A new Approach for Cells in Multidimensional Recurrent Neural Networks

Gundram Leifert Tobias Strauß Roger Labahn Welf Wustlich

December 21, 2012

Abstract

A recent approach for offline handwriting recognition is to use multidimensional recurrent neural networks (MDRNN) with connectionist temporal classification which has shown to yield very good results on several datasets. MDRNNs contain special units – multidimensional Long Short-Term Memory (MDLSTM) cells. These cells suffer from instability especially for higher dimensionality. We analyze the reasons for this effect and introduce several cells with better stability. We present a method to design stable multidimensional cells using the theory of linear shift invariant systems. The new cells are compared to MDLSTMs on the Arabic and French ICDAR datasets, where they yield better results.

1 notations

1.1 unit

Normal hidden units in a time discrete recurrent neural networks (RNN) update their new accumulation $a(t)$ and activation $b(t)$ to time $t \in \mathbb{Z}$ according to equations

$$a_u(t) = \sum_{h=1}^{H} w_{h,u} b_h(t-1) + \sum_{i=1}^{I} w_{i,u}^{in} b_i(t), \tag{1}$$

$$b_u(t) = f\left(a_u(t)\right). \tag{2}$$

To extend the RNN to a multidimensional RNN (MDRNN) let $\boldsymbol{p} \in \mathbb{Z}^D$ be a multidimensional date of dimension D. Instead of $a(t)$ in 1-dimensional case we write $a^{\boldsymbol{p}}$ as accumulation in the multidimensional case. The upper index $\boldsymbol{p} = (p_1, p_2, \ldots, p_D)$ with $p_i \in \{1, 2, \ldots, D_i^{\max}\}$ will be used to define the position. The upper index $\boldsymbol{p}_d^-$ denotes the position on step back in dimension d. So $\boldsymbol{p}_d^- = (p_1, \ldots, p_{d-1}, p_d - 1, p_{d+1}, \ldots, p_D)$. In the same way $\boldsymbol{p}_d^+$ is defined. If there is at least one dimension $d \in \{1, \ldots, D\}$ with $p_d \notin \{1, \ldots, D_d^{\max}\}$ we set the state and activation to 0. The connection from source unit to target unit is denoted with $w_{[target][source]}$. When the weight is from source unit in the past in dimension d to a target unit, we denote the weight with $w^d_{[target][source]}$. We assume a hidden layer with H units, I units in the layer below and K units in the Layer above. Similar to (1) we can calculated the accumulation and activation in a multidimensional case.

1.1.1 Forward pass

$$a_u^{\boldsymbol{p}} = \sum_{i=1}^{I} w_{u,i} b_i + \sum_{d=1}^{D} \sum_{h=1}^{H} w_{u,h}^d b_h^{\boldsymbol{p}_d^-}, \tag{3}$$

$$b_u^{\boldsymbol{p}} = f_u(a_u^{\boldsymbol{p}}). \tag{4}$$

1.1.2 Backward pass

Let

$$\varepsilon_u^{\boldsymbol{p}} := \frac{\partial E}{\partial b_u^{\boldsymbol{p}}} \tag{5}$$

be the error of the output of neuron u at time $\boldsymbol{p}$ and let

$$\delta_u^{\boldsymbol{p}} = \frac{\partial E}{\partial a_u^{\boldsymbol{p}}} \tag{6}$$

be the error after the accumulation. Let K be the set of errors from the upper layer yield from the units where unit u is connected with. Then they can be calculated by

$$\varepsilon_u^{\boldsymbol{p}} = \sum_{k=1}^{K} \delta_k^{\boldsymbol{p}} w_{u,k} + \sum_{d=1}^{D} \sum_{h=1}^{H} w_{h,u}^d \delta_h^{\boldsymbol{p}_d^+} \tag{7}$$

$$\delta_u^{\boldsymbol{p}} = f_u'(a_u^{\boldsymbol{p}}) \varepsilon_u^{\boldsymbol{p}} \tag{8}$$

1.2 cell

A cell has input connections and a activation function like a unit, but furthermore, it has a set of "gates". The gates nearly work like units with the feature, they can have a connection to the internal state of the

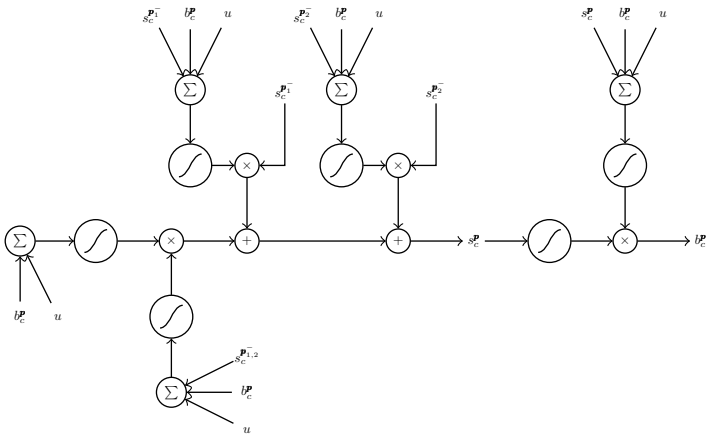


Figure 1: schematic diagram of a MDLSTM cell

cell, called peephole connections. The cell combines the internal state and the activations of the gate in an (hopefully) intelligent way, so that they can solve more difficult tasks than a normal unit. c denotes the internal state of a cell.

The LSTM cell has an input gate (IG), an output gate (OG) and D forget gates (FG). Let ι be the IG, (ϕ, d) be the FG of dimension d and ω be the OG. All activation functions of gates are the logistic function

$$f_\iota(x) = f_\omega(x) = f_{\phi,d}(x) = \frac{1}{1+e^{-x}}, \tag{9}$$

the activation functions of the cell are the tanh-function

$$f_c(x) = g_c(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{10}$$

1.2.1 forward pass

cell input:

$$a_c^{\boldsymbol{p}} = \sum_{i=1}^{I} w_{c,i} b_i^{\boldsymbol{p}} + \sum_{d=1}^{D} \sum_{h=1}^{H} w_{c,h}^d b_h^{\boldsymbol{p}_d^-} \tag{11}$$

$$u_c^{\boldsymbol{p}} = f_c(a_c^{\boldsymbol{p}}) \tag{12}$$

input gate:

$$a_\iota^{\boldsymbol{p}} = \sum_{i=1}^{I} w_{\iota,i} b_i^{\boldsymbol{p}} + \sum_{d=1}^{D} \left(w_{\iota,c}^d s_c^{\boldsymbol{p}_d^-} + \sum_{h=1}^{H} w_{\iota,h}^d b_h^{\boldsymbol{p}_d^-} \right) \tag{13}$$

$$b_\iota^{\boldsymbol{p}} = f_\iota(a_\iota^{\boldsymbol{p}}) \tag{14}$$

forget gates:

$$a_{\phi,d}^{\boldsymbol{p}} = w_{(\phi,d),c} s_c^{\boldsymbol{p}_d^-} + \sum_{i=1}^{I} w_{(\phi,d),i} b_i^{\boldsymbol{p}} + \sum_{d'=1}^{D} \left(\sum_{h=1}^{H} w_{(\phi,d),h}^{d'} b_h^{\boldsymbol{p}_{d'}^-} \right) \tag{15}$$

$$u_{\phi,d}^{\boldsymbol{p}} = f_{\phi,d}(a_{\phi,d}^{\boldsymbol{p}}) \tag{16}$$

cell state:

$$s_c^{\boldsymbol{p}} = b_\iota^{\boldsymbol{p}} u_c^{\boldsymbol{p}} + \sum_{d=1}^{D} s_c^{\boldsymbol{p}_d^-} b_{\phi,d}^{\boldsymbol{p}} \tag{17}$$

output gate:

$$a_\omega^{\boldsymbol{p}} = \sum_{i=1}^{I} w_{\omega,i} b_i^{\boldsymbol{p}} + w_{\omega,c} s_c^{\boldsymbol{p}} + \sum_{d=1}^{D} \sum_{h=1}^{H} w_{\omega,h}^d b_h^{\boldsymbol{p}_d^-} \tag{18}$$

$$b_\omega^{\boldsymbol{p}} = f_\omega(a_\omega^{\boldsymbol{p}}) \tag{19}$$

cell output:

$$b_c^{\boldsymbol{p}} = b_\omega^{\boldsymbol{p}} g_c(s_c^{\boldsymbol{p}}) \tag{20}$$

1.2.2 backward pass

cell output:

$$\varepsilon_c^{\boldsymbol{p}} = \frac{\partial E}{\partial b_c^{\boldsymbol{p}}} = \sum_{k=1}^{K} w_{k,c} \delta_k^{\boldsymbol{p}} + \sum_{d=1}^{D} \sum_{h=1}^{H} \left(w_{h,c}^d \delta_h^{\boldsymbol{p}_d^+} + w_{\iota,c}^d \delta_\iota^{\boldsymbol{p}_d^+} + w_{\omega,c}^d \delta_\omega^{\boldsymbol{p}_d^+} + \sum_{d'=1}^{D} w_{(\phi,d'),c}^d \delta_{\phi,d'}^{\boldsymbol{p}_d^+} \right) \tag{21}$$

output gate:

$$\varepsilon_\omega^{\boldsymbol{p}} = \frac{\partial E}{\partial b_\omega^{\boldsymbol{p}}} = \varepsilon_c^{\boldsymbol{p}} g(s_c^{\boldsymbol{p}}) \tag{22}$$

$$\delta_\omega^{\boldsymbol{p}} = \frac{\partial E}{\partial a_\omega^{\boldsymbol{p}}} = f'_\omega(a_\omega^{\boldsymbol{p}}) \varepsilon_\omega^{\boldsymbol{p}} \tag{23}$$

cell state:

$$\varepsilon_s^{\boldsymbol{p}} = \frac{\partial E}{\partial s_c^{\boldsymbol{p}}} = b_\omega^{\boldsymbol{p}} g'_c(s_c^{\boldsymbol{p}}) \varepsilon_c^{\boldsymbol{p}} + \delta_\omega^{\boldsymbol{p}} w_{\omega,c} + \sum_{d=1}^{D} \varepsilon_s^{\boldsymbol{p}_d^+} b_{\phi,d}^{\boldsymbol{p}_d^+} + w_{\iota,c}^d \delta_\iota^{\boldsymbol{p}_d^+} + w_{(\phi,d),c} \delta_{\phi,d}^{\boldsymbol{p}_d^+} \tag{24}$$

forget gates:

$$\varepsilon_{\phi,d}^{\boldsymbol{p}} = s_c^{\boldsymbol{p}_d^-} \varepsilon_s^{\boldsymbol{p}} \tag{25}$$

$$\delta_{\phi,d}^{\boldsymbol{p}} = f'_{\phi,d}(a_{\phi,d}^{\boldsymbol{p}}) \varepsilon_{\phi,d}^{\boldsymbol{p}} \tag{26}$$

input gates:

$$\varepsilon_\iota^{\boldsymbol{p}} = u_c^{\boldsymbol{p}} \varepsilon_s^{\boldsymbol{p}} \tag{27}$$

$$\delta_\iota^{\boldsymbol{p}} = f'_\iota(a_\iota^{\boldsymbol{p}}) \varepsilon_\iota^{\boldsymbol{p}} \tag{28}$$

cell input:

$$\delta_c^{\boldsymbol{p}} = f'_c(a_c^{\boldsymbol{p}}) b_\iota^{\boldsymbol{p}} \varepsilon_s^{\boldsymbol{p}} \tag{29}$$

2 LSTM-Problems

The original LSTM cell was created for the one-dimensional case. It just has an input gate and an output gate. The new cell state is calculate by

$$s_c^{\boldsymbol{p}} = s_c^{p^-} + b_\iota^{\boldsymbol{p}} b_c^{\boldsymbol{p}}. \tag{30}$$

Assuming a constantly positive input $a_c^{\boldsymbol{p}}$, the cell state increases to infinity. Obviously, if $|s_c^{\boldsymbol{p}^-}| \gg |b_i|$, b_i has only a small contribution to the current cell state $s_c^{\boldsymbol{p}}$ and therefore also to the cell output. To avoid this excessively influence on the cell output the forget gate was introduced aiming to decrease the contribution of the cell state if necessary. Therefore, the forget gate takes values from the interval $(0, 1)$ and is multiplied by the past cell state. Note that an upper bound is not guaranteed, especially, if the forget gate is continuously close to 1.

The D-dimensional case, the cell state already might increase (in relation to the average of past cell states) if the forget gate has an activation greater or equal to $\frac{1}{D}$. In contrast to the one-dimensional case, the cells can exhibit self-reinforcing tendencies due to the sum of different directions. To see this, we provide a little toy example:

Example 1. Assume just one cell with recurrent connections in D dimensions. For great $s_c^{\boldsymbol{p}_d^-}$ and positive $w_{(\phi,d),c}$, the cell state dominates the activation of the forget gate (note that we assume b_i to an element of a bounded subset of $\mathbb{R}$) such that the input gate b_ι is close to 1. If

$$-b_\iota^{\boldsymbol{p}} b_c^{\boldsymbol{p}} < \min_d s_c^{\boldsymbol{p}_d^-} b_{\phi,d}^{\boldsymbol{p}} \approx \min_d s_c^{\boldsymbol{p}_d^-},$$

then

$$s_c^{\boldsymbol{p}} > \max_d s_c^{\boldsymbol{p}_d^-}.$$

This means the cell state increases independent of the input. Typically, in such an situation $s_c^{\boldsymbol{p}'}$ increases continuously for $\boldsymbol{p}'$ with $\boldsymbol{p}'_d > \boldsymbol{p}_d$ until all units of the cell are dominated by the cell state. Then the cell output converges to a constant value ($\in \{0, \pm 1\}$). For very small $s_c^{\boldsymbol{p}_d^-}$ and negative $w_{(\phi,d),c}$ the cell state might decrease uncontrolled independent of the input.

This self-reinforcing effect is not just a theoretic one but for $D = 2$ it happens in the practical implementations to about a quarter of all cells according to our experience. A cause for this lies in the following theorem.

Theorem 2. Assume that $\operatorname{sgn}(w_{\phi,d}) = \operatorname{sgn}(w_{\phi,d'}) \neq 0$ for any $d, d' \in \{1, \dots, D\}$ and that there exists a $M > 0$ all $|b_i^{\boldsymbol{p}}| < M$ for any $i \in \{1, \dots, I\}$. Then there exists an $s_c^{\boldsymbol{p}} \in \mathbb{R}$ such that the cell state is increasing.

Proof. W.l.o.g. $w_{(\phi,d)} > 0$. We show that for any $\boldsymbol{p}$ $s_c^{\boldsymbol{p}_d^-}$ is bounded below by some constant C_d. We use this constant to show that $s_c^{\boldsymbol{p}} < s_c^{((\boldsymbol{p})_1^+)_2^+}$.

The function $\varphi(x) := \frac{x}{1+\exp(-ax-b)}$ is continuous in x and negative for $x < 0$. Since $\lim_{x\to-\infty} \varphi(x) = 0 = \varphi(0)$, there exists a constant C such that $\varphi(x) > -C$. Because $b_i^{\boldsymbol{p}}$ and $b_c^{\boldsymbol{p}}$ are bounded, this shows that there are C_d such that $b_{(\phi,d)} s_c^{p_d^-} > -C_d$. The lower bound of $b_\iota^{\boldsymbol{p}} f(a_c^{\boldsymbol{p}})$ is denoted by $-C_i$.

Again because of the boundedness of $b_i^{\boldsymbol{p}}$ and $b_c^{\boldsymbol{p}}$, for any $\epsilon > 0$ there exist a constant $M > 0$ such that if $s_c^{\boldsymbol{p}} > M$, $b_{\phi,1}^{\boldsymbol{p}_1^+} \geq 1 - \epsilon$ and $b_{\phi,1}^{\boldsymbol{p}_2^+} \geq 1 - \epsilon$. Let $M_{\boldsymbol{q}}^\epsilon$ be the lower bound for $s_c^{\boldsymbol{p}}$ such that $b_{\phi,1}^{\boldsymbol{q}} \geq 1 - \epsilon$ for $\boldsymbol{q} > \boldsymbol{p}$.

Now, choose $\epsilon > 0$ such that $2(1-\epsilon)^2 > 1$ and

$$s_c^{\boldsymbol{p}} > \max\left\{ M_{\boldsymbol{p}_1^+}^\epsilon, M_{\boldsymbol{p}_2^+}^\epsilon, M_{(\boldsymbol{p}_1^+)_2^+}^\epsilon, M_{(\boldsymbol{p}_2^+)_1^+}^\epsilon, \frac{(C_i + \sum_{\delta=1}^D C_\delta)(2(1-\epsilon)+1)}{2(1-\epsilon)^2 - \varpi} \right\},$$

for $2(1-\epsilon)^2 > \varpi > 1$. Then for $d \in \{1,2\}$

$$s_c^{\boldsymbol{p}_d^+} > (1-\epsilon)s_c^{\boldsymbol{p}} - C_i - \sum_{\substack{\delta=1 \\ \delta\neq d}}^{D} C_\delta$$

and

$$\begin{aligned}
s^{(\boldsymbol{p}_1^+)_2^+} >& (1-\epsilon)\left[(1-\epsilon)s_c^{\boldsymbol{p}} - C_i - \sum_{\substack{\delta=1 \\ \delta\neq 1}}^{D} C_\delta\right] + (1-\epsilon)\left[(1-\epsilon)s_c^{\boldsymbol{p}} - C_i - \sum_{\substack{\delta=1 \\ \delta\neq 2}}^{D} C_\delta\right] - C_i - \sum_{\delta=3}^{D} C_\delta \\
>& 2(1-\epsilon)^2 s_c^{\boldsymbol{p}} - \left(C_i + \sum_{\delta=1}^{D} C_\delta\right)(2(1-\epsilon)+1) \\
>& \varpi s_c^{\boldsymbol{p}}
\end{aligned}$$

This means $s_c^{\boldsymbol{p}}$ increases by the factor ϖ. □

3 Stabilisierung der Zellen

3.1 stability-discussion

In this section we want to discuss the infuences of gates and of the last internal stat on the internal state. According to (17) we calculate the new state with

$$s_c^{\boldsymbol{p}} = b_\iota^{\boldsymbol{p}} u_c^{\boldsymbol{p}} + \sum_{d=1}^{D} s_c^{\boldsymbol{p}_d^-} b_{\phi,d}^{\boldsymbol{p}}. \tag{31}$$

We discuss two different criteria which we would like to fulfill in regarding the equation above. When there is an error on the internal state, we want to have an update-equation, that this error is reduced in every timestep. More exactly, let $\overline{s}_c^{\boldsymbol{p}} = s_c^{\boldsymbol{p}} + err_c^{\boldsymbol{p}}$ be a noiced state with the error $err_c^{\boldsymbol{p}} := \overline{s}_c^{\boldsymbol{p}} - s_c^{\boldsymbol{p}}$. For any time the error at time $\boldsymbol{p}$ must be smaller than the error at time $\boldsymbol{p}_d^-$ for all $d = 1, \ldots, D$, more exactly we want to find a criterion for the forgetgates such that

$$\forall \boldsymbol{p} \in \mathbb{Z}^D : |err_c^{\boldsymbol{p}}| \leq \max_{d=1,\ldots,D} \left| err_c^{\boldsymbol{p}_d^-} \right| \tag{32}$$

holds. Using equation (31) we can make the following estimate.

$$\begin{aligned}
|err_c^{\boldsymbol{p}}| &= \left| \sum_{d=1}^{D} \overline{s}_c^{\boldsymbol{p}_d^-} b_{\phi,d}^{\boldsymbol{p}} - \sum_{d=1}^{D} s_c^{\boldsymbol{p}_d^-} b_{\phi,d}^{\boldsymbol{p}} \right| && (33)\\
&= \left| \sum_{d=1}^{D} \left(\overline{s}_c^{\boldsymbol{p}_d^-} - s_c^{\boldsymbol{p}_d^-} \right) b_{\phi,d}^{\boldsymbol{p}} \right| && (34)\\
&\leq \sum_{d=1}^{D} \left| \overline{s}_c^{\boldsymbol{p}_d^-} - s_c^{\boldsymbol{p}_d^-} \right| b_{\phi,d}^{\boldsymbol{p}} && (35)\\
&\leq \max_{d=1,\ldots,D} \left| err_c^{\boldsymbol{p}_d^-} \right| \sum_{d=1}^{D} b_{\phi,d}^{\boldsymbol{p}} && (36)
\end{aligned}$$

To hold (32) the activation of the forgetgates have to fulfill

$$\sum_{d=1}^{D} b_{\phi,d}^{\boldsymbol{p}} \leq 1 \tag{37}$$

3.2 LSTM Stable cell

Dividing all activation by the dimension, you fulfill (37). But it is not possible any more to memorize a state in one dimension for a long time. Instead of having a criterion for the forget gates, we can have a criterion for the states.

A better idea is to reduce all states $s_c^{\boldsymbol{p}_i^-}$ to one state $s_c^{\boldsymbol{p}^-}$ and take the 1-dimensional format of the LSTM cell. Then an error from a state cannot grow over time $\boldsymbol{p}$.
So we need a function

$$s_c^{\boldsymbol{p}^-} = f(s_c^{\boldsymbol{p}_1^-}, \ldots, s_c^{\boldsymbol{p}_D^-}, a_{(\lambda,1)}^{\boldsymbol{p}}, \ldots, a_{(\lambda,D)}^{\boldsymbol{p}}) \tag{38}$$

with

$$a_{(\lambda,d)}^{\boldsymbol{p}} = w_{(\lambda,d),c} s_c^{\boldsymbol{p}_d^-} + \sum_{i=1}^{I} w_{(\lambda,d),i} b_i^{\boldsymbol{p}} + \sum_{d'=1}^{D} \left(\sum_{h=1}^{H} w_{(\lambda,d),h}^{d'} b_h^{\boldsymbol{p}_{d'}^-} \right) \tag{39}$$

$$\tag{40}$$

like in (15) with trainable weights w. We want to chose $f(.)$ that the following two benefits of the 1-dimensional LSTM-Cell remain:

1. $\forall a^{\boldsymbol{p}}_{(\lambda,1)}, \ldots, a^{\boldsymbol{p}}_{(\lambda,D)} \in \mathbb{R} : \left| s_c^{\boldsymbol{p}^-} \right| \leq \max_{d=1,\ldots,D} \left| s_c^{\boldsymbol{p}_d^-} \right| \rightarrow$ stability

2. $\forall d = 1, \ldots, D \quad \exists a^{\boldsymbol{p}}_{(\lambda,1)}, \ldots, a^{\boldsymbol{p}}_{(\lambda,D)} \in \mathbb{R} : s_c^{\boldsymbol{p}^-} \approx s_c^{\boldsymbol{p}_d^-} \rightarrow$ memory in dimension d

A convex combination

$$s_c^- = \sum_{d=1}^{D} \alpha_d^{\boldsymbol{p}} s_c^{\boldsymbol{p}_i^-}, \forall d = 1, \ldots, D : \alpha_d^{\boldsymbol{p}} \geq 0, \sum_{d=1}^{D} \alpha_d^{\boldsymbol{p}} = 1 \tag{41}$$

of all states with trainable coefficients $\alpha_d^{\boldsymbol{p}}$ satisfy these both points. To hold (41) we define

$$\alpha_d^{\boldsymbol{p}} := \frac{f_m\left(a^{\boldsymbol{p}}_{(\lambda,d)}\right)}{\sum_{d'=1}^{d} f_m\left(a^{\boldsymbol{p}}_{(\lambda,d')}\right)} \tag{42}$$

with a strictly increasing and differentiable function $f_m : \mathbb{R} \rightarrow (0, \infty)$, $\lim_{t \rightarrow -\infty} f_m(t) = 0$. The value of $f_m\left(a^{\boldsymbol{p}}_{(\lambda,d)}\right)$ shows how important the last state of dimension d is for the new state. Now we can create a multidimensional Cell, which holds (32).

3.2.1 forward pass

cell input:

$$a_c^{\boldsymbol{p}} = \sum_{i=1}^{I} w_{c,i} b_i^{\boldsymbol{p}} + \sum_{d=1}^{D} \sum_{h=1}^{H} w_{c,h}^d b_h^{\boldsymbol{p}_d^-} \tag{43}$$

$$u_c^{\boldsymbol{p}} = f_c(a_c^{\boldsymbol{p}}) \tag{44}$$

input gate:

$$a_\iota^{\boldsymbol{p}} = \sum_{i=1}^{I} w_{\iota,i} b_i^{\boldsymbol{p}} + \sum_{d=1}^{D} \left(w_{\iota,c}^d s_c^{\boldsymbol{p}_d^-} + \sum_{h=1}^{H} w_{\iota,h}^d b_h^{\boldsymbol{p}_d^-} \right) \tag{45}$$

$$b_\iota^{\boldsymbol{p}} = f_\iota(a_\iota^{\boldsymbol{p}}) \tag{46}$$

lambda gates:

$$a_{\lambda,d}^{\boldsymbol{p}} = w_{(\lambda,d),c} s_c^{\boldsymbol{p}_d^-} + \sum_{i=1}^{I} w_{(\lambda,d),i} b_i^{\boldsymbol{p}} + \sum_{d'=1}^{D} \left(\sum_{h=1}^{H} w_{(\lambda,d),h}^{d'} b_h^{\boldsymbol{p}_{d'}^-} \right) \tag{47}$$

$$b_{\lambda,d}^{\boldsymbol{p}} = \frac{f_m(a_{\lambda,d}^{\boldsymbol{p}})}{\sum_{d'=1}^{D} f_m(a_{\lambda,d'}^{\boldsymbol{p}})} \tag{48}$$

last state:

$$s_c^{\boldsymbol{p}^-} = \sum_{d=1}^{D} s_c^{\boldsymbol{p}_d^-} b_{\lambda,d}^{\boldsymbol{p}} \tag{49}$$

forget gate:

$$a_\phi^{\boldsymbol{p}} = w_{\phi,c} s_c^{\boldsymbol{p}^-} + \sum_{i=1}^{I} w_{\phi,i} b_i^{\boldsymbol{p}} + \sum_{d=1}^{D}\sum_{h=1}^{H} w_{\phi,h}^d b_h^{\boldsymbol{p}_d^-} \tag{50}$$

$$b_\phi^{\boldsymbol{p}} = f_\phi(a_\phi^{\boldsymbol{p}}) \tag{51}$$

cell state:

$$s_c^{\boldsymbol{p}} = b_\iota^{\boldsymbol{p}} u_c^{\boldsymbol{p}} + s_c^{\boldsymbol{p}^-} b_\phi^{\boldsymbol{p}} \tag{52}$$

output gate:

$$a_\omega^{\boldsymbol{p}} = \sum_{i=1}^{I} w_{\omega,i} b_i^{\boldsymbol{p}} + w_{\omega,c} s_c^{\boldsymbol{p}} + \sum_{d=1}^{D}\sum_{h=1}^{H} w_{\omega,h}^d b_h^{\boldsymbol{p}_d^-} \tag{53}$$

$$b_\omega^{\boldsymbol{p}} = f_\omega(a_\omega^{\boldsymbol{p}}) \tag{54}$$

cell output:

$$b_c^{\boldsymbol{p}} = b_\omega^{\boldsymbol{p}} g_c(s_c^{\boldsymbol{p}}) \tag{55}$$

3.2.2 backward pass

cell output:

$$\varepsilon_c^{\boldsymbol{p}} = \frac{\partial E}{\partial b_c^{\boldsymbol{p}}} = \sum_{k=1}^{K} w_{k,c}\delta_k^{\boldsymbol{p}} + \sum_{d=1}^{D}\sum_{h=1}^{H}\left(w_{h,c}^d \delta_h^{\boldsymbol{p}_d^+} + w_{\iota,c}^d \delta_\iota^{\boldsymbol{p}_d^+} + w_{\omega,c}^d \delta_\omega^{\boldsymbol{p}_d^+} + w_{\phi,c}^d \delta_\phi^{\boldsymbol{p}_d^+} + \sum_{d'=1}^{D} w_{(\lambda,d'),c}^d \delta_{\lambda,d'}^{\boldsymbol{p}_d^+} \right) \tag{56}$$

output gate:

$$\varepsilon_\omega^{\boldsymbol{p}} = \frac{\partial E}{\partial b_\omega^{\boldsymbol{p}}} = \varepsilon_c^{\boldsymbol{p}} g(s_c^{\boldsymbol{p}}) \tag{57}$$

$$\delta_\omega^{\boldsymbol{p}} = \frac{\partial E}{\partial a_\omega^{\boldsymbol{p}}} = f'_\omega(a_\omega^{\boldsymbol{p}})\varepsilon_\omega^{\boldsymbol{p}} \tag{58}$$

cell state:

$$\varepsilon_s^{\boldsymbol{p}} = \frac{\partial E}{\partial s_c^{\boldsymbol{p}}} = b_\omega^{\boldsymbol{p}} g'_c(s_c^{\boldsymbol{p}})\varepsilon_c^{\boldsymbol{p}} + \delta_\omega^{\boldsymbol{p}} w_{\omega,c} + \sum_{d=1}^{D}\left(\varepsilon_s^{\boldsymbol{p}_d^+} b_\phi^{\boldsymbol{p}_d^+} + w_{\phi,c}\delta_\phi^{\boldsymbol{p}_d^+} \right) b_{\lambda,d}^{\boldsymbol{p}_d^+} + w_{\iota,c}^d \delta_\iota^{\boldsymbol{p}_d^+} + w_{(\lambda,d),c}\delta_{\lambda,d}^{\boldsymbol{p}_d^+} \tag{59}$$

forget gate:

$$\varepsilon_\phi^{\boldsymbol{p}} = s_c^{\boldsymbol{p}_d^-} \varepsilon_s^{\boldsymbol{p}} \tag{60}$$

$$\delta_\phi^{\boldsymbol{p}} = f'_\phi(a_\phi^{\boldsymbol{p}})\varepsilon_\phi^{\boldsymbol{p}} \tag{61}$$

lambda gates:

$$\varepsilon_{\lambda,d}^{\boldsymbol{p}} = \left(\delta_\phi^{\boldsymbol{p}} w_{\phi,c} + \varepsilon_s^{\boldsymbol{p}} b_\phi^{\boldsymbol{p}} \right) s_c^{\boldsymbol{p}_d^-} \tag{62}$$

$$\delta_{\lambda,d}^{\boldsymbol{p}} = \frac{f'_m(a_{\lambda,d}^{\boldsymbol{p}})\varepsilon_{\lambda,d}^{\boldsymbol{p}} \sum\limits_{\substack{d'=1 \\ d'\neq d}}^{D} f_m(a_{\lambda,d'}^{\boldsymbol{p}}) - f_m(a_{\lambda,d}^{\boldsymbol{p}}) \sum\limits_{\substack{d'=1 \\ d'\neq d}}^{D} f'_m(a_{\lambda,d'}^{\boldsymbol{p}})\varepsilon_{\lambda,d'}^{\boldsymbol{p}}}{\left(\sum\limits_{d'=1}^{D} f_m(a_{\lambda,d'}^{\boldsymbol{p}}) \right)^2} \tag{63}$$

input gate:

$$\varepsilon_\iota^{\boldsymbol{p}} = u_c^{\boldsymbol{p}} \varepsilon_s^{\boldsymbol{p}} \tag{64}$$

$$\delta_\iota^{\boldsymbol{p}} = f_\iota'(a_\iota^{\boldsymbol{p}}) \varepsilon_\iota^{\boldsymbol{p}} \tag{65}$$

cell input:

$$\delta_c^{\boldsymbol{p}} = f_c'(a_c^{\boldsymbol{p}}) b_\iota^{\boldsymbol{p}} \varepsilon_c^{\boldsymbol{p}} \tag{66}$$

3.2.3 the 2-dimensional spezial case

In the often used 2-dimensional case it is possible to simplify the lambda gates. Therefor we have to chose the activation function $f_m(x) = e^x$ in equation(48). It follows with the logistic function $f_l(x)$

$$b_{\lambda,1}^{\boldsymbol{p}} = \frac{e^{a_{\lambda,1}^{\boldsymbol{p}}}}{e^{a_{\lambda,1}^{\boldsymbol{p}}} + e^{a_{\lambda,2}^{\boldsymbol{p}}}} = \frac{1}{1 + e^{a_{\lambda,2}^{\boldsymbol{p}} - a_{\lambda,1}^{\boldsymbol{p}}}} = f_l\left(a_{\lambda,1}^{\boldsymbol{p}} - a_{\lambda,2}^{\boldsymbol{p}}\right), \tag{67}$$

$$b_{\lambda,2}^{\boldsymbol{p}} = \frac{e^{a_{\lambda,2}^{\boldsymbol{p}}}}{e^{a_{\lambda,2}^{\boldsymbol{p}}} + e^{a_{\lambda,1}^{\boldsymbol{p}}}} = \frac{1}{1 + e^{a_{\lambda,1}^{\boldsymbol{p}} - a_{\lambda,2}^{\boldsymbol{p}}}} = f_l\left(a_{\lambda,2}^{\boldsymbol{p}} - a_{\lambda,1}^{\boldsymbol{p}}\right). \tag{68}$$

When we define $w_{\lambda,i} := w_{(\lambda,1),i} - w_{(\lambda,2),i}$, $w_{\lambda,h}^d := w_{(\lambda,1),h}^d - w_{(\lambda,2),h}^d$ and change the sign of $w_{(\lambda,2),c}$ in equation (47), we can replace (47) and (48) with

$$a_\lambda^{\boldsymbol{p}} := a_{\lambda,1}^{\boldsymbol{p}} - a_{\lambda,2}^{\boldsymbol{p}} = \sum_{i=1}^{I} w_{\lambda,i} b_i^{\boldsymbol{p}} + \sum_{d=1}^{D} \left(w_{(\lambda,d),c} s_c^{\boldsymbol{p}_{\bar{d}}} + \sum_{h=1}^{H} w_{\lambda,h}^d b_h^{\boldsymbol{p}_{\bar{d}}} \right), \tag{69}$$

$$b_{\lambda,1}^{\boldsymbol{p}} = f_l\left(a_\lambda^{\boldsymbol{p}}\right), \tag{70}$$

$$b_{\lambda,2}^{\boldsymbol{p}} = f_l\left(-a_\lambda^{\boldsymbol{p}}\right) = 1 - f_l\left(a_\lambda^{\boldsymbol{p}}\right). \tag{71}$$

So we have to calculate just one lambda gate in the 2-dimensional case.

4 more stable cells

4.1 stability-discussion

In section 3.1 we discussed the growing of an error $err_c^{\boldsymbol{p}} = \overline{s}_c^{\boldsymbol{p}} - s_c^{\boldsymbol{p}}$. We construct cells which do not increase this error over time. But in experiments the internal state grows (linearly). Another well known stability-criterion is the Bounded-Input-Bounded-Output-Stability (BIBO-stability). In our multidimensional case we can define BIBO-stability for cells:

Definition 3 (BIBO-stability)**.** Let $u_c^{\boldsymbol{p}} = f_t(a_c^{\boldsymbol{p}})$ be the input of a cell and $s_c^{\boldsymbol{p}}$ the internal state. We call the cell BIBO-stable if

$$\forall M \in (0,\infty) : \left\{\forall \boldsymbol{p} \in \mathbb{Z}^D : |u_c^{\boldsymbol{p}}| \leq M \Rightarrow \exists C \in (0,\infty) \,:\, |s_c^{\boldsymbol{p}}| \leq C\, M\right\} \tag{72}$$

holds.

In our cells we use $u_c^{\boldsymbol{p}} = f_c(a_c^{\boldsymbol{p}}) = \tanh(a_c^{\boldsymbol{p}})$. With

$$|u_c| = |\tanh(a_c)| \leq \max_{a_c \in \mathbb{R}} |\tanh(a_c)| = 1 \tag{73}$$

we get the bound $M = 1$. Now we want to find an inequation for the gate activations that fulfill the BIBO-stability. Therefore we assume bounded states in the past

$$\left| s_c^{\boldsymbol{p}_d^-} \right| \leq C \quad \forall d = 1, \ldots, D. \tag{74}$$

With (72), (73) and (74) we can define criteria to the gate activations of the cells.

Lemma 4 (BIBO-stability of LSTM-cells)**.** If

$$\frac{b_\iota^{\boldsymbol{p}}}{C} + \sum_{d=1}^{D} b_{\phi,d}^{\boldsymbol{p}} \leq 1 \tag{75}$$

holds for any activations $b_\iota^{\boldsymbol{p}}, b_{\phi,1}^{\boldsymbol{p}}, \ldots, b_{\phi,D}^{\boldsymbol{p}}$, the LSTM-cell is BIBO-stable.

Proof. The inequation

$$|s_c^{\boldsymbol{p}}| = \left| b_\iota^{\boldsymbol{p}} u_c^{\boldsymbol{p}} + \sum_{d=1}^{D} s_c^{\boldsymbol{p}_d^-} b_{\phi,d}^{\boldsymbol{p}} \right| \leq b_\iota^{\boldsymbol{p}} + \sum_{d=1}^{D} C b_{\phi,d}^{\boldsymbol{p}} = C \left(\frac{b_\iota^{\boldsymbol{p}}}{C} + \sum_{d=1}^{D} b_{\phi,d}^{\boldsymbol{p}} \right) \leq C \tag{76}$$

is fulfilled, if

$$\frac{b_\iota^{\boldsymbol{p}}}{C} + \sum_{d=1}^{D} b_{\phi,d}^{\boldsymbol{p}} \leq 1 \tag{77}$$

holds. □

Lemma 5 (BIBO-stability of LSTM-stable-cells)**.** If

$$b_\iota^{\boldsymbol{p}} \leq C \left(1 - b_\phi^{\boldsymbol{p}}\right) \tag{78}$$

holds for any activations $b_\iota^{\boldsymbol{p}}, b_\phi^{\boldsymbol{p}}$ the LSTM-stable-cell is BIBO-stable. The stability is independent of the activations $b_{\lambda,1}^{\boldsymbol{p}}, \ldots, b_{\lambda,D}^{\boldsymbol{p}}$.

Proof. Using $\sum_{d=1}^{D} b_{\lambda,d}^{\boldsymbol{p}} = 1$, the inequation

$$|s_c^{\boldsymbol{p}}| = \left| b_\iota^{\boldsymbol{p}} u_c^{\boldsymbol{p}} + b_\phi^{\boldsymbol{p}} \sum_{d=1}^{D} s_c^{\bar{\boldsymbol{p}_d}} b_{\lambda,d}^{\boldsymbol{p}} \right| \leq b_\iota^{\boldsymbol{p}} + b_\phi^{\boldsymbol{p}} C \sum_{d=1}^{D} b_{\lambda,d}^{\boldsymbol{p}} \; = C \left(\frac{b_\iota^{\boldsymbol{p}}}{C} + b_\phi^{\boldsymbol{p}} \right) \leq C \tag{79}$$

is fulfilled, if

$$\frac{b_\iota^{\boldsymbol{p}}}{C} + b_\phi^{\boldsymbol{p}} \leq 1 \quad \Leftrightarrow \quad b_\iota^{\boldsymbol{p}} \leq C \left(1 - b_\phi^{\boldsymbol{p}} \right) \tag{80}$$

holds. □

Now we construct a cell which fulfill the BIBO-stability for an arbitrary C and a fixed $M = 1$. The key idea is to fulfill (78) and expand the forward pass with a scalar $G \in (0, \infty)$ such that it substitute C.

4.2 forward pass

cell input:

$$a_c^{\boldsymbol{p}} = \sum_{i=1}^{I} w_{c,i} b_i + \sum_{d=1}^{D} \sum_{h=1}^{H} w_{c,h}^{d} b_h^{\bar{\boldsymbol{p}_d}} \tag{81}$$

$$u_c^{\boldsymbol{p}} = f_c \left(a_c^{\boldsymbol{p}} \right) \tag{82}$$

lambda gates:

$$a_{\lambda,d}^{\boldsymbol{p}} = w_{(\lambda,d),c} s_c^{\bar{\boldsymbol{p}_d}} + \sum_{i=1}^{I} w_{(\lambda,d),i} b_i + \sum_{d'=1}^{D} \left(\sum_{h=1}^{H} w_{(\lambda,d),h}^{d'} b_h^{\bar{\boldsymbol{p}_{d'}}} \right) \tag{83}$$

$$b_{\lambda,d}^{\boldsymbol{p}} = \frac{f_m(a_{\lambda,d})}{\sum_{d'=1}^{D} f_m(a_{\lambda,d'})} \tag{84}$$

last state:

$$s_c^{\bar{\boldsymbol{p}}} = \sum_{d=1}^{D} s_c^{\bar{\boldsymbol{p}_d}} b_{\lambda,d}^{\boldsymbol{p}} \tag{85}$$

forget gate and input gate: The accumulations (45) and (50) were combined to one accumulation. We set $f_\phi(x) := f_l(x)$ and get:

$$a_\phi^{\boldsymbol{p}} = \sum_{i=1}^{I} w_{\phi,i} b_i + \sum_{d=1}^{D} \left(w_{\phi,c}^{d} s_c^{\bar{\boldsymbol{p}_d}} + \sum_{h=1}^{H} w_{\phi,h}^{d} b_h^{\bar{\boldsymbol{p}_d}} \right) \tag{86}$$

$$b_\phi^{\boldsymbol{p}} = f_\phi(a_\phi^{\boldsymbol{p}}) \tag{87}$$

$$b_\iota^{\boldsymbol{p}} = f_\phi(-a_\phi^{\boldsymbol{p}}) = 1 - f_\phi(a_\phi^{\boldsymbol{p}}) \tag{88}$$

cell state: The Input is multiplied by G:

$$s_c^{\boldsymbol{p}} = G \, b_\iota^{\boldsymbol{p}} u_c^{\boldsymbol{p}} + s_c^{\bar{\boldsymbol{p}}} b_\phi^{\boldsymbol{p}} \tag{89}$$

output gate:

$$a_\omega^{\boldsymbol{p}} = \sum_{i=1}^{I} w_{\omega,i} b_i + w_{\omega,c} s_c^{\boldsymbol{p}} + \sum_{d=1}^{D} \sum_{h=1}^{H} w_{\omega,h}^{d} b_h^{\bar{\boldsymbol{p}_d}} \tag{90}$$

$$b_\omega^{\boldsymbol{p}} = f_\omega(a_\omega^{\boldsymbol{p}}) \tag{91}$$

cell output:

$$b_c^{\boldsymbol{p}} = b_\omega^{\boldsymbol{p}} g_c(s_c^{\boldsymbol{p}}) \tag{92}$$

4.3 backward pass

cell output:

$$\varepsilon_c^{\boldsymbol{p}} = \frac{\partial E}{\partial b_c^{\boldsymbol{p}}} = \sum_{k=1}^{K} w_{k,c}\delta_k^{\boldsymbol{p}} + \sum_{d=1}^{D}\sum_{h=1}^{H}\left(w_{h,c}^d \delta_h^{\boldsymbol{p}_d^+} + w_{\omega,c}^d \delta_\omega^{\boldsymbol{p}_d^+} + w_{\phi,c}^d \delta_\phi^{\boldsymbol{p}_d^+} + \sum_{d'=1}^{D} w_{(\lambda,d'),c}^d \delta_{\lambda,d'}^{\boldsymbol{p}_d^+}\right) \tag{93}$$

output gate:

$$\varepsilon_\omega^{\boldsymbol{p}} = \frac{\partial E}{\partial b_\omega^{\boldsymbol{p}}} = \varepsilon_c^{\boldsymbol{p}} g(s_c^{\boldsymbol{p}}) \tag{94}$$

$$\delta_\omega^{\boldsymbol{p}} = \frac{\partial E}{\partial a_\omega^{\boldsymbol{p}}} = f_\omega'(a_\omega^{\boldsymbol{p}})\varepsilon_\omega^{\boldsymbol{p}} \tag{95}$$

cell state:

$$\varepsilon_s^{\boldsymbol{p}} = \frac{\partial E}{\partial s_c^{\boldsymbol{p}}} = b_\omega^{\boldsymbol{p}} g_c'(s_c^{\boldsymbol{p}})\varepsilon_c^{\boldsymbol{p}} + \delta_\omega^{\boldsymbol{p}} w_{\omega,c} + \sum_{d=1}^{D}\left(\varepsilon_s^{\boldsymbol{p}_d^+} b_\phi^{\boldsymbol{p}_d^+} + w_{\phi,c}\delta_\phi^{\boldsymbol{p}_d^+}\right) b_{\lambda,d}^{\boldsymbol{p}_d^+} + w_{(\lambda,d),c}\delta_{\lambda,d}^{\boldsymbol{p}_d^+} \tag{96}$$

forget gate and input gate:

$$\varepsilon_\phi^{\boldsymbol{p}} = s_c^{\boldsymbol{p}_d^-}\varepsilon_s^{\boldsymbol{p}} \tag{97}$$

$$\varepsilon_\iota^{\boldsymbol{p}} = G\, u_c^{\boldsymbol{p}}\varepsilon_s^{\boldsymbol{p}} \tag{98}$$

$$\delta_\phi^{\boldsymbol{p}} = f_\phi'(a_\phi^{\boldsymbol{p}})\left(\varepsilon_\phi^{\boldsymbol{p}} - \varepsilon_\iota^{\boldsymbol{p}}\right) \tag{99}$$

lambda gates:

$$\varepsilon_{\lambda,d}^{\boldsymbol{p}} = \left(\delta_\phi^{\boldsymbol{p}} w_{\phi,c} + \varepsilon_s^{\boldsymbol{p}} b_\phi^{\boldsymbol{p}}\right) s_c^{\boldsymbol{p}_d^-} \tag{100}$$

$$\delta_{\lambda,d}^{\boldsymbol{p}} = \frac{f_m'(a_{\lambda,d})\varepsilon_{\lambda,d}\sum\limits_{\substack{d'=1\\d'\neq d}}^{D} f_m(a_{\lambda,d'}) - f_m(a_{\lambda,d})\sum\limits_{\substack{d'=1\\d'\neq d}}^{D} f_m'(a_{\lambda,d'})\varepsilon_{\lambda,d'}^{\boldsymbol{p}}}{\left(\sum\limits_{d'=1}^{D} f_m(a_{\lambda,d'})\right)^2} \tag{101}$$

cell input:

$$\delta_c^{\boldsymbol{p}} = f_c'\left(a_c^{\boldsymbol{p}}\right) G\, b_\iota^{\boldsymbol{p}}\varepsilon_c^{\boldsymbol{p}} \tag{102}$$

4.4 BIBO-stability of the leaky-cell

Lemma 6 (BIBO-stability of LSTM-stable-cells)**.** The LSTM-leaky-cell is BIBO-stable for any activations $b_\phi^d, b_{\lambda,1}^{\boldsymbol{p}}, \ldots, b_{\lambda,D}$ with an arbitrary $C \in (0,\infty)$.

Proof. Using $\sum_{d=1}^{D} b_{\lambda,d}^{\boldsymbol{p}} = 1$, $b_{\iota}^{\boldsymbol{p}} + b_{\phi}^{\boldsymbol{p}} = 1$ and setting $G := C$, the inequation

$$\left|s_c^d\right| = \left| G\, b_{\iota}^{\boldsymbol{p}} u_c^{\boldsymbol{p}} + b_{\phi}^{\boldsymbol{p}} \sum_{d=1}^{D} \bar{s_c^{\boldsymbol{p}_d}}\, b_{\lambda,d}^{\boldsymbol{p}} \right| \leq G\, b_{\iota}^{\boldsymbol{p}} + b_{\phi}^{\boldsymbol{p}} C \sum_{d=1}^{D} b_{\lambda,d}^{\boldsymbol{p}} = C \left(1 - b_{\phi}^{\boldsymbol{p}} + b_{\phi}^{\boldsymbol{p}}\right) = C \tag{103}$$

$$\tag{104}$$

holds. □

5 general derivation of Leaky-Cells

in this chapter we introduce a more general way to create cells. Therefor we combine the results of the previous sections with the theory of time-discrete linear shift invariant (LSI)-systems. In many signal processing tasks there exists an information signal added with white noise. The aim is to filter the information out of the noisy signal. Often the signals have a specific frequency. When this (often low) frequency is known, an LSI-system can be used to get the information by suppressing the noise. For the theory we orientate towards Poularikas (2000) and Schlichthärle (2000).

5.1 LSI-systems

Let $u : \mathbb{R}^+ \to \mathbb{R}$ be a causal signal. We denote $U(s) = \mathcal{L}\{u(t)\}$ the Laplace-transform of u (Poularikas, 2000, 5.1),(Schlichthärle, 2000, 1.2.1). Sampling the signal by the sample time $T_s > 0$, we get a sequence $u[n] = \{u(nT_s)\}$. We denote $U(z) = \mathcal{Z}\{u[n]\}$ the (one-sided) $\mathcal{Z}$-transform of u (Poularikas, 2000, 6.2),(Schlichthärle, 2000, 3.3). Let $y[n] = f(u[n])$ be the an LSI-system with an input sequence $u[n]$ and an output sequence $y[n]$. The ouput sequence can be calculated by convolving the input sequence with the impulse response $h[n]$ of the LSI-system:

$$y[n] = u[n] * h[n] = \sum_{i=-\infty}^{\infty} u[i]h[n-i] = \sum_{i=-\infty}^{\infty} u[n-i]h[i] = h[n] * u[n] \tag{105}$$

For a causal LSI-system we get $h[n] = 0\,\forall n < 0$. So we can change the limits

$$y[n] = \sum_{i=0}^{\infty} u[n-i]h[i]. \tag{106}$$

Let $u(t) = e^{j\omega t}, j^2 = -1$ be a harmonic function and $u[n] = \{u(nT_s)\}$ the sampled sequence. The output sequence

$$y[n] = u[n]\ H(z)|_{z=e^{j\omega T_s}} \tag{107}$$

$$\sum_{i=0}^{N} a_i y[n-i] = \sum_{i=0}^{M} b_i u[n-i] \tag{108}$$

with $N, M \in \mathbb{N}\ ,\ a_i, b_i \in \mathbb{R}$ a time-discrete LSI-system of the order $\max\{N, M\}$. In some literature a linear shift invariant (LSI)-system is called time-discrete linear time invariant (LTI)-system.
If we $\mathcal{Z}$-transform equation (108) we get:

$$\mathcal{Z}\left\{\sum_{i=0}^{N} a_i y[n-i]\right\} = \mathcal{Z}\left\{\sum_{i=0}^{M} b_i u[n-i]\right\} \tag{109}$$

$$\Leftrightarrow \sum_{i=0}^{N} a_i \mathcal{Z}\{y[n-i]\} = \sum_{i=0}^{M} b_i \mathcal{Z}\{u[n-i]\} \tag{110}$$

$$\Leftrightarrow \sum_{i=0}^{N} a_i z^{-i} \mathcal{Z}\{y[n]\} = \sum_{i=0}^{M} b_i z^{-i} \mathcal{Z}\{u[n]\} \tag{111}$$

$$\Leftrightarrow \mathcal{Z}\{y[n]\} = \frac{\sum_{i=0}^{M} b_i z^{-i}}{\sum_{i=0}^{N} a_i z^{-i}} \mathcal{Z}\{u[n]\} \tag{112}$$

$$\Leftrightarrow Y(z) = H(z)U(z) \tag{113}$$

The rational function

$$H(z) = \frac{Y(z)}{U(z)} = \frac{\sum_{i=0}^{M} b_i z^{-i}}{\sum_{i=0}^{N} a_i z^{-i}} \tag{114}$$

is know as transfer function of an LSI-system (Poularikas, 2000, 6.9). We can split $H(z)$ into a transfer function $H_1(z)$ with infinite impulse response (IIR) and $H_2(z)$ with finite inpulse response (FIR):

$$H(z) = \frac{\sum_{i=0}^{M} b_i z^{-i}}{\sum_{i=0}^{N} a_i z^{-i}} = \underbrace{\frac{1}{\sum_{i=0}^{N} a_i z^{-i}}}_{H_1(z)} \underbrace{\sum_{i=0}^{M} b_i z^{-i}}_{H_2(z)} = H_1(z)H_2(z) \tag{115}$$

Converting this back to difference equations and introducing an internal state $x[n]$, we get

$$\begin{aligned}
X(z) =& H_1(z)U(z) && (116)\\
\Leftrightarrow X(z) =& \frac{1}{\sum_{i=0}^{N} a_i z^{-i}} U(z) && (117)\\
\Leftrightarrow \sum_{i=0}^{N} a_i z^{-i} X(z) =& U(z) && (118)\\
\Leftrightarrow \sum_{i=0}^{N} a_i x[n-i] =& u[n] && (119)\\
\Leftrightarrow x[n] =& \frac{1}{a_0}\left(u[n] - \sum_{i=1}^{N} a_i x[n-i]\right) && (120)\\
\Leftrightarrow x[n] =& \alpha_0 u[n] + \sum_{i=1}^{N} \alpha_i x[n-i] && (121)
\end{aligned}$$

with $\alpha_0 = a_0^{-1}$, $\alpha_i = -a_0^{-1} a_i \forall i = 1, \ldots, N$ and

$$\begin{aligned}
Y(z) =& H_2(z)X(z) && (122)\\
\Leftrightarrow Y(z) =& \sum_{i=0}^{M} b_i z^{-i} X(z) && (123)\\
\Leftrightarrow y[n] =& \sum_{i=0}^{M} b_i x[n-i]. && (124)
\end{aligned}$$

The advantige of these equations is the dependency just of the previous activations of $x[n]$. We want to design a first order LTI-system with trainable coefficients $\alpha_0, \alpha_1, b_0, b_1 \in \mathbb{R}$. One of the well-known properties of LTI-systems is the following:

Lemma 7. Let $u[n] = \left\{e^{j\omega n T_s}\right\}$ be a harmonic input sequence with the imaginary number $j^2 = -1$. Let $H_1(z), H_2(z)$ be the transfer function as defined before. When the poles of $H_1(z)$ are inside the circle $|z| = 1$, the internal state $x[n]$ and the output $y[n]$ of the LTI-system are also harmonic sequences with the same frequency ω, but with different amplitude and phase and can be calculated as $x[n] = H_1(\omega)u[n]$ and $y[n] = H_2(\omega)x[n] = H_1(\omega)H_2(\omega)u[n] = H(\omega)u[n]$ with $H_{1,2}(\omega) = H_{1,2}(z)|_{z=e^{j\omega T_s}}$.

To analyze $H(z)$ we divide it into the amplitude $|H(z)|$ and phase $\arg(H(z))$ and transform from $\mathcal{Z}$- to

$\mathcal{F}$-transform. The amplitude of $H_1(\omega) = H_1(z)|_{z=e^{j\omega T_s}}$ is calculated by

$$|H_1(\omega)| = \left| \frac{\alpha_0}{1 + \alpha_1 e^{j\omega T_s}} \right| \tag{125}$$

$$= \frac{|\alpha_0|}{|1 + \alpha_1 \cos(\omega T_s) + \alpha_1 j \sin(\omega T_s)|} \tag{126}$$

$$= \frac{|\alpha_0|}{\sqrt{(1 + \alpha_1 \cos(\omega T_s))^2 + \alpha_1^2 \sin^2(\omega T_s)}}. \tag{127}$$

To get the low frequency of $u[n]$ we have to set $\alpha_1 \geq 0$. To have the poles of $H_1(z)$ into the circle $|z| = 1$, it follows $|\alpha_1| < 1$, so $\alpha_1 \in [0, 1)$. To get a maximal gain of $C_1 := \max_\omega |H_1(\omega)|$ we get the constraint $|\alpha_0| \leq C_1 (1 - \alpha_1)$. In the same way in analyze $H_2(z)$:

$$|H_2(\omega)| = \left| b_0 + b_1 e^{j\omega T_s} \right| \tag{128}$$

$$= \sqrt{(b_0 + b_1 \cos(\omega T_s))^2 + b_1^2 \sin^2(\omega T_s)} \tag{129}$$

To get the maximal gain at low frequency the parameters b_0 and b_1 must have the same sign. To fulfill the additional constraint $C_2 := \max_\omega |H_2(\omega)|$, we get $|b_0 + b_1| \leq C_2$.
With the bounds for the parameters we now can define a new cell type. Our parameters should be activations of units like the gates in LSTM-cells. We have to find the right activation functions to fulfill the inequations above. Using the weight-space symmetries in a network with at least one hidden layer ((Bishop, 2006, 5.1.1)), wlog. we set $\alpha_0, \alpha_1, b_0, b_1 \geq 0$. Like in 4 we have a cell input $u_c^{\boldsymbol{p}}$ and a previous internal state $s_c^{\boldsymbol{p}^-}$ as weighted convex combination of all other previous states $s_c^{\boldsymbol{p}_d^-}, d = 1, \ldots, D$. The internal state $s_c^{\boldsymbol{p}}$ is calculated like in 4.2 with $G := C_1$. For the output we set $C_2 := 2$.
output gate 0:

$$a_{\omega_0}^{\boldsymbol{p}} = \sum_{i=1}^{I} w_{\omega_0,i} b_i + w_{\omega_0,c} s_c^{\boldsymbol{p}} + \sum_{d=1}^{D} \sum_{h=1}^{H} w_{\omega_0,h}^d b_h^{\boldsymbol{p}_d^-} \tag{130}$$

$$b_{\omega_0}^{\boldsymbol{p}} = f_{\omega_0}(a_{\omega_0}^{\boldsymbol{p}}) \tag{131}$$

output gate 1:

$$a_{\omega_1}^{\boldsymbol{p}} = \sum_{i=1}^{I} w_{\omega_1,i} b_i + w_{\omega_1,c} s_c^{\boldsymbol{p}} + \sum_{d=1}^{D} \sum_{h=1}^{H} w_{\omega_1,h}^d b_h^{\boldsymbol{p}_d^-} \tag{132}$$

$$b_{\omega_1}^{\boldsymbol{p}} = f_{\omega_1}(a_{\omega_1}^{\boldsymbol{p}}) \tag{133}$$

With the logistic functions $f_{\omega_0}(x) = f_{\omega_1}(x) = f_l(x)$ the inequation $b_0 + b_1 \leq 2 = C_2$ is fulfilled. The output is squashed by $g_c(x) := \tanh(x)$ to get one more non-linearity into the cell.
cell output:

$$b_c^{\boldsymbol{p}} = g_c \left(b_{\omega_0}^{\boldsymbol{p}} s_c^{\boldsymbol{p}} + b_{\omega_1}^{\boldsymbol{p}} s_c^{\boldsymbol{p}^-} \right) \tag{134}$$

How can we interpret different activations of the gates? When we assume small input $|u_c^{\boldsymbol{p}}| \ll 1$, the output can be approximated with $b_c^{\boldsymbol{p}} \approx b_{\omega_0}^{\boldsymbol{p}} s_c^{\boldsymbol{p}} + b_{\omega_1}^{\boldsymbol{p}} s_c^{\boldsymbol{p}^-}$. For constant activations of the gate we get the transfer function

$$H(z) = \frac{Y(z)}{U(z)} = \alpha_0 \frac{b_0 + b_1 z^{-1}}{1 - \alpha_1 z^{-1}} = G(1 - b_\iota^{\boldsymbol{p}}) \frac{b_{\omega_0}^{\boldsymbol{p}} + b_{\omega_1}^{\boldsymbol{p}} z^{-1}}{1 - b_\iota^{\boldsymbol{p}} z^{-1}} \tag{135}$$

and the update equations

$$x[n] = \alpha_0 u[n] + \alpha_1 x[n-1] \qquad \Leftrightarrow s_c^{\boldsymbol{p}} = G(1 - b_\iota^{\boldsymbol{p}}) u_c^{\boldsymbol{p}} + b_\iota^{\boldsymbol{p}} s_c^{\boldsymbol{p}^-}, \tag{136}$$

$$y[n] = b_0 x[n] + b_1 x[n-1] \qquad \Leftrightarrow b_c^{\boldsymbol{p}} \approx b_{\omega_0}^{\boldsymbol{p}} s_c^{\boldsymbol{p}} + b_{\omega_1}^{\boldsymbol{p}} s_c^{\boldsymbol{p}^-} \tag{137}$$

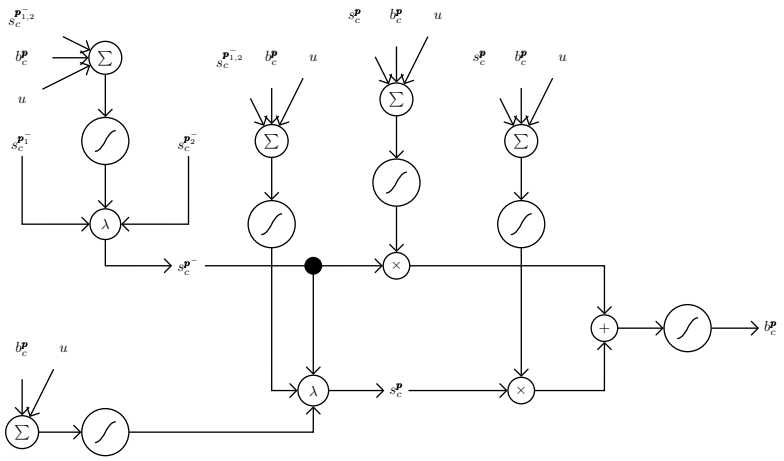


Figure 2: schematic diagram of a LeakyLP cell

$b_\iota^{\boldsymbol{p}}$	$b_{\omega_0}^{\boldsymbol{p}}$	$b_{\omega_1}^{\boldsymbol{p}}$	$\frac{H(z)}{G}$	interpretation
0	1	0	1	normal unit
$\in (0,1)$	$\in (0,1)$	0	$(1-b_\iota^{\boldsymbol{p}})\frac{b_{\omega_0}^{\boldsymbol{p}}}{1-b_\iota^{\boldsymbol{p}} z^{-1}}$	cell of section 4
$\in (0,1)$	0	$\in (0,1)$	$(1-b_\iota^{\boldsymbol{p}})\frac{b_{\omega_1}^{\boldsymbol{p}}}{1-b_\iota^{\boldsymbol{p}} z^{-1}} z^{-1}$	cell of section 4 with delay 1
$\in (-1,1)$	0.5	0.5	$\frac{(1-b_\iota^{\boldsymbol{p}})}{2}\frac{\left(1+z^{-1}\right)}{1-b_\iota^{\boldsymbol{p}} z^{-1}}$	Butterworth lowpass filter (see (138))

Table 1: interpretation of gate activations

For special activations of the gates there exists an interpretation. In table 1 we show some of them. The most interesting row is the last one. There is a direct relation between the cutoff-frequency of a discrete Butterworth lowpass filter and the activation of $b_\iota^{\boldsymbol{p}}$: Let f_{cutoff} be the frequency, where amplitude response is reduced to $\frac{1}{\sqrt{2}}$ of the maximal gain. Let $T_a = 1$ be the sample time. The cutoff frequency f_{cutoff} can be calculated by

$$f_{cutoff} = \frac{1}{\pi} \arctan\left(\frac{1-b_\iota^{\boldsymbol{p}}}{1+b_\iota^{\boldsymbol{p}}}\right), \tag{138}$$

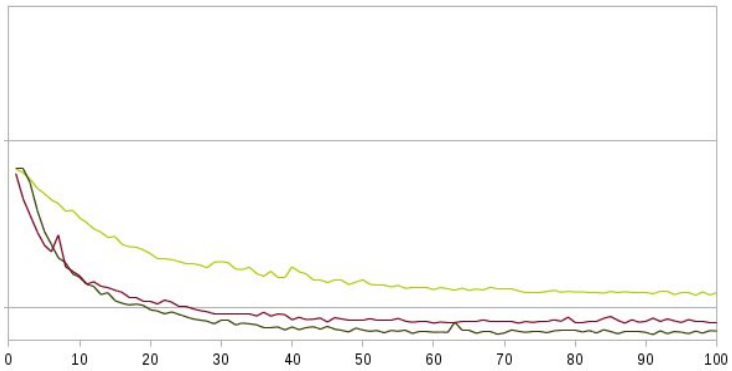
$$b_\iota^{\boldsymbol{p}} = \frac{1+\tan(\pi f_{cutoff})}{1-\tan(\pi f_{cutoff})}. \tag{139}$$

with the bounds $f_{cutoff} \in (0, 0.5)$ and $b_\iota^{\boldsymbol{p}} \in (-1, 1)$. [very more details here....]

6 experiments

6.1 the arabic dataset

To compare the cells with state of the art, we took the 2007 Arabic dataset from ICDAR. This dataset contains of 7 (a-f,s) sets, where 5 (a-e) are available for training and validation. With all information we got from Graves, we were able to reproduce the training-log of him. The networks are trained with gradient decent, using learning rate of 10^-4 and momentum of 0.9, if not otherwise specified. To compare the different networks we take the Label-Error-Rate (LER) how described in (REFERENCE of ALEX or write down by yourself!) at the validation set. First, we want to discuss the typical training of these network, taking a network with LSTMs in all layers. To compare cells we take the neural



network described in Graves (2008). The only thing we do is to substitute the cells. There are three multidimensional (MD) layers. The first MD-layer has just two cells in each direction, but there is a longer MD time series than in the second and third MD-layer. So, for the first layer the stability criterion perhaps is more important than the performance. To compare the different cells with each other, we train 10 networks with different initialization of weights per cell. Of these networks, we take the minimal LER at validation set over all epochs. Afterwords we take the minimum, maximum and median of these 10 values. So we have 3 values to compare different cells.

6.2 different cells in the lowest layer

In this section we chose the lowest layer for comparison. Here, we take all cells described in this paper. All network are trained 100 epochs. The 3 values are shown in the table.

	Label-Error-Rate in Percent		
Celltype	min	max	median
LSTM	8,58%	14,73%	10,58%
Stable	8,78%	11,75%	9,55%
Leaky	8,87%	10,47%	9,10%
LeakyLP	8,24%	9,40%	8,93%

6.3 different cells in other layers

Celltype in layers			Label-Error-Rate in Percent		
1st	2nd	3rd	min	max	median
LSTM	LSTM	LSTM	8,58%	14,73%	10,58%
LeakyLP	LSTM	LSTM	8,24%	9,40%	8,93%
LeakyLP	LeakyLP	LSTM	8,35%	11,27%	8,91%
LeakyLP	LeakyLP	LeakyLP	8,92%	11,69%	9,74%

6.4 stability of cells regarding learning-rate

		Label-Error-Rate in Percent		
Celltype	BP-delta	min	max	median
LSTM	$1 \cdot 10^{-4}$	8,58%	14,73%	10,58%
LSTM	$2 \cdot 10^{-4}$	9,15%	16,86%	10,51%
LSTM	$5 \cdot 10^{-4}$	9,03%	21,77%	11,44%
LSTM	$1 \cdot 10^{-3}$	10,21%	30,20%	11,44%
LeakyLP	$1 \cdot 10^{-4}$	8,92%	11,69%	9,74%
LeakyLP	$2 \cdot 10^{-4}$	8,38%	9,09%	8,81%
LeakyLP	$5 \cdot 10^{-4}$	8,25%	8,95%	8,78%
LeakyLP	$1 \cdot 10^{-3}$	8,29%	9,20%	8,88%
LeakyLP	$2 \cdot 10^{-3}$	8,95%	12,81%	9,55%

6.5 French: different cells in the lowest layer

In this section we chose the lowest layer for comparation. Here, we take all cells decribed in this paper again. All network are trained 111 epochs. The 3 values are shown in the table.

	Label-Error-Rate in Percent		
Celltype	min	max	median
LSTM	14,96%	17,63%	16,50%
Stable	14,45%	16,02%	15,11%
Leaky	14,77%	16,39%	15,85%
LeakyLP	14,63%	15,78%	15,30%

6.6 French: different cells in other layers

Celltype in layers			Label-Error-Rate in Percent		
1st	2nd	3rd	min	max	median
LSTM	LSTM	LSTM	14,96%	17,63%	16,50%
LeakyLP	LSTM	LSTM	14,63%	15,78%	15,30%
LeakyLP	LeakyLP	LSTM	14,21%	15,57%	14,92%
LeakyLP	LeakyLP	LeakyLP	14,94%	16,18%	15,52%

6.7 French: stability of cells regarding learning-rate

Celltype	BP-delta	Label-Error-Rate in Percent min	max	median
LSTM	$1 \cdot 10^{-4}$	14,96%	17,63%	16,50%
LSTM	$2 \cdot 10^{-4}$	14,41%	16,88%	15,61%
LSTM	$5 \cdot 10^{-4}$	15,05%	16,27%	15,47%
LeakyLP	$1 \cdot 10^{-4}$	14,94%	16,18%	15,52%
LeakyLP	$5 \cdot 10^{-4}$	12,68%	13,95%	13,57%
LeakyLP 1st & 2nd	$2 \cdot 10^{-4}$	13,26%	14,04%	13,65%
LeakyLP 1st & 2nd	$5 \cdot 10^{-4}$	12,08%	13,42%	12,87%

References

Bishop, C. M. (2006). *Pattern Recognition and Mashine Learning*. Springer.

Graves, A. (2008). *Supervised Sequence Labelling with Recurrent Neural Networks*. PhD thesis.

Poularikas, A. D. (2000). *The Trainsforms and Applications Handbook*. CRC Press, 2. edition edition.

Schlichthärle, D. (2000). *Digital Filters: basics and design*. Springer.