

Сравнение эффективности различных классификаторов при распознавании эмоций по речи

Ян Цзяфэн Р33212

Нгуен Чан Минь Р33211

Цель

Исследование преимуществ различных классификаторов на основе разных наборов данных при распознавании эмоций по речи.

Задачи

1. Собрать классические наборы данных, содержащих эмоциональную речь.
2. Обработка и анализ аудиофайлов для извлечения признаки.
3. Реализовать различные классификаторы временных рядов и обучить модели.
4. Тестировать модели.
5. Сравнить и анализировать эффективность каждого классификатора.

Аннотация

В настоящее время развитие распознавания речи можно считать очень зрелым, но это пока далеко от нашей цели - естественного взаимодействия человека и компьютера. Одна из причин этого заключается в том, что машины все еще не могут понимать наши эмоции, когда мы говорим. И это важная мотивация для исследования распознавания речевых эмоций. В данном проекте в основном сравниваются эффективности различных классификаторов при распознавании эмоций по речи.

Задача 1: сбор данных

[RAVDESS](#) (The Ryerson Audio-Visual Database of Emotional Speech and Song)

Английский, около 1500 аудиозаписей от 24 человек (12 мужчин и 12 женщин), включая 8 различных эмоций (третье число в имени файла представляет эмоциональный тип): 01 = нейтральный, 02 = спокойный, 03 = счастливый, 04 = грустный, 05 = злой, 06 = испуганный, 07 = отвращение, 08 = удивленный.

Задача 2: извлечения признаков

- Используем библиотеку обработки звука Librosa для извлечения признаков.

```
feature = np.mean(librosa.feature.mfcc(y=X, sr=sample_rate, n_mfcc=40).T, axis=0)
```

```
feature=np.mean(librosa.feature.chroma_stft(S=np.abs(librosa.stft(X)),sr=sample_rate).T,axis=0)
```

```
feature = np.mean(librosa.feature.melspectrogram(X, sr=sample_rate).T, axis=0)
```

- Используем инструмент openSMILE

(IS09_emotion)(всего 384 признаков)

Команда cmd: SMILExtract_Release -C config_file -I input_file -O output_file

где, config_file: путь к выбранному файлу набора признаков opensmile,

input_file: путь к входному аудиофайлу,

output_file: путь к файлу результата вывода извлечения признаков

Задача 3: Реализовать классификаторы и обучить модели

В этом проекте для обучения и получения моделей используются классификаторы SVM, MLP и CNN.

SVM

openSmile

```
from sklearn.svm import SVC
import numpy as np
X = np.asarray(X_train_IS09)
Y = np.asarray(Y_train_IS09)
clf_svc = SVC(decision_function_shape='ovo', kernel='rbf', C=200,
               gamma=0.00000001, probability=True, degree=0.1)
clf_svc.fit(X, Y)
print(clf_svc.score(X, Y))
```

```
from sklearn.metrics import accuracy_score
y_hat = clf_svc.predict(X_test_IS09)
accuracy_score(y_hat, Y_test_IS09)
```

0.6507936507936508

0.4027777777777778

Librosa

```
from sklearn import svm
clf = svm.SVC(C=9, gamma=0.001, decision_function_shape='ovo')
clf.fit(X_train, y_train)
print(clf.score(X_train, y_train))
print(clf.score(X_test, y_test))
```

0.9989626556016598

0.5798319327731093

MLP

openSmile

```
from sklearn.neural_network import MLPClassifier
mlp_classifier = MLPClassifier(alpha=0.912, batch_size=32, learning_rate='adaptive', max_iter=500)
mlp_classifier.fit(X, Y)
print(mlp_classifier.score(X, Y))
y_hat = mlp_classifier.predict(X_test_IS09)
accuracy_score(y_hat, Y_test_IS09)
```

0.7073412698412699

0.41203703703703703

Librosa

```
from sklearn.neural_network import MLPClassifier
mlp_classifier = MLPClassifier(alpha=0.01, batch_size=32, hidden_layer_sizes=(64,32,32,16,8), learning_rate='adaptive', max_iter=200)
```

```
mlp_classifier.fit(X_train, y_train)
```

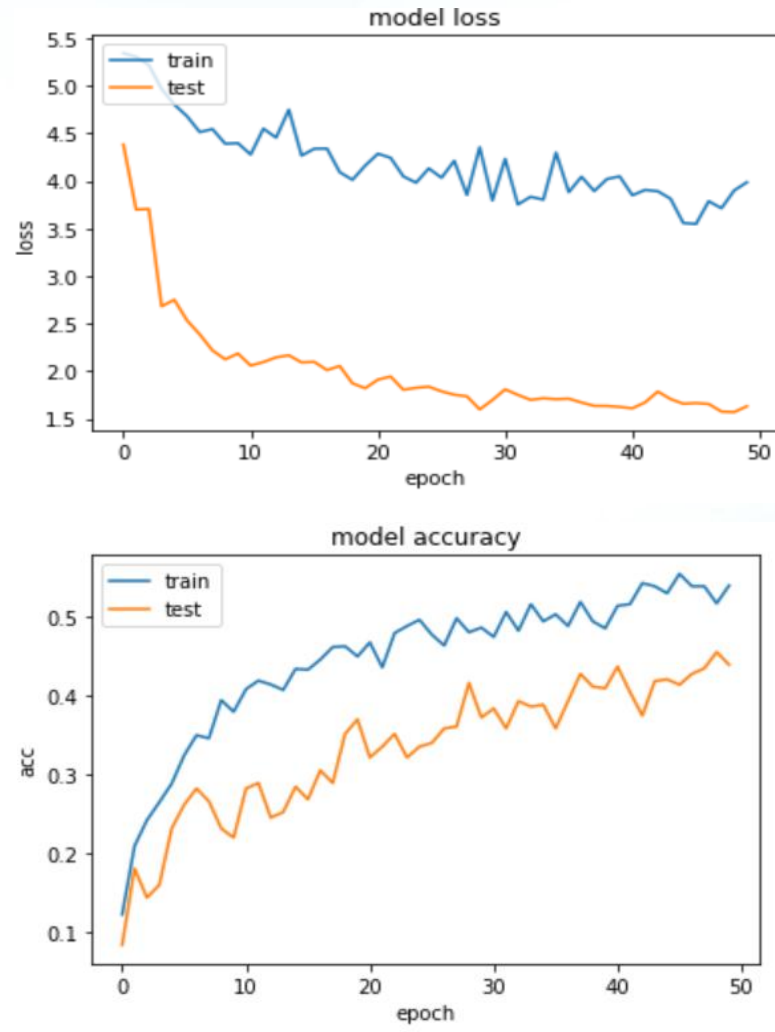
```
MLPClassifier(activation='relu', alpha=0.01, batch_size=32, beta_1=0.9,
              beta_2=0.999, early_stopping=False, epsilon=1e-08,
              hidden_layer_sizes=(64, 32, 32, 16, 8), learning_rate='adaptive',
              learning_rate_init=0.001, max_iter=200, momentum=0.9,
              n_iter_no_change=10, nesterovs_momentum=True, power_t=0.5,
              random_state=None, shuffle=True, solver='adam', tol=0.0001,
              validation_fraction=0.1, verbose=False, warm_start=False)
```

```
from sklearn.metrics import classification_report
predictions = mlp_classifier.predict(X_test)
print(classification_report(y_test, predictions))
```

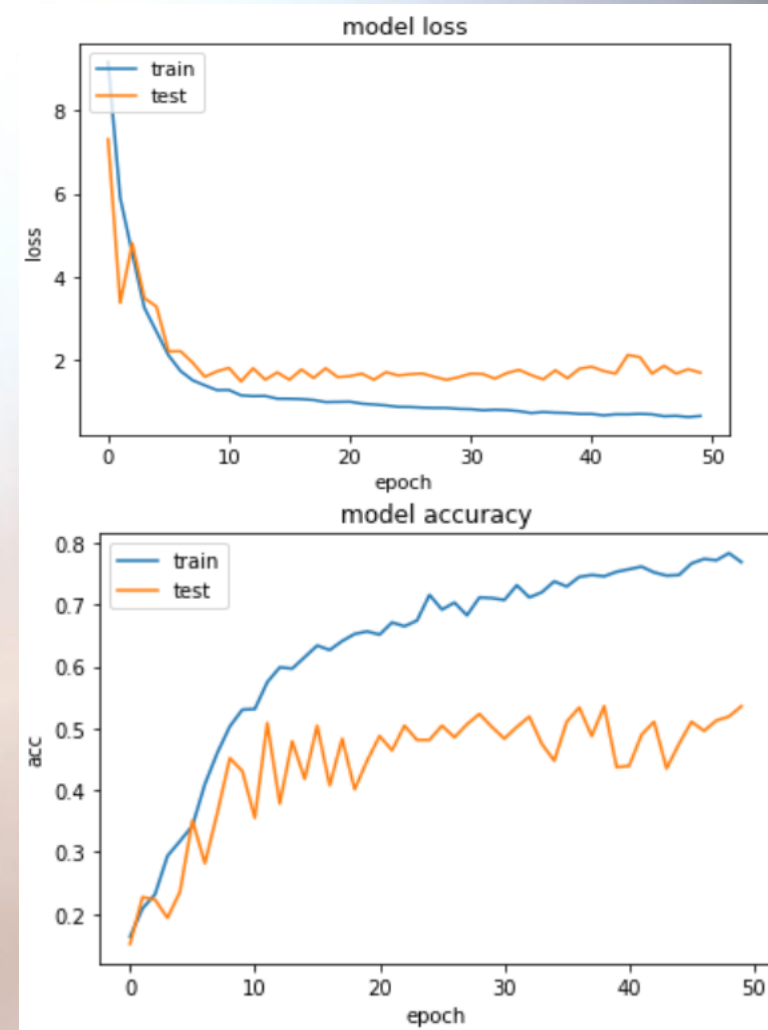
	precision	recall	f1-score	support
0	0.00	0.00	0.00	28
1	0.45	0.58	0.51	72
2	0.35	0.31	0.33	58
3	0.29	0.43	0.35	60
4	0.52	0.58	0.55	60
5	0.41	0.60	0.49	55
6	0.47	0.20	0.28	74
7	0.42	0.38	0.40	69
micro avg	0.41	0.41	0.41	476
macro avg	0.36	0.39	0.36	476
weighted avg	0.39	0.41	0.39	476

CNN

openSmile



Librosa



Выводы

- openSmile vs Librosa:

В этом проекте openSmile может извлекать несколько типов наборов признаков, но извлечение признаков занимает больше времени, а точность полученной модели ниже. Librosa извлекает один тип набора функций, время извлечения короче, а полученная модель более точна.

- SVM vs MLP vs CNN:

Время обучения модели CNN больше, но точность лучше.

Варианты дальнейших исследований

- Точность полученной модели все еще слишком низкая, и для получения высокоточной модели необходимо постоянно корректировать параметры классификатора.
- Слишком мало сравнительных образцов приводит к недостаточной уверенности в выводе. Необходимо использовать несколько баз данных для повышения достоверности выводов.
- Попробовать использовать для сравнения несколько типов наборов признаков Librosa и другие наборы признаков openSmile.