Deep Learning - COSC2779

Sequential Data - Applications

Dr. Ruwan Tennakoon

RMIT UNIVERSITY

Sep 13, 2021

Reference: *Chapter 10: Ian Goodfellow et. al., "Deep Learning", MIT Press, 2016.*

Outline

Revision

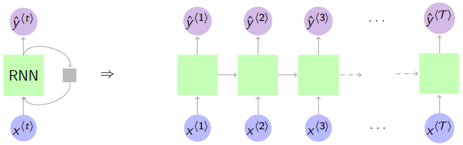
Sequential Data:

- Data points with variable length.
- Order of the data matters
- Shared feature across time is useful.

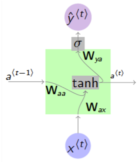
Revision

Sequential Data:

- Data points with variable length.
- Order of the data matters
- Shared feature across time is useful.



$\hat{y}^{\langle t \rangle}$
RNN
$x^{\langle t \rangle}$
$\Rightarrow$
$\hat{y}^{\langle 1 \rangle}$ $\hat{y}^{\langle 2 \rangle}$ $\hat{y}^{\langle 3 \rangle}$ ... $\hat{y}^{\langle T \rangle}$
$x^{\langle 1 \rangle}$ $x^{\langle 2 \rangle}$ $x^{\langle 3 \rangle}$ ... $x^{\langle T \rangle}$

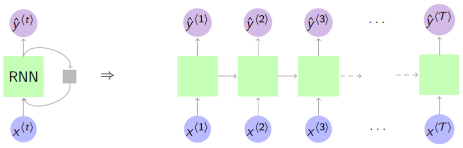


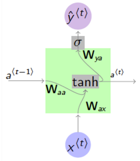
Simple RNN Cell

Revision

Sequential Data:

- Data points with variable length.
- Order of the data matters
- Shared feature across time is useful.

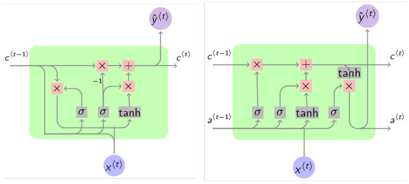




Simple RNN Cell

Improvements:

- Capture long range dependencies:



GRU LSTM

- Bi-Directional: Use both historical and future information.
- Deep RNN: Increase model capacity.

Objectives of the lecture

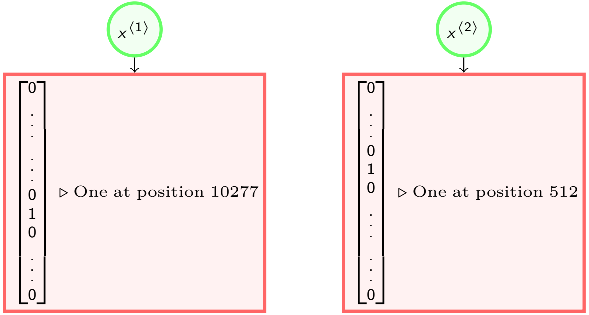
- Understand how RNNs are used in key application areas:
  - Text Classification.
  - Machine Translation.
  - Speech Recognition.
- Will cover the fundamental concepts that will enable you to explore the state-of-the-art.

Outline

Outline

One-Hot Representation

	My	experience	so	far	has	been	fantastic
	10277	512	12011	611	854	325	625
$\mathbf{x}^{(i)} =$	$x^{\langle 1\rangle}$	$x^{\langle 2\rangle}$	$x^{\langle 3\rangle}$	$x^{\langle 4\rangle}$	$x^{\langle 5\rangle}$	$x^{\langle 6\rangle}$	$x^{\langle 7\rangle}$



$$x^{\langle 1\rangle} \to \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad x^{\langle 2\rangle} \to \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix}$$

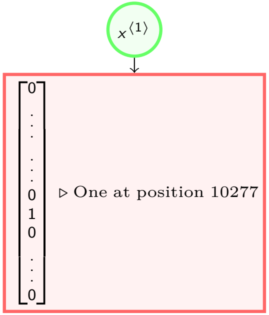
$x^{(i)}$ is a matrix with dimensions 20,000 x 7.

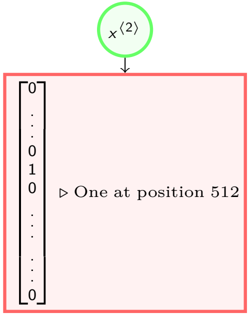
Index	**Word**
1	a
2	ability
3	able
...	...
325	been
...	...
512	experience
...	...
611	far
...	...
625	fantastic
...	...
854	has
...	...
10277	My
...	...
12011	So
...	...

Vocabulary: Assume 20k Words

One-Hot Representation

	My	experience	so	far	has	been	fantastic
	10277	512	12011	611	854	325	625
$\mathbf{x}^{(i)} =$	$x^{\langle 1 \rangle}$	$x^{\langle 2 \rangle}$	$x^{\langle 3 \rangle}$	$x^{\langle 4 \rangle}$	$x^{\langle 5 \rangle}$	$x^{\langle 6 \rangle}$	$x^{\langle 7 \rangle}$





vocabulary:

- apple: [1, 0, 0, 0, 0]
- orange: [0, 1, 0, 0, 0]
- cricket: [0, 0, 1, 0, 0]
- kabaddi: [0, 0, 0, 1, 0]
- man: [0, 0, 0, 0, 1]

$|\text{apple} - \text{orange}| = 2$
$|\text{apple} - \text{cricket}| = 2$

$x^{(i)}$ is a matrix with dimensions 20,000 x 7.

One-Hot to feature Representation

- One-Hot does not capture similarity between words.

  Seen During Training: $\boxed{\text{\underline{Joe Root} was in the English cricket team.}}$

  During inference: $\boxed{\text{\underline{Dan Murphy} was in the Australian Kabaddi squad.}}$

- If the representation for ‘cricket’ $\leftrightarrow$ ‘kabaddi’ and ‘team’ $\leftrightarrow$ ‘squad’ was close, then it is clear that “Dan Murphy” in this context is a persons name - not an organization.
- A representation that capture relationships between words can enable inferring on unseen sequences during training.

Hand-Crafted Word Embedding

	cricket	kabaddi	orange	apple	man
Gender	0.1	0.01	0	0	-1
age	0	0	0	0	1
food	0	0	1	1	0
sport	1	1	0	0	0
alive	0	0	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮
noune	1	1	1	1	1
	$e_{cricket}$	$e_{kabaddi}$	e_{orange}	e_{apple}	e_{man}

Assume we have 300 features (rows in the table above).

The vocabulary size is 20k words.

Feature values can be floating point numbers (not binary).

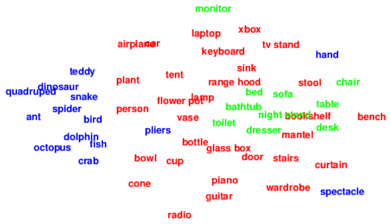
- $o_i \in \mathbb{R}^{20,000} \rightarrow$ One hot encoding of word i in vocabulary.
- $e_i \in \mathbb{R}^{300} \rightarrow$ Feature encoding of word i in vocabulary.
- $E \in \mathbb{R}^{300\times 20,000} \rightarrow$ Feature encoding of all words in vocabulary (Embedding Matrix).

$$e_i = E \cdot o_i$$

$$|e_{cricket} - e_{kabaddi}| < |e_{cricket} - e_{orange}|$$

Properties of word Embedding

Project word embedding from 300D to 2D using method like T-SNE.



Learning Word Embeddings

Learn by language modelling. E.g. Predict Next word:

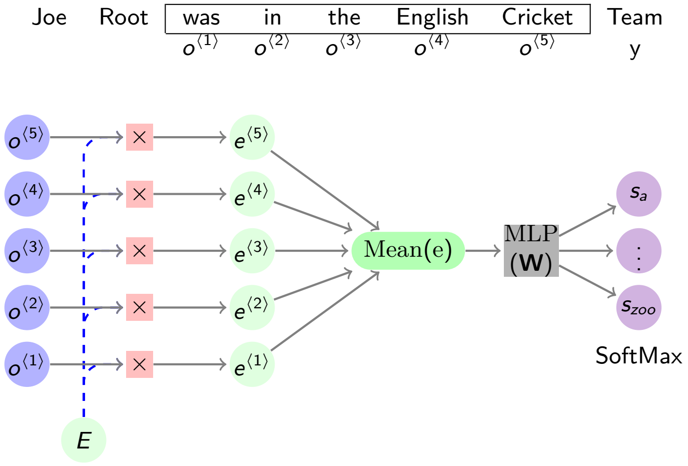
$\boxed{\text{Joe Root was in the English cricket } \underline{???}.}$

Can create a supervised learning task where **x** is a fixed length sequence (window selected by user) and y is the next word in the sequence.

Joe	Root	was	in	the	English	Cricket	Team
		$x^{\langle 1\rangle}$	$x^{\langle 2\rangle}$	$x^{\langle 3\rangle}$	$x^{\langle 4\rangle}$	$x^{\langle 5\rangle}$	y

A Neural Probabilistic Language Model

Learning Word Embeddings



Train for parameters E and the weights of the MLP (**W**) via back-propagation.

Learning Word Embeddings

Joe	Root	was	in	the	English	Cricket	Team
		$x^{\langle 1\rangle}$	$x^{\langle 2\rangle}$	$x^{\langle 3\rangle}$	$x^{\langle 4\rangle}$	$x^{\langle 5\rangle}$	y
				context			target

Context can be:

- Last 5 words.
- 5 words before and 5 words after.
- Last one word.
- Near by word.

Word2Vec and GloVe are more recent practical word embedding learning techniques.

Word2Vec

The *skip-grams*:

- Randomly pick a word as context, c.
- Randomly pick another word within a window of c as the target, t.

"Joe Root was in the English Cricket Team"

Context	Target
'English'	'Team'
'Team'	'the'

Large number of Context, target pairs are picked from a large text corpus (e.g. Wikipedia)

Efficient Estimation of Word Representations in Vector Space

Word2Vec

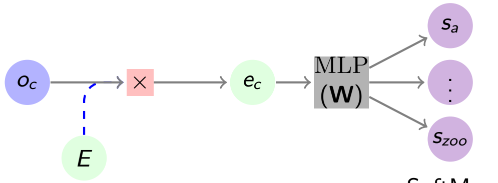
"Joe Root was in the English Cricket Team"

Context	Target
'English'	'Team'
'Team'	'cricket'
⋮	⋮

$$p(t \mid c) = \frac{e^{W^\top e_c}}{\sum_{j=1}^{N_v} e^{W^\top e_j}}$$

N_v is the vocabulary size, which is very large.

Word2Vec employs hierarchical SoftMax to make it efficient.



SoftMax

Word2Vec: Negative sampling

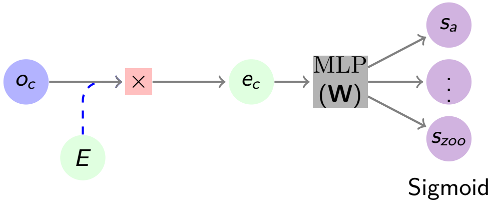
“Joe Root was in the English Cricket Team”

Context	Word	Target
‘English’	‘Team’	1
‘English’	‘apple’	0
‘English’	‘bus’	0

$$p(y = 1 \mid c, t) = \sigma \left(W^\top e_c \right)$$

Train multiple sigmoid NN instead of the SoftMax.

More recent model:
Paper: GloVe: Global Vectors for Word Representation



Outline

Sentiment Classification

"The process of computationally identifying and categorizing opinions expressed in a piece of text"

Sentiment analysis models focus on:

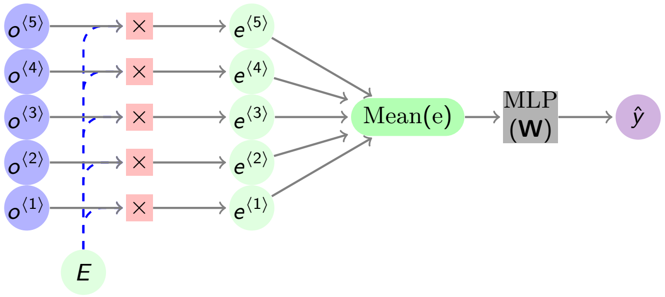
- Polarity: positive, negative, neutral.
- Feelings/Emotions: angry, happy, sad, etc.
- Intentions: interested v. not interested.

$\mathbf{x}$	y
"My experience so far has been fantastic"	Positive
"Your support team is useless"	Negative
⋮	⋮

Usually the datasets are NOT very large.

Sentiment Classification

Your	support	team	was	useless	Negative
$x^{\langle 1\rangle}$	$x^{\langle 2\rangle}$	$x^{\langle 3\rangle}$	$x^{\langle 4\rangle}$	$x^{\langle 5\rangle}$	y



Train for parameters E and the weights of the MLP (**W**) via back-propagation.

- Dataset can be too small to learn a good word embedding.
- Does not take the order of the words into account.

Transfer learning with word embedding

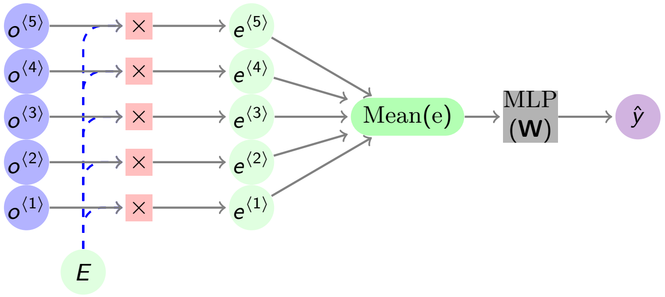
Dataset can be too small to learn a good word embedding.

Transfer learning with word embedding:

1. Learn a word embedding using a large corpus (or download a word embedding).
2. Transfer embedding to the model (Set E to the pre-learned word embedding). Learn the model parameters of the task ($\mathbf{W}$).
3. (Optional) Fine tune word embedding.

Order Matters

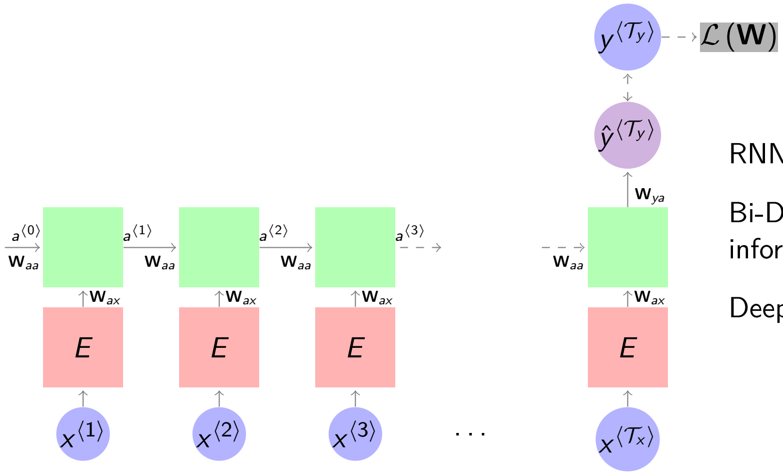
Does not take the order of the words into account.



“Completely lacks *good* food or *good* service.” $\rightarrow$?

Many-to-One RNN

"Completely lacks *good* food or *good* service." $\rightarrow$?



RNN cell can be: LSTM/GRU

Bi-Directional to capture future information.

Deep models to increase capacity.

Outline

Outline

Sequence to Sequence

Both input and output are sequences.

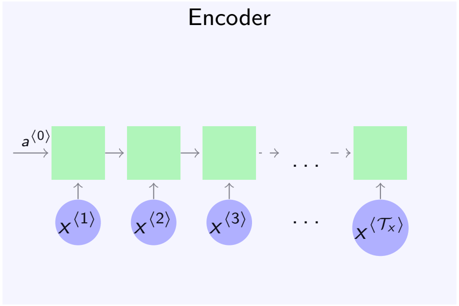
English		**Sinhala**
"How are you today"	$\Leftrightarrow$	"ada obata kohomada?"

- The two sequences are not of the same length.
- One-to-one translation does not work.

$$\begin{tikzcd} \text{ada} \arrow[d] & \text{obata} \arrow[d] & \text{kohomada} \arrow[d] \\ \text{today} & \text{you} & \text{how} \end{tikzcd}$$

Encoder-Decoder RNN

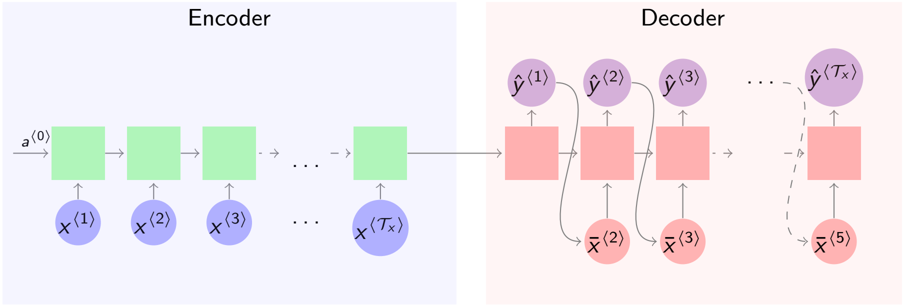
How	are	you	today	$\Leftrightarrow$	ada	obata	kohomada
$x^{\langle 1\rangle}$	$x^{\langle 2\rangle}$	$x^{\langle 3\rangle}$	$x^{\langle 4\rangle}$		$y^{\langle 1\rangle}$	$y^{\langle 2\rangle}$	$y^{\langle 3\rangle}$



Sequence to Sequence Learning with Neural Networks

Encoder-Decoder RNN

How	are	you	today	$\Leftrightarrow$	ada	obata	kohomada
$x^{\langle 1\rangle}$	$x^{\langle 2\rangle}$	$x^{\langle 3\rangle}$	$x^{\langle 4\rangle}$		$y^{\langle 1\rangle}$	$y^{\langle 2\rangle}$	$y^{\langle 3\rangle}$


Encoder
Decoder
$a^{\langle 0\rangle}$
$x^{\langle 1\rangle}$
$x^{\langle 2\rangle}$
$x^{\langle 3\rangle}$
...
$x^{\langle T_x\rangle}$
$\hat{y}^{\langle 1\rangle}$
$\hat{y}^{\langle 2\rangle}$
$\hat{y}^{\langle 3\rangle}$
$\hat{y}^{\langle T_x\rangle}$
$\bar{x}^{\langle 2\rangle}$
$\bar{x}^{\langle 3\rangle}$
$\bar{x}^{\langle 5\rangle}$

Sequence to Sequence Learning with Neural Networks

Machine Translation

The decoder we use in machine translation is similar to the one used for language modelling (e.g. text generation).

In language modelling we are interested in learning:

$$p\left(y^{\langle 1\rangle}, y^{\langle 2\rangle}, \dots, y^{\langle T_y\rangle}\right)$$

In machine translation we are interested in learning a conditional model:

$$p\left(y^{\langle 1\rangle}, y^{\langle 2\rangle}, \dots, y^{\langle T_y\rangle} \mid x^{\langle 1\rangle}, x^{\langle 2\rangle}, \dots, x^{\langle T_x\rangle}\right)$$

Image Captioning

We can use the same encoder decoder architecture to do image captioning. The main difference would be that the encoder in this task is a CNN (not a RNN).

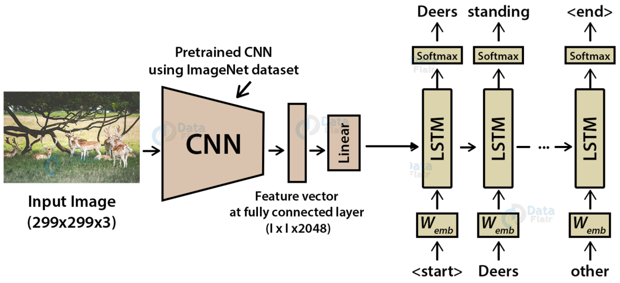

Pretrained CNN using ImageNet dataset
CNN
Input Image (299x299x3)
Feature vector at fully connected layer (1 x 1 x2048)
Linear
LSTM
Softmax
W_{emb}
Deers
standing
<end>
<start>
Deers
other

Image: Python based Project – Learn to Build Image Caption Generator with CNN & LSTM

If interested, a nice worked example at: Python based Project – Learn to Build Image Caption Generator with CNN & LSTM

Deep Captioning with Multimodal Recurrent Neural Networks (m-RNN)

Outline

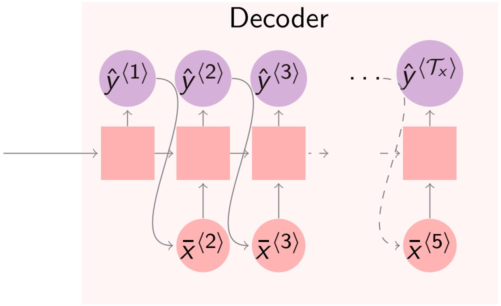
Machine Translation

In machine translation we are interested in learning a conditional model:

$$\underset{y^{\langle 1\rangle},y^{\langle 2\rangle},\dots}{\arg\max}\; p\left(y^{\langle 1\rangle}, y^{\langle 2\rangle}, \dots, y^{\langle T_y\rangle} \mid \mathbf{x}\right)$$

Option 1: **Greedy search**.

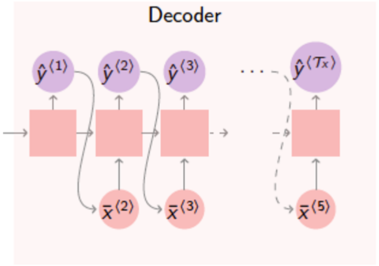
- Pick the word with maximum probability using $y^{\langle 1\rangle}$
- use it as the input to the next step
- Repeat till end of sentence.


Decoder
$\hat{y}^{\langle 1\rangle}$
$\hat{y}^{\langle 2\rangle}$
$\hat{y}^{\langle 3\rangle}$
$\hat{y}^{\langle T_x\rangle}$
$\bar{x}^{\langle 2\rangle}$
$\bar{x}^{\langle 3\rangle}$
$\bar{x}^{\langle 5\rangle}$

Problem with Greedy Search

Sinhalese sentence: "Ruwan November mase lankavata yanwa"
Possible English translations:

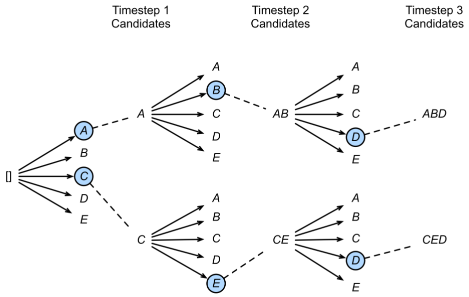
- Ruwan is visiting Sri Lanka in November.
- Ruwan is going to be visiting Sri Lanka in November.


Decoder
$\hat{y}^{\langle 1 \rangle}$
$\hat{y}^{\langle 2 \rangle}$
$\hat{y}^{\langle 3 \rangle}$
$\hat{y}^{\langle T_x \rangle}$
$\bar{x}^{\langle 2 \rangle}$
$\bar{x}^{\langle 3 \rangle}$
$\bar{x}^{\langle 5 \rangle}$

The first sentence is a better translation, but the greedy search might pick the second one because going is more common than visiting.

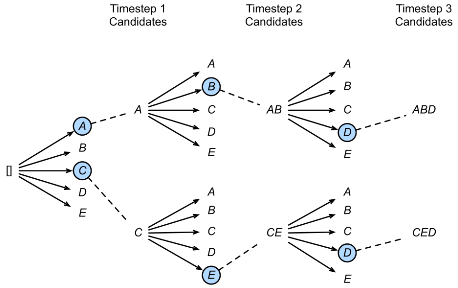
Beam Search

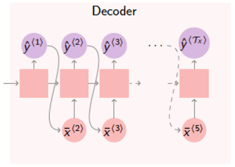
Instead of picking one candidate at each time step, you can pick best k candidates. Here the k is known as the beam width.

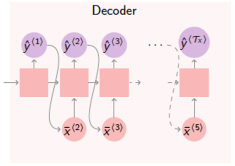


Beam Search

At each time-step you need k copies of the network (k=2 in example).




Decoder
$\hat{y}^{\langle 1 \rangle}$
$\hat{y}^{\langle 2 \rangle}$
$\hat{y}^{\langle 3 \rangle}$
$\hat{y}^{\langle T_x \rangle}$
$\bar{x}^{\langle 2 \rangle}$
$\bar{x}^{\langle 3 \rangle}$
$\bar{x}^{\langle 5 \rangle}$


Decoder
$\hat{y}^{\langle 1 \rangle}$
$\hat{y}^{\langle 2 \rangle}$
$\hat{y}^{\langle 3 \rangle}$
$\hat{y}^{\langle T_x \rangle}$
$\bar{x}^{\langle 2 \rangle}$
$\bar{x}^{\langle 3 \rangle}$
$\bar{x}^{\langle 5 \rangle}$

- Ruwan is visiting Sri Lanka in November.
- Ruwan is going to be visiting Sri Lanka in November.

Outline

Why Attention?

Assume you are given the following paragraph to be translate to another language:

"*Seq2seq is a family of machine learning approaches used for language processing. Applications include language translation, image captioning, conversational models and text summarization.*"

Would you:

- Read the complete paragraph and then start translating
- Translate one small part at a time.

Why Attention?

Assume you are given the following paragraph to be translate to another language:

"*Seq2seq is a family of machine learning approaches used for language processing. Applications include language translation, image captioning, conversational models and text summarization.*"

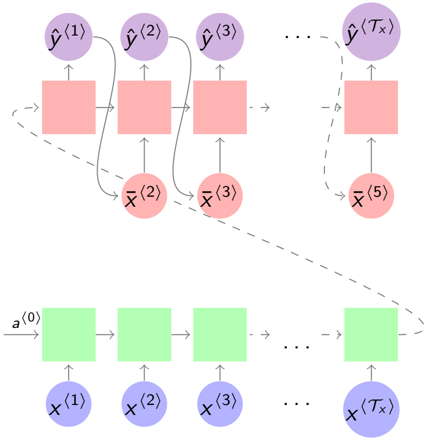
Would you:

- Read the complete paragraph and then start translating
- Translate one small part at a time.

How to decide what part to be looked at in making the decision?

Attention

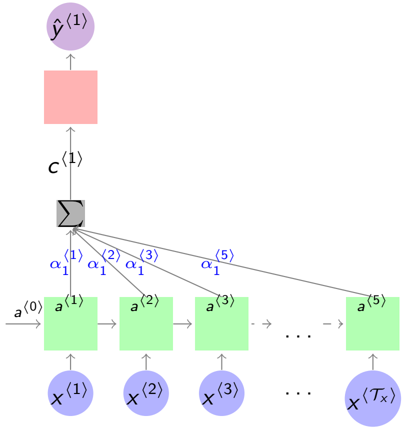
Use a small neural network to decide the attention in making each decision.

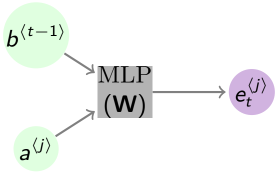


$\hat{y}^{\langle 1 \rangle}$
$\hat{y}^{\langle 2 \rangle}$
$\hat{y}^{\langle 3 \rangle}$
$\hat{y}^{\langle T_x \rangle}$
$\bar{x}^{\langle 2 \rangle}$
$\bar{x}^{\langle 3 \rangle}$
$\bar{x}^{\langle 5 \rangle}$
$a^{\langle 0 \rangle}$
$x^{\langle 1 \rangle}$
$x^{\langle 2 \rangle}$
$x^{\langle 3 \rangle}$
$x^{\langle T_x \rangle}$

Machine translation with regular RNN.

Attention





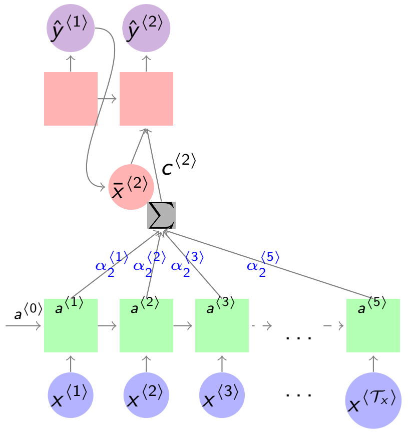
$$\alpha_t^{\langle j \rangle} = \frac{\exp(e_t^{\langle j \rangle})}{\sum_j \exp(e_t^{\langle j \rangle})}$$

$b^{\langle t-1 \rangle}$ is the cell state from the previous time step of the decoder.

Neural Machine Translation by Jointly Learning to Align and Translate

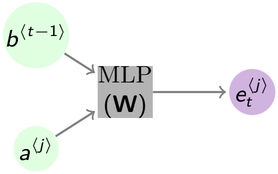
$$c^{\langle t \rangle} = \sum_j \alpha_t^{\langle j \rangle} a^{\langle j \rangle}$$

Attention



$\hat{y}^{\langle 1 \rangle}$
$\hat{y}^{\langle 2 \rangle}$
$c^{\langle 2 \rangle}$
$\bar{x}^{\langle 2 \rangle}$
$\sum$
$\alpha_2^{\langle 1 \rangle}$
$\alpha_2^{\langle 2 \rangle}$
$\alpha_2^{\langle 3 \rangle}$
$\alpha_2^{\langle 5 \rangle}$
$a^{\langle 0 \rangle}$
$a^{\langle 1 \rangle}$
$a^{\langle 2 \rangle}$
$a^{\langle 3 \rangle}$
$a^{\langle 5 \rangle}$
$x^{\langle 1 \rangle}$
$x^{\langle 2 \rangle}$
$x^{\langle 3 \rangle}$
$x^{\langle T_x \rangle}$

$$c^{\langle t \rangle} = \sum_j \alpha_t^{\langle j \rangle} a^{\langle j \rangle}$$

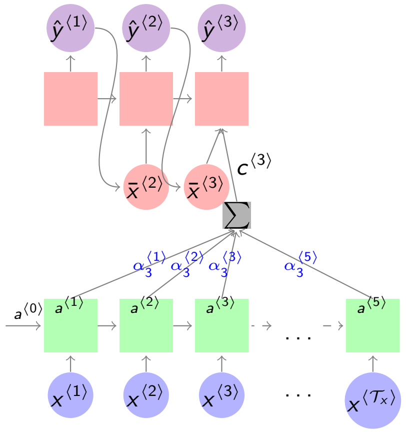


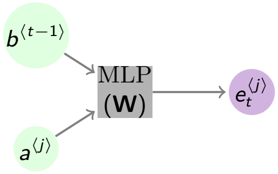
$$\alpha_t^{\langle j \rangle} = \frac{\exp(e_t^{\langle j \rangle})}{\sum_j \exp(e_t^{\langle j \rangle})}$$

$b^{\langle t-1 \rangle}$ is the cell state from the previous time step of the decoder.

Neural Machine Translation by Jointly Learning to Align and Translate

Attention


$\hat{y}^{\langle 1 \rangle}$
$\hat{y}^{\langle 2 \rangle}$
$\hat{y}^{\langle 3 \rangle}$
$\bar{x}^{\langle 2 \rangle}$
$\bar{x}^{\langle 3 \rangle}$
$c^{\langle 3 \rangle}$
$\sum$
$\alpha_3^{\langle 1 \rangle}$
$\alpha_3^{\langle 2 \rangle}$
$\alpha_3^{\langle 3 \rangle}$
$\alpha_3^{\langle 5 \rangle}$
$a^{\langle 0 \rangle}$
$a^{\langle 1 \rangle}$
$a^{\langle 2 \rangle}$
$a^{\langle 3 \rangle}$
...
$a^{\langle 5 \rangle}$
$x^{\langle 1 \rangle}$
$x^{\langle 2 \rangle}$
$x^{\langle 3 \rangle}$
...
$x^{\langle T_x \rangle}$



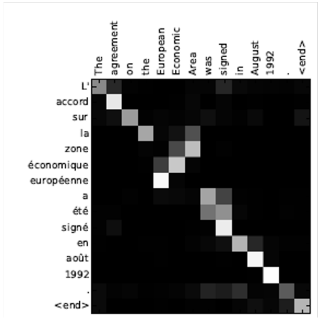
$$\alpha_t^{\langle j \rangle} = \frac{\exp(e_t^{\langle j \rangle})}{\sum_j \exp(e_t^{\langle j \rangle})}$$

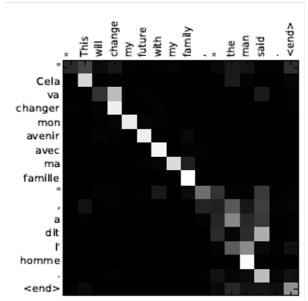
$b^{\langle t-1 \rangle}$ is the cell state from the previous time step of the decoder.

Neural Machine Translation by Jointly Learning to Align and Translate

$$c^{\langle t \rangle} = \sum_j \alpha_t^{\langle j \rangle} a^{\langle j \rangle}$$

Visualize Attention



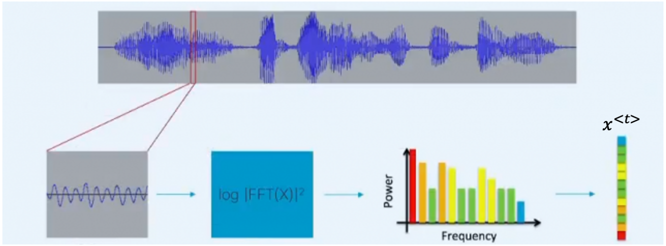


Attention weights (α) of English to French translation.

Outline

Speech Recognition

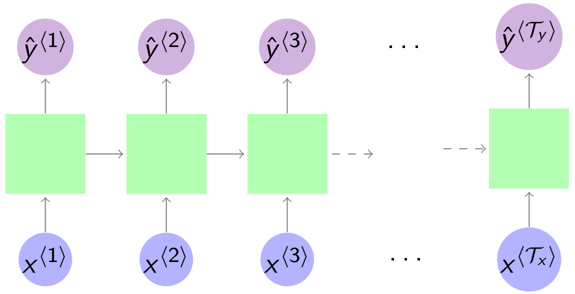
Speech signals are usually represented as a spectrogram.



$x^{\langle t \rangle}$ is a vector where number of elements equal the number of frequencies in the spectrogram.

Speech Recognition

We can use a many-to-many RNN for speech recognition (e.g. Attention model).

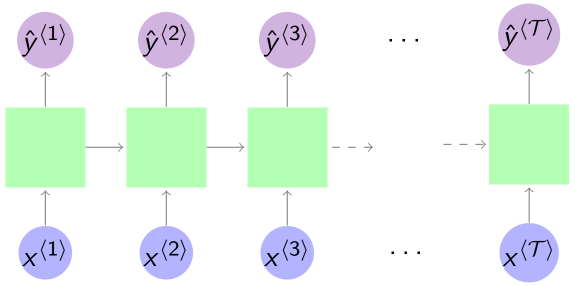


$\hat{y}^{\langle 1 \rangle}$
$\hat{y}^{\langle 2 \rangle}$
$\hat{y}^{\langle 3 \rangle}$
$\dots$
$\hat{y}^{\langle T_y \rangle}$
$x^{\langle 1 \rangle}$
$x^{\langle 2 \rangle}$
$x^{\langle 3 \rangle}$
$\dots$
$x^{\langle T_x \rangle}$

Problem: x vectors are generated at fixed time intervals (higher rate e.g. 100Hz). However the words in text are not (much lower rate). $T_y \ll T_x$

Speech Recognition

Allow multiple outputs for the same text-character.


$\hat{y}^{\langle 1 \rangle}$
$\hat{y}^{\langle 2 \rangle}$
$\hat{y}^{\langle 3 \rangle}$
$\dots$
$\hat{y}^{\langle T \rangle}$
$x^{\langle 1 \rangle}$
$x^{\langle 2 \rangle}$
$x^{\langle 3 \rangle}$
$\dots$
$x^{\langle T \rangle}$

Output of RNN: he__ll_l__oo_∼wo_rr__l_d
Collapse repeated characters not separated by "_"
CTC Output: "hello world"

Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks

Summary

- How to represents input: Text, Speech
- Main considerations in applying RNN to real-world-applications.
- Encoder-decoder networks for sequence to sequence models.

Next week: Representation learning.

Lab: Sequence to Sequence model