VTC Academy
Think Ahead, Beyond Boundaries

Developed by COTAI
CENTER OF TALENT IN AI

Recurrent Neural Networks: LSTM, GRU

Hung Ngo

COTAI

hung.ngo@cot.ai | FB: curiousAI

Past: AINovation, MLR Stuttgart, IDSIA

VTCA-COTAI AI Foundations for Practitioners - Week 4.

Outline

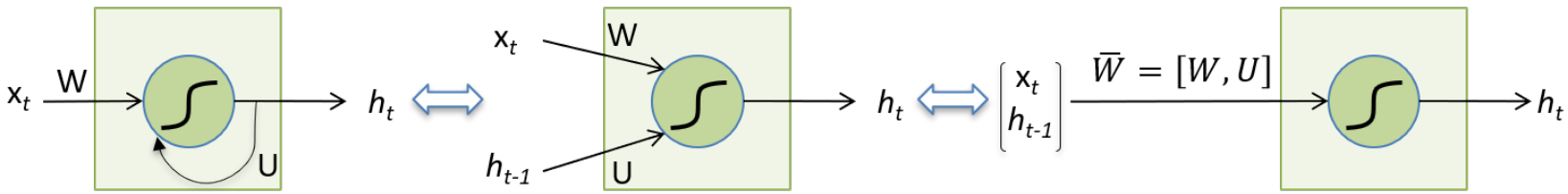
Motivation

How to learn from *sequential* data of *variable* lengths?

- Language understanding: text and speech recognition, translation, and generation, dialog systems, etc.
- Visual understanding: image captioning, video-based object recognition and tracking, learning to look, VQA, etc.
- Time series forecast & anomaly detection.
- Planning: robotic control, game play, resource scheduling, etc.
- Protein homology detection

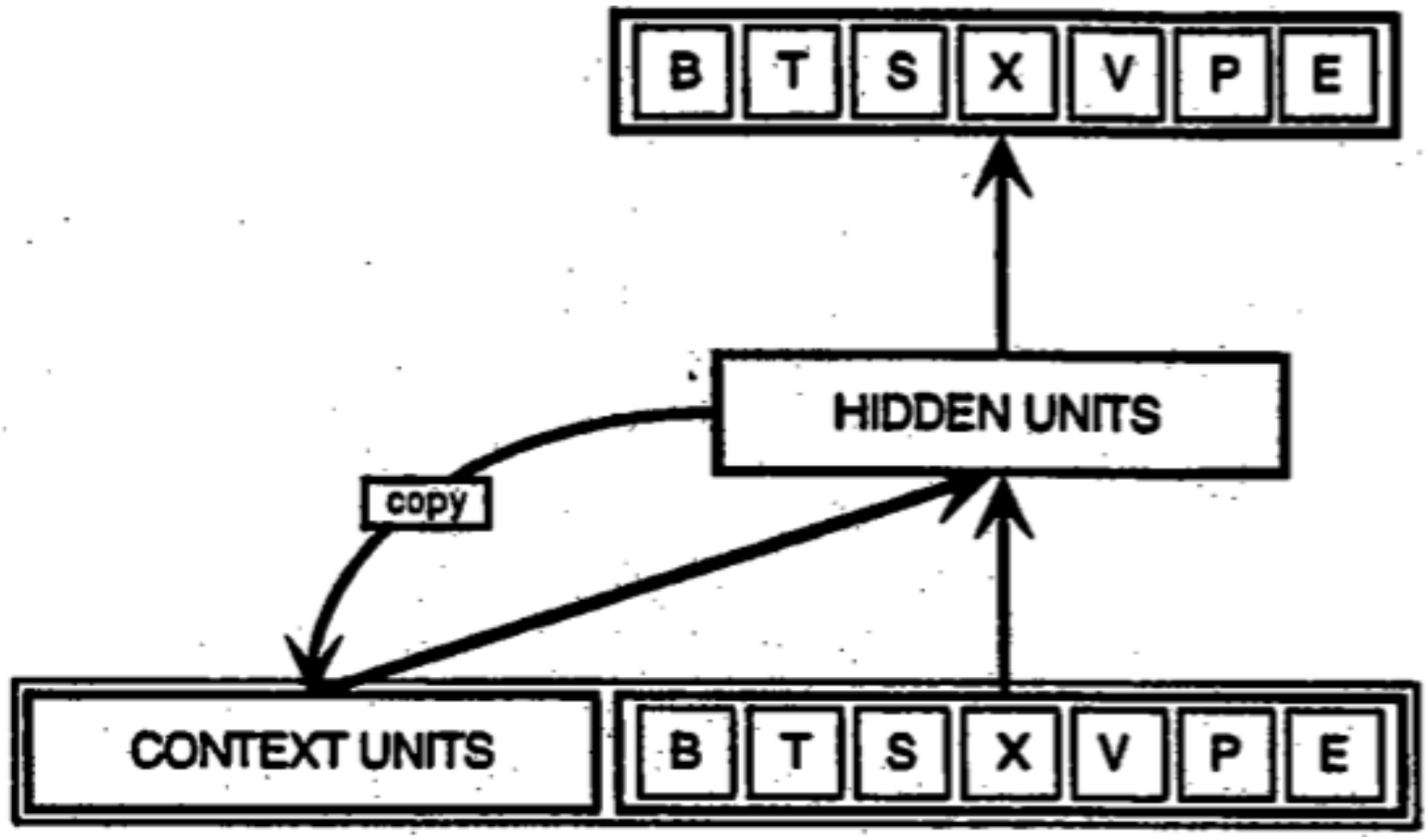
Recurrent Neural Network (RNN)

- RNN *vanilla* (basic) unit



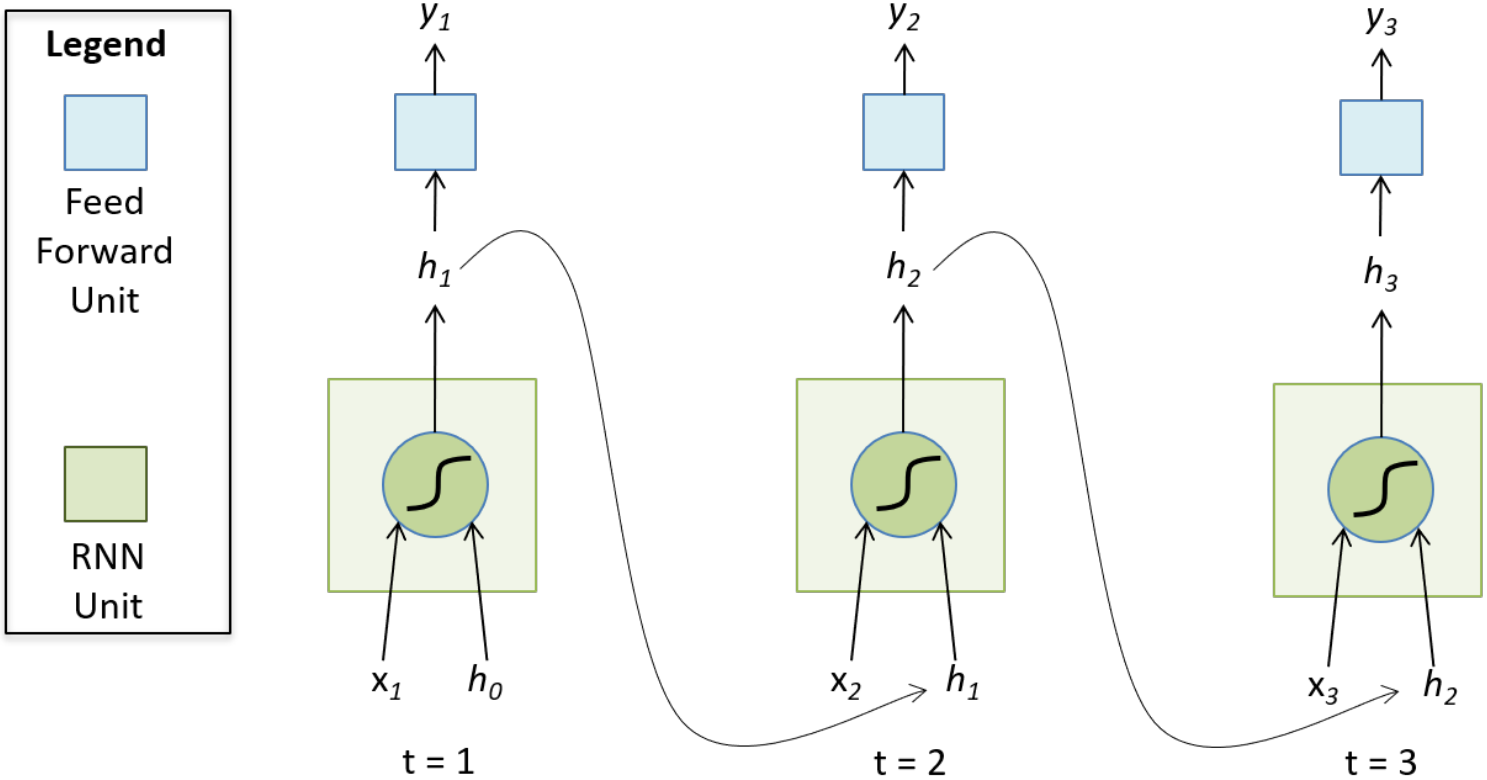
- Feed-forward neural network: $h_t = \gamma(Wx_t + b)$
  $x_t \in \mathbb{R}^d, h_t \in \mathbb{R}^n, \gamma(\cdot) = \{\text{sigmoid}, \tanh, \text{softmax}, \text{ReLU}, \ldots\}$
- RNN: $h_t = \sigma(Wx_t + Uh_{t-1} + b) = \sigma(\bar{W} \begin{bmatrix} x_t \\ h_{t-1} \end{bmatrix} + b)$
  with $\bar{W} = [W, U], W \in \mathbb{R}^{n \times d}, U \in \mathbb{R}^{n \times n}, b \in \mathbb{R}^n$.
- Historical (contextual, memory) information h_{t-1} is shifted (translated) by latent embedding of x_t: $Wx_t + Uh_{t-1} + b$.
- Weight-sharing: *same* weights $\bar{W}$ are *copied* for all time steps.

Early idea of a simple RNN by Elman



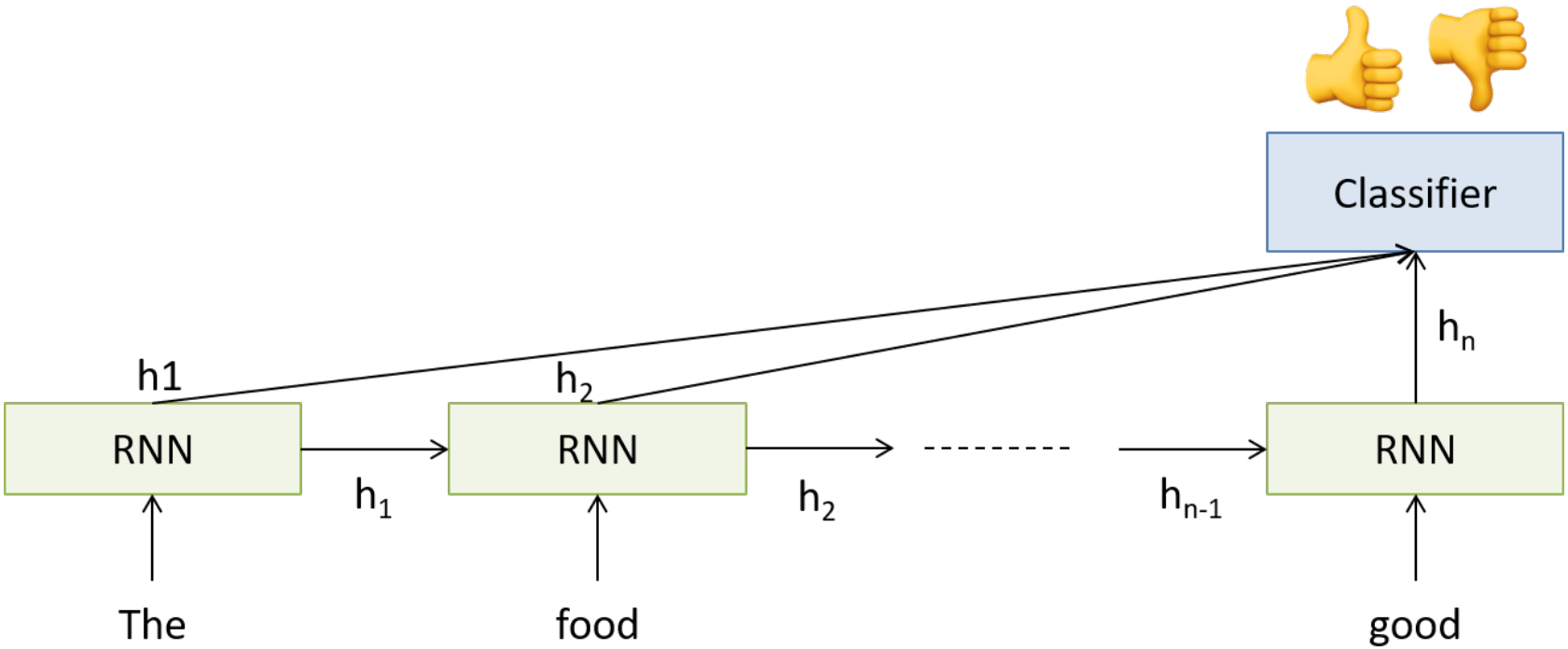
RNN unfold in time

"Weight sharing" through time ⇒ very efficient learning.

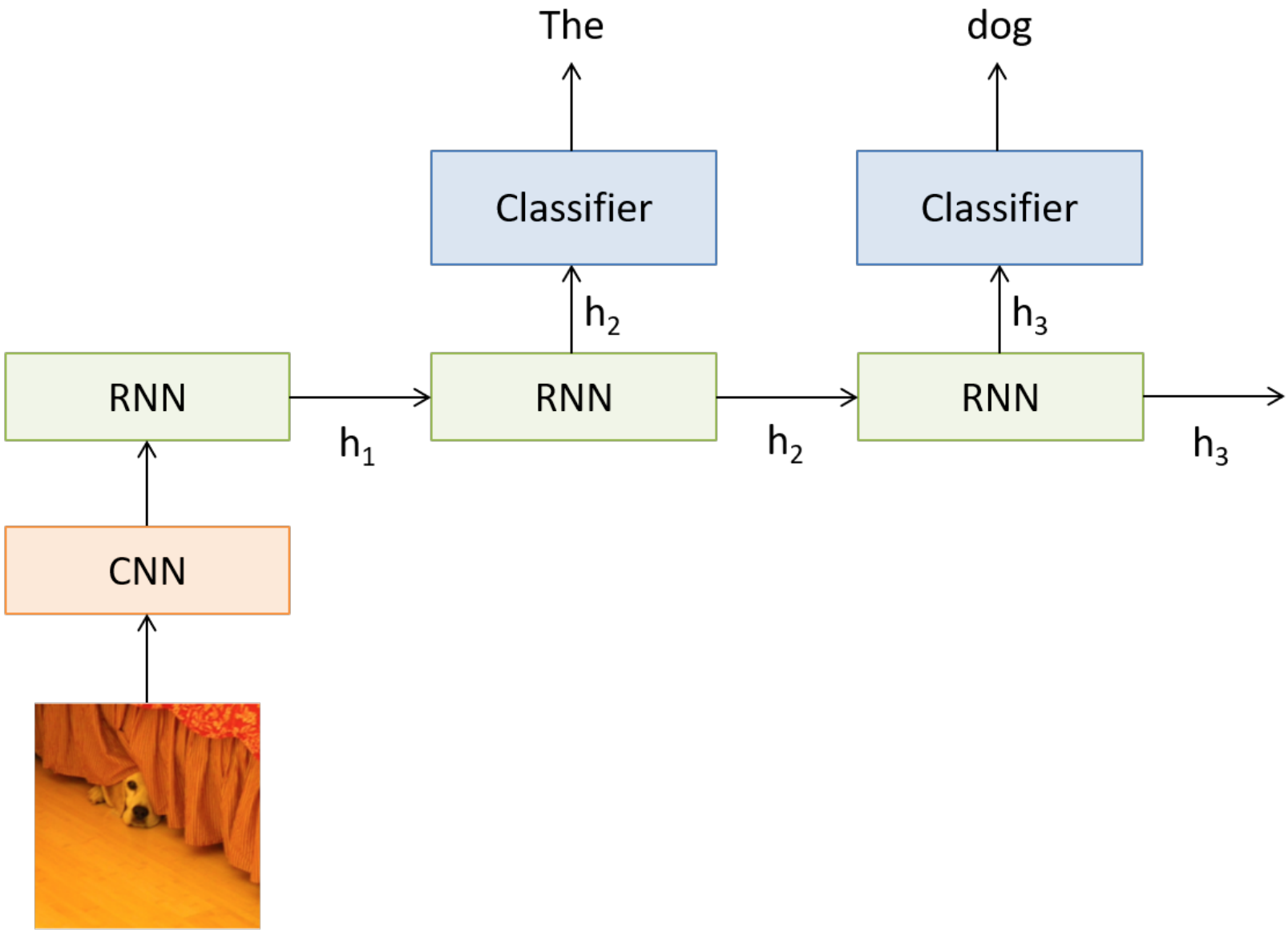


Slides adopted from Arun Mallya's lectures.

Example: RNNs for sentiment classification



Example: RNNs for image captioning



Example: RNNs for image captioning

A person riding a motorcycle on a dirt road.

Two dogs play in the grass.

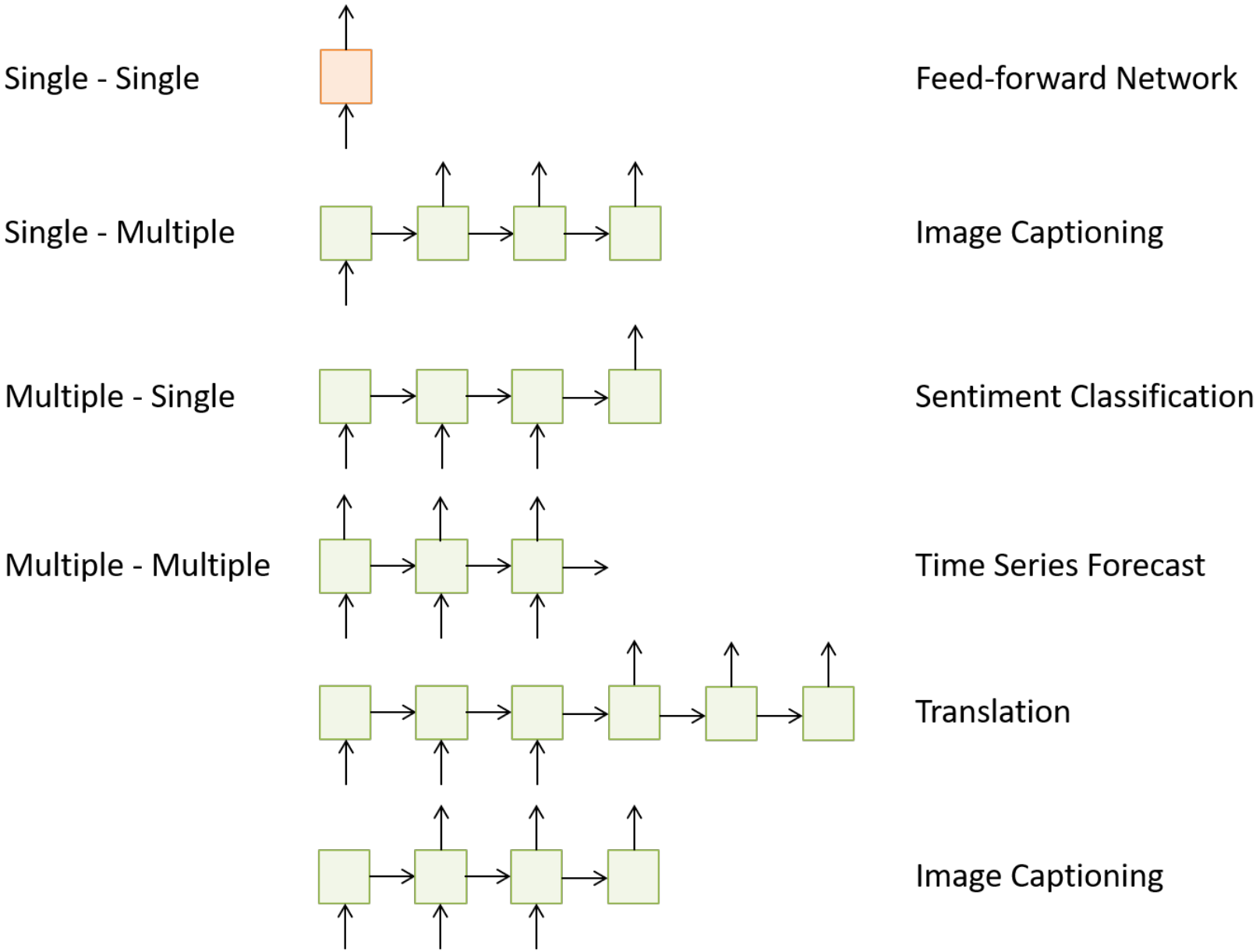
A herd of elephants walking across a dry grass field.

A group of young people playing a game of frisbee.

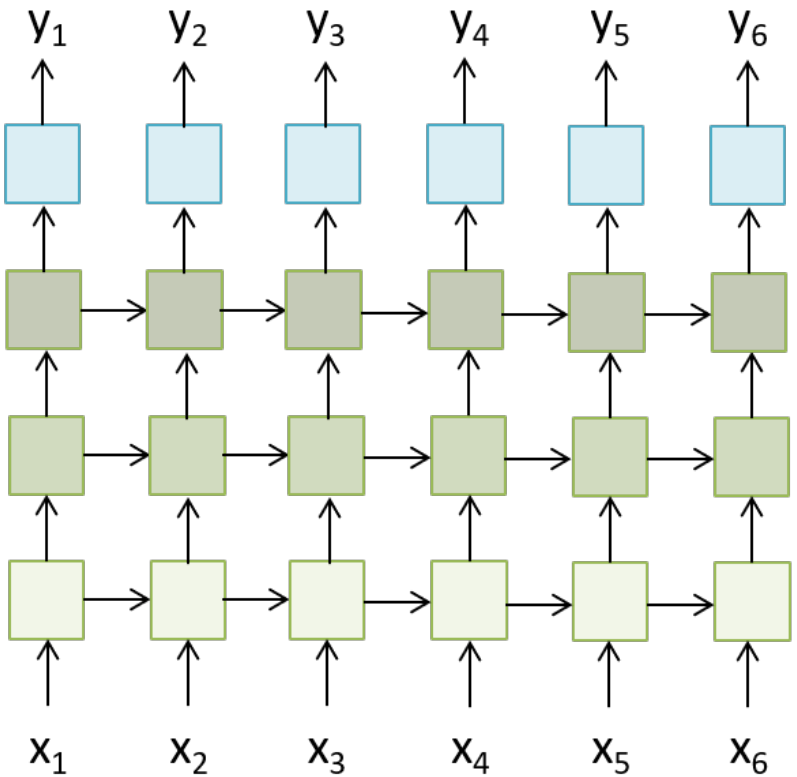
Two hockey players are fighting over the puck.

A close up of a cat laying on a couch.

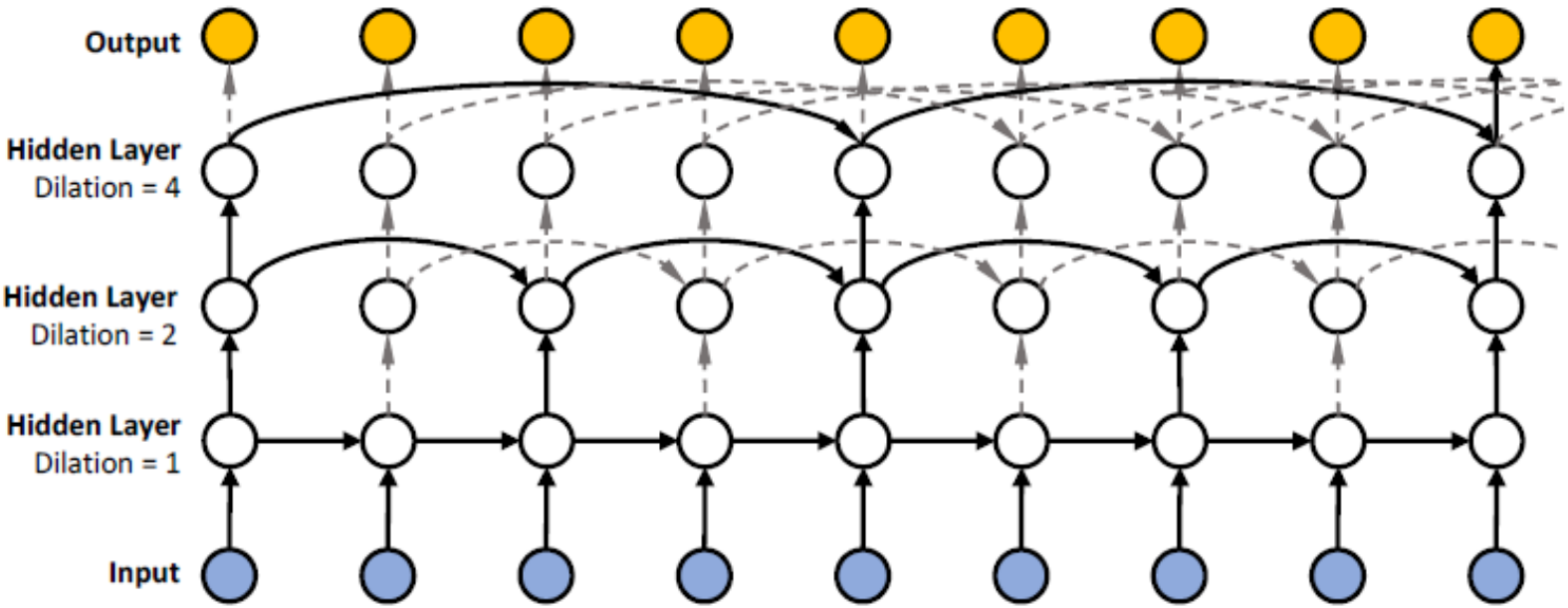
RNNs models & applications



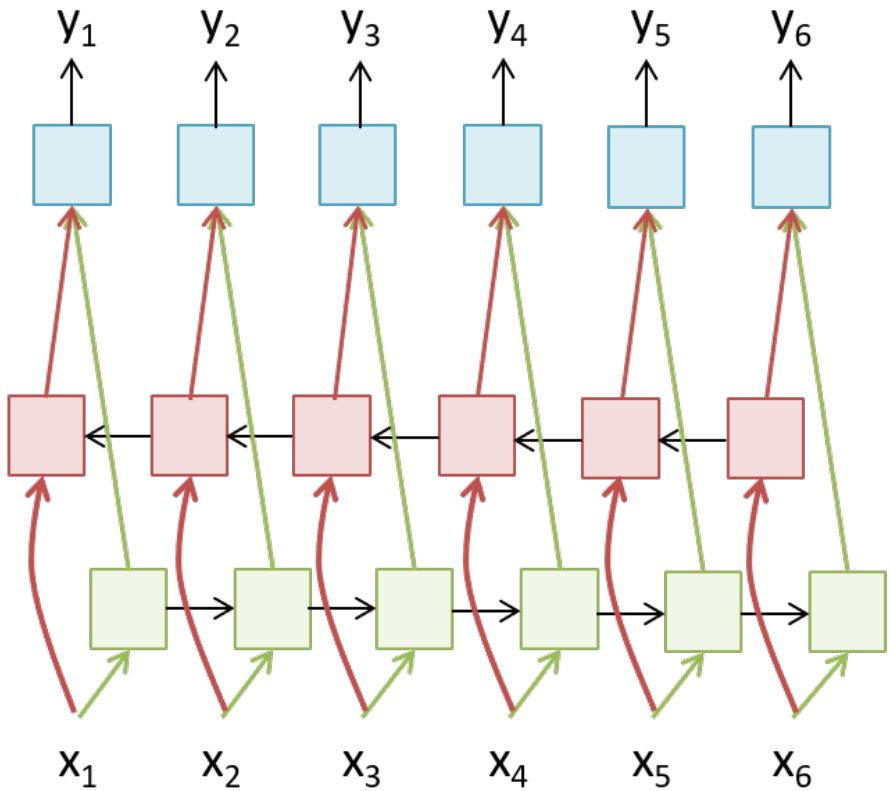
RNNs models: deep (stacked, multi-layer)



RNNs models: deep + skipped (dilated)



RNNs models: bi-directional



Popular in speech recognition tasks.

RNNs: Training

Gradient-based method: back-propagation through time (BPTT)

1. Treat the unfolded network as a big feed-forward network.
2. The whole input sequence is given to the FFNN.
3. The weight updates are computed for each copy in the unfolded network using the usual back-propagation method.
4. All the updates are then summed (or averaged) and then applied to the RNN (shared) weights.

RNNs: Training difficulties

Learning with *vanilla* RNNs on long sequences is a hard:

- Complex dependencies.

  Backpropagating gradients for *all* time steps often requires infeasibly large amounts of memory, so essentially every implementation of a recurrent model *truncates* the model and only backpropagates gradient k times steps: BPTT(k). See R2RT blogpost.

- Vanishing/exploding gradients.

  The product of repeated matrix multiplications $W^t x_t$ may shrink to zero or explode to infinity.

- Inefficient parallelization.

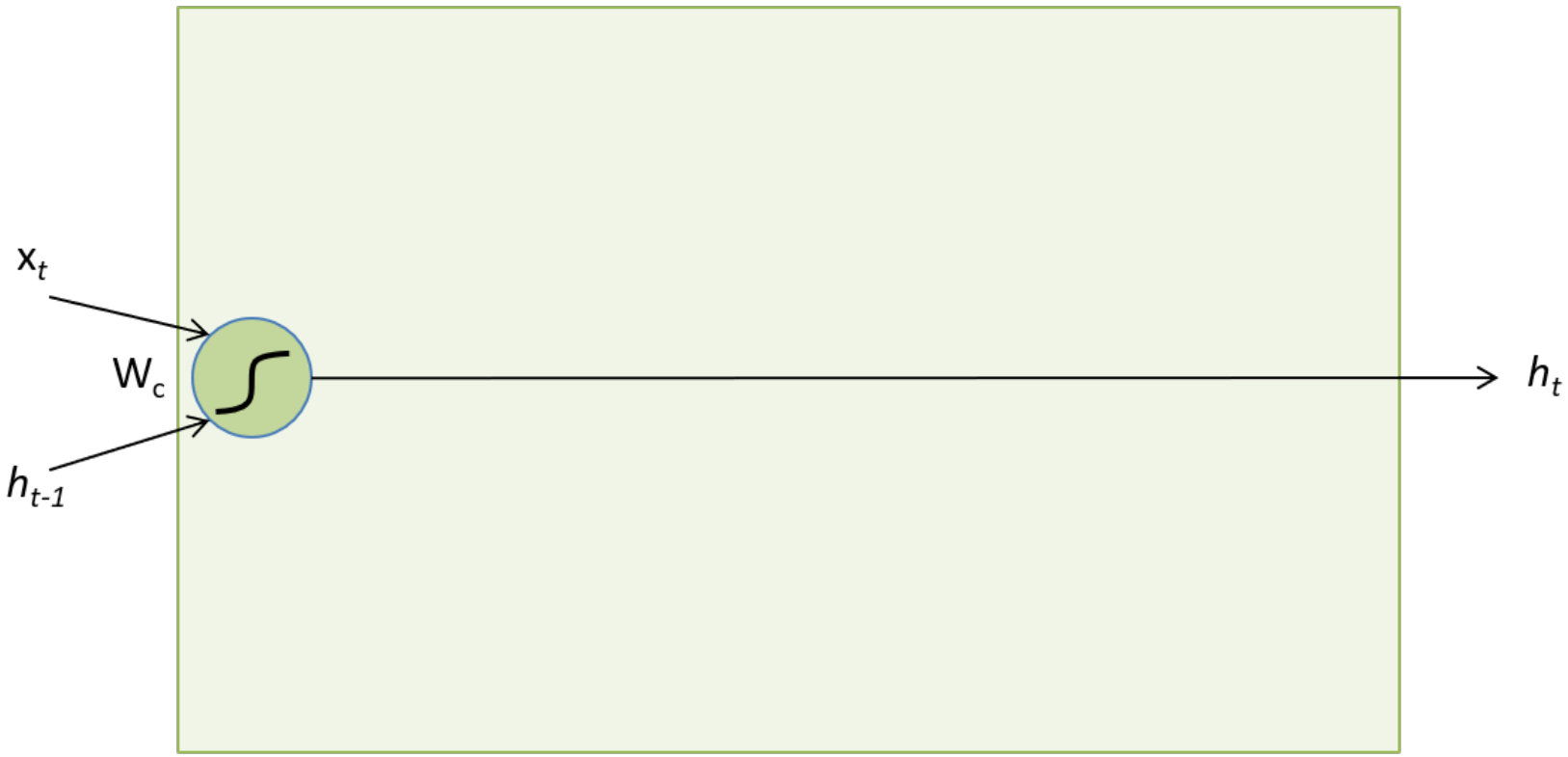
Outline

From RNNs to LSTM

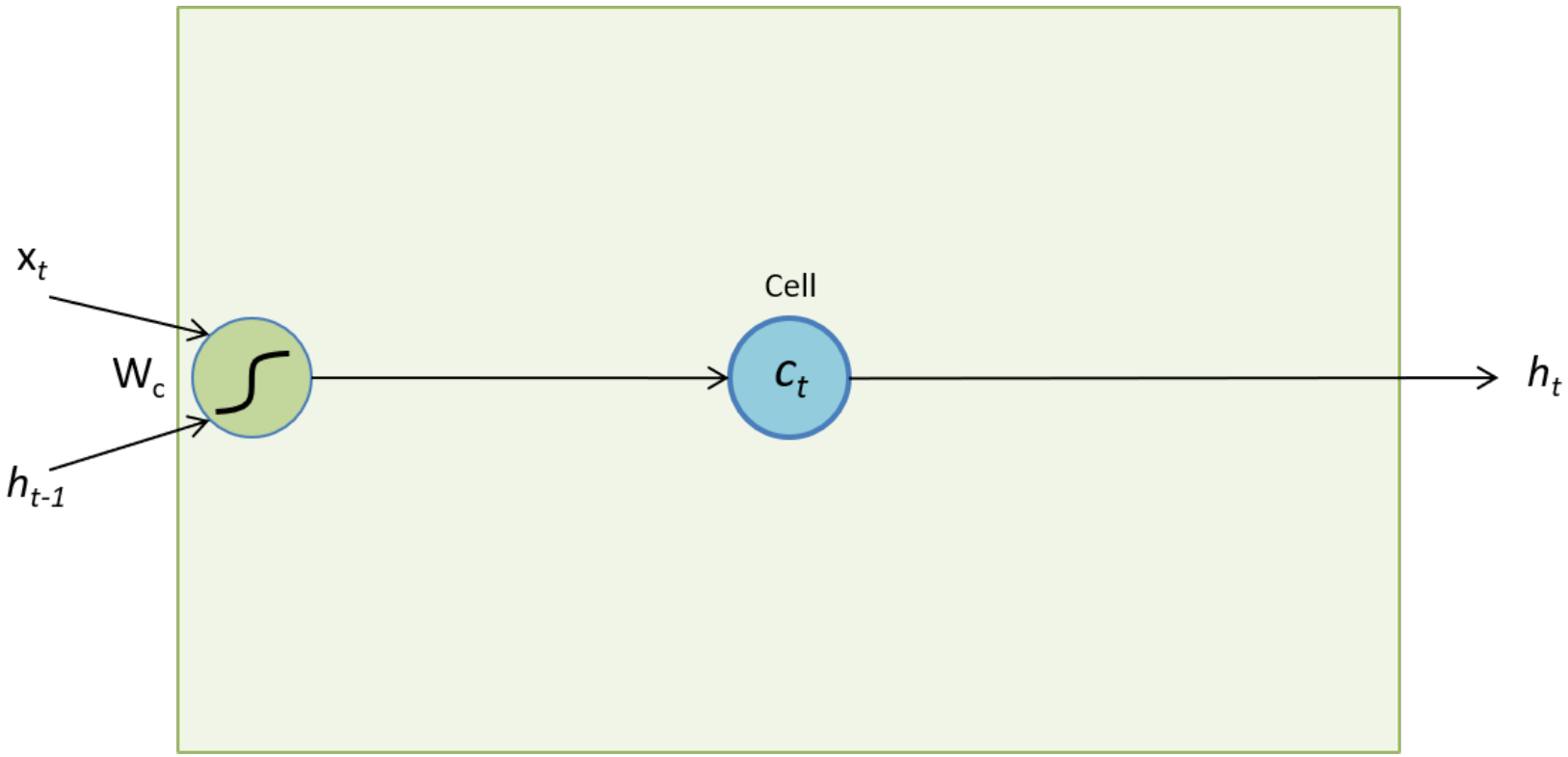
Best fix for vanila RNNs: LSTM with memory cell & update gates.

- Using an identity relationship between the hidden states: $h_t = h_{t-1} + Wx_t \Rightarrow$ the gradient does not decay as the error is propagated all the way back, aka "constant error flow".
- The LSTM uses a *memory cell* that acts like an accumulator (contains the identity relationship) over time.
- The LSTM uses the idea of "constant error flow" for RNNs to create a "constant error carousel" (CEC) which ensures that gradients don't decay.
- An LSTM with large positive forget gate bias works best!
- Exploding gradients: controlled by clipping gradient. (Recently: weight initialization, batch normalization, and various improved activation functions e.g. ReLU, ELU, etc.)

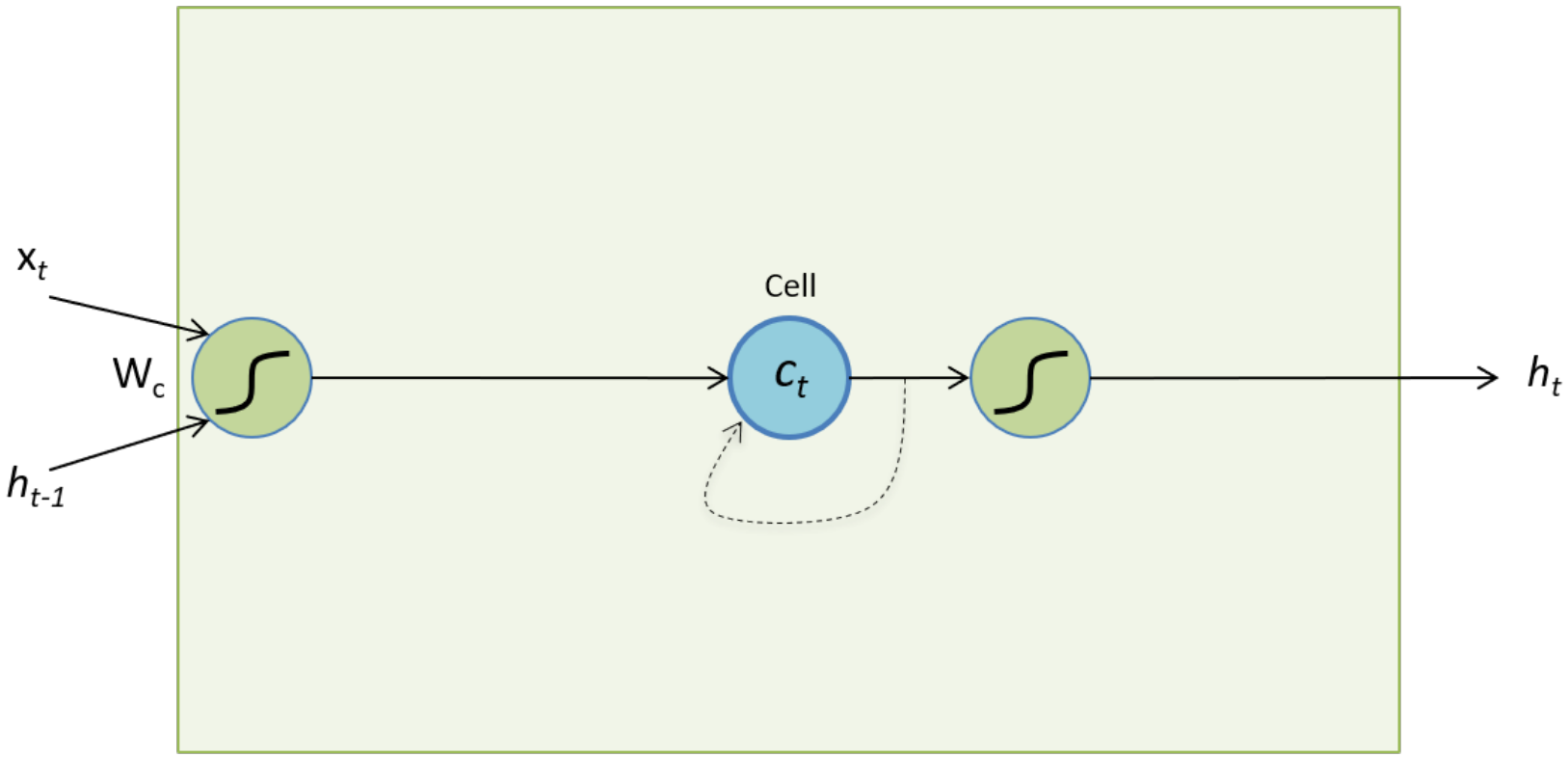
LSTM memory cell and update gates



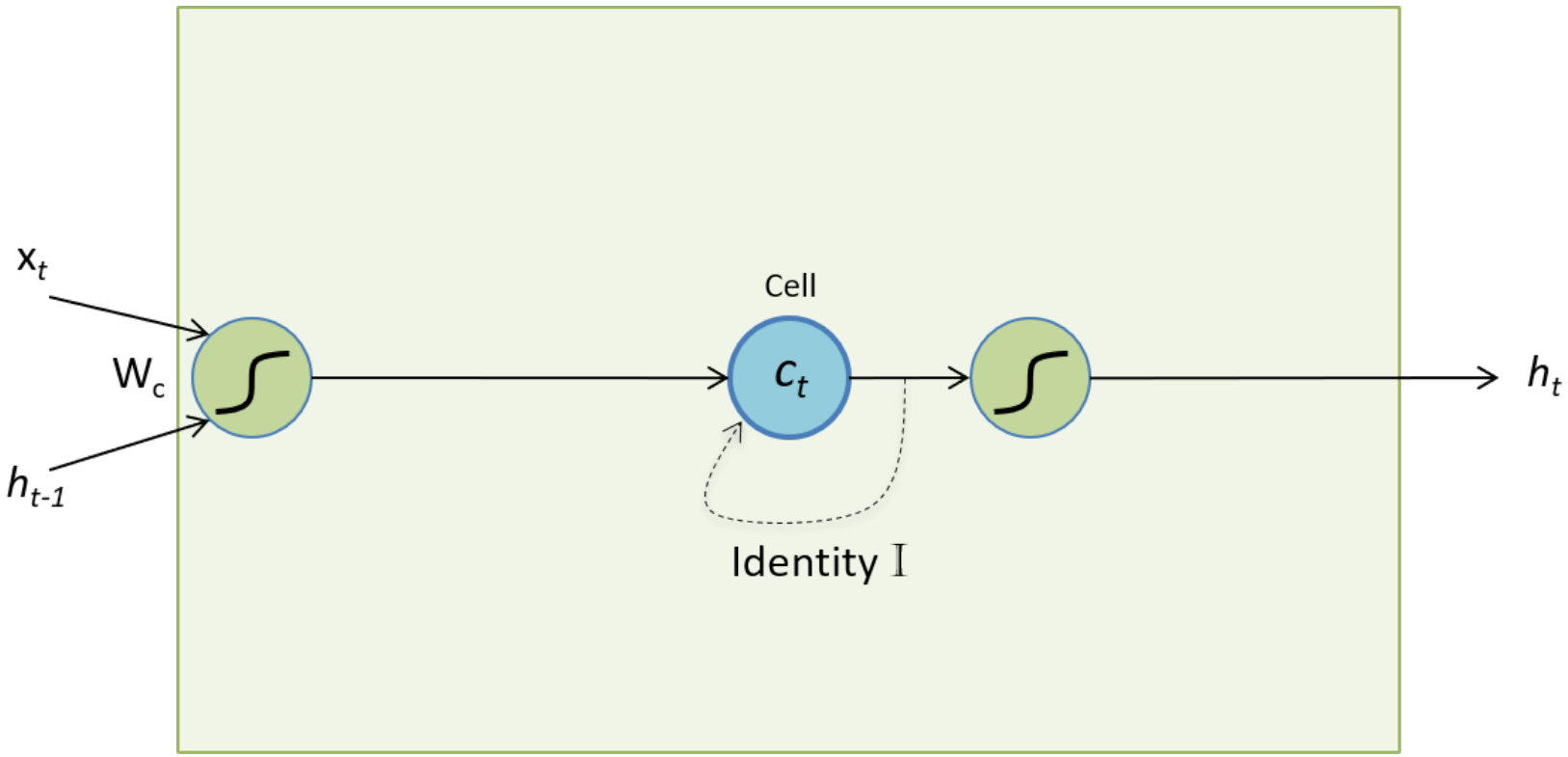
LSTM memory cell and update gates



LSTM memory cell and update gates

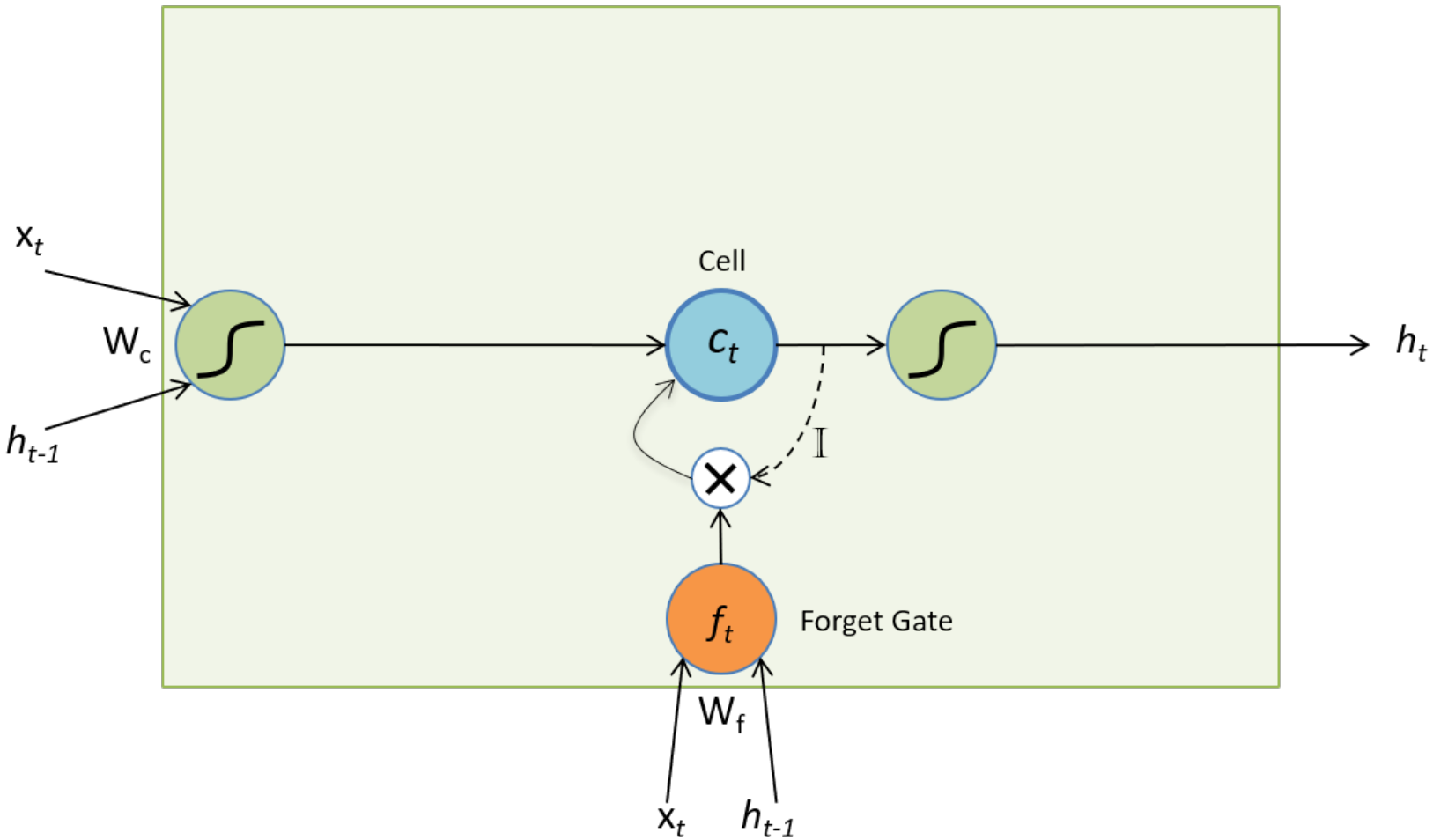


LSTM memory cell and update gates

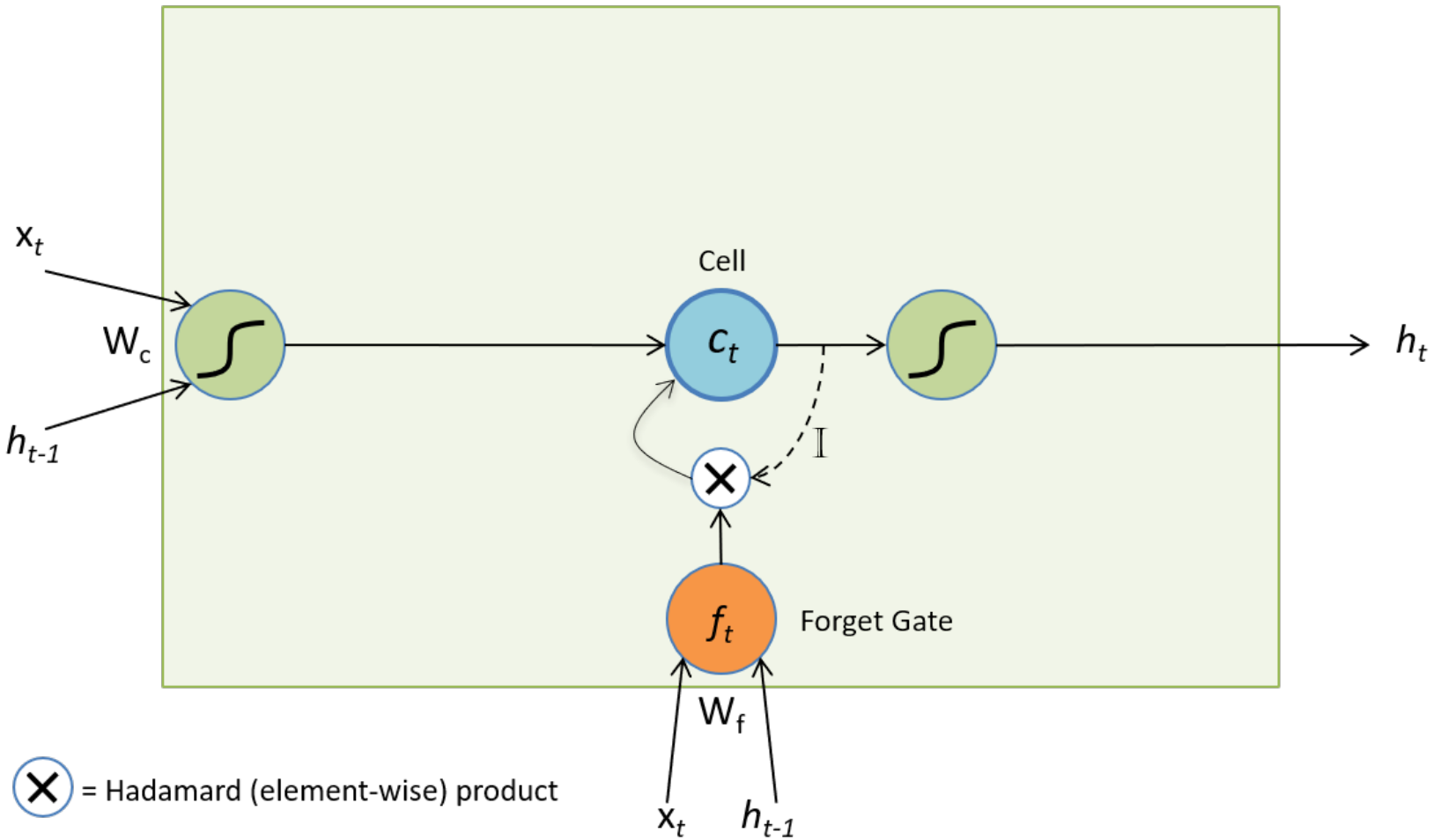


"Constant Error Carousel" (CEC)

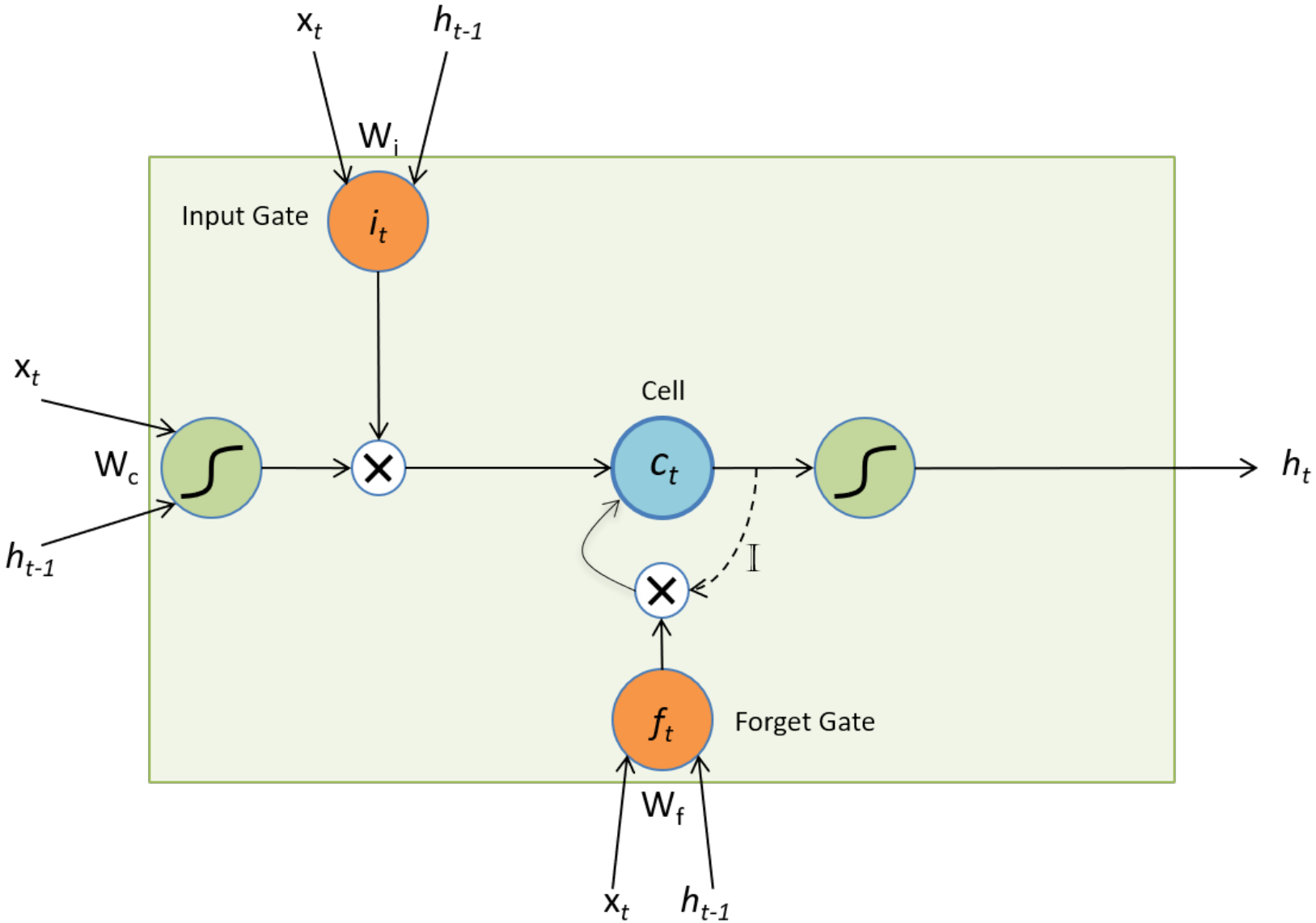
LSTM memory cell and update gates



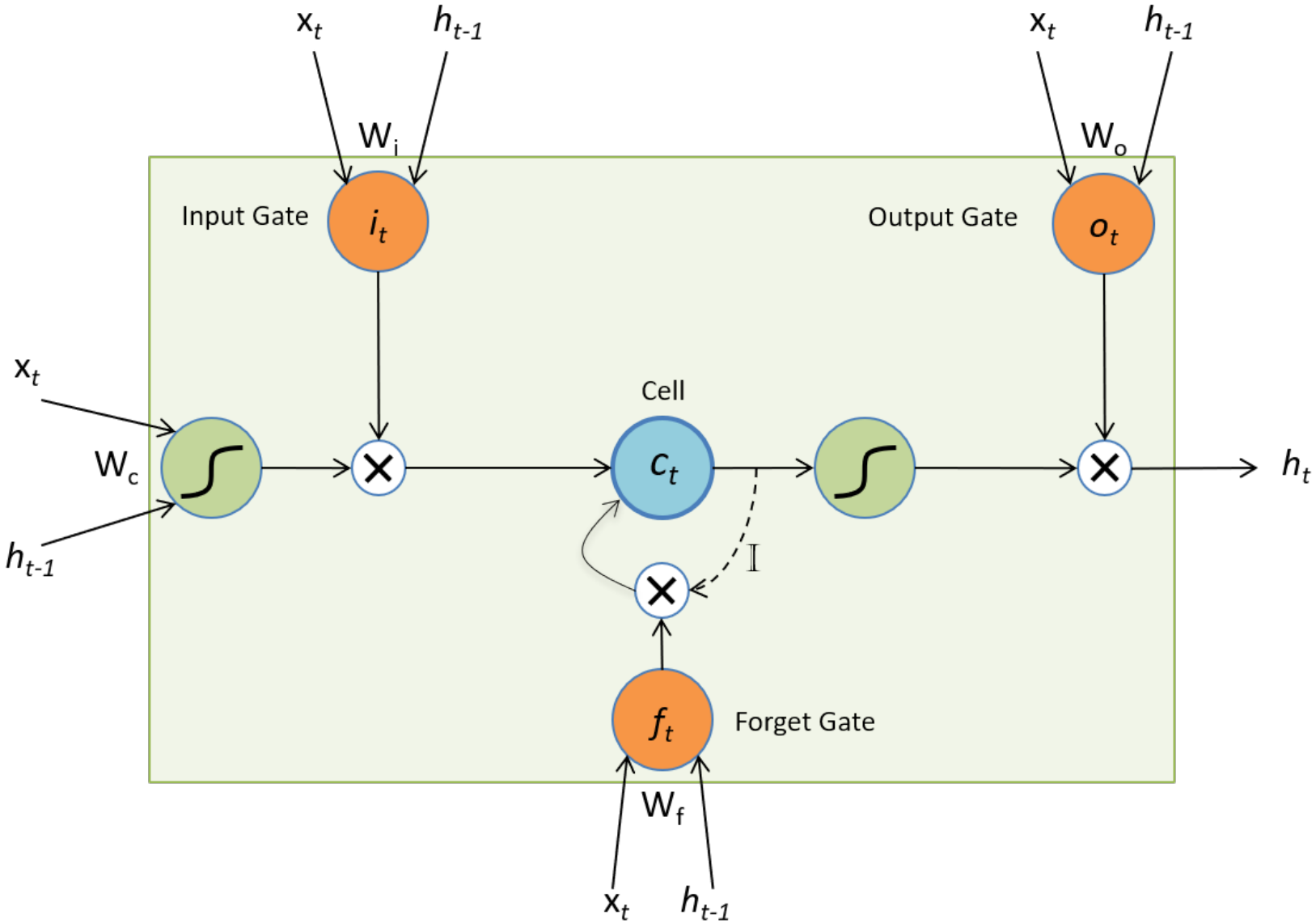
LSTM memory cell and update gates



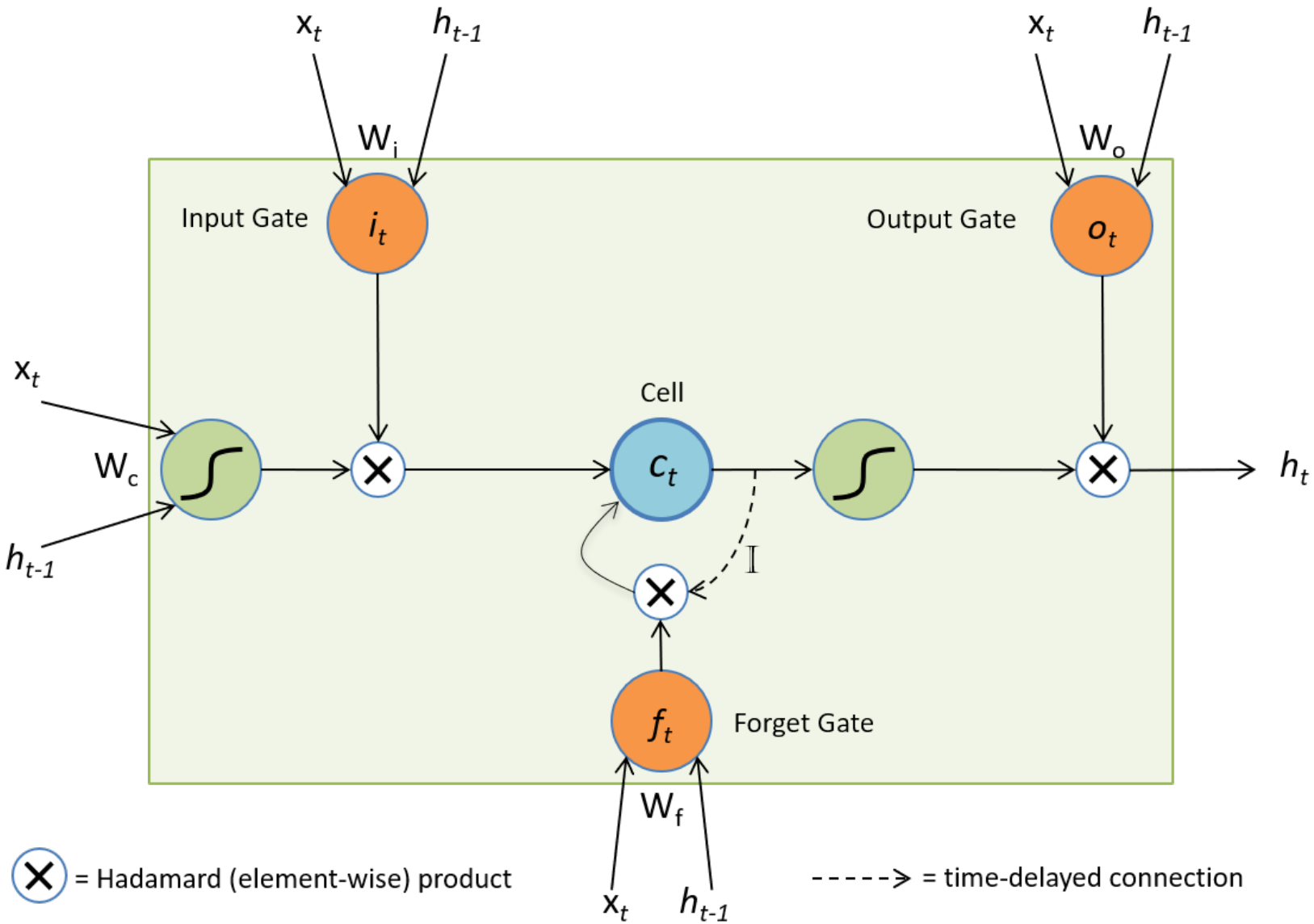
LSTM memory cell and update gates



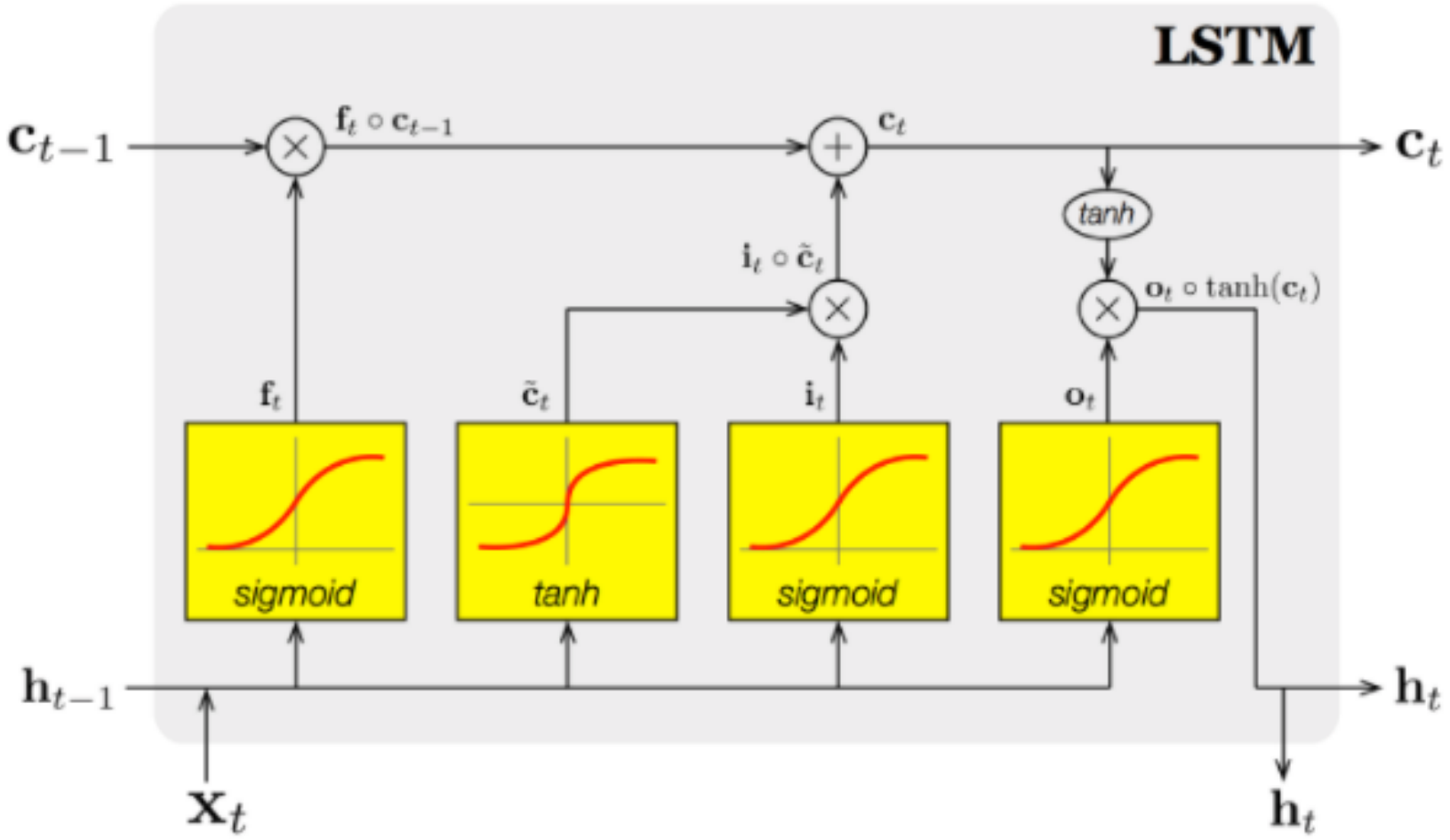
LSTM memory cell and update gates



LSTM memory cell and update gates

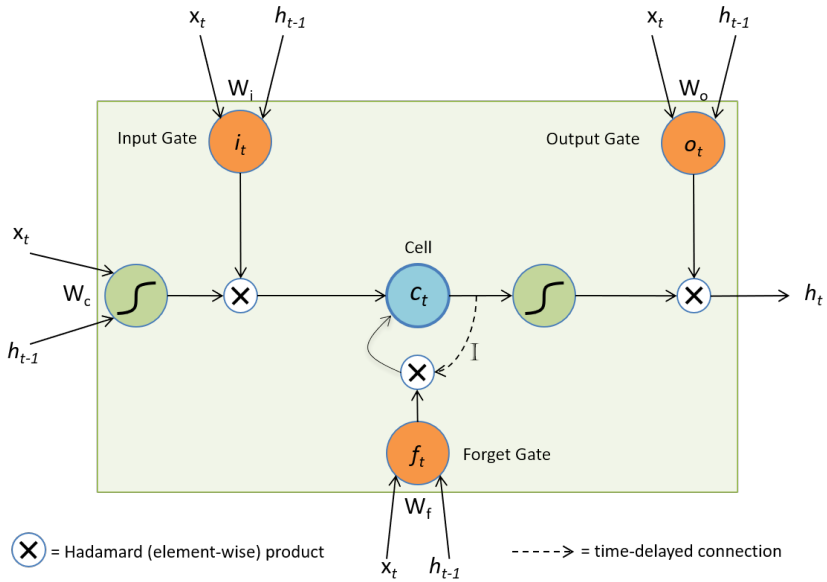


LSTM memory cell and update gates



Another View of LSTM Cell

LSTM memory cell and update gates



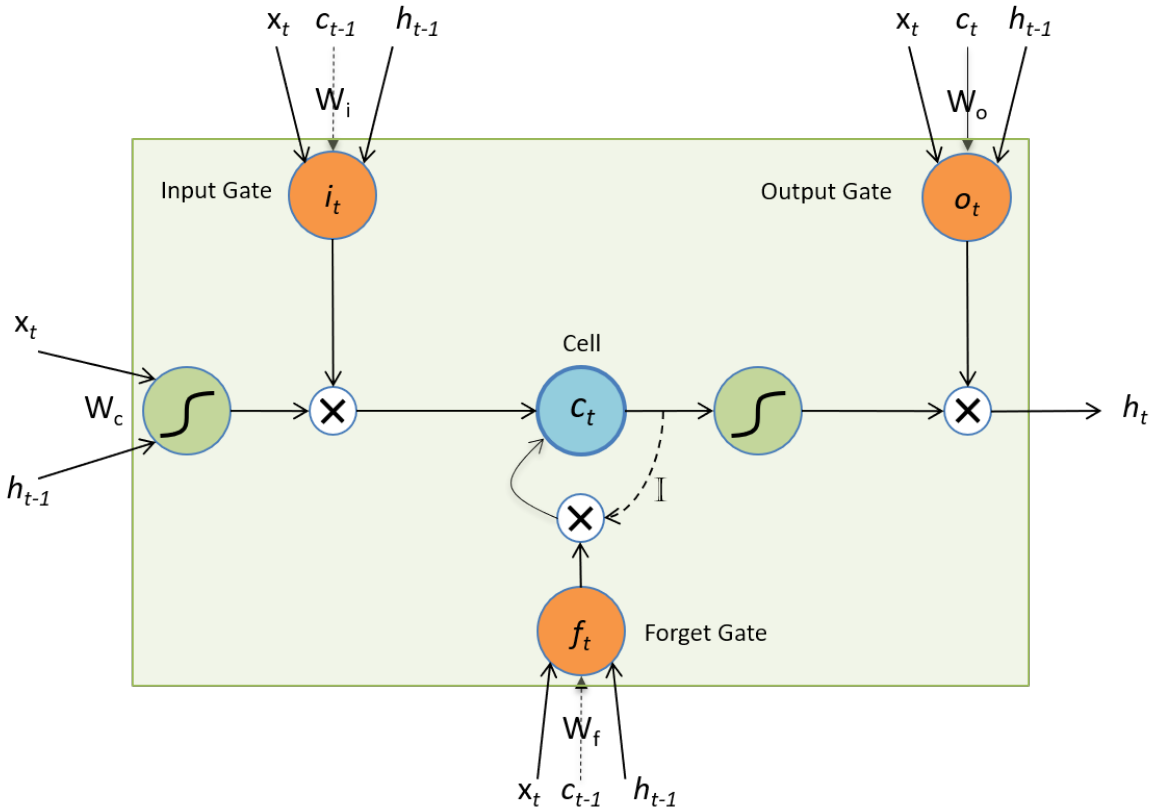
Below: assume x_t already augmented to subsume bias term b.

$$a_t = \tanh(W_c x_t + U_c h_{t-1}) \qquad i_t = \sigma(W_i x_t + U_i h_{t-1})$$
$$f_t = \sigma(W_f x_t + U_f h_{t-1}) \qquad o_t = \sigma(W_o x_t + U_o h_{t-1})$$
$$c_t = i_t \odot a_t + f_t \odot c_{t-1} \qquad h_t = o_t \odot \tanh(c_t)$$

Memory cells c_t are updated with a *gated* combination of a_t and c_{t-1}, with $\odot$ the Hadamard (element-wise) product.

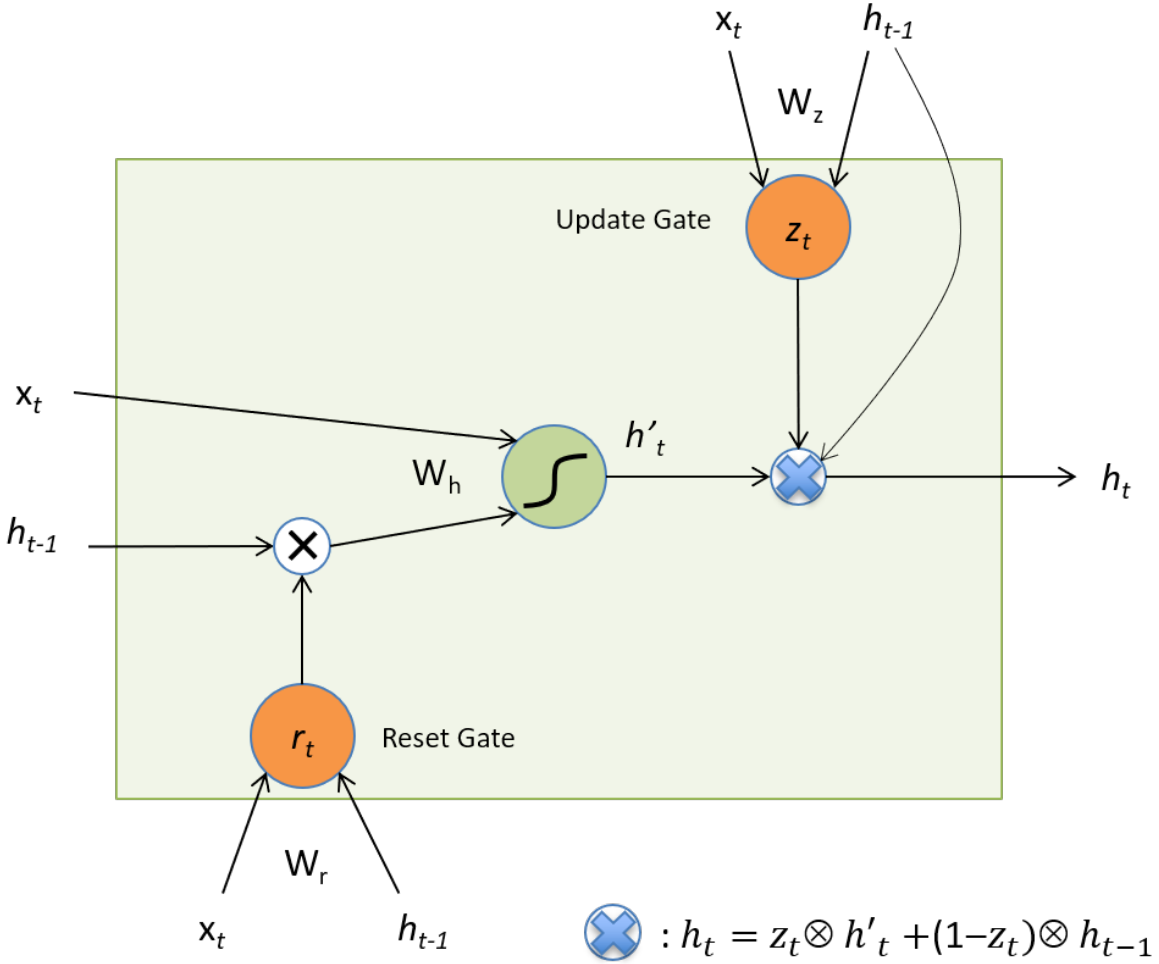
LSTM variant: peephole

If the output gate is closed, other gates (controlling CEC cell) can only see output ≈ 0 of the previous step.



LSTM variant: GRU

- Merge input+forget gates into a single *update* gate.
- Merge cell c_t into hidden state h_t.
- No more output gate.



Reading list

- S. Hochreiter, and J. Schmidhuber, Long short-term memory, Neural computation, 1997 9(8), pp.1735-1780
- F.A. Gers, and J. Schmidhuber, Recurrent nets that time and count, IJCNN 2000
- Felix A. Gers; Jürgen Schmidhuber; Fred Cummins (2000). "Learning to Forget: Continual Prediction with LSTM". Neural Computation.
- R. Pascanu, T. Mikolov, and Y. Bengio, On the difficulty of training recurrent neural networks, ICML 2013
- K. Cho, B. Van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, Learning phrase representations using RNN encoder-decoder for statistical machine translation, ACL 2014
- R. Jozefowicz, W. Zaremba, and I. Sutskever, An empirical exploration of recurrent network architectures, JMLR 2015
- K. Greff , R.K. Srivastava, J. Koutník, B.R. Steunebrink, and J. Schmidhuber, LSTM: A search space odyssey, IEEE transactions on neural networks and learning systems, 2016
- Juergen Schmidhuber's page: `http://people.idsia.ch/~juergen/rnn.html`
- Maths of RNN/LSTM: `arunmallya.github.io/writeups/nn/lstm`, ArXiv:1808.03314
- Blogs by Karpathy, Edwin Chen, SkyMind.
- More: curated lists, AntisymmetricRNN (ICLR'19)

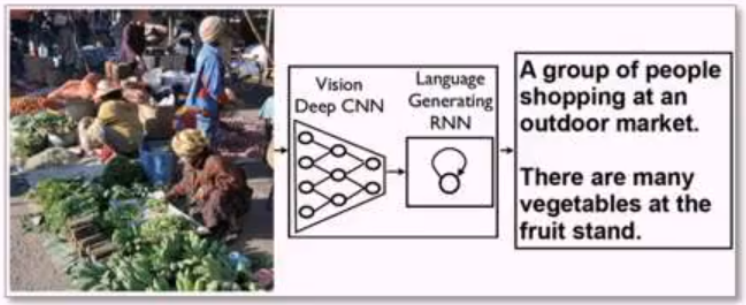
Outline

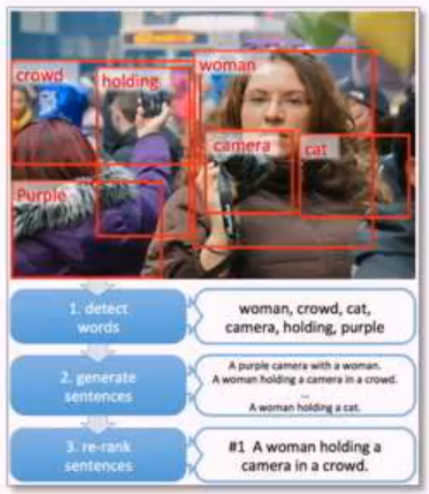
Image Captioning

Two major paradigms for image captioning

Vector-to-Sequence

Adopted **encoder-decoder** framework from machine translation, Popular: Google, Montreal, Stanford, Berkeley





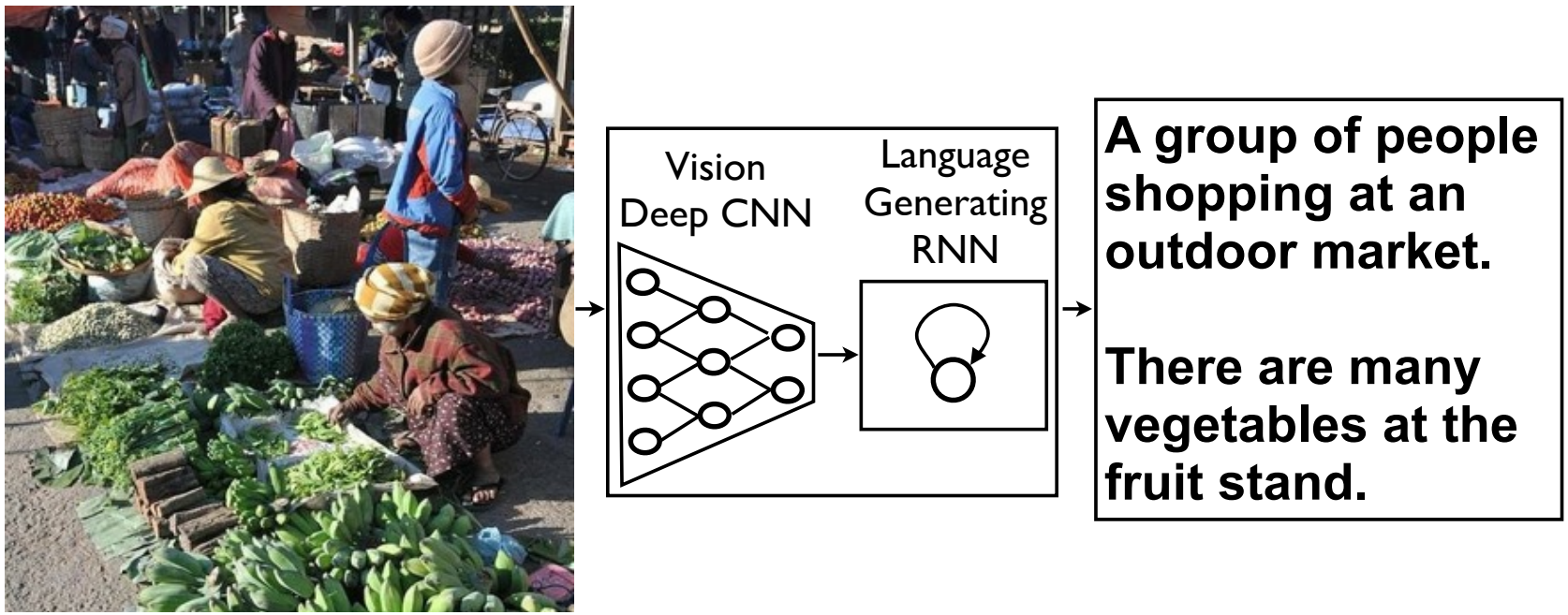
Compositional framework

Visual **concept detection** => caption **candidates generation** => Deep **semantic ranking**

Compositional framework can potentially exploit non paired image-caption data more effectively

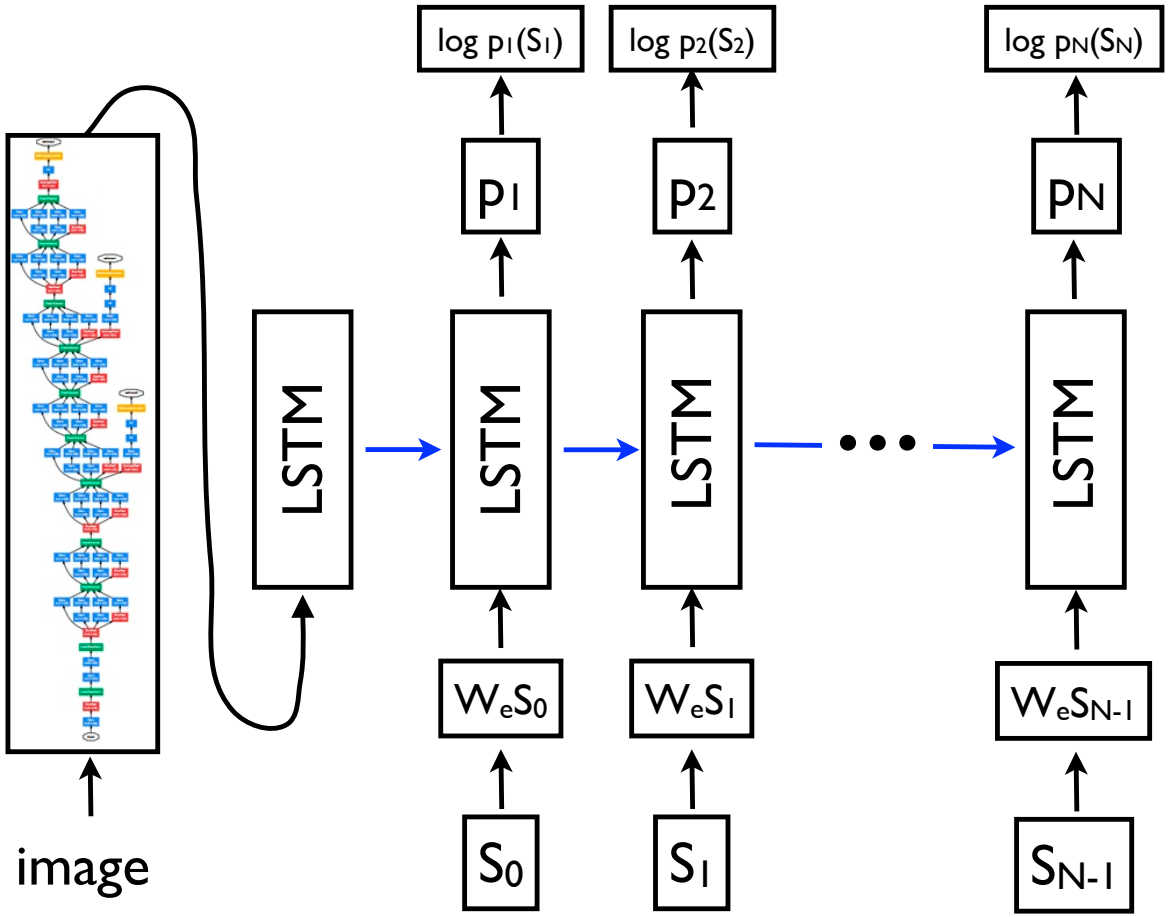
Source: Microsoft Talk'17 (video) by Xiaodong He.

Image Captioning



"empirically verified that feeding the image at each time step as an extra input yields inferior results, as the network can explicitly exploit noise in the image and overfits more easily." Oriol Vinyals et al., ArXiv:1411.4555.

Show and Tell: A Neural Image Caption Generator



CNN as image embedder and W_e for word embeddings. Training loss is the sum of the negative log likelihood of the correct word at each step. Inference: sampling or beam search.

Compare other architectures with minor variations

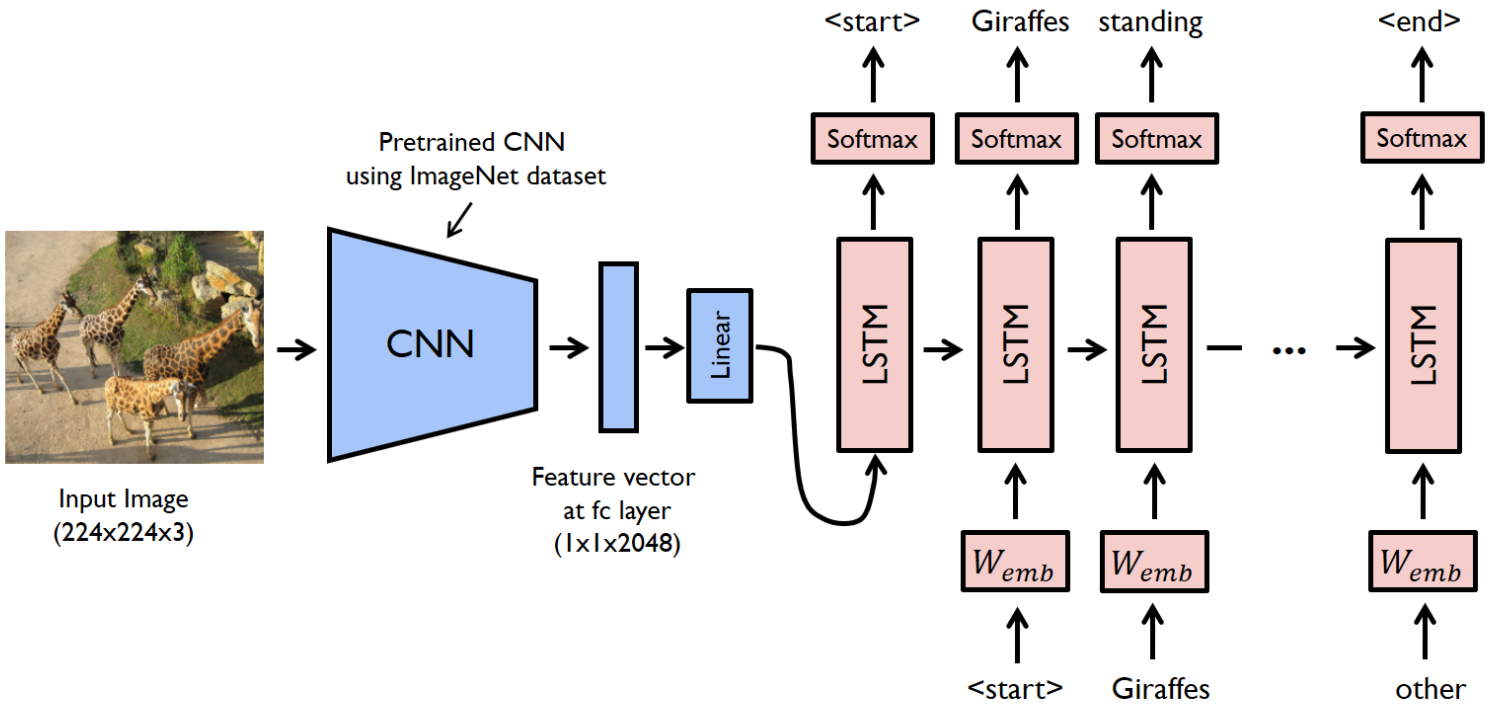


Image embedding is fed to LSTM to predict ⟨START⟩ character instead of skipping one time step?

Show and Tell: A Neural Image Caption Generator

A man throwing a frisbee in a park.
A man holding a frisbee in his hand.
A man standing in the grass with a frisbee.
A close up of a sandwich on a plate.
A close up of a plate of food with french fries.
A white plate topped with a cut in half sandwich.
A display case filled with lots of donuts.
A display case filled with lots of cakes.
A bakery display case filled with lots of donuts.

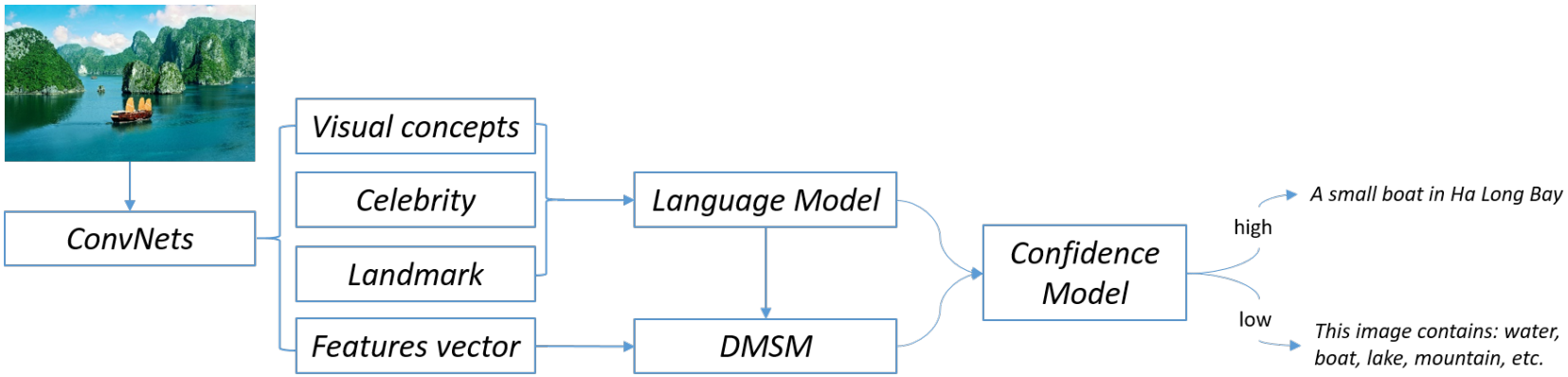
Table: N-best examples from the MSCOCO test set. Bold lines indicate a novel sentence not present in the training set.

Show and Tell: A Neural Image Caption Generator

Source: Oriol Vinyals et al., ArXiv:1411.4555

Outline

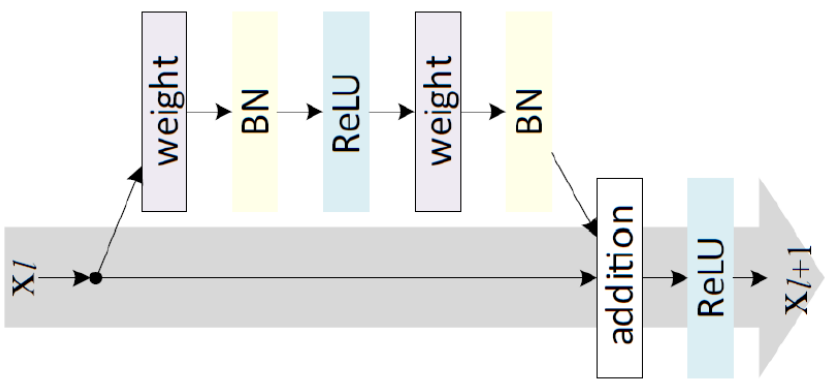
Rich Image Captioning in the Wild



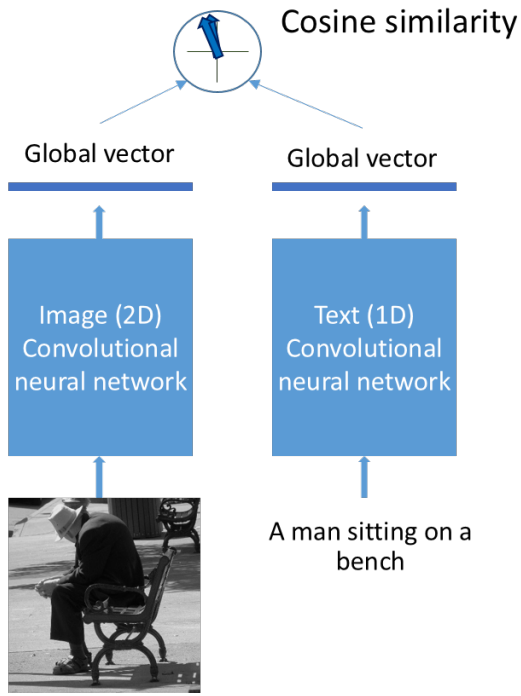
Source: Kenneth Tran et al., ArXiv:1603.09016. See also `captionbot.ai`

The system is decomposed into independent components, trained separately, then integrated in the main pipeline:

1. A deep residual network-based vision model that detects a broad range of visual concepts



Rich Image Captioning in the Wild



2. Instead of using GRNN: maximum entropy language model (MELM) for candidates generation (with beam search) and a deep multimodal semantic model (DMSM) for caption ranking.
3. DeepCNNs trained to identify thousands of celebrities and landmarks
4. A classifier trained with confidence score for each output caption.