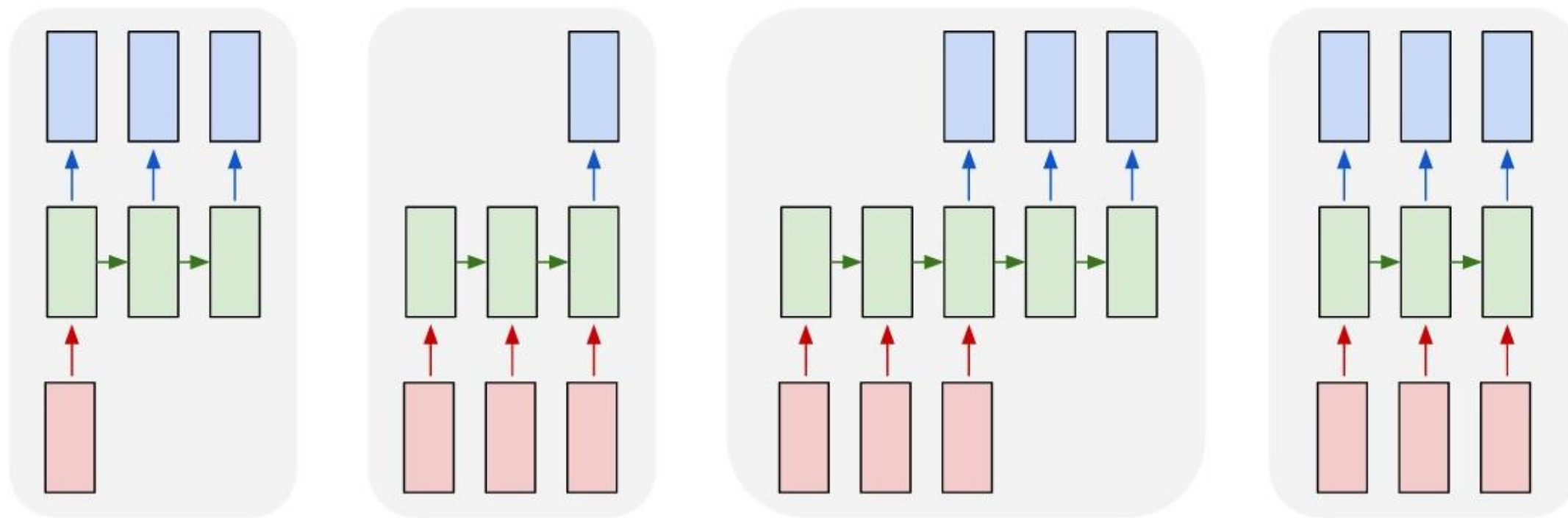


Recurrent neural networks



Outline

- Examples of sequential prediction tasks
- Common recurrent units
 - Vanilla RNN unit (and how to train it)
 - Long Short-Term Memory (LSTM)
 - Gated Recurrent Unit (GRU)
- Recurrent network architectures
- Applications in (a bit) more detail
 - Sequence classification
 - Language modeling
 - Image captioning
 - Machine translation

Sequential prediction tasks

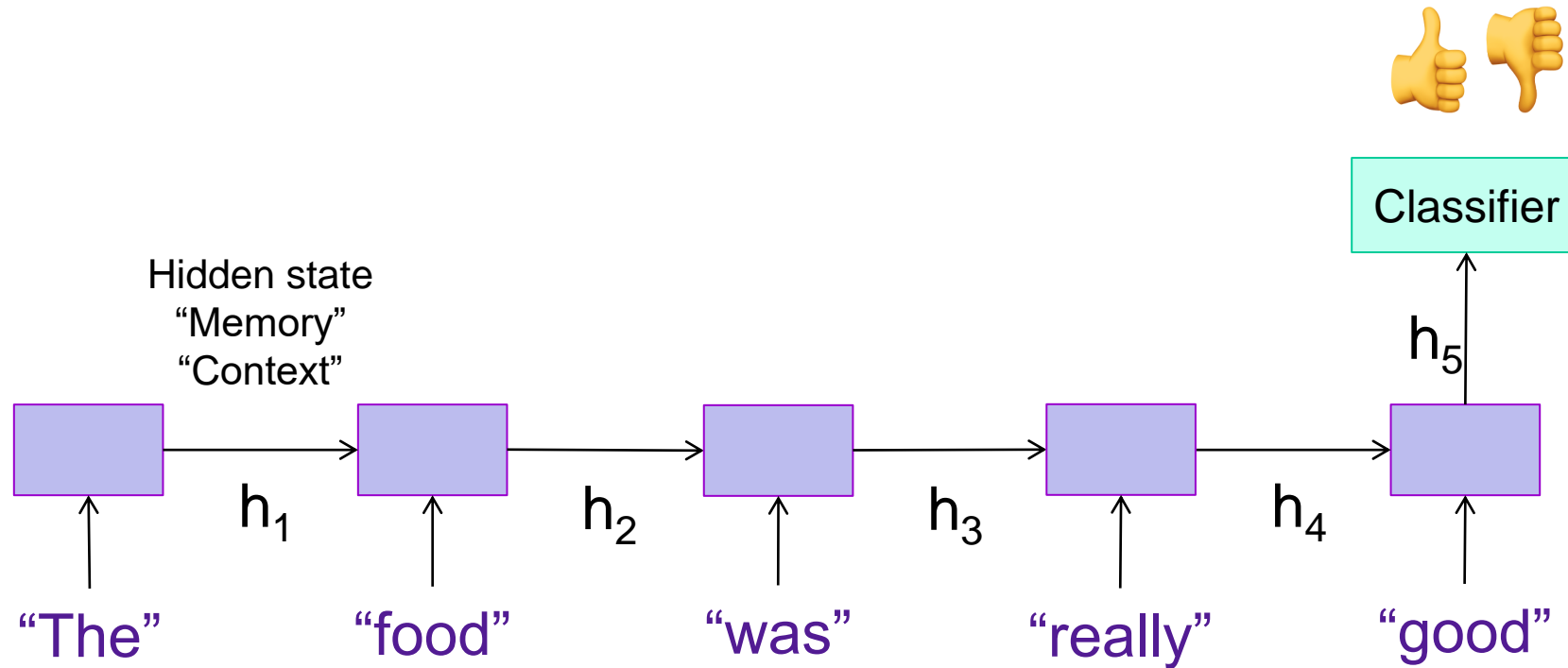
- So far, we focused mainly on prediction problems with fixed-size inputs and outputs
- But what if the input and/or output is a variable-length sequence?

Example 1: Sentiment classification

- Goal: classify a text sequence (e.g., restaurant, movie or product review, Tweet) as having positive or negative sentiment
 - “The food was really good”
 - “The vacuum cleaner broke within two weeks”
 - “The movie had slow parts, but overall was worth watching”
- What makes this problem challenging?
- What feature representation or predictor structure can we use for this problem?

Example 1: Sentiment classification

- Recurrent model:



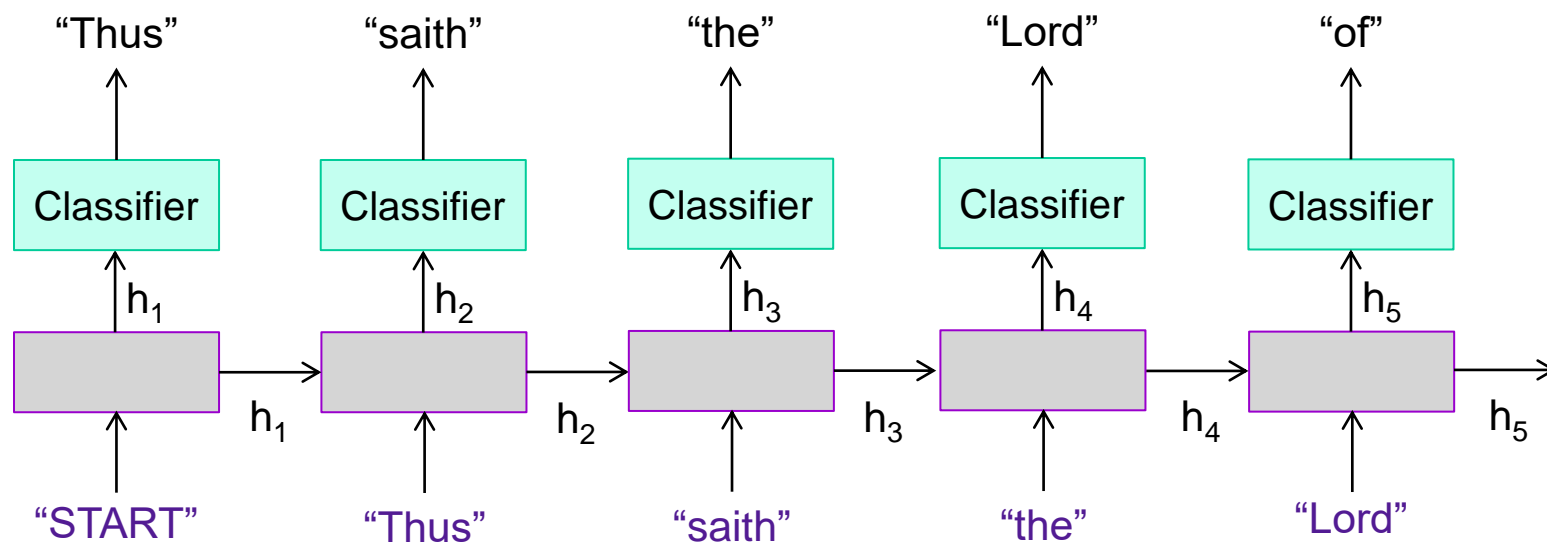
Example 2: Text generation

- Sample from the distribution of a given text corpus (also known as language modeling)



Example 2: Text generation

- Sample from the distribution of a given text corpus (also known as language modeling)
- Can be done one character or one word at a time:



Example 3: Image caption generation



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



Two people walking on the beach with surfboards



A tennis player in action on the court

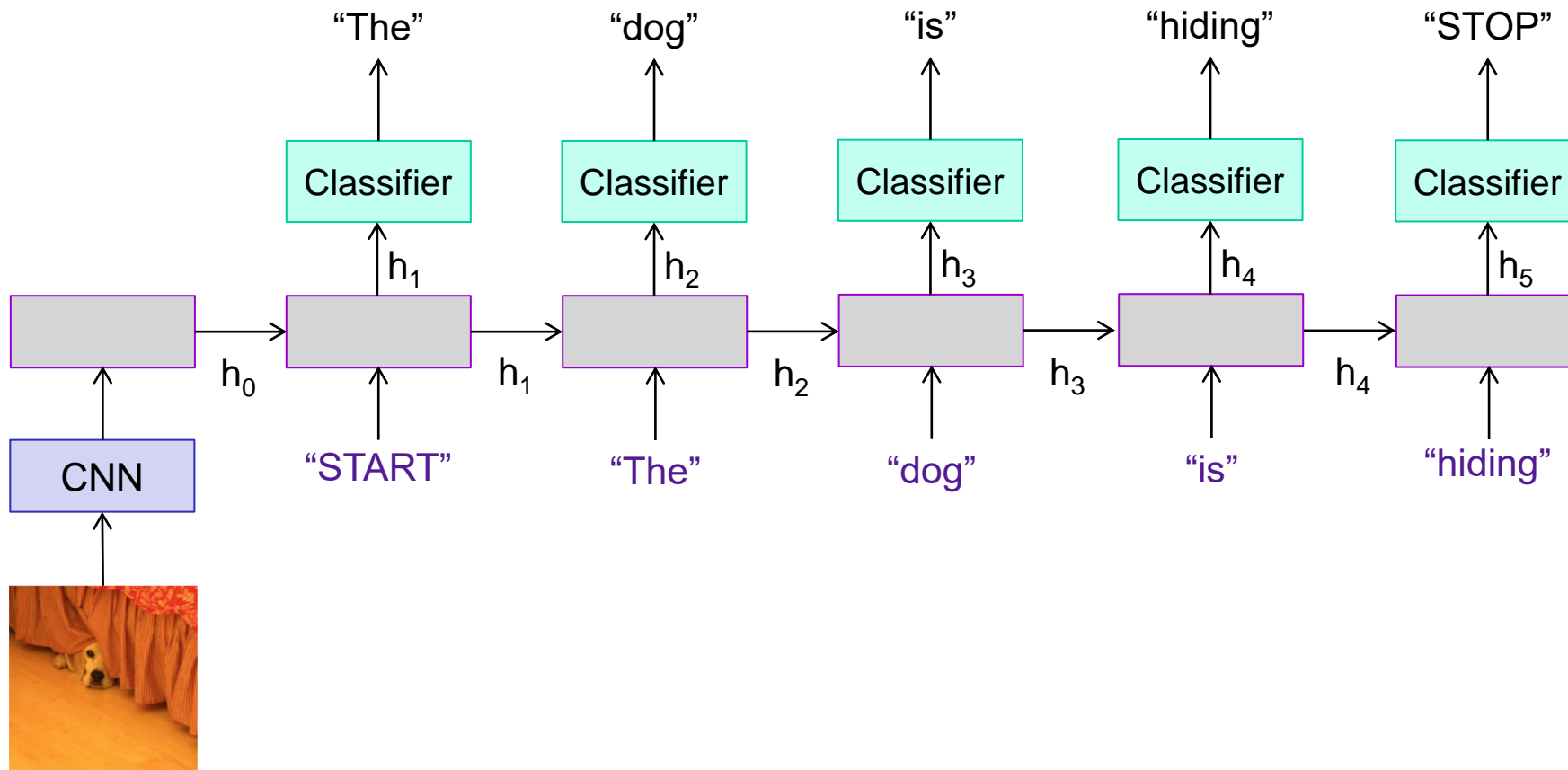


Two giraffes standing in a grassy field

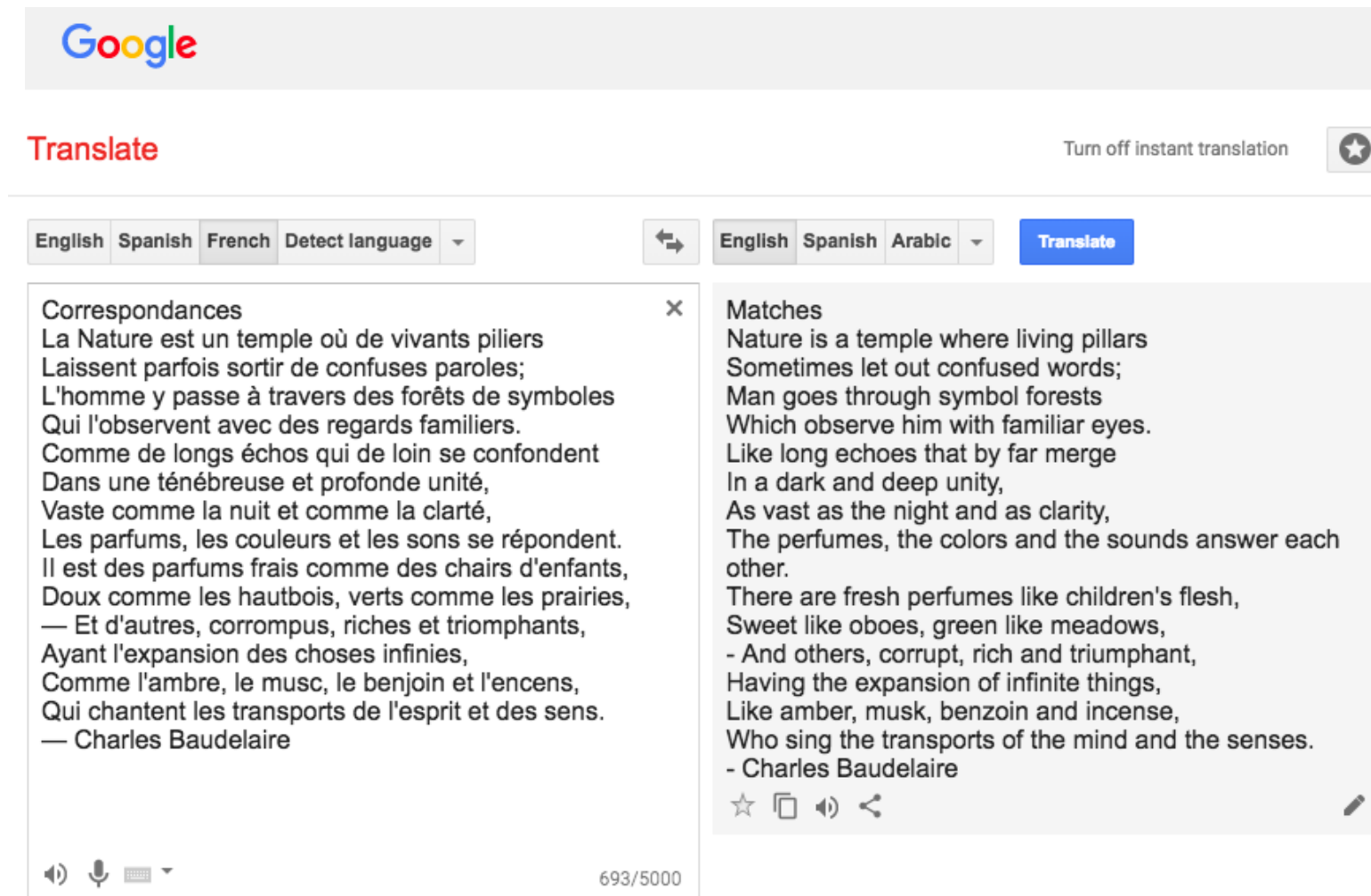


A man riding a dirt bike on a dirt track

Example 3: Image caption generation



Example 4: Machine translation



The screenshot shows the Google Translate web interface. At the top is the Google logo. Below it, the word "Translate" is in red. To the right, there is a link "Turn off instant translation" and a star icon. The main area has two language selection dropdowns: the first is set to "English" and "French", and the second is set to "English" and "Arabic". A blue "Translate" button is to the right of the second dropdown. Below the dropdowns, there are two text boxes. The left box contains the French text of a poem by Charles Baudelaire, titled "Correspondances". The right box contains the English translation of the same poem, titled "Matches". At the bottom of the interface, there are icons for voice input, a keyboard, and a page number "693/5000".

Google

Translate

Turn off instant translation

English Spanish French Detect language

English Spanish Arabic Translate

Correspondances

La Nature est un temple où de vivants piliers
Laissent parfois sortir de confuses paroles;
L'homme y passe à travers des forêts de symboles
Qui l'observent avec des regards familiers.
Comme de longs échos qui de loin se confondent
Dans une ténébreuse et profonde unité,
Vaste comme la nuit et comme la clarté,
Les parfums, les couleurs et les sons se répondent.
Il est des parfums frais comme des chairs d'enfants,
Doux comme les hautbois, verts comme les prairies,
— Et d'autres, corrompus, riches et triomphants,
Ayant l'expansion des choses infinies,
Comme l'ambre, le musc, le benjoin et l'encens,
Qui chantent les transports de l'esprit et des sens.
— Charles Baudelaire

Matches

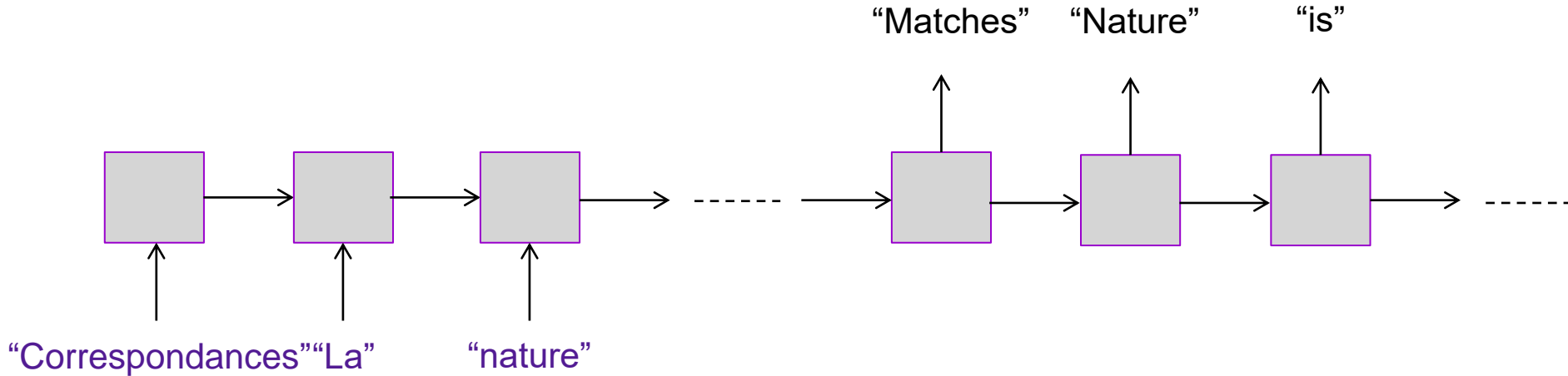
Nature is a temple where living pillars
Sometimes let out confused words;
Man goes through symbol forests
Which observe him with familiar eyes.
Like long echoes that by far merge
In a dark and deep unity,
As vast as the night and as clarity,
The perfumes, the colors and the sounds answer each other.
There are fresh perfumes like children's flesh,
Sweet like oboes, green like meadows,
- And others, corrupt, rich and triumphant,
Having the expansion of infinite things,
Like amber, musk, benzoin and incense,
Who sing the transports of the mind and the senses.
- Charles Baudelaire

693/5000

<https://translate.google.com/>

Example 4: Machine translation

- Multiple input – multiple output (or sequence to sequence) scenario:



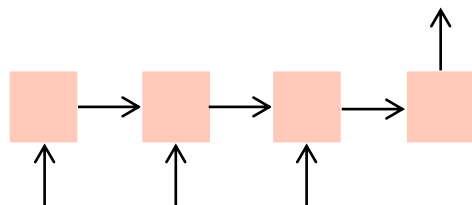
Summary: Input-output scenarios

Single -
Single



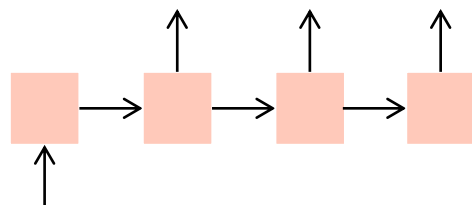
Feed-forward
Network

Multiple -
Single



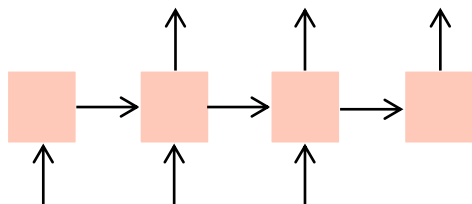
Sequence
Classification

Single -
Multiple



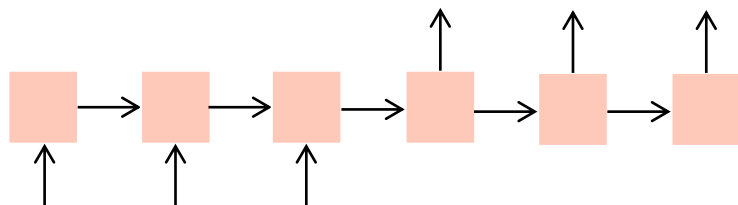
Sequence generation,
captioning

Multiple -
Multiple



Sequence generation,
captioning

Multiple -
Multiple

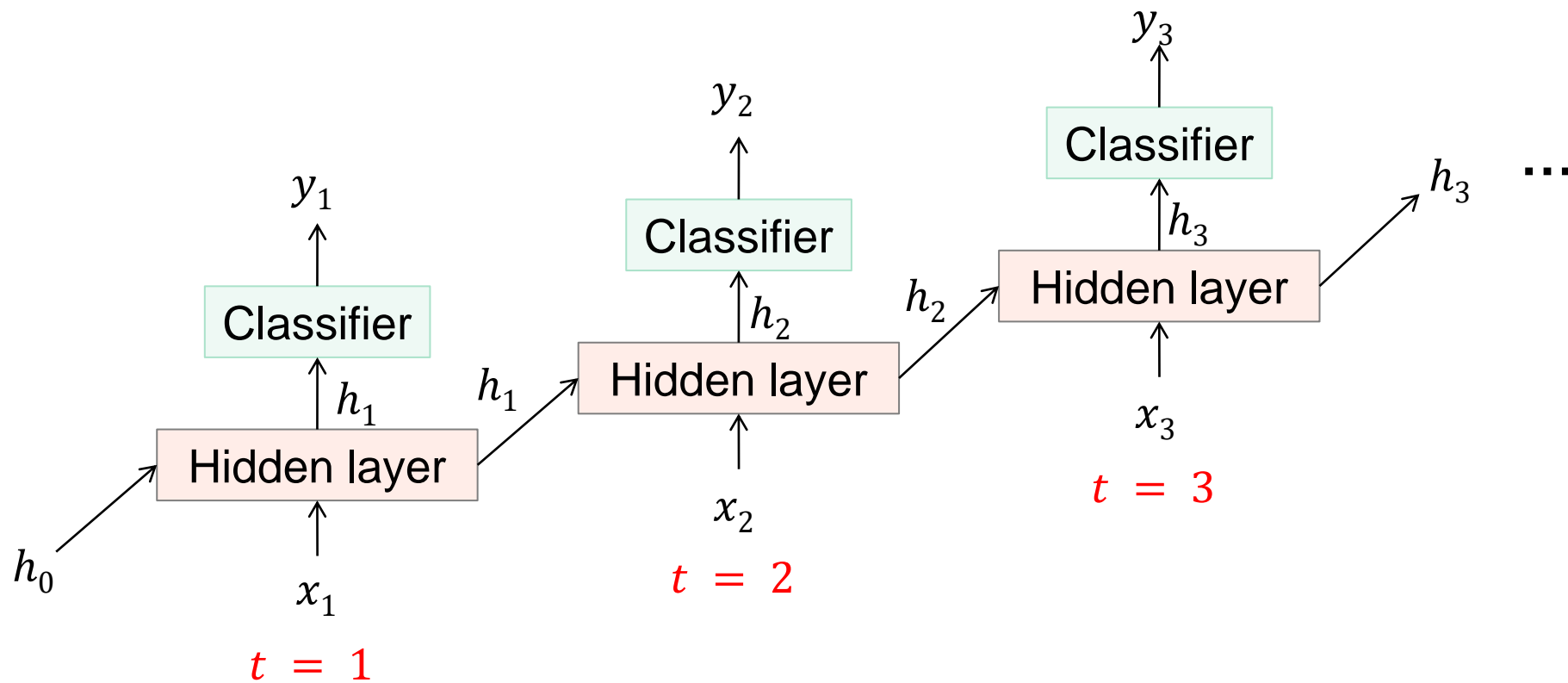


Translation

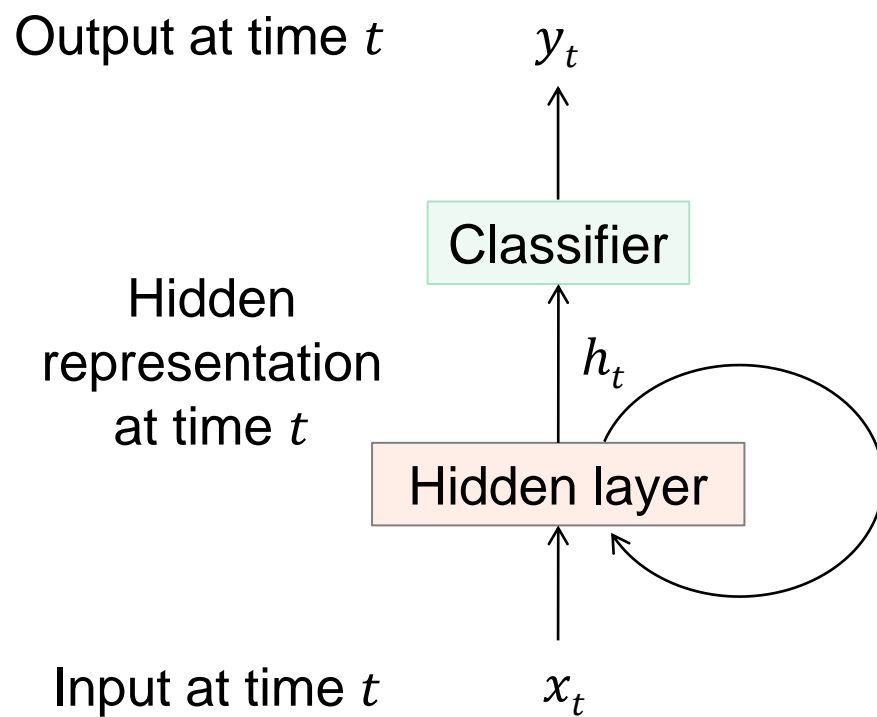
Outline

- Examples of sequential prediction tasks
- Common recurrent units
 - Vanilla RNN unit
 - Long Short-Term Memory (LSTM)
 - Gated Recurrent Unit (GRU)

Recurrent unit



Recurrent unit



Recurrence:

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

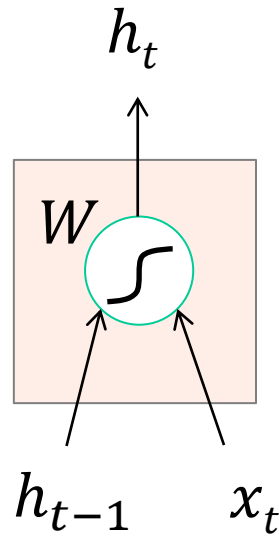
new
state

function
of W

input at
time t

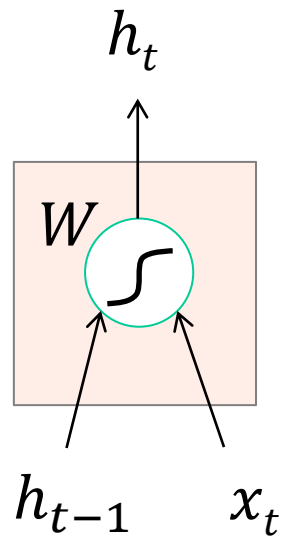
old state

Vanilla RNN cell

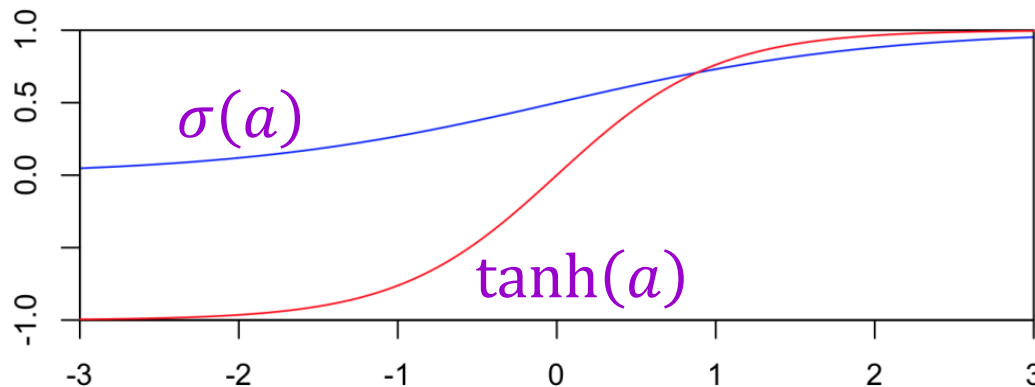


$$\begin{aligned} h_t &= f_W(x_t, h_{t-1}) \\ &= \tanh W \begin{pmatrix} x_t \\ h_{t-1} \end{pmatrix} \end{aligned}$$

Vanilla RNN cell

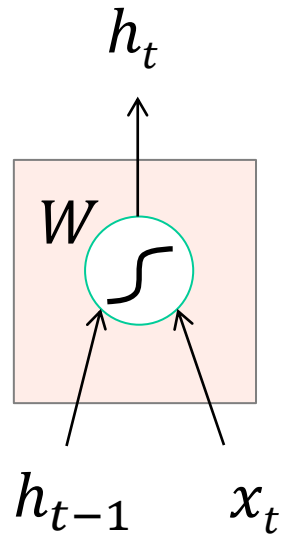


$$\begin{aligned} h_t &= f_W(x_t, h_{t-1}) \\ &= \tanh W \begin{pmatrix} x_t \\ h_{t-1} \end{pmatrix} \end{aligned}$$

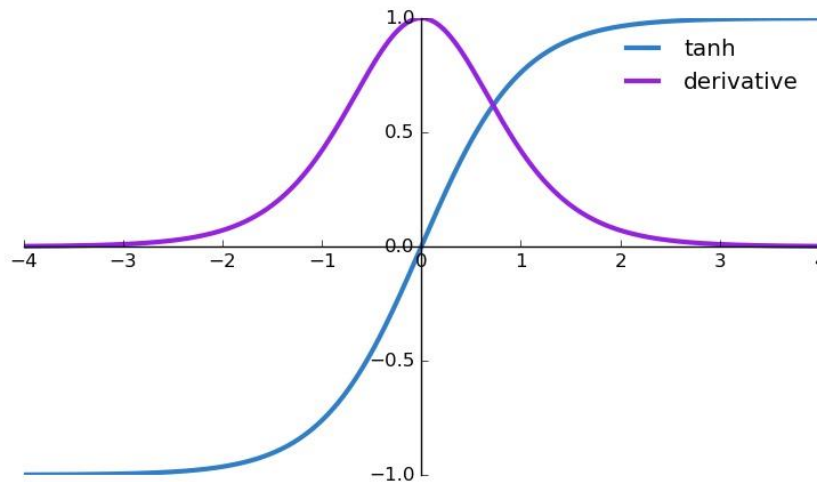


$$\begin{aligned} \tanh(a) &= \frac{e^a - e^{-a}}{e^a + e^{-a}} \\ &= 2\sigma(2a) - 1 \end{aligned}$$

Vanilla RNN cell

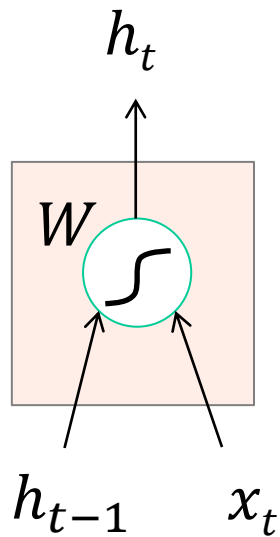


$$h_t = f_W(x_t, h_{t-1})$$
$$= \tanh W \begin{pmatrix} x_t \\ h_{t-1} \end{pmatrix}$$

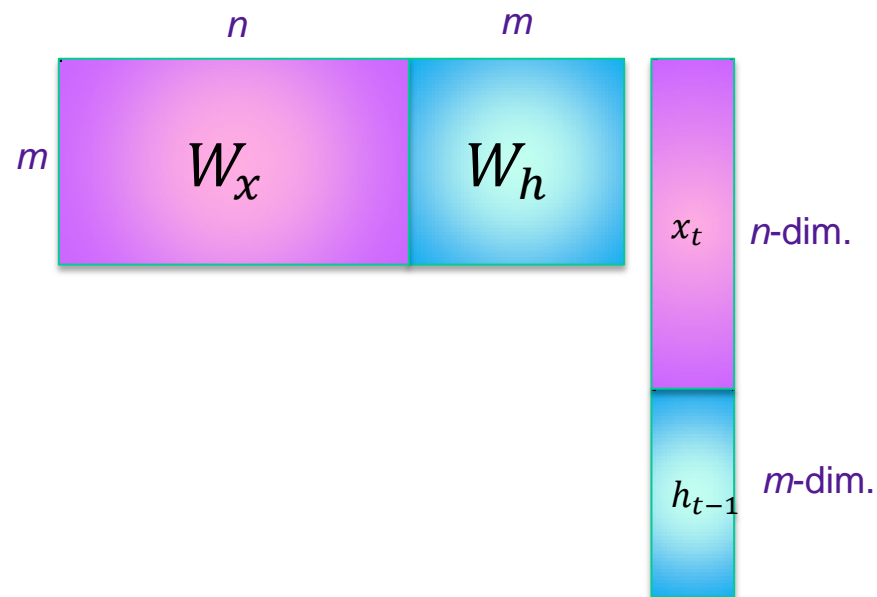


$$\frac{d}{da} \tanh(a) = 1 - \tanh^2(a)$$

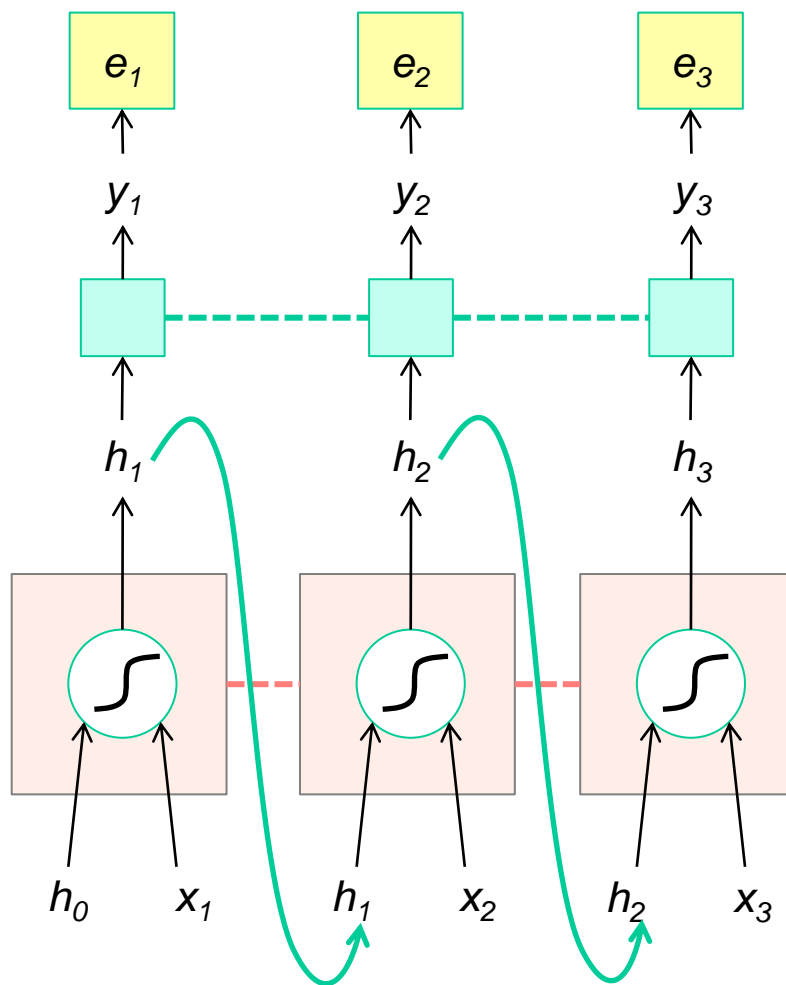
Vanilla RNN cell



$$\begin{aligned} h_t &= f_W(x_t, h_{t-1}) \\ &= \tanh W \begin{pmatrix} x_t \\ h_{t-1} \end{pmatrix} \\ &= \tanh(W_x x_t + W_h h_{t-1}) \end{aligned}$$



RNN forward pass



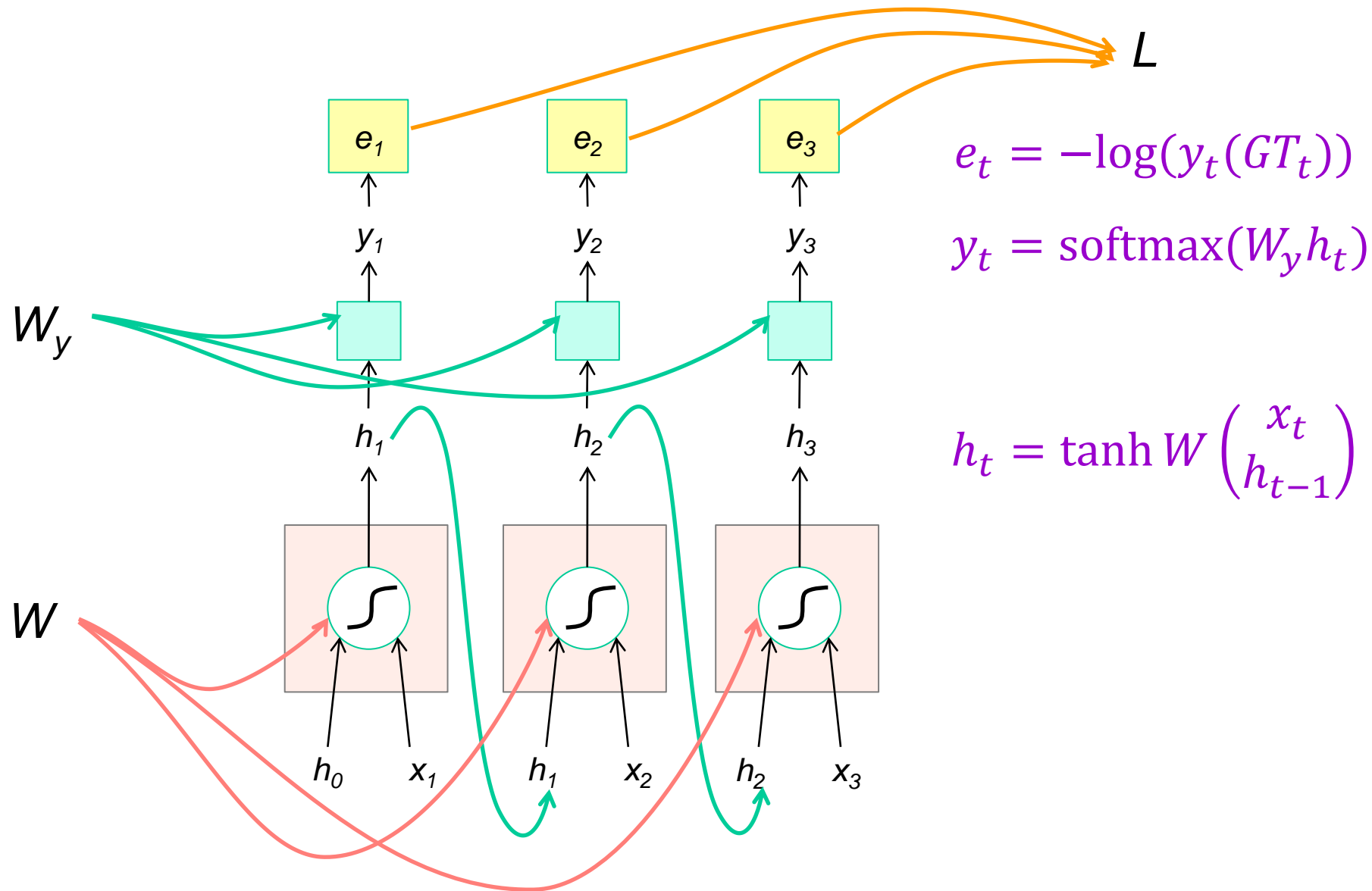
$$e_t = -\log(y_t(GT_t))$$

$$y_t = \text{softmax}(W_y h_t)$$

$$h_t = \tanh W \begin{pmatrix} x_t \\ h_{t-1} \end{pmatrix}$$

----- shared weights

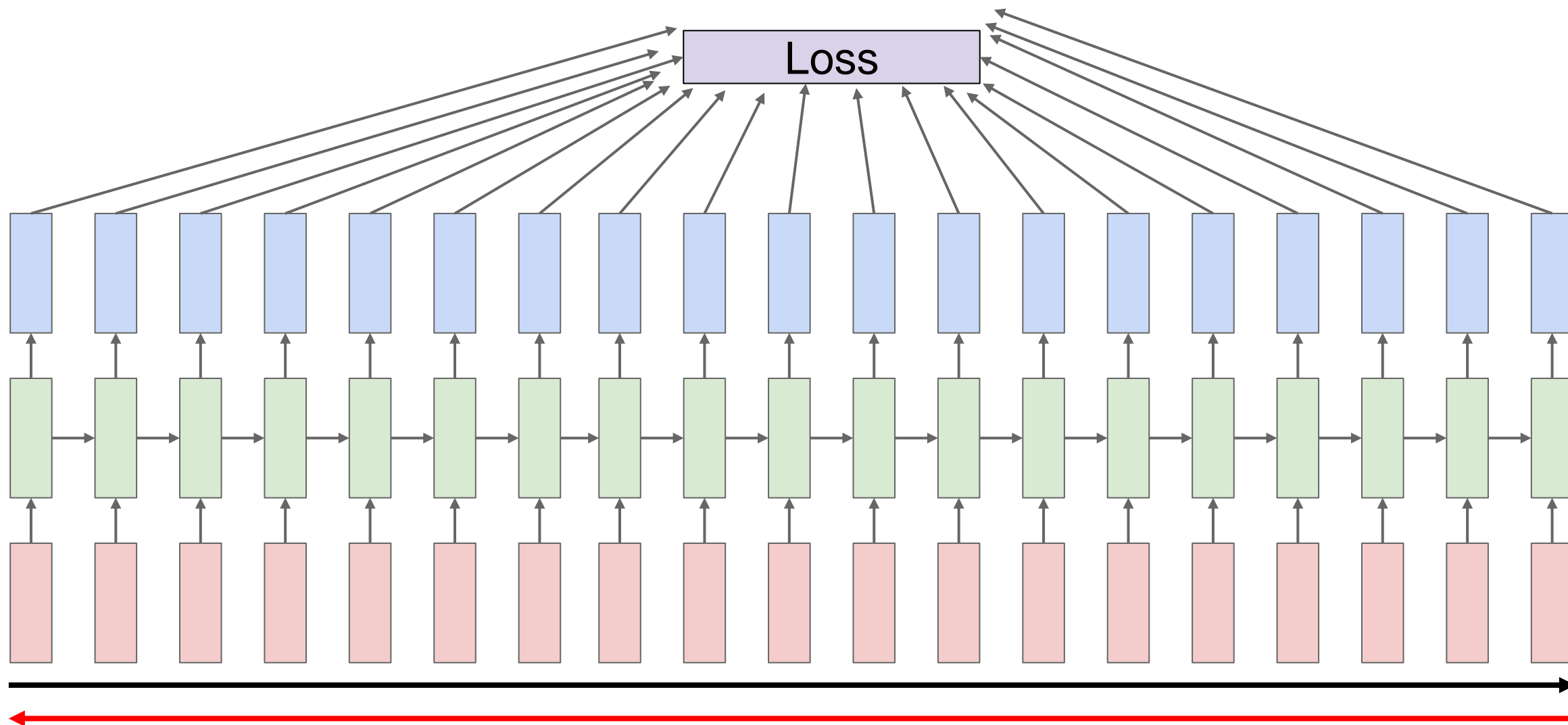
RNN forward pass: Computation graph



Training: Backpropagation through time (BPTT)

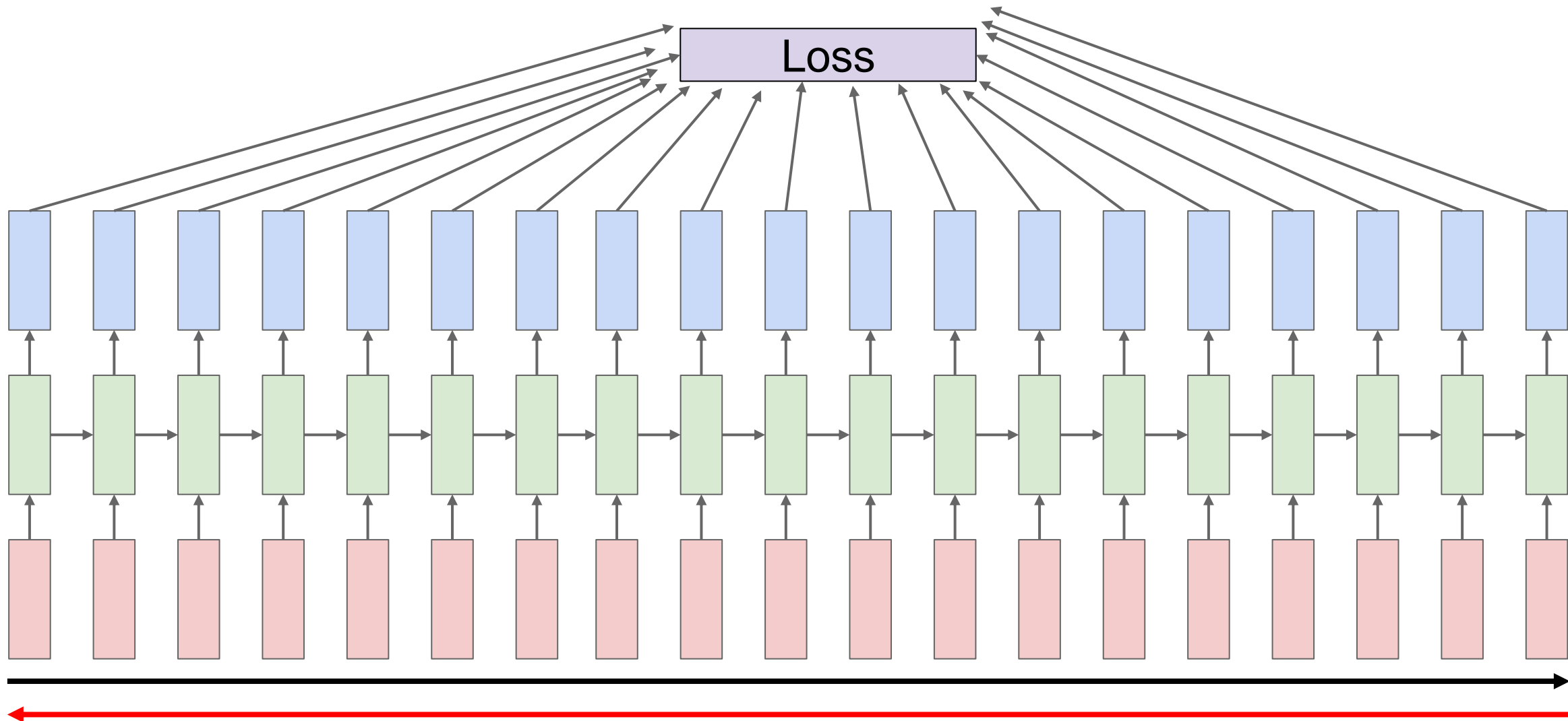
- The unfolded network (used during forward pass) is treated as one big feed-forward network that accepts the whole time series as input
- The weight updates are computed for each copy in the unfolded network, then summed (or averaged) and applied to the RNN weights

Backpropagation through time



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

Backpropagation through time

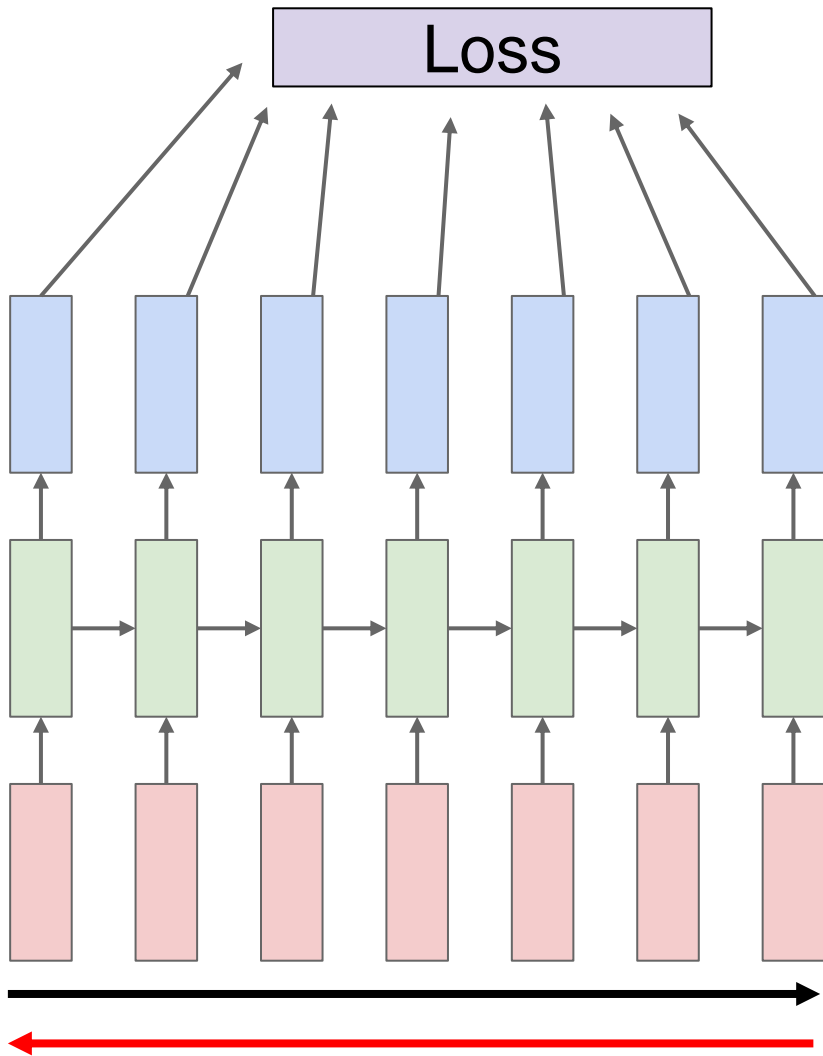


Problem: Takes a lot of memory for long sequences!

Training: Backpropagation through time (BPTT)

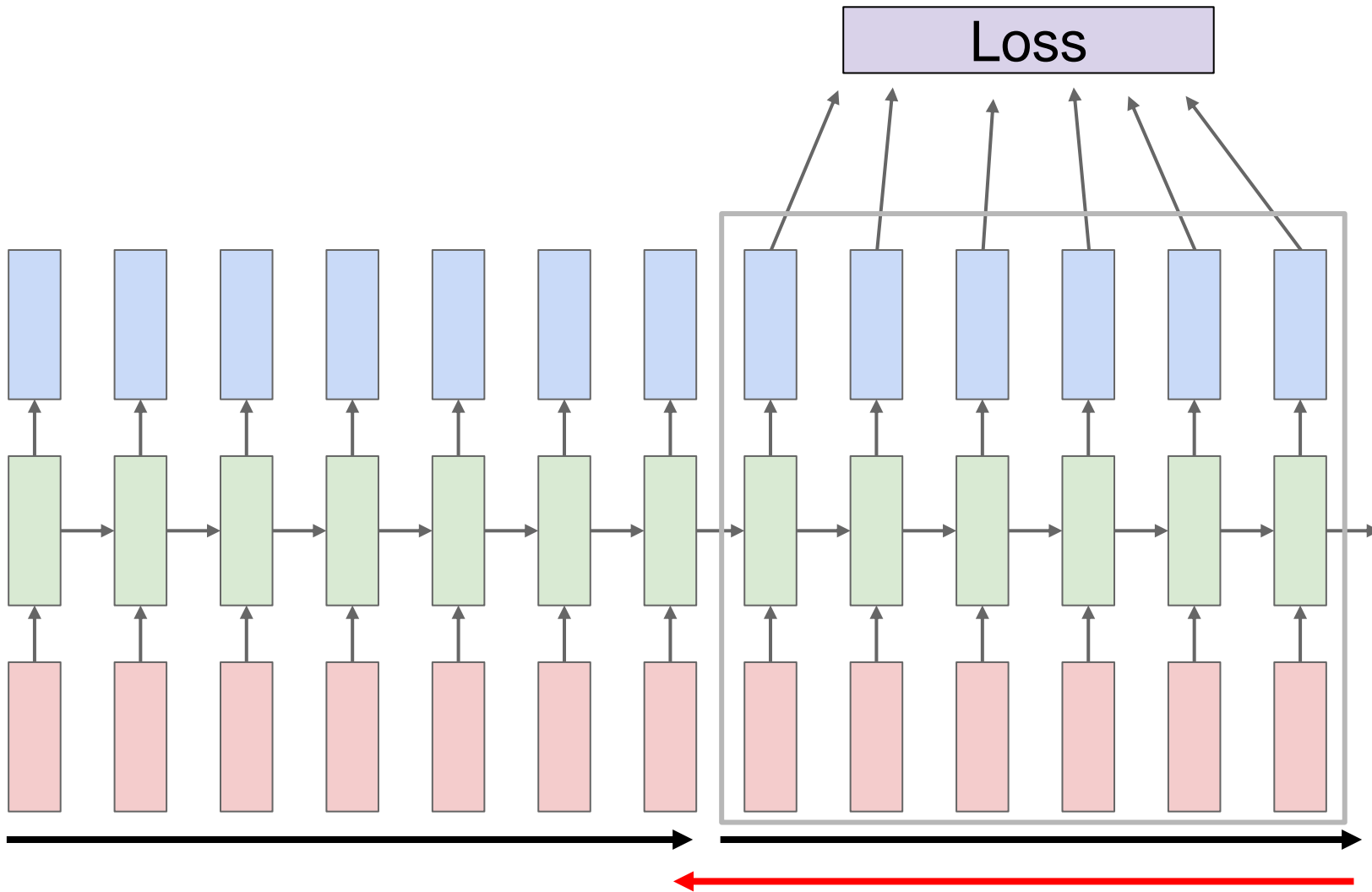
- The unfolded network (used during forward pass) is treated as one big feed-forward network that accepts the whole time series as input
- The weight updates are computed for each copy in the unfolded network, then summed (or averaged) and applied to the RNN weights
- In practice, *truncated* BPTT is used: run the RNN forward k_1 time steps, propagate backward for k_2 time steps

Truncated backpropagation through time



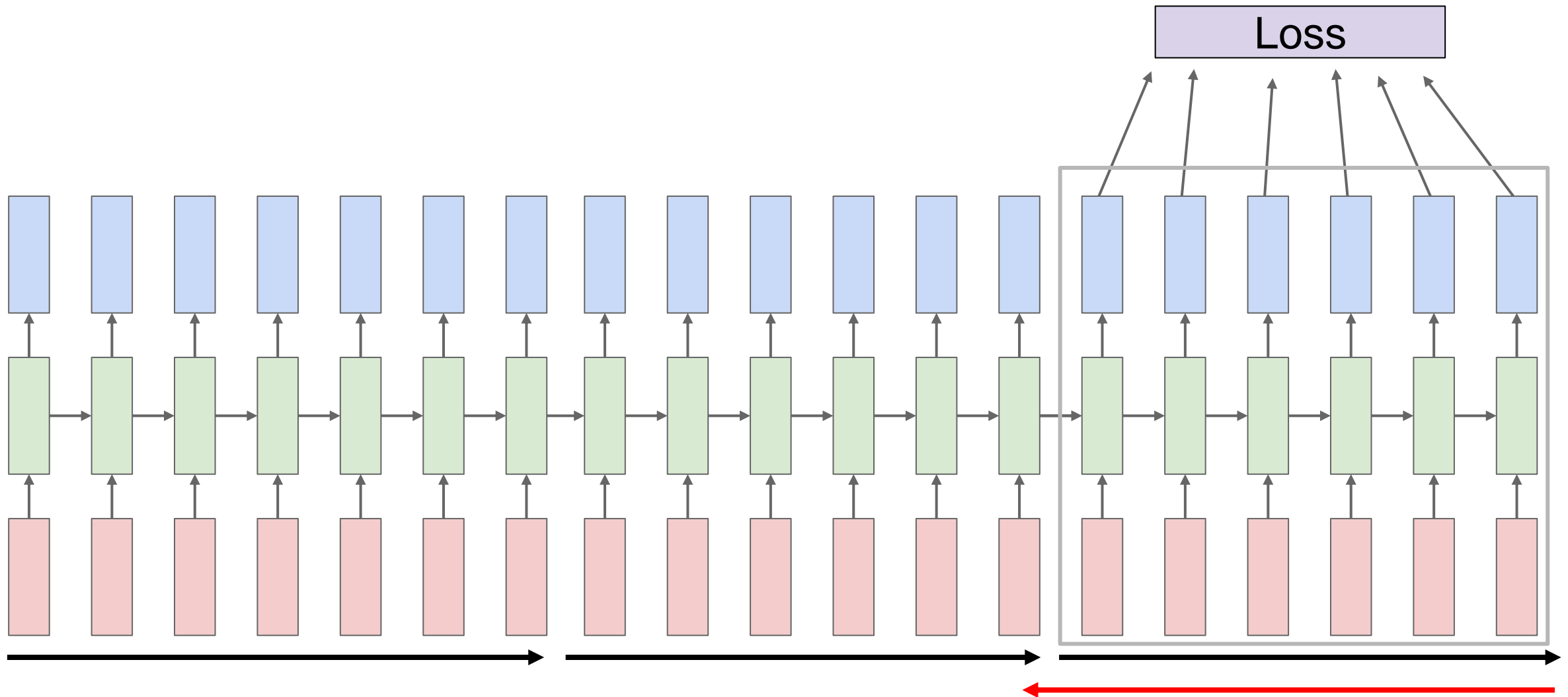
Run forward and backward through chunks of the sequence instead of whole sequence

Truncated backpropagation through time

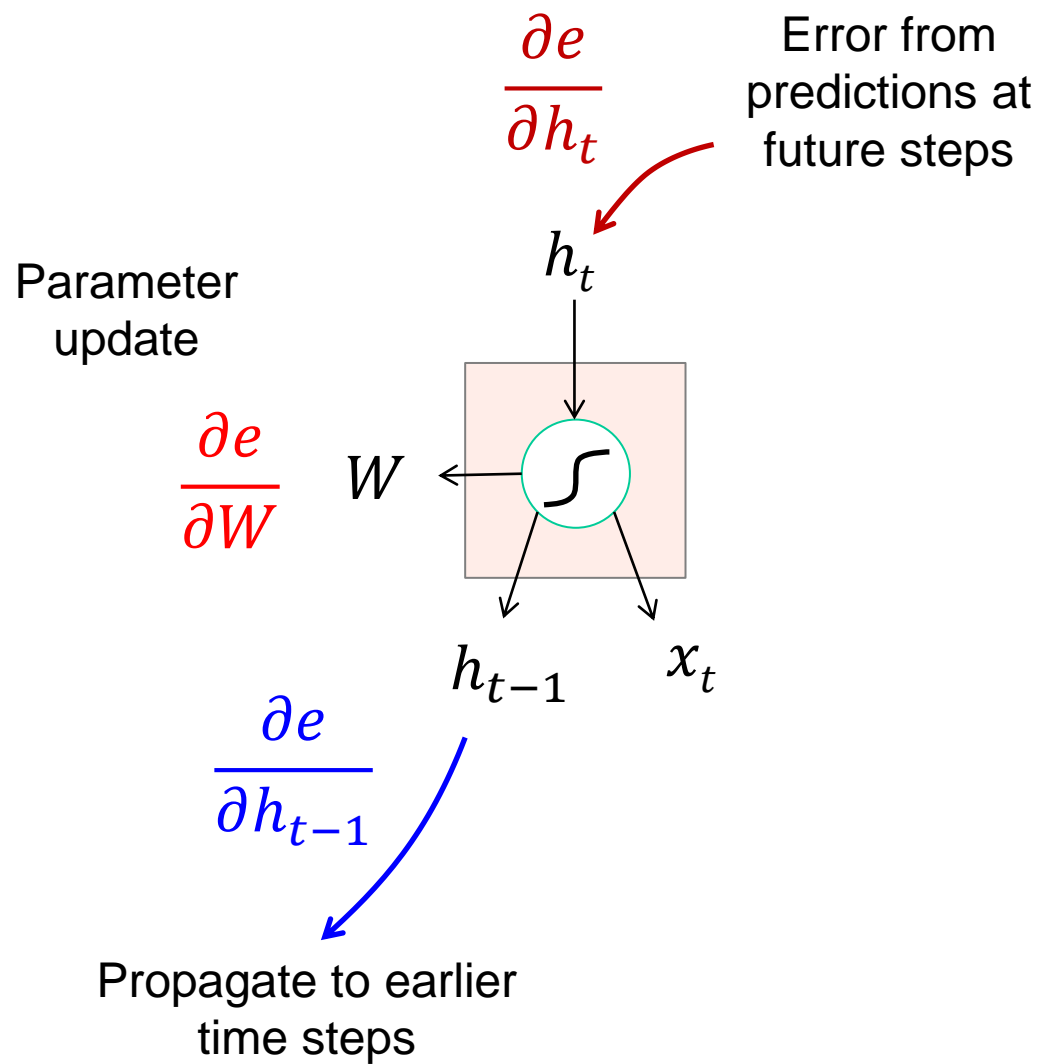


Carry hidden states forward in time further, but only backpropagate for some smaller number of steps

Truncated backpropagation through time



RNN backward pass



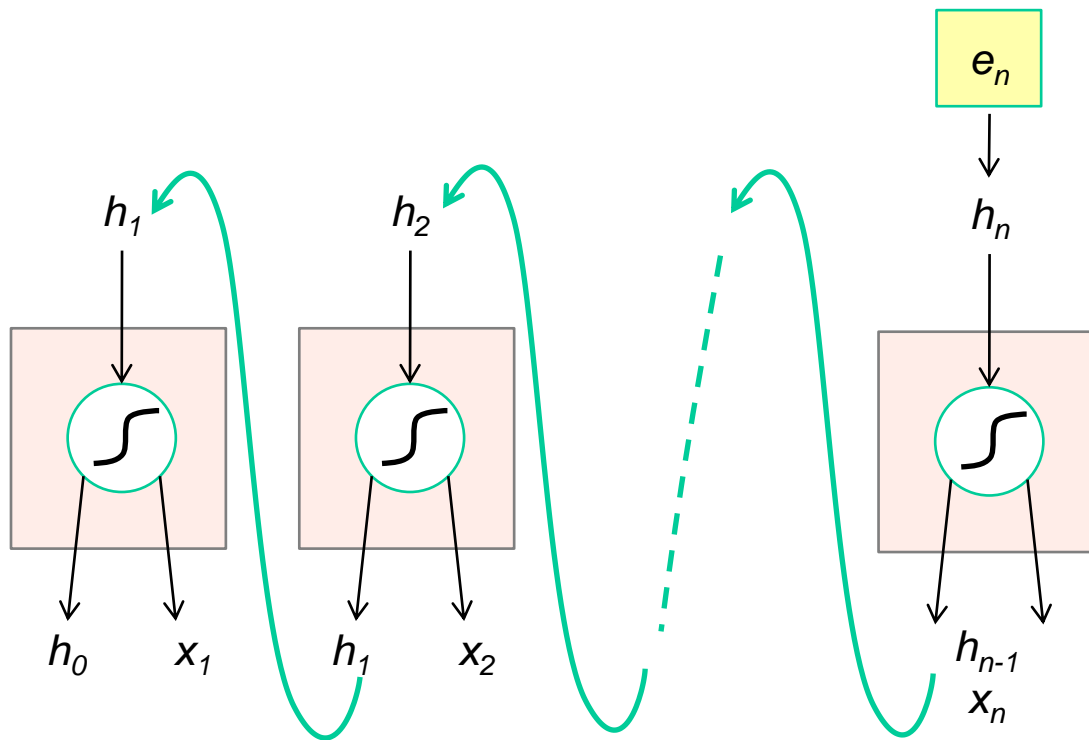
$$h_t = \tanh(W_x x_t + W_h h_{t-1})$$

$$\frac{\partial e}{\partial W_h} = \frac{\partial e}{\partial h_t} \odot (1 - \tanh^2(W_x x_t + W_h h_{t-1})) h_{t-1}^T$$

$$\frac{\partial e}{\partial W_x} = \frac{\partial e}{\partial h_t} \odot (1 - \tanh^2(W_x x_t + W_h h_{t-1})) x_t^T$$

$$\frac{\partial e}{\partial h_{t-1}} = W_h^T (1 - \tanh^2(W_x x_t + W_h h_{t-1})) \odot \frac{\partial e}{\partial h_t}$$

Vanishing and exploding gradients



$$\frac{\partial e}{\partial h_{t-1}} = W_h^T (1 - \tanh^2(W_x x_t + W_h h_{t-1})) \odot \frac{\partial e}{\partial h_t}$$

Computing gradient for h_0 involves many multiplications by W_h^T (and rescalings between 0 and 1)

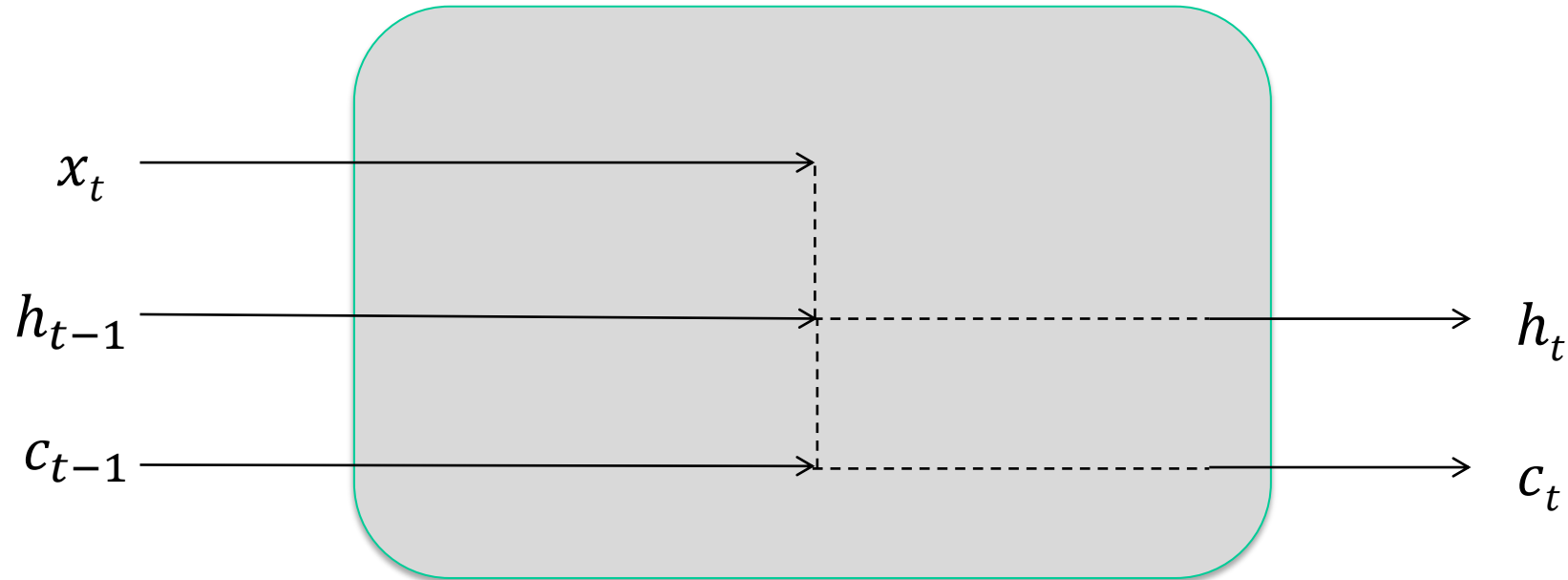
Gradients will *vanish* if largest singular value of W_h is less than 1 and *explode* if it's greater than 1

Outline

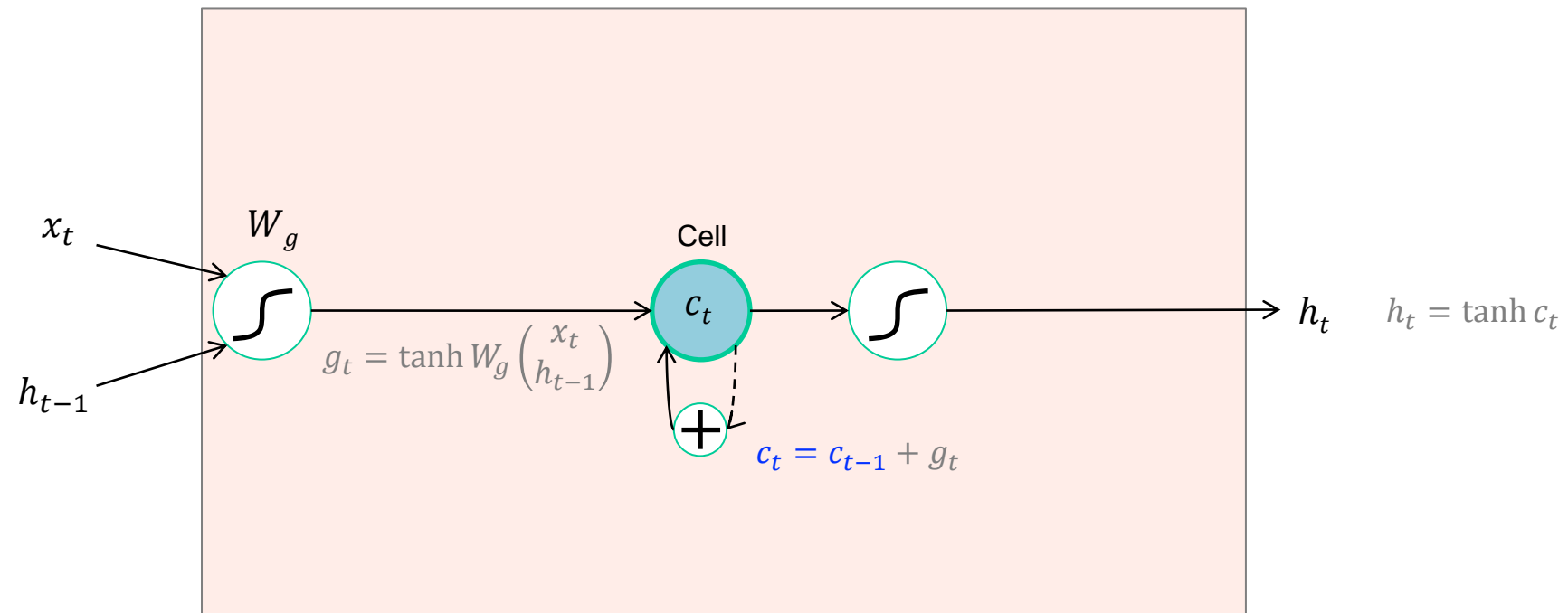
- Examples of sequential prediction tasks
- Common recurrent units
 - Vanilla RNN unit (and how to train it)
 - Long Short-Term Memory (LSTM)
 - Gated Recurrent Unit (GRU)

Long short-term memory (LSTM)

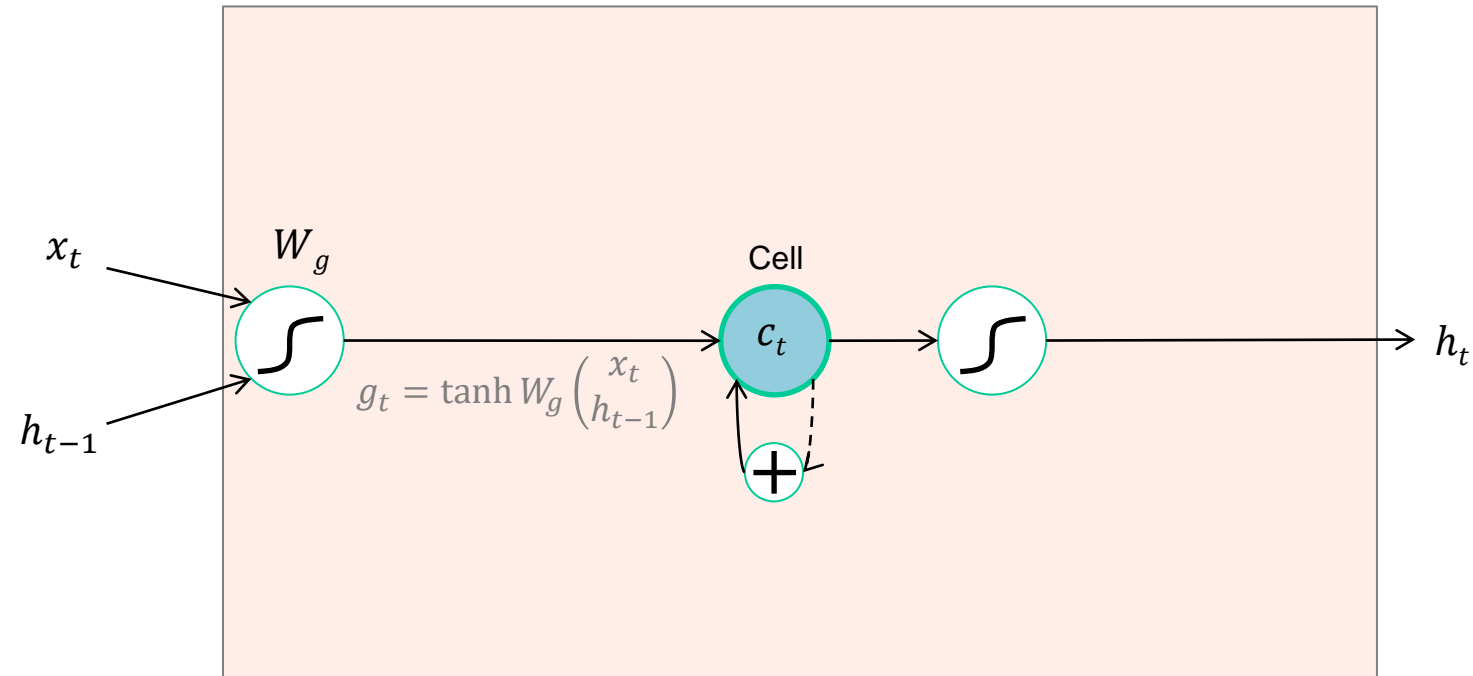
- Add a *memory cell* that is not subject to matrix multiplication or squashing, thereby avoiding gradient decay



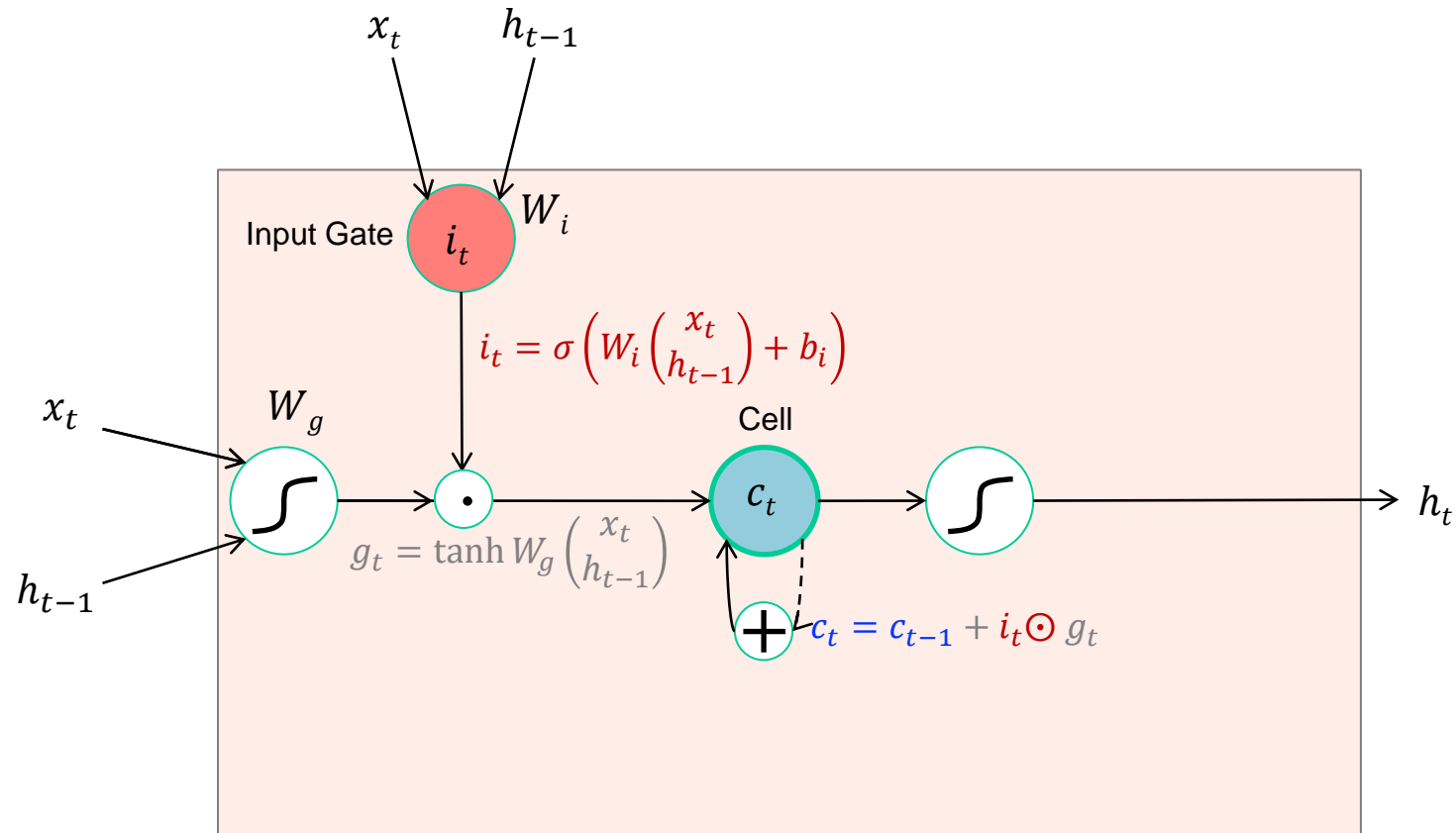
The LSTM cell



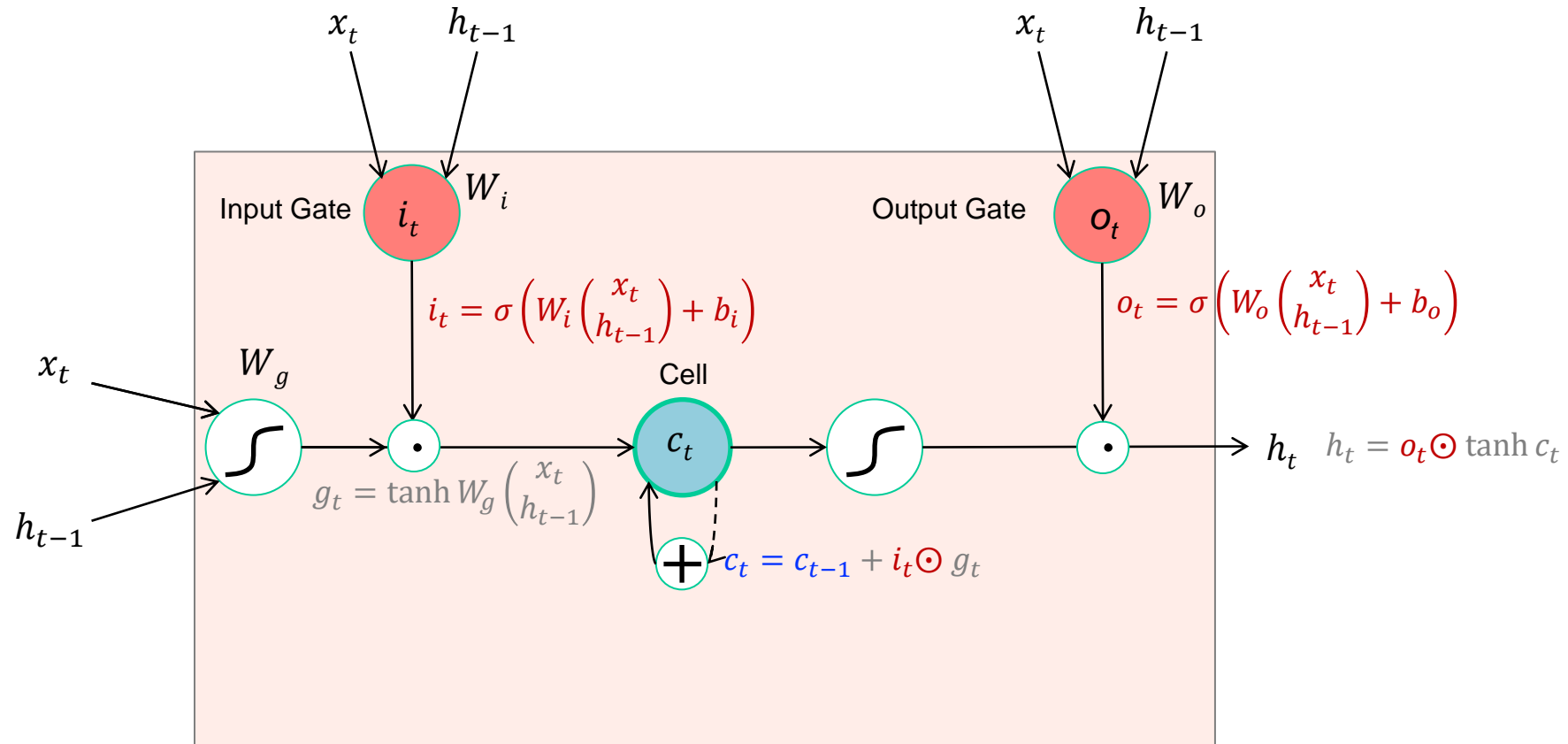
The LSTM cell



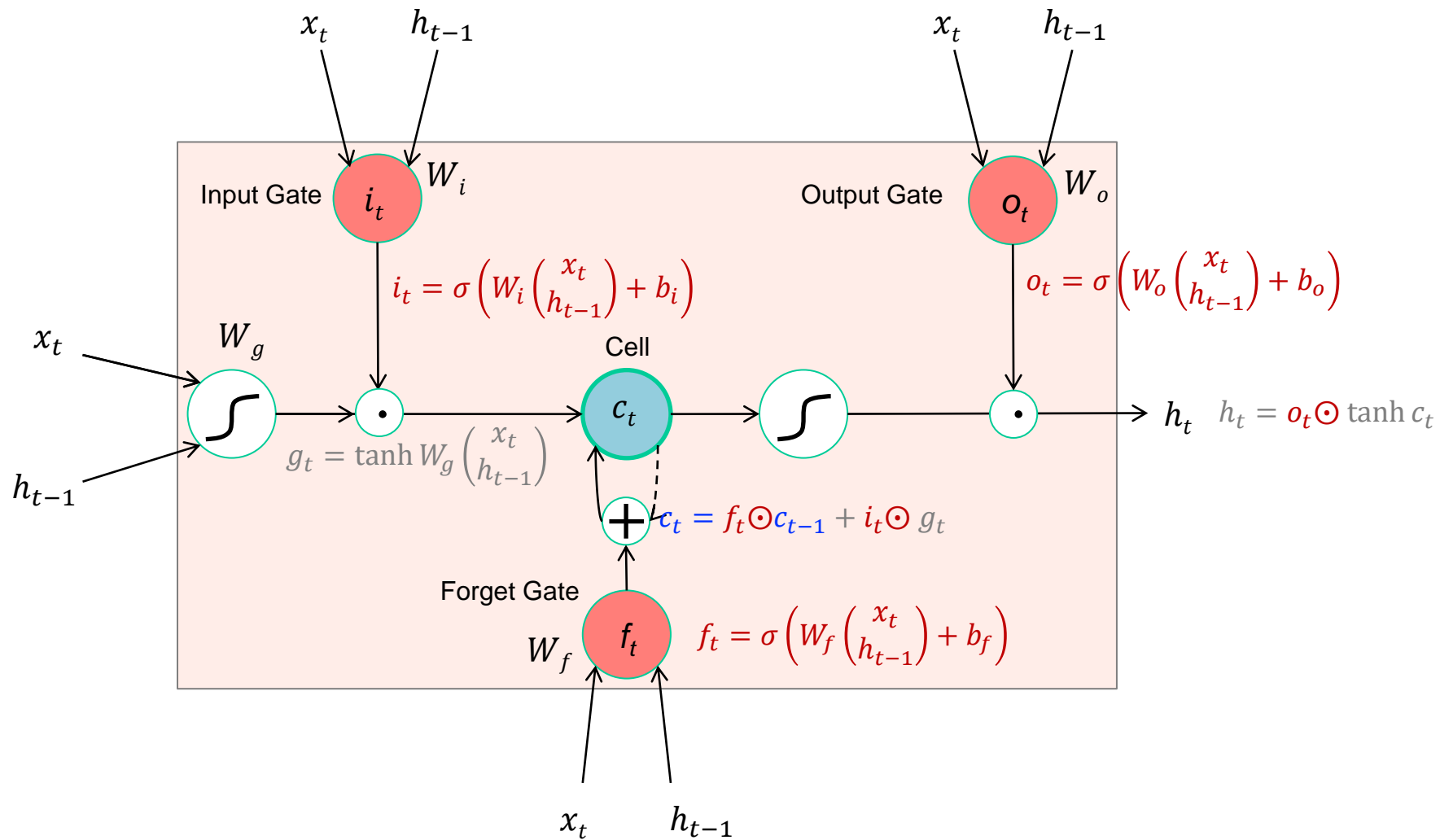
The LSTM cell



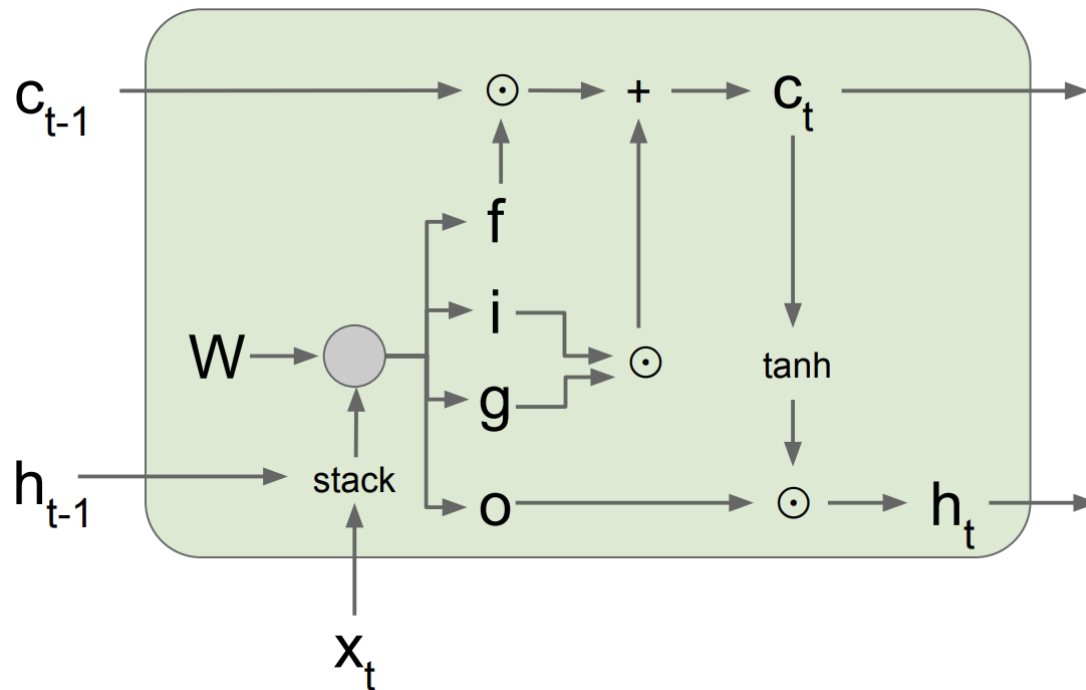
The LSTM cell



The LSTM cell



LSTM forward pass summary

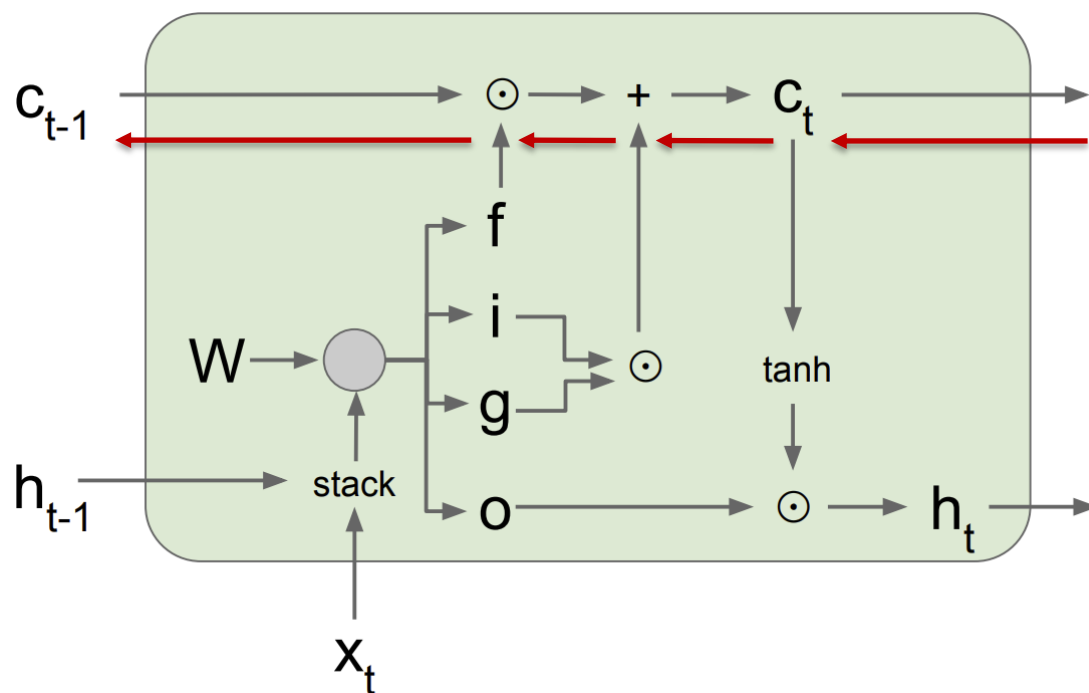


$$\begin{pmatrix} g_t \\ i_t \\ f_t \\ o_t \end{pmatrix} = \begin{pmatrix} \tanh \\ \sigma \\ \sigma \\ \sigma \end{pmatrix} \begin{pmatrix} W_g \\ W_i \\ W_f \\ W_o \end{pmatrix} \begin{pmatrix} x_t \\ h_{t-1} \end{pmatrix}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t$$

$$h_t = o_t \odot \tanh c_t$$

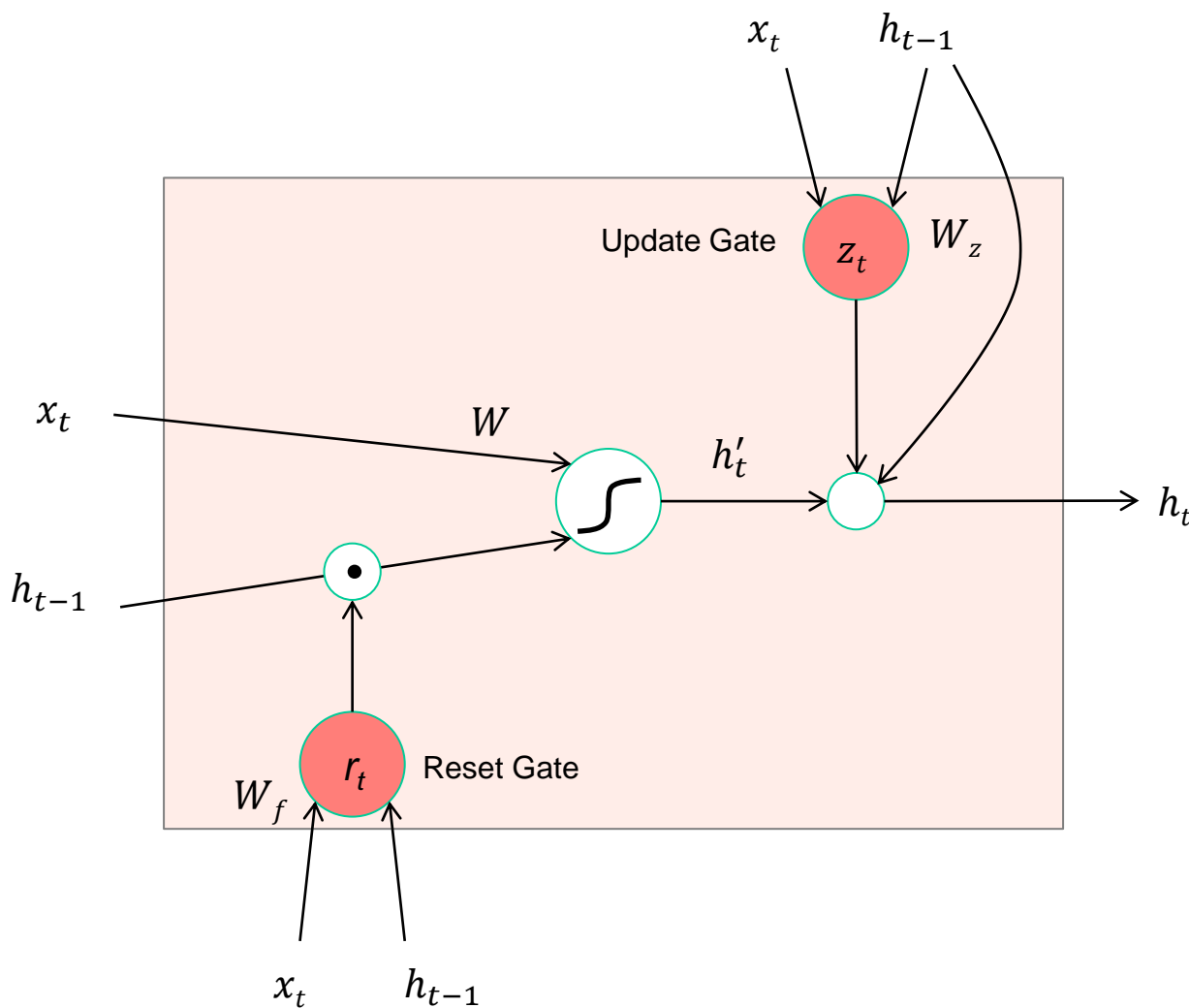
LSTM backward pass



Gradient flow from c_t to c_{t-1} only involves back-propagating through addition and elementwise multiplication, not matrix multiplication or tanh

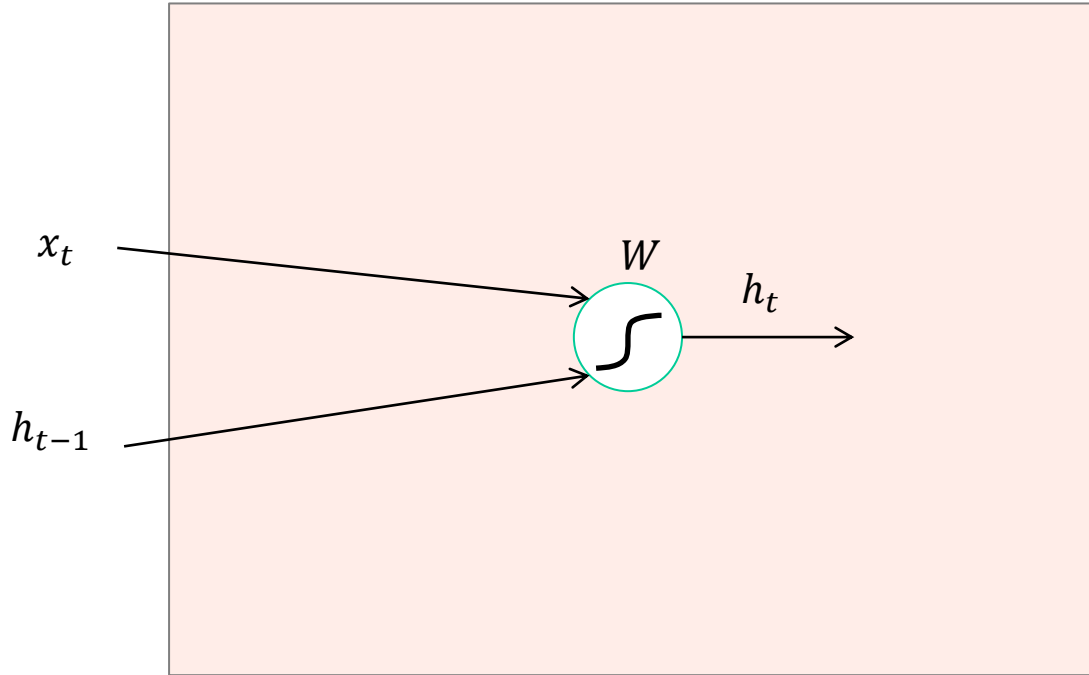
For complete details: [Illustrated LSTM Forward and Backward Pass](#)

Gated recurrent unit (GRU)



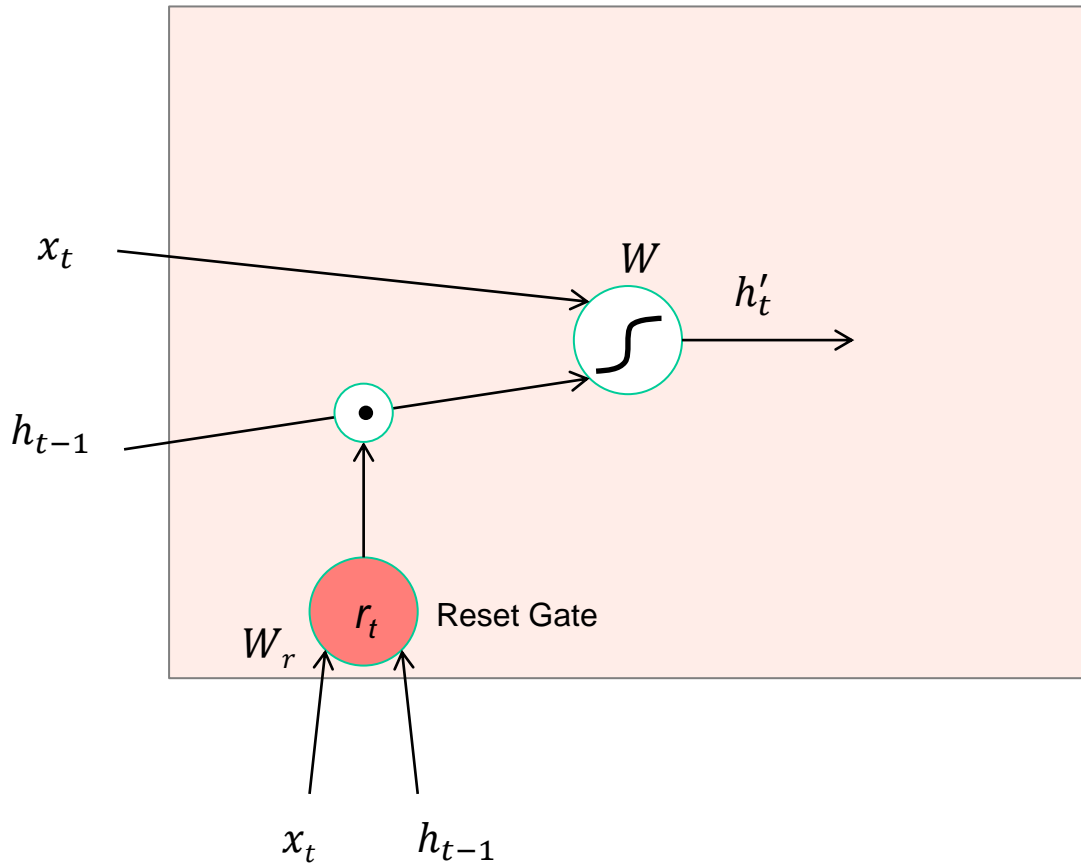
- Get rid of separate cell state
- Merge “forget” and “output” gates into “update” gate

Gated recurrent unit (GRU)



$$h_t = \tanh W \begin{pmatrix} x_t \\ h_{t-1} \end{pmatrix}$$

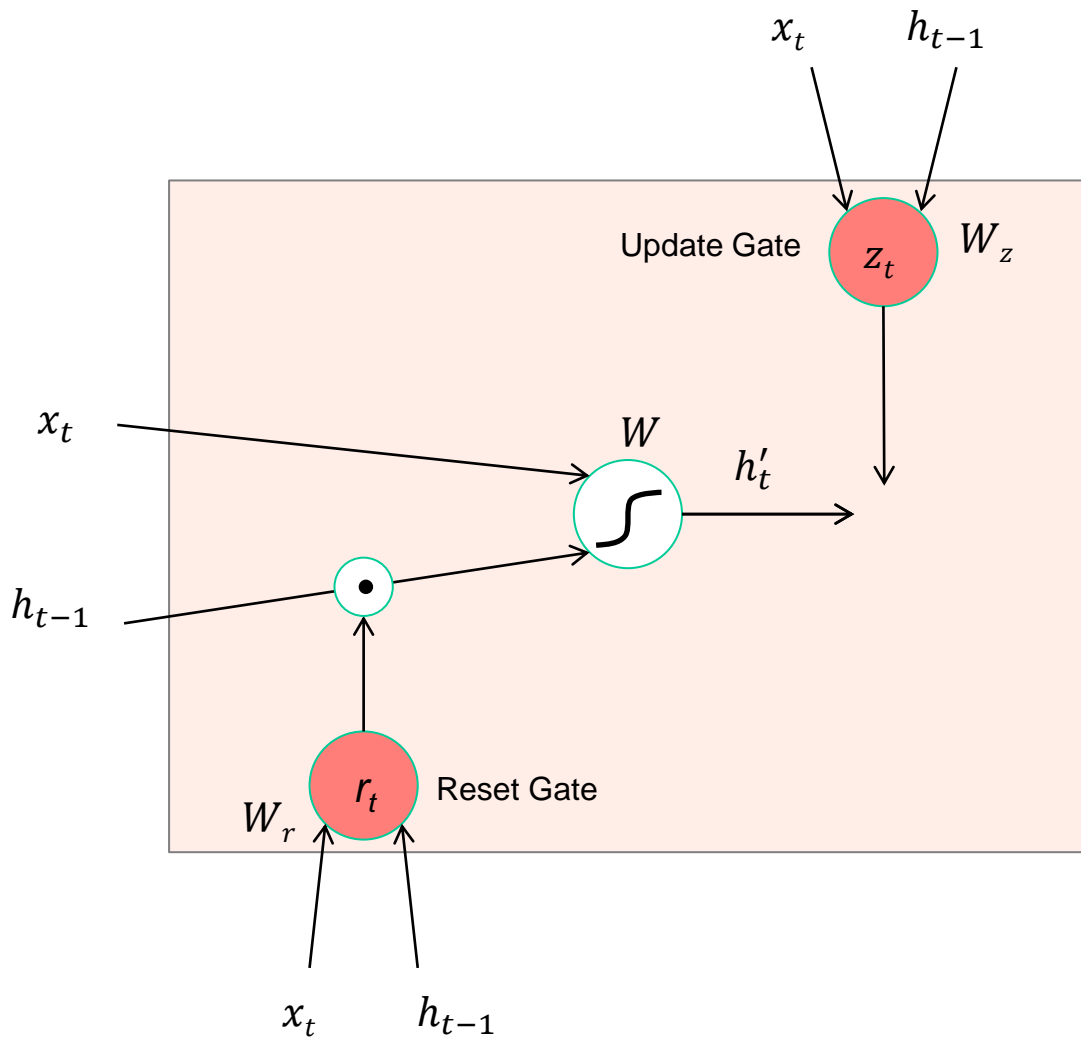
Gated recurrent unit (GRU)



$$r_t = \sigma \left(W_r \begin{pmatrix} x_t \\ h_{t-1} \end{pmatrix} + b_r \right)$$

$$h'_t = \tanh W \left(r_t \odot h_{t-1} \right)$$

Gated recurrent unit (GRU)

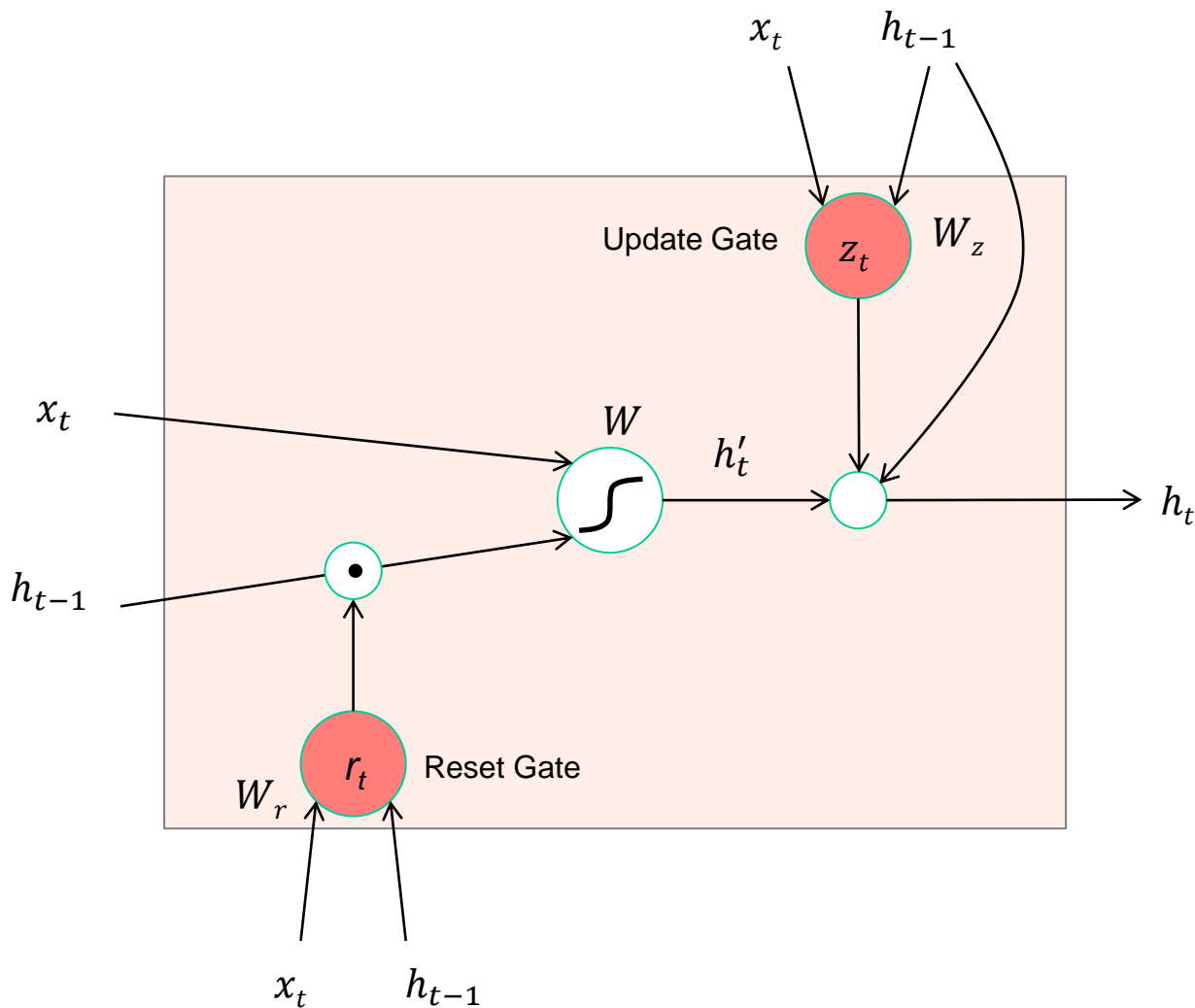


$$r_t = \sigma \left(W_r \begin{pmatrix} x_t \\ h_{t-1} \end{pmatrix} + b_r \right)$$

$$h'_t = \tanh W \left(r_t \odot h_{t-1} \oplus x_t \right)$$

$$z_t = \sigma \left(W_z \begin{pmatrix} x_t \\ h_{t-1} \end{pmatrix} + b_z \right)$$

Gated recurrent unit (GRU)



$$r_t = \sigma \left(W_r \begin{pmatrix} x_t \\ h_{t-1} \end{pmatrix} + b_r \right)$$

$$h'_t = \tanh W \left(r_t \odot h_{t-1} \parallel x_t \right)$$

$$z_t = \sigma \left(W_z \begin{pmatrix} x_t \\ h_{t-1} \end{pmatrix} + b_z \right)$$

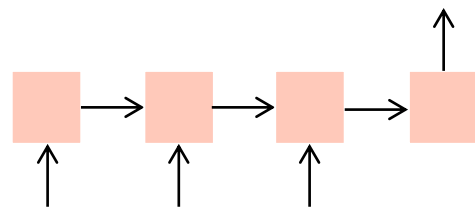
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot h'_t$$

Outline

- Examples of sequential prediction tasks
- Common recurrent units
 - Vanilla RNN unit (and how to train it)
 - Long Short-Term Memory (LSTM)
 - Gated Recurrent Unit (GRU)
- Recurrent network architectures

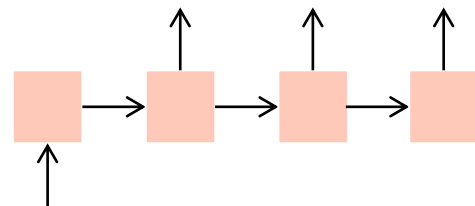
Summary: Input-output scenarios

Multiple -
Single



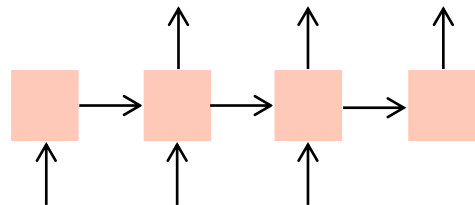
Sequence
Classification

Single -
Multiple



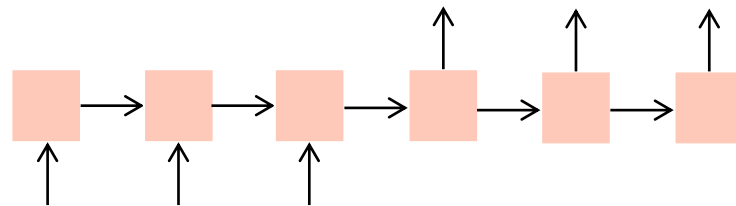
Sequence generation,
captioning

Multiple -
Multiple



Sequence generation,
captioning

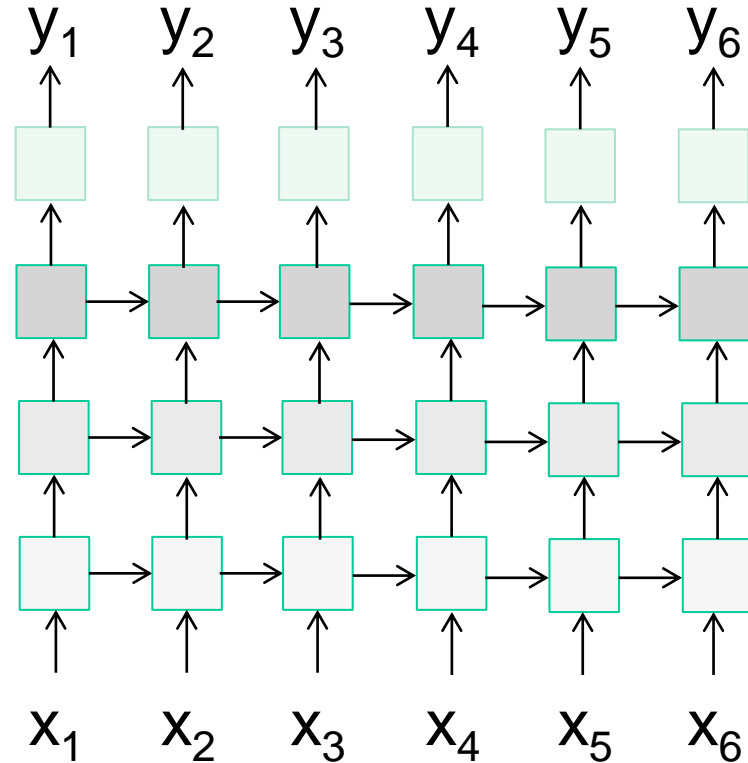
Multiple -
Multiple



Translation

Multi-layer RNNs

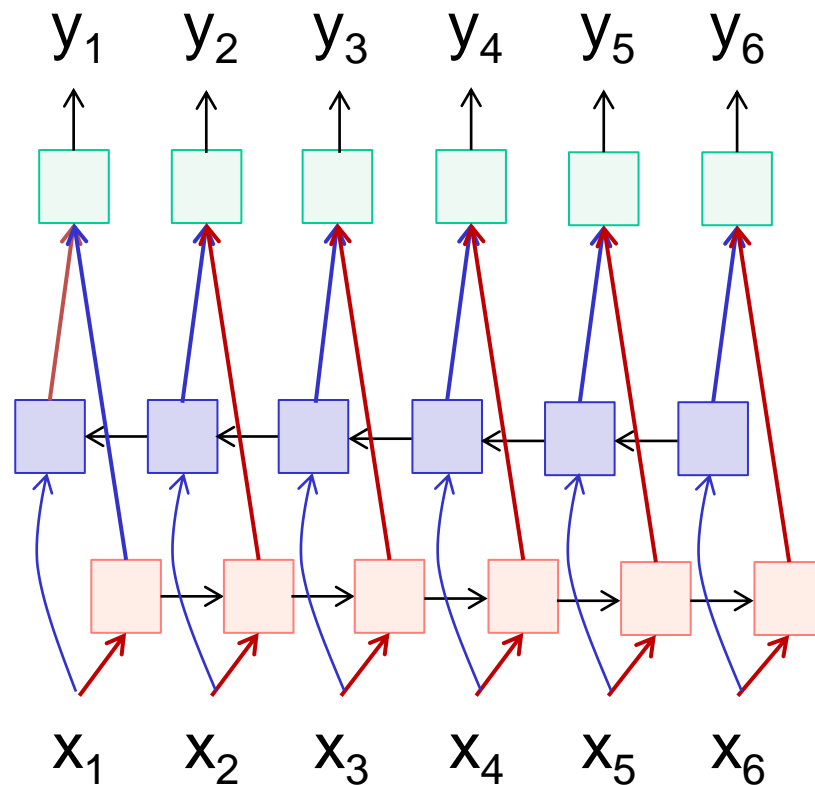
- We can of course design RNNs with multiple hidden layers



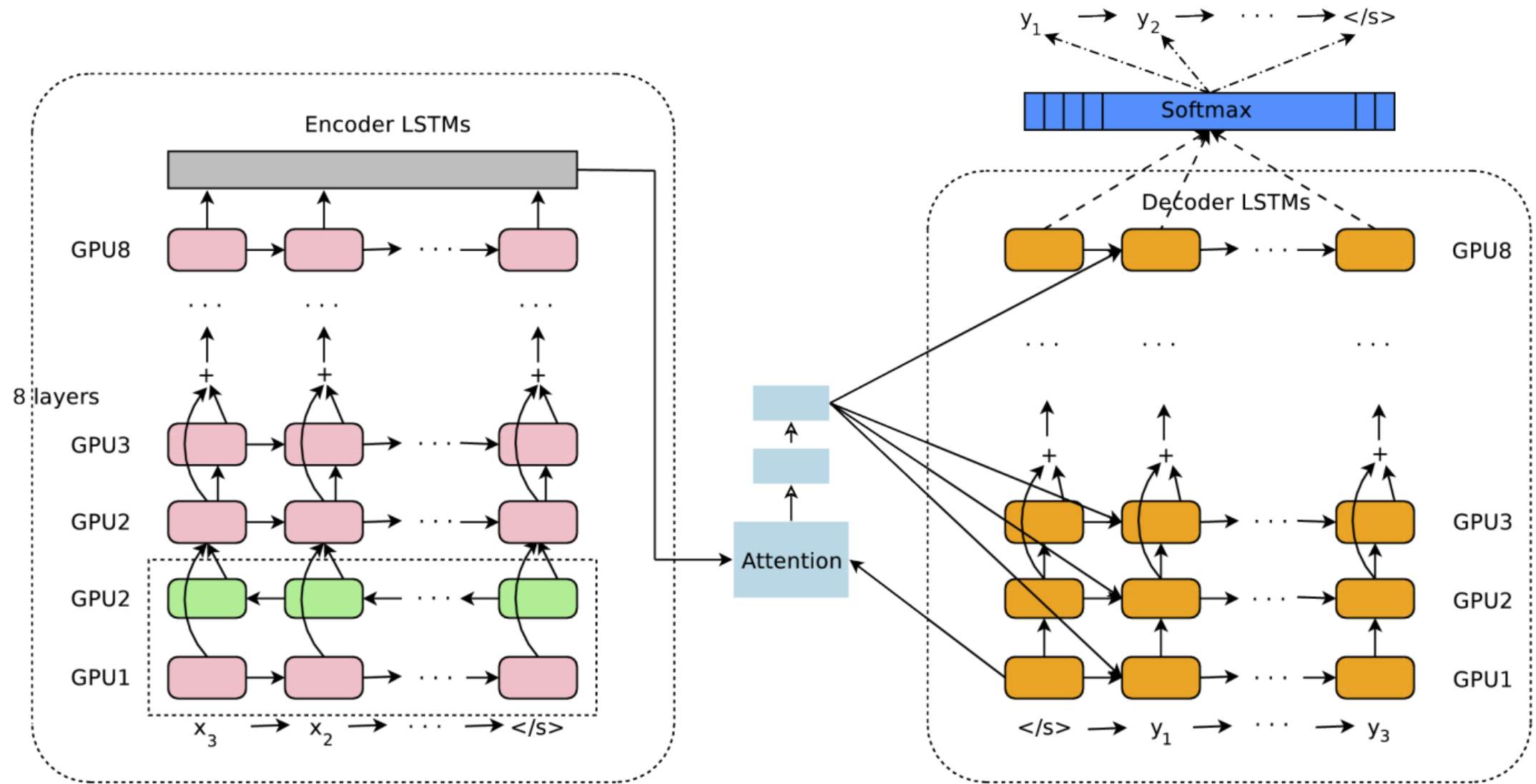
- Anything goes: skip connections across layers, across time, ...

Bi-directional RNNs

- RNNs can process the input sequence in forward and in the reverse direction (common in speech recognition)



Google Neural Machine Translation (GNMT)

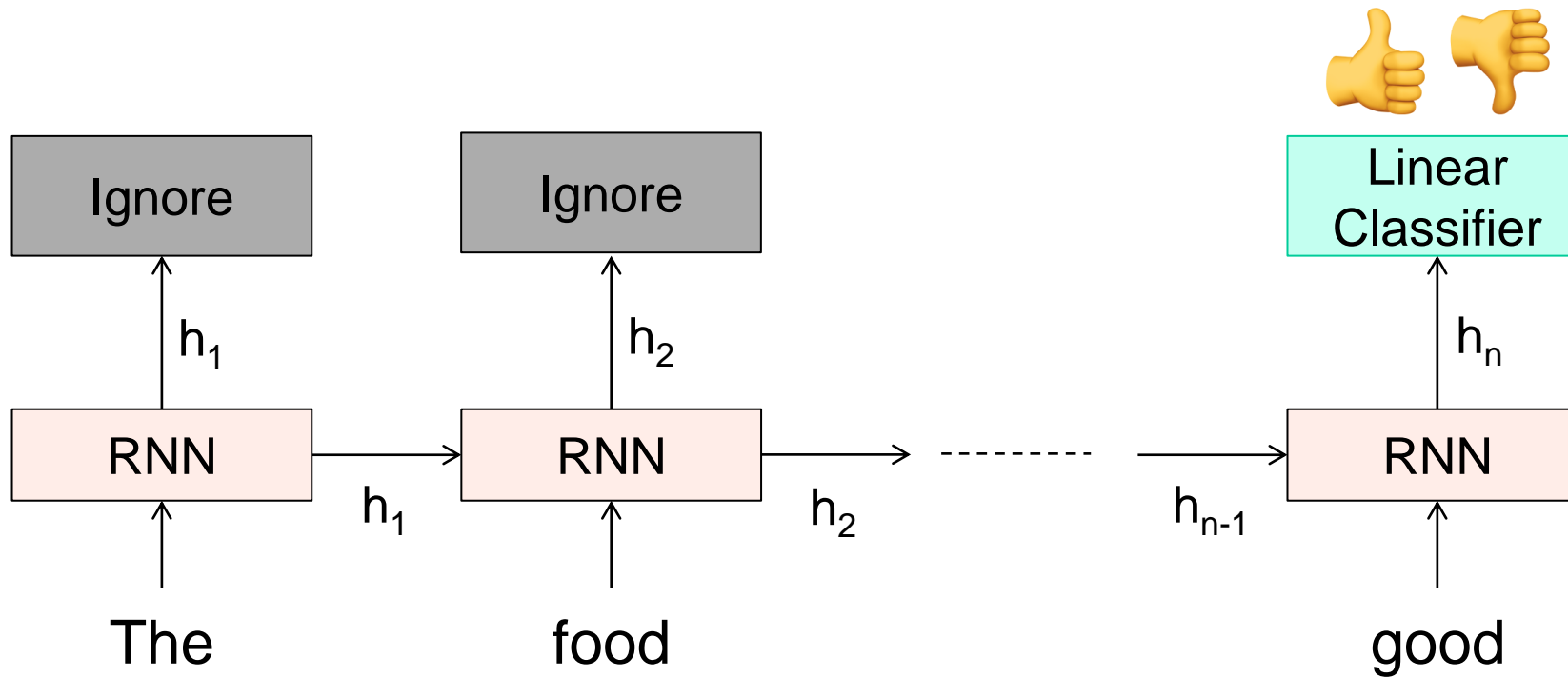


Y. Wu et al., [Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation](#), arXiv 2016

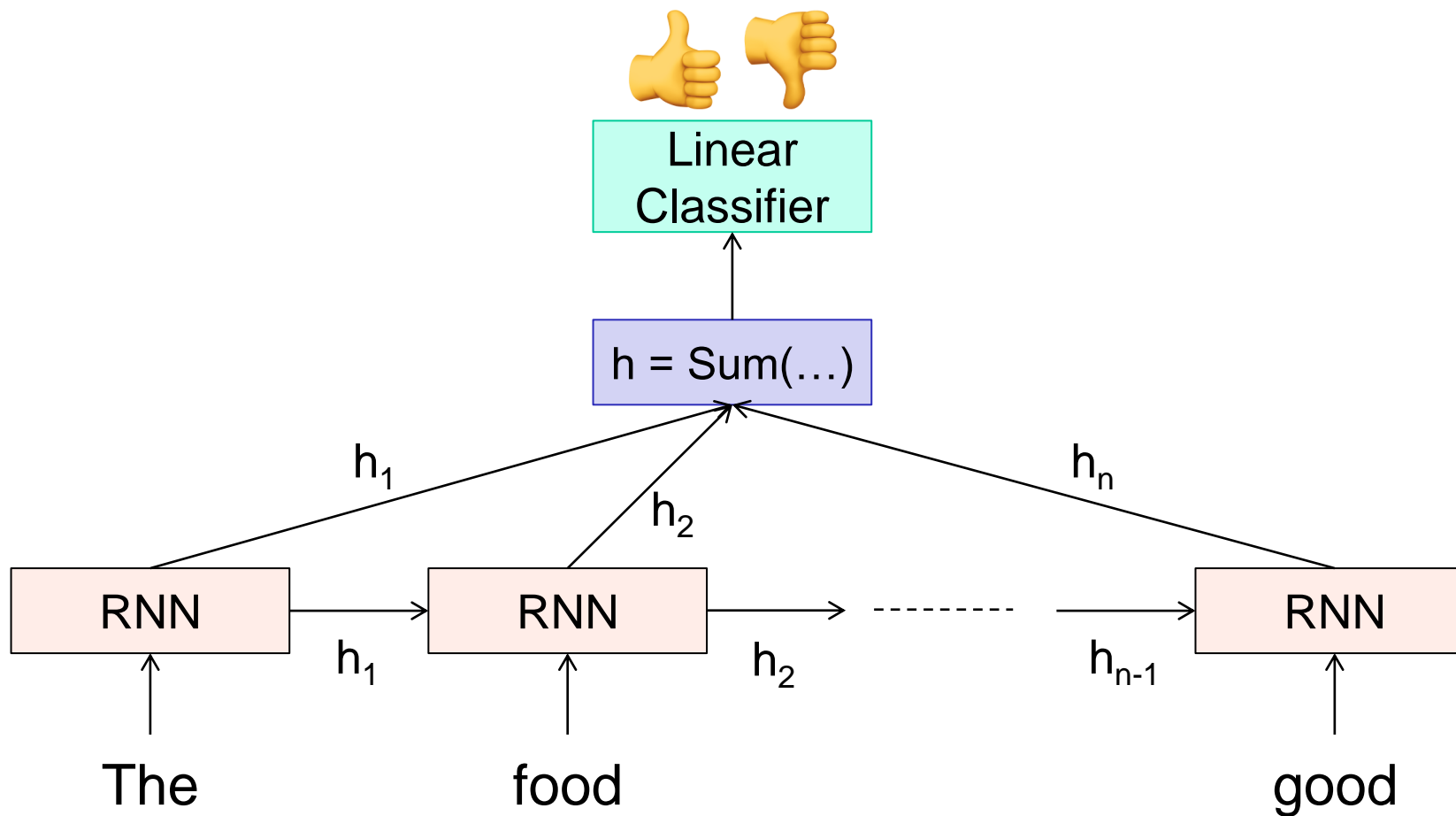
Outline

- Examples of sequential prediction tasks
- Common recurrent units
 - Vanilla RNN unit
 - Long Short-Term Memory (LSTM)
 - Gated Recurrent Unit (GRU)
- Recurrent network architectures
- Applications in (a bit) more detail
 - Sequence classification
 - Language modeling
 - Image captioning
 - Machine translation

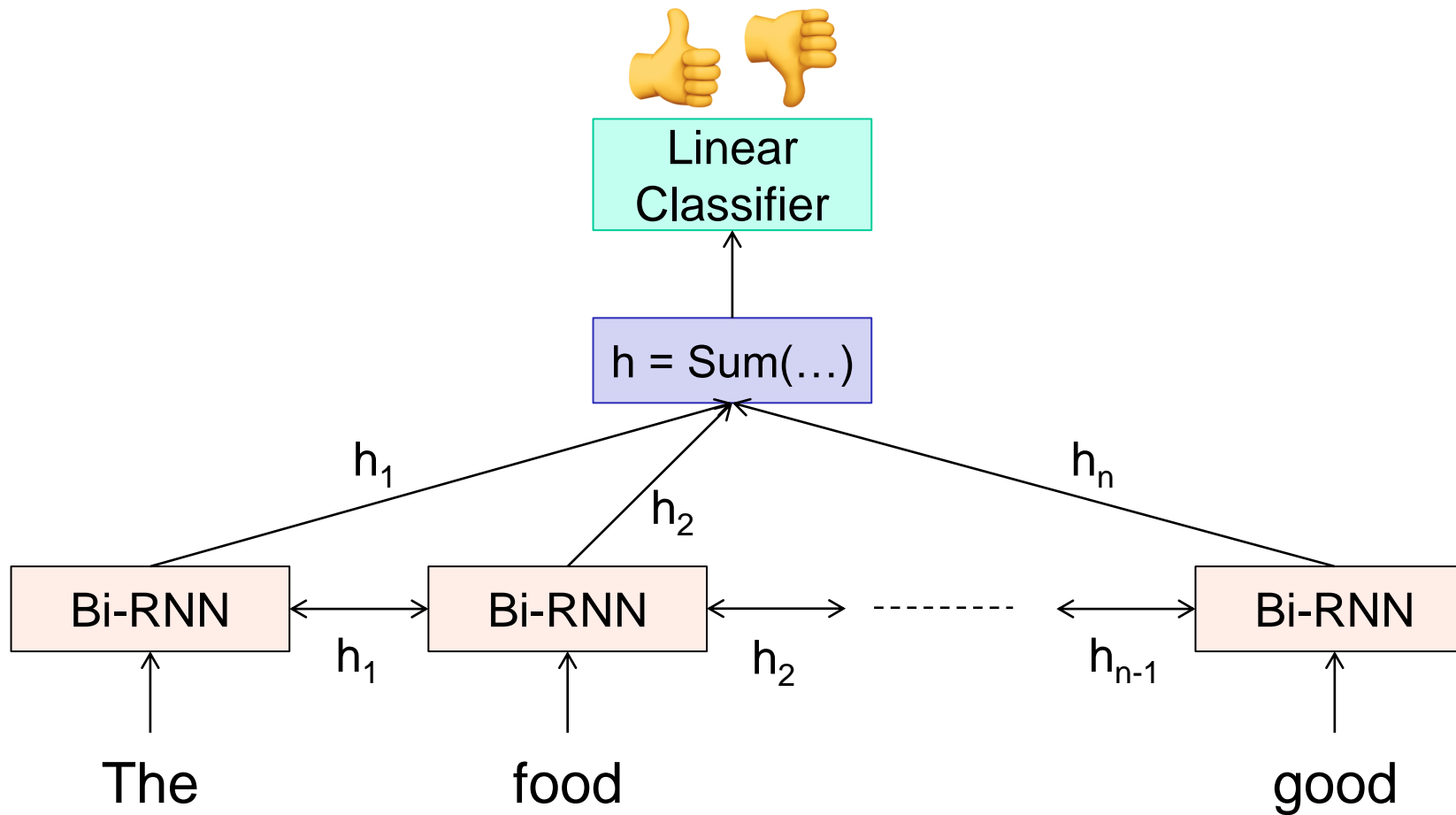
Sequence classification



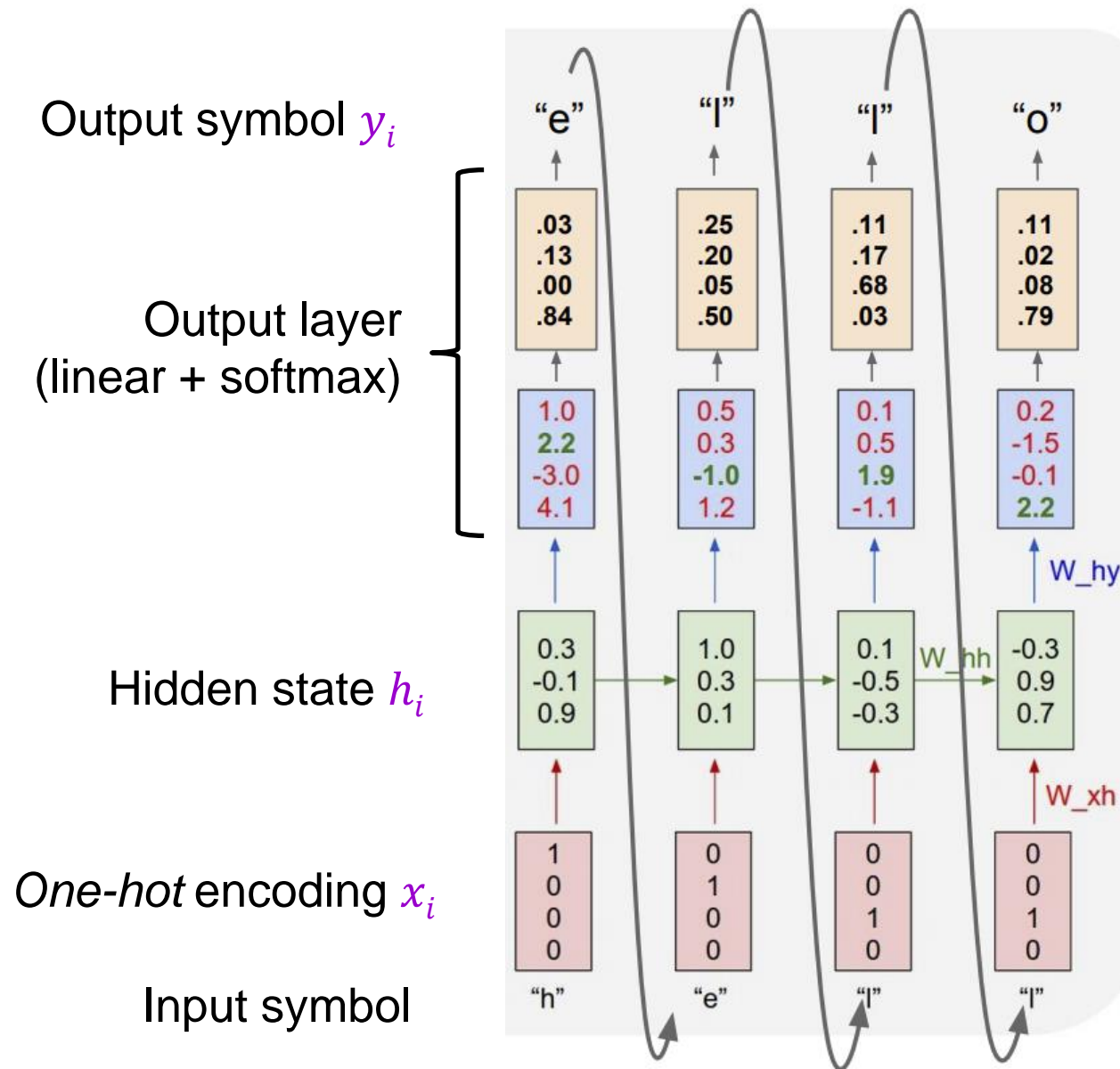
Sequence classification



Sequence classification



Language modeling: Character RNN



$$\begin{aligned} p(y_1, y_2, \dots, y_n) &= \prod_{i=1}^n p(y_i | y_1, \dots, y_{i-1}) \\ &\approx \prod_{i=1}^n P_W(y_i | h_i) \end{aligned}$$

Language modeling: Character RNN

100th
iteration

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

↓
train more

300th
iteration

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

↓
train more

700th
iteration

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

2000th
iteration

"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.

Searching for interpretable hidden units

"You mean to imply that I have nothing to eat out of.... On the contrary, I can supply you with everything even if you want to give dinner parties," warmly replied Chichagov, who tried by every word he spoke to prove his own rectitude and therefore imagined Kutuzov to be animated by the same desire.

Kutuzov, shrugging his shoulders, replied with his subtle penetrating smile: "I meant merely to say what I said."

quote detection cell

Searching for interpretable hidden units

The sole importance of the crossing of the Berezina lies in the fact that it plainly and indubitably proved the fallacy of all the plans for cutting off the enemy's retreat and the soundness of the only possible line of action--the one Kutuzov and the general mass of the army demanded--namely, simply to follow the enemy up. The French crowd fled at a continually increasing speed and all its energy was directed to reaching its goal. It fled like a wounded animal and it was impossible to block its path. This was shown not so much by the arrangements it made for crossing as by what took place at the bridges. When the bridges broke down, unarmed soldiers, people from Moscow and women with children who were with the French transport, all--carried on by vis inertiae--pressed forward into boats and into the ice-covered water and did not, surrender.

line position tracking cell

Searching for interpretable hidden units

```
static int __dequeue_signal(struct sigpending *pending, sigset_t *mask,
                           siginfo_t *info)
{
    int sig = next_signal(pending, mask);
    if (sig) {
        if (current->notifier) {
            if (sigismember(current->notifier_mask, sig)) {
                if (!(current->notifier)(current->notifier_data)) {
                    clear_thread_flag(TIF_SIGPENDING);
                    return 0;
                }
            }
        }
        collect_signal(sig, pending, info);
    }
    return sig;
}
```

if statement cell

Searching for interpretable hidden units

```
/* Duplicate LSM field information. The lsm_rule is opaque, so
 * re-initialized. */
static inline int audit_dupe_lsm_field(struct audit_field *df,
                                     struct audit_field *sf)
{
    int ret = 0;
    char *lsm_str;
    /* our own copy of lsm_str */
    lsm_str = kstrdup(sf->lsm_str, GFP_KERNEL);
    if (unlikely(!lsm_str))
        return -ENOMEM;
    df->lsm_str = lsm_str;
    /* our own (refreshed) copy of lsm_rule */
    ret = security_audit_rule_init(df->type, df->op, df->lsm_str,
                                  (void *)&df->lsm_rule);
    /* Keep currently invalid fields around in case they
     * become valid after a policy reload. */
    if (ret == -EINVAL) {
        pr_warn("audit rule for LSM '%s' is invalid\n",
                df->lsm_str);
        ret = 0;
    }
    return ret;
}
```

quote/comment cell

Searching for interpretable hidden units

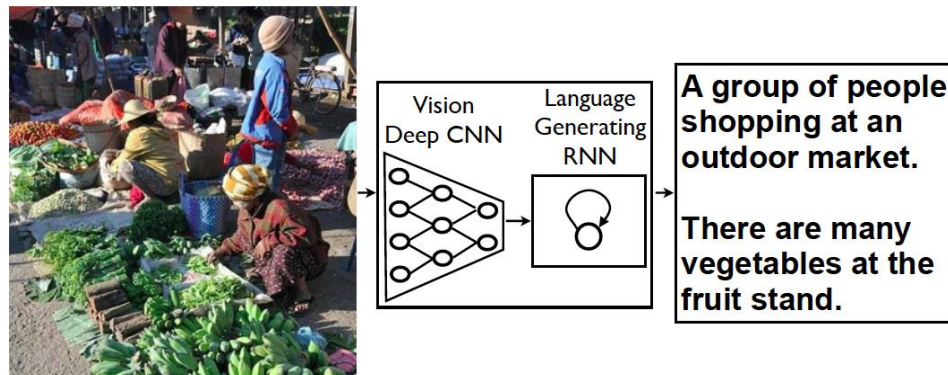
```
#ifdef CONFIG_AUDITSYSCALL
static inline int audit_match_class_bits(int class, u32 *mask)
{
    int i;
    if (classes[class]) {
        for (i = 0; i < AUDIT_BITMASK_SIZE; i++)
            if (mask[i] & classes[class][i])
                return 0;
    }
    return 1;
}
```

code depth cell

RNNs: Outline

- Examples of sequential prediction tasks
- Common recurrent units
 - Vanilla RNN unit
 - Long Short-Term Memory (LSTM)
 - Gated Recurrent Unit (GRU)
- Recurrent network architectures
 - Multilayer, bidirectional, skip connections
- Applications in (a bit) more detail
 - Sequence classification
 - Language modeling
 - Image captioning
 - Machine translation

Image caption generation



Training time

- Maximize likelihood of reference captions

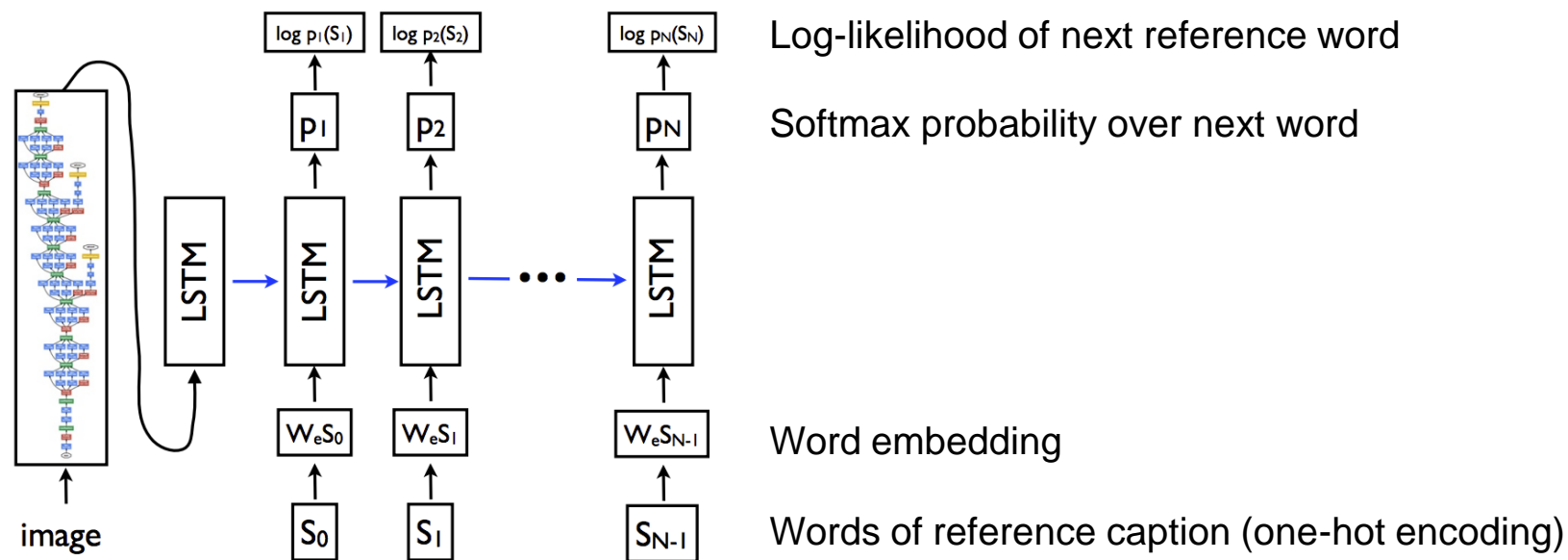


Image caption generation: Training time

- Minimize negative log-likelihood of the ground truth caption $Y^* = (Y_1^*, \dots, Y_N^*)$ given image I :

$$L(I, Y^*) = - \sum_{t=1}^N \log P_W(Y_t^* | Y_1^*, \dots, Y_{t-1}^*, I)$$

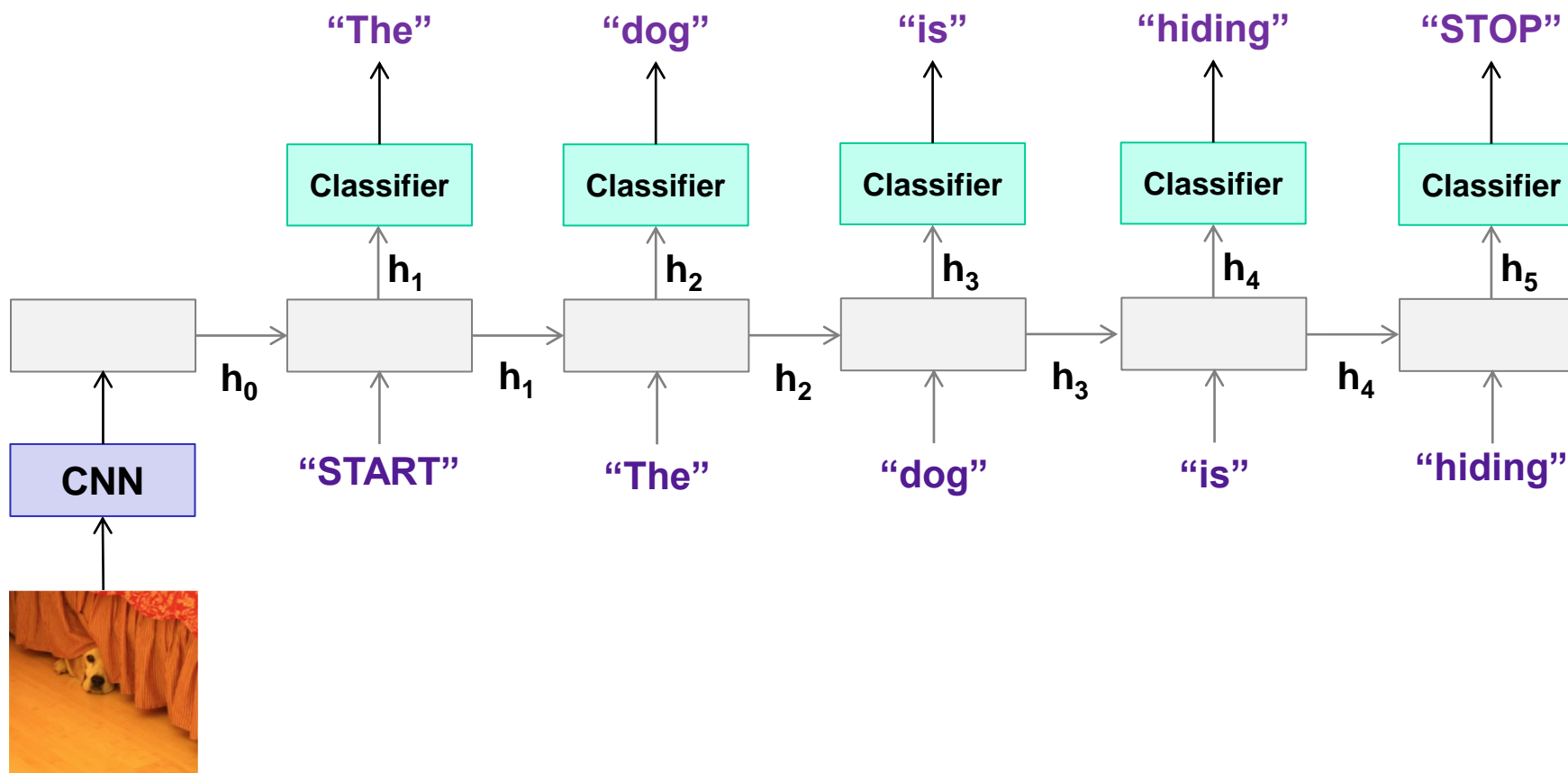


Image caption generation: Test time

- Sample next word according to posterior distribution of classifier
 - Sentences quickly become incoherent
- Always choose the highest-likelihood word
 - Does this necessarily maximize the likelihood of the overall sentence?

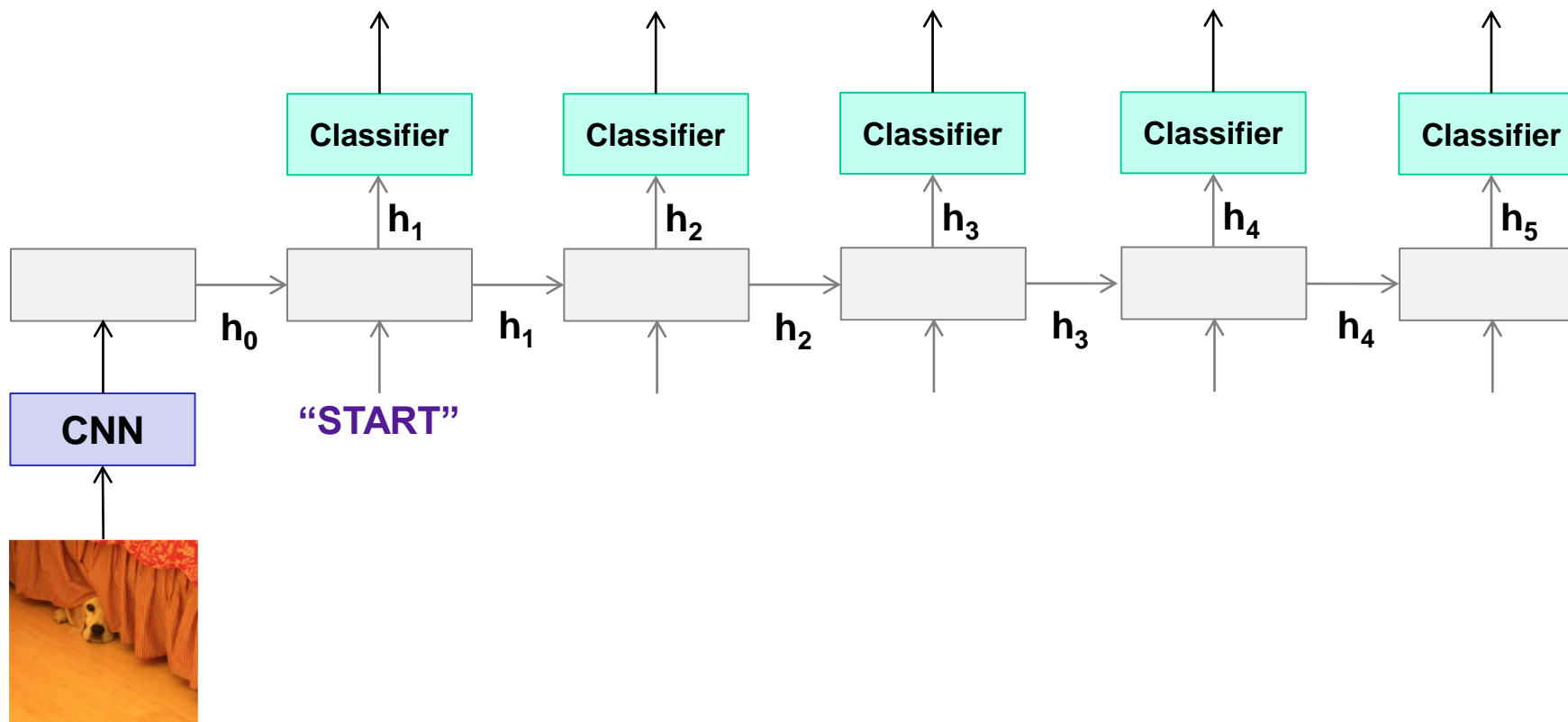


Image caption generation: Test time

- Beam search:
 - Maintain k (*beam width*) top-scoring candidate sentences according to sum of per-word log-likelihoods (or some other score)
 - At each step, generate all their successors and keep the best k

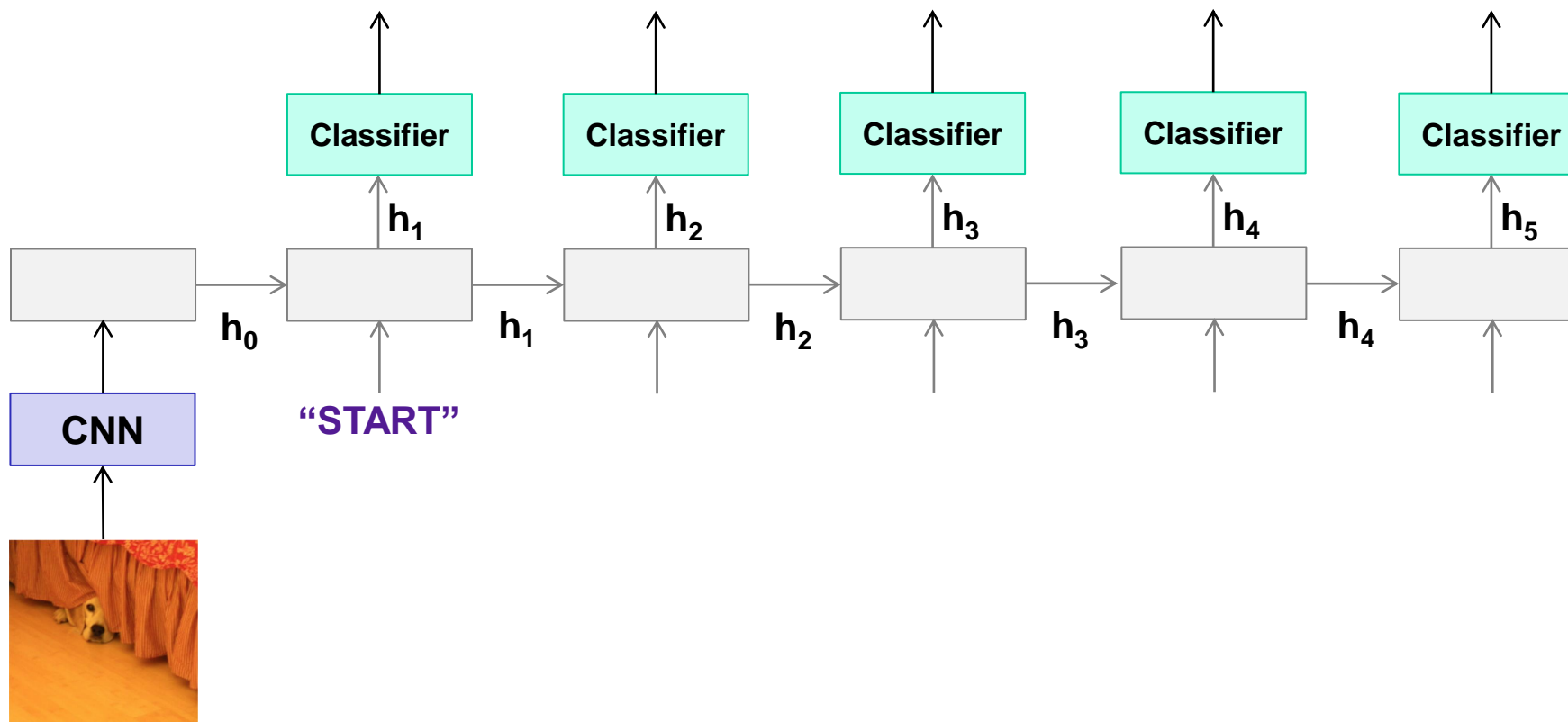


Image caption generation: Beam search

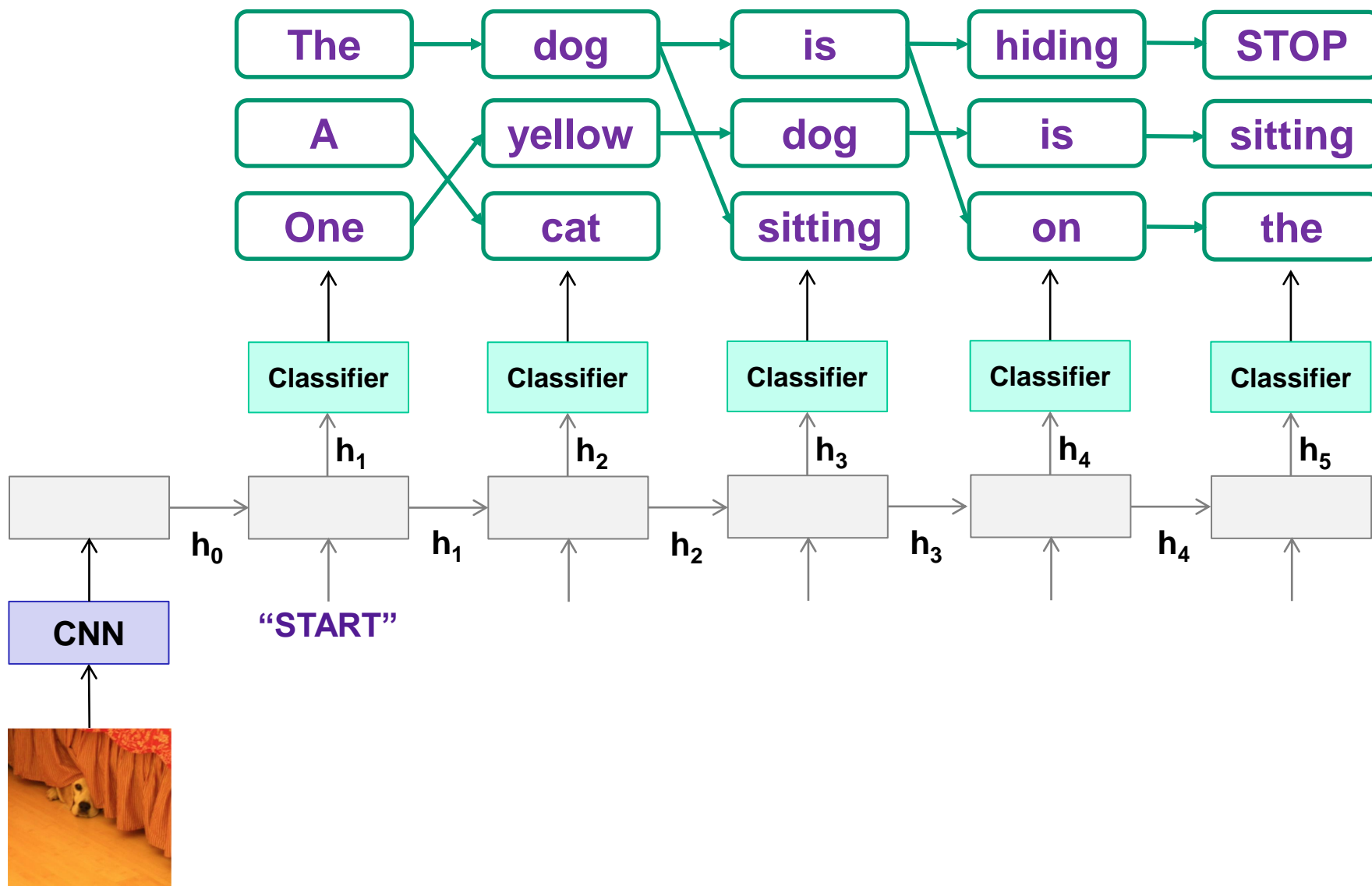


Image caption generation: Example outputs

A person riding a motorcycle on a dirt road.



Two dogs play in the grass.



A skateboarder does a trick on a ramp.



A dog is jumping to catch a frisbee.



A group of young people playing a game of frisbee.



Two hockey players are fighting over the puck.



A little girl in a pink hat is blowing bubbles.



A refrigerator filled with lots of food and drinks.



A herd of elephants walking across a dry grass field.



A close up of a cat laying on a couch.



A red motorcycle parked on the side of the road.



A yellow school bus parked in a parking lot.



Describes without errors

Describes with minor errors

Somewhat related to the image

Unrelated to the image

How to evaluate image captioning?



Reference sentences (written by human annotators):

- “A dog hides underneath a bed with its face peeking out of the bed skirt”
- “The small white dog is peeking out from under the bed”
- “A dog is peeking its head out from underneath a bed skirt”
- “A dog peeking out from under a bed”
- “A dog that is under a bed on the floor”

Generated sentence:

- “A dog is hiding”

BLEU: Bilingual Evaluation Understudy

- **N-gram precision:** count the number of n-gram matches between candidate and reference translation, divide by total number of n-grams in candidate translation
 - Clip counts by the maximum number of times an n-gram occurs in any reference translation
 - Multiply by *brevity penalty* to penalize short translations
- Most commonly used measure for image captioning and machine translation despite multiple [shortcomings](#)



Overview

Challenges

Download

Evaluate

Leaderboard

Table-C5

Table-C40

2015 Captioning Challenge

Last update: June 8, 2015. Visit [CodaLab](#) for the latest results.

| | CIDEr-D | Meteor | ROUGE-L | BLEU-1 | BLEU-2 | BLEU-3 | BLEU-4 |
|-------------------------------------|---------|--|---------|--------|--------|--------|--------|
| m-RNN (Baidu/ UCLA) ^[16] | 0.886 | 0.238 | 0.524 | 0.72 | 0.553 | 0.41 | 0.302 |
| m-RNN ^[15] | 0.847 | 0.240 | 0.504 | 0.710 | 0.545 | 0.404 | 0.299 |
| MSR Captiva | | | | | | | 0.308 |
| Google ^[4] | CIDEr-D | CIDEr: Consensus-based Image Description Evaluation | | | | | 0.309 |
| Berkeley LR | METEOR | Meteor Universal: Language Specific Translation Evaluation for Any Target Language | | | | | 0.277 |
| Nearest Neig | Rouge-L | ROUGE: A Package for Automatic Evaluation of Summaries | | | | | 0.28 |
| MSR ^[8] | BLEU | BLEU: a Method for Automatic Evaluation of Machine Translation | | | | | 0.291 |
| Montreal/Toronto ^[10] | 0.85 | 0.243 | 0.513 | 0.689 | 0.515 | 0.372 | 0.268 |
| PicSOM ^[13] | 0.833 | 0.231 | 0.505 | 0.683 | 0.51 | 0.377 | 0.281 |
| Tsinghua Bigeye ^[14] | 0.673 | 0.207 | 0.49 | 0.671 | 0.494 | 0.35 | 0.241 |
| MLBL ^[7] | 0.74 | 0.219 | 0.499 | 0.666 | 0.498 | 0.362 | 0.26 |
| Human ^[5] | 0.854 | 0.252 | 0.484 | 0.663 | 0.469 | 0.321 | 0.217 |



Table-C5

Table-C40

2015 Captioning Challenge

Last update: June 8, 2015. Visit [CodaLab](#) for the latest results.

| | M1 | ↓ M2 | M3 | M4 | M5 |
|-------------------------------------|-------|--|-------|-------|-------|
| Human ^[5] | 0.638 | 0.675 | 4.836 | 3.428 | 0.352 |
| Google ^[4] | 0.272 | 0.217 | 4.107 | 2.712 | 0.222 |
| MSR ^[8] | M1 | Percentage of captions that are evaluated as better or equal to human caption. | | | |
| Montreal | M2 | Percentage of captions that pass the Turing Test. | | | |
| MSR Ca | M3 | Average correctness of the captions on a scale 1-5 (incorrect - correct). | | | |
| Berkeley | M4 | Average amount of detail of the captions on a scale 1-5 (lack of details - very detailed). | | | |
| m-RNN ^[1] | M5 | Percentage of captions that are similar to human description. | | | |
| Nearest Neighbor ^[11] | 0.216 | 0.255 | 3.801 | 2.716 | 0.196 |
| PicSOM ^[13] | 0.202 | 0.250 | 3.965 | 2.552 | 0.182 |
| Brno University ^[3] | 0.194 | 0.213 | 3.079 | 3.482 | 0.154 |
| m-RNN (Baidu/ UCLA) ^[16] | 0.190 | 0.241 | 3.831 | 2.548 | 0.195 |
| MIL ^[6] | 0.168 | 0.197 | 3.349 | 2.915 | 0.159 |
| MLBL ^[7] | 0.167 | 0.196 | 3.659 | 2.420 | 0.156 |

Generative model for diverse captioning

- We would like to sample diverse captions given an image to accurately reflect intrinsic open-endedness of the task



LSTM + beam search output lacks diversity

a close up of a plate of food with a sandwich on a table
a close up of a sandwich on a plate
a close up of a plate of food on a table
a close up of a plate of food with a sandwich on it
a close up of a plate of food on a white plate

Generative model for diverse captioning

- We would like to sample diverse captions given an image to accurately reflect intrinsic open-endedness of the task



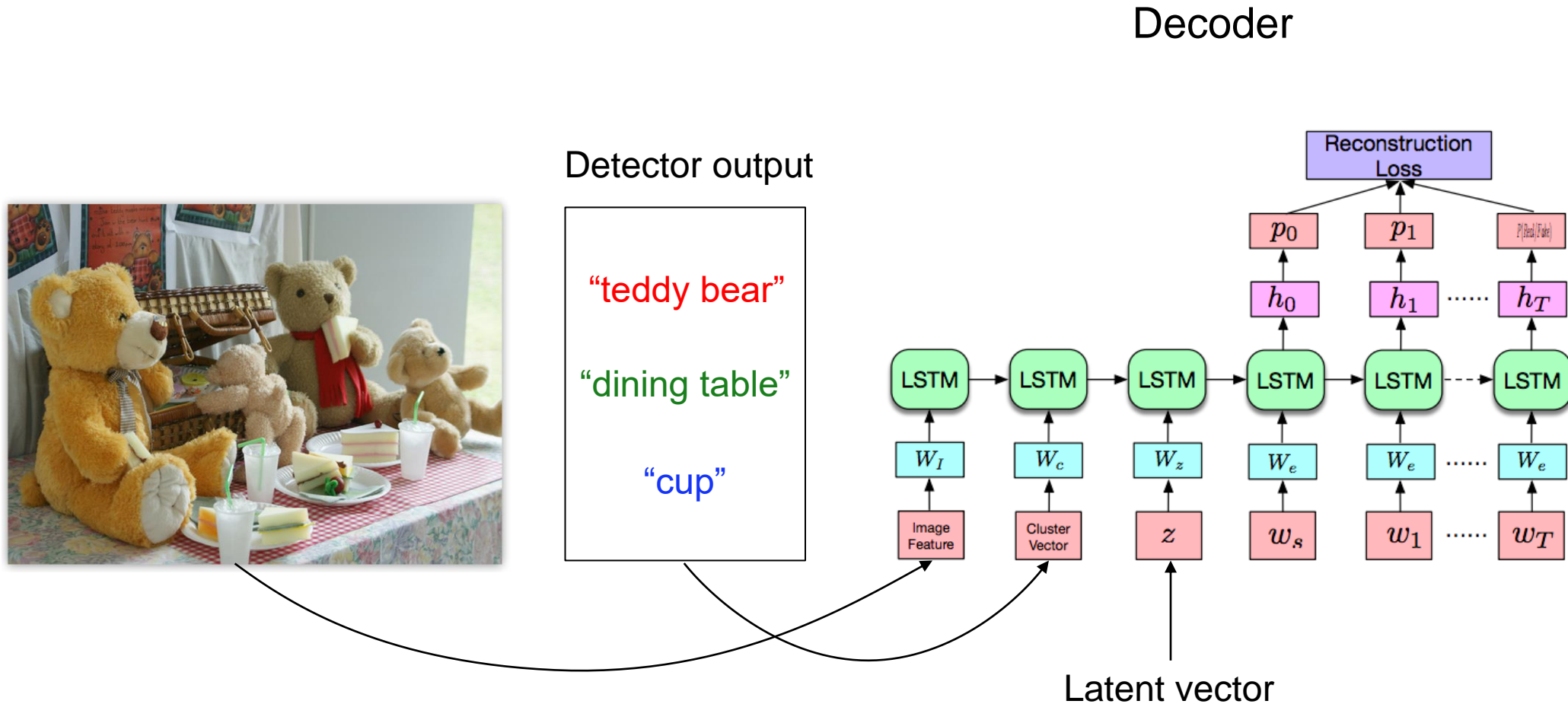
LSTM + beam search output lacks diversity

a close up of a plate of food with a sandwich on a table
a close up of a sandwich on a plate
a close up of a plate of food on a table
a close up of a plate of food with a sandwich on it
a close up of a plate of food on a white plate

Conditional variational auto-encoder with additive Gaussian space (AG-CVAE)

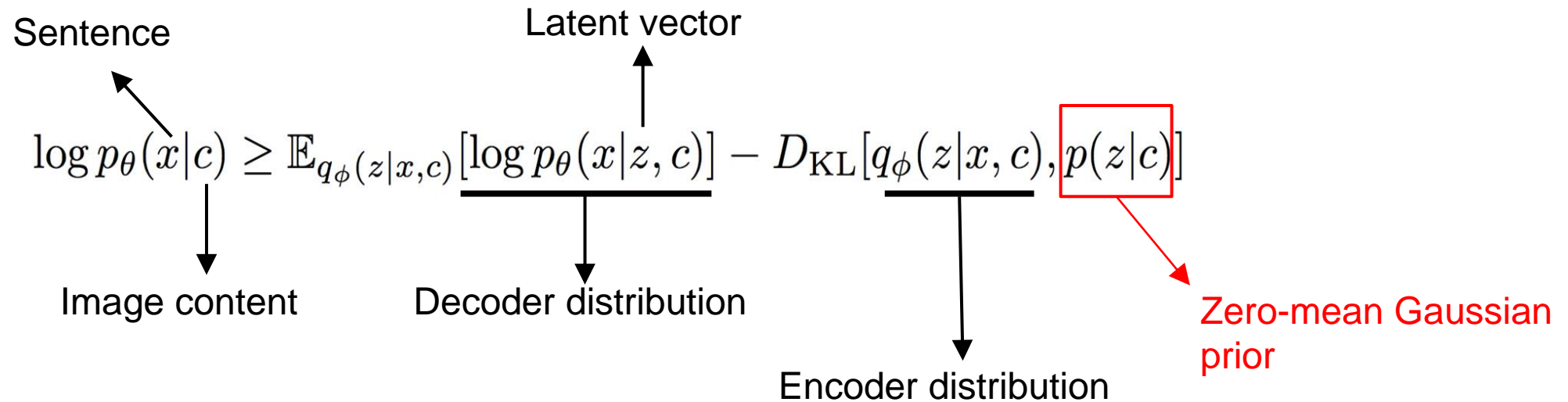
a close up of a plate of food on a table
a table with a plate of food on it
a plate of food with a sandwich on it
a white plate topped with a plate of food
a plate of food on a table next to a cup of coffee

CVAE for captioning



CVAE for captioning

Standard CVAE objective:

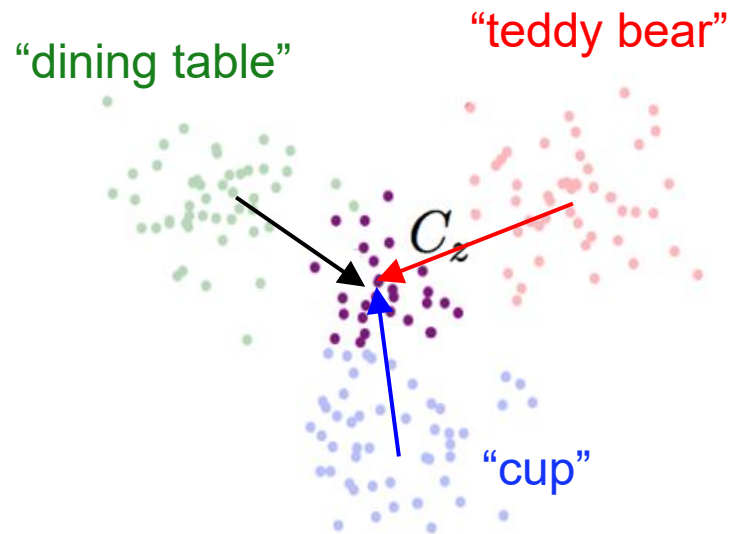


CVAE with additive Gaussian prior

Proposed objective: shift prior mean based on image content

$$\max_{\theta, \phi} \sum_{i=1}^N \log p_{\theta}(x^i | z^i, c^i) - D_{\text{KL}}[q_{\phi}(z | x, c), p(z | c)], \quad \text{s.t. } \forall i \ z^i \sim q_{\phi}(z | x, c).$$

$$p(z | c) = \mathcal{N} \left(z \mid \sum_{k=1}^K c_k \mu_k, \sigma^2 \mathbf{I} \right)$$



Results

- More controllable captions: changing the conditioning vector of object labels changes the caption in a reasonable way



Predicted Object Labels:

'person' 'cup' 'donut' 'dining table'

AG-CVAE:

a woman sitting at a table with a cup of coffee
a person sitting at a table with a cup of coffee
a table with two plates of donuts and a cup of coffee
a woman sitting at a table with a plate of coffee
a man sitting at a table with a plate of food

LSTM Baseline:

a close up of a table with two plates of coffee
a close up of a table with a plate of food
a close up of a plate of food on a table
a close up of a table with two plates of food
a close up of a table with plates of food

Results

- More controllable captions: changing the conditioning vector of object labels changes the caption in a reasonable way



Object Labels: 'person'

AG-CVAE sentences:

a man and a woman standing in a room
a man and a woman are playing a game
a man standing next to a woman in a room
a man standing next to a woman in a field
a man standing next to a woman in a suit

Object Labels: 'person', 'remote'

AG-CVAE sentences:

a man and a woman playing a video game
a man and a woman are playing a video game
a man and woman are playing a video game
a man and a woman playing a game with a remote
a woman holding a nintendo wii game controller

Acknowledgement

Thanks to the following courses and corresponding researchers for making their teaching/research material online

- Deep Learning, Stanford University
- Introduction to Deep Learning, University of Illinois at Urbana-Champaign
- Introduction to Deep Learning, Carnegie Mellon University
- Convolutional Neural Networks for Visual Recognition, Stanford University
- Natural Language Processing with Deep Learning, Stanford University
- And Many More