Deep Competitive Pathway Networks

Jia-Ren Chang FOLLOWWAR.CS00G@NCTU.EDU.TW
Yong-Sheng Chen YSCHEN@CS.NCTU.EDU.TW
Department of Computer Science, National Chiao Tung University, Hsinchu, Taiwan

Abstract

In the design of deep neural architectures, recent studies have demonstrated the benefits of grouping subnetworks into a larger network. For examples, the Inception architecture integrates multi-scale subnetworks and the residual network can be regarded that a residual unit combines a residual subnetwork with an identity shortcut. In this work, we embrace this observation and propose the Competitive Pathway Network (CoPaNet). The CoPaNet comprises a stack of competitive pathway units and each unit contains multiple parallel residual-type subnetworks followed by a max operation for feature competition. This mechanism enhances the model capability by learning a variety of features in subnetworks. The proposed strategy explicitly shows that the features propagate through pathways in various routing patterns, which is referred to as pathway encoding of category information. Moreover, the cross-block shortcut can be added to the CoPaNet to encourage feature reuse. We evaluated the proposed CoPaNet on four object recognition benchmarks: CIFAR-10, CIFAR-100, SVHN, and ImageNet. CoPaNet obtained the state-of-the-art or comparable results using similar amounts of parameters. The code of CoPaNet is available at: https://github.com/JiaRenChang/CoPaNet.

Keywords: CNN, object recognition, competitive mechanism

1. Introduction

Deep convolutional neural networks (CNNs) have been shown to be highly effective in image classification with large datasets, such as CIFAR-10/100 (Krizhevsky and Hinton, 2009), SVHN (Netzer et al., 2011), and ImageNet (Deng et al., 2009). Improvements in computer hardware and network architectures have made it possible to train deeper and more complex networks.

Network grouping is an efficient technique to improve the accuracy in model learning. The Inception architecture (Szegedy et al., 2015) was proposed to aggregate abundant features via multi-scale subnetworks. In addition, dueling architecture (Wang et al., 2015) in deep reinforcement learning can explicitly exploit subnetworks to represent state value and action advantages. Recently, the Residual Networks (ResNets) (He et al., 2015a, 2016) can be regarded that a residual unit includes an identity shortcut and a residual subnetwork. This approach can alleviate the vanishing gradient problem by bypassing the gradients without attenuation and thus can increase the network depth up to more than 100 layers. As suggested in (Abdi and Nahavandi, 2016; Huang et al., 2016b; Veit et al., 2016), ResNets gains its superior performance by implicitly averaging many subnetworks.

The redundancy problem of ResNets has been raised in (Huang et al., 2016b; Zagoruyko and Komodakis, 2016). Some studies primarily aimed at the improvement of the propaga-

© 2017 J.-R. Chang & Y.-S. Chen.

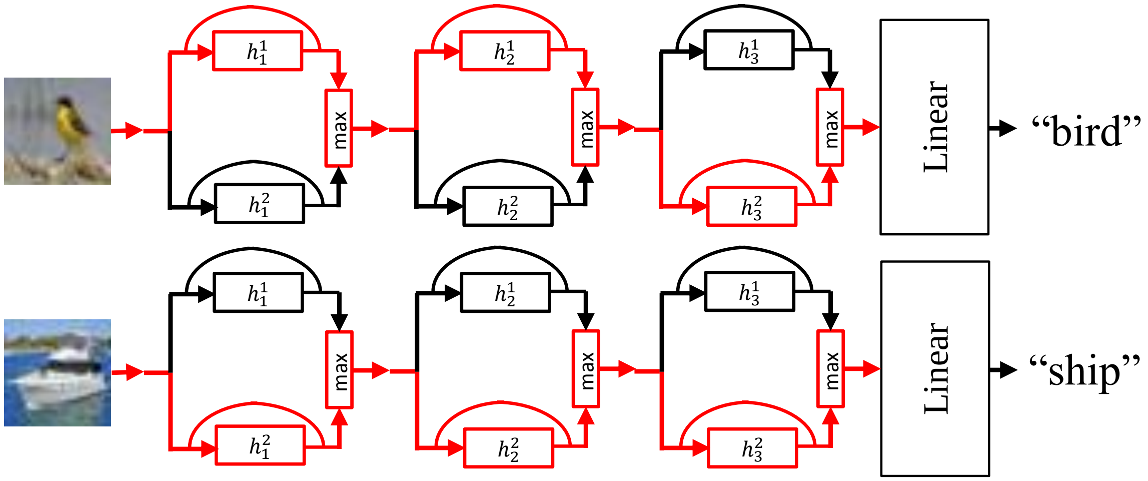


Figure 1: The concept of *pathway encoding* in the proposed architecture. The category information is encoded on the route (red arrows) through which features propagate.

tion in ResNet, thereby reducing the redundancy problem. Stochastic Depth (Huang et al., 2016b) tackled this problem by randomly disabling residual units during training. Wide Residual Networks (Zagoruyko and Komodakis, 2016) addressed this problem by decreasing the depth and increasing the width of residual units for faster training. Both of these network architectures are attempts to shorten the network and thereby improve information back-propagation during training. Without shortening network, a recent work (He et al., 2016) analyzed various usages of rectified linear unit (ReLU) and batch normalization (BN) in ResNets for direct propagation, and proposed methods for identity mapping in residual units to improve training in very deep ResNets.

Some studies encouraged the direct feature reuse by replacing the element-wise addition in ResNets with concatenation. FractalNet (Larsson et al., 2016) repeatedly combines many subnetworks in a fractal expansion rule to obtain large nominal network depth. DenseNet (Huang et al., 2016a) is similar to FractalNet with the difference that DenseNet connects each layer to all of its preceding layers. These approaches exhibit a behavior of mimicking deep supervision, which is important to the learning of discriminative features.

Some studies aimed at the improvement of the residual units by representing the residual function with many tiny subnetworks. Inception-ResNet (Szegedy et al., 2016) presented Inception-type residual units. PolyNet (Zhang et al., 2016) replaces the original residual units with polynomial combination of Inception units for enhancing the structural diversity. Multi-residual networks (Abdi and Nahavandi, 2016) and ResNeXt (Xie et al., 2016) both aggregate residual transformations from many tiny subnetworks.

The idea behind the use of subnetworks is to simplify network for efficient training. By explicitly factoring the network into a series of operations, features can be learned independently. In this work, we embrace this observation and propose a novel deep architecture referred to as Competitive Pathway Network (CoPaNet). Because the underlying mapping function can be decomposed into the maximum of multiple simpler functions and the residual learning (He et al., 2015a) is a good strategy for approximating the mapping functions, the proposed competitive pathway (CoPa) unit was designed to comprise multiple parallel residual-type subnetworks followed by a max operation for feature competition. Furthermore, identity cross-block shortcuts can be added to the CoPaNet to enhance feature reuse. These strategies offer several advantages: 1. Feature redundancy can be reduced by dropping unimportant features through competition. 2. The competitive mechanism facilitates the network to modularize itself into multiple parameter-sharing subnetworks for parameter efficiency (Srivastava et al., 2013). 3. CoPaNet uses residual-type subnetworks and therefore inherits the advantage of ResNet for training very deep network. 4. With competitive mechanism and residual-type subnetworks, the CoPaNet explicitly exhibits the property of pathway encoding, as shown in Figure 1. Because the residual-type subnetwork can preserve feature identity such that the winning path can be traced back within the entire network. That is, the *routing pattern of propagating features* encodes category information. 5. The cross-block shortcuts encourage coarse feature reuse and implicit deep supervision.

CoPaNet was evaluated using several benchmark datasets such as CIFAR-10, CIFAR-100, SVHN, and ImageNet. Our resulting models performed equally to or better than the state-of-the-art methods on the above-mentioned benchmark datasets.

2. Related Work

2.1. Residual Networks (ResNets)

ResNets (He et al., 2015a) are motivated by the counterintuitive observation that the performance of neural networks actually gets worse when developed to a very great depth. This problem can be attributed to the fact that the gradient vanishes when information back-propagates through many layers. He et al. (2015a) proposed skipping some of the layers in convolutional networks through the implementation of shortcut connections, in the formulation of an architecture referred to as residual units. The original residual unit performs the following computation:

$$x_{l+1} = ReLU(id(x_l) + f_l(x_l))\,,$$

where x_l denotes the input feature of the l-th residual unit, $id(x_l)$ performs identity mapping, and f_l represents layers of the convolutional transformation of the l-th residual unit.

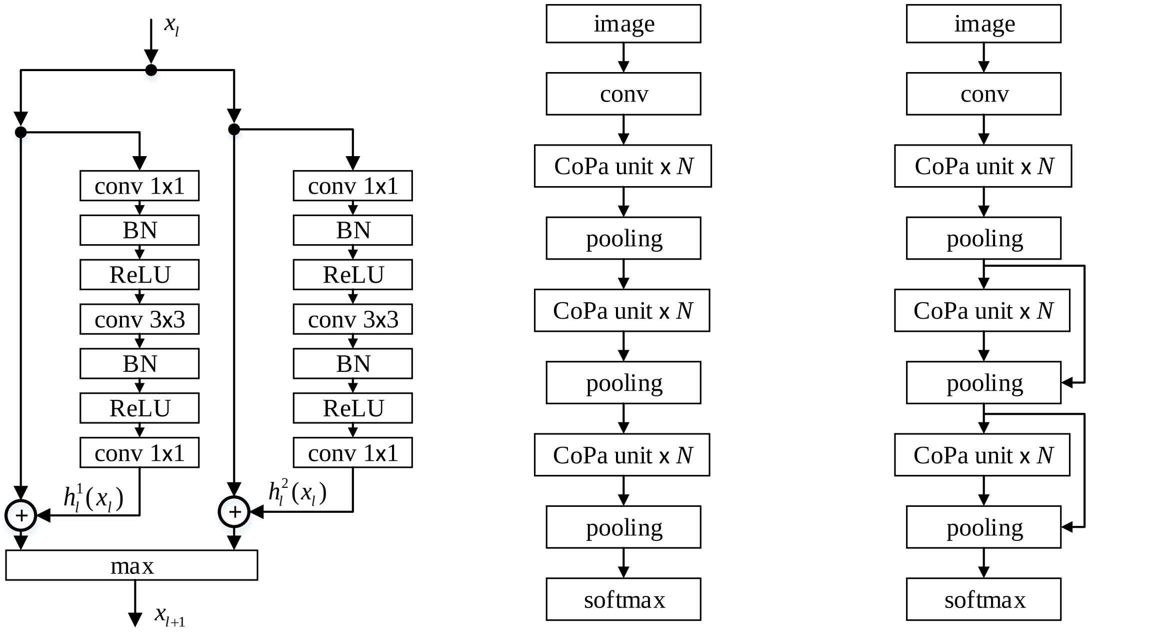
He et al. (2016) further suggested to replace ReLU with another identity mapping, allowing the information to be propagated directly. Thus, they proposed a pre-activation residual unit with the following form:

$$x_{l+1} = id(x_l) + f_l(x_l)\,.$$

Furthermore, the positions of BN and ReLU are changed to allow the gradients to be back-propagated without any transformation. Their experimental results demonstrated the high efficiency of pre-activation residual units.

2.2. Competitive Nonlinear functions

Maxout Networks (Goodfellow et al., 2013) were recently introduced to facilitate optimization and model averaging via Dropout. The authors of this work proposed a competitive



(a)The proposed CoPa unit

(b) The CoPaNet architecture

(c) The CoPaNet-R architecture

Figure 2: The proposed CoPa unit and network architecture.

nonlinearity referred to as maxout, which was constructed by obtaining the maximum across several maxout hidden pieces. Maxout Networks can be regarded as universal approximators and can provide better gradient back-propagation than other activation functions. Without down-sampling the features, Local Winner-Take-All (LWTA) (Srivastava et al., 2013) was inspired by the characteristics of biological neural circuits. Each LWTA block contains several hidden neurons and produces an output vector determined by local competition between hidden neurons activations. Only the winning neuron retains its activation, whereas other hidden neurons are forced to shut off their activation. In empirical experiments, both network architectures have been shown to have advantages over ReLU.

3. Competitive Pathway Network

3.1. Competitive pathway unit

CoPaNet is an attempt to separate model into subnetworks through competition. In the following, we refer to residual-type subnetworks as pathways. In a CoPa unit, multiple pathways are compiled in parallel and features are selected by using a max operation. A CoPa unit includes output x_{l+1} with K pathways, which can be formulated as follows:

$$x_{l+1} = \max_{k \in [1,...,K]} z_l^k,$$

where $z_l^k = id(x_l) + h_l^k(x_l)$, x_l is the input feature, and $h_l^k(x_l)$ represents layers of transformations on the k-th pathway at the l-th CoPa unit. Figure 2a illustrates the CoPa unit (featuring two pathways) used in this paper.

Competitive pathways appear complex; however, the proposed CoPaNet is easy to train. Notice that residual learning (He et al., 2015a) is based on the hypothesis that underlying mapping function $H(x)$ is very hard to fit. Nevertheless, the mapping function can be decomposed into two simpler functions: $H(x) = x + F(x)$. He et al. (2015a) claimed that the residual function $F(x)$ is easier to approximate. Motivated by the idea of streamlining the process of approximating the underlying mapping function, we first decompose the underlying mapping function into the maximum of two simpler functions, that is, $H(x) = \max\{f(x), g(x)\}$. We then use residual learning (He et al., 2015a) and let $f(x) = x + h^1(x)$ and $g(x) = x + h^2(x)$. The desired mapping function becomes $H(x) = \max\{x + h^1(x), x + h^2(x)\}$. This illustrates the need for two parallel networks (one each for $h^1(x)$ and $h^2(x)$), each of which comprises several stacked layers in order to approximate discrete residual functions. Because $f(x)$ and $g(x)$ are simpler, it would be easier to approximate $h^1(x)$ and $h^2(x)$ than the original residual learning (He et al., 2015a). Our CoPa unit is different from maxout unit (Goodfellow et al., 2013). The original maxout unit is constructed to obtain the maximum across several elementary neurons. Our method replaces the elementary neurons with generic functions, which are modeled by ResNets.

Further, the property of pathway encoding reveals in this architecture. We consider a 2-pathway (denote as h_l^1, h_l^2) CoPaNet with three stacked CoPa units, as show in Figure 1. We denote that the output of the first CoPa unit is $y_1 = x + h_1^1(x)$ (if h_1^1 wins) where x is the input feature. The output of the second CoPa unit can be written as $y_2 = y_1 + h_2^1(y_1)$ (if h_2^1 wins). The output of the third CoPa unit can be written as $y_3 = y_2 + h_3^2(y_2)$ (if h_3^2 wins). The final output actually can be expressed as $y_3 = x + h_1^1(x) + h_2^1(y_1) + h_3^2(y_2)$. This indicates that the final output is contributed by three winning subnetworks h_1^1, h_2^1, h_3^2 with reference to x. Thus, the routing pattern can be revealed by propagating x through the entire network.

Within a biological context, competitive mechanisms play an important role in attention (Lee et al., 1999). Researchers formulated a biological computational model in which attention activates a winner-take-all competition among neurons tuned to different visual patterns. In this model, attention alters the thresholds used to detect orientations and spatial frequencies. This suggested that winner-take-all competition can be used to explain many of the basic perceptual consequences of attention (Lee et al., 1999).

3.2. CoPaNet Architecture

CoPaNets can be simply constructed by stacking CoPa units. Let the opponent factor k denote the number of pathway in a CoPa unit and the widening factor m multiplies the number of features in convolutional layers. That is, the baseline CoPa unit corresponds to $k = 2$, $m = 1$; whereas ResNet corresponds to $k = 1$, $m = 1$.

Figure 2b shows the architecture for CIFAR and SVHN as well as Table 1 detailed the deployment. The residual shortcut in the proposed network performs identity mapping and the projection shortcut is used only to match dimensions (using 1×1 convolutions) as ResNet (He et al., 2015a, 2016). For each pathway, we adopted a "bottleneck" residual-

Table 1: Network architectures for CIFAR/SVHN (left) and ImageNet (right). Parameters of competitive pathway units are presented in braces (see also Figures 2b and c). Construction parameters for internal pathways are shown in brackets. The number of pathway is determined by the factor k and the network width is determined by the factor m. The numbers in CoPaNet-26/50/101/164 denote the depths of neural network. For the sake of clarity, the final classification layer has been omitted.

CIFAR/SVHN	
Output size	CoPaNet-164
32×32	$3 \times 3, 16$
32×32	$\left\{ \begin{bmatrix} 1 \times 1, 12 \times m \\ 3 \times 3, 12 \times m \\ 1 \times 1, 48 \times m \end{bmatrix} \times k \right\} \times 18$
16×16	2×2 avg-pool, st. 2
16×16	$\left\{ \begin{bmatrix} 1 \times 1, 24 \times m \\ 3 \times 3, 24 \times m \\ 1 \times 1, 96 \times m \end{bmatrix} \times k \right\} \times 18$
8×8	2×2 avg-pool, st. 2
8×8	$\left\{ \begin{bmatrix} 1 \times 1, 45 \times m \\ 3 \times 3, 45 \times m \\ 1 \times 1, 180 \times m \end{bmatrix} \times k \right\} \times 18$
1×1	8×8 avg-pool

ImageNet				
Output size	CoPaNet-26	CoPaNet-50	CoPaNet-101	CoPaNet-R-101
112×112	$7 \times 7, 64$, st. 2	$7 \times 7, 64$, st. 2	$7 \times 7, 64$, st. 2	$7 \times 7, 64$, st. 2
56×56	3×3 max-pool, st. 2	3×3 max-pool, st. 2	3×3 max-pool, st. 2	3×3 max-pool, st. 2
56×56	$\left\{ \begin{bmatrix} 1 \times 1, 45 \\ 3 \times 3, 45 \\ 1 \times 1, 180 \end{bmatrix} \times 2 \right\} \times 2$	$\left\{ \begin{bmatrix} 1 \times 1, 45 \\ 3 \times 3, 45 \\ 1 \times 1, 180 \end{bmatrix} \times 2 \right\} \times 3$	$\left\{ \begin{bmatrix} 1 \times 1, 45 \\ 3 \times 3, 45 \\ 1 \times 1, 180 \end{bmatrix} \times 2 \right\} \times 3$	$\left\{ \begin{bmatrix} 1 \times 1, 42 \\ 3 \times 3, 42 \\ 1 \times 1, 168 \end{bmatrix} \times 2 \right\} \times 3$
28×28	2×2 avg-pool, st. 2	2×2 avg-pool, st. 2	2×2 avg-pool, st. 2	2×2 avg-pool, st. 2
28×28	$\left\{ \begin{bmatrix} 1 \times 1, 90 \\ 3 \times 3, 90 \\ 1 \times 1, 360 \end{bmatrix} \times 2 \right\} \times 2$	$\left\{ \begin{bmatrix} 1 \times 1, 90 \\ 3 \times 3, 90 \\ 1 \times 1, 360 \end{bmatrix} \times 2 \right\} \times 4$	$\left\{ \begin{bmatrix} 1 \times 1, 90 \\ 3 \times 3, 90 \\ 1 \times 1, 360 \end{bmatrix} \times 2 \right\} \times 4$	$\left\{ \begin{bmatrix} 1 \times 1, 84 \\ 3 \times 3, 84 \\ 1 \times 1, 336 \end{bmatrix} \times 2 \right\} \times 4$
14×14	2×2 avg-pool, st. 2	2×2 avg-pool, st. 2	2×2 avg-pool, st. 2	2×2 avg-pool, st. 2
14×14	$\left\{ \begin{bmatrix} 1 \times 1, 180 \\ 3 \times 3, 180 \\ 1 \times 1, 720 \end{bmatrix} \times 2 \right\} \times 2$	$\left\{ \begin{bmatrix} 1 \times 1, 180 \\ 3 \times 3, 180 \\ 1 \times 1, 720 \end{bmatrix} \times 2 \right\} \times 6$	$\left\{ \begin{bmatrix} 1 \times 1, 180 \\ 3 \times 3, 180 \\ 1 \times 1, 720 \end{bmatrix} \times 2 \right\} \times 23$	$\left\{ \begin{bmatrix} 1 \times 1, 168 \\ 3 \times 3, 168 \\ 1 \times 1, 672 \end{bmatrix} \times 2 \right\} \times 23$
7×7	2×2 avg-pool, st. 2	2×2 avg-pool, st. 2	2×2 avg-pool, st. 2	2×2 avg-pool, st. 2
7×7	$\left\{ \begin{bmatrix} 1 \times 1, 360 \\ 3 \times 3, 360 \\ 1 \times 1, 1440 \end{bmatrix} \times 2 \right\} \times 2$	$\left\{ \begin{bmatrix} 1 \times 1, 360 \\ 3 \times 3, 360 \\ 1 \times 1, 1440 \end{bmatrix} \times 2 \right\} \times 3$	$\left\{ \begin{bmatrix} 1 \times 1, 360 \\ 3 \times 3, 360 \\ 1 \times 1, 1440 \end{bmatrix} \times 2 \right\} \times 3$	$\left\{ \begin{bmatrix} 1 \times 1, 336 \\ 3 \times 3, 336 \\ 1 \times 1, 1344 \end{bmatrix} \times 2 \right\} \times 3$
1×1	7×7 avg-pool	7×7 avg-pool	7×7 avg-pool	7×7 avg-pool

type unit comprising three convolutional layers (1×1, 3×3, 1×1). Alternatively, we could select a "basic" residual-type unit comprising two convolutional layers (3×3, 3×3). In practice, a "bottleneck" residual-type unit is deeper than a "basic" one, providing higher dimensional features. In the proposed CoPaNet, we placed BN and ReLU after all but the last convolutional layer in every pathway.

3.3. Cross-block Shortcut

The cross-block shortcuts were motivated by DenseNet (Huang et al., 2016a) which reused features from all previous layers with matching feature map sizes. In contrast to DenseNet (Huang et al., 2016a), we propose a novel feature reuse strategy: to reuse the features from previous CoPa block (stacked by many CoPa units). This is accomplished by adding identity shortcuts after pooling layers and concatenate with the output of the next block. We refer to our model with the cross-block shortcuts as CoPaNet-R, as shown in Figure 2c.

4. Experiments

We have tested the proposed CoPaNets and CoPaNets-R on several datasets, and compared the results with those of the state-of-the-art network architectures, especially ResNets.

Table 2: Comparison of test error on CIFAR and SVHN. The value of k denotes the number of hidden pieces or pathways used in a given competition. The symbol "+" indicates data augmentation (translation and horizontal flipping).

Method	Dropout	Depth	Params	C10+	C100+	SVHN
Maxout Network (k=2) (Goodfellow et al., 2013)	√	-	-	9.38	38.57	2.47
Network In Network (Lin et al., 2014)	√	-	0.98 M	8.81	35.68	2.35
Maxout Network In Network (k=5) Chang and Chen (2015)	√	-	1.6 M	6.75	28.86	1.81
Highway Network (Srivastava et al., 2015)		-	-	7.60	32.34	-
ResNet (He et al., 2015a)		110	1.7 M	6.43	-	-
Stochastic Depth Huang et al. (2016b)		110	1.7 M	5.23	24.58	1.75
		1202	19.4 M	4.91	-	-
pre-activation ResNet (He et al., 2016)		164	1.7 M	5.46	24.33	-
		1001	10.2 M	4.62	22.71	-
Wide ResNet (width=8) (Zagoruyko and Komodakis, 2016)		16	11.0 M	4.27	20.43	-
(width=10)	√	28	36.5 M	3.89	18.85	-
DenseNet (growth rate=24) (Huang et al., 2016a)		100	27.2 M	3.74	19.25	1.59
DenseNet-BC (growth rate=40)		190	25.6 M	3.46	**17.18**	-
CoPaNet (k=2, width=1)	√	164	1.75 M	4.50	22.86	1.86
CoPaNet (k=2, width=2)	√	164	6.98 M	4.10	20.48	1.83
CoPaNet (k=2, width=4)	√	164	27.9 M	3.74	18.67	1.73
CoPaNet-R (k=2, width=2)	√	164	7.00 M	3.55	20.29	1.72
CoPaNet-R (k=2, width=3)	√	164	15.7 M	**3.38**	18.90	**1.58**

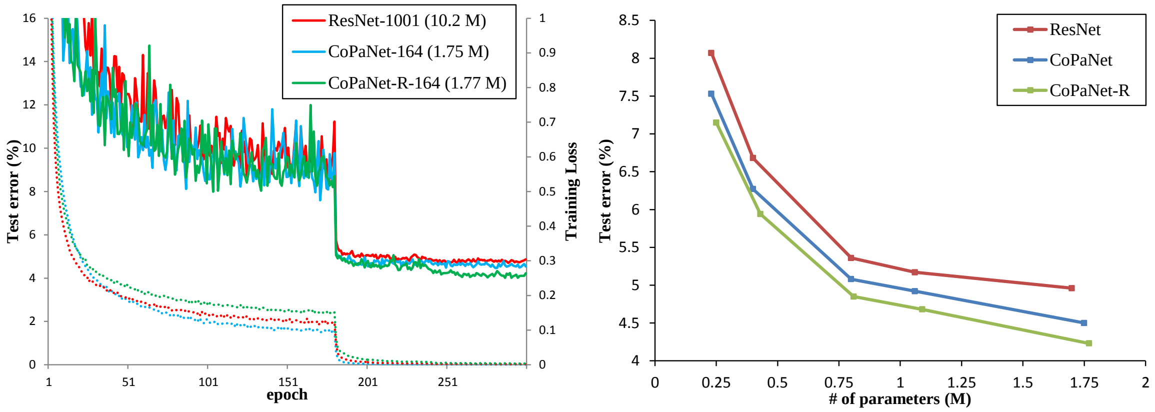
4.1. Training

We constructed a CoPaNet-164, with a set number of pathways ($k = 2$), and network width ($m = 1, 2, 4$), detailed in Table 1. Furthermore, we constructed a CoPaNet-R-164, with a set number of pathways ($k = 2$), and network width ($m = 2, 3$).

The networks were trained from scratch by using Stochastic Gradient Descent with 300 and 20 epochs for CIFAR and SVHN datasets, respectively. The learning rate for CIFAR began at 0.1, divided by 10 at 0.6 and 0.8 fractions of the total number of training epochs. The learning rate for SVHN began at 0.1, divided by 10 at 0.5 and 0.75 fractions of the total number of training epochs. A batch size of 128 was used for all tests, except for $m = 4$ when we used a batch size of 64.

On ImageNet, we trained from scratch for 100 epochs. As shown in Table 1, we constructed several CoPaNets with 2 pathways for ImageNet. The learning rate began at 0.1 and was divided by 10 after every 30 epochs. The model was implemented using Torch7 from the Github repository *fb.resnet.torch* (`https://github.com/facebook/fb.resnet.torch`). Other settings were set exactly the same as those used for ResNet.

We adopted a weight decay of 0.0001 and momentum of 0.9 as in (He et al., 2015a). Weights were initialized in accordance with the methods outlined by He et al. (2015b). We also applied Dropout (Srivastava et al., 2014a) after the average poolings except the last pooling, and it was deterministically multiplied by (1 - Dropout-rate) at test time. The Dropout rate was set to 0.2 for CIFAR and SVHN as well as 0.1 for ImageNet. The test error was evaluated using the model obtained from the final epoch at the end of training.



(a) (b)

Figure 3: (a) Training loss (dashed line) and test error (solid line) curves of the pre-activation ResNet-1001 (10.2M), CoPaNet-164 (1.75M), and CoPaNet-R-164 (1.75M). (b) Comparison of the parameter efficiency between pre-activation ResNets, CoPaNet, and CoPaNet-R.

4.2. CIFAR-10

The CIFAR-10 dataset consists of natural color images, 32×32 pixels in size, from 10 classes, and with 50,000 training and 10,000 test images. Color normalization was performed as data preprocessing. To enable a comparison with previous works, the dataset was augmented by translation as well as random flipping on the fly throughout training.

As shown in Table 2, we obtained test error of 4.50%, 4.10%, and 3.74% when using network width of $m = 1$, 2, and 4, respectively. We then compared CoPaNet-164 (1.75 M, $m = 1$) to pre-activation ResNet-1001 (10.2 M), for which He et al. (2016) reported test error of 4.62% (we obtained 4.87% in our training procedure). Figure 3a presents a comparison of training and testing curves. Furthermore, Our best result on CIFAR-10 was obtained by CoPaNet-R. We obtained 3.38% test error with only 15.7 M parameters.

4.3. CIFAR-100

The CIFAR-100 dataset is the same size and format as CIFAR-10; however, it contains 100 classes. Thus, the number of images in each class is only one tenth that of CIFAR-10. Color normalization was performed as data preprocessing. We also performed data augmentation (translation and horizontal flipping) on the CIFAR-100 dataset.

As shown in Table 2, we obtained the test error of 22.86%, 20.48%, and 18.67% for network width of $m = 1$, 2, and 4 with Dropout, respectively. CoPaNet-164 (1.75 M, $m = 1$) was compared to pre-activation ResNet-164 (1.7 M) for which He et al. (2016) reported test error of 24.33%. This puts the proposed network on par with pre-activation ResNet-1001 (10.2 M) which achieved test error of 22.71%. However, CoPaNet-R showed few benefits on CIFAR-100, and it obtained same level of accuracy.

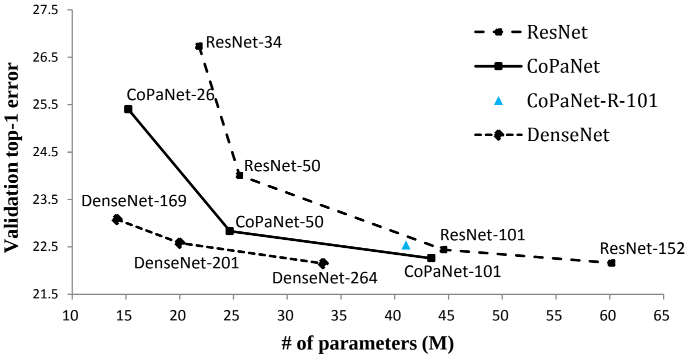


Figure 4: The comparison of top-1 validation error (single model and single crop with size 224×224) across various number of parameters among ResNet, DenseNet, and CoPaNet.

4.4. SVHN

The SVHN dataset consists of color images of house numbers (32×32 pixels) collected from Google Street View. This includes 73,257 digits in the training set, 26,032 digits in the test set, and 531,131 in an extra set. We used the entire training set and extra set for training. We did not perform any data augmentation or preprocessing except for dividing the image intensity by 255.

As shown in Table 2, the CoPaNet-164 (1.75 M, width $m = 1$) with test error of 1.86%. CoPaNet-R-164 (width $m = 3$) achieved the state-of-the-art results (1.58%) with only 15.7 M parameters.

4.5. ImageNet

The ImageNet 2012 dataset consists of 1000 classes of images with 1.28 millions for training, 50,000 for validation, and 100,000 for testing. As shown in Table 1, we constructed two-pathway CoPaNet with various depths for ImageNet. However, we reduce the number of feature maps to approximately 70% in order to retain a similar number of parameters. For a fair comparison, all results were achieved when the crop size was 224×224. Our results of single crop top-1 validation error showed better performance than ResNet, as shown in Figure 4. These results reveal that CoPaNets perform on par with the state-of-the-art ResNets, while requiring fewer parameters. CoPaNets performed worse than DenseNet with similar amounts of parameters. The major reason could be that DenseNets were much deeper than CoPaNets.

5. Discussion

5.1. Parameter Efficiency

The competitive mechanism modularizes the network into multiple parameter-sharing subnetworks and thus can improve parameter efficiency (Srivastava et al., 2013). We trained multiple small networks with various depths on CIFAR-10+. As shown in Figure 3b, both

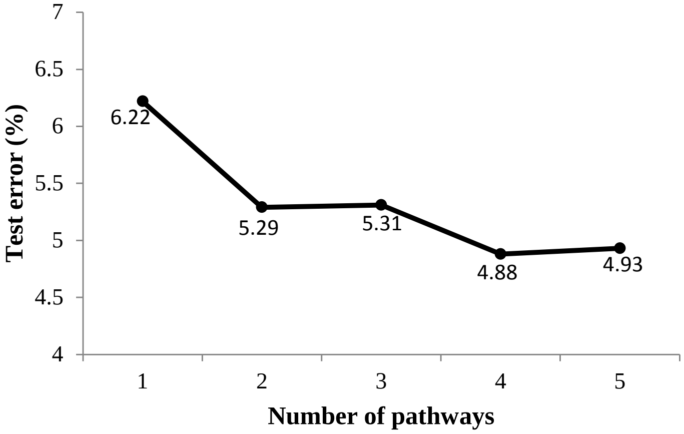


Figure 5: The influence of the number of pathways on performance in experiments based on CIFAR-10+. More pathways tends to lower test errors at the expense of more parameters.

CoPaNet and its variant outperformed pre-activation ResNet. The CoPaNet-R achieved better performance than CoPaNet. When achieving the same level of accuracy, furthermore, CoPaNet requires around a half of the parameters of pre-activation ResNet.

5.2. Number of Pathways

Figure 5 demonstrates that CoPaNet has the capacity to exploit many pathways. We trained several CoPaNets-56 (width $m = 1$) for use on CIFAR-10+ using various numbers of pathways with the Dropout rate set to 0.2. As shown in Figure 5, CoPaNet gains its benefit by increasing the number of pathways to handle complex dataset. More pathways tend to lower test errors at the expense of more parameters. Nonetheless, we adopted two pathways in our experiments to restrict the number of parameters.

5.3. Pathway Encoding

One paper (Srivastava et al., 2014b) argued that ReLU network can also encode on subnetwork activation pattern, such as maxout and LWTA networks. Srivastava et al. (2014b) discussed about the activation pattern of many filters in the same layer. In contrast to Srivastava et al. (2014b), we demonstrated the routing pattern that *one feature map propagate through many stacked pathways (subnetworks)*.

We suppose that the routing patterns are similar within the same semantics and are different between distinct semantics, which is termed as pathway encoding. As shown in Figure 6, we calculated the preference of routing patterns in a trained 2-pathway CoPaNet-164 (width $m = 1$). The preference of pathway was statistically estimated from the CIFAR-10 test set and can reveal the characteristics of the category. We illustrates the routing patterns in the last block (comprising 18 CoPa units) which contained high-level features. Each sub-figure showns the routing pattern of one feature map (4 representative feature maps were manually selected from the total of 180), and the color denoted the preference of pathways. As shown in Figure 6a, a selected routing pattern can be regarded as encoding the non-living or living groups and the routing patterns are similar in the same group. Figure 6b illustrates that the routing pattern may be encoding the flying concept such that the routing patterns of airplanes are similar to those of birds. Notice that although airplanes belong to non-living group, there exists a special pattern resembling those of animals, including the bird, as shown in Figure 6c. Furthermore, Figure 6d illustrates the diversity of routing patterns for different categories. The similarity and diversity support our hypothesis that CoPaNet is able to use pathway encoding to well represent the object images of different groups.

5.4. Coarse Feature Reuse

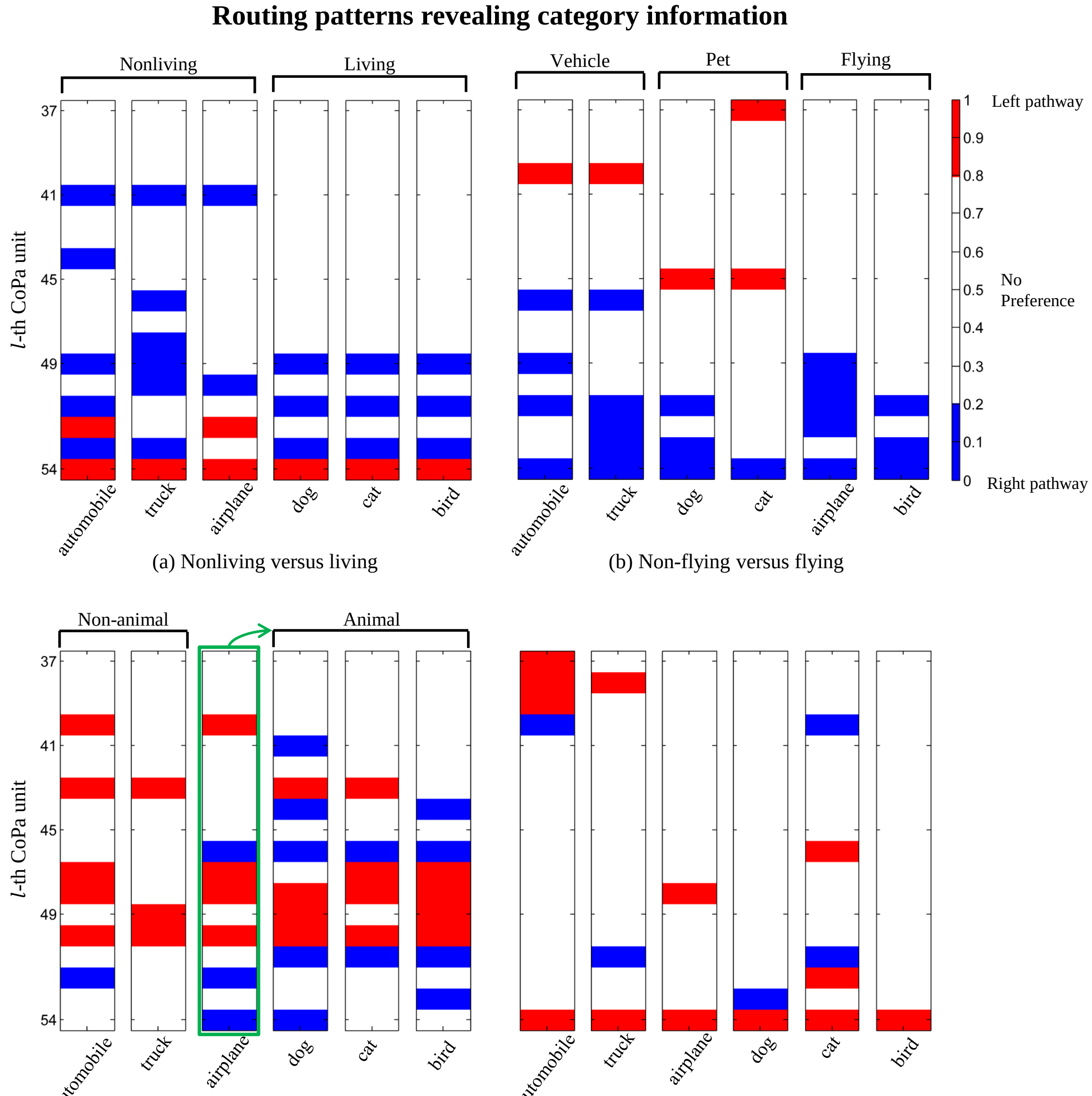
The CoPaNet-R architecture adds identity cross-block shortcuts to encourage feature reuse. This facilitates that the last classification layer can reuse coarse features from all previous blocks. Thus those shortcuts provide additional supervision because classifiers are attached to every CoPa blocks. We trained a CoPaNet-R-164 (width $m = 2$) on CIFAR-10+ and it achieved 3.55% test error, as shown in Table 2. Figure 7 shows the L^1-norm of weights of the last classification layer. In this figure, we can observe that the last classification layer uses features from early blocks. The concentration towards the final block suggests that high-level features dominate in classification.

However, CoPaNet-R did not outperform CoPaNet on CIFAR-100 and ImageNet. This may be due to the relatively few training samples for each class (500 samples per class in CIFAR-100 as well as around 1000 samples per class in ImageNet). We conducted an experiment to demonstrate this effect. We used a small CIFAR-10 dataset (1000 training samples per class) to train CoPaNet-164 and CoPaNet-R-164, both with width $m = 2$, and achieved test errors of 12.58% and 12.53%, respectively. There is no significant difference in this case. With full training set (5000 training samples per class), CoPaNet-R has significant improvement compared to CoPaNet, as shown in Table 2. The coarse feature reuse may be effective only when the amount of training samples is large enough for each class.

6. Conclusions

This paper proposes a novel convolutional neural network architecture, the CoPaNet. It introduces a nice property that input features transmit through various routing patterns for different category information, called pathway encoding. Empirical results demonstrate that the category information plays a role in selecting pathways. We showed that CoPaNet inherits the advantages of ResNet which can scale up to hundreds of layers. In our experiments, CoPaNet yielded improvements in accuracy as the number of parameters increased. Moreover, CoPaNet requires fewer parameters to achieve the same level of accuracy as the state-of-the-art ResNet. We further proposed a novel feature reuse strategy, CoPaNet-R: adding cross-block shortcuts in order to encourage the reuse of output from all previous blocks. According to our experiments, CoPaNet-R can learn accurate models by exploiting the reuse of coarse features.

Our study showed that network partitioning, feature competition, and feature reuse can lead to performance improvements. CoPaNet and its variant obtained the state-of-the-art



(a) Nonliving versus living

(b) Non-flying versus flying

(c) Airplane have properties of "animal".

(d) Different pattern in each category

Figure 6: Routing patterns showing the preference of pathway selection in a trained 2-pathway CoPaNet-164 for the CIFAR-10 test dataset. Red color denotes a preference for the left pathway, blue color for the right pathway, and white color for no preference. The vertical axis denotes the l-th CoPa units, where l indicates the depth. The category information can be represented by the routing pattern, which is referred to as *pathway encoding* in the proposed work. Each sub-figure denotes the routing pattern that one feature map propagates through its preferred route in the network. Routing patterns between (a) non-living vs. living, (b) non-flying vs. flying, (c) non-animal vs. animal, and (d) different categories are illustrated. Notice that the airplane category shows the routing pattern of "bird" in the "animal" group.

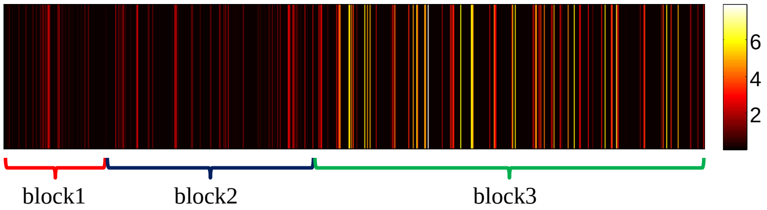


Figure 7: The color-encoded L^1-norm of the weights of the last classification layer. Notice that the last classification layer concatenates outputs from all of the three CoPa blocks through cross-block shortcuts.

or competitive results on several image recognition datasets. Other studies showed that competitive networks have other beneficial properties such as mitigation of catastrophic forgetting (Srivastava et al., 2013). In the future, we will try to adopt the trained CoPaNet to perform other tasks, such as object detection and segmentation.

Acknowledgement

This work was supported in part by the Taiwan Ministry of Science and Technology (Grants MOST-106-2221-E-009-164-MY2 and MOST-105-2218-E-009-033).

References

Masoud Abdi and Saeid Nahavandi. Multi-residual networks. *arXiv preprint arXiv:1609.05672*, 2016.

Jia-Ren Chang and Yong-Sheng Chen. Batch-normalized maxout network in network. *arXiv preprint arXiv:1511.02583*, 2015.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR09*, pages 248–255. IEEE, 2009.

Ian J Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron C Courville, and Yoshua Bengio. Maxout networks. *ICML (3)*, 28:1319–1327, 2013.

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

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, 2015b.

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

Gao Huang, Zhuang Liu, and Kilian Q Weinberger. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*, 2016a.

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

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images, 2009.

Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals. *arXiv preprint arXiv:1605.07648*, 2016.

Dale K Lee, Laurent Itti, Christof Koch, and Jochen Braun. Attention activates winner-take-all competition among visual filters. *Nature neuroscience*, 2(4):375–381, 1999.

Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *International Conference on Learning Representations*, abs/1312.4400, 2014. URL http://arxiv.org/abs/1312.4400.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 5. Granada, Spain, 2011.

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

Rupesh K Srivastava, Jonathan Masci, Sohrob Kazerounian, Faustino Gomez, and Jürgen Schmidhuber. Compete to compute. In *Advances in neural information processing systems*, pages 2310–2318, 2013.

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

Rupesh Kumar Srivastava, Jonathan Masci, Faustino Gomez, and Jürgen Schmidhuber. Understanding locally competitive networks. *arXiv preprint arXiv:1410.1165*, 2014b.

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.

Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv preprint arXiv:1602.07261*, 2016.

Andreas Veit, Michael J Wilber, and Serge Belongie. Residual networks behave like ensembles of relatively shallow networks. In *Advances in Neural Information Processing Systems*, pages 550–558, 2016.

Ziyu Wang, Tom Schaul, Matteo Hessel, Hado van Hasselt, Marc Lanctot, and Nando de Freitas. Dueling network architectures for deep reinforcement learning. *arXiv preprint arXiv:1511.06581*, 2015.

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *arXiv preprint arXiv:1611.05431*, 2016.

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

Xingcheng Zhang, Zhizhong Li, Chen Change Loy, and Dahua Lin. Polynet: A pursuit of structural diversity in very deep networks. *arXiv preprint arXiv:1611.05725*, 2016.