پرسش و پاسخ تصویری در فارسی

مریم سادات هاشمی
دانشکده مهندسی کامپیوتر
دانشگاه علم و صنعت ایران
m_hashemi94@cs.iust.ac.ir

علیرضا اصغری
دانشکده مهندسی کامپیوتر
دانشگاه علم و صنعت ایران
a_asghari@comp.iust.ac.ir

چکیده

در سال‌های اخیر پیشرفت‌های زیادی در مسائل هوش مصنوعی و یادگیری عمیق که در تقاطع دو حوزه پردازش زبان طبیعی و بینایی ماشین قرار می‌گیرند؛ رخ داده است. یکی از مسائلی که اخیرا مورد توجه قرارگرفته است؛ پرسش و پاسخ تصویری است. در این مسئله، با توجه به یک تصویر و یک سؤال به زبان طبیعی، سیستم سعی می‌کند با استفاده از عناصر بصری تصویر و استنتاج جمع‌آوری شده از سوال متنی، پاسخ صحیح را پیدا کند. هدف ما در این پروژه، حل مسئله‌ی VQA در زبان فارسی است. بدین منظور مجموعه‌داده‌ای را فراهم کردیم و سه روش LSTM Q + norm I ، Stacked Attention Network و HieCoAttention را بر روی این مجموعه‌داده پیاده‌سازی و اجرا کردیم.

۱ مقدمه

در طی سال‌های متمادی، محققان به دنبال ساخت ماشین‌هایی بودند که به اندازه‌ی کافی باهوش باشند که از آن به طور موثر همانند انسان‌ها برای تعامل استفاده کنند. مسئله‌ی پرسش و پاسخ تصویری یکی از پله‌های رسیدن به این رویای هوش مصنوعی است و از این جهت حائز اهمیت است.

پرسش و پاسخ تصویری نسخه گسترش یافته مسئله پرسش و پاسخ متنی است که اطلاعات بصری به مسئله اضافه شده است. شکل ۱ گویای تفاوت این دو مسئله است. در سیستم پرسش و پاسخ متنی، یک متن و یک سوال متنی به عنوان ورودی به سیستم داده می‌شود و انتظار می‌رود که سیستم با توجه به درک و تفسیری که از متن و سوال بدست می‌آورد؛ یک جواب متنی

شکل ۱: مثالی از سیستم پرسش و پاسخ متنی و تصویری

را خروجی دهد. اما در سیستم پرسش و پاسخ تصویری، یک تصویر و یک سوال متنی به ورودی سیستم داده می‌شود و انتظار می‌رود که سیستم بتواند با استفاده از عناصر بصری تصویر و تفسیری که از سوال بدست می‌آورد؛ یک پاسخ متنی را در خروجی نشان دهد. مسئله پرسش و پاسخ تصویری پیچیدگی بیشتری نسبت به مسئله پرسش و پاسخ متنی دارد زیرا تصاویر بعد بالاتر و نویز بیشتری نسبت به متن دارند. علاوه بر این، تصاویر فاقد ساختار و قواعد دستوری زبان هستند. در نهایت هم، تصاویر غنای بیشتری از دنیای واقعی را ضبط می‌کنند، در حالی که زبان طبیعی در حال حاضر نشانگر سطح بالاتری از انتزاع دنیای واقعی است [۱۴].

کاربردهای بسیاری برای پرسش و پاسخ تصویری وجود دارد. یکی از مهم‌ترین موارد دستیار هوشمند برای افراد کم‌بینا و نابینا [۱] است [۳]. علاوه بر این، در سال‌های اخیر دستیاران صوتی [۲] و عامل‌های گفتگو [۳] مانند Siri ، Cortana و Alexa در بازار عرضه شدند که می‌توانند با انسان‌ها با استفاده از زبان طبیعی ارتباط برقرار کنند. در حال حاضر این دستیاران با استفاده از صوت و متن این ارتباط را برقرار می‌کنند در نتیجه گفتگوی بین این دستیاران با انسان‌ها مشابه دنیای واقعی نمی‌باشد. این ارتباط را می‌توان با استفاده از داده‌های تصویری و ویدئویی به واقعیت نزدیک‌تر کرد. اینجاست که مسئله‌ی پرسش و پاسخ تصویری برای نزدیک کردن تعامل بین انسان و عامل‌های گفتگو به دنیای واقعی می‌تواند موثر باشد. همین موضوع را می‌توانیم به صورت گسترده‌تری در ربات‌ها مشاهده کنیم. برای این‌که ربات بتواند بهتر با انسان‌ها ارتباط برقرار کند و به سوالات و درخواست‌ها پاسخ دهد؛ نیاز دارد که درک و فهم درستی از اطراف داشته باشد که این مستلزم داشتن تصویری دقیق از پیرامون است. بنابراین این ربات می‌تواند برای پاسخ به پرسش‌ها از دانشی که از طریق تصویر پیرامون خود بدست می‌آورد، جواب درستی را بدهد. کاربرد دیگر این مسئله در پزشکی است. در بسیاری موارد تحلیل تصاویر پزشکی مانند تصاویر CT اسکن و x-ray برای یک پزشک متخصص هم دشوار است. اما یک سیستم پرسش و پاسخ تصویری می تواند با تحلیل و تشخیص موارد غیرطبیعی موجود در تصویر، به عنوان نظر دوم به پزشک متخصص کمک کند. از طرفی ممکن است در بعضی اوقات بیمار دسترسی به پزشک را نداشته باشد تا شرح تصاویر را متوجه شود. وجود سیستم پرسش و پاسخ تصویری می‌تواند آگاهی بیمار را نسبت به بیماری افزایش دهد و از نگرانی او بکاهد.

۲ کارهای مرتبط / پیش‌زمینه

در سال‌های اخیر، رویکردهای بیشماری برای VQA پیشنهاد شده است. همه رویکردهای موجود شامل موارد زیر است:

۱. رویکردهای مبتنی بر ترکیب ویژگی

۲. رویکردهای مبتنی بر attention

۳. رویکرد های مبتنی بر استدلال

در این پروژه ما از سه روش استفاده کرده‌ایم که روش LSTM Q + norm I مبتنی بر ترکیب ویژگی هاست و دو روش Stacked Attention Network و HieCoAttention مبتنی بر attention هستند. بر این اساس، کارهای انجام شده در این دو دسته را مرور خواهیم کرد.

۱.۲ رویکردهای مبتنی بر ترکیب ویژگی

این رویکردها هم ویژگی‌های تصویری و هم ویژگی‌های سوال را به یک فضای مشترک برای پیش‌بینی پاسخ منتقل می‌کنند. برای استخراج ویژگی‌های تصاویر، اکثر الگوریتم‌ها از CNN های از قبل آموزش دیده استفاده می‌کنند که بر روی مجموعه داده ImageNet آموزش داده شده‌اند. برخی از شبکه‌های رایج عبارتند از: GoogLeNet[۱۳] ، ResNet [۴] و VGGNet [۱۲]. برای استخراج ویژگی‌ها از سوالات، از روش‌هایی مانند کیسه کلمات (BOW) ، GRU [۲] و LSTM [۵] استفاده می‌شود. در این رویکرد عموماً مسئله VQA را یک مسئله طبقه‌بندی در نظر می‌گیرند و روش‌های متعددی برای ترکیب ویژگی‌های تصویر و سوال وجود دارد. بعضی از این روش‌ها ساده می‌باشند از جمله: concatenation ، elementwise addition ، elementwise multiplication و bilinear pooling . اما ممکن است از روش‌های پیچیده‌تری مانند Bayesian models نیز استفاده شود. دقتی که از روش‌های مبتنی بر این رویکرد بدست می‌آید متفاوت است و وابستگی زیادی به انتخاب هایپرپارامترها، پیکربندی سیستم و تنظیمات آزمایش‌ها دارد.

---

[۱] https://vizwiz.org/

[۲] Voice Assistant

[۳] Conversational Agents

	تعداد تصاویر	تعدادسوالات	تعداد پاسخ‌ها
داده‌های آموزشی	۸۲،۷۸۳	۲۴۸،۳۴۹	۲،۴۸۳،۴۹۰
داده‌های ارزیابی	۴۰،۵۰۴	۱۲۱،۵۱۲	۱،۲۱۵،۱۲۰
داده‌های تست	۸۱،۴۳۴	۲۴۴،۳۰۲	

جدول ۱: مشخصات مجموعه‌داده VQA

۲.۲ رویکردهای مبتنی بر attention

مدل‌های مبتنی بر attention به ناحیه‌هایی از تصاویر که مربوط به سوال است، توجه می‌کنند. مدل‌های موجود در این رویکرد یا به تصویر و یا به سوال و یا به هر دو توجه می‌کنند. به عنوان مثال، در [۱۱] مدلی را پیشنهاد داده است که با انتخاب یک منطقه تصویری که مربوط به متن سؤال باشد، پاسخ را پیش بینی می‌کند. در این روش به به تصویر توجه شده است. اما در مثالی دیگر [۹] از چندین لایه coattention استفاده می‌کند و هر کلمه از سوال با هر منطقه در تصویر در تعامل است و بالعکس. روش‌های پیشنهادی در این رویکرد بسیار است مانند linear Attention Network (BAN) [۷] و Question Type guided Attention (QTA) [۱۰] .

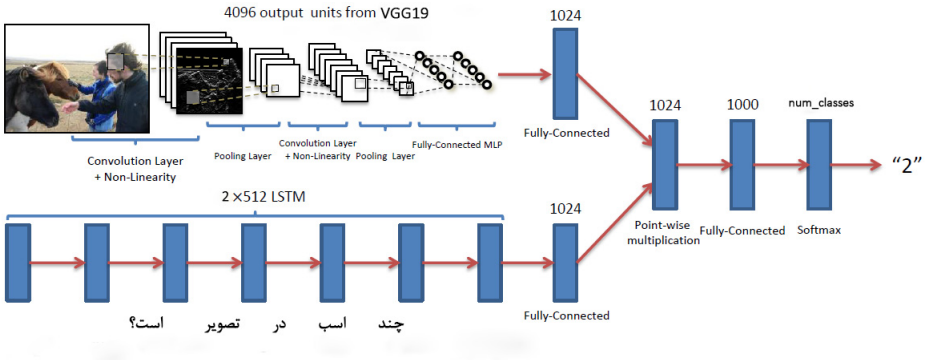
۳ مدل پیشنهاد شده

در این بخش ابتدا نحوه‌ی آماده‌سازی مجموعه‌داده را توضیح می‌دهیم و سپس به شرح روش‌های پیاده‌سازی شده در این پروژه می‌پردازیم.

۱.۳ تهیه‌ی مجموعه‌داده

مجموعه‌داده‌ای که برای حل این مسئله انتخاب کردیم؛ مجموعه داده VQA v1 است. مشخصات کامل مجموعه داده را می‌توانید در جدول ۱ مشاهده کنید. برای ترجمه مجموعه‌داده از دو ابزار Google و ترگمان استفاده کردیم. در این مجموعه‌داده برای هر تصویر سه سوال وجود دارد و برای هر سوال ۱۰ پاسخ موجود می‌باشد. در این مجموعه‌داده سه نوع سوال وجود دارد. نوع اول بله و خیر است. نوع دوم تعداد یک شی در تصویر است و نوع سوم مربوط به سوالات دیگر است.

۲.۳ LSTM Q + norm I [۱]



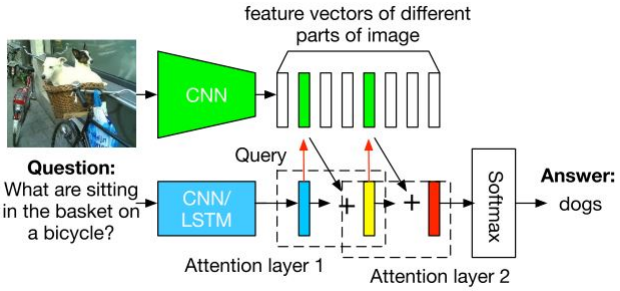
شکل ۲: ساختار کلی روش LSTM Q + norm I .

این روش ساده‌ترین روش یادگیری عمیق برای حل مسئله پرسش و پاسخ تصویری است. در اینجا مسئله VQA به عنوان یک مسئله طبقه‌بندی در نظر گرفته می‌شود که در آن ۱۰۰۰ پاسخ پرتکرار به عنوان کلاس‌ها انتخاب می‌شوند. ساختار کلی این شبکه در شکل ۲ نشان داده شده است. ابتدا با عبور دادن تصاویر از شبکه VGG19 برای هر تصویر یک بردار ویژگی ۴۰۹۶تایی در لایه‌ی ماقبل آخر در شبکه‌ی VGG19 تولید می‌شود. از طرفی دیگر با عبور سوال‌ها از لایه‌یEembedding برای هر کلمه موجود در سوال یک بردار ۳۰۰تایی تولید می‌شود. سپس از طریق ۲ لایه LSTM بردار ویژگی معنایی سوال استخراج می‌شود. هر یک از بردارهای ویژگی تصویر و سوال را به یک لایه‌ی Dense ۱۰۲۴ واحدی می‌دهیم تا ابعاد بردارها مشابه هم شوند.

برای ترکیب بردار ویژگی سوال و تصویر از ضرب نقطه‌ای استفاده می‌کنیم. از این بردار ترکیب شده به عنوان ورودی برای لایه‌ی کاملا متصل استفاده می‌کنیم و در نهایت با عبور از یک لایه softmax کلاس(پاسخ) پیش‌بینی شده بدست می‌آید.

۳.۳ Stacked Attention Network [۱۵]

ایده‌ی اصلی روش SAN این است که ابتدا از سوال، یک بازنمایی معنایی و مفهومی استخراج می‌کند. سپس از آن به عنوان یک کوئری برای پیدا کردن مناطقی از تصویر که مرتبط با سوال است؛ استفاده می‌کند. غالباً در مسئله VQA نیاز است تا چندین مرحله استدلال صورت بگیرد. بنابراین در این شبکه از چندین لایه برای جستجو در تصویر استفاده می‌کنیم تا به تدریج به جواب مورد نظر برسیم. ساختار کلی شبکه SAN را در شکل ۳ می‌توانید مشاهده کنید. شبکه SAN از سه جز اصلی تشکیل شده است: ۱) مدل تصویر که با استفاده از CNN ویژگی‌های سطح بالایی را از تصویر استخراج می‌کند. ۲) مدل سوال که با استفاده از CNN یا LSTM ویژگی‌های معنایی سوال را استخراج می‌کند. ۳) مدل stacked attention که از طریق استدلال چند مرحله‌ای مناطقی از تصویر که مرتبط به سوال است را پیدا می‌کند تا پاسخ را پیش‌بینی کند.



شکل ۳: ساختار کلی روش SAN .

۱.۳.۳ مدل تصویر

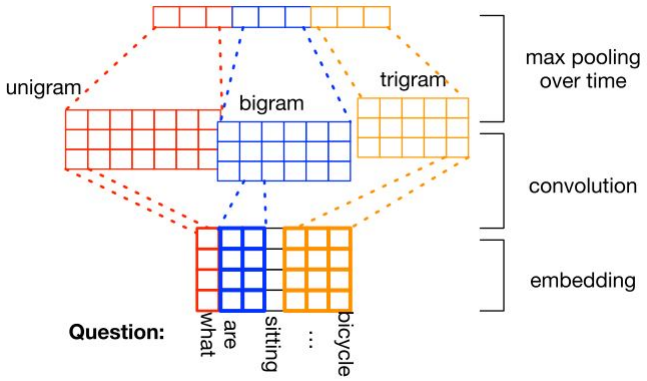
در این بخش برای استخراج ویژگی از شبکه‌ی VGG16 استفاده می‌کنیم و ویژگی‌ها را از آخرین لایه‌ی pooling شبکه بدست می‌آوریم. ابتدا تمام تصاویر را به 448×448 تغییر سایز می‌دهیم و بعد از این که تابع پیش‌پردازش موجود برای شبکه‌ی VGG16 را بر روی تصاویر اعمال کردیم، تصاویر را برای استخراج ویژگی به شبکه می‌دهیم. بنابراین برای هر تصویر یک ویژگی با ابعاد $14\times14\times512$ حاصل می‌شود. در حقیقت، برای هر تصویر به تعداد 14×14 منطقه استخراج می‌شود که هر منطقه به وسیله‌ی یک بردار ویژگی ۵۱۲تایی بازنمایی می‌شود. برای راحتی، از یک لایه‌ی Dense بعد از شبکه‌ی VGG16 استفاده می‌کنیم تا ابعاد بردار ویژگی مناطق مشابه با ابعاد بردار ویژگی سوال شود.

۲.۳.۳ مدل سوال

برای استخراج ویژگی‌های معنایی از سوال، از هر دو روش LSTM و CNN یک بعدی استفاده می‌کنیم. در هر دو روش ابتدا سوال را به یک دنباله‌ی عددی تبدیل می‌کنیم و سپس این دنباله‌ها را به یک لایه‌ی Embedding می‌دهیم. در روش LSTM خروجی لایه Embedding را به دو لایه‌ی LSTM می‌دهیم و خروجی آخرین لایه‌ی مخفی LSTM را به عنوان بردار ویژگی سوال در نظر می‌گیریم. در روش CNN خروجی Embedding را به سه لایه‌ی کانولوشنی یک بعدی با فیلترهایی با سایز ۱، ۲، ۳ می‌دهیم که به ترتیب ترکیب‌های یک کلمه‌ای ، دو کلمه‌ای و سه کلمه‌ای را برای ما استخراج می‌کند. در نهایت بر روی خروجی هر سه لایه تابع maxpooling را اعمال می‌کنیم و با قرار دادن این سه خروجی در کنار هم بردار ویژگی سوال بدست می‌آید. شکل ۴ مدل سوال بر اساس CNN را نشان می‌دهد.

۳.۳.۳ مدل stacked attention

در این بخش، مدل stacked attention با توجه به ماتریس ویژگی تصویر و بردار ویژگی سوال پاسخ را از طریق استدلال چند مرحله‌ای پیش‌بینی می‌کند. در بسیاری از موارد، یک پاسخ فقط مربوط به یک ناحیه کوچک از تصویر است. بنابراین، استفاده از یک ماتریس ویژگی کلی برای تصویر می‌تواند به دلیل وجود نویزهای مناطق بی‌ربط به پاسخ، منجر به نتایج نامطلوبی شود. در عوض، استدلال از طریق چندین لایه توجه، قادر است به تدریج مناطق غیرمرتبط با جواب را فیلتر کند و از ماتریس



شکل ۴: مدل سوال براساس CNN .

ویژگی تصویر حذف کند. بدین منظور ماتریس ویژگی تصویر v_I و بردار ویژگی سوال v_Q ، را به یک لایه Dense می‌دهیم و خروجی این لایه را به یک تابع softmax می‌دهیم تا توزیع توجه را بر روی نواحی تصویر بدست آوریم. بنابراین داریم:

$$h_A = tanh(W_{I,A} v_I \oplus (W_{Q,A} v_Q + b_A)) \quad (۱)$$

$$p_I = softmax(W_P h_A + b_P) \quad (۲)$$

بر اساس توزیع توجه p_i ، جمع وزن‌دار بردارهای تصویر را که هر کدام متناظر به یک منطقه هست را محاسبه می‌کنیم. سپس $\tilde{v}_I$ را با بردار ویژگی سوال ترکیب می‌کنیم و یک کوئری برای لایه‌ی بعدی توجه ایجاد می‌کنیم.

$$\tilde{v}_I = \sum_i p_i v_i, \quad (۳)$$

$$u = \tilde{v}_I + v_Q. \quad (۴)$$

این روش را به تعداد k بار تکرار می‌کنیم. در نهایت از u در لایه‌ی k برای پیش‌بینی پاسخ استفاده می‌کنیم:

$$p_{ans} = softmax(W_u u^K + b_u) \quad (۵)$$

۴.۳ HieCoAttention [۸]

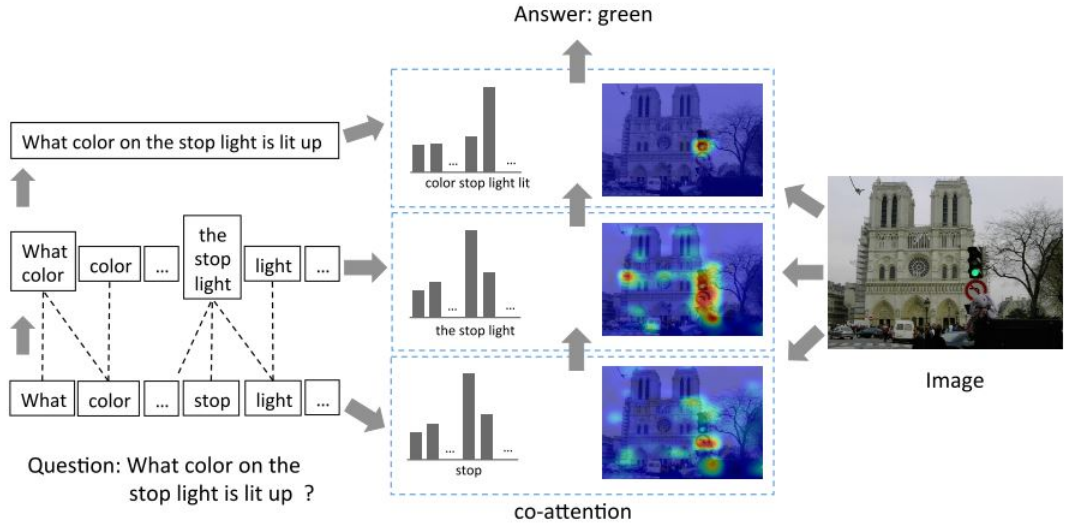
روش پیشنهاد شده در [۸] دارای دو ویژگی مهم است. ویژگی اول بازنمایی سلسله‌مراتبی سوال و ویژگی دوم مکانیزم coatten-tion می‌باشد. در ادامه این دو خصوصیات را شرح می‌دهیم.

۱.۴.۳ بازنمایی سلسله‌مراتبی سوال

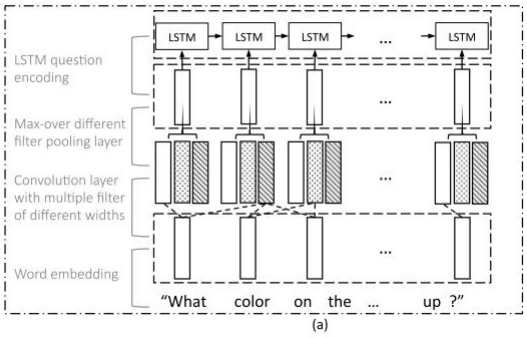
در این بخش برای هر سوال سه سطح Embedding را محاسبه می‌کنیم. اولین Embedding مربوط به کلمات است که بعد از این‌که سوال را به دنباله‌های عددی تبدیل کردیم؛ با عبور دادن این دنباله‌ها از لایه‌ی Embedding ، بردارهای Embedding کلمات بدست می‌آید. برای محاسبه سطح بعدی Embedding که مربوط به عبارات است از کانولوشن‌های یک بعدی با فیلترهایی با سایز ۱، ۲ و ۳ استفاده می‌کنیم و سپس با اعمال تابع Maxpooling بردار Embedding هر عبارت بوجود می‌آید. در نهایت از Embedding عبارات برای محاسبه‌ی Embedding کل سوال استفاده می‌کنیم. این کار توسط یک لایه LSTM انجام می‌شود. بنابراین برای هر سوال به صورت سلسله‌مراتبی سه سطح Embedding کلمه، عبارت و سوال تولید می‌شود. بازنمایی سلسله‌مراتبی سوال در شکل ۶ به تصویر کشیده شده است.

۲.۴.۳ مکانیزم coattention

در [] دو مکانیزم برای coattention پیشنهاد شده است که از نظر ترتیب تولید attention map برای سوال و تصویر با هم تفاوت دارند. اولین مکانیزم که parallel coattention نامیده می‌شود، باعث تولید attention به طور همزمان برای سوال

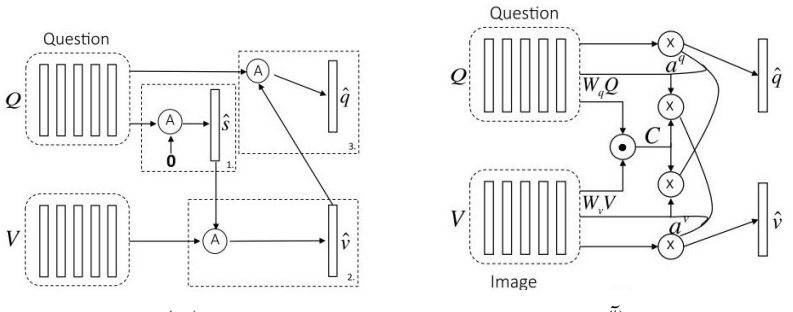


شکل ۵: ساختار کلی روش HieCoAttention



شکل ۶: بازنمایی سلسله‌مراتبی سوال.

و تصویر می‌شود. به مکانیزم دوم alternating coattention می‌گویند که برای تولید attention برای سوال و تصویر به صورت تناوبی عمل می‌کند (شکل ۷). این مکانیزم coattention در هر سه سطح سلسله‌مراتبی سؤال اجرا می‌شوند. در این پروژه ما از مکانیزم parallel coattention استفاده می‌کنیم. در این مکانیزم با محاسبه شباهت بین ویژگی‌های تصویر و سوال، تصویر و سؤال را به هم متصل می‌کنیم. اگر بردار ویژگی تصویر را با V و بازنمایی سوال را با Q نشان دهیم؛ ماتریس



(ب) (آ)

شکل ۷: (آ) parallel coattention (ب) alternating coattention

شباهت C به صورت زیر محاسبه می‌شود:

$$C = tanh(Q^T W_b V) \tag{۶}$$

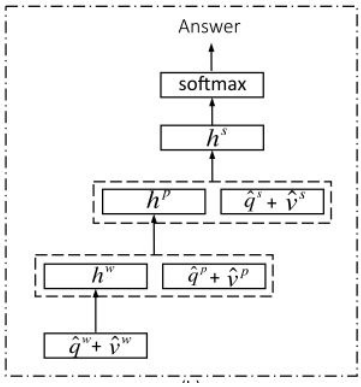
پس از محاسبه ماتریس شباهت، برای محاسبه بردار وزن‌های attention برای تصویر و سوال از روابط زیر استفاده می‌کنیم:

$$\begin{aligned} H^v &= tanh(W_v V + (W_q Q)C), \quad & H^q &= tanh(W_q Q + (W_v V)CT) \\ a^v &= softmax(w_{hv}^T H^v), \quad & a^q &= softmax(w_{hq}^T H^q) \end{aligned} \tag{۷}$$

که در عبارت ۷ W_v ، W_q ، w_{hv} و w_{hq} پارامترهای وزن هستند. a_v و a_q نیز به ترتیب وزن‌های attention برای تصویر و سوال هستند. با توجه به وزن‌های attention ، بردارهای توجه تصویر و سوال به وسیله‌ی جمع وزن‌دار ویژگی‌های تصویر و ویژگی‌های سوال با وزن‌های attention محاسبه می‌شوند:

$$\hat{v} = \sum_{n+1}^{N} a_n^v v_n, \quad \hat{q} = \sum_{t=1}^{T} a_t^q q_t \tag{۸}$$

۳.۴.۳ پیش‌بینی پاسخ



شکل ۸: پیش‌بینی پاسخ

ما پاسخ را بر اساس coattention تصویر و سوال بدست آمده در هر سه سطح Embedding پیش‌بینی می‌کنیم. از یک پرسپترون چندلایه (MLP) استفاده می‌کنیم تا ویژگی‌های attention را همان طور که در شکل ۸ نشان داده شده است؛ ترکیب کنیم.

$$\begin{aligned} h^w &= tanh(W_w(\hat{q}^w + \hat{v}^w)) \\ h^p &= tanh(W_p[\hat{q}^p + \hat{v}^p), h^w]) \\ h^s &= tanh(W_s[(\hat{q}^s + \hat{v}^s), h^p]) \\ p &= softmax(W_h h^s) \end{aligned} \tag{۹}$$

W_W ، W_p ، W_s و W_h پارامترهای وزن هستند. p احتمال پاسخ نهایی است.

۴ نتایج و تحلیل

در این بخش ابتدا روش ارزیابی مورد استفاده در این مسئله را بررسی می‌کنیم. سپس به این موضوع می‌پردازیم که مشکل بیش برازش در اینجا چگونه تعریف می‌شود و برای حل آن باید چه کارهایی انجام دهیم. در نهایت نتایج هر کدام از روش‌ها را بررسی خواهیم کرد. دقت شود که در برخی از جدول‌ها از اصلاح hard و soft استفاده شده است. که hard به معنای این است که Early stopping بر اساس خطای داده‌های ارزیابی است و در این حالت مدل حداکثر قدرت خود نخواهد رسید. soft به این معنی است که Early stopping براساس دقت داده‌های ارزیابی است و مدل به حداکثر قدرت خود خواهید رسید. کدهای پروژه را می‌توانید در گیت هاب مشاهده کنید.

۱.۴ پروتکل ارزیابی

در مقاله [۱] ، معیار ارزیابی خاصی برای این مساله استفاده شده است. به عبارتی دقت را مانند روش‌های معمول در یادگیری ماشین محاسبه نمی‌کنیم. برای محاسبه دقت ارزیابی پاسخ‌های تولید شده برای سوال (نه انتخاب از بین پاسخ‌های چندگزینه‌ای) درمجموعه مورد ارزیابی، فرمول زیر را داریم :

$$accuracy = min(\frac{\#humansthatprovidedthatanswer}{3}, 1) \qquad (۱۰)$$

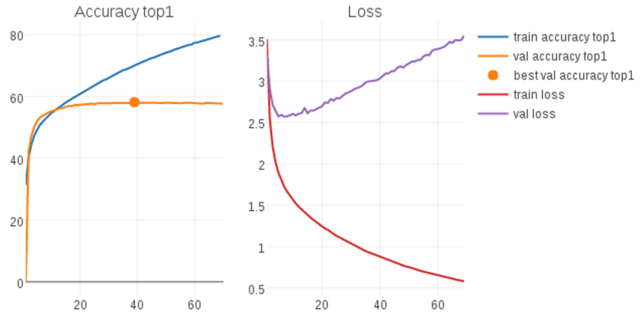
در مجموعه‌داده اصلی برای داده‌های آموزش و ارزیابی، به ازای هر سوال ۱۰ پاسخ انسانی گردآوری شده است که افراد مختلف به آن سوال با سطوح اطمینان مختلف، پاسخ داده‌اند. حال با توجه به این‌که ما مدل را بر روی مجموعه آموزش، آموزش داده‌ایم و بر روی مجموعه ارزیابی، آن را مورد ارزیابی قرار می‌دهیم، دقت را از این طریق محاسبه می‌کنیم.

طبق این فرمول برای این‌که پاسخ پیش‌بینی شده توسط ماشین به یک سوال کاملا صحیح در نظر گرفته شود، می‌بایست پاسخ ماشین، با پاسخ حداقل ۳ عامل انسانی یکسان باشد و در نتیجه امتیاز کامل ۱ را از ارزیابی آن سوال دریافت می‌کند. با توجه به فرمول اگر ۲ نفر پاسخشان با پاسخ مدل یکی باشد، امتیاز ۶۶.۰ و اگر پاسخ مدل تنها با پاسخ یک نفر از آن ۱۰ عامل انسانی یکسان باشد، امتیاز ۳۳.۰ دریافت می‌کند.

همانطور که می‌بینیم ارزیابی سهل گیرانه تری نسبت به ارزیابی معمولی یک پاسخ (که وضعیت ۰ یا صدی صحت برای آن متصور می‌شود)، معرفی شده‌است. این روش برای ارزیابی مسئله VQA برای اولین بار در [۱] معرفی شده است. در حال حاضر که حدود ۵ سال از انتشار آن می‌گذرد، قریب به اتفاق مقالات و روش‌های دیگر در این زمینه از این فرمول برای ارزیابی روش یادگیری خود استفاده نموده‌اند و به عنوان فرمولی استاندارد و مناسب به عنوان پروتکل ارزیابی در VQA شناخته شده است.

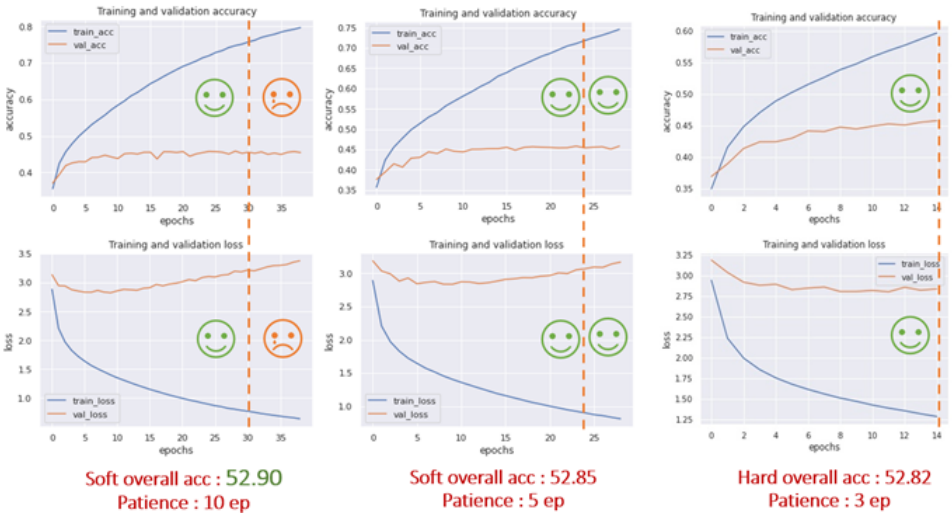
۲.۴ بیش برازش، زمان رخ دادن و حل آن

در فاز پیشرفت پروژه، نتایجی که بر روی روش پایه مقاله با داده‌های کم بدست آوردیم؛ نشان از بیش برازش داشت. در این فاز بررسی‌ها و آزمایش‌های متعددی انجام دادیم تا علت مشکل و روش حل آن را بیابیم. مقاله [۶] ریشه این مسئله و رواج داشتن آن به شکل کلی و معمول در VQA را بررسی کرده است. زمانی که معیار خطا را cross entropy در نظر می‌گیریم؛ به صورت سخت‌گیرانه عمل می‌کنیم زیرا اگر پاسخ مدل دقیقا برابر با پاسخ صحیح نباشد، مدل جریمه می‌شود. این در حالی است که عبارت ۱۰ از روش سهل گیرانه‌تری استفاده می‌کند. در نتیجه این امر باعث اختلاف بین معیار خطا cross entropy و دقت مدل می‌شود. در [۶] به معرفی معیار خطا soft cross entropy پرداخته شده است و در محاسبه خطا، تمامی پاسخ‌های انسانی را درنظر می‌گیرد که اختلاف بین رفتار loss و دقت کاهش می‌یابد و همگرایی آموزش مدل نیز بهبود می‌یابد. بنابراین معیار اصلی برای تشخیص بیش برازش، استفاده از عبارت ۱۰ است که به عنوان پروتکل استاندارد ارزیابی در این حوزه شناخته می‌شود.



شکل ۹

برای مثال در شکل ۹ از نظر cross entropy بیش برازش داریم اما از نظر پروتکل ارزیابی بیش برازش اتفاق نیفتاده است و بهترین دقت در گام ۴۰ رخ داده است.



شکل ۱۰

شکل ۱۰ نتایج سه آزمایشی که با استفاده از Early stopping برمبنای دقت با patience های متفاوت و خطای cross entropy اجرا شده است را نشان می‌دهد. در ستون سمت راست از p برابر با ۳ استفاده شده است و یادگیری در گام چهاردهم متوقف شده است. در این آزمایش از نظر خطا، بیش برازشی اتفاق نیفتاده است و به دقت 52.82 رسیده‌ایم. در آزمایش دوم p را بیشتر کرده‌ایم و یادگیری در گام ۲۴ به اتمام می‌رسد.از نظر تابع خطا تا گام ۲۴ هم بیش برازش اتفاق افتاده است. اما دقت بیشتر شده است پس طبق پروتکل بیش برازشی اتفاق نیفتاده است. در ستون سوم با افزایش p یادگیری در گام ۳۰ به پایان می‌رسد و دقت از مرحله قبل بیشتر شده است. این نشان می‌دهد که در ستون دوم حتی بعد از گام ۲۴ نیز بیش برازش رخ نداده است و تنها جایی که احتمالاً دچار بیش برازش اتفاق افتاده است، از گام ۳۰ به بعد است (چرا که ممکن است اگر p را بیشتر می کردیم سناریوی آزمایش‌های قبل تکرار شود.). در نهایت بهترین مدل در گام ۳۰ می‌باشد.

زمانی که مشکل بیش برازش رخ می‌دهد، برای حل آن از نرمالیزه سازی I۲ تصاویر، recurrent dropout و بهینه سازهای متفاوت مثل adam و adadelta و یا RmsProp با نرخ یادگیری کاهشی استفاده می‌کنیم.

LSTM Q + norm I ۳.۴

آزمایش‌های متعددی برای این روش انجام شده است که جزئیات کامل آن در فایل‌های ضمیمه قرار داده شده است. در اینجا به بررسی مهم‌ترین نتایج می‌پردازیم.

پس از بررسی نتایج آزمایش‌های ضمیمه شده به نتایج زیر دست یافتیم:

۱. بهترین بهینه‌ساز adadelta با نرخ یادگیری ۱ است که همگرایی مدل را افزایش می‌دهد.
۲. برای پیش پردازش سوال ها از کتابخانه hazm استفاده کرده‌ایم که منجر به افزایش دقت شد و همچنین روش pre در padding عملکرد بهتری از حالت post داشت .
۳. برای embedding سوال‌ها از fasttext استفاده می‌کنیم زیرا عملکرد آن نسبت به Glove بهتر است.
۴. استفاده از recurrent dropout به جای dropout معمولی نتایج را بهبود می‌دهد.
۵. استفاده از BatchNormalization دقت مدل را افزایش می‌دهد.

نتایج این مدل را برای زبان فارسی در جدول ۲ می‌توانید مشاهده کنید. در هر دو حالت سخت و نرم دقت برای ترجمه‌های Google بهتر از ترگمان است و همان طور که پیش بینی می‌کردیم دقت مدل‌های نرم بیشتر از مدل‌های سخت است و نشان می‌دهد که مدل از حداکثر ظرفیتش استفاده کرده است.

	Google Translation				Targoman Translation			
Method	**yes/no**	**Number**	**Other**	**All**	**yes/no**	**Number**	**Other**	**All**
lstm Q + VGG19(hard)	76.14	32.97	35.78	50.53	75.58	32.61	33.53	49.15
lstm Q + VGG19(soft)	76.74	32.5	36.98	**51.3**	76.86	31.85	36.26	**50.91**

جدول ۲: دقت روش baseline بر روی مجموعه‌داده فارسی تهیه شده.

همین آزمایش را برای زبان انگلیسی در دو حالت استفاده از ویژگی‌های آماده یا ویژگی‌های تولید شده توسط خودمان اجرا کردیم. نتایج در جدول ۳ آورده شده است. در حالت نرم دقت ها به دقت [۱] بسیار نزدیک شده است. فاصله‌ی دقت‌ها در هر دو حالت نرم و سخت بین ویژگی‌های آماده و ویژگی‌هایی که ما تولید کرده‌ایم بسیار کم است و این نشان می‌دهد که توانسته‌ایم این بخش را تا حد خوبی به درستی پیاده‌سازی و اجرا کنیم.

	English-paperToken				English-kerasToken			
Method	**yes/no**	**Number**	**Other**	**All**	**yes/no**	**Number**	**Other**	**All**
lstm Q + VGG19(hard)	78.43	33.7	37.99	52.58	78.53	31.91	38.78	52.79
lstm Q + VGG19(soft)	79.34	32.69	40.41	**54.01**	79.41	33.62	39.42	**53.66**

جدول ۳: دقت روش baseline بر روی مجموعه‌داده انگلیسی .

آزمایش های دیگری نیز انجام دادیم که برای استخراج ویژگی از تصویر از شبکه ی resNet152 و برای استخراج ویژگی از سوال از LSTM ، BiLSTM و CNN های یک بعدی استفاده کرده‌ایم. با توجه به جدول‌های ۲ و ۴ استفاده از resNet152 به جای VGG19 منجر به افزایش دقت شده است. نکته‌ی دیگر این است که استفاده از BiLSTM در هر دو حالت نرم و سخت باعث کاهش دقت شده است. بنابراین بهترین مدلی که در این روش بدست می‌آید زمانی است که از ترجمه‌های Google و از شبکه‌ی resNet152 و LSTM استفاده کنیم که دقت 53.58 را می‌دهد.

	Google Translation			
Method	**yes/no**	**Number**	**Other**	**All**
BilstmQ+resNet152(hard)	76.46	31.63	38.6	51.89
lstmQ+resNet152(hard)	76.83	31.75	38.77	52.13
CNNQ+resNet152(hard)	78.34	31.91	38.98	52.82
BilstmQ+resNet152(soft)	78.22	33	39.89	53.37
lstmQ+resNet152(soft)	78.5	31.76	40.4	53.58
CNNQ+resNet152(soft)	78.38	32.36	38.99	52.9

جدول ۴: دقت روش baseline با استفاده از ویژگی‌های استخراج شده از شبکه ResNet۱۵۲ .

۴.۴ Stacked Attention Network

این روش را به دو صورت آزمایش کرده‌ایم. در حالت اول برای استخراج ویژگی از سوال، از دو لایه LSTM ۱۰۲۴ تایی با recurrent dropout با نرخ 0.5 استفاده می‌کنیم. بعد از هر لایه‌ی LSTM یک لایه‌ی BatchNormalization قرار می‌دهیم. در حالت دوم برای استخراج ویژگی از سوال، از روش CNN های یک بعدی با فیلتر سایزهای ۱، ۲ و ۳ که به ترتیب تعداد فیلترها برای هر کدام ۲۵۶، ۲۵۶ و ۵۱۲ است؛ استفاده می‌کنیم. در هر دو حالت آزمایش از دو لایه‌ی attention به ابعاد ۱۰۲۴استفاده می‌کنیم. برای پیاده‌سازی این روش از Tensorflow،keras استفاده کرده‌ایم. از Adam به عنوان بهینه‌ساز با نرخ یادگیری 0.0005 استفاده کردیم. سایز batch را برای همه‌ی آزمایش‌های این بخش ۳۰۰ قرار دادیم. شبکه را با حداکثر ۵۰ گام به همراه Early stopping آموزش می‌دهیم تا زمانی که خطا روی داده‌های ارزیابی در ۳ گام آخر تغییر نکند.

نتایج حاصل از این شبکه را در جدول ۵ می‌توانید مشاهده کنید. همانطور که پیداست به طور کلی دقت برای حالتی که از ترجمه‌های Google استفاده می‌کنیم بیشتر است. زمانی که از ترجمه‌های Google استفاده می‌کنیم، روش LSTM دقت بالاتری دارد اما زمانی که از ترجمه‌های ترگمان استفاده می‌کنیم دقت روش CNN بیشتر است. نکته‌ی حائز اهمیت در اینجا این است

که دقت بین دو حالت LSTM و CNN چندان تفاوتی ندارد اما از لحاظ منابع محاسباتی روش CNN به صرفه‌تر است زیرا تعداد پارامترها در روش CNN تقریبا ۸ میلیون و در روش LSTM ۲۲ میلیون می‌باشد.

	Google Translation				Targoman Translation			
Method	**yes/no**	**Number**	**Other**	**All**	**yes/no**	**Number**	**Other**	**All**
SAN_LSTM_2	77.83	33.19	39.08	52.84	75.95	31.61	36.82	50.81
SAN_CNN_2	77.49	33.17	39.18	52.76	76.48	32.29	37.37	51.37

جدول ۵: دقت روش Stacked Attention Network.

برای بررسی تاثیر تعداد لایه‌های attention ، مدل را در سه حالت که تعداد لایه‌های attention ۱، ۲ و ۳ باشد؛ آموزش می‌دهیم. با توجه به جدول ۶ دقت وقتی تعداد لایه‌های attention ۲ است بیشتر است. این نشان دهنده‌ی این است که ما برای بدست آوردن پاسخ نیاز به استدلال چند مرحله‌ای داریم. به همین خاطر یک لایه‌ی attention کافی نیست. از طرفی اگر تعداد لایه‌ها بیشتر از حدی باشد منجر به پاسخ اشتباه می‌شود. در اینجا زمانی که تعداد لایه‌ها را بیشتر از ۲ قرار دهیم منجر به کاهش عملکرد مدل می‌شود.

	Google Translation			
Method	**yes/no**	**Number**	**Other**	**All**
SAN_LSTM_1	77.46	32.23	38.35	52.22
SAN_LSTM_2	77.83	33.19	39.08	52.84
SAN_LSTM_3	77.12	32.56	38.62	52.27

جدول ۶: بررسی تاثیر تعداد لایه‌های attention در روش Stacked Attention Network.

۵.۴ HieCoAttention

برای پیاده‌سازی این روش از Tensorflow.keras استفاده کرده‌ایم. از Adam به عنوان بهینه‌ساز با نرخ یادگیری 0.0005 استفاده کردیم. سایز batch را برای همه‌ی آزمایش‌های این بخش ۳۰۰ قرار دادیم. شبکه را با حداکثر ۵۰ گام به همراه Early stopping آموزش می‌دهیم تا زمانی که خطا روی داده‌های ارزیابی در ۳ گام آخر تغییر نکند. ابعاد لایه Embedding و لایه‌های پنهان را ۵۱۲ قرار دادیم. نرخ dropout را 0.5 تنظیم کرده‌ایم.

نتایج حاصل از این شبکه را در جدول ۷ می‌توانید مشاهده کنید. انتظار ما این بود که بهترین نتایج برای این شبکه باشد اما تنظیم هایپرپارامترها در این شبکه اهمیت زیادی در دقت نهایی دارد. همچنین همگرایی این مدل به کندی اتفاق می‌افتد و زمان آموزش آن بسیار زیاد است. به همین دلیل ما زمان و منابع محاسباتی کافی برای اجرا درست این شبکه را نداشته‌ایم. با این حال دقت این مدل برای ترجمه‌های Google برابر با 51.85 و برای ترگمان برابر با 48.07 است.

	Google Translation				Targoman Translation			
Method	**yes/no**	**Number**	**Other**	**All**	**yes/no**	**Number**	**Other**	**All**
CoAttention	76.62	32.7	38.12	51.85	74.18	32.41	32.47	48.07

جدول ۷: دقت روش HieCoAttention.

دقت تمامی مدل های اجرا شده بر روی مجموعه‌داده فارسی در حالت سخت را در جدول ۸ آورده‌ایم. بهترین مدل برای Google مدل SAN_LSTM_2 با دقت 52.84 است و برای ترگمان مدل SAN_CNN_2 با دقت 51.37 است.

	Google Translation				Targoman Translation			
Method	**yes/no**	**Number**	**Other**	**All**	**yes/no**	**Number**	**Other**	**All**
lstm Q + VGG19	76.14	32.97	35.78	50.53	75.58	32.61	33.53	49.15
BilstmQ+resNet152	76.46	31.63	38.6	51.89	-	-	-	-
lstmQ+resNet152	76.83	31.75	38.77	52.13	-	-	-	-
CNNQ+resNet152	78.34	31.91	38.98	52.82	-	-	-	-
SAN_LSTM_2	77.83	33.19	39.08	**52.84**	75.95	31.61	36.82	50.81
SAN_CNN_2	77.49	33.17	39.18	52.76	76.48	32.29	37.37	**51.37**
CoAttention	76.62	32.7	38.12	51.85	74.18	32.41	32.47	48.07

جدول ۸: دقت کلی در حالت سخت

۵ پیاده‌سازی مدل بر بستر وب

ما حصل هر فکر، ایده و پژوهشی رسیدن به دانش و یا محصولی است که بتواند به نوع بشر کمک کند تا راحت‌تر با مشکلاتش دست و پنجه نرم کند. مدل ما نیز از این قاعده مستثنی نیست. به همین دلیل، بهترین مدلی که در این پروژه بدست آوردیم را در بر بستر فضای ابری برای عموم به اشتراک گذاشته‌ایم (مشاهده دمو). در حال حاضر مدل ما به سوال پرسیده شده از یک تصویر حداکثر در ۱۰ ثانیه پاسخ می‌دهد.

مراجع

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.

[2] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[3] D. Gurari, Q. Li, A. J. Stangl, A. Guo, C. Lin, K. Grauman, J. Luo, and J. P. Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3608–3617, 2018.

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[5] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[6] I. Ilievski and J. Feng. A simple loss function for improving the convergence and accuracy of visual question answering models. *arXiv preprint arXiv:1708.00584*, 2017.

[7] J.-H. Kim, J. Jun, and B.-T. Zhang. Bilinear attention networks. In *Advances in Neural Information Processing Systems*, pages 1564–1574, 2018.

[8] J. Lu, J. Yang, D. Batra, and D. Parikh. Hierarchical question-image co-attention for visual question answering. In *Advances in neural information processing systems*, pages 289–297, 2016.

[9] D.-K. Nguyen and T. Okatani. Improved fusion of visual and language representations by dense symmetric co-attention for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6087–6096, 2018.

[10] Y. Shi, T. Furlanello, S. Zha, and A. Anandkumar. Question type guided attention in visual question answering. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 151–166, 2018.

[11] K. J. Shih, S. Singh, and D. Hoiem. Where to look: Focus regions for visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4613–4621, 2016.

[12] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[13] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[14] Q. Wu, D. Teney, P. Wang, C. Shen, A. Dick, and A. van den Hengel. Visual question answering: A survey of methods and datasets. *Computer Vision and Image Understanding*, 163:21–40, 2017.

[15] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 21–29, 2016.