Automatic Speech Recognition: State-of-the-Art in Transition

A Neural Paradigm Change?

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: Patrick Doetsch, Pavel Golik, Tobias Menne, Zoltan Tüske, Albert Zeyer, ...
  – language modeling: Martin Sundermeyer, Kazuki Irie, ...
  – cf. hltpr.rwth-aachen.de/web/Publications

• toolkits used for our own 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)
  – RETURNNN: RWTH Extensible Training for Universal Recurrent Neural Networks (new!)
  – ...
  – cf. hltpr.rwth-aachen.de/web/Software
Outline

Human Language Technology: Overview & History

Statistical Approach

Neural Network and Statistical Approach

Deep Learning for Acoustic Modelling

Deep Learning for Language Modelling

Current State-of-the-Art in ASR

References
Outline

Human Language Technology: Overview & History

Statistical Approach

Neural Network and Statistical Approach

Deep Learning for Acoustic Modelling

Deep Learning for Language Modelling

Current State-of-the-Art in ASR

References
Human Language Technology: Overview & History.

Terminology:
- speech: acoustic signal, spoken language
- language: text, sequence of characters, written language
- scientific disciplines:
  - NLP: natural language processing (in the strict sense): written language only
  - HLT: human language technology: spoken and written language

Characteristic task properties:
- well-defined 'classification' tasks:
  - 5000-year history of (written!) language
  - well-defined classes: letters or words of the language
- easy task for humans (at least for natives!)
- hard task for computers (as last 50 years have shown!)

Specific well-defined tasks in HLT:
- Automatic Speech Recognition (ASR)
  - Text image recognition (printed and handwritten text, offline) (HWR)
- Machine Translation (MT)
  - wir wollen diese große Idee erhalten
  - we want to preserve this great idea
Human Language Technology: Overview & History

Speech and Language: Characteristic Properties

Typical situation:

input sequence → output sequence

Tasks:

• speech recognition: speech signal → words/letter sequence
• recognition of image text: text image → words/letter sequence (printed/written characters)
• machine translation: source word/letter sequence → target words/letter sequence

Common property:

output sequence = natural language word/letter sequence

Terminology:

• compound decision theory
• contextual pattern recognition
• structured output

elementary pattern classification and machine learning:

single class index without any structure
Speech recognition

What is the problem?
- ambiguities at all levels
- interdependencies of decisions

Approach [CMU and IBM 1975]:
- hypothesis scores
- probabilistic framework
- statistical decision theory

Modern terminology:
- machine learning
History Speech Recognition 1975-2015

- steady increase of challenges:
  - vocabulary size: 10 digits ... 1000 ... 10,000 ... 500,000 words
  - speaking style: read speech ... colloquial/spontaneous speech

- steady improvement of statistical methods: HMM, Gaussians and mixtures, statistical trigram language model, adaptation methods, discriminative sequence training, artificial neural nets, ...

- 1985-93: criticism about statistical approach
  - too many parameters and saturation effect
  - ... 'will never work for large vocabularies' ...

- remedy(?) by rule-based approach:
  - language models (text): linguistic grammars and structures
  - phoneme models (speech): acoustic-phonetic expert systems
  - limited success for various reasons:
    - huge manual effort is required!
    - problem of coverage and consistency of rules
    - lack of robustness

- evaluations, experimental tests:
  - the same evaluation criterion on the same test data
  - direct comparison of algorithms and systems
Bayes Architecture for Speech Recognition (and other HLT tasks)

Speech Recognition = Modeling + Statistics + Efficient Algorithms
Bayes Architecture for Speech Recognition (and other HLT tasks)

Speech Recognition = Modeling + Statistics + Efficient Algorithms + Performance Measure
Statistical Approach

Ingredients:

- **performance measure** (often edit distance):
  to judge the quality of the system output

- **probabilistic models** (with a suitable structure):
  capture dependencies within/between input observation sequence $X$ and output word sequence $W$
  - elementary observations: Gaussian mixtures, log-linear models, SVMs, NNs, ...
  - sequence context: $n$-gram Markov chains, HMMs, CRFs, RNNs, ...
  - effectively: discrimination function needed

- **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, ...
ASR Architecture

Speech Input

Samples \( s_1 \ldots s_M \)

Feature Extraction

Feature Vectors \( x_1 \ldots x_T \)

Global Search Process:

\[
\text{maximize } \quad p(w_1 \ldots w_N) \cdot p(x_1 \ldots x_T | w_1 \ldots w_N)
\]

over \( w_1 \ldots w_N \)

Recognized Word Sequence \( \{w_4 \ldots w_N\}_{\text{opt}} \)

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

Acoustic Model

Language Model

Statistical Approach to Automatic Speech Recognition (ASR) [Bahl & Jelinek \(^+\) 1983]
Bayes Decision Rule: Sources of Errors

Why does a 'Bayes' decision system make errors?

To be more exact: Why errors in addition to so-called Bayes errors, i.e. the minimum that can be achieved?

Reasons from the viewpoint of Bayes' decision rule:

- probability models:
  - 'incorrect' observation $x$: only incomplete part or poor transformation of true observations used
  - incorrect models, e.g. $p_\theta(c|x)$ or $p_\theta(c_{1}^{N}|x_{1}^{T})$

- training conditions:
  - poor training criterion
  - not enough training data
  - mismatch conditions between training and test data

- training criterion + efficient algorithm:
  - suboptimal algorithm for training (e.g. gradient descent)

- decision rule:
  - incorrect error measure, e.g. MAP rule in ASR and MT

- decision rule + efficient algorithm:
  - suboptimal search procedure, e.g. beam search or N-best lists
Human Language Technology: Overview & History

ASR Architecture: Neural Networks

neural feature transformation:
- tandem [Hermansky & Ellis\textsuperscript{+} 2000]
- bottleneck [Grézl & Karafiát\textsuperscript{+} 2007]

earlier introduced as non-linear LDA [Fontaine & Ris\textsuperscript{+} 1997]
ASR Architecture: Neural Networks

neural acoustic modeling:
- connectionist temporal classification (CTC) [Graves & Fernández 2006]
Human Language Technology: Overview & History.

ASR Architecture: Neural Networks

Speech Input

Samples 

\[ s_1 \ldots s_M \]

Global Search Process:

\[
\text{maximize} \quad p(w_1 \ldots w_N) \cdot p(s_1 \ldots s_M \mid w_1 \ldots w_N) \\
\text{over} \quad w_1 \ldots w_N
\]

Recognized Word Sequence

\[ \{w_1 \ldots w_N\}_{\text{opt}} \]

integrated learning of acoustic model and feature extraction

- single channel [Palaz & Collobert\textsuperscript{+} 2013]
  [Tüske & Golik\textsuperscript{+} 2014]
  [Golik & Tüske\textsuperscript{+} 2015]
- multichannel [Sainath & Weiss\textsuperscript{+} 2015]
ASR Architecture: Neural Networks

neural language modeling:
- feed-forward (FF) [Schwenk 2007]
- recurrent [Mikolov & Karafiát+ 2010]
- LSTM [Sundermeyer & Schlüter+ 2012]
- long-context FF [Tüske & Irie+ 2016]
ASR Architecture: Neural Networks

Speech Input

Samples $s_1 ... s_M$

Feature Extraction

Feature Vectors $x_1 ... x_T$

Global Search Process:

maximize $g(x_1 ... x_T, w_1 ... w_N)$

over $w_1 ... w_N$

Recognized Word Sequence $\{w_1 ... w_N\}_{opt}$

integrated NN approach:

- attention, encoder/decoder approach [Bahdanau & Chorowski+ 2015] [Chan & Jaitly+ 2015]
- segmental/inverted HMM [Lu & Kong+ 2016] [Doetsch & Hegselmann+ 2016]
Outline

Human Language Technology: Overview & History

Statistical Approach
  Principles
  Acoustic Features
  Acoustic Modeling
  Language Modeling
  Search

Neural Network and Statistical Approach

Deep Learning for Acoustic Modelling

Deep Learning for Language Modelling

Current State-of-the-Art in ASR

References
Outline

Human Language Technology: Overview & History

Statistical Approach
  Principles
  Acoustic Features
  Acoustic Modeling
  Language Modeling
  Search

Neural Network and Statistical Approach

Deep Learning for Acoustic Modelling

Deep Learning for Language Modelling

Current State-of-the-Art in ASR

References
Starting Points

- very complex problem: no perfect knowledge of the dependencies in speech and language:
  - different from conventional computer science
  - like a problem in natural sciences (cf. approximative modeling in physics)
- perfect solution will be difficult:
  - we accept that the system will make errors
  - but we try to find the best compromise
- fairly general view:
  - input sequence (ASR: sequence over time \( t: X := x_1...x_t...x_T \))
  - output sequence: \( W := w_1...w_n...w_N \) of unknown length \( N \)
- we need a generation mechanism:
  \[ X \rightarrow W = \hat{W}(X) \]
- to this purpose, we assume a
  - posterior distribution \( pr(W|X) \)
  - which can be extremely complex: both arguments are sequences!
Bayes Decision Rule for Sequences

- performance measure or cost function $L[\tilde{W}, W]$ (e.g. edit distance) between true output sequence $\tilde{W}$ and hypothesized output sequence $W$.
- Bayes decision rule minimizes expected cost:
  $$X \to \overline{W}(X) := \arg \min_W \left\{ \sum_{\tilde{W}} pr(\tilde{W}|X) \cdot L[\tilde{W}, W] \right\}$$

- standard decision rule uses sequence-level cost (MAP rule):
  $$X \to \hat{W}(X) := \arg \max_W \left\{ pr(W|X) \right\}$$
  since [Bahl & Jelinek, 1983], this simplified Bayes decision rule is widely used for speech recognition, handwriting recognition, machine translation, ...
  well-known inconsistency! [Jelinek, 1997, pp. 4-5]
- however, standard decision rule works well, as often both decision rules agree, which can be proven under certain conditions [Schlüter & Nussbaum, 2012], e.g.:
  $$L[W, \tilde{W}] \text{ is a metric, and } \max_W pr(W|X) \geq 0.5 \implies \overline{W}(X) = \hat{W}(X)$$
- approximative (second pass) sequence-level cost approaches provide good improvements [Stolcke & König, 1997, Mangu & Brill, 1999, Goel & Byrne, 2000, Wessel & Schlüter, 2001]
Statistical Approach
Principles

Generative vs. Discriminative Approach

Bayes Decision Rule:

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

practical considerations:

• unknown distribution \( pr(W|X) \):
  remedy: replace true \( pr(W|X) \) by a model \( p(W|X) \)
  and learn its free parameters from a HUGE set of examples

• important problem:
  – compositional modelling for \( p(W|X) \) is needed since \( W \) and \( X \) are sequences
  – units smaller than the whole sequence are needed (e.g. phrases/word groups, words, letters)

• two principal approaches:
  – generative approach: \( p(W, X) = p(W) \cdot p(X|W) \)
    language model \( p(W) \), trained on text data
    acoustic model \( p(X|W) \), trained on (transcribed) audio data
  – discriminative (or direct) approach: \( p(W|X) = p(W, X)/\sum_{\tilde{W}} p(\tilde{W}, X) \)
Statistical Approach
Principles

Generative vs. Discriminative Training

Starting point:
- models $p_{\theta}(W)$ and $p_{\theta}(X|W)$ with unknown parameters $\theta$
- training data: set of (audio, sentence) pairs $(X_r, W_r)$, $r = 1, ..., R$

Training:
- generative model: maximum likelihood (along with EM/Viterbi algorithm):
  $$F(\theta) = \sum_r \log p_{\theta}(W_r, X_r) = \sum_r \log p_{\theta}(W_r) + \sum_r \log p_{\theta}(X_r|W_r)$$
  nice property: decomposition into two separate problems (also: separate training data):
  - language model $p_{\theta}(W)$: without annotation!
  - acoustic model $p_{\theta}(X|W)$: with annotation!
- discriminative model: discriminative training
  - optimizes decision boundaries, e.g. maximum mutual information (MMI)
  - ideally: optim. error rate, e.g. minimum classification error (MCE), minimum phone error (MPE)
  - in practice:
    - initialization by maximum likelihood
    - complex optimization problem: sum over all sentences in denominator
    - approximation: word lattice, many shortcuts, ...
  - experiments: relative improvement by 5-10% over maximum likelihood
Statistical Approach
Acoustic Features

Outline

Human Language Technology: Overview & History

Statistical Approach
  Principles
  Acoustic Features
    Acoustic Modeling
    Language Modeling
    Search

Neural Network and Statistical Approach

Deep Learning for Acoustic Modelling

Deep Learning for Language Modelling

Current State-of-the-Art in ASR

References
**Statistical Approach**

**Acoustic Features**

**Alternative Acoustic Feature Streams**

**SPEECH SIGNAL**

- **PREEMPHASIS AND WINDOWING**
  - **MAGNITUDE SPECTRUM**
  - **MEL FREQUENCY WARPING**
    \[ f_{\text{mel}} = 2595 \lg (1 + \frac{f}{700 \text{ Hz}}) \]
  - **CRITICAL BAND INTEGRATION**
- **LOGARITHM**
- **CEPSTRAL DECORRELATION**
- **CEPSTRAL MEAN NORM.**
- **ENERGY NORM.**
- **SPECTRAL DYNAMIC FEATURES**

**ACOUSTIC VECTOR**

- **MFCC**

**SPEECH SIGNAL**

- **PREEMPHASIS AND WINDOWING**
  - **GAMMATONE FILTERBANK**
    \[ h(t) = k \cdot t^{n-1} \exp(-2\pi \cdot B \cdot t) \cdot \cos(2\pi \cdot f_c \cdot t + \phi) \]
  - **RECTIFYING**
  - **TEMPORAL INTEGRATION**
  - **SPECTRAL INTEGRATION**
  - **10th ROOT**
  - **CEPSTRAL DECORRELATION**
  - **CEPSTRAL MEAN NORM.**
  - **ENERGY NORM.**
  - **SPECTRAL DYNAMIC FEATURES**

**ACOUSTIC VECTOR**

- **GT**

**SPEECH SIGNAL**

- **POWER SPECTRUM**
  - **BARK FREQUENCY WARPING**
    \[ f_{\text{bark}} = 6 \ln(f/600 + [(f/600)^2 + 1]^{0.5}) \]
  - **TRAPEZOID CRITICAL BAND INTEGRATION**
  - **EQUAL LOUDNESS PREEMP.**
  - **INTENSITY-LOUDNESS POWER LAW**
  - **AUTOREGRESSIVE MODELING**
  - **LPC TO CEPSTRAL COEFF.**
  - **CEPSTRAL MEAN NORM.**
  - **ENERGY NORM.**
  - **SPECTRAL DYNAMIC FEATURES**

**ACOUSTIC VECTOR**

- **PLP**

**SPEECH SIGNAL**

- **POWER SPECTRUM**
  - **MEL FREQUENCY WARPING**
    \[ f_{\text{mel}} = 2595 \lg (1 + \frac{f}{700 \text{ Hz}}) \]
  - **CRITICAL BAND INTEGRATION**
- **EQUAL LOUDNESS PREEMP.**
- **INTENSITY-LOUDNESS POWER LAW**
- **AUTOREGRESSIVE MODELING**
- **LPC TO CEPSTRAL COEFF.**
- **CEPSTRAL MEAN NORM.**
- **ENERGY NORM.**
- **SPECTRAL DYNAMIC FEATURES**

**ACOUSTIC VECTOR**

- **MF-PLP**

---

Hierarchical MRASTA Filtering

• Long-term features:
  – Representations relAtive SpecTrA (RASTA) filtering [Hermansky & Fousek 2005].
  – Modulation frequency range (≈1-20Hz) relevant for speech perception.
• Multi-resolutioonal smoothing of temporal trajectories of critical band energies (CRBE)
• Filtering with first and second derivatives of Gaussians, $g_1, g_2$
  – $\sigma$ varying in the range 8-60 ms
  – E.g. 12 temporal filters applied on 20 CRBEs + derivatives in freq.
• Processing fast and slow modulation spectrum by hierarchical MLPs

Remarks:
• FF MLPs: currently best results using MRASTA
• LSTM RNNs: filter banks sufficient, though
Outline

Human Language Technology: Overview & History

Statistical Approach
- Principles
- Acoustic Features
  Acoustic Modeling
- Language Modeling
- Search

Neural Network and Statistical Approach

Deep Learning for Acoustic Modelling

Deep Learning for Language Modelling

Current State-of-the-Art in ASR

References
**Statistical Approach**

**Acoustic Modeling**

**Speaking Rate Variation**

- fundamental problem in ASR: variation in speaking rate, necessitates non-linear time alignment
- stochastic finite state machine:
  - linear chain of states $s = 1, ..., S$
  - transitions: forward, loop and skip
- trellis:
  - unfold over time $t = 1, ..., T$
  - path: state sequence $s_1^T = s_1s_ts_T$
  - observations: $x_1^T = x_1x_tx_T$

**general view:**

- two sequences without synchronization: acoustic vectors and states (with labels)
- mechanism that takes care of the synchronization (=alignment) problem
Hidden Markov Models (HMM)

The acoustic model $p(X|W)$ provides the link between sentence hypothesis $W$ and observations sequence $X = x_1^T = x_1...x_t...x_T$:

- 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)

- phonetic labels (allophones, sub-phones): $(s, W) \rightarrow \alpha = \alpha_{sW}$
  \[
  p(x_t|s, W) = p(x_t|\alpha_{sW})
  \]

- typical approach: models for phonemes with left and right phonetic context (triphones):
  decision tree (CART) clustering for finding equivalence classes

- temporal context: augment feature vector with context window around position $t$

- exploit first-order HMM structure for efficient search and training
Baseline HMM training:
- maximum likelihood by EM (expectation/maximization) algorithm
- looks like the ultimate and perfect solution

Positive properties:
- FULL generative model: \( p_\theta(W, X) = p_\theta(W) \cdot p_\theta(X|W) \) along with HMM for \( p_\theta(X|W) \): describes the problem completely
- natural training criterion:
  - maximum likelihood, i.e. \( \max_\theta \left\{ \sum_r \log p_\theta(W_r, X_r) \right\} \)
  - virtually closed form solutions by EM algorithm
  - nice from the mathematical point of view

Negative properties:
- EM or maximum likelihood criterion
  - solves a problem that is more complex than required, i.e. \( p_\theta(W, X) \) vs. \( p_\theta(W|X) \)
  - VERY hard from the estimation (learning) point of view
- well-known in classical pattern recognition, but ignored/overlooked in ASR:
  - density estimation, i.e. learning \( p_\theta(X|W) \) or \( p_\theta(x_t|\alpha) \), is much harder than classification, i.e. learning \( p_\theta(W|X) \) or \( p_\theta(\alpha|x_t) \)
Outline

Human Language Technology: Overview & History

Statistical Approach
  Principles
  Acoustic Features
  Acoustic Modeling
Language Modeling
Search

Neural Network and Statistical Approach

Deep Learning for Acoustic Modelling

Deep Learning for Language Modelling

Current State-of-the-Art in ASR

References
Statistical Approach
Language Modeling

Statistical Modeling of Syntax and Semantics

Definition of a language model (LM):
- \( p(w_1^N) \) : (prior) probability of the word sequence \( w_1^N := w_1...w_n...w_N \)

Need for language model in Bayes decision rule in ASR (also SMT!):
\[
x_1^T \rightarrow \hat{w}_1^N(x_1^T) = \arg\max_{N,w_1^N} \{ p(w_1^N) \cdot p(x_1^T|w_1^N) \}
\]

Observations about the language model \( p(w_1^N) \):
- it can be learned from text only (unlabeled data!)
- it can improve performance dramatically

Perplexity:
- quality measure for LM (based on text data, i.e. w/o a recognition experiment)
- geometric average of probability per word by computing \( N \)-th root:
\[
PP := \left( p(w_1^N) \right)^{-1/N} = \left( \prod_{n=1}^{N} p(w_n|w_1^{n-1}) \right)^{-1/N}
\]
- geometric average of inverse probability \( \rightarrow \) interpretation: average effective vocabulary size
Markov Chain, Count Models

Conventional approach:

- assume Markov chain of order $k$:
  limit the dependence on the full history $w_0^{n-1}$ to the immediate $k$ predecessor words:

\[ p(w_n|w_0^{n-1}) := p_\theta(w_n|w_{n-k}^{n-1}) \]

terminology: $(k+1)$-gram, e.g. four-, tri-, bi-, unigram ($w_{n-1}^{n-1}$ defines empty context for unigram)

- free parameters $\theta$ to be learned from training data:
  conditional probabilities $p_\theta(w_n|w_{n-k}^{n-1})$ for the $(k+1)$-gram events

- natural training criterion for a corpus $w_1^N$: minimum perplexity

\[
\max_{\theta} \left\{ \frac{1}{N} \sum_{n=1}^{N} \log p_\theta(w_n|w_{n-k}^{n-1}) \right\} \xrightarrow{N \to \infty} \max_{\theta} \left\{ \sum_{w,h_1^k} pr(w|h_1^k) \cdot \log p_\theta(w|h_1^k) \right\}
\]

- equivalent to cross-entropy training (or maximum likelihood)
- resulting estimates: relative frequencies based on event counts
Unseen Events, Smoothing

Problem:
- most of the events are never seen in training data
- example: vocabulary of $100k = 10^5$ words results in $10^{15}$ possible trigrams
- result: virtually all event counts are zero

Remedy:
- interpolation/combination of LMs of various orders $k$, e.g. fivegrams, fourgram, trigram, bigram and unigram events
- various strategies:
  - models: interpolation or back-off
  - estimation: cross-validation or leave-one-out
  - concept of generalized marginal distributions, e.g. going from trigrams to bigrams
- most strategies implemented in LM toolkit by SRI
Outline

Human Language Technology: Overview & History

Statistical Approach
  Principles
  Acoustic Features
  Acoustic Modeling
  Language Modeling

Search

Neural Network and Statistical Approach

Deep Learning for Acoustic Modelling

Deep Learning for Language Modelling

Current State-of-the-Art in ASR

References
Statistical Approach

Search

Search Space

Combinatorial complexity
- *Bayes* decision rule involves optimization over all possible word sequences and alignments
- Number of word sequences and number of alignment paths rise exponential with length

Dynamic programming
- Markov assumptions in HMM and LM can be exploited for efficient search
- Recursion equations reduce complexity to being linear in input length and polynomial in vocabulary size
- For limited vocabularies and LM context **exact** solution of optimization problem possible.
**Statistical Approach**

**Search**

**Beam Search**

Large vocabulary
- even for moderate LM context, for large vocabularies ($\gtrsim 10k$), exhaustive search becomes prohibitive
- **approximations** are needed for efficient search
- utilize probabilistic scoring for hypothesis pruning

Dynamic programming hypothesis pruning
- time-synchronous propagation of partial dynamic programming hypotheses
- discard hypotheses relative to current best hypotheses
- goal: complexity overall linear in input

**Interrelation with Modeling**
- more sophisticated models usually introduce higher complexity into system
- **however**: scores become more pronounced
- allows for tighter pruning, compensates increase in complexity
Outline

Human Language Technology: Overview & History

Statistical Approach

Neural Network and Statistical Approach

Basics
Training and Probabilistic Interpretation
Softmax Revisited: Relation to Generative Modeling
Recurrent Neural Networks for Sequence Processing

Deep Learning for Acoustic Modelling

Deep Learning for Language Modelling

Current State-of-the-Art in ASR

References
Neural Network and Statistical Approach

Outline

Human Language Technology: Overview & History

Statistical Approach

Neural Network and Statistical Approach

Basics

Training and Probabilistic Interpretation

Softmax Revisited: Relation to Generative Modeling

Recurrent Neural Networks for Sequence Processing

Deep Learning for Acoustic Modelling

Deep Learning for Language Modelling

Current State-of-the-Art in ASR

References
Neural Network and Statistical Approach

Basics

(First) NN Renaissance around 1986

Various interpretations/justifications:

• human/biological brain
• massive parallelism
• mathematical viewpoint:
  modelling ANY input-output relation

Typical ANN structure:

• MLP: feedforward multi-layer perceptron
• with input, hidden and output layers

Theoretical results:

• one hidden layer should be sufficient (!?)
  [Cybenko 1989, Hornik & Stinchcombe+ 1989]

Training:

• (hard) optimization problem with millions of free parameters (= weights)
Classical Architecture:

Feedforward Multi-Layer Perceptron (FF-MLP)

- task: classification with observation vector \( x \in \mathbb{R}^D \) and associated class \( c \)

Architecture:

- several layers (feedforward links only, no recurrence)
- input layer = observation vector \( x \):
  - each node represents a vector component
- between layers:
  - matrix-vector product for layer pair
  - nonlinear activation function
- output layer:
  - softmax normalization
  - each output node represents a class \( c \) and its associated score \( p_\theta(c, x) \)
- set \( \theta \) of all weights (parameters) of the FF-MLP
ANN Activation Functions

Examples of activation functions:

- sigmoid function (also called logistic function):
  \[ u \rightarrow \sigma(u) = \frac{1}{1 + \exp(-u)} \in [0, 1] \]

- hyperbolic tangent:
  \[ u \rightarrow \tanh(u) = 2 \sigma(2u) - 1 \in [-1, 1] \]
  – in principle: no difference to sigmoid \( \sigma(\cdot) \)
  – in practice: difference due to side effects

- rectifying linear unit:
  \[ u \rightarrow r(u) = \max\{0, u\} \]
  – so far: not useful in symbolic processing (?

- softmax function:
  \[ u_c \rightarrow S(u_c) = \frac{\exp(u_c)}{\sum_{\tilde{c}} \exp(u_{\tilde{c}})} \quad \text{with} \quad \sum_{c} S(u_c) = 1.0 \]
  – generates normalized output for (probability distribution over) each node \( c \) of the layer under consideration (typically: output layer)
Outline

Human Language Technology: Overview & History

Statistical Approach

Neural Network and Statistical Approach
  Basics
  Training and Probabilistic Interpretation
  Softmax Revisited: Relation to Generative Modeling
  Recurrent Neural Networks for Sequence Processing

Deep Learning for Acoustic Modelling

Deep Learning for Language Modelling

Current State-of-the-Art in ASR

References
Classification with Artificial Neural Networks

Decision rule for observation (vector) $x$:

$$x \rightarrow \hat{c}_x := \arg\max_c \left\{ p_\theta(c, x) \right\}$$

Ideal values at output nodes:
- correct class: 1
- wrong class: 0

Distinguish varying conditions for decision rule:
- no context, in isolation (here)
- context of a sequence (see later)

Training criteria:
- squared error: unconstrained output: $p_\theta(c, x) \in \mathbb{R}$

$$F_{SE}(\theta) := \frac{1}{N} \sum_{n=1}^{N} \sum_c \left[ p_\theta(c, x_n) - \delta(c, c_n) \right]^2$$
- cross-entropy: normalized output: $p_\theta(c, x) \in [0, 1]: \sum_c p_\theta(c, x) = 1$

$$F_{CE}(\theta) := \frac{1}{N} \sum_{n=1}^{N} \log p_\theta(c_n|x_n)$$
Training Criteria: Interpretation & Relation to Error Rate

Straightforward analysis shows important result for both training criteria:

- ANN outputs are (estimates of) true class posterior probabilities!
- result independent of any training strategy (e.g. type of backpropagation)
- assumes sufficient flexibility and parameters in ANN
- generalization capability from training to test set: not addressed

Gradient search (backpropagation):

- we can only find a local optimum
- there may be a huge number of local optima; but most of them seem to be equivalent
- experimental evidence: backpropagation able to find local optimum that’s typically ’good enough’
- generalization capability: implicitly taken into account by cross-validation (early stopping) ?

Relation between error rate and training criteria?

- we need a strict distinction:
  - error rate for the true distribution: Bayes classification error
  - error rate for the learned distribution: model classification error
- training criteria: tight upper bound for squared difference between these two error rates [Ney 2003]
- remark: this result does not address the generalization problem
Outline

Human Language Technology: Overview & History

Statistical Approach

Neural Network and Statistical Approach
  Basics
  Training and Probabilistic Interpretation
  Softmax Revisited: Relation to Generative Modeling
  Recurrent Neural Networks for Sequence Processing

Deep Learning for Acoustic Modelling

Deep Learning for Language Modelling

Current State-of-the-Art in ASR

References
Neural Network and Statistical Approach

Softmax Revisited: Relation to Generative Modeling

Conventional view: consider MLP with softmax output
- input layer: raw input vector $z$
- hidden layers perform feature extraction:
  $$x = f(z)$$
  with feature vector $x \in \mathbb{R}^D$ before output layer
  note: no dependence on class labels $c = 1, ..., C$
- output layer: probability distribution over classes $c$
  $$p(c|x) = \frac{\exp(\lambda_c^T \cdot x + \gamma_c)}{\sum_{c'} \exp(\lambda_{c'}^T \cdot x + \gamma_{c'})}$$
  with output layer weights $\lambda_c \in \mathbb{R}^D$ and offsets (biases) $\gamma_c \in \mathbb{R}$

Interpretation of MLP with softmax output:
- feature extraction followed by a log-linear classifier

Relation to generative modeling [Heigold & Schlüter 2012]:
- softmax operation results from using class posterior distribution of a Gaussian model
Outline

Human Language Technology: Overview & History

Statistical Approach

Neural Network and Statistical Approach
  Basics
  Training and Probabilistic Interpretation
  Softmax Revisited: Relation to Generative Modeling
  Recurrent Neural Networks for Sequence Processing

Deep Learning for Acoustic Modelling

Deep Learning for Language Modelling

Current State-of-the-Art in ASR

References
Sequence Processing

So far:
- handling of (input, output) pairs \((c, x)\) in isolation
- no internal structure in \(c\) or \(x\) (unlike sequences)

From single events to sequences:
- consider a pair of synchronized input and output sequence over time \(t\):
\[
(c_t, x_t), \ t = 1, \ldots, T
\]
with input vectors \(x_t\) and class labels \(c_t\)
- goal: model the conditional probability \(p(c_{1:T}^T | x_{1:T}^T)\) of the sequence \(c_{1:T}^T\)
  (assuming causality and a special start symbol \(c_0\)):
\[
p(c_{1:T}^T | x_{1:T}^T) = \prod_t p(c_t | \ldots)
\]
with ANN output vector \(y_t = p(c_t | \ldots)\) at each time \(t\)
Sequences with Synchronisation

Illustration:
- model with 1:1 correspondence between class labels $c_t^T$ and observations $x_t^T$
- sequence length $T$ is known

| Observations $x_t^T$: $x_1$ $x_2$ ... $x_{t-1}$ $x_t$ $x_{t+1}$ ... $x_{T-1}$ $x_T$ |
|----------------------------------|----------------------------------|
| Class labels $c_t^T$: $c_1$ $c_2$ ... $c_{t-1}$ $c_t$ $c_{t+1}$ ... $c_{T-1}$ $c_T$ |

typical problems:
- spelling correction (character level)
- POS tagging (POS: parts of speech)
- frame labelling in ASR (incl. pronunciation and language models!)
  and acoustic scores in hybrid HMMs
- recognition problems with no problems of boundary detection:
  isolated words, printed character recognition, ...
Factorization of Conditional Probability $p(c_1^T | x_1^T)$

- conditional independence in $c_1^T$ with look-ahead for $x_1^T$: $p(c_1^T | x_1^T) = \prod_{t=1}^{T} p_t(c_t | x_1^T)$

<table>
<thead>
<tr>
<th>observations $x_1^T$:</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>...</th>
<th>$x_{t-1}$</th>
<th>$x_t$</th>
<th>$x_{t+1}$</th>
<th>...</th>
<th>$x_{T-1}$</th>
<th>$x_T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>class labels $c_1^T$:</td>
<td>$___$</td>
<td>$___$</td>
<td>$___$</td>
<td>$___$</td>
<td>$___$</td>
<td>$___$</td>
<td>$___$</td>
<td>$___$</td>
<td>$___$</td>
</tr>
</tbody>
</table>

- conditional dependence in $c_1^T$ without look-ahead in $x_1^T$: $p(c_1^T | x_1^T) = \prod_{t=1}^{T} p(c_t | c_{t-1}^0, x_1^t)$

<table>
<thead>
<tr>
<th>observations $x_1^T$:</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>...</th>
<th>$x_{t-1}$</th>
<th>$x_t$</th>
<th>$___$</th>
<th>$___$</th>
<th>$___$</th>
<th>$___$</th>
</tr>
</thead>
<tbody>
<tr>
<td>class labels $c_1^T$:</td>
<td>$c_1$</td>
<td>$c_2$</td>
<td>...</td>
<td>$c_{t-1}$</td>
<td>$___$</td>
<td>$___$</td>
<td>$___$</td>
<td>$___$</td>
<td>$___$</td>
</tr>
</tbody>
</table>

- conditional dependence in $c_1^T$ with look-ahead in $x_1^T$: $p(c_1^T | x_1^T) = \prod_{t=1}^{T} p(c_t | c_{t-1}^0, x_1^T)$

<table>
<thead>
<tr>
<th>observations $x_1^T$:</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>...</th>
<th>$x_{t-1}$</th>
<th>$x_t$</th>
<th>$x_{t+1}$</th>
<th>...</th>
<th>$x_{T-1}$</th>
<th>$x_T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>class labels $c_1^T$:</td>
<td>$c_1$</td>
<td>$c_2$</td>
<td>...</td>
<td>$c_{t-1}$</td>
<td>$___$</td>
<td>$___$</td>
<td>$___$</td>
<td>$___$</td>
<td>$___$</td>
</tr>
</tbody>
</table>
Recurrent Neural Network (RNN): 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:
  $$s_t = o_t \odot c_t$$
**RNN and probabilities: What does a general RNN compute?**

note: general RNN includes LSTM as a special case

two sequences over time $t = 1, \ldots, T$:

input: sequence of observations: $x_T^1 = x_1 \ldots x_t \ldots x_T$

output: sequence of class labels: $c_T^1 = c_1 \ldots c_t \ldots c_T$

consider the posterior probability of the output sequence:

factorization over time $t$: $p(c_T^1 | x_T^1) = \prod_{t=1}^{T} p(c_t | c_{t-1}, x_T^1)$

marginalization for time $t$: $\sum_{c_T^1: c_t = c} p(c_T^1 | x_T^1) = p_t(c | x_T^1)$

more ...

notation for RNN output vector with nodes = classes $c = 1, \ldots, C$:

$y_t = [y_t(c)] = [p_t(c | \ldots)]$
RNN: Variant 1

uni-directional, no feedback of output labels

\[
\begin{align*}
&\cdots \quad y_{t-1} \quad y_t \quad \cdots \\
&\cdots \quad h_{t-1} \quad h_t \quad \cdots \\
&\cdots \quad x_{t-1} \quad x_t \quad \cdots 
\end{align*}
\]

RNN output vector:

\[
y_t(c) = p_t(c | x_1^t)
\]
RNN: Variant 2

uni-directional, with feedback of output labels

\[
\begin{align*}
&\cdots \\
&\cdots \\
&\cdots \\
&x_{t-1} \\
&h_{t-1} \\
y_{t-1} \\
&\cdots \\
&\cdots \\
&\cdots \\
&x_t \\
&h_t \\
y_t \\
&\cdots \\
&\cdots \\
&\cdots \\
&y_{t-1} \\
&y_t \\
&\cdots \\
\end{align*}
\]

RNN output vector:

\[
y_t(c) = p_t(c | c_0^{t-1}, x_1^t)
\]
RNN: Variant 3

bi-directional, no feedback of output label

Internal Structure: Separate Forward and Backward Hidden Layers
RNN: Variant 3

bi-directional, no feedback of output label

\[
\begin{align*}
\cdots & \quad \cdots & \quad y_{t-1} & \quad y_t & \quad y_{t+1} & \quad \cdots \\
\cdots & \quad \cdots & \quad h_{t-1} & \quad h_t & \quad h_{t+1} & \quad \cdots \\
\cdots & \quad \cdots & \quad x_{t-1} & \quad x_t & \quad x_{t+1} & \quad \cdots \\
\end{align*}
\]

RNN output vector:

\[
y_t(c) = p_t(c|x_1^T)
\]
RNN: Variant 4

bi-directional, with uni-directional feedback of output label

RNN output vector:

\[ y_t(c) = p_t(c | c_{t-1}, x_1^T) \]
RNN: Variant 5

bi-directional, with bi-directional feedback of output label

\( \ldots \) \hspace{1cm} \ldots \hspace{1cm} y_{t-1} \hspace{1cm} y_t \hspace{1cm} y_{t+1} \hspace{1cm} \ldots \)

\( \ldots \) \hspace{1cm} \ldots \hspace{1cm} h_{t-1} \hspace{1cm} h_t \hspace{1cm} h_{t+1} \hspace{1cm} \ldots \)

\( \ldots \) \hspace{1cm} \ldots \hspace{1cm} x_{t-1} \hspace{1cm} x_t \hspace{1cm} x_{t+1} \hspace{1cm} \ldots \)

RNN output vector:

\[ y_t(c) = p_t(c | c_0^{t-1}, c_{t+1}^T, x_1^T) \]
Overview of RNN Outputs

<table>
<thead>
<tr>
<th>label feedback</th>
<th>no</th>
<th>uni-direct.</th>
<th>bi-direct.</th>
</tr>
</thead>
<tbody>
<tr>
<td>uni-dir. RNN</td>
<td>$p_t(c</td>
<td>x_t^t)$</td>
<td>$p_t(c</td>
</tr>
<tr>
<td>bi-dir. RNN</td>
<td>$p_t(c</td>
<td>x_T^T)$</td>
<td>$p_t(c</td>
</tr>
</tbody>
</table>

- experiments: typically $p_t(c|x_T^T)$
- exploitation of recurrence within each layer
Deep Learning for Acoustic Modelling

Outline

Human Language Technology: Overview & History
Statistical Approach
Neural Network and Statistical Approach

Deep Learning for Acoustic Modelling
  Approach & History
  Training
  Empirical Overview of Current Methods

Deep Learning for Language Modelling

Current State-of-the-Art in ASR

References
Deep Learning for Acoustic Modelling
Approach & History

Outline

Human Language Technology: Overview & History

Statistical Approach

Neural Network and Statistical Approach

Deep Learning for Acoustic Modelling
Approach & History
Training
Empirical Overview of Current Methods

Deep Learning for Language Modelling

Current State-of-the-Art in ASR

References
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 \( \alpha \) 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 \( \alpha = 1, \ldots, 5000 \) vs. vectors \( x_t \in \mathbb{R}^{D=40} \)
  – well-known result in pattern recognition (but ignored in ASR!)
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
Deep Learning for Acoustic Modelling

Approach & History

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 & Sharma 1998]: TRAPS: learning temporal patterns of spectral energies
- [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 & Liwicki\(^+\) 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%
What is Different Now after 25 Years? - A (Simplified) Summary

Comparison of today’s systems vs. 1989-1994:
• number of hidden layers: 10 (or more) rather than 2-3
• number of output nodes: 5000 (or more) rather than 50
• optimization strategy:
  practical experience and heuristics,
  e.g. layer-by-layer pretraining
• computation power: much more

Terminology (for feedforward and recurrent nets):
• deep neural network
• deep learning
Outline

Human Language Technology: Overview & History

Statistical Approach

Neural Network and Statistical Approach

Deep Learning for Acoustic Modelling
  Approach & History
  Training
  Empirical Overview of Current Methods

Deep Learning for Language Modelling

Current State-of-the-Art in ASR

References
Training Strategies

Frame level: cross-entropy $\log p_\theta(\alpha_{st}, W | x_t)$
- required: single best path for each training sentence
- re-alignments during backprop learning: yes ... occasionally ... no
→ simple implementation due to decoupling of best path and backprop

Sentence level: *discriminative sequence training*:
- includes language model $p(W)$
- requires sentence level posterior probability $p(W | x_1^T)$
- improvement: use exponents for language model, transition probabilities and acoustic model
- approximations: single best path, lattice with/without re-computation, ...
- three types of discriminative criteria:
  - logarithm of posterior probability
  - MPE applied to phones: 1 out of 50
  - MPE applied to CART labels: 1 out of 5000
→ complex implementation
Deep Learning for Acoustic Modelling
Empirical Overview of Current Methods

Outline

Human Language Technology: Overview & History

Statistical Approach

Neural Network and Statistical Approach

Deep Learning for Acoustic Modelling
Approach & History
Training
Empirical Overview of Current Methods

Deep Learning for Language Modelling

Current State-of-the-Art in ASR

References
Experimental Setup

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

<table>
<thead>
<tr>
<th>Training Criterion</th>
<th>WER [%]</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>50h</td>
<td>250h</td>
<td></td>
</tr>
<tr>
<td></td>
<td>dev</td>
<td>eval</td>
<td>dev</td>
</tr>
<tr>
<td>Maximum likelihood</td>
<td>24.4</td>
<td>31.6</td>
<td>22.1</td>
</tr>
<tr>
<td>MMI at frame level</td>
<td>23.9</td>
<td>30.9</td>
<td>22.1</td>
</tr>
<tr>
<td>MMI at sentence level</td>
<td>24.1</td>
<td>31.2</td>
<td>21.7</td>
</tr>
<tr>
<td>Minimum phone error</td>
<td>23.6</td>
<td>30.2</td>
<td>20.4</td>
</tr>
</tbody>
</table>

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:
    \[
    493 \cdot 2000 + 5 \cdot 2000^2 + 2000 \cdot 4501 = 30M
    \]
- improvement over best GMM: 20% relative

<table>
<thead>
<tr>
<th>hidden layers</th>
<th>WER [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>dev</td>
</tr>
<tr>
<td>1</td>
<td>24.5</td>
</tr>
<tr>
<td>2</td>
<td>22.0</td>
</tr>
<tr>
<td>3</td>
<td>20.5</td>
</tr>
<tr>
<td>4</td>
<td>19.8</td>
</tr>
<tr>
<td>5</td>
<td>20.1</td>
</tr>
<tr>
<td>6</td>
<td>19.6</td>
</tr>
<tr>
<td>7</td>
<td>19.7</td>
</tr>
<tr>
<td>8</td>
<td>19.6</td>
</tr>
<tr>
<td>9</td>
<td>19.3</td>
</tr>
<tr>
<td>best GMM</td>
<td>23.6</td>
</tr>
</tbody>
</table>
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)

<table>
<thead>
<tr>
<th>Model</th>
<th>Criterion</th>
<th>WER [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>50h</td>
</tr>
<tr>
<td></td>
<td></td>
<td>dev</td>
</tr>
<tr>
<td>MLP</td>
<td>frame MMI</td>
<td>19.6</td>
</tr>
<tr>
<td></td>
<td>MPE</td>
<td>17.5</td>
</tr>
<tr>
<td>best GMM</td>
<td>MPE</td>
<td>23.6</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>activation function</th>
<th>WER [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>50h</td>
</tr>
<tr>
<td></td>
<td>dev</td>
</tr>
<tr>
<td>sigmoid</td>
<td>19.6</td>
</tr>
<tr>
<td>RLU</td>
<td>17.7</td>
</tr>
<tr>
<td>best GMM</td>
<td>23.6</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>LSTM layers</th>
<th>#params</th>
<th>time / epoch</th>
<th>WER [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>dev</td>
<td>eval</td>
</tr>
<tr>
<td>1</td>
<td>6.7M</td>
<td>0:28h</td>
<td>17.6</td>
</tr>
<tr>
<td>2</td>
<td>12.7M</td>
<td>1:00h</td>
<td>14.6</td>
</tr>
<tr>
<td>3</td>
<td>18.7M</td>
<td>1:11h</td>
<td>14.0</td>
</tr>
<tr>
<td>4</td>
<td>24.7M</td>
<td>1:33h</td>
<td>13.5</td>
</tr>
<tr>
<td>5</td>
<td>30.7M</td>
<td>1:48h</td>
<td>13.6</td>
</tr>
<tr>
<td>6</td>
<td>36.7M</td>
<td>2:10h</td>
<td>13.5</td>
</tr>
<tr>
<td>7</td>
<td>42.7M</td>
<td>2:36h</td>
<td>13.8</td>
</tr>
<tr>
<td>8</td>
<td>48.7M</td>
<td>3:14h</td>
<td>14.2</td>
</tr>
<tr>
<td>best MLP</td>
<td>42.7M</td>
<td>0:35h</td>
<td>15.3</td>
</tr>
<tr>
<td>(9x2000)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>best GMM</td>
<td>31.3M</td>
<td>–</td>
<td>23.6</td>
</tr>
</tbody>
</table>
Effect of ANNs in Acoustic Modelling

Compare three types of emission models in HMMs:

- GMM: Gaussian mixture model
- MLP: deep multi-layer perceptron
- LSTM RNN: recurrent neural network with long short-term memory

Experimental results for QUAERO English 2011:

<table>
<thead>
<tr>
<th>approach</th>
<th>layers</th>
<th>WER[%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>conventional: best GMM</td>
<td>–</td>
<td>30.2</td>
</tr>
<tr>
<td>hybrid: best MLP</td>
<td>9</td>
<td>20.3</td>
</tr>
<tr>
<td>hybrid: best LSTM RNN</td>
<td>6</td>
<td>17.5</td>
</tr>
</tbody>
</table>

Remarks:

- comparative evaluations in QUAERO 2011: competitive results with LIMSI Paris and KIT Karlsruhe
- best improvement over Gaussian mixture models by 40% relative using an LSTM RNN
Deep Learning for Language Modelling

Outline

Human Language Technology: Overview & History

Statistical Approach

Neural Network and Statistical Approach

Deep Learning for Acoustic Modelling

Deep Learning for Language Modelling

  History of Neural Networks in Language Modeling
  Perplexity vs. Word Error Rate
  Neural Network based Language Modeling
  Empirical Overview of Current Methods

Current State-of-the-Art in ASR

References
Outline

Human Language Technology: Overview & History

Statistical Approach

Neural Network and Statistical Approach

Deep Learning for Acoustic Modelling

Deep Learning for Language Modelling

History of Neural Networks in Language Modeling

Perplexity vs. Word Error Rate

Neural Network based Language Modeling

Empirical Overview of Current Methods

Current State-of-the-Art in ASR

References
History of Neural Networks in Language Modeling

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

Today: neural network based language models show competitive results.
Deep Learning for Language Modelling
Perplexity vs. Word Error Rate

Outline

Human Language Technology: Overview & History
Statistical Approach
Neural Network and Statistical Approach
Deep Learning for Acoustic Modelling

Deep Learning for Language Modelling
  History of Neural Networks in Language Modeling
  Perplexity vs. Word Error Rate
  Neural Network based Language Modeling
  Empirical Overview of Current Methods

Current State-of-the-Art in ASR

References
Reminder: perplexity (PP)

- geometric average of inverse probability → interpretation: average effective vocabulary size

\[
PP := \left( p(w_1^N) \right)^{-1/N} = \left( \prod_{n=1}^{N} p(w_n|w_1^{n-1}) \right)^{-1/N}
\]

define \( w_1^0 \) as empty sequence

---

64 of 78 Automatic Speech Recognition: State-of-the-Art in Transition - A Neural Paradigm Change?
KITP Workshop on the Physics of Hearing, KITP, Santa Barbara, CA
Schlüter et al. — Human Language Technology and Pattern Recognition
RWTH Aachen University — June 26, 2017
Extended Range: Perplexity vs. Word Error Rate

- empirical results, originally proposed by [Klakow & Peters 2002]
- analytical error bound exists [Schlüter & Nußbaum-Thom 2013] (upper bound only)
- proof of approximate power law still missing
Deep Learning for Language Modelling

Perplexity vs. Word Error Rate

Word Error Rate vs. Local Perplexity

(3-word window, 20 bins)
Deep Learning for Language Modelling
Neural Network based Language Modeling

Outline

Human Language Technology: Overview & History
Statistical Approach
Neural Network and Statistical Approach
Deep Learning for Acoustic Modelling

Deep Learning for Language Modelling
  History of Neural Networks in Language Modeling
  Perplexity vs. Word Error Rate
  Neural Network based Language Modeling
  Empirical Overview of Current Methods

Current State-of-the-Art in ASR

References
Neural Network based Language Modeling

- distinguish:
  - sub-symbolic processing: speech/audio, text images, image/video (computer vision)
  - symbolic processing: language modeling (and machine translation)

- word sequence $w_1^N := w_1...w_n...w_N$

- 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! [Mikolov & Karafiat+ 2010]
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 across 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
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 $w_1...w_n...w_N$

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

- input to RNN in position $n$:
  - output $h_{n-1}$ of hidden layer at position $(n-1)$
  - immediate predecessor word $w_{n-1}$
Deep Learning for Language Modelling
Neural Network based Language Modeling

**LSTM RNN**  [Hochreiter & Schmidhuber 1997, Gers & Schraudolph\(\dagger\) 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
Deep Learning for Language Modelling
Empirical Overview of Current Methods

Outline

Human Language Technology: Overview & History

Statistical Approach

Neural Network and Statistical Approach

Deep Learning for Acoustic Modelling

Deep Learning for Language Modelling
  History of Neural Networks in Language Modeling
  Perplexity vs. Word Error Rate
  Neural Network based Language Modeling
  Empirical Overview of Current Methods

Current State-of-the-Art in ASR

References
Empirical Overview of Current Methods

• 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:
    \[150k \cdot 600 + 600 \cdot 150k = 180M\]

<table>
<thead>
<tr>
<th>approach</th>
<th>hidden layers</th>
<th>PPL</th>
</tr>
</thead>
<tbody>
<tr>
<td>count model</td>
<td>–</td>
<td>163.7</td>
</tr>
<tr>
<td>10-gram MLP</td>
<td>2</td>
<td>136.5</td>
</tr>
<tr>
<td>RNN</td>
<td>1</td>
<td>125.2</td>
</tr>
<tr>
<td>LSTM-RNN</td>
<td>2</td>
<td>107.8</td>
</tr>
</tbody>
</table>

perplexity PPL on dev data:

observation:
(huge) improvement by 40%
Complexity: Computation Times

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

<table>
<thead>
<tr>
<th>Models</th>
<th>PPL</th>
<th>CPU Time (Order)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count model</td>
<td>163.7</td>
<td>30 min</td>
</tr>
<tr>
<td>MLP</td>
<td>136.5</td>
<td>1 week</td>
</tr>
<tr>
<td>LSTM-RNN</td>
<td>107.8</td>
<td>3 weeks</td>
</tr>
</tbody>
</table>

- 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
Interpolated Language Models: Perplexity and WER

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

<table>
<thead>
<tr>
<th>Models</th>
<th>PPL</th>
<th>WER [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>count model</td>
<td>131.2</td>
<td>12.4</td>
</tr>
<tr>
<td>+ 10-gram MLP</td>
<td>112.5</td>
<td>11.5</td>
</tr>
<tr>
<td>+ Recurrent NN</td>
<td>108.1</td>
<td>11.1</td>
</tr>
<tr>
<td>+ LSTM-RNN</td>
<td>96.7</td>
<td>10.8</td>
</tr>
<tr>
<td>+ 10-gram MLP with 2 layers</td>
<td>110.2</td>
<td>11.3</td>
</tr>
<tr>
<td>+ LSTM-RNN with 2 layers</td>
<td>92.0</td>
<td>10.4</td>
</tr>
</tbody>
</table>

- experimental result:
  - significant improvements by NN language models
  - best improvement in perplexity: 30% reduction (from 131 to 92)
  - best improvement in WER: 16% reduction (from 12.4% to 10.4%)
  - empirical observation:
    power law between WER and perplexity (cube to square root)
Current State-of-the-Art in ASR

Outline

Human Language Technology: Overview & History

Statistical Approach

Neural Network and Statistical Approach

Deep Learning for Acoustic Modelling

Deep Learning for Language Modelling

Current State-of-the-Art in ASR

Net Effect of NN Modeling in ASR

State-of-the-Art Results Switchboard

ASR vs. Human Performance

References
Outline

- Human Language Technology: Overview & History
- Statistical Approach
- Neural Network and Statistical Approach
- Deep Learning for Acoustic Modelling
- Deep Learning for Language Modelling

Current State-of-the-Art in ASR
- Net Effect of NN Modeling in ASR
- State-of-the-Art Results Switchboard
- ASR vs. Human Performance

References
Overall Improvements by ANNs in ASR

<table>
<thead>
<tr>
<th>Language Model</th>
<th>PP</th>
<th>Acoustic Model</th>
<th>WER[%]</th>
</tr>
</thead>
</table>
| Count Fourgram            | 131.2 | Gaussian Mixture 
  deep MLP 
  LSTM RNN    | 19.2 |
| + LSTM-RNN                | 92.0 | Gaussian Mixture 
  deep MLP 
  LSTM RNN    | 16.5 |

Remarks:
- overall improvements by ANNs: 50%
- lion’s share of improvement: acoustic model
- acoustic input features: optimized for model
Current State-of-the-Art in ASR
State-of-the-Art Results Switchboard

Outline

Human Language Technology: Overview & History
Statistical Approach
Neural Network and Statistical Approach
Deep Learning for Acoustic Modelling
Deep Learning for Language Modelling

Current State-of-the-Art in ASR
Net Effect of NN Modeling in ASR
State-of-the-Art Results Switchboard
ASR vs. Human Performance

References
Recent Switchboard State-of-the-Art Systems

Acoustic modeling
- convolutional models:
  - visual geometry group (VGG) - very deep convolutional network (adopted from CV)
  - residual nets (ResNet) - even deeper, incl. short-cut connections (adopted from CV)
  - layer-wise context expansion with attention (LACE) - TDNN + short-cuts + attention mask
- bidirectional long-short term memory (BLSTM) recurrent network (IBM+MSR)

Language modeling
- N-gram vs. LSTM-NN

Experimental results:
- challenging task
- training on 2000h
- single systems
- sites compared:
  - IBM Research [Saon & Kurata+ 17]
  - Microsoft Research (MSR) [Xiong & Droppo+ 17]

<table>
<thead>
<tr>
<th>site</th>
<th>acoustic model</th>
<th>LM, WER [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>N-gram</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SWB</td>
</tr>
<tr>
<td>IBM</td>
<td>BLSTM</td>
<td>7.2</td>
</tr>
<tr>
<td></td>
<td>ResNet</td>
<td>7.6</td>
</tr>
<tr>
<td>MSR</td>
<td>BLSTM</td>
<td>8.3</td>
</tr>
<tr>
<td></td>
<td>ResNet</td>
<td>8.6</td>
</tr>
<tr>
<td></td>
<td>VGG</td>
<td>9.1</td>
</tr>
<tr>
<td></td>
<td>LACE</td>
<td>8.4</td>
</tr>
</tbody>
</table>
Outline

Human Language Technology: Overview & History
Statistical Approach
Neural Network and Statistical Approach
Deep Learning for Acoustic Modelling
Deep Learning for Language Modelling

Current State-of-the-Art in ASR
Net Effect of NN Modeling in ASR
State-of-the-Art Results Switchboard
ASR vs. Human Performance

References
Human - Machine Comparison

How does state-of-the-art ASR compare against human performance?

- current best ASR systems obtained using system combination
- two human speech recognition studies

Results on Switchboard task cited from

- IBM Research [Saon & Kurata\textsuperscript{+} 17]
- Microsoft Research (MSR) [Xiong & Droppo\textsuperscript{+} 17]

<table>
<thead>
<tr>
<th>recognition</th>
<th>site</th>
<th>WER</th>
<th>[%] CH</th>
</tr>
</thead>
<tbody>
<tr>
<td>machine</td>
<td>MSR</td>
<td>5.8</td>
<td>11.0</td>
</tr>
<tr>
<td></td>
<td>IBM</td>
<td>5.5</td>
<td>10.3</td>
</tr>
<tr>
<td>human</td>
<td>MSR</td>
<td>5.9</td>
<td>11.3</td>
</tr>
<tr>
<td></td>
<td>IBM</td>
<td>5.1</td>
<td>6.8</td>
</tr>
</tbody>
</table>
Thank you for your attention

Any questions?
References

Outline

Human Language Technology: Overview & History

Statistical Approach

Neural Network and Statistical Approach

Deep Learning for Acoustic Modelling

Deep Learning for Language Modelling

Current State-of-the-Art in ASR

References
References


References


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.


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...
References


References


References


References


References


References


References


References


References


References


References


References


References


References


References


S. Wiesler, A. Richard, R. Schlüter, H. Ney: “Mean-normalized Stochastic Gradient for Large-Scale Deep Learning,” IEEE Intern. Conf. on Acoustics,
References


