Mixed Link Networks

Wenhai Wang[*1], **Xiang Li**[*2], **Tong Lu**[†1], **Jian Yang**[†2]
[1] National Key Lab for Novel Software Technology, Nanjing University
[2] DeepInsight@PCALab, Nanjing University of Science and Technology
wangwenhai362@163.com, xiang.li.implus@njust.edu.cn, lutong@nju.edu.cn, csjyang@njust.edu.cn

Abstract

On the basis of the analysis by revealing the equivalence of modern networks, we find that both ResNet and DenseNet are essentially derived from the same "dense topology", yet they only differ in the form of connection – addition (dubbed "inner link") *vs.* concatenation (dubbed "outer link"). However, both forms of connections have the superiority and insufficiency. To combine their advantages and avoid certain limitations on representation learning, we present a highly efficient and modularized Mixed Link Network (MixNet) which is equipped with flexible inner link and outer link modules. Consequently, ResNet, DenseNet and Dual Path Network (DPN) can be regarded as a special case of MixNet, respectively. Furthermore, we demonstrate that MixNets can achieve superior efficiency in parameter over the state-of-the-art architectures on many competitive datasets like CIFAR-10/100, SVHN and ImageNet.

1 Introduction

The exploration of connectivity patterns of deep neural networks has attracted extensive attention in the literature of Convolutional Neural Networks (CNNs). LeNet [LeCun *et al.*, 1998] originally demonstrated its *layer-wise feed-forward* pipeline, and later GoogLeNet [Szegedy *et al.*, 2015] introduced more effective *multi-path* topology. Recently, ResNet [He *et al.*, 2016a; He *et al.*, 2016b] successfully adopted *skip connection* which transferred early information through identity mapping by element-wisely adding input features to its block outputs. DenseNet [Huang *et al.*, 2017] further proposed a seemingly "different" topology by using *densely connected path* to concatenate all the previous raw input features with the output ones.

For the two recent ResNet and DenseNet, despite their *externally large* difference in path topology (*skip connection vs. densely connected path*), **we discover and prove that both of them are essentially derived from the *same* "dense topology"** (Fig. 1 (a)), where their only difference lies in the form of connection ("+" in Fig. 1 (b) *vs.* "‖" in Fig. 1 (c)). Here, "dense topology" is defined as a path topology in which each layer H_ℓ is connected with all the previous layers $H_0, H_1, ..., H_{\ell-1}$ using the connection function $C(\cdot)$. The great effectiveness of "dense topology" has been proved via the significant success of both ResNet and DenseNet, yet the form of connection in ResNet and DenseNet still has room for improvement. For example, too many additions on the same feature space may impede the information flow in ResNet [Huang *et al.*, 2017], and there may be the same type of raw features from different layers, which leads to a certain redundancy in DenseNet [Chen *et al.*, 2017]. Therefore, the question "does there exist a more efficient form of connection in the dense topology" still remains to be further explored.

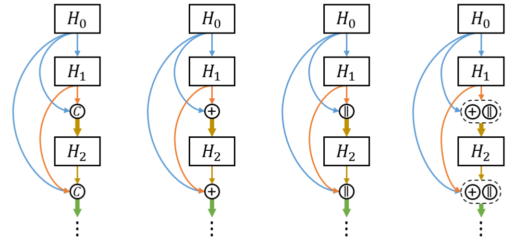
To address the problem, in this paper, we propose a novel Mixed Link Network (MixNet) with an efficient form of connection (Fig. 1 (d)) in the "dense topology". That is, we mix the connections in ResNet and DenseNet, in order to combine both the advantages of them and avoid their possible limitations. In particular, the proposed MixNets are equipped with both inner link modules and outer link modules, where an inner link module refers to *additive* feature vectors (similar connection in ResNet), while an outer link module stands for *concatenated* ones (similar connection in DenseNet). More importantly, in the architectures of MixNets, these two types of link modules are flexible with their positions and sizes. As a result, ResNet, DenseNet and the recently proposed Dual Path Network (DPN) [Chen *et al.*, 2017] can be regarded as a special case of MixNet, respectively (see the details in Fig. 4 and Table 1).

To show the efficiency and effectiveness of the proposed MixNets, we conduct extensive experiments on four competitive benchmark datasets, namely, CIFAR-10, CIFAR-100, SVHN and ImageNet. The proposed MixNets require fewer parameters than the existing state-of-the-art architectures whilst achieving better or at least comparable results. Notably, on CIFAR-10 and CIFAR-100 datasets, MixNet-250

[*] Authors contributed equally

[†] Corresponding authors

[2] Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education, School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, 210094, P.R. China & Jiangsu Key Laboratory of Image and Video Understanding for Social Security, School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, 210094, P.R. China



(a) Dense Topology (b) ResNet (c) DenseNet (d) MixNet

Figure 1: The topological relations of different types of neural networks. The symbols "+" and "∥" denote element-wise addition and concatenation, respectively. (a) shows the general form of "dense topology". $C(\cdot)$ refers to the connection function. (b) shows ResNet in the perspective of "dense topology". (c) shows the path topology of DenseNet. (d) shows the path topology of MixNet.

surpasses ResNeXt-29 (16×64d) with 57% less parameters. On ImagNet dataset, the results of MixNet-141 are comparable to the ones of DPN-98 with 50% fewer parameters.

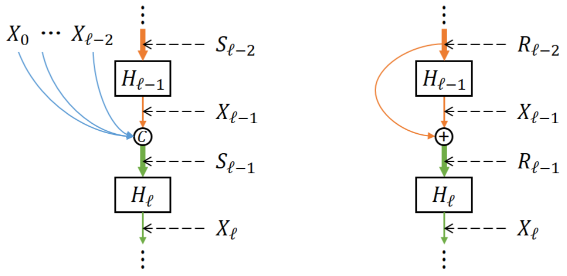
The main contributions of this paper are as follows:

- ResNet and DenseNet are proved to have the *same* path topology – "dense topology" essentially, whilst their only difference lies in the form of connections.
- A Mixed Link Network (MixNet) is proposed, which has a more efficient connection – the blending of flexible inner link modules and outer link modules.
- The relation between MixNet and modern networks (ResNet, DenseNet and DPN) is discussed, and these networks are shown to be specific instances of MixNets.
- MixNet demonstrates its superior efficiency in parameter over the state-of-the-art architectures on many competitive benchmarks.

The remainder of the paper is organized as follows. We summarize related work in Section 2 and devote Section 3 to study the "dense topology". Section 4 contains the main technical design of Mixed Link Networks. Experimental results and comparisons are presented in Section 5. Finally, we conclude this paper in Section 6.

2 Related Work

Designing effective path topologies always pushes the frontier of the advanced neural network architecture. Following the initial layer-wise feed-forward pipeline [LeCun *et al.*, 1998], AlexNet [LeCun *et al.*, 1998] and VGG [Simonyan and Zisserman, 2015] showed that building deeper networks with tiny convolutional kernels is a promising way to increase the learning capacity of neural network. GoogLeNet [Szegedy *et al.*, 2015] demonstrated that a multi-path topology (codenamed Inception) could easily outperform previous feed-forward baselines by blending various information flows. The effectiveness of multi-path topology was further validated in FractalNet [Larsson *et al.*, 2016], Highway Networks [Srivastava *et al.*, 2015], DFN [Wang *et al.*, 2016], DFN-MR [Zhao *et al.*, 2016], and IGC [Zhang *et al.*, 2017]. Perhaps the most revolutionary topology – skip connection was successfully adopted by ResNet [He *et al.*, 2016a; He *et al.*, 2016b], where micro-blocks were built sequentially and the skip connection bridged the micro-block's input features with its output ones via identity mappings. Since then, different works based on ResNet have arisen, aiming to find a more efficient transformer of that micro-block, such as WRN [Zagoruyko and Komodakis, 2016], Multi-ResNet [Abdi and Nahavandi, 2016] and ResNeXt [Xie *et al.*, 2017]. Furthermore, DenseNet [Huang *et al.*, 2017] achieved comparable accuracy with deep ResNet by proposing the densely connected topology, which connects each layer to its previous layers by concatenation. Recently, DPN [Chen *et al.*, 2017] directly combines the two paths – ResNet path and DenseNet path together by a shared feature embedding in order to enjoy a mutual improvement.



(a) (b)

Figure 2: The key annotations for $H(\cdot)$, X, S and R. Function $H_\ell(\cdot)$ represents a non-linear transformation which is composite function of several operations. X_ℓ is the output of function $H_\ell(\cdot)$. S_ℓ gives the result of the connection function $C(\cdot)$ whose inputs come from all the previous feature-maps X (i.e., $X_0, X_1, , X_\ell$). R refers to the feature-maps directly after the skip connection.

3 Dense Topology

In this section, we first introduce and formulate the "dense topology". We then prove that both ResNet and DenseNet are intrinsically derived from the same "dense topology", and they only differ in the specific form of connection (addition *vs.* concatenation). Furthermore, we present analysis on strengths and weaknesses of these two network architectures, which motivates us to develop Mixed Link Networks.

Definitions of "dense topology". Let us consider a network that comprises L layers, each of which implements a non-linear transformation $H_\ell(\cdot)$, where ℓ indexes the layer. $H_\ell(\cdot)$ could be a composite function of several operations such as linear transformation, convolution, activation function, pooling [LeCun *et al.*, 1998], batch normalization [Ioffe and Szegedy, 2015]. As illustrated in Fig. 2 (a), X_ℓ refers to the immediate output of the transformation $H_\ell(\cdot)$ and S_ℓ is the result of the connection function $C(\cdot)$ whose inputs come from all the previous feature-maps X (i.e., $X_0, X_1, ..., X_\ell$). Initially, S_0 equals X_0. As mentioned in Section 1, "dense topology" is defined as a path topology where each layer is connected with all the previous layers. Therefore, we can formulate the general form of "dense topology" simply as:

$$X_\ell = H_\ell(C(X_0, X_1, \cdots, X_{\ell-1})). \quad (1)$$

DenseNet is derived from "dense topology" obviously. For DenseNet [Huang *et al.*, 2017], the input of ℓ^{th} layer is the concatenation of the outputs $X_0, X_1, ..., X_{\ell-1}$ from all the preceding layers. Therefore, we can write DenseNet as:

$$X_\ell = H_\ell(X_0 \parallel X_1 \parallel \cdots \parallel X_{\ell-1}), \quad (2)$$

where "‖" refers to the concatenation. As shown in Eqn. (1) and Eqn. (2), DenseNet directly follows the formulation of "dense topology", whose connection function is the pure concatenation (Fig. 1 (c)).

ResNet is also derived from "dense topology". We then explain that ResNet also follows the "dense topology" whose connection is accomplished by addition. Given the standard definition from [He *et al.*, 2016b], ResNet poses a skip connection that bypasses the non-linear transformations $H_\ell(\cdot)$ with an identity mapping as:

$$R_\ell = H_\ell(R_{\ell-1}) + R_{\ell-1}, \quad (3)$$

where R refers to the feature-maps directly after the skip connection (Fig. 2 (b)). Initially, R_0 equals X_0. Now we concentrate on X_ℓ which is the output of $H_\ell(\cdot)$ as well:

$$X_\ell = H_\ell(R_{\ell-1}). \quad (4)$$

By substituting Eqn. (3) into Eqn. (4) recursively, we can rewrite Eqn. (4) as:

$$\begin{aligned} X_\ell &= H_\ell(R_{\ell-1}) = H_\ell(H_{\ell-1}(R_{\ell-2}) + R_{\ell-2}) \\ &= H_\ell(H_{\ell-1}(R_{\ell-2}) + H_{\ell-2}(R_{\ell-3}) + R_{\ell-3}) \\ &= \cdots \\ &= H_\ell(\sum\nolimits_{i=1}^{\ell-1} H_i(R_{i-1}) + R_0) \\ &= H_\ell(\sum\nolimits_{i=1}^{\ell-1} X_i + X_0) \\ &= H_\ell(X_0 + X_1 + \cdots + X_{\ell-1}). \quad (5) \end{aligned}$$

As shown in Eqn. (5) clearly, $R_{\ell-1}$ in ResNet is deduced to be the element-wise addition result of all the previous layers – $X_0, X_1, ..., X_{\ell-1}$. It proves that ResNet is actually identical to a form of "dense topology", where the connection function $C(\cdot)$ is specified to the addition (Fig. 1 (b)).

The above analyses reveal that ResNet and DenseNet share the same "dense topology" in essence. Therefore, the "dense topology" is confirmed to be a *fundamental* and *significant* path topology that works practically, due to the extraordinary effectiveness of both ResNet and DenseNet in the recent progress. Meanwhile, from Eqn. (2) and Eqn. (5), the only difference between ResNet and DenseNet is the connection function $C(\cdot)$ ("+" *vs.* "‖") obviously.

Analysis of ResNet. The connection in ResNet is *only* the additive form ("+") that operates on the entire feature map. It combines the features from previous layers by element-wise addition, which makes the features more expressive and eases the gradient flow for optimization simultaneously. However, too many additions on the same feature space may impede the information flow in the network [Huang *et al.*, 2017], which motivates us to develop a "shifted additions", by dislocating/shifting the additive positions in subsequent feature spaces along multiple layers (e.g., the black arrow in Fig. 4 (e)), to alleviate this problem.

Analysis of DenseNet. The connection in DenseNet is *only* the concatenative connection ("‖") which increases the feature dimension gradually along the depths. It concatenates the raw features from previous layers to form the input of the new layer. Concatenation allows the new layer to receive the raw features directly from previous layers and it also improves the flow of information between layers. However, there may be the same type of features from different layers, which leads to a certain redundancy [Chen *et al.*, 2017]. This limitation also inspires us to introduce the "shifted additions" (e.g., the black arrow in Fig. 4 (e)) on these raw features in purpose of a modification to avoid that redundancy to some extent.

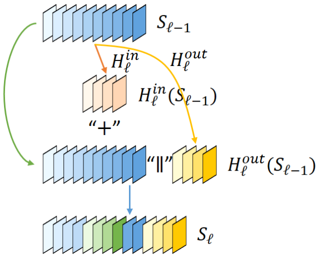


Figure 3: The example of mixed link architecture. The symbol "+" and "‖" represent addition and concatenation, respectively. The green arrows denote duplication operation.

4 Mixed Link Networks

In this section, we first introduce and formulate the inner/outer link modules. Next, we present the generalized mixed link architecture with flexible inner/outer link modules and propose Mixed Link Network (MixNet), a representative form of the generalized mixed link architecture. At last, we describe the implementation details of MixNets.

4.1 Inner/Outer Link Module

The inner link modules are based on the additive connections. Following the above preliminaries, we denote the output S_ℓ^{in} which contains the inner link part as:

$$S_\ell^{in} = \sum\nolimits_{i=0}^{\ell} X_i = S_{\ell-1}^{in} + X_\ell = S_{\ell-1}^{in} + H_\ell^{in}(S_{\ell-1}^{in}), \quad (6)$$

where $H_\ell^{in}(\cdot)$ refers to the function of producing feature-maps for inner linking – element-wisely adding new features $H_\ell^{in}(S_{\ell-1}^{in})$ inside the original ones $S_{\ell-1}^{in}$.

The outer link modules are based on the concatenated connection. Similarly, we have S_ℓ^{out} as:

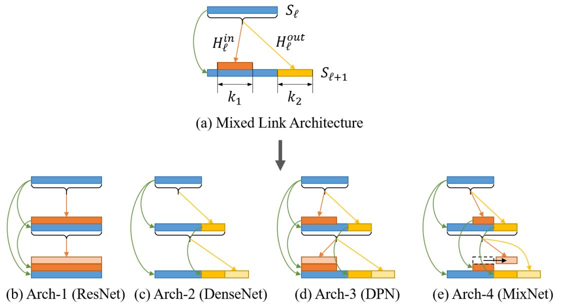
$$\begin{aligned} S_\ell^{out} &= X_0 \parallel X_1 \parallel \cdots \parallel X_\ell = S_{\ell-1}^{out} \parallel X_\ell \\ &= S_{\ell-1}^{out} \parallel H_\ell^{out}(S_{\ell-1}^{out}), \quad (7) \end{aligned}$$

where $H_\ell^{out}(\cdot)$ refers to the function of producing feature-maps for outer linking – appending new features $H_\ell^{out}(S_{\ell-1}^{out})$ outside the original ones $S_{\ell-1}^{out}$.

4.2 Mixed Link Architecture

Due to the analyses in Section 3, we introduce the mixed link architecture which embraces both inner link modules and outer link modules (Fig. 3). The mixed link architecture can be formulated as Eqn. (8), a flexible combination of Eqn. (6) and Eqn. (7), to get a blending feature output S_ℓ:

$$S_\ell = (S_{\ell-1} + H_\ell^{in}(S_{\ell-1})) \parallel H_\ell^{out}(S_{\ell-1}). \quad (8)$$



(a) Mixed Link Architecture

(b) Arch-1 (ResNet) (c) Arch-2 (DenseNet) (d) Arch-3 (DPN) (e) Arch-4 (MixNet)

Figure 4: Four architectures derived from mixed link architecture. The view is placed on the channels of one location of feature-maps in convolutional neural networks. The orange arrows denote the function $H_\ell^{in}(\cdot)$ for the inner link module. The yellow arrows denote the function $H_\ell^{out}(\cdot)$ for the outer link module. The green arrows refer to duplication operation. The vertically aligned features are merged by element-wise addition, and the horizontally aligned features are merged by concatenation.

Definitions of parameters (k_1, k_2, fixed/unfixed) for mixed link architecture. Here we denote the channel number of feature-maps produced by $H_\ell^{in}(\cdot)$ and $H_\ell^{out}(\cdot)$ as k_1 and k_2, respectively. That is, k_1 is the inner link size for inner link modules, and k_2 controls the outer link size for outer link modules. As for the positional control for inner link modules, we simplify it into two choices – "fixed" or "unfixed". The "fixed" term is easy to understand – all the features are merged together by addition over the same fixed space, as in ResNet. Here is the explanation for "unfixed": there are extremely exponential combinations to pose the inner link modules' positions along multiple layers, and learning the variable position is currently unavailable since their arrangement is not derivable directly. Therefore, we make a compromise and choose one simple series of the unfixed-position version – the "shifted addition" (Fig. 4 (e)) as mentioned in our motivations in Section 3. Specifically, the position of inner link part exactly aligns with the growing boundary of entire feature embedding (see the black arrow in Arch-4) when the outer link parts increase the overall feature dimension. We denote this Arch-4 (Fig. 4 (e)) to be our proposed model exactly – Mixed Link Network (MixNet). In summary, we have defined the above two simple options for controlling the positions of inner link modules as – "fixed" and "unfixed" .

Modern networks are special cases of MixNets. It can be seen from Fig. 4 that the mixed link architecture (Fig. 4 (a)) with different parametric configurations can reach four representative architectures (Fig. 4 (b)(c)(d)(e)). The configurations of these corresponding architectures are listed in Table 1. We show that MixNet is a more generalized form than other exsiting modern networks, under the perspective of mixed link architecture. Therefore, ResNet, DenseNet and DPN can be treated as a specific instance of MixNets, respectively.

4.3 Implementation Details of MixNets

The proposed network consists of multiple mixed link blocks. Each mixed link block has several layers, whose structure follows Arch-4 (Fig. 4 (e)). Motivated from the common practices [He *et al.*, 2016a], we introduce bottleneck layers as unitary elements in MixNets. That is, we implement both $H_\ell^{in}(\cdot)$ and $H_\ell^{out}(\cdot)$ with such a bottleneck layer – BN-ReLU-Conv(1, 1)-BN-ReLU-Conv(3, 3). Here BN, ReLU, and Conv refer to batch normalization, rectified linear units and convolution, respectively.

Architecture	Inner Link Module Setting	Outer Link Module Setting
Arch-1 (ResNet)	$k_1 > 0$, fixed	$k_2 = 0$
Arch-2 (DenseNet)	$k_1 = 0$	$k_2 > 0$
Arch-3 (DPN)	$k_1 > 0$, fixed	$k_2 > 0$
Arch-4 (MixNet)	$k_1 > 0$, unfixed	$k_2 > 0$

Table 1: The configurations of the four representative architectures.

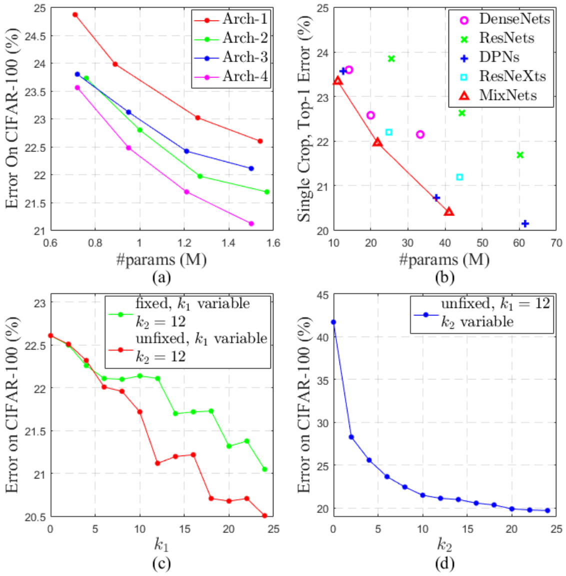


Figure 5: The illustrations of the experimental results. (a) shows the parameter efficiency comparisons among the four architectures. (b) is the comparison of the MixNets and the state-of-the-art architectures top-1 error (single-crop) on the ImageNet validation set as a function of model parameters. (c) shows error rates of the models, whose inner link modules are fixed or unfixed. (d) shows error rates of the models with different outer link parameter k_2.

On CIFAR-10, CIFAR-100 and SVHN datasets, the MixNets used in our experiments have three mixed link blocks with the same amount of layers. Before entering the first mixed link block, a convolution with $max(k_1, 2 \times k_2)$ output channels is performed on the input images. For convolutional layers with kernel size 3×3, each side of the inputs is zero-padded by one pixel to keep the feature-map size fixed. We use 1×1 convolution followed by 2×2 average pooling as transition layers between two contiguous blocks. At the end of the last block, a global average pooling is performed and then a softmax classifier is attached. The feature-map sizes in the three blocks are 32×32, 16×16, and 8×8, respectively. We survey the network structure with three configurations: $\{L = 100, k_1 = 12, k_2 = 12\}$, $\{L = 250, k_1 = 24, k_2 = 24\}$ and $\{L = 190, k_1 = 40, k_2 = 40\}$ in practice.

In our experiments on ImageNet dataset, we follow Arch-4 and use the network structure with four mixed link blocks on

Layers	Output Size	MixNet-105 ($k_1 = 32, k_2 = 32$)	MixNet-121 ($k_1 = 40, k_2 = 40$)	MixNet-141 ($k_1 = 48, k_2 = 48$)
Convolution	112×112	7×7 conv, stride 2		
Pooling	56×56	3×3 max pool, stride 2		
Mixed Link Block (1)	56×56	$\left[\left[\begin{array}{c}1 \times 1, \text{conv}\\ 3 \times 3, \text{conv}\end{array}\right] \times 2\right] \times 6$	$\left[\left[\begin{array}{c}1 \times 1, \text{conv}\\ 3 \times 3, \text{conv}\end{array}\right] \times 2\right] \times 6$	$\left[\left[\begin{array}{c}1 \times 1, \text{conv}\\ 3 \times 3, \text{conv}\end{array}\right] \times 2\right] \times 6$
Convolution	56×56	1×1 conv		
Pooling	28×28	2×2 average pool, stride 2		
Mixed Link Block (2)	28×28	$\left[\left[\begin{array}{c}1 \times 1, \text{conv}\\ 3 \times 3, \text{conv}\end{array}\right] \times 2\right] \times 12$	$\left[\left[\begin{array}{c}1 \times 1, \text{conv}\\ 3 \times 3, \text{conv}\end{array}\right] \times 2\right] \times 12$	$\left[\left[\begin{array}{c}1 \times 1, \text{conv}\\ 3 \times 3, \text{conv}\end{array}\right] \times 2\right] \times 12$
Convolution	56×56	1×1 conv		
Pooling	28×28	2×2 average pool, stride 2		
Mixed Link Block (3)	14×14	$\left[\left[\begin{array}{c}1 \times 1, \text{conv}\\ 3 \times 3, \text{conv}\end{array}\right] \times 2\right] \times 20$	$\left[\left[\begin{array}{c}1 \times 1, \text{conv}\\ 3 \times 3, \text{conv}\end{array}\right] \times 2\right] \times 24$	$\left[\left[\begin{array}{c}1 \times 1, \text{conv}\\ 3 \times 3, \text{conv}\end{array}\right] \times 2\right] \times 30$
Convolution	56×56	1×1 conv		
Pooling	28×28	2×2 average pool, stride 2		
Mixed Link Block (4)	7×7	$\left[\left[\begin{array}{c}1 \times 1, \text{conv}\\ 3 \times 3, \text{conv}\end{array}\right] \times 2\right] \times 12$	$\left[\left[\begin{array}{c}1 \times 1, \text{conv}\\ 3 \times 3, \text{conv}\end{array}\right] \times 2\right] \times 16$	$\left[\left[\begin{array}{c}1 \times 1, \text{conv}\\ 3 \times 3, \text{conv}\end{array}\right] \times 2\right] \times 20$
Classification Layer	1×1	7×7 global average pool		
	1000	1000D fully-connected, softmax		

Table 2: MixNet architectures for ImageNet. k_1 and k_2 denote the parameters for inner and outer link modules, respectively.

224×224 input images. The initial convolution layer comprises $max(k_1, 2 \times k_2)$ filters whose size is 7×7 and stride is 2. The sizes of feature-maps in the following layers are determined by the settings of inner link parameter k_1 and outer link parameter k_2 (Table 2), consequently.

5 Experiment

In this section, we empirically demonstrate MixNets effectiveness and efficiency in parameter over the state-of-the-art architectures on many competitive benchmarks.

5.1 Datasets

CIFAR. The two CIFAR datasets [Krizhevsky and Hinton, 2009] consist of colored natural images with 32×32 pixels. The training and test sets contain 50K and 10K images, respectively. We follow the standard data augmentation scheme that is widely used for these two datasets [He *et al.*, 2016a; Huang *et al.*, 2016; Li *et al.*, 2018]. For preprocessing, we normalize the data using the channel means and standard deviations. For the final run we use all 5K training images and report the final test error at the end of training.

SVHN. The Street View House Numbers (SVHN) dataset [Netzer *et al.*, 2011] contains 32×32 colored digit images. There are 73,257 images in the training set, 26,032 images in the test set, and 531,131 images for extra training data. Following common practice [Huang *et al.*, 2016; Lin *et al.*, 2014], We use all the training data (training set and extra training data) without any data augmentation, and a validation set with 6,000 images is split from the training set. In addition, the pixel values in the dataset are divided by 255 and thus they are in the [0, 1] range. We select the model with the lowest validation error during training and report the test error.

ImageNet. The ILSVRC 2012 classification dataset [Deng *et al.*, 2009] contains 1.2 million images for training, and 50K for validation, from 1K classes. We adopt the same data augmentation scheme for training images as in [He *et al.*, 2016a; He *et al.*, 2016b], and apply a single-crop with size 224×224 at test time. Following [He *et al.*, 2016a; He *et al.*, 2016b], we report classification errors on the validation set.

5.2 Training

All the networks are trained by using stochastic gradient descent (SGD). On CIFAR and SVHN we train using batch size 64 for 300 epochs. The initial learning rate is set to 0.1, and is divided by 10 at 50% and 75% of the total number of training epochs. On ImageNet, we train models with a mini-batch size 150 (MixNet-121) and 100 (MixNet-141) due to GPU memory constraints. To compensate for the smaller batch size, the models are trained for 100 epochs, and the learning rate is lowered by 10 times at epoch 30, 60 and 90. Following [He *et al.*, 2016a], we use a weight decay of 10^{-4} and a Nesterov momentum [Sutskever *et al.*, 2013] of 0.9 without dampening. We adopt the weight initialization introduced by [He *et al.*, 2015]. For the the dataset without data augmentation (i.e., SVHN), we follow the DenseNet setting [Huang *et al.*, 2017] and add a dropout layer [Srivastava *et al.*, 2014] after each convolutional layer (except the first one) by setting the dropout rate as 0.2.

5.3 Ablation Study for Mixed Link Architecture

Efficiency comparisons among the four architectures. We first evaluate the efficiency of the four representative architectures which are derived from the mixed link architecture. The comparisons are based on various amount of parameters (#params). Specifically, we increase the complexities of the four architectures in parallel and evaluate them on CIFAR-100 dataset. The experimental results are reported in Fig. 5 (a), from which we can find that with various similar parameters, Arch-4 outperforms all other three architectures by a margin. It demonstrates the superior efficiency in parameter of Arch-4 which is exactly used in our proposed MixNets.

Fixed *vs.* unfixed for the inner link modules. Next we investigate "which is the more effective setting for the inner link

Method	Depth	#params	CIFAR-10	CIFAR-100	SVHN
DFN [Wang *et al.*, 2016]	50	3.9M	6.24	27.52	-
DFN-MR [Zhao *et al.*, 2016]	50	24.8M	3.57	19.00	**1.55**
FractalNet [Larsson *et al.*, 2016]	21	38.6M	4.60	23.73	1.87
ResNet with Stochastic Depth [Huang *et al.*, 2016]	110	1.7M	5.25	24.98	1.75
ResNet-164 (pre-activation) [He *et al.*, 2016b]	164	1.7M	4.80	22.11	-
ResNet-1001 (pre-activation) [He *et al.*, 2016b]	1001	10.2M	4.92	22.71	-
WRN-28-10 [Zagoruyko and Komodakis, 2016]	28	36.5M	4.00	19.25	-
ResNeXt-29 (8 × 64d) [Xie *et al.*, 2017]	29	34.4M	3.65	17.77	-
ResNeXt-29 (16 × 64d) [Xie *et al.*, 2017]	29	68.1M	3.58	17.31	-
DenseNet-100 ($k = 24$) [Huang *et al.*, 2017]	100	27.2M	3.74	19.25	1.59
DenseNet-BC-190 ($k = 40$) [Huang *et al.*, 2017]	190	25.6M	3.46	**17.18**	-
DPN-28-10 [Chen *et al.*, 2017]	28	47.8M	3.65	20.23	-
IGC-$L32M26$ [Zhang *et al.*, 2017]	20	24.1M	**3.31**	18.75	**1.56**
MixNet-100 ($k_1 = 12, k_2 = 12$)	100	1.5M	4.19	21.12	1.57
MixNet-250 ($k_1 = 24, k_2 = 24$)	250	29.0M	**3.32**	**17.06**	**1.51**
MixNet-190 ($k_1 = 40, k_2 = 40$)	190	48.5M	**3.13**	**16.96**	-

Table 3: Error rates (%) on CIFAR and SVHN datasets. k_1 and k_2 denote the parameters for inner and outer link modules, respectively. The best, second-best, and third-best accuracies are highlighted in red, blue, and green.

modules – fixed or unfixed?". To ensure a fair comparison, we hold the outer link parameter k_2 constant and train MixNets with different inner link parameter k_1. In details, we set k_2 to 12, and let k_1 increase from 0 to 24. The models are also evaluated on CIFAR-100 dataset. Fig. 5 (c) shows the experimental results, from which we can find that with the growing of k_1, the test error rate keeps dropping. Furthermore, with the same inner link parameter k_1, the models with unfixed inner link modules (red curve) have much lower test errors than the models with the fixed ones (green curve), which suggests the superiority of unfixed inner link module.

Outer link size. We then study the effect of outer link size k_2 by setting $k_1 = 12$, under the configurations with the effective unfixed inner link modules on CIFAR-100 dataset. Fig. 5 (d) illustrates that the increasement of k_2 reduces the test error rate consistently. However, the performance gain becomes tiny when k_2 is relatively large. Therefore, we prefer to set $k_2 = k_1$ for a better space-time tradeoff in this study.

5.4 Experiments on CIFAR and SVHN

We train MixNets with different depths L, inner link parameters k_1 and outer link parameters k_2. The main results on CIFAR and SVHN are shown in Table 3.

As can be seen from the bottom rows of Table 3, MixNet-190 outperforms many state-of-the-art architectures consistently on CIFAR datasets. Its error rates, 3.13% on CIFAR-10 and 16.96% on CIFAR-100, are significantly lower than the error rates achieved by DPN-29-10. Our results on SVHN are also remarkable. MixNet-100 achieves comparable test errors with DFN-MR (24.1M) and IGC-$L32M26$ (24.8M) whilst costing only 1.5M parameters.

5.5 Experiments on ImageNet

We evaluate MixNets with different depths and inner/outer link parameters on the ImageNet classification task, and compare it with the representative state-of-the-art architectures. We report the single-crop validation errors of MixNets on ImageNet in Table 4. The single-crop top-1 validation errors of MixNets and different state-of-the-art architectures as a function of the number of parameters are shown in Fig. 5 (b). The results reveal that MixNets perform on par with the state-of-the-art architectures, whilst requiring significantly fewer parameters to achieve better or at least comparable performance. For example, MixNet-105 outperforms DenseNet-169 and DPN-68 with only 11.16M parameters. MixNet-121 (21.86M) yields better validation error than ResNeXt-50 (25M) and Densenet-264 (33.34M). Furthermore, the results of MixNet-141 are very close to the ones of DPN-98 with 50% fewer parameters.

Method	#params	top-1	top-5
ResNet-50 [He *et al.*, 2016a]	25.56M	23.9	7.1
ResNet-101 [He *et al.*, 2016a]	44.55M	22.6	6.4
ResNet-152 [He *et al.*, 2016a]	60.19M	21.7	6.0
DenseNet-169 [Huang *et al.*, 2017]	14.15M	23.8	6.9
DenseNet-201 [Huang *et al.*, 2017]	20.01M	22.6	6.3
DenseNet-264 [Huang *et al.*, 2017]	33.34M	22.2	6.1
ResNeXt-50 (32 × 4d) [Xie *et al.*, 2017]	25M	22.2	-
ResNeXt-101 (32 × 4d) [Xie *et al.*, 2017]	44M	21.2	5.6
DPN-68 (32 × 4d) [Chen *et al.*, 2017]	12.61M	23.7	7.0
DPN-92 (32 × 3d) [Chen *et al.*, 2017]	37.67M	20.7	5.4
DPN-98 (32 × 4d) [Chen *et al.*, 2017]	61.57M	20.2	5.2
MixNet-105 ($k_1 = 32, k_2 = 32$)	11.16M	23.3	6.7
MixNet-121 ($k_1 = 40, k_2 = 40$)	21.86M	21.9	5.9
MixNet-141 ($k_1 = 48, k_2 = 48$)	41.07M	20.4	5.3

Table 4: The top-1 and top-5 error rates on the ImageNet validation set, with single-crop testing.

6 Conclusion

In this paper, we first prove that ResNet and DenseNet are essentially derived from the same *fundamental* "dense topology", whilst their only difference lies in the specific form of connection. Next, by the analysis of superiority and insufficiency of their distinct connections, we propose a highly efficient form of it – the Mixed Link Networks (MixNets), making an effective bridge between ResNet and DenseNet. Extensive experimental results demonstrate that our proposed MixNet is efficient in parameter.

Acknowledgements

The authors would like to thank the editor and the anonymous reviewers for their critical and constructive comments and suggestions. The work was supported by the Natural Science Foundation of China under Grant No. 61672273, No. 61272218 and No. 61321491, the Science Foundation for Distinguished Young Scholars of Jiangsu under Grant No. BK20160021, Scientific Foundation of State Grid Corporation of China (Research on Ice-wind Disaster Feature Recognition and Prediction by Few-shot Machine Learning in Transmission Lines), the National Science Fund of China under Grant Nos. U1713208 and 61472187, the 973 Program No.2014CB349303, and Program for Changjiang Scholars.

References

[Abdi and Nahavandi, 2016] Masoud Abdi and Saeid Nahavandi. Multi-residual networks. *CoRR, abs/1609.05672*, 8, 2016.

[Chen *et al.*, 2017] Yunpeng Chen, Jianan Li, Huaxin Xiao, Xiaojie Jin, Shuicheng Yan, and Jiashi Feng. Dual path networks. In *NIPS*, 2017.

[Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[He *et al.*, 2015] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015.

[He *et al.*, 2016a] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[He *et al.*, 2016b] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, 2016.

[Huang *et al.*, 2016] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *ECCV*, 2016.

[Huang *et al.*, 2017] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. Densely connected convolutional networks. In *CVPR*, 2017.

[Ioffe and Szegedy, 2015] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

[Krizhevsky and Hinton, 2009] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Tech Report*, 2009.

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

[LeCun *et al.*, 1998] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Li *et al.*, 2018] Xiang Li, Shuo Chen, Xiaolin Hu, and Jian Yang. Understanding the disharmony between dropout and batch normalization by variance shift. *arXiv preprint arXiv:1801.05134*, 2018.

[Lin *et al.*, 2014] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. In *ICLR*, 2014.

[Netzer *et al.*, 2011] 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*, 2011.

[Simonyan and Zisserman, 2015] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[Srivastava *et al.*, 2014] 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, 2014.

[Srivastava *et al.*, 2015] Rupesh K Srivastava, Klaus Greff, and Jürgen Schmidhuber. Training very deep networks. In *NIPS*, 2015.

[Sutskever *et al.*, 2013] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *ICML*, 2013.

[Szegedy *et al.*, 2015] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[Wang *et al.*, 2016] Jingdong Wang, Zhen Wei, Ting Zhang, and Wenjun Zeng. Deeply-fused nets. *arXiv preprint arXiv:1605.07716*, 2016.

[Xie *et al.*, 2017] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017.

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

[Zhang *et al.*, 2017] Ting Zhang, Guo-Jun Qi, Bin Xiao, and Jingdong Wang. Interleaved group convolutions. In *CVPR*, 2017.

[Zhao *et al.*, 2016] Liming Zhao, Jingdong Wang, Xi Li, Zhuowen Tu, and Wenjun Zeng. On the connection of deep fusion to ensembling. *arXiv preprint arXiv:1611.07718*, 2016.