Lecture: Recurrent Neural Network (RNN)
An Introduction

Dr. Goutam Chakraborty

SAS® Professor of Marketing Analytics

Director of MS in Business Analytics and Data Science* (http://analytics.okstate.edu/mban/)
Director of Graduate Certificate in Business Data Mining (http://analytics.okstate.edu/certificate/grad-data-mining/)
Director of Graduate Certificate in Marketing Analytics (http://analytics.okstate.edu/certificate/grad-marketing-analytics/)

- *Name change pending internal approval.
-
- Some of the slides were developed by Mr. Sanjoy Dey, used with permission.

1

Outline

- Typical applications of RNN
- Some idea about mechanics of RNN

2

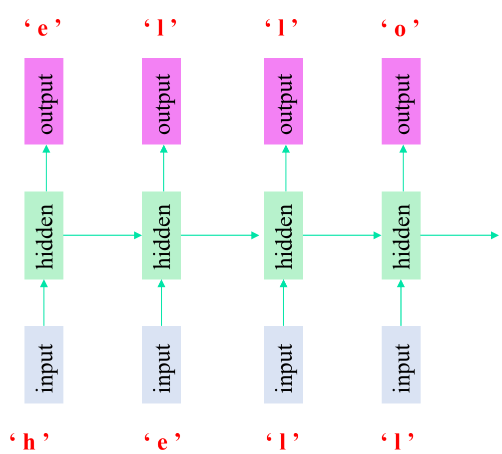
CNN vs. RNN

- A CNN will learn to recognize patterns *across space*
  - CNNs are stateless
- RNN will similarly learn to recognize patterns *across time/sequence*
  - RNNs are stateful

3

Use Cases of RNN: One to One

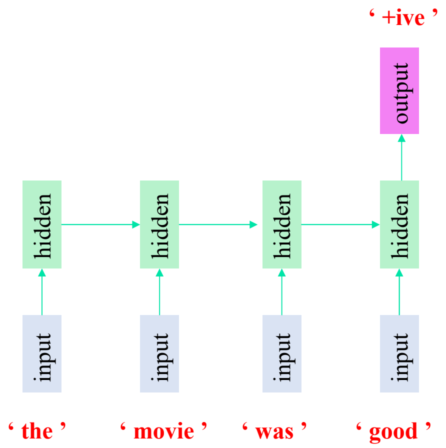
Predict next word (or character) of a sequence



4

Use Cases of RNN: Many to One

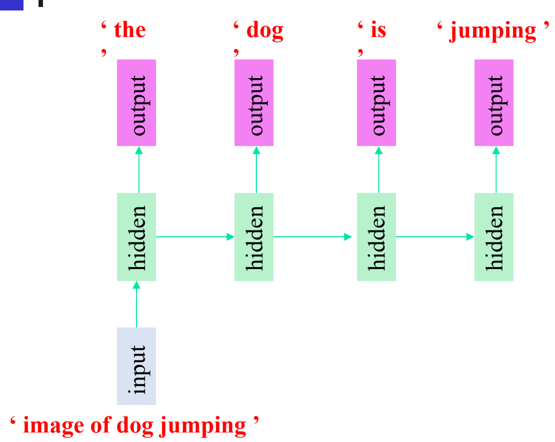
Predict sentiment from a movie review



5

Use cases of RNN : One to Many

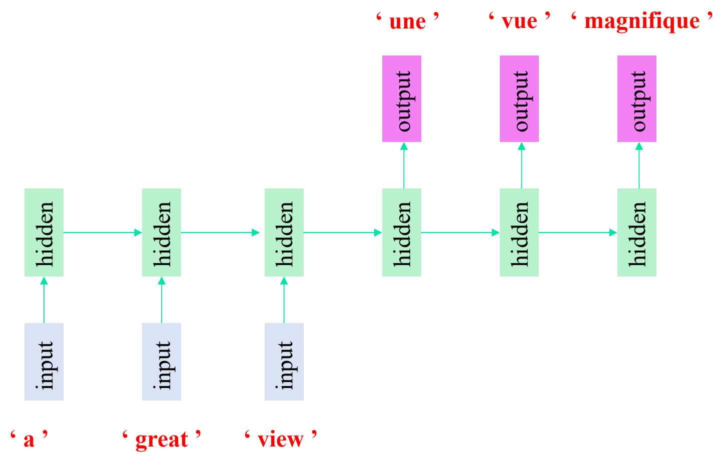
Generate caption for a picture



6

Use cases of RNN: Many to Many

Machine translation



7

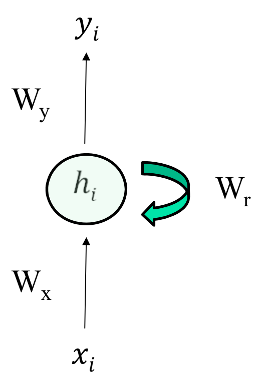
Capturing Dependence

Example of deep dependence

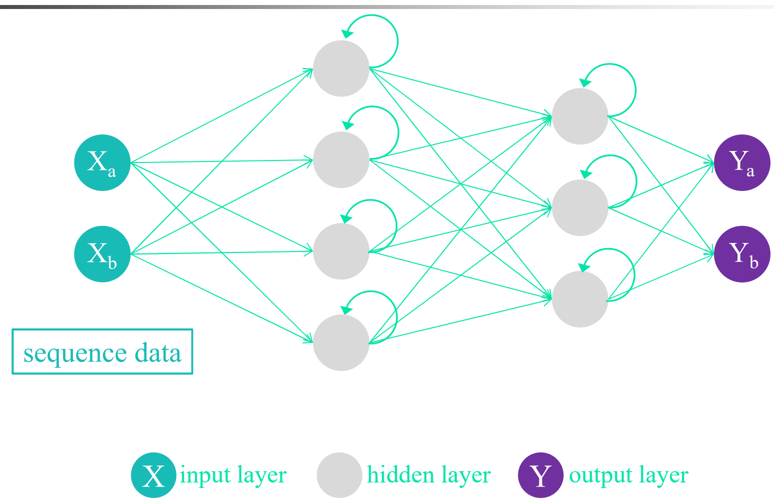
“Michelle lives in the capital city of France. Paris is the capital of France. Michelle lives in …….”

8

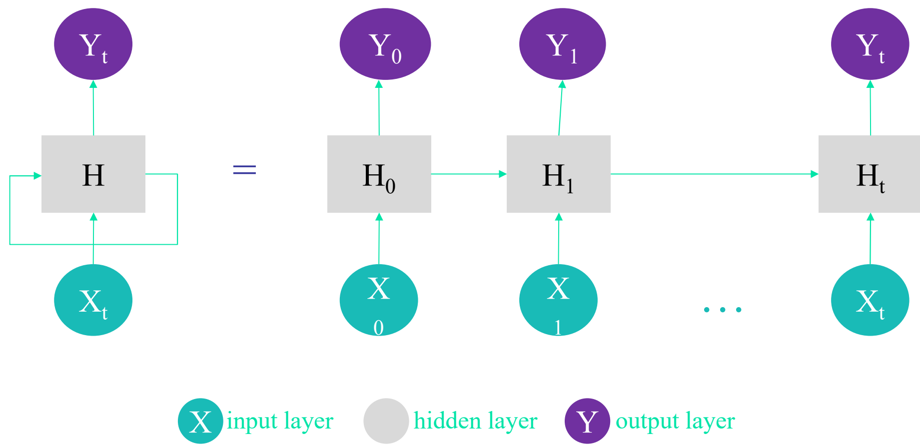
A Recurrent Neuron



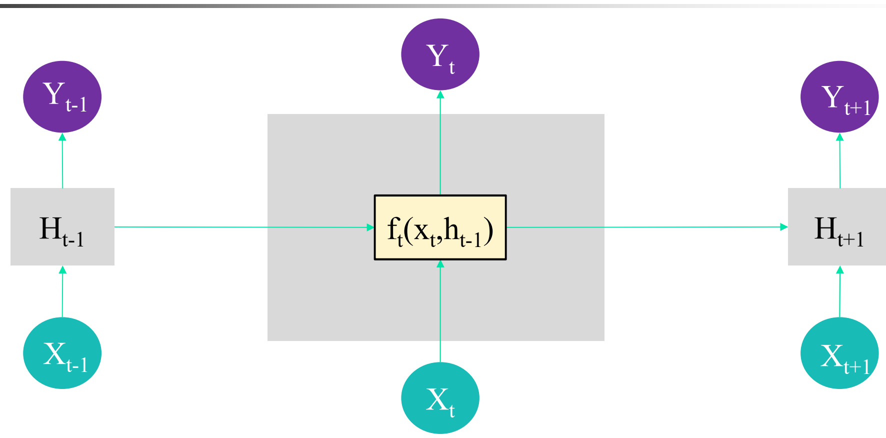
Recurrent Neural Network (RNN)



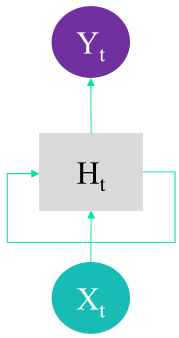
RNN Unfolded



RNN Weights



RNN Weights



weights for output

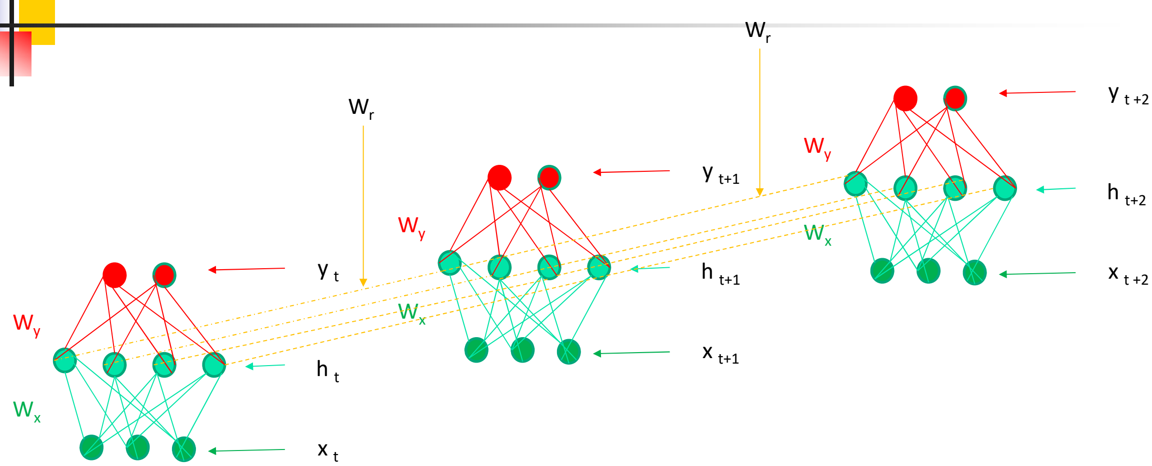
$$Y_t = sigmoid\left[\boxed{W_Y}\cdot H_t + \boxed{b_Y}\right]$$

$$H_t = \tanh\left[\boxed{W_H}\cdot H_{t-1} + W_X \cdot X_t + \boxed{b_H}\right]$$

weights for hidden unit

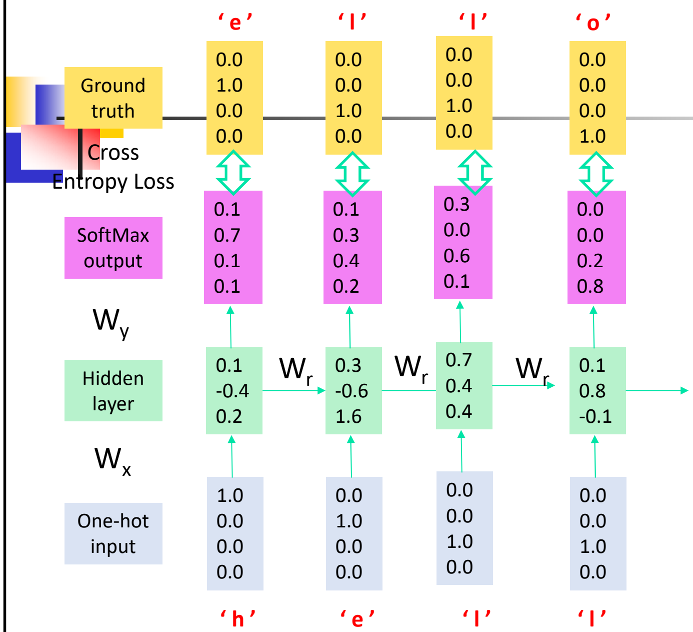
W_H is the recurrent weight that connects previous neuron to current neuron

An Expanded view of an RNN



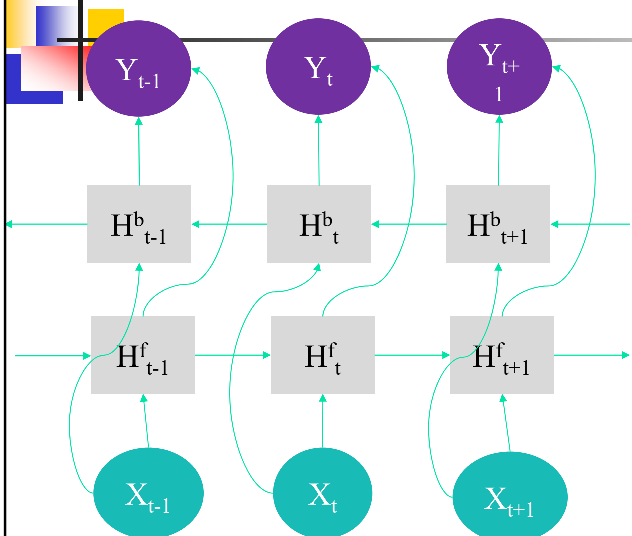
14

A Simple Training Scenario of RNN



15

Bidirectional RNN: Two Hidden layers



$$Y_t = sigmoid[W_Y^f \cdot H_t^f + W_Y^b \cdot H_t^b + b_Y]$$

$$H_t^f = \tanh[W_H^f \cdot H_{t-1}^f + W_X^f \cdot X_t + b_H^f]$$

$$H_t^b = \tanh[W_H^b \cdot H_{t+1}^b + W_X^b \cdot X_t + b_H^b]$$

Lecture: RNN
Vanishing and Exploding Gradients

Dr. Goutam Chakraborty

SAS® Professor of Marketing Analytics

Director of MS in Business Analytics and Data Science* (http://analytics.okstate.edu/mban/)
Director of Graduate Certificate in Business Data Mining (http://analytics.okstate.edu/certificate/grad-data-mining/)
Director of Graduate Certificate in Marketing Analytics (http://analytics.okstate.edu/certificate/grad-marketing-analytics/)

- *Name change pending internal approval.
-
- Some of the slides were developed by Mr. Sanjoy Dey, used with permission.

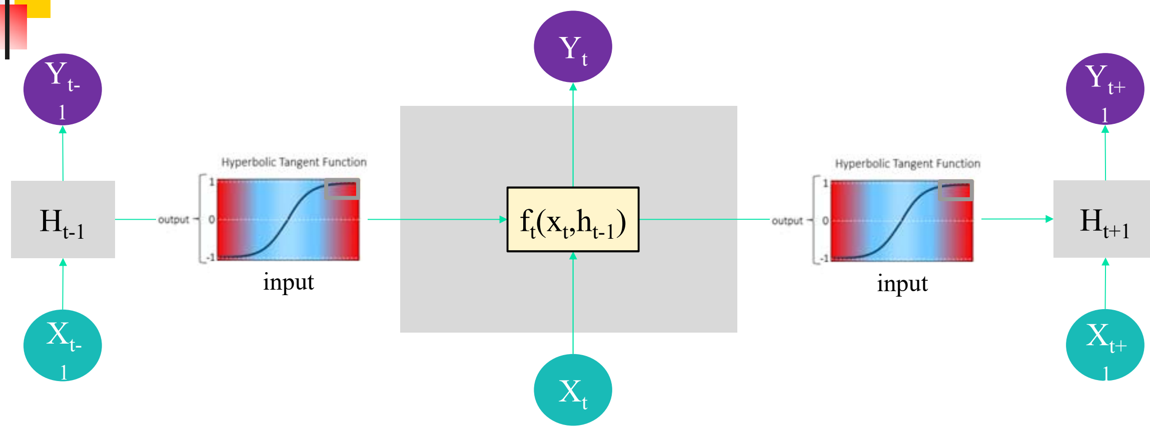
17

Outline

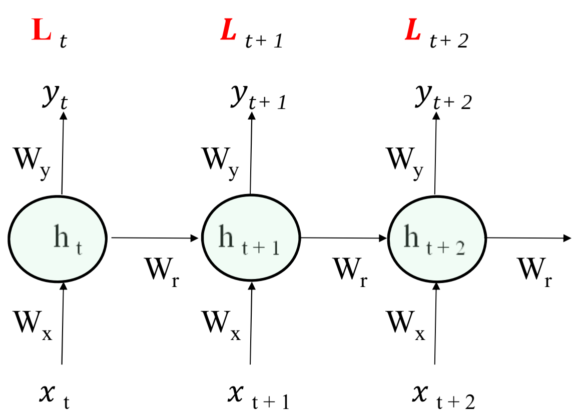
- Vanishing gradient problem
- Exploding gradient problem

18

Vanishing Gradient



Chain Rule's Contribution to Vanishing Gradient



$$h_t = g_h(W_x\, x_t + W_r\, h_{t-1} + b_h)$$

$$y_t = g_y(W_y\, h_t + b_y)$$

$$\frac{\partial L}{\partial W_r} = \sum_t \frac{\partial L_t}{\partial W_r}$$

$$\frac{\partial L_2}{\partial W_r} = \frac{\partial L_2}{\partial y_2} \cdot \frac{\partial y_2}{\partial h_2} \cdot \frac{\partial h_2}{\partial g} \cdot \frac{\partial g}{\partial a} \cdot \frac{\partial a}{\partial W_r}$$

$$\frac{\partial L_T}{\partial W_r} \propto |W_r|^{T} \left|\frac{\partial g}{\partial a}\right|^{T}$$

20

Lecture: RNN
GRU and LSTM

Dr. Goutam Chakraborty

SAS® Professor of Marketing Analytics

Director of MS in Business Analytics and Data Science* (http://analytics.okstate.edu/mban/)
Director of Graduate Certificate in Business Data Mining (http://analytics.okstate.edu/certificate/grad-data-mining/)
Director of Graduate Certificate in Marketing Analytics (http://analytics.okstate.edu/certificate/grad-marketing-analytics/)

- *Name change pending internal approval.
-
- Some of the slides were developed by Mr. Sanjoy Dey, used with permission.

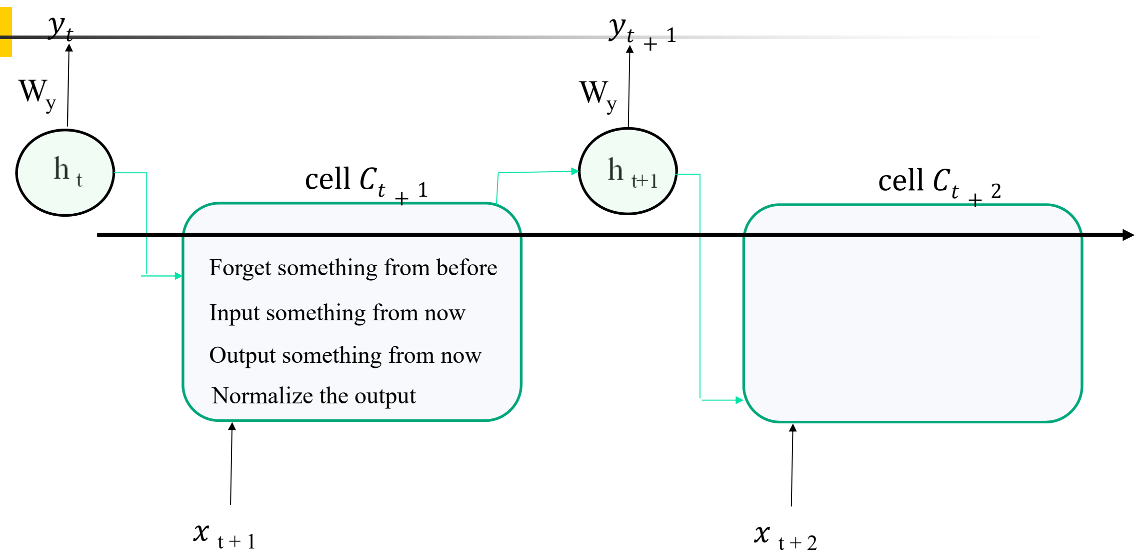
21

Outline

- Gated Recurrent Unit (GRU) Network
- Long Short-term Memory (LSTM)

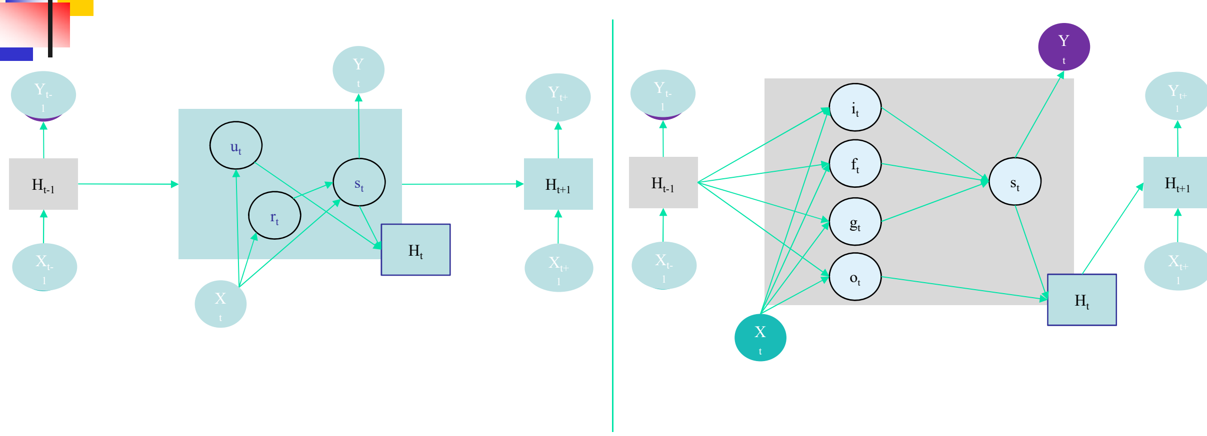
22

Cell State

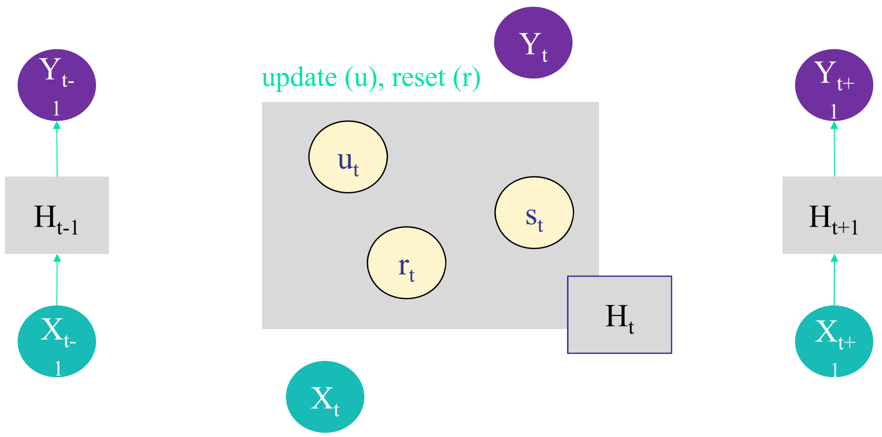


23

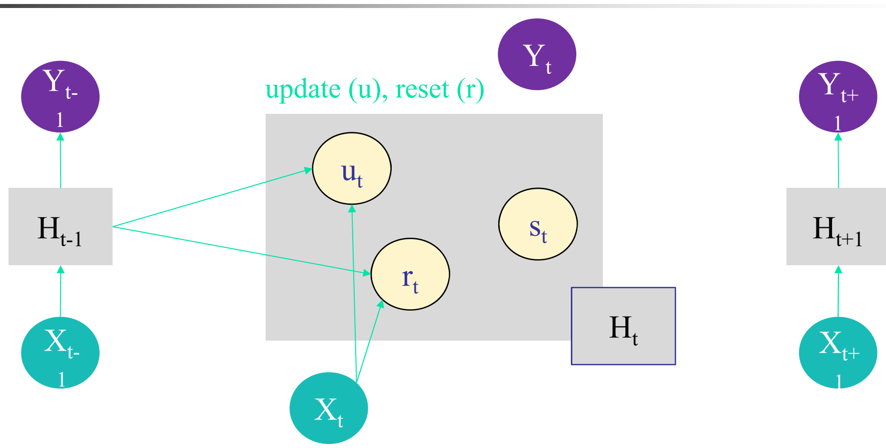
Gated Recurrent Units (GRU) | Long Short-Term Memory (LSTM)



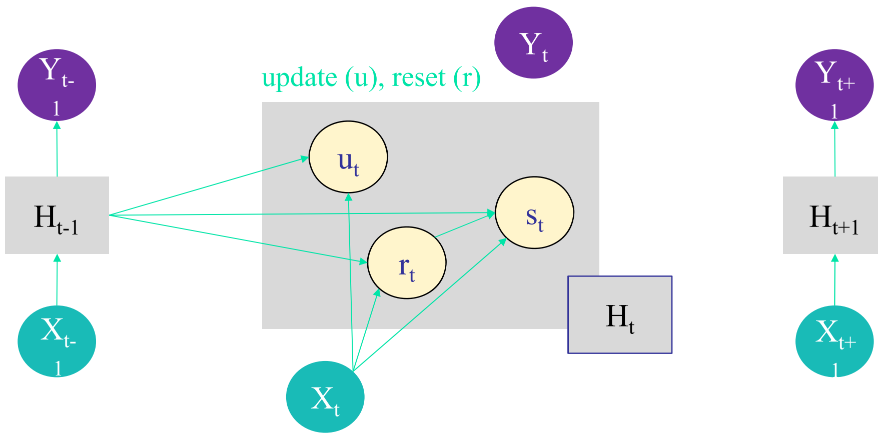
Gated Recurrent Neural Network



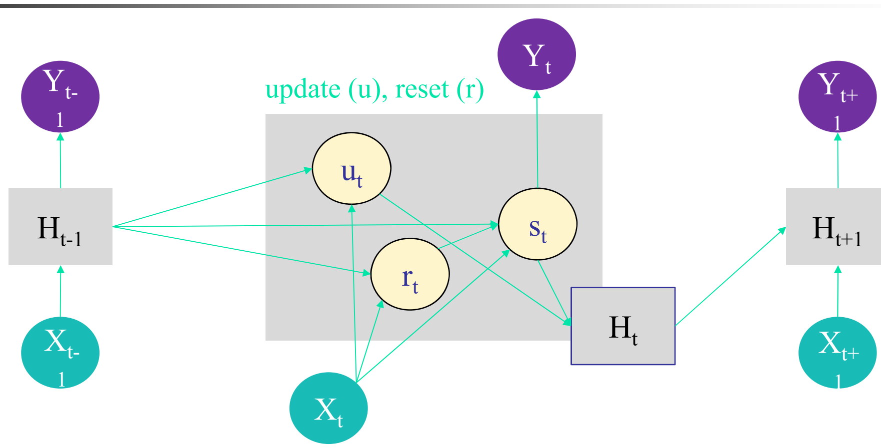
Gated Recurrent Neural Network



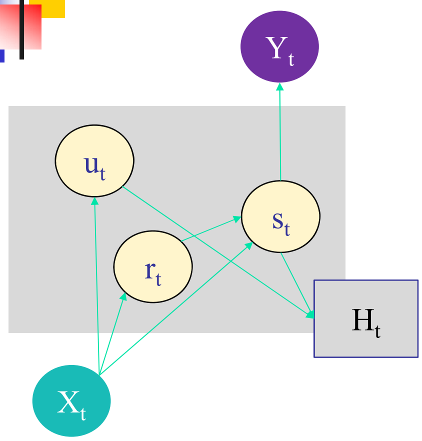
Gated Recurrent Neural Network



Gated Recurrent Neural Network



Gated Recurrent Neural Network



$$u_t = sigmoid\left[W_u \cdot X_t + U_u \cdot H_{t-1} + b_u\right]$$

$$r_t = sigmoid\left[W_r \cdot X_t + U_r \cdot H_{t-1} + b_r\right]$$

$$s_t = \tanh\left[U_s \cdot (r_t \circ H_{t-1}) + W_s \cdot X_t + b_H\right]$$

$$H_t = u_t \circ H_{t-1} + (1 - u_t) \circ s_t$$

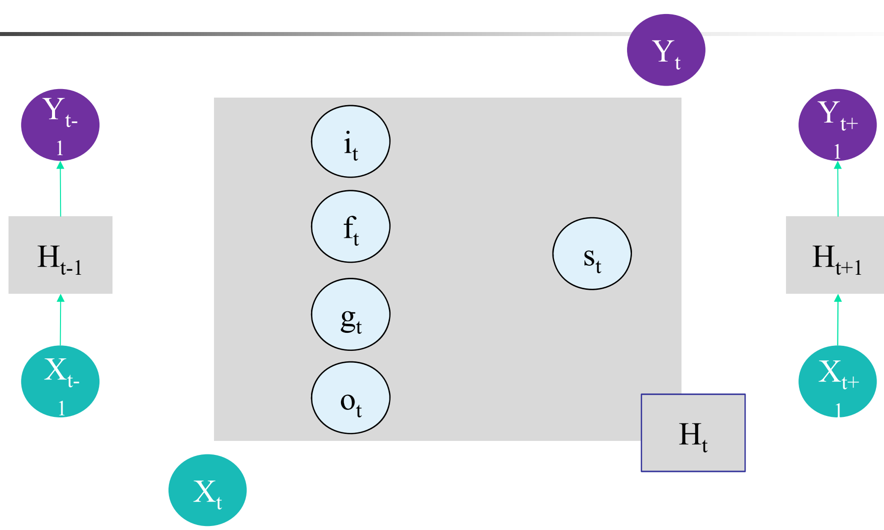
$$Y_t = sigmoid\left[W_Y \cdot H_t + b_Y\right]$$

W, U, b: weight parameters
Dot: dot product of vectors
Hollow dot: elementwise multiplication (Hadamard product)

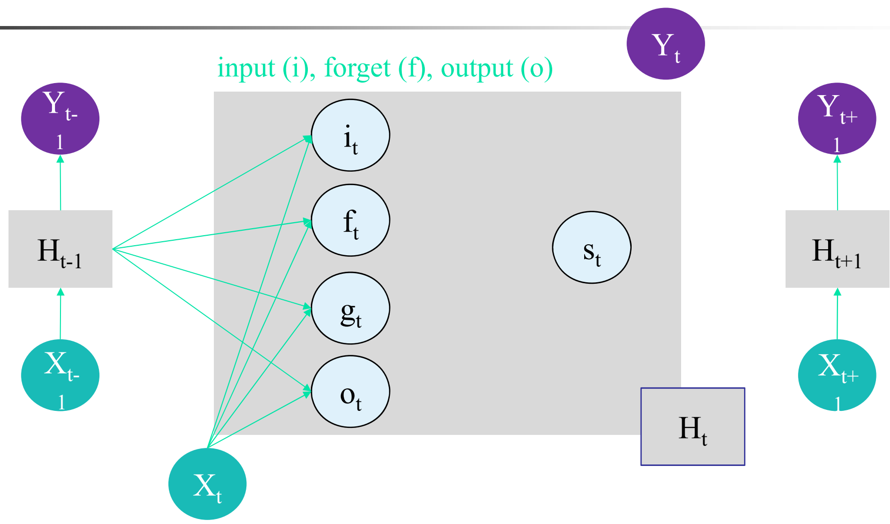
For example, the Hadamard product for a 3 × 3 matrix **A** with a 3 × 3 matrix **B** is

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \circ \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} = \begin{bmatrix} a_{11}\,b_{11} & a_{12}\,b_{12} & a_{13}\,b_{13} \\ a_{21}\,b_{21} & a_{22}\,b_{22} & a_{23}\,b_{23} \\ a_{31}\,b_{31} & a_{32}\,b_{32} & a_{33}\,b_{33} \end{bmatrix}$$

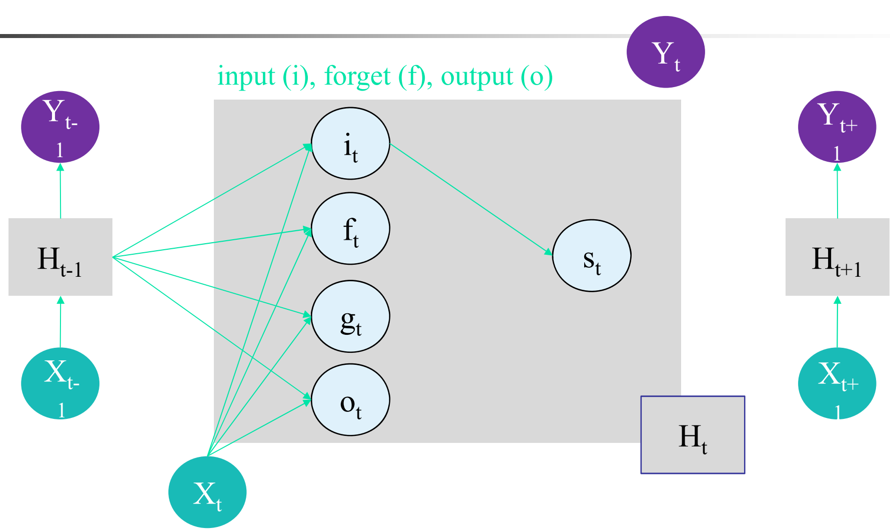
Long Short-Term Memory Neural Network



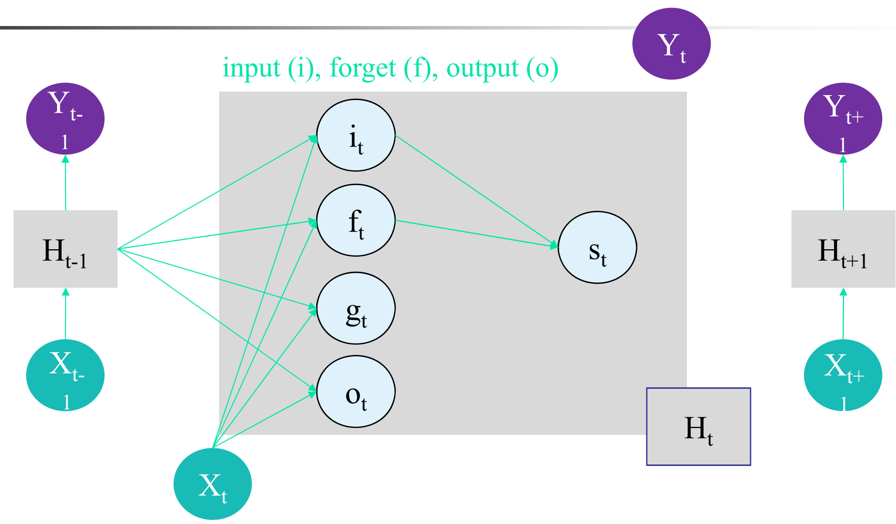
Long Short-Term Memory Neural Network



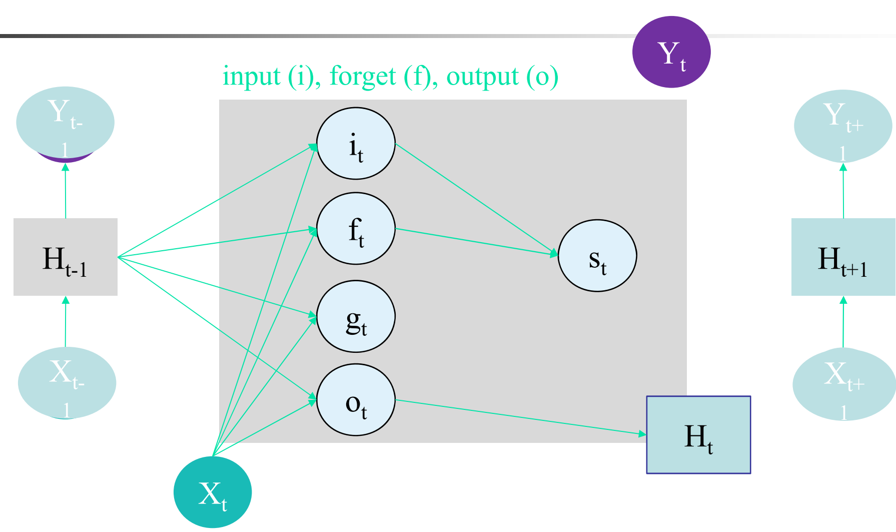
Long Short-Term Memory Neural Network



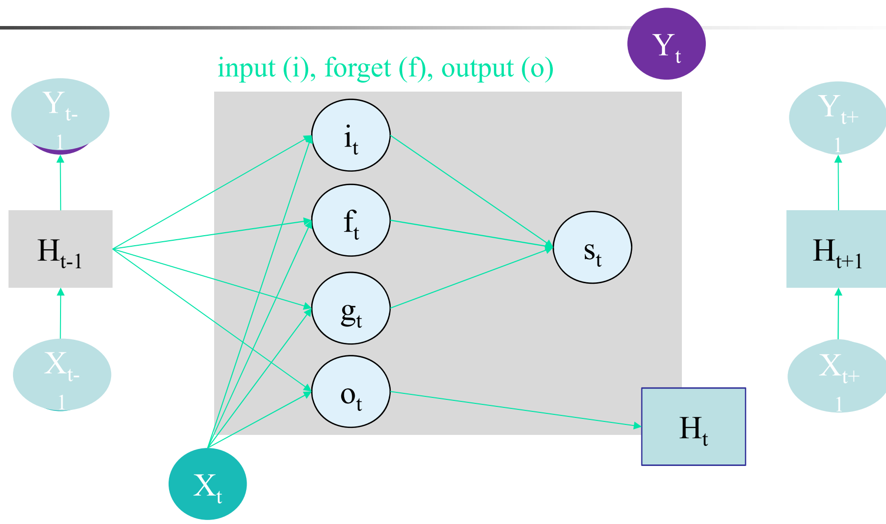
Long Short-Term Memory Neural Network



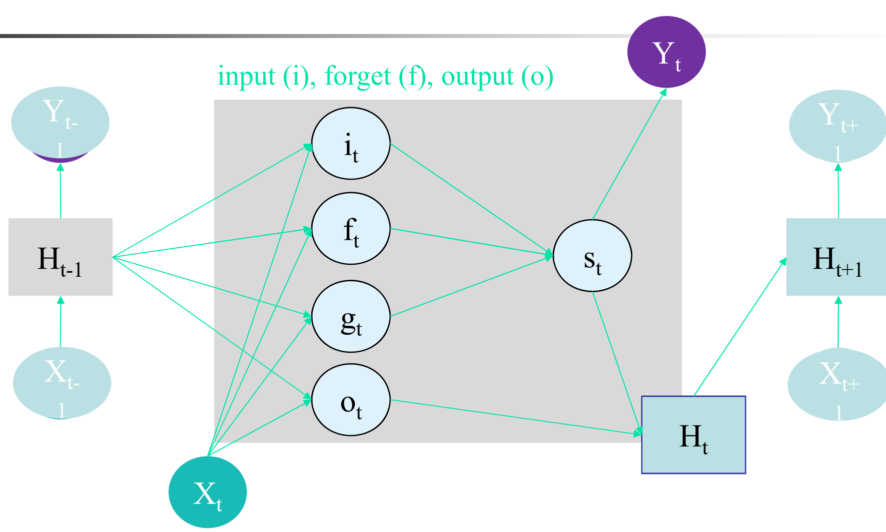
Long Short-Term Memory Neural Network



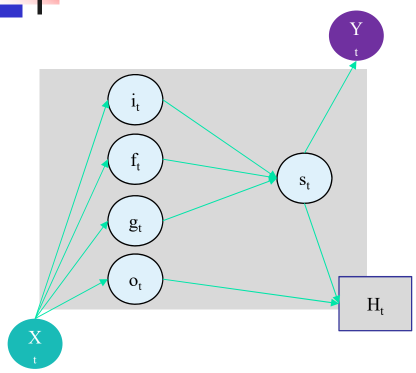
Long Short-Term Memory Neural Network



Long Short-Term Memory Neural Network



Long Short-Term Memory Neural Network



$$i_t = sigmoid\left[W_i \cdot X_t + U_i \cdot H_{t-1} + b_i\right]$$

$$f_t = sigmoid\left[W_f \cdot X_t + U_f \cdot H_{t-1} + b_f\right]$$

$$o_t = sigmoid\left[W_o \cdot X_t + U_o \cdot H_{t-1} + b_o\right]$$

$$g_t = \tanh\left[W_g \cdot H_{t-1} + U_g \cdot X_t + b_g\right]$$

$$s_t = i_t \circ g_t + f_t \circ s_{t-1}$$

$$H_t = \tanh\left[s_t\right] \circ o_t$$

$$Y_t = sigmoid\left[W_Y \cdot H_t + b_Y\right]$$

W, U, b: weight parameters
Dot: dot product of vectors.
Hollow dot: elementwise multiplication (**Hadamard product**)

How Does LSTM Solve the Vanishing Gradient Problem?

- There is no recurrent weight matrix W_r any more
- Error is being propagated between two steps not by a gradient flow of W_r , but by an addition operation (input gate) or a subtraction operation (forget gate)
- If forget gate = 1, and input gate =0 then the cell state is intact and error is propagated intact
- Modulation of the gates is a learned parameter and that is how distant states are remembered or forgotten
- The output gate just adds another layer of flexibility, making the model training more fine tuned

38

Lecture: RNN
Word Embedding

Dr. Goutam Chakraborty

SAS® Professor of Marketing Analytics

Director of MS in Business Analytics and Data Science* (http://analytics.okstate.edu/mban/)
Director of Graduate Certificate in Business Data Mining (http://analytics.okstate.edu/certificate/grad-data-mining/)
Director of Graduate Certificate in Marketing Analytics (http://analytics.okstate.edu/certificate/grad-marketing-analytics/)

- *Name change pending internal approval.
-
- Some of the slides were developed by Mr. Sanjoy Dey, used with permission.

39

Outline

- Problems with one-hot vector representation in terms-by-document matrix
- Introduce concept of word embedding
- How is it done in practice?
  - Word2Vec
  - Glove

40

Problems of One Hot Vector Representation

- It is typically an **enormous and sparse** vector
  - e.g., for a 5,000 words vocab, for each word the vector will be 1X5000 with 1 in one place and 0 in the rest 4,999 places
- The vector does not capture the meaning of the word (semantic info.)
- The vector does not have any information on where the word is placed in the sentence (syntactic info.)
- For example 'house' and 'apartment' should be **nearer** in vector space than 'house' and 'moon', but in one-hot representation:
  - The dot product of 'house" and 'apartment' = 0
  - The dot product of 'house" and 'moon' = 0

41

Acquiring Semantic Information

- Consider the 2 sentences
  - There are four people living in this house
  - People living in this apartment work in Google
- If we associate the word 'living' with the words 'house' and 'apartment' and *somehow embed this association*, then we can build a semantic similarity between 'house' and 'apartment'
- This technique is called word embedding

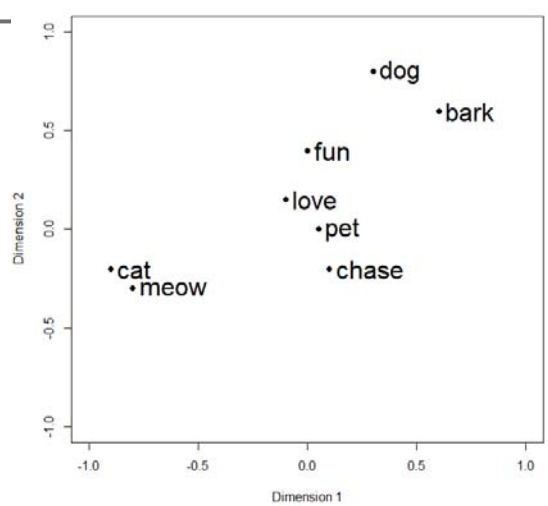
42

Characteristics of Word Embedding

- They are **dense** vectors of pre-determined size
- They capture semantic content by association with other words in the corpora
- They capture some syntactic content too
- They are trainable
- Training can *be unsupervised*, e.g., running the algorithm over a large corpora such as Wikipedia

43

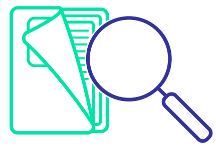
Word Embedding from a Pet Blog Corpus



Word Embedding

1. Learn word representations for entire corpus
   - Matrix factorization methods
   - Similar to SVD

2. Learn word representations locally
   - Shallow window-based methods
   - Words near word of interest



Word Embedding Algorithms

- Word2vec which has 2 implementations
  - Skip gram model
  - Combined Bag of Words model
- GloVe

46

How is it Done ? Word2Vec

u_o

There are four people living in this house

v_c

Steps to compute the vectors:

1. Initialize all the u and v vectors to some small random number of a fixed dimension
2. Choose the hyperparameters e.g., window size, update frequency, etc.
3. Compute the objective function which is below and maximize it

$$J_t(\boldsymbol{\theta}) = \log \sigma (u_o{}^T v_c) + \sum_{J \sim P(w)} [\log \sigma (- u_j{}^T v_c)]$$

4. Train a neural network to obtain the parameters and get the best values of vectors u and v
5. Since u and v are word representations of the same vocabulary, we synthesize the two by adding

47

How is it Done ? Word2Vec (Contd.)

There are four people living in this house

v_c

48

How is it Done ? Word2Vec (Contd.)

There are four people living in this house

v_c

49

How is it Done ? Word2Vec (Contd.)

There are four people living in this house

v_c

50

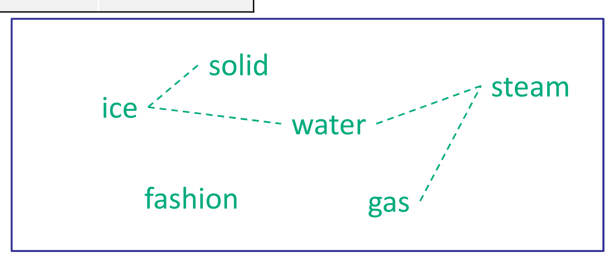
How is it Done ? Word2Vec (Contd.)

There are four people living in this house

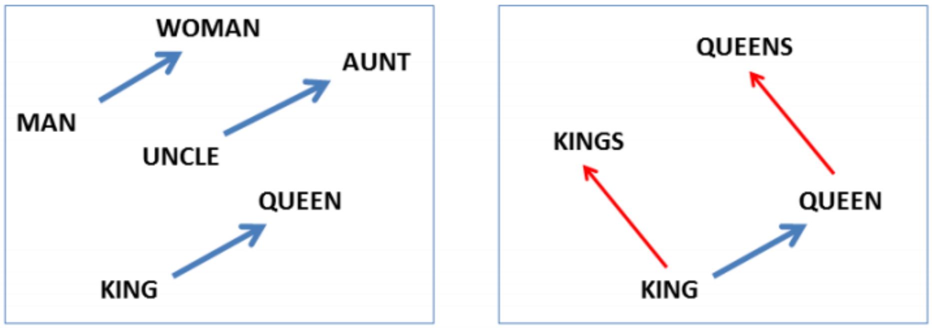
V_c

GloVe Algorithm

Probability and Ratio	k=solid	k=gas	k=water	k=fashion
P(k|ice)	$1.9x10^{-4}$	$6.6x10^{-5}$	$3.0x10^{-3}$	$1.7x10^{-5}$
P(k|steam)	$2.2x10^{-5}$	$7.8x10^{-4}$	$2.2x10^{-3}$	$1.8x10^{-5}$
P(k|ice) / P(k|steam)	8.9	$8.5x10^{-2}$	1.36	0.96



Derivations from Word Embedding



(Mikolov et al., NAACL HLT, 2013)

53