

1. 인공지능과 가위바위보 하기

○ 커널 연결됨 (Local)

1-1. 인공지능과
가위바위보 하기
30분

1-2. 데이터를
준비하자!
30분

1-3. 딥러닝 네트워크
설계하기
30분

1-4. 딥러닝 네트워크
학습시키기
30분

○ 1-5. 얼마나 잘
만들었는지 확인하기
30분

1-6. 더 좋은 네트워크
만들어 보기
30분

1-7. 프로젝트:
가위바위보 분류기
만들기
180분

«

1-5. 얼마나 잘 만들었는지 확인하기

테스트 데이터로 성능을 확인해 보자

사실 위의 인식 정확도는 학습용 데이터(`x_train`)을 가지고 구한 것입니다. 즉, 연습문제를 잘푸는 인공지능을 만든 거죠. 우리가 만든 딥러닝 네트워크는 실제 시험도 잘 볼 수 있을까요?

자 그러면 시험용 데이터(`x_test`)를 가지고 확인해 봅시다.

[Input]

```
test_loss, test_accuracy = model.evaluate(x_test_resaped,y_test, va
print("test_loss: {}".format(test_loss))
print("test_accuracy: {}".format(test_accuracy))
```



실행 ▶

[Output]

결과가 어떻게 나오나요? 99.57점을 받을 줄 알았는데, 98.85로 시험점수가 소폭 하락했네요. 역시 연습문제보다 실제 시험문제가 더 어려운가 봅니다. 위 MNIST 데이터셋 참고문헌을 보시면 학습용 데이터와 시험용 데이터의 손글씨 주인이 다른 것을 알 수 있습니다. 즉, 한 번도 본적이 없는 필체의 손글씨가 섞여 있을 가능성이 높습니다.



기존 데이터를 본 후 모델 실행하기. 그리고 모델 평가하기

`model.evaluate()` 대신 `model.predict()` 를 사용하면 model이 입력값을 보고 실제로 추론한 확률분포를 출력할 수 있습니다. 우리가 만든 model이란 사실 10개의 숫자 중 어느 것일지에 대한 확률값을 출력하는 함수입니다.

이 함수의 출력값 즉 확률값이 가장 높은 숫자가 바로 model이 추론한 숫자가 되는 거죠.

[Input]

```
predicted_result = model.predict(x_test_reshaped) # model이 추론한
predicted_labels = np.argmax(predicted_result, axis=1)

idx=3 #1번째 x_test를 살펴보자.
print('model.predict() 결과 : ', predicted_result[idx])
print('model이 추론한 가장 가능성이 높은 결과 : ', predicted_labels[idx])
print('실제 데이터의 라벨 : ', y_test[idx])
```

실행 ▶

[Output]

`model.predict()` 결과가 [9.5208375e-15 2.8931768e-11 1.2696462e-09
2.0265421e-08 6.1321614e-11 2.9599554e-12 1.0710074e-15
1.0000000e+00 1.0549885e-11 3.8589491e-08] 의 벡터 형태로 나왔나요?

이 벡터는 model이 추론한 결과가 각각 0, 1, 2, ..., 7, 8, 9일 확률을 의미합니다.

이 경우라면 model이 추론한 결과가 7일 확률이 1.00에 근접하고 있다, 즉 이 model은 입력한 이미지가 숫자 7이라는 걸 아주 확신하고 있다는 뜻이 됩니다.



정말 숫자 7인가요?

[Input]

```
plt.imshow(x_test[idx], cmap=plt.cm.binary)
plt.show()

print(y_test[idx])
```



실행 ▶

[Output]

그렇다면 model이 추론해 낸 숫자와 실제 라벨의 값이 다른 경우는 어떤 경우인지 직접 확인해 볼 수도 있겠습니다.

[Input]

```
import random
wrong_predict_list=[]
for i, _ in enumerate(predicted_labels):
    # i번째 test_labels과 y_test이 다른 경우만 모아 봅시다.
    if predicted_labels[i] != y_test[i]:
        wrong_predict_list.append(i)

# wrong_predict_list 에서 랜덤하게 5개만 뽑아봅시다.
samples = random.choices(population=wrong_predict_list, k=5)

for n in samples:
    print("예측확률분포: " + str(predicted_result[n]))
    print("라벨: " + str(y_test[n]) + ", 예측결과: " + str(predicted_labels[n]))
    plt.imshow(x_test[n], cmap=plt.cm.binary)
    plt.show()
```



실행 ▶



AIFFEL | 대표 김승일 | 주소 서울특별시 강남구 역삼로 156(역삼동) 4층, 모두의연구소

전화 070-7743-5882 | 이메일 support@aiffel.io | 개인정보보호책임자 이지석

© 2020 모두의연구소

