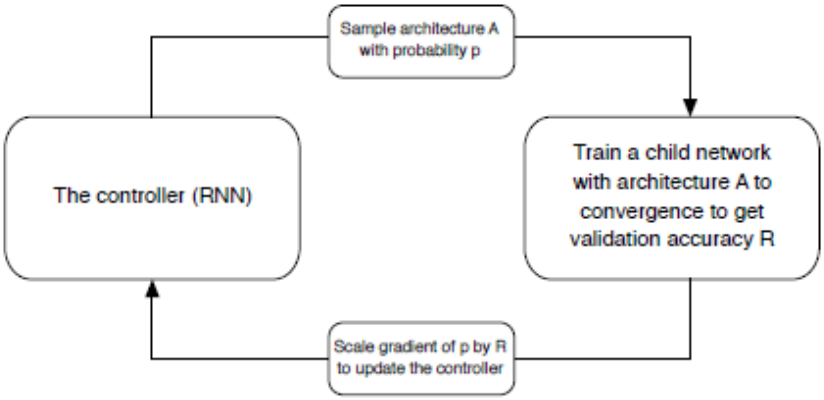
-1

روش پیشنهادی از روش‌های جستجو برای یافتن معماری های کانولوشن خوب بر روی یک مجموعه داده‌ی مورد نظر استفاده می‌کند. این روش با الهام از چارچوب جستجوی معماری عصبی (NAS) که از یک روش جستجوی یادگیری تقویتی (reinforcement learning) برای بهینه سازی تنظیمات معماری استفاده می کند. در NAS ، یک کنترل‌کننده شبکه عصبی بازگشتی (RNN) از شبکه‌های فرزند با معماری‌های مختلف نمونه‌برداری می کند. شبکه‌های فرزند برای همگرایی به مقداری دقت بر روی مجموعه داده‌ی اعتبارسنجی آموزش داده شده اند. از دقت بدست آمده برای به‌روزرسانی کنترل‌کننده استفاده می شود تا با گذشت زمان کنترل‌کننده معماری های بهتری ایجاد کند. وزن‌های کنترل کننده با سیاست گرادیان به‌روز می شوند (شکل1).

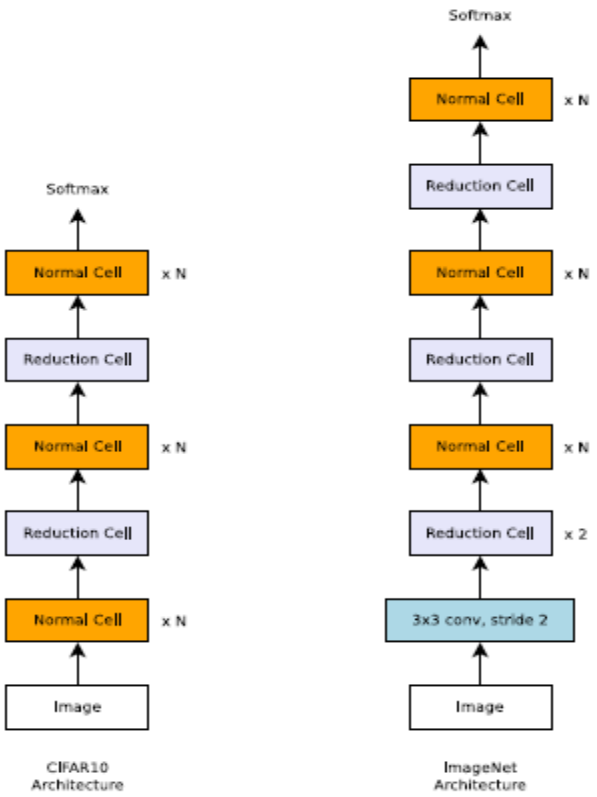


شکل 1. بررسی اجمالی معماری جستجوی عصبی (NAS) . یک کنترل‌کننده RNN معماری A را از یک فضای جستجو با احتمال p پیش بینی می‌کند. یک شبکه فرزند با معماری A برای دستیابی به دقت R آموزش دیده‌است. گرادیان p را با R مقیاس کنید تا کنترل‌کنندهRNN به روز شود.

یک انگیزه برای فضای جستجوی NASNet ، درک این نکته است که مهندسی معماری با CNNها اغلب الگوهای تکراری متشکل از ترکیب بانک‌های فیلترکانولوشنی ، توابع غیرخطی و یک انتخاب محتاطانه از اتصالات را برای دستیابی به نتایج لبه‌ی دانش (مانند ماژول های تکرار شده در مدلهای Inception و ResNet) شناسایی می‌شود. این مشاهدات نشان می‌دهد که برای کنترل کننده RNN پیش بینی یک سلول کانولوشن عمومی بیان شده با توجه به این الگوها ممکن است. سپس می توان این سلول را به صورت سری انباشت تا ورودی‌هایی از ابعاد فضایی و عمق فیلتر دلخواه را کنترل کند. در این رویکرد معماری کلی شبکه‌های کانولوشن به صورت دستی از پیش تعیین شده است. آنها از سلول‌های کانولوشن که بارها تکرار شده‌اند ساخته

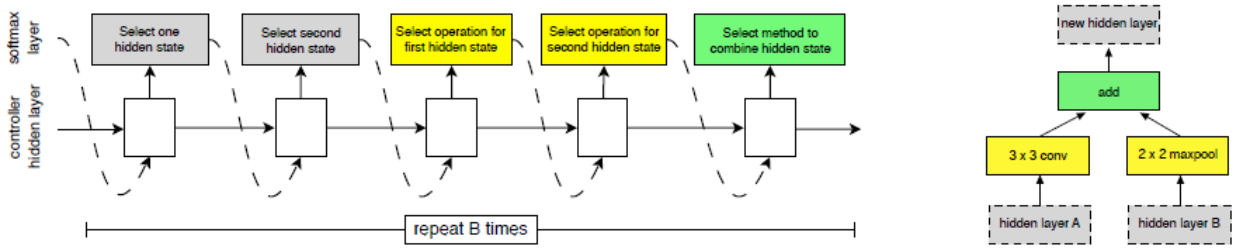
شده‌اند در حالی که هر سلول کانولوشن دارای معماری یکسان ، اما وزن متفاوت است. برای ساخت آسان معماری مقیاس‌پذیر برای تصاویر از هر اندازه ، ما به دو نوع سلول کانولوشن برای ارائه دو عملکرد اصلی هنگام گرفتن نقشه ویژگی (feature map) به عنوان ورودی نیاز داریم: (1) سلول‌های کانولوشن که نقشه مشخصه‌ای از همان بعد را برمی گردانند ، و (2) سلولهای کانولوشن که نقشه مشخصه‌ای را برمی‌گردانند که در آن طول و عرض نقشه ویژگی با ضریب دو کاهش می یابد. سلول‌های نوع اول و دوم کانولوشنی را به ترتیب سلول نرمال (Normal Cell) و سلول کاهشی (Reduction Cell) می‌نامیم. برای سلول کاهشی ، می‌توانیم بر روی عملکرد اولیه اعمال شده بر ورودی های سلول stride = 2 را به منظور کاهش ارتفاع و عرض انجام دهیم. تمام عملیاتی که برای ساخت سلول‌های کانولوشن در نظر می‌گیریم دارای گزینه stride کردن هستند.

شکل 2 محل قرارگیری سلولهای نرمال و کاهشی برای CIFAR-10 و ImageNet را نشان می دهد. در ImageNet ما سلول‌های کاهشی بیشتری داریم ، زیرا اندازه تصویر ورودی 299 x299 در مقایسه با 32x32 برای CIFAR است.



شکل 2. معماری‌های مقیاس‌پذیر برای طبقه‌بندی تصویر از دو الگوی تکراری با نام سلول نرمال و سلول کاهشی تشکیل شده‌است.

سلول نرمال و سلول کاهشی می‌توانند از معماری یکسانی برخوردار باشند ، اما از نظر تجربی یادگیری دو معماری جداگانه مفید خواهد بود. هر زمان که اندازه فعال سازی فضایی کاهش می یابد ، از یک اکتشاف معمولی برای دو برابر کردن تعداد فیلترهای خروجی استفاده می کنیم تا ابعاد حالت پنهان (hidden state) را تقریباً ثابت حفظ کنیم. دقیقاً مانند مدل های Inception و ResNet ، ما تعداد تکرارهای الگوها N و تعداد فیلترهای کانولوشن اولیه را به عنوان پارامترهای آزاد در نظر می گیریم که متناسب با مقیاس مسئله طبقه‌بندی تصویر تنظیم می شوند. آنچه در شبکه های کانولوشن متفاوت است ساختارهای سلول‌های نرمال و کاهشی است که توسط کنترل کننده RNN جستجو می شود. ساختارهای سلول‌ها را می توان در یک فضای جستجو جستجو کرد. در فضای جستجوی ما ، هر سلول به عنوان ورودی دو حالت اولیه پنهان h_i و h_{i-1} را دریافت می کند که خروجی دو سلول در دو لایه پایین قبلی یا تصویر ورودی است. کنترل کننده RNN با توجه به این دو حالت پنهان اولیه ، بقیه ساختار سلول کانولوشن را بصورت بازگشتی پیش‌بینی می‌کند (شکل 3).



شکل 3. معماری مدل کنترل کننده برای ساخت بازگشتی یک بلوک از سلول کانولوشن. هر بلوک نیاز به انتخاب 5 پارامتر گسسته دارد که هر یک از آنها مربوط به خروجی یک لایه softmax است. نمونه بلوک ساخته شده در سمت راست نشان داده شده است. یک سلول کانولوشن شامل B بلوک است ، از این رو کنترل کننده شامل 5B لایه softmax برای پیش بینی ساختار سلول کانولوشن است.

پیش بینی های کنترل کننده برای هر سلول در B بلوک دسته بندی می شوند ، جایی که هر بلوک دارای 5 مرحله پیش بینی است که توسط 5 طبقه بند متمایز softmax متناظر با انتخاب‌های گسسته عناصر یک بلوک انجام می‌شود:

مرحله 1. یک حالت پنهان از h_i، h_{i-1} یا از مجموعه حالت‌های پنهان ایجاد شده در بلوک‌های قبلی ، انتخاب کنید.

مرحله 2. حالت پنهان دوم را از همان گزینه‌های مرحله 1 انتخاب کنید.

مرحله 3. عملیاتی را انتخاب کنید که بر روی حالت پنهان انتخاب شده در مرحله 1 اعمال شود.

مرحله 4. عملیاتی را انتخاب کنید که بر روی حالت پنهان انتخاب شده در مرحله 2 اعمال شود.

مرحله 5. برای ایجاد یک حالت پنهان جدید ، روشی را برای ترکیب خروجی های مرحله 3 و 4 انتخاب کنید.

الگوریتم حالت پنهان ایجاد شده جدید را به مجموعه‌ای از حالت‌های پنهان موجود به عنوان ورودی بالقوه در بلوک‌های بعدی اضافه می‌کند. کنترل‌کننده RNN 5 مرحله پیش بینی بالا را B بار متناظر با B بلوک در سلول کانولوشن تکرار می کند. در مراحل 3 و 4 ، کنترل کننده RNN عملیاتی را برای اعمال بر روی حالت‌های پنهان انتخاب می کند. مجموعه‌ای از این عملیات بر اساس شیوع آنها در ادبیات CNN به صورت زیر است:

- identity
- 1x3 then 3x1 convolution
- 1x7 then 7x1 convolution
- 3x3 dilated convolution
- 3x3 average pooling
- 3x3 max pooling
- 5x5 max pooling
- 7x7 max pooling
- 1x1 convolution
- 3x3 convolution
- 3x3 depth wise-separable conv
- 5x5 depth wise-separable conv
- 7x7 depth wise-separable conv

در مرحله 5 کنترل کننده RNN روشی را برای ترکیب دو حالت پنهان انتخاب می‌کند ، (1) جمع بین عناصر دو حالت پنهان یا (2) اتصال بین دو حالت پنهان در امتداد بعد فیلتر. سرانجام ، تمام حالت‌های پنهان بلااستفاده تولید شده در سلول کانولوشن در عمق با هم ادغام می شوند تا خروجی سلول نهایی فراهم شود. برای اینکه اجازه دهیم کنترلر RNN هم سلول نرمال و هم سلول کاهشی را پیش بینی کند ،کنترل کننده را طوری می‌سازیم تا در کل 2x5B پیش‌بینی کند، جایی که اولین 5B پیش‌بینی مربوط به سلول نرمال و 5B پیش بینی دوم مربوط به سلول کاهشی است.

سرانجام ، از یادگیری تقویتی در NAS استفاده می‌کنیم. با این وجود ، می توان از جستجوی تصادفی برای جستجوی معماری در فضای جستجوی NASNet نیز استفاده کرد. در جستجوی تصادفی ، به جای نمونه‌برداری تصمیمات از طبقه‌بند softmax در کنترلر RNN ، می توانیم تصمیمات را از توزیع یکنواخت نمونه برداری کنیم.

این روش معماری‌های مدل را به صورت مستقیم از روی مجموعه داده‌ی مورد نظر یاد می‌گیرد. این رویکرد در وقتی مجموعه داده بزرگ باشد پر هزینه است ، بنابراین پیشنهاد می‌شود یک بلوک ساختمان معماری را بر روی یک مجموعه داده کوچک جستجو کنیم و سپس آن را به یک مجموعه داده بزرگتر منتقل کنیم. مهم‌ترین کار طراحی فضای جستجو جدید است ("فضای جستجوی NASNet") که قابلیت انتقال را فراهم می‌کند تا پیچیدگی معماری مستقل از عمق شبکه و اندازه تصاویر ورودی باشد. به طور دقیق تر ، تمام شبکه های کانولوشن در فضای جستجوی ما از لایه های کانولوشن (یا "سلول") با ساختار یکسان اما وزن متفاوت تشکیل شده‌اند. بنابراین جستجوی بهترین معماری های کانولوشن به جستجوی بهترین ساختار سلول کاهش می‌یابد.

جستجوی بهترین ساختار سلول دو مزیت اصلی دارد: بسیار سریعتر از جستجوی کل ساختار شبکه است و خود سلول احتمالاً به سایر مسائل تعمیم می‌یابد. علاوه بر این ، با تغییر ساده تعداد سلول های کانولوشن و تعداد فیلترها در سلول های کانولوشن ، می‌توانیم نسخه های مختلف NASNetها را با نیاز محاسباتی متفاوت ایجاد کنیم. به لطف این ویژگی سلول‌ها ، ما می‌توانیم خانواده‌ای از مدل‌ها را تولید کنیم که به دقت بالاتری نسبت به همه مدل های طراحی شده توسط بشر با هزینه‌های محاسباتی معادل یا کوچکتر برسند. سرانجام ، ویژگی های تصویری که از طبقه بندی تصویر (image classification) آموخته اند ، بسیار مفید هستند و می توانند به سایر مشکلات بینایی کامپیوتر مثل تشخیص اشیاء (object detection) منتقل شوند.

-2

$$\mu_j = \frac{1}{N}\sum_{i=1}^{N} x_{ij}$$

$$\sigma_j^2 = \frac{1}{N}\sum_{i=1}^{N}\left(x_{ij} - \mu_j\right)^2$$

$$\hat{x}_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j}$$

$$y_{ij} = \gamma_j \hat{x}_{ij} + \beta_j$$

اگر خروجی قبل از batch normalization را h بنامیم داریم:

h = w1x1 + w2x2 + b

h11 = -0.4 * 121 + -0.3 * 16.8 = -53.44

h12 = 1.2 * 121 + 0.3 * 16.8 = 150.24

h21 = -0.4 * 114 + -0.3 * 15.2 = -50.16

h22 = 1.2 * 114 + 0.3 * 15.2 = 141.36

h31 = -0.4 * 210 + -0.3 * 9.4 = -86.82

h32 = 1.2 * 210 + 0.3 * 9.4 = 254.82

h41 = -0.4 * 195 + -0.3 * 8.1 = -80.43

h42 = 1.2 * 195 + 0.3 * 8.1 = 236.43

μ_1 = (-53.44 - 50.16 - 86.82 - 80.43) / 4 = -67.7125

σ_1^2 = ((-53.44+ 67.7125)^2 + (-50.16+ 67.7125)^2 + (-86.82 + 67.7125)^2 + (-80.43+ 67.7125)^2) / 4 = (203.7 + 308.1 + 365.1 + 161.7) / 4 = 259.65

σ_1 = 16.11

$\hat{h}_{11}$ = (-53.44 +67.7125) / 16.11 = 0.88

$\hat{h}_{21}$ = (-50.16 +67.7125) / 16.11 = 1.09

$\hat{h}_{31}$ = (-86.82 +67.7125) / 16.11 = -1.18

$\hat{h}_{41}$ = (-80.43 +67.7125) / 16.11 = -0.79

$\mu_2 = (150.24 + 141.36 + 254.82 + 236.43) / 4 = 195.7125$

$\sigma_2^2 = ((150.24- 195.7125)^2 + (141.36– 195.7125)^2 + (254.82 - 195.7125)^2 + (236.43 – 195.7125)^2) / 4 = (2067.75 + 2954.2 + 3493.7 + 1657.9) / 4 = 2543.38 = 50.43$

$\hat{h}_{12} = (150.24 - 195.7125) / 50.43 = -0.9$

$\hat{h}_{22} = (141.36 - 195.7125) / 50.43 = -1.08$

$\hat{h}_{32} = (254.82 - 195.7125) / 50.43 = 1.17$

$\hat{h}_{42} = (236.43- 195.7125) / 50.43 = 0.8$

$z_{11} = 0.5 \times 0.88+ 0.5 = 0.94$

$z_{21} = 0.5 \times 1.09 + 0.5 = 1.045$

$z_{31} = 0.5 \times (-1.18) + 0.5 = -0.09$

$z_{41} = 0.5 \times (-0.79) + 0.5 = 0.105$

$z_{12} = 0.5 \times (-0.9)+ 0.5 = 0.05$

$z_{22} = 0.5 \times (-1.08) + 0.5 = -0.04$

$z_{32} = 0.5 \times 1.17+ 0.5 = 1.08$

$z_{42} = 0.5 \times 0.8 + 0.5 = 0.9$

$\alpha_{11} = \max(0,0.94) = 0.94$

$\alpha_{21} = \max(0,1.045) = 1.045$

$\alpha_{31} = \max(0,-0.09) = 0$

$\alpha_{41} = \max(0,0.105) = 0.105$

$\alpha_{12} = \max(0,0.05) = 0.05$

$\alpha_{22} = \max(0,-0.04) = 0$

$\alpha_{32} = \max(0,1.08) = 1.08$

$\alpha_{42} = \max(0,0.9) = 0.9$