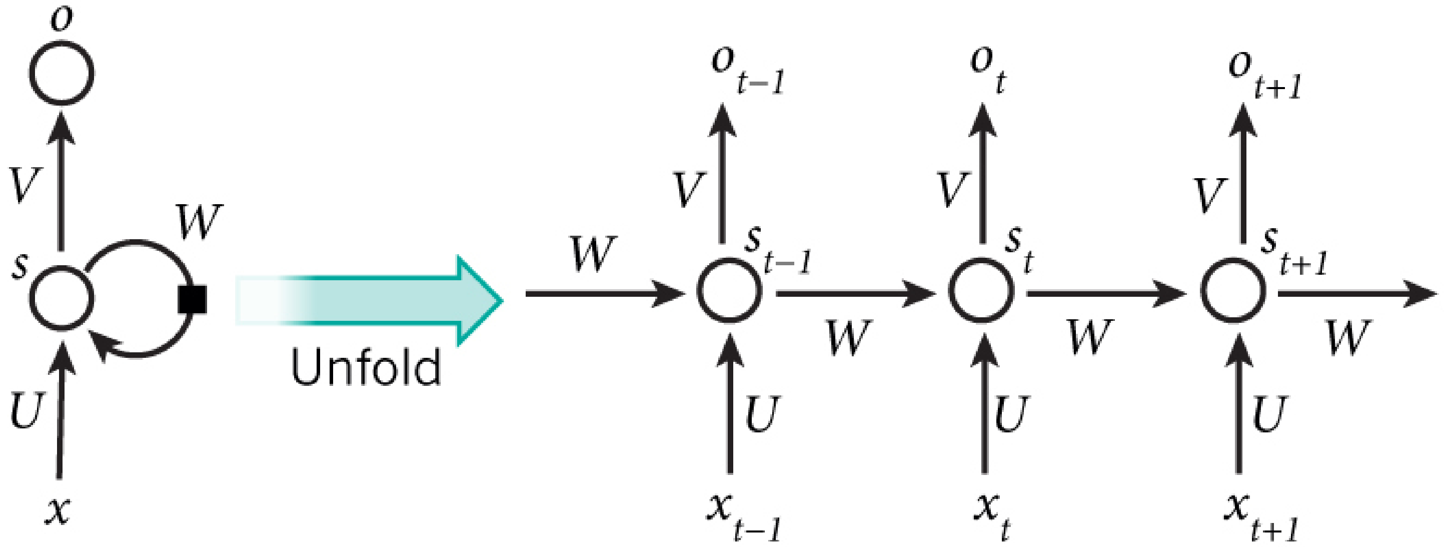
Recurrent Neural Networks

By: Leena Shekhar
Slides from: Heeyoung Kwon
Noah Weber
Stanford CS224N

Recurrent Neural Network

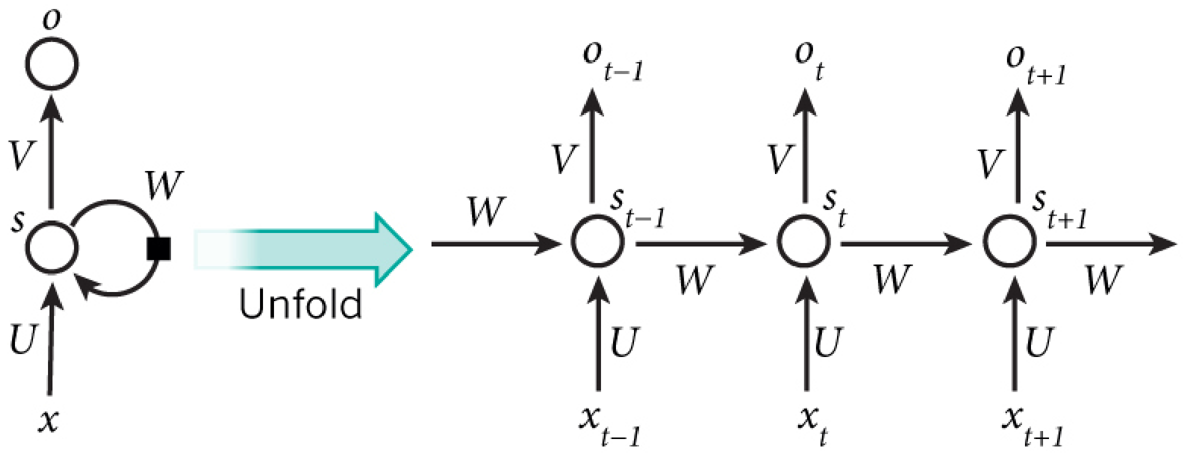
- A recurrent neural network (RNN) is a class of artificial neural network where connections between units form a directed cycle.



Picture from wildml.com

Recurrent Neural Network

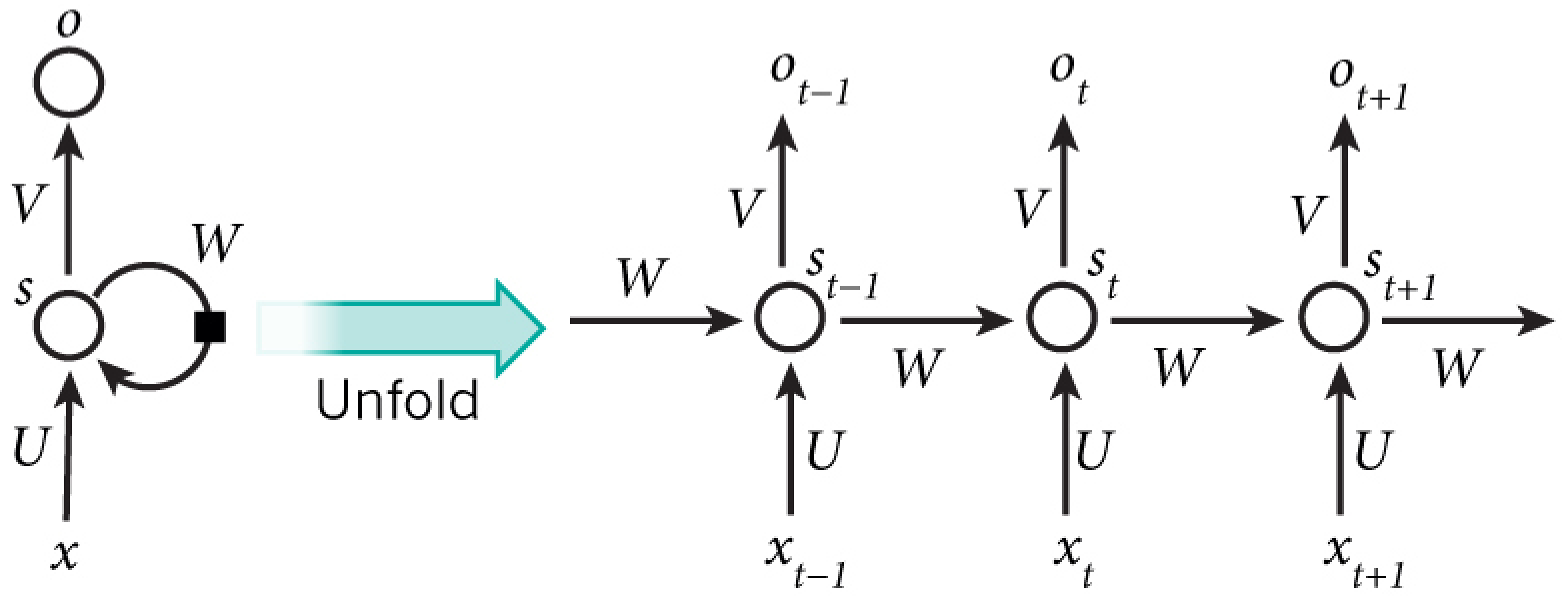
- At its simplest: A NN whose activations can be used again as input (s)
- Can think of it as the neural network keeping some type of state/memory of previous inputs
- Can be used for tasks involving sequential input/output



Recurrent Neural Network – Mathematical form

$$h_t = \phi(Ux_t + Wh_{t-1})$$

$$y_t = \phi(Vh_t)$$



Training RNNs to do good things

- We can unroll the RNN to get feed forward NN
- But the weight matrices , , are shared across all timesteps
- For each time step, compute the error, compute the gradient w respect to this error
- To get final gradient, sum over all gradients for each time step

Training RNNs to do good things is HARD

- RNNS are hard to train
  - Specifically, they don’t do what they expected to do (capture long term dependencies)

- Our gradients are vanishing or exploding!

Vanishing and Exploding Gradients

- There are many many multiplications of the same matrix
- If weights too small, vanishing gradients
- If weights too big….

RNN is not suitable for long-term dependencies

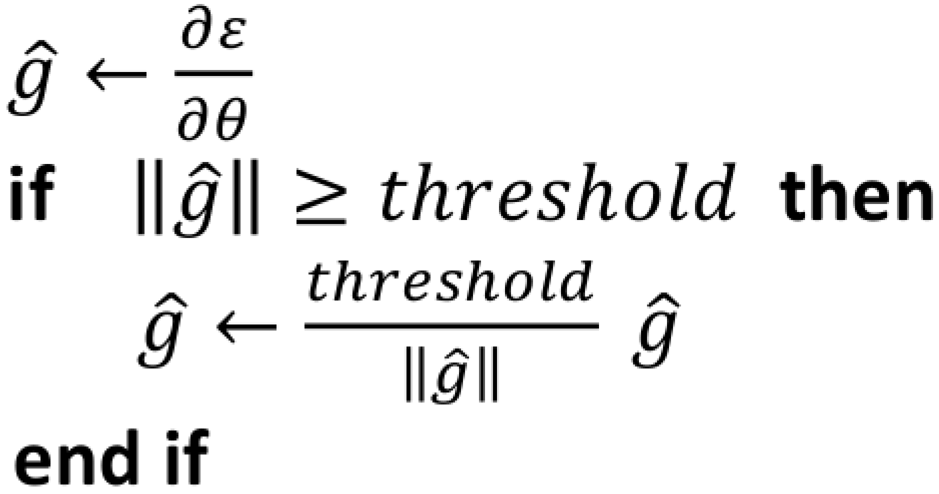
- Gradient Explodes
  - Large increase in the norm of the gradient
  - May lead to oscillating weights

- Gradient Vanishes
  - Long term components go exponentially fast to norm 0
  - Will learn nothing for long-term dependencies

https://weberna.github.io/blog/2017/11/15/LSTM-Vanishing-Gradients.html

Exploding Gradient

- Gradient Clipping
  - Rescale gradients

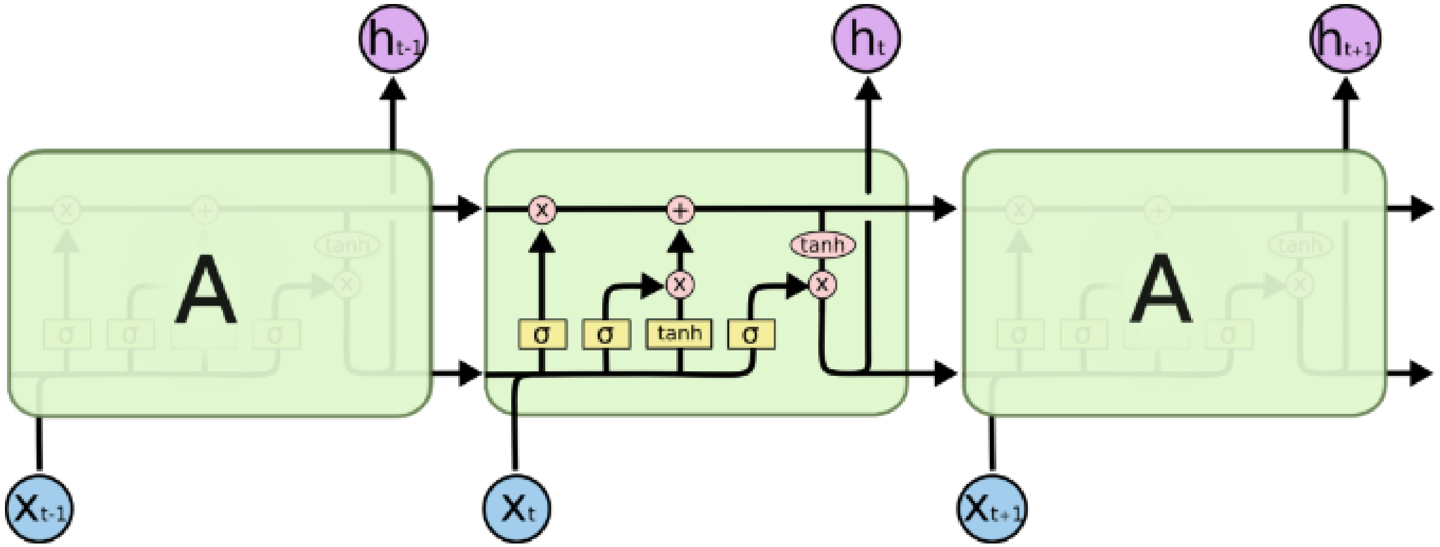


```
$\hat{g} \leftarrow \frac{\partial \varepsilon}{\partial \theta}$
if   $\|\hat{g}\| \geq threshold$  then
    $\hat{g} \leftarrow \frac{threshold}{\|\hat{g}\|} \hat{g}$
end if
```

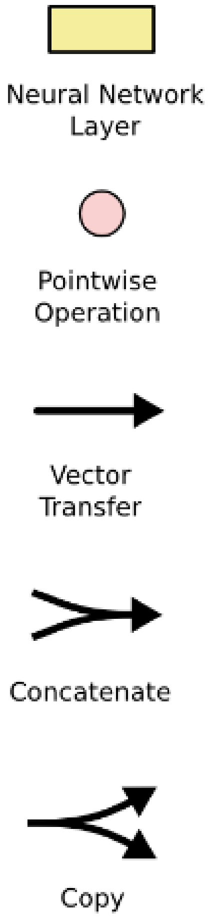
Vanishing Gradient

- More challenging as we cannot tell whether:
  - No dependency between t and t+n in data, or
  - Wrong configuration of parameters
- As remedy, LSTM was introduced

Structure of LSTM Cells



The repeating module in an LSTM contains four interacting layers.



Neural Network Layer

Pointwise Operation

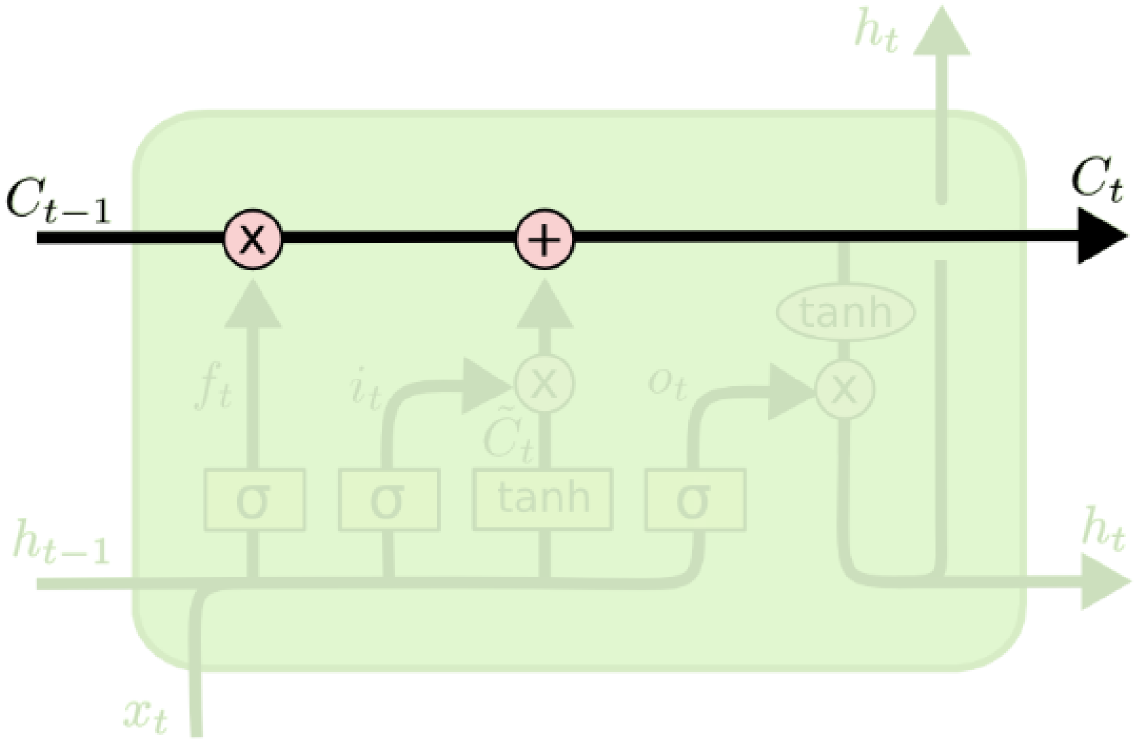
Vector Transfer

Concatenate

Copy

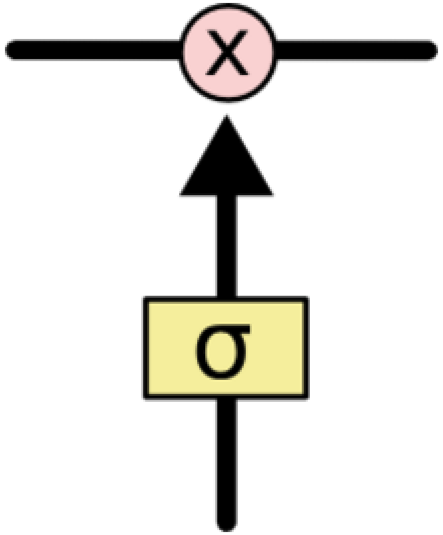
Images are from http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Core Idea – Cell State



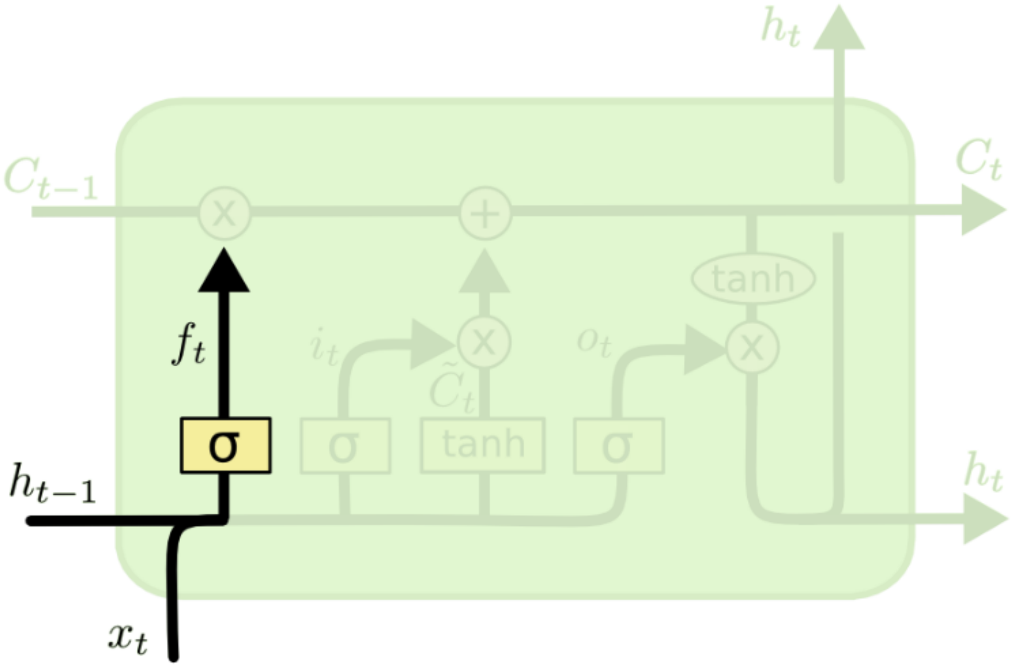
Images are from http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Core Idea – Gates



Images are from http://colah.github.io/posts/2015-08-Understanding-LSTMs/

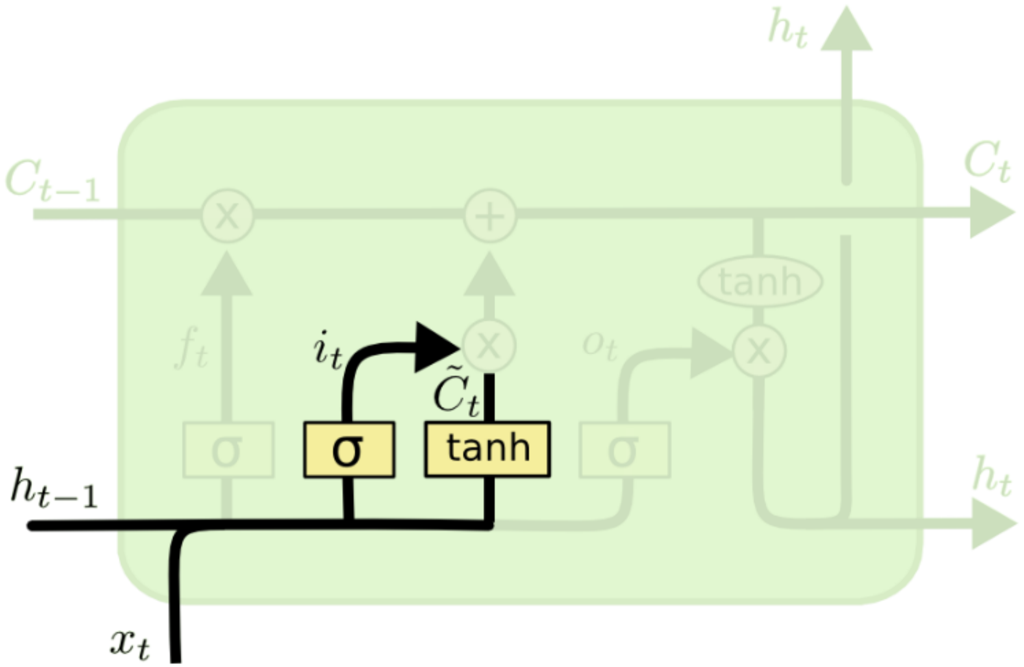
Forget gate



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

Images are from http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Input Gate

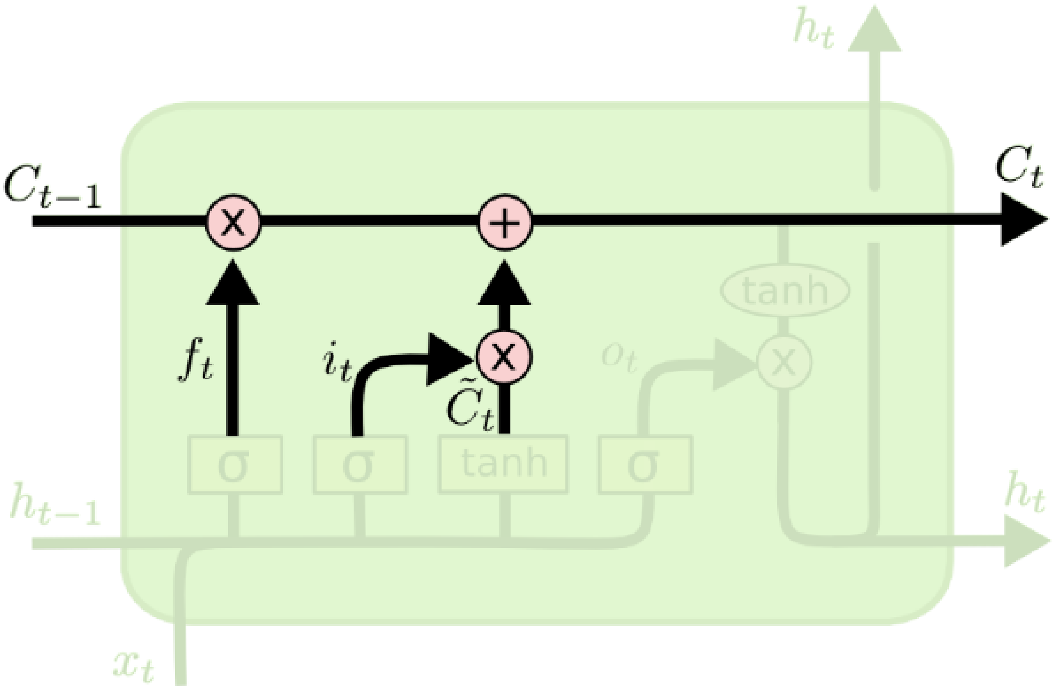


$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$

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

Images are from http://colah.github.io/posts/2015-08-Understanding-LSTMs/

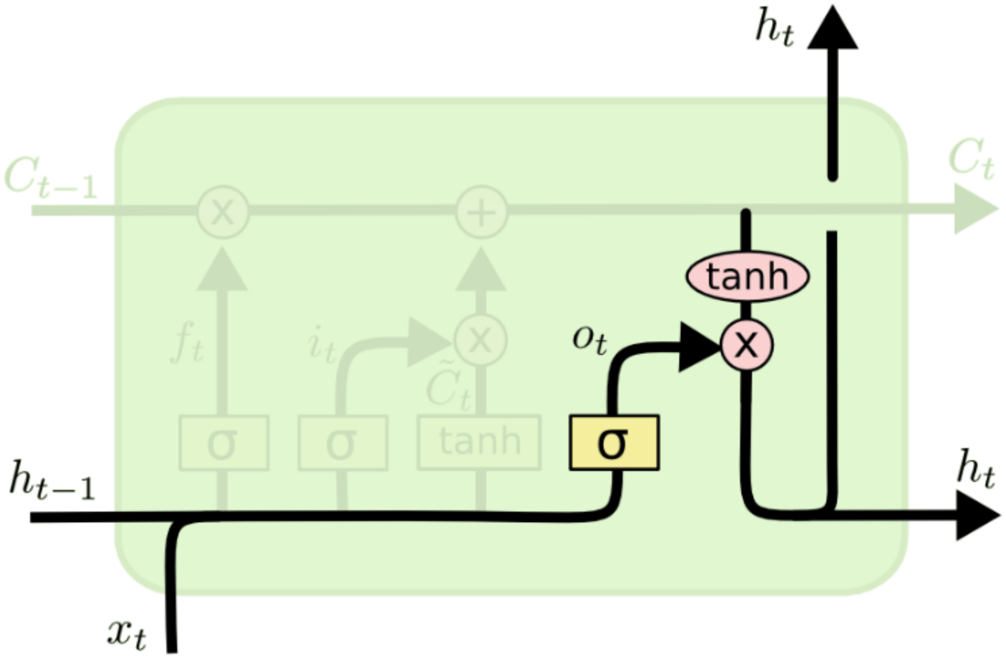
Update Cell State



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

Images are from http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Output Gate



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

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

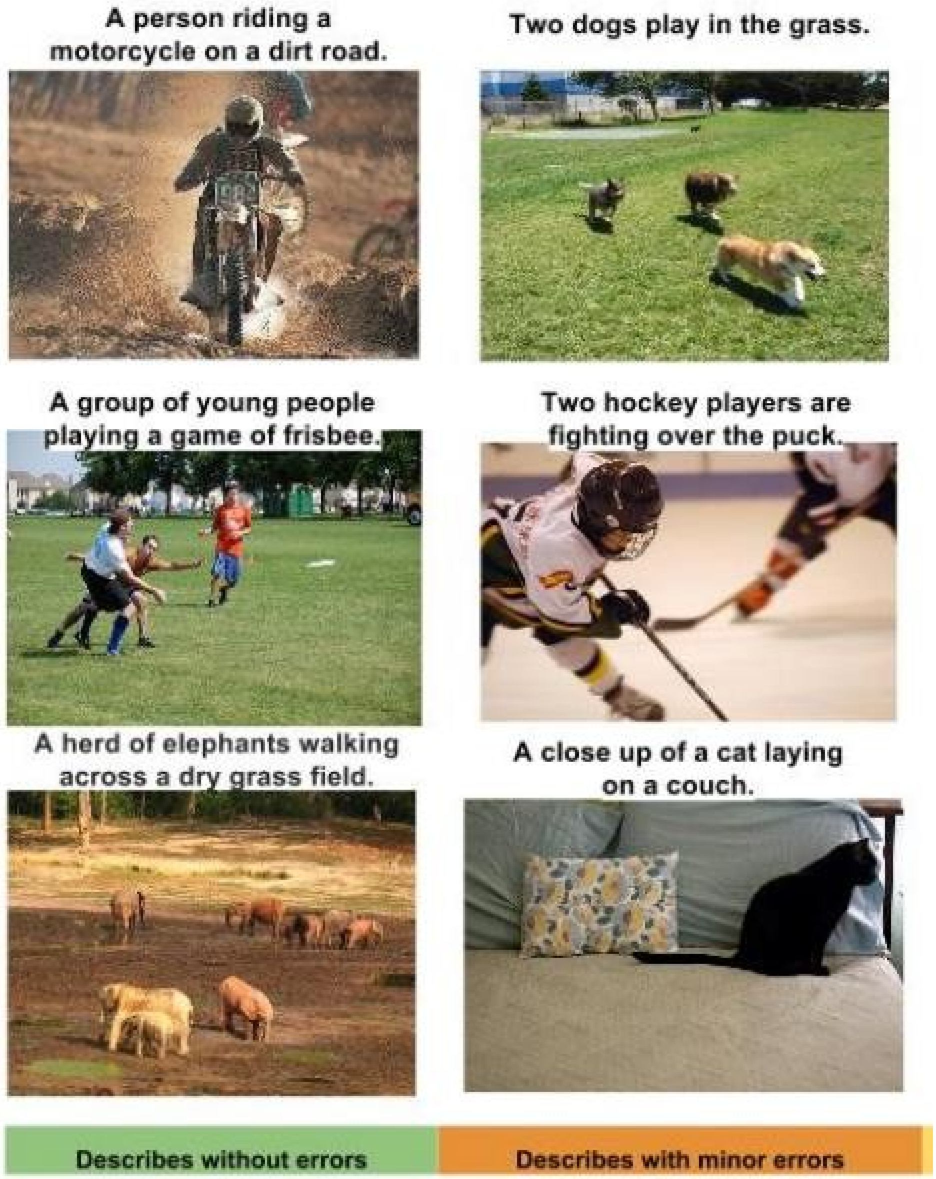
Images are from http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Applications

- Language Modeling
- Machine Translation
- Image Captioning
- Hand writing recognition / generation
- Question Answering (Answer Sentence Selection)
- Video to text

Applications

- Language Modeling
- Machine Translation
- Image Captioning
- Hand writing recognition / gener
- Question Answering (Answer Se
- Video to text



A person riding a motorcycle on a dirt road.

Two dogs play in the grass.

A group of young people playing a game of frisbee.

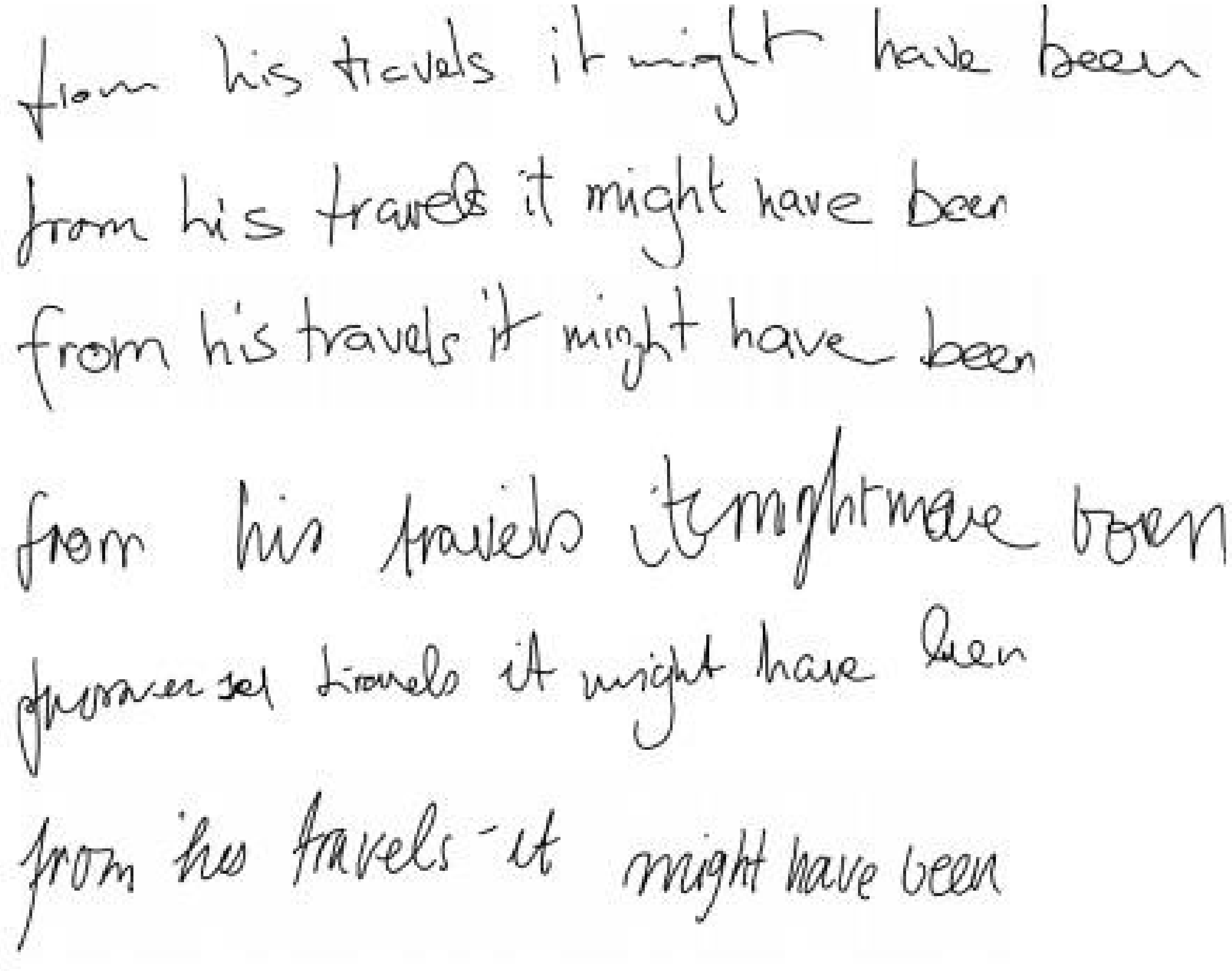
Two hockey players are fighting over the puck.

A herd of elephants walking across a dry grass field.

A close up of a cat laying on a couch.

Applications

- Language Modeling
- Machine Translation
- Image Captioning
- Hand writing recogni
- Question Answering
- Video to text



from his travels it might have been

from his travels it might have been

from his travels it might have been

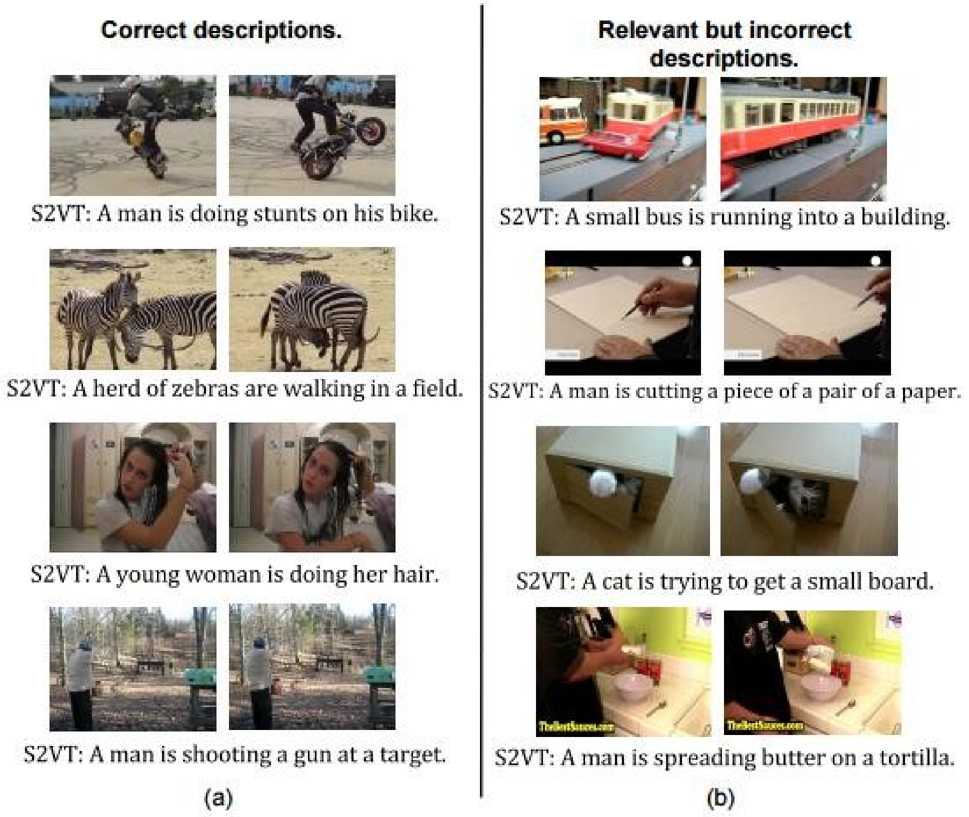
from his travels itmightmave been

[illegible] travels it might have been

from his travels - it might have been

Applications

- Language Modeling
- Machine Translation
- Image Captioning
- Video to text
- Question Answering
- Hand writing recognition / generation



Correct descriptions.

S2VT: A man is doing stunts on his bike.

S2VT: A herd of zebras are walking in a field.

S2VT: A young woman is doing her hair.

S2VT: A man is shooting a gun at a target.

(a)

Relevant but incorrect descriptions.

S2VT: A small bus is running into a building.

S2VT: A man is cutting a piece of a pair of a paper.

S2VT: A cat is trying to get a small board.

S2VT: A man is spreading butter on a tortilla.

(b)

Reference

Cho, Kyunghyun et al. “Learning Phrase Representations using RNN Encoder-Decoder for Statistica Translation.” EMNLP (2014).

Gers, Felix A. et al. “Learning to Forget: Continual Prediction with LSTM.” Neural Computation 12 (2 2471.

Graves, Alex. “Generating Sequences With Recurrent Neural Networks.” CoRR abs/1308.0850 (201

Greff, Klaus et al. “LSTM: A Search Space Odyssey.” CoRR abs/1503.04069 (2015): n. pag.

Hochreiter, Sepp and Jürgen Schmidhuber. “Long Short-Term Memory.” Neural Computation 9 (199 1780.

Józefowicz, Rafal et al. “An Empirical Exploration of Recurrent Network Architectures.” ICML (2015)

Pascanu, Razvan, Tomas Mikolov, and Yoshua Bengio. "On the difficulty of training recurrent neural networks." *ICML (3)* 28 (2013): 1310-1318.

Venugopalan, Subhashini et al. “Sequence to Sequence -- Video to Text.” 2015 IEEE International C Computer Vision (ICCV) (2015).

Vinyals, Oriol et al. “Show and tell: A neural image caption generator.” 2015 IEEE Conference on C Vision and Pattern Recognition (CVPR) (2015).

Yin, Wenpeng et al. “Comparative Study of CNN and RNN for Natural Language Processing.” (2017