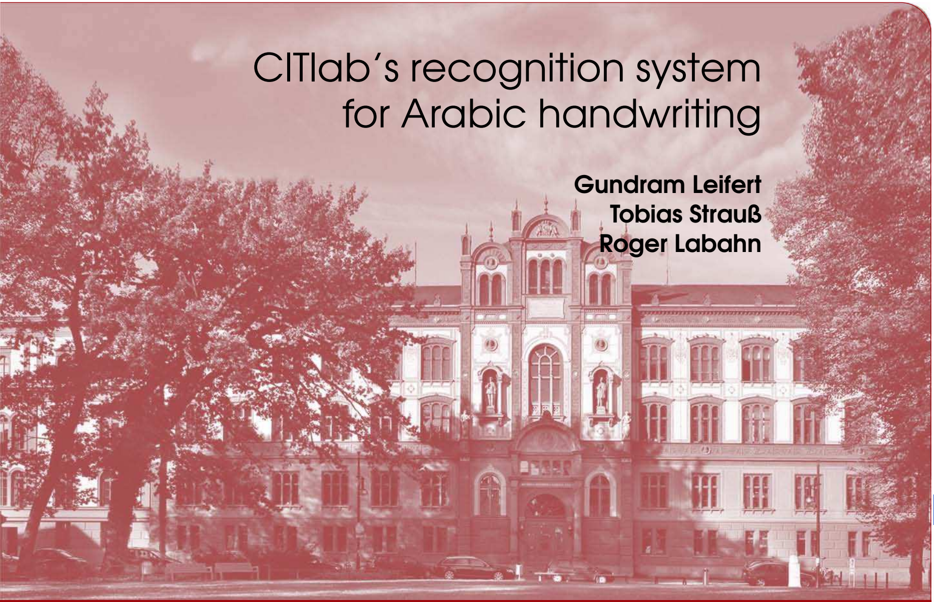
Universität Rostock Traditio et Innovatio

CITlab Computational Intelligence Technology



CITlab´s recognition system for Arabic handwriting

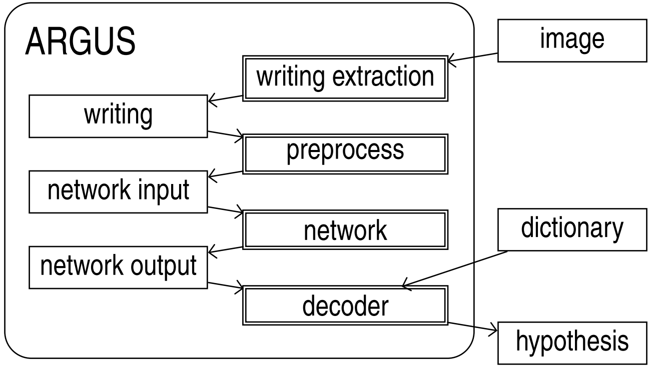
Gundram Leifert
Tobias Strauß
Roger Labahn

System description

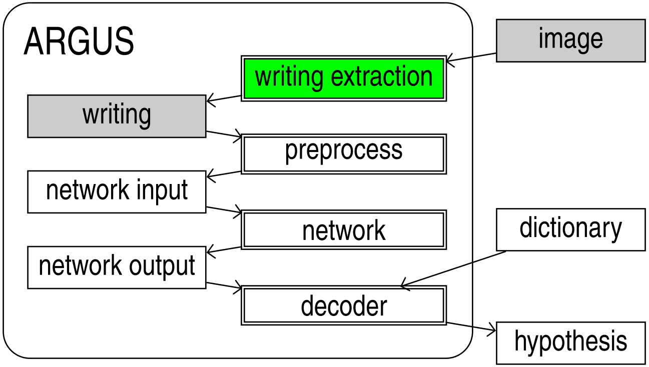
Results

Further experiments

Layout of the recognition system

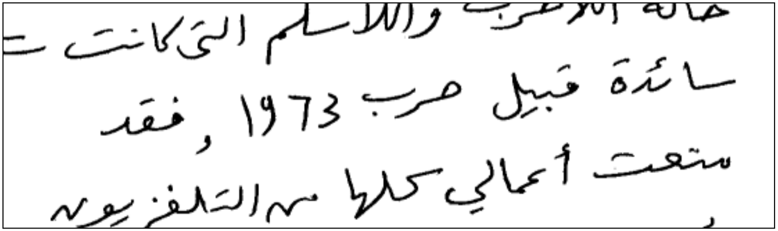


Layout of the recognition system



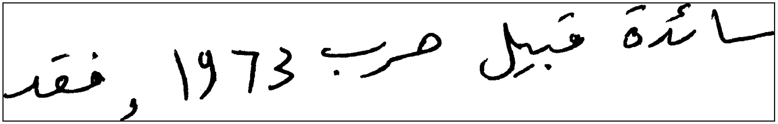
Writing extraction

Image (part of a page)



حالة اللاحرب واللاسلم التي كانت
سائدة قبيل حرب 1973، فقد
منعت أعمالي كلها من التلفزيون

Writing (extracted writing using the line polygon)

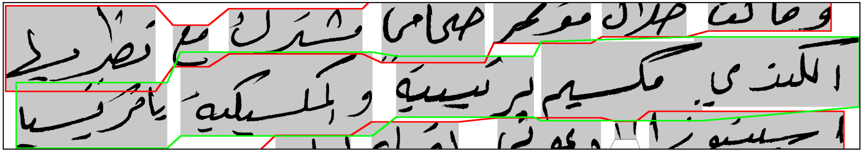


سائدة قبيل حرب 1973، فقد

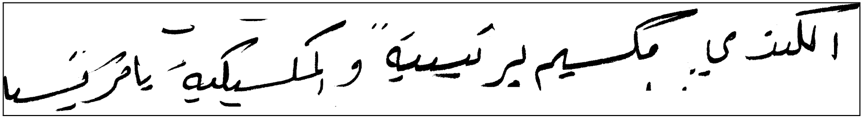
Difference between training and evaluation data

- The system processes entire lines of the images.
- The xml-files of the evaluation sets provide polygon around the lines.
- This polygon is not available for the other sets, so we had to construct it from the word's polygon.

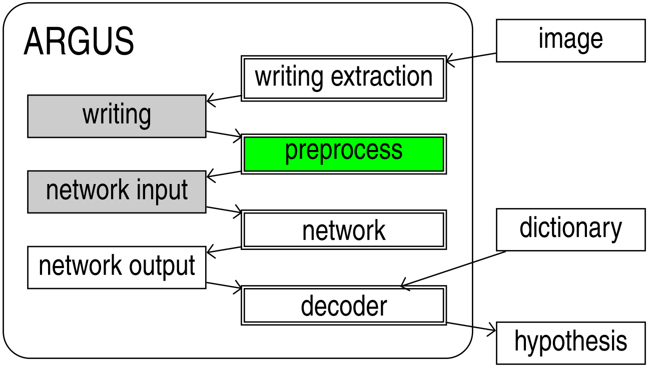
Word polygons (shaded) and generated line polygon (colored line).



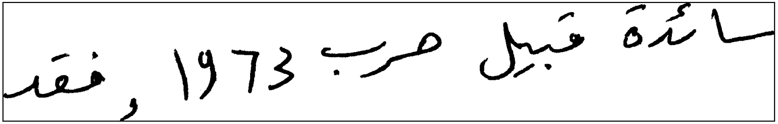
Extracted writing using the line polygon



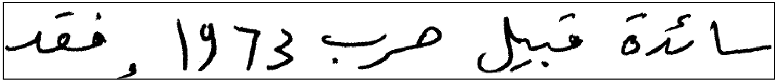
Layout of the recognition system



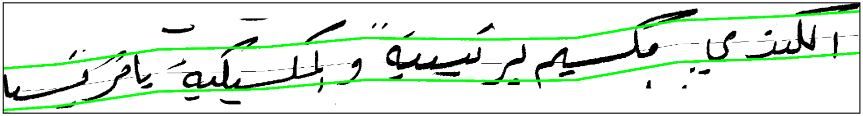
Extracted writing using the line polygon


سـائدة قبيل حرب 1973 , فقد

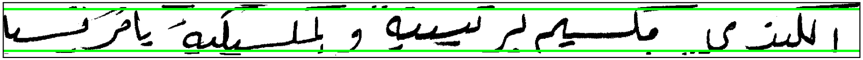
Preprocessed writing


سـائدة قبيل حرب 1973 , فقد

Locally calculated main body of the writing

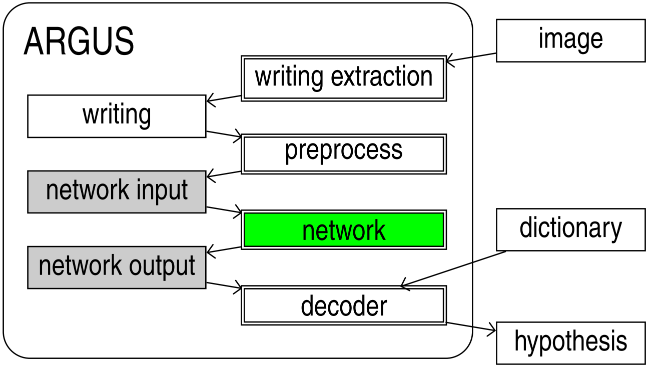


Shifted main body with shrinked ascenders and descenders

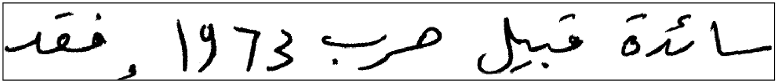


⇒ The neural network processes writing images of fixed height.

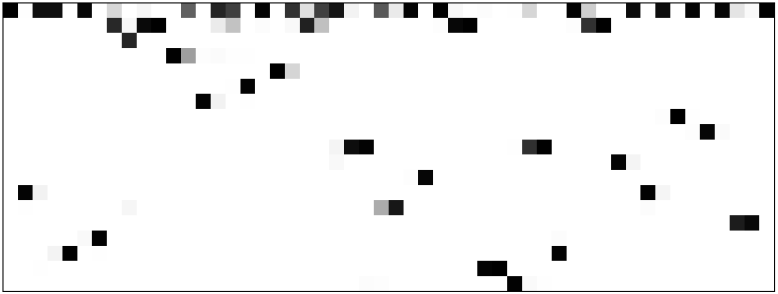
Layout of the recognition system



Preprocessed writing

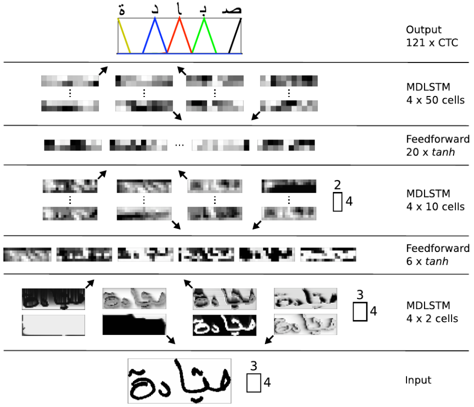


Character (rows) probabilities per position (columns)

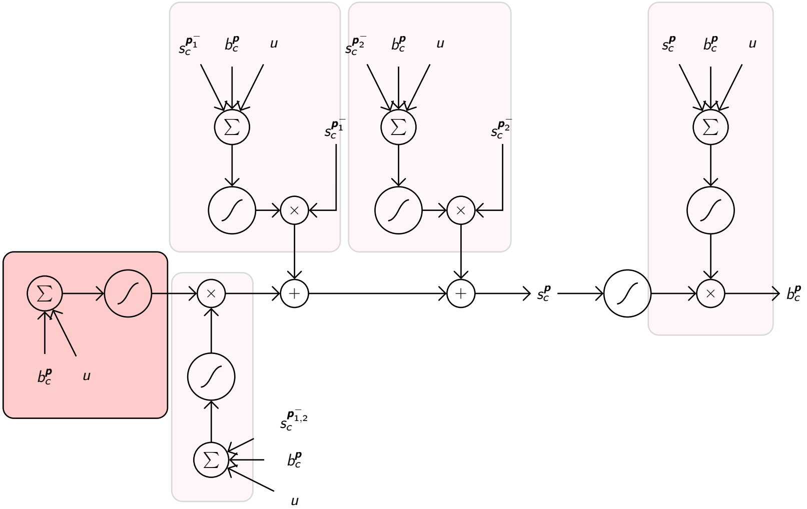


Network layout

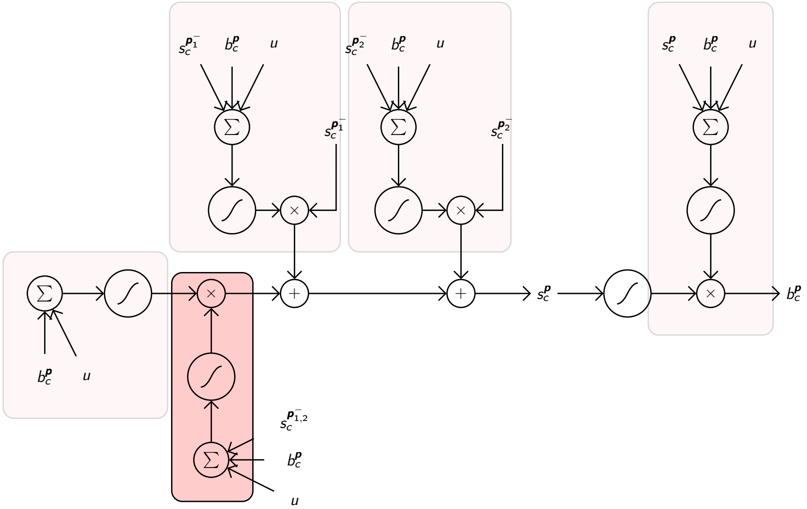
- The network is copied from [A. Graves and J. Schmidhuber, “*Offline handwriting recognition with multidimensional recurrent neural networks*”].
- Each hidden layer has 50% more units.



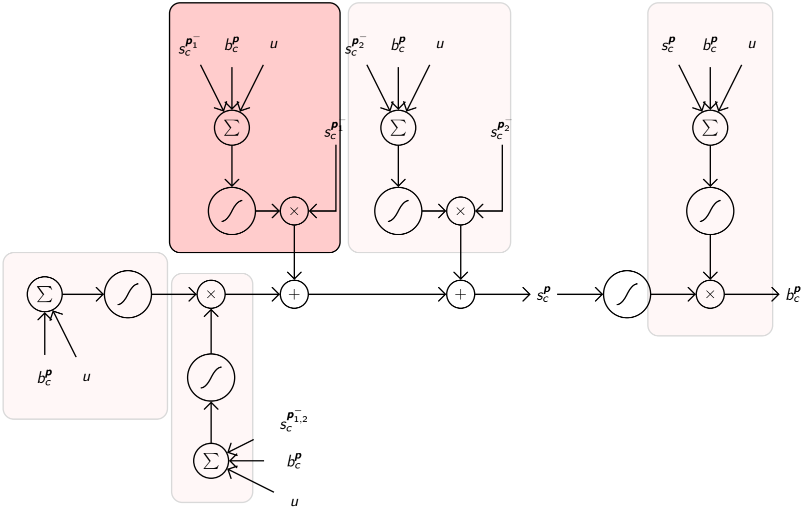
Block diagram of a 2D-LSTM cell



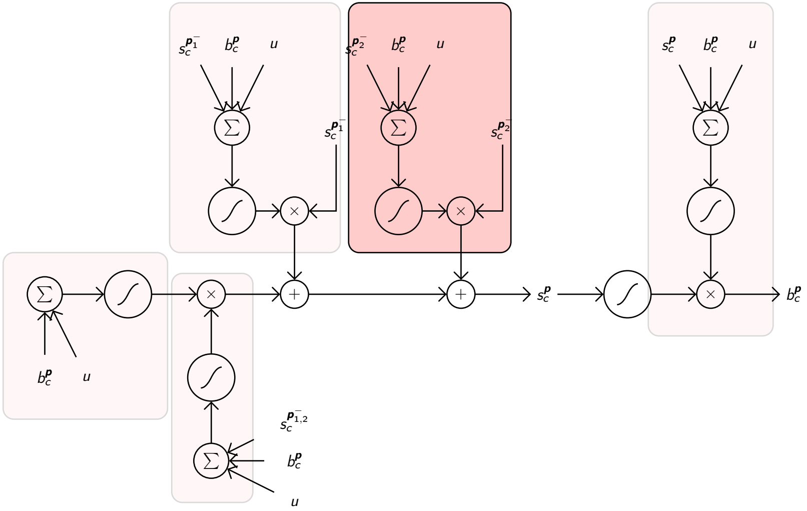
Block diagram of a 2D-LSTM cell



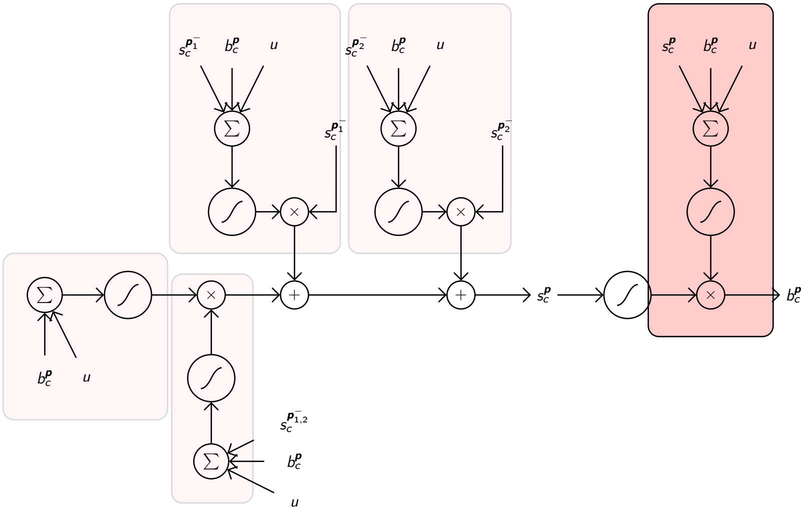
Block diagram of a 2D-LSTM cell



Block diagram of a 2D-LSTM cell



Block diagram of a 2D-LSTM cell



Layout of a multidimensional Leaky cell

- Reduce the D previous states $s_c^{\mathbf{p}_i^-}$ $i = 1, \ldots, D$ to one previous state $s_c^{\mathbf{p}^-}$ by convex combination

$$s_c^{\mathbf{p}^-} = \sum_{d=1}^{D} s_c^{\mathbf{p}_d^-} b_{\lambda,d}^{\mathbf{p}} \quad , \quad b_{\lambda,d}^{\mathbf{p}} \geq 0, \sum_{d=1}^{D} b_{\lambda,d}^{\mathbf{p}} = 1$$

- Calculate the current internal state $s_c^{\mathbf{p}}$ as convex combination of the single previous state $s_c^{\mathbf{p}^-}$ and the new input $u_c^{\mathbf{p}}$

$$s_c^{\mathbf{p}} = \left(1 - b_\phi^{\mathbf{p}}\right) u_c^{\mathbf{p}} + b_\phi^{\mathbf{p}} s_c^{\mathbf{p}^-} \quad , \quad b_\phi^{\mathbf{p}} \in [0, 1]$$

- Calculate the output $b_c^{\mathbf{p}}$ as weighted sum of the previous state $s_c^{\mathbf{p}^-}$ and the current internal state $s_c^{\mathbf{p}}$, and squash it by $\tanh(\cdot)$

$$b_c^{\mathbf{p}} = \tanh\left(b_{\omega_0}^{\mathbf{p}} s_c^{\mathbf{p}} + b_{\omega_1}^{\mathbf{p}} s_c^{\mathbf{p}^-}\right) \quad , \quad b_{\omega_0}^{\mathbf{p}}, b_{\omega_1}^{\mathbf{p}} \in [0, 1]$$

Layout of a multidimensional Leaky cell

- Reduce the D previous states $s_c^{\mathbf{p}_i^-}$ $i = 1, \dots, D$ to one previous state $s_c^{\mathbf{p}^-}$ by convex combination

$$s_c^{\mathbf{p}^-} = \sum_{d=1}^{D} s_c^{\mathbf{p}_d^-} b_{\lambda,d}^{\mathbf{p}} \quad , \quad b_{\lambda,d}^{\mathbf{p}} \geq 0, \sum_{d=1}^{D} b_{\lambda,d}^{\mathbf{p}} = 1$$

- Calculate the current internal state $s_c^{\mathbf{p}}$ as convex combination of the single previous state $s_c^{\mathbf{p}^-}$ and the new input $u_c^{\mathbf{p}}$

$$s_c^{\mathbf{p}} = \left(1 - b_\phi^{\mathbf{p}}\right) u_c^{\mathbf{p}} + b_\phi^{\mathbf{p}} s_c^{\mathbf{p}^-} \quad , \quad b_\phi^{\mathbf{p}} \in [0,1]$$

- Calculate the output $b_c^{\mathbf{p}}$ as weighted sum of the previous state $s_c^{\mathbf{p}^-}$ and the current internal state $s_c^{\mathbf{p}}$, and squash it by $\tanh(\cdot)$

$$b_c^{\mathbf{p}} = \tanh\left(b_{\omega_0}^{\mathbf{p}} s_c^{\mathbf{p}} + b_{\omega_1}^{\mathbf{p}} s_c^{\mathbf{p}^-}\right) \quad , \quad b_{\omega_0}^{\mathbf{p}}, b_{\omega_1}^{\mathbf{p}} \in [0,1]$$

Layout of a multidimensional Leaky cell

- Reduce the D previous states $s_c^{\mathbf{p}_i^-}$ $i = 1, \ldots, D$ to one previous state $s_c^{\mathbf{p}^-}$ by convex combination

$$s_c^{\mathbf{p}^-} = \sum_{d=1}^{D} s_c^{\mathbf{p}_d^-} b_{\lambda,d}^{\mathbf{p}} \quad , \quad b_{\lambda,d}^{\mathbf{p}} \geq 0, \sum_{d=1}^{D} b_{\lambda,d}^{\mathbf{p}} = 1$$

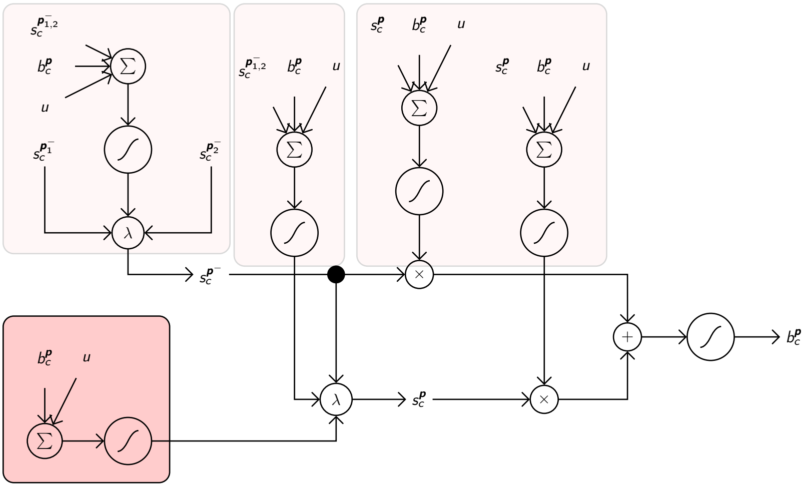
- Calculate the current internal state $s_c^{\mathbf{p}}$ as convex combination of the single previous state $s_c^{\mathbf{p}^-}$ and the new input $u_c^{\mathbf{p}}$

$$s_c^{\mathbf{p}} = \left(1 - b_\phi^{\mathbf{p}}\right) u_c^{\mathbf{p}} + b_\phi^{\mathbf{p}} s_c^{\mathbf{p}^-} \quad , \quad b_\phi^{\mathbf{p}} \in [0, 1]$$

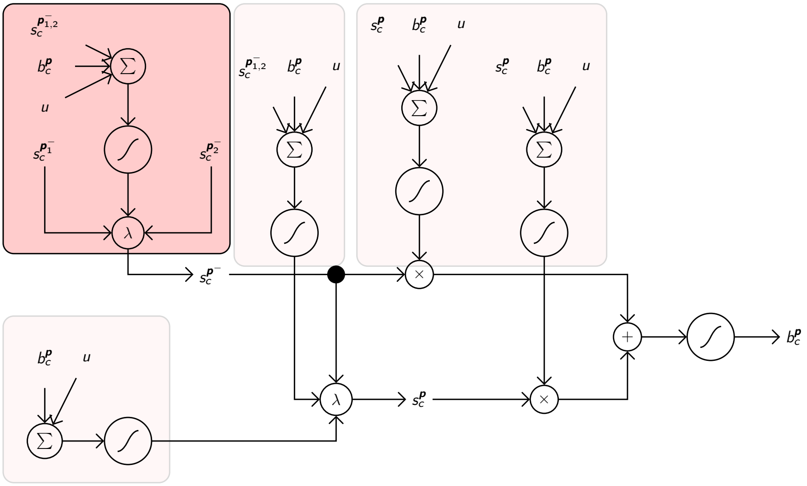
- Calculate the output $b_c^{\mathbf{p}}$ as weighted sum of the previous state $s_c^{\mathbf{p}^-}$ and the current internal state $s_c^{\mathbf{p}}$, and squash it by $\mathtt{tanh}(\cdot)$

$$b_c^{\mathbf{p}} = \mathtt{tanh}\left(b_{\omega_0}^{\mathbf{p}} s_c^{\mathbf{p}} + b_{\omega_1}^{\mathbf{p}} s_c^{\mathbf{p}^-}\right) \quad , \quad b_{\omega_0}^{\mathbf{p}}, b_{\omega_1}^{\mathbf{p}} \in [0, 1]$$

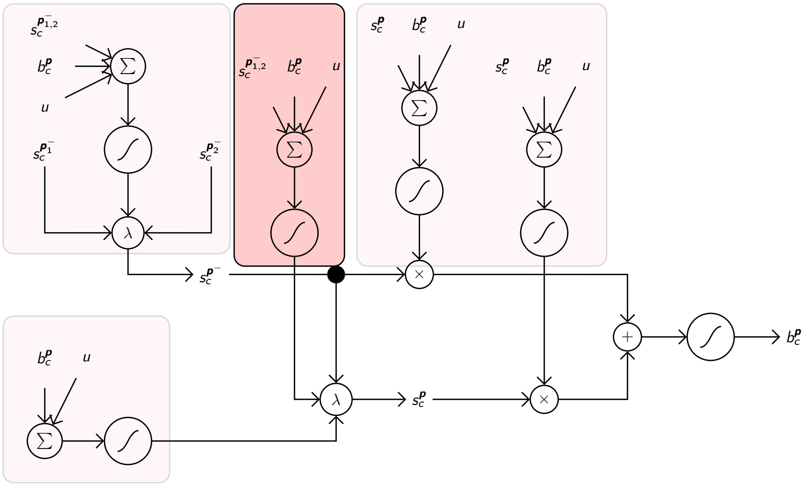
Block diagram of a 2D-Leaky cell



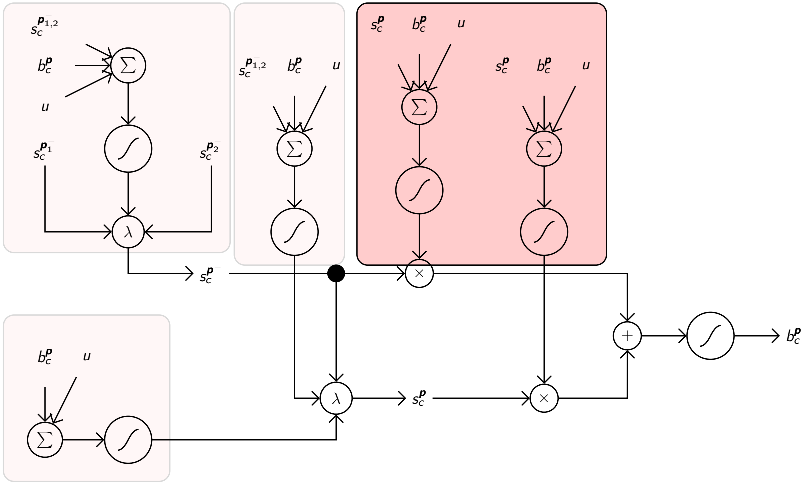
Block diagram of a 2D-Leaky cell



Block diagram of a 2D-Leaky cell



Block diagram of a 2D-Leaky cell



The Character set with 153 characters

- Arabic letters (46)

  آ أ ؤ إ ئ ا ب ة ت ث ج ح خ د ذ ر ز س ش ص ض ط ظ ع غ ـ ف ق ك ل م ن ه و ى ي ً ٌ ٍ َ ِ ّ ے إ لإ

- Latin letters (53)

  A B C D E F G H I J K L M N O P Q R S T U V W X Z a b c d e f g h i j k l m n o p q r s t u v w x y z è ì

- Digits (10)

  0 1 2 3 4 5 6 7 8 9

- Signs (43) including space character and extra characters for “..” and “...”

  ! " % & ' ( ) * + , - . / : ; < = > ? @ [\ ] ^ _ { } ~ © «   · » × ، ؟ ؛ ° ٪ .. ... ●

- the “blank” character

Training setup

– The network is trained with Backpropagation-Through-Time (BPTT) using the Connectionist Temporal Classification (CTC) algorithm desribed in [A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, "*Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural network*"].

Training setup

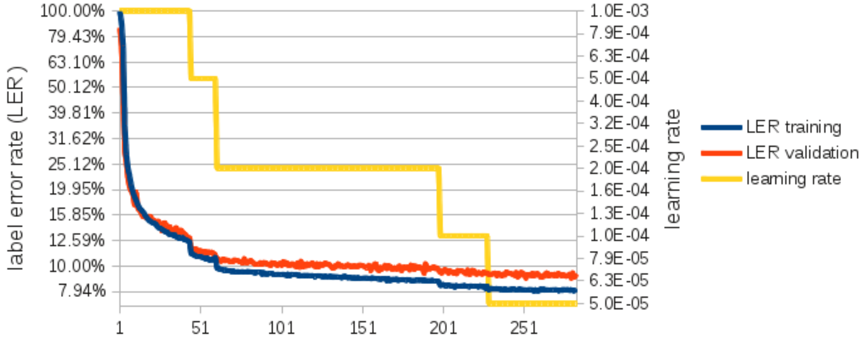
- Let $(x, y) \in S$ be an input-target pair of a training set S, where $x \in [0, 1]^{n \times m}$ is the network input and $y \in L^k$ is a sequence of the character set L of length k, which represent the text in x.
- For one input-target pair (x, y) we maximize the probability of the target sequence y for a given input x, by reducing its logarithmic probability.

$$\mathcal{L}(x, y) = -\ln p(y|x)$$

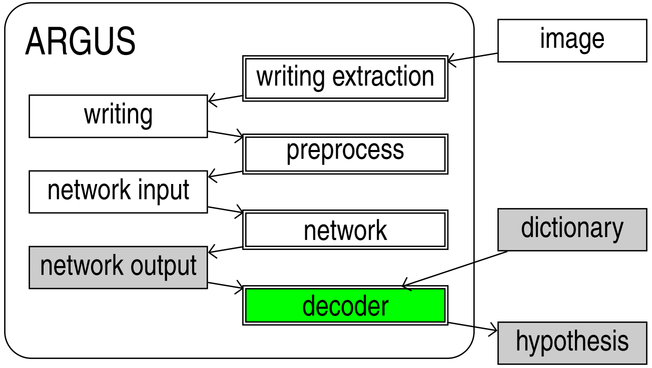
Training setup

- One single line x of a page with the associated target sequence y is one training item (x, y).
- One training epoch consists of one randomly chosen line from each of the 27,915 pictures of the MADCAT Phase 2 training set.
- One validation epoch consists of one randomly chosen line from each of the 4,540 pictures of the MADCAT Phase 3 training set.
- The learning rate is reduced from $1 \cdot 10^{-3}$ to $5 \cdot 10^{-5}$ over 283 epochs with momentum 0.9.

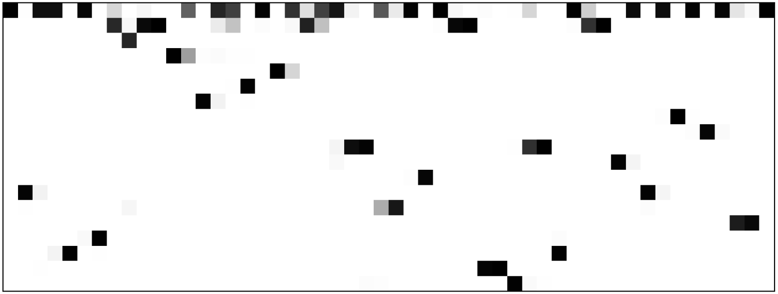
Training of the primary system



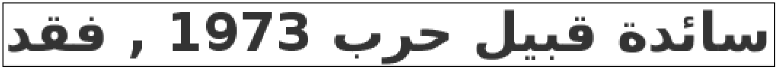
Layout of the recognition system



Character (rows) probabilities per position (columns)



Most probable for a given network output matrix



سائدة قبيل حرب 1973 , فقد

Dictionary lookup

- For the hypothesis string we use the most probable sequence which arises by the output of the network, using the CTC-algorithm.
- For improving the recognition rate by a dictionary lookup, we extract a dictionary.

Dictionary extraction

- We take a specific set of MADCAT xml-files provided for OpenHaRT 2013.
- We count the occurrences of the <token>'s <source> content, which contain only Arabic letters, including those of status “TYPO” or “MISSING”.
- If this is lower than a specific number, we assume it is a typo and we erase the entry from dictionary.
- For the primary system, we took the xml-files of MADCAT Phase 1-3 Training Set and Phase 1 Evaluation Set.
- This dictionary contains $107,059$ entries.

Parsing the network output

- For a given network output, we calculate the most probable sequence of entries of the dictionary, using CTC.
- If the average character probability over the best dictionary entry falls below a constant threshold θ, we assume that the true word is not in dictionary.
- If so, we directly take the most probable output sequence of the network.
- As default, we use $\theta = \frac{1}{e}$, but also tried the larger value $\theta = \frac{1}{\sqrt{e}}$.

Main results

decoder	difference to primary system	WER	Δ WER
#5		26.27	
#1	no dictionary	33.14	+6.87
#2	dictionary sources include Dryrun Set	26.31	+0.04
#3	enlarged $\theta = \frac{1}{\sqrt{e}}$	24.60	-1.67
#6	dictionary's words appear at least 3 times	25.35	-0.92
#4	combining decoders #2 and #6	25.18	-1.09

Main results - primary network

WER on the evaluation set in %			
		dictionary's words appear at least 3 times	
		no	yes
θ	$\frac{1}{e}$	26.27	25.35
	$\frac{1}{\sqrt{e}}$	24.60	23.32

Unsupervised pretraining

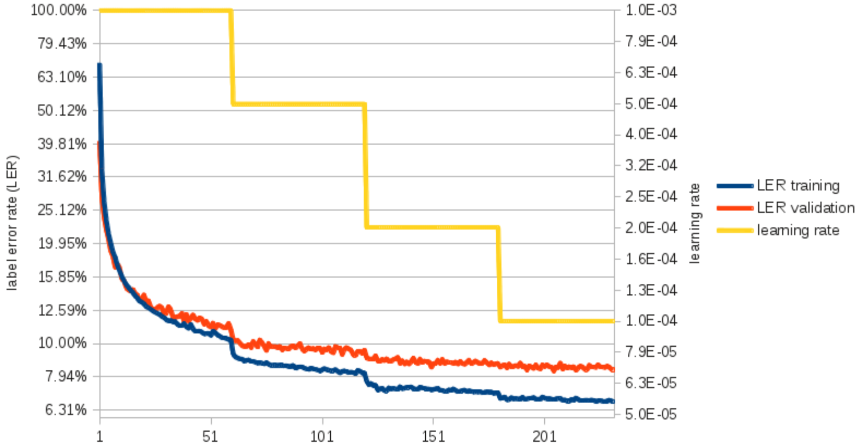
– Unsupervised pretraining improves many deep networks or makes it even possible to train deep architectures.

Neural network layout with pretrained features

– The lowest MD-layer is substituted by a deep believe net (DBN).

$\rightarrow$ Neurons in the `tanh`-layer have 250 instead of 144 source connections.

Training of the unsupervised neural network



Main results - unsupervised pretrained neural network

WER on the evaluation set in %			
		dictionary's words appear at least 3 times	
		no	yes
θ	$\frac{1}{e}$	24.00	23.08
	$\frac{1}{\sqrt{e}}$	22.42	21.75

Conclusion

decoder	difference to primary system	WER	Δ WER
#5		26.27	
#3	enlarged $\theta = \frac{1}{\sqrt{e}}$	24.60	-1.67
#6	dictionary's words appear at least 3 times	25.35	-0.92
	combining #3, #6	23.32	-2.95
	using classical LSTM cells	27.62	1.35
	using unsupervised features	24.00	-2.27
	combining #3, #6 and unsupervised features	21.75	-4.52

Conclusion

decoder	difference to primary system	WER	Δ WER
#5		26.27	
#3	enlarged $\theta = \frac{1}{\sqrt{e}}$	24.60	-1.67
#6	dictionary's words appear at least 3 times	25.35	-0.92
	combining #3, #6	23.32	-2.95
	using classical LSTM cells	27.62	1.35
	using unsupervised features	24.00	-2.27
	combining #3, #6 and unsupervised features	21.75	-4.52

Thanks for attention!