RTTH Summer School on Speech Technology:
A Deep Learning Perspective
TALP Research Center, UP de Catalunya, Barcelona, Spain, 09-July-2015

Speech Recognition and Neural Networks

Hermann Ney
Human Language Technology and Pattern Recognition

RWTH Aachen University, Aachen, Germany
DIGITEO Chair, LIMSI-CNRS, Paris, France

My talk is based on joint work with members of my chair (Informatik 6):

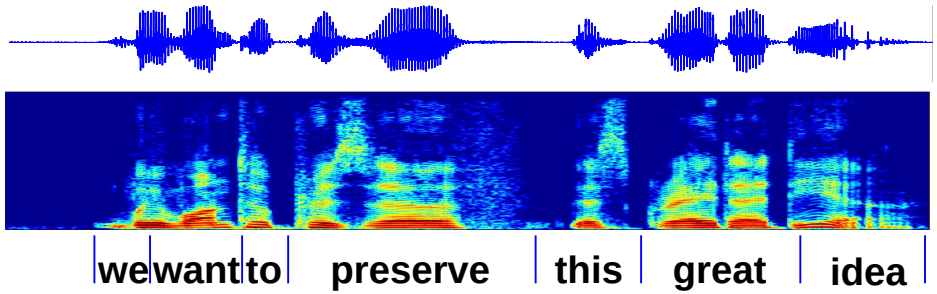
- **acoustic modelling:**
  Ralf Schlüter, Zoltan Tüske, Simon Wiesler, Muhammad Ali Tahir, ...
- **language modelling:**
  Martin Sundermeyer, Kazuki Irie, ...
- **conference papers:**
  Interspeech and ICASSP
- **public toolkits with open source code**
  for speech recognition and neural networks for acoustic and language modelling

webpage: RWTH Aachen, Informatik 6, Software

Outline

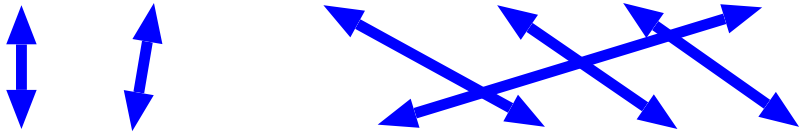
1 Speech and Language Technology

Speech Recognition



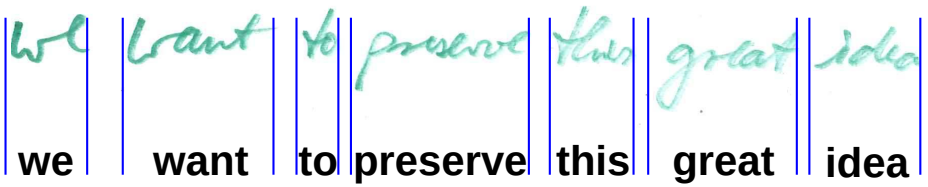
Machine Translation

wir wollen diese große Idee erhalten



we want to preserve this great idea

Text Image Recognition



three tasks for machine learning:
– automatic speech recognition
– text image recognition
– machine translation

terminology: tasks in speech and natural language processing (NLP)

- **automatic speech recognition (ASR)**
- **text image recognition (prined and handwritten text, offline)**
- **machine translation (MT) (of language and speech)**
- **...**

characteristic properties of these tasks:

- **well-defined 'classification' tasks:**
  – due to 5000-year history of (written!) language
  – well-defined classes: letters or words of the language
- **easy task for humans**
  (at least in their native language!)
- **hard task for computers**
  (as the last 40 years have shown!)

Projects

activities of RWTH team in joint projects:

- **TC-STAR 2004-2007: funded by EU**
  first research system for speech-to-speech translation on real-life data (EU parliament)
- **GALE 2005-2011: funded by US DARPA**
  emphasis on Chinese and Arabic speech and text
- **BOLT 2011-2015: funded by US DARPA**
  emphasis on colloquial text for Arabic and Chinese
- **QUAERO 2008-2013: funded by OSEO France**
  European languages, more colloquial speech, handwriting
- **BABEL 2012-2017: funded by US IARPA**
  spoken term detection with noisy and limited training data
- **EU projects 2012-2014: EU-Bridge, TransLectures**
  emphasis on recognition and translation of lectures (academic, TED, ...)

Speech and Language: Characteristic Properties

typical situation:

$$\text{input string} \rightarrow \text{output string}$$

tasks:

- **speech recognition:**
  speech signal $\rightarrow$ string of words/letters
- **recognition of image text (printed and written characters):**
  text image $\rightarrow$ string of words/letters
- **machine translation:**
  string of source words $\rightarrow$ string of target words/letters

common property:

output string = string of words/letters in a natural language

terminology:
– compound decision theory
– contextual pattern recognition
– structured output

Bayes Decision Rule

most general case:
– input sequence: $X := x_1 ... x_t ... x_T$
– output sequence: $W := w_1 ... w_n ... w_N$ **of unknown length** N
– true distribution $pr(W|X)$ **(can be extremely complex!)**

performance measure or loss function (e. g. edit distance)
between true output sequence $\tilde{W}$ **and hypothesized output sequence** W**:**

$$L[W, \tilde{W}]$$

Bayes decision rule minimizes expected loss:

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

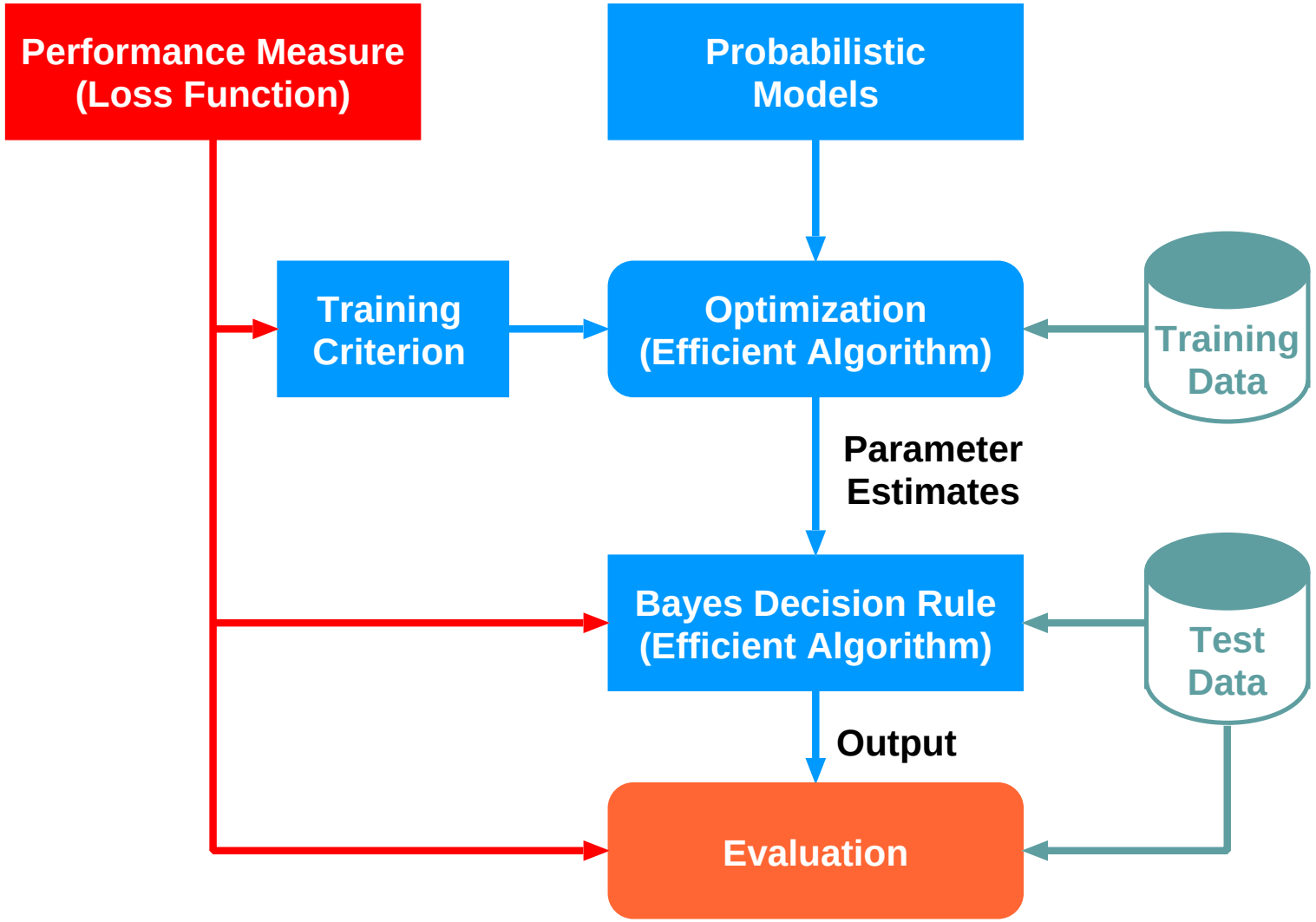
Under these two conditions:

$$L[W, \tilde{W}]: \quad \text{satisfies triangle inequality}$$
$$\max_{W} \{pr(W|X)\} > 0.5$$

we have [Schlueter & Ney 2005]:

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

Bayes Architecture for Speech Recognition (and other NLP tasks)



Speech Recognition = Modelling + Statistics + Efficient Algorithms

2 Speech Recognition: Principles

Problem in Bayes decision rule:

- **– true posterior distribution: unknown**
- **– to replace it, assume suitable model distributions with free parameters:**

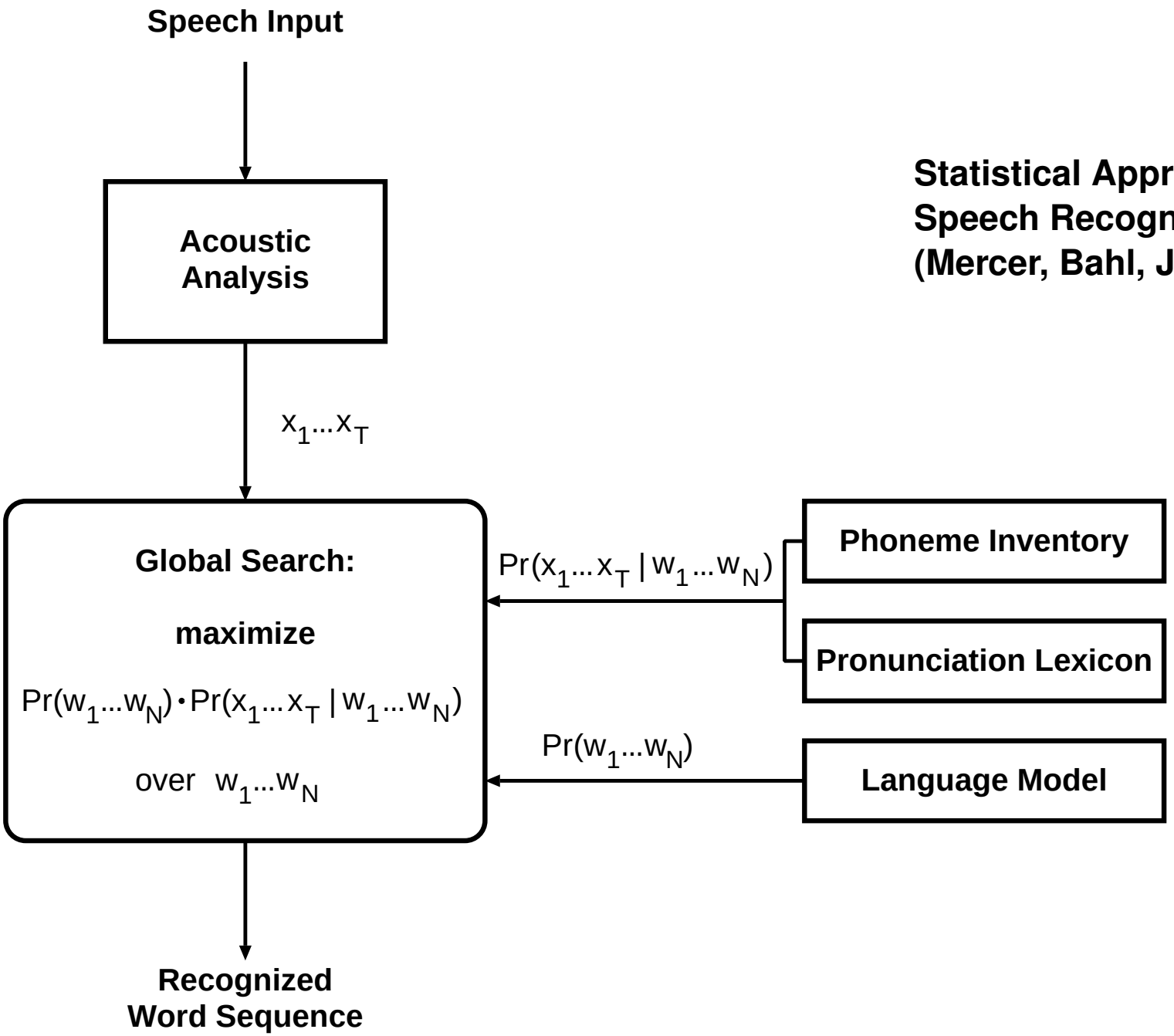
$$p(W|X) = \frac{p(W) \cdot p(X|W)}{\sum_{\tilde{W}} p(\tilde{W}) \cdot p(X|\tilde{W})}$$

- **– generative model: language model $p(W)$ and acoustic model $p(X|W)$**

structure of spoken language implies a hierarchy of levels:

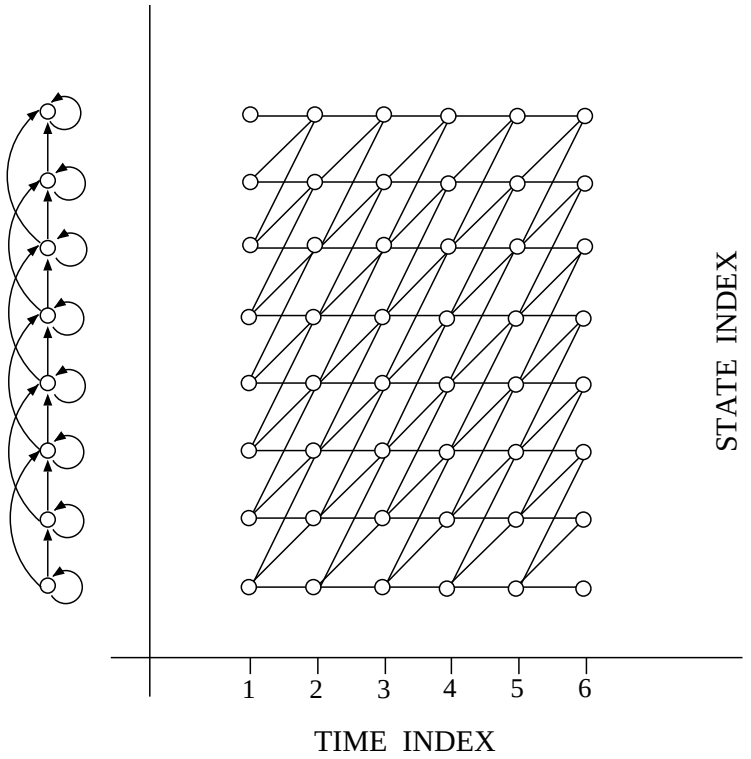
- **sentence: concatenation of words $W = w_1 ... w_n ... w_N$ modelled by (statistical) language model $p(W)$**
- **words: concatenation of phonemes (sounds) modelled by (conventional) pronunciation dictionary**
- **phonemes (sounds): time waveform**
  - **– acoustic (spectral) analysis every 10 msec**
  - **– result: sequence of observed acoustic vectors (10-ms events) $X = x_1 ... x_t ... x_T$**
  - **– fundamental problem: variability in speaking rate**

Statistical Approach to Automatic Speech Recognition (ASR) (Mercer, Bahl, Jelinek 1982)



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: $x_1^T = x_1 ... x_t ... x_T$**



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:**

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

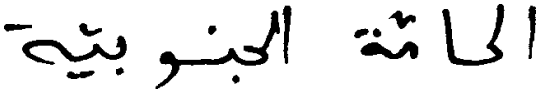
- **two types of distributions:**
  - **transition probability $p(s|s', W)$: not important**
  - **emission probability $p(x_t|s, W)$: key quantity realized by GMM: Gaussian mixtures models (trained by EM algorithm)**
- **phonetic labels (allophones, sub-phones): $(s, W) \rightarrow \alpha = \alpha_{sW}$**

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

  typical approach: phoneme models in triphone context: decision trees (CART) for finding equivalence classes

- **refinements:**
  - **augmented feature vector: context window around position t**
  - **subsequent LDA (linear discriminant analysis)**

image text recognition:
– define vertical slots over horizontal axis
– result: image signal = (quasi) one-dim. structure like speech signal

Language	Database	Example
French	RIMES	
Arabic	IfN/ENIT	
English	IAM	

3 Training: Generative vs. Discriminative

training phase:

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

training criteria in ASR:

- **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:
  – language model $p_\theta(W)$: without annotation!
  – acoustic model $p_\theta(X|W)$: with annotation!

- **sentence posterior probability (MMI = maximum mutual information) [1986 Mercer, 1991 Normandin, ...]:**

$$F(\theta) = \sum_r \log p_\theta(W_r|X_r) \qquad p_\theta(W|X_r) := \frac{p_\theta(W)\ p_\theta(X_r|W)}{\sum_{W'} p_\theta(W')\ p_\theta(X_r|W')}$$

- **(C.-H. Lee's) MCE: minimum classification error (old concept in pattern recognition) with smoothing parameter β**

$$F(\theta) = \sum_r \frac{1}{1 + \left(\frac{p_\theta(X_r, W_r)}{\sum_{W \neq W_r} p_\theta(X_r, W)} \right)^{2\beta}}$$

- **(Povey's) MWE/MPE: minimum word/phoneme error (= expected 'accuracy'):**

$$F(\theta) = \sum_r \sum_W p_\theta(W|X_r) \cdot A(W, W_r)$$

with the accuracy $A(W, W_r)$ of hypothesis W for correct sentence W_r:
= count of correct frame labels (e.g. phoneme: 1 out of 50)

practical details:
– 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

training of HMM:
– maximum likelihood by EM (expectation/maximization) algorithm
– looks like the ultimate and perfect solution

positive properties: Yes, it is the perfect solution.

- **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. $\max_\theta \prod_r p_\theta(W_r, X_r)$
  – virtually closed form solutions by EM algorithm

negative properties: No, it is not the perfect solution.

- **machinery of EM algorithm: nice from the mathematical point of view, but wrong criterion**
- **maximum likelihood considers** $p_\theta(X, W) = p_\theta(W) \cdot p_\theta(X|W)$**: this estimation problem is more complex than really required:** $p_\theta(W|X)$
- **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** $p_\theta(\alpha|x_t)$

4 Acoustic Modelling and ANN

ASR: automatic speech recognition
ANN: artificial neural networks

until 1989: ASR and ANN were 'independent' areas

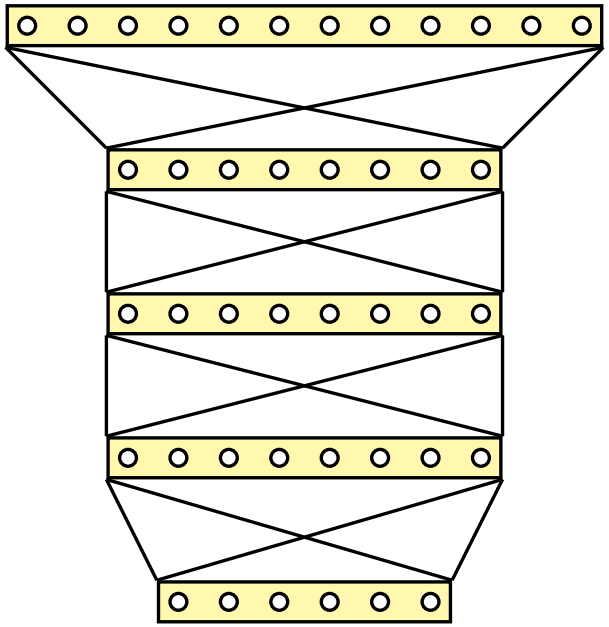
today's approach to ASR:

- **introduced by IBM research around 1980 and extended since then by many teams**
- **components:**
  - **language model**
  - **pronunciation model**
  - **phoneme/allophone models based on Hidden Markov Model (HMM)**

Artificial Neural Networks (ANN)

(first) renaissance around 1986
with various interpretations:

- **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 (!?)

training: (hard) optimization problem with
millions of parameters

History: ANN in Acoustic Modelling

hybrid approach in ASR:
in HMM, replace emission probability by ANN output

- **1990 Bourlard & Wellekens:**
  - **ANN outputs can be interpreted as probabilities (rediscovered)**
  - **they advocated the use of MLP to replace the GMM in HMM**
- **1990 Bridle:**
  softmax operation for probability normalization
- **1994 Robinson: recurrent neural network**
  - **competitive results on WSJ task**
  - **work remained a singularity in ASR**
- **...**

experimental situation:

until recently, ANNs were never really competitive with GMMs

History: ANN in Acoustic Modelling

related approaches:

- **1989 Waibel, Hanazawa, Hinton, Shikano, Lang:**
  Phoneme recognition using time-delay neural networks
- **1996 Fritsch, Finke, Waibel:**
  Hierarchical mixtures of experts
- **1997 Hochreiter & Schmidhuber:**
  Long short-term memory neural computation
- **1998 Yann LeCun: convolutional neural networks**

(second) renaissance: concepts of deep learning:

- **2000 Hermansky & Ellis: tandem approach: ANN for feature extraction in ASR**
- **2002 Utgoff & Stracuzzi: many-layered learning for symbolic processing**
- **2006 Hinton: deep learning (belief nets)**
- **2011 Seide, Deng, Yu (Microsoft):**
  – combined deep learning with hybrid approach
  – significant improvement by deep MLP on a large-scale task

consider modelling the acoustic vector x_t:

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

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

  – prior probability $p(\alpha)$: estimated as relative frequencies
  – for recognition purposes: the term $p(x_t)$ can be dropped

- result: model the label posterior probability by an ANN:

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

  rather than the state emission distribution $p(x_t|\alpha)$

- 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!)

consider class posterior probability of Gaussian model (= emission model of an HMM) for observation $x = x_t \in \mathbb{R}^D$ and class $c = \alpha$:

$$p(c|x) = \frac{p(c)\,\mathcal{N}(x|\mu_c, \Sigma_c)}{\sum_{c'} p(c')\,\mathcal{N}(x|\mu_{c'}, \Sigma_{c'})} = \ldots = \frac{1}{Z(x)} \cdot \exp(x^T \Lambda_c x + \lambda_c^T x + \gamma_c)$$

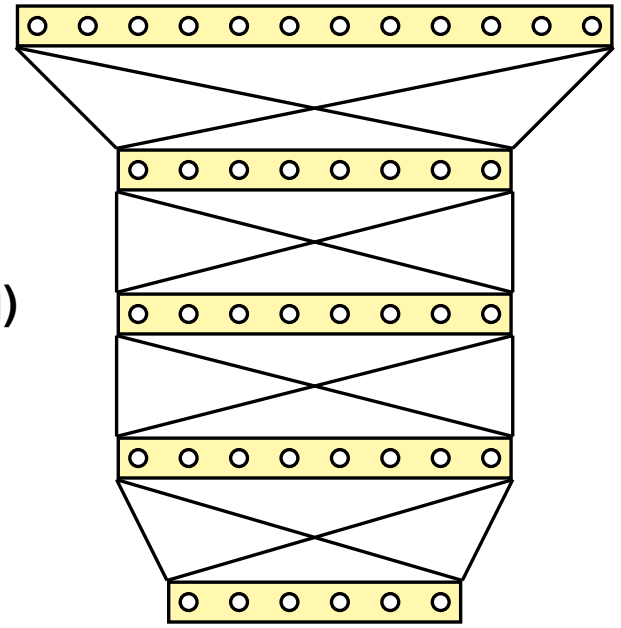
with the parameters: $\gamma_c \in \mathbb{R}, \lambda_c \in \mathbb{R}^D, \Lambda_c \in \mathbb{R}^{D \cdot D}$:

$$\begin{aligned}
\gamma_c &:= -\frac{1}{2}\mu_c^t \Sigma_c^{-1} \mu_c - \frac{1}{2}\log\det(2\pi\Sigma_c) + \log p(c) \\
\lambda_c &:= \mu_c^t \Sigma_c^{-1} \\
\Lambda_c &:= -\frac{1}{2}\Sigma_c^{-1}
\end{aligned}$$

observations:
– result: log-linear model
– natural training criterion:
  class posterior probability (= MMI, $\rightarrow$ convex optimization problem)
– class independent covariance matrix $\Sigma_c = \Sigma$:
  softmax operation

Hybrid Approach: Feature Functions

- open problem in log-linear models:
  how to find good feature functions?
- neural networks in hybrid approach:
  - output layer: softmax = log-linear model
  - feature function: remaining part of neural net
  - training criterion: cross entropy (as in log-linear modelling)
  - disadvantage: convexity is lost!
- polynomial features:
  new features: = polynomial expansion of raw features



Recognition Experiments

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 acoustic model:

- **feature vector: 16 MFCC (mel frequency cepstral coefficients)**
- **augmented feature vector:** $9 \cdot 16$
- **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 Mixtures: Training Criteria

word error rates [%]:

Training Criterion	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

Log-linear Model: Polynomial Features

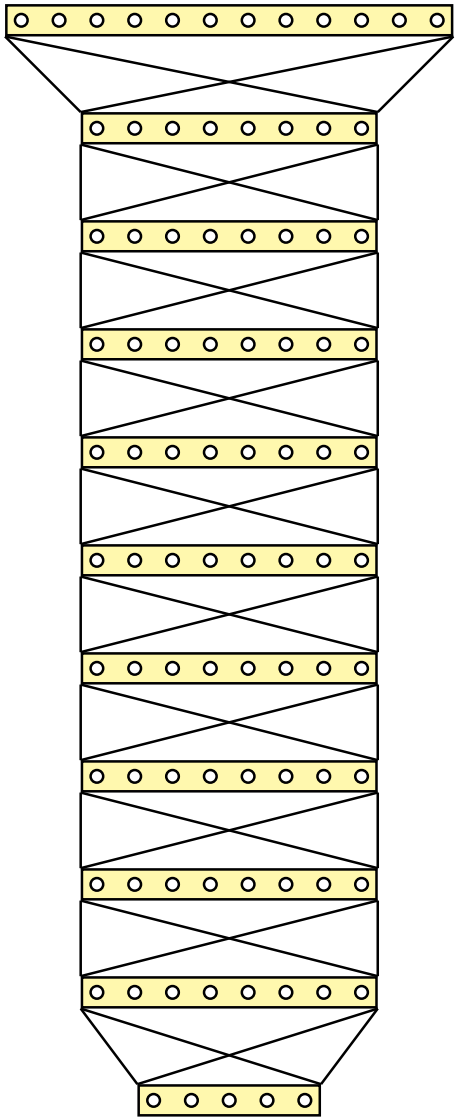
method for generating polynomial features:

- **raw features: 493 = 17 · 16 MFCC + 13 · (16 Δ MFCC + 1 $\Delta^2 E$)**
- **polynomial features:**
  - **squaring features and subsequent reduction by linear transformation**
  - **repeat this operation: orders 1,2,3**
- **free paramaters to be trained:**
  reduction matrices and log-linear model parameters

word error rates [%] for 50h training (criterion: frame MMI = cross-entropy)

Approach	order	parameters	dev	eval
Gaussian mixtures with frame MMI	–	31.3M	23.9	30.9
log-linear model	1	1.7M	24.3	31.8
	2	2.8M	21.2	27.4
	3	3.9M	20.7	27.0
best MLP (RLU, 6 hidden layers)	6	30.0M	17.7	23.5

Deep MLP: Number of Hidden Layers



word error rates [%] for 50-h training corpus as a function of number of hidden layers:

structure of MLP:

- input: 493 (like polynomial features)
- 2000 nodes per hidden layer
- nonlinearity: sigmoid
- number of parameters:
  6-layer MLP contains 30.0M parameters:
  $493 \cdot 2000 + 5 \cdot 2000^2 + 2000 \cdot 4501$
  $= 30.0M$

best relative improvement over GMM: 20%

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

Practicalities of ANN 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 very important**
- **implementation: proprietary (similar to Quicknet)**
- **use of GPUs: speed-up by a factor 10 over multithreaded CPUs**

public toolkit with source code: RWTH, Informatik 6, software

MLP: Effect of Width

structure: single hidden layer with sigmoid function

word error rates as a function of the width of the hidden layer:

Width	WER [%]	
	dev	eval
1000	26.2	33.3
2000	24.5	31.3
3000	23.8	30.6
15000	23.2	30.2

conclusion: depth is much more important than width!

Training Criterion: MPE vs. CE

word error rates for two training criteria:
– baseline: cross-entropy = frame MMI
– MPE: minimum phone error ('discriminative sequence training')

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

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, 2000 nodes per hidden layer**
- **training condition:**
  - **L2 regularization (weight decay): important**
  - **momentum term**

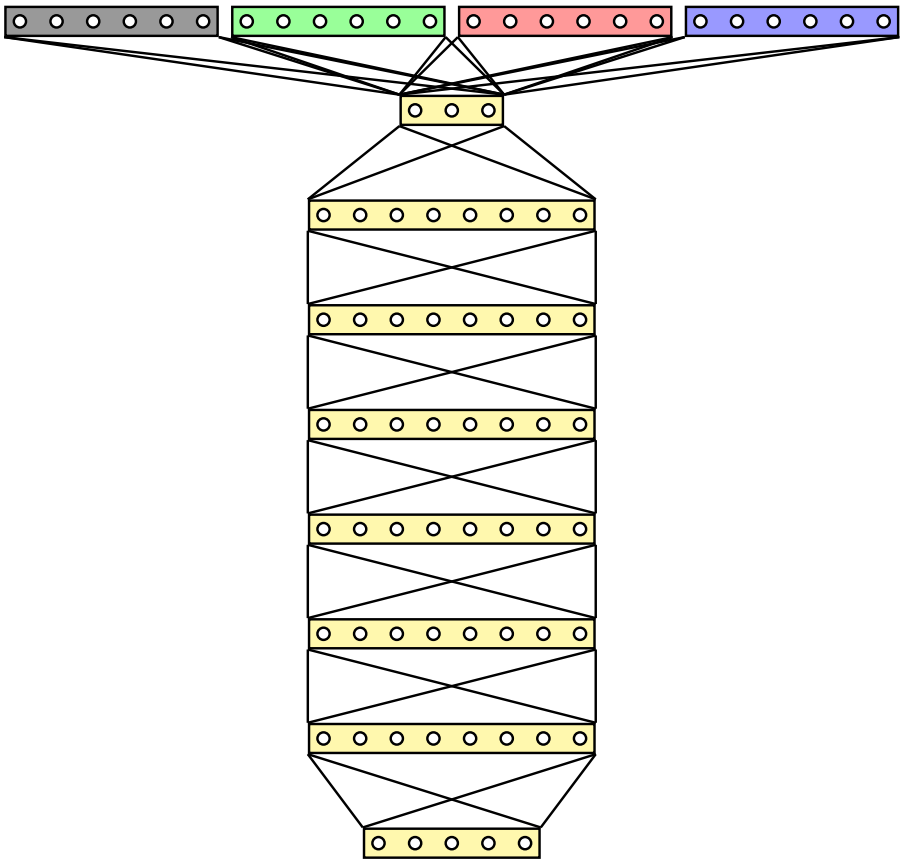
word error rates for activations functions: sigmoid vs. RLU:

activity 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

experimental result: improvement of 5-10% by RLUs

Effect of Multilingual Training

- experiments using a highly tuned system:
  - input: 1501 GammaTone features
  - MLP: RLU, 12 hidden layers with 2000 nodes
  - output: 12000 CART labels
  - MPE training
- additional training data:
  300h French, 150h German, 100h Polish
- after training using the four languages:
  adapting to the target language (English)
  using a few epochs



word error rates for English test data
(training on 250/550 hours):

	WER [%]	
	dev	eval
baseline system	14.7	19.6
tuned system	12.0	16.2
multilingual tuned system	11.6	15.6

Effect of Pretraining (different data!)

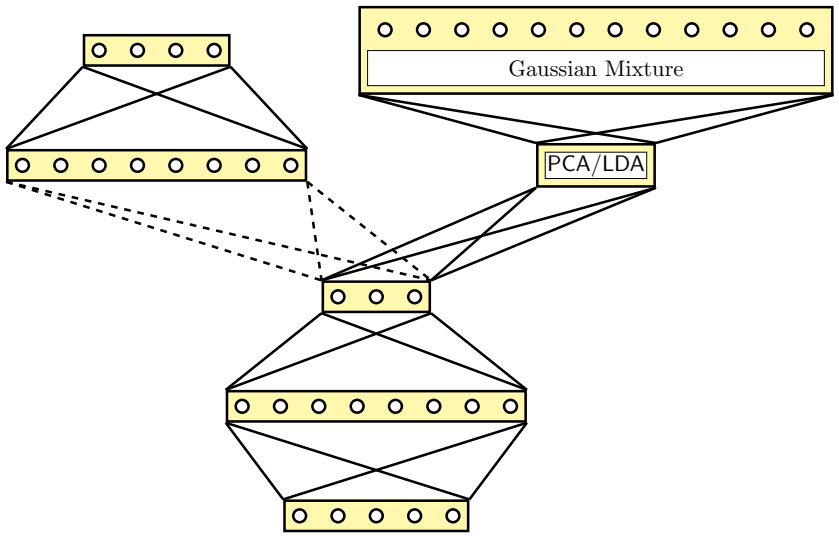
two different language for broadcast news and conversations:

- **French**
  - **training: 350h**
  - **Sigmoid MLP, 3-hidden layer**
- **German**
  - **training: 150h**
  - **Sigmoid MLP, 6-hidden layer**

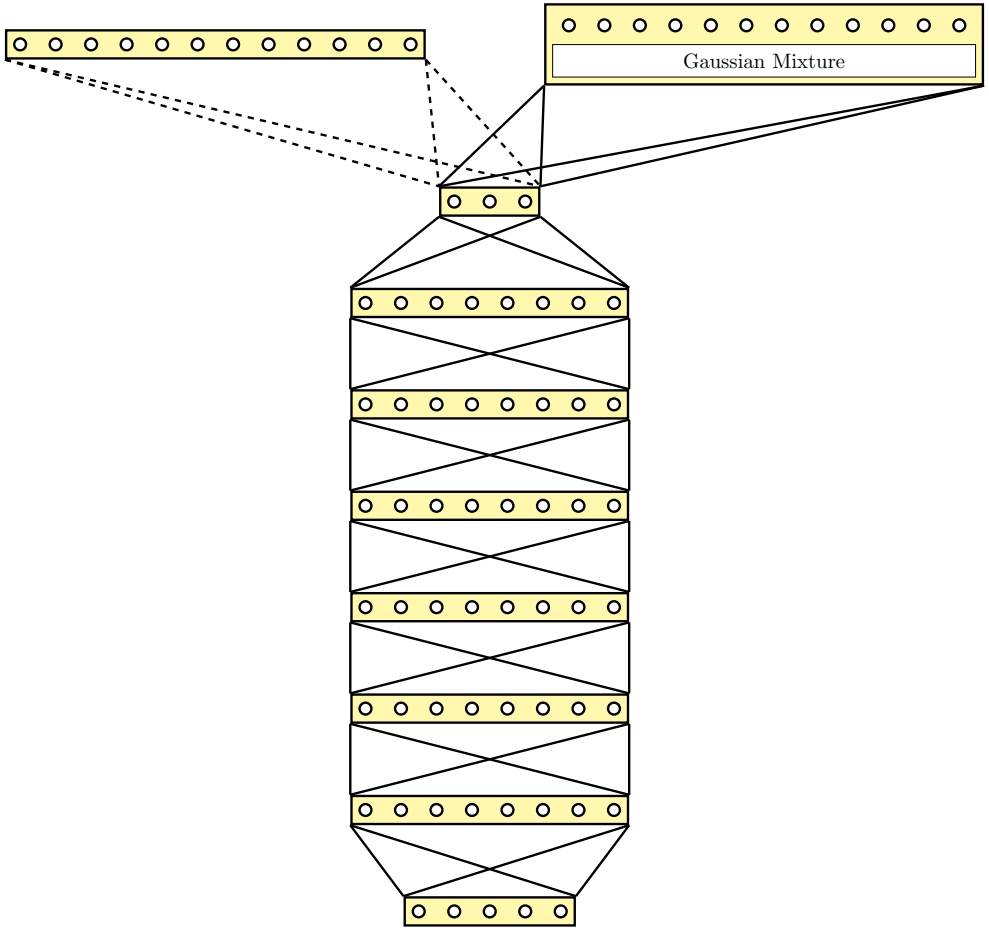
Task	Pretraining	WER [%]	
		dev	eval
FR	no	21.0	21.5
	yes	20.9	21.4
GE	no	16.8	
	yes	17.2	

approach:

- tandem: use MLP for feature extraction in an Gaussian mixture model (Hermansky, Ellis, Sharma 2000)
- bottleneck: one narrow hidden layer (Grezl et al. 2007; Grezl & Fousek 2008)



RWTH's Tandem Structure



Comparison: Tandem vs. Hybrid Approach

general structure of a tandem system:

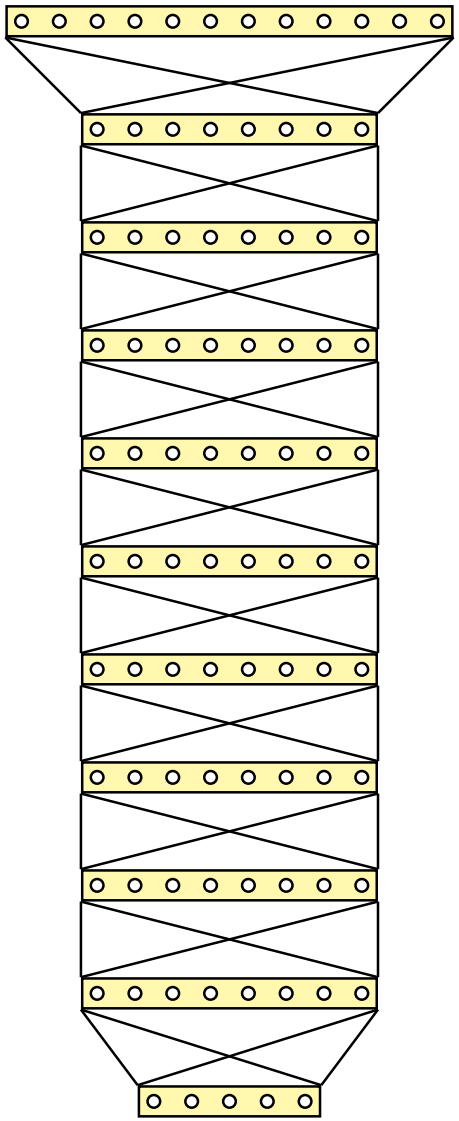
- first part: feature extraction
  - modelled by an MLP: RLU, 6 hidden layers with 2000 nodes each and bottlneck layer with 256 nodes
  - training (like a hybrid system): cross-entropy (CE) = frame MMI
- second part: emssion distributions
  - Gaussian mixture models (GMM)
  - training: maximum likelihood (ML)

new idea for tandem approach: JOINT training of MLP and GMM by CE

word error rates for English test data (training on 50 hours):

	#params	training criterion	WER [%]	
			dev	eval
hybrid: MLP only	30M	CE	17.7	23.5
tandem: MLP+GMM	58M	–	–	–
seq. training		MLP:CE + GMM: ML	18.5	24.4
joint training		(MLP + GMM): CE	17.2	22.9

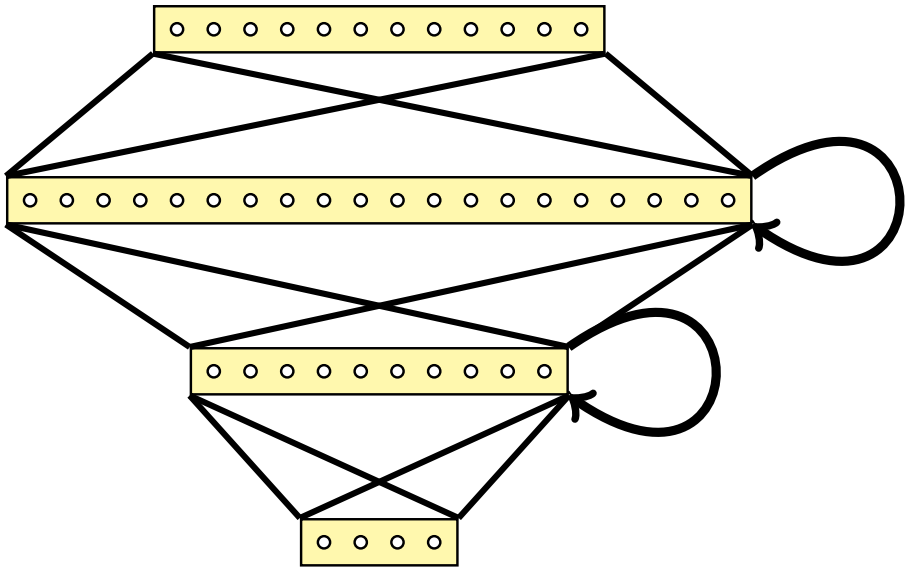
(Simplified) Summary: What is Different Now after 25 Years?

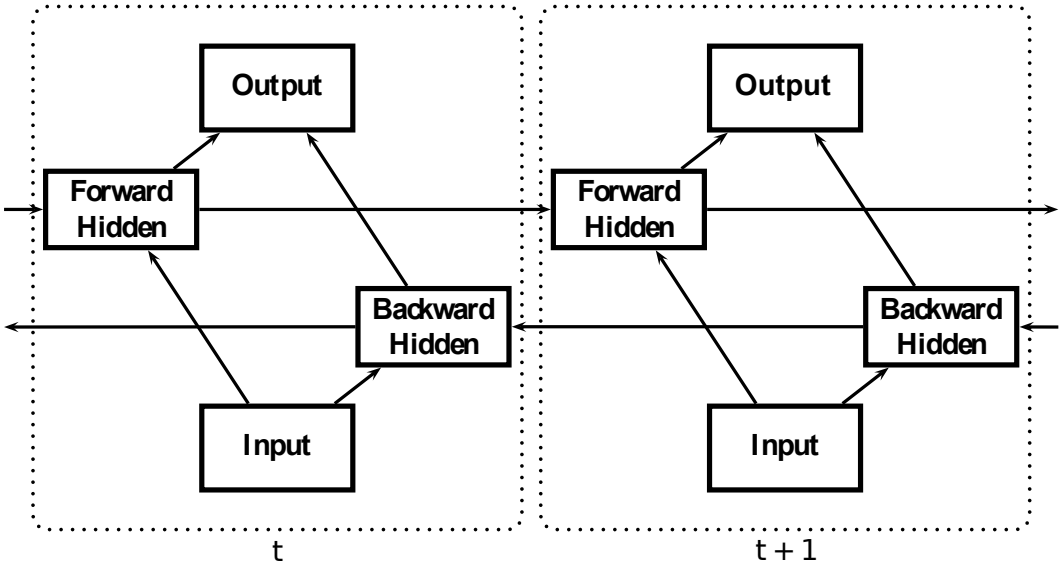


comparison: today's systems vs. 1989-1994:

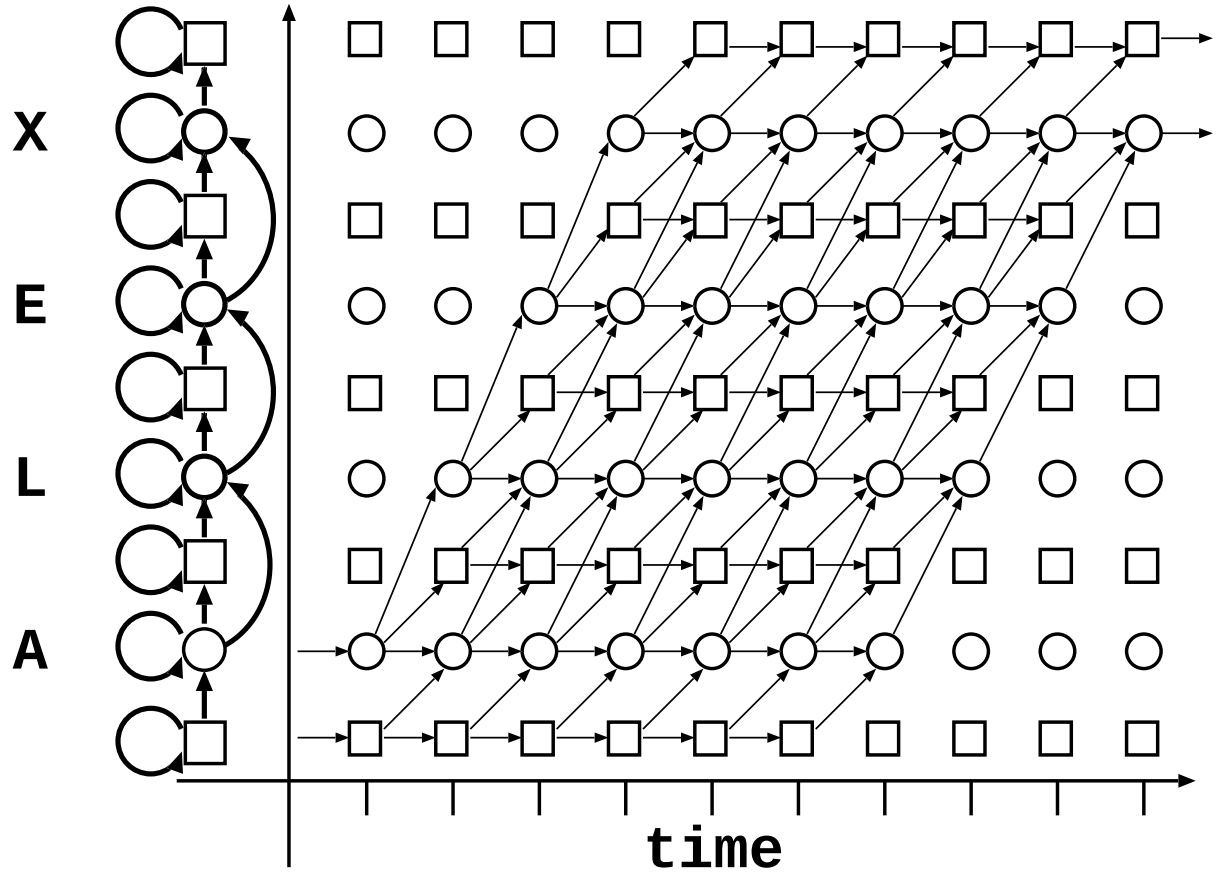
- **number of hidden layers: 10**
  rather than 2-3
- **number of output nodes: 5000**
  rather than 50
- **optimization strategy:**
  practical experience and heuristics,
  e.g. layer-by-layer pretraining
- **computation power: much more**

(Bidirectional) Recurrent Neural Network
[LSTM: Hochreuter & Schmidhuber 1997]





CTC: Connectionist Temporal Classification
[Graves, Fernandez, Gomez, Schmidhuber 2006]



5 Language Modelling and ANN

so far: *sub-symbolic* processing:
speech/audio, text images, image/video (computer vision)

now: *symbolic* processing for language modelling (and translation):

- **1989 M. Nakamura & K. Shikano:**
  English word category prediction based on neural networks.
- **1997 J.M. Castro, F. Casacuberta, F. Prat:**
  Towards connectionist language models.
- **1997 A. Castano, F. Casacuberta:**
  A connectionist approach to machine translation.
- **2007 H. Schwenk:**
  Continuous space language models.
- **2006 H. Schwenk, M.R. Costa-jussa, J.A.R. Fonollosa:**
  Smooth bilingual n-gram translation.
- **2012 H. Son Le, A. Allauzen, F. Yvon:**
  Continuous space translation models with neural networks.

today: ANNs in language show competitive results.

Review: Language Modelling

language model: word sequence $w_1^N := w_1 ... w_n ... w_N$

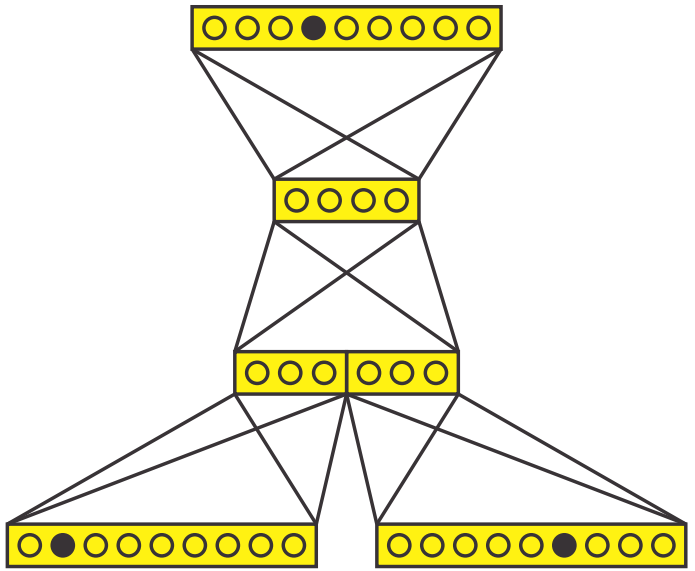
language model: conditional probability $p(w_n|w_0^{n-1})$

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

with artificial start symbol w_0

approaches to modelling $p(w_n|w_0^{n-1})$

- **count models (Markov chain):**
  - **limit history** w_0^{n-1} **to** k **predecessor words**
  - **smooth relative frequencies**
- **MLP models:**
  - **limit history too**
  - **use predecessor words as input to MLP**
- **RNN models: unlimited history!**



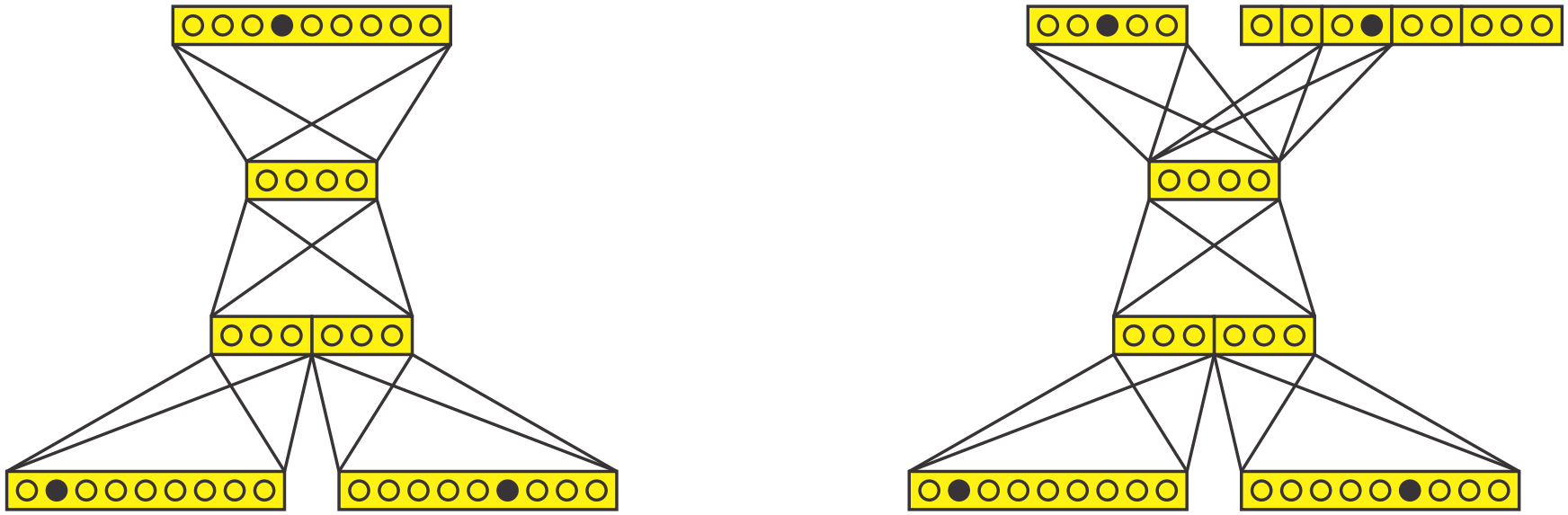
Feedforward Multi-Layer Perceptron (MLP)

structure of feedforward MLP:

- **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**

MLP without and with Word Classes: Two Predecessor Words



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 classe g for each word h

Experimental Conditions

- results on Quaero English (like before):
  - vocabulary size: 150k words
  - training text: 50M words
  - development corpus: 39k words
  - evaluation corpus: 35k words
- structure of MLP:
  - projection layer: 300 nodes
  - hidden layer: 600 nodes
  - size of MLP is dominated by input and output layers:
    $150k \cdot 300 + 600 \cdot 150k = 135M$
- perplexity on development data

Approach	PPL
4-gram count model	163.7
10-gram MLP	136.5
10-gram MLP with 2 layers	130.9

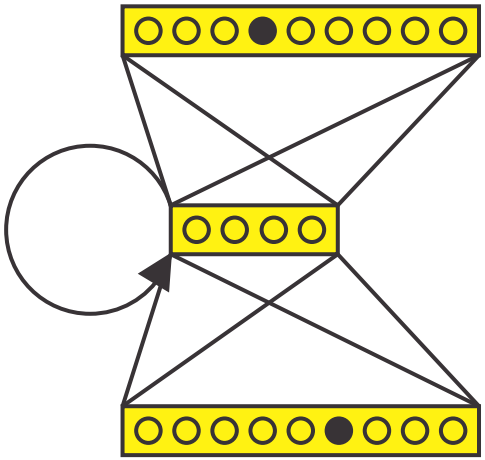
RNN without and with Word Classes

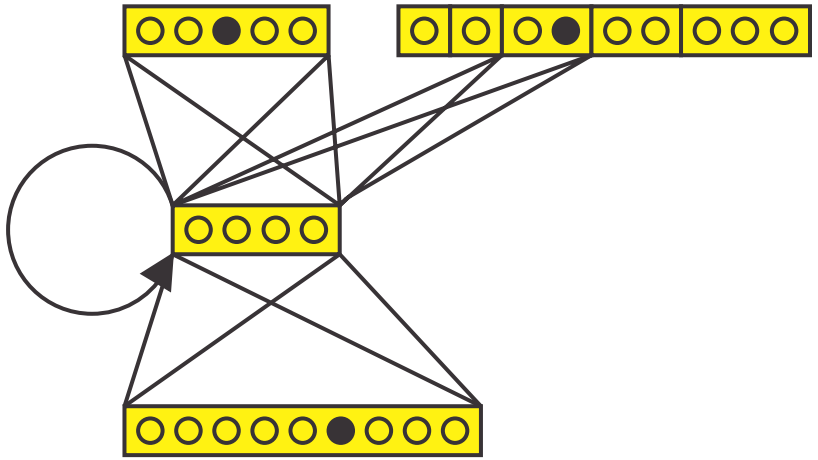
ANN with memory for sequence processing:
left-to-right processing of word sequence $w_1 ... w_n .... w_N$:

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

with output h_{n-1} of hidden layer at position $(n-1)$

input to RNN: immediate predecessor word w_{n-1}

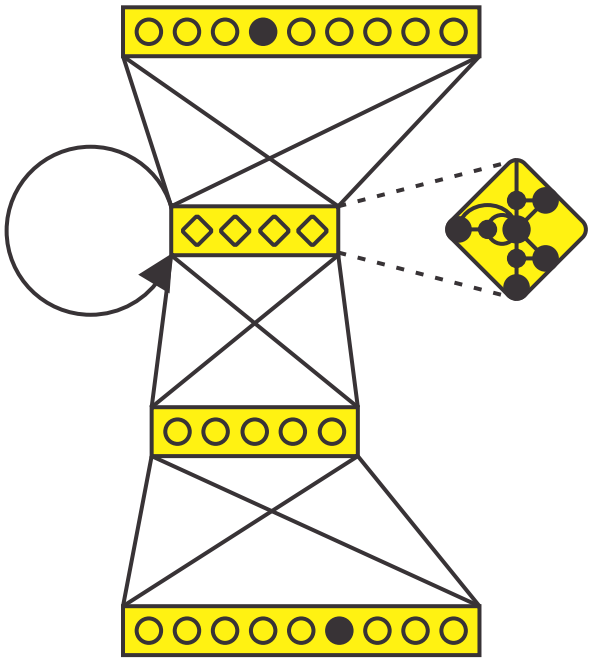




LSTM RNN

refinement of RNN:
LSTM = long-short term memory

– RNN: problems with vanishing gradients
– remedy: cells with gates rather than nodes
– details: see literature



Summary of Results: Perplexity Only

structure of recurrent neural nets (RNN):
– projection and hidden layer: each 600 nodes
– size of RNN is dominated by input and output layers (like for MLP):
$150k \cdot 600 + 600 \cdot 150k = 180M$

Perplexity PPL of four approaches on dev data:

Approach	PPL
count model	163.7
10-gram MLP	136.5
RNN	125.2
LSTM-RNN	107.8
10-gram MLP with 2 layers	130.9
LSTM-RNN with 2 layers	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
  – ANN models: trained on a small corpus: 50 Million words
- resulting language model:
  linear interpolation of TWO models

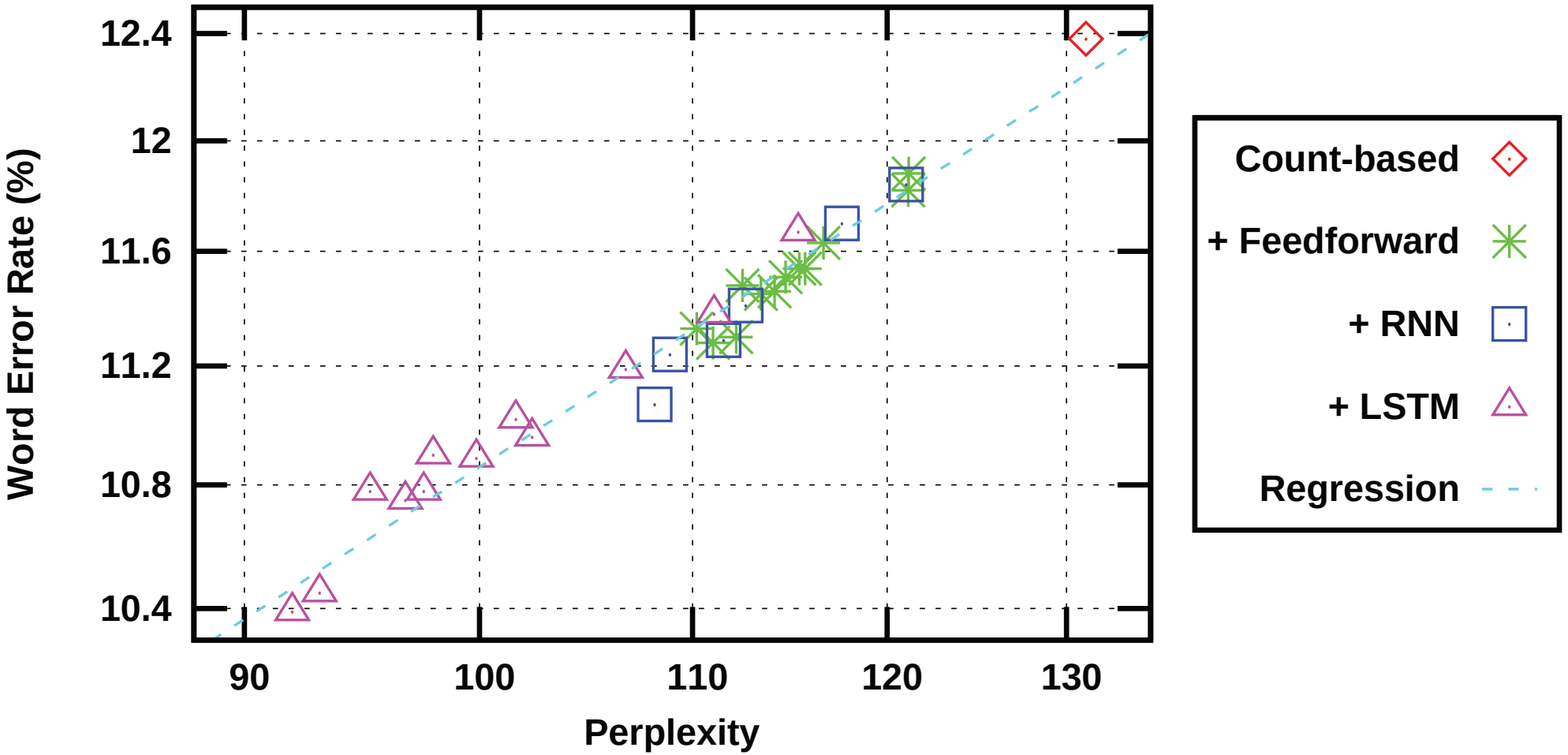
Interpolated Language Models: Perplexity and WER

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

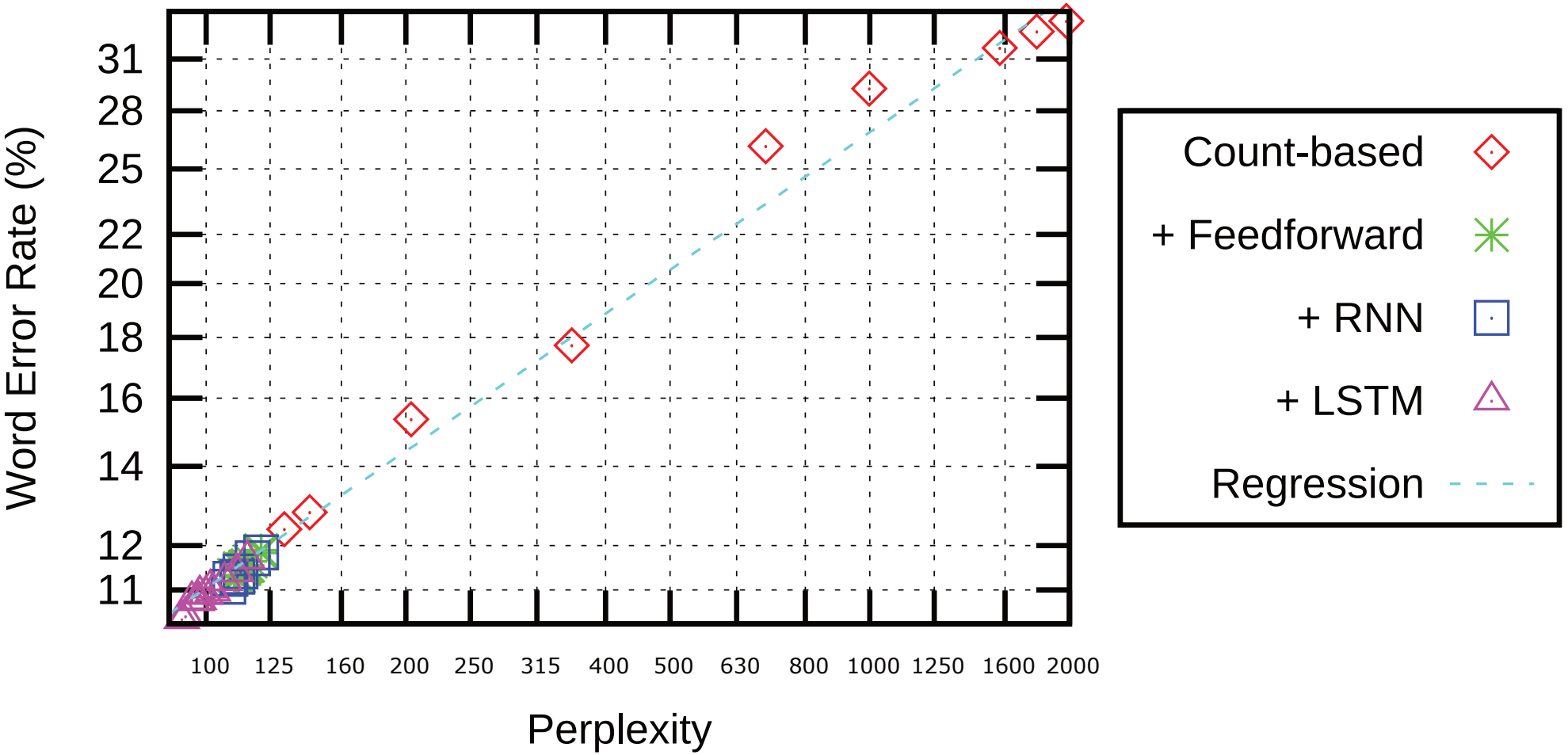
Models	PPL	WER[%]
count model	131.2	12.4
+ 10-gram MLP	112.5	11.5
+ Recurrent NN	108.1	11.1
+ LSTM-RNN	96.7	10.8
+ 10-gram MLP with 2 layers	110.2	11.3
+ LSTM-RNN with 2 layers	92.0	10.4

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

Plot: Perplexity vs. Word Error Rate



Extended Range: Perplexity vs. Word Error Rate



6 Conclusions

deep MLPs and ANNs:

- **result in significant improvements**
- **seem to learn 'better' (than conventional models), for both acoustic modelling and language modelling**
- **maybe: computationally expensive (in training), but cleaner/simpler architecture for acoustic modelling**
- **overfitting: apparently no problem, due to cross-validation (early stopping) and MLP structure (?)**
- **general experience in ASR:**
  The correct principles are not sufficient, we must get ALL the details right.
- **long-term view:**
  it took 25 years or more ...
- **...**

Conclusions

ANN and statistics:

- **ANNs provide one type of probabilitic models, they are part of the statistical approach**
- **they must be seen in the context of the other architecture issues:**
  - **– choice of performance measure: errors at string, word, phoneme, frame level**
  - **– probabilistic dependencies and the interaction between these levels**
  - **– Bayes decision rule along with an efficient implementation**
  - **– training criterion and optimization algorithm**
- **ANNs are useful for other tasks: image text recognition, machine translation, computer vision, ...**

THE END