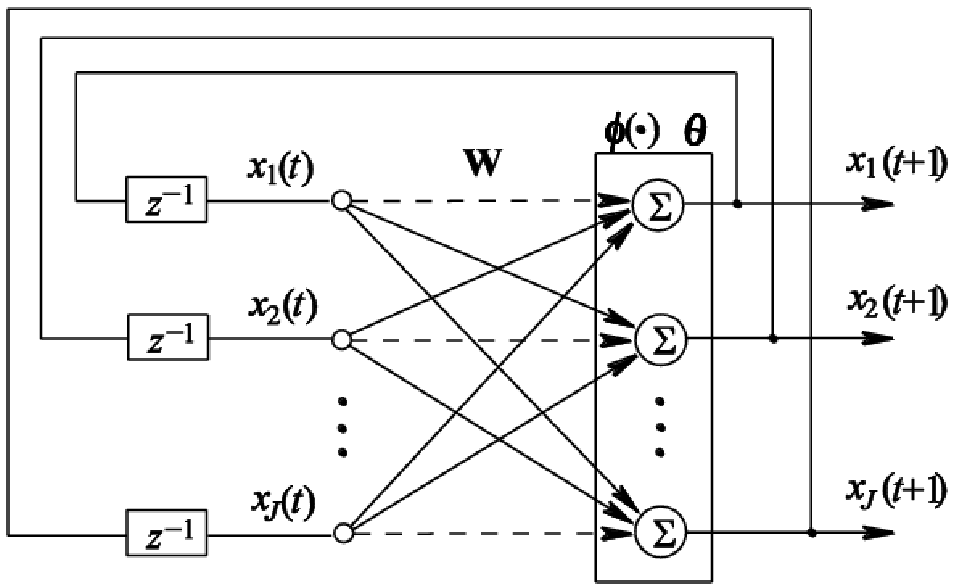
EEE511
Recurrent Neural Networks

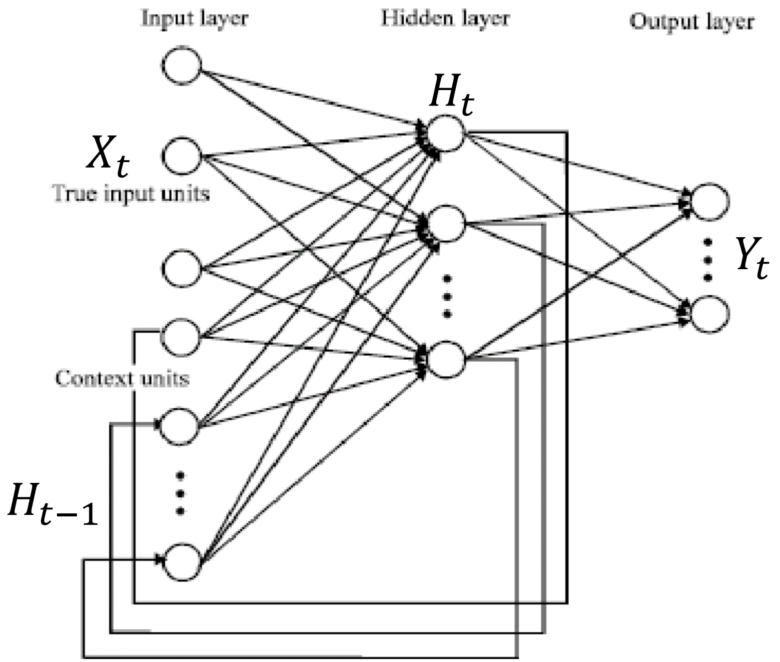
Equations help provide great insight!



$$X_{t+1} = \varphi(WX_t + \theta)$$

Hopfield network

Hopfield, John J. "Neurons with graded response have collective computational properties like those of two-state neurons." *Proceedings of the national academy of sciences* 81.10 (1984): 3088-3092.



$$Y_t = W_{hy}^T H_t$$

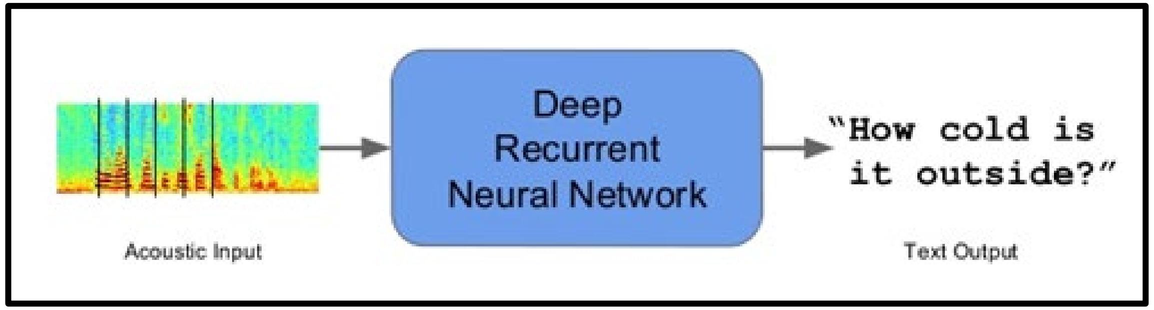
$$H_t = \tanh(W_{hh}^T H_{t-1} + W_{xh}^T X_t)$$

Elman network

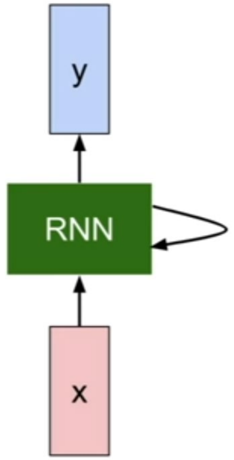
J.L. Elman, Generalization, simple recurrent networks, and the emergence of structure, in: Proceedings of the 20th Annual Conference of the Cognitive Science Society, Mahway, NJ, 1998.

Sequence modeling and applications

- RNNs are well suited for sequence modeling (order matters…), such as natural language processing, music generation, image captioning, stock market predictions, DNA sequencing, and more
- Recurrence connections provide memory/store information
- (Truncated) backpropagation through time well suited to train RNN
- LSTM or GRU use gates to control information flow: forget not useful info and store useful info



Recurrent Neural Networks (RNN)



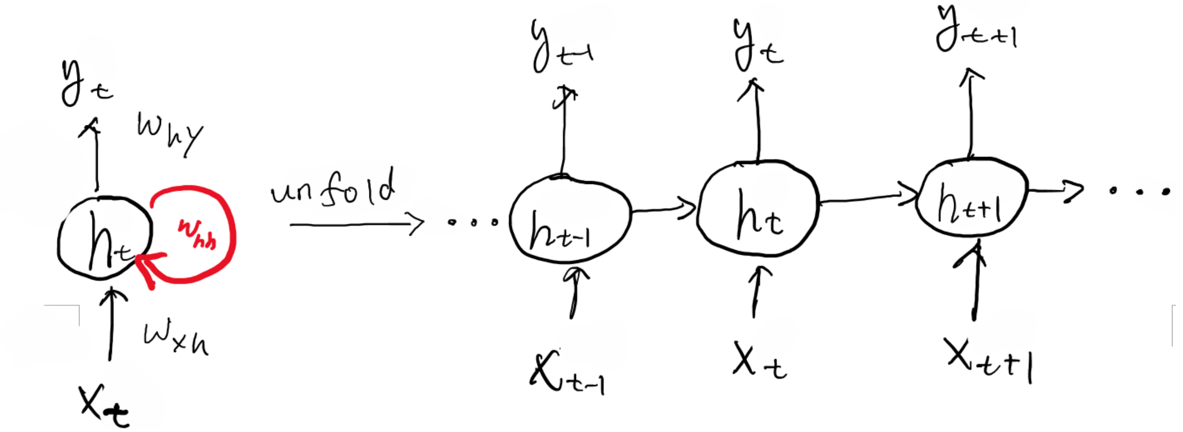
$$\boxed{h_t} = \boxed{f_W}(\boxed{h_{t-1}}, \boxed{x_t})$$

new state

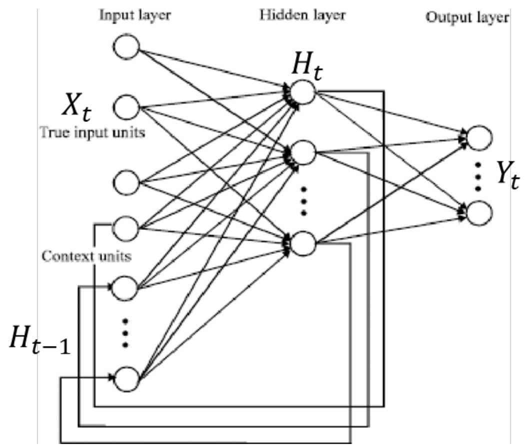
some function
with parameters W

old state

input vector at
some time step



Elman network



Output vector:

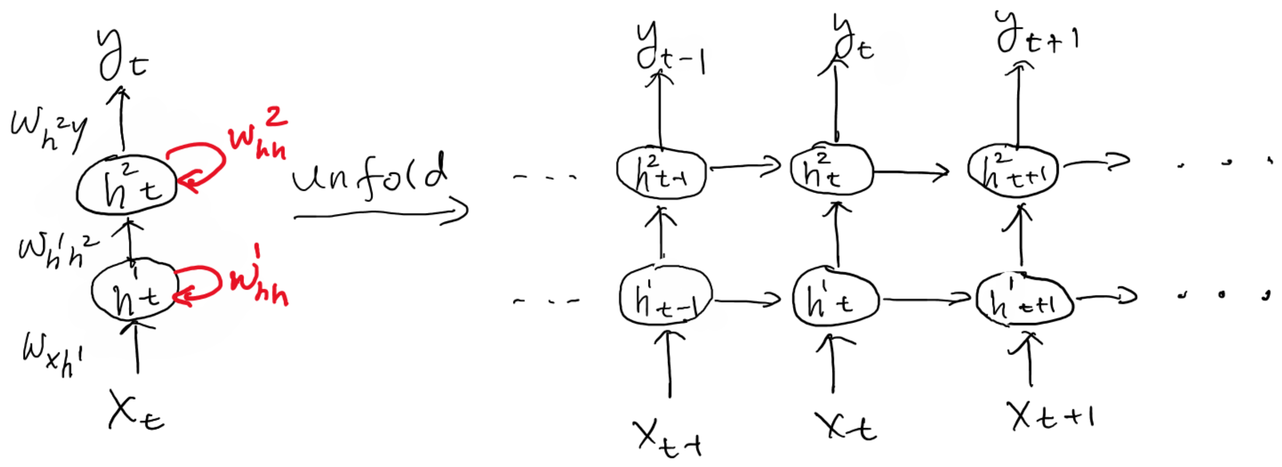
$$y_t = \boldsymbol{W}_{hy}^{\boldsymbol{T}} h_t$$

Update hidden state:

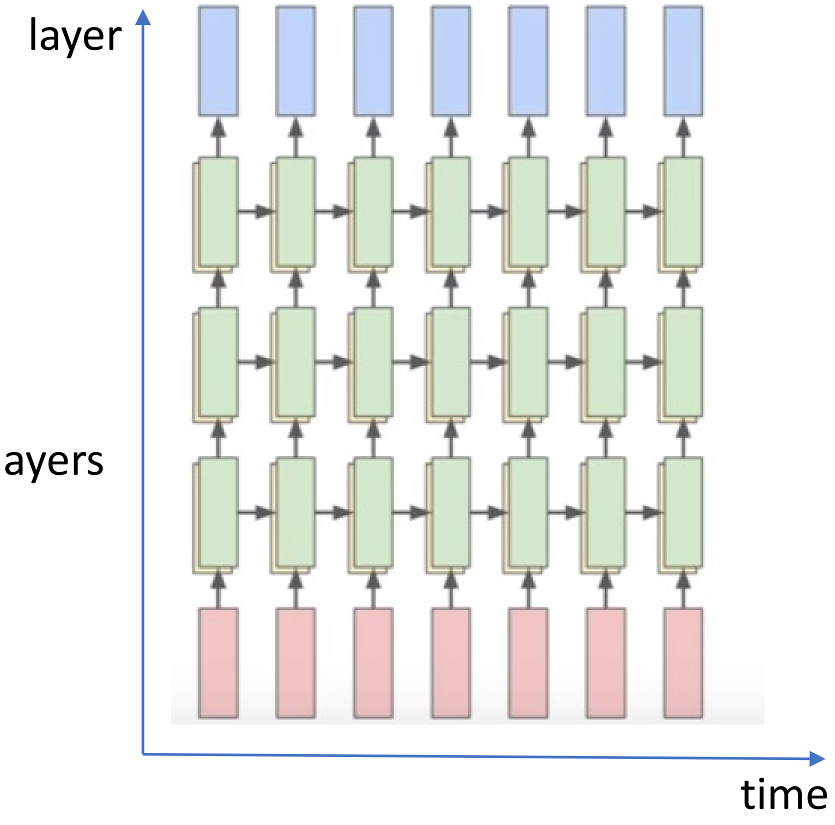
$$h_t = \tanh\left(\boldsymbol{W}_{hh}^{\boldsymbol{T}} h_{t-1} + \boldsymbol{W}_{xh}^{\boldsymbol{T}} x_t\right)$$

Input vector: x_t

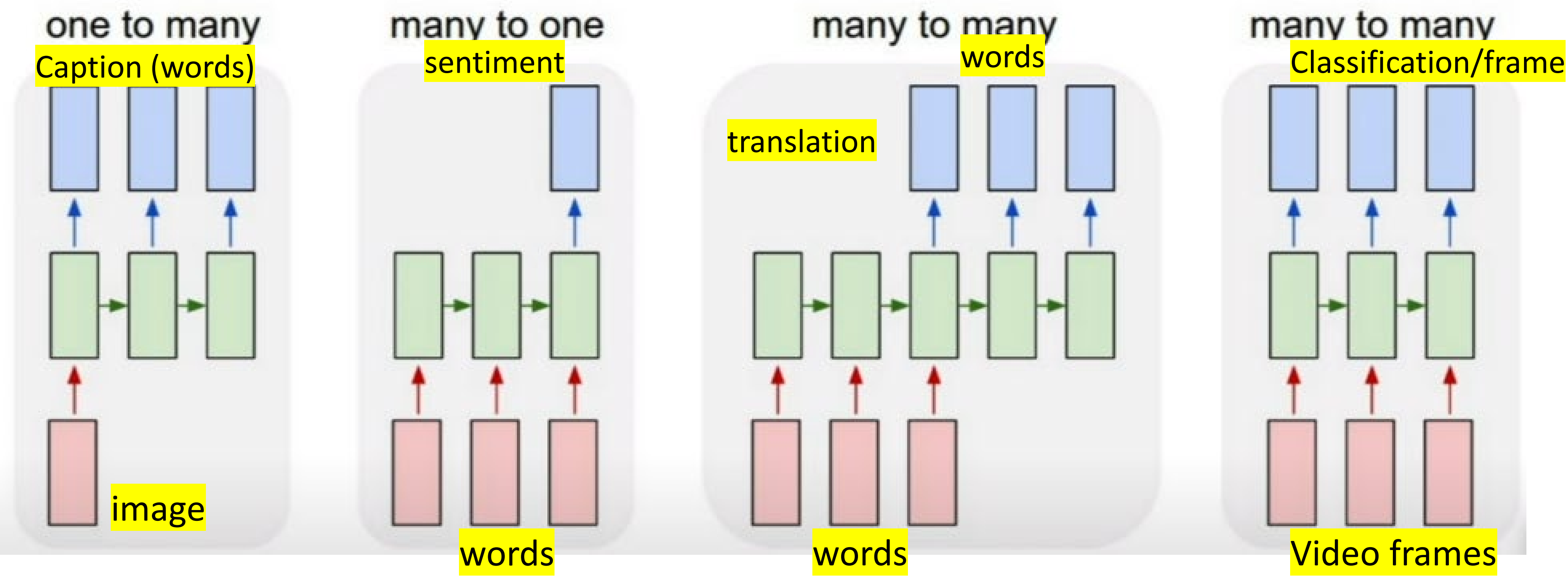
2 hidden layer RNN



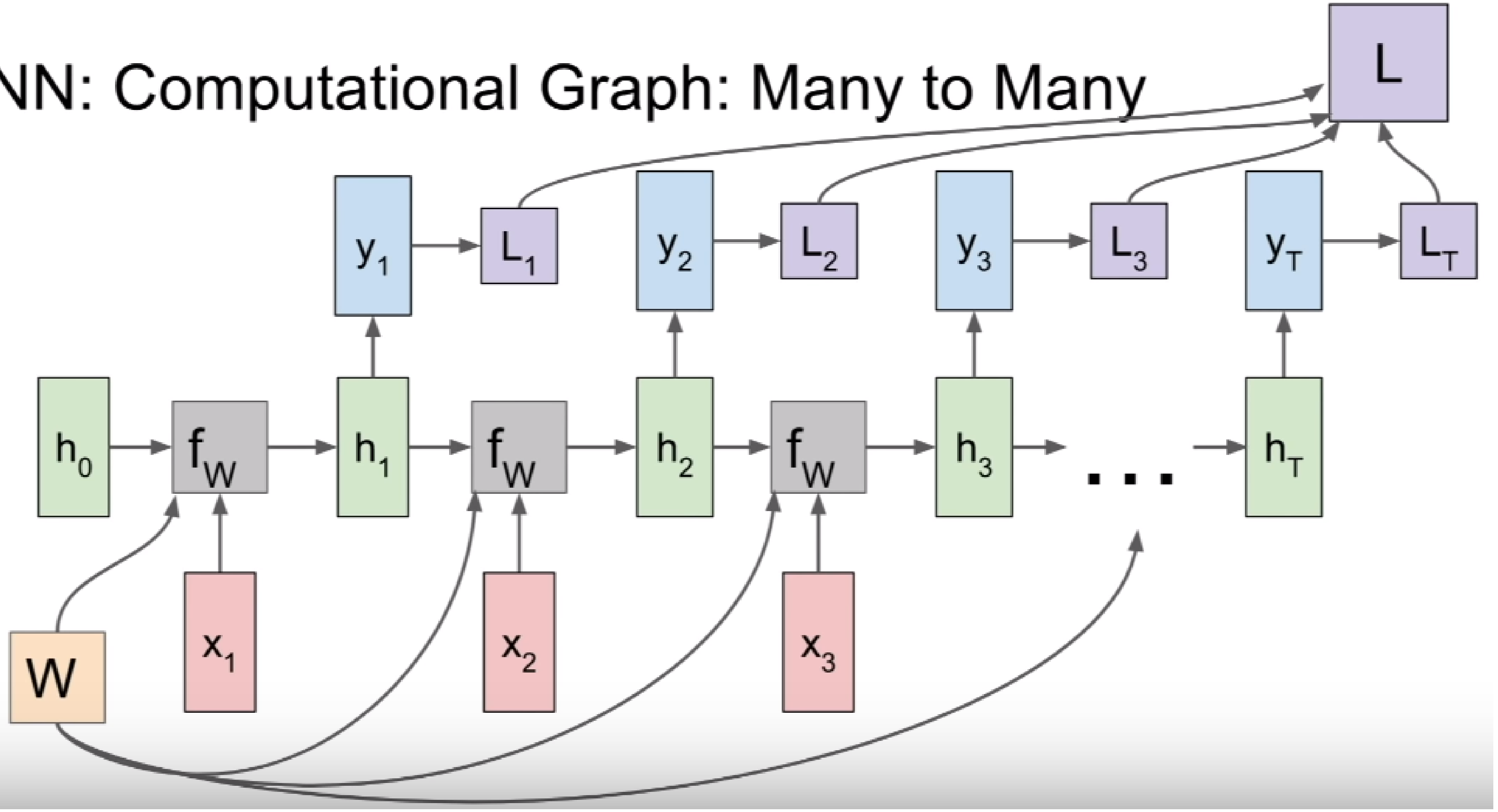
Usually 2-3 hidden layers



RNN deals with sequential information (examples as shown below)



RNN: Computational Graph: Many to Many

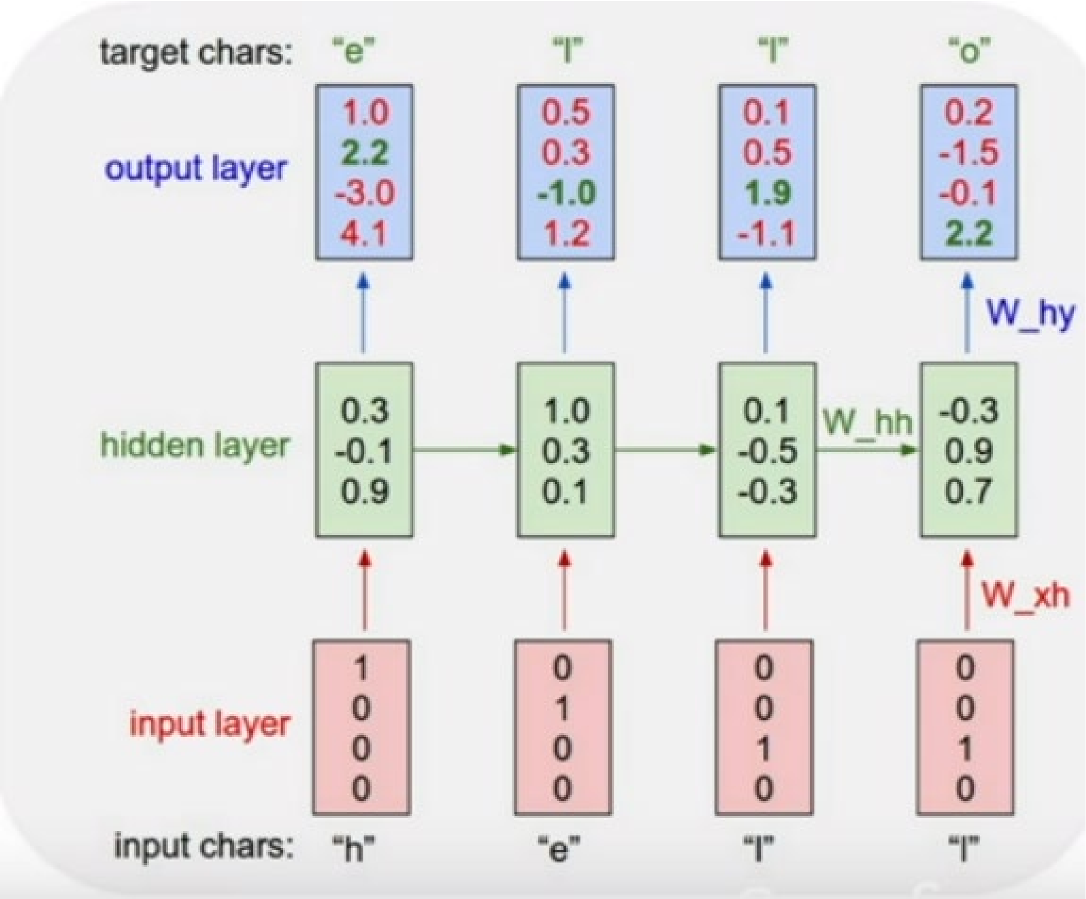


Example: Language modeling (determining the probability of a sequence of words)

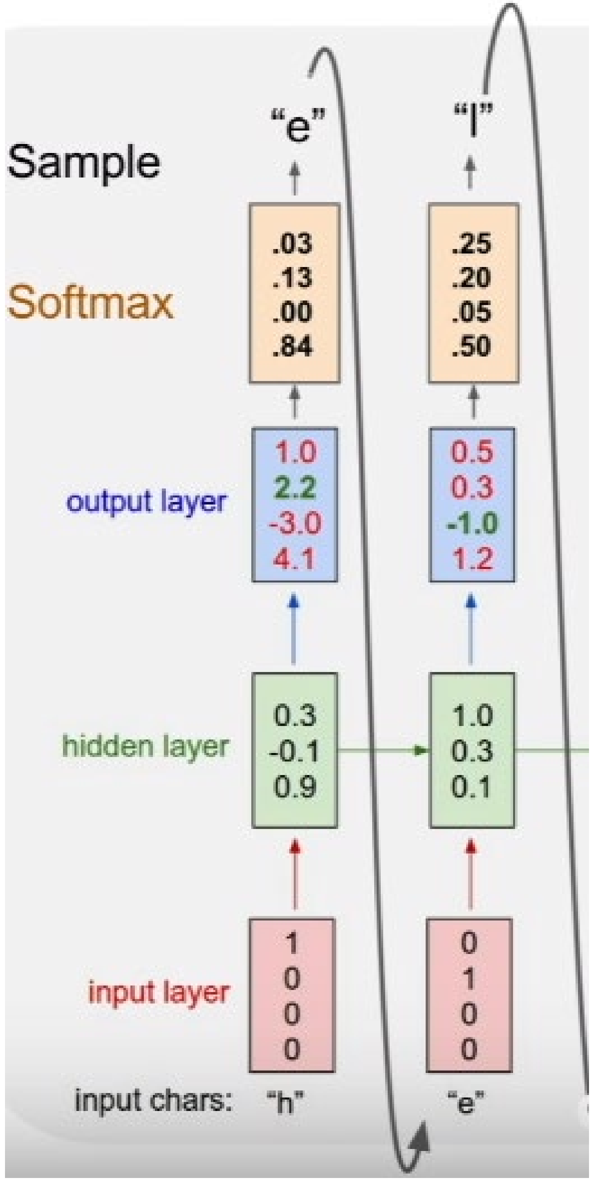
Character-level language model

Vocab = [h e l o]

Training sample sequence: hello



Testing: sample characters one at a time
Input: one character
Output: prediction of the next character
…



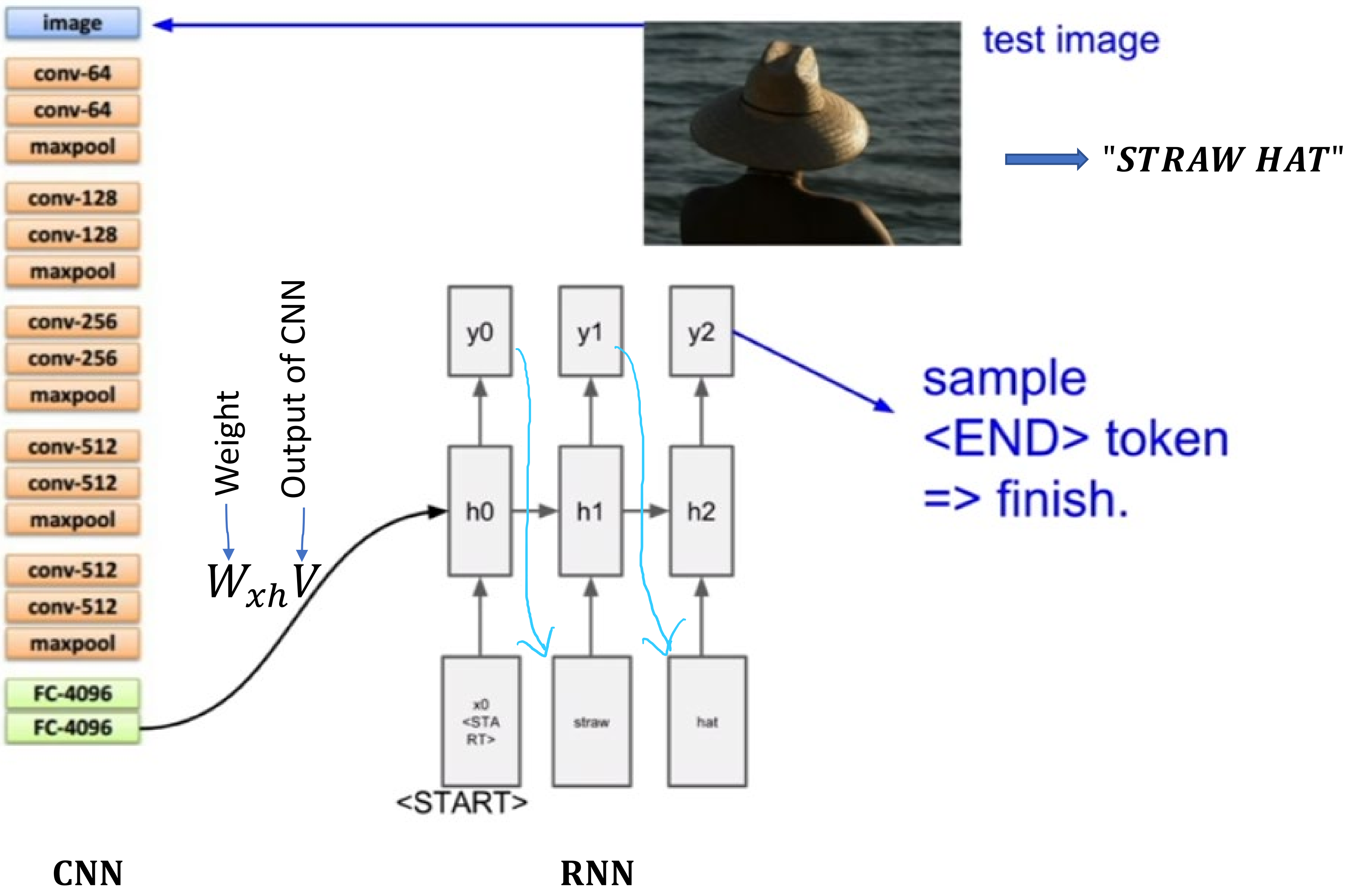


image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
test image
"STRAW HAT"
Weight
Output of CNN
$W_{xh}V$
y0
y1
y2
h0
h1
h2
x0 <START>
straw
hat
<START>
sample <END> token => finish.
CNN
RNN

Not quite successful…

A bird is perched on a tree branch

A man in a baseball uniform throwing a ball

A woman standing on a beach holding a surfboard

Successful captioning

A cat sitting on a suitcase on the floor

A cat is sitting on a tree branch

A dog is running in the grass with a frisbee

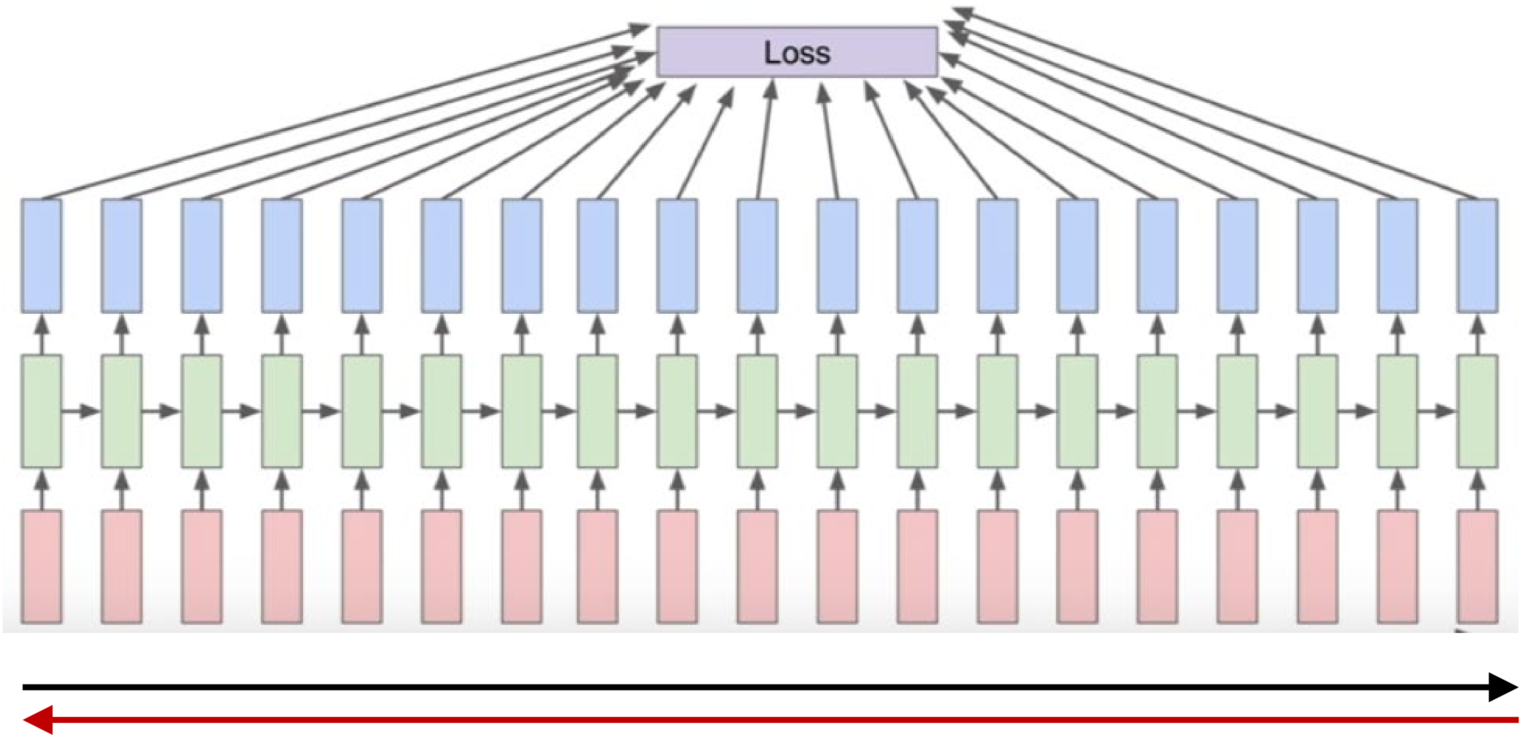
A white teddy bear sitting in the grass

- Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9, no. 8 (1997): 1735-1780.
- Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, *5*(2), 157-166.
- Hochreiter, S., Bengio, Y., Frasconi, P., & Schmidhuber, J. (2001). Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.

Backpropagation through time (BTT):

Forward through entire sequence to computer loss

Backward through entire sequence to compute gradient

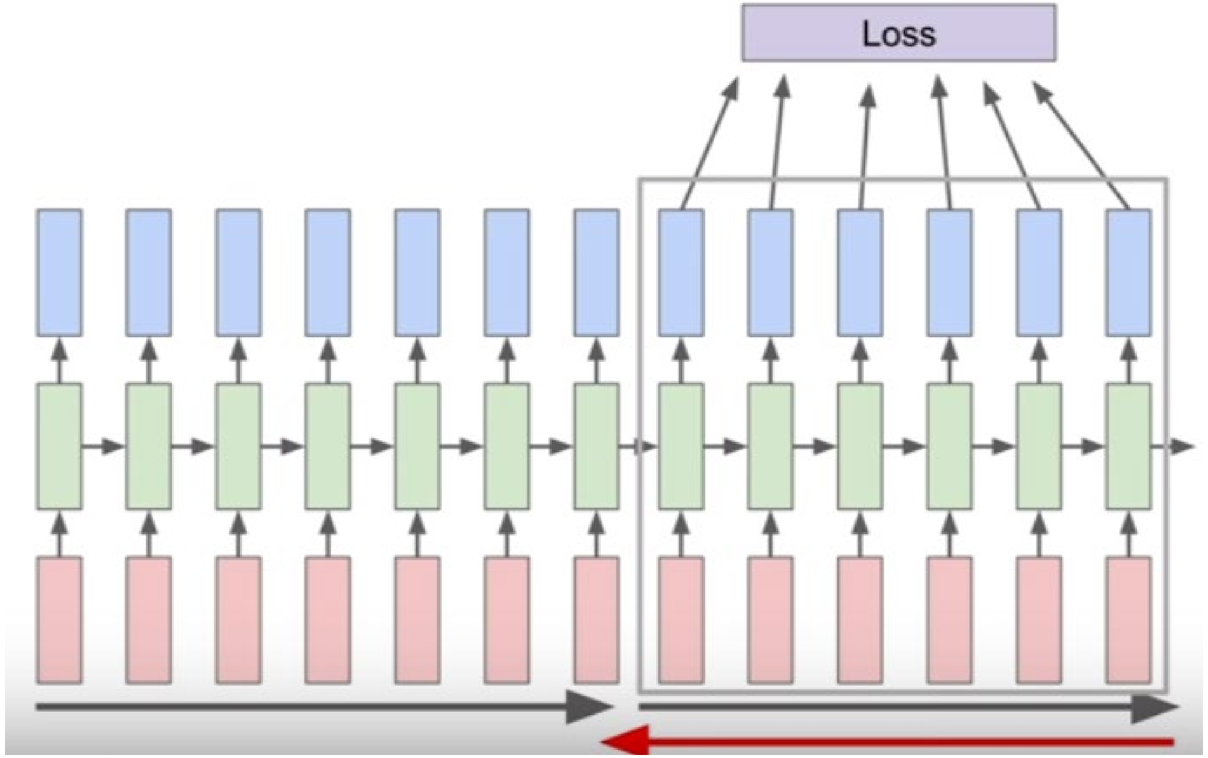


Werbos, Paul J. "Backpropagation through time: what it does and how to do it." Proceedings of the IEEE 78, no. 10 (1990): 1550-1560.

Truncated backpropagation through time (Truncated BTT):

Carry hidden states forward in time

Only backpropage for a chunk (~100) of the hidden states or smaller number of steps



Problems:

Exploding gradient (accumulation of large derivatives, unstable model, not effective learning – gradient clipping)

Vanishing gradient (accumulation of small gradients, model cannot learn, not updated effectively)

To see that… notice the equation below”

$$h_3 = (f \ldots (f(f(h_0, x_0), x_1), x_2 \ldots)$$

for example, $0.3^{10} \to 10^{-7}$, $1.7^{10} \to 201$

Solutions:

Ideas: introduce gated cells to control information flow

Long Short-Term Memory networks (LSTMs): track info flow through time

Gated Recurrent Units (GRUs)

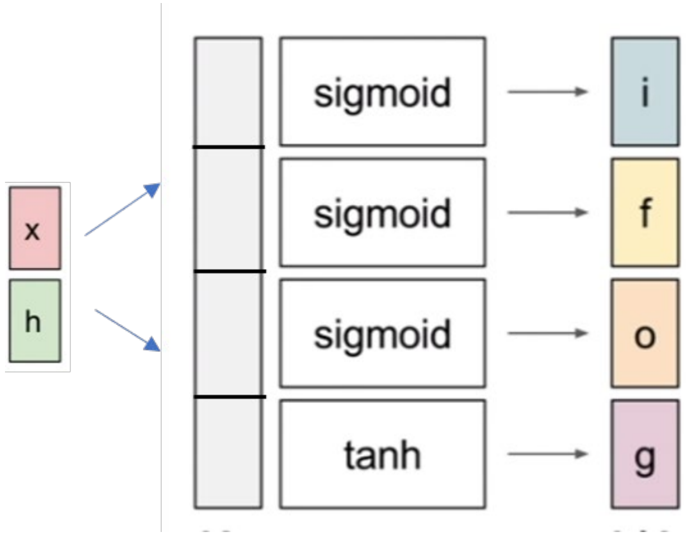
Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

Introduce a new hidden (memory) cell c_t
f gate (forget gate): erase or not?
i gate (input gate): write to cell or not?
g gate: info into cell
o gate (output gate): amount of info revealed

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

$$c_t = f \odot c_{t-1} + i \odot g$$

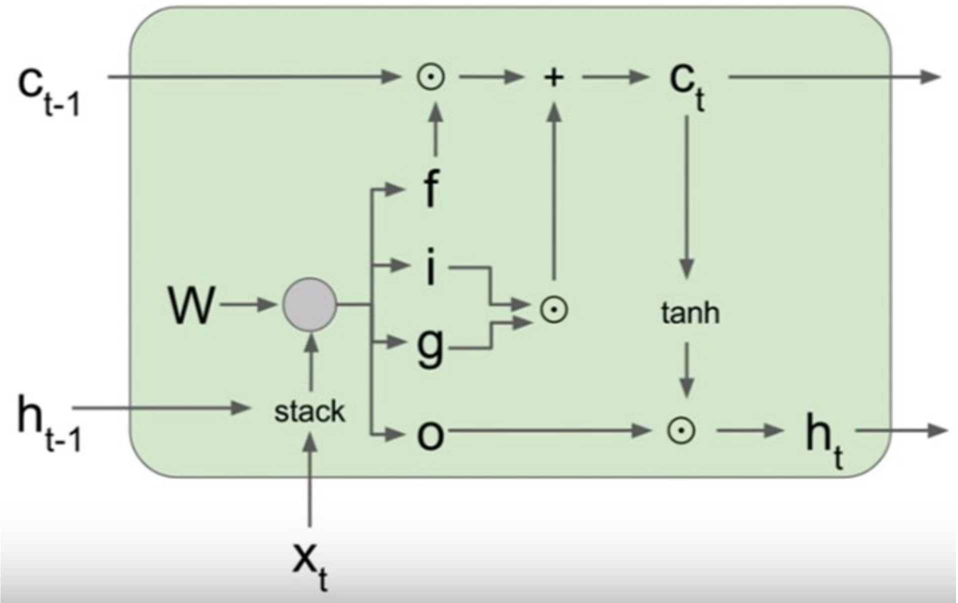
$$h_t = o \odot \tanh(c_t)$$



In 2015, Google reported cutting transcription errors in their speech recognition service by up to 49%, a huge increase after years of incremental progress.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

Controlled forward information flow
Uninterrupted gradient flow



$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

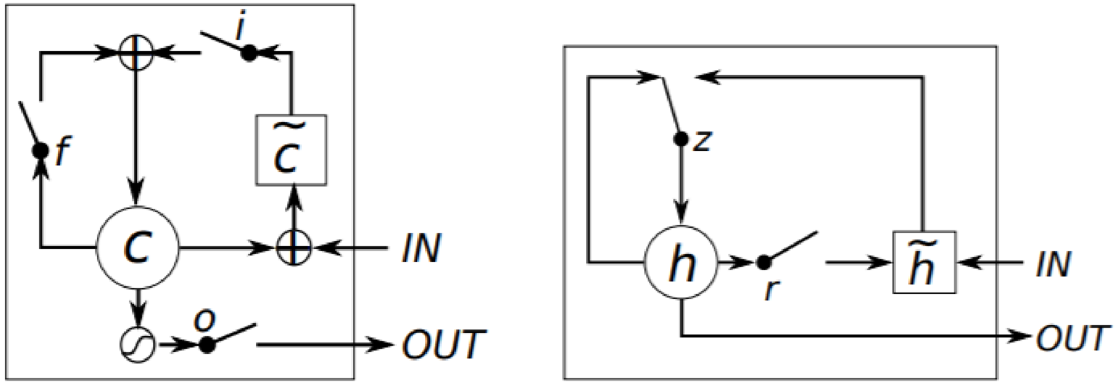
$$c_t = f \odot c_{t-1} + i \odot g$$

$$h_t = o \odot \tanh(c_t)$$

Element-wise multiplication (not matrix multiplications) during error backpropagation
Recurrence on cell state, f gate ~(0,1) may have vanishing gradient (initialize to ~1)

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.

GRU: Gated Recurrent Unit



(a) Long Short-Term Memory

(b) Gated Recurrent Unit

$$r_t = \sigma(W_{xr}x_t + W_{hr}h_{t-1} + b_r)$$
$$z_t = \sigma(W_{xz}x_t + W_{hz}h_{t-1} + b_z)$$
$$\tilde{h}_t = \tanh(W_{xh}x_t + W_{hh}(r_t \odot h_{t-1}) + b_h)$$
$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t$$

Sequence modeling and applications

- RNNs are well suited for sequence modeling applications such as natural language processing, music generation, image captioning, etc…
- Recurrence connections provide memory/store information
- (Truncated) backpropagation through time well suited to train RNN
- LSTM or GRU use gates to control information flow: forget not useful info and store useful info