Term Project
End-to-End Key word spotting

2018 Spring
Bae Jaesung (bjsd3@kaist.ac.kr)
Kang Minsung (minsung2646@kaist.ac.kr)

Contents

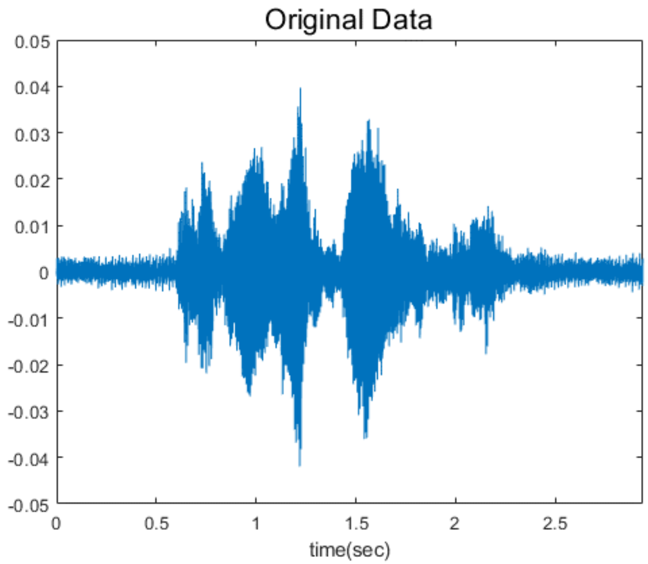
Objective

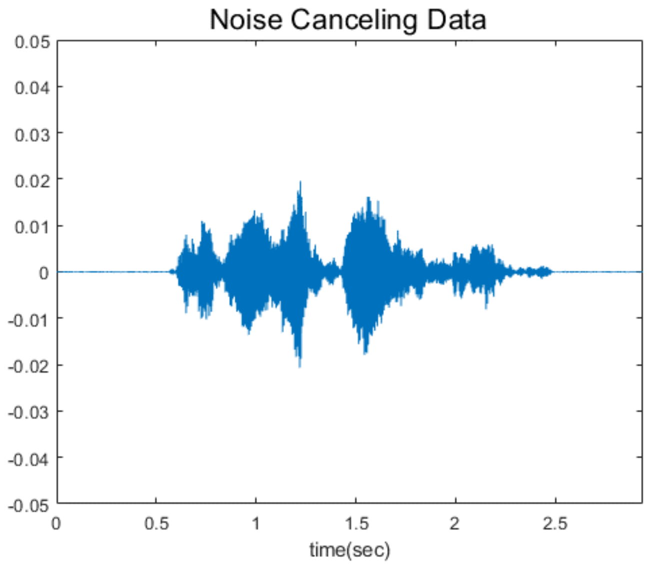
- Keyword spotting
  - Identification of keywords in utterances.

- Our goal is recognize 3 syllable keyword from microphone data.

Preprocessing

- **Noise canceling**
  - Refrigerator has the stationary noise.
  - Human can not speak right after turning on the microphone.
  - From the first few samples, we can estimate the stationary noise.
  - Delete the stationary noise in frequency domain.





Preprocessing

Feature Extraction

- **Sampling rate convert**
  - For using DNN algorithm, we need a lot of data set.
  - Therefore we decide add 3 times more data by converting sampling rate.
  - Slow : 90% & Fast : 110%

Noise Original

Noise Slow

Noise Fast

NC Original

NC Slow

NC Fast

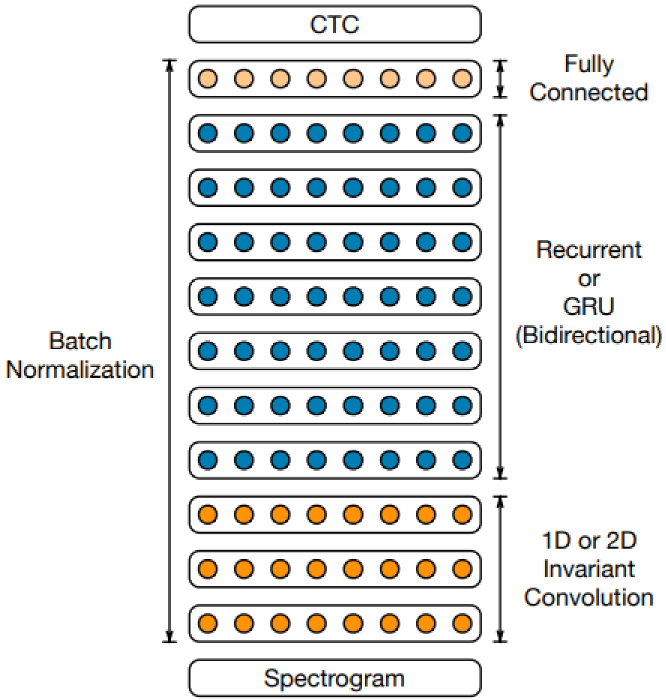
Data Set

- **Data set**

Type	3 kinds of Keywords (3 syllable)		Non Keyword (sentence)	
	TV	냉장고	TV	냉장고
Training	135*3	270*3	15000	30000
Test	7*3	7*3	500	500

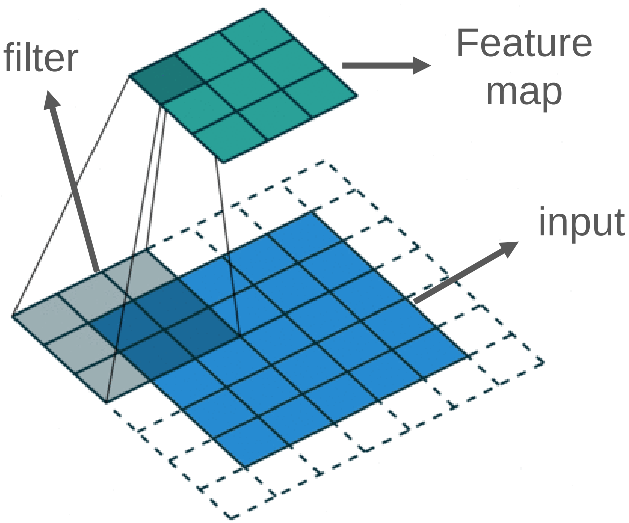
- **Data Augmentation**
  - increase 3 times training sample of TV and 6 times training sample of refrigerator.
- **Sampling ratio = 18 kHz**

- **Deep Speech 2: End-to-End Speech Recognition in English and Mandarin[3]**
- **Deep speech2 Model Architecture**
  - **Deep** architecture to increase model capacity for large dataset.
  - Convolutional layer + Recurrent layer + Fully-connected layer + CTC Loss
  - **Batch Normalization** in every layer.

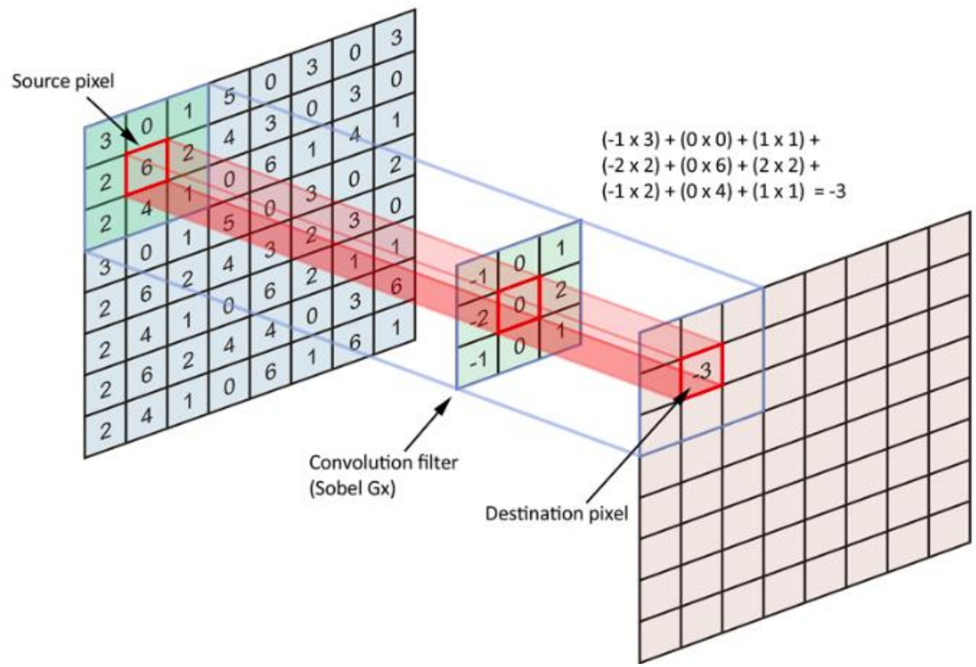


Deep speech2
Methodology

- **1. Convolutional layer.**
  - Execurate a convolution by sliding the filter over the input.
  - Matrix multiplication is performed and sums the result onto the feature map.
  - Could extract the local feature.



https://medium.com/syncedreview/a-guide-to-receptive-field-arithmetic-for-convolutional-neural-networks-42f33d4378e0



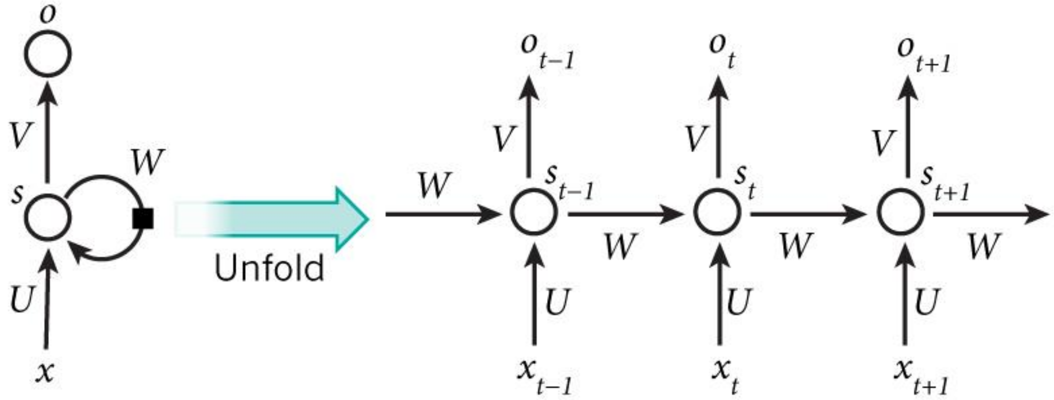
https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

▪ 2. Recurrent layer.

- Recurren Neural Netwrok(RNN), Long-shor-term-memory(LSTM), bidirectional LSTM
- Idea: output is dependent on the previous computations; learn sequence data's feature.
- x_t: input at time step t.
- s_t: hidden state at time step t.
- o_t: output at time step t.
- f: activation function, usually tanh or ReLU.

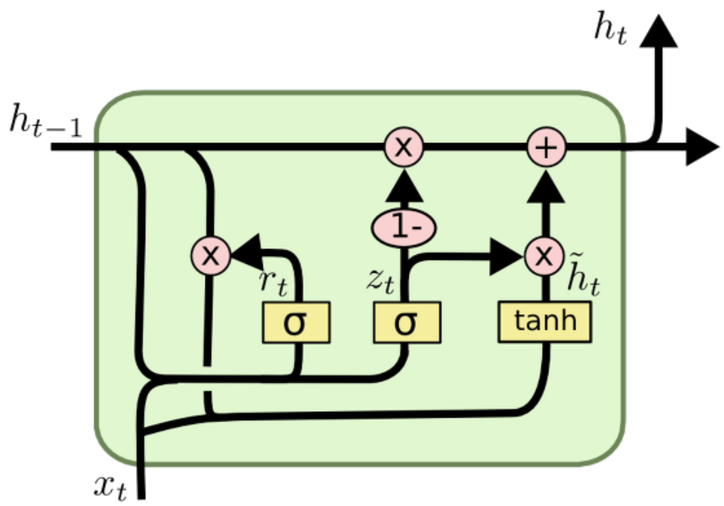
$$s_t = f(Ux_t + Ws_{t-1})$$

$$o_t = softmax(Vs_t)$$

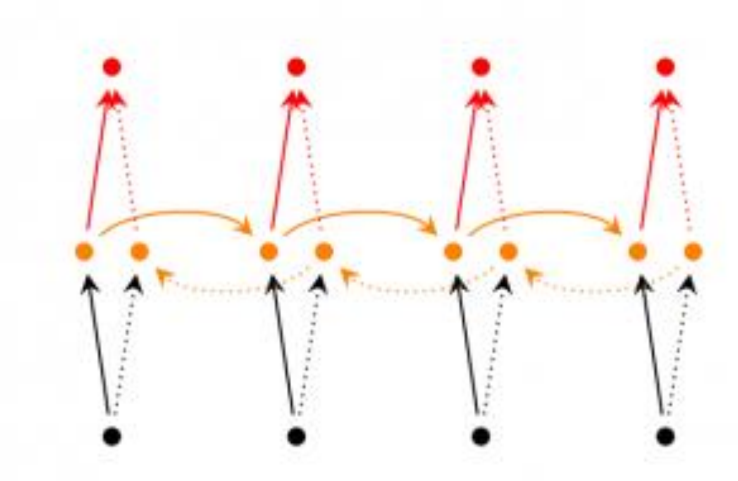


http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/

- **2. Recurrent layer.**
  - Recurren Neural Netwrok(RNN), Long-shor-term-memory(LSTM), bidirectional LSTM
  - LSTM: much better at capturing long-term dependencies than vanilla RNNs.
  - Bidirectional LSTM: not only depend on the previous elements, but also future emelments.



http://colah.github.io/posts/2015-08-Understanding-LSTMs/



http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/

3. Batch Normalization

- Prevent the Gradient Vanishing/Gradient Exploding.
- Normalize the mini-batch input distribution to fix 'internal covariance shift' and speed up the training process.
- In DeepSpeech2 Model: ***Sequence-wise* normalization** technique is used in bidirectional RNN batch normalization.

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1\ldots m}\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_\mathcal{B} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma_\mathcal{B}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_\mathcal{B})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_\mathcal{B}}{\sqrt{\sigma_\mathcal{B}^2 + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

$$\mathcal{B}(x) = \gamma \frac{x - \mathrm{E}[x]}{(\mathrm{Var}[x] + \epsilon)^{1/2}} + \beta.$$

$$\overrightarrow{h}_t^l = f(\mathcal{B}(W^l h_t^{l-1}) + \overrightarrow{U}^l \overrightarrow{h}_{t-1}^l).$$

https://shuuki4.wordpress.com/2016/01/13/batch-normalization-%EC%84%A4%EB%AA%85-%EB%B0%8F-%EA%B5%AC%ED%98%84/

▪ 4. CTC loss[4]

- For audio feature $X = [x_1, x_2, \dots, x_T]$ and the label sequence $Y = [y_1, y_2, \dots, y_U]$.
- Both X and Y is vary in length and don't have accurate alignment.
- Introduce extra 'blank' label; to make repeated label possible.

$$p(Y \mid X) \quad = \quad \sum_{A \in \mathcal{A}_{X,Y}} \quad \prod_{t=1}^{T} p_t(a_t \mid X)$$

The CTC conditional **probability**

marginalizes over the set of valid alignments

computing the **probability** for a single alignment step-by-step.

h	h	e	ϵ	ϵ	l	l	l	ϵ	l	l	o

First, merge repeat characters.

h	e	ϵ	l	ϵ	l	o

Then, remove any ϵ tokens.

h	e		l		l	o

The remaining characters are the output.

h	e	l	l	o

Deep speech2

Methodology

▪ 5. Decoder

– Greedy Decoder: choose the label which has maximum probability, and ignore the ‘blank’ label.

blank	0	0.5	0.1	0.1	0.2	0.3	0.1	0.1
A	1	0.1	0.1	0.3	0.2	0.1	0.1	0.2
B	2	0.1	0.1	0.1	0.2	0.1	0.5	0.3
C	3	0.1	0.3	0.1	0.2	0.2	0.1	0.2
D	4	0.1	0.2	0.2	0.1	0.2	0.1	0.1
E	5	0.1	0.2	0.2	0.1	0.1	0.1	0.1

time

(ex)
Argmax output: _ C A B _ B B
Merge: C A B B

– For other decoder we could consider beam search decoder.

– Do not use language model; in the original paper, they use n-gram language model.

Experiment

- **Text Preprocessing.**
  - There are some choice needed texts. Choose second one (Hangul)
    - (ex) 우리 만난 지 벌써 [4]/[사]년이나 되었구나
  - Convert 'Hangul' to Phoneme with given lexicon.
  - There are some words that its phoneme are not given. -> ignore these audio.
    - (ex) 잠시 뒤 [11]/[열한]시 [15]/[십 오]분 **PD수첩**에서는 무상 급식
    - (ex) **나+B354더** 걱정이 많은 사람 중에 한 사람이지
    - (ex) 저는 경남 마산에 사는 **9년차** 주부랍니다
  - All phoneme that are used.
    - 'B', 'D', 'E', 'G', 'H', 'N', 'S', 'U', 'Wi', 'Z', 'a', 'b', 'c', 'd', 'e', 'g', 'h', 'i', 'jE', 'ja', 'je', 'jo', 'ju', 'jv', 'k', 'm', 'n', 'o', 'p', 'r', 's', 't', 'u', 'v', 'wE', 'wa', 'we', 'wi', 'wv', 'xb', 'xd', 'xg', 'xl', 'xm', 'xn', 'z'
    - Add blank label at the first. (For CTC loss)
- **Audio Preprocessing**
  - Use 40 log-mel filterbank with delta and delta-delta. (120 dimension)
  - 20msec window, 10msec window step.
- **Data Augmentation**
  - For training dataset, augmented 3 times for TV and 6 times for 냉장고 noise.
  - TV: Slow(x0.9), Normal(x1.0), Fast(x1.1)
  - 냉장고: [Noise, Noise canceled] x [Slow, Normal, Fast]

Experiment

- **Phoneme Error Rate(PER)**
  - Because Hangul is converted in to Phoneme in English, we use PER for measure the performance.
  - Similar to Word Error Rate (WER)

$$WER = \frac{S+D+I}{N} = \frac{S+D+I}{S+D+C}$$

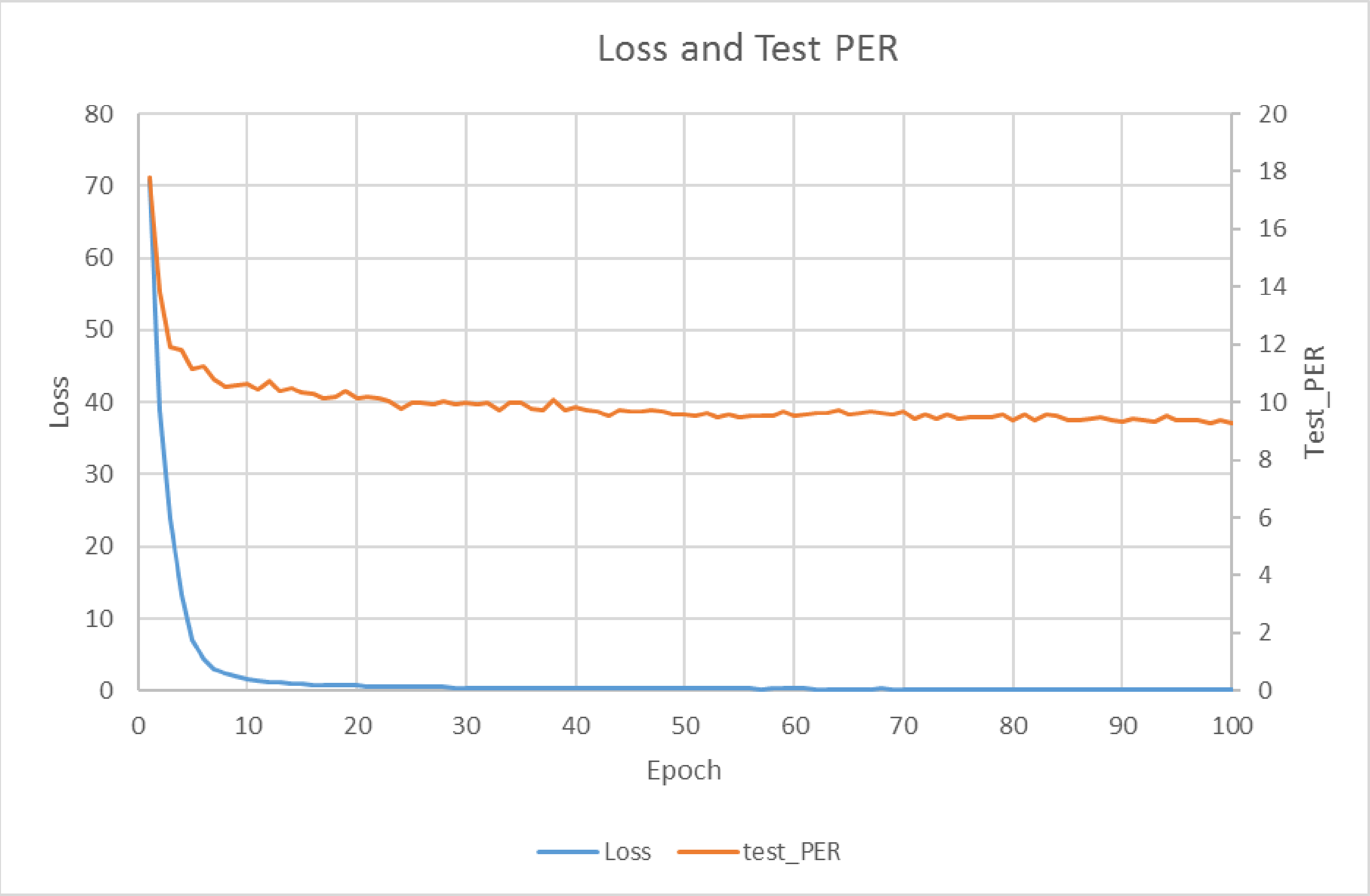
    - S is the number of substitutions, D is the number of deletions, I is the number of insertions, C is the number of correct words, and N is the number of words in the reference (N=S+D+C)

- **Training Process.**
  - Use Adam optimizer with learning rate 3e-4
    - In the orginal paper, they use new 'SortaGrad' for training, but we use simple Adam optimizer.
  - Batch size: 30

- **Implementation**
  - Use MATLAB for noise canceling and pytorch library for DeepSpeech2 model.[5]

Result

- **Loss and Accuracy(Test dataset PER) graph.**
  - Final Test PER: 9.25%



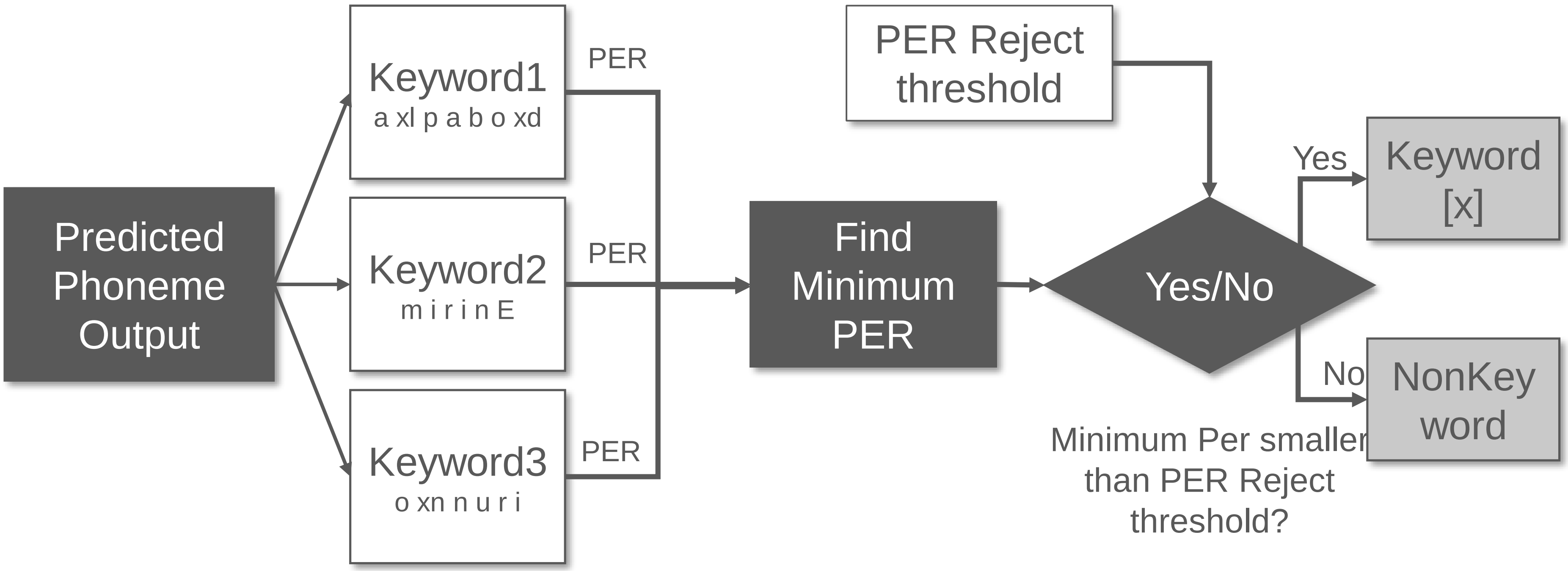
Result

- ## Result Example

	Keyword	Nonkeyword
Predicted	m i xn n E	v xm z v N m a xn s a i t U r U xl s o g E H E z u s jv xn n e jo
GT	m i r i n E	v xm c v N n a xn s a i t U r U xl s o g E H E z u s jv xn n e jo
Predicted	a xl p a b o xd	n e i r e t e i xm s U t a t U g a i S U xl g v xd g a t a jo
GT	a xl p a b o xd	m e i n e g e i xm s U t a t U g a i S U xl g v xd g a t a jo
Predicted	o xn n u r i	b a s i xn g a xn n a n U xn m a n U xn g jv xn s v N g a g i n i r jv s v
GT	o xn n u r i	g U s u xn g a xn n a n U xn m a n U xn b a xn s v N U xl h a g e d we v S v
Predicted	m i r i n E	n a xl S i g a n v m u z o a z v z i D U xn g v g a g o E xd Z v n U xn
GT	m i r i n E	n a xl S i g a n v m u z o a s v z a z v xn g v t a g o c u xl g U xn
Predicted	a xl p a b o xd je r U	n i g a n a z o a H a z i a xn n U xn d a g o h E g i jo
GT	a xl p a b o xd	n i g a n a z o a H a z i a xn n U xn d a g u h E d u
Predicted	m i r i n E	g U r v xm n U G i m i z u u xn s a ja g i i xd D a m jv
GT	m i r i n E	g U r v xn n U G i m i d U n U xn s a r a m i i xd D a m jv xn

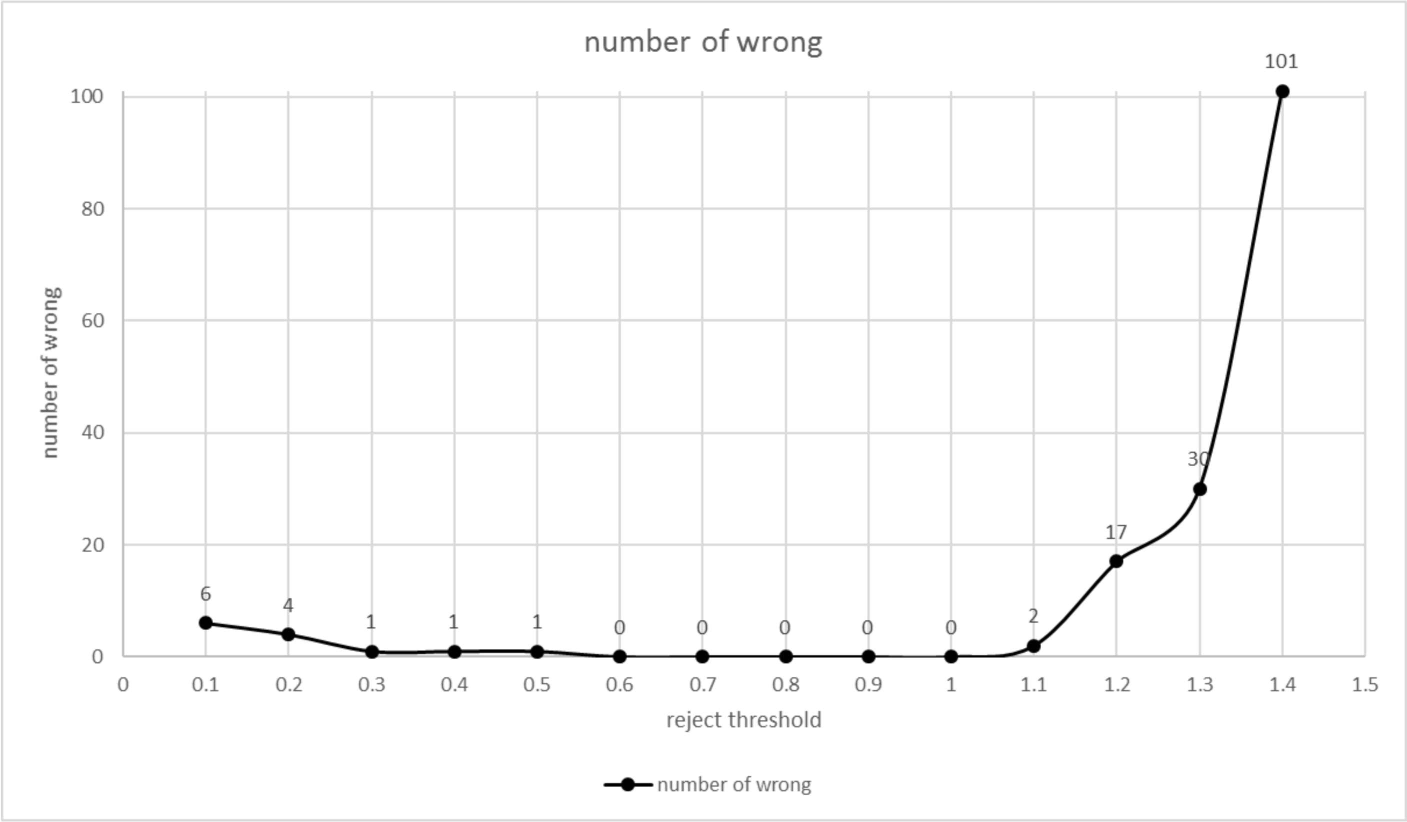
Result

- **Final decision Algorithm based on PER**



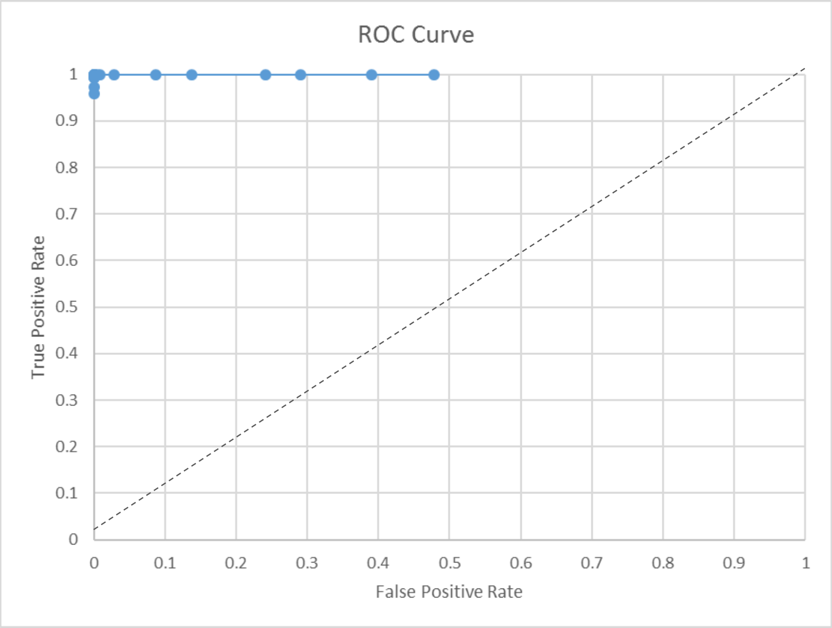
Result

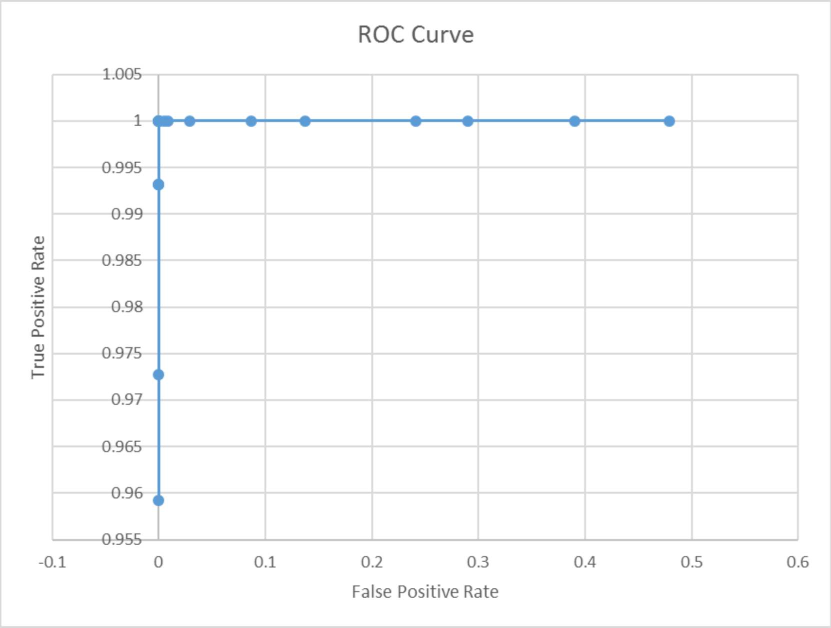
- **Reject threshold – Number of wrong Graph.**
  - Plot the number of wrongly classified test speech by change of reject threshold
- **Final Keyword Detection Accuracy: <u>100%!</u>**



Result

- **Receiver Operating Characteristic (ROC) curve**
  - Our model does not give output as probability value.
  - ROC curve for the our 'PER Reject threshold'.





Discussion & Summary

- **Dicussion.**
  - Applying **language model** could give more accurate result for non-keyword speech for the future work.
  - Applying more amount of data for training, the performance would be better.
    - DeepSpeech2 model was originally trained with very large amount of data.
    - 11,940 hours English dataset containing 8 million utterances.
- **Summary.**
  - Our model is **end-to-end** model.
  - We did **noise canceling** based on Wiener filter.
  - By **data augmentation** we increase the size of data so that we could get enough data for deep learning approach.
  - We successfuly adapted **DeepSpeech2** Model to **Korean speech** which was originally done by English and Mandarin.
  - We solved the keyword spotting task with speech recognition system.

Thank you

Q & A

Reference

[1] Marc C. Green, Damian Murphy, “Acoustic Scene Classification Using Spatial Features”, 2017

[2] Richard Schultz-Amling, Fabian Kuech, Oliver Thiergart, Markus Kallinger, “Acoustical zomming Based on a Parametric Sound Field Representation”, JAES 2010

[3] Amodei, Dario, et al. "Deep speech 2: End-to-end speech recognition in english and mandarin." International Conference on Machine Learning. 2016.

[4] Graves, Alex, et al. "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks." Proceedings of the 23rd international conference on Machine learning. ACM, 2006

[5] https://github.com/SeanNaren/deepspeech.pytorch