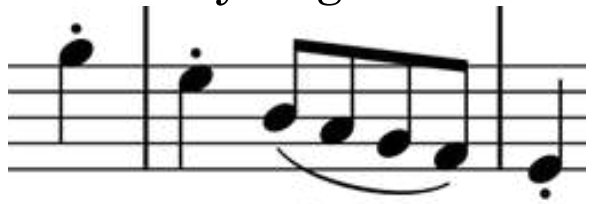
deeplearning.ai

Recurrent Neural Networks

Why sequence models?

Examples of sequence data

Speech recognition		"The quick brown fox jumped over the lazy dog."
Music generation		
Sentiment classification	"There is nothing to like in this movie."	★☆☆☆☆
DNA sequence analysis	AGCCCCTGTGAGGAACTAG	AGCCCCTGTGAGGAACTAG
Machine translation	Voulez-vous chanter avec moi?	Do you want to sing with me?
Video activity recognition		Running
Name entity recognition	Yesterday, Harry Potter met Hermione Granger.	Yesterday, Harry Potter met Hermione Granger.

Andrew Ng

deeplearning.ai

Recurrent Neural Networks

Notation

Motivating example

NLP ~ Notation ~

Input sentence:

x:

Harry Potter and Hermione Granger invented a new spell.

Sequence → $x^{\langle 1 \rangle}$ $x^{\langle 2 \rangle}$ $x^{\langle 3 \rangle}$ .... $x^{\langle t \rangle}$ .... $x^{\langle 9 \rangle}$

$T_x = 9$ len. de la seq

Output sequence:

y: 1 1 0 1 1 0 0 0 0

$y^{\langle 1 \rangle}$ $y^{\langle 2 \rangle}$ $y^{\langle 3 \rangle}$ .... $y^{\langle 9 \rangle}$

El ej. es de "Name Entity Recognition." Busca id. donde hay Nombres en la sequence.

$X^{(i)\langle t \rangle}$ $y^{(i)\langle t \rangle}$

Notación p/ el elemento t de la seq (Train example) i

$T_x^{(i)} = 9$

$T_y^{(i)}$

$T_y = 9$

Is

T_y Puede ser ≠ de T_x y

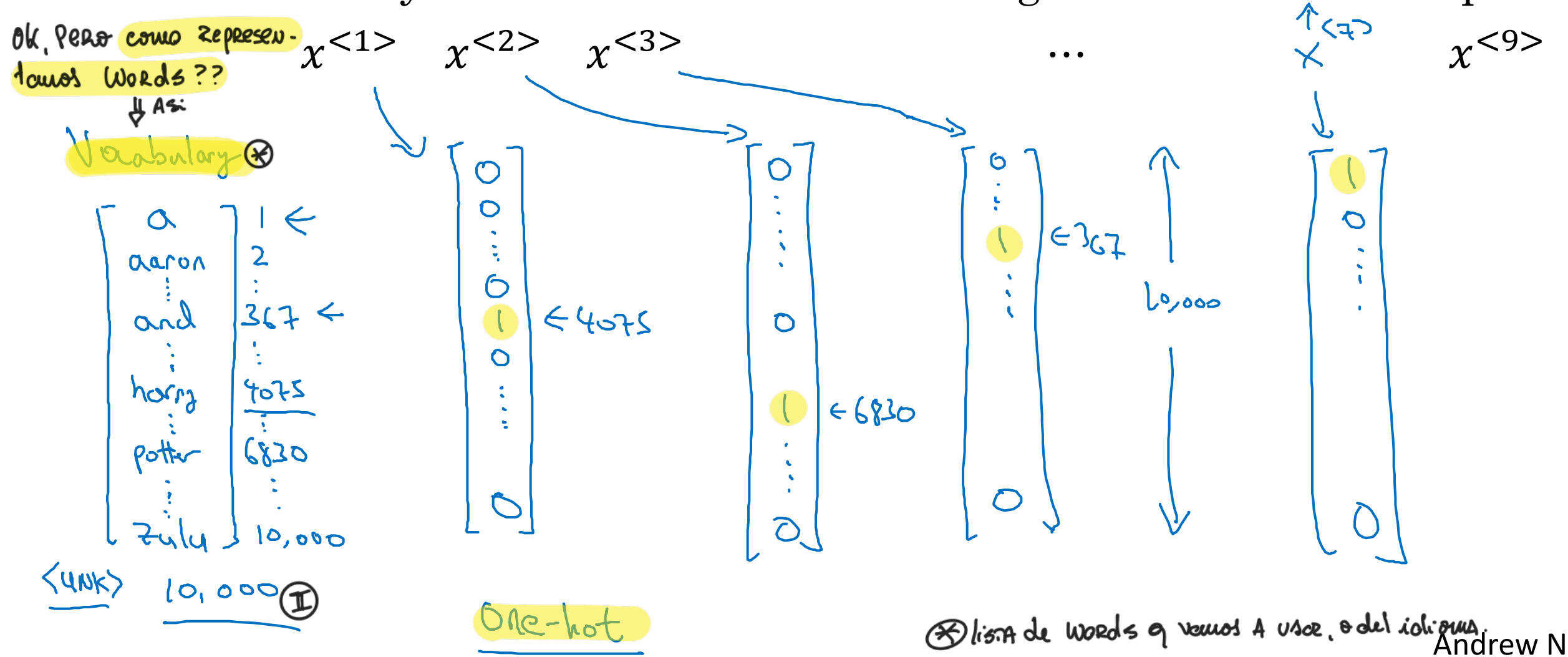
Además c/ Training example

Puede ser len ≠ → $T_x^{(i)} \neq T_x^{(j)}$

Andrew Ng

Representing words

x: Harry Potter and Hermione Granger invented a new spell.



Representing words

x: Harry Potter and Hermione Granger invented a new spell.

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

And = 367
Invented = 4700
A = 1
New = 5976
Spell = 8376
Harry = 4075
Potter = 6830
Hermione = 4200
Gran… = 4000

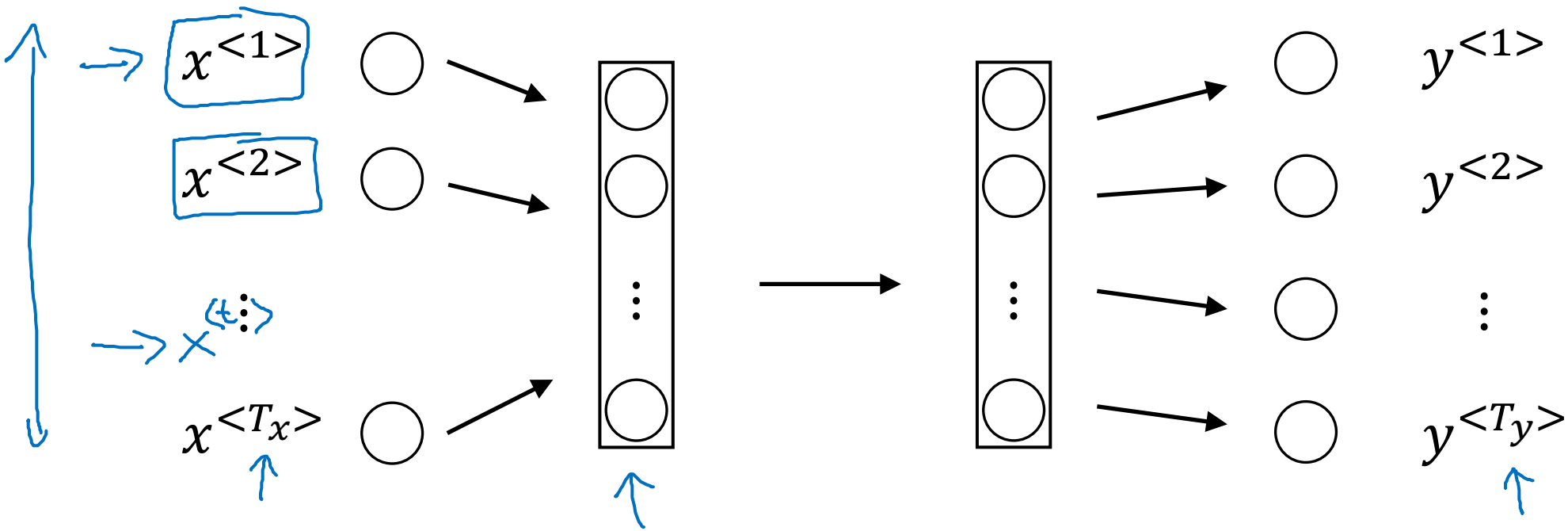
deeplearning.ai

Recurrent Neural Networks

Recurrent Neural Network Model

Why not a standard network?

feed forward NLP



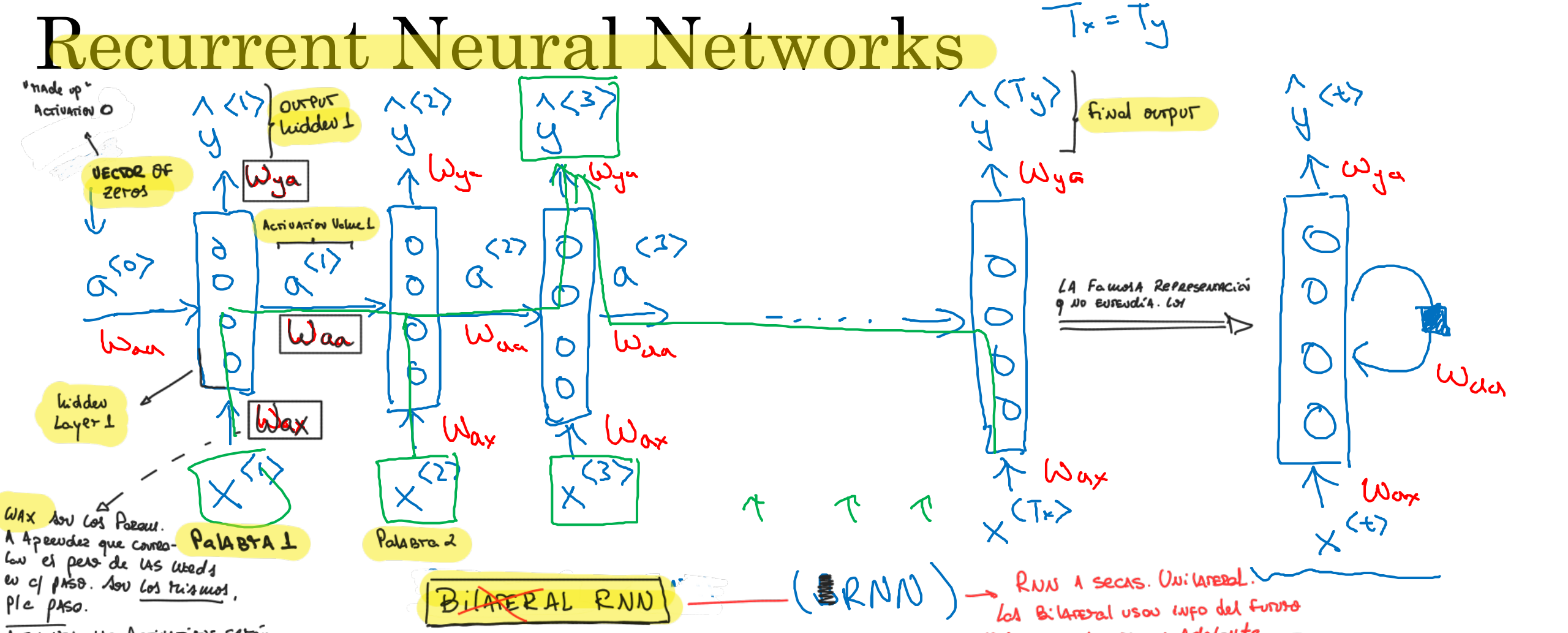
Problems:

- Inputs, outputs can be different lengths in different examples.
- Doesn't share features learned across different positions of text.

Análogo a la convolución. No explota la estructura secuencial y no generaliza a través de ella.

Andrew Ng

Recurrent Neural Networks



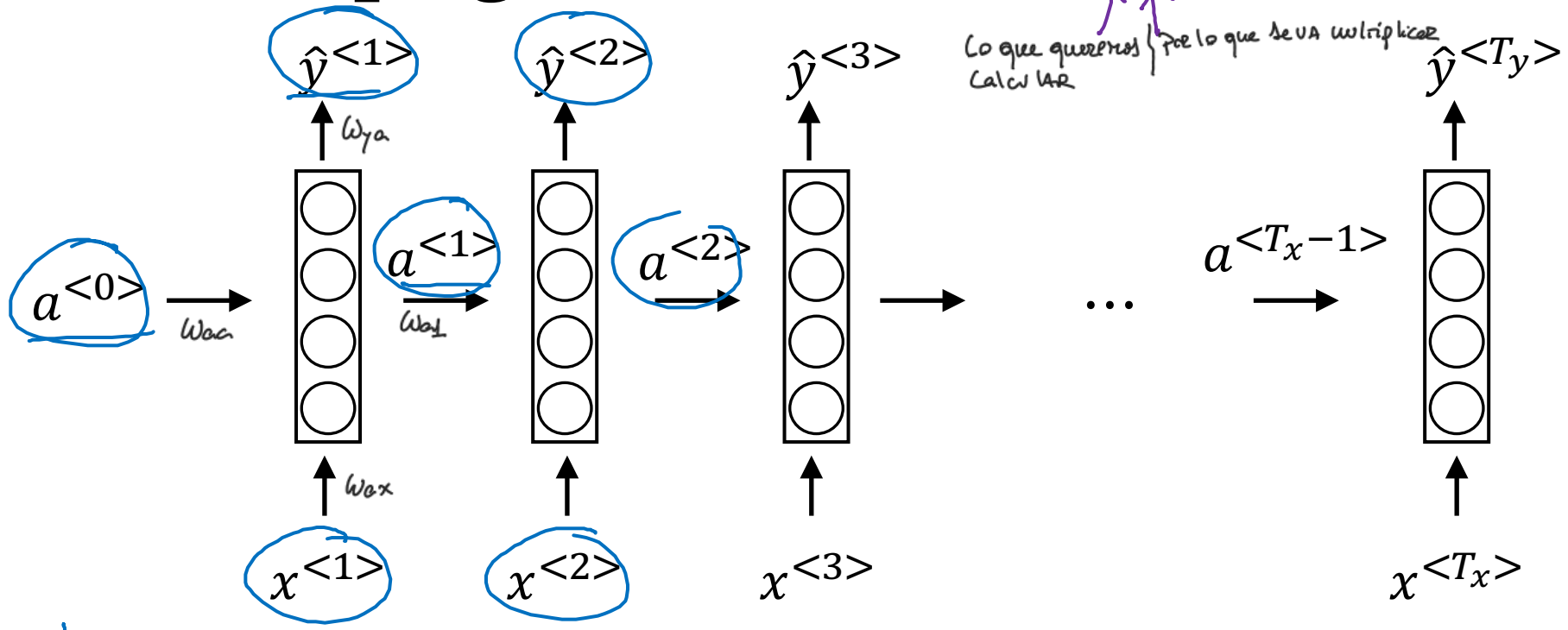
Andrew Ng

Forward Propagation

Notación de los Weights: (I)

$a \leftarrow W_{ax} x^{\langle 1 \rangle}$

Lo que queremos calcular | por lo que se va multiplicar



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

$a^{\langle 1 \rangle} = g_1(W_{aa} a^{\langle 0 \rangle} + W_{ax} x^{\langle 1 \rangle} + b_a)$ ← tanh / ReLU (g1)

$\hat{y}^{\langle 1 \rangle} = g_2(W_{ya} a^{\langle 1 \rangle} + b_y)$ ← Sigmoid → g_2 → puede ser +

VER (I)

RNN Forward Prop.

$a^{\langle t \rangle} = g(W_{aa} a^{\langle t-1 \rangle} + W_{ax} x^{\langle t \rangle} + b_a)$ → Generalizando

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

Andrew Ng

Simplified RNN notation

Matricialmente. y simplificando con notación.

Partiendo de (notación matricial)

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

W_{aa}: 100 $\in \mathbb{R}^{(100,100)}$ — W_{ax}: 10,000 $\in \mathbb{R}^{(100,10,000)}$

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

Analogamente, simplificando:

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

Finalmente:

Forward Prop. RNN Simplificado

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

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

Reescribimos ($\equiv$)

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

donde:

① $W_a = \begin{bmatrix} W_{aa} ; W_{ax} \end{bmatrix}$ (100 × 100 | 10,000) $\in \mathbb{R}^{(100, 10100)}$

La matriz W_a es W_{ax} y W_{aa} "pegadas" horizontalmente a lo largo del eje x.

② $[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] = \begin{bmatrix} a^{\langle t-1 \rangle} \\ x^{\langle t \rangle} \end{bmatrix}$ (100 + 10000 = 10100)

A su vez, esta not. es poner los $\vec{v}$ uno arriba del otro.

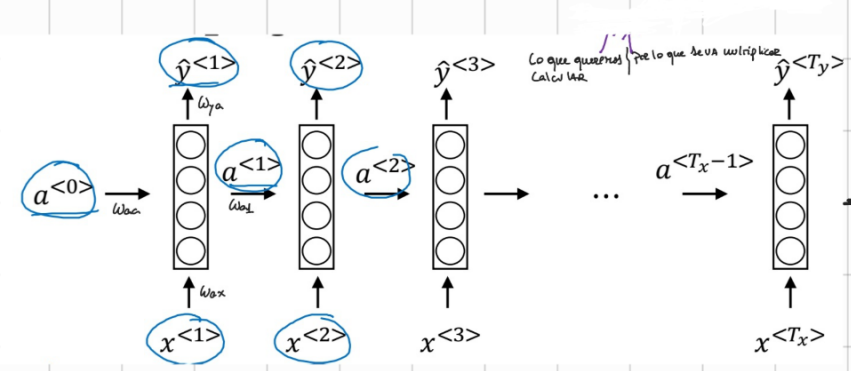
③ $[W_{aa} ; W_{ax}] \begin{bmatrix} a^{\langle t-1 \rangle} \\ x^{\langle t \rangle} \end{bmatrix} = W_{aa}a^{\langle t-1 \rangle} + W_{ax}x^{\langle t \rangle}$

($\equiv$)

Andrew Ng

Lo mismo, más ordenado:

Si partimos de nuestra notación original p/ la BRNN, tenemos:



FORWARD PROP:

ver (1)

$$a^{<t>} = g(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a) \quad (1)$$
$$\hat{y}^{<t>} = g(W_{ya}a^{<t>} + b_y)$$

Si escribimos (1) en forma matricial, obtenemos:

$$\begin{cases} a^{<t>} = g(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a) \\ \hat{y}^{<t>} = g(W_{ya}a^{<t>} + b_y) \end{cases}$$

y con la notación "stackeada" definida en ① y ② de la página anterior, obtenemos por último:

RNN

Forward Prop. Simplificado

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

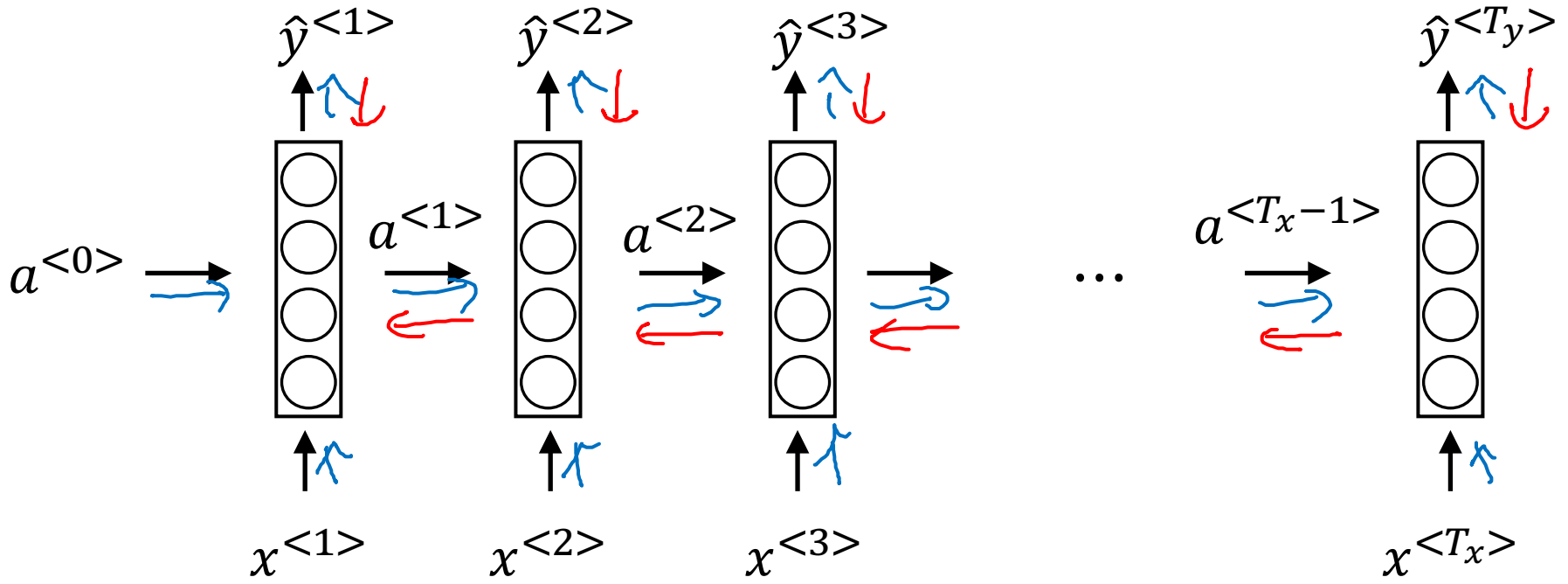
deeplearning.ai

Recurrent Neural Networks

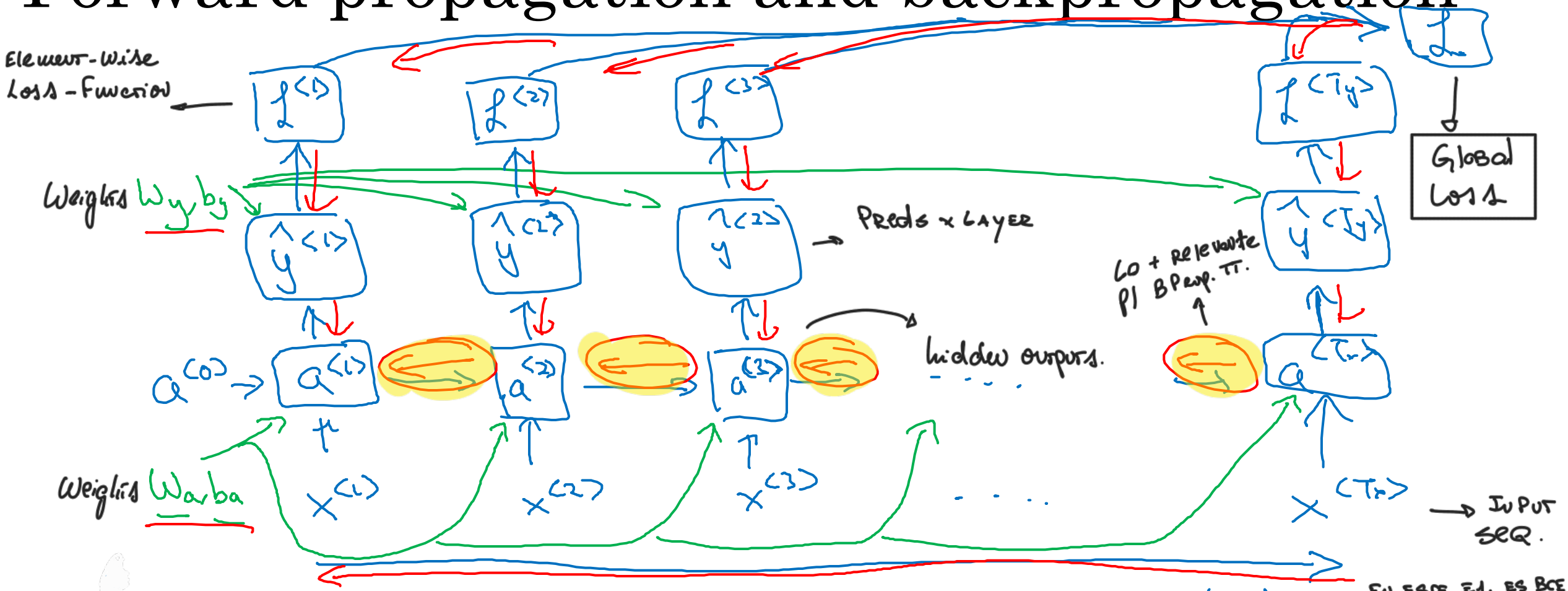
Backpropagation through time

Forward propagation and backpropagation

- FORWARD
- BACKWARD



Forward propagation and backpropagation



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

En este ej. es BCE Loss. Asociada con la loss de una pos. en particular. (una word)

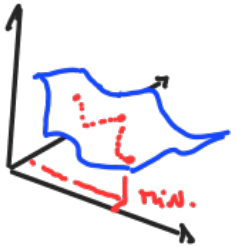
$$\mathcal{L}(\hat{y}, y) = \sum_{t=1}^{T_y} \mathcal{L}^{<t>}(\hat{y}^{<t>}, y^{<t>})$$

La Loss p/ toda la sequencia.

Backpropagation through time

Andrew Ng

↳ Calculando la Loss, ya podes calcular las derivadas de $\frac{d\,h}{d\,W_i}$ c/ parametro y con el $\vec{\nabla}$ gradiente de derivadas parciales de Params, minimizar $h(W_i)$ en el ESPACIO de PARAMETROS con GRADIENT DESCENT ☺.



Recurrent Neural Networks

deeplearning.ai

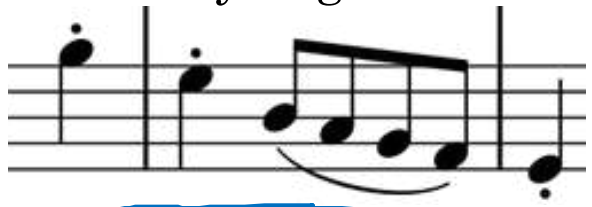
Different types of RNNs

HASTA AHORA solo vimos ARCHS que inputs = outputs ($T_x = T_y$). pero ESTO puede no ser siempre así:

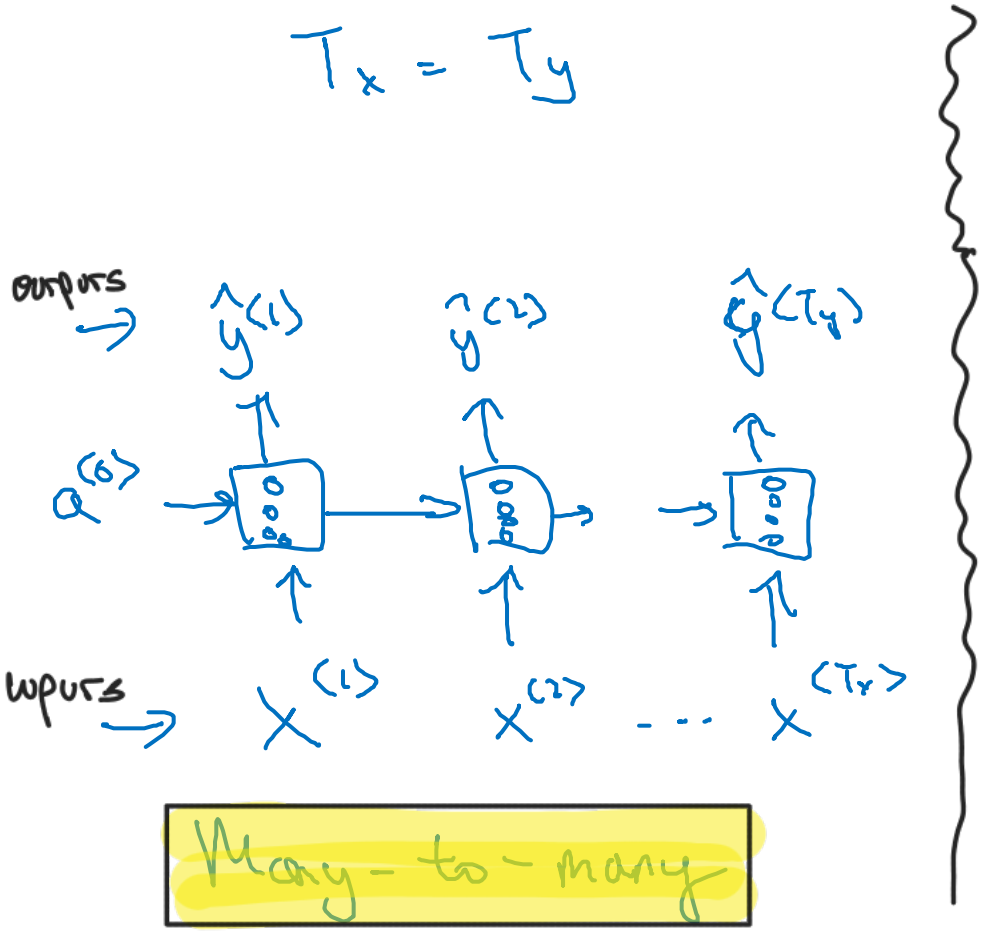
Examples of sequence data

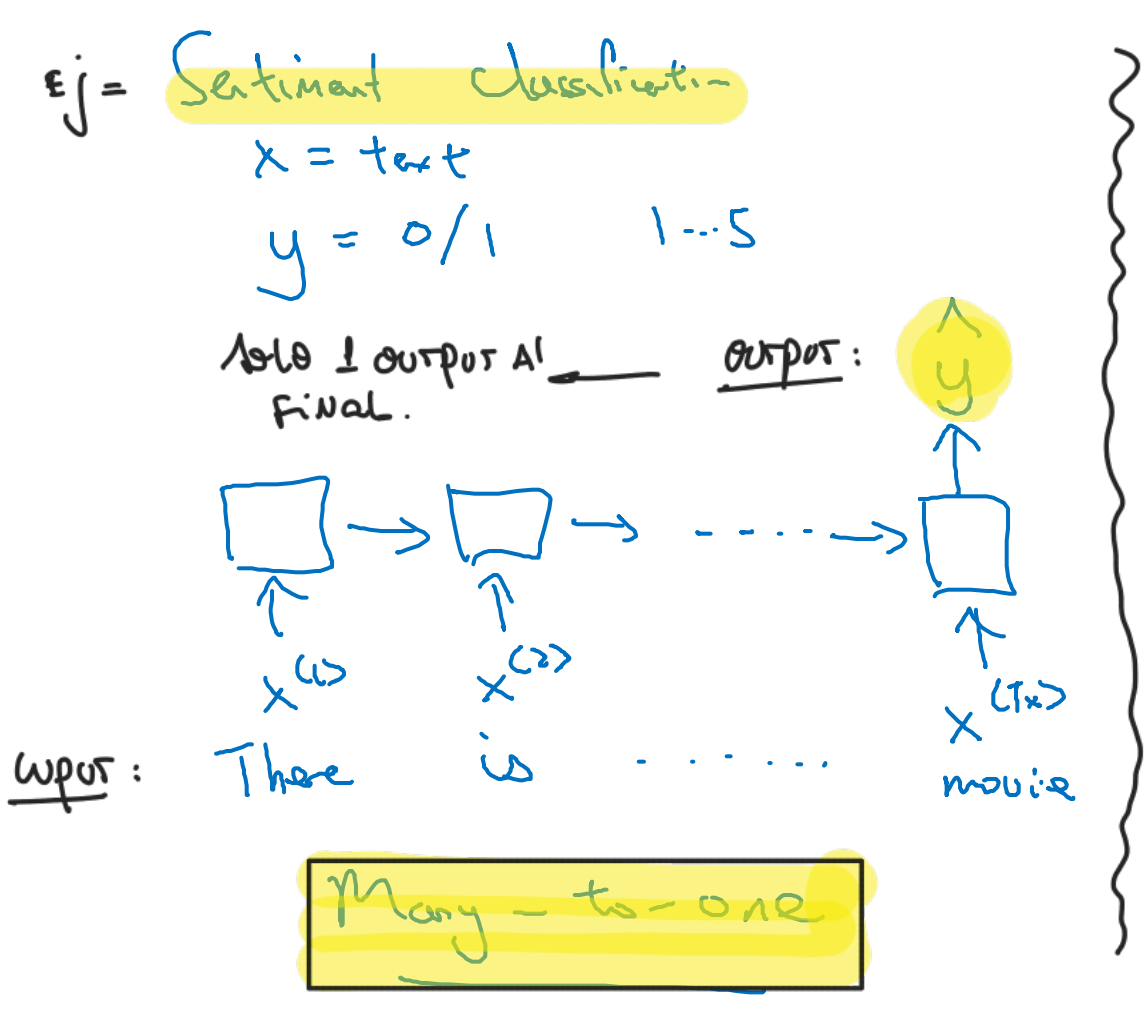
Tx Ty

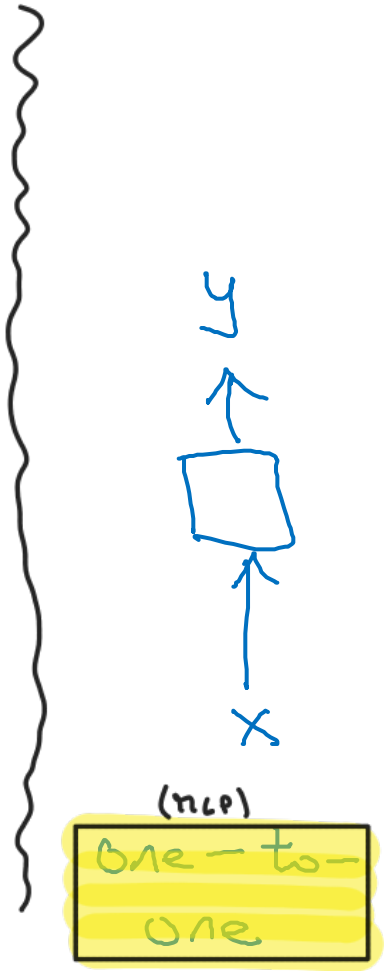
x y

Speech recognition		→	"The quick brown fox jumped over the lazy dog."
Music generation	∅	→	
Sentiment classification	"There is nothing to like in this movie."	Ej de Tx f Ty →	
DNA sequence analysis	AGCCCCTGTGAGGAACTAG	→	AGCCCCTGTGAGGAACTAG
Machine translation	Voulez-vous chanter avec moi?	Ej de Tx f Ty →	Do you want to sing with me?
Video activity recognition		→	Running
Name entity recognition	Yesterday, Harry Potter met Hermione Granger.	→	Yesterday, Harry Potter met Hermione Granger.

Examples of RNN architectures

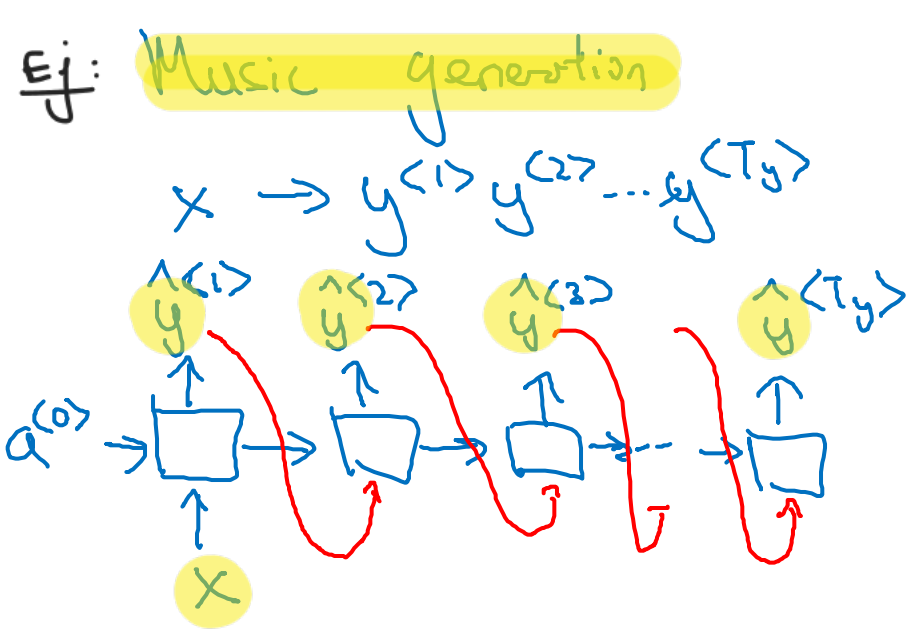


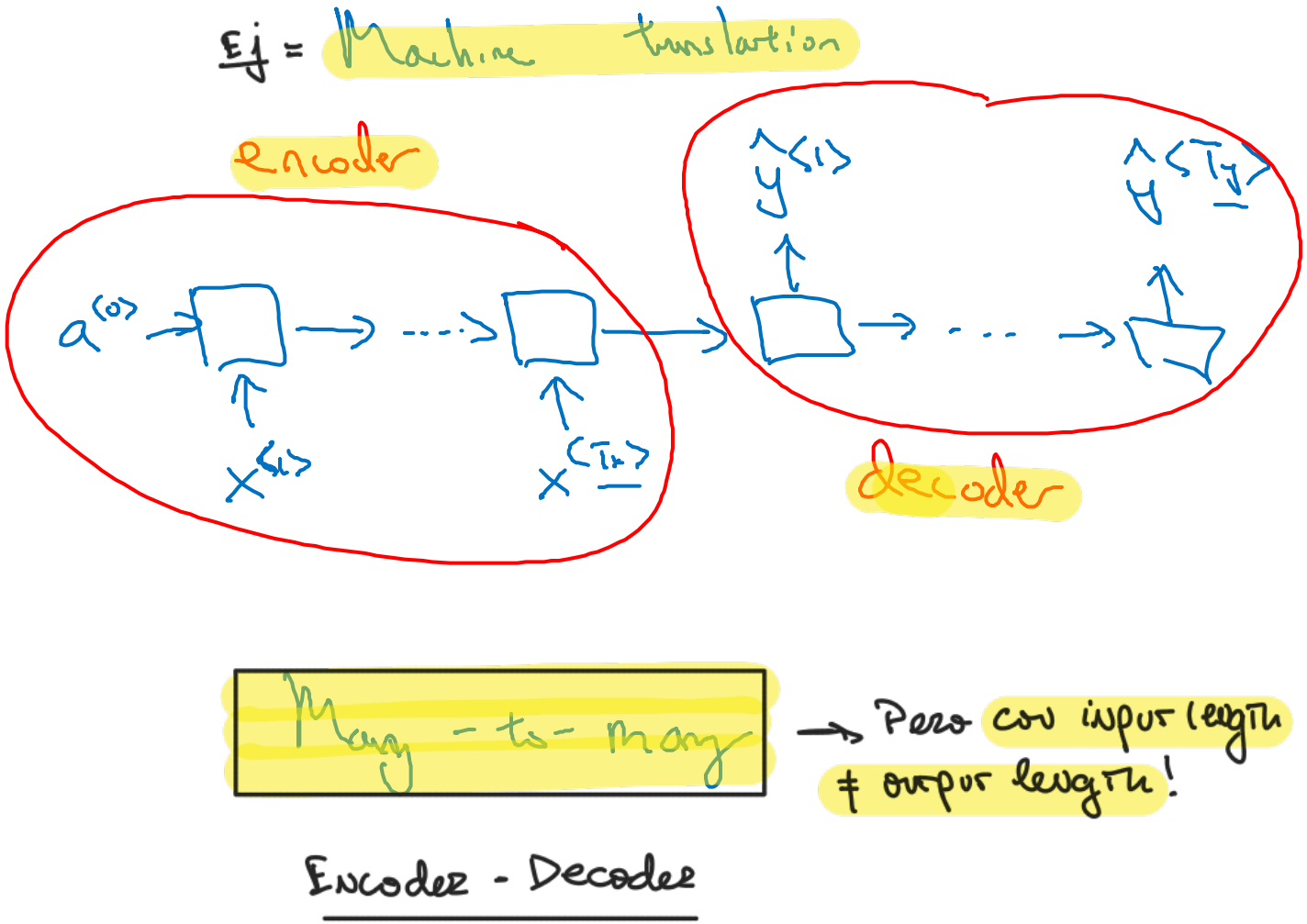




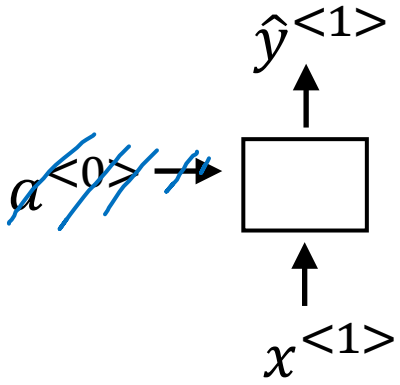
Andrew Ng

Examples of RNN architectures

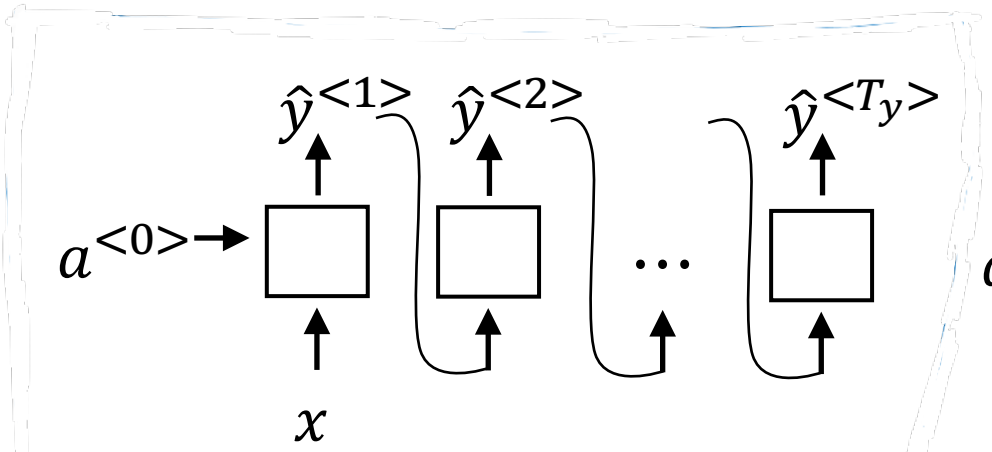




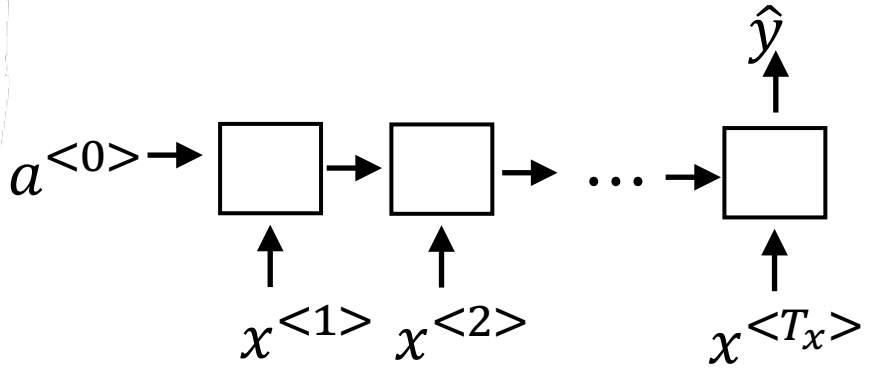
Summary of RNN types



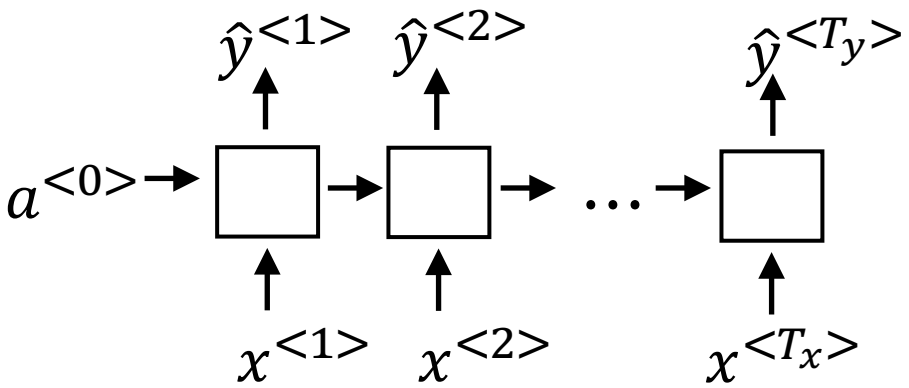
One to one



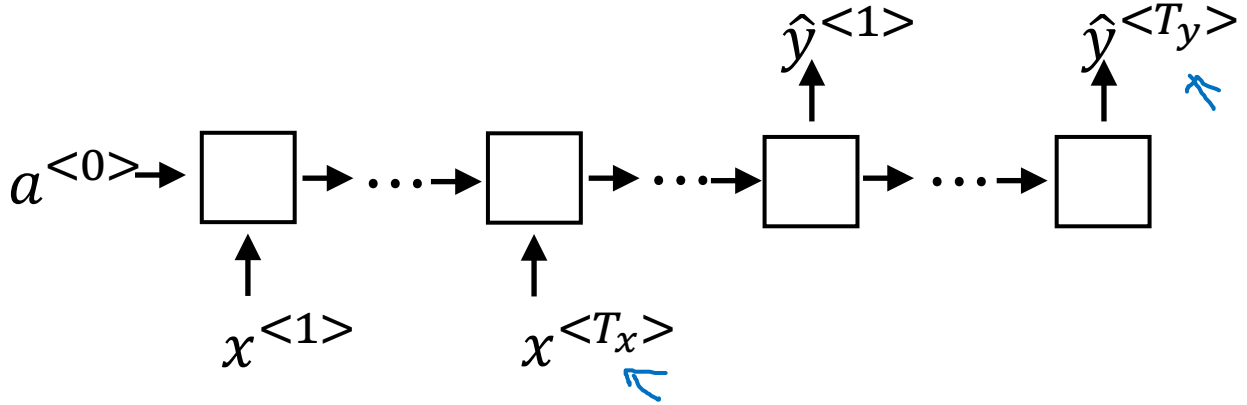
One to many



Many to one



Many to many $T_x = T_y$



Many to many $T_x \neq T_y$

Andrew Ng

deeplearning.ai

Recurrent Neural Networks

Language model and sequence generation

god. sequence gen.
music, text, etc.
god.

What is language modelling?

Speech recognition

The apple and pair salad.

→ The apple and pear salad.

Si dije esto (Audio), cuál es más probable que haya dicho poRA?

$P(\text{The apple and pair salad}) = 3.2 \times 10^{-13}$

$P(\text{The apple and pear salad}) = 5.7 \times 10^{-10}$

$\Rightarrow P(\text{PEAR}) > P(\text{PAIR})$

$\Rightarrow$ $P(\text{Sentence}) = ?$ $\longrightarrow$ $P(y^{\langle 1 \rangle}, y^{\langle 2 \rangle}, \ldots, y^{\langle T_y \rangle})$

Language modelling with an RNN

y... Cómo build decimos un Language model?

Training set: large corpus of english text.

Tokenize → Por AHORA Andrew solo habla de one-hot encoding. le sig. semana entran a jugar los embeddings.

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

$y^{<1>}$ $y^{<2>}$ $y^{<3>}$ ... $y^{(8)}$ $y^{<9>}$

$x^{<t>} = y^{<t-1>}$ → andrew dice q vamos a terminar setteando el Nº de inputs $x^{<t>} = y^{<t-1>}$, que después nos explica pq.

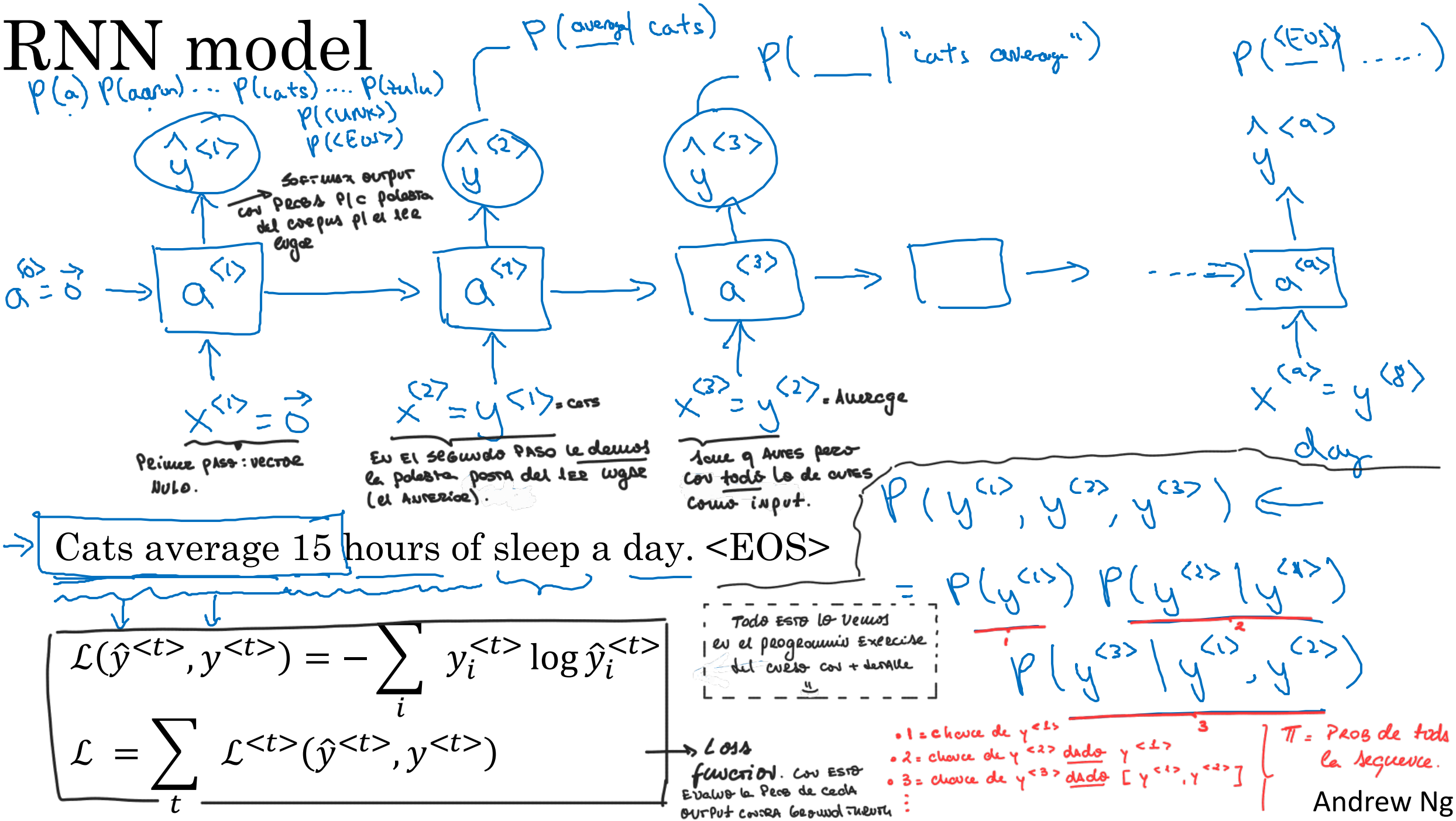
The Egyptian ~~Mau~~ is a bread of cat. <EOS>

<UNK>

10,000

token p/ PALABRAS que ∄ corpus.

RNN model



Andrew Ng

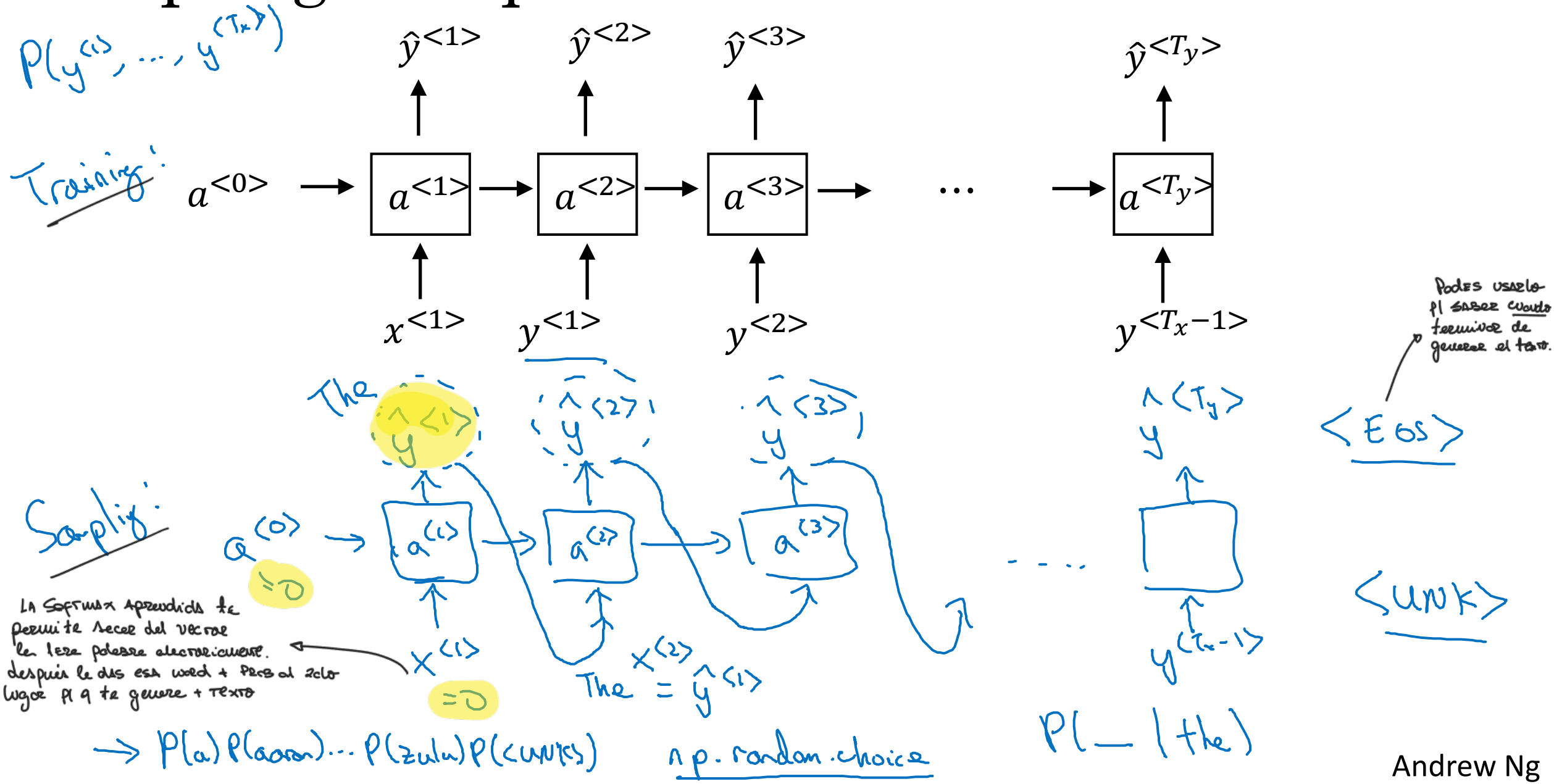
deeplearning.ai

Recurrent Neural Networks

Sampling novel sequences

Una vez que entrenamos el modelo, podemos pedirle q haga esto.

Sampling a sequence from a trained RNN



Character-level language model

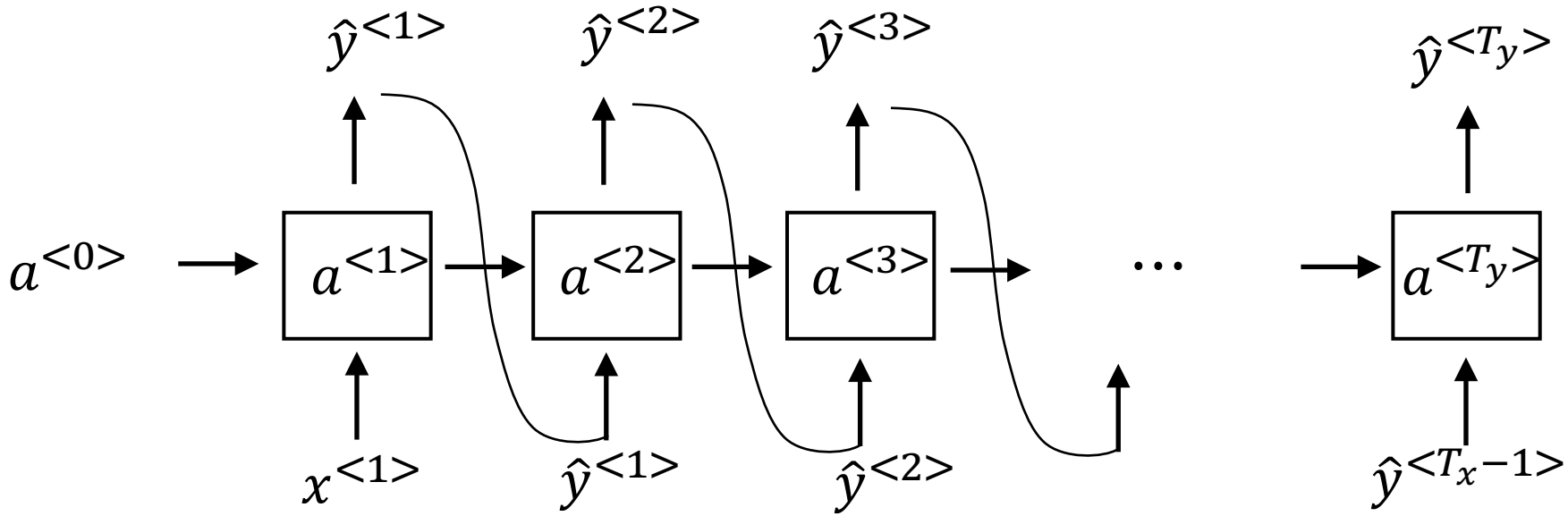
- Vocabulary = [a, aaron, …, zulu, <UNK>] → word level

- Vocabulary = [a, b, c, …, z, ␣, ., ,, ;, 0, …, 9, A, …, Z] → CHARACTER LEVEL

④ more computationally expensive pero te dejan modelar palabras que no ✱.

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

Cat average … Mau



Andrew Ng

Sequence generation

News

President enrique peña nieto, announced sench's sulk former coming football langston paring.

"I was not at all surprised," said hich langston.

"Concussion epidemic", to be examined.

The gray football the told some and this has on the uefa icon, should money as.

Shakespeare

The mortal moon hath her eclipse in love.

And subject of this thou art another this fold.

When besser be my love to me see sabl's.

For whose are ruse of mine eyes heaves.

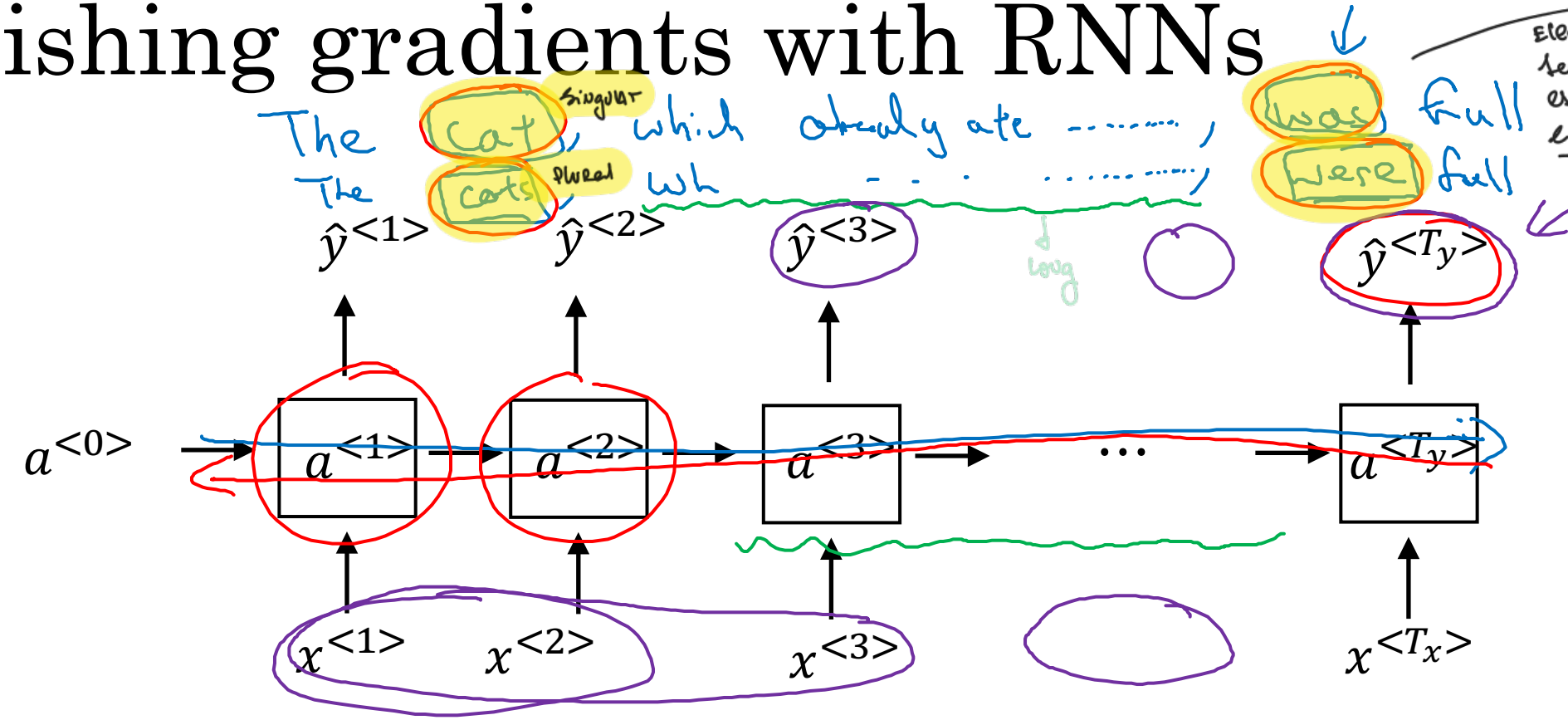
deeplearning.ai

Recurrent Neural Networks

GRU's and LSTM's

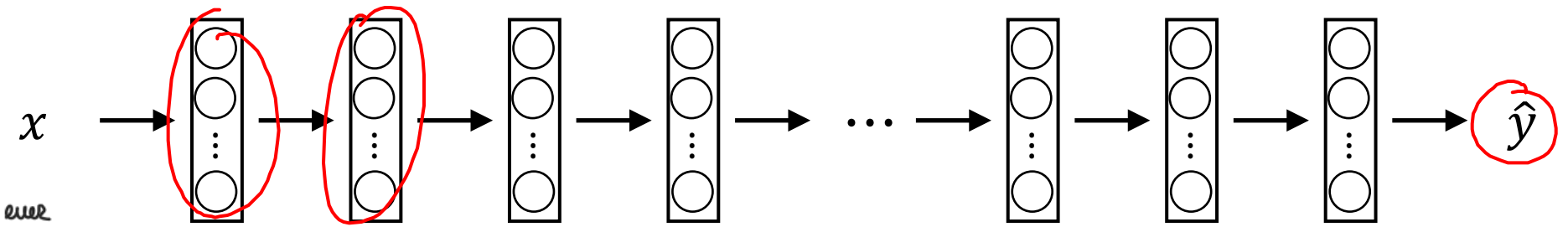
Vanishing gradients with RNNs

Vanishing gradients with RNNs



El modelo debe elegir uno según el cese pl lee correcta. el term es que depende cats q está lejos → long term dependency

Vanishing Gradients Problem



También podemos tener este problema. aunque es menos frecuente. es más fácil de ver por el overflow y q se vuelven a los gorros los Parem.

Exploding gradients.

NaN

Gradient clipping → max threshold q pones pl escalar Exploding Gradients.

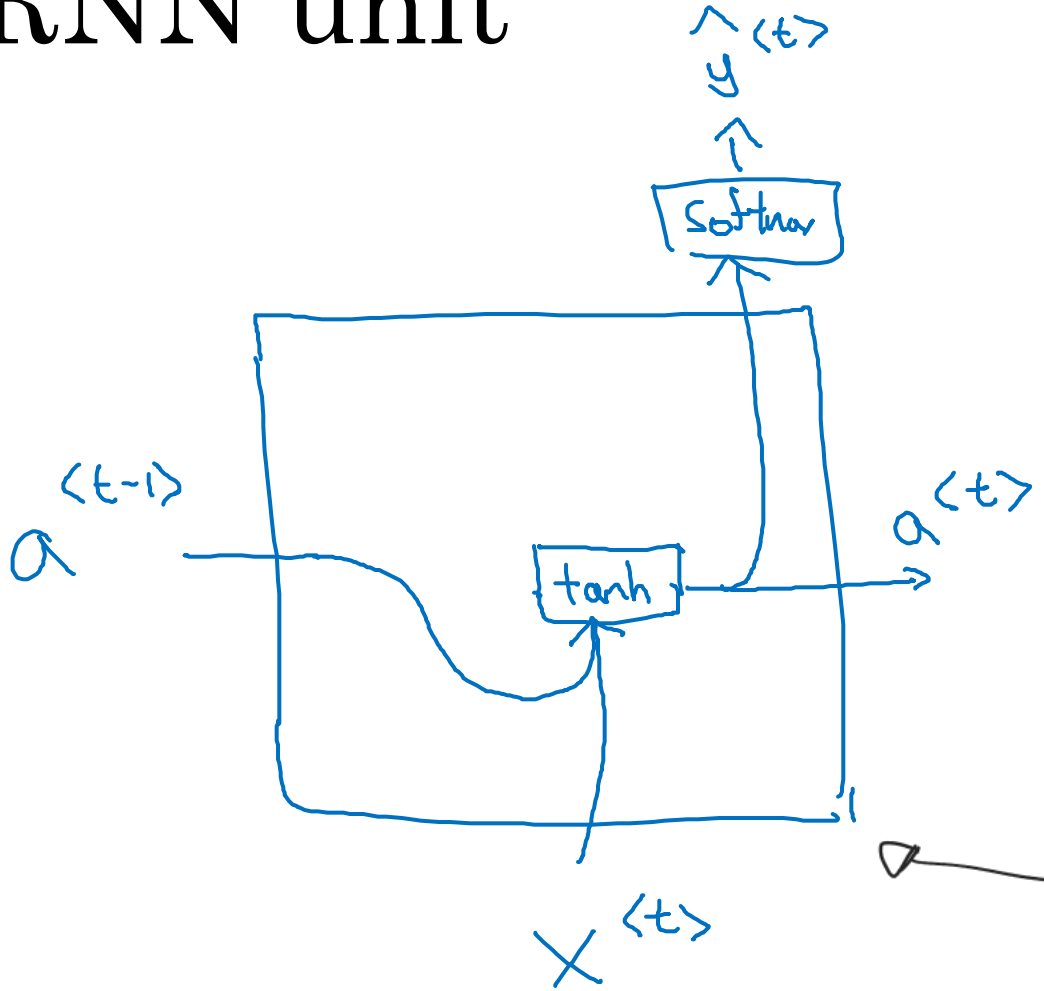
Andrew Ng

deeplearning.ai

Recurrent Neural Networks

Gated Recurrent Unit (GRU)

RNN unit



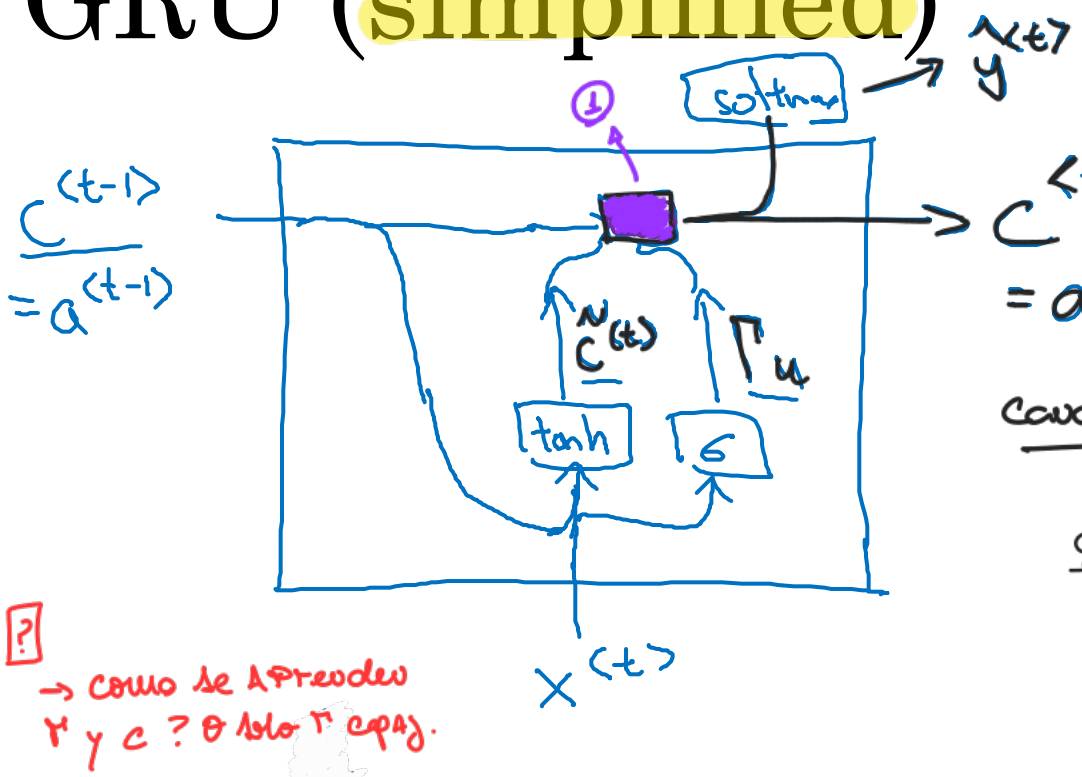
→ Fórmula para la activación de una RNN, tal como lo vimos:

tanh

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

Gráficamente, la hidden layer sería algo así.

GRU (simplified)



? → como se aprenden Γ y c ? Θ solo Γ cpaz.

candidato que en el timestamp va a INTENTAR reemplazar la memoria "$c^{\langle t \rangle}$"

es la hidden unit pero con otro nombre.

c = memory cell

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

Valor previo de la memoria

Valor actual leg.

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

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

"update"

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

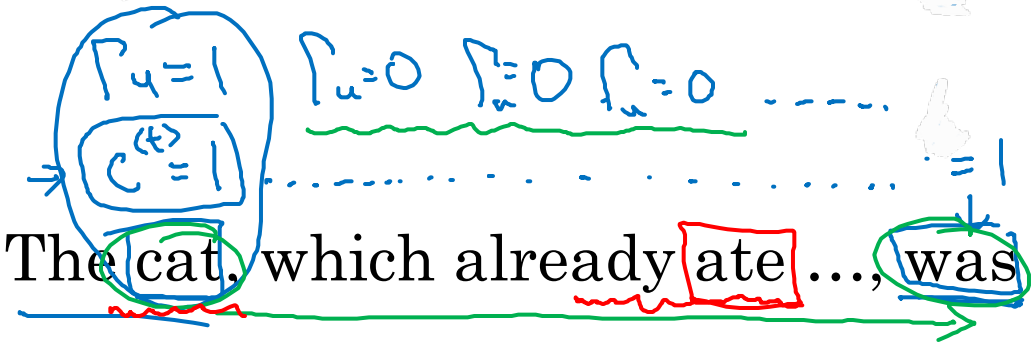
$\Gamma_u = 1$

Recalculamos $c^{\langle t \rangle}$ memoria en base a Γ y $\hat{c}$. el gate filtra al candidato.

element-wise

$\Gamma_u = 0.000001$

Mantiene el valor de c por ser $\Gamma \approx 0$ Previene los Vanishing ∇.



→ The cat, which already ate …, was full.

[Cho et al., 2014. On the properties of neural machine translation: Encoder-decoder approaches]

[Chung et al., 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling]

Andrew Ng

Full GRU

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

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

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

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

→ Notación de Andrew

→ Notación más académica.

Aprende la Relevancia de $c^{<t-1>}$ p/ computar el sig. candidato p/ $c^{<t>}$

→ Porque agregamos Γ_r y no usamos la versión simple de antes? el research indica que tenerla es mejor

→ En cuanto a como se aprenden los gates, creo que la Rta esta en los W_r y W_u q serían los weights (matrices) de estas unidades.

The cat, which ate already, was full.

Andrew Ng

deeplearning.ai

Recurrent Neural Networks

LSTM (long short term memory) unit

Andrew dice que son + power que las GRU.

GRU and LSTM

GRU

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

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

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

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

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

Γ_f

LSTM

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

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

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

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

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

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

[Hochreiter & Schmidhuber 1997. Long short-term memory]

LSTM units

GRU

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

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

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

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

una sola gate actua como updt + forget

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

so, is $c^{<t>}$ the "hidden state" or dim in LSTM's?

ACA: $a^{<t>} \neq c^{<t>}$

dim = #hidden units LSTM $a^{<t>}$

LSTM

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

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

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

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

2 gates ↑ ↑

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

$$a^{<t>} = \Gamma_o * \tanh(c^{<t>})$$ → final output of the unit. A softmax is then applied to obtain $y^{<t>}$.

[Hochreiter & Schmidhuber 1997. Long short-term memory]

Andrew Ng

LSTM in pictures

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

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

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

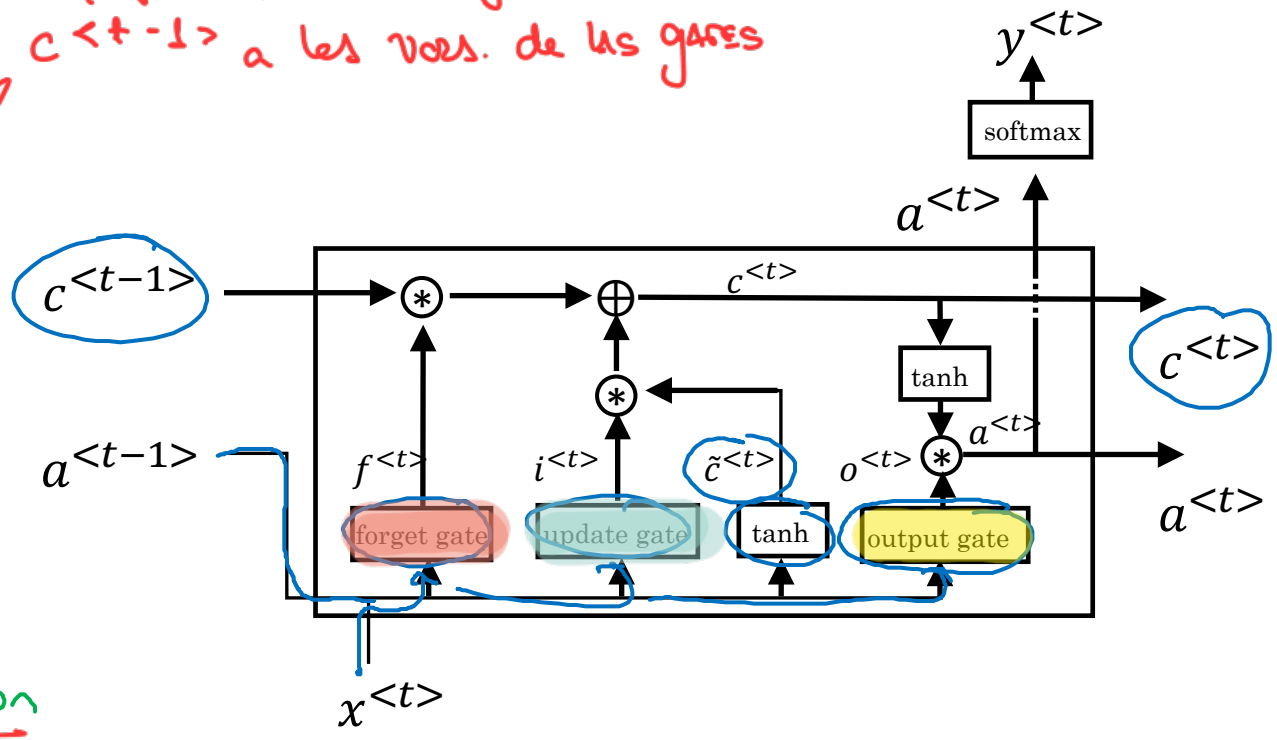
$$\Gamma_o = \sigma(W_o[a^{<t-1>}, x^{<t>}] + b_o)$$

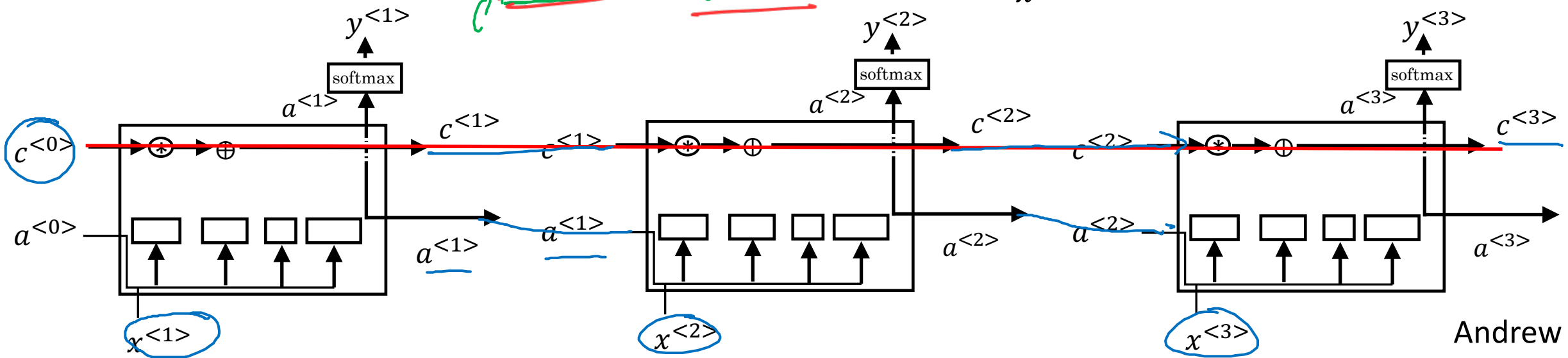
$$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + \Gamma_f * c^{<t-1>}$$

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

Hay gente q ademas agrega el $c^{<t-1>}$ a las ecs. de las gates

peephole connection





Andrew Ng

→ No hay consenso de cuando usar GRU's y LSTMs. Conviene probar los dos.

deeplearning.ai

Recurrent Neural Networks

Ox for lunch

Bidirectional RNN

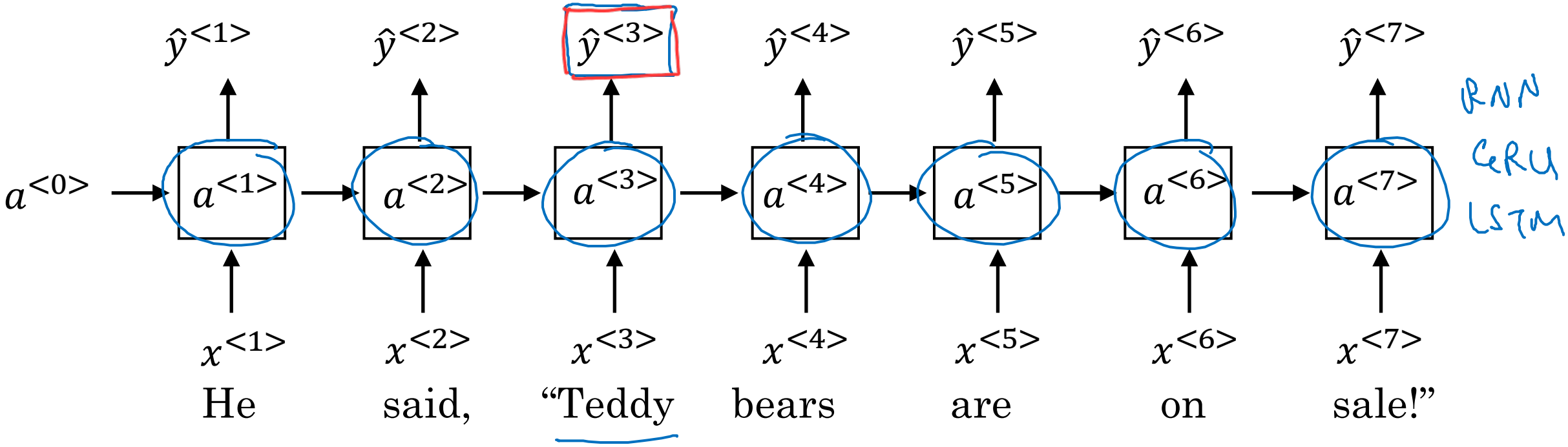
getting info from the future.

Getting information from the future

He said, “Teddy bears are on sale!”

He said, “Teddy Roosevelt was a great President!”

Bear or President? The answer is in the future.



RNN

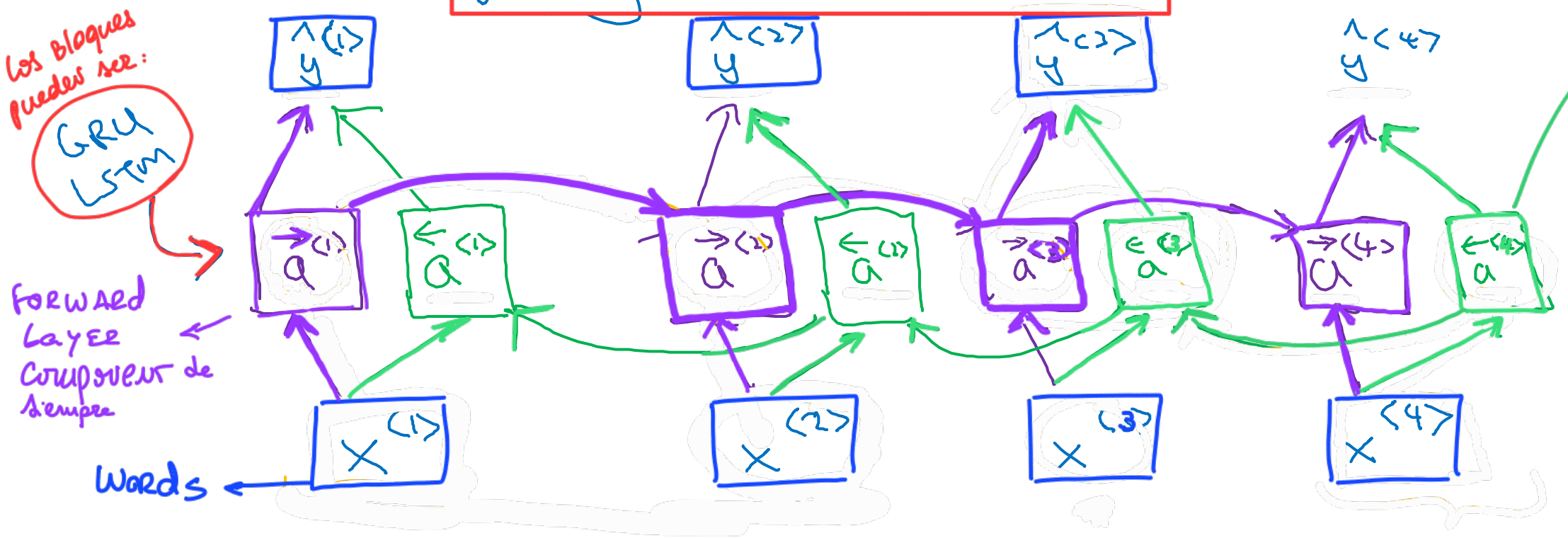
GRU

LSTM

Bidirectional RNN (BRNN)

En el paso 3, para ej. te llega info de 2 y 1 en $\overrightarrow{a}$ e info de 4 en $\overleftarrow{a}$

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



Acyclic graph

desventaja ⇒ necesitas toda la seq de una pl hacer la predicción. pl real time no sirve tanto pq tienes q esperar a que termine de hablar la persona.

He said "Teddy Roosevelt ..."

BRNN w/ LSTM

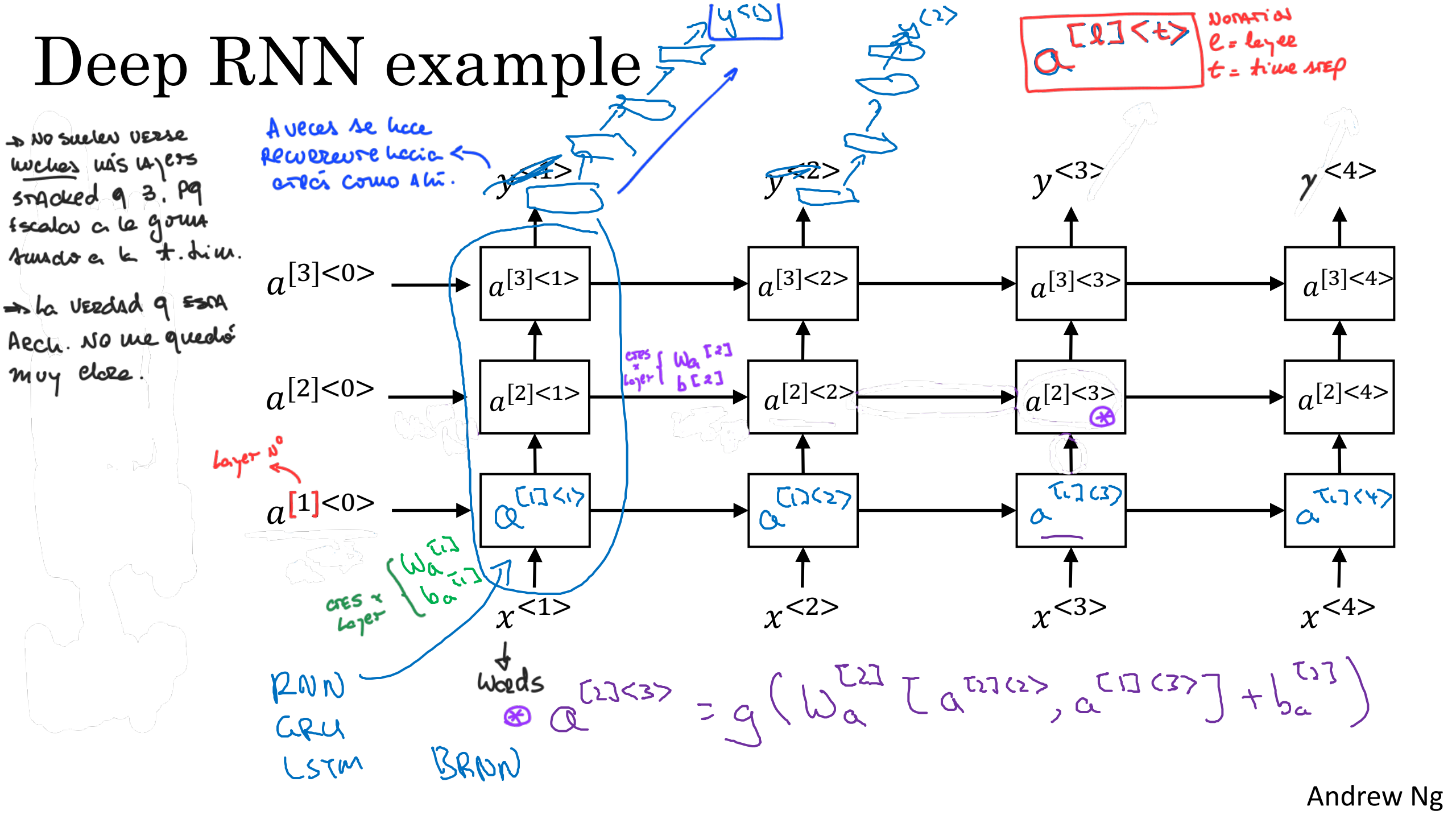
deeplearning.ai

Recurrent Neural Networks

Deep RNNs

Stack multiple layers of RNN's together

Deep RNN example



Andrew Ng

Copyright Notice

deeplearning.ai

NLP and Word Embeddings

Word representation

Word representation

V = [a, aaron, …, zulu, <UNK>]

$|V| = 10{,}000$

1-hot representation

Man (5391)	Woman (9853)	King (4914)	Queen (7157)	Apple (456)	Orange (6257)

O_{5391} O_{9853} → Posición (subíndice) de un Vocab. one hot.

I want a glass of orange juice.

I want a glass of apple ____?.

El problema de representar one hot es q el algoritmo no tiene forma de saber que Apple y Orange tienen algo q ver. esto es q el producto interno de cualquier par one hot es cero y ☹ su diss. tambien.

→ Buscamos otra representación, CON features

Featurized representation: word embedding

"features"	Man (5391)	Woman (9853)	King (4914)	Queen (7157)	Apple (456)	Orange (6257)
Gender	-1	1	-0.95	0.97	0.00	0.01
Royal	0.01	0.02	0.93	0.95	-0.01	0.00
Age	0.03	0.02	0.7	0.69	0.03	-0.02
Food	0.04	0.01	0.02	0.01	0.95	0.97
⋮	⋮	⋮				
Size						
Cost						
Alive						
Verb						

300

e_{5391} e_{9853}

Words

Ahora su representación en este espacio vectorial es parecida

Word Embedding

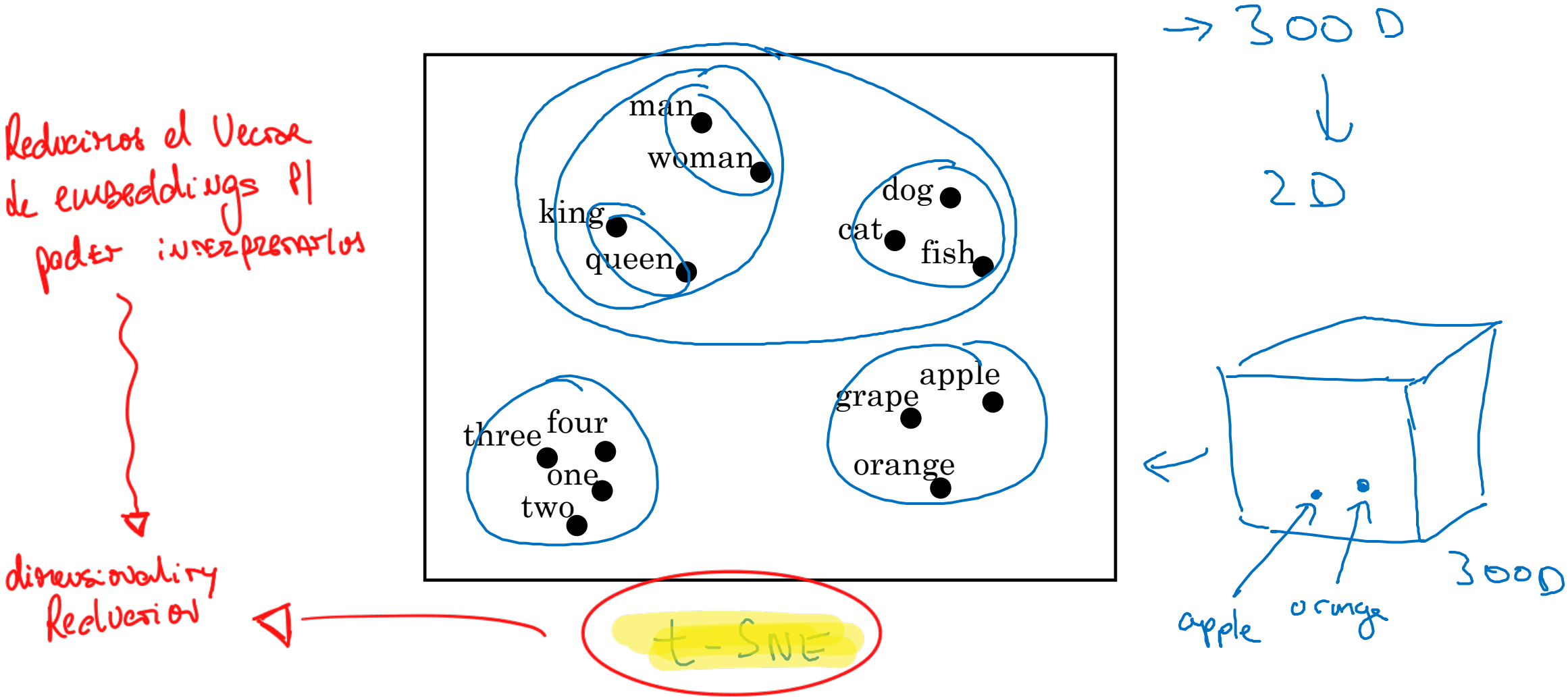
I want a glass of orange juice.

I want a glass of apple juice.

¿Como los aprendemos a los features de c/ palabra?

Andrew Ng

Visualizing word embeddings



[van der Maaten and Hinton., 2008. Visualizing data using t-SNE]

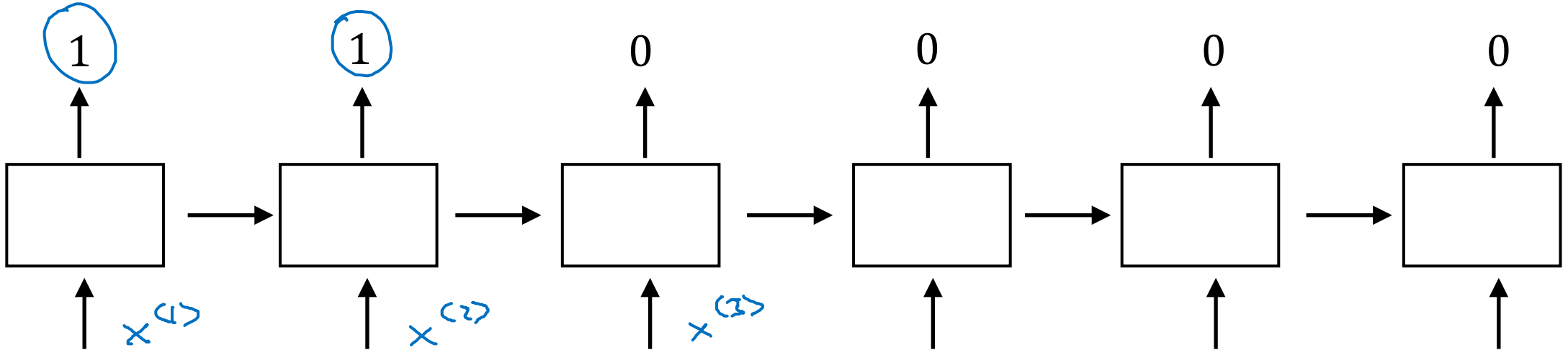
deeplearning.ai

NLP and Word Embeddings

Using word embeddings

(*) Durian y cultivator son palabras raras q pueden no estén en el training set → pero si tri W. emb. sepe q son, puede inferir igual a partir de eso, ya que la palabra en sí comparte info x ej. fruta y x ej. farmer.

Named entity recognition example



Sally Johnson is an orange farmer

Robert Lin is ~~an~~ ~~apple~~ ~~farmer~~

Andrew Ng

Transfer learning and word embeddings

A

1. Learn word embeddings from large text corpus. (1-100B words)

   (Or download pre-trained embedding online.)

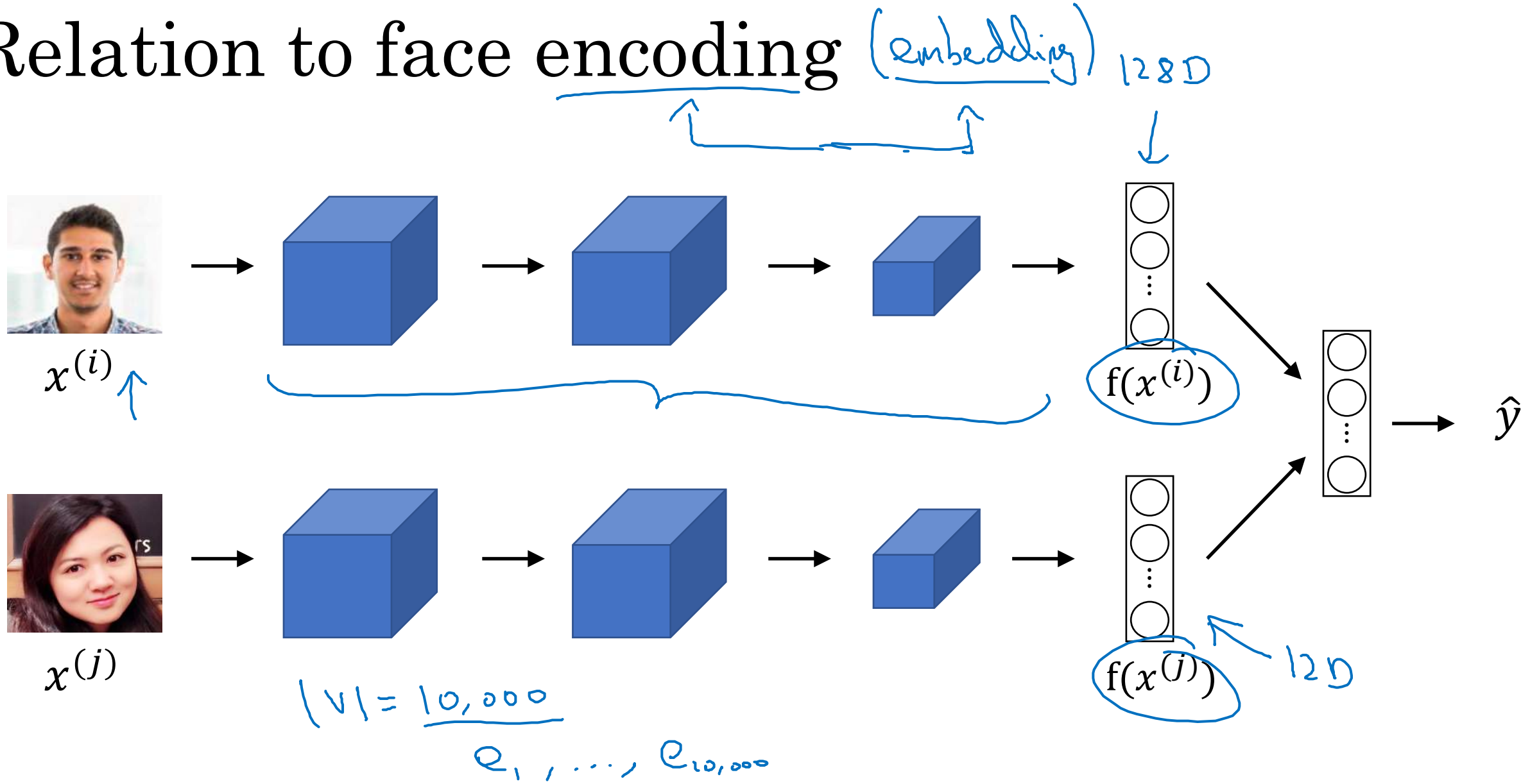
B

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

3. Optional: Continue to finetune the word embeddings with new data.

⟹ Por lo gral, los embedding son claves p/ tasks que no tenemos datasets grandes!

Relation to face encoding



[Taigman et. al., 2014. DeepFace: Closing the gap to human level performance]

deeplearning.ai

NLP and Word Embeddings

Properties of word embeddings

Analogies

	Man (5391)	Woman (9853)	King (4914)	Queen (7157)	Apple (456)	Orange (6257)
Gender	−1	1	-0.95	0.97	0.00	0.01
Royal	0.01	0.02	0.93	0.95	-0.01	0.00
Age	0.03	0.02	0.70	0.69	0.03	-0.02
Food	0.09	0.01	0.02	0.01	0.95	0.97

e_{5391} e_{man} e_{woman}

Man → Woman as King → ? Queen

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

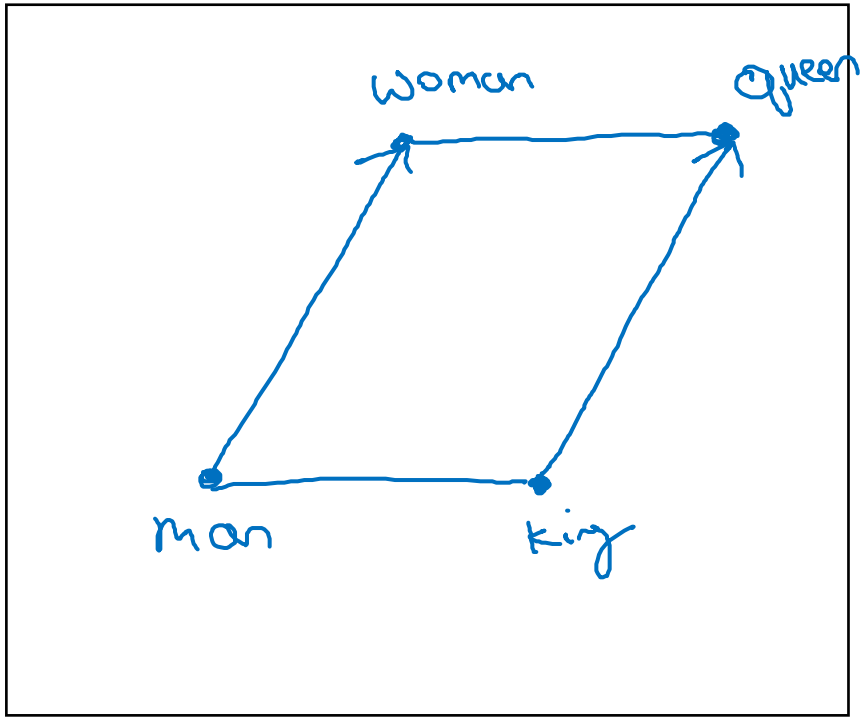
$e_{?} \approx e_{queen}$

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

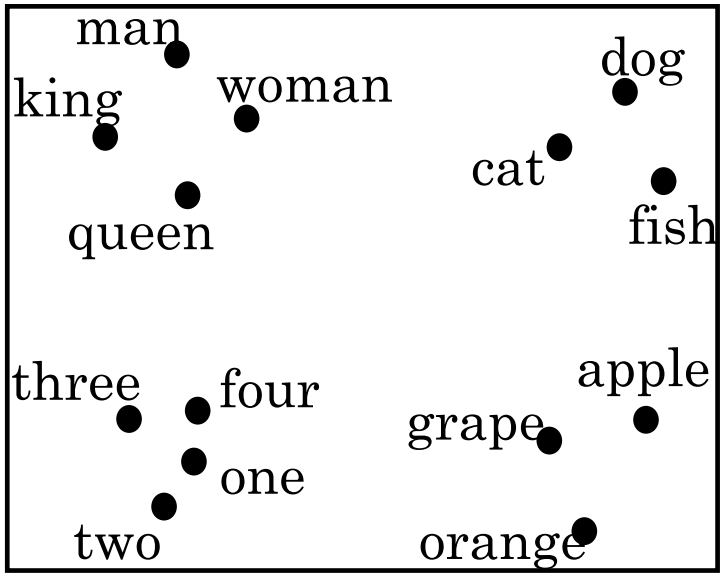
$e_{king} - e_{queen} \approx \begin{bmatrix} -2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

[Mikolov et. al., 2013, Linguistic regularities in continuous space word representations]

Analogies using word vectors



300D



$300 \rightarrow 2D$

t-SNE

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

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

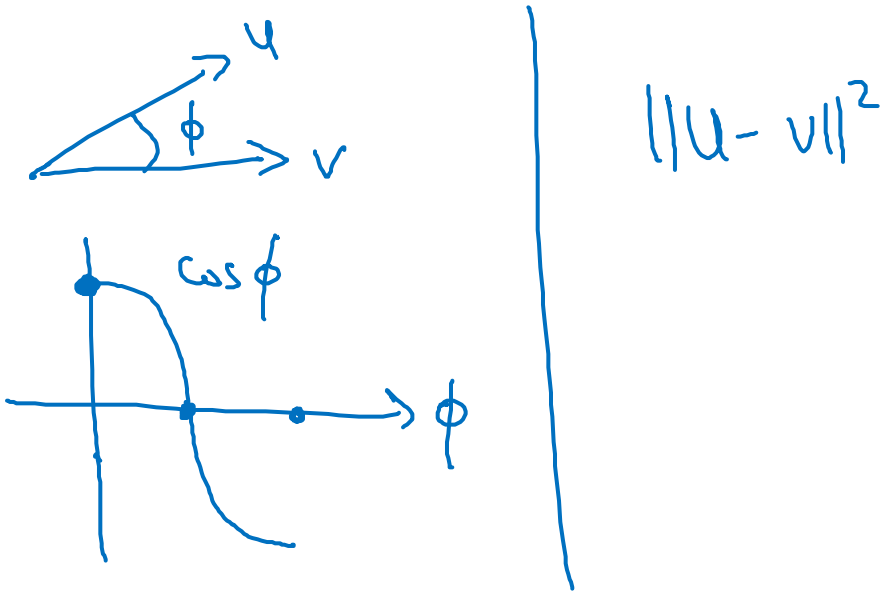
30-75%

Andrew Ng

Cosine similarity

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

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



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

Man:Woman as Boy:Girl

Ottawa:Canada as Nairobi:Kenya

Big:Bigger as Tall:Taller

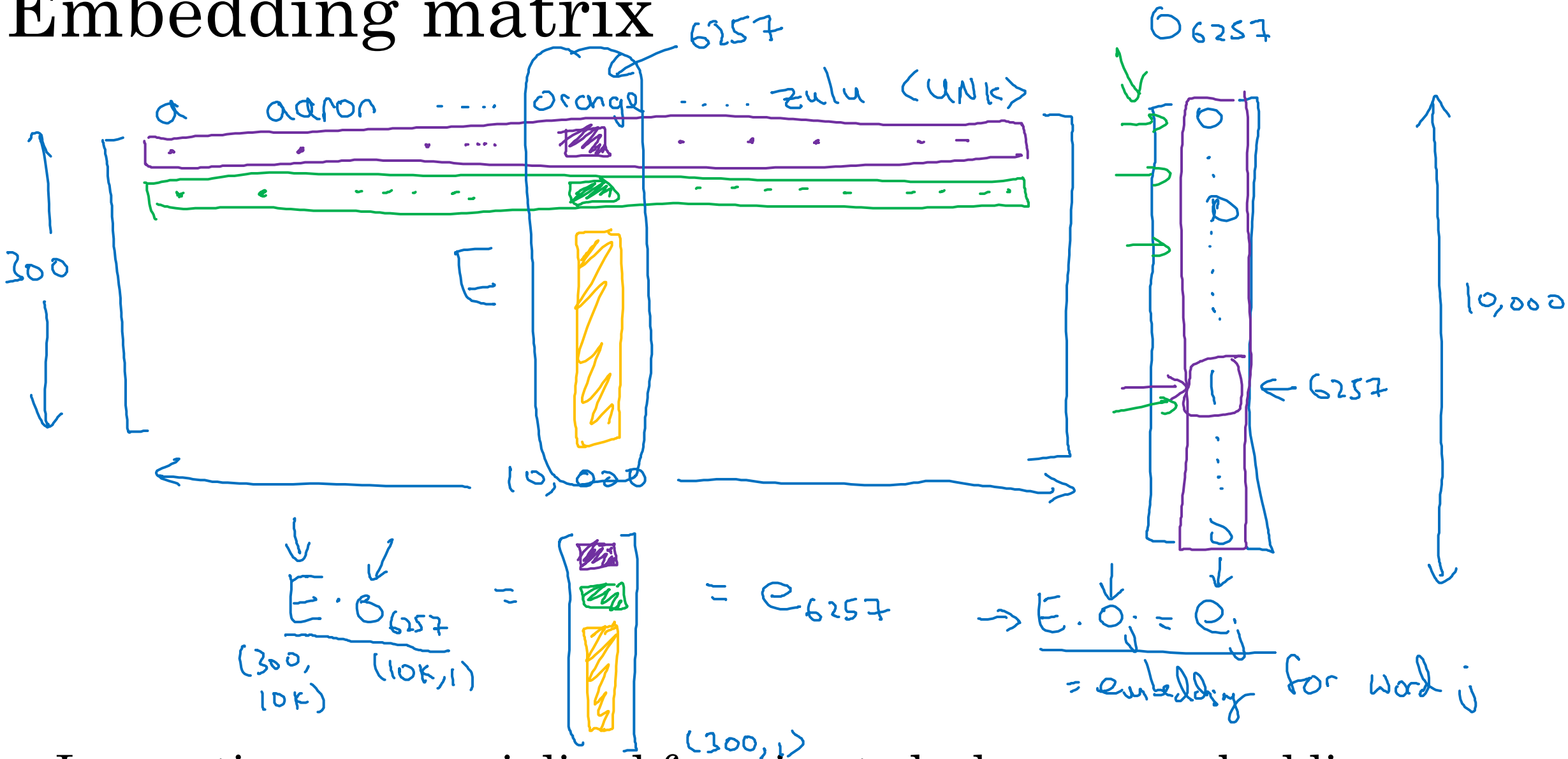
Yen:Japan as Ruble:Russia

deeplearning.ai

NLP and Word Embeddings

Embedding matrix

Embedding matrix



In practice, use specialized function to look up an embedding.

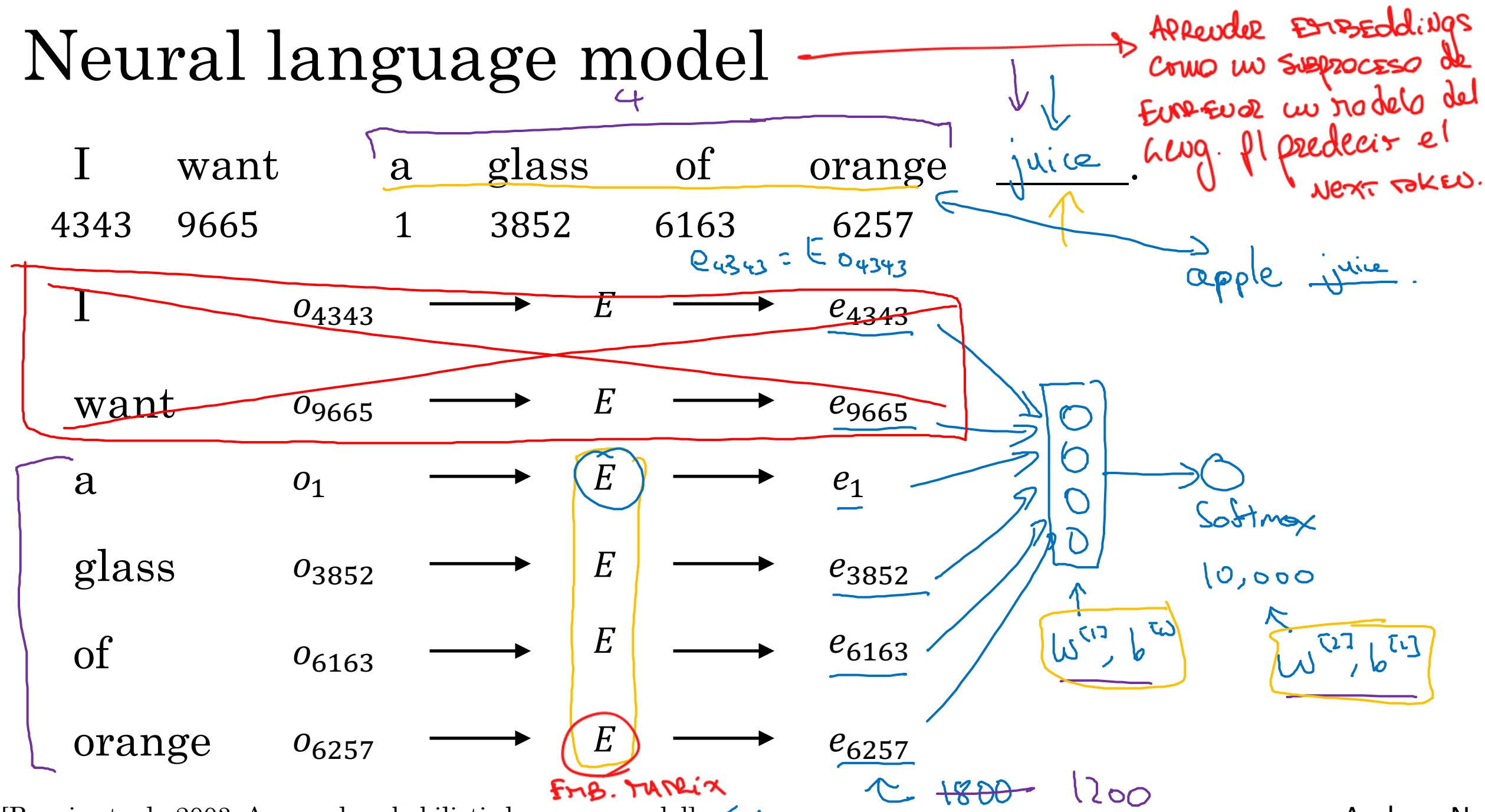
Andrew Ng

deeplearning.ai

NLP and Word Embeddings

Learning word embeddings

Neural language model



[Bengio et. al., 2003, A neural probabilistic language model]

Other context/target pairs

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

context

target

Context: Last 4 words.

- 4 words on left & right
- Last 1 word
- Nearby 1 word

a glass of orange ? to go alg with

orange ?

glass ?

skip gram

deeplearning.ai

NLP and Word Embeddings

Word2Vec

Skip-grams

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

Context	Target
orange	juice
orange	glass
orange	my

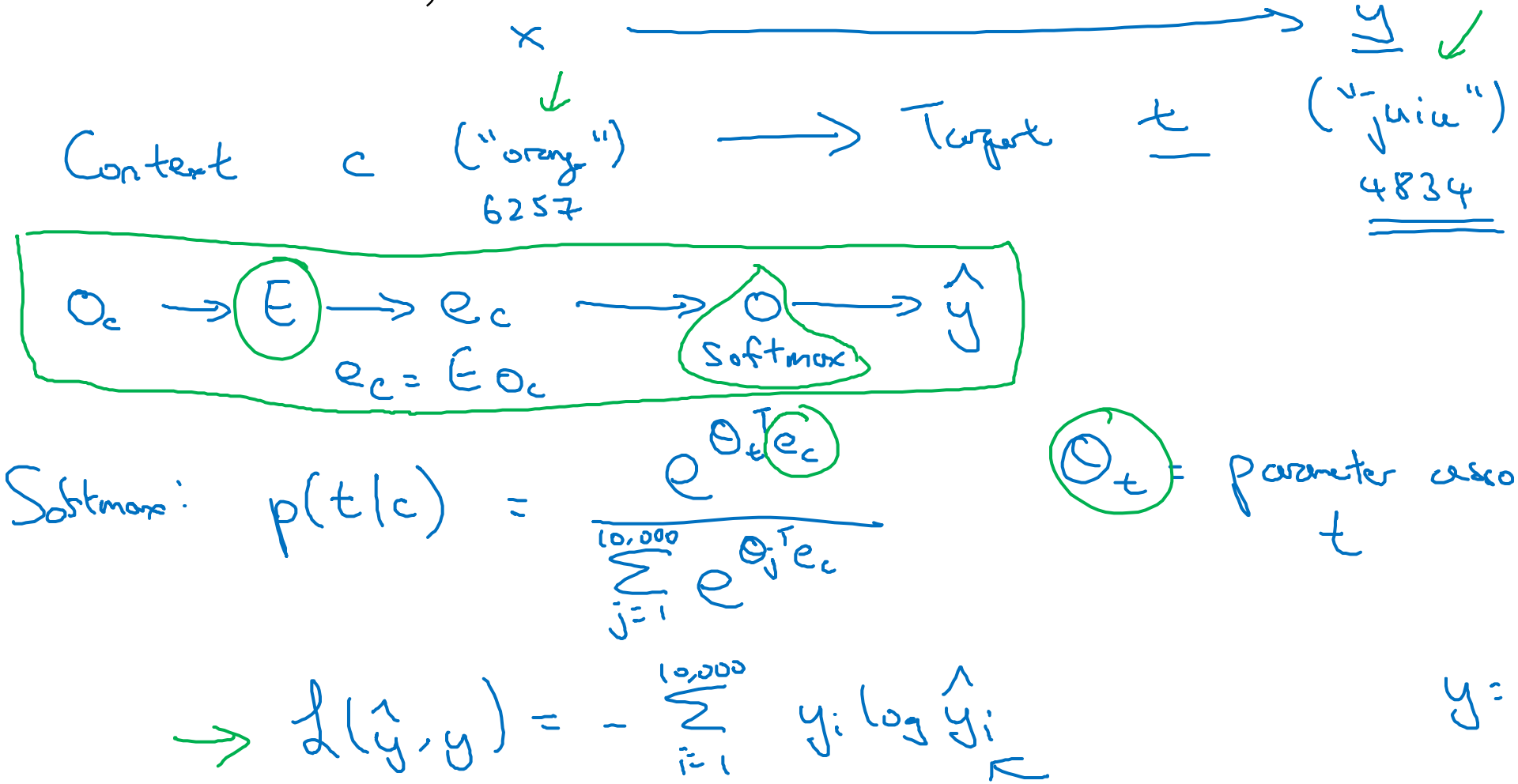
idem. ANTERIOR pero con un CONTEXTO mucho más chico → 1 sola palabra a una distancia SKIPEADA!.

↳ PARA APRENDER embeddings va JOYA.

[Mikolov et. al., 2013. Efficient estimation of word representations in vector space.]

Model

Vocab size = 10,000k

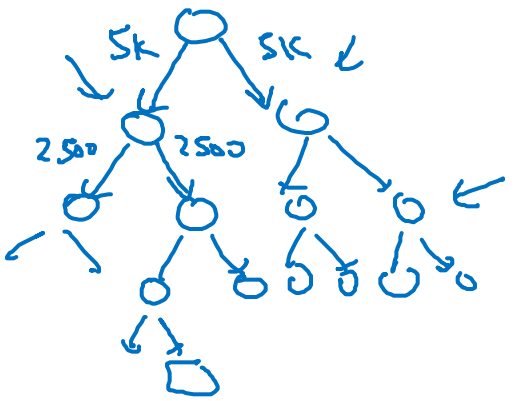


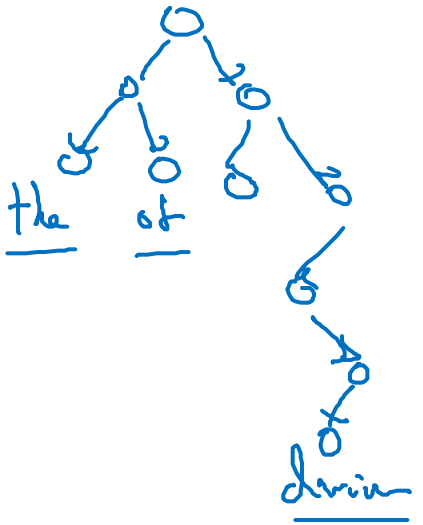
Andrew Ng

Problems with softmax classification

$$p(t|c) = \frac{e^{\theta_t^T e_c}}{\sum_{j=1}^{10,000} e^{\theta_j^T e_c}}$$

Hierarchical softmax.





How to sample the context c?

→ the, of, a, and, to, ...

→ orange, apple, durian

durian

Andrew Ng

deeplearning.ai

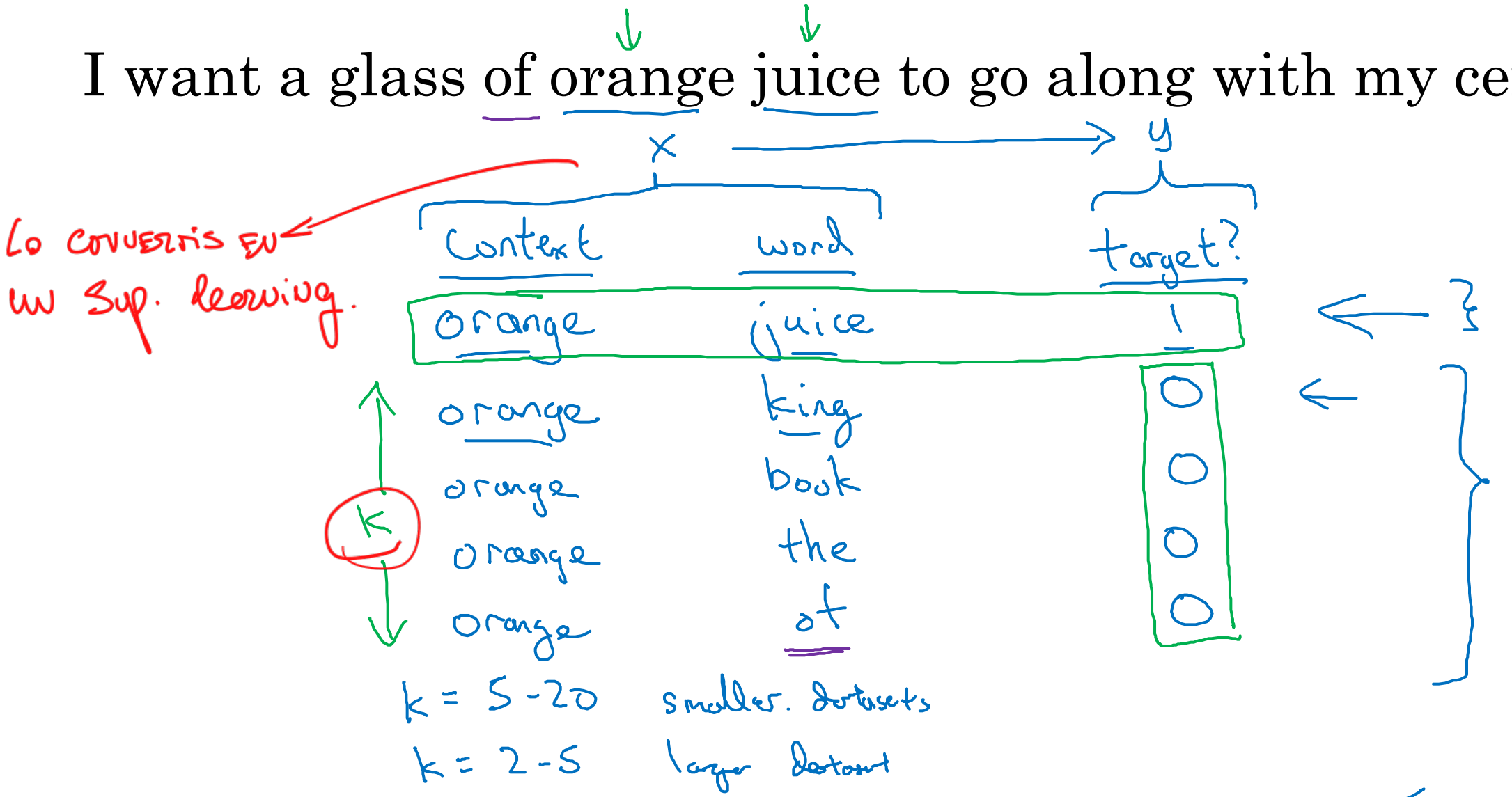
NLP and Word Embeddings

Negative sampling

PARA REDUCIR LA ESCALA DE LA SOFTMAX

Defining a new learning problem

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



[Mikolov et. al., 2013. Distributed representation of words and phrases and their compositionality]

Model

Softmax: $p(t|c) = \frac{e^{\theta_t^T e_c}}{\sum_{j=1}^{10,000} e^{\theta_j^T e_c}}$ — 10,000-way softmax

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

context	word	target?
orange	juice	1
orange	king	0
orange	book	0
orange	the	0
orange	of	0

k

c t y

Orange 6257

$o_{6257} \rightarrow E \rightarrow e_{6257}$

juice? king

10,000

10,000 binary classification problem

k+1

Selecting negative examples

context	word	target?
orange	juice	1
orange	king	0
orange	book	0
orange	the	0
orange	of	0

pos

k neg

$$P(w_i) = \frac{f(w_i)^{3/4}}{\sum_{j=1}^{10,000} f(w_j)^{3/4}}$$

Prob q recomienda el paper pl samplear palabras. de ultima leer del paper.

$\frac{1}{|V|}$

En vez de entrenar una softmax pl 10k, entreno (k+1) clasificadores binarios c/u. Mucho mas barato.

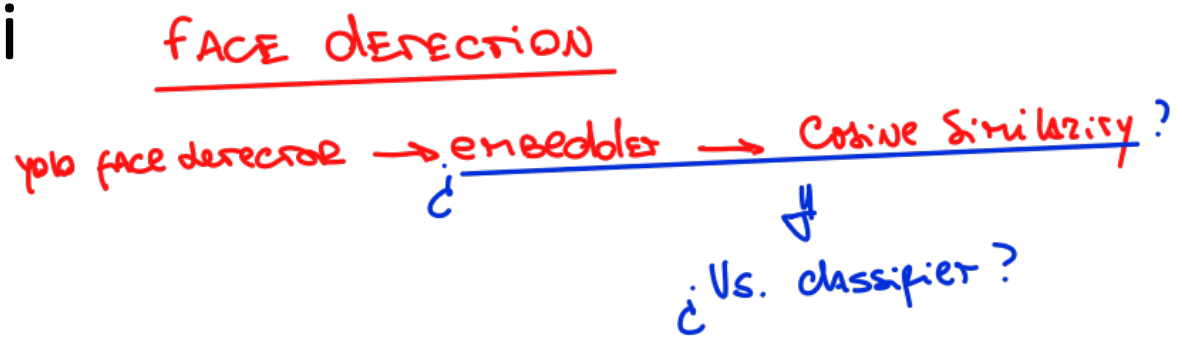
the, of, and, ...

k neg Random y 1 pos posta.

deeplearning.ai

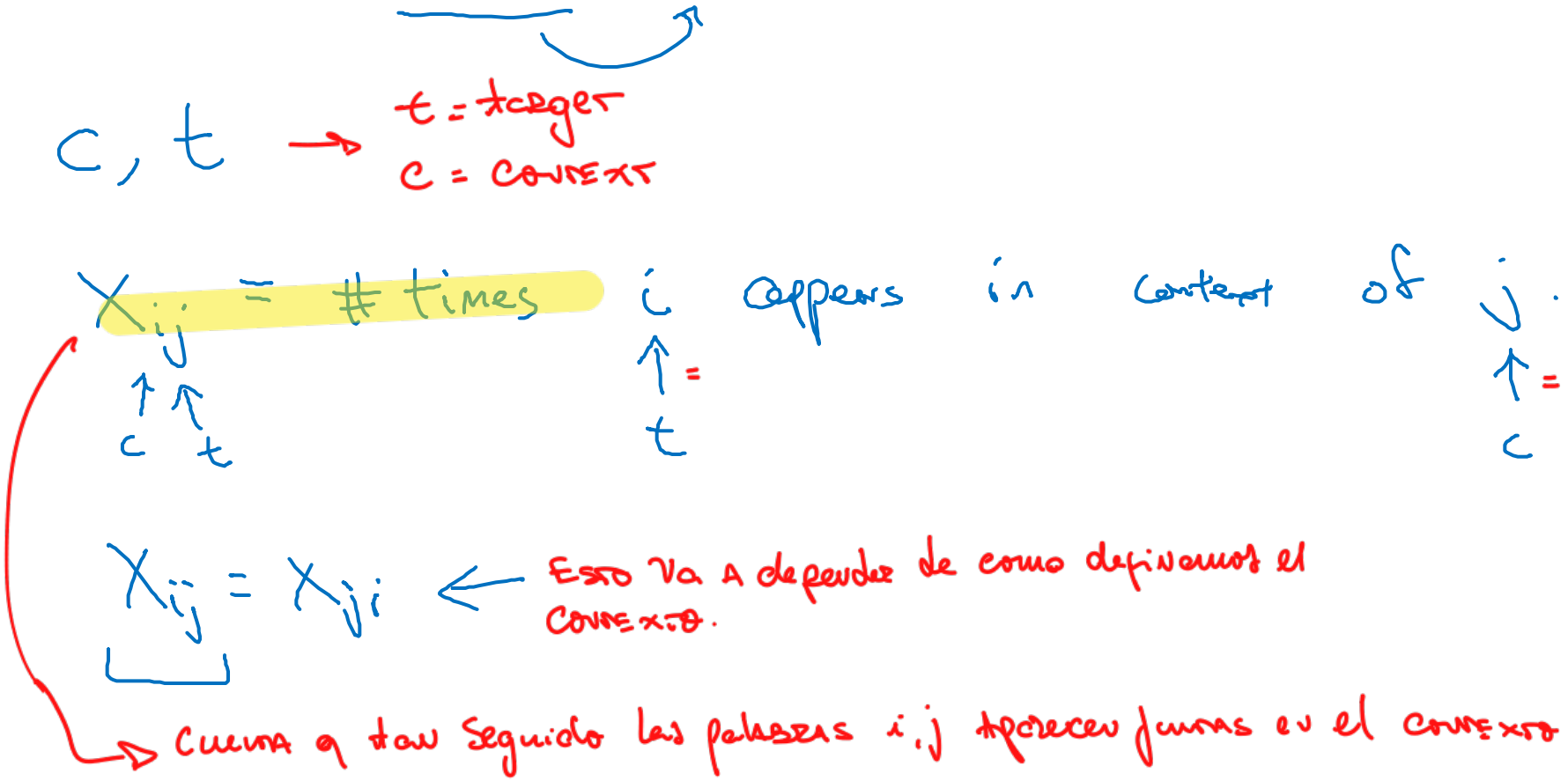
NLP and Word Embeddings

GloVe word vectors



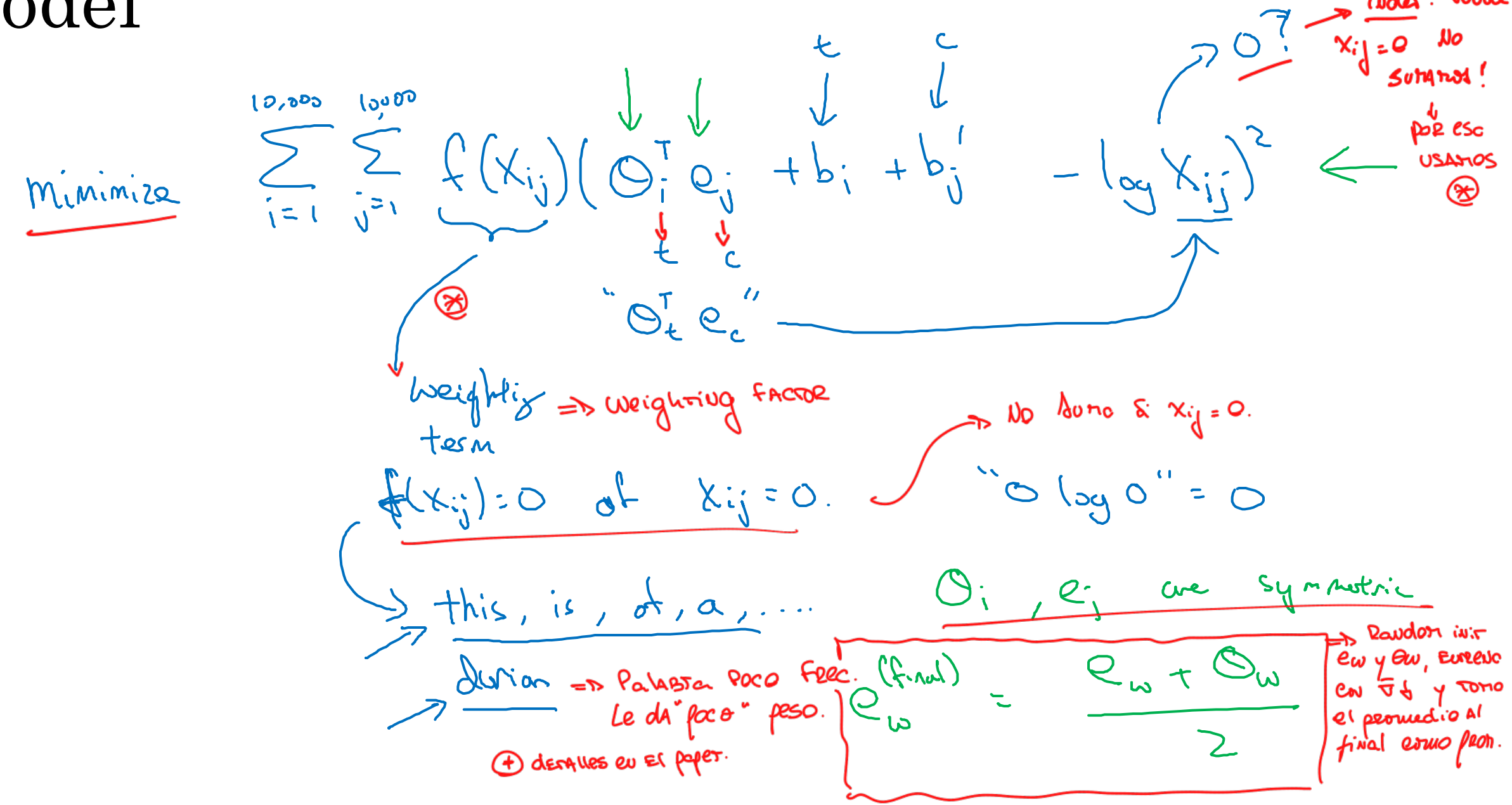
GloVe (global vectors for word representation)

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



[Pennington et. al., 2014. GloVe: Global vectors for word representation]

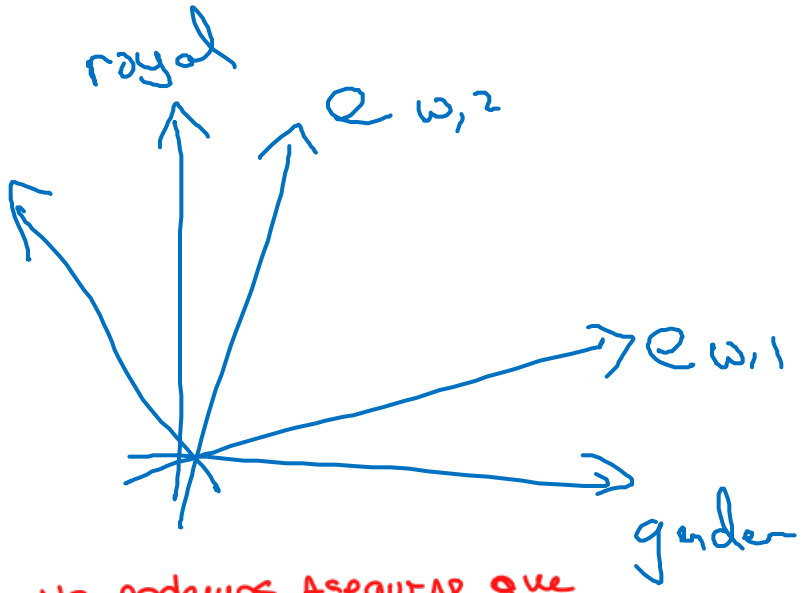
Model



Andrew Ng

A note on the featurization view of word embeddings

	Man (5391)	Woman (9853)	King (4914)	Queen (7157)	
Gender	−1	1	-0.95	0.97	←
Royal	0.01	0.02	0.93	0.95	←
Age	0.03	0.02	0.70	0.69	←
Food	0.09	0.01	0.02	0.01	←



⇒ No podemos asegurar que c/ dirección tenga una interpretabilidad. Sin embargo, las analogías y similaridades se mantienen.

$$\text{minimize} \sum_{i=1}^{10,000} \sum_{j=1}^{10,000} f(X_{ij}) \left(\theta_i^T e_j + b_i - b_j' - \log X_{ij}\right)^2$$

$\hookrightarrow (A\theta_i)^T (A^{-T} e_j) = \theta_i^T A^T A^{-T} e_j$

Andrew Ng

deeplearning.ai

NLP and Word Embeddings

Sentiment classification

Sentiment classification problem

goal: CN Embeddings paperos broader sentiment classifiers CN pocos labels!

x → y

The dessert is excellent. ★★★★☆

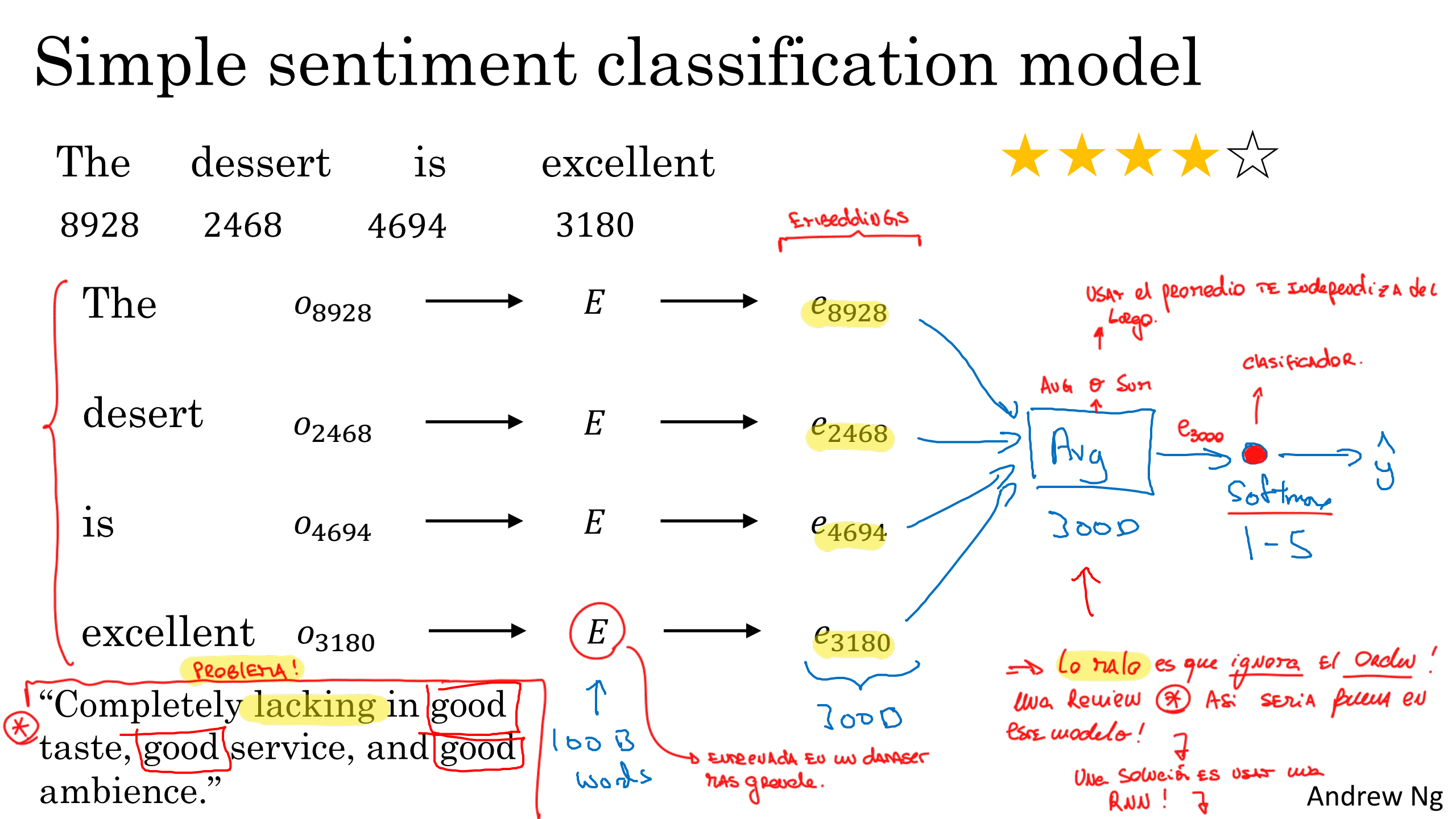
Service was quite slow. ★★☆☆☆

Good for a quick meal, but nothing special. ★★★☆☆

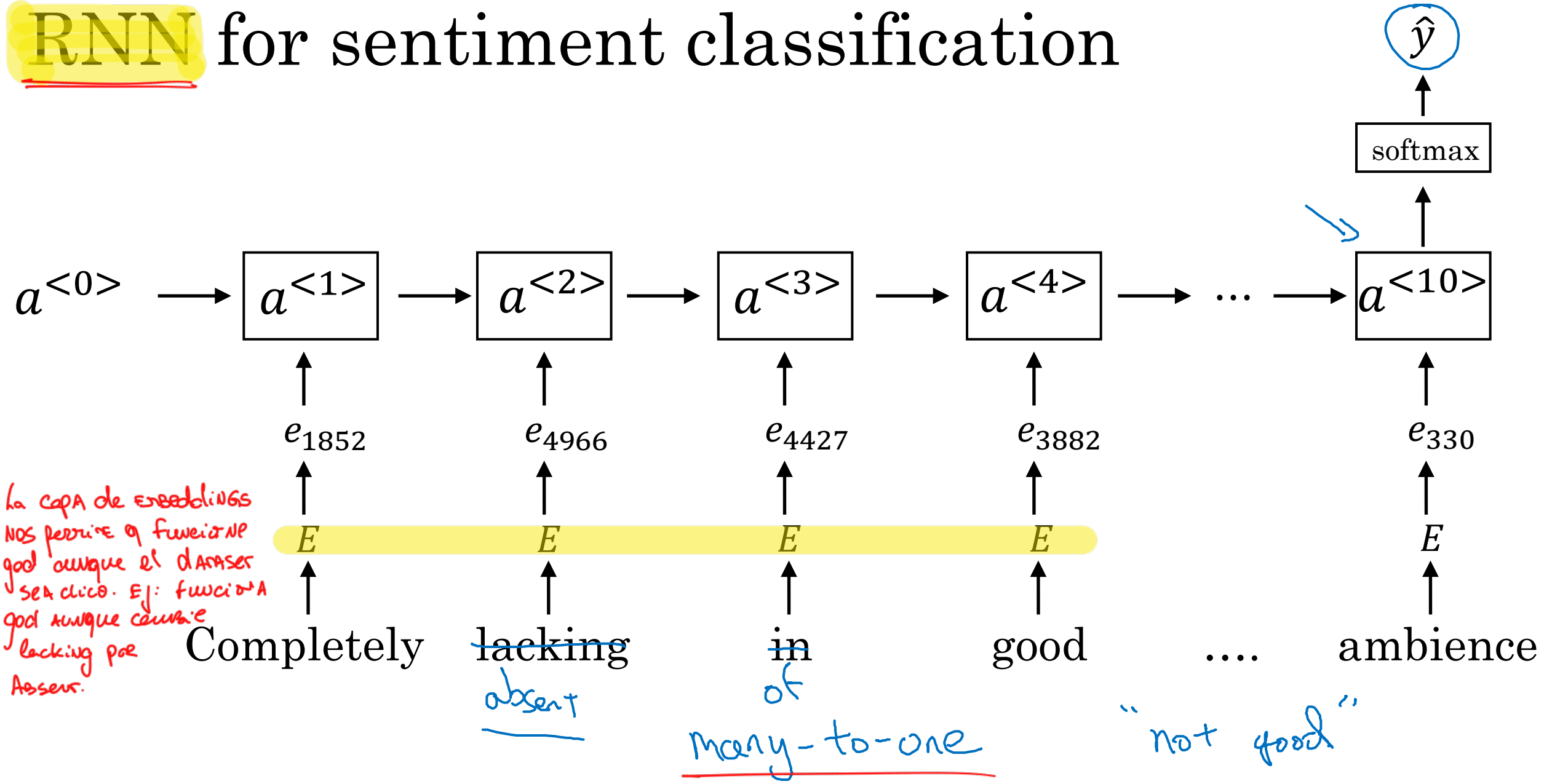
Completely lacking in good taste, good service, and good ambience. ★☆☆☆☆

10,000 → 100,000 words

Simple sentiment classification model



RNN for sentiment classification



La capa de embeddings nos permite q funcione good aunque el dataset sea chico. Ej: funciona good aunque cambie lacking por Absent.

deeplearning.ai

NLP and Word Embeddings

Debiasing word embeddings

The problem of bias in word embeddings

gender, race, etc.

Man:Woman as King:Queen

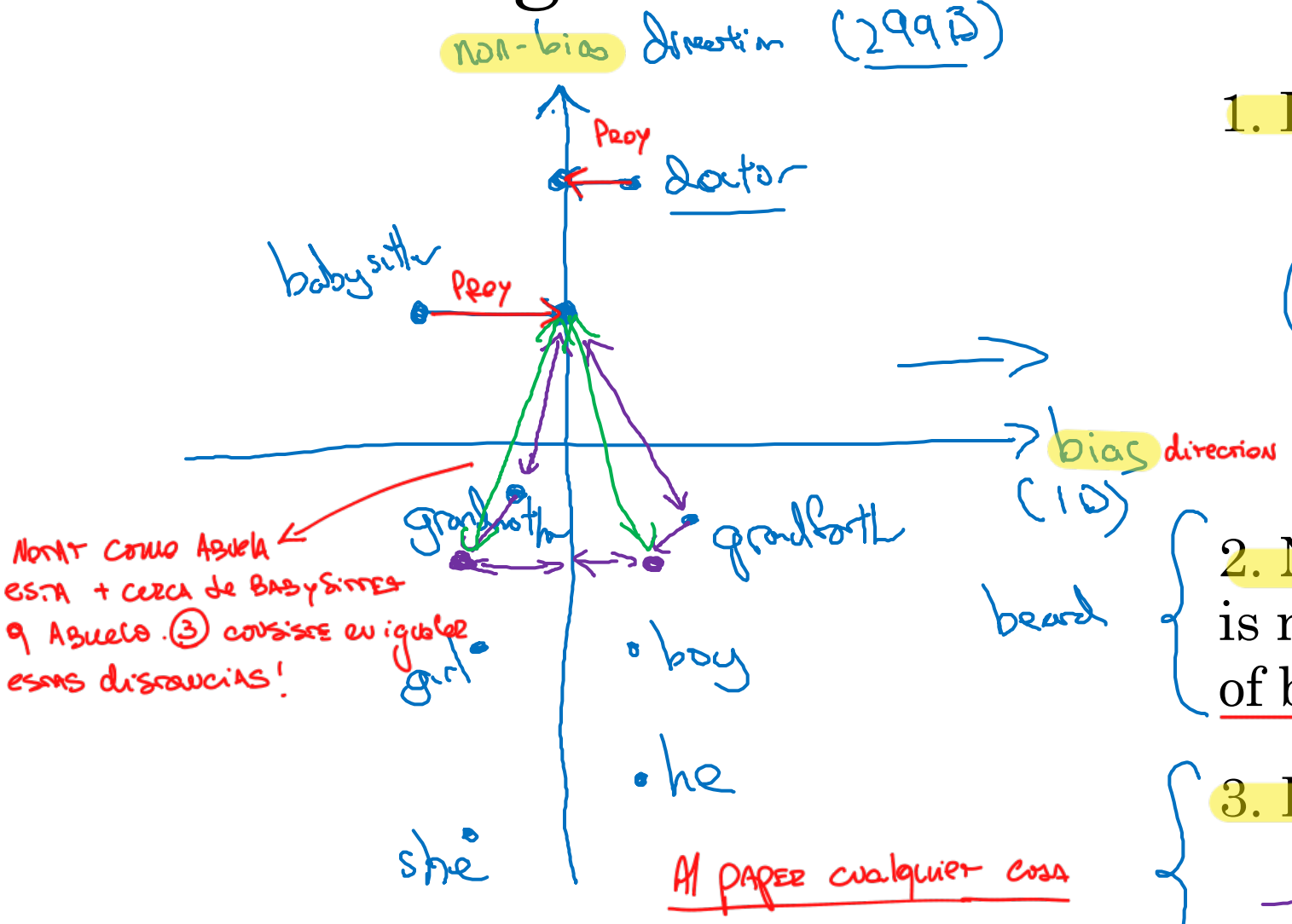
Man:Computer_Programmer as Woman:Homemaker ✗ !

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.

[Bolukbasi et. al., 2016. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings]

Addressing bias in word embeddings → GOD



1. Identify bias direction.

$e_{he} - e_{she}$

$e_{male} - e_{female}$

⋮

average

(*) PALABRAS q NO TIENEN EMBEBIDO el género. (≠ ABUELA, ABUELO xq)

2. Neutralize: For every word that is not definitional, project to get rid of bias. (*)

3. Equalize pairs. (3) CON ALGEBRA BASICALLY

→ grandmother ↔ grandfather

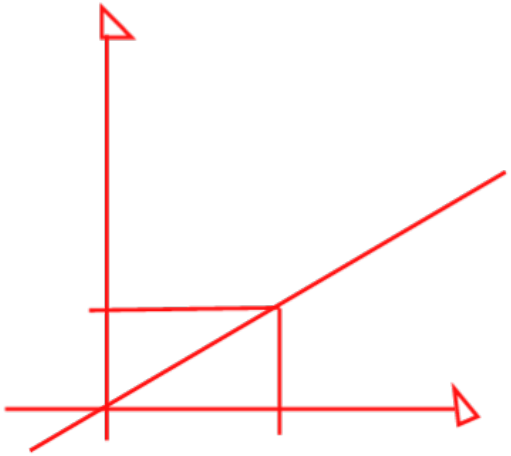
girl ↔ boy

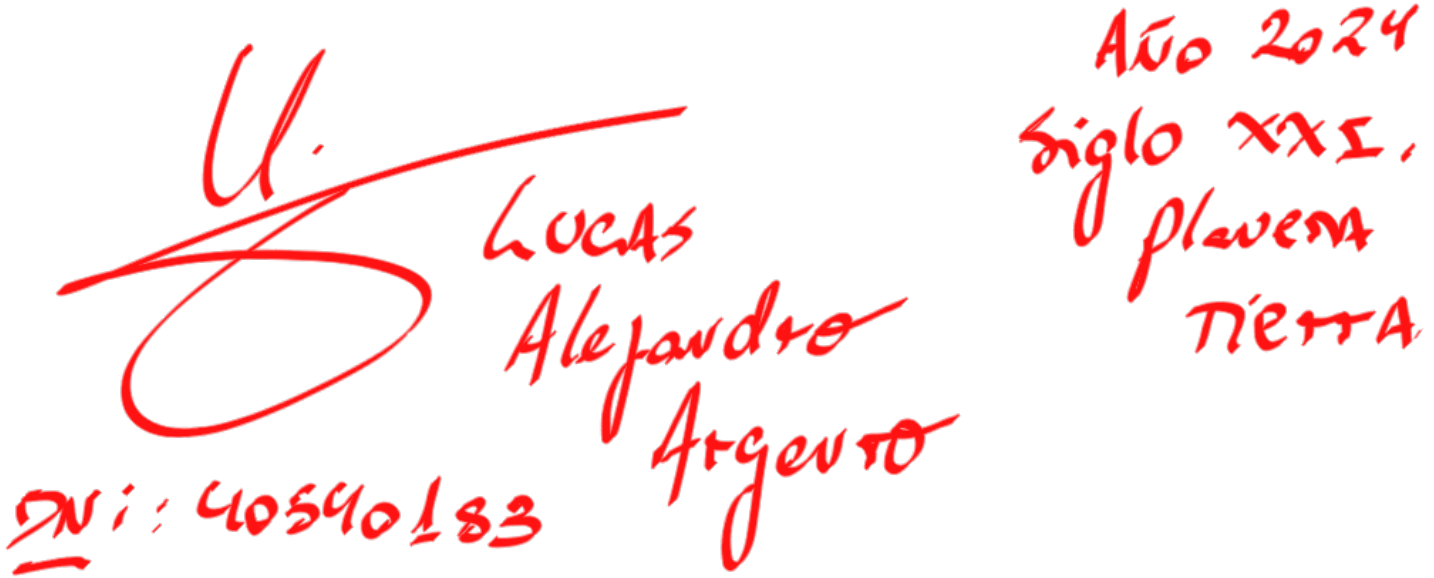
God

[Bolukbasi et. al., 2016. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings]

Andrew Ng

Copyright Notice





deeplearning.ai

Sequence to sequence models

¡Translation!

Basic models

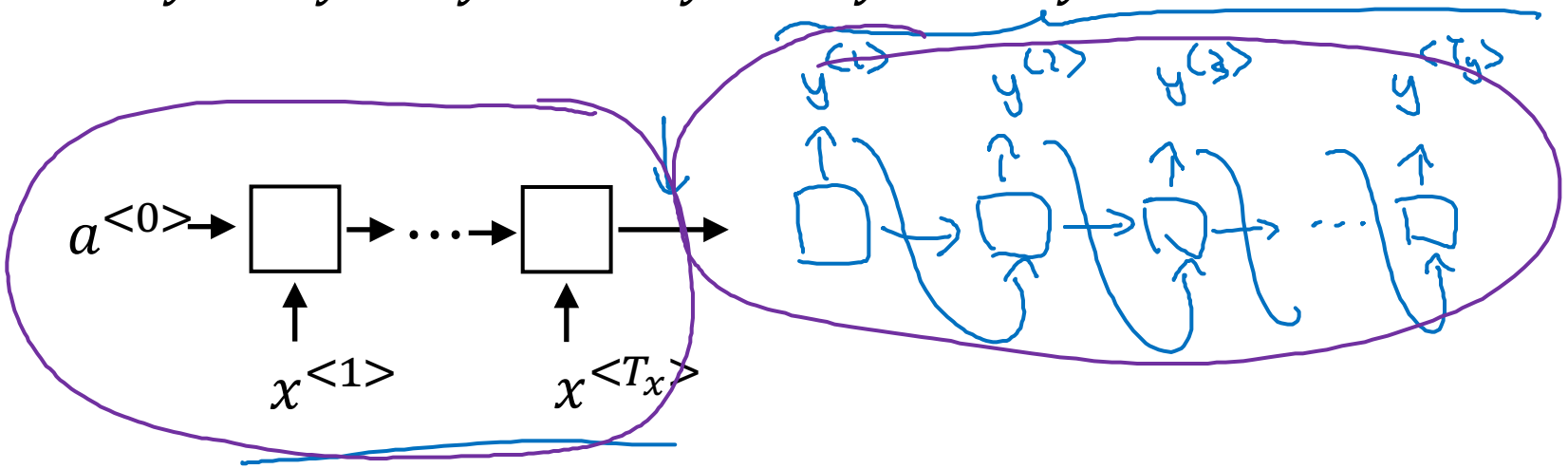
Sequence to sequence model

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

Jane visite l'Afrique en septembre

⟶ Jane is visiting Africa in September.

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

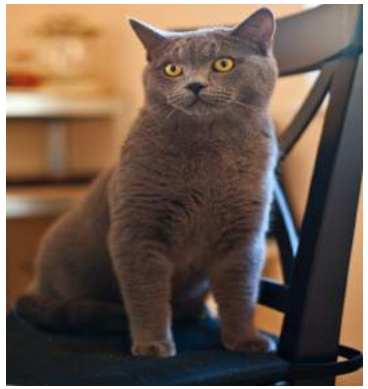


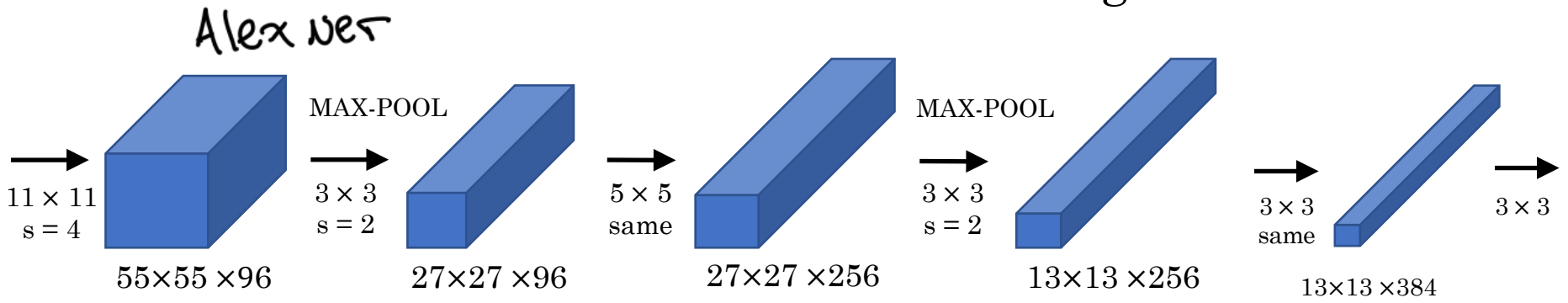
[Sutskever et al., 2014. Sequence to sequence learning with neural networks]

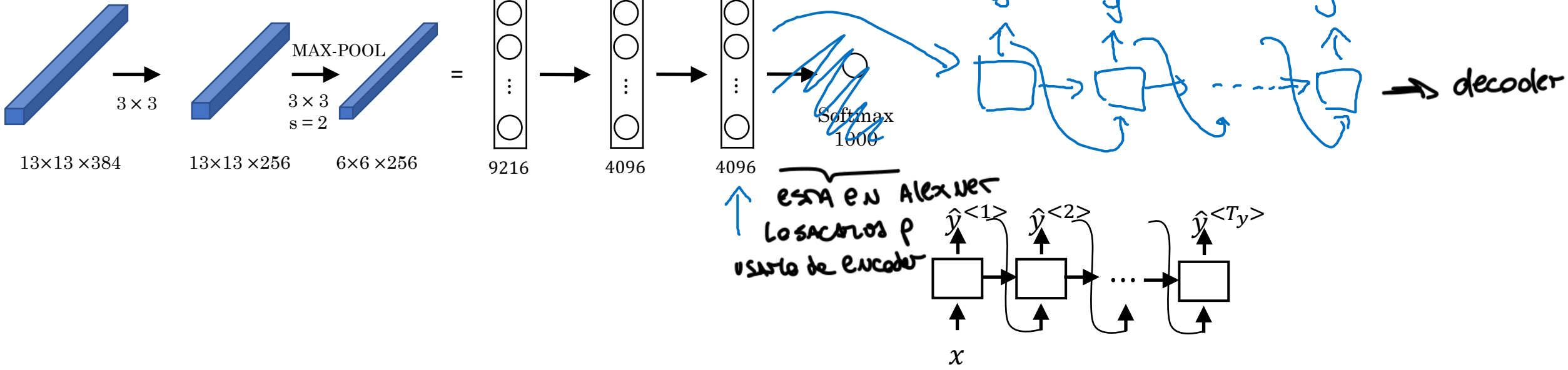
[Cho et al., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation]

Image captioning

$y^{<1>}$ $y^{<2>}$ $y^{<3>}$ $y^{<4>}$ $y^{<5>}$ $y^{<6>}$
A cat sitting on a chair







[Mao et. al., 2014. Deep captioning with multimodal recurrent neural networks]
[Vinyals et. al., 2014. Show and tell: Neural image caption generator]
[Karpathy and Li, 2015. Deep visual-semantic alignments for generating image descriptions]

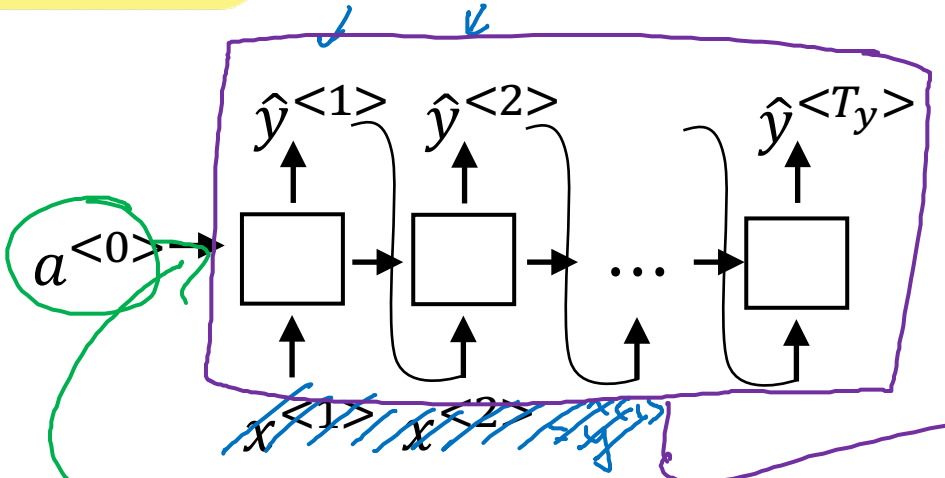
deeplearning.ai

Sequence to sequence models

Picking the most likely sentence

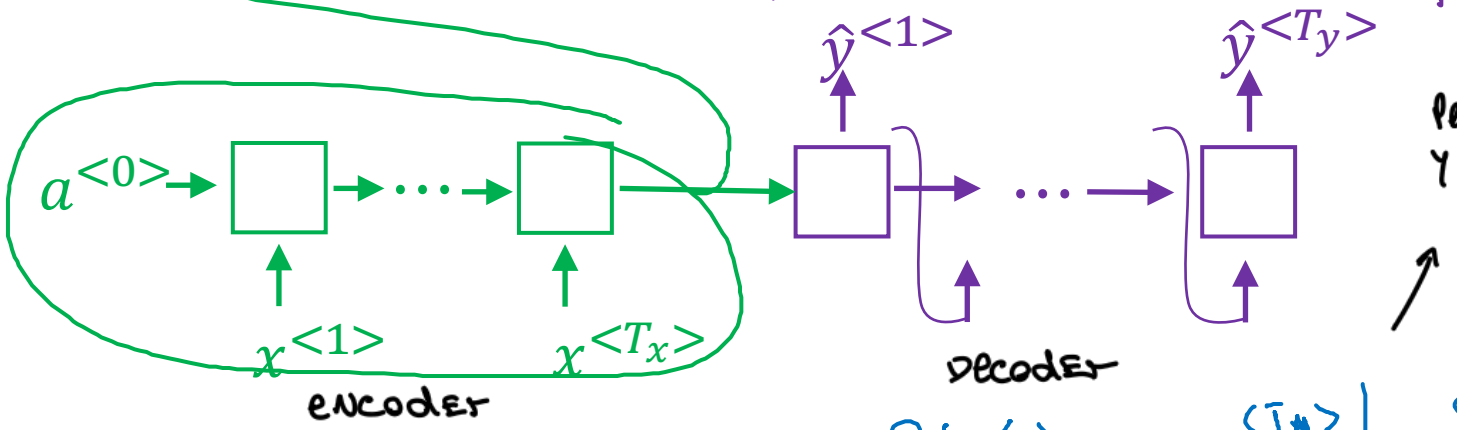
Machine translation as building a conditional language model

Language model:



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

Machine translation:



Prob de la traduction y dado x

"Conditional language model"

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

Finding the most likely translation

Jane visite l'Afrique en septembre.

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

English

French

→ Jane is visiting Africa in September.

→ Jane is going to be visiting Africa in September.

→ In September, Jane will visit Africa.

→ Her African friend welcomed Jane in September.

outputs del modelo
Cuál elegimos y
Cómo?

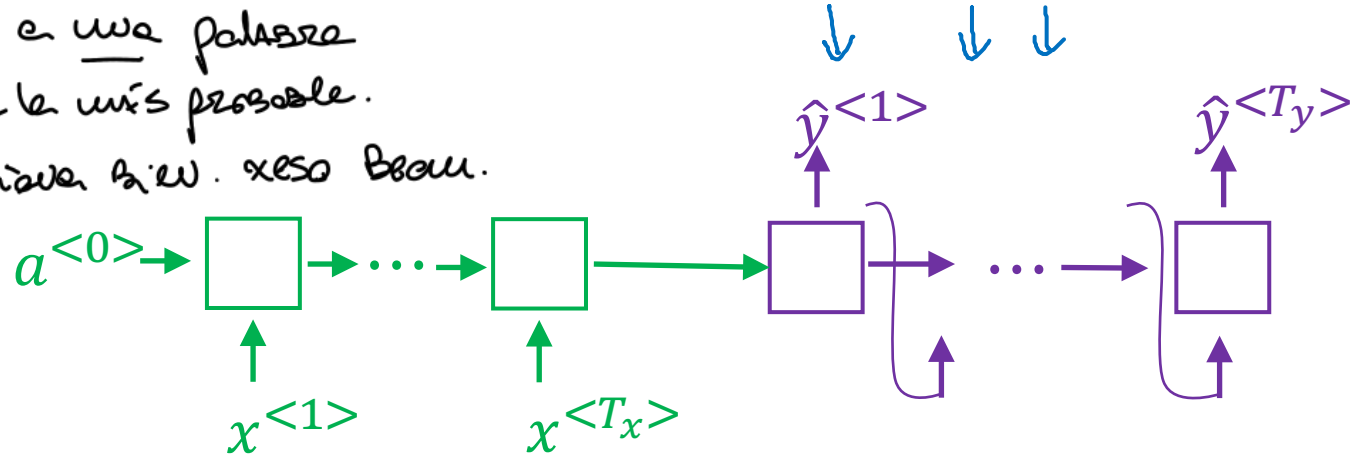
$$\arg\max_{y^{<1>},\ldots,y^{<T_y>}} P(y^{<1>}, \ldots, y^{<T_y>} \mid x)$$

⇒ queremos maximizar la prob. Buscamos ese y.
Se suele hacer con Beam Search

Why not a greedy search?

$p(\hat{y}^{<1>} | x)$

Vas eligiendo de a una palabra eligiendo siempre la más probable. pero esto no funciona bien. Beam.



$\arg\max_y P(\hat{y}^{<1>}, \hat{y}^{<2>}, \ldots, \hat{y}^{<T_y>} | x)$

10,000

10

$10{,}000^{10}$

$P(y|x)$

→ Jane is visiting Africa in September.

→ Jane is going to be visiting Africa in September.

$P(\text{Jane is going} | x) > P(\text{Jane is visiting} | x)$

Andrew Ng

deeplearning.ai

Sequence to sequence models

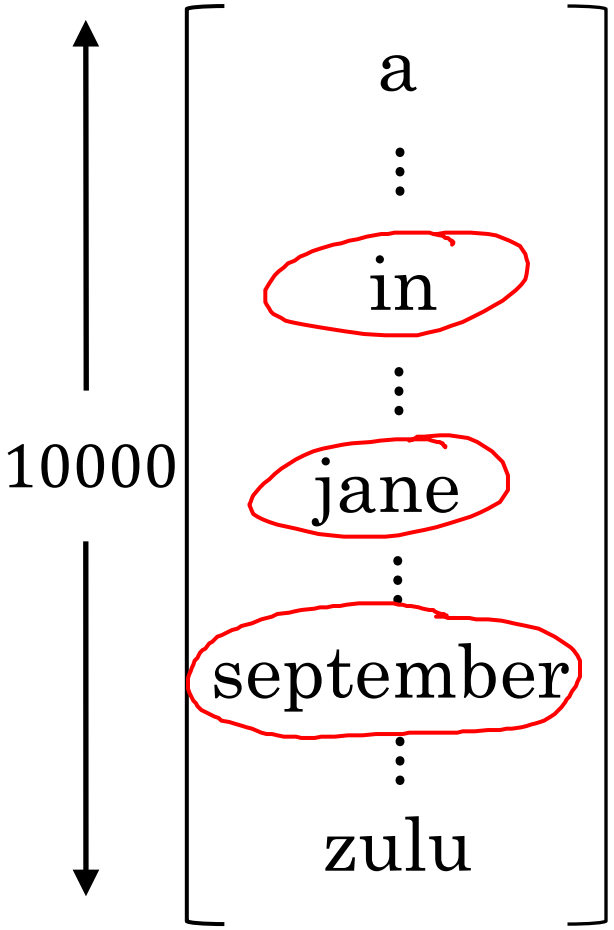
Beam search

Beam search algorithm

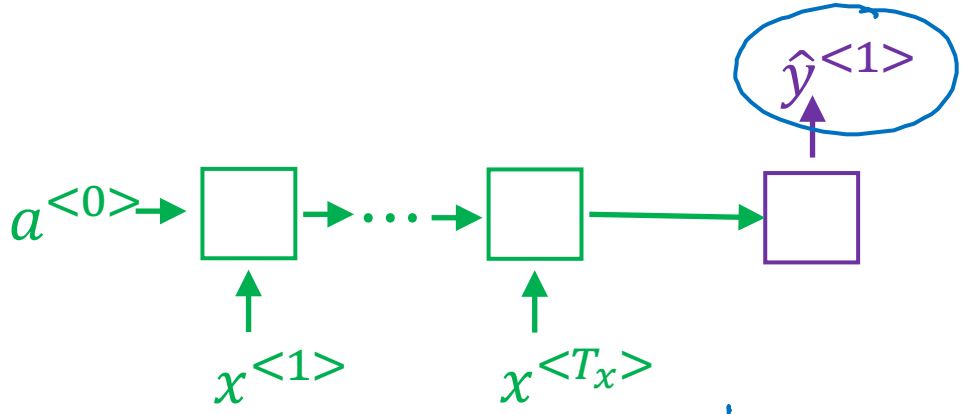
$B = 3$ (beam width)

Cantidad de palabras que evaluar como most likely output.

Step 1



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



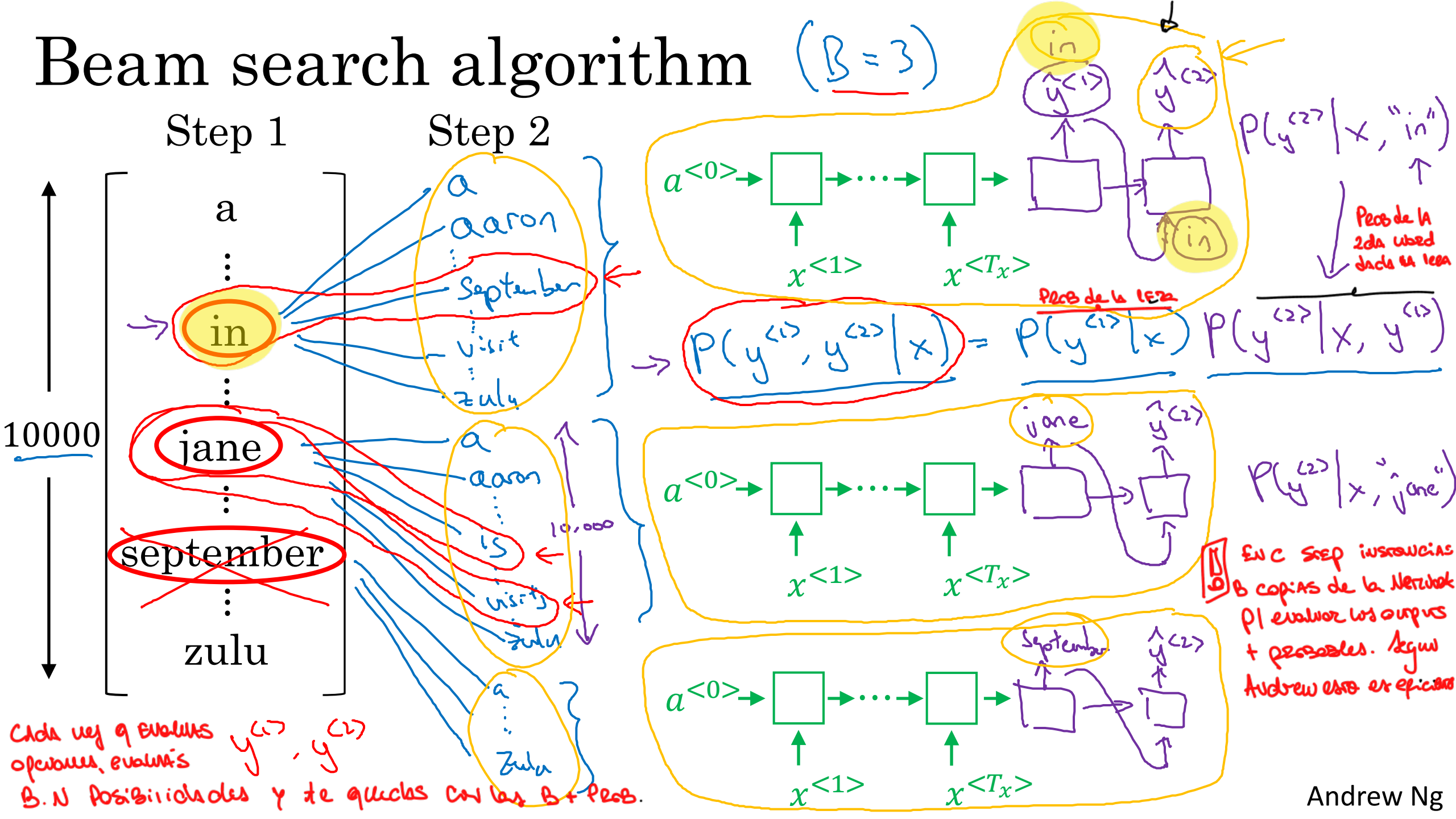
French sentence

N possible outputs in English pero me quedo con los top B

DESPUES DE ESO TE FIJAS CUÁL ES LA SIGUIENTE palabra + prob.

ESO LO HACES CON UNA NN

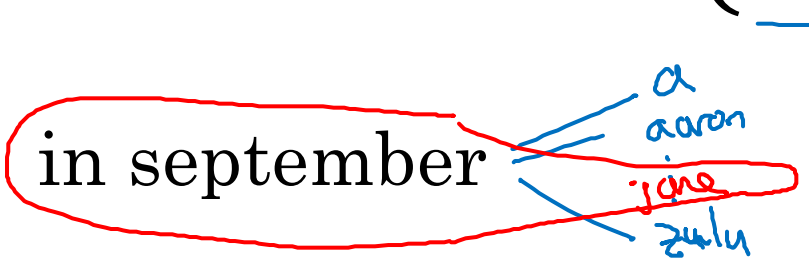
Beam search algorithm

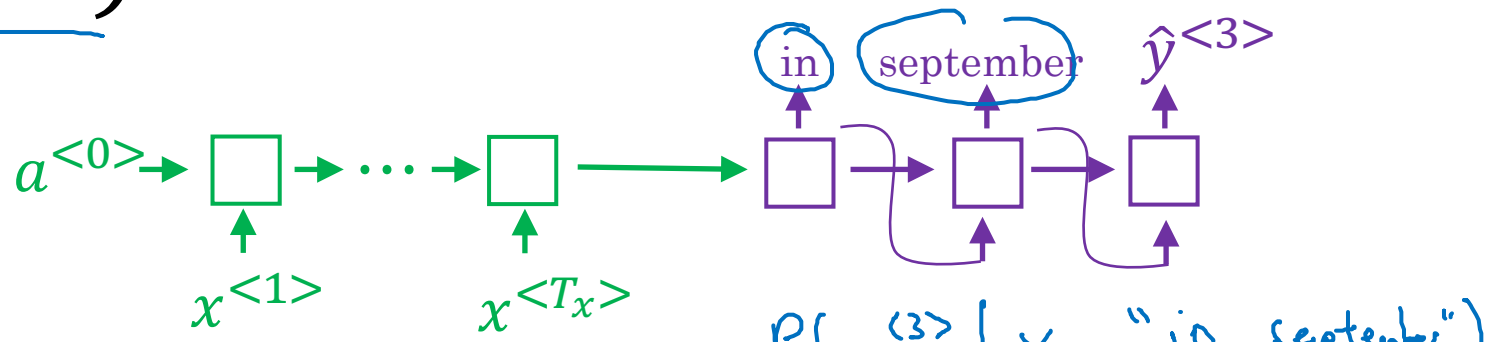


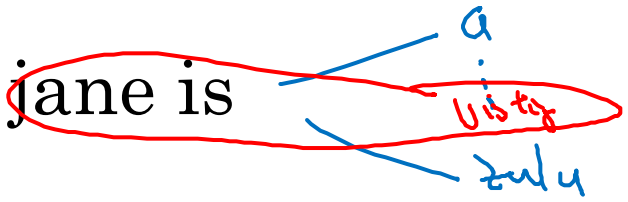
Andrew Ng

Beam search ($B = 3$)

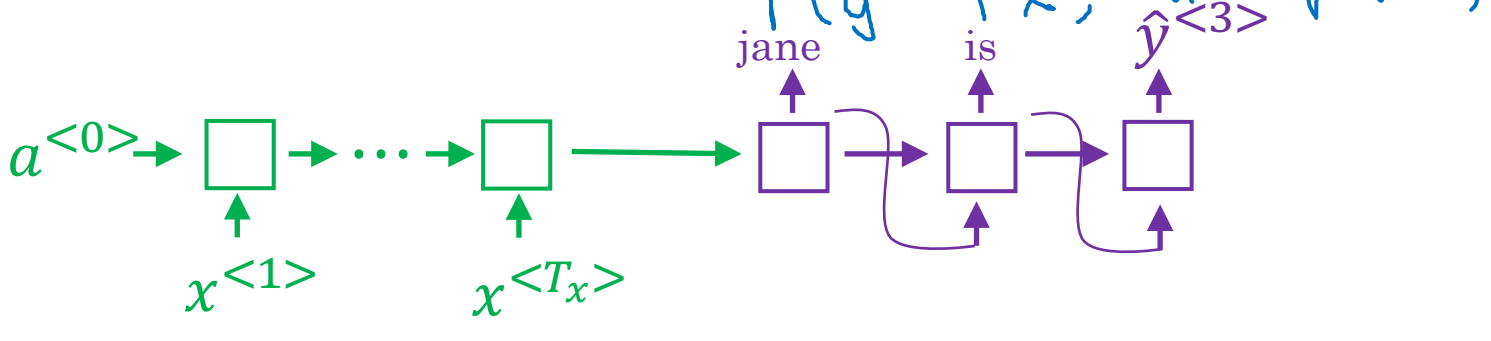
$B = 1$ ⇝ greedy search

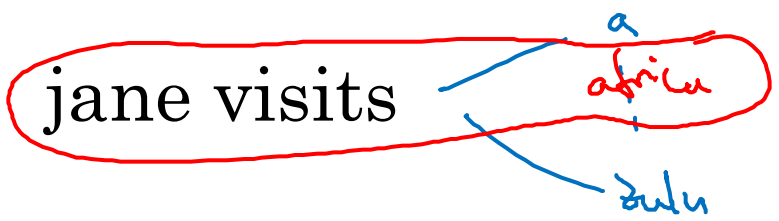


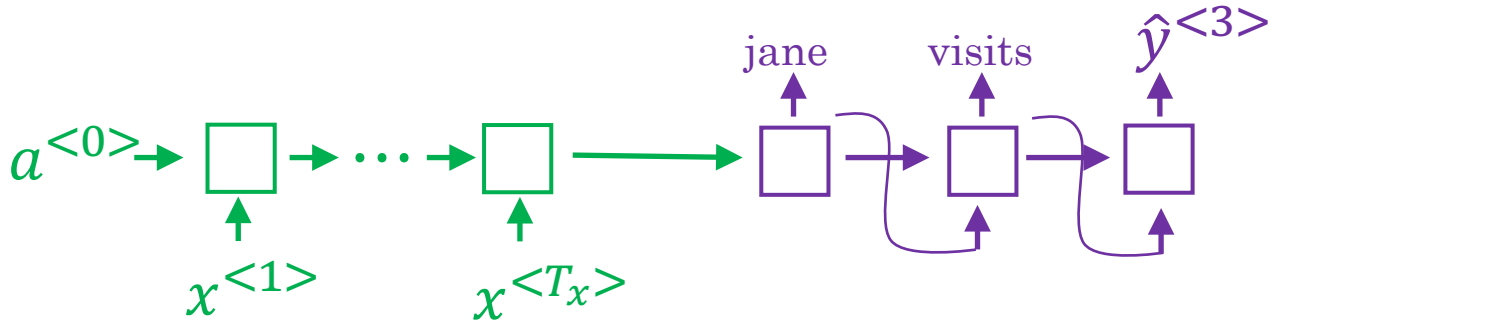




$P(y^{<3>} | x, \text{"in september"})$







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

jane visits africa in september. <EOS>

Andrew Ng

deeplearning.ai

Sequence to sequence models

Refinements to beam search

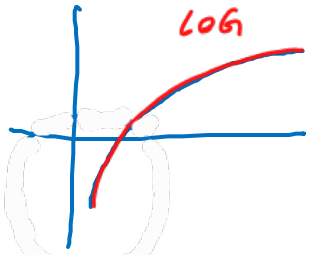
Length normalization

$$p(y^{<1>} \dots y^{<T_y>} | x) = p(y^{<1>}|x)\, p(y^{<2>}|x, y^{<1>}) \cdots$$
$$p(y^{<T_y>}|x, y^{<1>}, \dots, y^{<T_y-1>})$$

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

Puede haber underflow xq son probs muy chiquitas π. xeso tomamos log

log



$$\log P(y|x) \overset{\downarrow}{=} P(y|x)$$

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

← esto! si tengo una oración muy larga, estoy multiplicando muchos números < 1 y eso me da un producto (prob) muy chico. Lo mismo es cierto con el log (de valor + negativo). Para arreglarlo:

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

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

Elegís la sentence con el value + grande de esa de todas las q viste en beam search

$\alpha = 0.7$ hyperparam para controlar esto →

- $\alpha = 1$ → Normalizar full x length
- $\alpha = 0$ → sin normaliz

Andrew Ng

Beam search discussion

Large B: better result, slower

Small B: worse result, faster

Beam width B?

1 -> 3 -> 10, 100, 1000, -> 3000

Unlike exact search algorithms like BFS (Breadth First Search) or DFS (Depth First Search), Beam Search runs faster but is not guaranteed to find exact maximum for $\arg\max_y P(y|x)$.

deeplearning.ai

Sequence to sequence models

Error analysis on beam search

What if Beam is causing problems? How do you know its this?

Example

Componentes del modelo: RNN, Beam Search

CAS: siempre funciona pero a veces no es suf. → BT

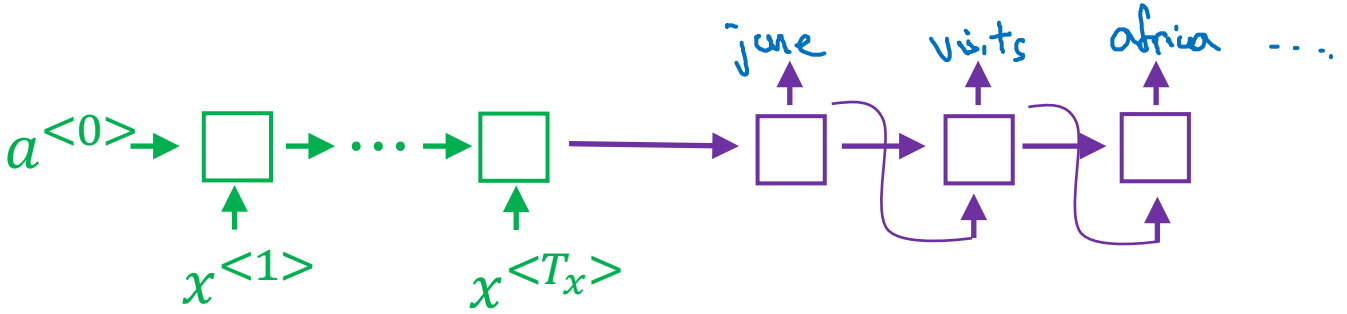
Jane visite l'Afrique en septembre.

⇓

Human: Jane visits Africa in September. (y^*)

Algorithm: Jane visited Africa last September. $(\hat{y})$ ⇐

RNN computes: $P(y^*|x) \lessgtr P(\hat{y}|x)$



Error analysis on beam search

Human: Jane visits Africa in September. (y^*)

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

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

Beam search chose $\hat{y}$. But y^* attains higher $P(y|x)$.

$\arg\max_y P(y|x)$

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

Conclusion: RNN model is at fault.

Andrew Ng

Error analysis process

Human	Algorithm	$P(y^*\|x)$	$P(\hat{y}\|x)$	At fault?
Jane visits Africa in September.	Jane visited Africa last September.	2×10^{-10}	1×10^{-10}	B
				R
				B
				R
				R
				⋮

Figures out what faction of errors are "due to" beam search vs. RNN model

Andrew Ng

deeplearning.ai

Sequence to sequence models

Bleu score (optional)

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.

MT output: the the the the the the the. (7)

Precision: $\frac{7}{7}$

de las 7 palabras, cuántas aparecen en el test?

No nos sirve

Modified precision: $\frac{2}{7}$

max times q aparece en un ejemplo

count de la palabra the

BLEU: Lo usamos para medir ACCURACY cuando hay más de una 2da correcta

Bilingual Evaluation Understudy

→ Intuition: Nos fijamos si las words que en el MT aparecen en el dev set correcto.

[Papineni et. al., 2002. Bleu: A method for automatic evaluation of machine translation]

Bleu score on bigrams

→ calcular pl más de una palabra

Example:
- Reference 1: The cat is on the mat.
- Reference 2: There is a cat on the mat.
- MT output: The cat the cat on the mat.

	Count en MT	Count clip
the cat	2	1
cat the	1	0
cat on	1	1
on the	1	1
the mat	1	1

Count clip → the Nº of times the bigram appears in at least one ref.

$$\frac{4}{6}$$

4 → Σ Count clip

6 → Σ Count MT

Modified precision on bigrams.

[Papineni et. al., 2002. Bleu: A method for automatic evaluation of machine translation]

Bleu score on unigrams

Example: Reference 1: The cat is on the mat.

Reference 2: There is a cat on the mat.

→ MT output: The cat the cat on the mat. ($\hat{y}$)

$p_1, p_2, \ldots = 1.0$

UNIGRAMS

$$p_1 = \frac{\sum_{\text{unigrams} \in \hat{y}} \text{Count clip}(\text{unigram})}{\sum_{\text{unigram} \in \hat{y}} \text{Count}(\text{unigram})}$$

NGRAMS

$$p_N = \frac{\sum_{\text{Ngrams} \in \hat{y}} \text{Count clip}(\text{Ngram})}{\sum_{\text{Ngrams} \in \hat{y}} \text{Count}(\text{Ngram})}$$

[Papineni et. al., 2002. Bleu: A method for automatic evaluation of machine translation]

Bleu details

p_n = Bleu score on n-grams only

Combined Bleu score: $\text{Bleu} = BP \cdot \exp\left(\frac{1}{4}\sum_{n=1}^{4} p_n\right)$

BP = BREVITY PENALTY → EVITA TRADUCCIONES MUY CORTAS

$$BP = \begin{cases} 1 & \text{if MT\_output\_length} > \text{reference\_output\_length} \\ \exp(1 - \text{MT\_output\_length}/\text{reference\_output\_length}) & \text{otherwise} \end{cases}$$

BÁSICAMENTE EL BLEU ES CLAVE PI CUANDO HAY MÚLTIPLES RSAS CORRECTAS.

[Papineni et. al., 2002. Bleu: A method for automatic evaluation of machine translation]

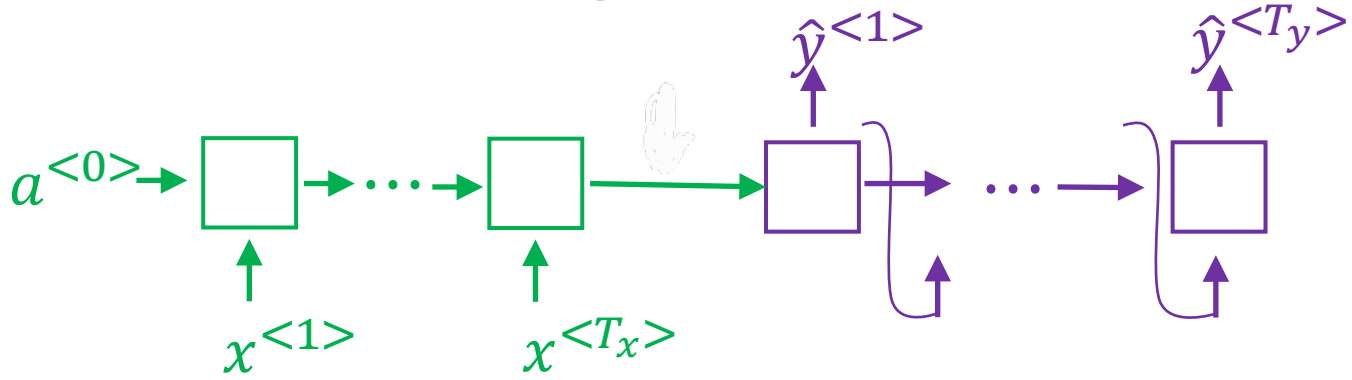
deeplearning.ai

Sequence to sequence models

Attention model intuition

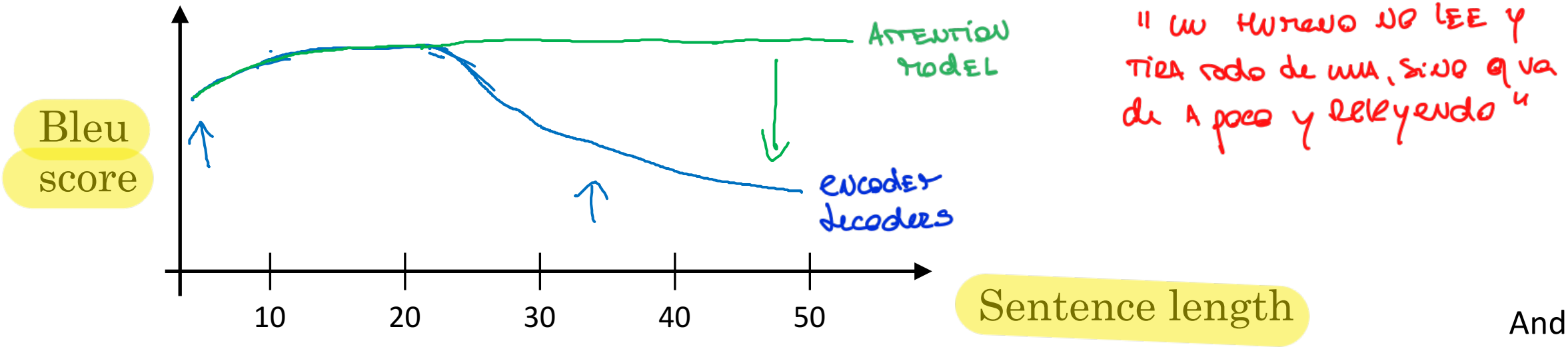
Road to Attention is all you need ♡

The problem of long sequences

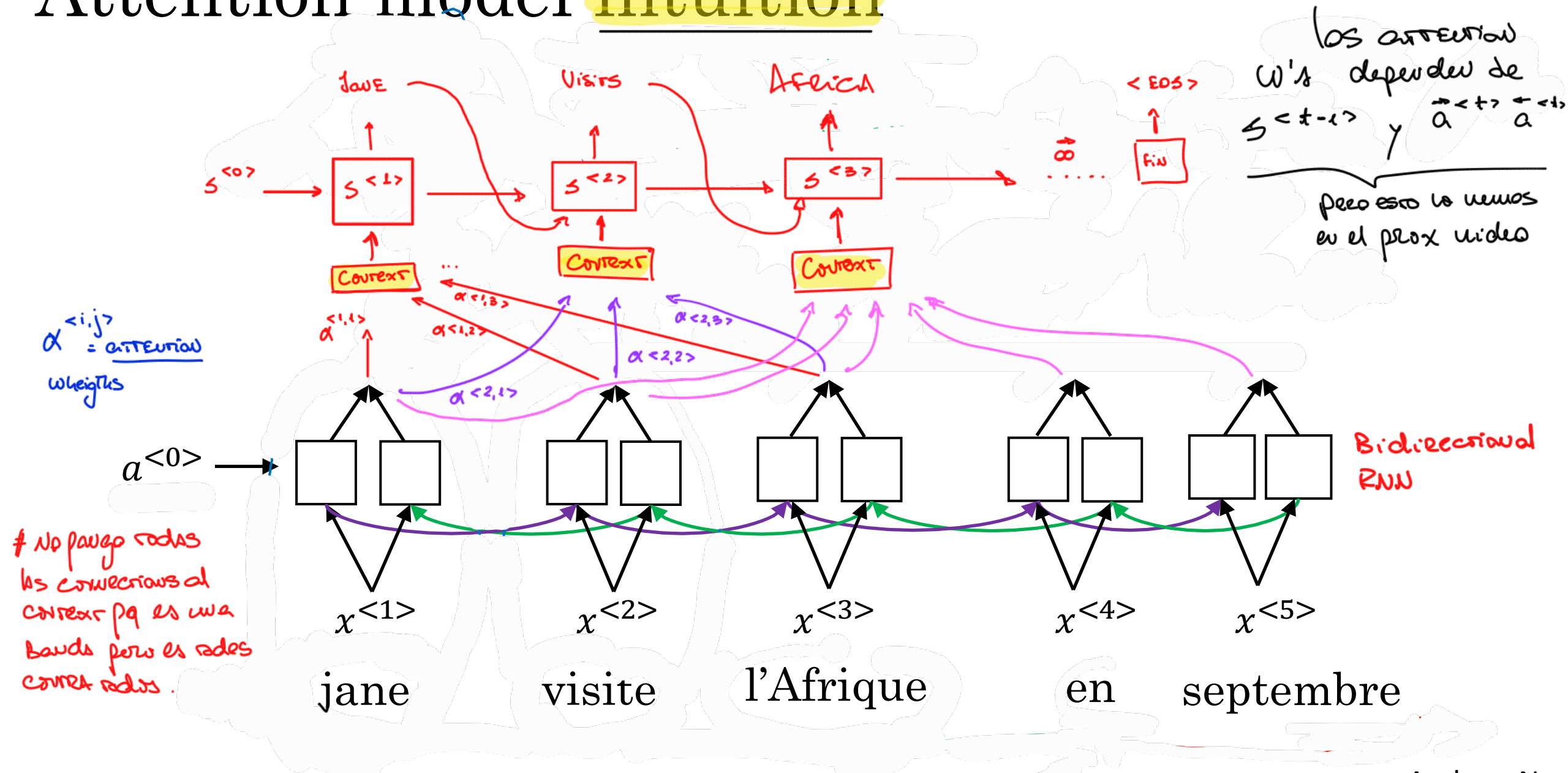


Jane s'est rendue en Afrique en septembre dernier, a apprécié la culture et a rencontré beaucoup de gens merveilleux; elle est revenue en parlant comment son voyage était merveilleux, et elle me tente d'y aller aussi.

Jane went to Africa last September, and enjoyed the culture and met many wonderful people; she came back raving about how wonderful her trip was, and is tempting me to go too.



Attention model intuition



[Bahdanau et. al., 2014. Neural machine translation by jointly learning to align and translate]

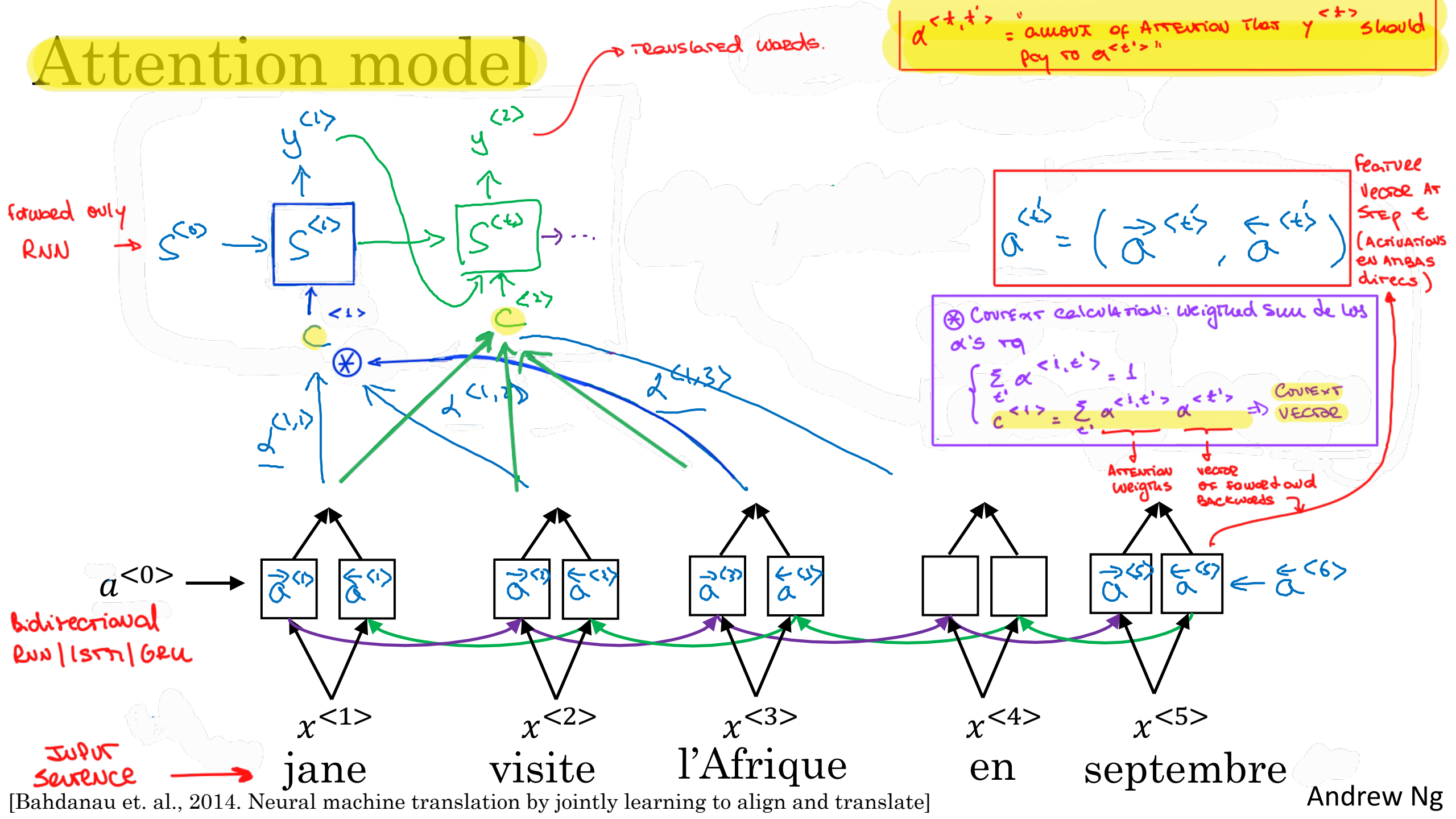
Andrew Ng

deeplearning.ai

Sequence to sequence models

Attention model

Attention model



[Bahdanau et. al., 2014. Neural machine translation by jointly learning to align and translate]

Andrew Ng

Computing attention $\alpha^{<t,t'>}$

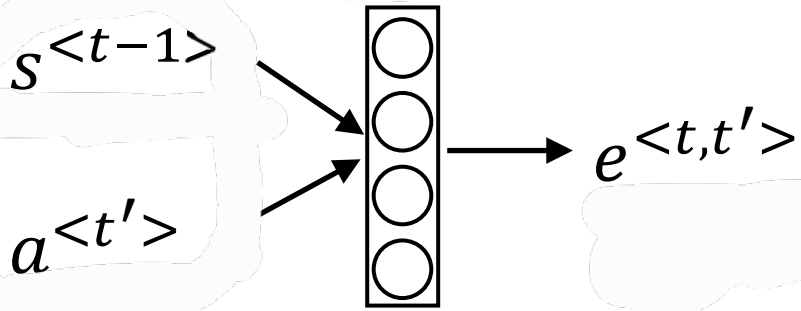
$\alpha^{<t,t'>}$ = amount of attention $y^{<t>}$ should pay to $a^{<t'>}$

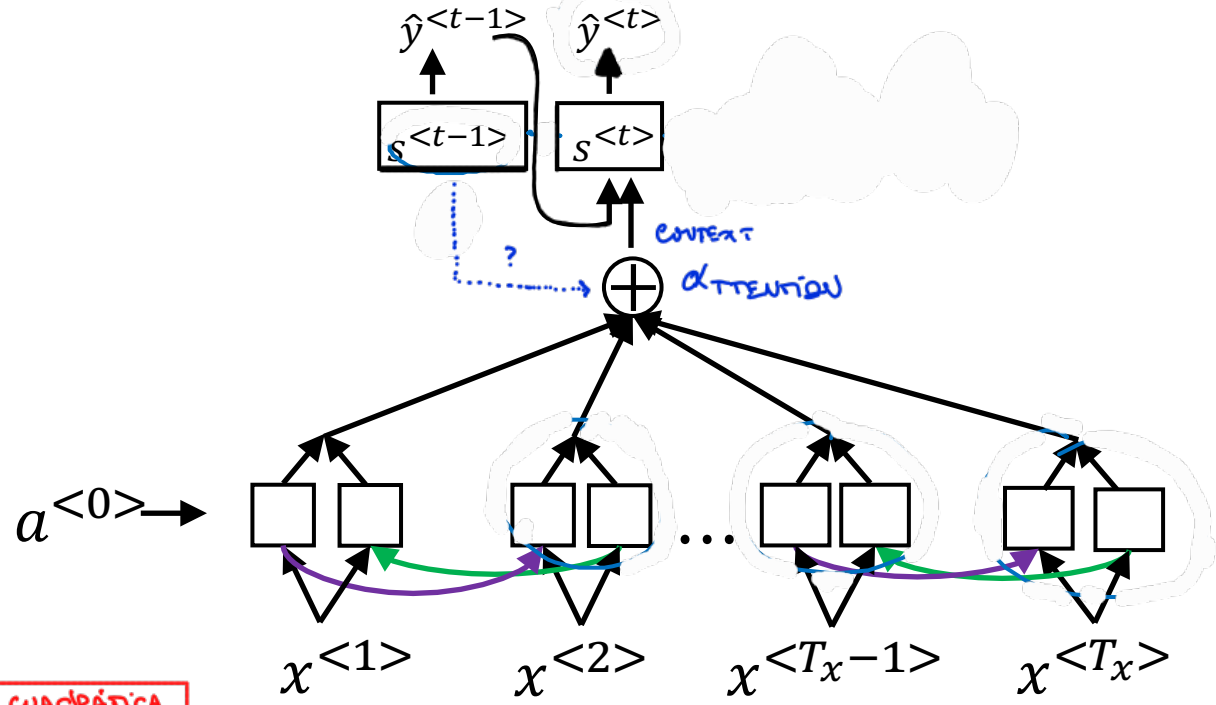
alfa! de attention — "a" de hidden state

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

→ Suman 1 p/c t'

Los "e" se calculan usando una mini red Neuronal.





S = la activation del paso anterior pero de la Red de Arriba. No entendí muy bien esto, si context necesita los α y los α necesitan los e q se calculan con los S, no está mal el dibujo?

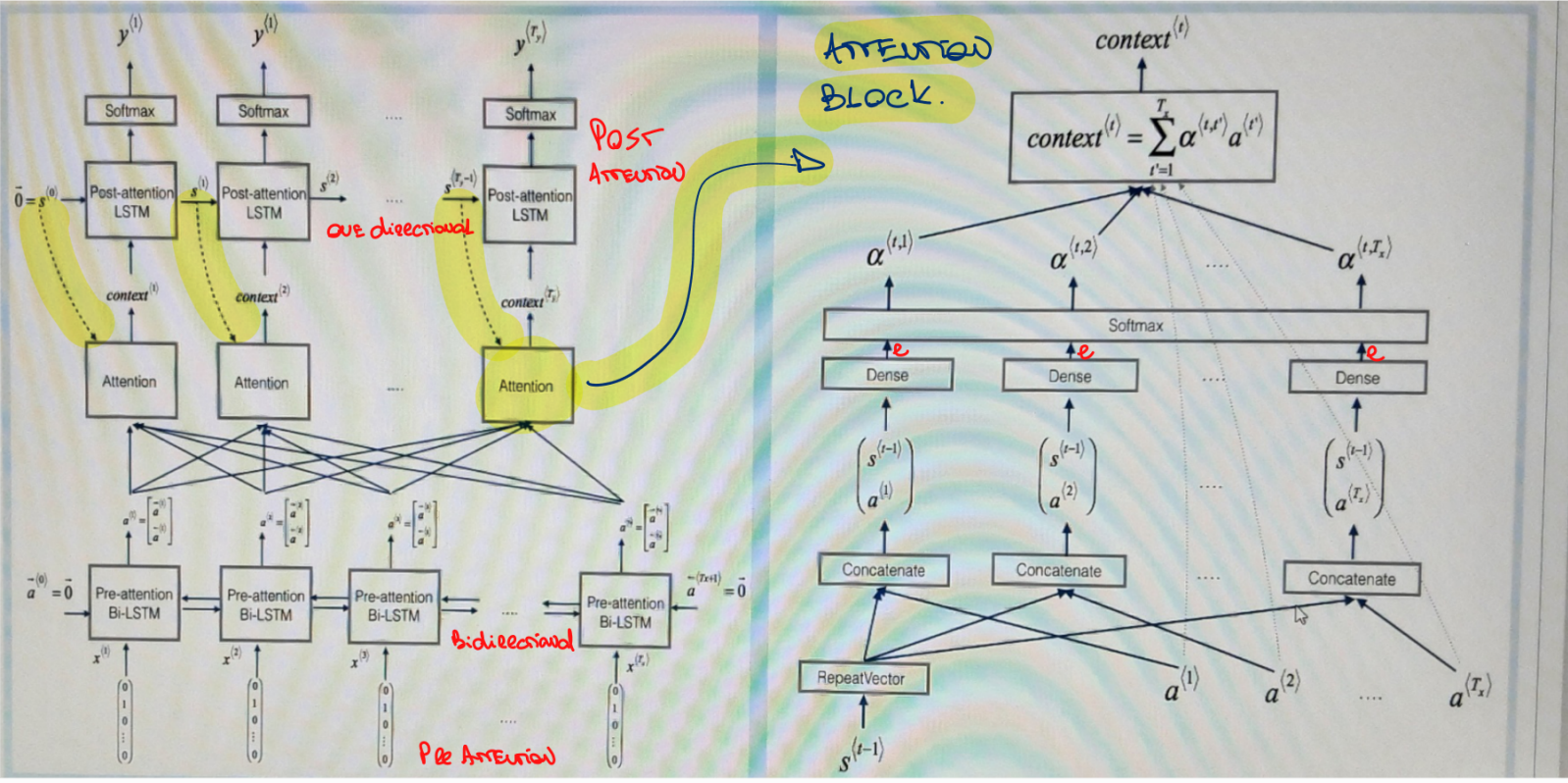
Escala de forma cuadrática este algoritmo. Hay research para tratar de hacerlo igual

Sí! Arriesga ok

[Bahdanau et. al., 2014. Neural machine translation by jointly learning to align and translate]

[Xu et. al., 2015. Show, attend and tell: Neural image caption generation with visual attention]

Andrew Ng



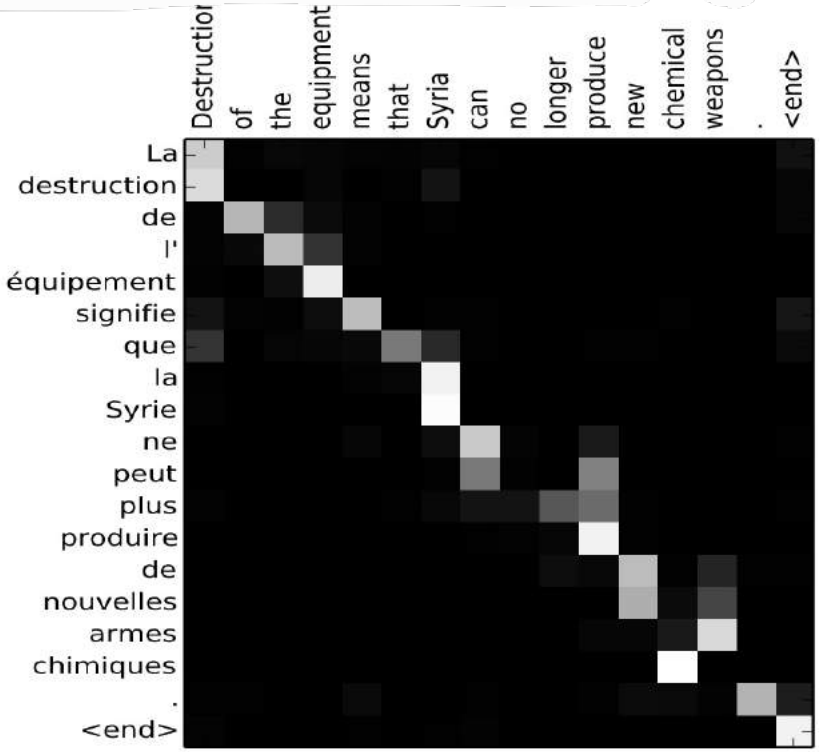
e = energies. tendrá algo que ver con los EBM's de yann lecun?

Attention examples

July 20th 1969 $\longrightarrow$ 1969 – 07 – 20

23 April, 1564 $\longrightarrow$ 1564 – 04 – 23

Visualization of $\alpha^{<t,t'>}$:



→ Attention Weights.

Andrew Ng

deeplearning.ai

Audio data

Speech recognition

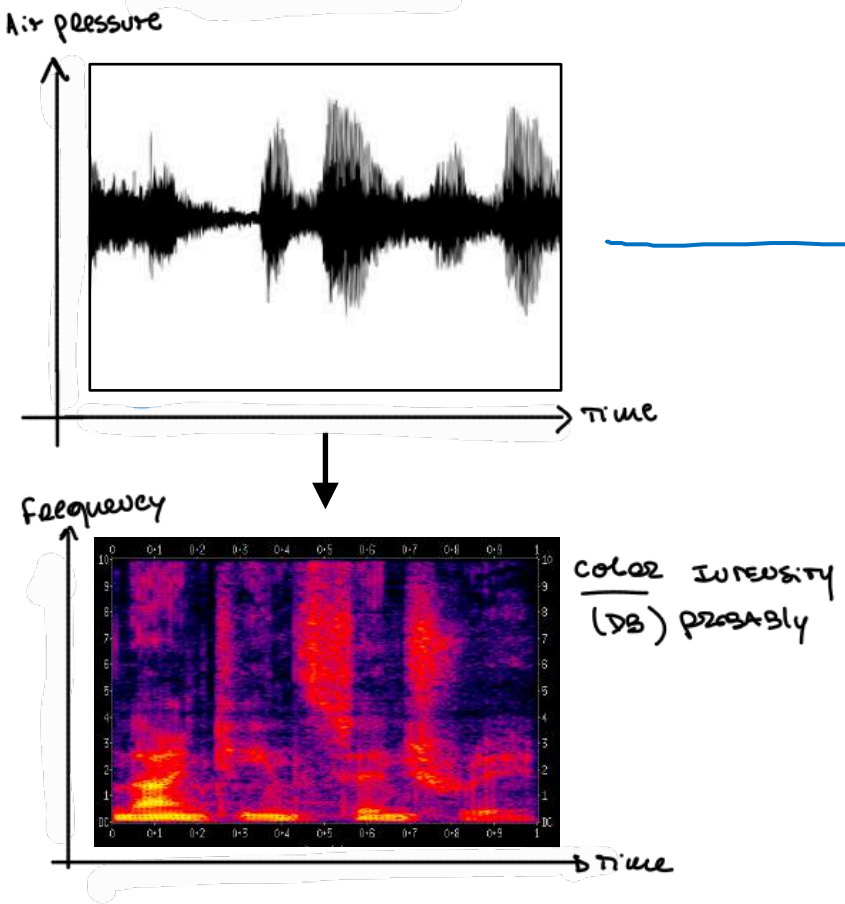
Speech recognition problem

x

audio clip

y

transcript



"the quick brown fox"

Antes, los sistemas de SR usaban "phonemes": basic units of sound. Con DL, eso no hace más falta.

→ phonemes: de kwik braun

300h

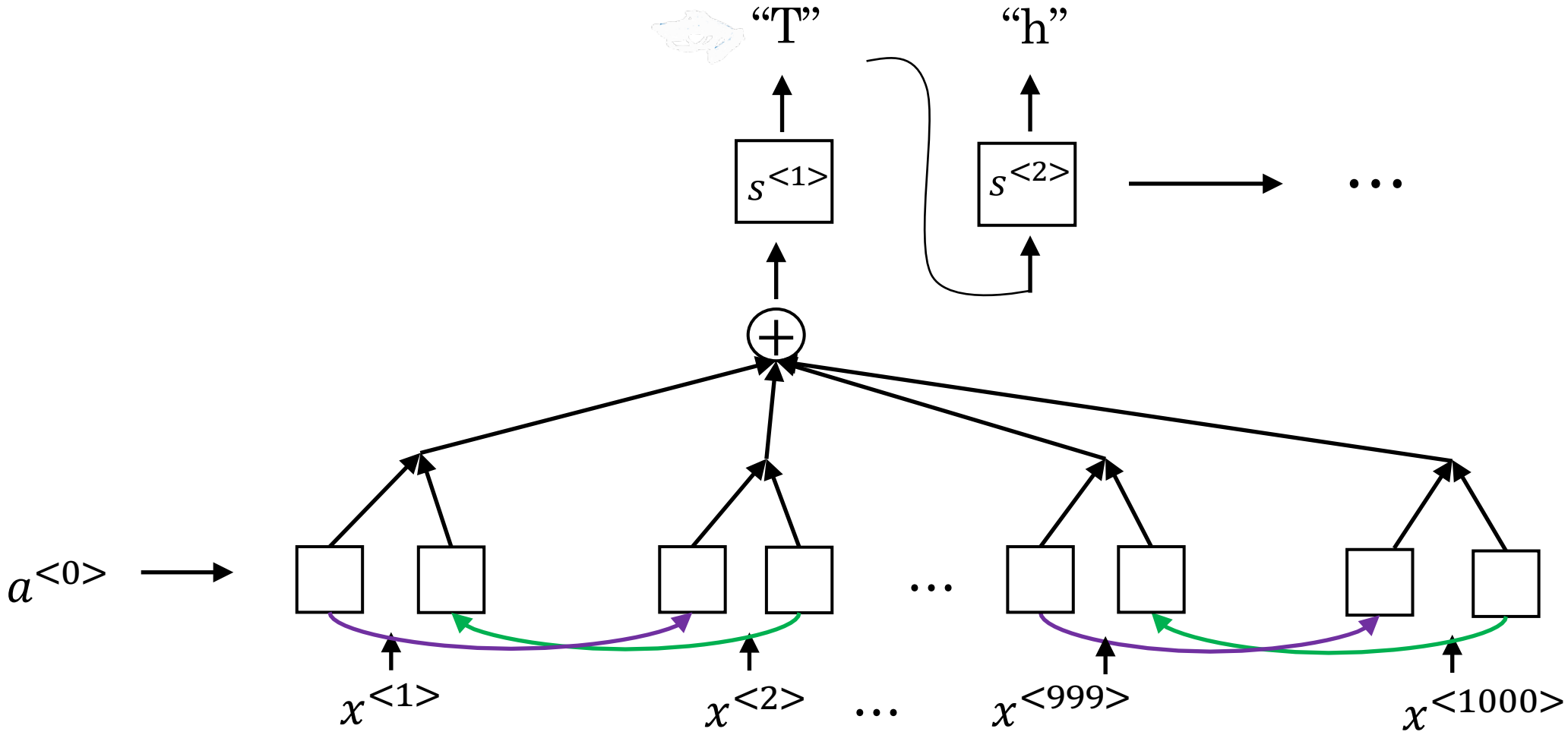
3000h

100,000h

cantidad de hs de audio de entrenamiento.

Andrew Ng

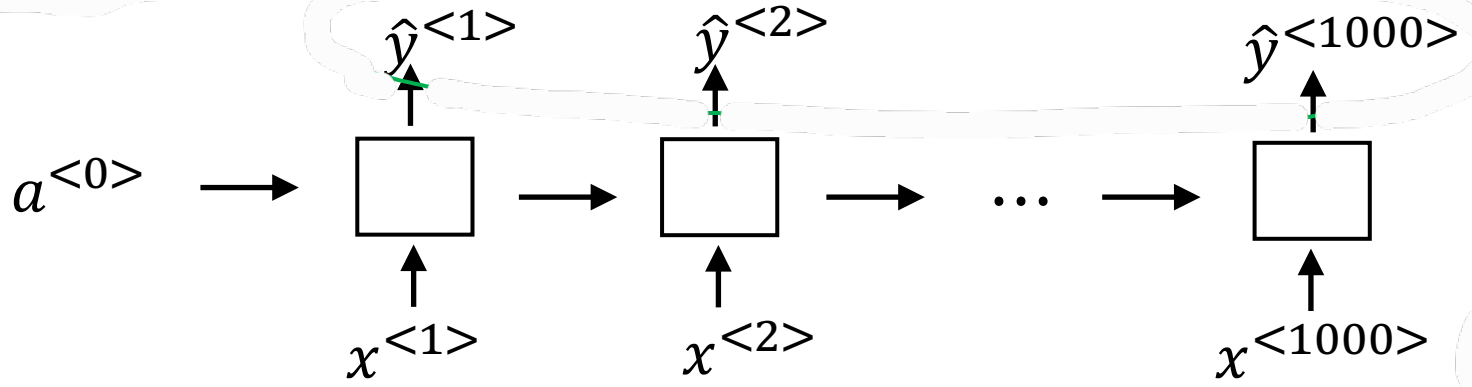
Attention model for speech recognition



CTC cost for speech recognition

(Connectionist temporal classification)

"the quick brown fox"



ttt_h_eee___ll___qqq__ → The_q

"space" "blank"

Basic rule: collapse repeated characters not separated by "blank"

[Graves et al., 2006. Connectionist Temporal Classification: Labeling unsegmented sequence data with recurrent neural networks]

Andrew Ng

deeplearning.ai

Audio data

Trigger word detection

What is trigger word detection?

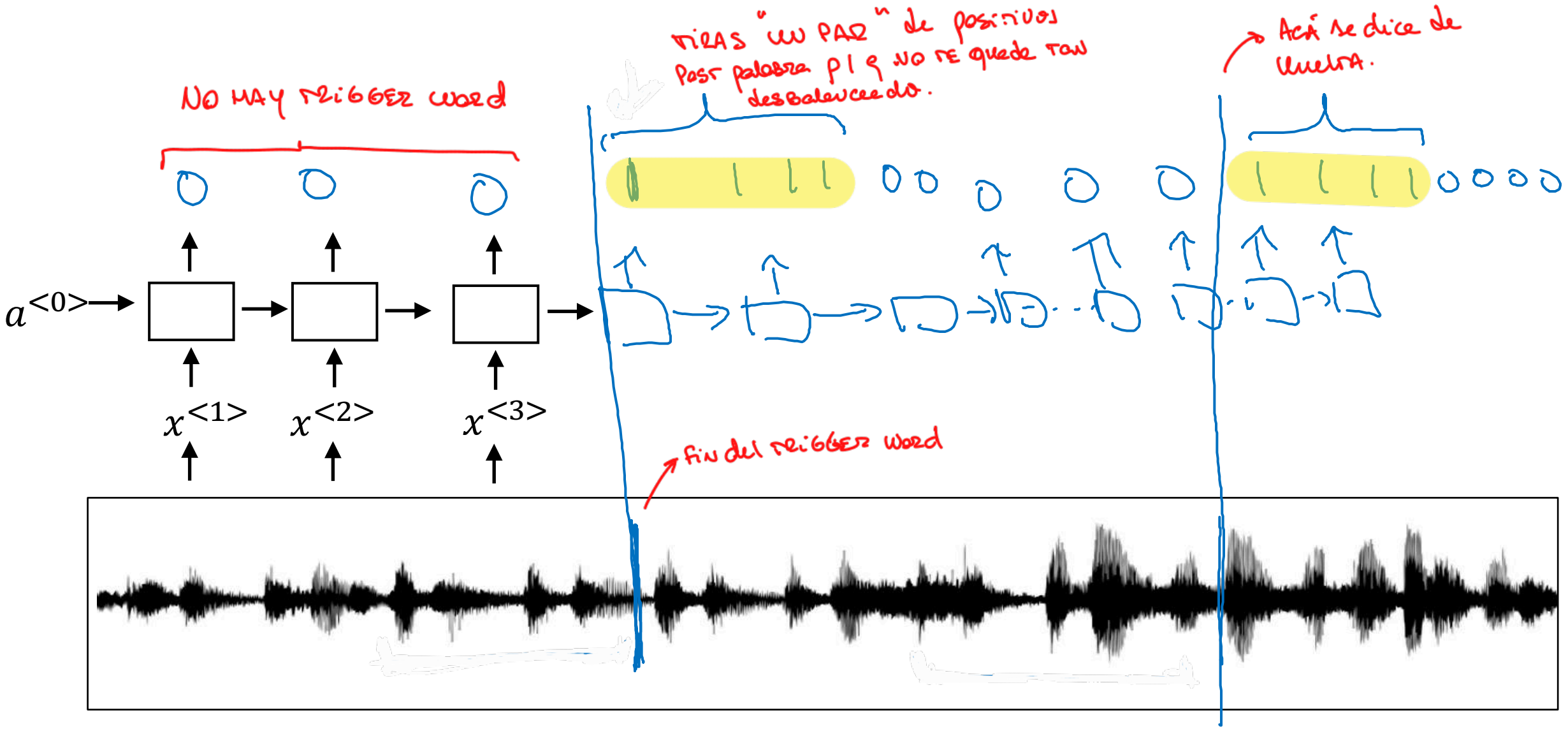
Amazon Echo
(Alexa)

Baidu DuerOS
(xiaodunihao)

Apple Siri
(Hey Siri)

Google Home
(Okay Google)

Trigger word detection algorithm



Andrew Ng

deeplearning.ai

Conclusion

Summary and thank you

Specialization outline

1. Neural Networks and Deep Learning
2. Improving Deep Neural Networks: Hyperparameter tuning, Regularization and Optimization
3. Structuring Machine Learning Projects
4. Convolutional Neural Networks
5. Sequence Models

Deep learning is a super power

Please buy this from shutterstock and replace in final video.

Andrew Ng

Thank you.

-Andrew Ng

Copyright Notice

deeplearning.ai

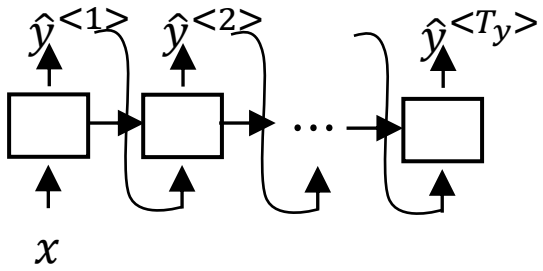
Sequence to sequence models

Transformers Intuition

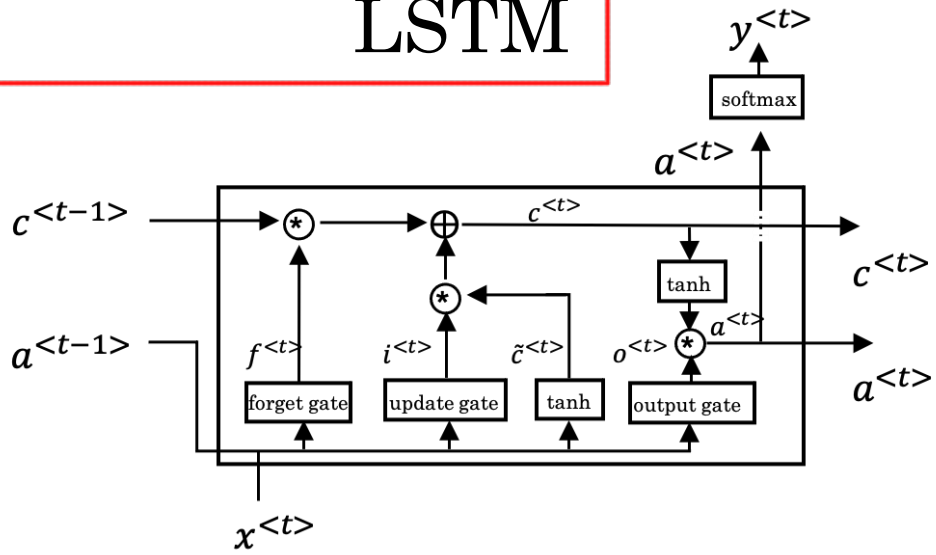
Transformers Motivation

Increased complexity, sequential

RNN → GRU → LSTM



Vanishing Gradients problems



Mucho más complejos y todos Secuencial

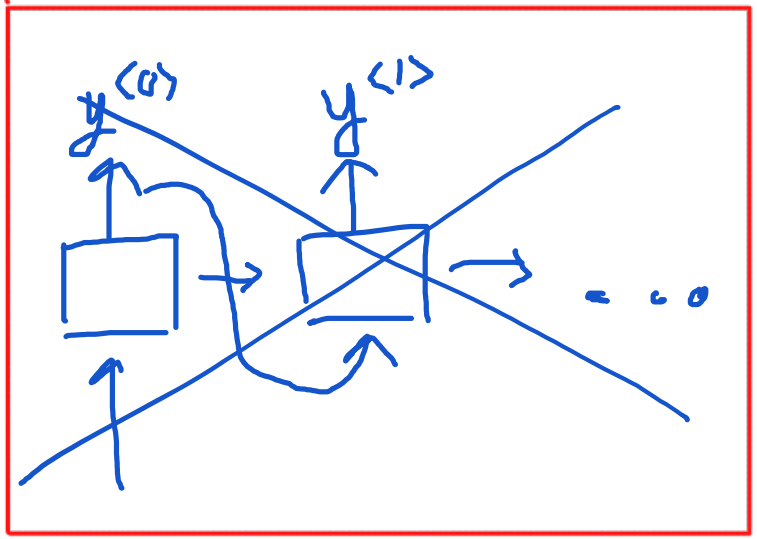
Process 1 token at a time!

Andrew Ng

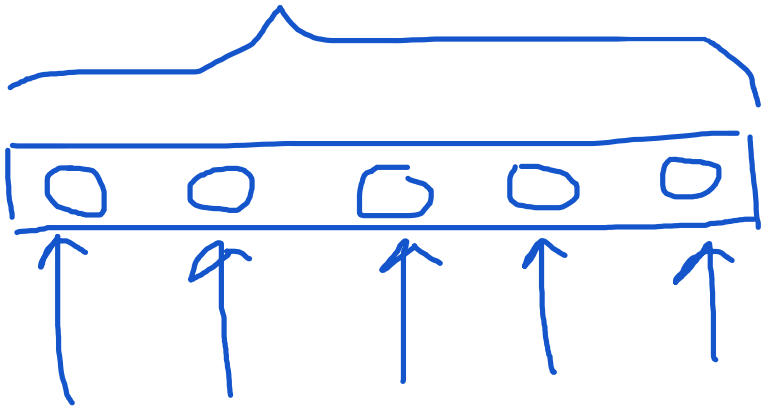
Transformers Intuition

- Attention + CNN → CNN "style"
  - Self-Attention → $A^{<1>} \ldots\ldots A^{<N>}$ ⇒ N representations in a sentence of N words & be made parallelizable
  - Multi-Head Attention → ≈ for loop sobre el proceso de self attention. → Muchas rep de Self Attention ≠ (# channels)

Sequential way of processing is Recurrent



in parallel style of CNN's



[Vaswani et al. 2017, Attention Is All You Need] ♡

deeplearning.ai

Sequence to sequence models

Self-Attention

Self-Attention Intuition

key, value, se explican luego

$A(q, K, V)$ = attention-based vector representation of a word

calculate for each word $A^{<1>}, \ldots, A^{<N>}$

RNN Attention

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

Transformers Attention

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

↳ P/ cada palabra { q = query, k = key, v = value

$A^{<3>}$

$x^{<1>}$	$x^{<2>}$	$x^{<3>}$	$x^{<4>}$	$x^{<5>}$
Jane	visite	l'Afrique	en	septembre

[Vaswani et al. 2017, Attention Is All You Need]

↳ dependiendo del contexto puede tener ≠ reps. de $A^{<3>}$

Self-Attention

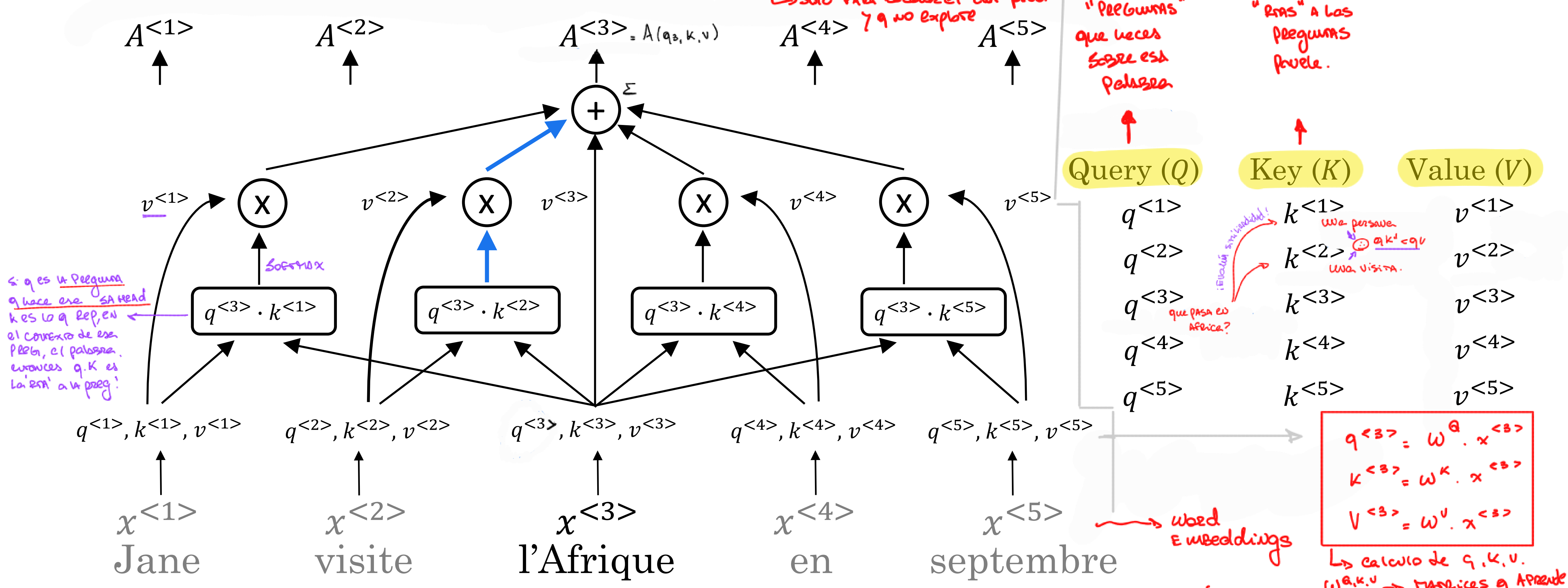
AKA: scaled dot product attention.

Vectorize Rep

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

Vectorized Rep → $$\boldsymbol{Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V}$$ → Si ponemos todas las computation de las N words juntas

└> solo para escalar el dot prod y q no explote



- La clave es q l'Afrique (ni nada) es un fixed word emb. Sino q dep de la SA head y su interpretación. En este caso: un lugar q alguien visita!

deeplearning.ai

Sequence to sequence models

Multi-Head Attention

Multi-Head Attention

Basically, "A big for loop" sobre el slide anterior } pero en realidad

Hay algo mas? ← Esto es lo PARALELIZABLE !! → NN a la q le paso todo.

$$MultiHead(Q,K,V) = concat(head_1 head_2 \dots head_n)W_o$$

$$head_i = Attention\left(W_i^Q Q, W_i^K K, W_i^V V\right)$$

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

W_o

concat

$head_3$

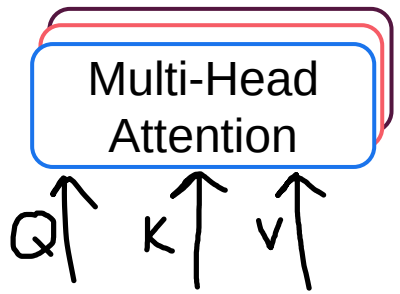
$head_2$

$head_1$

$Attention(W_i^Q Q, W_i^K K, W_i^V V)$

h = # heads

Representa todo esto en un recuadro



$W_i^Q q^{<1>}, W_i^K k^{<1>}, W_i^V v^{<1>}$

$W_i^Q q^{<2>}, W_i^K k^{<2>}, W_i^V v^{<2>}$

$W_i^Q q^{<3>}, W_i^K k^{<3>}, W_i^V v^{<3>}$

$W_i^Q q^{<4>}, W_i^K k^{<4>}, W_i^V v^{<4>}$

$W_i^Q q^{<5>}, W_i^K k^{<5>}, W_i^V v^{<5>}$

W_3^Q, W_3^K, W_3^V, Who?

W_2^Q, W_2^K, W_2^V, When?

W_1^Q, W_1^K, W_1^V, Did what?

↳ Hay muchas (i) matrices W^Q, W^K, W^V

$q^{<1>}, k^{<1>}, v^{<1>}$

$q^{<2>}, k^{<2>}, v^{<2>}$

$q^{<3>}, k^{<3>}, v^{<3>}$

$q^{<4>}, k^{<4>}, v^{<4>}$

$q^{<5>}, k^{<5>}, v^{<5>}$

$x^{<1>}$ Jane

$x^{<2>}$ visite

$x^{<3>}$ l'Afrique

$x^{<4>}$ en

$x^{<5>}$ septembre

W^Q, W^K, W^V

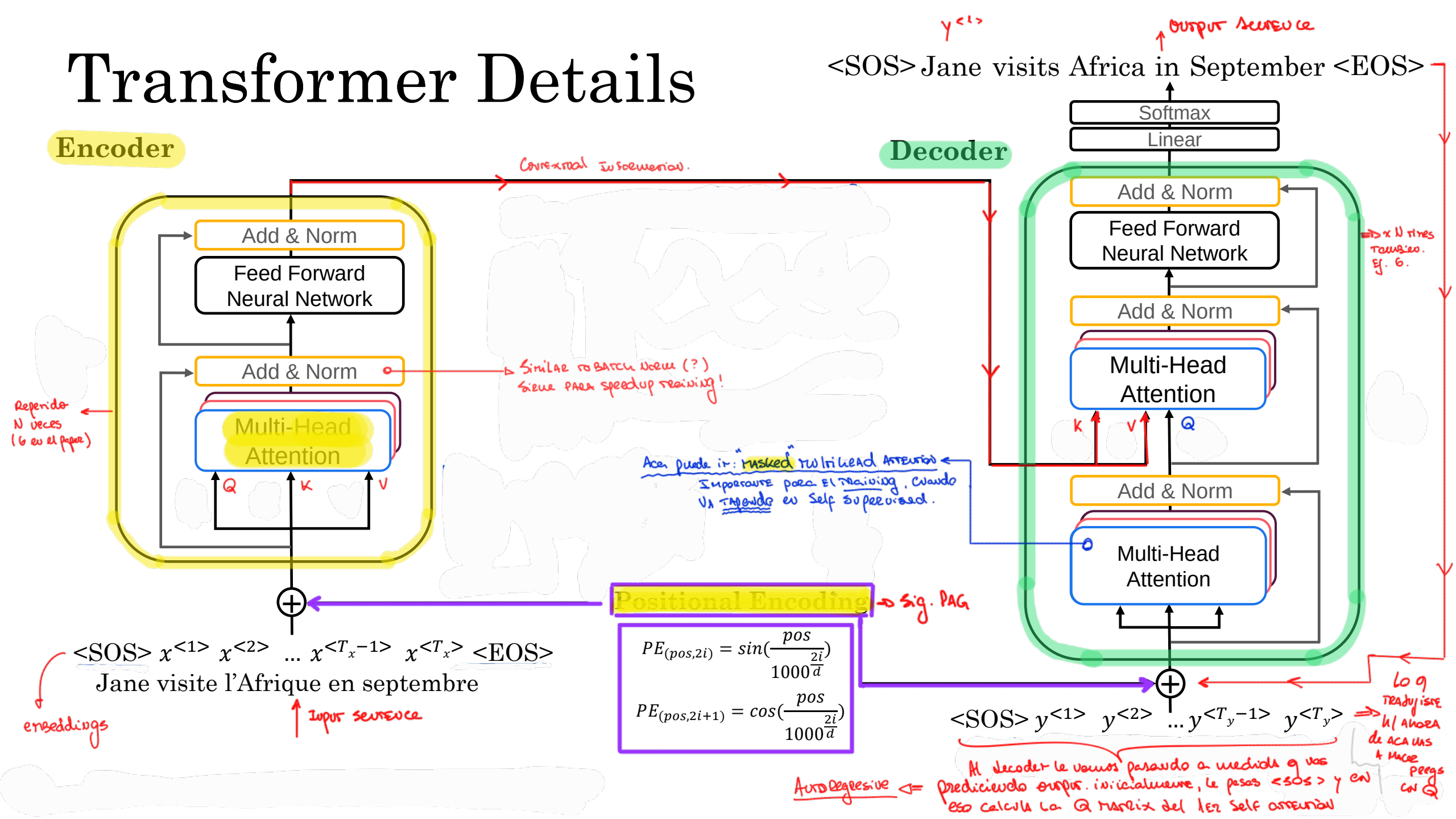
- $W_1^Q; W_1^K; W_1^V$ → Responden la preg "que esta pasando? p/c word (Beej). $W_2^{Q,K,V}$ podrian preguntar "cuando esta pasando?". etc. Se aprenden esas "preguntas" → como si fueran features! q le vamos a dar a una NN :)

deeplearning.ai

Sequence to sequence models

Transformers

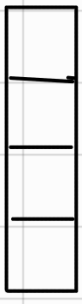
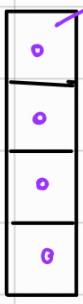
Transformer Details



Positional Encoding

⇒ attention no toma en cuenta el orden. por eso usamos el pe.

Pensemos que tenemos un input embedding de tamaño $d = 4$

$x^{<1>} =$  → creamos un positional embedding → $p^{<1>} =$ 

→ From Position 1 (vea gráfica)

$i = 0$	$K = 0$
$i = 0$	$K = 1$
$i = 1$	$K = 2$
$i = 1$	$K = 3$

pe. Encoding único pl esa palabra.

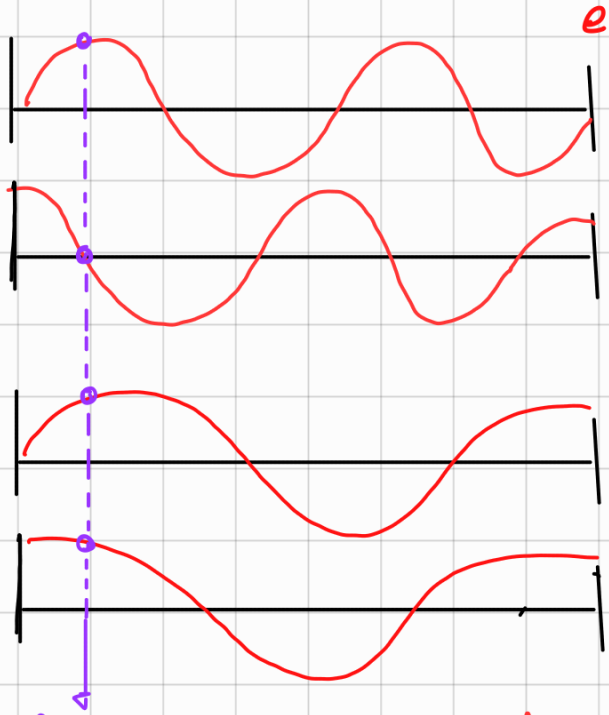
$$PE(pos; 2i) = \sin\left[\frac{Pos}{10000^{\frac{2i}{d}}}\right]$$

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

→ esto se le suma a los inputs de los embeddings

⇓

- $K = 2i = 0 \rightarrow i = 0$
- $K = 2i+1 = 0 \rightarrow i = 0$
- $K = 2i = 2 \rightarrow i = 1$
- $K = 2i+1 = 3 \rightarrow i = 1$



⇒ Menor freq

Position 1 → es un valor único pl cada pos.