Towards End-to-End Speech Recognition Using Deep Neural Networks

Tara N. Sainath
September 16, 2015

Acknowledgements

Work in this talk is presented in collaboration with the following colleagues:

- Andrew Senior
- Oriol Vinyals
- Hasim Sak
- Ron Weiss
- Kevin Wilson
- Yedid Hoshen

Speech Recognition Problem

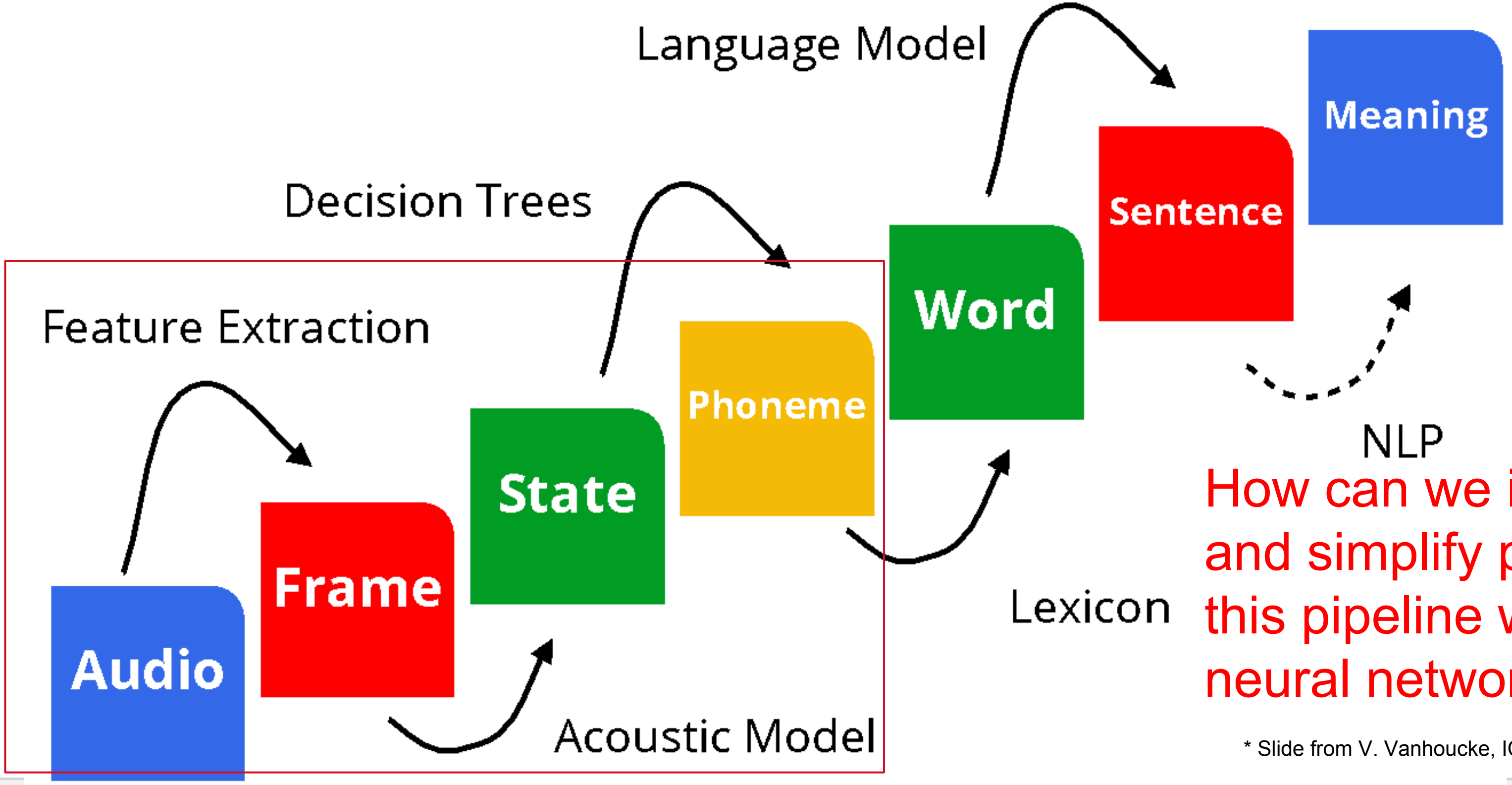
Audio Waveform

$$\underset{W}{\operatorname{argmax}}\; p(W \mid O) = \underset{W}{\operatorname{argmax}}\; \underbrace{p(O \mid W)}_{\text{Acoustic Model}}\; \underbrace{p(W)}_{\text{Language Model}}$$

Word Sequence

* Slide from V. Vanhoucke, ICML 2013 Keynote

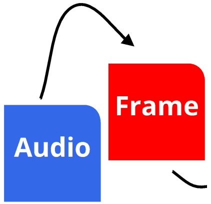
Speech Recognition as Probabilistic Transduction



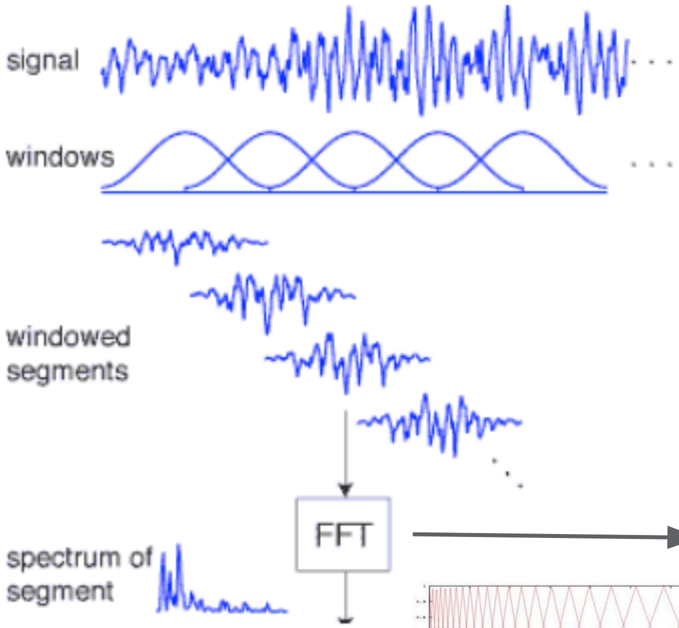
How can we improve and simplify parts of this pipeline with neural networks?

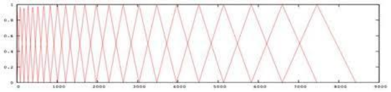
* Slide from V. Vanhoucke, ICML 2013 Keynote

(1) Feature Extraction

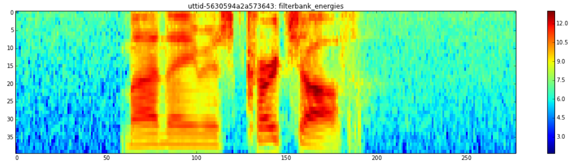


Can we replace this complex pipeline with a neural network?

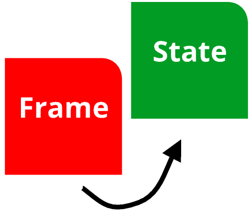




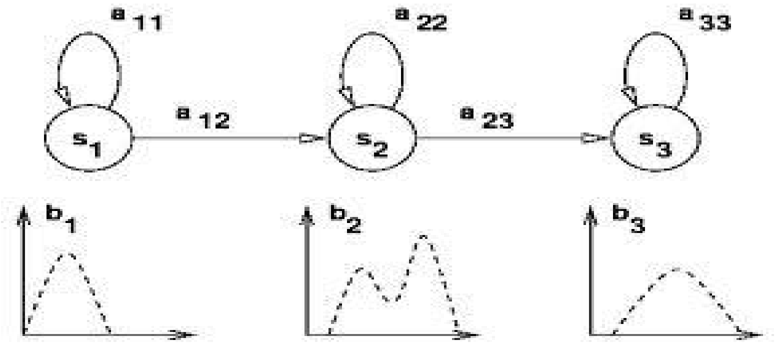
Mel Filterbank → Log



(2) Sub-word unit modeling



- Acoustic modeling is the process of modeling a set of sub-word units
- Each sub-word unit is modeled by a 3 state left-to-right HMM
- Output distribution in each state given by a Deep Neural Network



Deep Learning Technical Revolution

What architecture can I use for for speech tasks?

- First resurgence
  - A. Mohamed, G. Dahl and G. Hinton *"Deep belief networks for phone recognition,"* In NIPS Workshop on Deep Learning for Speech Recognition and Related Applications, 2009.

2009

- DNNs for Large Scale Tasks
  - F. Seide, G. Li, and D. Yu, "*Conversational Speech Transcription Using Context-Dependent Deep Neural Networks*," in Proc. Interspeech 2011.

2011

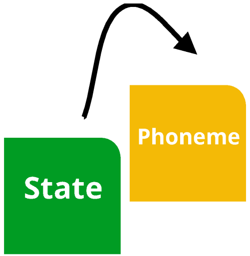
- CNNs for Large Scale Tasks
  - T. N. Sainath, A. Mohamed, B. Kingsbury and B. Ramabhadran, "Deep Convolutional Neural Networks for LVCSR," in Proc. ICASSP, 2013.

2013

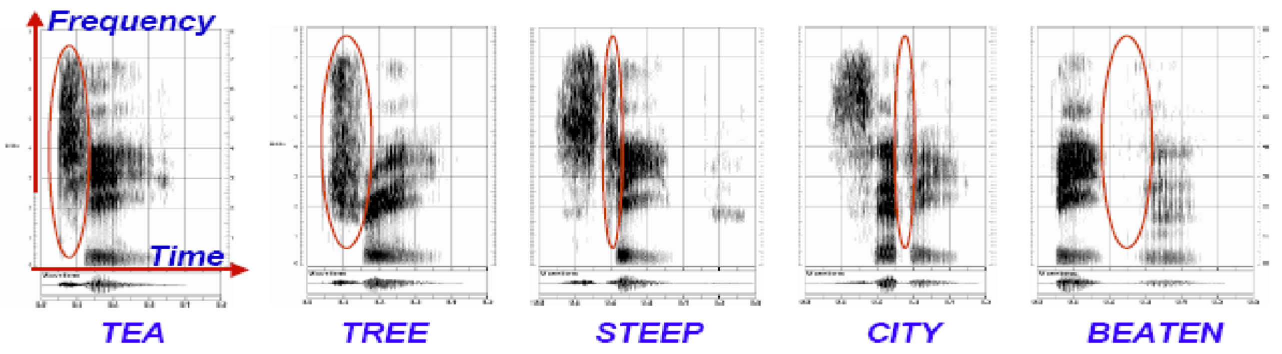
- LSTMs for Large Scale Tasks
  - H. Sak, A. Senior and F. Beaufays, "Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling," in Proc. Interspeech, 2014.

2014

(3) Subword Units



- Acoustic realization of a phoneme depends strongly on context
- We model sub-word units as triphones (context-dependent states)
- 41 phones → total number of CD states ~ 200K (3x41^3)
- Use decision tree clustering to reduce the # of CD states ~ 2K-10K
- Drawbacks:
  - Need to cluster to find the CD states
  - Need to align each frame to a CD HMM state



Towards End-to-End Speech Recognition

1. Feature representation
   - Getting log-mel filterbanks can be complex
   - If neural networks are good at feature learning, can we have it learn features from the raw signal?

2. Acoustic modeling
   - Either DNNs, CNNs or LSTMs are used for acoustic modeling
   - Can we do better by combining these architectures?

3. Training requires an existing alignment and CD states
   - Are CD states really necessary or can we go simpler to phones?
   - Can we use CTC to learn the alignment?

Outline

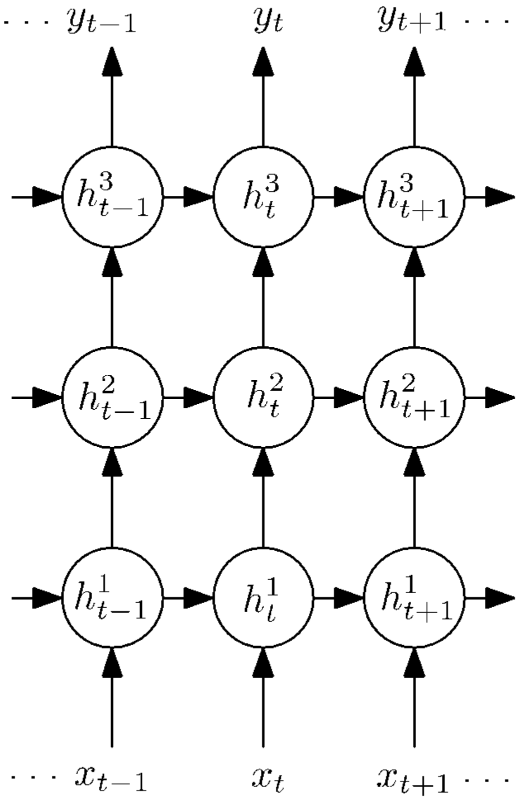
- Motivation
- CLDNNs
- Raw-waveform CLDNNs
- CTC

Motivation

- DNNs have achieved tremendous success for LVCSR tasks in recent years [Hinton et al, 2012]
- Further improvements over standard DNNs have been seen for LVCSR tasks more recently
  - Convolutional Neural Networks [T.N. Sainath, ICASSP 2013]
  - Long Short-Term Memory [H. Sak, Interspeech 2014]
- CNNs, LSTMs and DNNs are individually limited in their modeling capabilities

Basic Deep RNN/LSTM

- Frame t
- Input x_t
- Hidden units h_t
- Output y_t



Limitations of LSTMs [Pascanu, 14]

1. Temporal modeling done directly on input feature x_t

- Higher-level modeling of x_t can help to disentangle underlying factors of variation within the input, which should then make it easier to learn temporal structure
- Convolutional layers are good at reducing spectral variation in the input and map features to a canonical speaker space
- We will explore proceeding LSTM layers with a few CNN layers

Limitations of LSTMs [Pascanu, 14]

2. LSTM mapping between h_t and output y_t is not deep, meaning there is no intermediate nonlinear hidden layer

- By reducing factors of variation in h_t, the hidden state of the model could summarize the history of previous inputs more efficiently. In turn, this could make the output easier to predict.
- Reducing variation in the hidden states can be modeled by having DNN layers after the LSTM layers

CLDNN

- To address the limitations of LSTMs, we proposed the following architecture
  - Pass input feature x_t into CNN layers to reduce spectral variations
  - Pass this to the LSTM for temporal modeling
  - Pass the output of LSTM into DNNs to transform the features into a more separable space
- We term this combined CNN+LSTM+DNN architecture "CLDNN"

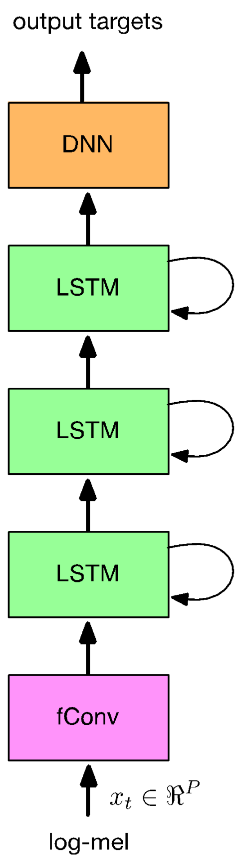
Connection to Speech Recognition Systems

- The following recipe has been shown to be effective for GMM/HMM systems [Soltau, 2010]
  - Speaker-adapted features (VTLN, fMLLR)
  - Model temporally via GMM/HMM system
  - Training GMM/HMM model discriminatively (BMMI)
- Intuitively our model is capturing a similar order of steps
  - CNNs for “speaker-adapted” type features
  - LSTM to perform temporal modeling
  - DNN layers for better discrimination

[T.N. Sainath et al, ICASSP 2015]

CLDNN

- Input x_t is a 40-dimensional log-mel feature
- Frequency convolution (fConv) [Sainath, ICASSP 2013]:
  - 8x1 filter, 256 outputs, pool by 3 without overlap
  - 8x256 output fed into a linear low-rank layer
- LSTM layer [H. Sak, Interspeech 2014]:
  - 2-3 layers
  - 832 cells/layer with 512 projection layer
  - Unroll for 20 time steps
- DNN layer:
  - 1 1,024 Relu layer
  - 1 linear low-rank layer with 512 outputs



Experimental Details

- Initial experiments to explore CLDNN architecture on 300K clean utterances (~200 hrs) Voice Search Task
- CLDNN details:
  - 40-dimensional log-mel filterbank features
  - Networks trained using ASGD with DistBelief [Dean, NIPS 2012]
  - 13,522 output targets
  - Initial experiments run with 2 LSTM layers
- Decoding details:
  - Clean Test Set with 30,000 utterances (~20 hrs)
  - Results always reported after Cross-Entropy training and Sequence training (when noted)

CNN + LSTM

- LSTM baseline WER=18.0
- Improvements by adding CNN layers before LSTM help but saturates after 1 layers
- Reducing spectral variations helps with temporal modeling

# CNN Layers	WER
0	18.0 (LSTM)
1	17.6
2	17.6

Google

LSTM + DNN

- Improvements by adding DNN layers after LSTM help but saturates after 2 layers
- Results illustrate the benefit of creating a more discriminative space with DNN layers after temporal modeling with the LSTM

# DNN Layers	WER
0	18.0 (LSTM)
1	17.8
2	**17.6**
3	17.6

CLDNN

- Gains from adding CNN layers before LSTM and DNN layers after LSTM are complementary
- Overall, CLDNN achieves a 4% relative improvement in WER over the LSTM

Method	WER
LSTM	18.0
CNN+LSTM	17.6
LSTM+DNN	17.6
CLDNN	**17.3**

Investigations on Larger Data Sets

- Initial experiments with CLDNNs on 200 hrs were just to get a quick understanding of CLDNNs
- We provide further analysis of LSTMs and CLDNNs on a larger test set trained on 3M noisy utterances (~2,000 hrs)
- Models trained and evaluated in matched conditions, on a noisy set of 30,000 utterances (~20 hrs)

Additional LSTM Layers

- Are gains from CLDNNs coming because we just have extra layers?
- Increasing number of LSTM layers after 3 seems to saturate performance
- CLDNN performance also improves by increasing number of LSTM layers

Method	LSTM WER	CLDNN WER
LSTM – 2 layers	17.1	16.3
LSTM – 3 layers	16.6	**16.0**
LSTM – 4 layers	16.6	16.2

Effect of Context

- CNNs typically use log-mel feature surrounded by temporal context
- Can the LSTM capture the temporal context alone? YES
- Lack of need for temporal input context simplifies CLDNN

Input Context	WER
l=0,r=0	**16.0**
l=10,r=0	**16.0**

Final Results: 16kHz Clean and Noisy Voice Search

- CLDNN is 1 conv, 3 LSTM, 1 DNN layer
- Models trained on 16 kHz Clean, 3M utterances, results on Clean
- CLDNN shows a **5% relative improvement** in WER

Method	WER - Seq
LSTM	13.2
CLDNN	**12.6**

- Training on 16 kHz MTR, 3M utterances, results on MTR
- CLDNN shows a **4% relative improvement** in WER

Method	WER - Seq
LSTM	14.5
CLDNN	**13.9**

Final Results: 8kHz Clean and MTR Voice Search

- Models trained on 8 kHz Clean, 3M utterances, results on Clean
- CLDNN shows a **8% relative improvement** over LSTM

Method	WER – Seq
LSTM	8.9
CLDNN	**8.2**

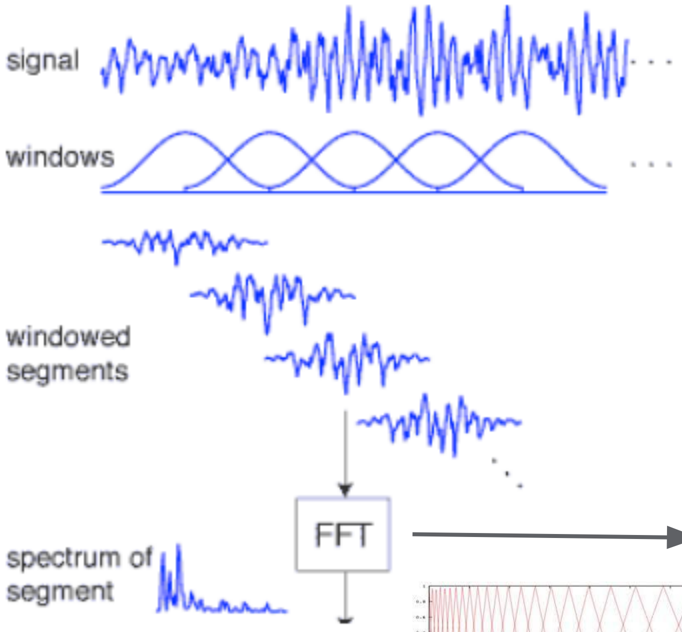
- Models trained on 8 kHz MTR, 3M utterances, results on MTR
- CLDNN shows a **7% relative improvement** over the LSTM

Method	WER – Seq
LSTM	18.8
CLDNN	**17.4**

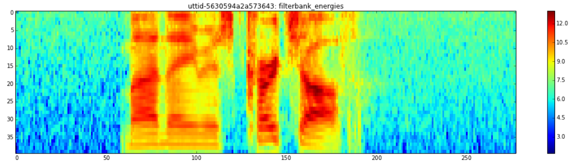
Outline

- Motivation
- CLDNNs
- Raw-waveform CLDNNs
- CTC

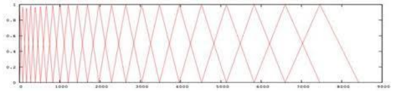
Sketch of the standard frontend



- Can we replace this processing with a neural network?
- Is there benefit learning this jointly with the rest of the network?

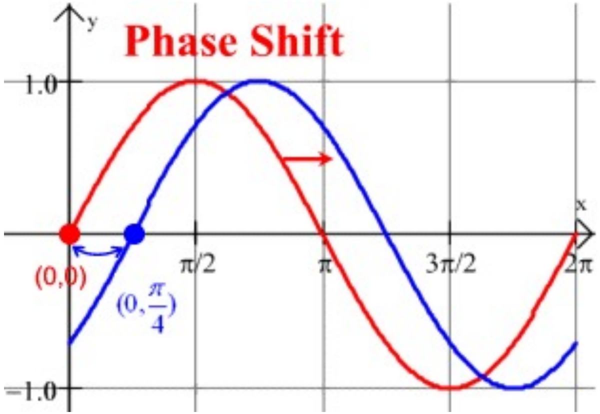


Mel Filterbank → Log

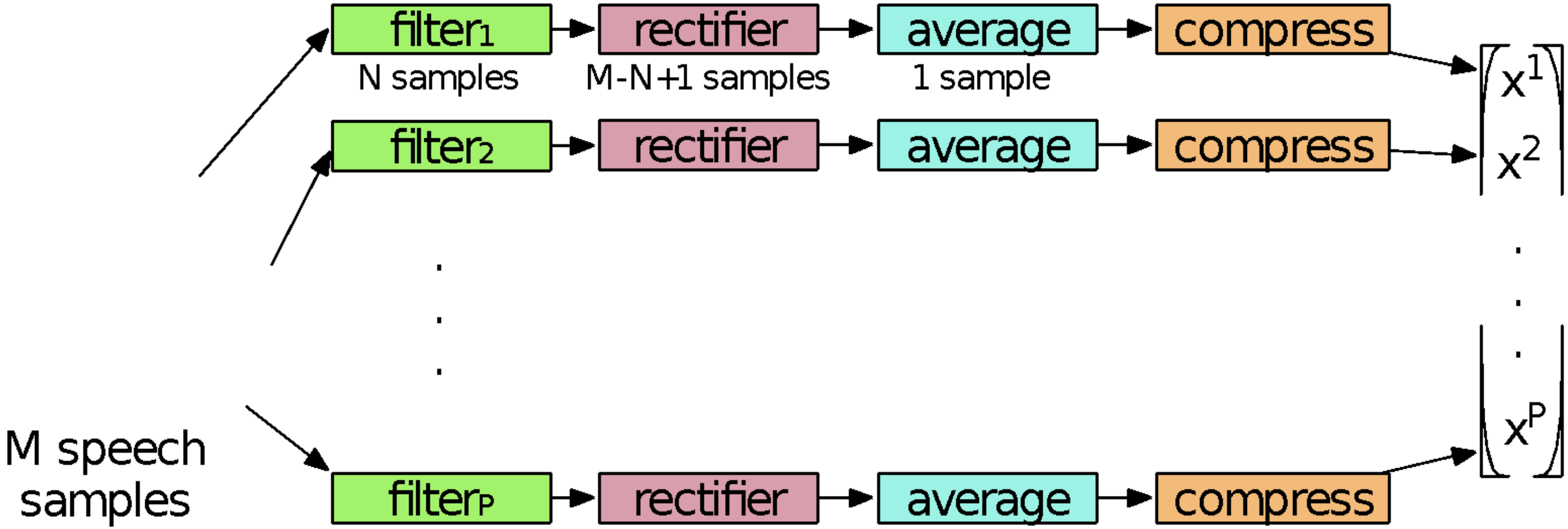


Difficulties of Modeling Raw-Waveform

- No past work has shown improvements with raw-waveform over a log-mel trained neural network [Jaitly 2011, Tuske 2014, Hoshen 2014, Palaz 2015]
- Perceptually and semantically identical sounds can appear at different phase shifts so its critical to model this

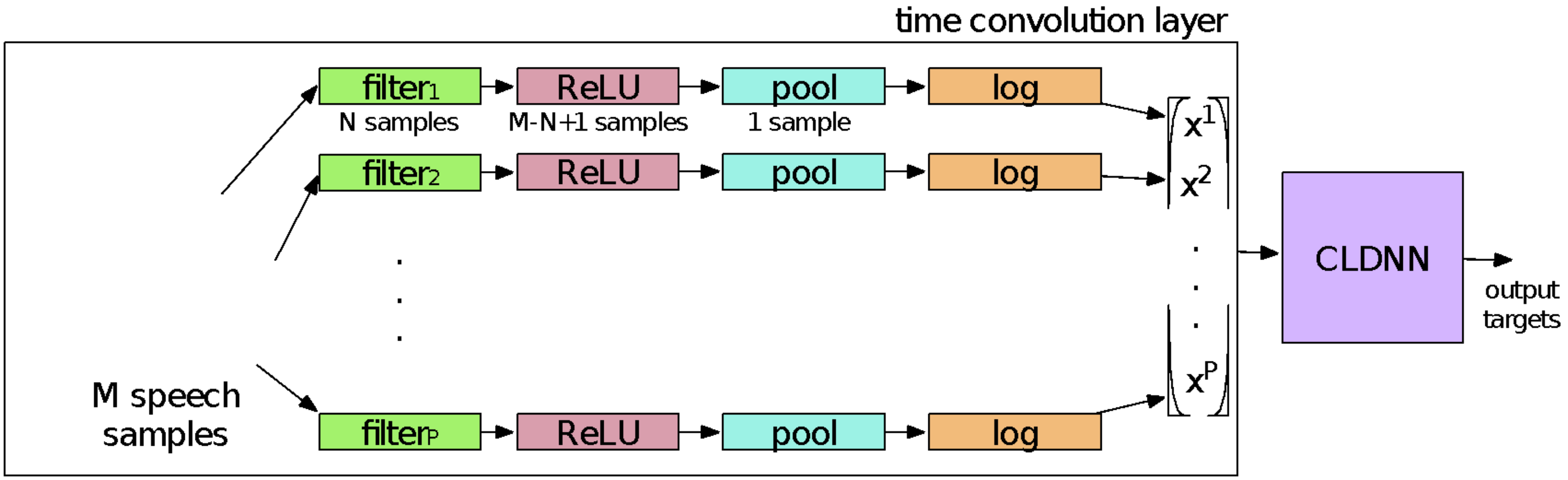


Inspiration from Gammatone Processing



All of these operations can be done with a neural network!

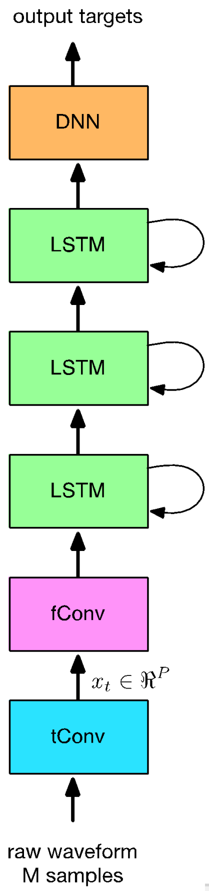
Time Domain Convolution



Frame-level features created by shifting window around M raw input samples by 10ms

Raw CLDNN

- Time convolution (tConv) produces a 1xP dimension frame
- CLDNN architecture same as [T.N. Sainath, ICASSP 2015]
- Frequency convolution (fConv):
  - 8x1 filter, 256 outputs, pool by 3 without overlay
  - 8x256 output fed into a linear low-rank layer
- LSTM layer:
  - 3 layers
  - 832 cells/layer with 512 projection layer
- DNN layer:
  - 1 1,024 Relu layer
  - 1 linear low-rank layer with 512 outputs
- tConv and CLDNN layers trained jointly



Experimental Details

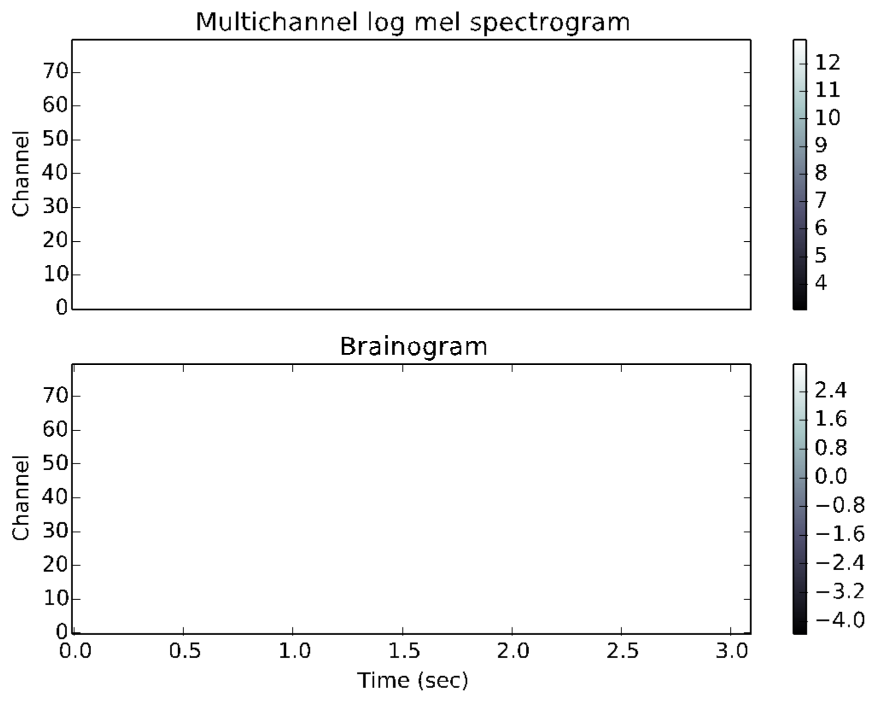
- Initial experiments to explore CLDNN architecture on 3M utterances (~2,000 hrs) Voice Search Task
- CLDNN details:
  - 40-dimensional log-mel filterbank features
  - Networks trained using ASGD with DistBelief [Dean, NIPS 2012]
  - 13,522 output targets
- Decoding details:
  - Test Set with 30,000 utterances (~20 hrs)
  - Results always reported after Cross-Entropy training and Sequence training (when noted)

Initial Results

A. Pooling in time to reduce temporal variations is important
B. Using a gammatone initalization helps slightly
C. Not training time-convolution layer is slightly worse, showing importance of learning filters for the task at hand

Label	Time Convolution Filter Size N (ms)	Input Window Size M (ms)	Filter Initalization	WER
A	400 (25ms)	400 (25ms)	random	19.9
	400	560 (35ms)	random	16.4
B	400	560	gammatone	**16.2**
C	400	560	gammatone untrained	16.4

Plot of Learned Features



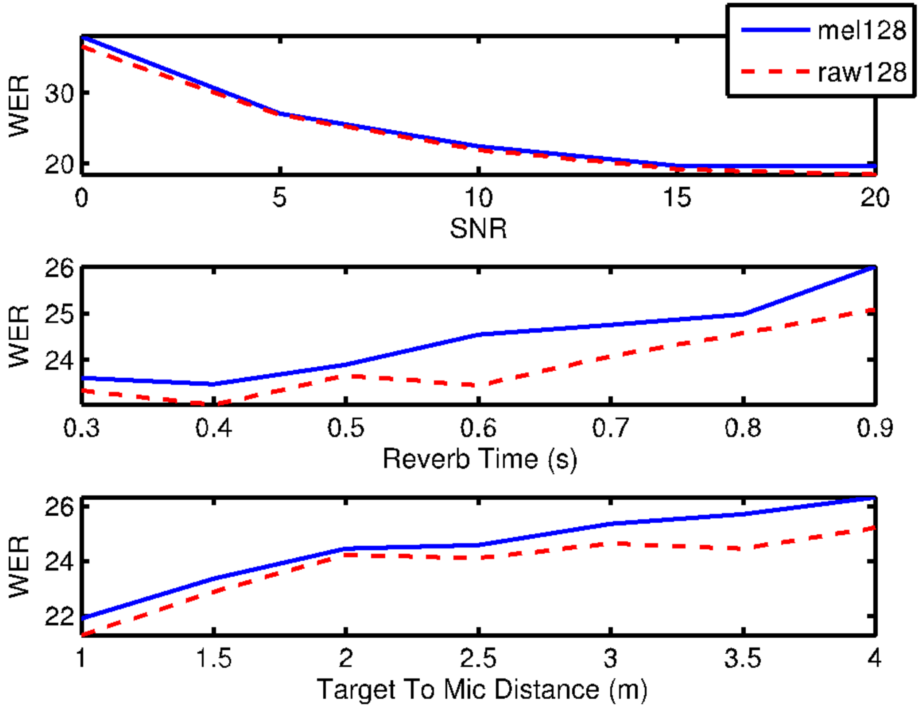
Learned features seem to look sensible and have a time-frequency representation

Comparison to Log-mel

- All results reported with same number of filters *P=40*
- This is the first time raw-waveform performance has match/improved over log-mel
- Let's look at why….

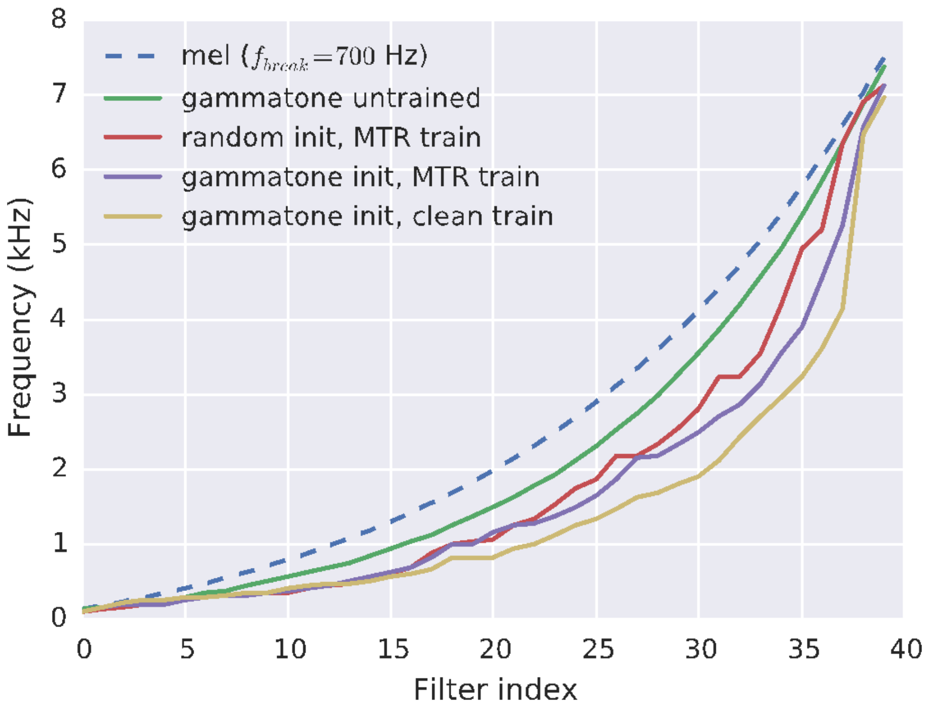
Method	Feature	WER-CE	WER-Seq
Clean	Log-mel	14.0	12.8
Clean	Raw	**13.7**	**12.7**
MTR ~ 20dB	Log-mel	16.2	**14.2**
MTR ~ 20 dB	Raw	16.2	**14.2**
MTR ~ 12 dB	Log-mel	25.2	20.7
MTR ~ 12 dB	Raw	**23.5**	**19.4**

WER Breakdown



Magnitude Response of Learned Filters

- Network seems to learn auditory-like filterbanks of bandpass filters
- Bandwidth increases with center frequency
- Learned filters give more resolution in lower frequencies
- Filterbank learning adapts to the data its trained on



Removing Convolutional Layers

- Analyze results for different CxLyDz architectures
- Log-mel and raw-waveform match in performance if we remove frequency convolution layers (2)
- No difference in performance when randomly initializing time-convolution layer
- Frequency convolution layer requires ordering of features coming out of time convolution layer

	Feature	Model	WER
(1)	log-mel	C1L3D1	16.2
	raw	C1L3D1, gammatone init	16.2
	raw	C1L3D1, rand init	16.4
(2)	log-mel	L3D1	16.5
	raw	L3D1, gammatone init	16.5
	raw	L3D1 rand init	16.5

Removing LSTM Layers

- Once we reduce LSTM layers to one (4) or none (5), log-mel performs better than raw-waveform
- Time convolution layer helps to reduce variations in time/phase shifts but cannot provide invariance on all relevant time scales
- LSTMs further helps to model variations across time frames

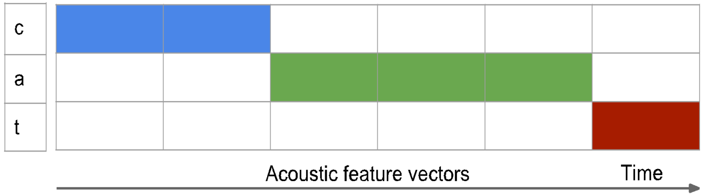
	Feature	Model	WER
(3)	log-mel	C1L2D1	16.6
	raw	C1L2D1	16.6
(4)	log-mel	C1L1D1	17.3
	raw	C1L1D1	17.8
(5)	log-mel	D6	22.3
	raw	D6	23.2

Outline

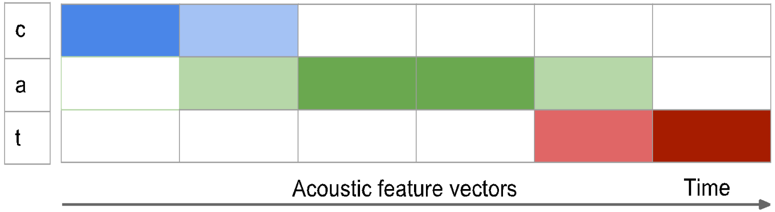
- Motivation
- CLDNNs
- Raw-waveform CLDNNs
- CTC

Acoustic Frame Labeling

- Training conventional DNN/RNN models require target labels for acoustic frames
- Acoustic modeling units / labels: HMM states, context dependent (CD), context independent (CI) phones...
- Hard labels / Viterbi alignment

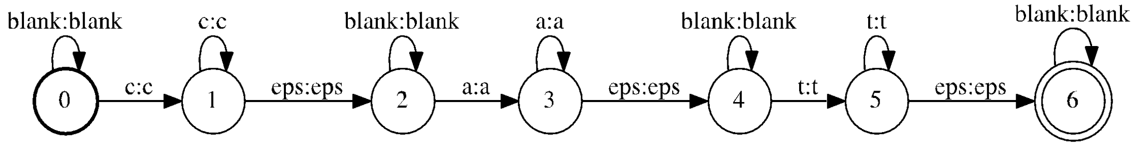


- Soft labels / Baum-Welch alignment (Forward-backward algorithm)



Connectionist Temporal Classification

- Sequence labeling technique using RNNs (Graves, 2006)
- Bidirectional CTC LSTM RNN models for handwriting recognition (Graves et al., 2009), phone recognition (Graves, Mohamed and Hinton, 2013)
- Align input sequences *x1, x2, …., xT*
  with target label sequences *l1, l2, …., lN*



- Not a conventional alignment: additional *blank* label
- "Collapse" label sequences by removing repeats and removing blanks
  "aaa----b-b--cccc--" → "abbc"
- CTC learns the acoustic model jointly with the alignment

Acoustic Frame Labeling with CTC vs. Cross-Entropy

Cross-Entropy

- CE tries to maximize the correct class at each frame with a frame level alignment

$$\mathcal{L}_{CE} = -\sum_{(\boldsymbol{x},\boldsymbol{l})} \sum_{t=1}^{|\boldsymbol{x}|} \sum_{l} \delta(l, \boldsymbol{l}_t) \log y_l^t.$$

- Gradient wrt inputs to softmax a^t

$$\frac{\partial \mathcal{L}(\boldsymbol{x}, \boldsymbol{l})}{\partial a_l^t} = y_l^t - \delta(l, \boldsymbol{l}_t)$$

CTC

- Define **z**^{l} as the lattice encoding all possible alignments of **x** with ***l***
- CTC loss

$$\mathcal{L}_{CTC} = -\sum_{(\boldsymbol{x},\boldsymbol{l})} \ln p(\boldsymbol{z}^{\boldsymbol{l}}|\boldsymbol{x}) = -\sum_{(\boldsymbol{x},\boldsymbol{l})} \mathcal{L}(\boldsymbol{x}, \boldsymbol{z}^{\boldsymbol{l}})$$

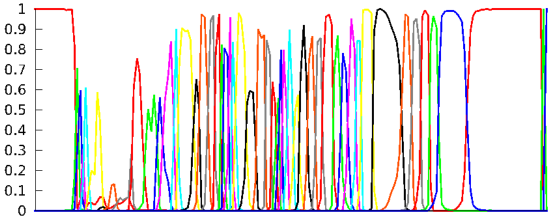
- Probability for correct labelings $p(\boldsymbol{z}^l|\boldsymbol{x})$ computed via forward-backward

$$p(\boldsymbol{z}^{\boldsymbol{l}}|\boldsymbol{x}) = \sum_{u=1}^{|\boldsymbol{z}^{\boldsymbol{l}}|} \alpha_{x,z^l}(t,u)\beta_{x,z^l}(t,u)$$

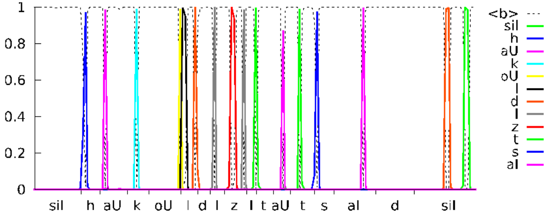
- Gradient

$$\frac{\partial \mathcal{L}(\boldsymbol{x}, \boldsymbol{z}^{\boldsymbol{l}})}{\partial a_l^t} = y_l^t - \frac{1}{p(\boldsymbol{z}^{\boldsymbol{l}}|\boldsymbol{x})} \sum_{u \in \{u: \boldsymbol{z}_u^{\boldsymbol{l}} = l\}} \alpha_{x,z^l}(t,u)\beta_{x,z^l}(t,u)$$

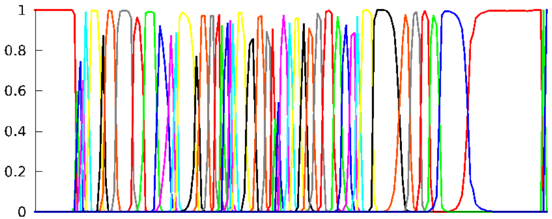
Posteriors for CE and CTC Training



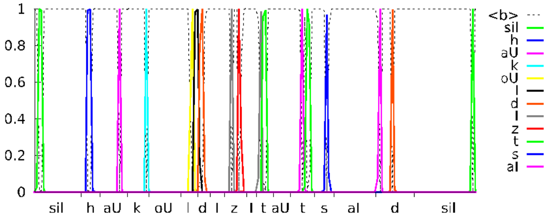
(c) unidirectional CD state CE



(g) unidirectional phone CTC



(e) bidirectional CD state CE



(i) bidirectional phone CTC

Experimental Details

- Initial experiments to explore CLDNN architecture on 3M clean 8kHz utterances (~2,000 hrs) Voice Search Task
- Training details:
  - 40-dimensional log-mel filterbank features
  - Explore unidirectional and bi-directional LSTMs
  - Networks trained using ASGD with DistBelief [Dean, NIPS 2012]
  - 13,522 output targets for conventional models
  - 41 phones for CTC models
- Decoding details:
  - Clean 8kHz Test Set with 30,000 utterances (~20 hrs)
  - Results reported after CE and Seq training

CTC Results for LSTM RNN Acoustic Models

[H. Sak et al, ICASSP 2015]

- LSTM RNN architecture from [H. Sak et al, Interspeech 2014]
- For unidirectional models, LSTM CTC with phone labels comes very close to LSTM with with fixed CD state alignment

Alignment	Label	CE	Seq
Fixed	Phone	13.2	-
Fixed	CD state	11.0	8.9
CTC	Phone	10.5	9.4

CTC Results with Bidirectional Modelling

- For bidirectional models, LSTM CTC with phone labels outperforms LSTM with with fixed CD state alignment
- With CTC, we can remove the complexity of CD states and the need for an existing alignment!

Alignment	Label	CE	Seq
Fixed	Phone	11.0	-
Fixed	CD state	9.7	9.1
CTC	Phone	9.5	8.5

Conclusions

- Removing assumptions within the speech pipeline with "neural-network" inspired models helps to improve performance
- Feature representation
  - Modeling directly from the raw waveform removes the need for complex front-end
- Acoustic Model
  - CLDNNs uses convolutional layers to model spectral variations, LSTMs for temporal variations and DNNs for discrimination
- CD states and alignments
  - CTC removes the need for alignments and CD phones
- Future work will look at combining raw waveform CLDNNs and CTC

References

- Modeling with CLDNNs:
  - T. N. Sainath, O. Vinyals, A. Senior and H. Sak, "Convolutional, Long Short-Term Memory, Fully Connected Deep Neural Networks," in Proc. ICASSP 2015.
- Frontend with Raw-waveform CLDNN:
  - T. N. Sainath, R. J. Weiss, A. Senior, K. W. Wilson and O. Vinyals, "Learning the Speech Front-end with Raw Waveform CLDNNs," to appear in Proc. Interspeech 2015.
- Acoustic Labeling with CTC:
  - H. Sak, A. Senior, K. Rao, O. Irsoy, A. Graves, F. Beaufays, J. Schalkwyk, “Learning Acoustic Frame Labeling for Speech Recognition with Recurrent Neural Networks,” in Proc. ICASSP 2015.

Questions

?