Deep Learning - COSC2779

Modelling Sequential (Time Series) Data

Dr. Ruwan Tennakoon

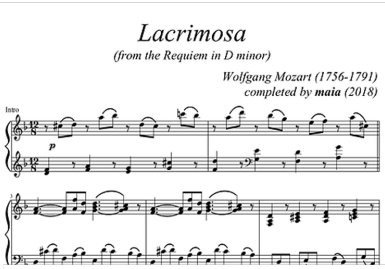
RMIT UNIVERSITY

Sep 6, 2021

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

We started out with the intention of creating an AI that could complete Mozart's unfinished composition Lacrimosa —the eighth sequence of the Requiem—which was written only till the eighth bar at the time of his passing.

We ended up creating a deep neural network called maia—play on the term 'music ai'—that can generate original piano solo compositions by learning patterns of harmony, rhythm, and style from a corpus of classical music by composers like Mozart, Chopin, and Bach.



MAIA - AI for music creation

Automated Image Captioning with ConvNets and Recurrent Nets

Andrej Karpathy, Fei-Fei Li

a young boy is holding a baseball bat

a group of people sitting at a table with wine glasses

Automated Image Captioning with ConvNets and Recurrent Nets

Andrej Karpathy, Fei-Fei Li

NeuralTalk Sentence Generation Results

Outline

Revision

A look back at what we have learned:

- Deep neural network Building blocks:
  - **Week2**: Feed forward NN Model and cost functions.
  - **Week3**: Optimising deep models challenges and solutions.
  - **Week4**: Convolution neural network: for data with spatial structure.
  -
- Case study:
  - **Week5**: Famous networks for computer vision applications
- Putting things together:
  - **Week6**: Practice methodology

A look back at what we have learned:

- Deep neural network Building blocks:
  - **Week2**: Feed forward NN Model and cost functions.
  - **Week3**: Optimising deep models challenges and solutions.
  - **Week4**: Convolution neural network: for data with spatial structure.
  -
- Case study:
  - **Week5**: Famous networks for computer vision applications
- Putting things together:
  - **Week6**: Practice methodology

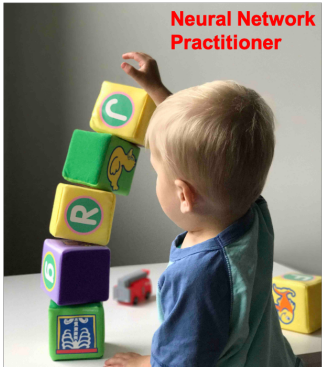


Image: Unsplash

Revision

A look back at what we have learned:

- Deep neural network Building blocks:
  - **Week2**: Feed forward NN Model and cost functions.
  - **Week3**: Optimising deep models challenges and solutions.
  - **Week4**: Convolution neural network: for data with spatial structure.
  - **Week7-8**: Recurrent neural network: for data with sequential structure.
- Case study:
  - **Week5**: Famous networks for computer vision applications
- Putting things together:
  - **Week6**: Practice methodology

Type of NN?

- Supervised learning with fixed-size vectors: Deep Feed-forward models
- Input has topological structure: CNN.
- Input or Output is a sequence: LSTM or GRU (will be discussed in future).

Objectives of this lecture

- Understand the main building blocks of RNNs designed to handle sequential data.
- Improvements to basic structure and the intuition.

Outline

Why do we need another model type?

Sentiment analysis:

"My experience so far has been fantastic" $\rightarrow$ *Positive*

"Your support team is useless" $\rightarrow$ *Negative*

How can we represent this data (for NN)?

Why do we need another model type?

Sentiment analysis:

"My experience so far has been fantastic" $\rightarrow$ *Positive*

"Your support team is useless" $\rightarrow$ *Negative*

How can we represent this data (for NN)?

- Character level: Each character mapped to a number.

Why do we need another model type?

Sentiment analysis:

"My experience so far has been fantastic" → *Positive*

"Your support team is useless" → *Negative*

How can we represent this data (for NN)?

- Character level: Each character mapped to a number.
- Word level

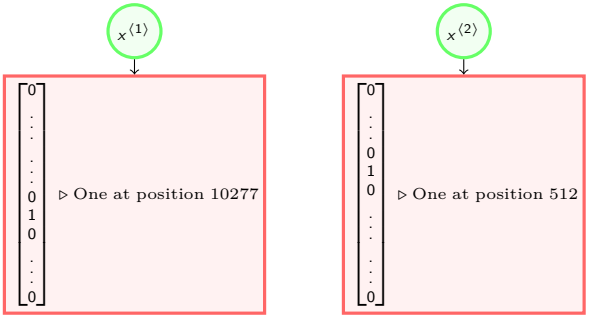
	My	experience	so	far	has	been	fantastic
	10277	512	12011	611	854	325	625
$\mathbf{x}^{(i)} =$	$x^{(i)\langle 1\rangle}$	$x^{(i)\langle 2\rangle}$	$x^{(i)\langle 3\rangle}$	$x^{(i)\langle 4\rangle}$	$x^{(i)\langle 5\rangle}$	$x^{(i)\langle 6\rangle}$	$x^{(i)\langle 7\rangle}$

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
$\mathbf{x}^{(i)} =$	$x^{(i)\langle 1\rangle}$	$x^{(i)\langle 2\rangle}$	$x^{(i)\langle 3\rangle}$	$x^{(i)\langle 4\rangle}$	$x^{(i)\langle 5\rangle}$	$x^{(i)\langle 6\rangle}$	$x^{(i)\langle 7\rangle}$



$x^{(i)}$ 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

Why do we need another model type?

Sentiment analysis:

"My experience so far has been fantastic" → *Positive*

"Your support team is useless" → *Negative*

$$\mathbf{x}^{(i)} = [\overset{\text{My}}{x^{\langle 1\rangle}}, \overset{\text{experience}}{x^{\langle 2\rangle}}, \overset{\text{so}}{x^{\langle 3\rangle}}, \overset{\text{far}}{x^{\langle 4\rangle}}, \overset{\text{has}}{x^{\langle 5\rangle}}, \overset{\text{been}}{x^{\langle 6\rangle}}, \overset{\text{fantastic}}{x^{\langle 7\rangle}}]$$

We have now represented a sentence as a vector, Can we use FF-NN?

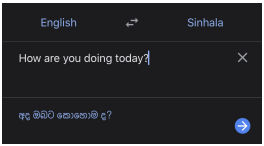
- Variable length vector.
- Context of data (elements) matters
  - You were *right*.
  - Make a *right* turn at the light.

Sequential Data: Examples

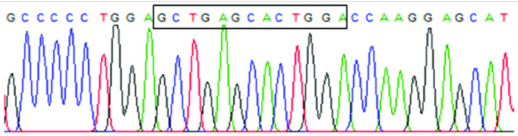
Speech recognition:

→ *"How are you"*

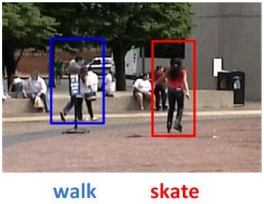
Machine translation:



DNA sequence analysis:

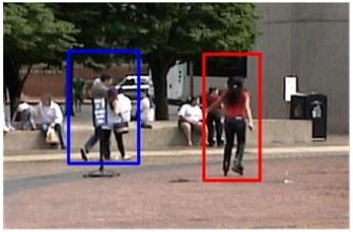


Video Activity recognition:



Sequential Data

- Data points with variable length .
- Order of the data matters. Order can be:
  - Time related: Video analysis
  - Not related to time: DNA sequence
- Shared feature across time is useful - context.



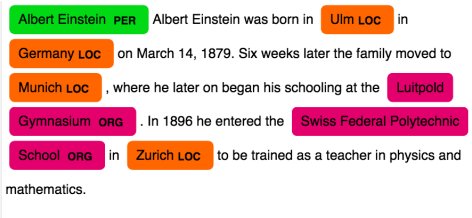
Revision Questions

- Why is word based representation preferred over character based representation in NLP?
- For a simple NLP problem, what will be the one-hot representation for a word with index 3, If the vocabulary size is 10?
- What are the main properties of sequential data?

Outline

Named Entity Recognition (NER)

"Automatically find information units like names, including person, organization and location names, and numeric expressions including time, date, money and percent expressions from unstructured text."



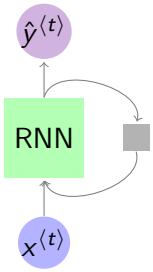
Simplified example: Only person names.

The	retired	English	test	cricketer	Mark	Butcher	is
$x^{\langle 1 \rangle}$	$x^{\langle 2 \rangle}$	$x^{\langle 3 \rangle}$	$x^{\langle 4 \rangle}$	$x^{\langle 5 \rangle}$	$x^{\langle 6 \rangle}$	$x^{\langle 7 \rangle}$	$x^{\langle 8 \rangle}$
0	0	0	0	0	1	1	0
$y^{\langle 1 \rangle}$	$y^{\langle 2 \rangle}$	$y^{\langle 3 \rangle}$	$y^{\langle 4 \rangle}$	$y^{\langle 5 \rangle}$	$y^{\langle 6 \rangle}$	$y^{\langle 7 \rangle}$	$y^{\langle 8 \rangle}$

Dictionary size: 20k

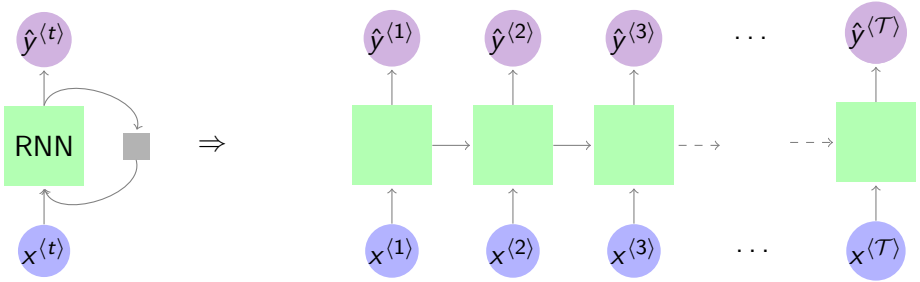
Simple solution: Represent each word as one-hot vector, predict name/not-name with FF-NN. Does not work for words like *"Mark"*, *"Butcher"*.

Recurrent neural networks have loops in them, allowing information to persist.



The	retired	English	Test	cricketer	Mark	Butcher	is
$x^{\langle 1 \rangle}$	$x^{\langle 2 \rangle}$	$x^{\langle 3 \rangle}$	$x^{\langle 4 \rangle}$	$x^{\langle 5 \rangle}$	$x^{\langle 6 \rangle}$	$x^{\langle 7 \rangle}$	$x^{\langle 8 \rangle}$
0	0	0	0	0	1	1	0

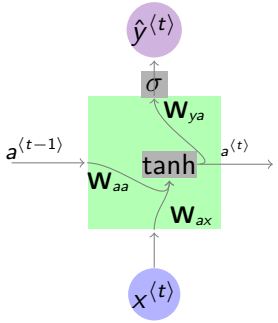
Recurrent neural networks have loops in them, allowing information to persist.



Unrolled Recurrent Neural Network

Same cell repeated. Not multiple cells.

Recurrent Neural Network: Cell



$$a^{\langle t \rangle} = g_1\left(W_{aa}a^{\langle t-1 \rangle} + W_{ax}x^{\langle t \rangle} + b_a\right)$$

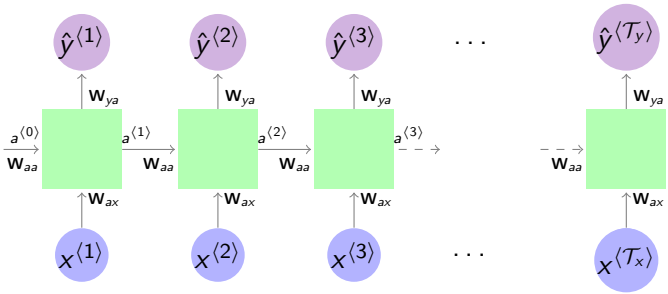
$$\hat{y}^{\langle t \rangle} = g_2\left(W_{ya}a^{\langle t \rangle} + b_y\right)$$

Assume:

- $x^{\langle t \rangle}$ is 20000-dimensional (Vocab size)
- $a^{\langle t \rangle}$ is 100-dimensional (user defined)
- $y^{\langle t \rangle}$ is 1-dimensional (output dimension)

Then weight matrices (Learned):

- $\mathbf{W}_{ax}$ is 100 x 20000
- $\mathbf{W}_{aa}$ is 100 x 100
- $\mathbf{W}_{ya}$ is 1 x 100
- b_a is 100 x 1
- b_y is 1 x 1



All $\mathbf{W}_{aa}, \mathbf{W}_{ax}, \mathbf{W}_{ya}$ are shared across RNN cells

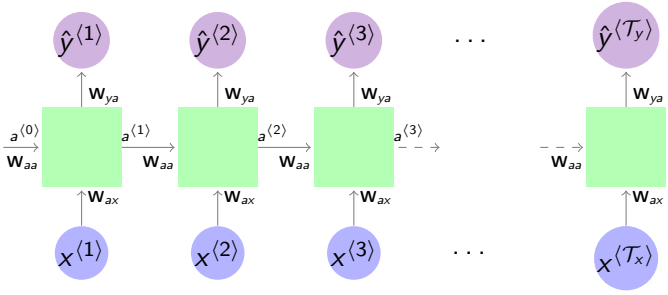
$$a^{\langle t \rangle} = g_1\left(W_{aa}a^{\langle t-1 \rangle} + W_{ax}x^{\langle t \rangle} + b_a\right)$$

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

For a data point (i):

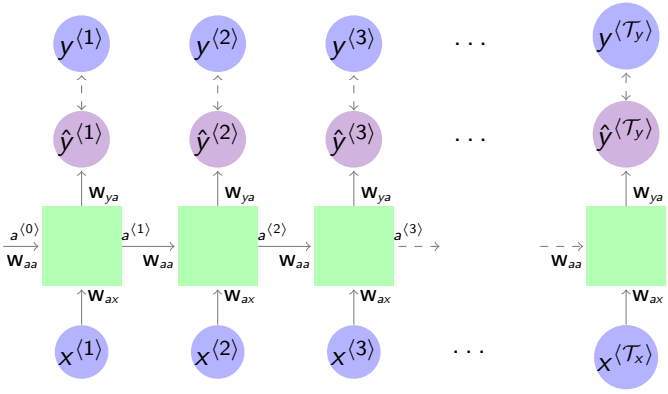
- $\mathcal{T}_x$ Number of elements in the input
- $\mathcal{T}_y$ Number of elements in the output
- $a^{\langle 0 \rangle}$ Initial state can be all zero vector.

Recurrent Neural Network: Back-propagation



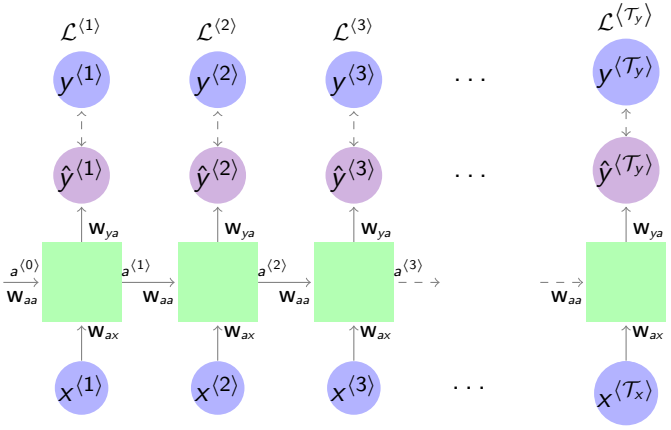
All $\mathbf{W}_{aa}, \mathbf{W}_{ax}, \mathbf{W}_{ya}$ are shared across RNN cells

Recurrent Neural Network: Back-propagation

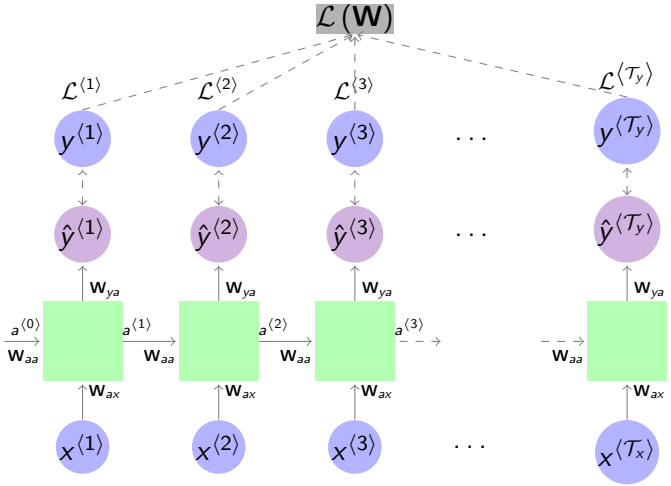


All $\mathbf{W}_{aa}, \mathbf{W}_{ax}, \mathbf{W}_{ya}$ are shared across RNN cells

Recurrent Neural Network: Back-propagation



All $\mathbf{W}_{aa}, \mathbf{W}_{ax}, \mathbf{W}_{ya}$ are shared across RNN cells



All $\mathbf{W}_{aa}, \mathbf{W}_{ax}, \mathbf{W}_{ya}$ are shared across RNN cells

$$\mathcal{L}(\mathbf{W}) = \sum_{t=1}^{\mathcal{T}_y} \mathcal{L}^{\langle t \rangle}(\mathbf{W})$$

- What values are usually used for $a^{\langle 0 \rangle}$?
- What will be an appropriate function for output activation (g_2) of a RNN if the task is regression?

Outline

There are several variants of the basic RNN depending on the shapes of inputs and outputs.

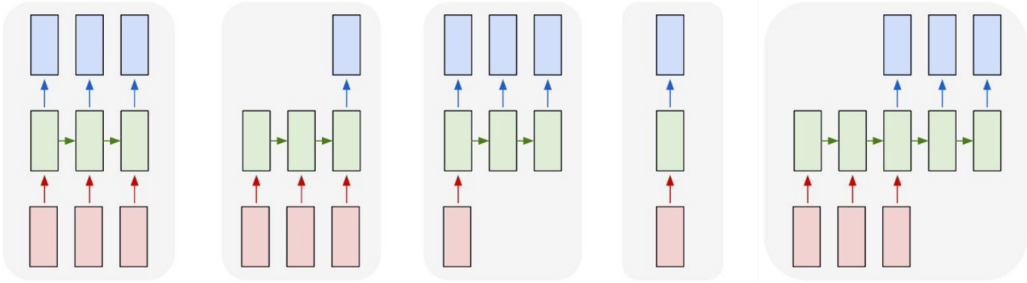


Image: The Unreasonable Effectiveness of Recurrent Neural Networks - Andrej Karpathy

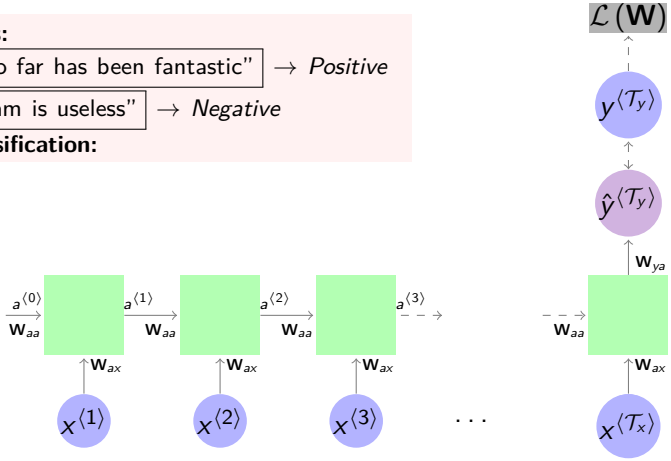
Many-to-Many ($\mathcal{T}_y = \mathcal{T}_x$): We have already discussed this - e.g. Named Entity Recognition.

Sentiment analysis:

"My experience so far has been fantastic" $\rightarrow$ *Positive*

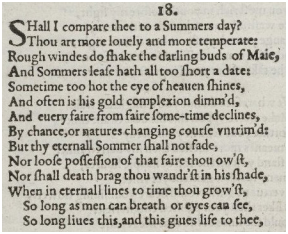
"Your support team is useless" $\rightarrow$ *Negative*

Video Action Classification:



All $\mathbf{W}_{aa}, \mathbf{W}_{ax}, \mathbf{W}_{ya}$ are shared across RNN cells

Generating Text/Music.
E.g Generating text similar to Shakespeare's writing.



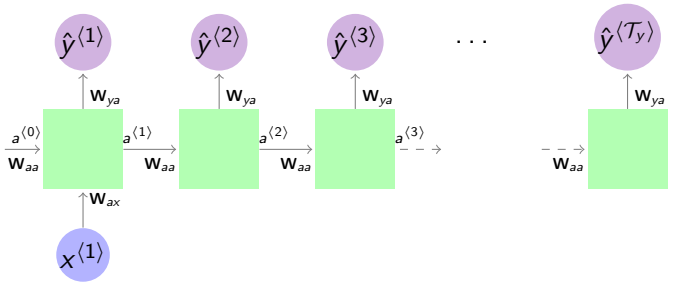
18.

SHall I compare thee to a Summers day?
Thou art more louely and more temperate:
Rough windes do shake the darling buds of Maie,
And Sommers lease hath all too short a date:
Sometime too hot the eye of heauen shines,
And often is his gold complexion dimm'd,
And euery faire from faire some-time declines,
By chance,or natures changing course vntrim'd:
But thy eternall Sommer shall not fade,
Nor loose possession of that faire thou ow'st,
Nor shall death brag thou wandr'st in his shade,
When in eternall lines to time thou grow'st,
So long as men can breath or eyes can see,
So long liues this,and this giues life to thee,

Image: Sonnet 18 in the 1609 Quarto of Shakespeare's sonnets.

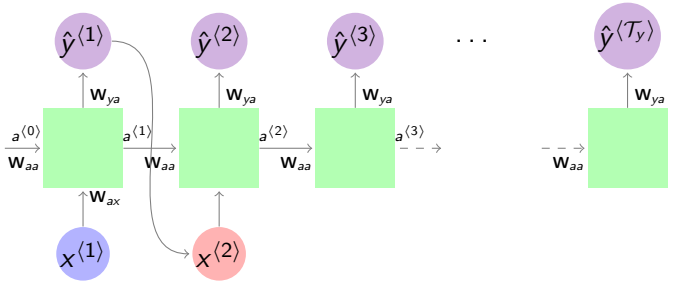
Given some text from Shakespeare's writing generate novel sentences that look similar.

Generating Shakespeare's writing.
$x^{\langle t \rangle}$ is a one-hot with size equal to number of characters.
$\hat{y}^{\langle t \rangle}$ is a soft-max-out with size equal to number of characters.



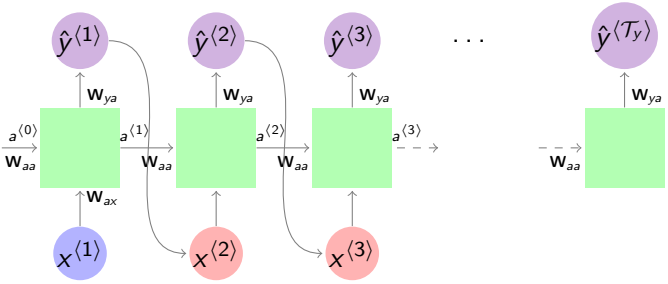
All $\mathbf{W}_{aa}, \mathbf{W}_{ax}, \mathbf{W}_{ya}$ are shared across RNN cells

Inference: Convert $\hat{y}^{\langle t \rangle}$ to one-hot by sampling and input as $x^{\langle t+1 \rangle}$

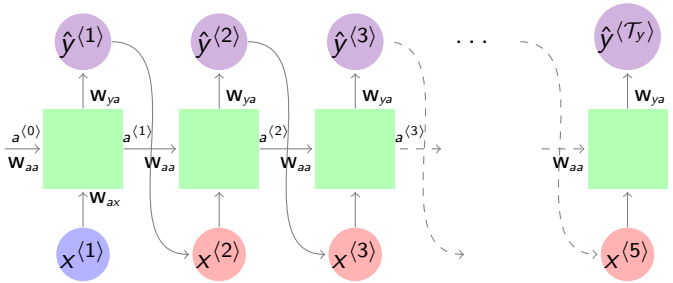


All $\mathbf{W}_{aa}, \mathbf{W}_{ax}, \mathbf{W}_{ya}$ are shared across RNN cells

One-to-Many

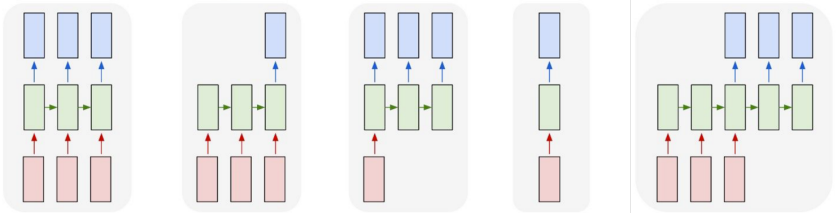


All $\mathbf{W}_{aa}, \mathbf{W}_{ax}, \mathbf{W}_{ya}$ are shared across RNN cells



All $\mathbf{W}_{aa}, \mathbf{W}_{ax}, \mathbf{W}_{ya}$ are shared across RNN cells

Many-to-Many $\mathcal{T}_y \neq \mathcal{T}_x$



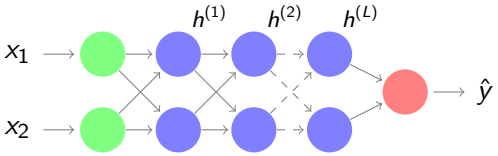
There are any situations where X, Y does not have one to one relationship.
E.g French to English translation, video captioning.

- The sequence lengths are not the same.
- Word to word translation does not work.

- You are given the task of predicting the genre of short music clips (not fixed length). What type of RNN is suitable for this task? (many-to-many, one-to-many, many-to-one, many-to-many).
- You are given the task of generating image captions. What type of RNN is suitable for this task?
- You are given the task of designing an automated exam marking system. How would RNN help you in this task?

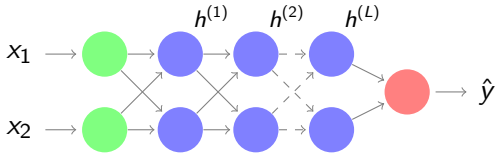
Outline

Vanishing Gradients

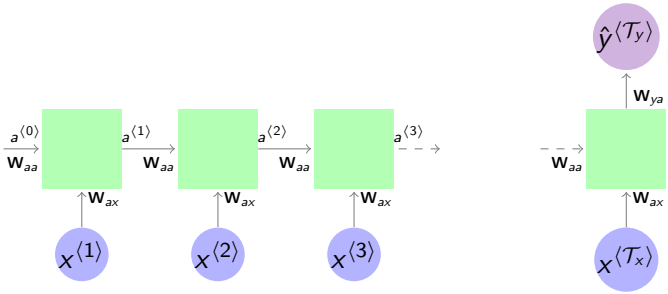


Simple Feed Forward Neural Network

- Vanishing gradient: use Skip connections, BN, ReLU, etc.
- Exploding gradient: use gradient clipping.

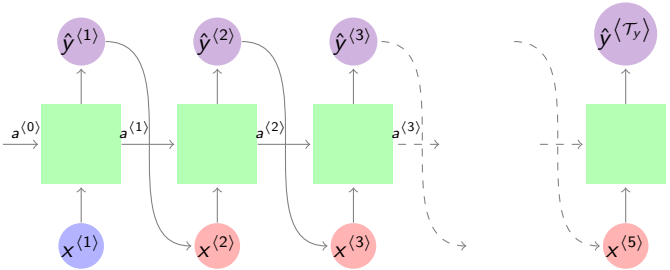


Simple Feed Forward Neural Network

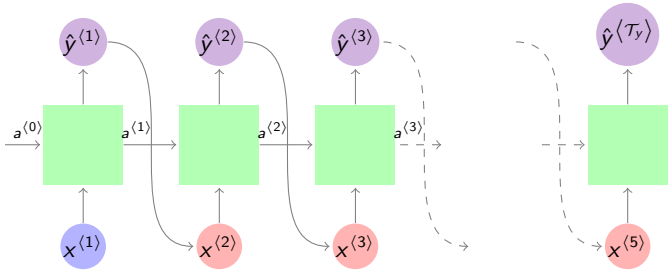


- Vanishing gradient: use Skip connections, BN, ReLU, etc.
- Exploding gradient: use gradient clipping.

Long Range Dependencies



The *cat*, who ate ..., (was/were) full.

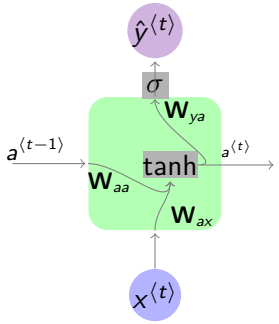


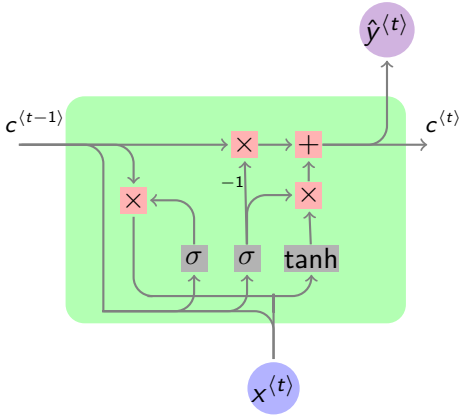
The *cat*, who ate . . . , was full.

The *cats*, who ate . . . , were full.

In theory, RNNs are absolutely capable of handling such "long-term dependencies". In practice, RNNs don't seem to be able to learn them.

Change the structure of the units





$$a^{\langle t \rangle} = g_1\left(W_{aa}a^{\langle t-1 \rangle} + W_{ax}x^{\langle t \rangle} + b_a\right)$$

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

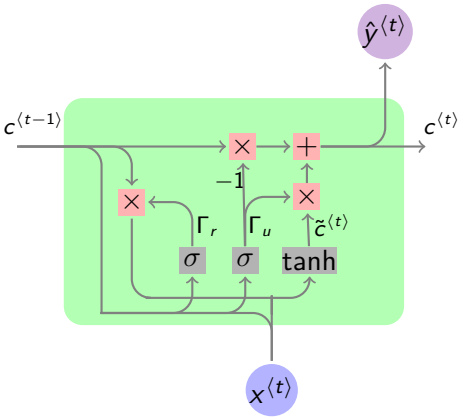
On the Properties of Neural Machine Translation: Encoder-Decoder Approaches

Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling

Gated Recurrent Units (GRU)

$$\tilde{c}^{\langle t \rangle} = \tanh\left(\mathbf{W}_c \left[c^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_c\right) \quad \triangleright \text{Cell State}$$

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



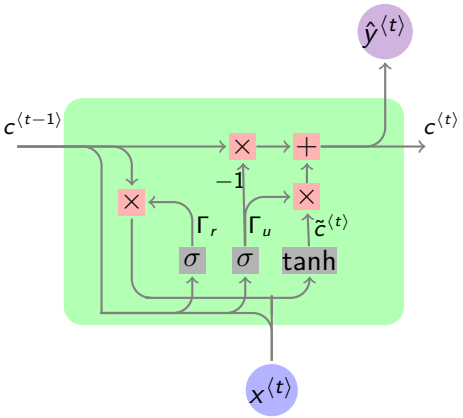
On the Properties of Neural Machine Translation: Encoder-Decoder Approaches
Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling

Gated Recurrent Units (GRU)

$$\tilde{c}^{\langle t \rangle} = \tanh\left(\mathbf{W}_c \left[c^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_c\right) \quad \triangleright \text{Cell State}$$

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

$$\Gamma_u = \sigma\left(\mathbf{W}_u \left[c^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_u\right) \quad \triangleright \text{Update gate}$$



On the Properties of Neural Machine Translation: Encoder-Decoder Approaches
Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling

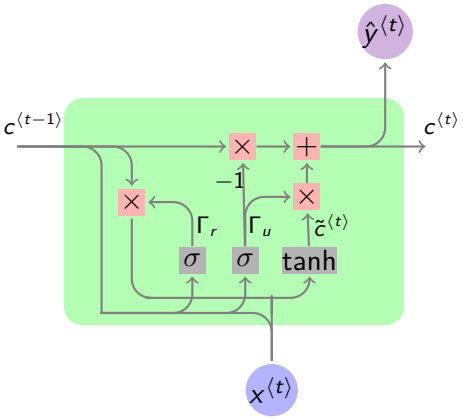
Gated Recurrent Units (GRU)

$$\tilde{c}^{\langle t \rangle} = \tanh\left(\mathbf{W}_c \left[\Gamma_r \odot c^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_c\right) \quad \triangleright \text{Cell State}$$

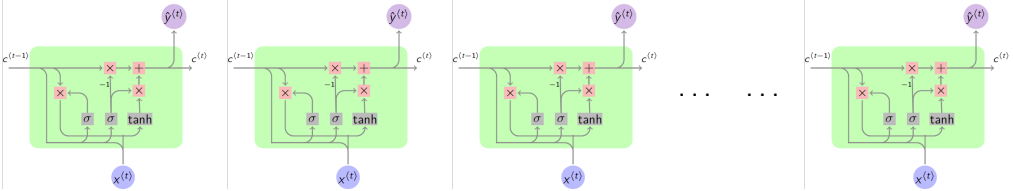
$$c^{\langle t \rangle} = (1 - \Gamma_u) \odot c^{\langle t-1 \rangle} + \Gamma_u \odot \tilde{c}^{\langle t \rangle}$$

$$\Gamma_u = \sigma\left(\mathbf{W}_u \left[c^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_u\right) \quad \triangleright \text{Update gate}$$

$$\Gamma_r = \sigma\left(\mathbf{W}_r \left[c^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_r\right) \quad \triangleright \text{Relevance gate}$$



On the Properties of Neural Machine Translation: Encoder-Decoder Approaches
Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling

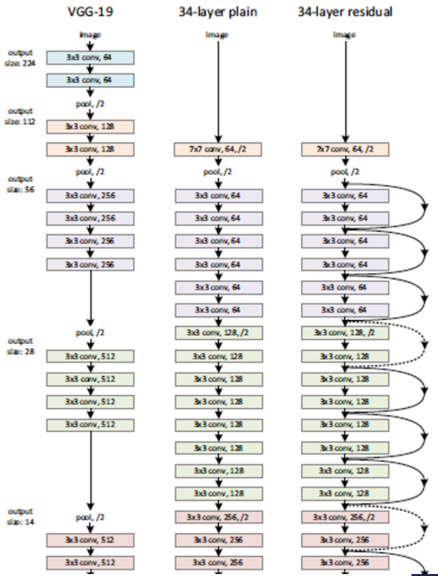


Unrolled GRU network is identical to simple-RNN except for the cell type.

Note the **horizontal line** running through the top of the diagram. like a conveyor belt, it runs straight down the entire chain, with only some minor linear interactions. It is very easy for information to just flow along it unchanged (recall ResNet and skip connections).

Gates are a way to optionally let information through.

GRU Network



VGG-19
34-layer plain
34-layer residual
image
output size: 224
3x3 conv, 64
3x3 conv, 64
pool, /2
output size: 112
3x3 conv, 128
3x3 conv, 128
pool, /2
output size: 56
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
pool, /2
output size: 28
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
pool, /2
output size: 14
3x3 conv, 512
3x3 conv, 512
image
7x7 conv, 64, /2
pool, /2
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 128, /2
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 256, /2
3x3 conv, 256
3x3 conv, 256
image
7x7 conv, 64, /2
pool, /2
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 64
3x3 conv, 128, /2
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 128
3x3 conv, 256, /2
3x3 conv, 256
3x3 conv, 256

Outline

$$\Gamma_u = \sigma\left(\mathbf{W}_u\left[a^{\langle t-1\rangle}, x^{\langle t\rangle}\right] + b_u\right) \quad \triangleright \text{Update gate}$$

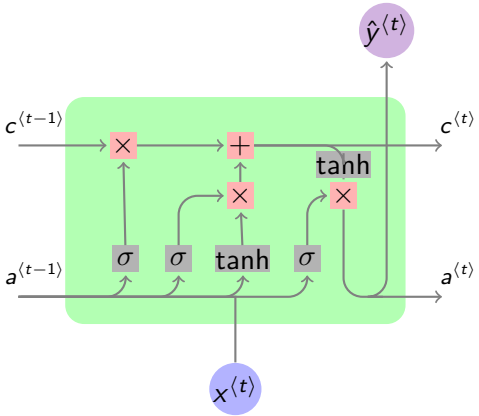
$$\Gamma_f = \sigma\left(\mathbf{W}_f\left[a^{\langle t-1\rangle}, x^{\langle t\rangle}\right] + b_f\right) \quad \triangleright \text{Forget gate}$$

$$\Gamma_o = \sigma\left(\mathbf{W}_o\left[a^{\langle t-1\rangle}, x^{\langle t\rangle}\right] + b_o\right) \quad \triangleright \text{Output gate}$$

$$\tilde{c}^{\langle t\rangle} = \tanh\left(\mathbf{W}_c\left[a^{\langle t-1\rangle}, x^{\langle t\rangle}\right] + b_c\right) \quad \triangleright \text{Cell State}$$

$$c^{\langle t\rangle} = \Gamma_f \odot c^{\langle t-1\rangle} + \Gamma_u \odot \tilde{c}^{\langle t\rangle}$$

$$a^{\langle t\rangle} = \Gamma_o \odot \tanh\left(c^{\langle t\rangle}\right)$$



Uses two gates (Γ_u,Γ_f) to make the update compared to single Γ_u in GRU.

Long-short term memory

LSTM was developed first, GRU is a simplification of the LSTM cell.

How to decide GRU vs. LSTM:

- No universally applicable rule to select one over other for a given application.
- GRU: simpler and allow much larger models.
- LSTM: More powerful and flexible - Much older than GRU and historically proven.

Either GRU or LSTM will enable the model to capture long range dependencies.

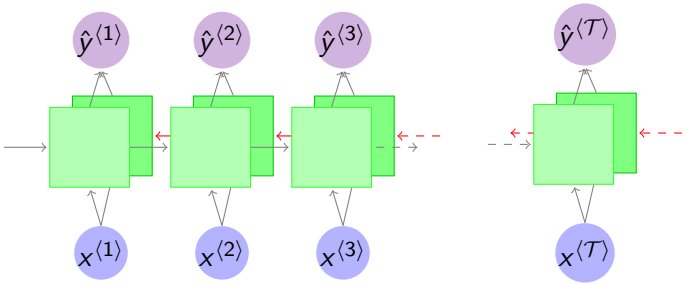
Outline

RNNs can only use information from the past to make predictions at time t.

In some cases, past information is not enough to make prediction. Future information can be helpful.

Mark	Butcher	is	a	retired	English	Test	cricketer
$x^{\langle 1 \rangle}$	$x^{\langle 2 \rangle}$	$x^{\langle 3 \rangle}$	$x^{\langle 4 \rangle}$	$x^{\langle 5 \rangle}$	$x^{\langle 6 \rangle}$	$x^{\langle 7 \rangle}$	$x^{\langle 8 \rangle}$
1	1	0	0	0	0	0	0

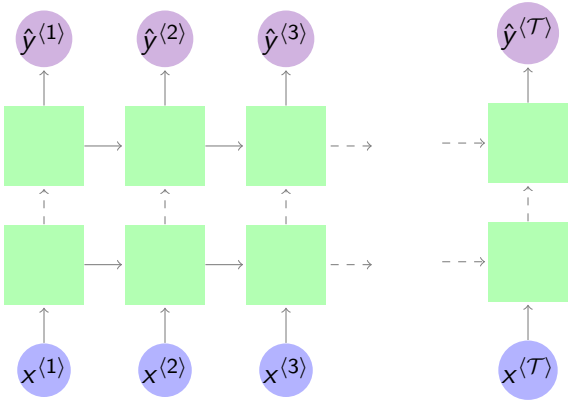
"Mark my word . . . "



Unrolled Bi-Directional Recurrent Neural Network

Two RNN. One goes from first word to last, the other goes from last to first.

Not suitable for real time systems as the future is not available for such systems.



Unrolled Recurrent Neural Network

Stack RNN cells on top of each other.

Can be expensive in terms of memory. Not as many layers as CNN.

Summary

- Sequential data: variable length, order matters, context is important.
- Recurrent neural networks have loops in them, allowing information to persist.
- Modifications to simple RNN:
  - GRU/LSTM: capture Long range dependencies.
  - Bi-Directional: Use historical and future information.
  - Deep: More complex models.

Next week: Applications of sequential data modelling

Lab: Getting started with RNN - Sentiment analysis