Recurrent Neural Networks

Arun Pandian J
Bennett University (Sabbaticals)

Sequence Application Variation

- Audio Signal to Sequence - Speech Recognition
- Nothing to Sequence or Single Parameter to Sequence - Music Generation
- Sequence to Single Output - Sentiment Classification
- Sequence to Sequence - Machine Translation
- Video Frame Sequence to Output - Activity Recognition
- Sub-Sequence from a Sequence - Finding Specific Protein from a DNA Sequence
- Outlining Specific parts of a sequence - Name Entity Recognition

Notation Understanding

X: Rama Conquered Ravana to install the virtue of dharma

$x^{<1>}$ $x^{<2>}$ $x^{<3>}$ …… $x^{<t>}$ …… $x^{<9>}$

T_x = 9 (Length of training sequence: 9)

$x^{i<t>}$: t^{th} word of ith training sequence

Y: 1 0 1 0 …… 0 0 0 0

$y^{<1>}$ $y^{<2>}$ $y^{<3>}$ …… $y^{<t>}$ …… $y^{<9>}$

T_y = 9 (Length of output sequence: 9)

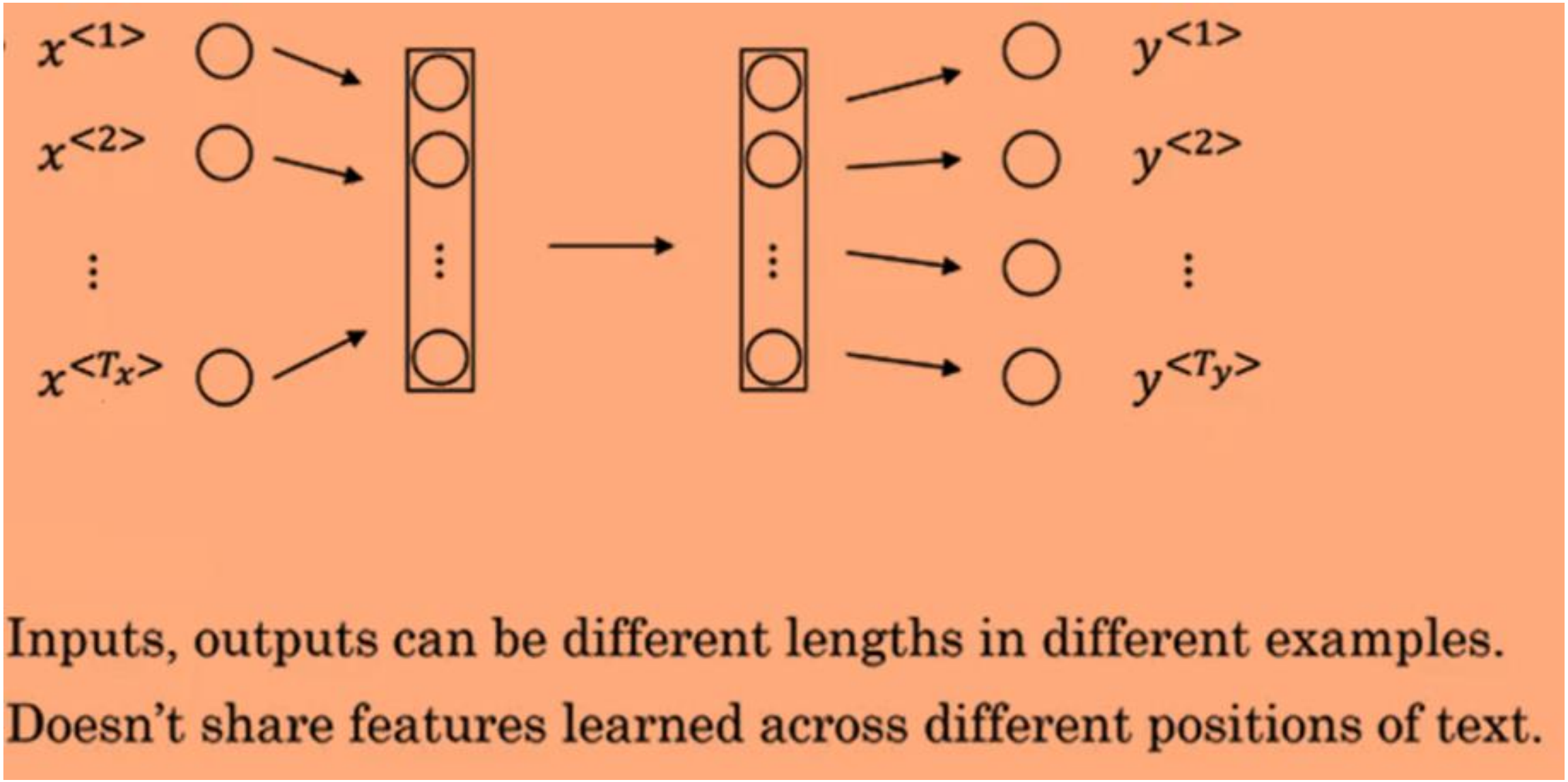
$y^{i<t>}$: t^{th} word of ith output sequence

Representing words and one-hot encoding

X: Rama Conquered Ravana to install the virtue of dharma

A	1	Rama	Ravana
:		0	0
:		0	0
Conquered	329	0	0
:		0	0
:		0	0
Install	4521	:	:
:		:	:
:		:	:
Rama	7689	1 -7689	:
:		:	1-7900
Ravana	7900	:	:
:		0	0
ZZZ	10000	0	0

Standard Neural Network Does not works out to give a good application for sequence models



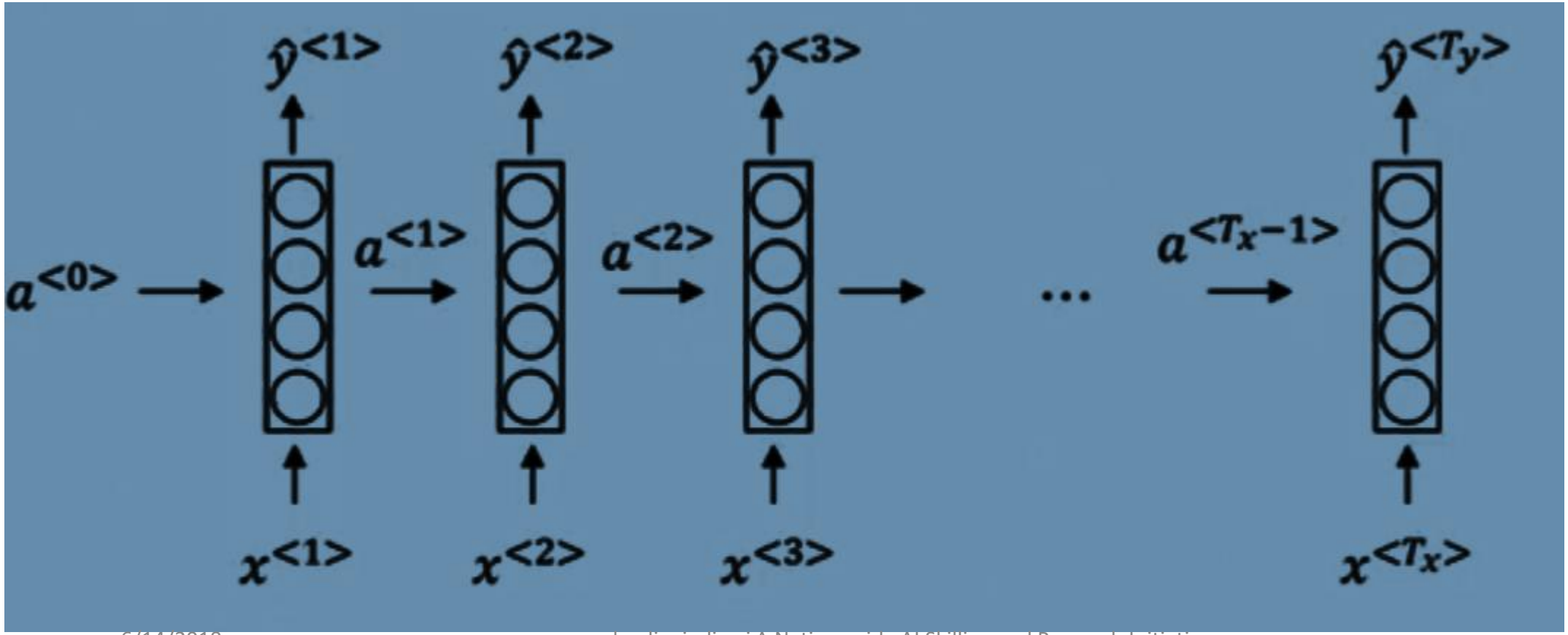
Inputs, outputs can be different lengths in different examples.

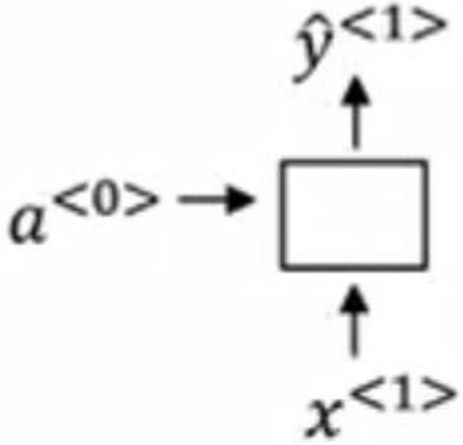
Doesn't share features learned across different positions of text.

Forward Propagation

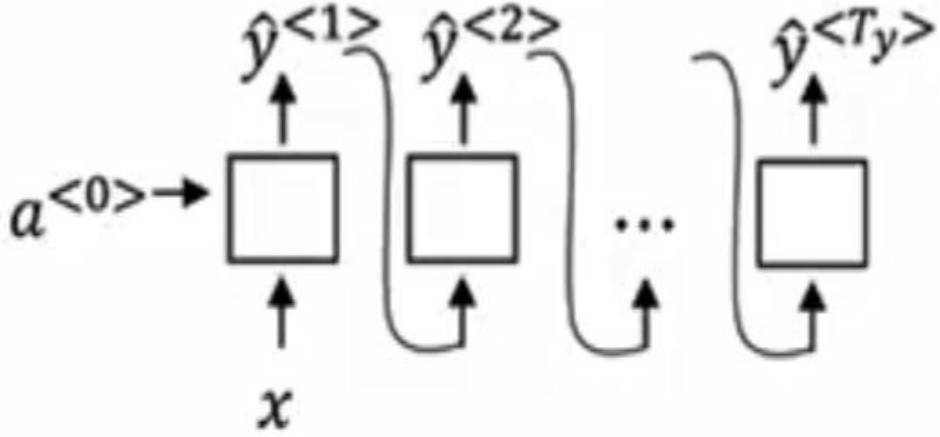
$$a^{<t>} = g(W_a[a^{<t-1>}, x^{<t>}] + b_a)$$

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

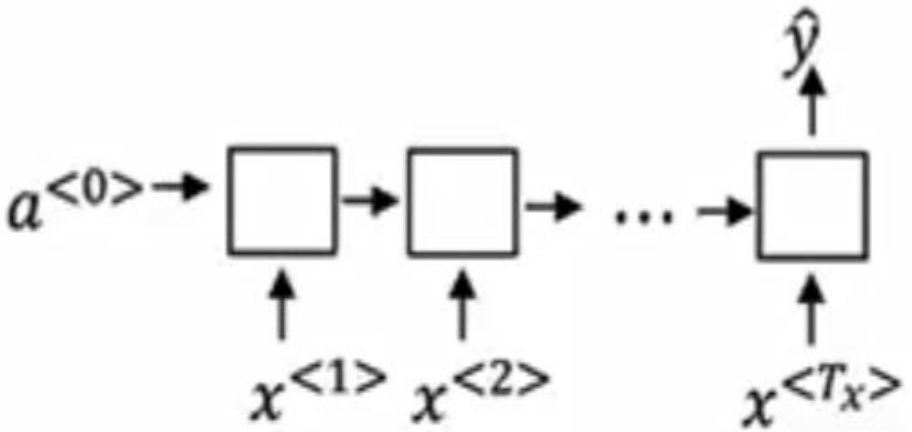




One to one

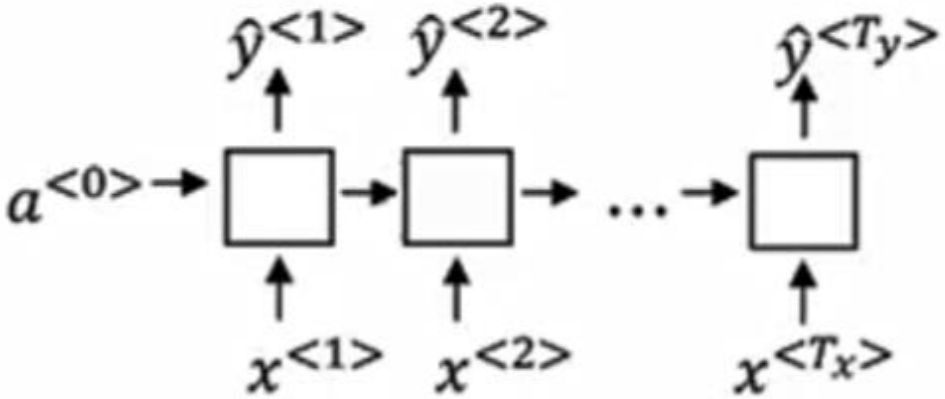


One to many

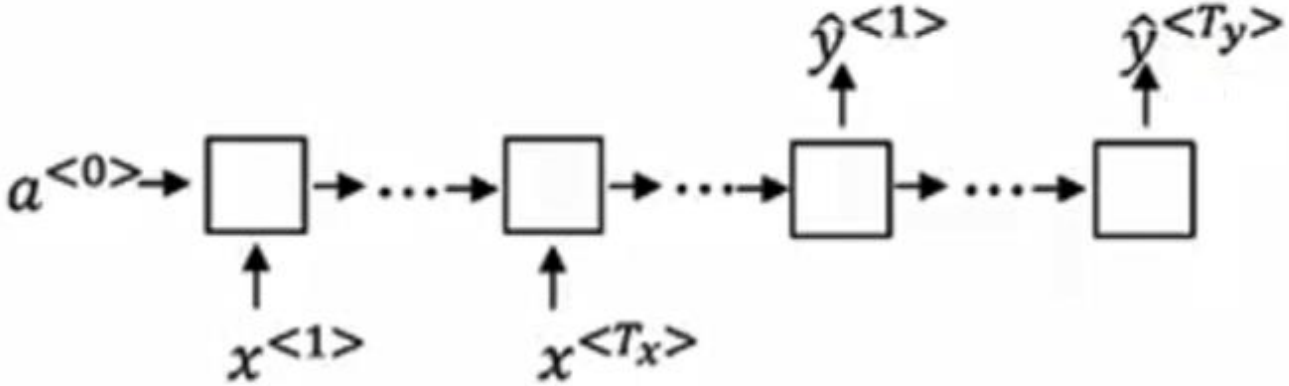


Many to one

Different Type of RNN Architectures



Many to Many



Many to Many

Word Level Language Model

Train your Language model on a large data.

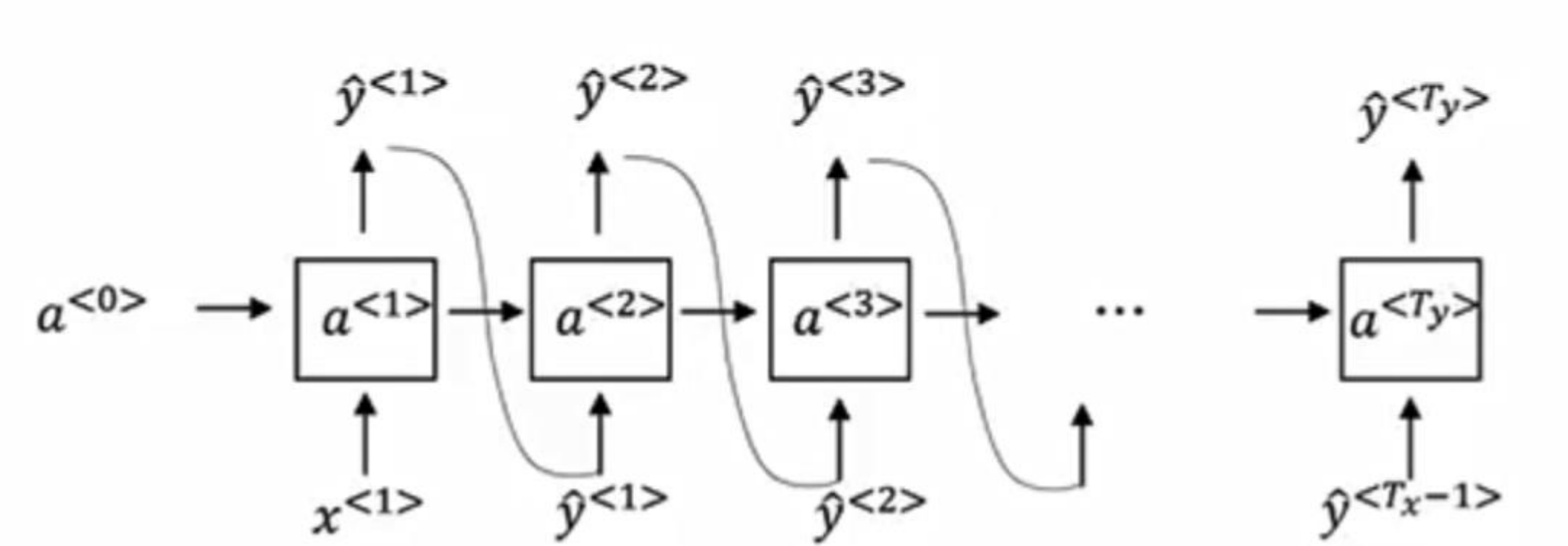
Then You can build on it different kinds of NLP applications as discussed.

Also, You will be able to generate new sentences or paragraphs etc as per the requirements of your application.

Difference between Word Level and Character Level Language Model

- Word Level Language models are more common due to their better performance as of now.
- Word level language models have <EOS> and <UNK> also as tokens in the corpus.
- Character level corpus is very small as compared to word level corpus
- In most cases word level corpus are of size 30-50k but in some cases can be upto 1 million
- In character level language model it becomes hard to predict and relate the relationship between far off characters as the distance becomes very large as compared to word language models.

Word level and character level language model



Sampling a novel sequence

Once you have a trained model on a corpus you can also have a RNN that can sample new sequences for you.

In that case you initialize with a zero and your first output gives a probability in terms of softmax function of the size of the no of categories equal to the size of your corpus.

You Choose a random word as the first output and then that word acts as the input for the second input and so on.

If you get a <UNK> then you can reject that token and continue with the next guess. It can go on until you get a <EOS> token.

Vanishing Gradients Issue

Regular RNNs are mostly influenced by local variations.

The cat , which already ate a lot while enjoying the party, was full.

The cats , which already ate a lot while enjoying the party, were full.

If the RNNs are not able to take care of long term dependencies, then the performance will be below expectations.

For Feed forward RNNs and also for back propagation, it is very difficult to reflect/relay the long term dependencies.

Gated Recurrence Unit

- Helps a Lot in Long Term Connection and also helps a lot in vanishing gradient issue .
- Main difference is that bring in the change in the hidden unit calculations and we introduce a memory cell
- We introduce an update gate u which will have value 0 or 1
- When the value of Gate is 0 that means it will keep the previous value and will not update else it will update the previous value with the new value
- We also introduce a gate r which means relevance of previous memory cell with the current cell

GRU

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

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

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

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

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

LSTM (Long Short Term Memory)

- In this we don’t have memory Cell value equivalent to activation value.
- We also introduce an additional gate called forget gate. It gives us the option to keep the previous values and also to add/update this gate with additional value from update gate.
- LSTM is most popular now for dealing with long term dependencies
- We also have a output gate.
- LSTM is more robust than GRU, but people use both of them based on applications.

LSTM

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

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

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

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

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

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

Featurized Representation: Word Embeddings

	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
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						

If we have 300 such properties and 10000 Words then it will be a 300x10000 Matrix and is denoted as Embedding Matrix (E) and O_{man} is One hot Vector for Man Word and e_{man} can be embedding vector for Man word.

I Want A glass of orange __Juice
I want a glass of Apple

Transfer Learning and Word Embeddings

- Learn Word Embeddings from Large Text Corpus (1-100 Billion Words)
- Pre-trained embeddings are available online
- Transfer Embedding to a new task with smaller training set
- Continue to finetune word embeddings with new data

Analogies using Word Vectors

As Man-> Woman King->?

As INR->India Dollar->?

As Delhi->India Kathmandu->?

As Tall->taller Big->?

As Man->Woman Boy->?

……

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

Find a word w : Maximize similarity(e_w , $e_{king}-e_{man}+e_{woman}$)

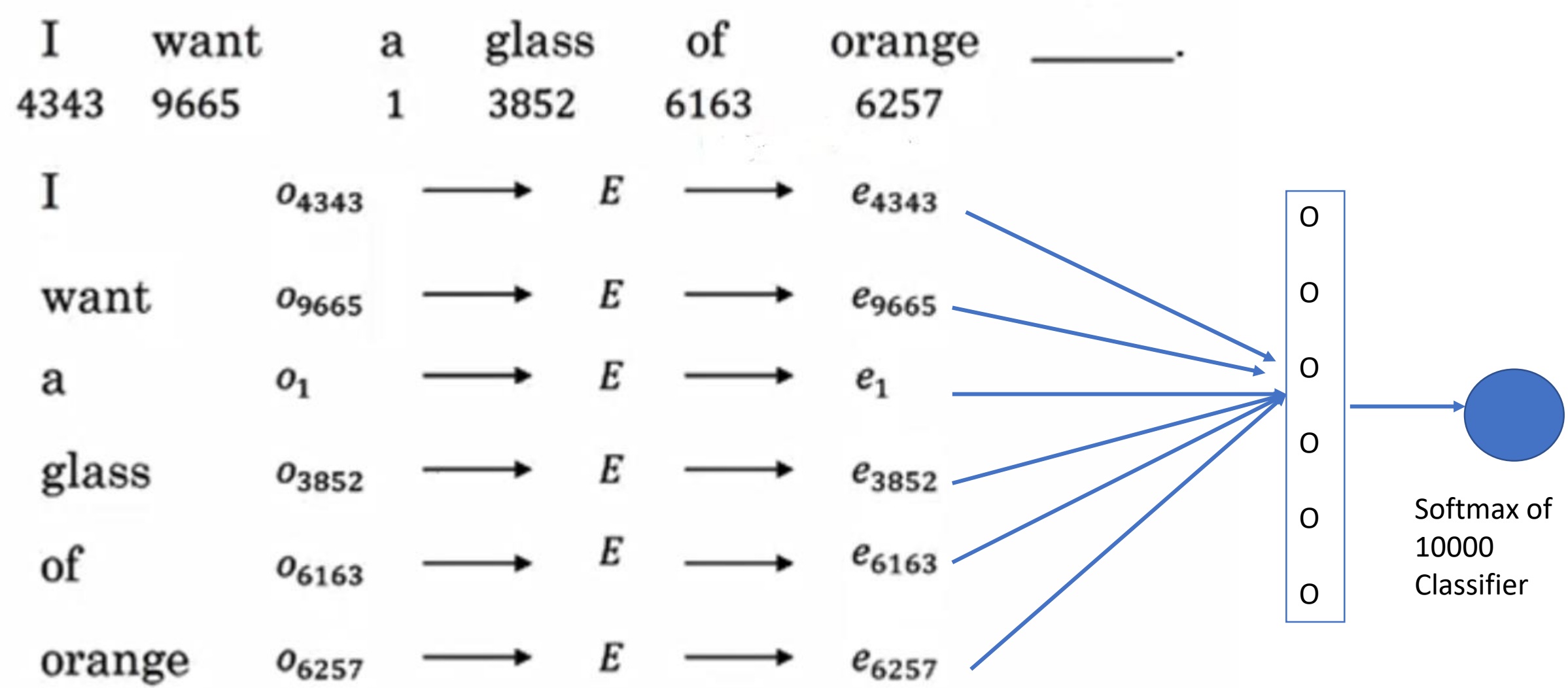
Cosine similarity

Sim(u,v)= $u^T v$ / ||u|| ||v||

Contexts which can help to learn

- Last 4 words
- Last x words and next x words
- Last one word
- One nearby word
- Randomly pick a word to be a context word and randomly pick a target word with in a window of x of the context word – Skip Gram Model

Neural Language Model



Skip Gram Model

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

Context : Orange Target: Juice

Context: Orange Target : Glass

Context: Orange Target: my

O_c->E->e_c------→Softmax Classifier→ŷ

Θ_t is the paameter associated with output t i.e chance of being the label

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

Negative Sampling Algorithm

Defining a new learning problem

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

We will use of pair of words and then find out whether the second word can be a target word for the first context word

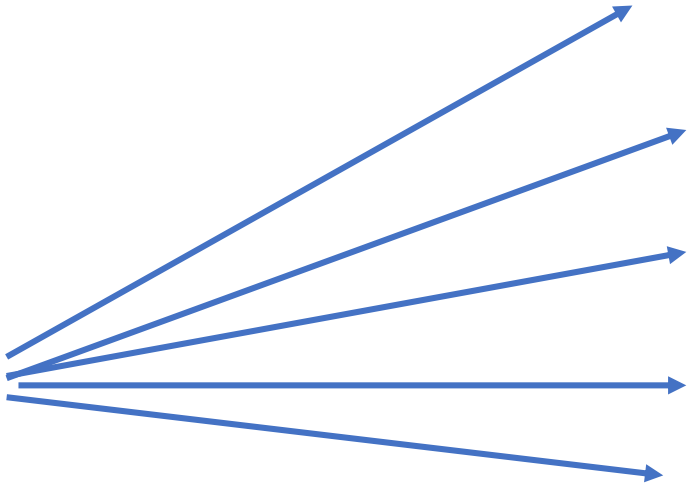
Context	Word	Target	
Orange	Juice	1	(We choose First example as positive example)
Orange	King	0	
Orange	Table	0	
Orange	to	0	
Orange	Pencil	0	

Generally we choose 5-20 such pairs for smaller datasets and 2-4 words for bigger data sets

Converting Softmax to Logistic Classifier

- O_c->E->e_c------→10000 Binary Logistic Classifier→$\hat{y}$

- O_c->E->e_c

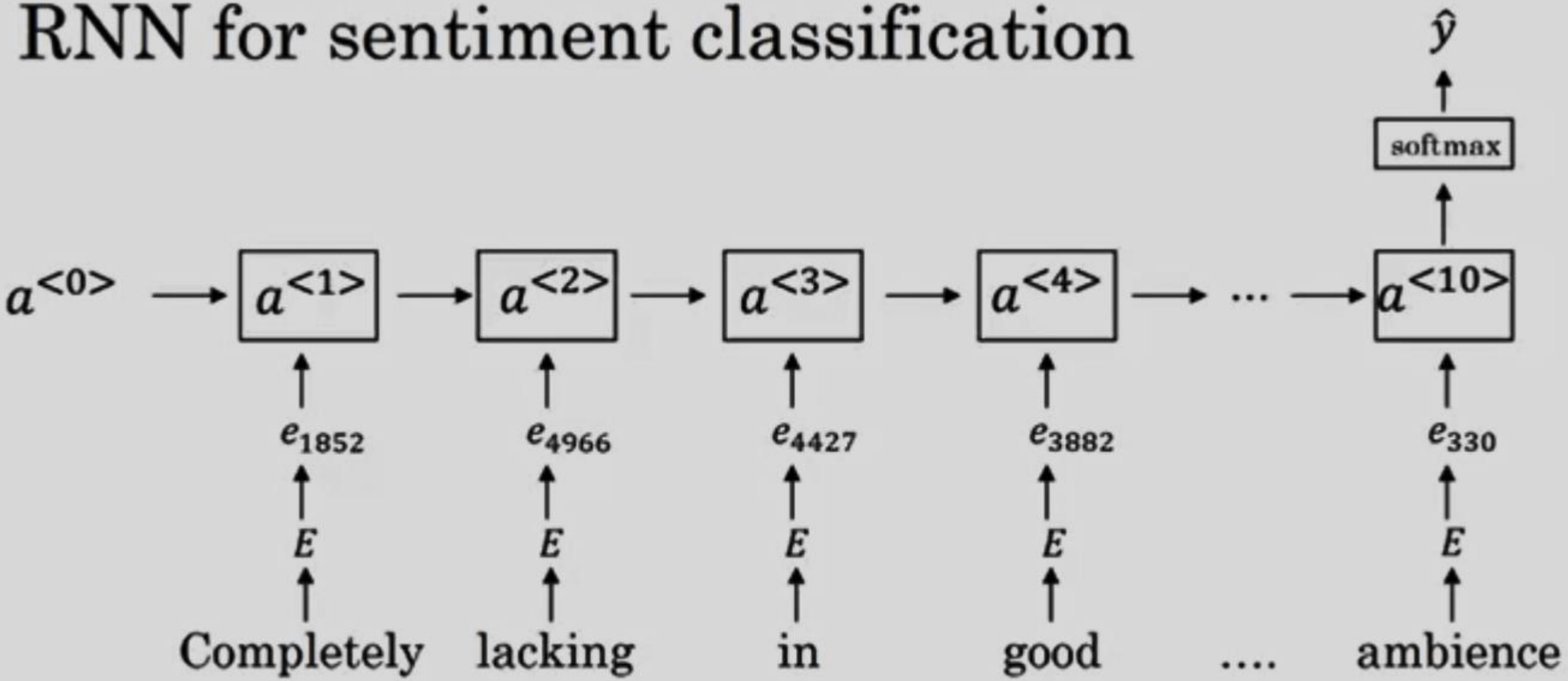


- We have 10000 binary classifiers but we only train 5 random pair words in every iteration

Simple Sentiment Classification Model

Counter Example :Completely Lacking in Good Ambience, Good Taste, Good Service

RNN for sentiment classification



Removing Biases in NLP

- It related to Gender and ethnicity biases and we need to be very careful about this
- Man: Computer_Programmer as Women Homemaker
- Father: Doctor Mother: Nurse

- Biases will be picked from the text it has been trained upon
- First step is to identify Bias Direction e.g. male to female
- Next is to Neutralize the bias for all the non-definitional word for example Father, Mother, He, She are definitional word for Gender and should not get changes due to this. However, Non-definitional word like soldier, doctor, Manager, Programmer etc should be neutralized for bias
- Last step is to equalize pairs like niece, nephew; grandmother, grandfather and they should be equidistant from words like babysitter etc.

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.

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

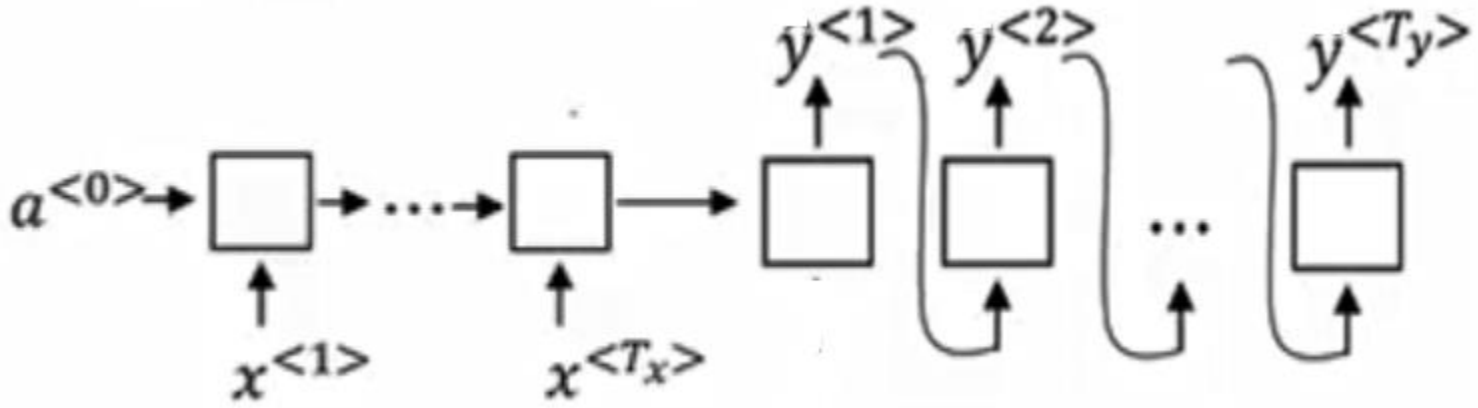
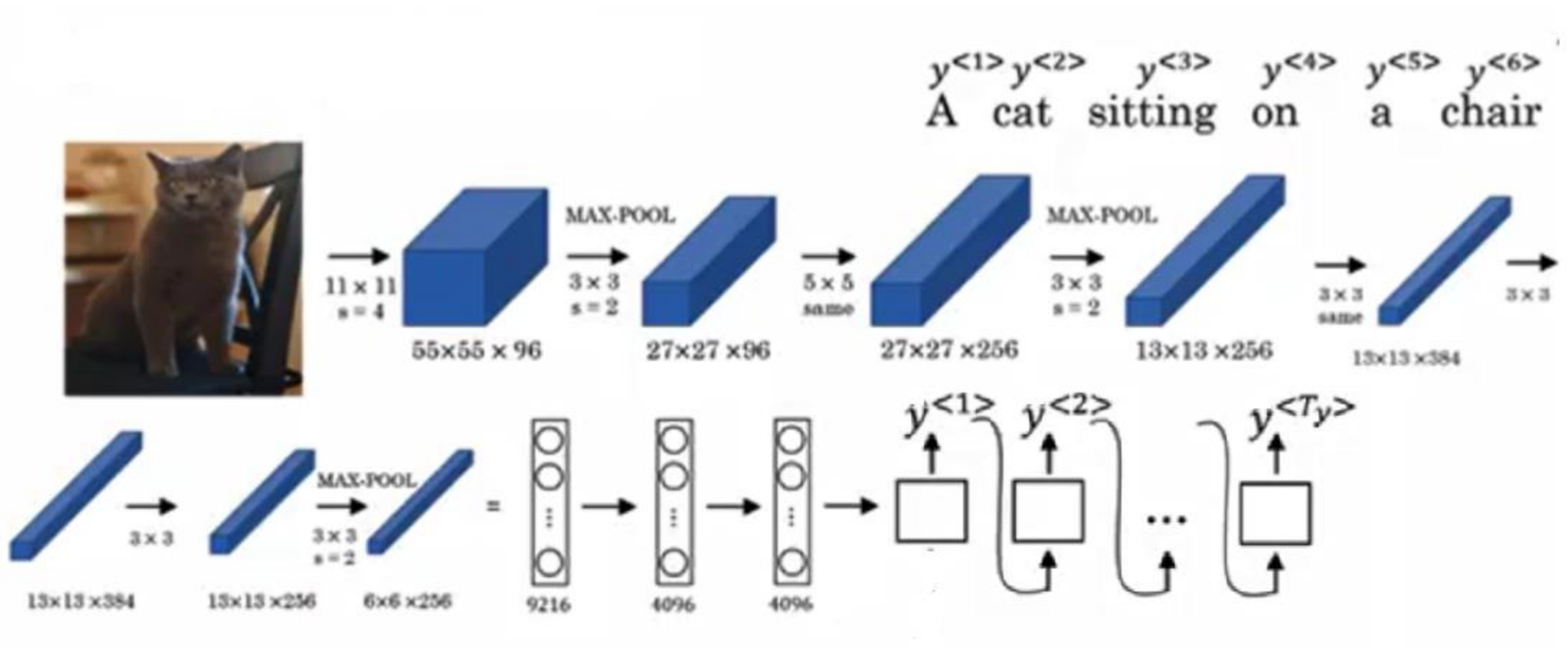
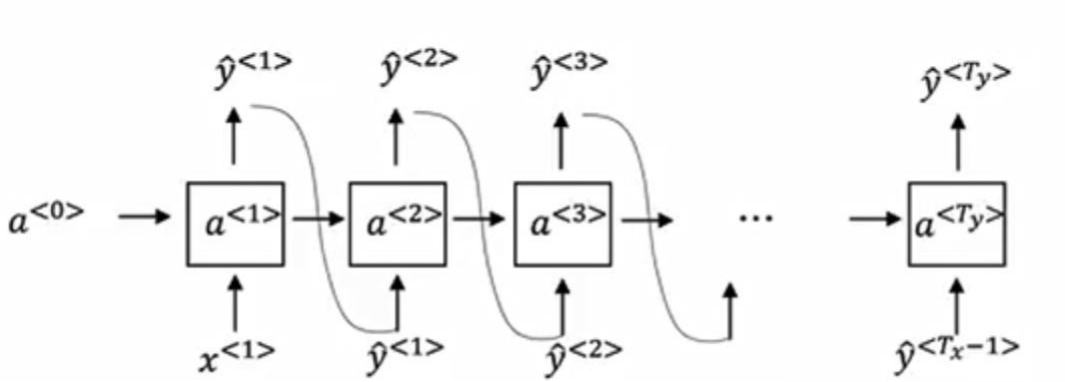


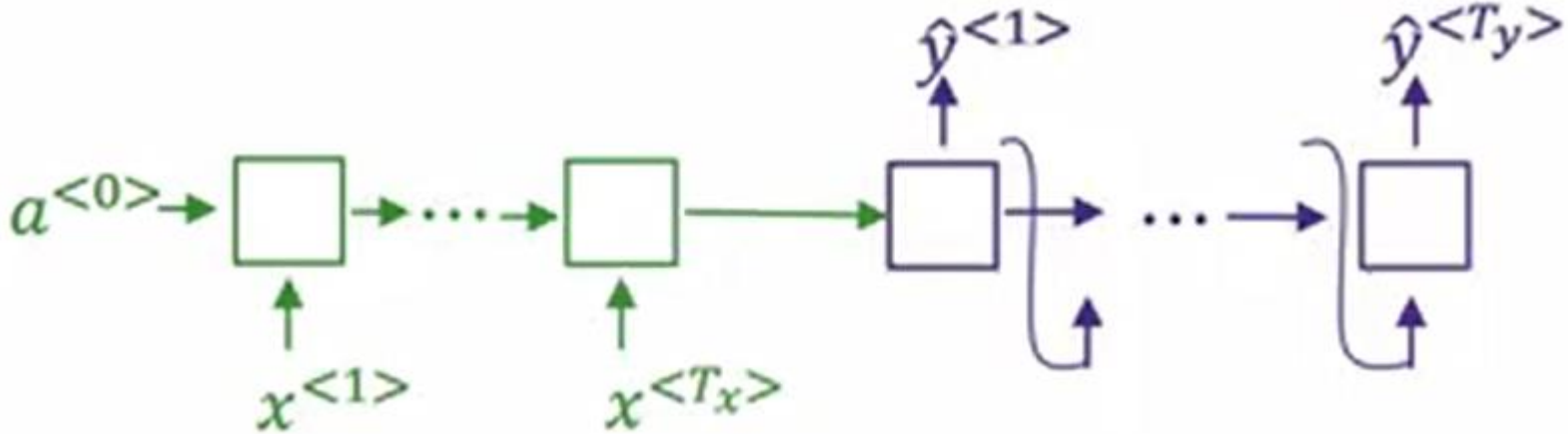
Image captioning model



Machine translation as a conditional Language Model



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



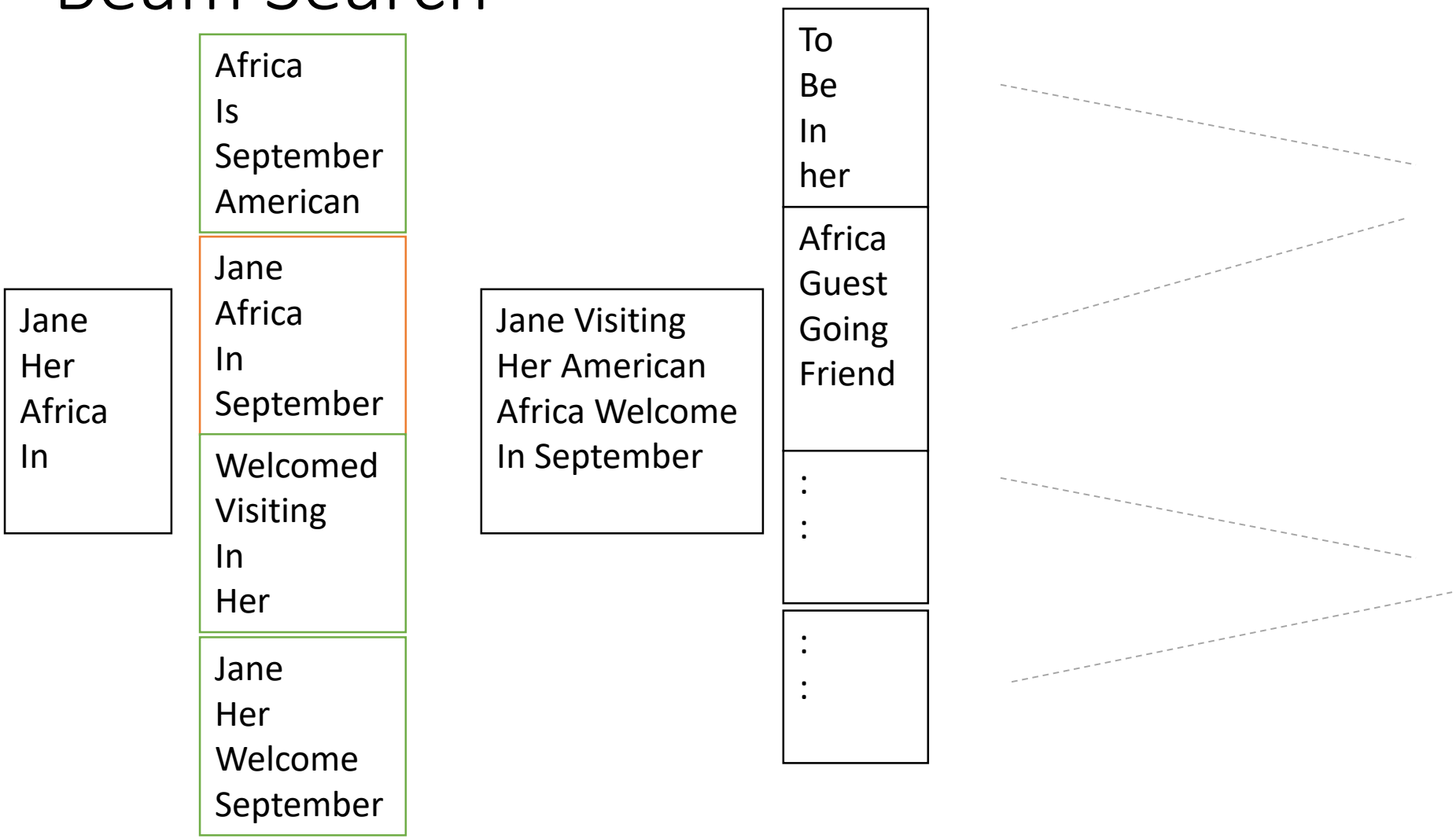
Finding the Most likely translation

Jane Visite l'afrique en septembre

- Jane is visiting Africa in September
- Jane is going to be visiting Africa in September
- In September, Jane will visit Africa
- Her American Friend Welcomed Jane in September

- If the length of the output sequence is 10 and the size of the corpus is 10000 then the possible number of permutations are 10000^{10}
- In a Greedy search algorithm we can choose the first word which is most likely to come (with maximum probability) and then the next output word of the translation with the maximum probability and so on. But, in practice this does not works and does not gives us a good translation.

Beam Search


Jane
Her
Africa
In
Africa
Is
September
American
Jane
Africa
In
September
Welcomed
Visiting
In
Her
Jane
Her
Welcome
September
Jane Visiting
Her American
Africa Welcome
In September
To
Be
In
her
Africa
Guest
Going
Friend
:
:
:
:

Beam Search

In this technique we will choose width of the Beam (Lets say 4)

It means that now we will select 4 words with highest probability given by the output of the softmax function in the last layer

Now corresponding to the each word in the first output we will choose 4 words who have max probability to come as a second word given first word.

Out of these 16 combinations of first two words, we will only choose 4 pairs with highest total probability

Now we will choose the third word corresponding to these four pairs of first two words and so on.

Beam Width

- Large beam width will result in slow performance but better results
- Smaller beam width will result in good performance with compromise in accuracy,
- Generally the acceptable length of the beam width in production systems will be in the range of 3-10 based on the applications, while in case of researchers it can even go to 100.

Bleu Score

- Bilingual Evaluation Understudy **Score**, or **BLEU**for short, is a metric for evaluating a generated sentence to a reference sentence. A perfect match results in a **score** of 1.0, whereas a perfect mismatch results in a **score** of 0.0
- Le Chat est sur le tapis
- The cat is on the mat –Ref 1
- There is a cat on the mat – Ref 2
- The the the the the the the
- Precision 7/7
- But in case of bleu score we use the clip count, which considers the maximum no of times the word appears in reference sequences. In this the appears twice in the Ref1 so clip count will be 2/7

Bleu score

It is important because there can be multiple right translations of a sentence, in that case also we need to choose one of them and it also helps in correctly assessing the error distance between the right translations and machine translation.

The cat is on the mat –Ref 1

There is a cat on the mat – Ref 2

The cat the cat on the mat – Machine Translation

The cat	2	1
Cat the	1	0
Cat on	1	1
On the	1	1
The mat	1	1

4/6 is the precision on the bigram

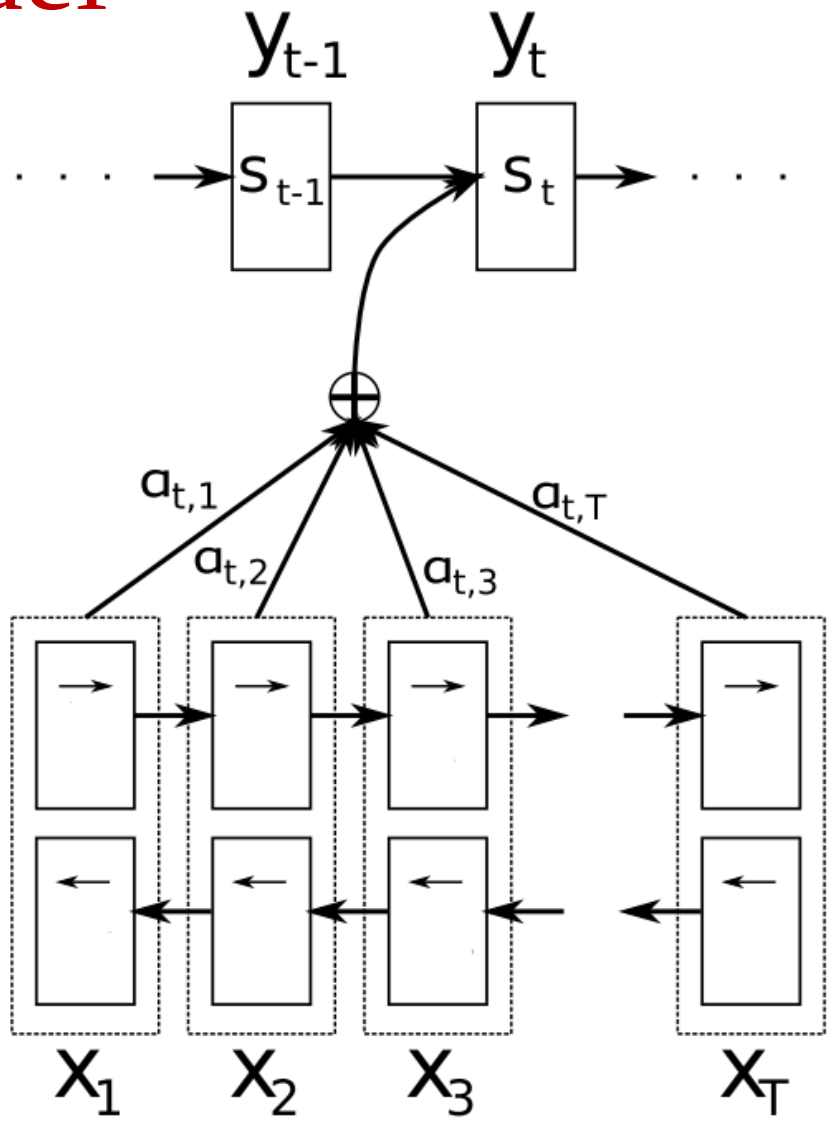
Similarly we will do on trigram and so on

It will help us to choose the right sequence

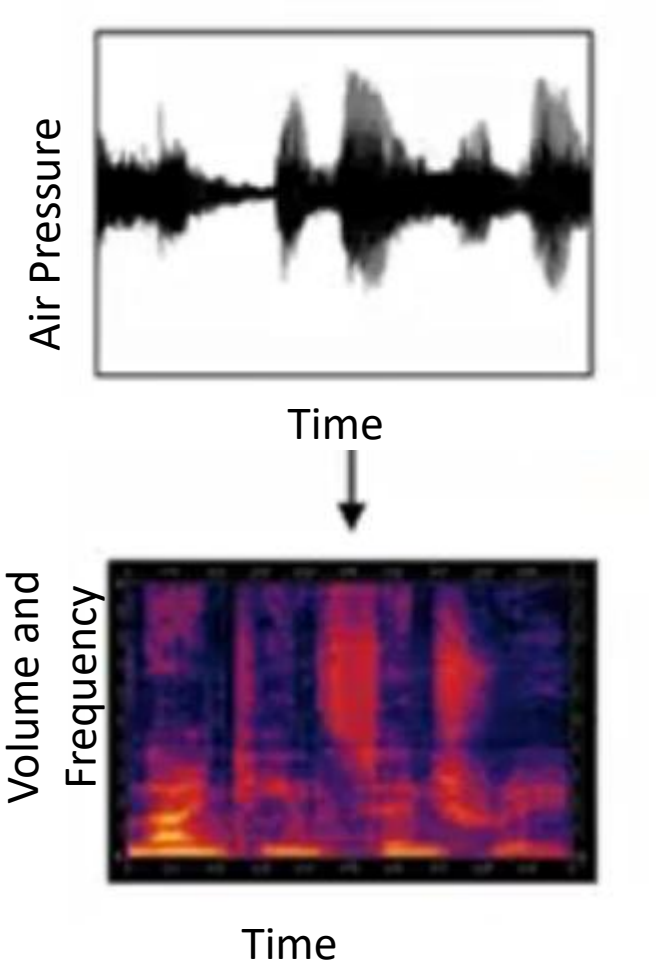
Problem of long sequences: Attention model

- When the sequences grow beyond a certain length (20 or more), then the bleu score goes down considerably and generally does not has good mapping with the goodness of the translation.
- To handle this recently Researchers came out with a new model called attention mechanism.
- It basically tells that how much attention needs to paid to each word of the source sequence for every position of the translated sequence.
- Total of attention values for any particular word should be equal to 1, so we can think of it as a softmax classifier with a small neural network for determining the probabilities of each attention value vector.

Attention Model



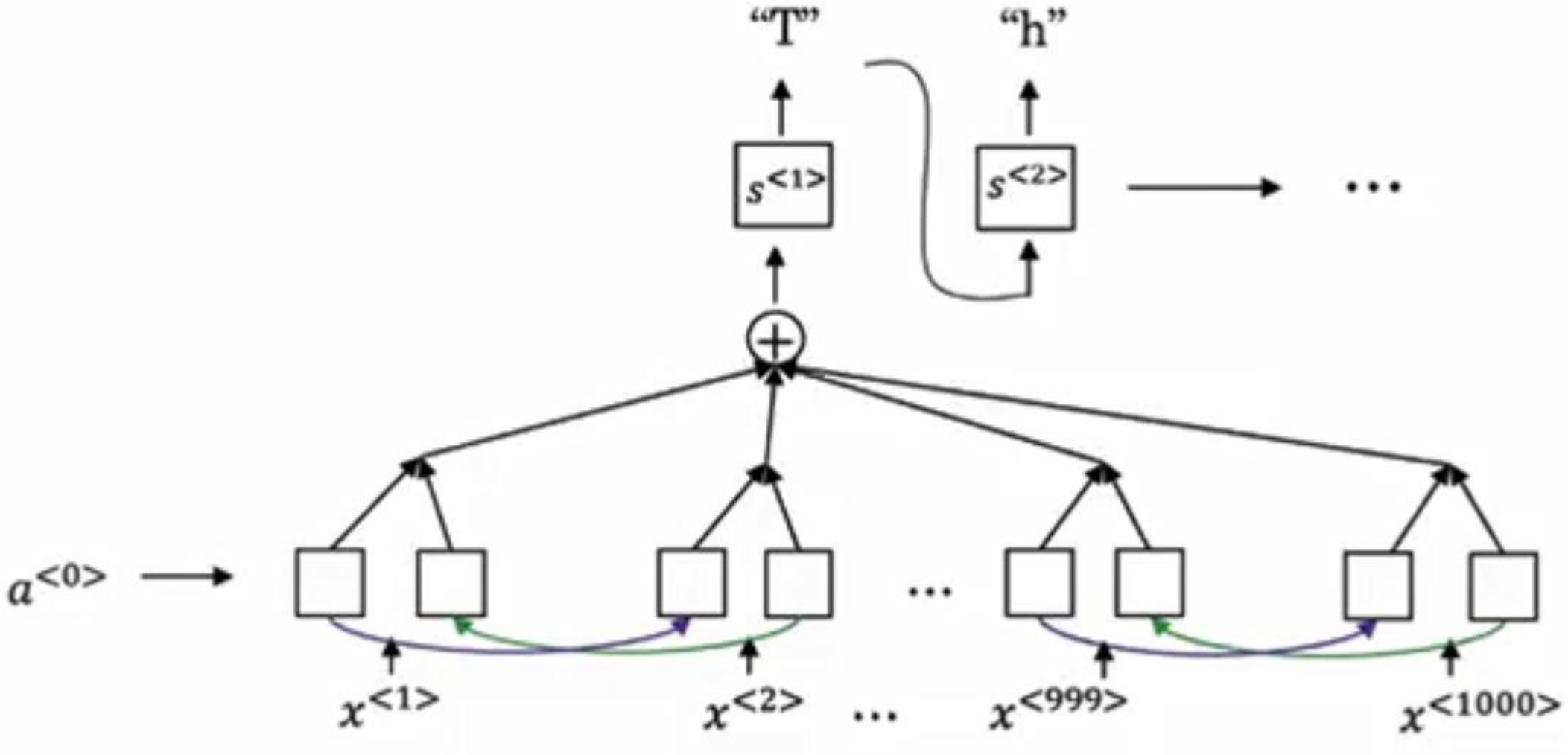
Speech Recognition



Previously we used to have Hand-engineered features consisting of different phonemes

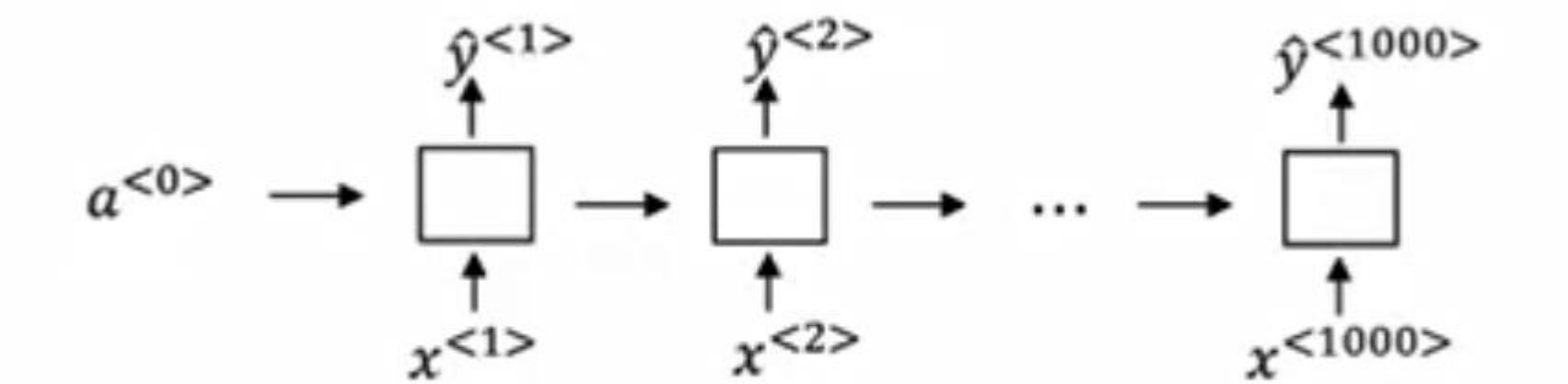
Thousands of hours of speech/audio data is used to train the data depending upon the application

Attention model for speech Systems



CTC Cost for Speech recognition

- Connectionist Temporal classification
- The Quick Brown Fox



- One of the issues in Speech Systems is that for 1000s of inputs your speech translation may actually produce only a few words. The reason is that the time frame we process to consider a individual sound unit is small as compared to the actual spoken words.So it is important to map the 1000s of outputs to a few words.
- E.g. ttt_h_eee____ ___ ␣_ _ _ _ _qqqq _ _ _ _ _
- ␣represents space and _ are blanks

Questions

Thanks