

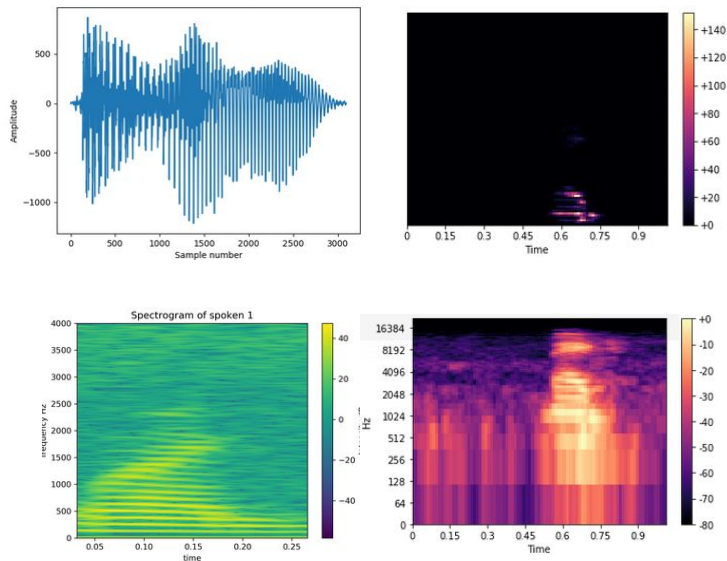
CNN Image Classification For Verbal Commands

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)



The latest news from Google AI

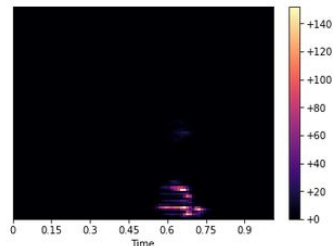
Launching the Speech Commands Dataset

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** activation after each layer
- **Dropout** 0.6 after each layer
- **Softmax** activation for output
- **Bath-size:** 32
- **Loss function:** categorical crossentropy
- **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: 89% 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          : 0.95234715938568115234375000000000
bird         : 0.00552873499691486358642578125000
cat          : 0.00161573465447872877120971679688
dog          : 0.00029072977486066520214080810547
down        : 0.00002668317392817698419094085693
eight       : 0.00104765326250344514846801757812
five        : 0.00118114787619560956954956054688
four        : 0.00015520499437116086483001708984
go          : 0.00082924577873200178146362304688
```

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

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

bed          : 0.00000228116914513520896434783936
bird         : 0.0000020492534247296134708449244
cat          : 0.00000102521119060838827863335609
dog          : 0.00722104031592607498168945312500
down        : 0.01036341022700071334838867187500
eight       : 0.00000125785084037488559260964394
five        : 0.0000002493662876190683164168149
four        : 0.00000288039177576138172298669815
go          : 0.51891952753067016601562500000000
happy       : 0.00000010753013413022927124984562
house       : 0.00228480435907840728759765625000
left        : 0.00000020597867944616155000403523
marvin      : 0.0000003590669095387960081034392
nine        : 0.000025059283835534896245334897
no          : 0.46082231402397155761718750000000
```