

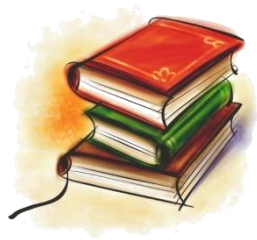
شبکه‌های عصبی مصنوعی

شبکه‌های بازگشتی

هادی ویسی

h.veisi@ut.ac.ir

دانشگاه تهران - دانشکده علوم و فنون نوین



- معرفی و کاربردهای شبکه عصبی بازگشتی
- تاریخچه شبکه عصبی بازگشتی
- شبکه عصبی هاپفیلد
- آموزش شبکه عصبی بازگشتی
- شبکه عصبی المان
 - ساختار، آموزش، کاربرد و مثال (گرامر ربر)
- شبکه عصبی جردن
- مشکل فراموشی در شبکه‌های بازگشتی
- شبکه عصبی حافظه کوتاه مدت ماندگار (LSTM)
 - ساختار، آموزش، کاربرد و مثال



معرفی ...

عقیده کاوی (دسته‌بندی نظرات کاربران)

• موافق/مخالف

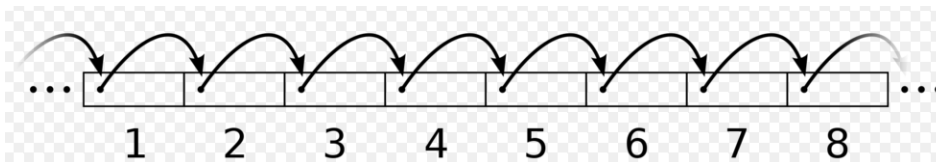
1	امید دلیر (1393/1/24) : فوق العادس	👍	•
0	محمود ابراهیمی جاویدی (1393/1/22) : من این لب تاپ 5 ماه دارم و از همه نظر عالی مثل: کیفیت صفحه نمایش یا کیفیت لولاش برای لمسش بودنش و...	👍	•
1	مرتضی اکبری (1393/1/17) : یکی از مشکلات از نظر من کارت گرافیکشه. منظورم باس کارت گرافیکه!	👍	•
0	هادی کریم (1392/12/22) : سلام..اصلا تو خریدنش شک نکید..من خریدم خیلی عالی..خیلی بیشتر از عکسش خوشگله..واقعا زیباست..اصلا لولایی دیده نمیشه واستحکامش خیلی بیشتره..به قیمتش می ارزه..صدای فنش هم اصلا اذیت نمیکه..چون میده برای بازی	👍	•
5	ابودر هومن (1392/12/17) : با سلام بعد از 4ماه کار مداوم از هر نظر عالی قدرت خوبی داره..بسیار سریع تاج خیلی خوبی داره..کیفیت واقعا عالی..صفحه عجیبی داره که اصلا خسته نمیشین موقع کار باهاش وزنش هم که عالی..یه همراه همیشگی میشه براتون مرسی	👍	•
0	هادی کریم (1392/12/1) : سلام..اصلا تو خریدنش شک نکید..من خریدم خیلی عالی..خیلی بیشتر از عکسش خوشگله..واقعا زیباست..اصلا لولایی دیده نمیشه واستحکامش خیلی بیشتره..به قیمتش می ارزه..صدای فنش هم اصلا اذیت نمیکه..چون میده برای بازی	👍	•

نمونه دسته بندی با
شبکه عصبی؟

معرفی ...

○ داده متوالی (Sequential Data): داده‌هایی که مقدار فعلی آنها به مقادیر قبلی وابسته است

- فریم‌های (نمونه‌های) سیگنال گفتار
- فریم‌های (تصاویر) متوالی ویدئو
- وضعیت آب و هوا
- قیمت سهام یک شرکت / صنعت
- دنباله‌های تولید شده توسط گرامرها
- کلمات داخل یک متن
- ...



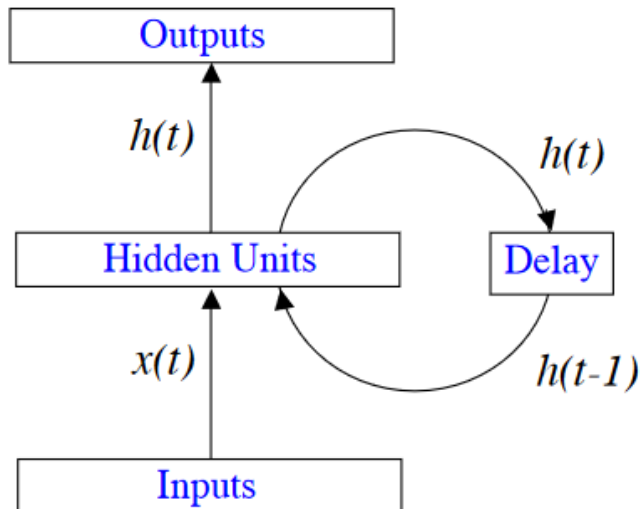
○ شبکه‌های عصبی بازگشتی (RNN: Recurrent Neural Networks)

- شبکه‌هایی که در ساختار آنها یال‌های بازگشت کننده وجود دارد
- بر خلاف شبکه‌های عصبی رو به جلو، یال‌ها می‌توانند تشکیل دور بدهند.

- به دلیل داشتن یال بازگشتی در ساختار خود، قدرت **حافظه‌ای** دارند.

○ مناسب برای پردازش داده‌های متوالی (Sequential Data)

- ساختار: شبیه MLP با بازگشت از نرون‌های مخفی



معرفی ...

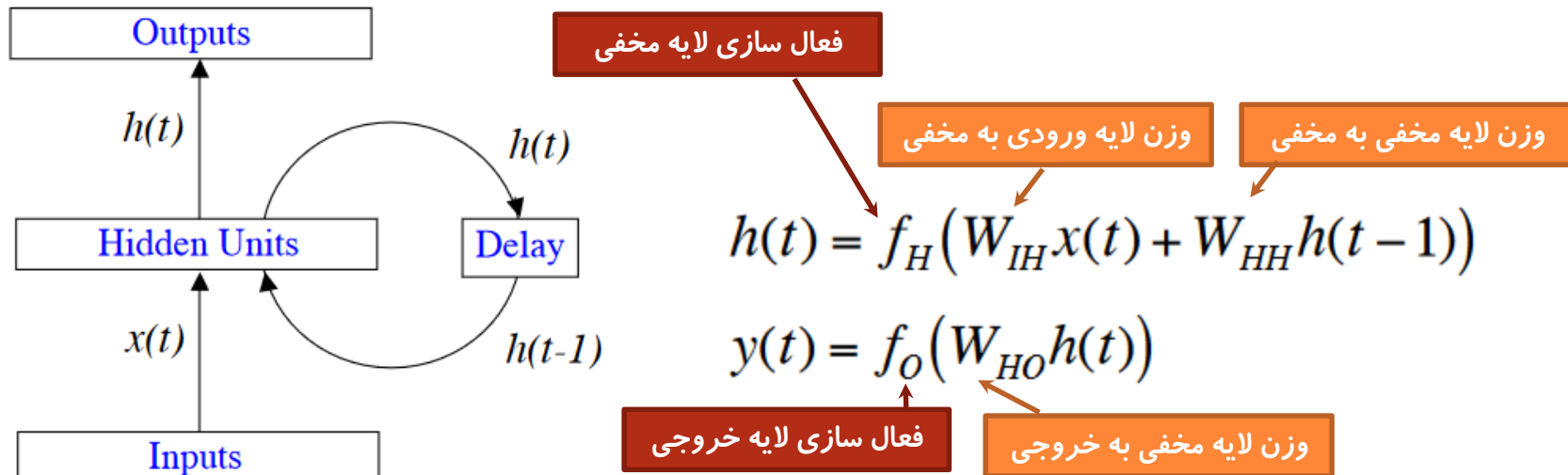
○ شبکه عصبی بازگشتی به عنوان یک سیستم پویا (Dynamic System)

- تعریف کامل سیستم با حالت (State) سیستم: مجموعه‌ای مقادیر حاوی همه اطلاعات

○ مقادیر فعال‌سازی‌های لایه مخفی: $h(t)$

○ مرتبه سیستم پویا = ابعاد فضای حالت

○ در اینجا = تعداد نرون‌های لایه مخفی





معرفی ...

○ قضیه

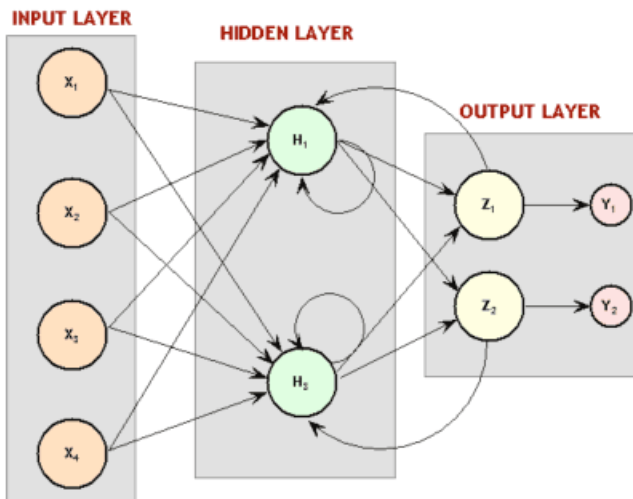
- شبکه عصبی بازگشتی یک تقریب‌زننده جهانی (Universal Approximation) است

یک سیستم پویای غیرخطی را می‌توان با هر تقریبی توسط یک شبکه عصبی بازگشتی که دارای تعداد کافی واحد مخفی سیگموئیدی دارد، تقریب زد

کاربردهای شبکه عصبی بازگشتی

○ مدل‌سازی و پیش‌بینی داده‌های متوالی (Sequential Data) و سری زمانی (Time Series)

- سیگنال گفتار: تشخیص گفتار
- دست‌نوشته: تبدیل دست نوشته به متن الکترونیکی
- امضا: تایید/تشخیص هویت
- دنباله نویسه‌ها یا واژه‌ها در یک متن: تشخیص موضوع متن
- تصاویر پشت سر هم (ویدئو): دنبال کردن اشیا
- پیش‌بینی نرخ ارز/شاخص سهام
- پیش‌بینی بارش باران در روزهای متوالی سال
- پیش‌بینی مقدار مصرف آب در روزهای متوالی
- ...





تاریخچه...

○ دهه ۷۰- اولین شبکه عصبی بازگشتی

• ۱۹۷۷- شبکه عصبی کاملاً بازگشتی

- ارائه شده توسط اندرسون و کوهونن
- هر نرون به تمامی نرون‌های دیگر متصل است

○ دهه ۸۰

• ۱۹۸۲- شبکه عصبی هاپفیلد

- ارائه توسط هاپفیلد
- قانون یادگیری: هب
- کاربرد: برای تشخیص «شناخته بودن» یا «ناشناس» بودن بردار ورودی استفاده می‌شود

• ۱۹۸۸- شبکه عصبی حافظه انجمنی دوسویه

- نام لاتین: Bidirectional Associative Memory (BAM)
- توسط Kosko ارائه شد
- یک شبکه دیگر انجمنی بازگشتی
- این شبکه شامل دولایه نرون است که کاملاً به یکدیگر متصل شده اند



تاریخچه...

○ دهه ۹۰

• ۱۹۹۰ - شبکه عصبی المان

- به این شبکه Simple Recurrent Network (SRN) نیز گفته می‌شود
- معرفی شده توسط جفری المان
- شامل ۴ لایه ورودی، پنهان، بافت و خروجی است.
- دارای اتصالات بازگشتی از لایه پنهان به لایه بافت

• ۱۹۹۶ - شبکه عصبی جردن

- ارائه شده توسط مایکل جردن
- شباهت بسیار زیادی به شبکه المان دارد
- دارای اتصالات بازگشتی از لایه خروجی به لایه بافت

• ۱۹۹۷ - شبکه عصبی بازگشتی دوطرفه

- نام لاتین شبکه Bidirectional Recurrent Neural Network است
- ارائه توسط شوستر و پالیوال
- شامل دو لایه پنهان مجزا
- مزیت: مقدار هدف در هر گام زمانی به کل دنباله ورودی وابسته است



تاریخچه...

○ ۱۹۹۷ - شبکه عصبی حافظه کوتاه مدت ماندگار

○ نام لاتین: Long Short Term Memory (LSTM)

○ توسط هاکریتز و اشمیدبر معرفی شد

○ نرون‌های لایه پنهان با بلوک‌های حافظه جایگزین شدند

○ حل شدن مشکل فراموشی دنباله‌های طولانی

○ ۲۰۰۰ به بعد

• ۲۰۰۱ - توسعه LSTM

○ توسط فلیکس گرز توسعه داده شد

○ اضافه کردن دروازه فراموشی به بلوک حافظه

شبکه هاپفیلد

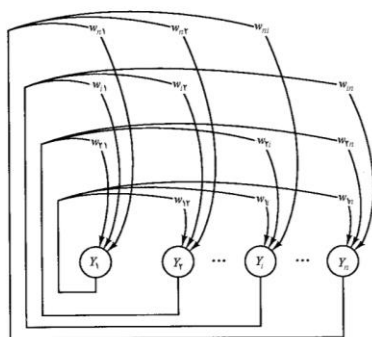
در اسلایدهای قبلی آمده است

شبکه‌های عصبی مصنوعی: حافظه انجمی (پیوند الگو)



شبکه هاپفیلد گستره، ساختار ...

- خروجی هر واحد، به همه واحدهای دیگر (به غیر از خودش) به عنوان ورودی با وزن مربوطه وارد می‌شود



Hadi Veisi (h.veisi@ut.ac.ir)

شبکه‌های عصبی مصنوعی: حافظه انجمی (پیوند الگو)



شبکه هاپفیلد گستره ...

معرفی

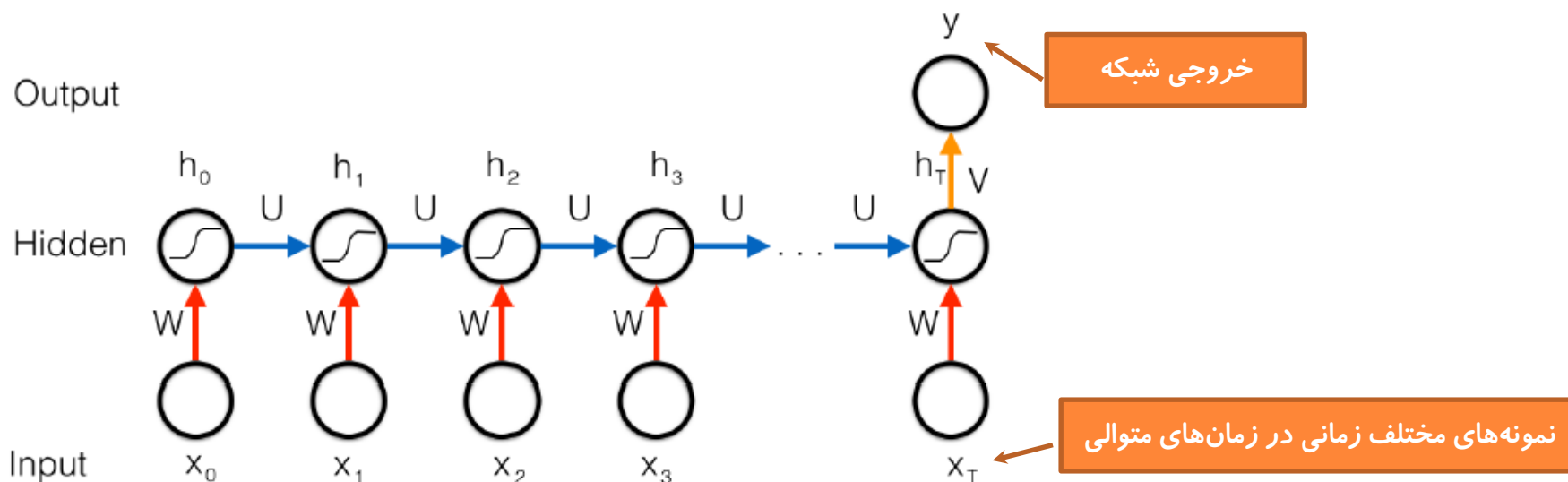
- ارائه توسط هاپفیلد (استاد فیزیک)
- اوایل دهه ۸۰ میلادی (سال‌های ۱۹۸۲ و ۱۹۸۴)
- یک شبکه خودانجمی تکراری شبیه به شبکه‌های توصیف شده
- یک شبکه عصبی کاملاً به هم متصل بوده و دارای وزن‌های متقارن و بدون اتصال به خود
- دارای تفاوت‌های اندک اما مهم با شبکه‌های دیگر خودانجمی تکراری (تأثیر بر همگرایی)
 - در هر بار فقط یکی از واحدها فعال‌سازی خود را به‌روز می‌کند که این به‌روز کردن بر اساس سیگنال‌های دریافتی از واحدهای دیگر است.
 - هر واحد شبکه، علاوه بر سیگنال‌های دریافتی از سایر واحدهای شبکه، یک سیگنال خارجی را نیز دریافت می‌کند که همان ورودی شبکه است
- اثبات همگرایی فعال‌سازی‌ها
 - به کمک تابع انرژی یا تابع لیاپونوف (Lyapunov Function)

Hadi Veisi (h.veisi@ut.ac.ir)

آموزش شبکه ...

عملکرد شبکه

- وابستگی خروجی شبکه به خروجی‌های لایه مخفی به ازای همه ورودی‌ها



$$f(x) = Vh_T$$

$$h_t = \sigma(Uh_{t-1} + Wx_t), \text{ for } t = T, \dots, 1$$

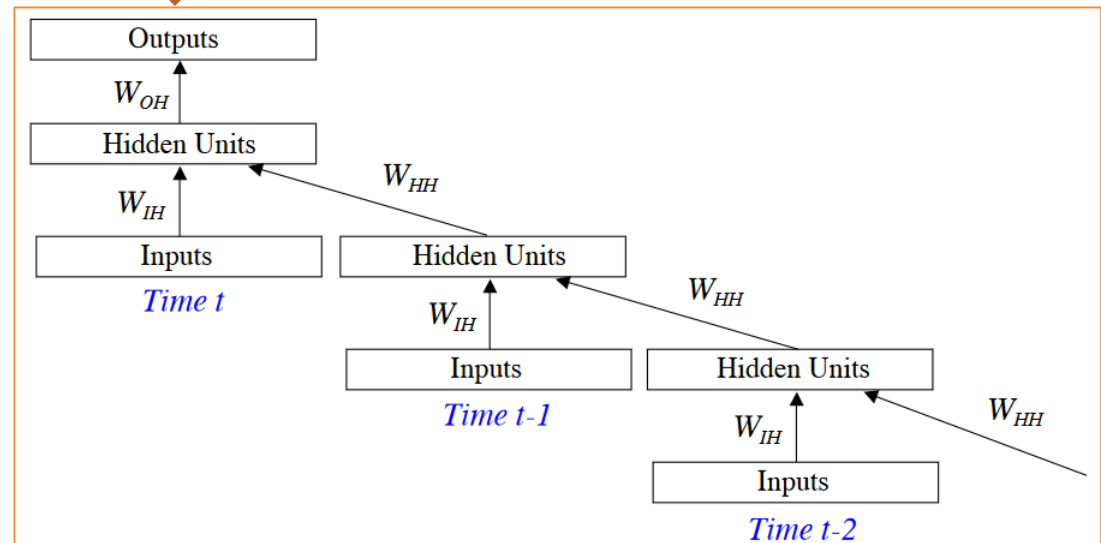
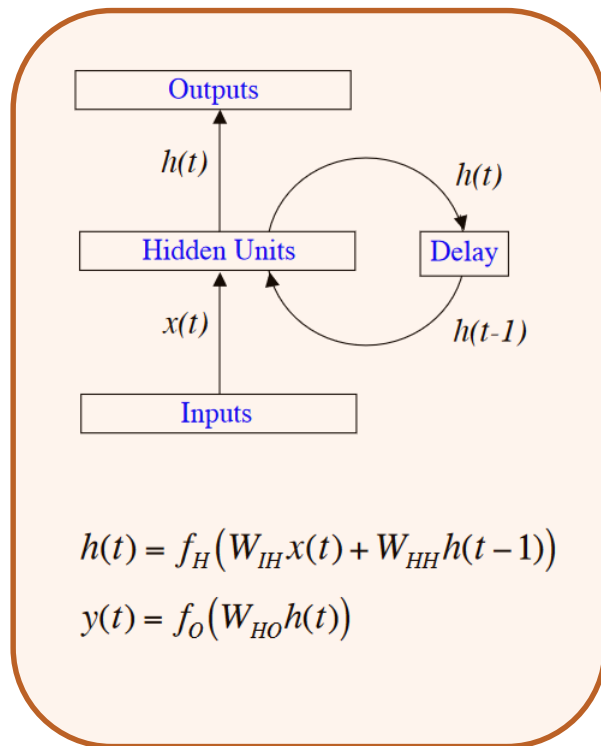
...

$$h_0 = \sigma(Wx_0)$$

آموزش شبکه ...

استفاده از الگوریتم پس‌انتشار ...

- محاسبه خطای خروجی و استفاده از گرادیان برای تخمین وزن‌ها
- تبدیل شبکه بازگشتی به شبکه جلوسو
- باز کردن در زمان (Unfolding over Time)





آموزش شبکه ...

○ استفاده از الگوریتم پس‌انتشار ...

- Backpropagation Through Time (BPTT)
- محاسبه خطای شبکه برای همه نمونه‌های بین دو زمان شروع t_0 و پایان t_1

$$E_{total}(t_0, t_1) = \sum_{t=t_0}^{t_1} E_{sse/ce}(t)$$

- گرادیان وزن‌های شبکه برای بدست آوردن مقدار تغییرات وزن

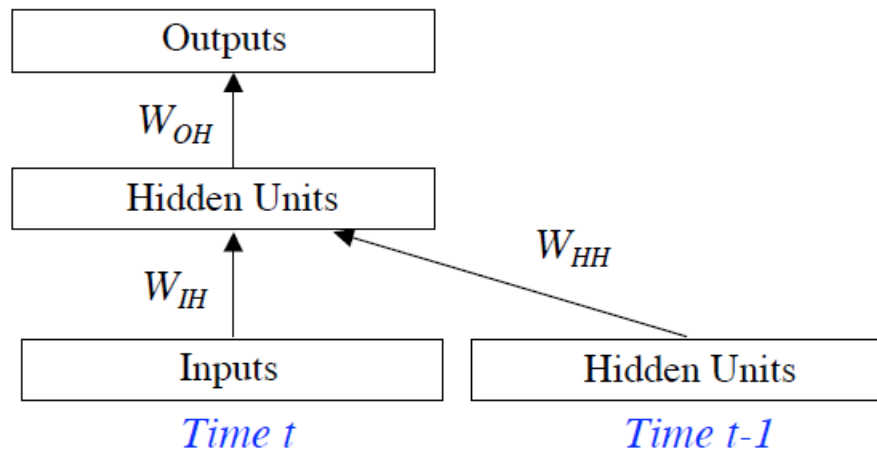
$$\Delta w_{ij} = -\eta \frac{\partial E_{total}(t_0, t_1)}{\partial w_{ij}} = -\eta \sum_{t=t_0}^{t_1} \frac{\partial E_{sse/ce}(t)}{\partial w_{ij}}$$

$$w_{ij} \in \{W_{IH}, W_{HH}\}$$

آموزش شبکه ...

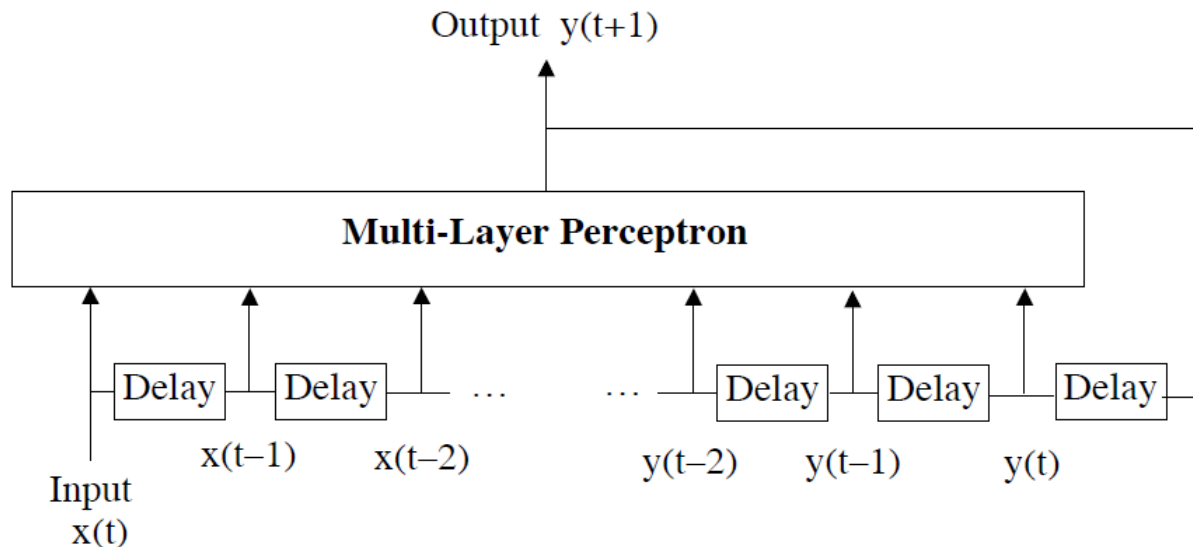
○ استفاده از الگوریتم پس‌انتشار

- استفاده از این روش نیازمند نگهداری حالت‌های قبلی شبکه (خروجی لایه مخفی) و کلیه ورودی‌های قبلی
- در عمل نگهداری همه اطلاعات قبلی مشکل است و تنها از تعداد محدودی از آنها (مثلاً ۳۰ مقدار قبلی) استفاده می‌شود = truncation
- حالت ساده = نگهداری فقط یک مرحله قبل = شبکه المان (Elman Network)



مدل NARX: Non-linear Auto-Regressive with eXogeneous inputs

- یک شبکه MLP با یک ورودی و یک خروجی
- ورودی‌ها و خروجی‌ها در زمان‌های مختلف تاخیر داده می‌شوند



- این ساختار برای پیش‌بینی سری‌های زمانی مناسب است

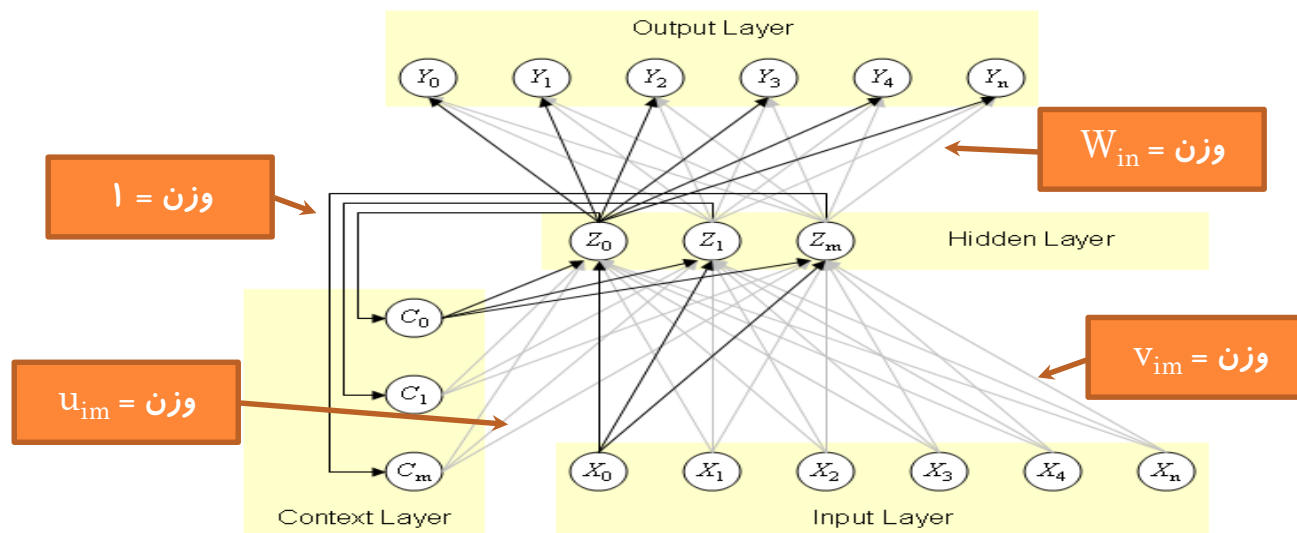
شبکه عصبی المان ...

○ ساختار

• دارای چهار لایه ورودی، مخفی، بافت و خروجی

• لایه بافت

- نرون‌های لایه بافت یک کپی از فعال‌سازهای نرون‌های لایه پنهان را دریافت می‌کنند
- اتصالات بازگشتی لایه بافت به لایه پنهان، یک حافظه کوتاه‌مدت را برای شبکه ایجاد می‌کند
- تعداد نرون‌های لایه بافت با تعداد نرون‌های لایه پنهان برابر است
- وزن یال‌هایی که لایه پنهان را به لایه بافت متصل می‌کنند برابر مقدار ثابت یک می‌باشد





شبکه عصبی المان: آموزش ...

○ به ابتدا و انتهای هر دنباله، نماد آغازین و نماد پایانی اضافه کنید

○ الگوریتم آموزش همان پس انتشار استاندارد است

- مرحله ۰: برای هر دنباله آموزش $X = (x_1, x_i, \dots, x_k)$ که x_1 و x_k به ترتیب نمادهای مربوط به شروع و خاتمه دنباله می‌باشد، مراحل ۱ تا ۱۴ را انجام دهید
- مرحله ۱: مقدار فعال‌ساز تمامی نرون‌های لایه بافت (C_i) را برابر ۰.۵ قرار دهید.
- مرحله ۲: مراحل ۳ تا ۱۴ را انجام دهید تا به انتهای دنباله برسید
- مرحله ۳: بردار x_i را به شبکه ارائه دهید.
- مرحله ۴: بردار x_{i+1} در دنباله را به عنوان پاسخ هدف به واحدهای خروجی ارائه دهید.
- مرحله ۵: مقدار فعال‌ساز نرون‌های لایه مخفی را محاسبه کنید

$$z_{in_q} = \sum_{i=1}^n x_i v_{iq} + \sum_{i=1}^m C_i u_{iq} \quad , \quad Z_q = f(z_{in_q})$$



شبکه عصبی المان: آموزش ...

- مرحله ۶: مقدار فعال‌ساز نرون‌های لایه خروجی را محاسبه کنید

$$y_{in_j} = \sum_{i=1}^m Z_i w_{ij} \quad , \quad y_j = f(y_{in_j})$$

- مرحله ۷: دلتای نرون‌های لایه خروجی را محاسبه کنید

$$\delta_j = (t_j - y_j) f'(y_{in_j})$$

- مرحله ۸: تغییرات وزن نرون‌های لایه پنهان به لایه خروجی را محاسبه کنید

$$\Delta W_{ij} = \alpha \delta_j Z_i$$

- مرحله ۹: خطای ورودی به نرون‌های لایه مخفی را محاسبه کنید (پس انتشار خطا)

$$\delta_{in_q} = \sum_{i=1}^n \delta_i w_{qi}$$

- مرحله ۱۰: دلتای نرون‌های لایه پنهان را محاسبه کنید

$$\delta_q = \delta_{in_q} f'(z_{in_q})$$

- مرحله ۱۱: تغییرات وزن نرون‌های لایه ورودی به لایه پنهان را محاسبه کنید

$$\Delta V_{iq} = \alpha x_i \delta_q$$



شبکه عصبی المان: آموزش

- مرحله ۱۲: تغییرات وزن نرون‌های لایه بافت به لایه پنهان را محاسبه کنید

$$\Delta U_{iq} = \alpha C_i \delta_q$$

- مرحله ۱۳: کلیه وزن‌ها را بروز رسانی کنید

- مرحله ۱۴: شرط توقف را بررسی کنید:

- اگر هدف نماد پایانی باشد، الگوریتم را خاتمه دهید، در غیر این صورت فعال‌ساز نرون‌های لایه مخفی را در نرون‌های لایه بافت کپی کنید.

شبکه عصبی المان: (مثال) پیش‌بینی دنباله گرامر...

گرامر ربر

سال ۱۹۶۷ •

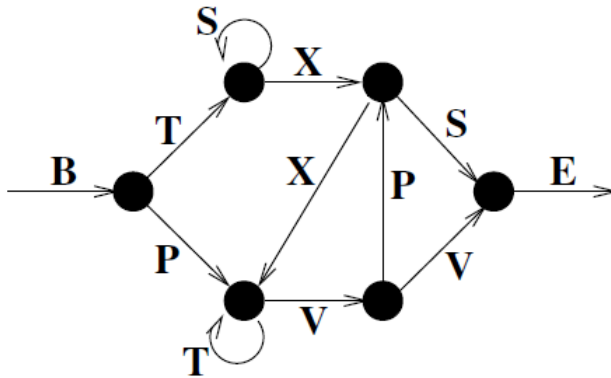
تولید دنباله‌های با طول‌های مختلف (حتی نامحدود) •

شروع با B و خاتمه با E •

BTSSXXTVVE •

BPVVE •

BPVPXVPXVPXVVE •



هدف: ارائه هر نماد دنباله به شبکه و پیش‌بینی نماد بعدی توسط شبکه •

دنباله آموزش: (B, T, X, S, E) •

نماد آغازین: B •

نماد پایانی: E •

سایر نمادها: S, T, X, V, P •

شبکه عصبی المان: (مثال) پیش‌بینی دنباله گرامر رپر...

بردار B و E هر دو اولین نرون در نظر گرفته شدند و بردار معادل برای هر دو $(1,0,0,0,0,0)$ است

○ ساختار

- تعداد نرون‌های لایه ورودی : ۶

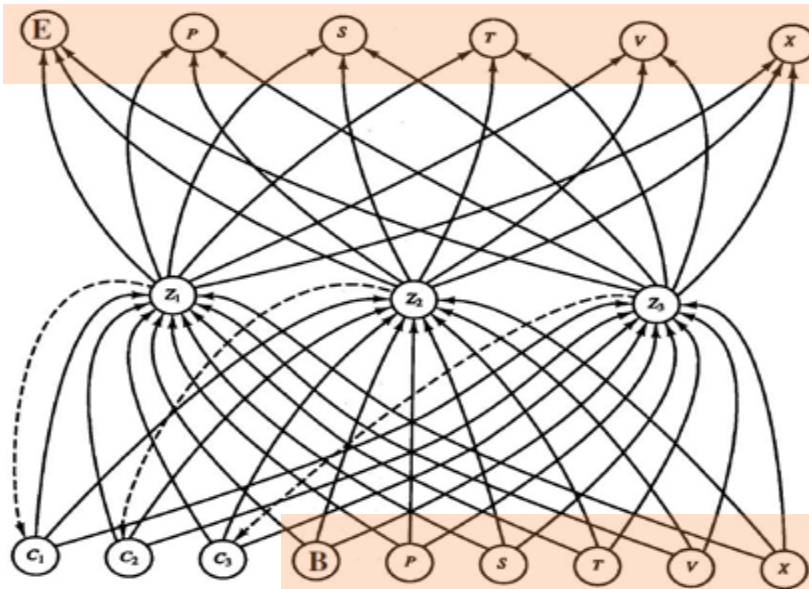
○ مربوط به ۵ نماد (P,S,T,V,X) و نماد آغازین B (فقط در ابتدا می آید)

- تعداد نرون‌های لایه خروجی: ۶

○ مربوط به ۵ نماد (P,S,T,V,X) و نماد پایانی E (فقط در پایان می آید)

- نرون اول معادل B ، نرون دوم معادل P ، ... است.

○ مثال: برای تولید B مقدار نرون اول یک و مقدار سایر نرون‌های ورودی را برابر صفر قرار می‌دهیم



$$B=(1,0,0,0,0,0)$$

...

$$T=(0,0,0,1,0,0)$$



شبکه عصبی المان: (مثال) پیش‌بینی دنباله گرامر ربر...

آموزش ...

- مرحله ۰: دنباله آموزش (B, T, X, S, E) را در نظر بگیرید
- مرحله ۱: وزن‌های اولیه را به صورت تصادفی انتخاب کنید

لایه مخفی

-0.4674	-0.0393	-0.3091
0.0612	0.4816	-0.0717
0.3819	-0.3436	-0.0180
0.1692	0.3555	-0.3794
-0.3096	0.1448	0.0895
-0.1311	-0.1237	-0.2738

لایه خروجی

-0.1154	-0.2096	0.3244	-0.1561	0.4063	-0.2393
0.0830	0.1171	0.4827	0.0841	0.3797	0.0944
-0.2482	-0.2347	0.2302	-0.3922	0.3178	-0.4775

لایه مخفی

-0.0747	-0.3212	0.0985
-0.1873	-0.0771	-0.0291
-0.3385	-0.4058	0.1959

وزن یال‌های اتصالی از لایه ورودی به لایه مخفی V

۶ نرون ورودی و ۳ نرون مخفی

وزن یال‌های اتصالی از لایه مخفی به لایه خروجی W

۳ نرون مخفی و ۶ نرون خروجی

وزن یال‌های اتصالی از لایه بافت به لایه مخفی U

۳ نرون مخفی و ۳ نرون معادل در لایه بافت



شبکه عصبی المان: (مثال) پیش‌بینی دنباله گرامر...

آموزش ...

- مرحله ۲: آموزش را برای این دنباله (B, T, X, S, E) آغاز کنید
- مرحله ۳: نماد $B=(1,0,0,0,0,0)$ را به شبکه ارائه دهید (ورودی)
- مرحله ۴: بردار هدف را برابر $T=(0,0,0,1,0,0)$ قرار دهید (نماد بعدی در دنباله)
- مرحله ۵: مقدار فعال‌ساز نرون‌های لایه مخفی را محاسبه کنید

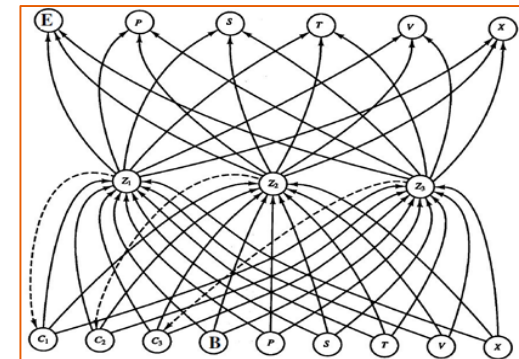
$$z_in_q = \sum_{i=1}^n x_i v_{iq} + \sum_{i=1}^m C_i u_{iq}, \quad Z_q = f(z_in_q)$$

$$z_in_1 = ((-0.4674 * 1) + (0.612 * 0) + (0.3819 * 0) + (0.1692 * 0) + (-0.3096 * 0) + (-0.1311 * 0)) + ((-0.0747 * 0.5) + (-0.1873 * 0.5) + (-0.3385 * 0.5)) = -0.7677$$

$$z_in = \begin{bmatrix} -0.7677 & -0.4413 & -0.1764 \end{bmatrix}$$

$$Z = \begin{bmatrix} 0.3170 & 0.3914 & 0.4560 \end{bmatrix}$$

$$Z_1 = \text{Sigmoid}(-0.7677) = 0.3170$$



شبکه عصبی المان: (مثال) پیش‌بینی دنباله گرامر رپر...

- مرحله ۶: مقدار فعال‌ساز نرون‌های لایه خروجی را محاسبه کنید

$$y_{in_j} = \sum_{i=1}^m Z_i w_{ij}, \quad y_j = f(y_{in_j})$$

$$((-0.1154 * 0.3170) + (0.0830 * 0.3914) + (-0.2482 * 0.456)) = -0.1173$$

$y_{in} =$	-0.1173	-0.1276	0.3967	-0.1954	0.4223	-0.2567
$Y =$	0.4707	0.4681	0.5979	0.4513	0.6040	0.4362

$$\text{Sigmoid}(-0.1173) = 0.4707$$

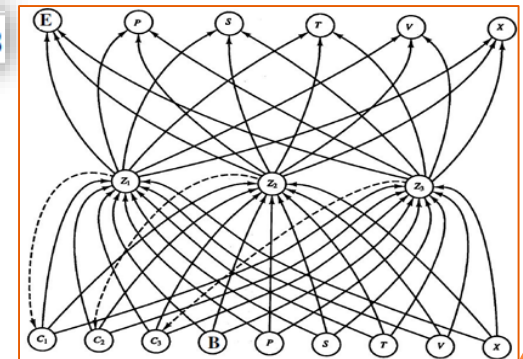
- مرحله ۷: دلتای نرون‌های لایه خروجی را محاسبه کنید

$$\delta_j = (t_j - y_j) f'(y_{in_j})$$

$$\delta_j =$$

-0.1173	-0.1166	-0.1437	0.1359	-0.1445	-0.1073
---------	---------	---------	--------	---------	---------

$$((0 - 0.4707) * (0.4707 * (1 - 0.4707))) = -0.1173$$





شبکه عصبی المان: (مثال) پیش‌بینی دنباله گرامر رپر...

مرحله ۸: تغییرات وزن نرون‌های لایه پنهان به لایه خروجی را محاسبه کنید

$$\Delta W_{ij} = \alpha \delta_j Z_i$$

لایه خروجی

$\Delta W_{11} = 0.1 * (-0.1173) * 0.3170 = -0.0037$

$\Delta W =$

-0.0037	-0.0037	-0.0046	0.0043	-0.0046	-0.0034
-0.0046	-0.0046	-0.0056	0.0053	-0.0057	-0.0042
-0.0053	-0.0053	-0.0066	0.0062	-0.0066	-0.0049

• مرحله ۹: خطای ورودی به نرون‌های لایه مخفی را محاسبه کنید (پس انتشار خطا)

$$\delta_{in_q} = \sum_{i=1}^n \delta_i w_{qi}$$

$\delta_{in} =$

-0.0629	-0.1463	-0.0246
---------	---------	---------

$$(-0.1173 * (-0.1154)) + (-0.1166 * (-0.2096)) + (-0.1437 * 0.3244) + (0.1359 * (-0.1561)) + (-0.1445 * 0.4063) + (-0.1073 * (-0.2393)) = -0.0629$$

• مرحله ۱۰: دلتای نرون‌های لایه پنهان را محاسبه کنید

$$\delta_q = \delta_{in_q} f'(z_{in_q})$$

$((-0.0629) * (0.3170 * (1 - 0.3170))) = -0.0136$

$\delta_m =$

-0.0136	-0.0349	-0.0061
---------	---------	---------

شبکه عصبی المان: (مثال) پیش‌بینی دنباله گرامر ربر...

○ آموزش ...

$$\Delta V_{iq} = \alpha x_i \delta_q$$

- مرحله ۱: تغییرات وزن نرون‌های لایه ورودی به لایه پنهان را محاسبه کنید

لایه مخفی

$\Delta V_{11} = 0.1 * 1 * -0.0136 = -0.0014$

$\Delta V =$

-0.0014	-0.0035	-6.1056e-04
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

لایه ورودی

- مرحله ۲: تغییرات وزن نرون‌های لایه بافت به لایه پنهان را محاسبه کنید

$$\Delta U_{iq} = \alpha C_i \delta_q$$

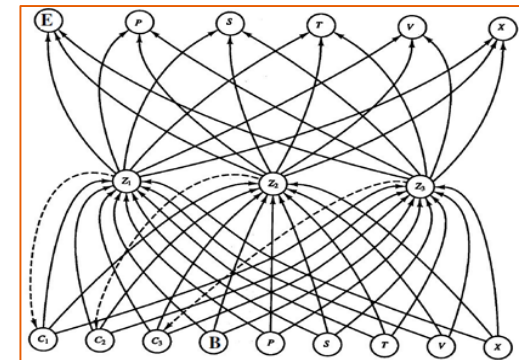
لایه مخفی

$\Delta U =$

-6.8109e-04	-0.0017	-3.0528e-04
-6.8109e-04	-0.0017	-3.0528e-04
-6.8109e-04	-0.0017	-3.0528e-04

لایه بافت

$$\Delta U_{11} = 0.1 * 0.5 * (-0.0136) = -0.00068$$





شبکه عصبی المان: (مثال) پیش‌بینی دنباله گرامر...

مرحله ۱۳: کلیه وزن‌ها را بروز رسانی کنید

$$W_{11} = -0.1154 + (-0.0037) = -0.1191$$

لایه خروجی

لایه مخفی

-0.1191	-0.2133	0.3198	-0.1518	0.4017	-0.2427
0.0784	0.1125	0.4770	0.0894	0.3740	0.0902
-0.2535	-0.2400	0.2237	-0.3860	0.3112	-0.4824

$$W_{new} = W_{old} + \Delta W$$

$$V_{11} = -0.4674 + (-0.0014) = -0.4688$$

لایه مخفی

لایه ورودی

-0.4688	-0.0428	-0.3097
0.0612	0.4816	-0.0717
0.3819	-0.3436	-0.0180
0.1692	0.3555	-0.3794
-0.3096	0.1448	0.0895
-0.1311	-0.1237	-0.2738

$$V_{new} = V_{old} + \Delta V$$

$$U_{11} = -0.0747 + (-0.00068) = -0.0754$$

لایه مخفی

لایه بافت

-0.0754	-0.3230	0.0982
-0.1880	-0.0789	-0.0294
-0.3392	-0.4075	0.1956

$$U_{new} = U_{old} + \Delta U$$



شبکه عصبی المان: (مثال) پیش‌بینی دنباله گرامر...

○ آموزش ...

- مرحله ۱۴: شرط توقف را بررسی کنید:

- شرایط توقف برقرار نیست (مقدار هدف برابر با نماد پایانی نیست)
- فعال‌سازهای نرون‌های لایه مخفی را در لایه بافت کپی کنید

$$C = Z = \begin{array}{|c|c|c|} \hline 0.3170 & 0.3914 & 0.4560 \\ \hline \end{array}$$

ادامه آموزش با ارائه نمونه بعدی از دنباله
به عنوان ورودی

شبکه عصبی المان: (مثال) پیش‌بینی دنباله گرامر رپر...

آموزش ...

- مرحله ۲: آموزش برای دومین حرف رشته (B, T, X, S, E)
- مرحله ۳: نماد $T=(0,0,0,1,0,0)$ را به شبکه ارائه دهید (ورودی)
- مرحله ۴: بردار هدف را برابر $X=(0,0,0,0,0,1)$ قرار دهید
- مرحله ۵: مقدار فعال‌ساز نرون‌های لایه مخفی را محاسبه کنید

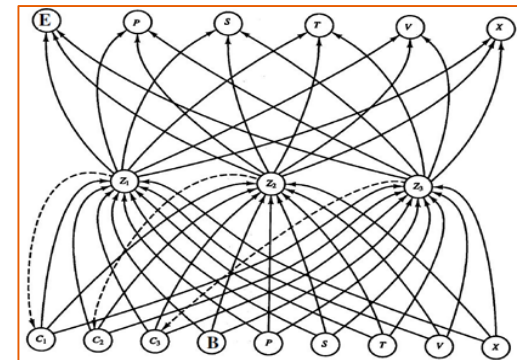
$$z_in_q = \sum_{i=1}^n x_i v_{iq} + \sum_{i=1}^m C_i u_{iq}, \quad Z_q = f(z_in_q)$$

$$((-0.4688 * 0) + (0.612 * 0) + (0.3819 * 0) + (0.1692 * 1) + (-0.3096 * 0) + (-0.1311 * 0)) + ((-0.0754 * 0.3170) + (-0.1880 * 0.3914) + (-0.3392 * 0.4560)) = -0.0830$$

$$z_in = \begin{bmatrix} -0.0830 & 0.0364 & -0.2705 \end{bmatrix}$$

$$Z = \begin{bmatrix} 0.4793 & 0.5091 & 0.4328 \end{bmatrix}$$

$$Z_1 = \text{Sigmoid}(-0.0830) = 0.4793$$





شبکه عصبی المان: (مثال) پیش‌بینی دنباله گرامر رپر...

- مرحله ۶: مقدار فعال‌ساز نرون‌های لایه خروجی را محاسبه کنید

$$y_in_j = \sum_{i=1}^m Z_i w_{ij}, \quad y_j = f(y_in_j)$$

$$((-0.1191 * 0.4793) + (0.0784 * 0.5091) + (-0.2535 * 0.4328)) = -0.1269$$

$$y_in = \begin{bmatrix} -0.1269 & -0.1488 & 0.4930 & -0.1943 & 0.5176 & -0.2792 \end{bmatrix}$$

$$Y = \begin{bmatrix} 0.4683 & 0.4629 & 0.6208 & 0.4516 & 0.6266 & 0.4307 \end{bmatrix}$$

$$\text{Sigmoid}(-0.1269) = 0.4683$$

- مرحله ۷: دلتای نرون‌های لایه خروجی را محاسبه کنید

$$\delta_j = (t_j - y_j) f'(y_in_j)$$

$$\delta_j = \begin{bmatrix} -0.1166 & -0.1151 & -0.1461 & -0.1118 & -0.1466 & 0.1396 \end{bmatrix}$$

$$((0 - 0.4683) * (0.4683 * (1 - 0.4683))) = -0.1166$$



شبکه عصبی المان: (مثال) پیش‌بینی دنباله گرامر رپر...

- مرحله ۸: تغییرات وزن نرون‌های لایه پنهان به لایه خروجی را محاسبه کنید

$$\Delta W_{ij} = \alpha \delta_j Z_i$$

$$\Delta W_{11} = 0.1 * (-0.1166) * 0.4793 = -0.0056$$

لایه خروجی

$\Delta W =$

-0.0056	-0.0055	-0.0070	-0.0054	-0.0070	0.0067
-0.0059	-0.0059	-0.0074	-0.0057	-0.0075	0.0071
-0.0050	-0.0050	-0.0063	-0.0048	-0.0063	0.0060

- مرحله ۹: خطای ورودی به نرون‌های لایه مخفی را محاسبه کنید (پس انتشار خطا)

$$\delta_{in_q} = \sum_{i=1}^n \delta_i w_{qi}$$

$$\delta_{in} =$$

-0.0841	-0.1440	-0.0453
---------	---------	---------

$$\delta_{in_1} = (-0.1173 * (-0.1154)) + (-0.1166 * (-0.2096)) + (-0.1437 * 0.3244) + (0.1359$$

$$\delta_q = \delta_{in_q} f'(z_{in_q})$$

- مرحله ۱۰: دلتای نرون‌های لایه پنهان را محاسبه کنید

$$\delta_m =$$

-0.0210	-0.0360	-0.0111
---------	---------	---------

$$\delta_1 = ((-0.0841) * (0.4793 * (1 - 0.4793))) = -0.0210$$

شبکه عصبی المان: (مثال) پیش‌بینی دنباله گرامر...

○ آموزش ...

$$\Delta V_{iq} = \alpha x_i \delta_q$$

- مرحله ۱۱: تغییرات وزن نرون‌های لایه ورودی به لایه پنهان را محاسبه کنید

لایه مخفی

$\Delta V_{11} = 0.1 * 0 * (-0.0210) = 0$

$\Delta V =$

0	0	0
0	0	0
0	0	0
-0.0021	-0.0036	-0.0011
0	0	0
0	0	0

لایه ورودی

$$\Delta U_{iq} = \alpha C_i \delta_q$$

- مرحله ۱۲: تغییرات وزن نرون‌های لایه بافت به لایه پنهان را محاسبه کنید

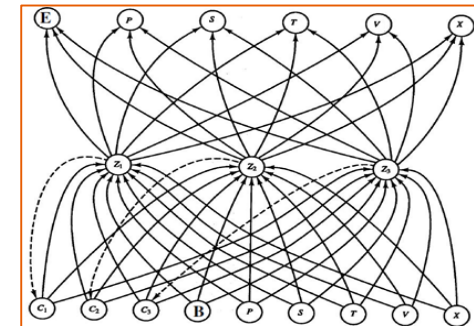
لایه مخفی

$\Delta U =$

-6.6534e-04	-0.0011	-3.5241e-04
-8.2159e-04	-0.0014	-4.3517e-04
-9.5718e-04	-0.0016	-5.0699e-04

لایه بافت

$$\Delta U_{11} = 0.1 * 0.3170 * (-0.0210) = -0.00066$$





شبکه عصبی المان: (مثال) پیش‌بینی دنباله گرامر رپر...

○ آموزش ...

$$W_{11} = -0.1191 + (-0.0056) = -0.1247$$

• مرحله ۱۳: کلیه وزن‌ها را بروز رسانی کنید لایه خروجی

$W =$

لایه مخفی

-0.1247	-0.2188	0.3128	-0.1572	0.3947	-0.2360
0.0725	0.1067	0.4696	0.0837	0.3665	0.0973
-0.2586	-0.2450	0.2174	-0.3909	0.3048	-0.4763

$$W_{new} = W_{old} + \Delta W$$

$$V_{11} = -0.4688 + 0 = -0.4688$$

لایه مخفی

$V =$

لایه ورودی

-0.4688	-0.0428	-0.3097
0.0612	0.4816	-0.0717
0.3819	-0.3436	-0.0180
0.1671	0.3519	-0.3805
-0.3096	0.1448	0.0895
-0.1311	-0.1237	-0.2738

$$V_{new} = V_{old} + \Delta V$$

$$U_{11} = -0.0754 + (-0.00066) = -0.0761$$

لایه بافت

$U =$

لایه مخفی

-0.0761	-0.3241	0.0979
-0.1888	-0.0803	-0.0298
-0.3402	-0.4092	0.1951

$$U_{new} = U_{old} + \Delta U$$



شبکه عصبی المان: (مثال) پیش‌بینی دنباله گرامر رپر...

○ آموزش ...

- مرحله ۱۴: شرط توقف را بررسی کنید:

- شرایط توقف برقرار نیست (مقدار هدف برابر با نماد پایانی نیست)
- فعال‌سازهای نرون‌های لایه مخفی را در لایه بافت کپی کنید

$$C = Z = \begin{array}{|c|c|c|} \hline 0.4793 & 0.5091 & 0.4328 \\ \hline \end{array}$$

ادامه آموزش با ارائه نمونه بعدی از دنباله
به عنوان ورودی



شبکه عصبی المان: (مثال) پیش‌بینی دنباله گرامر رپر...

○ آموزش

- مرحله ۲: آموزش برای سومین حرف رشته (B, T, X, S, E)

○ مرحله ۳: نماد $X=(0,0,0,0,0,1)$ را به شبکه ارائه دهید

○ مرحله ۴: بردار هدف را برابر $S=(0,0,1,0,0,0)$ قرار دهید

○ مرحله ۵ و ۶: مطابق الگوریتم آموزش پاسخ شبکه را محاسبه کنید.

○ مرحله ۷ تا ۱۳: مقدار خطا را محاسبه کرده، پس انتشار دهید و وزن‌ها را بروز کنید.

○ مرحله ۱۴: فعال‌سازهای نرون‌های لایه مخفی را در لایه بافت کپی کنید

- مرحله ۲: آموزش برای چهارمین حرف رشته (B, T, X, S, E)

○ مرحله ۳: نماد $S=(0,0,1,0,0,0)$ را به شبکه ارائه دهید

○ مرحله ۴: بردار هدف را برابر $E=(1,0,0,0,0,0)$ قرار دهید

○ مرحله ۵ و ۶: مطابق الگوریتم آموزش پاسخ شبکه را محاسبه کنید.

○ مرحله ۷ تا ۱۳: مقدار خطا را محاسبه کرده، پس انتشار دهید و وزن‌ها را بروز کنید.

○ مرحله ۱۴: بردار هدف برابر نماد پایانی است، آموزش برای این رشته کامل شده است

- ادامه الگوریتم در صورت وجود رشته دیگری در داده‌های آموزش



شبکه عصبی المان: الگوریتم کاربرد

○ الگوریتم کاربرد (آزمون)

○ مرحله ۰: برای هر دنباله آزمون $X = (x_1, x_i, \dots, x_k)$ مراحل ۱ تا ۱۴ را انجام دهید

○ مرحله ۱: مقدار فعال‌ساز تمامی نرون‌های لایه بافت را برابر ۰.۵ قرار دهید.

○ مرحله ۲: برای تمامی بردارهای x_i مراحل ۳ تا ۶ را انجام دهید

○ مرحله ۳: بردار x_i را به شبکه ارائه دهید.

○ مرحله ۴: مقدار فعال‌ساز نرون‌های لایه مخفی را محاسبه کنید

$$z_in_m = \sum_{i=1}^n x_i v_{i m} + \sum_{i=1}^m C_i u_{i m} \quad , \quad Z_m = f(z_in_m)$$

○ مرحله ۵: مقدار فعال‌ساز نرون‌های لایه خروجی را محاسبه کنید

$$y_in_n = \sum_{i=1}^m Z_i w_{i n} \quad , \quad Y_n = f(y_in_n)$$

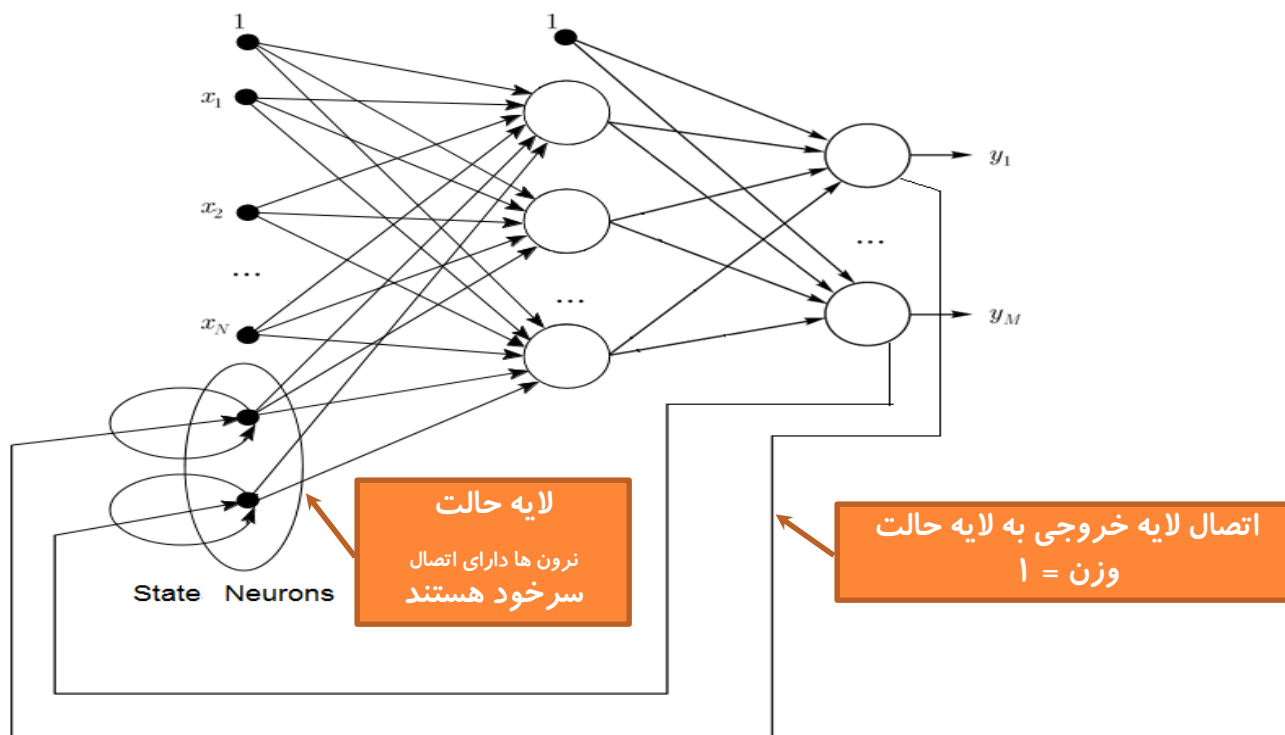
• مرحله ۶: نماد با بیشترین مقدار فعال‌ساز را به عنوان خروجی شبکه اعلام کنید

شبکه عصبی جردن ...



معرفی و ساختار

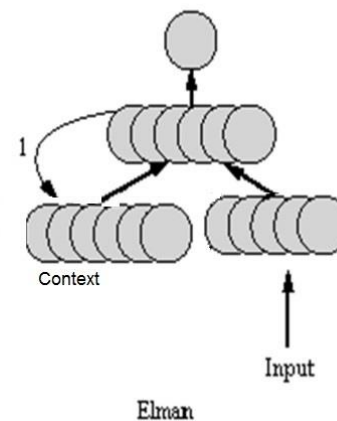
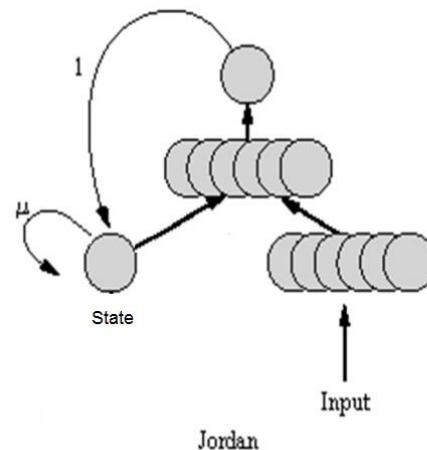
- ارائه شده در سال ۱۹۹۶ توسط مایکل جردن
- دارای شباهت بسیار زیاد به شبکه عصبی المان
- شبکه دارای اتصالات بازگشتی از لایه خروجی به لایه حالت و همچنین از لایه حالت به خودش می‌باشد



شبکه عصبی جردن: تفاوت با شبکه المان

○ در شبکه جردن

- اتصالات بازگشتی به جای لایه پنهان از لایه خروجی شروع می‌شود (با وزن ثابت یک)
 - در این شبکه لایه بافت، لایه حالت (State Layer) نامیده می‌شود
 - لایه حالت شامل اتصالات بازگشتی از خودش به خودش با وزن ثابت می‌باشد
 - تعداد نرون‌های لایه حالت با تعداد نرون‌های لایه خروجی برابر است
- نرون‌های لایه حالت یک کپی از فعال‌سازهای نرون‌های لایه خروجی را دریافت می‌کنند.

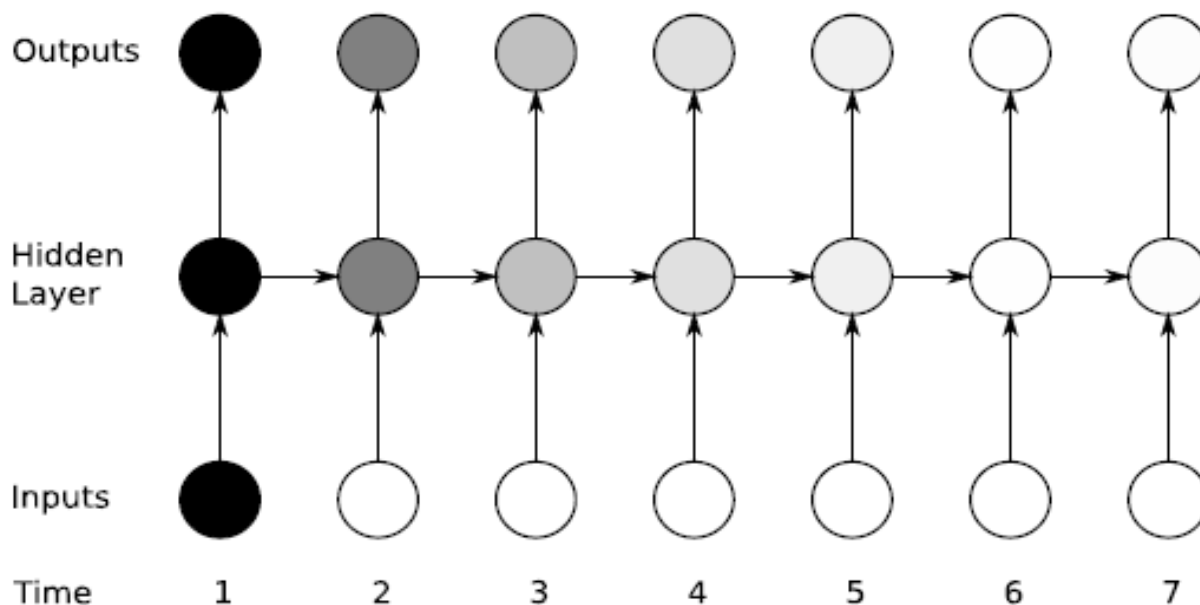


مشکل فراموشی در شبکه‌های عصبی بازگشتی ...

○ مشکل

- با طولانی شدن دنباله ورودی، شبکه عصبی بازگشتی به مرور داده‌های اولیه را فراموش می‌کند که به آن مشکل فراموشی گفته می‌شود

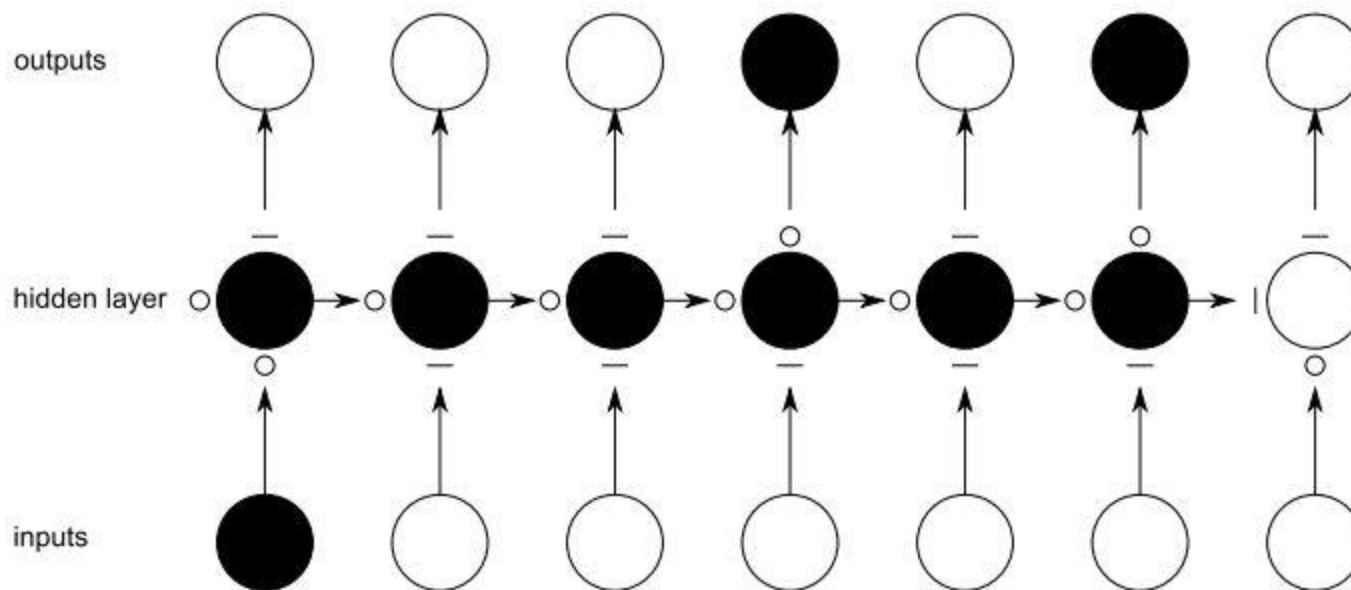
○ سایه‌های پررنگ‌تر به معنای تاثیر بیشتر بر لایه پنهان و خروجی می‌باشد



مشکل فراموشی در شبکه‌های عصبی بازگشتی

○ حل مشکل فراموشی

- کنترل ورود و خروج داده در واحدهای لایه میانی شبکه

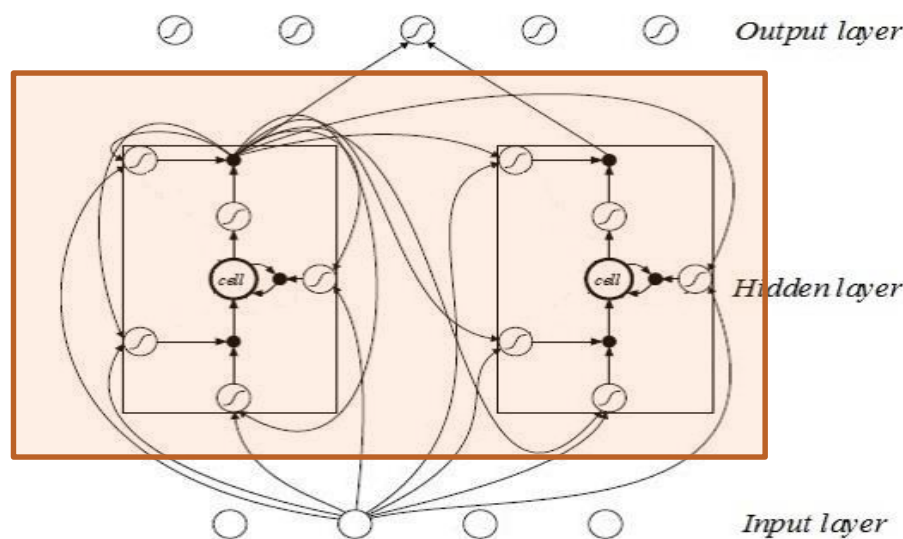
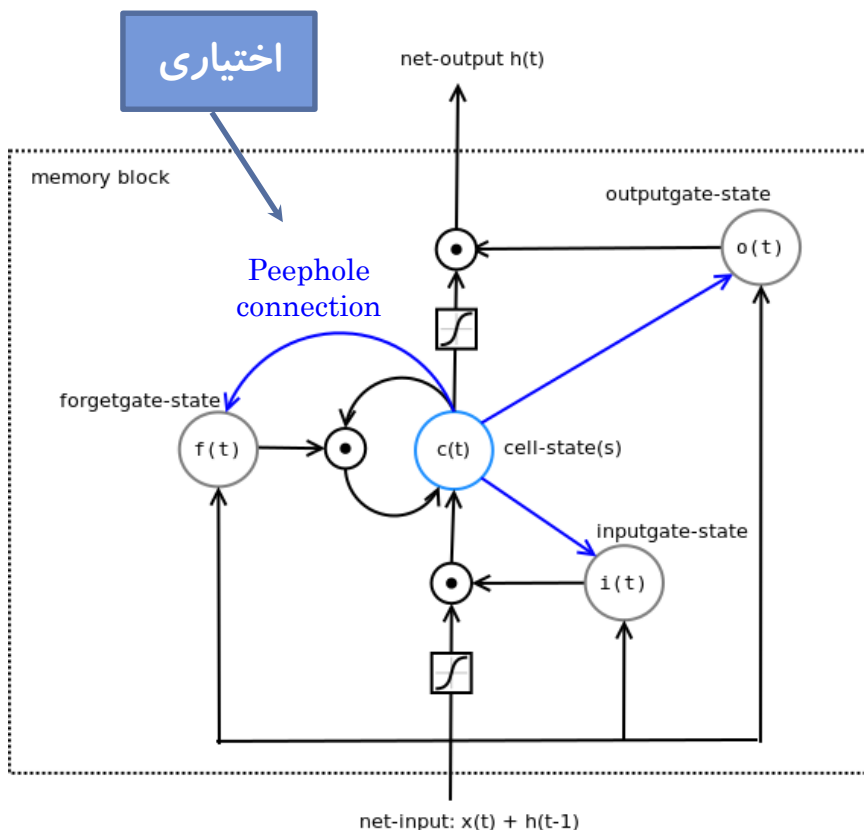


شبکه عصبی حافظه کوتاه مدت ماندگار (LSTM) ...

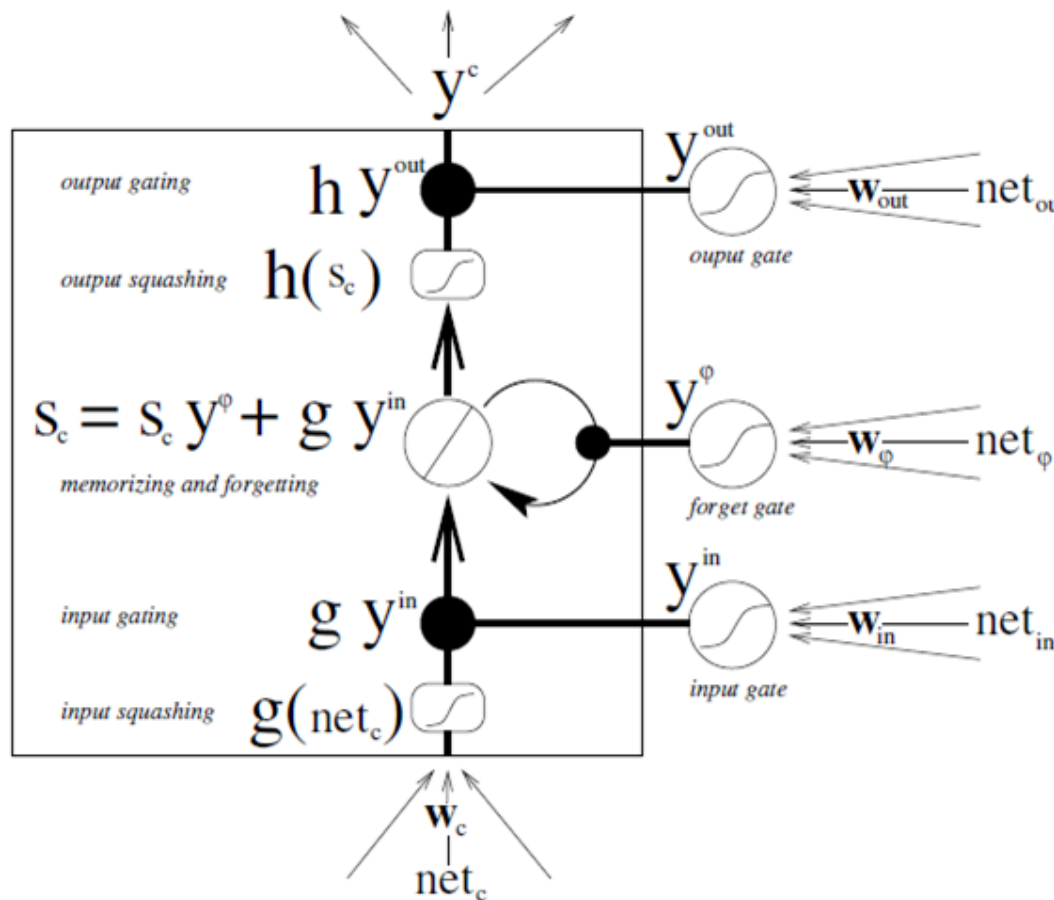
○ شبکه Long Short Term Memory (LSTM)

• نرون‌های لایه پنهان با بلوک‌های حافظه جایگزین شدند

○ حل شدن مشکل فراموشی دنباله‌های طولانی



شبکه عصبی LSTM: ساختار بلوک حافظه ...



○ هر بلوک حافظه، شامل

• سلول

○ برای ذخیره اطلاعات در بلوک

• دروازه ورودی

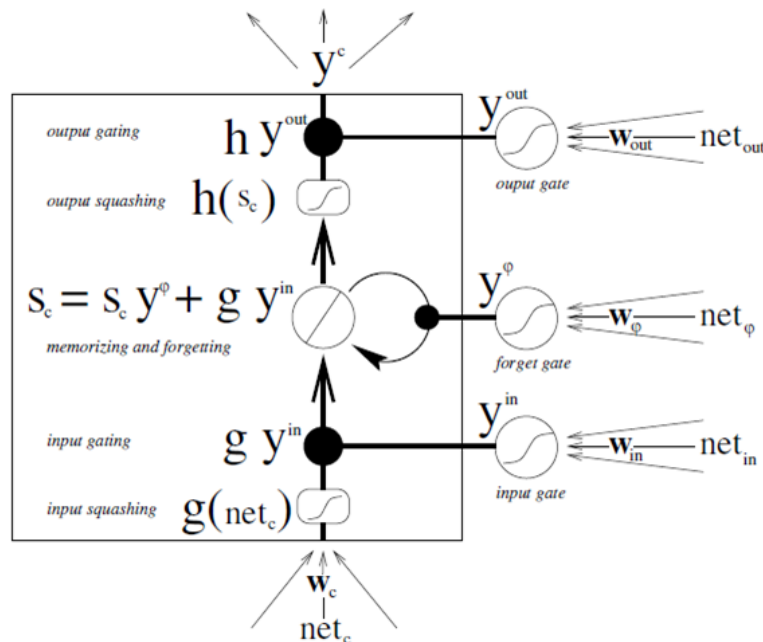
• دروازه فراموشی

• دروازه خروجی

○ دروازه‌ها وظیفه کنترل ورود و خروج داده‌ها و پاک کردن حافظه بلوک را برعهده دارند

شبکه عصبی LSTM: ساختار بلوک حافظه ...

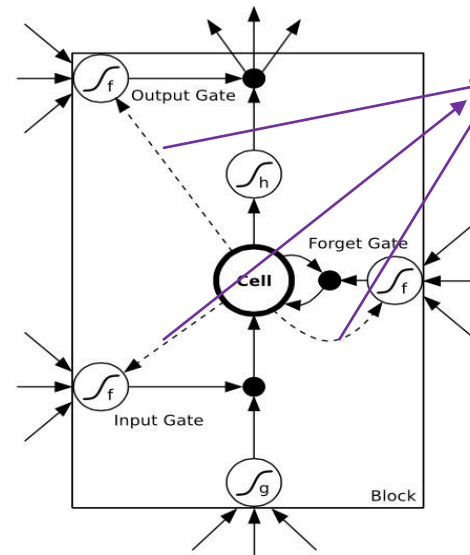
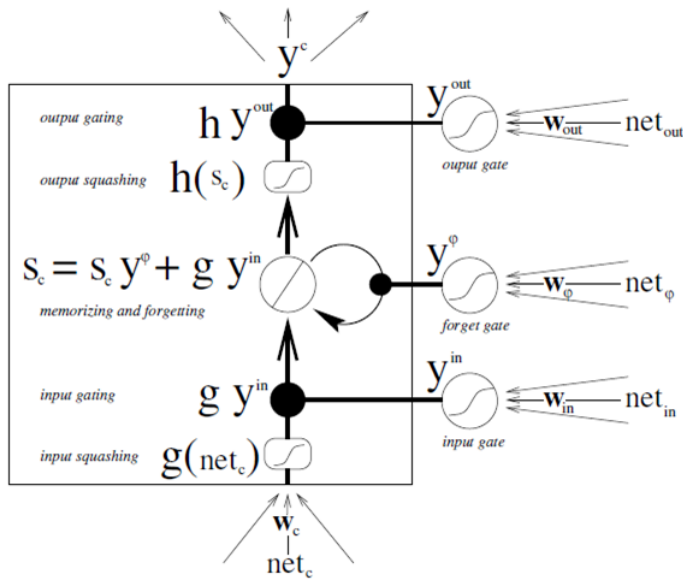
- وظیفه دروازه‌ها: کنترل ورود و خروج داده‌ها و پاک کردن حافظه‌ی بلوک
 - فعال‌ساز دروازه‌ها مقداری بین صفر و یک می‌گیرند.
 - در صورتی که دروازه کاملاً باز باشد فعال‌ساز آن برابر یک و در صورتی که کاملاً بسته باشد فعال‌ساز آن برابر صفر است
- هر سلول حافظه در مرکز خود یک واحد به نام CEC دارد که به فعال‌سازی آن **حالت سلول**، S_c می‌گویند



شبکه عصبی LSTM: ساختار بلوک حافظه

○ اتصالات روزنه (peephole)

- این اتصالات دلخواه هستند (optional) و حالت سلول، S_c ، را به دروازه‌ها متصل می‌سازند
- در این اسلایدها از این اتصالات صرف نظر شده است



Peephole connections

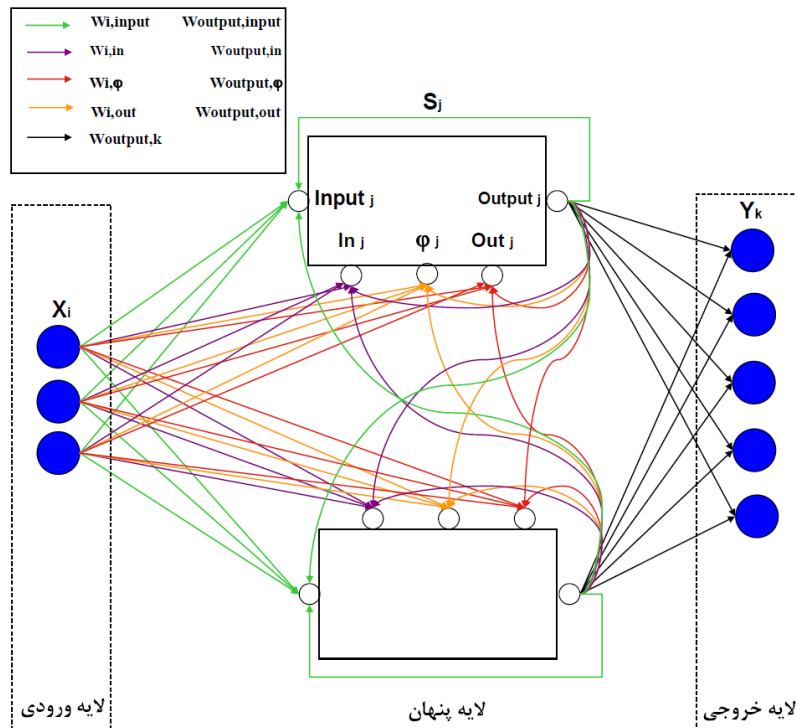
شبکه عصبی LSTM: اتصالات (وزن‌ها)

از ورودی

- وزن بین لایه ورودی و بلوک حافظه
- وزن بین لایه ورودی و دروازه ورودی
- وزن بین لایه ورودی و دروازه فراموشی
- وزن بین لایه ورودی و دروازه خروجی

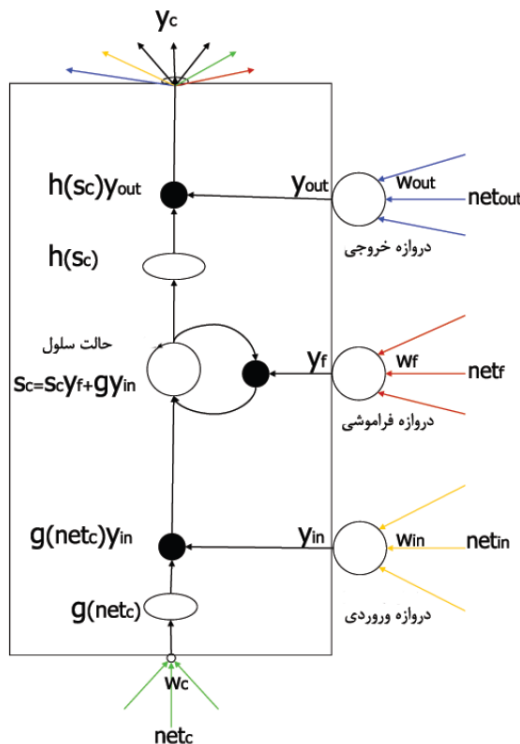
از خروجی بلوک (سلول حافظه)

- وزن بین خروجی بلوک‌ها و ورودی بلوک‌ها
- وزن بین خروجی بلوک‌ها و دروازه ورودی
- وزن بین خروجی بلوک‌ها و دروازه فراموشی
- وزن بین خروجی بلوک‌ها و دروازه خروجی
- وزن بین خروجی‌های بلوک‌ها و لایه خروجی



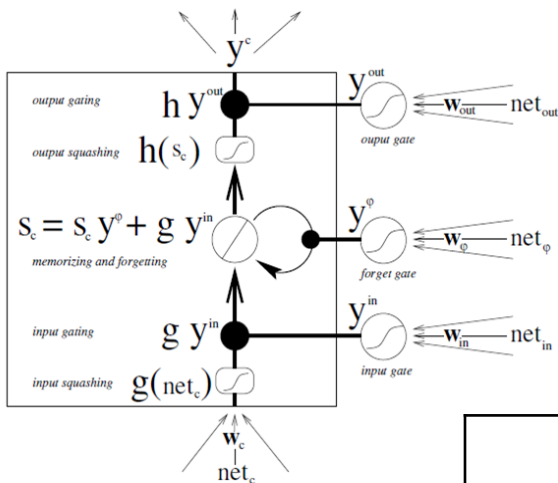
شبکه عصبی LSTM: آموزش ...

○ نمادها



$net_{c_j}(t)$	مقدار خالص ورودی به سلول j
$net_{in_j}(t)$	مقدار خالص ورودی به دروازه ورودی سلول j
$net_{\phi_j}(t)$	مقدار خالص ورودی به دروازه فراموشی سلول j
$net_{out_j}(t)$	مقدار خالص ورودی به دروازه خروجی سلول j
$s_{c_j}(t)$	حالت سلول j
$y^{in_j}(t)$	فعال ساز دروازه ورودی سلول j
$y^{out_j}(t)$	فعال ساز دروازه خروجی سلول j
$y^{\phi_j}(t)$	فعال ساز دروازه فراموشی سلول j
$y^{c_j}(t)$	فعال ساز خروجی سلول j
$net_k(t)$	مقدار خالص ورودی به نرون خروجی شماره k
$y^k(t)$	فعال ساز نرون خروجی شماره k
w_{lm}	یال اتصالی از واحد m به واحد l (اندیس کلی وزن‌ها)
$\delta_{out_j}(t)$	خطای دروازه خروجی سلول j
$\delta_k(t)$	خطای نرون خروجی شماره k
$e_{s_{c_j}}(t)$	خطای حالت سلول j
$0 \leq \alpha \leq 1$	نرخ یادگیری

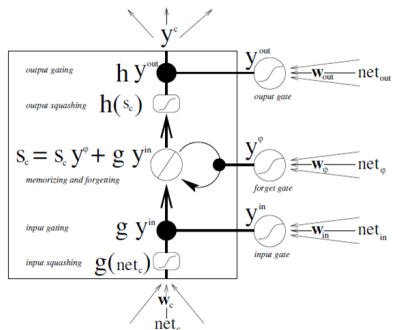
شبکه عصبی LSTM: آموزش ...



○ نمادها

$f(x) = \frac{1}{1 + e^{-x}}$	فعال ساز $f(x)$
$h(x) = \frac{2}{1 + e^{-x}} - 1$	فعال ساز $h(x)$
$g(x) = \frac{4}{1 + e^{-x}} - 2$	فعال ساز $g(x)$
به فعال ساز واحد m در گام زمانی قبل اشاره می‌کند. در صورتی که واحد m معادل نرون i از لایه ورودی باشد این مقدار معادل ورودی i م شبکه (یعنی x_i) در مرحله زمانی فعلی خواهد بود.	
	$y^m(t - 1)$

شبکه عصبی LSTM: آموزش ...

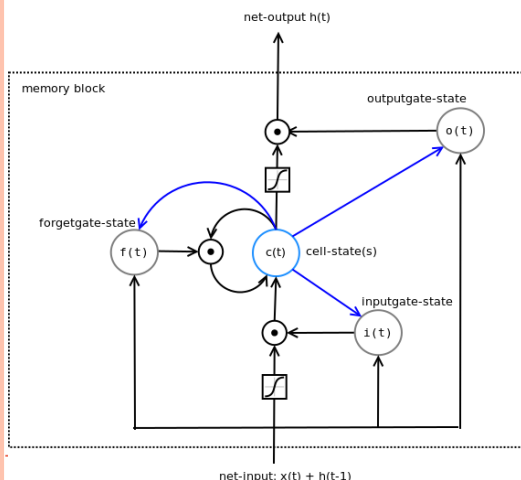


- برای هر دنباله آموزش $X^T = (x_1, x_2, \dots, x_{T-1}, x_T)$ مراحل ۱ تا ۱۵ را انجام دهید.
- مرحله ۰: انتخاب وزن‌های اولیه به صورت تصادفی
- مرحله ۱: مقدار اولیه حالت سلول‌ها، s_c ، را برابر صفر قرار دهید
- مرحله ۲: مقدار خالص همه ورودی‌های ممکن به دروازه‌های ورودی و فعال‌سازهای آن‌ها را محاسبه کنید

$$net_{in_j}(t) = \sum_m w_{in_j m} y^m(t-1), \quad y^{in_j}(t) = f(net_{in_j}(t))$$

فعال سازی بین صفر و یک

- net_{in_j} معادل جمع وزندار کلیه ورودی‌های ممکن به دروازه ورودی j آمین بلوک حافظه می‌باشد که این ورودی‌ها عبارتند از:



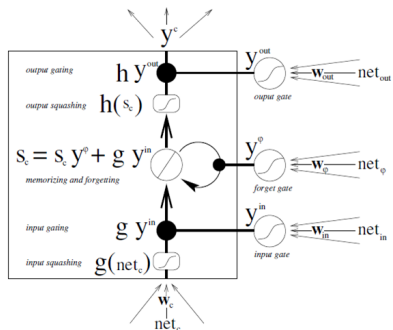
۱. نرون‌های لایه ورودی

که در این حالت $y^m(t-1)$ معادل بردار X خواهد بود.

۱. خروجی مرحله قبل کلیه بلوک‌های شبکه از جمله خود بلوک j

که در این حالت $y^m(t-1)$ معادل بردار $y^c(t-1)$ خواهد بود.

شبکه عصبی LSTM: آموزش ...



- مرحله ۳: مقدار خالص همه ورودی‌های ممکن به دروازه‌های فراموشی و فعال‌سازهای آن‌ها را محاسبه کنید

$$net_{\varphi_j}(t) = \sum_m w_{\varphi_j m} y^m(t-1) \quad , \quad y^{\varphi_j}(t) = f(net_{\varphi_j}(t))$$

فعال‌سازی بین صفر و یک

- net_{φ_j} معادل جمع وزندار کلیه ورودی‌های ممکن به دروازه فراموشی j آمین بلوک حافظه می‌باشد که این ورودی‌ها شامل:

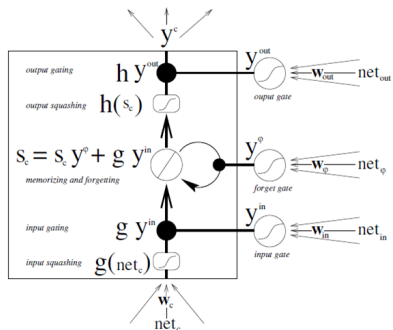
1. نرون‌های لایه ورودی: که در این حالت $y^m(t-1)$ معادل بردار X خواهد بود.
2. خروجی مرحله قبل کلیه بلوک‌های شبکه از جمله خود بلوک j : که در این حالت $y^m(t-1)$ معادل بردار $y^c(t-1)$ خواهد بود.

- مرحله ۴: مقدار خالص ورودی به سلول‌ها و همچنین حالت سلول‌ها را محاسبه کنید

$$net_{c_j}(t) = \sum_m w_{c_j m} y^m(t-1)$$

$$s_{c_j}(t) = s_{c_j}(t-1) y^{\varphi_j}(t) + y^{in_j}(t) g\left(net_{c_j}(t)\right) ; \text{ for } t > 0$$

شبکه عصبی LSTM: آموزش ...



- مرحله ۴: مقدار خالص همه ورودی‌های ممکن به سلول‌ها و همچنین حالت سلول‌ها را محاسبه کنید

$$\text{net}_{c_j}(t) = \sum_m w_{c_j m} y^m(t-1)$$

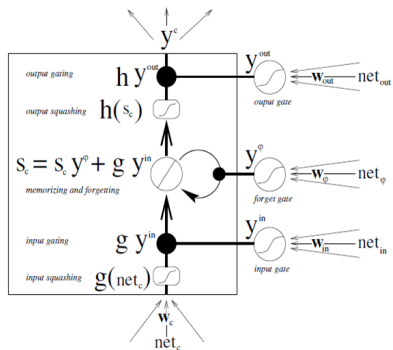
- net_{c_j} معادل جمع وزندار کلیه ورودی‌های ممکن به j آمین بلوک حافظه می‌باشد که این ورودی‌ها عبارتند از:

1. نرون‌های لایه ورودی: که در این حالت $y^m(t-1)$ معادل بردار X خواهد بود.
2. خروجی مرحله قبل کلیه بلوک‌های شبکه از جمله خود بلوک j : که در این حالت $y^m(t-1)$ معادل بردار $y^c(t-1)$ خواهد بود.

$$s_{c_j}(t) = s_{c_j}(t-1)y^{\phi_j}(t) + y^{\text{in}_j}(t)g(\text{net}_{c_j}(t)) ; \text{ for } t > 0$$

فعال سازی بین +۲ و -۲

شبکه عصبی LSTM: آموزش ...



- مرحله ۵: مقدار خالص همه ورودی‌های ممکن به دروازه‌های خروجی و فعال‌سازهای آن‌ها را محاسبه کنید

$$net_{out_j}(t) = \sum_m w_{out_j m} y^m(t-1) \quad , \quad y^{out_j}(t) = f\left(net_{out_j}(t)\right)$$

فعال‌سازی بین صفر و یک

- net_{out_j} معادل جمع وزندار کلیه ورودی‌های ممکن به دروازه خروجی j آمین بلوک حافظه می‌باشد که این ورودی‌ها شامل:

- نرون‌های لایه ورودی: که در این حالت $y^m(t-1)$ معادل بردار X خواهد بود.
- خروجی مرحله قبل کلیه بلوک‌های شبکه از جمله خود بلوک j : که در این حالت $y^m(t-1)$ معادل بردار $y^c(t-1)$ خواهد بود.

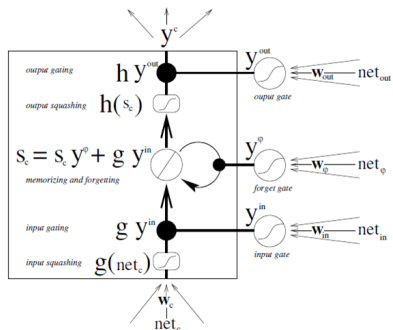
است

- مرحله ۶: خروجی سلول‌ها را محاسبه کنید

$$y^{c_j}(t) = y^{out_j}(t) h(s_{c_j}(t))$$

فعال‌سازی بین +۱ و -۱

شبکه عصبی LSTM: آموزش ...



مرحله ۷: مقدار خالص ورودی به نرون‌های لایه خروجی و فعال‌سازهای آنها را محاسبه کنید

$$\text{net}_k(t) = \sum_m w_{km} y^m(t) \quad , \quad y^k(t) = f(\text{net}_k(t))$$

مرحله ۸: خطای نرون‌های لایه خروجی را محاسبه کنید

$$\delta_k(t) = f'_k(\text{net}_k(t)) e_k(t) \quad , \quad e_k(t) := t^k(t) - y^k(t)$$

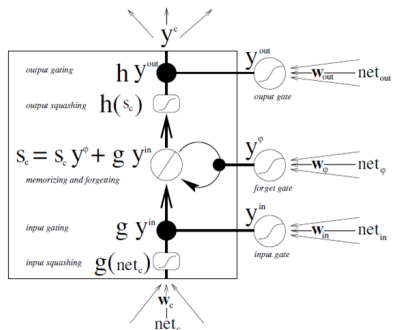
مرحله ۹: خطای دروازه‌های خروجی را محاسبه کنید

$$\delta_{\text{out}_j}(t) = f'_{\text{out}_j}(\text{net}_{\text{out}_j} t)) h(s_{c_j}(t)) \left(\sum_K w_{k c_j} \delta_k(t) \right)$$

وزن یال اتصالی از خروجی بلوک j به نرون k از لایه خروجی

جمع وزن‌دار خطاهای رسیده از لایه خروجی به بلوک j از حافظه

شبکه عصبی LSTM: آموزش ...



- مرحله ۱۰: خطای حالت سلول‌ها را به کمک رابطه زیر محاسبه کنید

$$e_{s_{c_j}}(t) = y^{out_j}(t) h'(s_{c_j}(t)) \left(\sum_k w_{k \ c_j} \delta_k(t) \right)$$

- مرحله ۱۱: برای محاسبه تغییرات وزنی یال‌های متصل به ورودی سلول‌ها، دروازه‌های ورودی و دروازه‌های فراموشی رُندهای زیر را محاسبه کنید

$$\frac{\partial s_{c_j}(t=0)}{\partial w_{lm}} = 0 \quad ; \text{ for } l \in \{\varphi, in, c_j\}$$

مقدار اولیه کلیه رُندها را برابر صفر قرار می‌دهیم

$$\frac{\partial s_{c_j}(t)}{\partial w_{c_j m}} = \frac{\partial s_{c_j}(t-1)}{\partial w_{c_j m}} y^{\varphi_j}(t) + g' \left(net_{c_j}(t) \right) y^{in_j}(t) y^m(t-1)$$

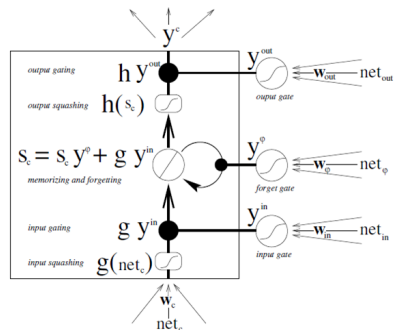
$$\frac{\partial s_{c_j}(t)}{\partial w_{in_j m}} = \frac{\partial s_{c_j}(t-1)}{\partial w_{in_j m}} y^{\varphi_j}(t) + g \left(net_{c_j}(t) \right) f'_{in_j} \left(net_{in_j}(t) \right) y^m(t-1)$$

$$\frac{\partial s_{c_j}(t)}{\partial w_{\varphi_j m}} = \frac{\partial s_{c_j}(t-1)}{\partial w_{\varphi_j m}} y^{\varphi_j}(t) + s_{c_j}(t-1) f'_{\varphi_j} \left(net_{\varphi_j}(t) \right) y^m(t-1)$$

- مرحله ۱۲: تغییرات وزنی یال‌های متصل به لایه خروجی سلول به نرون‌های خروجی را محاسبه کنید

$$\Delta w_{k \ c_j}(t) = \alpha \delta_k(t) y^{c_j}(t)$$

شبکه عصبی LSTM: آموزش ...



- مرحله ۱۳: تغییرات وزنی یال‌های متصل (وارد) به دروازه‌های ورودی و دروازه‌های فراموشی را محاسبه کنید

$$\Delta w_{lm}(t) = \alpha e_{s_{c_j}}(t) \frac{\partial s_{c_j}(t)}{\partial w_{lm}} \text{ for } l \in \{\varphi, in\}$$

- مرحله ۱۴: تغییرات وزنی یال‌های متصل به سلول‌ها (لایه ورودی به سلول) را محاسبه کنید

$$\Delta w_{c_j m}(t) = \alpha e_{s_{c_j}}(t) \frac{\partial s_{c_j}(t)}{\partial w_{c_j m}}$$

- مرحله ۱۵: تغییرات وزنی یال‌های متصل (وارد) به دروازه‌های خروجی را محاسبه کنید

$$\Delta w_{out_j m}(t) = \alpha \delta_{out_j}(t) y^m(t-1)$$

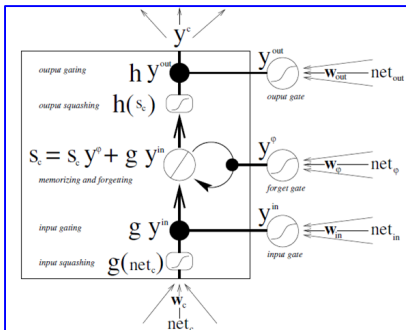


شبکه عصبی LSTM: آموزش

○ نکات کاربردی مهم

- در کلیه روابط، $y^m(t-1)$ به کلیه منابع ورودی به سلول در گام زمانی قبل اشاره می‌کند
 - اگر $y^m(t-1)$ به نرون‌های لایه ورودی اشاره کند آن را برابر مقدار نرون‌های لایه ورودی در مرحله زمانی فعلی قرار می‌دهیم.
- در انتهای هر دنباله آموزش، مقدار حالت و خروجی تمامی سلول‌ها و همچنین تمامی رُندها را برابر صفر قرار می‌دهیم
- به‌روز رسانی وزن‌ها در انتهای هر دنباله آموزشی صورت می‌گیرد.
 - مجموع تغییرات وزن‌ها به ازای همه دنباله‌ها به وزن‌های قبلی اضافه می‌شود
- بهتر است وزن‌های اولیه به‌صورت تصادفی و در بازه -0.1 تا 0.1 انتخاب شود.
- مقدار نرخ یادگیری را کوچک بگیرید، مثلاً 0.01 یا 0.001

شبکه عصبی LSTM: کاربرد ...



- مرحله ۰: برای هر دنباله آزمون $X^T = (x_1, x_2, \dots, x_{T-1}, x_T)$ مراحل ۱ تا ۷ را انجام دهید

- مرحله ۱: مقدار اولیه حالت تمامی سلول‌ها را برابر صفر قرار دهید

- مرحله ۲: برای تمامی بردارهای x_i مراحل ۳ تا ۸ را انجام دهید

- مرحله ۳: مقدار خالص ورودی به دروازه‌های ورودی و فعال‌سازهای آن‌ها را محاسبه کنید

$$net_{in_j}(t) = \sum_m w_{in_j m} y^m(t-1), \quad y^{in_j}(t) = f(net_{in_j}(t))$$

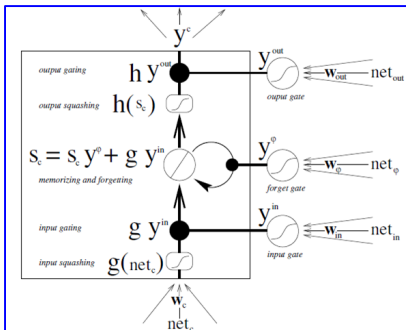
- net_{in_j} معادل جمع وزندار کلیه ورودی‌های ممکن به دروازه ورودی j آمین بلوک حافظه می‌باشد که این ورودی‌ها شامل:

۱. نرون‌های لایه ورودی: که در این حالت $y^m(t-1)$ معادل بردار X خواهد بود.

۲. خروجی مرحله قبل کلیه بلوک‌های شبکه از جمله خود بلوک j : که در این حالت $y^m(t-1)$ معادل بردار $y^c(t-1)$ خواهد بود.

است

شبکه عصبی LSTM: کاربرد ...



- مرحله ۴: مقدار خالص ورودی به دروازه‌های فراموشی و فعال‌سازهای آن‌ها را محاسبه کنید

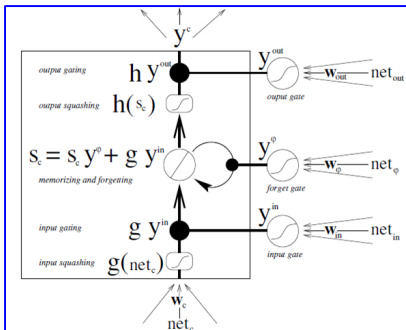
$$\text{net}_{\varphi_j}(t) = \sum_m w_{\varphi_j m} y^m(t-1) \quad , \quad y^{\varphi_j}(t) = f(\text{net}_{\varphi_j}(t))$$

- معادل جمع وزندار کلیه ورودی‌های ممکن به دروازه فراموشی j آمین بلوک حافظه می‌باشد که این ورودی‌ها شامل:

- نرون‌های لایه ورودی: که در این حالت $y^m(t-1)$ معادل بردار X خواهد بود.
- خروجی مرحله قبل کلیه بلوک‌های شبکه از جمله خود بلوک j : که در این حالت $y^m(t-1)$ معادل بردار $y^c(t-1)$ خواهد بود.

است

شبکه عصبی LSTM: کاربرد ...



- مرحله ۵: مقدار خالص ورودی به سلول‌ها و همچنین حالت سلول‌ها را محاسبه کنید

$$net_{c_j}(t) = \sum_m w_{c_j m} y^m(t-1)$$

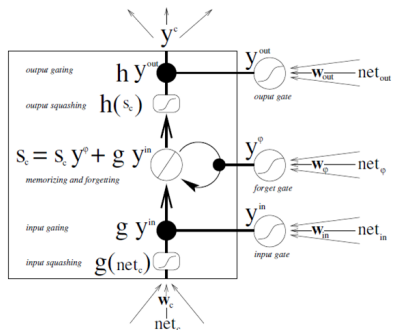
- معادل جمع وزندار کلیه ورودی‌های ممکن به j آمین بلوک حافظه می‌باشد که این ورودی‌ها شامل:

1. نرون‌های لایه ورودی: که در این حالت $y^m(t-1)$ معادل بردار X خواهد بود.
2. خروجی مرحله قبل کلیه بلوک‌های شبکه از جمله خود بلوک j : که در این حالت $y^m(t-1)$ معادل بردار $y^c(t-1)$ خواهد بود.

است

$$s_{c_j}(t) = s_{c_j}(t-1)y^{\phi_j}(t) + y^{in_j}(t)g\left(net_{c_j}(t)\right) ; for t > 0$$

شبکه عصبی LSTM: کاربرد ...



○ مرحله ۶: مقدار خالص ورودی به دروازه‌های خروجی و فعال‌سازهای آنها را محاسبه کنید

$$net_{out_j}(t) = \sum_m w_{out_j m} y^m(t-1) \quad , \quad y^{out_j}(t) = f(net_{out_j}(t))$$

○ net_{out_j} معادل جمع وزندار کلیه ورودی‌های ممکن به دروازه خروجی j اُمین بلوک حافظه می‌باشد که این ورودی‌ها شامل:

1. نرون‌های لایه ورودی: که در این حالت $y^m(t-1)$ معادل بردار X خواهد بود.

2. خروجی مرحله قبل کلیه بلوک‌های شبکه از جمله خود بلوک j : که در این حالت $y^m(t-1)$ معادل بردار $y^c(t-1)$ خواهد بود.

است

○ مرحله ۷: خروجی سلول‌ها را محاسبه کنید

$$y^{c_j}(t) = y^{out_j}(t) h(s_{c_j}(t))$$

○ مرحله ۸: مقدار خالص ورودی به نرون‌های لایه خروجی و فعال‌سازهای آنها را محاسبه کنید

$$net_k(t) = \sum_m w_{km} y^m(t) \quad , \quad y^k(t) = f(net_k(t))$$

شبکه عصبی LSTM: مثال ...

هدف: تمامی عناصر ممکن با توجه به
عنصر ورودی فعلی. یعنی دو کاراکتر
P و T

ورودی: کاراکتر B

ورودی

هدف

x_1	B	T	P	S	X	V	E		B	T	P	S	X	V	E	t_1
	1	0	0	0	0	0	0		0	1	1	0	0	0	0	
	0	1	0	0	0	0	0		1	0	0	0	0	0	0	
	1	0	0	0	0	0	0		0	1	1	0	0	0	0	
	0	0	1	0	0	0	0		0	1	0	0	0	1	0	
	0	1	0	0	0	0	0		0	1	0	0	0	1	0	
	0	0	0	0	0	1	0		0	0	1	0	0	1	0	
	0	0	1	0	0	0	0		0	0	0	1	1	0	0	
	0	0	0	0	1	0	0		0	1	0	0	0	1	0	
	0	0	0	0	0	1	0		0	0	1	0	0	1	0	
	0	0	0	0	0	1	0		0	0	0	0	0	0	1	
	0	0	0	0	0	0	1		0	1	0	0	0	0	0	
x_{10}	0	1	0	0	0	0	0		0	0	0	0	0	0	1	t_{10}

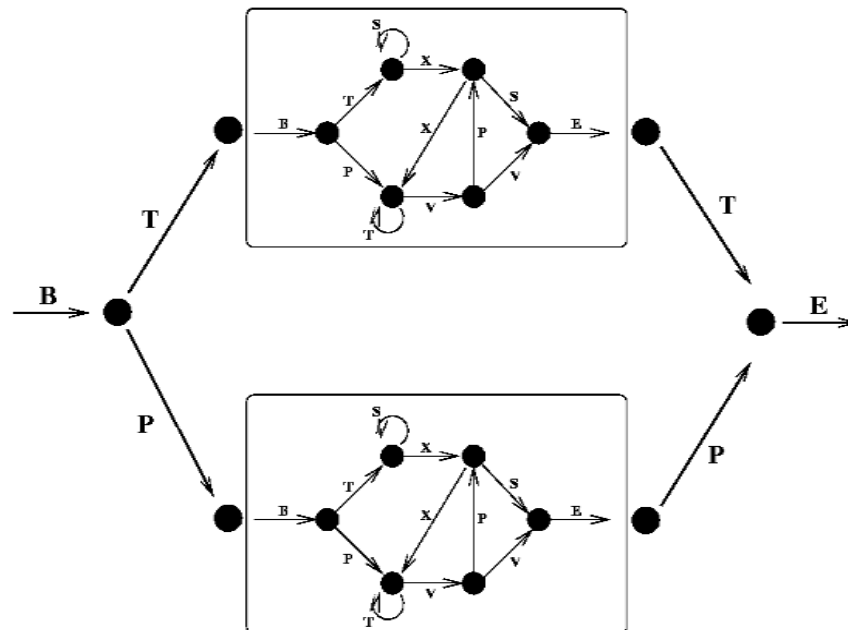
پیش بینی دنباله تولید شده توسط گرامر ربر

• دنباله آموزشی و دنباله هدف متناظر

$$X = (X_1, X_2, \dots, X_{10})$$

$$X_1 = (1, 0, 0, 0, 0, 0) = B$$

$$t_1 = (0, 1, 1, 0, 0, 0, 0) = T, P$$



شبکه عصبی LSTM: مثال ...

○ ساختار شبکه

- تعداد نرون‌های لایه ورودی: ۷

○ مربوط به ۷ نماد (B,P,S,T,V,X,E) که در دنباله ورودی می‌آید

- تعداد نرون‌های لایه خروجی: ۷

○ مربوط به ۷ نماد (B,P,S,T,V,X,E) که در دنباله هدف می‌آید

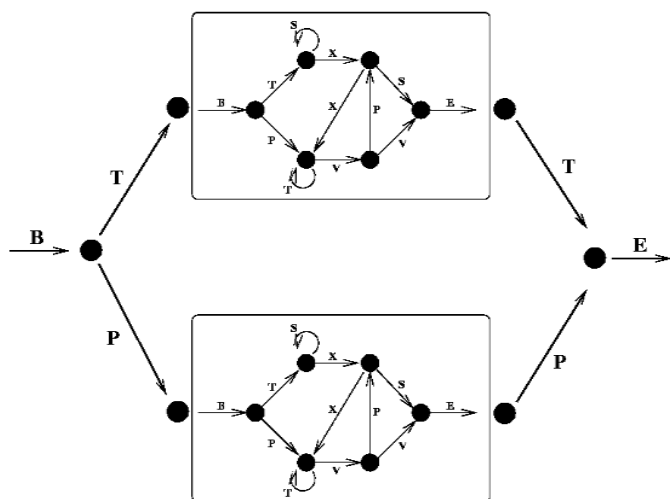
- تعداد بلوک‌های لایه پنهان: ۳

- نرخ یادگیری: ۰.۰۱

- وزن‌های اولیه: تصادفی در بازه $[-\frac{1}{\sqrt{f}}, \frac{1}{\sqrt{f}}]$

- تعداد داده‌های آموزشی

○ 60000 دنباله تصادفی





شبکه عصبی LSTM: مثال ...

آموزش ...

مرحله ۰: انتخاب وزن‌های اولیه به صورت تصادفی

وزن اولیه بین لایه ورودی و بلوک حافظه:

بصورت تصادفی بین -0.1 تا 0.1

بلوک حافظه

-0.1551	0.0359	0.0667
-0.1798	0.0956	-0.0473
-0.1779	-0.0722	0.0875
-0.1895	0.1393	-0.0325
0.1822	-0.0025	-0.1776
0.0101	-0.1541	0.0340
0.1025	0.1249	-0.0805

لایه ورودی

دروازه ورودی

-0.0602	-0.1342	-0.0259
-0.1914	0.0028	0.0784
-0.0636	0.0033	-0.1934
-0.0462	-0.1794	-0.1390
-0.1627	-0.0532	0.1598
0.1877	0.0339	-0.1814
-0.1240	0.1979	-0.1883

لایه ورودی

وزن اولیه بین لایه ورودی و دروازه ورودی:

بصورت تصادفی بین -0.1 تا 0.1

شبکه عصبی LSTM: مثال ...

دروازه فراموشی

-0.0362	-0.1848	-0.1138
0.0344	0.1690	-0.0059
-0.0738	-0.0507	-0.1122
-0.1411	0.1957	0.1747
0.1966	0.0174	-0.0890
-0.1570	0.0253	0.1812
-0.0663	-0.1248	0.1260

وزن اولیه بین لایه ورودی و دروازه فراموشی:

بصورت تصادفی بین -0.1 تا 0.1

دروازه خروجی

0.0298	-0.1180	-0.1193
-0.0638	-0.1049	-0.1492
-0.1123	0.1537	0.0751
0.1616	0.1685	0.1643
0.0732	0.1127	0.1004
-0.0513	-0.1032	0.1292
-0.0396	0.0334	-0.1100

وزن اولیه بین لایه ورودی و دروازه خروجی:

بصورت تصادفی بین -0.1 تا 0.1

ورودی بلوک‌ها

-0.0503	0.0632	-0.0233
-0.1866	0.0301	0.1111
-0.0375	0.1904	0.0278

وزن اولیه بین خروجی بلوک‌ها و ورودی بلوک‌ها:

بصورت تصادفی بین -0.1 تا 0.1

شبکه عصبی LSTM: مثال ...

دروازه ورودی

خروجی بلوک‌ها

-0.0453	-0.1578	-0.0879
-0.0325	0.0125	-0.1209
0.1743	0.0485	-0.0366

• وزن اولیه بین خروجی بلوک‌ها و دروازه ورودی:

• بصورت تصادفی بین -0.1 تا 0.1

دروازه فراموشی

خروجی بلوک‌ها

-0.0550	-0.0238	-0.0115
-0.1352	0.0681	-0.1800
0.1574	-0.1158	-0.0016

• وزن اولیه بین خروجی بلوک‌ها و دروازه فراموشی:

• بصورت تصادفی بین -0.1 تا 0.1

دروازه خروجی

خروجی بلوک‌ها

-0.0921	-0.0433	0.0446
0.0737	0.0235	0.0325
0.0992	0.0929	-0.1718

• وزن اولیه بین خروجی بلوک‌ها و دروازه خروجی:

• بصورت تصادفی بین -0.1 تا 0.1

لایه خروجی

خروجی بلوک‌ها

-0.1916	-0.1121	0.1680	-0.0545	-0.1061	-0.0808	0.1258
0.0049	-0.1078	-0.0724	-2.7298e-...	-0.1911	0.1304	0.1444
-0.0370	-0.0246	-0.0863	0.1254	-0.0796	-0.1454	0.0212

• وزن اولیه بین خروجی‌های بلوک‌ها و لایه خروجی:

• بصورت تصادفی بین -0.1 تا 0.1



شبکه عصبی LSTM: مثال ...

○ آموزش ...

- مرحله ۱: مقدار اولیه حالت و خروجی تمامی سلول‌ها را برابر صفر قرار دهید
- مرحله ۲: مقدار خالص ورودی به دروازه‌های ورودی و فعال‌سازهای آن‌ها را محاسبه کنید

$$((-0.0602*1)+(-0.1914*0)+(-0.0636*0)+(-0.0462*0)+(0.1627*0)+(0.1877*0)+(-0.1240*0))+((0*(-0.0453))*(0*(-0.0325))*(0*0.1743))=-0.0602$$

$$net_{in} = \begin{bmatrix} -0.0602 & -0.1342 & -0.0259 \end{bmatrix}$$

○ مقدار خالص ورودی به دروازه‌های ورودی

$$net_{in_j}(t) = \sum_m w_{in_j m} y^m(t-1)$$

$$y^{in} = \begin{bmatrix} 0.4850 & 0.4665 & 0.4935 \end{bmatrix}$$

$$\text{Sigmoid}(-0.0602)=0.4850$$

○ فعال‌ساز دروازه‌های ورودی

$$y^{in_j}(t) = f_{in_j}(net_{in_j}(t))$$



شبکه عصبی LSTM: مثال ...

- مرحله ۳: مقدار خالص ورودی به دروازه‌های فراموشی و فعال‌سازهای آن‌ها را محاسبه کنید

$$((-0.0362*1)+(0.0344*0)+(-0.0738*0)+(-0.1411*0)+(0.1966*0)+(-0.1570*0)+(-0.0633*0))+((0*(-0.0550))* (0*(-0.1352)) * (0*0.1574)) = -0.0362$$

$net_{\varphi} =$

-0.0362	-0.1848	-0.1138
---------	---------	---------

○ مقدار خالص ورودی به دروازه‌های فراموشی

$$net_{\varphi_j}(t) = \sum_m w_{\varphi_j m} y^m(t-1)$$

$y^{\varphi} =$

0.4910	0.4539	0.4716
--------	--------	--------

○ فعال‌سازهای دروازه‌های فراموشی

$$y^{\varphi_j}(t) = f_{\varphi_j}(net_{\varphi_j}(t))$$

$Sigmoid(-0.0362) = 0.4910$



شبکه عصبی LSTM: مثال ...

- مرحله ۴: مقدار خالص ورودی به سلول‌ها و همچنین حالت سلول‌ها را محاسبه کنید

○ مقدار خالص ورودی به سلول‌ها

$$(0.0359 * 1) + (0.0956 * 0) + (-0.0722 * 0) + (0.1393 * 0) + (-0.0025 * 0) + (-0.1541 * 0) + (0.1249 * 0) = 0.0359$$

$$net_c = \begin{bmatrix} -0.1551 & 0.0359 & 0.0667 \end{bmatrix}$$

$$net_{c_j}(t) = \sum_m w_{c_j m} y^m(t-1)$$

○ حالت سلول‌ها

$$s_c = \begin{bmatrix} -0.0751 & 0.0168 & 0.0329 \end{bmatrix}$$

$$s_{c_j}(t) = s_{c_j}(t-1)y^{\phi_j}(t) + y^{in_j}(t) g\left(net_{c_j}(t)\right)$$

$$(0 * 0.4910) + (0.485 * (-0.1548)) = -0.0751$$



شبکه عصبی LSTM: مثال ...

- مرحله ۵: مقدار خالص ورودی به دروازه‌های خروجی و فعال‌سازهای آنها را محاسبه کنید
- مقدار خالص ورودی به دروازه‌های خروجی

$$(0.0298 * 1) + (-0.0638 * 0) + (-0.1123 * 0) + (0.1616 * 0) + (0.0732 * 0) + (-0.0513 * +(-0.0396 * 0)) + ((0 * (-0.0921)) * (0 * 0.0737) * (0 * 0.0992)) = 0.0298$$

$$net_{out} = \begin{bmatrix} 0.0298 & -0.1180 & -0.1193 \end{bmatrix}$$

$$net_{out_j}(t) = \sum_m w_{out_j m} y^m(t-1)$$

- فعال‌سازهای دروازه‌های خروجی

$$y^{out} = \begin{bmatrix} 0.5074 & 0.4705 & 0.4702 \end{bmatrix}$$

$$y^{out_j}(t) = f_{out_j}(net_{out_j}(t))$$

$$\text{Sigmoid}(0.0298) = 0.5074$$



شبکه عصبی LSTM: مثال ...

$$0.5074 * (-0.0375) = -0.190$$

$$y^c = \begin{bmatrix} -0.0190 & 0.0039 & 0.0077 \end{bmatrix}$$

- مرحله ۶: خروجی سلول‌ها را محاسبه کنید

$$y^{c_j}(t) = y^{out_j}(t) h(s_{c_j}(t))$$

- مرحله ۷: مقدار خالص ورودی به نرون‌های لایه خروجی و فعال‌سازهای آنها را محاسبه کنید

○ مقدار خالص ورودی به نرون‌های لایه خروجی

$$\begin{aligned} &(-0.0190 * (-0.1916)) + (0.0039 * 0.0049) \\ &+ (0.0077 * (-0.0370)) = 0.0034 \end{aligned}$$

$$net_k(t) = \sum_m w_{km} y^m(t)$$

$$net_k = \begin{bmatrix} 0.0034 & 0.0015 & -0.0042 & 0.0020 & 6.5059e-04 & 9.2639e-04 & -0.0017 \end{bmatrix}$$

○ فعال‌سازهای نرون‌های لایه خروجی

$$y^k =$$

$$\begin{bmatrix} 0.5008 & 0.5004 & 0.4990 & 0.5005 & 0.5002 & 0.5002 & 0.4996 \end{bmatrix}$$

$$\text{Sigmoid}(0.0034) = 0.5008$$

$$y^k(t) = f(net_k(t))$$



شبکه عصبی LSTM: مثال ...

مرحله ۸: خطای نرون‌های لایه خروجی را محاسبه کنید

$$\delta_k(t) = f'_k(\text{net}_k(t)) e_k(t), e_k(t) := t^k(t) - y^k(t)$$

$$\delta_k = \begin{bmatrix} -0.1252 & 0.1249 & 0.1253 & -0.1251 & -0.1250 & -0.1251 & -0.1249 \end{bmatrix}$$

$$(0 - 0.5008) * (0.5008) * (1 - 0.5008) = -0.1251$$

مرحله ۹: خطای دروازه‌های خروجی را محاسبه کنید

$$\delta_{out} = \begin{bmatrix} -4.2666\text{e-}04 & -7.0063\text{e-}05 & 2.2690\text{e-}06 \end{bmatrix}$$

$$\delta_{out_j}(t) = f'_{out_j}(\text{net}_{out_j}(t)) h(s_{c_j}(t)) \left(\sum_K w_{k c_j} \delta_k(t) \right)$$

$$\begin{aligned} & 0.5074 * (1 - 0.5074) * (-0.0375) * \\ & ((-0.1252) * (-0.1916)) + (0.1249 * (-0.1121)) + (0.1253 * 0.1680) + ((-0.1251) * (-0.0545)) \\ & + ((-0.1250) * (-0.1061)) + ((-0.1251) * (-0.0808)) + (-0.1249 * 0.1258) = -0.00042 \end{aligned}$$



شبکه عصبی LSTM: مثال ...

- مرحله ۱۰: خطای حالت سلول‌ها را محاسبه کنید

$$e_{s_{c_j}}(t) = y^{out_j}(t) h'(s_{c_j}(t)) \left(\sum_k w_{k \ c_j} \delta_k(t) \right)$$

$e_{s_c} =$

0.0115	-0.0079	1.3007e-04
--------	---------	------------

$$0.5074 * 0.4993 * ((-0.1252) * (-0.1916)) + (0.1249 * (-0.1121)) + (0.1253 * 0.1680) + ((-0.1251)$$



شبکه عصبی LSTM: مثال ...

• مرحله ۱۱: رُندهای زیر را محاسبه کنید

○ محاسبه رُنْد یال‌های بین لایه ورودی و ورودی بلوک‌ها:

$$\frac{\partial s_{c_j}(t)}{\partial w_{c_j m}} = \frac{\partial s_{c_j}(t-1)}{\partial w_{c_j m}} y^{\phi_j}(t) + g' \left(net_{c_j}(t) \right) y^{in_j}(t) y^m(t-1)$$

ورودی بلوک‌ها

$$(0 * 0.4910) + (0.9940 * 0.4850 * 1) = 0.4821$$

لایه ورودی

0.4821	0.4664	0.4930
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0



شبکه عصبی LSTM: مثال ...

محاسبه رُند یال‌های بین خروجی بلوک‌ها و ورودی بلوک‌ها:

$$\frac{\partial s_{c_j}(t)}{\partial w_{c_j m}} = \frac{\partial s_{c_j}(t-1)}{\partial w_{c_j m}} y^{\varphi_j}(t) + g'(\text{net}_{c_j}(t)) y^{\text{inj}}(t) y^m(t-1)$$

ورودی بلوک‌ها

0	0	0
0	0	0
0	0	0

خروجی بلوک‌ها

$$(0 * 0.4910) + (0.9940 * 0.4850 * 0) = 0$$

محاسبه رُند یال‌های بین لایه ورودی و دروازه‌های ورودی:

$$\frac{\partial s_{c_j}(t)}{\partial w_{\text{inj } m}} = \frac{\partial s_{c_j}(t-1)}{\partial w_{\text{inj } m}} y^{\varphi_j}(t) + g(\text{net}_{c_j}(t)) f'_{\text{inj}}(\text{net}_{\text{inj}}(t)) y^m(t-1)$$

دروازه ورودی

-0.0387	0.0089	0.0167
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

لایه ورودی

$$(0 * 0.4910) + (-0.1548 * 0.4850 * (1 - 0.4850) * 1) = -0.0387$$

شبکه عصبی LSTM: مثال ...

○ محاسبه رُند یال‌های بین خروجی بلوک‌ها و دروازه‌های ورودی:

$$\frac{\partial s_{c_j}(t)}{\partial w_{in_j m}} = \frac{\partial s_{c_j}(t-1)}{\partial w_{in_j m}} y^{\varphi_j}(t) + g\left(\text{net}_{c_j}(t)\right) f'_{in_j}\left(\text{net}_{in_j}(t)\right) y^m(t-1)$$

خروجی بلوک‌ها

دروازه ورودی

0	0	0
0	0	0
0	0	0

$$(0 * 0.4910) + (-0.1548 * 0.4850 * (1 - 0.4850) * 0) = 0$$

○ محاسبه رُند یال‌های بین لایه ورودی و دروازه‌های فراموشی:

$$\frac{\partial s_{c_j}(t)}{\partial w_{\varphi_j m}} = \frac{\partial s_{c_j}(t-1)}{\partial w_{\varphi_j m}} y^{\varphi_j}(t) + s_{c_j}(t-1) f'_{\varphi_j}\left(\text{net}_{\varphi_j}(t)\right) y^m(t-1)$$

دروازه فراموشی

0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

$$(0 * 0.4910) + (0 * 0.4910 * (1 - 0.4910) * 1) = 0$$

لایه ورودی

شبکه عصبی LSTM: مثال ...

محاسبه رُند یال‌های بین خروجی بلوک‌ها و دروازه‌های فراموشی

$$\frac{\partial s_{c_j}(t)}{\partial w_{\varphi_j m}} = \frac{\partial s_{c_j}(t-1)}{\partial w_{\varphi_j m}} y^{\varphi_j}(t) + s_{c_j}(t-1) f'_{\varphi_j}(\text{net}_{\varphi_j}(t)) y^m(t-1)$$

دروازه فراموشی

خروجی بلوک‌ها

0	0	0
0	0	0
0	0	0

$$(0 * 0.4910) + (0 * 0.4910 * (1 - 0.4910) * 0) = 0$$

مرحله ۱۲: تغییرات وزنی یال‌های ورودی به لایه خروجی را محاسبه کنید

$$\Delta w_{k c_j}(t) = \alpha \delta_k(t) y^{c_j}(t)$$

$$0.1 * (-0.1251) * (-0.190) = 0.00238$$

لایه خروجی

خروجی بلوک‌ها

2.3841e-05	-2.3783e-05	-2.3850e-05	2.3825e-05	2.3808e-05	2.3812e-05	2.3781e-05
-4.9366e-06	4.9246e-06	4.9385e-06	-4.9333e-06	-4.9299e-06	-4.9306e-06	-4.9242e-06
-9.6905e-06	9.6668e-06	9.6942e-06	-9.6838e-06	-9.6773e-06	-9.6786e-06	-9.6661e-06

شبکه عصبی LSTM: مثال ...

- مرحله ۱۳: تغییرات وزنی یال‌های ورودی به دروازه‌های ورودی و دروازه‌های فراموشی را محاسبه کنید

محاسبه تغییر وزن یال‌های بین لایه ورودی و دروازه‌های ورودی

$$\Delta w_{in\ m}(t) = \alpha e_{s_{c_j}}(t) \frac{\partial s_{c_j}(t)}{\partial w_{l\ m}}$$

$$0.1 * 0.0115 * (-0.0387) = -0.000044$$

دروازه ورودی

-4.4571e-06	-7.0592e-07	2.1689e-08
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

لایه ورودی

محاسبه تغییر وزن یال‌های بین خروجی بلوک‌ها و دروازه‌های ورودی

$$\Delta w_{in\ m}(t) = \alpha e_{s_{c_j}}(t) \frac{\partial s_{c_j}(t)}{\partial w_{l\ m}}$$

$$0.1 * 0.0115 * 0 = 0$$

دروازه ورودی

0	0	0
0	0	0
0	0	0

خروجی بلوک‌ها

شبکه عصبی LSTM: مثال ...

محاسبه تغییر وزن یال‌های بین لایه ورودی و دروازه‌های فراموشی: $(\Delta w_{\phi m})$

$$\Delta w_{\phi m}(t) = \alpha e_{s_{c_j}}(t) \frac{\partial s_{c_j}(t)}{\partial w_{l m}}$$

دروازه فراموشی

$$0.1 * 0.0115 * 0 = 0$$

لایه ورودی

0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

محاسبه تغییر وزن یال‌های بین خروجی بلوک‌ها و دروازه‌های فراموشی:

$$0.1 * 0.0115 * 0 = 0$$

دروازه فراموشی

خروجی بلوک‌ها

0	0	0
0	0	0
0	0	0

$$\Delta w_{\phi m}(t) = \alpha e_{s_{c_j}}(t) \frac{\partial s_{c_j}(t)}{\partial w_{l m}}$$

شبکه عصبی LSTM: مثال ...

- مرحله ۱۴: تغییرات وزنی یال‌های ورودی به بلوک‌ها را محاسبه کنید

محاسبه تغییر وزن یال‌های بین لایه ورودی و ورودی بلوک‌ها:

$$0.1 * 0.0115 * 0.4821 = 0.00055$$

ورودی بلوک‌ها

5.5564e-05	-3.6822e-05	6.4122e-07
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

لایه ورودی

$$\Delta w_{c_j m}(t) = \alpha e_{s_{c_j}}(t) \frac{\partial s_{c_j}(t)}{\partial w_{c_j m}}$$

محاسبه تغییر وزن یال‌های بین خروجی بلوک‌ها و ورودی بلوک‌ها:

$$0.1 * 0.0115 * 0 = 0$$

ورودی بلوک‌ها

0	0	0
0	0	0
0	0	0

خروجی بلوک‌ها

$$\Delta w_{c_j m}(t) = \alpha e_{s_{c_j}}(t) \frac{\partial s_{c_j}(t)}{\partial w_{c_j m}}$$

شبکه عصبی LSTM: مثال ...

- مرحله ۵: تغییرات وزنی یال‌های ورودی به دروازه‌های خروجی را محاسبه کنید

محاسبه تغییر وزن یال‌های بین لایه ورودی و دروازه‌های خروجی:

$$0.1 * (-0.00042) * 1 = -0.000042$$

دروازه خروجی

-4.2666e-06	-7.0063e-07	2.2690e-08
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

لایه ورودی

$$\Delta w_{out_j m}(t) = \alpha \delta_{out_j}(t) y^m(t-1)$$

محاسبه تغییر وزن یال‌های بین خروجی بلوک‌ها و دروازه‌های خروجی

دروازه خروجی

$$0.1 * (-0.00042) * 0 = 0$$

0	0	0
0	0	0
0	0	0

خروجی بلوک‌ها

$$\Delta w_{out_j m}(t) = \alpha \delta_{out_j}(t) y^m(t-1)$$



شبکه عصبی LSTM: مثال

- مرحله ۱۶: تغییرات وزن محاسبه شده در این گام زمانی تنها ذخیره کنید (با وزن‌های قبلی شبکه جمع نشود)
- مرحله ۱۷: برای تمامی گام‌های زمانی تمامی مراحل ۱ تا ۱۶ را تکرار کنید
- مرحله ۱۸: در انتهای آخرین گام زمانی مجموع تغییرات وزن‌های ذخیره شده در مرحله ۱۶ را به وزن‌های شبکه اضافه نمایید

برای دنباله آموزشی بعدی تمامی گام‌های ۱ تا ۱۸ را تکرار کنید.



شبکه عصبی LSTM: مثال از کاربرد ...

فرض کنید وزن‌های بدست آمده از مرحله آموزش گرامر بصورت زیر باشد:
بلوک حافظه

○ وزن بین لایه ورودی و بلوک حافظه:

لایه ورودی

3.9151	-3.2348	-3.0269
0.9615	1.2860	4.2637
4.7583	0.7312	3.1965
4.3264	-2.0329	-2.1017
2.6357	-1.6746	0.8470
2.6612	-1.5102	-3.3226
4.5521	-0.3698	-0.7786

○ وزن بین لایه ورودی و دروازه ورودی:

دروازه ورودی

لایه ورودی

4.8625	2.0639	0.9591
1.4073	0.4284	1.7322
0.8088	-0.9127	-0.5602
0.3574	1.1875	0.7015
1.7834	0.8217	2.1451
-0.3265	0.8664	0.3634
1.0398	-0.0357	0.2478

شبکه عصبی LSTM: مثال از کاربرد ...

دروازه فراموشی

لایه ورودی

-4.2569	0.7874	2.2658
0.3351	0.7287	3.9433
2.2501	0.7359	-4.0696
-0.4863	1.3139	-0.2710
-0.5954	1.0097	0.3341
3.0903	0.9282	-2.2383
2.1039	0.2374	-0.2583

○ وزن بین لایه ورودی و دروازه فراموشی:

دروازه خروجی

○ وزن بین لایه ورودی و دروازه خروجی:

لایه ورودی

4.7166	1.0076	-0.1894
3.2706	3.2495	2.7557
-0.9194	0.9137	3.1509
-2.2420	3.6590	2.5515
1.5076	3.6636	4.4850
2.1366	-3.3384	1.4218
2.0888	2.3657	1.5189

○ وزن بین خروجی بلوک‌ها و ورودی بلوک‌ها:

ورودی بلوک‌ها

خروجی بلوک‌ها

-1.7355	0.3540	-5.7765
-3.1873	1.9654	-2.9279
8.1901	-1.0391	4.1037



شبکه عصبی LSTM: مثال از کاربرد ...

دروازه ورودی

خروجی بلوکها

1.2279	0.2777	1.0664
-4.1167	-2.1342	-3.4557
0.2021	0.3564	-0.1380

○ وزن بین خروجی بلوکها و دروازه ورودی:

دروازه فراموشی

خروجی بلوکها

0.5370	2.6669	-0.0521
1.1137	-2.7448	2.6063
-1.2385	-0.7886	-1.8696

○ وزن بین خروجی بلوکها و دروازه فراموشی:

دروازه خروجی

خروجی بلوکها

-1.0283	2.0204	-0.7760
-3.2926	4.3884	-0.2293
3.8964	-1.5900	1.4185

○ وزن بین خروجی بلوکها و دروازه خروجی:

لایه خروجی

خروجی بلوکها

-2.6071	6.4393	10.3822	-16.7756	-16.7744	8.1711	-2.4454
2.4884	-3.3992	14.0191	-2.1289	-2.1267	11.1964	2.1982
0.1496	14.6971	-5.2526	-2.0799	-2.0826	8.8300	-0.8454

○ وزن بین خروجی‌های بلوکها و لایه خروجی:



شبکه عصبی LSTM: مثال از کاربرد ...

- مرحله ۰: دنباله آزمون را معادل دنباله زیر قرار دهید:

ورودی

	B	T	P	S	X	V	E
x_1	1	0	0	0	0	0	0
	0	1	0	0	0	0	0
	1	0	0	0	0	0	0
	0	1	0	0	0	0	0
	0	0	0	1	0	0	0
	0	0	0	1	0	0	0
	0	0	0	0	1	0	0
	0	0	0	1	0	0	0
	0	0	0	0	0	0	1
x_{10}	0	1	0	0	0	0	0

هدف

	B	T	P	S	X	V	E
t_1	0	1	1	0	0	0	0
	1	0	0	0	0	0	0
	0	1	1	0	0	0	0
	0	0	0	1	1	0	0
	0	0	0	1	1	0	0
	0	0	0	1	1	0	0
	0	0	0	0	0	0	1
	0	1	0	0	0	0	0
t_{10}	0	0	0	0	0	0	1

- مرحله ۱: مقدار اولیه حالت و خروجی تمامی سلول‌ها را برابر صفر قرار دهید
- مرحله ۲: برای تمامی بردارهای x_i مراحل ۳ تا ۶ را انجام دهید



شبکه عصبی LSTM: مثال ...

○ مرحله ۳: دنباله ورودی برابر x_i و هدف را برابر t_i قرار دهید

$$X_1 = (1, 0, 0, 0, 0, 0) = \mathbf{B}$$

$$t_1 = (0, 1, 1, 0, 0, 0, 0) = \mathbf{T}, \mathbf{P}$$

○ مرحله ۴: مقدار خالص ورودی به دروازه‌های ورودی و فعال‌سازهای آنها را محاسبه کنید

$$(4.8625 * 1) + (1.4073 * 0) + (0.8088 * 0) + (0.3574 * 0) + (1.7834 * 0) + (-0.3265 * 0) + (1.0398$$

$$net_{in} = \begin{bmatrix} 4.8625 & 2.0639 & 0.9591 \end{bmatrix}$$

$$y^{in} = \begin{bmatrix} 0.9923 & 0.8873 & 0.7229 \end{bmatrix}$$

$$\text{Sigmoid}(4.8625) = 0.9923$$

○ مقدار خالص ورودی به دروازه‌های ورودی

$$net_{in_j}(t) = \sum_m w_{in_j m} y^m(t-1)$$

○ فعال‌ساز دروازه‌های ورودی

$$y^{in_j}(t) = f_{in_j}(net_{in_j}(t))$$



شبکه عصبی LSTM: مثال ...

- مرحله ۳: مقدار خالص ورودی به دروازه‌های فراموشی و فعال‌سازهای آن‌ها را محاسبه کنید

$$((-4.2569 * 1) + (0.3351 * 0) + (2.2501 * 0) + (-0.4863 * 0) + (-0.5954 * 0) + (3.0903 * 0) + (2.1039 * 0)) + ((0 * 0.5370) + (0 * 1.1137) + (0 * (-1.2385))) = -4.2569$$

$$net_{\varphi} = \begin{bmatrix} -4.2569 & 0.7874 & 2.2658 \end{bmatrix}$$

$$y^{\varphi} = \begin{bmatrix} 0.0140 & 0.6873 & 0.9060 \end{bmatrix}$$

$$\text{Sigmoid}(-4.2569) = 0.014$$

○ مقدار خالص ورودی به دروازه‌های فراموشی

$$net_{\varphi_j}(t) = \sum_m w_{\varphi_j m} y^m(t-1)$$

○ فعال‌سازهای دروازه‌های فراموشی

$$y^{\varphi_j}(t) = f_{\varphi_j}(net_{\varphi_j}(t))$$



شبکه عصبی LSTM: مثال ...

- مرحله ۴: مقدار خالص ورودی به سلول‌ها و همچنین حالت سلول‌ها را محاسبه کنید

○ مقدار خالص ورودی به سلول‌ها

$$((3.9151 * 1) + (0.9615 * 0) + (4.7583 * 0) + (4.3264 * 0) + (2.6357 * 0) + (2.6612 * 0) + (4.5521 * 0)) + ((0 * (-1.7355)) + (0 * (-3.1873)) + (0 * 8.1901)) = 3.9151$$

$$net_c = \begin{bmatrix} 3.9151 & -3.2348 & -3.0269 \end{bmatrix}$$

$$net_{c_j}(t) = \sum_m w_{c_j m} y^m(t-1)$$

○ حالت سلول‌ها

$$S_c = \begin{bmatrix} 1.9071 & -1.6403 & -1.3122 \end{bmatrix}$$

$$(0 * 0.014) + (0.9923 * 1.9218) = 1.907$$

$$s_{c_j}(t) = s_{c_j}(t-1)y^{\phi_j}(t) + y^{in_j}(t) g(net_{c_j}(t))$$



شبکه عصبی LSTM: مثال ...

- مرحله ۵: مقدار خالص ورودی به دروازه‌های خروجی و فعال‌سازهای آنها را محاسبه کنید
- مقدار خالص ورودی به دروازه‌های خروجی

$$((4.7166 * 1) + (3.2706 * 0) + (-0.9194 * 0) + (-2.2420 * 0) + (1.5076 * 0) + (2.1366 * 0) + (2.0888 * 0)) + ((0 * (-1.0283)) + (0 * (-3.2926)) + (0 * 3.8964)) = 4.7166$$

$$net_{out} = \begin{bmatrix} 4.7166 & 1.0076 & -0.1894 \end{bmatrix}$$

$$net_{out_j}(t) = \sum_m w_{out_j m} y^m(t-1)$$

- فعال‌سازهای دروازه‌های خروجی

$$y^{out} = \begin{bmatrix} 0.9911 & 0.7326 & 0.4528 \end{bmatrix}$$

$$y^{out_j}(t) = f_{out_j}(net_{out_j}(t))$$

$$\text{Sigmoid}(4.7166) = 0.9911$$



شبکه عصبی LSTM: مثال ...

$$0.9911 * (0.7414) = 0.7348$$

$$y^c = \begin{bmatrix} 0.7348 & -0.4946 & -0.2607 \end{bmatrix}$$

- مرحله ۶: خروجی سلول‌ها را محاسبه کنید

$$y^{cj}(t) = y^{outj}(t) h(s_{cj}(t))$$

- مرحله ۷: مقدار خالص ورودی به نرون‌های لایه خروجی و فعال‌سازهای آن‌ها را محاسبه کنید

○ مقدار خالص ورودی به نرون‌های لایه خروجی

$$(0.7348 * (-2.6071)) + (-0.4946 * 2.4884) + (-0.2670 * 0.1496) = -3.1854$$

$$net_k(t) = \sum_m w_{km} y^m(t)$$

$$net_k = \begin{bmatrix} -3.1854 & 2.5813 & 2.0647 & -10.7316 & -10.7312 & -1.8352 & -2.6637 \end{bmatrix}$$

○ فعال‌سازهای نرون‌های لایه خروجی

$$y^k = \begin{bmatrix} 0.0397 & 0.9296 & 0.8874 & 2.1843e-05 & 2.1853e-05 & 0.1376 & 0.0652 \end{bmatrix}$$

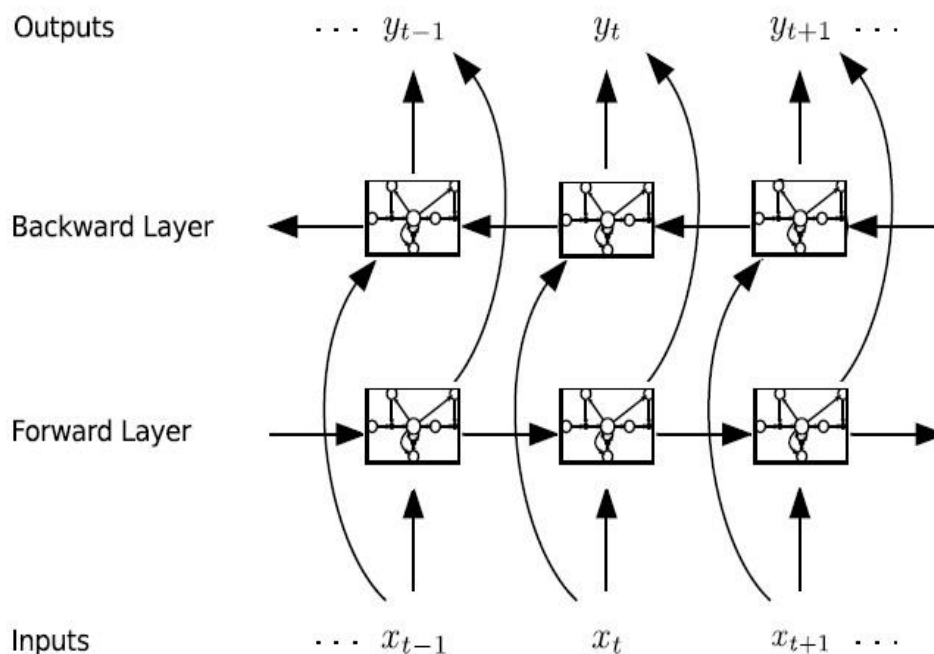
$$\text{Sigmoid}(-3.1854) = 0.0397$$

$$y^k(t) = f(net_k(t))$$

شبکه‌های عصبی LSTM: دوطرفه (Bidirectional) ...

○ ساختار

- شامل دو لایه پنهان بازگشتی مجزا (هر لایه پنهان شامل بلوک‌های LSTM می‌باشد).
- بین این دو لایه پنهان هیچ اتصالی وجود ندارد.
- هر دو لایه پنهان به یک لایه خروجی متصل شده‌اند.



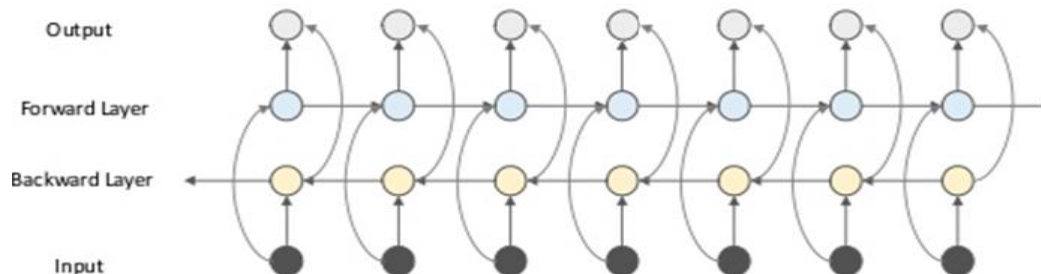
شبکه‌های عصبی LSTM: دوطرفه (Bidirectional)

○ ایده اصلی

هر دنباله ورودی در دو جهت زمانی رو به جلو و از انتها به دو لایه پنهان بازگشتی مجزا داده شود

- فرض کنید دنباله آموزشی به صورت $X^T = (x_1, x_2, \dots, x_{T-1}, x_T)$ و دنباله هدف متناظر برابر
- $t^T = (t_1, t_2, \dots, t_{T-1}, t_T)$ باشد
- در هر مرحله بردار x_i را به لایه Forward و $x_{T-(i-1)}$ را به لایه Backward ارسال کرده و مقدار هدف را بردار t_i قرار می‌دهیم.
- آموزش شبکه را با استفاده از الگوریتم مربوط به شبکه LSTM دنبال می‌کنیم

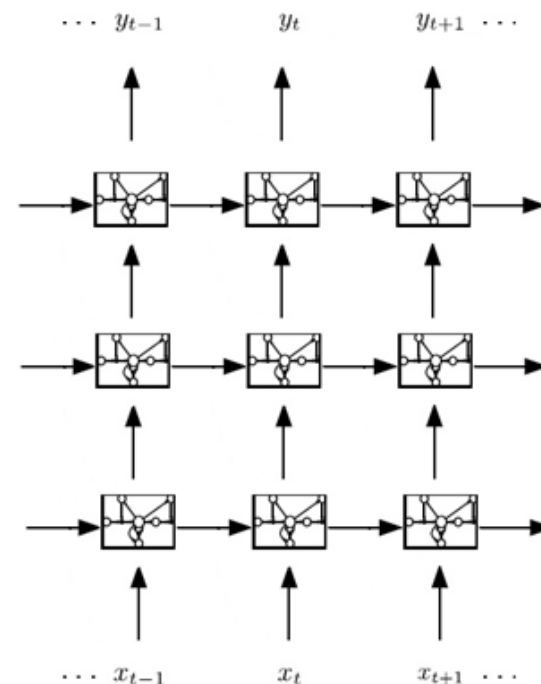
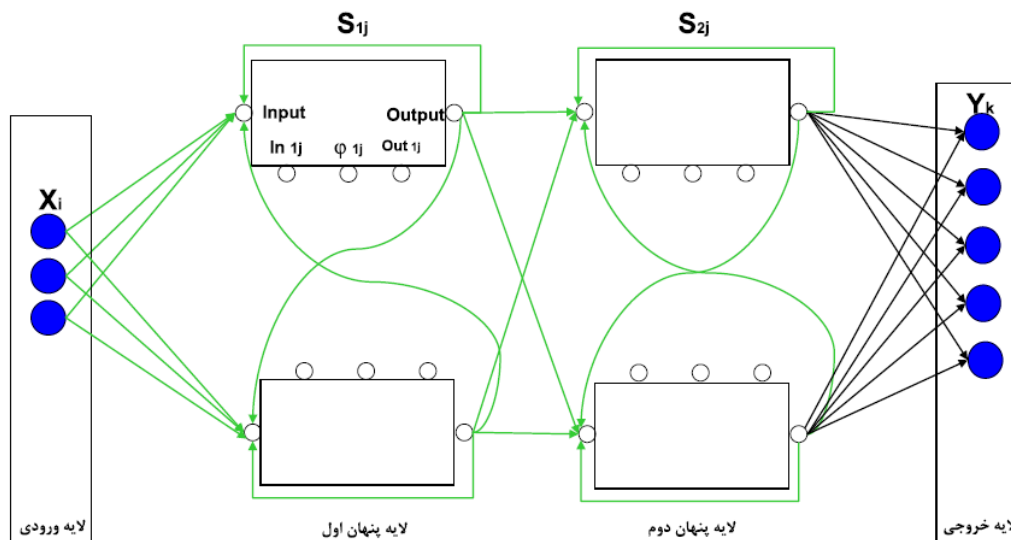
مقدار خالص رسیده به لایه خروجی جمع وزن‌دار مقدار خالص دو لایه Forward و Backward است



شبکه‌های عصبی LSTM: عمیق

○ شبکه عصبی با بیش از یک لایه مخفی

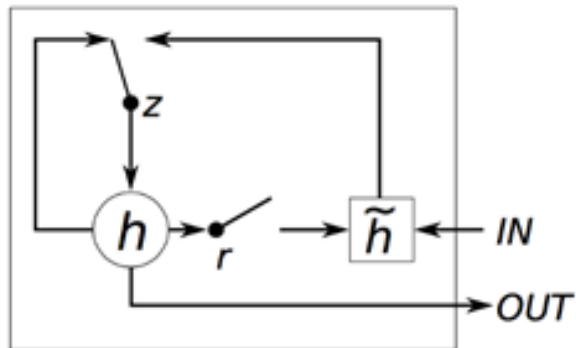
- خروجی هر لایه پنهان ورودی لایه پنهان بالاتر
- تقریب زننده جهانی
- قابلیت یادگیری بیشتر



شبکه‌های عصبی LSTM: نسخه ساده شده

○ شبکه عصبی واحد بازگشتی دروازه‌ای (GRU)

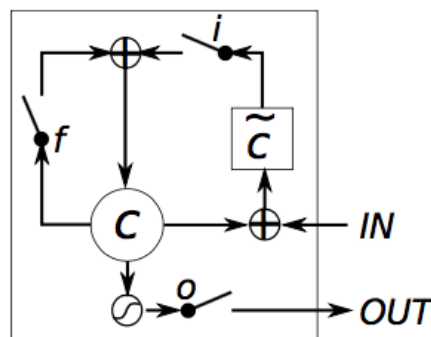
- عدم وجود دروازه خروجی



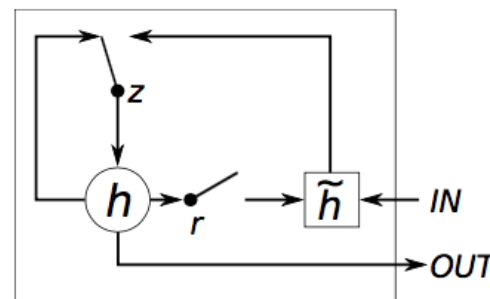
- دارای دو دروازه (LSTM دارای ۳ دروازه)

○ راه اندازی مجدد (reset) و بروزرسانی (update)

- عبور تمام مقدار سلول به خروجی یا ورودی سایر بلوک‌ها



(a) Long Short-Term Memory



(b) Gated Recurrent Unit

Figure 1: Illustration of (a) LSTM and (b) gated recurrent units. (a) i , f and o are the input, forget and output gates, respectively. c and \tilde{c} denote the memory cell and the new memory cell content. (b) r and z are the reset and update gates, and h and \tilde{h} are the activation and the candidate activation.



شبکه عصبی LSTM: تشخیص واج‌های فارسی ...

○ داده‌ها: فارسی‌دات

- شامل ۶۰۸۰ سیگنال صوتی
- داده آموزش: ۹۵٪ کل داده (معادل ۵۶۹۸ سیگنال)
- داده آزمون: ۵٪ کل داده (معادل ۳۸۲ سیگنال)

○ استخراج ویژگی

- روش مورد استفاده: MFCC
- تعداد ضرایب هر فریم: ۳۹
- طول فریم: ۱۶ میلی ثانیه

○ شبکه مورد استفاده: LSTM

○ ساختار شبکه

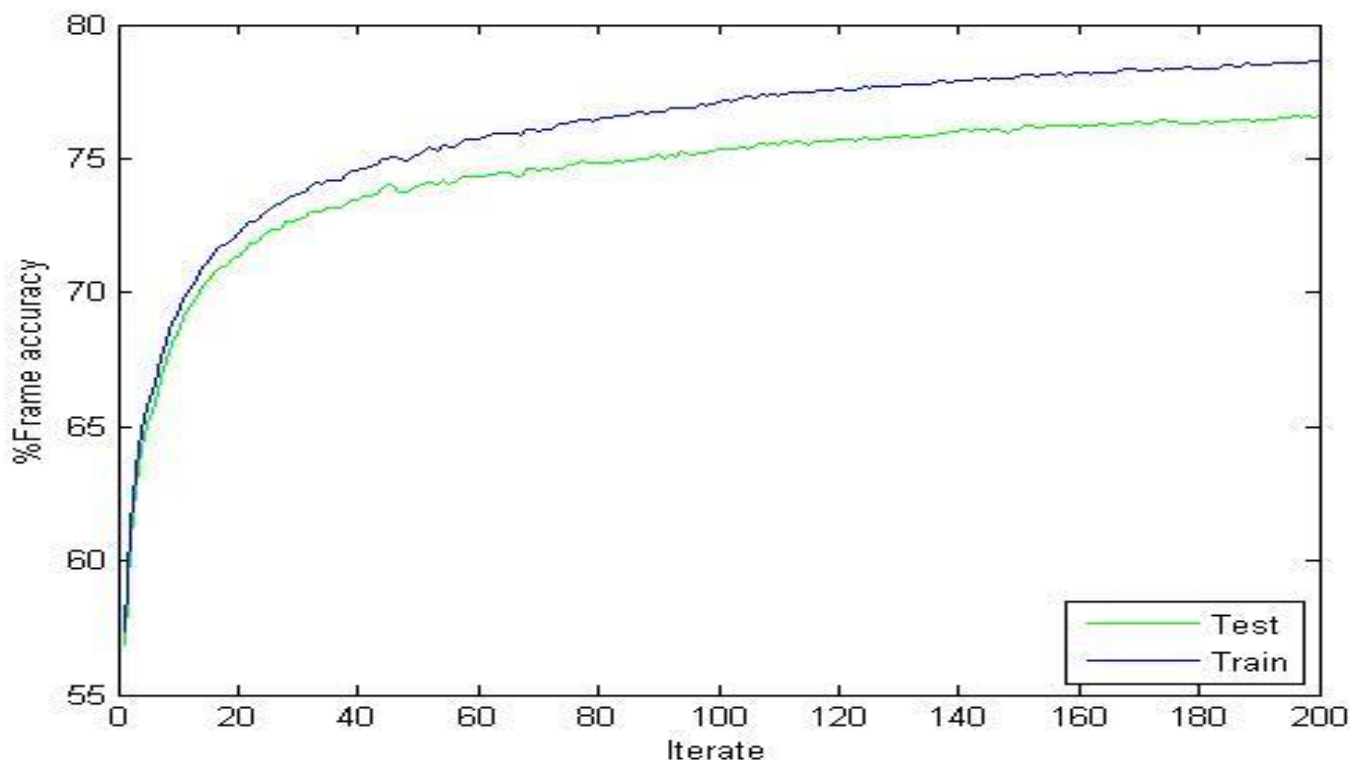
- نرون‌های لایه ورودی: ۳۹
- نرون‌های لایه خروجی: ۳۰
- وزن‌های اولیه: تصادفی در بازه $[-\frac{1}{\sqrt{f}}, \frac{1}{\sqrt{f}}]$
- نرخ یادگیری: ۰/۰۰۰۳
- تعداد بلوک حافظه: ۱۵۰

شبکه عصبی LSTM: تشخیص واج‌های فارسی ...

○ دقت به دست آمده روی فریم‌ها بعد از ۲۰۰ تکرار

• داده‌های آموزش: ۷۸/۶۲٪

• داده‌های آزمون: ۷۶/۵۹٪





شبکه عصبی LSTM دو طرفه: تشخیص واج‌های فارسی ...

○ داده‌ها: فارسی‌دات

- شامل ۶۰۸۰ سیگنال صوتی
- داده آموزش: ۸۰٪ کل داده (معادل ۴۸۶۴ سیگنال)
- داده آزمون: ۲۰٪ کل داده (معادل ۱۲۱۶ سیگنال)

○ استخراج ویژگی

- روش مورد استفاده: MFCC
- تعداد ضرایب هر فریم: ۳۹
- طول فریم: ۱۶ میلی ثانیه

○ شبکه مورد استفاده: Bidirectional LSTM

○ ساختار شبکه

- نرون‌های لایه ورودی: ۳۹
- نرون‌های لایه خروجی: ۳۰
- وزن‌های اولیه: تصادفی در بازه $[-\frac{1}{\sqrt{f}}, \frac{1}{\sqrt{f}}]$
- نرخ یادگیری: ۰/۰۰۰۳
- تعداد بلوک حافظه: ۱۲۰

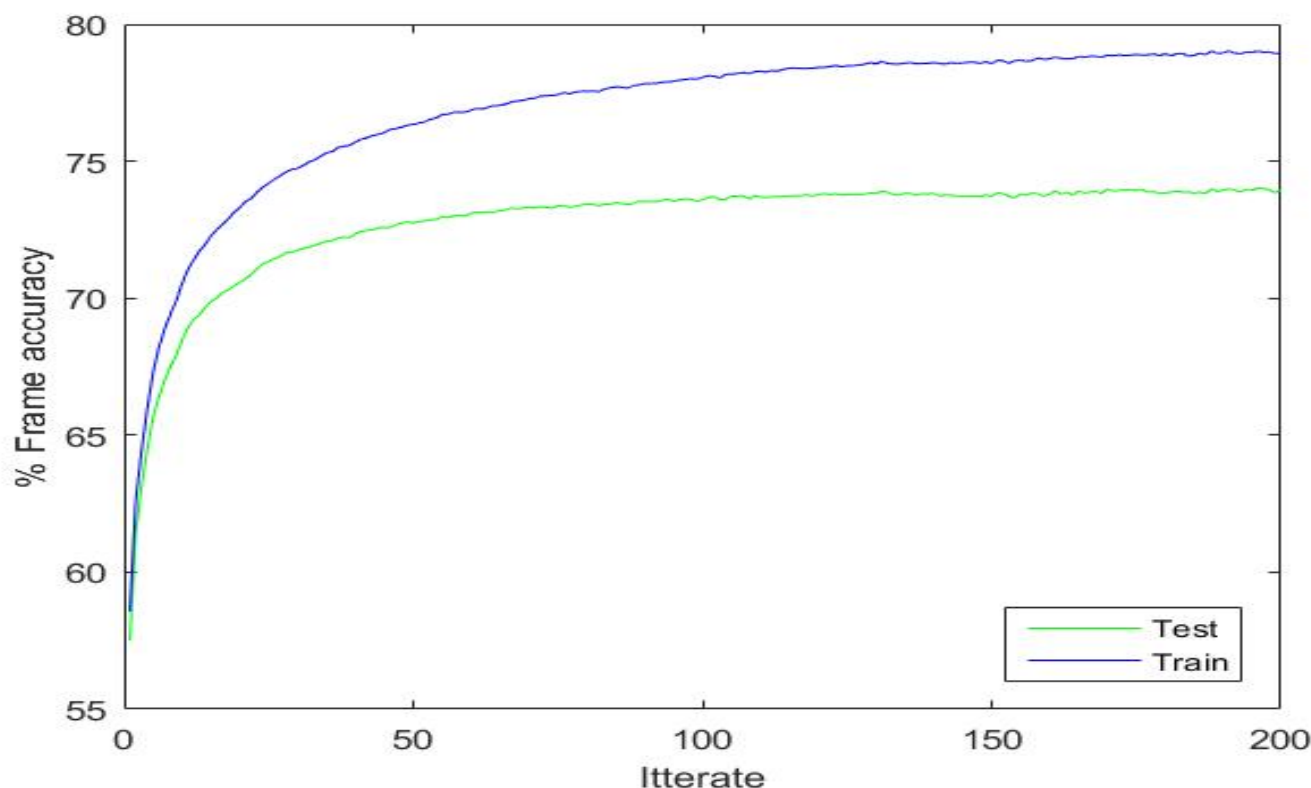


شبکه عصبی LSTM دوطرفه: تشخیص واج‌های فارسی ...

○ دقت به دست آمده روی فریم‌ها بعد از ۲۰۰ تکرار

• داده‌های آموزش: ۷۹.۰۱٪

• داده‌های آزمون: ۷۳.۸۴٪





شبکه عصبی LSTM دوطرفه: تشخیص واج‌های فارسی

○ دقت روی واج

- کارایی بالاتر شبکه‌های دوطرفه نسبت به شبکه‌های یک طرفه
- کارایی بالاتر شبکه‌های عمیق نسبت به شبکه‌های غیر عمیق

