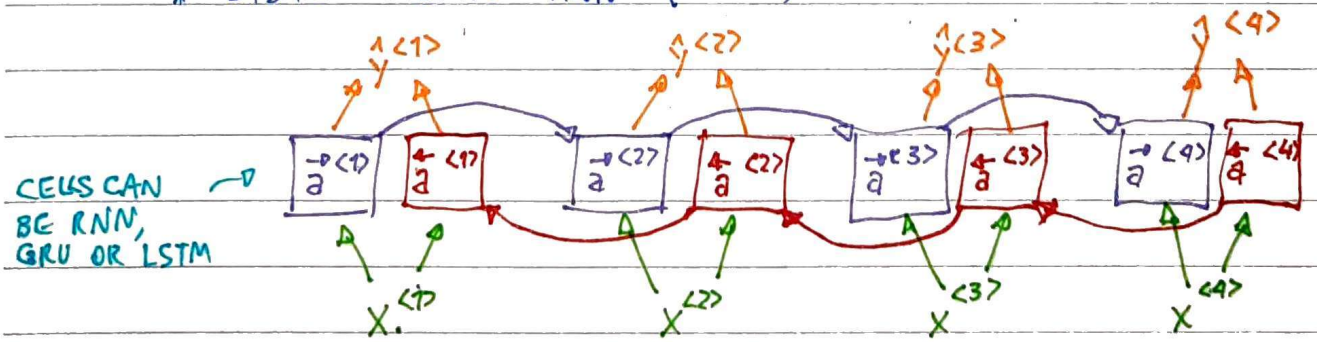
HOW TO PREDICT THE NEXT WORD WITH BETTER CORRELATION WITH CONTEXT?

GATHERING INFORMATION FROM FUTURE!

* CONVENTIONAL RNN



* BIDIRECTIONAL RNN (BRNN)



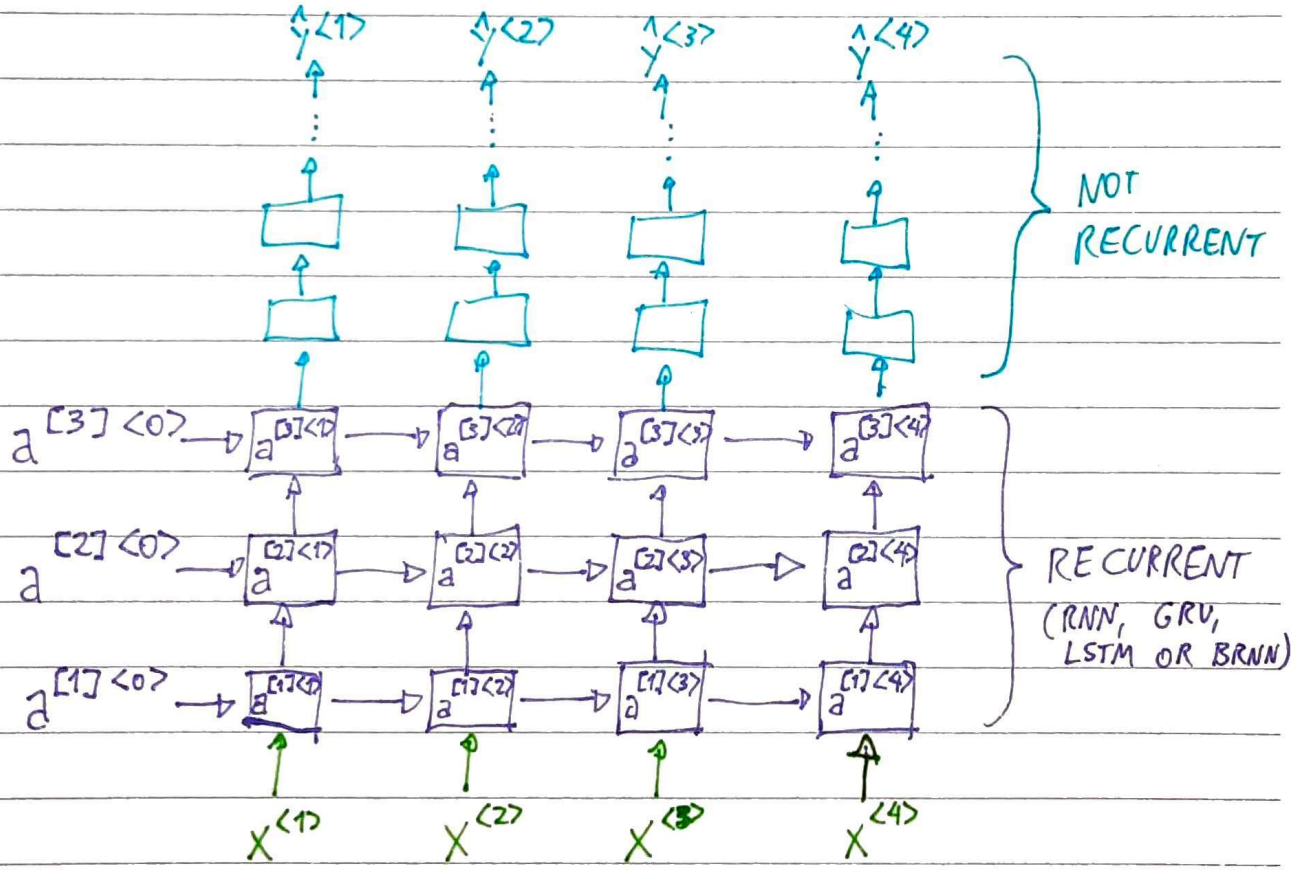
- ACYCLIC GRAPH

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

D-5-14

→ DEEP RECURRENT NEURAL NETWORKS (DEEP RNN'S)

* STACKING RNN, GRU OR LSTM CELLS IN SEQUENCE, FOLLOWED BY OTHER KINDS OF LAYERS (NOT NECESSARELY RECURRENT)
* RECURRENT LAYERS ARE COMPUTATIONALLY EXPENSIVE TO TRAIN, SO A FEW RECURRENT LAYERS ARE TIPICALLY USED ON A DEEP NETWORK



* NOTATION: $a^{[l]\langle t \rangle}$

ACTIVATION — LAYER — TIMESTAMP

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

→ WORD REPRESENTATION

* ONE-HOT VOCABULARY REPRESENTATION

- WEAKNESS: DOESN'T ALLOW SEMANTIC RELATIONSHIP BETWEEN WORDS

I WANT A GLASS OF ORANGE JUICE
I WANT A GLASS OF APPLE ?????

APPLE AND ORANGE HAVE A CORRELATION (BOTH ARE FRUITS), BUT ONE-HOT ENCODING DOESN'T ALLOW THE MODEL TO LEARN THIS SEMANTIC RELATIONSHIP. IN OTHER WORDS, THE MODEL WON'T GENERALIZE TO PREDICT THAT THE NEXT WORD MIGHT BE "JUICE".

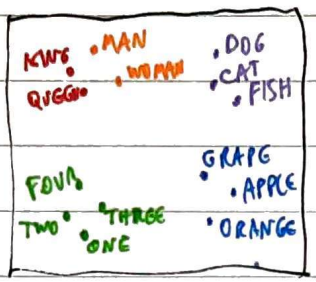
→ FEATURIZED WORD REPRESENTATION: WORD EMBEDDING

	(5391) MAN	(9853) WOMAN	(4914) KING	(7157) QUEEN	(456) APPLE	(6257) 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
FOOD	0,03	0,01	0,07	0,01	0,95	0,97
⋮	⋮	⋮	⋮	⋮	⋮	⋮
VERB	⋮	⋮	⋮	⋮	⋮	⋮
NOUN	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮
	e_{5391}	e_{9853}				

300 FEATURES

D-5-16

→ VISUALIZING WORD EMBEDDINGS

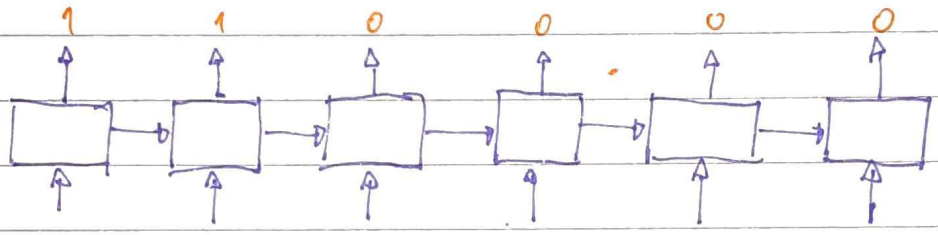


300 DIMENSIONS → 2-D EMBEDDING

↑

ALGORITHM t-SNE

→ NAMED ENTITY RECOGNITION EXAMPLE



Sally Johnson is an orange farmer. → AN ORANGE FARMER IS A PERSON

Robert Lin is an apple farmer. → APPLE/ORANGE ARE SIMILAR; ROBERT LIN IS A PERSON

Robert Lin is a durian cultivator. → WITH EMBEDDINGS, IT'S POSSIBLE TO IDENTIFY EVEN UNCOMMON WORDS.

- TRAINING WITH 1B ~ 100B WORDS
- NAME RECOGNITION DATASET: 100K WORDS (TRANSFER LEARNING)

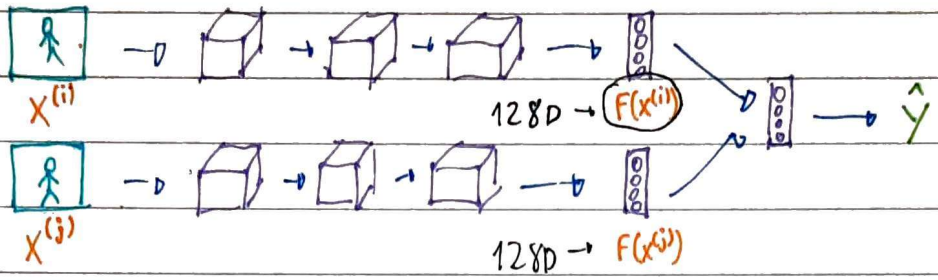
→ TRANSFER LEARNING AND WORD EMBEDDING

1. LEARN WORD EMBEDDINGS FROM LARGE TEXT CORPUS. (1~100 B WORDS)
   (OR DOWNLOAD PRE-TRAINED EMBEDDING ONLINE)
2. TRANSFER EMBEDDING TO NEW TASK WITH SMALLER TRAINING SET
   (SAY, 100K WORDS)
3. CONTINUE TO FINETUNE THE WORD EMBEDDINGS WITH NEW DATA.

OBS: NOT TOO USEFUL FOR LANGUAGE MODELING AND MACHINE TRANSLATION

D-5-17

→ RELATION TO FACE ENCODING (OR EMBEDDING)



* $F(x^{(i)})$ AND $F(x^{(j)})$ ARE ENCODINGS (OR EMBEDDINGS)
* DIFFERENCE: FACE RECOGNITION COMPUTES ENCODINGS FOR ANY IMAGE INPUT; NLP HAS A STRICT VOCABULARY

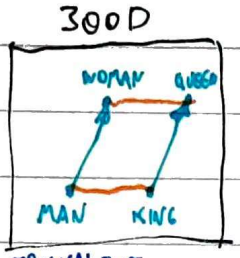
→ ANALOGIES

	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	0,70	0,69	0.03	-0.02
FOOD	0,09	0,01	0,02	0,01	0.95	0.97
	e_{MAN}	e_{WOMAN}	e_{KING}	e_{QUEEN}		

MAN → WOMAN AS KING → ??? (QUEEN)

$$e_{MAN} - e_{WOMAN} \approx \begin{bmatrix} -2 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \leftarrow \text{BOTH ARE CLOSE (ALMOST EQUAL)} \rightarrow \quad e_{KING} - e_{QUEEN} \approx \begin{bmatrix} -2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

→ ANALOGIES USING WORD VECTORS



EQUIVALENT VECTOR DISTANCES

ATTENTION: t-SNE "BREAKS" LINEARITY AND WE CAN'T RELY ON THESE "PARALLELOGRAM" DISTANCES ON A t-SNE REPRESENTATION.

$$e_{MAN} - e_{WOMAN} \approx e_{KING} - e_{W}$$

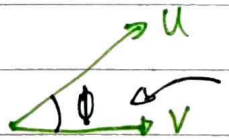
FIND WORD w: $\underset{w}{\text{argmax}} \left[\text{SIMILARITY}\left(e_w, e_{KING} - e_{MAN} + e_{WOMAN} \right) \right]$

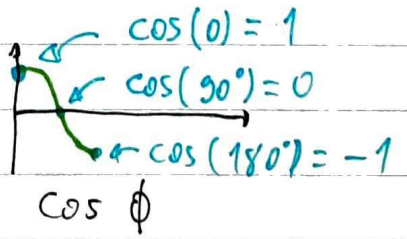
- APPROX. 30 ~ 75% OF ACCURACY

→ COSINE SIMILARITY

$$\text{SIMILARITY}\left(e_w, e_{KING} - e_{MAN} + e_{WOMAN} \right)$$

$$\text{SIM}(u, v) = \frac{u^T v}{\|u\|_2 \cdot \|v\|_2}$$ ← COSINE ANGLE



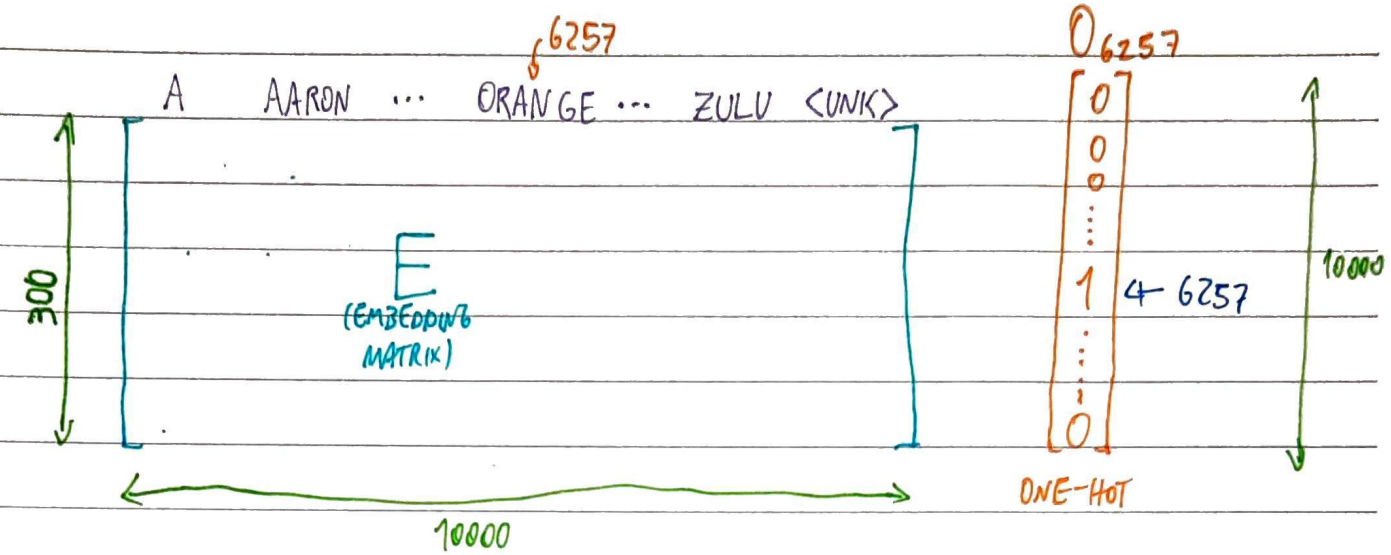


→ EXAMPLES OF ANALOGIES

* MAN : WOMAN AS BOY : GIRL
* OTTAWA : CANADA AS NAIROBI : KENYA
* BIG : BIGGER AS TALL : TALLER
* YEN : JAPAN AS RUBLE : RUSSIA

D-5-19

→ EMBEDDING MATRIX

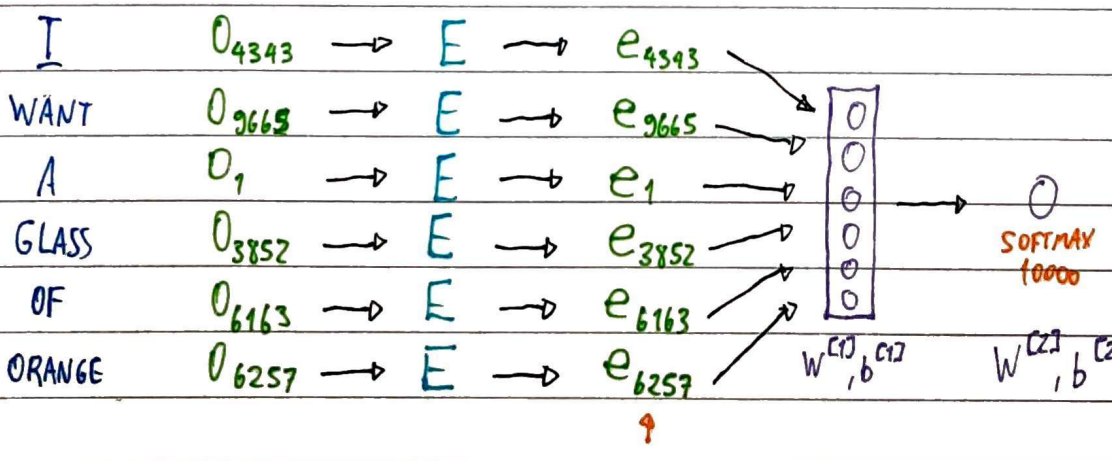


$$E_{(300,10000)} \cdot O_{6257\,(10000,1)} = e_{6257\,(300,1)}$$

EMBEDDING MATRIX — ONE-HOT ENCODING FOR "ORANGE" — EMBEDDING VECTOR FOR "ORANGE"

→ NEURAL LANGUAGE MODEL

I	WANT	A	GLASS	OF	ORANGE	JUICE
4343	9665	1	3852	6163	6257	



D-5-20

→ FIXED HISTORY ON NEURAL LANGUAGE MODELS

* INPUT SIZES ALWAYS FIXED
* USE ALWAYS THE LAST N WORDS TO PREDICT THE NEXT

→ 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 ALONG WITH
- → 4 WORDS ON LEFT & RIGHT
- → LAST 1 WORD → ORANGE ____
- → NEARBY 1 WORD → GLASS ... ____

→ SKIP-GRAMS (WORD2VEC)

* RANDOMLY PICK A WORD TO BE THE CONTEXT WORD
* RANDOMLY PICK ANOTHER WORD WITHIN SOME WINDOW (± N WORDS) TO BE THE TARGET WORD

I WANT A GLASS OF ORANGE JUICE TO GO ALONG WITH MY CEREAL.

CONTEXT	ORANGE	ORANGE	ORANGE
TARGET	JUICE	GLASS	MY

* LEARN GOOD WORD EMBEDDINGS

D-5-21

MODEL (VOCAB SIZE = 10000)

* EXAMPLE: LEARN MAPPING FROM CONTEXT WORD TO TARGET WORD

CONTEXT C ("ORANGE")$_{6257}$ → TARGET t ("JUICE")$_{4834}$

O_c → E → e_c → (SOFTMAX) → $\hat{y}$

O_c: ONE-HOT

E: EMBEDDING VECTOR

$e_c = E \cdot O_c$

SOFTMAX

$\hat{y}$: TARGET PREDICTION

SOFTMAX: $p(t|c) = \frac{e^{\theta_t^T e_c}}{\sum_{j=1}^{10000} e^{\theta_j^T e_c}}$ (θ_t: PARAMETER ASSOCIATED WITH TARGET t)

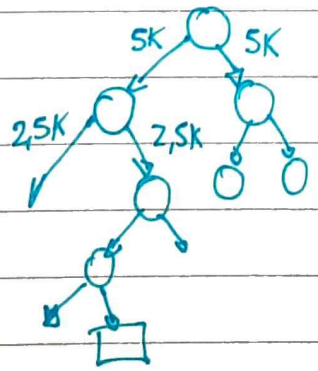
LOSS FOR SOFTMAX: $\mathcal{L}(\hat{y}, y) = -\sum_{i=1}^{10000} y_i \log \hat{y}_i$

$\hat{y}$: VECTOR WITH SOFTMAX PROBABILITIES

y: ONE-HOT VECTOR FOR TARGET WORD

PROBLEMS WITH SOFTMAX CLASSIFICATION

* COMPUTATIONALLY EXPENSIVE
* DENOMINATOR OF PROBABILITY WITH HUGE AMOUNT OF TERMS
* HIERARCHICAL SOFTMAX: (LOG COMPLEXITY RATHER THAN LINEAR)



IN PRACTICE, A BALANCED/PERFECTLY SYMMETRICAL TREE ISN'T USED; COMMON WORDS ARE POSITIONED ON TOP, AND LESS COMMON WORDS ARE PLACED DEEPER IN THE TREE.

* HOW TO SAMPLE CONTEXT C?

THE, OF, A, AND, TO... ← COMMON

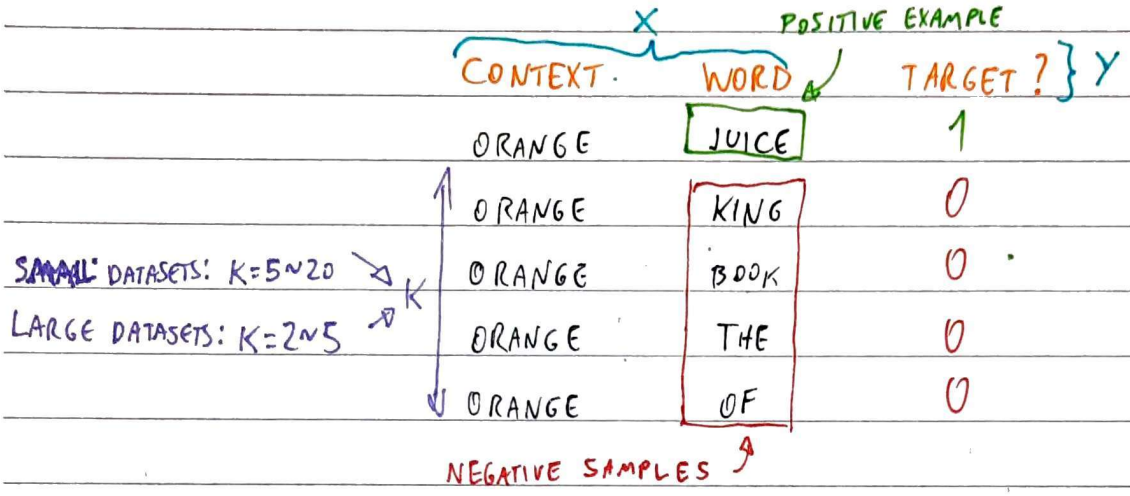
ORANGE, APPLE, DURIAN.. ← LESS COMMON

} DON'T TAKE UNIFORM SAMPLES; TECHNIQUES TO AVOID OVERCALCULUS OF COMMON WORDS

D-5-22

→ DEFINING A NEW LEARNING PROBLEM

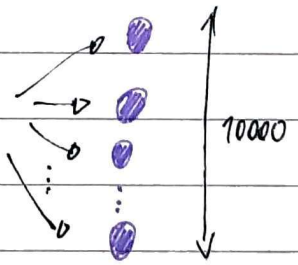
* CHOOSE A PAIR OF WORDS THAT HAVE A CONTEXT - TARGET RELATIONSHIP
* RANDOMLY SAMPLE OTHER TARGET EXAMPLES WITHOUT RELATIONSHIP WITH THE CONTEXT WORD



* LOGISTIC REGRESSION MODEL (LOWER COMPUTATIONAL COST)

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

$O_c \rightarrow E \rightarrow e_c$



INSTEAD OF TRAINING WITH 10000 EXAMPLES AT EVERY ITERATION, TRAIN WITH K+1 EXAMPLES AT EVERY ITERATION

→ SELECTING NEGATIVE EXAMPLES

* EMPIRICALLY TAKEN HEURISTIC VALUE

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

$f(w_i)$: OBSERVED FREQUENCY OF THE WORD w_i IN THE TRAINING SET TEXT CORPUS

D-5-23

→ GLOVE (GLOBAL VECTORS FOR WORD REPRESENTATION)

X_{ij} = # TIMES j APPEARS IN CONTEXT OF i

↳ $X_{ij} = X_{ji}$ — j: TARGET, i: CONTEXT

* MODEL:

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

$f(X_{ij})$: WEIGHTING TERM

$\theta_i^T e_j + b_i + b'_j$: "$\theta_t^T e_c$" ($t$, c)

* IF $\log X_{ij} = 0$: → $f(X_{ij}) = 0$ ("$0 \log 0$" = 0)
* WEIGHTING TERM HELPS TO BALANCE THE AMOUNT OF COMPUTATION FOR COMMON AND UNCOMMON WORDS
* IN GLOVE, θ_i AND e_j ARE SYMMETRIC → $e_w^{(FINAL)} = \frac{(e_w + \theta_w)}{2}$

→ NOTE ON THE FEATURIZATION VIEW OF WORD EMBEDDINGS

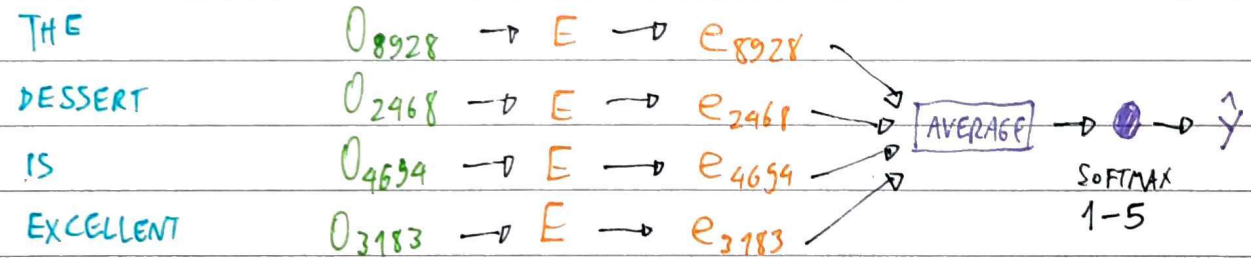
* IN PREVIOUS EXAMPLES, EMBEDDING FEATURES WERE REPRESENTED AS "GENDER", "ROYAL", "AGE", "FOOD", ETC.
* IN REAL APPLICATIONS, EMBEDDINGS MAY NOT BE EASILY INTERPRETABLE
* EMBEDDINGS MAY BE COMBINATIONS OF FEATURES
* IT'S VERY DIFFICULT TO LOOK AT INDIVIDUAL COMPONENTS AND ASSIGN HUMAN INTERPRETATION
* PARALLELOGRAM MAP STILL WORKS FOR REAL WORD EMBEDDINGS

→ SENTIMENT CLASSIFICATION PROBLEM

X	Y
THE DESSERT IS EXCELLENT.	★★★★☆
SERVICE WAS QUITE SLOW.	★★☆☆☆
GOOD FOR A QUICK MEAL, BUT NOTHING SPECIAL.	★★★☆☆
COMPLETELY LACKING IN GOOD TASTE, SERVICE AND AMBIENCE.	★☆☆☆☆

→ SIMPLE SENTIMENT CLASSIFICATION MODEL

THE DESSERT IS EXCELENT
8928 2468 4694 3183



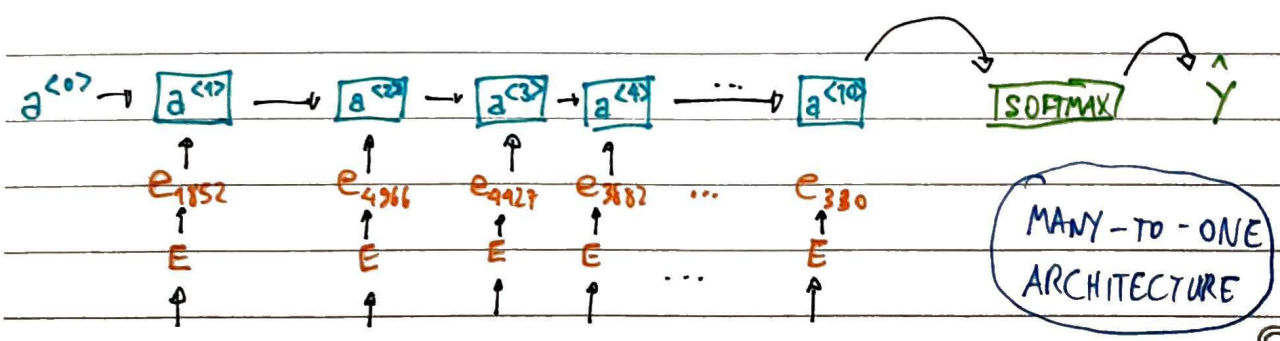
* AVERAGE MAY NOT WORK WELL IN SOME CASES.

COMPLETELY LACKING IN GOOD TASTE, GOOD SERVICE, AND GOOD AMBIENCE.

NEGATIVE REVIEW

THE WORD "GOOD" APPEARS THREE TIMES

→ RNN FOR SENTIMENT CLASSIFICATION



D-5-25

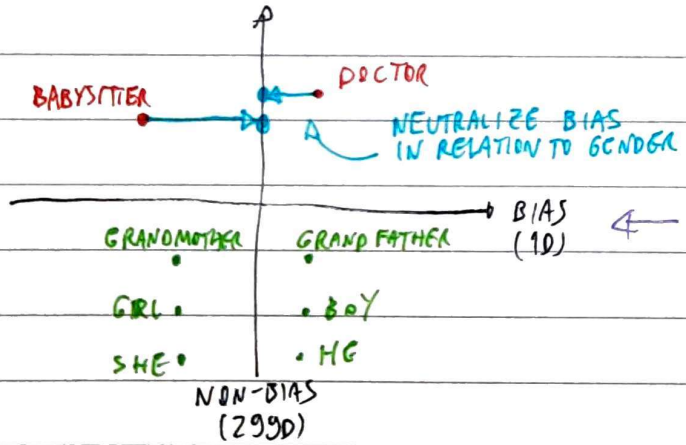
→ THE PROBLEM OF BIAS IN WORD EMBEDDINGS

* MAN: WOMAN AS KING: QUEEN ✓
* MAN: COMPUTER_PROGRAMMER AS WOMAN: HOMEMAKER ✗
* FATHER: DOCTOR AS MOTHER: NURSE ✗

WORD EMBEDDINGS CAN REFLECT GENDER, ETHNICITY, AGE, SEXUAL ORIENTATION, AND OTHER BIASES OF THE TEXT USED TO TRAIN THE MODEL.

→ ADDRESSING BIAS IN WORD EMBEDDINGS.

* EXAMPLE! BIAS IN GENDER DIRECTION



← NOT ALWAYS BIAS IS IN ONLY ONE DIMENSION. THIS IS AN EXAMPLE.

* STEPS FOR DEBIASING:

1. IDENTIFY BIAS DIRECTION. $e_{he} - e_{she}$ ... AVERAGE
   $e_{male} - e_{female}$
2. NEUTRALIZE: FOR EVERY WORD THAT IS NOT DEFINITIONAL, PROJECT TO GET RID OF BIAS.
3. EQUALIZE PAIRS. GRANDMOTHER - GRANDFATHER
   GIRL - BOY
   ⋮

D-5-26

→ BASIC SEQUENCE MODELS

* SEQUENCE-TO-SEQUENCE MODEL (EXAMPLE: MACHINE TRANSLATION)

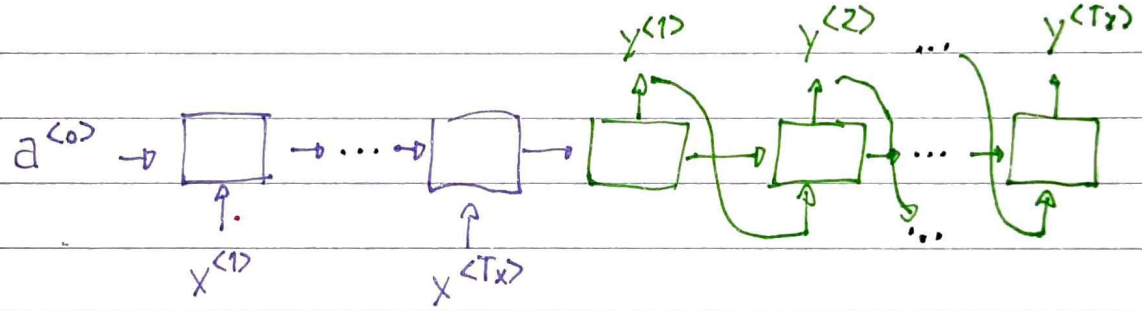
JANE VISITE L'AFRIQUE EN SEPTEMBRE (FRENCA)

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

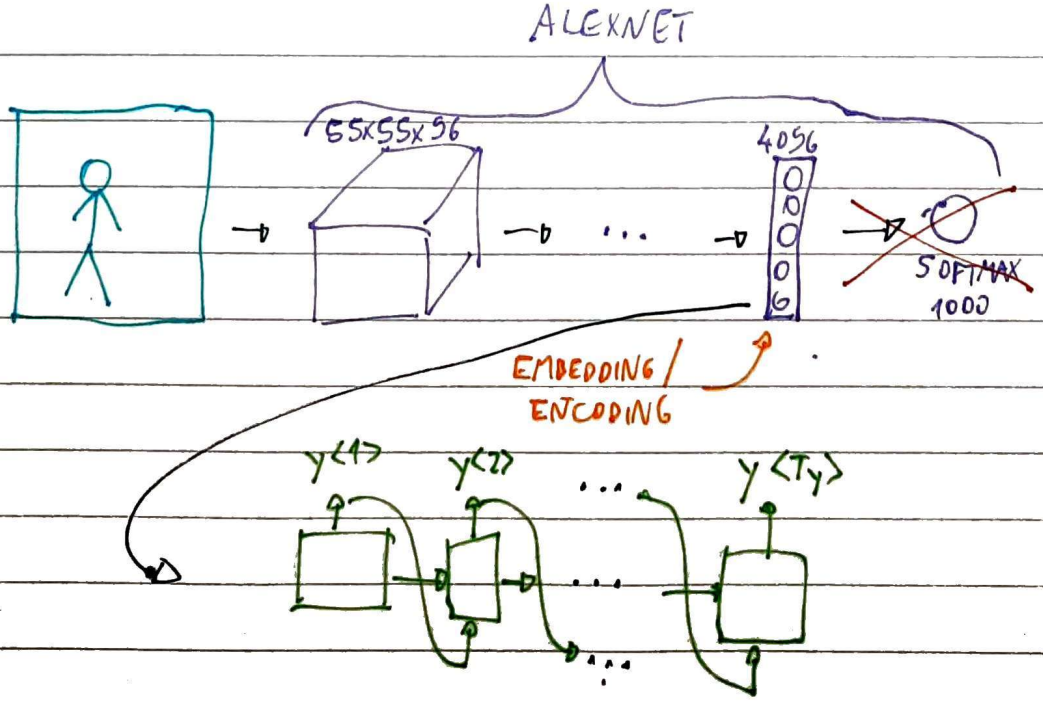
JANE IS VISITING AFRICA IN SEPTEMBER (ENGLISH)

$y^{<1>}$ $y^{<2>}$ $y^{<3>}$ $y^{<4>}$ $y^{<5>}$ $y^{<6>}$

ENCODER - DECODER ARCHITECTURE

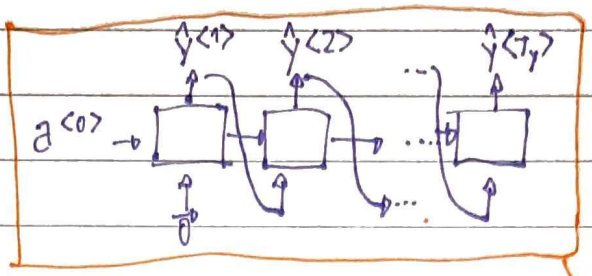


* IMAGE CAPTIONING (EXAMPLE WITH ALEXNET)



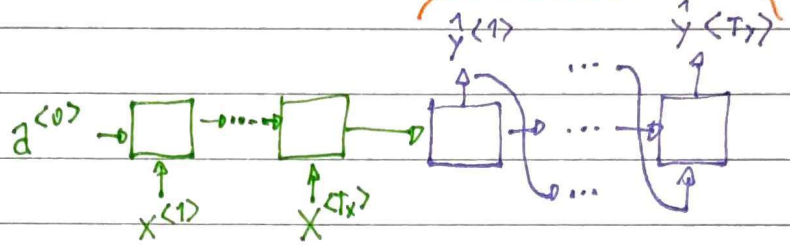
→ MACHINE TRANSLATION AS BUILDING A CONDITIONAL LANGUAGE MODEL

LANGUAGE MODEL:



$P(y^{<1>}, \ldots, y^{<T_y>})$

MACHINE TRANSLATION:



"CONDITIONAL LANGUAGE MODEL": $P(y^{<1>}, \ldots, y^{<T_y>} | x^{<1>}, \ldots, x^{<T_x>})$

→ FINDING THE MOST LIKELY TRANSLATION

JANE VISITE L'AFRIQUE EN SEPTEMBRE.

- JANE IS VISITING AFRICA IN SEPTEMBER. – BEST TRANSLATION
- JANE IS GOING TO BE VISITING AFRICA IN SEPTEMBER. – GOOD TRANSLATION, BUT MORE VERBOSE
- IN SEPTEMBER, JANE WILL VISIT AFRICA. – ACCEPTABLE TRANSLATION
- HER AFRICAN FRIEND WELCOMED JANE IN SEPTEMBER. – WRONG TRANSLATION

$$\underset{y^{<1>}, \ldots, y^{<T_y>}}{\text{argmax}} \; P(y^{<1>}, \ldots, y^{<T_y>} | x)$$

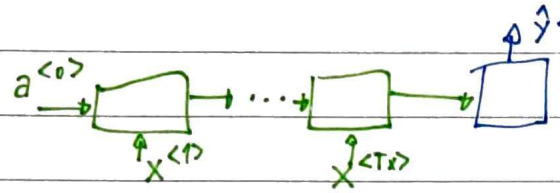
→ WHY NOT GREEDY SEARCH?

* BEST INDIVIDUAL WORD PROBABILITIES DOESN'T ALWAYS LEAD TO THE BEST TRANSLATIONS
* THEY COULD LEAD TO ACCEPTABLE TRANSLATIONS, BUT WITH MORE VERBOSE
* TOTAL POSSIBLE COMBINATIONS OF WORDS EXPONENTIALLY LARGER; COMPUTATIONALLY EXPENSIVE

→ BEAM SEARCH ALGORITHM

* EXAMPLE: TRANSLATE "JANE VISITE L'AFRIQUE EN SEPTEMBRE"

STEP 1: PICK B (BEAM WIDTH) POSSIBLE FIRST WORDS.

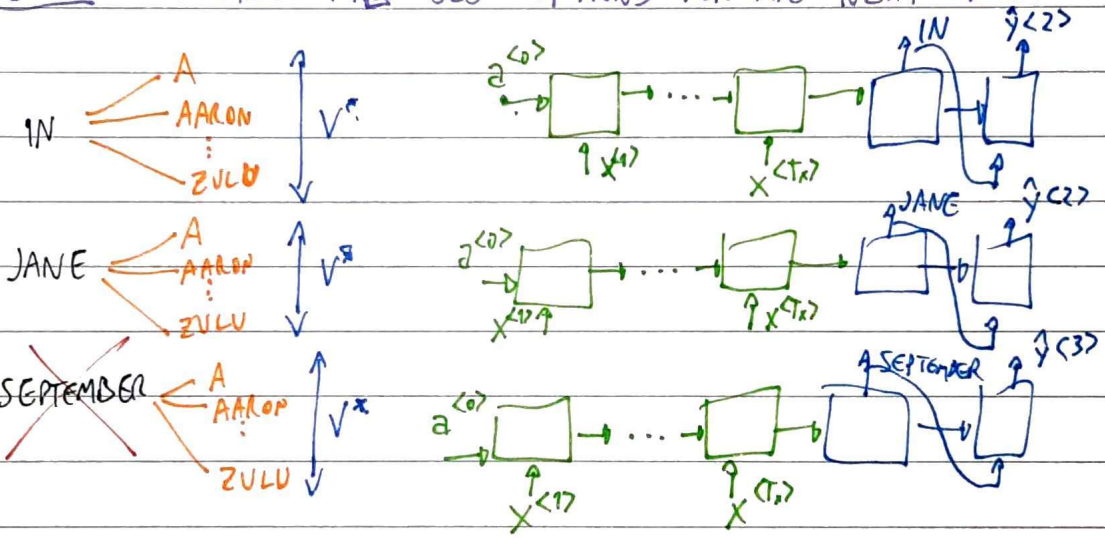


→ B=3: BEST 3 PROBABILITIES

IN / JANE / SEPTEMBER

$P(y^{<1>} | x)$

STEP 2: COMBINE THREE BEST OPTIONS FOR THE NEXT WORD



* V: VOCABULARY SIZE

BEST 3 PROBABILITIES: IN SEPTEMBER / JANE IS / JANE VISITS

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

STEP 3 AND NEXTS: REPEAT STEP 2 UNTIL <EOS>

IN SEPTEMBER JANE / JANE IS VISITING / JANE VISITS AFRICA

⋮ ⋮ ⋮

JANE VISITS AFRICA IN SEPTEMBER. <EOS>

* IF B=1: BEAM SEARCH = GREEDY SEARCH

→ LENGTH NORMALIZATION

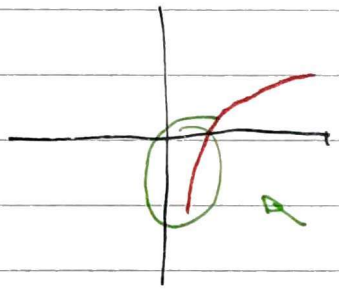
* MULTIPLY A LOT OF SMALL NUMBERS CAN CAUSE UNDERFLOW (NUMBERS SMALL ENOUGH TO BE STORED WITHOUT PRECISION)

$$\underset{y}{\text{argmax}} \prod_{t=1}^{T_y} P(y^{<t>} | x, y^{<1>}, \ldots, y^{<t-1>})$$

LOG:

CHANGES π TO Σ

$$\underset{y}{\text{argmax}} \sum_{t=1}^{T_y} \log P(y^{<t>} | x, y^{<1>}, \ldots, y^{<t-1>})$$



PROBABILITES ARE SMALL: SUM CAN RETRIEVE A LARGE NEGATIVE NUMBER

NORMALIZATION TERM

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

$\alpha = 0$: NO NORMALIZATION.

$\alpha = 1$: FULL NORMALIZATION.

→ BEAM SEARCH DISCUSSION

* LARGE B: BETTER RESULT, SLOWER
* SMALL B: WORSE RESULT, FASTER
* PRODUCTION: B ≈ 10
* ACADEMIC RESEARCH: B ≈ 1000 ~ 3000

UNLIKE EXACT SEARCH ALGORITHMS LIKE BREADTH/DEPTH FIRST SEARCH, BEAM SEARCH RUNS FASTER, BUT IS NOT GUARANTEED TO FIND EXACT MAXIMUM FOR $\underset{y}{\text{argmax}} P(y|x)$.

→ ERROR ANALYSIS ON BEAM SEARCH

* HOW TO FIGURE OUT IF RNN OR BEAM SEARCH IS THE MAJOR CAUSE OF ERRORS?

* EXAMPLE: "JANE VISITE L'AFRIQUE EN SEPTEMBRE"

- HUMAN: JANE VISITS AFRICA IN SEPTEMBER (y^*)
- ALGORITHM: JANE VISITED AFRICA LAST SEPTEMBER $(\hat{y})$

COMPUTE $P(y^*|x)$ AND $P(\hat{y}|x)$ WITH RNN

CASE 1: $P(y^*|x) > P(\hat{y}|x)$

BEAM SEARCH CHOSE $\hat{y}$. BUT $P(y^*|x)$ ATTAINS HIGHER THAN $P(\hat{y}|x)$.

CONCLUSION: BEAM SEARCH IS AT FAULT.

CASE 2: $P(y^*|x) \leq P(\hat{y}|x)$

y^* IS A BETTER TRANSLATION THAN $\hat{y}$. BUT RNN PREDICTED $P(y^*|x) \leq P(\hat{y}|x)$.

CONCLUSION: RNN MODEL IS AT FAULT.

* ERROR ANALYSIS PROCESS

HUMAN	ALGORITHM	$P(y^*\|x)$	$P(\hat{y}\|x)$	AT FAULT?
ENGLISH 1-H	ENGLISH 1-A	2×10^{-10}	1×10^{-10}	BEAM
ENGLISH 2-H	ENGLISH 2-A	2×10^{-10}	6×10^{-10}	RNN
ENGLISH 3-H	ENGLISH 3-A	3×10^{-9}	2×10^{-11}	BEAM
⋮	⋮	⋮	⋮	⋮

FIGURES OUT WHAT FRACTION OF ERRORS ARE "DUE TO" BEAM SEARCH VS. RNN MODEL.

→ BLEU SCORE (BILINGUAL EVALUATION UNDERSTUDY)

* EVALUATING MACHINE TRANSLATION

FRENCH: LE CHAT EST SUR LE TAPIS.

REFERENCE 1: THE CAT IS ON THE MAT.

REFERENCE 2: THERE IS A CAT ON THE MAT.

MACHINE TRANSLATION: THE THE THE THE THE THE THE

- FIRST PRECISION TENTATIVE: COUNT HOW MANY (TIMES) PREDICTED WORDS APPEAR IN THE REFERENCES.

→ PRECISION: $\frac{7}{7}$

INCORRECT!

MODIFIED PRECISION: "CREDIT" THE WORDS WITH THE MAXIMUM NUMBER OF TIMES THEY APPEAR IN THE REFERENCES.

→ PRECISION: $\frac{2}{7}$ ← $\text{COUNT}_{\text{CLIP}}$("THE")

MORE REASONABLE. ← COUNT ("THE")

→ BLEU SCORE ON BIGRAMS

* EXAMPLE: THE CAT THE CAT ON THE MAT.

	COUNT	$\text{COUNT}_{\text{CLIP}}$
THE CAT (2x)	2	1
CAT THE	1	0
CAT ON	1	1
ON THE	1	1
THE MAT	1	1

PRECISION = $\frac{4}{6}$

BLEU SCORE ON UNIGRAMS

* EXAMPLE:

REFERENCE 1: THE CAT IS ON THE MAT.

REFERENCE 2: THERE IS A CAT ON THE MAT.

MACHINE TRANSLATION: THE CAT THE CAT ON THE MAT. ($\hat{y}$)

$$P_1 = \frac{\sum_{\text{UNIGRAMS} \in \hat{y}} \text{COUNT}_{\text{CLIP}}(\text{UNIGRAM})}{\sum_{\text{UNIGRAMS} \in \hat{y}} \text{COUNT}(\text{UNIGRAM})}$$

UNIGRAM

BLEU SCORE ON N-GRAMS

$$P_N = \frac{\sum_{\text{N-GRAMS} \in \hat{y}} \text{COUNT}_{\text{CLIP}}(\text{N-GRAM})}{\sum_{\text{N-GRAMS} \in \hat{y}} \text{COUNT}(\text{N-GRAM})}$$

* IF MT = REFERENCE: $P_1, P_2, \ldots, P_N = 1.0$

BLEU DETAILS

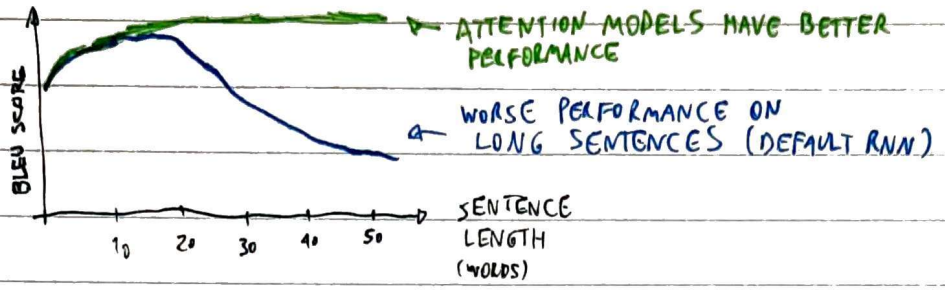
P_N – BLEU SCORE ON N-GRAMS ONLY

COMBINED BLEU SCORE: $BP \cdot \exp\left(\frac{1}{4}\sum_{n=1}^{4} P_n\right)$

BP – BREVITY PENALTY

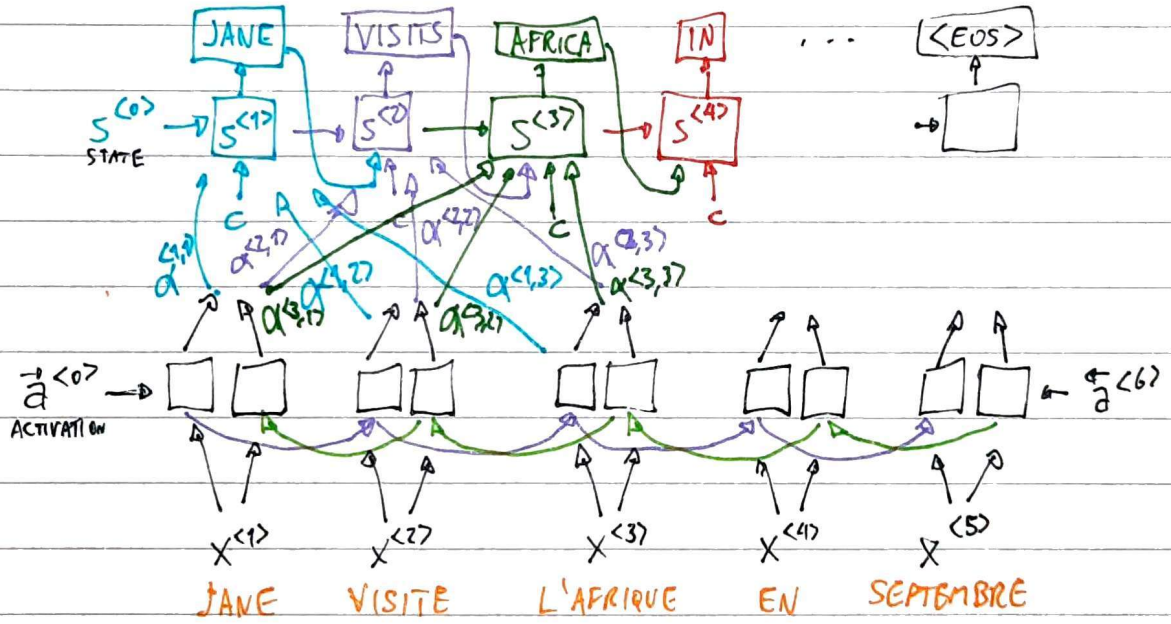
$$BP = \begin{cases} 1, & \text{IF MT\_OUTPUT\_LENGTH} > \text{REFERENCE\_OUTPUT\_LENGTH} \\ \exp(1 - \text{REFERENCE\_OUTPUT\_LENGTH}/\text{MT\_OUTPUT\_LENGTH}), & \text{OTHERWISE} \end{cases}$$

→ THE PROBLEM OF LONG SEQUENCES



ATTENTION MODELS TRY TO TRANSLATE PART BY PART: BETTER PERFORMANCE

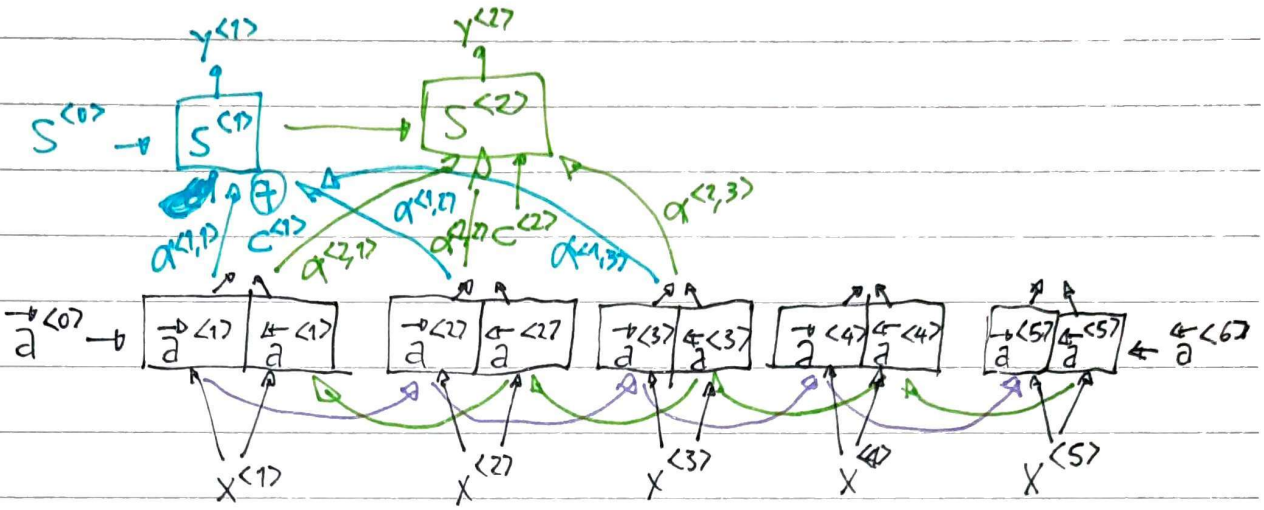
→ ATTENTION MODEL INTUITION



* α: ATTENTION WEIGHT.
* $\alpha^{<t,w>}$: ATTENTION WEIGHT FROM WORD w (FRENCH) TO GENERATE t (ENGLISH).

→ ATTENTION MODEL

* CONSIDERING $\vec{a}^{\langle t \rangle} = (\vec{a}^{\langle t \rangle}, \overleftarrow{a}^{\langle t \rangle})$; t' FOR FRENCH SENTENCE TIMESTEP; t FOR ENGLISH SENTENCE TIMESTEP



$$\sum_{t'} \alpha^{\langle 1,t' \rangle} = 1$$

$$c^{\langle 1 \rangle} = \sum_{t'} \alpha^{\langle 1,t' \rangle} a^{\langle t' \rangle}$$

$$c^{\langle 2 \rangle} = \sum_{t'} \alpha^{\langle 2,t' \rangle} a^{\langle t' \rangle}$$

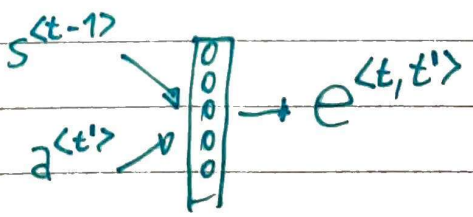
* $\alpha^{\langle t,t' \rangle}$: AMOUNT OF "ATTENTION" $y^{\langle t \rangle}$ SHOULD PAY TO $a^{\langle t' \rangle}$.

→ COMPUTING ATTENTION $\alpha^{\langle t,t' \rangle}$

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

$e^{\langle t,t' \rangle}$: TRAINED PARAMETER

SOFTMAX!



* QUADRATIC COMPLEXITY ORDER (BUT IT'S ACTUALLY ACCEPTABLE)
* EXAMPLE OF USE OF ATTENTION MODELS: DATE NORMALIZATION

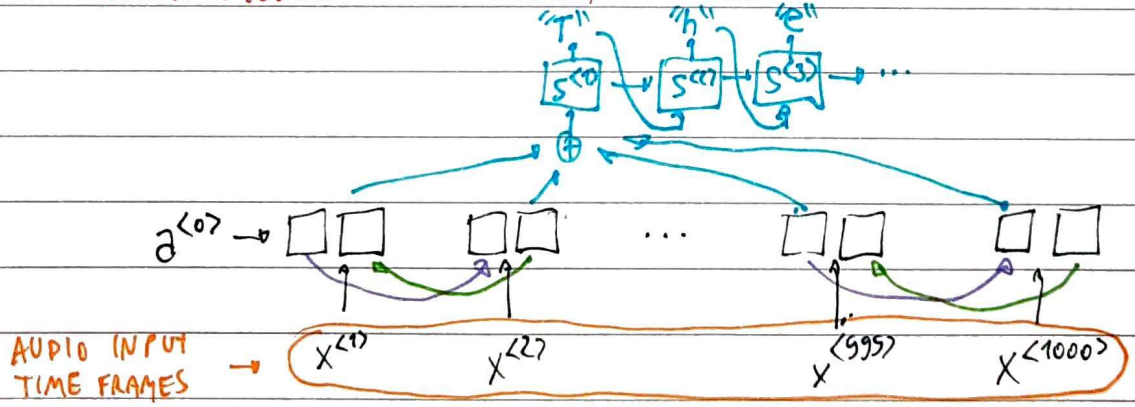
→ SPEECH RECOGNITION PROBLEM

X — AUDIO CLIP → Y — TRANSCRIPT

USE RAW SAMPLES OR SPECTROGRAM → "THE QUICK BROWN FOX"

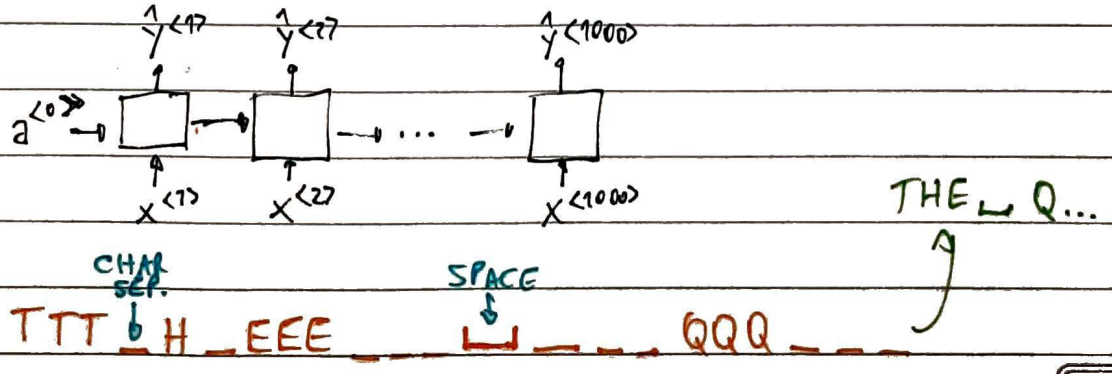
* PAST APPROACHES USED PHONEMES; WITH END-TO-END DEEP LEARNING, THIS ISN'T NECESSARY ANYMORE
* TRAINING ON LARGE DATASET: ~300 HOURS OF AUDIO
* PRODUCTION SYSTEMS USUALLY TRAIN WITH 10000 ~ 100000 HOURS

→ ATTENTION MODEL FOR SPEECH RECOGNITION



→ CTC (CONNECTIONIST TEMPORAL CLASSIFICATION) COST FOR SPEECH RECOGNITION

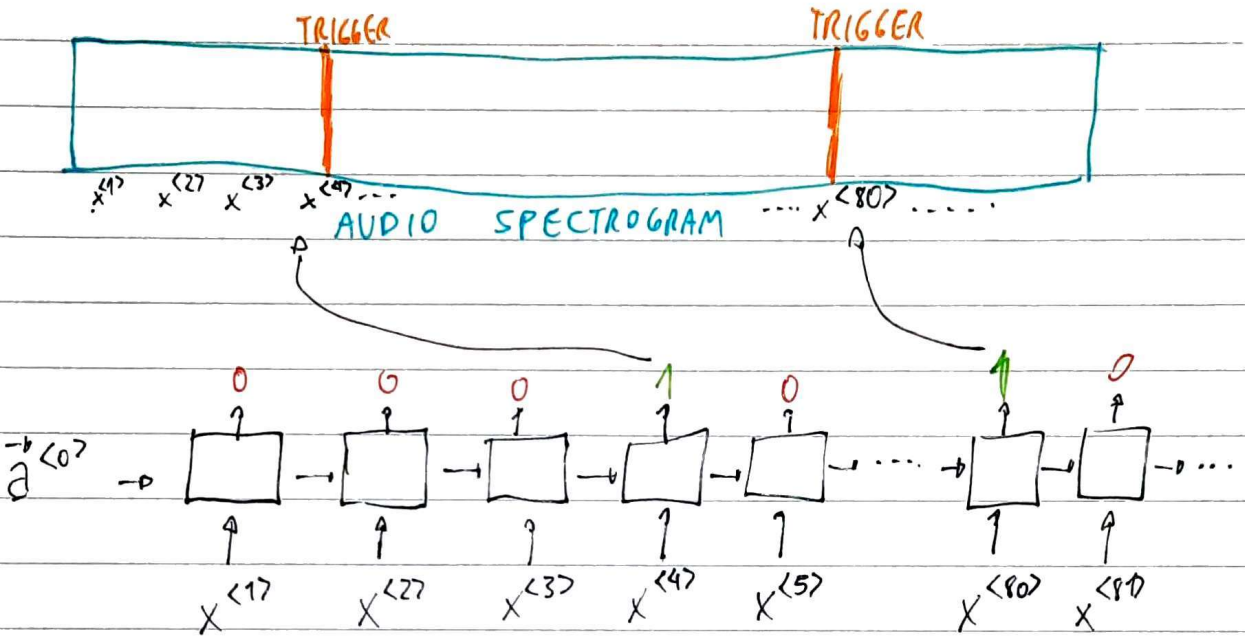
* EXAMPLE: "THE QUICK BROWN FOX" → (19 CHARACTERS)



→ TRIGGER WORD DETECTION

* EXAMPLE: AMAZON ECHO (ALEXA), GOOGLE HOME (OK GOOGLE)

* ALGORITHM:

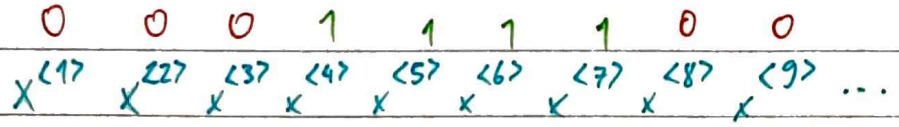


* PROBLEMS:

- UNBALANCED DATASET
- MORE 0'S THAN 1'S

* FIXING (HACKING) THE PROBLEM:

- EXTEND 1'S FOR A LITTLE BIT



TRANSFORMER NETWORK MOTIVATION

RNN → GRU → LSTM

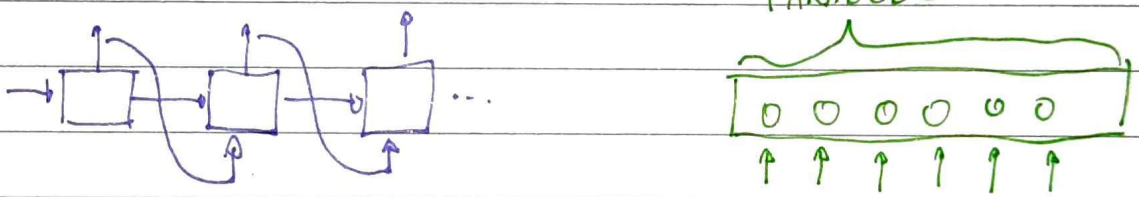
INCREASED COMPLEXITY

SEQUENTIAL (ONE TOKEN AT A TIME)

* HOW TO MAKE IT MORE EFFICIENT? IS THERE A WAY TO MAKE A PARALLEL INPUT PROCESSING?

TRANSFORMER NETWORK INTUITION

* COMBINE ATTENTION WITH CNN STYLE OF PROCESSING



RNN

CNN

* ATTENTION + CNN

SELF-ATTENTION: COMPUTE N REPRESENTATIONS FOR A SENTENCE WITH N WORDS IN PARALLEL

MULTI-HEAD ATTENTION: BASICALLY A "FOR LOOP" OVER SELF-ATTENTION. MULTIPLE VERSIONS OF REPRESENTATIONS

→ SELF-ATTENTION INTUITION

$A(q, K, V)$ = ATTENTION-BASED VECTOR REPRESENTATION OF A WORD

↳ CALCULATE FOR EACH WORD $A^{\langle 1 \rangle}, \dots, A^{\langle N \rangle}$

RNN ATTENTION:

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

TRANSFORMERS ATTENTION

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

$A^{\langle 1 \rangle}$	$A^{\langle 2 \rangle}$	$A^{\langle 3 \rangle}$	$A^{\langle 4 \rangle}$	$A^{\langle 5 \rangle}$
JANE	VISITE	L'AFRIQUE	EN	SEPTEMBRE.

→ CALCULATING $A(q, K, V)$

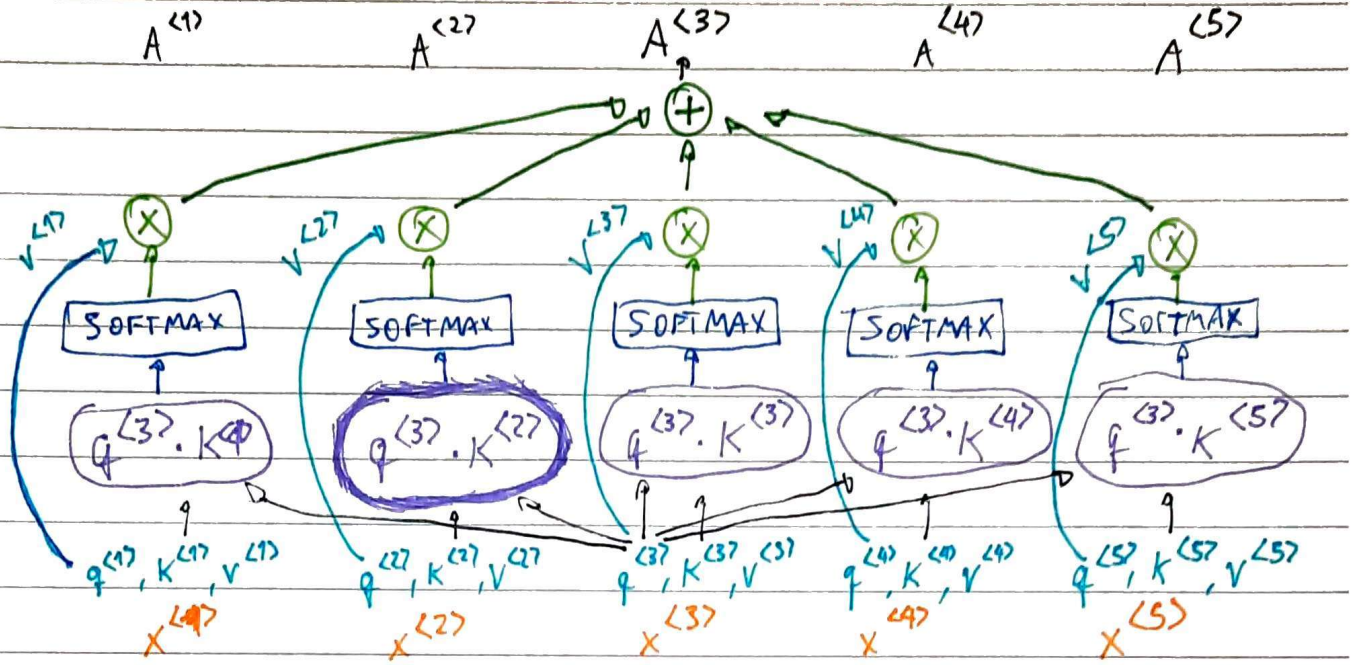
QUERY (q)	KEY (K)	VALUE (V)
$q^{\langle 1 \rangle}$	$K^{\langle 1 \rangle}$	$V^{\langle 1 \rangle}$
$q^{\langle 2 \rangle}$	$K^{\langle 2 \rangle}$	$V^{\langle 2 \rangle}$
$q^{\langle 3 \rangle}$	$K^{\langle 3 \rangle}$	$V^{\langle 3 \rangle}$
$q^{\langle 4 \rangle}$	$K^{\langle 4 \rangle}$	$V^{\langle 4 \rangle}$
$q^{\langle 5 \rangle}$	$K^{\langle 5 \rangle}$	$V^{\langle 5 \rangle}$

5 WORDS.

* ANALOGY WITH DATABASES: "QUERY" MIGHT BE A QUESTION; "KEY-VALUE" SHOULD BE LIKE "ANSWERS"
* q, K AND V CALCULATED WITH TRAINED PARAMETERS

$$q^{\langle N \rangle} = W^q \cdot x^{\langle N \rangle} \quad | \quad K^{\langle N \rangle} = W^K \cdot x^{\langle N \rangle} \quad | \quad V^{\langle N \rangle} = W^V \cdot x^{\langle N \rangle}$$

EXAMPLE OF SELF-ATTENTION CALCULUS



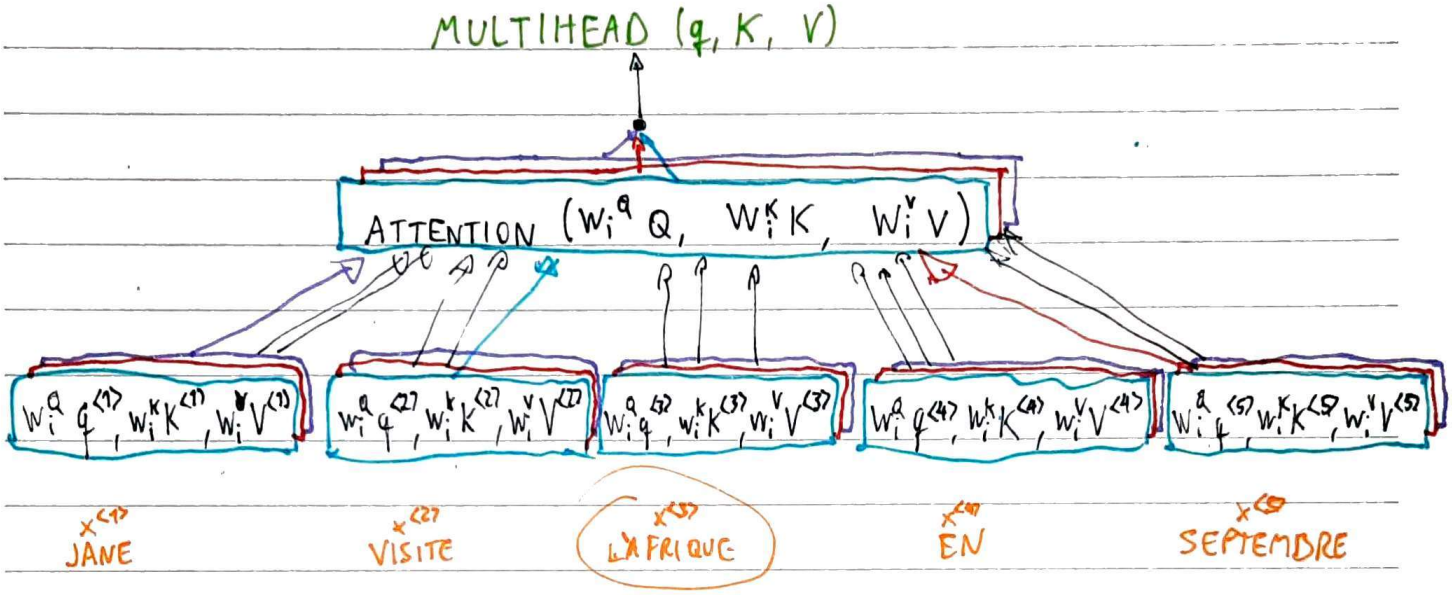
* CALCULATING $A^{\langle 3 \rangle}$:

* $q^{\langle 3 \rangle}$ MIGHT REPRESENT A QUESTION LIKE "WHATS HAPPENING THERE?"
* MAYBE $K^{\langle 2 \rangle}$ SHOULD BE THE BEST ANSWER.
  "THERE IS HAPPENING A VISIT".
  $q^{\langle 3 \rangle} \cdot K^{\langle 2 \rangle}$ SHOULD HAVE THE GREATEST CONTRIBUTION.

$$\text{ATTENTION}(q, K, V) = \text{SOFTMAX}\left(\frac{qK^T}{\sqrt{d_k}}\right)V$$

→ MULTI-HEAD ATTENTION

* "HEAD": EACH SELF-ATTENTION CALCULATED



* EXAMPLES OF WHAT EACH HEAD SHOULD REPRESENT?

HEAD 1: WHAT'S HAPPENING?
W_1^Q, W_1^K, W_1^V
BEST ANSWER: $K^{<2>}$

HEAD 2: WHEN?
W_2^Q, W_2^K, W_2^V
BEST ANSWER: $K^{<5>}$

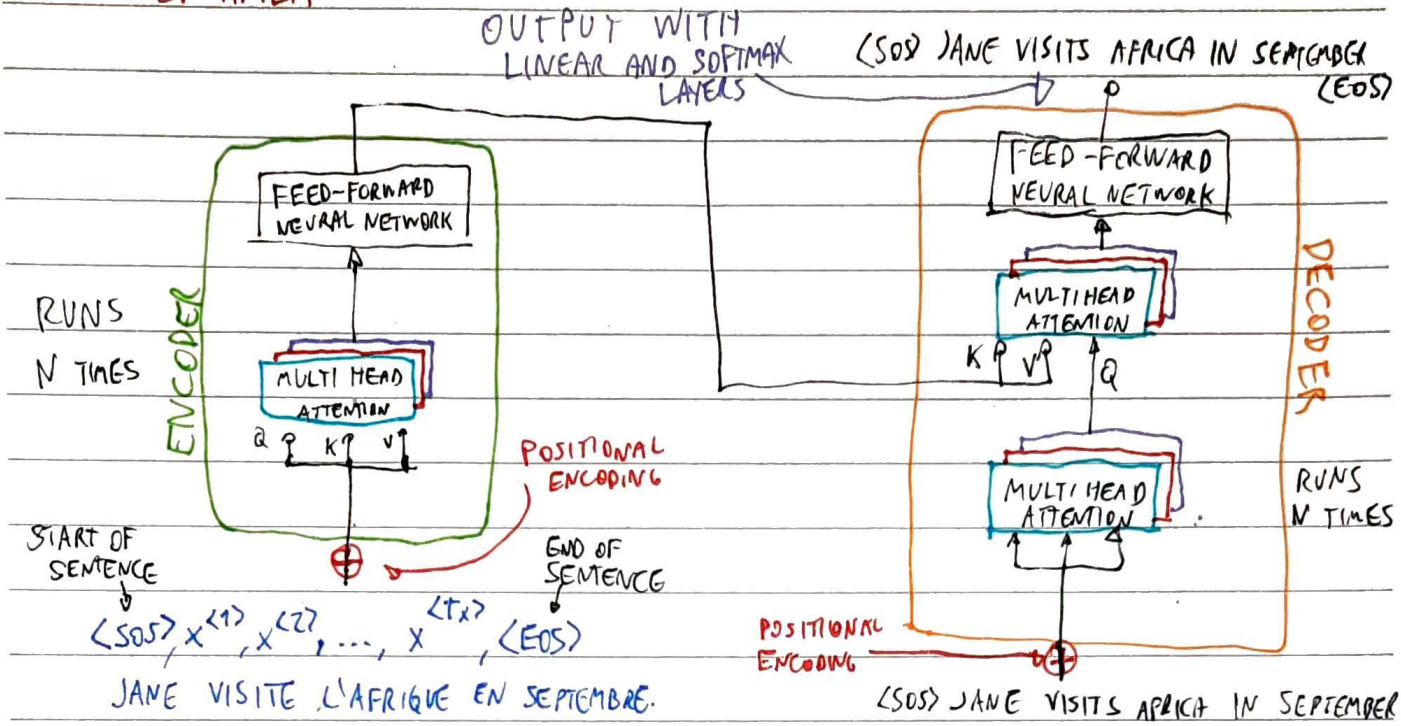
HEAD 3: WHO?
W_3^Q, W_3^K, W_3^V
BEST ANSWER: $K^{<1>}$

$$\text{MULTIHEAD}(q, K, V) = \text{CONCAT}(\text{HEAD}_1, \text{HEAD}_2, \cdots, \text{HEAD}_h) W_o$$

$$\text{HEAD}_i = \text{ATTENTION}(W_i^q q, W_i^K K, W_i^V V)$$

• h: NUMBER OF HEADS

TRANSFORMER



TRANSFORMER DETAILS

* POSITIONAL ENCODING

$$PE(pos, 2i) = \sin\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$

pos: POSITION OF WORD ON THE SENTENCE

i: FEATURE OF THE EMBEDDING VECTOR

$$PE(pos, 2i+1) = \cos\left(\frac{pos}{10000^{\frac{2i}{d}}}\right)$$

- POSITIONAL ENCODING CREATES A UNIQUE POSITION VECTOR FOR EACH WORD, WHICH ENCODES THE POSITION OF THE WORD
- POSITIONAL ENCODINGS COULD ALSO BE PASSED THRU RESIDUAL CONNECTIONS

* ADD & NORM

- "BATCHNORM-LIKE" LAYER

* MASKED MULTI-HEAD ATTENTION

- TECHNIQUE FOR TRAINING