Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Ilkovičova 2, 842 16 Bratislava 4

Neurónové siete

maCapella

Semestrálny projekt

Členovia tímu: Bc. Adam Puškáš

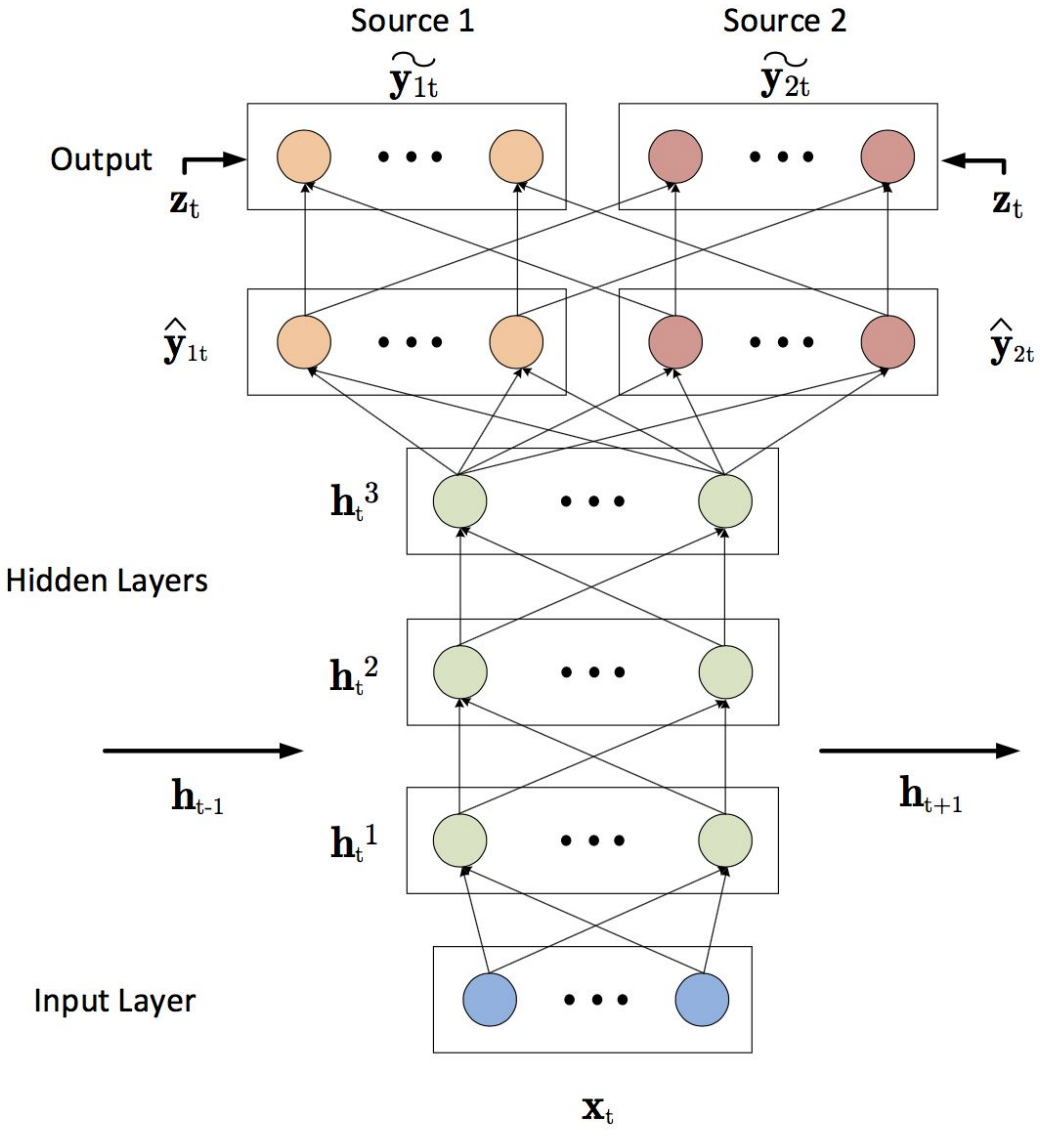
Bc. Lukáš Radoský

Akademický rok: 2019/2020

Úvod

Cieľom projektu je vytvoriť neurónovú sieť schopnú separovať hlas interpreta z hudobných klipov. Jej vstupom je jednokanálový hudobný klip vo formáte *.wav*, pričom na výstupe je jednokanálový *.wav* súbor obsahujúci len hudobný podklad z klipu.

Základný baseline projektu je postavený na existujúcej implementácii neurónovej siete podobného zamerania[1], ktorá separuje hlas aj hudbu z hudobného klipu súčasne, vetvením skrytých vrstiev pri konci sekvencie. Trénovanie jej autori uskutočňujú nad datasetom *MIR-1K*[2], ktorý obsahuje 110 karaoke piesní naspievaných prevažne amatérskymi participantmi.



Obrázok č. 1: Architektúra treťostrannej implementácie

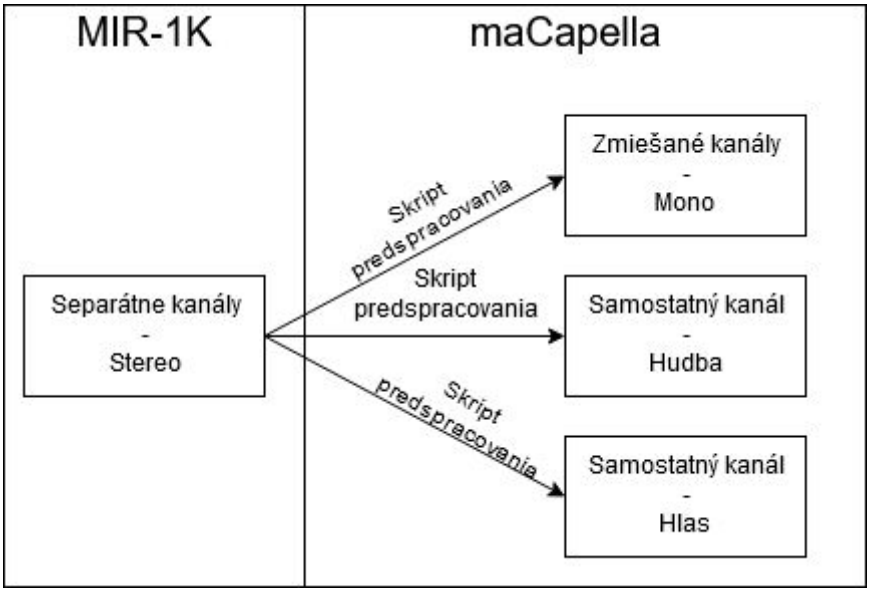
[1] https://github.com/andabi/music-source-separation
[2] https://sites.google.com/site/unvoicedsoundseparation/mir-1k

Predspracovanie dát

Dataset *MIR-1K* využíva aj projekt *maCapella*. Uloženie hudby a hlasu v separátnych kanáloch je veľmi výhodné pri danej problematike. Predspracovanie dát prebieha v dvoch fázach:

- jednorazové predspracovanie
- predspracovanie pri spustení.

Jednorazové predspracovanie je vykonané len raz, mimo spustenia trénovania či validácie modelu. Daný dataset poskytuje hudobné klipy obsahujúce hudbu a hlas v separátnych kanáloch. Každý z klipov poskytuje aj rozdelený do viacerých menších klipov, pre pohodlnejšie spracovanie. Pre projekt *maCapella* je potrebné pre každý z nich vytvoriť kópiu so zmiešanými kanálmi, ako aj kópiu s izolovaným kanálom (hudbou alebo hlasom, v závislosti od cieľa projektu). Tvorba týchto kópií je realizovaná v samostatných skriptoch, ktoré sú jednorazovo vykonané pred spustením modelu. Každé spustenie modelu potom operuje nad takto predspracovaným datasetom. Výhodou je zníženie výpočtovej náročnosti trénovania aj testovania.



Obrázok č. 2: Jednorazové predspracovanie

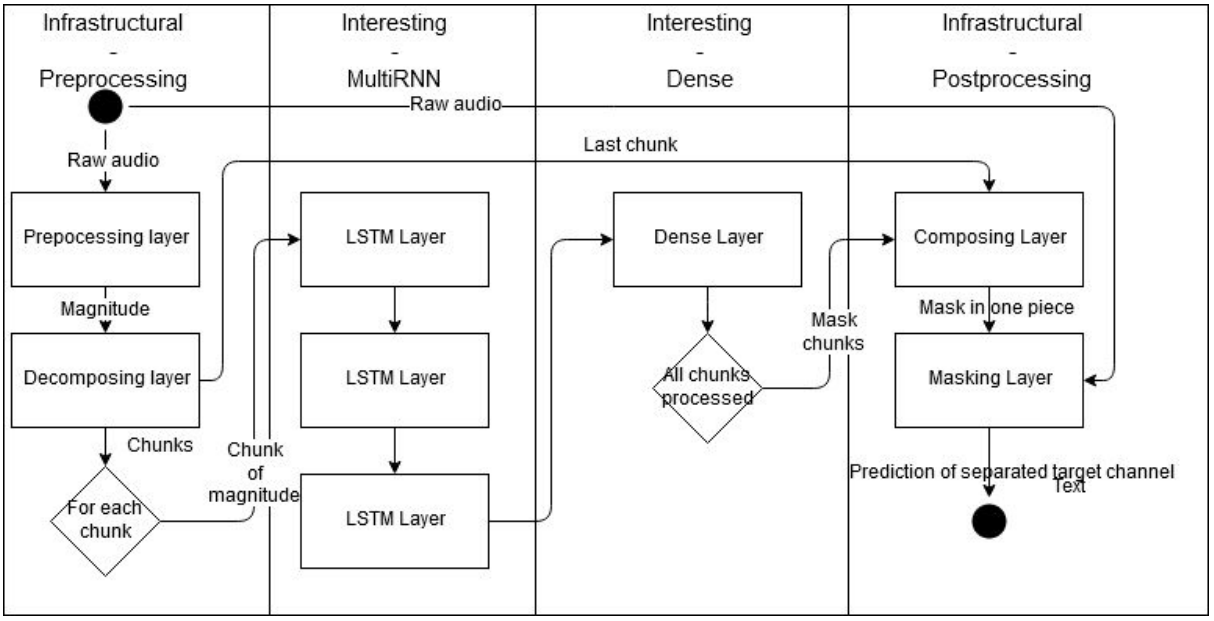
Z pohľadu súčasného riešenia je podstatné vytvorenie súboru so zmiešanými kanálmi a so samostatným kanálom s hudbou. Na základe jedného atribútu v konfiguračnom súbore je však možné ľahko zmeniť cieľ projektu na opačný - izoláciu hlasu a odstránenie hudby.

Predspracovanie pri spustení je vykonané pri každom trénovaní modelu a je podstatné z hľadiska použitej architektúry rekurentných neurónových sietí. Bližšie je popísané v nasledujúcej časti, nakoľko je realizované ako siete v modeli.

Architektúra modelu

Hlavná zložka modelu je postavená na rekurentných neurónových sieťach, pričom obsahuje aj plne prepojenú[3] vrstvu na konci. Tieto vrstvy realizujú zaujímavé výpočty, teda požadované učenie. Okrem toho model obsahuje niekoľko infraštrukturálnych vrstiev. Tieto vrstvy majú dvojaký účel:

- **predspracovanie dát** - aby vrstvy realizujúce skutočné výpočty a učenie operovali nad vhodnými dátami
- **post-spracovanie dát** - aby výstupom modelu boli dáta vhodné pre výpočet stratovej funkcie či presnosti modelu.



Obrázok č. 3: Architektúra modelu

Preprocessing layer

Je vlastná vrstva projektu. Na vstupe prijíma “raw” audio, teda vektor číselných hodnôt. Je výstupom je magnitúda audia, teda matica nezáporných číselných hodnôt.

Decomposing layer

Je vlastná vrstva projektu. Jej vstupom je magnitúda z predošlej vrstvy, jej výstupom je postupnosť magnitúd. Rozdeľuje vstup na niekoľko dávok, *chunks*. Cieľom je zabezpečiť, aby LSTM vrstvy dostávali vstup o jednotnej veľkosti. Jednotlivé *chunks* sú spracované zaujímavými vrstvami, a napokon spojené do výsledného vektora.

[3] z angl. *dense layer*

LSTM layer

Je klasická LSTM rekurentná vrstva, ktorá využíva implementáciu poskytnutú frameworkom *Tensorflow.* V modeli sa nachádzajú 3 za sebou, tvoriac tzv. multi-RNN vrstvu.

Dense layer

Je klasická plne prepojená vrstva, často používaná ako výstupná. Z pohľadu zaujímavých vrstiev je v modeli finálnou, teda výstupnou vrstvou.

Composing layer

Je vlastná vrstva projektu. Jej vstupom je postupnosť *chunks*, teda kusov masky. Na výstupe ich vracia spojené do jednej veľkej masky.

Masking layer

Je vlastná vrstva projektu, založená na treťostrannej implementácii, z ktorej projekt vychádza. Na vstupe prijíma masku vytvorenú rekurentnou neurónovou sieťou a pôvodný zmiešaný signál, pričom vracia audio, ktoré vznikne aplikáciou tejto masky na daný zmiešaný signál. Toto audio by pri natrénovanom modeli malo byť samostatnou hudbou, resp. želanou zložkou.