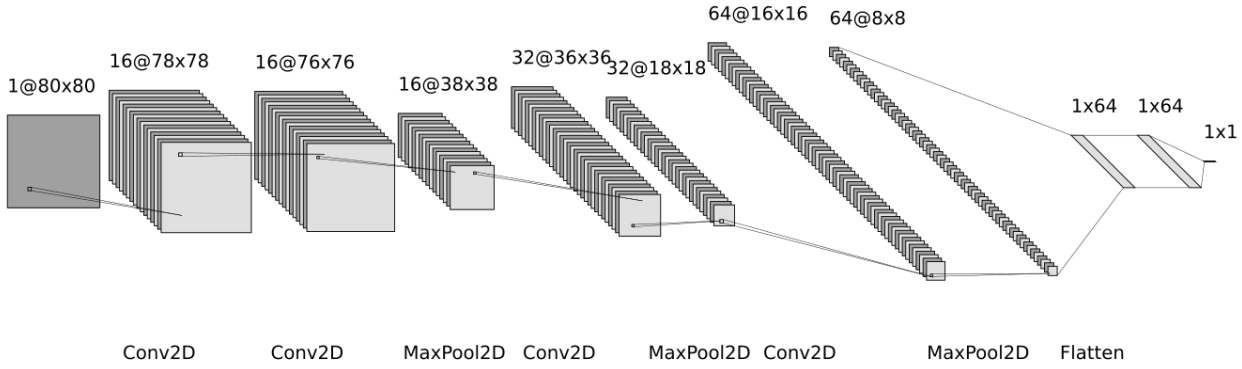
Dokumentácia k projektu

Peter Lukáč - xlukac11

Apríl 2020

1 Klasifikácia tváre

Pre klasifikáciu tváre sme použili konvolučnú neuronovú sieť implementovanú v jazyku `python` pomocou knižníc `Keras` a `tensorflow`. Sieť ma klasickú klasifikančnú acrchitektúru.



Obr. 1: Architektúra siete

Všetky konvolučné vrstvy používajú zero padding, krok 1 a ako aktivačnú funkciu používajú `relu`. Plne prepojené vrstvy používajú ako aktivančnú funkciu `relu`. Posledná vrstva používa `sigmoid`. Medzi plne prepojené vrstvy je vložený `Dropout` s hodnotou 0.5 pre zníženie pretrénovania. Ako loss funkcia je použitá `binary crossentropy` a ako optimalizátor je použitý `adam`. Sieť použitá pre vyhodnotenie evaluačných dát bola natrénovaná na `100% accuracy` a `100% val_accuracy`.

1.1 Príprava dat

Ako vstup siete používame jednokanálové čiernobielé obrázky rovnakej veľkosti ako sú doložené pre trénovanie. Pre trénovanie používame `train` aj `dev` data. Z dat je vymedzených 15% z oboch tried pre validáciu. Všetky data sú pred trénovaním normalizované na nulovú strednú hodnotu a jednotkovú smerodatnú odchylku. Normalizačné hodnoty sú uložené a použíté pri evaluácií.

1.2 Experimenty a trénovanie

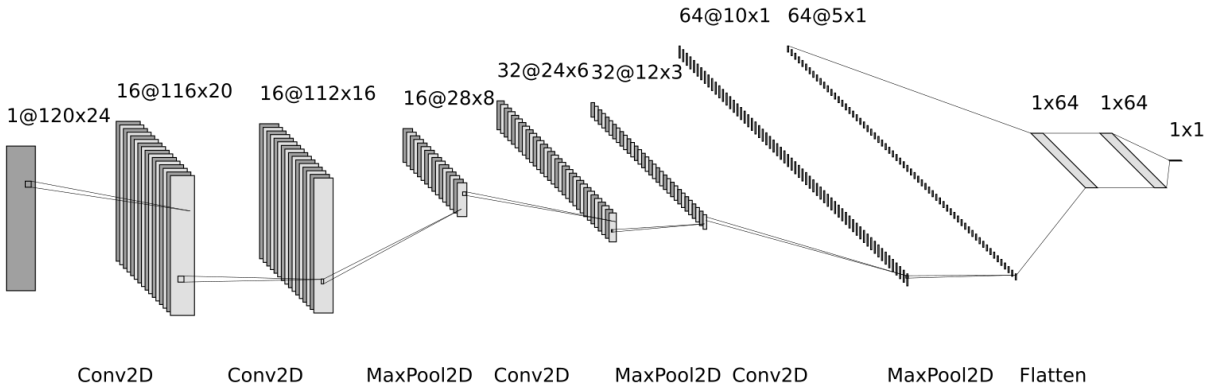
Pri experimentoch sa skúšali rôzne konfigurácie vrstviev a optimalizátora. Pôvodne bolo použitých 6-7 konvolučných vrstiev a veľkosť plneprepojených vrstiev bola 256. Trénovanie veľkej siete na malom počte data sa ukazovalo ako veľmi náchylne na počiatočné nastavenie váh a šum v trénovaním spôsobený použitím batch shuffle. Batch má veľkosť 1. Znížením počtu parametrov sa dosiahli stabilnejšie výsledky. V súčasntej konfigurácí sieť väčšinou dosiahne `100% accuracy` a `100% val_accuracy` po 20 epochách.

1.3 Možné vylepšenia

V trénovacích datách je veľká podobnosť niektorých obrázkov najme v target triede a celkovo malý počet dát. Sieť sme teda validovali na malom počte dát, ktoré sú silno korelované, preto sieť môže byť pretrénovaná aj pri `100% val_accuracy`. Ideálne riešenie tohto problému by bola augmentácia dát.

2 Klasifikácia reči

Pre klasifikáciu reči používame podobnú architektúru a knižnice ako pre klasifikáciu tvári. Líšia sa veľkosťou konvolučných jadier a veľkosťou pooling vrstiev. Reč spracujeme tak aby sme získaly kontextové okná, ktoré sa dajú klasifikovať konvolučnou sieťou.



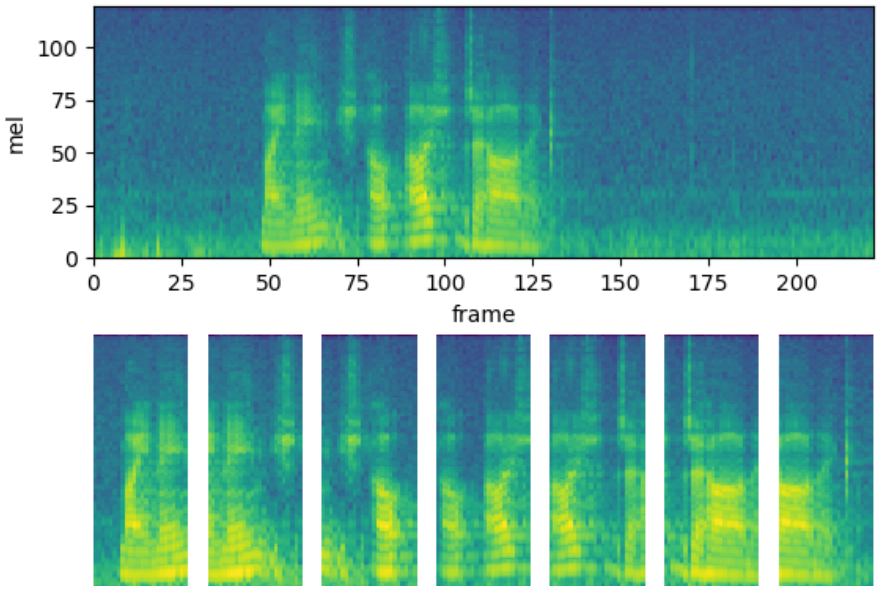
Obr. 2: Architektúra siete

Sieť použitá pre vyhodnotenie evaluačných dát bola natrénovaná približne na `97% accuracy` a `93% val_accuracy`.

2.1 Príprava dat

Pomocou knižnice `scipy` obdržíme spektrogram reči. Veľkosť okna je 500 vzorkov a prekryv 150 vzorkov. Následne vytvoríme power spektrum umocnením spektrogramu. Power spektrum prevedieme do mel frekvenčnej škály pomocou mel bank filtru s počtom filtrov 120. Ako poslednú aplikujeme kompresiu pomocou `log10`.

Z takto upraveného spektrogramu odstraníme ticho tým že vypočítame energiu pre každý rámec a odstránime skupiny rámcov, ktoré majú energiu nižšiu ako threshold(menej ako 25% rozsahu energie danej nahrávky). Z rámcov, ktoré obsahujú reč potom rozdelíme na úseky ktoré budeme trénovať a klasifikovať. Každý úsek má 24 rámcov, čo predstavuje približne 0.5 sekundy. Krok úseku je 12 rámcov, takže úseky sa majú perkriv 12 rámcov.



Obr. 3: Segmentácia rámcov

Segmenty sú pred trénovaním a klasifikáciou normalizované obdobne ako u klasifikácií tvári.

2.2 Experimenty a trénovanie

Pri trénovaní sme mali opäť vymedzených asi 15% dat pre validáciu. Používame opať `binary crossentropy` ako loss funkciu a optimalizátor je použitý `adam`. Dosahovali sme približne `97% accuracy` a `93% val_accuracy`. Experimentovali sme s veľkosťami konvolučných jadier aj veľkosťou kontextového okna. Väčšie okno(24 rámcov) dosiahlo o niečo lepšie výsledky ako pôvodne zamýšľané(16 rámcov). Zväčšenie konvolučného jadra na 5x5 v úvodných vrstvách taktiež prinieslo drobné zlepšenie.

Skóre celej nahrávky je vyhodnotená ako priemer pravdepodobností všetkých segmentov.

`val_accuracy` sa prestalo zlepšovať asi po 12 epochách preto bolo trénovanie zastavené na 12 epoch. Približných `93% val_accuracy` vyzeralo povzbudilo. Neskôr sa ukázalo, že spravidla target rámce boli často klasifikované ako positive false. Vo validačných datách dosahovali non target nahrávky skóre 0.0-0.2 a target nahrávky 0.42-0.8. Toto bolo pravdepodobne spôsobené nevyváženosťou dát, kde približne 92% dat sú non target po tom, čo sú data rozsegmentované. Riešenie problému je teda amatérske a rozhodovacia hranica bola posunutá z 0.5 na 0.35.

2.3 Možné vylepšenia

Všetky nahrávky majú reverb a šum. Toto by sa dalo odstrániť derverbáciou a denosingom. Pre spracovavanie sekvencií by bolo monžno vhodné použiť rekurentné vrstvy.

3 Spustenie a prekvizity

3.1 spustenie

Neuronové siete pre tváre aj reč sa natrénujú príkazom:

```
python3 train_face_cnn.py|train_voice.py KERAS_MODEL.h5
```

Trénovacie data musia byť rozbalené v rovnakom priečinku ako trénovacie skripty.

Evaluácia:

```
python3 eval_face.py|eval_voice.py KERAS_MODEL.h5 EVAL_DATA_FOLDER OUTPUT_FILE
```

3.2 Prekvizity

```
python3
numpy
scipy
Keras
tensorflow
SoundFile
matplotlib
cv2
```