A Comparison of Sequence-Trained Deep Neural Networks and Recurrent Neural Networks Optical Modeling For Handwriting Recognition

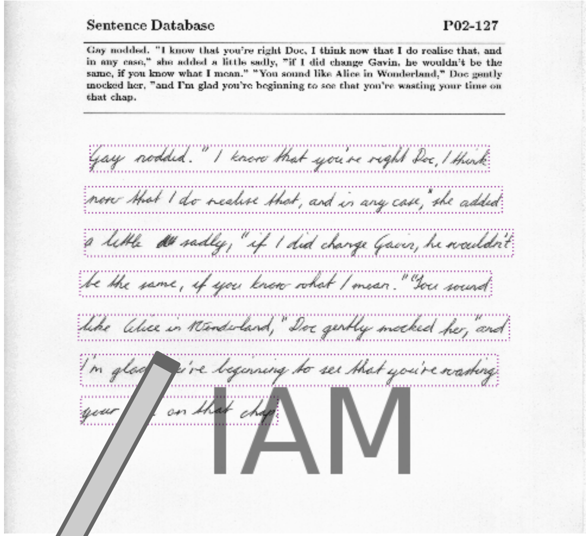
Théodore Bluche, Hermann Ney, Christopher Kermorvant

SLSP'14, Grenoble October 16, 2014

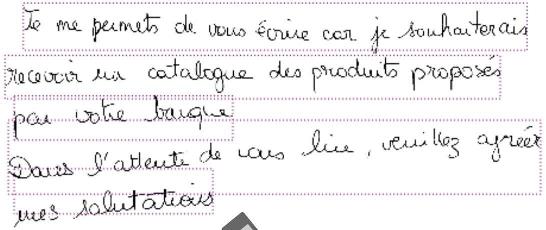
Outline

- Handwriting Recognition with Hybrid NN/HMM
  - Offline Handwriting Recognition
  - Experimental Setup
- Deep Neural Networks (DNN)
  - DNN training, Sequence-Discriminative Training
  - Results
- Recurrent Neural Networks (RNN)
  - LSTM, CTC, Depth, Dropout
  - Results
- Conclusions
  - System Combination and Final Results
  - Future Work

Offline Handwriting Recognition



IAM

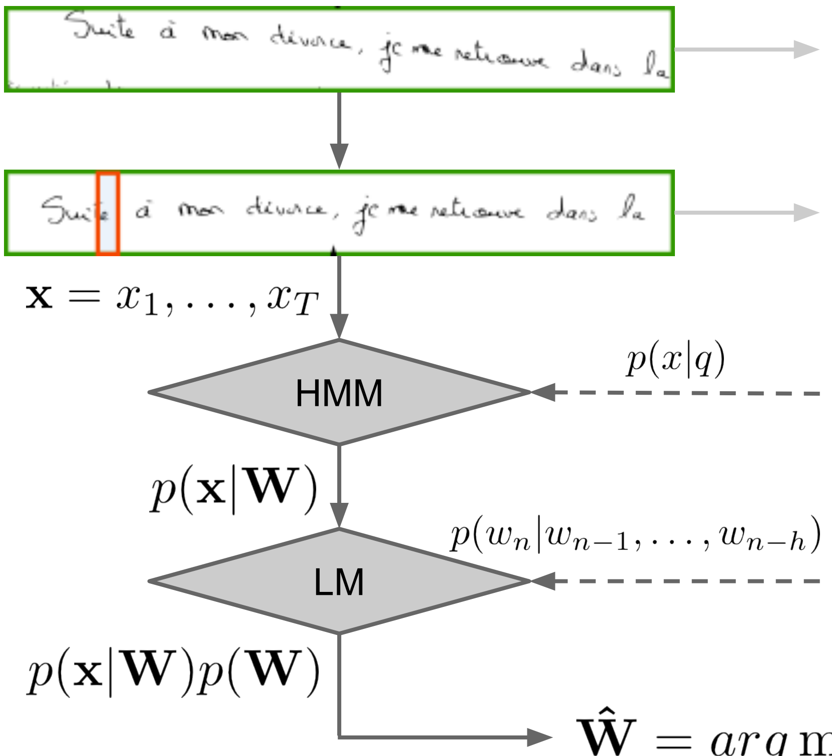


Rimes

Gay nodded. " I know that ...

Je me permets de vous écrire ...

Handwriting Recognition



$$\hat{\mathbf{W}} = arg\max_{\mathbf{W}} p(\mathbf{W}|\mathbf{x}) = arg\max_{\mathbf{W}} p(\mathbf{x}|\mathbf{W})p(\mathbf{W})$$

Preprocessing — slope and slant correction, contrast enhancement, region-dependent height normalization (72px)

Feature extraction — sliding window, extraction of vectors of handcrafted features, or extraction of pixel intensities

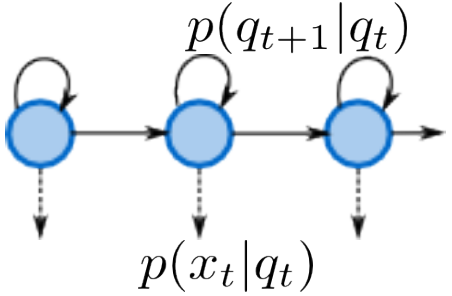
Optical Modeling — state emission probabilities, with GMMs (generative), or transformed posteriors from neural networks (discriminative)

Language Modeling — Lexicon (sequences of characters → sequences of words) + statistical *n-gram* language model (probability of word given history of *n-1* words)

Hybrid NN / HMM

Hidden Markov Model (HMM)

- Characters are modeled by state sequence (first-order Markov chain) – 6 for IAM, 5 for Rimes
- Associated with a transition model …
- … and an **emission model**: generative, usually mixtures of Gaussians (GMMs)



Hybrid Neural Network (NN) / HMM

- NN computes state posterior probability given input vector (usually a concatenation of several consecutive frames)
- **Rescaling by state priors**, we get a discriminative NN emission model to replace GMMs

$$\frac{p(q_t|x_t)}{p(q_t)} \approx \frac{p(x_t|q_t)}{p(x_t)}$$

Experimental Setup — Databases

IAM - English

	Pages	Lines	Words (7,843)	Characters (79)
Train	747	6,482	55,081	287,727
Validation	116	976	8,895	43,050
Test	336	2,915	25,920	128,531

Rimes - French

	Pages	Lines	Words (8,061)	Characters (99)
Train	1,351	10,203	73,822	460,201
Validation	149	1,130	8,380	51,924
Test	100	778	5,639	35,286

Experimental Setup — LM

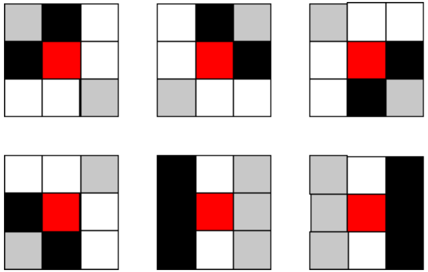
Optical model recognizes text lines, but the **LM is incorporated at the paragraph level** (makes more sense w.r.t sentence boundaries)
→ **1-3% absolute WER improvement**

				Validation		*Test*	
Database	**Voc.Size**	**Corpus**	**LM**	**OOV%**	**PPL**	**OOV%**	**PPL**
IAM	50,000	LOB* + Brown + Wellington	3-gram with mod. KN	4.3%	298	3.7%	329
Rimes	12,000	Training set annotations	4-gram with mod. KN	2.9%	18	2.6%	18

** the lines of the LOB corpus present in the validation and evaluation data have been ignored in the training of the language model*

Experimental Setup — Features

Handcrafted features



- Sliding window of 3px, with 3px step
- **56 handcrafted features** extracted from each frame
  - 3 pixel density measures in the frame and different horizontal regions
  - 2 measures of the center of gravity
  - 12 pixel configuration relative counts (6 from the whole frame and 6 in the core region)
  - 3 pixel density in vertical regions
  - HoG in 8 directions
  - + deltas (= 28 + 28)

Pixels

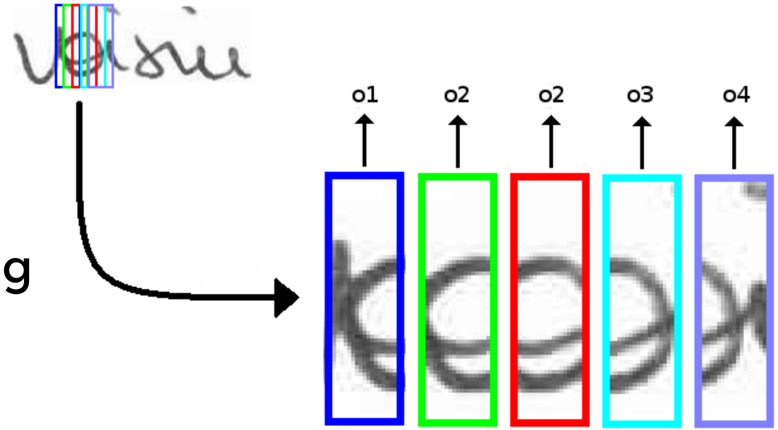
- Sliding window of 45px, with 3px step
- Rescaled to 20 x 32px (keeps aspect-ratio)
- Extraction of the **640 gray-level pixel intensities** per frame

Deep Multi-Layer Perceptrons

Deep Multi-Layer Perceptrons are …

- ★ … MLPs with many (3+) hidden layers
- ★ … state-of-the-art and now standard in HMM-based speech recognition
- ★ … widely used in Computer Vision, e.g. for isolated character or digit recognition
- ★ … not yet applied to HMM-based unconstrained handwritten text line recognition

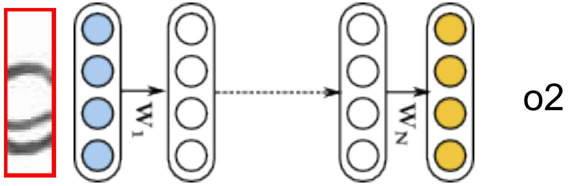
DNN Training



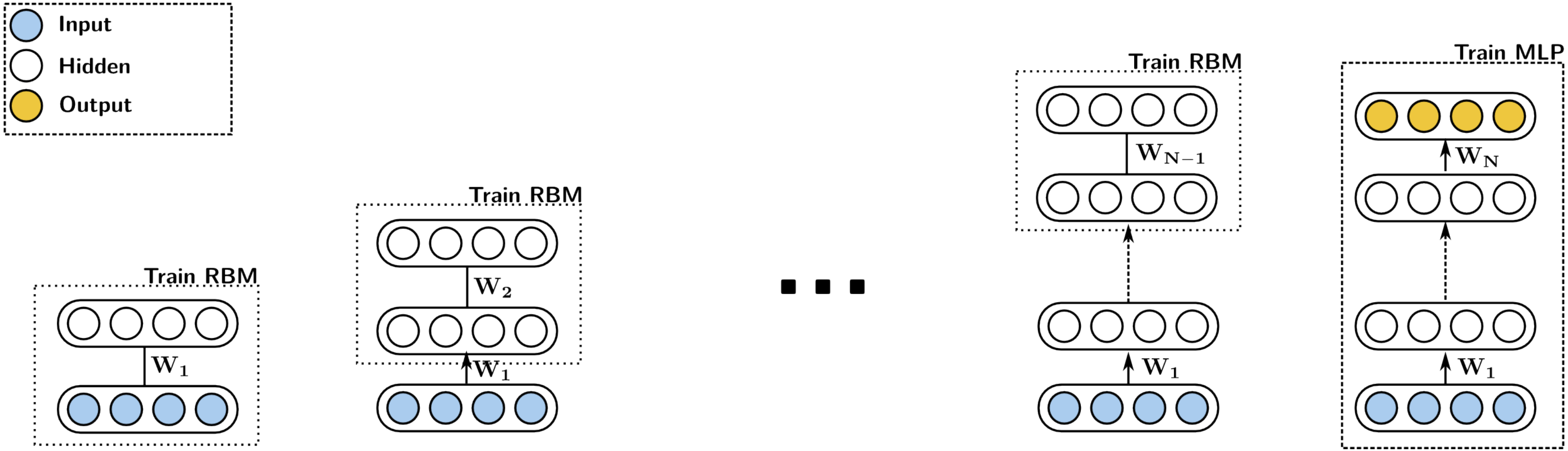
- **Forced alignments** with a bootstrapping system
  → training set of frames with correct state

- Train DNN to classify each frame **among all different states**

- **Stochastic Gradient Descent** with classification costs (e.g. Negative Log-Likelihood, Cross-Entropy)



DNN Training

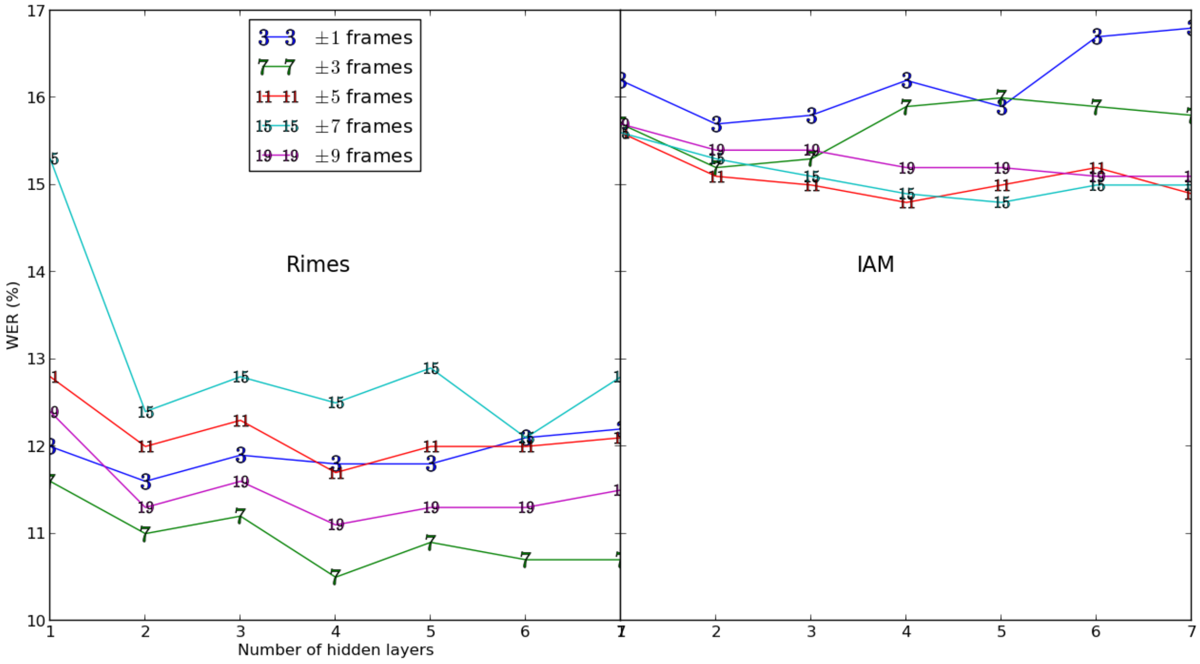


- Weight initialization with 1 epoch of Contrastive Divergence unsupervised training of Restricted Boltzmann Machine, layer by layer (Hinton, 2006)

- Finally, standard supervised training of the whole network (cross-entropy, SGD)

Hinton, G., Osindero, S., & Teh, Y. W. (2006). **A fast learning algorithm for deep belief nets**. *Neural computation*, *18*(7), 1527-1554.

DNN — Depth and Context



Features

Pixels

Depth	Rimes	IAM
1	17.0%	16.7%
2	15.1%	15.5%
3	14.9%	15.4%
4	14.7%	15.4%
5	14.6%	15.7%
6	15.0%	15.4%
7	14.8%	15.6%

DNN Sequence-Discriminative Training

Goal: **improve** nnet response for the **correct state sequence**, while **decreasing the prediction of concurrent hypotheses** (sMBR: state-level Minimum Bayes Risk)

$$\mathcal{F}_{MBR} = \sum_{i \in \mathcal{I}} \log \frac{\sum_W p(O_i|S)^\kappa P(W) A(W, W_i)}{\sum_{W'} p(O_i|S')^\kappa P(W')}$$

- Compute forced alignments
- Extract lattices with unigram LM
- Compute the cost. The accuracy A is 1 if the considered state is in the forced alignments at this position, 0 otherwise
- Gradients are obtained with the forward-backward algorithm

WER (%)		Framewise	+ sMBR
RIMES	Features	14.1	13.5 (-4.2%)
	Pixels	13.6	13.1 (-3.7%)
IAM	Features	12.4	11.7 (-5.6%)
	Pixels	12.4	11.8 (-4.8%)

Kingsbury, Brian. **Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling.** (2009) International Conference on Acoustics, Speech and Signal Processing (ICASSP). 3761-3764

Recurrent Neural Networks

- **Recurrent** = hidden layer at time *t* receives input from previous layer, and from hidden layer at ***t-1***

- May also go through the sequence of inputs backwards
  $\longrightarrow$ **Bidirectional RNNs** (BRNNs)

- Special recurrent units to avoid training problems (vanishing gradient) and learn arbitrarily long dependencies:
  Long Short-Term Memory units (LSTM)
  $\longrightarrow$ BLSTM-RNNs

- Instead of a sequence of features vectors, one may use the image as input
  $\longrightarrow$ Multi-Dimensional (MD)LSTM-RNN

RNNs for Handwriting Recognition

RNNs in **handwritten text recognition contests** :

winner of ICDAR'09 (recognition of French and Arabic words), ICDAR'11 (French text lines), OpenHart'13 (Arabic paragraphs), Maurdor'14 (Multilingual paragraphs), HTRtS'14 (English lines)...

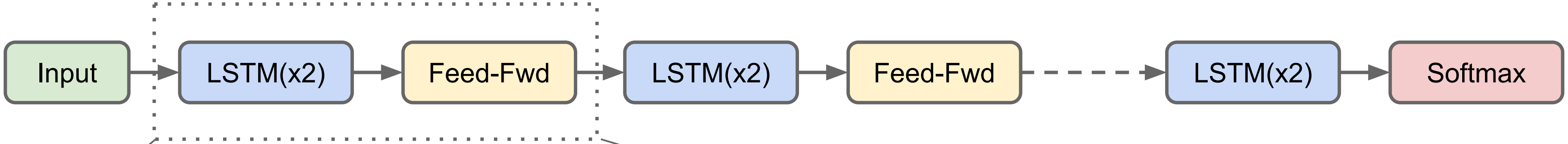
RNNs used :

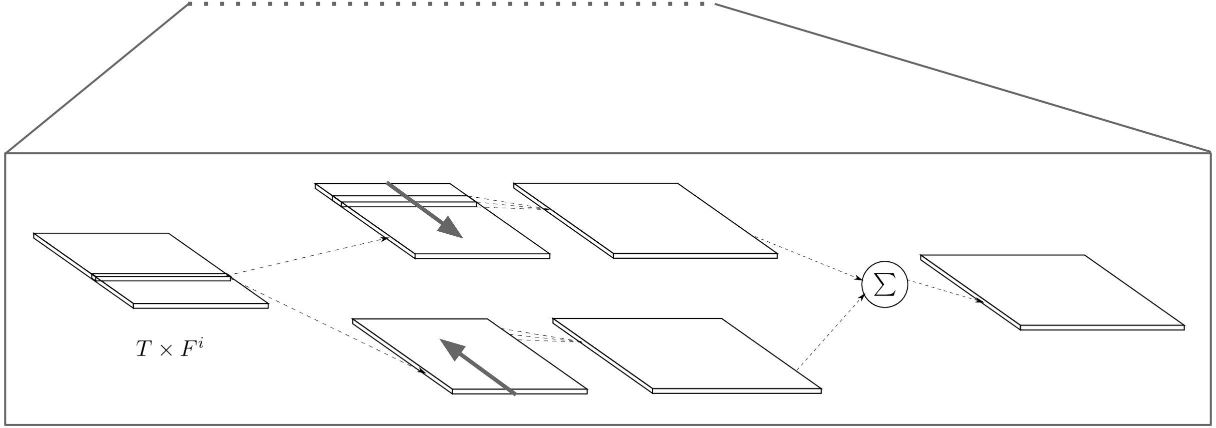
- **MDLSTM-RNNs** : image inputs, *few features* (2x4) at the bottom, increasing number of features along with maps subsampling
- **BLSTM-RNNs** : sequence of feature vector inputs, *few LSTM layers* with many features

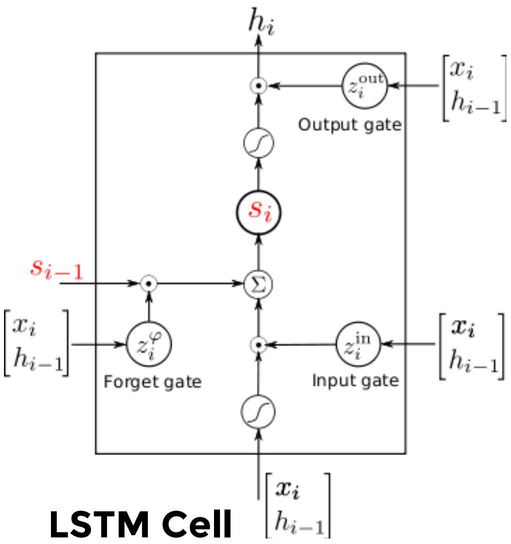
This work :

Deep BLSTM-RNNs with many features

BLSTM-RNNs — Architecture







LSTM Cell

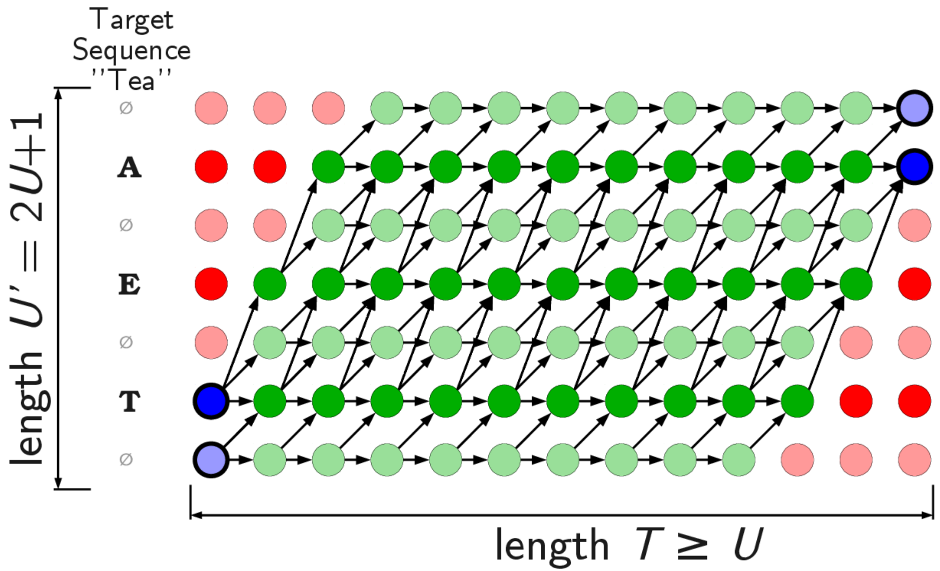
RNNs — CTC training

Connectionist Temporal Classification
(CTC; Graves, 2006)

- RNN has **one output for each character**, plus one *blank (Ø)* output
- *blank* = optional between two successive and different characters, mandatory if the characters are the same

[Ø …] T … [Ø …] E … [Ø …] A … → TEA

- During training, **consider all possible *labelings/segmentations*** of the input sequence
- **Minimize the Negative Log-Likelihood** of the correct label sequence (w/o *blank*)
- Computed efficiently with forward-backward

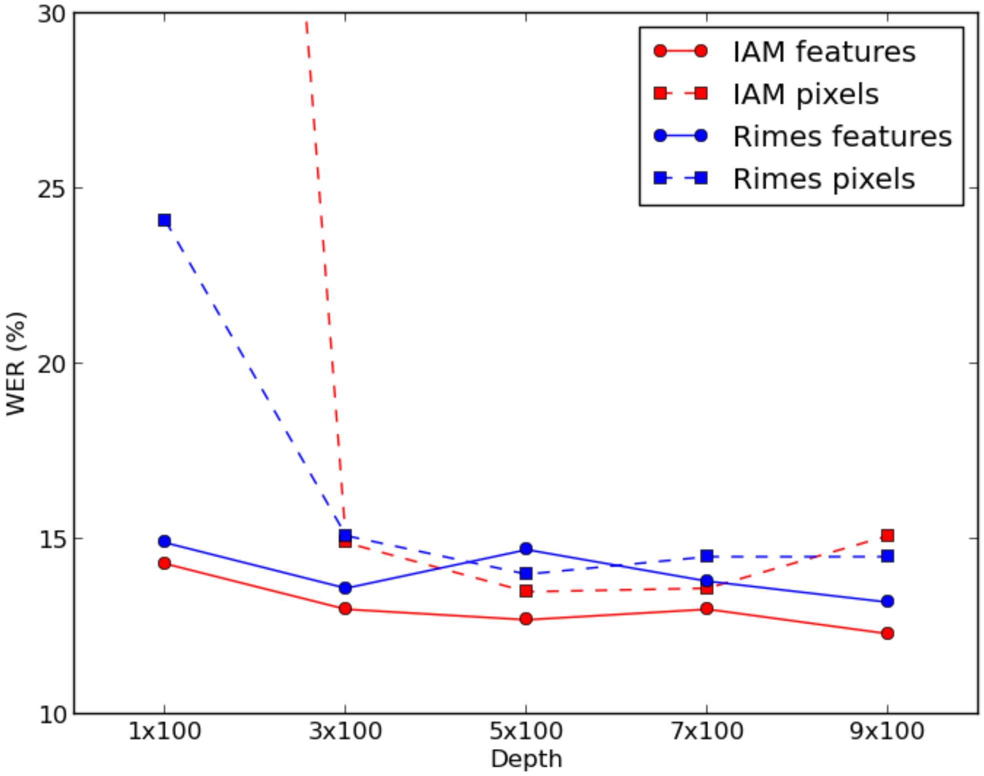


$$-\log p(\mathbf{z}|\mathbf{y})$$

Label truth (here: TEA)

RNN outputs

RNNs — Depth



More than one LSTM hidden layer is generally **better**

For **pixel intensities inputs**:

– one hidden LSTM is largely insufficient

– **many hidden layer** yield RNNs **competitive** with those trained with features

– support the idea that **deep networks learn useful representations** of input in early layers

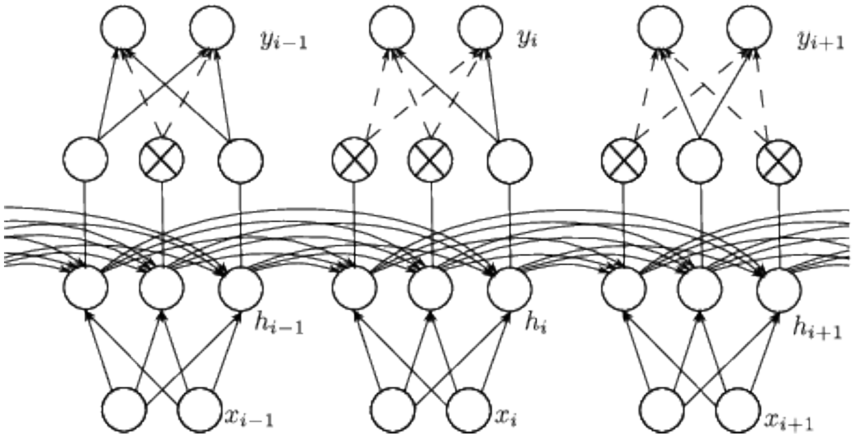
RNNs — Dropout

- Regularization technique – prevents co-adaptation: make units useful on their own, not in combination with outputs of others

- Training: **randomly drop hidden activations with probability *p***
  ≈ sample for 2^N architectures sharing weights

- Decoding: **keep all activations but scale them by *1-p*** to compensate
  ≈ geometric mean of 2^N networks

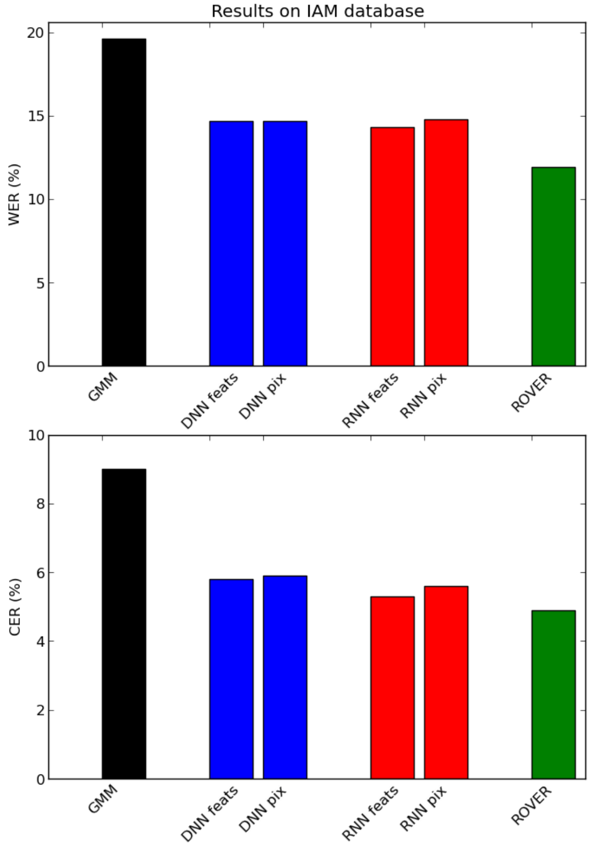
- In RNNs, dropout is applied to the outputs of LSTM units



WER (%)		**7x200**	**+ dropout**
RIMES	**Features**	14.1	12.7 (-9.9%)
	Pixels	14.7	13.6 (-7.5%)
IAM	**Features**	12.9	11.9 (-7.8%)
	Pixels	13.1	11.8 (-9.9%)

Vu Pham, Théodore Bluche, Christopher Kermorvant, Jérôme Louradour (2014) ***Dropout improves recurrent neural networks for handwriting recognition.*** In International Conference on Frontiers in Handwriting Recognition (ICFHR), 285-290.

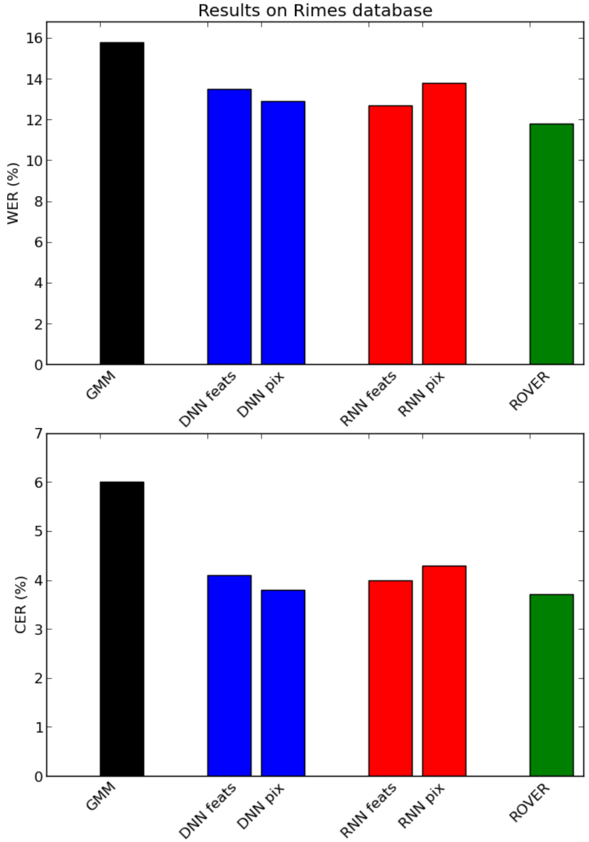
Results — IAM



	WER (%)	CER (%)
GMM-HMM	19.6	9.0
DNN features	14.7	5.8
DNN pixels	14.7	5.9
RNN features	14.3	5.3
RNN pixels	14.8	5.6
ROVER combination	11.9	**4.9**
Doetsch et al., 2014 *	12.2	**4.7**
Kozielski et al., 2013 *	13.3	5.1
Pham et al., 2014	13.6	5.1
Messina et al., 2014 *	19.1	-

* *open-vocabulary*

Results — Rimes



	WER (%)	CER (%)
GMM-HMM	15.8	6.0
DNN features	13.5	4.1
DNN pixels	12.9	3.8
RNN features	12.7	4.0
RNN pixels	13.8	4.3
ROVER combination	**11.8**	**3.7**
Pham et al., 2014	12.3	**3.3**
Doetsch et al., 2014	12.9	4.3
Messina et al., 2014	13.3	-
Kozielski et al., 2013	13.7	4.6

Conclusions

- With **deep** (MLP or recurrent) **neural networks**, features do not seem important
  → mere **pixel intensities yield competitive if not better results**

- Deep MLP are also very good for handwriting recognition
  → RNNs are not the only option

- Even for BLSTM-RNNs, depth improves the final performance
  → Don’t stop at one or two LSTM layers

- Both approaches are complementary → ROVER combination

- Key aspects : **context, depth, sequence-training, dropout, LM on paragraphs**

Future Work

- ➔ *Deep MLPs*
  - dropout, CTC training
- ➔ *Recurrent Neural Networks*
  - sequence-discriminative training
- ➔ Going further...
  - include other types of NN, such as Convolutional Neural Networks, also very popular in Computer Vision, and MDLSTM-RNNs
  - Tandem combination: extract features from NNs

Thank you!

Théodore Bluche
tb@a2ia.com

… and thanks to A.-L. Bernard, V. Pham, J. Louradour for some of the illustrations of this presentation …