فصل ۲۱- ذخیره و بارگذاری مدل‌های آموزش‌دیده

۲۱.۰ مقدمه

در ۲۰ فصل گذشته و حدود ۲۰۰ دستور العمل، نحوه برداشت داده‌های خام و استفاده از یادگیری ماشینی برای ایجاد مدل‌های پیش‌بینی با عملکرد خوب را پوشش داده‌ایم. با این حال، برای اینکه همه‌ی کار ما ارزشمند باشد، در نهایت باید کاری را با مدل خود انجام دهیم، مانند ادغام آن با یک برنامه نرم افزاری موجود. برای دستیابی به این هدف، باید بتوانیم هم مدل‌های خود را پس از آموزش ذخیره کنیم و هم در صورت نیاز در یک برنامه، آنها را بارگذاری کنیم. تمرکز فصل آخر ما نیز روی همین دو مورد است.

۲۱.۱ ذخیره و بارگذاری یک مدل Sicit-Learn

مسئله

شما یک مدل آموزش‌دیده‌ای دارید که می‌خواهید آن را ذخیره کنید و در جای دیگری بارگذاری کنید.

راه‌حل

مدل را به عنوان فایل pickle ذخیره کنید:

```
# Load libraries
from sklearn.ensemble import RandomForestClassifier
from sklearn import datasets
from sklearn.externals import joblib


# Load data
iris = datasets.load_iris()
features = iris.data
target = iris.target


# Create decision tree classifer object
classifer = RandomForestClassifier()


# Train model
model = classifer.fit(features, target)


# Save model as pickle file
joblib.dump(model, "model.pkl")
```

```
['model.pkl']
```

هنگامی‌که مدل ذخیره شد، می‌توانیم از scikit-learn در برنامه مقصد خود (به عنوان مثال یک برنامه وب) بـرای بارگـذاری مدل استفاده کنیم:

```
# Load model from file
classifer = joblib.load("model.pkl")
```

و از آن برای پیش‌بینی استفاده کنیم:

```
# Create new observation
new_observation = [[ 5.2, 3.2, 1.1, 0.1]]

# Predict observation's class
classifer.predict(new_observation)
```

```
array([0])
```

بحث

اولین قدم در استفاده از یک مدل در تولید یک نرم‌افزار، ذخیره آن مدل به عنوان فایلی است که می‌تواند توسط یک برنامه یا در تعامل با برنامه‌های دیگری بارگذاری و استفاده شود. ما می‌توانیم این کار را با ذخیره مدل به عنوان یک فایل pickle، که یک فرمت داده‌ای خاص پایتون است انجام دهیم. به طور خاص، برای ذخیـره یـک مـدل، از joblib اسـتفاده مـی‌کنیم کـه کتابخانه‌ای است برای مواردی که آرایه‌های NumPy بزرگ داریم. این یک اتفاق رایج برای مدل‌های آموزش‌دیده در scikit-learn است.

هنگام ذخیره مدل‌های scikit-learn، توجه داشته باشید که مدل‌های ذخیره شـده ممکـن اسـت بـا نسـخه‌های متفـاوت از scikit-learn سازگار نباشند. بنابراین، گنجاندن نسخه scikit-learn مورد استفاده در نام مدل می‌تواند مفید باشد:

```
# Import library
import sklearn

# Get scikit-learn version
scikit_version = joblib.__version__

# Save model as pickle file
joblib.dump(model, "model_{version}.pkl".format(version=scikit_version))
```

```
['model_0.11.pkl']
```

۲۱.۲ ذخیره و بارگذاری یک مدل Keras

مسئله

شما یک مدل Keras آموزش‌دیده دارید و می‌خواهید آن را ذخیره کنید و در جای دیگری بارگذاری کنید.

راه‌حل

مدل را به عنوان HDF5 ذخیره کنید:

```
# Load libraries
import numpy as np
from keras.datasets import imdb
from keras.preprocessing.text import Tokenizer
from keras import models
from keras import layers
from keras.models import load_model

# Set random seed
np.random.seed(0)

# Set the number of features we want
number_of_features = 1000

# Load data and target vector from movie review data
(train_data, train_target), (test_data, test_target) = imdb.load_data(
num_words=number_of_features)

# Convert movie review data to a one-hot encoded feature
matrix
tokenizer = Tokenizer(num_words=number_of_features)
train_features = tokenizer.sequences_to_matrix(train_data,
mode="binary")
test_features = tokenizer.sequences_to_matrix(test_data, mode="binary")

# Start neural network
network = models.Sequential()

# Add fully connected layer with a ReLU activation function
network.add(layers.Dense(units=16, activation="relu",
input_shape=(number_of_features,)))

# Add fully connected layer with a sigmoid activation
function
network.add(layers.Dense(units=1, activation="sigmoid"))

# Compile neural network
network.compile(loss="binary_crossentropy", # Cross-entropy
optimizer="rmsprop", # Root Mean Square Propagation
metrics=["accuracy"]) # Accuracy performance metric

# Train neural network
history = network.fit(train_features, # Features
train_target, # Target vector
epochs=3, # Number of epochs
verbose=0, # No output
batch_size=100, # Number of observations per batch
validation_data=(test_features, test_target)) # Test data
```

```
Using TensorFlow backend.
```

سپس می‌توانیم مدل را در برنامه دیگری یا برای آموزش اضافی بارگذاری کنیم:

```
# Load neural network
network = load_model("model.h5")
```

بحث

برخلاف scikit-learn، Keras توصیه نمی‌کند مدل‌ها را با استفاده از pickle ذخیره کنید. در عوض، مدل‌ها به عنوان یک فایل HDF5 ذخیره می‌شوند. فایل HDF5 حاوی همه چیزهایی است که نه تنها برای بارگذاری مدل برای پیش‌بینی (یعنی معماری و پارامترهای آموزش‌دیده)، بلکه برای راه‌اندازی مجدد آموزش (یعنی تنظیمات از دست دادن و بهینه‌ساز و وضعیت فعلی) نیاز دارید.