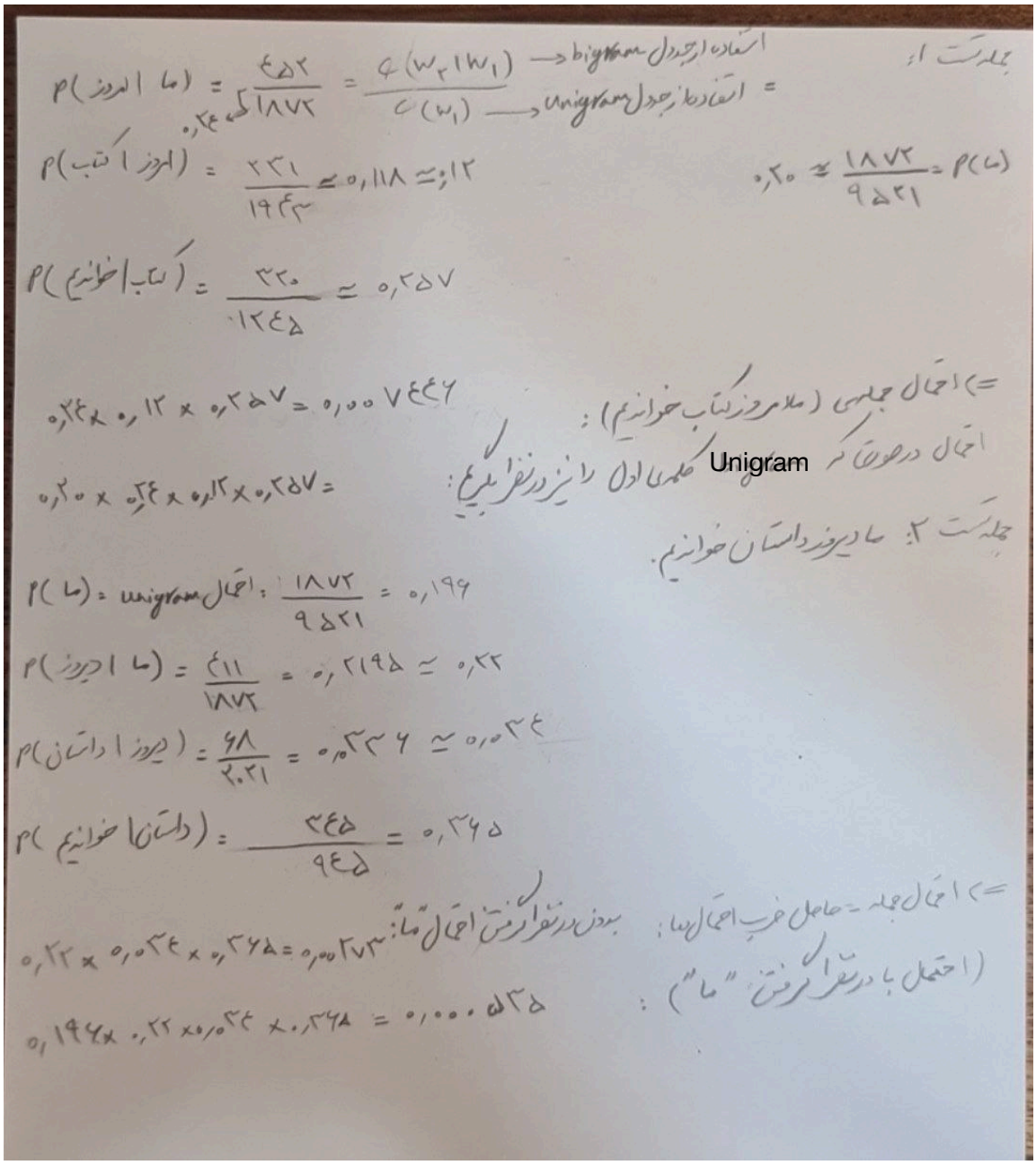
سیده شکیبا انارکی فیروز ۹۹۴۴۲۰۴۷

تمرین سوم درس پردازش زبان ها

(1



۲) می دانیم: $P(B|A) = \frac{P(A,B)}{P(A)}$ و $P(A,B) = P(A)P(B|A)$ ①

(Chain Rule) همچنین از تساوی variable ها را زیاد کنیم به طور کلی داریم:

$$P(x_1, \dots, x_n) = P(x_1) P(x_2|x_1) P(x_3|x_1, x_2) \dots P(x_n|x_1, \dots, x_{n-1})$$

$\Rightarrow$ پس احتمال رخ دادن یک جمله به صورت دنباله‌ای از کلمات w_1 تا w_n را نیز می‌توان به روش مشابه بالا نشان داد:

$$P(w_1, \dots, w_n) = P(w_1) P(w_2|w_1) \dots P(w_n|w_1, \dots, w_{n-1})$$

۳) برای به دست آوردن احتمال یک جمله از w_1 تا w_n طبق قانون زنجیره‌ای احتمال داریم: (راه دیگر)

$$P(w_{1:n}) = P(w_1) P(w_2|w_1) P(w_3|w_{1:2}) \dots P(w_n|w_{1:n-1}) =$$

$$\prod_{k=1}^{n} P(w_k|w_{1:k-1})$$

منظور از $w_{1:k}$: w_1 و w_2 و ... و w_k پشت سر هم است.

طبق بیش n-gram ها می‌توان گفت در کمین هر کلمه برچسب آینده به کل کلمات جمله قبل از آن وابسته‌اند می‌توان به صورت تقریبی وابسته به کمین کلمه‌ی قبلش در نظر گرفت. (برای bigram)

بدین ترتیب می‌توان گفت: $P(w_n|w_{1:n-1}) \approx P(w_n|w_{n-1})$ ← Markow assumption

بدین ترتیب با جایگذاری این احتمالات در معادله‌ی اول داریم:

$$P(w_{1:n}) \approx \prod_{k=1}^{n} P(w_k|w_{k-1})$$

۳) a) در مرحله‌ی اول فقط ۲ گزینه داریم و beam search در این مرحله تاثیری ندارد:

"network" (-۰,۳۴) ← probability log و "neural" (-۰,۴۸)

۳) b) تمامی حالت‌های ممکن را بررسی کرده و در نهایت ۲ تای برتر را نگه می‌داریم.

"neural network" (-۱,۲۸)

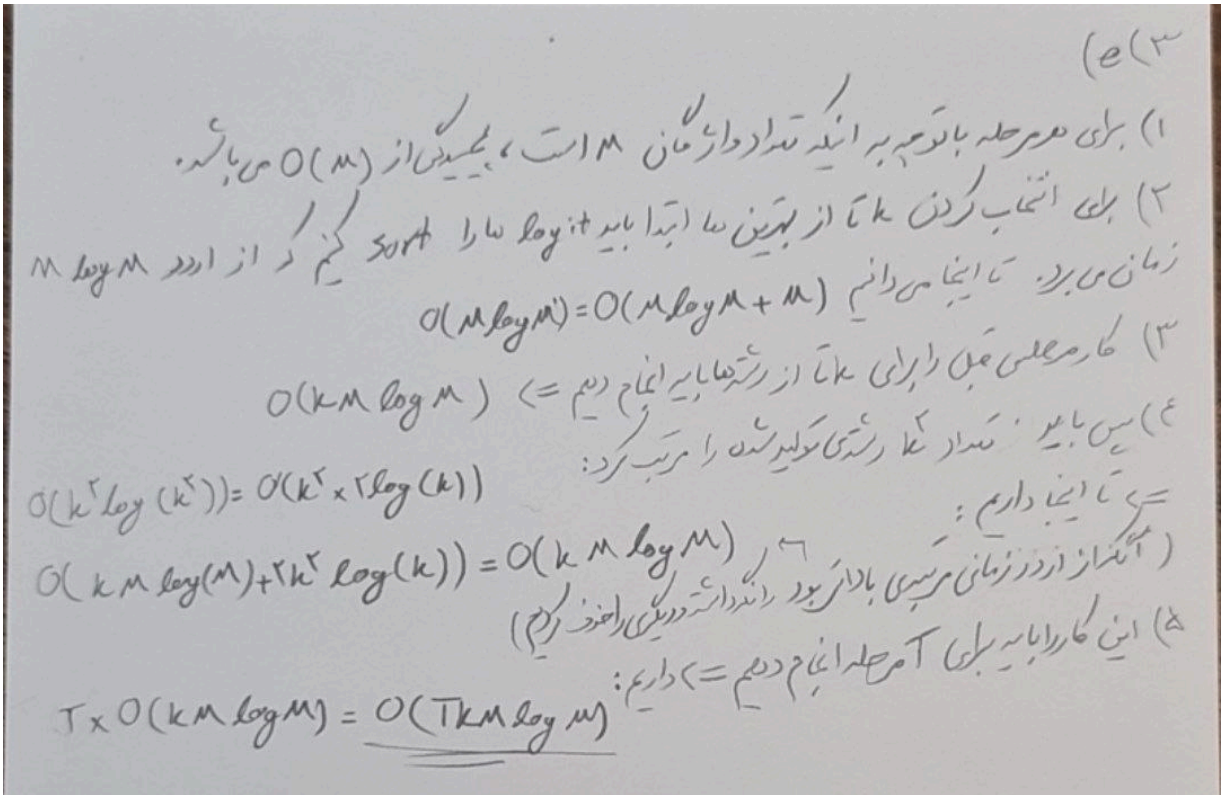
"network neural" (-۰,۷۳ - ۰,۶ = -۱,۳۳) ← log probability

۳) c) در مرحله‌ی سوم دنباله‌های سه کلمه‌ای را با استفاده از دو عبارت برگزیده در مرحله‌ی دوم می‌سازیم. سپس مانند مرحله‌ی قبل ۲ عبارت برتر را انتخاب می‌کنیم که به صورت زیر است.

"Neural network network": (-۰,۴۵ + -۰,۶ + -۰,۶ = -۱,۸۵)

"Network neural network": (-۰,۷۳ - ۰,۶ - ۰,۶ = -۱,۹۳)

۳) d) خیر این اتفاق نمی‌افتد زیرا دنباله‌ای که بیشترین احتمال را دارد "neural neural neural" است با کلیه احتمال -۱,۶۴ = -۰,۰۱ + -۰,۸ + -۰,۴۵ . اما در مرحله‌ی دوم از گزینه‌های مولد توجه خارج می‌شود زیرا احتمال "neural neural" جزو دو عبارت اول با طول ۲ نیست.



(A.4

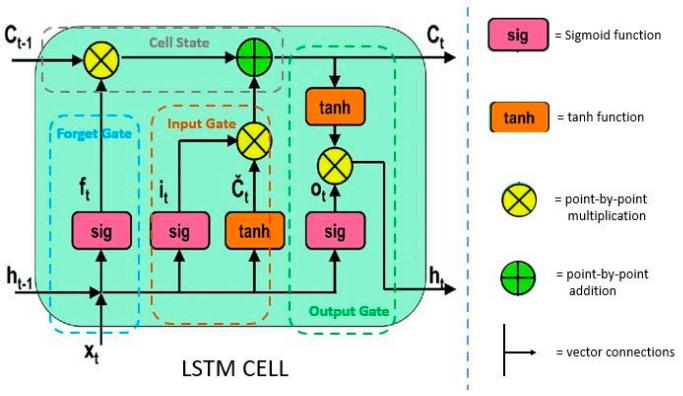
وقتی که فقط گیت Forget یک LSTM را حفظ کنیم و گیت‌های Input و Output را حذف کنیم، تغییرات زیادی در عملکرد شبکه اتفاق می‌افتد. برای درک بهتر این تغییرات، ابتدا به ساختار LSTM و نقش هر گیت در آن نگاهی می اندازیم.

LSTM یک نوع خاص از شبکه‌های عصبی بازگشتی (RNN) است که برای حفظ اطلاعات بلندمدت و جلوگیری از مشکل محو شدن گرادیان (Vanishing Gradient) در RNN طراحی شده است.

ساختار LSTM شامل چهار بخش اصلی است:

1. گیت فراموشی (Forget Gate): این گیت تصمیم می‌گیرد که کدام اطلاعات از حافظه‌ی سلول (Cell Memory) قبلی باید فراموش شود و کدام اطلاعات باقی بمانند.
2. گیت ورودی (Input Gate): این گیت تصمیم می‌گیرد که چه اطلاعات جدیدی باید به حافظه‌ی سلول اضافه شود.
3. حافظه‌ی سلول (Cell Memory): این بخش حافظه‌ی میان‌مدت است که اطلاعات در طول زمان در آن ذخیره می‌شود.
4. گیت خروجی (Output Gate): این گیت تصمیم می‌گیرد که اطلاعات چه قدر از حافظه‌ی سلول باید به عنوان خروجی شبکه استفاده شود.

در تصویر زیر ساختار توابع استفاده شده در این گیت ها را مشاهده میکنیم:



LSTM CELL

اگر فقط گیت Forget را حفظ کنیم و گیت‌های Input و Output را حذف کنیم، این به این معنی است که شبکه دیگر نمی‌تواند تصمیم بگیرد که کدام اطلاعات باید به حافظه‌ی سلول اضافه شود و کدام اطلاعات باید برای تولید خروجی استفاده شوند.
بدین ترتیب حافظه‌ی سلول به طور مداوم تغییر نخواهد کرد، زیرا اطلاعات جدید به آن اضافه نمی‌شود و همچنین بدون گیت خروجی، شبکه قادر به انتقال اطلاعات از حافظه‌ی سلول به خروجی نخواهد بود و خروجی ثابت و بی‌تغییر می شود. پس ساختار شبکه بسیار ساده‌تر شده و نمی‌تواند به خوبی اطلاعات را جذب کند و خروجی‌های مختلفی تولید کند. این تغییرات باعث افت شدید عملکرد شبکه و توانایی آن در یادگیری و پردازش داده‌ها خواهد شد.

(B.4

وقتی که مقدار گیت Forget را به صفر تنظیم می‌کنیم، شبکه LSTM دیگر هیچ اطلاعاتی را از حافظه ی سلول (Cell Memory) خود ذخیره نمی‌کند و شبیه RNN می شود. این تغییرات می‌تواند تأثیرات جالبی بر روی عملکرد شبکه داشته باشد:

- از دست دادن سازگاری: شبکه توانایی تنظیم پویا مقدار اطلاعات را بر اساس ورودی و زمینه فعلی از دست می دهد. این باعث کاهش انعطاف پذیری آن در رسیدگی به انواع مختلف توالی می شود.
- ناپدید شدن گرادیان: فراموش کردن مداوم تمام اطلاعات می تواند منجر به ناپدید شدن گرادیان در طول آموزش شود. این امر یادگیری وابستگی های طولانی مدت در داده ها را برای شبکه دشوار می کند.
- کاهش توانایی یادگیری: با فراموشی محدود، شبکه تلاش می کند تا بر مرتبط ترین اطلاعات تمرکز کند و وضعیت داخلی خود را به طور موثر به روز کند. این مانع از ظرفیت کلی یادگیری آن می شود.

(C.4

مزایا:

- بهبود نمایش ویژگی‌های پیچیده:

با افزایش تعداد لایه‌ها، شبکه قادر به یادگیری ویژگی‌های پیچیده‌تر و سطوح بالاتری از داده می‌شود. هر لایه LSTM به شبکه اجازه می‌دهد که ویژگی‌های جدیدی از داده را استخراج کرده و به لایه‌های بالاتر منتقل کند.

- افزایش توانایی یادگیری داده‌ها:

شبکه LSTM با لایه‌های بیشتر می‌تواند الگوهای زمانی پیچیده‌تری را یاد بگیرد. این امر می‌تواند بهترین نتایج را در وظایفی مانند ترجمه ماشینی یا پیش‌بینی دنباله‌ای (sequence prediction) فراهم کند.

- کاهش مشکل محو شدن گرادیان:

با افزایش عمق شبکه (تعداد لایه‌ها)، ممکن است مشکل محو شدن گرادیان که معمولاً در شبکه‌های عمیق به وجود می‌آید، کاهش یابد. این به این معنی است که شبکه قادر به بهترین استفاده از اطلاعات تاریخی در داده‌های توالی خواهد بود.

معایب:

- افزایش پیچیدگی و زمان آموزش:

با افزایش تعداد لایه‌ها، پیچیدگی شبکه افزایش می‌یابد و زمان آموزش نیز افزایش می‌یابد. شبکه‌های عمیق‌تر نیاز به تعداد بیشتری پارامتر (وزن‌ها) دارند و به دنباله‌های طولانی‌تری نیاز دارند تا آموزش ببینند.

- مشکل Overfitting:

شبکه‌های عمیق‌تر ممکن است به سرعت به داده‌های آموزشی بیش‌برازش کنند یعنی ویژگی‌های غیرضروری یا نویز در داده‌های آموزش را یاد بگیرند و عملکرد خوبی بر روی داده‌های تازه‌ای (که در فاز آموزش دیده نشده‌اند) ارائه ندهند.

- پیچیدگی بالا برای تنظیم پارامترها:

با افزایش تعداد لایه‌ها، تنظیم پارامترهای شبکه (مانند نرخ یادگیری و نرخ Dropout) نیاز به دقت و دانش بیشتری دارد.

(I.5

افزایش متغیر (استفاده از نقاط داده گذشته بیشتر):

مزایا:

- ثبت روندهای بلندمدت: ترکیب داده های بیشتر به LSTM اجازه می دهد تا روندهای بازار گسترده تری را ثبت کند و به طور بالقوه چرخه ها یا الگوهای بلندمدتی را شناسایی کند که ممکن است در بازه زمانی کوتاه تر قابل مشاهده نباشند.
- بهبود استحکام: یک مجموعه داده بزرگتر می تواند مدل را در برابر نویز و نوسانات در داده ها قوی تر کند و منجر به پیش بینی های پایدارتر و قابل اعتمادتر شود.

معایب:

- Overfitting بیش از حد: گنجاندن داده‌های زیاد می‌تواند منجرoverfitting شود، مدل الگوهای خاصی را در داده‌های آموزشی و نویز ها به خاطر می‌سپارد و در تعمیم به داده‌های دیده نشده عملکرد ضعیف خواهد داشت.
- افزایش زمان آموزش: آموزش LSTM با مجموعه داده بزرگتر به منابع محاسباتی بیشتری نیاز دارد و زمان بیشتری می برد.

کاهش متغیر (با استفاده از نقاط داده گذشته کمتر):

مزایا:

برعکس قسمت قبل کاهش متغیرمی تواند باعث کاهش زمان آموزش - کاهش Overfitting شود.

معایب:

برعکس قسمت قبل می توان به از دست دادن روندهای بلندمدت و امکان پیش‌بینی‌های نادرست وافزایش حساسیت به نویز اشاره کرد.

Used links:

https://aroussi.com/post/python-yahoo-finance

https://matplotlib.org/stable/gallery/color/named_colors.html

https://www.youtube.com/watch?v=b61DPVFX03I

https://medium.com/@CallMeTwitch/building-a-neural-network-zoo-from-scratch-the-long-short-term-memory-network-1cec5cf31b7