Finding the Secret of CNN Parameter Layout under Strict Size Constraint

Lixin Liao
Institute of Information Science, Beijing Jiaotong University, Beijing Key Laboratory of Advanced Information Science and Network Technology, Beijing, China 100044

Yao Zhao
Institute of Information Science, Beijing Jiaotong University, Beijing Key Laboratory of Advanced Information Science and Network Technology, Beijing, China 100044

Shikui Wei*
Institute of Information Science, Beijing Jiaotong University, Beijing Key Laboratory of Advanced Information Science and Network Technology, Beijing, China 100044
shkwei@bjtu.edu.cn

Jingdong Wang
Microsoft Research Asia
Beijing, China 100044

Ruoyu Liu
Institute of Information Science, Beijing Jiaotong University, Beijing Key Laboratory of Advanced Information Science and Network Technology, Beijing, China 100044

ABSTRACT

Although deep convolutional neural networks (CNNs) have significantly boosted the performance of many computer vision tasks, their complexities (the size or the number of parameters) are also dramatically increased even with slight performance improvement. However, the larger network leads to more computation requirements, which are unfavorable to resource-constrained scenarios, such as the widely used embedded systems. In this paper, we tentatively explore the essential effect of CNN parameter layout, *i.e.*, the allocation of parameters in the convolution layers, on the discriminative capability of CNN. Instead of enlarging the breadth or depth of networks, we attempt to improve the discriminative ability of CNN by changing its parameter layout under strict size constraint. Toward this end, a novel energy function is proposed to represent the CNN parameter layout, which makes it possible to model the relationship between the allocation of parameters in the convolution layers and the discriminative ability of CNN. According to extensive experimental results with plain CNN models and Residual Nets, we find that the higher the energy of a specific CNN parameter layout is, the better its discriminative ability is. Following this finding, we propose a novel approach to learn the better parameter layout. Experimental results on two public image classification datasets show that the CNN models with the learned parameter layouts achieve the better image classification results under strict size constraint.

KEYWORDS

Convolutional Neural Network; Network Layout; Parameters

ACM Reference format:
Lixin Liao, Yao Zhao, Shikui Wei, Jingdong Wang, and Ruoyu Liu. 2017. Finding the Secret of CNN Parameter Layout under Strict Size Constraint. In *Proceedings of MM'17, October 23–27, 2017, Mountain View, CA, USA., ,* 9 pages.
DOI: https://doi.org/10.1145/3123266.3123346

1 INTRODUCTION

Starting from the amazing debut on 2012 ImageNet competition [20], deep CNNs have been developed greatly in the last few years, especially on image representation learning. Totally different from the traditional hand-craft features [18, 26, 28, 38–40], the discriminative CNN features are learned automatically by the deep convolutional neural networks. Employing CNNs, lots of computer vision tasks, such as image retrieval [1, 24], image labeling [41], semantic segmentation [25, 42] and object detection [30, 31], have achieved unprecedented performance.

Generally speaking, most researches on deep CNNs [12, 20, 22, 23, 32, 35] have paid much attention to boosting discriminative capability of network architectures, but they didn't fully take into account the constrained conditions in the practical scenarios, such as the limitations in computational resources, memory usage, and real-time capability. For example, the outstanding AlexNet [20] was significantly deeper than the classic LeNet [22], which employed a new

*Shikui Wei is the corresponding author.

MM'17, October 23–27, 2017, Mountain View, CA, USA.

DOI: https://doi.org/10.1145/3123266.3123346

activating function (*i.e.*,ReLu) [27] to avoid the problem of vanishing gradients in deep CNNs. Although AlexNet outperforms the traditional shallow networks remarkably, it has to learn a great deal of parameters. The number to be learned is up to 60 million. Furthermore, Simonyan *et al.* [32] proposed the VGG Net, in which there were 19 layers. In spite of employing smaller convolution filters (*i.e.*,3×3 size), VGG Net owned more parameters than AlexNet. To control the computational cost, Szegedy *et al.* [35] proposed a more powerful network, called GoogLeNet, which was carefully designed by increasing the depth and breadth of the network. To address the convergence problem in the extremely deep CNN, He *et al.* [12] reformulated the layers as learning residual functions with reference to the layer inputs, which led to a more powerful but extremely deep convolutional neural network. In brief, most of existing CNN models with the same network structure attempt to boost the performance by enlarging the breadth or depth of the networks. However, the larger network gives rise to more computational cost and higher memory usage, which are impracticable in many resource-constrained scenarios like widely used embedded systems.

To deal with this issue, many recent researches have focused on developing economic and efficient CNNs. Canziani *et al.* [3] presented a comprehensive analysis of important metrics among those outstanding networks mentioned above, such as parameters, operations count, and the inference time. Moreover, some approaches [9, 10, 15, 16, 29] about the topic of network compression have been reported. For example, Han *et al.* [10] proposed deep compression, which can reduce the memory usage of the existed CNNs by a series of processes with no loss of accuracy. However, we work in a different way. In our work, we improve the discriminative ability of CNN under strict size constraint.

In this paper, we attempt to design CNN models from a new perspective, *i.e.*, the parameter layout of CNN models under strict size constraint. The parameter layout of CNN refers to the allocation of parameters in the convolution layers. Instead of changing the network scale, we enhance the discriminative ability of CNN meanwhile controlling the total amount of parameters in the convolution layers. In essence, for the CNN models with the same type of network structure, the number of parameters is directly proportional to the computational cost and memory usage of the network. Hence, we aim to improve the discriminative ability of CNN meanwhile remain its computational cost and memory usage. Toward this end, we propose an energy function to model the relationship between the allocation of parameters and the discriminative ability of CNN. In the energy function, each feature map in a specific convolution layer is treated as the indeterminate symbol sent by the information source, and the energy of CNN parameter layout is the product of the information entropies of the convolution layers. According to extensive experiments, we find that the higher the energy of a specific CNN parameter layout is, the better its discriminative ability is. Following this finding, we propose a novel approach to learn the better CNN parameter layouts under strict parameters constraint. Experimental results show that the learned CNN parameter layouts maintain the proportional trend and achieve the best image classification results, compared with other parameter layouts under strict size constraint.

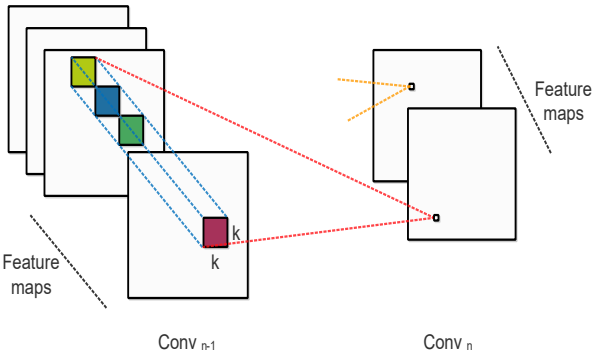

Figure 1: The convolution operation in the convolution layers. *Conv* represents the convolution layer. The convolutional kernels with different colors indicate that the number of feature maps in the previous convolution layer ($Conv_{n-1}$) will directly influence the number of parameters of the current convolution layer ($Conv_n$).

The main contributions of this work are summarized as follows:

(1) We discover the relationship between the allocation of parameters in the convolution layers and the discriminative ability of CNN. It provides a new perspective for understanding the inner work mechanism of CNN.
(2) We propose an energy function to measure the discriminative ability of CNN when the total amount of parameters in the convolution layers remains invariant. Feature maps in the corresponding convolution layer can be regraded as the possible outcomes, and they are in a state of total uncertainty as the convolution kernels are randomly initialized. Hence the information entropy of each convolution layer is the biggest as the probabilities of feature maps are same. The energy function is a product of the information entropies of the convolution layers. To the best of our knowledge, it is the first time that an energy function is proposed to model the relationship between the allocation of parameters and the discriminative ability of CNN.
(3) The energy function can be used to change the existed CNNs or design a new CNN, which will be beneficial to many computer vision problems.

2 RELATED WORK

Convolutional neural networks have been greatly developed, and lots of outstanding networks are proposed to deal with the specific problems of computer vision. Since this paper mainly focuses on the CNN parameter layout, we only introduce some representative works from three aspects, the convolution kernel, the network architecture and the network compression.

2.1 Convolutional Kernel

Since the convolution operation is the critical step in CNN, different sizes of convolution kernels will directly affect the performance of CNN. Lots of works [32, 44] had pointed out that the smaller convolution kernels would benefit the performance of CNN. Karen *et al.* [32] took advantage of this property. They set all convolution kernels as very small (3×3) convolution kernels. In fact, a stack of two convolution layers with (3×3) convolution kernels has a more effective field than that of one convolution layer with (5×5) convolution kernels. More importantly, it has much more nonlinearity and fewer parameters. Residual Net [12] followed the rule of convolution kernels and took advantage of (3×3) convolution kernels. On the other hand, Szegedy *et al.* [32] invented the inception model, which concatenated the feature maps computed by ($1 \times 1, 3 \times 3, 5 \times 5$) convolution kernels, to capture more detailed information.

In addition to the convolution operation at fixed locations of the feature map, the deformable convolution [5] which added 2D offsets on the feature map can enhance CNN's capacity of modeling geometric transformation.

2.2 Network Architecture

The neural network is not a new research area, which has been developed for nearly half a century. Cybenko [4] and Hornik *et al.* [13] had pointed out that neural networks with only one hidden layer can describe any bounded continuous function, while the universal approximation property requires an exponential number of neurons. Hence, deepening the neural network is a feasible and promising direction [2, 7, 11]. LeNet [22] was the starting point of the convolutional neural network, which gained satisfactory performance on the document recognition. AlexNet [20] had five convolution layers, which was deeper than LeNet. Following this way, VGG Net [32] and GoogLeNet [35] reached 19-layer and 22-layer respectively. While deeper neural networks were more difficult to train, Srivastava *et al.* [34] proposed Highway Network which allowed the unimpeded information flow across several layers on information highways. Moreover, He *et al.* [12] reformulated the layers as learning residual functions with reference to the layer inputs and built 152-layer Residual Net.

Recent works [21, 36, 37, 45] pointed out that a Residual Net is a mixture of many dependent networks. Andreas *et al.* [36] interpreted a Residual Net as a collection of many paths of differing length, and revealed these paths to show ensemble-like behavior in the sense which does not strongly depend on each other. On the other hand, there are also some works [10, 14, 43] about changing the layout of Residual Net. Han *et al.* [10] proposed a pyramidal network structure in which the number of feature maps is gradually increased.

2.3 Network Compression

Deep CNNs lead to huge computational cost and memory usage, which are impracticable in many resource-constrained scenarios like widely used embedded systems. To address this problem, a straightforward way is the network compression which aims to develop the economic and efficient CNNs. Lots of approaches [9, 10, 15, 29] about the topic have been proposed. Han *et al.* [10] employed pruning, trained quantization, and Huffman coding to compress neural networks, and gained the great decrease of the storage requirement. Recently, Han *et al.* [9] further designed a hardware accelerator called EIE [9] based on the deep compression. Moreover, Hubara *et al.* [15] quantized weights and activations to reduce memory size and accesses, and used bit-wise operations to replace most arithmetic operations. Different from these methods that compress the existed models, Iandola [16] invented a squeeze convolution layer to construct SqueezeNet that had fewer parameters but equivalent accuracy compared with AlexNet.

There are also some other works that aim to improve the discriminative ability of CNN, such as batch normalization [17], dropout [33], adaptive gradient methods [6]. Since the points of views on improving the discriminative ability of CNN are different from ours, we do not give more details.

3 THE ENERGY FUNCTION CONNECTING CNN PARAMETER LAYOUT AND DISCRIMINATIVE CAPABILITY

In order to model the relationship between the allocation of parameters and the discriminative ability of CNN, we design the energy function inspired by the information entropy. In this section, we first introduce the convolutional neural network in brief. Then, we introduce the details of the energy function. Finally, we transform the energy function into the objective function and propose a novel approach to learn the better CNN parameter layouts.

3.1 Convolutional Neural Network

Convolutional neural networks have significantly boosted the performance of many computer vision tasks. In fact, since the first convolutional neural network, *i.e.*, LeNet, was proposed, the basic building components of CNNs have not changed much. AlexNet also contains the convolution layer, the pooling layer, and the fully-connected layer, as LeNet does. Here, we will introduce the convolutional neural network via introducing AlexNet.

AlexNet contains five convolution layers, three pooling layers, and two fully-connected layers. Different building components have different functions on the network. The convolution layers are the vital components of CNN, which

extract information from the original inputs. The pooling layers are mainly used to reduce the network complexity. At the end of the network, the fully-connected layers are used to integrate information for classification. It is clear to find that the parameters in the convolution layers play a key role in the performance of the network. We take AlexNet as one example to compute the number of parameters in the convolution layers. Note that the parameters do not include the parameters in the fully-connected layers.

To evidently describe AlexNet, we first declare some variables. The input of AlexNet has n_0 channels. The size of the convolution kernel in the feature map is $k \times k$, as shown in Fig. 1. The number of convolution kernels in the i^{th} convolution layer is n_i. In the following sections, we will use the form $(n_1, n_2, ..)$ to represent the network's parameter layout. The input should be forwardly propagated in the order of architecture. The detailed information of the convolution operation is shown in Fig. 1. We can clearly notice that the number of feature maps in the previous convolution layer is identical to the number of convolution kernels in the current convolution layer. It indicates that the number of convolution kernels in the previous convolution layer will directly influence the number of parameters in the current convolution layer. Hence, the number of parameters in the i^{th} convolution layer is $k \times k \times n_{i-1} \times n_i$. Now we compute the number of parameters of AlexNet by the following equation:

$$\begin{aligned} P_{all} &= k \times k \times n_0 \times n_1 + k \times k \times \sum_{i=2}^{N} n_{i-1} \times n_i \\ &= k^2 \times (n_0 \times n_1 + \sum_{i=2}^{N} n_{i-1} \times n_i), \end{aligned} \tag{1}$$

where P_{all} is the total amount of parameters in AlexNet, and N represents the number of convolution layers in AlexNet.

3.2 Definition of the Energy Function

Our aim is to find a function which can model the relationship between the allocation of parameters in the convolution layers and the discriminative ability of CNN. Using the function, we can provide a new perspective for configuring the optimal CNN parameter layout under strict size constraint.

Inspired by the information theory, we find that the feature maps of the corresponding convolution layer can be regarded as the possible outcomes of an information source. According to this assumption, we can use the information entropy to measure the energy of the convolution layer. The random initialization of training CNN makes these outputs in a state of total uncertainty. Hence, the information entropy of the convolution layer is the biggest as each feature map has the same probability. It is formulated as follows:

$$I_{n_i} = -\sum_{j=1}^{n_{ij}} \frac{1}{n_{ij}} log(\frac{1}{n_{ij}}) = log(n_i), \tag{2}$$

where n_{ij} represents the j^{th} feature map of the i^{th} convolution layer.

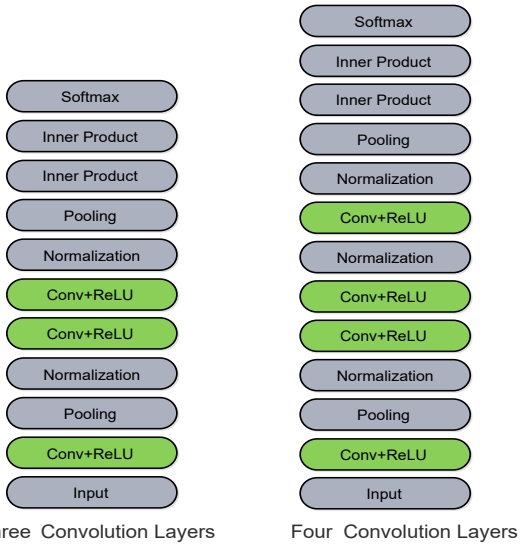

Figure 2: The architectures of two types of Plain CNN models in the experiments. One type contains three convolution layers, and the other contains four convolution layers.

From Eq. 1 and Fig. 1, we can clearly find out that the number of feature maps in the previous convolution layer has the multiplicative effect on the size of convolution kernels in the current convolution layer. In our energy function, we do the multiply operation between information entropies of two adjacent convolution layers. Note that the energy function holds just when the total amount of parameters remains invariant. As [10] has pointed out that the network with the gradually increased number of convolution kernels in convolution layers has the better discriminative ability, we further add the constraint. The energy function of a CNN model with N convolution layers can be formulated as follows:

$$\begin{aligned} I_N &= \prod_{i=1}^{N} I_{n_i}, i = 1, .., N \\ s.t. &\sum_{i=1}^{N} n_{i-1} \times n_i = C, n_i < n_{i+1}, \end{aligned} \tag{3}$$

where C is the constant value that represents the number of parameters.

In the section of experiments, extensive experiments demonstrate that the energy of CNN is proportional to the discriminative ability of CNN. It motivates us that the energy function can be transformed into the objective function with some constraints. In the next section, we will give more details.

3.3 Learn the Better CNN Parameter Layout

As we have designed the energy function to model the relationship between the allocation of parameters in the convolution

layers and the discriminative ability of CNN, we can take advantage of the energy function to learn the better CNN parameter layout. In the section of experiments, we have validated that the energy of CNN parameter layout is proportional to the discriminative ability of CNN. It means that the higher the energy of a specific CNN parameter layout is, the better its discriminative ability is. Hence, finding the better CNN parameter layout is transformed into finding the biggest energy. The objective function is formulated as follows:

$$\begin{aligned} &\max \prod_{i=1}^{N} I_{n_i}, i = 1, .., N \\ &s.t. \sum_{i=1}^{N} n_{i-1} \times n_i = C, n_i < n_{i+1}. \end{aligned} \tag{4}$$

Once solving the objective function, we can find the better CNN parameter layout. However, when the number of convolution layers is increasing, the time consuming of resolving this objective function is unacceptable. Due to the objective function is an NP-hard problem, it will be difficult for us to find the better parameter layout when the CNN is too deep, especially for the residual network. Here we give one simplified solution for finding a sub-optimal parameter layout of the residual network. The size of convolutional kernels in the regular building block unit of ResNet [12] is the same. We can simplify the objective function by taking the regular building block unit as one convolution layer. It means that we also keep the size of convolution kernels in the regular building block unit fixed. In this way, the number of "convolution layers" in ResNet will sharply decrease. And the time consuming of resolving the objective function will be reduced greatly.

4 EXPERIMENTS

In order to validate the energy function, we conduct a series of experiments with plain CNN models and Residual Nets on two image classification datasets. These experiments are mainly divided into three parts. In the first part, we evaluate two types of plain CNN models under strict size constraint. Their parameter layouts are different, and the total number of parameters is fixed at 4,288 and 66,304 respectively. In the second part, we evaluate 32-layer Residual Nets with different parameter layouts when total number of parameters is fixed at 51,248. Note that the total number of parameters in these experiments are divided by the constant value (3×3). In the third part, we evaluate CNN models with the learned better parameter layouts under the corresponding circumstances.

4.1 Experiment Setup

CIFAR-10 and CIFAR-100: CIFAR-10 [19] dataset consists of 10 classes of colored images, while the CIFAR-100 [19] dataset is made up of 100 classes. For each image in both datasets, it contains 32×32 pixels. Generally speaking, the images of CIFAR-100 are much more variable than them of CIFAR-10. They are all divided into two parts, *i.e.*, the training set (50,000 images) and the test set (10,000 images).

Table 3: The architecture of Residual Net in the experiments. n_i represents the number of convolution kernels in the i_{th} residual block unit.

Residual Block Unit	Block Type
`Conv1`	$\begin{bmatrix} 3 \times 3, & n_1 \end{bmatrix}$ x 1
`Conv2`	$\begin{bmatrix} 3 \times 3, & n_2 \end{bmatrix}$ x 10
`Conv3`	$\begin{bmatrix} 3 \times 3, & n_3 \end{bmatrix}$ x 10
`Conv4`	$\begin{bmatrix} 3 \times 3, & n_4 \end{bmatrix}$ x 10

Configuration and Training Setting of plain CNN Models: The configurations of two types of plain CNN models are shown in Fig. 2. One type of models contains three convolution layers, and the other kind of models contains four convolution layers. These models are not employed with the dropout technique, which just include the convolution operation, the ReLU activation, the pooling operation and the regular batch normalization. For each type of CNN models, their parameter layouts are different, but their number of parameters are all equivalent to 4,288 or 66,304. In this way, we can focus on evaluating the relationship between the allocation of parameters and the discriminative ability of CNN.

All these CNN models are implemented on Tensorflow which is an open-source software. The initial learning rate is set to 0.1, decayed by a factor of 0.1 based on the number of steps. The batch size is set to 128, and the max training step is 40,000. They are trained by Stochastic Gradient Descent on CIFAR-10 and CIFAR-100.

Configuration and Training Setting of Residual Nets: Residual Nets on two datasets have the same architecture with only different parameter layouts. The architecture of Residual Net is shown in Tab. 3. There are 31 convolution layers in four kinds of residual block units and one pooling layer. The first residual block unit contains one convolution layer. As for the second, third and fourth residual block units, each of them contains ten convolution layers. The total number of parameters is 51,248.

These Residual Nets are also implemented on Tensorflow. The initial learning rate is set to 0.1 and reduced to 0.01 when the training step is up to 60,000. The batch size is set to 128, and the max training step is 80,000. They are also trained by Stochastic Gradient on CIFAR-10 and CIFAR-100.

4.2 Evaluations on the Energy Function with Plain CNN Models

In order to validate the relationship between the allocation of parameters in the convolution layers and the discriminative ability of CNN, we conduct experiments with plain CNN models on CIFAR-10 and CIFAR-100. The configurations of these CNN models with different allocations of 4,288 and 66,304 parameters in the convolution layers are shown in Tab. 1 and Tab. 2. Note that if the number of qualified CNN parameter layouts is over 32, we randomly select 32 samples

Table 1: Plain CNN models with different parameter layouts under 4,288 parameters. *Convs* represents the convolution layers, and *Number* represents the number of models with different parameter layouts.

Type/Number	Different Parameter Layouts
Three Convs/24	(8,8,525),(8,13,320),(8,26,156),(8,41,96),(8,52,74),(11,23,174),(11,37,104),(14,22,179) (16,16,249),(16,20,196),(16,40,90),(16,53,64),(17,19,206),(18,29,128),(20,28,131),(21,25,148) (24,31,112),(24,34,100),(32,32,99),(34,46,57),(35,47,54),(36,38,74),(36,44,59),(36,43,58)
Three Convs/32	(8,14,24,159),(8,16,44,78),(8,18,40,86),(8,20,38,88),(8,20,38,88),(8,21,32,107),(8,23,48,62) (8,25,32,102),(8,27,46,61),(8,33,40,67),(8,38,44,52),(9,27,49,55),(9,29,40,71),(10,23,38,83) (10,31,42,63),(11,17,36,96),(11,35,43,55),(12,21,32,104),(12,28,44,64),(13,19,46,68),(13,28,37,77) (14,29,40,67),(14,33,43,55),(14,33,44,53),(15,22,43,69),(16,22,36,86),(16,32,39,65),(16,34,44,50) (17,29,36,75),(18,25,44,61),(19,26,37,75),(20,34,44,51),(23,29,37,67)

Table 2: Plain CNN models with different parameter layouts under 66,304 parameters. *Convs* represents the convolution layers, and *Number* represents the number of models with different parameter layouts.

Type/Number	Different Parameter Layouts
Three Convs/32	(10,26,2539),(16,101,640),(16,164,388),(17,19,3470),(17,209,300),(18,50,1307),(18,106,607) (28,35,1864),(28,44,1477),(28,77,832),(28,110,574),(28,140,445),(28,154,402),(34,79,804) (34,158,385),(36,52,1237),(37,37,1752),(47,109,560),(50,62,1017),(50,97,632),(50,194,291) (55,59,1066),(56,56,1125),(58,170,331),(64,64,969),(69,157,352),(100,116,469),(104,113,480) (128,128,387),(128,160,284),(148,178,222),(148,185,208)
Four Convs/32	(10,34,66,965),(11,53,69,899),(12,132,157,280),(12,133,188,211),(15,29,88,719),(15,29,136,455) (16,159,181,193),(18,81,104,542),(22,25,42,1539),(22,37,87,715),(25,113,121,411),(28,70,153,350) (28,81,112,490),(36,71,172,299),(38,58,107,540),(38,109,112,445),(42,67,124,444),(46,79,81,693) (46,111,142,319),(47,61,64,928),(55,59,118,474),(56,81,175,271),(64,76,96,562),(64,88,189,232) (65,125,128,328),(66,83,92,576),(68,85,145,331),(74,103,148,292),(78,145,148,225),(88,91,93,533) (98,100,146,285),(122,131,138,231)

Table 4: Residual Nets with different parameter layouts under 51,248 parameters. Each parameter layout generates the energy by the energy function.

Residual Nets	Energy
(1,11,42,59)	0
(1,27,34,59)	0
(5,8,17,72)	40.551
(5,10,27,68)	51.536
(7,16,34,63)	78.825
(7,24,25,65)	83.096
(8,22,48,50)	97.341
(13,20,35,61)	112.304
(15,16,37,61)	111.4536
(15,32,38,53)	135.5464
(16,16,32,64)	110.0631
(19,34,41,49)	150.0631
(20,28,42,52)	147.4244

from them. These models with different allocations of parameters have the corresponding energies computed by Eq. 3. We plot to scatter diagrams with the energy and the accuracy of plain CNN models, which are shown in Fig. 3 and Fig. 4, respectively. In these scatter diagrams, the y-coordination is the top-1 accuracy on image classification datasets, and the x-coordination is the energy of the corresponding plain CNN model.

From Fig. 3 and Fig. 4, it is easy to find a common trend that the energy of the CNN parameter layout is proportional to its discriminative ability. The accuracy of CNN will go higher while its energy becomes bigger. From the blue circle points in a of Fig. 3, we find that the accuracy of CNN with the largest energy on CIFAR-10 is 2.5% higher than that with the smallest energy. In c of Fig. 3, we find that the accuracy of CNN with the largest energy is almost 3% higher than that with the smallest energy. These results indicate that the higher the energy of a specific CNN parameter layout is, the better its discriminative ability is. Moreover, they also reveal that the energy function correctly models the relationship between the allocation of parameters in the convolution layers and the discriminative ability of CNN. It is worth noting that the input image size of CIFAR-10/CIFAR-100 is only 32×32. In fact, we have also evaluated the proposed approach on a high-resolution image dataset, i.e., PascalVOC 2007 [8]. The experimental results demonstrate consist conclusions with these obtained from CIFAR-10/CIFAR-100. For example, the classification accuracy of the plain CNN model with the

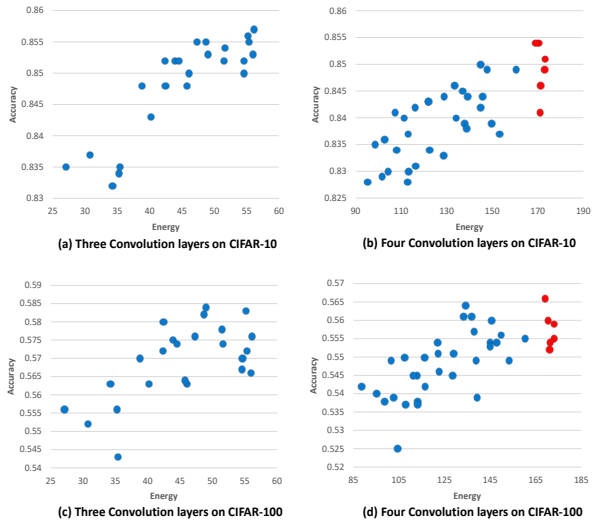

Figure 3: Experimental results on two types of plain CNN models under 4,288 parameters. The blue circle points are the results of randomly selected CNN parameter layouts, and the red circle points are the results of learned CNN parameter layouts. The results in the first row are on CIFAR-10, and these in the second row are on CIFAR-100.

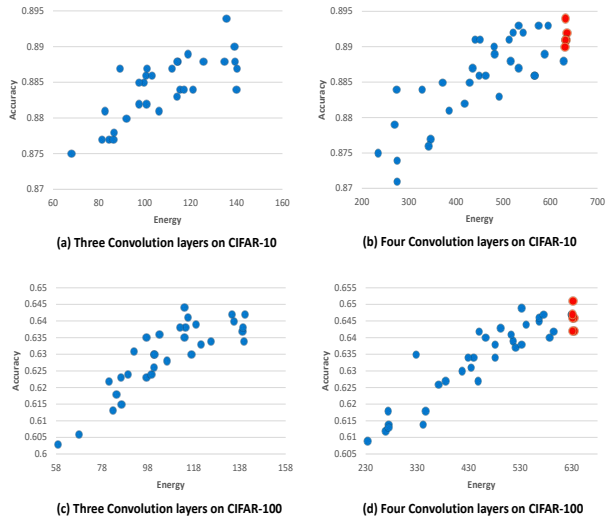

Figure 4: Experimental results on two types of plain CNN models under 66,304 parameters. The blue circle points are the results of randomly selected CNN parameter layouts, and the red circle points are the results of learned CNN parameter layouts. The results in the first row are on CIFAR-10, and these in the second row are on CIFAR-100.

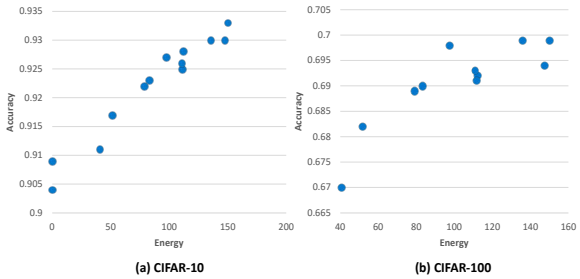

Figure 5: Experimental results on Residual Nets with different parameter layouts. The blue circle points are the results of different parameter layouts. (a) demonstrates the results on CIFAR-10, and (b) demonstrates the results on CIFAR-100.

energy 173.03 is almost higher 2% than that with the energy 95.05 on the PascalVOC 2007 dataset.

We notice that the accuracies of CNN models with approximate energies are not same, which is related to the training mechanism of CNN. That is, the trained CNN models with approximate energies are not guaranteed to find the same local optimal values. However, the differences among the accuracies are very small, and the trend between the energy of CNN and its discriminative ability is still obvious. Hence, the relationship modeled by the energy function is reliable.

4.3 Evaluations on the Energy Function with Residual Nets

In order to further evaluate the energy function, we conduct experiments on CIFAR-10 and CIFAR-100 with deeper Residual Nets. Different allocations of 51,248 parameters in the convolution layers are shown in the left column of Tab. 4, and there are 12 different parameter layouts when the total amount of parameters in convolution layers and the network structure remain invariant. The energies of Residual Nets are computed by Eq. 3, shown in the right column in Tab. 4. To demonstrate the relationship between the energy and the discriminative ability of Residual Nets more evidently, we plot scatter diagrams with the energy and the discriminative ability of Residual Nets like the above section. In Fig. 5, a scatter diagram depicts the relationship between the energy and the performance of Residual Nets on CIFAR-10, and b scatter diagram depicts the relationship on CIFAR-100.

From the scatter diagrams on two datasets in Fig. 5, we can clearly notice that the energy of Residual Net is proportional to its discriminative ability. The accuracies of upper right blue circle points are all 3% higher than these of lower left points on CIFAR-10 and CIFAR-100. It indicates that the higher the energy of a specific Residual Net parameter layout is, the better its discriminative ability is. This conclusion is consistent with that on the plain CNN models. It further reveals that the energy function correctly models the relationship between the allocation of parameters in different convolution layers and the discriminative ability of CNN.

Table 5: Plain CNN models with the top six energies under 4,288 and 66,304 parameters respectively.

Parameter Number	Six Learned Best Parameter Layouts
4,288	(35,36,37,43),(33,35,37,47) (35,37,38,39),(32,36,38,44) (30,31,38,55),(28,29,32,77)
66,304	(135,136,137,211),(133,135,137,215) (124,125,128,269),(137,141,142,187) (123,125,128,270),(128,130,140,222)

4.4 Evaluations on the Energy Function with the learned CNN parameter Layouts

In order to validate effectiveness of the learned CNN layouts by the energy function, we conduct experiments with plain CNN models on CIFAR-10 and CIFAR-100. As we have claimed in the previous section, if the number of qualified CNN parameter layouts is over 32, we randomly select 32 samples from them. In addition, if the number is smaller than 32, the best layout in the previous sections will be directly employed. Similarly, we conduct experiments on plain CNN models with four convolution layers under 4,288 parameters and 66,304 parameters. Through Eq. 4, we obtain CNN models with the top six energies for avoiding the effect of randomness in CNN training. These models are shown in Tab. 5. To be more intuitive to discriminate the learned CNN parameter layouts, we plot their energies and accuracies with red circle points to the corresponding scatter diagram in Fig. 3 and Fig. 4.

From the scatter diagrams in Fig. 3 and Fig. 4, we can clearly see that the accuracies of CNN models with the top six energies are located at the upper right corner of the corresponding scatter diagram. We can clearly notice the improvements of 4,288 parameters in Fig. 3, as there are obvious gaps of the energies among the randomly selected parameter layouts and the learned parameter layouts. Though the gaps of the energies are relatively small in Fig. 4, one of the accuracies of learned parameter layouts is higher than the others. These results show that the energy function can be used to guide the learning of the better CNN layout, which further validate that the energy function correctly models the relationship between the allocation of parameters in different convolution layers and the discriminative ability of CNN.

5 CONCLUSION

In this work, we attempt to improve the discriminative ability of CNN under strict size constraint. Instead of enlarging the breadth or depth, we tentatively optimize the CNN model by changing the parameter layout in the convolution layers. To this end, we propose an energy function to model the relationship between the allocation of parameters in the convolution layers and the discriminative ability of CNN. According to the information theory, we build the energy function, in which the energy is nearly equivalent to the product of the information entropies of the convolution layers. Extensive experiments with plain CNN models and Residual Nets show that the energy of CNN parameter layout is proportional to the discriminative ability of CNN. Following the finding, we propose a novel approach to learn the better layout, which can guideline the design of the CNN architecture under strict size constraint. Experimental results on two image classification datasets show that the CNN models with the learned parameter layouts achieve the better image classification results.

ACKNOWLEDGMENTS

This work was supported in part by National Natural Science Foundation of China (No.61532005, No.61572065), National Key Research and Development of China (No.2016YFB0800404), Joint Fund of Ministry of Education of China and China Mobile (No.MCM20160102), and the Fundamental Research Funds for the Central Universities (No.2017YJS060).

REFERENCES

[1] Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky. 2014. Neural Codes for Image Retrieval. In *ECCV*.
[2] Yoshua Bengio and Olivier Delalleau. 2011. On the Expressive Power of Deep Architectures. In *International Conference on Algorithmic Learning Theory*.
[3] Alfredo Canziani, Adam Paszke, and Eugenio Culurciello. 2016. An Analysis of Deep Neural Network Models for Practical Applications. *arXiv preprint arXiv:1605.07678* (2016).
[4] George Cybenko. 1989. Approximation by Superpositions of A Sigmoidal Function. *MCSS* 2, 4 (1989), 303–314.
[5] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. 2017. Deformable Convolutional Networks. *arXiv preprint arXiv:1703.06211* (2017).
[6] John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research* 12, Jul (2011), 2121–2159.
[7] Ronen Eldan and Ohad Shamir. 2016. The Power of Depth for Feedforward Neural Networks. *arXiv preprint* (2016).
[8] Mark Everingham, L Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. 2008. The pascal visual object classes challenge 2007 (voc 2007) results (2007). (2008).
[9] Song Han, Xingyu Liu, Huizi Mao, Jing Pu, Ardavan Pedram, Mark A Horowitz, and William J Dally. 2016. EIE: Efficient Inference Engine on Compressed Deep Neural Network. In *Proceedings of the 43rd International Symposium on Computer Architecture*.
[10] Song Han, Huizi Mao, and William J Dally. 2015. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. *arXiv preprint arXiv:1510.00149* (2015).
[11] Johan Hastad. 1986. Almost Optimal Lower Bounds for Small Depth Circuits. In *Proceedings of the eighteenth annual ACM symposium on theory of computing*.
[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *CVPR*.
[13] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 1989. Multilayer Feedforward Networks Are Universal Approximators. *Neural networks* 2, 5 (1989), 359–366.
[14] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. 2016. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993* (2016).
[15] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. 2016. Quantized Neural Networks: Training Neural Networks with Low Precision Weights and Activations. *arXiv preprint arXiv:1609.07061* (2016).
[16] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. 2016. SqueezeNet:

AlexNet-level Accuracy with 50x Fewer Parameters and¡ 0.5 MB Model Size. *arXiv preprint arXiv:1602.07360* (2016).
[17] Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv preprint arXiv:1502.03167* (2015).
[18] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. 2010. Aggregating local descriptors into a compact image representation. In *CVPR*.
[19] Alex Krizhevsky and Geoffrey Hinton. 2009. Learning Multiple Layers of Features from Tiny Images. (2009).
[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet Classification with Deep Convolutional Neural Networks. In *NIPS*.
[21] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. 2016. Fractalnet: Ultra-deep Neural Networks without Residuals. *arXiv preprint arXiv:1605.07648* (2016).
[22] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based Learning Applied to Document Recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.
[23] Min Lin, Qiang Chen, and Shuicheng Yan. 2013. Network in Network. *arXiv preprint arXiv:1312.4400* (2013).
[24] Ruoyu Liu, Yao Zhao, Shikui Wei, Zhenfeng Zhu, Lixin Liao, and Shuang Qiu. 2015. Indexing of CNN Features for Large Scale Image Search. *arXiv preprint arXiv:1508.00217* (2015).
[25] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully Convolutional Networks for Semantic Segmentation. In *CVPR*.
[26] David G Lowe. 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision* 60, 2 (2004), 91–110.
[27] Vinod Nair and Geoffrey E Hinton. 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. In *ICML*.
[28] Florent Perronnin and Christopher Dance. 2007. Fisher kernels on visual vocabularies for image categorization. In *CVPR*.
[29] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. 2016. Xnor-net: Imagenet Classification Using Binary Convolutional Neural Networks. In *ECCV*.
[30] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You Only Look Once: Unified, Real-time Object Detection. In *CVPR*.
[31] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards Real-time Object Detection with Region Proposal Networks. In *NIPS*.
[32] Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-scale Image Recognition. *arXiv preprint arXiv:1409.1556* (2014).
[33] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.
[34] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway Networks. *arXiv preprint arXiv:1505.00387* (2015).
[35] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going Deeper with Convolutions. In *CVPR*.
[36] Andreas Veit, Michael J Wilber, and Serge Belongie. 2016. Residual Networks Behave Like Ensembles of Relatively Shallow Networks. In *NIPS*.
[37] Jingdong Wang, Zhen Wei, Ting Zhang, and Wenjun Zeng. 2016. Deeply-fused Nets. *arXiv preprint arXiv:1605.07716* (2016).
[38] Shikui Wei, Dong Xu, Xuelong Li, and Yao Zhao. 2013. Joint optimization toward effective and efficient image search. *IEEE transactions on cybernetics* 43, 6 (2013), 2216–2227.
[39] Shikui Wei, Yao Zhao, Ce Zhu, Changsheng Xu, and Zhenfeng Zhu. 2011. Frame fusion for video copy detection. *IEEE Transactions on Circuits and Systems for Video Technology* 21, 1 (2011), 15–28.
[40] Shikui Wei, Yao Zhao, Zhenfeng Zhu, and Nan Liu. 2010. Multimodal fusion for video search reranking. *IEEE Transactions on Knowledge and Data Engineering* 22, 8 (2010), 1191–1199.
[41] Yunchao Wei, Wei Xia, Min Lin, Junshi Huang, Bingbing Ni, Jian Dong, Yao Zhao, and Shuicheng Yan. 2016. HCP: A flexible CNN framework for multi-label image classification. *IEEE transactions on pattern analysis and machine intelligence* 38, 9 (2016), 1901–1907.
[42] Zifeng Wu, Chunhua Shen, and Anton van den Hengel. 2016. High-performance Semantic Segmentation Using Very Deep Fully Convolutional Networks. *arXiv preprint arXiv:1604.04339* (2016).
[43] Sergey Zagoruyko and Nikos Komodakis. 2016. Wide Residual Networks. *arXiv preprint arXiv:1605.07146* (2016).
[44] Matthew D Zeiler and Rob Fergus. 2014. Visualizing and Understanding Convolutional Networks. In *ECCV*.
[45] Liming Zhao, Jingdong Wang, Xi Li, Zhuowen Tu, and Wenjun Zeng. 2016. On the Connection of Deep Fusion to Ensembling. *arXiv preprint arXiv:1611.07718* (2016).