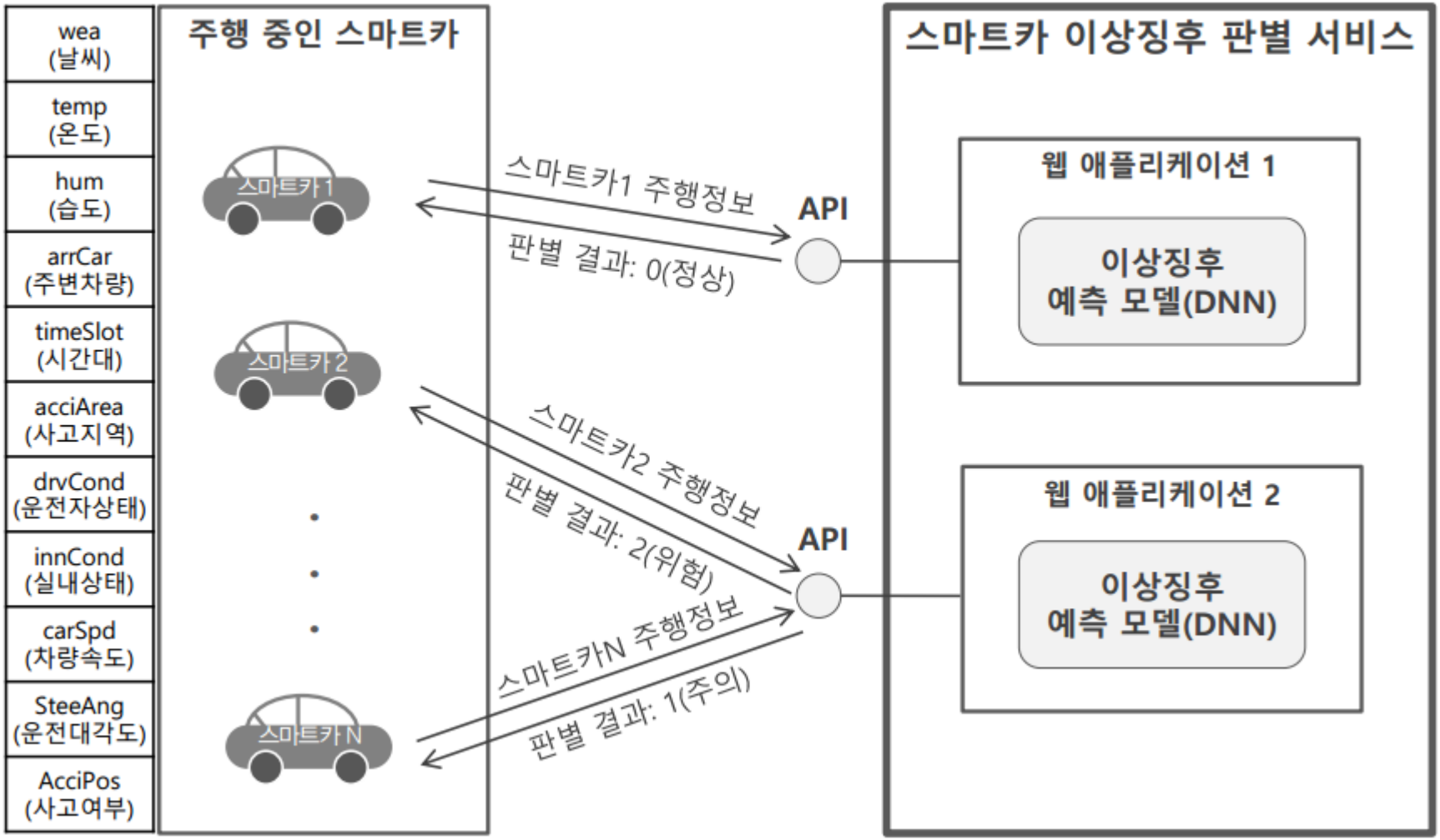
8.3 텐서플로 신경망 분석 – 스마트카 이상 징후

AI기반 애플리케이션 구축 (1/10)



8.3 텐서플로 신경망 분석 – 스마트카 이상 징후

AI기반 애플리케이션 구축 (2/10)

01. 먼저 웹 애플리케이션 환경을 구성하기 위해 플라스크(Flask)를 설치한다. 플라스크는 파이썬 기반의 웹서버 엔진으로 파이썬 환경에서 동작한다. 앞서 구성했던 파이썬 3.5 환경(py35)을 활성화하기 위해 [시작] → [모든 프로그램] → [Anaconda3 (64-bit)] → [Anaconda Prompt]를 차례로 선택한 후 다음 명령을 실행한다.

```
(base) C:\Users\[사용자명]> activate py35
```

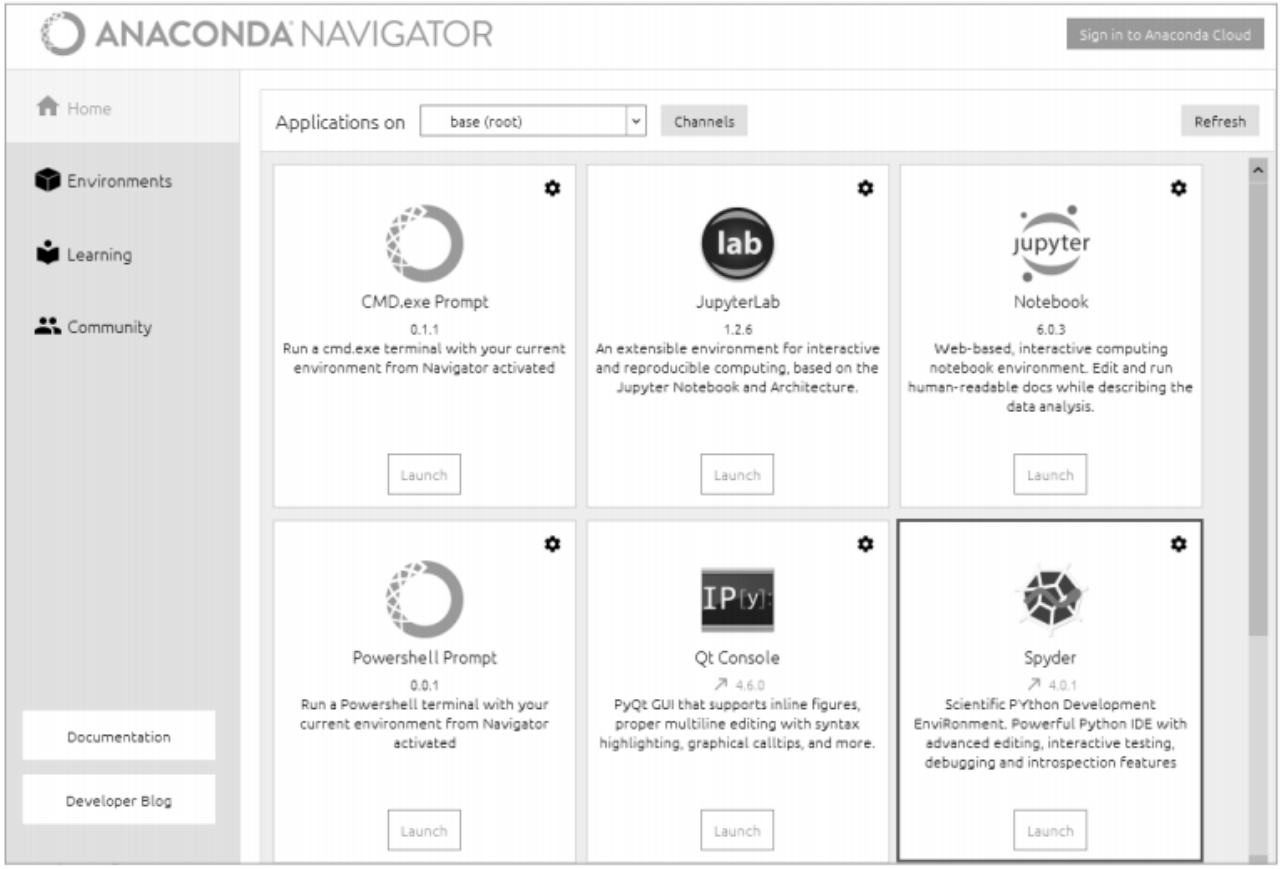
02. 파이썬 3.5 환경에서 pip 명령을 통해 플라스크를 설치한다.

```
(py35) C:\Users\[사용자명]> pip install flask
```

8.3 텐서플로 신경망 분석 – 스마트카 이상 징후

AI기반 애플리케이션 구축 (3/10)

03. 플라스크 설치가 완료되면 웹 애플리케이션 프로그램을 작성한다. [시작] → [모든 프로그램] → [Anaconda3 (64-bit)] → [Anaconda Navigator(Anaconda3)]를 실행한다. 아나콘다 내비게이터가 활성화되면 애플리케이션 대시보드에서 Spyder의 [Launch] 버튼을 클릭해 파이썬 개발 도구인 스파이더(Spyder)를 실행한다.



8.3 텐서플로 신경망 분석 – 스마트카 이상 징후

AI기반 애플리케이션 구축 (4/10)

04. 스파이더가 실행되면 에디터 창에 플라스크의 REST API로 사용할 소스코드를 그림 8.33처럼 작성한다. 관련 소스 코드는 "C://예제소스/bigdata2nd-master/CH08/smartcar_dnn_service.py"로 제공되므로 참고한다.

smartcar_dnn_service.py

```
3   import flask
4   import pandas as pd
5   import numpy as np
6   import tensorflow as tf
7
8   from flask import Flask, jsonify, request
9   from keras.models import load_model
10
11
12  app = Flask(__name__)
13
14  global graph
15  graph = tf.get_default_graph()
16  model = load_model('C:/models/smartcar_dnn_model.h5')
17
18
19  @app.route('/smartcar/predict',  methods=["GET","POST"])
20  def predict():
21      data = {"success": False}
22
```

```
23      params = flask.request.json
24      if (params == None):
25          params = flask.request.args
26
27      # if parameters are found, return a prediction
28      if (params != None):
29          x=pd.DataFrame.from_dict(params, orient='index').transpose()
30
31          with graph.as_default():
32
33              data["prediction"] = str(model.predict(x).argmax())
34              data["success"] = True
35
36      # return a response in json format
37      return flask.jsonify(data)
38
39  if __name__ == '__main__':
40      app.run(host='0.0.0.0', port=9001)
```

그림 8.33 플라스크 REST API 소스코드 – 스마트카 이상징후 판별 서비스

위 소스코드의 주요 내용은 다음과 같다.

- 16번째 줄: 저장했던 DNN 모델 로드
- 19번째 줄: API URL 및 HTTP 전송 방식(GET, POST) 설정
- 25번째 줄: 스마트카에서 전송한 상태 데이터 수신
- 33번째 줄: 모델을 이용해 스마트카 이상징후 예측
- 37번째 줄: 예측 결과를 해당 스마트카로 전달

8.3 텐서플로 신경망 분석 – 스마트카 이상 징후

AI기반 애플리케이션 구축 (6/10)

05. 파일럿 PC의 C 드라이브에 "C://flask" 폴더를 생성하고, flask 하위 폴더로 "app" 폴더를 생성한다.

```
C://flask/app/
```

06. 스파이더 에디터에서 [File] → [Save] 메뉴를 선택하고 앞에서 작성한 플라스크 REST API의 소스코드를 "C://flask/app/" 경로에 "smartcar_dnn_service.py"라는 이름으로 저장한다.

07. 이제 스마트카 이상징후 판별 서비스를 위한 플라스크 서비스 서버를 실행한다. [시작] → [모든 프로그램] → [Anaconda3 (64-bit)] → [Anaconda Prompt (Anaconda3)]를 차례로 선택해 py35(파이썬 3.5) 환경을 활성화하고, 앞에서 저장한 "smartcar_dnn_service.py" 프로그램을 실행한다.

```
(base) C:\Users\[사용자명]> activate py35
(py35) C:\Users\[사용자명]> python C://flask/app/smartcar_dnn_service.py
```

8.3 텐서플로 신경망 분석 – 스마트카 이상 징후

AI기반 애플리케이션 구축 (7/10)

08. 플라스크의 REST API 서비스가 정상적으로 실행되면 다음과 같은 메시지가 출력된다.

"Running on http://0.0.0.0:9001/"

```
(py35) C:₩Users₩Administrator>python C://flask/app/smartcar_dnn_service.py
Using TensorFlow backend.
2020-04-21 13:39:28.884657: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that thi
s TensorFlow binary was not compiled to use: AVX AVX2
 * Serving Flask app "smartcar_dnn_service" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://0.0.0.0:9001/ (Press CTRL+C to quit)
```

8.3 텐서플로 신경망 분석 – 스마트카 이상 징후

AI기반 애플리케이션 구축 (8/10)

09. 주행 중인 스마트카의 10개 상태 값(0~2)을 가정해 정의한다.

- 주행 중인 스마트카의 상태값 정의: wea(날씨)=2, temp(온도)=0, hum(습도)=1, arrOthCar(주변차량)=2, time(시간대)=0, acciArea(사고다발지역)=1, drvCond(운전자상태)=1, innCond(실내상태)=2, carSpd(차량속도)=2, steeAng(운전대각도)=0

10. 크롬 브라우저를 실행해 플라스크에서 실행 중인 스마트카 이상징후 판별 REST API 서비스를 호출한다. 주행 중인 스마트카의 상태값 10개를 정의한 전체 URL은 다음과 같다.

- REST API 서비스 URL: http://127.0.0.1:9001/smartcar/predict?wea=2&temp=0&hum=1&arrOthCar=2&time=0&acciArea=1&drvCond=1&innCond=2&carSpd=2&steeAng=0

위 REST API URL을 크롬 브라우저에 입력해 스마트카 이상징후 판별 서비스 API를 호출한다.

8.3 텐서플로 신경망 분석 – 스마트카 이상 징후

AI기반 애플리케이션 구축 (9/10)

11. API 호출 결과가 그림 8.35처럼 출력된다. “prediction:”은 이상징후 판별 결과이고, “success:”는 API 처리 결과다.

8.3 텐서플로 신경망 분석 – 스마트카 이상 징후

AI기반 애플리케이션 구축 (10/10)

실습