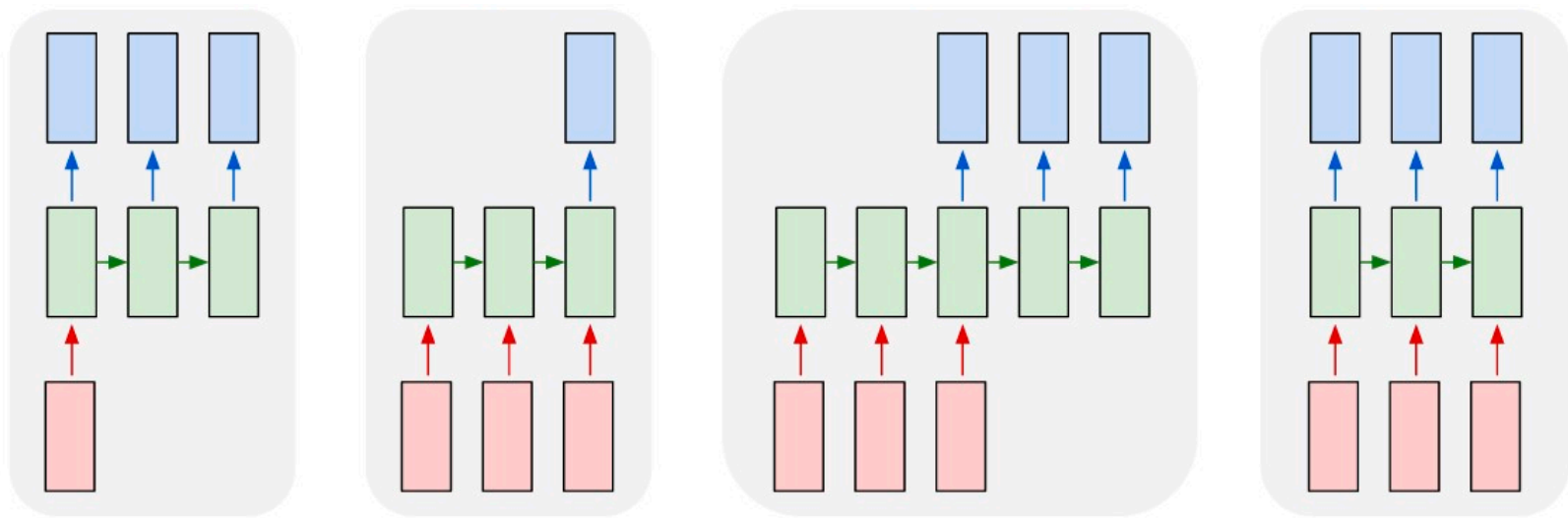


Recurrent networks



Many slides adapted from Arun Mallya and [Justin Johnson](#) (and Stanford CS231n)

[Image source](#)

Outline

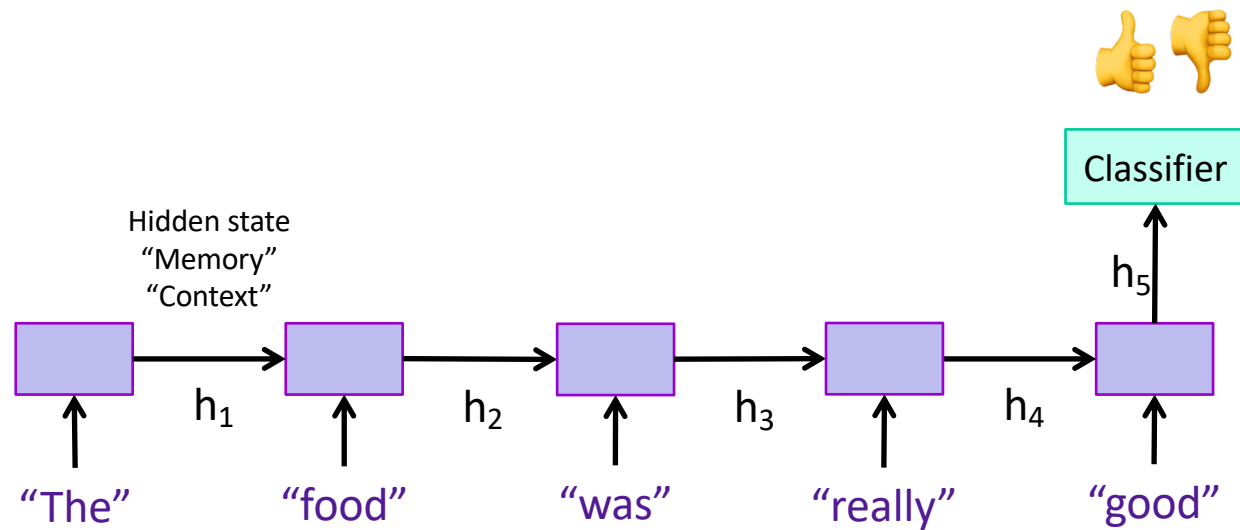
- Sequential prediction tasks
- Common recurrent units
 - Vanilla RNN unit (and how to train it)
 - Long Short-Term Memory (LSTM)
 - Gated Recurrent Unit (GRU)
- Recurrent network architectures
- Applications in (a bit) more detail
 - Language modeling
 - Image captioning

Sequential prediction 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”

Sequential prediction example 1: Sentiment classification

- Recurrent model:



Sequential prediction example 2: Text generation

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



RNN Bible
@RNN_Bible

Random bible verses generated using Recurrent Neural Networks (char-rnn).
Joined May 2015

Tweets 2,197 Following 1 Followers 485

Tweets Tweets & replies

RNN Bible @RNN_Bible · 20 Jun 2016
24:11 Thus saith the LORD of hosts; Ask now this stones are for the righteous and the children of Israel.
1 2 3

RNN Bible @RNN_Bible · 19 Jun 2016
24:16 And they took up twelve stones out of the city of David, and discomfit Jordan.
1

RNN Bible @RNN_Bible · 19 Jun 2016
3:20 And the LORD shall send a proverb against the LORD thy God, and shalt not each laugh.
1 5 3

RNN Bible @RNN_Bible · 19 Jun 2016
23:2 And the vision of the breaking thereof shall be in rubbick, and they shall take away the stones out of the land.
1



DeepDrumpf
@DeepDrumpf

I'm a Neural Network trained on Trump's transcripts. Priming text in []s. Donate (gofundme.com/deepdrumpf) to interact! Created by @hayesbh.
Joined March 2016
7 Following 24.6K Followers

Tweets Tweets & replies Media Likes

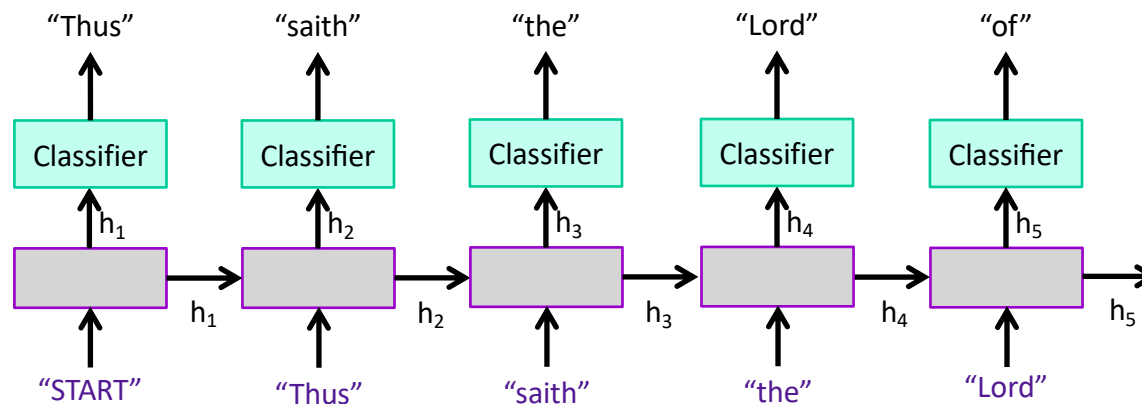
DeepDrumpf @DeepDrumpf · May 31, 2017
[Despite the negative press #covfefe] look at what's going on. They shoot media. Usually that's a bad sign of things to come.
6 38 124

DeepDrumpf @DeepDrumpf · Apr 7, 2017
When I have to build a hotel, we're bombing the hell out of them. Lots of money. To those suffering, I say vote for Donald. #SyriaStrikes
1 71 173

DeepDrumpf @DeepDrumpf · Mar 20, 2017
Replying to @Thomas1774Paine
There will be no amnesty. It is going to pass because the people are going to be gone. I'm giving a mandate. #ComeyHearing @Thomas1774Paine

Sequential prediction example 2: Text generation

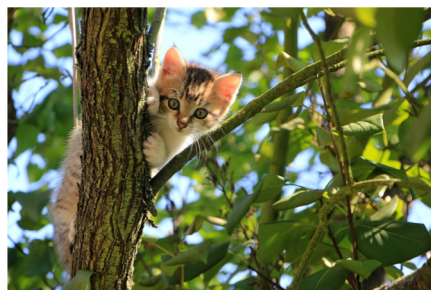
- Sample from the distribution of a given text corpus – also known as *language modeling*
- Can be done one character or one word at a time:



Sequential prediction example 3: Image 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



Two people walking on the beach with surfboards



A tennis player in action on the court



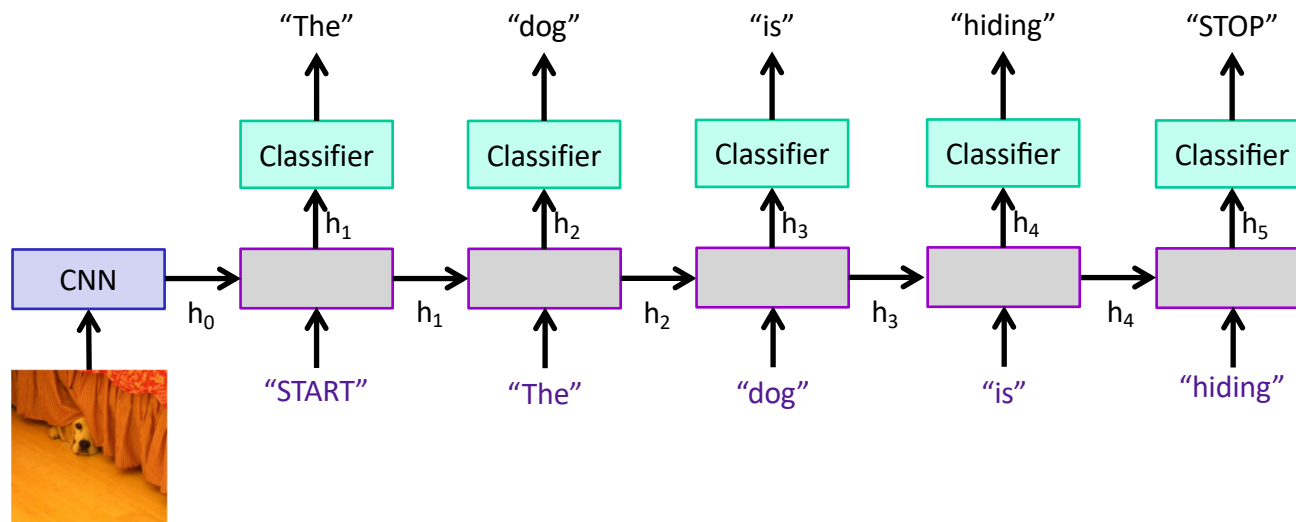
Two giraffes standing in a grassy field



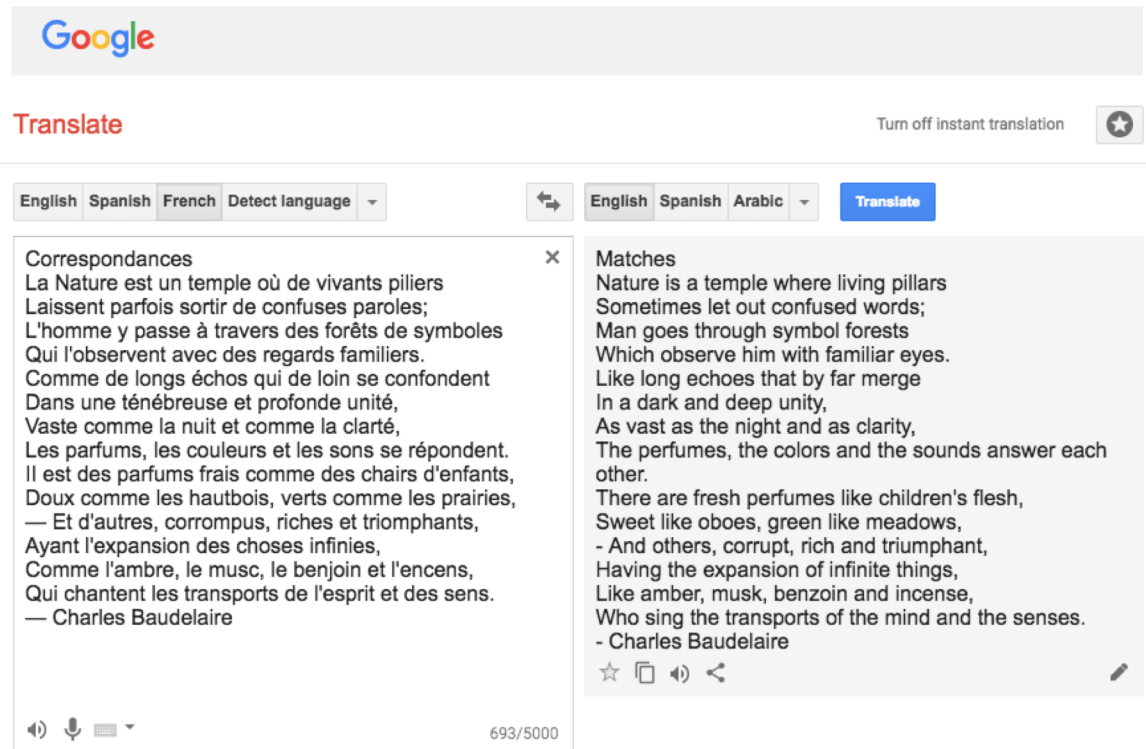
A man riding a dirt bike on a dirt track

Source: [J. Johnson](#)
Captions generated using [neuraltalk2](#)

Sequential prediction example 3: Image captioning



Example 4: Machine translation



The screenshot shows the Google Translate interface. At the top is the Google logo. Below it, the word "Translate" is displayed in red. To the right of "Translate" is a link that says "Turn off instant translation" and a star icon. Below this, there are two language selection bars. The left bar has buttons for "English", "Spanish", "French", and "Detect language", with a dropdown arrow. The right bar has buttons for "English", "Spanish", and "Arabic", with a dropdown arrow and a blue "Translate" button. The main content area is split into two columns. The left column is titled "Correspondances" and contains a French poem by Charles Baudelaire. The right column is titled "Matches" and contains the English translation of the poem. At the bottom of the left column, there are icons for a speaker, a microphone, and a document, along with the text "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

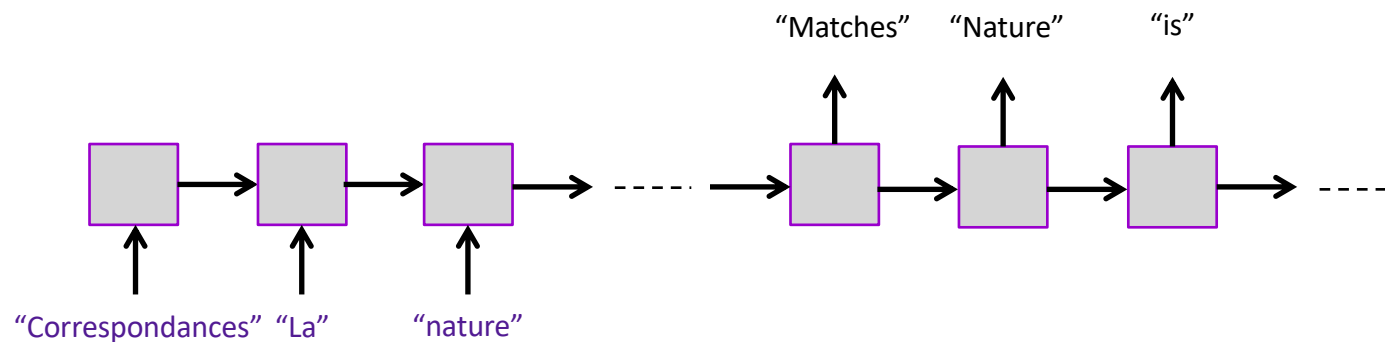
☆ 📄 🔊 🔗 ✎

693/5000

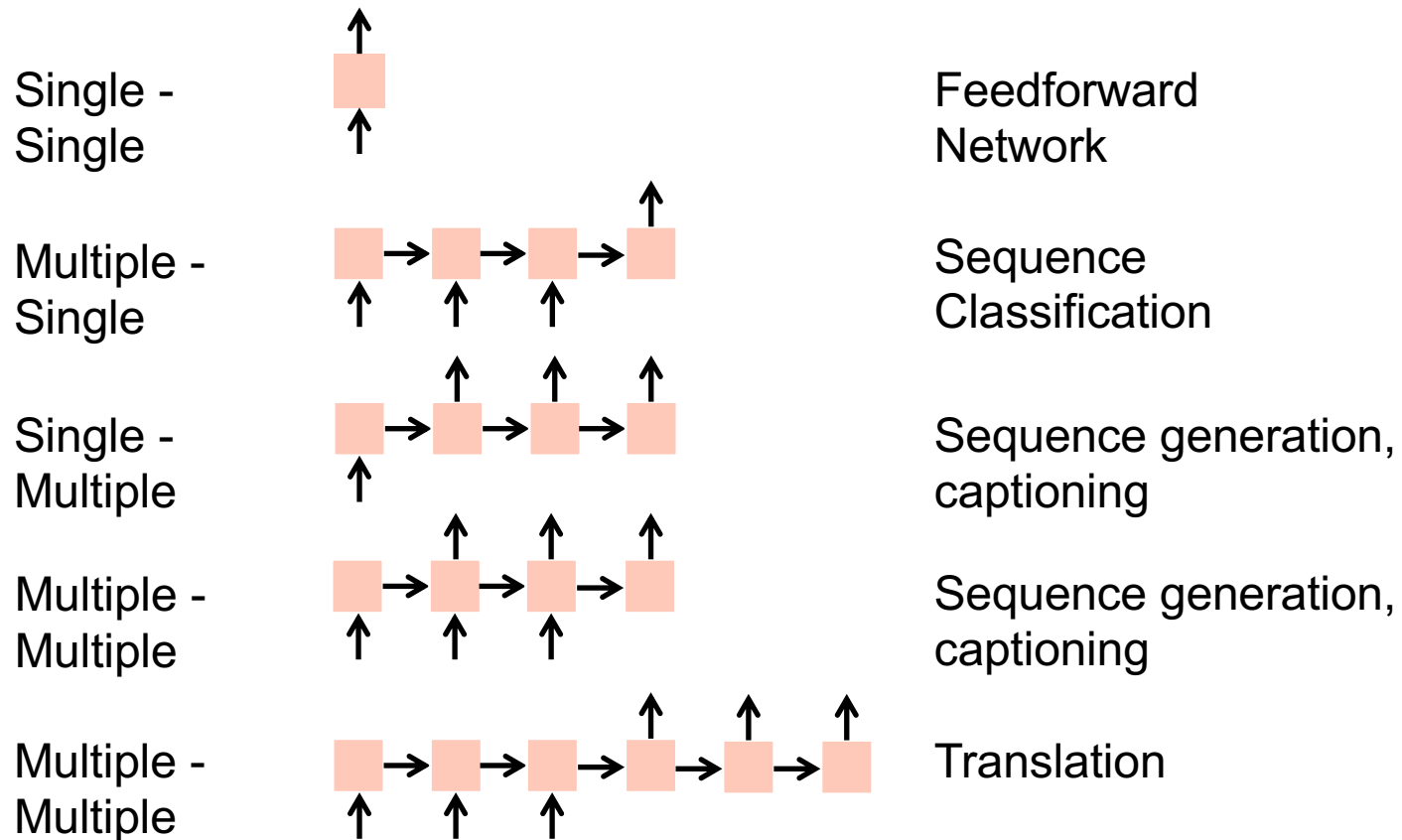
<https://translate.google.com/>

Example 4: Machine translation

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



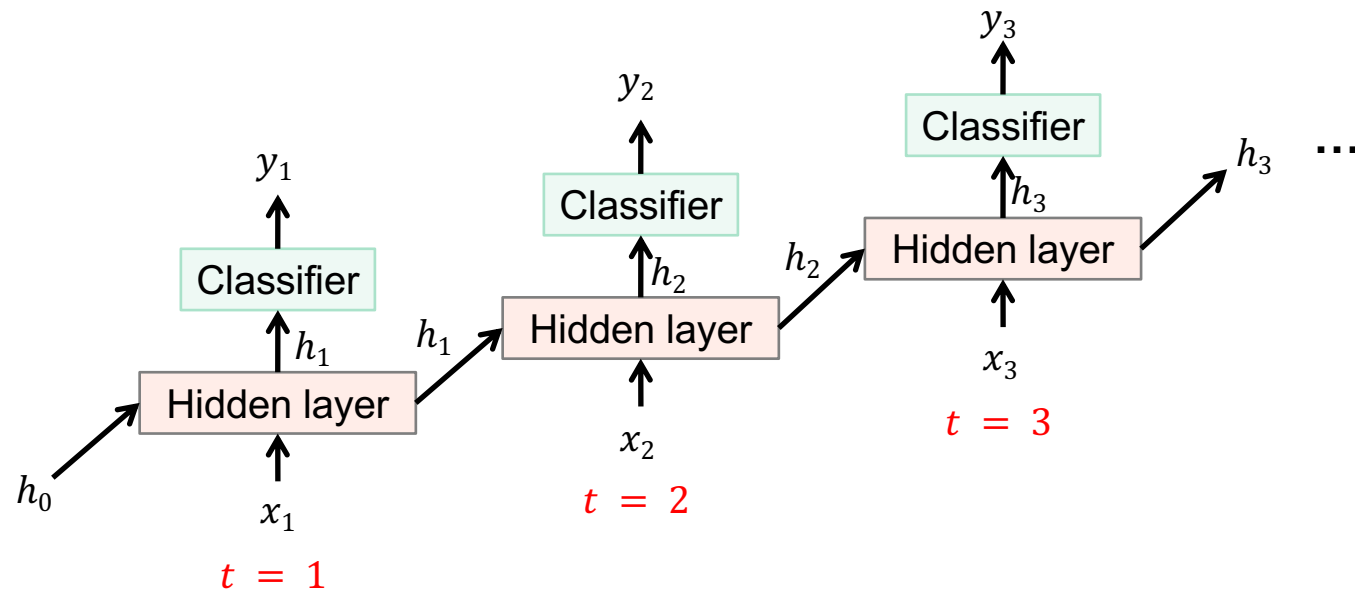
Summary: Input-output scenarios



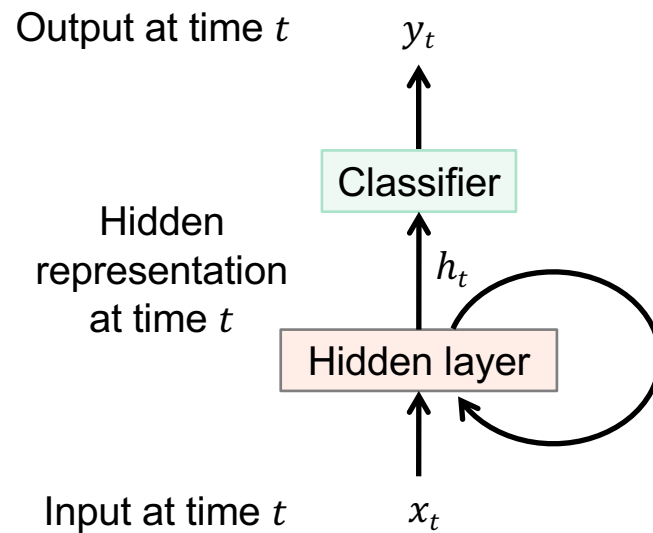
Outline

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

Recurrent unit



Recurrent unit

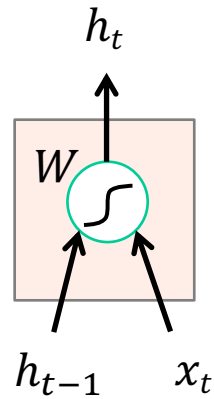


Recurrence:

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

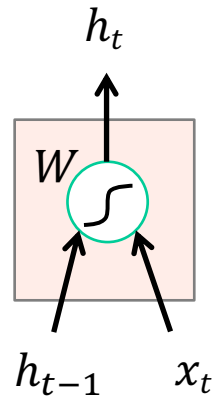
new state function of W input at time t old state

Vanilla RNN cell

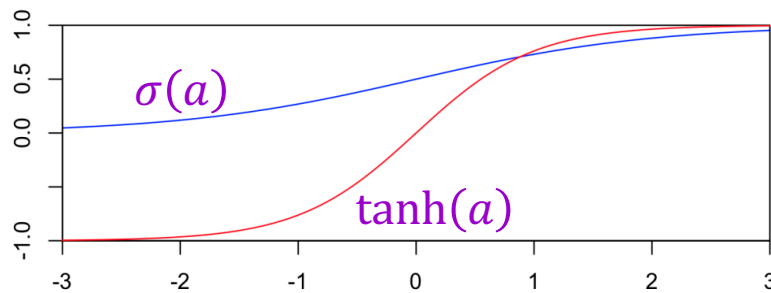


$$h_t = f_W(x_t, h_{t-1})$$
$$= \tanh W \begin{pmatrix} x_t \\ h_{t-1} \end{pmatrix}$$

Vanilla RNN cell

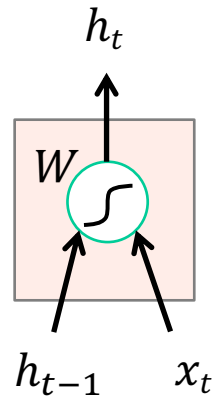


$$\begin{aligned} h_t &= f_W(x_t, h_{t-1}) \\ &= \tanh W \begin{pmatrix} x_t \\ h_{t-1} \end{pmatrix} \end{aligned}$$

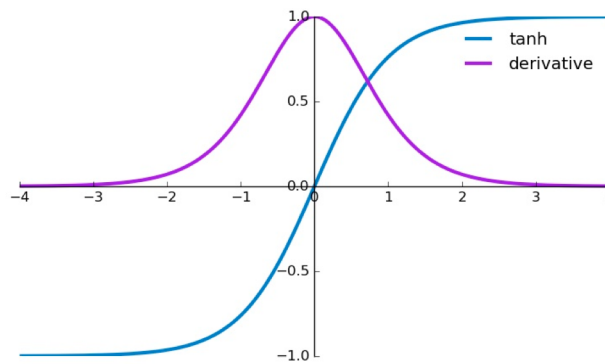


$$\begin{aligned} \tanh(a) &= \frac{e^a - e^{-a}}{e^a + e^{-a}} \\ &= 2\sigma(2a) - 1 \end{aligned}$$

Vanilla RNN cell

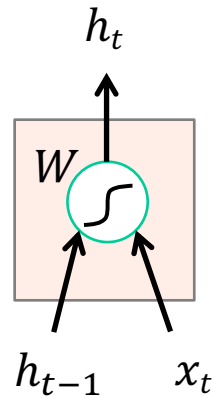


$$\begin{aligned} h_t &= f_W(x_t, h_{t-1}) \\ &= \tanh W \begin{pmatrix} x_t \\ h_{t-1} \end{pmatrix} \end{aligned}$$

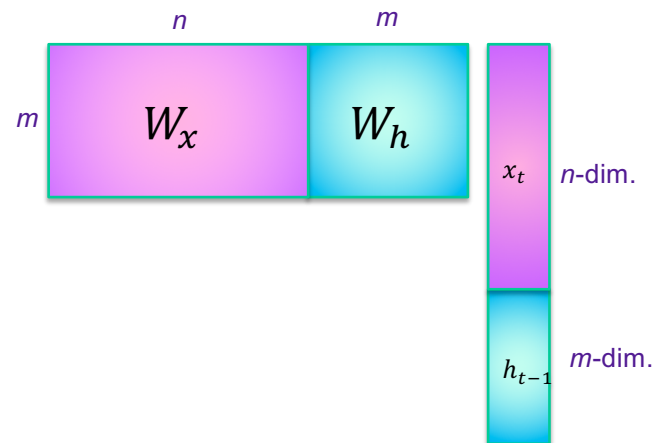


$$\frac{d}{da} \tanh(a) = 1 - \tanh^2(a)$$

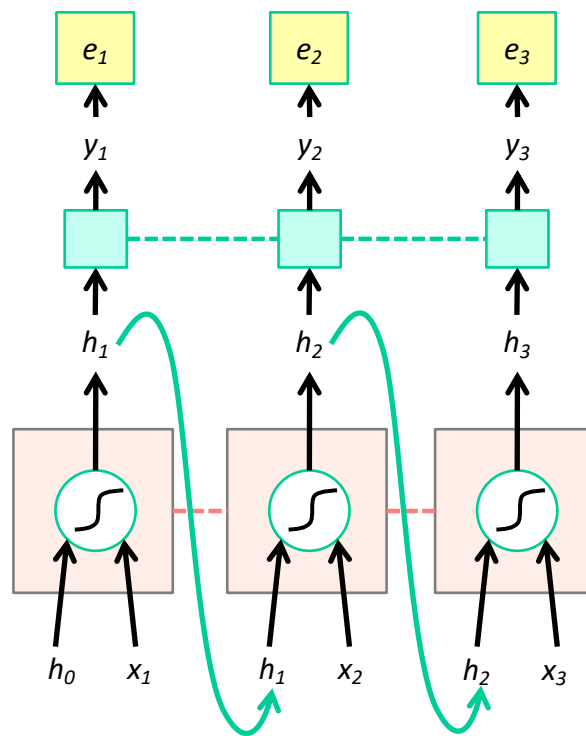
Vanilla RNN cell



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



RNN forward pass



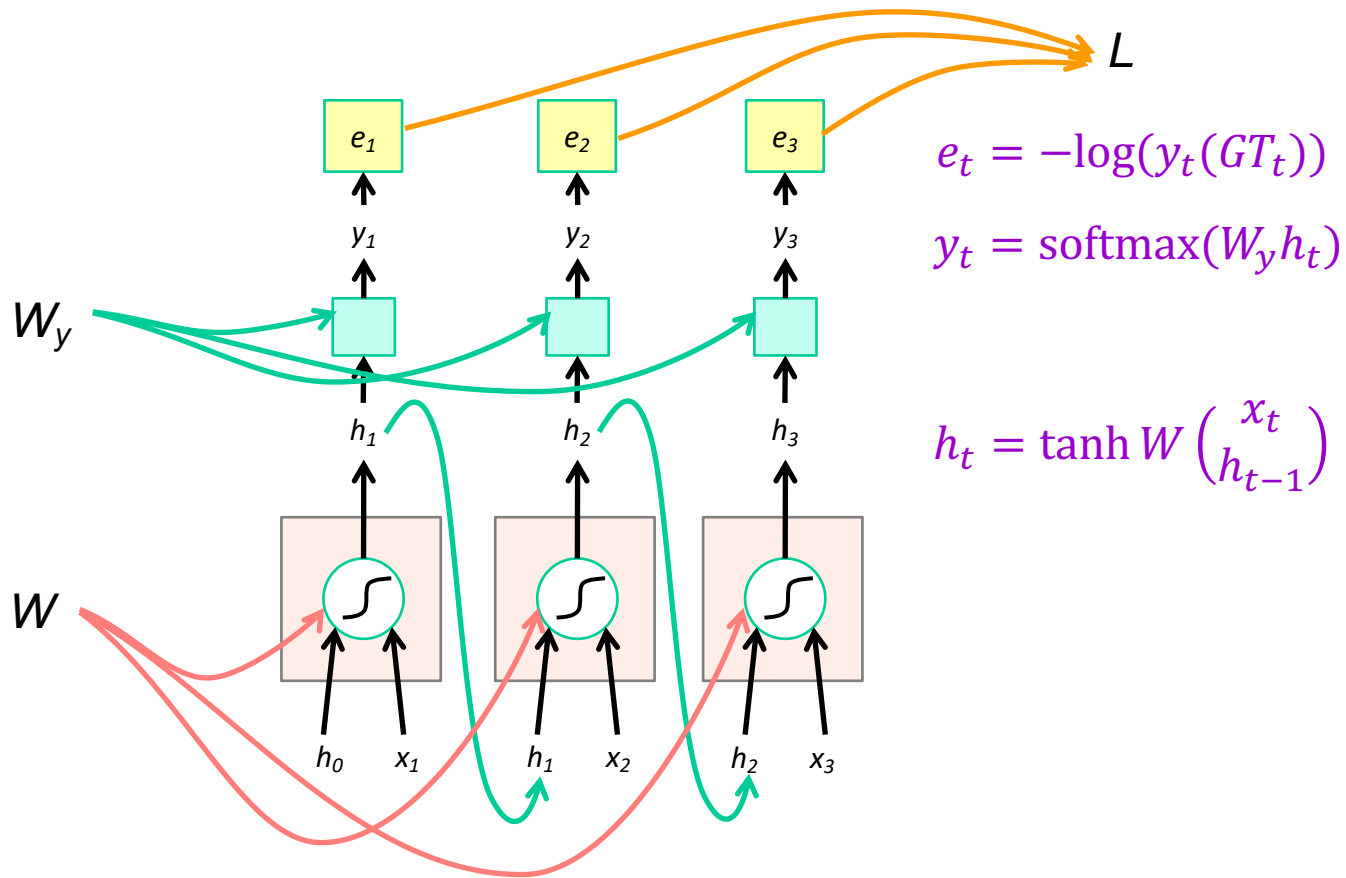
$$e_t = -\log(y_t(GT_t))$$

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

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

----- shared weights

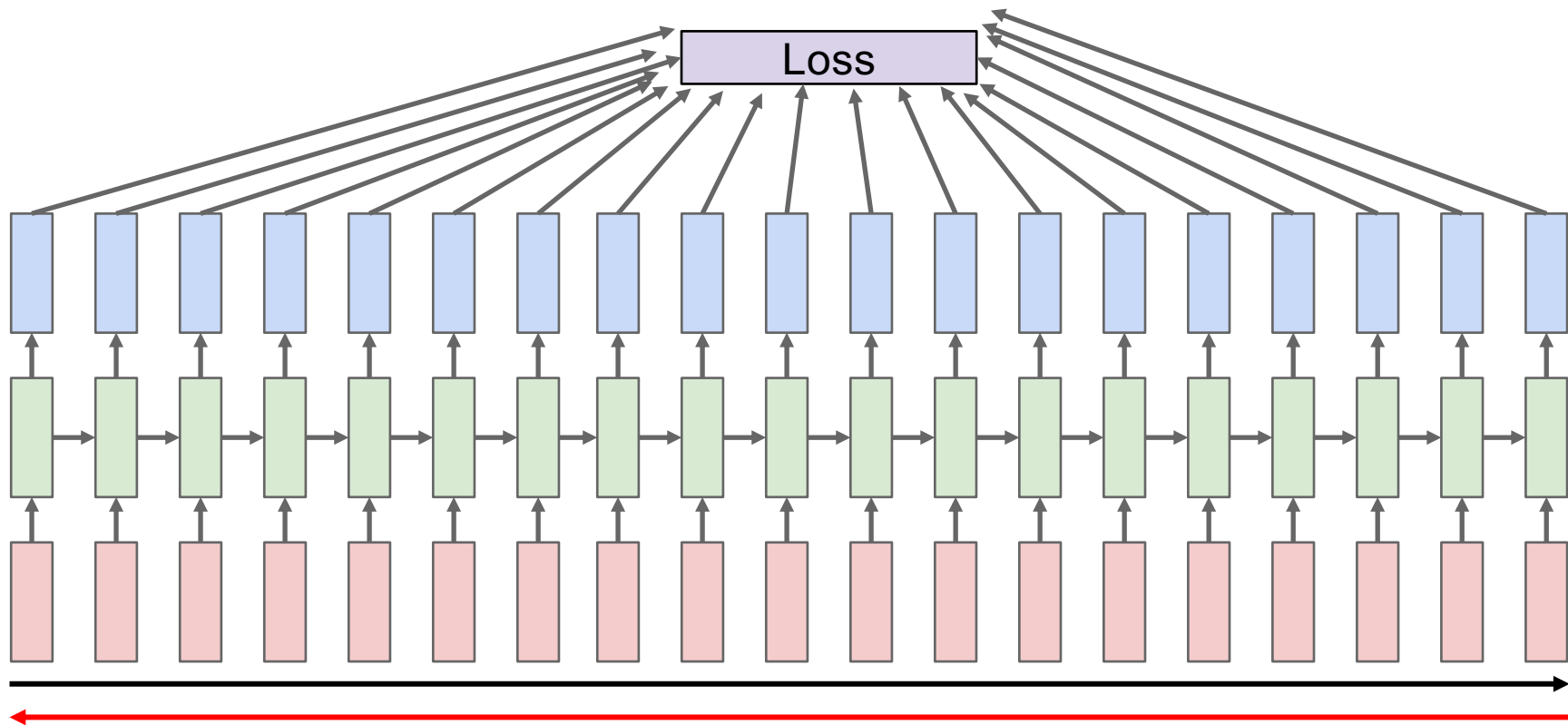
RNN forward pass: Computation graph



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

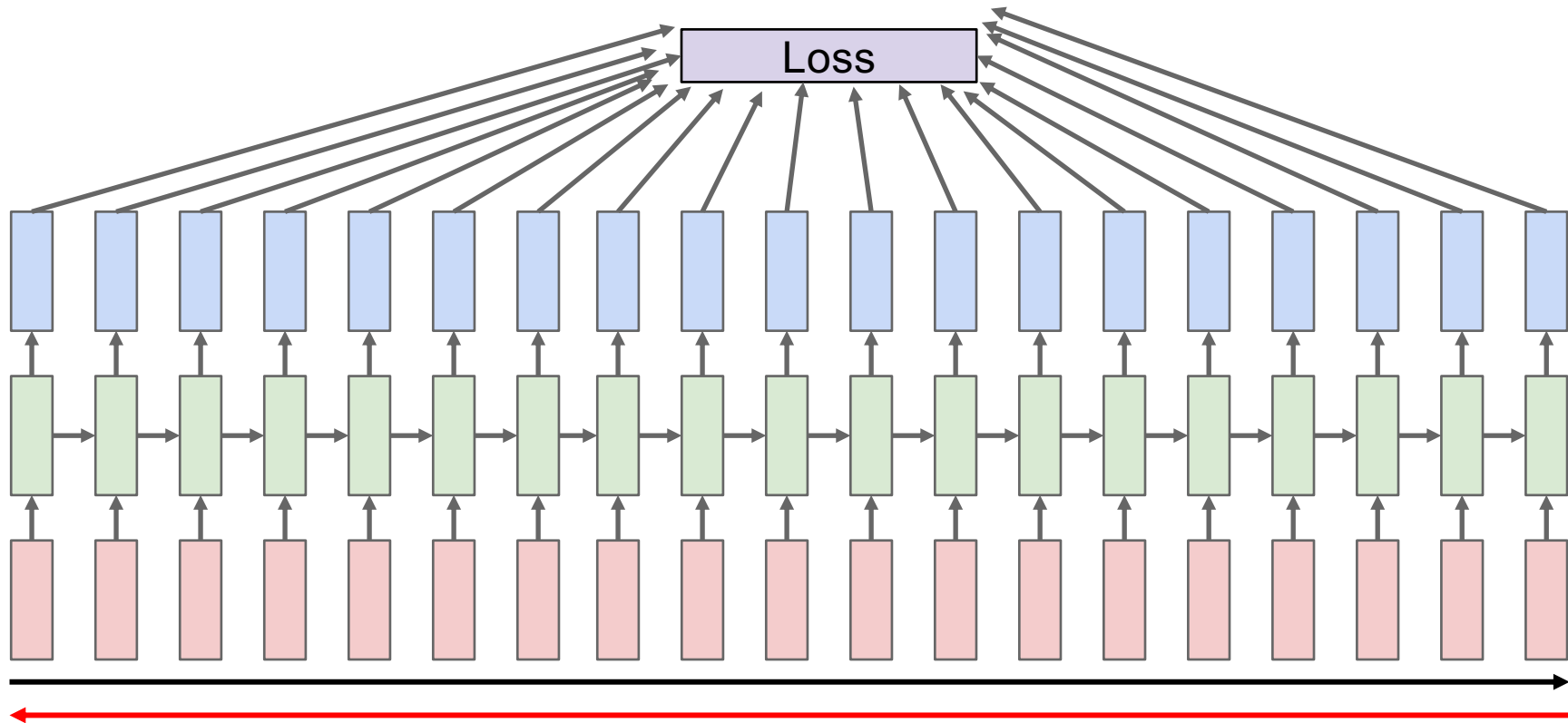
Backpropagation through time



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

Source: [J. Johnson](#)

Backpropagation through time



Problem: Takes a lot of memory for long sequences!

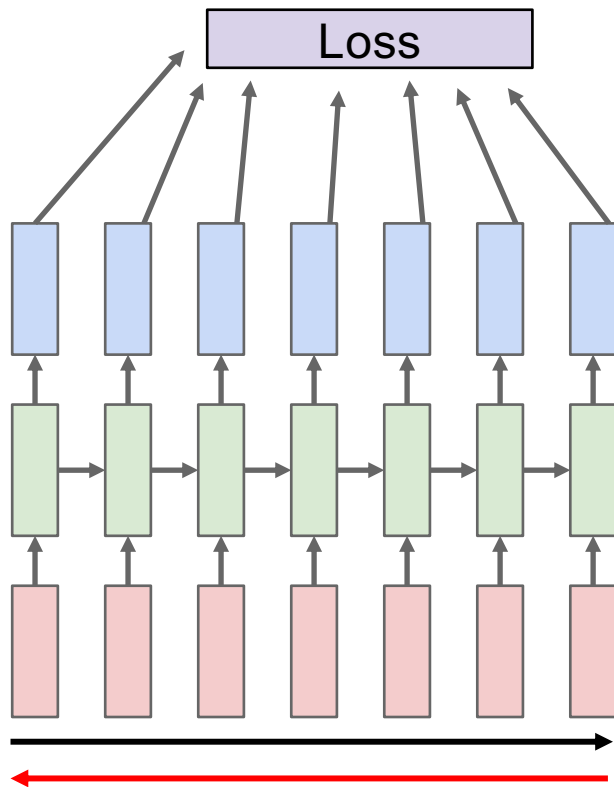
Source: [J. Johnson](#)

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

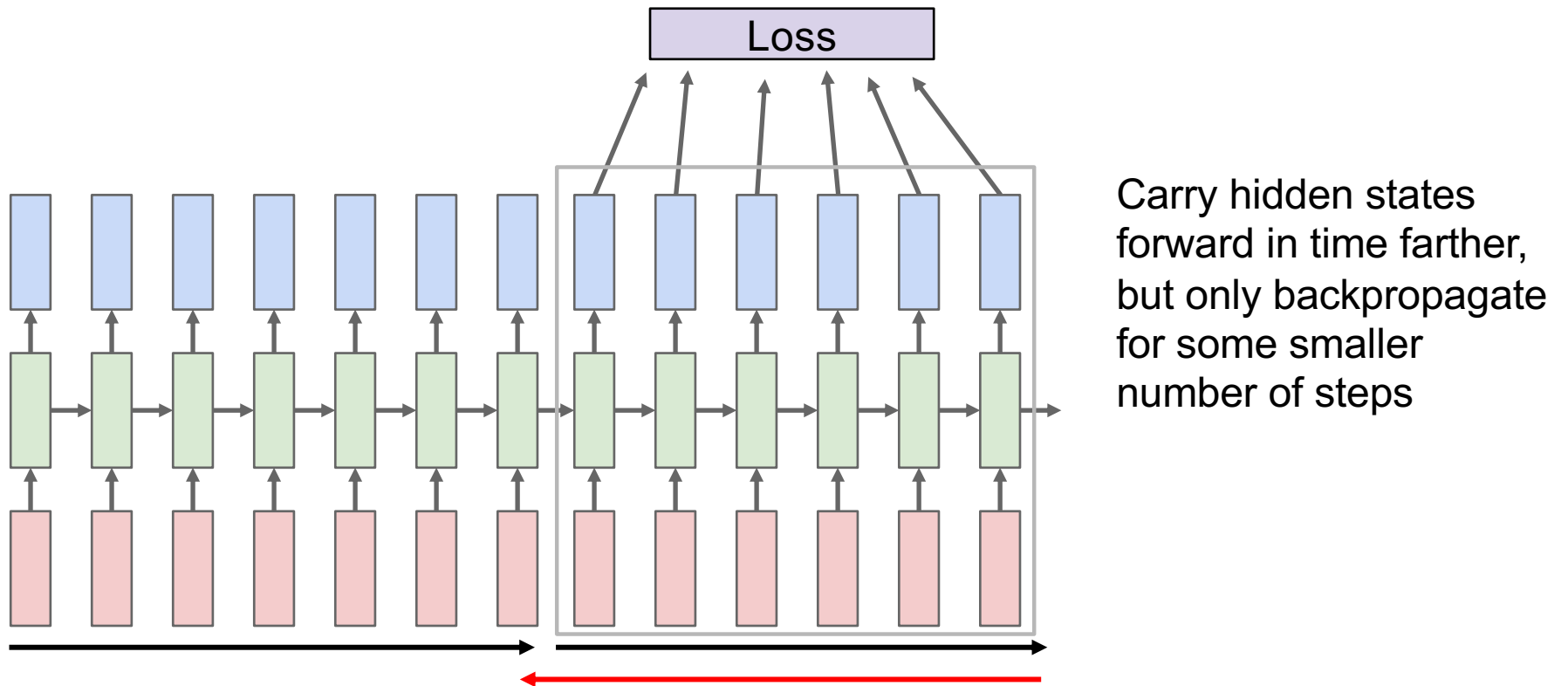
<https://machinelearningmastery.com/gentle-introduction-backpropagation-time/>
http://www.cs.utoronto.ca/~ilya/pubs/ilya_sutskever_phd_thesis.pdf

Truncated backpropagation through time

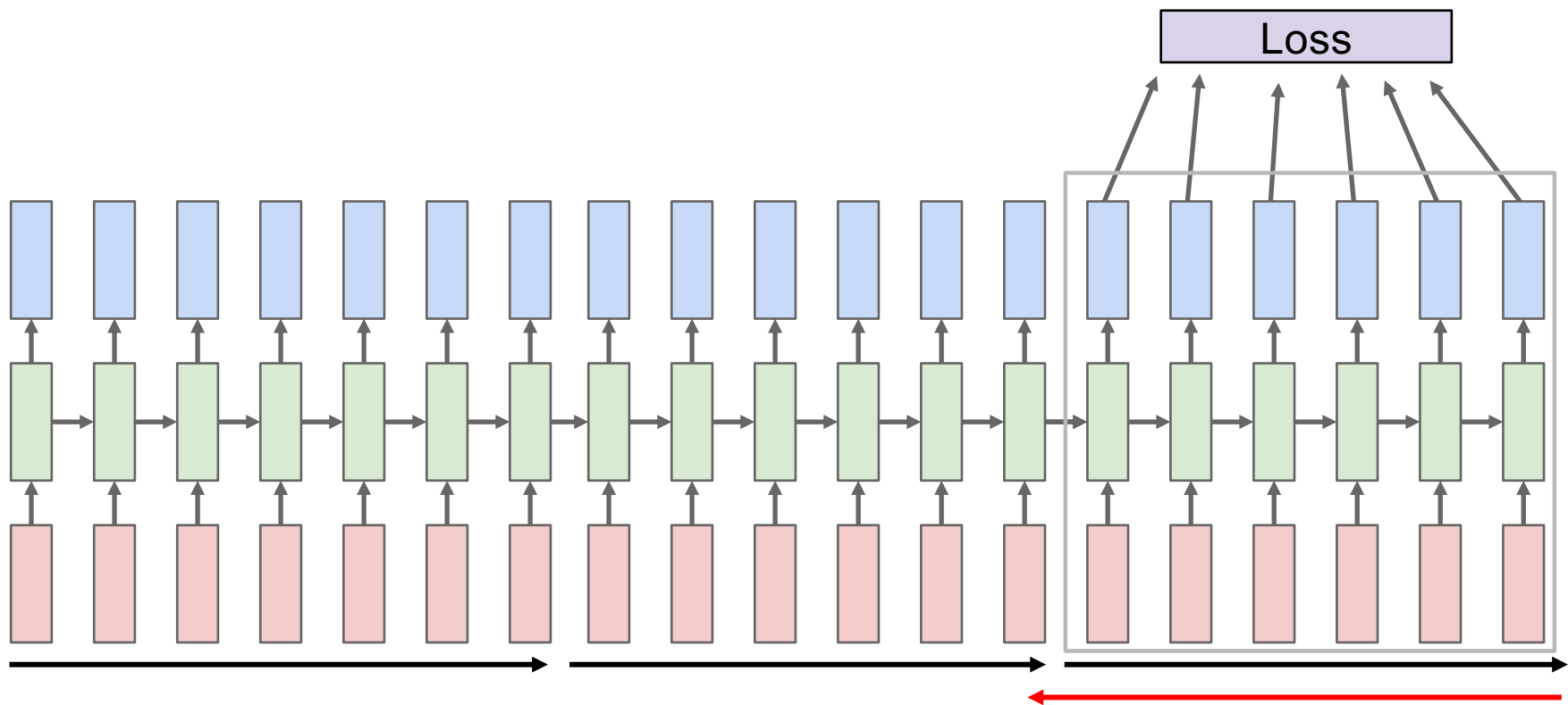


Run forward and backward through chunks of the sequence instead of whole sequence

Truncated backpropagation through time

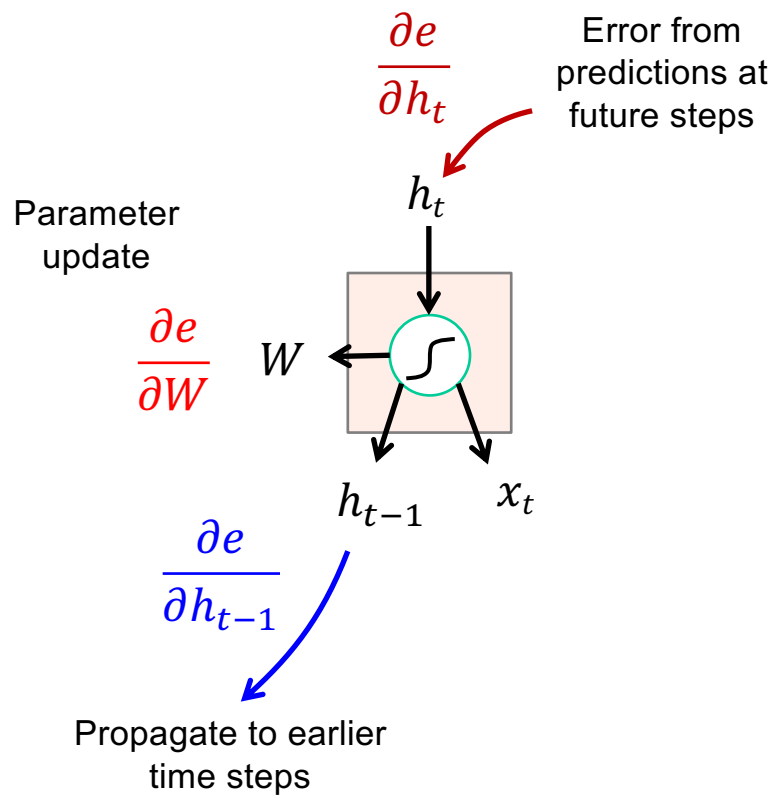


Truncated backpropagation through time



Source: [J. Johnson](#)

RNN backward pass



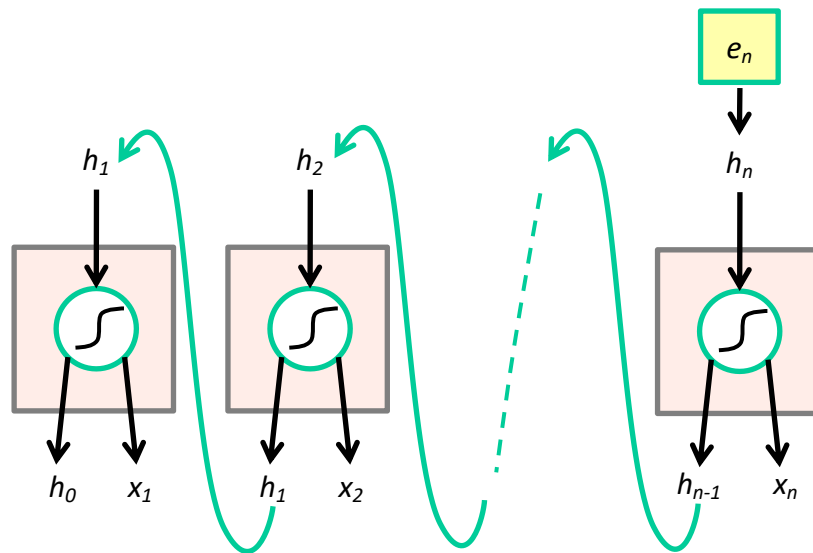
$$h_t = \tanh(W_x x_t + W_h h_{t-1})$$

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

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

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

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

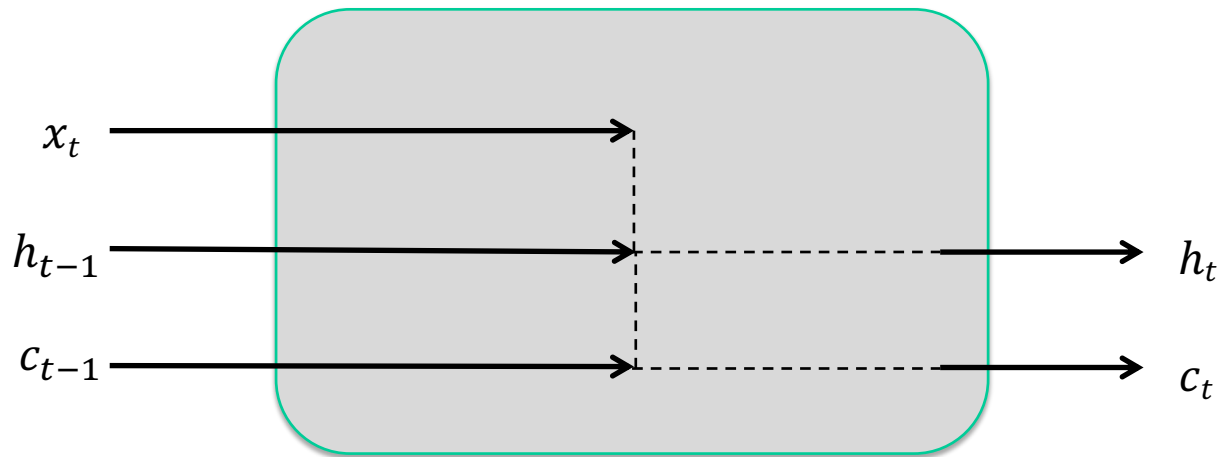
Gradients will *vanish* if largest singular value of W_h is less than 1 and *explode* if it's greater than 1

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)

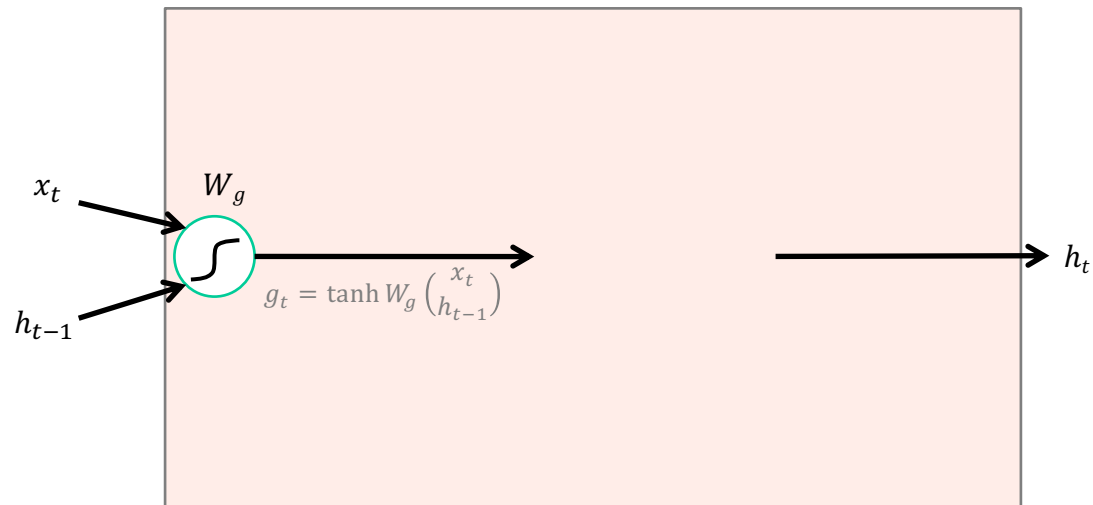
Long short-term memory (LSTM)

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

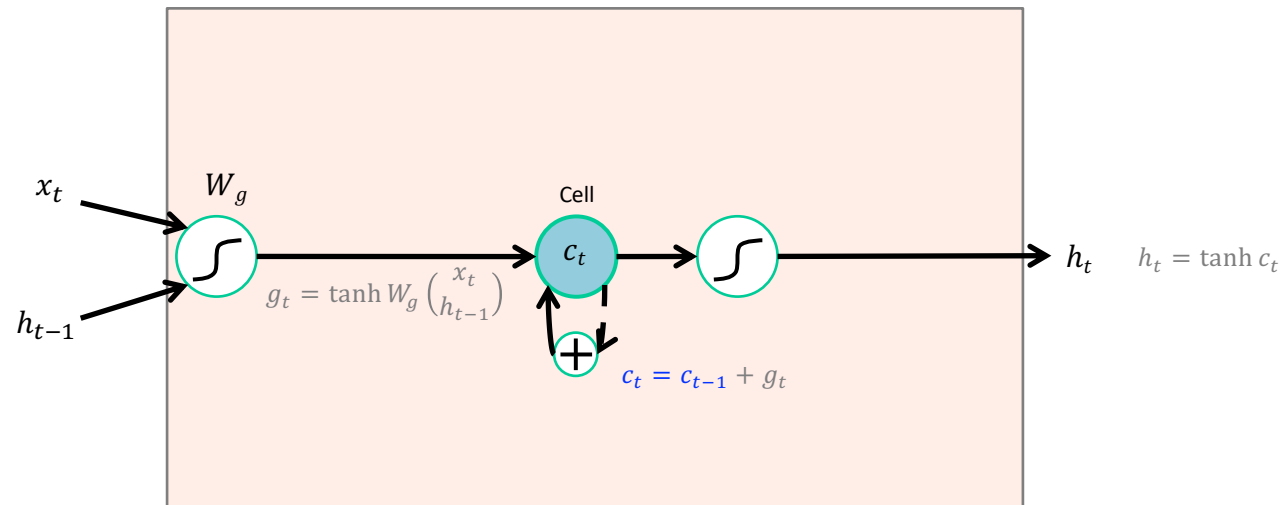


S. Hochreiter and J. Schmidhuber, [Long short-term memory](#), Neural Computation 9 (8), pp. 1735–1780, 1997

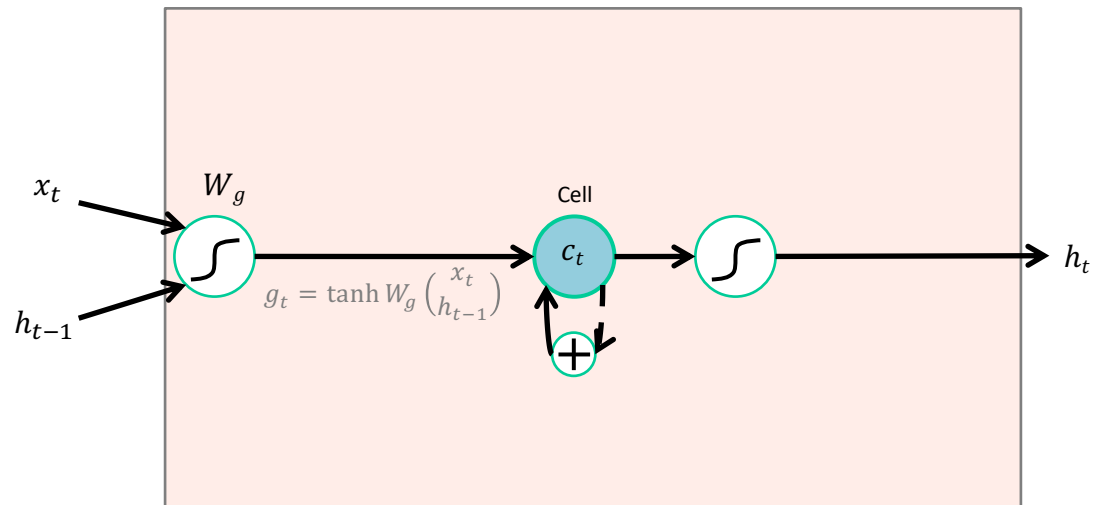
The LSTM cell



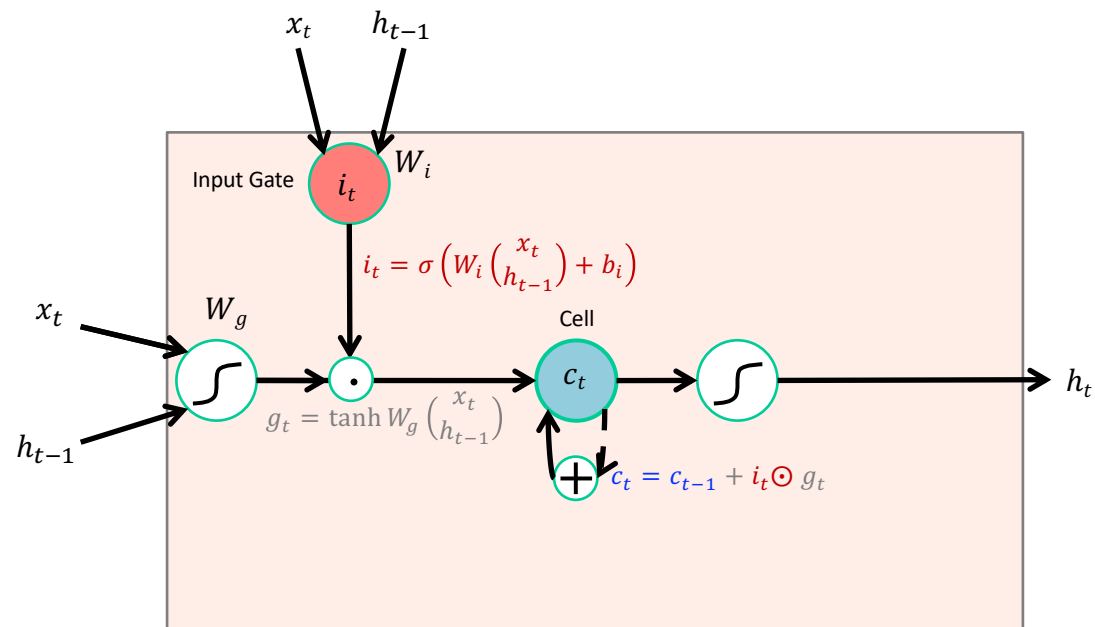
The LSTM cell



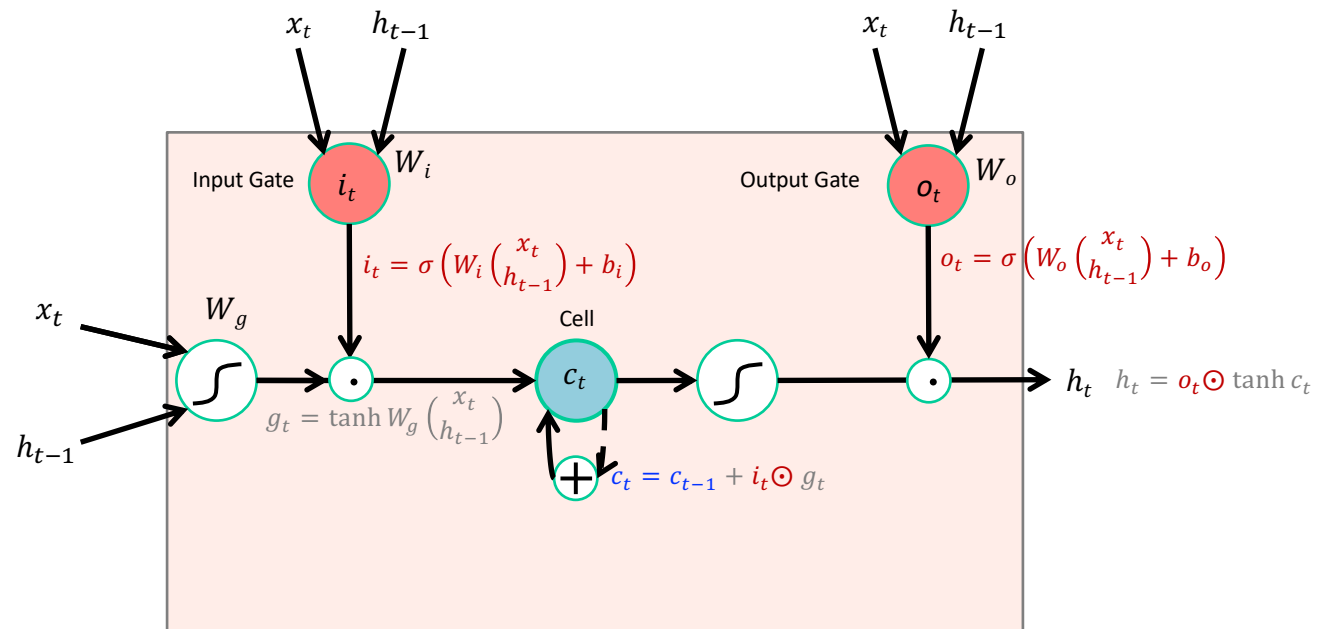
The LSTM cell



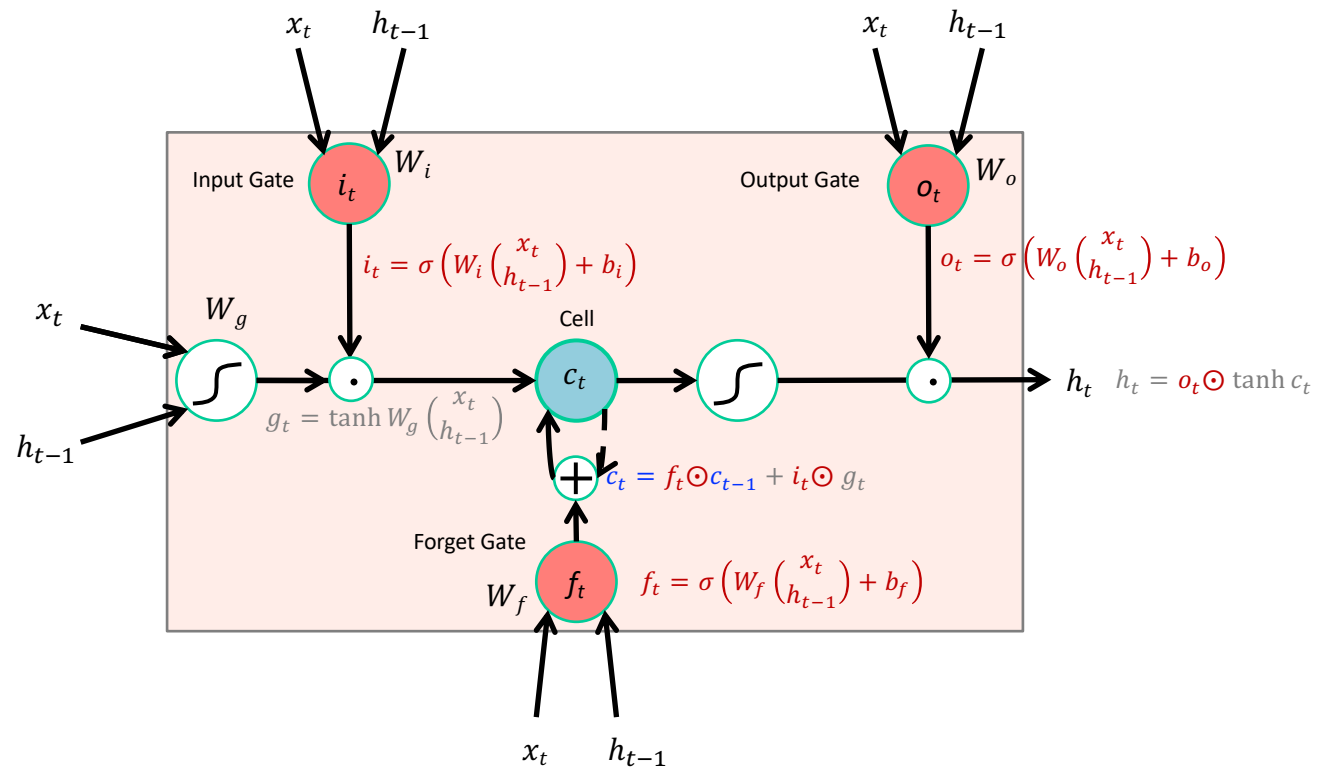
The LSTM cell



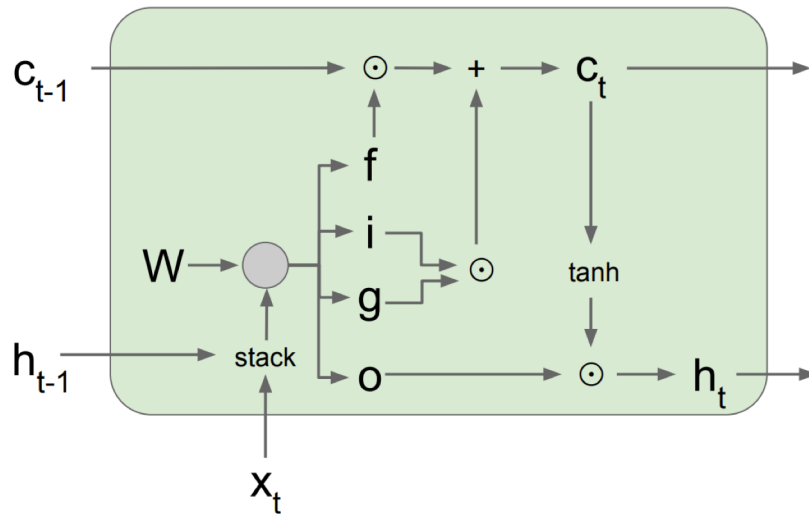
The LSTM cell



The LSTM cell



LSTM forward pass summary



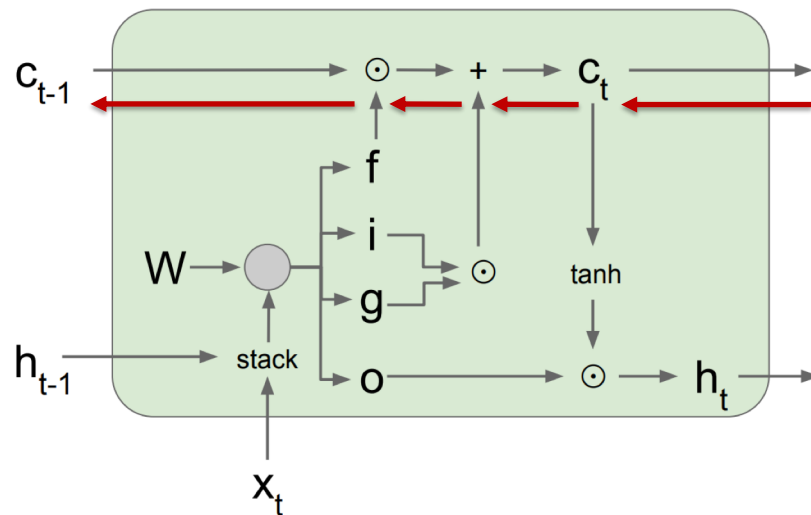
$$\begin{pmatrix} g_t \\ i_t \\ f_t \\ o_t \end{pmatrix} = \begin{pmatrix} \tanh \\ \sigma \\ \sigma \\ \sigma \end{pmatrix} \begin{pmatrix} W_g \\ W_i \\ W_f \\ W_o \end{pmatrix} \begin{pmatrix} x_t \\ h_{t-1} \end{pmatrix}$$

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

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

[Figure source](#)

LSTM backward pass

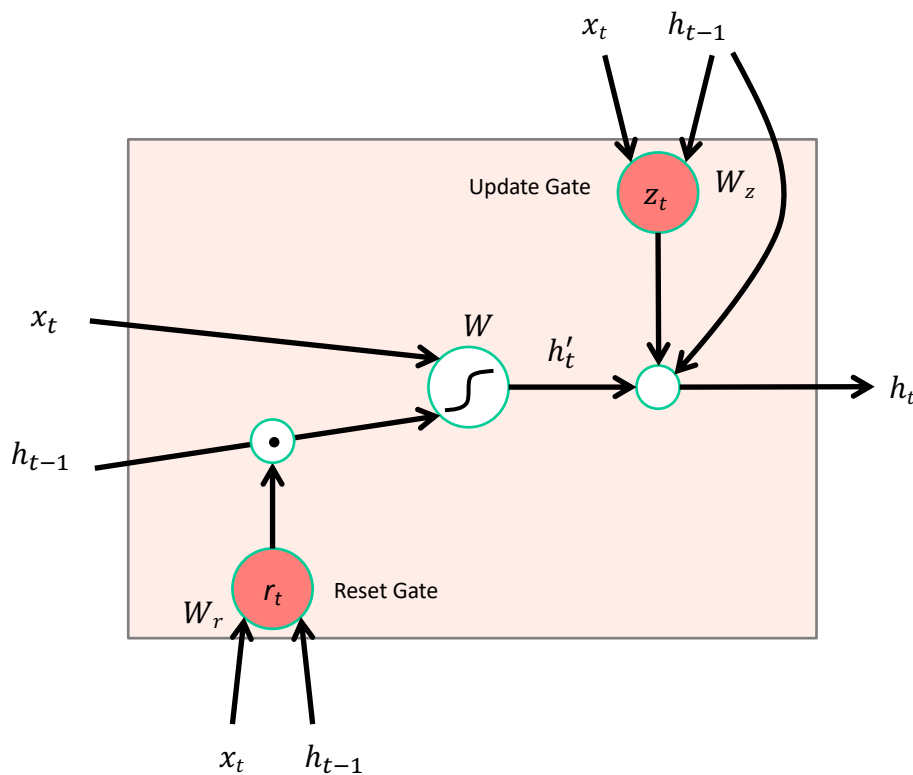


Gradient flow from c_t to c_{t-1} only involves back-propagating through addition and elementwise multiplication, not matrix multiplication or tanh

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

[Figure source](#)

LSTM variant: Gated recurrent unit (GRU)



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

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

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

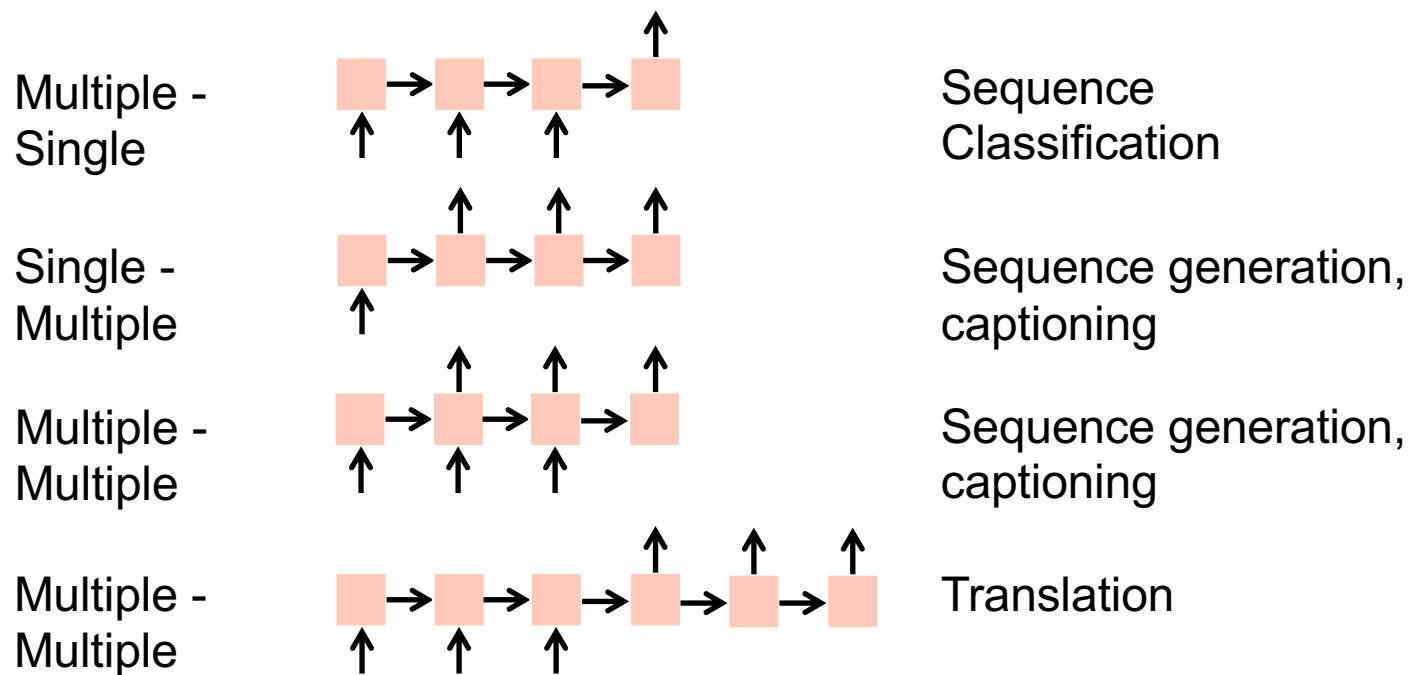
$$z_t = \sigma \left(W_z \begin{pmatrix} x_t \\ h_{t-1} \end{pmatrix} + b_z \right)$$

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

Outline

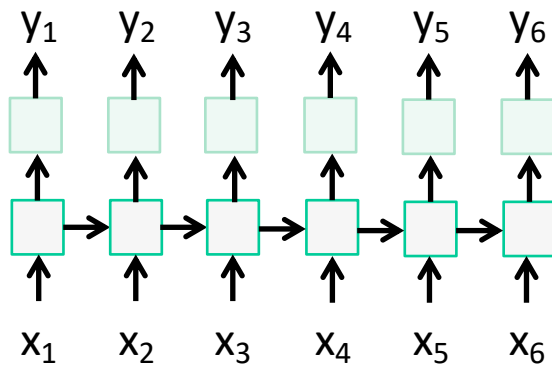
- Examples of sequential prediction tasks
- Common recurrent units
 - Vanilla RNN unit (and how to train it)
 - Long Short-Term Memory (LSTM)
 - Gated Recurrent Unit (GRU)
- Recurrent network architectures

Recall: Input-output scenarios



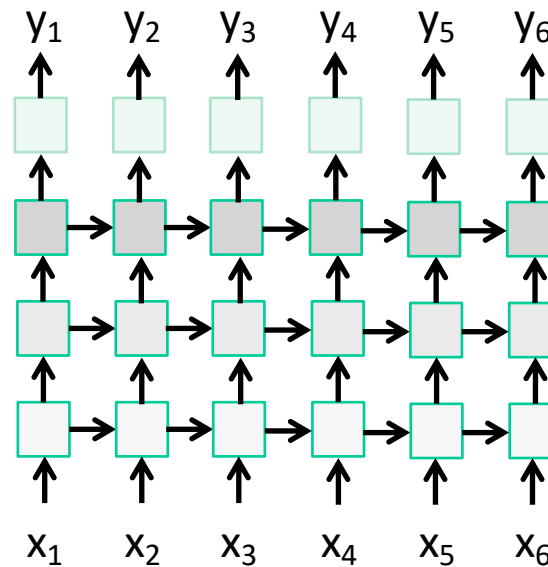
RNN architectures

- Most general configuration:



Multi-layer RNNs

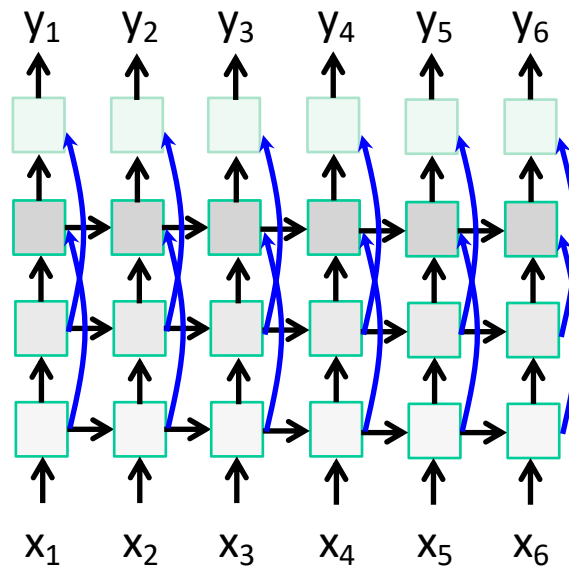
- We can of course design RNNs with multiple hidden layers



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

Multi-layer RNNs

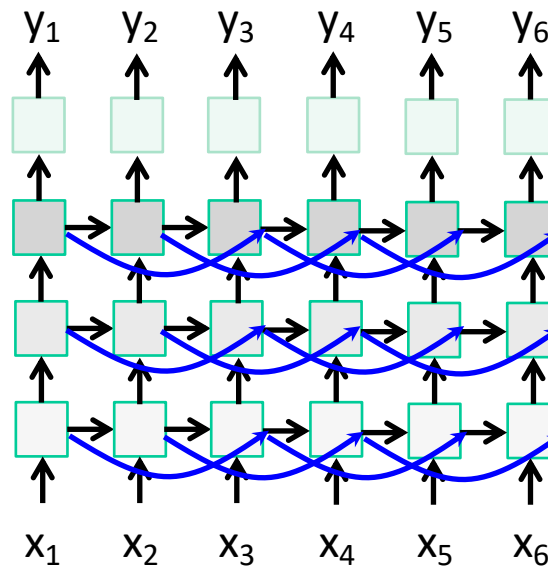
- We can of course design RNNs with multiple hidden layers



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

Multi-layer RNNs

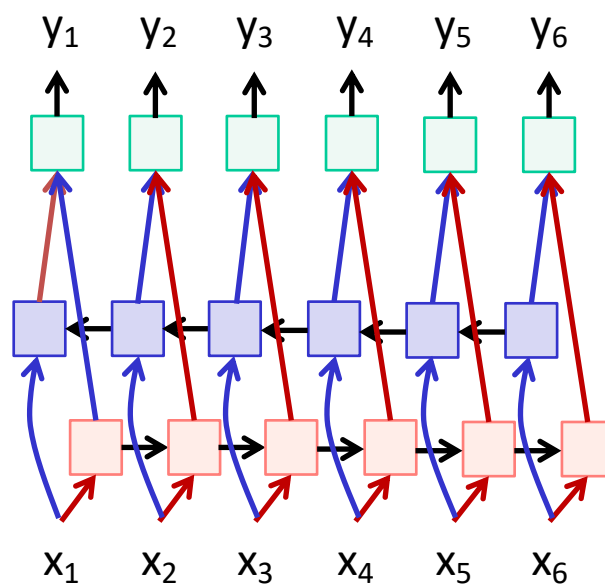
- We can of course design RNNs with multiple hidden layers



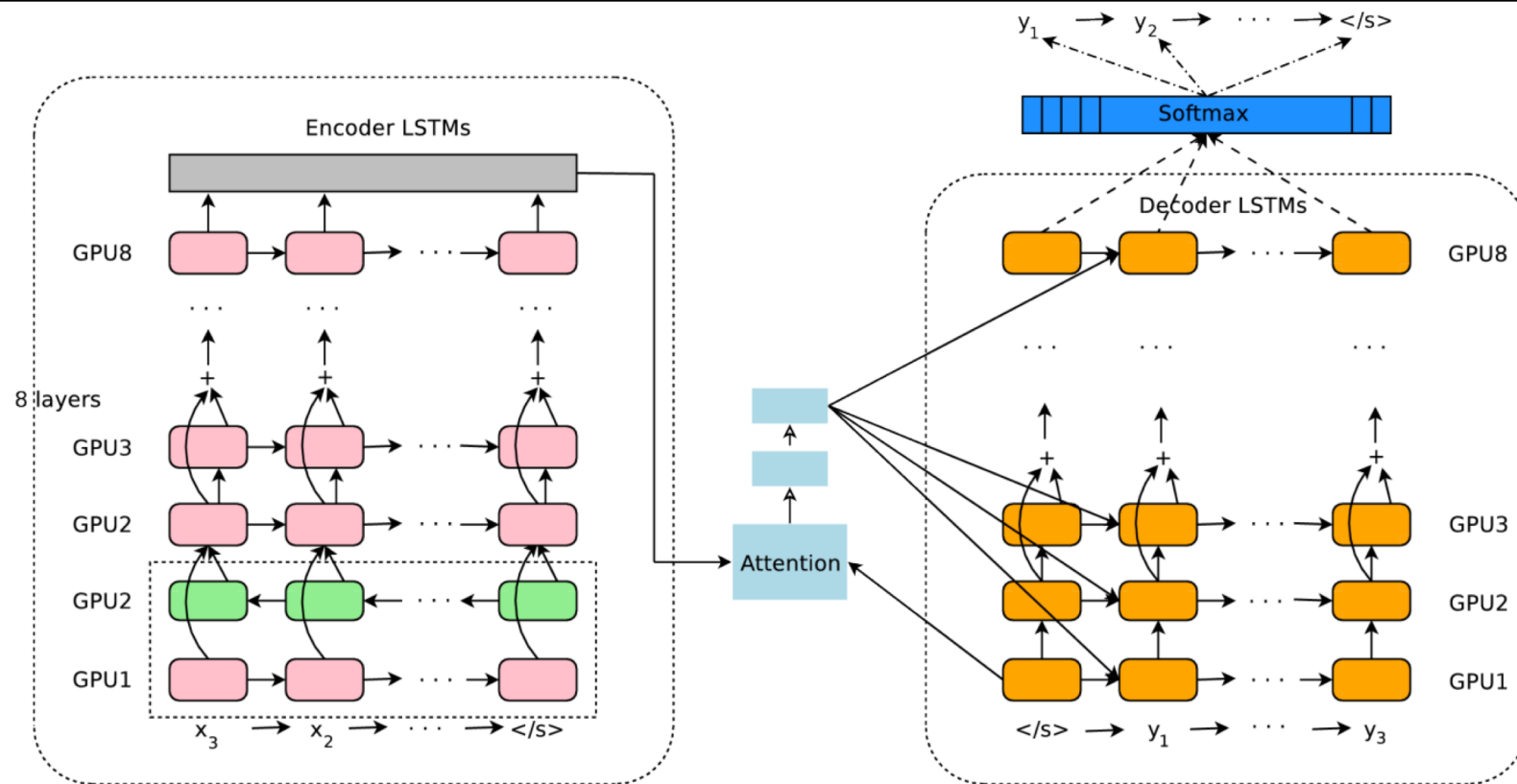
- Anything goes: skip connections across layers, across time, ...

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)

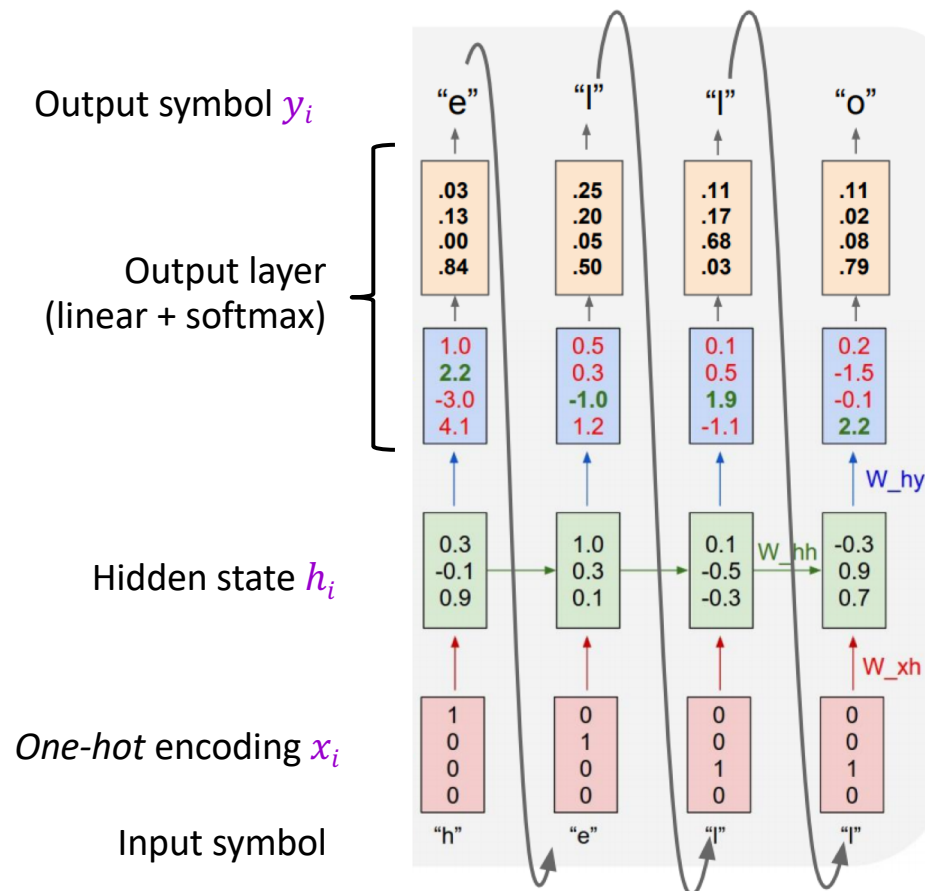


Y. Wu et al., [Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation](#), arXiv 2016

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
 - Language modeling
 - Image captioning
 - Machine translation

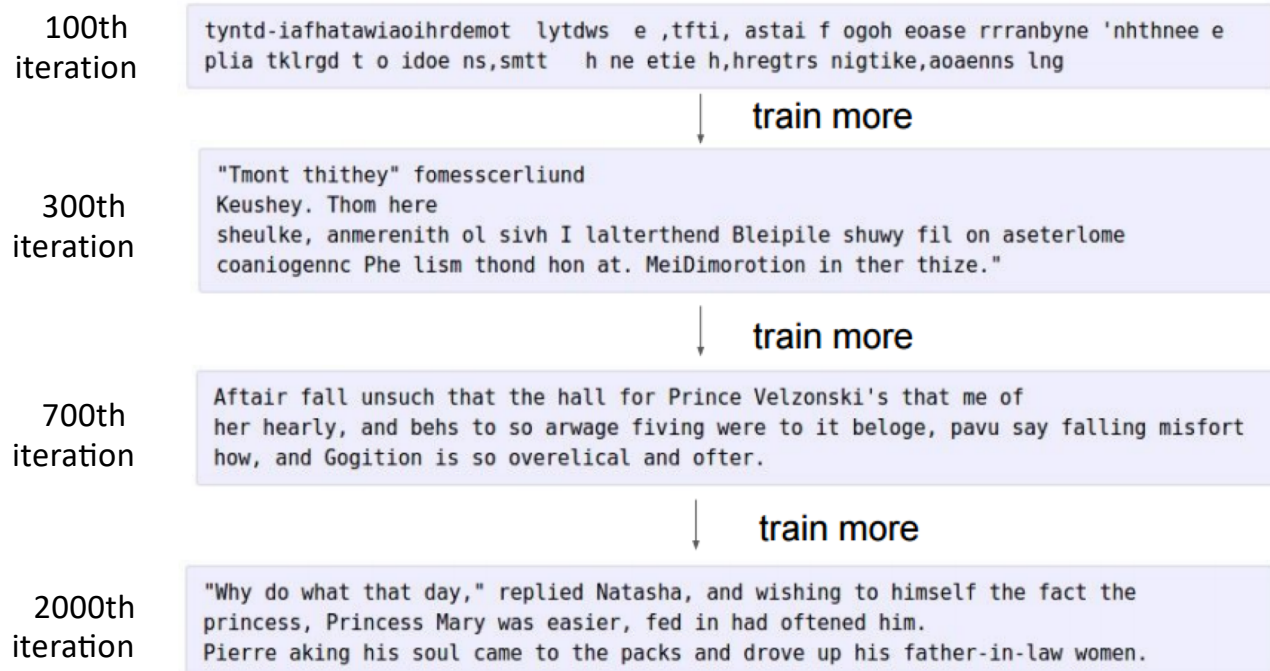
Language modeling: Character RNN



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

<http://karpathy.github.io/2015/05/21/rnn-effectiveness/>

Language modeling: Character RNN



<http://karpathy.github.io/2015/05/21/rnn-effectiveness/>

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

A. Karpathy, J. Johnson, and L. Fei-Fei, [Visualizing and Understanding Recurrent Networks](#), ICLR Workshop 2016

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

Searching for interpretable hidden units

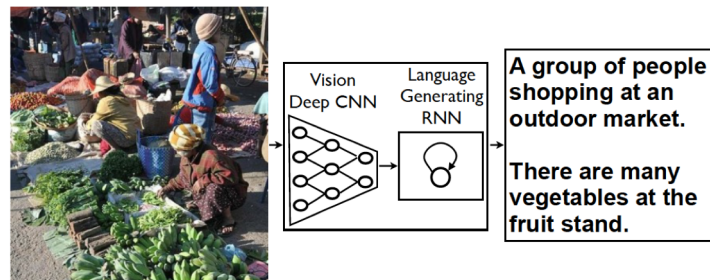
```
/* Unpack a filter field's string representation from user-space
 * buffer. */
char *audit_unpack_string(void **bufp, size_t *remain, size_t len)
{
    char *str;
    if (!*bufp || (len == 0) || (len > *remain))
        return ERR_PTR(-EINVAL);
    /* Of the currently implemented string fields, PATH_MAX
     * defines the longest valid length.
     */
}
```

(ツ)

Recurrent models: Outline

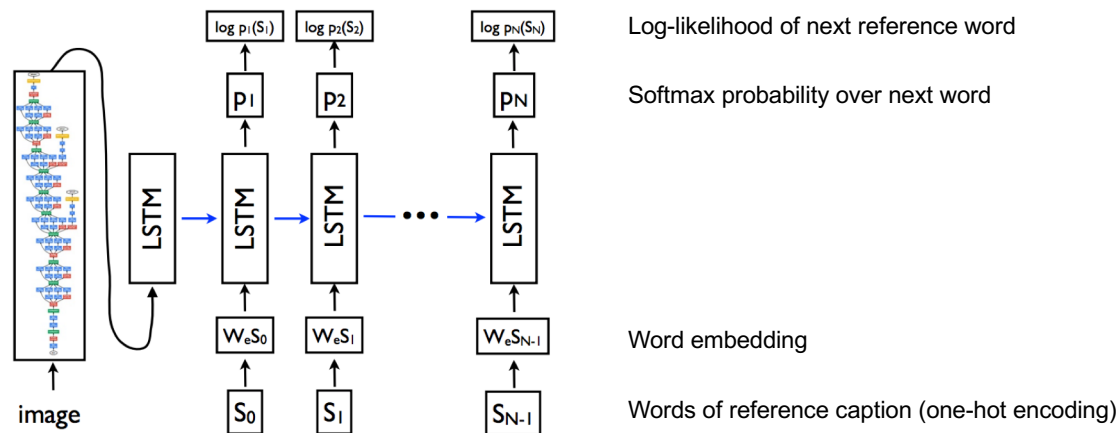
- Examples of sequential prediction tasks
- Common recurrent units
 - Vanilla RNN unit
 - Long Short-Term Memory (LSTM)
 - Gated Recurrent Unit (GRU)
- Recurrent network architectures
- Applications in (a bit) more detail
 - Language modeling
 - Image captioning

Image caption generation



Training time

- Maximize likelihood of reference captions



O. Vinyals, A. Toshev, S. Bengio, D. Erhan, [Show and Tell: A Neural Image Caption Generator](#), CVPR 2015

Image caption generation: Test time

- How do we produce a caption given a test image?
 - How about always choosing the highest-likelihood word?

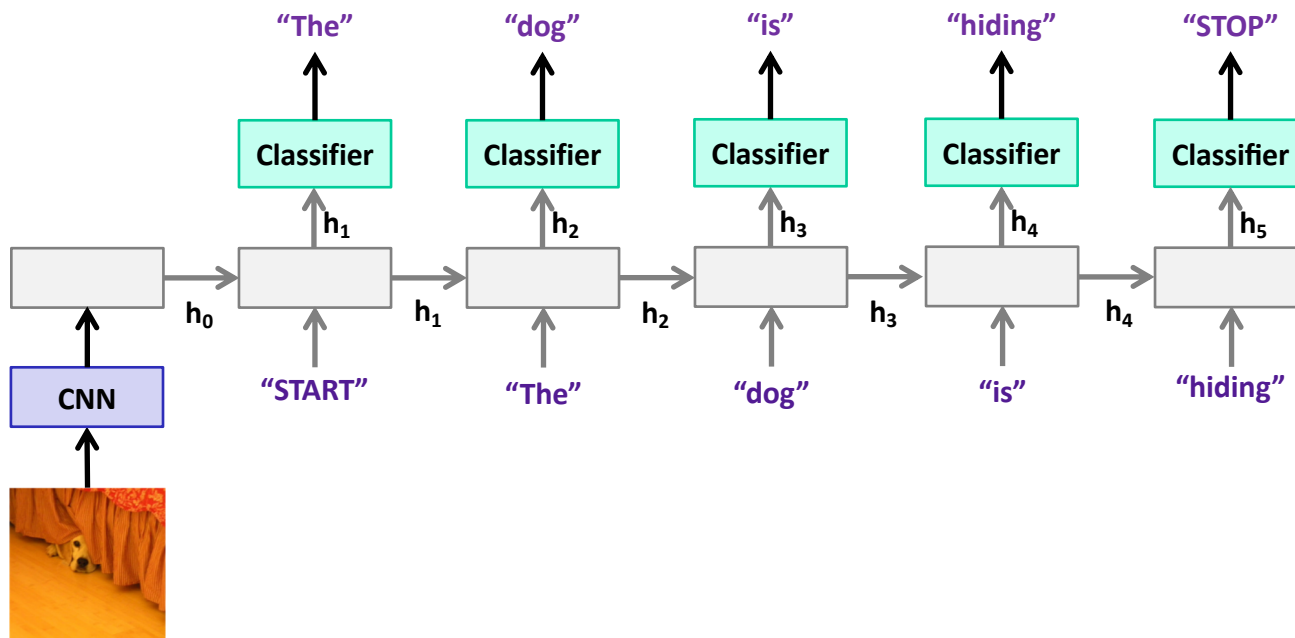


Image caption generation: Beam search

- 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

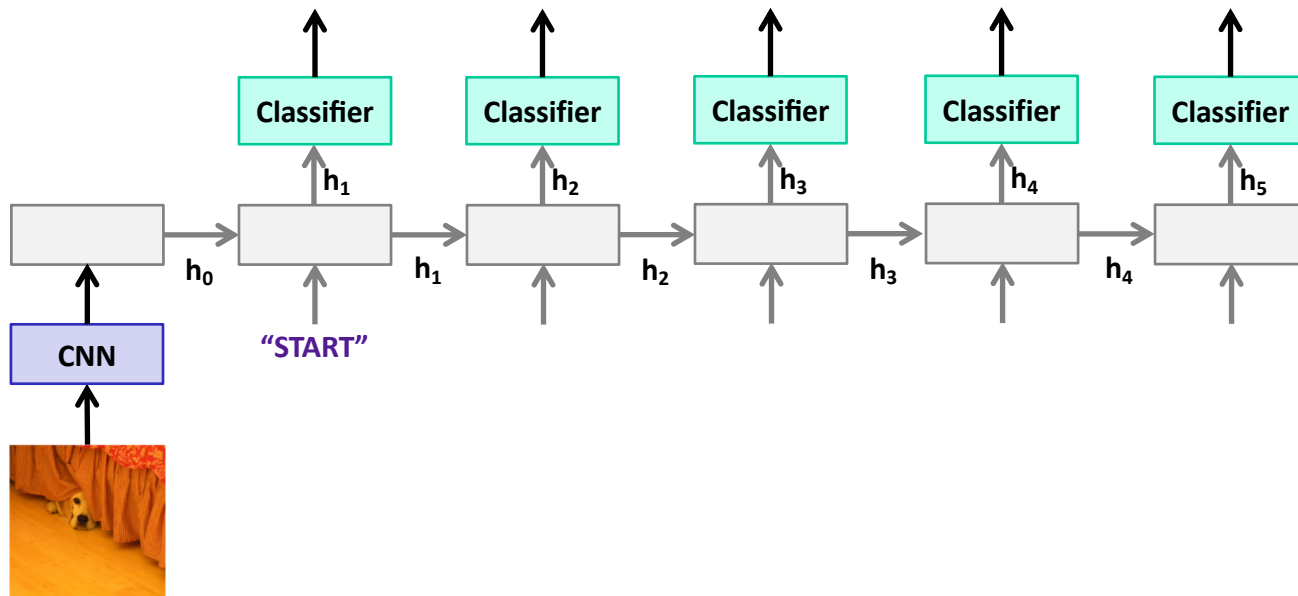


Image caption generation: Beam search

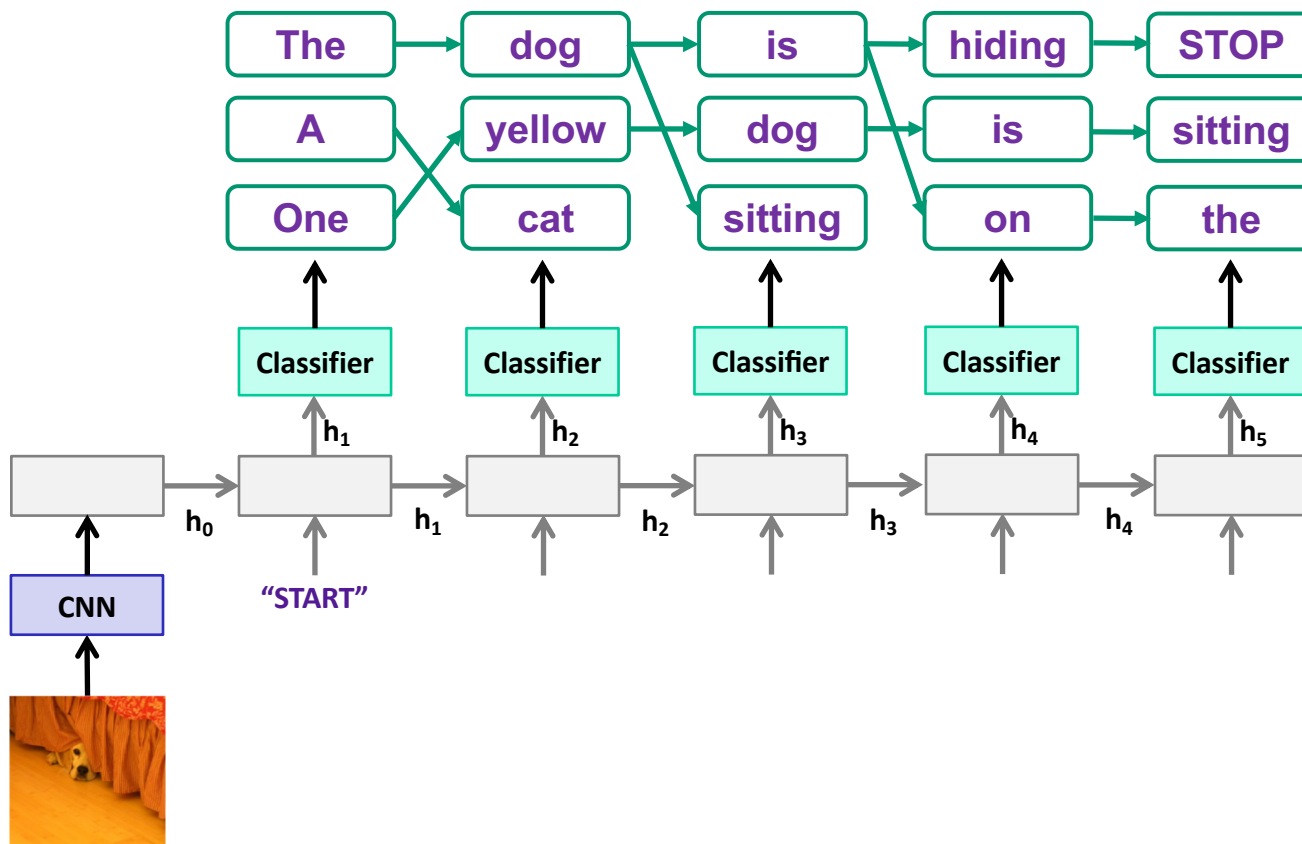


Image caption generation: Example outputs

<p>A person riding a motorcycle on a dirt road.</p> 	<p>Two dogs play in the grass.</p> 	<p>A skateboarder does a trick on a ramp.</p> 	<p>A dog is jumping to catch a frisbee.</p> 
<p>A group of young people playing a game of frisbee.</p> 	<p>Two hockey players are fighting over the puck.</p> 	<p>A little girl in a pink hat is blowing bubbles.</p> 	<p>A refrigerator filled with lots of food and drinks.</p> 
<p>A herd of elephants walking across a dry grass field.</p> 	<p>A close up of a cat laying on a couch.</p> 	<p>A red motorcycle parked on the side of the road.</p> 	<p>A yellow school bus parked in a parking lot.</p> 
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](#)

BLEU: Bilingual Evaluation Understudy

Original (French): J'ai mangé la pomme.

Reference translation: I ate the apple.

Based on BLEU, these are all “equally bad” output sentences.

I consumed the apple.

I ate an apple.

I ate the potato.

<https://towardsdatascience.com/evaluating-text-output-in-nlp-bleu-at-your-own-risk-e8609665a213>



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]	0.845	0.218	0.504	0.712	0.545	0.404	0.299
MSR Captiva	0.845	0.218	0.504	0.712	0.545	0.404	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

<http://mscoco.org/dataset/#captions-leaderboard>



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]	0.630	0.647	4.407	3.740	0.600
MSR ^[8]	M1	Percentage of captions that are evaluated as better or equal to human caption.			
Montreal ^[1]	M2	Percentage of captions that pass the Turing Test.			
MSR Ca ^[9]	M3	Average correctness of the captions on a scale 1-5 (incorrect - correct).			
Berkeley ^[10]	M4	Average amount of detail of the captions on a scale 1-5 (lack of details - very detailed).			
m-RNN ^[12]	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