Soutenance de thèse pour l'obtention du grade de docteur en informatique de l'Université Paris-Sud

Deep Neural Networks for Large Vocabulary Handwritten Text Recognition

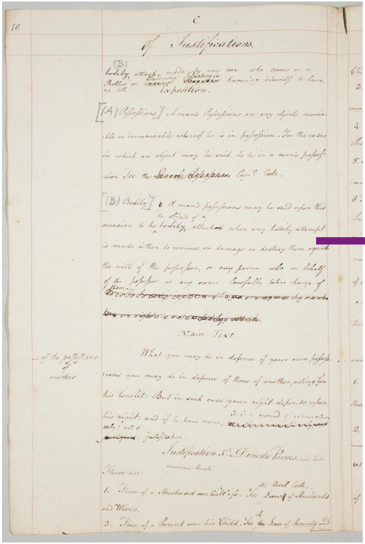
Théodore Bluche

A2iA
Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur
École Doctorale d'Informatique de Paris-Sud

13 mai 2015

Comprendre le monde, construire l'avenir®

What is Handwriting Recognition?



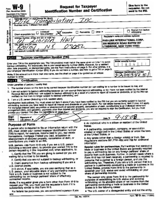
C
10
Of Justifications.
(B)
bodily attack. made by any one who comes as a
Clandestine Destroyer
Robber or Criminal <gap/> knowing himself to have
Exposition.
no title.
[(A) Possessions] A man's Possessions are any objects movea=
:ble or immoveable whereof he is in possession. for the cases
in which an object may be said to be in a man's posses=
:sion See the Law of Possession. Civil Code.
[(B) Bodily] <gap/> A man's possessions may be said upon this
the objects of a
occasion to be bodily attacked when any bodily attempt
is made either to remove or damage or destroy them against
the will of the possessor, or any person who on behalf
of the possessor or any owner lawfully takes charge of
them
.2. The bare signing or accepting a conveyance by one who
has no right is not a bodily attack.
Main - Text .
What you may do in defence of your own possess=
:ions you may do in defence of those of another, acting for
his benefit: But in such case your right depends upon
it is a ground of extenuation
his right : and if he have none, you stand excused only, not
only; not of
justified. justification.
Justification 5th. Domestic Powers. and Sub=
These are :servience thereto.
the Civil Code,
1. Those of a Husband over his Wife. See Laws of Husbands
and Wives.
ib.
2. Those of a Parent over his Child . See the Law of Parents and
Children

from Bentham database (Sánchez et al., 2014)

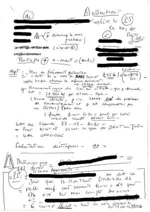
Why do Handwriting Recognition?

Applications:

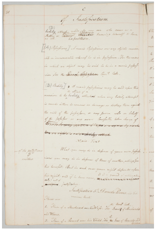
Cheque Processing



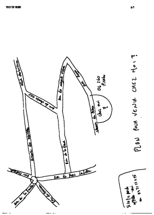
Forms



Mail



Archives



etc...

The recognition result is used for:

- mailroom automation
- tax form processing
- genealogical research

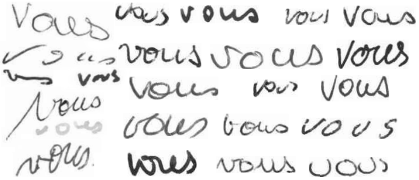
Is it difficult?

- Natural language = hard for computers
- The nature of the input signal adds to the challenge

Is it difficult?

- Natural language = hard for computers
- The nature of the input signal adds to the challenge

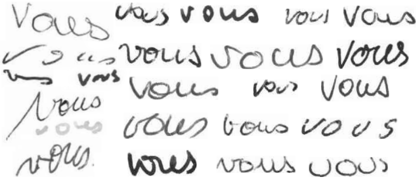
Coping with different writing styles



Is it difficult?

- Natural language = hard for computers
- The nature of the input signal adds to the challenge

Coping with different writing styles

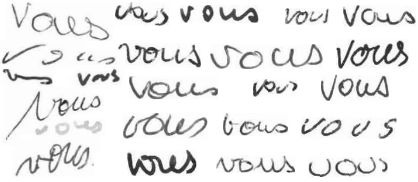


Cursive nature $\Rightarrow$ hard to segment characters before recognition

Is it difficult?

- Natural language = hard for computers
- The nature of the input signal adds to the challenge

Coping with different writing styles

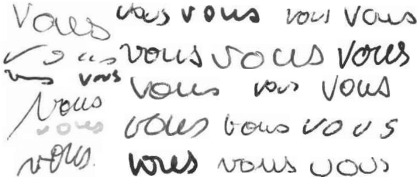


Cursive nature ⇒ hard to segment characters before recognition

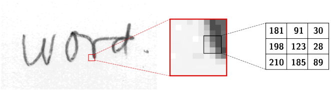
Is it difficult?

- Natural language = hard for computers
- The nature of the input signal adds to the challenge

Coping with different writing styles



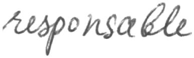
Dealing with digital images



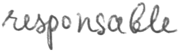
Cursive nature $\Rightarrow$ hard to segment characters before recognition

Preliminary Steps to Handwriting Recognition

Text line image preprocessing:

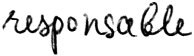


Input image



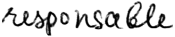
Correct of the inclination of the text

(Buse et al., 1997)



Normalize the contrast of the image

(Roeder, 2009)

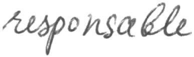


Normalize the size of the image

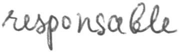
(Toselli et al., 2004)

Preliminary Steps to Handwriting Recognition

Text line image preprocessing:

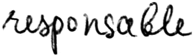


Input image



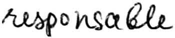
Correct of the inclination of the text

(Buse et al., 1997)



Normalize the contrast of the image

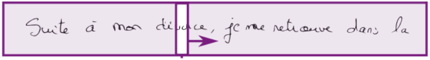
(Roeder, 2009)



Normalize the size of the image

(Toselli et al., 2004)

Feature extraction with a sliding window:



feature vector

$$\mathbf{x_t} = [x_{1,t} \cdots x_{D,t}]$$

$$\mathbf{X} = \mathbf{x_1} \cdots \mathbf{x_T} \xrightarrow{\text{Recognition}} \mathbf{W} = w_1 \cdots w_n$$

(Kaltenmeier et al., 1993)

Recognition

$$\underset{\mathbf{x} = \mathbf{x_1} \cdots \mathbf{x_T}}{\text{Sequence of observations}} \xrightarrow{\text{Optical Model}} \underset{\mathbf{C} = c_1 \cdots c_m}{\text{Sequence of characters}} \xrightarrow{\text{Language Model}} \underset{\mathbf{W} = w_1 \cdots w_n}{\text{Sequence of words}}$$

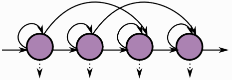
Recognition

Sequence of observations $\mathbf{x} = \mathbf{x_1} \cdots \mathbf{x_T}$ —Optical Model→ Sequence of characters $\mathbf{C} = c_1 \cdots c_m$ —Language Model→ Sequence of words $\mathbf{W} = w_1 \cdots w_n$

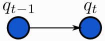
GMM → Sequence of states $\mathbf{q} = q_1 \cdots q_T$ —HMM→

Hidden Markov Models (HMMs)

$c_k \equiv$



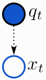
Transition model $P(q_t|q_{t-1})$

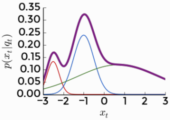


Handles the sequential aspect of the reading task

Emission model $p(x_t|q_t)$

Explains the observations





e.g. in (Bianne-Bernard, 2011; Kozielski et al., 2012, 2014)

Recognition

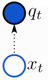
Sequence of observations $\xrightarrow{\text{Optical Model}}$ Sequence of characters $\xrightarrow{\text{Language Model}}$ Sequence of words

$\mathbf{x} = \mathbf{x_1} \cdots \mathbf{x_T}$ $\quad \mathbf{C} = c_1 \cdots c_m$ $\quad \mathbf{W} = w_1 \cdots w_n$

1990's: GMM → Sequence of states $\mathbf{q} = q_1 \cdots q_T$ → HMM

2000's: NN → Sequence of states $\mathbf{q} = q_1 \cdots q_T$ → HMM

2010's



Neural Networks

$P(q_t|x_t)$

- More classical models in pattern recognition
- Predicts the state from the observation

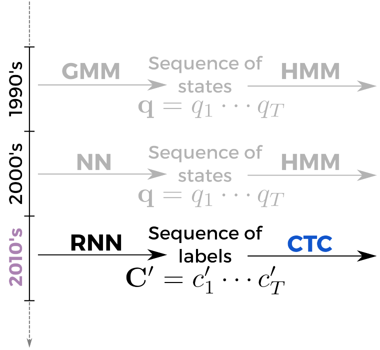
Hybrid NN/HMMs (Bourlard & Morgan, 1994)

e.g. in (Dreuw et al., 2011; Espana-Boquera et al., 2011; Doetsch et al., 2014), with 1-2 hidden layer NNs

Recognition

Sequence of observations $\xrightarrow{\text{Optical Model}}$ Sequence of characters $\xrightarrow{\text{Language Model}}$ Sequence of words

$\mathbf{x} = \mathbf{x_1} \cdots \mathbf{x_T}$ $\qquad$ $\mathbf{C} = c_1 \cdots c_m$ $\qquad$ $\mathbf{W} = w_1 \cdots w_n$



Neural Networks handling the **sequential aspect**

The neural network:

- looks at the whole observation sequence (length T)
- predicts the whole character sequence (length $m \leq T$)

Connectionist Temporal Classification (CTC; Graves et al. (2006))

e.g. in (Strauß et al., 2014; Moysset et al., 2014; Pham et al., 2014)

Recognition

Sequence of observations $\xrightarrow{\text{Optical Model}}$ Sequence of characters $\xrightarrow{\text{Language Model}}$ Sequence of words

$\mathbf{x} = \mathbf{x}_1 \cdots \mathbf{x}_\mathrm{T}$ $\qquad \mathbf{C} = c_1 \cdots c_m$ $\qquad \mathbf{W} = w_1 \cdots w_n$

Lexicon

- Constrain the sequences of characters to form words from a fixed vocabulary

$\mathtt{w} \rightarrow \mathtt{o} \rightarrow \mathtt{r} \rightarrow \mathtt{d} \equiv \mathtt{word}$

Recognition

Sequence of observations $\xrightarrow{\text{Optical Model}}$ Sequence of characters $\xrightarrow{\text{Language Model}}$ Sequence of words

$\mathbf{x} = \mathbf{x}_1 \cdots \mathbf{x}_T$ $\quad$ $\mathbf{C} = c_1 \cdots c_m$ $\quad$ $\mathbf{W} = w_1 \cdots w_n$

Lexicon

- Constrain the sequences of characters to form words from a fixed vocabulary

$\texttt{w} \rightarrow \texttt{o} \rightarrow \texttt{r} \rightarrow \texttt{d} \equiv \texttt{word}$

Language Model

- Constrain the sequences of words

  e.g. to have a high probability $P(\mathbf{W}) = P(w_1, \cdots, w_N)$

- **n-gram** models, estimated from frequencies of sequences of n words in a corpus

Example: `we are ... ?`

$$
\begin{aligned}
P(\texttt{not}|\texttt{are}, \texttt{we}) &= 7.0\% \\
P(\texttt{to}|\texttt{are}, \texttt{we}) &= 4.9\% \\
P(\texttt{in}|\texttt{are}, \texttt{we}) &= 3.0\%
\end{aligned}
$$

... also hybrid word/character language models (Kozielski et al., 2013b; Messina & Kermorvant, 2014)

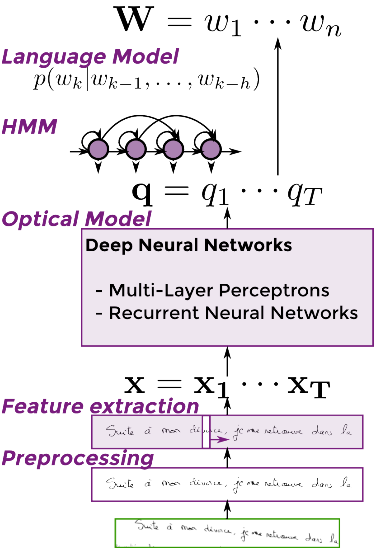
State-of-the-art Handwriting Recognition

- **GMM-HMM** with carefully chosen features and hybrid word/char LM (Kozielski et al., 2013b, 2012, 2014)
- **Tandem RNN/HMM** approach: features for a GMM-HMM extracted with an RNN (Kozielski et al., 2014, 2013a)
- **Hybrid RNN/HMM**: an RNN predicts HMM states (Doetsch et al., 2014)
- **MDRNN+CTC** approach: an RNN predicts character sequences from the whole image (Strauß et al., 2014; Moysset et al., 2014; Pham et al., 2014; Bluche et al., 2014)

Overview

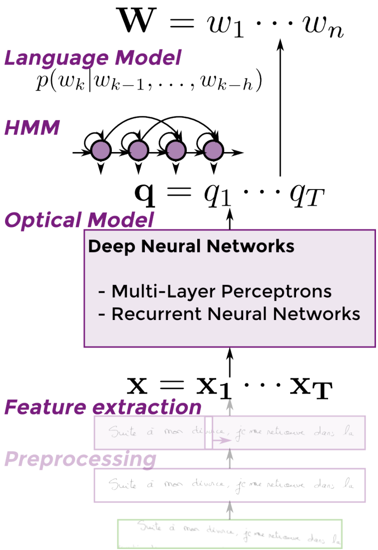
Scope and Contributions

Scope of this Thesis



Hybrid NN/HMM system

Scope of this Thesis



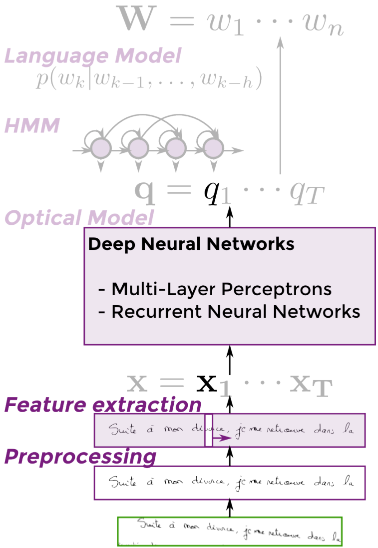
Hybrid NN/HMM system

Similar to...

Speech Recognition

Deep neural networks (MLPs)
Sequence training, ...

Scope of this Thesis



Hybrid NN/HMM system

Similar to...

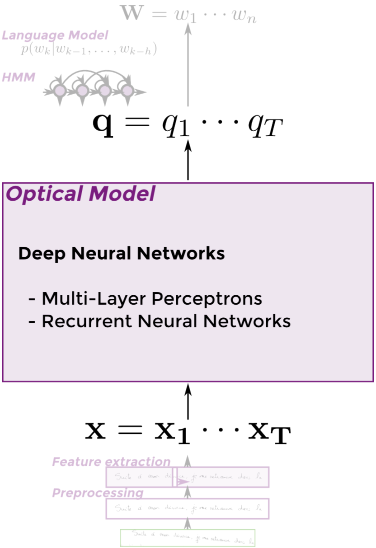
Speech Recognition

Deep neural networks (MLPs)
Sequence training, ...

Object Recognition

Deep neural networks (ConvNNs)
Pixel inputs, ...

Focus of the Work



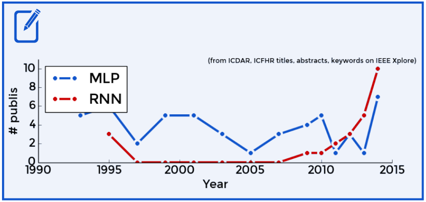
Experimental evaluation of different aspects of

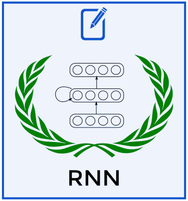
Deep Neural Network Optical Models

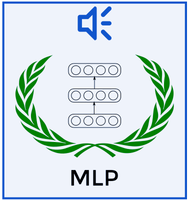
Evaluation

- error rate of the neural network alone (at the frame or character level)
- error rate of the complete system (Neural Network+HMM+LM) : normalized edit distance between output word/char. sequence and reference

Question 1

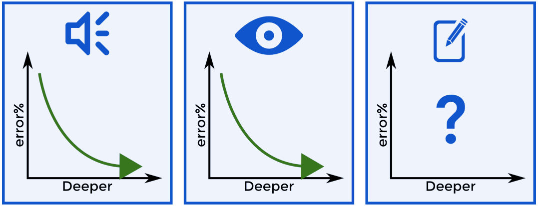






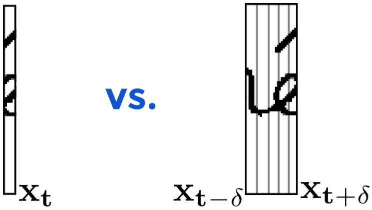
Are RNNs better than deep MLPs?

Question 2



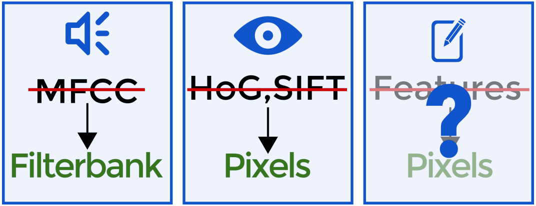
Is deeper better?

Question 3



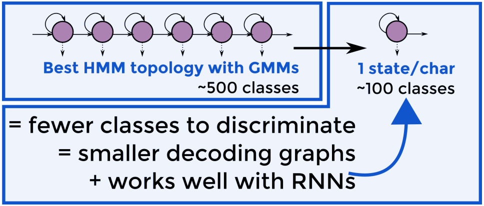
What is the importance of (explicit) input context in MLPs and RNNs?

Question 4



Do we need handcrafted features?

Question 5

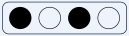


How does the output topology influence the NN performance?

Question 6

Sequence-discriminative training

Dropout



RNN+CTC

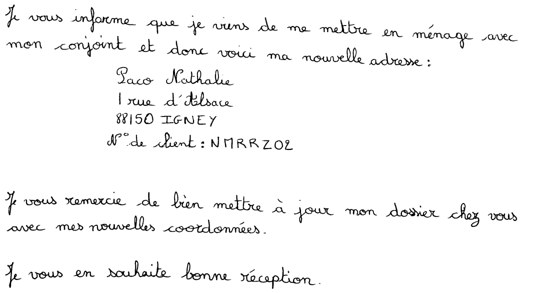
What are the good training strategies for neural networks for handwriting recognition?

Contributions

- State-of-the-art GMM/HMM systems (not presented here)
- Comparison of different **neural network inputs** (type, size of context)
- State-of-the-art continuous handwriting recognition with **deep**, densely connected **neural networks** (MLPs, RNNs) in hybrid NN/HMMs
- Study of **training strategies** of neural network optical models (cross-entropy, CTC, sequence training, dropout)

Experimental Setup

Databases –Rimes

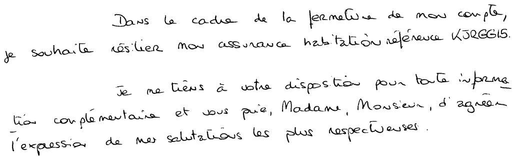


Je vous informe que je viens de me mettre en ménage avec mon conjoint et donc voici ma nouvelle adresse :

Paco Nathalie
1 rue d'Alsace
88150 IGNEY
N° de client : NMRRZ02

Je vous remercie de bien mettre à jour mon dossier chez vous avec mes nouvelles coordonnées.

Je vous en souhaite bonne réception.



Dans le cadre de la fermeture de mon compte, je souhaite résilier mon assurance habitation référence KJRGG15.

Je me tiens à votre disposition pour toute informa-tion complémentaire et vous prie, Madame, Monsieur, d'agréer l'expression de mes salutations les plus respectueuses.

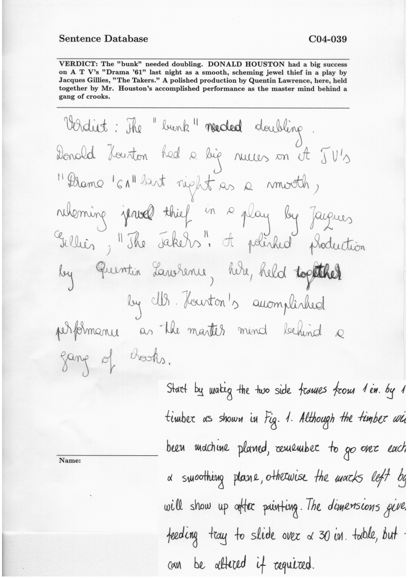
Simulated mail
(imposed scenario)
constrained language
many dates, codes, ...
many writers

French

- 1,600 pages
- ≈ 80,000 words
- 97 different characters

(Augustin et al., 2006)

Databases –IAM


Sentence Database C04-039

VERDICT: The "bunk" needed doubling. DONALD HOUSTON had a big success on A T V's "Drama '61" last night as a smooth, scheming jewel thief in a play by Jacques Gillies, "The Takers." A polished production by Quentin Lawrence, here, held together by Mr. Houston's accomplished performance as the master mind behind a gang of crooks.

Verdict : The "bunk" needed doubling.
Donald Houston had a big success on A TV's
"Drama '61" last night as a smooth,
scheming jewel thief in a play by Jacques
Gillies, "The Takers". A polished production
by Quentin Lawrence, here, held together
by Mr. Houston's accomplished
performance as the master mind behind a
gang of crooks.

Start by making the two side frames from 1 in. by 1
timber as shown in Fig. 1. Although the timber wi
been machine planed, remember to go over each
a smoothing plane, otherwise the marks left by
will show up after painting. The dimensions give
feeding tray to slide over a 30 in. table, but
can be altered if required.

Name:

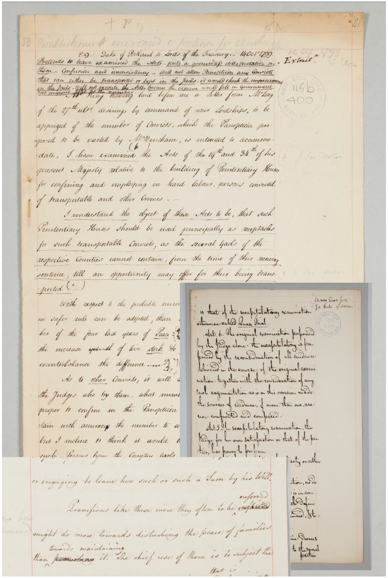
Copied passages of litterature
pretty clean handwriting
controlled content
many writers
rich language

English

- 1,200 pages
- ≈ 90,000 words
- 79 different characters

(Marti & Bunke, 2002)

Databases –Bentham



Historical documents (19th century): notes of the philosopher Jeremy Bentham

one author, a few writers

difficult handwriting

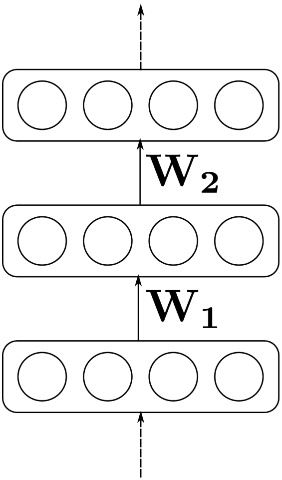
hyphenation, crossed-out text

English

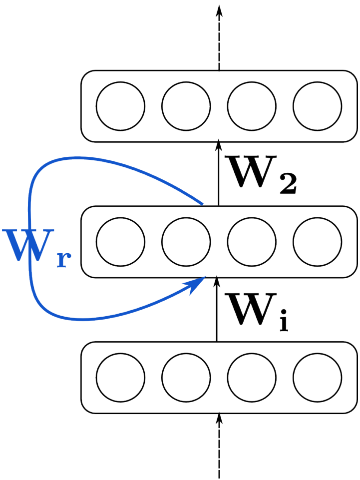
- 433 pages
- ≈ 95,000 words
- 93 different characters

(Sánchez et al., 2014)

Artificial Neural Networks

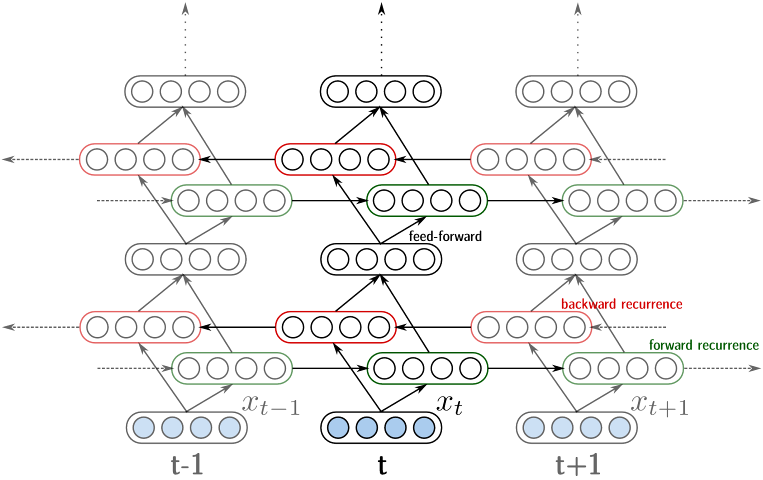


Multi-Layer Perceptron (MLP)



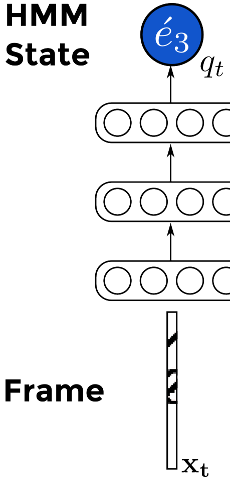
Recurrent Neural Network (RNN)

Artificial Neural Networks



Bidirectional RNN

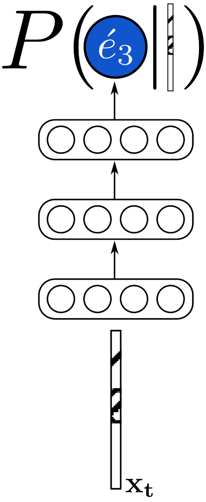
Neural Networks for Classification



- The **outputs** of the network are the different classes (HMM states, characters), and represent a score for each of them

- The **inputs** of the network are the frames extracted with the sliding window (or rather the resulting features)

Neural Networks for Classification

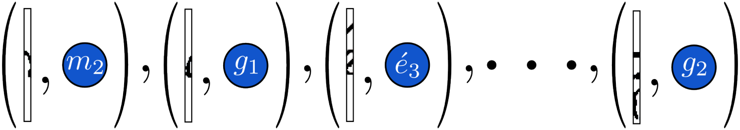


- NN outputs can be considered as **posterior probabilities**
- **Hybrid NN/HMM** Framework **(Bourlard & Morgan, 1994)**

$$p(x_t|q_t) \propto \frac{P(q_t|x_t)}{P(q_t)}$$

Framewise Cross-Entropy Training

1. Compute the forced alignments of the frame sequence with the HMM of the correct word sequence
   $\longrightarrow$ labeled dataset of frames $\mathcal{S} = \{(x_t, q_t)\}$



2. Train the network to classify each frame individually

Cross-entropy cost function:

$$E_{xent} = - \sum_{(x_t, q_t) \in \mathcal{S}} \log P(q_t | x_t)$$

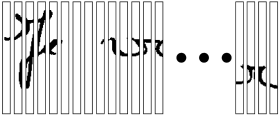
Evaluation

Frame Error Rate (FER%)

$$\frac{\text{\# incorrectly classified frames}}{\text{\# of frames}}$$

Connectionnist Temporal Classification Training (CTC)

❶ Use the dataset of frame sequence, with character sequence targets $\mathcal{S} = \{(\mathbf{x}, \mathbf{c})\}$

( , J e _ v o … r), • • •

❷ Train to predict the character sequence $\mathbf{c}$ directly

- NN outputs = characters + $\oslash$
- Mapping $\mathcal{B}$: $a\ a \oslash \oslash b\ b \oslash b\ a \mapsto abba$

CTC cost function:

$$E_{ctc} = -\sum_{(\mathbf{x},\mathbf{c})\in\mathcal{S}} \log P(\mathbf{c}|\mathbf{x})$$

with

$$P(\mathbf{c}|\mathbf{x}) = \sum_{\mathbf{q}\in\mathcal{B}^{-1}(\mathbf{c})} P(\mathbf{q}|\mathbf{x}) = \sum_{\mathbf{q}\in\mathcal{B}^{-1}(\mathbf{c})} \prod_t P(q_t|\mathbf{x})$$

Evaluation

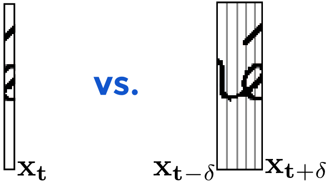
NN - Character Error Rate (NN-CER%)

$$\frac{\text{edit distance between reference and recognition}}{\text{\# of reference characters}}$$

(Graves et al., 2006)

Hybrid Deep Neural Networks / HMMs

Inputs

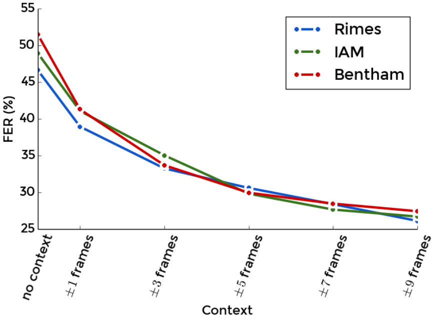


Handcrafted features
56 geometrical and statistical features from (Bianne-Bernard, 2011)

Pixel Values
640 gray-level pixel intensities

q3: (What) do we gain by explicitly including context?

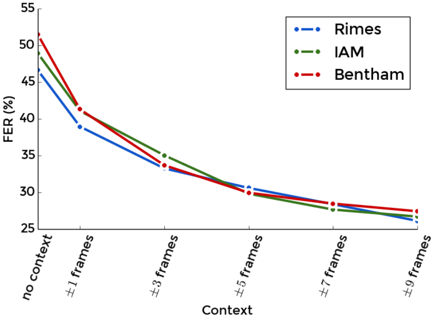
in MLPs... (Neural Network alone (FER%))



(Handcrafted features)

q3: (What) do we gain by explicitly including context?

in MLPs... (Neural Network alone (FER%))

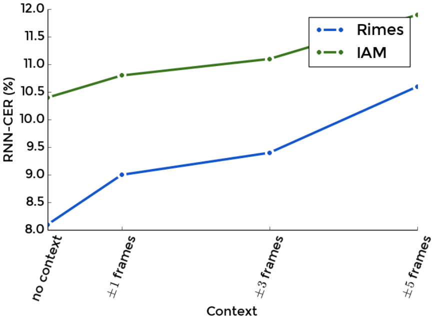


the improvements are not so clear in the complete systems including LM, but ...
$\longrightarrow$ **2.4-22% relative WER improvement** with best amount of context

(Handcrafted features)

q3: (What) do we gain by explicitly including context?

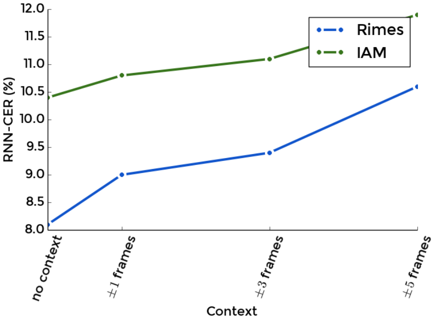
in RNNs... (Neural Network alone (RNN-CER%))



(Handcrafted features)

q3: (What) do we gain by explicitly including context?

in RNNs... (Neural Network alone (RNN-CER%))



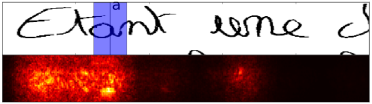
$\longrightarrow$ explicitly including context **increases the error rate**

(Handcrafted features)

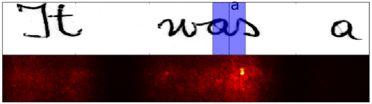
What context RNNs learn?

- Visulalization: gradient of the output w.r.t. the inputs (Graves et al., 2013)
- Top: input image, sliding window and prediction at time t
- Bottom: gradient of the prediction w.r.t the inputs

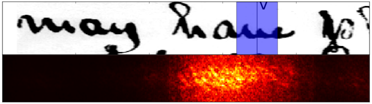
Rimes



IAM



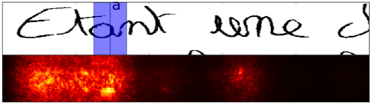
Bentham



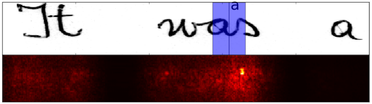
What context RNNs learn?

- Visulalization: gradient of the output w.r.t. the inputs (Graves et al., 2013)
- Top: input image, sliding window and prediction at time t
- Bottom: gradient of the prediction w.r.t the inputs

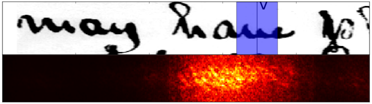
Rimes



IAM



Bentham

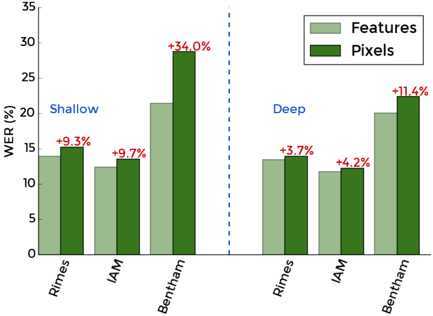


$\longrightarrow$ RNNs automatically use the context, which can even **extend beyond character boundaries**

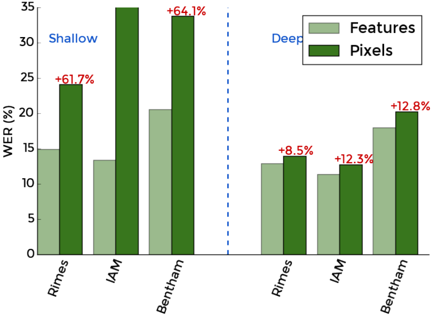
q4: Are pixel values sufficient?

Complete systems (with LM; WER%)

MLPs



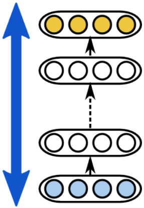
RNNs



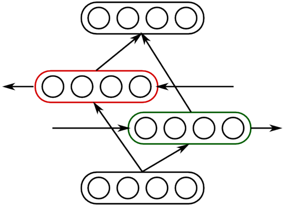
⟶ Deep NNs **reduce the performance gap** between features and pixels

Architecture

Depth



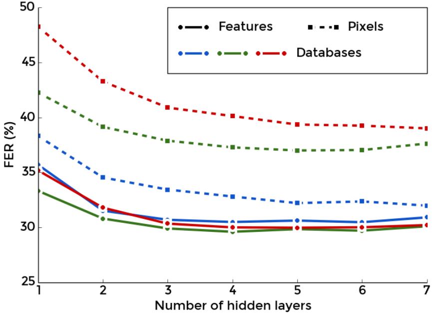
Recurrence



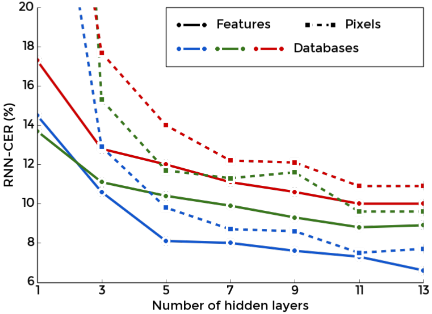
q2: Is deeper better?

Neural networks alone

MLPs



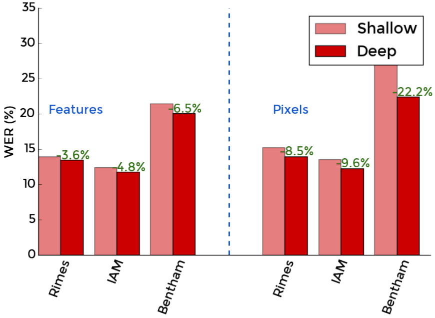
RNNs



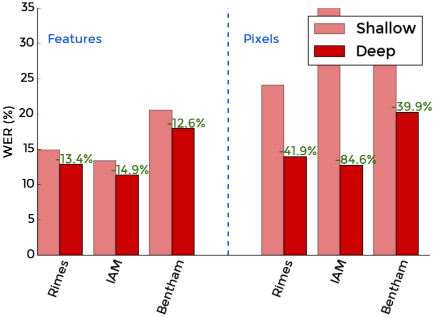
q2: Is deeper better?

Complete systems (with LM; WER%)

MLPs

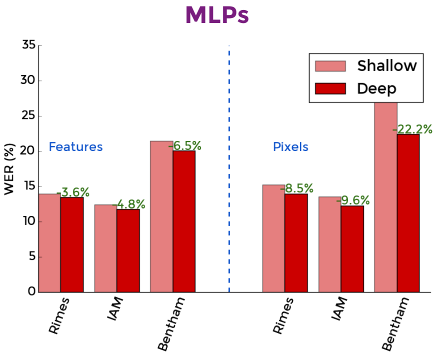


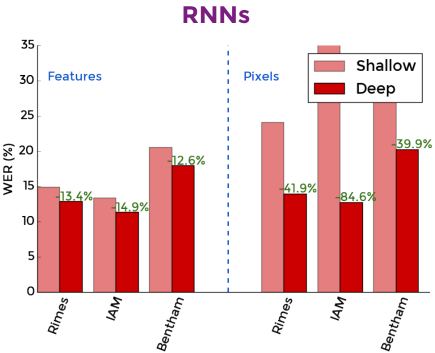
RNNs



q2: Is deeper better?

Complete systems (with LM; WER%)

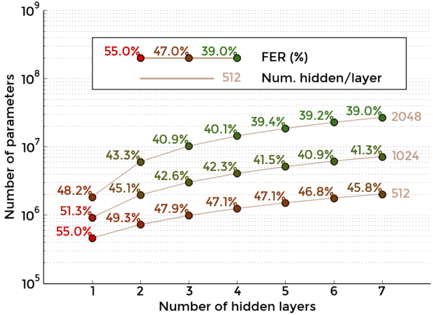




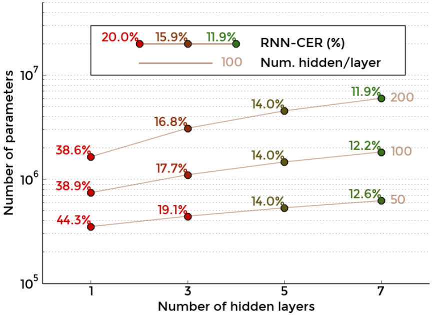
⟶ **Significant improvements (4-40%) with deep NNs** (more for RNNs, and more for pixels)

What is the effect of depth vs. number of parameters?

MLPs



RNNs

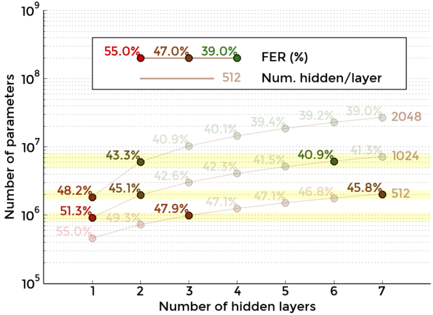


⟶ results improve with both increasing depth and number of parameters

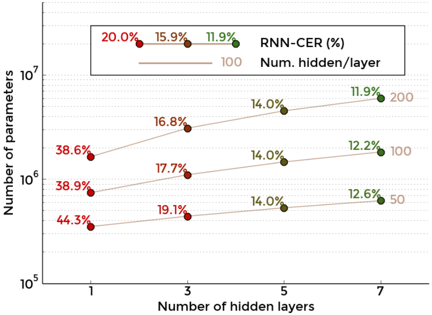
(Bentham Database - Pixels)

What is the effect of depth vs. number of parameters?

MLPs



RNNs

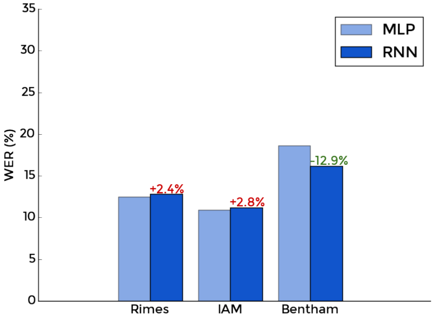


⟶ at constant number of parameters, **deeper is better**

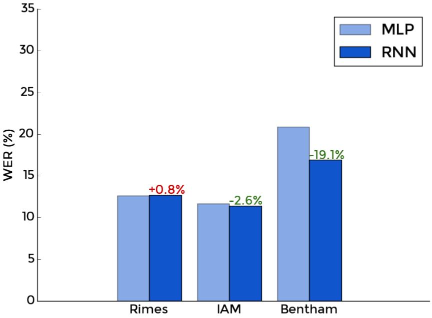
(Bentham Database - Pixels)

q1: How deep MLPs compare to deep RNNs?

Complete systems (with LM; WER%)



Features

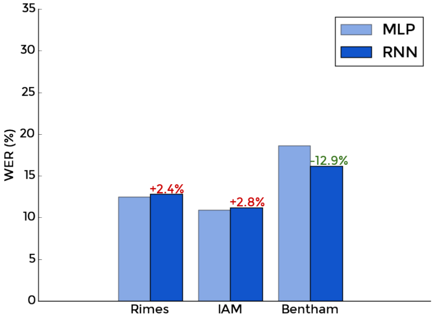


Pixels

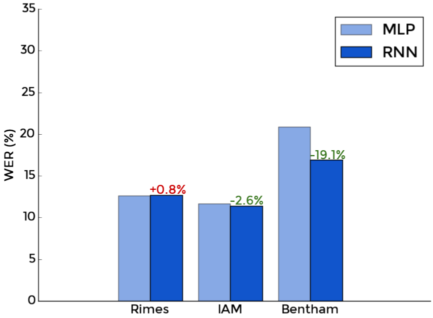
(Cross-entropy training for MLP - CTC training for RNN - with LM)

q1: How deep MLPs compare to deep RNNs?

Complete systems (with LM; WER%)



Features



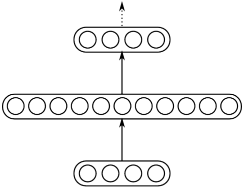
Pixels

$\longrightarrow$ MLPs can achieve **competitive performance to RNNs** (Rimes, IAM) but with limited amount of time, easier to train RNNs (Bentham)

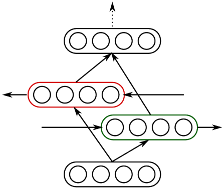
(Cross-entropy training for MLP - CTC training for RNN - with LM)

What is the impact of recurrence?

With "RNN" architecture: no input context, CTC training, alternating recurrent and feed-forward layers. Switching recurrent (**R**) to feed-forward (**F**) layers.



F



R

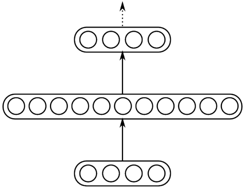
Effect of recurrence on the character error rate of the RNN alone (RNN-CER%)

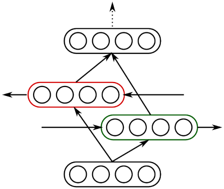
	Features		Pixels	
	Rimes	IAM	Rimes	IAM
FFF	44.0	39.6	38.0	32.8

(CTC training - 5 hidden layers = 3 "blocks" F/**R**)

What is the impact of recurrence?

With "RNN" architecture: no input context, CTC training, alternating recurrent and feed-forward layers. Switching recurrent (**R**) to feed-forward (**F**) layers.





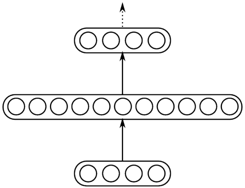
Effect of recurrence on the character error rate of the RNN alone (RNN-CER%)

	Features		Pixels	
	Rimes	IAM	Rimes	IAM
FFF	44.0	39.6	38.0	32.8
RFF	13.2	13.7	62.2	61.3
F**R**F	**12.3**	13.7	20.6	19.2
FF**R**	13.0	**12.5**	**17.5**	**17.5**

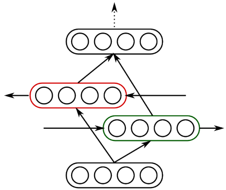
(CTC training - 5 hidden layers = 3 "blocks" F/**R**)

What is the impact of recurrence?

With "RNN" architecture: no input context, CTC training, alternating recurrent and feed-forward layers. Switching recurrent (**R**) to feed-forward (**F**) layers.



F



R

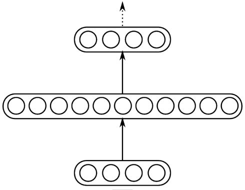
Effect of recurrence on the character error rate of the RNN alone (RNN-CER%)

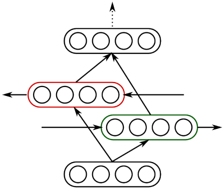
	Features		Pixels	
	Rimes	IAM	Rimes	IAM
FFF	44.0	39.6	38.0	32.8
RFF	13.2	13.7	62.2	61.3
FRF	12.3	13.7	20.6	19.2
FFR	13.0	12.5	17.5	17.5
RRF	11.6	23.1	20.8	20.3
RFR	11.6	11.8	23.0	19.6
FRR	11.6	12.0	15.3	17.5

(CTC training - 5 hidden layers = 3 "blocks" F/R)

What is the impact of recurrence?

With "RNN" architecture: no input context, CTC training, alternating recurrent and feed-forward layers. Switching recurrent (**R**) to feed-forward (**F**) layers.





Effect of recurrence on the character error rate of the RNN alone (RNN-CER%)

	Features		Pixels	
	Rimes	IAM	Rimes	IAM
FFF	44.0	39.6	38.0	32.8
RFF	13.2	13.7	62.2	61.3
F**R**F	**12.3**	13.7	20.6	19.2
FF**R**	13.0	**12.5**	**17.5**	**17.5**
RRF	**11.6**	23.1	20.8	20.3
RF**R**	**11.6**	**11.8**	23.0	19.6
F**RR**	**11.6**	12.0	**15.3**	**17.5**
RRR	**9.7**	**11.4**	16.7	18.9

(CTC training - 5 hidden layers = 3 "blocks" F/**R**)

Training

- Sequence-discriminative training
- Dropout
- RNN+CTC

q6.1: What improvement do we observe with sequence discriminative training of MLPs?

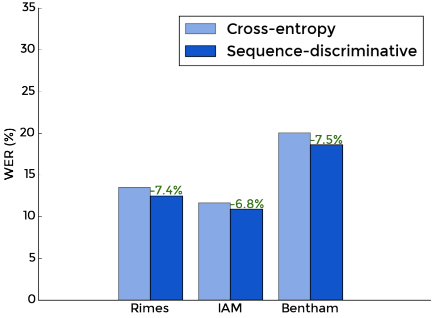
- Goal: optimize NN in the context of the whole system (max. $P(\mathbf{W}|\mathbf{x})$, or min. error rate)
- Involves a sum over all possible word sequences $\rightarrow$ in practice, computed in recognition lattices

State-Level Minimum Bayes Risk (sMBR; Kingsbury (2009)), maximize:

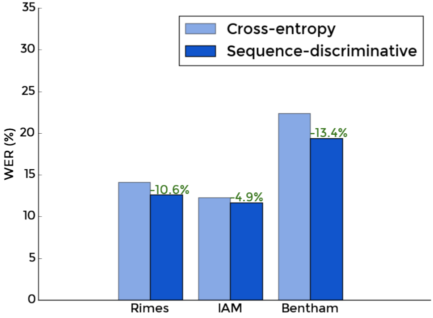
$$E_{sMBR} = \sum_{(\mathbf{x},\mathbf{W}_{ref})\in\mathcal{S}} \frac{\sum_{\mathbf{W}} p(\mathbf{x}|\mathbf{W})P(\mathbf{W})A(\mathbf{W},\mathbf{W}_{ref})}{\sum_{\mathbf{W}'} p(\mathbf{x}|\mathbf{W}')P(\mathbf{W}')}$$

q6.1: What improvement do we observe with sequence discriminative training of MLPs?

Complete systems (with LM; WER%)



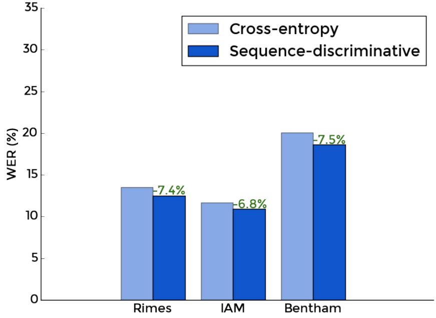
Features



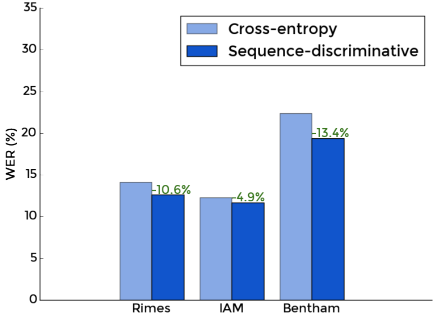
Pixels

q6.1: What improvement do we observe with sequence discriminative training of MLPs?

Complete systems (with LM; WER%)



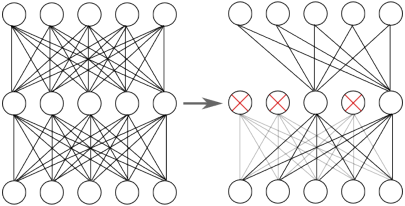
Features



Pixels

$\longrightarrow$ **5-13% relative WER improvement**: consistent with what we observe in speech recognition

Dropout



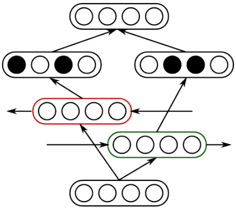
- Regularization technique for big NNs that tend to overfit
- During training, randomly drop neurons in a layer with probability p
- At test time, keep all neurons but multiply outgoing weights by $(1 - p)$
- Applied to MDRNN in (Pham et al., 2014)

(Hinton et al., 2012)

q6.2: What is the impact of the position of dropout in RNNs?

... compared to RNNs without any regularization

Position relative to the recurrent layers:

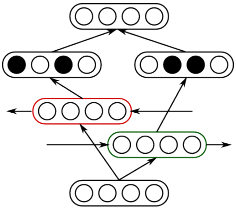


After
(Pham et al., 2014)

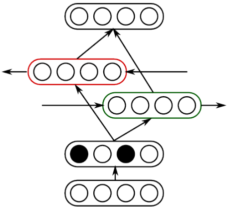
q6.2: What is the impact of the position of dropout in RNNs?

... compared to RNNs without any regularization

Position relative to the recurrent layers:



After
(Pham et al., 2014)

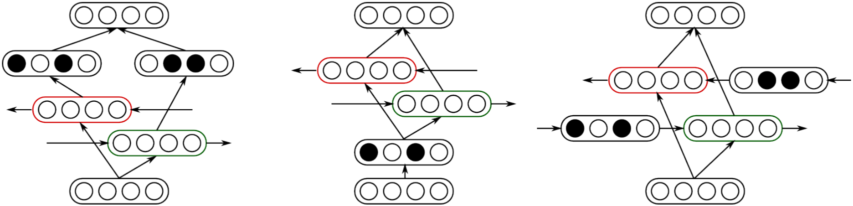


Before

q6.2: What is the impact of the position of dropout in RNNs?

... compared to RNNs without any regularization

Position relative to the recurrent layers:

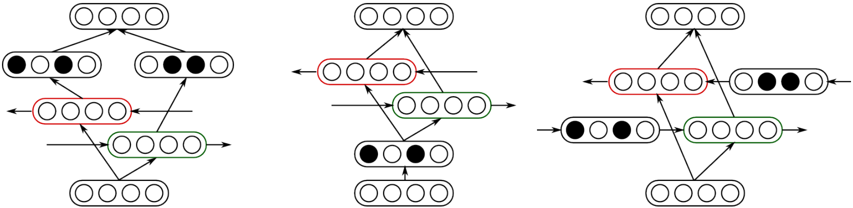


After (Pham et al., 2014) | **Before** | **Inside**

q6.2: What is the impact of the position of dropout in RNNs?

... compared to RNNs without any regularization

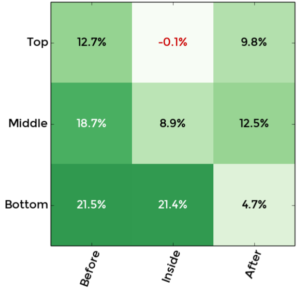
Position relative to the recurrent layers:

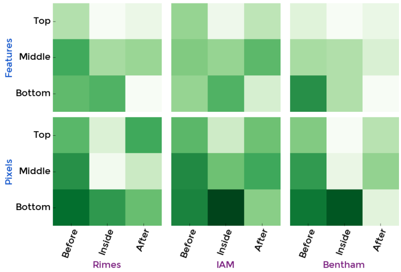


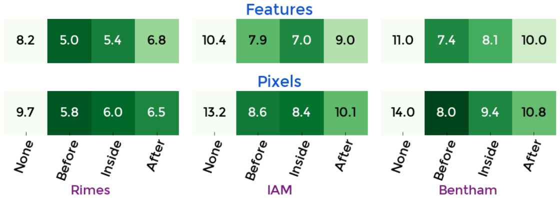
After (Pham et al., 2014) | Before | Inside

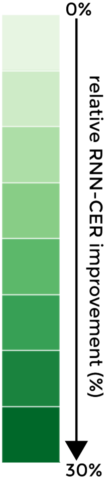
Postion inside the network: **bottom**, **middle**, or **top** recurrent layer.

What is the impact of the position of dropout in RNNs?







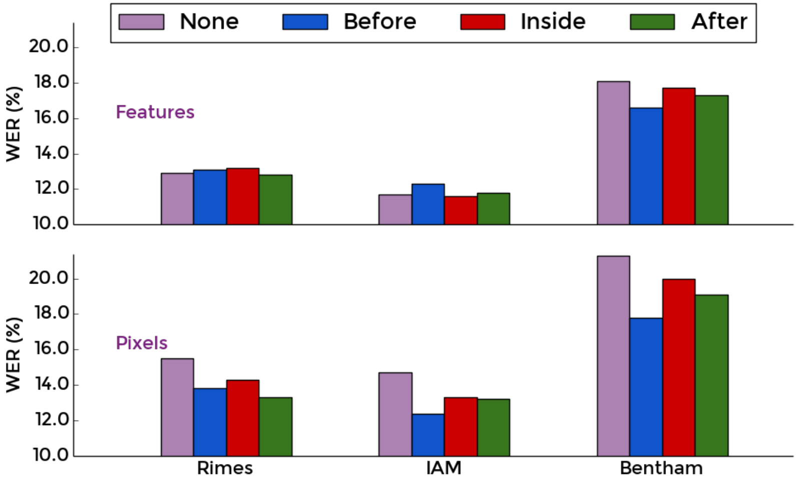


⟶ improvements over the method of (Pham et al., 2014)

(CTC training - RNN alone (5 hidden layers of 200 nodes))

Where to apply dropout in RNNs?

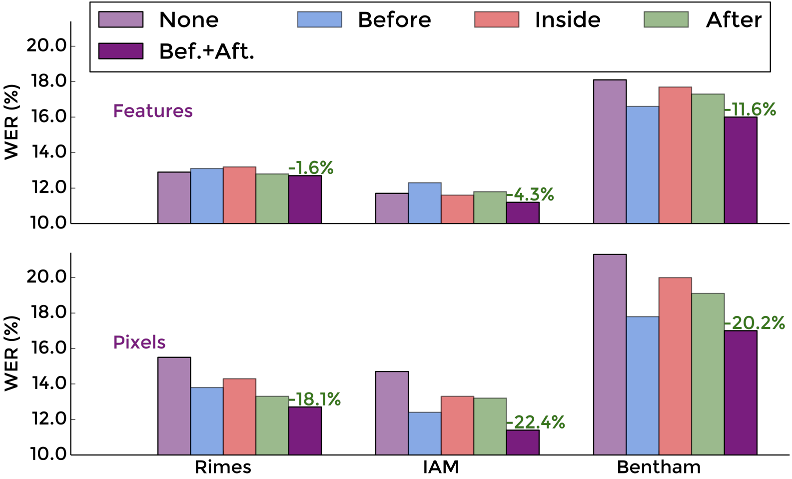
Complete systems (with LM; WER%)



(CTC training - RNN+LM (5 hidden layers of 200 nodes))

Where to apply dropout in RNNs?

Complete systems (with LM; WER%)



(CTC training - RNN+LM (5 hidden layers of 200 nodes))

q5-6: What is the impact of the outputs and training stategies?

	Training cost	Outputs
Framewise cross-entropy (MLPs)	$-\log \prod_t P(q_t \| x_t)$	HMM states (5-6 / character)
CTC (RNNs) (Graves et al., 2006)	$-\log \sum_{\mathbf{q}} \prod_t P(q_t \| \mathbf{x})$	Characters and blank label $\oslash$

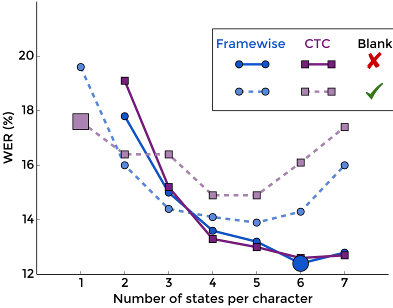
q5-6: What is the impact of the outputs and training stategies?

	Training cost	Outputs
Framewise cross-entropy (MLPs)	$-\log \prod_t P(q_t \mid x_t)$	HMM states (5-6 / character)
CTC (RNNs) (Graves et al., 2006)	$-\log \sum_{\mathbf{q}} \prod_t P(q_t \mid \mathbf{x})$	Characters and blank label $\oslash$
HMM training (NN/HMM) (Hennebert et al., 1997)	$-\log \sum_{\mathbf{q}} \prod_t \frac{P(q_t \mid x_t)}{P(q_t)} P(q_t \mid q_{t-1})$	HMM states (5-6 / character)

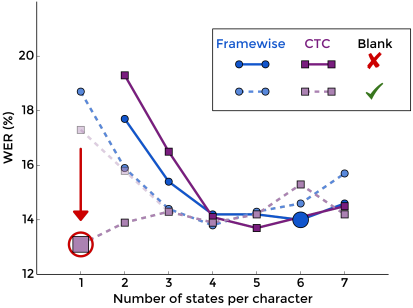
q5-6: What is the impact of the outputs and training stategies?

Complete systems (with LM; WER%)

MLPs



RNNs



⟶ CTC works well with RNNs, not so much with MLPs

⟶ Summation aspect does not improve the results, except for RNN+blank

⟶ The blank symbol only helps with a few states

⟶ CTC+blank, with one-state models, is especially suited to RNNs

(MLP: 2x1024, ± 5 frames - RNN: 1x100))

Results

Final Results –Rimes database

Final results on Rimes database

		WER%	CER%
GMM-HMM	Features	15.8	6.0
MLP	Features	12.7	3.7
	Pixels	12.4	3.9
RNN	Features	12.6	3.9
	Pixels	13.8	4.6
	Combination	11.2	3.5
	Pham et al. (2014)	12.3	3.3
RWTH AACHEN UNIVERSITY	Doetsch et al. (2014)	12.9	4.3
	Messina & Kermorvant (2014)	13.3	-
RWTH AACHEN UNIVERSITY	Kozielski et al. (2013a)	13.7	4.6
	Messina & Kermorvant (2014)	14.6	-
	Menasri et al. (2012)	15.2	7.2

Final Results –IAM database

Final results on IAM database

		WER%	CER%
GMM-HMM	Features	19.6	9.0
MLP	Features	13.3	5.4
	Pixels	13.8	5.6
RNN	Features	13.2	5.0
	Pixels	14.4	5.7
	Combination	10.9	4.4
RWTH AACHEN UNIVERSITY	Doetsch et al. (2014) *	12.2	4.7
RWTH AACHEN UNIVERSITY	Kozielski et al. (2013a) *	13.3	5.1
	Pham et al. (2014)	13.6	5.1
	Messina & Kermorvant (2014) *	19.1	-
	Espana-Boquera et al. (2011)	22.4	9.8

* : open-vocabulary

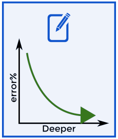
Final Results –Bentham database

Final results on Bentham database

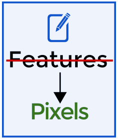
		WER%	CER%
MLP	Features	18.6	7.5
	Pixels	20.9	8.2
RNN	Features	16.2	5.4
	Pixels	16.9	5.9
	Combination	14.1	5.0
	CITlab	14.6	-
	Ours (Competition)	15.1	-

Conclusions and Perspectives

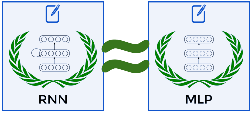
Conclusions



Deeper is better

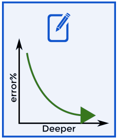


Use pixels with deep neural networks

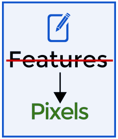


RNNs are not the only solution

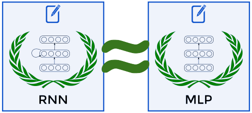
Conclusions



Deeper is better



Use pixels with deep neural networks



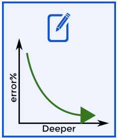
RNNs are not the only solution

RNN+CTC

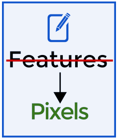
MLP+HMM

... although with RNNs: no need to tune context size or HMM topology, use CTC and no bootstrapping system

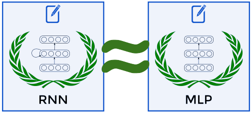
Conclusions



Deeper is better



Use pixels with deep neural networks



RNNs are not the only solution

RNN+CTC

MLP+HMM

... although with RNNs: no need to tune context size or HMM topology, use CTC and no bootstrapping system

... and don't forget

Sequence-discriminative training

Dropout

...

Perspectives

Focus on **other components** of the systems:

- inputs: sliding window vs. whole images and ConvNNs/MDLSTM-RNNs
- Word language models and fixed vocabularies seem to be a limitation
  $\longrightarrow$ e.g. hybrid word/char LMs (Kozielski et al., 2013b; Messina & Kermorvant, 2014)

For **industrial applications** ...

- less training data, less clean
- no line segmentation
- no transcript
- smaller models

Thank you for your attention!

tb@a2ia.com

Publications I

Bluche, T., Louradour, J., Knibbe, M., Moysset, B., Benzeghiba, M. F., & Kermorvant, C. (2014a). The A2iA Arabic Handwritten Text Recognition System at the Open HaRT2013 Evaluation. In 11th IAPR International Workshop on Document Analysis Systems (DAS), (pp. 161--165). IEEE.

Bluche, T., Moysset, B., & Kermorvant, C. (2014b). Automatic Line Segmentation and Ground-Truth Alignment of Handwritten Documents. In 14th International Conference on Frontiers in Handwriting Recognition (ICFHR2014), (pp. 667--672).

Bluche, T., Ney, H., & Kermorvant, C. (2013a). Feature Extraction with Convolutional Neural Networks for Handwritten Word Recognition. In 12th International Conference on Document Analysis and Recognition (ICDAR), (pp. 285--289). IEEE.

Bluche, T., Ney, H., & Kermorvant, C. (2013b). Tandem HMM with convolutional neural network for handwritten word recognition. In 17th International Conference on Acoustics, Speech and Signal Processing (ICASSP), (pp. 2390--2394). IEEE.

Bluche, T., Ney, H., & Kermorvant, C. (2014c). A Comparison of Sequence-Trained Deep Neural Networks and Recurrent Neural Networks Optical Modeling for Handwriting Recognition. In International Conference on Statistical Language and Speech Processing, (pp. 199--210).

Kermorvant, C., Bianne-Bernard, A.-L., Bluche, T., & Louradour, J. (2012). On using alternative recognition candidates and scores for handwritten documents classification. Tech. Rep. A2iA-RR-2012-1, A2iA.

Louradour, J., Bluche, T., Bianne-Bernard, A.-L., Menasri, F., & Kermorvant, C. (2012). De l'usage des scores et des alternatives de reconnaissance pour la classification d'images de documents manuscrits. In Colloque International Francophone sur l'Ecrit et le Document (CIFED).

Moysset, B., Bluche, T., Knibbe, M., Benzeghiba, M. F., Messina, R., Louradour, J., & Kermorvant, C. (2014). The A2iA Multi-lingual Text Recognition System at the second Maurdor Evaluation. In 14th International Conference on Frontiers in Handwriting Recognition (ICFHR2014), (pp. 297--302).

Pham, V., Bluche, T., Kermorvant, C., & Louradour, J. (2014). Dropout improves recurrent neural networks for handwriting recognition. In 14th International Conference on Frontiers in Handwriting Recognition (ICFHR2014), (pp. 285--290).

Stutzmann, D., Bluche, T., Lavrentev, A., Leydier, Y., & Kermorvant, C. (2015). From Text and Image to Historical Resource: Text-Image Alignment for Digital Humanists. In Digital Humanities (DH) -- to appear.

Go to...

Back to start...

Main – **Intro** • **Setup** • **Inputs** • **Archi** • **Training** • **Results** • **Conclusion** • **References**

Introduction – **What is HWR?** • **Approach** • **Scope** • **DBs** • DBs (figures) • **NNs** • **Training** • Base System • Base System (optim)

Inputs – **Input context** • Context WER • Context in RNNs • **Pixels vs Feats**

Architecture – **Depth** • **MLP vs RNN** • NN archis • NN training • **Recurrence** • MLP filters • RNN filters • Depth vs Params

Training – **Seq. Training** • sMBR training • sMBR training (Results) • **Dropout** • **Framewise/CTC** • Framewise/CTC (Details) • Framewise/CTC (Results) • Framewise/CTC (Outputs) • Dropout (Results)

Results – **Rimes** • **IAM** • **Bentham** • International Evaluations • Linguistic constraints • LM limitations • Decoding params • Combination • HTRtS contest

Misc –

References

Augustin, E., Carré, M., Grosicki, E., Brodin, J.-M., Geoffrois, E., & Preteux, F. (2006). RIMES evaluation campaign for handwritten mail processing. In Proceedings of the Workshop on Frontiers in Handwriting Recognition, 1.

Bertolami, R., & Bunke, H. (2008). Hidden Markov Model Based Ensemble Methods for Offline Handwritten Text Line Recognition. Pattern Recognition, 41(11), 3452 -- 3460.

Bianne-Bernard, A.-L. (2011). Reconnaissance de mots manuscrits cursifs par modèles de Markov cachés en contexte. Ph.D. thesis, Telecom ParisTech.

Bloomberg, D. S., Kopec, G. E., & Lakshmi Dasari (1995). Measuring document image skew and orientation. Proc. SPIE Document Recognition II, 2422(302), 302--316.

Bluche, T., Louradour, J., Knibbe, M., Moysset, B., Benzeghiba, M. F., & Kermorvant, C. (2014). The A2iA Arabic Handwritten Text Recognition System at the Open HaRT2013 Evaluation. In 11th IAPR International Workshop on Document Analysis Systems (DAS), (pp. 161--165). IEEE.

Bourlard, H., & Morgan, N. (1994). Connectionist speech recognition: a hybrid approach Chapter 7, vol. 247 of The Kluwer international series in engineering and computer science: VLSI, computer architecture, and digital signal processing. Kluwer Academic Publishers.

Buse, R., Liu, Z. Q., & Caelli, T. (1997). A structural and relational approach to handwritten word recognition. IEEE Transactions on Systems, Man and Cybernetics, 27(5), 847--61.

Doetsch, P., Kozielski, M., & Ney, H. (2014). Fast and robust training of recurrent neural networks for offline handwriting recognition. (pp. --).

Dreuw, P., Doetsch, P., Plahl, C., & Ney, H. (2011). Hierarchical hybrid MLP/HMM or rather MLP features for a discriminatively trained gaussian HMM: a comparison for offline handwriting recognition. In Image Processing (ICIP), 2011 18th IEEE International Conference on, (pp. 3541--3544). IEEE.

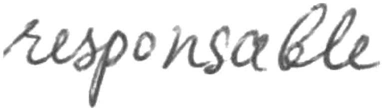
Espana-Boquera, S., Castro-Bleda, M. J., Gorbe-Moya, J., & Zamora-Martinez, F. (2011). Improving offline handwritten text recognition with hybrid HMM/ANN models. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 33(4), 767--779.

Fiscus, J. G. (1997). A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER). In IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU1997), (pp. 347--354). IEEE.

Gatos, B., Louloudis, G., Stamatopoulos, N., Ntirogiannis, K., Papandreou, A., Pratikakis, I., Zagoris, K., Sánchez, J. A., Romero, V., Toselli, A., Vidal, E., Villegas, M., Álvaro, F., & Bosch, V. (2013). Description and evaluation of tools for DIA HTR and KWS (M12). URL `http://transcriptorium.eu/pdfs/deliverables/tranScriptorium-D3.1.2-31December2013.pdf`

Gers, F. (2001). Long Short-Term Memory in Recurrent Neural Networks TH ESE. 2366.

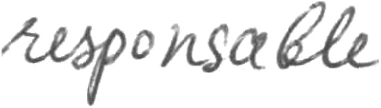
Coping with different writing styles



Goal: eliminate some of the variability of images

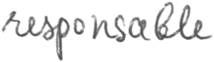
Examples:

Coping with different writing styles



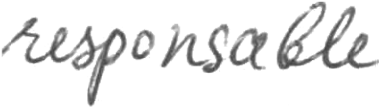
Goal: eliminate some of the variability of images

Examples:



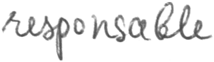
Correct of the
inclination of the text

Coping with different writing styles

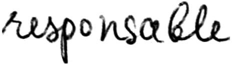


Goal: eliminate some of the variability of images

Examples:

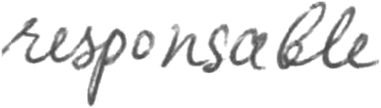


Correct of the inclination of the text



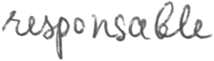
Normalize the contrast of the image

Coping with different writing styles

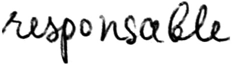


Goal: eliminate some of the variability of images

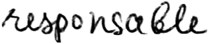
Examples:



Correct of the inclination of the text

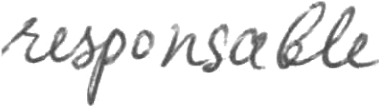


Normalize the contrast of the image



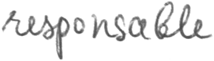
Normalize the size of the image

Coping with different writing styles

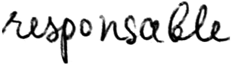


Goal: eliminate some of the variability of images

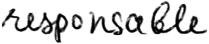
Examples:



Correct of the inclination of the text



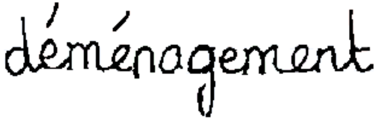
Normalize the contrast of the image



Normalize the size of the image

Other examples: correct the inclination of text lines (deskew), normalize the thickness of the writing, ...

How to deal with characters segmentation?



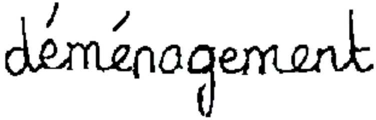
No segmentation
Whole-word (holistic) recognition

How to deal with characters segmentation?

No segmentation
Whole-word (holistic) recognition

Grapheme segmentation
Heuristic over-segmentation into part of characters

How to deal with characters segmentation?

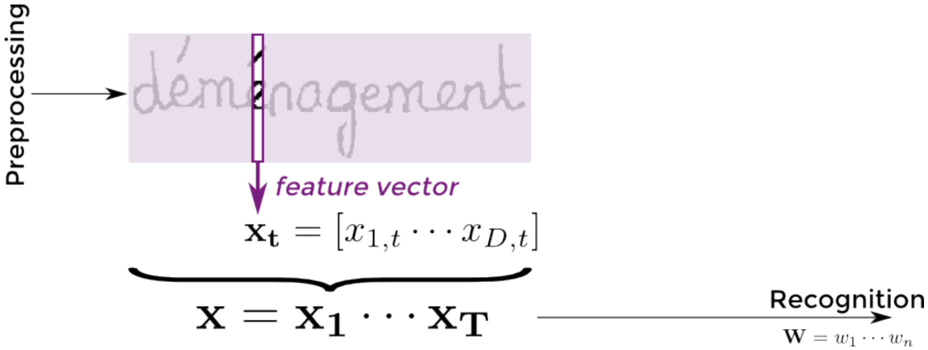


No segmentation
Whole-word (holistic) recognition

Grapheme segmentation
Heuristic over-segmentation into part of characters

Sliding Window
Sequences of image frames

How to extract relevant information from images?



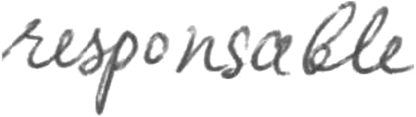
Low-level features – pixel counts and densities, black-white transitions, moments, centre of gravity, profiles, ...

High-level features – derivatives, contours, filters, HoG, pixel configurations, concavity features, ...

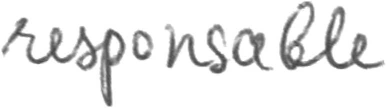
Shape features – loops, junctions, ascenders/descenders, ...

Technical Description of the Base System

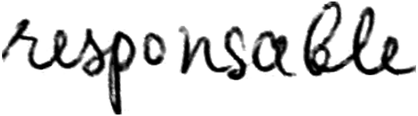
Preprocessing



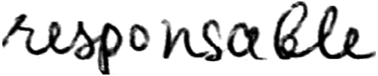
Correct skew



Correct slant



Normalize contrast by interpolation



Normalize height of different regions

Technical Description of the Base System

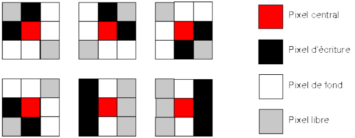
Preprocessing

Correct skew (Bloomberg et al., 1995) → Correct slant (Buse et al., 1997) → Normalize contrast by interpolation (Roeder, 2009) → Normalize height of different regions (Toselli et al., 2004)

Feature Extraction

Handcrafted features (Bianne-Bernard, 2011)

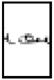
- Sliding window of 3px, with 3px step
- **56 handcrafted features** extracted from each frame
  - 8 pixel density measures
  - 12 pixel configurations
  - HoG in 8 directions
  - + deltas (= 28 + 28)



(Bianne-Bernard, 2011)

Pixel values

- 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



Technical Description of the Base System

Preprocessing

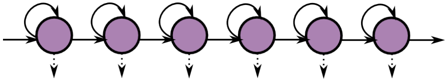
Correct skew (Bloomberg et al., 1995) → Correct slant (Buse et al., 1997) → Normalize contrast by interpolation (Roeder, 2009) → Normalize height of different regions (Toselli et al., 2004)

Feature Extraction

Handcrafted Features
Sliding win.: width 3px / shift 3px
Features: 56 geometrical and statistical features from Bianne-Bernard (2011)

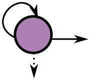
Pixels
Sliding win.: width 45px / shift 3px (Bentham 57px/3px), rescaled to height 20px
Features: 640 pixel values (Bentham: 800)

Optical Model



Emission: GMMs, MLPs

Transition: 6-state character models (5 for Rimes) and 2-state whitespace models



Emission: RNNs

Transition: 1-state character and blank models (CTC)

Technical Description of the Base System

Preprocessing

Correct skew (Bloomberg et al., 1995) $\rightarrow$ Correct slant (Buse et al., 1997) $\rightarrow$ Normalize contrast by interpolation (Roeder, 2009) $\rightarrow$ Normalize height of different regions (Toselli et al., 2004)

Feature Extraction

Handcrafted Features

Sliding win.: width 3px / shift 3px
Features: 56 geometrical and statistical features from Bianne-Bernard (2011)

Pixels

Sliding win.: width 45px / shift 3px (Bentham 57px/3px), rescaled to height 20px
Features: 640 pixel values (Bentham: 800)

Optical Model

Emission Model

Gaussian Mixture Models (GMMs; Baseline)
Multi-Layer Perceptrons (MLPs)
Recurrent Neural Networks (RNNs)

Transition Model

Loop + transition to next state
6-state character models (5 for Rimes) and 2-state whitespace models (GMMs, MLPs)
1-state character and blank models (RNNs)

Language Model

Database	Vocabulary	OOV Rate (Dev.)	n-gram	Training	Perplexity (Dev)
Rimes	5k	2.9%	4	Training set	18
IAM	50k	4.3%	3	LOB+Brown+Wellington	298
Bentham	33k	5.6%	3	Training set	108

Base system (optimization)

(GMM/HMM IAM, small training set, small LM)

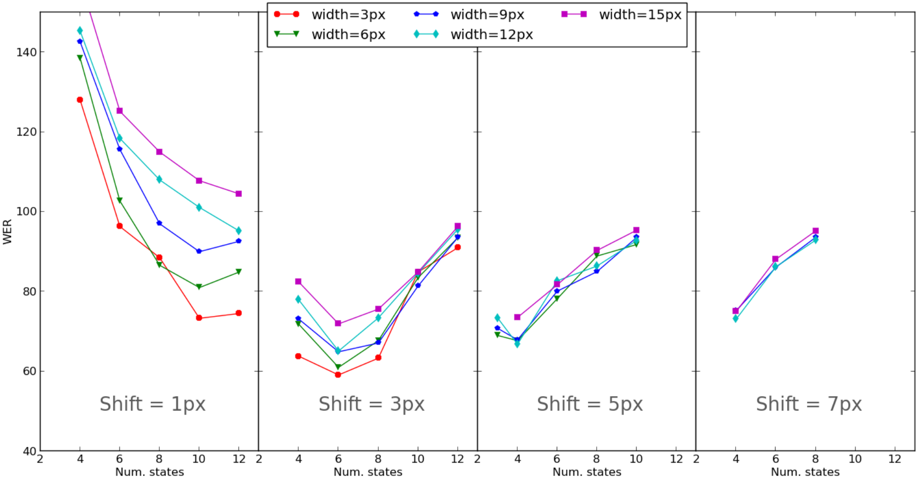
Contrast enhancement

	Window size:	6px	9px
Method	None	54.2%	58.0%
	Adaptive	57.2%	58.5%
	Interpolation	**53.1%**	57.2%

Height normalization

	Window size:	6px	9px
Method	None	56.9%	59.6%
	Fixed (72px)	54.2%	58.7%
	Region (22px, 33px, 17px)	58.7%	63.8%
	Region (24px, 24px, 24px)	**53.1%**	57.2%

Base system (optimization)



Base system (variations)

(GMM/HMM IAM)

Context-dependent models

Model	WER	CER
Context-independent	16.2	6.9
Context-dependent	16.3	6.6

LM at paragraph level

LM scope	WER	CER
Lines	16.2	6.9
Paragraphs	15.2	6.3

State-of-the-Art GMM/HMM Performance for HWR

Results on Rimes databas

	WER
Our GMM/HMM	15.8
GMM/HMM systems	
Kozielski et al. (2014)	15.7
Grosicki & El-Abed (2011)	31.2
Other systems	
Pham et al. (2014)	12.3
Doetsch et al. (2014)	12.9
Messina & Kermorvant (2014)	13.3

Results on Bentham database (Dev.)

	WER
Our GMM/HMM	27.9
GMM/HMM systems	
Gatos et al. (2013)	32.6

Results on IAM database

	WER
Our GMM/HMM	19.6
GMM/HMM systems	
Kozielski et al. (2013b)	17.3
Kozielski et al. (2013b)	22.2
Toselli et al. (2010)	25.8
Bertolami & Bunke (2008)	32.8
Other systems	
Doetsch et al. (2014)	12.2
Kozielski et al. (2013a)	13.3
Pham et al. (2014)	13.6

DB statistiques

Number of pages, lines, words and characters in each dataset

	Set	#Pages	#Lines	#Words	(unique)	#Characters	(unique)
Rimes (French)	Train	1,391	10,203	73,822	(8,061)	460,201	(97)
	Dev.	149	1,130	8,380		51,924	
	Eval.	100	778	5,639		35,286	

	Set	#Pages	#Lines	#Words	(unique)	#Characters	(unique)
IAM (English)	Train	747	6,482	55,081	(7,843)	287,727	(79)
	Dev.	116	976	8,895		43,050	
	Eval.	336	2,915	25,920		128,531	

	Set	#Pages	#Lines	#Words	(unique)	#Characters	(unique)
Bentham (English)	Train	350	9,198	76,707	(12,104)	419,764	(93)
	Dev.	50	1,415	11,580		64,070	
	Eval.	33	860	7,868		40,231	

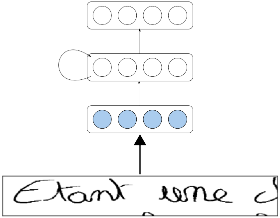
context

	Rimes	IAM	Bentham
none	14.5%	12.4%	26.0%
±1	14.1%	12.1%	23.4%
±3	13.9%	13.1%	21.2%
±5	14.5%	12.4%	20.1%
±7	15.8%	12.4%	20.8%

	RNNs Rimes	IAM
none	14.1%	12.2%
±1	14.2%	12.1%
±3	13.7%	12.6%
±5	14.1%	12.6%

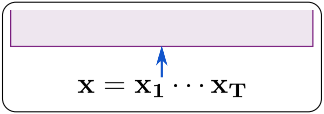
What context RNNs learn?

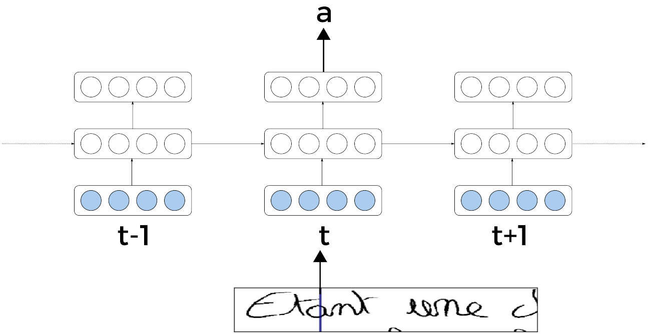
$$\mathbf{x} = \mathbf{x_1} \cdots \mathbf{x_T}$$



(Graves et al., 2013)

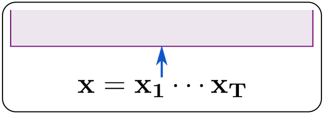
What context RNNs learn?

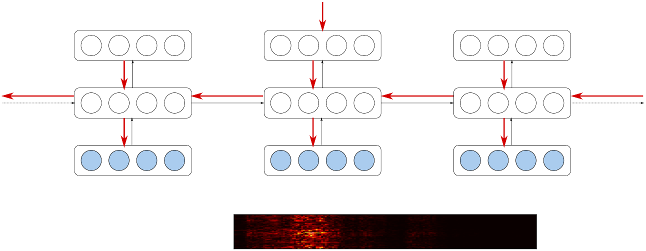




(Graves et al., 2013)

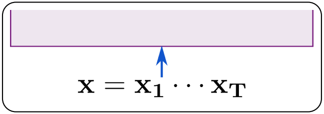
What context RNNs learn?

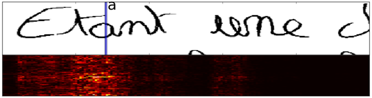




(Graves et al., 2013)

What context RNNs learn?

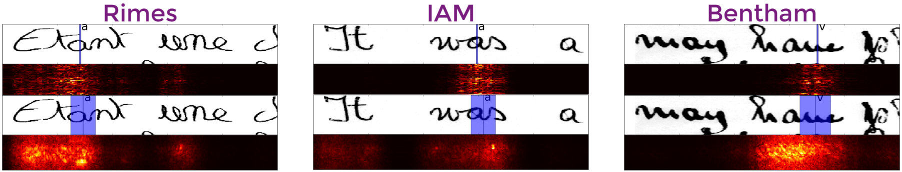




(Graves et al., 2013)

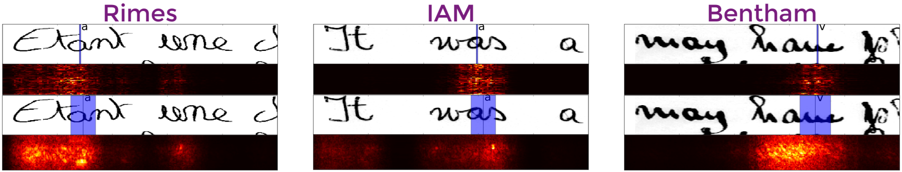
What context RNNs learn?

- Visulalization: gradient of the output wrt the input (Graves et al., 2013)
- Top: features; Bottom: pixels



What context RNNs learn?

- Visulalization: gradient of the output wrt the input (Graves et al., 2013)
- Top: features; Bottom: pixels



⟶ RNNs automatically use the context, which can even **extend beyond character boundaries**

Neural Network Architectures

MLPs

- **Inputs:** concatenation of $\pm\delta$ frames around the current one
- **Hidden layers:** linear+bias and sigmoid activation
- **Outputs:** one per HMM state (≈ 500) and softmax

		Context	Layers
Rimes	Features	± 3 fr.	3×512
	Pixel	-	5×512
IAM	Feature	± 3 fr.	5×256
	Pixels	-	$5 \times 1,024$
Bentham	Feature	± 5 fr.	7×512
	Pixels	-	6×512

RNNs

- **Inputs:** sequences of frames (no context)
- **Hidden layers:** alternate
  - LSTM layers, one in each direction with the same number of LSTM units
    no peephole connection, cell input and input/output/forget gates have the same inputs
    $tanh$ activation
  - feedforward $tanh$ layer (after linear transform of concat output of LSTMs in both directions)
- **Outputs:** one per character + blank $\oslash$ (≈ 500) and softmax

		Context	layers
Rimes	Features	-	7×200
	Pixel	-	5×200
IAM	Feature	-	5×200
	Pixels	-	7×200
Bentham	Feature	-	5×200
	Pixels	-	7×200

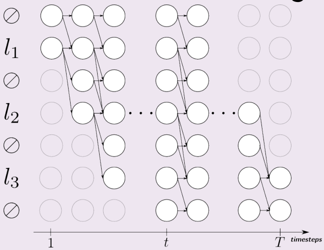
NN training

MLPs

1. forced alignments with GMM-HMM system
2. layerwise pretraining of RBMs with CD1 (1st is Gauss.-Bern, others are Bern.-Bern.)
   $LR = 0.001$, L_2 reg. $\lambda = 0.0002$
3. Cross-entropy fine-tuning $LR = 0.008$ start halving when impr < 0.2
4. stop when impr. < 0.01
5. sMBR $LR = 0.00001$

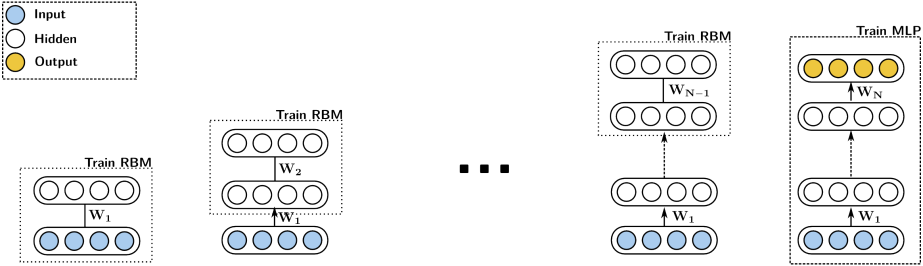
RNNs

CTC training

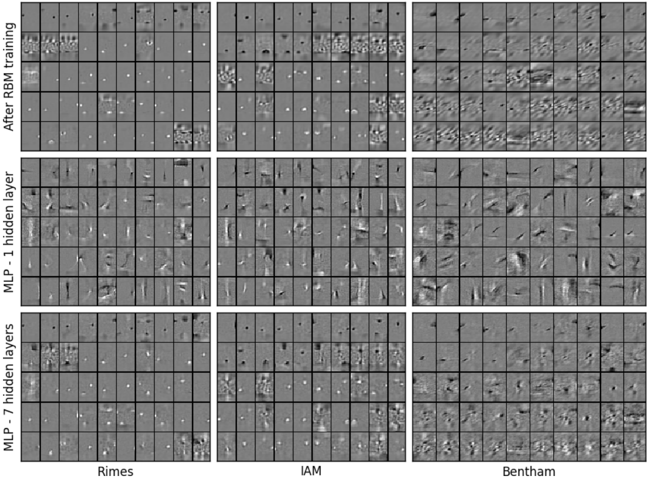


- no forced alignment, targets are the character sequences
- CTC training (summation over all possible segmentations) with $LR = 0.01$
- early stopping: keep best net if not improvement of CTC cost for over 20 epochs
- no regularization

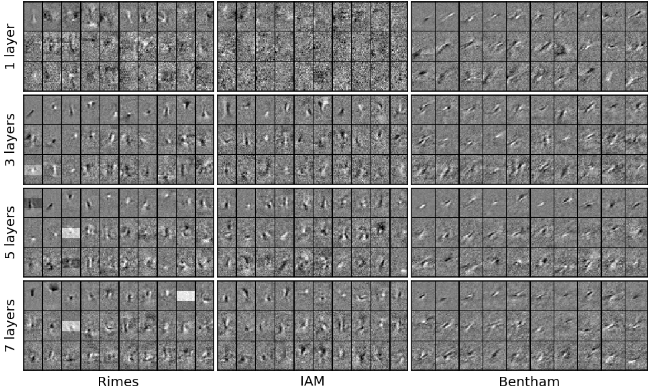
Pre-training of MLPs



MLP Weights


After RBM training
MLP - 1 hidden layer
MLP - 7 hidden layers
Rimes
IAM
Bentham

RNN Weights



Effect of Depth (WER%)

MLPs

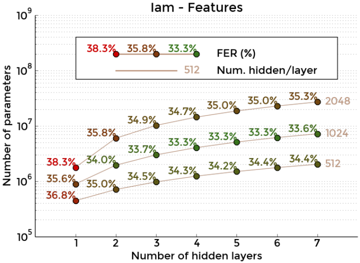
WER%	Shallow	→	Deep
	Features		
Rimes	14.0	→	13.5
IAM	12.4	→	11.8
Bentham	21.5	→	20.1
	Pixels		
Rimes	15.3	→	14.0
IAM	13.6	→	12.3
Bentham	28.8	→	22.4

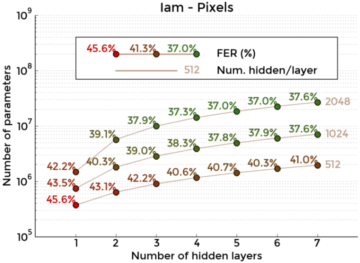
RNNs

WER%	Shallow	→	Deep
	Features		
Rimes	14.9	→	12.9
IAM	13.4	→	11.4
Bentham	20.6	→	18.0
	Pixels		
Rimes	24.1	→	14.0
IAM	-	→	12.8
Bentham	33.8	→	20.3

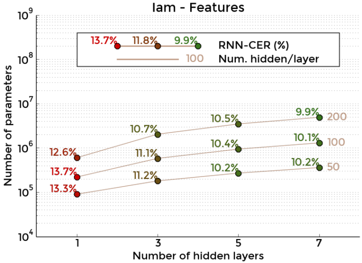
Effect of Depth (not parameters) –IAM

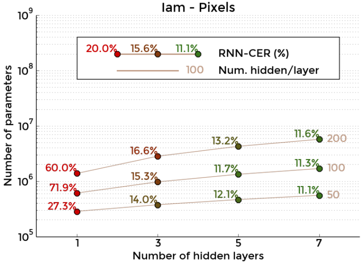
MLPs





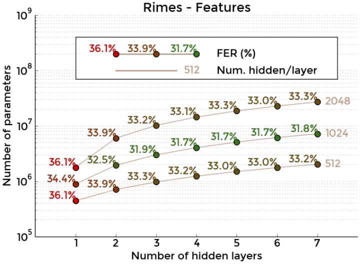
RNNs

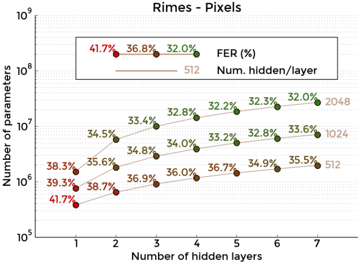




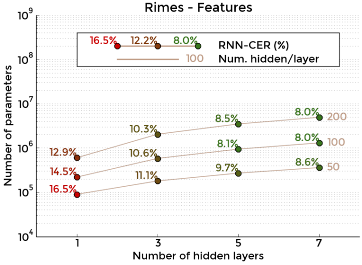
Effect of Depth (not parameters) –Rimes

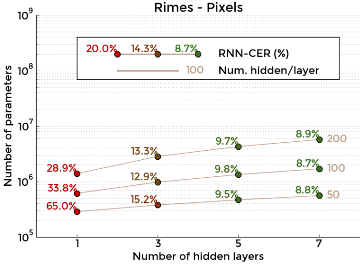
MLPs





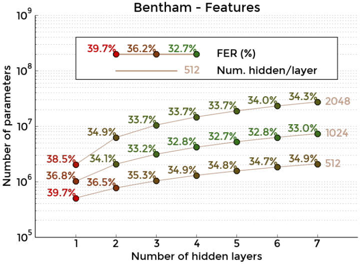
RNNs

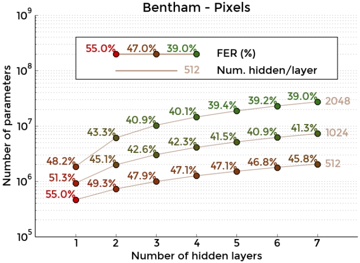




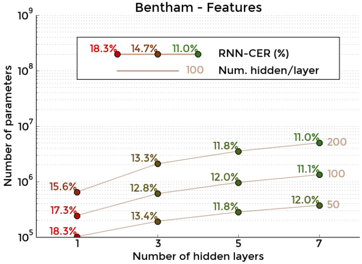
Effect of Depth (not parameters) –Bentham

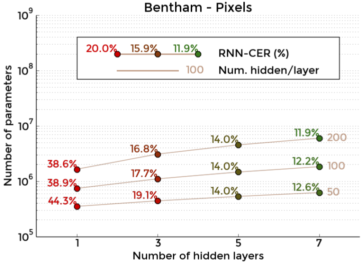
MLPs



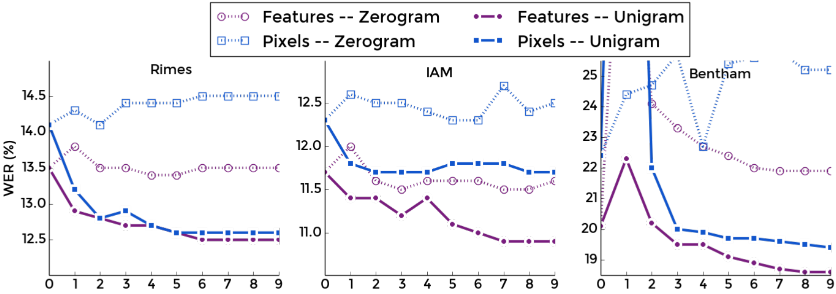


RNNs

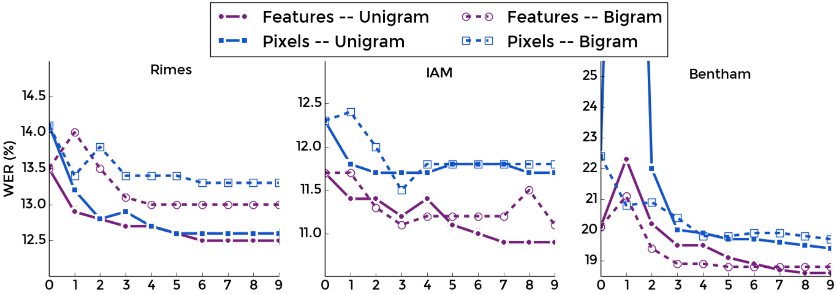




Sequence Training of MLPs



Sequence Training of MLPs



sMBR Results

		Features		Pixels	
		WER%	CER%	WER%	CER%
Rimes	Cross-entropy	13.5	3.8	14.1	4.2
	+ sMBR	12.5	3.4	12.6	3.8
		(-7.4%)	(-10.5%)	(-10.6%)	(-9.5%)
IAM	Cross-entropy	11.7	4.2	12.3	4.2
	+ sMBR	10.9	3.7	11.7	4.0
		(-6.8%)	(-11.9%)	(-4.9%)	(-4.5%)
Benth.	Cross-entropy	20.1	8.5	22.4	10.6
	+ sMBR	18.6	7.4	19.4	8.4
		(-7.5%)	(-12.9%)	(-13.4%)	(-20.8%)

Dropout Results

	Dropout	Before	Inside	After	All
Rimes (Features)	None	8.2			
	Bottom	6.8	6.7	8.2	**6.7**
	Middle	**6.6**	7.3	7.2	6.8
	Top	7.4	8.6	8.0	8.8
	All	**5.0**	5.4	6.8	7.1
(Pixels)	None	9.7			
	Bottom	**7.1**	7.6	8.1	**7.2**
	Middle	7.5	9.6	9.0	8.8
	Top	8.0	9.2	7.8	9.1
	All	**5.8**	6.0	6.5	7.4

	Dropout	Before	Inside	After	All
IAM (Features)	None	10.4			
	Bottom	9.1	**8.5**	9.8	8.8
	Middle	8.9	9.1	8.6	**8.7**
	Top	9.1	10.2	9.5	10.4
	All	7.9	**7.0**	9.0	9.4
(Pixels)	None	13.2			
	Bottom	10.0	**9.1**	11.4	**10.1**
	Middle	10.1	11.1	10.6	10.8
	Top	10.9	12.3	11.1	12.6
	All	8.6	**8.4**	10.1	11.4

	Dropout	Before	Inside	After	All
Bentham (Features)	None	11.0			
	Bottom	**8.5**	9.9	12.3	**8.8**
	Middle	9.8	9.9	10.4	10.0
	Top	10.5	11.2	10.7	12.3
	All	**7.4**	8.1	10.0	8.5
(Pixels)	None	14.0			
	Bottom	10.4	**9.9**	13.4	**9.7**
	Middle	11.0	13.6	12.2	13.0
	Top	12.0	15.1	12.7	14.4
	All	**8.0**	9.4	10.8	12.3

Dropout Results

Rimes		Handcrafted Features RNN-CER (%)	Handcrafted Features WER (%)	Handcrafted Features CER (%)	Pixels RNN-CER (%)	Pixels WER (%)	Pixels CER (%)
5 hidden layers	no dropout	8.2	12.9	3.7	9.7	15.5	4.8
	after	6.8	**12.8**	**3.6**	6.5	**13.3**	4.1
	inside	5.4	13.2	3.8	6.0	14.3	4.6
	before	**5.0**	13.1	3.7	**5.8**	13.8	**4.0**
7 hidden layers	no dropout	8.0	14.1	4.1	8.9	14.7	5.0
	after	5.7	**12.7**	**3.6**	6.0	**13.6**	**4.1**
	inside	5.3	**12.7**	3.7	5.9	14.2	4.6
	before	**4.8**	**12.7**	3.7	**5.3**	13.7	4.2
IAM							
5 hidden layers	no dropout	10.4	11.7	4.0	13.2	14.7	5.7
	after	9.0	11.8	4.1	10.1	13.2	4.7
	inside	**7.0**	**11.6**	**3.9**	**8.4**	13.3	5.0
	before	7.9	12.3	4.2	8.6	**12.4**	**4.5**
7 hidden layers	no dropout	10.1	12.9	4.6	11.6	14.6	5.5
	after	8.1	11.9	**3.9**	**7.5**	**11.8**	**4.0**
	inside	**7.1**	11.9	4.1	7.9	13.0	4.7
	before	7.4	**11.7**	4.1	8.3	13.2	4.8
Bentham							
5 hidden layers	no dropout	11.0	18.1	7.0	14.0	21.3	9.0
	after	10.0	17.3	6.9	10.8	19.1	7.7
	inside	8.1	17.7	6.8	9.4	20.0	8.5
	before	**7.4**	**16.6**	**6.2**	**8.0**	**17.8**	**6.9**
7 hidden layers	no dropout	11.0	18.0	7.0	11.9	20.6	8.4
	after	8.9	17.2	6.7	8.9	18.7	7.3
	inside	7.1	17.4	6.5	8.7	20.1	8.4
	before	**6.5**	**16.7**	**6.1**	**7.5**	**17.7**	**6.4**

Dropout Results

Dataset	Layers	Dropout	Handcrafted Features RNN-CER (%)	Handcrafted Features WER (%)	Handcrafted Features CER (%)	Pixels RNN-CER (%)	Pixels WER (%)	Pixels CER (%)
Rimes	5 hidden layers	after all	6.8	12.8	**3.6**	6.5	13.3	4.1
		before all	**5.0**	13.1	3.7	5.8	13.8	4.0
		bef. 1 / aft. 2-3	5.5	12.8	**3.6**	6.3	13.5	4.0
		bef. 1-2 / aft. 3	5.6	**12.7**	**3.6**	6.0	13.7	4.2
		bef.+aft. all	5.4	**12.7**	3.7	**5.3**	**12.7**	**3.9**
	7 hidden layers	after all	5.5	**12.7**	**3.6**	6.0	13.6	4.1
		before all	4.8	**12.7**	3.7	**5.3**	13.7	4.2
		bef. 1-2 / aft. 3-4	5.3	**12.7**	3.7	6.2	13.6	4.1
		bef. 1-2-3 / aft. 4	5.1	13.3	3.8	5.9	13.6	4.1
		bef.+aft. all				5.6	13.7	4.2
IAM	5 hidden layers	after all	9.0	11.8	4.1	10.1	13.2	4.7
		before all	7.9	12.3	4.2	8.6	12.4	4.5
		bef. 1 / aft. 2-3	8.2	11.6	4.0	8.0	11.9	4.1
		bef. 1-2 / aft. 3	8.1	**11.2**	**3.8**	8.3	11.8	4.2
		bef.+aft. all	7.8	12.2	4.1	7.9	11.6	4.1
	7 hidden layers	after all	8.1	11.9	3.9	**7.5**	**11.4**	**3.9**
		before all	**7.4**	11.7	4.1	8.3	13.2	4.8
		bef. 1-2 / aft. 3-4	8.0	11.5	3.9	7.9	11.6	4.0
		bef. 1-2-3 / aft. 4	7.5	11.6	3.9	8.2	12.3	4.2
		bef.+aft. all				8.1	13.3	4.5
Bentham	5 hidden layers	after all	10.0	17.3	6.9	10.8	19.1	7.7
		before all	7.4	16.6	6.2	8.0	17.8	6.9
		bef. 1 / aft. 2-3	7.1	16.1	5.8	8.4	17.6	6.7
		bef. 1-2 / aft. 3	7.4	**16.0**	6.0	8.7	18.1	6.7
		bef.+aft. all	7.3	17.1	6.3	7.5	17.5	6.7
	7 hidden layers	after all	8.9	17.2	6.7	8.9	18.7	7.3
		before all	**6.5**	16.7	6.1	7.5	17.7	6.4
		bef. 1-2 / aft. 3-4	6.7	16.1	5.8	**7.1**	**17.0**	**6.2**
		bef. 1-2-3 / aft. 4	6.7	16.3	**5.7**	7.3	17.6	6.4
		bef.+aft. all				**7.1**	17.7	6.5

Training strategies of Hybrid NN/HMM

	Framewise (cross-entropy)	**HMM training** (NN/HMM) (Hennebert et al., 1997)	**CTC training** (Graves et al., 2006)
Output/Topology			
Num. states/char.	Several (HMM)	Several (HMM)	1
Special NN output	✗	✗	✓($\oslash$)
Training/Cost			
Cost function	$-\log \prod_t p(q_t \vert x_t)$	$-\log \sum_{\mathbf{q}} \prod_t \frac{p(q_t \vert x_t)}{p(q_t)} p(q_t \vert q_{t-1})$	$-\log \sum_{\mathbf{q}} \prod_t p(q_t \vert \mathbf{x})$
Transition probas	✗	✓	✗
Prior probas	✗	✓	✗
Forward-backward	✗	✓	✓
α, β	✗	Same eqns. except for transition/prior probabilities	

Training strategies of Hybrid NN/HMM

	Framewise (cross-entropy)	HMM training (NN/HMM) (Hennebert et al., 1997)	CTC training (Graves et al., 2006)
Output/Topology			
Num. states/char.	Several (HMM)	Several (HMM)	1
Special NN output	✗	✗	✓(∅)
Training/Cost			
Cost function	$-\log \prod_t p(q_t\|x_t)$	$-\log \sum_{\mathbf{q}} \prod_t \frac{p(q_t\|x_t)}{p(q_t)} p(q_t\|q_{t-1})$	$-\log \sum_{\mathbf{q}} \prod_t p(q_t\|\mathbf{x})$
Transition probas	✗	✓	✗
Prior probas	✗	✓	✗
Forward-backward	✗	✓	✓
α, β	✗	Same eqns. except for transition/prior probabilities	

N.B. – CTC is associated with a specific topology for standalone NN recognition

Training strategies of Hybrid NN/HMM

	Framewise (cross-entropy)	HMM training (NN/HMM) (Hennebert et al., 1997)	CTC training (Graves et al., 2006)
Output/Topology			
Num. states/char.	Several (HMM)	Several (HMM)	1
Special NN output	✘	✘	✓($\oslash$)
Training/Cost			
Cost function	$-\log \prod_t p(q_t \mid x_t)$	$-\log \sum_{\mathbf{q}} \prod_t \frac{p(q_t \mid x_t)}{p(q_t)} p(q_t \mid q_{t-1})$	$-\log \sum_{\mathbf{q}} \prod_t p(q_t \mid \mathbf{x})$
Transition probas	✘	✓	✘
Prior probas	✘	✓	✘
Forward-backward	✘	✓	✓
α, β	✘	Same eqns. except for transition/prior probabilities	

N.B. – CTC is associated with a specific topology for standalone NN recognition

CTC = HMM training, without transition/prior probabilities (zeroth-order model), and with a specific topology (for standalone NN recognition)
$\Longrightarrow$ CTC could be applied with different topologies, to other kinds of NN than RNN

Training strategies of Hybrid NN/HMM

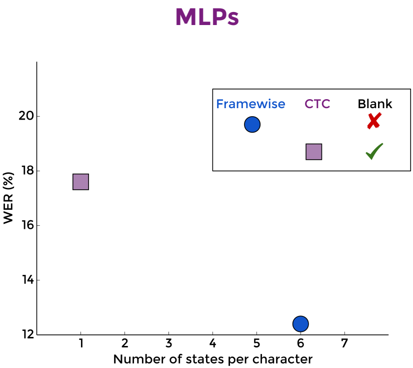
	Framewise (cross-entropy)	**HMM training** (NN/HMM) (Hennebert et al., 1997)	**CTC training** (Graves et al., 2006)
Output/Topology			
Num. states/char.	Several (HMM)	Several (HMM)	1
Special NN output	✗	✗	✓($\oslash$)
Training/Cost			
Cost function	$-\log \prod_t p(q_t \mid x_t)$	$-\log \sum_{\mathbf{q}} \prod_t \frac{p(q_t \mid x_t)}{p(q_t)} p(q_t \mid q_{t-1})$	$-\log \sum_{\mathbf{q}} \prod_t p(q_t \mid \mathbf{x})$
Transition probas	✗	✓	✗
Prior probas	✗	✓	✗
Forward-backward	✗	✓	✓
α, β	✗	Same eqns. except for transition/prior probabilities	

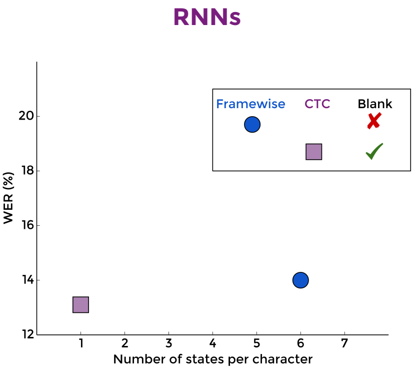
N.B. – CTC is associated with a specific topology for standalone NN recognition

CTC = HMM training, without transition/prior probabilities (zeroth-order model), and with a specific topology (for standalone NN recognition)
$\Longrightarrow$ CTC could be applied with different topologies, to other kinds of NN than RNN

CTC = Cross-entropy training + forward-backward to consider all possible segmentations
$\Longrightarrow$ we can compare the training strategies, see the effect of forward-backward, with different topologies

Framewise and CTC

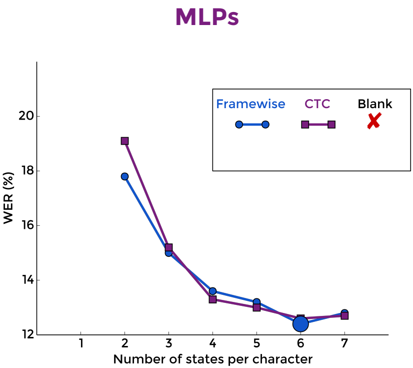


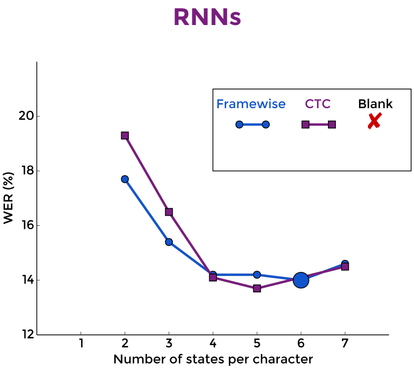


$\longrightarrow$ CTC works well with RNNs, not so much with MLPs

(MLP: 2x1024, ± 5 frames - RNN: 1x100))

Framewise and CTC

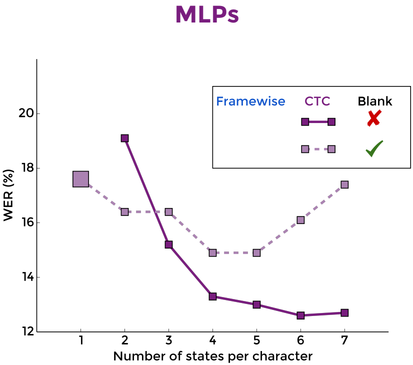


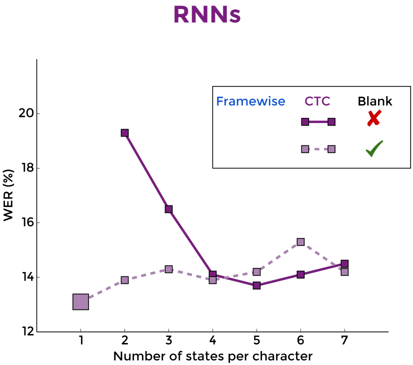


$\longrightarrow$ Forward-backward aspect do not improve the results, and is worse with too few states

(MLP: 2x1024, ± 5 frames - RNN: 1x100))

Framewise and CTC



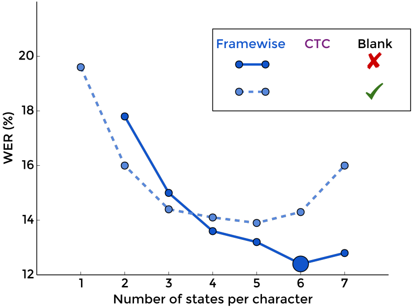


$\longrightarrow$ The blank symbol only helps with a few states for CTC training, ...

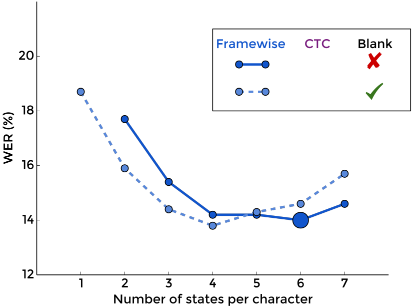
(MLP: 2x1024, ± 5 frames - RNN: 1x100))

Framewise and CTC

MLPs



RNNs

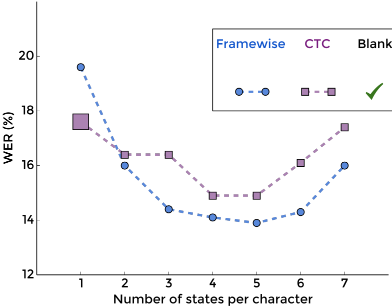


$\longrightarrow$ … and for framwise training too, although not as much as adding a state to the character models

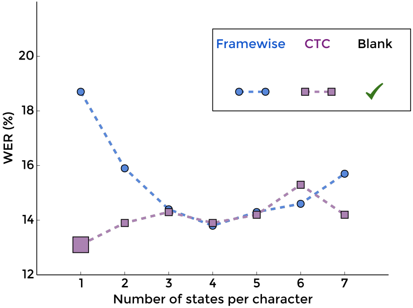
(MLP: 2x1024, ± 5 frames - RNN: 1x100))

Framewise and CTC

MLPs



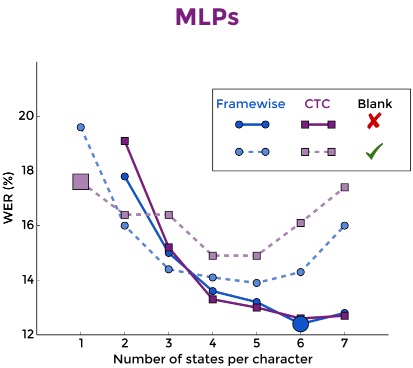
RNNs

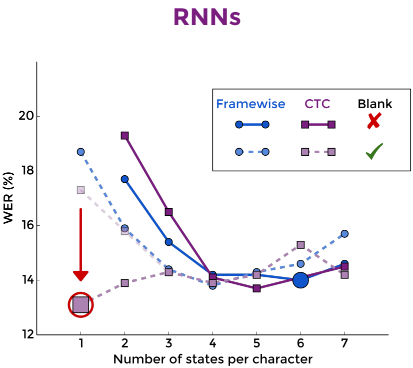


$\longrightarrow$ Forward-backward with blank do not improve so much the results except with only a few states

(MLP: 2x1024, ± 5 frames - RNN: 1x100))

Framewise and CTC





$\longrightarrow$ CTC+blank, with one-state models, is especially suited to RNNs

(MLP: 2x1024, ± 5 frames - RNN: 1x100))

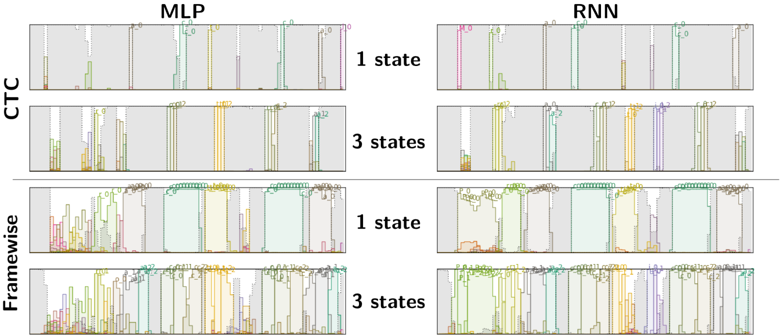
Framewise vs. CTC - Nets alone

Framewise -- Label Classification Error (frame level)								
	States	**1**	**2**	**3**	**4**	**5**	**6**	**7**
MLP	No blank		23.8	24.7	25.8	26.2	28.2	29.3
	Blank	17.1	18.8	20.8	22.0	23.2	25.4	28.5
RNN	No blank		14.4	15.4	16.3	17.2	19.6	20.7
	Blank	11.3	12.8	14.2	15.0	16.0	19.0	22.2
CTC -- Label Edit Distance (sequence level)								
	States	**1**	**2**	**3**	**4**	**5**	**6**	**7**
MLP	No blank		77.0	53.8	44.4	39.6	34.8	32.6
	Blank	18.5	18.9	21.8	26.1	23.9	22.9	24.0
RNN	No blank		23.6	19.0	17.7	16.6	15.6	15.8
	Blank	9.2	10.7	11.5	11.6	12.2	13.0	13.0

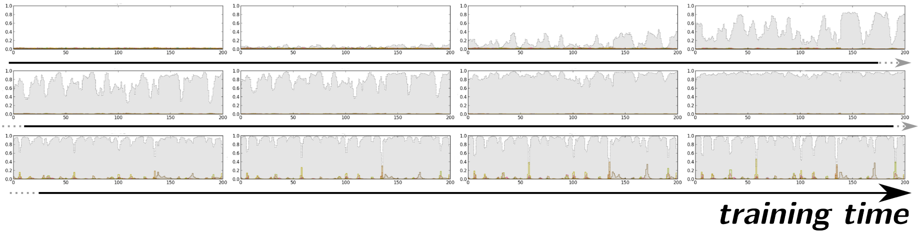
Framewise vs. CTC – Net+LM

	States	Without blank Framewise	Without blank CTC	With blank Framewise	With blank CTC
MLP	1	-	-	19.6 / 9.0	17.6 / 7.4
	2	17.8 / 8.2	19.1 / 8.5	16.0 / 6.3	16.4 / 6.7
	3	15.0 / 6.1	15.2 / 6.1	14.4 / 5.5	16.4 / 6.5
	4	13.6 / 5.3	13.3 / 4.9	14.1 / 5.2	14.9 / 5.6
	5	13.2 / 4.8	13.0 / 4.5	13.9 / 5.2	14.9 / 5.6
	6	**12.4** / 4.6	12.6 / **4.3**	14.3 / 5.9	16.1 / 6.4
	7	12.8 / 4.8	12.7 / 4.3	16.0 / 6.7	17.4 / 7.0
RNN	1	-	-	18.7 / 8.2	**13.1 / 4.9**
	2	17.7 / 7.5	19.3 / 8.0	15.9 / 6.1	13.9 / 5.0
	3	15.4 / 5.6	16.5 / 6.1	14.4 / 5.8	14.3 / 5.2
	4	14.2 / 5.4	14.1 / 5.3	13.8 / 5.3	13.9 / 5.1
	5	14.2 / 5.1	13.7 / 5.0	14.3 / 5.2	14.2 / 5.1
	6	14.0 / 5.1	14.1 / 4.9	14.6 / 5.8	15.3 / 5.8
	7	14.6 / 5.2	14.5 / 5.1	15.7 / 6.4	14.2 / 5.4

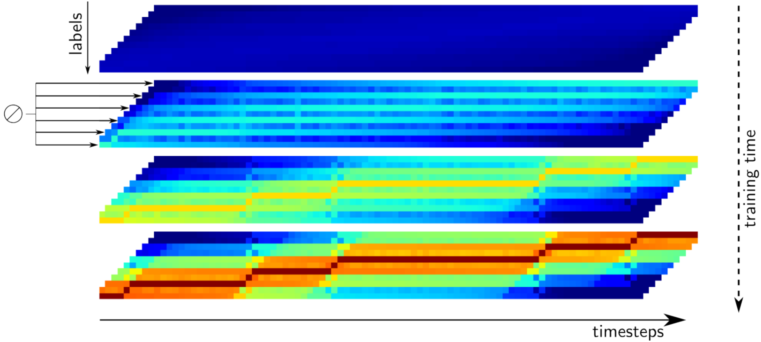
Framewise vs. CTC – Outputs



CTC - Outputs in training

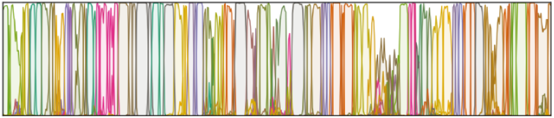


CTC – Why peaks



CTC - No blank problem

Framewise



CTC random init

CTC init 1ep framewise

Combination

Two methods:

- **ROVER**: transcription-level (Fiscus, 1997)
- **Lattice**-based (Xu et al., 2011)

		Rimes		IAM	
		WER%	CER%	WER%	CER%
Deep MLP	Features	12.5	3.4	10.9	3.7
	Pixels	12.6	3.8	11.7	4.0
Deep RNN	Features	12.8	3.8	11.2	3.8
	Pixels	12.7	4.0	11.4	3.9
ROVER combination		11.3	3.5	9.6	3.6
Lattice combination		11.2	3.3	9.6	3.3

International Evaluations

With **A2iaLab**:

- **1st** in OpenHaRT'13 restricted track
  2nd in unrestricted track
- **1st** in MAURDOR'13 evaluation
- **participation** to HTRtS'15 evaluation (results not yet public)

Own system:

- **2nd** in HTRtS'14 restricted track
  2nd in unrestricted track

Effect of linguistic constraints

MLPs

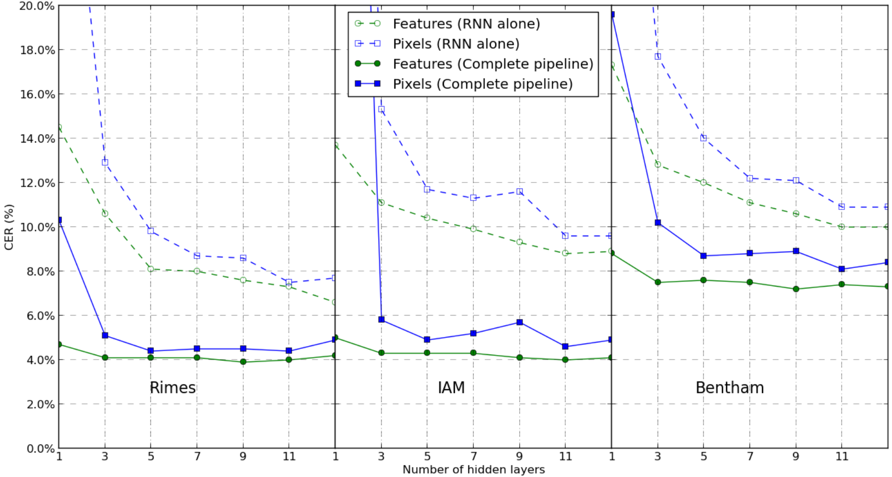
		Features		Pixels	
		WER%	CER%	WER%	CER%
Rimes	no lexicon	61.1	17.8	59.5	17.8
	lexicon	26.9	6.8	26.1	7.2
	lexicon+LM	12.5	3.4	12.6	3.8
IAM	no lexicon	54.7	15.8	54.2	15.6
	lexicon	24.7	7.7	25.5	8.0
	lexicon+LM	10.9	3.7	11.7	4.0

RNNs

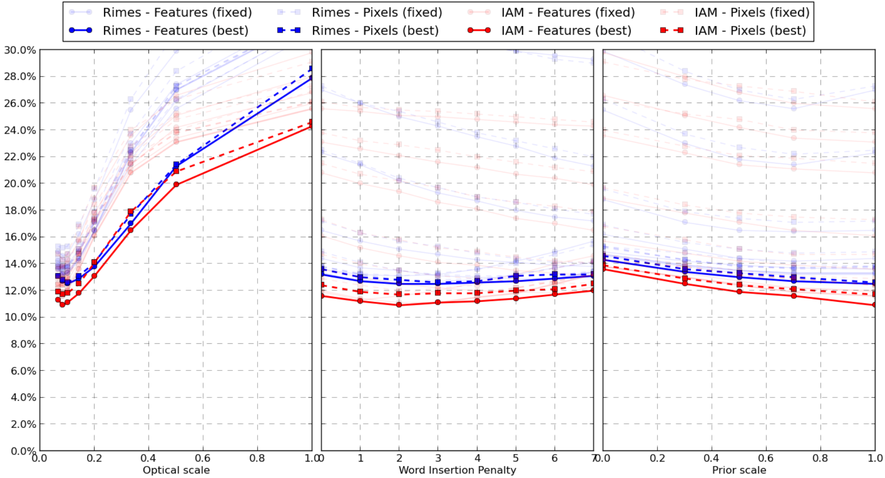
		Features		Pixels	
		WER%	CER%	WER%	CER%
Rimes	no lexicon	20.1	5.1	20.9	5.6
	lexicon	16.7	5.3	16.4	4.3
	lexicon+LM	12.8	3.8	12.7	4.0
IAM	no lexicon	27.5	7.9	24.7	7.3
	lexicon	17.6	5.5	16.7	5.3
	lexicon+LM	11.2	3.8	11.4	3.9

Illustration of LM limitations

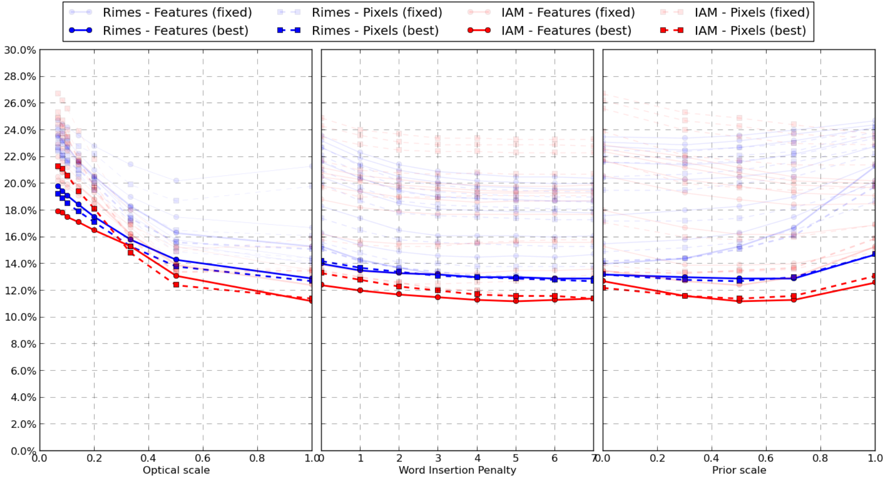
(RNNs, comparison of RNN-CER of net alone and CER with LM)



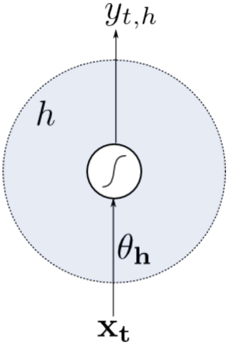
Effect of decoding parameters (MLPs)



Effect of decoding parameters (RNNs)



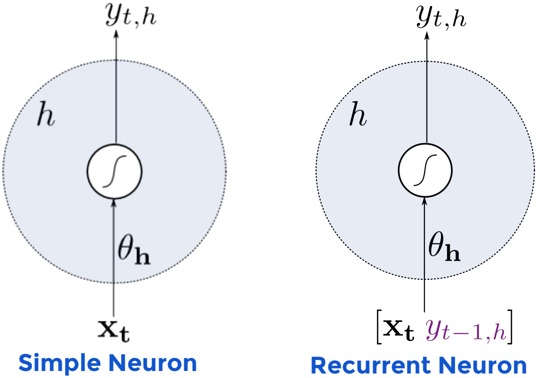
Artificial Neurons



Simple Neuron

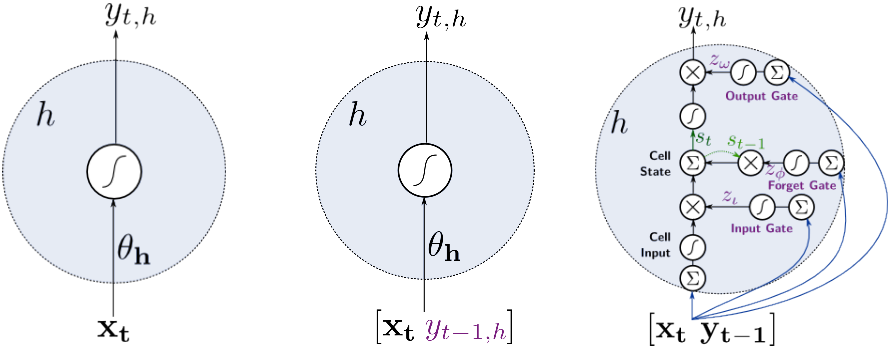
- Each term of a vector of input is multiplied by some weight
- A non-linear activation function is applied to the sum
- The result is the output of the neuron
- The weights are the parameters of the model, adjusted by training

Artificial Neurons



The inputs of the neuron include the output at the previous timestep.

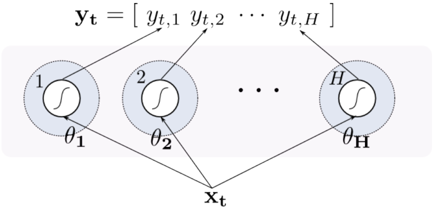
Artificial Neurons



Simple Neuron | Recurrent Neuron | LSTM Neuron

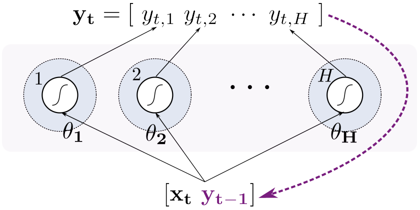
A gating mechanism, with adjustable weights, controls the flow of information into and out of the neuron, and the update of the internal state. (Hochreiter & Schmidhuber, 1997; Gers, 2001)

Artificial Neural Networks



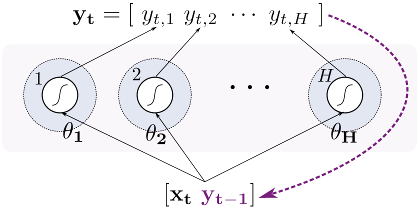
Layered Structure

Artificial Neural Networks

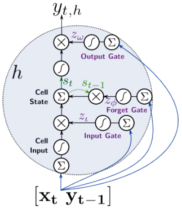


Layered Structure

Artificial Neural Networks

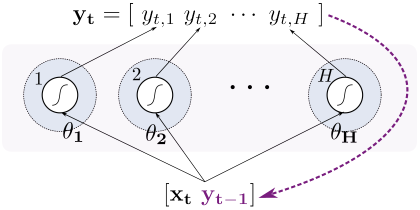


Layered Structure

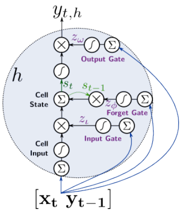


LSTM Neuron

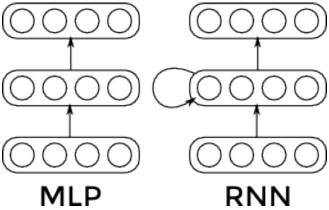
Artificial Neural Networks



Layered Structure

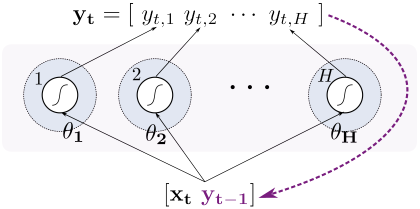


LSTM Neuron

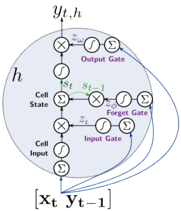


Neural Networks

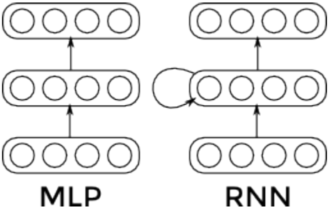
Artificial Neural Networks



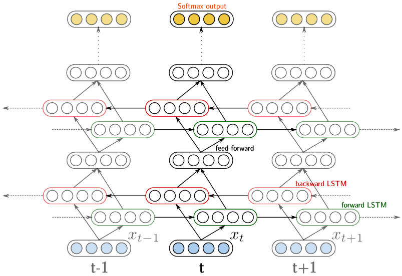
Layered Structure



LSTM Neuron



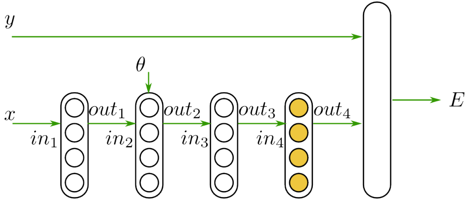
Neural Networks



Bidirectional RNN

Neural Network Training

Gradient-descent by backpropagation of the error

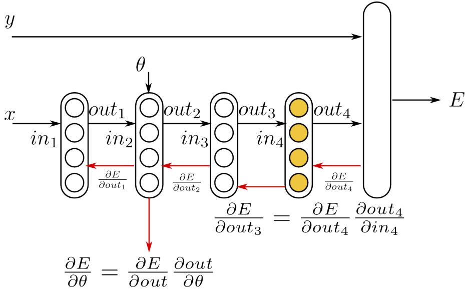


Given a training set $\mathcal{S} = \{(\mathbf{x}, y)\}$

1. compute the output of each layer in turn ($in_{i+1} = out_i$)
2. compute a measure of error E between actual and expected output

Neural Network Training

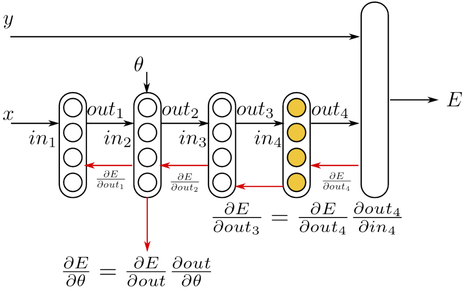
Gradient-descent by backpropagation of the error



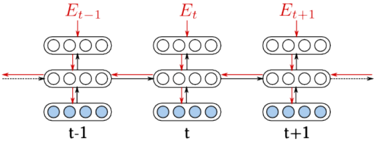
1. propagate the error backward using $\frac{\partial E}{\partial in} = \frac{\partial E}{\partial out} \frac{\partial out}{\partial in}$
2. compute the gradient wrt the parameters $\frac{\partial E}{\partial \theta} = \frac{\partial E}{\partial out} \frac{\partial out}{\partial \theta}$
3. update the parameters using $\theta \leftarrow \theta - \eta \frac{\partial E}{\partial \theta}$

Neural Network Training

Gradient-descent by backpropagation of the error



1. **propagate the error** backward using $\frac{\partial E}{\partial in} = \frac{\partial E}{\partial out} \frac{\partial out}{\partial in}$
2. **compute the gradient** wrt the parameters $\frac{\partial E}{\partial \theta} = \frac{\partial E}{\partial out} \frac{\partial out}{\partial \theta}$
3. **update the parameters** using $\theta \leftarrow \theta - \eta \frac{\partial E}{\partial \theta}$



For recurrent networks, also propagate the error back in time (Werbos, 1990).