



## گزارش پروژه دوم درس شبکه های عصبی مصنوعی

محمدرضا یادگاری

### ۱ مقدمه

در این پروژه قصد داریم با شبکه های عصبی بازگشتی و برخی از مدل های مشهور آن آشنا بشویم و با استفاده از این مدل ها نتایجی روی دیتاست توییتز فارسی بدست بیاوریم. بسیاری از شبکه های عصبی مصنوعی، شبکه های پیش خور<sup>۱</sup> هستند. یعنی سیگنال در این شبکه ها فقط در یک جهت از لایه ورودی، به لایه های مخفی و سپس به لایه خروجی حرکت می کند و داده های قبلی به حافظه سپرده نمی شوند. اما شبکه های عصبی بازگشتی دارای لایه بازخورد هستند که در آن خروجی شبکه به همراه ورودی بعدی، به شبکه بازگردانده می شود. این لایه یا لایه های بازخورد باعث می شوند که شبکه به نوعی دارای "حافظه" باشد. به دلیل همین حافظه داخلی، شبکه های عصبی بازگشتی برای داده های متوالی، مانند سری های زمانی، گفتار، متن، داده های مالی، صوت، ویدئو و... عملکرد خوبی از خود نشان می دهد. در واقع، شبکه های عصبی بازگشتی، می توانند نسبت به سایر الگوریتم ها، درک عمیق تری از توالی و دانش پس زمینه ورودی ها کسب کند.

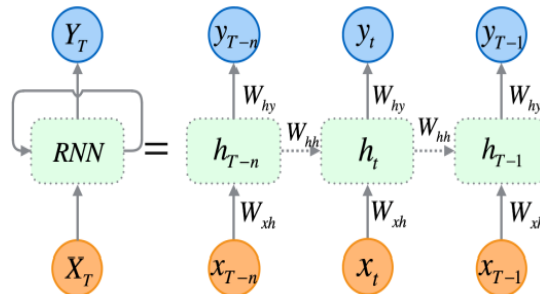
---

<sup>1</sup>Feed Forward

## ۲ مدل های به کار گرفته شده در این پروژه

### ۱.۲ RNN

از اولین و قدیمی ترین مدل های شبکه عصبی بازگشتی هستند و قدرت کمتری نسبت به شبکه های LSTM و GRU دارند. بزرگترین مشکل این شبکه ها ، محوشدگی گرادینان<sup>۲</sup> و انفجار گرادینان<sup>۳</sup> است.



شکل ۱: شماتیک یک شبکه RNN

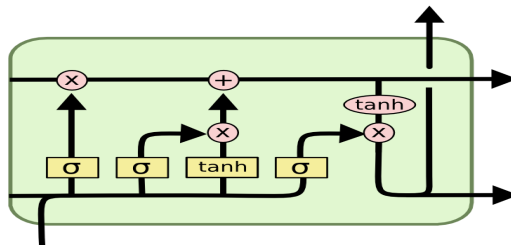
### ۲.۲ LSTM

یکی از مواردی که شبکه RNN به خوبی از پس آن برنمی آید، زمانی است که حالت فعلی به حالتی در گذشته بستگی دارد که زمان زیادی از آن گذشته است.

LSTM ، توسعه ای از شبکه عصبی بازگشتی به حساب می آید که در واقع حافظه را گسترش می دهد. در نتیجه، این مدل برای مواردی بسیار مهم است که لازم باشد از تجربه هایی که زمان طولانی از آنها گذشته است هم یادگیری اتفاق بیفتد. LSTM ها به RNN این امکان را می دهند که ورودی ها را پس از یک زمان طولانی به خاطر بسپارند، تقریباً شبیه حافظه کامپیوتر. LSTM می تواند اطلاعات را از حافظه اش بخواند، بنویسد یا آنها را پاک کند.

<sup>2</sup>Vanishing Gradient

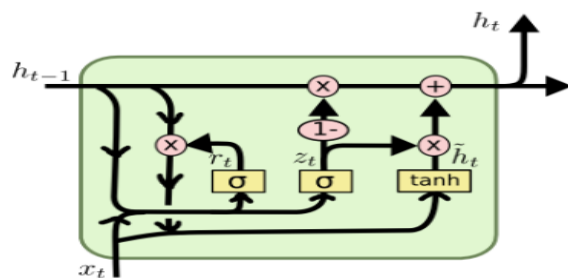
<sup>3</sup>Exploding Gradients



شکل ۲: شماتیک یک سلول LSTM

### ۳.۲ GRU

این نوع از شبکه عصبی بازگشتی، مانند LSTM برای این طراحی شده است که مشکل کوتاه بودن حافظه RNN را حل کند. اما در شبکه بازگشتی گیتی، به جای سلول وضعیت، از لایه‌های پنهان برای مدیریت و دسته‌بندی اطلاعات استفاده می‌شود. و به جای سه دروازه، دارای دو دروازه است: دروازه بازنشانی و دروازه به روز رسانی. مانند LSTM ها، این دو دروازه هستند که تعیین می‌کنند کدام اطلاعات مفید هستند و کدام‌ها مفید نیستند. با توجه به ساختار گفته شده، تعداد پارامترهای یادگیری یک شبکه GRU از تعداد پارامترهای یادگیری شبکه LSTM کمتر است و این یک مزیت برای GRU محسوب می‌شود.



شکل ۳: شماتیک یک سلول GRU

### ۳ پیش پردازش داده ها

مجموعه داده ما شامل ۲۴۹۹۸۱ جمله فارسی است که این جملات توسط کاربران فارسی زبان در توییتر نوشته شده اند. دیتایی که ما دریافت کردیم تقریباً پاک شده بود و من فقط هرچیز غیر از حرف و عدد را از دیتاست پاک کردم. سپس هر جمله را در یک لیست ریخته که هر عضو آن یک کلمه در جمله است. در ادامه به وسیله متد tokenizer برای هر کلمه داخل لیست یک عدد نسبت می‌دهیم در نتیجه با این کار کلمات را به اعداد تبدیل می‌کنیم تا بتوانیم با آنها کار بکنیم.

سپس به وسیله تابع partitionlines داده ها را به گونه ای آماده میکنیم که مناسب این باشد که یک کلمه را به شبکه عصبی بدهیم و شبکه عصبی بتواند کلمه بعدی را پیشبینی بکند. در انتها قابل ذکر است که به دلیل حجم بالای داده ها، نتوانستیم از کل داده ها استفاده کنیم زیرا colab در حین اجرای کد ها کرش می کرد. در نتیجه مجبور شدیم از ۱۰۰۰۰ جمله استفاده کنیم که این ۱۰۰۰۰ جمله تبدیل به ۹۳۶۹۲ تا داده ورودی (کلمه) برای دادن به شبکه عصبی شد. سپس به نسبت ۸۰ به ۲۰ داده های ترین و تست را جدا کردم و داده ها کاملاً آماده کار برای شبکه عصبی شدند.

## ۴ مدل ها

در این تمرین از مدل های RNN،LSTM،GRU استفاده شده است. در ادامه نتایج هرکدام از مدل ها آورده شده است و در آخر نیز مقایسه ای کلی صورت می گیرد.

## ۵ نتایج

### ۱.۵ LSTM

اولین مدل ما یک مدل LSTM است که شامل ۵۰۰ استیت مخفی (سلول) است. دقت مدل در آخرین Epoch روی داده های ترین و تست (Validation) گزارش شده است. نتایج آزمایش با این مدل در جدول زیر آمده است:

Learning Rate	Epoch	Optimizer	Dropout	Batch Size	Accuracy	Val-Accuracy	Trainable Params
0.001	30	Adam	0.2	128	0.8171	0.127	1,909,128

Table 1: Result Of LSTM

همانطور که مشخص است مدل اصلاً روی داده های Validation دقت خوبی ندارد ولی روی داده های ترین دقتش خوب است. این نشان دهنده Overfitting زیاد مدل می باشد. در شکل زیر یک نمونه از پیشبینی مدل روی داده های تست را می بینیم:

✓ [47] 1 words[100]  
0s

['سلب', 'تابعت', 'از', 'نظام', 'جمهوری', 'السلامی', 'ترین', 'مبارزه']

✓ [48] 1 predicts(5, words[100][0])  
1s

input = سلب, predicted = [('0.99922967', 'تابعت'), ('0.45605597', 'از'), ('0.99403185', 'نظام'), ('0.9916877', 'جمهوری'), ('0.99993205', 'السلامی')]

شکل ۴: پیشبینی مدل LSTM روی یک نمونه داده تست

## ۲.۵ GRU

دومین مدل ما یک GRU با ۵۰۰ استیت مخفی (سلول) است. دقت مدل در آخرین Epoch روی داده های ترین و تست (Validation) گزارش شده است. نتایج آزمایش با این مدل در جدول زیر آمده است:

Learning Rate	Epoch	Optimizer	Dropout	Batch Size	Accuracy	Val-Accuracy	Trainable Params
0.001	15	Adam	0.2	128	0.5712	0.1308	1,574,000

Table 2: Result Of GRU

همانطور که در مدل قبل دیدیم مدل ما دارای Overfitting زیادی بود و تقریباً از Epoch ۱۰ به بعد تغییر خاصی روی دقت داده های Validation رخ نداد. در نتیجه تعداد Epoch ها را به ۱۵ کاهش دادیم. می بینیم که نتایج این مدل کمی بهتر است. همچنین تعداد پارامترهای یادگیری این مدل کمتر است که انتظار داشتیم اینگونه باشد اما همچنان Overfitting زیادی داریم. در شکل زیر نمونه ایی از پیشبینی مدل روی یک کلمه (همان کلمه ایی که مدل LSTM پیشبینی کرده بود) از داده های تست می بینیم:

```
[69] 1 wordss[100]
```

['سلب', 'تبعیت', 'از', 'نظام', 'جمهوری', 'اسلامی', 'ترین', 'مبارزه']

```
[70] 1 predictg(5, wordss[100][0])
```

input = سلب, predicted = [(0.99048084, 'تبعیت'), (0.34082583, 'است'), (0.53297913, 'و'), (0.73799336, 'تاریخ'), (0.37329602, 'بوده')]

شکل ۵: پیشبینی مدل GRU روی یک نمونه داده تست

## ۳.۵ RNN

سومین و آخرین مدل ما یک RNN با ۵۰۰ لایه مخفی می باشد. دقت مدل در آخرین Epoch روی داده های ترین و تست (Validation) گزارش شده است. نتایج آزمایش با این مدل در جدول زیر آمده است:

Learning Rate	Epoch	Optimizer	Dropout	Batch Size	Accuracy	Val-Accuracy	Trainable Params
0.001	15	Adam	0.2	128	0.2559	0.1131	943,500

Table 3: Result Of RNN

انتظار داشتیم که نتایج این مدل تفاوت زیادی با مدل های قبل نداشته باشد زیرا نیازی به حافظه بلند مدت برای این تسک خاص نداشتیم. همچنین واضح است که تعداد پارامترهای یادگیری این مدل باید کمتر از تعداد پارامترهای یادگیری دو مدل گذشته باشد. مدل RNN هم دچار Overfitting شده و جواب خوبی به ما

نداده اما مقدار Overfitting آن کمتر است زیرا تعداد پارامترهای یادگیری آن کمتر است.  
در شکل زیر نتیجه مدل روی کلمه ایی از داده تست که روی دو مدل قبل هم نمایش داده شد را می بینید:

```
[78] 1 wordss[100]
```

```
['سلب', 'تلبیت', 'از', 'نظام', 'جمهوری', 'اسلامی', 'ترین', 'مبارزه']
```

```
[79] 1 predictr(5, wordss[100][0])
```

```
input = سلب, predicted = [('0.9995702', 'تلبیت'), ('0.28245297', 'خود'), ('0.78951687', 'را'), ('0.20800145', 'به'), ('0.09701262', 'از')]
```

شکل ۶: پیشبینی مدل RNN روی یک نمونه داده تست

## ۶ جمع بندی

انتظار داشتیم که شبکه های مبتنی بر LSTM و GRU نسبت به RNN ها قوی تر عمل کنند و تا حدودی در این تمرین هم مشخص بود. همچنین Overfitting به ترتیب روی شبکه RNN کمتر از GRU و کمتر LSTM بود و دلیل آن نیز تعداد پارامترها بود که در این سه شبکه به ترتیب RNN از GRU کمتر و GRU از LSTM کمتر بود. اگر میتوانستیم تعداد داده ها را بالا ببریم، احتمالاً Overfitting کم می شد و به جواب های بهتری می رسیدیم زیرا مدل ها را با سلول های کمتر هم آزمایش کردم اما Val-Accuracy برای آنها نیز در همین حدود Val-Accuracy مدل های ذکر شده در گزارش بود و تنها تفاوت آن این بود که چون مدل ساده تر شده بود، در نتیجه به خوبی نمی توانست روی داده های ترین به جواب های خوبی برسد و Accuracy آن پایین تر بود. پس احتمالاً تنها راهی که به جواب های خوب برسیم این است که تعداد داده ها را افزایش دهیم که متأسفانه این امکان برای من وجود ندارد زیرا چندین بار Colab حین اجرای کد کرش کرد.