Automatic Speech Recognition based on Neural Networks

Ralf Schlüter
Human Language Technology and Pattern Recognition
Lehrstuhl Informatik 6
Department of Mathematics, Computer Science and Natural Sciences
RWTH Aachen University

Preamble

- joint work with members of HLT & PR lab (Informatik 6):
  - acoustic modeling: Zoltan Tüske, Pavel Golik, Albert Zeyer, Patrick Doetsch, ...
  - language modeling: Martin Sundermeyer, Kazuki Irie, ...
  - cf. `hltpr.rwth-aachen.de/web/Publications`

- toolkits used for results presented here are available on our web site:
  - RASR: RWTH Automatic Speech Recognition toolkit (also handwriting)
  - RWTHLM: RWTH neural network based Language Modeling toolkit (esp. LSTM)
  - RETURNN: RWTH Extensible Training for Universal Recurrent Neural Networs (**new!**)
  - ...
  - cf. `hltpr.rwth-aachen.de/web/Software`

Overview

Outline

Sequence Classification

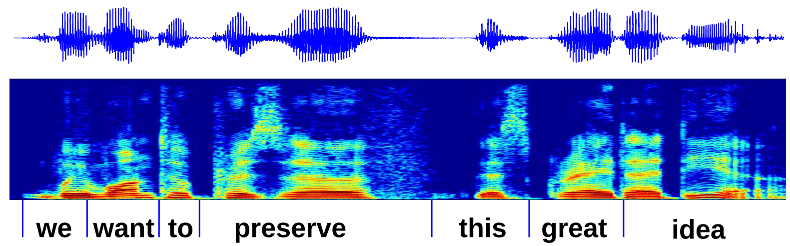
Tasks for machine learning:

- automatic speech recognition
- text image recognition
- machine translation

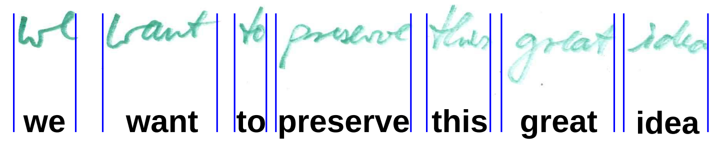
Most general case:

- input sequence:
  $X := x_1 \ldots x_t \ldots x_T$
- output sequence (of unknown length N):
  $W := w_1 \ldots w_n \ldots w_N$
- true distribution $pr(W|X)$
  (can be extremely complex!)

Speech Recognition

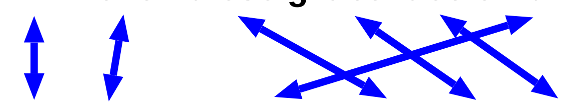


Text Image Recognition



Machine Translation

wir wollen diese große Idee erhalten



we want to preserve this great idea

Sequence Decision Rule

- performance measure or loss function $L[\widetilde{W}, W]$ (e.g. edit distance) between true output sequence $\widetilde{W}$ and hypothesized output sequence W.
- Bayes decision rule minimizes expected loss:

$$X \rightarrow \overline{W}(X) := \arg\min_{W} \left\{ \sum_{\widetilde{W}} pr(\widetilde{W}|X) \cdot L[\widetilde{W}, W] \right\}$$

- Standard decision rule uses sequence-level loss:

$$X \rightarrow \widehat{W}(X) := \arg\max_{W} \left\{ pr(W|X) \right\}$$

  Since [Bahl & Jelinek+ 1983], this simpified Bayes decision rule is widely used for speech recognition, handwriting recognition, machine translation, ...
- Works well, as often both decision rules coincide.
  This can be proven under certain conditions [Schlüter & Nussbaum+ 2012], e.g.:

$$L[W, \widetilde{W}] \text{ is a metric, and } \max_{W} \; pr(W|X) \geq 0.5 \quad \Rightarrow \quad \overline{W}(X) = \widehat{W}(X)$$

Outline

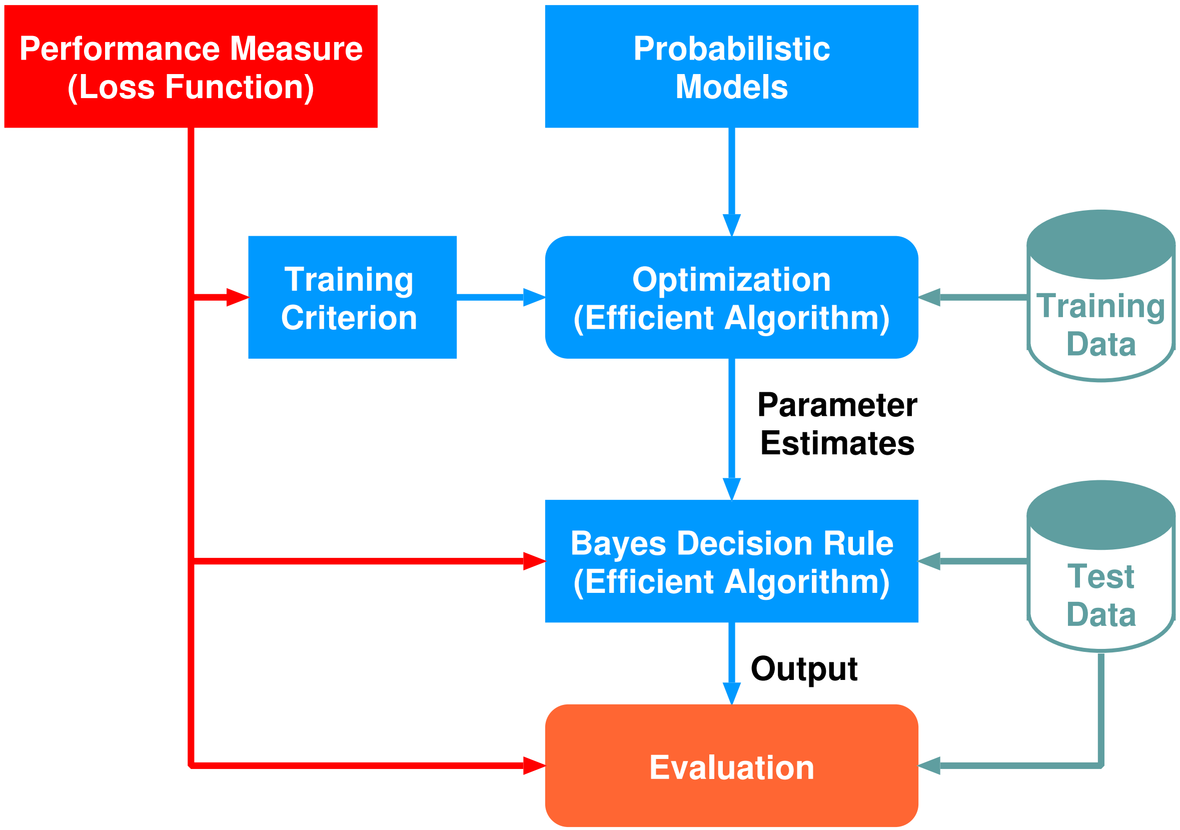
Statistical Approach Revisited

Ingredients:

- **performance measure** (often edit distance):
  to judge the quality of the system output
- **probabilistic models** (with a suitable structure):
  to capture the dependencies within and between X and W
  - elementary observations: Gaussian mixtures, log-linear models, SVMs, NNs, ...
  - strings: n-gram Markov chains, HMMs, CRFs, RNNs, ...
- **training criterion**:
  to learn the free parameters of the models
  - ideally should be linked to performance criterion
  - might result in complex mathematical optimization (efficient algorithms!)
- **Bayes decision rule**:
  to generate the output word sequence
  - combinatorial problem (efficient algorithms)
  - should exploit structure of models

  Examples: dynamic programming and beam search, A^* and heuristic search, ...

Bayes Architecture for Speech Recognition (and other NLP tasks)



Speech Recognition = Modeling + Statistics + Efficient Algorithms

Outline

Sequence Classification

- Problem in Bayes decision rule:
  - true posterior distribution: unknown
  - to replace it, assume suitable model distributions with free parameters:

$$p(W|X) = \frac{p(W) \cdot p(X|W)}{\sum_{W'} p(W') \cdot p(X|W')}$$

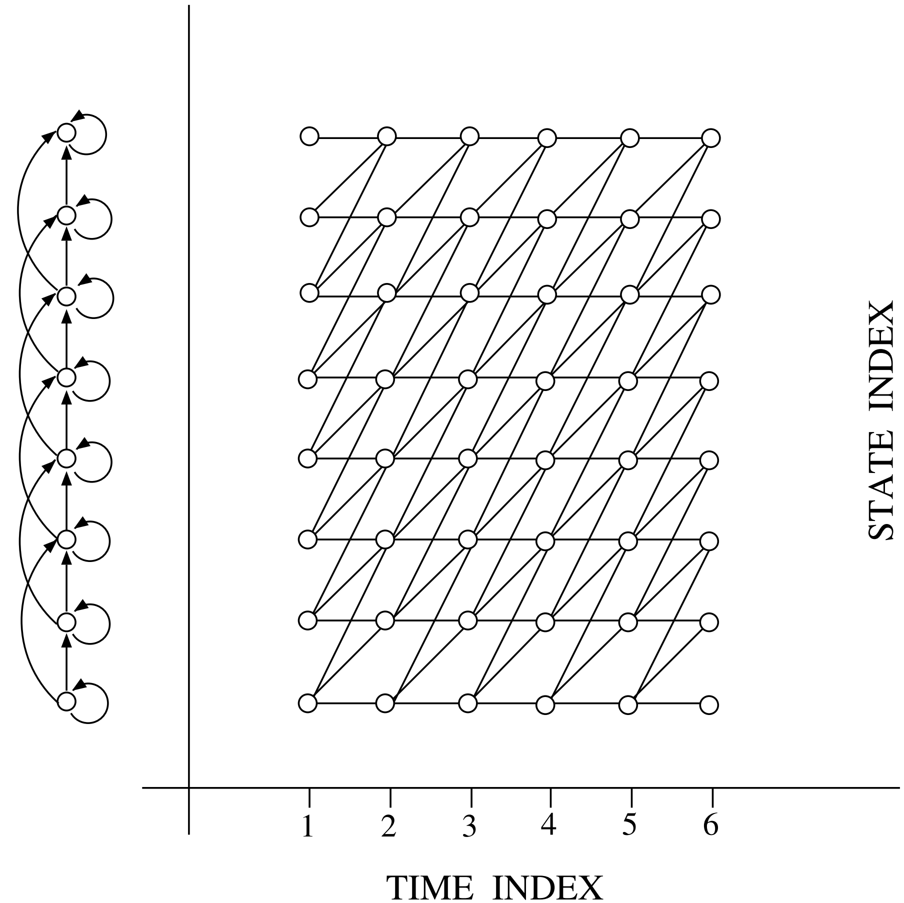
  - generative model: language model $p(W)$ and acoustic model $p(X|W)$
- Acoustic model $p(X|W)$ provides link between sentence hypothesis W and observation sequence $X = x_1^T = x_1 \ldots x_t \ldots x_T$:
  - acoustic probability $p(x_1^T|W)$ using hidden state sequences s_1^T:

$$p(x_1^T|W) = \sum_{s_1^T} p(x_1^T, s_1^T|W) = \sum_{s_1^T} \prod_t [p(s_t|s_{t-1}, W) \cdot p(x_t|s_t, W)]$$

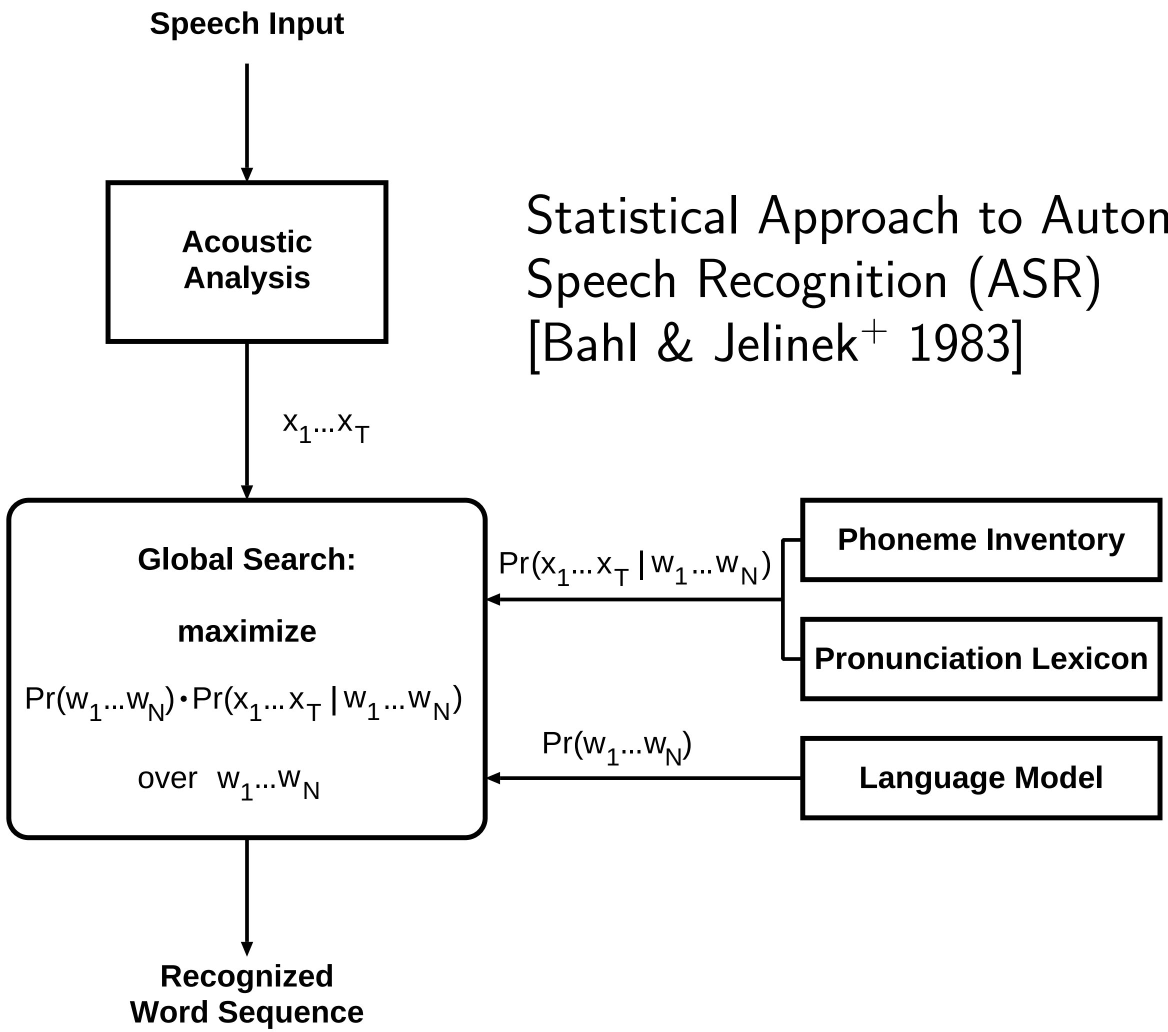
  - two types of distributions:
    * transition probability $p(s|s', W)$: not important
    * emission probability $p(x_t|s, W)$: key quantity
      realized by GMM: Gaussian mixtures models (trained by EM algorithm)

Hidden Markov Models (HMM)

- fundamental problem in ASR:
  non-linear time alignment
- Hidden Markov Model:
  - linear chain of states $s = 1, ..., S$
  - transitions: forward, loop and skip
- trellis:
  - unfold HMM over time $t = 1, ..., T$
  - path: state sequence $s_1^T = s_1 ... s_t ... s_T$
  - observations: $x_1^T = x_1 ... x_t ... x_T$



ASR Architecture



Statistical Approach to Automatic Speech Recognition (ASR) [Bahl & Jelinek[+] 1983]

Outline

HMM using Artificial Neural Network Output: Hybrid Approach

consider modeling the acoustic vector x_t in an HMM:

- phonetic labels (allophones, sub-phones): $(s, W) \rightarrow \alpha = \alpha_{sW}$
  (typical approach: decision trees, e.g. CART):

$$p(x_t|s, W) = p(x_t|\alpha_{sW})$$

- re-write the emission probability for label α and acoustic vector x_t:

$$p(x_t|\alpha) = \frac{p(x_t) \cdot p(\alpha|x_t)}{p(\alpha)}$$

  - prior probability $p(\alpha)$: estimated as relative frequencies (alternatively averaged NN posteriors)
  - for recognition purposes: term $p(x_t)$ can be dropped
- result: rather than the state emission distribution $p(x_t|\alpha)$,
  model the label posterior probability by an NN:

$$x_t \rightarrow p(\alpha|x_t)$$

- justification:
  - easier learning problem: labels $\alpha = 1, ..., 5000$ vs. vectors $x_t \in \mathbb{R}^{D=40}$
  - well-known result in pattern recognition (but ignored in ASR!)

Outline

History: Artificial Neural Networks in Acoustic Modeling

approaches in ASR:

- [Waibel & Hanazawa^{+} 1988]: phoneme recognition using time-delay neural networks
- [Bridle 1989]: softmax operation for probability normalization in output layer
- [Bourlard & Wellekens 1990]:
  - for squared error criterion, NN outputs can be interpreted as class posterior probabilities (rediscovered: Patterson & Womack 1966)
  - they advocated the use of MLP outputs to replace the emission probabilities in HMMs
- [Robinson 1994]: recurrent neural network
  - competitive results on WSJ task
  - his work remained a singularity in ASR
- ...

experimental situation:
until 2011, NNs were never really competitive with(out) Gaussian Mixture Models

History: Artificial Neural Networks in Acoustic Modeling

related approaches:

- [LeCun & Bengio$^+$ 1994]: convolutional neural networks
- A. Waibel's team [Fritsch & Finke$^+$ 1997]: hierarchical mixtures of experts
- [Hochreiter & Schmidhuber 1997]: long short-term memory neural computation (LSTM RNN) with extensions [Gers & Schraudolph$^+$ 2002]

(second) renaissance of NN: concepts of deep learning and related ideas:

- [Hermansky & Ellis$^+$ 2000]: tandem approach - multiple layers of processing by combining Gaussian model and NN for ASR
- [Utgoff & Stracuzzi 2002]: many-layered learning for symbolic processing
- [Hinton & Osindero$^+$ 2006]: introduced what they called *deep learning (belief nets)*
- [Graves & Bunke$^+$ 2008]: good results for LSTM RNN on handwriting task
- Microsoft Research [Seide & Li$^+$ 2011, Dahl & Yu$^+$ 2012]:
  - combined Hinton's deep learning with hybrid approach
  - significant improvement by deep MLP on a large-scale task
- since 2012: other teams confirmed reductions of WER by 20% to 30%

Outline

Empirical Overview of Current Methods

Experimental conditions:

- QUAERO task: English broadcast news and conversations (evaluation campaign 2011)
- training data: two conditions: 50 and 250 hours
- test data: dev and eval sets, each 3 hours
- language model: vocabulary size of 150k (OOV: 0.4%) and perplexity of 130

Baseline Gaussian mixture HMM based acoustic model:

- feature vector: 16 MFCC (mel frequency cepstral coefficients)
- augmented feature vector: $9 \cdot 16 = 144$
- high-performance baseline system:
  Gaussian mixtures with pooled diagonal covariance matrix:
  - reduction by LDA to 45-dimensional vector
  - 4501 CART labels
  - 680k densities
  - total number of free parameters: $680k \cdot (45 + 1) = 31.3M$

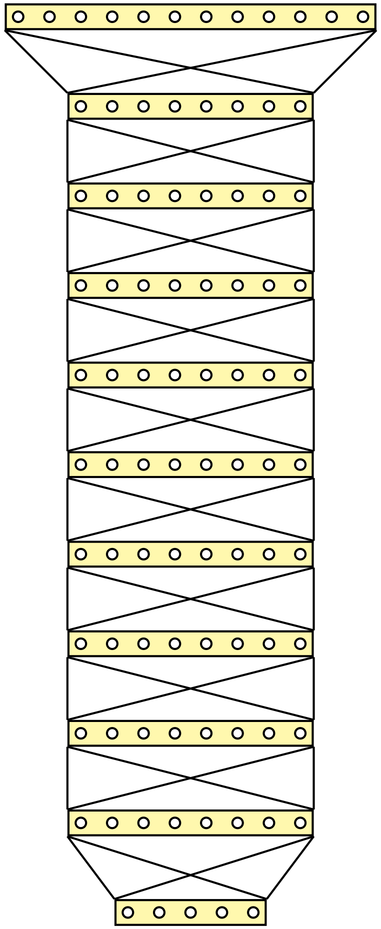
Gaussian Mixture Models (GMM): Influence of Training Criteria

Training Criterion	WER [%]			
	50h		250h	
	dev	eval	dev	eval
Maximum likelihood	24.4	31.6	22.1	28.6
MMI at frame level	23.9	30.9	22.1	28.6
MMI at sentence level	24.1	31.2	21.7	28.1
Minimum phone error	23.6	30.2	20.4	26.2

remarks:

- best improvement over maximum likelihood:
  5-10% relative by MPE (Minimum Phone Error)
- comparative evaluations in QUAERO:
  competitive results with LIMSI Paris and KIT Karlsruhe

Deep MLP: Number of Hidden Layers



- WER vs. number of hidden layers for 50-h training corpus
- Structure of MLP:
  - input dimension: 493 (window + derivatives)
  - 2000 nodes per hidden layer
  - nonlinearity: sigmoid
  - number of parameters for 6-layer MLP:

$$\begin{aligned} & 493 \cdot 2000 \\ & +5 \cdot 2000^2 \\ & +2000 \cdot 4501 \\ = & \ 30\text{M} \end{aligned}$$

- improvement over best GMM: 20% relative

hidden layers	WER [%]	
	dev	eval
1	24.5	31.3
2	22.0	28.3
3	20.5	26.7
4	19.8	26.1
5	20.1	26.0
6	19.6	25.4
7	19.7	25.5
8	19.6	25.7
9	19.3	25.3
best GMM	23.6	30.2

Practicalities of NN Training: Implementation and Software

typical procedure:

- input data: (sentence-wise) mean and variance normalization
- random initialization of weights: [-0.1,...,+0.1]
- training criterion: (frame-wise) cross-entropy
- stopping: cross-validation on 10% of training data
- sigmoid function
- no regularization, no momentum term, no drop-out (so far!)
- learning rate: reduced over time by a factor of 20-50
- use of minibatches: 512 frames
- pretraining:
  - supervised pretraining: layer by layer
  - in general: not crucial
- use of GPUs: speed-up by a factor of 10 over multithreaded CPUs

Discriminative Sequence Training: MPE vs. CE

Comparison of two training criteria (MLP with 6 hidden layers, 2000 nodes each):

- baseline: cross-entropy = frame MMI
- MPE: minimum phone error (context of pron. lexicon and language model)

Model	Criterion	WER [%]			
		50h		250h	
		dev	eval	dev	eval
MLP	frame MMI	19.6	25.4	15.2	20.4
	MPE	17.5	23.3	14.1	19.2
best GMM		23.6	30.2	20.4	26.4

experimental result: improvement of 5-10% by MPE over frame MMI

Activation Function: Sigmoid vs. RLU

- activation functions:
  - sigmoid function: $u \rightarrow f(u) = 1/(1 + e^{-u})$
  - RLU=rectified linear unit: $u \rightarrow f(u) = \max\{0, u\}$
- structure of MLP:
  - 6 hidden layers, each with 2000 nodes
  - training condition:
    - (frame-wise) cross-entropy
    - L2 regularization (weight decay): important
    - momentum term
- word error rates for activations functions: sigmoid vs. RLU:

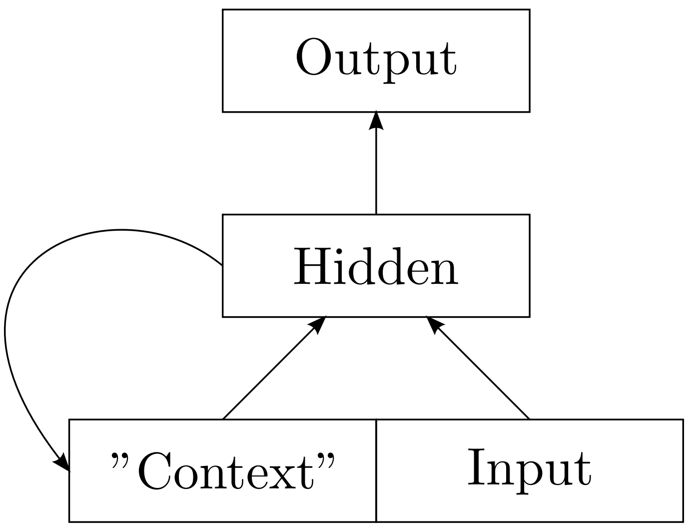
activation function	WER [%]			
	50h		250h	
	dev	eval	dev	eval
sigmoid	19.6	25.4	15.2	20.4
RLU	17.7	23.5	14.7	19.6
best GMM	23.6	30.2	20.4	26.4

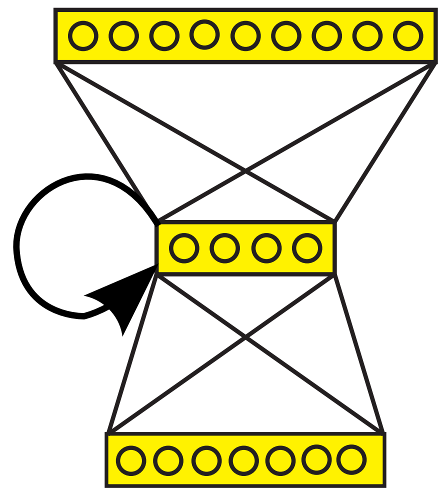
- experimental result: improvement of 5-10% by RLU over sigmoid

Recurrent Neural Network (RNN): Principle

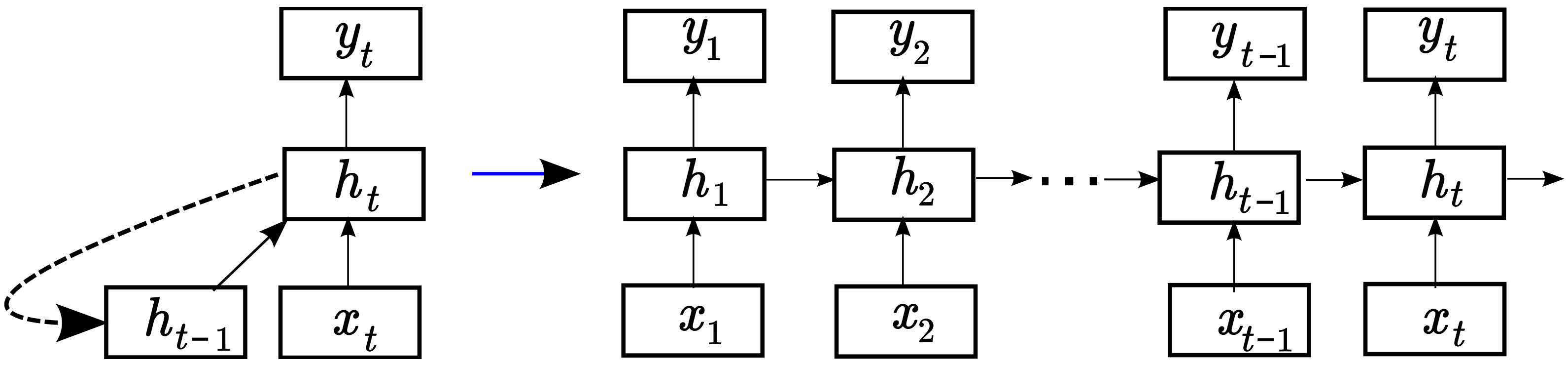
principle:

- introduce a **memory** (or context) component to keep track of history
- result: there are two types of input: memory h_{t-1} and observation x_t





Unfolding RNN over Time



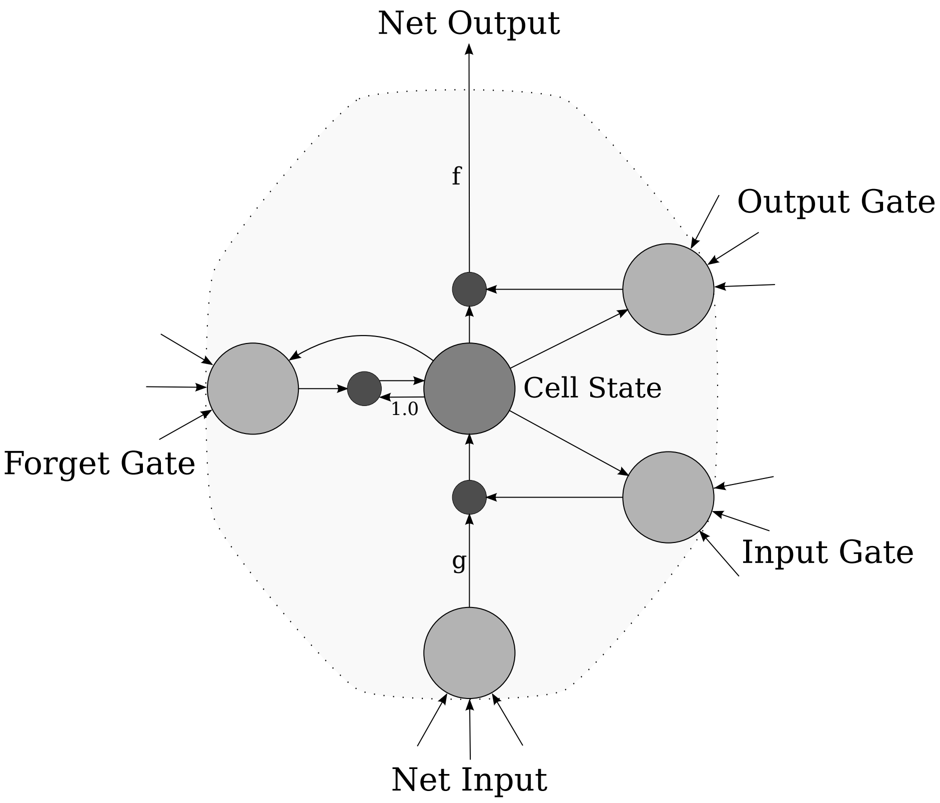
The architecture of RNN can be unfolded over time:

- We get a feedforward network with a special **deep** architecture.
- The application of the backpropagation algorithm to this unfolded network is called **backpropagation through time**.

LSTM RNN [Hochreiter & Schmidhuber 1997, Gers & Schraudolph$^+$ 2002]

extension of (simple) RNN by
LSTM: long short-term memory

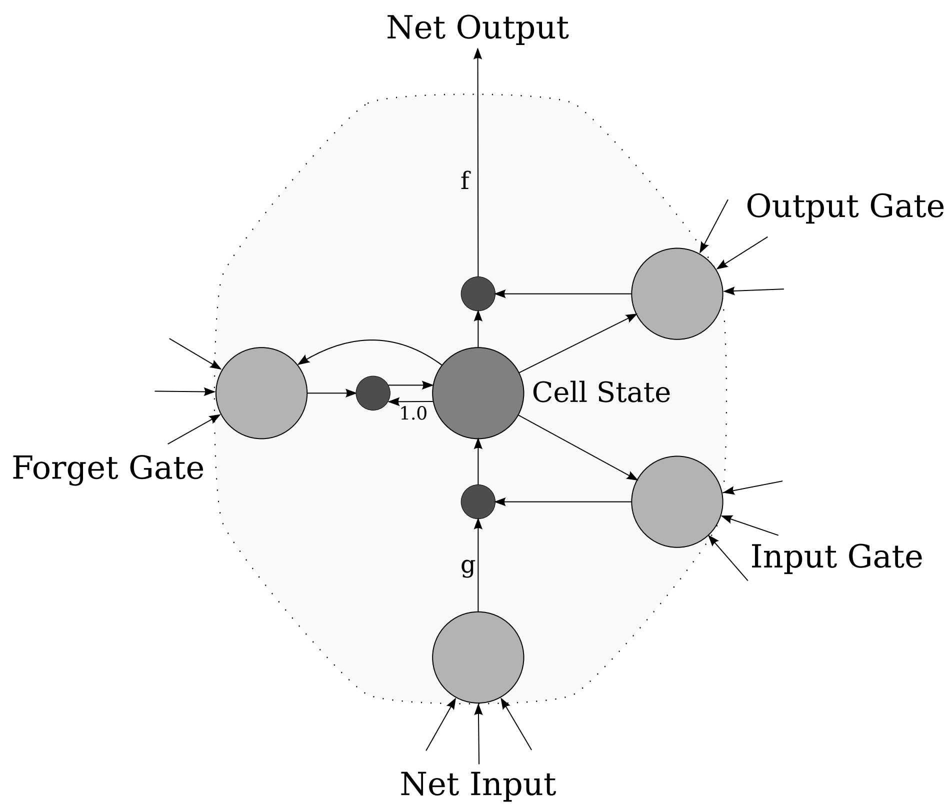
- problems of simple RNN:
  - vanishing/exploding gradients
  - no protection of memory h_t
- remedy by LSTM architecture:
  control the access to its internal memory
  by introducing gates/switches
- refinements:
  - bidirectional structure
  - several hidden layers



LSTM RNN [Hochreiter & Schmidhuber 1997, Gers & Schraudolph+ 2002]

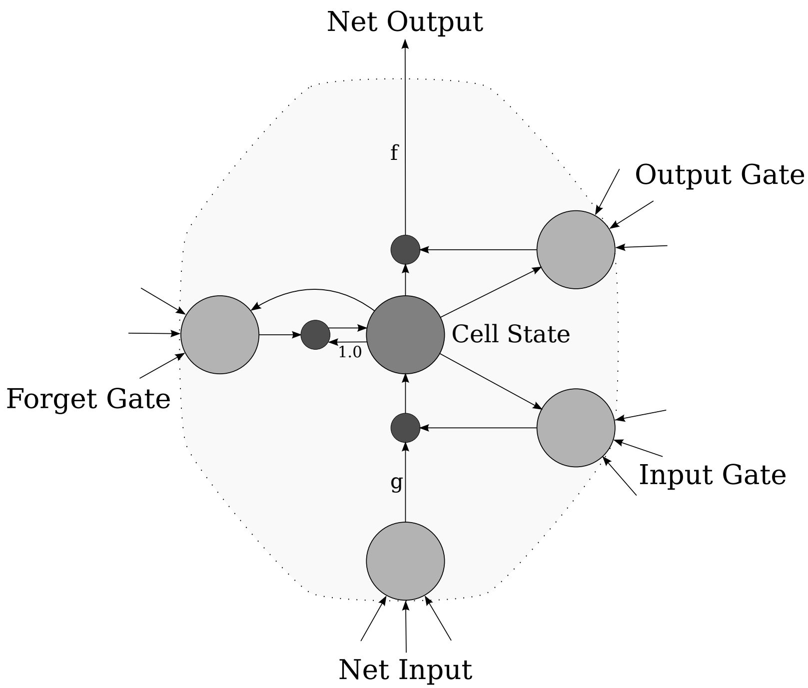
LSTM approach:

- split RNN hidden vector h_t into (memory) cell state c_t and net output s_t
- overall LSTM operations involve three 'input' vectors at time t: s_{t-1}, c_{t-1}, x_t
- update operations at time t:
  cell state: $c_t = c_t(s_{t-1}, c_{t-1}, x_t)$
  net output: $s_t = s_t(s_{t-1}, c_{t-1}, x_t)$
  output layer: $y_t = y_t(s_t)$ with softmax
- introduce three gates (input, output, forget) to control the information flow



LSTM Architecture

- three vectors (over time t): c_t, s_t, x_t
- gates (or switches): use sigmoid function $\sigma(\cdot)$
- full matrices $(A_2, R; A_i, R_i, A_f, R_f, A_o, R_o)$ and diagonal matrices (W_i, W_f, W_o)
- usual matrix and vector operations and element-wise multiplication $\odot$
- Net Input (like update formula of simple RNN):
$$z_t = \tanh(A_2 x_t + R s_{t-1})$$
- Should this Net Input z_t access the Cell State c_t?
Input Gate: $i_t = \sigma(A_i x_t + R_i s_{t-1} + W_i c_{t-1})$
- Should the Cell State c_{t-1} be forgotten?
Forget Gate: $f_t = \sigma(A_f x_t + R_f s_{t-1} + W_f c_{t-1})$
- Based on i_t and f_t, update the Cell State c_t:
$$c_t = f_t \odot c_{t-1} + i_t \odot z_t$$
- Should this update c_t be output?
Output Gate: $o_t = \sigma(A_o x_t + R_o s_{t-1} + W_o c_t)$
- Based on o_t, compute the Net Output:
$$s_t = o_t \odot c_t$$



Deep LSTM-RNN

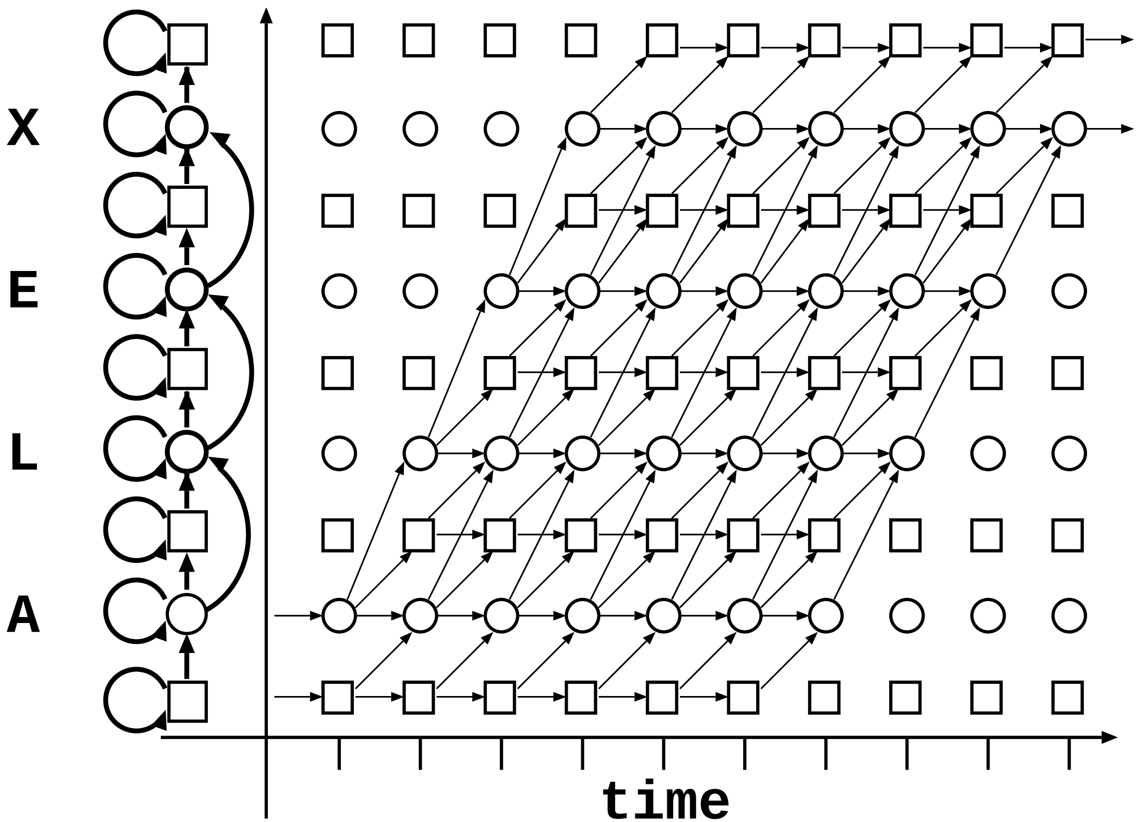
50h QUAERO training corpus:

- baseline: best MLP:
  - input: 50 Gammatone features
  - 9 hidden layers
  - RLU
  - training criterion: cross-entropy
- LSTM-RNN structure:
  - input: 50 Gammatone features
  - training criterion: cross-entropy
  - bidirectional with several hidden layers
  - 500 nodes per hidden layer
  - training on a single GPU
- eval improvements:
  - 14% relative over MLP
  - 42% relative over GMM

LSTM layers	#params	time / epoch	WER [%] dev	WER [%] eval
1	6.7M	0:28h	17.6	22.7
2	12.7M	1:00h	14.6	18.8
3	18.7M	1:11h	14.0	18.4
4	24.7M	1:33h	13.5	17.7
5	30.7M	1:48h	13.6	17.7
6	36.7M	2:10h	13.5	17.5
7	42.7M	2:36h	13.8	18.0
8	48.7M	3:14h	14.2	18.4
best MLP (9x2000)	42.7M	0:35h	15.3	20.3
best GMM	31.3M	–	23.6	30.2

CTC: Connectionist Temporal Classification

[Graves & Fernández+ 2006, Graves & Bunke+ 2008]



Related Research Directions

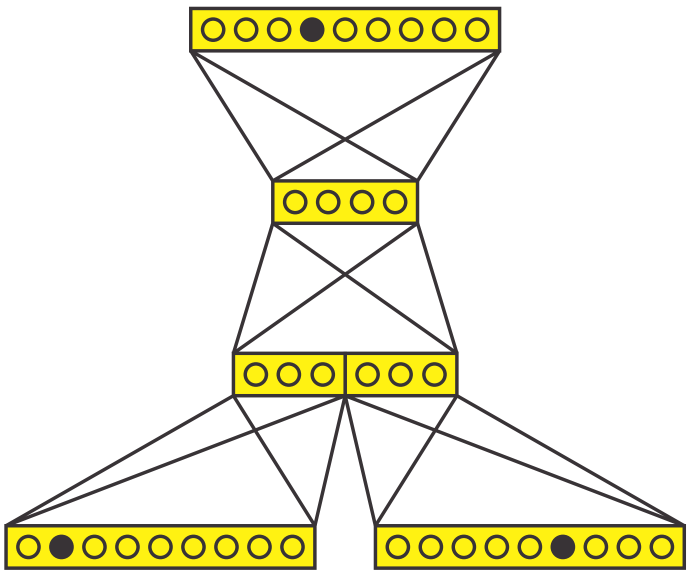
- CTC: What is different from an HMM? What is important?
  - topology: several vs. single state per symbol
  - training criterion: sum vs. maximum
  - no transition probabilities
  - NN structure: RNN-LSTM
- recent neural network approaches (replacing the HMM alignment?):
  - end-to-end approaches
  - mechanism of attention

Outline

Review: Language Modeling

- distinguish:
  - *sub-symbolic* processing: speech/audio, text images, image/video (computer vision)
  - *symbolic processing:* language modeling (and machine translation)
- word sequence $w_1^N := w_1 \ldots w_n \ldots w_N$
- language model: conditional probability $p(w_n|w_0^{n-1})$ (with artificial start symbol w_0):

$$p(w_1^N) = \prod_{n=1}^{N} p(w_n|w_0^{n-1})$$



- approaches to modeling $p(w_n|w_0^{n-1})$
  - count models (Markov chain):
    * limit history w_0^{n-1} to k predecessor words
    * smooth relative frequencies (e.g. SRI toolkit)
  - MLP models:
    * limit history, too
    * use predecessor words as input to MLP
  - RNN models: unlimited history!

History of Neural Networks in Language Modeling

- [Nakamura & Shikano 1989]:
  English word category prediction based on neural networks.
- [Castano & Vidal+ 1993]:
  Inference of stochastic regular languages through simple recurrent networks
- [Bengio & Ducharme+ 2000]:
  A neural probabilistic language model
- [Schwenk 2007]:
  Continuous space language models
- [Mikolov & Karafiat+ 2010]:
  Recurrent neural network based language model
- RWTH Aachen [Sundermeyer & Schlüter+ 2012]:
  LSTM recurrent neural networks for language modeling
- RWTH Aachen [Sundermeyer & Tüske+ 2014]:
  long range LM rescoring beyond N-best lists

Today: neural network based language models show competitive results.

Outline

Language Modeling

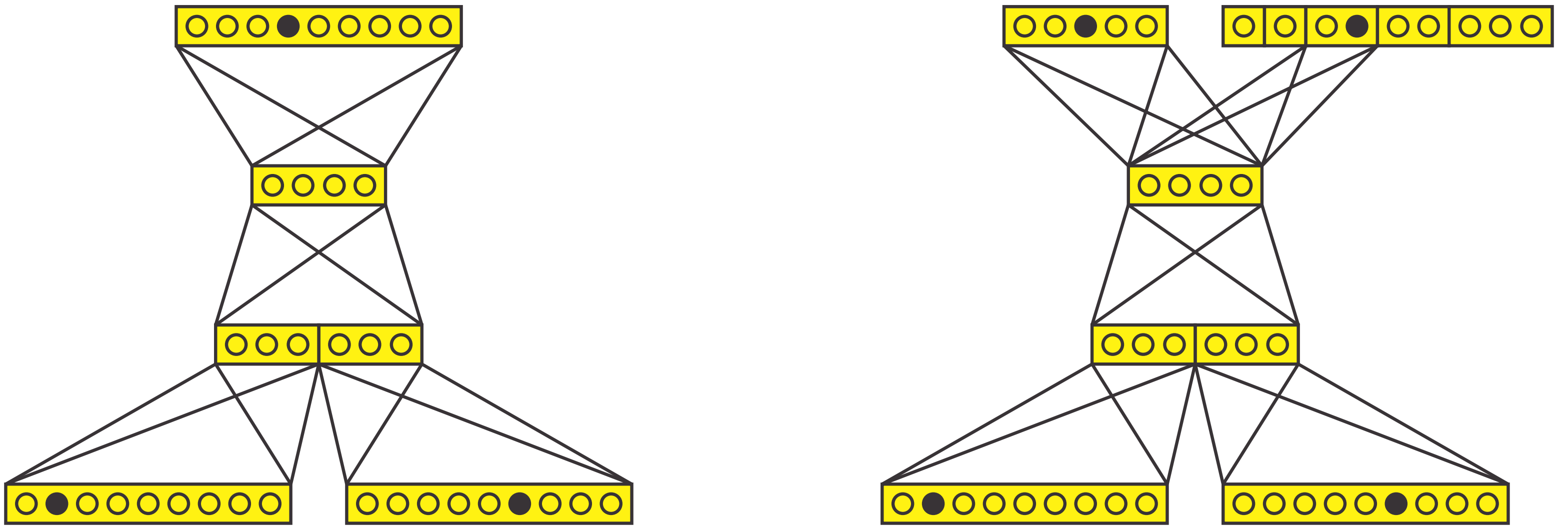
Structure of Neural Network for Language Modeling

- input layer: k predecessor words with 1-of-V coding (V = vocabulary size)
- first layer: *projection layer*
  - idea: dimension reduction (e.g. from 150k to 600!)
  - a linear operation (matrix multiplication) without sigmoid activation
  - shared accross all predecessor words of the history h
- output layer:
  - conditional probability of language model $p(w|h)$
  - softmax operation for normalization
- training criterion:
  - perplexity: equivalent to cross-entropy
  - early stopping using cross-validation on dev corpus
- properties of softmax operation:
  - computationally expensive (sum over full vocabulary)
  - remedy: word classes (automatically trained)
  - normalized outputs of softmax fit nicely into perplexity criterion

Language Modeling

Word Classes

MLP w/o and with Word Classes: Trigram LM



factorization of conditional language model probability $p(w|h)$ for each history h:

$$p(w|h) = p(g|h) \cdot p(w|g,h)$$

using a unique word class g for each word w

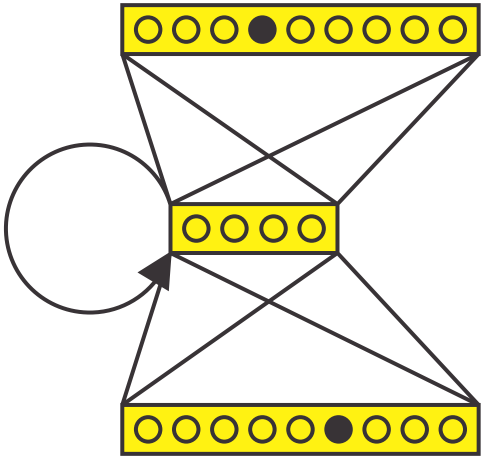
Word Classes

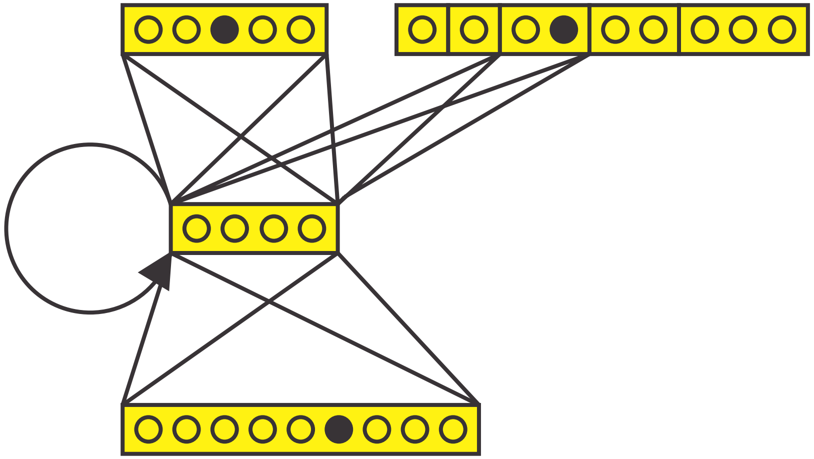
RNN without and with Word Classes

- NN with memory for sequence processing
- left-to-right processing of word sequence $w_1 \ldots w_n \ldots . w_N$

$$p(w_1^N) = \prod_n p(w_n | w_0^{n-1}) = \prod_n p(w_n | w_{n-1}, h_{n-1})$$

- input to RNN in position n:
  - output h_{n-1} of hidden layer at position $(n-1)$
  - immediate predecessor word w_{n-1}



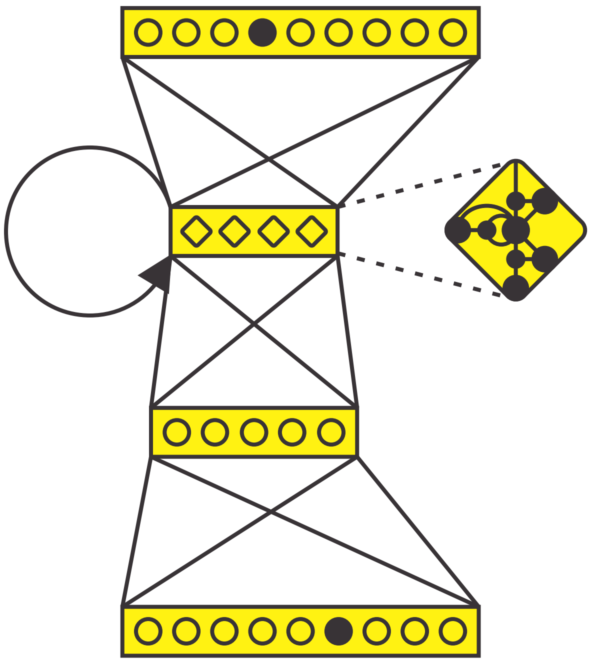


Language Modeling

LSTM RNN [Hochreiter & Schmidhuber 1997, Gers & Schraudolph+ 2002]

refinement of RNN:
LSTM = long-short term memory

- RNN: problems with vanishing/exploding gradients
- remedy: cells with gates rather than nodes
- details: see literature



Outline

Experiments

- results on QUAERO English (like before):
  - vocabulary size: 150k words
  - training text: 50M words
  - dev and eval sets: 39k and 35k words
- MLP: structure:
  - projection layer: 300 nodes
  - hidden layer: 600 nodes
  - size of MLP is dominated by input and output layers: $150k \cdot 300 + 600 \cdot 150k = 135M$
- RNN (and LSTM RNN): structure
  - projection and hidden layer: each 600 nodes
  - size of RNN is dominated by input and output layers: $150k \cdot 600 + 600 \cdot 150k = 180M$

perplexity PPL on dev data:

approach	hidden layers	PPL
count model	–	163.7
10-gram MLP	1	136.5
	2	130.9
RNN	1	125.2
LSTM-RNN	1	107.8
	2	100.5

observation:
(huge) improvement by 40%

Complexity: Computation Times

Training times (without GPUs!) for training corpus of 50 Million words:

Models	PPL	CPU Time (Order)
Count model	163.7	30 min
MLP	136.5	1 week
LSTM-RNN	107.8	3 weeks

- problem: high computation times
- remedy: two types of language models:
  - count model: trained on a huge corpus: 3.1 Billion words
  - NN models: trained on a small corpus: 50 Million words
- resulting language model:
  linear interpolation of *two* models

Language Modeling

Interpolated Language Models: Perplexity and WER

- linear interpolation of *two* models: count model + NN model
- perplexity and word error rate on test data:

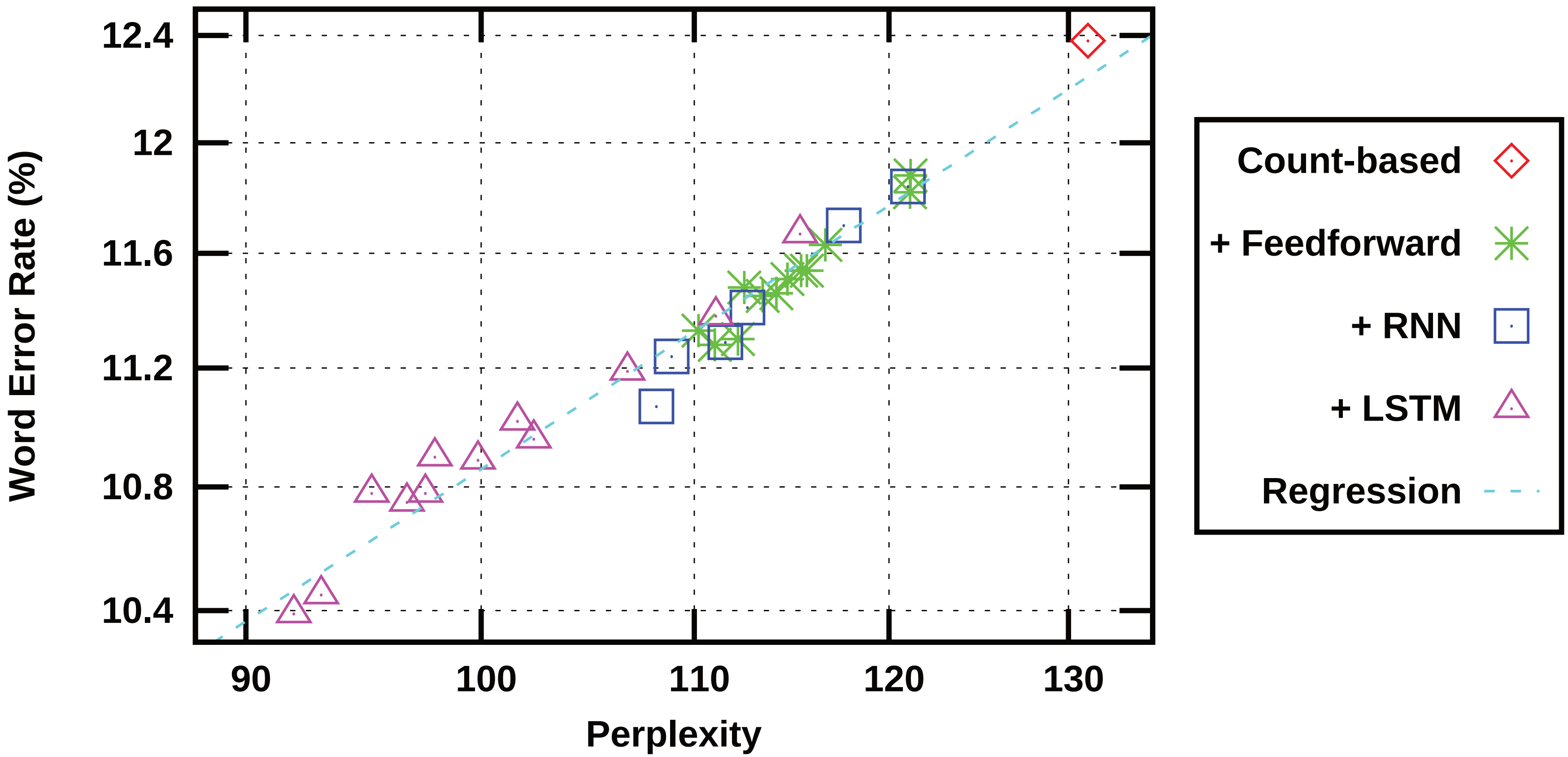
Models	PPL	WER[%]
count model	131.2	12.4
+ 10-gram MLP	112.5	11.5
+ Recurrent NN	108.1	11.1
+ LSTM-RNN	96.7	10.8
+ 10-gram MLP with 2 layers	110.2	11.3
+ LSTM-RNN with 2 layers	92.0	10.4

- experimental result:
  - significant improvements by NN language models
  - best improvement in perplexity: 30% reduction (from 131 to 92)
  - empirical observation:
    power law between WER and perplexity (cube to square root)

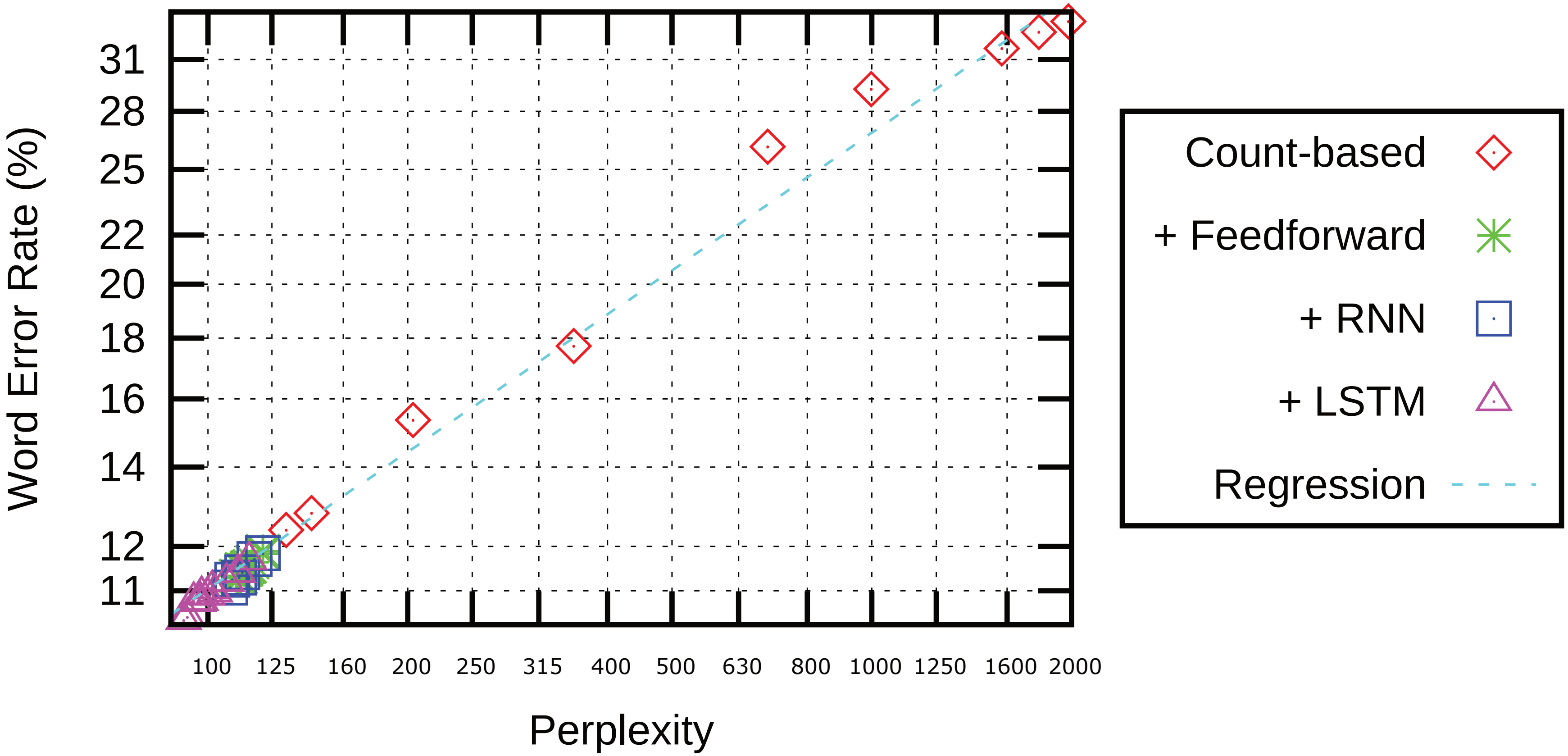
Outline

Language Modeling

Perplexity vs. Word Error Rate

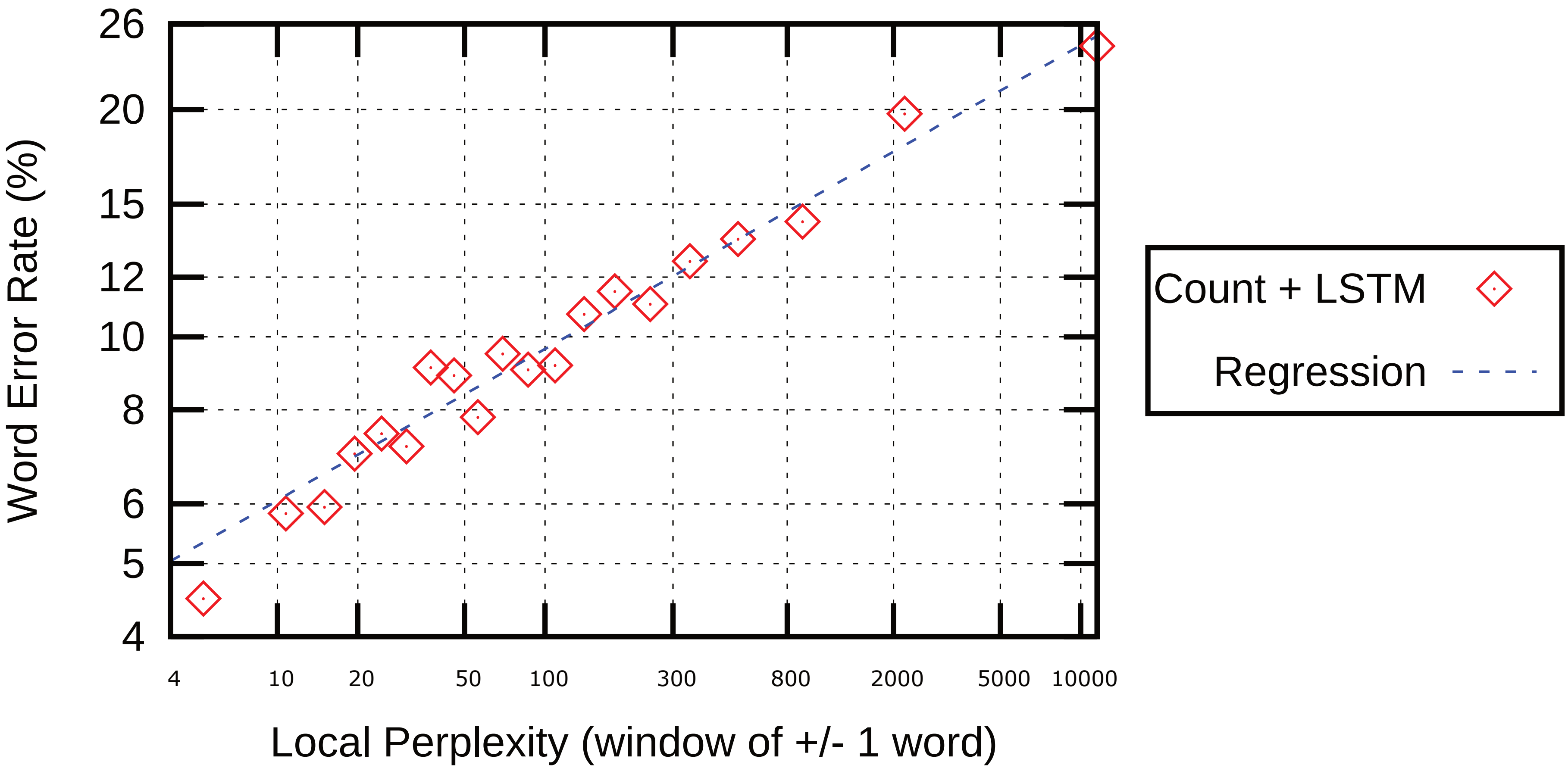


Extended Range: Perplexity vs. Word Error Rate



Language Modeling

Word Error Rate vs. Local Perplexity (3-word window, 20 bins)



Outline

Motivation

- End-to-end model:
  - Consistence of modeling, training, and decoding.
  - Cover segmentation problem by NN structure:
    sequence length, duration, and positioning of words are unknown.
  - Context dependence needs to be modeled.

- Ultimate goals (not fully achieved yet):
  - Integration of NN models into Bayes decision rule.
  - Separation of acoustic & language model (resources usually differ).
  - Consistence between decision rule, evaluation measure,
    and training objective.

Review: Hidden Markov Modeling

- models words/word sequences by HMM state sequences
- within Bayes decision rule:

$$\begin{aligned}
\arg\max_{N,w_1^N} p(w_1^N) \cdot p(x_1^N|w_1^N) &= \arg\max_{N,w_1^N} p(w_1^N) \cdot \sum_{s_1^T : w_1^N} p(x_1^T, s_1^T|w_1^N) \\
&= \arg\max_{N,w_1^N} p(w_1^N) \cdot \sum_{s_1^T : w_1^N} \prod_{t=1}^{T} p(x_t|x_1^{t-1}, s_1^t) \cdot p(s_t|x_1^{t-1}, s_1^{t-1}) \\
&= \arg\max_{N,w_1^N} p(w_1^N) \cdot \sum_{s_1^T : w_1^N} \prod_{t=1}^{T} p(x_t|s_t) \cdot p(s_t|s_{t-1}) \quad \text{1}^{\text{st}} \text{ order Markov} \\
&\approx \arg\max_{N,w_1^N} p(w_1^N) \cdot \max_{s_1^T : w_1^N} \prod_{t=1}^{T} p(x_t|s_t) \cdot p(s_t|s_{t-1}) \quad \text{Viterbi approx.}
\end{aligned}$$

Review: Hidden Markov Modeling

Discussion:

- HMM-based standard decision rule:

$$\arg\max_{N,w_1^N} p(w_1^N) \cdot \max_{s_1^T : w_1^N} \prod_{t=1}^{T} p(x_t|s_t) \cdot p(s_t|s_{t-1})$$

  - In practice: **maximum** over segmentations, **especially in search** (Viterbi approximation)
  - Ideally: sum over segmentations.
- Inconsistency for (hybrid) NN integration into acoustic model:

$$p(x_t|s) = \frac{p(s|x_t) \cdot p(x_t)}{p(s)}$$

  - NN provides state posterior, but state cond. probability needed.
  - $p(s)$ approximated, e.g. [Manohar & Povey$^+$ 2015].

Review: Hidden Markov Modeling

Discussion:

- Assumption of independence of acoustic context:
  - Can be relaxed by considerung window around each time frame t: $x_{t-\delta}^{t+\delta}$
  - Hybrid modeling: emission probability modelled by rescaled state posteriors $p(s|x_t)$
  - observation here appears in condition only and may be replaced by full acoustic context:
    $\rightarrow\ p(s|t, x_1^T)$ (e.g. obtained by bi-directional recurrent modeling).
- Segmentation/alignment of observations to HMM states:
  - Stochastic: ideally sum over all aligments.
  - Explicit in case of Viterbi approximation: maximizing alignment.
- Integration of language model:
  - Clearly defined, can be trained separately
    (text data vs. transcribed acoustic data).
  - However, language model scaling exponent statistically unclear.
  - Open issue: interaction of context dependence on observation and symbol/word level.

Outline

End-to-End Approach

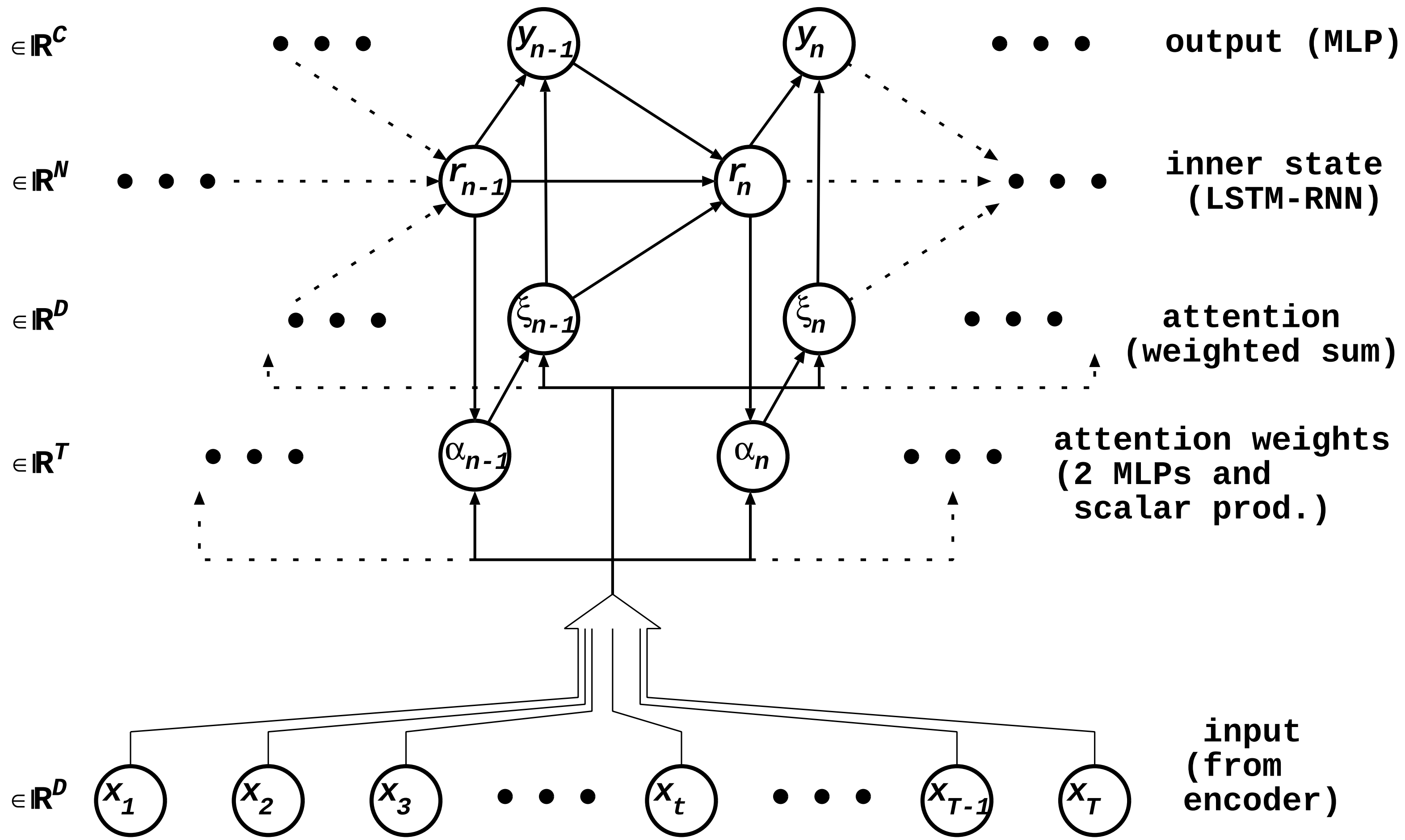
- Motivation: End-to-end trainable neural network recognizer
  - Consistently integrate input and output sequences.
  - Does not need explicit segmentation.
  - Avoids Markov and independence assumptions.

- Sequence-to-sequence modeling [Sutskever & Vinyals^{+} 2014]:
  - Idea: separate processing of input and output into two models:
  - **Encoder**: Read the inputs and generate discriminative features
  - **Decoder**: Write the output symbol sequence label by label considering all encoded features

- Encoder can be viewed as non-linear transformation of input:
  - Similar to tandem in hybrid approach (hierarchical model), **but**:
  - Encoder output is not related to specific output labels.
  - Jointly trained within the complete end-to-end structure.

End-to-End Approach

"Listen, Attend and Spell" [Chan & Jaitly+ 2015]



End-to-End Approach

"Listen, Attend and Spell" [Chan & Jaitly+ 2015]

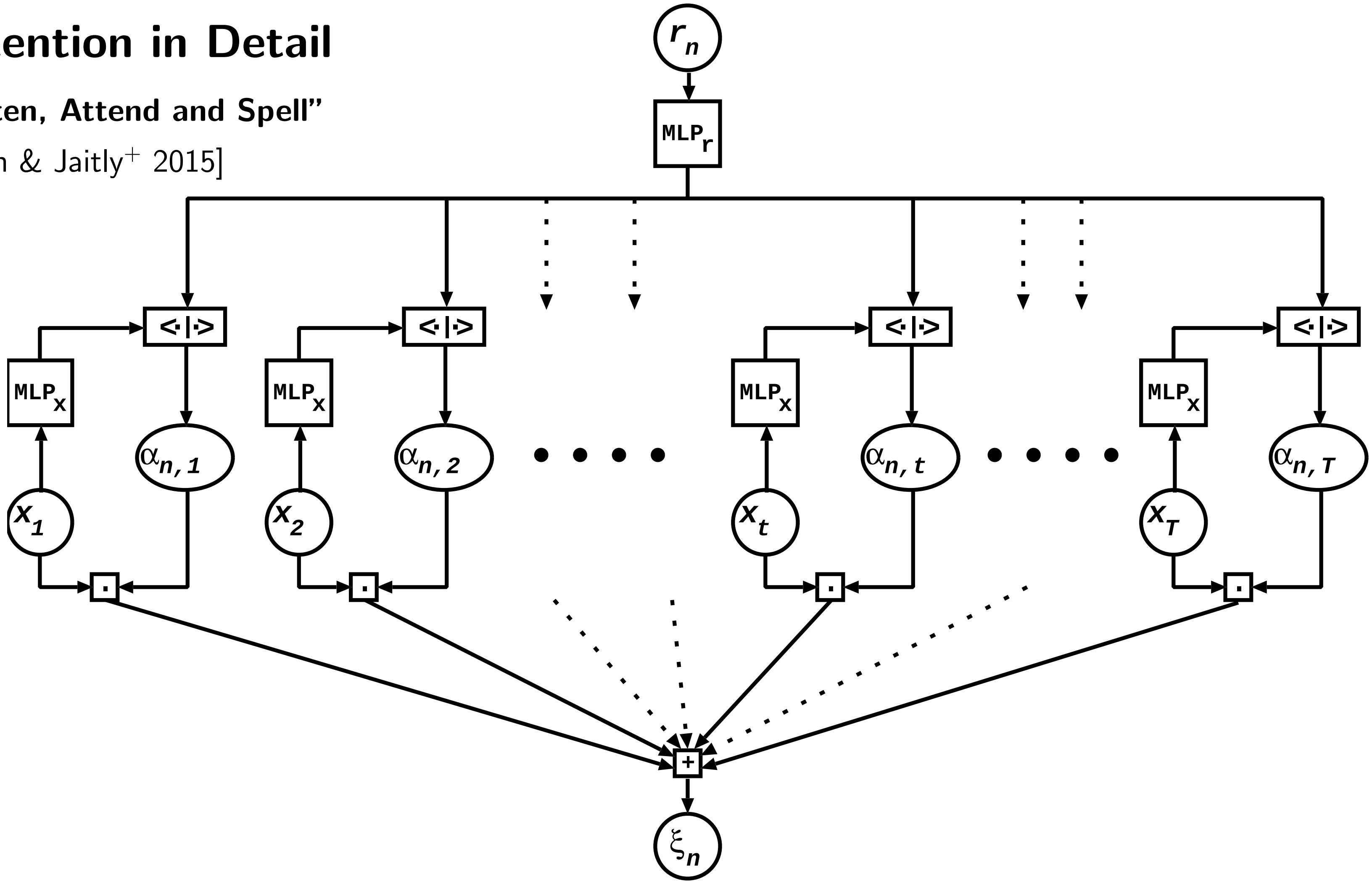
Approach:

1. **"Listen"**:
   i. Encode input (bidirectional recurrent (LSTM) network, omitted in figure). Encoding usually includes gradual temporal subsampling/integration.
2. **"Attend"**: at each output symbol position n:
   i. Compute the current inner state value r_n from previous state r_{n-1}, output y_{n-1}, and expected input ξ_{n-1} from attention.
   ii. Compute attention weights $\alpha_n = attend(r_n, \ldots)$ from current state r_n and further input (see next slide).
   iii. Compute expected network input ξ_n as linear combination of input sequence x_1^T weighted by $\alpha_{n,1}^T$
3. **"Spell"**:
   i. Recurrently classify characters (symbols) from current state r_n and input ξ_n from attention.

Attention in Detail

“Listen, Attend and Spell”

[Chan & Jaitly^{+} 2015]

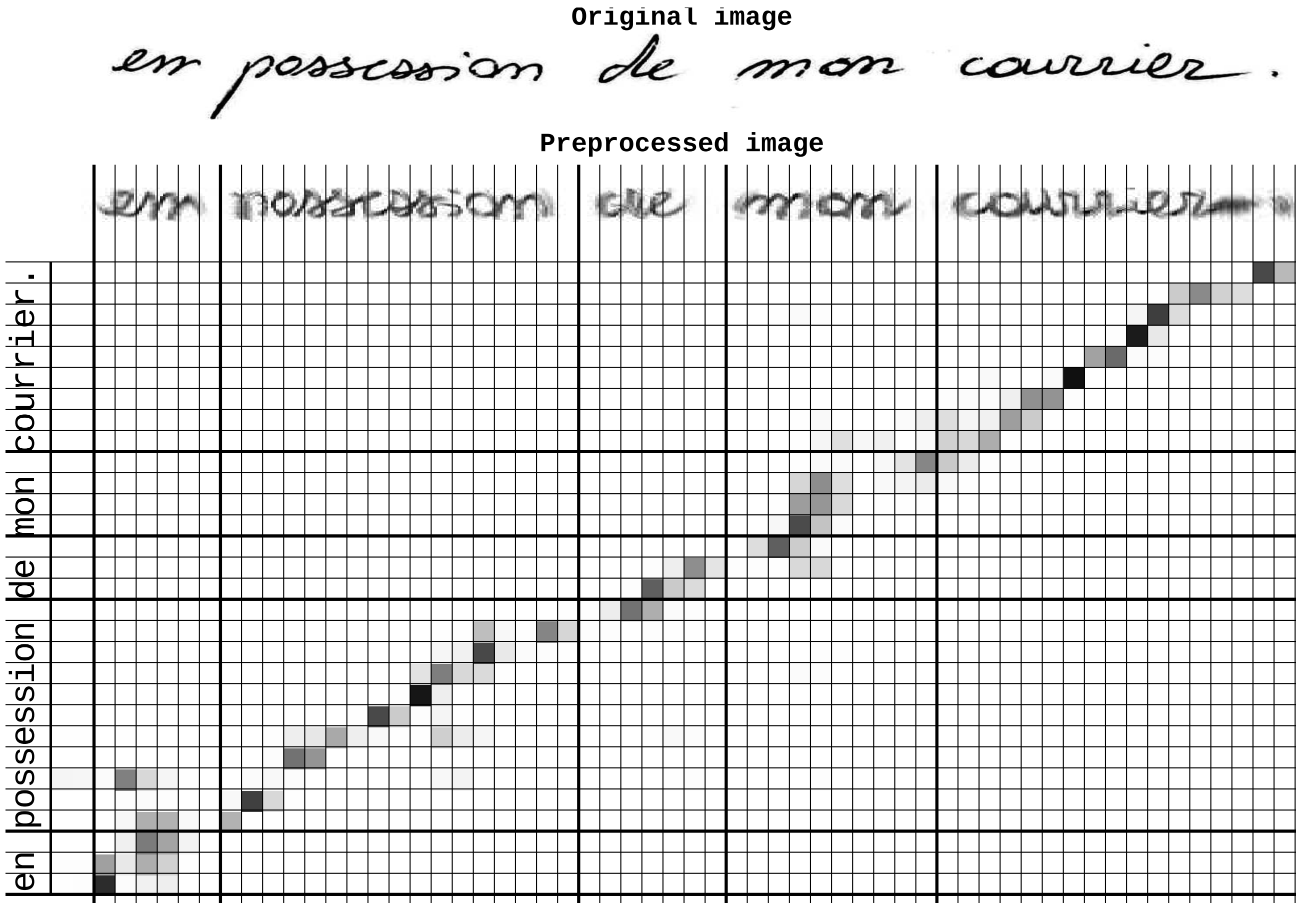


Outline

Discussion

- The attention process controls the segmentation
  $\rightarrow$ (soft) alignment between symbol position and observations.
- The dependencies of the attention process still are an open research issue, choices investigated:
  - [Chan & Jaitly+ 2015] ("Listen, Attend and Spell"): $\alpha_n = attend(r_n, x_1^T)$
  - [Bahdanau & Chorowski+ 2015]: $\alpha_n = attend(r_{n-1}, y_{n-1}, \xi_{n-1})$
- Discussion:
  - No explicit alignment to specific input vectors needed.
  - However, attention is **determined** by context, i.e. it is not handled as an independent hidden stochastic variable.
  - As a consequence, suboptimal attention results (misalignments) cannot be rectified in the subsequent search process, as in HMM based modeling.

Attention Modeling Example from Handwriting



Sequence-to-Sequence Approach

Results: RIMES Offline Handwriting Recognition

- Input: 8×32 image slices resulting from sliding window (shift 3).
- Input layer: CNN with filter size 3×3 and 64 features, no pooling.
- Hybrid: 4 BLSTM layers with 512 cells in each direction,
  - realignment: retraining on new alignment created based on hybrid.
- Attention-based: encoder (almost) equal to hybrid:
  - "subsampling" by factor of 2 after 2$^{\text{nd}}$ and 4$^{\text{th}}$ BLSTM layer (stacking) (no subsampling/stacking in framewise system).
- decoder network: single BLSTM with 512 cells for each direction.
- # params: $\sim$ 20.8M for encoder/hybrid +700k for decoder BLSTM.

Approach	WER [%]	CER [%]
Hybrid HMM	13.0	7.6
+ realignment	12.9	5.8
Attention-based	16.2	8.0
+ LM rescoring	14.2	6.3

Outline

Inverted Search

- Neural network based modeling provides HMM state posteriors.
- Can (sub)word sequences directly be modeled using state posteriors?
- Idea: **invert** alignment problem:
  - state boundaries t_1^N as hidden variables,
  - (triphone state) label sequence α_1^N directly represents word (sequence) template.
  - Approach: alternative decomposition by chain rule/Bayes identity:

$$
\begin{aligned}
p(\alpha_1^N|x_1^T) &= \sum_{t_1^N} p(\alpha_1^N, t_1^N|x_1^T) \\
&= \sum_{t_1^N} p(\alpha_1^N|t_1^N, x_1^T) \cdot p(t_1^N|x_1^T) \\
&= \sum_{t_1^N} \prod_{n=1}^{N} p(\alpha_n|\alpha_1^{n-1}, t_1^N, x_1^T) \cdot p(t_n|t_1^{n-1}, x_1^T) \\
&\stackrel{?}{=} \sum_{t_1^N} \prod_{n=1}^{N} \underbrace{p(\alpha_n|\alpha_1^{n-1}, t_{n-1}, t_n, x_1^T)}_{\text{NN-based posterior}} \cdot \underbrace{p(t_n|t_{n-1})}_{\text{length model}}
\end{aligned}
$$

Inverted Search

Discussion:

- **inverted search**, as times are aligned to triphone (state) labels, instead of vice versa.

$$p(\alpha_1^N | x_1^T) = \sum_{t_1^N} \prod_{n=1}^{N} \underbrace{p(\alpha_n | \alpha_1^{n-1}, t_{n-1}, t_n, x_1^T)}_{\text{NN-based posterior}} \cdot \underbrace{p(t_n | t_{n-1})}_{\text{length model}}$$

- Symbol by symbol hypothesis generation.
- Language model integrated into state posterior.

Open questions:

- How to model state posterior? - not necessearily the same, as in hybrid approach: here state posterior covers multiple time frames.
- Length model? - existing HMM based work less successful.
- Where are the words? - word sequence determines state sequence: Effectively states represent subwords (or even words itself!).
- How to fit in (**separately trained**) language model?

Outline

Specific Work

Acoustic Modeling of Raw Time Signal [Golik & Tüske$^+$ 2015]

- large effort went into **feature engineering** for DNNs (e.g. [Seide & Li$^+$ 2011, Yu & Yao$^+$ 2013], ...)
- previous work [Tüske & Golik$^+$ 2014] showed:
  - a simple fully connected 12-hidden-layers DNN performs well even **without any feature extraction**
  - WER: 22.1% (MFCC) vs. 25.5% (raw time signal)
  - first layer weights learned impulse responses of band pass filters
  - the learned filter bank roughly resembles manually defined filter bank
- **convolutional neural network (CNN)** is a natural tool that combines learning a filter bank and acoustic modeling
- research questions:
  - how much do CNNs reduce the performance gap to hand-crafted features?
  - how can we interpret the learned weights?

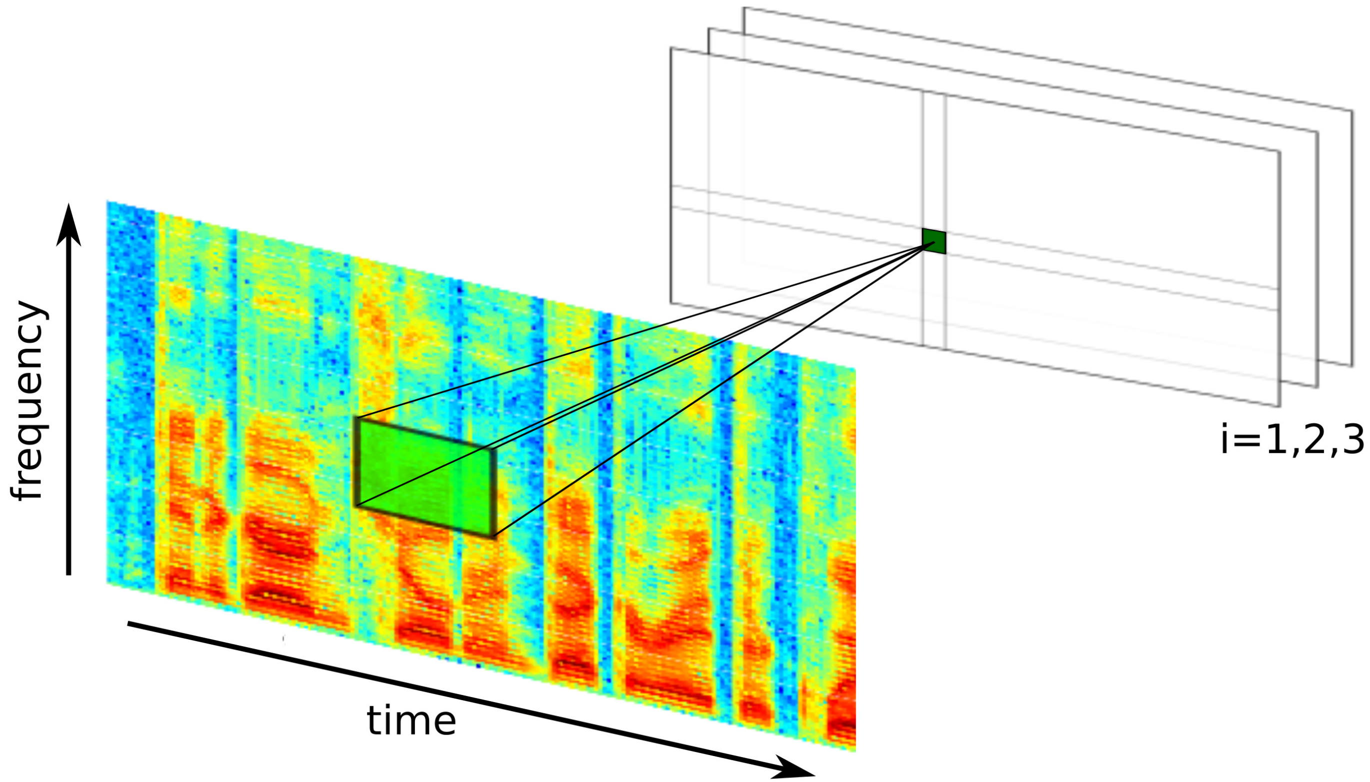
Convolutional neural networks

- CNNs and were introduced about 25 years ago [LeCun & Boser+ 1989]
- today: state-of-the-art in computer vision
  ([Krizhevsky & Sutskever+ 2012, Jaderberg & Simonyan+ 2015])
- applied to speech recognition tasks by [Abdel-Hamid & Mohamed+ 2012]:
  2D filters perform convolution on a “spectrogram”
- convolution on raw time signal: 1D operation along time axis only
- output of convolutional unit i at position m:

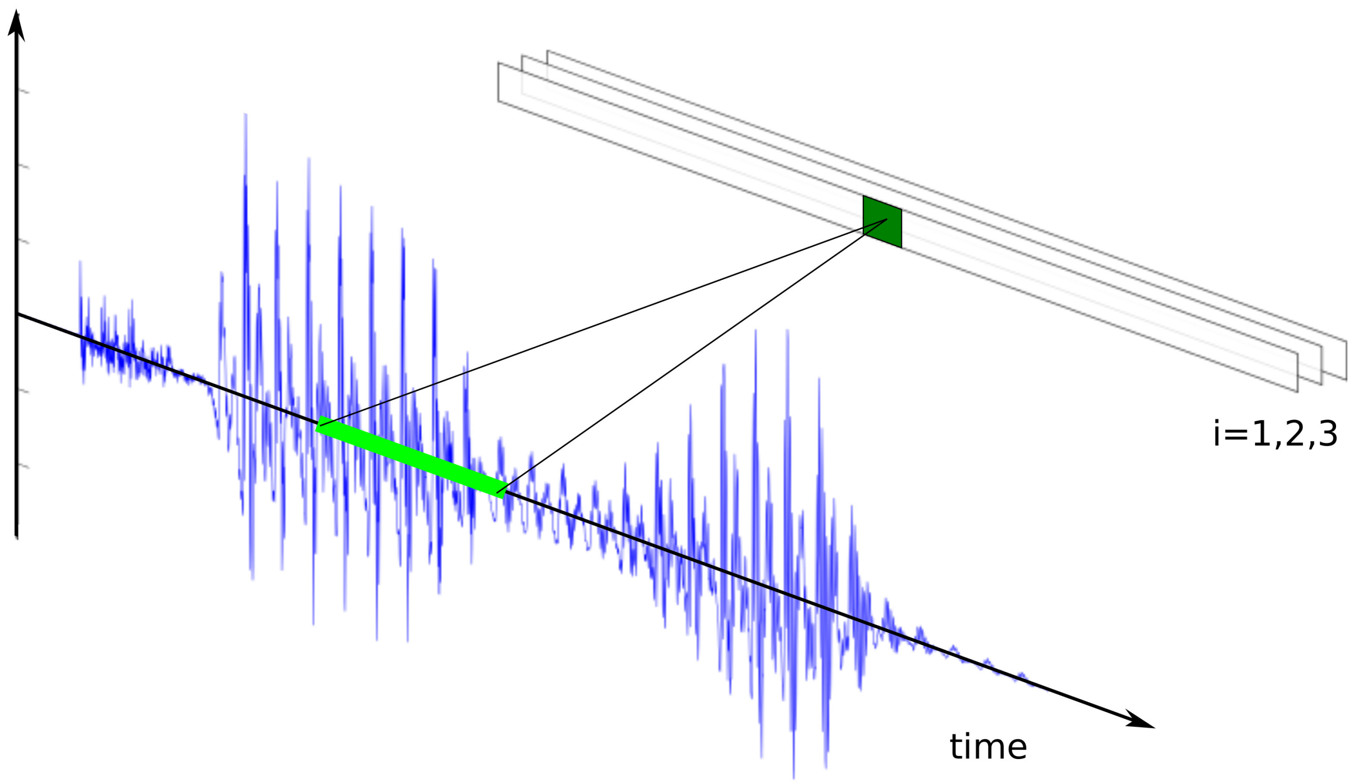
$$y_{i,m} = \sigma\left(\sum_{j=m}^{m+k-1} w_{i,j-m} x_j + b_i\right)$$

  - x_j are the PCM samples
  - $\{w_{i,\cdot}, b_i\}$: trainable parameters shared across all positions in the input
  - k is the length of the impulse response of a filter
- temporal sub-sampling by shifting m in steps of 32 and max pooling

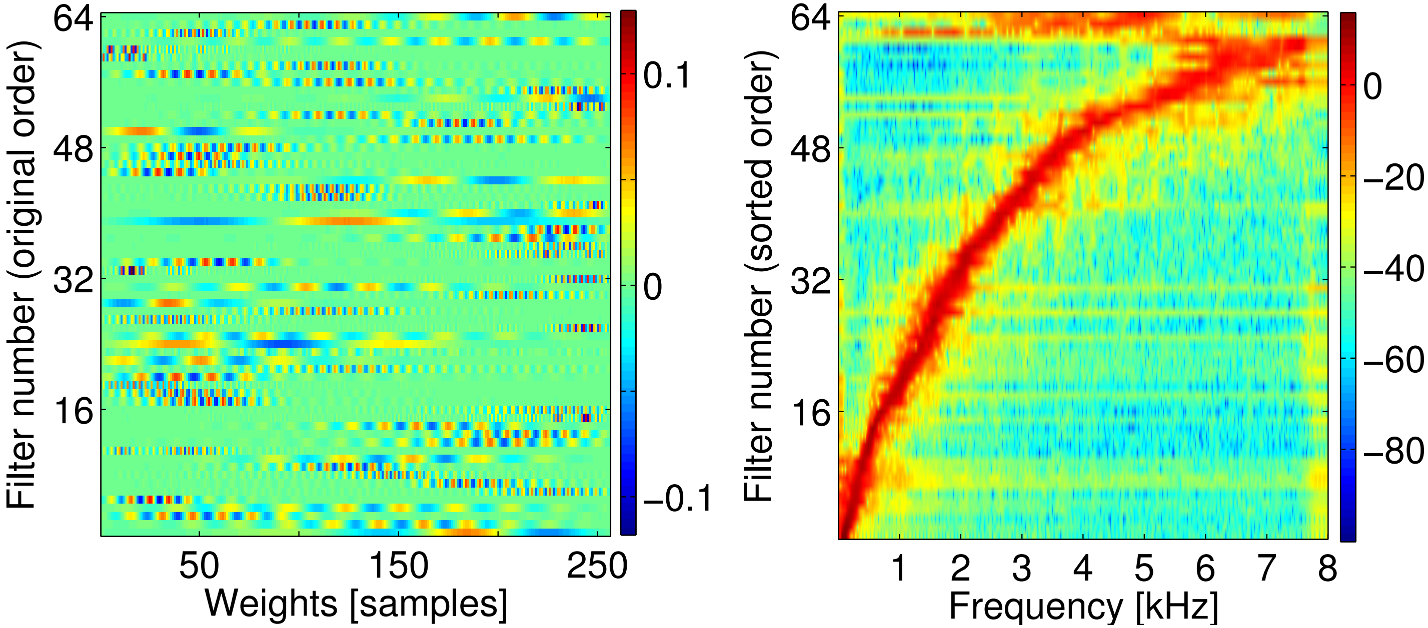
2D convolution in time/frequency (for ASR)



1D convolution in time only



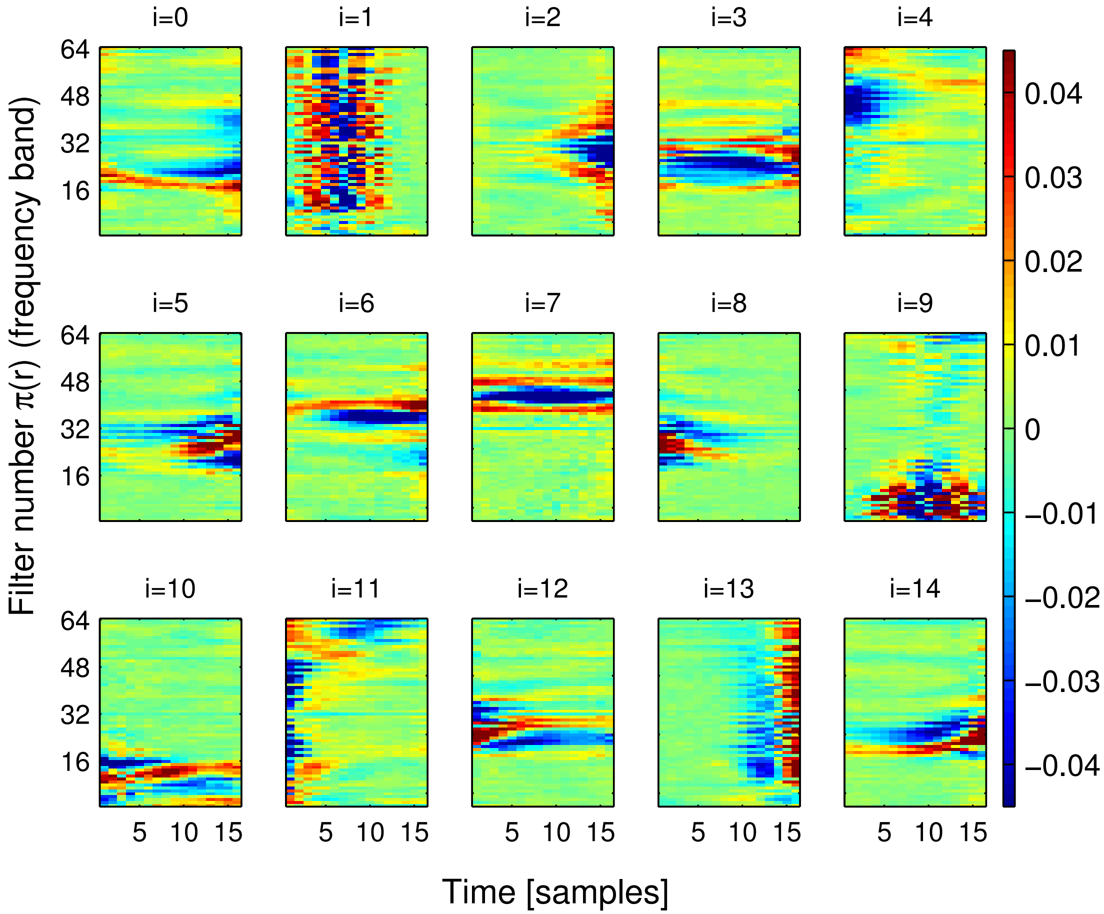
Learned weights: first convolutional layer



- Thus, after reordering, the output of the first convolutional layer approximates critical band energies

Learned weights: second convolutional layer

- reordered weights of some of the 128 filters i in the 2nd convolutional layer
- vertical: frequency axis, horizontal: time axis
- dynamic patterns in both time and frequency



Conclusions

- training on **raw time signal** works surprisingly well
- convolutional layers improve ASR performance over fully-connected layers
- the gap to MFCC's performance reduces from 15% to 6% relative WER

model	input	WER [%]
DNN	MFCC	22.1
	raw time signal	25.5
CNN		23.4

- non-stationary patterns can be captured precisely
- first and second layer weights can be interpreted as filters in time/frequency

- for sufficient amounts of training data, models trained on the raw time signal can even outperform standard preprocessing, even for multichannel scenarios [Sainath & Weiss^{+} 2015]

Outline

Specific Work

Multilingual MLP Features [Tüske & Schlüter+ 2013]

- Exploitation of language independent information is viable:
  - Cross-lingual application of MLP features can improve performance [Stolcke & Grézl+ 2006].
  - Training MLP on target language usually better for similar amount of training data.

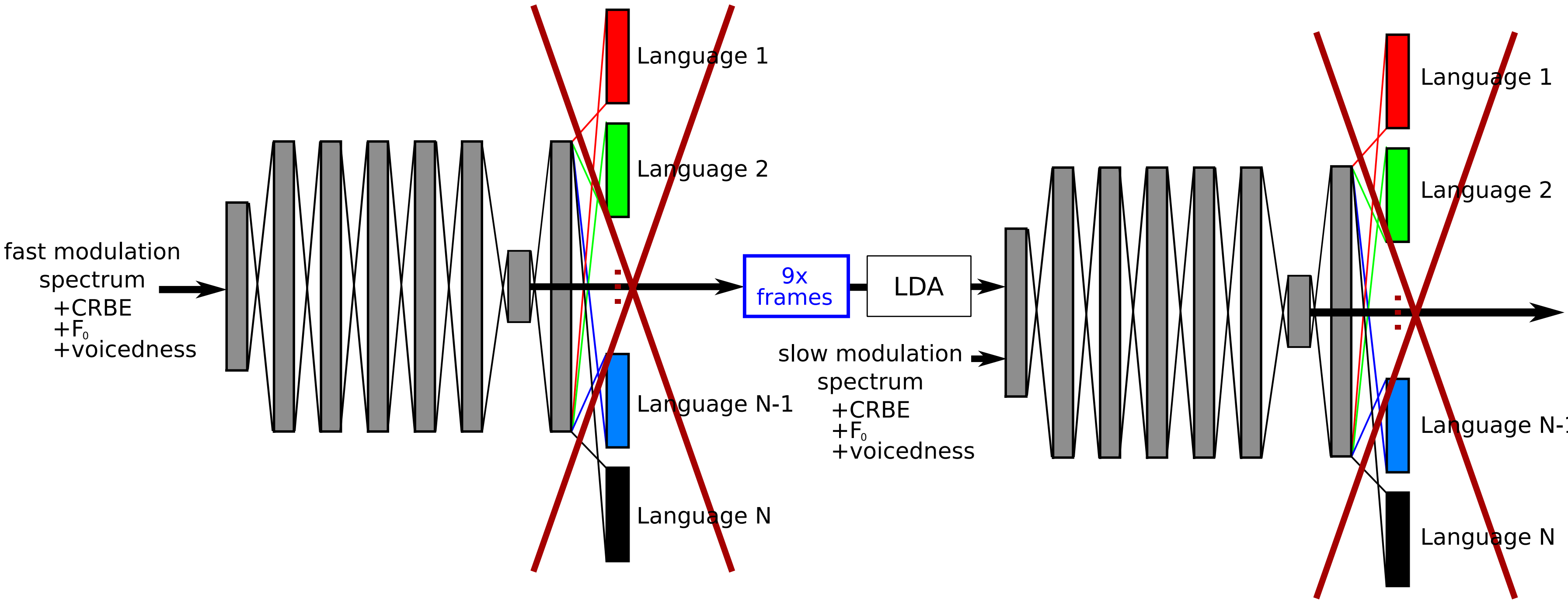
- Training MLPs on multiple languages
  - Spoken languages are based on the same speech production mechanisms.
  - Allows parameter sharing between languages.
  - Idea: share common bottleneck layer for multiple languages.
  - Robust feature: better portability to new language.
  - Exploits data available in other/multiple languages.
  - Serves as initialization prior to additional language specific training/fine-tuning.

Multilingual Bottleneck MLP

Handling multiple targets:

- Phone set incl. **language id** [Grézl & Karafiát+ 2011]:
  - NN also has to learn language identification.
- Mapping to **common phone set** [Schultz & Waibel 2001]:
  - Knowledge based (e.g IPA, SAMPA):
    often ambiguous due to simplified lexicons.
  - Data-driven.
- **Language dependent output layer** [Scanzio & Laface+ 2008]:
  - No need to map phonetical units to common set.
  - Error back-propagation only from the active output.
  - Related to multi-task training.

Architecture of Multilingual Hierarchical Bottleneck MLP



Specific Work

Experiments - Quaero, Small Scale

- Experimental setup
  - Target task: French.
  - 50h of speech per language (balanced corpus size)
  - Data available for French (FR), English (EN), German (DE), Polish (PL)
  - Tandem/bottleneck approach
  - GMM: 4500 tied-states for each language
  - Shallow BN-MLPs (7000,60,7000), with phoneme targets
  - Speaker independent WER reported on Eval11
- Effect of number of languages

training languages				WER
FR	EN	PL	DE	[%]
✓				22.2
✓			✓	21.6
✓		✓	✓	21.5
✓	✓	✓	✓	**21.1**

  - The more languages, the better.

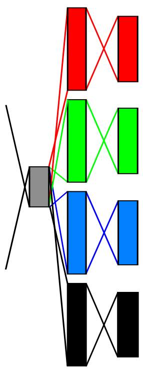
Effect of Multi- and Unilingual Bottleneck Features

input features	WER [%] for languages: FR	EN	DE	PL
MFCC	25.5	31.6	25.0	18.9
$+\mathrm{BN}_{\mathrm{uni}}$	22.2	26.8	21.3	15.7
$+\mathrm{BN}_{\mathrm{multi}}$	21.1	24.9	20.1	15.4

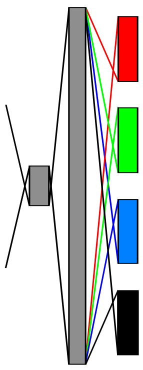
- All languages benefit from multilingual bottleneck features $\mathrm{BN}_{\mathrm{multi}}$.
- 2–5% rel. improvement over unilingual features $\mathrm{BN}_{\mathrm{multi}}$.
- 17-21% overall rel. improvement over MFCC baseline.

Experiments - Quaero, Large Scale

- Speaker adaptative training.
- Unbalanced corpus sizes for languages: 100h to 300h.
- Deep NN structure and context-dependent NN targets.
- Tuning the language dependent part of the MLP:
  - Language dependent hidden layer
    increases no. of parameters, but same training time
    last layer: huge, but **block diagonal** weight matrix (8000x6000)



  - Large, but common hidden layer
    increases no. of parameters even further, slower training
    last layer: huge **full** weight matrix (8000x6000)



Experiments - Quaero, Large Scale

	WER [%] for languages:			
intput features	FR	EN	DE	PL
MFCC	21.6	26.4	21.4	15.9
$+\mathrm{BN}_{\mathrm{uni}}$	17.3	19.7	17.2	12.3
$+\mathrm{BN}_{\mathrm{multi}}$	17.0	19.2	16.3	12.1
$+$deep $\mathrm{BN}_{\mathrm{uni}}$	16.7	18.8	16.8	12.1
$+$deep $\mathrm{BN}_{\mathrm{multi}}$	16.2	18.1	15.7	11.7
w/lang. dep. hidden layer	16.3	18.2	15.7	11.7
w/large lang. indep. hidden layer	16.0	17.7	15.4	11.7

- Multilingual always outperform monolingual model.
- Deep structure increases margin between uni- and multilingual:
  relative improvement in WER: shallow BN: 2–5%, **deep BN: 3–7%**.
- 25–30% rel. WER impr. over speaker adaptive MFCC baseline.

Multilingual Hybrid NN: Quaero English

- Hybrid NN acoustic model with recent improvements.
  - 50 dim. gammatone input features, 17 frames context.
  - 12 hidden layers, 2000 nodes each.
  - Activation function: rectified linear units.
  - Low-rank factorized 12k output using 512 dim. linear BN.
  - WER reported on Quaero Eval corpus, 250h training data.

	Model	Criterion	WER [%]
unilingual	GMM	MPE	26.2
	hybrid NN	MPE	16.2
multilingual	hybrid NN	CE	17.3
	+fine-tuning	CE	16.7
		MPE	15.6

- Initial multilingual hybrid NN results w/o further training.
- Fine tuning: further optimization on target data.
- Still $\sim$4% rel. improvement by multilingual training.

Outline

Log-Linear Interpolation of Multi-Domain Neural Network LM

[Tüske & Irie+ 2016]

- Usual approach: linear interpolation of count LMs trained on different domains/data sets.
  - Interpolation weights optimized on target domain validation set.
  - Optimized using expectation maximization (EM) algorithm.
  - Count models are suited to be linearly combined into one single model (with union of n-grams and recomputing back-off weights)

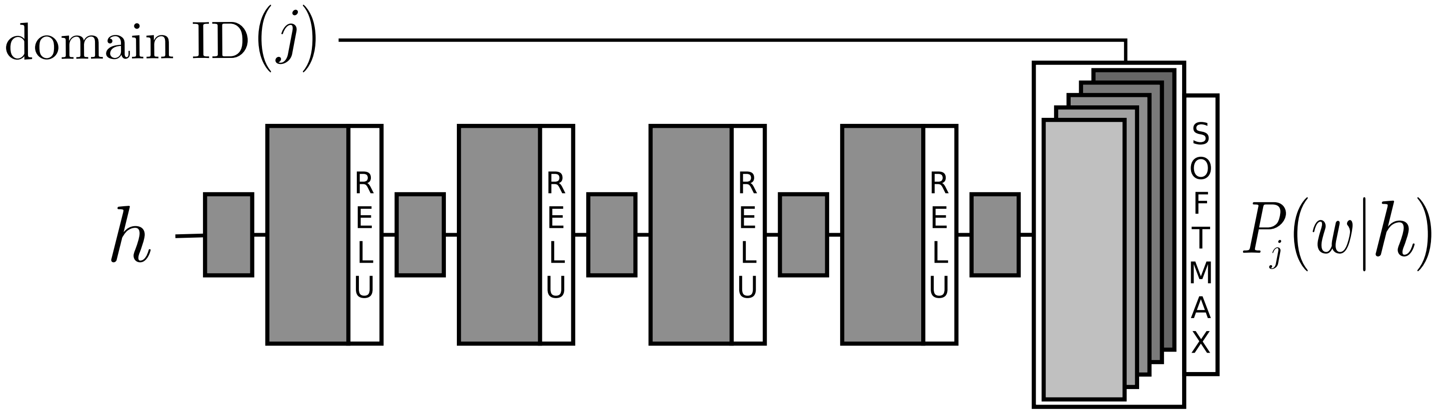
- Goal: combination approach for neural network LMs.
  - Aiming at **single model** after interpolation of neural network LMs.
  - Linear interpolation not straightforward for NN LMs to obtain single model.

    Log-Linear combination fits better;

- Initial investigation using feed-forward NN LMs.

Specific Work

Joint Model



- Multiple posterior estimates
  - Active output: selected by the domain of the input vector
  - Hidden layers are shared between the domains
  - Shared vocabulary, common softmax

- Log-linear combination to obtain single overall neural network LM:
  - Leads to weighted sum of domain specific output layers.
  - Weighted sum of softmax outputs can rewritten as a single softmax output layer.

Experimental Results: Perplexities

- Training corpus: 3B words, 11 domains (Gigaword, BN/BC, TED, IWSLT, ...)
  - 50M and 2M best matching subset selected for fine-tuning
- KN 4-gram: 132.7 PPL after interpolation
- 50M LSTM-RNN: 100.5

- Retraining only multi-domain output (**log-linear!**) on the best BN, and interpolation: PPL **92.0**

LM	multi domain	log-lin. interp.	fine-tuning 50M	fine-tuning 2M	PPL
50M					110.5
				×	109.0
3B					129.0
			×	×	96.2
	×				133.1
	×		×	×	95.7*
	×	×			117.6
	×	×	×	×	**94.3**

*using the best matching output

Specific Work

Experimental Results: WER

- Lattice generation with count model
- Lattice rescoring using `rwthlm` [Sundermeyer & Alkhouli+ 2014]
  - Traceback lattice approximation
  - Linear-interpolation of NN LM and count LM (KN 4-gram)
- Measuring word error rate
  - Acoustic model: 12-layer multilingual BN (800h), fine tuned on 250h BN/BC target data
  - Standard Viterbi (Vi.) and confusion network (CN) decoding of the lattices

Language Model	Dev			Eval		
	PPL	Vi.	CN	PPL	Vi.	CN
KN4	132.7	12.6	12.3	133.4	15.4	15.0
+ 50M FFNN	96.5	11.4	11.1	95.0	14.2	13.8
+ 3B, fine-tune	89.6	10.9	10.7	88.0	13.7	13.4
+ Multi-domain,log-lin,fine-tune	88.5	10.8	9.1	87.0	13.7	13.5
+ 50M LSTM	91.6	10.9	9.0	91.0	13.7	13.5

Outline

Specific Work

Tandem vs. Hybrid - Integrating GMM into DNN [Tüske & Tahir+ 2015]

- State-of-the-art acoustic models (AM) are
  - Tandem acoustic models
    * Gaussian Mixture Models (GMM) are trained on the output of a neural network based features
    * Probabilistic or bottleneck (BN) tandem approach [Hermansky & Ellis+ 2000, Grézl & Karafiát+ 2007]
    * **Joint training**, e.g. in [Paulik 2013]
  - Hybrid models
    * Proposed in the early 90's [Bourlard+Morgan:1993]
    * Estimates state posterior probabilities $p(s|x)$ directly
    * BN layer to train efficiently on huge number of states [Sainath & Kingsbury+ 2013]
- After careful optimization they show similar performance
- **Goal**: convert tandem into hybrid neural network representation [Tüske & Tahir+ 2015]
- **Idea**: rewrite GMM to equivalent log-linear model [Anderson 1982, Heigold & Wiesler+ 2010]
  $\rightarrow$ **softmax NN layer**

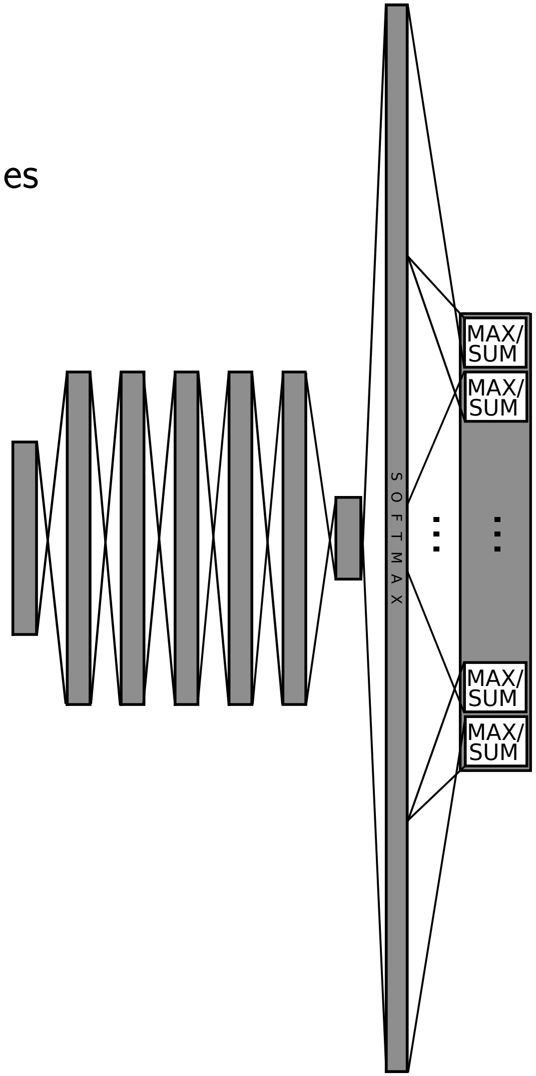
Specific Work

Joint GMM and Bottleneck DNN Training

- GMM with pooled covariance is a softmax layer with hidden variables
- Maximum approximation, for fast score calculation:

$$\frac{\sum_i \exp(w_{si}^T y + b_{si})}{Z(y)} \approx \left.\frac{\exp(w_{s\hat{\imath}}^T y + b_{s\hat{\imath}})}{Z(y)}\right|_{\hat{\imath}=\operatorname{argmax}_i (w_{si}^T y + b_{si})}$$

- No need for special element to implement:
  - sum- or max-pooling
- Efficient softmax is crucial (low-rank factorization; GPU)
  - GMM of 4500 states after 8 splits: $\sim$ 1 million nodes
- Joint training of BN and GMM:
  - Maximum likelihood training of GMM on BN features
  - Convert to LMM
  - Start the joint training
- Remark: maximum approximation with given labeling (s,i) same as classical hybrid, E-M style training is also possible



ASR Experiments

- Task: Quaero English (250h BC/BN)
- MLP structure:
  - 12 hidden layers
  - 50 dimensional Gammatone input

System	low rank	joint training	#output	#param.	split	criterion	WER [%]	
							dev	eval
Hybrid	no	–	4.5k	54.7M	-	CE	13.3	18.1
	yes			49.0M			13.5	18.2
			12.0k	52.8M			13.0	17.7
BN tandem	–	no	4.5k	613.0M	8	ML	14.2	19.0
		yes		83.5M	4	CE	13.1	17.8

- Same results with less tied-triphone states
- Smaller lexical prefix-tree

Outline

Conclusions

Statistical approach

- four key ingredients:
  - choice of performance measure: errors at string, word, phoneme, frame level
  - probabilistic models at these levels and the interaction between these levels
  - training criterion along with an optimization algorithm
  - Bayes decision rule along with an efficient implementation

- about recent work on artificial neural nets (2009-15):
  - significant improvements by deep MLPs and LSTM-RNNs
  - they provide one more type of probabilistic models

- long-term research topics at RWTH:
  - training criteria and error rates (at frame, phoneme, word, sentence levels)
  - open lexicon ASR: any letter sequence can be recognized
  - (fully) unsupervised training: without *any* transcribed training data

Future Challenges

- specific future challenges for statistical approach (incl. NNs) in general:
  - complex mathematical model that is difficult to analyze
  - questions: can we find suitable mathematical approximations with more explicit descriptions of the dependencies and level interactions and of the performance criterion (error rate)?

- specific challenges for artificial neural networks:
  - methods with better convergence?
  - can the HMM-based alignment mechanism be replaced?
  - can we find NNs with more explicit probabilistic structures?

Conclusions

Questions and Interpretations

- Do the NNs discover dependencies that we cannot model explicitly?
- Is it a better way of smoothing that makes the NN better?
- Is it the use of crossvalidation that makes NNs succesful?
- ...

Thank you for your attention

Any questions?

Outline

References

- O. Abdel-Hamid, A.R. Mohamed, H. Jiang, G. Penn: "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition," *IEEE Intern. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4277–4280, Mar. 2012.
- J. Anderson: "Logistic Discrimination," *Handbook of Statistics 2*, P.R. Krishnaiah and L.N. Kanal, eds., pp. 169–191, North-Holland, 1982.
- D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, Y. Bengio: "End-to-End Attention-based Large Vocabulary Speech Recognition," *arXiv preprint*, arXiv:1508.04395, Aug. 2015.
- L. R. Bahl, F. Jelinek, R. L. Mercer: A Maximum Likelihood Approach to Continuous Speech Recognition. IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 5, pp. 179-190, March 1983.

References

- L. R. Bahl, P. F. Brown, P. V. de Souza, R. L. Mercer: Maximum mutual information estimation of hidden Markov parameters for speech recognition. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), Tokyo, pp.49-52, April 1986.
- Y. Bengio, R. De Mori, G. Flammia, R. Kompe: “Global optimization of a neural network - hidden markov model hybrid,” *IEEE Transactions on Neural Networks*, Vol. 3, pp. 252–259, Mar. 1991.
- Y. Bengio, R. Ducharme, P. Vincent: A neural probabilistic language model. Advances in Neural Information Processing Systems (NIPS), pp. 933-938, Denver, CO, Nov. 2000.
- Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle: “Greedy layer-wise training of deep networks,” *Advances in neural information processing systems*, Vol. 19: Proceedings of the 2016 conference, pp. 153–160, 2007.

References

- Y. Bengio, P. Simard, P. Frasconi: “Learning long-term dependencies with gradient descent is difficult,” *IEEE Transactions on Neural Networks*, Vol. 5, No. 2, pp. 157–166, 1994.
- R. Botros, K. Irie, M. Sundermeyer, H. Ney: On Efficient Training of Word Classes and Their Application to Recurrent Neural Network Language Models. Interspeech, pp.1443-1447, Dresden, Germany, Sep. 2015.
- H. Bourlard, C. J. Wellekens: 'Links between Markov Models and Multilayer Perceptrons', in D.S. Touretzky (ed.): "Advances in Neural Information Processing Systems I", Morgan Kaufmann Pub., San Mateo, CA, pp.502-507, 1989.
- H. Bourlard, N. Morgan: *Connectionist Speech Recognition: a Hybrid Approach*, Kluwer Academic Publishers, Norwell, MA, 1993.

References

- J. S. Bridle: Probabilistic Interpretation of Feedforward Classification Network Outputs with Relationships to Statistical Pattern Recognition, in F. Fogelman-Soulie, J. Herault (eds.): 'Neuro-computing: Algorithms, Architectures and Applications', NATO ASI Series in Systems and Computer Science, Springer, New York, 1989.
- P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, R. L. Mercer: Mathematics of Statistical Machine Translation: Parameter Estimation. Computational Linguistics, Vol. 19.2, pp. 263-311, June 1993.
- M.A. Castano, E. Vidal, F. Casacuberta: Inference of stochastic regular languages through simple recurrent networks. IEE Colloquium on Grammatical Inference: Theory, Applications and Alternatives, pp. 16/1-6, Colchester, UK, April 1993.

References

- M. Castano, F. Casacuberta: A connectionist approach to machine translation. European Conf. on Speech Communication and Technology (Eurospeech), pp. 91–94, Rhodes, Greece, Sep. 1997.
- M. Castano, F. Casacuberta, E. Vidal: Machine translation using neural networks and finite-state models. Int. Conf. on Theoretical and Methodological Issues in Machine Translation (TMI), pp. 160-167, Santa Fe, NM, July 1997.
- W. Chan, N. Jaitly, Q. V. Le, O. Vinyals: "Listen, Attend and Spell," *arXiv preprint*, arXiv:1508.01211, Aug. 2015.
- X. Chen, A. Eversole, G. Li, D. Yu, F. Seide: "Pipelined Back-Propagation for Context-Dependent Deep Neural Networks," *Interspeech*, pp. 26–29, Portland, OR, Sep. 2012.
- K. Cho, B. Gulcehre, D. Bahdanau, F. Schwenk, Y. Bengio: "Learning phrase representations using RNN encoderÂdecoder for statistical machine translation,"

References

Conf. on Empirical Methods in Natural Language Processing (EMNLP), pp. 1724–1734, Doha, Qatar, Oct. 2014,

- G. Cybenko: "Approximation by Superpositions of a Sigmoidal Function," *Mathematics of Control, Signals and Systems*, Vol. 2, No. 4, pp. 303–314, 1989.
- G. E. Dahl, D. Yu, L. Deng, A. Acero: Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition. IEEE Tran. on Audio, Speech and Language Processing, Vol. 20, No. 1, pp. 30-42, Jan. 2012.
- J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, Q. Le, A. Ng: "Large Scale Distributed Deep Networks," in F. Pereira, C. Burges, L. Bottou, K. Weinberger (eds.): *Advances in Neural Information Processing Systems (NIPS)*, pp. 1223–1231, Nips Foundation, http://books.nips.cc, 2012.

References

- J. Devlin, R. Zbib, Z. Huang, T. Lamar, R. Schwartz, J. Makhoul: Fast and Robust Neural Network Joint Models for Statistical Machine Translation. Annual Meeting of the ACL, pp. 1370–1380, Baltimore, MA,, June 2014.
- P. Dreuw, P. Doetsch, C. Plahl, G. Heigold, H. Ney: "Hierarchical hybrid MLP/HMM or rather MLP features for a discriminatively trained Gaussian HMM: A comparison for offline handwriting recognition," *Intern. Conf. on Image Processing*, 2011.
- S. Espana-Boquera, M. Castro-Bleda, J. Gorbe-Moya, F. Zamora-Martinez: "Improving offline handwritten text recognition with hybrid HMM/ANN models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 33, No. 4, pp. 767–779, Apr. 2011.

References

- J. Fritsch, M. Finke, A. Waibel: Adaptively Growing Hierarchical Mixtures of Experts. NIPS, Advances in Neural Information Processing Systems 9, MIT Press, pp. 459-465, 1997.
- K. Fukushima: “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,” *Biological Cybernetics*, Vol. 36, No. 4, pp. 193–202, April 1980.
- F. A. Gers, J. Schmidhuber, F. Cummin: Learning to forget: Continual prediction with LSTM. Neural computation, Vol 12, No. 10, pp. 2451-2471, 2000.
- F. A. Gers, N. N. Schraudolph, J. Schmidhuber: Learning precise timing with LSTM recurrent networks. Journal of Machine Learning Research, Vol. 3, pp. 115-143, 2002.

References

- X. Glorot, Y. Bengio: "Understanding the difficulty of training deep feedforward neural networks," *Int. Conf. on Artificial Intelligence and Statistics*, pp. 249–256, 2010.
- I. Goodfellow, Y. Bengio, A. Courville: *Deep Learning*, Book in preparation for MIT Press, `http://www.deeplearningbook.org`, 2016.
- J. Goodman: "Classes for fast maximum entropy training," *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 1, pp. 561–564, Salt Lake City, UT, May 2001.
- P. Golik, P. Doetsch, H. Ney: "Cross-Entropy vs. Squared Error Training: a Theoretical and Experimental Comparison," *Interspeech*, pp. 1756–1760, Lyon, France, Aug 2013.

References

- P. Golik, Z. Tüske, R. Schlüter, H. Ney: "Convolutional Neural Networks for Acoustic Modeling of Raw Time Signal in LVCSR," *Interspeech*, pp. 26-30, Dresden, Germany, September 2015.
- A. Graves, H. Bunke, S. Fernandez, M. Liwicki, J. Schmidhuber: "Unconstrained online handwriting recognition with recurrent neural networks," In Advances in Neural Information Processing Systems, Vol. 20. MIT Press, 2008.
- A. Graves, S. Fernández, F. Gomez, J. Schmidhuber: "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," *Int. Conf. on Machine Learning (ICML)*, pp. 369–376, Helsinki, Finland, June 2006.
- F. Grézl, M. Karafiát, M. Janda: "Study of probabilistic and bottle-neck features in multilingual environment," *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 359–364, Waikoloa, HI, Dec. 2011.

References

- F. Grézl, M. Karafiát, S. Kontár, J. Cernocký: "Probabilistic and Bottle-neck Features for LVCSR of Meetings," *IEEE Intern. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 757–760, Honolulu, HI, April 2007.
- G. Heigold, E. McDermott, V. Vanhoucke, A. Senior, M. Bacchiani: "Asynchronous stochastic optimization for sequence training of deep neural networks," *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5587–5591, Florence, Italy, May 2014.
- G. Heigold, S. Wiesler, M. Nussbaum, P. Lehnen, R. Schlẗer, H. Ney: "Discriminative HMMs, Log-Linear Models, and CRFs: What is the Difference?" *IEEE Intern. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 5546–5549, Dallas, TX, March 2010.
- H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," Journal of the Acoustical Society of America, Vol. 8, No. 4, pp. 1738–1752, 1990

References

- H. Hermansky, D. Ellis, S. Sharma: “Tandem Connectionist Feature Extraction for Conven- tional HMM Systems,” *IEEE Intern. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Vol. 3, pp. 1635–1638, Istanbul, Turkey, June 2000.
- H. Hermansky, P. Fousek: “Multi-resolution RASTA filtering for TANDEM-based ASR,” Interspeech, pp. 361–364, Lisbon, Portugal, Sept. 2005.
- G. Hinton, S. Osindero, Y. Teh: “A Fast Learning Algorithm for Deep Belief Nets,” *Neural Computation*, Vol. 18, No. 7, pp. 1527Â-1554, July 2006.
- G. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R. Salakhutdinov: “Improving neural networks by preventing co-adaptation of feature detectors,” *arXiv preprint* arXiv:1207.0580, 2012.
- J. Hochreiter: *Untersuchungen zu dynamischen neuronalen Netzen*, diploma thesis, Computer Science, TU München, June 1991.

References

- S. Hochreiter, J. Schmidhuber: Long short-term memory. Neural Computation, Vol. 9, No. 8, pp. 1735–1780, Nov. 1997.
- M. Jaderberg, K. Simonyan, A. Zisserman: "Spatial Transformer Networks," *Advances in Neural Information Processing Systems*, pp. 2017–2025, 2015.
- B. Kingsbury: "Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling," *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3761–3764, Taipei, Taiwan, April 2009.
- B. Kingsbury, T. Sainath, H. Soltau: "Scalable Minimum Bayes Risk Training of Deep Neural Network Acoustic Models Using Distributed Hessian-free Optimization," *Interspeech*, Portland, OR, Sep. 2012.
- R. Kneser, H. Ney: "Improved clustering techniques for class-based statistical language modelling," *Eurospeech*, Vol. 93, pp. 973–976, Berlin, Germany, Sep. 1993.

References

- P. Koehn, F. J. Och, D. Marcu: Statistical Phrase-Based Translation. HLT-NAACL 2003, pp. 48-54, Edmonton, Canada, May-June 2003.
- M. Kozielski, P. Doetsch, H. Ney: "Improvements in RWTH's system for off-line handwriting recognition," *12th International Conference on Document Analysis and Recognition (ICDAR)*, pp. 935–939, Buffalo, NY, Aug. 2013.
- A. Krizhevsky, I. Sutskever, G. Hinton: "Imagenet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.
- H.S. Le, A. Allauzen, F. Yvon: Continuous space translation models with neural networks. NAACL-HLT 2012, pp. 39-48, Montreal, QC, Canada, June 2002.
- Y. LeCun, Y. Bengio: Word-level training of a handwritten word recognizer based on convolutional neural networks. Int. Conf. on Pattern Recognition, Jerusalem, Israel, pp. 88-92, Oct. 1994.

References

- Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, L. Jackel: "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Computation*, Vol. 1, No. 4, pp. 541–551, 1989.
- V. Manohar, D. Povey, S. Khudanpur: "Semi-supervised Maximum Mutual Information Training of Deep Neural Network Acoustic Models," Interspeech, Dresden, Germany, Sept. 2015.
- T. Mikolov, M. Karafiat, L. Burget, J. ernocky, S. Khudanpur: Recurrent neural network based language model. Interspeech, pp. 1045-1048, Makuhari, Chiba, Japan, Sep. 2010.
- A. Mohamed, G. Dahl, G. Hinton: "Acoustic Modeling Using Deep Belief Networks," *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 20, No. 1, pp. 14–22, Jan. 2012.

References

- V. Nair, G. Hinton: “Rectified Linear Units Improve Restricted Boltzmann Machines,” *Intern. Conf. on Machine Learning (ICML)*, pp. 807–814, Haifa, Israel, June 2010.
- M. Nakamura, K. Shikano: A Study of English Word Category Prediction Based on Neural Networks. ICASSP 89, p. 731-734, Glasgow, UK, May 1989.
- H. Ney, U. Essen, R. Kneser: “On the estimation ofsmall probabilities by leaving-one-out,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 12, pp. 1202–1212, 1995.
- F. J. Och, H. Ney: A Systematic Comparison of Various Alignment Models. *Computational Linguistics,* Vol. 29, No. 1, pp. 19-51, March 2003.
- F. J. Och, H. Ney: The Alignment Template Approach to Statistical Machine Translation. *Computational Linguistics*, Vol. 30, No. 4, pp. 417-449, Dec. 2004.

References

- F. J. Och, C. Tillmann, H. Ney: Improved Alignment Models for Statistical Machine Translation. Joint ACL/SIGDAT Conf. on Empirical Methods in Natural Language Processing and Very Large Corpora, College Park, MD, pp. 20-28, June 1999.
- M. Paulik: “Lattice-based training of bottleneck feature extraction neural networks,” *Interspeech*, 2013.
- D. Povey, P. Woodland: “Minimum phone error and I- smoothing for improved discriminative training,” *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Vol. 1, pp. 105–108, Orlando, FL, May 2002.
- A. J. Robinson: An Application of Recurrent Nets to Phone Probability Estimation. IEEE Trans. on Neural Networks, Vol. 5, No. 2, pp. 298-305, March 1994.

References

- T. Robinson, M. Hochberg, S. Renals: "IPA: Improved Phone Modelling with Recurrent Neural Networks," *IEEE Intern. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Vol. I, pp. 37–40, Adelaide, Australia, Apr. 1994.
- D. Rumelhart, G. Hinton, R. Williams: "Learning Representations By Back-Propagating Errors," Nature Vol. 323, pp. 533–536, Oct. 1986.
- T. Sainath, B. Kingsbury, V. Sindhwani, E. Arisoy, B. Ramabhadran: "Low-rank matrix factorization for deep neural network training with high-dimensional output targets," *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013.
- T.N. Sainath, , R.J. Weiss, K.W. Wilson, A. Narayanan, M. Bacchiani: "Speaker Location and Microphone Spacing Invariant Acoustic Modeling from Raw Multichannel Waveforms," *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 30–36, Dec. 2015.

References

- S. Scanzio, P. Laface, L. Fissore, R. Gemello, F. Mana: “On the Use of a Multilingual Neural Network Front-End,” *Interspeech*, pp. 2711–2714, Brisbane, Australia, Sept. 2008.
- R. Schlüter, M. Nussbaum-Thom, H. Ney: Does the Cost Function Matter in Bayes Decision Rule? IEEE Trans. PAMI, No. 2, pp. 292–301, Feb. 2012.
- R. Schlüter, I. Bezrukov, H. Wagner, H. Ney: “Gammatone Features and Feature Combination for Large Vocabulary Speech Recognition,” *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 649–652, Honolulu, HI, April 2007.
- T. Schultz, A. Waibel: “Language-independent and language-adaptive acoustic modeling for speech recognition,” *Speech Communication*, Vol. 35, No. 1-2, pp. 31–51, Aug. 2001.

References

- H. Schwenk: Continuous space language models. Computer Speech and Language, Vol. 21, No. 3, pp. 492–518, July 2007.
- H. Schwenk: Continuous Space Translation Models for Phrase-Based Statistical Machine Translation. 24th Int. Conf. on Computational Linguistics (COLING), Mumbai, India, pp. 1071–1080, Dec. 2012.
- H. Schwenk , M. R. Costa-jussa, J. A. R. Fonollosa: Smooth bilingual n-gram translation. Joint Conf. on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pp. 430–438, Prague, June 2007.
- H. Schwenk, D. Déchelotte, J. L. Gauvain: Continuous Space Language Models for Statistical Machine Translation. COLING/ACL 2006, pp. 723–730, Sydney, Australia July 2006.

References

- F. Seide, G. Li, D. Yu: "Conversational Speech Transcription using Context-Dependent Deep Neural Networks," *Interspeech*, pp. 437–440, Florence, Italy, Aug. 2011.
- K. Simonyan, A. Zisserman: "Very Deep Convolutional Networks for Large-Scale Image Recognition," CoRR, abs/1409.1556, http://arxiv.org/abs/1409.1556, Oct. 2014.
- A. Stolcke, F. Grézl, M.-Y. Hwang, X. Lei, N. Morgan, D. Vergyri: "Cross-domain and cross-language portability of acoustic features estimated by multilayer perceptrons," 'textitIEEE Int. Conf. on Acoustics, Speech and Signal Processing, pp. 321–324, Toulouse, France, May 2006.
- H. Su, G. Li, D. Yu, F. Seide: "Error Back Propagation For Sequence Training Of Context-Dependent Deep Networks For Conversational Speech Transcription," *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Vancouver, Canada, May 2013.

References

- M. Sundermeyer, T. Alkhouli, J. Wuebker, H. Ney: Translation Modeling with Bidirectional Recurrent Neural Networks. Conf. on Empirical Methods in Natural Language Processing (EMNLP), pp. 14–25, Doha, Qatar, Oct. 2014.
- M. Sundermeyer, H. Ney, R. Schlüter: From feedforward to recurrent LSTM neural networks for language modeling. IEEE/ACM Trans. on Audio, Speech, and Language Processing, Vol. 23, No. 3, pp. 13–25, March 2015.
- M. Sundermeyer, R. Schlüter, H. Ney: LSTM neural networks for language modeling. Interspeech, pp. 194–197, Portland, OR, Sep. 2012.
- M. Sundermeyer, Z. Tüske, R. Schlüter, H. Ney: "Lattice Decoding and Rescoring with Long-Span Neural Network Language Models," *Interspeech*, pp. 661–665, Singapore, Sep. 2014.
- I. Sutskever, O. Vinyals, Q. V. Le: "Sequence to Sequence Learning with Neural Networks," *arXiv preprint*, arXiv:1409.3215, Sep. 2014.

References

- Z. Tüske, P. Golik, R. Schlüter, H. Ney: "Acoustic Modeling with Deep Neural Networks Using Raw Time Signal for LVCSR," *Interspeech*, pp. 890–894, Singapore, September 2014.
- Z. Tüske, P. Golik, R. Schlüter, H. Ney: "Speaker Adaptive Joint Training of Gaussian Mixture Models and Bottleneck Features," *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 596–603, Scottsdale, AZ, Dec. 2015.
- Z. Tüske, K. Irie, R. Schlüter, H. Ney: "Investigation on Log-Linear Interpolation of Multi-Domain Neural Network Language Model," *IEEE Inter. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 6005–6009, Shanghai, China, Mar. 2016.

References

- Z. Tüske, M. Sundermeyer, R. Schlüter, H. Ney: "Context-Dependent MLPs for LVCSR: TANDEM, Hybrid or Both?" *Interspeech*, pp. 18–21, Portland, OR, Sept. 2012.
- Z. Tüske, M. Tahir, R. Schlüter, H. Ney: "Integrating Gaussian Mixtures into Deep Neural Networks: Softmax Layer with Hidden Variables," *IEEE Intern. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 4285–4289, Brisbane, Australia, April 2015.
- Z. Tüske, R. Schlüter, H. Ney: "Multilingual Hierarchical MRASTA Features for ASR," Interspeech, pp. 2222–2226. Lyon, France, Aug. 2013.
- P. E. Utgoff, D. J. Stracuzzi: Many-layered learning. Neural Computation, Vol. 14, No. 10, pp. 2497-2539, Oct. 2002.

References

- F. Valente, H. Hermansky, "Hierarchical and parallel processing of modulation spectrum for ASR applications," *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 4165–4168, Las Vegas, NV, Mar./Apr. 2008.
- F. Valente, J. Vepa, C. Plahl, C. Gollan, H. Hermansky, R. Schlüter: "Hierarchical Neural Networks Feature Extraction for LVCSR System," Interspeech, pp. 42–45, Antwerp, Belgium, Aug. 2007.
- A. Vaswani, Y. Zhao, V. Fossum, D. Chiang: Decoding with Large-Scale Neural Language Models Improves Translation. Conf. on Empirical Methods in Natural Language Processing (EMNLP, pp. 1387–1392, Seattle, Washington, Oct. 2013.
- K. Veselý, A. Ghoshal, L. Burget, D. Povey: "Sequence-discriminative training of deep neural networks," *Interspeech*, pp. 2345–2349, Lyon, France, Aug. 2013.

References

- K. Veselý, M. Karafiát, F. Grézl: “Convolutive bottleneck network features for LVCSR,” *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 42–47, Waikoloa, HI, Dec. 2011.
- S. Vogel, H. Ney, C. Tillmann: HMM-based word alignment in statistical translation. Int. Conf. on Computational Linguistics (COLING), pp. 836-841, Copenhagen, Denmark, Aug. 1996.
- A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, K. L. Lang: Phoneme Recognition: Neural Networks vs. Hidden Markov Models. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), New York, NY, pp.107-110, April 1988.
- A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, K. Lang: “Phoneme Recognition: Neural Networks vs. Hidden Markov Models,” *IEEE Intern. Conf.*

References

on Acoustics, Speech, and Signal Processing (ICASSP), Vol. 1, pp. 107–110, Glasgow, Scotland, April 1989.

- S. Wiesler, A. Richard, R. Schlüter, H. Ney: “Mean-normalized Stochastic Gradient for Large-Scale Deep Learning,” *IEEE Intern. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 180–184, Florence, Italy, May 2014.
- D. Yu, K. Yao, H. Su, G. Li, F. Seide: “KL-Divergence Regularized Deep Neural Network Adaptation for Improved Large Vocabulary Speech Recognition,” *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7893–7897, Vancouver, Canada, May 2013.
- R. Zens, F. J. Och, H. Ney: Phrase-Based Statistical Machine Translation. 25th Annual German Conf. on AI, pp. 18–32, LNAI, Springer 2002.