

مزایا و معایب معماری RNN به صورت خلاصه در جدول زیر آورده شده‌اند :

مزایا	معایب
<ul style="list-style-type: none"> امکان پردازش ورودی با هر طولی اندازه‌ی مدل مطابق با اندازه‌ی ورودی افزایش نمی‌یابد اطلاعات (زمان‌های) گذشته در محاسبه در نظر گرفته می‌شود وزن‌ها در طول زمان به اشتراک گذاشته می‌شوند 	<ul style="list-style-type: none"> محاسبه کند می‌شود دشواری بودن دسترسی به اطلاعات مدت‌ها پیش در نظر نگرفتن ورودی‌های بعدی در وضعیت جاری

راهنمای کوتاه شبکه‌های عصبی برگشتی

اثنین عیدی و شروین عیدی

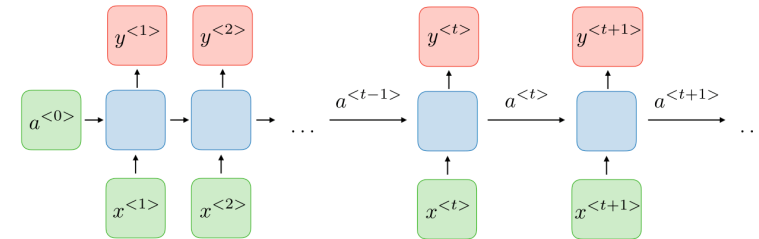
۱۵ شهریور ۱۳۹۸

□ کاربردهای RNN ها – مدل‌های RNN غالباً در حوزه‌ی پردازش زبان طبیعی و حوزه‌ی بازشناسایی گفتار به کار می‌روند. کاربردهای مختلف آنها به صورت خلاصه در جدول زیر آورده شده‌اند :

ترجمه به فارسی توسط الیستر. بازبینی توسط عرفان نوری.

نمای کلی

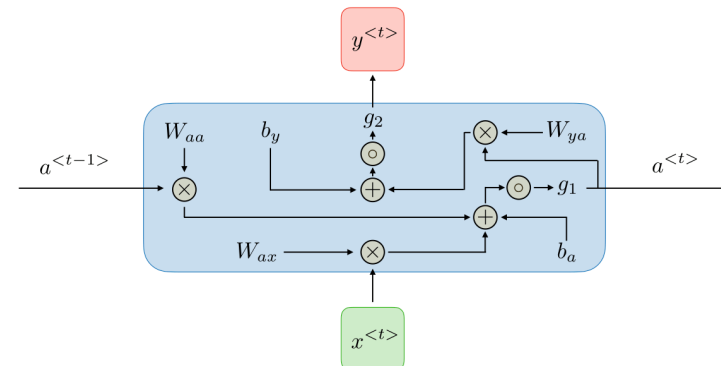
□ معماری RNN سنتی – شبکه‌های عصبی برگشتی که همچنین با عنوان RNN شناخته می‌شوند، دسته‌ای از شبکه‌های عصبی‌اند که این امکان را می‌دهند خروجی‌های قبلی به عنوان ورودی استفاده شوند و در عین حال حالت‌های نهان داشته باشند. این شبکه‌ها به طور معمول عبارت‌اند از :



به ازای هر گام زمانی t ، فعال سازی $a^{<t>}$ و خروجی $y^{<t>}$ به صورت زیر بیان می‌شود :

$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a) \quad \text{و} \quad y^{<t>} = g_2(W_{ya}a^{<t>} + b_y)$$

که در آن W_{ax} , W_{aa} , W_{ya} , b_a , b_y ضرایبی‌اند که در راستای زمان به اشتراک گذاشته می‌شوند و g_1 , g_2 توابع فعال سازی هستند.



نوع RNN	نگاره	مثال
یک به یک $T_x = T_y = 1$		شبکه‌ی عصبی سنتی
یک به چند $T_x = 1, T_y > 1$		تولید موسیقی
چند به یک $T_x > 1, T_y = 1$		دسته‌بندی حالت احساسی
چند به چند $T_x = T_y$		بازشناسایی موجودیت اسمی
چند به چند $T_x \neq T_y$		ترجمه ماشینی

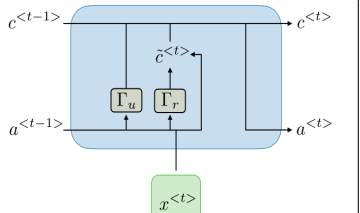
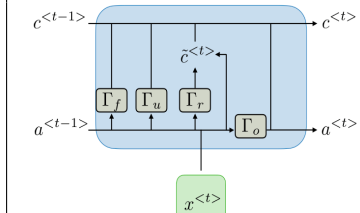
□ **انواع دروازه (types of gates)** – برای حل مشکل مشتق صفرشونده/متفجرشونده، در برخی از انواع RNN ها، دروازه‌های خاصی استفاده می‌شود و این دروازه‌ها عموماً هدف معینی دارند. این دروازه‌ها عموماً با نماد Γ نمایش داده می‌شوند و برابرند با:

$$\Gamma = \sigma(Wx^{<t>} + Ua^{<t-1>} + b)$$

که W, U, b ضرایب خاص دروازه و σ تابع سیگموئید است. دروازه‌های اصلی به صورت خلاصه در جدول زیر آورده شده‌اند:

نوع دروازه	نقش	به‌کار رفته در
Γ_u دروازه‌ی به‌روزرسانی	چه میزان از گذشته اکنون اهمیت دارد؟	GRU, LSTM
Γ_r دروازه‌ی ربط (میزان اهمیت)	اطلاعات گذشته رها شوند؟	GRU, LSTM
Γ_f دروازه‌ی فراموشی	سلول حذف شود یا خیر؟	LSTM
Γ_o دروازه‌ی خروجی	چه میزان از (محتوای) سلول آشکار شود؟	LSTM

□ **GRU/LSTM** – واحد برگشتی دروازه‌دار (Gated Recurrent Unit, GRU) و واحدهای حافظه‌ی کوتاه-مدت طولانی (Long Short-Term Memory units, LSTM) مشکل مشتق صفرشونده که در RNN های سنتی رخ می‌دهد، را بر طرف می‌کنند، درحالی‌که LSTM تعمیمی از GRU است. در جدول زیر، معادله‌های توصیف‌کننده هر معماری به صورت خلاصه آورده شده‌اند:

واحد برگشتی دروازه‌دار (GRU)	حافظه‌ی کوتاه-مدت طولانی (LSTM)	
$\tanh(W_c[\Gamma_r * a^{<t-1>}, x^{<t>}] + b_c)$	$\tanh(W_c[\Gamma_r * a^{<t-1>}, x^{<t>}] + b_c)$	$\tilde{c}^{<t>}$
$\Gamma_u * \tilde{c}^{<t>} + (1 - \Gamma_u) * c^{<t-1>}$	$\Gamma_u * \tilde{c}^{<t>} + \Gamma_f * c^{<t-1>}$	$c^{<t>}$
$c^{<t>}$	$\Gamma_o * c^{<t>}$	$a^{<t>}$
		وابستگی‌ها

نکته: * نشانه‌ی ضرب عنصر به‌عنصر دو بردار است.

□ **انواع RNN ها** – جدول زیر سایر معماری‌های پرکاربرد RNN را به صورت خلاصه نشان می‌دهد.

□ **تابع خطا (loss function)** – در شبکه عصبی برگشتی، تابع خطا \mathcal{L} برای همی گام‌های زمانی براساس خطا در هر گام به صورت زیر محاسبه می‌شود:

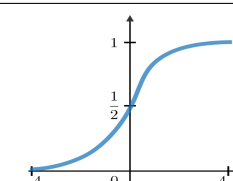
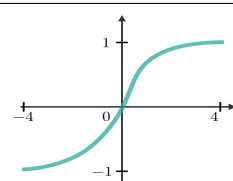
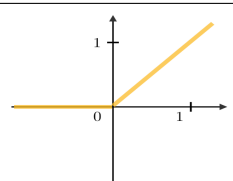
$$\mathcal{L}(\hat{y}, y) = \sum_{t=1}^{T_y} \mathcal{L}(\hat{y}^{<t>}, y^{<t>})$$

□ **انتشار معکوس در طول زمان (backpropagation through time)** – انتشار معکوس در هر نقطه از زمان انجام می‌شود. در گام زمانی T ، مشتق خطا \mathcal{L} با توجه به ماتریس وزن W به‌صورت زیر بیان می‌شود:

$$\frac{\partial \mathcal{L}^{(T)}}{\partial W} = \sum_{t=1}^T \frac{\partial \mathcal{L}^{(T)}}{\partial W} \Big|_{(t)}$$

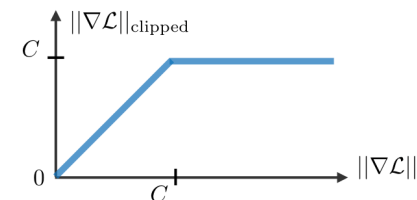
کنترل وابستگی‌های بلندمدت

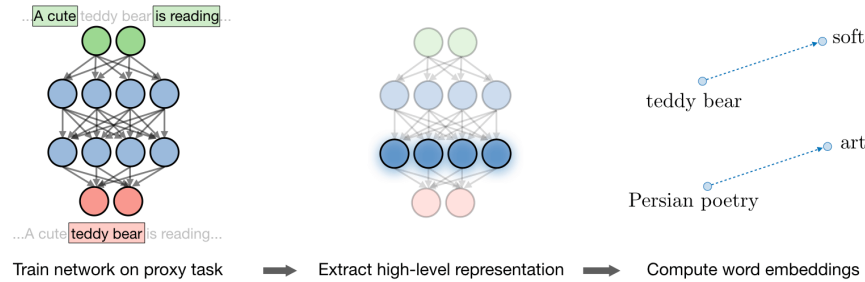
□ **توابع فعال‌سازی پرکاربرد** – رایج‌ترین توابع فعال‌سازی به‌کاررفته در ماژول‌های RNN به شرح زیر است:

سیگموئید (Sigmoid)	تانژانت هذلولوی (Tanh)	یکسو ساز (ReLU)
$g(z) = \frac{1}{1 + e^{-z}}$	$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	$g(z) = \max(0, z)$
		

□ **مشتق صفرشونده/متفجرشونده (vanishing/exploding gradient)** – پدیده مشتق صفرشونده و متفجرشونده غالباً در بستر RNN ها رخ می‌دهند. علت چنین رخدادی این است که به دلیل گرادیان ضربی، که می‌تواند با توجه به تعداد لایه‌ها به صورت نمایی کاهش/افزایش می‌یابد، به‌دست آوردن وابستگی‌های بلندمدت سخت است.

□ **برش گرادیان (gradient clipping)** – یک روش برای مقابله با انفجار گرادیان است که گاهی اوقات هنگام انتشار معکوس رخ می‌دهد. با تعیین حداکثر مقدار برای گرادیان، این پدیده در عمل کنترل می‌شود.





❑ **اسکیپگرام (skip-gram)** – مدل اسکپگرام word2vec یک وظیفه‌ی یادگیری بانظارت است که تعبیه‌های کلمه را با ارزیابی احتمال وقوع کلمه‌ی t هدف با کلمه‌ی زمینه c یاد می‌گیرد. با توجه به اینکه نماد θ_t پارامتری مرتبط با t است، احتمال $P(t|c)$ به صورت زیر به دست می‌آید:

$$P(t|c) = \frac{\exp(\theta_t^T e_c)}{\sum_{j=1}^{|V|} \exp(\theta_j^T e_c)}$$

نکته: جمع کل واژگان در بخش مقسوم‌الیه بیشینه‌ی هموار باعث می‌شود که این مدل از لحاظ محاسباتی گران شود. مدل **CBOW** مدل **word2vec** دیگری است که از کلمات اطراف برای پیش‌بینی یک کلمه مفروض استفاده می‌کند.

❑ **نمونه‌گیری منفی (negative sampling)** – مجموعه‌ای از دسته‌بندی‌های دودویی با استفاده از رگرسیون لجستیک است که مقصودش ارزیابی احتمال ظهور همزمان کلمه‌ی مفروض هدف و کلمه‌ی مفروض زمینه است، که در اینجا مدل‌ها براساس مجموعه k مثال منفی و 1 مثال مثبت آموزش می‌بینند. با توجه به کلمه‌ی مفروض زمینه c و کلمه‌ی مفروض هدف t ، پیش‌بینی به صورت زیر بیان می‌شود:

$$P(y = 1|c, t) = \sigma(\theta_t^T e_c)$$

نکته: این روش از لحاظ محاسباتی ارزان‌تر از مدل **skip-gram** است.

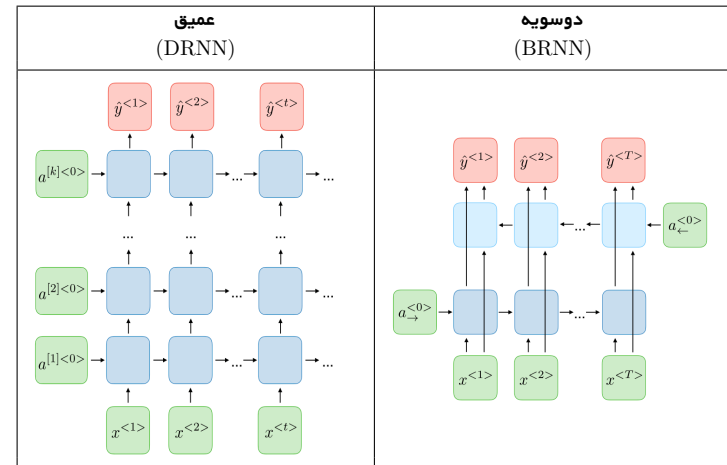
❑ **GloVe** – مدل GloVe، مخفف بردارهای سراسری بازنمایی کلمه، یکی از روش‌های تعبیه کلمه است که از ماتریس هم‌رویدادی X استفاده می‌کند که در آن هر $X_{i,j}$ به تعداد دفعاتی اشاره دارد که هدف i با زمینه j رخ می‌دهد. تابع هزینه‌ی J به صورت زیر است:

$$J(\theta) = \frac{1}{2} \sum_{i,j=1}^{|V|} f(X_{i,j})(\theta_i^T e_j + b_i + b'_j - \log(X_{i,j}))^2$$

که در آن f تابع وزن‌دهی است، به طوری که $f(X_{i,j}) = 0 \implies X_{i,j} = 0$. به توجیه به تقارنی که e و θ در این مدل دارند، نمایش تعبیه‌ی نهایی کلمه $e_w^{(final)}$ به صورت زیر محاسبه می‌شود:

$$e_w^{(final)} = \frac{e_w + \theta_w}{2}$$

تذکر: مولفه‌های مجزا در نمایش تعبیه‌ی یادگرفته‌شده‌ی کلمه الزاماً قابل تفسیر نیستند.



یادگیری بازنمایی کلمه

در این بخش، برای اشاره به واژگان از V و برای اشاره به اندازه‌ی آن از $|V|$ استفاده می‌کنیم.

❑ **روش‌های بازنمایی** – دو روش اصلی برای بازنمایی کلمات به صورت خلاصه در جدول زیر آورده شده‌اند:

بازنمایی تک‌فعال (1-hot representation)	تعبیه‌ی کلمه (word embedding)
– نشان داده شده با نماد o_w – رویکرد ساده، فاقد اطلاعات تشابه	– نشان داده شده با نماد e_w – به حساب آوردن تشابه کلمات

❑ **ماتریس تعبیه (embedding matrix)** – به ازای کلمه‌ی مفروض w ، ماتریس تعبیه E ماتریسی است که بازنمایی تک‌فعال o_w را به نمایش تعبیه‌ی e_w نگاشت می‌دهد:

$$e_w = E o_w$$

نکته: یادگیری ماتریس تعبیه را می‌توان با استفاده از مدل‌های درست‌نمایی هدف/متن (زمینه) انجام داد.

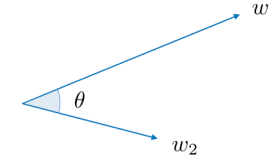
❑ **Word2vec** – Word2vec چهارچوبی است که با محاسبه‌ی احتمال قرار گرفتن یک کلمه‌ی خاص در میان سایر کلمات، تعبیه‌های کلمه را یاد می‌گیرد. مدل‌های متداول شامل skip-gram، نمونه‌برداری منفی (negative sampling) و CBOW هستند.

مقایسه‌ی کلمات

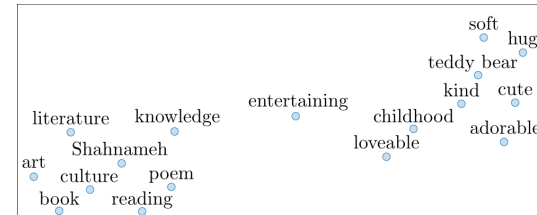
□ **شباهت کسینوسی (cosine similarity)** – شباهت کسینوسی بین کلمات w_1 و w_2 به صورت زیر بیان می‌شود:

$$\text{similarity} = \frac{w_1 \cdot w_2}{||w_1|| ||w_2||} = \cos(\theta)$$

نکته: θ زاویه بین کلمات w_1 و w_2 است.



□ **t -SNE – t -SNE** (Embedding) روشی است که هدف آن کاهش تعبیه‌های ابعاد بالا به فضایی با ابعاد پایین‌تر است. این روش در تصویرسازی بردارهای کلمه در فضای ۲ بعدی کاربرد فراوانی دارد.



مدل زبانی

□ **نمای کلی** – هدف مدل زبان تخمین احتمال جملی $P(y)$ است.

□ **مدل ان‌گرام (n -gram model)** – این مدل یک رویکرد ساده با هدف اندازه‌گیری احتمال نمایش یک عبارت در یک نوشته است که با دفعات تکرار آن در داده‌های آموزشی محاسبه می‌شود.

□ **سرگشتگی (perplexity)** – مدل‌های زبانی معمولاً با معیار سرگشتگی، که با PP هم نمایش داده می‌شود، سنجیده می‌شوند، که مقدار آن معکوس احتمال یک مجموعه داده است که تقسیم بر تعداد کلمات T می‌شود. هر چه سرگشتگی کمتر باشد بهتر است و به صورت زیر تعریف می‌شود:

$$PP = \prod_{t=1}^T \left(\frac{1}{\sum_{j=1}^{|V|} y_j^{(t)} \cdot \hat{y}_j^{(t)}} \right)^{\frac{1}{T}}$$

نکته: PP عموماً در t -SNE کاربرد دارد.

ترجمه ماشینی

□ **نمای کلی** – مدل ترجمه‌ی ماشینی مشابه مدل زبانی است با این تفاوت که یک شبکه‌ی رمزنگار قبل از آن قرار گرفته است. به همین دلیل، گاهی اوقات به آن مدل زبان شرطی می‌گویند. هدف آن یافتن جمله y است بطوری که:

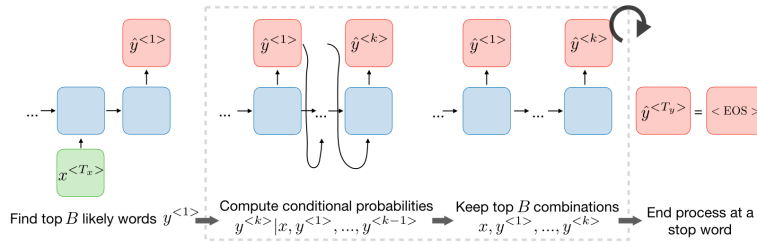
$$y = \arg \max_{y^{<1>}, \dots, y^{<T_y>}} P(y^{<1>}, \dots, y^{<T_y>} | x)$$

□ **(beam search)** – یک الگوریتم جستجوی اکتشافی است که در ترجمه‌ی ماشینی و بازتشفیص گفتار برای یافتن محتمل‌ترین جملی y باتوجه به ورودی مفروض x بکار برده می‌شود.

• **گام ۱:** یافتن B کلمه‌ی محتمل برتر $y^{<1>}$

• **گام ۲:** محاسبه احتمالات شرطی $y^{<k>} | x, y^{<1>}, \dots, y^{<k-1>}$

• **گام ۳:** نگاه‌داشتن B ترکیب برتر $x, y^{<1>}, \dots, y^{<k>}$ خاتمه فرایند با کلمه‌ی توقف



نکته: اگر پهنای پرتو ۱ باشد، آنگاه با جستجویی حریصانه ساده برابر خواهد بود.

□ **پهنای پرتو (beam width)** – پهنای پرتوی B پارامتری برای جستجوی پرتو است. مقادیر بزرگ B به نتیجه بهتر منتهی می‌شوند اما عملکرد آهسته‌تری دارند و حافظه را افزایش می‌دهند. مقادیر کوچک B به نتایج بدتر منتهی می‌شوند اما بار محاسباتی پایین‌تری دارند. مقدار استاندارد B حدود ۱۰ است.

□ **نرمال‌سازی طول (length normalization)** – برای بهبود ثبات عددی، جستجوی پرتو معمولاً با تابع هدف نرمال‌شده‌ی زیر اعمال می‌شود، که اغلب اوقات هدف درست‌نمایی لگاریتمی نرمال‌شده نامیده می‌شود و به صورت زیر تعریف می‌شود:

$$\text{Objective} = \frac{1}{T_y} \sum_{t=1}^{T_y} \log \left[p(y^{<t>} | x, y^{<1>}, \dots, y^{<t-1>}) \right]$$

تذکر: پارامتر α را می‌توان تعدیل‌کننده نامید و مقدارش معمولاً بین ۰/۵ و ۱ است.

□ **تحلیل خطا (error analysis)** – زمانی که ترجمه‌ی پیش‌بینی‌شده‌ی \hat{y} به دست می‌آید که مطلوب نیست، می‌توان با انجام تحلیل خطای زیر از خود پرسید که چرا ترجمه y^* خوب نیست:

قضیه	$P(y^* x) > P(\hat{y} x)$	$P(y^* x) \leq P(\hat{y} x)$
ریشه‌ی مشکل	جستجوی پرتوی معیوب	RNN معیوب
راه‌حل	افزایش پهنای پرتو	– امتحان معماری‌های مختلف – استفاده از تنظیم‌کننده – جمع‌آوری داده‌های بیشتر

□ **امتیاز Bleu** – جایگزین ارزشیابی دوزبانه (bleu) میزان خوب بودن ترجمه ماشینی را با محاسبه‌ی امتیاز تشابه بر مبنای دقت ان‌گرام اندازه‌گیری می‌کند. (این امتیاز) به صورت زیر تعریف می‌شود:

$$\text{bleu score} = \exp \left(\frac{1}{n} \sum_{k=1}^n p_k \right)$$

که p_n امتیاز bleu تنها براساس ان گرام است و به صورت زیر تعریف می‌شود :

$$p_n = \frac{\sum_{n\text{-gram} \in \hat{y}} \text{count}_{\text{clip}}(n\text{-gram})}{\sum_{n\text{-gram} \in \hat{y}} \text{count}(n\text{-gram})}$$

تذکر : ممکن است برای پیشگیری از امتیاز اغراق آمیز تصنعی *bleu* ، برای ترجمه‌های پیش‌بینی‌شده‌ی کوتاه از جریمه اختصار استفاده شود.

ژرفنگری

□ **مدل ژرفنگری** – این مدل به RNN این امکان را می‌دهد که به بخش‌های خاصی از ورودی که حائز اهمیت هستند توجه نشان دهد که در عمل باعث بهبود عملکرد مدل حاصل‌شده خواهد شد. اگر $\alpha^{<t,t'>}$ به معنای مقدار توجهی باشد که خروجی $y^{<t>}$ باید به فعال‌سازی $a^{<t'>}$ داشته باشد و $c^{<t>}$ نشان‌دهنده‌ی زمینه (متن) در زمان t باشد، داریم :

$$c^{<t>} = \sum_{t'} \alpha^{<t,t'>} a^{<t'>} \quad \sum_{t'} \alpha^{<t,t'>} = 1$$

نکته : امتیازات ژرفنگری عموماً در عنوان‌سازی متنی برای تصویر (*image captioning*) و ترجمه ماشینی کاربرد دارد.



A cute teddy bear is reading Persian literature



A cute teddy bear is reading Persian literature

□ **وزن ژرفنگری** – مقدار توجهی که خروجی $y^{<t>}$ باید به فعال‌سازی $a^{<t'>}$ داشته باشد به وسیله‌ی $\alpha^{<t,t'>}$ به دست می‌آید که به صورت زیر محاسبه می‌شود :

$$\alpha^{<t,t'>} = \frac{\exp(e^{<t,t'>})}{\sum_{t''=1}^{T_x} \exp(e^{<t,t''>})}$$

نکته : پیچیدگی محاسباتی به نسبت T_x از نوع درجه‌ی دوم است.

★ ★ ★