

# 레드빈즈 딥러닝 스터디

## 모두를 위한 얇고 넓은 딥러닝

---

with



오 준 오

Machine Learning & Pattern Analysis laboratory

25<sup>th</sup> September 2019

# Introduction



오준오 / Demian Oh

MLPA 연구실 제 2 공학관 503호, 학부연구생

Lecture Notes and Example Codes

<https://github.com/DemieOh/RedBeansDeepLearning>

# Introduction

정말 **쉬운**, 초보자를 위한 강의

**적은**, 그러나 **필수**적인 이론 & 다양한 **예제**를 통해 배우는 사용법

강의는 대략 1시간 ~ 1시간 30분 가량 진행

앞부분은 이론 소개, 뒷부분은 예제를 따라가며 실습으로 진행

주로 컴퓨터 비전과 관련된 예제 사용

학습 목표는 다음과 같습니다.

1. 딥러닝에 대한 개념 잡기
2. 자신만의 데이터와 모델로 실험하기

# Syllabus

## 첫 세 스터디

### 첫 번째 스터디

딥러닝이란 무엇인가 / 설치부터 데이터 다루기

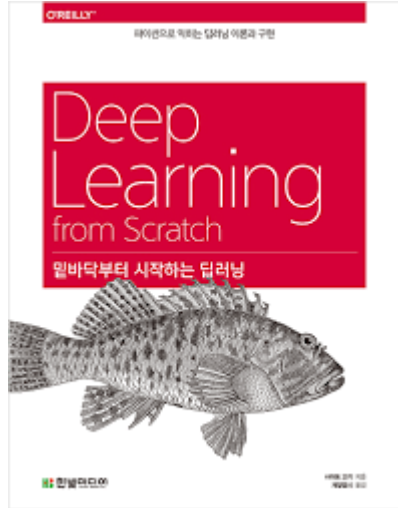
### 두 번째 스터디

왜 딥러닝이어야 하는가 / 모델과 학습

### 세 번째 스터디

컴퓨터의 눈으로 보기 / 이미지 처리하기

# 시작하기 앞서



밑바닥부터 시작하는 딥러닝

사이토 고키, 2017

Only Python

## 추천 강의

Stanford University

CS231n: Convolutional Neural Networks  
for Visual Recognition

<http://cs231n.stanford.edu/syllabus.html>

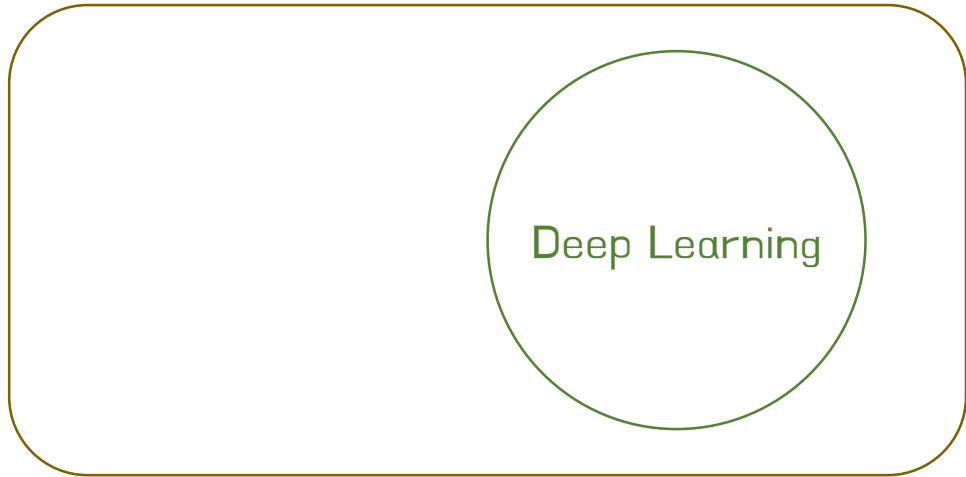
Professor Hun Kim

모두를 위한 머신러닝/딥러닝 강의

<https://hunkim.github.io/ml/>

# 딥러닝, 시작

Machine Learning



머신 러닝 / 기계 학습 안에 포함

기본적인 개념은 1969년에 이미 정립  
“Perceptron” by Marvin Minsky, MIT AI Lab

심층신경망을 모델로 사용하는 머신러닝

딥러닝 역시 머신러닝의 장점과 한계 공유

## 딤러닝, 시작



사람은 쉽게 이 사진이 고양이임을 안다.

그러나 컴퓨터에게 물어보면?

# 딥러닝, 시작

## 전문가 시스템의 등장

인간이 특정 분야에 대하여 가지고 있는 전문적인 지식을 컴퓨터에 저장

이를 통해 일반인도 이 전문 지식을 사용 가능



털이 있고 귀가 뾰족하게 위로 솟아 있고 얼굴이 역삼각형인 걸 보니

이 사진은 “고양이”를 뜻하는 구나!

그런데 만약 귀가 접힌 고양이 사진이었다면?



# 딥러닝, 시작

이러한 전문 지식을 찾는 과정을 컴퓨터에게 시킬 순 없을까?

정말 정말 많은 데이터

BIG DATA



의미 있는 지식

# 딥러닝, 무엇인가요?

## :: Definition of Deep Learning ::

여러 비선형 변환 기법의 조합을 통해 높은 수준의

추상화(다량의 데이터나 복잡한 자료들 속에서 핵심적인 내용 또는 기능을 요약하는 작업)를

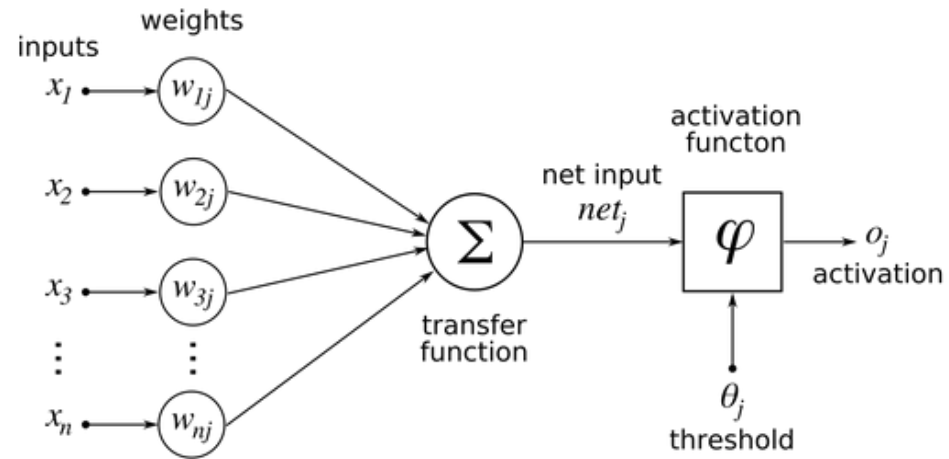
시도하는 기계학습 알고리즘의 집합

데이터 속의 정형화된 패턴을 찾고 문제를 해결하기 위해 이를 활용하는 것

# 딥러닝, 무엇인가요?

## 퍼셉트론 / Perceptron

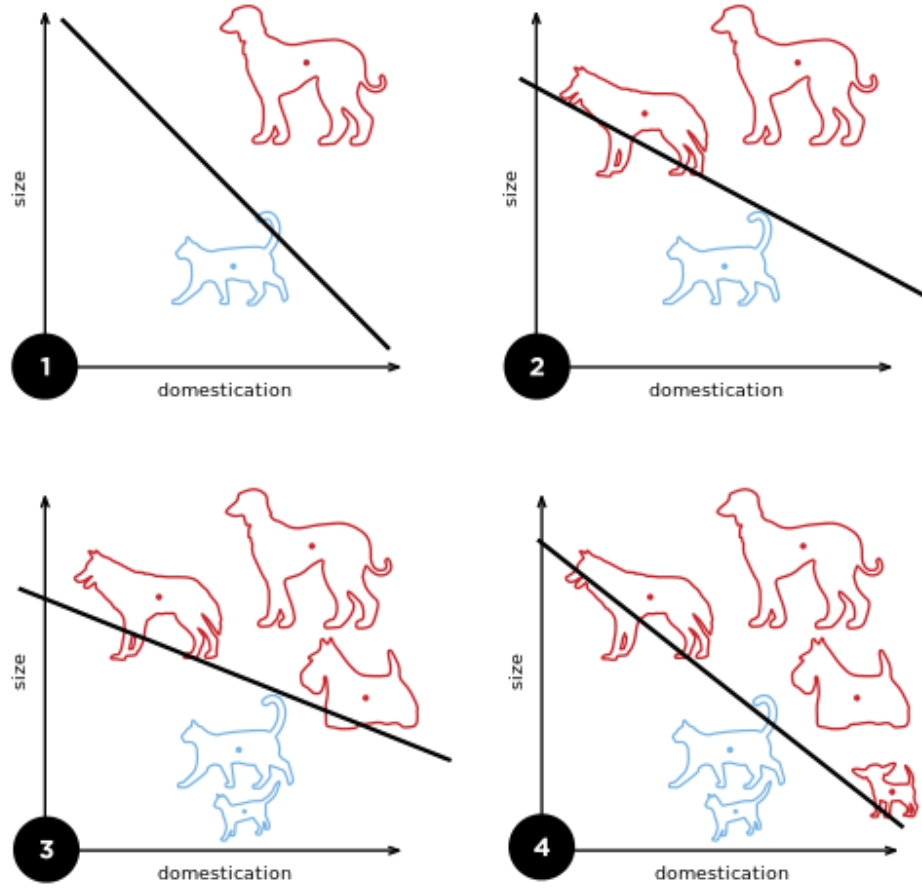
Frank Rosenblatt, 1958



$$f(x) = w_1x_1 + w_2x_2 + \dots + w_nx_n + b$$

$$\text{if } f(x) \leq \theta, \text{ then } o_j = 0; \text{ else } o_j = 1$$

# 딥러닝, 무엇인가요?



좌표 위에 점(데이터)을 구분하기 위해

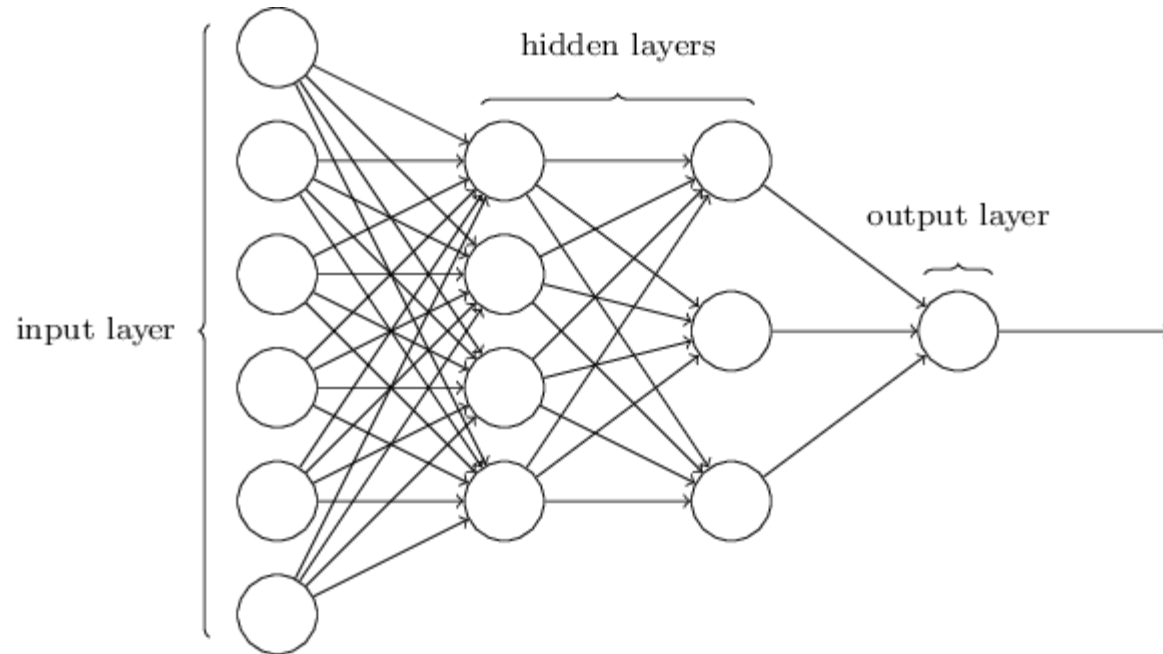
모델은 직선을 그림

점이 추가될 때 마다 이를 반영하여 직선을 수정

이 과정이 사람의 학습 과정과 유사하여 학습이라 함

딥러닝, 무엇인가요?

## MLP (Multi Layer Perceptron)



퍼셉트론의 한계를 극복하기 위해 등장

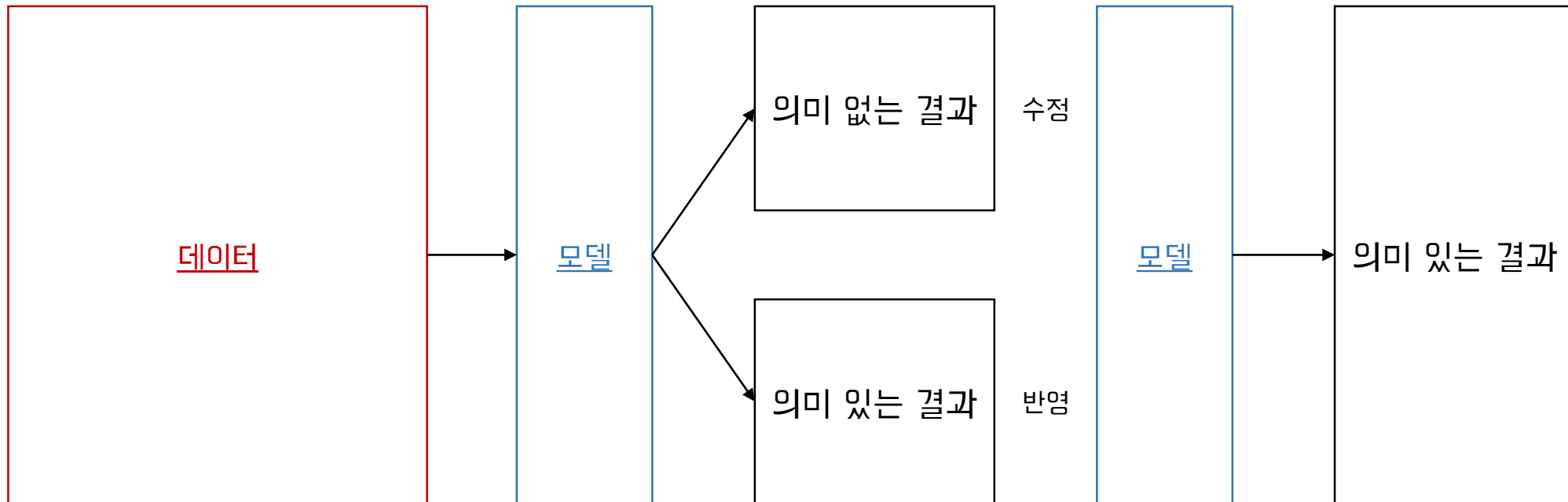
현재 사용되는 모든 인공지능망 모델의 기반이 됨

# 딥러닝, 무엇인가요?

## 딥러닝의 필수 요소

데이터, 모델, 학습

만족할 때 까지 반복



# 오늘의 목표

## 딥러닝의 3요소 중 첫번째, 데이터 다루기

1. 개발환경을 구성합니다.
2. PyTorch를 설치합니다.
3. 내장된 모듈로 데이터를 불러와 봅니다.
4. 데이터를 학습 가능한 형태로 만듭니다.

실습을 시작하기 전에

개발 환경 구성하기



Anaconda Python Distribution

+



JetBrains PyCharm IDE



# 실습을 시작하기 전에

## Anaconda Python Distribution

<https://www.anaconda.com>



패키지 관리와 디플로이를 단순하게 할 목적으로 만들어진 파이썬 배포판

과학 계산(데이터 과학, 기계학습, 대규모 데이터 처리, 예측 분석 등)에 특화

Windows, Linux, macOS 용 1,400개 이상의 유명 데이터 과학 패키지가 포함

Venv(Virtual Environment) 기능을 손쉽게 관리할 수 있음

Virtual Environment 기능은 JVM(Java Virtual Machine)과 유사

# 실습을 시작하기 전에

## JetBrains PyCharm IDE

<https://www.jetbrains.com/pycharm/>

JetBrains에서 제작한 Python 용 통합개발환경(IDE)

IntelliJ IDEA에 기반을 두고 있으며 현용 Python 개발 툴 중에 가장 완성도 높은 제품

코드 자동 완성 기능과 디버깅 기능을 지원함 - 매우 중요!

VirtualEnv는 네이티브로, Anaconda는 전용 plug-in으로 적극 지원



# 실습을 시작하기 전에

PyTorch

<https://pytorch.org>



Python 기반의 오픈 소스 딥러닝 라이브러리, 페이스북 AI 연구팀에 의해 개발

간결하고 빠른 구현, Tensorflow에 비해 훨씬 익히기 쉬움

Tensorflow에 비해 높은 추상화 레벨을 가지고 있기 때문

패러다임 또한 직관적으로 이해하기 쉬운 Define-by-Run 구조

기능들이 잘 모듈화되어 있고 이름 또한 직관적이라 찾고 이해하기 쉬움

# 딥러닝, 첫걸음

## CIFAR 10

<https://www.cs.toronto.edu/~kriz/cifar.html>

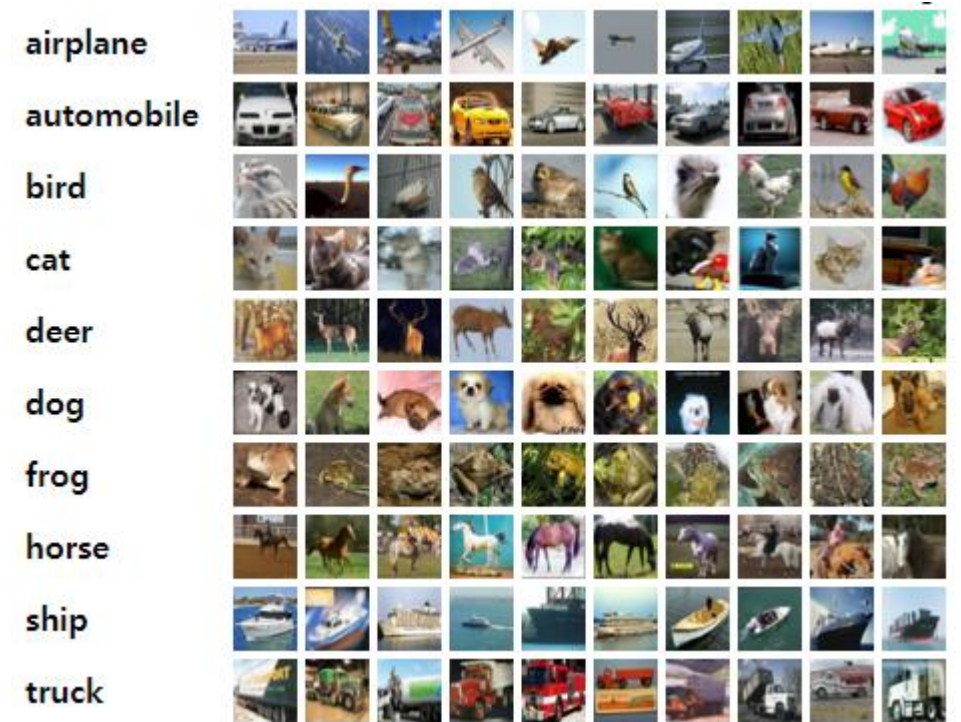
32 x 32 (pixel size) 크기의 이미지 60,000장

10개의 클래스로 분류되며 각 클래스 당 6,000개의 이미지가 포함

50,000개의 훈련 데이터와 10,000개의 시험 데이터로 나뉨

데이터셋에 포함된 클래스는 다음과 같습니다.

[비행기, 차, 새, 고양이, 사슴, 개, 개구리, 말, 배, 트럭]

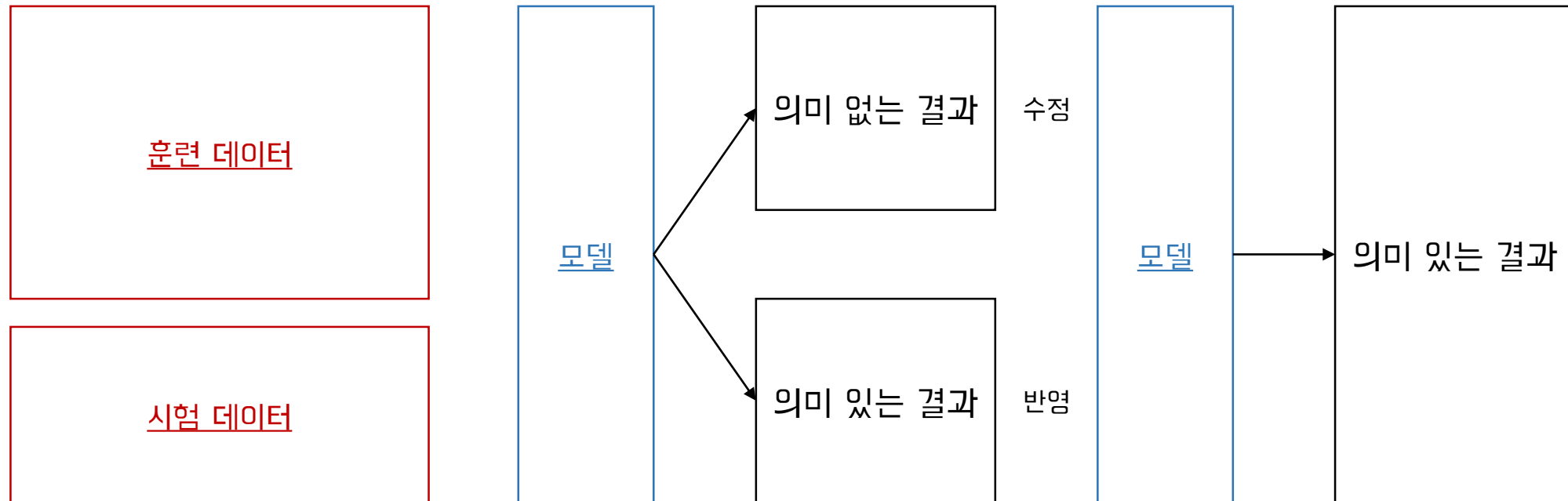


<Example of CIFAR 10 dataset images>

# 딥러닝, 첫걸음

## CIFAR 10

훈련 데이터와 시험 데이터를 나누지 않으면 모든 결과가 의미 없게 된다. 왜 일까?



# 딥러닝, 첫걸음

```
import torch
import torchvision
import torchvision.transforms as transforms

import matplotlib.pyplot as plt
import numpy as np
```

```
transform = transforms.Compose(
    [transforms.ToTensor(),
     transforms.Normalize(mean=[0.5, 0.5, 0.5],
                          std=[0.5, 0.5, 0.5])])
```

```
trainset = torchvision.datasets.CIFAR10(root='./data', train=True, download=True, transform=transform)
testset = torchvision.datasets.CIFAR10(root='./data', train=False, download=True, transform=transform)
```

# 딥러닝, 첫걸음



```
trainloader = torch.utils.data.DataLoader(trainset, batch_size=4, shuffle=True, num_workers=2)
testloader = torch.utils.data.DataLoader(testset, batch_size=4, shuffle=False, num_workers=2)

classes = ('plane', 'car', 'bird', 'cat', 'deer', 'dog', 'frog', 'horse', 'ship', 'truck')
```



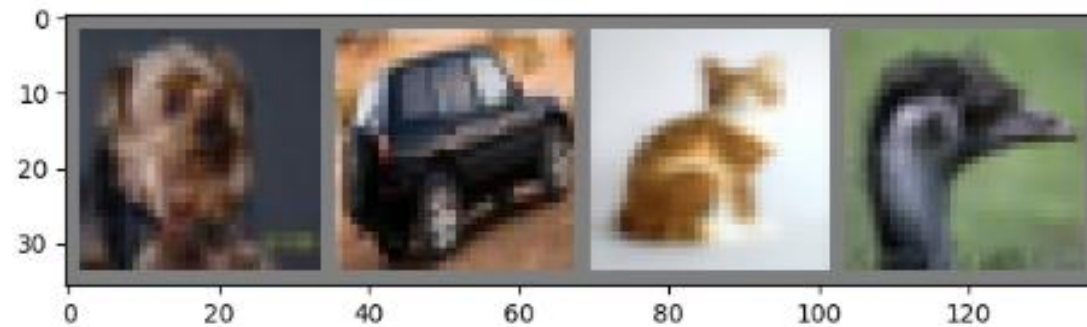
```
dataiter = iter(trainloader)
images, labels = dataiter.next()

imshow(torchvision.utils.make_grid(images))
print(' '.join('%5s' % classes[labels[j]] for j in range(4)))
```



```
def imshow(img):
    img = img / 2 + 0.5
    npimg = img.numpy()
    plt.imshow(np.transpose(npimg, (1, 2, 0)))
    plt.show()
```

## 딥러닝, 첫걸음



```
C:\Users\32162719\Anaconda3\envs\wintorch_tensorboard\python.exe C:/Users/32162719/404_Not_Found/Red_Beans/lecture1.py
Files already downloaded and verified
Files already downloaded and verified
  dog  car  cat  bird

Process finished with exit code 0
```



수고하셨습니다  
감사합니다