بسم الله الرحمن الرحیم

شبکه‌های عصبی مصنوعی

فصل ۳۶

شبکه‌های عصبی بازگشتی

Recurrent Neural Networks (RNN)

کاظم فولادی قلعه

دانشکده مهندسی، پردیس فارابی

دانشگاه تهران

http://courses.fouladi.ir/nn

شبکه‌های عصبی بازگشتی

۱

مقدمات

دنباله‌ها

SEQUENCES

دنباله: داده‌های بعدی به داده‌های قبلی وابستگی دارند.

هدف: پیش‌بینی اینکه چه چیزی بعداً می‌آید؟

$$\Pr(x) = \prod_i \Pr(x_i | x_1, \dots, x_{i-1})$$

در نظر گرفتن تکه‌های x_i ⇐
تعداد پارامتر کمتر + مدل‌سازی ساده‌تر + امکان تعمیم به طول دلخواه

معمولاً به‌جای طول دلخواه، یک قاب (frame) به‌طول T برداشته می‌شود:

$$\Pr(x) = \prod_i \Pr(x_i | x_{i-T}, \dots, x_{i-1})$$

دنباله‌ها

ویژگی‌ها

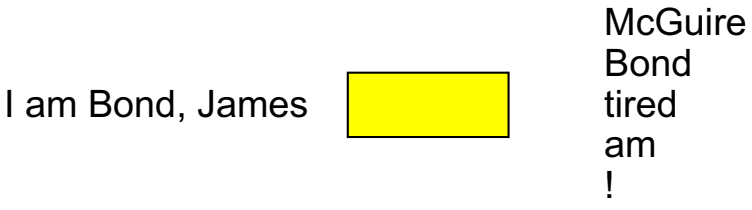
SEQUENCES

- داده‌های داخل یک دنباله، iid نیستند.
  - یعنی مستقل و دارای توزیع یکسان نیستند.
- کلمه‌ی بعدی، وابسته به کلمه‌های قبلی است.
  - به‌صورت ایده‌آل، به همه‌ی کلمه‌های قبلی وابسته است.
- برای تحلیل دنباله نیازمند **مضمون** (context) و نیز **حافظه** (memory) هستیم.

مدل کردن مضمون و حافظه

مثال (۱ از ۲)

MODELLING CONTEXT AND MEMORY



کلمه‌ی بعدی کدام باید باشد؟

مدل کردن مضمون و حافظه

مثال (۲ از ۲)

MODELLING CONTEXT AND MEMORY

I am Bond, James **Bond**

McGuire
Bond
tired
am
!

مدل کردن مضمون و حافظه

بردارهای تک–داغ

One-Hot Vectors

$$x_i \equiv \text{one-hot vector}$$

برداری که همه‌ی عناصر آن صفر است، غیر از یک مقدار 1 برای بعد فعال آن

برای مثال: اگر ۱۲ کلمه در یک دنباله داشته باشیم، ۱۲ بردار تک–داغ خواهیم داشت.

Vocabulary	One-Hot Vectors					
I	I 1	I 0	I 0	I 0	I 0	I 0
am	am 0	am 1	am 0	am 0	am 0	am 0
Bond	Bond 0	Bond 0	Bond 1	Bond 0	Bond 0	Bond 0
James	James 0	James 0	James 0	James 1	James 0	James 0
tired	tired 0	tired 0	tired 0	tired 0	tired 1	tired 0
,	, 0	, 0	, 0	, 0	, 0	, 0
McGuire	McGuire 0	McGuire 0	McGuire 0	McGuire 0	McGuire 0	McGuire 0
!	! 0	! 0	! 0	! 0	! 0	! 0

پس از ایجاد بردارهای تک–داغ، یک روش تعبیه (مثل Word2Vec یا GloVE) اعمال می‌شود.

مدل کردن مضمون و حافظه

حافظه

MEMORY

حافظه، بازنمایی گذشته است.

اطلاعات در گام زمانی t با استفاده از پارامتر θ بر روی یک فضای نهفته‌ی c_t افکنده می‌شود.

از اطلاعات افکنده شده، از لحظه‌ی t در لحظه‌ی $t+1$ استفاده‌ی مجدد می‌شود:

$$c_{t+1} = h(x_{t+1}, c_t; \theta)$$

پارامتر بازگشتی θ برای تمام گام‌های زمانی به‌اشتراک گذاشته می‌شود:

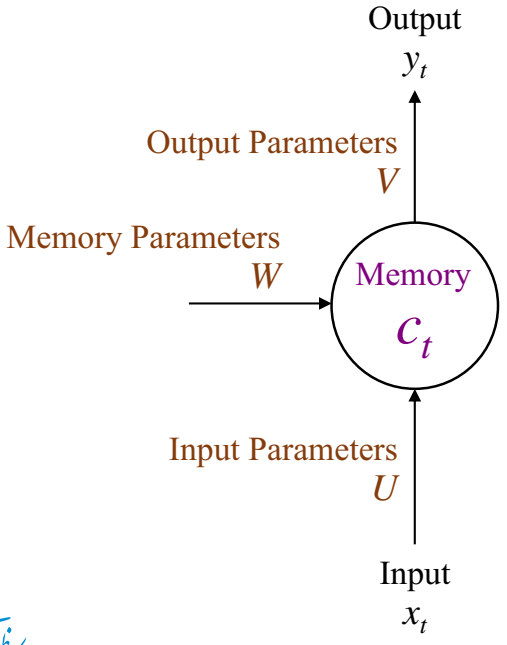
$$c_{t+1} = h(x_{t+1}, h(x_t, h(x_{t-1}, \dots h(x_1, c_0; \theta); \theta); \theta); \theta)$$

مدل کردن مضمون و حافظه

مدل کردن حافظه در قالب یک گراف

MEMORY AS A GRAPH

ساده‌ترین مدل

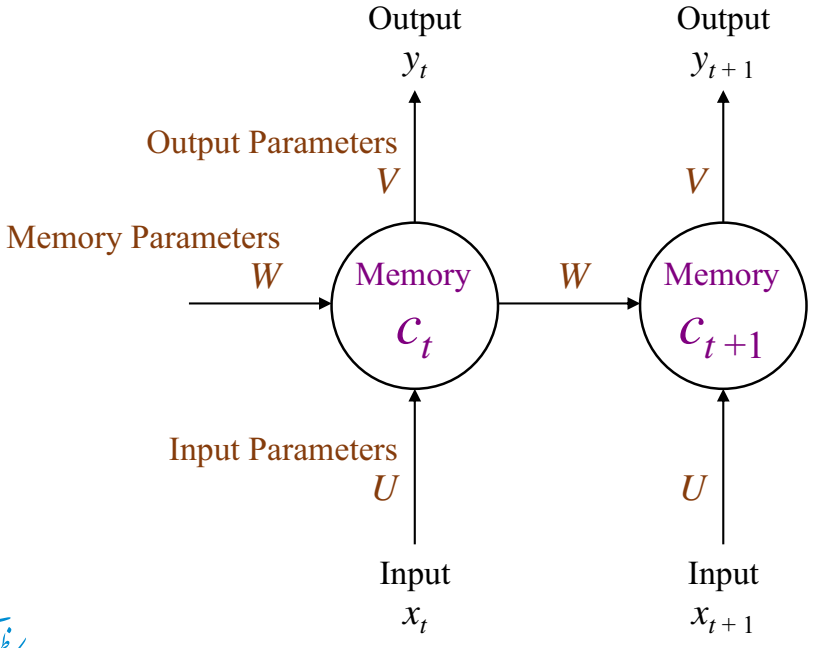


مدل کردن مضمون و حافظه

مدل کردن حافظه در قالب یک گراف

MEMORY AS A GRAPH

سادەترین مدل

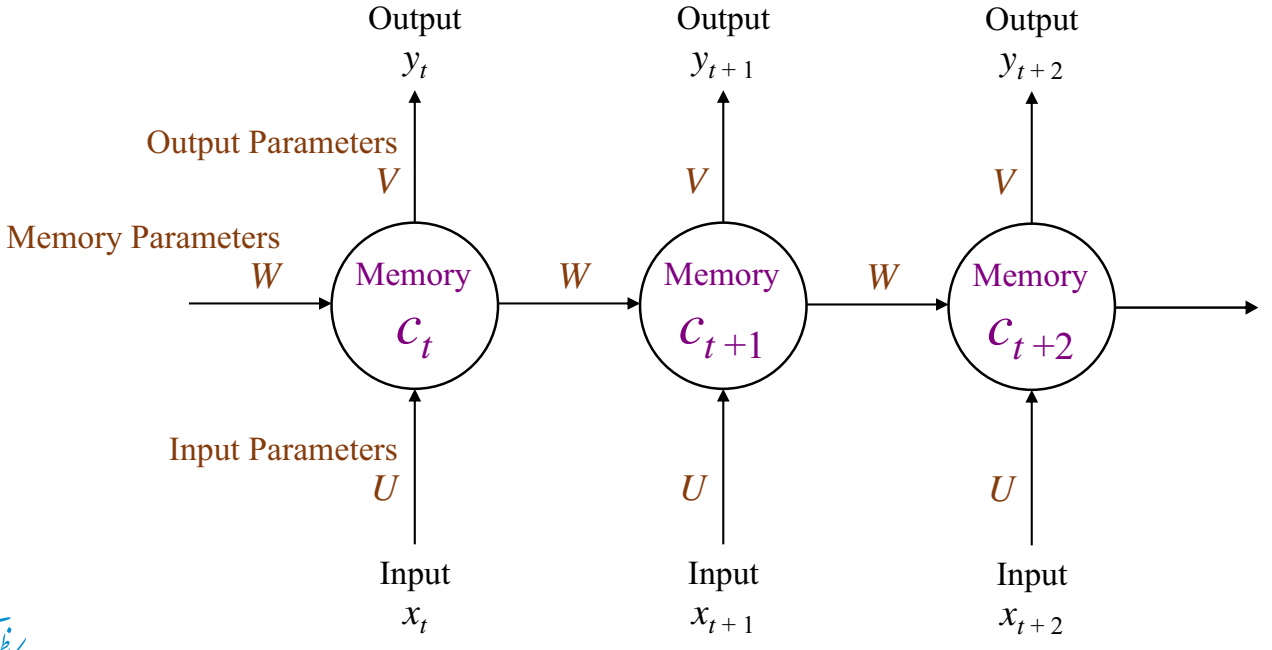


مدل کردن مضمون و حافظه

مدل کردن حافظه در قالب یک گراف

MEMORY AS A GRAPH

ساده‌ترین مدل



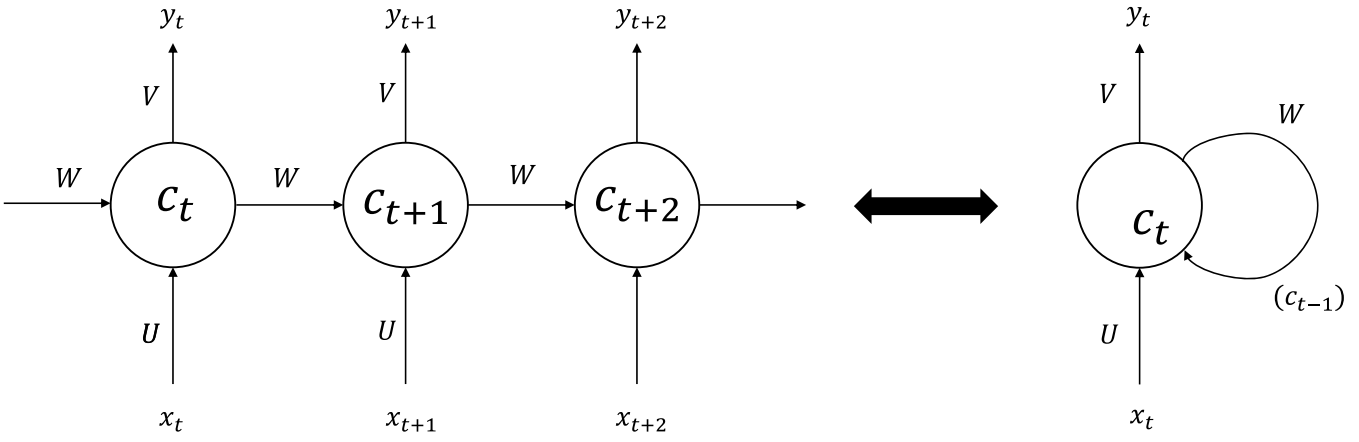
مدل کردن مضمون و حافظه

تا کردن حافظه

FOLDING THE MEMORY

شبکه‌ی باز شده / تا نشده
Unrolled / Unfolded Network

شبکه‌ی تا شده
Folded Network

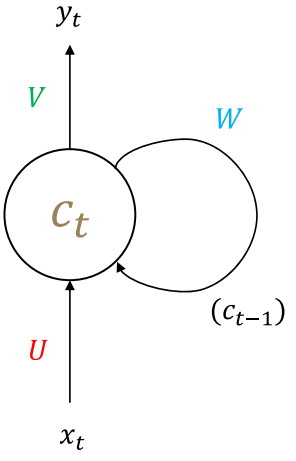


شبکه‌ی عصبی بازگشتی

RECURRENT NEURAL NETWORK (RNN)

شبکه‌ی عصبی بازگشتی، با استفاده از دو معادله تعریف می‌شود:

$$c_t = \tanh(U\, x_t + W\, c_{t-1})$$
$$y_t = \text{softmax}(V\, c_t)$$



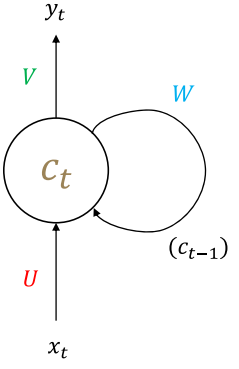
شبکه‌ی عصبی بازگشتی

مثال

RECURRENT NEURAL NETWORK (RNN)

- Vocabulary of 5 words
- A memory of 3 units [Hyperparameter that we choose like layer size]
  - c_t: $[3 \times 1]$, W: $[3 \times 3]$
- An input projection of 3 dimensions
  - U: $[3 \times 5]$
- An output projections of 10 dimensions
  - V: $[10 \times 3]$

$$c_t = \tanh(U\, x_t + W\, c_{t-1})$$
$$y_t = \text{softmax}(V\, c_t)$$



$$U \cdot x_{t=4} = \begin{bmatrix} 0.1 & -0.3 & 1.2 & 0.6 & -0.8 \\ -0.2 & 0.4 & 0.5 & 0.9 & -0.1 \\ -0.1 & 0.2 & -0.7 & -0.8 & 0.3 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.6 \\ 0.9 \\ -0.8 \end{bmatrix} = U^{(4)}$$

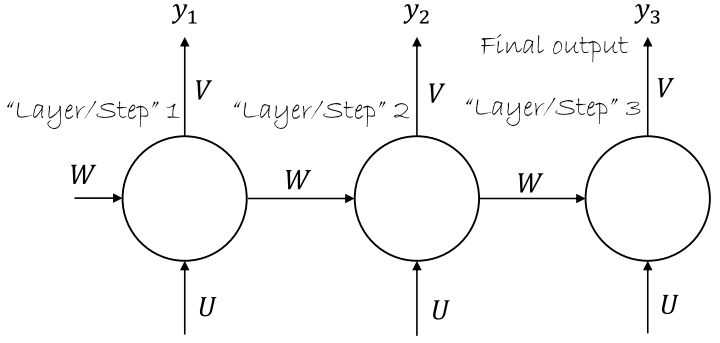
شبکه‌ی عصبی بازگشتی

مقایسه‌ی شبکه‌ی بسته شده با شبکه‌ی چندلایه

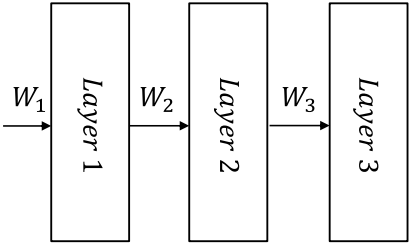
ROLLED NETWORK VS. MULTI-LAYER NETWORK

تفاوت‌ها:

- در شبکه‌ی بسته شده، گام‌ها به‌جای لایه‌ها نقش بازی می‌کنند.
- در شبکه‌ی بسته شده، پارامترهای گام مشترک است؛ در حالی که در شبکه‌ی چندلایه، متفاوت است.



3-gram unrolled recurrent network



3-layer neural network

شبکه‌ی عصبی بازگشتی

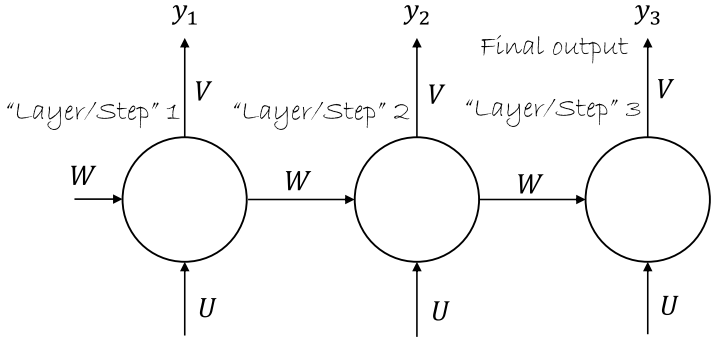
مقایسه‌ی شبکه‌ی بسته شده با شبکه‌ی چندلایه

ROLLED NETWORK VS. MULTI-LAYER NETWORK

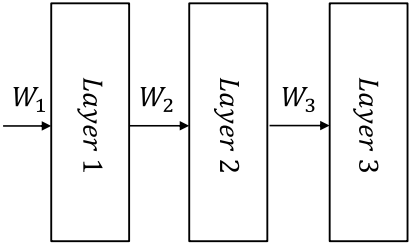
تفاوت‌ها:

- در شبکه‌ی بسته شده، گام‌ها به‌جای لایه‌ها نقش بازی می‌کنند.
- در شبکه‌ی بسته شده، پارامترهای گام مشترک است؛ در حالی که در شبکه‌ی چندلایه، متفاوت است.

گاهی خروجی‌های میانی مورد نیاز نیستند،
با حذف آنها تقریباً به یک مدل استاندارد شبکه‌ی عصبی می‌رسیم:



3-gram unrolled recurrent network



3-layer neural network

آموزش شبکه‌های عصبی بازگشتی

TRAINING RECURRENT NEURAL NETWORKS

از تابع اتلاف آنتروپی متقابل استفاده می‌کنیم:
Cross-Entropy Loss

$$P = \prod_{t,k} y_{tk}^{l_{tk}} \quad \Rightarrow \quad \mathcal{L} = -\log P = \sum_t \mathcal{L}_t = -\frac{1}{T}\sum_t l_t \log y_t$$

الگوریتم مورد استفاده: پس‌انتشار در طول زمان (Backpropagation Through Time: BPTT)

- مجدداً از قاعده‌ی زنجیره‌ای استفاده می‌شود.
- تنها تفاوت: گرادیان ها بر روی گام‌های زمانی باقی می‌مانند.

آموزش شبکه‌های عصبی بازگشتی

پس‌انتشار در طول زمان: مثال (۱ از ۴)

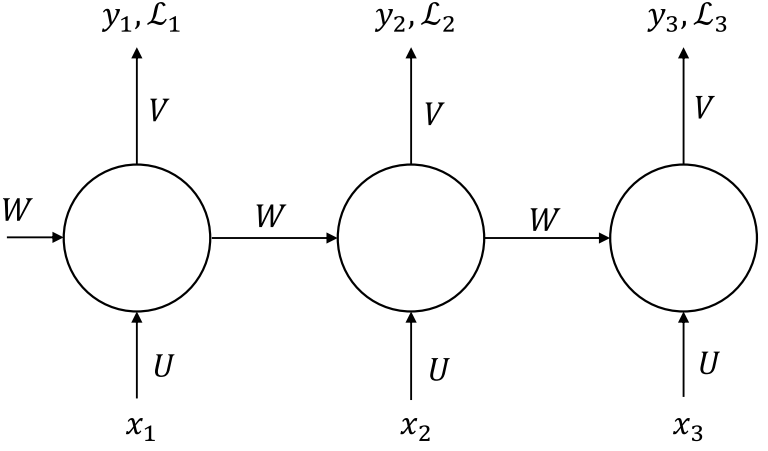
BACKPROPAGATION THROUGH TIME: BPTT

$$\frac{\partial \mathcal{L}}{\partial V}, \frac{\partial \mathcal{L}}{\partial W}, \frac{\partial \mathcal{L}}{\partial U}$$

برای ساده‌تر کردن، بر روی گام ۳ تمرکز می‌کنیم:

$$\frac{\partial \mathcal{L}_3}{\partial V}, \frac{\partial \mathcal{L}_3}{\partial W}, \frac{\partial \mathcal{L}_3}{\partial U}$$

$$c_t = \tanh(U\, x_t + W c_{t-1})$$
$$y_t = \text{softmax}(V c_t)$$
$$\mathcal{L} = -\sum_t l_t \log y_t = \sum_t \mathcal{L}_t$$



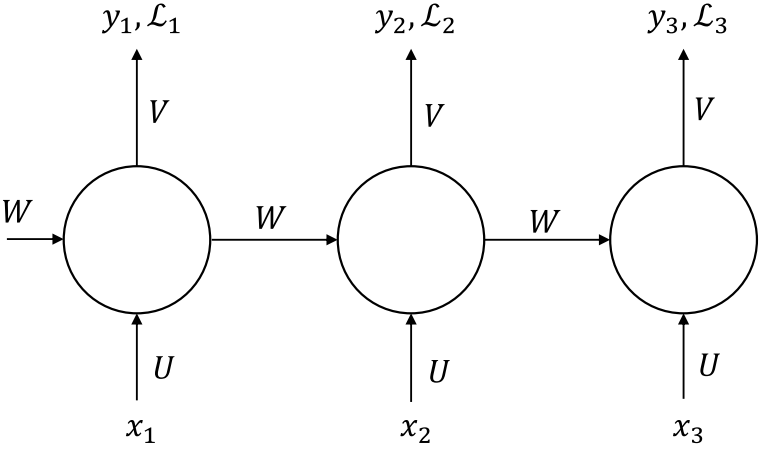
آموزش شبکه‌های عصبی بازگشتی

پس‌انتشار در طول زمان: مثال (۲ از ۴)

BACKPROPAGATION THROUGH TIME: BPTT

$$\frac{\partial \mathcal{L}_3}{\partial V} = \frac{\partial \mathcal{L}_3}{\partial y_3}\frac{\partial y_3}{\partial V} = (y_3 - l_3)\cdot c_3$$

$$c_t = \tanh(U\,x_t + W c_{t-1})$$
$$y_t = \text{softmax}(V c_t)$$
$$\mathcal{L} = -\sum_t l_t \log y_t = \sum_t \mathcal{L}_t$$



آموزش شبکه‌های عصبی بازگشتی

پس‌انتشار در طول زمان: مثال (۳ از ۴)

BACKPROPAGATION THROUGH TIME: BPTT

$$\frac{\partial \mathcal{L}_3}{\partial W} = \frac{\partial \mathcal{L}_3}{\partial y_3}\frac{\partial y_3}{\partial c_3}\frac{\partial c_3}{\partial W}$$

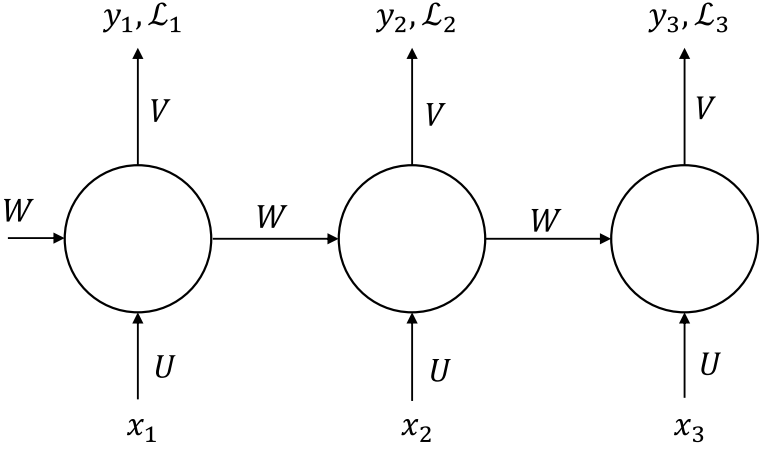
باید رابطه‌ی بین c_3 و W را به‌دست آوریم:

Two-fold: $c_t = \tanh(U\, x_t + W c_{t-1})$

$$\frac{\partial\, f(\varphi(x), \psi(x))}{\partial x} = \frac{\partial f}{\partial \varphi}\frac{\partial \varphi}{\partial x} + \frac{\partial f}{\partial \psi}\frac{\partial \psi}{\partial x}$$

$$\frac{\partial c_3}{\partial W} \propto c_2 + \frac{\partial c_2}{\partial W} \quad (\frac{\partial W}{\partial W} = 1)$$

$$c_t = \tanh(U\, x_t + W c_{t-1})$$
$$y_t = \text{softmax}(V c_t)$$
$$\mathcal{L} = -\sum_t l_t \log y_t = \sum_t \mathcal{L}_t$$



آموزش شبکه‌های عصبی بازگشتی

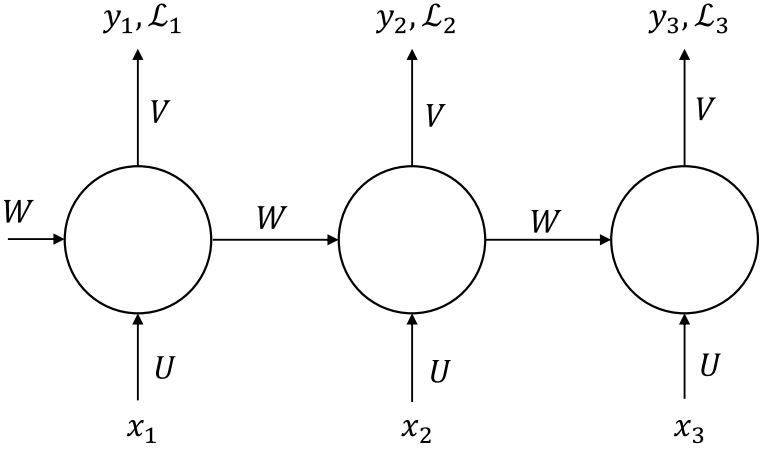
پس‌انتشار در طول زمان: مثال (۴ از ۴)

BACKPROPAGATION THROUGH TIME: BPTT

به صورت بازگشتی:

$$\left.\begin{aligned}\frac{\partial c_3}{\partial W} &= c_2 + \frac{\partial c_2}{\partial W}\\ \frac{\partial c_2}{\partial W} &= c_1 + \frac{\partial c_1}{\partial W}\\ \frac{\partial c_1}{\partial W} &= c_0 + \frac{\partial c_0}{\partial W}\end{aligned}\right\} \quad \frac{\partial c_3}{\partial W} = \sum_{t=1}^{3} \frac{\partial c_3}{\partial c_t}\frac{\partial c_t}{\partial W} \quad \Rightarrow \quad \boxed{\frac{\partial \mathcal{L}_3}{\partial W} = \sum_{t=1}^{3} \frac{\partial \mathcal{L}_3}{\partial y_3}\frac{\partial y_3}{\partial c_3}\frac{\partial c_3}{\partial c_t}\frac{\partial c_t}{\partial W}}$$

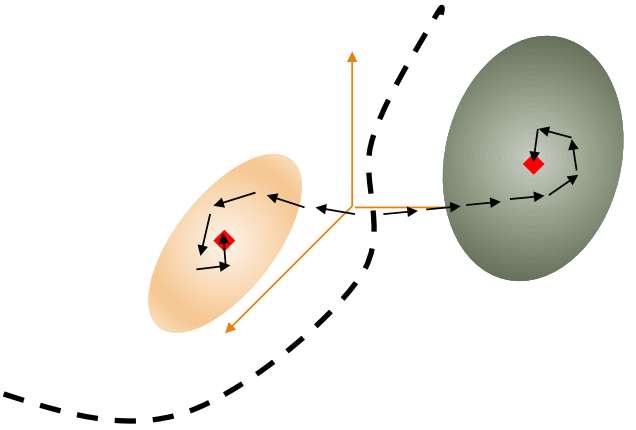
$$\begin{aligned} c_t &= \tanh(U\,x_t + W c_{t-1})\\ y_t &= \text{softmax}(V c_t)\\ \mathcal{L} &= -\sum_t l_t \log y_t = \sum_t \mathcal{L}_t \end{aligned}$$



آموزش شبکه‌های عصبی بازگشتی

دشواری‌های آموزش

- فضای حافظه‌ی نهفته، از چندین بعد تشکیل شده است.
- یک زیرفضا از فضای حالت حافظه، می‌تواند اطلاعات را ذخیره کند، اگر چند بستر جذب در برخی ابعاد موجود باشد.
- گرادیان‌ها باید در نزدیک بستر جذب قوی باشند.



آموزش شبکه‌های عصبی بازگشتی

دشواری‌های آموزش

آموزش شبکه‌ی عصبی بازگشتی دشوار است، به‌دلیل:

- **اضمحلال گرادیان‌ها (Vanishing Gradients)**
  پس از چند گام زمانی، گرادیان‌ها تقریباً صفر می‌شوند.
- **انفجار گرادیان‌ها (Exploding Gradients)**
  پس از چند گام زمانی، گرادیان‌ها بسیار بزرگ می‌شود.
- **عدم تسخیر وابستگی‌های طولانی–مدت**

فرمول‌بندی جایگزین برای شبکه‌های عصبی بازگشتی

ALTERNATIVES FORMULATION FOR RNNS

یک فرمول‌بندی جایگزین:

$$c_t = W \cdot \tanh(c_{t-1}) + U \cdot x_t + b$$

$$\mathcal{L} = \sum_t \mathcal{L}_t(c_t)$$

فرمول‌بندی جایگزین برای شبکه‌های عصبی بازگشتی

نگاه دیگری به گرادیان‌ها

ANOTHER LOOK AT THE GRADIENTS

$$\mathcal{L} = L(c_T(c_{T-1}(\dots(c_1(x_1, c_0; W); W); W); W); W)$$

$$\frac{\partial \mathcal{L}_t}{\partial W} = \sum_{\tau=1}^{t} \frac{\partial \mathcal{L}_t}{\partial c_t} \frac{\partial c_t}{\partial c_\tau} \frac{\partial c_\tau}{\partial W}$$

$$\frac{\partial \mathcal{L}}{\partial c_t} \frac{\partial c_t}{\partial c_\tau} = \frac{\partial \mathcal{L}}{\partial c_t} \cdot \frac{\partial c_t}{\partial c_{t-1}} \cdot \frac{\partial c_{t-1}}{\partial c_{t-2}} \cdot \dots \cdot \frac{\partial c_{\tau+1}}{\partial c_\tau} \leq \eta^{t-\tau} \frac{\partial \mathcal{L}_t}{\partial c_t}$$

$Rest$ → short-term factors $t \gg \tau$ → long-term factors

η determines the norm of the gradients

گرادیان‌های RNN، یک حاصل‌ضرب بازگشتی از $\partial c_t / \partial c_{t-1}$ است.

فرمول‌بندی جایگزین برای شبکه‌های عصبی بازگشتی

گرادیان‌های RNN در یک بعد

RNN GRADIENTS IN 1D

- $$\frac{\partial \mathcal{L}}{\partial c_t} = \frac{\partial \mathcal{L}}{\partial c_T} \cdot \frac{\partial c_T}{\partial c_{T-1}} \cdot \frac{\partial c_{T-1}}{\partial c_{T-2}} \cdot \dots \cdot \frac{\partial c_{t+1}}{\partial c_{c_t}}$$ (<1 <1 <1) $\Rightarrow \frac{\partial \mathcal{L}}{\partial W} \ll 1 \Longrightarrow$ Vanishing gradient
- $$\frac{\partial \mathcal{L}}{\partial c_t} = \frac{\partial \mathcal{L}}{\partial c_T} \cdot \frac{\partial c_T}{\partial c_{T-1}} \cdot \frac{\partial c_{T-1}}{\partial c_{T-2}} \cdot \dots \cdot \frac{\partial c_1}{\partial c_{c_t}}$$ (>1 >1 >1) $\Rightarrow \frac{\partial \mathcal{L}}{\partial W} \gg 1 \Longrightarrow$ Exploding gradient

فرمول‌بندی جایگزین برای شبکه‌های عصبی بازگشتی

گرادیان‌های RNN در چند بعد

RNN GRADIENTS IN N-D

When $c_T \in \mathbb{R}^N$ then $\frac{\partial c_t}{\partial c_{t-1}}$ is a Jacobian

- $$\frac{\partial \mathcal{L}}{\partial c_t} = \frac{\partial \mathcal{L}}{\partial c_T} \cdot \frac{\partial c_T}{\partial c_{T-1}} \cdot \frac{\partial c_{T-1}}{\partial c_{T-2}} \cdot \ldots \cdot \frac{\partial c_{t+1}}{\partial c_t}$$ (<1 <1 <1) $\Rightarrow \frac{\partial \mathcal{L}}{\partial \theta} \ll 1 \Rightarrow$ Vanishing gradient
- $$\frac{\partial \mathcal{L}}{\partial c_t} = \frac{\partial \mathcal{L}}{\partial c_T} \cdot \frac{\partial c_T}{\partial c_{T-1}} \cdot \frac{\partial c_{T-1}}{\partial c_{T-2}} \cdot \ldots \cdot \frac{\partial c_{t+1}}{\partial c_t}$$ (>1 >1 >1) $\Rightarrow \frac{\partial \mathcal{L}}{\partial \theta} \gg 1 \Rightarrow$ Exploding gradient

$$y \in \mathbb{R}^2, x \in \mathbb{R}^3: \quad \frac{dy}{dx} = \begin{bmatrix} \frac{\partial y^{(1)}}{\partial x^{(1)}} & \frac{\partial y^{(1)}}{\partial x^{(2)}} & \frac{\partial y^{(1)}}{\partial x^{(3)}} \\ \frac{\partial y^{(2)}}{\partial x^{(1)}} & \frac{\partial y^{(2)}}{\partial x^{(2)}} & \frac{\partial y^{(2)}}{\partial x^{(3)}} \end{bmatrix}$$

فرمول‌بندی جایگزین برای شبکه‌های عصبی بازگشتی

گرادیان‌های RNN در چند بعد

RNN GRADIENTS IN N-D

When $c_T \in \mathbb{R}^N$ then $\frac{\partial c_t}{\partial c_{t-1}}$ is a Jacobian

شعاع طیفی ژاکوبی (= بزرگ‌ترین مقدار ویژه = ρ) پارامتر مهمی است:

- $$\frac{\partial \mathcal{L}}{\partial c_t} = \frac{\partial \mathcal{L}}{\partial c_T} \cdot \underbrace{\frac{\partial c_T}{\partial c_{T-1}}}_{\rho<1} \cdot \underbrace{\frac{\partial c_{T-1}}{\partial c_{T-2}}}_{\rho<1} \cdot \dots \cdot \underbrace{\frac{\partial c_{t+1}}{\partial c_t}}_{\rho<1} \Rightarrow \frac{\partial \mathcal{L}}{\partial c_t} \ll 1 \Rightarrow \text{Vanishing gradient}$$
- $$\frac{\partial \mathcal{L}}{\partial c_t} = \frac{\partial \mathcal{L}}{\partial c_T} \cdot \underbrace{\frac{\partial c_T}{\partial c_{T-1}}}_{\rho>1} \cdot \underbrace{\frac{\partial c_{T-1}}{\partial c_{T-2}}}_{\rho>1} \cdot \dots \cdot \underbrace{\frac{\partial c_{t+1}}{\partial c_t}}_{\rho>1} \Rightarrow \frac{\partial \mathcal{L}}{\partial c_t} \gg 1 \Rightarrow \text{Exploding gradient}$$

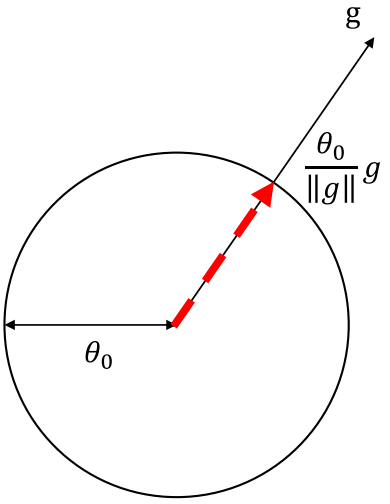
برش گرادیان برای جلوگیری از انفجار گرادیان

GRADIENT CLIPPING

مقیاس‌بندی گرادیان با یک مقدار آستانه:

Pseudocode

1. $g \leftarrow \frac{\partial \mathcal{L}}{\partial W}$
2. if $\|g\| > \theta_0$:
   $g \leftarrow \frac{\theta_0}{\|g\|} g$
   else:
   print('Do nothing')



این الگوریتم ساده است، اما به‌خوبی کار می‌کند!

اضمحلال گرادیان‌ها

VANISHING GRADIENTS

گرادیان خطا نسبت به سلول میانی:

$$\frac{\partial \mathcal{L}_t}{\partial W} = \sum_{\tau=1}^{t} \frac{\partial \mathcal{L}_r}{\partial y_t} \frac{\partial y_t}{\partial c_t} \frac{\partial c_t}{\partial c_\tau} \frac{\partial c_\tau}{\partial W}$$

$$\frac{\partial c_t}{\partial c_\tau} = \prod_{t \geq k \geq \tau} \frac{\partial c_k}{\partial c_{k-1}} = \prod_{t \geq k \geq \tau} W \cdot \partial \tanh(c_{k-1})$$

اضمحلال گرادیان‌ها

VANISHING GRADIENTS

گرادیان خطا نسبت به سلول میانی:

$$\frac{\partial \mathcal{L}_t}{\partial W} = \sum_{\tau=1}^{t} \frac{\partial \mathcal{L}_r}{\partial y_t} \frac{\partial y_t}{\partial c_t} \frac{\partial c_t}{\partial c_\tau} \frac{\partial c_\tau}{\partial W}$$

$$\frac{\partial c_t}{\partial c_\tau} = \prod_{t \geq k \geq \tau} \frac{\partial c_k}{\partial c_{k-1}} = \prod_{t \geq k \geq \tau} W \cdot \partial \tanh(c_{k-1})$$

- For $t = 1, r = 2 \Rightarrow \frac{\partial \mathcal{L}_2}{\partial W} \propto \frac{\partial c_2}{\partial c_1}$
- For $t = 1, r = 3 \Rightarrow \frac{\partial \mathcal{L}_3}{\partial W} \propto \frac{\partial c_3}{\partial c_1} = \frac{\partial c_3}{\partial c_2} \cdot \frac{\partial c_2}{\partial c_1}$
- For $t = 1, r = 4 \Rightarrow \frac{\partial \mathcal{L}_4}{\partial W} \propto \frac{\partial c_4}{\partial c_1} = \frac{\partial c_4}{\partial c_3} \cdot \frac{\partial c_3}{\partial c_2} \cdot \frac{\partial c_2}{\partial c_1}$

اضمحلال گرادیان‌ها

VANISHING GRADIENTS

گرادیان خطا نسبت به سلول میانی:

$$\frac{\partial \mathcal{L}_t}{\partial W} = \sum_{\tau=1}^{t} \frac{\partial \mathcal{L}_r}{\partial y_t} \frac{\partial y_t}{\partial c_t} \frac{\partial c_t}{\partial c_\tau} \frac{\partial c_\tau}{\partial W}$$

$$\frac{\partial c_t}{\partial c_\tau} = \prod_{t \geq k \geq \tau} \frac{\partial c_k}{\partial c_{k-1}} = \prod_{t \geq k \geq \tau} W \cdot \partial \tanh(c_{k-1})$$

وابستگی‌های طولانی–مدت موجب می‌شوند وزن‌ها به‌صورت نمایی کوچک و کوچک‌تر شوند.

اضمحلال گرادیان‌ها

VANISHING GRADIENTS

بازمقیاس‌دهی گرادیان‌های مضمحل‌شده راه حل خوبی نیست!

وزن‌ها بین گام‌های زمانی مشترک هستند ⇐ اتلاف‌ها بر روی گام‌های زمانی جمع می‌شوند:

$$\mathcal{L} = \sum_t \mathcal{L}_t \quad \Rightarrow \quad \frac{\partial \mathcal{L}}{\partial W} = \sum_t \frac{\partial \mathcal{L}_t}{\partial W}$$

$$\frac{\partial \mathcal{L}_t}{\partial W} = \sum_{\tau=1}^{t} \frac{\partial \mathcal{L}_t}{\partial c_\tau} \frac{\partial c_\tau}{\partial W} = \sum_{\tau=1}^{t} \frac{\partial \mathcal{L}_t}{\partial c_t} \frac{\partial c_t}{\partial c_\tau} \frac{\partial c_\tau}{\partial W}$$

بازمقیاس‌دهی برای یک گام زمانی، بر همه‌ی گام‌های زمانی تأثیر می‌گذارد

⇓

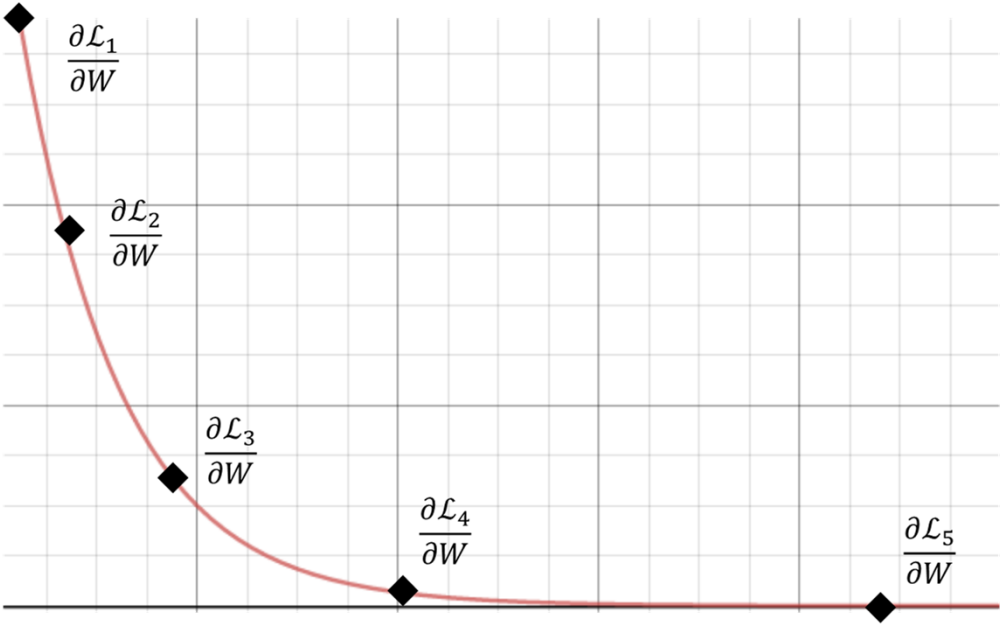
ضریب بازمقیاس‌دهی برای یک گام زمانی، برای گام زمانی دیگر کار نمی‌کند!

اضمحلال گرادیان‌ها

مثال (۱ از ۲)

VANISHING GRADIENTS

$$\frac{\partial \mathcal{L}}{\partial W} = \frac{\partial \mathcal{L}_1}{\partial W} + \frac{\partial \mathcal{L}_2}{\partial W} + \frac{\partial \mathcal{L}_3}{\partial W} + \frac{\partial \mathcal{L}_4}{\partial W} + \frac{\partial \mathcal{L}_5}{\partial W}$$

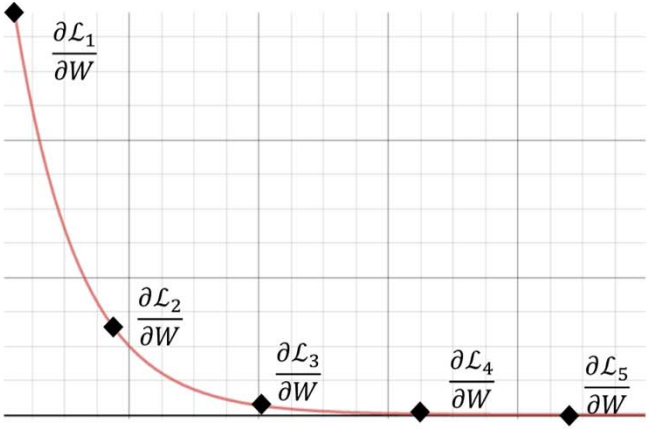


اضمحلال گرادیان‌ها

مثال (۲ از ۲)

VANISHING GRADIENTS

- Let's say $\frac{\partial \mathcal{L}_1}{\partial w} \propto 1, \frac{\partial \mathcal{L}_2}{\partial w} \propto 1/10, \frac{\partial \mathcal{L}_3}{\partial w} \propto 1/100, \frac{\partial \mathcal{L}_4}{\partial w} \propto 1/1000, \frac{\partial \mathcal{L}_5}{\partial w} \propto 1/10000$
- $\frac{\partial \mathcal{L}}{\partial w} = \sum_r \frac{\partial \mathcal{L}_r}{\partial w} = 1.1111$
- If $\frac{\partial \mathcal{L}}{\partial w}$ rescaled to 1 → $\frac{\partial \mathcal{L}_5}{\partial w} \propto 10^{-5}$



وابستگی‌های طولانی–مدت قابل چشم‌پوشی است:
(یادگیری فقط بر روی کوتاه–مدت تمرکز می‌کند.)

$$\frac{\partial \mathcal{L}}{\partial w} = \frac{\partial \mathcal{L}_1}{\partial w} + \frac{\partial \mathcal{L}_2}{\partial w} + \frac{\partial \mathcal{L}_3}{\partial w} + \frac{\partial \mathcal{L}_4}{\partial w} + \frac{\partial \mathcal{L}_5}{\partial w}$$

اضمحلال گرادیان‌ها

رفع مشکل

FIXING VANISHING GRADIENTS

* رگولاریزاسیون بر روی وزن‌های بازگشتی (سیگنال خطا را مجبور می‌کند که مضمحل نشود).

$$\Omega = \sum_t \Omega_t = \sum_t \left(\frac{\left| \frac{\partial \mathcal{L}}{\partial c_{t+1}} \frac{\partial c_{t+1}}{\partial c_t} \right|}{\left| \frac{\partial \mathcal{L}}{\partial c_{t+1}} \right|} - 1 \right)^2$$

* استفاده از ماژول‌های بازگشتی پیشرفته، مانند:
  - **ماژول حافظه‌ی کوتاه‌مدت طولانی** (Long Short-Term Memory Module)
  - **ماژول واحد بازگشتی دروازه‌گذاری شده** (Gated Recurrent Unit Module)

اضمحلال گرادیان‌ها

رفع مشکل

FIXING VANISHING GRADIENTS

سیگنال خطا در طول زمان، باید دارای نُرم خیلی بزرگ یا خیلی کوچک نباشد.

راه‌حل: استفاده از یک تابع فعال‌سازی که مشتق آن مساوی با 1 باشد.

⇓

گرادیان‌ها نه خیلی بزرگ می‌شوند و نه خیلی کوچک

شبکه‌های عصبی بازگشتی

۲

حافظه‌ی کوتاه-مدت طولانی

حافظه‌ی کوتاه‌مدت طولانی

LONG SHORT-TERM MEMORY

$$i = \sigma\left(x_t U^{(i)} + m_{t-1} W^{(i)}\right)$$

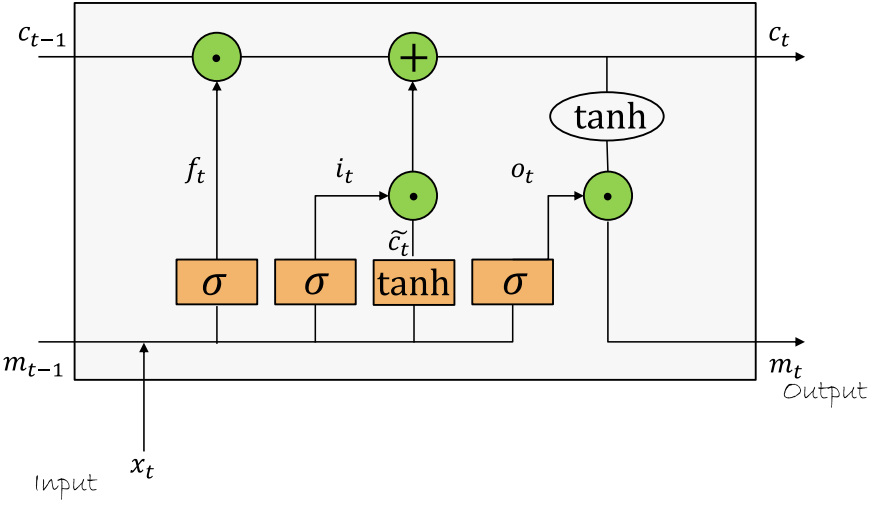
$$f = \sigma\left(x_t U^{(f)} + m_{t-1} W^{(f)}\right)$$

$$o = \sigma\left(x_t U^{(o)} + m_{t-1} W^{(o)}\right)$$

$$\tilde{c_t} = \tanh(x_t U^{(g)} + m_{t-1} W^{(g)})$$

$$c_t = c_{t-1} \odot f + \tilde{c_t} \odot i$$

$$m_t = \tanh(c_t) \odot o$$



حافظه‌ی کوتاه‌مدت طولانی

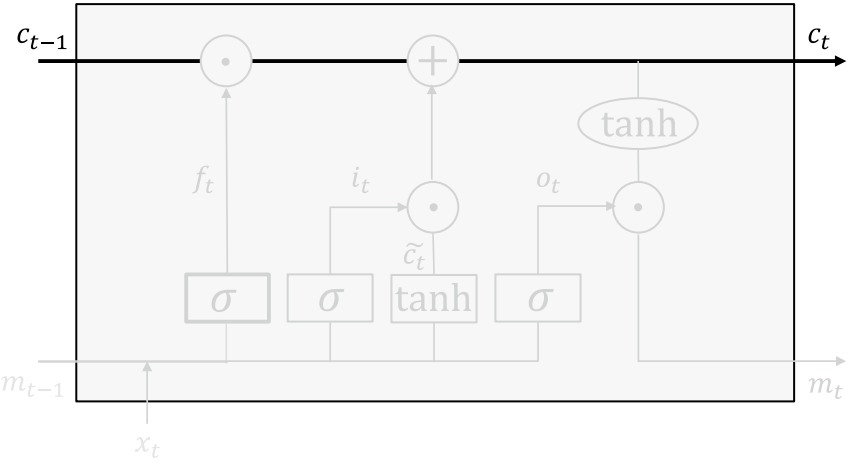
حالت سلول

CELL STATE

حالت سلول، اطلاعات اساسی را در طول زمان حمل می‌کند.

Cell state line

$$i = \sigma\left(x_t U^{(i)} + m_{t-1} W^{(i)}\right)$$
$$f = \sigma\left(x_t U^{(f)} + m_{t-1} W^{(f)}\right)$$
$$o = \sigma\left(x_t U^{(o)} + m_{t-1} W^{(o)}\right)$$
$$\tilde{c}_t = \tanh\left(x_t U^{(g)} + m_{t-1} W^{(g)}\right)$$
$$c_t = c_{t-1} \odot f + \tilde{c}_t \odot i$$
$$m_t = \tanh(c_t) \odot o$$



حافظه‌ی کوتاه‌مدت طولانی

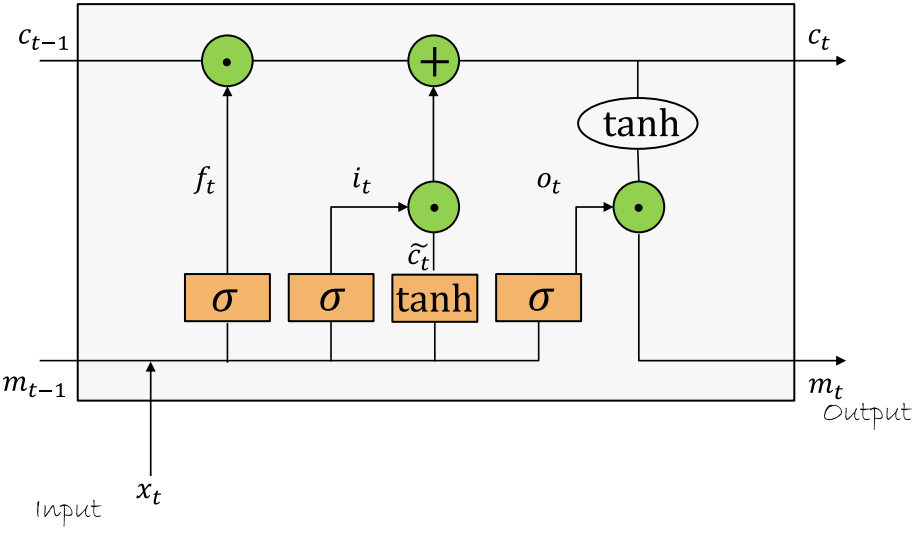
غیرخطیت‌های LSTM

LSTM Non-Linearities

$\sigma \in (0,1)$ **گیت (دروازه) کنترل (control gate):** شبیه یک سوئیچ عمل می‌کند.

$\tanh \in (-1,1)$ غیرخطیت بازگشتی

$$i = \sigma\left(x_t U^{(i)} + m_{t-1} W^{(i)}\right)$$
$$f = \sigma\left(x_t U^{(f)} + m_{t-1} W^{(f)}\right)$$
$$o = \sigma\left(x_t U^{(o)} + m_{t-1} W^{(o)}\right)$$
$$\tilde{c}_t = \tanh\left(x_t U^{(g)} + m_{t-1} W^{(g)}\right)$$
$$c_t = c_{t-1} \odot f + \tilde{c}_t \odot i$$
$$m_t = \tanh(c_t) \odot o$$



حافظه‌ی کوتاه‌مدت طولانی

گام به گام با LSTM: گام ۱

LSTM STEP-BY-STEP: STEP 1

برای مثال: می‌خواهیم یک جمله را مدل کنیم.

باید تصمیم بگیریم برای حافظه‌ی جدید چه چیزی را فراموش کنیم و چه چیزی را به‌خاطر بیاوریم.

سیگموئید = 1 ⇐ به‌یادآوری همه‌چیز
سیگموئید = 0 ⇐ فراموش کردن همه‌چیز

$$i_t = \sigma\left(x_t U^{(i)} + m_{t-1} W^{(i)}\right)$$

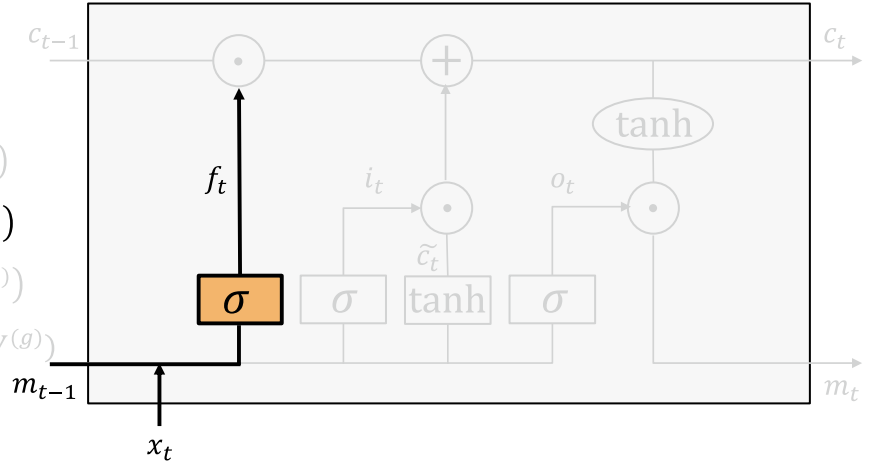
$$f_t = \sigma\left(x_t U^{(f)} + m_{t-1} W^{(f)}\right)$$

$$o_t = \sigma\left(x_t U^{(o)} + m_{t-1} W^{(o)}\right)$$

$$\tilde{c}_t = \tanh\left(x_t U^{(g)} + m_{t-1} W^{(g)}\right)$$

$$c_t = c_{t-1} \odot f + \tilde{c}_t \odot i$$

$$m_t = \tanh(c_t) \odot o$$



حافظه‌ی کوتاه‌مدت طولانی

گام به گام با LSTM: گام ۲

LSTM STEP-BY-STEP: STEP 2

باید تصمیم بگیریم چه اطلاعات جدیدی باید به حافظه‌ی جدید اضافه شود.

- ورودی i_t را مدوله می‌کنیم.
- حافظه‌های کاندیدای $\widetilde{c}_t$ را تولید می‌کنیم.

$$i_t = \sigma\left(x_t U^{(i)} + m_{t-1} W^{(i)}\right)$$

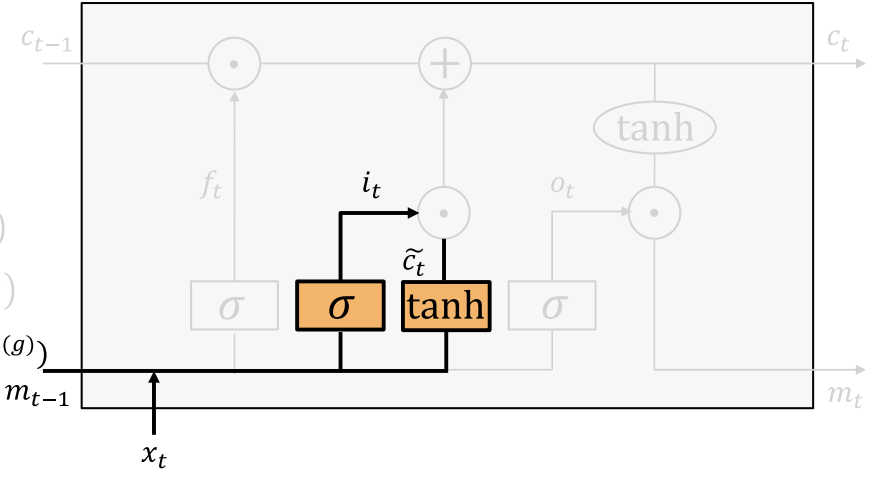
$$f_t = \sigma\left(x_t U^{(f)} + m_{t-1} W^{(f)}\right)$$

$$o_t = \sigma\left(x_t U^{(o)} + m_{t-1} W^{(o)}\right)$$

$$\widetilde{c}_t = \tanh\left(x_t U^{(g)} + m_{t-1} W^{(g)}\right)$$

$$c_t = c_{t-1} \odot f + \widetilde{c}_t \odot i$$

$$m_t = \tanh(c_t) \odot o$$



حافظه‌ی کوتاه‌مدت طولانی

گام به گام با LSTM: گام ۳

LSTM STEP-BY-STEP: STEP 3

حالت فعلی سلول c_t را محاسبه و به‌هنگام می‌کنیم، بر اساس:

- حالت قبلی سلول
- آنچه تصمیم داریم فراموش کنیم
- آنچه به‌عنوان ورودی مجاز می‌دانیم
- حافظه‌های کاندیدا

$$i_t = \sigma\left(x_t U^{(i)} + m_{t-1} W^{(i)}\right)$$

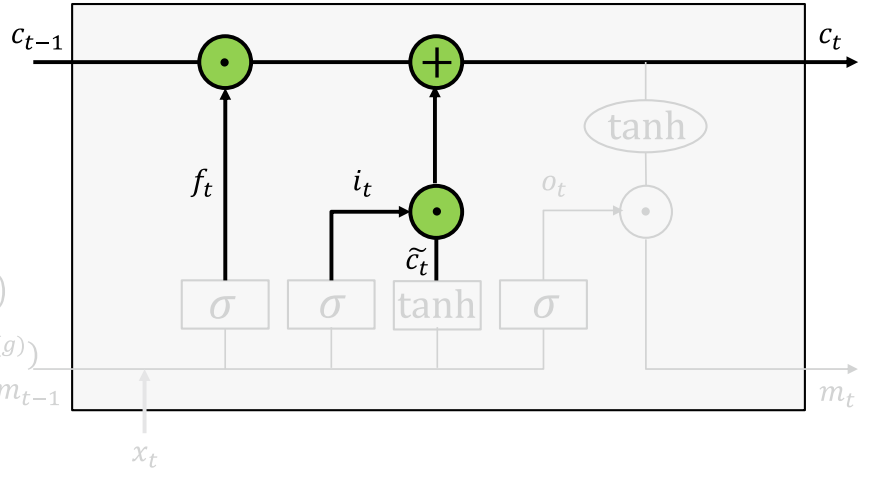
$$f_t = \sigma\left(x_t U^{(f)} + m_{t-1} W^{(f)}\right)$$

$$o_t = \sigma\left(x_t U^{(o)} + m_{t-1} W^{(o)}\right)$$

$$\tilde{c}_t = \tanh\left(x_t U^{(g)} + m_{t-1} W^{(g)}\right)$$

$$c_t = c_{t-1} \odot f + \tilde{c}_t \odot i$$

$$m_t = \tanh(c_t) \odot o$$



حافظه‌ی کوتاه‌مدت طولانی

گام به گام با LSTM: گام ۴

LSTM STEP-BY-STEP: STEP 4

خروجی را مدوله می‌کنیم؛ حافظه‌ی جدید را تولید می‌کنیم.

اگر حالت سلول حاوی چیز مربوطی است
⇐ سیگموئید = 1

$$i_t = \sigma\left(x_t U^{(i)} + m_{t-1} W^{(i)}\right)$$

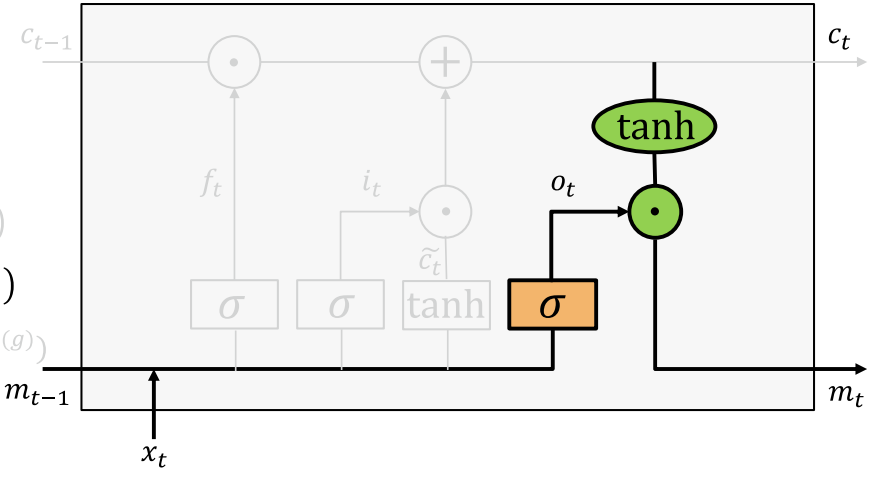
$$f_t = \sigma\left(x_t U^{(f)} + m_{t-1} W^{(f)}\right)$$

$$o_t = \sigma\left(x_t U^{(o)} + m_{t-1} W^{(o)}\right)$$

$$\tilde{c}_t = \tanh\left(x_t U^{(g)} + m_{t-1} W^{(g)}\right)$$

$$c_t = c_{t-1} \odot f + \tilde{c}_t \odot i$$

$$m_t = \tanh(c_t) \odot o$$

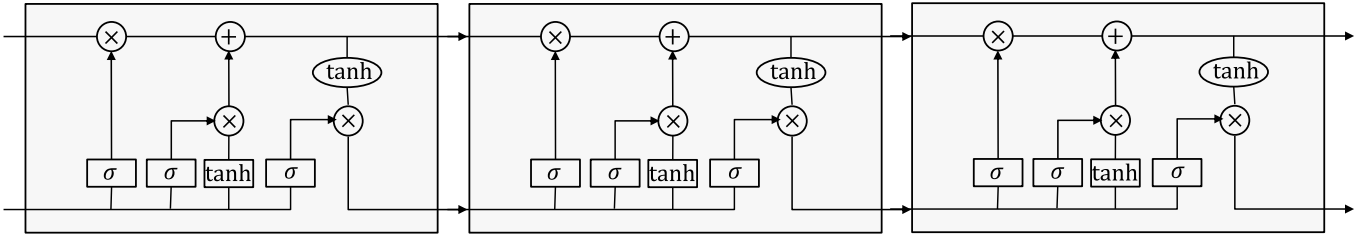


حافظه‌ی کوتاه‌مدت طولانی

شبکه‌ی بازشده

LSTM UNROLLED NETWORK

به‌لحاظ ماکروسکوپی، بسیار شبیه به شبکه‌های عصبی بازگشتی استاندارد است؛
اما موتور آن کمی متفاوت است (پیچیده‌تر)
[زیرا گیت‌های LSTMهای آن وابستگی‌های کوتاه مدت و بلند را تسخیر می‌کند.]

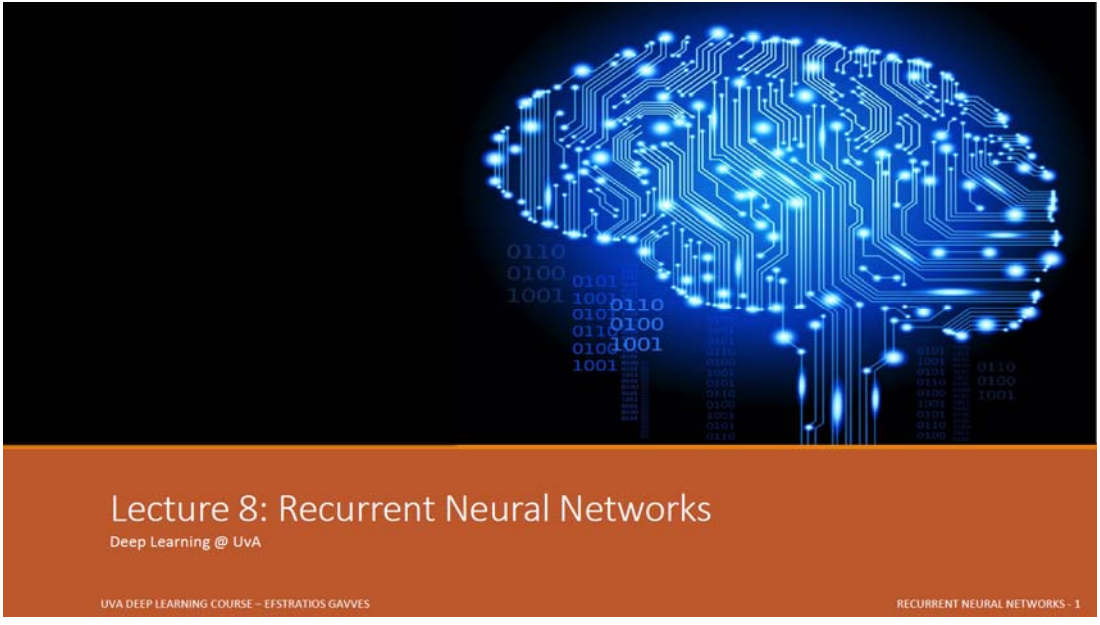


شبکه‌های عصبی بازگشتی

۳

منابع

منبع اصلی


Lecture 8: Recurrent Neural Networks
Deep Learning @ UvA
UVA DEEP LEARNING COURSE – EFSTRATIOS GAVVES
RECURRENT NEURAL NETWORKS - 1