Project 2

Signals & Systems

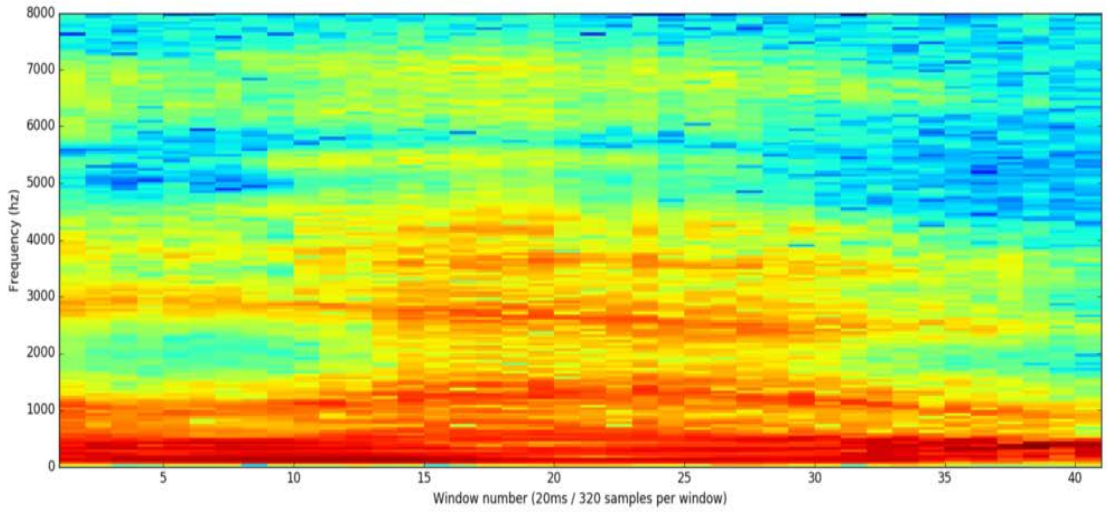
Yes or No Speech Recognition

Written by Soheil Rastegar

توضیحات برنامه

ابتدا با استفاده از ماژول Wavio داده های موجود خوانده می شود. سپس هر ۲۰ میلی ثانیه از هر کدام از فایل های آموزشی به صورت جدا پردازش می شود. از آنجایی که ۲۰ میلی ثانیه بازه ی کوچکی است، به احتمال زیاد در هرکدام از آن ها فرد مشغول ادای حرف خاصی است یا سکوت کرده است و اگر بتوان این ۲۰ میلی ثانیه را تشخیص داد، با توجه به اطلاعات قبلی و بعدی می توان لغت ادا شده را نیز پیدا کرد.

سپس در هرکدام از نمونه های ۲۰ میلی ثانیه ای، تبدیل فوریه گرفته شده و میزان انرژی سیگنال در هر بازه ی ۵۰ هرتزی حساب می شود. این کار در واقع به مثابه ی تهیه اثر انگشت از نمونه است.



شکل ۱ – طیف انرژی موجود در هر ۵۰ هرتز از هر ۲۰ میلی ثانیه از یک سیگنال صوتی

ایده اولیه که در نسخه ۱ (YesNo-v1.py) پیاده سازی شد، این بود که از این طیف ها برای داده های آموزشی موجود در هر کلاس، میانگین گیری شود و سپس داده های ورودی با آن تطبیق داده شود. اما این ایده دو مشکل داشت. اول اینکه طول سیگنال های صوتی متفاوت بود، و

دوم اینکه ممکن است فرد در زمان متفاوتی کلمه ی مورد نظر را ادا کند، یا زمان بیشتری طول بکشد که آن کلمه را ادا کند.

برای حل این مشکل، این نکته در نظر گرفته شد که لحظه ای که فرد شروع به ادای کلمه می کند و به ازای هر حرفی که ادا کند، در اثرانگشت ۲۰ میلی ثانیه ی سیگنال به صورت ناگهانی تغییرات زیادی ایجاد می شود. و می توان در هر سیگنال ورودی، ۳ نقطه ای که بیشترین تغییرات را دارند پیدا کرد، میانگین طیف انرژی سیگنال بین این ۳ نقطه را پیدا کرده و آن هارا با هم مقایسه کرد.

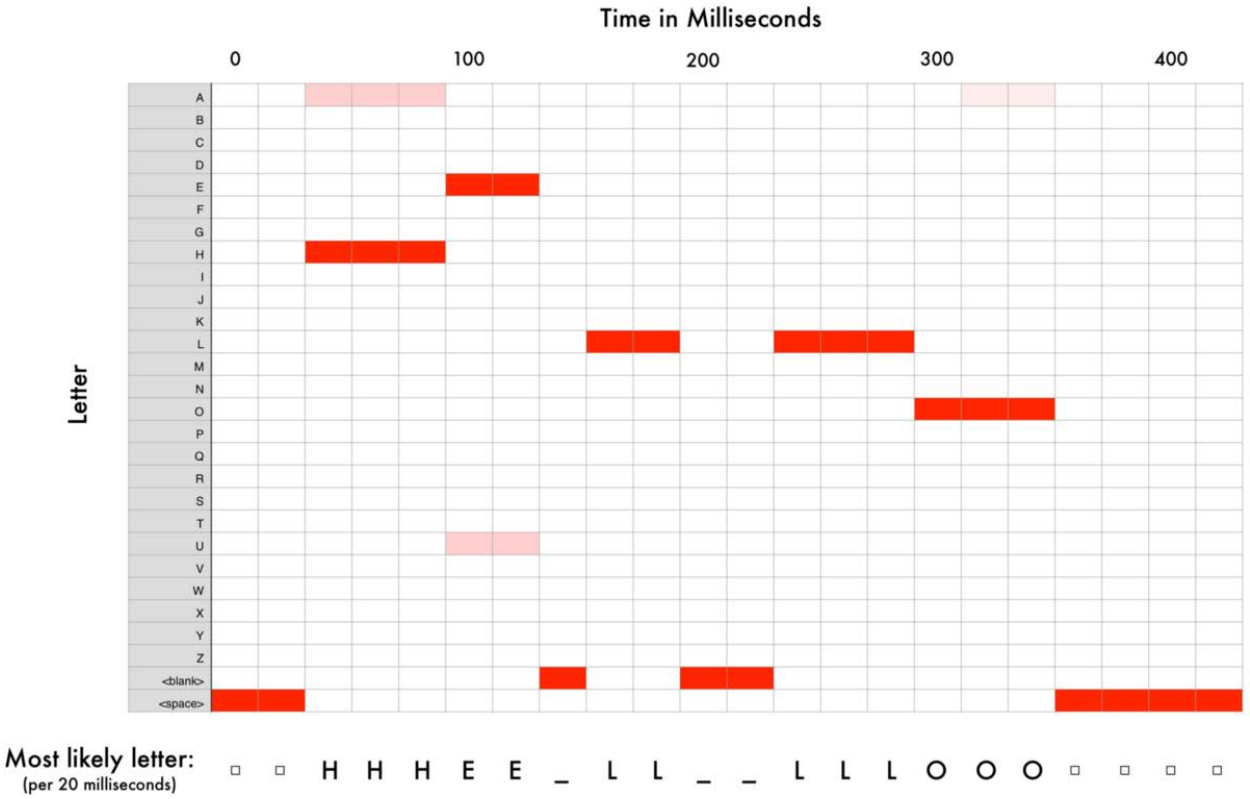
این روش نسخه ی دوم برنامه است و در کد YesNo-V2.py پیاده سازی شده است، مقداری عملکرد و دقت برنامه را بهبود بخشید، اما هنوز مشکلاتی وجود داشت. برای مثال ایده اصلی این بود که مثلا ادای کلمه ی No سه نقطه ی تغییرات دارد، تبدیل از بی صدا به N ، تبدیل از N به O و تبدیل از o به بی صدا. اما روش های مختلف ادا کردن ممکن است تعداد این نقاط را تغییر بدهد و نتوان مرز بین حروف را با این روش به دست آورد.

```
Accuracy For No:  97.32441471571906
Accuracy For Yes:  50.0
```

ارزیابی روش دوم بر روی داده های آموزشی و آزمایشی یکسان

```
Accuracy For No:  97.0
Accuracy For Yes:  45.0
```

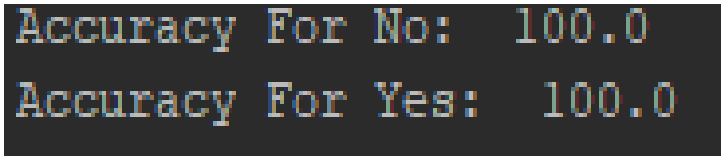
ارزیابی روش دوم برای داده های آموزشی و آزمایشی متفاوت



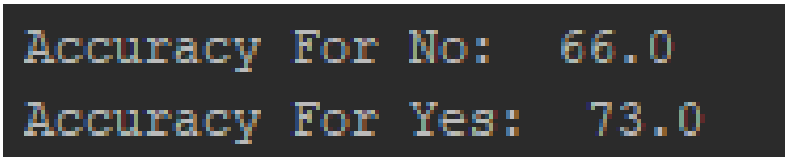
شکل ۲ - تطبیق نمونه های ۲۰ میلی ثانیه ای با حروف مختلف. اینکه در داده های آموزشی مشخص نبود هر حرف از کجا شروع می شود، باعث شد نقطه ی تغییرات شدید به عنوان نقطه ی تغییر حرف ادا شده در نظر گرفته شود

برای همین، لزوم وجود یک طبقه بند اجتناب ناپذیر به نظر رسید و از طبقه بند nearest neighbor استفاده شد. با وجود اینکه روش های مختلفی برای ادا کردن وجود دارد اما چون این طبقه بند به جای میانگین گیری که گاهی اوقات باعث خراب شدن اطلاعات موجود در داده ها می شود، نمونه هارا نگه داشته و سیگنال های ورودی را با آن ها مقایسه می کند، عملکرد بسیاری بهتری دارد و در برابر روش های مختلف ادا کردن که نزدیک به داده های آموزشی باشند به خوبی عمل می کند.

در نسخه ی سوم برنامه (YesNo-v3.py) از طبقه بند برای بهبود کارای استفاده شد و نتایج بدست آمده بهبود یافت.



ارزیابی روش سوم برای داده های آموزشی و آزمایشی یکسان



ارزیابی روش سوم برای داده های آموزشی و آزمایشی متفاوت

کد اصلی (YesNo.py) همان نسخه ی سوم است که به صورت آنلاین از میکروفون ورودی می گیرد و خروجی می دهد.

منبع:

https://medium.com/@ageitgey/machine-learning-is-fun-part-6-how-to-do-speech-recognition-with-deep-learning-28293c162f7a