Effective, Fast, and Memory-Efficient Compressed Multi-function Convolutional Neural Networks with Compact Inception-V4

Anonymous Author(s)
Affiliation
Address
email

Abstract

Google's Inception-V4 using an activation function RELU is a very deep convolutional neural network (CNN) that consists of 4 Inception-A blocks, 7 Inception-B blocks, and 3 Inception-C blocks. To improve classification performance, reduce training and testing times, and reduce power consumption and memory usage (model size), a new "Compressed Multi-function Inception-V4" (CMIV4) using different activation functions is created by using k Inception-A blocks, m Inception-B blocks, and n Inception-C blocks where $k \in \{1, 2, 3, 4\}$, $m \in \{1, 2, 3, 4, 5, 6, 7\}$, $n \in \{1, 2, 3\}$, and $(k + m + n) < 14$. For performance analysis, two datasets for two different applications (classifying brain MRI images into one of the four stages of Alzheimer's disease and using a sample of CIFAR-10 data) are used to compare three CMIV4 architectures with Inception-V4 in terms of F1-score, training and testing times (related to power consumption), and memory usage (model size). Overall, simulations show that the new CMIV4 can outperform both the commonly used single-function CNN with Inception-V4 and multi-function CNNs with Inception-V4. In the future, other compressed multi-function CNNs, such as compressed multi-function ResNets and compressed multi-function DenseNets with a reduced number of convolutional blocks using different activation functions, will be developed to increase classification accuracy, reduce training and testing times, reduce computational power, and reduce memory usage (model size) for industrial applications in IoT, big data mining, green computing, etc.

1 Introduction

In recent years, deep learning techniques have been effectively used in various applications in computer vision, pattern recognition, etc. [1-15]. The new 28nm Two-Dimensional Convolutional Neural Network (CNN)-DSA accelerator with an ultra power-efficient performance of 9.3 TOPS/Watt was implemented for low-end mobile and embedded platforms and MCUs (Microcontroller Units) [16]. Since DenseNets require large GPU memory, new methods were developed to reduce the memory consumption for training them [17]. Currently, it is especially important to build effective, power-efficient, memory-efficient, and compact CNNs for applications in the internet of things (IoT), big data mining, green computing, etc. Traditional CNNs usually use the same activation function, such as Google's very deep Inception-V4 network [15] using the popular rectified linear unit (RELU). However, traditional CNNs using RELU may not be optimal for power-efficient and memory-efficient applications. Thus, we design an effective, fast, power-efficient, memory-efficient, and compact multi-function CNN architecture based on Inception-V4 ("Compressed Multi-function Inception-V4) (CMIV4), by using different activation functions and reducing the number of convolutional blocks.

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

2 Compressed Multi-function CNNs with Compact Inception-V4

A very deep convolutional neural network Inception-V4 with a commonly used activation function RELU uses 4 Inception-A blocks, 7 Inception-B blocks, and 3 Inception-C blocks [15]. However, Inception-V4 with RELU may not be optimal for different applications. Thus, a new Multi-function Inception-V4 (MIV4) is developed by using different activation functions. In order to improve classification performance, reduce training and testing times, reduce power consumption, reduce memory usage (model size), the new CMIV4 using different activation functions for compact Inception-V4 uses k Inception-A blocks, m Inception-B blocks, and n Inception-C blocks where $k \in \{1,2,3,4\}$, $m \in \{1,2,3,4,5,6,7\}$, $n \in \{1,2,3\}$, and $(k+m+n) < 14$. For instance, a CMIV4 using 1 Inception-A block, 2 Inception-B blocks, and 1 Inception-C block can run faster (use less power) and has a smaller model size (129MB) than the original Inception-V4 with RELU, which has a larger model size of 323MB. The CMIV4 uses 58 convolutional blocks with 58 functions and the Inception-V4 with RELU uses 149 convolutional blocks with 149 functions. The goal is to discover a CMIV4 model with better classification performance, faster training and testing times, and less power consumption and memory usage (model size) than the popular Google's Inception-V4.

3 Experimental Results

Let "CMIV4_x" and "RELx" mean that a CMIV4 and a compressed Inception-V4 with RELU have x Inception-A, x+1 Inception-B, and x Inception-C blocks. "MIV4" and "REL" means that a MIV4 and the original Inception-V4 with RELU have 4 Inception-A, 7 Inception-B, and 3 Inception-C blocks. Stratified 3-fold cross validation was used to evaluate and compare the three CMIV4 models, the MIV4 model, the three compressed Inception-V4 models with RELU, and the original Inception-V4 using multi-class classification metrics (i.e. training F1-score (F1_train), validation F1-scores (F1_valid), training times (Time_train) in seconds, and classification testing times (Time_test) in seconds. An activation function set {RELU, SIG, TANH, ELU} was used to build all of the multi-function models. Each activation function is randomly chosen from this set. The model sizes of CMIV4_1, CMIV_2, CMIV_3, and MIV4 are 129MB, 190MB, 252MB, and 323MB, respectively.

3.1 Application 1: Brain MRI Images

A dataset of 436 brain MRI images (cross-sectional collection of 416 subjects aged 18 to 96 and with extra data for 20 subjects), pre-processed and ready to be used, is used for performance analysis [18]. This research work uses all brain MRI images for a 4-class classification problem to determine the Alzheimer's Disease stage (non-demented, very mild dementia, mild dementia, or moderate dementia) of a person [18][19]. For each architecture (CMIV4_1, CMIV4_1, CMIV4_1, or MIV4), 10 random CMIV4 models and 10 random MIV4 models are created and tested. The highest cross-validation F1-score for each architecture is shown in Table 1 (50 training epochs). Table 1 shows that MIV4 using 323MB memory is better than the best CMIV4 by only 0.01 for F1_valid, but CMIV4_2 using 190MB memory is much faster (more power-efficient) and more memory-efficient. In addition, Table 1 shows that the three best CMIV4 models and one MIV4 model always performed better than both the three compressed Inception-V4 models with RELU (REL1, REL2 and REL3) and the original Google's Inception-v4 using RELU. REL1, REL2, and REL3 performed better, trained and predicted faster, and used less power and memory than REL did.

Table 1: Comparing the Best CMIV4 Models and MIV4 Model for Brain Images

Model:	CMIV4_1	REL1	CMIV4_2	REL2	CMIV4_3	REL3	MIV4	REL
F1_train	0.77	0.76	0.85	0.76	0.83	0.74	0.83	0.73
F1_valid	0.77	0.76	0.81	0.74	0.81	0.74	0.82	0.73
Time_train (s)	845	815	1139	1117	1456	1394	1869	1793
Time_test (s)	1.31	1.25	1.60	1.56	1.93	1.86	2.50	2.35

Average performance results for 10 CMIV4_1 models, 10 CMIV4_2 models, 10 CMIV4_3 models, and 10 MIV4 models are shown in Table 2 (90 training epochs). CMIV4_3 has shorter training and classification times, and less power consumption and memory usage (252MB), and it can perform better than the MIV4 model, which uses more memory (323MB).

Table 2: Average Performance of 30 CMIV4 Models and 10 MIV4 Models for Brain Images

Model:	CMIV4_1	CMIV4_2	CMIV4_3	MIV4
Avg. F1_train	0.726	0.758	0.772	0.773
Avg. F1_valid	0.717	0.745	0.760	0.759
Avg. Time_train (s)	1690	2325	2930	3751
Avg. Time_test (s)	1.31	1.54	1.87	2.36

3.2 Application 2: CIFAR10

A sample of the CIFAR10 data was used to test the performance of CMIV4 models compared to that of MIV4 models using RELU by randomly selecting the activation function for each neuron [20]. The training sample size is 1000 and the test sample size is 300. For each architecture (CMIV4_1, CMIV4_2, CMIV4_3, or MIV4), 8 random CMIV4 models and 8 random MIV4 models are created and tested. The highest cross-validation F1-score for each architecture is shown in Table 3. 40 training epochs were used. Table 3 shows that CMIV_1 and CMIV4_2 performed better than MIV4 and have faster training and test times. REL1, REL2, and REL3 performed better than REL and have faster training and test times. In addition, Table 3 shows that the best three CMIV4 models and one MIV4 model always performed better than both the three compressed Inception-V4 models with RELU (REL1, REL2 and REL3) and the original Google's Inception-v4 using RELU in terms of both cross-validation training F1-scores and validation F1-scores.

Table 3: Comparing the Best CMIV4 Models and MIV4 Model for CIFAR10

Model:	CMIV4_1	REL1	CMIV4_2	REL2	CMIV4_3	REL3	MIV4	REL
F1_train	0.59	0.44	0.61	0.20	0.49	0.14	0.54	0.10
F1_valid	0.56	0.42	0.57	0.20	0.47	0.15	0.53	0.09
Time_train (s)	2862	2801	3780	3766	4783	4714	6260	6001
Time_test (s)	7.05	6.84	8.68	8.43	10.0	9.93	13.0	12.6

Average performance results for 8 CMIV4_1 models, 8 CMIV4_2 models, 8 CMIV4_3 models, and 8 MIV4 models are shown in Table 4 (40 training epochs). All three compressed multi-function CNN models have shorter training and classification times, and less power consumption and memory usage than MIV4, and can still perform better than MIV4.

Table 4: Average Performance of 24 CMIV4 Models and 8 MIV4 Models for CIFAR10

Model:	CMIV4_1	CMIV4_2	CMIV4_3	MIV4
Avg. F1_train	0.477	0.465	0.449	0.445
Avg. F1_valid	0.459	0.443	0.430	0.423
Avg. Time_train (s)	2858	3811	4781	6123
Avg. Time_test (s)	7.39	8.59	10.1	12.9

4 Conclusions and Future Works

Simulation results show that CMIV4 can achieve both better performance, shorter training and testing times (i.e., less power consumption), and less memory usage (model size) than both MIV4 and REL. Thus, compressed CNNs using a small number of convolutional blocks with different activation functions are useful for power-efficient and memory-efficient applications. In the future, better and automatic optimization algorithms will be developed to efficiently find the most effective, power-efficient, and memory-efficient CMIV4 models. Other compressed multi-function CNNs, such as compressed multi-function ResNets and compressed multi-function DenseNets with a reduced number of convolutional blocks using different activation functions, will be developed to increase classification accuracy, reduce training and testing times, reduce computational power, and reduce memory usage (model size) for industrial applications in IoT, big data mining, green computing, etc.

References

[1] LeCun, Y., Bengio, Y. & Hinton, G.E. (2015) Deep learning. Nature 521, pp. 436–444.

[2] Krizhevsky, A., Sutskever, I. & Hinton, G.E. (2012) Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pp. 1097–1105. Cambridge, MA: MIT Press.

[3] He, K., Zhang, X., Ren, S. & Sun, J. (2016) Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778.

[4] Esteva, A., Kuprel, B., Novoa, R.A., Ko, J., Swetter, S.M., Blau, H.M. & Thrun, S. (2017) Dermatologist-level classification of skin cancer with deep neural networks. *Nature* **542**(7639):115–118.

[5] Nugraha, B.T, Su, S.-F. & Fahmizal, F. (2017) Towards self-driving car using convolutional neural network and road lane detector. In Proceedings of the 2nd International Conference on Automation, Cognitive Science, Optics, Micro Electro-Mechanical System, and Information Technology (ICACOMIT), pp. 65–69.

[6] Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Driessche, G.V.D., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T. & Hassabis, D. (2016) Mastering the game of Go with deep neural networks and tree search. Nature (529), pp 484–503.

[7] Fukushima, K. (1979) Neural network model for a mechanism of pattern recognition unaffected by shift in position-Neocognitron. *Transactions of the IECE* **J62-A**(10):658–665.

[8] LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. (1998) Gradient-based learning applied to document recognition. *Proc. IEEE* **86**(11):2278–2324.

[9] *Large Scale Visual Recognition Challenge 2012 (ILSVRC2012)* (2012) [Online.] Available: http://www.image-net.org/challenges/LSVRC/2012/results.html.

[10] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed S., Anguelov D., Erhan, D., Vanhoucke, V. & Rabinovich, A. (2015) Going Deeper with Convolutions. In Proceedings of 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–9.

[11] *Large Scale Visual Recognition Challenge 2014 (ILSVRC2014)* (2014) [Online.] Available: http://www.image-net.org/challenges/LSVRC/2014/.

[12] *Large Scale Visual Recognition Challenge 2015 (ILSVRC2015)* (2015) [Online.] Available: http://www.image-net.org/challenges/LSVRC/2015/index.

[13] He K. (2016) Deep Residual Networks - Deep Learning Gets Way Deeper. [Online.] Available: https://icml.cc/2016/tutorials/icml2016_tutorial_deep_residual_networks_kaiminghe.pdf.

[14] *COCO 2015 Object Detection Task* (2015) [Online.] Available: http://cocodataset.org/#detection-2015.

[15] Szegedy, C., Ioffe, S., Vanhoucke, V. & Alemi, A. (2017) Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17), pp. 4278–4284.

[16] Pleiss G., Chen D., Huang G., Li T., Maaten L. v. d., Weinberger K. Q. (2017) Memory-Efficient Implementation of DenseNets. [Online.] Available: https://arxiv.org/abs/1707.06990

[17] Sun B., Yang L., Dong P., Zhang W., Dong J., Young C. (2018) Ultra Power-Efficient CNN Domain Specific Accelerator with 9.3TOPS/Watt for Mobile and Embedded Applications. [Online.] Available: https://arxiv.org/abs/1805.00361

[18] *OASIS Brains Datasets*. [Online]. Available: http://www.oasis-brains.org/#data.

[19] Marcus, D.S., Wang, T.H., Parker, J., Csernansky, J.G., Morris, J.C. & Buckner, R.L. (2007) Open access series of imaging studies (oasis): cross-sectional MRI data in young, middle aged, nondemented, and demented older adults. Journal of cognitive neuroscience 19(9), 1498–1507.

[20] Krizhevsky, A. & Hinton G. (2009) Learning Multiple Layers of Features from Tiny Images. Technical report, University of Toronto 1 (4), 7.