یادگیری ژرف

نیم‌سال اول ۱۴۰۱ـ۱۴۰۰

دانشکده‌ی مهندسی کامپیوتر

مدرس: دکتر بیگی

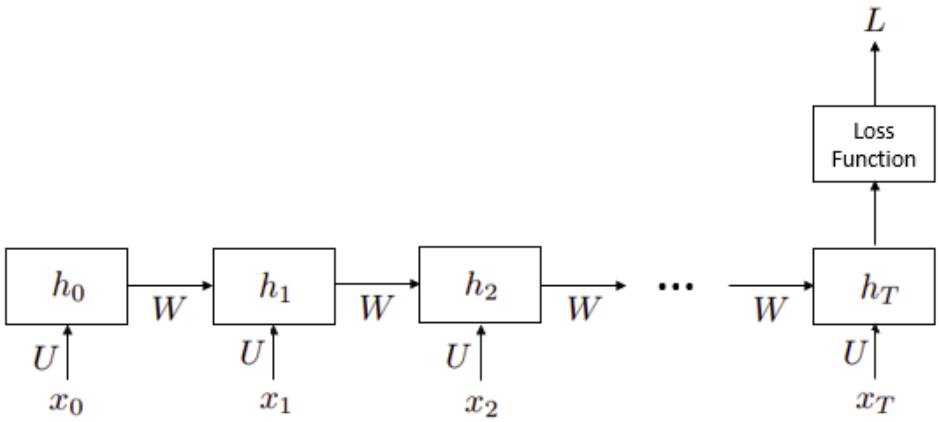
پاسخ تمرین سری سوم | شبکه‌های عصبی بازگشتی | موعد تحویل: ۲۷ آذر

- این نسخه صرفا حاوی پاسخ‌های پیشنهادی است و به پاسخ‌های شما در صورت منطقی و درست بودن نمره تعلق می‌گیرد.

سوالات نظری (۷۰ نمره)

مسئله‌ی ۱. (۲۰+۵ نمره)

(بخش ۱) با توجه به شبکه عصبی بازگشتی شکل زیر به سوالات پاسخ دهید. دقت کنید که برای سادگی تمام مقادیر یعنی ورودی ها و وزن ها و خروجی مقادیر اسکالر هستند. همچنین فرض کنید تمام توابع فعالساز σ هستند.



(آ) ابتدا گرادیان h_t یعنی $\frac{\partial L}{\partial h_t}$ را بر حسب گرادیان h_{t+1} یعنی $\frac{\partial L}{\partial h_{t+1}}$ بنویسید. $(1 \leqslant t \leqslant T-1)$

پاسخ:

$$h_{t+1} = \sigma(Ux_{t+1} + Wh_t)$$

$$\frac{\partial L}{\partial h_t} = \frac{\partial L}{\partial h_{t+1}} \frac{\partial h_{t+1}}{\partial h_t} = \frac{\partial L}{\partial h_{t+1}} .\sigma'(Ux_{t+1} + Wh_t).W$$

(ب) حال از رابطه قسمت قبل استفاده کرده و به شکل زنجیر وار گرادیان h_0 را بر حسب گرادیان h_T بنویسید.

پاسخ:

$$\frac{\partial L}{\partial h_0} = \frac{\partial L}{\partial h_T} \frac{\partial h_T}{\partial h_{T-1}} \cdots \frac{\partial h_2}{\partial h_1} \frac{\partial h_1}{\partial h_0}$$

$$= \frac{\partial L}{\partial h_T} (\sigma'(Ux_T + Wh_{T-1}).W) \ldots (\sigma'(Ux_2 + Wh_1).W)(\sigma'(Ux_1 + Wh_0).W)$$

(بخش ۲) حال می‌خواهیم روش هایی برای جلوگیری از محوشدگی و انفجار گرادیان را معرفی و تحلیل کنیم.

(آ) یکی از روش های مهم جلوگیری از محوشدگی و انفجار گرادیان مقداردهی اولیه صحیح وزن های شبکه است. توضیح دهید حداکثر مقدار اولیه W چند باشد تا فارغ از ورودی مطمئن باشیم که از همان ابتدا انفجار گرادیان رخ ندهد. (راهنمایی: یک حد بالا برای گرادیان h_0 پیدا کنید.)
پاسخ:

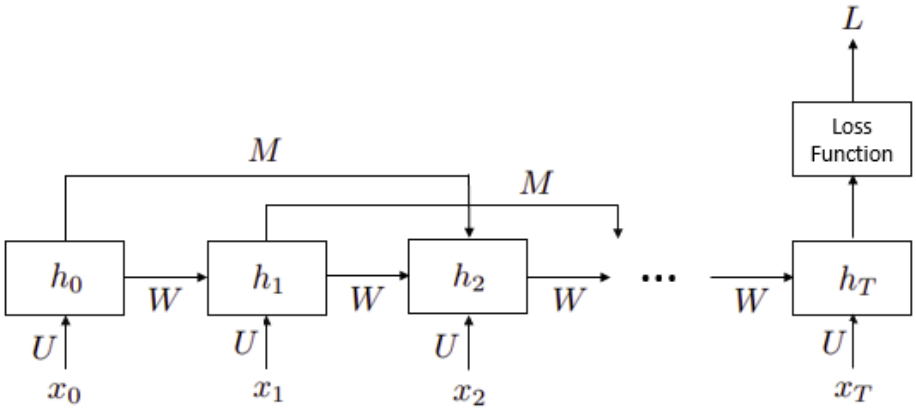
می‌دانیم حداکثر مقدار $\sigma'(x)$ برای هر x برابر $\frac{1}{4}$ است پس:

$$\frac{\partial L}{\partial h_0} = \frac{\partial L}{\partial h_T}(\sigma'(Ux_T + Wh_{T-1}).W)...(\sigma'(Ux_2 + Wh_1).W)(\sigma'(Ux_1 + Wh_0).W)$$

$$\leqslant \frac{\partial L}{\partial h_T}(\frac{1}{4}.W)...(\frac{1}{4}.W) = \frac{\partial L}{\partial h_T}(\frac{W}{4})^T$$

پر واضح است که اگر W عددی بیشتر از ۴ باشد حد بالای $\frac{\partial L}{\partial h_0}$ به سمت ∞ می‌رود پس برای اطمینان از این که شاهد انفجار گرادیان از همان ابتدا نباشم حداکثر مقدار ابتدایی W باید برابر ۴ باشد.

(ب) یکی از راه‌های جلوگیری از محوشدگی گرادیان استفاده از skip-connection ها است. شکل زیر را در نظر بگیرید که در ان هر h_t علاوه بر h_{t+1} به h_{t+2} هم متصل است. حال دوباره گرادیان h_t را برحسب گرادیان h_{t+1} و h_{t+2} نوشته و توضیح دهید چرا اینکار تا حد خوبی باعث کاهش اثر محوشدگی گرادیان می شود.($1 \leqslant t \leqslant T-2$)



پاسخ:

$$\frac{\partial L}{\partial h_t} = \frac{\partial L}{\partial h_{t+1}}\frac{\partial h_{t+1}}{\partial h_t} + \frac{\partial L}{\partial h_{t+2}}\frac{\partial h_{t+2}}{\partial h_t}$$

$$= \frac{\partial L}{\partial h_{t+1}}(\sigma'(Ux_{t+1} + Wh_t + Mh_{t-1}).W) + \frac{\partial L}{\partial h_{t+2}}(\sigma'(Ux_{t+2} + Wh_{t+1} + Mh_t).M)$$

در این حالت گرادیان از چند مسیر به حالت نهان h_t منتقل شده و اگر معادله را ادامه داده و ان را تا $\frac{\partial L}{\partial h_T}$ باز کنیم مشاهده خواهیم کرد که $\frac{\partial L}{\partial h_t}$ جمع تعداد جمله خواهد بود که تعداد این جملات برحسب $T-t$ نمایی خواهد بود. چون گرادیان کل از جمع این جملات محاسبه می شود شاهد کم رنگ شدن اثر محوشدگی گرادیان خواهیم بود.

(ج) یکی از راه‌حل های جلوگیری از انفجار گرادیان، برش گرادیان[۱] است که این خودبه دو زیرراه‌حل برش توسط مقدار[۲] و برش توسط اندازه[۳] تقسیم می شود. این دو را جداگانه توضیح دهید. برتری برش توسط اندازه را به برش توسط مقدار را توضیح دهید.

پاسخ:

در برش توسط مقدار ما یک آستانه مانند T در نظر گرفته و هر عنصر در بردار گرادیان که بزرگتر از T باشد را به T کاهش می دهیم، عناصر دیگر بدون تغییر باقی میمانند.
در برش توسط اندازه ما نیز ما یک آستانه مانند T اما این‌بار برروی اندازه بردار گرادیان قرار داده و اگر اندازه بردار گرادیان بیش‌تر از T باشد، بردار گرادیان را طوری نرمال کرده که اندازه آن برابر T بشود.
در برش توسط مقدار چونکه ما گرادیان را در برخی جهات تقطیع کرده و در برخی دیگر از جهات نگه‌می‌داریم، این کار باعث می‌شود بردار گرادیان نهایی در جهت بردار گرادیان ابتدایی نباشد در حالی که در برش توسط اندازه چون تمام عناصر گرادیان به یک اندازه scale می شوند این اتفاق رخ نخواهد داد.

مسئله‌ی ۲. (۲۰+۱۰ نمره)

در این مسئله میخواهیم با مفاهیمی در تولید دنباله در شبکه های Seq۲Seq و مزایا و معایب آن‌ها آشنا شویم.

(بخش ۱) در بخش اول می‌خواهیم مفهموم teacher forcing را بررسی کنیم. برای تولید دنباله ما می‌توانیم یک استراتژی خام اولیه در نظر بگیریم، می توان برای تولید نشانه[۴] $t+۱$ توسط رمزگشای[۵] زمان $t+۱$، نشانه تولید شده توسط شبکه در زمان t را به عنوان ورودی به دیکودر زمان $t+۱$ بدهیم اما این حالت مشکلاتی دارد.

(آ) ابتدا توضیح دهید این مشکلات چه چیز هایی هستند و سپس روش teacher forcing را توضیح داده و بگویید که teacher forcing چگونه این مشکلات را برطرف می کند.

پاسخ:

اگر در تولید دنباله در زمان آموزش ما در هر مرحله نشانه ایجاد شده در زمان t را به ورودی شبکه در زمان بدهیم روند همگرایی شبکه بسیار کند و ناپایدار خواهد شد چرا که مخصوصا در ابتدای روند آموزش، شبکه بد عمل کرده و نشانه های ایجاد شده صحیح نخواهند بود اما روش teacher forcing بجای دادن نشانه خروجی شبکه در زمان t به ورودی شبکه در زمان $t+۱$ می‌آید و حقیقت مبنا[۶] خروجی مرحله ی t را به ورودی مرحله $t+۱$ می دهد.

(ب) مشکل اصلی teacher forcing موضوعی به نام exposure bias است. این مشکل را توضیح دهید.

پاسخ:

این مشکل در حقیقت باعث ایجاد distribution shift در زمان اموزش و تست می‌شود چرا که در زمان آموزش مدل ما به دیدن حقیقت مبنا نشانه مرحله زمانی قبل عادت کرده است و تولید خروجی در زمان $t+۱$ را با فرض داشتن برچسب خروجی های زمان های ۱ تا t انجام می‌دهد اما هنگامی که زمان تست فرامی‌رسد شبکه دیگر هنگام تولید خروجی برای زمان $t+۱$ به برچسب خروجی زمان های ۱ تا t دسترسی ندارد و باید از خروجی‌های تولید شده توسط خودش در زمان‌های قبل استفاده کند که این پیش‌بینی ها لزوما برچسب واقعی آن مراحل زمانی نیست که این عامل باعث افت دقت و کارایی شبکه در زمان تست می‌شود.

(ج) یکی از راه‌حل‌های مشکل exposure bias تکنیک scheduled sampling است، این تکنیک را توضیح داده و بگویید این تکنیک چگونه باعث کاهش اثر exposure bias می شود.

---

۱ gradient clipping
۲ clipping by value
۳ clipping by norm
۴ token
۵ decoder
۶ ground truth

پاسخ:

در این روش ما در مرحله زمانی $t+1$ یک تصمیم‌گیری انجام می‌دهیم بدین صورت که فرض کنید یک نمونه از متغیر تصادفی برنولی با احتمال p در نظر می‌گیریم اگر نمونه برابر ۱ شد برچسب واقعی مرحله t را به ورودی شبکه می‌دهیم و اگر نمونه برابر ۰ شد خروجی تولید شده توسط شبکه را از مرحله زمانی t به ورودی می‌دهیم. بدین منظور منطقی است در ابتدای روند آموزش شبکه مقدار p بزرگ باشد و هرچه جلوتر می‌رویم مقدار آن را مطابق با یک زمان‌بندی(خطی یا نمایی یا ...) کوچک کنیم. واضح است در انتهای آموزش مدل تقریبا از همان توزیعی ورودی دریافت می‌کند که در زمان اعتبارسنجی و تست دریافت می‌کند و بدین صورت اثر مشکل exposure bias کاهش می‌یابد. بدین‌ترتیب هرچقدر که در طول آموزش جلو میرویم مدل یاد می‌گیرد که بیشتر به پیش‌بینی خود اتکا کند.

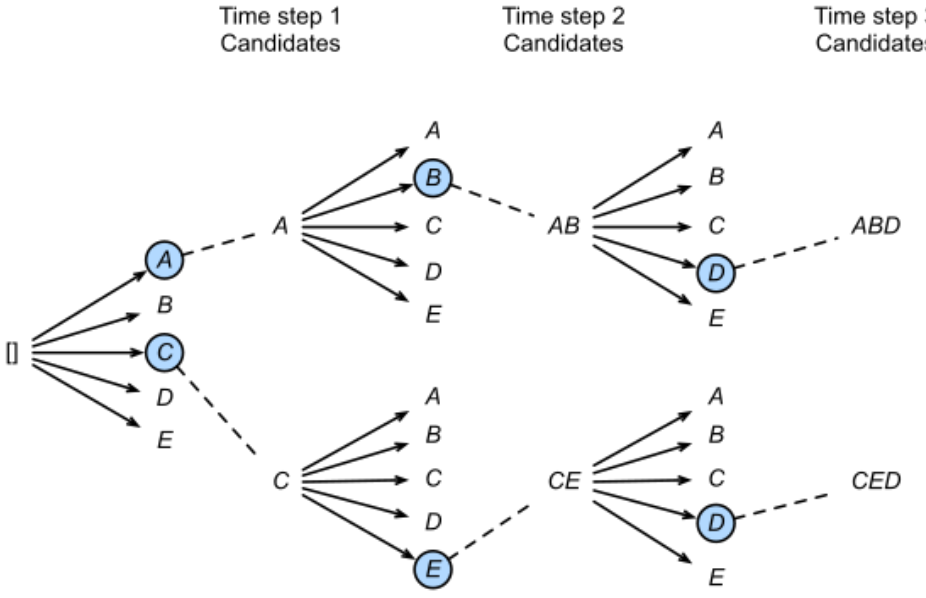
(بخش ۲) حال در بخش دوم مسئله میخواهیم بر روی الگوریتم جستجوی موجی[۷] تمرکز کنیم. این الگوریتم در تقابل با الگوریتم حریصانه برای تولید دنباله در زمان رمزگشایی مطرح می شود.

(آ) ابتدا تفاوت دو الگوریتم جستجوی موجی و الگوریتم حریصانه برای تولید دنباله را بیان کنید.

پاسخ:

در الگوریتم حریصانه در هر مرحله‌ی زمانی نشانه‌ای با بیشترین احتمال در لایه‌ی softmax انتخاب می‌شود که با توجه به حریصانه بودن الگوریتم لزوما بهینگی خوبی حاصل نمی‌شود.

در جستجوی موجی اما ما هر زمان فقط نشانه با بیشترین احتمال را انتخاب نمی‌کنیم بلکه در ابتدا k نشانه با بیشترین احتمال را نگه‌داشته و برای هرکدام سراغ پیش‌بینی نشانه مرحله زمانی دوم میرویم. در اینجا دوباره حداکثر k شاخه با بیشترین احتمال دو_نشانه‌ای را نگه‌داشته و به سراغ پیش‌بینی نشانه سوم برای هرکدام از آن k شاخه می‌رویم و این الگو تا انتهای تولید دنباله ادامه یافته و درنهایت شاخه با بیشترین احتمال به عنوان خروجی برگردادنده میشود. در شکل پایین جستجوی موجی برای ایجاد دنباله با $k=2$ را می‌بینیم.



(ب) در الگوریتم جستجوی موجی ابرپارامتری بنام k وجود دارد که حداکثر تعداد شاخه‌های جستجوی ما در هر زمان را نشان می‌دهد. توضیح دهید که کاهش بیش از حد k باعث چه مشکلاتی می‌شود. همچنین توضیح دهید افزایش بیش از اندازه k چه مشکلاتی بوجود می‌آورد.

پاسخ:

[۷] beam search

اگر k رابیش از اندازه کم کنیم ما فضای جستجو را برای تولید یک دنباله خوب محدود می‌کنیم و لزوما نمی‌توان به یک دنباله مناسب دست‌یافت برای مثال درحالت تولید جمله کم کردن بیش از اندازه k ممکن است باعث شود که نتوانیم جملاتی با ظاهر گرامری مناسب تولید کنیم، همچنین واضح است که در حالتی که $k = ۱$ باشد جستجوی موجی به همان الگوریتم حریصانه تبدیل می‌شود.
بزرگ کردن بیش از اندازه k هر چند باعث می‌شود فضای جستجوی ما بزرگ و تولید دنباله به‌ظاهر بهبود یابد اما مشکلی دارد و آن این است که این‌کار باعث تولید دنباله‌هایی با احتمال بالا می‌شود و این موضوع تنوع تولید دنباله را از مدل شما می‌گیرد برای مثال در چت‌بات‌ها و سامانه‌های گفتگوی خودکار این‌کار باعث ایجاد جملاتی ابتدایی و کاملا روزمره می‌شود زیرا این جملات احتمال بالایی دارند، برای مثال ممکن است این چت‌بات در در پاسخ به خیلی از سوالات شما جمله "نمی‌دانم" را استفاده کند.

(بخش ۳) حال در بخش سوم مسئله می خواهیم به موضوع دیگری برای تولید دنباله بپردازیم. در الگوریتم حریصانه همیشه کلمه با بیشترین احتمال در لایه‌ی softmax به عنوان کلمه خروجی انتخاب می شد، اما روش دیگری برای این‌کار وجود دارد و آن انتخاب تصادفی کلمه خروجی براساس احتمال های لایه softmax است.

(آ) توضیح دهید که مزایای این حالت به حالت انتخاب کلمه با بیشترین احتمال چیست.

پاسخ:

این کار باعث می‌شود که بتوانیم دنباله‌هایی با تنوع بیشتر را در خروجی تولید کنیم.

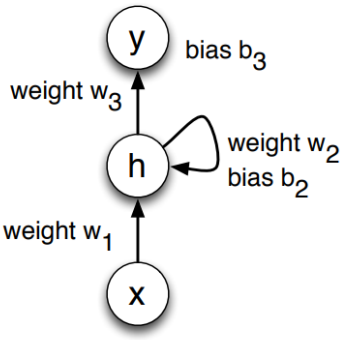
(ب) براین اساس دو روش sampling بنام های pure sampling و top-k sampling معرفی می شوند تفاوت این دو روش نمونه برداری را توضیح دهید. اثرات و مزایا و معایب زیاد یا کم‌کردن k در top-k sampling را شرح دهید.

پاسخ:

در pure sampling نمونه گیری برروی تمام خروجی لایه softmax انجام می‌شود اما در top-k sampling ما k تا از خروجی‌ها با بیشترین احتمال را نگه‌داشته و فقط از آن‌ها نمونه برداری می‌کنیم.
اگر k را کم کنیم در حقیقت داریم تنوع خروجی را کم می‌کنیم اما در عوض داریم با ریسک کمتری کار می‌کنیم و احتمال تولید جملاتی با گرامر نامناسب یا جملات غیرعادی را پایین می‌آوریم در حالت $k = ۱$ همان الگوریتم حریصانه را خواهیم داشت.
با زیاد کردن k دقیقا برعکس اتفاقات بالا رخ می‌دهد یعنی هرچند می‌توانیم دنباله‌هایی با تنوع بیشتری تولید کنیم اما این دنباله‌ها دارای ریسک بیشتری هستند. برای مثال در این حالت مدل شما ممکن است یک کلمه کاملا غیرعادی در جمله تولید کند. در حالتی هم k بیشترین مقدار خود را بگیرد top-k sampling به pure sampling تبدیل می‌شود. همانطور که می‌شود حدس زد k هم یک ابرپارامتر مهم در تولید دنباله است و برای تولید دنباله‌هایی مناسب با توجه به کاربرد باید تنظیم شود.

مسئله‌ی ۳. (۱۰ نمره)

یک شبکه بازگشتی به صورت مقابل را در نظر بگیرید. وزن ها و بایاس ها را به گونه ای تعیین کنید که در هر دنباله ای از اعداد تا زمانی که ورودی شبکه ۱ باشد، خروجی شبکه یک باقی بماند و به محض اینکه ورودی شبکه به صفر تغییر کند خروجی شبکه صفر شده و صفر باقی بماند. برای مثال خروجی شبکه به ازای ورودی ۱۱۱۰۱۰۱ برابر با ۱۱۱۰۰۰۰ می باشد.



پاسخ:

h_t	y_t
0	1
1	0

h_{t-1}	x_t	h_t
0	0	1
0	1	0
1	0	1
1	1	1

One possible setting of the weights and biases which achieves these relationships is:

$w_1 = -1$

$w_2 = 1$

$b_2 = 0.5$

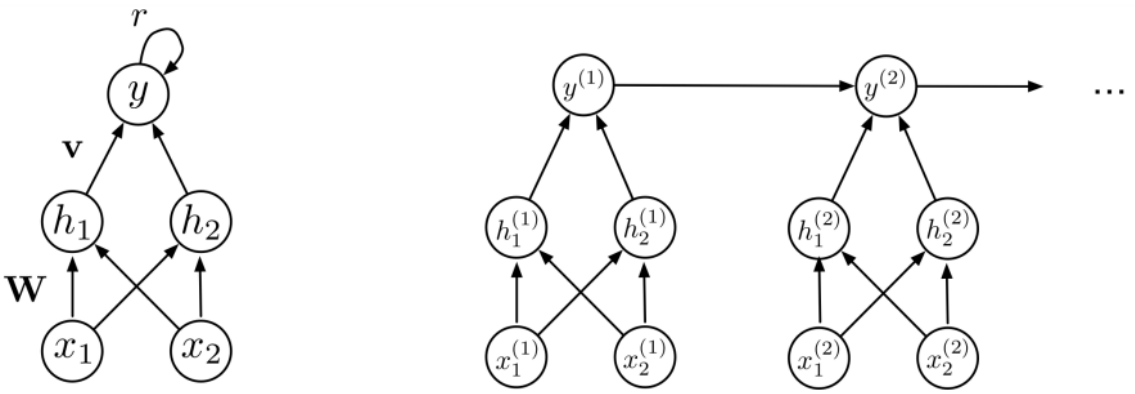
$w_3 = -1$

$b_3 = 0.5$

Scaling all these numbers by the same positive constant will also be a valid solution.

مسئله‌ی ۴. (۵ نمره)

یک شبکه بازگشتی بصورت مقابل را در نظر بگیرید. فرض کنید این شبکه دو دنباله از اعداد صفر و یک را دریافت کرده و اگر دو دنباله برابر بودند عدد ۱ و در غیر اینصورت عدد صفر را به عنوان خروجی بر می گرداند.



$$\mathbf{h}^{(t)} = \phi\left(\mathbf{W}\mathbf{x}^{(t)} + \mathbf{b}\right)$$

$$y^{(t)} = \begin{cases} \phi\left(\mathbf{v}^\top \mathbf{h}^{(t)} + r y^{(t-1)} + c\right) & \text{for } t > 1 \\ \phi\left(\mathbf{v}^\top \mathbf{h}^{(t)} + c_0\right) & \text{for } t = 1, \end{cases}$$

$$\phi(z) = \begin{cases} 1 & \text{if } z > 0 \\ 0 & \text{if } z \leq 0 \end{cases}$$

ماتریس $\mathbf{W}$ یک ماتریس ۲ $\times$ ۲ و b و v بردارهای دو بعدی و c و r و c، مقادیر اسکالر می باشد. آن ها را به گونه ای تعیین کنید که شبکه کارکرد تعریف شده را داشته باشد. (راهنمایی: خروجی $y^{(t)}$ در هر لحظه نشان می دهد آیا دو دنباله تا آن لحظه برابر بوده‌اند یا خیر. لایه مخفی اول نشان میدهد آیا دو ورودی در لحظه t صفر بوده‌اند یا خیر و لایه مخفی دوم نشان می‌دهد آیا دو ورودی در لحظه t، ۱ بوده‌اند یا خیر.)

پاسخ:

$$\mathbf{W} = \begin{pmatrix} -1 & -1 \\ 1 & 1 \end{pmatrix}$$
$$\mathbf{b} = \begin{pmatrix} 0.5 \\ -1.5 \end{pmatrix}$$
$$\mathbf{v} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$
$$r = 1$$
$$c = -1.5$$
$$c_0 = -0.5$$

سوالات عملی (۵۰ نمره)

مسئله‌ی ۵. (۲۰ نمره)

در این سوال می‌خواهیم بااستفاده از شبکه LSTM یک دسته‌بندی برروی دیتاست Yelp انجام دهیم. نوت‌بوک Q۵ را باز کرده و سلول‌های حاضر را اجرا کرده تا داده‌ی آموزش و اعتبارسنجی شما آماده شود. توجه داشته باشید که باید به عنوان ورودی کلمات به شبکه از بردارهای از پیش آموزش دیده Glove استفاده کنید. پس از آموزش مقدار امتیاز f۱ داده‌های اعتبارسنجی را برای هر epoch رسم کنید.
در طراحی شبکه و ابرپارامترهای آن آزاد هستید.

مسئله‌ی ۶. (۲۵+۵ نمره)

در این تمرین هدف پیاده‌سازی دو شبکه LSTM و GRU و پیش‌بینی بازار سهام بوسیله آن‌ها می‌باشد. به نوتبوک Q۶ مراجعه شود.