

رسالة محمد

یادگیری عمیق

مدرس: محمدرضا محمدی

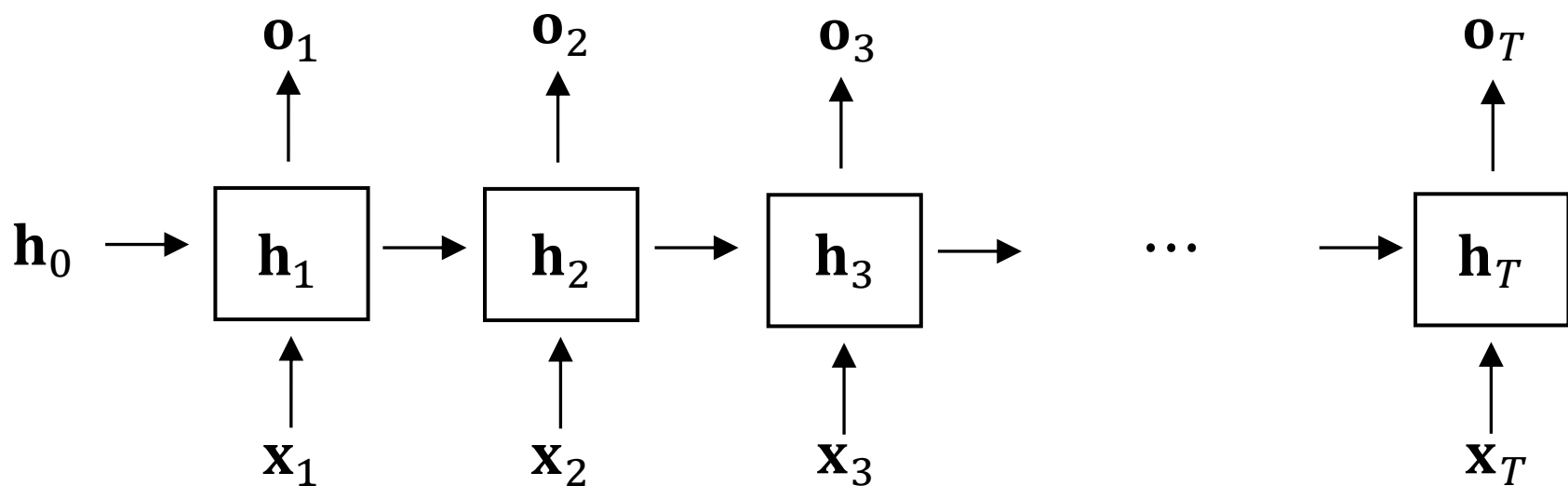
بهار ۱۴۰۲

شبکه‌های عصبی بازگشتی

Recurrent Neural Networks

شبکه‌های بازگشتی دوجهته

- تمام RNN‌هایی که تا به حال بررسی کرده‌ایم دارای ساختار «علّی» هستند
 - حالت پنهان در زمان t فقط به ورودی‌های گذشته، $\mathbf{x}_1, \dots, \mathbf{x}_{t-1}$ و ورودی فعلی \mathbf{x}_t وابسته است
- در برخی از کاربردها، می‌توانیم از تمام مقادیر قبلی و بعدی برای پیش‌بینی \mathbf{o}_t استفاده کنیم

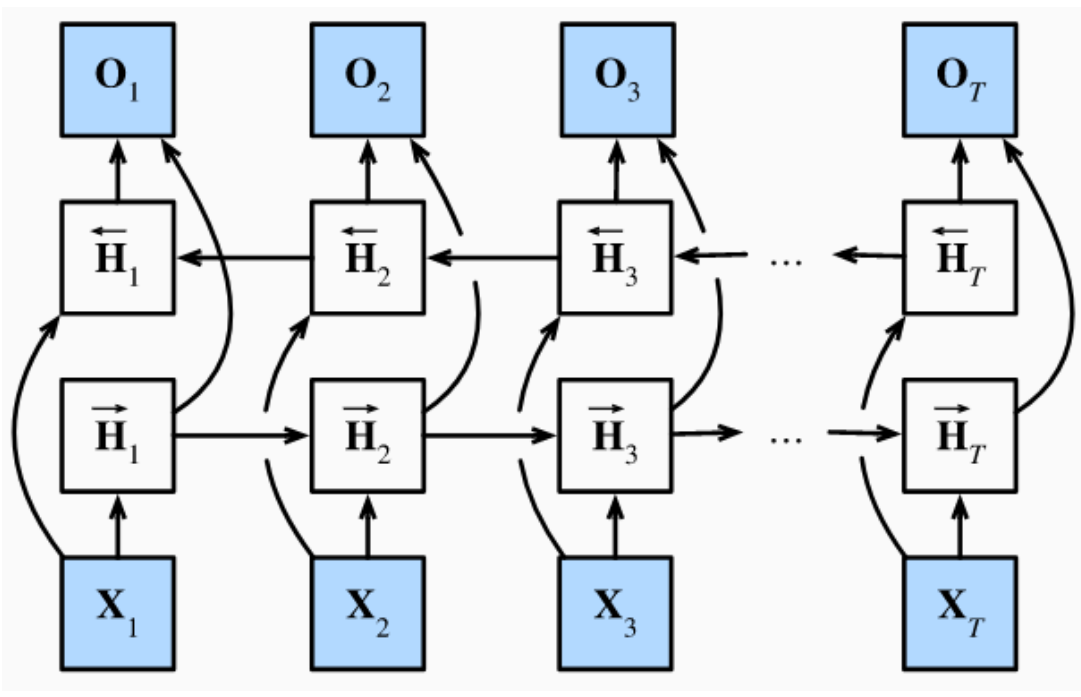


شبکه‌های بازگشتی دوجبهته

- RNN‌های دوجبهته متداول شامل یک RNN رو به جلو از ابتدای دنباله و یک RNN رو به عقب از انتهای دنباله هستند

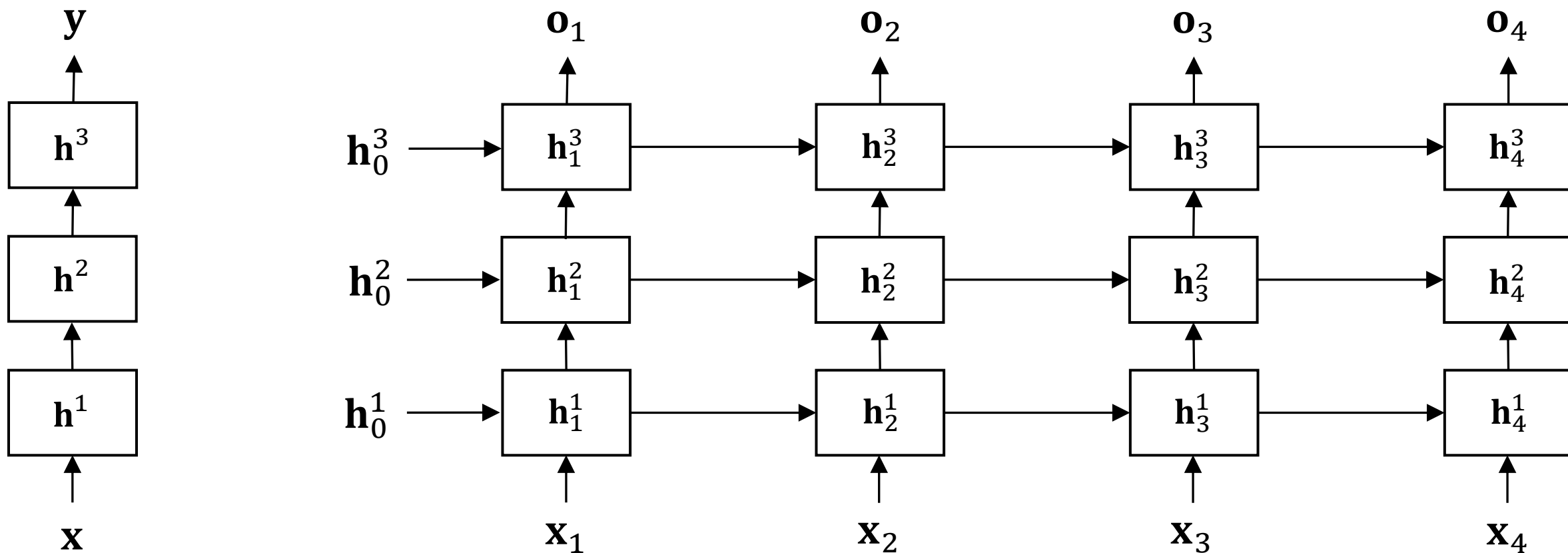
- \mathbf{o}_t هم به زمان گذشته و هم به زمان آینده بستگی دارد

- هر کدام از SimpleRNN و LSTM و GRU در این ساختار دوجبهته قابل استفاده هستند



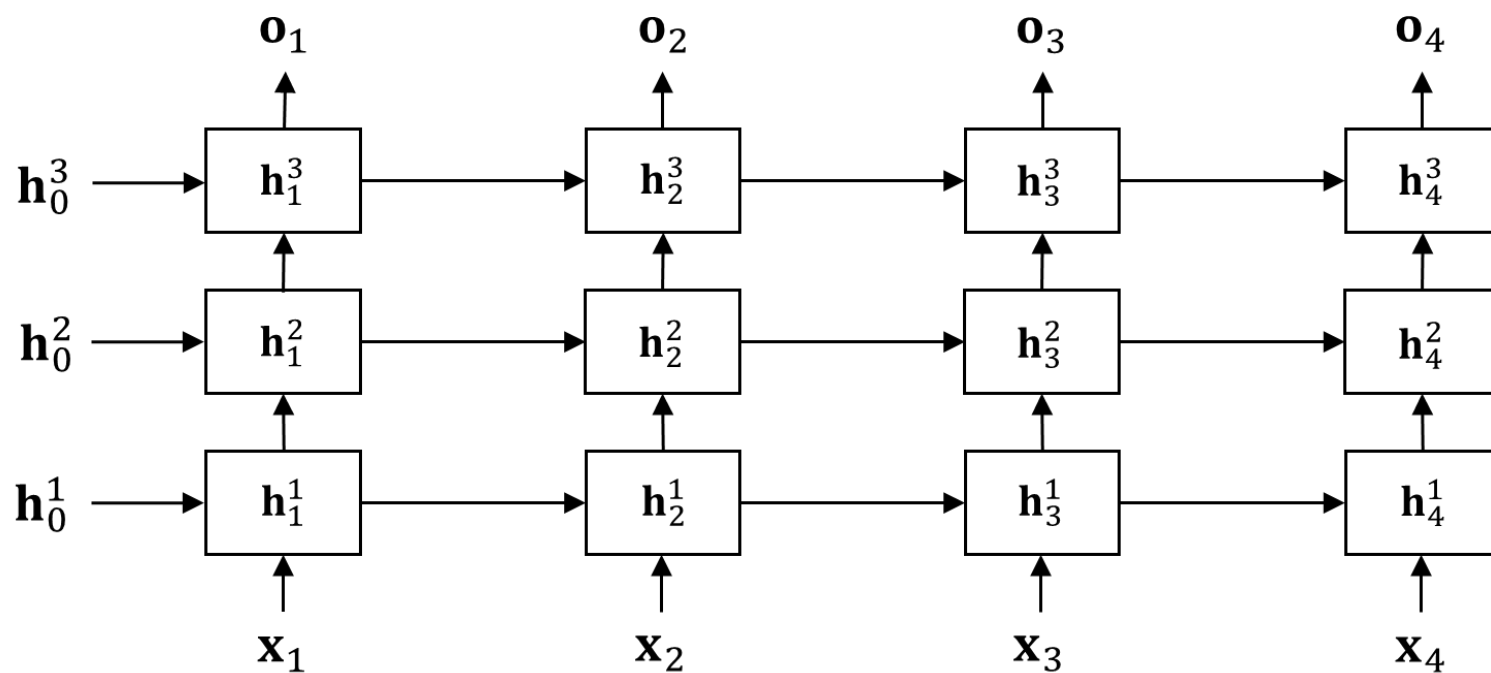
شبکه‌های بازگشتی عمیق

$$\mathbf{h}_3^2 = \tanh(\mathbf{W}^2 [\mathbf{h}_2^2 \quad \mathbf{h}_3^1] + \mathbf{b}^2)$$



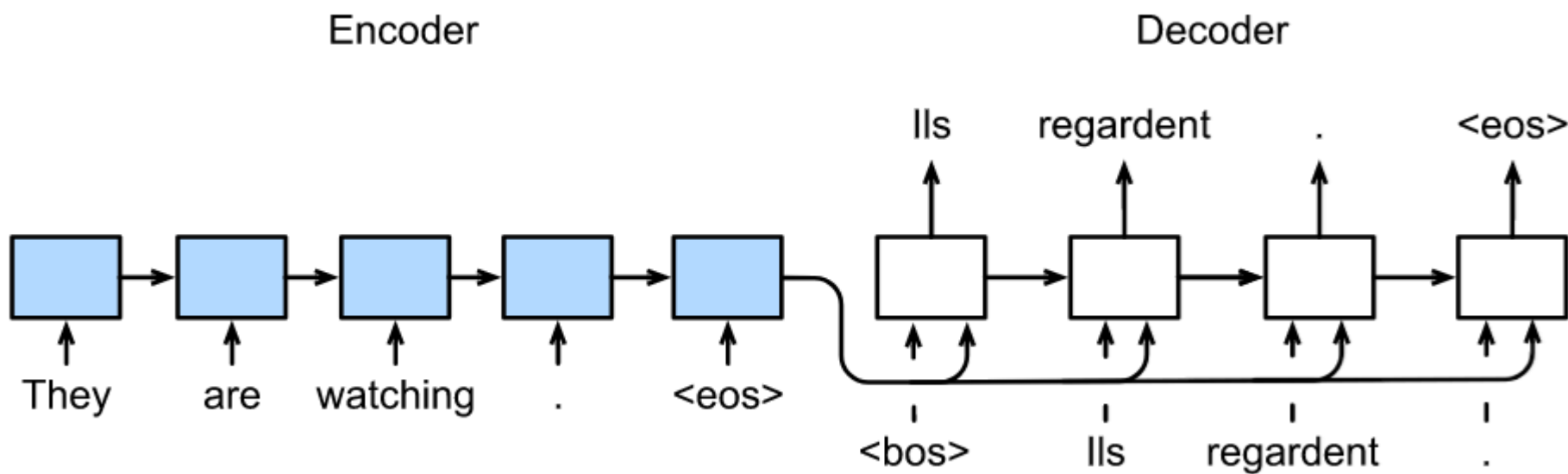
شبکه‌های بازگشتی عمیق

- برای RNN ها، داشتن سه لایه بسیار زیاد است
- بلوک‌های مورد استفاده می‌توانند SimpleRNN، GRU یا LSTM باشد
- هر کدام می‌توانند دوجهته باشند



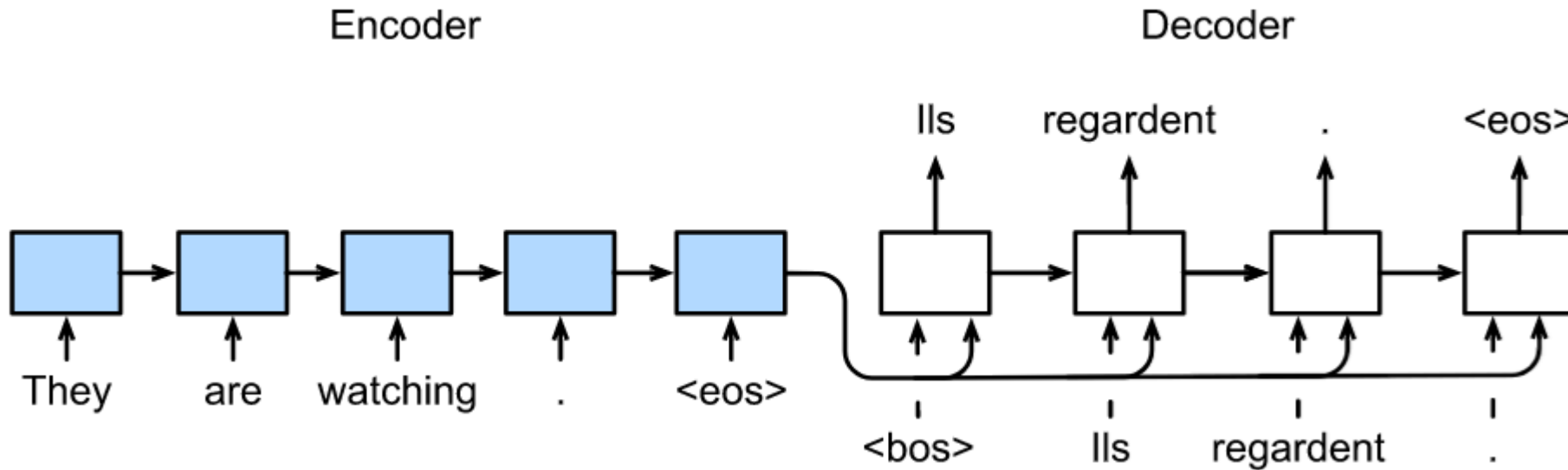
ترجمه ماشینی

- ترجمه ماشینی با استفاده از معماری Encoder-Decoder قابل پیاده‌سازی است
 - توسط Encoder دنباله ورودی به یک بردار تبدیل می‌شود
 - توسط Decoder این بردار به یک دنباله تبدیل می‌شود
- برای آنکه بتوانیم شبکه را با minibatch آموزش بدهیم، لازم است با pad طول دنباله‌ها را یکی کنیم



ترجمه ماشینی

- پس از آموزش مدل برای ترجمه ماشینی، در زمان آزمون می‌توان توکن‌های خروجی را یکی یکی تولید کرد تا به توکن `<eos>` رسید
 - به این رویکرد حریصانه (Greedy) گفته می‌شود
 - اگر توکن اول به خوبی انتخاب نشود ممکن است دنباله تولید شده مطلوب نباشد



جستجوی حریصانه

- در هر گام زمانی، توکن دارای بالاترین احتمال شرطی انتخاب می‌شود

$$y_{t'} = \arg \max_{y \in \mathcal{Y}} P(y | y_1, \dots, y_{t'-1}, \mathbf{c})$$

- در حالت بهینه، لازم است دنباله‌ای انتخاب شود که احتمال آن در مجموع بالاتر باشد

$$\prod_{t'=1}^{T'} P(y_{t'} | y_1, \dots, y_{t'-1}, \mathbf{c})$$

- جستجوی حریصانه ممکن است به دنباله بهینه منجر نشود
- احتمال دنباله اول ۰.۰۴۸ و احتمال دنباله دوم ۰.۰۵۴ است

Time step	1	2	3	4
A	0.5	0.1	0.2	0.0
B	0.2	0.4	0.2	0.2
C	0.2	0.3	0.4	0.2
<eos>	0.1	0.2	0.2	0.6

Time step	1	2	3	4
A	0.5	0.1	0.1	0.1
B	0.2	0.4	0.6	0.2
C	0.2	0.3	0.2	0.1
<eos>	0.1	0.2	0.1	0.6

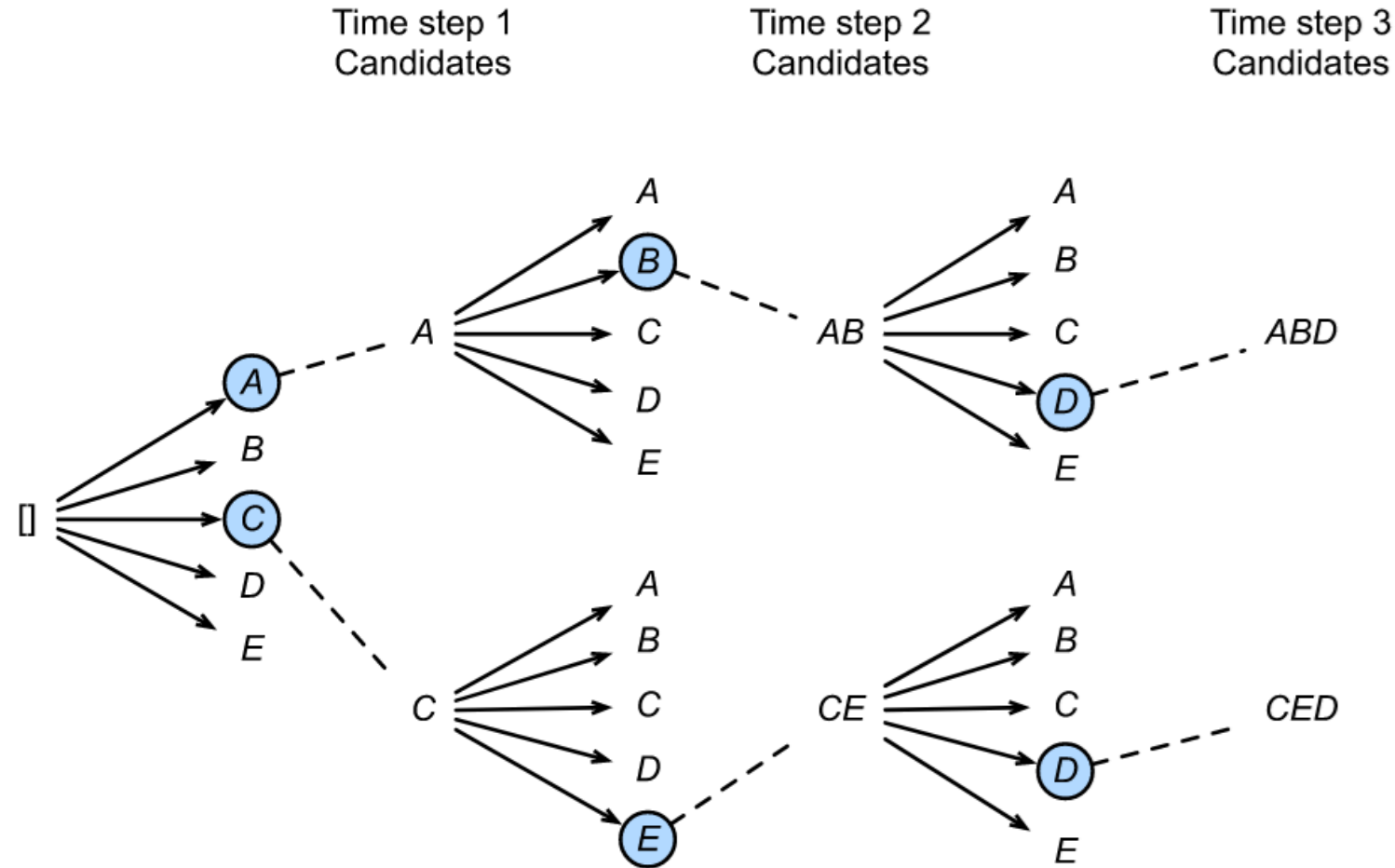
جستجوی جامع (Exhaustive)

- برای بدست آوردن دنباله بهینه، می توان احتمال تمام دنباله های ممکن را محاسبه کرد
- هزینه محاسباتی این جستجو $O(|Y|^{T'})$ می تواند خیلی زیاد باشد
 - به عنوان مثال اگر $|Y| = 10000$ و $T' = 10$ باشد، لازم است احتمال 10^{40} دنباله مختلف محاسبه شود
- هزینه محاسباتی جستجوی حریصانه $O(|Y|^{T'})$ است
- می توان از یک روش جستجوی میانی استفاده کرد که هم احتمال دستیابی به بهترین دنباله بالا باشد و هم هزینه محاسباتی خیلی زیاد نباشد

جستجوی Beam

- در گام ۱، k توکن دارای بالاترین احتمال شرطی انتخاب می‌شوند
- k یک ابرپارامتر است که beam size نامیده می‌شود
- در گام ۲، k دنباله دارای بالاترین احتمال که با یکی از توکن‌های گام ۱ شروع می‌شوند را انتخاب می‌کنیم
- در گام‌های بعدی، k دنباله دارای بالاترین احتمال که با دنباله‌های قبل شروع می‌شوند را انتخاب می‌کنیم

جستجوی Beam



$$P(y_1 | \mathbf{c}) \Rightarrow A, C$$

$$\begin{cases} P(A, y_2 | \mathbf{c}) = P(A | \mathbf{c})P(y_2 | A, \mathbf{c}) \\ P(C, y_2 | \mathbf{c}) = P(C | \mathbf{c})P(y_2 | C, \mathbf{c}) \end{cases}$$

$$\Rightarrow (A, B), (C, D)$$

جستجوی Beam

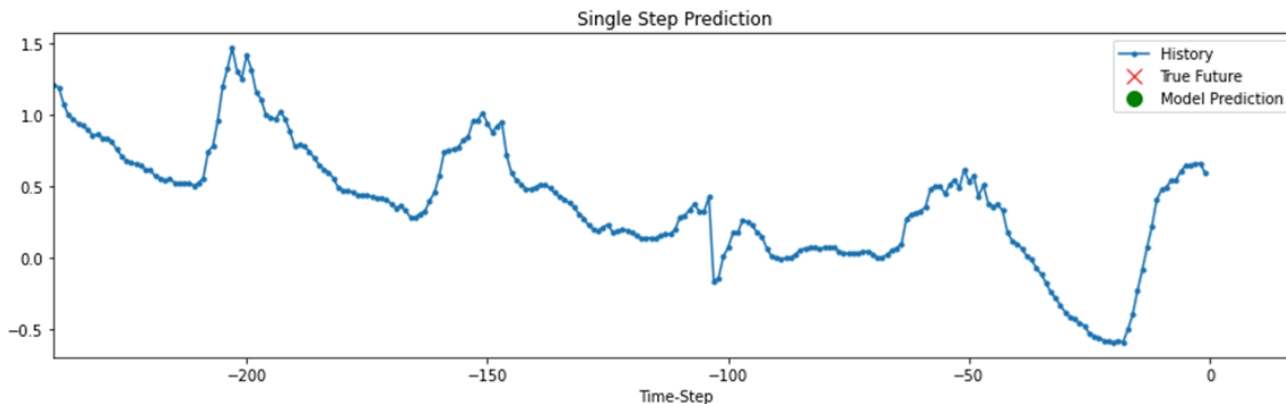
- در پایان، از تمام $|\mathcal{Y}|^{T'}$ دنباله ممکن، تنها احتمال kT' دنباله محاسبه می شود
- احتمال دنباله های طولانی کوچک خواهد شد، به همین دلیل احتمال یک دنباله را بر اساس طول آن نرمالیزه می کنیم

$$\frac{1}{L^\alpha} \log P(y_1, \dots, y_L | \mathbf{c}) = \frac{1}{L^\alpha} \sum_{t'=1}^L \log P(y_{t'} | y_1, \dots, y_{t'-1}, \mathbf{c})$$

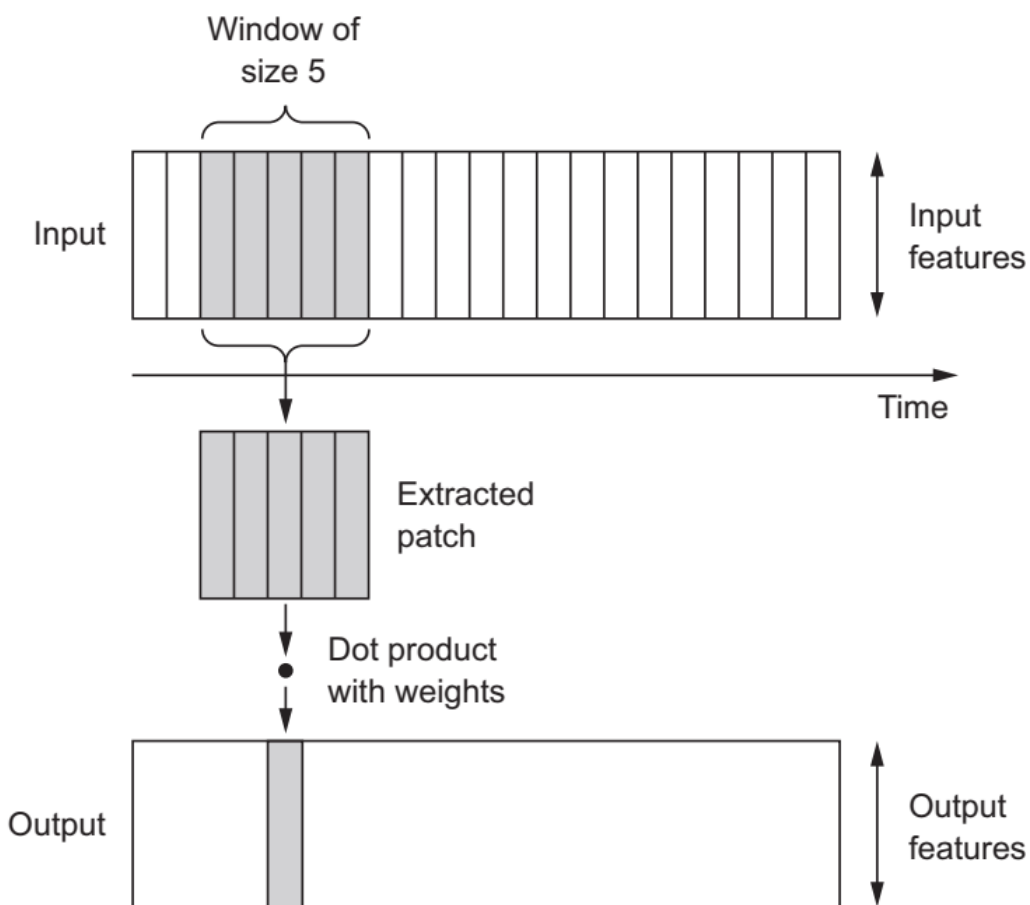
- اگر $k = 1$ باشد، معادل با جستجوی حریصانه است
- اگر $k = |\mathcal{Y}|$ باشد، معادل با جستجوی جامع است

پردازش دنباله‌ها با لایه‌های کانولوشنی

- به همان دلایلی که لایه‌های کانولوشنی در بینایی ماشین بسیار موفق بوده‌اند، برای پردازش دنباله‌ها نیز موثر هستند
- زمان در دنباله‌ها مشابه با یک بعد مکانی در یک تصویر دو بعدی است
- شبکه‌های کانولوشنی یک بعدی می‌توانند با RNN‌ها در مسائل پردازش دنباله‌ها رقابت کنند
 - معمولاً با هزینه محاسباتی بسیار کمتر



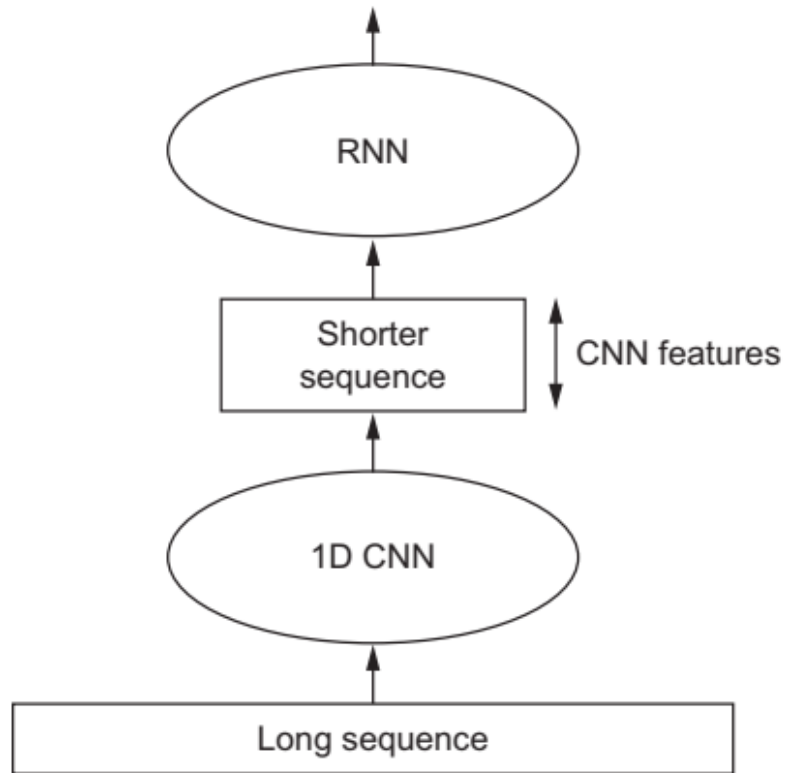
کانولوشن یک بعدی



- لایه های کانولوشنی یک بعدی می توانند الگوهای محلی را در یک دنباله تشخیص دهند
- الگویی که در یک موقعیت خاص در یک دنباله آموخته می شود، بعداً می تواند در موقعیت دیگری تشخیص داده شود، که باعث می شود لایه های کانولوشنی یک بعدی نسبت به جابجایی تغییرناپذیر باشند
- می توان از گام (Stride) و تجمیع (Pooling) هم استفاده کرد

ترکیب RNN ها و CNN ها

- یک استراتژی برای ترکیب مزایای CNN ها (از جمله سرعت) با مزایای RNN ها (از جمله حساسیت به ترتیب مقادیر)، استفاده از CNN یک بعدی به عنوان مرحله پیش پردازش قبل از RNN است

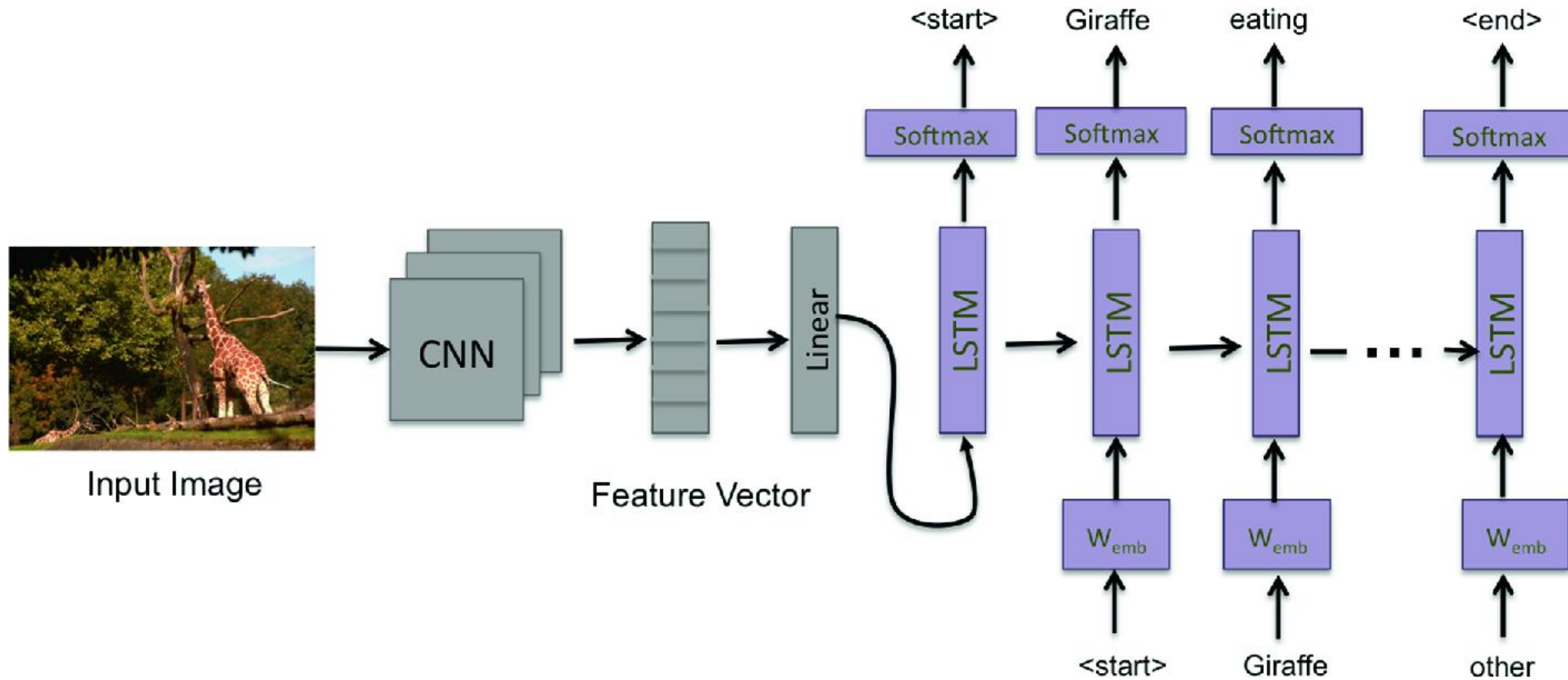


- به خصوص برای پردازش دنباله های طولانی مفید است که نمی توان آنها را به طور موثری تنها با استفاده از RNN پردازش کرد

ترکیب RNN ها و CNN ها

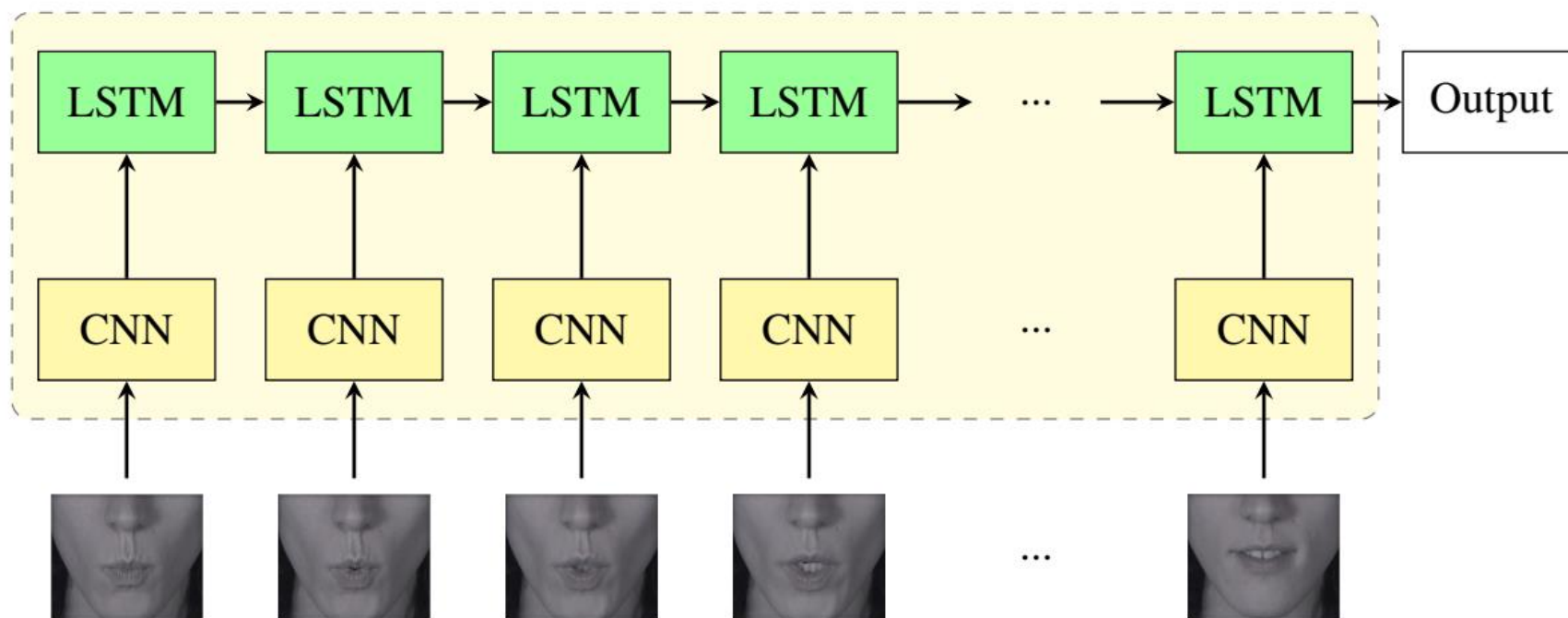
- شبکه‌های کانولوشنی یک‌بعدی جایگزین سریع‌تری برای RNN ها در برخی مسائل هستند
- از آنجایی که RNN ها برای پردازش دنباله‌های بسیار طولانی، بسیار گران هستند، اما CNN های یک‌بعدی ارزان هستند، استفاده از یک CNN یک‌بعدی به عنوان گام پیش‌پردازش قبل از RNN برای کوتاه کردن دنباله و استخراج بازنمایی‌های موثر می‌تواند مفید باشد

شرح تصویر با CNN ها و RNN ها

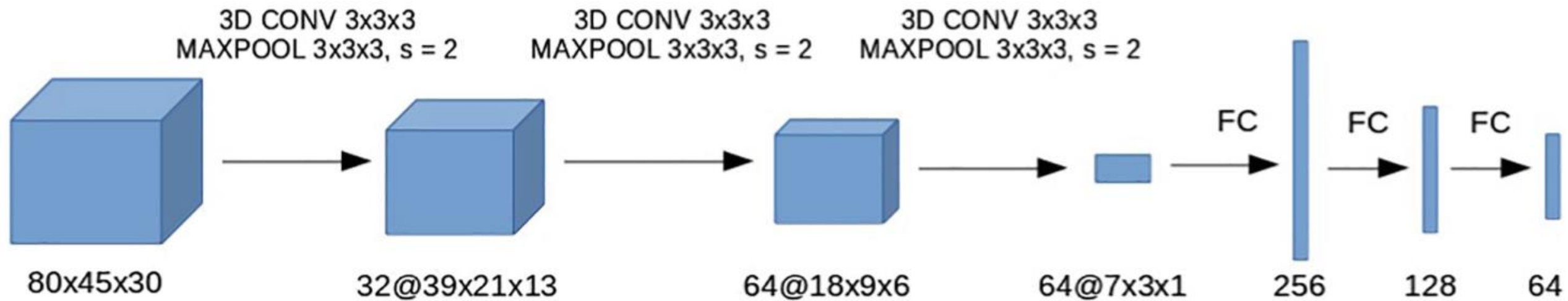
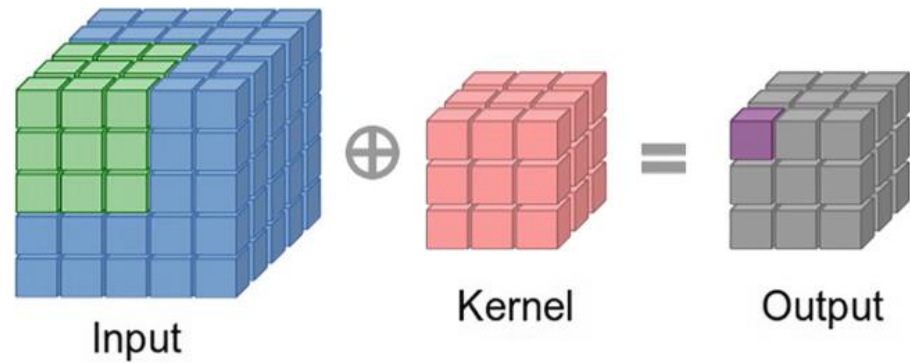


تحلیل ویدئو با CNN ها و RNN ها

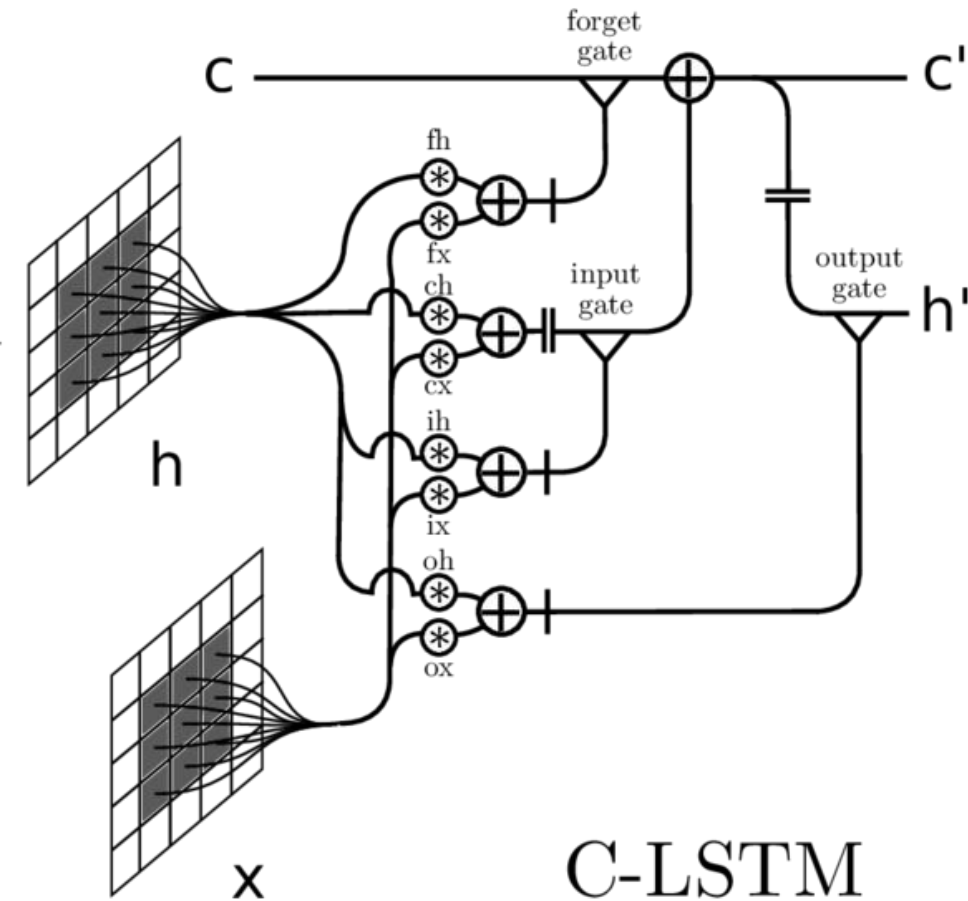
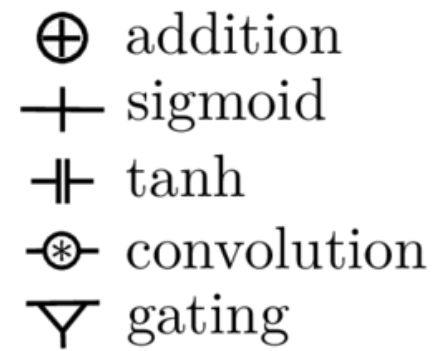
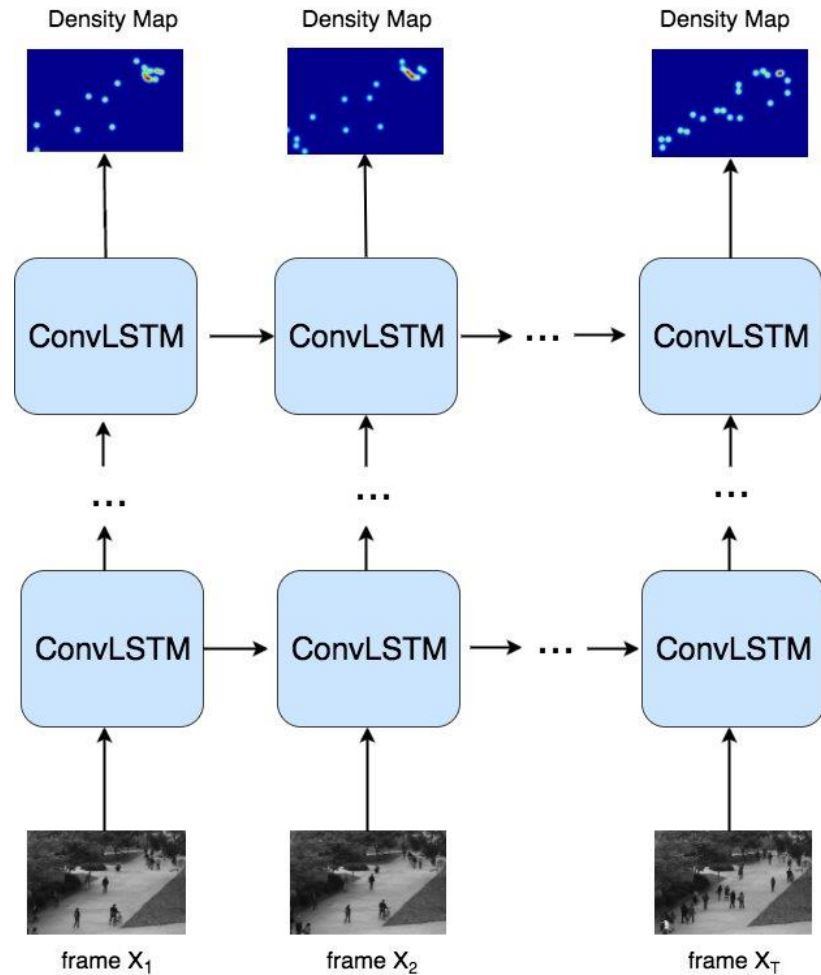
- ترکیبی از CNN و RNN پرکاربردترین معماری یادگیری عمیق برای لبخوانی خودکار (ALR) است



3D CNN



ConvLSTM



C-LSTM