

| | | | |
|----|-------|-------|-------------------------------|
| ۱۲ | | | ۳/۳ تنظیم فراستخ |
| ۱۲ | | | ۳/۳/۱ مقداردهی اولیه‌ی وزن‌ها |
| ۱۲ | | | ۳/۳/۲ بهینه‌سازی همگرایی |
| ۱۲ | | | ۳/۴ نظام بخشی |
| ۱۲ | | | ۳/۵ عادت‌های خوب |

راهنمای کوتاه ویژه : یادگیری عمیق

افشین عمیدی و شروین عمیدی

۱۵ شهریور ۱۳۹۸

جدول محتوا

| | |
|----|---|
| ۱ | شبکه‌های عصبی پیچشی |
| ۲ | ۱/۱ نمای کلی |
| ۲ | ۱/۲ انواع لایه‌ها |
| ۲ | ۱/۳ ابرفراستخ‌های فیلتر |
| ۳ | ۱/۴ تنظیم ابرفراستخ‌ها |
| ۳ | ۱/۵ توابع فعال‌سازی پرکاربرد |
| ۴ | ۱/۶ شناسایی شبیه |
| ۵ | ۱/۶/۱ تایید چهره و بازشناسی |
| ۵ | ۱/۶/۲ انتقال سبک عصبی |
| ۶ | ۱/۶/۳ معماهایی که از ترفندات محاسباتی استفاده می‌کنند |
| ۷ | ۲ شبکه‌های عصبی برگشتی |
| ۷ | ۲/۱ نمای کلی |
| ۸ | ۲/۲ کنترل وابستگی‌های بلندمدت |
| ۹ | ۲/۳ یادگیری بازنامه‌ی کلمه |
| ۹ | ۲/۳/۱ انگیزه و نمادها |
| ۹ | ۲/۳/۲ تعبیه‌ی کلمه |
| ۱۰ | ۲/۴ مقایسه‌ی کلمات |
| ۱۰ | ۲/۵ مدل زبانی |
| ۱۰ | ۲/۶ ترجمه‌ی ماشینی |
| ۱۱ | ۲/۷ ژرفنگری |
| ۱۱ | ۳ نکات و ترفنداتی که در یادگیری عمیق |
| ۱۱ | ۳/۱ پردازش داده |
| ۱۲ | ۳/۲ آموزش یک شبکه‌ی عصبی |
| ۱۲ | ۳/۲/۱ تعاریف |
| ۱۲ | ۳/۲/۲ یافتن وزن‌های بهینه |

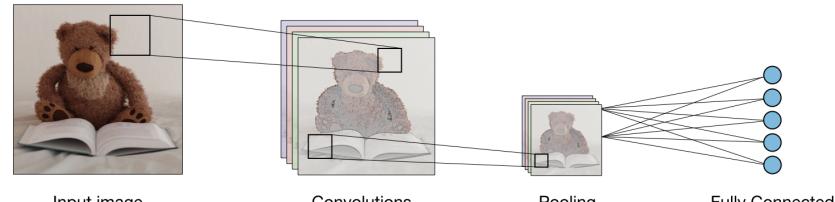
| ادغام میانگین | ادغام بیشینه | هدف |
|--|---|---------|
| هر عمل ادغام مقدار میانگین نمای فعلی را انتخاب می‌کند | هر عمل ادغام مقدار بیشینه نمای فعلی را انتخاب می‌کند | |
| | | نگاره |
| - کاستن نگاشت ویژگی - در (معماری) LeNet استفاده شده است | - ویژگی‌های شناسایی شده را حفظ می‌کند - اغلب مورد استفاده قرار می‌گیرد | توضیحات |

۱ شبکه‌های عصبی پیچشی

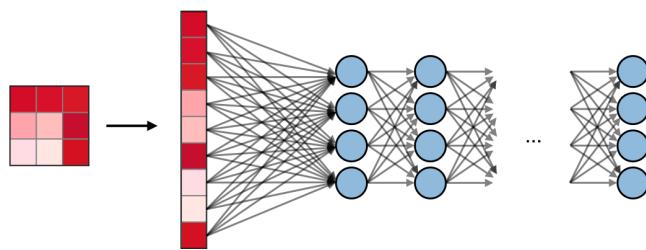
ترجمه به فارسی توسط الیستر و احسان کرمانی. بازبینی شده توسط عرفان نوری.

۱/۱ نمای کلی

معماری یک CNN سنتی – شبکه‌های عصبی مصنوعی پیچشی، که همچنین با عنوان CNN شناخته می‌شوند، یک نوع خاص از شبکه‌های عصبی هستند که عموماً از لایه‌های زیر تشکیل شده‌اند:



لایه‌ی کانولوشنی و لایه‌ی ادغام می‌توانند به نسبت ابرفراسنخ‌هایی که در بخش‌های بعدی بیان شده‌اند تنظیم و تعدیل شوند.

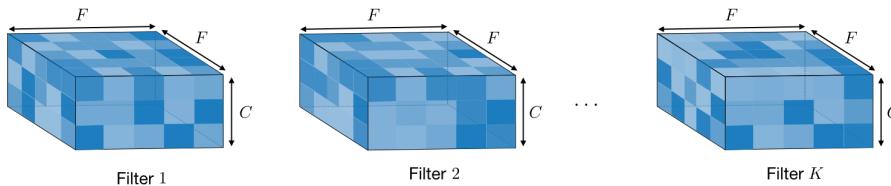


۱/۳ ابرفراسنخ‌های فیلتر

۱/۲ انواع لایه‌ها

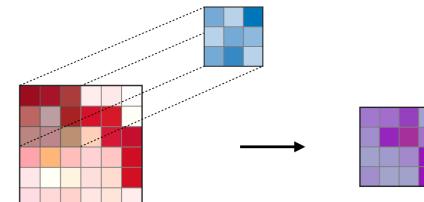
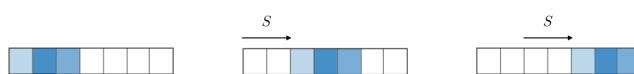
لایه‌ی کانولوشنی شامل فیلترهایی است که دانستن مفهوم نهفته در فراسنخ‌های آن اهمیت دارد.

لایه کانولوشنی (CONV) – یک فیلتر به اندازه $F \times F$ اعمال شده بر روی یک ورودی حاوی C کanal، یک توده $F \times F \times C$ پیش‌ورودی I به نسبت ابعادش، اجرا می‌کند. ابرفراسنخ‌های آن شامل اندازه فیلتر F و گام S هستند. خروجی حامل شده O نگاشت ویژگی یا نگاشت فعال‌سازی نامیده می‌شود.



نکته: اعمال K فیلتر به اندازه $F \times F$, منتج به یک نگاشت ویژگی خروجی به اندازه $O \times O$ می‌شود.

لایه گام (stride) – در یک عملیات ادغام یا پیچشی، اندازه گام S به تعداد پیکسل‌هایی که پنجره بعد از هر عملیات جابجا می‌شود، اشاره دارد.



نکته: مرحله کانولوشنی همچنین می‌تواند به موارد یک بعدی و سه بعدی تعمیم داده شود.

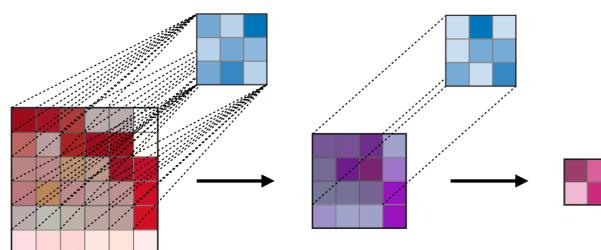
لایه ادغام (POOL) – لایه ادغام (POOL) یک عمل نمونه‌گاهی است، که معمولاً بعد از یک لایه کانولوشنی اعمال می‌شود، که تا حدی متبر به ناورداپی مکانی می‌شود. به طور خاص، ادغام بیشینه و میانگین انواع خاص ادغام هستند که به ترتیب مقدار بیشینه و میانگین گرفته می‌شود.

| FC | POOL | CONV | |
|--|--|--|----------------|
| | | | نگاره |
| N_{in} | $I \times I \times C$ | $I \times I \times C$ | اندازه ورودی |
| N_{out} | $O \times O \times C$ | $O \times O \times K$ | اندازه خروجی |
| $(N_{in} + 1) \times N_{out}$ | 0 | $(F \times F \times C + 1) \cdot K$ | تعداد فراسنخها |
| - ورودی سطح شده است - یک پیش‌قدرت به ازای هر نورون - تعداد نورون‌های FC فاقد محدودیت‌های ساختاری است | - عملیات ادغام به صورت کاتال بکاتال انجام می‌شود - در بیشتر موارد $S < F$ - در بیشتر موارد $S = F$ است | - یک پیش‌قدرت به ازای هر فیلتر - در بیشتر رایج برای $2C, K$ است | ملاحظات |

□ **ناحیه تاثیر (receptive field)** - ناحیه تاثیر در لایه k محدوده‌ای از ورودی $R_k \times R_k$ است که هر پیکسل k -می‌تواند 'بیند'. با ذکر F_j به عنوان اندازه فیلتر لایه j و S_i مقدار کام لایه i و با این توافق که $S_0 = 1$ است، ناحیه تاثیر در لایه k فرمول زیر محاسبه می‌شود:

$$R_k = 1 + \sum_{j=1}^k (F_j - 1) \prod_{i=0}^{j-1} S_i$$

در مثال زیر داریم، $S_1 = S_2 = 1$ و $F_1 = F_2 = 3$ که متناسب با $R_2 = 1 + 2 \cdot 1 + 2 \cdot 1 = 5$ است.



۱/۵ توابع فعال‌سازی پرکاربرد

□ **تابع یکسوساز خطی (Rectified Linear Unit)** - تابع یکسوساز خطی (ReLU) یک تابع فعال‌سازی g است که بر روی تمامی عنامر توده اعمال می‌شود. هدف آن ارائه (رفتار) غیرخطی به شبکه است. انواع آن در جدول زیر به صورت خلاصه آمدند:

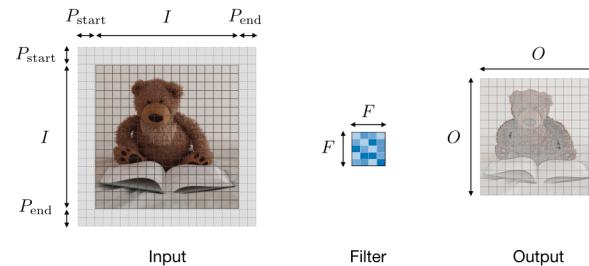
□ **حاشیه‌ی صفر (zero-padding)** - حاشیه‌ی صفر به فرآیند افزودن P صفر به هر طرف از کرانه‌های ورودی اشاره دارد. این مقدار می‌تواند به طور خودکار به سه روش زیر تعیین گردد:

| Full | Same | Valid | |
|---|--|---------|-------|
| $P_{start} \in [0, F - 1]$ $P_{end} = F - 1$ | $P_{start} = \left\lceil \frac{S \lceil \frac{I}{S} \rceil - I + F - S}{2} \right\rceil$ $P_{end} = \left\lceil \frac{S \lceil \frac{I}{S} \rceil - I + F - S}{2} \right\rceil$ | $P = 0$ | مقدار |
| | | | نگاره |
| - (اعمال) حاشیه به طوری که اندازه نگاشت ویژگی $\left\lceil \frac{I}{S} \right\rceil$ باشد - (محاسبه) اندازه خروجی به لحاظ ریاضیاتی آسان است - همچنین حاشیه 'نیمه' نامیده می‌شود | - قادر حاشیه - اگر ابعاد مطابقت ندارد آخرین کانولوشنی را رها کن | | هدف |

۱/۶ تنظیم ابرفراستخ‌ها

□ **سازش پذیری فراستخ در لایه کانولوشنی** - با ذکر I به عنوان طول اندازه توده ورودی، F طول فیلتر، P میزان حاشیه‌ی صفر، S گام، اندازه خروجی نگاشت ویژگی در امتداد ابعاد خواهد بود:

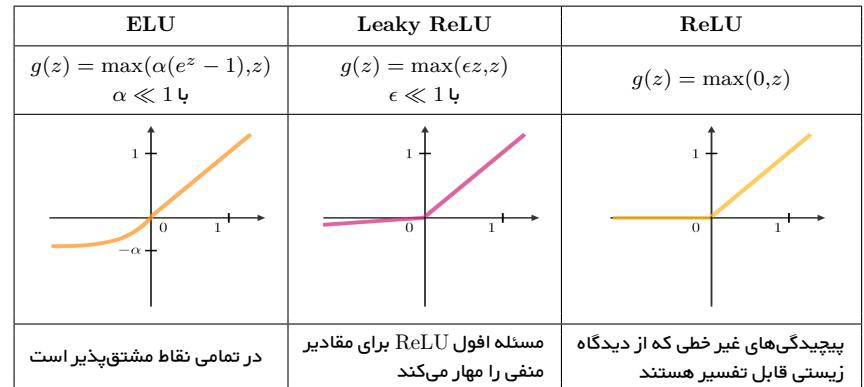
$$O = \frac{I - F + P_{start} + P_{end}}{S} + 1$$



نکته: اغلب $P_{start} = P_{end} \triangleq P$ است، در این صورت $P_{start} + P_{end} \triangleq 2P$ در فرمول بالا جایگزین کرد.

□ **درک پیچیدگی مدل** - برای برآورد پیچیدگی مدل، اغلب تعیین تعداد فراستخ‌های که معماری آن می‌تواند داشته باشد، مقید است. در یک لایه مفروض شبکه پیچشی عمیقی این امر به صورت زیر انجام می‌شود:

| شناسایی نقاط (برجسته) | پیش‌بینی کادر مخصوص‌کننده |
|---|---|
| - یک شکل یا مشخصات یک شیء (مثل چشمها) را شناسایی می‌کند - موشکافانه‌تر | بخشی از تصویر که شیء در آن قرار گرفته را شناسایی می‌کند |
| | |
| نقاط مرتع (l1x, l1y), ..., (lnx, lny) | مرکز کادر (b_x, b_y)، ارتفاع b_h و عرض b_w |

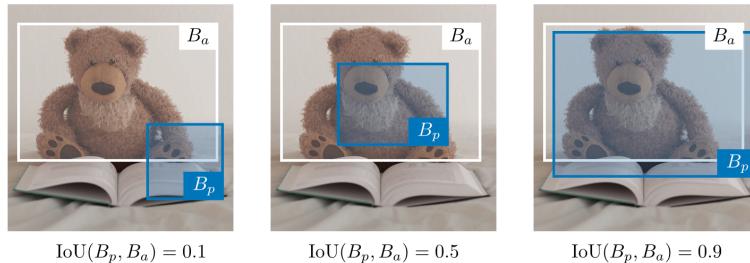


□ **بیشینه هموار (softmax)** – مرحله بیشینه هموار را می‌توان به عنوان یک تابع لجستیکی تعیین داده شده که یک بردار $x \in \mathbb{R}^n$ را از ورودی می‌گیرد و یک بردار خروجی احتمال $p \in \mathbb{R}^n$ ، بواسطه تابع بیشینه هموار در انتهایی معماری، تولید می‌کند. این تابع به صورت زیر تعریف می‌شود:

$$p = \begin{pmatrix} p_1 \\ \vdots \\ p_n \end{pmatrix} \quad \text{با} \quad p_i = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}}$$

۱/۶ شناسایی شیء

□ **انواع مدل** – سه نوع اصلی از الگوریتم‌های بازشناسایی وجود دارد، که ماهیت آنچه که شناسایی شده متفاوت است. این الگوریتم‌ها در جدول زیر توضیح داده شده‌اند:



نکته: همواره داریم $IoU \in [0, 1]$. به صورت قرارداد، یک کادر مخصوص‌کننده B_p را می‌توان نسبتاً خوب در نظر گرفت اگر $IoU(B_p, B_a) \geq 0.5$.

□ **کادرهای مخصوص (anchor boxes)** – کادر بندی مخصوص روشنی است که برای پیش‌بینی کادرهای مخصوص‌کننده هموشان استفاده می‌شود. در عمل، شبکه این اجزا را دارد که بیش از یک کادر به صورت همزمان پیش‌بینی کند جایی که هر پیش‌بینی کادر محدود به داشتن یک مجموعه خصوصیات هندسی مفروض است. به عنوان مثال، اولین پیش‌بینی می‌تواند یک کادر مستطیلی با قالب خاص باشد حال آنکه کادر دوم، یک کادر مستطیلی مخصوص با قالب هندسی متفاوتی خواهد بود.

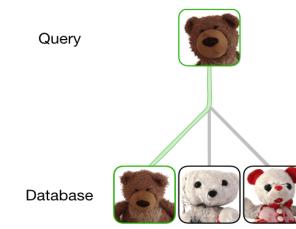
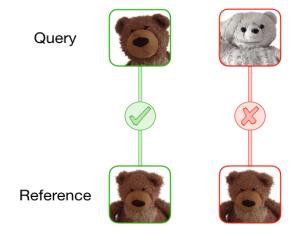
□ **فروداشت غیربیشینه (non-max suppression)** – هدف روش فروداشت غیربیشینه، حذف کادرهای مخصوص‌کننده هموشان تکراری دسته کیسان با انتخاب معرفت‌ترین‌ها است. بعد از حذف همه کادرهایی که احتمال پیش‌بینی پایین‌تر از ۰/۶ دارند، مراحل زیر با وجود آنکه کادرهایی باقی می‌مانند، تکرار می‌شوند:

- **کام اول**: کادر با بالاترین احتمال پیش‌بینی را انتخاب کن
- **کام دوم**: هر کادری که $IoU \geq 0.5$ نسبت به کادر پیشین دارد را کن

| شناسایی | دسته‌بندی با موقعیت‌یابی | دسته‌بندی تصویر |
|---|---|---|
| | | |
| – یک شیء را در یک عکس شناسایی می‌کند – احتمال اشیاء و موقعیت آنها را پیش‌بینی می‌کند | – یک شیء را در یک عکس شناسایی می‌کند – احتمال اشیاء و موقعیت آنها را پیش‌بینی می‌کند | یک عکس را دسته‌بندی می‌کند احتمال شیء را پیش‌بینی می‌کند |
| R-CNN, YOLO | R-CNN | CNN |

۱/۶/۱ تایید چهره و بازشناسایی

□ نوع مدل - دو نوع اصلی از مدل در جدول زیر به صورت خلاصه آورده شده‌اند :

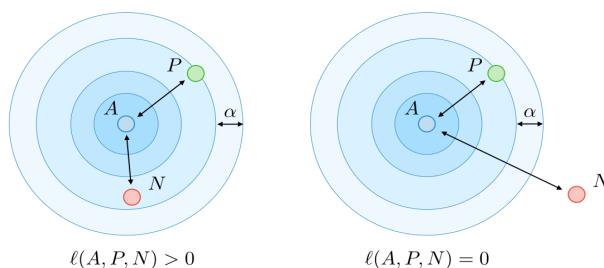
| بازشناسایی چهره | تایید چهره |
|---|---|
| - این فرد یکی از K فرد پایگاه داده است? - جستجوی یک‌به‌یک | - فرد مورد نظر است? - جستجوی یک‌به‌یک |
|  |  |

□ یادگیری یکباره‌ای (One Shot Learning) - یادگیری یکباره‌ای یک الگوریتم تایید چهره است که از یک مجموعه آموزشی محدود برای یادگیری یکتابع مشابهت که میزان اختلاف دو تصویر مفروض را تعیین می‌کند، بهره می‌برد. تابع مشابهت اعمال شده بر روی دو تصویر اغلب با نماد $d(\text{image 1}, \text{image 2})$ نمایش داده می‌شود.

□ شبکه‌ی Siamese - هدف شبکه‌ی Siamese یادگیری طریقه رمزگاری تصاویر و سپس تعیین اختلاف دو تصویر است. برای یک تصویر مفروض ورودی $x^{(i)}$, خروجی رمزگاری شده اغلب با نماد $f(x^{(i)})$ نمایش داده می‌شود.

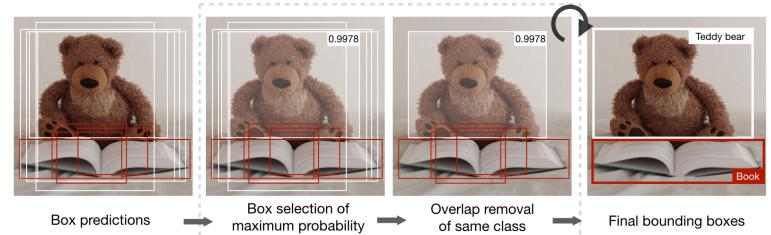
□ خطای سه‌گانه (triplet loss) - خطای سه‌گانه ℓ یکتابع خط است که بر روی بازنمایی تعیینه سه‌گانه تصاویر A محور تعاق دارند، حال آنکه نمونه منفی به دسته دیگری تعلق دارد. با تأمین $\alpha \in \mathbb{R}^+$ (به عنوان) فراسنج حاشیه، این خطابه صورت زیر تعریف می‌شود :

$$\ell(A, P, N) = \max(d(A, P) - d(A, N) + \alpha, 0)$$



۱/۶/۲ انتقال سبک عصبی

□ انگیزه - هدف انتقال سبک عصبی تولید یک تصویر G بر مبنای یک محتوا مفروض C و سبک مفروض S است.



□ YOLO - «شما فقط یکبار نگاه می‌کنید» (You Only Look Once, YOLO) یک الگوریتم شناسایی شیء است که مراحل زیر را اجرا می‌کند :

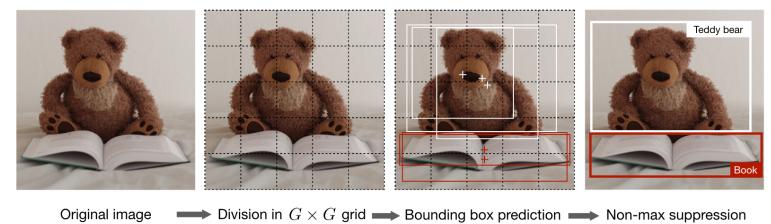
- گام اول : تصویر ورودی را به یک شبکه $G \times G$ تقسیم کن

- گام دوم : برای هر سلول شبکه، یک CNN که y را به شکل زیر پیش‌بینی می‌کند، اجرا کن :

$$y = \left[\underbrace{p_c, b_x, b_y, b_h, b_w, c_1, c_2, \dots, c_p, \dots}_{\text{repeated } k \text{ times}} \right]^T \in \mathbb{R}^{G \times G \times k \times (5+p)}$$

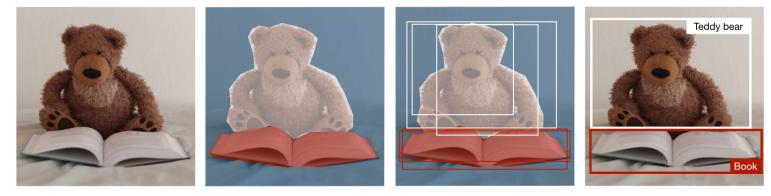
که p_c احتمال شناسایی یک شیء است، b_x, b_y, b_h, b_w اندازه‌های نسبی کادر محیطی شناسایی شده است، c_1, \dots, c_p نمایش تک‌فعال یک دسته از p دسته که تشخیص داده شده است، و k تعداد کادرهای محوری است.

- گام سوم : الگوریتم فروداشت غیربیشینه را برای حذف هر کادر محصور کننده همپوشان تکراری بالقوه، اجرا کن.

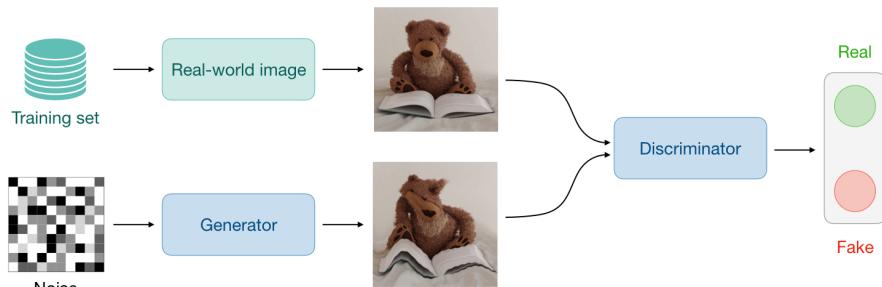


نکته : زمانی که $p_c = 0$ است، شبکه هیچ شبیه را شناسایی نمی‌کند. در چنین حالتی، پیش‌بینی‌های متناصر b_x, \dots, b_w بایستی تادیله گرفته شوند.

□ R-CNN - ناحیه با شبکه‌های عصبی پیچشی (Region with Convolutional Neural Networks, R-CNN) یک الگوریتم شناسایی شیء است که ابتدا تصویر را برای یافتن کادرهای محصور کننده مربوط بالقوه قطعه‌بندی می‌کند و سپس الگوریتم شناسایی را برای یافتن محتمل‌ترین اشیاء در این کادرهای محصور کننده اجرا می‌کند.



نکته : هرچند الگوریتم اصلی به لحظه محاسباتی پرهزینه و کند است، معماری‌های جدید از قبیل Faster R-CNN و Fast R-CNN باعث شدند که الگوریتم سریعتر اجرا شود.



□ **فعال‌سازی (activation)** – در یک لایه مفروض l ، فعال‌سازی با $a^{[l]}$ نمایش داده می‌شود و به ابعاد $n_H \times n_w \times n_c$ است

□ **تابع هزینه‌ی محتوا (content cost function)** – تابع هزینه‌ی محتوا $J_{\text{content}}(C, G)$ برای تعیین میزان اختلاف تصویر تولیدشده G از تصویر اصلی C استفاده می‌شود. این تابع به صورت زیر تعریف می‌شود :

$$J_{\text{content}}(C, G) = \frac{1}{2} \|a^{[l]}(C) - a^{[l]}(G)\|^2$$

نکته : موارد استفاده متعدد GAN ها شامل تبدیل متن به تصویر، تولید موسیقی و تلفیقی از آنهاست.

□ **ResNet** – معماری شبکه‌ی پسماند (همچنین با عنوان ResNet شناخته می‌شود) از بلاک‌های پسماند با تعداد لایه‌های زیاد به منظور کاهش خطای آموزش استفاده می‌کند. بلکه پسماند معادله‌ای با خصوصیات زیر دارد :

$$a^{[l+2]} = g(a^{[l]} + z^{[l+2]})$$

□ **شبکه‌ی Inception** – این معماری از مازول‌های inception مختلف برای افزایش کارایی از طریق تنوع بخشی ویژگی‌ها است. به طور خامن، این معماری از تردد کانولوشنی 1×1 برای محدود سازی بار محاسباتی استفاده می‌کند.

□ **ماتریس سبک (style matrix)** – ماتریس سبک $G^{[l]}$ یک لایه مفروض l ، یک ماتریس گرم (Gram) است که هر کدام از عنامر $G_{kk'}^{[l]}$ میزان همبستگی کانال‌های k و k' را می‌سنجد. این ماتریس نسبت به فعال‌سازی‌های $a^{[l]}$ به صورت زیر محاسبه می‌شود :

$$G_{kk'}^{[l]} = \sum_{i=1}^{n_H^{[l]}} \sum_{j=1}^{n_w^{[l]}} a_{ijk}^{[l]} a_{ijk'}^{[l]}$$

نکته : ماتریس سبک برای تصویر تولید شده، به ترتیب با $G^{[l](S)}$ و $G^{[l](G)}$ نمایش داده می‌شوند.

□ **تابع هزینه‌ی سبک (style cost function)** – تابع هزینه‌ی سبک $J_{\text{style}}(S, G)$ برای تعیین میزان اختلاف تصویر تولیدشده G و سبک S استفاده می‌شود. این تابع زیر تعریف می‌شود :

$$J_{\text{style}}^{[l]}(S, G) = \frac{1}{(2n_H n_w n_c)^2} \|G^{[l](S)} - G^{[l](G)}\|_F^2 = \frac{1}{(2n_H n_w n_c)^2} \sum_{k, k'=1}^{n_c} \left(G_{kk'}^{[l](S)} - G_{kk'}^{[l](G)} \right)^2$$

□ **تابع هزینه‌ی کل** – تابع هزینه‌ی کل به صورت ترکیبی از تابع هزینه‌ی سبک و محتوا تعیین شده است که با فراسنج‌های α, β ، به شکل زیر وزن دار شده است :

$$J(G) = \alpha J_{\text{content}}(C, G) + \beta J_{\text{style}}(S, G)$$

نکته : مقدار بیشتر α مدل را به توجه بیشتر به محتوا و می‌دارد حال آنکه مقدار بیشتر β مدل را به توجه بیشتر به سبک و می‌دارد.

۱/۶/۳ معماری‌هایی که از ترفندهای محاسباتی استفاده می‌کنند

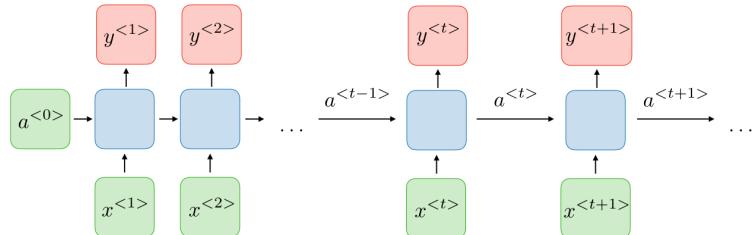
□ **شبکه‌ی همآورد مولد (Generative Adversarial Network)** – شبکه‌ی همآورد مولد، همچنین با نام GAN شناخته می‌شود، ترکیبی از یک مدل مولد و تمیزدهنده هستند، جایی‌که مدل مولد هدفش تولید واقعی ترین خروجی است که به (مدل) تمیزدهنده تغذیه می‌شود و این (مدل) هدفش تفکیک بین تصویر تولیدشده و واقعی است.

۲ شبکه‌های عصبی برگشتی

ترجمه به فارسی توسط الیستر. بازبینی توسط عرفان نوری.

۱/۱ نمای کلی

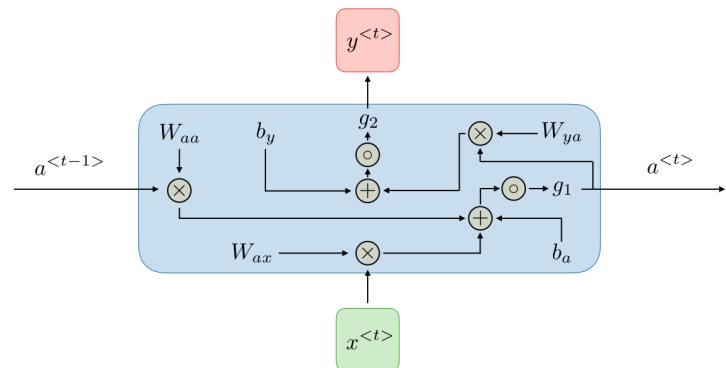
نمای کلی RNN سنتی - شبکه‌های عصبی برگشتی که همچنین با عنوان RNN شناخته می‌شوند، دسته‌ای از شبکه‌های عصبی‌اند که این امکان را می‌دهند خروجی‌های قبلی به عنوان ورودی استفاده شوند و در عین حال حالت‌های نهان داشته باشند. این شبکه‌ها به طور معمول عبارت‌اند از:



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

$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a) \quad 9 \quad y^{<t>} = g_2(W_{ya}a^{<t>} + b_y)$$

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 فعال‌سازی هستند.



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

| معایب | مزایا |
|--|--|
| - محاسبه کند می‌شود | - امکان پردازش ورودی با هر طولی |
| - دشوار بودن دسترسی به اطلاعات مدت‌ها پیش | - اندازه‌ی مدل مطابق با اندازه‌ی ورودی افزایش نمی‌یابد |
| - در نظر نگرفتن ورودی‌های بعدی در وضعیت جاری | - اطلاعات (زمان‌های) گذشته در محاسبه در نظر گرفته شود |
| | - وزن‌ها در طول زمان به اشتراک گذاشته می‌شوند |

نمای کلی RNN ها - مدل‌های RNN غالباً در حوزه‌ی پردازش زبان طبیعی و حوزه‌ی بازناسایی گفتار به کار می‌روند. کاربردهای مختلف آنها به صورت خلاصه در جدول زیر آورده شده‌اند:

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

تابع خطا (loss function) - در شبکه‌ی عصبی برگشتی، تابع خطا \mathcal{L} برای همه‌ی 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

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

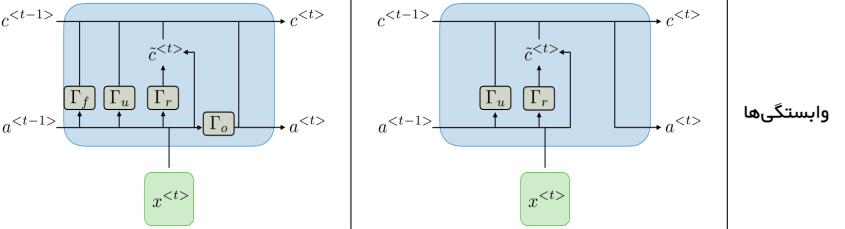
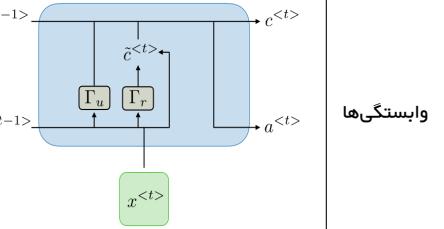
می‌شود. در گام زمانی T , مشتق خطای \mathcal{L} با توجه به ماتریس وزن W به صورت زیر بیان می‌شود :

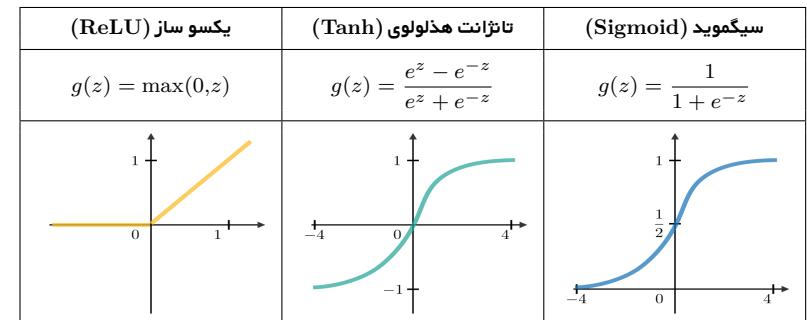
$$\frac{\partial \mathcal{L}^{(T)}}{\partial W} = \sum_{t=1}^T \frac{\partial \mathcal{L}^{(T)}}{\partial W} \Big|_{(t)}$$

۲/۲ کنترل وابستگی‌های بلندمدت

□ **توابع فعال‌سازی پرکاربرد** – رایج‌ترین توابع فعال‌سازی بهکار رفته در مآژول‌های RNN به شرح زیر است :

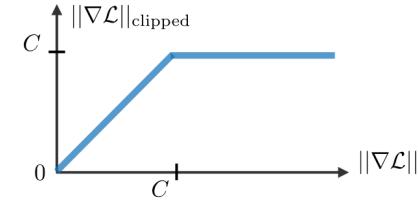
– واحد برگشتی دروازه‌دار (GRU/LSTM) □ **حافظه‌ی کوتاه‌مدت طولانی (LSTM)** و **واحد برگشتی دروازه‌دار (GRU)** (Long Short-Term Memory units, LSTM) مشکل مشتق صفرشونده که در RNN‌های سنتی رخ می‌دهد، را بر طرف می‌کنند، در حالی‌که LSTM تعمیمی از GRU است. در جدول زیر، معادله‌های توصیف‌کننده هر معماری به صورت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

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



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

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



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

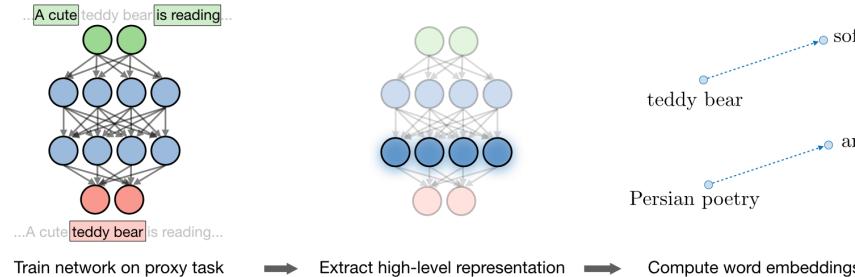
نکته : نشانه‌ی * نمایان‌گر ضرب عنصری به عنصر دو بردار است.

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

□ **انواع RNN‌ها** – جدول زیر سایر معماری‌های پرکاربرد RNN را به صورت خلاصه نشان می‌دهد. که W, U, b ضرایب خاص دروازه و σ تابع سیگموید است. دروازه‌های اصلی به صورت خلاصه در جدول زیرآورده شده‌اند :

۲/۳/۳ تعبیه کلمه

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



اسکیپ‌گرام (skip-gram) – مدل اسکیپ‌گرام word2vec یک وظیفه‌ای یادگیری بانظارت است که تعبیه‌های کلمه را ارزیاب احتمال وقوع کلمه t هدف با کلمه زمینه c یاد می‌گیرد. با توجه به اینکه نماد θ_t پارامتری مرتبط با t است، احتمال $P(t|c)$ به صورت زیر بدست 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 بیشینه‌های هموار باعث می‌شود که این مدل از لحاظ محاسباتی گران شود. مدل word2vec CBOW دیگری است که از کلمات اطراف برای پیش‌بینی یک کلمه مفروض استفاده می‌کند.

نمونه‌گیری منفی (negative sampling) – مجموعه‌ای از دسته‌بندی‌های دو دوی با استفاده از رگرسیون لجستیک است که مجموعه ارزیاب احتمال ظهور هفتمان کلمه مفروض هدف و کلمه مفروض زمینه است، که در اینجا مدل‌ها براساس مجموعه k مثال منفی و ۱ مثال مثبت آموزش می‌بینند. با توجه به کلمه مفروض زمینه c و کلمه 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.

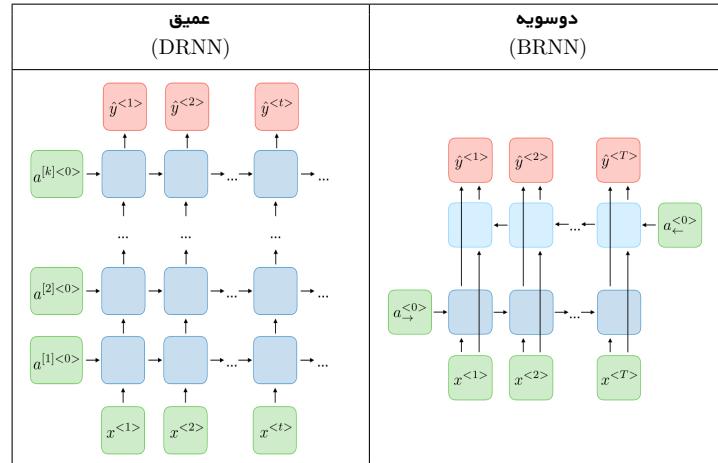
GloVe – مدل GloVe □، مخفف بردارهای سراسری بازنمایی کلمه، یکی از روش‌های تعبیه کلمه است که از ماتریس هم‌رویدادی X استفاده می‌کند که در آن هر $X_{i,j}$ به تعداد دقیعی اشاره دارد که هدف i با زمینه j رخ می‌دهد.تابع هزینه J به صورت زیر است:

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

که در آن f تابع وزن‌دهی است، به طوری که $f(X_{i,j}) = 0 \Rightarrow f(X_{i,j}) = 0$. با توجه به تقارنی که e و θ در این مدل دارند، نمایش تعبیه‌ی نهایی کلمه $e_w^{(\text{final})}$ ۵ صورت زیر محاسبه می‌شود:

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

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



۲/۳ پادگیری بازنمایی کلمه

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

۱/۳/۱ انگیزه و نمادها

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

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

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

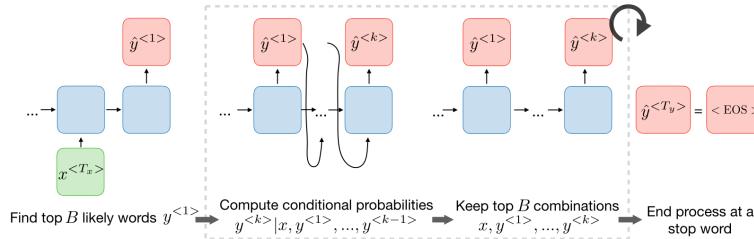
$$e_w = E o_w$$

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

۲/۴ مقایسه کلمات

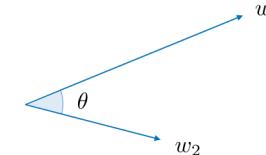
□ **شماحت کسینووسی (cosine similarity)** – یک الگوریتم جستجوی اکتشافی است که در ترجمه‌ی ماشینی و بازنشخیص گفتار برای یافتن محتمل‌ترین جمله‌ی y با توجه به ورودی مفروض x بکار برده می‌شود.

- گام ۱ : یافتن B کلمه‌ی محتمل برتر $y^{<1>}|x, y^{<1>}|, \dots, y^{<k-1>}|x, y^{<1>}|, \dots, y^{<k-1>}|$
- گام ۲ : محاسبه احتمالات شرطی $x, y^{<1>}|, \dots, y^{<k>}|$ ، خاتمه فرآیند با کلمه‌ی توقف
- گام ۳ : نگهداشتن B ترکیب برتر $x, y^{<1>}|, \dots, y^{<k>}|$ ، ...

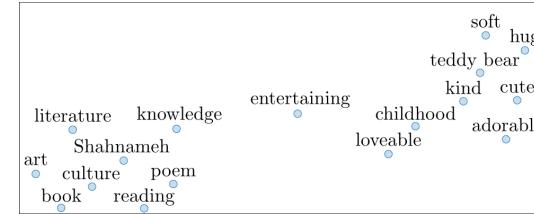


نکته: θ زاویه بین کلمات w_1 و w_2 است.

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



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



۲/۵ مدل زبانی

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

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

□ **سرگشتنگی (perplexity)** – مدل‌های زبانی معمولاً با معیار سرگشتنی، 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 سرگشتنگی 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 $t\text{-SNE}$ کاربرد دارد.

۲/۶ ترجمه ماشینی

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

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

□ **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 پافتن جمله y است بطوری که:

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

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

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

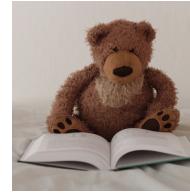
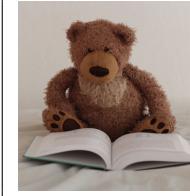
۳ نکات و ترفندهای یادگیری عمیق

ترجمه به فارسی توسط البستر. بازبینی توسط عرفان نوری.

۳/۱ پردازش داده

داده‌افزایی (data augmentation) – مدل‌های یادگیری عمیق معمولاً به داده‌های زیادی نیاز دارند تا بتوانند به خوبی آموزش بینند. اغلب، استفاده از روش‌های داده‌افزایی برای گرفتن داده‌ی بیشتر از داده‌های موجود، مفید است. اصلی‌ترین آنها در جدول زیر به اختصار آمده‌اند. به عبارت دقیق‌تر، با در نظر گرفتن تصویر ورودی زیر، روش‌هایی که می‌توان اعمال کرد بین شرح هستند:

تذکر: ممکن است برای پیشگیری از امتیاز اغراق آمیز تصنیع bleu، برای ترجمه‌های پیش‌بینی‌شده‌ی کوتاه از جرمیه اختصار استفاده شود.

| برش تعادلی | چرخش | قرینه | تصویر اصلی |
|--|---|---|---|
|  |  |  |  |
| <ul style="list-style-type: none"> - روی ناحیه‌ای تصادفی از تصویر متمرکز می‌شود - چندین برش تصادفی را می‌توان پشت سرهم انجام داد | <ul style="list-style-type: none"> - چرخش با زاویه‌ی انداخت - خط افق نادرست را شبیه‌سازی می‌کند | <ul style="list-style-type: none"> - قرینه‌شده نسبت به مجموعی که معنای (محتوا) تصویر را حفظ می‌کند | <ul style="list-style-type: none"> - تصویر (آغازین) بدون هیچ‌گونه تغییری |

| تغییر تابیه | هدرفت اطلاعات | اضافه‌کردن نویز | تغییر رنگ |
|---|---|---|--|
|  |  |  |  |
| <ul style="list-style-type: none"> - تغییر درخشندگی - با توجه به زمان روز تفاوت نمایش (تصویر) را کنترل می‌کند | <ul style="list-style-type: none"> - بخش‌هایی از تصویر نادیده گرته می‌شوند - تقاید (شبیه سازی) هدرفت بالقوه بخش هایی از تصویر | <ul style="list-style-type: none"> - افزودگی نویز - مقاومت بیشتر نسبت به تغییر کیفیت تصاویر ورودی | <ul style="list-style-type: none"> - عنامر RGB کمی تغییر کرده است - نویزی که در هنگام مواجه شدن با نور رخ می‌دهد را شبیه‌سازی می‌کند |

نرمال‌سازی دسته‌ای (batch normalization) – یک مرحله از فرآینج‌های β, γ که دسته‌ی $\{x_i\}$ را نرمال می‌کند. نماد μ_B, σ_B^2 به میانگین و وردابی دسته‌ای که می‌خواهیم آن را اصلاح کنیم اشاره دارد که به صورت زیر است:

$$x_i \leftarrow \gamma \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} + \beta$$

معمولاً بعد از یک لایه‌ی تمام‌متصل یا لایه‌ی کانولوشنی و قبل از یک لایه‌ی غیرخطی اعمال می‌شود و امکان استفاده از نرخ یادگیری بالاتر را می‌دهد و همچنین باعث می‌شود که وابستگی شدید مدل به مقداردهی اولیه کاهش یابد.

۲/۷ 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 حاصل شده خواهد شد. اگر $\alpha^{<t,t'} >$ به معنای مقدار توجهی باشد که خروجی $y^{<t>}$ 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 $y^{<t>}$ باشد به فعال‌سازی $a^{<t,t'} >$ داشته باشد به‌وسیله‌ی $\alpha^{<t,t'} >$ بدست 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.

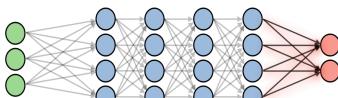
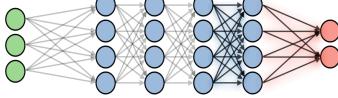
* * *

۱۳/۳ تنظیم فراسنج

۱/۳ مقداردهی اولیه وزن‌ها

□ مقداردهی اولیه Xavier – به جای مقداردهی اولیه وزن‌ها به شیوه‌ی کاملاً تصادفی، مقداردهی اولیه Xavier این امکان را فراهم می‌سازد تا وزن‌های اولیه‌ای داشته باشیم که ویژگی‌های منحصر به فرد معماری را به حساب می‌آورند.

□ پادگیزی انتقالی (transfer learning) – آموزش یک مدل پادگیزی عمیق به داده‌های زیاد و مهتر از آن به زمان زیادی احتیاج دارد. اغلب بهتر است که از وزن‌های پیش‌آموخته روی پایگاه داده‌های عظیم که آموزش بر روی آن‌ها روزها یا هفته‌ها طول می‌گشند استفاده کرد، و آن‌ها را برای موارد استفاده‌ی خود به کار برد. بسته به میزان داده‌هایی که در اختیار داریم، در زیر روش‌های مختلفی که می‌توان از آنها بهره جست آورده شده‌اند:

| توضیح | نگاره | تعداد داده‌های آموزش |
|--|--|----------------------|
| منجمد کردن تمامی لایه‌ها، آموزش وزن‌ها در بیشینه هموار |  | کوچک |
| منجمد کردن اکثر لایه‌ها، آموزش وزن‌ها در لایه‌های آخر و بیشینه هموار |  | متوسط |
| آموزش وزن‌ها در (تمامی) لایه‌ها و بیشینه هموار با مقداردهی اولیه وزن‌ها بر طبق مقادیر پیش‌آموخته |  | بزرگ |

۱۴/۳ بهینه‌سازی همگرایی

□ نرخ پادگیزی (learning rate) – نرخ پادگیزی اغلب با نماد α و گاهی اوقات با نماد η نمایش داده می‌شود و بیانگر سرعت گام (گام) پیروزرسانی وزن‌ها است که می‌تواند مقداری ثابت داشته باشد یا به صورت سازگارشونده تغییر کند. محبوب‌ترین روش حال حاضر Adam نام دارد، روشی است که نرخ پادگیزی را در هین فرآیند آموزش تنظیم می‌کند.

□ نرخ‌های پادگیزی سازگارشونده – داشتن نرخ پادگیزی متغیر در فرآیند آموزش یک مدل، می‌تواند زمان آموزش را کاهش دهد و راه حل بهینه عددی را بهبود ببخشد. با آنکه بهینه‌ساز Adam محبوب‌ترین روش مورد استفاده است، دیگر روش‌ها نیز می‌توانند مفید باشند. این روش‌ها در جدول زیر به اختصار آمده‌اند:

۱۵/۳ آموزش یک شبکه‌ی عصبی

۱/۳ تعاریف

□ تکرار (epoch) – در مضمون آموزش یک مدل، تکرار اصطلاحی است که مدل در یک دوره تکرار تمامی نمونه‌های آموزشی را برای به روزرسانی وزن‌ها می‌بیند.

□ گرادیان نزولی دسته‌ی کوچک (mini-batch gradient descent) – در فاز آموزش، به روزرسانی وزن‌ها معمولاً بر مبنای تمامی مجموعه‌ی آموزش به علت پیچیدگی‌های محاسباتی، یا یک نمونه داده به علت مشکل نویز، بیست. در عوض، گام به روزرسانی بر روی دسته‌های کوچک انجام می‌شود، که تعداد نمونه‌های داده در یک دسته یک ابرفراسنج است که میتوان آن را تنظیم کرد.

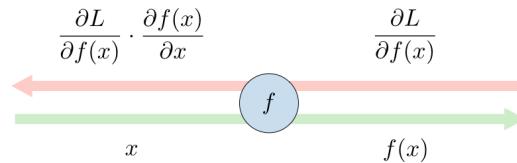
□ (loss function) – به منظور سنجش کارایی یک مدل مفروض، معمولاً ازتابع خطای L برای ارزیابی اینکه تا چه حد خروجی حقیقی y به شکل صحیح توسط خروجی z مدل پیش‌بینی شده‌اند، استفاده می‌شود.

□ خطا آنتروپی متقاطع (cross-entropy loss) – در مضمون دسته‌بندی دودویی در شبکه‌های عصبی، عموماً ازتابع خطای آنتروپی متقاطع $L(z,y)$ استفاده و به صورت زیر تعریف می‌شود:

$$L(z,y) = - \left[y \log(z) + (1-y) \log(1-z) \right]$$

۱۶/۳ یافتن وزن‌های بهینه

□ انتشار معکوس (backpropagation) – انتشار معکوس روشی برای به روزرسانی وزن‌ها با توجه به خروجی واقعی و خروجی مورد انتظار در شبکه‌ی عصبی است. مشتق نسبت به هر وزن w توسط قاعده‌ی زنجیری محاسبه می‌شود.

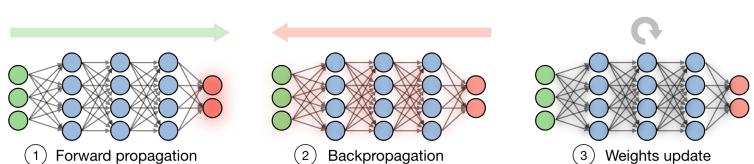


با استفاده از این روش، هر وزن با قانون زیر به روزرسانی می‌شود :

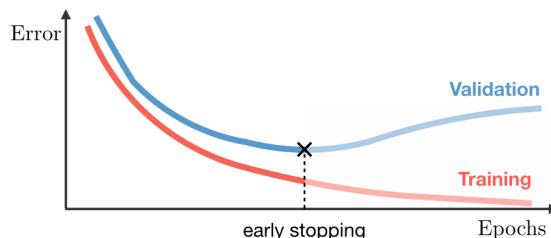
$$w \leftarrow w - \alpha \frac{\partial L(z,y)}{\partial w}$$

□ به روزرسانی وزن‌ها – در یک شبکه‌ی عصبی، وزن‌ها به شکل زیر به روزرسانی می‌شوند :

- **گام ۱ :** یک دسته از داده‌های آموزشی گرفته شده و با استفاده از انتشار مستقیم خطای محاسبه می‌شود
- **گام ۲ :** با استفاده از انتشار معکوس مشتق خطای نسبت به هر وزن محاسبه می‌شود
- **گام ۳ :** با استفاده از مشتقات، وزن‌های شبکه به روزرسانی می‌شوند



□ توقف زودهنگام (early stopping) – این روش نظامبخشی، فرآیند آموزش را به مخف اینکه خطای اعتبارسنجی ثابت می‌شود یا شروع به افزایش پیدا کند، متوقف می‌کند.



۳/۵ عادت‌های خوب

□ بیشبرازش روی دسته‌ی کوچک – هنگام اشکال‌زدایی یک مدل، اغلب مفید است که یک سری آزمایش‌های سریع برای اطمینان از اینکه هیچ مشکل عمده‌ای در معماری مدل وجود ندارد، انجام شود. به طور خاص، برای اطمینان از اینکه مدل می‌تواند به شکل صحیح آموزش ببیند، یک دسته‌ی کوچک (از داده‌ها) به شبکه داده می‌شود تا دریابیم که مدل می‌تواند به آنها بیشبرازش کند. اگر نتواند، بدین معناست که مدل از پیچیدگی بالایی برخوردار است یا پیچیدگی کافی برای بیشبرازش شدن روی دسته‌ی کوچک ندارد، چه برسد به یک مجموعه آموزشی با اندازه عالی.

□ وارسی گرادیان (gradient checking) – وارسی گرادیان روشی است که در طول پیاده‌سازی گذر روبره‌عقب یک شبکه‌ی عصبی استفاده می‌شود. این روش مقدار گرادیان تحلیلی را با گرادیان عددی در نقطه‌های مفروض مقایسه می‌کند و نقش بررسی درستی را ایفا می‌کند.

| گرادیان تحلیلی | گرادیان عددی | فرمول |
|--------------------------------------|---|-------|
| $\frac{df}{dx}(x) = f'(x)$ | $\frac{df}{dx}(x) \approx \frac{f(x+h) - f(x-h)}{2h}$ | |
| - نتیجه 'عینی' | - پرهزینه (از نظر محاسباتی)، خطای باید دو بار برای هر بعد محاسبه شود | |
| - محاسبه مستقیم | - برای تایید صحت پیاده‌سازی تحلیلی استفاده می‌شود | |
| - در پیاده‌سازی نهایی استفاده می‌شود | - مصالحه در انتخاب h : | |
| | - نه بسیار کوچک (نایابی عددی) و نه خیلی بزرگ (تخمین گرادیان ضعیف) باشد | |

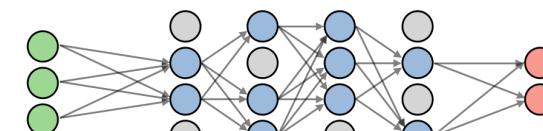
* * *

| روش | توضیح | بهروزرسانی b | بهروزرسانی w |
|----------|--|---|---|
| Momentum | - نوسانات را تعدیل می‌دهد - بهبود SGD - دو فراستنگ که نیاز به تنظیم دارند | $b \leftarrow b - \alpha v_{db}$ | $w \leftarrow w - \alpha v_{dw}$ |
| RMSprop | - انتشار جذر میانگین مریعات - سرعت بخشیدن به الگوریتم یادگیری با کنترل نوسانات | $b \leftarrow b - \alpha \frac{db}{\sqrt{s_{db}}}$ | $w \leftarrow w - \alpha \frac{dw}{\sqrt{s_{dw}}}$ |
| Adam | - تخمین سازگارشونده ممان - محبوب‌ترین روش - چهار فراستنگ که نیاز به تنظیم دارد | $b \leftarrow b - \alpha \frac{v_{db}}{\sqrt{s_{db}} + \epsilon}$ | $w \leftarrow w - \alpha \frac{v_{dw}}{\sqrt{s_{dw}} + \epsilon}$ |

نکته : سایر متدها شامل SGD و Adagrad, Adadelta هستند.

۴/۱ نظامبخشی

□ بروون‌اندازی (dropout) – بروون‌اندازی روشی است که در شبکه‌های عصبی برای جلوگیری از بیشبرازش بر روی داده‌های آموزشی با حدف نعمادی نورون‌ها با احتمال $p > 0$ استفاده می‌شود. این روش مدل را مجبور می‌کند تا از تکیه کردن بیش‌ازحد بر روی مجموعه خاصی از ویژگی‌ها خودداری کند.



نکته : بیشتر کتابخانه‌های یادگیری عمیق بروون‌اندازی را با استفاده از فراستنگ 'نگهداشت' $-p - 1$ کنترل می‌کنند.

□ نظامبخشی وزن – برای اطمینان از اینکه (مقادیر) وزن‌ها بیش‌ازحد بزرگ نیستند و مدل به مجموعه آموزش بیشبرازش نمی‌کند، روش‌های نظامبخشی معمولاً بر روی وزن‌های مدل اجرا می‌شوند. اصلی‌ترین آنها در جدول زیر به اختصار آمده‌اند :

| Elastic Net | Ridge | LASSO |
|--|--|--|
| بین انتخاب متغیر و ضرایب کوچک مصالحه می‌کند | ضرایب را کوچکتر می‌کند | - ضرایب را تا صفر کاهش می‌دهد - برای انتخاب متغیر مناسب است |
| | | |
| $(1-\alpha) \theta _1 + \alpha \theta _2^2 \leq 1$ | $ \theta _2 \leq 1$ | $ \theta _1 \leq 1$ |
| $\dots + \lambda \left[(1-\alpha) \theta _1 + \alpha \theta _2^2 \right]$ $\lambda \in \mathbb{R}, \alpha \in [0,1]$ | $\dots + \lambda \theta _2^2$ $\lambda \in \mathbb{R}$ | $\dots + \lambda \theta _1$ $\lambda \in \mathbb{R}$ |