NetScore: Towards Universal Metrics for Large-scale Performance Analysis of Deep Neural Networks for Practical On-Device Edge Usage

Anonymous Author(s)
Affiliation
Address
email

Abstract

Much of the focus in the design of deep neural networks had been on improving accuracy, leading to more powerful yet highly complex network architectures that are difficult to deploy in practical scenarios. As a result, there has been a recent interest in the design of quantitative metrics for evaluating deep neural networks that accounts for more than just model accuracy as the sole indicator of network performance. In this study, we continue the conversation towards universal metrics for evaluating the performance of deep neural networks for practical on-device edge usage by introducing **NetScore**, a new metric designed specifically to provide a quantitative assessment of the balance between accuracy, computational complexity, and network architecture complexity of a deep neural network. In what is one of the largest comparative analysis between deep neural networks in literature, the NetScore metric, the top-1 accuracy metric, and the popular information density metric were compared across a diverse set of 60 different deep convolutional neural networks for image classification on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC 2012) dataset. The evaluation results across these three metrics for this diverse set of networks are presented in this study to act as a reference guide for practitioners in the field.

1 Introduction

There has been a recent urge in both research and industrial interests in deep learning [4], with deep neural networks such as deep convolutional neural networks [6, 5] demonstrating state-of-the-art performance across a wide variety of applications [19, 22, 11]. However, the practical industrial deployment bottlenecks associated with the powerful yet highly complex deep neural networks in research literature has become even increasingly visible, and as a result, the design of deep neural networks that strike a strong balance between accuracy and complexity become a very hot area of research focus [18, 14, 34, 33, 26, 28, 36]. One of the key challenges in designing practical deep neural networks lies in the difficulties with assessing how well a particular network architecture is striking that balance. One of the most widely cited metrics is the information density metric proposed by [1], which attempts to measure the relative amount of accuracy given network size. However, information density does not account for computational requirements for performing network inference (e.g., MobileNet [14] has more parameters than SqueezeNet [18] but has lower computational requirements for network inference). Therefore, the exploration and investigation towards universal performance metrics that account for accuracy, architectural complexity, and computational complexity is highly desired as it has the potential to improve network model search and design. In this study, we introduce **NetScore**, a new metric designed specifically to provide a quantitative assessment of the balance between accuracy, computational complexity, and network architecture complexity of a deep neural network.

Submitted to 32nd Conference on Neural Information Processing Systems (NIPS 2018). Do not distribute.

2 NetScore: Design Principles

The proposed NetScore metric (denoted here as Ω) for assessing the performance of a deep neural network $\mathcal{N}$ for practical usage can be defined as:

$$\Omega(\mathcal{N}) = 20 \log \left(\frac{a(\mathcal{N})^{\alpha}}{p(\mathcal{N})^{\beta} m(\mathcal{N})^{\gamma}} \right) \quad (1)$$

where $a(\mathcal{N})$ is the accuracy of the network, $p(\mathcal{N})$ is the number of parameters in the network, $m(\mathcal{N})$ is the number of multiply–accumulate (MAC) operations performed during network inference, and α, β, γ are coefficients that control the influence of accuracy, architectural complexity, and computational complexity of the network on Ω.

Control coefficients We set $\alpha = 2$ to better emphasize the importance of model accuracy in assessing the overall performance of a network in practical usage, as networks that have unreasonably low accuracy remain unusable in practical scenarios, regardless how small or fast the network is. Furthermore, we set $\beta = 0.5$ and $\gamma = 0.5$ since, while architectural and computational complexity are both very important factors to assessing the overall performance of a network in practical scenarios, the most important metric remains the model accuracy given that, as eluded to before, networks with unreasonably low model accuracy are not useful in practical scenarios regardless of size and speed. Given these coefficients, NetScore is in the units of squared percentage accuracy per root parameter per root MAC operation, and represents the capacity of a network architecture to utilize its full learning and computing capacity.

Logarithmic scaling: A difficulty in comparing the overall performance of different deep neural networks with each other is their great diversity in their model accuracy, architectural complexity, and computational complexity. This makes the dynamic range of the performance metric quite large and unwieldy for practitioners to compare for model search and design purposes. To account for this large dynamic range, we take inspiration from the field of signal processing; in particular, the logarithmic scale commonly used to express the ratio between one value of a property to another. Here, we transform the ratio between the model accuracy property ($a(\mathcal{N})$) and the model architectural and computational complexity ($p(\mathcal{N})$ and $m(\mathcal{N})$) into the logarithmic scale to reduce the dynamic range to within a more readily interpretable range.

3 Experimental Results and Discussion

To get a better sense regarding the overall performance of the huge wealth of deep convolutional neural networks introduced in research literature in the context of practical usage, we perform a large-scale comparative analysis across a diverse set of 60 different deep convolutional neural networks designed for image classification using the following quantitative performance metrics: i) top-1 accuracy, ii) information density, and iii) the proposed NetScore metric. The dataset of choice for the comparative analysis in this study is the ImageNet Large Scale Visual Recognition Challenge (ILSVRC 2012) dataset [23], which consists of 1000 different classes. To the best of the author's knowledge, this comparative analysis is one of the largest in research literature and the hope is that the results presented in this study can act as a reference guide for practitioners in the field.

The set of deep convolutional neural networks being evaluated in this study are: AlexNet [19], AmoebaNet-A (4, 50) [24], AmoebaNet-A (6, 190) [24], AmoebaNet-A (6, 204) [24], AmoebaNet-B (3, 62) [24], AmoebaNet-B (6, 190) [24], AmoebaNet-C (4, 50) [24], AmoebaNet-C (6, 228) [24], CondenseNet (G=C=4) [16], CondenseNet (G=C=8) [16], DenseNet-121 (k=32) [17], DenseNet-169 (k=32) [17], DenseNet-161 (k=48) [17], DenseNet-201 (k=32) [17], DPN-131 [2], GoogleNet [31], IGC-L100M2 [35], IGC-L16M16 [35], IGC-L100M2 [35], Inception-ResNetv2 [30], Inceptionv2 [32], Inceptionv3 [32], Inceptionv4 [30], MobileNetv1 (1.0-224) [14], MobileNetv1 (1.0-192) [14], MobileNetv1 (1.0-160) [14], MobileNetv1 (1.0-128) [14], MobileNetv1 (0.75-224) [14], MobileNetv2 [26], MobileNetv2 (1.4) [26], NASNet-A (4 @ 1056) [38], NASNet-A (6 @ 4132) [38], NASNet-B (4 @ 1536) [38], NiN [20], OverFeat [27], PNASNet-5 (4, 216) [21], PolyNet [37], PreResNet-152 [13], PreResNet-200 [13], PyramidNet-101 (alpha=250) [9], PyramidNet-200 (alpha=300) [9], PyramidNet-200 (alpha=450) [9], ResNet-152 [12], ResNet-50 [12], ResNet-101 [12], ResNeXt-101, SENet [15], ShuffleNet (1.5) [36], ShuffleNet (x2) [36], SimpleNet [10], SqueezeNet [18], SqueezeNetv1.1 [18], SqueezeNext (1.0-23v5) [7], SqueezeNext (2.0-23) [7], SqueezeNext (2.0-23v5) [7], TinyDarkNet [25], VGG16 [29], Xception [3], ZynqNet [8]. In this study, the units used for $p(\mathcal{N})$ and $m(\mathcal{N})$ are in M-Params (millions of parameters) and G-MACs (billions of MAC operations), respectively.

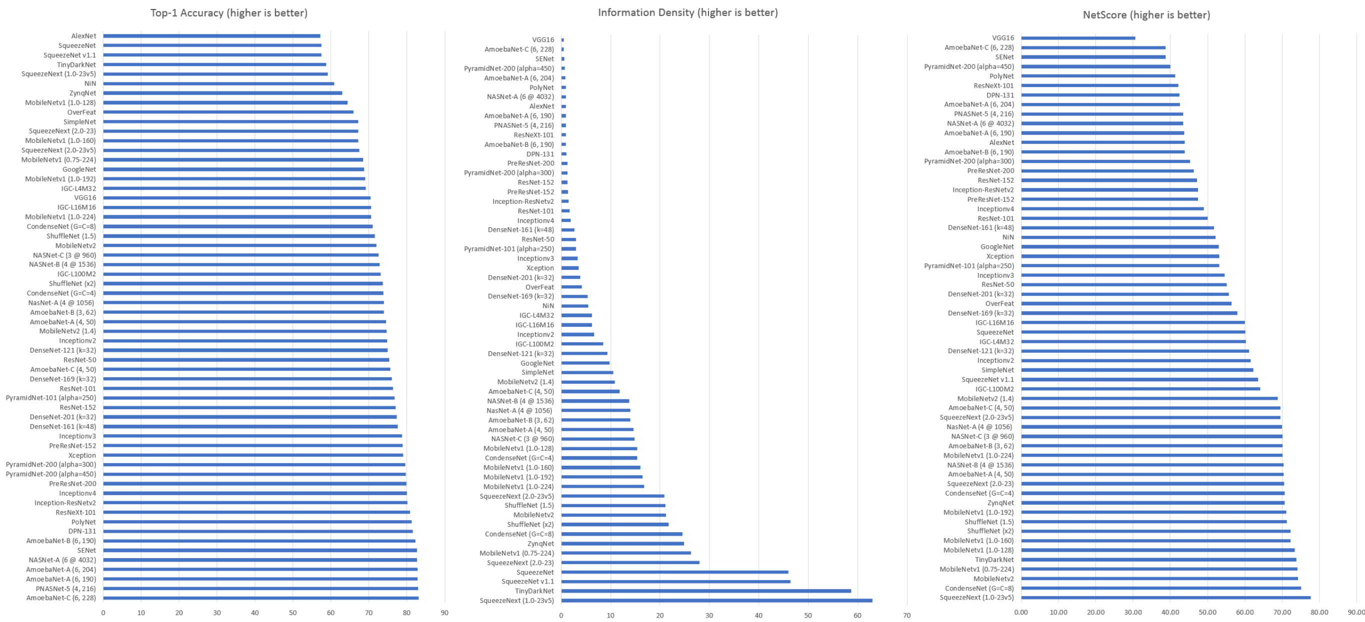

Figure 1: Top-1 accuracy, information density, and NetScore across 60 different deep convolutional neural networks for the ILSVRC 2012 dataset. Units are in %/M-Params for information density.

The top-1 accuracies across 60 different networks (shown in Fig. 1(left)) clearly illustrate the significant progress made in network design for image classification over the past six years, with the difference between the network with the highest top-1 accuracy in this study (i.e., AmoebaNet-C (6, 228)) and that of AlexNet exceeding 25%. The information densities across 60 different deep convolutional neural networks for the ILSVRC 2012 dataset, shown in Fig. 1(middle), clearly illustrates that the deep convolutional neural networks that were specifically designed for efficiency (e.g., MobileNetv1, MobileNetv2, ShuffleNet, SqueezeNet, Tiny DarkNet, and SqueezeNext) have significantly higher information densities compared to networks that were designed purely with accuracy as a metric. More specifically, the SqueezeNext (1.0-23v5), Tiny DarkNet, and the SqueezeNet family of networks had the highest information density by a wide margin compared to the other tested deep convolutional neural networks, which can be attributed to their significantly lower architectural complexity in terms of number of network parameters. Another notable observation from the results in Fig. 1(middle) is that the dynamic range of the information density metric is quite large across the diverse set of 60 deep convolutional neural networks evaluated in this study. Finally, the NetScore across 60 different deep convolutional neural networks for the ILSVRC 2012 dataset is shown in Fig. 1(right). Similar to the trend observed in Fig. 1(middle), it can be clearly observed that many of the deep convolutional neural networks that were specifically designed for efficiency have significantly higher NetScores compared to networks that were designed purely with accuracy as a metric. However, what is interesting to observe is that the NetScore ranking amongst these efficient networks are quite different than that when using the information density metric. In particular, the top ranking deep convolutional neural networks with the highest NetScores are SqueezeNext (1.0-23v5), CondenseNet (G=C=8), and MobileNetv2.

A number of examples illustrate the efficacy of NetScore over information density for providing a more complete profile of network efficiency and performance. For example, CondenseNet(G=C=8) has slightly lower information density than ZynqNet, but has $\sim 2\times$ lower computational complexity and much higher accuracy. The NetScore, in this case, is much higher for CondenseNet(G=C=8) compared to ZynqNet (higher by >4 units). In another example, MobileNetv1(0.75-224) has more than $2\times$ parameters than SqueezeNet, and thus has much lower information density. However, the computational complexity of SqueezeNet is $> 26\times$ greater than MobileNetv1(0.75-224) and accuracy much lower, and as such is reflected by a much higher NetScore for MobileNetv1(0.75-224) compared to SqueezeNet (higher by >14 units).

The proposed NetScore metric, which by no means is perfect, could potentially be useful for guiding practitioners in model search and design and hopefully push the conversation towards better universal metrics for evaluating deep neural networks for use in practical scenarios. NetScore can, for example, be used to narrow down a selection of network architecture candidates from a huge number of network architectures available to evaluate deeper on target hardware for hardware-specific usage.

References

[1] A. Canziani, A. Paszke, and E. Culurciello. An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678*, 2017.

[2] Yunpeng Chen, Jianan Li, Huaxin Xiao, Xiaojie Jin, Shuicheng Yan, and Jiashi Feng. Dual path networks. *CoRR*, abs/1707.01629, 2017.

[3] François Chollet. Xception: Deep learning with depthwise separable convolutions. *CoRR*, abs/1610.02357, 2016.

[4] Y. Le Cun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 2015.

[5] Y. Le Cun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of IEEE*, 1998.

[6] Y. Le Cun, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel. Handwritten digit recognition with a back-propagation network. *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, 1989.

[7] Amir Gholami, Kiseok Kwon, Bichen Wu, Zizheng Tai, Xiangyu Yue, Peter H. Jin, Sicheng Zhao, and Kurt Keutzer. Squeezenext: Hardware-aware neural network design. *CoRR*, abs/1803.10615, 2018.

[8] D. Gschwend. Zynqnet: An fpga-accelerated embedded convolutional neural network. *https://github.com/dgschwend/zynqnet*, 2016.

[9] Dongyoon Han, Jiwhan Kim, and Junmo Kim. Deep pyramidal residual networks. *CoRR*, abs/1610.02915, 2016.

[10] Seyyed Hossein HasanPour, Mohammad Rouhani, Mohsen Fayyaz, and Mohammad Sabokrou. Lets keep it simple, using simple architectures to outperform deeper and more complex architectures. *CoRR*, abs/1608.06037, 2016.

[11] K. He, G. Gkioxari, P. Dollar, and R. Girshick. Mask r-cnn. *ICCV*, 2017.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. *CoRR*, abs/1603.05027, 2016.

[14] A. Howard, M. Zhu, B. Chen, D. Kalenichenko, W., T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[15] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. *CoRR*, abs/1709.01507, 2017.

[16] Gao Huang, Shichen Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Condensenet: An efficient densenet using learned group convolutions. *CoRR*, abs/1711.09224, 2017.

[17] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *CoRR*, abs/1608.06993, 2016.

[18] F. Iandola, S. Han, M. Moskewicz, K. Ashraf, W. Dally, and K. Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

[19] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[20] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *CoRR*, abs/1312.4400, 2013.

[21] Chenxi Liu, Barret Zoph, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan L. Yuille, Jonathan Huang, and Kevin Murphy. Progressive neural architecture search. *CoRR*, abs/1712.00559, 2017.

[22] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. Berg. SSD: Single shot multibox detector. In *ECCV*, 2016.

[23] H. Su J. Krause S. Satheesh S. Ma Z. Huang A. Karpathy A. Khosla M. Bernstein et al. journal=International Journal of Computer Vision year=2015 O. Russakovsky, J. Deng. Imagenet large scale visual recognition challenge.

[24] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V. Le. Regularized evolution for image classifier architecture search. *CoRR*, abs/1802.01548, 2018.

[25] J. Redmon. Tiny darknet. *https://pjreddie.com/darknet/tiny-darknet/*, 2016.

[26] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. *arXiv preprint arXiv:1704.04861*, 2017.

[27] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR*, abs/1312.6229, 2013.

[28] M. Shafiee, F. Li, B. Chwyl, and A. Wong. Squishednets: Squishing squeezenet further for edge device scenarios via deep evolutionary synthesis. In *NIPS*, 2017.

[29] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[30] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *CoRR*, abs/1602.07261, 2016.

[31] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[32] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015.

[33] Alexander Wong, Mohammad Javad Shafiee, and Michael St. Jules. muNet: A highly compact deep convolutional neural network architecture for real-time embedded traffic sign classification. *CoRR*, abs/1804.00497, 2018.

[34] Alexander Wong, Mohammad Javad Shafiee, Francis Li, and Brendan Chwyl. Tiny SSD: A tiny single-shot detection deep convolutional neural network for real-time embedded object detection. *CoRR*, abs/1802.06488, 2018.

[35] Ting Zhang, Guo-Jun Qi, Bin Xiao, and Jingdong Wang. Interleaved group convolutions for deep neural networks. *CoRR*, abs/1707.02725, 2017.

[36] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. *CoRR*, abs/1707.01083, 2017.

[37] Xingcheng Zhang, Zhizhong Li, Chen Change Loy, and Dahua Lin. Polynet: A pursuit of structural diversity in very deep networks. *CoRR*, abs/1611.05725, 2016.

[38] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable architectures for scalable image recognition. *CoRR*, abs/1707.07012, 2017.