Fast and robust training of recurrent neural networks for offline handwriting recognition

Patrick Doetsch, Michal Kozielski, Hermann Ney

Human Language Technology and Pattern Recognition
Computer Science Department, RWTH Aachen University
D-52056 Aachen, Germany

Sep. 3, 2014

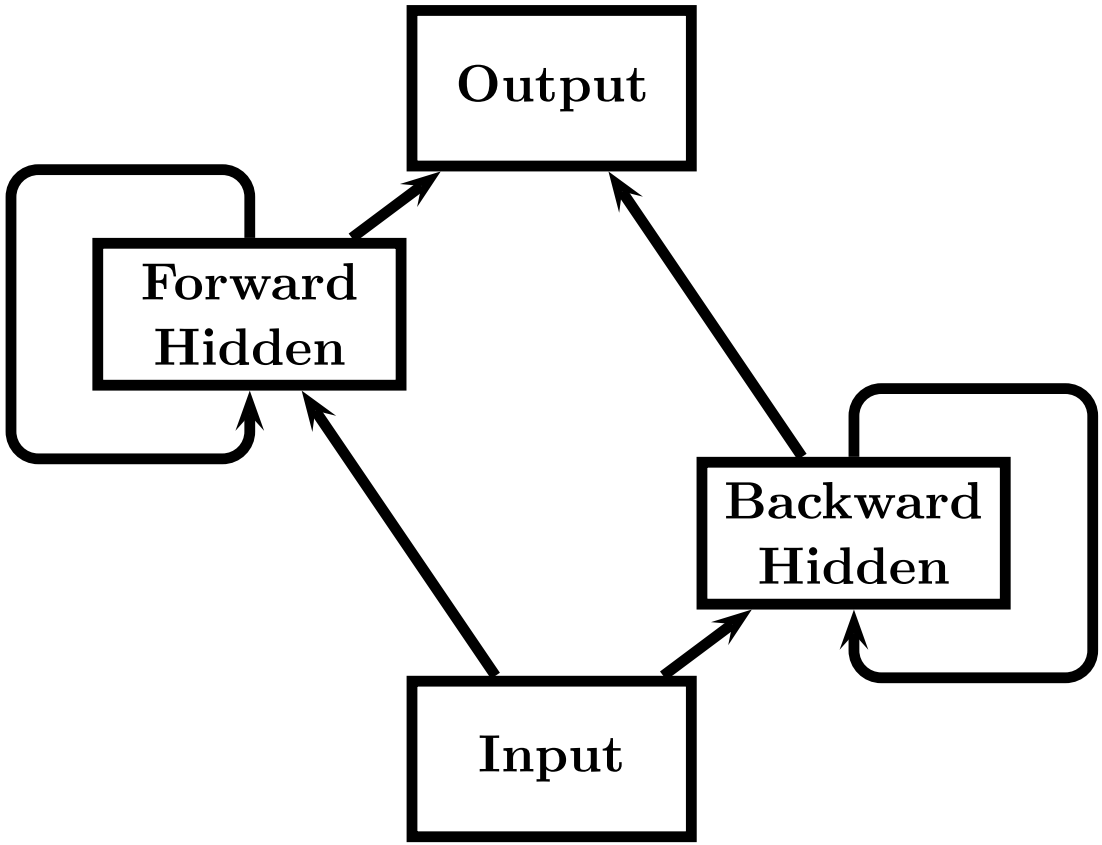
Outline

- Frame-wise bidirectional long-short term memory training
- Accelerating the training procedure
- Optimizing the shape of gating unit activations
- Experimental results

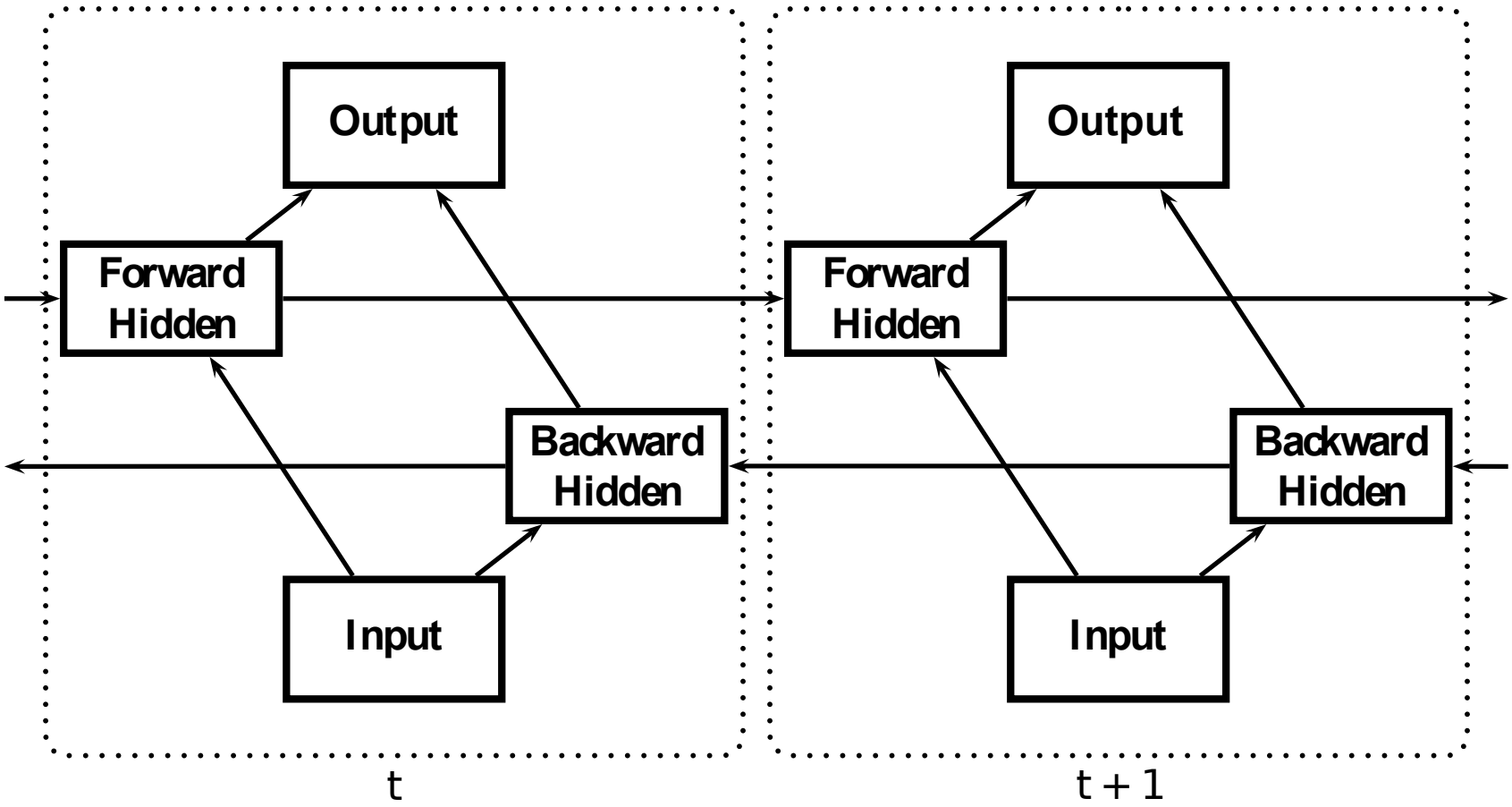
State-of-the-art

- Gradient flow in recurrent neural networks
  [Glorot and Bengio 2010].
- Long short-term memory for handwritten text recognition
  [Liwiki et al. 2007].
- LVSR with long short-term memories
  [Sak et al. 2014].
- Offline handwriting recognition with LSTM/HMM hybrids
  [Kozielski et al. 2013].

Bidirectional recurrent neural networks

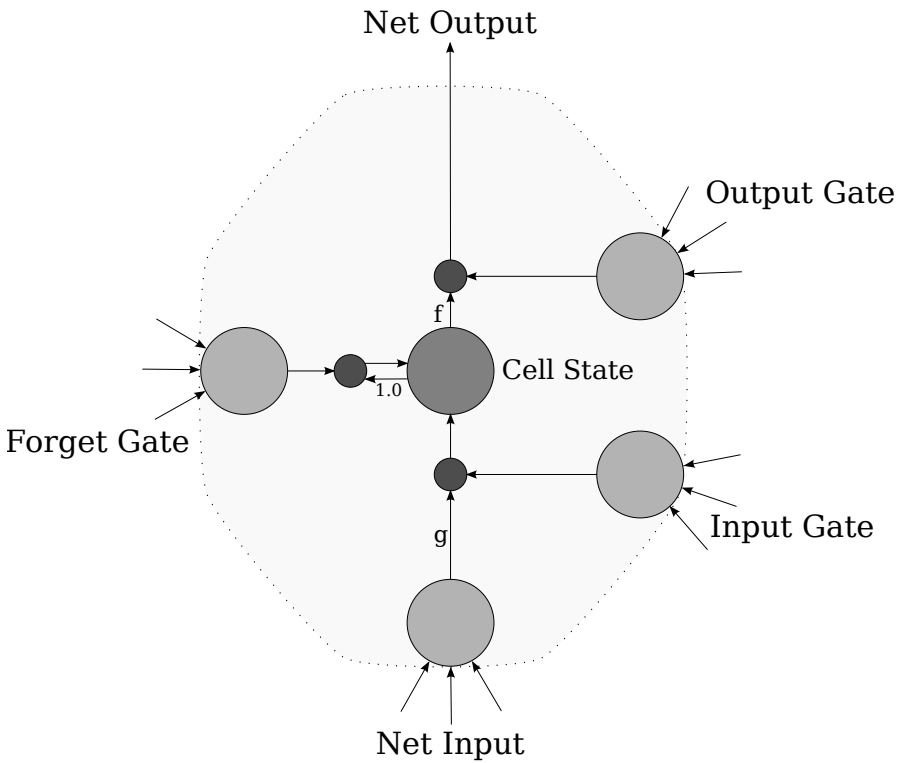


Backpropagation through time



Long-Short-Term-Memory

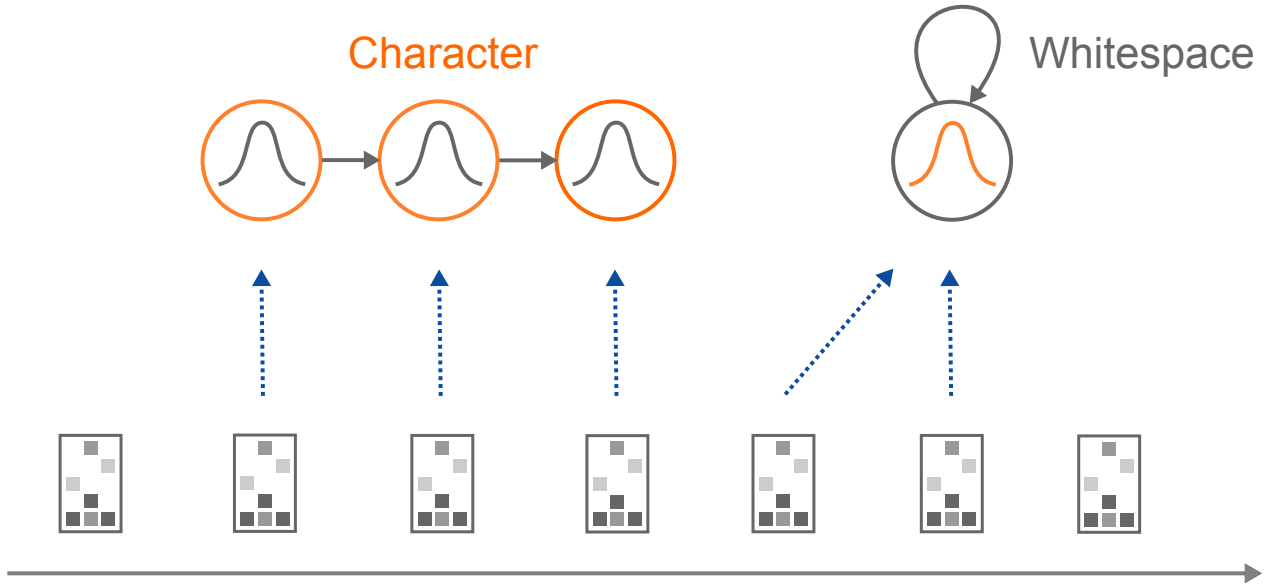
- Constant error flow without loss of short time lag capability
- Idea: Replace units by *memory cells*



- Input Gate:
  - Protects error flow inside cell from irrelevant inputs
- Output Gate:
  - Protects error flow of other cells from irrelevant inputs
- Forget Gate:
  - Provides a way to reset cell state

Hidden Markov model integration

Variable length HMM for each character

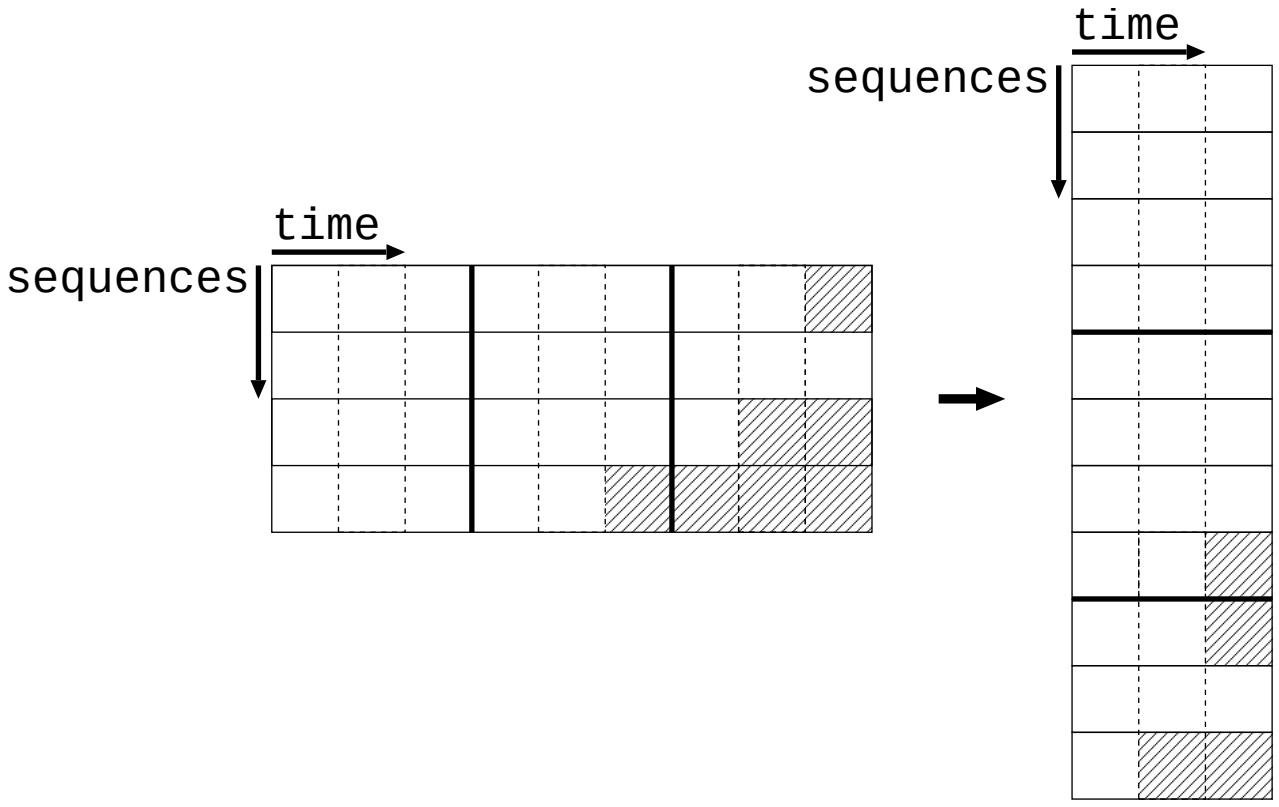


Simulate state emission probability by softmax outputs:

$$p(x|s) = \frac{p(s|x)}{p(s)^{\alpha}}$$

Accelerating BPTT training

- Cut sequences into chunks of constant length
- Combine chunks into large batches



- Training time on IAM using a NVIDIA GTX680 GPU

chunk size	chunks per batch	FER[%]	training time
10	4000	22.7	3.4 min
50	800	19.8	4.7 min
100	400	19.5	6.7 min
500	80	19.3	15 min
∞	$\approx$ 65	19.2	16.2 min

$\Rightarrow$ A chunk size of 100 seems to be a good trade-off

Gate Scaling

- Idea: Learn shape of activation function in LSTM gating units
- Scale soft thresholding of sigmodial activation functions

$$y_{\chi_j}^{(l)} = sig_{\chi}^{(l)}\left(z_{\chi_j}^{(l)}\right) = sig\left(\lambda_{\chi}^{(l)} z_{\chi_j}^{(l)}\right), \chi \in \{\iota, \phi, \omega\}$$

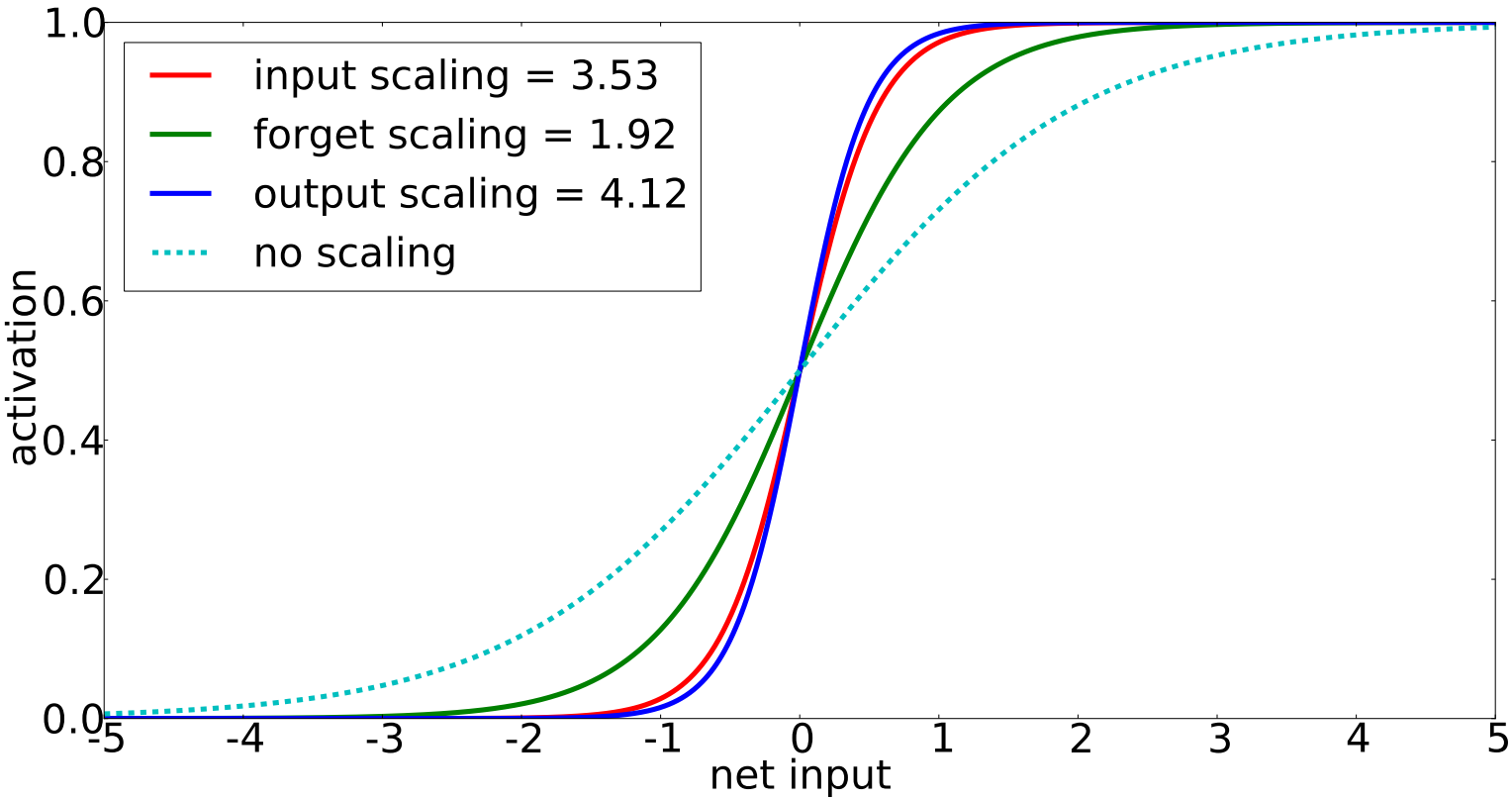
$$z_{\chi_j}^{(l)} = \sum_i \alpha_{\chi_j i}^{(l-1)} y_i^{(l-1)} + \sum_i \alpha_{\chi_j i}^{(l)} y_i^{(l)}$$

$\Rightarrow$ Trainable with gradient descent and backpropagation

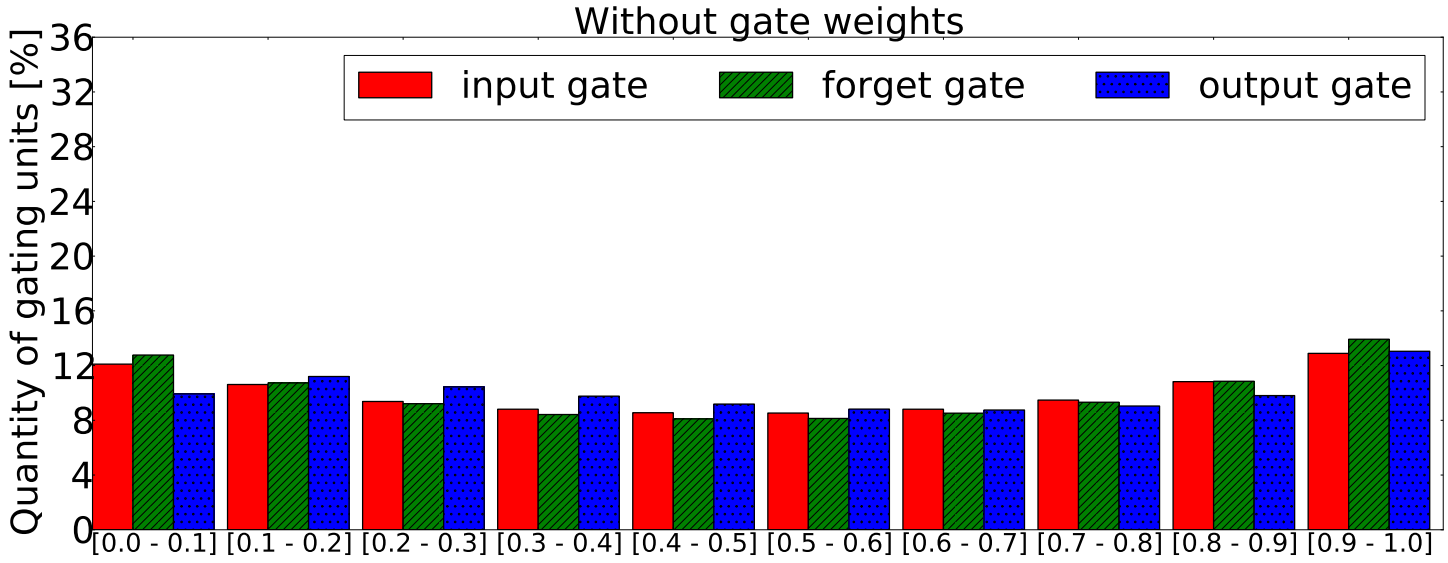
Network topology

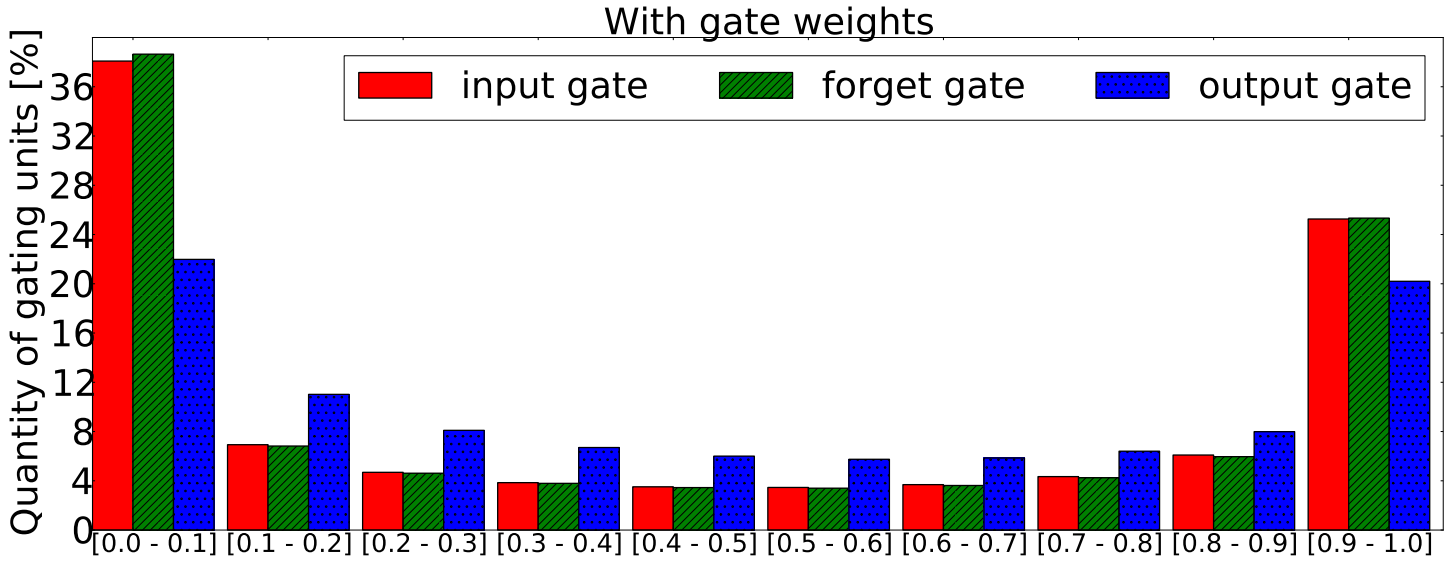
- 24-dimensional input vector
  - Sliding window of slant corrected / contrast normalized image
  - Moment-based image normalization and PCA reduction
- 3 hidden LSTM layers with 500 memory cells in each layer
  - For forward and backward directions respectively
- Softmax output layer with one unit for each state label
  - Cross-entropy minimization on frames labeled by HMM

Learned shapes of sigmoids



Distribution of saturated gating units





Databases

IAM

English handwritten text lines
747 train images, 116 test images
50k vocabulary size
3-gram word LM, PPL 420
10-gram character LM, PPL 3.7

RIMES

French handwritten text lines
1500 train images, 100 test images
3.7k closed vocabulary
4-gram LM, PPL 26

Experimental results

Systems	IAM		RIMES	
	WER [%]	CER [%]	WER [%]	CER [%]
GMM	10.7	3.8	15.7	5.5
LSTM-RNN / HMM (w/o gate weights)	8.9	2.8	13.3	4.6
LSTM-RNN / HMM (with gate weights)	8.4	2.5	12.9	4.3

Experimental results: Comparison on IAM

Systems	WER [%]		CER [%]	
	Dev.	Eval	Dev.	Eval
RWTH	8.4	12.2	2.5	4.7
Kozielski et al.	9.5	13.3	2.7	5.1
Boquera et al.	19.0	22.4	-	9.8
Dreuw et al.	22.7	32.9	7.7	12.4

Experimental results: Comparison on RIMES

Systems	WER [%]	CER [%]
RWTH	12.9	4.3
A2IA	12.6	3.5
Kozielski et al.	13.7	4.6
Telecom ParisTech	31.2	-

Conclusions

- Sequence chunking allows for fast training of large networks
  - 3x faster training without decrease in system performance
- Gate scaling lead to optimized shapes of activation functions
  - Gates switch very quickly between 0.0 and 1.0
- Substantial improvements obtained through gate scaling:
  - CER reduction from 2.8% to 2.5% and 4.6% to 4.3%
  - WER reduction from 8.9% to 8.4% and 13.3% to 12.9%

Thank you for your attention

doetsch@i6.informatik.rwth-aachen.de