

## 전이학습 실습해 보기

### 학습 내용

- 환경 설정 및 라이브러리 로드
- 데이터 준비
- 모델 구성 및 전이 학습 적용
- 이미지 객체 인식

## 전이학습에 대한 이해 및 ResNet 이해해 보기

### 객체 인식

- 전이학습
- "전이학습"은 한 작업에서 학습한 모델을 다른 관련 작업에 맞게 조정하는 기계 학습 기법입니다.
- 처음부터 모델을 학습하는 대신, 일반적으로 대규모 데이터셋에서 학습된 사전 학습된 모델을 시작점으로 사용

### ResNet 예제

- ResNet50은 딥러닝 분야에서 널리 사용되는 심층 신경망(Deep Neural Network) 아키텍처 중 하나
- ResNet(Residual Network)은 2015년 마이크로소프트 리서치 팀이 개발한 모델
- 딥러닝에서 흔히 발생하는 **기울기 소실 문제(Vanishing Gradient Problem)**를 해결하기 위해 도입

### 특징

- 깊은 네트워크: ResNet50은 "50"이라는 숫자가 나타내듯이, 총 50개의 층(layer)으로 구성되어 있습니다. 이러한 깊은 구조는 이미지의 매우 복잡한 패턴을 학습하는 데 적합. (예: ResNet101, ResNet152 등)
- 높은 성능: ResNet은 ImageNet 데이터셋에서 매우 높은 성능을 기록하며 이미지 분류, 객체 탐지, 이미지 세분화 등 다양한 컴퓨터 비전 작업에서 널리 사용

```
In [2]: # 필요한 라이브러리를 가져옵니다.
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.preprocessing import image
import numpy as np
from tensorflow.keras.applications.resnet50 import preprocess_input, decode_predictions
from keras.applications.imagenet_utils import preprocess_input
```

```
In [4]: # 이미지를 로드하고 전처리하는 함수
def preprocess_image(image_path):
    img = image.load_img(image_path, target_size=(224, 224))
```

```
img = image.img_to_array(img) # 이미지를 배열로 변환
img = np.expand_dims(img, axis=0) # 모델이 기대하는 형태로 만든다. (1, 224, 224, 3)
img = preprocess_input(img) # 이미지 데이터를 모델에 적합한 형태로 전처
return img
```

```
In [5]: # 이미지 객체 인식을 수행하는 함수
def predict_objects(image_path):
    # 사전에 훈련된 ResNet50 모델을 불러옵니다.
    model = keras.applications.resnet50.ResNet50(weights='imagenet')

    # 이미지를 전처리합니다.
    img = preprocess_image(image_path)

    # 이미지를 모델에 입력하여 예측을 수행합니다.
    preds = model.predict(img)

    # 예측 결과를 해석합니다.
    decoded_preds = decode_predictions(preds, top=3)[0]

    # 예측 결과를 출력합니다.
    for _, label, confidence in decoded_preds:
        print(f"{label}: {confidence * 100}%")
```

```
In [6]: from IPython.display import Image
```

## 파일 테스트

```
In [7]: # 이미지 파일 경로를 지정합니다.
image_path = 'Dog_rawPixel01.jpg'

# 이미지 표시
Image(image_path)
```

Out[7]:



```
In [8]: # 이미지 객체 인식을 수행합니다.  
predict_objects(image_path)
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/resnet/resnet50_weights_tf_dim_ordering_tf_kernels.h5  
102967424/102967424 ————— 1s 0us/step  
1/1 ————— 2s 2s/step  
Downloading data from https://storage.googleapis.com/download.tensorflow.org/data/imagenet_class_index.json  
35363/35363 ————— 0s 0us/step  
standard_poodle: 93.5053288936615%  
miniature_poodle: 4.46309819817543%  
golden_retriever: 0.6950932554900646%
```

## 결과

- 스탠다드 푸들: 93.5053288936615%
- 미니어처 푸들: 4.46309819817543%
- 골든 리트리버: 0.6950932554900646%

```
In [9]: # 이미지 파일 경로를 지정합니다.  
image_path = 'life_unsplash.jpeg'  
  
# 이미지 표시  
Image(image_path)
```

Out[9]:



```
In [10]: # 이미지 객체 인식을 수행합니다.  
         predict_objects(image_path)
```

```
1/1 ————— 2s 2s/step  
mosquito_net: 32.021260261535645%  
crib: 20.114001631736755%  
quilt: 18.42556595802307%
```

- 결과
- 모기장: 32.021260261535645%
- 아기 침대: 20.114001631736755%
- 이불: 18.42556595802307%

```
In [11]: # 이미지 파일 경로를 지정합니다.  
         image_path = 'Deer_un.jpeg'  
  
         # 이미지 표시  
         Image(image_path)
```

Out[11]:



```
In [ ]: # 이미지 객체 인식을 수행합니다.  
        predict_objects(image_path)
```

```
1/1 [=====] - 1s 1s/step  
ibex: 57.7500581741333%  
impala: 12.893131375312805%  
gazelle: 10.757070779800415%
```

**이벡스(Ibex):** 이벡스는 산악 지형에 서식하는 동물로, 주로 유럽, 아시아, 아프리카 지역에서 발견됩니다. 큰 뿔과 특이한 외모를 가지고 있으며, 주로 암벽이나 절벽에서 생활합니다.

**임팔라(Impala):** 임팔라는 아프리카 대륙에서 서식하는 큰 영양 동물로, 주로 사바나와 숲 지역에서 발견됩니다. 얇은 몸과 날씬한 다리를 가지고



있으며, 높은 점프력과 빠른 속도로 알려져 있습니다.

**가젤(Gazelle):** 가젤은 아프리카와 아시아 지역에서 발견되는 작은 동물로, 주로 사막과 초원에서 서식합니다. 가늘고 우아한 몸과 긴 다리를 가지고 있으며, 빠른 속도와 재빠른 움직임으로 유명합니다.

## 추가적인 학습해 보기

- 다양한 사전 학습 모델 비교: VGG, ResNet, Inception, EfficientNet 등 다양한 사전 학습 모델을 사용하여 성능 차이를 비교
- 고정 특징 추출 vs. 미세 조정(Fine-Tuning): 전이 학습을 할 때 사전 학습된 모델의 일부 층을 고정한 채로 사용하는 것과 전체 모델을 미세 조정하는 것의 성능 차이를 분석
- 데이터 증강(Data Augmentation): 이미지의 회전, 이동, 확대/축소, 색상 변화 등 다양한 증강 기법을 사용하여 모델의 일반화 성능을 향상
- 강화 학습(Reinforcement Learning)과의 통합: 이미지 인식 기술을 강화 학습 알고리즘과 결합하여, 예를 들어 자율주행차의 시각 시스템에서 물체 인식을 강화할 수 있는 방법을 탐구
- 모델 설명력 향상: 이미지 인식 모델이 특정 예측을 내린 이유를 시각화하거나 설명하는 기술(예: Grad-CAM, LIME)을 탐구
- 모델 압축 및 경량화: MobileNet, SqueezeNet 등과 같은 경량 모델을 탐구하고, 프루닝(pruning), 양자화(quantization), 지식 증류(Knowledge Distillation) 등의 기법을 통해 모델을 최적화하는 방법을 연구
- 실시간 이미지 인식: 실시간 응용을 위한 모델의 속도와 정확도 균형을 맞추기 위해 다양한 최적화 기법을 실험
- 다중 모달 데이터 통합: 텍스트, 오디오, 비디오 데이터와 함께 이미지 데이터를 사용하여 보다 풍부한 정보 표현을 위해 다중 모달 학습 기법을 연구

In [ ]: