UNIVERSITY OF CAMBRIDGE

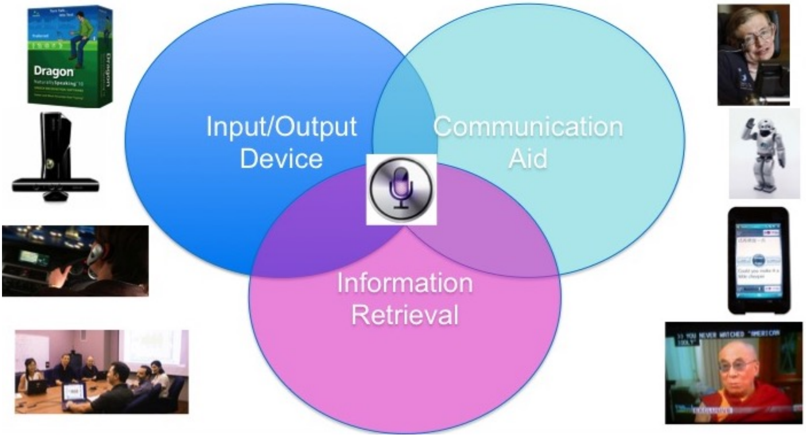
Deep Learning for Speech Recognition

Mark Gales

July 2017

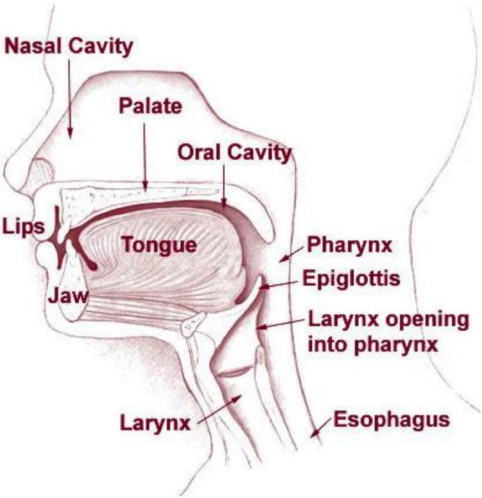
Apple Siri (2011)

Speech Application Areas

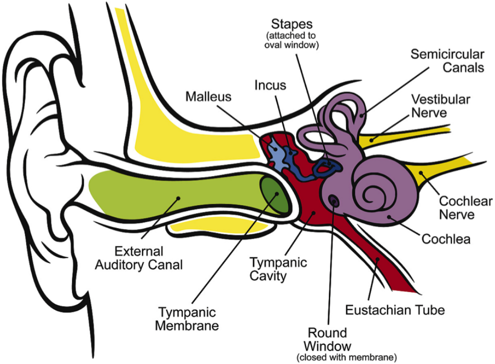


Speech Processing: Proof of Concept

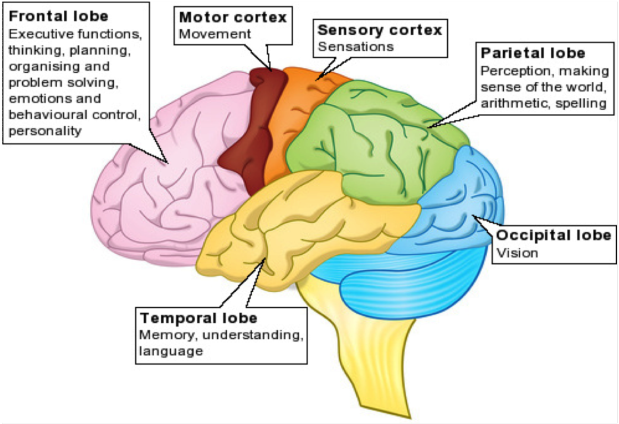
Speech Production (Synthesis)



Speech Perception (Recognition)

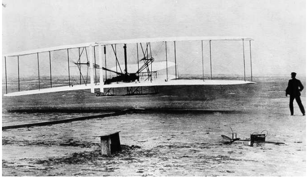


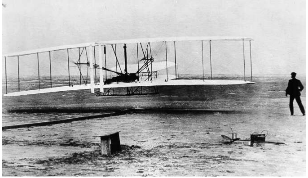
Speech Understanding



Should Speech Recognisers have Ears?

Should Speech Recognisers have Ears?

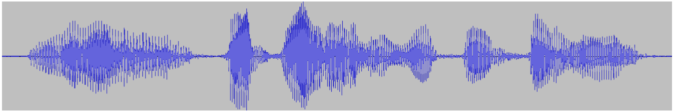




No - I'm an Engineer!

Speech Recognition

Speech Recognition

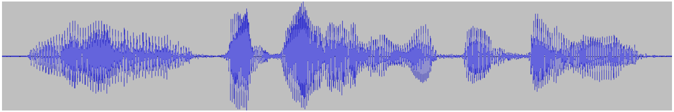


Waveform

ya uphethiloli wona usuwuthengile

Words

Speech Recognition (Traditional)



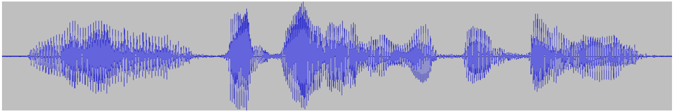
Waveform

Features

ya uphethiloli wona usuwuthengile

Words

Speech Recognition (Traditional)



Waveform

Features

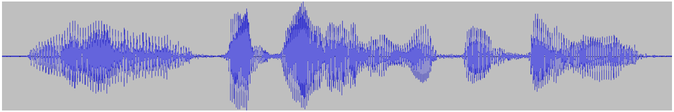
/w/ /O/ /n/ /a/

Phones

ya uphethiloli wona usuwuthengile

Words

Speech Recognition (Traditional)



Waveform

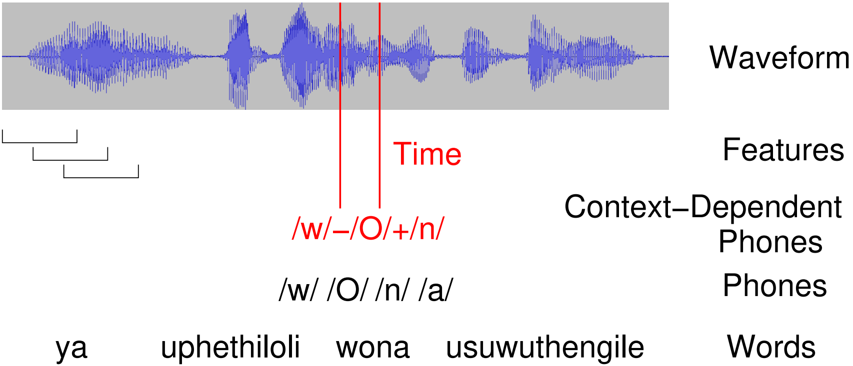
Features

/w/–/O/+/n/ — Context–Dependent Phones

/w/ /O/ /n/ /a/ — Phones

ya uphethiloli wona usuwuthengile — Words

Speech Recognition (Traditional)



Sequence-to-Sequence Modelling

- Sequence-to-sequence modelling central to speech/language:
  - machine translation:

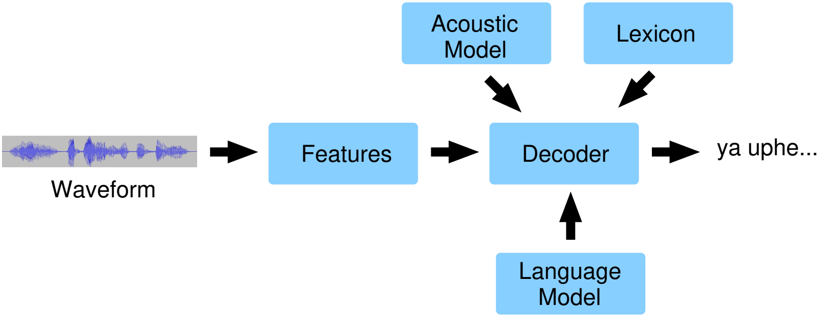
    word sequence (discrete) $\rightarrow$ word sequence (discrete)
  - speech synthesis:

    word sequence (discrete) $\rightarrow$ waveform (continuous)
  - speech recognition:

    waveform (continuous) $\rightarrow$ word sequence (discrete)

- The sequence lengths on either side can differ
  - waveform sampled at 10ms/5ms frame-rate - T-length $\boldsymbol{x}_{1:T}$
  - word/token sequences - L-length $\boldsymbol{\omega}_{1:L}$

Speech Recognition Framework (Traditional)



- Acoustic model: likelihood model generating observed features
- Language model: probability of any word sequence
- Lexicon: maps words to sub-word units (phones)

Generative Models [2, 3]

- Consider two sequences (note $L \leq T$):
  - features: $\boldsymbol{x}_{1:T} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T\}$
  - words: $\boldsymbol{\omega}_{1:L} = \{\omega_1, \omega_2, \ldots, \omega_L\}$
- Consider generative model

$$p(\omega_{1:L}, \boldsymbol{x}_{1:T}) \quad = \quad P(\omega_{1:L})p(\boldsymbol{x}_{1:T}|\omega_{1:L})$$

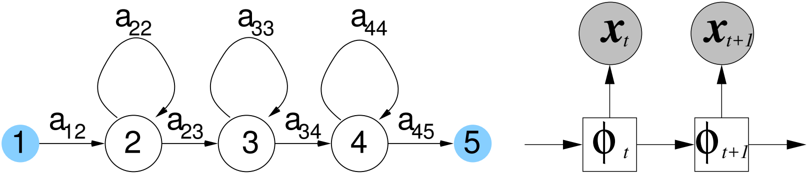
  - $P(\omega_{1:L})$: language model
  - $p(\boldsymbol{x}_{1:T}|\boldsymbol{\omega}_{1:L})$: acoustic model

Language Model: N-grams [15]

<s>	the	cat	sat	on	the	mat	</s>
<s>	the	cat	sat	on	the	mat	</s>
<s>	the	cat	sat	on	the	mat	</s>
<s>	the	cat	sat	on	the	mat	</s>
<s>	the	cat	sat	on	the	mat	</s>
<s>	the	cat	sat	on	the	mat	</s>
<s>	the	cat	sat	on	the	mat	</s>
<s>	the	cat	sat	on	the	mat	</s>

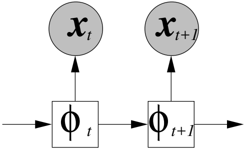
$$P(\boldsymbol{\omega}_{1:L}) = \prod_{i=1}^{L} P(\omega_i | \boldsymbol{\omega}_{1:i-1}) \approx \prod_{i=1}^{L} P(\omega_i | \boldsymbol{\omega}_{i-N+1:i-1})$$

Acoustic Model: Hidden Markov Models [1, 8, 23]



- HMMs standard model for many year (1970s-2010s)
  - each (context-dependent) phone modelled by an HMM
  - typically 3-emitting state topology, left-right
  - non-emitting (end) states used for "gluing" models together
- $\phi_{1:T}$ is the T-length state-sequence
  - ϕ_t indicates the HMM-state at time instance t

- Important sequence model: hidden Markov model (HMM)
  - an example of a dynamic Bayesian network (DBN)



- discrete latent variables
  - ϕ_t describes discrete state-space
  - conditional independence assumptions

$$P(\phi_t|\boldsymbol{\phi}_{1:t-1}) = P(\phi_t|\phi_{t-1})$$
$$p(\boldsymbol{x}_t|\boldsymbol{x}_{1:t-1}, \boldsymbol{\phi}_{1:t}) = p(\boldsymbol{x}_t|\phi_t)$$

- The likelihood of the data is

$$p(\boldsymbol{x}_{1:T}|\boldsymbol{\omega}_{1:L}) = \sum_{\boldsymbol{\phi}_{1:T} \in \boldsymbol{\Phi}_{\boldsymbol{\omega}_{1:L}}} \left(\prod_{t=1}^{T} p(\boldsymbol{x}_t|\phi_t) P(\phi_t|\phi_{t-1}) \right)$$

Decoding (Traditional) [23]

- Use Bayes' Decision Rule

$$\begin{aligned}\hat{\omega} &= \arg\max_{\omega}\left\{P(\omega|\mathbf{x}_{1:T})\right\} \\ &= \arg\max_{\omega}\left\{P(\omega,\mathbf{x}_{1:T})\right\} \\ &= \arg\max_{\omega}\left\{P(\omega)p(\mathbf{x}_{1:T}|\omega)\right\}\end{aligned}$$

  - need to efficiently search over all possible word sequences
- Viterbi decoding used for efficiency with HMMs & N-grams
  - leverages model conditional independence assumptions

Deep Learning and Recurrent Neural Networks

What is Deep Learning?

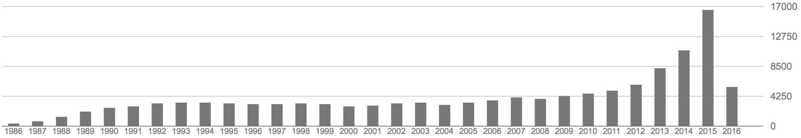
From Wikipedia:

> *Deep learning is a branch of machine learning based on a set of algorithms that attempt to model high-level abstractions in data by using multiple processing layers, with complex structures or otherwise, composed of multiple non-linear transformations.*

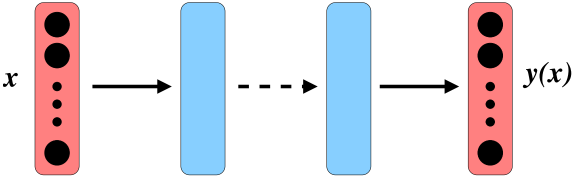
What is Deep Learning?

From Wikipedia:

> *Deep learning is a branch of machine learning based on a set of algorithms that attempt to model high-level abstractions in data by using multiple processing layers, with complex structures or otherwise, composed of multiple non-linear transformations.*



Deep Neural Networks [13]

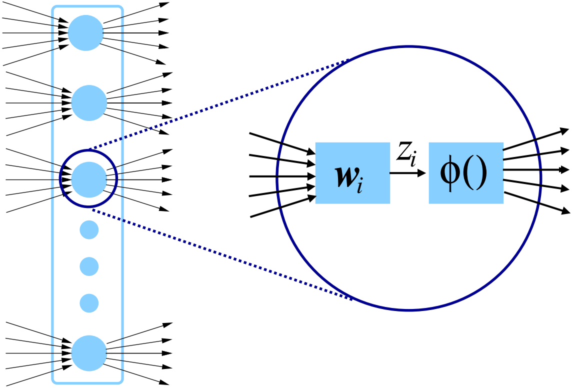


- General mapping process from input $\boldsymbol{x}$ to output $\boldsymbol{y}(\boldsymbol{x})$

$$\boldsymbol{y}(\boldsymbol{x}) = \mathcal{F}(\boldsymbol{x})$$

  - deep refers to number of hidden layers
- Output from the previous layer connected to following layer:
  - $\boldsymbol{x}^{(k)}$ is the input to layer k
  - $\boldsymbol{x}^{(k+1)} = \boldsymbol{y}^{(k)}$ the output from layer k

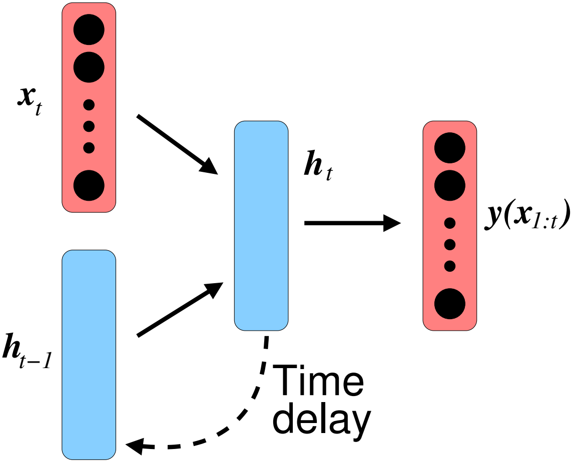
Neural Network Layer/Node



- General form for layer k:

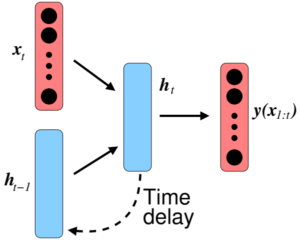
$$y_i^{(k)} = \phi(\mathbf{w}_i' \mathbf{x}^{(k)} + b_i) = \phi(z_i^{(k)})$$

Recurrent Neural Networks [19, 18]



Recurrent Neural Networks

- Consider a causal sequence of observations $\boldsymbol{x}_{1:t} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t\}$



- Introduce recurrent units

$$\begin{aligned}\boldsymbol{h}_t &= \mathbf{f}^{\mathrm{h}}\left(\mathbf{W}_{\mathrm{h}}^{\mathrm{f}}\boldsymbol{x}_t + \mathbf{W}_{\mathrm{h}}^{\mathrm{r}}\boldsymbol{h}_{t-1} + \boldsymbol{b}_{\mathrm{h}}\right)\\ \boldsymbol{y}(\boldsymbol{x}_{1:t}) &= \mathbf{f}^{\mathrm{f}}\left(\mathbf{W}_{\mathrm{y}}\boldsymbol{h}_t + \boldsymbol{b}_{\mathrm{y}}\right)\end{aligned}$$

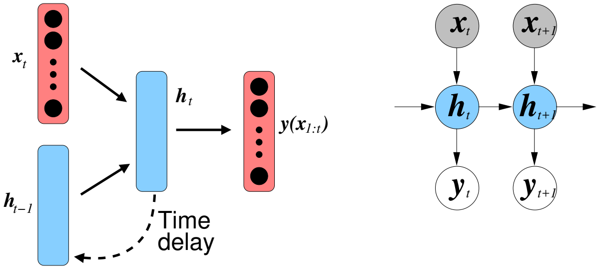
  - $\boldsymbol{h}_t$ history vector at time t

- Uses approximation to model history of observations

$$\mathcal{F}(\boldsymbol{x}_{1:t}) = \mathcal{F}(\boldsymbol{x}_t, \boldsymbol{x}_{1:t-1}) \approx \mathcal{F}(\boldsymbol{x}_t, \boldsymbol{h}_{t-1}) \approx \mathcal{F}(\boldsymbol{h}_t) = \boldsymbol{y}(\boldsymbol{x}_{1:t})$$

  - network has (causal) memory encoded in history vector ($\boldsymbol{h}_t$)

RNN: Dynamic Bayesian Network



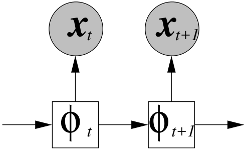
- Maps between two sequences $\boldsymbol{x}_{1:T} \rightarrow \boldsymbol{y}_{1:T}$
- Figure on right is unwrapped in time
  - shows dependencies - shaded blue are deterministic mappings
- Seen similar models - HMMs, CRFs, SSVMs ..
  - doesn't handle sequence length mappings in ASR

RNN: Variants and Extensions [21, 6, 14, 11, 22, 7]

- Extensions of standard RNN structure:
  - bi-directional RNN (depends on future and past)
  - latent-variable RNNs (continuous latent variables)

- Modification to the recurrent units (gating)
  - long-short term memory units (LSTMs)
  - gated recurrent units (GRUs)
  - highway connections (gating in time)

Acoustic Modelling



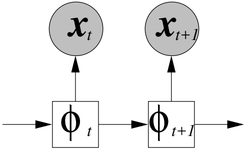
- Discrete latent variables
  - ϕ_t describes discrete state-space
  - conditional independence assumptions

$$P(\phi_t|\phi_{1:t-1}) = P(\phi_t|\phi_{t-1})$$
$$p(\mathbf{x}_t|\mathbf{x}_{1:t-1}, \phi_{1:t}) = p(\mathbf{x}_t|\phi_t)$$

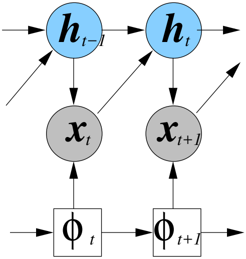
- The likelihood of the data is

$$\begin{aligned} p(\mathbf{x}_{1:T}|\boldsymbol{\omega}_{1:L}) &= \sum_{\phi_{1:T} \in \boldsymbol{\Phi}_{\boldsymbol{\omega}_{1:L}}} p(\mathbf{x}_{1:T}|\phi_{1:T}) P(\phi_{1:T}) \\ &= \sum_{\phi_{1:T} \in \boldsymbol{\Phi}_{\boldsymbol{\omega}_{1:L}}} \left(\prod_{t=1}^{T} p(\mathbf{x}_t|\phi_t) P(\phi_t|\phi_{t-1}) \right) \end{aligned}$$

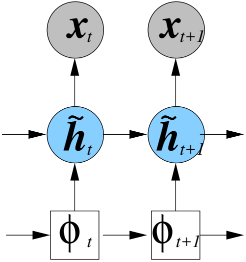
History Approximations and Inference



HMM



Finite State



Finite Feature

$\prod_{t=1}^{T} p(\boldsymbol{x}_t|\phi_t)$ $\quad$ $\prod_{t=1}^{T} p(\boldsymbol{x}_t|\phi_t, \boldsymbol{h}_{t-1})$ $\quad$ $\prod_{t=1}^{T} p(\boldsymbol{x}_t|\tilde{\boldsymbol{h}}_t)$

- Inference costs significantly different:
  - finite state: all past history observed - deterministic
  - finite feature: past history unobserved - depends on path

Acoustic Model Approximations

- HMM: simplest form of approximation

$$p(\boldsymbol{x}_{1:T}|\phi_{1:T}) \approx \prod_{t=1}^{T} p(\boldsymbol{x}_t|\phi_t)$$

- Finite State:

$$p(\boldsymbol{x}_{1:T}|\phi_{1:T}) \approx \prod_{t=1}^{T} p(\boldsymbol{x}_t|\phi_t, \boldsymbol{x}_{1:t-1}) \approx \prod_{t=1}^{T} p(\boldsymbol{x}_t|\phi_t, \boldsymbol{h}_{t-1})$$

- Finite Feature:

$$p(\boldsymbol{x}_{1:T}|\phi_{1:T}) \approx \prod_{t=1}^{T} p(\boldsymbol{x}_t|\phi_{1:t}) \approx \prod_{t=1}^{T} p(\boldsymbol{x}_t|\tilde{\boldsymbol{h}}_t)$$

"Likelihoods" [3]

- Deep learning can be used to estimate distributions
  - mixture density neural network (MDNN)
  - more often trained as a discriminative model
  - need to convert to a "likelihood"

"Likelihoods" [3]

- Deep learning can be used to estimate distributions
  - mixture density neural network (MDNN)
  - more often trained as a discriminative model
  - need to convert to a "likelihood"
- Most common form (for RNN acoustic model):

$$
\begin{aligned}
p(\boldsymbol{x}_t|\phi_t, \boldsymbol{h}_{t-1}) &= \frac{P(\phi_t|\boldsymbol{x}_t, \boldsymbol{h}_{t-1})p(\boldsymbol{x}_t|\boldsymbol{h}_{t-1})}{P(\phi_t|\boldsymbol{h}_{t-1})} \\
&\propto \frac{P(\phi_t|\boldsymbol{x}_t, \boldsymbol{h}_{t-1})}{P(\phi_t|\boldsymbol{h}_{t-1})} \\
&\approx \frac{P(\phi_t|\boldsymbol{x}_t, \boldsymbol{h}_{t-1})}{P(\phi_t)}
\end{aligned}
$$

  - $P(\phi_t|\boldsymbol{x}_t, \boldsymbol{h}_{t-1})$: modelled by a standard RNN
  - $P(\phi_t)$: state/phone prior probability

“Baseline” Acoustic Training Criteria

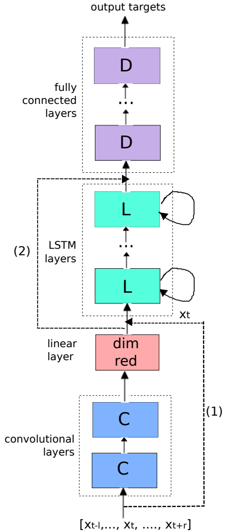
- Originally generative models (GMM-HMM systems) used ML

$$\begin{aligned}\mathcal{F}_{\texttt{ml}} &= \log\left(p(\boldsymbol{x}_{1:T}|\boldsymbol{\omega}_{\texttt{ref}})\right)\\ &= \log\left(\sum_{\phi_{1:T}\in\boldsymbol{\Phi}_{\boldsymbol{\omega}_{\texttt{ref}}}} p(\boldsymbol{x}_{1:T}|\phi_{1:T})P(\phi_{1:T})\right)\end{aligned}$$

- Neural networks: Cross-Entropy with fixed alignment,

$$\begin{aligned}\mathcal{F}_{\text{ce}} &= -\sum_{t=1}^{T}\log\left(P(\hat{\phi}_t|\boldsymbol{x}_t,\boldsymbol{h}_{t-1})\right)\\ \hat{\phi}_{1:T} &= \arg\max_{\phi_{1:T}\in\boldsymbol{\Phi}_{\boldsymbol{\omega}_{\texttt{ref}}}}\left\{P(\phi_{1:T}|\boldsymbol{x}_{1:T})\right\}\end{aligned}$$

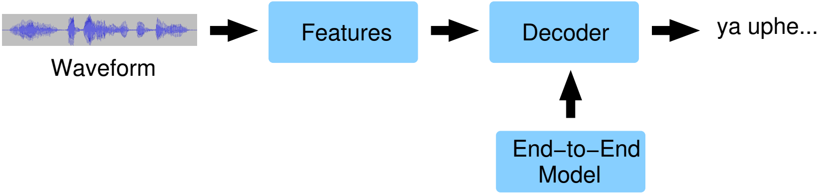
Example "Generative" Acoustic Model [20]



- Example Architecture from Google (2015)
  - C: CNN layer (with pooling)
  - L: LSTM layer
  - D: fully connected layer
- Two multiple layer "skips"
  - (1) connects input to LSTM input
  - (2) connects CNN output to DNN input
- Additional linear projection layer
  - reduces dimensionality
  - and number of network parameters!

Discriminative Models
(“End-to-End” Models)

Speech Recognition Framework



- Apply Bayes' Decision Rule

$$\hat{\omega} = \arg\max_{\omega} \left\{ P(\omega|\mathbf{x}_{1:T}) \right\}$$

- Directly train model to solve task ("speech-to-text")
  - single model trained
  - no separate acoustic and language models
- More complicated to incorporate additional LM data

Discriminative Models [2]

- Compute posterior of word sequence

$$P(\boldsymbol{\omega}_{1:L}|\boldsymbol{x}_{1:T}) = \sum_{\phi_{1:T} \in \boldsymbol{\Phi}_{\boldsymbol{\omega}_{1:L}}} P(\boldsymbol{\omega}_{1:L}|\phi_{1:T}) P(\phi_{1:T}|\boldsymbol{x}_{1:T})$$

Discriminative Models

- Compute posterior of word sequence

$$P(\boldsymbol{\omega}_{1:L}|\boldsymbol{x}_{1:T}) = \sum_{\phi_{1:T} \in \boldsymbol{\Phi}_{\boldsymbol{\omega}_{1:L}}} \overbrace{P(\boldsymbol{\omega}_{1:L}|\phi_{1:T})}^{1} P(\phi_{1:T}|\boldsymbol{x}_{1:T})$$

Discriminative Models

- Compute posterior of word sequence

$$P(\boldsymbol{\omega}_{1:L}|\boldsymbol{x}_{1:T}) = \sum_{\phi_{1:T} \in \boldsymbol{\Phi}_{\boldsymbol{\omega}_{1:L}}} \cancelto{1}{P(\boldsymbol{\omega}_{1:L}|\phi_{1:T})} P(\phi_{1:T}|\boldsymbol{x}_{1:T})$$

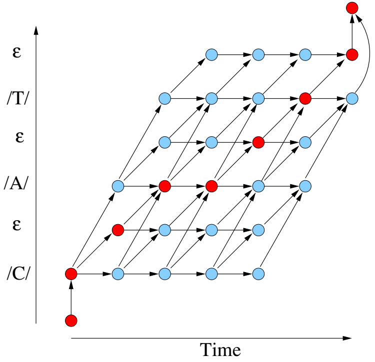
  - finite state RNNs used to model history/alignment

$$\begin{aligned} P(\phi_{1:T}|\boldsymbol{x}_{1:T}) &\approx \prod_{t=1}^{T} P(\phi_t|\boldsymbol{x}_{1:t}) \\ &\approx \prod_{t=1}^{T} P(\phi_t|\boldsymbol{x}_t, \boldsymbol{h}_{t-1}) \approx \prod_{t=1}^{T} P(\phi_t|\boldsymbol{h}_t) \end{aligned}$$

- Expression does not have a language model

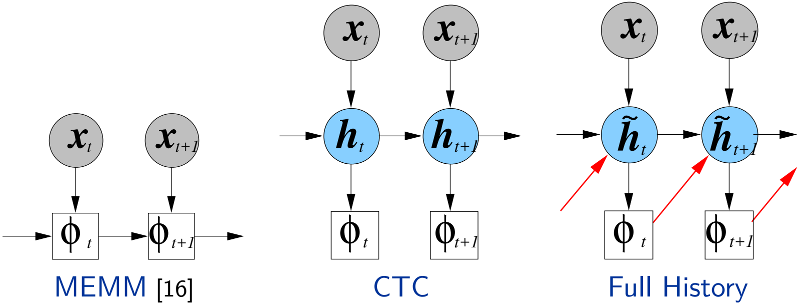
Connectionist Temporal Classification [10]

- CTC: discriminative model, no explicit alignment model
  - introduces a `blank` output symbol (ϵ)



- Consider word: `CAT`
  - Pronunciation: `/C/ /A/ /T/`
- Observe 7 frames
  - possible state transitions
  - example path:
    `/C/` ϵ `/A/` `/A/` ϵ `/T/` ϵ

Including State History?



- Interesting to consider state dependencies (right)

$$P(\phi_{1:T}|\mathbf{x}_{1:T}) \approx \prod_{t=1}^{T} P(\phi_t|\mathbf{x}_{1:t}, \phi_{1:t-1}) \approx \prod_{t=1}^{T} P(\phi_t|\tilde{\mathbf{h}}_t)$$

Nature of Targets

- One trend for discriminative models:

  Graphemes (letters) rather than context-dependent phones

- Take the example of the lexicon entry `cat`: `/k/ /a/ /t/`

`sil`	`k`	`a`	`t`	`sil`
`sil`	`sil-/k/+/a/`	`/k/-/a/+/t/`	`/a/-/t/+sil`	`sil`
`sil`	`sil-/c/+/a/`	`/c/-/a/+/t/`	`/a/-/t/+sil`	`sil`
`sil`	`c`	`a`	`t`	`sil`

- Can be run at the character level
  - no need to have a lexicon (hence no OOVs)
  - language model implicit by history vector (of features)

Discriminative Models and "Priors" [12]

- No language models in (this form of) discriminative model
  - in CTC the word history "captured" in frame history
  - no explicit dependence on state (word) history
- Treat as a product of experts (log-linear model): for CTC

$$P(\boldsymbol{\omega}_{1:L}|\boldsymbol{x}_{1:T}) = \frac{1}{Z(\boldsymbol{x}_{1:T})}\exp\left(\boldsymbol{\alpha}^{\mathrm{T}}\left[\begin{array}{c}\log\left(\sum_{\phi_{1:T}\in\boldsymbol{\Phi}_{\boldsymbol{\omega}_{1:L}}} P(\phi_{1:T}|\boldsymbol{x}_{1:T})\right)\\ \log\left(\tilde{P}(\boldsymbol{\omega}_{1:L})\right)\end{array}\right]\right)$$

  - $\boldsymbol{\alpha}$ trainable parameter (related to LM scale)
  - $\tilde{P}(\boldsymbol{\omega}_{1:L})$ standard "prior" (language) model
- Normalisation term not required in decoding
  - $\boldsymbol{\alpha}$ often empirically tuned

UNIVERSITY OF CAMBRIDGE

Encoder-Decoder Style Models

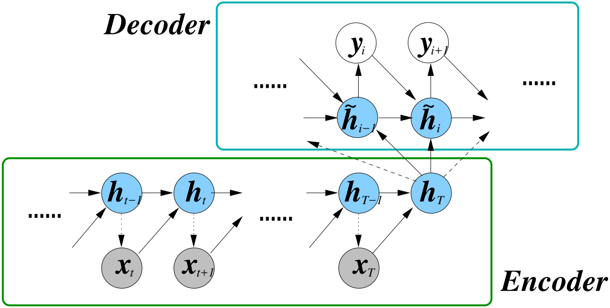
- Directly model relationship

$$
\begin{aligned}
P(\boldsymbol{\omega}_{1:L}|\boldsymbol{x}_{1:T}) &= \prod_{i=1}^{L} P(\omega_i|\boldsymbol{\omega}_{1:i-1}, \boldsymbol{x}_{1:T}) \\
&\approx \prod_{i=1}^{L} P(\omega_i|\boldsymbol{\omega}_{i-1}, \tilde{\boldsymbol{h}}_{i-2}, \boldsymbol{c})
\end{aligned}
$$

  - looks like an RNN LM with additional dependence on $\boldsymbol{c}$

$$\boldsymbol{c} = \phi(\boldsymbol{x}_{1:T})$$

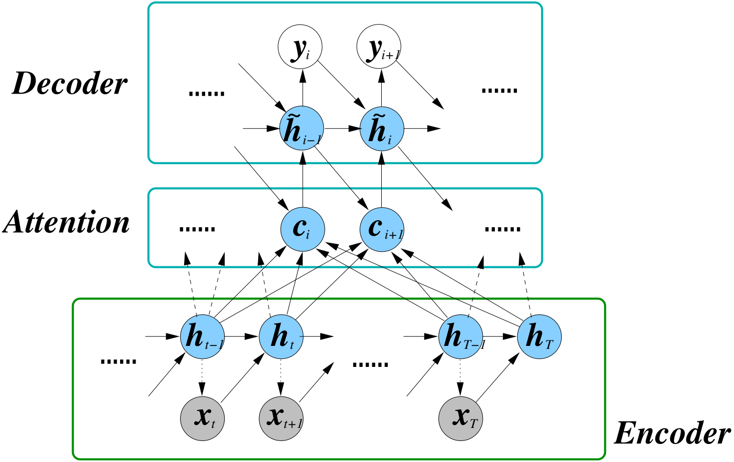
  - $\boldsymbol{c}$ is a fixed length vector - like a sequence kernel



- Simplest form is to use hidden unit from acoustic RNN/LSTM

$$\boldsymbol{c} = \phi(\boldsymbol{x}_{1:T}) = \boldsymbol{h}_T$$

  - dependence on context is global via $\boldsymbol{c}$ - possibly limiting


Decoder
......
y_i
y_{i+1}
......
$\tilde{h}_{i-1}$
$\tilde{h}_i$
Attention
......
c_i
c_{i+1}
......
h_{t-1}
h_t
h_{T-1}
h_T
......
......
x_t
x_{t+1}
x_T
Encoder

Attention-Based Models

- Introduce attention layer to system
  - introduce dependence on locality i

$$P(\boldsymbol{\omega}_{1:L}|\boldsymbol{x}_{1:T}) \approx \prod_{i=1}^{L} p(\omega_i|\boldsymbol{\omega}_{i-1}, \tilde{\boldsymbol{h}}_{i-2}, \boldsymbol{c}_i) \approx \prod_{i=1}^{L} p(\omega_i|\tilde{\boldsymbol{h}}_{i-1})$$

$$\boldsymbol{c}_i = \sum_{\tau=1}^{T} \alpha_{i\tau} \boldsymbol{h}_\tau; \quad \alpha_{i\tau} = \frac{\exp(e_{i\tau})}{\sum_{k=1}^{T} \exp(e_{ik})}, \quad e_{i\tau} = f^{\mathrm{e}}\left(\tilde{\boldsymbol{h}}_{i-2}, \boldsymbol{h}_\tau\right)$$

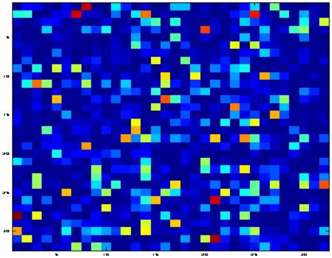
  - $e_{i\tau}$ how well position $i-1$ in input matches position τ in output
  - $\boldsymbol{h}_\tau$ is representation (RNN) for the input at position τ
- Attention can “wander” with large input size (T)
  - use a pyramidal network to reduce frame-rate for attention

Conclusions

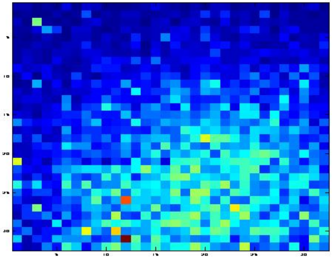
It's an interesting time!

- Deep learning integrated into standard speech toolkits
    - Kaldi, HTK etc
- Rich variety of models and topologies supported by:
    - large quantities of training data
    - GPU-based training (and parallel implementations)
    - array of software tools: TensorFlow, CNTK, Theano ...
- Most state-of-the-art still "generative"
    - but next conference in August ...

Network Interpretation [24]



Standard /ay/



Stimulated /ay/

- Deep learning usually highly distributed - hard to interpret
  - awkward to adapt/understand/regularise
  - modify training - add stimulation regularisation
  - improves ASR performance ...

Thank-you!

[1] L. Baum and J. Eagon, "An Inequality with Applications to Statistical Estimation for Probabilistic Functions of Markov Processes and to a Model for Ecology," *Bull Amer Math Soc*, vol. 73, pp. 360–363, 1967.

[2] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer Verlag, 2006.

[3] H. Bourlard and N. Morgan, "Connectionist speech recognition: A hybrid approach," 1994.

[4] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell," *CoRR*, vol. abs/1508.01211, 2015. [Online]. Available: http://arxiv.org/abs/1508.01211

[5] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," *CoRR*, vol. abs/1506.07503, 2015. [Online]. Available: http://arxiv.org/abs/1506.07503

[6] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[7] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, "A recurrent latent variable model for sequential data," *CoRR*, vol. abs/1506.02216, 2015. [Online]. Available: http://arxiv.org/abs/1506.02216

[8] M. Gales and S. Young, "The application of hidden Markov models in speech recognition," *Foundations and Trends in Signal Processing*, vol. 1, no. 3, 2007.

[9] A. Graves, "Sequence transduction with recurrent neural networks," *CoRR*, vol. abs/1211.3711, 2012. [Online]. Available: http://arxiv.org/abs/1211.3711

[10] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 369–376.

[11] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 6645–6649.

[12] G. E. Hinton, "Products of experts," in *Proceedings of the Ninth International Conference on Artificial Neural Networks (ICANN 99)*, 1999, pp. 1–6.

[13] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[15] F. Jelinek, *Statistical methods for speech recognition*, ser. Language, speech, and communication. Cambridge (Mass.), London: MIT Press, 1997.

[16] H.-K. Kuo and Y. Gao, "Maximum entropy direct models for speech recognition," *IEEE Transactions Audio Speech and Language Processing*, 2006.

[17] L. Lu, X. Zhang, K. Cho, and S. Renals, "A study of the recurrent neural network encoder-decoder for large vocabulary speech recognition," in *Proc. INTERSPEECH*, 2015.

[18] T. Robinson and F. Fallside, "A recurrent error propagation network speech recognition system," *Computer Speech & Language*, vol. 5, no. 3, pp. 259–274, 1991.

[19] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1," D. E. Rumelhart, J. L. McClelland, and C. PDP Research Group, Eds. Cambridge, MA, USA: MIT Press, 1986, ch. Learning Internal Representations by Error Propagation, pp. 318–362.

[20] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, long short-term memory, fully connected deep neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4580–4584.

[21] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[22] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Highway networks," *CoRR*, vol. abs/1505.00387, 2015. [Online]. Available: http://arxiv.org/abs/1505.00387

[23] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," vol. 13, no. 2, pp. 260–269, 1967.

[24] C. Wu, P. Karanasou, M. Gales, and K. C. Sim, "Stimulated deep neural network for speech recognition," in *Proceedings interspeech*, 2016.