



تمرین سری سوم

یادگیری ژرف

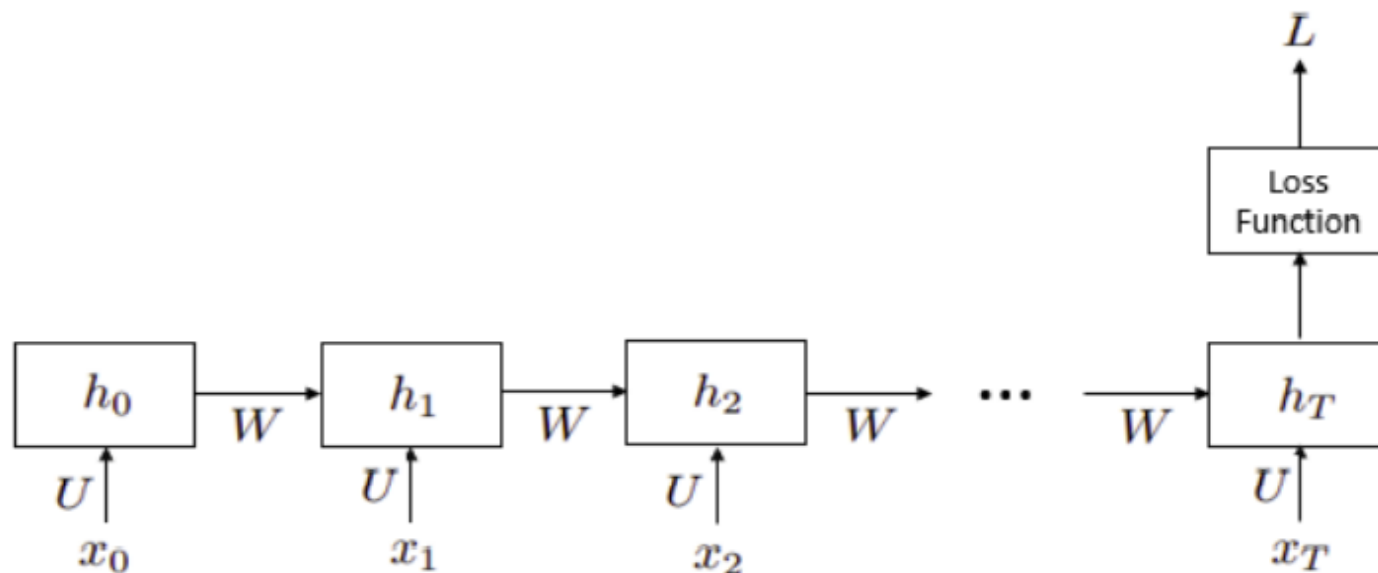
امیر حسین محمدی

۹۹۲۰۱۰۸۱

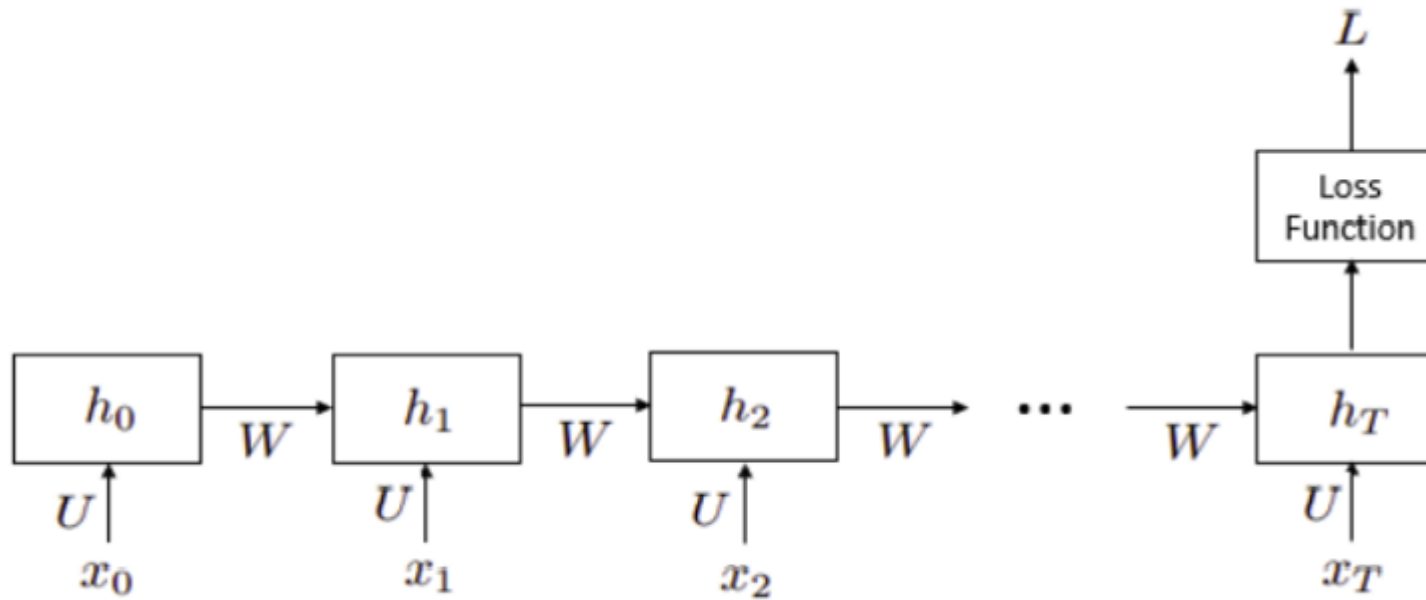


مسئله ۱. (۱۵+۵ نمره)

(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. (۱ ≤ t ≤ T - ۱) (۳ نمره)



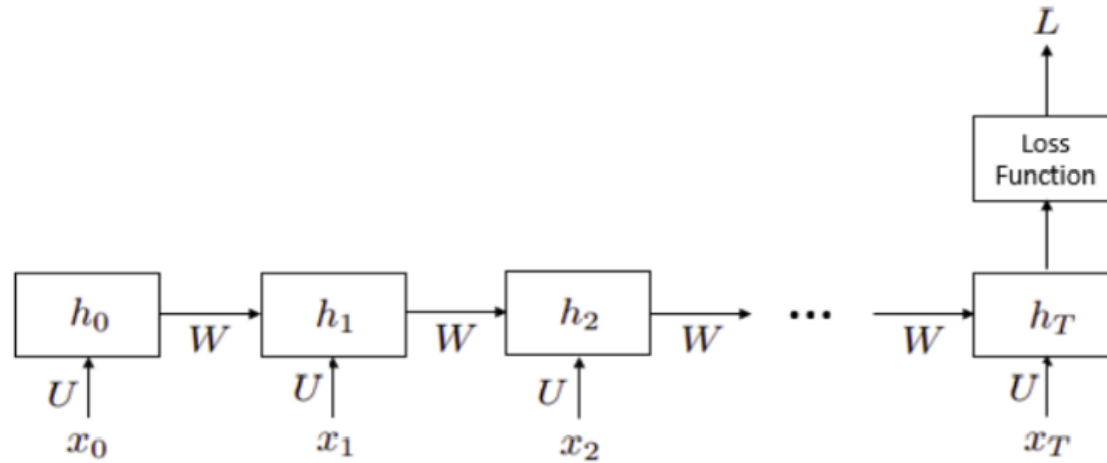
ابتدا فاز Forward را می نویسیم:

$$h_{0-out} = Ux_0 \quad \longrightarrow \quad h_0 = \sigma(Ux_0)$$

$$h_{1-out} = W[\sigma(Ux_0)] + Ux_1 \quad \longrightarrow \quad h_1 = \sigma(W[\sigma(Ux_0)] + Ux_1)$$

$$h_{2-out} = W[\sigma(W[\sigma(Ux_0)] + Ux_1)] + Ux_2 \quad \longrightarrow \quad h_2 = \sigma(W[\sigma(W[\sigma(Ux_0)] + Ux_1)] + Ux_2)$$

$$h_{T-out} = W[\sigma(W h_{t-1})] + Ux_T \quad \longrightarrow \quad h_T = \sigma(W[\sigma(W h_{t-1})] + Ux_T)$$



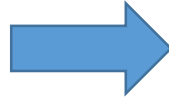
$$\diamond h_{T-out} = W[\sigma(W h_{t-1})] + U x_T \quad \longrightarrow \quad h_T = \sigma(W[\sigma(W h_{t-1})] + U x_T)$$

❖ حال بر اساس رابطه ی بدست آمده داریم برای زمان $t+1$:

$$\diamond h_{t+1-out} = h_t * W + U x_{t+1} \quad \longrightarrow \quad \diamond h_{t+1} = \sigma(h_{t+1-out})$$

$$\diamond \frac{\partial L}{\partial h_t} = \frac{\partial L}{\partial h_{t+1}} * \frac{\partial h_{t+1}}{\partial h_{t+1-out}} * \frac{\partial h_{t+1-out}}{\partial h_t} \quad \longrightarrow \quad \left\{ \begin{array}{l} \diamond \frac{\partial h_{t+1}}{\partial h_{t+1-out}} = \sigma(h_{t+1-out}) * (1 - \sigma(h_{t+1-out})) \\ \diamond \frac{\partial h_{t+1-out}}{\partial h_t} = W \end{array} \right.$$

$$\diamond \frac{\partial L}{\partial h_t} = \frac{\partial L}{\partial h_{t+1}} * \frac{\partial h_{t+1}}{\partial h_{t+1-out}} * \frac{\partial h_{t+1-out}}{\partial h_t}$$



$$\begin{cases} \diamond \frac{\partial h_{t+1}}{\partial h_{t+1-out}} = \sigma(h_{t+1-out}) * (1 - \sigma(h_{t+1-out})) \\ \diamond \frac{\partial h_{t+1-out}}{\partial h_t} = W \end{cases}$$



$$\frac{\partial L}{\partial h_t} = \frac{\partial L}{\partial h_{t+1}} * \sigma(h_{t+1-out}) * (1 - \sigma(h_{t+1-out})) * W = \frac{\partial L}{\partial h_{t+1}} * \sigma(h_t * W + Ux_{t+1}) * (1 - \sigma(h_t * W + Ux_{t+1})) * W$$

با توجه به فرضیات مسئله یک مقدار عددی است و آن را مثلا لاندا در نظر می گیریم

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



(ب) حال از رابطه قسمت قبل استفاده کرده و به شکل زنجیر وار گرادیان h را بر حسب گرادیان h_T بنویسید. (۲ نمره)



با توجه به قسمت الف داریم:

$$\frac{\partial L}{\partial h_t} = \frac{\partial L}{\partial h_{t+1}} * \sigma(h_t * W + Ux_{t+1}) * (1 - \sigma(h_t * W + Ux_{t+1})) * W$$

بنابراین با توجه به این قسمت دیدیم که گرادیان زمان t را می توان بر اساس گرادیان زمان $t+1$ نوشت با یک ضربی که از شامل یک یک سگمیوید و یک منهای سگمیوید است و یک W وزن های شبکه است. بنابراین بر همین اساس می توان گرادیان h_0 را بر حسب h_1 و گرادیان h_1 را بر حسب h_2 و به هم ترتیب تا h_T نوشت، بنابراین داریم:

$$\frac{\partial L}{\partial h_0} = \lambda * \frac{\partial L}{\partial h_1} \quad \longrightarrow \quad \lambda = W * \sigma(h_0 * W + Ux_1) * (1 - \sigma(h_0 * W + Ux_1))$$

$$\frac{\partial L}{\partial h_1} = \lambda * \frac{\partial L}{\partial h_2} \quad \longrightarrow \quad \lambda = W * \sigma(h_1 * W + Ux_2) * (1 - \sigma(h_1 * W + Ux_2))$$

$$\frac{\partial L}{\partial h_2} = \lambda * \frac{\partial L}{\partial h_3} \quad \longrightarrow \quad \lambda = W * \sigma(h_2 * W + Ux_3) * (1 - \sigma(h_2 * W + Ux_3))$$

⋮
⋮
⋮

$$\frac{\partial L}{\partial h_{T-1}} = \lambda * \frac{\partial L}{\partial h_T} \quad \longrightarrow \quad \lambda = W * \sigma(h_{T-1} * W + Ux_T) * (1 - \sigma(h_{T-1} * W + Ux_T))$$



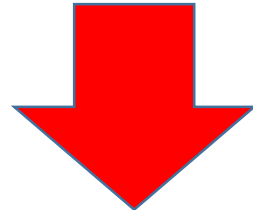
$$\frac{\partial L}{\partial h_0} = \lambda * \frac{\partial L}{\partial h_1} \Rightarrow \lambda = W * \sigma(h_0 * W + U_{x_1}) * (1 - \sigma(h_0 * W + U_{x_1}))$$

$$\frac{\partial L}{\partial h_1} = \lambda * \frac{\partial L}{\partial h_2} \Rightarrow \lambda = W * \sigma(h_1 * W + U_{x_2}) * (1 - \sigma(h_1 * W + U_{x_2}))$$

$$\frac{\partial L}{\partial h_2} = \lambda * \frac{\partial L}{\partial h_3} \Rightarrow \lambda = W * \sigma(h_2 * W + U_{x_3}) * (1 - \sigma(h_2 * W + U_{x_3}))$$

⋮

$$\frac{\partial L}{\partial h_{T-1}} = \lambda * \frac{\partial L}{\partial h_T} \Rightarrow \lambda = W * \sigma(h_{T-1} * W + U_{x_T}) * (1 - \sigma(h_{T-1} * W + U_{x_T}))$$



❖ همانطور که گفته شد ما می توانیم گرادیان h_0 را بر اساس گرادیان h_1 و h_1 را بر حسب h_2 و الی آخر بنویسیم. بنابراین ما می توانیم گرادیان h_0 را بر اساس h_T یک قاعده ی زنجیری بنویسیم.

می توان کل این پارامتر یک مقدار مثل η در نظر گرفت.

$$\frac{\partial L}{\partial h_0} = \lambda * \frac{\partial L}{\partial h_T} \Rightarrow \lambda = W^T * \prod_{i=0}^T \sigma(h_i * W + U_{x_{i+1}}) * (1 - \sigma(h_i * W + U_{x_{i+1}}))$$

$$\Rightarrow \lambda = W^T * \eta$$



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

(آ) یکی از روش‌های مهم جلوگیری از محوشدگی و انفجار گرادیان مقداردهی اولیه صحیح وزن‌های 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 پیدا کنید.) (۵ نمره)



$$\frac{\partial L}{\partial h_0} = \lambda * \frac{\partial L}{\partial h_T} \quad \rightarrow \quad \lambda = W^T * \eta$$

$$\rightarrow \frac{\partial L}{\partial h_0} = (W^T * \eta) * \frac{\partial L}{\partial h_T} \rightarrow$$

همانطور که مشاهده می شود در پس انتشار h_0 یک یک ضرب گه شامل وزن ها به توان T ضرب در η که خود حاصل چندین ضرب است ظاهر می شود. این دو ضرب می توانند سبب مسئله ی محوشدگی گرادیان و یا انفجار گرادیان شوند. اما برای انفجار از بین این دو ضرب، ضرب W^T تاثیر بیشتری دارد زیرا مقادیر سیگموید و یک منهای سیگموید همواره بین صفر و یک است و همچنین حاصل ضرب آنها هم بین صفر یک است بنابراین η که خود برابر حاصل ضرب چندین عبارت سیگمویدی است بین صفر و یک باقی می ماند پس عملا باعث انفجار مارا نمی کند. از طرفی طبق خواسته ی مسئله گفته شده در ابتدای شبکه می خواهیم بدون در نظر گرفت x مقدار دهی کنیم. بنابراین کلا از η صرف نظر می کنیم. بنابراین داریم:

یک عدد است و می توان از آن چشم پوشی کرد:

$$\diamond \frac{\partial L}{\partial h_0} = (W^T) \frac{\partial L}{\partial h_T}$$

❖ بنابراین اگر بخواهیم یک حد بالا مثلا θ را به عنوان آستانه برای وقوع انفجار گرادیان در نظر بگیریم آنگاه بهتر است وزن اولیه w کوچکتر مساوی رادیکال θ با فرجه T باشد.

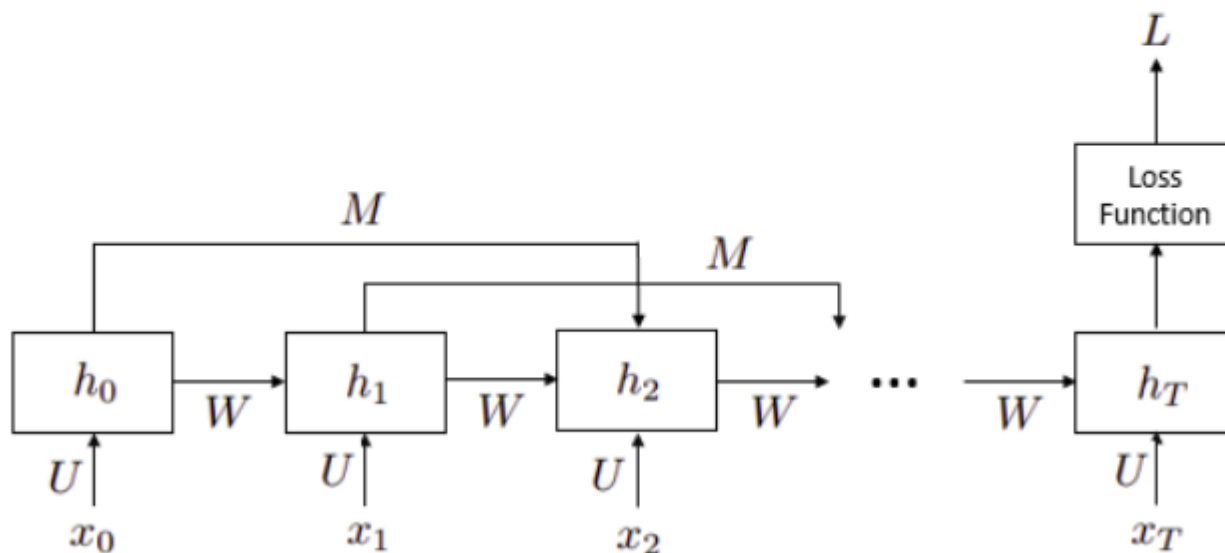
$$\frac{\partial L}{\partial h_0} = W^T$$

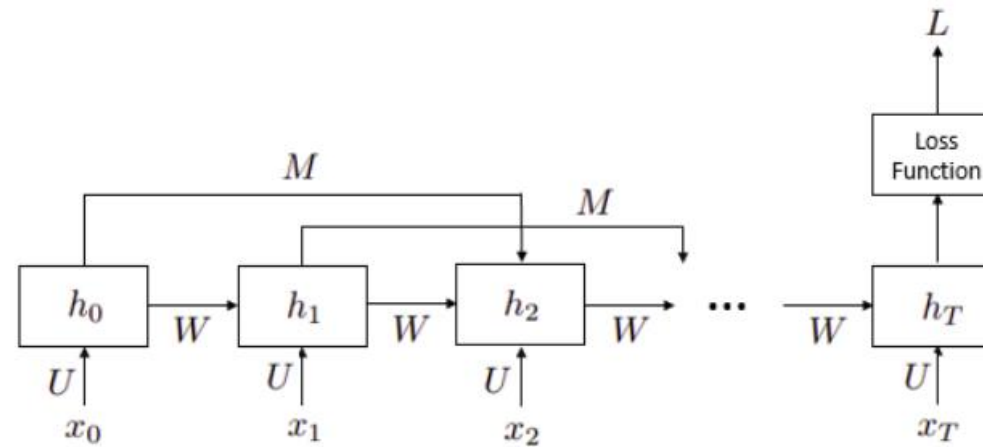
بنابراین کل چیزی که باعث مسئله ی انفجار می شود W^T است. بنابراین اگر می خواهیم از انفجار جلوگیری کنیم با مقدار w را درست انتخاب کنیم زیرا اگر مثلا مقدار اولیه w ۱.۱ باشد و $T=51$ آنگاه مقدار W^T برابر می شود با 117.4

$$w \leq \sqrt[T]{\theta}$$



(ب) یکی از راه‌های جلوگیری از محوشدگی گرادیان استفاده از skip-connection ها است. شکل زیر را در نظر بگیرید که در آن 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 می‌شود. (۱ ≤ t ≤ $T - 2$) (۵ نمره)





ابتدا مرحله ی Forward path را در نظر می گیریم:

$$\diamond h_{0-out} = x_0 U \quad \Rightarrow \quad h_0 = \sigma(x_0 U)$$

$$\diamond h_{1-out} = W[\sigma(x_0 U)] + Ux_1 \quad \Rightarrow \quad h_1 = \sigma(W[\sigma(x_0 U)] + Ux_1)$$

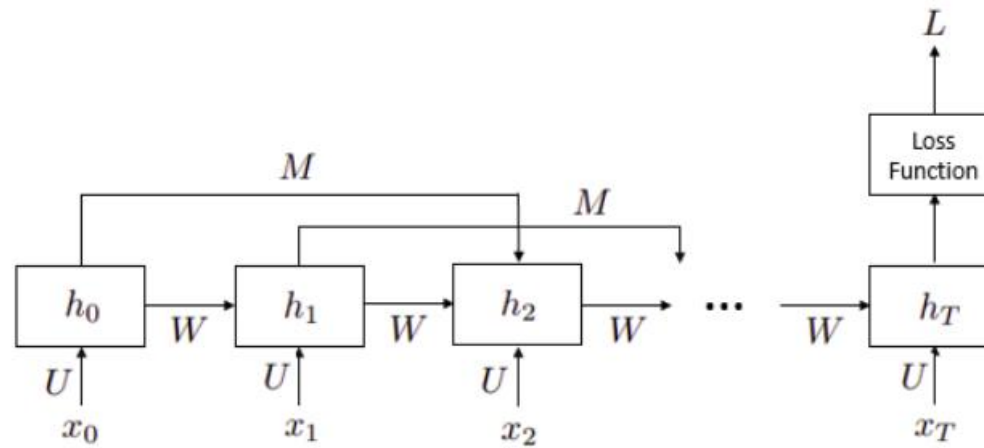
$$\diamond h_{2-out} = W[\sigma(W[\sigma(x_0 U)] + Ux_1)] + Ux_2 + M * h_0 \quad \Rightarrow \quad h_2 = \sigma(W[\sigma(W[\sigma(x_0 U)] + Ux_1)] + Ux_2 + M * h_0)$$

$h_0 = \sigma(x_0 U)$

\vdots

$$\diamond h_{T-out} = Wh_{T-1} + Mh_{T-2} + Ux_T ,$$

$$\diamond h_T = \sigma(Wh_{T-1} + Mh_{T-2} + Ux_T)$$



$$\diamond h_{T-out} = Wh_{T-1} + Mh_{T-2} + Ux_T,$$



$$\diamond h_T = \sigma(Wh_{T-1} + Mh_{T-2} + Ux_T)$$

نوشتن مرحله ی پس انتشار به صورت فرم عمومی:

$$\diamond h_{t+2-out} = Wh_{t+1} + Mh_t + Ux_{t+2},$$

$$h_{t+1-out} = Wh_t + Mh_{t-1} + Ux_{t+1}$$

$$\diamond h_{t+2} = \sigma(Wh_{t+1} + Mh_t + Ux_{t+2}),$$

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

محاسبه ی مرحله ی backward را در نظر می گیریم:

$$\diamond \frac{\partial L}{\partial h_t} = \frac{\partial L}{\partial h_{t+2}} \times \frac{\partial h_{t+2}}{\partial h_{t+2-out}} \times \frac{\partial h_{t+2-out}}{\partial h_t}$$



$$\diamond \frac{\partial h_{t+2}}{\partial h_{t+2-out}} = \sigma(h_{t+2-out})(1 - \sigma(h_{t+2-out}))$$

$$\diamond \frac{\partial h_{t+2-out}}{\partial h_t} = M + \frac{\partial h_{t+2-out}}{\partial h_{t+1}} \times \frac{\partial h_{t+1}}{\partial h_{t+1-out}} \times \frac{\partial h_{t+1-out}}{\partial h_t}$$



$$\diamond \frac{\partial L}{\partial h_t} = \frac{\partial L}{\partial h_{t+2}} \times \frac{\partial h_{t+2}}{\partial h_{t+2-out}} \times \frac{\partial h_{t+2-out}}{\partial h_t}$$

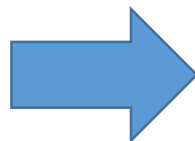


$$\diamond \frac{\partial h_{t+2}}{\partial h_{t+2-out}} = \sigma(h_{t+2-out})(1 - \sigma(h_{t+2-out}))$$

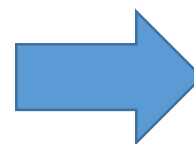
$$\diamond \frac{\partial h_{t+2-out}}{\partial h_t} = M + \frac{\partial h_{t+2-out}}{\partial h_{t+1}} \times \frac{\partial h_{t+1}}{\partial h_{t+1-out}} \times \frac{\partial h_{t+1-out}}{\partial h_t}$$

$$\diamond \frac{\partial h_{t+1}}{\partial h_{t+1-out}} = \sigma(h_{t+1-out})(1 - \sigma(h_{t+1-out}))$$

$$\diamond \frac{\partial h_{t+1-out}}{\partial h_t} = W$$



$$\diamond \frac{\partial L}{\partial h_t} = \frac{\partial L}{\partial h_{t+2}} \times \frac{\partial h_{t+2}}{\partial h_{t+2-out}} \times \left(M + \frac{\partial h_{t+2-out}}{\partial h_{t+1}} \times \frac{\partial h_{t+1}}{\partial h_{t+1-out}} \times \frac{\partial h_{t+1-out}}{\partial h_t} \right)$$



$$\diamond \frac{\partial L}{\partial h_t} = \underbrace{\frac{\partial L}{\partial h_{t+2}} \times \frac{\partial h_{t+2}}{\partial h_{t+2-out}}}_{\alpha} \times M + \underbrace{\frac{\partial L}{\partial h_{t+2}} \times \frac{\partial h_{t+2}}{\partial h_{t+2-out}} \times \frac{\partial h_{t+2-out}}{\partial h_{t+1}}}_{\frac{\partial L}{\partial h_{t+1}}} \times \underbrace{\frac{\partial h_{t+1}}{\partial h_{t+1-out}} \times \frac{\partial h_{t+1-out}}{\partial h_t}}_{\lambda}$$



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

α

$\frac{\partial L}{\partial h_{t+1}}$

λ



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



یکی از راه های جلوگیری از محو شدن گرادیان استفاده skip connection است. در این روش ما با استفاده از ارتباط های هایی که بین لایه های ایجاد می کنیم به صورت مستقیم به مسیرهای مختلف (لایه های مختلف می توانیم اطلاعات را جابه جا کنیم). همانطور که در بخش های بحث شد یکی از مشکلاتی که در شبکه های بازگشتی وجود دارد این است که برای عملیات BackProb ما باید گرادیان را در لایه های متعددی به عقب منتشر کنیم و در این حالت به دلیل ضرب های زیادی که بین یک سری W و خروجی سیگموید و یک منهای سیگموید وجود داشت (که همگی بین صفر و یک هستند) ما دچار محو شدگی گرادیان (Vanishing Gradient) می شدیم.



ضرایب

$$\frac{\partial L}{\partial h_t} = \lambda * \frac{\partial L}{\partial h_T}$$



$$\lambda = W^{T-t} * \prod_{i=t}^T \sigma(h_i * W + Ux_{i+1}) * (1 - \sigma(h_i * W + Ux_{i+1}))$$



$$\lambda = W^T * \eta$$

حال با استفاده از skip connection ما یک مسیر دیگر نیز داریم که بدون این ضرایب را داشته باشد مستقیماً به گرادیان به عقب منتشر می شود. بنابراین ما می توانیم به نوعی با این مسئله ی محو شدگی گرادیان مقابله کنیم زیرا این بار علاوه بر گرادیان هایی که از مسیر اصلی منتقل می شوند یک گرادیان دیگر که از یک مسیر دیگر منتقل می شود و از کانکشن هایی ایجاد شده است که به صورت مستقیم گرادیان را به جای ضرب های متوالی و پی در پی مستقیماً (با محاسباتی که تا حد زیادی موجب محو شدگی گرادیان نمی شود) به عقب منتقل می کند.



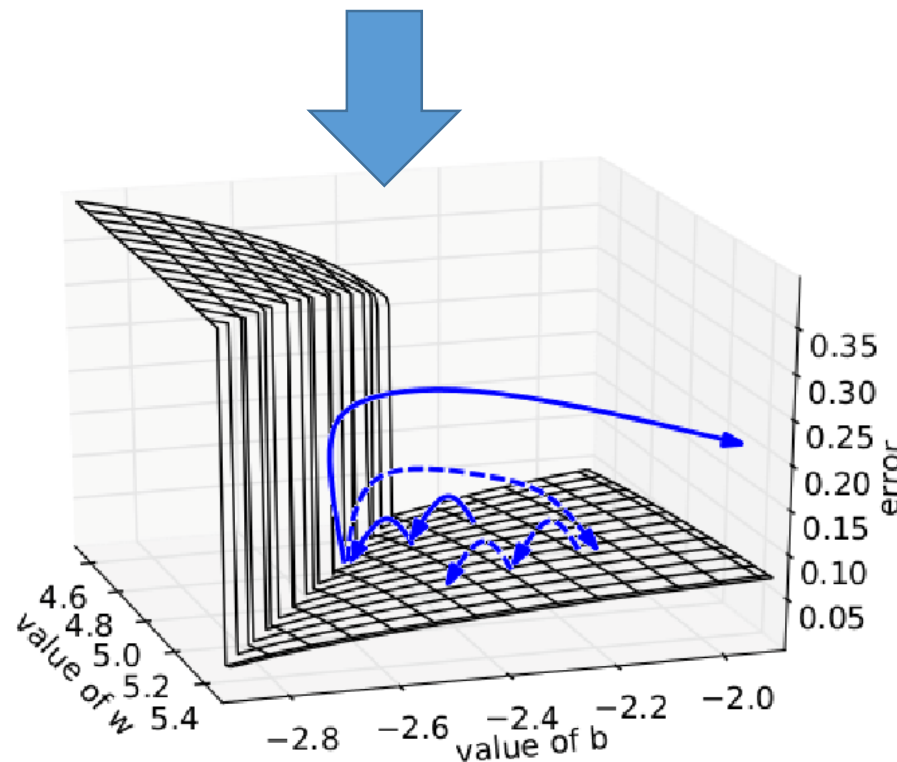
(ج) یکی از راه حل های جلوگیری از انفجار گرادیان، برش گرادیان^۱ است که این خودبه دو زیرراه حل برش توسط مقدار^۲ و برش توسط اندازه^۳ تقسیم می شود. این دو را جداگانه توضیح دهید. برتری برش توسط اندازه را به برش توسط مقدار را توضیح دهید. (۵ نمره امتیازی)



❖ همانطور که گفته شد یکی از مشکلات موجود بر سر راه الگوریتم پس انتشار مشکل انفجار گرادیان و ناپدید شدن گرادیان است همانطور که در شکل زیر می بینیم. یکی از راه های جلوگیری از انفجار گرادیان استفاده از Gradient clipping یا برش گرادیان است در واقع ما در این روش می خواهیم از وزن هایی که بیش از اندازه بزرگ می شود و سبب بروز مشکل انفجار گرادیان می شود جلوگیری کنیم. Gradient Clipping شامل دو روش است که عبارت اند از:

❖ Gradient Clipping by Value

❖ Gradient Clipping by norm

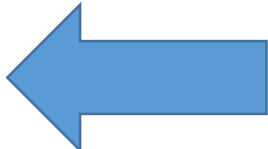




❖ Gradient Clipping by Value

❖ یکی از روش ها برای جلوگیری از انفجار گرادیان استفاده از روش **Gradient Clipping by Value** طبق این روش ما یک مقدار به عنوان آستانه به عنوان ماکسیمم مقدار در نظر میگیریم. اگر مقدار (اندازه) گرادیان از این آستانه بیشتر شد مقدار آن آستانه را به جای مقدار گرادیان قرار می دهیم و اگر هم کمتر شد هیچ تغییری انجام نمی شود. همانطور که در زیر میبینیم:

$$Gradient = \frac{\partial L}{\partial \theta}$$

اگر : $\|Gradient\| \geq Treshhold$ 

$$Gradient = Treshhold$$

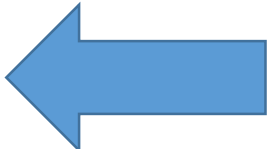
❖ بنابراین با استفاده از این روش از بزرگ شدن گرادیان ها که منجر به پدیده ی انفجار می شود جلوگیری می کنیم. البته ورژن های مختلفی از این الگوریتم وجود دارد مثلاً می توان علاوه بر مقدار آستانه (ماکسیمم) یک مقدار مینیمم نیز در نظر گرفت.




❖ Gradient Clipping by norm

❖ یکی از روش ها برای جلوگیری از انفجار گرادیان استفاده از روش **Gradient Clipping by Norm** طبق این روش ما یک مقدار به عنوان آستانه به عنوان ماکسیمم مقدار در نظر میگیریم. اگر مقدار گرادیان از این آستانه بیشتر شد مقدار آن آستانه ضرب در گرادیان تقسیم بر اندازه گرادیان (بردار واحد Unit vector) را به جای مقدار گرادیان قرار می دهیم و اگر هم کمتر شد هیچ تغییری انجام نمی شود. همانطور که در زیر میبینیم:

$$Gradient = \frac{\partial L}{\partial \theta}$$

اگر : $\|Gradient\| \geq Treshhold$ 

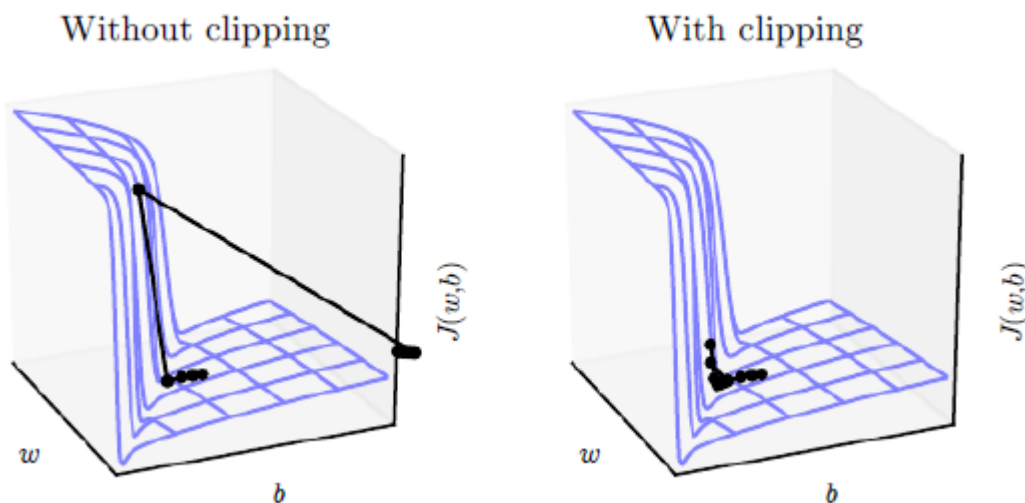
$$Gradient = Treshhold * \frac{Gradient}{\|Gradient\|}$$

 Unit vector

❖ بنابراین با استفاده از این روش از بزرگ شدن گرادیان ها که منجر به پدیده ی انفجار می شود جلوگیری می کنیم. البته لازم به ذکر است که ورژن های مختلفی از این الگوریتم وجود دارد مثلاً می توان مقدار آستانه را در بردار واحد ضرب نکرد.



نکته ی قابل توجه این است که هر دو روش ارائه شده Gradient Clipping by Value و Gradient Clipping by norm تا حد زیادی از انفجار گرادیان جلوگیری می کنند. اما یکی از مشکلات Gradient Clipping by Value امکان تغییر راستای گرادیان است (تغییر شیب است). در واقع وقتی ما component های مختلفی داشتیم و از روش برش بر اساس مقدار استفاده کنیم ممکن است یکی از component ها تغییر کرده و مثلاً نصف شود و دیگری نه بنابراین بر این اساس شیب و راستای گرادیان ما پس از برش ممکن است نسبت به حالت اصلی تفاوت کند. این در حالی است که در روش Gradient Clipping by norm هر یک از component ها و محورها چون بر نرم تقسیم می شود به یکی اندازه کاهش می یابند و این مسئله سبب می شود که راستا و شیب گرادیان تغییر نکند. بنابراین روش Gradient Clipping by norm از این جهت نسبت به Gradient Clipping by Value برتری دارد.





مسئله ۲. (۲۵+۱۰ نمره)

در این مسئله می‌خواهیم با مفاهیمی در تولید دنباله در شبکه های Seq2Seq و مزایا و معایب آن‌ها آشنا شویم. (بخش ۱) در بخش اول می‌خواهیم مفهوم teacher forcing را بررسی کنیم. برای تولید دنباله ما می‌توانیم یک استراتژی خام اولیه در نظر بگیریم، می‌توان برای تولید نشانه $t+1$ توسط رمزگشای t ، نشانه تولید شده توسط شبکه در زمان t را به عنوان ورودی به دیکودر زمان $t+1$ بدهیم اما این حالت مشکلاتی دارد.

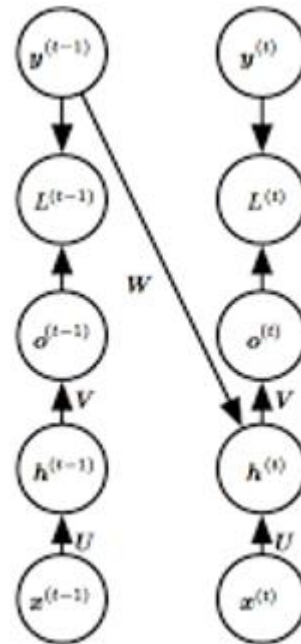
(آ) ابتدا توضیح دهید این مشکلات چه چیزهایی هستند و سپس روش teacher forcing را توضیح داده و بگویید که teacher forcing چگونه این مشکلات را برطرف می‌کند. (۵ نمره)



در مسائل مرتبط با شبکه های بازگشتی گاهی نیاز داریم تا خروجی پیش بینی شده در هر مرحله را به عنوان ورودی به مرحله ی قبل ارائه کنیم. این مسئله در مسائلی همچون Machine Translation، Text Summarization و Caption Generation وجود دارد. اما این مسائل همواره با مشکلاتی روبه رو هستند. یکی از مهمترین آنها پیچیدگی محاسباتی آنهاست زیرا ما باید به ازای هر ورودی مرحله ی forward و سپس backward را حساب کنیم که وقتی در مسئله ی ترجمه ی ماشینی هستیم این مسئله وقتی طول متن و تعداد کلمات بسیار زیاد است بسیار هزینه بر و طولانی و مسلا برای ترجمه ی یک کتاب عملا غیر ممکن می شود همچنین بسیار آهسته همگرا می شوند زیرا همانطور که گفته شد ما به ازای هر زمان باید عملیات پس انتشار را برای همه ی زمان های قبل از آن محاسبه کنیم. از طرفی مشکل دیگری که وجود دارد وقتی ورودی هر مرحله به خروجی پیش بینی شده در مرحله ی قبل وابسته است و چون در مرحله و لایه های اولیه ی شبکه ی بازگشتی احتمال بروز خطا بسیار است این مسئله سبب می شود تا زمانی که ما در لایه های اولیه خروجی اشتباه داریم در لایه های بعدی این خطا و اشتباه انتشار پیدا کند و به طور کامل شبکه ی ما به اشتباه کار کند بنابراین مدل ناپایدار است. همچنین این مدل ها معمولا تعمیم پذیر کمی دارد.



روش teacher forcing روشی است که تا حدی از مشکلات مطرح شده در اسلاید قبل جلوگیری می کند. این روش مبتنی بر Maximum likelihood estimation است. ایده ی این روش بر این اساس که به جای استفاده خروجی های پیش بنی شده در لایه ی قبل از خروجی های واقعی و مورد انتظار در لایه ی قبل استفاده می کند همانطور که در شکل زیر می بینید. این مسئله چند ویژگی بسیار مهم دارد و اولین مسئله این است از مشکل backpropagation true time که پیچیدگی محاسباتی زیادی تولید می کرد جلوگیری می کند زیرا همانطور که گفته شد به جای استفاده از خروجی پیش بینی شده در مرحله ی قبل از خروجی واقعی استفاده می کند بنابراین دیگر گرادیان ندارد و کل گراف محاسباتی ما یک گام محاسبه ی گرادیان دارد. همچنین به دلیل اینکه ما عینا از خروجی واقعی لایه ی قبل برای لایه ی بعد استفاده می کنیم بنابراین مسئله ی بروز خطا و ناپداری مدل به نوعی از بین می رود.



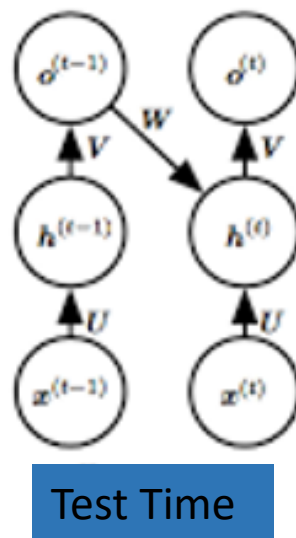
Train Time



(ب) مشکل اصلی teacher forcing موضوعی به نام exposure bias است. این مشکل را توضیح دهید. (۵ نمره)



همانطور که گفته شد ایده ی اصلی روش teacher forcing استفاده از خروجی های واقعی (Ground Truth) مرحله ی قبل به عنوان ورودی مرحله ی بعد است و این مسئله تا حدی از ناپایداری مدل جلوگیری می کند. اما مسئله ای که وجود دارد در مرحله ی test و inference عملاً ما به خروجی های واقعی داده ها دسترسی نداریم تا بتوانیم از آنها برای پیش بینی خروجی مدل استفاده کنیم. بنابراین در این حالت مدل آموزش دیده باید کاملاً بر اساس اطلاعات پیش بینی شده توسط خودش عمل کند و اگر مثلاً در مرحله ی $t-1$ یک خروجی را پیش بینی کند که مناسب نباشد این خروجی در تمامی زمان های بعد از $t-1$ منتشر می شود و می تواند سبب تولید خروجی های اشتباه شود. به این مشکل exposure bias میگویند.





(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 می‌شود. (۵ نمره)



(Scheduled Sampling, Bengio et al.)

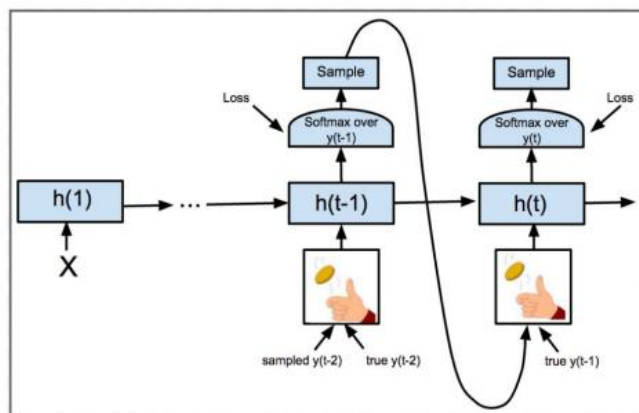


Figure 1: Illustration of the Scheduled Sampling approach, where one flips a coin at every time step to decide to use the true previous token or one sampled from the model itself.

همانطور که گفته شد اما مسئله ای که **teacher forcing** جود دارد این است که در مرحله ی **test** و **inference** عملا ما به خروجی های واقعی داده ها دسترسی نداریم تا بتوانیم از آنها برای پیش بینی خروجی مدل استفاده کنیم. بنابراین در این حالت مدل آموزش دیده باید کاملا بر اساس اطلاعات پیش بینی شده توسط خودش عمل کند و اگر مثلا در مرحله ی $t-1$ یک خروجی را پیش بینی کند که مناسب نباشد این خروجی در تمامی زمان های بعد از $t-1$ منتشر می شود و می تواند سبب تولید خروجی های اشتباه شود. برای حل مشکل **exposure bias** ما باید وابستگی به خروجی های واقعی (**ground truth**) کاهش دهیم. برای این کار از یک روش به اسم **Scheduled sampling** استفاده می شود. به این صورت که به جای اینکه خروجی های واقعی (**ground truth**) مرحله ی قبل به عنوان ورودی های این مرحله بعد داده شود، در هر مرحله به صورت رندم یا بر اساس یک احتمالی تصمیم بگیریم که خروجی واقعی مرحله ی قبل یا خروجی پیش بینی شده توسط مرحله ی قبل را به عنوان خروجی به مرحله ی بعد بدهیم. این مسئله تا حد زیادی متکی بودن به خروجی های واقعی را در این مسائل از بین می برد بنابراین از این طریق تا حدی با مشکل **exposure bias** جلوگیری می کنیم زیرا دقیقا مشکل **exposure bias** به این علت بود که مدل فقط بر اساس خروجی های واقعی آموزش دیده و وقتی بر اساس **Scheduled sampling** مدل آموزش می بیند به دلیل اینکه به صورتی ترکیبی از خروجی های واقعی و خروجی های واقعی به مرحله بعد داده می شود بنابراین مشکل **exposure bias** کنترل می شود.



(بخش ۲) حال در بخش دوم مسئله می‌خواهیم بر روی الگوریتم جستجوی موجی^۶ تمرکز کنیم. این الگوریتم در تقابل با الگوریتم حریصانه برای تولید دنباله در زمان رمزگشایی مطرح می‌شود.

(آ) ابتدا تفاوت دو الگوریتم جستجوی موجی و الگوریتم حریصانه برای تولید دنباله را بیان کنید. (۵ نمره)

Greedy Search

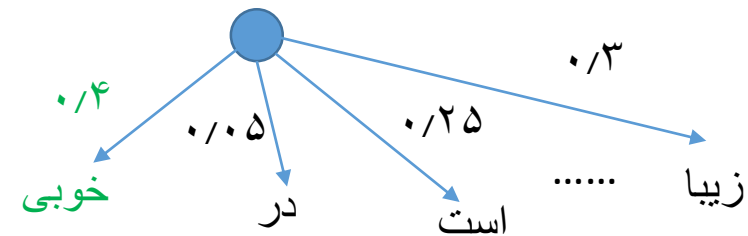
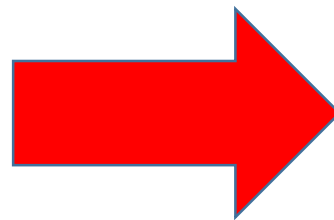
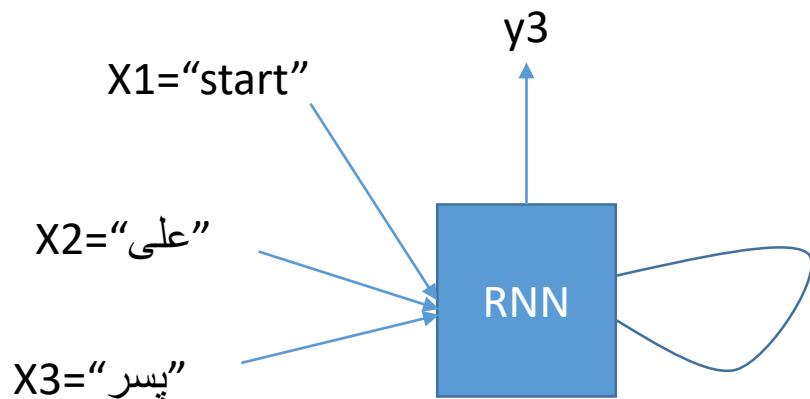


جستجوی حریصانه بر این اساس است که مثلاً ما وقتی می‌خواهیم یک دنباله‌ی متنی را (مثلاً جمله‌ی: "علی پسر...") به یک شبکه بازگشتی بدهیم همانطور که در شکل زیر می‌بینیم و سپس مثلاً ادامه‌ی آن جمله (اول کلمه بعد از پسر) را پیش‌بینی کنیم. پس از اینکه تمامی آن جمله و کلمات به عنوان ورودی به آن شبکه داده شد یک درخت به عمق یک و تمامی حالت‌هایی (کلماتی) که می‌تواند بعد از "پسر" قرار بگیرد به وجود می‌آید. برای هر یک از این کلمات بر اساس احتمال شرطی عددی اختصاص داده می‌شود. بنابراین باید یک کاندید برای کلمه‌ی بعد از پسر انتخاب شود. طبق الگوریتم حریصانه خروجی بین‌ی شده با بیشترین احتمال شرطی به عنوان کلمه بعد از کلمه "پسر" انتخاب کنیم. مثلاً در این مثال کلمه‌ی "خوبی" انتخاب می‌شود.

$$y_{t'} = \operatorname{argmax}_{y \in \mathcal{Y}} P(y \mid y_1, \dots, y_{t'-1}, \mathbf{c}),$$



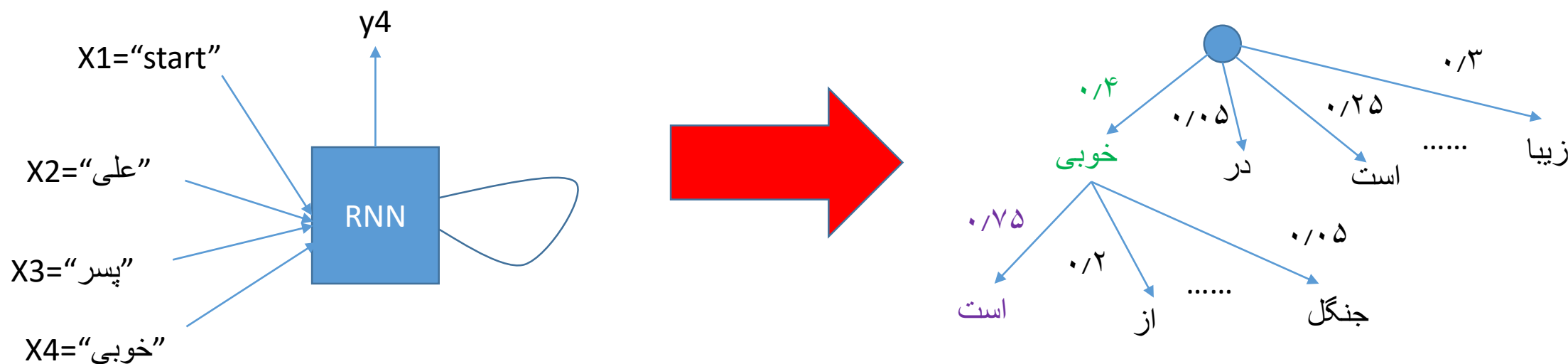
$$\prod_{t'=1}^{T'} P(y_{t'} \mid y_1, \dots, y_{t'-1}, \mathbf{c}).$$



Greedy Search



پس از پیش بینی کلمه ی خوبی این بار به ورودی های RNN "علی پسر خوبی..." است داده می شود. و عمق دو درخت ایجاد می شود و از بین کلمات ممکن اونی که بیشترین احتمال شرطی را دارد انتخاب می شود. و این مسئله همینطور برای کلمات بعدی نیز تکرار می شود تا زمانی که کلمه ی "eos" که نشان دهنده ی پایان جمله است انتخاب شود. این الگوریتم تضمین نمی کند که به جواب بهینه برسد.

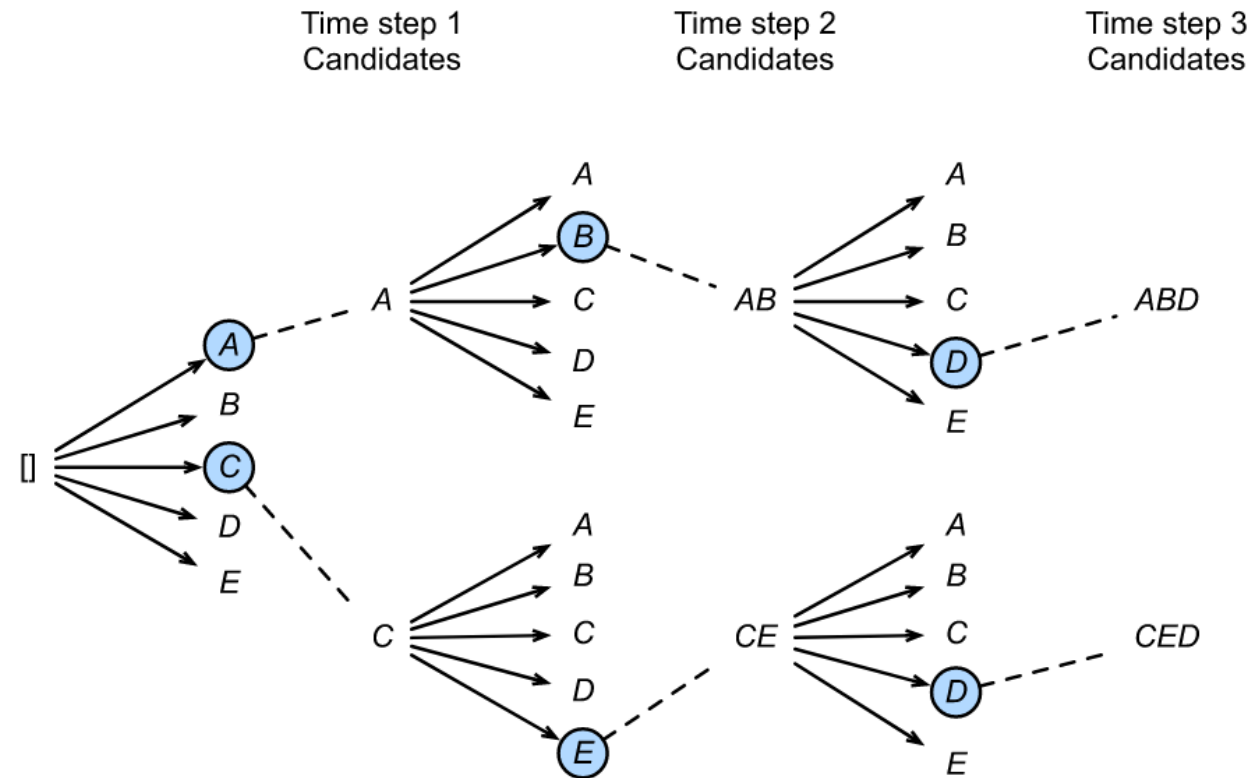




همانطور که گفته شد یکی از مشکلات جستجوی حریصانه نرسیدن به جواب بهینه است و این احتمال تقریباً زیاد است. دلیل این مسئله این است که ممکن است چون ما حریصانه از همان ابتدا در هر عمق درخت کلمه با بالاترین احتمال شرطی را انتخاب می‌کنیم و در مراحل ابتدایی که تقریباً دانش زیادی نداریم کلمه‌ی با احتمال بالایی را انتخاب کنیم اما زیر درخت‌ها مربوط به آن کلمه احتمال کمتری داشته باشند و در مجموع به جواب مطلوبی برای کامل‌تر کردن جمله نرسیم. این در صورتی است که اگر در عمق اول مثلاً یکی دومین کلمه با بالاترین احتمال را در نظر می‌گرفتیم آن‌گاه به جواب مطلوب‌تری می‌رسیدیم. برای همین جستجوی موجی بر این اساس است که در هر موقع k تا از کلمات با بالاترین احتمال‌ها را در نظر بگیریم و زیر درخت‌های آنها را بر اساس همین رویکرد ادامه داده و در هر عمق k تا از بالاترین احتمال‌ها تا آن کلمه را انتخاب کنیم (در واقع احتمالات مسیر در هم ضرب می‌شوند) و اینطوری تا حدی می‌توانیم از این مشکل جلوگیری کنیم. اما مشکلی در این حالت به وجود می‌آید هرچه تعداد کلمات و عمق بیشتر شود این احتمال کل (ضرب احتمال‌ها در هم) کوچکتر می‌شود. برای همین لگاریتم احتمال‌ها در طول مسیر ایجاد هر کلمه باهم جمع می‌شود تا از بروز چنین مشکلی جلوگیری شود.

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

Beam Search





(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. (۵ نمره امتیازی)



همانطور که گفته شد در beam Search ما در هر عمق درخت از بین کلمات موجود k تا از کلمات با بیشترین احتمال شرطی را انتخاب می کنیم. و این چنین تا حد زیادی از مشکلاتی که در روش Greedy وجود داشت مقابله می کنیم. اما مهم ترین نکته در روش beam search انتخاب مقدار k است. در این حالت وقتی که ما مقدار k را بسیار کوچک انتخاب می کنیم روش beam search بسیار شبیه به Greedy می شود بنابراین احتمال اینکه به جواب بهینه برسیم بسیار پایین است. از طرفی این مسئله سبب مشکلاتی مثل exposure bias می شود زیرا عملاً مدل یادگرفته شده ما تعمیم پذیری مناسبی ندارد. با این حال اگر مقدار k را بیش از اندازه زیاد کنیم در این صورت مقدار محاسبات و مقداری که باید ذخیره کنیم بسیار زیاد می شود و عملاً از نظر حافظه با محدودیت مواجه می شویم. هم چنین به دلیل اینکه باید زیر درخت های زیادی را مورد بررسی قرار دهیم از نظر سرعت هم دچار مشکل خواهیم شد.

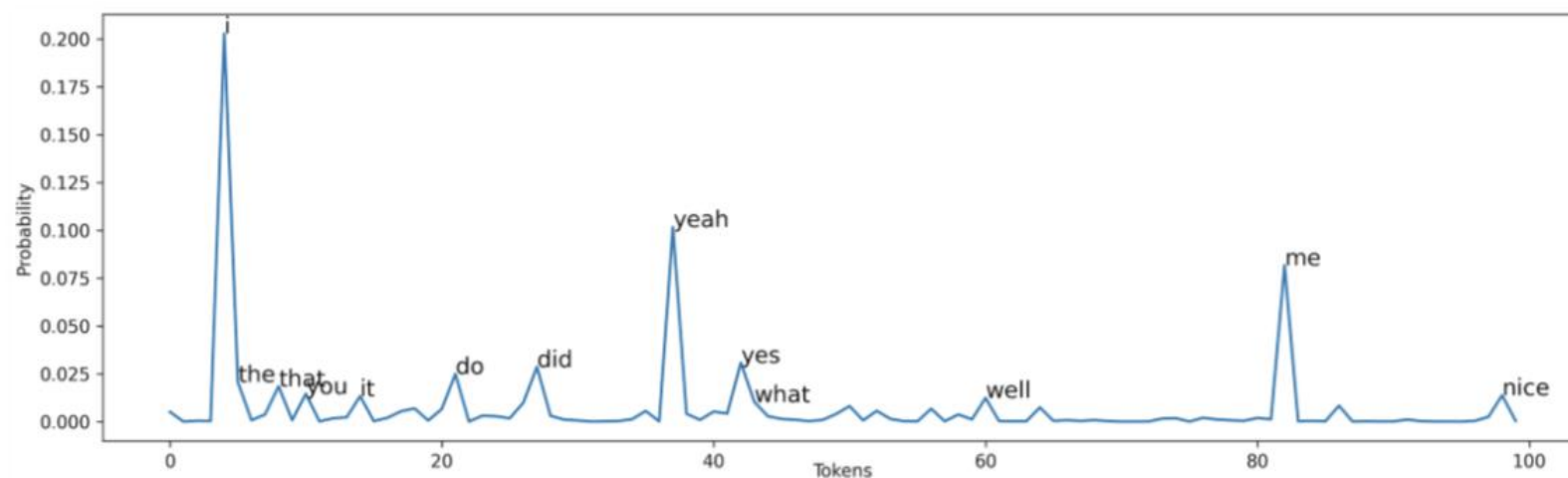


(بخش ۳) حال در بخش سوم مسئله می خواهیم به موضوع دیگری برای تولید دنباله پردازیم. در الگوریتم حریصانه همیشه کلمه با بیشترین احتمال در لایه softmax به عنوان کلمه خروجی انتخاب می شد، اما روش دیگری برای این کار وجود دارد و آن انتخاب تصادفی کلمه خروجی براساس احتمال های لایه softmax است.

(آ) توضیح دهید که مزایای این حالت به حالت انتخاب کلمه با بیشترین احتمال چیست. (۵ نمره)

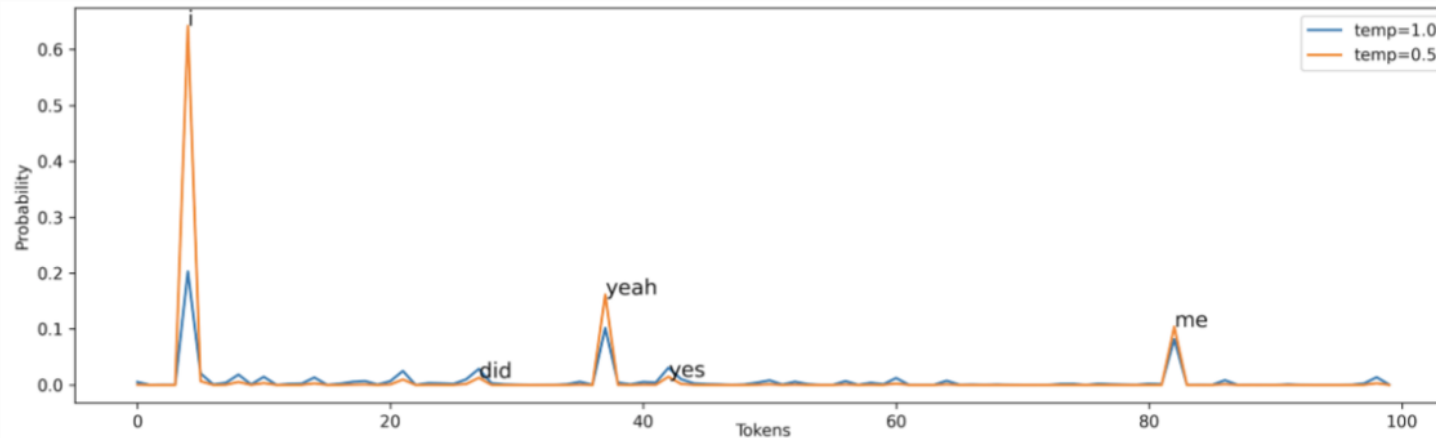


همانطور که گفته شد ایده ی جستجوی حریصانه بر این اساس بود که ما در هر مرحله کلمه ی با بیشترین احتمال را انتخاب می کردیم (که این احتمال از تابع سافت مکس بدست می آمد در واقع ما γ پیش بینی شده را که بازه ای از مثبت بی نهایت تا منفی بی نهایت می توانست داشته باشد به تابع سافت مکس می دادیم و عددی بین صفر و یک تولید می کرد). و سپس بر اساس آن کلمه احتمال های دیگری و کلمات دیگر در عمق های بعدی را تولید می کردیم. اما همانطور که گفته شد این روش مشکلاتی داشت. ایده ی بعدی که مطرح شد نمونه برداری تصادفی بود (random sampling) چون عملاً در روش Greedy بحث شد که انتخاب کلمه با بیشترین احتمال لزوماً نمی تواند ما را به نتیجه ی خوبی برساند و مثلاً ممکن بود جواب بهینه کلمه با سومین اندازه احتمال از نظر بزرگی بهترین باشد از طرفی روش Greedy سبب می شد که ما به صورت پیش فرض همیشه یک جواب تولدی کنیم و این مسئله تولید جملات متنوع را غیر ممکن میکرد. بر اساس نمونه گیری تصادفی ما در هر مرحله از فضای احتمالی که کلمات دارد به صورت تصادفی نمونه برداری می کنیم. به این صورت که فرض می کنیم که مثلاً هر کلمه ی احتمالی دارد و بر حسب تصادف یکی را انتخاب می کنیم. بدیهی است که در این روش نمونه برداری شانس کلماتی که احتمال بیشتری دارند بیشتر است اما کلماتی که شانس کمتری دارند هم از شانس کمی برخوردار نیستند (در واقع با این رویکرد از آن حالت حریصانه تا حدی جلوگیری می کنیم و به کلمات با احتمال های کمتر هم شانس می دهیم و این مسئله باعث میشد که ما جملات متنوعی تولید کنیم). و ممکن است انتخاب شوند. اما مسئله ای که وجود دارد در Greedy ما فقط کلمه با بیشترین احتمال را انتخاب می کردیم که این مسئله می توانست به خاطر حریصانه عمل کردن ما را به نتیجه مناسب نرساند. مثلاً اگر ما از روش greedy استفاده می کردیم پیوسته مجبور بودیم از کلمه i که بیشترین احتمال را دارد استفاده کنیم اما در نمونه برداری تصادفی امکان انتخاب لغات دیگر هم وجود دارد.

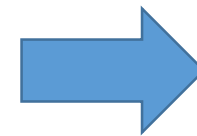




. از طرفی وقتی ما نمونه برداری تصادفی می کنیم در این حالت هم کلمات مثلا با احتمالا 0.00001 بازهم شانس انتخاب دارند اگر چه اندک است. بنابراین انتخاب نمونه از آنهایی که احتمال کمی دارند هم خود یک مشکل است و هیچ تضمینی وجود ندارد که مارا به نتیجه ی مطلوب برساند. برای همین ایده ای مطرح شد تحت عنوان اینکه به جای اینه که γ های پیش بینی شده را به صورت مستقیم به خروجی بدهیم یک مقدار تحت عنوان **temperature** به عنوان ضریب در γ پیش بینی شده ضرب شود و بعد به عنوان ورودی به سافت مکس داده شود. خوبی این روش این بود که به نوعی ما احتمال انتخاب کلمات نامناسب را کمتر و احتمال انتخاب کلمات مناسب را می توانیم بیشتر کنیم و تا حدی با این مشکل انتخاب کلمات نامناسب مقابله کنیم. مثلا وقتی ما ضریب **temperature** را بزرگتر از یک در نظر می گیریم در این حالت γ های بزرگتر بزرگتر و آنهایی که کوچک هستند کوچکتر می شوند. بنابراین پس از اینکه آنها را به عنوان ورودی به تابع سافت مکس می دهیم احتمال کلمات مناسب (با γ پیش بنی شده بزرگتر) بیشتر می شود. و به نوعی با مشکل نمونه برداری تا حدی مقابله کرده ایم و شانس انتخاب کلمات نامناسب را کاهش داده ایم. بدیهی است که زمانی که **temperature** کوچکتر از یک باشید ما به نوعی احتمال های را یکنواخت می کنیم به گونه ای که شانس انتخاب هر کلمه تا حدی یکسان باشد. زمانی که **temperature** خیلی بزرگ تر از یک باشد مثلا بی نهایت γ پیش بنی شده که ای بیشترین مقدار را دارد تقریبا بعد دادن به تابع سافت مکس مقدار تقریبا یک به خود می گیرد و شبیه **Greedy** می شود. همچنین می توان اینگونه برداشت که وقتی در روش **greedy** هر بار یک کلمه با احتمال زیاد انتخاب می شود شبکه به خوبی آموزش نمی بیند و ممکن از روی داده های تست به خوبی نتیجه ندهد نمونه برداری رندم از توزیع احتمال ها می تواند به رسیدن به وزن های مناسب تر کمک کند.



$$\frac{e^{ypred_i}}{\sum e^{ypred_i}}$$



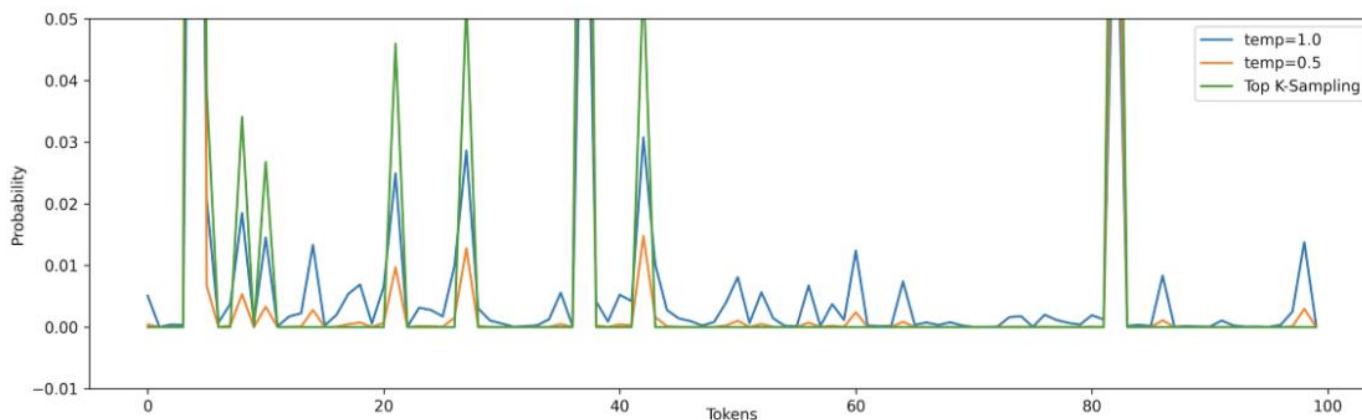
$$\frac{e^{t*ypred_i}}{\sum e^{t*ypred_i}}$$



(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 کم کردن k در top-k sampling را شرح دهید. (۵ نمره امتیازی)



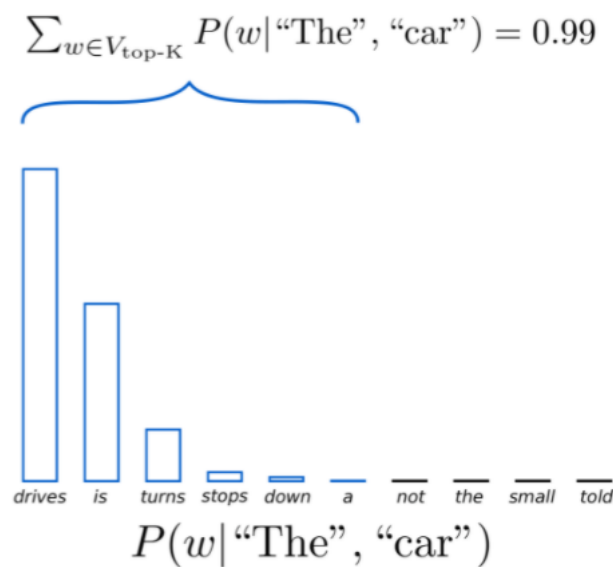
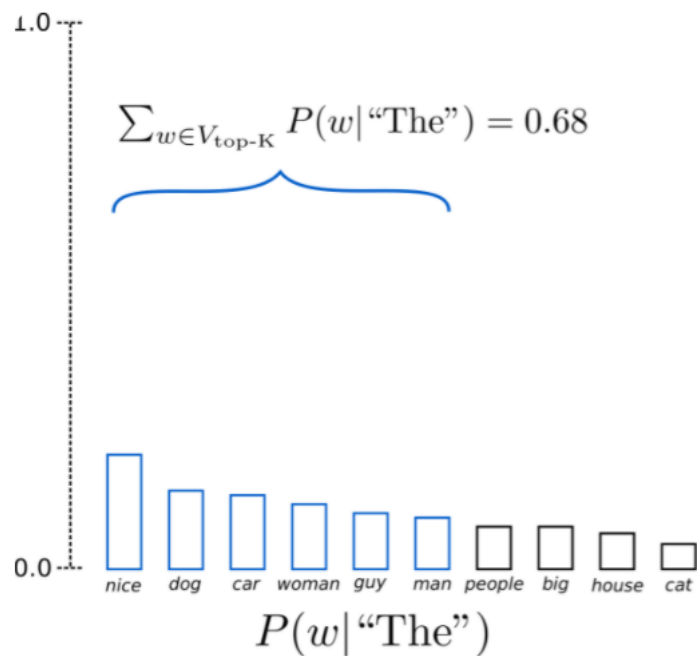
در Pure Sampling ما در هر مرحله مثل t ما از یک توزیع احتمال مثل pt به صورت کاملاً تصادفی نمونه بر میداریم تا کلمه ی بعد را پیش بینی کنیم. این روش شبیه Greedy است اما در pure sampling ما کسبیم را انتخاب نمی کنیم. از این رو این امکان برای هر کلمه وجود دارد که انتخاب شود. اما این مسئله می تواند سبب شود که ما به صورت تصادفی از کلمات نامناسب و نامرتب نمونه برداری کنیم. در روش top k sampling تضمین می کند که کلمات نامرتب و نامناسب انتخاب نمی شوند و در فضای نمونه برداری ما نیستند. در واقع طبق این روش ما k تا از نمونه ها (لغات) که بیشترین احتمال را دارند را به عنوان توزیعی که می خواهیم از آن نمونه برداری کنیم در نظر می گیریم و سپس نمونه های تصادفی را از آن انتخاب می کنیم. در این صورت از انتخاب کلماتی که احتمال کمی دارند و امکان دارد باعث خروجی های نامناسب شود جلوگیری می کنیم. همانطور که در شکل زیر می بینیم:



اما یکی از مشکلات این روش این است که ما باید k را انتخاب کنیم و اگر k خیلی بزرگ باشد مشکلی که پیش می آید این است که مدل در مرحله ی decoding مثلاً از ۱۰ کلمه با بیشترین احتمال می خواهد استفاده کنید و لی ما کلی کلمه ی دیگر با احتمال کمتر را نیز در نظر گرفته ایم. از طرفی وقتی k خیلی کوچک باشد در این حالت ممکن است ما کلماتی که با اینکه احتمال کمی دارند اما می توانند به جواب بهینه ما را برسانند را در نظر نمی گیریم و در این صورت تا حدی شبیه Greedy عمل می کنیم. از طرفی هرچه k را بزرگتر کنیم انگار شبیه pure sampling می شویم و امکان اینکه کلمات با احتمال کم هم نمونه برداری شوند هست (در صورتی که ما می خواستیم از این اتفاق جلوگیری کنیم).



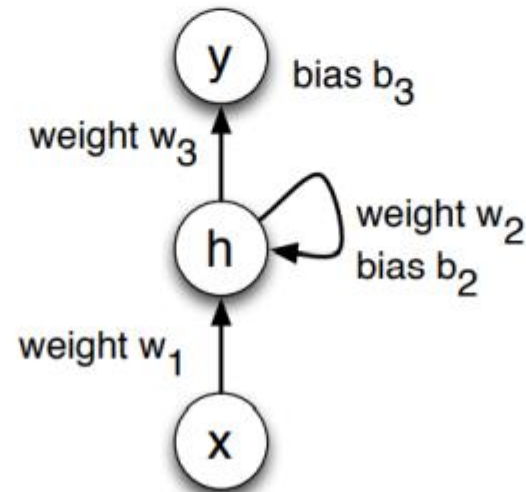
همچنین یکی از مشکلاتی که این روش دارد این است که وقتی k یک مقدار ثابت است مسئله ای که به وجود می آید این است که وقتی یک توزیع خیلی sharp است مانند توزیع سمت راست در شکل زیر (یه کلمه در آن احتمال بالایی دارد) ما از آن کلمه به همراه کلیدی کلمه که احتمال خیلی پایینی دارند نمونه برداری می کنیم. این مسئله باعث می شود که در ما از توزیع هایی که احتمال های یکنواختی (شکل چپ در شکل زیر) هم دارند k تا را انتخاب کنیم که این مسئله سبب می شود که کلیدی که کلماتی که احتمال های مناسب و قابل قبولی داشتند در نظر گرفته نشود و این مسئله می تواند مسئله ساز باشد. زیرا با از یک توزیع احتمال های نامناسبی که احتمال پایینی داشتند را انتخاب کرده ایم و این در حالی است که از یک توزیع که احتمال های مناسبی داشتند را در نظر نگرفته ایم و این مسئله سبب شد که روش $\text{top } p$ معرفی شود.

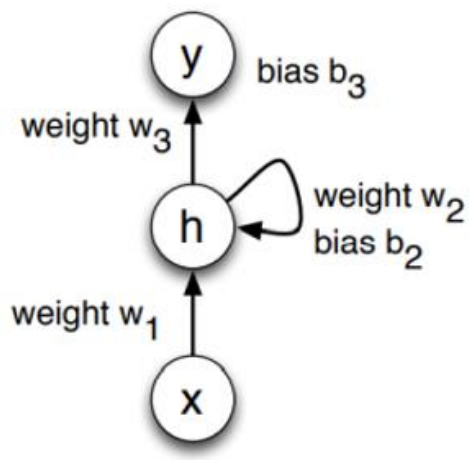




مسئله ۳. (۱۰ نمره)

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.





مقادیر پیشنهادی برای وزن‌ها	
W1	-2
W2	4
b2	0.5
W3	-1
b3	1

بر این اساس این ایده وزن‌ها را مقدار دهی می کنیم که h_t در هر مرحله (تا زمانی که صفر نیامده است) مقدارش صفر باشد و پس از اینکه اولین صفر آمد مقدارش یک و تا آخر یک بماند. از این رو مقدار w_1 یک عدد منفی انتخاب شد تا در زمانی هایی که هنوز صفر نیامده و h_t صفر است بر اساس $threshold$ مد نظر کمتر از صفر باشد و صفر را به عنوان خروجی برگرداند و سپس صفر در w_3 که مقدار منفی یک دارد ضرب می شود و یا یک جمع می شود و چون بزرگتر از صفر است مقدار یک را به عنوان خروجی برگردانیم. مقدار w_2 یک عدد مثبت و بزرگتر در نظر گرفته شده تا زمانی که اولین صفر آمد مقدار h_t را به یک تغییر دهد و پس از آن هر بار چه صفر بیاید چه یک خروجی گره h صفر یک خواهد بود و ضرب یک در w_3 که مقدار منفی یک دارد به اضافه یک می شود صفر و خروجی $threshold$ همواره صفر می شود.

دنباله ی ۱۰۱۰۱۱۰ را در نظر می گیریم

1

$$z = x * w_1 + h_{t-1} * W_2 + b_2 = 1 * -2 + 0 * 4 + 0.5 = -1.5 \quad \Rightarrow \quad h_t = -1.5 \leq 0 = 0$$

$$y = h_t * w_3 + b_3 = 0 * -1 + 1 = 1 \Rightarrow 1 > 0 = 1$$

0

$$z = x * w_1 + h_{t-1} * W_2 + b_2 = 0 * -2 + 0 * 4 + 0.5 = 0.5 \quad \Rightarrow \quad h_t = 0.5 > 0 = 1$$

$$y = h_t * w_3 + b_3 = 1 * -1 + 1 = 0 \Rightarrow 0 \leq 0 = 0$$



1

$z = x * w_1 + h_{t-1} * W_2 + b_2 = 1 * -2 + 1 * 4 + 0.5 = 2.5 \rightarrow h_t = 2.5 > 0 = 1$

$y = h_t * w_3 + b_3 = 1 * -1 + 1 = 0 \Rightarrow 0 \leq 0 = 0$

0

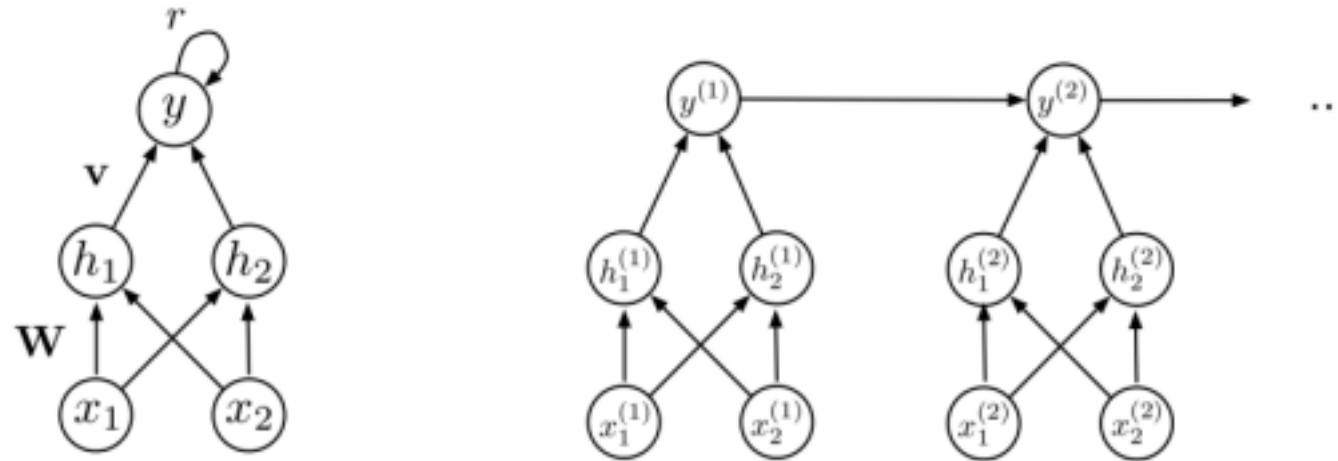
$z = x * w_1 + h_{t-1} * W_2 + b_2 = 0 * -2 + 1 * 4 + 0.5 = 4.5 \rightarrow h_t = 4.5 > 0 = 1$

$y = h_t * w_3 + b_3 = 1 * -1 + 1 = 0 \Rightarrow 0 \leq 0 = 0$



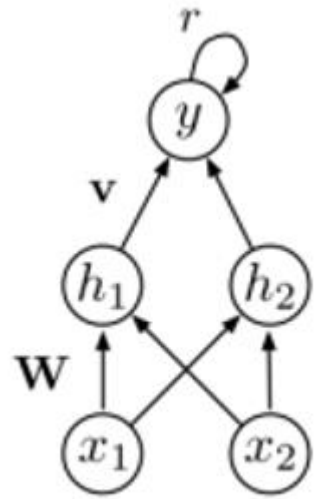
مسئله ۴. (۵ نمره)

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 گردانند.



$$\mathbf{h}^{(t)} = \phi(\mathbf{W}\mathbf{x}^{(t)} + \mathbf{b})$$
$$y^{(t)} = \begin{cases} \phi(\mathbf{v}^T \mathbf{h}^{(t)} + r y^{(t-1)} + c) & \text{for } t > 1 \\ \phi(\mathbf{v}^T \mathbf{h}^{(t)} + c_0) & \text{for } t = 1, \end{cases} \quad \phi(z) = \begin{cases} 1 & \text{if } z > 0 \\ 0 & \text{if } z \leq 0 \end{cases}$$

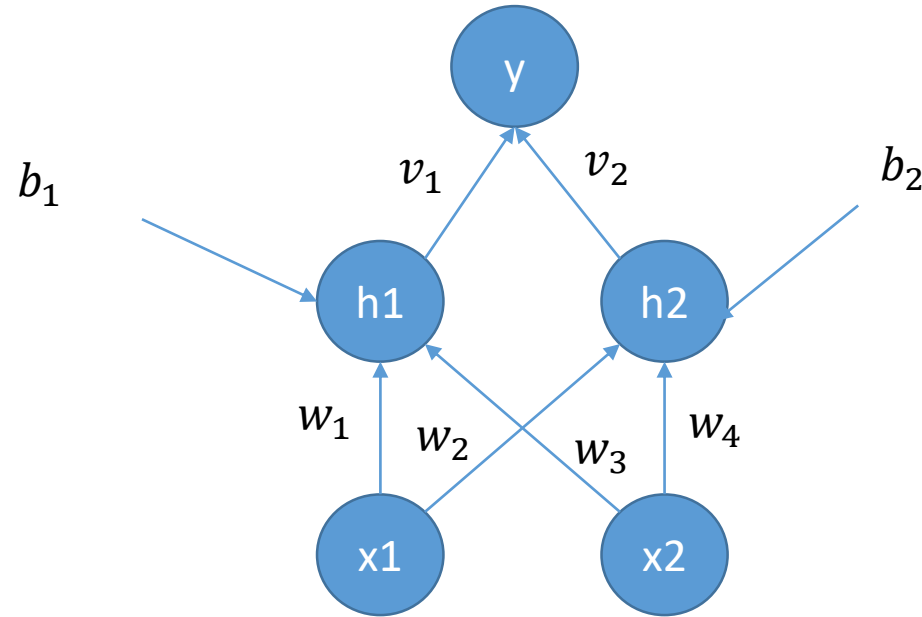
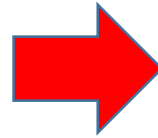
ماتریس \mathbf{W} یک ماتریس 2×2 و b و v بردارهای دو بعدی و c و r و c_0 مقادیر اسکالر می باشد. آن ها را به گونه ای تعیین کنید که شبکه کارکرد تعریف شده را داشته باشد. (راهنمایی: خروجی $y^{(t)}$ در هر لحظه نشان می دهد آیا دو دنباله تا آن لحظه برابر بوده اند یا خیر. لایه مخفی اول نشان میدهد آیا دو ورودی در لحظه t صفر بوده اند یا خیر و لایه مخفی دوم نشان می دهد آیا دو ورودی در لحظه t ، ۱ بوده اند یا خیر.)



همانطور که در صورت مسئله گفته شده ما به دنبال این هستیم که دنباله های باینری یکسان را شناسایی کنیم و در خروجی در ازای یکسان بودن یک را نمایش دهیم و در صورتی که یکسان نباشند یک صفر را به عنوان خروجی نمایش دهیم. همانطور که صورت مسئله گفته شده ما به ازای هر بیت یک بلوک داریم که خروجی این بلوک به عنوان ورودی در مرحله ی بعد تاثیر گذار است. ما به دنبال این هستیم که در دو دنباله نظیر به نظیر بیت ها را بررسی کنیم که آیا یکسان هستند یا خیر. برای این کار می توانیم یک معماری پیاده سازی کنیم که این کار انجام دهد.

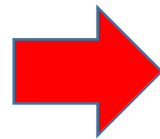
مقداردهی وزنها:

$$w = \begin{bmatrix} w_1 = 1 & w_2 = -1 \\ w_3 = 1 & w_4 = -1 \end{bmatrix}$$
$$v = [v_1 = 1, v_2 = 1]$$
$$b = [b_1 = -1.5, b_2 = 0.5]$$



$$c = -1$$
$$c_0 = -0.5$$
$$r = 1$$

همچنین طبق
فرضیات مسئله
می دانیم:



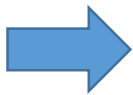
$$\mathbf{h}^{(t)} = \phi(\mathbf{W}\mathbf{x}^{(t)} + \mathbf{b})$$

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

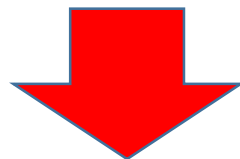
$$\phi(z) = \begin{cases} 1 & \text{if } z > 0 \\ 0 & \text{if } z \leq 0 \end{cases}$$



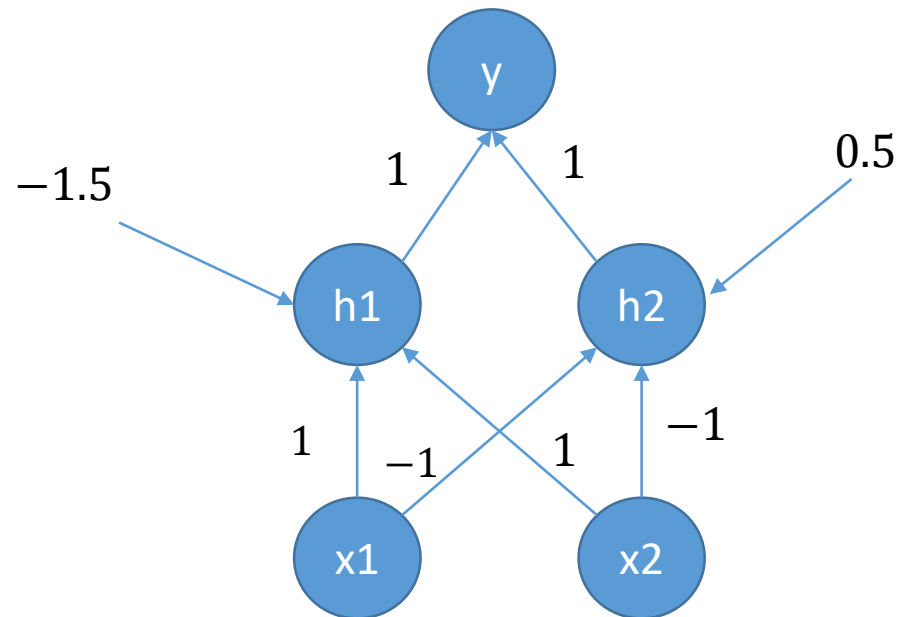
$$w = \begin{bmatrix} w_1 = 1 & w_2 = -1 \\ w_3 = 1 & w_4 = -1 \end{bmatrix}$$
$$v = [v_1 = 1, v_2 = 1]$$
$$b = [b_1 = -1.5, b_2 = 0.5]$$



$$c = -1$$
$$c_0 = -0.5$$
$$r = 1$$



تست



bse_1	1	0	0	1	1
bse_2	1	0	0	0	1
y_{t-1}	---	1	1	1	0
y_t	1	1	1	0	0

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



$$h1 = (1 * 1 + 1 * 1) - 1.5 = 0.5$$
$$\Rightarrow 0.5 > 0 \Rightarrow 1$$

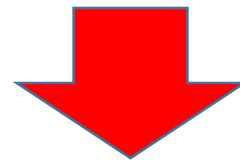


$$h2 = (1 * -1 + 1 * -1) + 0.5$$
$$= -1.5 \Rightarrow -1.5 \leq 0 \Rightarrow 0$$



برای زمان $t=1$
طبق فرضیات
مسئله داریم

$$y_t = (1 * 1 + 0 * 1) - 0.5 = 0.5$$
$$\Rightarrow 0.5 > 0 \Rightarrow 1$$



تست

bse_1	1	0	0	1	1
bse_2	1	0	0	0	1
y_{t-1}	---	1	1	1	0
y_t	1	1	1	0	0

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



$$h1 = (0 * 1 + 0 * 1) - 1.5 \\ = -1.5 \Rightarrow -1.5 \leq 0 \Rightarrow 0$$



$$h2 = (0 * -1 + 0 * -1) + 0.5 \\ = 0.5 \Rightarrow 0.5 > 0 \Rightarrow 1$$



برای زمان $t > 1$
طبق فرضیات
مسئله داریم

$$y_t = (1 * 1 + 0 * 1) - 1 + 1 * 1 = 1 \\ \Rightarrow 1 > 0 \Rightarrow 1$$