یادگیری ژرف

تمرین سوم، بخش تئوری

جواد راضی (۴۰۱۲۰۴۳۵۴)

سوال اول: LSTM

آ)

تصویر ارائه‌شده نمایی از سلول LSTM است که با اجزای بهینه‌ساز Adam تلفیق شده است. معادلات نمایش داده‌شده، خروجی‌های LSTM، بردار اولین لحظه (میانگین) و بردار دومین لحظه (واریانس بدون مرکز) را در قوانین به‌روزرسانی Adam نشان می‌دهند.

- خروجی V_t:این بردار دومین مومنت (لحظه) (واریانس بدون مرکز) در بهینه‌ساز Adam است که با ورودی وزن‌دار و مقیاس‌بندی شده توسط S به‌روزرسانی می‌شود.

$$v_t = \mu \cdot v_{t-1} + S \cdot W \cdot x_t$$

- خروجی m_t : این بردار اولین لحظه (میانگین) در بهینه‌ساز Adam است که با ورودی وزن‌دار مربع‌شده به‌روزرسانی می‌شود.

$$m_t = \beta \cdot m_{t-1} + (1 - \beta) \cdot (W \cdot x_t)^2$$

- خروجی h_t : این خروجی سلول LSTM است که با استفاده از تابع فعال‌سازی سیگموئید به مجموع وزن‌دار خروجی قبلی و نسبت بردار دومین لحظه به ریشه دوم بردار اولین لحظه اعمال می‌شود.

$$h_t = \sigma\left(U \cdot h_{t-1} + \frac{v_t}{\sqrt{m_t} + \epsilon}\right)$$

ب)

برای محاسبه گرادیان $\frac{\partial h_T}{\partial x_i}$ در یک شبکه LSTM که با بهینه‌ساز Adam ادغام شده است، با تکنیک Backpropagation Through Time تأثیر هر ورودی x_i بر خروجی نهایی h_T را محاسبه کنیم. این شامل محاسبه مشتقات جزئی در هر گام زمانی است.

با این مفروضات داریم:

$$\frac{\partial h_T}{\partial x_i} = \sum_{t=i}^{T} \frac{\partial h_T}{\partial h_t} \frac{\partial h_t}{\partial x_i}$$

مقادیر $\frac{\partial h_t}{\partial x_i}$ در هر گام زمانی t بر اساس مشتق‌گیری از معادلات فعال‌سازی و به‌روزرسانی LSTM حاصل می‌شود. این کار شامل مشتق‌گیری از توابع فعال‌سازی سیگموئید و تانژانت هایپربولیک، و همچنین وزن‌های به‌کاررفته در سلول LSTM است.

با توجه به پیچیدگی‌های موجود در معادلات LSTM و تابع هزینه، محاسبه دستی این گرادیان دشوار است و راهی به ذهنم نمی‌رسد که بدون محاسبات عددی، آن را حساب کرد.

(ج)

برای محاسبه $\frac{\partial h_T}{\partial h_i}$، ابتدا باید مشتق هر گام h_t نسبت به گام قبلی h_{t-1} را در نظر بگیریم. از آنجایی که هر h_t تابعی از h_{t-1} و سایر متغیرهای ورودی در زمان t است، می‌توانیم از قاعده زنجیره‌ای برای مشتق‌گیری استفاده کنیم.

با توجه به معادله داده شده برای h_t در شبکه LSTM با بهینه‌ساز Adam

$$h_t = \sigma\left(U \cdot h_{t-1} + \frac{v_t}{\sqrt{m_t} + \epsilon}\right)$$

مشتق σ را می‌توان با استفاده از قاعده مشتق تابع ترکیبی به دست آورد که برای تابع سیگموئید $\sigma(x) = \frac{1}{1+e^{-x}}$ به صورت زیر است:

$$\sigma'(x) = \sigma(x)(1 - \sigma(x))$$

حالا، با در نظر گرفتن اینکه مشتق h_T نسبت به h_i شامل حاصلضرب مشتقات در هر گام از i تا T است، می‌توانیم بنویسیم:

$$\frac{\partial h_T}{\partial h_i} = \prod_{j=i+1}^{T} \frac{\partial h_j}{\partial h_{j-1}}$$

برای هر گام j، مشتق h_j نسبت به h_{j-1} می‌تواند به صورت زیر نوشته شود:

$$\frac{\partial h_j}{\partial h_{j-1}} = \frac{\partial \sigma}{\partial h_{j-1}} = \sigma'(U \cdot h_{j-1} + \frac{v_j}{\sqrt{m_j} + \epsilon}) \cdot U$$

$$= \sigma(U \cdot h_{j-1} + \frac{v_j}{\sqrt{m_j} + \epsilon})(1 - \sigma(U \cdot h_{j-1} + \frac{v_j}{\sqrt{m_j} + \epsilon})) \cdot U$$

با جایگذاری این مشتق در معادله قبلی و حذف σ و $(1-\sigma)$ که از قاعده مشتق تابع سیگموید آمده‌اند، داریم:

$$\frac{\partial h_T}{\partial h_i} = \prod_{j=i+1}^{T} \left(\sigma(U \cdot h_{j-1} + \frac{v_j}{\sqrt{m_j} + \epsilon})(1 - \sigma(U \cdot h_{j-1} + \frac{v_j}{\sqrt{m_j} + \epsilon})) \right) \cdot U^{T-i}$$

(د)

معماری LSTM که با بهینه‌ساز Adam ترکیب شده است، با وجود طراحی پیشرفته برای مقابله با مشکلات ناپدید شدن و انفجار گرادیان‌ها، همچنان ممکن است با این مشکلات روبرو شود. استفاده از تابع فعال‌سازی سیگموئید می‌تواند در گام‌های زمانی طولانی موجب ناپدید شدن گرادیان شود، زیرا مشتقات این تابع مقادیر کوچکی دارند که می‌توانند در طول چندین گام زمانی به سرعت کاهش یابند.

برای مقابله با ناپدید شدن گرادیان، یکی از راه‌کارهای ساده می‌تواند این باشد که تابع سیگموئید را در یک ثابت بزرگتر از یک ضرب کنیم تا مقادیر مشتقات آن افزایش یابند. همچنین، استفاده از روش‌هایی نظیر Gradient Clipping برای جلوگیری از انفجار گرادیان پیشنهاد می‌شود که در آن مقادیر بزرگ گرادیان‌ها را محدود می‌کنیم تا از یک آستانه معین فراتر نروند. این تکنیک‌ها به حفظ پایداری در فرآیند یادگیری کمک می‌کنند و امکان رسیدن به نتایج بهینه‌تر در طول آموزش را فراهم می‌آورند.

(ه)

معماری بحث‌شده در این سوال، عملکردی شبیه به الگوریتم بهینه‌سازی Adam دارد. الگوریتم Adam، که مخفف Adaptive Moment Estimation است، برای تنظیم نرخ یادگیری هر پارامتر به صورت

انطباقی و بر اساس تخمین‌های اولین و دومین لحظه‌ی گرادیان‌ها کار می‌کند. این کار به الگوریتم اجازه می‌دهد که نرخ یادگیری را برای هر پارامتر به صورت مجزا تنظیم کند، که می‌تواند به کاهش مشکلات مربوط به ناپدید شدن گرادیان‌ها و انفجار گرادیان‌ها کمک کند و همگرایی سریع‌تر را فراهم آورد.

در معماری نشـــان داده شـــده، عملیات‌های به‌روزرســـانی وزن‌ها و حالت‌ها از طریق جمع‌آوری میانگین متحرک وزن‌دار گرادیان‌ها (معرفی شده به عنوان $\mathrm{m_t}$ و مربعات آنها (نشان داده شده به عنوان v_t انجام می‌شـود، که این دقیقاً همان رویکردی اسـت که توسـط Adam به کار گرفته می‌شـود. از آنجا که Adam همچنین از نرخ یادگیری انطباقی اســتفاده می‌کند که بر اســاس مقادیر $\mathrm{m_t}$ و v_t تنظیم می‌شـــود، این سـلول LSTM ترکیبی از ویژگی‌های LSTM و مکانیزم به‌روزرسـانی Adam را برای بهبود فرآیند یادگیری در بر دارد.

سوال دوم: RNN

بررسی اینکه یک دنباله باینری یکسان هستند، معادل عملکرد XNOR است. (که اگر بیت‌ها برابر باشند، مقدار خروجی یک است). می‌دانیم XNOR دو رشته دودویی، معادل XNOR گرفتن از هر جفت بیت متناظر، و AND کردن این بیت‌ها می‌باشد.

اگر بخواهیم این کار را با یک شبکه RNN مطابق شکل داده‌شده مدل کنیم، که در هر گام زمانی، دو بیت، یکی از هر ورودی گرفته می‌شود، قاعدتا باید در یک سلول این بیت‌ها XNOR شده، و با خروجی گام بعدی AND شوند.

اکنون سعی می‌کنیم پارامترهای هر سلول، و پارامترهای مربوط به کلیت معماری شبکه را در این راستا تعیین کنیم:

پارامترهای سلول:

مقدار وزن‌ها (W): می‌دانیم که

$$y^t = AND\left(y^{t-1}, XNOR(x_1^t, x_2^t)\right)$$

با توجه به اینکه XNOR را می‌توان با گیت‌های AND و NOR ساخت، یکی از توابع فعال‌ساز پله (h)، وزن‌ها باید بگونه‌ای باشد که یکی AND رو مدل کرده و یکی NOR را. با فرض اینکه $\mathrm{h_1}$ کار اولی را انجام داده و h_2 کار دومی را، می‌توان برای هر جفت ورودی (بیت)، و خروجی سلول قبل در زمان (y^{t-1}) یک Truth Table نمایش داد:

x_1	x_2	y^{t-1}	(y^t)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

طبق ساختار شبکه عصبی:

$$\begin{bmatrix} h_1 \\ h_2 \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

ابتدا سعی می‌کنیم صرفا به خود سلول فکر کنیم. همان‌طور که گفته شد یک سلول، یک XNOR را مدل می‌کند (با و خروجی زمان قبل AND می‌نماید.) برای مدل‌سازی XNOR، می‌تواند h1 را به صورت AND دو ورودی، و h2 را به صورت NOR آن دو در آورد. اگر یکی از این حالات برقرار باشد، آن دو ورودی برابرند (جمع h1 و h2).

برای AND، می‌تواند وزن‌ها را ۱+ گذاشته، و بایاس را مقداری نزدیک ۲-، مثلا ۱.۵- گذاشت. با این کار، تنها زمانی که هر دو ورودی ۱ باشند، تابع پله فعال خواهد شد.

برای NOR، وزن‌ها را ۱- می‌گذاریم. بایاس را نیز اندکی بزرگتر از صفر، مثلا ۰.۵ می‌گذاریم. با این کار،فقط زمانی که هر دو ورودی صفر باشند، به علت بایاس بزرگتر از صفر، تابع پله فعال خواهد شد. لازم به ذکر است که این حالت به حالت قبلی Mutually Exclusive است.

بنابراین داریم:

$$W = \begin{bmatrix} +1 & +1 \\ -1 & -1 \end{bmatrix}, B = \begin{bmatrix} -1.5 \\ +0.5 \end{bmatrix}$$

اکنون هر سلول شبکه، XNOR را مدل می‌کند. با تنظیم پارامترهای c0, c, r، می‌توانیم بررسی یکسان بودن کل ورودی را طبق رابطه داده‌شده در ابتدای سوال انجام دهیم:

مقدار v^t را یک بردارد دو بعدی با دو مقدار ۱ و ۱ در نظر می‌گیریم. (با همین بردار ساده می‌توان مسئله را مدل کرد.) برای زمان صفر، داریم:

$$y^0 = \phi(h_1 + h_2 + c_0) = \phi([h_1 OR\ h_2] + c_0)$$

در واقع حاصل ضرب *v* در *h*، یک اسکالر صفر یا یک است. برای زمان‌هایی که دو ورودی برابرند، این مقدار یک می‌باشد. اما برای زمان‌هایی که برابر نیستند و این مقدار صفر است، با تنظیم *c0* با مقداری کمتر از صفر، می‌توان یکسان‌بودن دو عدد را در زمان صفر بررسی نمود. پس داریم $c_0 = 0$

برای زمان *t* داریم:

$$y^t = \phi([h_1 OR\ h_2] + ry^{t-1} + c)$$

r را ۱ در نظر می‌گیریم که حاصل بررسی بیت‌های قبلی را کاملا داشته باشیم. با این اوصاف، اگر بیت‌ها بازم هم برابر باشند، مقدار جمع اولیه ۲+ می‌شود. اگر بیت‌ها این بار برابر نباشند، مقدار جمع اولیه ۱+ می‌شود. اگر بیت‌ها قبلا برابر نبودند نیز میزان جمع اولیه ۰ یا ۱+ می‌شود. در هر حالت، برای ما مهم است که تنها حالت اول تابع پله را فعال کند. برای همین، جایگذاری مقدار $c = -1.5$، این شرط را ارضاء می‌کند.

خلاصه:

به طور خلاصه طبق توضیحات، پارامترها اینگونه تعریف شدند:

$$W = \begin{bmatrix} +1 & +1 \\ -1 & -1 \end{bmatrix}, B = \begin{bmatrix} -1.5 \\ +0.5 \end{bmatrix}$$
$$v = \begin{bmatrix} +1 \\ +1 \end{bmatrix}, r = 1.0$$
$$c_0 = -0.5, c^t = -1.5$$