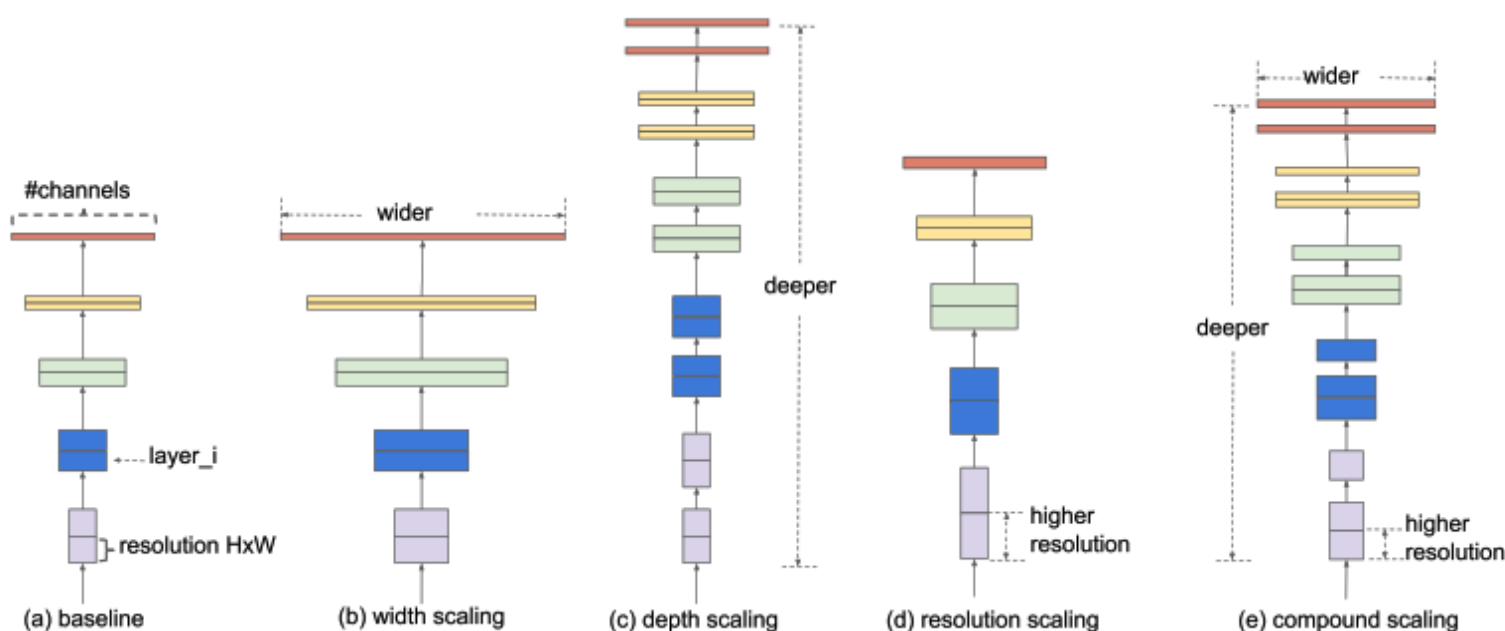


سوال 1:

بسیاری از معماری‌های طراحی شده مانند MobileNet به منظور استفاده بهینه از توان محاسباتی موجود، می‌باشند.

به طور کلی می‌توان روش‌های استفاده شده در شبکه‌های CNN، به منظور افزایش دقت را به سه دسته افزایش عمق شبکه، افزایش ارتفاع شبکه و افزایش رزولوشن عکس ورودی شبکه افراز کرد.

در واقع، مقاله Efficient-Net، به معرفی معماری جدیدی پرداخته است. بلکه به معرفی روش‌های بهینه افزایش دقت در نتیجه Scaling شبکه با توجه به توان محاسباتی و پردازشی هر دستگاهی پرداخته است.



با توجه به شکل بالا، با در نظر گرفتن يك baseline (شکل a) به صورت کلی سه روش برای Scaling شبکه برای به دست آوردن دقت بهینه وجود دارد.

سه روش برای Scale-up یک CNN به صورت زیر می‌باشد، طبق بحث‌های ابتدا، شامل افزایش عمق، افزایش عرض، افزایش رزولوشن عکس ورودی شبکه می‌باشد که به توضیح هر کدام از آنها می‌پردازیم:

- روش اول افزایش عمق: منظور از افزایش عمق، افزایش تعداد لایه های يك شبکه مي باشد.

- روش دوم افزایش عرض: منظور از عرض نیز مقدار کانال های يك شبکه مي باشد.

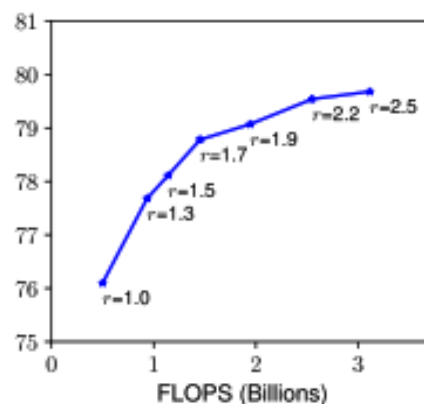
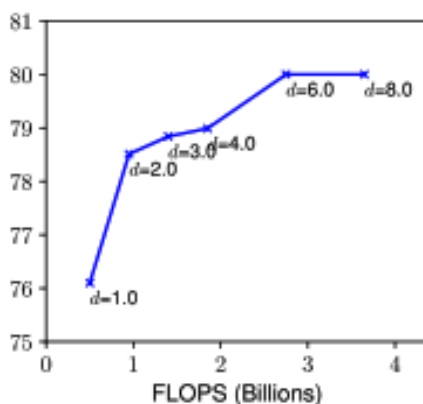
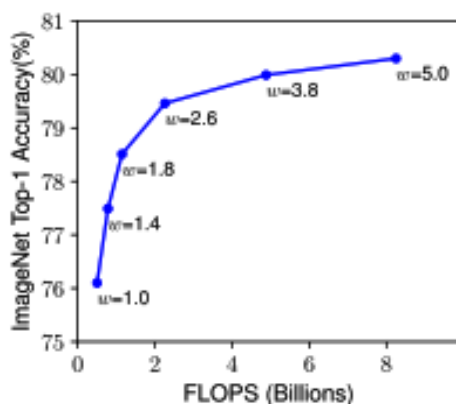
- روش سوم افزایش رزولوشن عكس ورودی: از این روش به ندرت در روش های Scale-up استفاده می شود.

روشی که این مقاله پیشنهاد می دهد، منجر به افزایش همزمان عمق، عرض و رزولوشن عكس ورودی می شود که به صورت بهینه به افزایش دقت شبکه کمک شایانی می کند. نام این روش مقیاس بندی مدل مرکب یا **compound model scaling** می باشد.

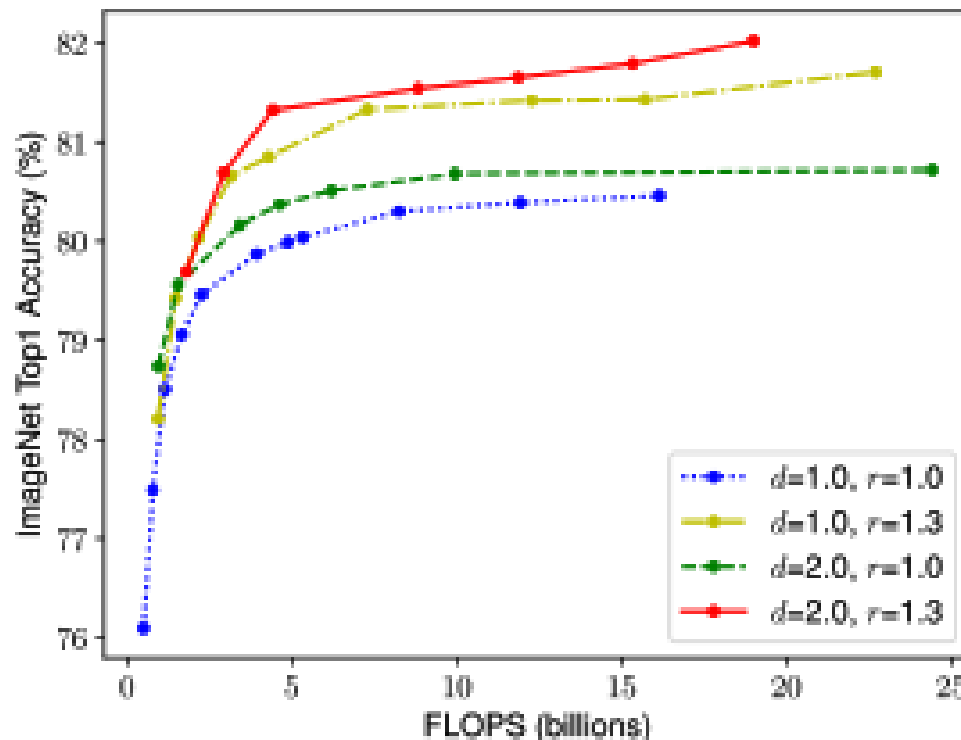
مقیاس بندی مدل مرکب

عموما، برای افزایش دقت، افراد به سراغ طراحی یک معماری جدید از شبکه های عصبی می روند. اما در این مقاله، روشی معرفی شده است که ابتدا یک مدل baseline در نظر گرفته می شود و سپس سعی می کند روش های Scaling در مدل baseline بهبود ایجاد کند. مسئله اصلی این کار فضای جستجویی است که باید برای یافتن بهینه ترین ضرایب Scaling، مورد بررسی قرار گیرد. برای ساده سازی و در نهایت کوچک کردن فضای جستجو، فرض می کنیم که در هر لایه ضرایب Scaling هر بعد، به صورت ثابت و یکنواخت رشد می کنند.

يکي از مشکلاتي که Scale-up مي تواند ایجاد کند، مشکل vanishing gradient شود که با skip connection قابل حل مي باشد.



طبق شکل بالا، افزایش هر يك از این ابعاد باعث افزایش دقت می‌شود، اما با رسیدن به مقدار 80%، دقت، اشباع می‌شود و افزایشی صورت نمی‌پذیرد.



طبق مشاهدات انجام شده، می‌توان با ترکیب این ابعاد با یکدیگر به میزان دقت بیشتری دست پیدا نمود. یا به بیان دیگر، این ابعاد از یکدیگر مستقل نبوده و با یکدیگر همبستگی دارند.

در ادامه به معرفی روش بهینه سازی معرفی شده در این مقاله به نام compound scaling method می‌پردازیم که به صورت زیر فرموله شده است. بدین صورت که α ، β و γ ، ثابت هایی در نظر گرفته می‌شوند که از طریق روش جستجوی حریصانه به دست می‌آیند. حال اگر بتوانیم با یک ضریب ثابت، هزینه محاسباتی را افزایش دهیم، مقدار ϕ را تغییر می‌دهیم.

$$\text{Depth: } d = \alpha^{\phi}$$

Width: $w = \beta^\phi$

Resolution: $r = \gamma^\phi$

s.t. $\alpha.\beta^2.\gamma^2 = 2$

$\alpha \geq 1, \beta \geq 1, \gamma \geq 1$

Phi هم با توجه به مقدار منابع در دسترس محاسباتی تغییر داده شود. میزان توان محاسباتی با FLOPS سنجیده می شود که در واقع نشان دهنده ی میزان تعداد عملیات محاسباتی شناور که در يك ثانیه قابل انجام است، می باشد. محدودیت $\alpha.\beta^2.\gamma^2 = 2$ به این معنی می باشد که میزان FLOPS ها به میزان ϕ^2 می تواند افزایش یابد.

معماري شبکه

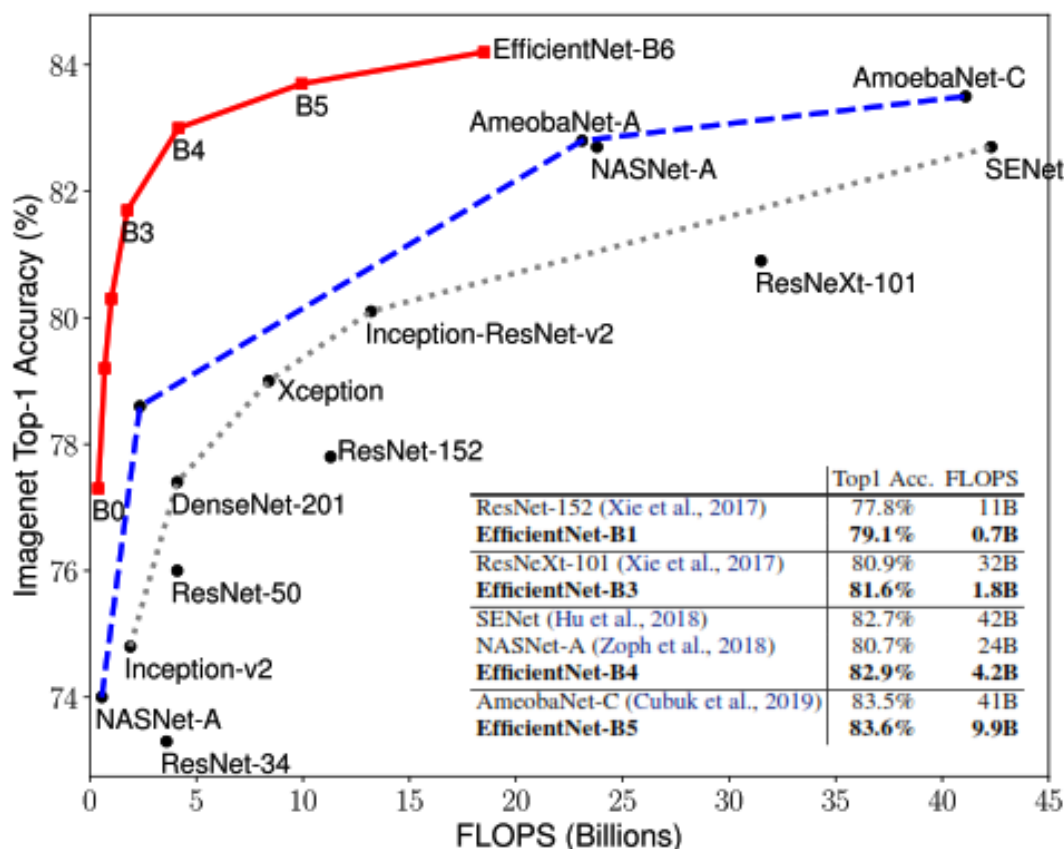
یکی از مهم ترین نکاتی که باید به آن توجه شود، انتخاب یک شبکه مناسب به عنوان baseline می باشد. همانطور که در بالاتر اشاره شد، efficient Net، یک معماری جدید نمی باشد. بلکه یک روش بهینه برای Scale up ابعاد مختلف شبکه می باشد. پس انتخاب یک شبکه مناسب baseline، باعث بهبود فوق العاده این بهینه سازی می شود.

efficient Net دارای هفت ورژن مختلف می باشد. که ابتدا B0 را با استفاده از روش و معماری معرفی شده به دست آورده شده و آن را به عنوان baseline در نظر گرفتند. سپس با دو گام ورژن هایی دیگر (B1-B7) را به دست می آمدند:

- گام اول: با در نظر گرفتن شرط ذکر شده $\alpha.\beta^2.\gamma^2 = 2$ ، ابتدا ϕ را ثابت و برابر يك در نظر میگیریم، فرض می کنیم که دو برابر توان محاسباتی در اختیار داریم و با استفاده از يك β ، α ، γ و grid search مناسب را پیدا میکنیم.
- گام دوم: در این قسمت آلفا، بتا و گاما را ثابت در نظر گرفته و با در نظر گرفتن ϕ های مختلف baseline یعنی B0 را scale up می نماییم.

نتایج

طبق نمودار زیر، دقت شبکه efficient Net نسبت به بقیه مدل ها بیشتر می باشد.



طبق این نمودار، همه ورژن های efficient Net با توجه به مصرف منابع محاسباتی کمتر نسبت به دیگر مدل ها، از دقت بالاتری برخوردار هستند. efficient Net B7 به دقت 84.3% که state-of-the-art می باشد رسیده است و 8.4x کوچک تر و 6.1x سریع تر می باشد. همچنین، این مدل بر روی دیتاست های مختلف هم اعمال شده که نتایج آن در جدول زیر قابل رویت می باشد.

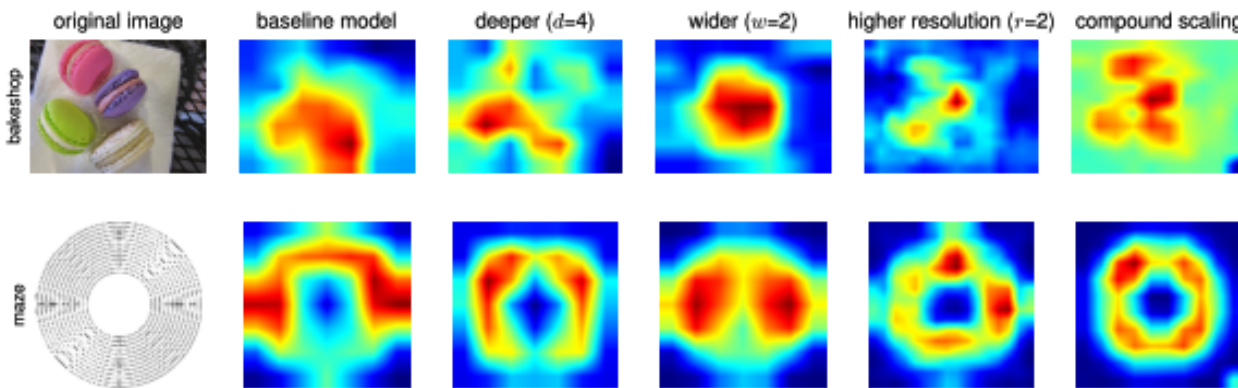
Model	Top-1 Acc.	Top-5 Acc.	#Params	Ratio-to-EfficientNet	#FLOPs	Ratio-to-EfficientNet
EfficientNet-B0	77.1%	93.3%	5.3M	1x	0.39B	1x
ResNet-50 (He et al., 2016)	76.0%	93.0%	26M	4.9x	4.1B	11x
DenseNet-169 (Huang et al., 2017)	76.2%	93.2%	14M	2.6x	3.5B	8.9x
EfficientNet-B1	79.1%	94.4%	7.8M	1x	0.70B	1x
ResNet-152 (He et al., 2016)	77.8%	93.8%	60M	7.6x	11B	16x
DenseNet-264 (Huang et al., 2017)	77.9%	93.9%	34M	4.3x	6.0B	8.6x
Inception-v3 (Szegedy et al., 2016)	78.8%	94.4%	24M	3.0x	5.7B	8.1x
Xception (Chollet, 2017)	79.0%	94.5%	23M	3.0x	8.4B	12x
EfficientNet-B2	80.1%	94.9%	9.2M	1x	1.0B	1x
Inception-v4 (Szegedy et al., 2017)	80.0%	95.0%	48M	5.2x	13B	13x
Inception-resnet-v2 (Szegedy et al., 2017)	80.1%	95.1%	56M	6.1x	13B	13x
EfficientNet-B3	81.6%	95.7%	12M	1x	1.8B	1x
ResNeXt-101 (Xie et al., 2017)	80.9%	95.6%	84M	7.0x	32B	18x
PolyNet (Zhang et al., 2017)	81.3%	95.8%	92M	7.7x	35B	19x
EfficientNet-B4	82.9%	96.4%	19M	1x	4.2B	1x
SENet (Hu et al., 2018)	82.7%	96.2%	146M	7.7x	42B	10x
NASNet-A (Zoph et al., 2018)	82.7%	96.2%	89M	4.7x	24B	5.7x
AmoebaNet-A (Real et al., 2019)	82.8%	96.1%	87M	4.6x	23B	5.5x
PNASNet (Liu et al., 2018)	82.9%	96.2%	86M	4.5x	23B	6.0x
EfficientNet-B5	83.6%	96.7%	30M	1x	9.9B	1x
AmoebaNet-C (Cubuk et al., 2019)	83.5%	96.5%	155M	5.2x	41B	4.1x
EfficientNet-B6	84.0%	96.8%	43M	1x	19B	1x
EfficientNet-B7	84.3%	97.0%	66M	1x	37B	1x
GPipe (Huang et al., 2018)	84.3%	97.0%	557M	8.4x	-	-

We omit ensemble and multi-crop models (Hu et al., 2018), or models pretrained on 3.5B Instagram images (Mahajan et al., 2018).

برای اینکه به صورت مصور هم بتوانیم نتایج را مشاهده کنیم، به تصویر Activation Map که در مقاله وجود دارد و در پایین هم آورده شده است را می توان دقت کرد. همان طور که مشخص می باشد با اضافه نمودن هر کدام از ویژگی های عمق، عرض و رزولوشن نتایج تا حدودی بهبود یافته اند، و زمانی که از روش معرفی شده در این مقاله compound scaling برای scale کردن شبکه استفاده شده است مشاهده می شود که نسبت به سه روش قبلی

بهبود چشم گیری ایجاد شده است.

EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks

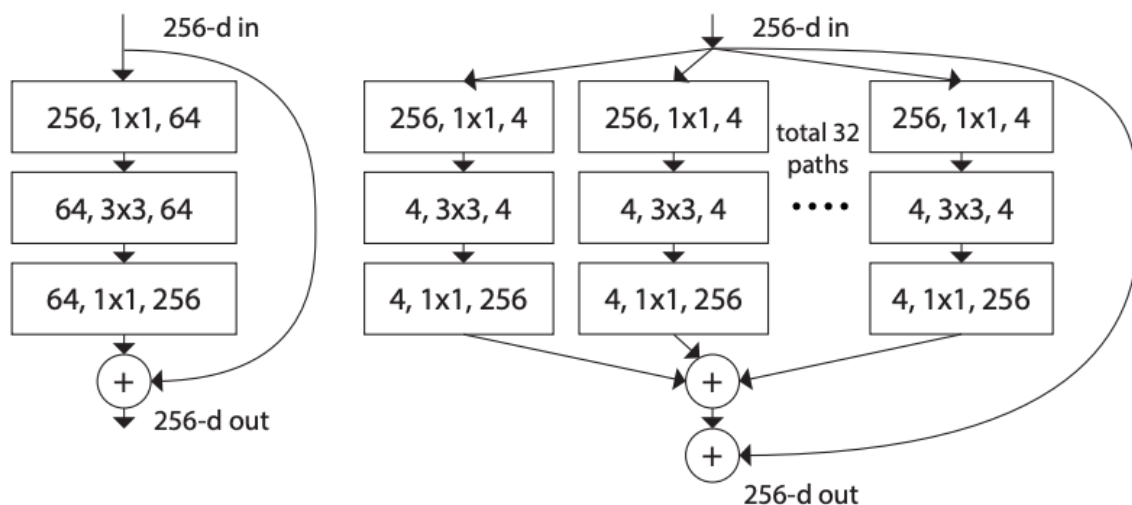


سوال 2:

ابتدا به معرفی دو شبکه ResNext و Inception-ResNet، می پردازیم:

:ResNext

یک شبکه عصبی همگن است که تعداد هاپیر پارامترهای مورد نیاز ResNet معمولی را کاهش می دهد. این امر با استفاده از "Cardinality" یا یک بعد اضافی در بالای عرض و عمق ResNet به دست می آید. در واقع کاردینالیتی به تعداد Transformer ها یا مسیر هایی که داده را از بخش ورودی به بخش خروجی مرتبط می سازد گفته می شود که در این جا و به عنوان مثال طبق تصویر زیر، عدد 32 می باشد.

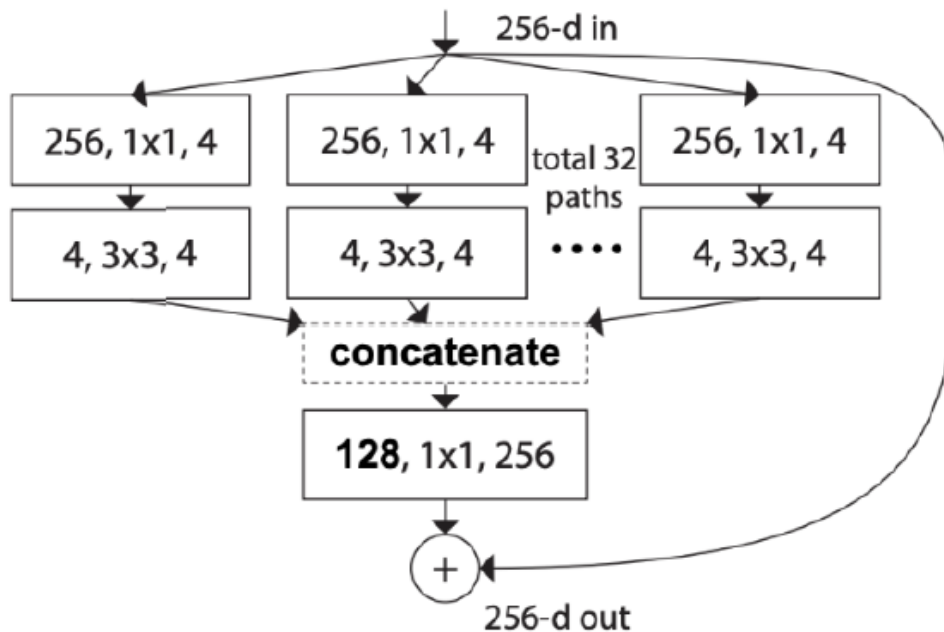


در تصویر بالا، بلاک سمت چپ یک ResNet Block و بلاک سمت راست یک ResNext Block با کاردینالیتی 32 می باشد. در این شبکه از ایده "استفاده هر بلاک از شبکه به عنوان نورون" موسوم به "Network in Neuron" استفاده شده است که عملاً به عنوان یک تابع غیر خطی عمل می کند. حال می توان این تابع که در بالا توضیح داده شد را به فرمت زیر توصیف کرد:

$$F(X) = \sum_{i=1}^C \tau(X)$$

:Inception-ResNet

در واقع این شبکه هم نوعی resnet می باشد که برای هدف دیگری نسبت به ResNext، طراحی شده است. طبق تصویر زیر، همانند شبکه ResNext، در شبکه Inception-ResNet هم مفهوم کاردینالیتی وجود دارد.



به طور کلی و با مقایسه بلاک های ResNext و Inception-ResNet، می توان پی برد که از مهم ترین اهدافی که شبکه های خانواده Inception دنبال می کنند، افزایش دقت به همراه کاهش هزینه های محاسباتی می باشد.

اما از طرف دیگر، شبکه های ResNext، تنها بر روی افزایش دقت محاسباتی با استفاده از افزایش ابعاد شبکه و عمیق تر کردن آن نسبت به Inception-ResNet ها تمرکز دارند.