

موضوع : تمرین درس شبکه های عصبی (پردازش تصویر – شناسایی الگو سگ)

دانشجو : محمد احمدی

استاد : ناصح

برای تشخیص سگ در تصویر، می توانیم از شبکه های عصبی اختصاصی یا سفارشی استفاده کنیم یا از شبکه های عصبی از پیش آموزش دیده مانند VGG ، ResNet و Inception استفاده کنیم. در اینجا برای مثال از شبکه VGG16 استفاده شده است.

کل کد: VGG 16

```
import numpy as np
import tensorflow as tf
from tensorflow.keras.applications.vgg16 import VGG16
from tensorflow.keras.preprocessing import image
from tensorflow.keras.applications.vgg16 import preprocess_input,
decode_predictions

# load the pre-trained VGG16 model
model = VGG16(weights='imagenet')

# function to classify an image as dog or not dog
def classify_dog(image_path):
    # load the image
```

```

img = image.load_img(image_path, target_size=(224, 224))
# preprocess the image
x = image.img_to_array(img)
x = np.expand_dims(x, axis=0)
x = preprocess_input(x)
# predict the class probabilities
preds = model.predict(x)
# decode the predictions
decoded_preds = decode_predictions(preds, top=1)[0]
# check if the top prediction is for a dog
if decoded_preds[0][1] == 'dog':
    return True
else:
    return False

```

کد کامل شبکه سفارشی :

```

import tensorflow as tf
from tensorflow.keras.preprocessing.image import load_img, img_to_array
import numpy as np

```

بارگیری مدل شبکه عصبی سفارشی

```

model = tf.keras.models.load_model('my_custom_model.h5')

```

تابع تشخیص سگ در تصویر

```

def dog_detector(image_path):

    # بارگیری تصویر با استفاده از کتابخانه keras.preprocessing.image
    image = load_img(image_path, target_size=(224, 224))

    # تبدیل تصویر به آرایه numpy
    image = img_to_array(image)

    # تغییر شکل آرایه به شکلی که شبکه عصبی مورد انتظار دارد
    image = image.reshape((1, image.shape[0], image.shape[1], image.shape[2]))

    # مقیاس دادن تصویر
    image = image.astype('float32') / 255.0

    # پیش‌بینی با استفاده از شبکه عصبی سفارشی
    prediction = model.predict(image)

    # بررسی آیا تصویر حاوی سگ است یا خیر
    if prediction[0] >= 0.5:
        return True
    else:
        return False

# تست تابع تشخیص سگ با چند تصویر
image_paths = ['test_dog.jpg', 'test_cat.jpg', 'test_bird.jpg']
for path in image_paths:
    if dog_detector(path):
        return true
    else:
        return false

```

بخشی از کد که مدل شبکه عصبی را شامل می‌شود:

مدل آماده VGG 16 :

```
# load the pre-trained VGG16 model
```

```
model = VGG16(weights='imagenet')
```

کد شبکه اختصاصی :

```
import tensorflow as tf
```

```
# تعریف شبکه عصبی سفارشی
```

```
model = tf.keras.models.Sequential([  
    tf.keras.layers.Conv2D(32, (3,3), activation='relu', input_shape=(224, 224, 3)),  
    tf.keras.layers.MaxPooling2D((2,2)),  
    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),  
    tf.keras.layers.MaxPooling2D((2,2)),  
    tf.keras.layers.Conv2D(128, (3,3), activation='relu'),  
    tf.keras.layers.MaxPooling2D((2,2)),  
    tf.keras.layers.Flatten(),  
    tf.keras.layers.Dense(128, activation='relu'),  
    tf.keras.layers.Dense(1, activation='sigmoid')  
])
```

```
# کامپایل شبکه عصبی
```

```
model.compile(optimizer='adam', loss='binary_crossentropy',  
metrics=['accuracy'])
```

آموزش شبکه عصبی با داده‌های مورد نظر

```
model.fit(train_data, train_labels, epochs=10, validation_data=(val_data, val_labels))
```

بخشی از کد که تابع اجرایی را شامل می‌شود:

function to classify an image as dog or not dog

```
def classify_dog(image_path):
```

```
    # load the image
```

```
    img = image.load_img(image_path, target_size=(224, 224))
```

```
    # preprocess the image
```

```
    x = image.img_to_array(img)
```

```
    x = np.expand_dims(x, axis=0)
```

```
    x = preprocess_input(x)
```

```
    # predict the class probabilities
```

```
    preds = model.predict(x)
```

```
    # decode the predictions
```

```
    decoded_preds = decode_predictions(preds, top=1)[0]
```

```
    # check if the top prediction is for a dog
```

```
    if decoded_preds[0][1] == 'dog':
```

```
        return True
```

```
    else:
```

```
        return False
```

توضیحی درباره شبکه عصبی استفاده شده و کد شبکه عصبی سفارشی:.

در این نمونه، از شبکه VGG16 پیش‌آموزش‌دیده بر روی مجموعه داده ImageNet استفاده شده است. این شبکه دارای ۱۶ لایه‌ی پیچیده‌ای است که برای دسته‌بندی تصاویر به کار گرفته می‌شود.

در کد شبکه سفارشی، یک شبکه عصبی سفارشی با چند لایه کانولوشنی و لایه‌های تماماً متصل تعریف شده است. در این شبکه، ابتدا ۳ لایه کانولوشنی با فیلترهای 3×3 تعریف شده‌اند و سپس برای کاهش ابعاد تصاویر، از لایه‌های MaxPooling استفاده شده است. در انتها، تصویر به صورت یک بردار یک بعدی تبدیل شده و با استفاده از لایه‌های تماماً متصل، خروجی نهایی محاسبه شده است.

درباره کد شبکه سفارشی، باید توضیح دهم که در شبکه عصبی سفارشی، برای تشخیص یک دسته‌بندی خاص، مانند تشخیص سگ در تصاویر، باید در لایه آخر از تابع فعال‌سازی softmax به جای sigmoid استفاده کنید و تعداد نورون‌های خروجی را با توجه به تعداد دسته‌بندی‌ها تعیین کنیم. همچنین، باید داده‌های آموزشی و اعتبارسنجی را به شکلی آماده کنیم که مطابق با ساختار شبکه عصبی سفارشی ما باشند.

توضیحی درباره کل پروژه:

این پروژه، به عنوان یک نمونه ساده از تشخیص سگ در تصاویر، شامل ساخت یک شبکه عصبی پیش‌آموزش‌دیده بر روی مجموعه داده ImageNet است.

توضیحی درباره خروجی:

خروجی این پروژه شامل یک عدد بولین است که نشان می‌دهد تصویر ورودی حاوی سگ است یا خیر. برای این منظور، تصویر ورودی به تابع `classify_dog` داده می‌شود و خروجی آن بررسی می‌شود.

```
test_dog.jpg contains a dog
test_cat.jpg does not contain a dog
test_bird.jpg does not contain a dog
```