چکیده

مدل های زبانی آماری بخشی حیاتی بسیاری از برنامه های موفقی چون بازشناسی خودکار صدا و ترجمه ماشین آماری (برای نمونه ترجمه گوگل) می باشد. روش های سنتی برای برآورد کردن این مدل ها برپایه شمار N–گرام ها می باشد. با وجود ضعف های شناخته شده N–گرام ها و تلاش های فراوان جامعه پژوهشی در بسیاری از شاخه ها (بازشناسی صدا، ترجمه ماشینی، هوش مصنوعی، پردازش زبان طبیعی، فشره سازی داده ها و ...)، N–گرام ها به عنوان فناوری روز بجا ماندند. انگیزه این پایان نامه این است که از شبکه های عصبی بازگردی بهبود یافته برای تولید هوشمند متن بکار برده شود.

کلیدواژه ها: شبکه های عصبی بازگردی، تولید متن، هوش مصنوعی

فهرست مطالب

فهرست شکل ها

فهرست جدول ها

فصل یک

مقدمه

۱ مقدمه

شبکه های عصبی بازگردی[۱] یک کلاس گرانبها از مدل های پویایی هستند که برای تولید دنباله ها در دامنه های گسترده ای چون موسیقی [٤]، [٦]، متن [۳۰] و داده های دریافت حرکت[۲] [۲۹] به کار برده می شوند. شبکه های عصبی بازگردی را می توان برای ساخت دنباله ها از راه پردازش گام به گام دنباله های با داده های راستین و پیش بینی خروجی آینده آن، آموزش داد. با پنداشتن این که پیش بینی ها احتمالاتی هستند، دنباله های یک رمان را می توان از یک شبکه آموزش دیده با نمونه گیری پی در پی از توزیع خروجی شبکه و سپس خوراندن نمونه به ورودی گام آینده، ساخت. به گفته ای دیگر، شبکه را وادار کنیم که با ساخته های خود همچون داده های راستین رفتار کند، همانگونه که یک آدم رویا می بینید. اگرچه شبکه خودش قطعی هست، تصادفی بودنی که با برداشتن نمونه ها افشانده (تزریق) می شود یک توزیع روی دنباله ها القا می کند. این توزیع شرطی است، چون حالت درونی شبکه و بنابراین توزیع پیشگویانه آن به ورودی های پیشین آن وابسته است.

شبکه های عصبی بازگردی 'فازی' هستند چون که آنها از قالب های دقیق داده های آموزشی برای پیش بینی به کار نمی برند بلکه تا اندازه ای – همچون دیگر شبکه های عصبی- از نمایش درونی شان برای انجام درون یابی با ابعاد بالا میان نمونه های آموزشی به کار می گیرند. این گفته آنها را از مدل های N-گرام و الگوریتم های فشرده سازی[۳] همچون پیش بینی با جوریابی جزئی[٤] [٥]، که توزیع های پیشگویانه شان با شمارش جورهای دقیق میان تاریخچه تازه و مجموعه آموزشی انجام می شود. دستاورد آن – که بی درنگ که از نمونه ها آشکار می باشد، این است که شبکه های عصبی بازگردی (وارون الگوریتم های برپایه قالب) داده های آموزشی را به گونه ای پیچیده

[۱] Recurrent neural networks
[۲] Motion capture data
[۳] Compression algorithms
[٤] Prediction by Partial Matching

ترکیب و بازسازی می کنند و خیلی کم یک چیز را دوبار می سازند. افزون بر این، پیش بینی های فازی آن دچار پیچیدگی ابعاد نمی شوند و بنابراین در مدل سازی داده های راستین و چندمتغیره از جوریابی های دقیق بهتر هستند. در اصل یک شبکه عصبی بازگردی به اندازه بسنده بزرگ می تواند برای ساخت دنباله های با پیچیدگی دلخواه بسنده باشد. ولی هنگام به کار بستن آنها، شبکه های عصبی بازگردی استاندارد توانایی نگهداری اطلاعات درباره ورودی های گذشته خیلی دور را ندارند [۱۵]. همچنین کم شدن توانایی آنها برای مدل سازی ساختار های با بازه های بزرگ، این 'فراموشی' آنها را مستعد ناپایداری هنگام ساخت دنبال ها می کند. مشکل (رایج به همه مدل های سازنده شرطی) این است که اگر پیش بینی های شبکه تنها بسته به چند ورودی تازه گذشته باشد و این ورودی ها خود به دست شبکه پیش بینی شده اند، و بنابراین فرصت اندکی برای بازیابی از اشتباهات گذشته دارد. داشتن یک حافظه بزرگتر یک اثر پایدار کننده دارد چون هرچند اگر شبکه شبکه نتواند از تاریخچه تازه خود چیزی بفهمد، می تواند به گذشته دورتر خود برگردد تا پیش بینی هایش را فرموله کند. مشکل ناپایداری به ویژه برای داده های راستین درست است، یعنی جایی که دروغی یا نادرستی در داده های آموزشی هستند و پیش بینی های آن بسیار گمراه کننده و نادرست می شود. چاره ای که برای مدل های شرطی پیشنهاد شده است این است که به پیش بینی ها پیش از خوراندن آنها به مدل نویز وارد کنیم [۳۱]، که از این راه توانمندی مدل به ورودی های ناخواسته بدست می آید. به هر روی ما باور داریم که یک حافظه بهتر چاره ای بهتر و سودمند تر است.

حافظه کوتاه–مدت دراز[۵] (LSTM) یک ساختار شبکه عصبی بازگردی هست و طراحی شده است تا در نگهداری و دستیابی اطلاعات از شبکه های عصبی بازگردی استاندارد بهتر باشد. به تازگی LSTM دستاوردهای بروز و درخوری در چندین کار پردازش دنباله داده است، همچون بازشناسی

[۵] Long short-term memory (LSTM)

دست نوشته و صدا [۱۰] و [۱۲] انگیزه اصلی ما این است که نشان دهیم که LSTM می تواند از حافظه خود برای ساخت دنباله های راستین و پیچیده ی دربردارنده ساختار دوربرد بهره ببرد.

فصل دوم
مروری بر روش‌های تولید و پیش بینی متن

۲ مروری بر پیشینه پژوهش

دنباله بردار ورودی $x = (x_1, ..., x_T)$ از پیوندهای وزندار به پشته ای از N لایه پنهان پیوند خورده بازگردی گذرانده می شود تا نخست دنباله های بردار پنهان $h_n = (h_1^n, ..., h_T^n)$ و سپس دنباله بردار برونداد $y = (y_1, ..., y_T)$ را محاسبه کند. هر بردار برونداد y_t برای پارامتری کردن یک توزیع پیشگویانه $\Pr(x_{t+1} | (y_t))$، روی ورودی های آینده ممکن x_{t+1} بکار برده می شود. عنصر نخست x_1 از هر دنباله ورودی همیشه یک بردار پوچ است که همه درایه های آن صفر هستند؛ بنابراین شبکه یک پیش بینی برای x_2، که اولین ورودی راستین است، با هیچ دانسته پیشینی بیرون می دهد. شبکه در هم فضا و هم زمان ژرف است، به این گونه که هر تکه از اطلاعاتی که یا عمودی و یا افقی از گراف محاسبه گذر می کند بدست چندین ماتریس وزن پی در پی و ناخطی بودن کنش خواهد داشت.

به 'پیوندهای پرشی[۶]' از ورودی ها به همه لایه های پنهان و از همه لایه های پنهان به ورودی ها بنگرید. اینها آموزش شبکه های ژرف را با کاهش شمار گام های پردازش میان پایین و بالای شبکه و بدنبال آن سبک کردن مشکل 'گرادیان سست شونده[۷]'، آسانتر می کند [۱]. در حالت ویژه ای که $N = 1$ ساختار به یک شبکه عصبی بازگردی تک لایه پیش بینی گام آینده کاهش می یابد.

برانگیختگی لایه پنهان با انجام پی در پی برابری های زیر از $t = 1$ تا T و از $n = 2$ تا N بدست می آید.

$$h_t^1 = H\left(W_{ih^1} x_t + W_{h^1 h^1} h_{t-1}^1 + b_h^1\right) \quad (1)$$

$$h_t^n = H\left(W_{ih^n} x_t + W_{h^{n-1} h^n} h_t^{n-1} + W_{h^n h^n} h_{t-1}^n + b_h^n\right) \quad (2)$$

[۶] Skip connections
[۷] vanishing gradient

که Wها نشان دهنده ماتریس های وزن می باشند (برای نمونه W_{ih^n} ماتریس وزن پیوند دهنده ورودی ها به nامین لایه پنهان، $W_{h^1h^1}$ پیوند بازگردی در لایه پنهان نخست و ... هستند) و b نشان دهنده بردارهای بایاس هستند و H تابع لایه پنهان است.

دنباله برونداد برپایه زیر بدست می آید:

$$\hat{y}_t = b_y + \sum_{n=1}^{N} W_{h^n y} h_t^n \quad (۳)$$

$$y_t = Y(\hat{y}_t) \quad (٤)$$

که Y تابع لایه خروجی است.

بردارهای برونداد y_t برای پارامتری کردن توزیع پیشگویانه $\Pr(x_{t+1}|y_t)$ برای ورودی آینده بکار برده می شوند. فرم $\Pr(x_{t+1}|y_t)$ باید با هشیاری برگزیده شود تا با داده ورودی جور شود. پیدا کردن یک توزیع پیشگویانه برای داده های راستین و با بعد های بالا می تواند بسیار چالش برانگیز باشد.

احتمالی که بدست شبکه به دنباله ورودی x داده می شود:

$$\Pr(x) = \prod_{t=1}^{T} \Pr(x_{t+1}|y_t) \quad (۵)$$

و دنباله زیان L(x) که برای آموزش شبکه بکار برده می شود لگاریتم منفی Pr(x) می باشد:

$$L(x) = -\sum_{t=1}^{T} \log \Pr(x_{t+1}|y_t) \quad (٦)$$

مشتق های جزئی آن با توجه به وزن های شبکه را می توان به گونه ای کارامد از راه انتشار رو بـه پشت از زمان محاسبه کرد [۳۳] و شبکه را می توان با گرادیان کاهشی آموزش داد.

۲-۱ حافظه کوتاه-مدت دراز

در بیشتر شبکه های عصبی بازگردی تابع لایه پنهان H یک کاربرد عنصر به عنصر تابع سیگموید[۸] است. به هر روی ما دریافته ایم که ساختار حافظه کوتاه-مدت دراز راکه از سلول های حافظه ساخته شده با انگیزه برای نگهداری اطلاعات بهره می برد، در پیدا کردن و بهره برداری از وابستگی های دوربرد[۹] در داده ها بهتر است. شکل ۱ یک تک سلول حافظه LSTM را نمایش می دهد. برای نگارشی از LSTM که در این گفتار [۷] بکار برده شده است H با تابع ترکیبی زیر پیاده سازی شده است:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (۷)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (۸)$$

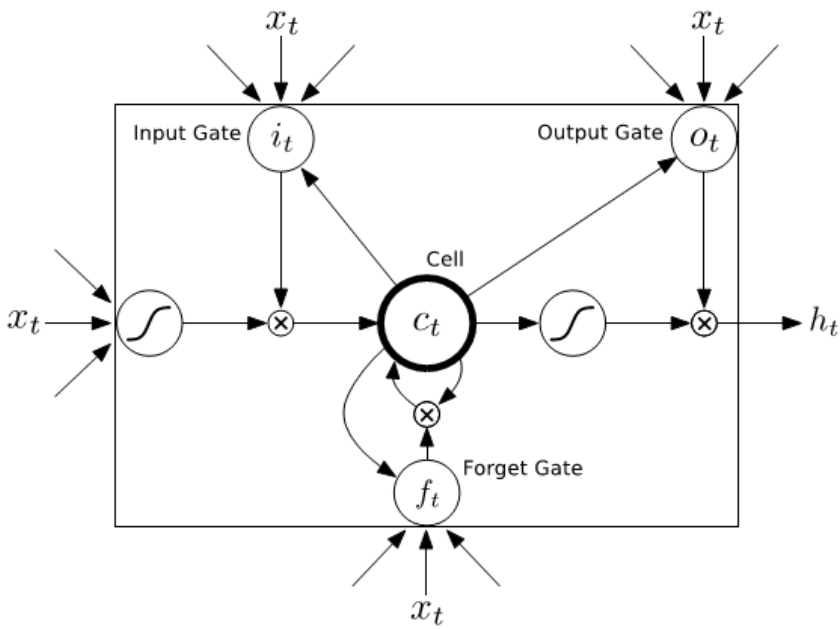
$$c_t = f_t c_{t-1} + i_t \, tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (۹)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_{t-1} + b_o) \quad (۱۰)$$

$$h_t = o_t \tanh(c_t) \quad (۱۱)$$

که σ تابع سیگموید منطقی، و i، f، o و c به ترتیب گیت ورودی، گیت فراموشی، گیت برونداد، سلول و بردارهای فعالسازی ورودی سلول هستند و همه آنها هم اندازه بردار پنهان h هستند. زیرنویس های ماتریس وزن معنای روشنی دارند، برای نمونه W_{hi} ماتریس گیت ورودی-پنهان است، W_{ho} ماتریس گیت ورودی-برونداد است و ... . ماتریس های وزن از بردارهای سلول به گیت (برای نمونه W_{hi}) قطری هستند، بنابراین عنصر m در هر بردار گیت تنها از عنصر m از بردار سلول ورودی دریافت می کند. بایاس ها (که به i، f، c و o افزوده می شود) برای روشن تر بودن نادیده گرفته شده اند.

[۸] Sigmoid
[۹] Long-range dependencies



شکل ۱ : سلول حافظه کوتاه- مدت دراز

الگوریتم پایه LSTM از یک محاسبه گرادیان سفارشی شده تقریبی بهره می برد که به وزن ها اجازه می داد که پس از هر گام زمانی بروز شوند [۱۶]. به هر روی از گرادیان را می توان بدون کم و کاستی و با انتشار رو به پشت از زمان محاسبه کرد، همانگونه که در این پایان نامه بکار برده شده است. یک چالش که در هنگام آموزش LSTM با خود گرادیان این است که گاهی مشتق ها خیلی بزرگ می شوند و به مشکلات شماره ای دامن می زنند. برای جلوگیری این، همه آزمایش ها مشتق زیان را با توجه به ورودی های شبکه به لایه های LSTM قیچی می شوند (پیش از این که تابع های سیگموید و تانژانت هایپربولیک بکار برده شوند) تا آنها در یک بازه از پیش گفته شده جای بگیرند.

۲-۲ پیش بینی متن

داده های متنی گسسته هستند. اگر روی هم K کلاس متنی داشته باشیم و کلاس k در زمان t به شبکه خورانده شود، آنگاه x_t یک بردار به طول K خواهد بود که درایه های آن همه صفر هستند بجز برای kام که یک است. توزیع $\Pr(x_{t+1}|y_t)$ بنابراین یک توزیع چندجمله ای است، که به گونه طبیعی می توان با یک تابع softmax در لایه خروجی پارامتری کرد:

$$\Pr(x_{t+1} = k|y_t) = y_t^k = \frac{\exp(\hat{y}_t^k)}{\sum_{k'=1}^{K} \exp(\hat{y}_t^{k'})} \quad (۱۲)$$

با جایگزینی در برابری (٦) می بینیم که

$$L(x) = -\sum_{t=1}^{T} \log y_t^{x_{t+1}} \quad (۱۳)$$

$$\Rightarrow \frac{\partial L(x)}{\partial \hat{y}_t^k} = y_t^k - - \delta\, k, x_{t+1} \quad (۱٤)$$

تنهای چیزی که برای تصمیم گیری به جا می ماند این است که کدام مجموعه از کلاس ها را بکار بگیریم. در بیشتر موارد، پیش بینی متن (که همچنین به آن مدل سازی زبان گفته می شود) در سطح واژه انجام می شود. بنابراین K شمار واژگان در فرهنگ واژگان است. این می تواند برای کارهای واقعی مشکل ساز باشد، که شمار واژگان (دربردارنده ترکیب ها، نام های درست و ...) بیشتر زمان ها از ۱۰۰،۰۰۰ فراتر می رود. همچنین نیازمندبودن پارامترهای بسیار برای مدل، داشتن کلاس های خیلی زیاد خواستار اندازه بسیاری از داده های آموزشی است تا به گونه ای درخور محتواهای ممکن را برای واژگان پوشش دهد. درباره مدل های softmax، دشواری دیگر هزینه محاسباتی بالای ارزیابی همه نماها در طی آموزش است (گرچه چندین روش طراحی شده اند تا آموزش لایه های بزرگ softmax را کاراتر کنند، که دربردارنده مدل های برپایه درخت [۲۵ و ۲۳]، تقریب زنی های درجه پایین [۲۷] و مشتق های تصادفی [۲٦]). افزون بر این، مدل های