Learning Neural LMs

CS 685, Fall 2025

Advanced Natural Language Processing

https://people.cs.umass.edu/~brenocon/cs685_f25/

Brendan O'Connor

College of Information and Computer Sciences

University of Massachusetts Amherst

Gradient-based learning

- Goal: learn all model parameters W
- Loss function L(W) based on dataset
- Choose W to minimize L(W) be following the negative gradient of the loss
- Intuition: cross-entropy gradient shifts probability mass to the data

- blackboard

Backpropagation

- Implemented for you in PyTorch
- Just define the forward computation graph, then it performs the backward pass for you!
- a = ...
- b = ...
- loss = ...
- loss.backward()
- a.grad # => the gradient vector!

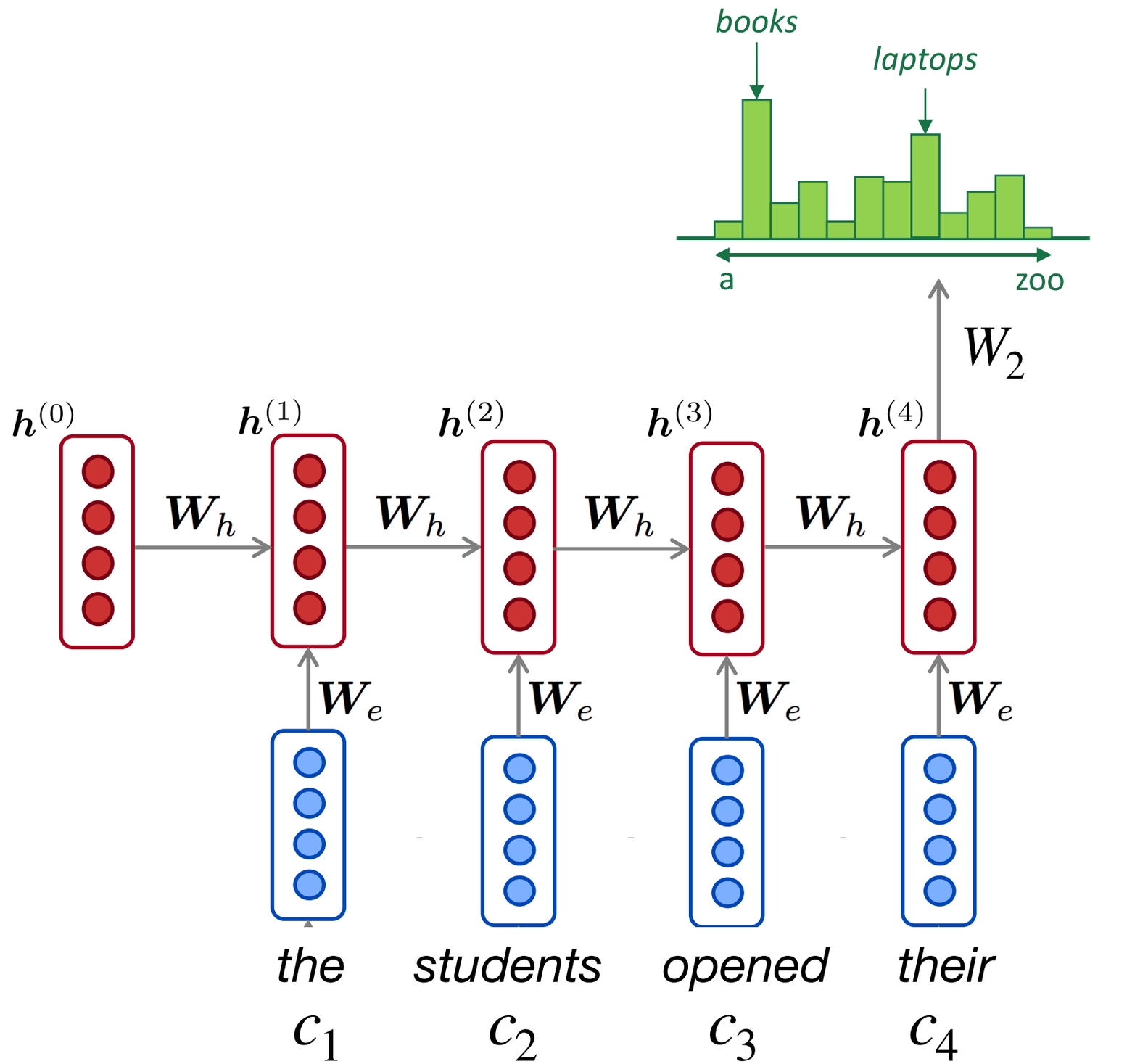
why is this good?

RNN **Advantages**:

- Can process any length input
- Model size doesn't increase for longer input
- Computation for step *t* can (in theory) use information from many steps back
- Weights are shared across timesteps → representations are shared

RNN **Disadvantages**:

- Recurrent computation is slow
- In practice, difficult to access information from many steps back

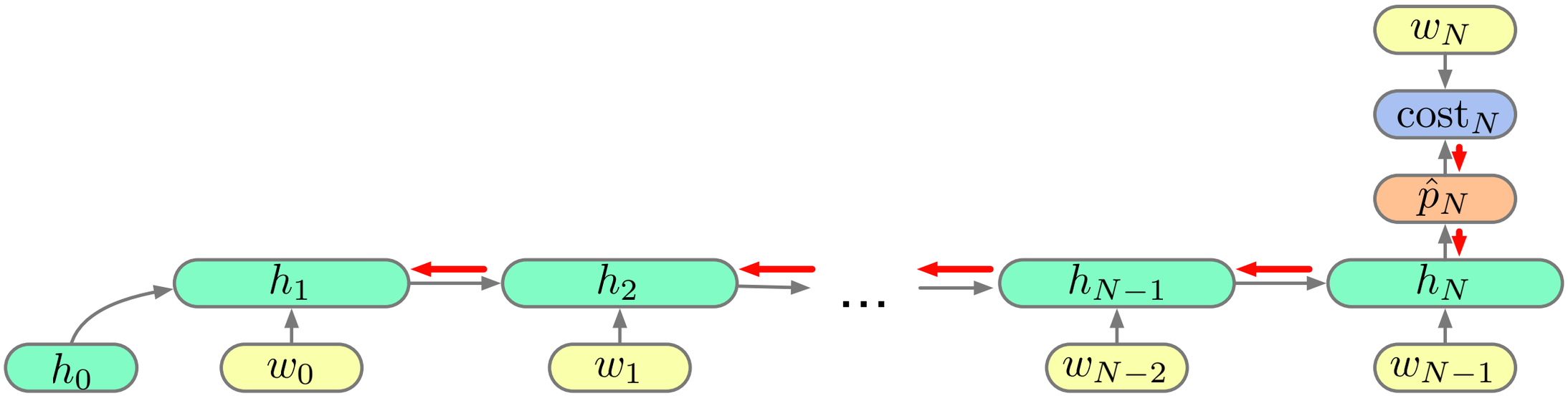


Capturing Long Range Dependencies

If an RNN Language Model is to outperform an n-gram model it must discover and represent long range dependencies:

p(sandcastle | Alice went to the beach. There she built a)

While a simple RNN LM can represent such dependencies in theory, can it learn them?

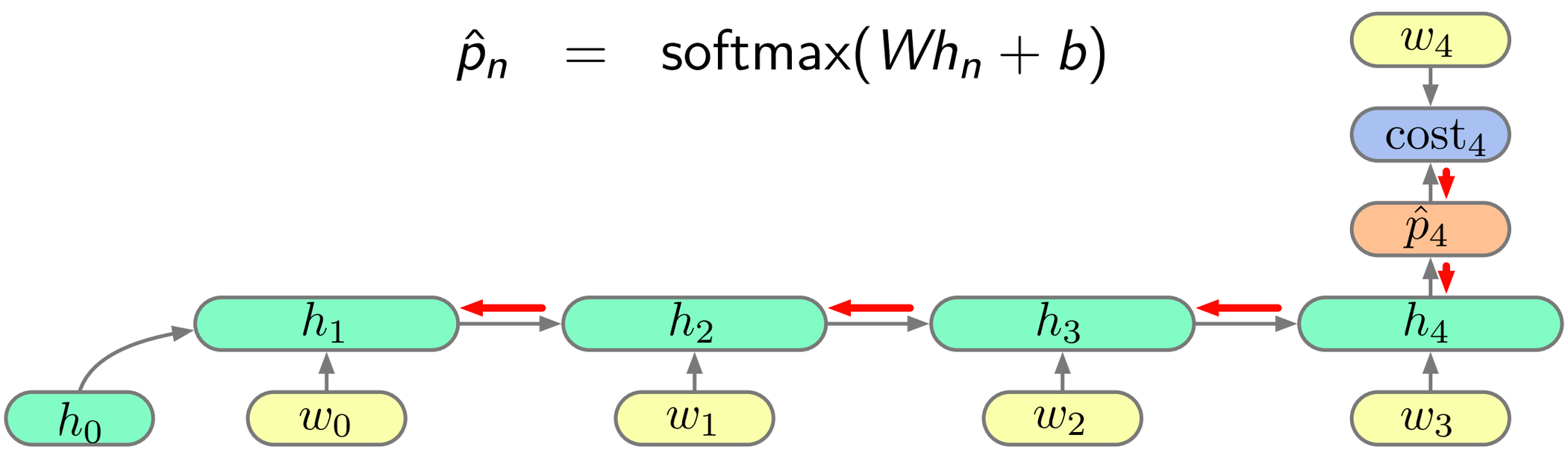


[Slide: Phil Blunsom]

RNNs: Exploding and Vanishing Gradients

Consider the path of partial derivatives linking a change in cost_4 to changes in h_1:

$$\begin{aligned} h_n &= g(V[x_n; h_{n-1}] + c) \\ \hat{p}_n &= \text{softmax}(Wh_n + b) \end{aligned}$$



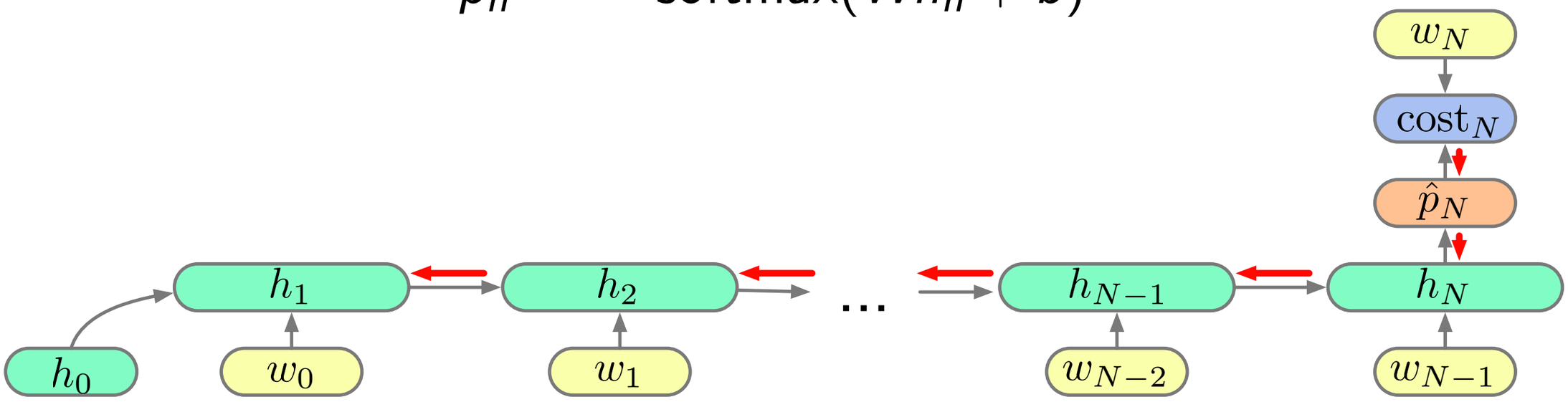
$$\frac{\partial \text{cost}_4}{\partial h_1} = \frac{\partial \text{cost}_4}{\partial \hat{p}_4} \frac{\partial \hat{p}_4}{\partial h_4} \frac{\partial h_4}{\partial h_3} \frac{\partial h_3}{\partial h_2} \frac{\partial h_2}{\partial h_1}$$

[Slide: Phil Blunsom]

RNNs: Exploding and Vanishing Gradients

Consider the path of partial derivatives linking a change in cost_N to changes in h_1:

$$\begin{aligned} h_n &= g(V[x_n; h_{n-1}] + c) \\ \hat{p}_n &= \text{softmax}(Wh_n + b) \end{aligned}$$



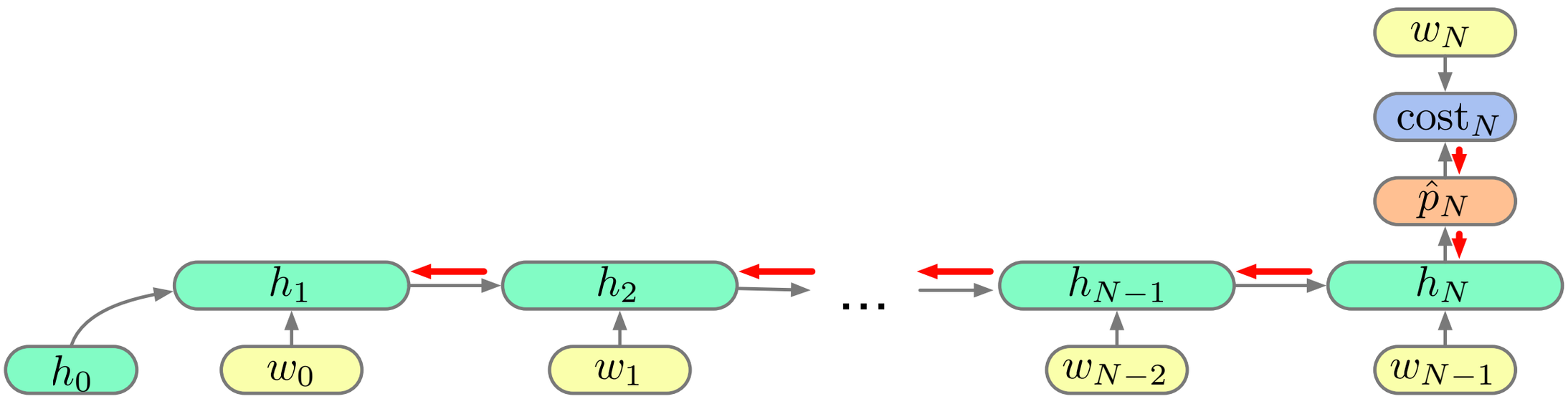
$$\frac{\partial \text{cost}_N}{\partial h_1} = \frac{\partial \text{cost}_N}{\partial \hat{p}_N} \frac{\partial \hat{p}_N}{\partial h_N} \left(\prod_{n \in \{N,\ldots,2\}} \frac{\partial h_n}{\partial h_{n-1}} \right)$$

[Slide: Phil Blunsom]

RNNs: Exploding and Vanishing Gradients

Consider the path of partial derivatives linking a change in cost_N to changes in h_1:

$$h_n = g(V[x_n; h_{n-1}] + c), \qquad \frac{\partial \text{cost}_N}{\partial h_1} = \frac{\partial \text{cost}_N}{\partial \hat{p}_N} \frac{\partial \hat{p}_N}{\partial h_N} \left(\prod_{n \in \{N,\ldots,2\}} \frac{\partial h_n}{\partial h_{n-1}} \right)$$

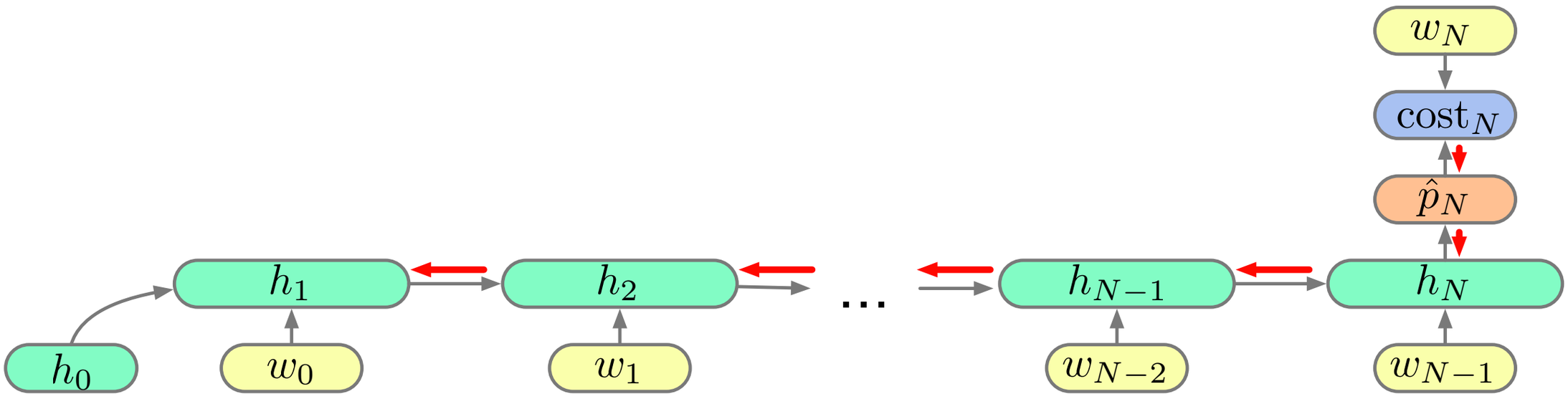


[Slide: Phil Blunsom]

RNNs: Exploding and Vanishing Gradients

Consider the path of partial derivatives linking a change in cost_N to changes in h_1:

$$h_n = g(\underbrace{V_x x_n + V_h h_{n-1} + c}_{z_n}), \quad \frac{\partial \text{cost}_N}{\partial h_1} = \frac{\partial \text{cost}_N}{\partial \hat{p}_N} \frac{\partial \hat{p}_N}{\partial h_N} \left(\prod_{n \in \{N,\ldots,2\}} \frac{\partial h_n}{\partial z_n} \frac{\partial z_n}{\partial h_{n-1}} \right)$$



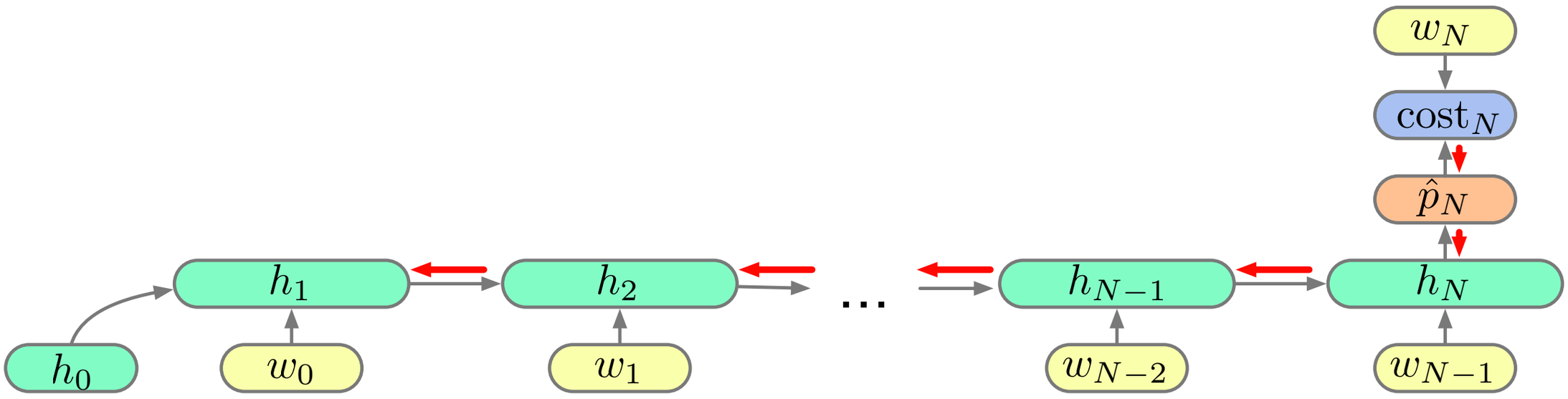
[Slide: Phil Blunsom]

RNNs: Exploding and Vanishing Gradients

Consider the path of partial derivatives linking a change in cost_N to changes in h_1:

$$h_n = g(\underbrace{V_x x_n + V_h h_{n-1} + c}_{z_n}), \quad \frac{\partial \text{cost}_N}{\partial h_1} = \frac{\partial \text{cost}_N}{\partial \hat{p}_N} \frac{\partial \hat{p}_N}{\partial h_N} \left(\prod_{n \in \{N,\ldots,2\}} \frac{\partial h_n}{\partial z_n} \frac{\partial z_n}{\partial h_{n-1}} \right)$$

$$\frac{\partial h_n}{\partial z_n} = \text{diag}\left(g'(z_n)\right) \qquad \frac{\partial z_n}{\partial h_{n-1}} = V_h$$



[Slide: Phil Blunsom]

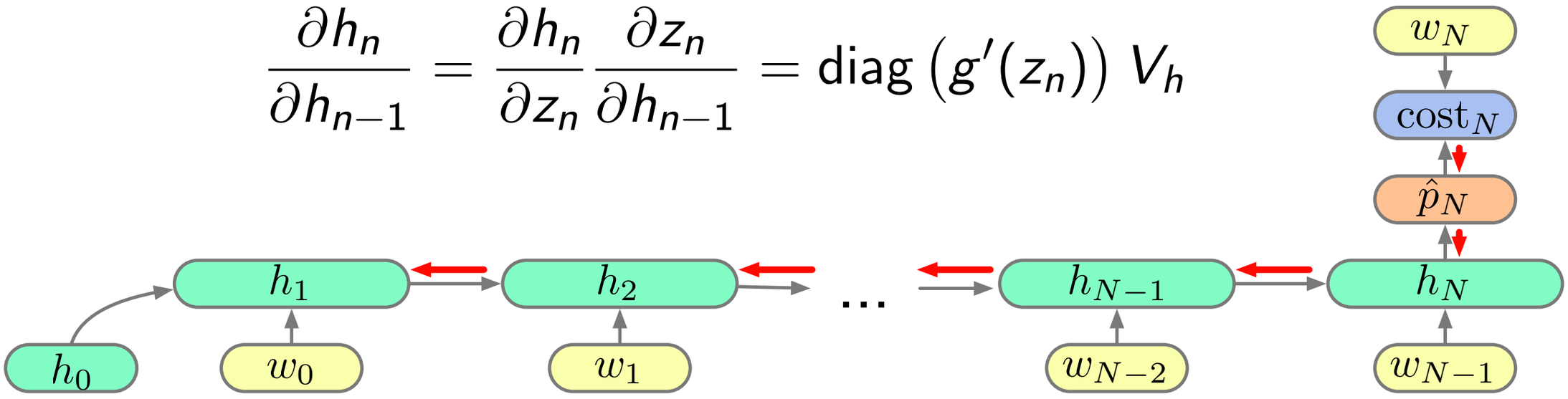
RNNs: Exploding and Vanishing Gradients

Consider the path of partial derivatives linking a change in cost_N to changes in h_1:

$$h_n = g(\underbrace{V_x x_n + V_h h_{n-1} + c}_{z_n}), \quad \frac{\partial \text{cost}_N}{\partial h_1} = \frac{\partial \text{cost}_N}{\partial \hat{p}_N} \frac{\partial \hat{p}_N}{\partial h_N} \left(\prod_{n \in \{N,\ldots,2\}} \frac{\partial h_n}{\partial z_n} \frac{\partial z_n}{\partial h_{n-1}} \right)$$

$$\frac{\partial h_n}{\partial z_n} = \text{diag}\left(g'(z_n)\right) \qquad \frac{\partial z_n}{\partial h_{n-1}} = V_h$$

$$\frac{\partial h_n}{\partial h_{n-1}} = \frac{\partial h_n}{\partial z_n} \frac{\partial z_n}{\partial h_{n-1}} = \text{diag}\left(g'(z_n)\right) V_h$$



[Slide: Phil Blunsom]

RNNs: Exploding and Vanishing Gradients

$$\frac{\partial \text{cost}_N}{\partial h_1} = \frac{\partial \text{cost}_N}{\partial \hat{p}_N} \frac{\partial \hat{p}_N}{\partial h_N} \left(\prod_{n \in \{N,\ldots,2\}} \text{diag}\left(g'(z_n)\right) V_h \right)$$

[Slide: Phil Blunsom]

RNNs: Exploding and Vanishing Gradients

$$\frac{\partial \text{cost}_N}{\partial h_1} = \frac{\partial \text{cost}_N}{\partial \hat{p}_N} \frac{\partial \hat{p}_N}{\partial h_N} \left(\prod_{n \in \{N,\ldots,2\}} \text{diag}\left(g'(z_n)\right) V_h \right)$$

The core of the recurrent product is the repeated multiplication of V_h. If the largest eigenvalue of V_h is:

- 1, then gradient will propagate,
- > 1, the product will grow exponentially (explode),
- < 1, the product shrinks exponentially (vanishes).

[Slide: Phil Blunsom]

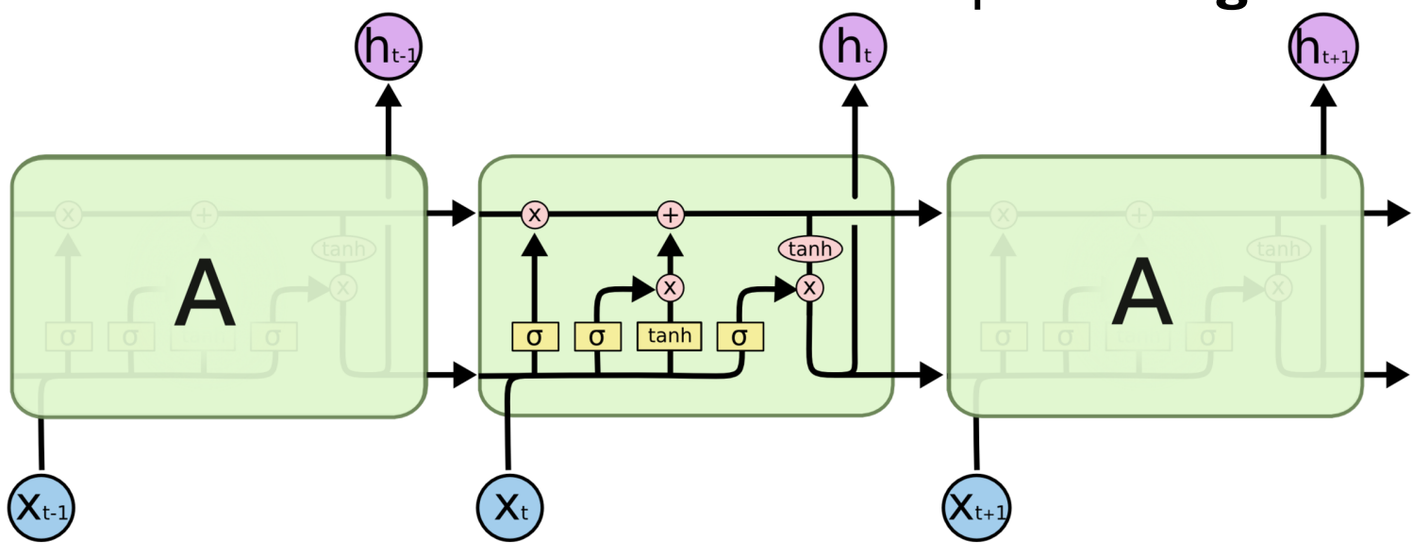
How to learn RNNs

- Learning heuristics
  - Gradient clipping and normalization
  - Architectures that mitigate vanishing gradients
  - *[Later lectures: just let your model just look back further, via attention mechanisms]*

LSTM (Long short-term memory)

- Goals:
  - 1. Be able to "remember" for longer distances
  - 2. Stable backpropagation during training
- Augment individual timesteps with a number of specialized vectors and gating functions. *(There are alternative gated RNNs, but at this point LSTM has won.)*
- Complicated! But maybe the most widely used RNN model.
  - [Newer work: state space models]

- Main state
  - **c**: Memory cell
  - **h**: Hidden state
- Update system
  - **g**: proposed new cell
  - **f, i, o**: Forget, Input, Output gates control acceptance of **g** into new cell & state



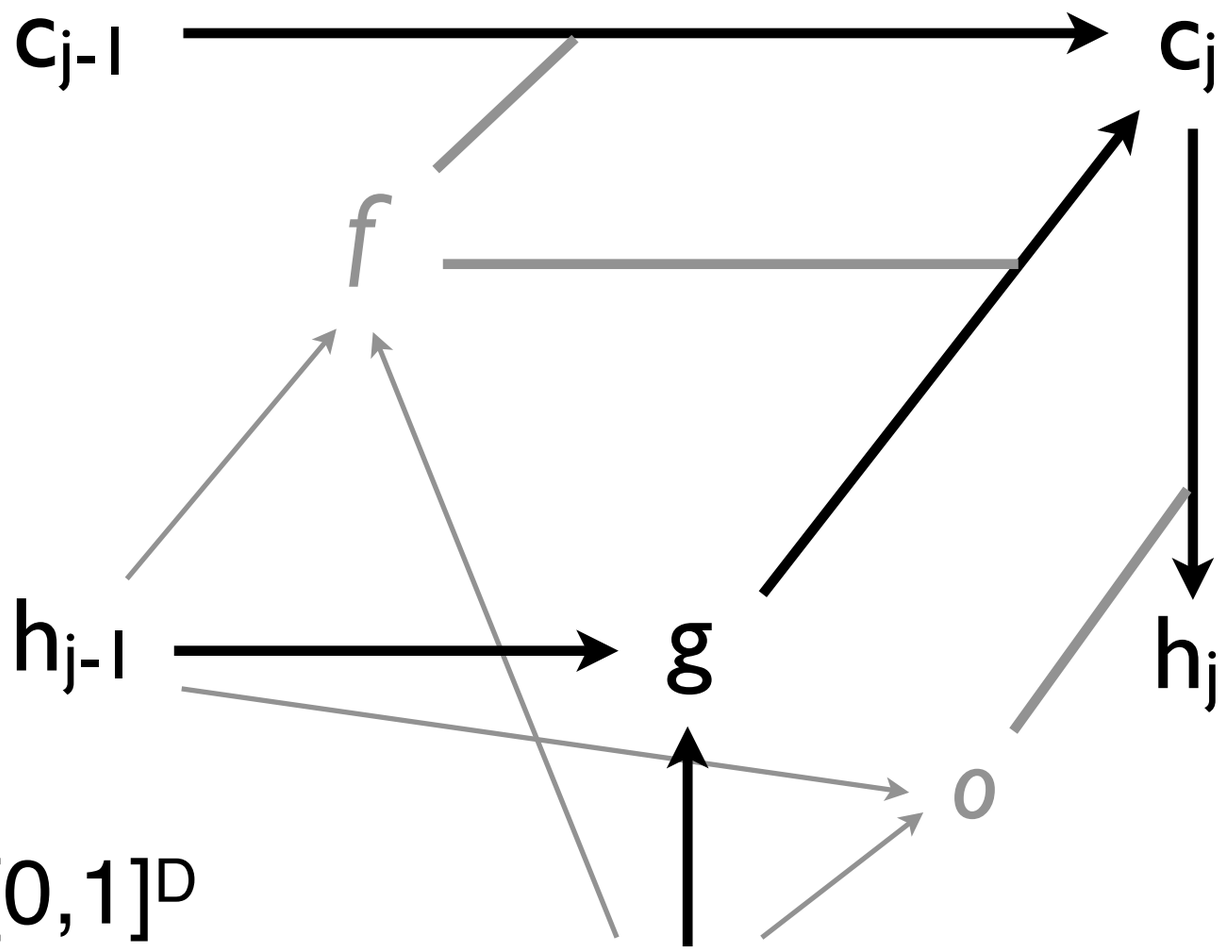
Christopher Olah: Understanding LSTM Networks
`colah.github.io/posts/2015-08-Understanding-LSTMs/`

LSTM

Coupled forget/input,
no peepholes

memory component ("cell")

$$\mathbf{c_j} = \mathbf{c_{j-1}} \odot \mathbf{f} + \mathbf{g} \odot (1 - \mathbf{f})$$



hidden state

$$\mathbf{h_j} = \tanh(\mathbf{c_j}) \odot \mathbf{o}$$

gates $\in [0,1]^D$

$$\mathbf{i} = \sigma(\mathbf{x_j}\mathbf{W^{xi}} + \mathbf{h_{j-1}}\mathbf{W^{hi}})$$

$$\mathbf{f} = \sigma(\mathbf{x_j}\mathbf{W^{xf}} + \mathbf{h_{j-1}}\mathbf{W^{hf}})$$

$$\mathbf{o} = \sigma(\mathbf{x_j}\mathbf{W^{xo}} + \mathbf{h_{j-1}}\mathbf{W^{ho}})$$

proposed cell

$$\mathbf{g} = \tanh(\mathbf{x_j}\mathbf{W^{xg}} + \mathbf{h_{j-1}}\mathbf{W^{hg}})$$

main information

gating function

[Goldberg 2016 notation]

Character LMs comparison

```
PANDARUS:
Alas, I think he shall be come approached and the day
When little srain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.
```

```
First Citizen:
Nay, then, that was hers,
It speaks against your other service:
But since the
youth of the circumstance be spoken:
Your uncle and one Baptista's daughter.

SEBASTIAN:
Do I stand till the break off.

BIRON:
Hide thy head.

VENTIDIUS:
He purposeth to Athens: whither, with the vow
I made to handle you.
```

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

http://nbviewer.jupyter.org/gist/yoavg/d76121dfde2618422139

Structure awareness

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.
```

Cell that turns on inside quotes:

```
"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."
```

Cell that robustly activates inside if statements:

```
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;
}
```

A large portion of cells are not easily interpretable. Here is a typical example:

```
/* 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.
   */
```

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

- Recurrent NNs, esp. LSTMs, can impressively capture many things
- Flexible: can be used either for fully unsupervised LMs, or as fully supervised classification or tagging

- ... But they struggle with longer distance and structural effects