UNIVERSITY OF WATERLOO

Recurrent Neural Networks

DSG Meeting

June 7, 2017

Salman Mohammed

David R. Cheriton School of Computer Science

University of Waterloo

Acknowledgement

Andrej Karpathy
http://karpathy.github.io/2015/05/21/rnn-effectiveness/

Christopher Olah
http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Richard Socher, Christopher Manning
http://web.stanford.edu/class/cs224n/syllabus.html

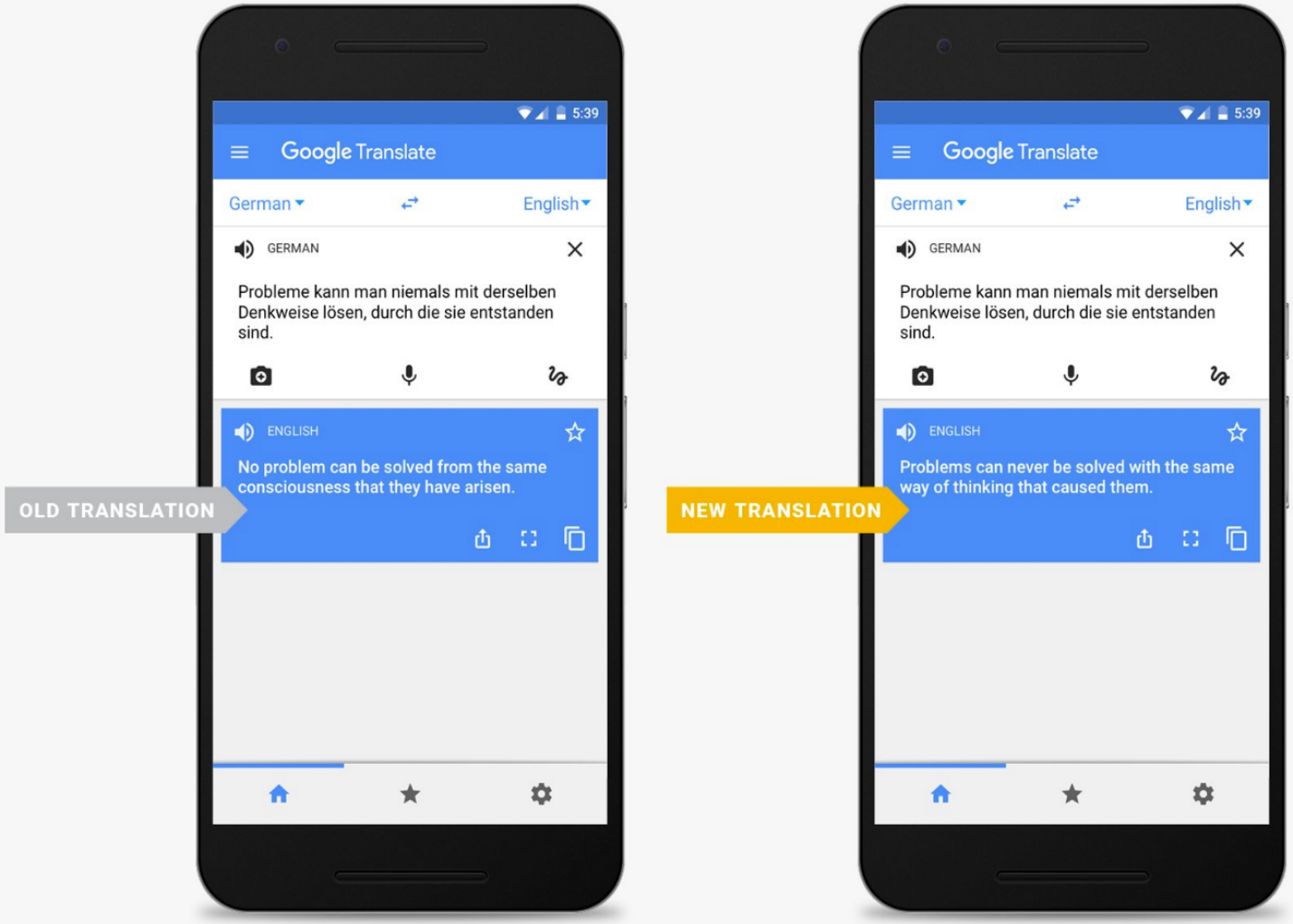
Jimmy Lin
slide template taken from https://lintool.github.io/bigdata-2017w

Denny Britz
http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/

Harini Suresh
http://introtodeeplearning.com/Sequence%20Modeling.pdf

Motivation

Source: https://www.apple.com/newsroom/2017/01/hey-siri-whos-going-to-win-the-super-bowl/



Source: https://blog.google/products/translate/found-translation-more-accurate-fluent-sentences-google-translate/

Assuming you know…

Word Vectors

dense vector representation for words

Fully Connected Neural Networks

every node in a layer connected to all nodes in the previous layer

Idea of Backpropagation

training a neural network

Limitations of NNs

Constrained API

fixed size input(image) and output(classes)

Modelling Sequences

traditional networks have no sense of ‘state’

use reasoning about previous events to make decision

Use of RNNs

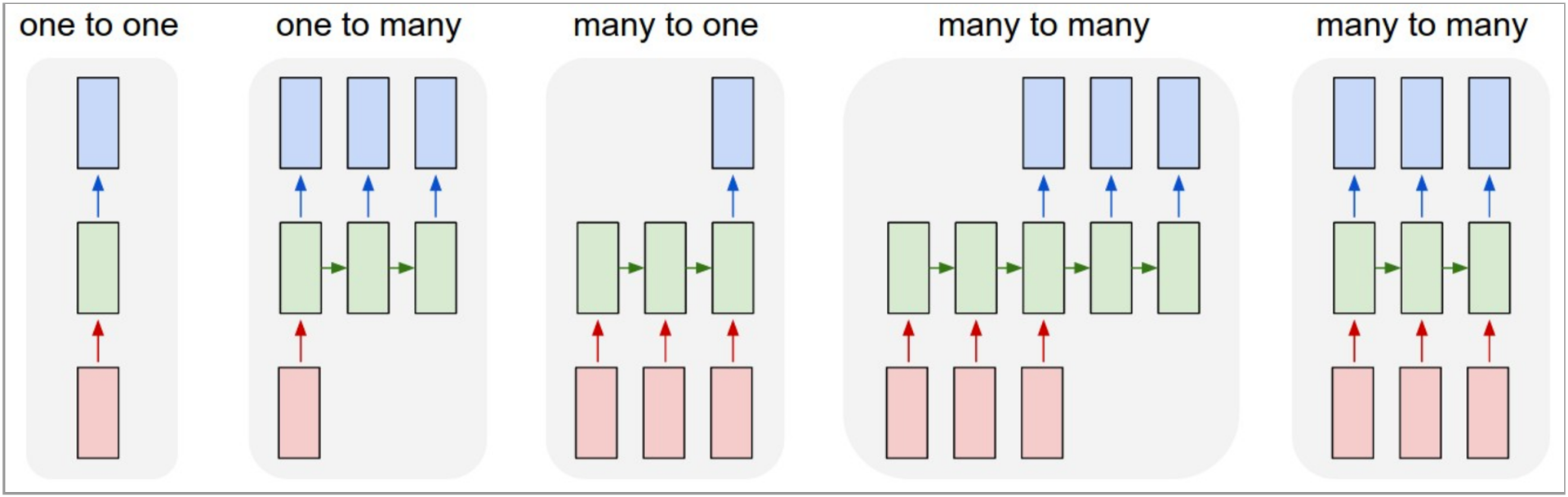


Image Captioning

Machine Translation

Image Classification
(ConvNets)

Sentiment Analysis
Text Classification
Relation Prediction

Entity Detection
Video Frame Classification

Recurrent NNs

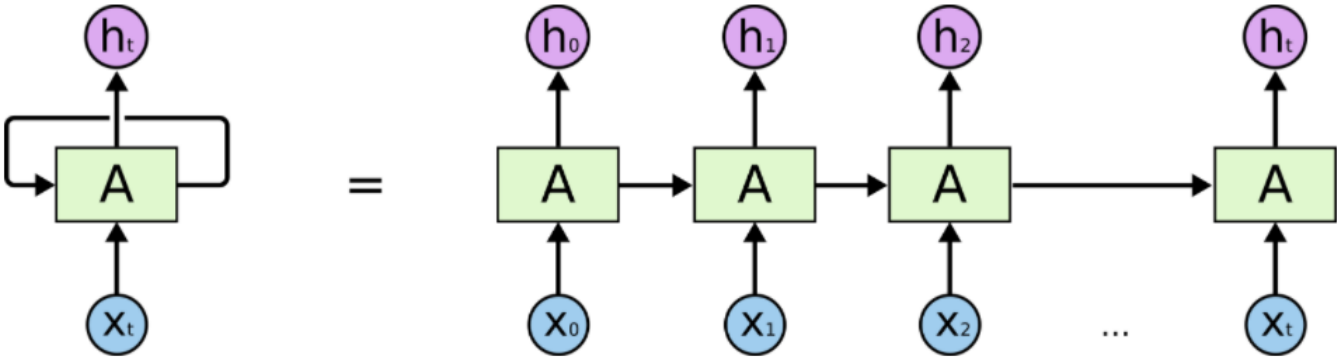
Input: x_t

word embedding

Memory/State: h_t

embedding based on current input and previous state

final state: think "sentence embedding"



An unrolled recurrent neural network.

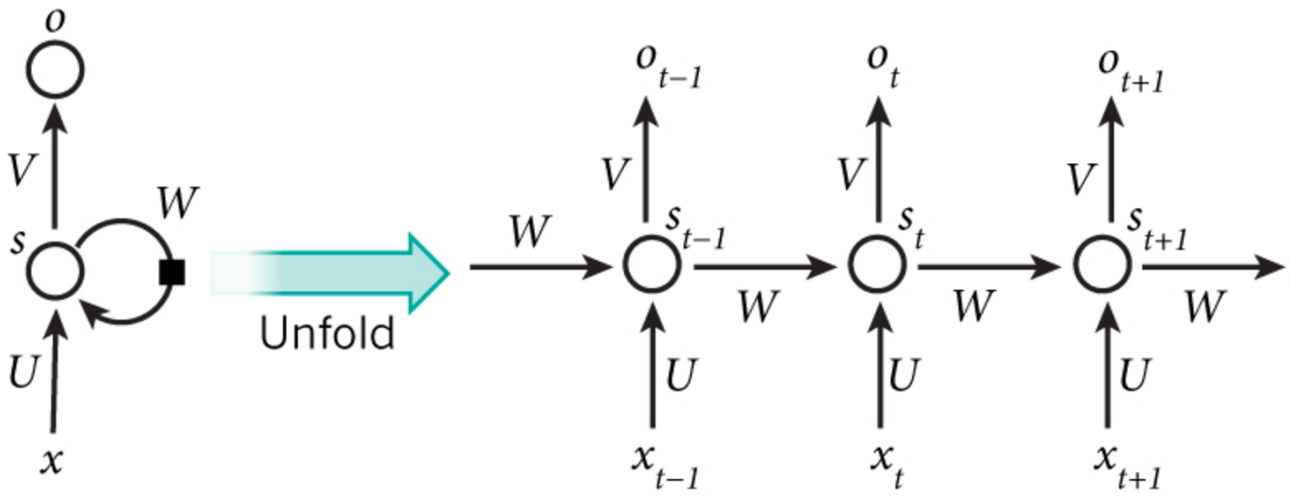
Recurrent NNs (more detail)

State: $s_t = f(U.x_t + W.s_{t-1})$

f is a non-linear function (ReLU, tanh)

Output: $o_t = softmax(V.s_t)$

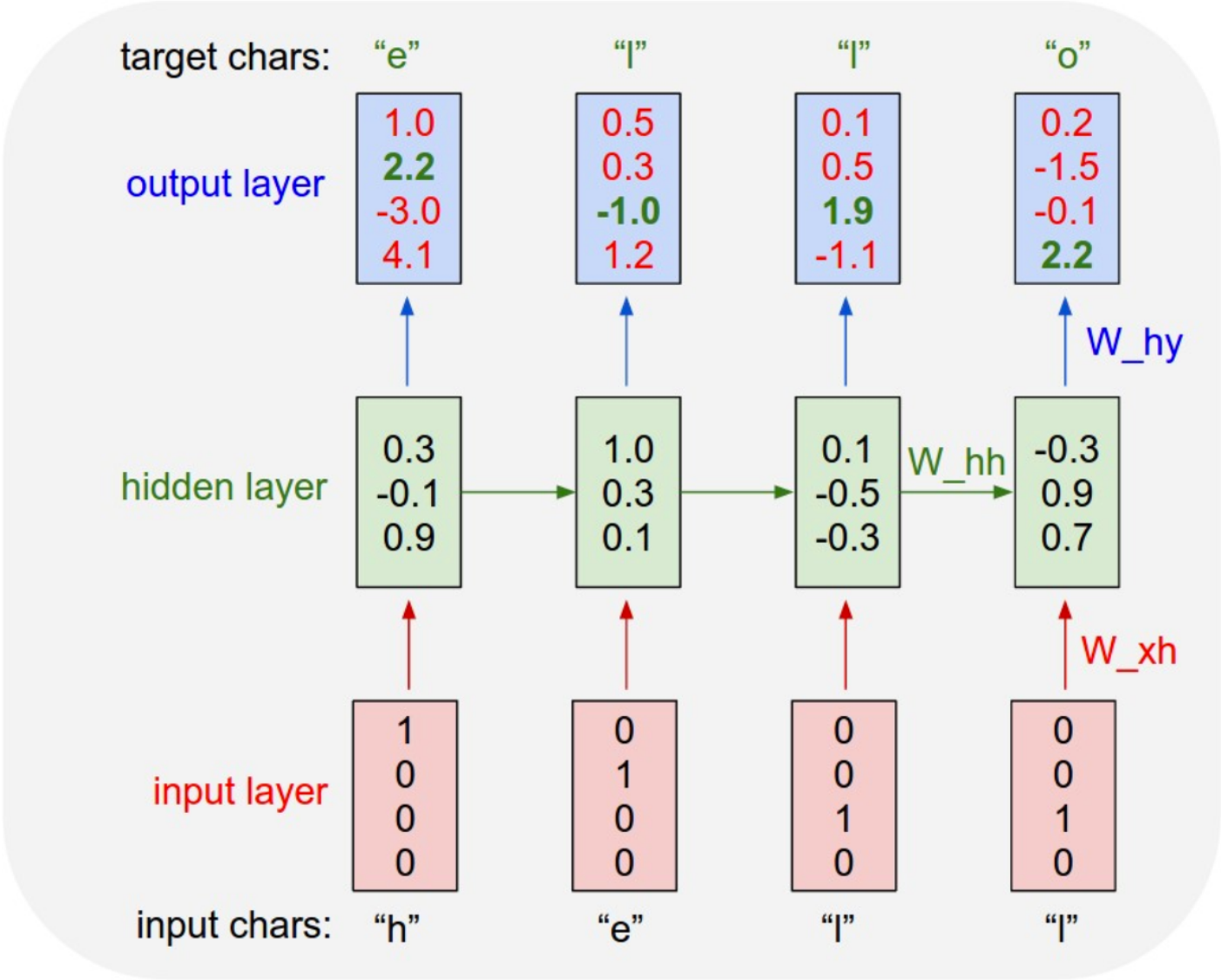
output a vector of probabilities



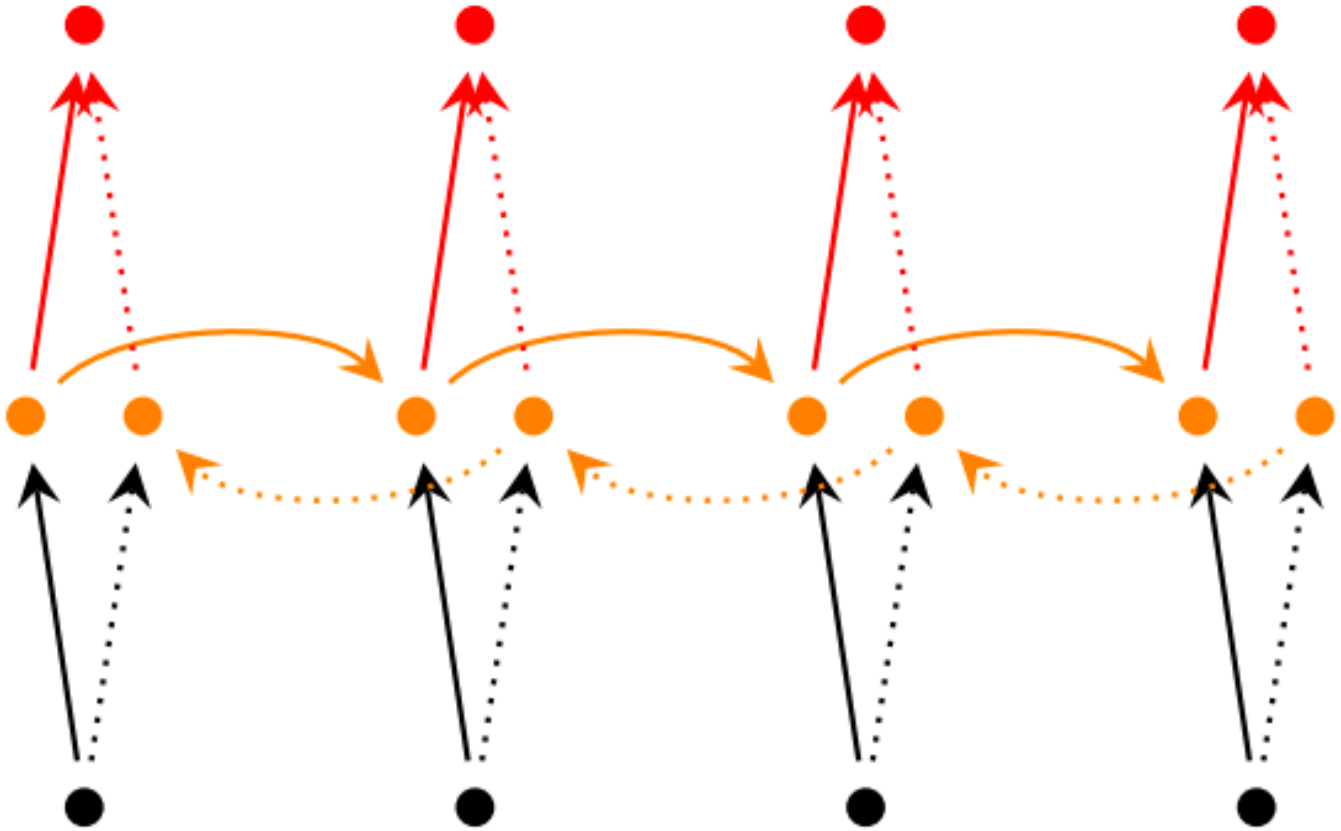
A recurrent neural network and the unfolding in time of the computation involved in its forward computation.

Source: Nature

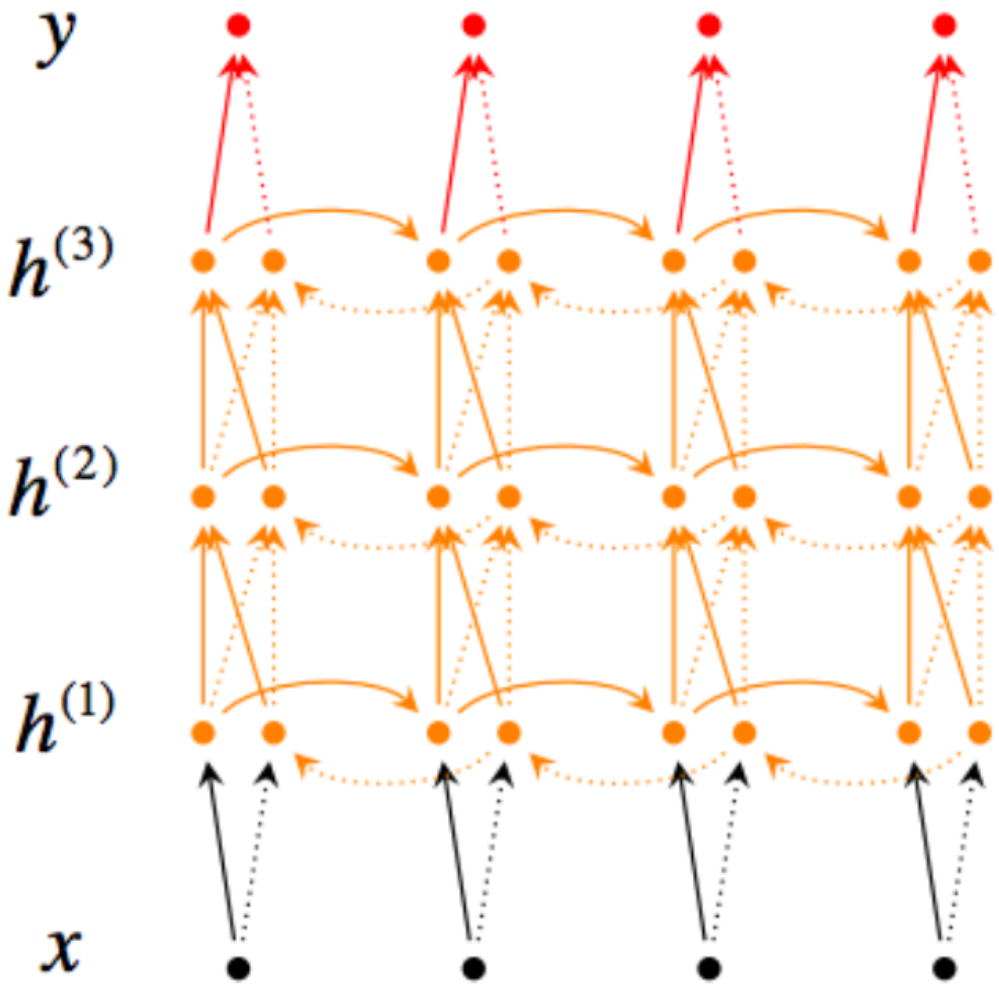
Example: Character Language Model



Bi-directional RNNs



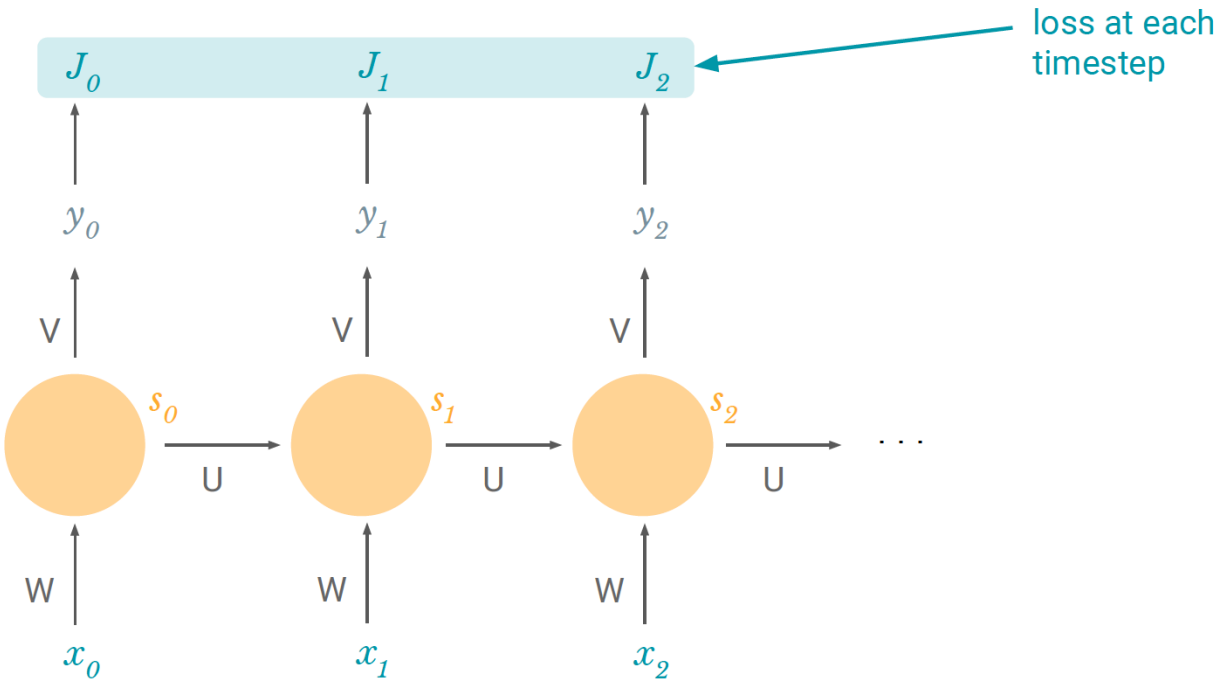
Deep Bi-directional RNNs



Backpropagation Through Time (BPTT)

Weights shared by all the time steps in the network

To calculate gradient at t=3, backpropagate 2 steps and sum up gradients



Problem with RNNs

Learning long-term dependencies

"I grew up in France … I speak fluent _____."

Vanishing/Exploding gradient problem

notice that the same weight matrix is multiplied at each time step during forward and backward propagation

Long Short Term Memory Networks (LSTMs)

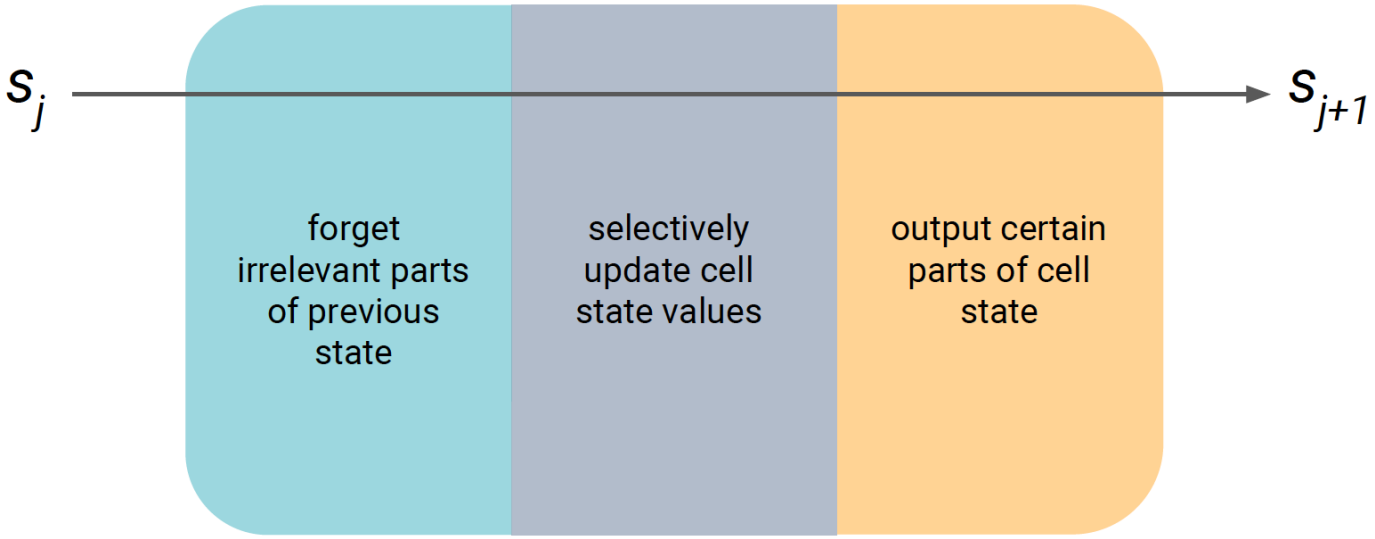
Avoid long term dependency problem

remember information for a long time

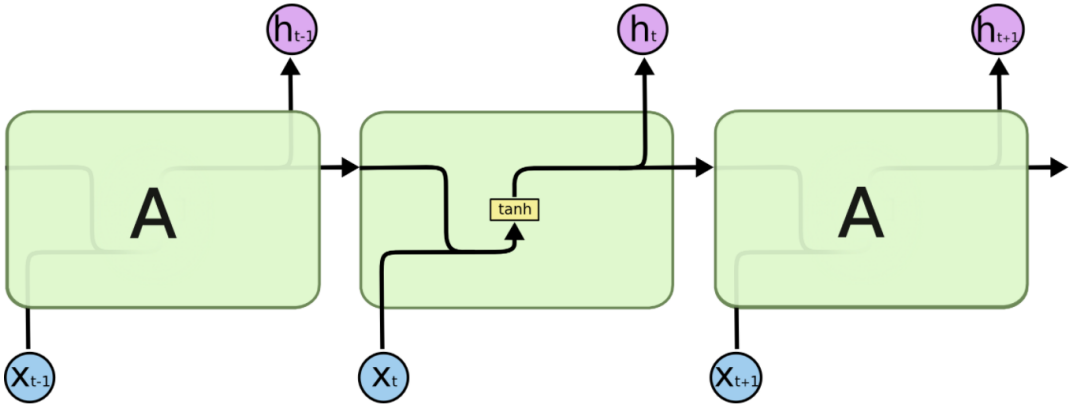
Idea: gated cells

complex node with gates controlling what information is passed through

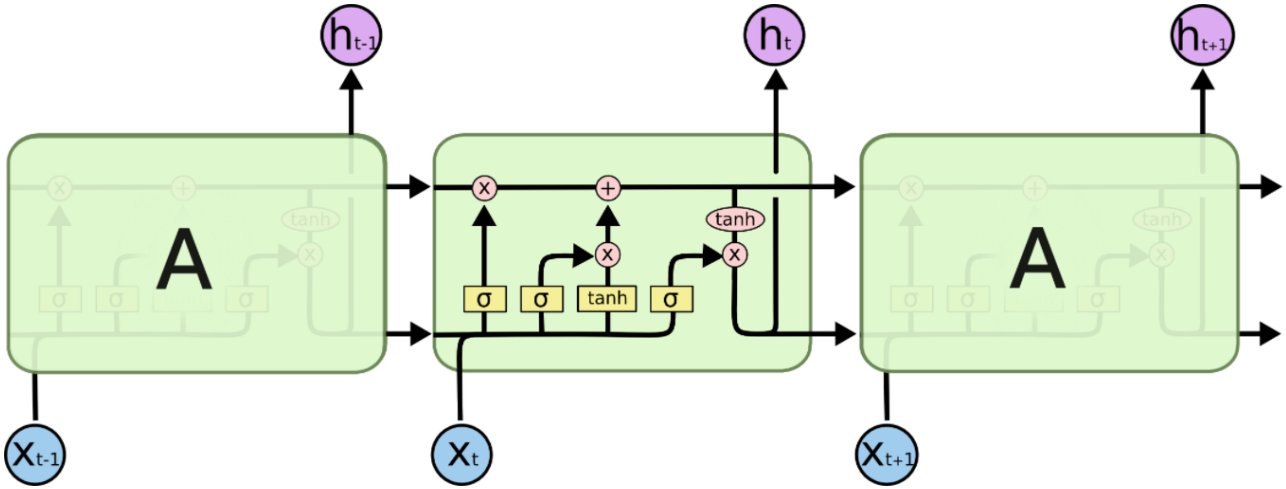
maintains an additional "cell state" - c_t



RNNs vs. LSTMs

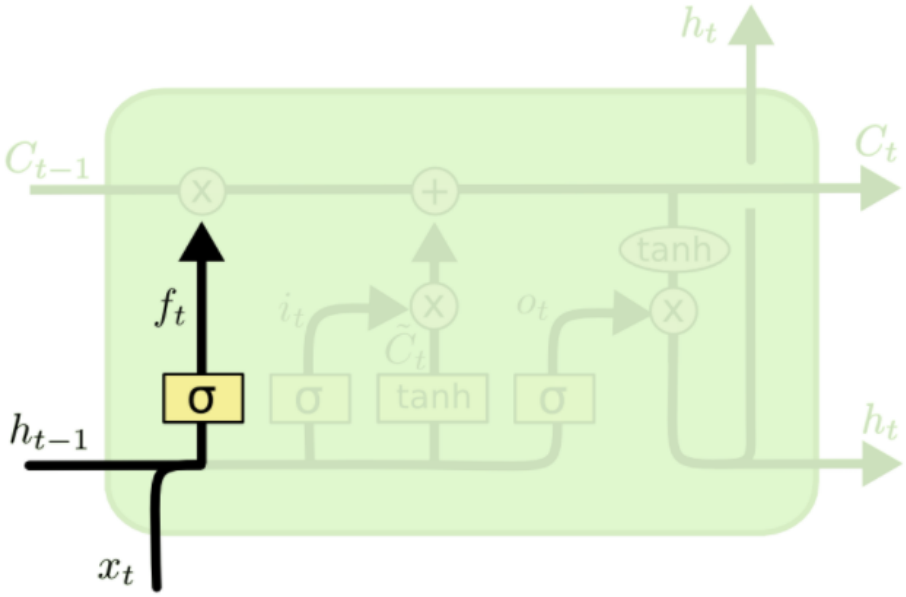


The repeating module in a standard RNN contains a single layer.



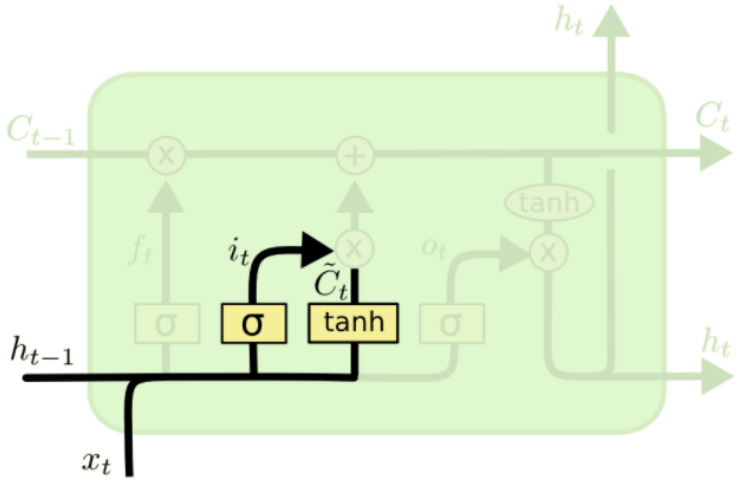
The repeating module in an LSTM contains four interacting layers.

Forget Gate



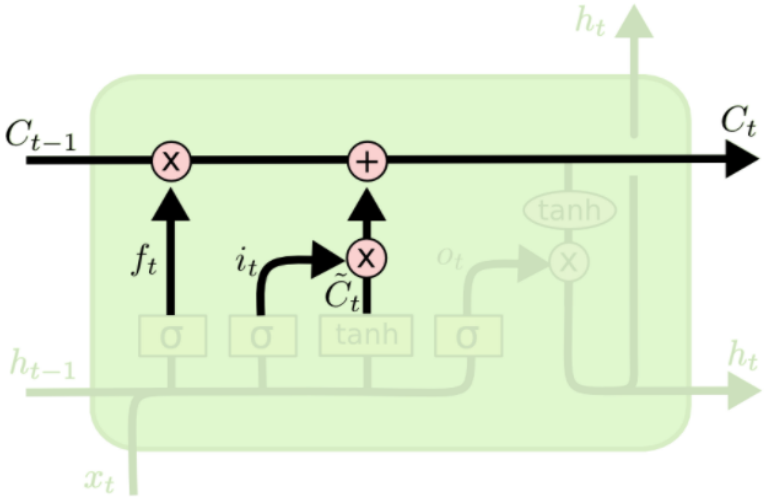
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Update Cell State



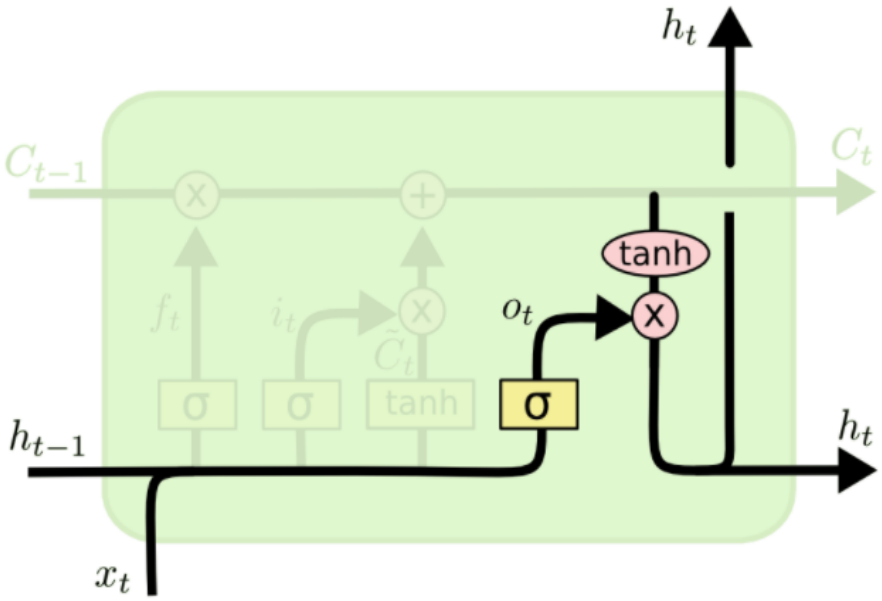
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

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



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

Output Gate



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

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

Source: Google

Tricks of the Trade

Activation function: try ReLU

prevents from shrinking gradients

Optimization algorithm: try Adam

computes adaptive learning rate; usually faster convergence

read: http://sebastianruder.com/optimizing-gradient-descent/index.html

Weight initialization: use Xavier initialization

make sure weights start out 'just right'

Prevent overfitting: dropout, L2 regularization

dropout prevents feature co-adaptation

remember to scale model weights at test time for droput

Tricks of the Trade (cont'd)

Random Hyperparameter Search

grid search is a bad idea; read: https://arxiv.org/abs/1206.5533

some hyper-parameters more important than others

Batch Normalization

make activations unit gaussian distribution at the beginning of the training

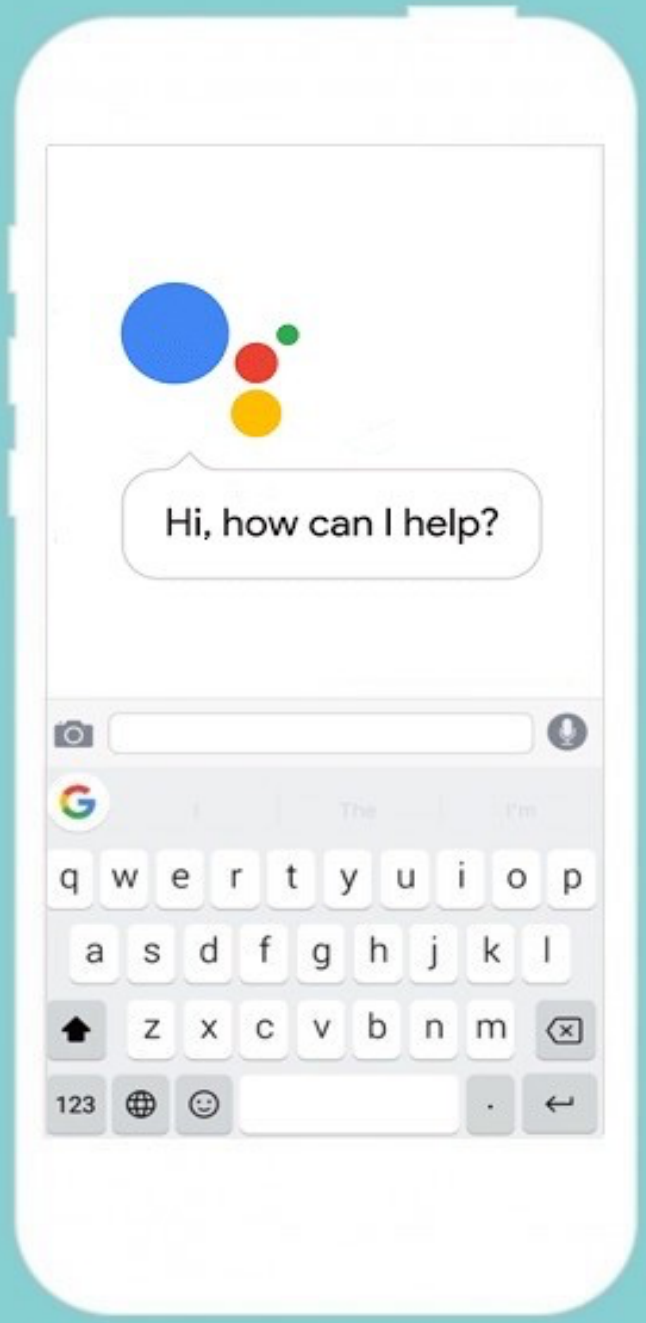
insert BatchNorm layer immediately after fully-connected/convolutional layers

Initialize recurrent weight matrix, W^{hx} & W^{hh}, to identity matrix

helps vanishing gradient problem. read: https://arxiv.org/pdf/1504.00941.pdf

Gradient clipping

helps exploding gradient problem



Questions?

Source: Google

Research:
Factoid Question Answering

Source: Google

Problem

Q: Who is the Falcons quarterback in 2012?

A: Matt Ryan

Q: Where did George Harrison live before he died?

A: Liverpool

Q: Who were the parents of Queen Elizabeth I?

A: Anne Boleyn, Henry VIII of England

Pretty difficult…

Google

Where did George Harrison live before he died?

All Images News Videos Shopping More Settings Tools

About 34,500,000 results (0.93 seconds)

Harrison died in 2001, aged 58, from lung cancer. He was cremated and his ashes were scattered in the Ganges and Yamuna rivers in **India**, in a private ceremony according to Hindu tradition. He left an estate of almost £100 million.

George Harrison - Wikipedia
https://en.wikipedia.org/wiki/George_Harrison

About this result Feedback

Approach

Q: Who were the parents of Queen Elizabeth I?

A: Anne Boleyn, Henry VIII of England

Entity: Queen Elizabeth I

Freebase Entity MID: *m.02rg_*

Relation: */people/person/parents*

Lookup Freebase: query (entityid, relation)

* Freebase is a large knowledge base.

Difficulties

No consistent way to do entity name to ID conversion

'JFK' could refer to a person, president, film, airport.

Evaluate correct answer

'Cuban Convertible Peso' vs. 'Cuban Peso'

Incomplete data

Freebase API now deprecated

Different versions of Freebase used for different datasets

Relation Prediction

- Dataset: Simple Questions
- Training set: ~76,000 examples
- Validation set: ~11,000 examples
- Number of classes: 1,837 relation types
- Model: Bi-directional LSTM (4 layers)
- Accuracy of validation set: ~81%

Source: Google