Multi-Residual Networks

Masoud Abdi and Saeid Nahavandi*
{mabdi,saeid.nahavandi}@deakin.edu.au

arXiv:1609.05672v1 [cs.CV] 19 Sep 2016

***Abstract*—In this article, we take one step toward understanding the learning behavior of deep residual networks, and supporting the hypothesis that deep residual networks are exponential ensembles by construction. We examine the *effective range* of ensembles by introducing *multi-residual networks* that significantly improve classification accuracy of residual networks. The multi-residual networks increase the number of residual functions in the residual blocks. This is shown to improve the accuracy of the residual network when the network is deeper than a threshold. Based on a series of empirical studies on CIFAR-10 and CIFAR-100 datasets, the proposed multi-residual network yield 6% and 10% improvement with respect to the residual networks with identity mappings. Comparing with other state-of-the-art models, the proposed multi-residual network obtains a test error rate of 3.92% on CIFAR-10 that outperforms all existing models.**

***Index Terms*—Deep residual networks, convolutional neural networks, deep neural networks, image classification.**

I. Introduction

CONVOLUTIONAL neural networks [1], [2] have led to a series of breakthroughs in tackling image recognition and visual understanding problems [2]–[4]. They have been applied in many areas of engineering and science [5]–[7]. Increasing the network depth is known to improve the model capabilities, which can be seen from AlexNet [2] with 8 layers, VGG [8] with 19 layers to GoogleNet [9] with 22 layers. However, increasing the depth can be challenging for the learning process because of the vanishing gradient problem [10], [11]. Deep residual networks [12] avoid this problem using identity skip-connections, which help the gradient to flow back into many layers without vanishing. The identity skip-connections facilitate training of very deep networks up to thousands of layers, which helped residual networks win five major image recognitions tasks in ILSVRC 2015 [13] and Microsoft COCO 2015 [14] competitions.

However, an obvious drawback of residual networks is that every percentage of improvement requires significantly increasing the number of layers, which linearly increases the computational and memory costs [12]. On CIFAR-10 classification dataset, deep residual networks with 164-layers and 1001-layers reach test error rate of 5.46% and 4.92% respectively, while the 1001-layer has six times more computational complexity than the 164-layer. On the other hand, wide residual networks [15] have 50 times fewer layers while outperforming the original residual networks by increasing the number of convolutional filters. It seems that the power of residual networks is due to the identity skip-connections, rather than extremely increasing the network depth.

* M. Abdi and S. Nahavandi are with the Institute for Intelligent Systems Research and Innovation, Deakin University, 3216 VIC, Australia

Manuscript received April 19, 2005; revised August 26, 2015.

Nevertheless, a recent study supports that deep residual networks act like ensembles [16]. It is claimed that residual networks are exponential ensembles of relatively shallow networks, rather than a single deep network. This is achieved by showing the existence of exponential paths from the output layer to the input layer that the gradient information can flow, and showing that most of the gradient updates during optimization come from ensembles of relatively shallow depth. Moreover, residual networks do not resolve the vanishing gradient problem by preserving the gradient through the entire depth of the network. Instead, they avoid the problem by ensembling exponential networks together. This raises the importance of *multiplicity* which is the number of possible paths from the input layer to the output layer [16].

Inspired by these observations, we introduce multi-residual networks which increase the multiplicity of the network, while keeping its depth fixed. This is achieved by increasing the number of residual functions in each residual block. We then show that a shallow multi-residual network recovers the accuracy of a 110-layer residual network, supporting the hypothesis that deep residual networks are ensembles, instead of extremely deep networks. Next, we examine the importance of *effective range*, which is the range of paths that significantly contribute toward gradient updates. A series of experiments and analyses on the influence of the effective range is also conducted to demonstrate the usefulness of the proposed multi-residual network.

We show that for a residual network deeper than a threshold n_0, increasing the number of residual functions leads to a better performance than increasing the network depth. This leads to a lower error rate for the multi-residual network with the same complexity as the deeper residual network. Finally, we perform experiments on CIFAR-10 and CIFAR-100 image classification datasets, and we show that multi-residual networks improve the accuracy of deep residual networks and outperform other state-of-the-art models. Using moderate data augmentation, multi-residual networks achieve an error rate of 4.35% and 20.42% on CIFAR-10 and CIFAR-100 receptively (based on five runs). This is 6% and 10% improvement compered to the residual networks with identity mappings [19] with almost the same computational and memory complexity. A more complicated multi-residual network achieves a test error rate of 3.92% on CIFAR-10. To the best of our knowledge, these results outperform those from the existing models.

In summary, the contributions of this research are:

- We take one step toward understanding deep residual networks and supporting the hypothesis that deep residual networks are exponential ensembles of shallow networks, rather than a single very deep network.

- Through a series of experiments, we show the importance of the effective range in residual networks that is the range of ensembles that significantly contribute gradient update during optimization.
- We introduce multi-residual networks based on our understanding of the residual networks, which is shown to improve the accuracy of residual networks and all other state-of-the-art models.

The rest of the paper is organized as follows. Section II details deep residual networks and other models capable of improving the original residual networks. The hypothesis that residual networks are exponential ensembles of relatively shallow networks is explained in Section III. The proposed multi-residual networks and the importance of the effective range are discussed in Section IV. The supporting experimental results are presented in Section V. Concluding remarks are provided in Section VI

II. Related Works

Deep residual networks [12] are able to significantly outperform many state-of-the-art models in image classification. The main idea is the use of identity skip-connections, which help the gradient to bypass the weight layers. The identity skip-connections preserve the gradient from vanishing, which in turn enables training of deep networks up to thousands of layers. On the other hand, increasing the network depth is known to improve the model capabilities [17]. Therefore, it seems that residual networks capture the power of increasing the network depth.

A residual block consists of a function f, and an identity connection (see Figure 2). It computes $x + f(x)$, instead of $f(x)$, where f contains convolution, activation (ReLU) and batch normalization [18] layers in a specific order. In the original residual network that won the ImageNet competition, the order of f is convolution-normalization-ReLU [12]. Whilst, in a newer version the order changes to normalization-ReLU-convolution which is known as pre-activation model [19].

A deep residual network contains many stacked residual blocks with $y = x + f(x)$, where x and y are the input and output of the block respectively, and f is the weight layer. Therefore, a deep residual network with the identity skip-connections [19] can be represented as follows:

$$x_{l+1} = x_l + f_l(x_l) \tag{1}$$

where x_l is the input of l^{th} residual block, and f_l contains the weight layers. Additionally, Highway Networks [20], [21] also employ parametrized skip-connections, which are referred to as *information highways*. The skip-connection parameters are learned during training, which control the amount of information that can pass through the skip-connections.

Residual networks with stochastic depth [22] use Bernoulli random variables to randomly disable the residual blocks during the training phase. This results in a shallower network at the training phase, while having a deeper network at the test phase. Residual networks with stochastic depth improve the accuracy of the residual networks with constant depth. This is because of the reduction in the network depth which strengthens the back-propagated gradients of the earlier layers, and because of ensembling networks of different depths.

Swapout [23] generalizes dropout [24] and networks with stochastic depth [22] using $px+qF(x)$, where p and q are two Bernoulli random variables. Swapout has the ability to sample from four network architectures $\{0, x, F(x), x+F(x)\}$, therefore having a larger domain for sampling ensembles. Wide residual networks [15] increase the number of convolutional filters, and are able yield a better performance than the original residual networks. This suggests that the power of residual networks originate in the residual connections, as opposed to extremely increasing the network depth.

III. Deep Residual Networks are Exponential Ensembles

Residual networks [12] are assumed to resolve the problem of vanishing gradients using the identity skip-connections that facilitate training of deep networks up to 1202 layers. Nonetheless, recent studies support that deep residual networks do not resolve the vanishing gradient problem by preserving the gradient flow through the entire depth of the network. Instead, they avoid the problem simply by ensembling exponential networks together [16].

Consider a residual network with three residual blocks, and let x_0 and x_3 be the input and output respectively, applying Equation 1 iteratively gives:

$$\begin{aligned} x_3 &= x_2 + f_3(x_2) \\ &= \Big[x_1 + f_2(x_1)\Big] + f_3(x_1 + f_2(x_1)) \\ &= \Big[x_0 + f_1(x_0) + f_2(x_0 + f_1(x_0))\Big] \\ &\quad + f_3(x_0 + f_1(x_0) + f_2(x_0 + f_1(x_0))) \end{aligned} \tag{2}$$

A graphical view of Equation 2 is presented in Figure 1a. It is clear that data flow along the exponential paths from the input to the output layer. In other words, every path is a unique configuration that either compute a particular residual function $f_l(l = 1, \ldots, n)$ or skip it. Therefore, the total number of possible paths from the input to the output is 2^n, where n is the number of residual blocks. This term is referred to as the *multiplicity* of the network. Furthermore, a residual network can be viewed as a very large implicit ensembles of many networks.

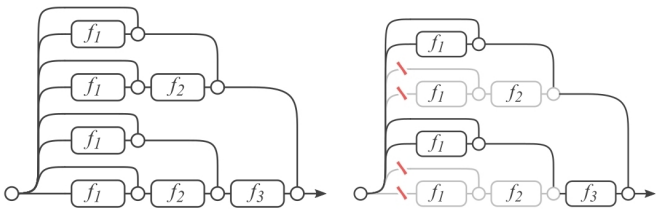

Fig. 1: (a) A residual network; (b) Deleting f_2 from a residual network [16]. It can be seen that residual networks have 2^n paths connecting the input to the output. Deleting a block from the residual network reduces the number of paths to 2^{n-1}.

Deep residual networks are resilient to dropping and reordering the residual blocks during the test phase. More precisely, removing a single block from a 110-layer residual

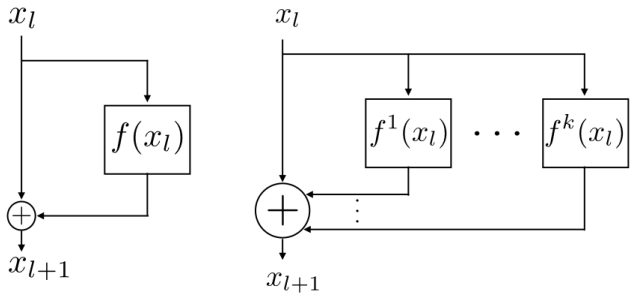

Fig. 2: A residual block (left) versus a multi-residual block (right).

network, during the test phase, has a negligible effect on its performance. Whereas, removing a layer from the traditional network architectures, such as AlexNex [2] or VGGnet [8], dramatically hurts the performance of the models (test error more than 80%) [16]. This supports the existence of exponential paths from the input to the output layer. Moreover, removing a single residual block during the test phase reduces the number of paths from 2^n to 2^{n-1} (see Figure 1b),

Additionally, shallow ensembles contribute significantly to the gradient updates during optimization . In other words, in a 110-layer residual network, most of the gradient updates come from paths with only 10-34 layers, and deeper paths do not have significant contribution towards the gradient updates. These are called the *effective paths*, which are relatively shallow compared to the network depth [16].

In order to verify the claim pertaining to the shallow ensembles, one can see that individual paths in a deep residual network have a binomial distribution, where the number of paths with length k is $\binom{n}{k} = \frac{n!}{k!(n-k)!}$. On the other hand, it has been known that the gradient magnitude, during back-propagation, decreases exponentially with the number of residual functions it goes through [10], [11]. Therefore, the total gradient magnitude contributed by paths of each length can be calculated by multiplying the number of paths with that length, and the expected gradient magnitude of the paths with the same length [16].

Accordingly, a residual network trained with only effective paths has a comparable performance with the full residual network [16]. This is achieved by randomly sampling a subset of residual blocks for each mini-batch, and forcing the computation to flow through the selected blocks only. In this case the network can only see the effective paths that are relatively shallow, and no long path is used.

IV. Multi-Residual Networks

Based on the aforementioned observations, we propose *multi-residual networks* that aim to increase the multiplicity of the residual network, while keeping the depth fixed. The multi-residual network employs multiple residual functions, f^i, instead of one residual function for each residual block (see Figure 2). As such, a deep multi-residual network with k residual functions has:

$$x_{l+1} = x_l + f_l^1(x_l) + f_l^2(x_l) + \cdots + f_l^k(x_l) \tag{3}$$

where f_l^i is the i^{th} residual function of the l^{th} residual block. Expanding Equation 3 for $k = 2$ residual functions and three multi-residual blocks gives:

$$\begin{aligned}
x_3 =& x_2 + f_3^1(x_2) + f_3^2(x_2) \\
=& \Big[x_1 + f_2^1(x_1) + f_2^2(x_1)\Big] + \Big[f_3^1(x_1 + f_2^1(x_1) + f_2^2(x_1))\Big] \\
&+ \Big[f_3^2(x_1 + f_2^1(x_1) + f_2^2(x_1))\Big] \\
=& \Big[x_0 + f_1^1(x_0) + f_1^2(x_0) + f_2^1(x_0 + f_1^1(x_0) + f_1^2(x_0)) \\
&+ f_2^2(x_0 + f_1^1(x_0) + f_1^2(x_0))\Big] \\
&+ \Big[f_3^1(x_0 + f_1^1(x_0) + f_1^2(x_0) + f_2^1(x_0 + f_1^1(x_0) + f_1^2(x_0)) \\
&+ f_2^2(x_0 + f_1^1(x_0) + f_1^2(x_0)))\Big] \\
&+ \Big[f_3^2(x_0 + f_1^1(x_0) + f_1^2(x_0) + f_2^1(x_0 + f_1^1(x_0) + f_1^2(x_0)) \\
&+ f_2^2(x_0 + f_1^1(x_0) + f_1^2(x_0)))\Big]
\end{aligned} \tag{4}$$

It can be seen that the number of terms in Equation 4 is exponentially more than the number of terms in Equation 2. Specifically, in a multi-residual block with $k = 2$ residual functions, the gradient flow has four possible paths: (1) skipping both f^1 and f^2, (2) skipping f^1 and performing f^2, (3) skipping f^2 and performing f^1, (4) performing both f^1 and f^2. Therefore, the multiplicity of the multi-residual network with $k = 2$ residual functions is 4^n. In other words, the multiplicity of a multi-residual network with k residual functions and n multi-residual blocks is 2^{kn}. This is because every residual function can be either computed or otherwise, giving a multiplicity of 2^k for a block, and a total multiplicity of 2^{kn} for the multi-residual network.

A. Residual Networks are Ensembles

Based on multi-residual networks, we show that residual networks are ensembles. A shallow multi-residual network with the same number of parameters as a 110-layer residual network is able to recover the accuracy of the residual network. This supports the hypothesis that residual networks are exponential ensembles of shallow networks, rather than very deep networks.

method	depth	k	parameters	CIFAR-10(%)
resnet [12]	110	1	1.7M	6.61
pre-resnet [19]	110	1	1.7M	6.37
multi-resnet [ours]	8	23	1.7M	7.37
	14	10	1.7M	6.42

TABLE I: Classification error on CIFAR-10 test set. A shallow multi-residual network is able to approximate the accuracy of a 110-layer residual network.

A multi-residual network with the depth of 8 and $k = 23$ residual functions, and a multi-residual network with the depth of 14 and $k = 10$ residual functions are trained. Both networks have roughly the same number of parameters, which is the same as those in the 110-layer residual network. The networks are trained with the same hyper-parameters and training policy

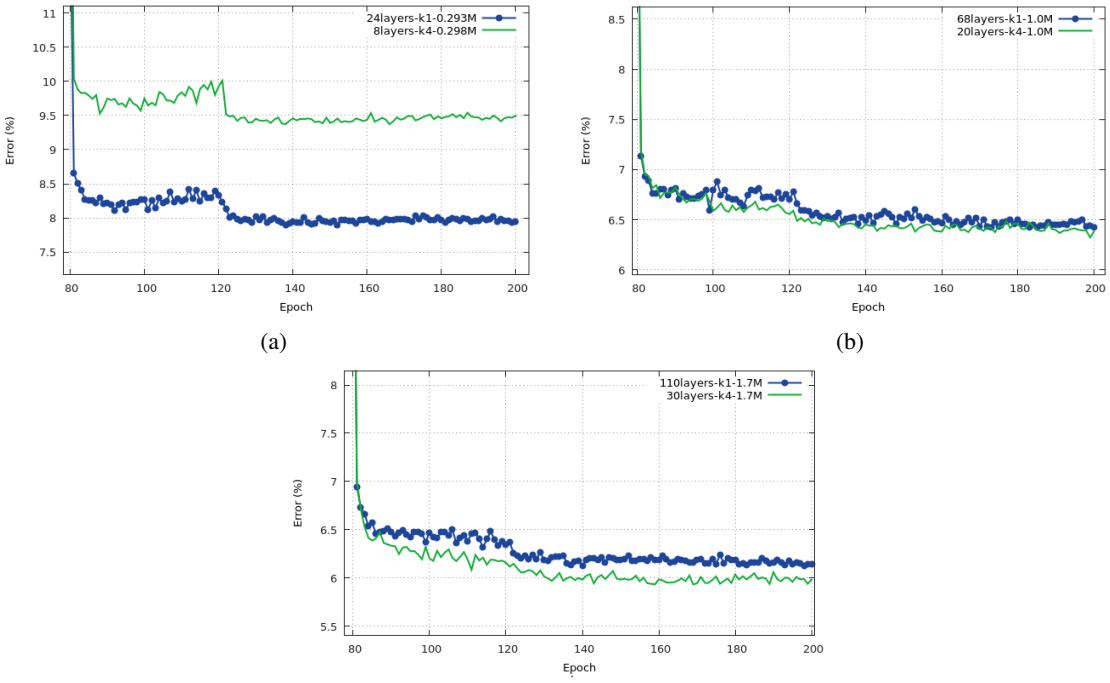

Fig. 3: Comparing residual network and the proposed multi-residual network on CIFAR-10 test set to show the effective range phenomena. Each curve is mean over 5 runs. (a) This is where the network depth $< n_0$ in which the multi-residual network performs worse than the original residual network; (b) Both networks have comparable performance; (c) The proposed multi-residual network outperforms the original residual network.

as in [12]. Table I summaries the test errors on CIFAR-10. It can be seen that classification accuracy of the shallow multi-residual network with 14-layer depth almost reaches that of the 110-layer residual network.

B. The Effective Range

Based on the hypothesis that residual networks are ensembles of shallow networks, a question is posed: **what is the relationship between the range of the effective paths and the depth of the residual network?** More precisely, what is the relationship between the effective range of a residual network with n residual blocks and that of a residual network with cn residual blocks, where c is a constant number?

We hypothesize that this relationship is **not linear**. This implies that if the effective range of a residual network with n blocks is $[a, b]$, the effective range of a residual network with cn blocks is not $[ca, cb]$. Instead, it is shifted and/or scaled toward shallower networks. This is because of the exponential reduction in the gradient magnitude. Eventually, the upper bound of the effective range is lower than cb. This could be a potential reason for the problem that every percentage of improvement in deep residual networks requires significantly increasing the number of layers.

C. Residual Networks versus Multi-Residual Networks

Consider a residual network, R, with n residual blocks, and let c be a constant integer. We would like to construct two residual networks by: (1) increasing the number of residual blocks to cn, which results in a residual network with c times depth of R (excluding the first and last layers), (2) retaining the same depth while increasing the number of residual functions by c. The number of parameters of the subsequent networks are roughly the same. One can also see that the multiplicity of both networks are 2^{cn}, but **how about the effective range of (1) and (2)**?

As discussed in the previous part, the effective range of (1) does not increase linearly, whereas the effective range of (2) increases linearly due to the increase in the residual functions. This is owing to the increase in the number of the paths of each length, which is a consequence of changing the binomial distribution to a multinomial distribution. Note that this analysis holds true for $n \geq n_0$, where n_0 is a threshold value; otherwise the power of the network depth is clear both in theory [17], [25], [26] and in practice [2], [8], [9].

V. Experimental Results

To support our analyses and show the effectiveness of the proposed multi-residual networks, a series of experiments has been conducted on CIFAR-10 and CIFAR-100 datasets. Both datasets contain 50,000 training samples and 10,000 test samples of 32×32 color images, with 10 (CIFAR-10) and 100 (CIFAR-100) different categories. We have used *"moderate data augmentation"* (flip/translation) as in [19], and training is done using stochastic gradient descent for 200 epochs with a weight decay of 10^{-4} and momentum of 0.9 [12]. The network weights have been initialized as in [27]. The learning rate starts with 0.1 and divides by 10 at epochs 80 and 120. The code is available at: https://github.com/masoudabd/multi-resnet

method				CIFAR-10(%)	CIFAR-100(%)
NIN [28]				8.81	35.68
DSN [29]				8.22	34.57
FitNet [30]				8.39	35.04
Highway [20]				7.72	32.39
All-CNN [31]				7.25	33.71
ELU [32]				6.55	24.28
method	depth	k,(w)	#parameters		
resnet [12]	110	1	1.7M	6.43$_{(6.61\pm0.16)}$	25.16
	1202	1	19.4M	7.93	27.82
pre-resnet [19]	110	1	1.7M	6.37	-
	164	1	1.7M	5.46	24.33
	1001	1	10.2M	4.62$_{(4.69\pm0.20)}$†	22.71$_{(22.68\pm0.22)}$
stoch-depth [22]	110	1	1.7M	5.25	24.58
	1001	1	10.2M	4.91	-
swapout [23]	20	1,(2)	1.1M	6.58	25.86
	32	1,(4)	7.43M	4.76	22.72
wide-resnet [15]	40	1,(4)	8.7M	4.97	22.89
	16	1,(8)	11.0M	4.81	22.07
	28	1,(10)	36.5M	4.17	20.50
multi-resnet [ours]	200	5	10.2M	**4.35**$_{(4.36\pm0.04)}$†	**20.42**$_{(20.44\pm0.15)}$†
	398	5	20.4M	**3.92**†	-

TABLE II: Test error rates on CIFAR-10 and CIFAR-100. The results in the form of $median(mean \pm std)$ are based on five runs, while others are based on one run. All results are obtained with a mini-batch size of 128, except † with a mini-batch size of 64. The number of residual functions is denoted as k, and (w) is the widening factor for wider models.

A. The Effective Range Phenomena

Consider a pre-activation version of the residual network with the basic-blocks [19]. Three pairs of residual network and multi-residual network are trained. The residual network is k times deeper than the corresponding multi-residual network (excluding the first and last layers). On the other hand, the multi-residual network computes k residual functions. A residual block might be removed to compensate the difference in the number of parameters to form a fair comparison between the pairs. The median of five runs with mean±std in the parentheses are reported in Table III. The test error curves are also depicted in Figure 3, where each curve is the mean of the five runs. All networks are trained with the same hyper-parameters and training policy with a mini-batch size of 128.

method	depth	k	parameters	CIFAR-10(%)
pre-resnet [19]	24	1	0.29M	7.75 $_{(7.76\pm0.13)}$
	68	1	1.0M	6.27 $_{(6.33\pm0.24)}$
	110	1	1.7M	6.02 $_{(6.02\pm0.11)}$
multi-resnet [ours]	8	4	0.29M	9.28 $_{(9.28\pm0.07)}$
	20	4	1.0M	6.31 $_{(6.29\pm0.22)}$
	30	4	1.7M	5.89 $_{(5.85\pm0.12)}$

TABLE III: CIFAR-10 test errors of the multi-residual networks and the original residual networks, where k is the number of residual functions. The results are in the form of $median$ with $mean \pm std$ in parentheses from five runs.

The multi-residual network with 8-layers depth has a test error rate of 9.28%, while the original residual network with 24 layers, and roughly the same number of parameters, has an error rate of 7.75%. This is the scenario whereby the network depth is too shallow ($depth < n_0$), and the multi-residual network performs worse than the residual network (see Figure 3a). On the contrary, the multi-residual network with 20-layers depth achieves 6.31% error rate, which is statistically no different than 6.27% for the 68-layer residual network. Test curves (Figure 3b) also show that both networks have a comparable performance.

Eventually, a 30-layer deep multi-residual network achieves 5.89% error rate. This is slightly better than the 110-layer residual network that have the error of 6.02% (6.37% in [19]). Figure 3c also clearly shows that the multi-residual network performance is superior to the original residual network. It can be seen that although each pair have almost the same number of parameters and computational complexity, they act very differently. These results supports the hypothesis pertaining to the effective range.

In the previous section, we argue that multi-residual network is able to improve classification accuracy of the residual network when the network is deeper than a threshold n_0. This effect can be seen in Figure 3. Based on the observations in Table III, for this particular dataset and network architecture, the threshold n_0 is approximately 20. Furthermore, by increasing the number of residual functions, better accuracy can be obtained. However, a trade-off has been observed between the network depth and the number of residual function. This means that, one might need to choose a suitable number of residual functions, and depth to achieve the best performance.

B. Improving Classification Accuracy of Existing Models

Table II shows the results of the multi-residual networks along with those from the original residual networks [12], [19] and other state-of-the-art models. The networks with $6n + 2$ layers use the basic block with two 3×3 convolutional layers, and the networks with $9n+2$ layers use the bottleneck block architecture, which has a single 3×3 convolution layer surrounded by two 1×1 convolutional layers [12].

One can see that the proposed multi-residual network outperforms the classification accuracy of existing models (to the best of our knowledge) on CIFAR-10 and CIFAR-100. These models include the original residual networks [12], pre-activation residual networks with identity mappings [19],

residual networks with stochastic depth [22], swapout [23], and wide residual networks [15].

Complexity of the proposed model. Increasing the number of residual functions increases the number of parameters by a factor of k, and the computational complexity of the multi-residual network also increases linearly with the number of residual functions. This results in a model with the same memory and computational complexity as the original residual networks [19]. It is necessary to mention that all of our results are obtained using the same number of convolutional filters as the original residual networks [12]. Therefore, it is possible that a wider version of multi-residual networks attain a better performance than those in Table II.

VI. Conclusions

Experiments in this article support the hypothesis that residual networks are ensembles, rather than extremely deep networks. Based on a series of analyses and observations, multi-residual networks are introduced. The proposed multi-residual network is capable of enhancing classification accuracy of the original residual network. Multi-residual networks exploit multiple residual functions for the residual blocks. We argue that this serves a better way to increase the representational capabilities of the residual networks because of the described effective range phenomena. However, a trade-off has been observed between the network depth and the number of residual functions. The proposed multi-residual networks significantly improve the classification accuracy of existing models.

Acknowledgment

The authors would like to thank the National Computational Infrastructure of Australia for providing us with high performance computing services. Special thanks to Chee Peng Lim for useful discussions. We also thank Christopher McAvaney and Matthew Watson for helping us with the facilities.

References

[1] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

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

[4] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pages 818–833. Springer, 2014.

[5] Izhar Wallach, Michael Dzamba, and Abraham Heifets. Atomnet: A deep convolutional neural network for bioactivity prediction in structure-based drug discovery. *arXiv preprint arXiv:1510.02855*, 2015.

[6] Rahul Mohan. Deep deconvolutional networks for scene parsing. *arXiv preprint arXiv:1411.4101*, 2014.

[7] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[8] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[9] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[10] Sepp Hochreiter. Untersuchungen zu dynamischen neuronalen netzen. *Diploma, Technische Universität München*, page 91, 1991.

[11] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[13] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[15] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[16] Andreas Veit, Michael Wilber, and Serge Belongie. Residual networks are exponential ensembles of relatively shallow networks. *arXiv preprint arXiv:1605.06431*, 2016.

[17] Ronen Eldan and Ohad Shamir. The power of depth for feedforward neural networks. *arXiv preprint arXiv:1512.03965*, 2015.

[18] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. *arXiv preprint arXiv:1603.05027*, 2016.

[20] Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks. In *Advances in neural information processing systems*, pages 2377–2385, 2015.

[21] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

[22] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Weinberger. Deep networks with stochastic depth. *arXiv preprint arXiv:1603.09382*, 2016.

[23] Saurabh Singh, Derek Hoiem, and David Forsyth. Swapout: Learning an ensemble of deep architectures. *arXiv preprint arXiv:1605.06465*, 2016.

[24] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[25] Johan Håstad and Mikael Goldmann. On the power of small-depth threshold circuits. *Computational Complexity*, 1(2):113–129, 1991.

[26] Johan Håstad. Computational limitations of small-depth circuits. 1987.

[27] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1026–1034, 2015.

[28] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[29] Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. Deeply-supervised nets. In *AISTATS*, volume 2, page 6, 2015.

[30] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

[31] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[32] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.