

CS 4803 / 7643: Deep Learning

Topics:

- Recurrent Neural Networks (RNNs)
 - RNN visualizations
 - Image Captioning, Beam Search
 - LSTMs

Dhruv Batra
Georgia Tech

Administrativa

- HW3 Reminder
 - Due: 10/20 11:59pm
 - Theory: Convolutions, Representation Capacity, Double Descent
 - Implementation: Saliency methods (e.g. Grad-CAM) in Python and PyTorch/Captum
- HW2 grades coming soon

Administrativa

- Guest Lecture: Ishan Misra (FAIR)
 - Thurs 10/21
 - Self-Supervised Learning for Vision



<http://imisra.github.io/>

Administrativa

- Guest Lecture: Michael Auli (FAIR)
 - Tue 10/26
 - Self-Supervised Learning for Speech



<https://michaেলাuli.github.io/>

Administrativa

- Guest Lecture: Arjun Majumdar
 - Thurs 10/28
 - Transformers, BERT, ViLBERT

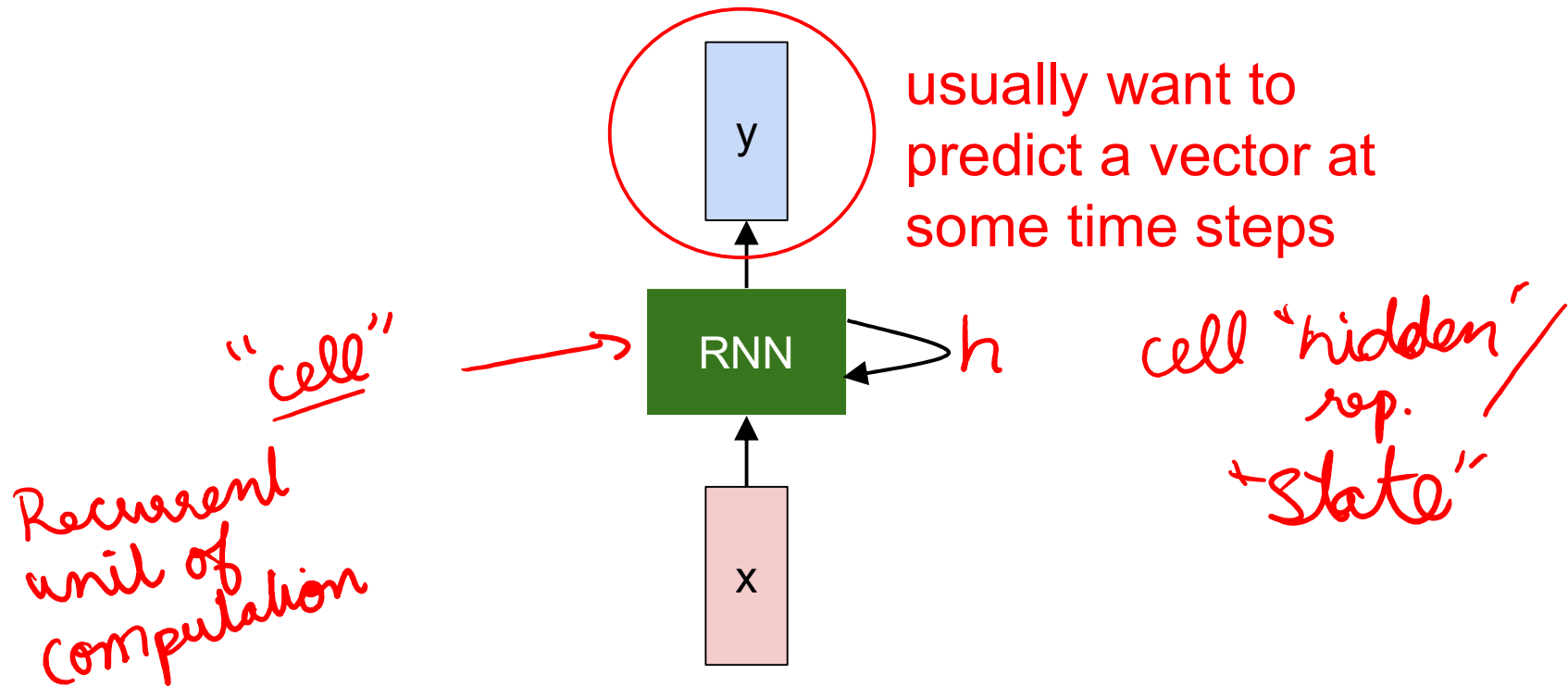


<https://arjunmajum.github.io/>



Recap from last time

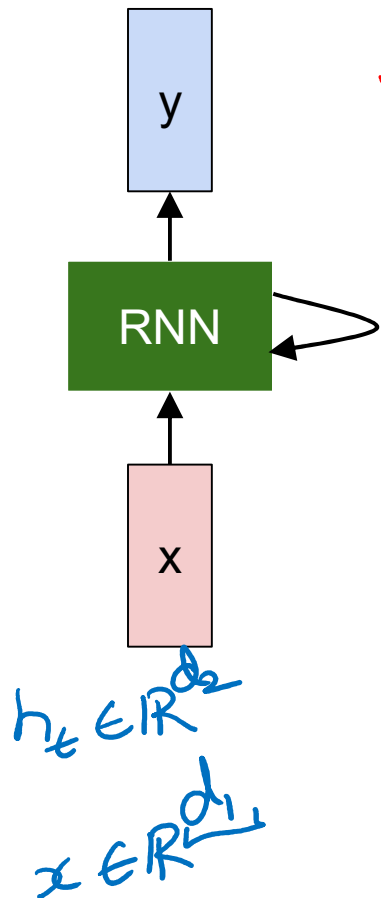
Recurrent Neural Network



$$P(y_t | x_1 \dots x_t) \approx P(y_t | h_t)$$

(Vanilla) Recurrent Neural Network

The state consists of a single "hidden" vector h :



"Scores" $\vec{y}_t = \text{Softmax}(W_{hy} \vec{h}_t + \vec{b}_y)$

$$\vec{h}_t = f_W(\vec{h}_{t-1}, \vec{x}_t)$$

$\frac{512}{1R} \Rightarrow \vec{h}_t = \tanh(W_{hh} \vec{h}_{t-1} + W_{xh} \vec{x}_t + \vec{b}_h)$

$\begin{bmatrix} W_{hh} & W_{xn} \end{bmatrix} \begin{bmatrix} \vec{h}_{t-1} \\ \vec{x}_t \end{bmatrix} + \vec{b}_n$

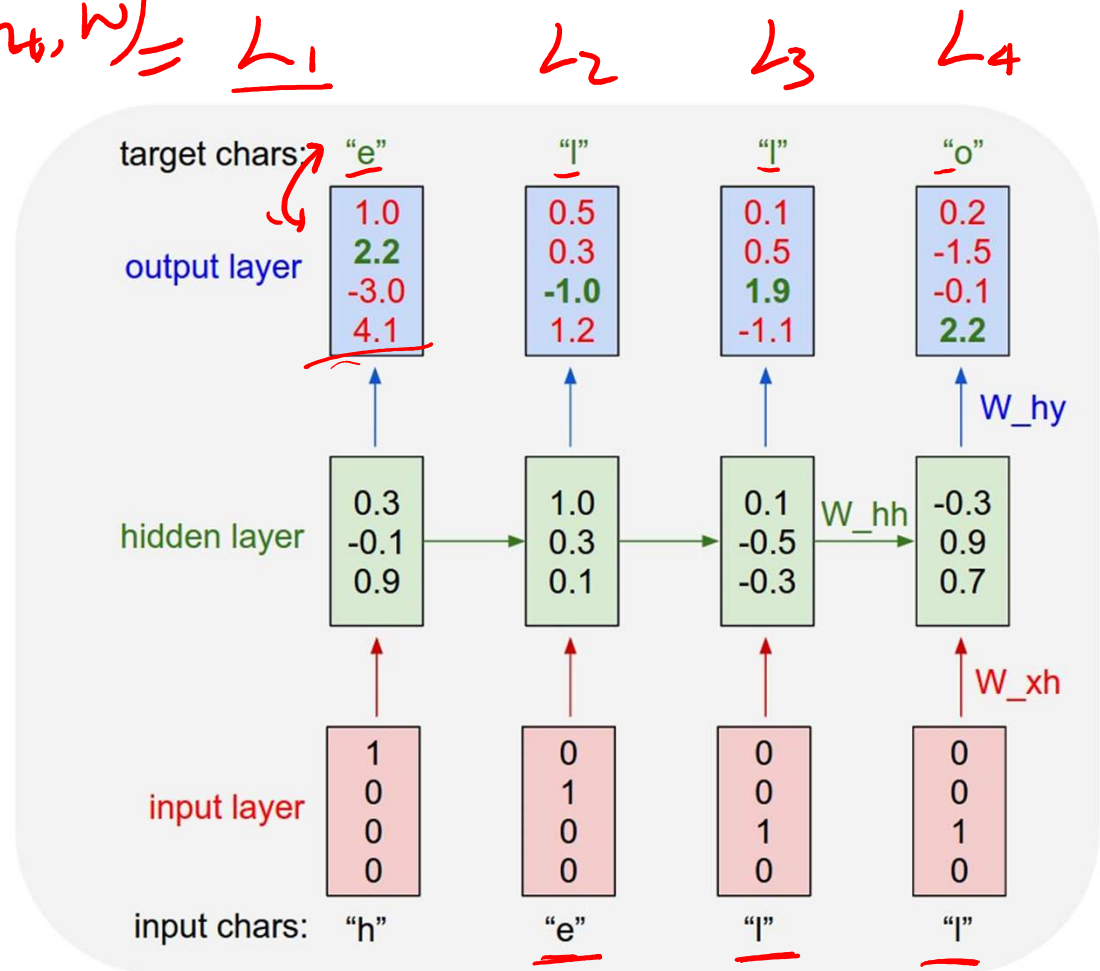
Sometimes called a "Vanilla RNN" or an "Elman RNN" after Prof. Jeffrey Elman
 Slide Credit: Fei-Fei Li, Justin Johnson, Serena Yeung, CS 231n

$$\min_{\vec{w}} \sum_t -\log P(y_t^{gt} | h_t, \vec{w}) = \mathcal{L}_1$$

Example: Character-level Language Model

Vocabulary:
[h,e,l,o]

Example training
sequence:
"hello"



$$x_2 = y_1^{gt} \quad x_3 = y_2^{gt} \quad \dots$$

Training Time: MLE / “Teacher Forcing”

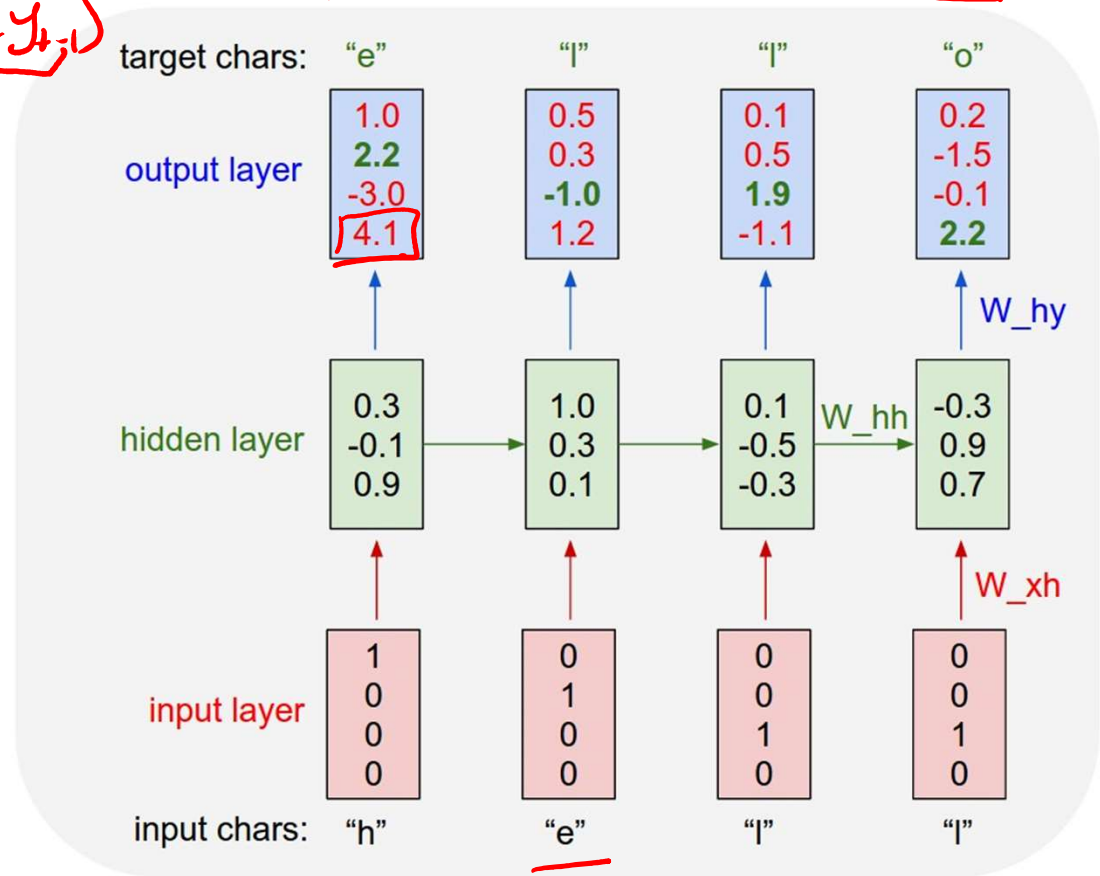
$$P(y_1 \dots y_T) = \prod_t P(y_t | y_{1:t-1})$$

Example:
Character-level
Language Model



Vocabulary:
 [h,e,l,o]

Example training
 sequence:
 “hello”



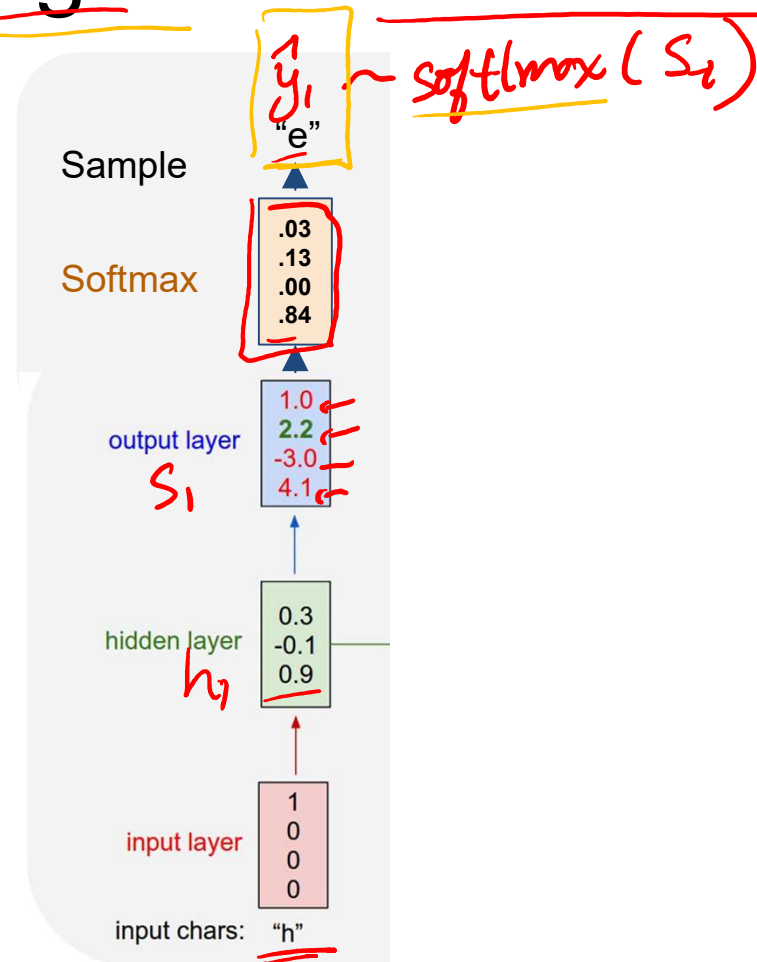
$$x_2 = y_1^{36}$$

Test Time: Sample / Argmax / Beam Search

Example: Character-level Language Model Sampling

Vocabulary:
[h,e,l,o]

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

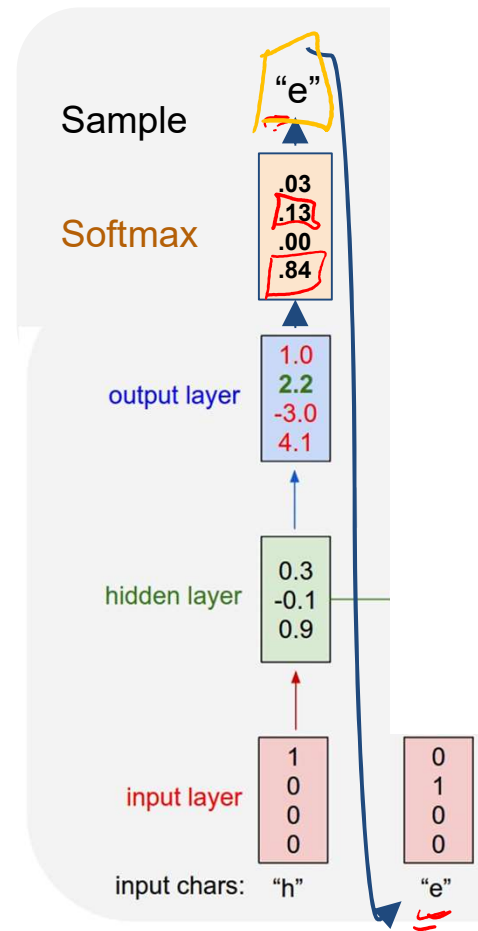


Test Time: Sample / Argmax / Beam Search

Example: Character-level Language Model Sampling

Vocabulary:
[h,e,l,o]

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

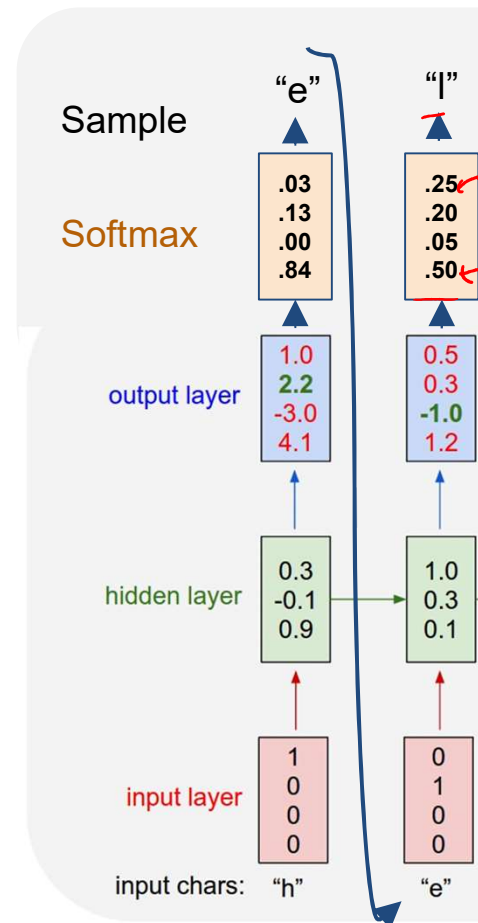


Test Time: Sample / Argmax / Beam Search

Example: Character-level Language Model Sampling

Vocabulary:
[h,e,l,o]

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

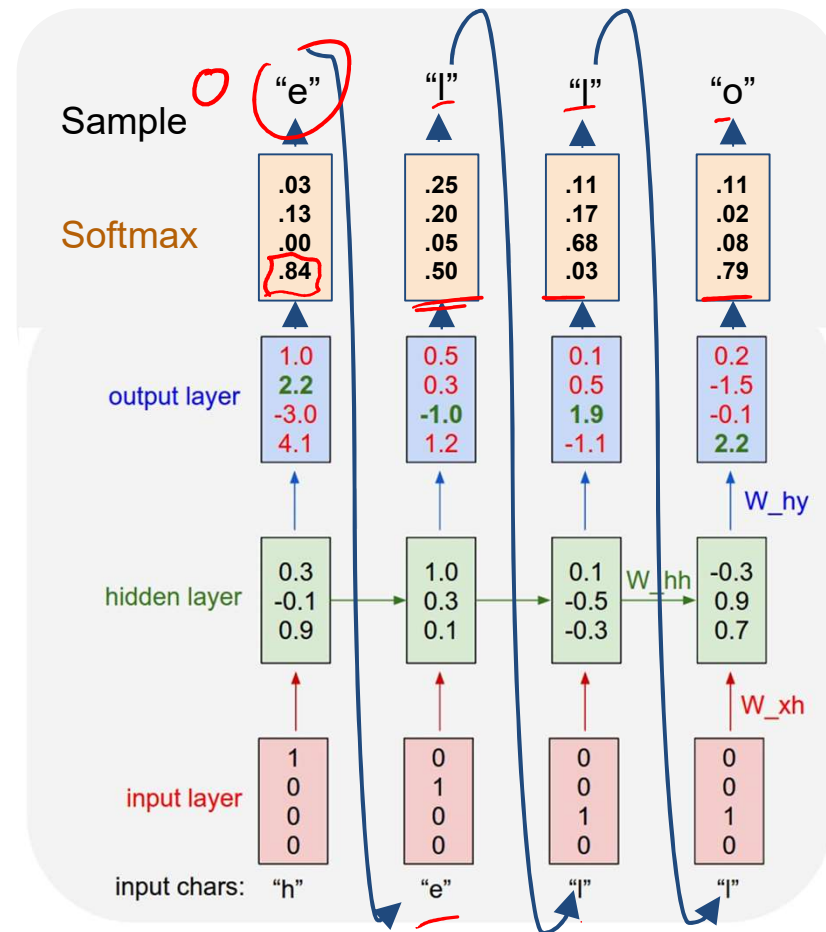


Test Time: Sample / Argmax / Beam Search

Example: Character-level Language Model Sampling

Vocabulary:
[h,e,l,o]

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



h o

Plan for Today

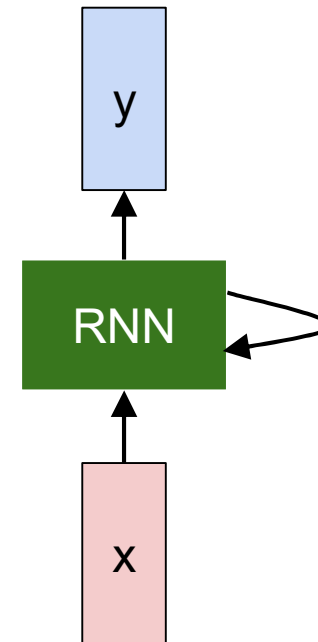
- Recurrent Neural Networks (RNNs)
 - (Finish) Visualization in Character RNNs
 - Inference: Beam Search
 - Example: Image Captioning
 - Multilayer RNNs
 - Problems with gradients in “vanilla” RNNs
 - LSTMs (and other RNN variants)

THE SONNETS

by William Shakespeare

From fairest creatures we desire increase,
That thereby beauty's rose might never die,
But as the ripper should by time decease,
His tender heir might bear his memory:
But thou, contracted to thine own bright eyes,
Feed'st thy light's flame with self-substantial fuel,
Making a famine where abundance lies,
Thyself thy foe, to thy sweet self too cruel:
Thou that art now the world's fresh ornament,
And only herald to the gaudy spring,
Within thine own bud buriest thy content,
And tender churl mak'st waste in niggarding:
Pity the world, or else this glutton be,
To eat the world's due, by the grave and thee.

When forty winters shall besiege thy brow,
And dig deep trenches in thy beauty's field,
Thy youth's proud livery so gazed on now,
Will be a tatter'd weed of small worth held:
Then being asked, where all thy beauty lies,
Where all the treasure of thy lusty days;
To say, within thine own deep sunken eyes,
Were an all-eating shame, and thriftless praise.
How much more praise deserv'd thy beauty's use,
If thou couldst answer 'This fair child of mine
Shall sum my count, and make my old excuse,'
Proving his beauty by succession thine!
This were to be new made when thou art old,
And see thy blood warm when thou feel'st it cold.



at first:

tyntd-iafhatawiao

rdemot 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 offer.

↓ train more

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

torvalds / linux

Watch - 3,711 | Star 23,054 | Fork 9,141

Linux kernel source tree

520,037 commits | 1 branch | 420 releases | 5,039 contributors

branch: master - linux / +

Merge branch 'drm-fixes' of git://people.freedesktop.org/~airlied/linux

torvalds	authored 9 hours ago	latest commit 4b1706927d
Documentation	Merge git://git.kernel.org/pub/scm/linux/kernel/git/hab/target-pending	6 days ago
arch	Merge branch 'x86-urgent-for-linus' of git://git.kernel.org/pub/scm/...	a day ago
block	block: discard bdi_unregister() in favour of bdi_destroy()	9 days ago
crypto	Merge git://git.kernel.org/pub/scm/linux/kernel/git/herbert/crypto-2.6	10 days ago
drivers	Merge branch 'drm-fixes' of git://people.freedesktop.org/~airlied/linux	9 hours ago
firmware	firmware/ihex2fw.c: restore missing default in switch statement	2 months ago
fs	vfs: read file_handle only once in handle_to_path	4 days ago
include	Merge branch 'perf-urgent-for-linus' of git://git.kernel.org/pub/scm/...	a day ago
init	init: fix regression by supporting devices with major:minor:offset fo...	a month ago
io	Merge branch 'for-linus' of git://git.kernel.org/pub/scm/linux/kernel/...	a month ago

Code | Pull requests (74) | Pulse | Graphs

HTTPS clone URL: `https://github.c`

You can clone with [HTTPS](#), [SSH](#), or [Subversion](#).

Clone in Desktop | Download ZIP

```

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 & 0x00000000ffffffff8) & 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 ");
}

```

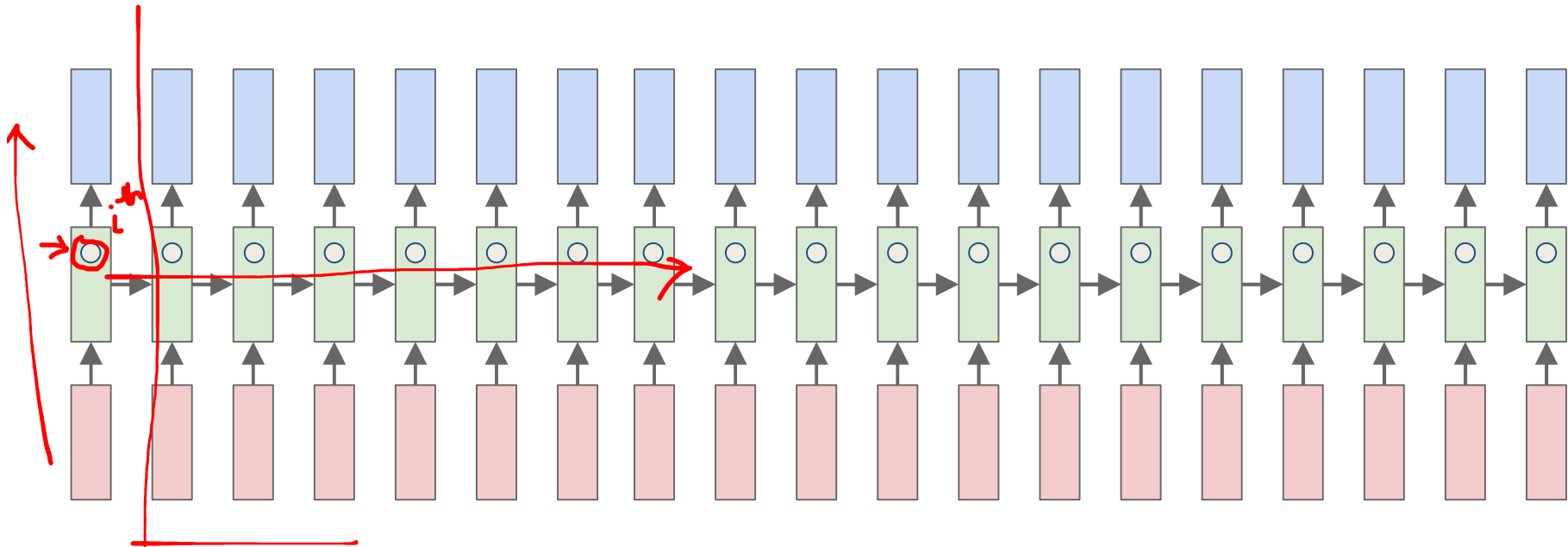
Generated C code

```
/*
 * Copyright (c) 2006-2010, Intel Mobile Communications. All rights reserved.
 *
 * This program is free software; you can redistribute it and/or modify it
 * under the terms of the GNU General Public License version 2 as published by
 * the Free Software Foundation.
 *
 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
 *
 * GNU General Public License for more details.
 *
 * You should have received a copy of the GNU General Public License
 * along with this program; if not, write to the Free Software Foundation,
 * Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
 */

#include <linux/kexec.h>
#include <linux/errno.h>
#include <linux/io.h>
#include <linux/platform_device.h>
#include <linux/multi.h>
#include <linux/ckevent.h>

#include <asm/io.h>
#include <asm/prom.h>
#include <asm/e820.h>
#include <asm/system_info.h>
#include <asm/setew.h>
#include <asm/pgproto.h>
```

Searching for interpretable cells



Karpathy, Johnson, and Fei-Fei: Visualizing and Understanding Recurrent Networks, ICLR Workshop 2016

Searching for interpretable cells

h_t []

\vec{h}_t

```
/* 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.
     */
}
```

Karpathy, Johnson, and Fei-Fei: Visualizing and Understanding Recurrent Networks, ICLR Workshop 2016

Figures copyright Karpathy, Johnson, and Fei-Fei, 2015; reproduced with permission

Searching for interpretable cells

"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]

Karpathy, Johnson, and Fei-Fei: Visualizing and Understanding Recurrent Networks, ICLR Workshop 2016

Figures copyright Karpathy, Johnson, and Fei-Fei, 2015; reproduced with permission

Searching for interpretable cells

Cell sensitive to position in line:

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 length tracking cell

Karpathy, Johnson, and Fei-Fei: Visualizing and Understanding Recurrent Networks, ICLR Workshop 2016

Figures copyright Karpathy, Johnson, and Fei-Fei, 2015; reproduced with permission

Searching for interpretable cells

```
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

Karpathy, Johnson, and Fei-Fei: Visualizing and Understanding Recurrent Networks, ICLR Workshop 2016

Figures copyright Karpathy, Johnson, and Fei-Fei, 2015; reproduced with permission

Searching for interpretable cells

Cell that turns on inside comments and quotes:

```
/* 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;
}
```

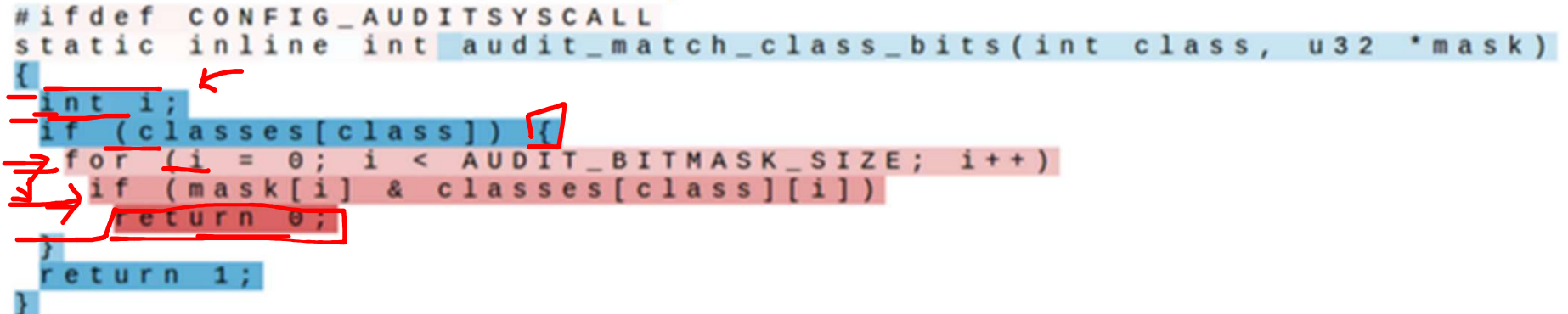
quote/comment cell

Karpathy, Johnson, and Fei-Fei: Visualizing and Understanding Recurrent Networks, ICLR Workshop 2016

Figures copyright Karpathy, Johnson, and Fei-Fei, 2015; reproduced with permission

Searching for interpretable cells

```
#ifdef CONFIG_AUDIT_SYSCALL
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;
}
```

The code snippet shows a function `audit_match_class_bits` that checks if a class matches a mask. Red annotations highlight the loop and the `return 0;` statement. A red arrow points to the `int i;` declaration. A red box highlights the `if (classes[class]) {` condition. A red arrow points to the `for` loop. A red box highlights the `return 0;` statement. A red arrow points to the `return 1;` statement.

code depth cell

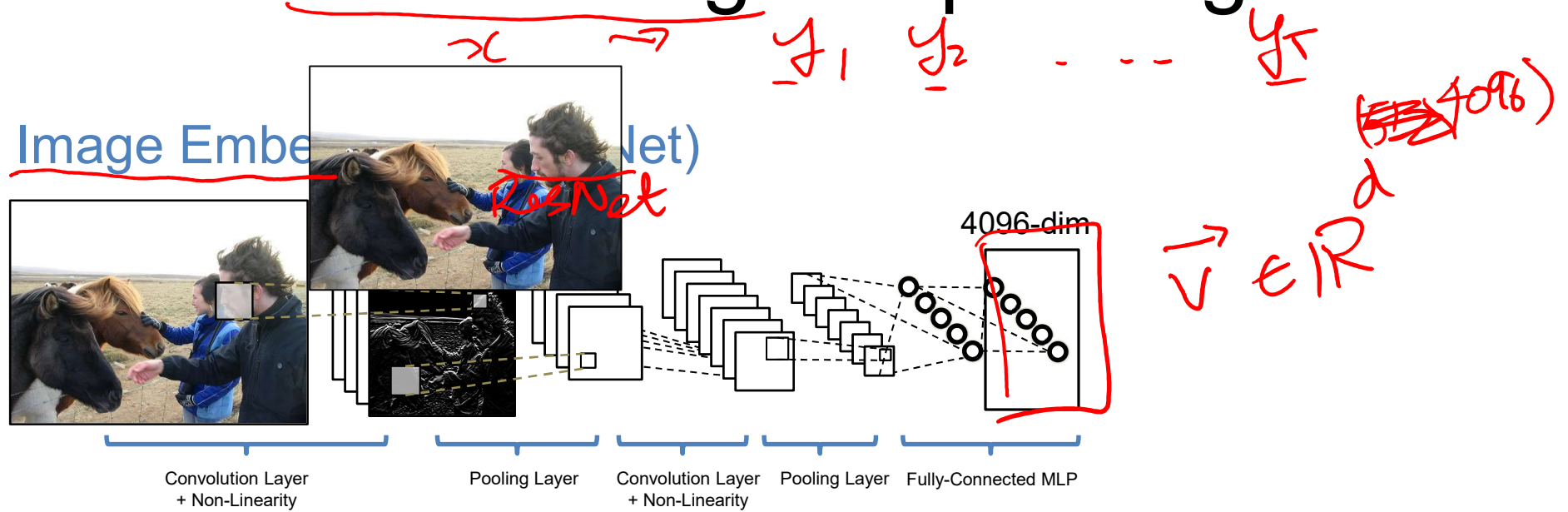
Karpathy, Johnson, and Fei-Fei: Visualizing and Understanding Recurrent Networks, ICLR Workshop 2016

Figures copyright Karpathy, Johnson, and Fei-Fei, 2015; reproduced with permission

Plan for Today

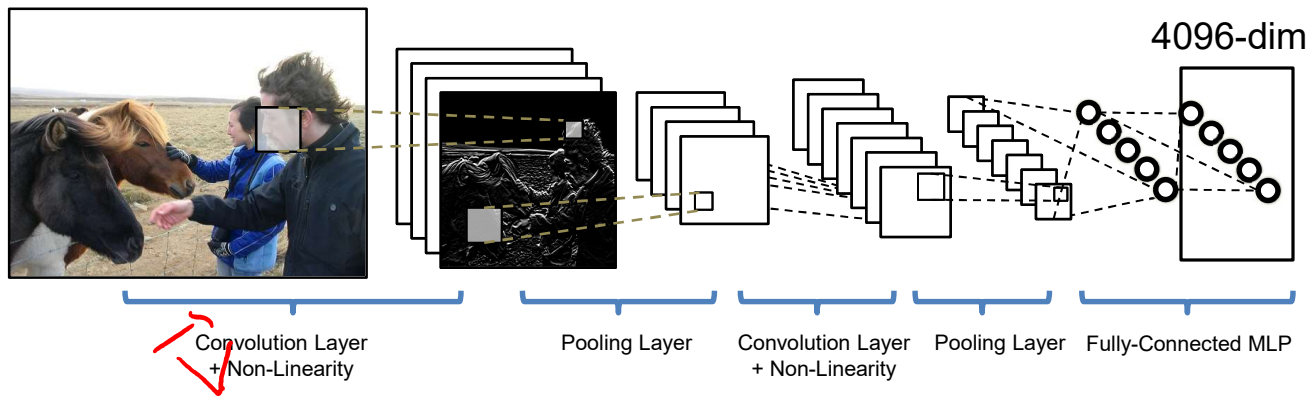
- Recurrent Neural Networks (RNNs)
 - Inference: Beam Search
 - Example: Image Captioning
 - Multilayer RNNs
 - Problems with gradients in “vanilla” RNNs
 - LSTMs (and other RNN variants)

Neural Image Captioning

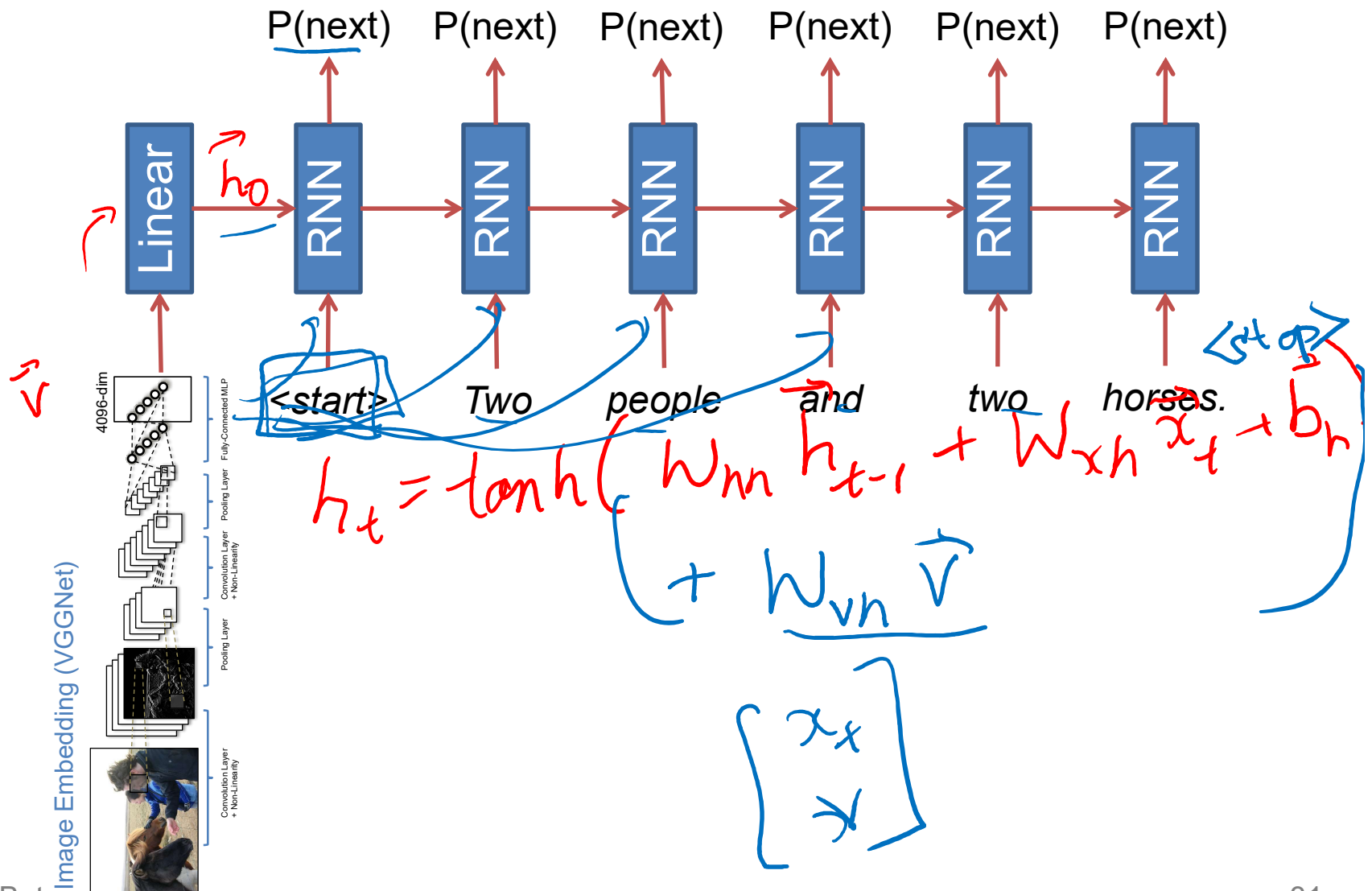


Neural Image Captioning

Image Embedding (VGGNet)



Neural Image Captioning



Beam Search Demo

- <http://dbs.cloudcv.org/captioning&mode=interactive>

Image Captioning: Example Results

Captions generated using
[neuraltalk2](#)
All images are [CC0 Public domain](#):
[cat suitcase](#) [cat tree](#) [dog bear](#)
[surfers tennis giraffe motorcycle](#)



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

Captions generated using [neuraltalk2](#)
All images are [CC0 Public domain](#): [fur](#)
[coat](#) [handstand](#) [spider web](#) [baseball](#)

Image Captioning: Failure Cases



A woman is holding a cat in her hand



A person holding a computer mouse on a desk



A woman standing on a beach holding a surfboard

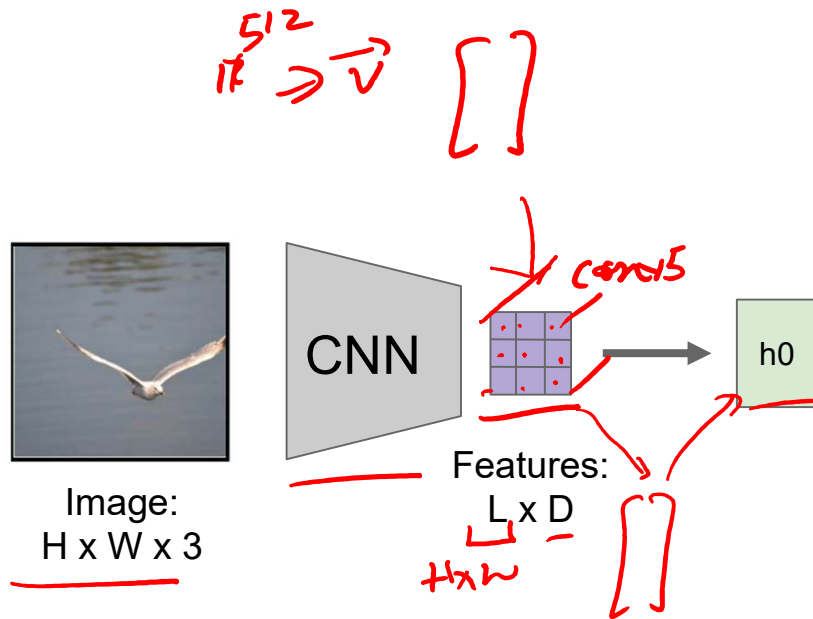


A bird is perched on a tree branch



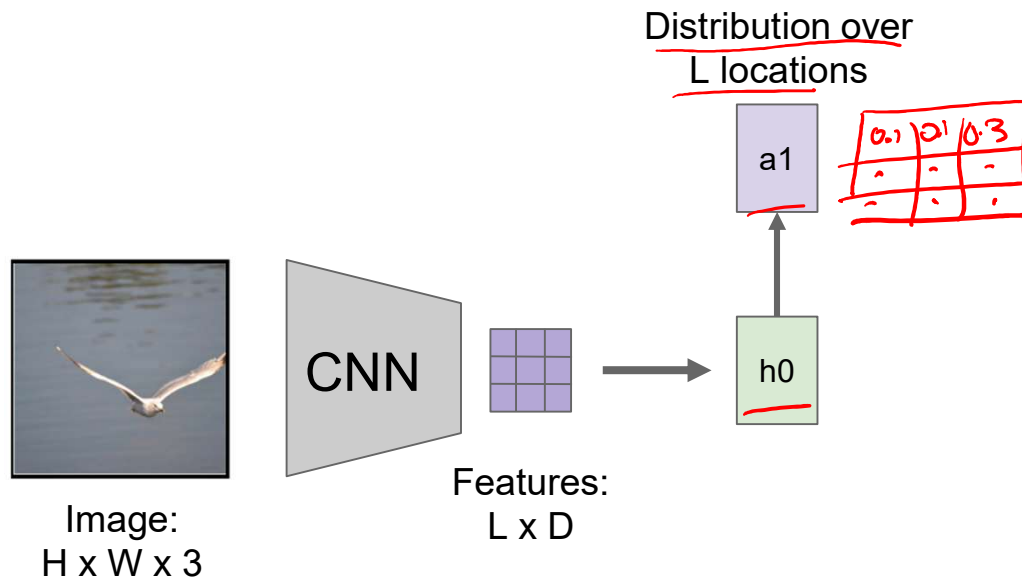
A man in a baseball uniform throwing a ball

Image Captioning with Attention



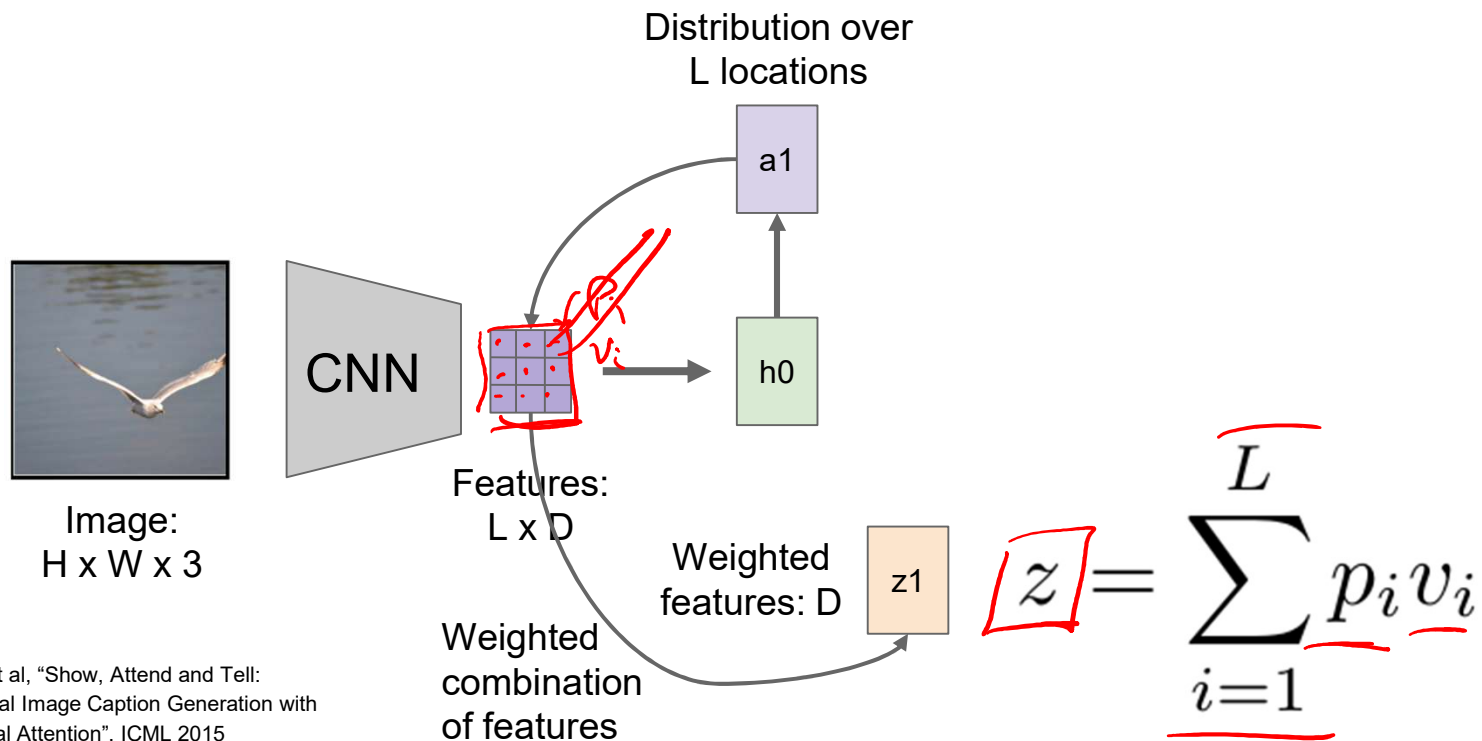
Xu et al, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention", ICML 2015

Image Captioning with Attention



Xu et al, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention", ICML 2015

Image Captioning with Attention



Xu et al, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention", ICML 2015

Image Captioning with Attention

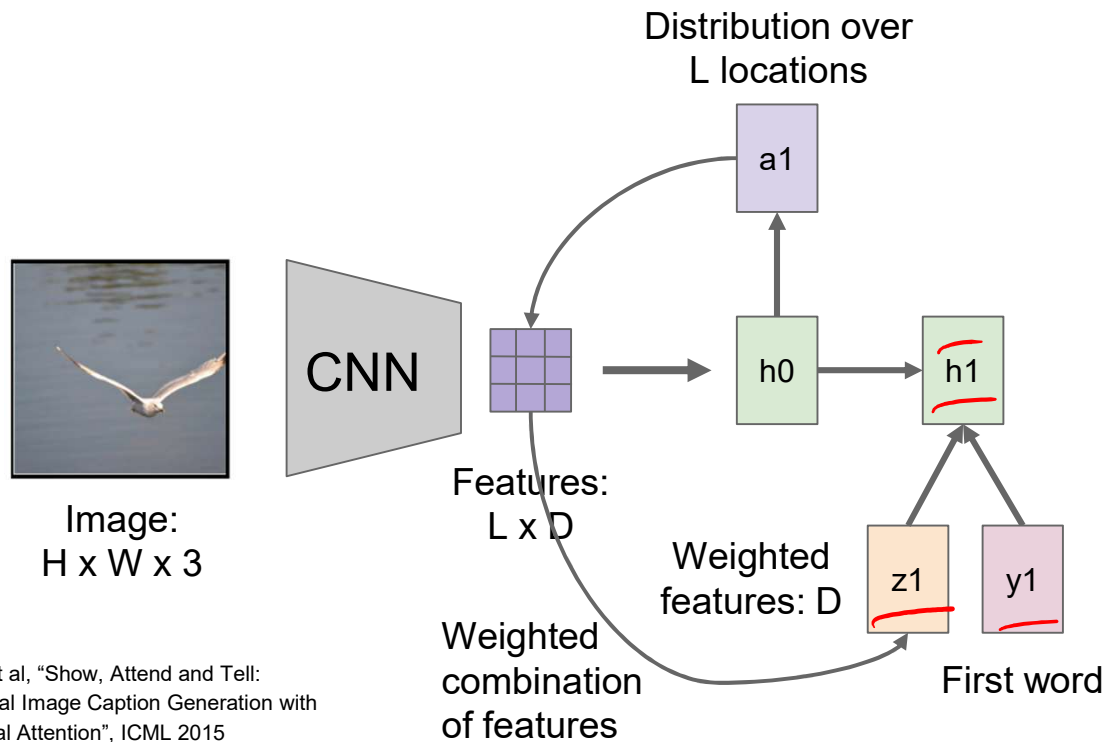
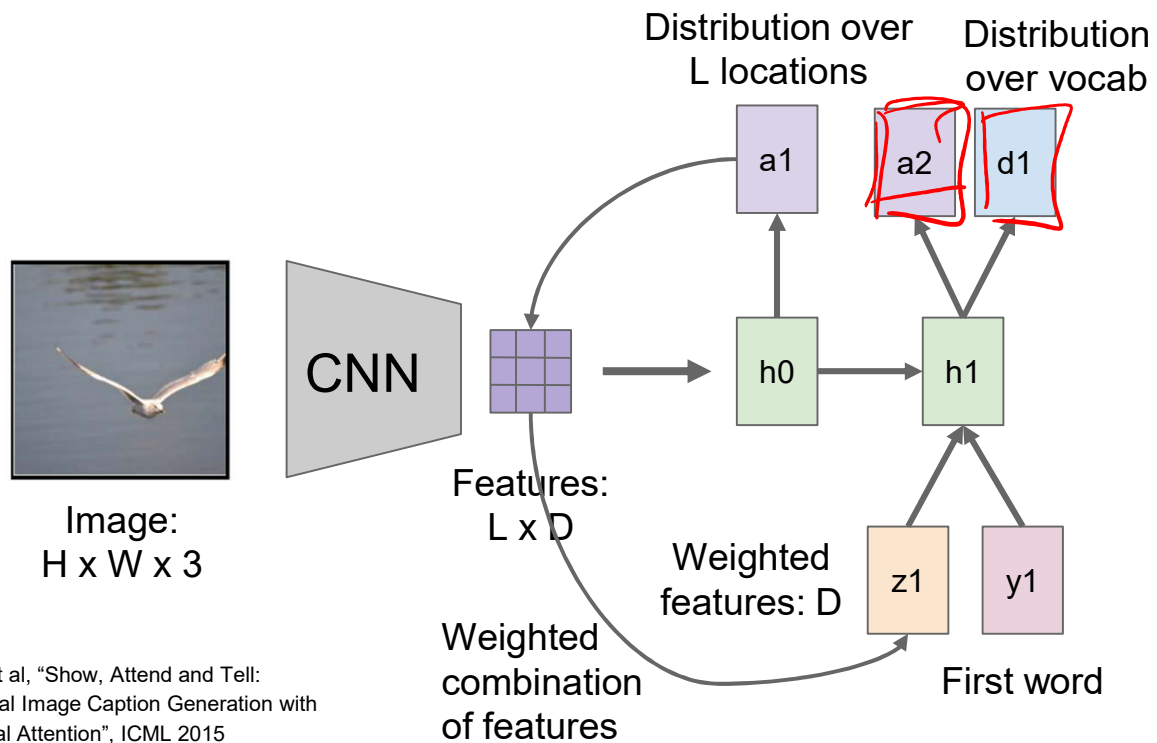
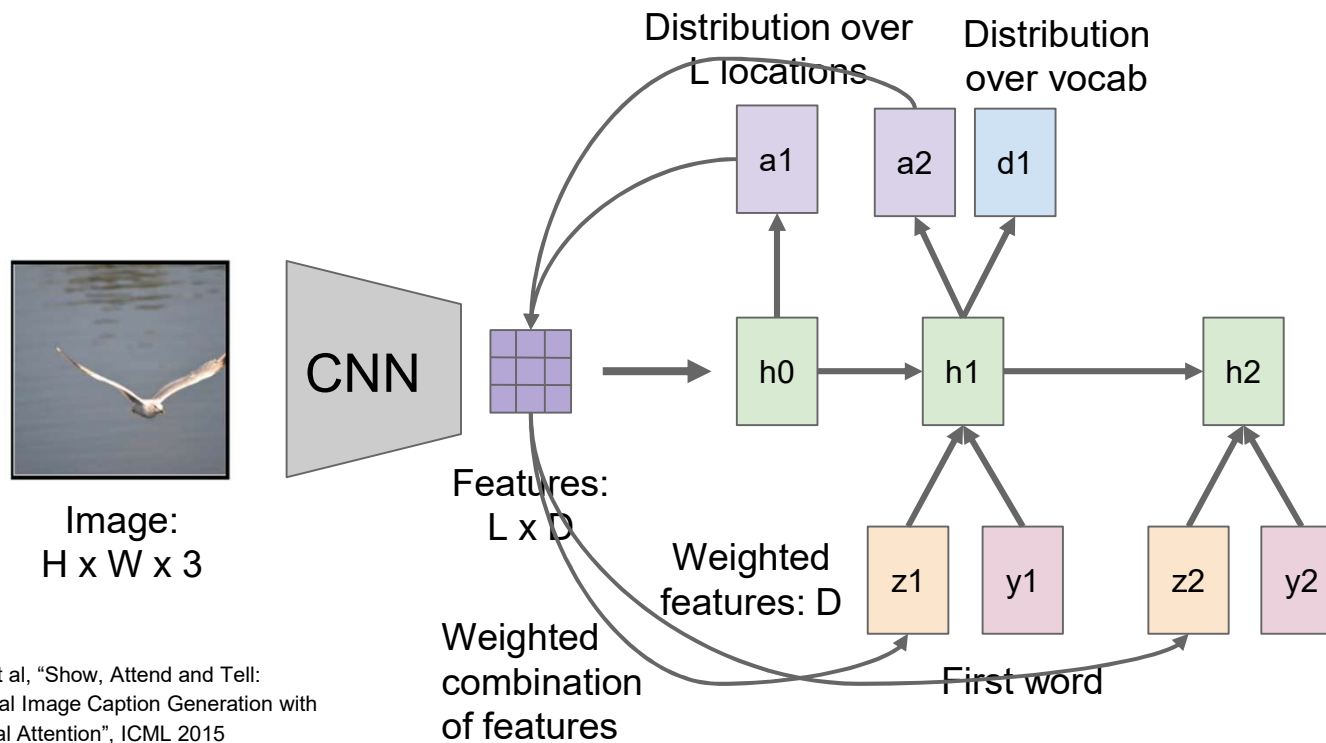


Image Captioning with Attention



Xu et al, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention", ICML 2015

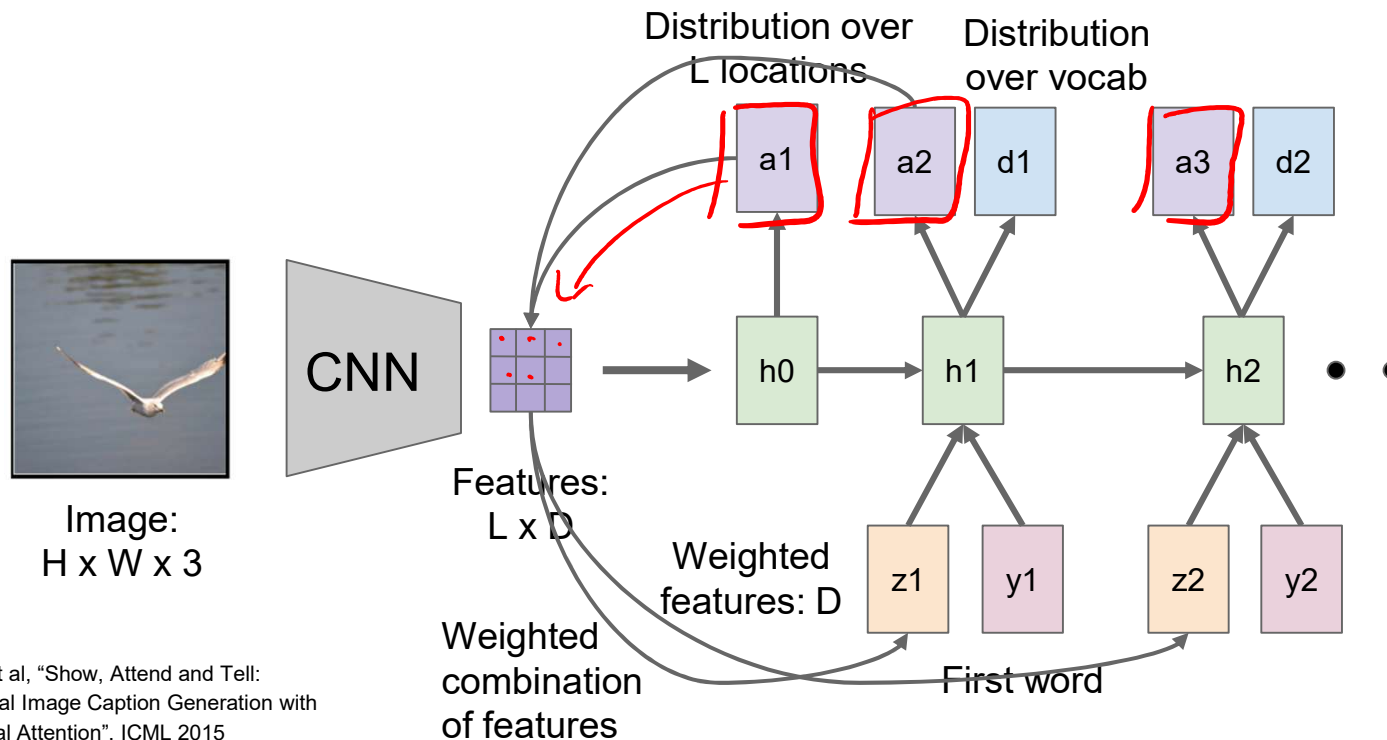
Image Captioning with Attention



Xu et al, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention", ICML 2015

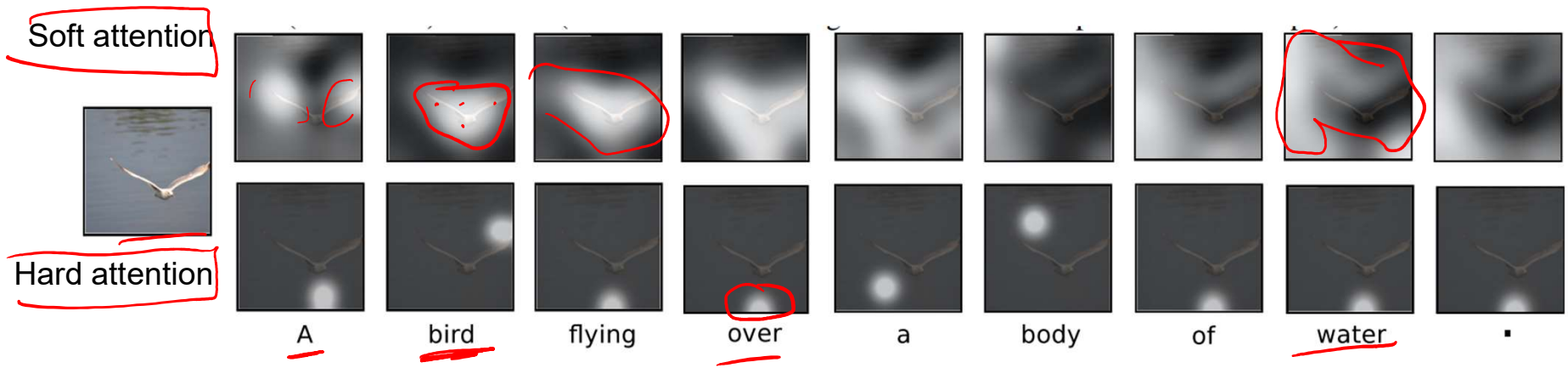
Image Captioning with Attention

*where do we
"look at"*



Xu et al, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention", ICML 2015

Image Captioning with Attention



Xu et al, "Show, Attend, and Tell: Neural Image Caption Generation with Visual Attention", ICML 2015
Figure copyright Kelvin Xu, Jimmy Lei Ba, Jamie Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Benchio, 2015. Reproduced with permission.

Image Captioning with Attention



A woman is throwing a frisbee in a park.



A dog is standing on a hardwood floor.



A stop sign is on a road with a mountain in the background.



A little girl sitting on a bed with a teddy bear.



A group of people sitting on a boat in the water.



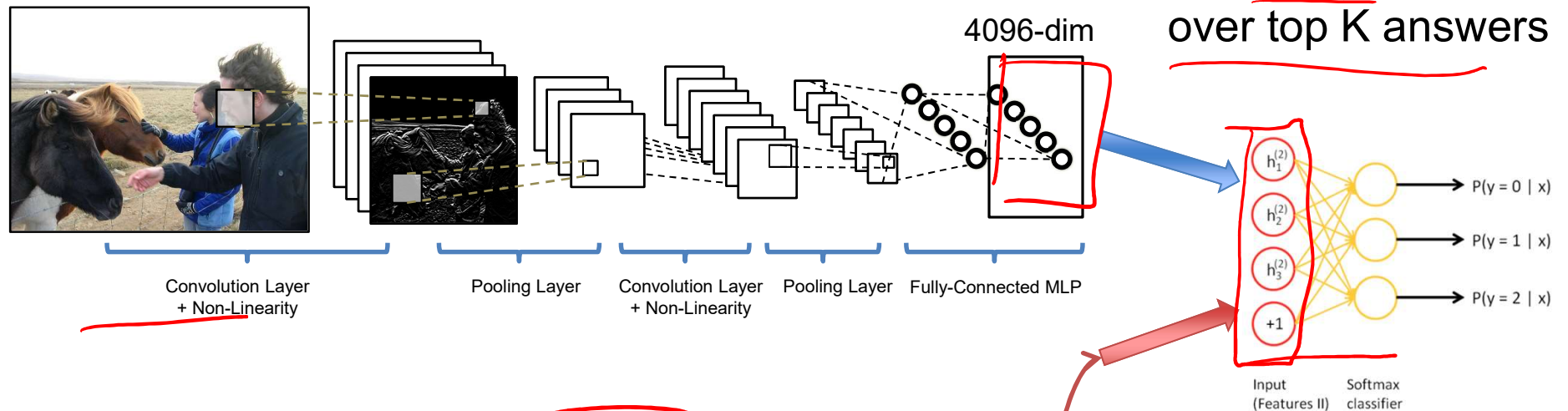
A giraffe standing in a forest with trees in the background.

Xu et al, "Show, Attend, and Tell: Neural Image Caption Generation with Visual Attention", ICML 2015

Figure copyright Kelvin Xu, Jimmy Lei Ba, Jamie Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard S. Zemel, and Yoshua Bengio, 2015. Reproduced with permission.

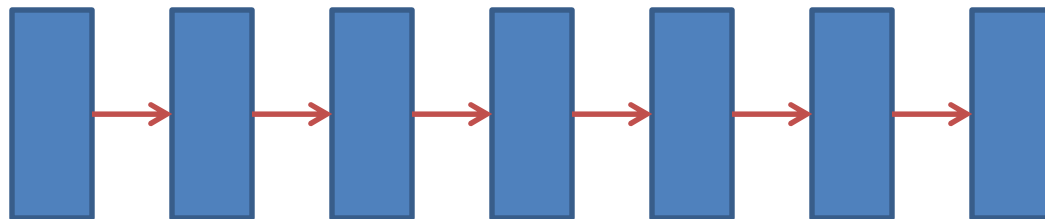
Typical VQA Models

Image Embedding (VGGNet)

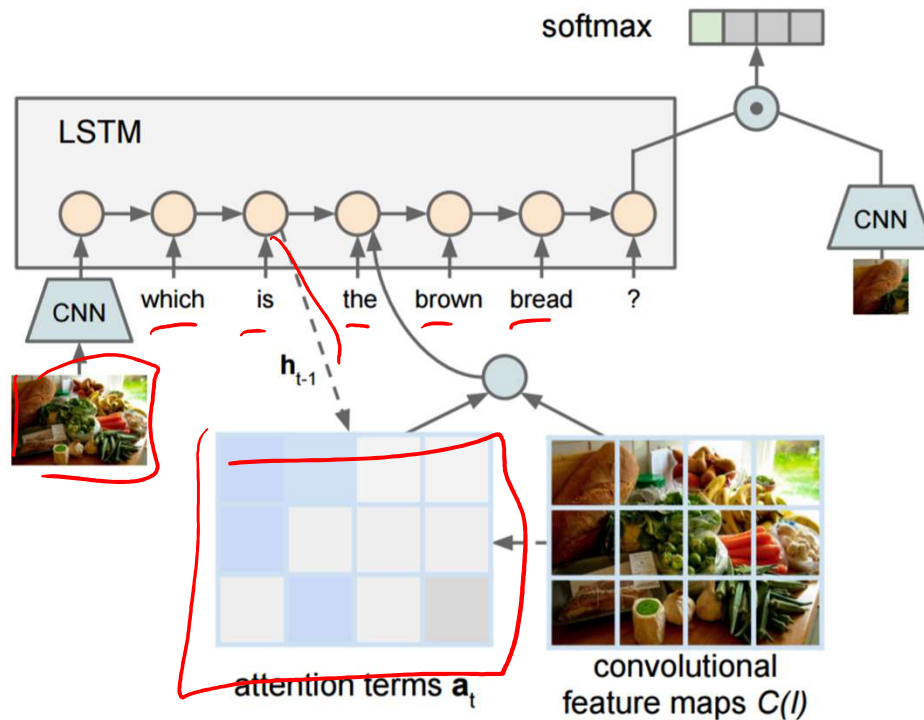


Question Embedding (LSTM)

"How many horses are in this image?"



Visual Question Answering: RNNs with Attention



What kind of animal is in the photo?

A cat.



Why is the person holding a knife?

To cut the cake with.

Zhu et al, "Visual 7W: Grounded Question Answering in Images", CVPR 2016
Figures from Zhu et al, copyright IEEE 2016. Reproduced for educational purposes.

Plan for Today

- Recurrent Neural Networks (RNNs)
 - Inference: Beam Search
 - Example: Image Captioning
 - Multilayer RNNs
 - Problems with gradients in “vanilla” RNNs
 - LSTMs (and other RNN variants)

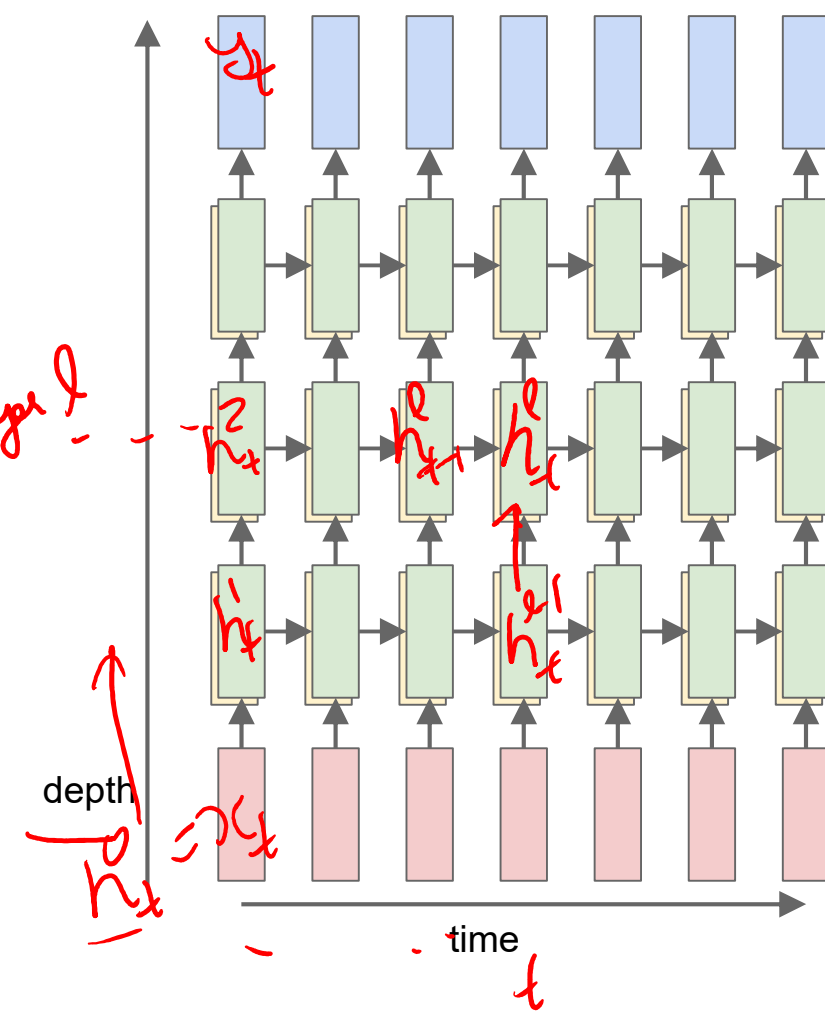
Multilayer RNNs

$$h_t^l = \tanh W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix} + b$$

$h \in \mathbb{R}^n$. $W^l [n \times 2n]$

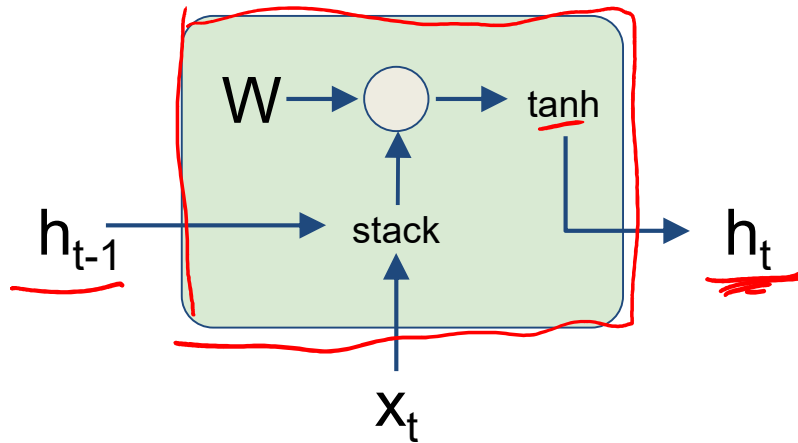
h_t^l

layer 1



Vanilla RNN Gradient Flow

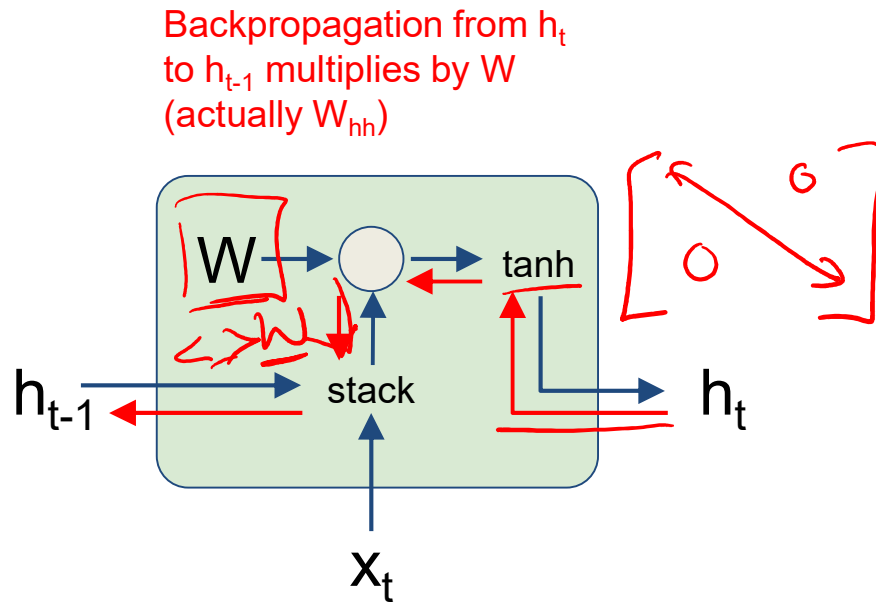
Bengio et al, "Learning long-term dependencies with gradient descent is difficult", IEEE Transactions on Neural Networks, 1994
Pascanu et al, "On the difficulty of training recurrent neural networks", ICML 2013



$$\begin{aligned} h_t &= \tanh(W_{hh}h_{t-1} + W_{hx}x_t) \\ &= \tanh\left(\begin{pmatrix} W_{hh} & W_{hx} \end{pmatrix} \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right) \\ &= \tanh\left(\underline{W} \begin{pmatrix} \underline{h_{t-1}} \\ \underline{x_t} \end{pmatrix}\right) \end{aligned}$$

Vanilla RNN Gradient Flow

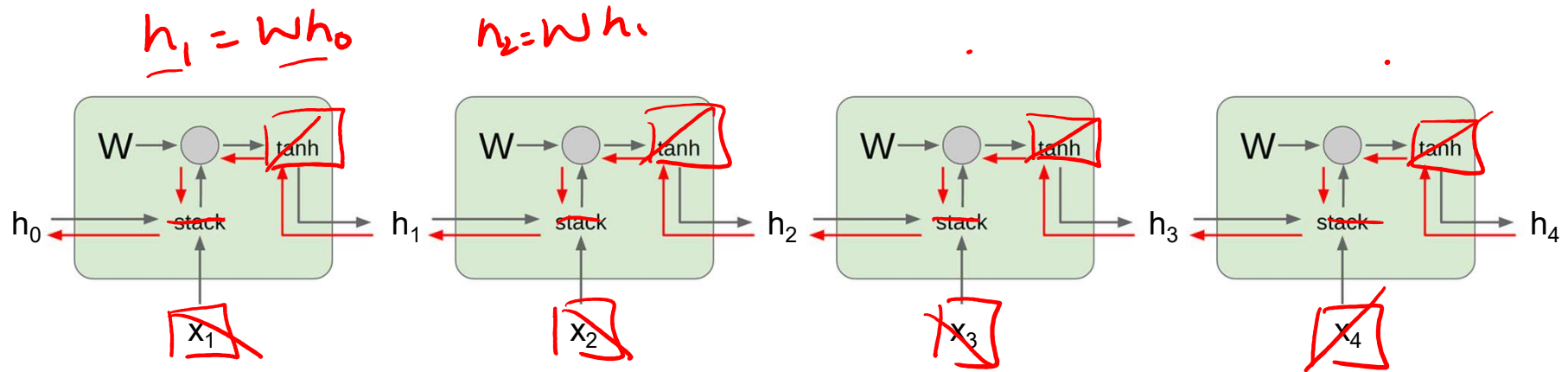
Bengio et al, "Learning long-term dependencies with gradient descent is difficult", IEEE Transactions on Neural Networks, 1994
Pascanu et al, "On the difficulty of training recurrent neural networks", ICML 2013



$$\begin{aligned} h_t &= \tanh(W_{hh}h_{t-1} + W_{hx}x_t) \\ &= \tanh\left((W_{hh} \quad W_{hx}) \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right) \\ &= \tanh\left(W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right) \end{aligned}$$

Vanilla RNN Gradient Flow

Bengio et al, "Learning long-term dependencies with gradient descent is difficult", IEEE Transactions on Neural Networks, 1994
 Pascanu et al, "On the difficulty of training recurrent neural networks", ICML 2013



Computing gradient of h_0 involves many factors of W (and repeated \tanh)

$$\frac{\partial h_T}{\partial h_0} = \left[\frac{\partial h_T}{\partial h_{T-1}} \right] \dots \frac{\partial h_1}{\partial h_0}$$

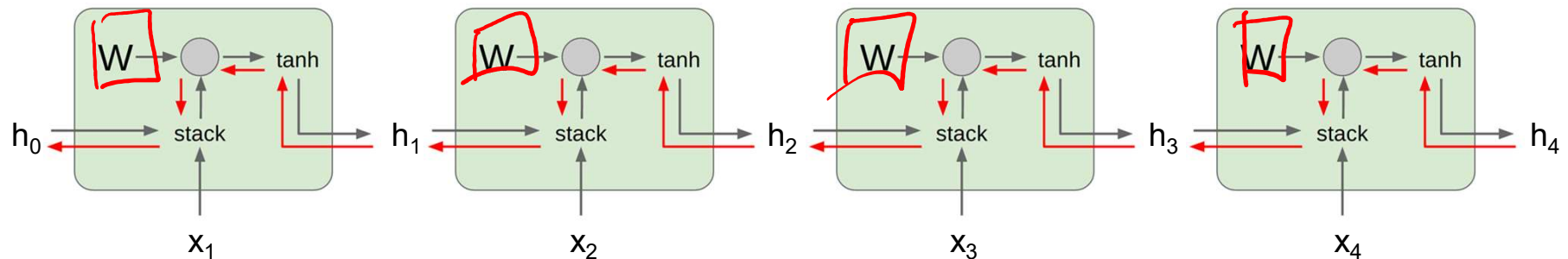
If $h_t \in \mathbb{R}^1$
 $W \in \mathbb{R}^1$
 $(W)^{T+1}$

$$= W \cdot W \cdot \dots \cdot W$$

$$= W^{(T+1)}$$

Vanilla RNN Gradient Flow

Bengio et al, "Learning long-term dependencies with gradient descent is difficult", IEEE Transactions on Neural Networks, 1994
Pascanu et al, "On the difficulty of training recurrent neural networks", ICML 2013



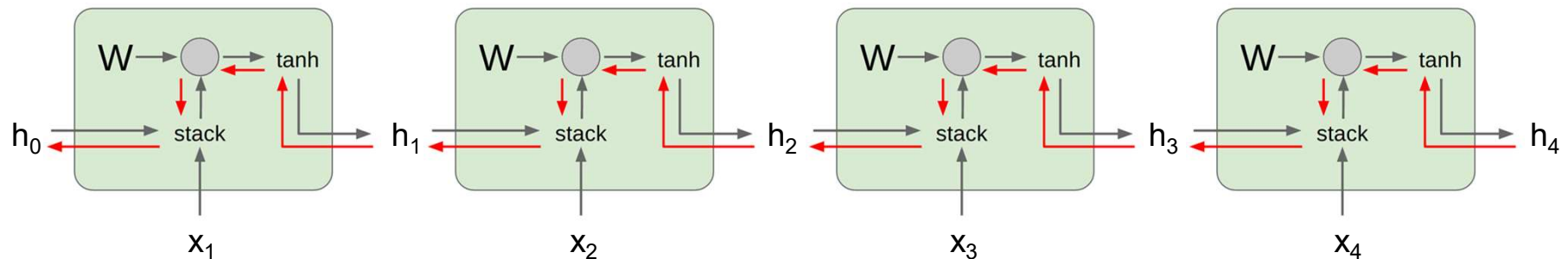
Computing gradient of h_0 involves many factors of W (and repeated tanh)

Largest singular value > 1 :
Exploding gradients

Largest singular value < 1 :
Vanishing gradients

Vanilla RNN Gradient Flow

Bengio et al, "Learning long-term dependencies with gradient descent is difficult", IEEE Transactions on Neural Networks, 1994
Pascanu et al, "On the difficulty of training recurrent neural networks", ICML 2013



Computing gradient of h_0 involves many factors of W (and repeated tanh)

Largest singular value > 1 :
Exploding gradients

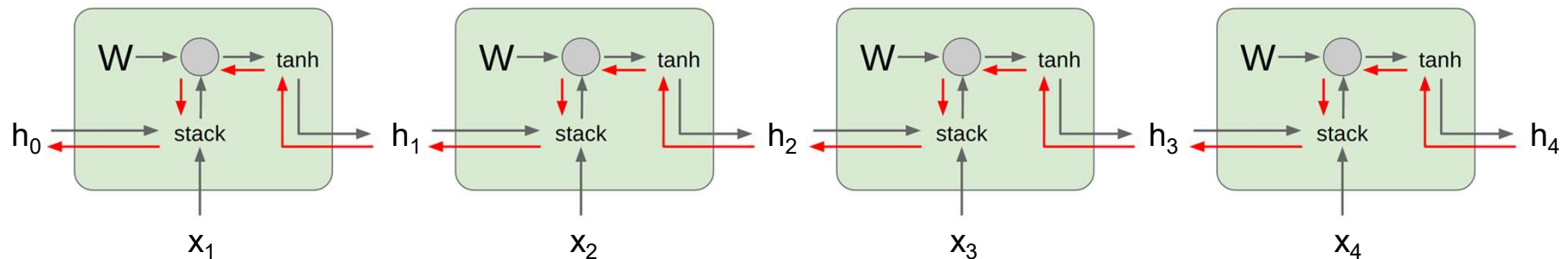
Largest singular value < 1 :
Vanishing gradients

Gradient clipping: Scale gradient if its norm is too big

```
grad_norm = np.sum(grad * grad)
if grad_norm > threshold:
    grad *= (threshold / grad_norm)
```

Vanilla RNN Gradient Flow

Bengio et al, "Learning long-term dependencies with gradient descent is difficult", IEEE Transactions on Neural Networks, 1994
Pascanu et al, "On the difficulty of training recurrent neural networks", ICML 2013



Computing gradient of h_0 involves many factors of W (and repeated tanh)

Largest singular value > 1 :
Exploding gradients

Largest singular value < 1 :
Vanishing gradients

→ Change RNN architecture

Long Short Term Memory (LSTM)

Vanilla RNN

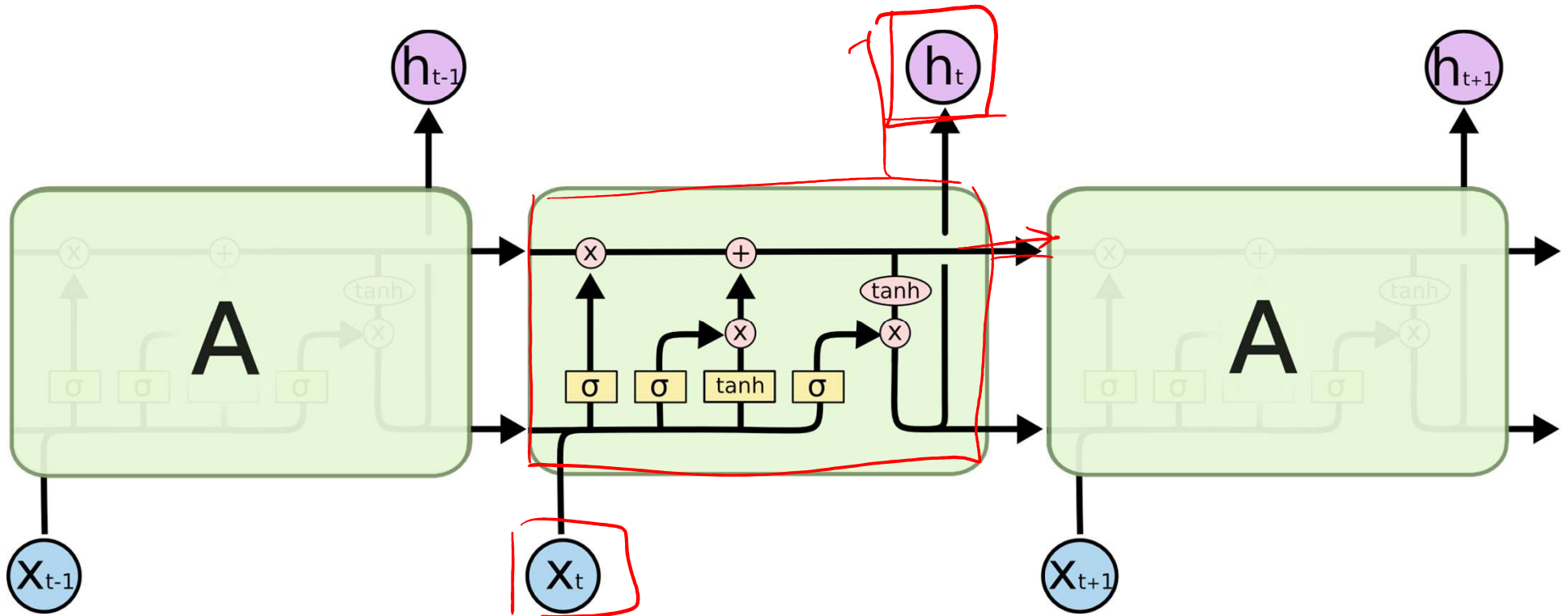
$$h_t = \tanh \left(W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix} \right)$$

LSTM

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$
$$c_t = f \odot c_{t-1} + i \odot g$$
$$h_t = o \odot \tanh(c_t)$$

Hochreiter and Schmidhuber, "Long Short Term Memory", Neural Computation 1997

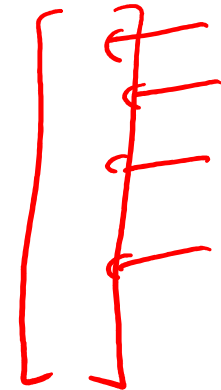
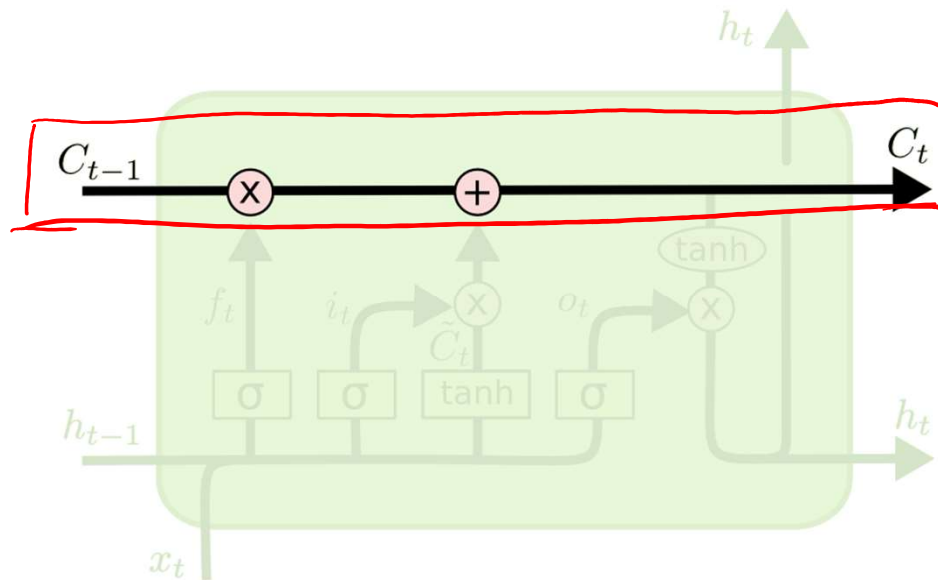
Meet LSTMs



LSTMs Intuition: Memory

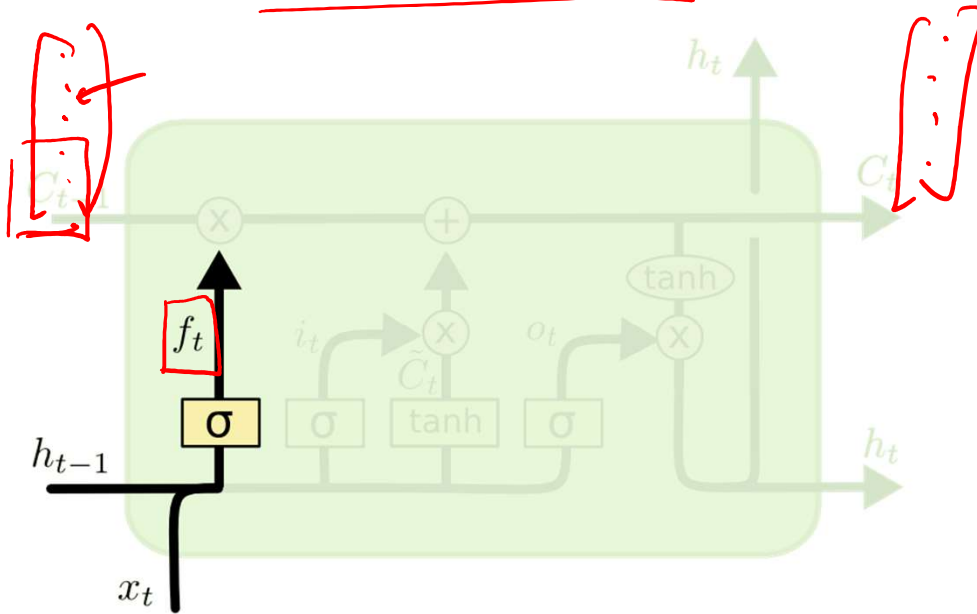
- Cell State / Memory

C_t : h_t hidden state
 $C_t \in \mathbb{R}^d$
 $h_t \in \mathbb{R}^d$



LSTMs Intuition: Forget Gate

- Should we continue to remember this “bit” of information or not?



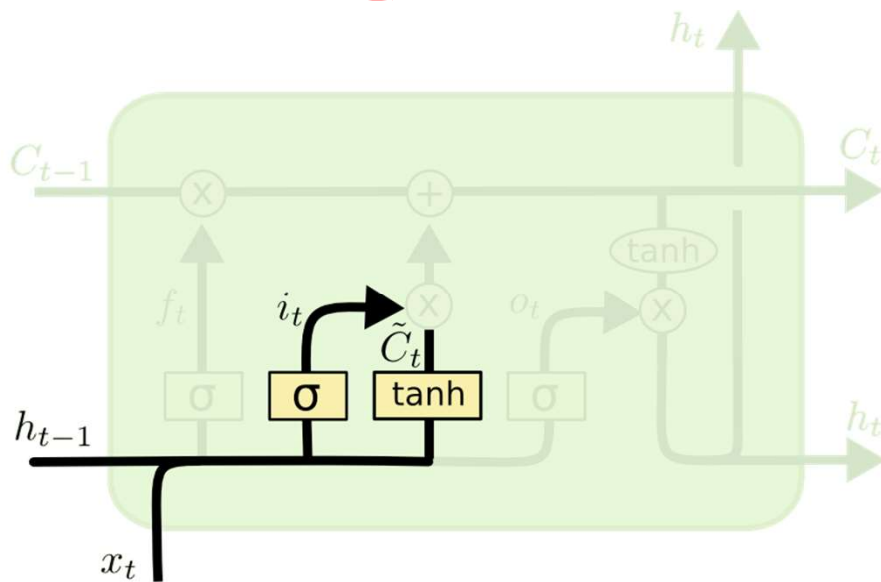
$$f_t \in \mathbb{R}^d \quad \begin{bmatrix} \vdots \\ \circ \\ \vdots \end{bmatrix}$$

$$\underline{f}_t = \underline{\sigma}(\underline{W}_f \cdot \underline{[h_{t-1}, x_t]} + \underline{b}_f)$$

LSTMs Intuition: Input Gate

- Should we update this “bit” of information or not?
 - If so, with what?

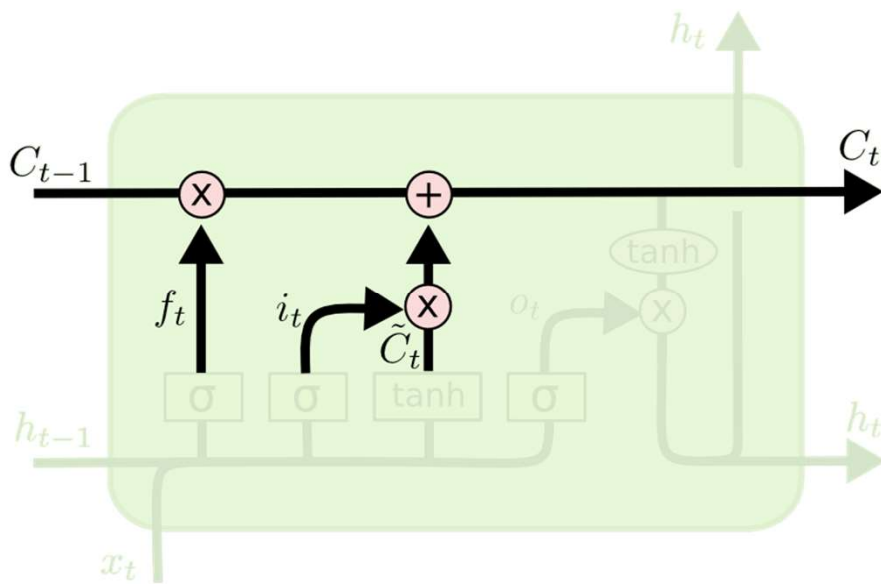
$$i_t \in \mathbb{R}^d \begin{bmatrix} \vdots \\ \vdots \\ \vdots \end{bmatrix}$$



$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

LSTMs Intuition: Memory Update

- Forget that + memorize this

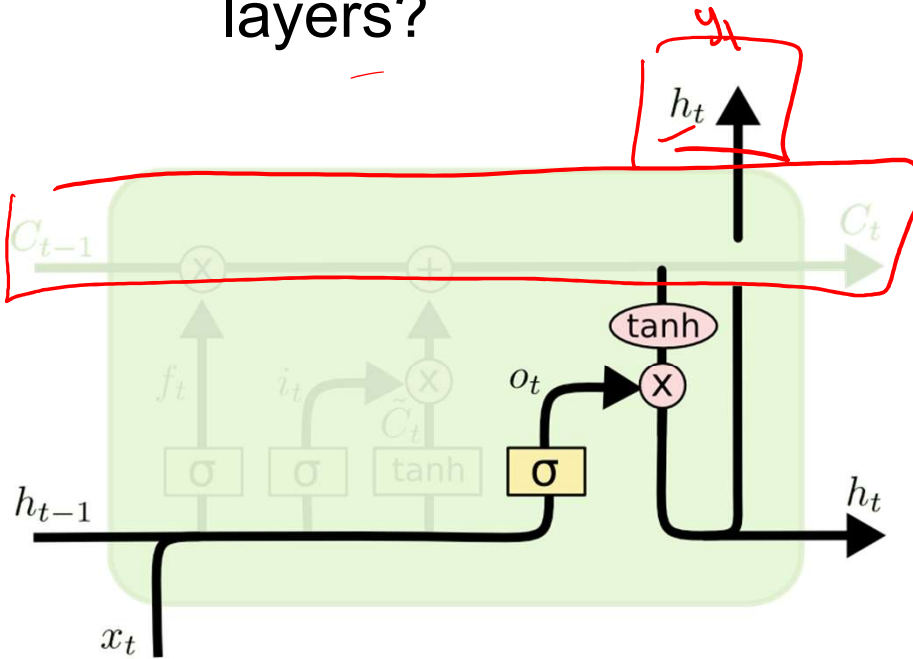


$$C_t = \boxed{f_t} * \underline{C_{t-1}} + \boxed{i_t} * \tilde{C}_t$$

$\underbrace{\begin{bmatrix} 0 & \dots & 0 \\ \vdots & & \vdots \end{bmatrix}}_{\text{forget}} + \begin{bmatrix} \vdots & \tilde{C}_t \end{bmatrix}$

LSTMs Intuition: Output Gate

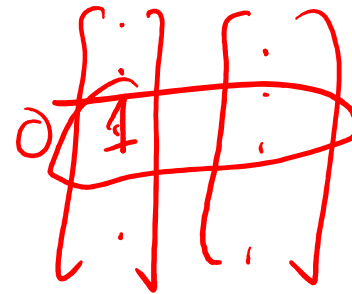
- Should we output this “bit” of information to “deeper” layers?



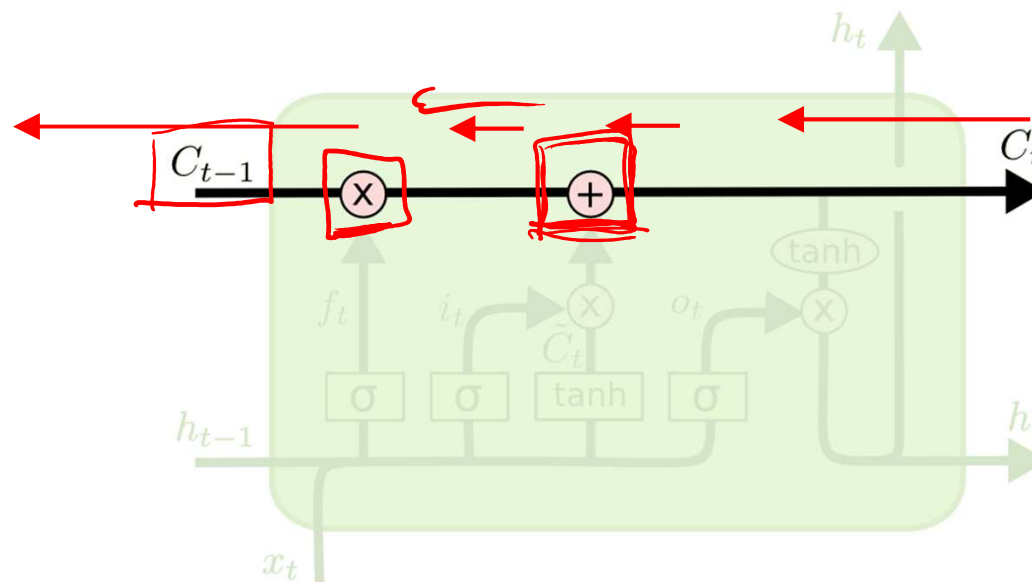
$$o_t \in \mathbb{R}^d$$

$$o_t = \sigma(W_o [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

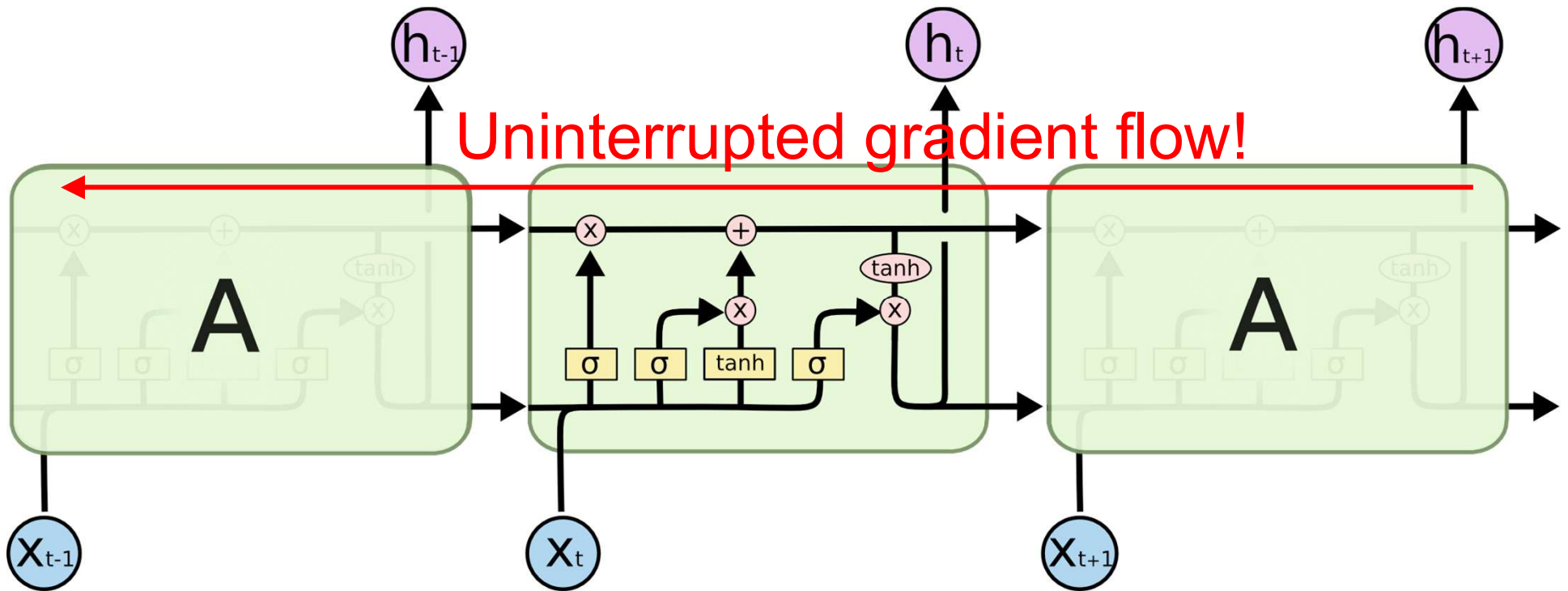


LSTMs Intuition: Additive Updates

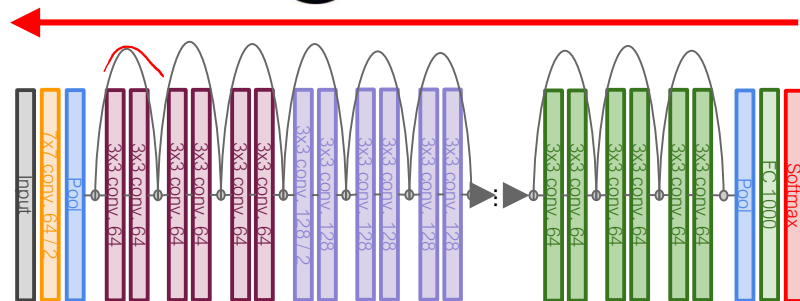
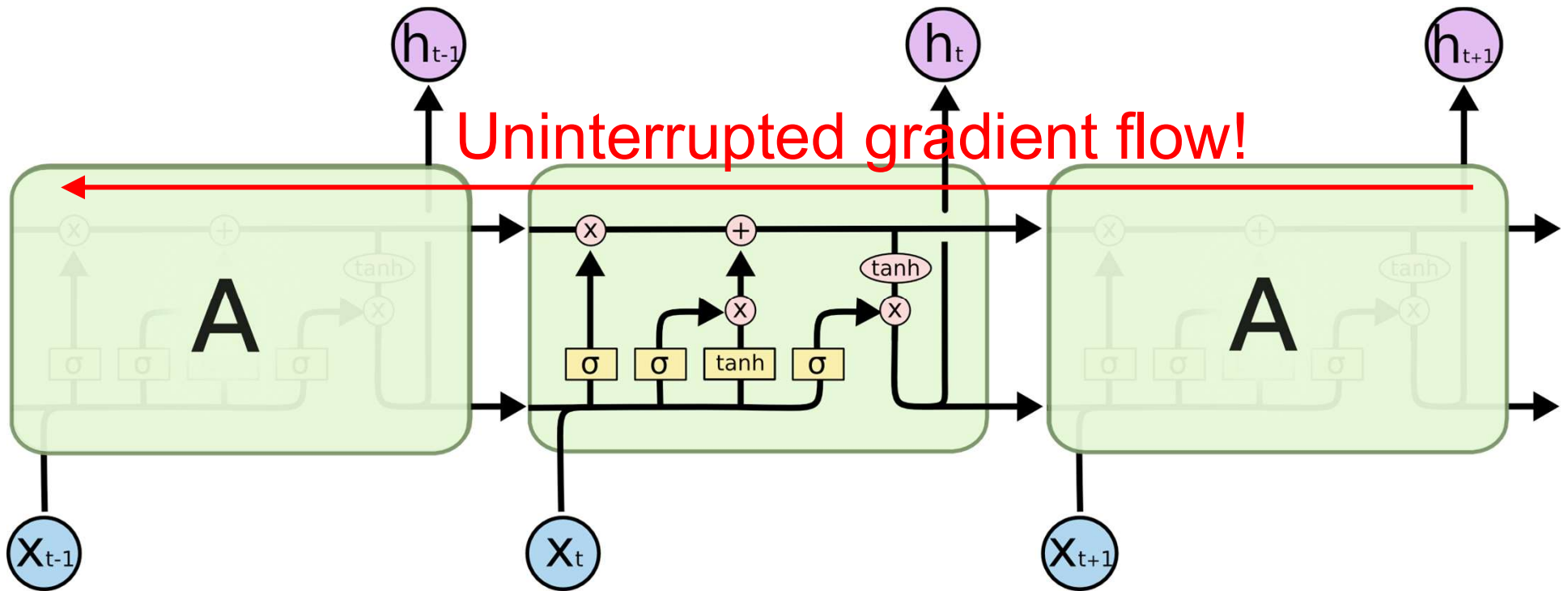


Backpropagation from c_t to c_{t-1} only
elementwise
multiplication by f , no
matrix multiply by W

LSTMs Intuition: Additive Updates



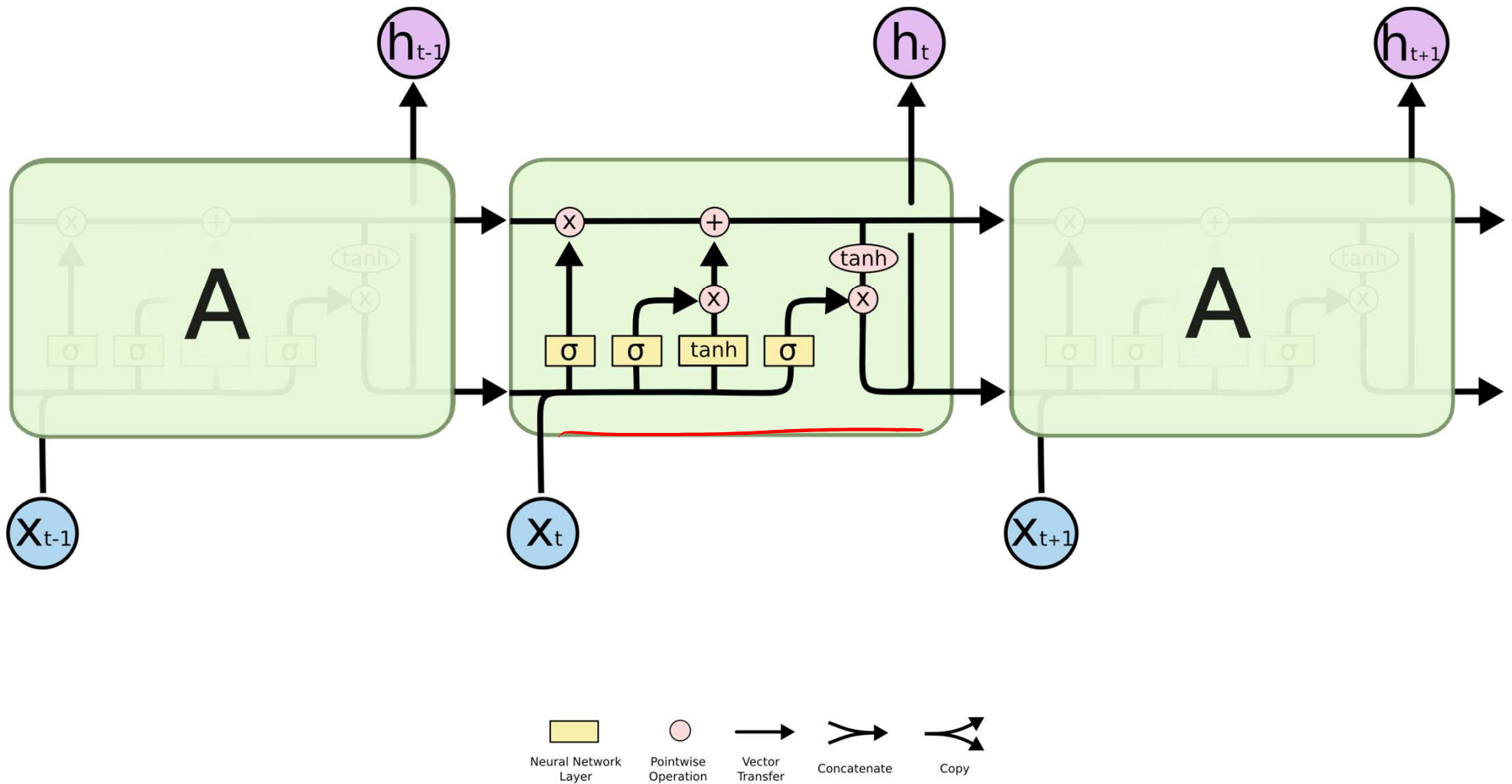
LSTMs Intuition: Additive Updates



Similar to ResNet!

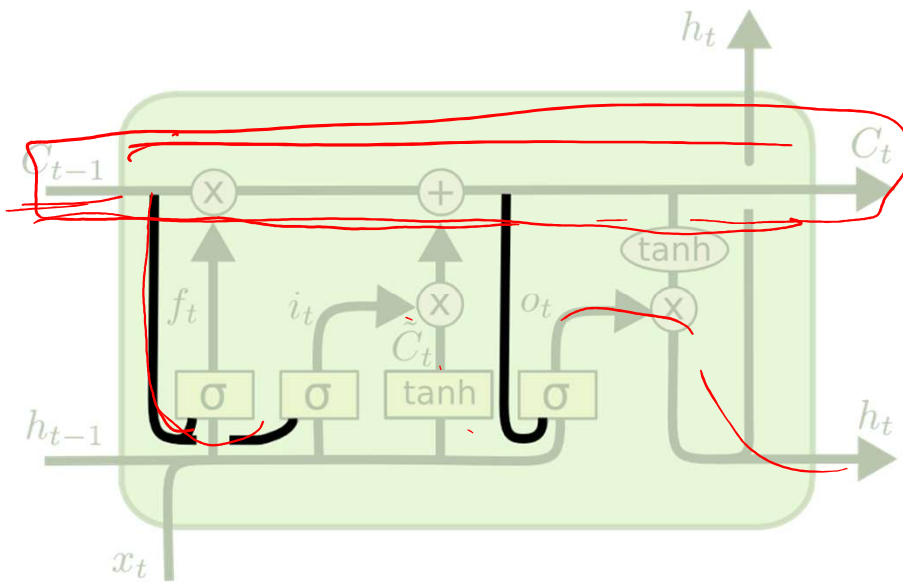
LSTMs

- A pretty sophisticated cell



LSTM Variants #1: Peephole Connections

- Let gates see the cell state / memory



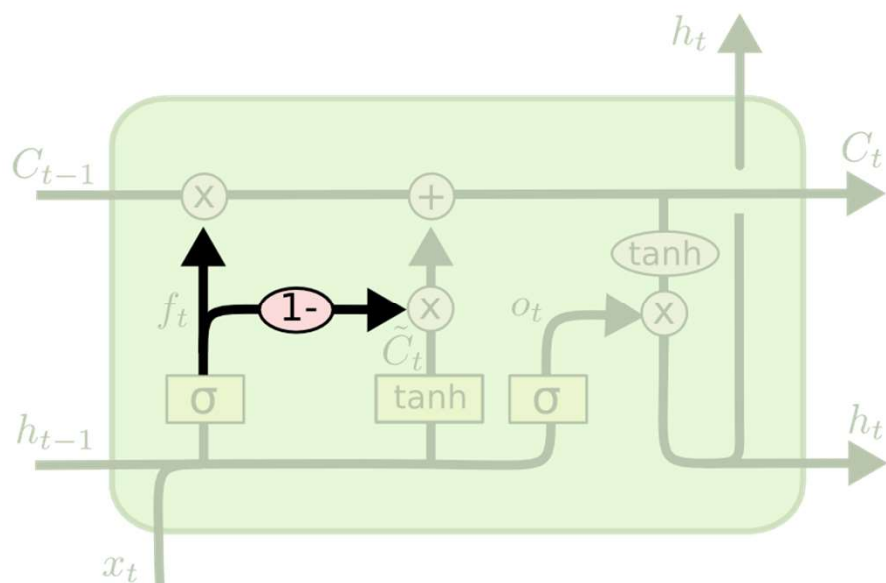
$$f_t = \sigma(W_f \cdot [C_{t-1}, h_{t-1}, x_t] + b_f)$$

$$i_t = \sigma(W_i \cdot [C_{t-1}, h_{t-1}, x_t] + b_i)$$

$$o_t = \sigma(W_o \cdot [C_t, h_{t-1}, x_t] + b_o)$$

LSTM Variants #2: Coupled Gates

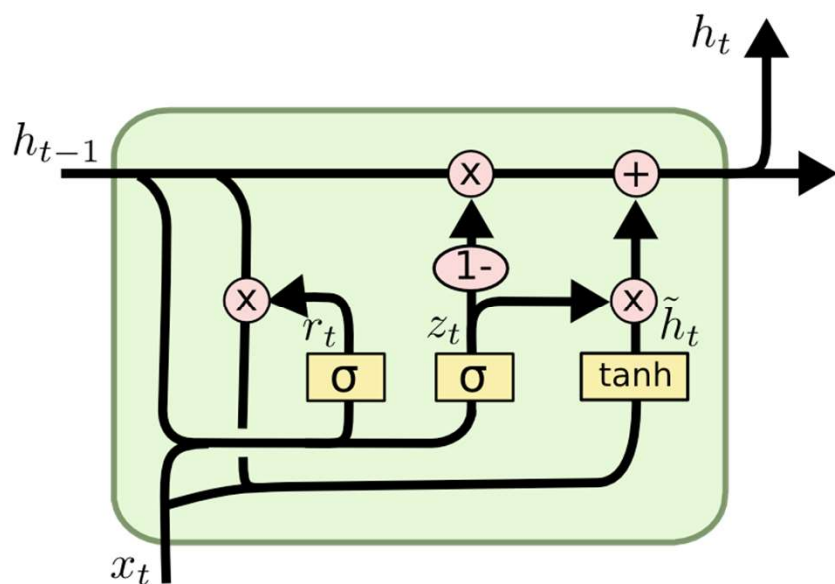
- Only memorize new if forgetting old



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

LSTM Variants #3: Gated Recurrent Units

- Changes:
 - No explicit memory; memory = hidden output
 - Z = memorize new and forget old



$$\underline{z_t} = \sigma (W_z \cdot [h_{t-1}, x_t])$$

$$\underline{r_t} = \sigma (W_r \cdot [h_{t-1}, x_t])$$

$$\underline{\tilde{h}_t} = \tanh (W \cdot [r_t * h_{t-1}, x_t])$$

$$\underline{h_t} = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Other RNN Variants

[An Empirical Exploration of
Recurrent Network Architectures,
Jozefowicz et al., 2015]

MUT1:

$$\begin{aligned} z &= \text{sigm}(W_{xz}x_t + b_z) \\ r &= \text{sigm}(W_{xr}x_t + W_{hr}h_t + b_r) \\ h_{t+1} &= \tanh(W_{hh}(r \odot h_t) + \tanh(x_t) + b_h) \odot z \\ &+ h_t \odot (1 - z) \end{aligned}$$

MUT2:

$$\begin{aligned} z &= \text{sigm}(W_{xz}x_t + W_{hz}h_t + b_z) \\ r &= \text{sigm}(x_t + W_{hr}h_t + b_r) \\ h_{t+1} &= \tanh(W_{hh}(r \odot h_t) + W_{xh}x_t + b_h) \odot z \\ &+ h_t \odot (1 - z) \end{aligned}$$

MUT3:

$$\begin{aligned} z &= \text{sigm}(W_{xz}x_t + W_{hz} \tanh(h_t) + b_z) \\ r &= \text{sigm}(W_{xr}x_t + W_{hr}h_t + b_r) \\ h_{t+1} &= \tanh(W_{hh}(r \odot h_t) + W_{xh}x_t + b_h) \odot z \\ &+ h_t \odot (1 - z) \end{aligned}$$

Summary

- RNNs allow a lot of flexibility in architecture design
- Vanilla RNNs are simple but don't work very well
- Common to use LSTM or GRU: their additive interactions improve gradient flow
- Backward flow of gradients in RNN can explode or vanish. Exploding is controlled with gradient clipping. Vanishing is controlled with additive interactions (LSTM)
- Better/simpler architectures are a hot topic of current research
- Better understanding (both theoretical and empirical) is needed.