

AID\_paper1-2

# Densenet 논문 리뷰

논문리뷰 스터디 - 정보컴퓨터공학부 인공지능전공 최지훈

# 목차

## 01 Abstract

논문 개관 내용 설명

---

## 02 Introduction

논문의 introduction 부분과 함께 덴스넷 설명

---

## 03 DenseNets

논문에서 나온 수식 학습내용 발표

---

## 04 Comparision

기존 신경망과의 로스값, 정확도 비교 그래프 분석

---

## Abstract

*Recent work has shown that convolutional networks can be substantially deeper, more accurate, and efficient to train if they contain shorter connections between layers close to the input and those close to the output. In this paper, we embrace this observation and introduce the Dense Convolutional Network (DenseNet), which connects each layer to every other layer in a feed-forward fashion. Whereas traditional convolutional networks with  $L$  layers have  $L$  connections—one between each layer and its subsequent layer—our network has  $\frac{L(L+1)}{2}$  direct connections. For each layer, the feature-maps of all preceding layers are used as inputs, and its own feature-maps are used as inputs into all subsequent layers. DenseNets have several compelling advantages: they alleviate the vanishing-gradient problem, strengthen feature propagation, encourage feature reuse, and substantially reduce the number of parameters. We evaluate our proposed architecture on four highly competitive object recognition benchmark tasks (CIFAR-10, CIFAR-100, SVHN, and ImageNet). DenseNets obtain significant improvements over the state-of-the-art on most of them, whilst requiring less computation to achieve high performance. Code and pre-trained models are available at*

# 요약

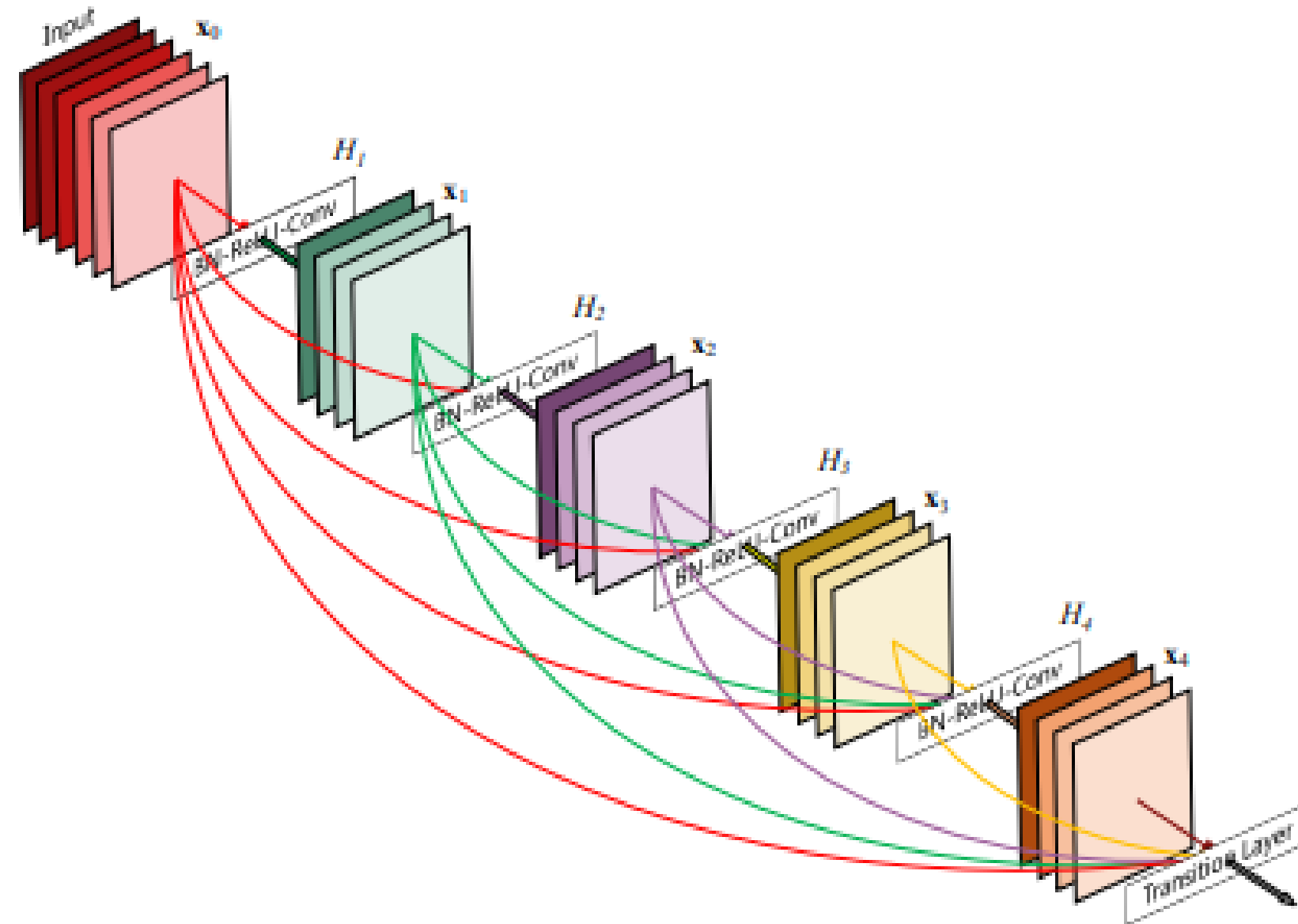
기본 아이디어: 입력층과 출력층 사이 짧은 연결을 포함하면 컨볼루션 신경망이 더 높은 정확성을 땀- 레스넷

덴스넷은 이를 바탕으로 모든 층이 이전의 모든 층의 값을 받아 피드포워드함

기존의 컨볼루션 신경망: L개의 층이 L개의 연결

덴스넷:  $L(L+1) / 2$  개의 연결 존재

장점: 기울기 소실 문제 완화, 층간 특징 전달 강화, 특징 재사용, 파라미터 수 대폭 감소



**Figure 1:** A 5-layer dense block with a growth rate of  $k = 4$ . Each layer takes all preceding feature-maps as input.

# 인트로덕션 중요내용

**vanishing gradient: 기울기 소실 문제, 기존의 컨볼루션 신경망에서 발생하는 문제**

**입력정보, 경사하강법에서의 기울기가 여러 층을 거치며 소실되는것**

이를 해결하기 위해 레스넷은 아이덴티티  $x$ 를 더해 정보를 전달함

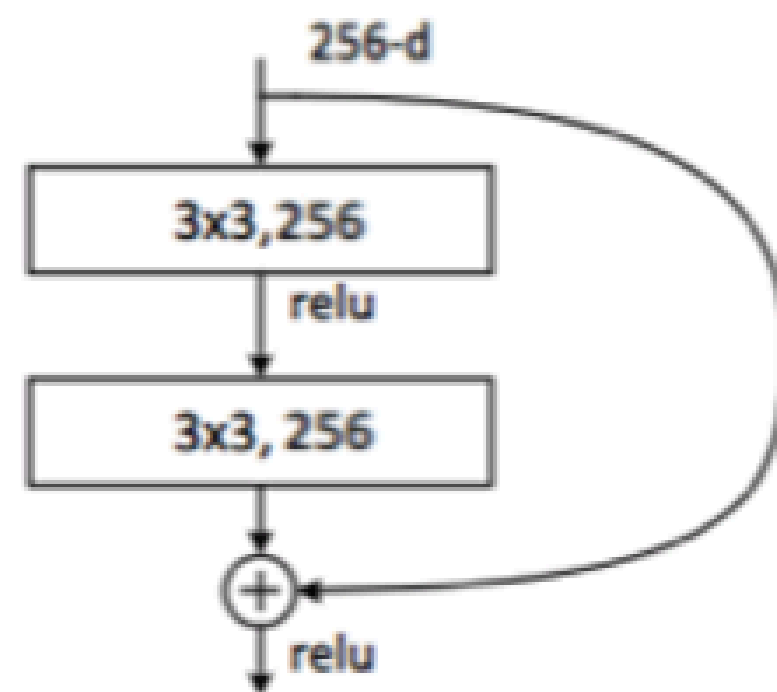
덴스넷은 네트워크의 모든 층을 연결함, 각 층이 이전 층의 출력을 입력으로 사용함

레스넷과의 차이: 아이덴티티 덧셈이 아닌, 층간의 연결을 통해 진행

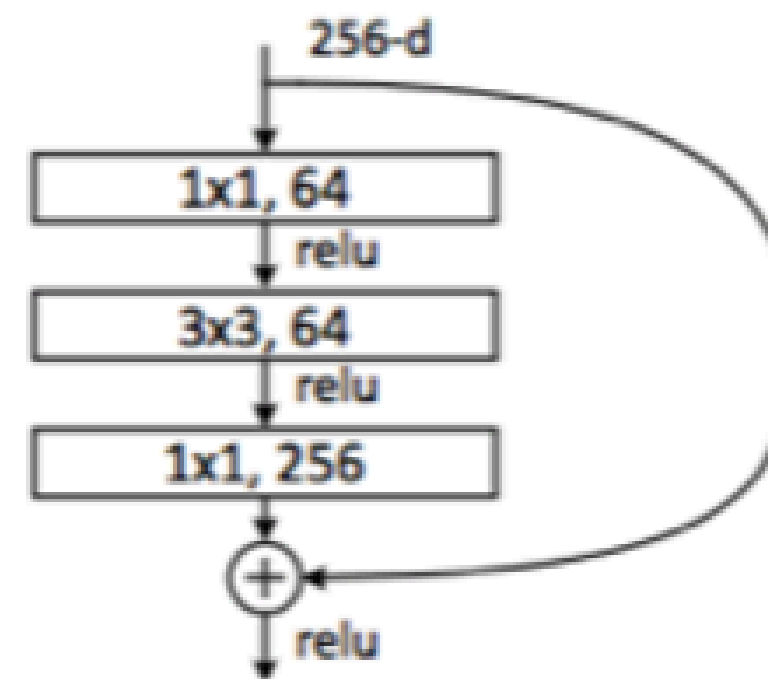
장점: 파라미터 효율성 - 중복된 특징 맵을 다시 학습할 필요가 없으므로 파라미터 수가 감소.

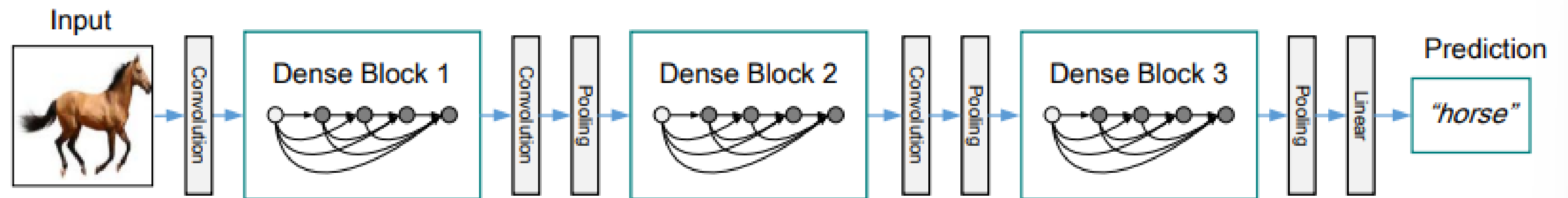
적은 파라미터를 통해 과적합 방지 효과도 있음

ResNet 34  
residual block



ResNet 50  
residual block





**Figure 2:** A deep DenseNet with three dense blocks. The layers between two adjacent blocks are referred to as transition layers and change feature-map sizes via convolution and pooling.



# DenseNet

**층 구성: Dense Block, Transition Layer, Bottleneck Layer**

**Dense layer: 동일 블록 내에서 이전 층의 출력(feature map)을 입력으로 사용하고 자신의 출력을 모든 층에 전달**

**Transition layer: Dense Block 간의 연결**

**Bottleneck Layer: Dense Block 내부에서 사용되는  $1 \times 1$  Convolution 레이어**

# 추가학습

## Bottleneck Layer

Resnet bottleneck : 3x3 convolution 전에 1x1 convolution을 거쳐서 입력 feature map의 channel 개수를 줄이고, 다시 입력 feature map의 channel 개수 만큼을 생성

Dense bottleneck : 3x3 convolution 전에 1x1 convolution을 거쳐서 입력 feature map의 channel 개수를 줄이고, growth rate 만큼의 feature map을 생성

## Transition Layer

feature map의 가로, 세로 사이즈를 줄여주고 feature map의 개수를 줄여주는 역할을 담당

Batch Normalization, ReLU, 1x1 convolution, 2x2 average pooling 으로 구성

1x1 convolution을 통해 feature map의 개수를 줄여주며 이 때 줄여주는 정도를 나타내는 theta 를 논문에서는 0.5를 사용하였으며 마찬가지로 이 값도 hyper-parameter입니다. 이 과정을 Compression이라 논문에서 표현

ResNet : 직전 레이어의 feature map을 한번 더해서 input에 넣는 방식

DenseNet : 이전 전체 레이어 각각의 feature map을 묶어서 input에 넣는 방식  
( feature map 끼리 Concatenation(연쇄))

# 추가학습

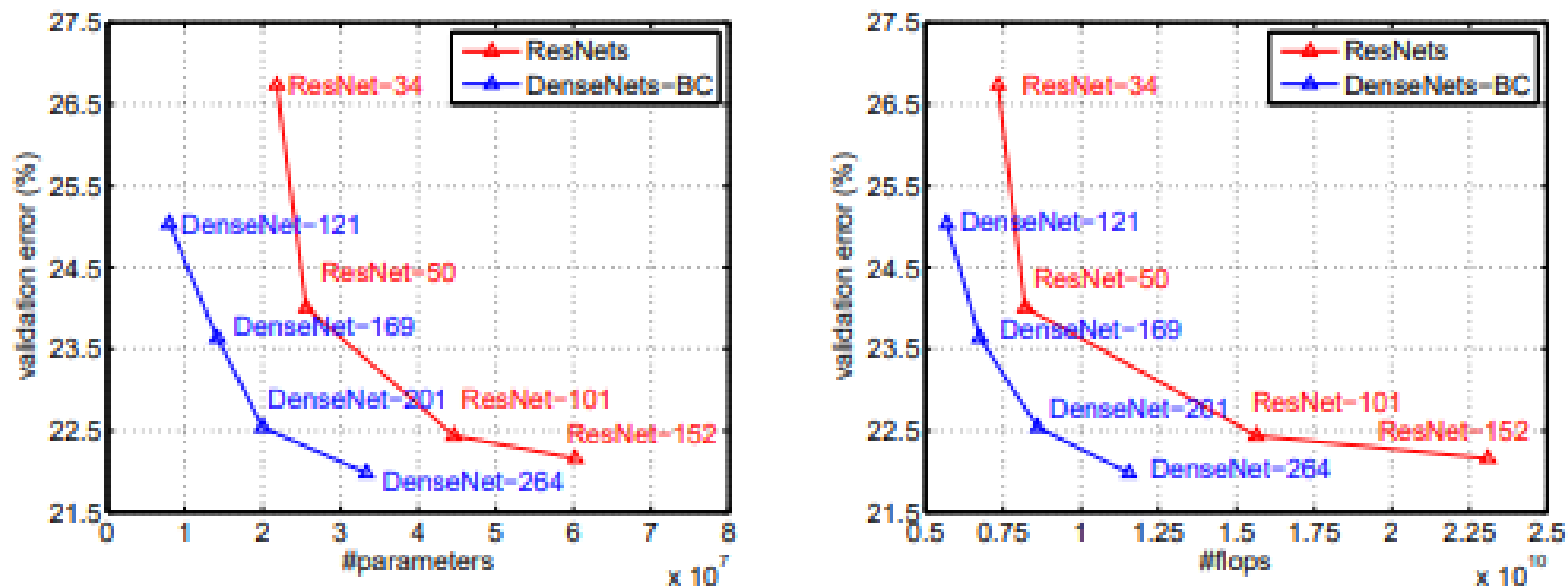
마지막에 분류 헤드(Classification Layer)에서 풀링을 통해서 신경망 층의 맵 크기를 줄이는 것 같다  
-Softmax 활성화 함수를 통해 최종 확률 출력.

보틀넥: 컨볼루션 내부 학습층

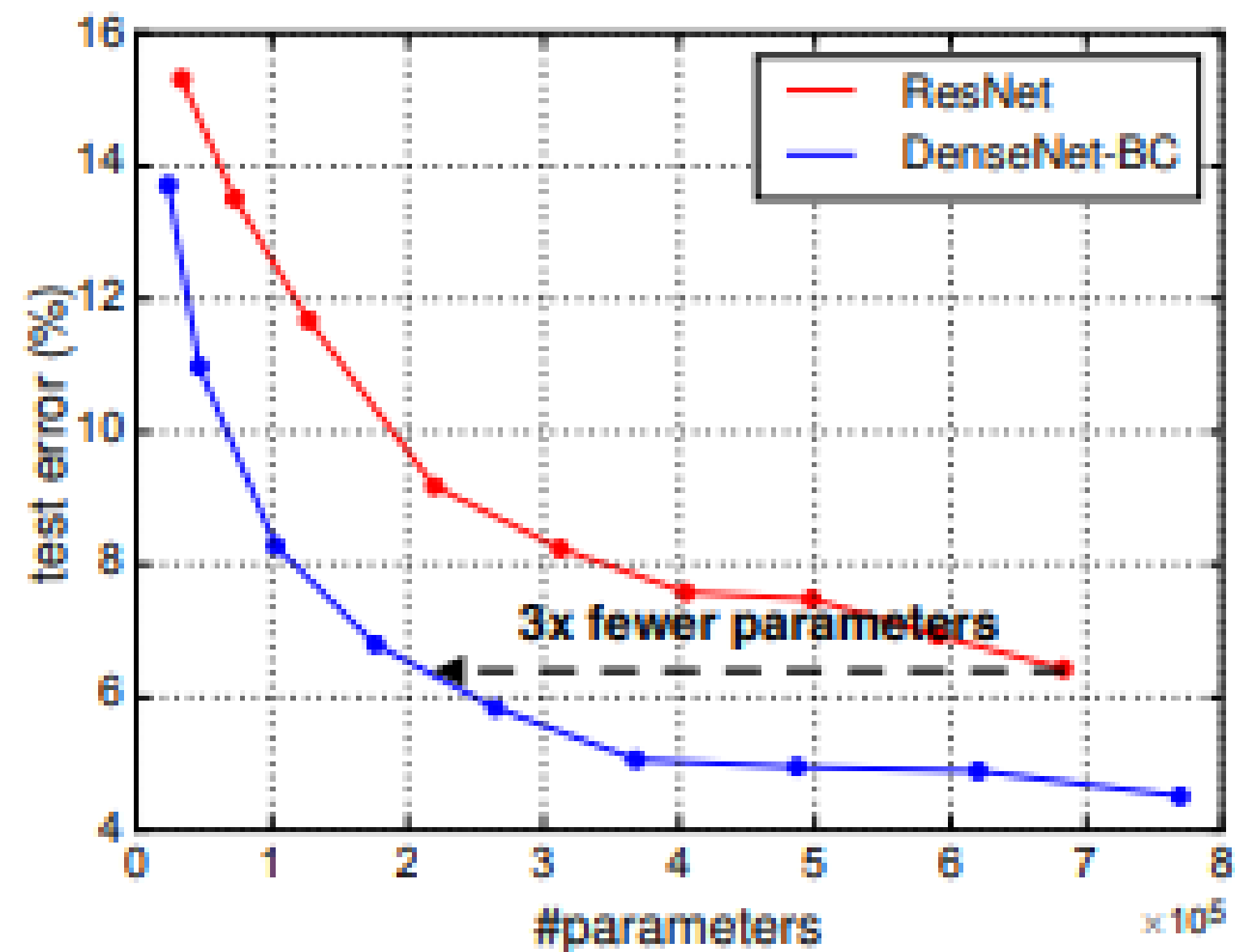
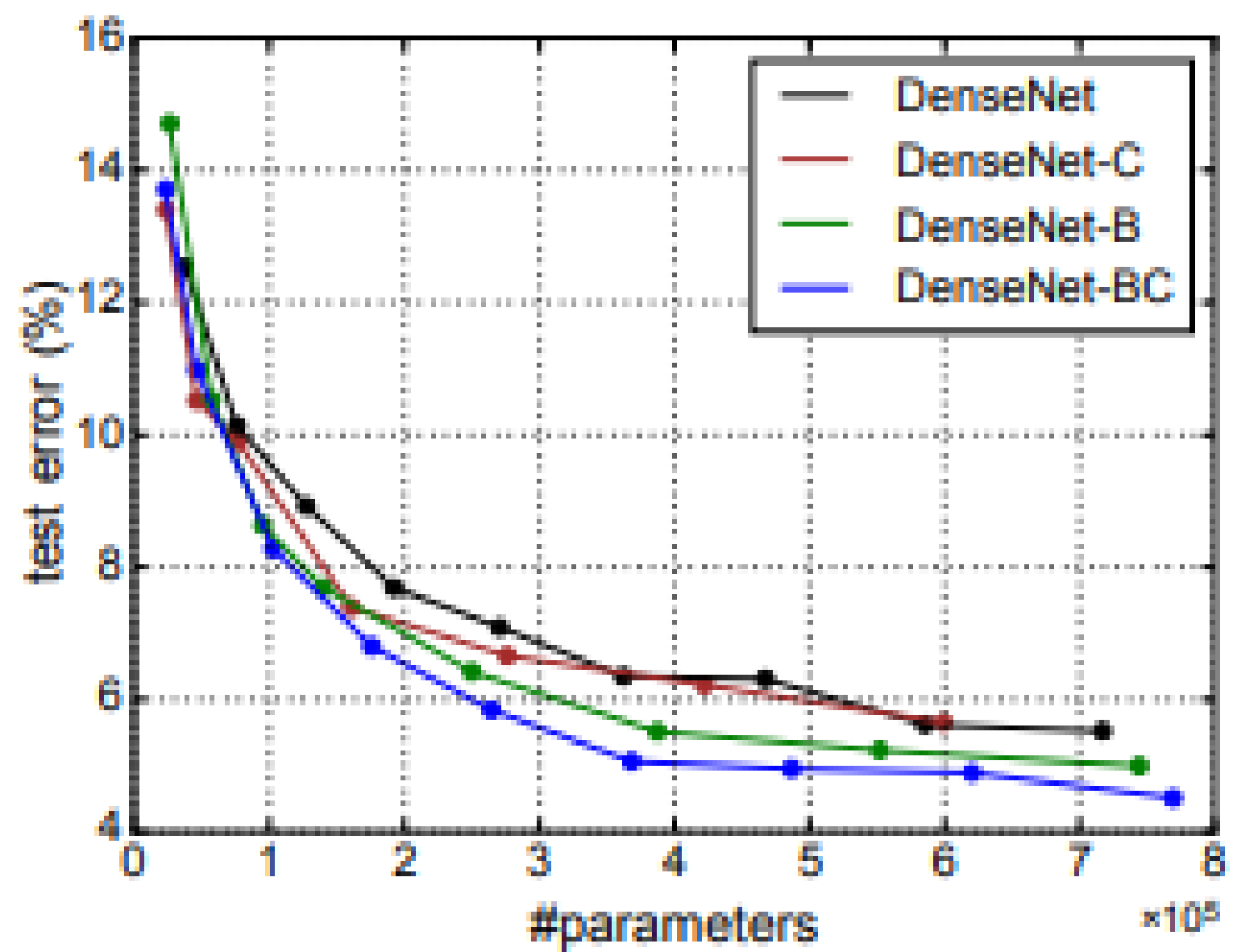
트랜지션: 층간 연결

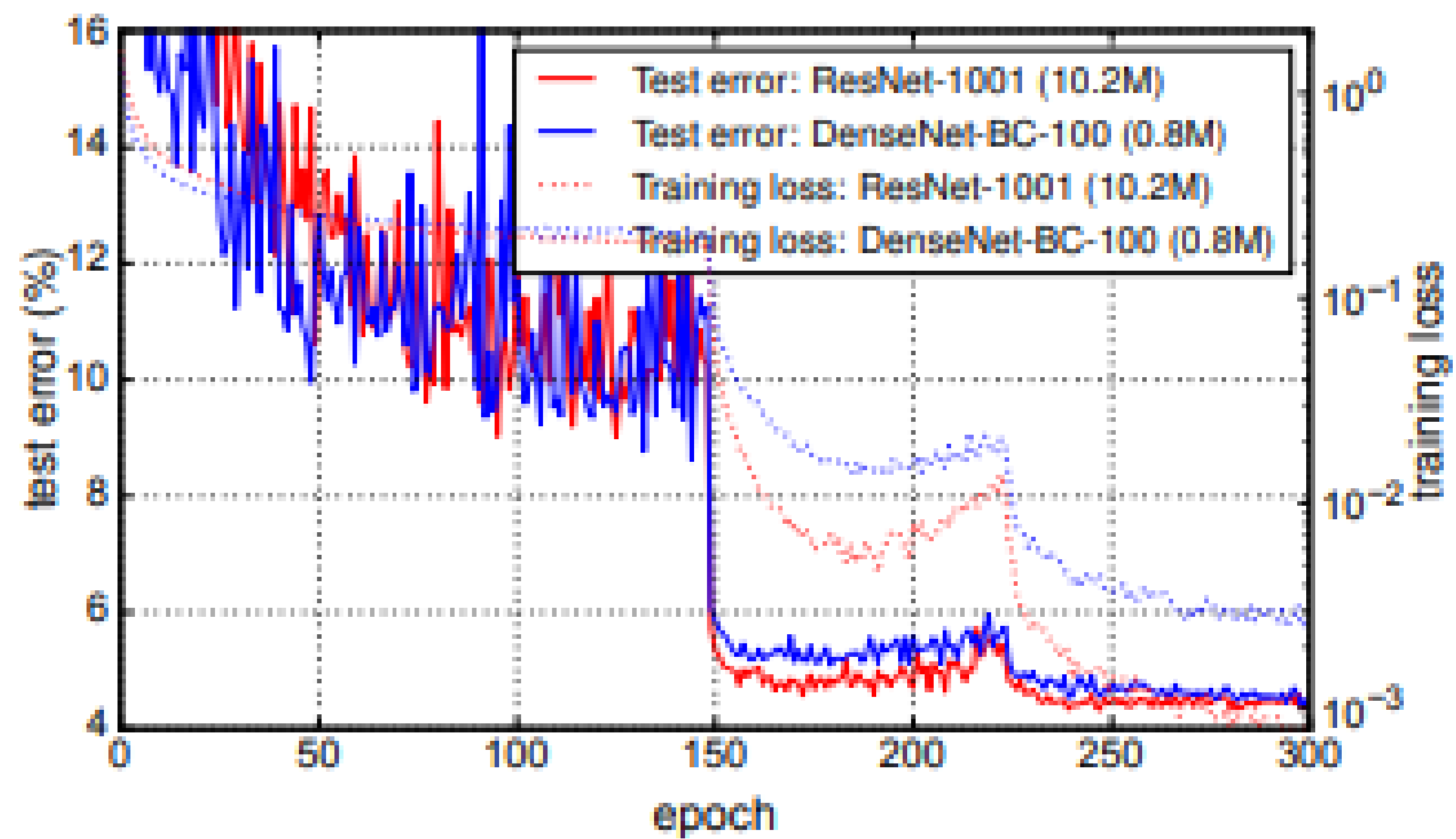
Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	$112 \times 112$	$7 \times 7$ conv, stride 2			
Pooling	$56 \times 56$	$3 \times 3$ max pool, stride 2			
Dense Block (1)	$56 \times 56$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	$56 \times 56$	$1 \times 1$ conv			
	$28 \times 28$	$2 \times 2$ average pool, stride 2			
Dense Block (2)	$28 \times 28$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	$28 \times 28$	$1 \times 1$ conv			
	$14 \times 14$	$2 \times 2$ average pool, stride 2			
Dense Block (3)	$14 \times 14$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 64$
Transition Layer (3)	$14 \times 14$	$1 \times 1$ conv			
	$7 \times 7$	$2 \times 2$ average pool, stride 2			
Dense Block (4)	$7 \times 7$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$
Classification Layer	$1 \times 1$	$7 \times 7$ global average pool			
		1000D fully-connected, softmax			

**Table 1:** DenseNet architectures for ImageNet. The growth rate for all the networks is  $k = 32$ . Note that each “conv” layer shown in the table corresponds the sequence BN-ReLU-Conv.



**Figure 3:** Comparison of the DenseNets and ResNets top-1 error rates (single-crop testing) on the ImageNet validation dataset as a function of learned parameters (*left*) and FLOPs during test-time (*right*).





**Q&A**



**THANKS**