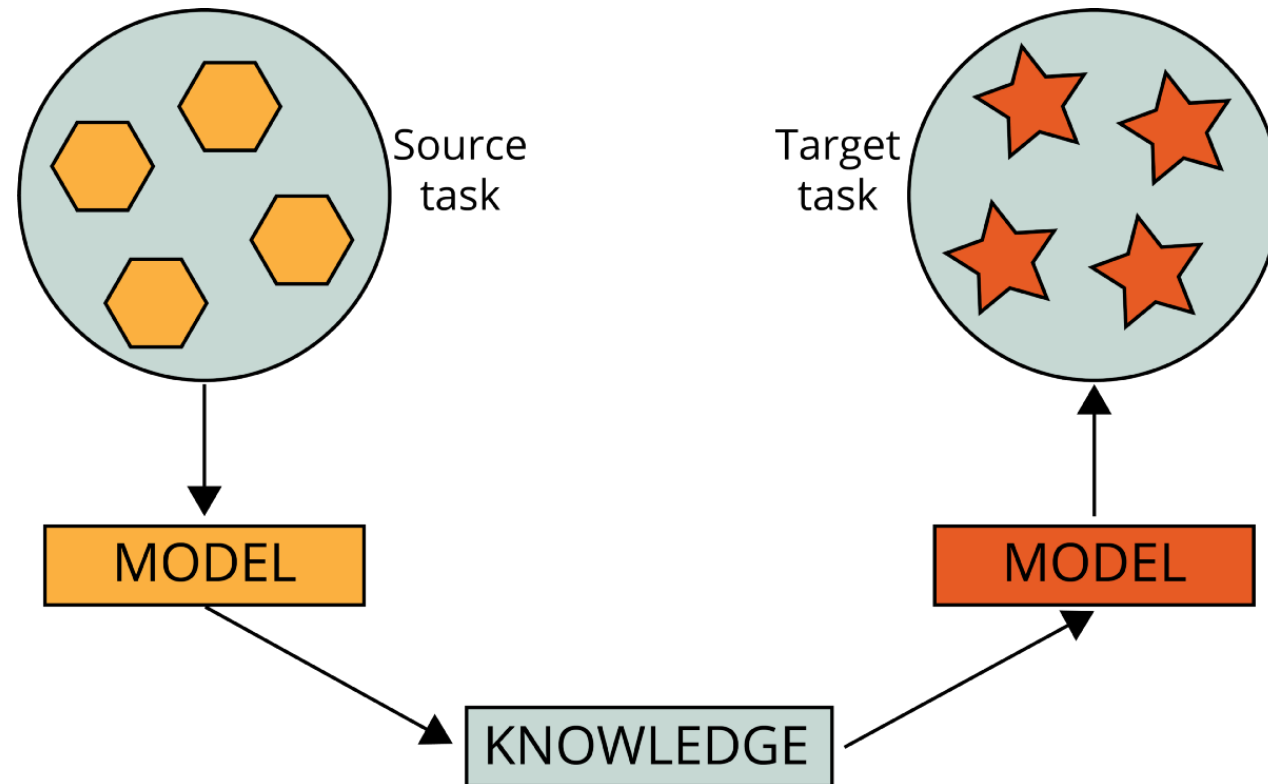


Transfer Learning



전이학습(Transfer Learning)

■ 전이학습

- 실제로 충분한 크기의 데이터셋을 갖추기는 상대적으로 드물기 때문에, (무작위 초기화를 통해) 맨 처음부터 합성곱 신경망(Convolutional Network) 전체를 학습하는 사람은 매우 적습니다.
- 대신, 매우 큰 데이터셋(예. 100가지 분류에 대해 120만개의 이미지가 포함된 ImageNet)에서 합성곱 신경망(ConvNet)을 미리 학습한 후, 이 합성곱 신경망을 관심있는 작업 을 위한 초기 설정 또는 고정된 특징 추출기(fixed feature extractor)로 사용합니다.

■ 전이학습 시나리오

- 합성곱 신경망의 미세조정(finetuning): 무작위 초기화 대신, 신경망을 ImageNet 1000 데이터셋 등으로 미리 학습한 신경망으로 초기화합니다. 학습의 나머지 과정들은 평상시와 같습니다.
- 고정된 특징 추출기로서의 합성곱 신경망: 여기서는 마지막에 완전히 연결 된 계층을 제외한 모든 신경망의 가중치를 고정합니다.
마지막의 완전히 연결된 계층은 새로운 무작위의 가중치를 갖는 계층으로 대체되어 이 계층만 학습합니다.

데이터

- 데이터 : https://download.pytorch.org/tutorial/hymenoptera_data.zip

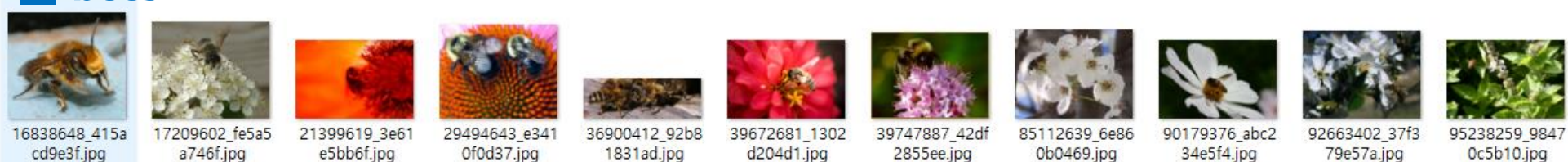
```
import urllib
import zipfile
import shutil
```

```
url = 'https://download.pytorch.org/tutorial/hymenoptera_data.zip'
filename = 'hymenoptera_data.zip'
urllib.request.urlretrieve(url, filename=filename)
zipfile.ZipFile(filename).extractall()
shutil.move('hymenoptera_data', 'data/hymenoptera_data')
```

ants



bees



Fine Tuning

```
model_ft = models.resnet18(pretrained=True)
num_ftrs = model_ft.fc.in_features

# 여기서 각 출력 샘플의 크기는 2로 설정합니다.
# 또는, nn.Linear(num_ftrs, len(class_names))로 일반화할 수 있습니다.
model_ft.fc = nn.Linear(num_ftrs, 2)

model_ft = model_ft.to(device)

criterion = nn.CrossEntropyLoss()

# 모든 매개변수들이 최적화되었는지 관찰
optimizer_ft = optim.SGD(model_ft.parameters(), lr=0.001, momentum=0.9)

# 7 에폭마다 0.1씩 학습률 감소
exp_lr_scheduler = lr_scheduler.StepLR(optimizer_ft, step_size=7, gamma=0.1)
```

학습 및 평가하기

- CPU에서 실행하는 경우 이전과 비교했을 때 약 절반 가량의 시간만이 소요될 것입니다.
- 이는 대부분의 신경망에서 경사도를 계산할 필요가 없기 때문입니다.
- 하지만, 순전파는 계산이 필요합니다..

```
model_ft = train_model(model_ft, criterion, optimizer_ft, exp_lr_scheduler,  
                        num_epochs=25)
```

```
visualize_model(model_ft)
```

predicted: bees



predicted: bees



predicted: bees



predicted: bees



predicted: bees



predicted: ants



고정된 특징 추출기로서의 합성곱 신경망

- 마지막 계층을 제외한 신경망의 모든 부분을 고정해야 합니다.
- `requires_grad == False` 로 설정하여 매개변수를 고정하여 `backward()` 중에
- 경사도가 계산되지 않도록 해야 합니다.

```
model_conv = torchvision.models.resnet18(pretrained=True)
for param in model_conv.parameters():
    param.requires_grad = False
```

```
# 새로 생성된 모듈의 매개변수는 기본값이 requires_grad=True 임
num_fters = model_conv.fc.in_features
model_conv.fc = nn.Linear(num_fters, 2)
```

```
model_conv = model_conv.to(device)
```

```
criterion = nn.CrossEntropyLoss()
```

```
# 이전과는 다르게 마지막 계층의 매개변수들만 최적화되는지 관찰
```

```
optimizer_conv = optim.SGD(model_conv.fc.parameters(), lr=0.001, momentum=0.9)
```

```
# 7 에폭마다 0.1씩 학습률 감소
```

```
exp_lr_scheduler = lr_scheduler.StepLR(optimizer_conv, step_size=7, gamma=0.1)
```

고정된 특징 추출기로써의 합성곱 신경망

`visualize_model(model_conv)`

predicted: bees



predicted: ants



predicted: ants



predicted: bees



predicted: bees



predicted: ants



Thank you