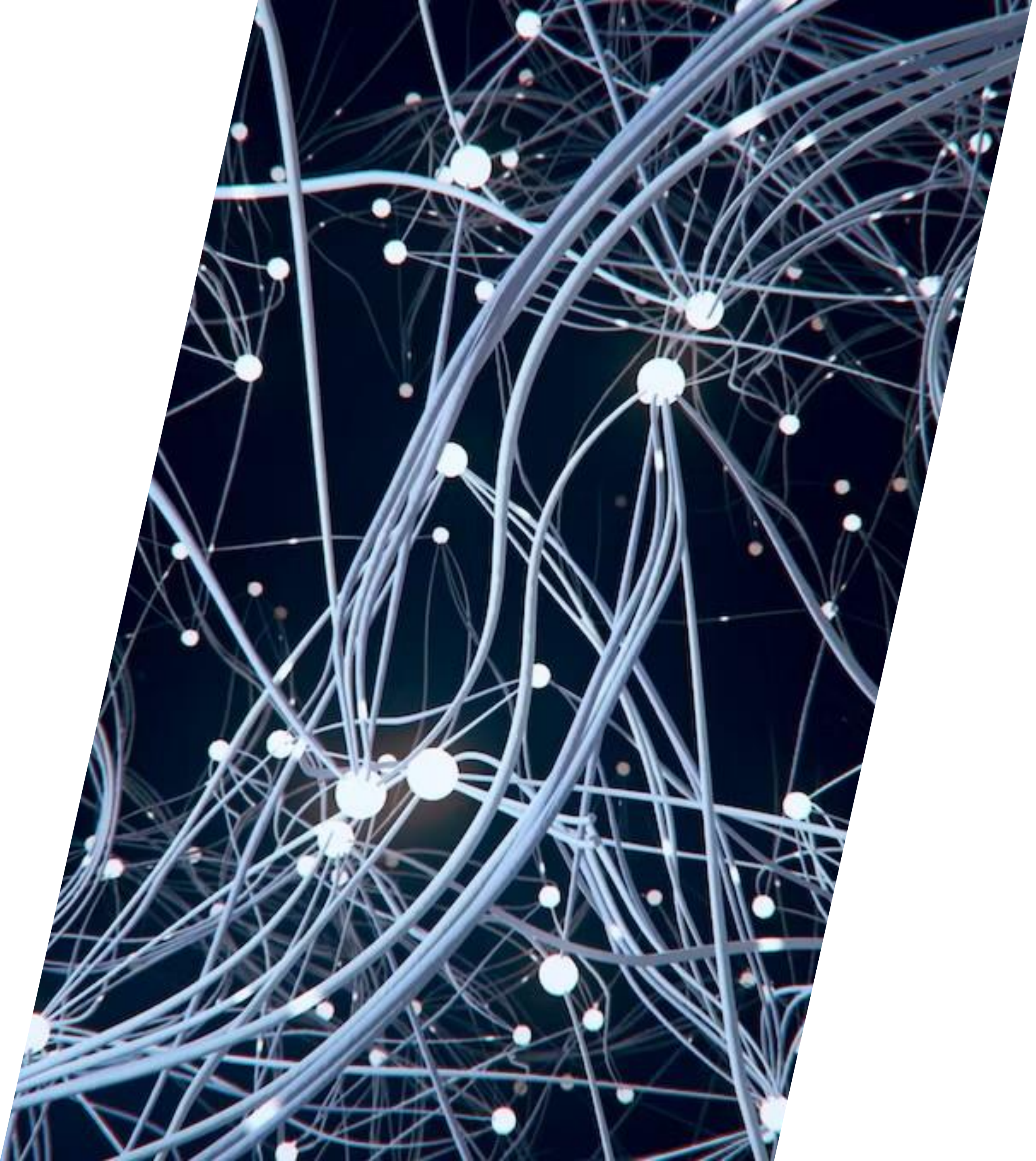
DEEP RESIDUAL LEARNING

RESNET

컴퓨터소프트웨어학부
2021088304
박현준



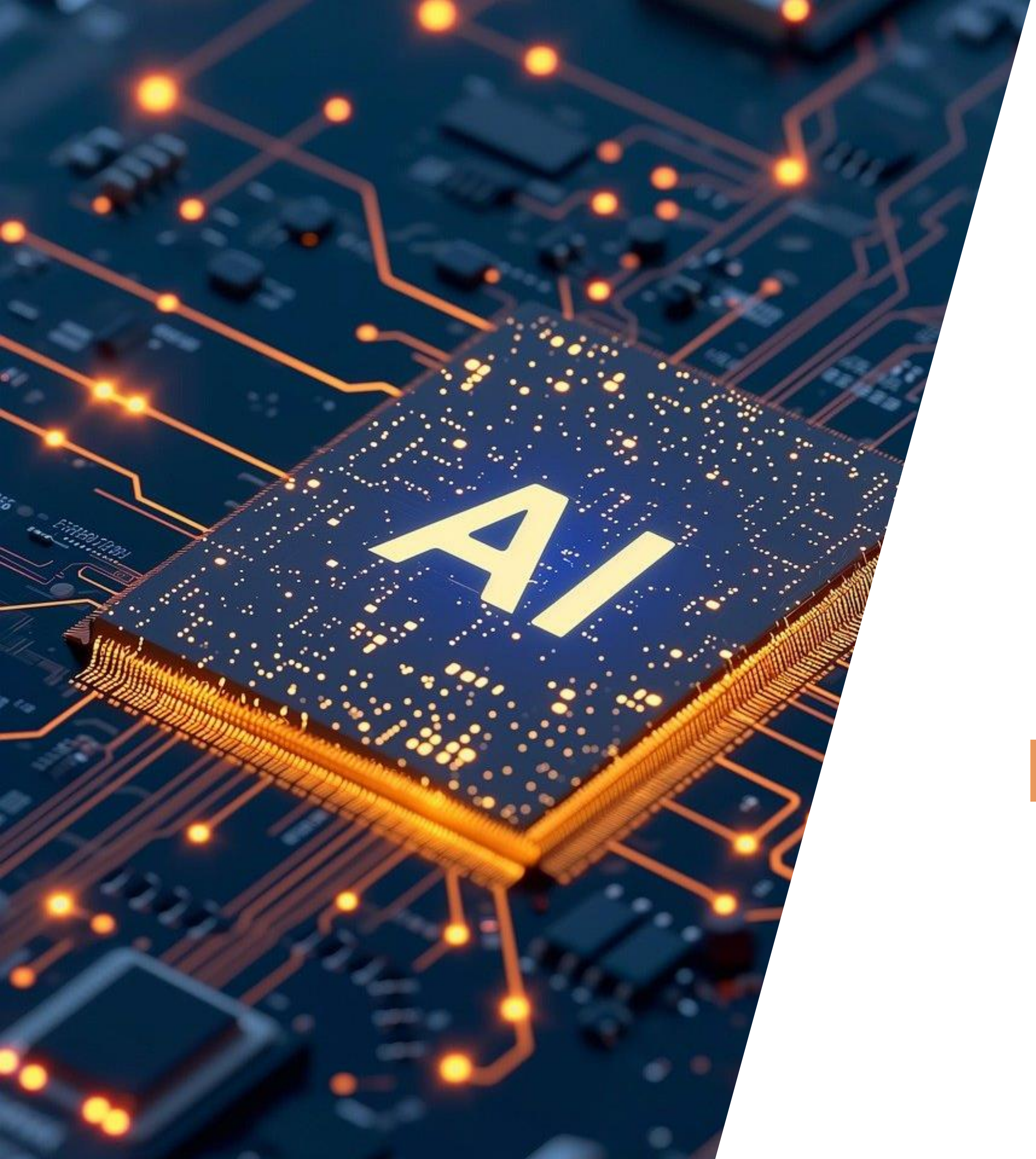
i . Introduction

ii . Deep Residual Learning

iii . Architecture

iv . Experiment

v . Conclusion



INTRODUCTION

CHAPTER 1

INTRODUCTION

Background

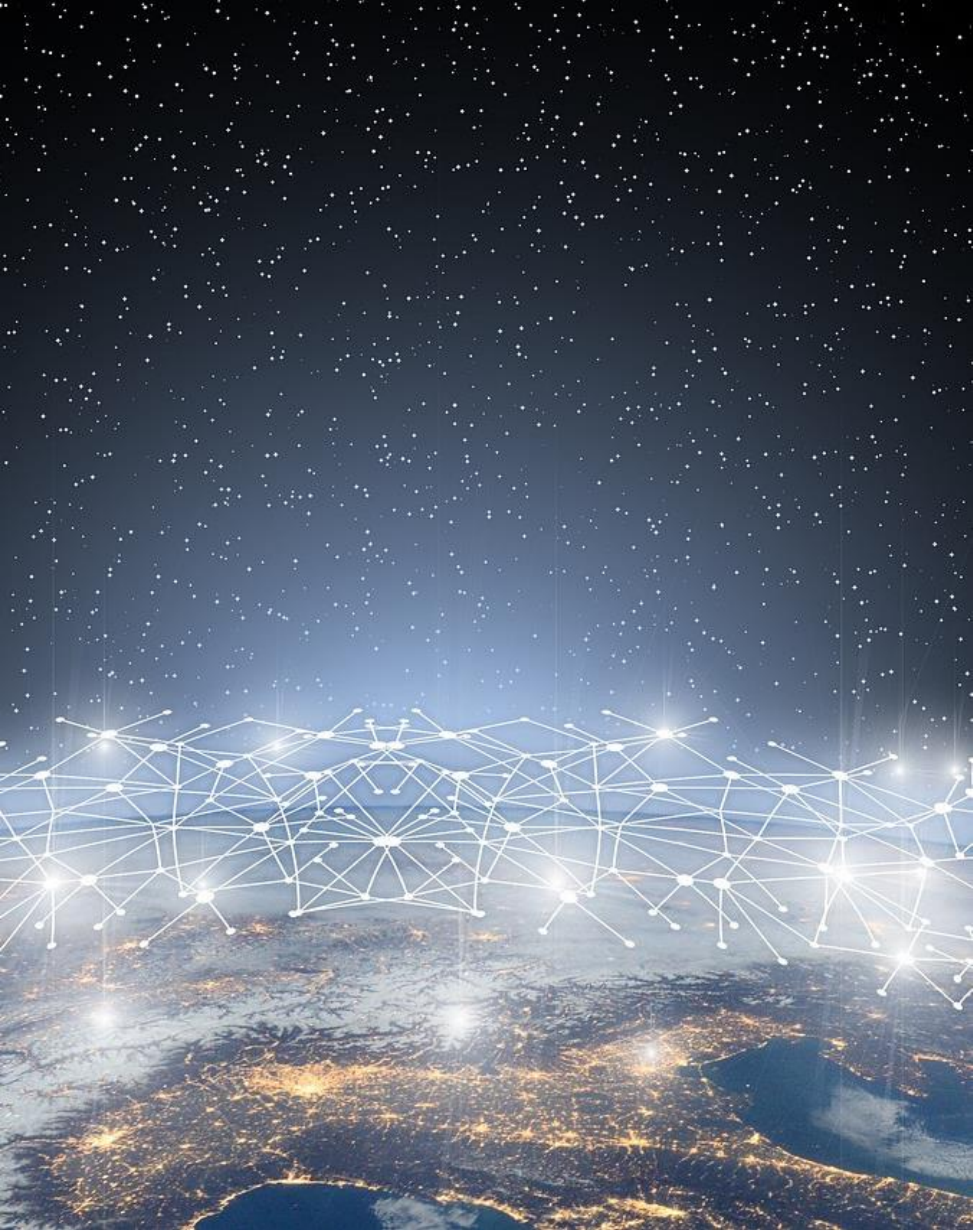
Convolutional Neural Network (CNN)

↓ key

Computer Vision

&

Image processing



INTRODUCTION

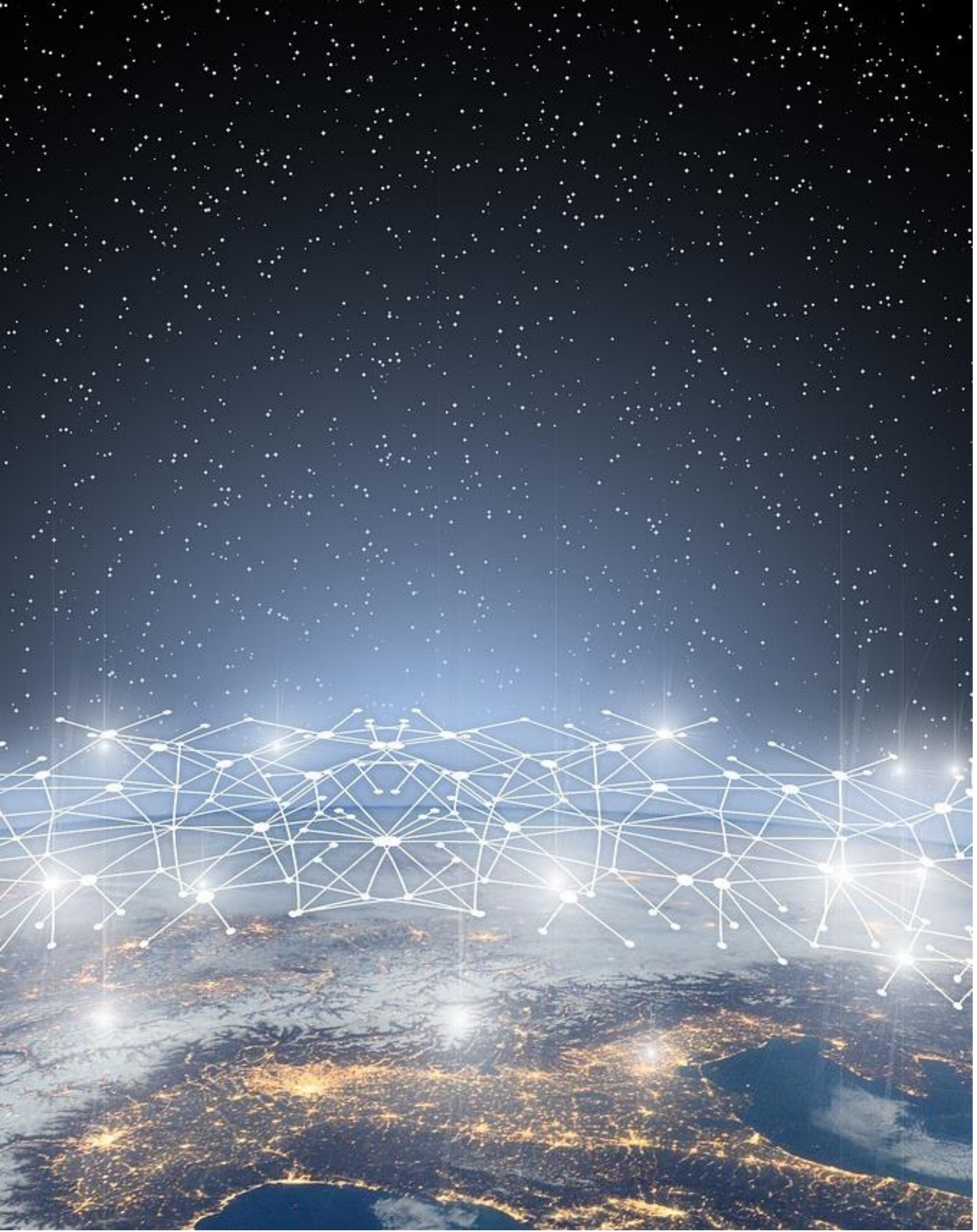
Question

#Layer ↑

= =

Network performance ↑

Is better networks as easy as stacking more layers?



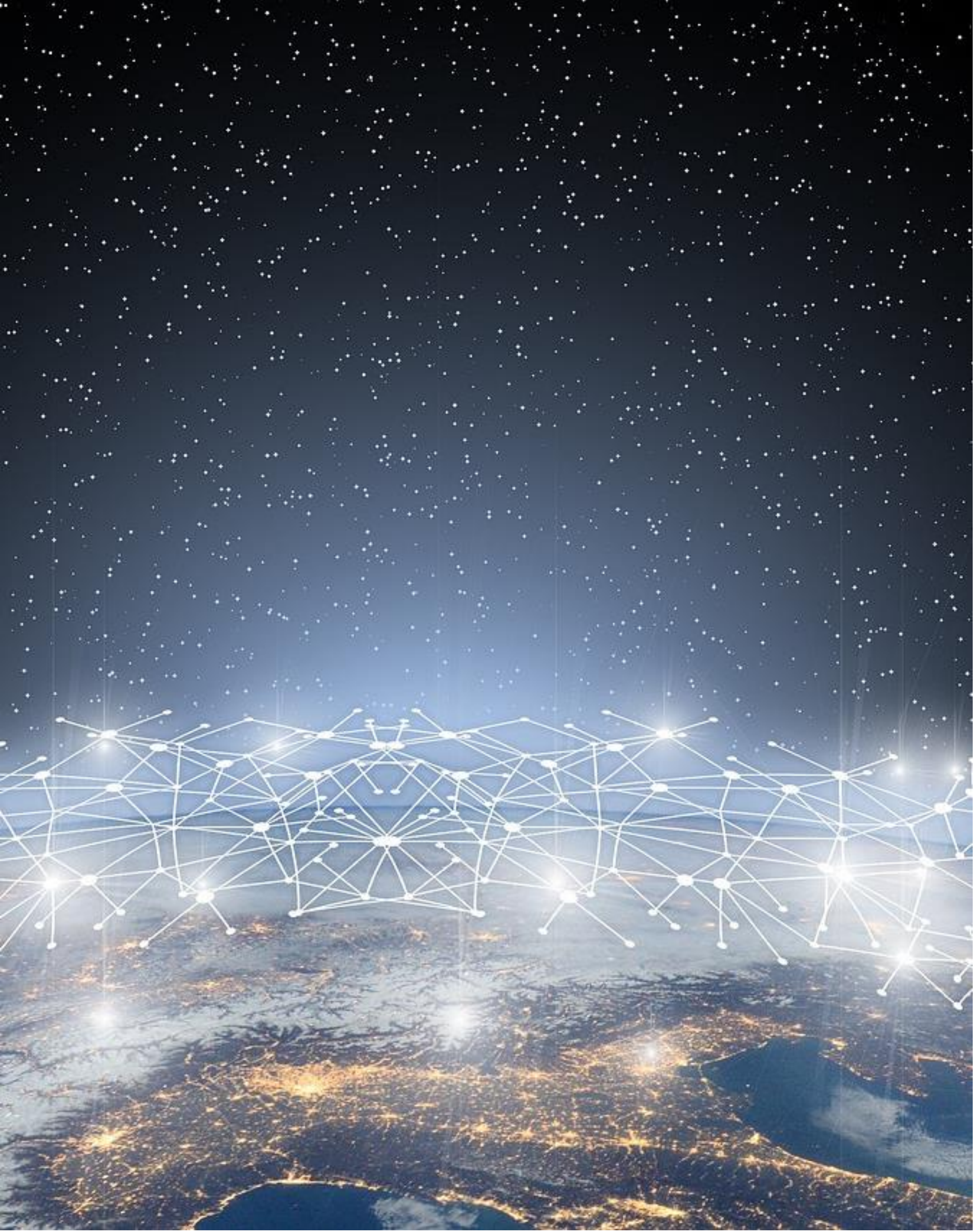
INTRODUCTION

Problem

Exploding / Vanishing

Gradient

(This problem, however, has been largely addressed
by normalized initialization [23, 9, 37, 13] and
intermediate normalization layers [16],
which enable networks with tens of layers to start converging
for stochastic gradient descent (SGD) with backpropagation [22].)



INTRODUCTION

Degradation

Degradation Problem

: #Layer ↑ == Performance ↓

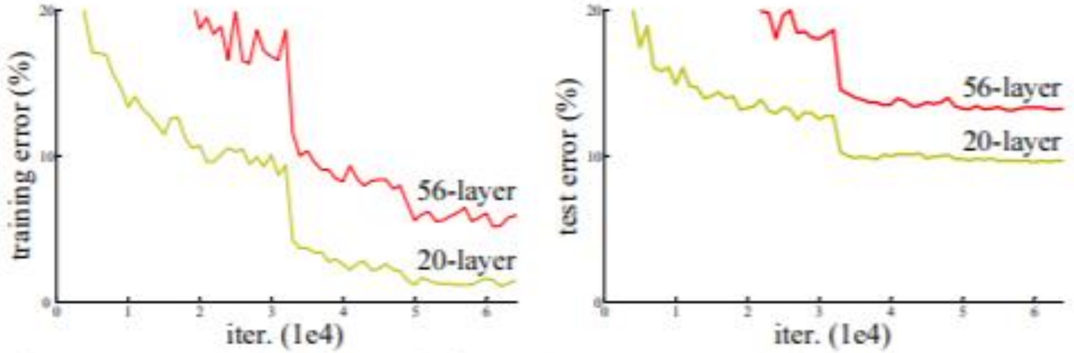
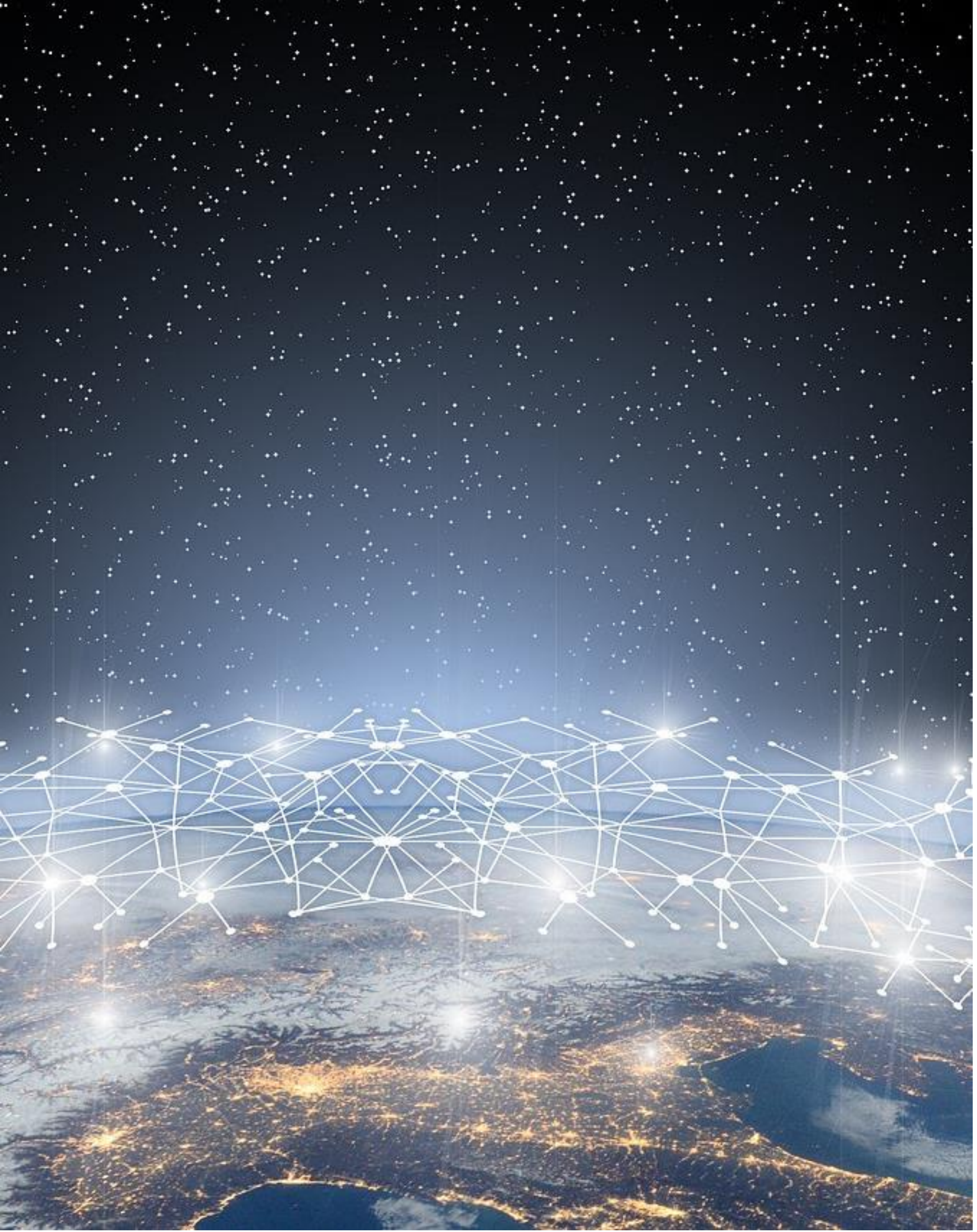


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

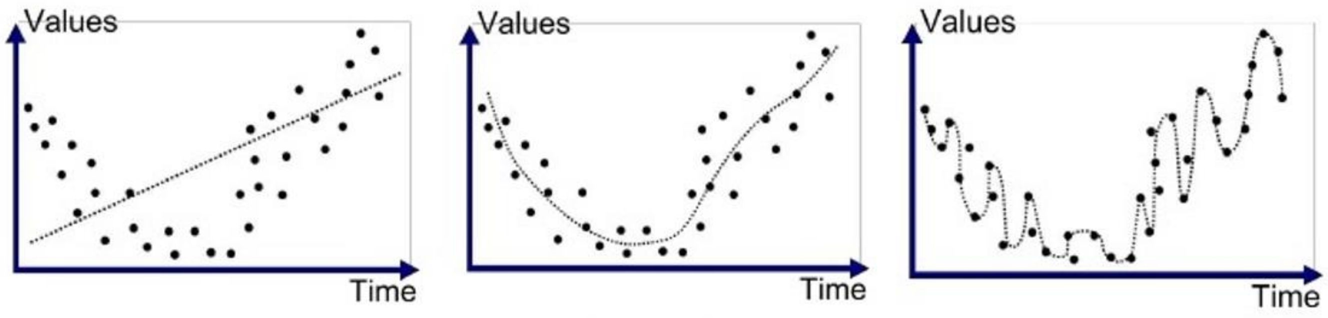


INTRODUCTION

Attention to ResNet

Reason why ResNet is in spotlight

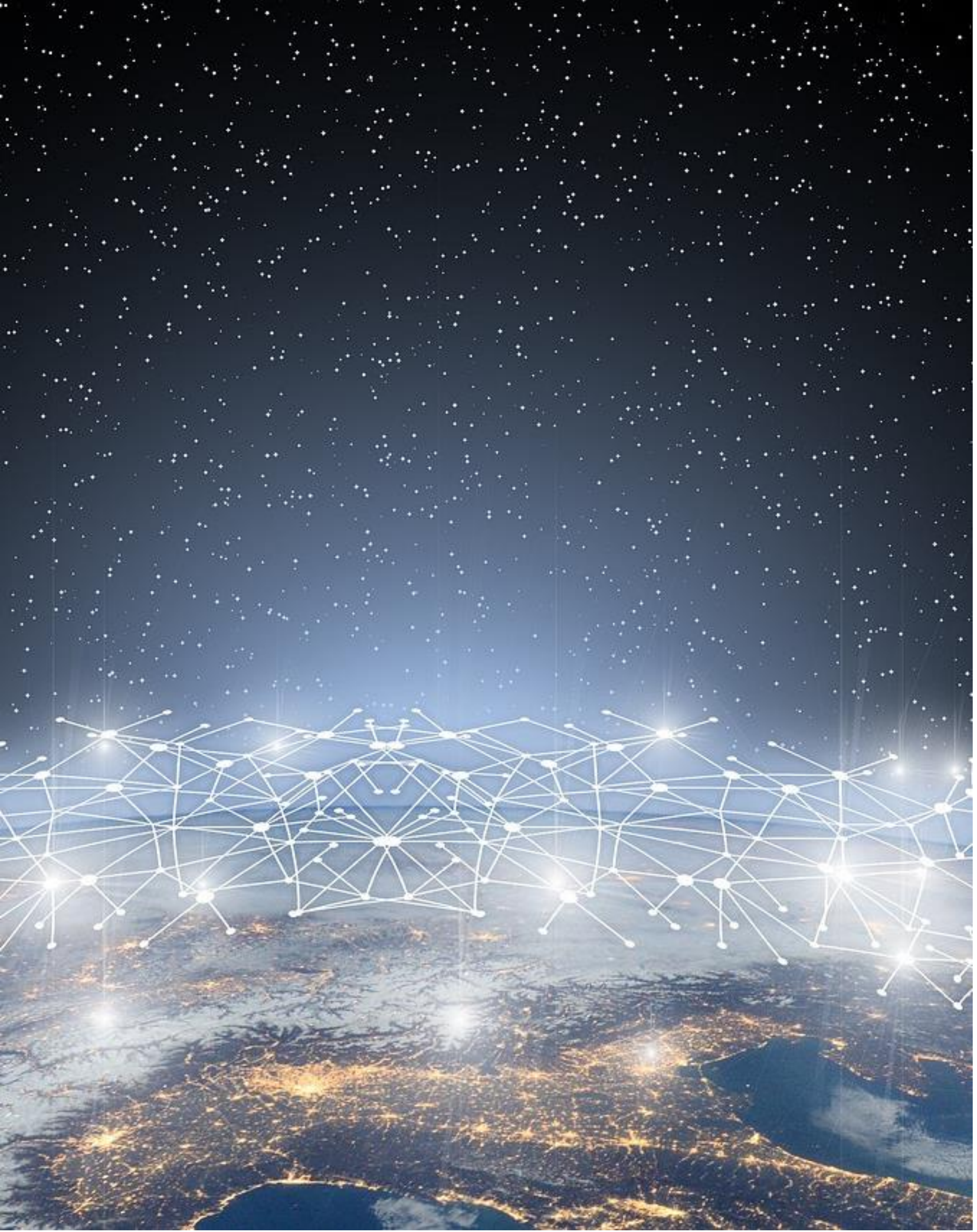
: Nothing to do with overfitting



Underfitted

Good Fit/Robust

Overfitted



INTRODUCTION

Idea from Degradation

Result by degradation problem

"Not all systems are similarly easy to optimize"

↓ **Solution**

Construction to the deeper model

"the added layers are identity mapping, and the other layers are copied from the learned shallower model"

→ Not good Solution

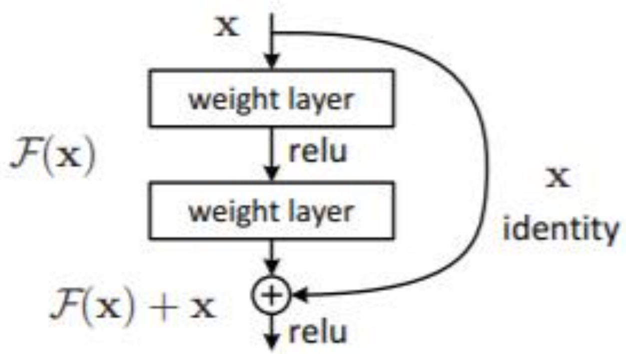


Figure 2. Residual learning: a building block.

INTRODUCTION

Idea

$$F(x) = H(x) - x$$

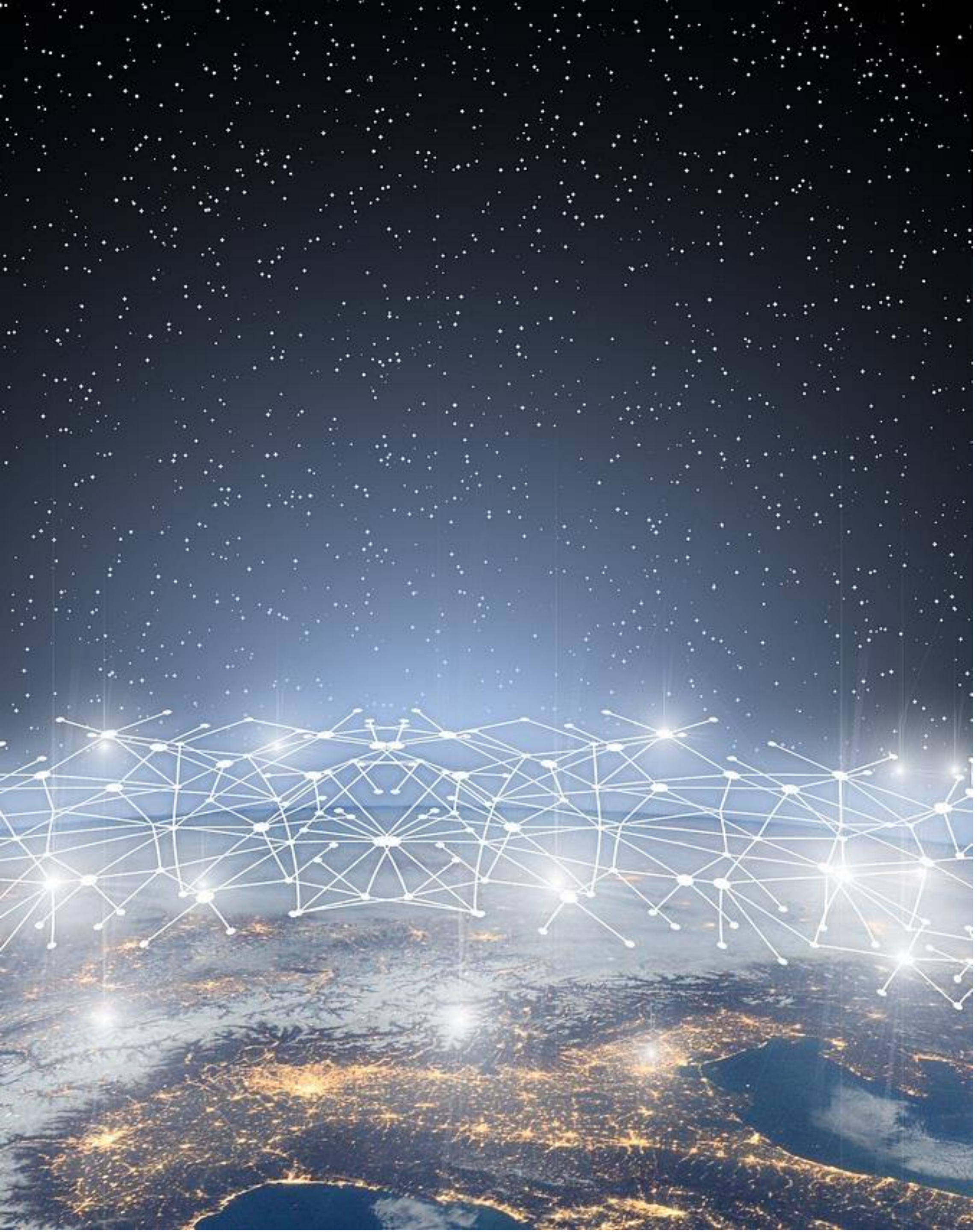
$$H(x) = F(x) + x$$

Postulate residual mapping is suitable for optimization

$F(x) + x$ = **Shortcut Connection**

→ **Skip layers one or more**

- Extra Parameter **X**
- Computational Complexity **X**



INTRODUCTION

Idea

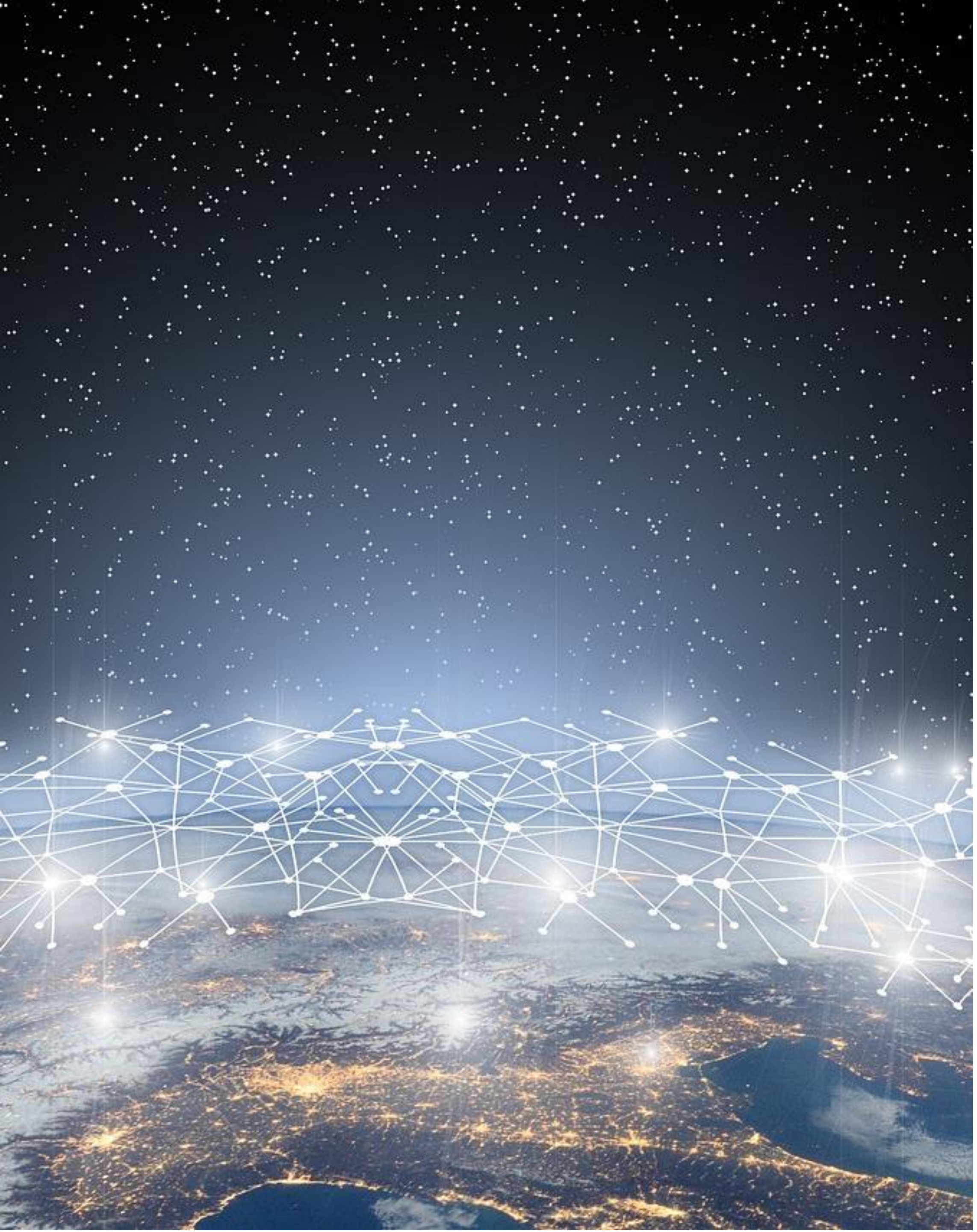
$$H(x) = F(x) + x$$

$$\downarrow$$

$$H'(x) = F'(x) + 1$$

- Vanishing gradient problem cannot occur

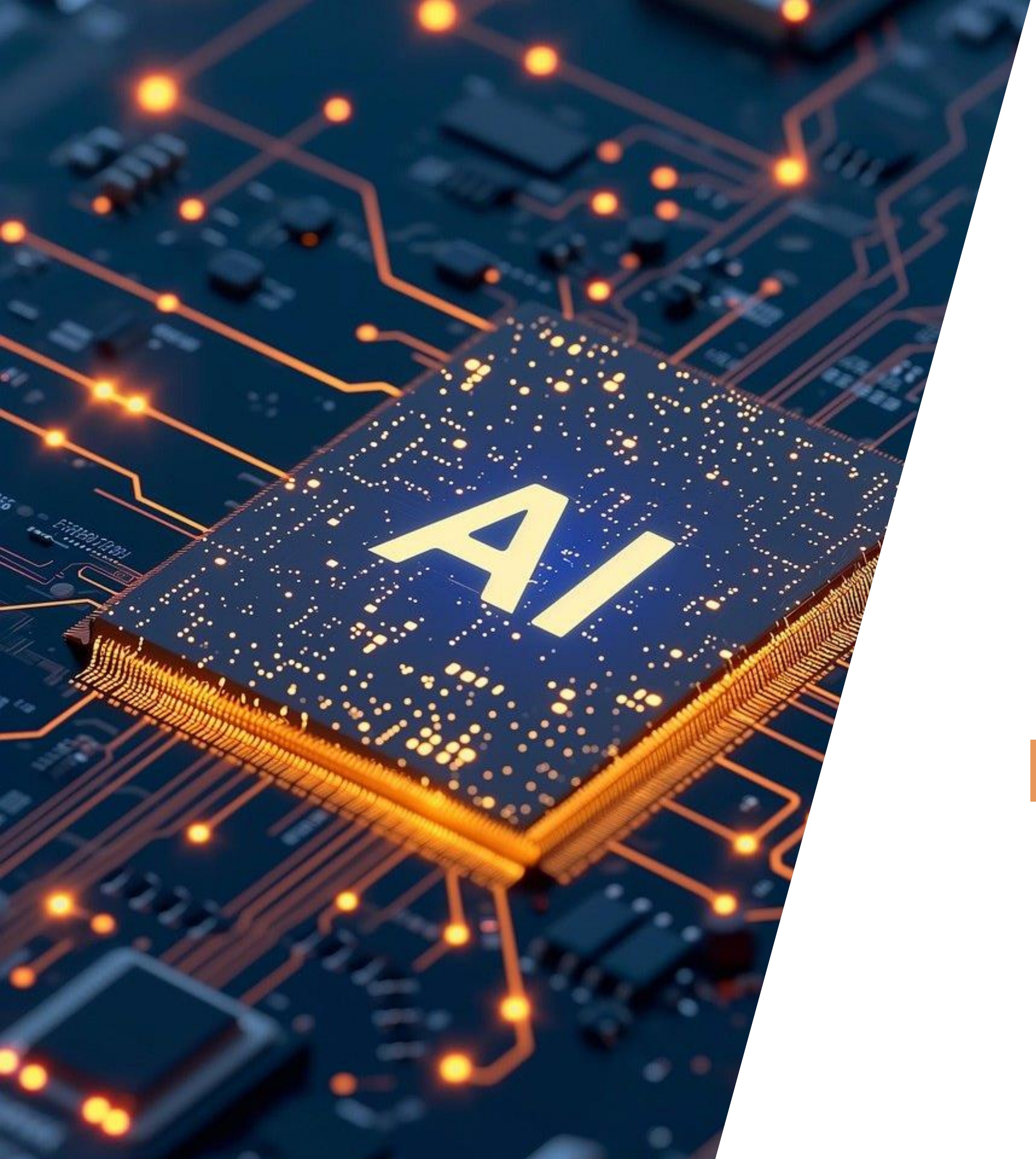
∵ Residual function's gradient always 1 or more



INTRODUCTION

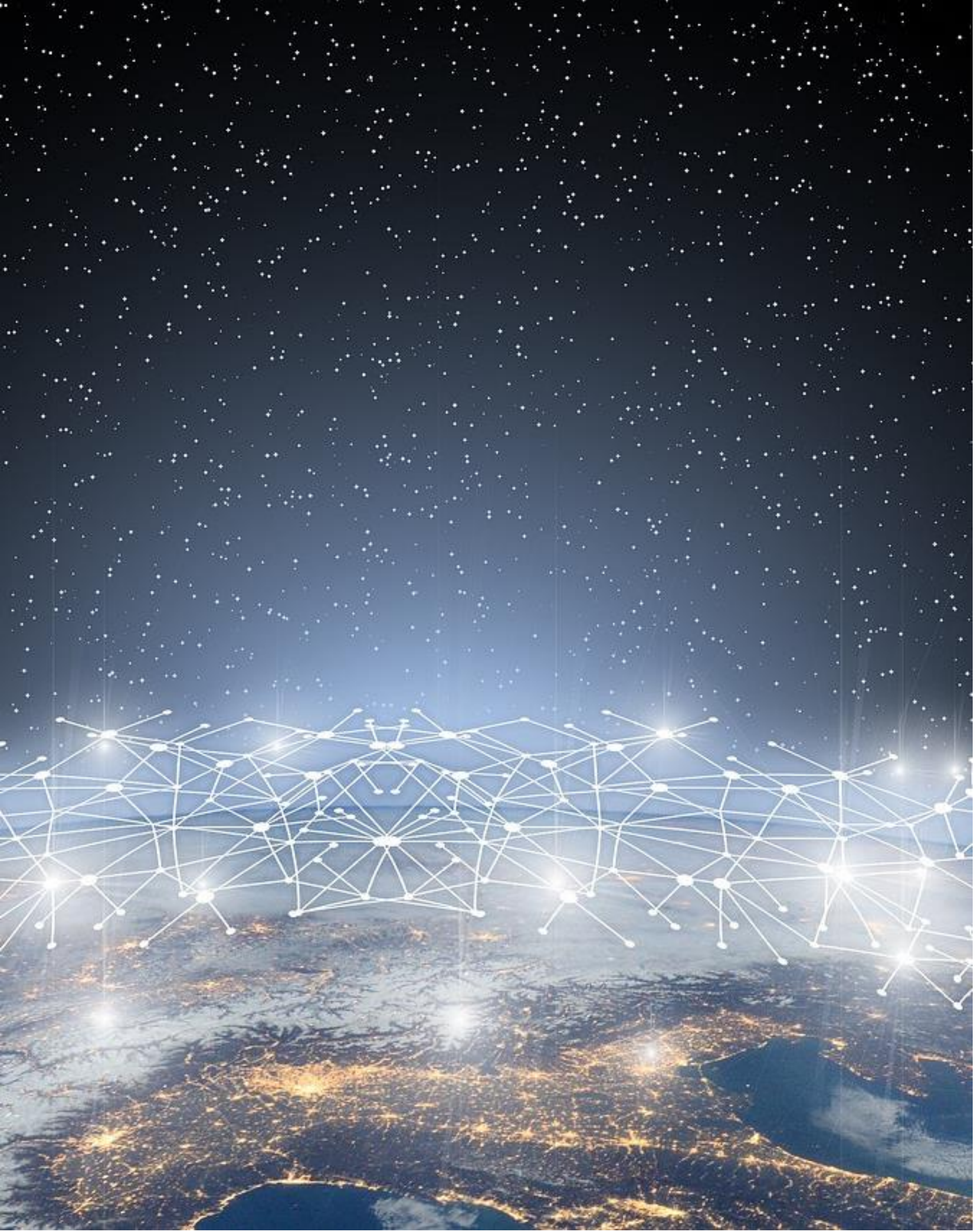
Goal

1. **Prove optimization is easier in ResNet than Plain Net**
2. **Accuracy of ResNet is proportional to depth of network**



DEEP RESIDUAL LEARNING

CHAPTER 2



DEEP RESIDUAL LEARNING

Residual

Residual

(A) 남은, 잔여의

(M) $|\hat{y} - y|$

DEEP RESIDUAL LEARNING

Shortcut

- Identity shortcut

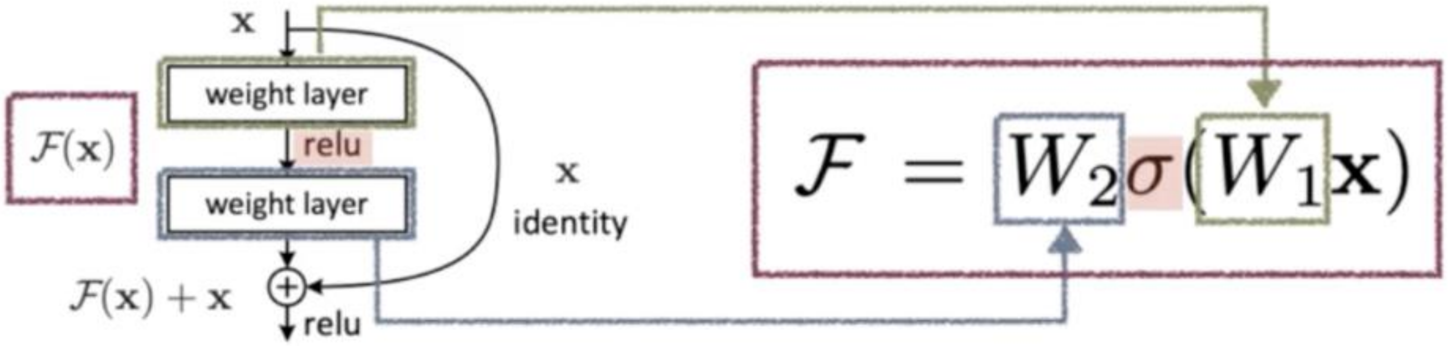
$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + \mathbf{x}.$$

  - y: output vector of the layer
  - x: input (identity) vector of the layer
  - W: weight

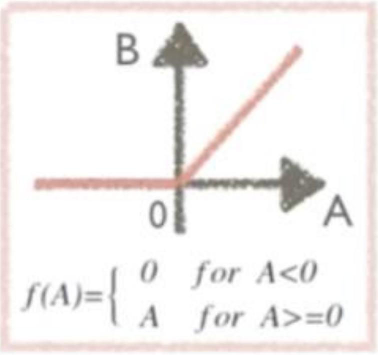
- Projection shortcut

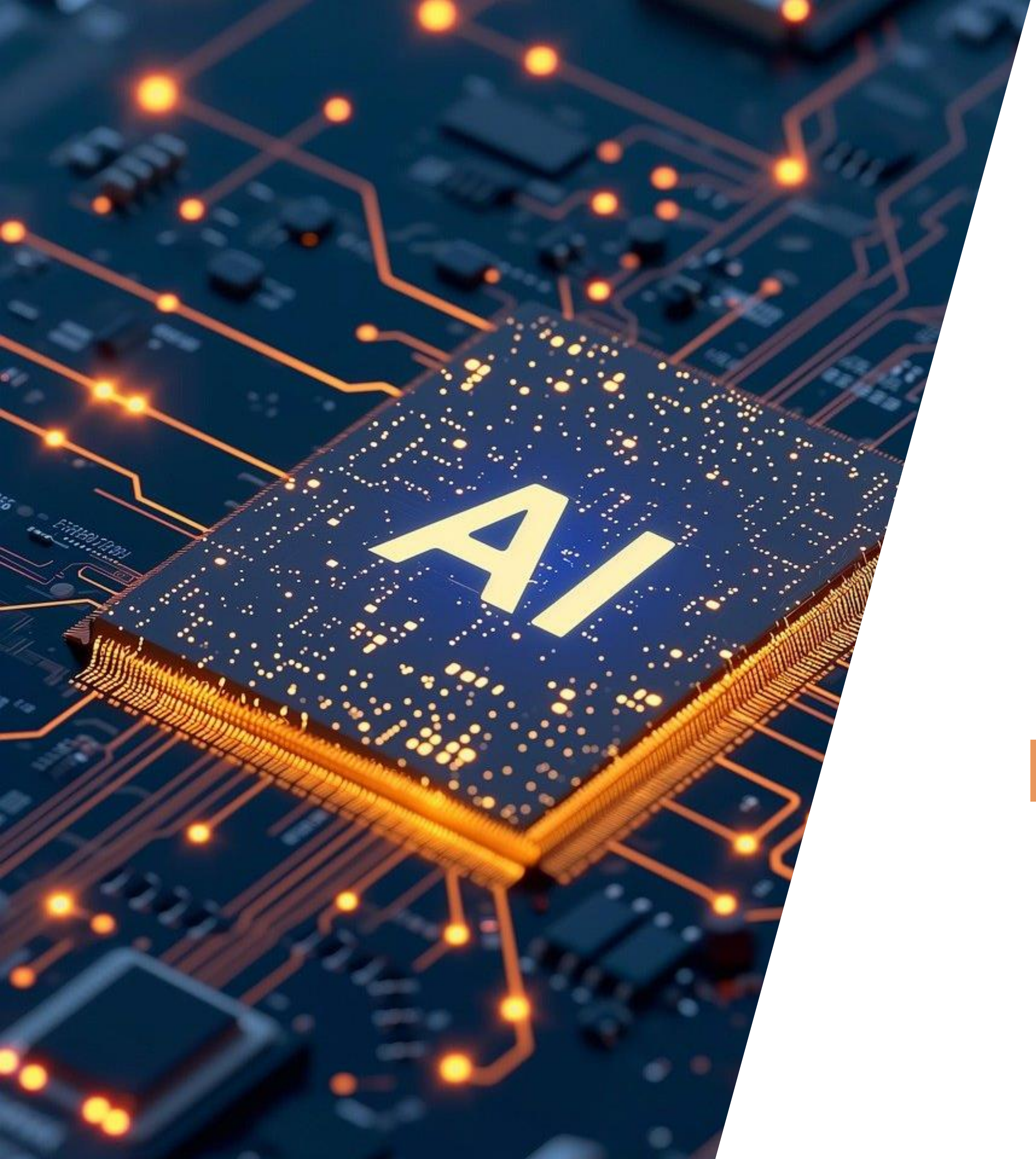
$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + W_s\mathbf{x}$$

  - y: output vector of the layer
  - x: input vector (identity) of the layer
  - W: weight
  - s: square matrix



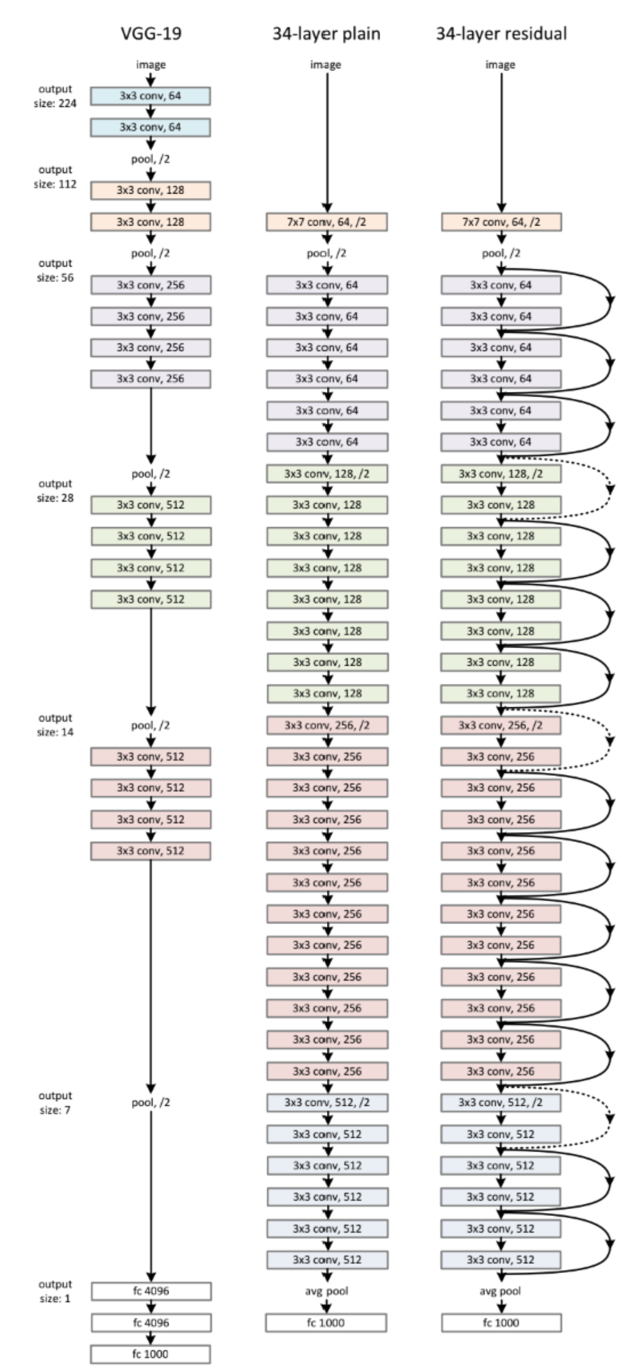
ReLU



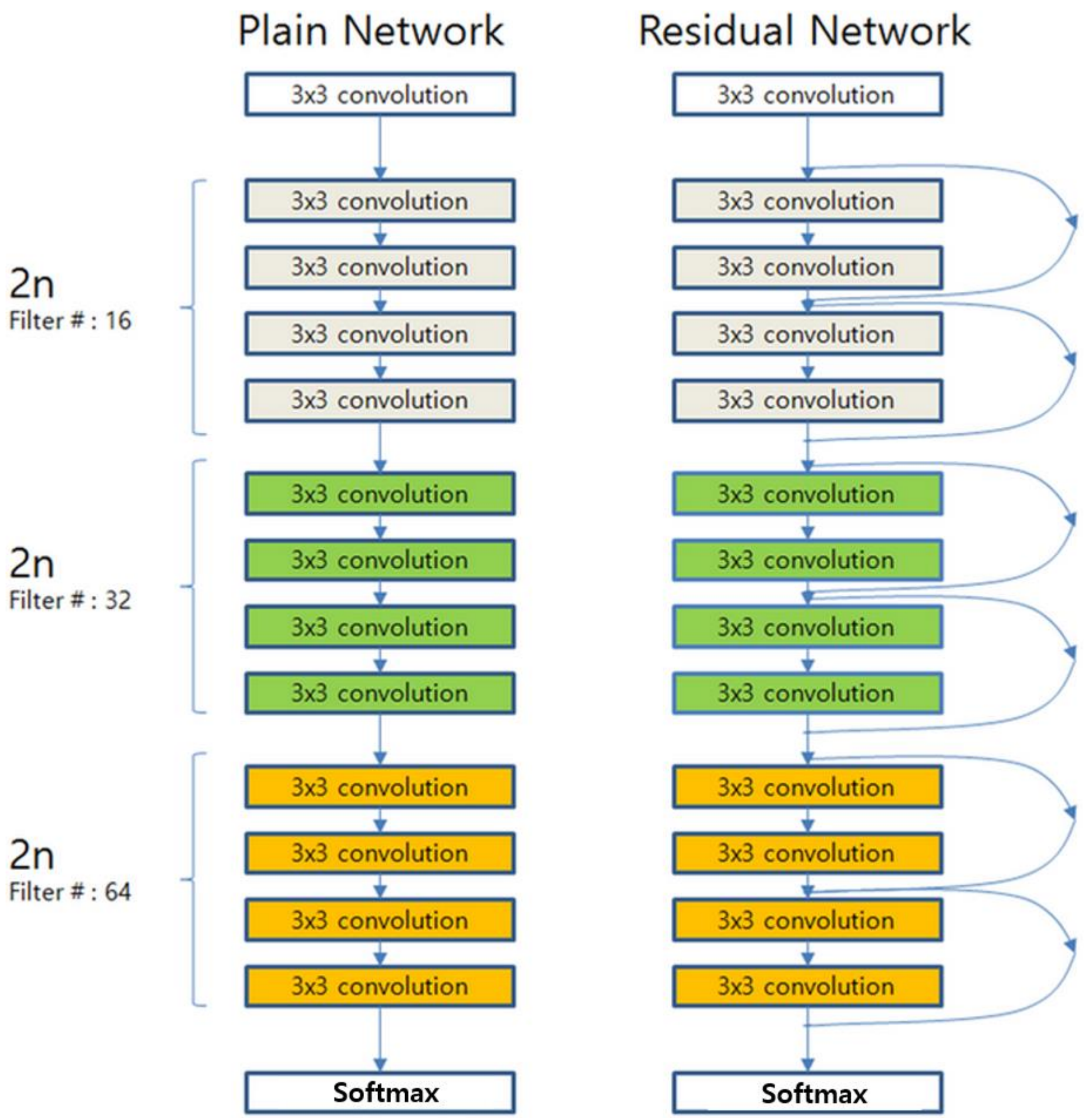


ARCHITECTURE

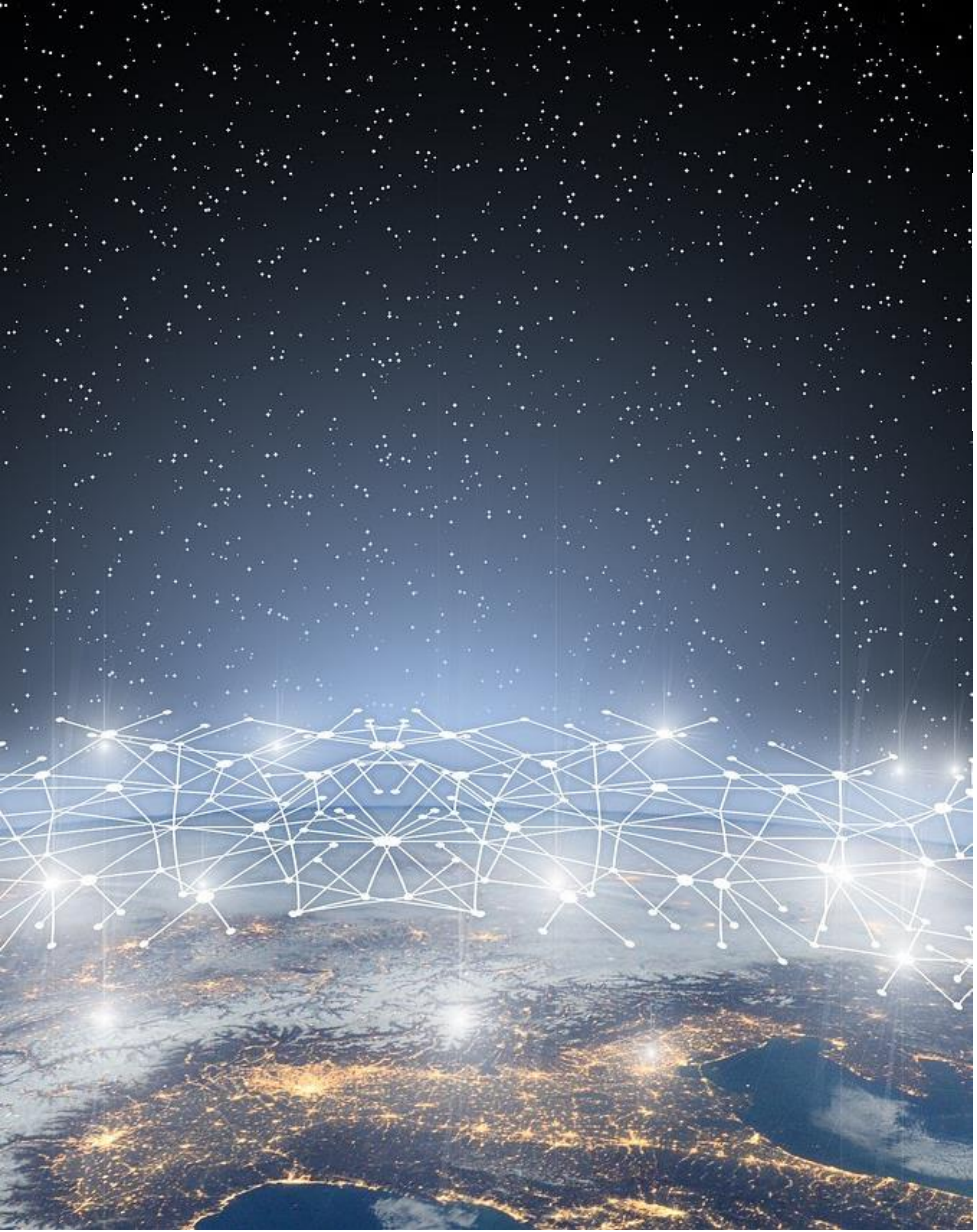
CHAPTER 3



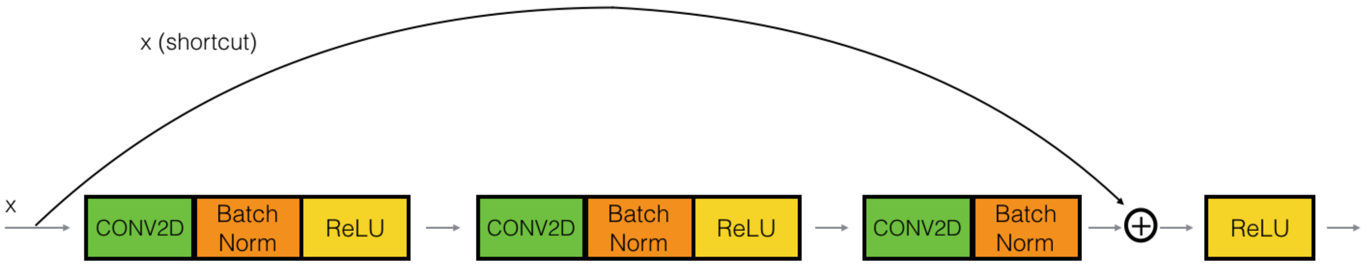
Architecture

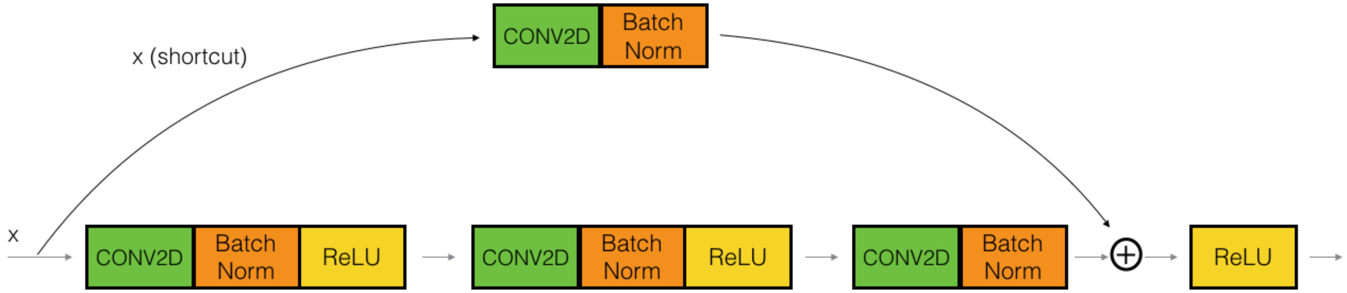


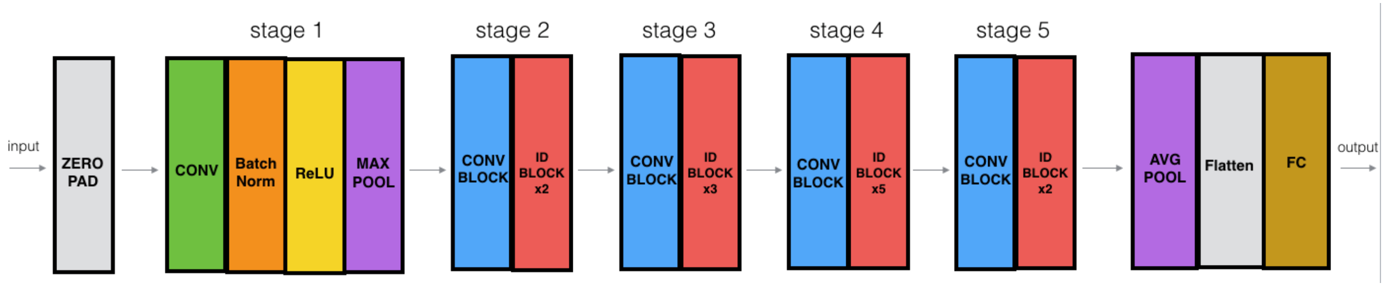
Importantly, dimension of Input and Output must be same

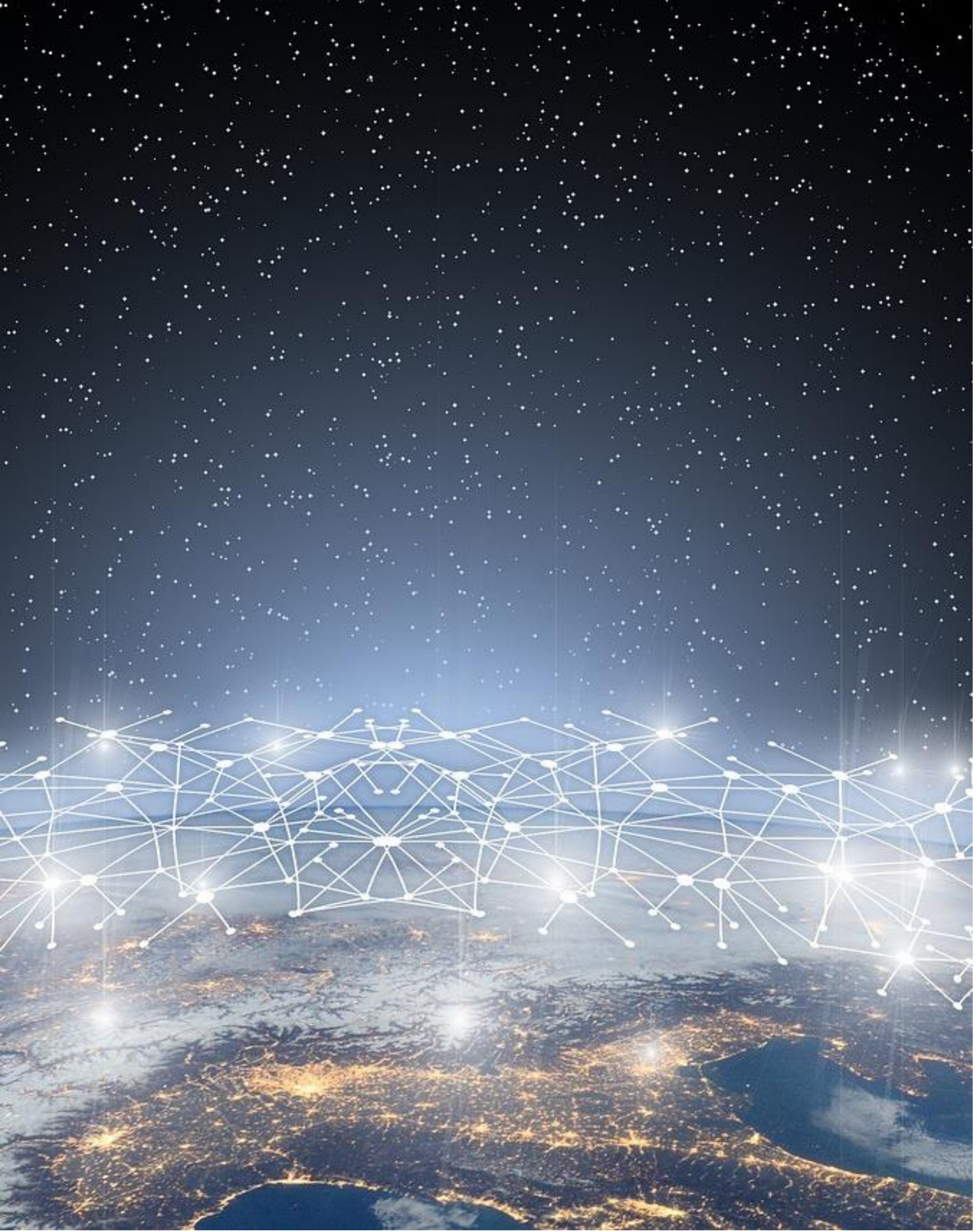


Architecture





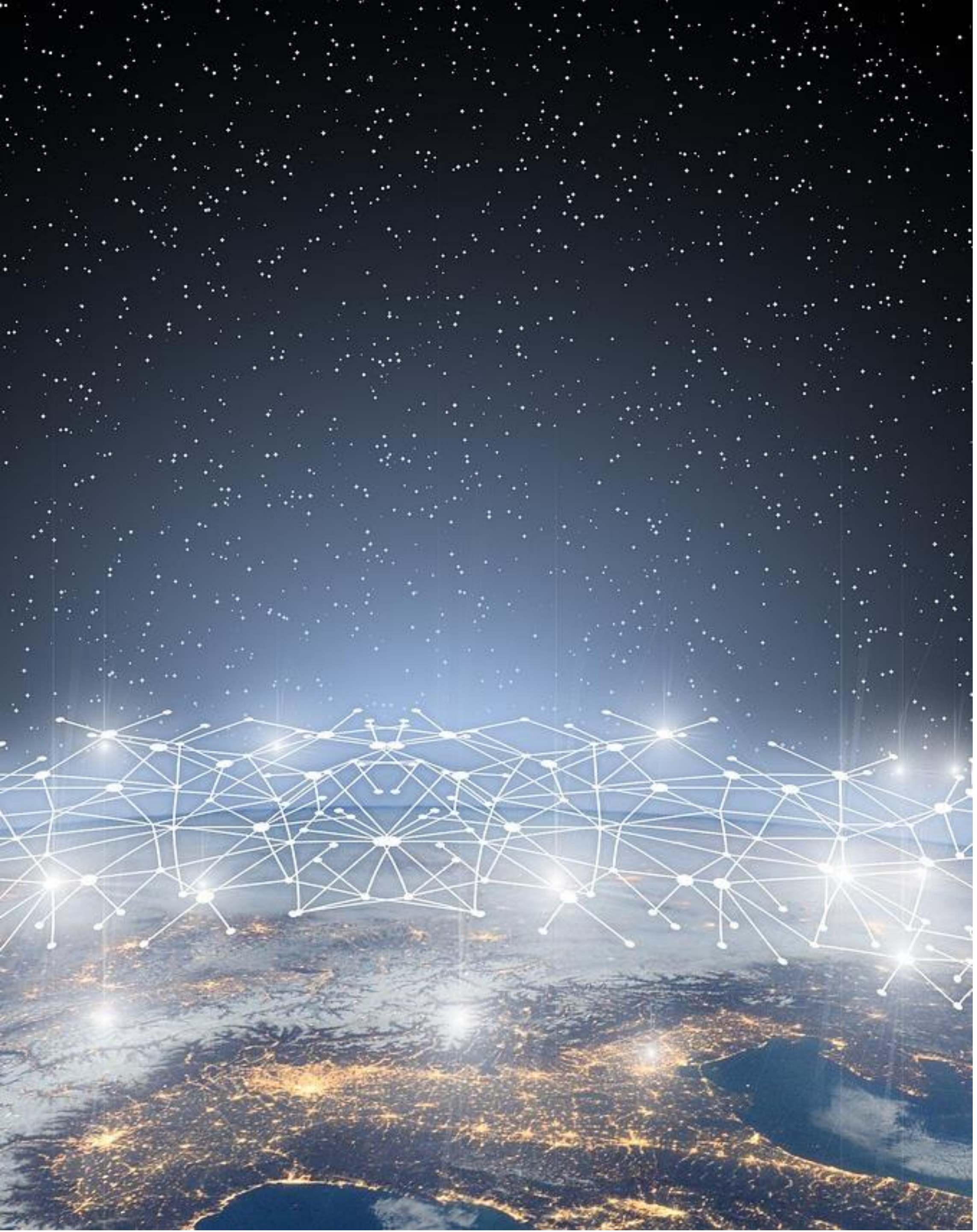




ARCHITECTURE

Plain vs Residual

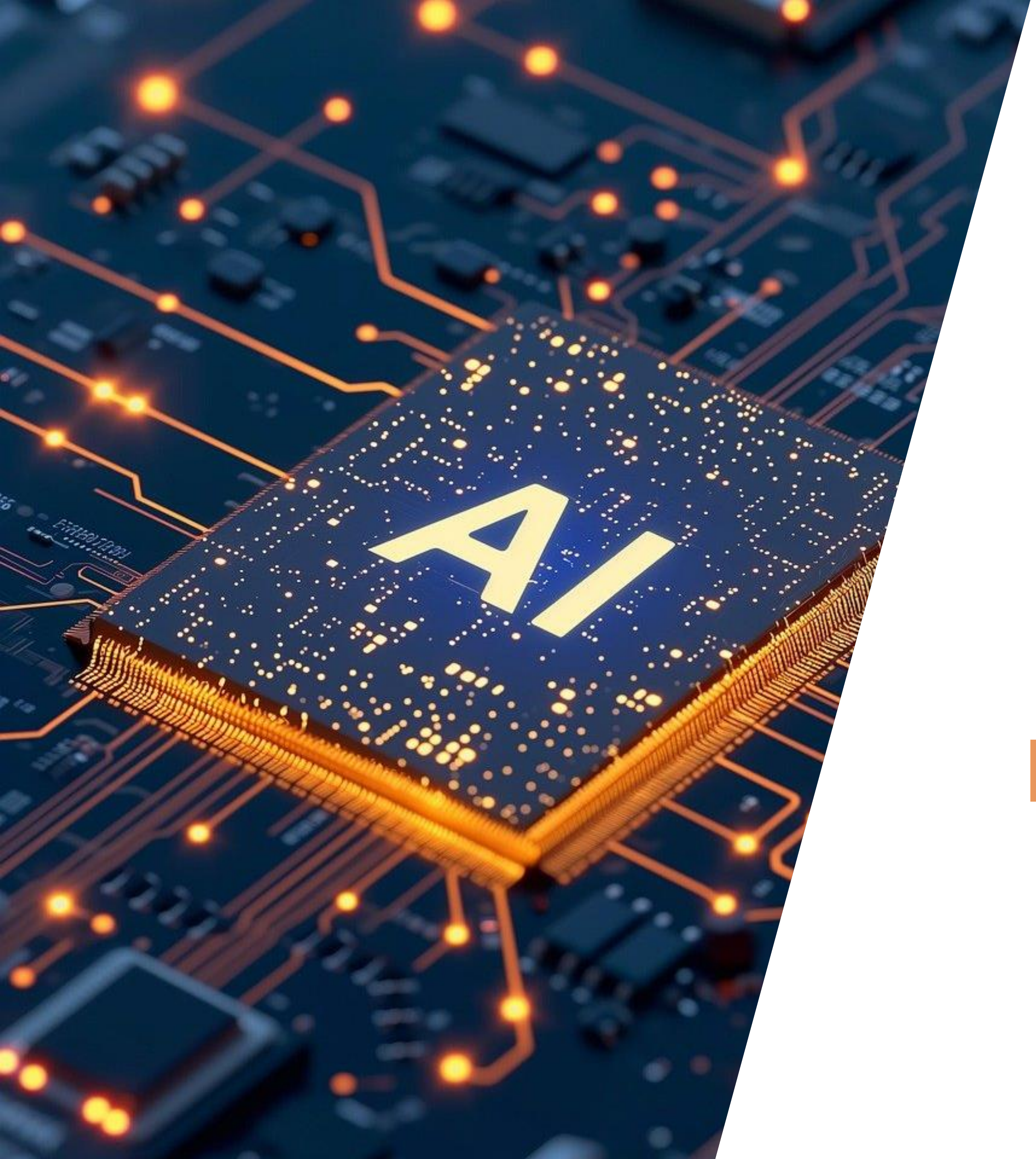
Plain	Residual
- Inspired by philosophy of VGG Net - **Convolutional Layer** : 3x3 filters & strides = 2 [design rules] • Same filter numbers for same feature map size • Feature map size x ½ = #filter x 2 - Global Average pooling layer - 1000-way fc layer with softmax - Total layer : 34 → 18% computation of VGG Net	- Plain Net + **shortcut connection** → dim(Input) == dim(Output) (A) Use Zero-padding to make dimension equal (B) Use linear projection that is used in $y = F(x, \{W_i\}) + W_s x$ → make x(input) and F(output) have equal dimension



ARCHITECTURE

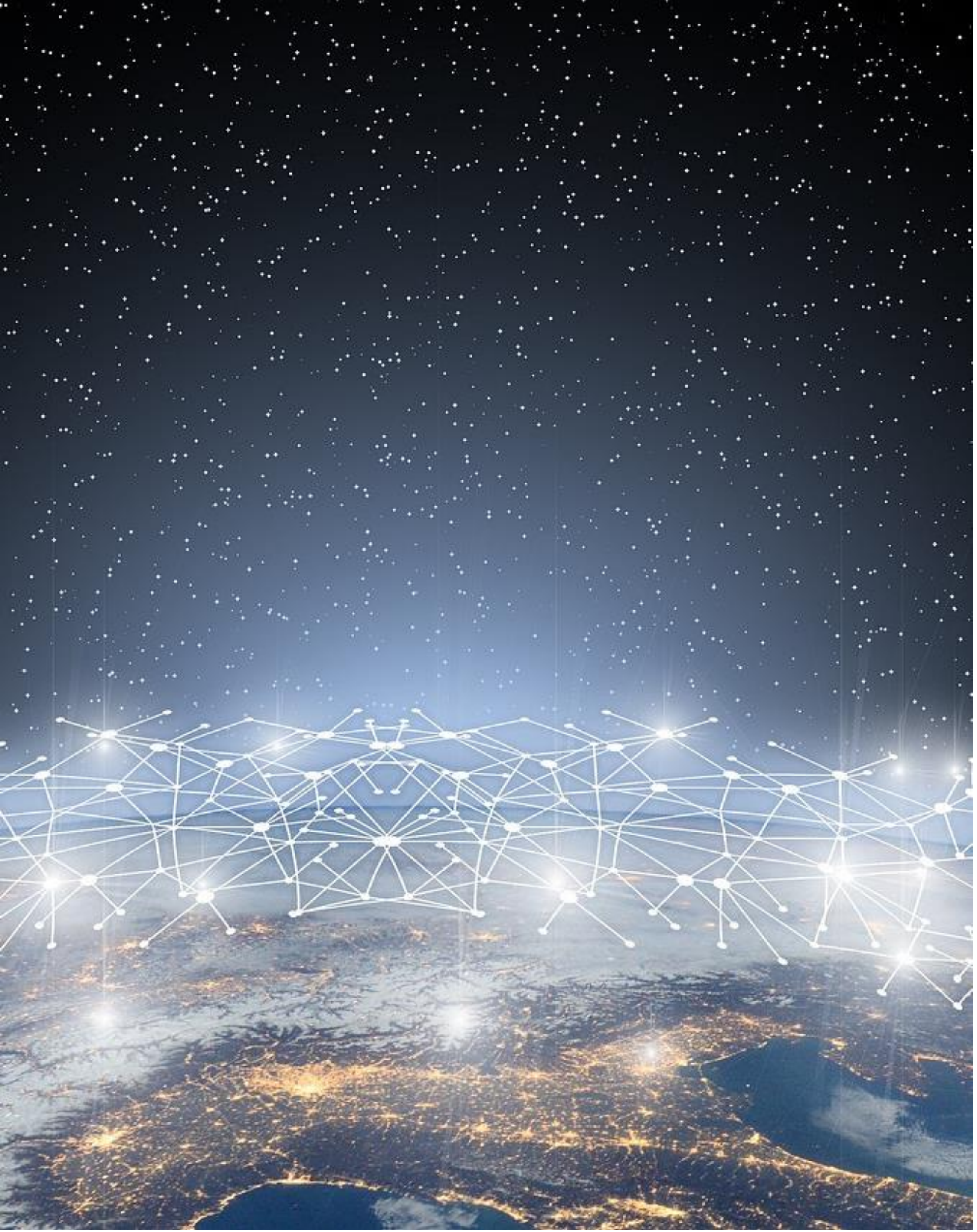
Implementation

Features
1. Image resized 224*224
2. Batch Normalization
3. Initialize Weights
4. SGD
5. 256 mini batches
6. Learning rate = 0.1
7. Iteration = 60*10^4
8. Weight decay = 0.0001
9. Momentum = 0.9
10. No dropout



EXPERIMENT

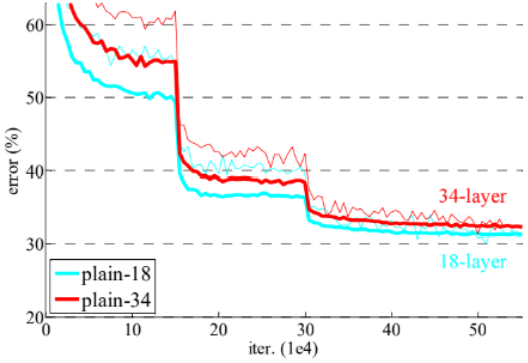
CHAPTER 4

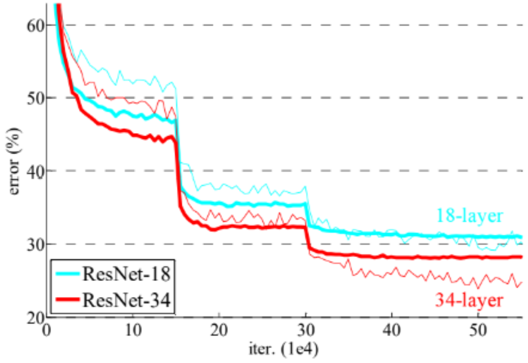


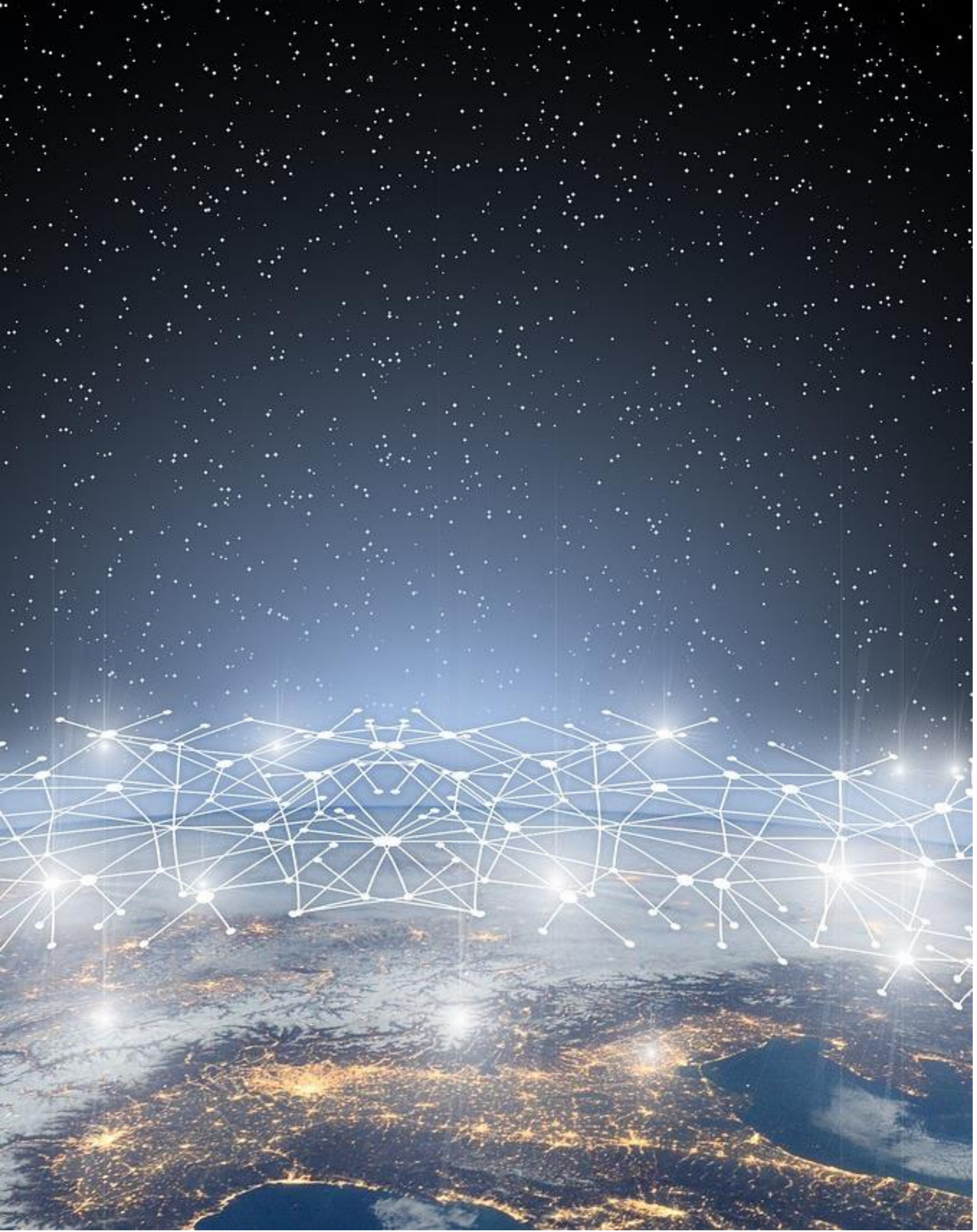
EXPERIMENT

ImageNet Classification

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9







EXPERIMENT

ImageNet Classification

	plain	ResNet
18 layers	27.94	27.88
34 layers	28.54	**25.03**

<Residual Network>

1. Performance : 18-Layer < 34-Layer

→ Degradation problem was effectively managed despite the increased depth of layers

2. Training error ↓

3. Similar accuracy, but 18-layer's convergence speed is faster

→ Optimization with SGD is easier in ResNet

EXPERIMENT

ImageNet Classification

model	top-1 err.	top-5 err.
VGG-16 [41]	28.07	9.33
GoogLeNet [44]	-	9.15
PReLU-net [13]	24.27	7.38
plain-34	28.54	10.02
ResNet-34 A	25.03	7.76
ResNet-34 B	24.52	7.46
ResNet-34 C	24.19	7.40
ResNet-50	22.85	6.71
ResNet-101	21.75	6.05
ResNet-152	**21.43**	**5.71**

<Identity vs Projection Shortcuts>

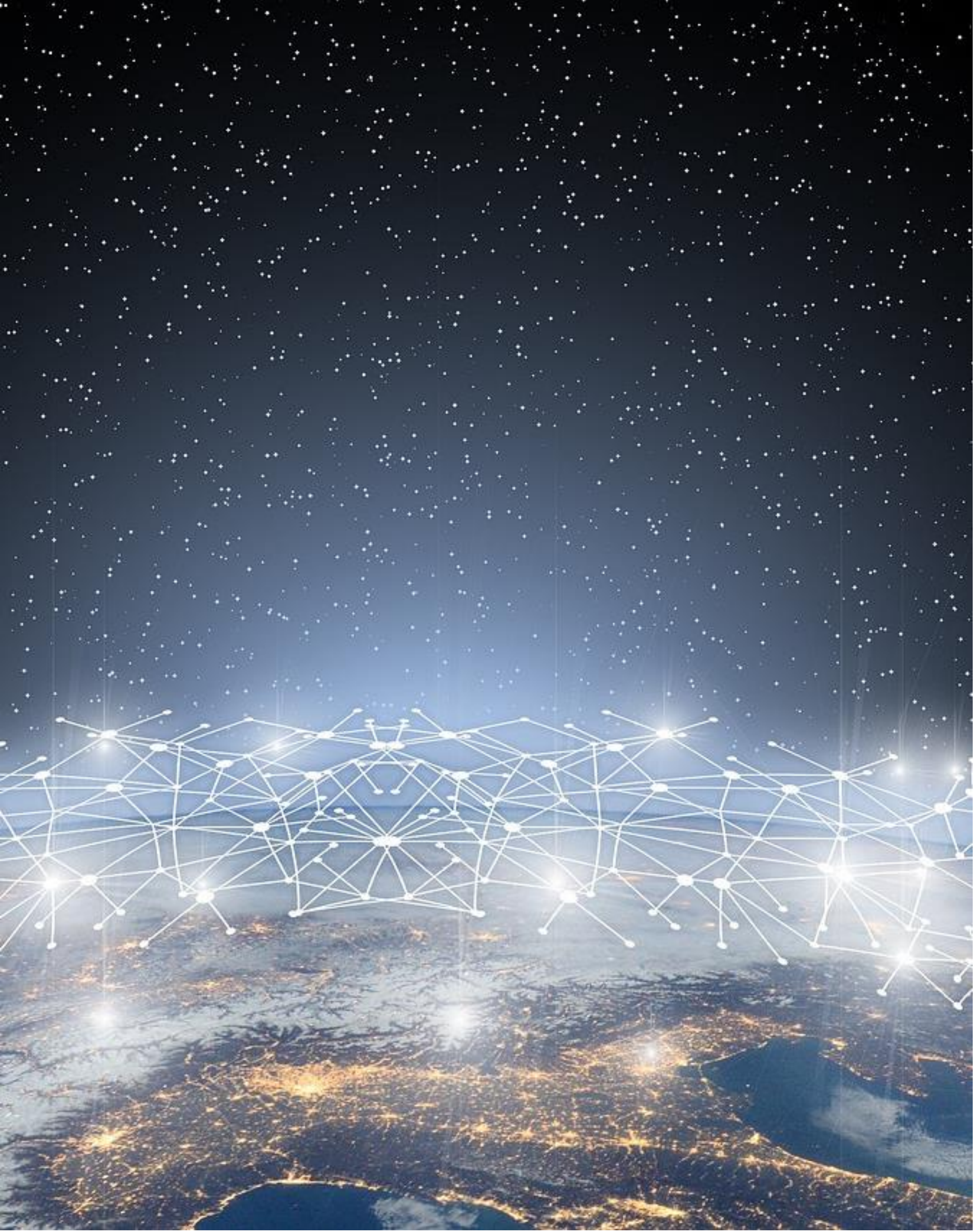
Compare three options

(A) zero-padding shortcuts (parameter free)

(B) projection shortcuts (others are identity)

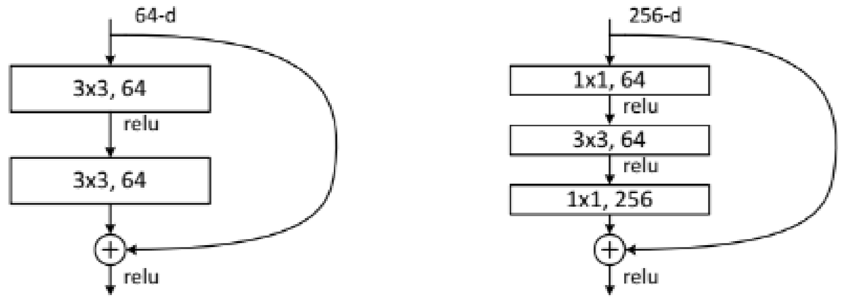
(C) All shortcuts are projections

→ degradation problem // projection shortcut



EXPERIMENT

ImageNet Classification

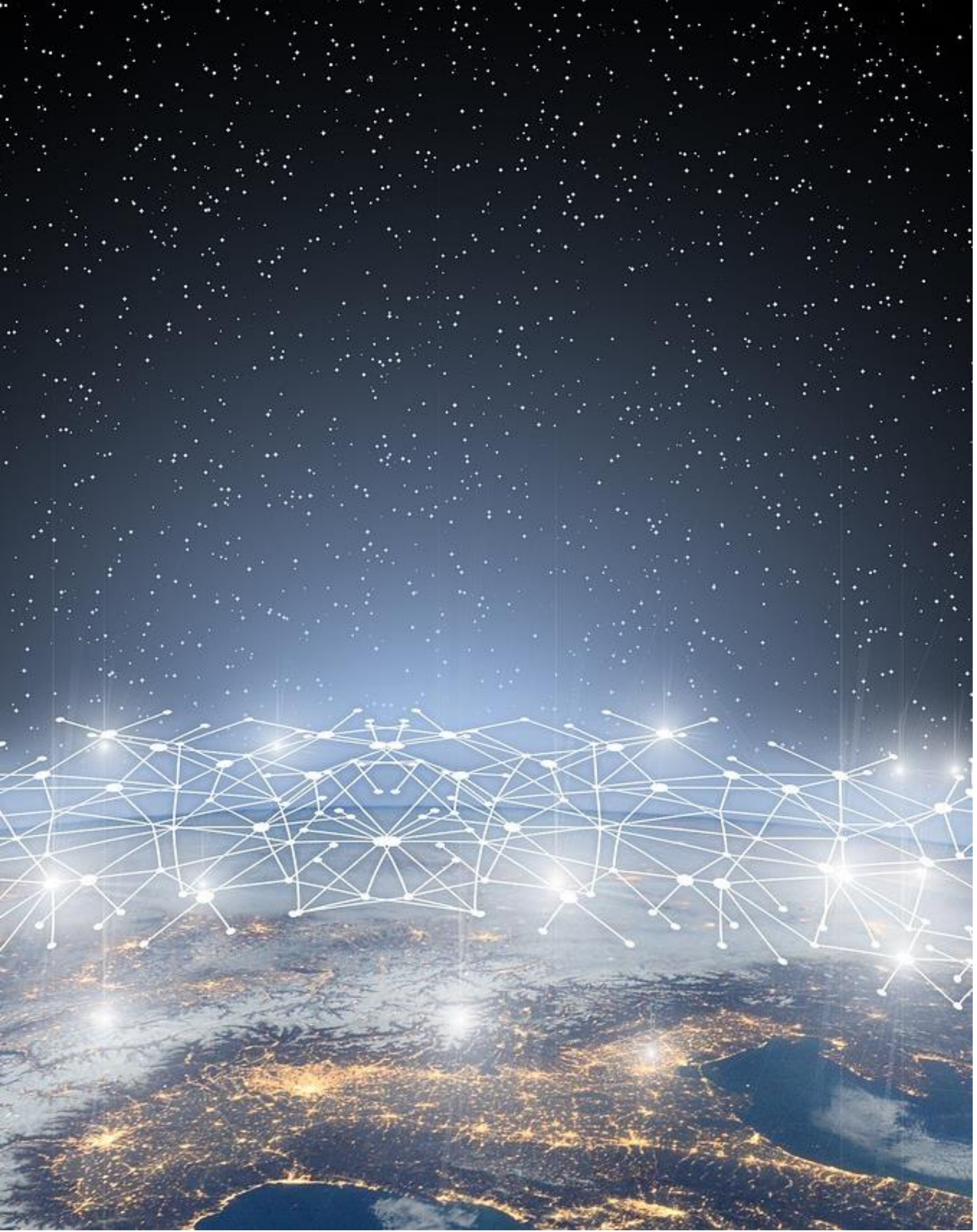


<Deeper Bottleneck Architecture>

For reducing and then increasing (restoring) dimensions

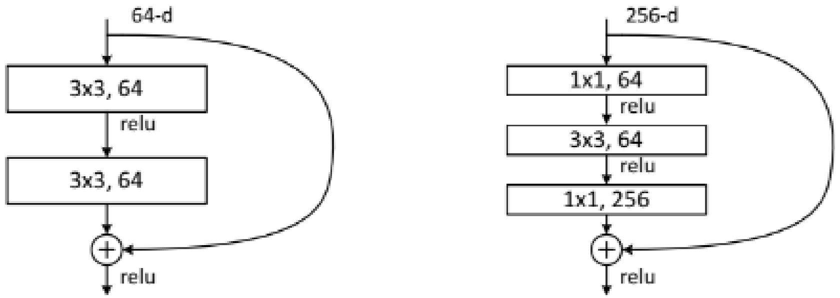
(1 x 1) → (3 x 3) → (1 x 1)

1. 50-layer
2. 101-layer and 152-layer ResNet
3. Comparisons with state-of-art Methods



EXPERIMENT

ImageNet Classification



<Deeper Bottleneck Architecture>

Reducing training time

ResNet : 3x3x64 + 3x3x64 = 1152

Bottleneck : 1x1x64 + 3x3x64 + 1x1x256 = 896

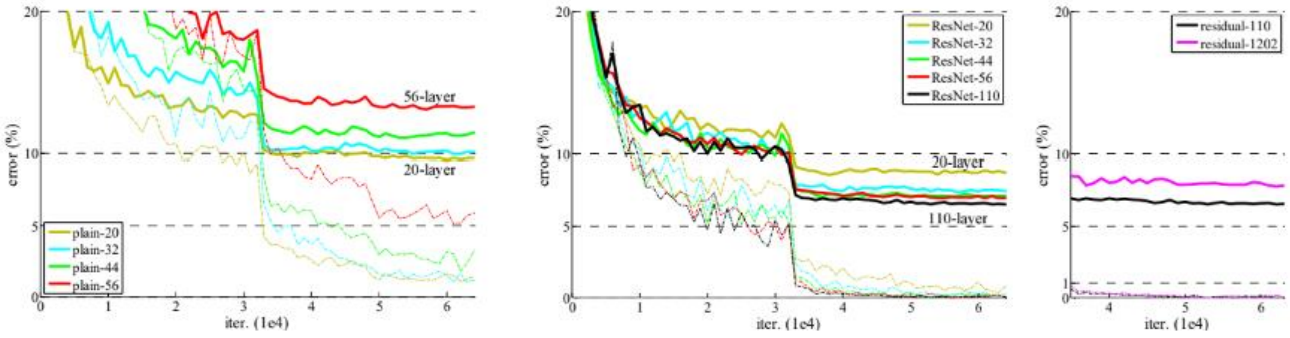
#Parameter reduced despite increase of #layer

EXPERIMENT

CIFAR-10 and Analysis

output map size	32×32	16×16	8×8
# layers	$1+2n$	$2n$	$2n$
# filters	16	32	64

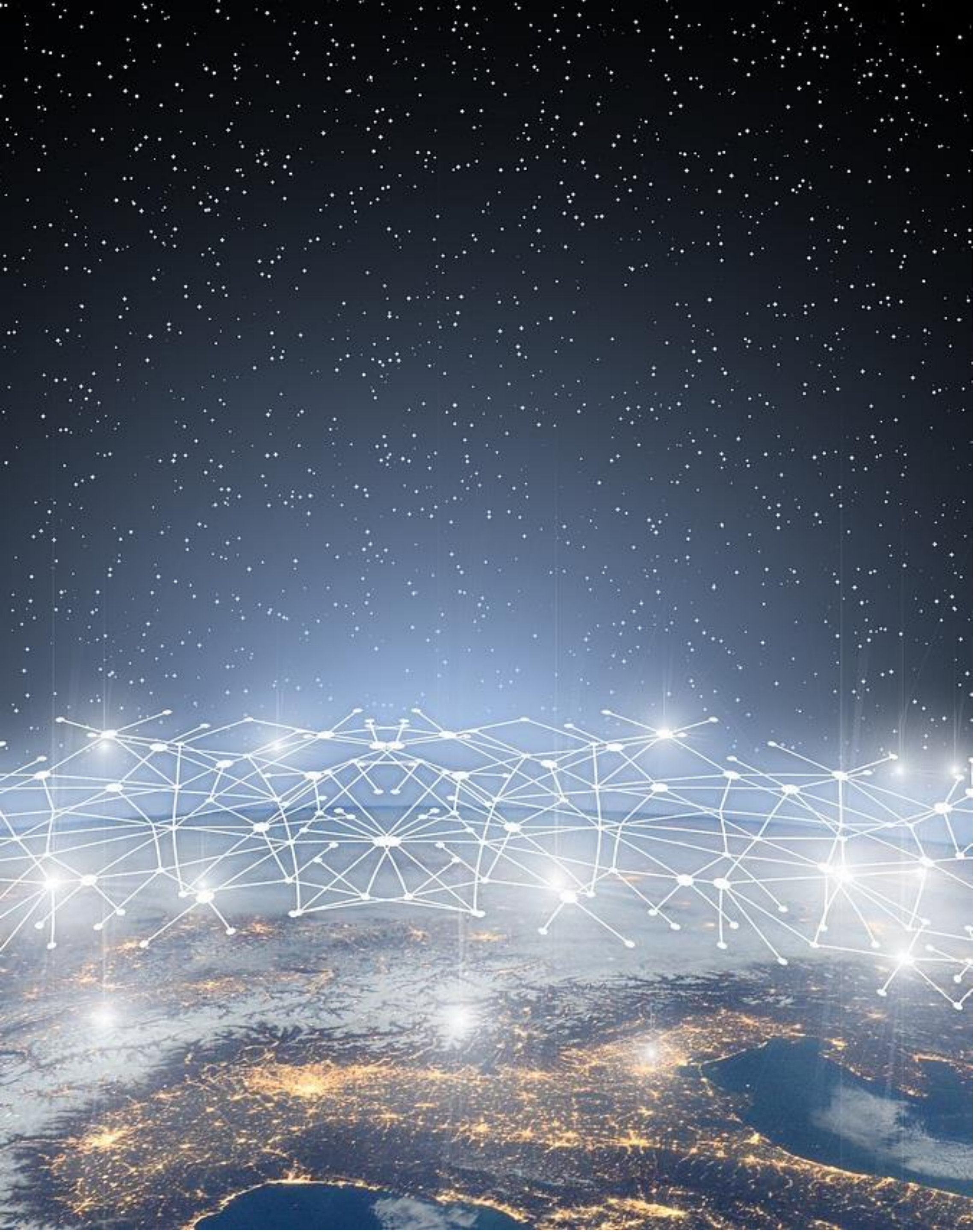
method			error (%)
Maxout [10]			9.38
NIN [25]			8.81
DSN [24]			8.22
	# layers	# params	
FitNet [35]	19	2.5M	8.39
Highway [42, 43]	19	2.3M	7.54 (7.72±0.16)
Highway [42, 43]	32	1.25M	8.80
ResNet	20	0.27M	8.75
ResNet	32	0.46M	7.51
ResNet	44	0.66M	7.17
ResNet	56	0.85M	6.97
ResNet	110	1.7M	**6.43** (6.61±0.16)
ResNet	1202	19.4M	7.93



#layers and performances are proportional

But, when trying more than 1000 layers, its error rate is worse than before because of **overfitting**.

So, it has to apply regularization methods.
(ex) **maxout**, **dropout**



EXPERIMENT

Object Detection on PASCAL and MS COCO

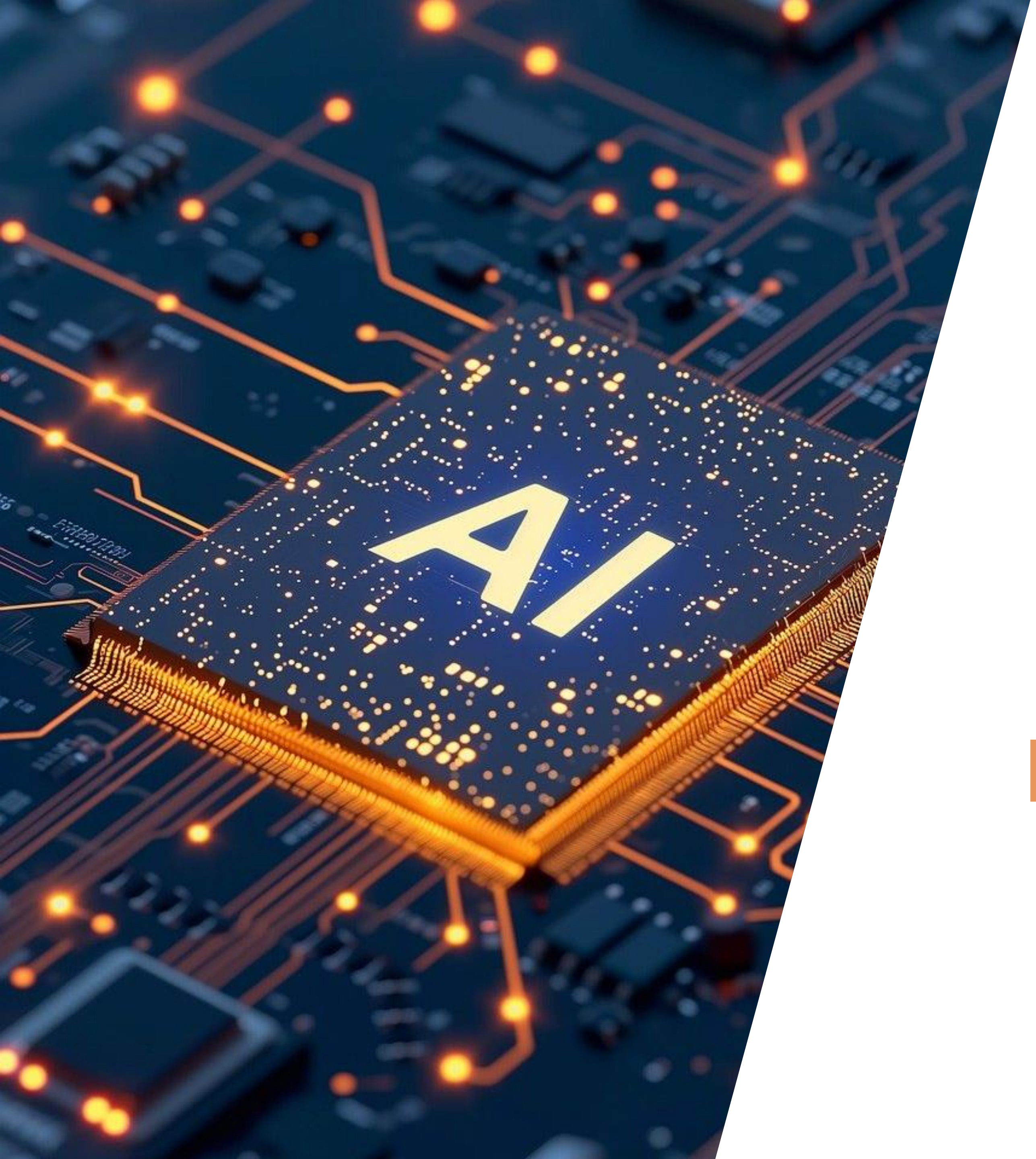
training data	07+12	07++12
test data	VOC 07 test	VOC 12 test
VGG-16	73.2	70.4
ResNet-101	**76.4**	**73.8**

Table 7. Object detection mAP (%) on the PASCAL VOC 2007/2012 test sets using **baseline** Faster R-CNN. See also Table 10 and 11 for better results.

metric	mAP@.5	mAP@[.5, .95]
VGG-16	41.5	21.2
ResNet-101	**48.4**	**27.2**

Table 8. Object detection mAP (%) on the COCO validation set using **baseline** Faster R-CNN. See also Table 9 for better results.

Adopt **Faster R-CNN** as detection method



CONCLUSION

CHAPTER 5

CONCLUSION

Proof

1. **Proof optimization is easier in ResNet than Plain Net**

   → degradation problem decreased

2. **Accuracy of ResNet is proportional to depth of network**

   → Training 1000+ layers

CONCLUSION

Result

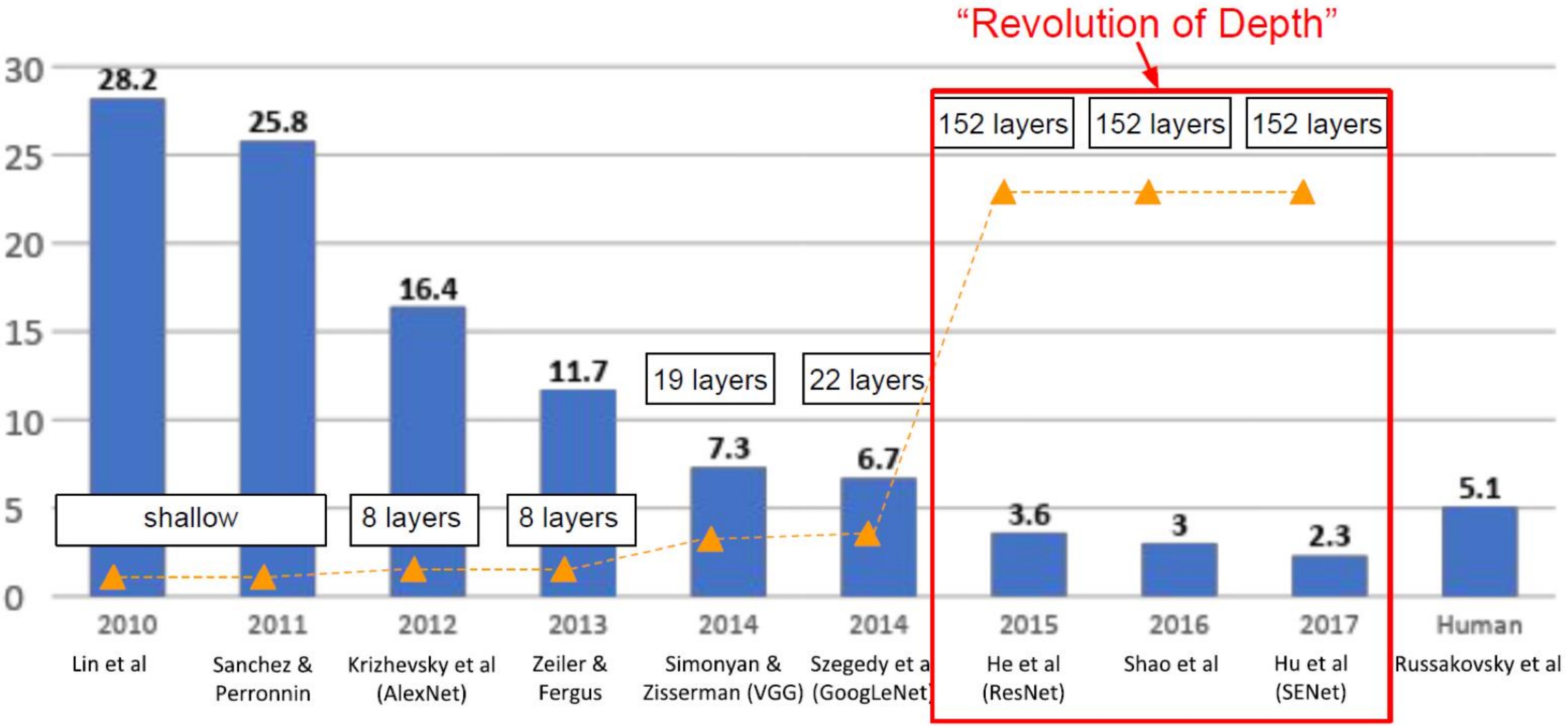
Degradation occurs when layers go beyond optimal depth

→ Create incredibly deep network

→ Send the value from optimal depth straight to the output

Skip Connection

ImageNet Large Scale Visual Recognition Challenge (ILSVRC) winners



THANK YOU

The End