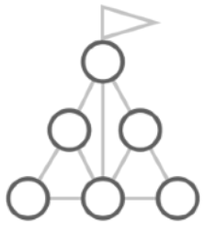
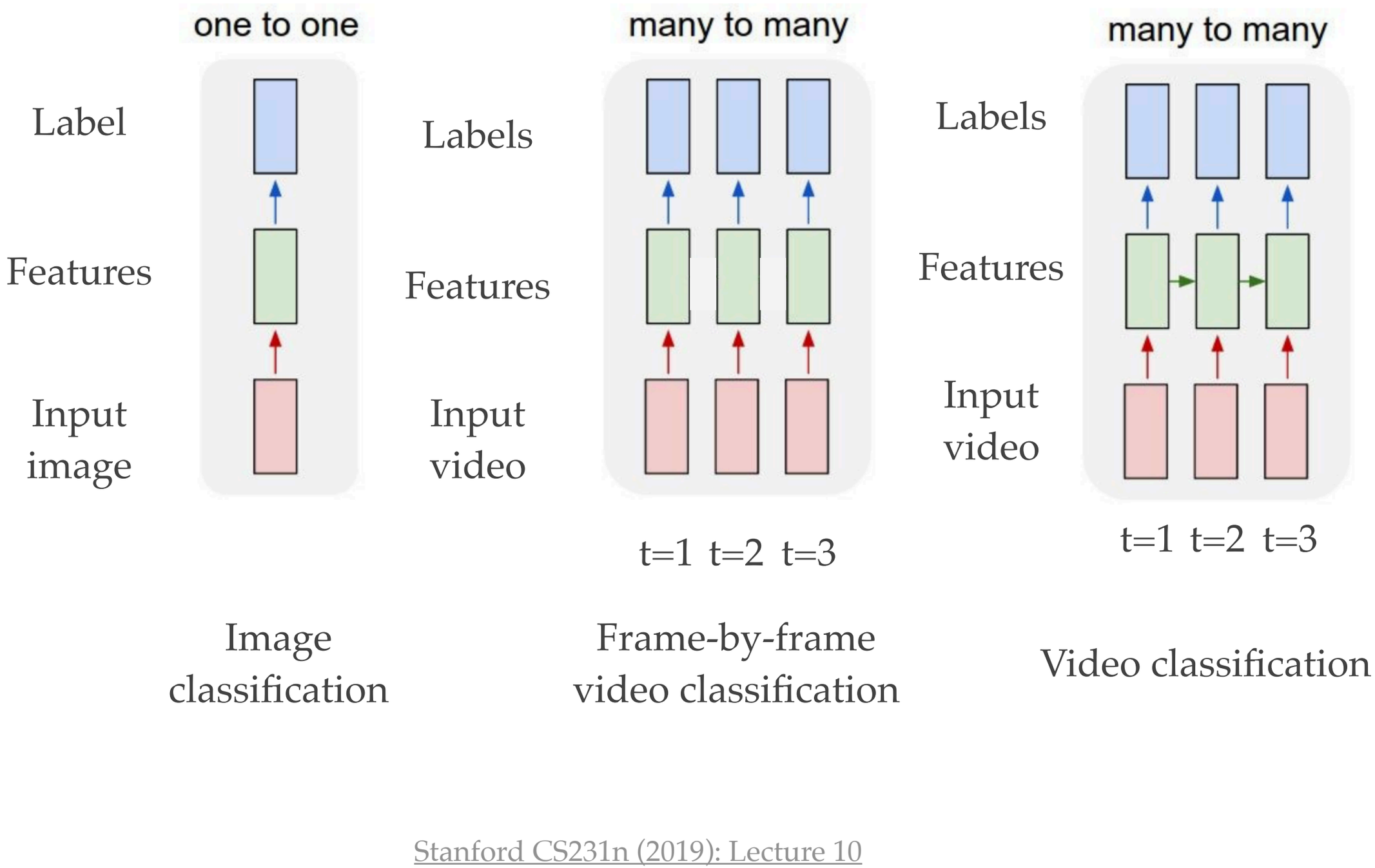


Recurrent Neural Network

Itthi Chatnuntawech

Recurrent Neural Network (RNN)



Stanford CS231n (2019): Lecture 10

Recurrent Neural Network (RNN)

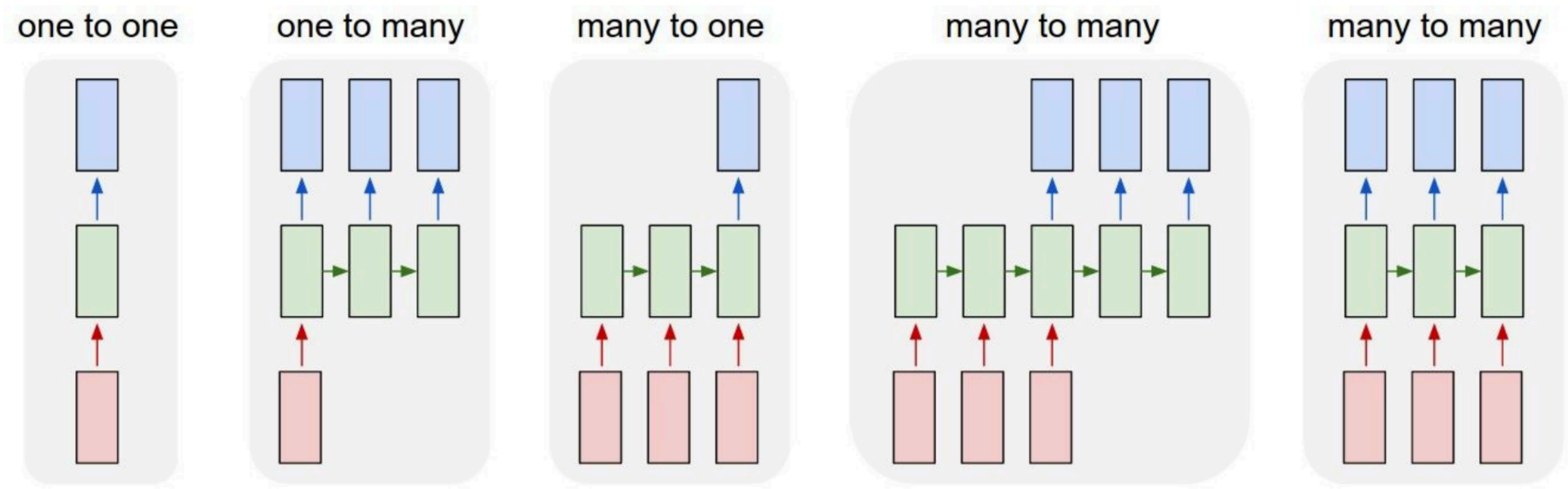


Image classification

Image captioning

Video classification

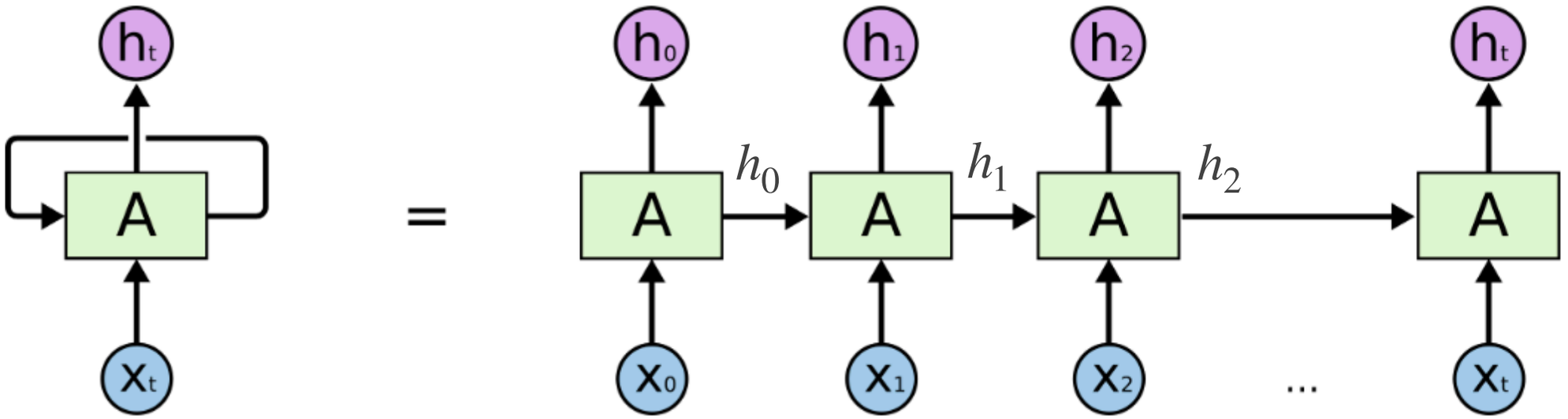
Machine translation

Frame-by-frame video classification

Stanford CS231n (2019): Lecture 10

From here on out, the goal is to understand the big picture through the pictorial explanation. Do not worry too much about the equations.

Recurrent Neural Network (RNN)



$$h_t = A(h_{t-1}, x_t)$$

New state | Old state | Current input

Summary of $x_0, x_1, \ldots, x_t$

Same weights over time

$$h_t = tanh(W_h h_{t-1} + W_x x_t)$$

$t = 0$ $\quad h_0 = tanh(W_x x_0)$

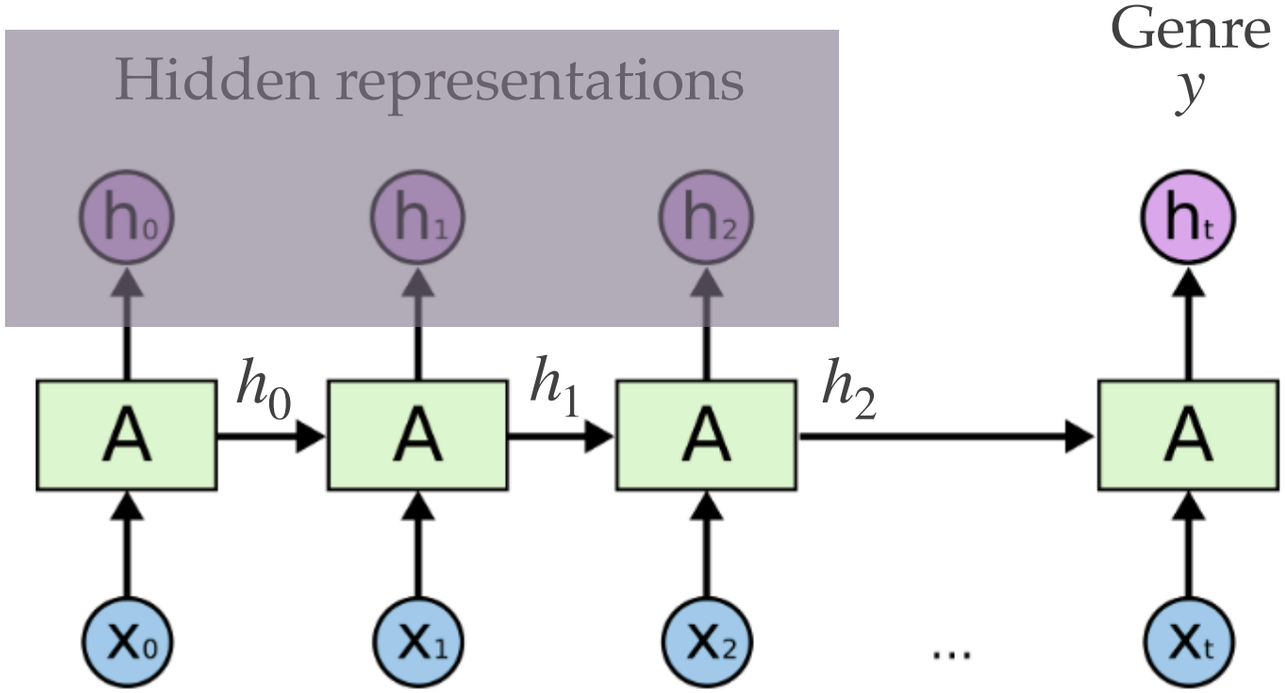
$t = 1$ $\quad h_1 = tanh(W_h h_0 + W_x x_1)$

$t = 2$ $\quad h_2 = tanh(W_h h_1 + W_x x_2)$

Understanding LSTM Networks by Christopher Colah

Recurrent Neural Network (RNN)

Task: Video classification (e.g., genre prediction)



$t = 0$ $h_0 = A(x_0)$

$t = 1$ $h_1 = A(h_0, x_1)$

$t = 2$ $h_2 = A(h_1, x_2)$

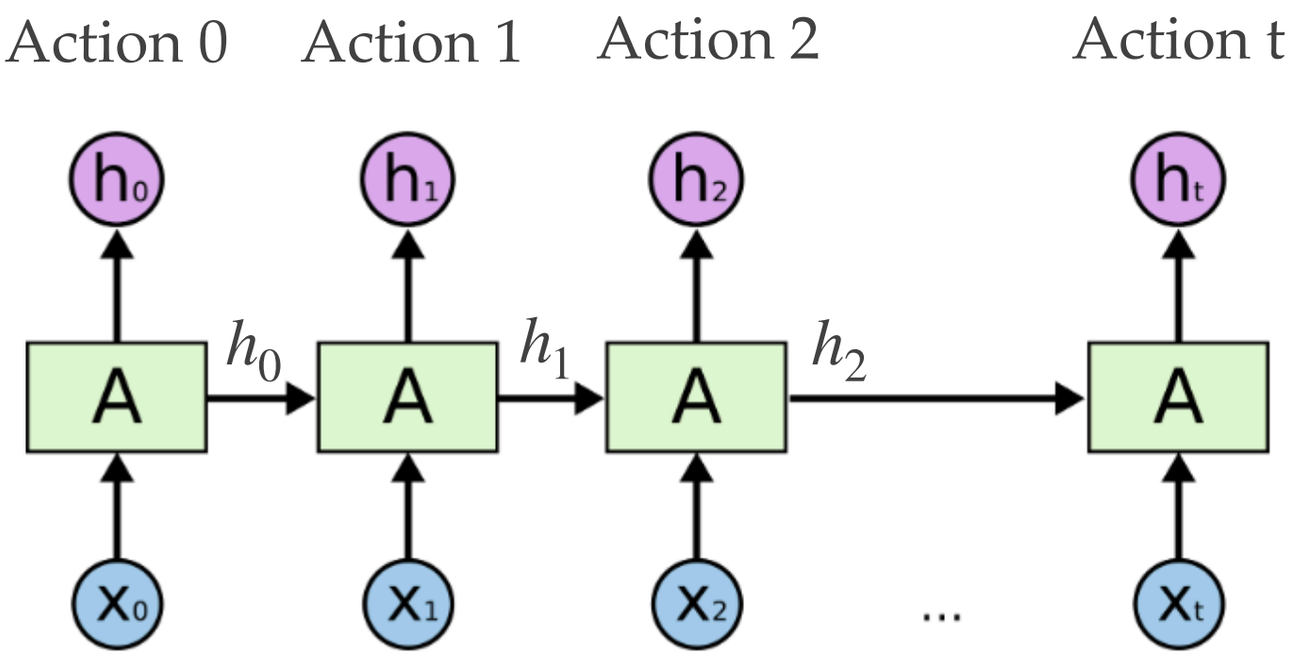
For example,

$y = B(h_t)$

Understanding LSTM Networks by Christopher Colah

Recurrent Neural Network (RNN)

Task: Frame-by-frame video classification (e.g., action prediction)



frame 0 frame 1 frame 2 Last frame

$t = 0$ $h_0 = A(x_0)$ $y_0 = B(h_0)$

$t = 1$ $h_1 = A(h_0, x_1)$ $y_1 = B(h_1)$

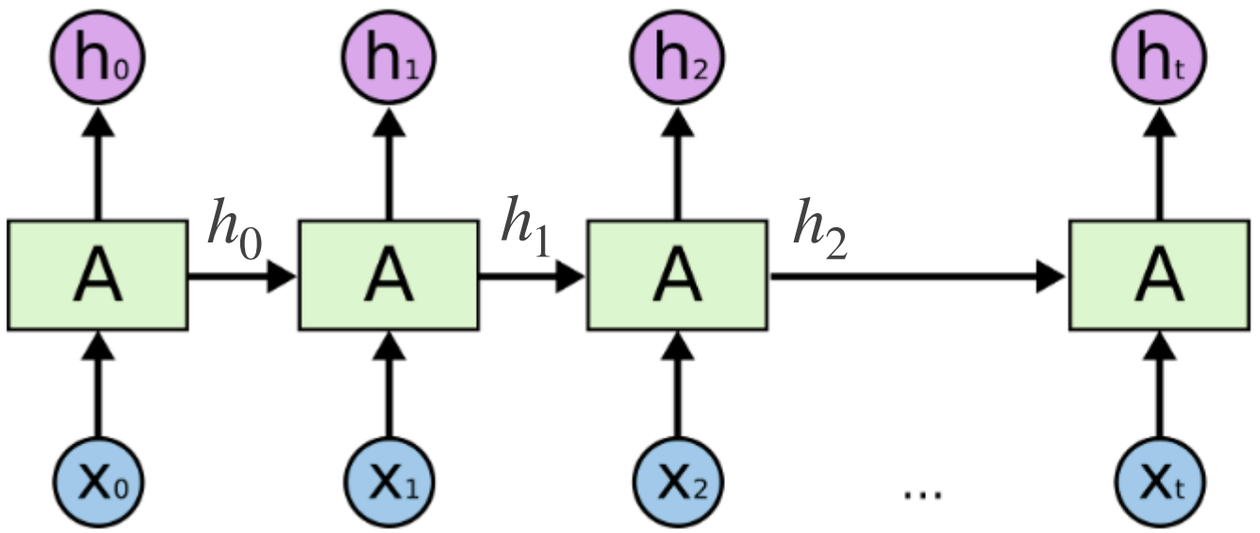
$t = 2$ $h_2 = A(h_1, x_2)$ $\vdots$

$y_t = B(h_t)$

Understanding LSTM Networks by Christopher Colah

Recurrent Neural Network (RNN)

Task: Image captioning (i.e., generate a sentence describing an input image)

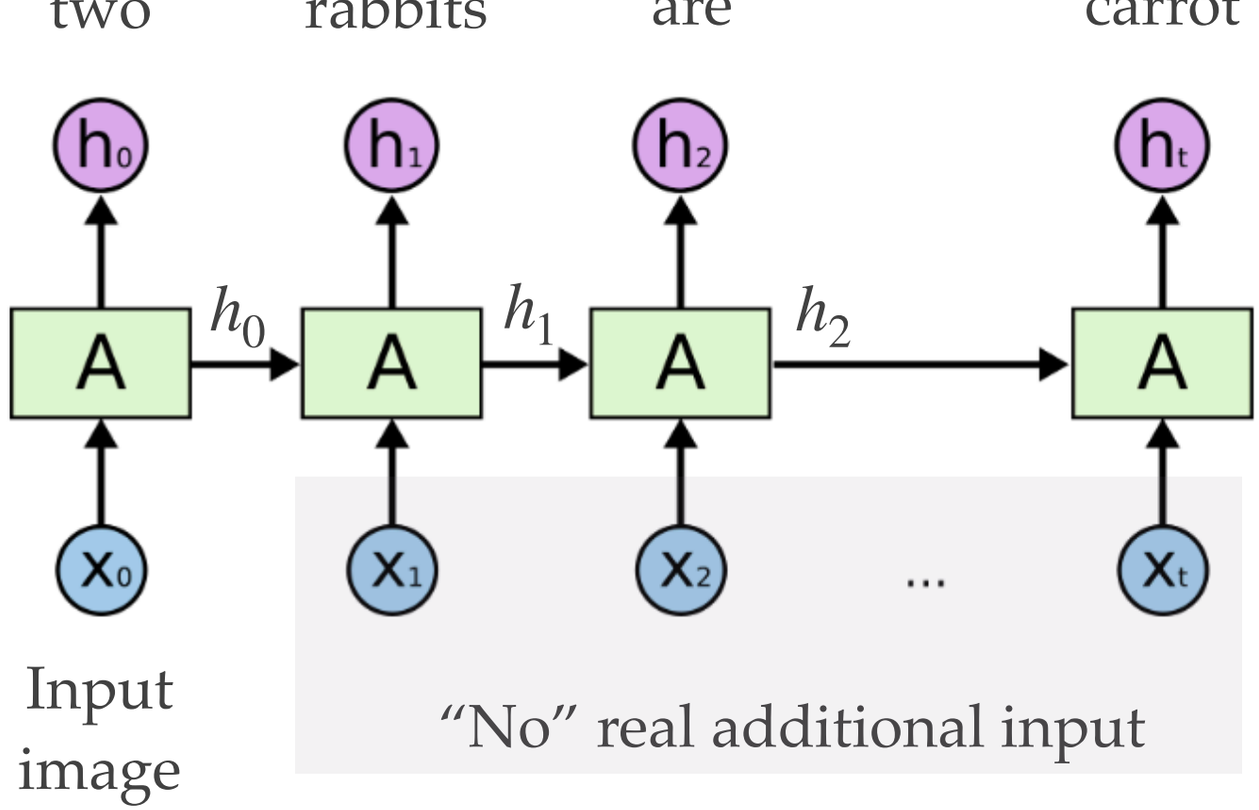


Two rabbits are eating a carrot

Understanding LSTM Networks by Christopher Colah

Recurrent Neural Network (RNN)

Task: Image captioning (i.e., generate a sentence describing an input image)

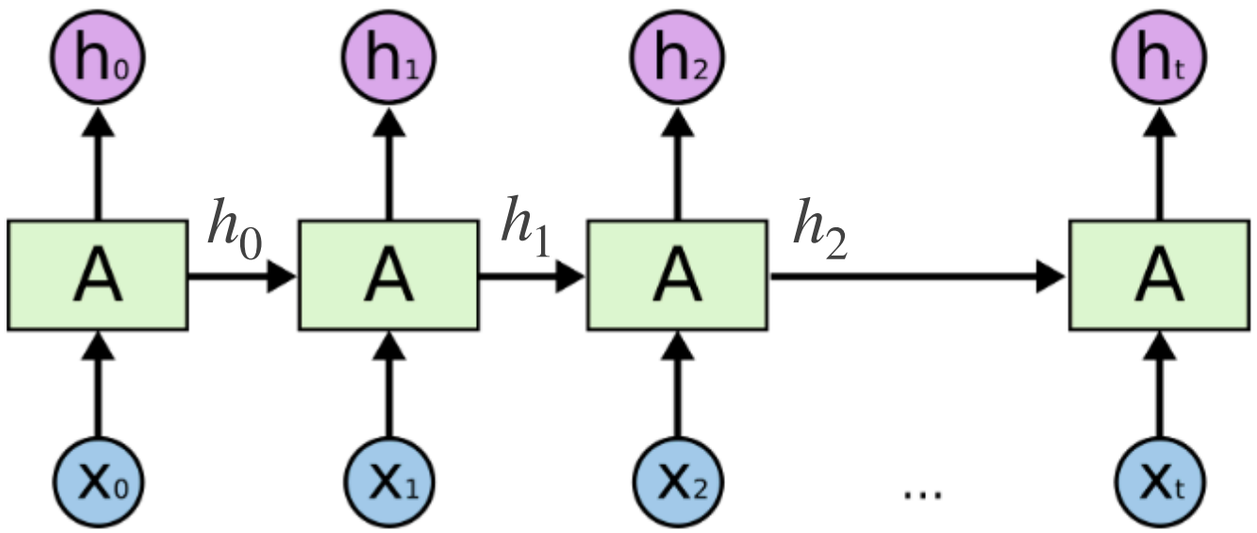


Two rabbits are eating a carrot

Understanding LSTM Networks by Christopher Colah

Recurrent Neural Network (RNN)

Task: Machine translation (e.g., translate language A to language B)

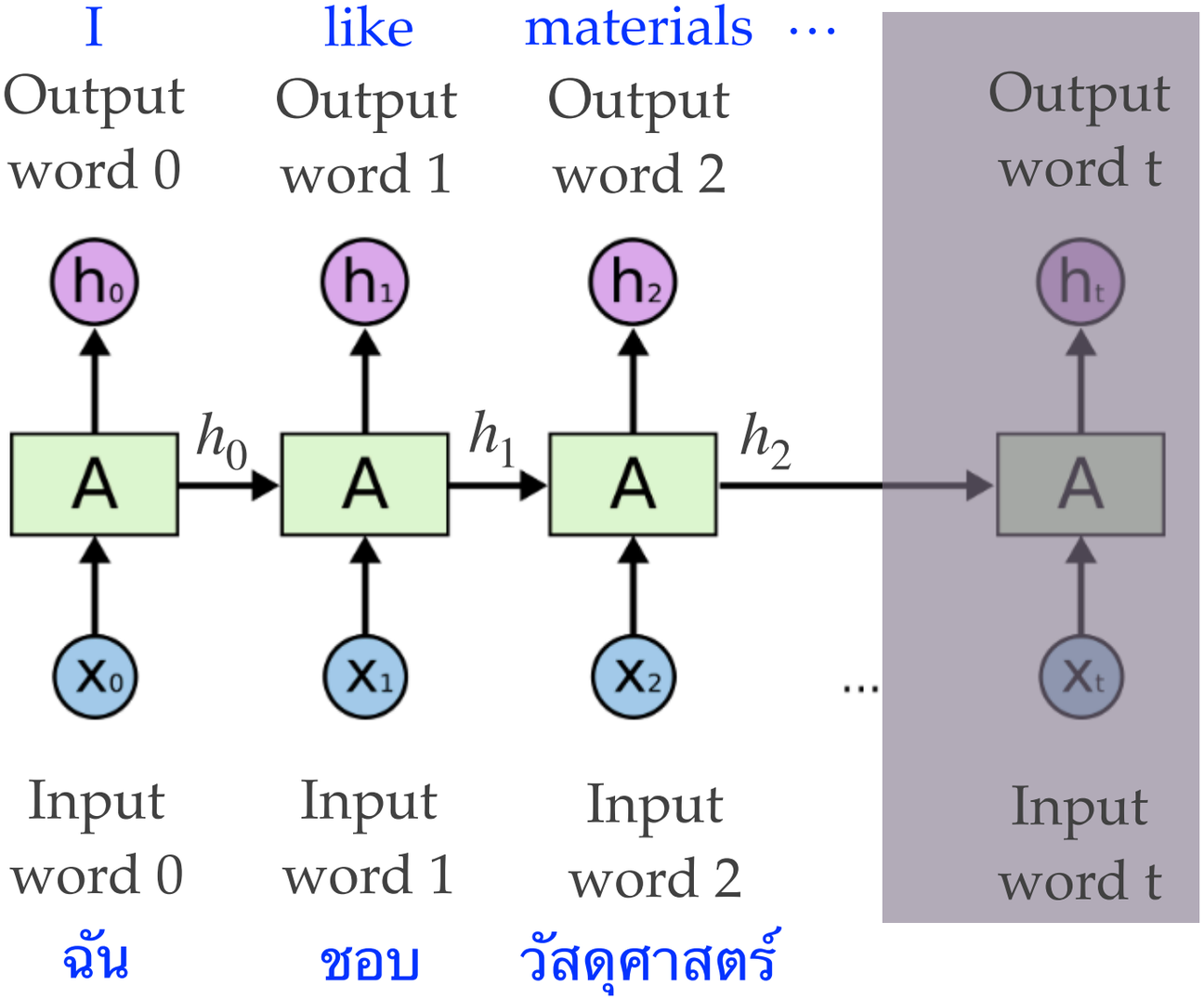


ฉันชอบวัสดุศาสตร์ → I like materials science

Understanding LSTM Networks by Christopher Colah

Recurrent Neural Network (RNN)

Task: Machine translation (e.g., translate language A to language B)

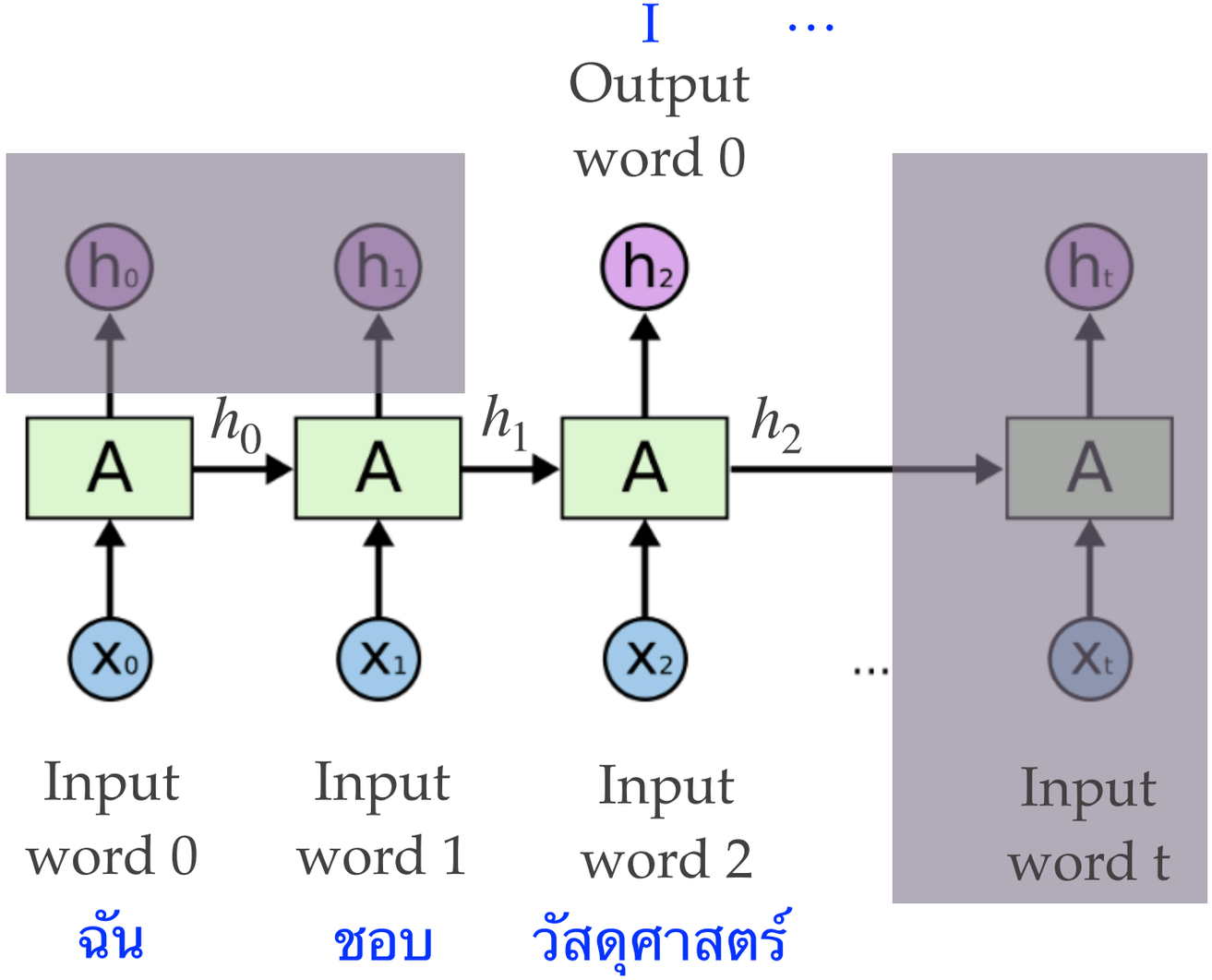


What is a potential problem with this model?

Understanding LSTM Networks by Christopher Colah

Recurrent Neural Network (RNN)

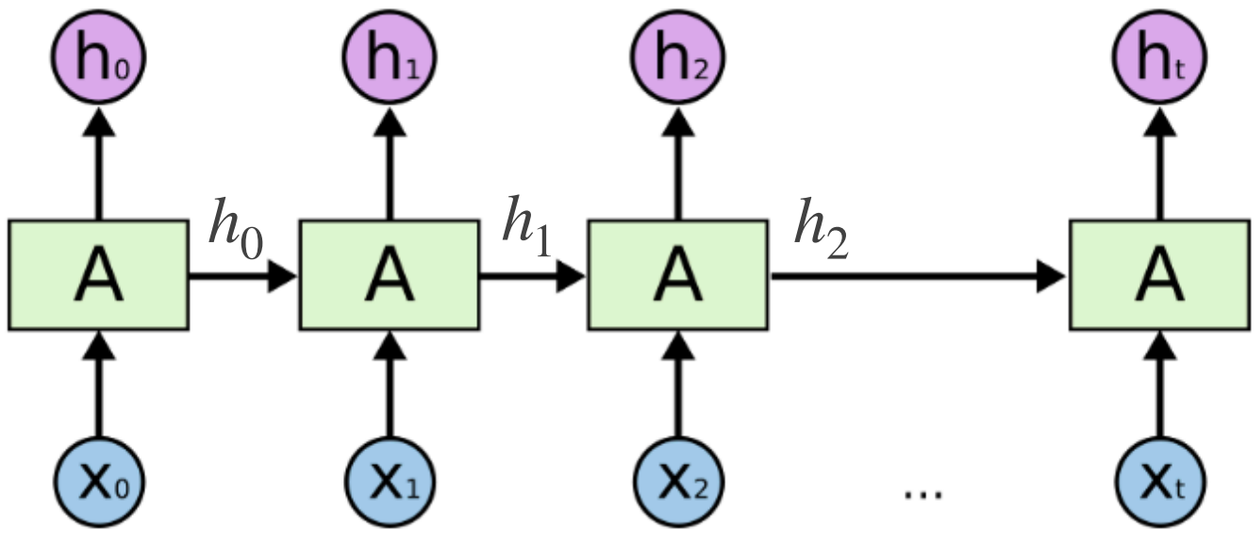
Task: Machine translation (e.g., translate language A to language B)



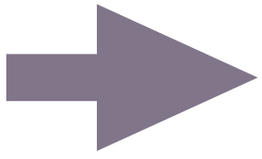
Understanding LSTM Networks by Christopher Colah

Recurrent Neural Network (RNN)

Task: Character-level language model sampling



h



(h)ello class. Today we are going to talk about recurrent neural networks.

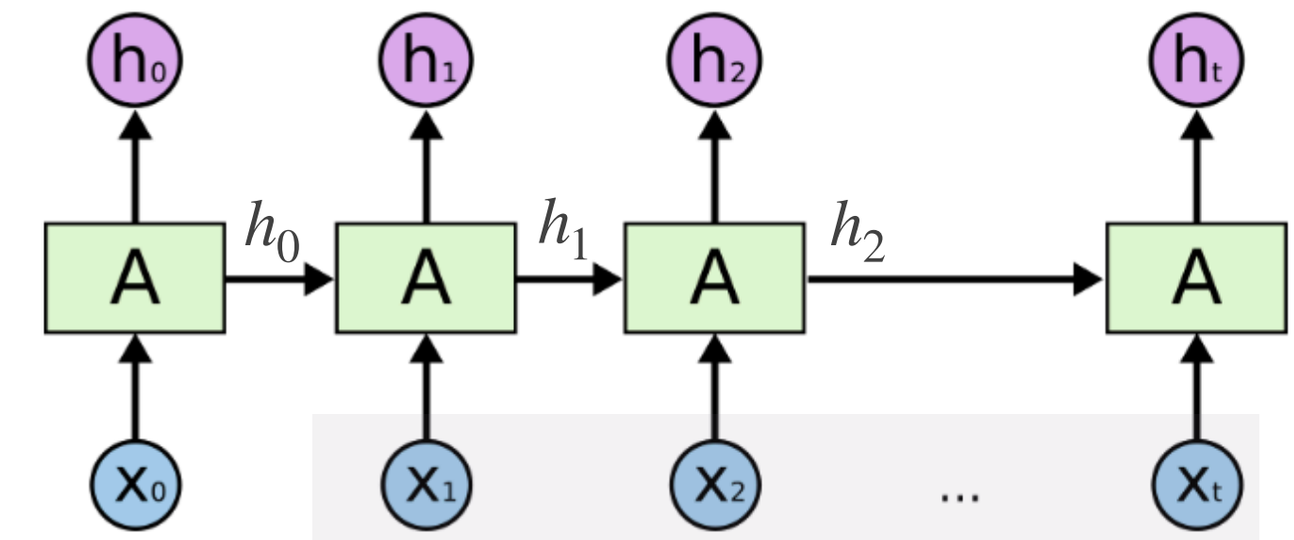
Understanding LSTM Networks by Christopher Colah

Recurrent Neural Network (RNN)

Task: Character-level language model sampling

e l l

Output character 0 | Output character 1 | Output character 2 | Output character t



Input character

h

"No" real additional input

Understanding LSTM Networks by Christopher Colah

keywords: greedy decoding, exhaustive search decoding, beam search decoding

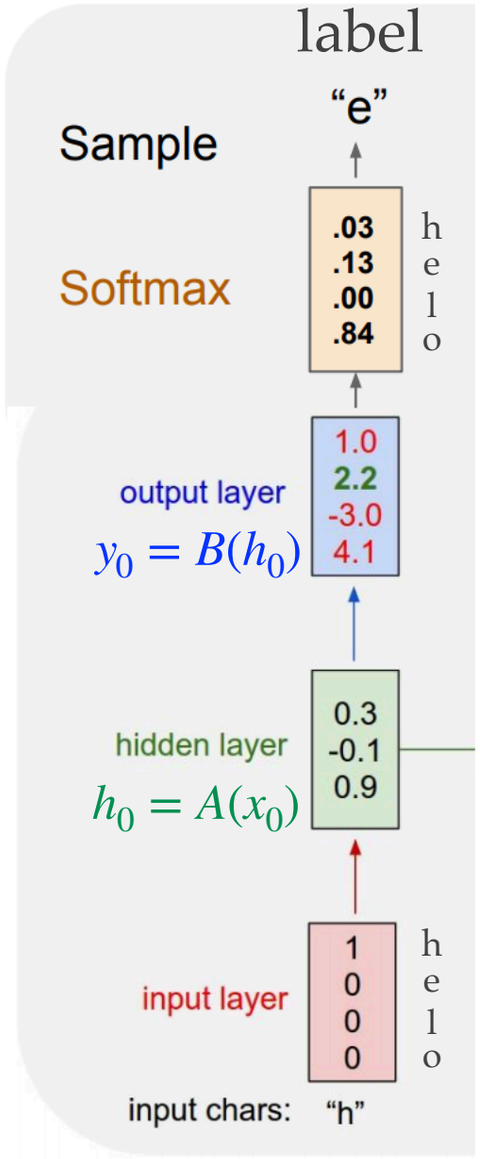
Recurrent Neural Network (RNN)

Task: Character-level language model sampling

Example:
Character-level
Language Model
Sampling

Vocabulary:
[h,e,l,o]

At test-time sample characters one at a time, feed back to model



One-hot encoding

keywords: greedy decoding, exhaustive search decoding, beam search decoding

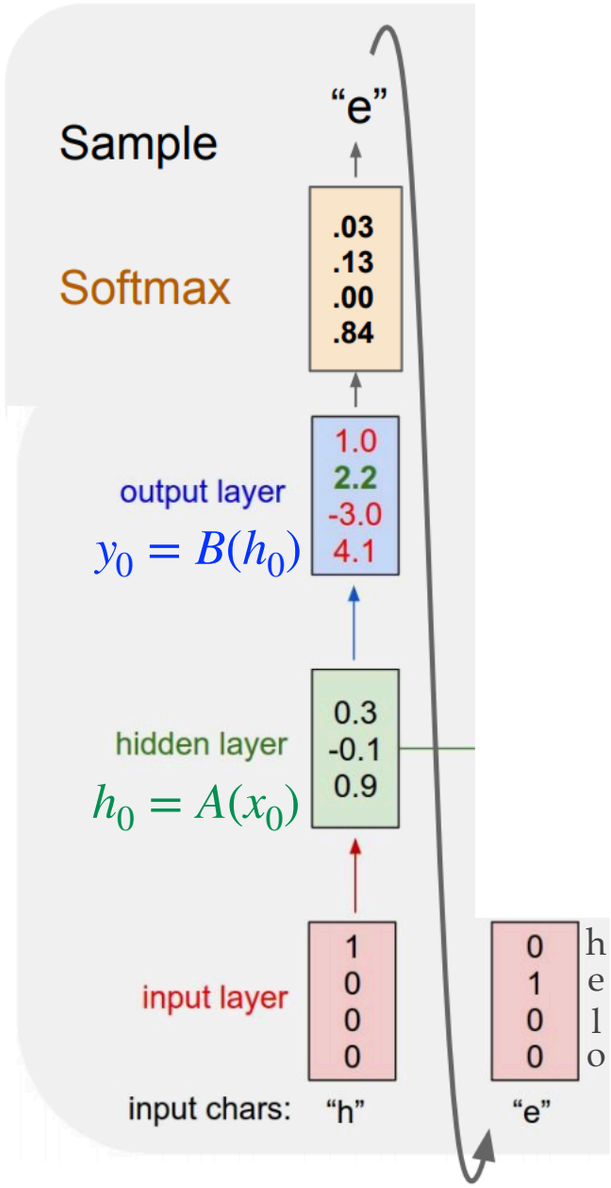
Recurrent Neural Network (RNN)

Task: Character-level language model sampling

Example: Character-level Language Model Sampling

Vocabulary:
[h,e,l,o]

At test-time sample characters one at a time, feed back to model



keywords: greedy decoding, exhaustive search decoding, beam search decoding

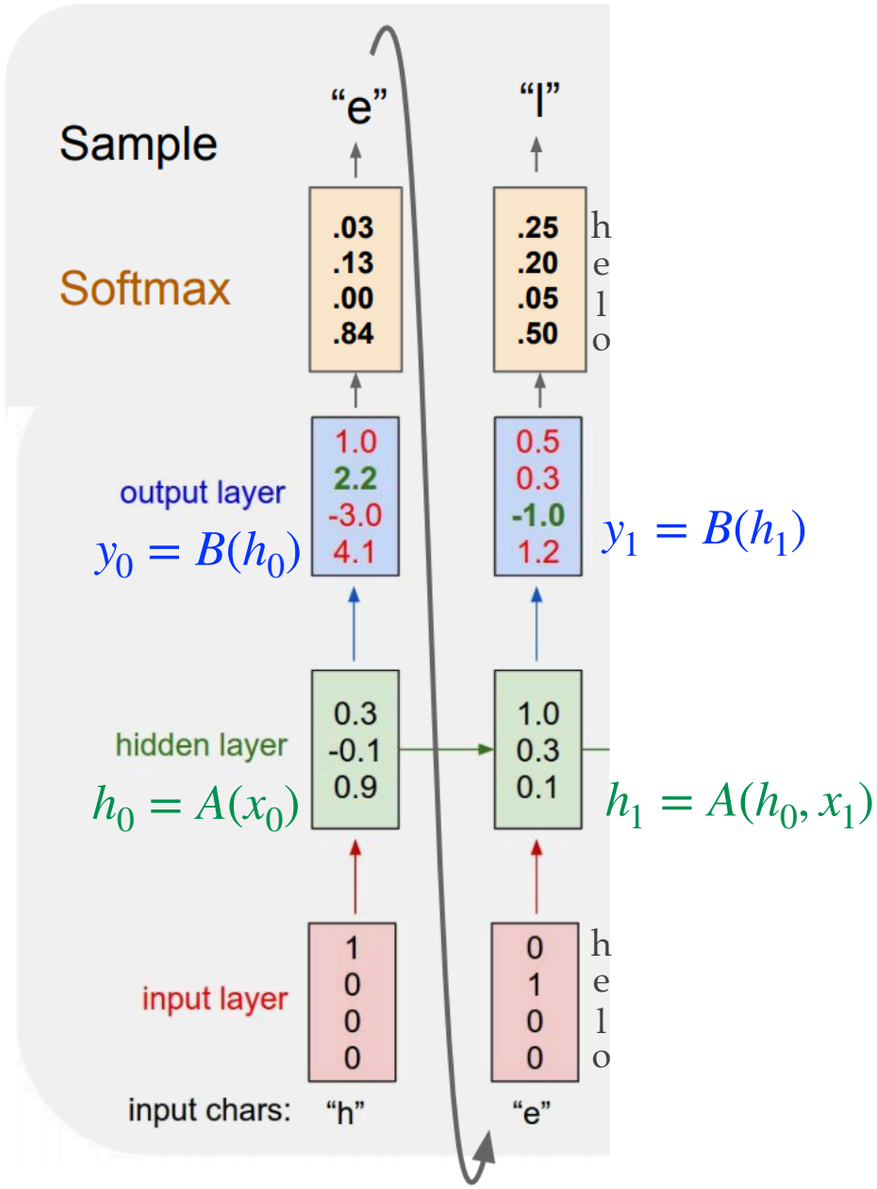
Recurrent Neural Network (RNN)

Task: Character-level language model sampling

Example: Character-level Language Model Sampling

Vocabulary:
[h,e,l,o]

At test-time sample characters one at a time, feed back to model



keywords: greedy decoding, exhaustive search decoding, beam search decoding

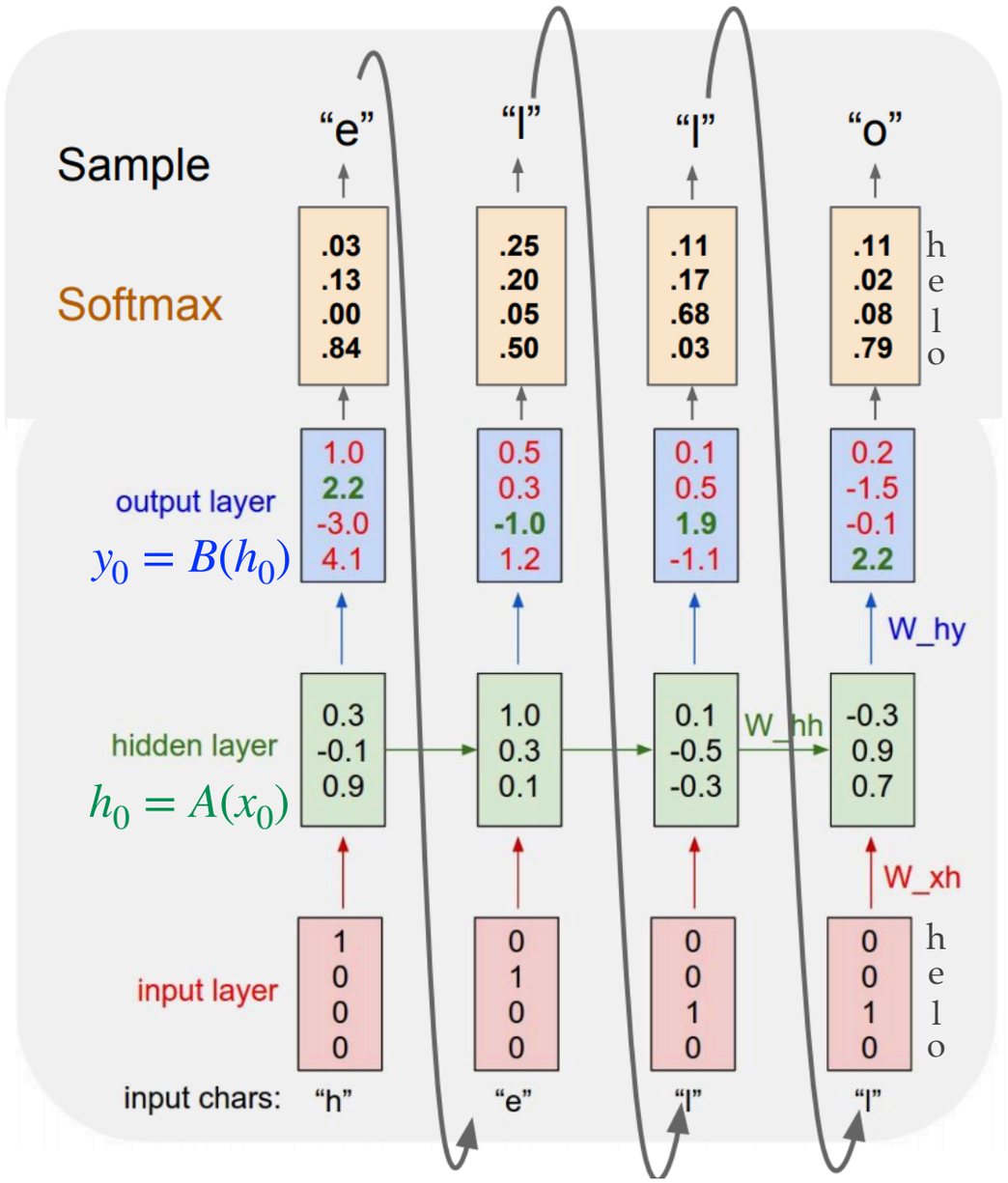
Recurrent Neural Network (RNN)

Task: Character-level language model sampling

Example: Character-level Language Model Sampling

Vocabulary:
[h,e,l,o]

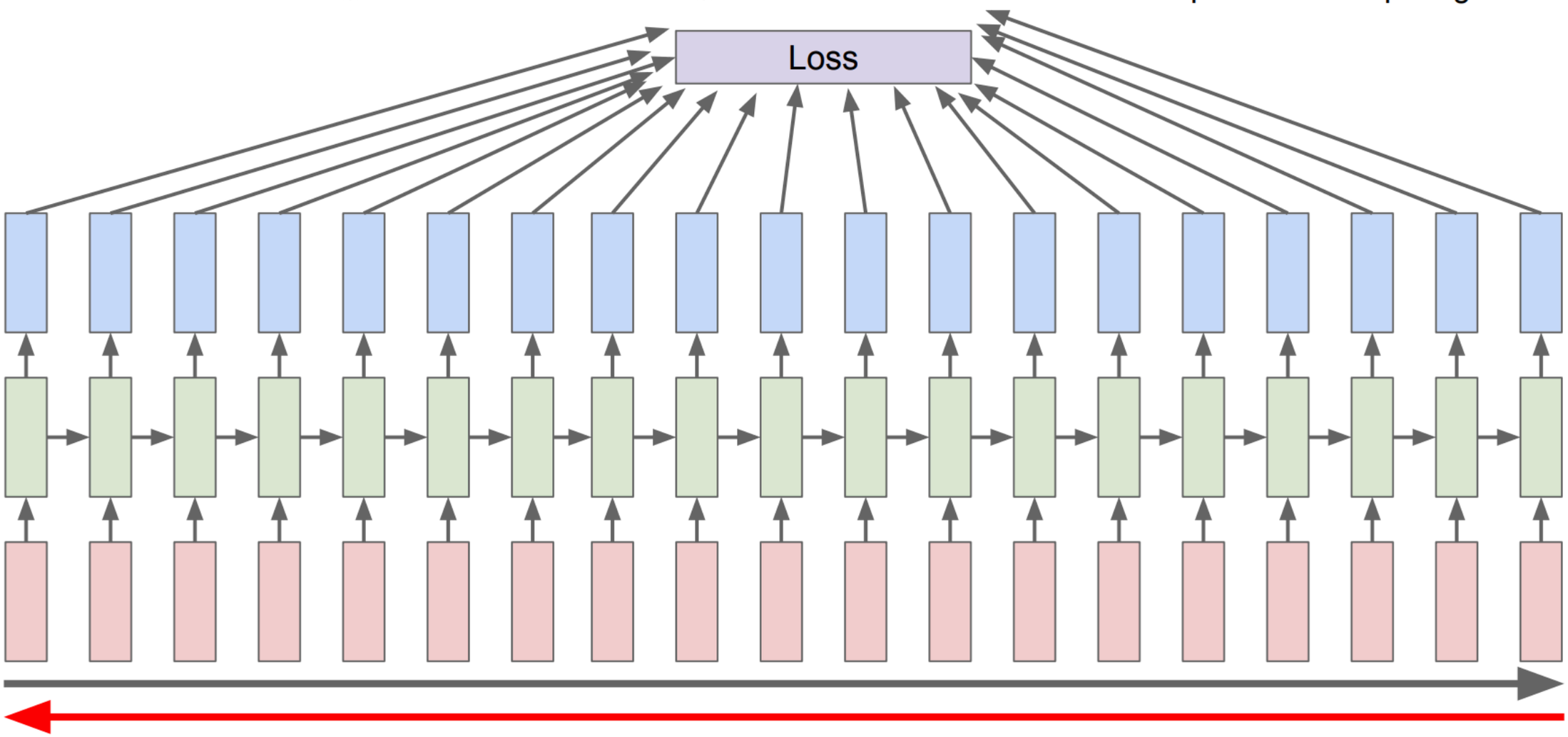
At test-time sample characters one at a time, feed back to model



RNN - Backpropagation

Backpropagation through time

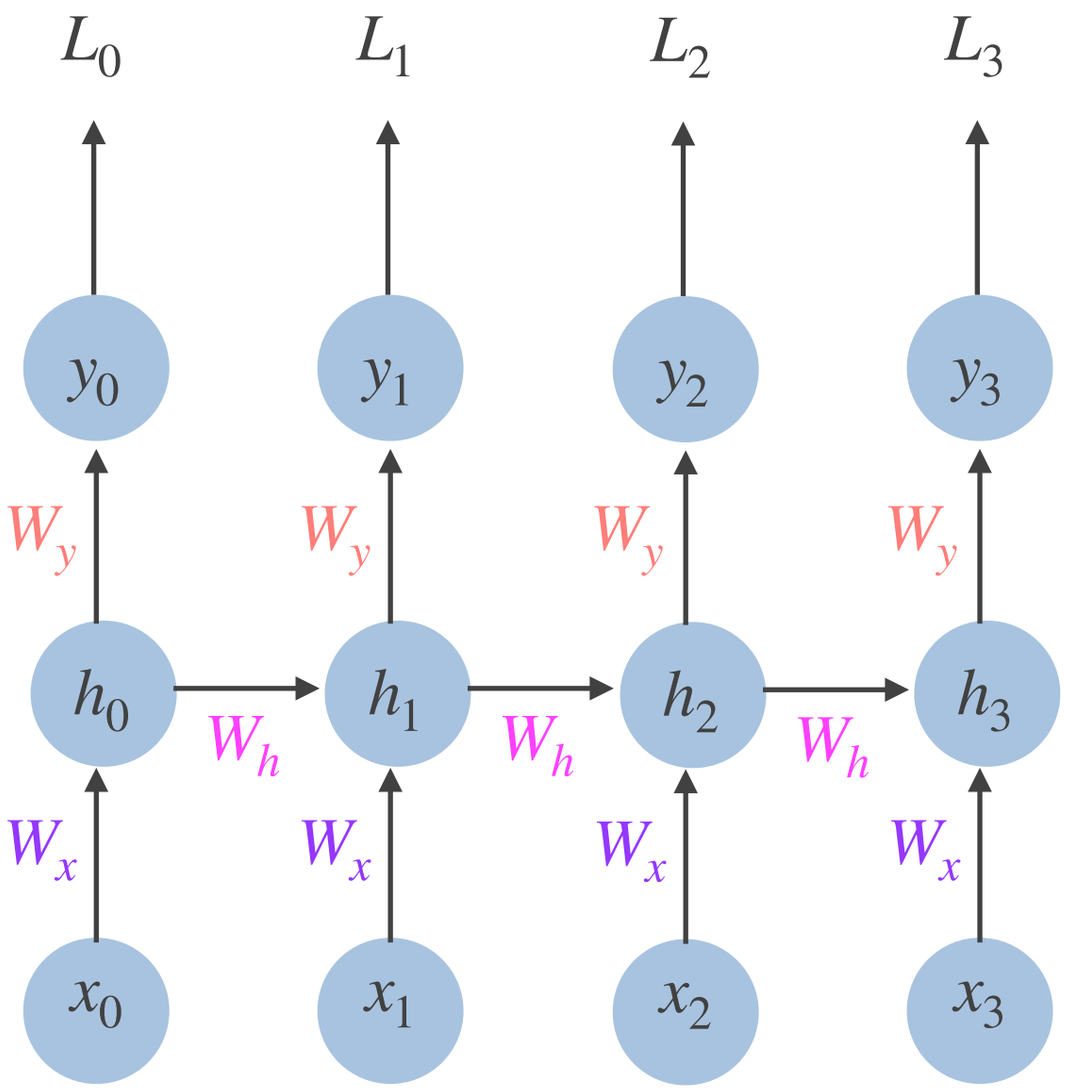
Forward through entire sequence to compute loss, then backward through entire sequence to compute gradient



Potentially encounter vanishing/exploding gradient problems!

Stanford CS231n (2019): Lecture 10

RNN - Backpropagation



$$h_t = \tanh(W_x x_t + W_h h_{t-1})$$

$$y_t = f(W_y h_t)$$

Want to update W_x using stochastic gradient descent

Loss function $L = L_0 + L_1 + L_2 + L_3$

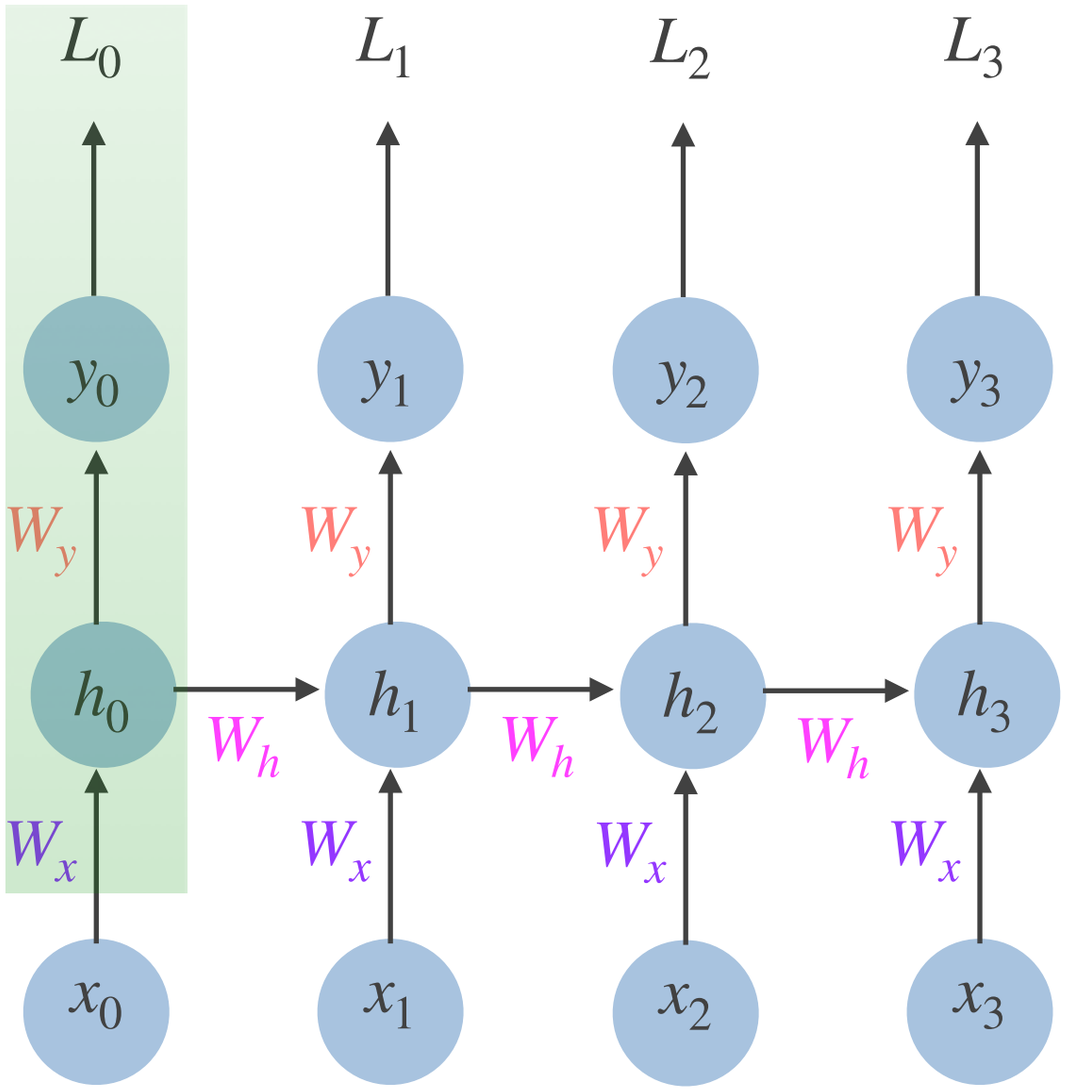
Update formula for iteration k

$$W_{x,(k)} = W_{x,(k-1)} - \alpha \nabla_{W_x} L(W_x)\,|_{W_x = W_{x,(k-1)}}$$

Decompose the gradient of L

$$\nabla_{W_x} L = \nabla_{W_x} L_0 + \nabla_{W_x} L_1 + \nabla_{W_x} L_2 + \nabla_{W_x} L_3$$

RNN - Backpropagation



$$\nabla_{W_x} L = \nabla_{W_x} L_0 + \nabla_{W_x} L_1 + \nabla_{W_x} L_2 + \nabla_{W_x} L_3$$

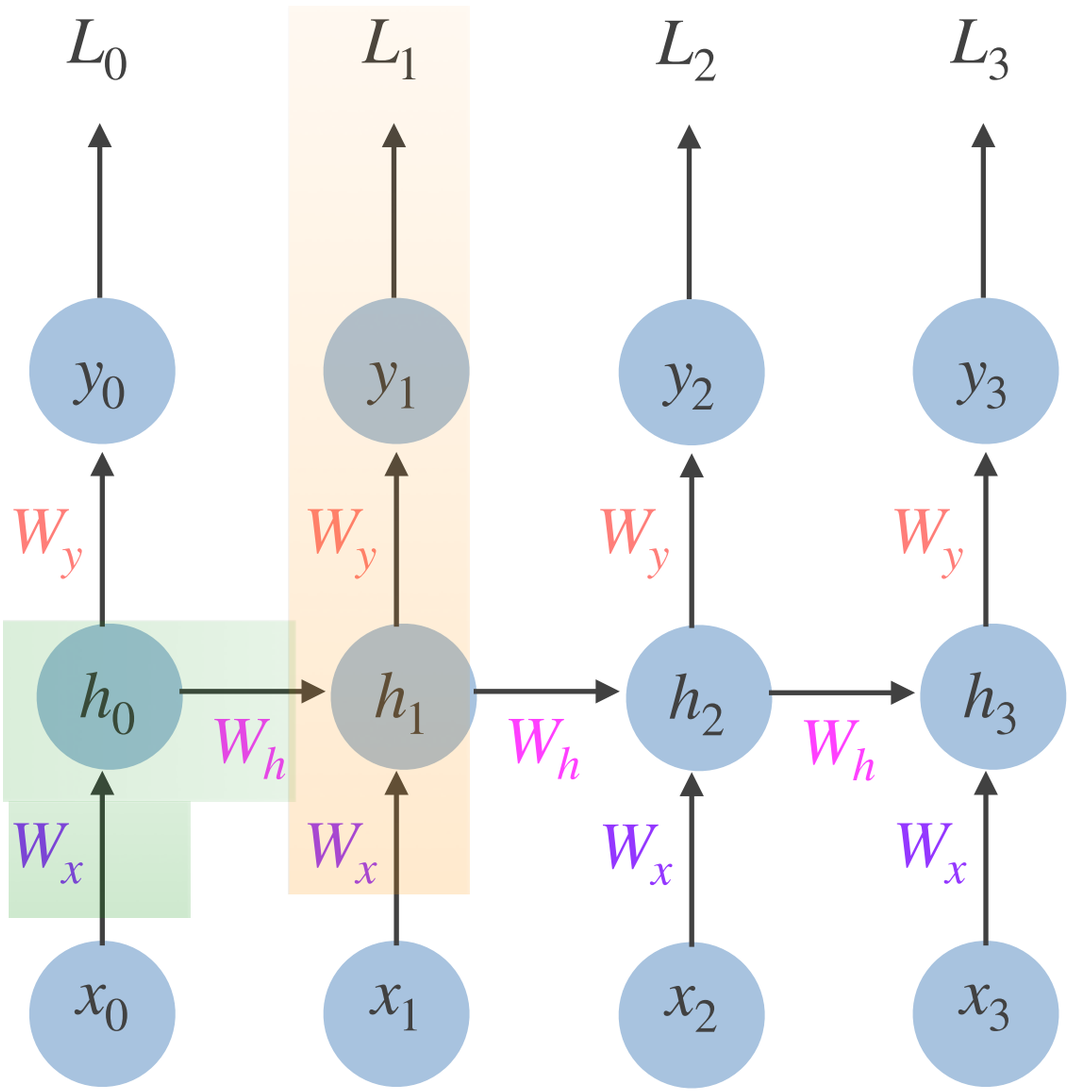
Using the chain rule

$$\frac{\partial L_0}{\partial W_x} = \frac{\partial L_0}{\partial y_0}\frac{\partial y_0}{\partial h_0}\frac{\partial h_0}{\partial W_x}$$

$$h_t = \tanh(W_x x_t + W_h h_{t-1})$$

$$y_t = f(W_y h_t)$$

RNN - Backpropagation



$$\nabla_{W_x} L = \nabla_{W_x} L_0 + \nabla_{W_x} L_1 + \nabla_{W_x} L_2 + \nabla_{W_x} L_3$$

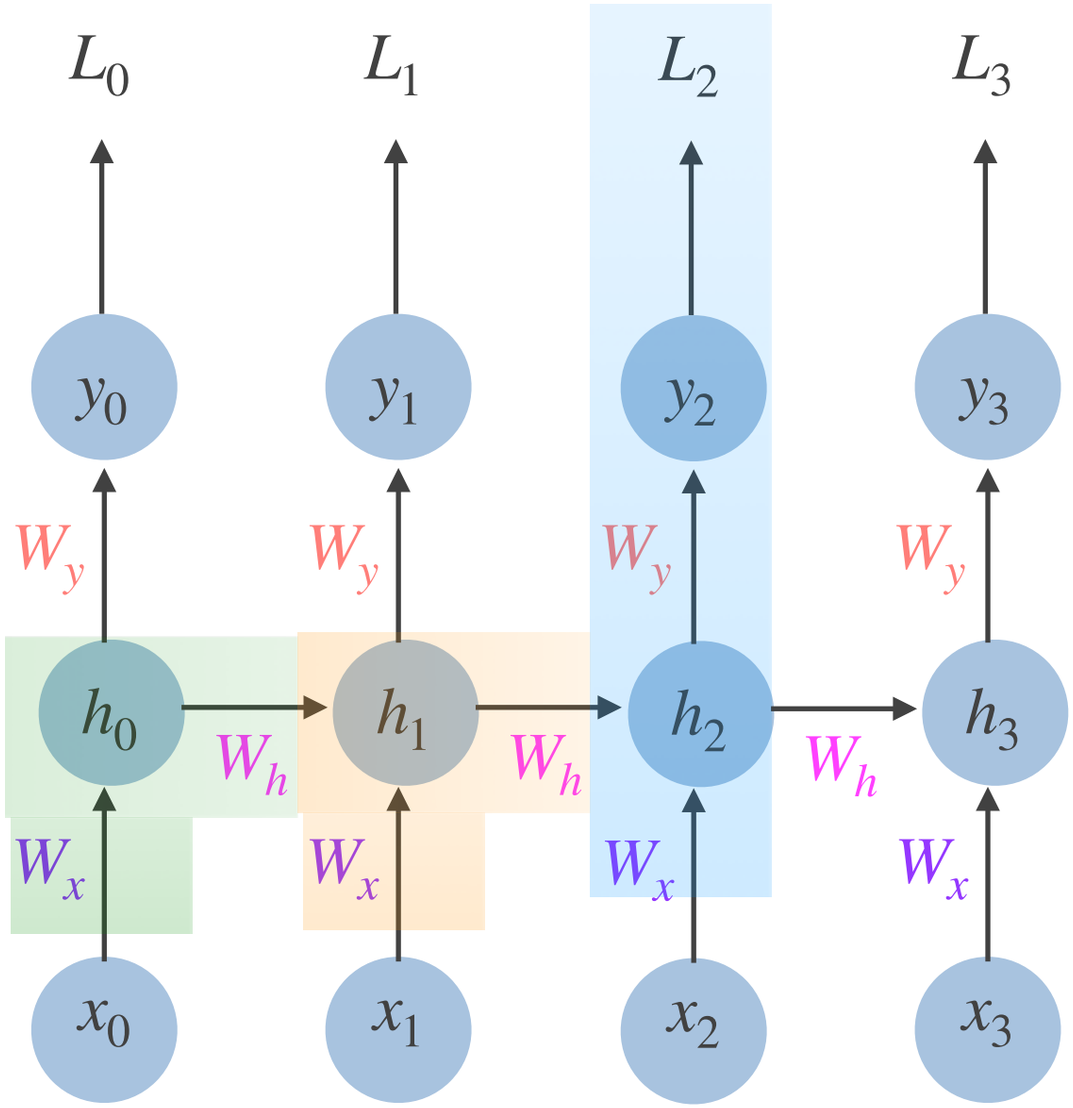
Using the chain rule

$$\frac{\partial L_1}{\partial W_x} = \frac{\partial L_1}{\partial y_1}\frac{\partial y_1}{\partial h_1}\left[\frac{\partial h_1}{\partial W_x} + \frac{\partial h_1}{\partial h_0}\frac{\partial h_0}{\partial W_x}\right]$$

$$h_t = \tanh(W_x x_t + W_h h_{t-1})$$

$$y_t = f(W_y h_t)$$

RNN - Backpropagation



$$\nabla_{W_x} L = \nabla_{W_x} L_0 + \nabla_{W_x} L_1 + \nabla_{W_x} L_2 + \nabla_{W_x} L_3$$

Using the chain rule

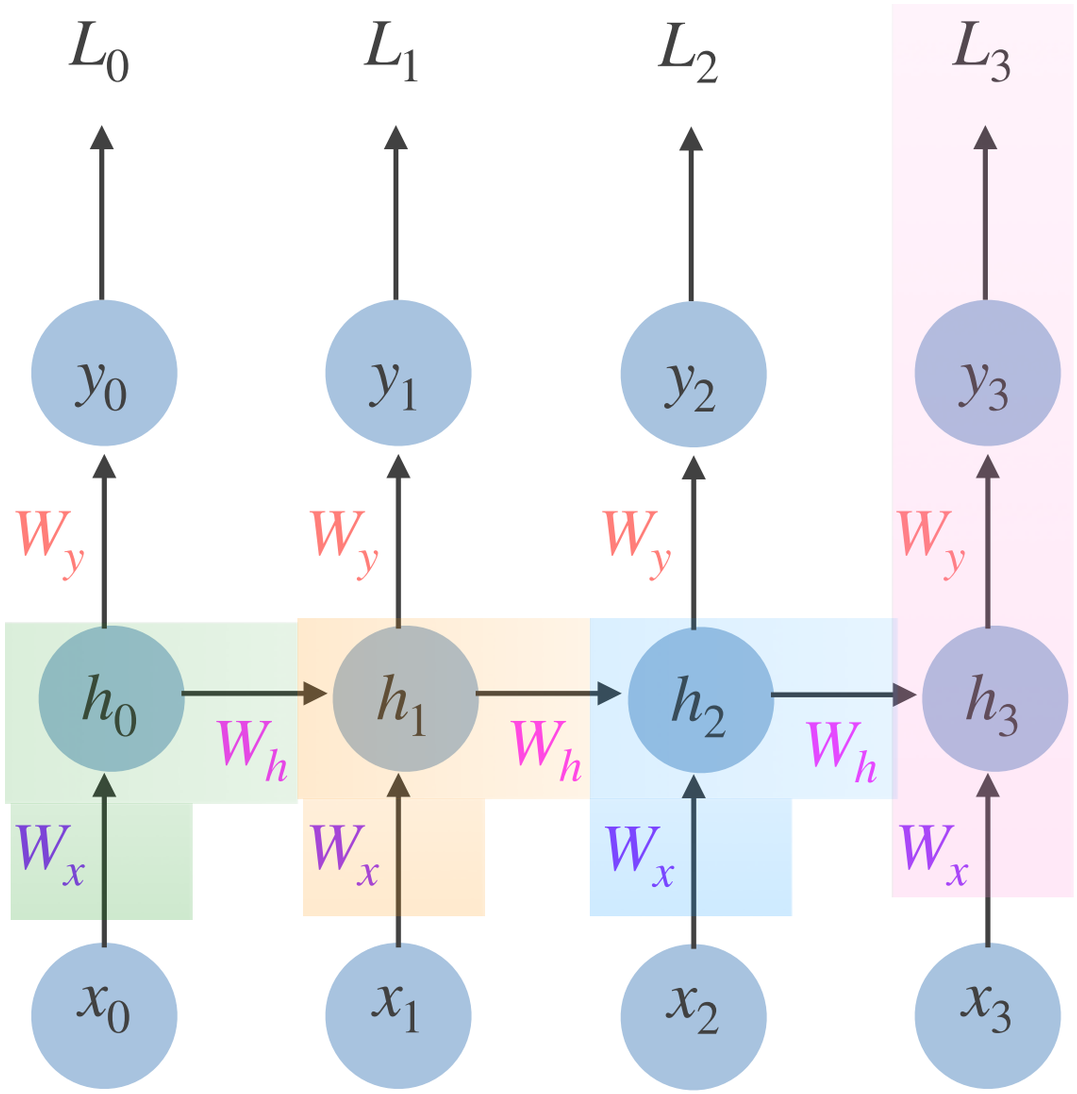
$$\frac{\partial L_2}{\partial W_x} = \frac{\partial L_2}{\partial y_2}\frac{\partial y_2}{\partial h_2}\cdots$$

$$\left[\frac{\partial h_2}{\partial W_x} + \frac{\partial h_2}{\partial h_1}\frac{\partial h_1}{\partial W_x} + \frac{\partial h_2}{\partial h_1}\frac{\partial h_1}{\partial h_0}\frac{\partial h_0}{\partial W_x}\right]$$

$$h_t = \tanh(W_x x_t + W_h h_{t-1})$$

$$y_t = f(W_y h_t)$$

RNN - Backpropagation



$$h_t = \tanh(W_x x_t + W_h h_{t-1})$$

$$y_t = f(W_y h_t)$$

$$\nabla_{W_x} L = \nabla_{W_x} L_0 + \nabla_{W_x} L_1 + \nabla_{W_x} L_2 + \nabla_{W_x} L_3$$

Using the chain rule

$$\frac{\partial L_3}{\partial W_x} = \frac{\partial L_3}{\partial y_3}\frac{\partial y_3}{\partial h_3}\cdots$$

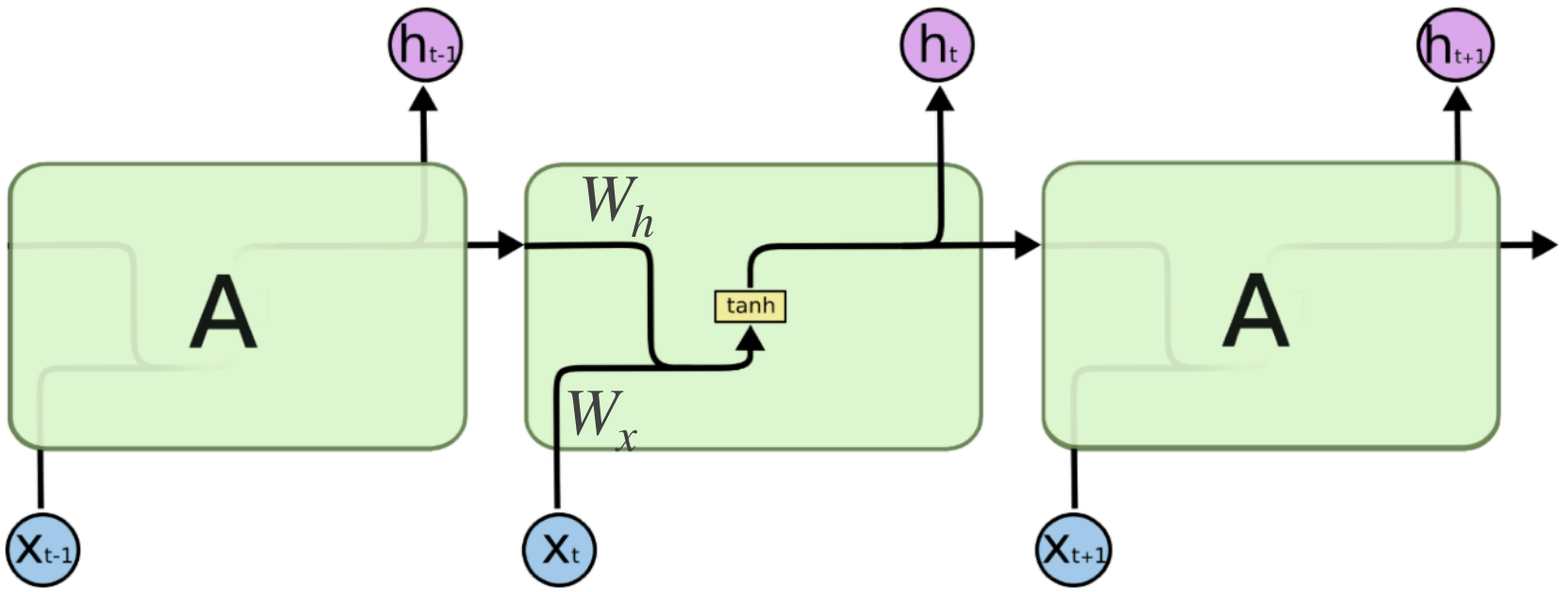
$$\left[\frac{\partial h_3}{\partial W_x} + \frac{\partial h_3}{\partial h_2}\frac{\partial h_2}{\partial W_x} + \frac{\partial h_3}{\partial h_2}\frac{\partial h_2}{\partial h_1}\frac{\partial h_1}{\partial W_x} + \frac{\partial h_3}{\partial h_2}\frac{\partial h_2}{\partial h_1}\frac{\partial h_1}{\partial h_0}\frac{\partial h_0}{\partial W_x}\right]$$

1 term 2 terms 3 terms 4 terms

More terms…potential vanishing/ exploding gradient problems

While the exploding gradient problem can be controlled with gradient clipping, the vanishing gradient problem is more difficult to solve.

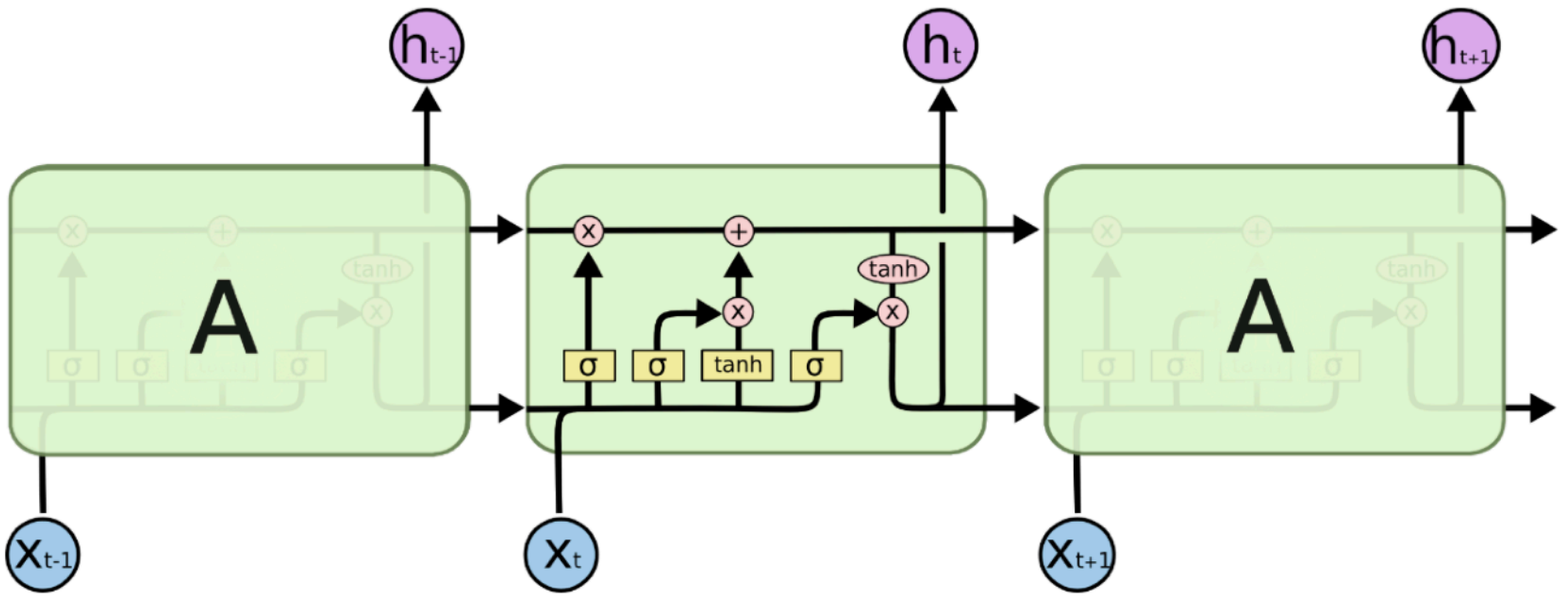
Long Short-Term Memory (LSTM)



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

$$h_t = \tanh(W_x x_t + W_h h_{t-1} + b)$$

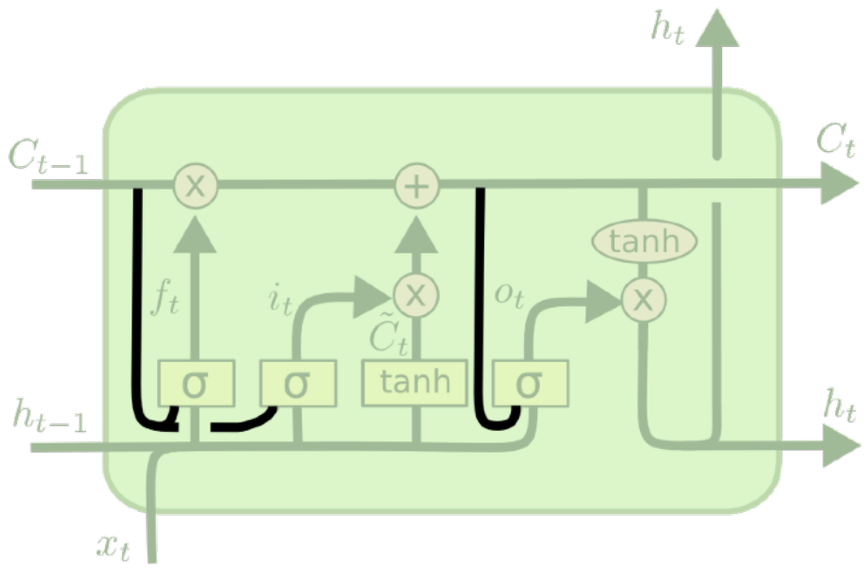
h_t could be the hidden state/output



The repeating module in an LSTM contains four interacting layers.

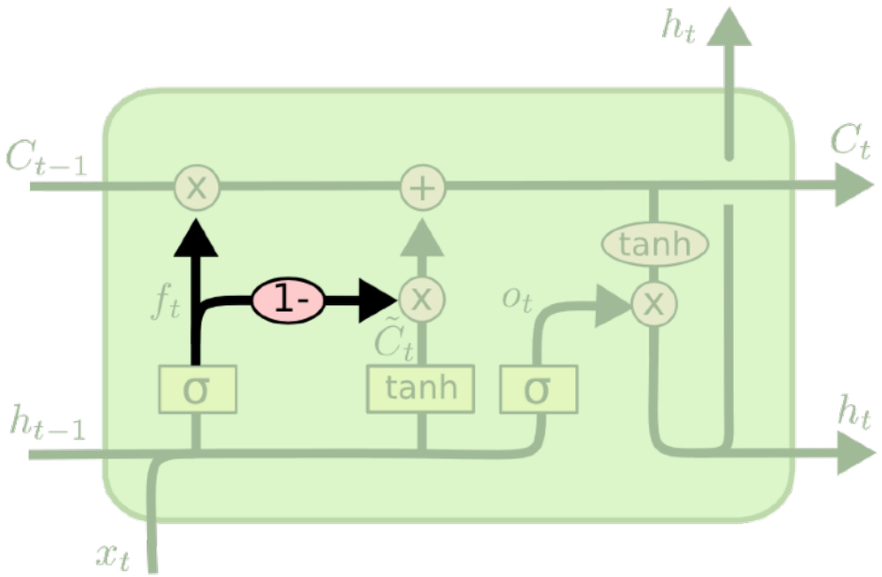
$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$$
$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)$$
$$\tilde{C}_t = \tanh(W_C[h_{t-1}, x_t] + b_C)$$
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$
$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$
$$h_t = o_t * \tanh(C_t)$$

Understanding LSTM Networks by Christopher Colah



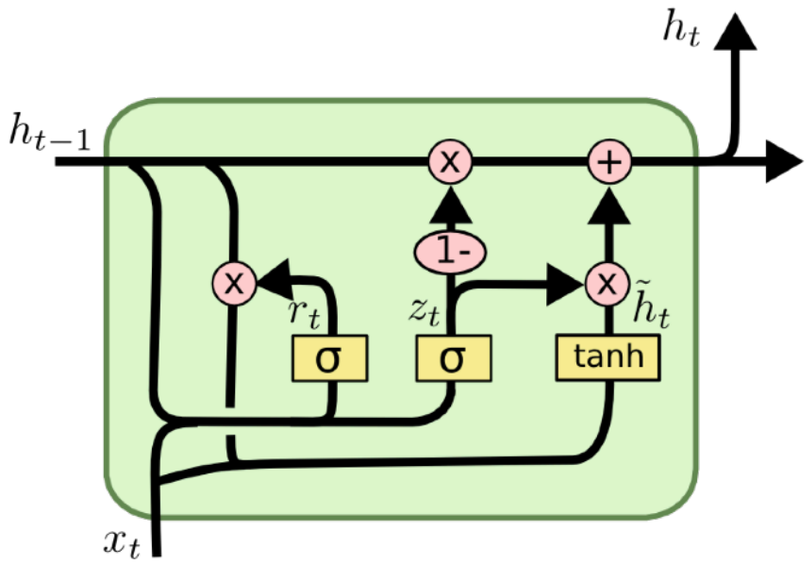
LSTM with peephole connections

$$f_t = \sigma\left(W_f \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] \; + \; b_f\right)$$
$$i_t = \sigma\left(W_i \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] \; + \; b_i\right)$$
$$o_t = \sigma\left(W_o \cdot [\boldsymbol{C_t}, h_{t-1}, x_t] \; + \; b_o\right)$$



LSTM with coupled input and forget gates

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



Gated Recurrent Units (GRU)

$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right) \qquad \text{update gate}$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right) \qquad \text{reset gate}$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Understanding LSTM Networks by Christopher Colah

Examples

at first:

```
tyntd-iafhatawiaoihrdemot  lytdws  e ,tfti, astai f ogoh eoase rrranbyne 'nhthnee e
plia tklrgd t o idoe ns,smtt   h ne etie h,hregtrs nigtike,aoaenns lng
```

train more

```
"Tmont thithey" fomesscerliund
Keushey. Thom here
sheulke, anmerenith ol sivh I lalterthend Bleipile shuwy fil on aseterlome
coaniogennc Phe lism thond hon at. MeiDimorotion in ther thize."
```

train more

```
Aftair fall unsuch that the hall for Prince Velzonski's that me of
her hearly, and behs to so arwage fiving were to it beloge, pavu say falling misfort
how, and Gogition is so overelical and ofter.
```

train more

```
"Why do what that day," replied Natasha, and wishing to himself the fact the
princess, Princess Mary was easier, fed in had oftened him.
Pierre aking his soul came to the packs and drove up his father-in-law women.
```

Examples

Proof. Omitted. □

Lemma 0.1. *Let $\mathcal{C}$ be a set of the construction.*

Let $\mathcal{C}$ be a gerber covering. Let $\mathcal{F}$ be a quasi-coherent sheaves of $\mathcal{O}$-modules. We have to show that

$$\mathcal{O}_{\mathcal{O}_X} = \mathcal{O}_X(\mathcal{L})$$

.

Proof. This is an algebraic space with the composition of sheaves $\mathcal{F}$ on $X_{étale}$ we have

$$\mathcal{O}_X(\mathcal{F}) = \{morph_1 \times_{\mathcal{O}_X} (\mathcal{G}, \mathcal{F})\}$$

where $\mathcal{G}$ defines an isomorphism $\mathcal{F} \to \mathcal{F}$ of $\mathcal{O}$-modules. □

Lemma 0.2. *This is an integer $\mathcal{Z}$ is injective.*

Proof. See Spaces, Lemma ??. □

Lemma 0.3. *Let S be a scheme. Let X be a scheme and X is an affine open covering. Let $\mathcal{U} \subset \mathcal{X}$ be a canonical and locally of finite type. Let X be a scheme. Let X be a scheme which is equal to the formal complex.*

The following to the construction of the lemma follows.

Let X be a scheme. Let X be a scheme covering. Let

$$b : X \to Y' \to Y \to Y \to Y' \times_X Y \to X.$$

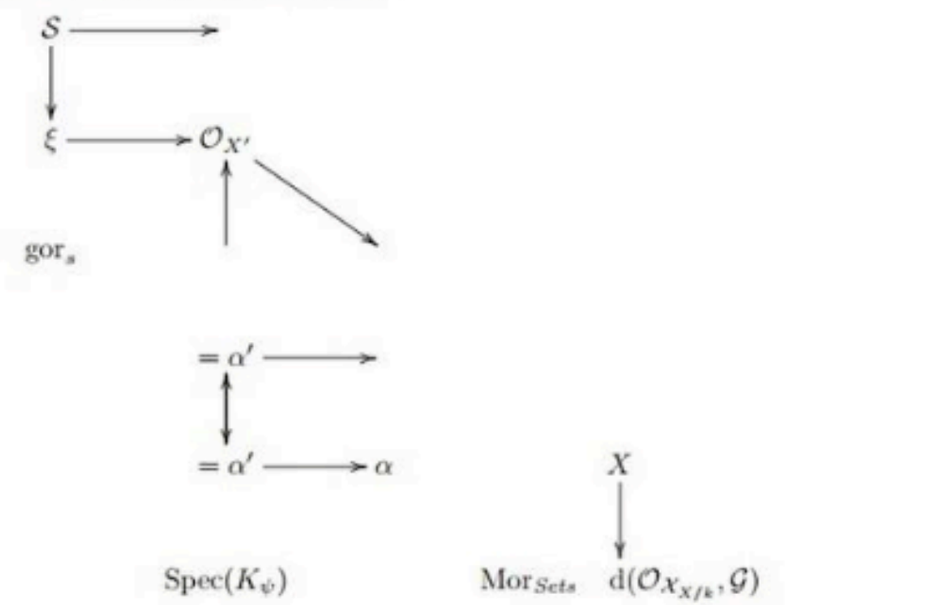
be a morphism of algebraic spaces over S and Y.

Proof. Let X be a nonzero scheme of X. Let X be an algebraic space. Let $\mathcal{F}$ be a quasi-coherent sheaf of $\mathcal{O}_X$-modules. The following are equivalent

(1) $\mathcal{F}$ is an algebraic space over S.
(2) If X is an affine open covering.

Consider a common structure on X and X the functor $\mathcal{O}_X(U)$ which is locally of finite type. □

This since $\mathcal{F} \in \mathcal{F}$ and $x \in \mathcal{G}$ the diagram



is a limit. Then $\mathcal{G}$ is a finite type and assume S is a flat and $\mathcal{F}$ and $\mathcal{G}$ is a finite type f_*. This is of finite type diagrams, and

- the composition of $\mathcal{G}$ is a regular sequence,
- $\mathcal{O}_{X'}$ is a sheaf of rings.

□

Proof. We have see that $X = \mathrm{Spec}(R)$ and $\mathcal{F}$ is a finite type representable by algebraic space. The property $\mathcal{F}$ is a finite morphism of algebraic stacks. Then the cohomology of X is an open neighbourhood of U. □

Proof. This is clear that $\mathcal{G}$ is a finite presentation, see Lemmas ??.
A *reduced above* we conclude that U is an open covering of $\mathcal{C}$. The functor $\mathcal{F}$ is a "field

$$\mathcal{O}_{X,x} \longrightarrow \mathcal{F}_{\overline{x}} \quad \text{-}1(\mathcal{O}_{X_{\acute{e}tale}}) \longrightarrow \mathcal{O}_{X_\ell}^{-1}\mathcal{O}_{X_\lambda}(\mathcal{O}_{X_\eta}^{\overline{v}})$$

is an isomorphism of covering of $\mathcal{O}_{X_i}$. If $\mathcal{F}$ is the unique element of $\mathcal{F}$ such that X is an isomorphism.
The property $\mathcal{F}$ is a disjoint union of Proposition ?? and we can filtered set of presentations of a scheme $\mathcal{O}_X$-algebra with $\mathcal{F}$ are opens of finite type over S.
If $\mathcal{F}$ is a scheme theoretic image points. □

If $\mathcal{F}$ is a finite direct sum $\mathcal{O}_{X_\lambda}$ is a closed immersion, see Lemma ??. This is a sequence of $\mathcal{F}$ is a similar morphism.

Examples

```
static void do_command(struct seq_file *m, void *v)
{
  int column = 32 << (cmd[2] & 0x80);
  if (state)
    cmd = (int)(int_state ^ (in_8(&ch->ch_flags) & Cmd) ? 2 : 1);
  else
    seq = 1;
  for (i = 0; i < 16; i++) {
    if (k & (1 << 1))
      pipe = (in_use & UMXTHREAD_UNCCA) +
        ((count & 0x00000000fffffff8) & 0x000000f) << 8;
    if (count == 0)
      sub(pid, ppc_md.kexec_handle, 0x20000000);
    pipe_set_bytes(i, 0);
  }
  /* Free our user pages pointer to place camera if all dash */
  subsystem_info = &of_changes[PAGE_SIZE];
  rek_controls(offset, idx, &soffset);
  /* Now we want to deliberately put it to device */
  control_check_polarity(&context, val, 0);
  for (i = 0; i < COUNTER; i++)
    seq_puts(s, "policy ");
}
```

Generated C code