

## Question 1

- شبکه های RNN حالت دار (stateful RNNs) دسته ای از شبکه های بازگشتی هستند که حالت داخلی شبکه برای همه Batch ها حفظ می شود. به این معنی که وقتی شبکه یک Batch را به عنوان ورودی دریافت می کند، وضعیت Batch قبلی به همراه Batch فعلی به شبکه منتقل می شود. این ویژگی به شبکه این اجازه را می دهد که بتواند سطح قابل توجهی از اطلاعات را از Batch ها ذخیره کند که اغلب برای آن دسته از داده هایی که بین هر Batch از آن ها وابستگی بسیار وجود دارد، کمک کننده و مفید می باشد.
- از طرف دیگر، شبکه های stateless RNN، فاقد این حالت ذخیره سازی می باشند و شبکه، هیچ اطلاعاتی رو بین Batch ها ذخیره سازی نمی کند. به این معنی که وقتی یک Batch جدید به عنوان ورودی به شبکه داده می شود، شبکه به دلیل این که هیچ آگاهی نسبت به Batch های قبلی رو ذخیره سازی نکرده، باید از ابتدا و بدون هیچ اطلاعاتی، Batch ورودی جدید رو process کند. این دسته از شبکه های بازگشتی را می توان به راحتی موازی سازی و یا همچنین مقیاس بندی (چند برابر کردن بزرگی شبکه) کرد، اما اصلا قادر نخواهند بود که وابستگی های میان Batch های داده ها را ثبت کنند.
- فواید شبکه های Stateful RNN:
  - می توانند اطلاعات ورودی های قبلی خودشان را ذخیره کنند که این برای تسک هایی که شبکه برای پیش بینی نیاز حتمی به اطلاعات ورودی های قبلی خود دارد، بسیار مفید و مهم می باشد.
  - آنها می توانند از نظر حافظه و محاسبات کارآمدتر باشند، زیرا بردار حالت پنهان را می توان برای چندین توالی ورودی بدون نیاز به ذخیره و پردازش همه ورودی های قبلی به طور جداگانه استفاده کرد.
- معایب شبکه های Stateful RNN:
  - آموزش و دیباگ کردن آنها ممکن است دشوار باشد، زیرا بردار حالت پنهان در هر مرحله زمانی بر اساس وضعیت پنهان قبلی و ورودی فعلی به روز می شود، که درک نحوه پردازش ورودی ها توسط شبکه را دشوارتر می کند.
  - می توانند به ترتیب توالی های ورودی حساس باشند، که اگر ترتیب ورودی ها ثابت نباشد یا شبکه نیاز به پردازش ورودی هایی داشته باشد که از نظم خارج می شوند، می تواند مشکل ساز شود.
- فواید شبکه های Stateless RNN:
  - آموزش و دیباگ کردن آنها ساده است، زیرا هر ورودی به طور مستقل پردازش می شود و شبکه هیچ اطلاعاتی را از ورودی های قبلی حفظ نمی کند.

- به ترتیب توالی های ورودی حساس نیستند، که می تواند برای کارهایی مفید باشند که ترتیب ورودی ها ثابت نیست یا ممکن است ورودی ها از نظم خارج شوند.
- معایب شبکه های Stateless RNN:
  - نمی توانند اطلاعات ورودی های قبلی را حفظ کنند، که می تواند برای کارهایی که شبکه باید این اطلاعات را به خاطر بسپارد و از آنها استفاده کند، مشکل ساز باشد.
  - ممکن است به حافظه و محاسبات بیشتری نیاز داشته باشند، زیرا شبکه باید تمام ورودی های قبلی را جداگانه ذخیره و پردازش کند تا خروجی یک توالی ورودی مشخص را تولید کند.

## Question2:

- Encoder-Decoder RNNs و Plain sequence-to-sequence RNNs، هر دو از انواع شبکه های عصبی ای هستند که می توانند برای انجام تسک هایی مثل ترجمه ماشینی، خلاصه سازی و سیستم های گفتگو استفاده شوند.
- تفاوت اصلی بین این دو نوع از شبکه های بازگشتی در روشی هست که این دو شبکه رشته های ورودی و خروجی رو encode و decode می کنند.
- در یک شبکه encoder-decoder RNN، رشته ورودی در ابتدا توسط یک encoder RNN رمزگذاری می شود و در نتیجه این رمزگذاری، رشته ورودی به یک بردار با طول ثابت به نام "context vector" تبدیل می شود. سپس این بردار از یک decoder RNN عبور داده می شود که باعث تولید خروجی نهایی شبکه می شود. در واقع معماری encoder-decoder این اجازه را به شبکه می دهد که رشته ورودی رو به یک بردار با طول ثابت تبدیل کند که این کار در تسک هایی که طول رشته ورودی بسیار بزرگ می باشد، مفید است.
- از طرف دیگر، شبکه های Plain Seq2Seq RNN، رمزگذار و رمزگشای جدا ندارند و در مقابل، رشته ورودی را از یک شبکه بازگشتی که Train شده تا رشته های ورودی رو به صورت مستقیم به خروجی شبکه map کند، عبور داده می شود و این معماری ساده تر از معماری encoder-decoder می باشد اما کارآمدی آن ها در برابر ورودی هایی با طول زیاد رو ندارد.
- به طور کلی encoder-decoder RNN ها برای تسک automatic translation ترجیح داده می شوند. چرا که آن ها به خوبی با ویژگی های زبان طبیعی سازگار هستند.
- یکی از دلایل اصلی کارآمدی شبکه های encoder-decoder RNN برای ترجمه، توانایی آن ها در مقابل رشته های ورودی با طول های متفاوت می باشد. جملات زبان طبیعی از لحاظ طول رشته می توانند بسیار متنوع باشند و از آنجا که شبکه های encoder-decoder، تمام ورودی ها رو به یک بردار با طول ثابت تبدیل می کند، شبکه شامل این معماری قادر به process کردن هر نوع ورودی می باشد.

### :Question3

- در یک سلول RNN با یک gated learning component، مانند سلول LSTM یا GRU، مکانیسم درگاهی، جریان اطلاعات را به داخل و خارج از cell state کنترل می‌کند. مقادیر درگاه معمولاً مقادیر باینری (0 یا 1) یا مقادیر پیوسته بین 0 و 1 هستند، بسته به ورودی‌های خاص و وضعیت قبلی سلول ممکن است یکی از این دو نوع رو داشته باشند. تابع فعال سازی مکانیسم دروازه را می‌توان برای آزمایش رفتارهای مختلف تغییر داد، اگرچه توابع فعال سازی مختلف ممکن است ویژگی‌های متفاوتی داشته باشند و ممکن است کم و بیش برای یک کار خاص مناسب باشند.
- در یک سلول LSTM، درگاه ورودی و درگاه فراموشی توابع فعال سازی سیگموئیدی هستند که مقادیر خروجی بین 0 و 1 دارند. دروازه ورودی این که کدام مقادیر از ورودی اجازه عبور به cell state را دارند، کنترل می‌کند، در حالی که درگاه فراموشی، کنترل می‌کند که کدام مقادیر از cell state قبلی کنار گذاشته می‌شوند. مقادیر درگاه ورودی و درگاه فراموشی بسته به ورودی‌های خاص و وضعیت قبلی سلول باید بین 0 تا 1 باشد. اگر درگاه ورودی مقدار 0 داشته باشد، هیچ مقداری از درگاه ورودی به cell state وارد نمی‌شود. اگر درگاه ورودی مقدار 1 داشته باشد، یعنی به تمام مقادیر ورودی اجازه می‌دهد تا به cell state منتقل شوند. به طور مشابه، اگر درگاه فراموشی مقدار 0 داشته باشد، مانع از عبور مقادیر سلول قبلی می‌شود و اگر مقدار آن 1 باشد، به تمام مقادیر سلول قبلی اجازه عبور می‌دهد.
- در یک سلول واحد بازگشتی درگاهی (GRU)، درگاه بازنشانی (reset) و درگاه به‌روزرسانی (update)، توابع فعال‌سازی سیگموئید هستند که مقادیر خروجی بین 0 و 1 دارند. درگاه بازنشانی کنترل می‌کند که کدام مقادیر از حالت قبلی کنار گذاشته می‌شوند، در حالی که درگاه به‌روزرسانی کنترل می‌کند در حالی که درگاه به‌روز رسانی کنترل می‌کند که کدام مقادیر از ورودی مجاز به به‌روز رسانی حالت قبلی هستند. مقادیر درگاه بازنشانی و درگاه به‌روز رسانی بسته به ورودی‌های خاص و وضعیت قبلی سلول باید بین 0 تا 1 باشد. اگر درگاه بازنشانی دارای مقدار 0 باشد، عبور تمام مقادیر وضعیت قبلی را از مسدود می‌کند و اگر مقدار آن 1 باشد، به تمام مقادیر حالت قبلی اجازه عبور می‌دهد. به همین ترتیب، اگر درگاه به‌روز رسانی مقدار 0 داشته باشد، تمام مقادیر ورودی را از به‌روز رسانی حالت قبلی منع می‌کند و اگر مقدار آن 1 باشد، به تمام مقادیر ورودی اجازه می‌دهد تا حالت قبلی را به‌روز کنند.

### :Question4

(4)

$$t=1: w_3 f(w_2 h_0 + b_2) + b_3 = 0$$

$$t=2: w_3 f(w_2 h_1 + b_2) + b_3 = 0$$

$$t=3: w_3 f(w_2 h_2 + b_2) + b_3 = 0$$

$$t=4: w_3 f(w_1 + w_2 h_3 + b_2) + b_3 = 1$$

$$t=5: w_3 f(w_1 + w_2 h_4 + b_2) + b_3 = 1$$

$$t=6: w_3 f(w_2 h_5 + b_2) + b_3 = 1$$

$$t=7: w_3 f(w_2 h_6 + b_2) + b_3 = 1$$

$$t=8: w_3 f(w_1 + w_2 h_7 + b_2) + b_3 = 1$$

for simplification, assume that  $b_3 = 0$ , then:

$$\bullet w_3 f(w_2 h_0 + b_2) = 0 \begin{cases} \rightarrow w_3 = 0 \times \\ \rightarrow f(w_2 h_0 + b_2) = 0 = h_1 \end{cases}$$

$$\rightarrow \boxed{w_2 h_0 + b_2 < 0} \rightarrow h_1 = h_2 = h_3 = 0$$

$$\bullet w_3 f(w_2 h_1 + b_2) = 0 \rightarrow \boxed{b_2 < 0} \rightarrow f(w_1 + w_2 h_3 + b_2) = 0 \times$$

$$\bullet w_3 f(w_1 + w_2 h_3 + b_2) = 1 \rightarrow f(w_1 + w_2 h_3 + b_2) = 1$$

$$\rightarrow \boxed{w_1 + b_2 \geq 0} \rightarrow \boxed{w_3 = 1}$$

$$\bullet w_3 f(w_1 + w_2 h_4 + b_2) = 1 \rightarrow w_1 + w_2 + b_2 \geq 0$$

Then, if we assume that:

$$b_2 = -1, h_0 = 0 \rightarrow w_1 = 2, w_2 = 1$$

$$b_3 = 0, w_3 = 1$$