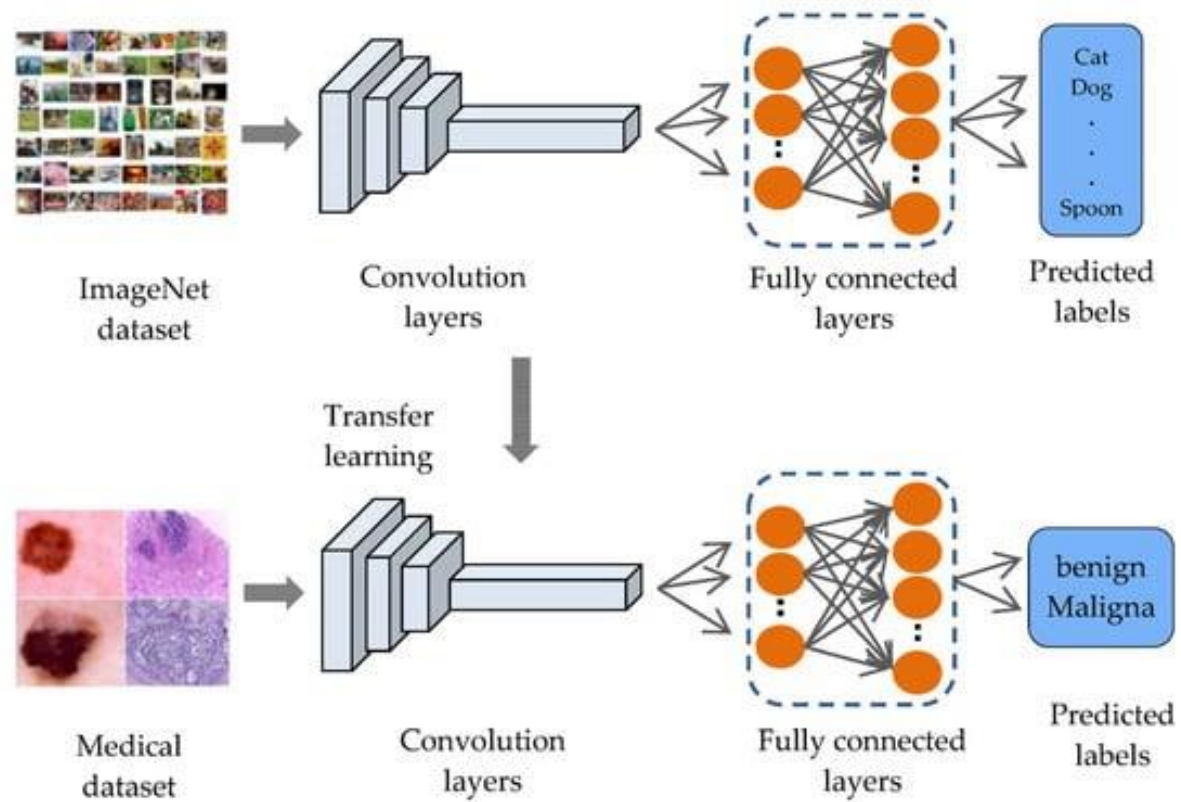


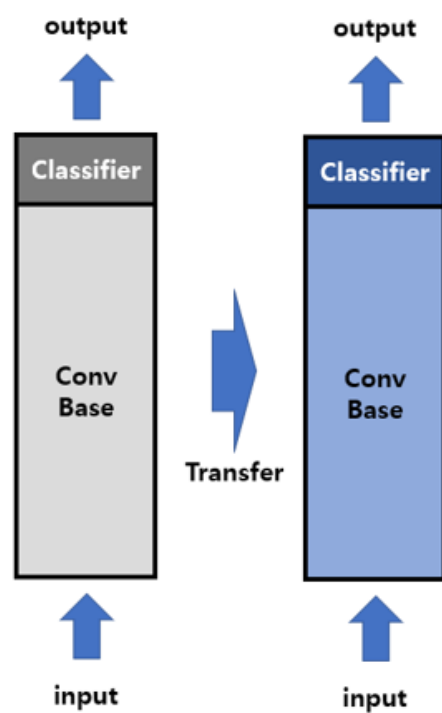
AI 실전 3주차

이미 학습이 되어 있는 모델을 학습하여 이용하기
전이 학습

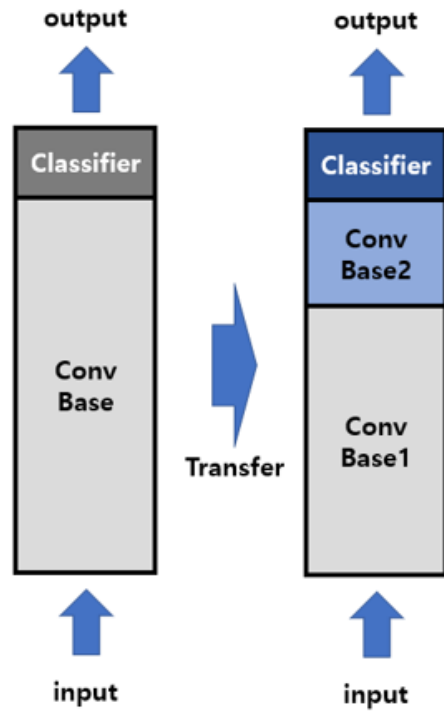
전이 학습



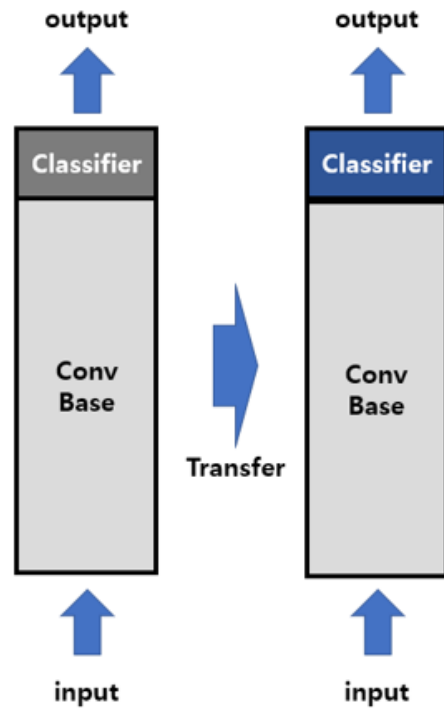
전이 학습 적용



Strategy 1
: 모델 전체를 학습하는 경우



Strategy 2
: 일부만 학습시키는 경우

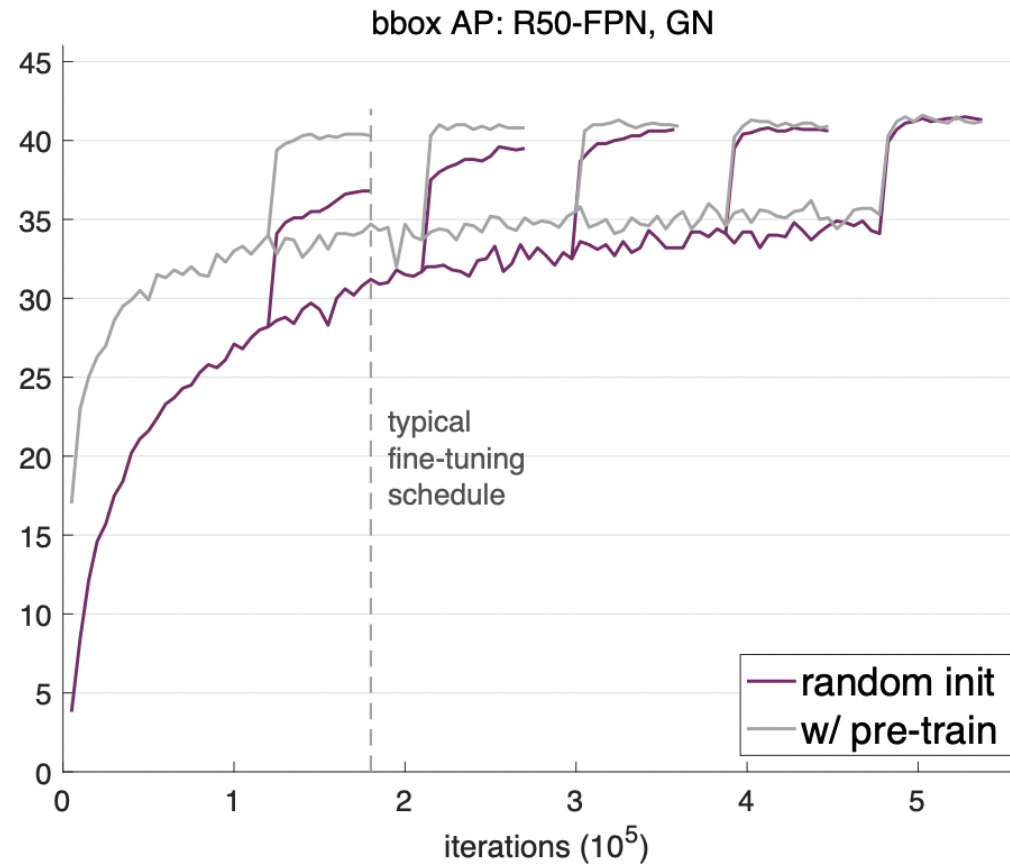


Strategy 3
: 학습 과정 없이 사용하는 경우

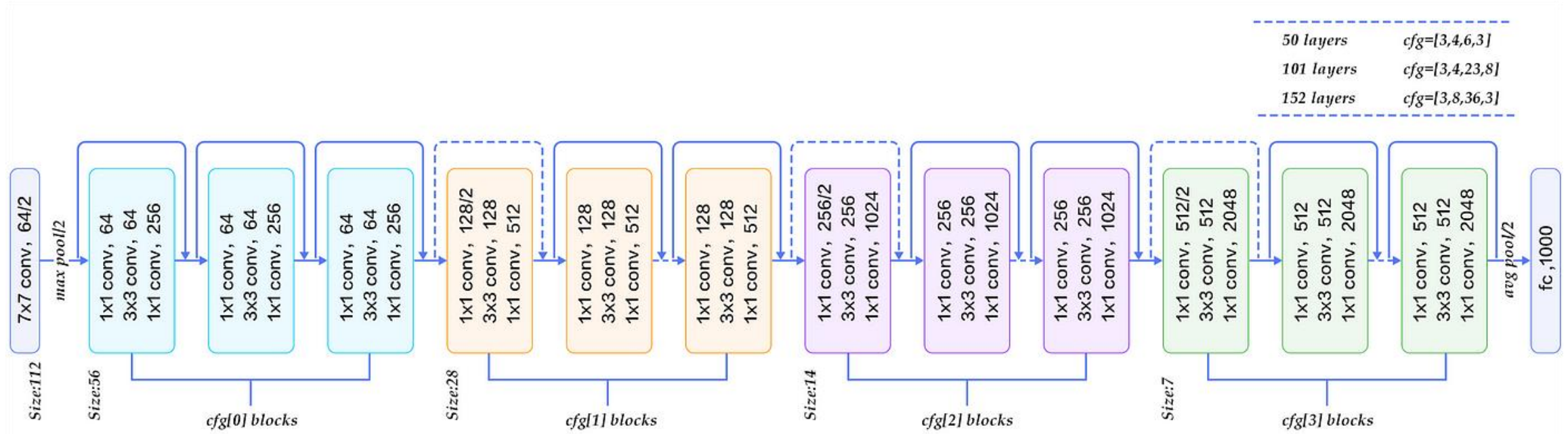
최근 파인튜닝 적용

- P-tuning
- Prefix tuning
- Prompt tuning
- LoRA
- ...

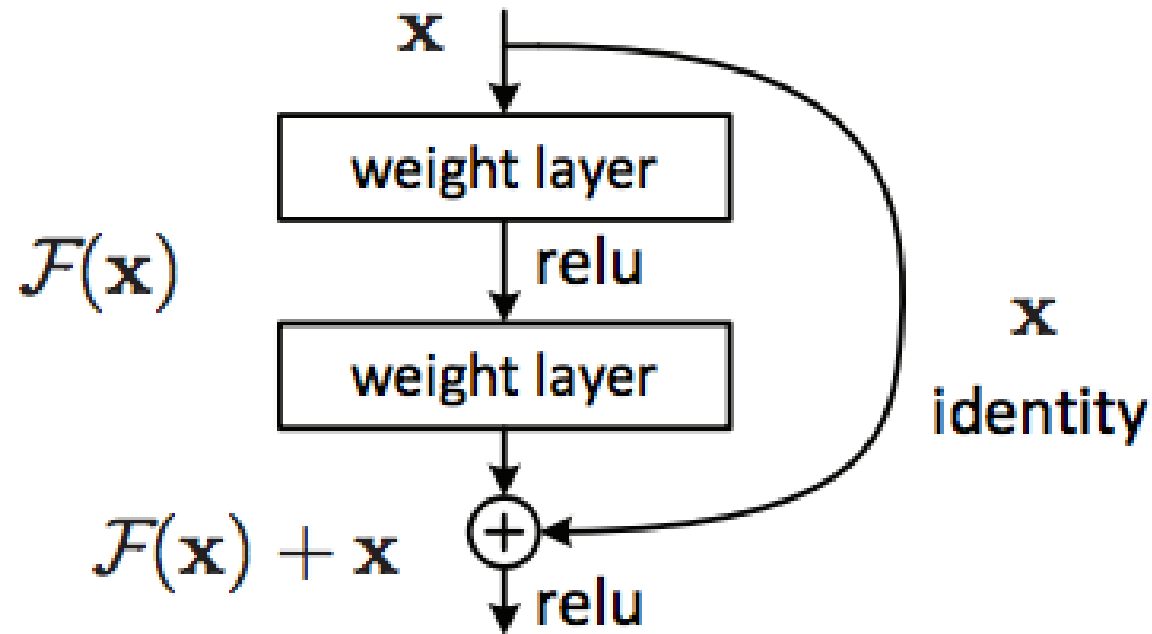
Q. 굳이 학습된 모델 안써도 되잖..?



resnet50



Resnet 적용한 기술 (skip connections)



필요한 라이브러리 불러오기



```
import torchvision.transforms as transforms
import torchvision.datasets as datasets
from torch.utils.data import DataLoader
```


학습할 데이터 변환



데이터 변환 정의

```
transform = transforms.Compose([  
    transforms.Resize((224, 224)),  
    transforms.ToTensor(),  
])
```

다운로드 안되는 학생을 위한 SSL 등록



```
# pip install certifi
import ssl
ssl._create_default_https_context = ssl._create_unverified_context
```

0. Download CIFAR10 Dataset



```
train_dataset = datasets.CIFAR10(root='./data', train=True,  
transform=transform, download=True)  
train_loader = DataLoader(train_dataset, batch_size=32, shuffle=True)
```

Load resnet50 model



```
import torch
import torch.nn as nn
import torchvision.models as models

# 모델 불러오기
model = models.resnet50(pretrained=True)
for param in model.parameters():
    param.requires_grad = False
```

OUTPUT Layer 데이터 맞춰서 변경




```
num_ftrs = model.fc.in_features  
model.fc = nn.Linear(num_ftrs, 10)
```

모델 학습 함수 생성

```
def train_model(model, dataloader, criterion, optimizer, num_epochs=10, device='cuda'):  
    model.to(device)  
    model.train()  
    for epoch in range(num_epochs):  
        running_loss = 0.0  
        for inputs, labels in dataloader:  
            inputs, labels = inputs.to(device), labels.to(device)  
            optimizer.zero_grad()  
            outputs = model(inputs)  
            loss = criterion(outputs, labels)  
            loss.backward()  
            optimizer.step()  
            running_loss += loss.item()  
        print(f'Epoch {epoch+1}/{num_epochs}, Loss:{running_loss/len(dataloader)}')  
    print('Training complete')
```

모델 학습



```
import torch.optim as optim

# 학습 설정
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.fc.parameters(), lr=0.001)
# 모델 학습
train_model(model, train_loader, criterion, optimizer, num_epochs=5, device='cpu')
```