Recurrent Neural Network

DSE 220

Topics

- Why Sequential Models
- Mapping time into State
  - RNN unrolled in time
- RNN Equations
- Type of RNN models
- LSTM Network
  - LSTM Cell
  - LSTM Layer
- RNN Applications
  - Word Embedding
  - Sentiment Classification
  - Image Captioning
  - Neural Machine Translation
  - Attention Networks

Why Sequential models:

Speech recognition

→ “The quick brown fox jumped over the lazy dog.”

Sentiment classification

“There is nothing to like in this movie.” →

Machine translation

Voulez-vous chanter avec moi? → Do you want to sing with me?

Name entity recognition

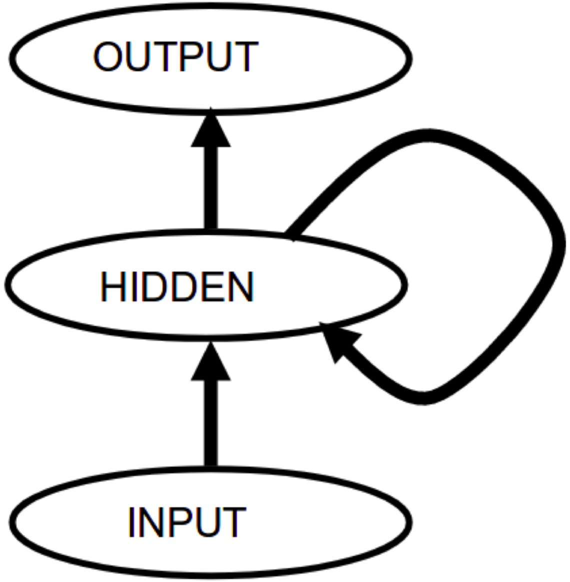
Yesterday, Harry Potter met Hermione Granger. → Yesterday, Harry Potter met Hermione Granger.

Topics

- Why Sequential Models
- Mapping time into State
    - RNN unrolled in time
- RNN Equations
- Type of RNN models
- LSTM Network
    - LSTM Cell
    - LSTM Layer
- RNN Applications
    - Word Embedding
    - Sentiment Classification
    - Image Captioning
    - Neural Machine Translation
    - Attention Networks

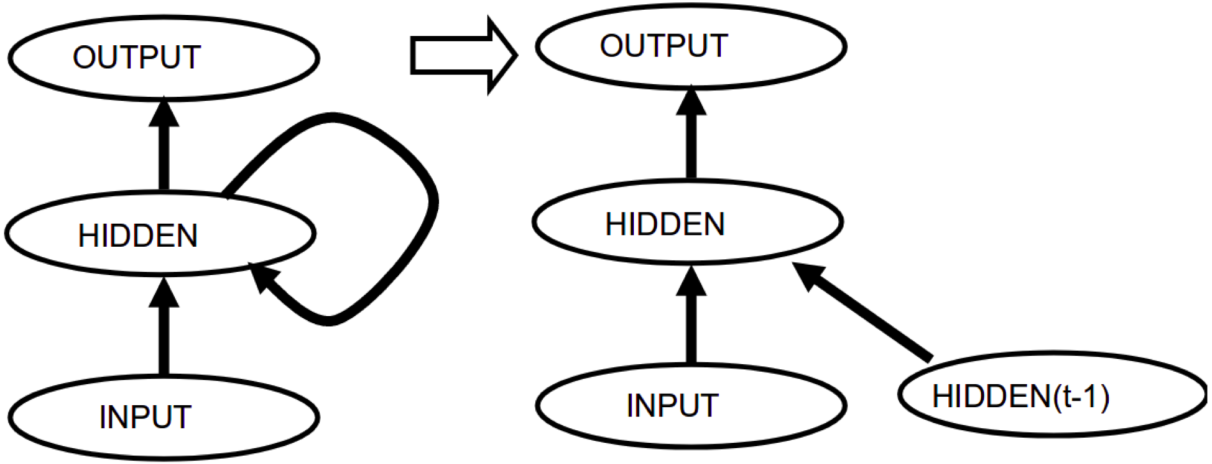
Mapping Time into State

- RNN have a memory/state.
  - It keeps a summary of historical data/past.
- Hidden state: Memory



Mapping Time into State

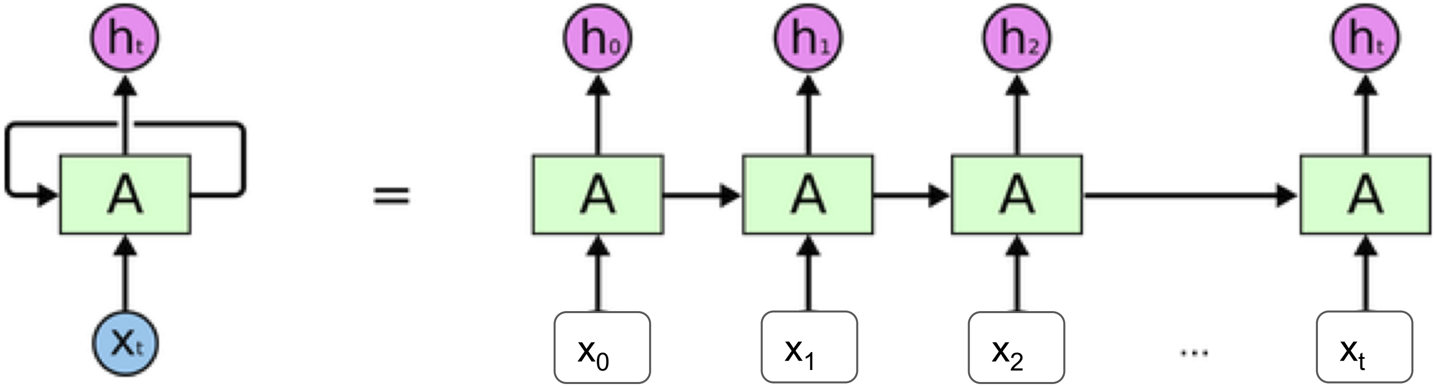
- Hidden units represent memory of network from previous inputs.



Topics

- Why Sequential Models
- Mapping time into State
    - RNN unrolled in time
- RNN Equations
- Type of RNN models
- LSTM Network
    - LSTM Cell
    - LSTM Layer
- RNN Applications
    - Word Embedding
    - Sentiment Classification
    - Image Captioning
    - Neural Machine Translation
    - Attention Networks

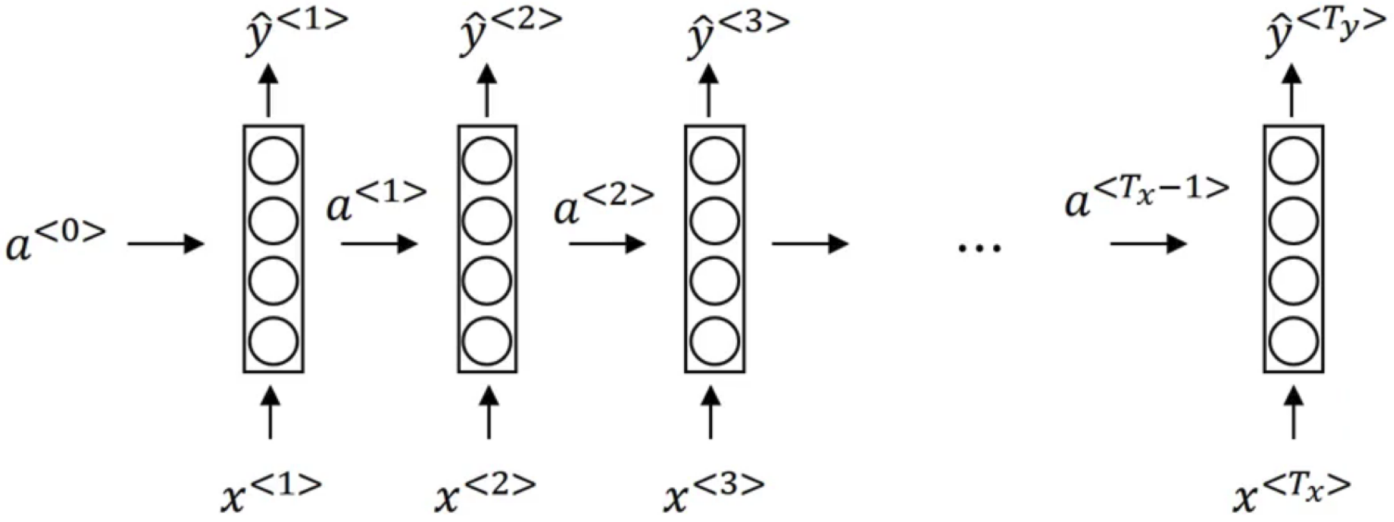
RNN Cell: Unrolled in Time



An unrolled recurrent neural network.

RNN Layer: Unrolled in Time

- A single RNN layer has multiple neurons ~ 128 - 512.



Topics

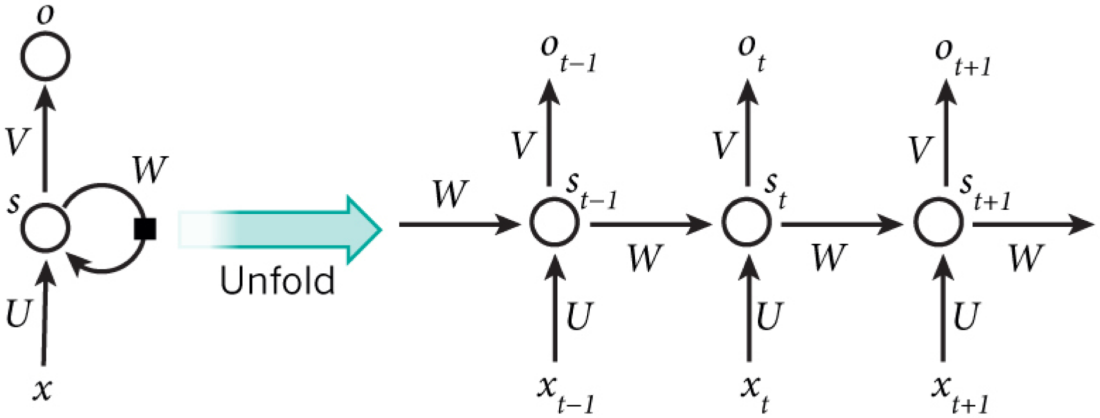
- Why Sequential Models
- Mapping time into State
  - RNN unrolled in time
- RNN Equations
- Type of RNN models
- LSTM Network
  - LSTM Cell
  - LSTM Layer
- RNN Applications
  - Word Embedding
  - Sentiment Classification
  - Image Captioning
  - Neural Machine Translation
  - Attention Networks

RNN Cell: Equation

Output Vector: $o_t = \sigma_o(V s_t + b_y)$

Hidden State: $s_t = \sigma_s(U x_t + W s_{t-1} + b_s)$

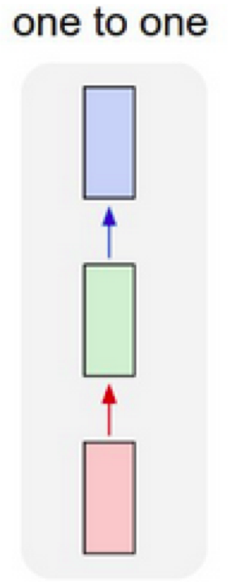
- X_t : Input vector
- S_t: hidden layer vector
- O_t: Output vector
- W,U, V, b: parameter matrices and vector
- σ_o, σ_s: Activation Functions



Topics

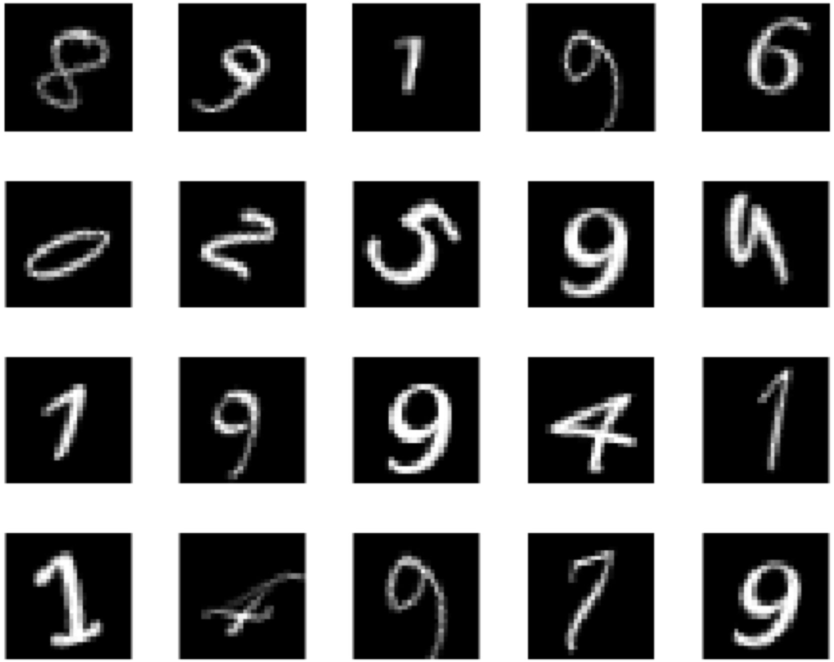
- Why Sequential Models
- Mapping time into State
  - RNN unrolled in time
- RNN Equations
- Type of RNN models
- LSTM Network
  - LSTM Cell
  - LSTM Layer
- RNN Applications
  - Word Embedding
  - Sentiment Classification
  - Image Captioning
  - Neural Machine Translation
  - Attention Networks

Type of RNN models



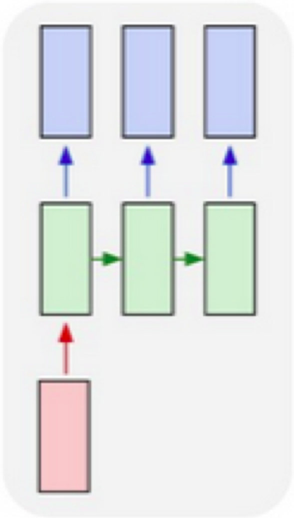
E.g: Multilayer Perceptron

Image Classification



Type of RNN models

one to many



E.g: Image Captioning

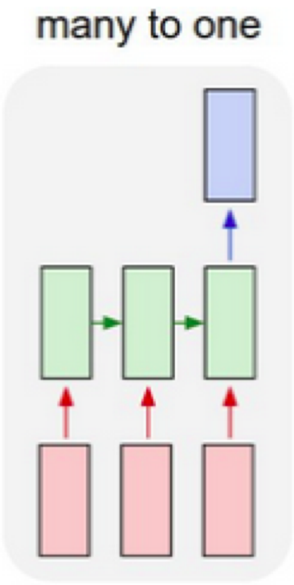
"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

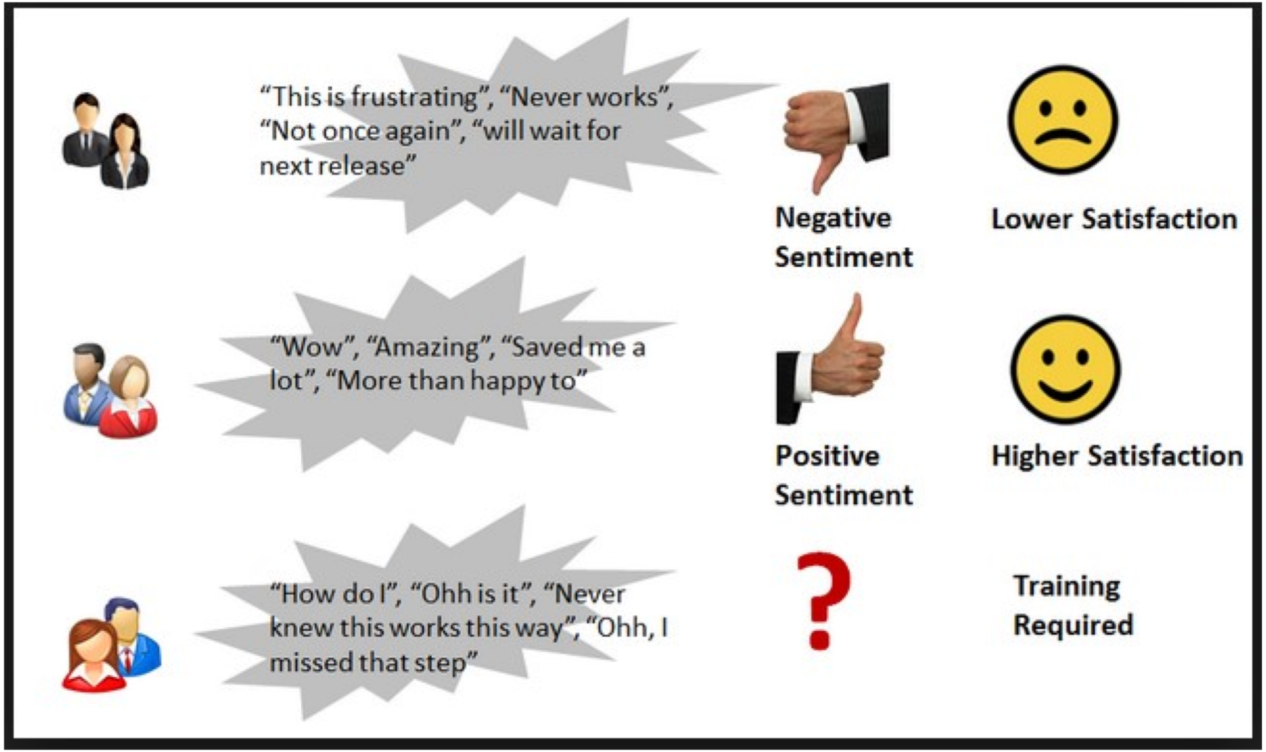
"girl in pink dress is jumping in air."

"black and white dog jumps over bar."

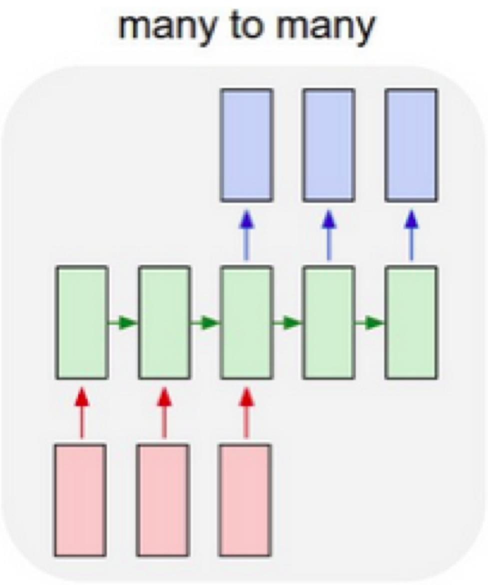
Type of RNN models



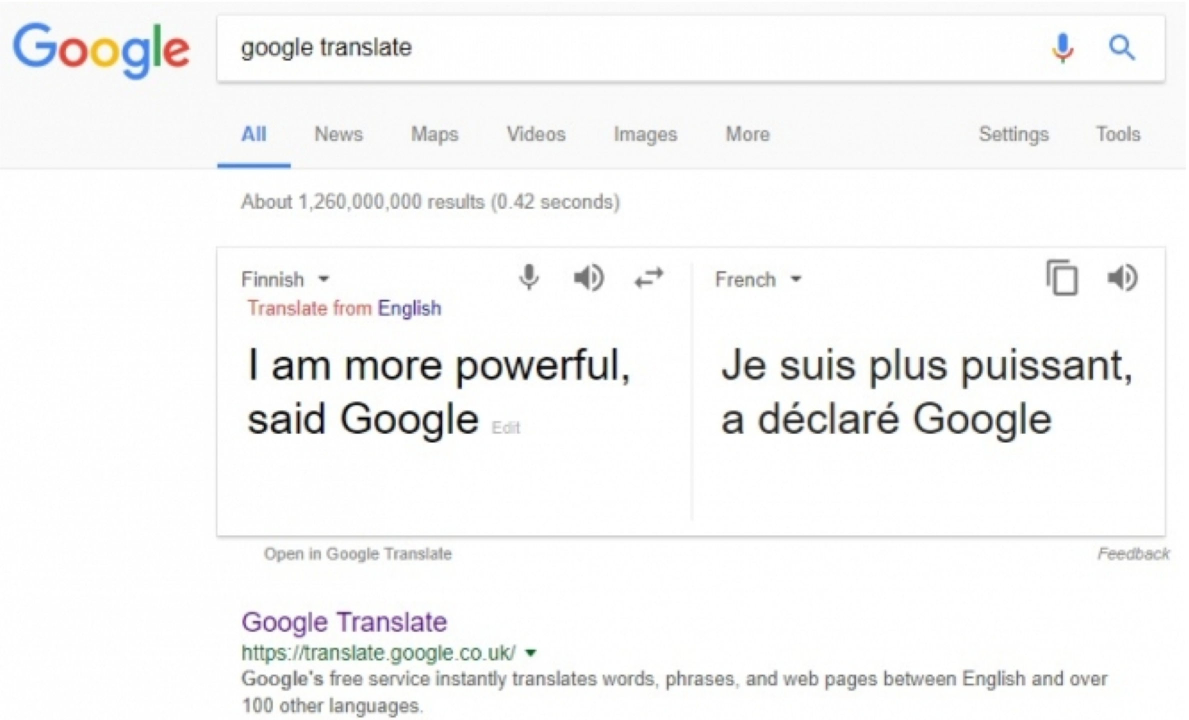
E.g: Sentiment Analysis



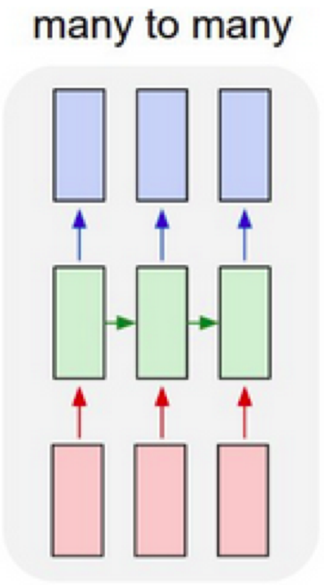
Type of RNN models



E.g: Language Translation



Type of RNN models



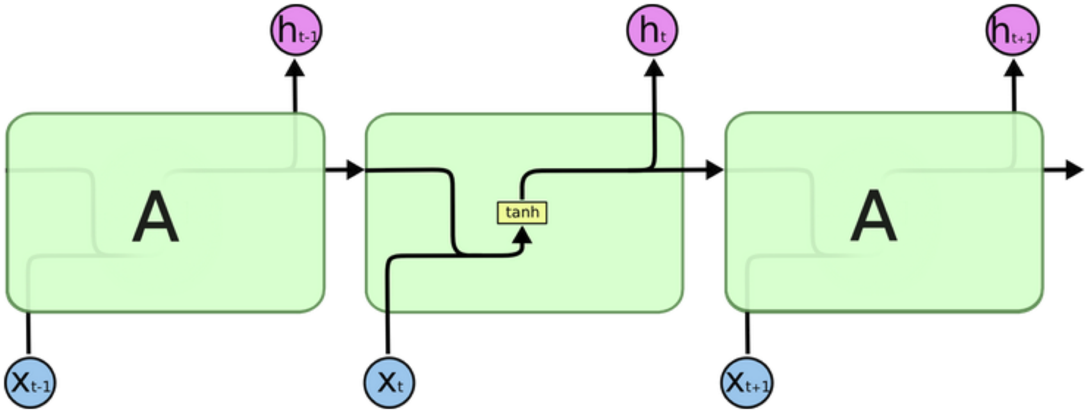
E.g: Stock Predictions

Topics

- Why Sequential Models
- Mapping time into State
  - RNN unrolled in time
- RNN Equations
- Type of RNN models
- LSTM Network
  - LSTM Cell
  - LSTM Layer
- RNN Applications
  - Word Embedding
  - Sentiment Classification
  - Image Captioning
  - Neural Machine Translation
  - Attention Networks

LSTM Network

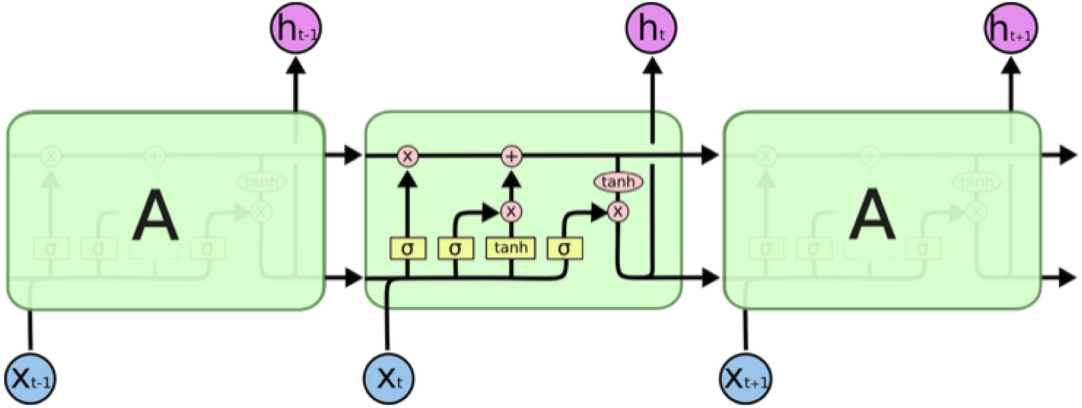
- Long short term memory
- Prevents problem of vanishing/exploding gradient



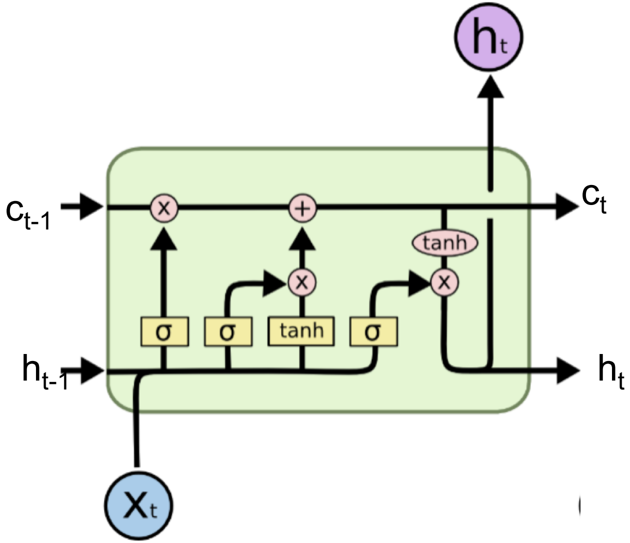
The repeating module in a standard RNN contains a single layer.

LSTM Network

- Long short term memory
- Prevents problem of vanishing/exploding gradient



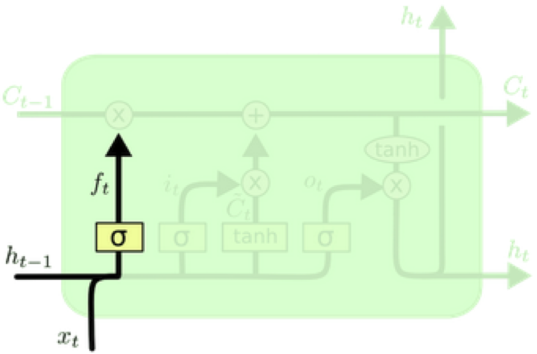
LSTM Cell



- Have two input:
  - Cell state from previous time step(c_{t-1})
  - Hidden state from previous time step(h_{t-1})
- **Forget Gate** (f_t)
  - How much we want to forget about previous cell state?
- **Input Gate**(i_t)
  - How much of current input we will save in memory.
- **Cell Update Equation**
  - Update cell state

LSTM Cell

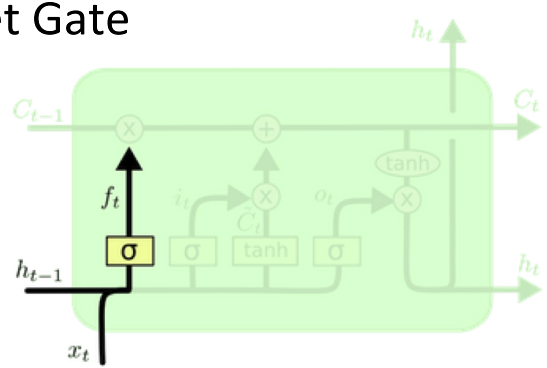
- Forget Gate



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

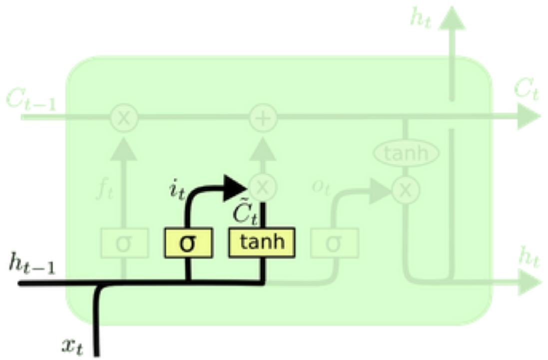
LSTM Cell

- Forget Gate



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

- Input Gate

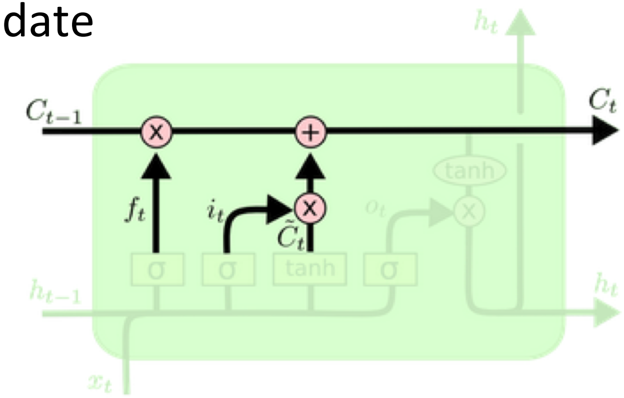


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

LSTM Cell

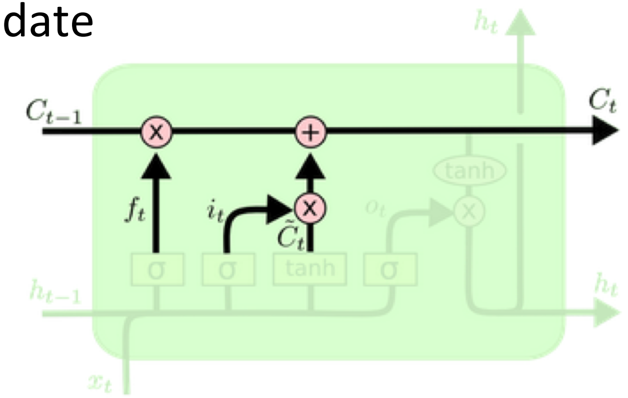
- Cell Update



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

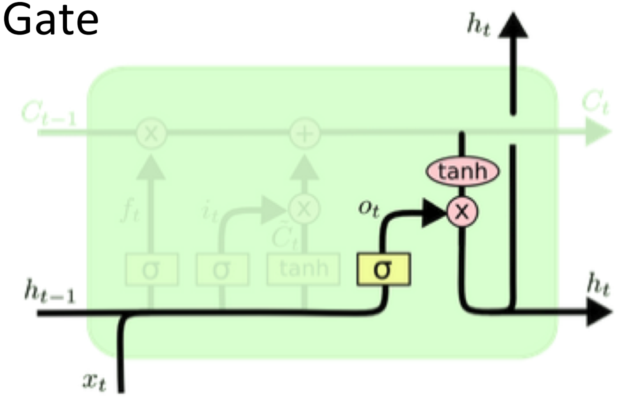
LSTM Cell

- Cell Update



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

- Output Gate



$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$

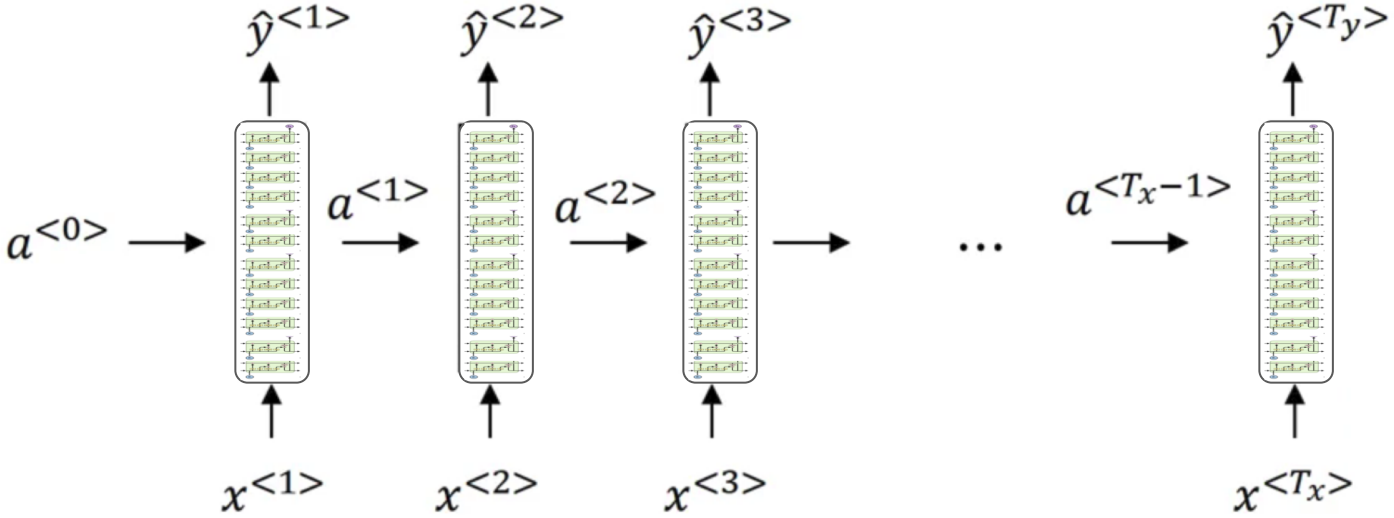
$$h_t = o_t * \tanh\left(C_t\right)$$

Topics

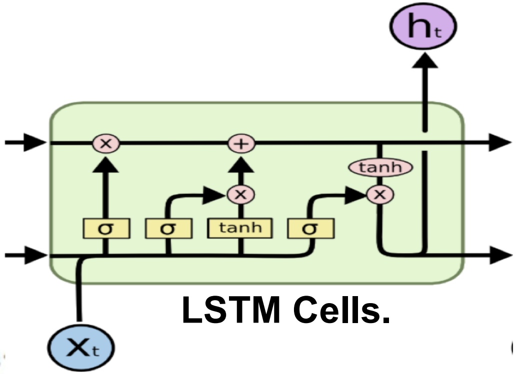
- Why Sequential Models
- Mapping time into State
  - RNN unrolled in time
- RNN Equations
- Type of RNN models
- LSTM Network
  - LSTM Cell
  - LSTM Layer
- RNN Applications
  - Word Embedding
  - Sentiment Classification
  - Image Captioning
  - Neural Machine Translation
  - Attention Networks

LSTM layer

An LSTM layer is a collection of multiple of LSTM Cells.

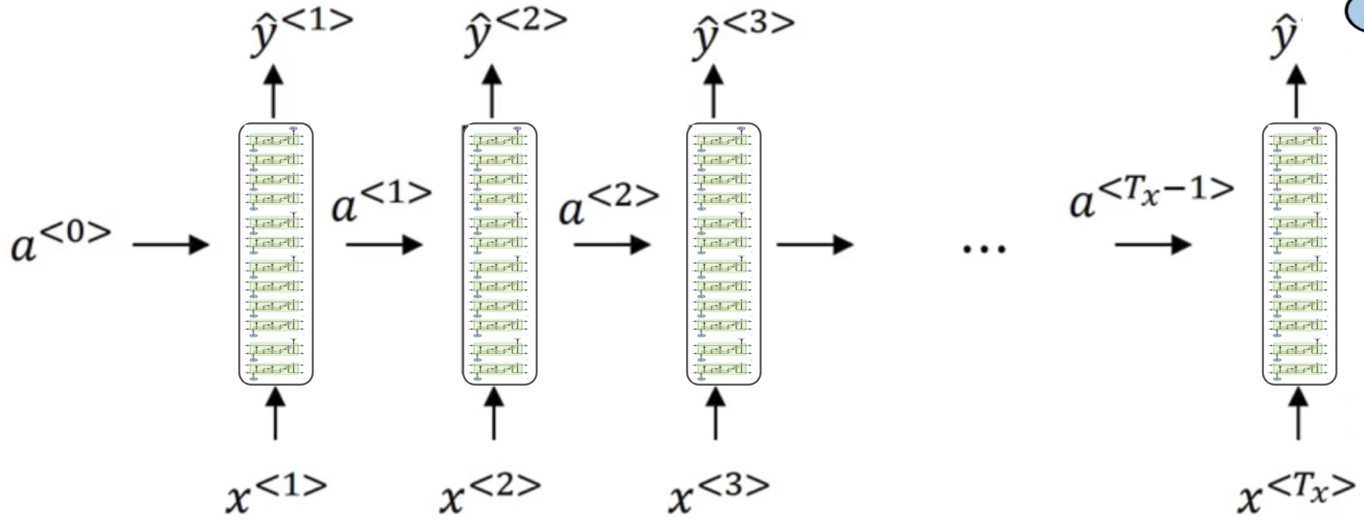


LSTM layer



LSTM Cells.

An LSTM layer is a collection of multiple of LSTM Cells.



Topics

- Why Sequential Models
- Mapping time into State
  - RNN unrolled in time
- RNN Equations
- Type of RNN models
- LSTM Network
  - LSTM Cell
  - LSTM Layer
- RNN Applications
  - Word Embedding
  - Sentiment Classification
  - Image Captioning
  - Neural Machine Translation
  - Attention Networks

Word Embedding

- Number of words in dictionary: 10000
- Words are represented in One-Hot representation of length = 10000

V = [a, aaron, ..., zulu, <UNK>]

1-hot representation

$$
\begin{array}{cccccc}
\text{Man} & \text{Woman} & \text{King} & \text{Queen} & \text{Apple} & \text{Orange} \\
(5391) & (9853) & (4914) & (7157) & (456) & (6257) \\
\begin{bmatrix}0\\0\\0\\0\\\vdots\\1\\\vdots\\0\\0\end{bmatrix} &
\begin{bmatrix}0\\0\\0\\0\\0\\\vdots\\1\\\vdots\\0\end{bmatrix} &
\begin{bmatrix}0\\0\\0\\\vdots\\1\\\vdots\\0\\0\\0\end{bmatrix} &
\begin{bmatrix}0\\0\\0\\0\\0\\\vdots\\1\\\vdots\\0\end{bmatrix} &
\begin{bmatrix}0\\\vdots\\1\\\vdots\\0\\0\\0\\0\\0\end{bmatrix} &
\begin{bmatrix}0\\0\\0\\0\\0\\\vdots\\1\\\vdots\\0\end{bmatrix}
\end{array}
$$

I want a glass of orange ______.

I want a glass of apple______.

Word Embedding

- Number of words in dictionary: 10000
- Words are represented in One-Hot representation of length = 10000
- Cons of One-hot representation:
  - Closely related word are orthogonal in the space.
  - High Embedding space(dimension)
  - Computational Complexity.

V = [a, aaron, ..., zulu, <UNK>]

1-hot representation

$$
\begin{array}{cccccc}
\text{Man} & \text{Woman} & \text{King} & \text{Queen} & \text{Apple} & \text{Orange} \\
(5391) & (9853) & (4914) & (7157) & (456) & (6257) \\
\begin{bmatrix}0\\0\\0\\0\\\vdots\\1\\\vdots\\0\\0\end{bmatrix} &
\begin{bmatrix}0\\0\\0\\0\\0\\\vdots\\1\\\vdots\\0\end{bmatrix} &
\begin{bmatrix}0\\0\\0\\\vdots\\1\\\vdots\\0\\0\\0\end{bmatrix} &
\begin{bmatrix}0\\0\\0\\0\\0\\\vdots\\1\\\vdots\\0\end{bmatrix} &
\begin{bmatrix}0\\\vdots\\1\\\vdots\\0\\0\\0\\0\\0\end{bmatrix} &
\begin{bmatrix}0\\0\\0\\0\\0\\\vdots\\1\\\vdots\\0\end{bmatrix}
\end{array}
$$

I want a glass of orange ______.

I want a glass of apple______.

Word Embedding

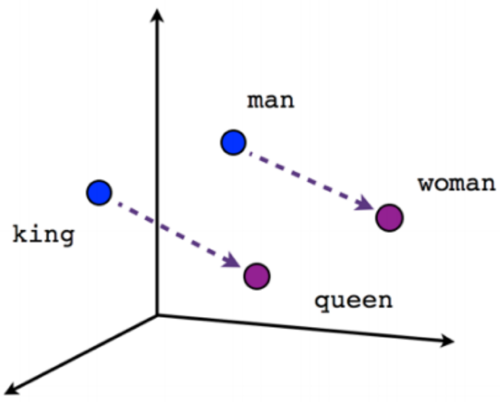
- Words Embedding already implemented in HW4
- Embed one-hot vectors into smaller dimension.
- Related words are closer in this smaller dimension.
    - Word2Vec (100, 200, 500)
    - Glove Embedding(100 - 500)
- Trained on large corpus of data ~ 100’s of Billions of text data.

Word Embedding: Interpretation

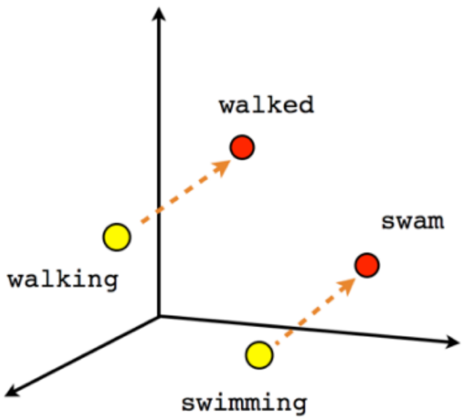
- Words are now represented in a smaller dimension of ~ 100/200

	Man	Woman	King	Queen	Apple	Orange
Gender	-1	1	-0.98	0.97	0	0.01
Royal	0.01	0.02	0.93	0.95	0.01	0.00
Age	0.02	0.01	0.7	0.69	0.03	-0.02
Food	0.01	0.01	0.02	0.01	0.95	0.97
Size						
Cost						

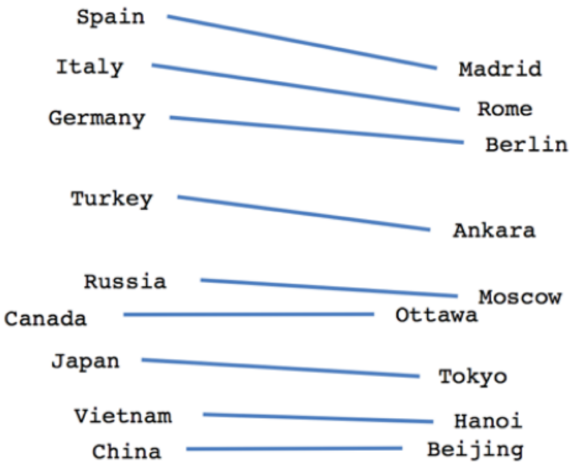
Word Embedding: Motivation



Male-Female



Verb tense



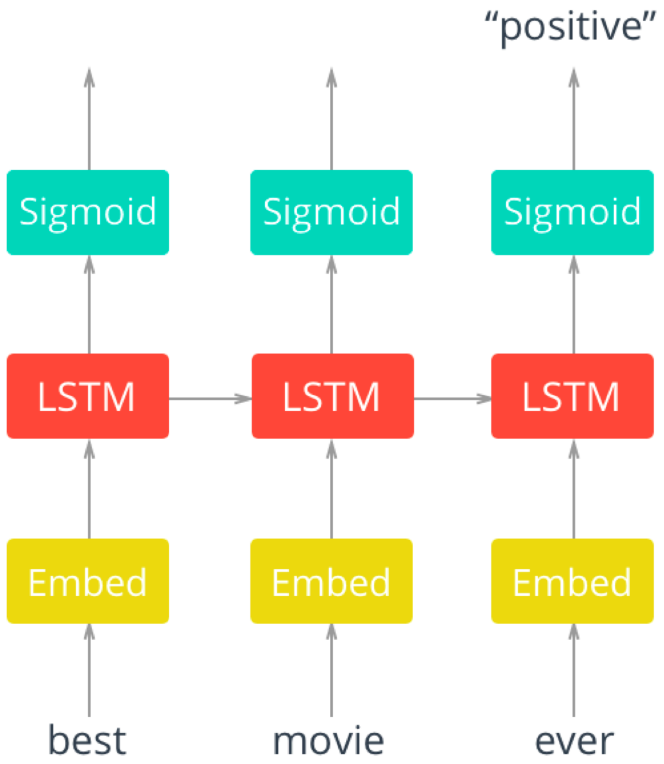
Country-Capital

- King - Man + Woman ≌ Queen
- His - He + She ≌ Her
- Paris - France + Italy ≌ Rome
- Windows - Microsoft + Google ≌ Android

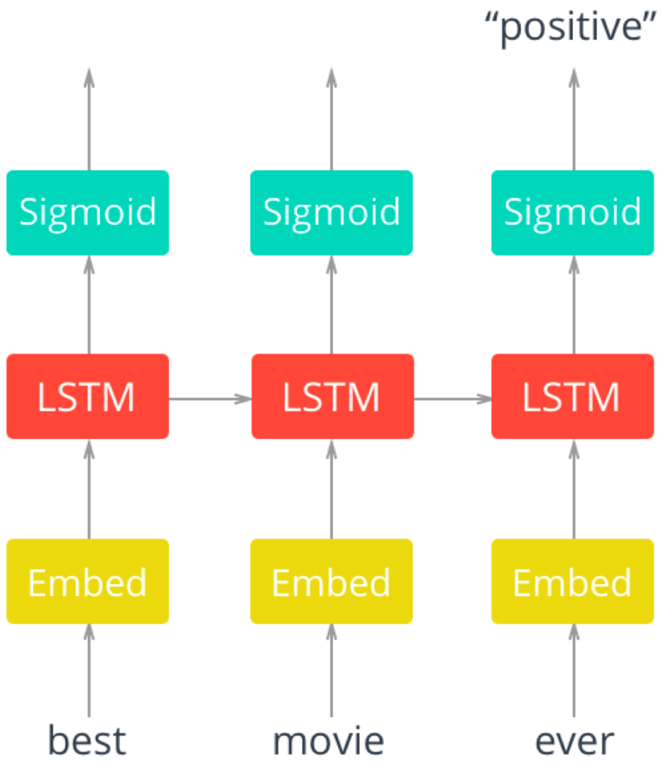
Topics

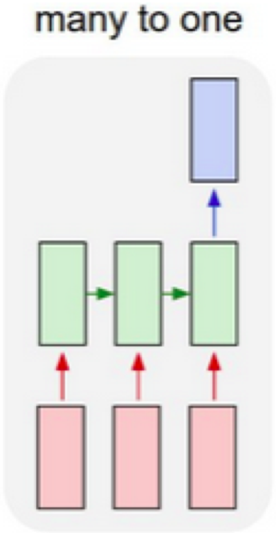
- Why Sequential Models
- Mapping time into State
  - RNN unrolled in time
- RNN Equations
- Type of RNN models
- LSTM Network
  - LSTM Cell
  - LSTM Layer
- RNN Applications
  - Word Embedding
  - Sentiment Classification
  - Image Captioning
  - Neural Machine Translation
  - Attention Networks

RNN Applications: Sentiment Classification

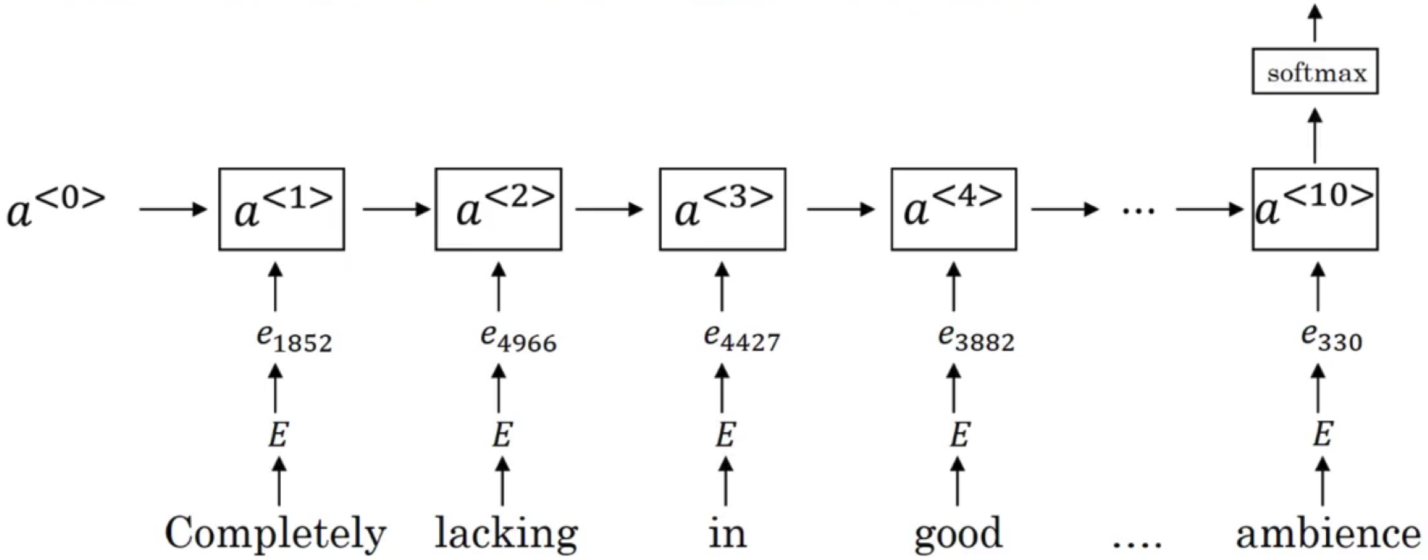


RNN Applications: Sentiment Classification





RNN Applications :Sentiment Classification(Multiple class)



Topics

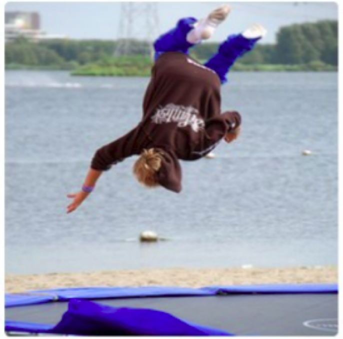
- Why Sequential Models
- Mapping time into State
  - RNN unrolled in time
- RNN Equations
- Type of RNN models
- LSTM Network
  - LSTM Cell
  - LSTM Layer
- RNN Applications
  - Word Embedding
  - Sentiment Classification
  - Image Captioning
  - Neural Machine Translation
  - Attention Networks

RNN Applications: Image Captioning

"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

"two young girls are playing with lego toy."



"boy is doing backflip on wakeboard."

"girl in pink dress is jumping in air."

"black and white dog jumps over bar."

"young girl in pink shirt is swinging on swing."

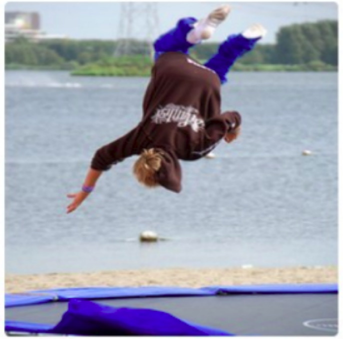
"man in blue wetsuit is surfing on wave."

RNN Applications: Image Captioning

"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

"two young girls are playing with lego toy."



"boy is doing backflip on wakeboard."

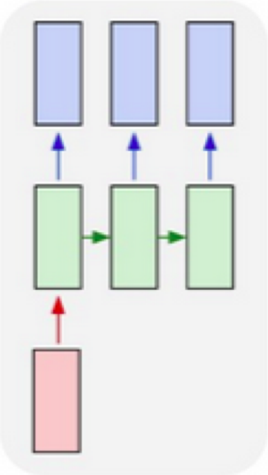
"girl in pink dress is jumping in air."

"black and white dog jumps over bar."

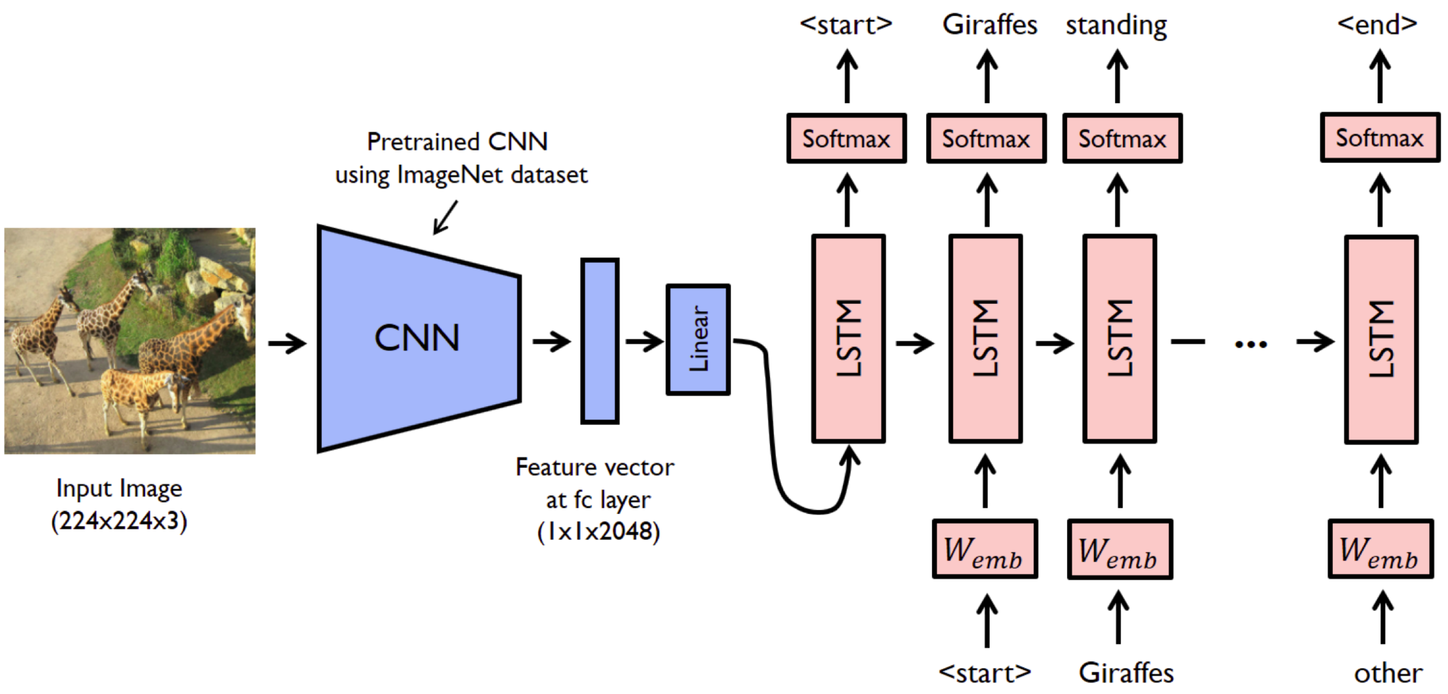
"young girl in pink shirt is swinging on swing."

"man in blue wetsuit is surfing on wave."

one to many



RNN Applications: Image Captioning

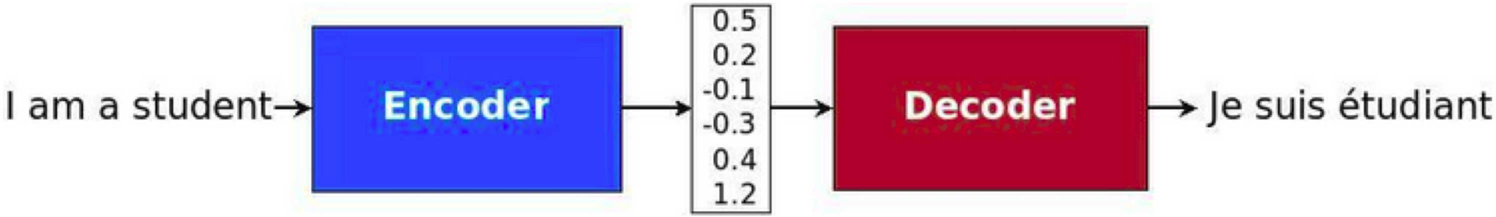


Topics

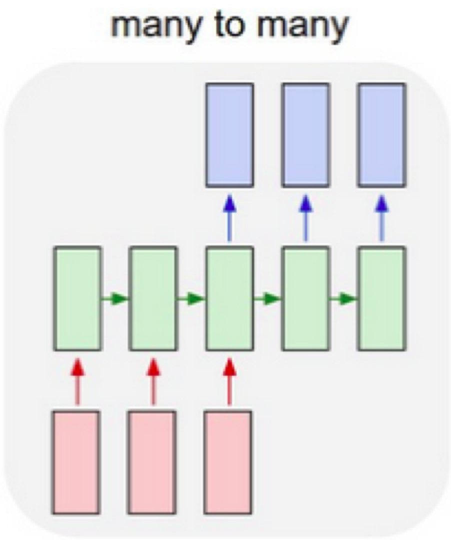
- Why Sequential Models
- Mapping time into State
  - RNN unrolled in time
- RNN Equations
- Type of RNN models
- LSTM Network
  - LSTM Cell
  - LSTM Layer
- RNN Applications
  - Word Embedding
  - Sentiment Classification
  - Image Captioning
  - Neural Machine Translation
  - Attention Networks

RNN Applications: Neural Machine Translation

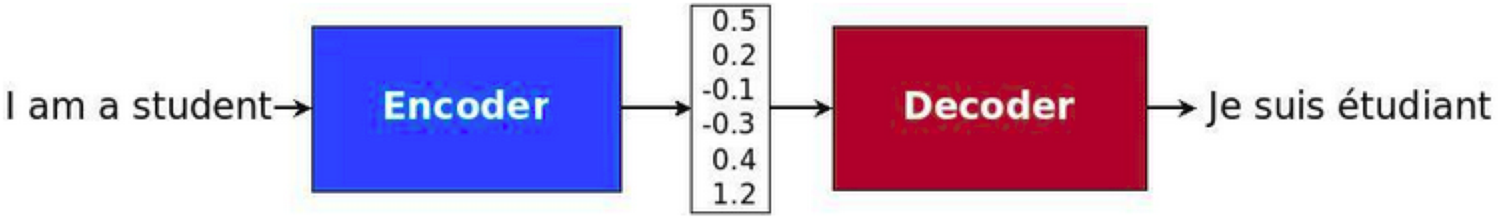
- Language translation: **English** to **French**
- Encode the input sentence to a high dimensional space with the help of an RNN
- Decode to French



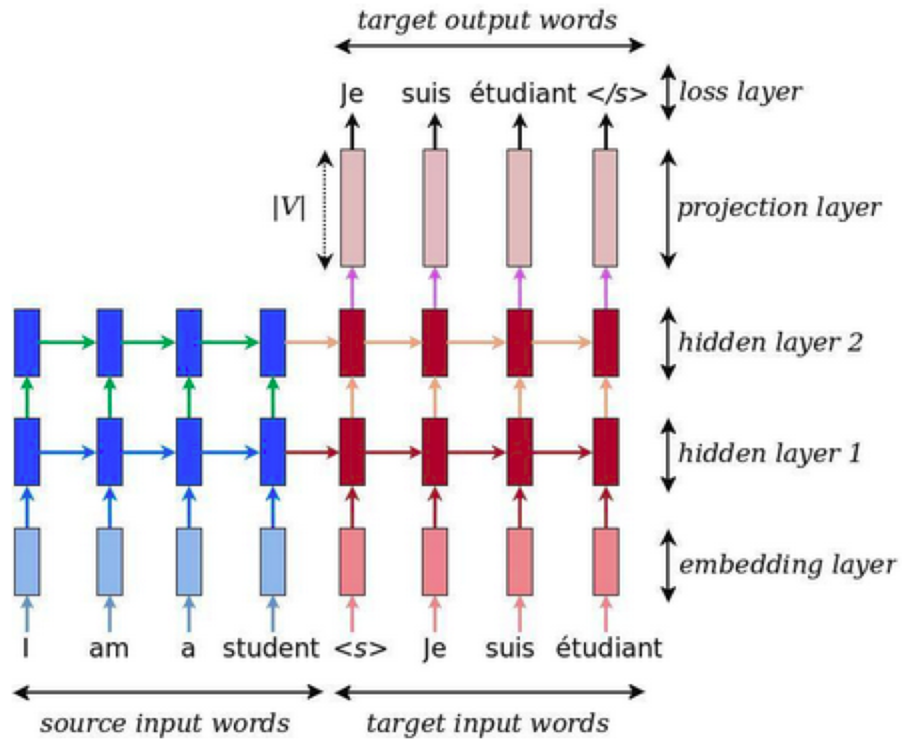
RNN Applications: Neural Machine Translation



- Language translation: **English** to **French**
- Encode the input sentence to a high dimensional space with the help of an RNN
- Decode to French

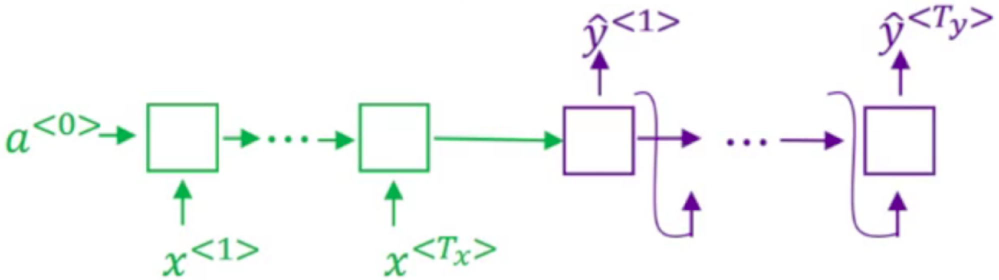


RNN Applications: Neural Machine Translation



RNN Applications: Problem with long sequences

- For long sequences, network may forget what it read earlier.



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.

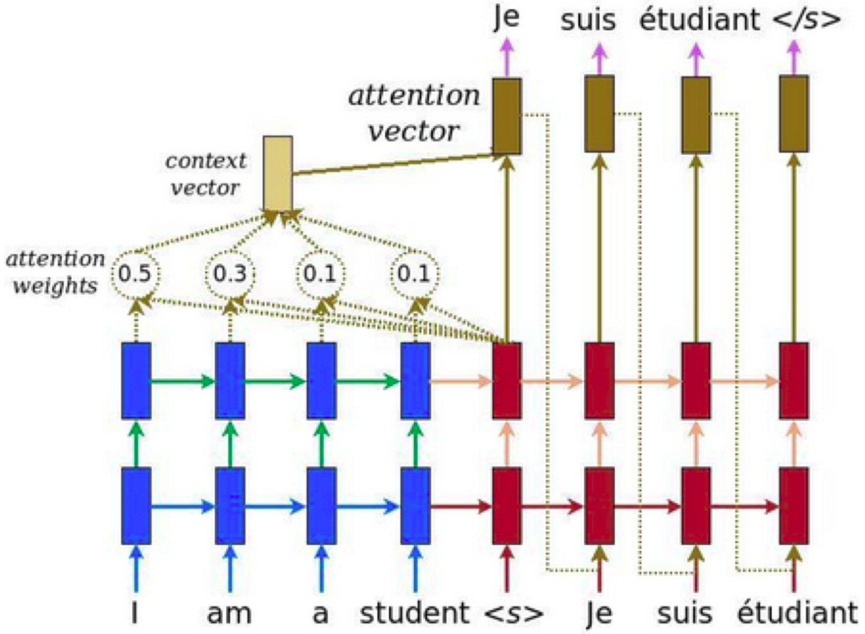
Jane went to Africa last September, and enjoyed the culture and met many wonderful people; she came back raving about how wonderful her trip was, and is tempting me to go too.

Topics

- Why Sequential Models
- Mapping time into State
  - RNN unrolled in time
- RNN Equations
- Type of RNN models
- LSTM Network
  - LSTM Cell
  - LSTM Layer
- RNN Applications
  - Word Embedding
  - Sentiment Classification
  - Image Captioning
  - Neural Machine Translation
  - Attention Networks

RNN Applications: Attention Networks

- At any instant, the network looks at its current hidden state, and all the input words.
- The input words are given a weighting and forms an attention vector.



RNN Applications: Attention calculation

$$\alpha_{ts} = \frac{\exp\left(\mathrm{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_s)\right)}{\sum_{s'=1}^{S} \exp\left(\mathrm{score}(\boldsymbol{h}_t, \bar{\boldsymbol{h}}_{s'})\right)} \qquad \text{[Attention weights]} \qquad (1)$$

$$\boldsymbol{c}_t = \sum_s \alpha_{ts} \bar{\boldsymbol{h}}_s \qquad \text{[Context vector]} \qquad (2)$$

$$\boldsymbol{a}_t = f(\boldsymbol{c}_t, \boldsymbol{h}_t) = \tanh(\boldsymbol{W}_{\boldsymbol{c}}[\boldsymbol{c}_t; \boldsymbol{h}_t]) \qquad \text{[Attention vector]} \qquad (3)$$