

ResNet 모델 전이 학습

2021. 05. 27

최원희

roses



dandelion



tulips



sunflowers



dandelion



roses



dandelion



roses



tulips



이미지 수집

데이터셋 만들기

```
from __future__ import absolute_import, division, print_function, unicode_literals

import os

import numpy as np

import matplotlib.pyplot as plt
```

```
try:
    # %tensorflow_version은 Colab에서만 지원됩니다.
    %tensorflow_version 2.x
except Exception:
    pass
import tensorflow as tf

keras = tf.keras
```

```
for image_batch, labels_batch in train_ds:
    print(image_batch.shape)
    print(labels_batch.shape)
    break
```

```
(32, 180, 180, 3)
(32,)
```




```
num_classes = 5
```

```
model = Sequential([
    layers.experimental.preprocessing.Rescaling(1./255, input_shape=(img_height, img_width, 3)),
    layers.Conv2D(16, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(32, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(64, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dense(num_classes)
])
```

모델 만들기

```
model.summary()
```



모델 요약

Model: "sequential"

Layer (type)	Output Shape	Param #
=====		
rescaling_1 (Rescaling)	(None, 180, 180, 3)	0

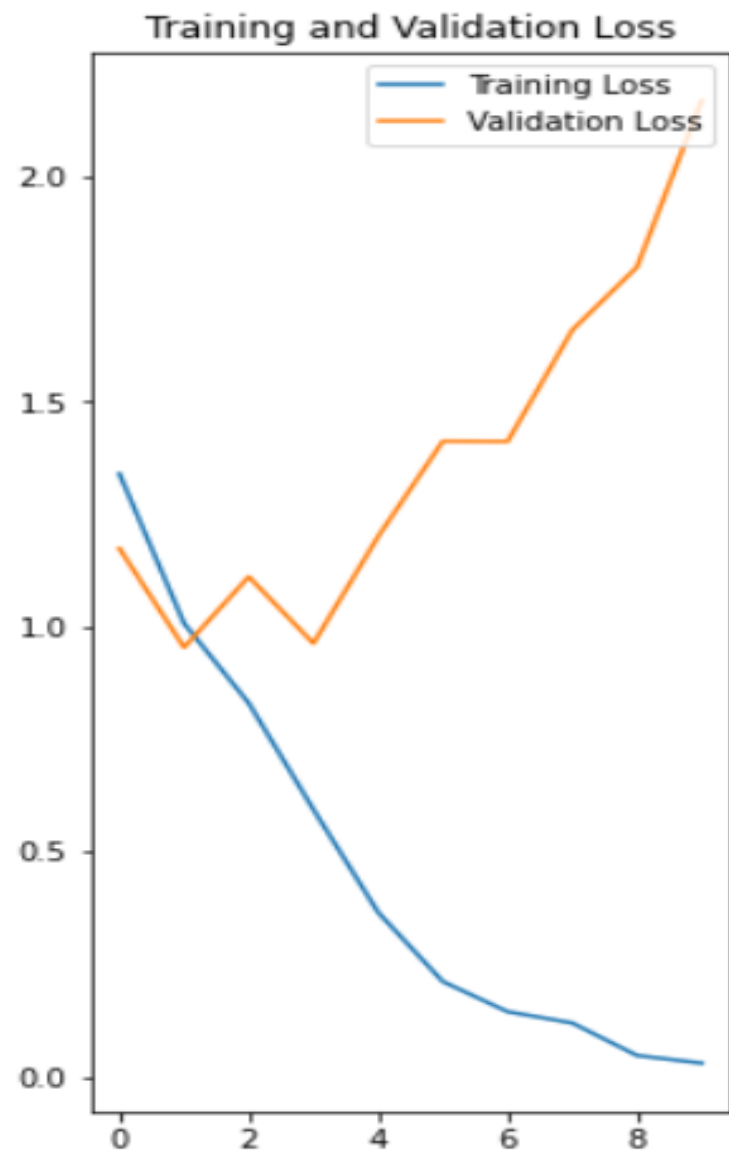
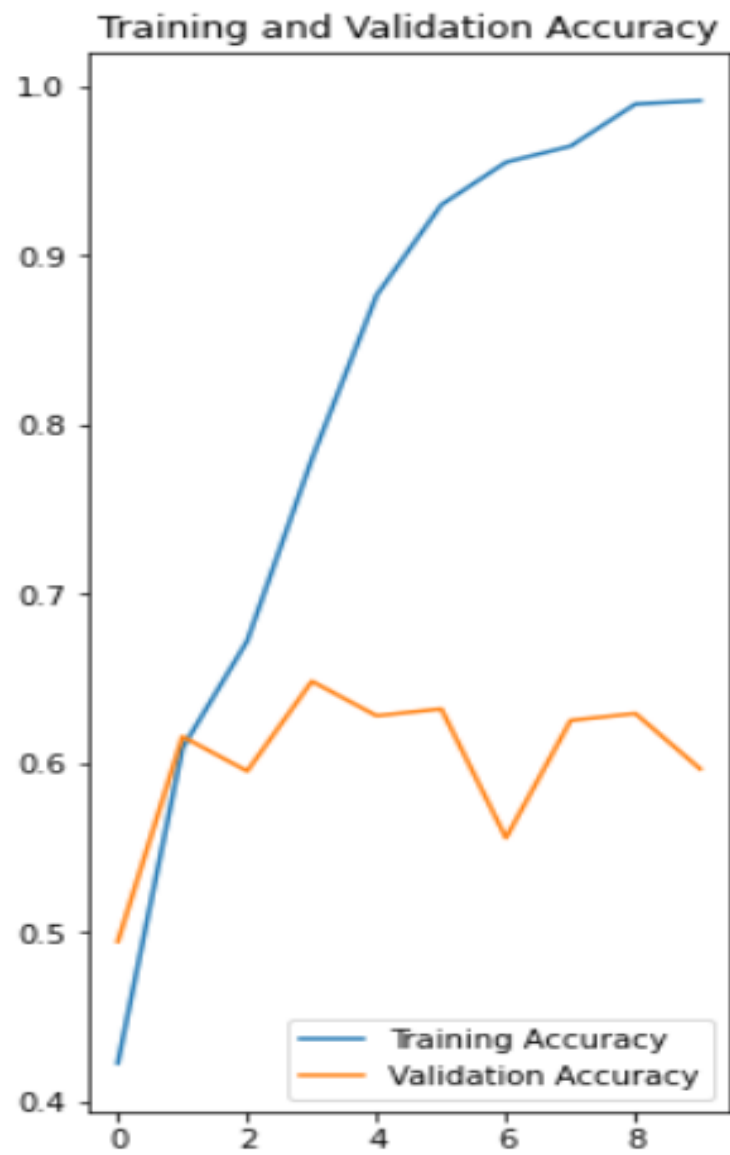
conv2d (Conv2D)	(None, 180, 180, 16)	448

max_pooling2d (MaxPooling2D)	(None, 90, 90, 16)	0

conv2d_1 (Conv2D)	(None, 90, 90, 32)	4640

max_pooling2d_1 (MaxPooling2D)	(None, 45, 45, 32)	0





훈련 결과 시각화

데이터 산출



```
model = Sequential([
    data_augmentation,
    layers.experimental.preprocessing.Rescaling(1./255),
    layers.Conv2D(16, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(32, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(64, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Dropout(0.2),
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dense(num_classes)
])
```

드롭아웃


```
sunflower_url = "https://storage.googleapis.com/download.tensorflow.org/example_images/592px-Red_sunflower"
sunflower_path = tf.keras.utils.get_file('Red_sunflower', origin=sunflower_url)

img = keras.preprocessing.image.load_img(
    sunflower_path, target_size=(img_height, img_width)
)
img_array = keras.preprocessing.image.img_to_array(img)
img_array = tf.expand_dims(img_array, 0) # Create a batch

predictions = model.predict(img_array)
score = tf.nn.softmax(predictions[0])

print(
    "This image most likely belongs to {} with a {:.2f} percent confidence."
    .format(class_names[np.argmax(score)], 100 * np.max(score))
)
```

예측하기

```
Downloading data from https://storage.googleapis.com/download.tensorflow.org/example_images/592px-Red_sunf
122880/117948 [=====] - 0s 0us/step
This image most likely belongs to sunflowers with a 98.55 percent confidence.
```