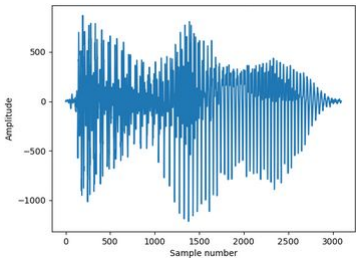
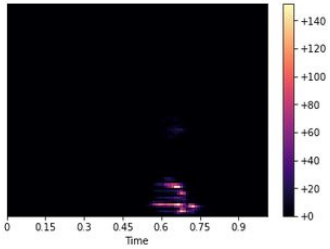
CNN Image Classification For Verbal Commands

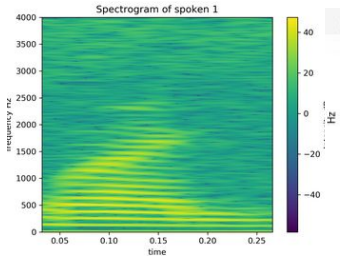
Chris Hickey 논문연구 1

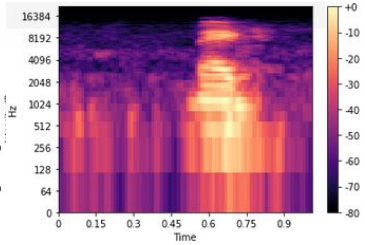
Introduction

Various ways to represent sound visually:









Question?

Can a network be trained to recognise meaning from these images?

Why is this important?

- **Provide alternative to RNN classifiers for simple speech recognition**
- **Leverages advances in computer vision/image recognition in the domain of NLP**
- **May be particularly useful for command recognition**

Dataset: 65,000 1 second utterances of 30 words spoken in noisy environments.

Roughly 2,000 samples for each class

Relatively new dataset (2 years old)

Google AI Blog

The latest news from Google AI

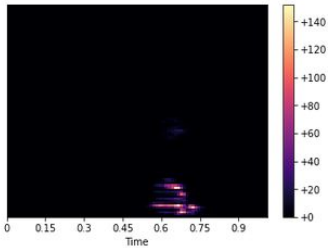
Launching the Speech Commands Dataset

Thursday, August 24, 2017

Results

Which visual representation of data performed best?

Initial experimentation found that the mel-spectrogram gave the image most trainable by NN.



What is a mel-spectrogram?

Spectrogram: Visual representation of a (audio) signal over time .

Mel Scale: A perceptual scale of pitches judged by listeners to be equal in distance. This a non-linear transformation of Hz scale, where the difference between 500 and 1000 Hz is obvious, whereas the difference between 7500 and 8000 Hz is barely noticeable.

Network Configurations

- 5-layer convolutional neural network.
- Batch normalization after 1st layer
- 2 x 2 Max Pooling after each layer
- ELU activatation after each layer
- Dropout 0.6 after each layer
- Softmax activation for output
- Bath-size 32
- Loss function *categorical crossentrophy*
- Optimizer adadelta

Results

Data Split

- 65,000 audio files split 70/15/15 for Train/Val/Test
- Classes were represented equally in each data subset

Results (100 epochs)

- Performance on combined Train/Val set: 90% accuracy
- Performance on Test set: 80% accuracy

Real Life Testing

- Tested using samples recorded on my iPhone
- Correct over 90% of time, often with high certainty

```
In [18]: filename = os.path.join('../SelfRecordedSamples', 'bSayingBed.wav')
         print_prediction(filename)

bed             :  0.9523471593856811523437500000000
bird            :  0.0055287349969148635842578125000
cat             :  0.0016157346544787287712097167968
dog             :  0.0002907297748606652021408081054
down            :  0.0000266831739281769841909408569
eight           :  0.0010476532625034451484680175781
five            :  0.0011811478761956095695495605468
four            :  0.0001552049943711608648300170898
go              :  0.0008292457787320017814636230468
```

- Mistakes are usually for words that sound similar with high uncertainty

```
In [17]: filename = os.path.join('../SelfRecordedSamples', 'meSayingNo.wav')
         print_prediction(filename)

bed             :  0.0000022811691451352089643478393
bird            :  0.0000002049253424729613470844924
cat             :  0.0000010252111906083882786333560
dog             :  0.0072210403159260749816894531250
down            :  0.0103634102270007133483886718750
eight           :  0.0000012578508403748855926096439
five            :  0.0000000249366287619068316416814
four            :  0.0000028803917757613817229866981
go              :  0.5189195275306701660156250000000
happy           :  0.0000001075301341302292712498456
house           :  0.0022848043590784072875976562500
left            :  0.0000002059786794461615500040352
marvin          :  0.0000000359066909538796608103439
nine            :  0.0000250592038355534896254539489
no              :  0.4608223140239715576171875000000
```