



Machine Learning Analysis



[illegible]

GyeongYeong Kim

- **Keras Additional Functions**

- **CIFAR10 Introduction**
- **Model Summation**
- **Data Augmentation**
- **Transfer Learning**

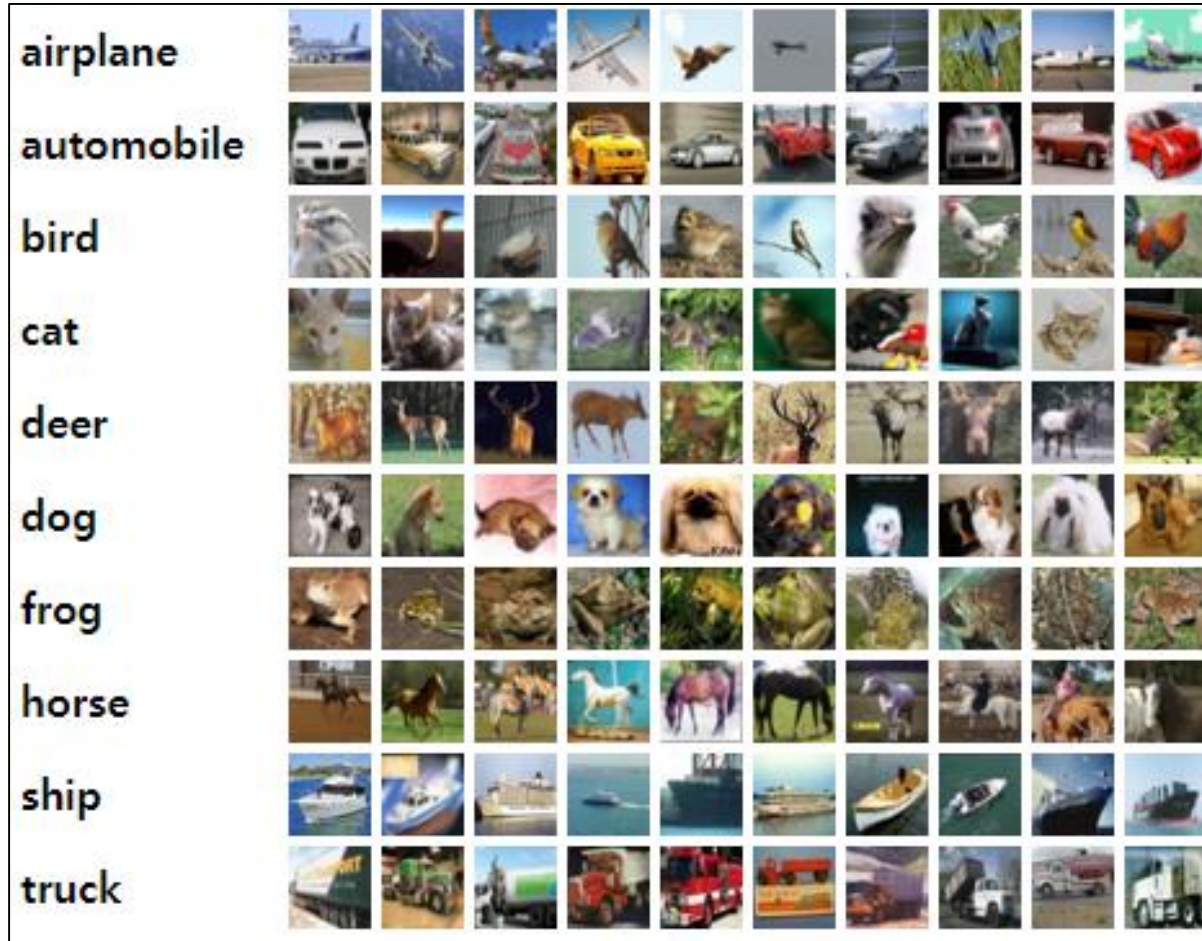
- **Pytorch Additional Functions**

- **CIFAR10 Introduction**
- **Data Augmentation**
- **Transfer Learning**
- **Training Customized Dataset**

CNN – CIFAR 10

PIAI Research Department

- 데이터셋 소개 – CIFAR 10



- CIFAR-10은 열 가지의 class로 이루어진 공인 Dataset
- 비행기, 자동차, 새, 고양이, 사슴, 개, 개구리, 말, 배, 트럭으로 구성
- 32 × 32 픽셀의 3채널 RGB 이미지
- Class당 6,000개의 이미지로 이루어져 있으며 총 60,000장
- 5만장이 Train / 1만장이 Test
- 머신러닝 / 딥러닝 챌린징 분야에서 널리 사용되는 데이터셋

Model Summation

PIAI Research Department

```
def conv_maxpool_layers(n_in):
    # Coding Time
    model = Sequential()
    model.add(Conv2D(16, kernel_size=(3, 3), padding='same', activation='relu', input_shape=(n_in)))
    model.add(Conv2D(32, kernel_size=(3, 3), padding='same', strides=(2, 2), activation='relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Flatten())
    model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    return model

def fc_layers(n_out):
    # Coding Time
    model = Sequential()
    model.add(Dense(units=128, input_shape=(2048,), activation='relu'))
    model.add(Dense(units=n_out, activation='softmax'))
    model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    return model
```



```
def CNN_sum(n_in, n_out):
    feature_extractor=conv_maxpool_layers(n_in)
    feature_extractor.trainable=True |
    ann_classifier = fc_layers(n_out)
    ann_classifier.trainable=True

    x = Input(shape=n_in)
    feature = feature_extractor(x)
    y = ann_classifier(feature)
    model = Model(inputs = x, outputs = y)
```

- trainable 특정 layer 중심으로 학습 여부를 설정할 수 있음 -> Fine Tuning 같은 역할을 할 수 있음
- Feature_extractor, ann_classifier이 customized layer같은 성격을 가질 수 있게 됨
- VGG / ResNet / InceptionNet에 있는 layer block 역할처럼 쓰임!

Data Augmentation & Transfer Learning

PIAI Research Department

**Collecting data is
the most difficult thing**



Data Augmentation & Transfer Learning

PIAI Research Department

Collecting data is
the most difficult thing

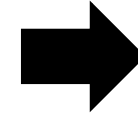
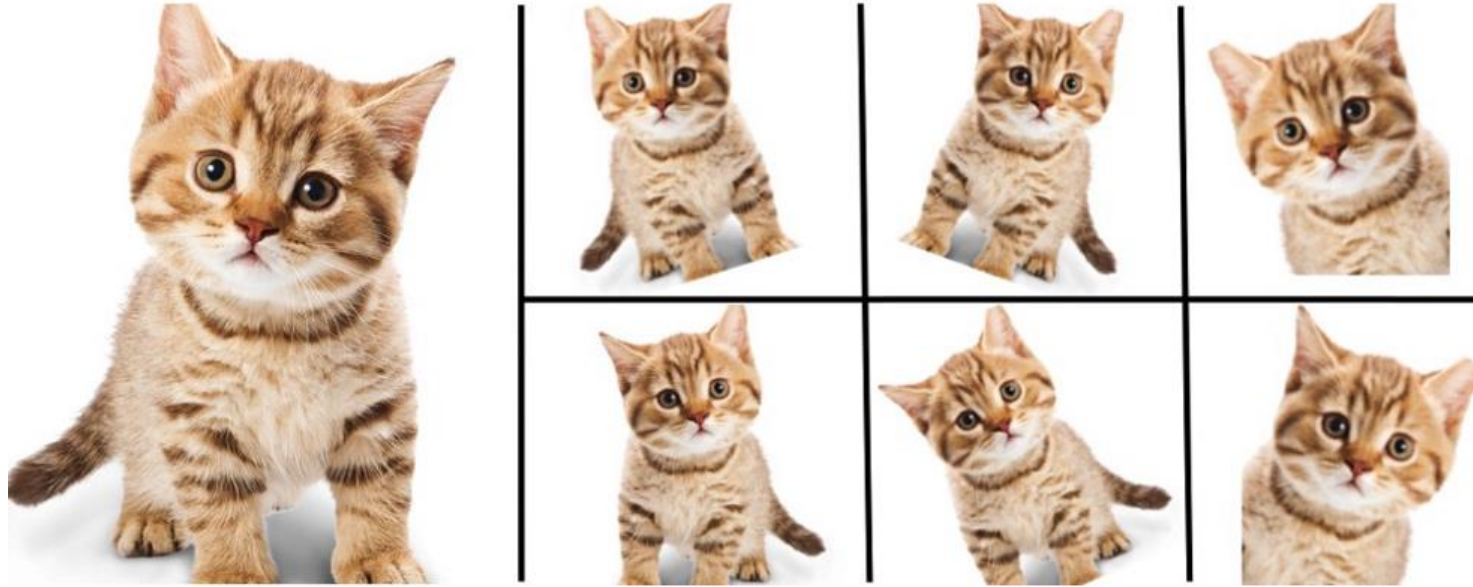


How to solve?

- Data Augmentation
- Transfer Learning
- Self-supervised Learning
- Semi-supervised Learning
- ...

Data Augmentation

PIAI Research Department



Keras ImageGenerator

Enlarge your Dataset

```
model.fit(X_train, Y_train,
```



```
model.fit(datagen.flow(X_train[:-10000], Y_train[:-10000], batch_size = 1000),
```

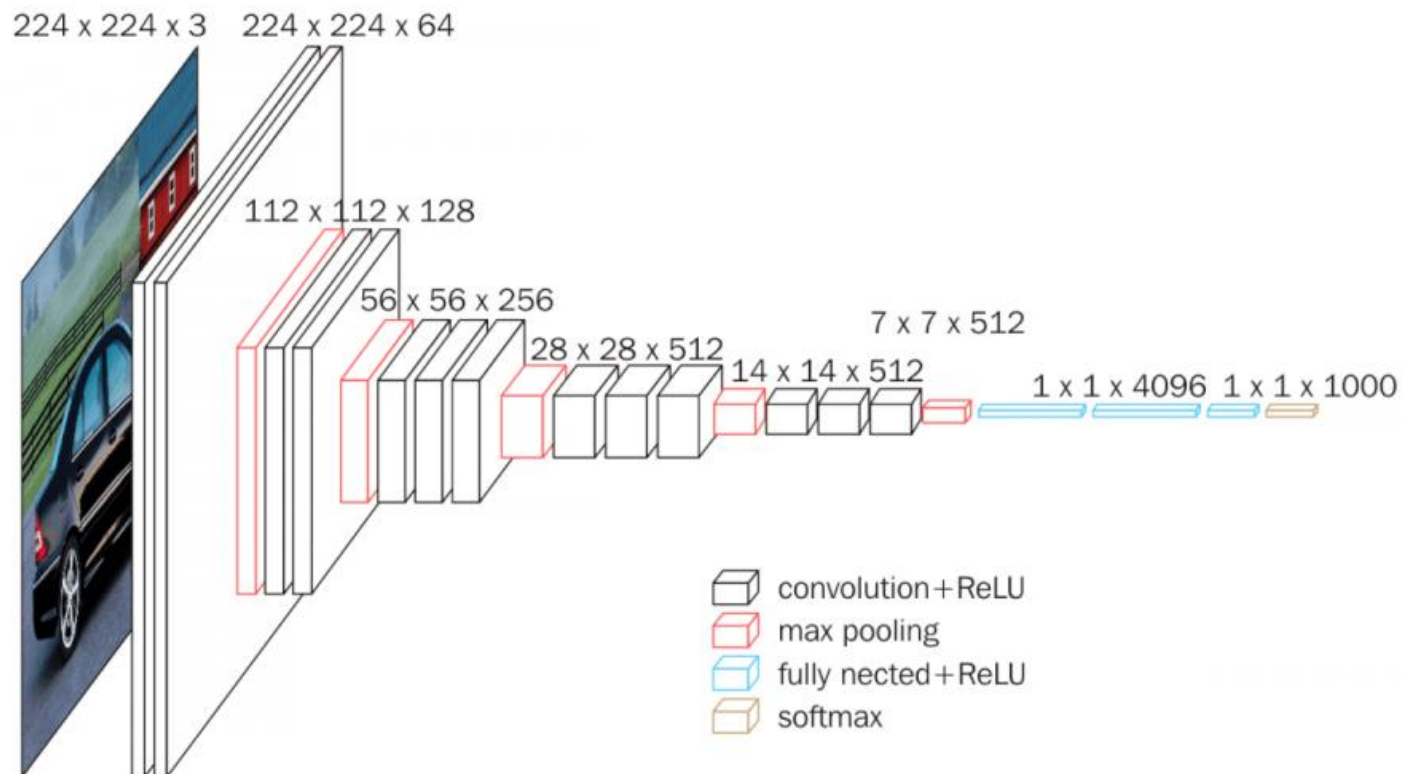
```
datagen = ImageDataGenerator(
    featurewise_center = False,
    samplewise_center = False,
    featurewise_std_normalization = False,
    samplewise_std_normalization = False,
    zca_whitening = False,
    rotation_range = 2, # 회전
    zoom_range = 0.1, # 확대 축소
    width_shift_range = 0.1, # 수평 이동
    height_shift_range = 0.1, # 수직 이동
    horizontal_flip = True, # 수평 반전
    vertical_flip = False # 수직 반전
)
```

```
datagen.fit(X_train)
```


Transfer Learning

PIAI Research Department

VGGNet

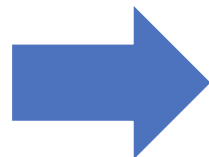
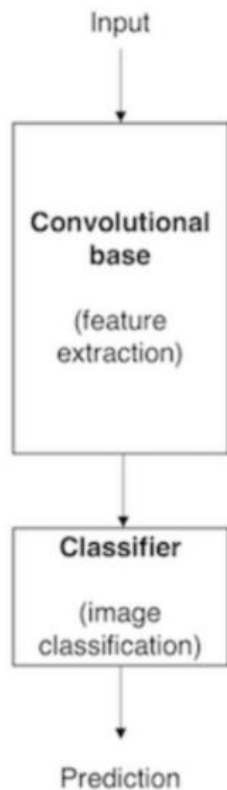


처음부터 모델을 구성하는 것이 아니
라 기존에 존재하는 성능 좋은 모델을
가져와 학습하는 기법

Transfer Learning

PIAI Research Department

Pretrained model



Strategy 1
Train the
entire model



Strategy 2
Train some layers and
leave the others frozen



Strategy 3
Freeze the
convolutional base



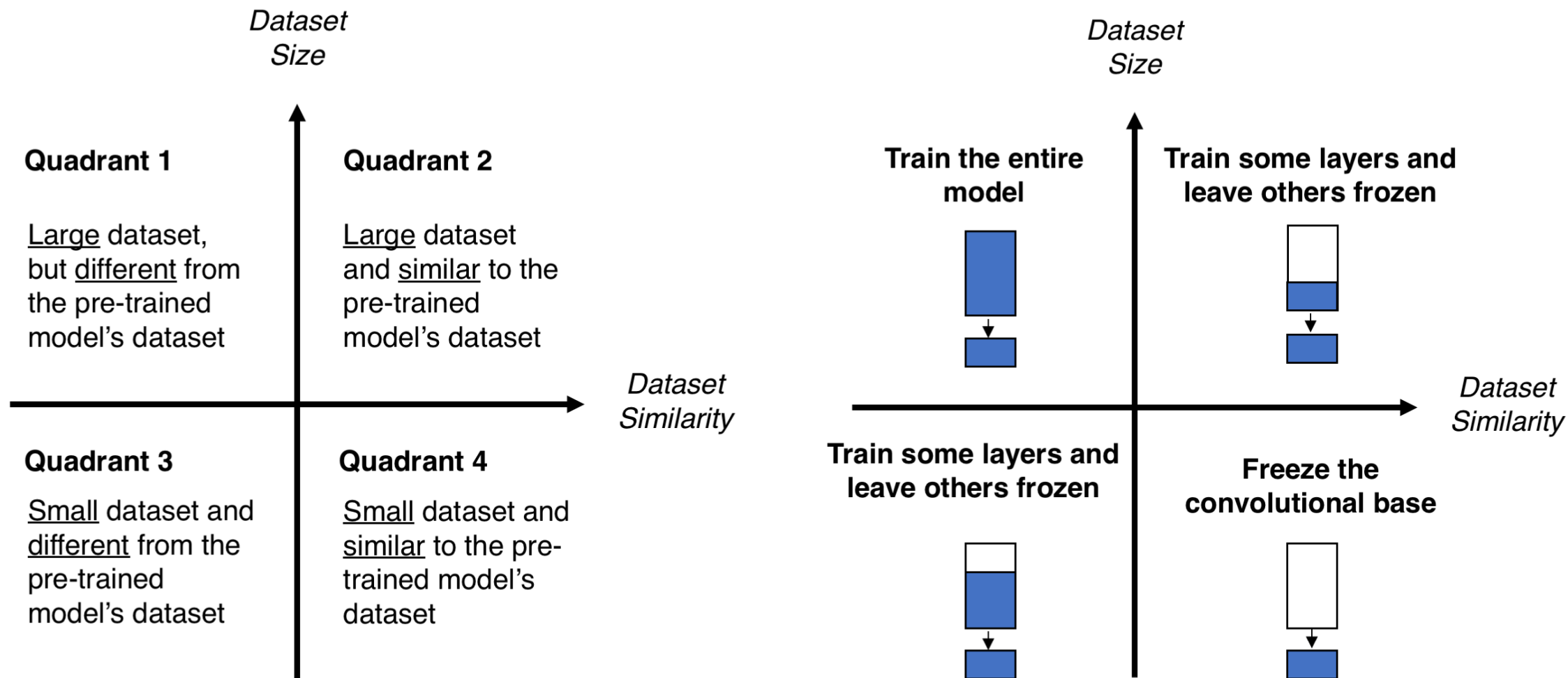
Legend:



Fine Tuning

Transfer Learning

PIAI Research Department



Code Running

(student) 3_KerasCNN – Additional Functions 1

- 머신러닝분석이슈

데이터

학습

성능

서비스

데이터

포맷 : 학습하기에 적절한 포맷인가?

규모 : 모델을 학습할 수 있을만큼 충분한 양인가? (Quantity / Level)

품질 : Outlier가 학습에 악영향을 끼칠만큼 있는가?

- 데이터 Outlier : 저품질이거나 대표성이 모호한 Outlier, 중복되는 데이터 (Noise Ratio)
- 데이터-라벨 Outlier : 데이터와 라벨에 대한 매칭이 잘못된 Outlier
- 학습/검증 Outlier : 학습과 검증하는 데이터의 편차 여부(10-fold validation)

구성 : 학습에 적절한 구성을 가지는가? (Imbalance / Bias / Breadth / Depth / 클래스 별 similarity)

라벨링 특성 : 데이터와 라벨의 구성이 합당한가? (성별 / 나이 / 표정)

학습

학습 방법 : 어떤 머신러닝 방식을 쓰는 것이 합당한가? (지도학습 / 비지도 학습 / ...)

데이터 연계 : 학습/ 평가하기에 적합한 전처리가 잘 진행되었는가?

네트워크 구조 :

- Layer나 Activation이 잘 설정 되었는가?(층 수, 종류, Input/Output 크기 등)
- 학습할 Parameter 수가 적정한가?

하이퍼 파라미터 : activation, loss, optimizer, learning_rate, metric들을 잘 조정했는가?

학습 방법 : learning rate schedule, batch_size, epoch, validation 구성이 적정한가?

학습 효율 : loss나 정확도가 충분히 수렴되었는가?

성능

Metric : 모델의 성능을 확인하기에 적절한 metric인가? (정량 성능과 실제 성능의 괴리)

예상치 : 모델의 성능이 예상되는 값의 범주에 속하는가? (모델의 성능 한계 예측)

Bias : 모델이 편향되도록 분류하거나 하진 않는가?

에러 패턴 : 결과 에러 중 흔하게 발생하는 경우가 있는가?

개선 방향성 : 네트워크? 학습 방법? 데이터 개선?

서비스

도메인 적합성 : 학습되는 데이터와 실제 사용될 데이터와의 차이가 없는가?

호환성 : 개발 모델이 실제 환경에 호환이 가능한 환경인가?(GPU, 이미지 크기 등)

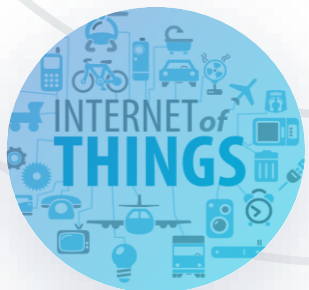
속도 : 해당 모델이 필요한 서비스에 적용가능한 속도인가? (Speed / Accuracy TradeOff)

백엔드 여부 : 모델 내 나타는 문제를 보완할 백엔드 기술이 필요한가?

적용성 : 쉬운 문제를 어렵게 풀건 아닐까..? (End-To-End / Process System)



PyTorch Additional Functions

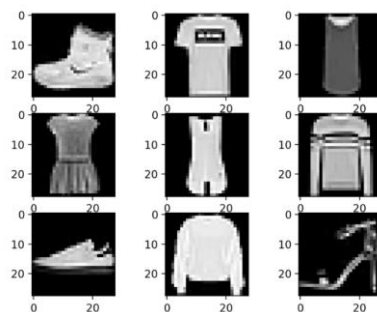
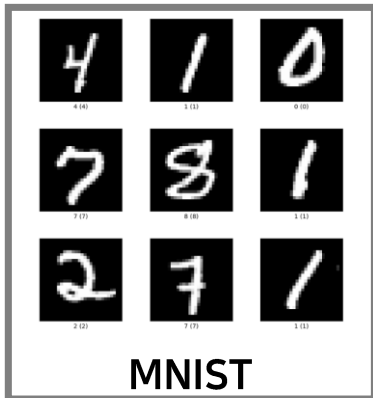


I-Eon, Na

CNN – 공인 데이터셋

PIAI Research Department

이전 수업에서 사용



Fashion MNIST



EMNIST

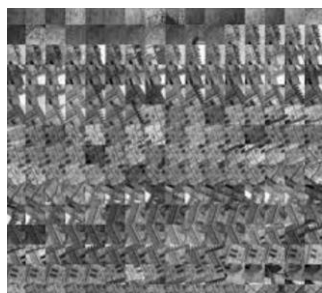


Photo tour



Flickr



CIFAR-10

- 머신러닝/딥러닝 모델을 평가하기 위한 다양한 공인 데이터셋
- MNIST와 다르게 RGB 3채널인 CIFAR-10
- 기타 PyTorch 데이터셋:

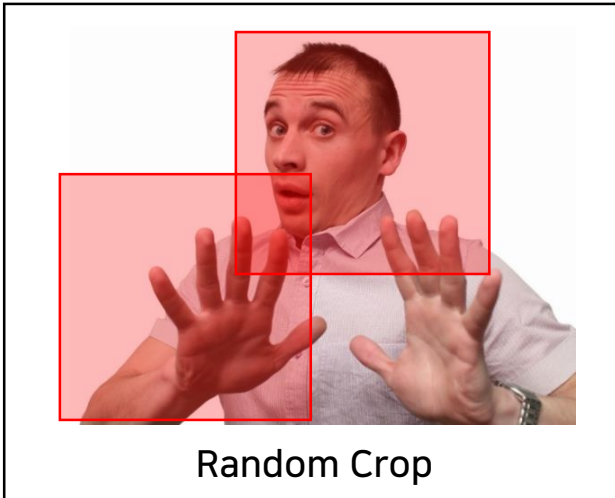
<https://teddylee777.github.io/pytorch/pytorch-mnist-dataloader-loading%ED%95%98%EA%B8%B0>

CNN – Data Augmentation

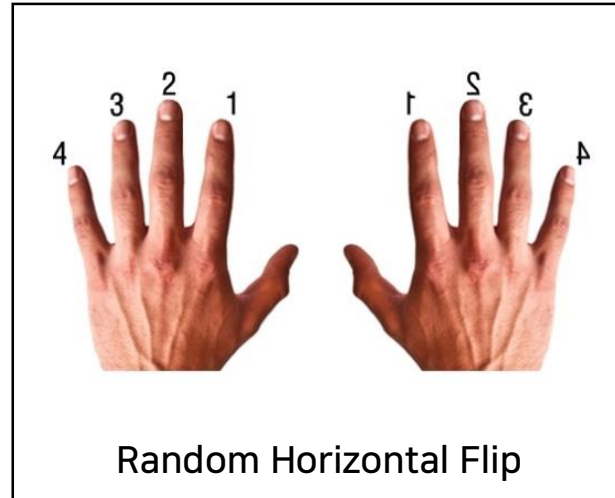
PIAI Research Department

• 데이터 증강 기법

```
# Load dataset into python variable
input_transform = transforms.Compose([
    Resize(56),
    RandomCrop(56, padding=6),
    RandomHorizontalFlip(),
    toTensor(),
])
```



출처: <https://bonlivre.tistory.com/833>



출처: <https://part2-what-do-we-want.tistory.com/11>

- 한정된 데이터로 더 정확도 높은 예측을 하기 위한 데이터 증강 기법

- Random Crop
- Random Horizontal Flip
- Image Rotation
- Brightness
- Contrast
- Saturation
- Hue
- Gaussian Noise
- Random Erasing

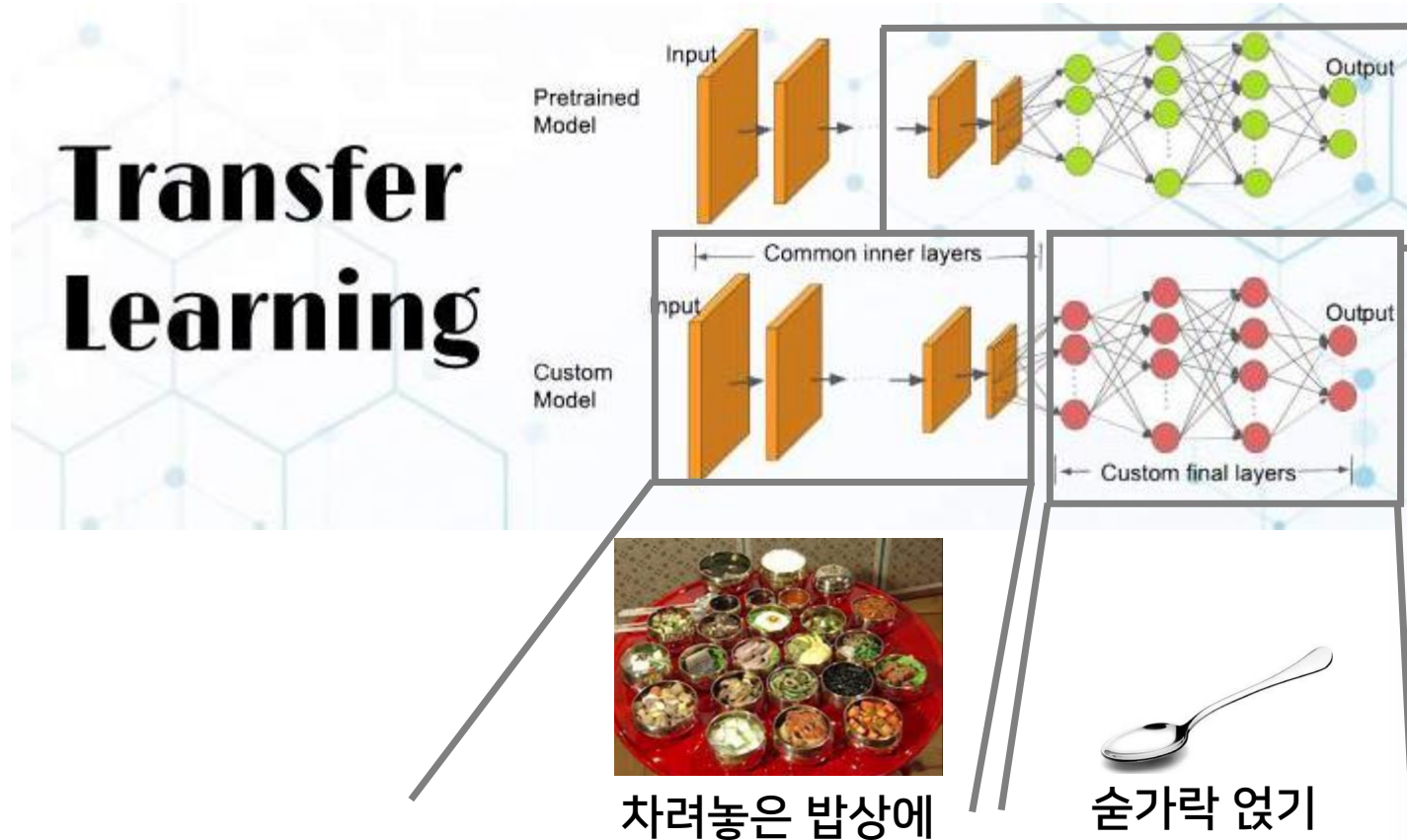
- 코드 사용 안내:

<https://androidkt.com/pytorch-image-augmentation-using-transforms/>

Transfer Learning – Custom Dataset Train

PIAI Research Department

- Save Time by training only few layers



- 학습 시간을 효과적으로 줄이기 위하여 이미 학습이 완료된 Pre-Trained 모델을 이용
- Pre-Trained 모델의 끝단 레이어만 학습
- 학습 시 필요한 데이터의 양과 학습 시간을 효율적으로 절약

Transfer Learning – Custom Dataset Train

PIAI Research Department

- Save Time by training only few layers

- AlexNet
- ConvNeXt
- DenseNet
- EfficientNet
- EfficientNetV2
- GoogLeNet
- Inception V3
- MNASNet
- MobileNet V2
- MobileNet V3
- RegNet
- ResNet
- ResNeXt
- ShuffleNet V2
- SqueezeNet
- SwinTransformer
- VGG
- VisionTransformer
- Wide ResNet

이번에 사용할 모델(ResNet 18)

```
resnet18 (*[, weights, progress])
```

ResNet-18 from Deep Residual Learning for Image Recognition.

```
resnet34 (*[, weights, progress])
```

ResNet-34 from Deep Residual Learning for Image Recognition.

```
resnet50 (*[, weights, progress])
```

ResNet-50 from Deep Residual Learning for Image Recognition.

```
resnet101 (*[, weights, progress])
```

ResNet-101 from Deep Residual Learning for Image Recognition.

```
resnet152 (*[, weights, progress])
```

ResNet-152 from Deep Residual Learning for Image Recognition.

Transfer Learning – Custom Dataset Train

PIAI Research Department

- 데이터셋 준비 – 원숭이와 사람 구분하기



- 분류하기 원하는 사진을 class 당 개수가 동일하도록 준비
- 데이터셋의 양은 class당 최대 1,000장

[How Do You Know You Have Enough Training Data?](#)

2019. 4. 22. — Computer Vision: For image classification using deep learning, a **rule of thumb** is **1,000 images per class**, where this number can go down ...

Transfer Learning – Custom Dataset Train

PIAI Research Department

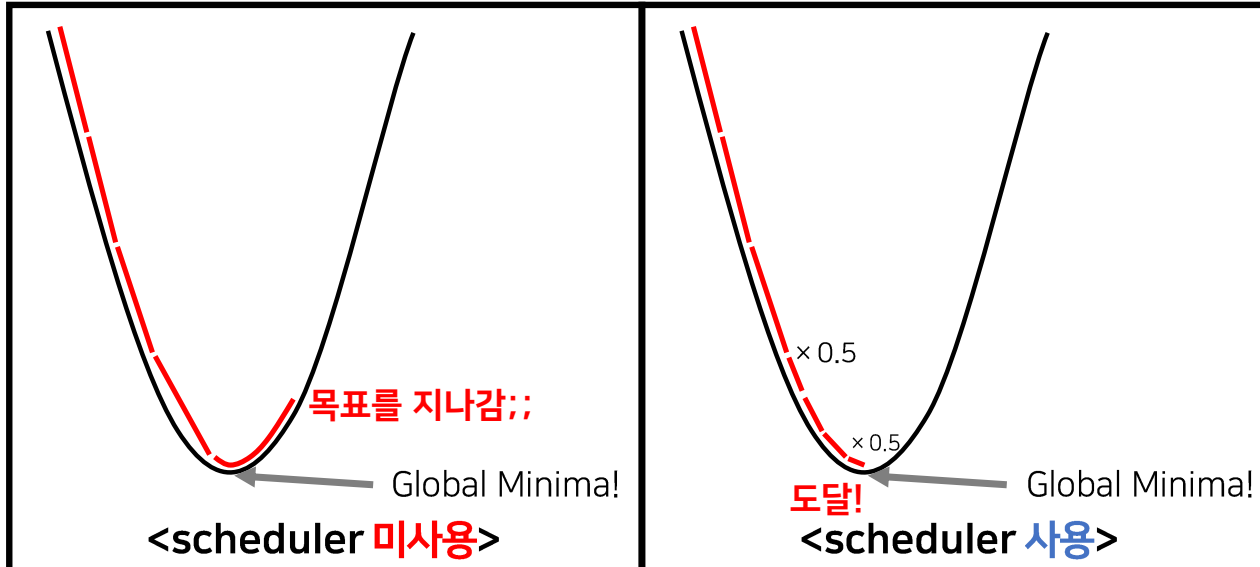
• Pre-Train Model – ResNet 18 이용

```
from torchvision import models
model_resnet = models.resnet18(pretrained = True)
```

```
num_fts = model_resnet.fc.in_features
model_resnet.fc = nn.Linear(num_fts, 2)
net = model_resnet.to(device)
```

```
maximum_epoch = 10
loss_fn = nn.CrossEntropyLoss()
optim = Adam(net.parameters(), lr=0.00001)
from torch.optim import lr_scheduler
lr_scheduler = lr_scheduler.StepLR(optim, step_size = 3, gamma = 0.5)
```

- 모델을 선언하고, pretrained 모드 설정
- 마지막 *FC 레이어의 입, 출력 개수를 변경
- 하이퍼 파라미터 설정
- 학습 추이에 따라 Learning Rate가 줄어들어 학습에 유리한 Scheduler 설정



FC레이어: Fully Connected Layer

Transfer Learning – Custom Dataset Train

PIAI Research Department

- Pre-trained 모델 전이 학습 결과



```

Iter: 10 >> T_loss 0.24788   T_acc 0.90714   V_loss 0.06752   V_acc 1.00000   4.119s
Iter: 9  >> T_loss 0.26694   T_acc 0.85714   V_loss 0.05087   V_acc 1.00000   3.900s
Iter: 8  >> T_loss 0.23141   T_acc 0.93571   V_loss 0.04966   V_acc 1.00000   3.961s
Iter: 7  >> T_loss 0.20230   T_acc 0.94286   V_loss 0.05406   V_acc 1.00000   4.053s
Iter: 6  >> T_loss 0.26257   T_acc 0.88571   V_loss 0.06644   V_acc 1.00000   4.012s
Iter: 5  >> T_loss 0.20305   T_acc 0.96429   V_loss 0.07678   V_acc 1.00000   3.990s
Iter: 4  >> T_loss 0.27384   T_acc 0.89286   V_loss 0.09413   V_acc 1.00000   3.985s
Iter: 3  >> T_loss 0.26819   T_acc 0.92857   V_loss 0.12793   V_acc 1.00000   3.947s
Iter: 2  >> T_loss 0.36696   T_acc 0.87857   V_loss 0.19387   V_acc 0.97500   4.015s
Iter: 1  >> T_loss 0.56073   T_acc 0.70000   V_loss 0.41837   V_acc 0.77500   4.255s
    
```

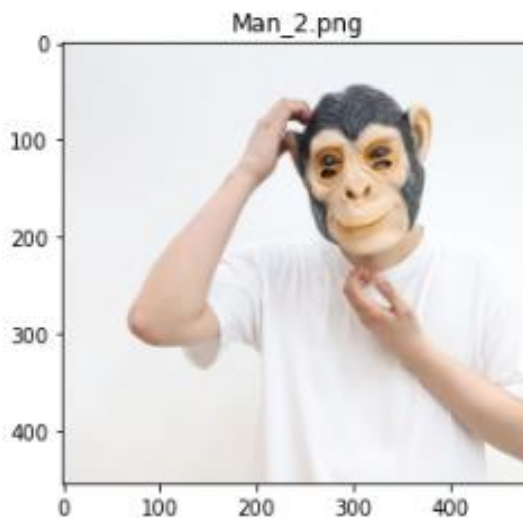
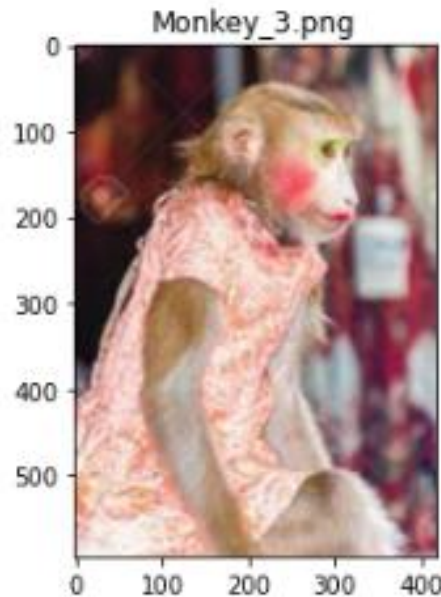
Best val Acc: 1.000000

- 3 epoch부터 Validation 분류 정확도 100% 달성
- [Weakly To Do]
직접 설계한 모델과 Transfer Learning의 성능을 CIFAR-10 데이터셋 기준으로 비교해보자

Transfer Learning – Custom Dataset Train

PIAI Research Department

나는 원숭이일까? 사람일까?



- 원숭이 탈을 쓴 사람과, 사람의 옷을 입은 원숭이를 모델이 제대로 분류할 수 있을까?
- 구분한다면 원리가 무엇일까?

Code Running

(student) 3_Pytorch CNN – Additional Functions 1