Lab 1: Deep Residual Learning

Lab Objective:

In this lab, you will be asked to build the state-of-the-art convolutional neural network architecture: *Residual Network (ResNet)* [1] and train it on the Cifar-10 dataset. Moreover, you need to use data augmentation during training.

Important Date:

1. Experiment Report Submission Deadline: 8/8 (Wen) 12:00
2. Demo date: 8/8 (Wen) 17:30

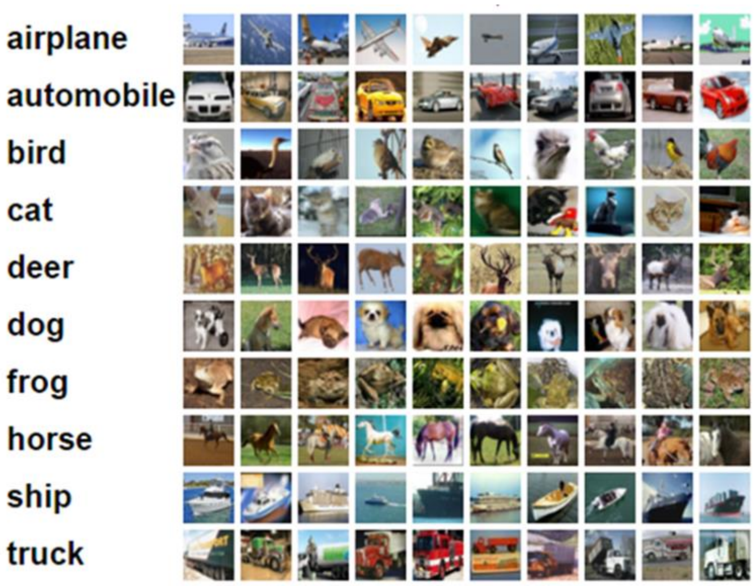
Requirements:

- Implement ResNet-20/56/110 for Cifar-10 [1]
- Train ResNet with data augmentation
- Compare to vanilla CNNs with same depth 20/56/110

Environment:

- Cifar-10 dataset

The CIFAR-10 dataset consists of 60000 32×32 color images (RGB) in **10** classes, with 6000 images per class. There are 50000 training images and 10000 test images.



Sample Code:

There are many cifar-10 sample codes for pytorch:

https://github.com/kuangliu/pytorch-cifar

Lab Description:

- **Deep residual learning**

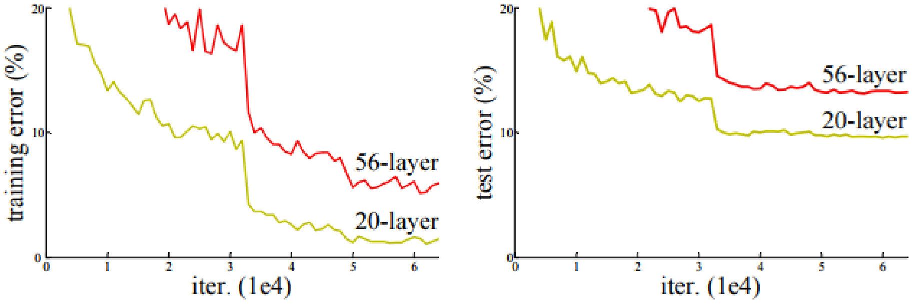


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer "plain" networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

  - Degradation problem: the network depth increasing, accuracy gets saturated (which might be unsurprising) and then degrades rapidly
  - Not overfitting, it's the vanishing gradient problem
  - Add shortcut connection! $F(x)$ now is fitting residuals!

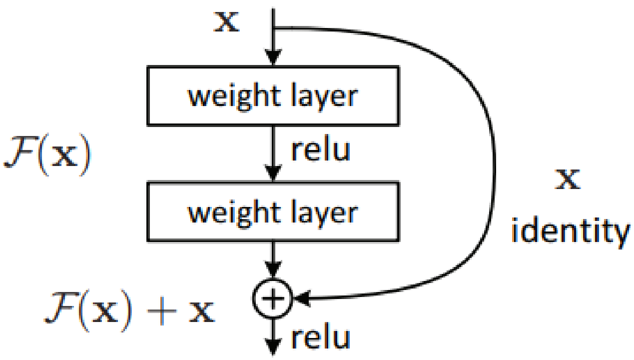


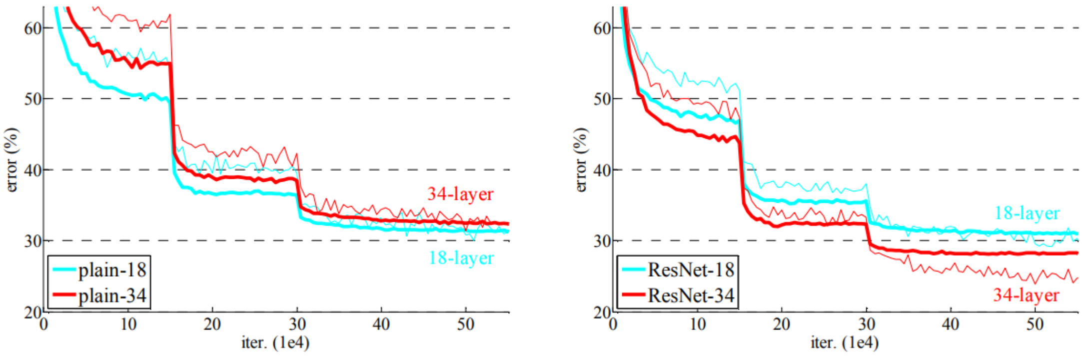
Figure 2. Residual learning: a building block.

  - Why ResNet can avoid vanishing gradient problem??

$$\mathbf{x}_L = \mathbf{x}_l + \sum_{i=l}^{L-1} \mathcal{F}(\mathbf{x}_i, \mathcal{W}_i),$$

$$\frac{\partial \mathcal{E}}{\partial \mathbf{x}_l} = \frac{\partial \mathcal{E}}{\partial \mathbf{x}_L} \frac{\partial \mathbf{x}_L}{\partial \mathbf{x}_l} = \frac{\partial \mathcal{E}}{\partial \mathbf{x}_L} \left(1 + \frac{\partial}{\partial \mathbf{x}_l} \sum_{i=l}^{L-1} \mathcal{F}(\mathbf{x}_i, \mathcal{W}_i)\right).$$

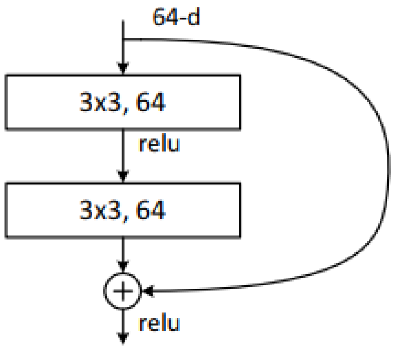
■ Learning better networks as easy as stacking more layer



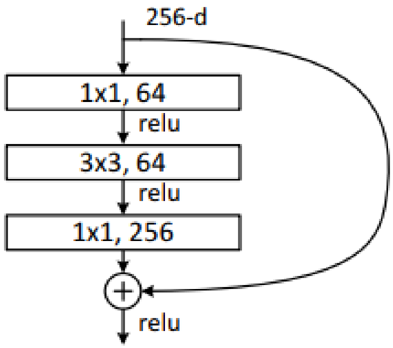
- Build Residual Block

  Example: a residual block with 64 feature maps

  Basic block



  Bottleneck block



- Network Architecture for Cifar-10

  ※For Basic block

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

Total Depth = 1 (conv) + 6n + 1(linear layer)

For example, to build ResNet-110, we need n= 18 ((110-2)/6).

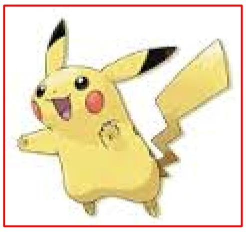
Note that there is global average pooling before linear layer.

- Data preprocessing:
  Color normalization
  Normalize each color channel (compute from entire CIFAR10 training set)

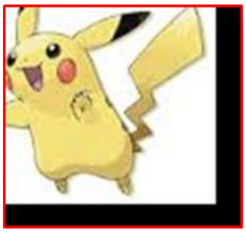
Mean $$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{pmatrix} 0.4914 \\ 0.4824 \\ 0.4467 \end{pmatrix}$$

Standard deviation $$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{pmatrix} 0.2471 \\ 0.2435 \\ 0.2616 \end{pmatrix}$$

- Data augmentation: Translation and Horizontal flipping:



Original



Translation

Horizontal flipping

Implementation Details:

■ Training Hyperparameters:
- Method: SGD with momentum
- Mini-batch size: 128 (391 iterations for each epoch)
- Total epochs: 164, momentum 0.9
- Initial learning rate: 0.1, divide by 10 at 81, 122 epoch
- Weight decay = 0.0001
- Weight initialization: torch.nn.init.kaiming_normal
- Loss function: cross-entropy

■ Data augmentation parameters:
- ■ Translation: Pad **4** zeros in each side and random cropping back to 32x32 size
- ■ Horizontal flipping: With probability **0.5**

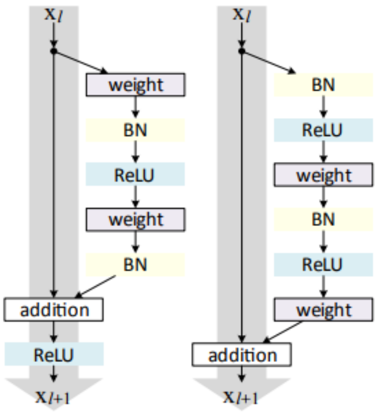
Methodology:

- ResNet-20 got 92.37% accuracy, Time: 0.58 hr
- ResNet-56 got 93.53% accuracy, Time: 1.48 hr
- ResNet-110 got 93.95% accuracy, Time: 2.87 hr

✓ On single Titan X (Maxwell)

Extra Bonus (+2):

- Identity Mapping in deep residual networks [2].
    - Pre-activation Residual Network (pre-act ResNet)



(a) original (b) proposed

method	# layers	# params	error (%)
Maxout [10]			9.38
NIN [25]			8.81
DSN [24]			8.22
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

-
    - Try pre-act ResNet-20/56/110

References:

[1] He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

[2] He, Kaiming, et al. "Identity mappings in deep residual networks." *European Conference on Computer Vision*. Springer International Publishing, 2016.

Report Spec: [black: Demo, Gray: No Demo]

1. Introduction (15%)
2. Experiment setup (5%, 10%)
   - The detail of your model
   - Report all your training hyper-parameters
3. Result
   - The comparison between ResNet and vanilla CNNs
     - Final Test error (5%, 15%)
     - Training loss curve (you need to record training loss every epoch) (10%, 20%)
     - Test error curve (you need to record test error every epoch) (10%, 20%)
4. Discussion (10%, 25%)

Demo (50%)

------- Criterion of result (ResNet-110)----

Accuracy > 93% = 100%

Accuracy: (93.0~90.0)% = 90%

Accuracy: (90.0~87.0)% = 80%

Accuracy < 87.0% = 70%

Accuracy: 10% = 0%

評分標準: 40%*實驗結果 + 60%*(報告+DEMO) + bonus