

## 인공지능 HW4

인공지능학부 215001 서가연

### 01. 전통적인 영상 인식 방법과 신경망을 이용한 영상 인식 방법의 차이점을 설명하라.

전통적인 영상 인식 방법에서는 인간이 수동으로 특징을 추출하거나 선택해 이를 바탕으로 분류 등의 작업을 수행했다면, 신경망을 이용한 영상 인식 방법에서는 특징 추출과 분류 등의 작업이 동시에 진행된다. 따라서 이를 전체 학습또는 종단학습(end-to-end learning)이라고 한다.

### 02. CIFAR-10 데이터 세트를 기본적인 심층 신경망으로 처리하는 프로그램을 작성해보자. 본문의 컨볼루션 신경망과 비교했을 때 어떤 쪽이 더 성능이 높은가?

```
cnn_model = Sequential([
    Conv2D(64, kernel_size=(3, 3), activation='relu'),
    MaxPooling2D(pool_size=(2, 2)),
    Conv2D(32, (3, 3), activation='relu'),
    Flatten(input_shape=(32, 3)),
    Dense(80, activation = 'relu'),
    Dense(10, activation = 'softmax')
])
```

컨볼루션 신경망 모델

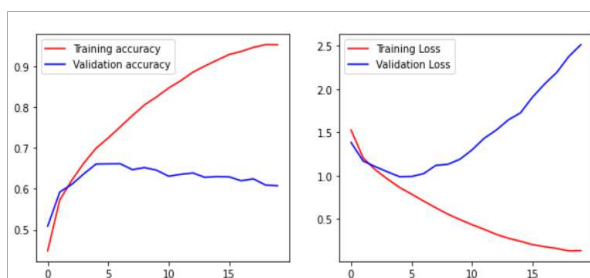
```
dnn_model = Sequential([
    Flatten(input_shape=(32, 32, 3)),
    Dense(1024, activation = 'relu'),
    Dense(512, activation = 'relu'),
    Dense(256, activation = 'relu'),
    Dense(128, activation = 'relu'),
    Dense(64, activation = 'relu'),
    Dense(32, activation = 'relu'),
    Dense(10, activation = 'softmax')
])
```

심층신경망 모델

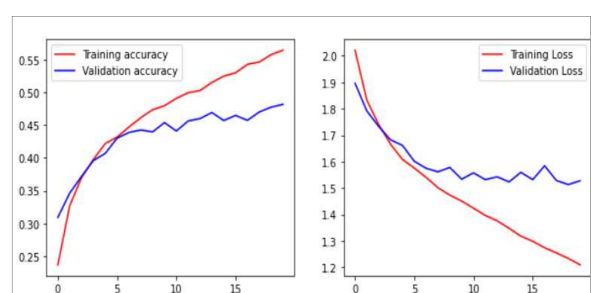
```
cnn_model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
0.9s

history = cnn_model.fit(X_train, y_train, epochs=20, validation_split=0.3, verbose=1)
```

모델 컴파일 및 학습(공통)



CNN 모델의 학습 곡선



DNN 모델의 학습 곡선

CNN 모델은 과대적합 문제가 발생한 것으로 보이나, 20 epochs 기준 DNN보다 더 빠르게 수렴하고 있는 것을 알 수 있으며, 과대적합 문제를 해결하는 다양한 기법을 적용하면 모델의 성능이 더 증가할 것으로 예상된다. 반면 DNN의 경우, CNN 모델보다는 수렴 속도가 느리지만 학습 횟수를 높이면 성능이 더 증가할 것으로 예상된다.

### 03. 데이터 증대(data augmentation)라는 것은 무엇이며, 왜 필요한가?

데이터 증대란 소량의 훈련 데이터에 인위적인 변형을 가하여 많은 수의 샘플을 생성해내는 것이다. 이를 통해 샘플의 수를 늘려 모델의 과대적합을 방지할 수 있다.

### 04. 케라스의 ImageDataGenerator() 메소드를 이용해서 주어진 이미지를 다양하게 변형하는 프로그램을 작성해보자.

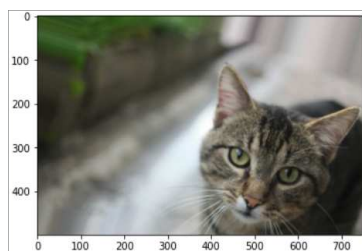
```
datagen = ImageDataGenerator(  
    rescale=1./255,  
    rotation_range=50, # rotation_range=-70  
    horizontal_flip=True,  
    vertical_flip=True,  
    zoom_range=[0.1, 1.5],  
    brightness_range=[0.01, 2.0],  
    width_shift_range=0.8,  
    height_shift_range=0.9)  
  
aug_image(sample)
```

```
# 증강 이미지 시각화  
def aug_image(image):  
    image = np.expand_dims(image, axis=0) # 4차원으로 늘려주기  
    obj = datagen.flow(image, batch_size=1)  
    aug_image = obj.next()  
    plt.imshow(aug_image[0])  
  
✓ 0.3s
```

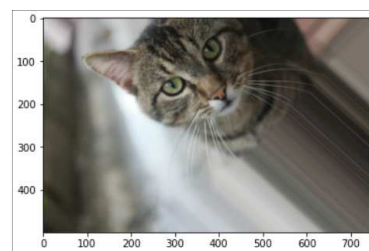
\* 아래는 주석 포함 코드 실행 결과, 실제 코드 실행 시 ImageDataGenerator의 옵션은 따로 실험함



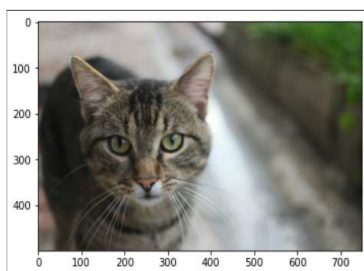
원본이미지(출처:나무위키)



rotation\_range=50



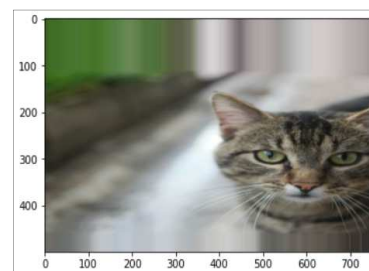
rotation\_range=-70



horizontal\_flip=True



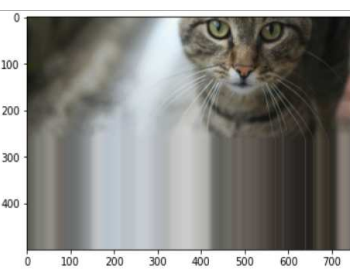
vertical\_flip=True



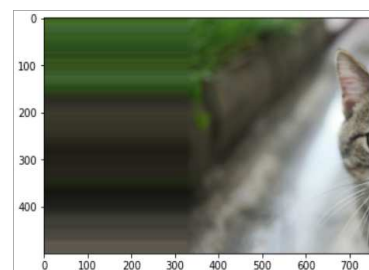
zoom\_range=[0.1, 1.5]



brightness\_range=[0.01, 2.0]



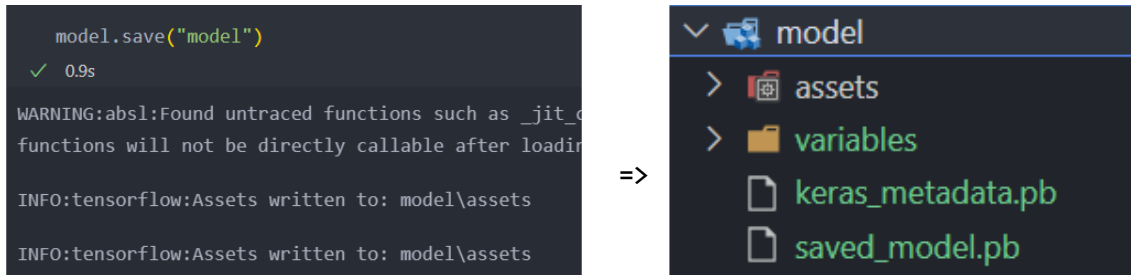
width\_shift\_range=0.8



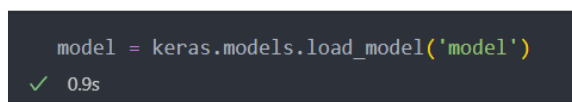
height\_shift\_range=0.9

#### 05. 케라스 라이브러리에서 가중치를 저장하려면 어떻게 해야하는가? 저장된 가중치를 불러와서 사용하려면 어떻게 해야하는가?

`model.save('model')` 코드를 활용하면 신경망 모델의 아키텍처, 학습된 모델의 가중치 값, 모델의 컴파일 정보, 옵티마이저와 현재 상태를 TensorFlow SavedModel 형식으로 저장한다.



`keras.models.load_model('model')` 코드를 활용하면 저장된 모델 및 가중치를 복원하여 재학습 및 예측 등의 작업을 수행할 수 있다.



#### 06. 피드 포워드 신경망과 순환신경망의 차이점은 무엇인가?

피드 포워드 신경망은 입력이 입력층에서 은닉층을 거쳐 출력층으로의 방향만 존재한다. 반면 순환신경망은 입력이 은닉층을 거쳐 다시 돌아오는 순환 구조를 가지는 신경망이다. 순환 신경망은 샘플을 처리하는 한 단계를 타임 스텝 개념을 도입하여 계산하고, 매 순간마다 노드에서 들어오고 나가는 순환 엣지가 존재한다.

#### 07. 장기 의존성이란 무엇인가?

장기 의존성은 입력과 출력의 거리가 멀어질수록 연관 관계가 적어져 먼 과거의 정보를 기억하지 못하는 문제이다. 이를 해결하기 위해서 LSTM, GRU와 알고리즘을 사용할 수 있다.

#### 08. 순환 신경망에서의 학습을 설명해보자.

순환신경망(RNN)은 순환 엣지를 사용하여 이전 은닉벡터의 출력값( $h_{t-1}$ )과 현재 입력( $x$ )에 각각 가중치합을 하고 활성화함수를 통과시켜 다음 은닉벡터( $h_t$ )에 전달하는 과정을 반복한다. 활성화함수로는 주로  $\tanh$ 가 사용된다. 순방향 계산을 통해 출력값을 계산한 다음, 시간에 따른 역전파 알고리즘을 도입해 반대방향으로 가중치를 업데이트한다. 이때 모든 시퀀스는 동일한 가중치를 공유하므로, 각 벡터에 대응되는 하나의 가중치 행렬에 대한 역전파 알고리즘이 수행된다.

#### 09. 순환 신경망은 주로 어디에 사용되는가?

순환 신경망은 음성 인식, 감정 분석, 언어 번역, 주가 예측 등에 사용될 수 있다. 구체적으로, 순환 신경망의 유형은 일대일, 일대다, 다대일, 다대다로 구분할 수 있다. 일대일은 단일 입력과 단일 출력이 있는 가장 일반적인 신경망으로 하나의 이미지가 들어왔을 때 객체를 탐지하는 데 사용할 수 있다. 일대다는 단일 입력과 다중 출력이 있는 신경망으로 하나의 이미지 입력에 대한 설명 캡션을 출력하는 이미지 캡셔닝에 사용된다. 다대일은 다중 입력에 대해 단일 출력을 생성하는 것으로, 감정 분석이나 스탬탐지, 주가 예측 등에 사용될 수 있다. 다대다는 다중 입력에 대한 다중 출력을 생성하는 것으로 기계 번역이나 문장의 다음 단어 예측 등에 사용될 수 있다.

#### 10. 왜 순환신경망에서는 그래디언트 소실이나 그래디언트 폭발과 같은 문제가 발생하기 쉬운가?

순환 신경망은 시퀀스 전체의 파라미터를 공유한다. 따라서 1보다 작은 가중치값이 여러번 곱해지게 되면 역전파 과정에서 그래디언트가 소실되는 문제가 발생하고, 1보다 큰 가중치값이 여러 번 곱해지면 그래디언트 폭발문제가 발생하게 된다.

#### 11. LSTM 셀은 어떻게 작동하는가?

LSTM은 장기 기억을 셀 상태에 저장하고, 입력/출력/삭제게이트를 이용해 정보의 흐름을 제어한다. 각 게이트에서는 주로 로지스틱 시그모이드 활성화함수를 사용해 0~1 사이의 출력값을 생성하고, 이는 게이트의 개폐 정도를 조절하게 된다. 입력게이트는 현시점의 정보가 얼마나 중요한지 반영하는 것이고, 망각게이트는 과거의 정보를 얼마나 잊을 지에 관한 것이며, 출력게이트는 출력할 양을 결정하는 것이다. 보다 구체적으로 LSTM에서 장기 기억은 셀 상태(cell state)에 저장되며, 업데이트 연산을 통해 셀 상태값이 계산된다. 먼저 셀 상태에 삭제게이트를 통해 들어온 값을 곱하여 일부 기억을 삭제한다. 여기에 입력 게이트를 통한 값을 더해줘 셀 상태에서 선택된 정보가 추가된다. 출력 게이트는 현재 시점의 입력값과 이전 시점의 은닉 상태를 통해 다음 시점의 은닉 상태에 전달할 값을 결정하게 된다.

#### 12. LSTM 셀은 어떻게 그래디언트 소실 문제를 해결하였는가?

LSTM에서 장기 기억은 셀 상태(cell state)에 저장된다. 저장된 정보는 입력/출력/삭제게이트를 통해 현재 시점까지 전달될 수 있으며, 따라서 그래디언트 정보가 중단되지 않으므로 그래디언트 소실 문제를 해결할 수 있다.