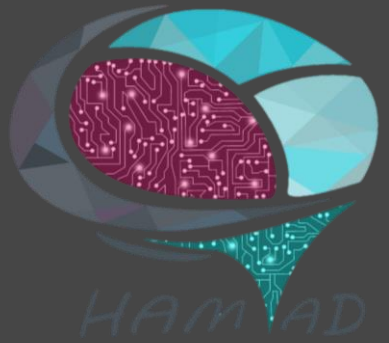


مدل‌سازی زبان
کارگاه یادگیری عمیق با پایتون

سید ناصر رضوی www.snrazavi.ir
۱۳۹۷

فهرست مطالب: پردازش زبان طبیعی

- دسته‌بندی متون
- مدل‌سازی زبان
- عنوان‌بندی تصاویر
- ترجمه ماشینی

آن محبت گفت در جستن نوا — تو مرا در شاه او شد بر سما
تا ابد داری که آیم سوی مرد — تا فراق او علامت‌های رخت
ترک پر از جاه او آگاه کرد — با همان خفاش شادی بیش دید
باد چون بستی ترا تاسه شود — پوست را از تازی یزدان شاه
زانک دندان است آن زهر پلید — دیو را بر گاو عزم انداختند
شادمانان و شتابان سوی ده — که بری خوردیم از ده مژده ده
گفت توبه کردم ای سلطان که من — وقت دولت رفت و شد آن را و شرم
این جهان نه این جهان بالاترم — همچو بینش جانب ده می‌فتاد
لیک در تجرید از تن راندند — نام آن گرگش ندرد یا ددش
ای خنک آن را که بیند روی تو — یا ز تلخی‌ها همه بیرون برون
حال ایشان هست کو از زخم دور — لیک پیش از نور و در وی خورد ستیز

فهرست مطالب: این جلسه

- مدل‌های زبانی.
  - کاربردها و رویکردها
- شبکه‌های عصبی برگشتی.
  - آموزش و نمونه‌برداری
- مشکل انفجار گرادیان و محو گرادیان.
  - حافظه کوتاه-مدت طولانی
- شبکه‌های برگشتی عمیق.
- اندازه واژگان.
  - مدل‌سازی زبان در سطح زیرکلمه و کاراکتر
- تنظیم در شبکه‌های برگشتی.
  - دورریزی (دراپ‌آوت) و دورریزی بیزی

مدل‌سازی زبان

- مدل‌سازی زبان. انتساب یک احتمال به هر دنباله از کلمات، به گونه‌ای که مجموع احتمال‌ها بر روی تمام دنباله‌های ممکن برابر با ۱ شود.

$$P(w_1, w_2, \dots, w_n) = P(w_1)P(w_2|w_1)P(w_3|w_1, w_2) \times \cdots \times P(w_n|w_1, w_2, \dots, w_{n-1})$$

- کاربردها.

$p(\text{he likes apple}) > p(\text{apple likes he})$

$p(\text{he likes apple}) > p(\text{he licks apple})$

رویکردها

- مدل‌های N-بخشی [مبتنی بر شمارش]
  - تخمین تاریخچه کلمات مشاهده شده تنها با استفاده از N - 1 کلمه قبل

مدل دو-بخشی ←

$$P(w_1, w_2, \dots, w_n) \approx P(w_1)P(w_2|w_1)P(w_3|w_2) \times \cdots \times P(w_n|w_{n-1})$$

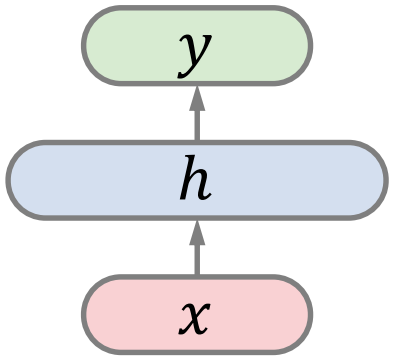
$$P(w_i|w_{i-1}) = \frac{P(w_{i-1}, w_i)}{P(w_{i-1})} = \frac{N(w_{i-1}, w_i)}{N(w_{i-1})}$$

- مدل‌های N-بخشی [مبتنی بر شبکه‌های عصبی]
  - تخمین تاریخچه کلمات مشاهده شده تنها با استفاده از N - 1 کلمه قبل در یک فضای پیوسته و یادگیری بهتر وابستگی (ارتباط معنایی) میان کلمات.

- شبکه‌های عصبی برگشتی.
  - فشرده‌سازی تاریخچه کامل کلمات مشاهده شده در یک بردار با اندازه ثابت و یادگیری وابستگی‌های بلندمدت میان کلمات.

مدل‌های مبتنی بر شبکه‌های عصبی

- شبکه عصبی. [با یک لایه مخفی]



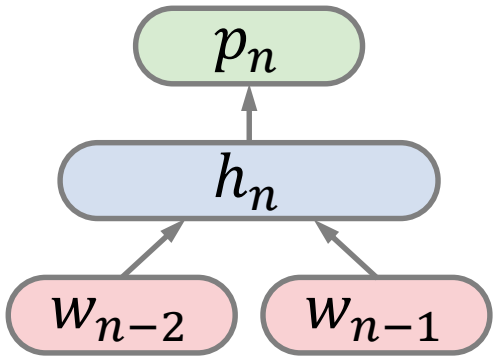
$$y = W_y h + b_y$$

$$h = f(W_h x + b_h)$$

$$x = [w_1; w_2; \ldots; w_{n-1}]$$ تاریخچه

مدل‌های مبتنی بر شبکه‌های عصبی

- مدل سه-بخشی. تخمین احتمال کلمه بعدی با داشتن ۲ کلمه قبلی.



$$p_n = softmax(W_y\ h_n + b_y)$$

$$h_n = f(W_h\ [w_{n-2}; w_{n-1}] + b_h)$$

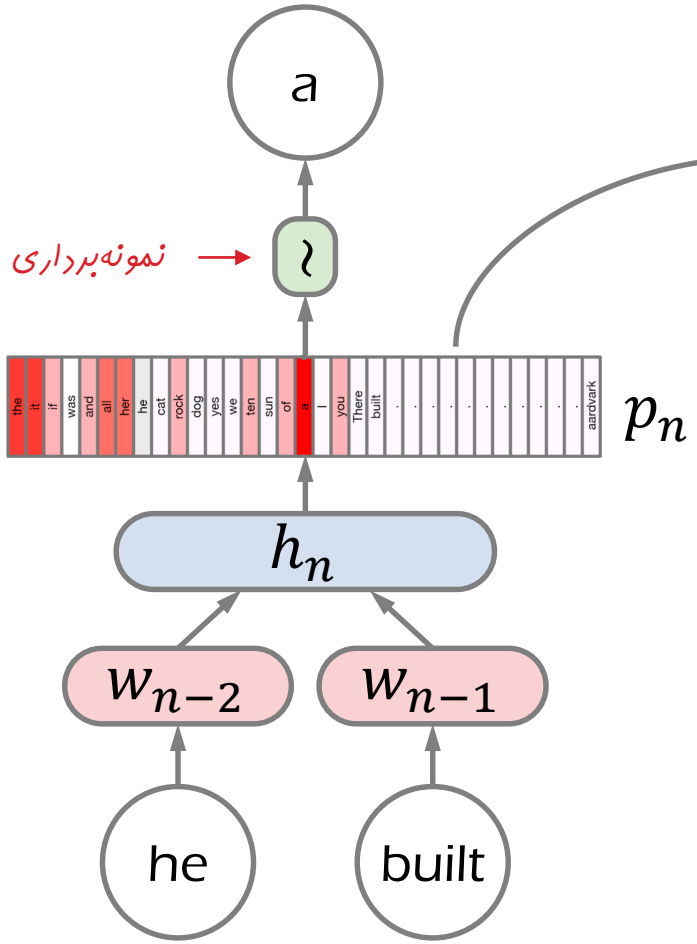
$$x_n = [w_{n-2}; w_{n-1}]$$

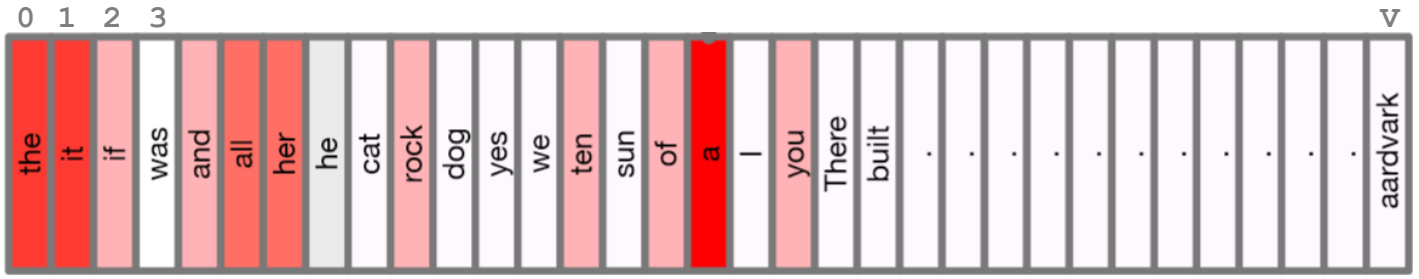
- بازنمایی کلمات با استفاده از روش بازنمایی one-hot

$$|w_i| = |p_i| = |V|$$

- تعداد کلمات مجموعه واژگان (V). معمولا بسیار بزرگ [چند ده هزار تا چند صد هزار]

مدل‌های مبتنی بر شبکه‌های عصبی: نمونه‌برداری



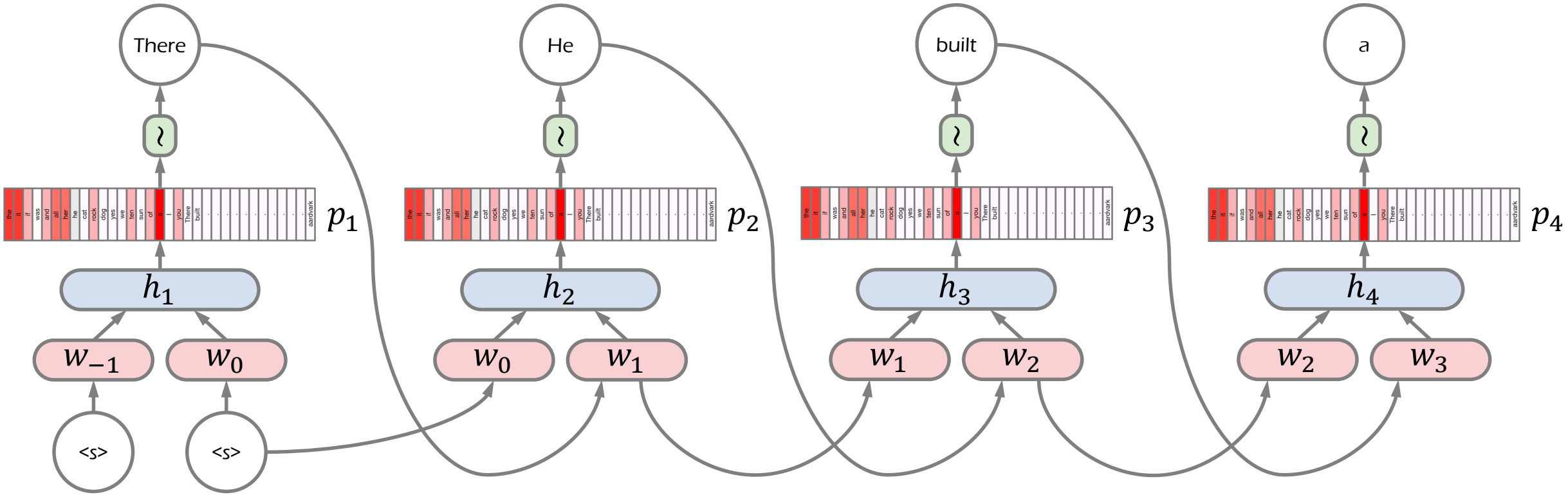


انتخاب عنصر واقع در اندیس صفر از تنسور خروجی

```
id = torch.multinomial(probs, 1)[0]
```

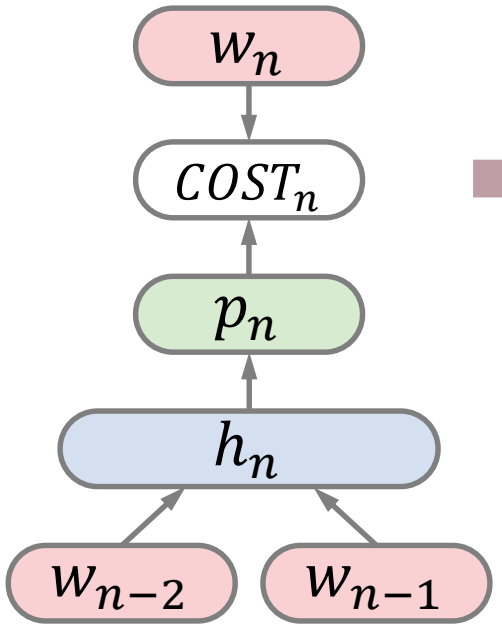
تعداد نمونه‌ها

مدل‌های مبتنی بر شبکه‌های عصبی: نمونه‌برداری



مدل‌های مبتنی بر شبکه‌های عصبی: آموزش

- تابع هزینه. آنتروپی متقابل [بیشینه‌سازی احتمال کلمه درست]



بردار one-hot

بردار توزیع احتمالات

$$COST_n(w_n, p_n) = -(w_n^T \cdot \log p_n)$$

هزینه تخمین یک کلمه

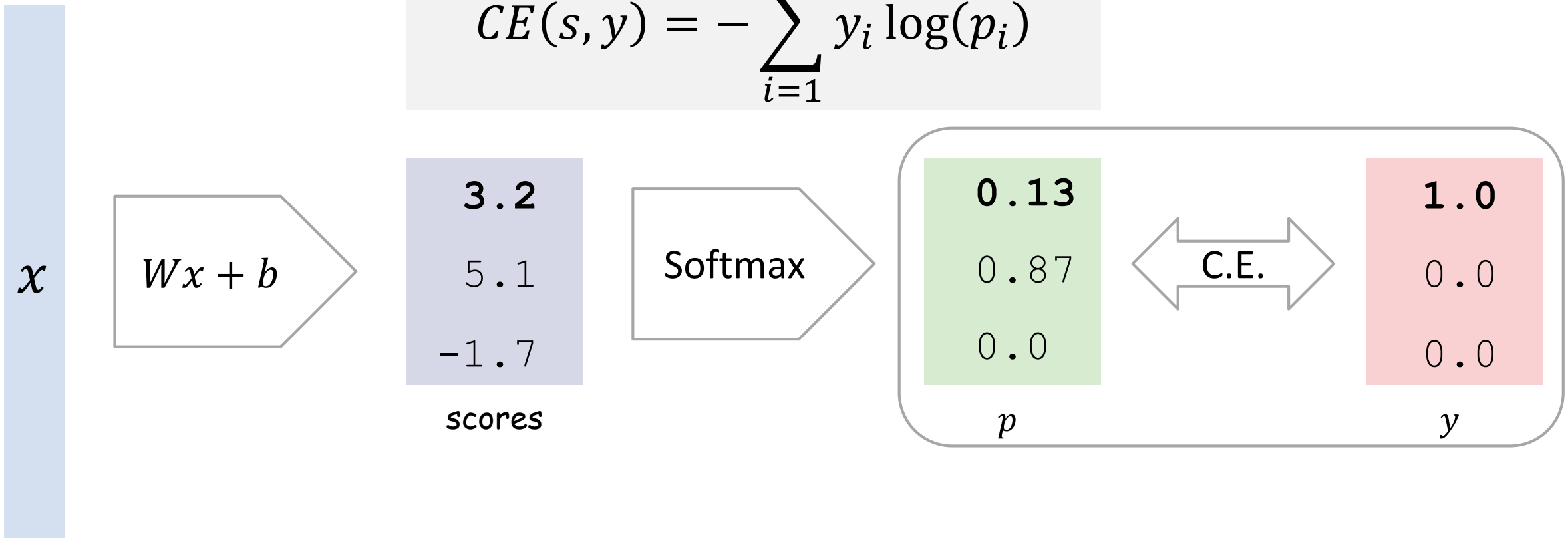
$$\mathcal{F} = -\frac{1}{N}\sum_n COST_n(w_n, p_n)$$

هزینه تخمین یک دنباله

https://www.youtube.com/watch?v=D2EFRXhJ8LM

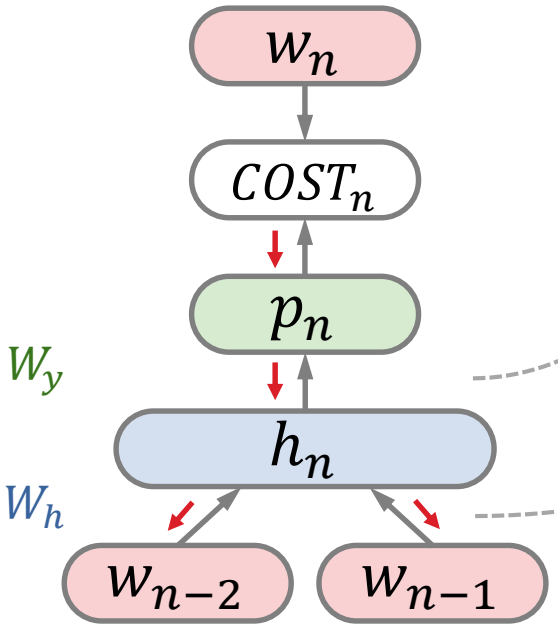
یادآوری: تابع هزینه آنتروپی متقابل

$$CE(s, y) = -\sum_{i=1}^{C} y_i \log(p_i)$$



مدل‌های مبتنی بر شبکه‌های عصبی: آموزش

- پس‌انتشار خطا.



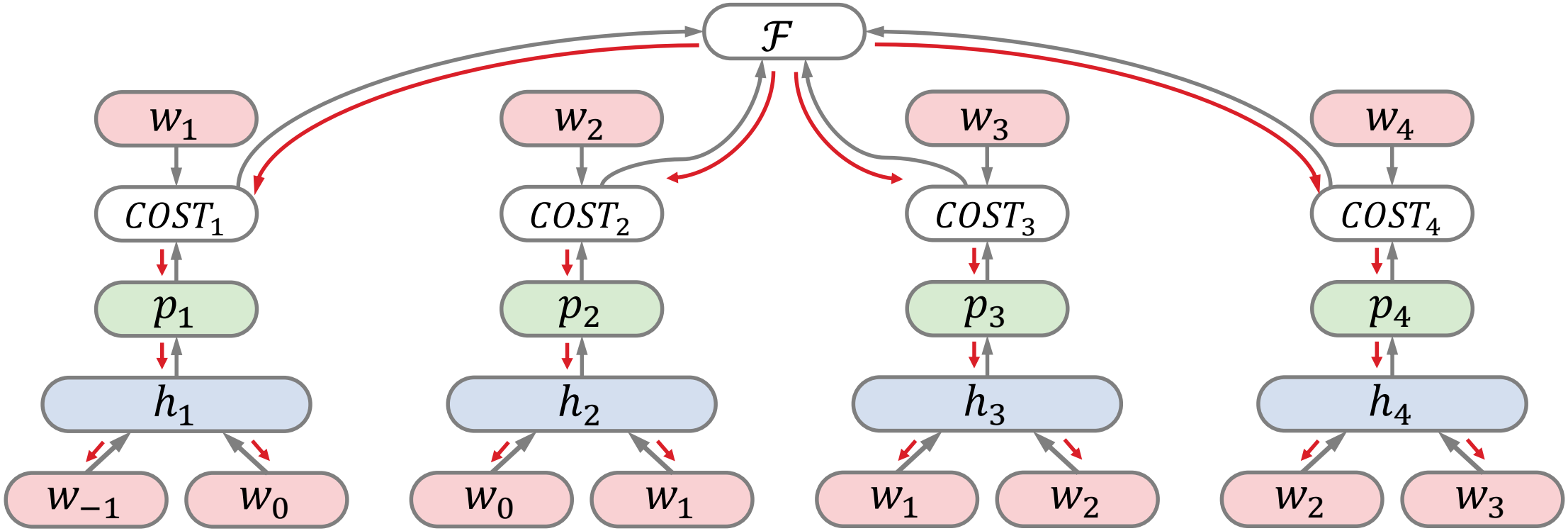
$$\frac{\partial \mathcal{F}}{\partial W_y} = -\frac{1}{N}\sum_n \frac{\partial COST_n}{\partial p_n} \cdot \frac{\partial p_n}{\partial W_y}$$

$$\frac{\partial \mathcal{F}}{\partial W_h} = -\frac{1}{N}\sum_n \frac{\partial COST_n}{\partial p_n} \cdot \frac{\partial p_n}{\partial h_n} \cdot \frac{\partial h_n}{\partial W_h}$$

https://www.youtube.com/watch?v=9JKXFWzf0yc

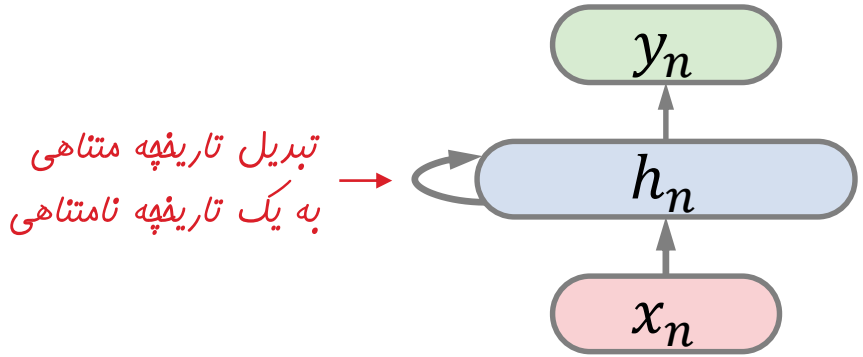
مدل‌های مبتنی بر شبکه‌های عصبی: آموزش

- پس‌انتشار. محاسبه گرادیان‌ها در هر مرحله زمانی، مستقل از مراحل دیگر است و بنابراین گرادیان‌ها در مراحل زمانی مختلف می‌توانند به صورت موازی محاسبه شده و سپس با یکدیگر جمع شوند.



مدل‌های زبانی مبتنی بر شبکه‌های برگشتی

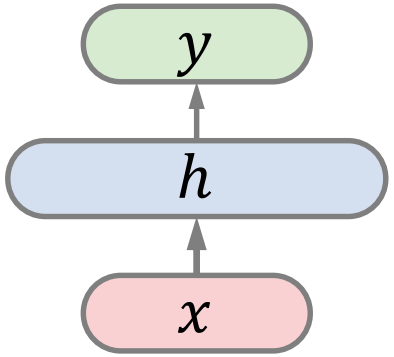
شبکه برگشتی



$$h_n = f(W_h\,[x_n; h_{n-1}] + b_h)$$

$$y = W_y\, h_n + b_y$$

شبکه پیش‌خور

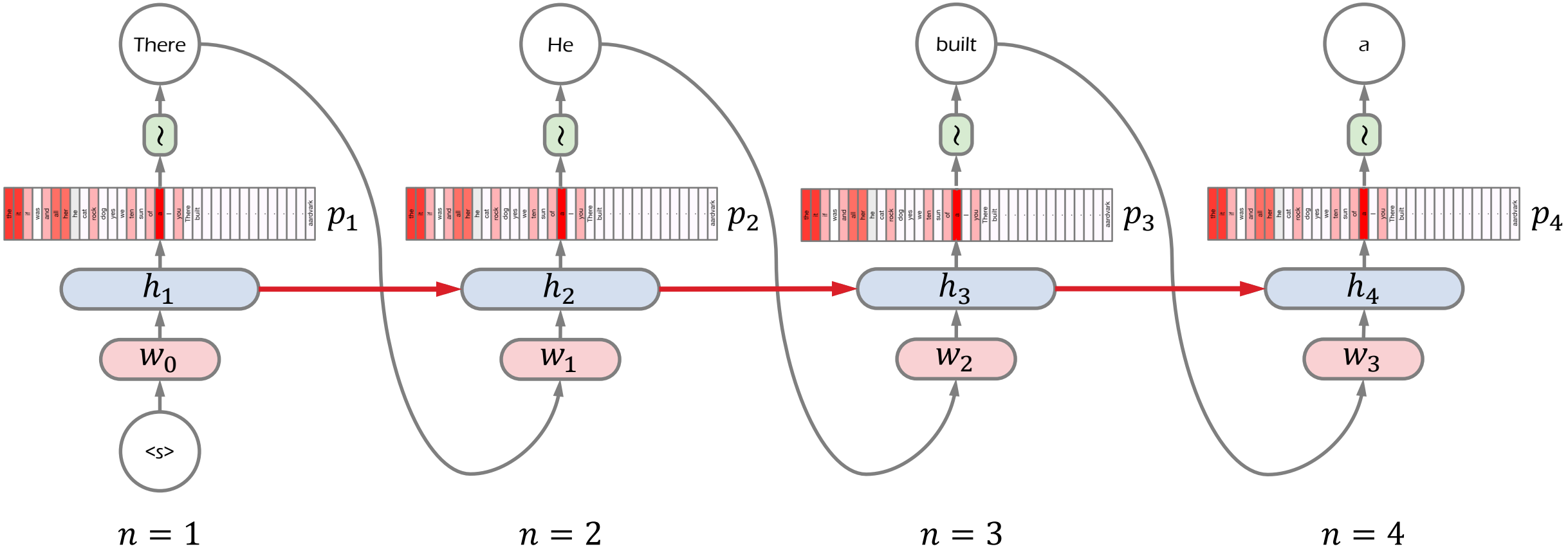


$$h = f(W_h\, x + b_h)$$

$$y = W_y\, h + b_y$$

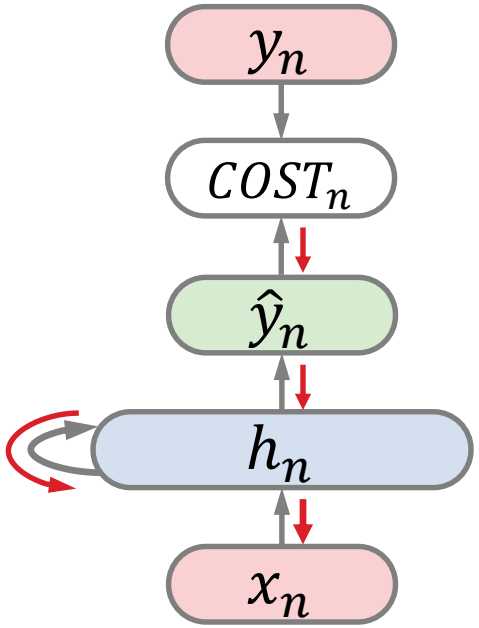
مدل‌های زبانی مبتنی بر شبکه‌های برگشتی: نمونه‌برداری

$$h_n = f(W_h\,[x_n; h_{n-1}] + b_h) \qquad x_n = w_{n-1} \rightarrow y_n = w_n = x_{n+1}$$

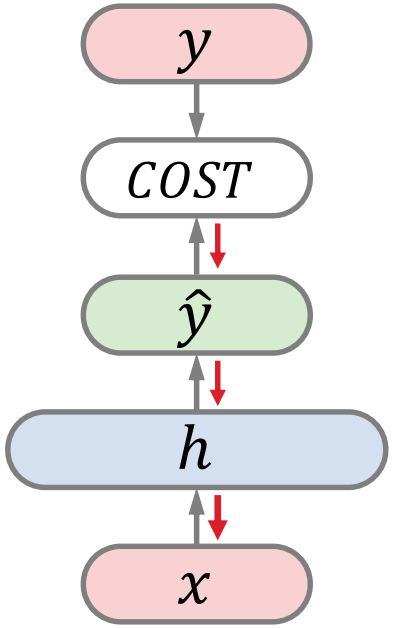


مدل‌های زبانی مبتنی بر شبکه‌های برگشتی: آموزش

شبکه برگشتی

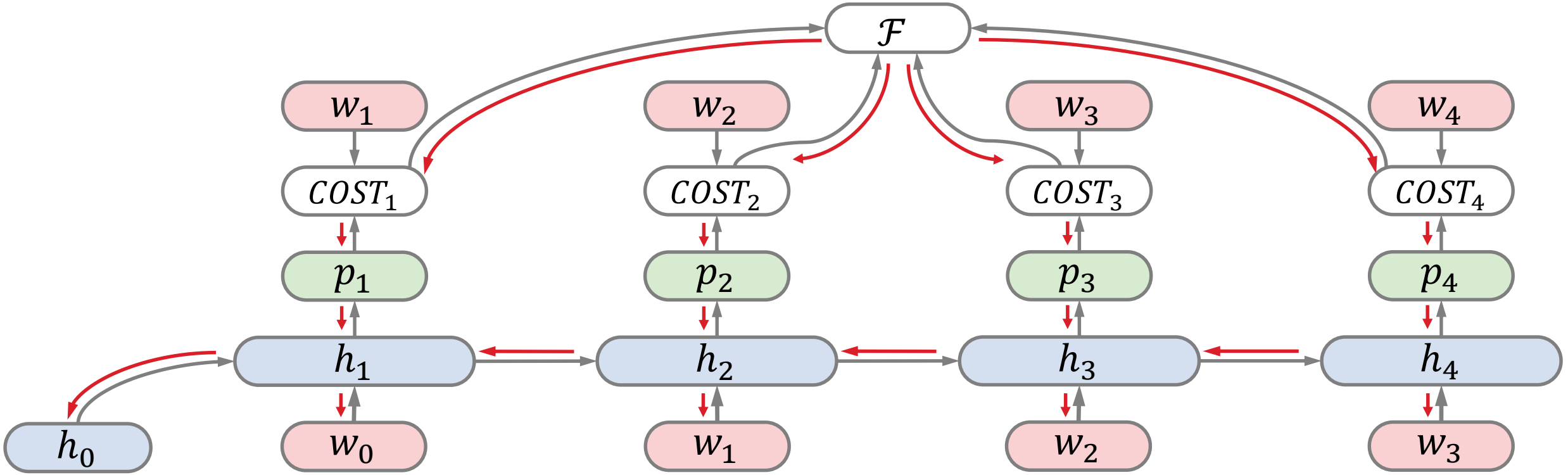


شبکه پیش‌خور



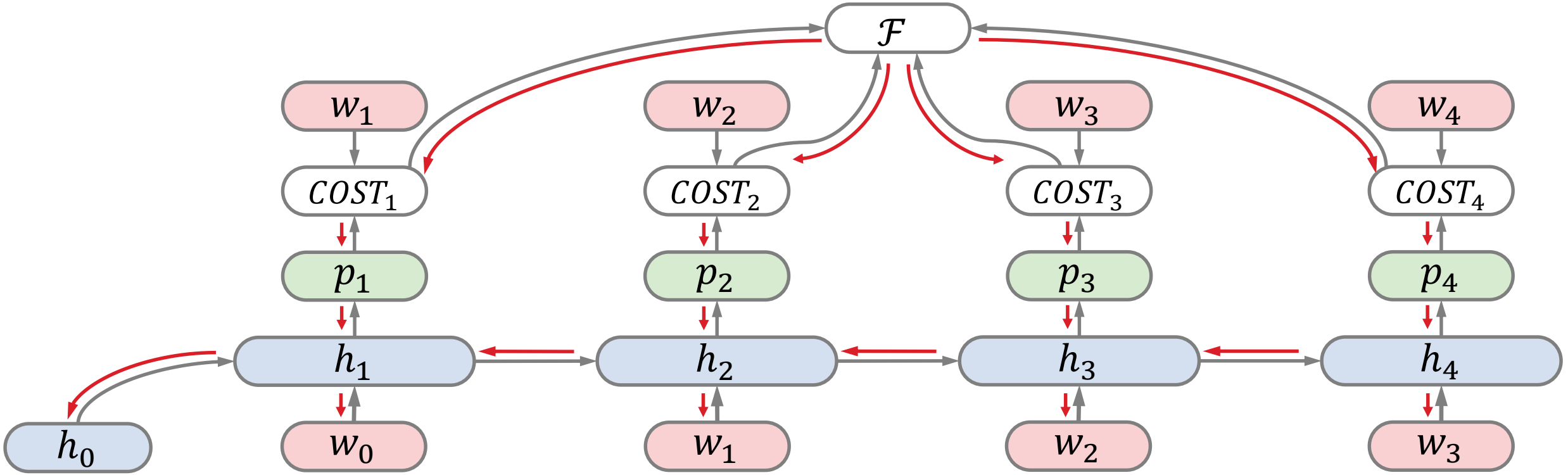
مدل‌های زبانی مبتنی بر شبکه‌های برگشتی: آموزش

- پس‌انتشار. شبکه برگشتی باز شده، یک گراف جهت‌دار بدون دور است. بنابراین، می‌توانیم به طور معمول از الگوریتم پس‌انتشار برای محاسبه گرادیان‌ها استفاده کنیم.



مدل‌های زبانی مبتنی بر شبکه‌های برگشتی: آموزش

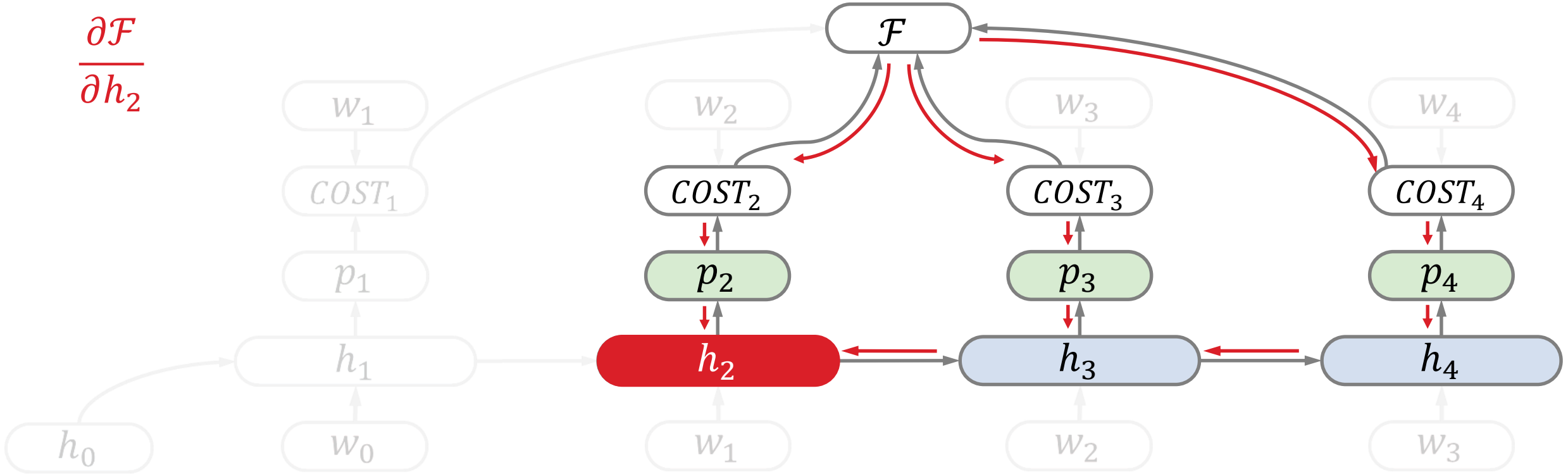
- پس‌انتشار. این الگوریتم، پس‌انتشار در طول زمان نام دارد. توجه داشته باشید که گرادیان‌ها در لحظه n به گرادیان‌ها در لحظه $n + \alpha$ بستگی دارند.



مدل‌های زبانی مبتنی بر شبکه‌های برگشتی: آموزش

- پس‌انتشار. این الگوریتم، پس‌انتشار در طول زمان نام دارد. توجه داشته باشید که گرادیان‌ها در لحظه n به گرادیان‌ها در لحظه $n+\alpha$ بستگی دارند.

$$\frac{\partial \mathcal{F}}{\partial h_2}$$

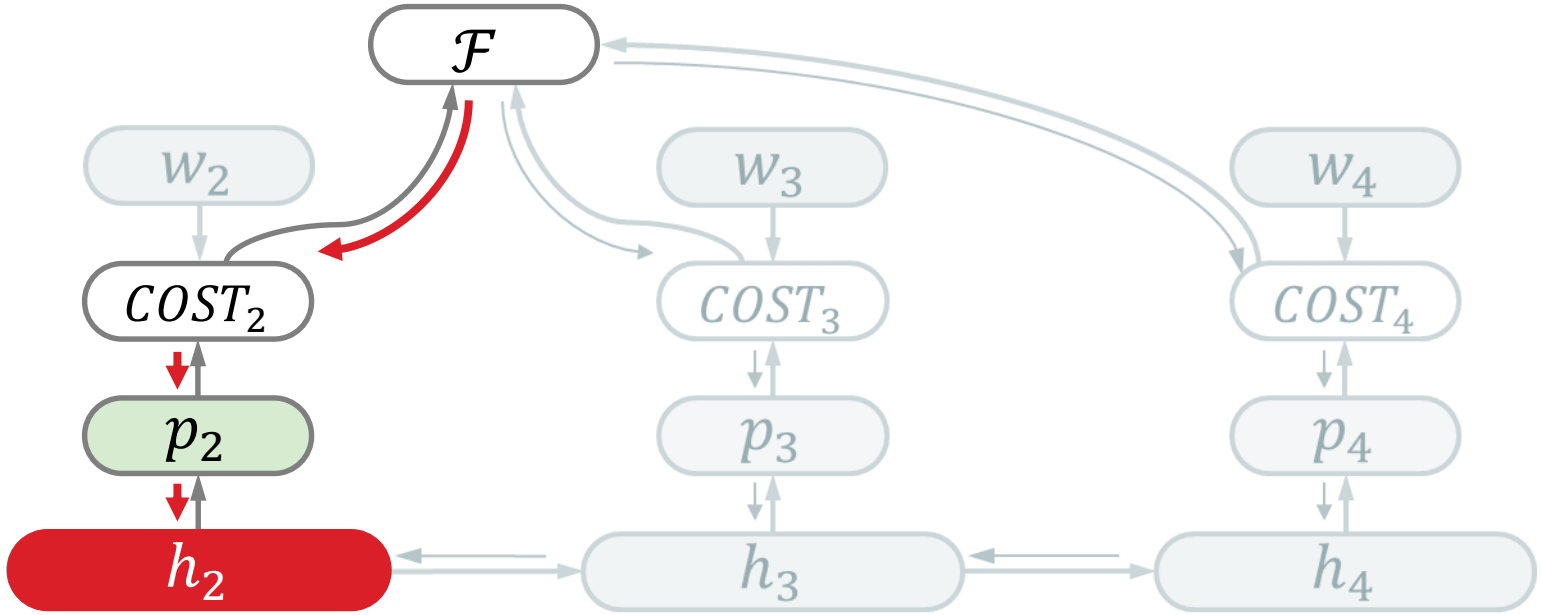


مدل‌های زبانی مبتنی بر شبکه‌های برگشتی: آموزش

- پس‌انتشار. این الگوریتم، پس‌انتشار در طول زمان نام دارد. توجه داشته باشید که گرادیان‌ها در لحظه n به گرادیان‌ها در لحظه $n+\alpha$ بستگی دارند.

$$\frac{\partial \mathcal{F}}{\partial h_2}$$

$$= \frac{\partial \mathcal{F}}{\partial COST_2} \cdot \frac{\partial COST_2}{\partial p_2} \cdot \frac{\partial p_2}{\partial h_2}$$



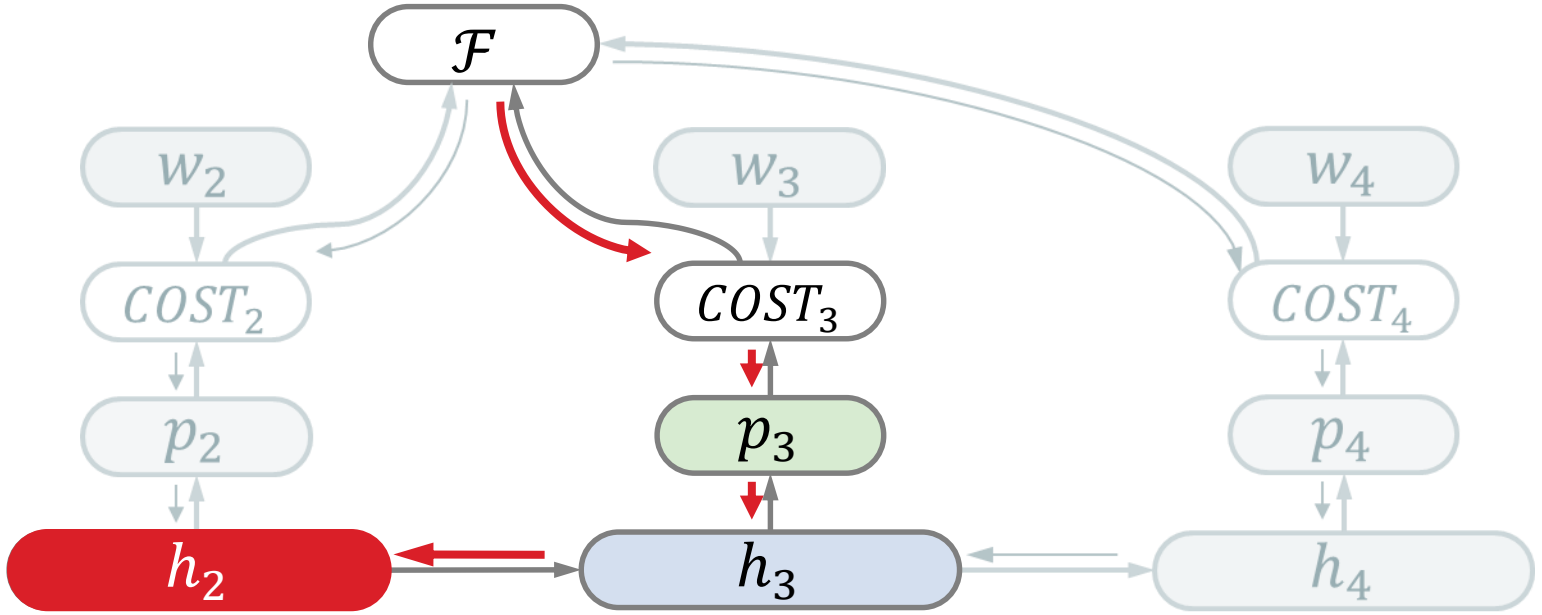
مدل‌های زبانی مبتنی بر شبکه‌های برگشتی: آموزش

- پس‌انتشار. این الگوریتم، پس‌انتشار در طول زمان نام دارد. توجه داشته باشید که گرادیان‌ها در لحظه n به گرادیان‌ها در لحظه $n+\alpha$ بستگی دارند.

$$\frac{\partial \mathcal{F}}{\partial h_2}$$

$$= \frac{\partial \mathcal{F}}{\partial COST_2} \cdot \frac{\partial COST_2}{\partial p_2} \cdot \frac{\partial p_2}{\partial h_2}$$

$$+ \frac{\partial \mathcal{F}}{\partial COST_3} \cdot \frac{\partial COST_3}{\partial p_3} \cdot \frac{\partial p_3}{\partial h_3} \cdot \frac{\partial h_3}{\partial h_2}$$



مدل‌های زبانی مبتنی بر شبکه‌های برگشتی: آموزش

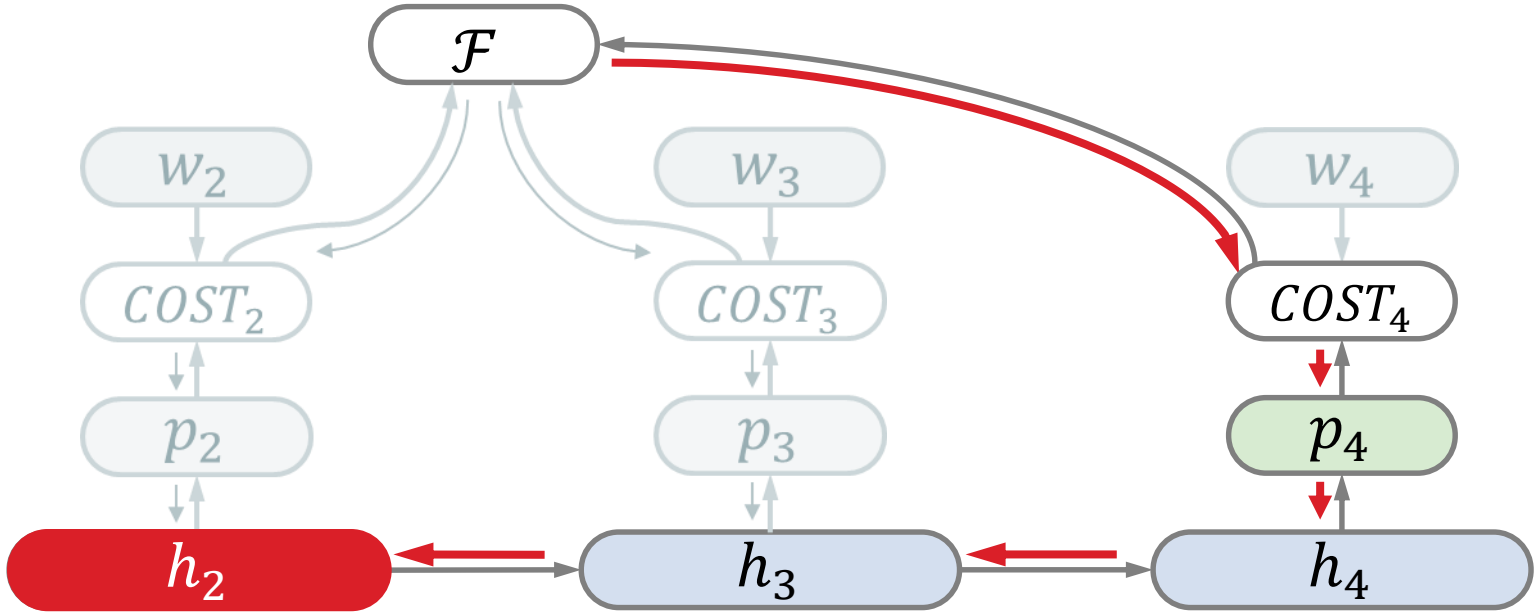
- پس‌انتشار. این الگوریتم، پس‌انتشار در طول زمان نام دارد. توجه داشته باشید که گرادیان‌ها در لحظه n به گرادیان‌ها در لحظه $n+\alpha$ بستگی دارند.

$$\frac{\partial \mathcal{F}}{\partial h_2}$$

$$= \frac{\partial \mathcal{F}}{\partial COST_2} \cdot \frac{\partial COST_2}{\partial p_2} \cdot \frac{\partial p_2}{\partial h_2}$$

$$+ \frac{\partial \mathcal{F}}{\partial COST_3} \cdot \frac{\partial COST_3}{\partial p_3} \cdot \frac{\partial p_3}{\partial h_3} \cdot \frac{\partial h_3}{\partial h_2}$$

$$+ \frac{\partial \mathcal{F}}{\partial COST_4} \cdot \frac{\partial COST_4}{\partial p_4} \cdot \frac{\partial p_4}{\partial h_4} \cdot \frac{\partial h_4}{\partial h_3} \cdot \frac{\partial h_3}{\partial h_2}$$



مدل‌های زبانی مبتنی بر شبکه‌های برگشتی: آموزش

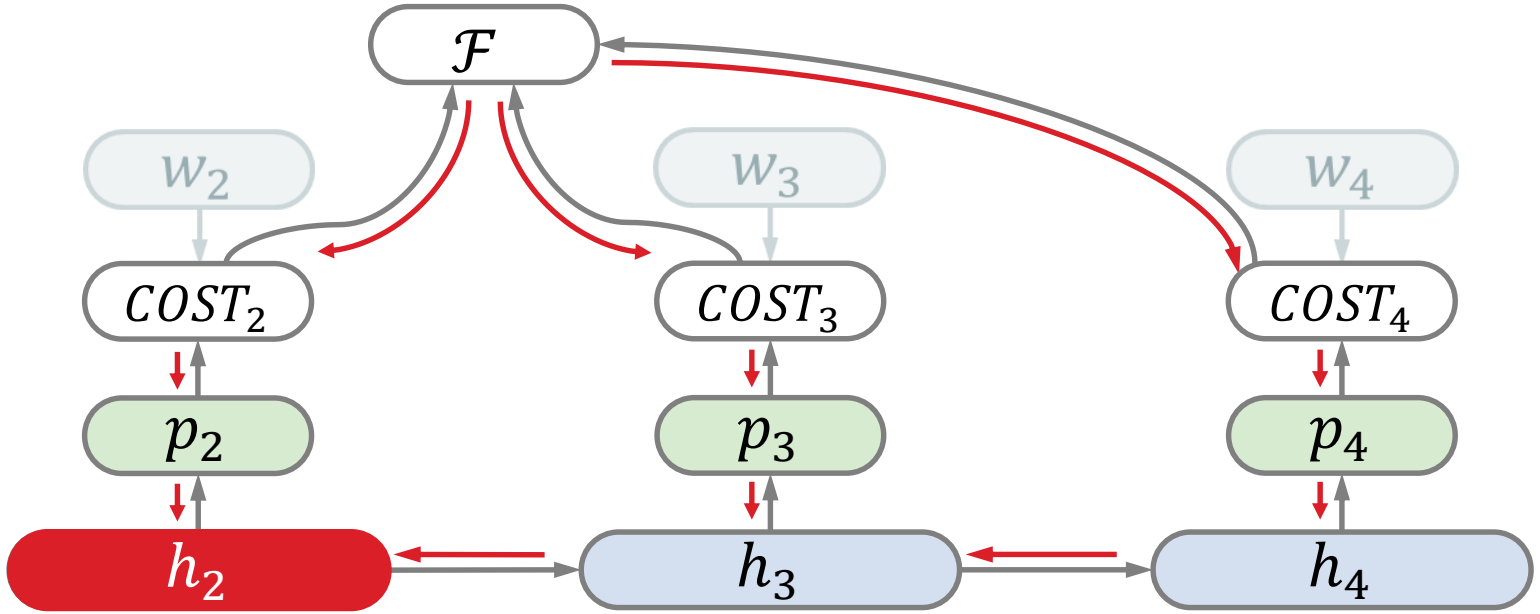
- پس‌انتشار. این الگوریتم، پس‌انتشار در طول زمان نام دارد. توجه داشته باشید که گرادیان‌ها در لحظه n به گرادیان‌ها در لحظه $n+\alpha$ بستگی دارند.

$$\frac{\partial \mathcal{F}}{\partial h_2}$$

$$= \frac{\partial \mathcal{F}}{\partial COST_2} \cdot \frac{\partial COST_2}{\partial p_2} \cdot \frac{\partial p_2}{\partial h_2}$$

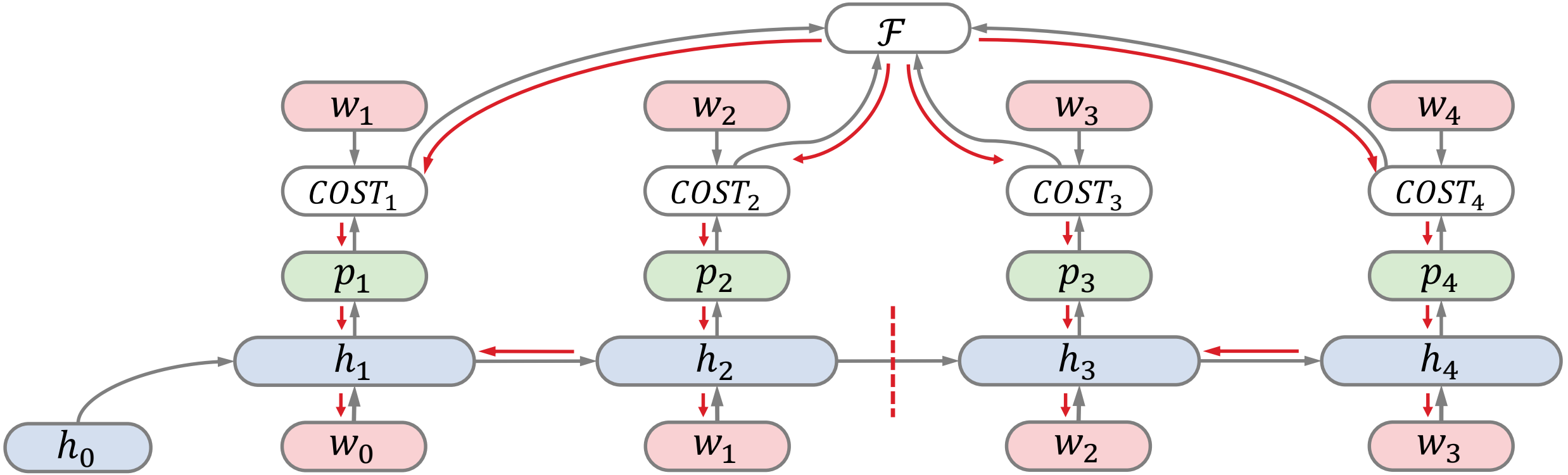
$$+ \frac{\partial \mathcal{F}}{\partial COST_3} \cdot \frac{\partial COST_3}{\partial p_3} \cdot \frac{\partial p_3}{\partial h_3} \cdot \frac{\partial h_3}{\partial h_2}$$

$$+ \frac{\partial \mathcal{F}}{\partial COST_4} \cdot \frac{\partial COST_4}{\partial p_4} \cdot \frac{\partial p_4}{\partial h_4} \cdot \frac{\partial h_4}{\partial h_3} \cdot \frac{\partial h_3}{\partial h_2}$$



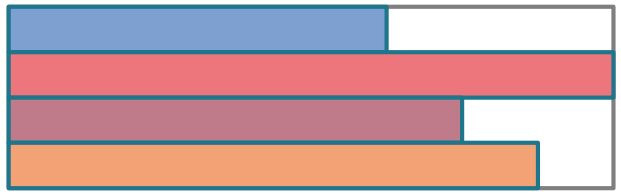
مدل‌های زبانی مبتنی بر شبکه‌های برگشتی: آموزش

- پس‌انتشار. اگر این وابستگی‌ها را پس از یک تعداد ثابت از مراحل زمانی برش بزنیم؛ الگوریتم حاصل، الگوریتم پس‌انتشار برش خورده در طول زمان نام دارد.



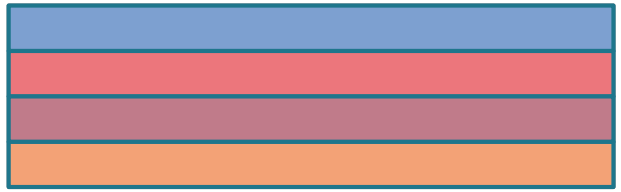
پردازش دسته‌ای و شبکه‌های برگشتی

- پس‌انتشار در طول زمان (BPTT).



  - پیچیدگی زمانی یک تابع خطی از طول بلندترین دنباله است.
  - از آنجا که جملات قرار گرفته در یک دسته ممکن است طول‌های متفاوتی داشته باشند، پردازش دسته‌ای می‌تواند ناکارا باشد.

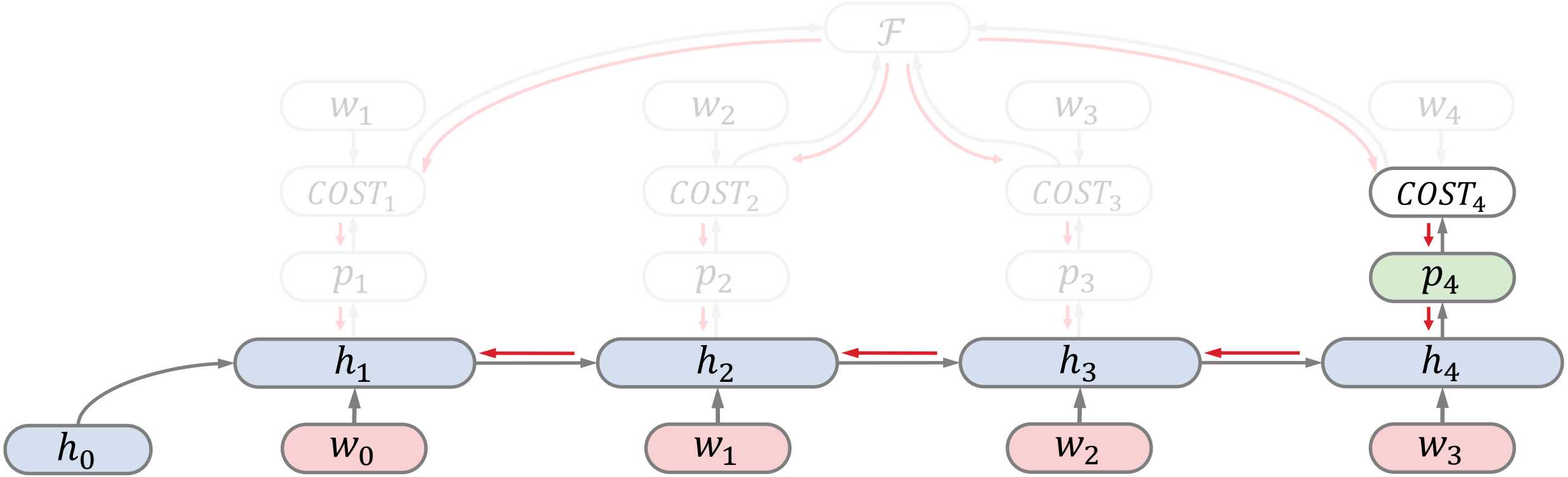
- پس‌انتشار برش خورده در طول زمان (TBPTT).



  - پیچیدگی زمانی برحسب طول برش ثابت است.
  - از آنجا که همه جملات قرار گرفته در یک دسته طول‌های یکسانی دارند، این روش برای پردازش دسته‌ای مناسب‌تر است.

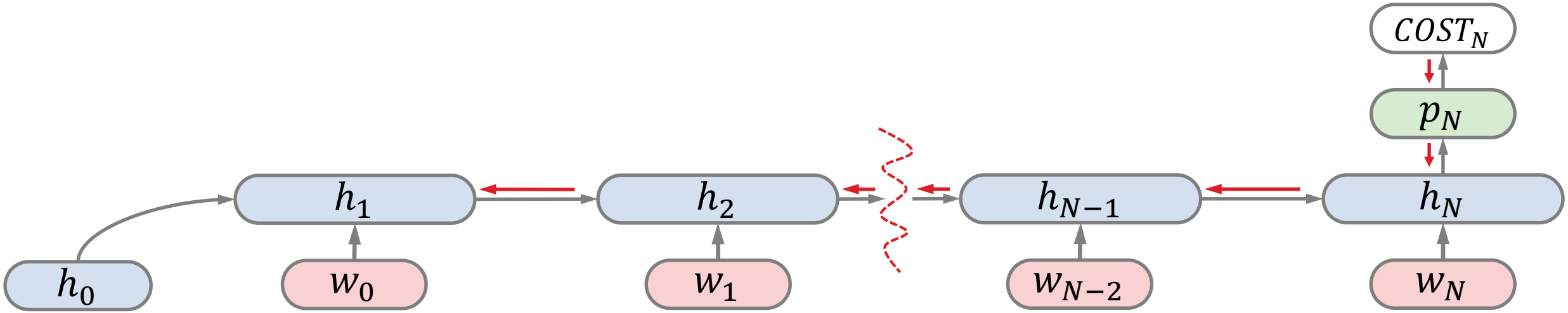
شبکه‌های برگشتی: انفجار و محو گرادیان

$$\frac{\partial COST_4}{\partial h_1} = \frac{\partial COST_4}{\partial p_4} \cdot \frac{\partial p_4}{\partial h_4} \cdot \frac{\partial h_4}{\partial h_3} \cdot \frac{\partial h_3}{\partial h_2} \cdot \frac{\partial h_2}{\partial h_1}$$



شبکه‌های برگشتی: انفجار و محو گرادیان

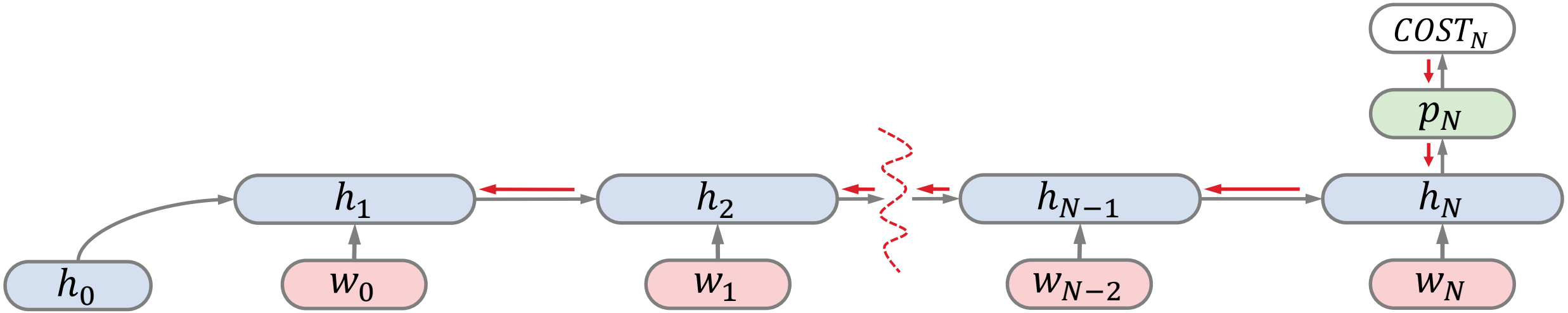
$$\frac{\partial COST_N}{\partial h_1} = \frac{\partial COST_N}{\partial p_N} \cdot \frac{\partial p_N}{\partial h_N} \cdot \left(\frac{\partial h_N}{\partial h_{N-1}} \cdot \frac{\partial h_{N-1}}{\partial h_{N-2}} \cdots \frac{\partial h_2}{\partial h_1} \right)$$



شبکه‌های برگشتی: انفجار و محو گرادیان

$$\frac{\partial COST_N}{\partial h_1} = \frac{\partial COST_N}{\partial p_N} \cdot \frac{\partial p_N}{\partial h_N} \cdot \left(\prod_{n=2}^{N} \frac{\partial h_n}{\partial h_{n-1}} \right)$$

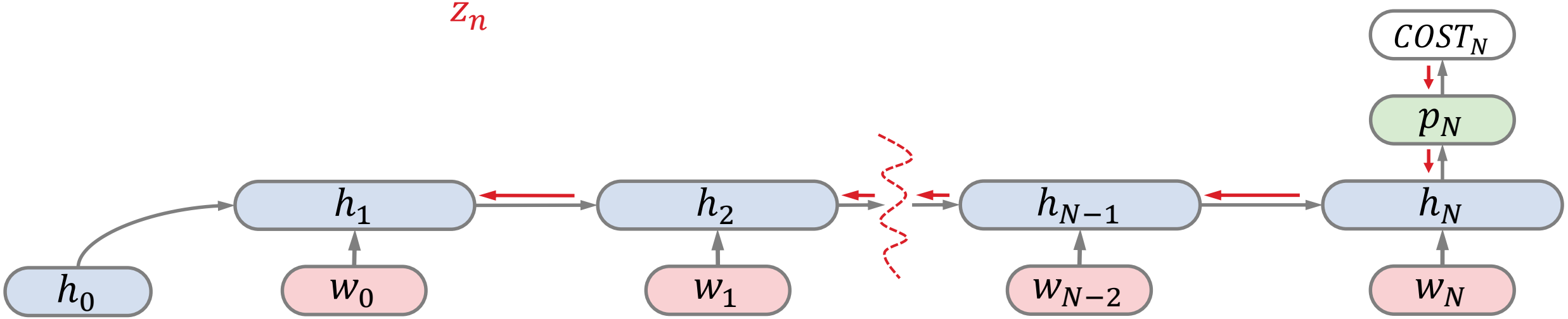
$$h_n = f(W_h \, [x_n; h_{n-1}] + b_h)$$



شبکه‌های برگشتی: انفجار و محو گرادیان

$$\frac{\partial COST_N}{\partial h_1} = \frac{\partial COST_N}{\partial p_N} \cdot \frac{\partial p_N}{\partial h_N} \cdot \left(\prod_{n=2}^{N} \frac{\partial h_n}{\partial h_{n-1}} \right)$$

$$h_n = f(\underbrace{W_{xh}\, x_n + W_{hh}\, h_{n-1} + b_h}_{z_n})$$



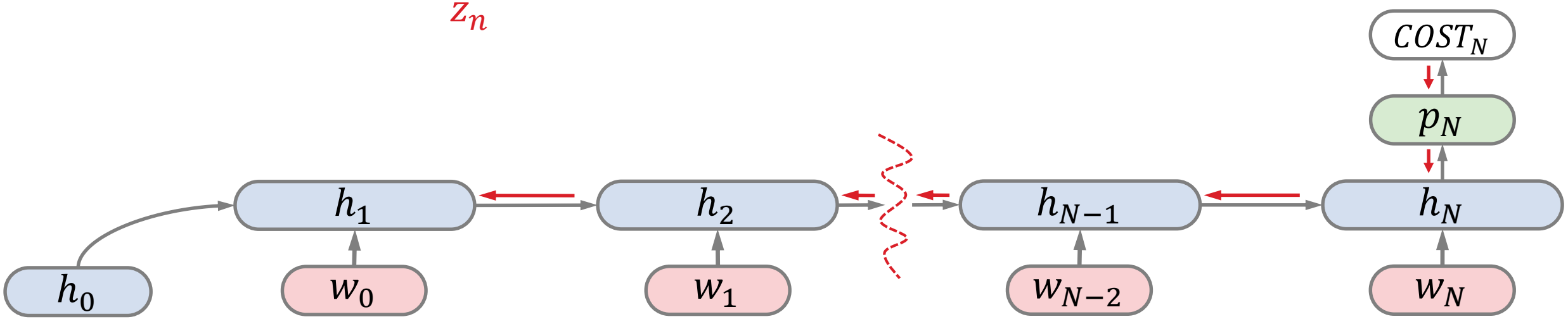
شبکه‌های برگشتی: انفجار و محو گرادیان

$$\frac{\partial COST_N}{\partial h_1} = \frac{\partial COST_N}{\partial p_N} \cdot \frac{\partial p_N}{\partial h_N} \cdot \left(\prod_{n=2}^{N} \frac{\partial h_n}{\partial z_n} \cdot \frac{\partial z_n}{\partial h_{n-1}} \right)$$

$$\frac{\partial h_n}{\partial z_n} = diag\left(f'(z_n)\right)$$

$$\frac{\partial z_n}{\partial h_{n-1}} = W_{hh}$$

$$h_n = f(\underbrace{W_{xh}\, x_n + W_{hh}\, h_{n-1} + b_h}_{z_n})$$



شبکه‌های برگشتی: انفجار و محو گرادیان

$$\frac{\partial COST_N}{\partial h_1} = \frac{\partial COST_N}{\partial p_N} \cdot \frac{\partial p_N}{\partial h_N} \cdot \left(\prod_{n=2}^{N} diag\left(f'(z_n)\right) \cdot W_{hh} \right) \qquad \left(\prod_{n=2}^{N} diag\left(f'(z_n)\right) \right) (W_{hh})^{N-1}$$

□ مشاهده. هسته اصلی در محاسبه گرادیان‌ها، ضرب مکرر ماتریس W_{hh} در خودش است. با در نظر گرفتن بزرگ‌ترین مقدار ویژه ماتریس W_{hh}، سه حالت ممکن است:

$\lambda_{max} = 1 \Leftarrow$ گرادیان به خوبی به سمت عقب انتشار می‌یابد

$\lambda_{max} > 1 \Leftarrow$ حاصل‌ضرب به صورت نمایی افزایش می‌یابد [انفجار گرادیان]

$\lambda_{max} < 1 \Leftarrow$ حاصل‌ضرب به صورت نمایی کاهش می‌یابد [محو گرادیان] ← *این حالت متداول‌تر است!*

شبکه‌های برگشتی: انفجار و محو گرادیان

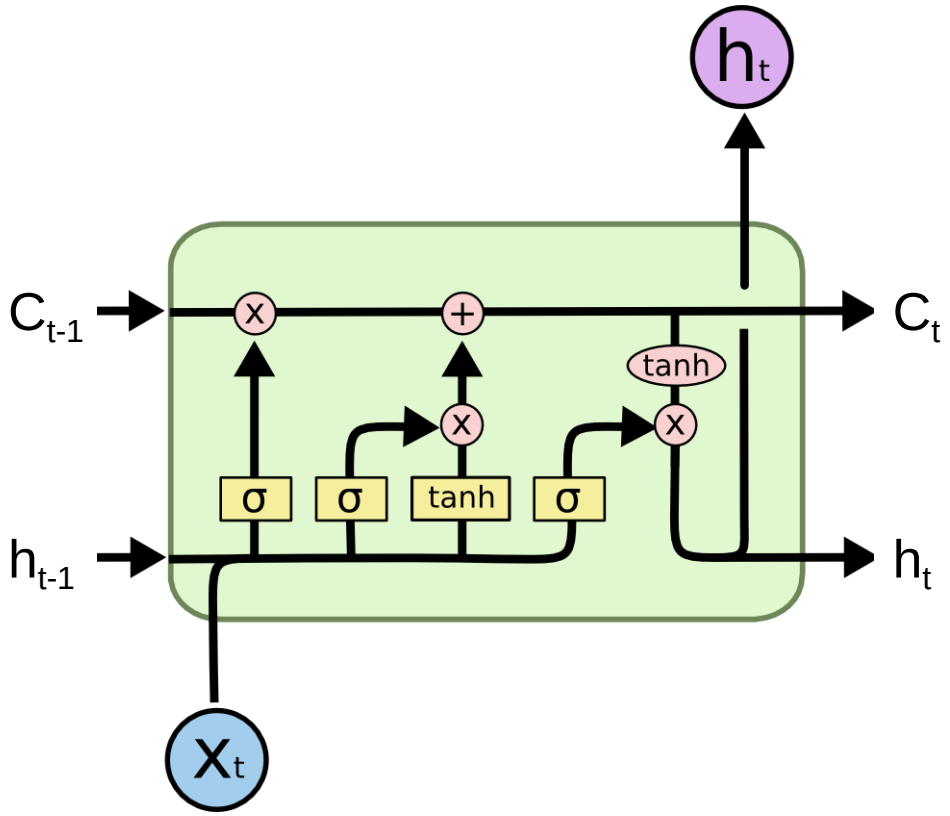
- جلوگیری از انفجار گرادیان.
  - تعیین سقف برای مقدار گرادیان‌ها پس از محاسبه گرادیان‌ها.

```
# backward
optimizer.zero_grad()
loss.backward()
torch.nn.utils.clip_grad_norm(model.parameters(), 0.5)
optimizer.step()
```

شبکه‌های برگشتی: انفجار و محو گرادیان

- روش‌های جلوگیری از محو گرادیان.
  - مقداردهی اولیه ماتریس W_{hh} به صورت هوشمندانه
  - استفاده از روش‌های بهینه‌سازی مرتبه دوم
    - روش‌های نیوتونی و شبه نیوتونی مانند L-BFGS
  - تغییر معماری شبکه برگشتی
    - استفاده از معماری‌هایی مانند LSTM و GRU

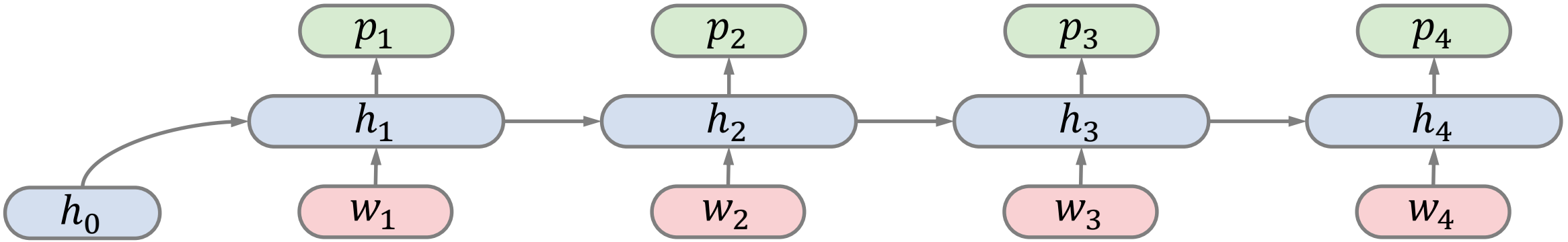
پرطرفدارترین راه‌حل



https://smerity.com/articles/2016/orthogonal_init.html

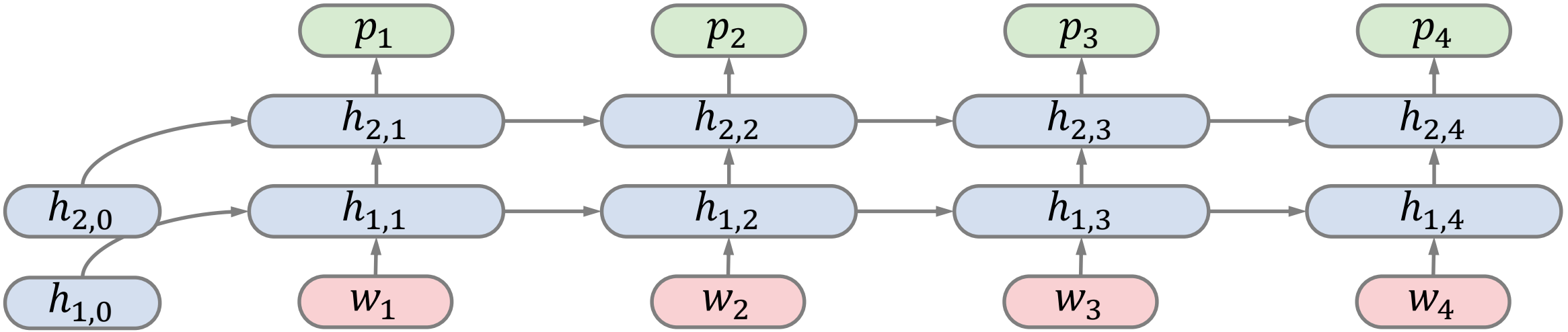
شبکه‌های برگشتی عمیق

- ظرفیت حافظه یک شبکه برگشتی را می‌توان با افزایش اندازه بردار h_n افزایش داد.
- اما اندازه مدل و پیچیدگی محاسباتی یک تابع درجه دوم از اندازه بردار h_n است.
- راه‌حل بهتر. استفاده از شبکه‌های برگشتی عمیق. [تأثیر خطی در میزان محاسبات]



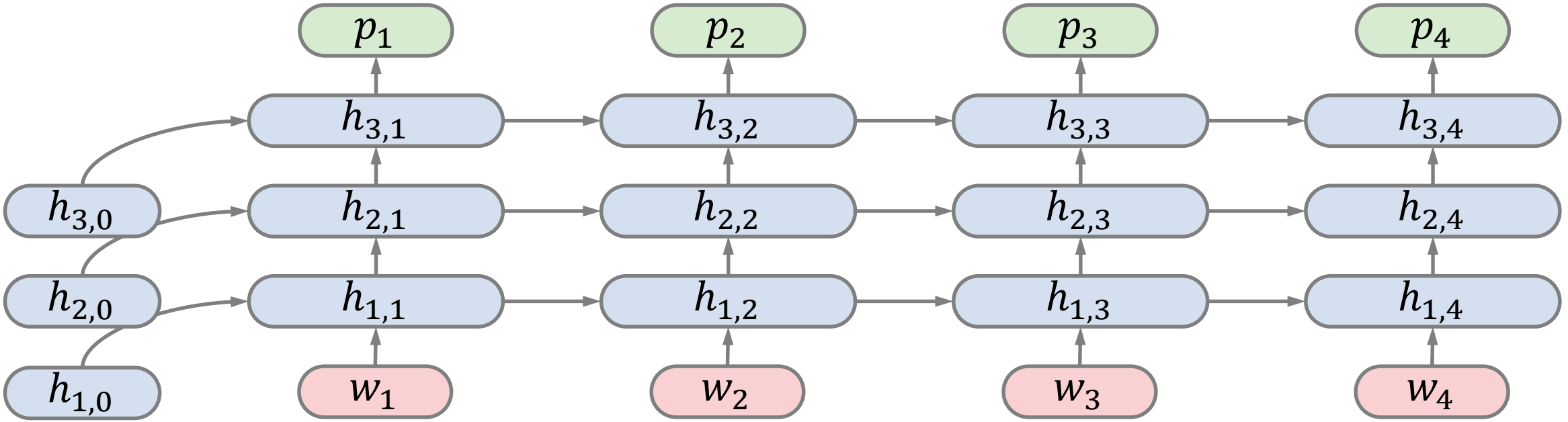
شبکه‌های برگشتی عمیق

- ظرفیت حافظه یک شبکه برگشتی را می‌توان با افزایش اندازه بردار h_n افزایش داد.
- اما اندازه مدل و پیچیدگی محاسباتی یک تابع درجه دوم از اندازه بردار h_n است.
- راه‌حل بهتر. استفاده از شبکه‌های برگشتی عمیق. [تأثیر خطی در میزان محاسبات]



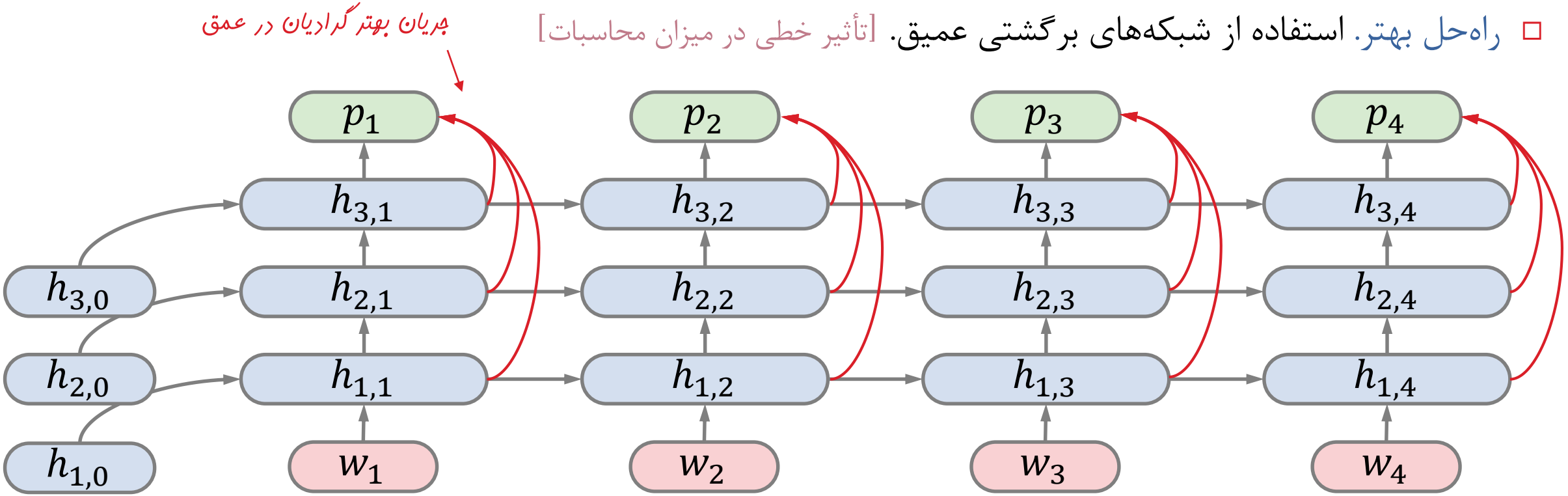
شبکه‌های برگشتی عمیق

- ظرفیت حافظه یک شبکه برگشتی را می‌توان با افزایش اندازه بردار h_n افزایش داد.
- اما اندازه مدل و پیچیدگی محاسباتی یک تابع درجه دوم از اندازه بردار h_n است.
- راه‌حل بهتر. استفاده از شبکه‌های برگشتی عمیق. [تأثیر خطی در میزان محاسبات]



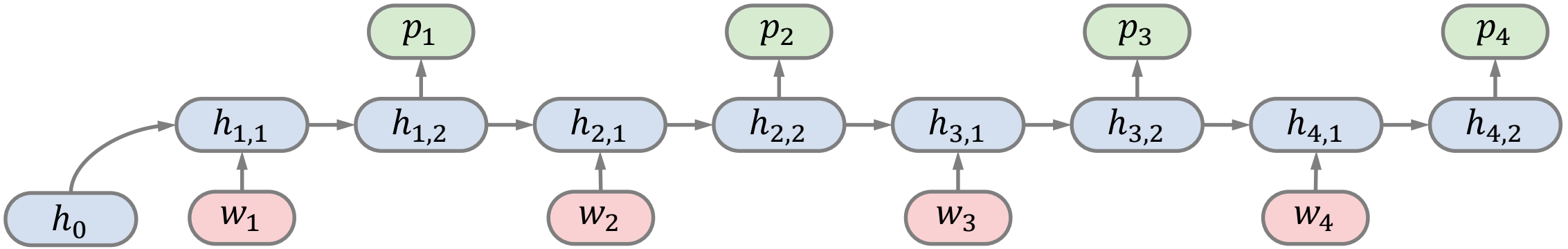
شبکه‌های برگشتی عمیق: اتصالات فرار

- ظرفیت حافظه یک شبکه برگشتی را می‌توان با افزایش اندازه بردار h_n افزایش داد.
- اما اندازه مدل و پیچیدگی محاسباتی یک تابع درجه دوم از اندازه بردار h_n است.
- راه‌حل بهتر. استفاده از شبکه‌های برگشتی عمیق. [تأثیر خطی در میزان محاسبات]



شبکه‌های برگشتی عمیق

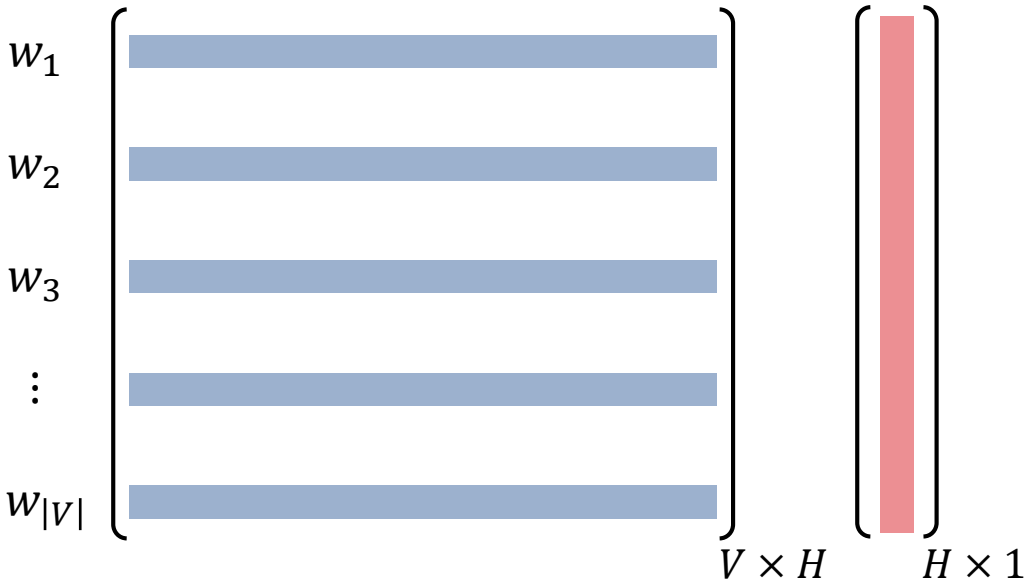
- افزایش عمق در طول زمان.
  - این روش تنها قدرت بازنمایی را افزایش می‌دهد و تأثیری بر افزایش ظرفیت حافظه ندارد.



مدل‌سازی زبانی و اندازه واژگان

- پرهزینه‌ترین لایه در یک شبکه برگشتی در مدل‌سازی زبان:

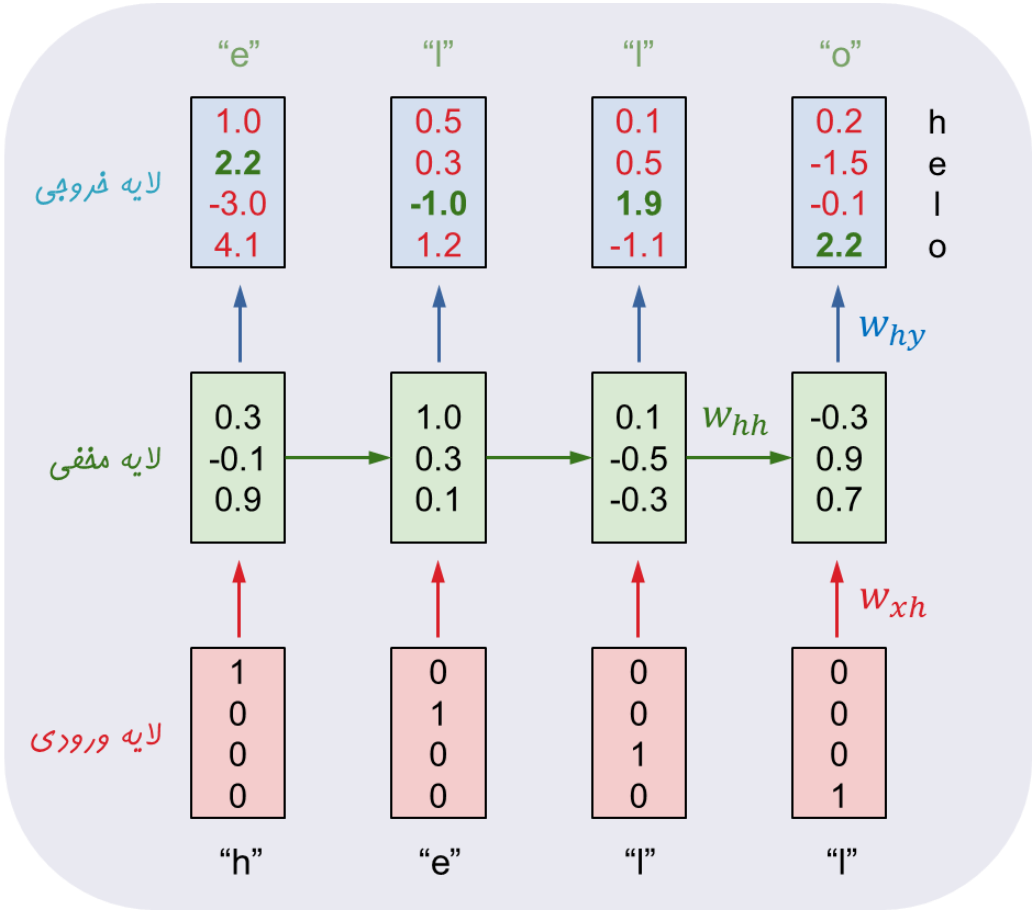
$$p_n = softmax(W_y \ h_n + b_y)$$



- برخی از راه‌حل‌های ممکن:
  - استفاده از V واژه متداول‌تر
  - تغییر تابع هزینه
  - استفاده از روش‌های دسته‌بندی چندسطحی
  - مدل‌سازی زبان در سطح زیرکلمات

مدل‌سازی زبان در سطح زیرکلمه

- مدل‌سازی در سطح کاراکتر.
  - اندازه مجموعه واژگان بسیار کوچک (چند صد کاراکتر)
  - عدم وجود کلمات ناشناخته
  - دنباله‌های طولانی‌تر و در نتیجه وابستگی‌های طولانی‌تر



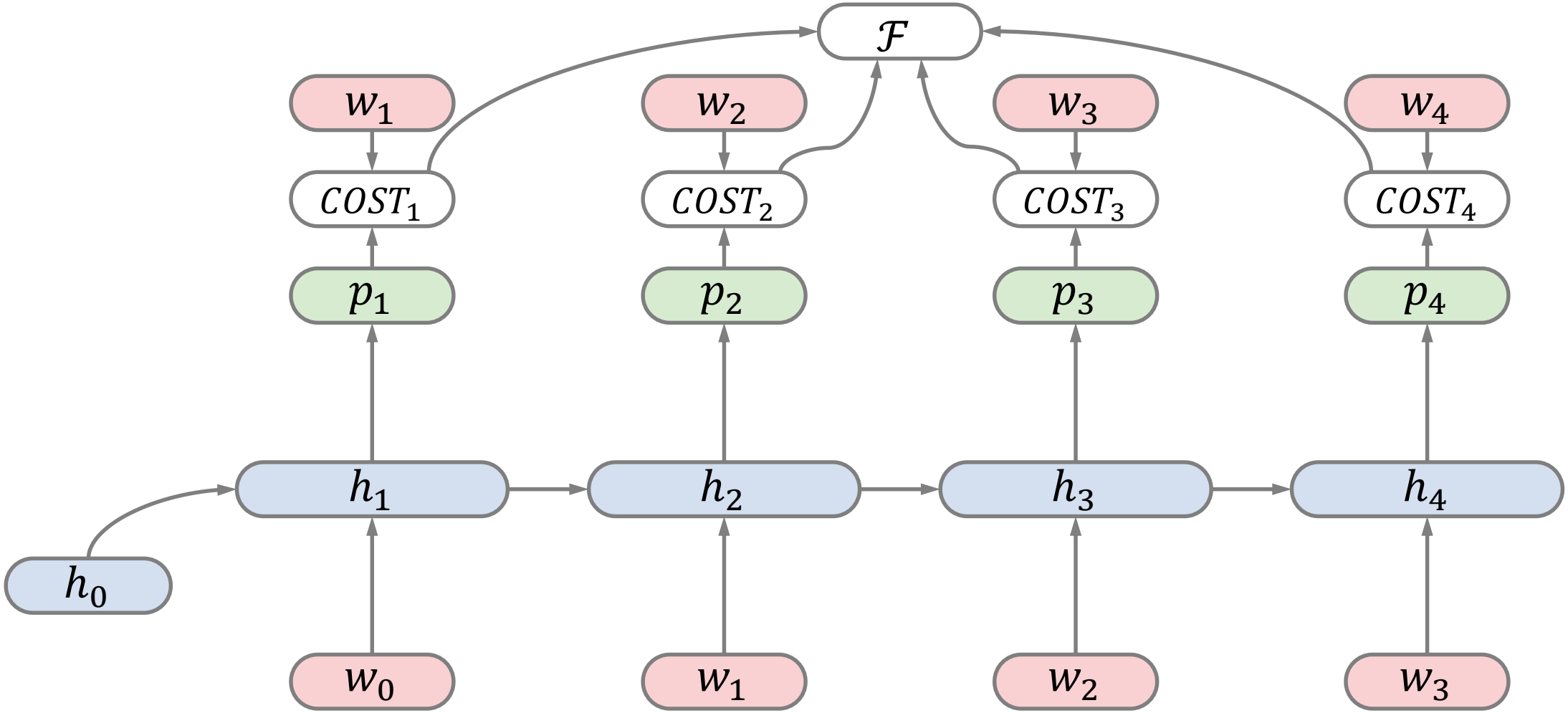
آینده مدل‌سازی زبان!

مدل‌سازی زبان در سطح زیرکلمه

- مدل‌سازی در سطح زیرکلمه.
  - برقراری توازن میان مدل‌سازی در سطح کلمه و مدل‌سازی در سطح کاراکتر

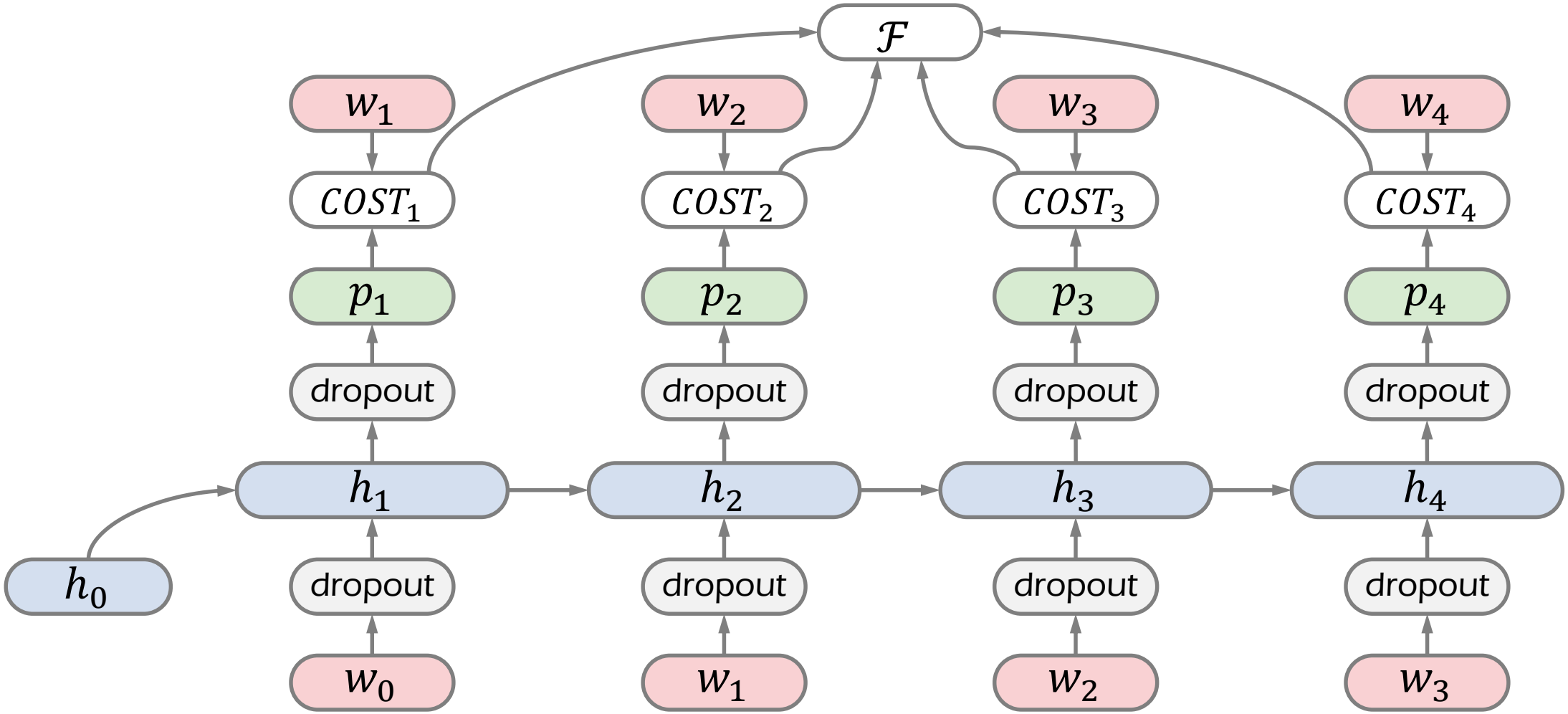
« برای همین است که همه در زمان اس■ پری برنز می ■■ گوی■ ند : خواه■ ش می ■■ کنم مرا نار■ نجی ن■ کن . فن کار این است که باید فردی کار■ دان را بی■ ا■ بید که بتواند مح■ لو■ ل مناسب را برای پوست شما به کار گیرد تا از یک « برنز زننده » که همه از آن می ■■ ترسیم ، اجتناب کنید . » اگر به یک سالن می ■■ رو■ ید ، ح■ ت■ ما ■ سؤال کنید که آیا مح■ لو■ ل ■■ های آنها از در■ جات مختلف برخوردار است یا نه . اگر مطمئن نیستید ، مح■ لو■ لی را برگزی■ نید که ۸ درصد دی ■■ اچ ■■ ا ■ ی دارد - ■- به گفته پر■ ای■ س این مح■ لو■ ل بر روی اکثر پوست ■■ ها با رنگ ■■ های مختلف خوب به نظر می ■■ رسد و شرکتی را پیدا کنید که از اس■ پری ■■ های اچ ■■ وی ■■ ال ■■ پی جدید و سریع ■■ تر استفاده می ■■ کند . بیشتر بر■ نز■ های اس■ پری شده ظرف هفت روز محو می ■■ شوند .

تنظیم: دورریزی [دراپ‌آوت]



Regularizing and Optimizing LSTM Language Models, 2017

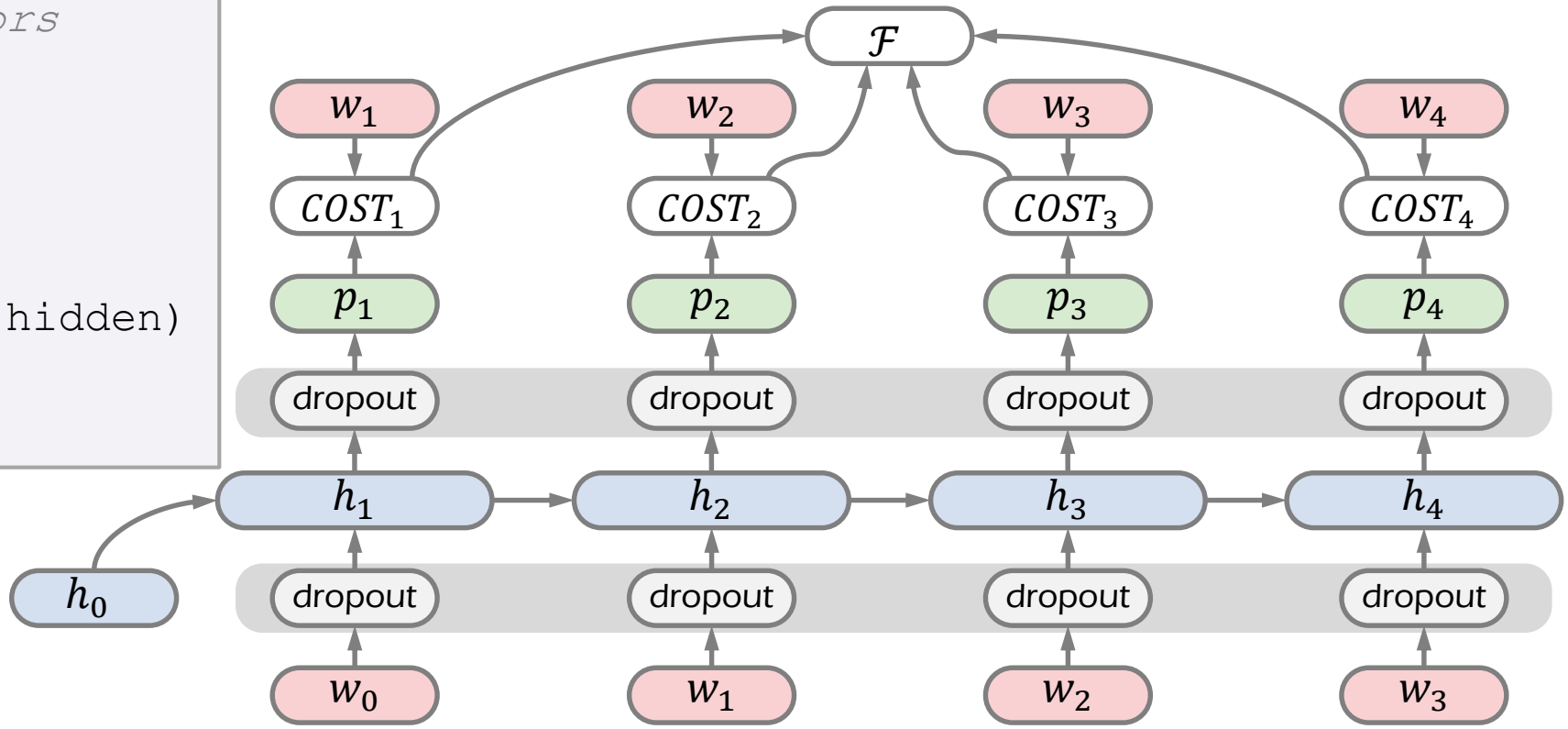
تنظیم: دورریزی [دراپ‌آوت]



Regularizing and Optimizing LSTM Language Models, 2017

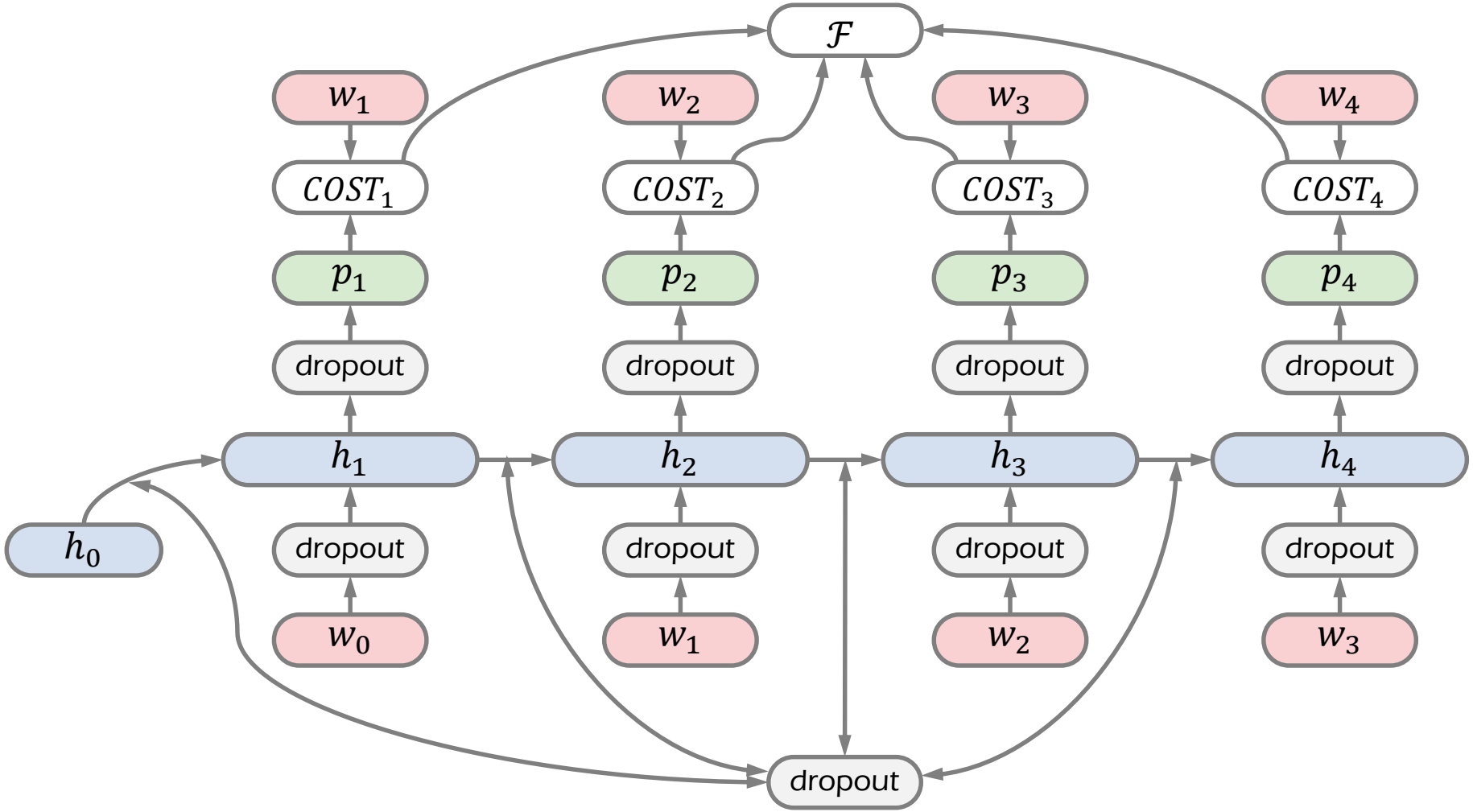
تنظیم: دورریزی [پیاده‌سازی]


```
def forward(self, x, hidden):
  # embed word ids to vectors
  x = self.embedding(x)

  x = self.dropout(x)

  # forward RNN step
  x, hidden = self.lstm(x, hidden)

  x = self.dropout(x)
```

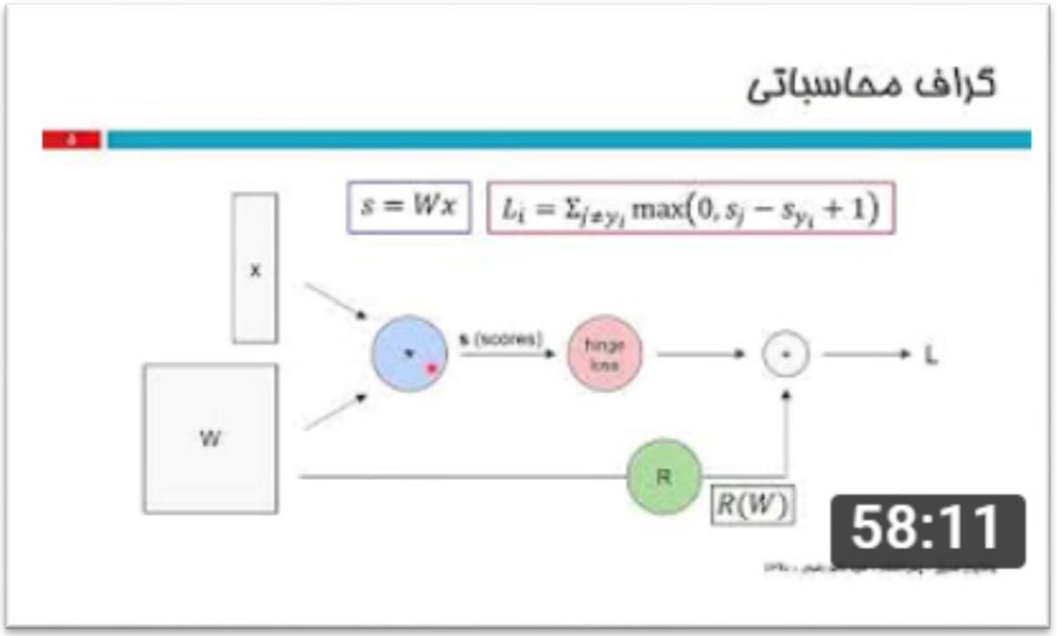


Regularizing and Optimizing LSTM Language Models, 2017

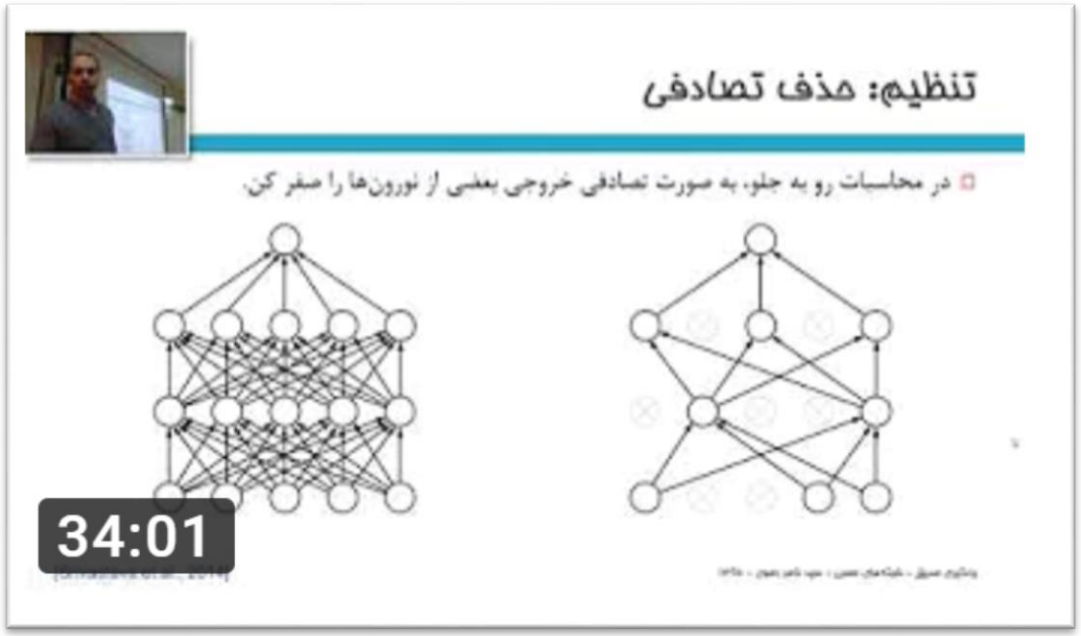
تنظیم: دورریزی بیزی



منابع و مراجع



https://www.youtube.com/watch?v=9JKXFWzf0yc



https://www.youtube.com/watch?v=xZu3wNn4uKY

منابع و مراجع

- Deep Learning Book - Chapter 10
- The Unreasonable Effectiveness of Recurrent Neural Networks
- Natural Language Processing with Deep Learning – Stanford
- Oxford Deep NLP 2017 course