Dokumentace k projektu z předmětu SUR

Jan Kastner

Květen 2025

1 Popis řešení a implementace

Tento projekt se zaměřuje na klasifikaci objektů do 31 tříd na základě dvou typů příznaků nahrávek řeči a obrázků daného člověka. Cílem projektu bylo sestavit klasifikátor, který i přes malé množství dat bude poskytovat uspokojivé výsledky. V rámci projektu byly implementovány tři typy klasifikátorů. Konvoluční neuronová síť (CNN), která je spolehlivým modelem, ale není tolik vhodná, pokud je datová sada velmi malá. Support Vector Machine (SVM), neboli metoda podpůrných vektorů, která byla využita pro klasifikaci na základě hlasových nahrávek a dosáhla poměrně dobrých výsledků a kombinovaný klasifikátor, který spojuje výsledky CNN a SVM, a který ze všech tří modelů dosáhl na validační sadě nejlepších výsledků.

1.1 Konvoluční neuronová síť

Jednotlivé obrázky ze souboru byly převedeny na šedotónové a jejich velikost zůstala zachována. V rámci projektu bylo experimentováno s podvzorkováním i se zachováním barevnosti, ale nejlepších výsledků bylo dosaženo při zachování původní velikosti a použití obrázků v odstínech šedé. Natrénování CNN však vyžaduje větší množství dat, aby nedošlo k přetrénování sítě na trénovací sadě. Proto bylo nutné provést augmentaci dat za účelem vytvoření rozsáhlejší datové sady. Náhodně byly aplikovány různé rotace, přidání šumu na úrovních 0.1, 0.2 až 0.5, změny v sytosti a jasu obrázků a také rozmazání. Pro každý prvek z původní datové sady bylo vytvořeno pět nových, augmentovaných vzorků. Pro lepší výsledky byla data dále normalizována tak, aby všechny hodnoty byly v rozmezí $\langle 0, 1 \rangle$. Při tréninku CNN docházelo k mizení gradientu, a tento proces byl tedy podpořen optimalizačním algoritmem Adam, založeným na momentu $m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ pro udržování směru gradientu a energii gradientu $v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ pro efektivní trénování sítě ve všech jejích částech. Adam byl použit se standardním nastavením parametrů, tedy s učící rychlostí i L_2 regularizačním parametrem nastaveným na jednu tisícinu. Po každé konvoluční vrstvě byla také implementována batch normalizace, která umožnila efektivnější trénování. Aby nedocházelo k přetrénování sítě, byl za poslední konvoluční vrstvou použit dropout. Experimenty ukázaly, že nejlepších výsledků bylo dosaženo při velmi vysoké hodnotě dropout, konkrétně 0,75, tedy 75% aktivací bylo vynulováno. Experimentálně bylo ověřeno, že nejlepších výsledků bylo dosaženo při velikosti batch 32 a počtu epoch 15. Jako ztrátová funkce byla použita cross-entropy. I přes veškeré použité metody se na takto malém množství dat

nepodařilo model dobře natrénovat a na validační sadě bylo dosaženo úspěšnosti mezi 32% a 45%.

1.2 Metoda podpůrných vektorů

Z jednotlivých načtených zvukových nahrávek byly extrahovány Mel-frekvenční cepstrální koeficienty (MFCC) za použití funkce `mfcc` z knihovny `python_speech_features`. Tato funkce přijímá několik parametrů, mezi které patří počet extrahovaných cepstrálních koeficientů, délka okna, překryv mezi okny, délka Fourierovy transformace a počet trojúhelníkových filtrů, na které je spektrum rozděleno. Experimentálně bylo ověřeno, že nejlepší výsledky model dosahuje při následujících parametrech: délka Fourierovy transformace 1024, počet filtrů 36, počet cepstrálních koeficientů 36, délka okna 25 ms a překryv mezi okny 10 ms. Z jednotlivých časových okamžiků byla následně extrahována střední hodnota cepstrálních koeficientů, směrodatná odchylka, šikmost a špičatost. Tyto hodnoty byly spojeny do jednoho vektoru a použity pro trénování SVM. Oproti trénování pouze na střední hodnotě bylo dosaženo dramatického zlepšení z 50 % na 85 % přesnosti na validační sadě. U nahrávek bylo vyzkoušeno odstranění prvních x milisekund kvůli "cvaknutí", které se na začátku objevovalo, ale to vždy vedlo k dalšímu zhoršení výsledků. Experimentálně bylo zjištěno, že nejlepších výsledků bylo dosaženo s radiálním jádrem.

1.3 Kombinovaný model

Model využívá výstupní soubory obou předchozích modelů, konkrétně 31 logaritmických pravděpodobností. Hodnoty v jednotlivých sloupcích jsou normalizovány pomocí střední hodnoty a směrodatné odchylky, tak aby měly střední hodnotu 0 a směrodatnou odchylku 1. Následně byly sečteny hodnoty obou modelů, přičemž byla zohledněna kvalita klasifikace, přičemž výstupy konvoluční sítě mají v součtu menší váhu než výstupy SVM. Tento model dosáhl lepších výsledků než předchozí modely a na trénovací sadě dosáhl úspěšnosti 92%.

2 Zhodnocení výsledků

Ačkoli je konvoluční neuronová síť dobrým nástrojem, pro její použití byla k dispozici příliš malá datová sada. Přes všechny použité techniky bylo dosaženo maximálně 45% úspěšnosti na validační sadě. Naopak metoda SVM dosáhla výrazně lepších výsledků pro klasifikaci na základě MFCC. Na výsledky klasifikace mělo v tomto případě velký vliv použití i jiných charakteristik než střední hodnoty (standardní odchylka, šikmost a spíčatost) v přiznacích a bylo dosaženo 85% úspěšnosti na validační sadě. Hibridní model využil přínosů obou modelů a přesnost tohoto modelu na validační sadě byla 92%. Pravděpodobně, kdyby byla zvolena technika GMM nebo SVM i pro obrázky, byly by výsledky modelu ještě lepší. Hlavním důvodem je malé množství dat.

3 Návod ke spuštění a struktura souborů

Struktura projektu

```
xkastn02.zip/
 dokumentace.pdf
 SRC/
    audio_SVM.py
    image_CNN.py
    hybrid_classifier.py
 audio_SVM_results.csv
 image_CNN_results.csv
 hybrid_classifier_results.csv
```

Požadavky

Pro spuštění systému je potřeba mít nainstalovaný Python verze 3.10.13 nebo novější. Dále je nutné doinstalovat následující knihovny:

```
pip install numpy scipy scikit-learn python_speech_features torch pillow
```

Spuštění

Všechny skripty se spouštějí stejně:

```
python <script>.py -t <trénovací_data> -v <validační_data> -o <výstup.csv>
```