Cells in Multidimensional Recurrent Neural Networks

Gundram Leifert  
Tobias Strauß  
Tobias Grüning  
Welf Wustlich  
Roger Labahn  
University of Rostock  
Institute of Mathematics  
18051 Rostock, Germany

Keywords: LSTM, MDRNN, CTC, handwriting recognition, neural network

Abstract

The transcription of handwritten text on images is one task in machine learning and one solution to solve it is using multi-dimensional recurrent neural networks (MDRNN) with connectionist temporal classification (CTC). The RNNs can contain special units, the long short-term memory (LSTM) cells. They are able to learn long term dependencies but they get unstable when the dimension is chosen greater than one. We defined some useful and necessary properties for the one-dimensional LSTM cell and extend them in the multi-dimensional case. Thereby we introduce several new cells with better stability. We present a method to design cells using the theory of linear shift invariant systems. The new cells are compared to the LSTM cell on the IFN/ENIT and Rimes database, where we can improve the recognition rate compared to the LSTM cell. So each application where the LSTM cells in MDRNNs are used could be improved by substituting them by the new developed cells.
1. Introduction

Since the last decade, artificial neural networks (NN) became state-of-the-art in many fields of machine learning, for example they can be applied to pattern recognition. Typical NN are feed-forward NN (FFNN) or recurrent NN (RNN), whereas the latter contain recurrent connections. When nearby inputs depend on each other, providing these inputs as additional information to the NN can improve its recognition result. FFNNs obtain these dependencies by making this nearby inputs accessible. If RNNs are used, the recurrent connections can be used to learn if the surrounding input is relevant, but these connections result in a vanishing dependency over time. In S. Hochreiter (1997) the authors develop the long short-term memory (LSTM) which is able to have a long term dependency. This LSTM is extended in A. Graves and Schmidhuber (2007) to the multi-dimensional (MD) case and is used in a hierarchical multi-dimensional RNN (MDRNN) which performed best in three competitions at the International Conference on Document Analysis and Recognition (ICDAR) in 2009 without any feature extraction and knowledge of the recognized language model.

In this paper we analyze these MD LSTM regarding the ability to provide long term dependencies in MDRNNs and show that it can easily have an unwanted growing dependency for higher dimensions. We define a more general description of an LSTM—a cell—and change the LSTM architecture which leads to new MD cell types, which also can provide long term dependencies. In two experiments we show that substituting the LSTM in MDRNNs by these cells works well. Due to this we assume that substituting the LSTM cell by the best performing cell, the LeakyLP cell, will improve the performance of an MDRNN also in other scenarios. Furthermore the new cell types could also be used for the one-dimensional (1D) case, so using them in a bidirectional RNN with LSTMs (BLSTM) could lead to better recognition rates.

In Section 2 we introduce the reader to the development of the LSTM cells (S. Hochreiter, 1997) and its extension (F. A. Gers and Cummins, 1999). Based on that in Section 3 we define two properties that probably lead to the good performance of the 1D LSTM cells. Both together guarantee that the cell can have a long term dependency. A third property ensures that dependencies cannot grow over time. In Section 4 we show that the MD version of the LSTM is still able to provide long term dependency whereas dependencies can grow easily. It is proven that for dimension \( D \geq 2 \) the LSTM cell can have a growing dependency. In Section 5 we change the architecture of the MD LSTM cell and reduce it to the 1D LSTM cell so that the cell fulfills the two properties for any dimension. Nevertheless the cell can become internally unstable over time. This problem is solved in Section 6 using a trainable convex combination of the input and the previous cell states. The new cell type can provide long term dependencies and has bounded dependencies. Motivated by the last sections we introduce a more general way to define MD cells in Section 7. Using the theory of linear shift-invariant systems and their frequency analysis we are able to get a new interpretation of the cells and we create 5 new cell types. To test the performance of the cells in Section 8 we take two datasets from the ICDAR 2009 competitions, where the MDRNNs with LSTM cell won. On these datasets we compare the recognition results of the MDRNNs when we substitute the LSTM
cells by the new developed cells. On both datasets, the IFN/ENIT dataset and the RIMES dataset we can improve the recognition rate using the new developed cells.
In this section we briefly want to introduce a recurrent neural network (RNN) and the development of the LSTM cell. In previous literature there are various notation to describe the update equations of RNNs an LSTMs. To unify the notations we will refer to their notation using “≜” (F. A. Gers and Cummins, 1999; S. Hochreiter, 1997; Graves and Schmidhuber, 2008). Therefore we concentrate on a simple hierarchical RNN with one input layer with the set of neurons $I$, one recurrent hidden layer with the set of neurons $H$ and one output layer with the set of neurons $K$. For each time step $t \in \mathbb{N}$ the layers are updated asynchronously in the order $I, H, K$. In one specific layer all neurons can be updated synchronously. In the hidden layer for one neuron $c \in H$ at time $t \in \mathbb{N}$ we calculate the neuron’s input activation $net_c(t)$ by

$$ net_c(t) = \sum_{i \in I} w_{c,i} y_i(t) + \sum_{h \in H} w_{c,h} y_h(t-1). $$

with weights $w_{\text{[target neuron],[source neuron]}}$. A bias in (1) can be added by extending the set $I := I \cup \{\text{bias}\}$ with $y_{\text{bias}}(t) = 1 \forall t \in \mathbb{N}$ and hence we will not write the bias in the equations, but we use them in our RNNs in section ??.
2.1 The long short-term memory

A standard LSTM cell $c$ has one input with an input activation $y_{cin}(t)$ a set of gates, one internal state $s_c$ and one output(-activation) $y_c (\triangleq y^c)$. The gates are also units and their task is to learn whether a signal should pass the gate or not. They almost always have the logistic activation function $f_{log}(x) := \frac{1}{1+\exp(-x)} (\triangleq f_1(x))$. The input of the standard LSTM cell is calculated from a unit with an odd activation function with a slope of 1 at $x = 0$. We use $f_c(x) = \tanh(x)$ in this paper, another solution could be $f_c(x) = 2 \tanh(x/2)$ (see S. Hochreiter 1997). The standard LSTM has two gates: The input gate (IG or $\iota$) and the output gate (OG or $\omega$). These both gates are calculated like a unit, so that

$$
(y^{inc}(t), b_t(t) \triangleq y_t(t) = f_{log}(\sum_{i \in I} w_{c,i}y_i(t) + \sum_{h \in H} w_{c,h}y_h(t-1))
$$

and

$$
(y^{out}(t), b\omega(t) \triangleq y\omega(t) = f_{log}(\sum_{i \in I} w_{\omega,i}y_i(t) + \sum_{h \in H} w_{\omega,h}y_h(t-1)).
$$

The input of an LSTM is defined like in (1) by

$$
\begin{align*}
(\text{net}_c(t) \triangleq net_c(t) = \sum_{i \in I} w_{c,i}y_i(t) + \sum_{h \in H} w_{c,h}y_h(t-1),
\end{align*}
$$

and

$$
\begin{align*}
(\text{g} (\text{net}_c(t)) , f_2 (\text{net}_c(t)) \triangleq y_{cin}(t) = f_c (\text{net}_c(t)).
\end{align*}
$$

The internal state $s_c(t)$ is calculated by

$$
\begin{align*}
s_c(t) = y_{cin}(t) \cdot y_t(t) + s_c(t-1),
\end{align*}
$$

the output activation $y_c(t)$ of the LSTM is calculated from

$$
\begin{align*}
(y^c(t), b_c(t) \triangleq y_c(t) = h_c(s_c(t)) \cdot y_\omega(t)
\end{align*}
$$

with $h_c(x) := \tanh(x) (\triangleq f_3(x))$. The LSTM can be interpreted as a kind of memory module where the internal state stores the information. For a given input $y_{cin}(t) \in (-1, 1)$ the IG “decides” if the new input is relevant for the internal state. If so, the input is added to the internal state. The information of the input is now saved in the activation of the internal state. The OG determines whether or not the internal activation should be displayed to the rest of the network. So the information, stored in the LSTM is just “readable” when the OG is active. To sum up, an open IG can be seen as a “write”-operation into the memory and an open OG as a “read”-operation of the memory.
Another way to understand the LSTM is to take a look at the gradient propagated through it. Similar to S. Hochreiter (1997) and F. A. Gers and Cummins (1999) we truncate the gradient at all weighted recurrent connections. More exactly \( \forall t \in \mathbb{N} \) we estimate

\[
\forall \gamma \in \{ c_{in}, t, \omega \} : \frac{\partial y_{\gamma}(t)}{\partial y_{c}(t-1)} = 0.
\]

Now, let \( E \) be an arbitrary error which is used to train the RNN and \( \frac{\partial E(t)}{\partial y_{c}(t)} \) the resulting derivative at the output of the LSTM. The OG can eliminate the gradient coming from the output, because

\[
\frac{\partial y_{c}(t)}{\partial s_{c}(t)} = h'_{c}(s_{c}(t)) \cdot y_{\omega}(t),
\]

so the OG decides when the gradient should go into the internal state. Especially for \(|s_{c}(t)| \ll 1\) we get

\[
\frac{\partial y_{c}(t)}{\partial s_{c}(t)} \approx y_{\omega}(t).
\]

The key idea of the LSTMs is that an error that occurs at the internal state neither grows nor decreases over time, more exactly:

\[
\text{Now we can calculate the partial derivative } \frac{\partial s_{c}(t)}{\partial s_{c}(t-1)}, \text{ which we call error carousel (EC) (for more details see S. Hochreiter, 1997). Using the truncated gradient for this derivative, we get}
\]

\[
\frac{\partial s_{c}(t)}{\partial s_{c}(t-1)} = y_{c_{in}}(t) \cdot \frac{\partial y_{c}(t)}{\partial s_{c}(t-1)} + y_{t}(t) \cdot \frac{\partial y_{c_{in}}(t)}{\partial s_{c}(t-1)} + 1
\]

\[
= y_{c_{in}}(t) \cdot \frac{\partial y_{c}(t)}{\partial y_{c}(t-1)} \cdot \frac{\partial y_{c}(t-1)}{\partial s_{c}(t-1)} + y_{t}(t) \cdot \frac{\partial y_{c_{in}}(t)}{\partial y_{c}(t-1)} \cdot \frac{\partial y_{c}(t-1)}{\partial s_{c}(t-1)} + 1
\]

\[
\Rightarrow \frac{\partial s_{c}(t)}{\partial s_{c}(t-1)} = 1. \tag{4}
\]

So, once having a gradient at the internal state we can use the chainrule and get \( \forall \tau \in \mathbb{N} : \frac{\partial s_{c}(t)}{\partial s_{c}(t-\tau)} = 1. \) This is called Constant EC (CEC).

Like the OG can eliminate the gradient coming from the LSTM output, the IG can do the same with the gradient coming from the internal state, that means it decides when the gradient should be injected to the source activations. This can be seen by taking a look at the partial derivative

\[
\frac{\partial s_{c}(t)}{\partial net_{c}(t)} = \frac{\partial s_{c}(t)}{\partial y_{c_{in}}(t)} \cdot \frac{\partial y_{c_{in}}(t)}{\partial net_{c}(t)} = y_{t}(t) f'_{c}(net_{c}(t)).
\]
If there is a small input $|net_c(t)| \ll 1$, we get $f'_c(net_c(t)) \approx 1$ and can estimate
\[
\frac{\partial s_c(t)}{\partial net_c(t)} \approx y_c(t).
\]

All in all, this LSTM is able to store information and learn long-term dependencies, but it has one drawback which will be discussed in 2.2.

### 2.2 Learning to forget

For long time series the internal state is unbounded (compare with F. A. Gers and Cummins, 1999, 2.1). Assuming a positive or negative input and a non zero activation of the IG, the absolute activation of the internal state grows over time. Using the weight-space symmetries in a network with at least one hidden layer (Bishop, 2006, 5.1.1) we assume without loss of generality $y_{cin}(t) \geq 0$, so $s_c(t) \to \infty$. Hence, the activation function $h_c$ saturates and (3) can be simplified to
\[
y_c(t) = h_c(s_c(t)) y_\omega(t) \approx y_\omega(t).
\]

Thus, for great activations of $s_c(t)$ the whole LSTM works like a unit with a logistic activation function. A similar problem can be observed for the gradient. The gradient coming from the output is multiplied by the activation of the OG and the derivative of $h_c$. For great values of $s_c(t)$ we get $h'_c(s_c(t)) \to 0$ and we can estimate the partial derivative
\[
\frac{\partial y_c(t)}{\partial s_c(t)} = h'_c((s_c(t)) \cdot y_\omega(t) \approx 0,
\]

which can be interpreted that the OG is not able to propagate back the gradient into the LSTM. Some solutions to solve the linear growing state problem are introduced in F. A. Gers and Cummins (1999). They tried to stabilize the LSTM with a “state decay” by multiplying the internal state in each time step with a value $\in (0, 1)$, which did not improve the performance. Another solution was to add an additional gate, the forget gate (FG or $\phi$). The last state $s_c(t-1)$ is multiplied by the activation of the FG before it is added to the current state $s_c(t)$. So we can substitute (2) by
\[
s_c(t) = y_{cin}(t) \cdot y_c(t) + s_c(t-1) \cdot y_\phi(t),
\]

so that the truncated gradient in (4) is changed to
\[
\frac{\partial s_c(t)}{\partial s_c(t-1)} = y_{cin}(t) \cdot \frac{\partial y_c(t)}{\partial s_c(t-1)} + y_c(t) \cdot \frac{\partial y_{cin}(t)}{\partial s_c(t-1)} + y_\phi(t)
\]
\[
= y_\phi(t)
\]

7
and for longer time series we get $\forall \tau \in \mathbb{N}$

$$\frac{\partial s_c(t)}{\partial s_c(t - \tau)} = \prod_{t' = 0}^{\tau - 1} y_\phi(t - t').$$

Now, the Extended LSTM is able to learn to forget its previous state. However, an Extended LSTM is still able to work like an standard LSTM without FG by having an activation $y_\phi(t) \approx 1$. In this paper we denote the Extended LSTM as LSTM.
3. Cells and their properties

In this section we want to introduce a general cell and figure out properties for these cells which probably lead to the good performance observed by LSTM cells.

Definition 1 (Cell, cf. Fig. 2) A cell, $c$, of order $k$ consists of

- one designated input unit, $c_{in}$, with sigmoid activation function $f_c$ (typically $f_c = \tanh$ unless specified otherwise);
- a set $\Gamma$ (not containing $c_{in}$) of units called gates $\gamma_1, \gamma_2, \ldots$ with sigmoid activation functions $f_{\gamma_i}$, $i = 1, \ldots$ (typically logistic $f_{\gamma_i} = f_{\text{log}}$ unless specified otherwise);
- an arbitrary function, $g_{\text{int}}$, and a cell activation function, $g_{\text{out}}$, mapping into $[-1, 1]$.

Each unit of $\Gamma \cup \{c_{in}\}$ receives the same set of input activations. The cell update in time step $t \in \mathbb{N}$ is performed in three subsequent phases:
1. Following the classical update scheme of neurons (see Section 2), all units in $\Gamma \cup \{c_{in}\}$ calculate synchronously their activations, which will be denoted by $y_{\Gamma}(t) := (y_{\gamma}(t))_{\gamma \in \Gamma}$ and $y_{c_{in}}(t)$. Furthermore, we call $y_{c_{in}}(t)$ the input activation of the cell.

2. Then, the cell computes it’s so-called internal state: $s_{c}(t) := g_{int}(y_{\Gamma}(t), y_{c_{in}}(t), s_{c}(t-1), \ldots, s_{c}(t-k))$.

3. Finally, the cell computes it’s so-called output activation: $y_{c}(t) := g_{out}(y_{\Gamma}(t), s_{c}(t), s_{c}(t-1), \ldots, s_{c}(t-k))$.

In this paper we concentrate on first order cells ($k = 1$). Now, we use Definition 1 to re-introduce the (Extended) LSTM cell:

**Remark 2 (LSTM cell)** An LSTM cell is a cell of order 1 where $h_{c} = \tanh$ and

- $\Gamma = \{t, \phi, \omega\}$
- $s_{c}(t) := g_{int}(y_{\Gamma}(t), y_{c_{in}}(t), s_{c}(t-1)) := y_{c_{in}}(t)y_{t}(t) + s_{c}(t-1)y_{\phi}(t)$
- $y_{c}(t) := g_{out}(y_{\Gamma}(t), s_{c}(t)) := h_{c}(s_{c}(t))y_{\omega}(t)$

**Properties of cells.** Developing the 1D LSTM cells, the main idea is to save exactly one piece of information over a long time series and to propagate the gradient back over this long time, so that the system can learn precise storage of this piece of information. In instance a given input $y_{c_{in}}$ (which represent the information) at time $t_{in}$ should be stored into the cell state $s_{c}$ until the information is required at time $t_{out}$. To be able to prove the following properties, we will assume the truncated gradient defined in 2.1. Nevertheless we will use the full gradient in our Experiments, because it turned out that it works much better. The next two properties of a cell ensure the ability to work as such a memory.

The first property should ensure that an input $y_{c_{in}}$ at time $t_{in}$ can be memorized (the cell input is open) in the internal activation $s_{c}$ until $t_{out}$ (the cell memorizes) and has a negligibly influence on the internal activation for $t > t_{out}$ (the cell forgets). In addition, the cell is able to prevent influence of other inputs at time steps $t \neq t_{in}$ (the cell input is closed).

**Definition 3 (Long term dependency (LTD))** A cell $c$ allows LTD $\iff$

For arbitrary $t_{in}, t_{out} \in \mathbb{N}, t_{in} \leq t_{out}, \forall \delta > 0$ there exist $\forall t \in \mathbb{N}$ gate activations $y_{\Gamma}(t)$ such that for any $t_{1}, t_{2} \in \mathbb{N}$

$$\frac{\partial s_{c}(t_{2})}{\partial y_{c_{in}}(t_{1})} \in \begin{cases} [1-\delta, 1] & \text{for } t_{1} = t_{in} \land t_{in} \leq t_{2} \leq t_{out} \\ [0, \delta] & \text{otherwise} \end{cases}$$

holds.
Figure 3: **Schematic diagram of a one-dimensional LSTM cell:** The input \( c_{\text{in}} \) is multiplied by the IG \( \iota \). The previous state \( s_c(t-1) \) is gated by the FG \( \phi \) and added to the activation coming from the IG and input. The output of the cell is the squashed internal state (squashed by \( h_c(x) = \tanh(x) \)) and gated by the OG \( \omega \).
The next definition guarantees that at any time \( t \in \mathbb{N} \) the gate activations can (the cell output is open) or not (the cell output is closed) distribute the piece of information saved in \( s_c \) to the network. This is an important property because the piece of information can be memorized in the cell without presenting it to the network. Note that the decision is just dependent on gate activations at time \( t \) and there are no constraints to previous gate activations. In Definition 1 we require \( y_c(t) \in [-1, 1] \) whereas \( s_c(t) \in \mathbb{R} \). So we cannot have arbitrarily small intervals of the derivative as in (5), but we can ensure two distinct intervals for open and closed cell output.

**Definition 4 (Controllable output dependency (COD))** A cell \( c \) of order \( k \) allows COD :\( \iff \exists \delta_1, \delta_2 \in (0, 1), \delta_2 < \delta_1 : \forall t \in \mathbb{N} \)

\[
\exists y^c(t) : \frac{\partial y_c(t)}{\partial s_c(t)} \in [\delta_1, 1] \tag{6}
\]

\[
\exists y^c(t) : \frac{\partial y_c(t)}{\partial s_c(t)} \in [0, \delta_2] \tag{7}
\]

When we take Definition 3 and 4 together, a cell is able to save an input over long term series, can decide at each time step whether or not it is presented to the network and can forget the saved input. The third property is a kind of stability criterion. An unwanted case is that a small change (caused by any noisy signal) at time step \( t_{in} \) has a growing influence at later time steps. This should be prohibited for any gate activations.

**Definition 5 (Not growing EC (NGEC))** A cell \( c \) has an NGEC :\( \iff \forall t_{in}, t \in \mathbb{N}, t_{in} < t, \forall y^c(t) : \)

\[
\frac{\partial s_c(t)}{\partial s_c(t_{in})} \in [0, 1]
\]

We think that a cell fulfilling these three properties can work as stable memory. To be able to prove these properties for the LSTM cell we have to consider the gate activations. In general, the activation function of the gates does not have to be the logistic activation function \( f_{\text{log}} \), whereas for this paper we set \( \forall \gamma \in \Gamma : f_\gamma := f_{\text{log}} \). So the activation of gates can never be exactly 0 or 1, because of a finite input activation \( \text{net}_\gamma(t) \) to the gate activation function. But a gate can have an activation \( y_\gamma(t) \in [1 - \varepsilon, 1] \) if it is opened or \( y_\gamma(t) \in (0, \varepsilon] \) if it is closed, because for a realistic large input activation \( \text{net}_\gamma(t) \geq 7 \) (low input activation \( \text{net}_\gamma(t) < -7 \)) we get an activation within the interval \( y_\gamma(t) \in [1 - \varepsilon, 1) (y_\gamma(t) \in (0, \varepsilon]) \) with \( \varepsilon < \frac{1}{1000} \). Handling with these activation intervals we can prove the definitions for the LSTM cell.

**Proving the properties for the LSTM cell.** Now we can prove whether or not the LSTM cell has these properties:
Theorem 6 (Properties of the LSTM cell) The 1D LSTM cell allows LTD and has an NGEC, but does not allow COD.

Proof see A.1 in appendix.

For the LSTM cell the internal state has no influence on the output when \(|s_c(t)| \gg 1\). Like shown in 2.2, the cell then can only work like a conventional unit with logistic activation function, so there are no more long term dependencies.
4. Expanding to more dimensions

In A. Graves and Schmidhuber (2007) the 1D LSTM cell is extended to an arbitrary number of dimensions; this is solved by using one FG for each dimension. In many publications using the MD LSTM cell in MDRNNs outperform state-of-the-art recognition systems (for example see Graves and Schmidhuber, 2008).

But by expanding the cell to the MD case, the absolute value of the internal state $|s_c|$ can grow over time more easily. This leads to a constant cell output, like it can be seen in Graves and Schmidhuber (2008, Figure 2) where two cells in the lowest MD LSTM layer are constant in a large part of their outputs. Our goal is to transfer the Definitions 3, 4 and 5 defined in Section 3 into the MD case and we will see that the MD LSTM cell has a growing EC. In the next sections we will provide alternative cell types, that fulfill two or all of these definitions.

Regarding the MD date we orientate our notation towards Graves and Schmidhuber (2008). To come from a conventional 1-D date to an MD date, we substitute the $(1D)$ date $t \in \mathbb{N}$ by the $D$-dimensional date $ppp \in \mathbb{N}^D$ and instead of $s_c(t), y_c(t), \ldots$ we write $s_{ppp}^c, y_{ppp}^c, \ldots$. A date one step back in dimension $d = 1, \ldots, D$, we define $ppp^{-d} = (p_1, \ldots, p_d-1, p_d-1, p_{d+1}, \ldots, p_D)$ and in the same way we define $ppp^{+d}$. For all dates where the date is not in the domain $\mathbb{N}^D$, we set the activations to zero ($\forall ppp \in \mathbb{Z}^D \setminus \mathbb{N}^D: s_{ppp}^c, y_{ppp}^c = 0$).

In the 1D case it is clear, that there is just one way to come from date $t_1$ to date $t_2$, when $t_1 < t_2$, by incrementing $t_1$ as long as $t_2$ is reached. For the MD case the number of paths depends on the number of dimensions and the distance between these two dates. An MD path is defined as follows.

**Definition 7 (MD path)** Let $p, q \in \mathbb{N}^D$ be two dates. A $p$-$q$-path $\pi$ of length $k \geq 0$ is a sequence

$$\pi := \{p = p_0, p_1, \ldots, p_k = q\}$$

with $\forall i \in \{1, \ldots, k\} \exists d \in \{1, \ldots, D\} : (p_i^d)^- = p_{i-1}^d$. Further, let $\pi_i := p_i$.

We can define the distance vector

$$\overrightarrow{pq} := q - p = \begin{pmatrix} q_1 - p_1 \\ \vdots \\ q_D - p_D \end{pmatrix} = \begin{pmatrix} \overrightarrow{pq}_1 \\ \vdots \\ \overrightarrow{pq}_D \end{pmatrix}$$

between the dates $p$ and $q$. When $\overrightarrow{pq}$ has at least one negative component, there exists no $p$-$q$-path. Otherwise there exist exactly

$$\#\{\overrightarrow{pq}\} := \frac{\left(\sum_{i=1}^D \overrightarrow{pq}_i\right)!}{\prod_{i=1}^D \overrightarrow{pq}_i!}$$

14
\( p \cdot q \)-paths (compare with the multinomial coefficient). We write \( p < q \) when \( \#(\{p\}^q) \geq 1 \) and \( p \leq q \) when \( p = q \lor p < q \). Now we can extend the definitions of the 1D case to the MD case, whereas we concentrate on the MD cells of order 1:

**Definition 8 (MD cell)** An MD cell, \( c \), of order 1 and dimension \( D \) consists of the same parts as a 1D cell of order 1. The cell update in date \( p \in \mathbb{N}^D \) is performed in three subsequent phases:

1. Following the classical update scheme of neurons (see Section 2), all units in \( \Gamma \cup \{c_{in}\} \) synchronously calculate their activations, which will be denoted by \( y^{p}_{\Gamma} = (y^p_{\gamma})_{\gamma \in \Gamma} \). Furthermore, we call \( y^{p}_{c_{in}} \) the input activation of the cell.

2. Then, the cell computes its so-called internal state:

\[
S^p_c := \text{gint} \left( y^{p}_{\Gamma}, y^{p}_{c_{in}}, s^{p}_{c}, \ldots, s^{p}_{c_{in}} \right).
\]

3. Finally, the cell computes its so-called output activation:

\[
y^p_c := \text{gout} \left( y^{p}_{\Gamma}, s^p_c, y^{p}_{c_{in}}, s^{p+1}_{c}, \ldots, s^{p}_c \right).
\]

Using this, we can reintroduce the LSTM cell as well as Definitions 3, 4 and 5 for the MD case:

**Definition 9 (MD LSTM cell)** An MD LSTM cell is a cell of dimension \( D \) and order 1 where \( h_c = \tanh \) and

- \( \Gamma = \{t, (\phi, 1), \ldots, (\phi, D), \omega\} \)
- \( s^p_c = \text{gint} \left( y^{p}_{\Gamma}, y^{p}_{c_{in}}, s^{p+1}_{c}, \ldots, s^{p}_c \right) = y^{p}_{c_{in}} + \sum_{d=1}^{D} s^{p+1}_{c_{in}} y^{p}_{\phi,d} \)
- \( y^p_c = \text{gout} \left( y^{p}_{\Gamma}, s^{p}_c \right) = h_c \left( s^{p}_c \right) y^{p}_{\omega} \)

**Definition 10 (MD Long term dependency (LTD))** An MD cell \( c \) allows an LTD if:

For arbitrary \( p_{in}, p_{out} \in \mathbb{N}^D \), \( p_{in} \leq p_{out} \), \( \forall \delta > 0 \) there exist \( \forall p \in \mathbb{N}^D \) gate activations \( y^p_{\Gamma} \) such that for any \( p_1, p_2 \in \mathbb{N}^D \)

\[
\frac{\partial s^p_{c_{in}}}{\partial y^p_{c_{in}}} \in tr \left\{ \begin{array}{l}
[1-\delta, 1] & \text{for} & p_1 = p_{in} \land p_{in} \leq p_2 \leq p_{out} \\
[0, \delta] & \text{otherwise}
\end{array} \right.
\]

holds.
**Definition 11 (MD Controllable output dependency (COD))** An MD cell $c$ allows an COD $\iff$

Let $p \in \mathbb{N}^D$ be an arbitrary time step. Then $\exists \delta_1, \delta_2 \in (0, 1), \delta_2 < \delta_1$:

$$\exists y_{p}^c : \frac{\partial y_{p}^c}{\partial s_{p}} \in [\delta_1, 1]$$

(9)

$$\exists y_{p}^c : \frac{\partial y_{p}^c}{\partial s_{p}} \in (0, \delta_2]$$

(10)

**Definition 12 (MD Not growing EC (NGEC))** An MD cell $c$ has an NGEC $\iff$

$$\forall p_{in}, p \in \mathbb{N}^D, p_{in} < p, \forall y_{p}^c :$$

$$\frac{\partial y_{p}^c}{\partial s_{p}} \in [0, 1]$$

We can now consider these definitions for the MD LSTM cell.

**Theorem 13 (LTD of MD LSTM cells)** An MD LSTM cell allows an LTD.

**Proof** see A.2 in appendix.

For arbitrary activations of FGs the partial derivative $\frac{\partial y_{p}^c}{\partial s_{p}}$ can grow over time:

**Theorem 14 (NGEC of MD LSTM cells)** An MD LSTM cell does not have an NGEC, when $D \geq 2$, so it has a growing EC.

**Proof** see A.3 in appendix.

The MD LSTM cell does not allow the COD, because the 1D case is a special case of the MD case. The instability can be seen in [Graves and Schmidhuber (2008), Figure 2]: Two LSTM cells in the lowest MD layer have a nearly constant activation of $y_{p}^c \in \{-1, 0, 1\}$. This happens when $|s_{p}| \to \infty$ and there are peephole connections (for peephole connection details see [E. A. Gers and Schmidhuber, 2002]). The product of the internal state and the peephole connection overlay the other activation-weight-products and this leads to an activation of the OG $y_{p}^c \in \{0, 1\}$ and a squashed internal state $h_{c} (s_{p}) \in \{-1, 1\}$. So the output of the cell is $y_{p}^c h_{c} (s_{p}) = y_{p}^c \in \{-1, 0, 1\}$. But also without peephole connection the internal state can grow. This leads to $h_{c} (s_{p}) \in \{-1, 1\}$ and the cell works more or less like a conventional unit with a logistic activation function. Our idea for the next sections is to change the MD LSTM layout, so that it has an NGEC.

16
5. Reducing the MD LSTM cell to one dimension

In the last section, we showed that the MD LSTM cell can have a growing EC. We tried different ways to bound the EC. For example we divided the activation of the FG by the number of dimensions. Then the EC is bounded by 1, but the cell cannot have long term dependencies any more. Another approach was to give the cells the opportunity to learn to stabilize itself, when the internal state starts diverging. Therefore we add an additional peephole connection between the square value of the previous internal states \( \left( \frac{p_{ppp}^d}{s_{ppp}^d} \right)^2 \) and the FGs so that the cell is able to learn that it has to close the FG for large internal states. This also does not make a significant difference. Also forcing the cell to learn to stabilize itself by adding an error

\[
Loss_{state} = \varepsilon \| s_{c}^p \|_p
\]

with \( p = \{1, 2, 3, 4\} \) and different learning rates \( \varepsilon \) does not work. So we tried to change the layout of the MD LSTM cell.

5.1 MD LSTM Stable cell

In Section 3 we realized that 1D LSTM cells work good and the EC cannot grow over time, but in the MD case the EC can grow over time. Our idea is to combine the previous states \( s_{ppp}^p \) at date \( p \) to one previous state \( s_{ppp}^c \) and take the 1D form of the LSTM cell. For this reason we call this cell **LSTM Stable cell**. Therefore, a function

\[
s_{c}^p = f \left( s_{c}^1, \ldots, s_{c}^D \right)
\]

is needed, so that the following two benefits of the 1D LSTM cell remain:

1. The MD LSTM Stable cell has an NGEC
2. The MD LSTM Stable cell allows LTD.

The convex combination

\[
s_{c}^p = f \left( s_{c}^1, \ldots, s_{c}^D \right) = \sum_{d=1}^{D} \lambda_{d}^p s_{c}^d \quad \forall p = 1, \ldots, D : \lambda_{d}^p \geq 0, \sum_{d=1}^{D} \lambda_{d}^p = 1
\]

of all states satisfies these both points (see Theorems 16 and 17). To calculate these \( D \) coefficients we want to use the activation of \( D \) gates and we call them lambda gates (LG or \( \lambda \)).
Definition 15 (MD LSTM Stable cell)  An MD LSTM Stable cell is a cell of dimension $D$ and order 1 where $h_c = \tanh$ and

- $\Gamma = \{c, (\lambda, 1), \ldots, (\lambda, D), \phi, \omega\}$
- $s^p_c = g_{conv}\left(y^p_c, s^p_c, \ldots, s^p_D\right) = \sum_{d=1}^{D} s_c^d \frac{y^p_{\lambda,d}}{\sum_{d'=1}^{D} y^p_{\lambda,d'}}$
- $s^p = g_{int}\left(y^p_{\Gamma}, y_{c,in}, s^p_c\right) = y^p_c y_{c,in} + s^p c^p y^p_{\phi}$
- $y_c = g_{out}\left(y^p_{\Gamma}, s^p\right) = y^p_c h_c\left(s^p_c\right)$

Using these equations we can test the cell for its properties. The MD LSTM Stable cell does not have the COD, because the 1D LSTM cell also does not have this property. For the other properties we get:

Theorem 16 (LTD of MD LSTM Stable cells)  An MD LSTM Stable cell allows LTD.

Proof See [A.4] in appendix.

Theorem 17 (NGEC of MD LSTM Stable cells)  An MD LSTM Stable cell has an NGEC.

Proof See [A.5] in appendix.

Reducing the number of gates by one.  When $D \geq 2$ an MD LSTM Stable cell has one more gate than a classical MD LSTM (for $D = 1$ the both cells are equivalent). But it is possible to reduce the number of LGs by one. One solution is to choose one dimension $d' \in \{1, \ldots, D\}$ which does not get an LG. Its activation is calculated by

$$y^p_{\lambda,d'} = \prod_{d \in \{1,\ldots,D\} \setminus \{d'\}} \left(1 - y^p_{\lambda,d}\right).$$

In the special case of $D = 2$ we can choose $d' = 2$ and we get $\sum_{d'=1}^{2} y_{\lambda,d'} = y_{\lambda,1} + (1 - y_{\lambda,1}) = 1$ and the update equation of the internal state can be simplified to

$$s^p_c = g_{int}\left(y^p_c, y^p_{\lambda,1}, y^p_{\phi}, s^p_c, s^p_{\phi}\right) = y^p_c y_{c,in} + y^p_{\lambda,1} s^p_c + \left(1 - y^p_{\lambda,1}\right) s^p_{\phi}.$$
6. Bounding the internal state

In the last sections we discussed the growing of the EC over time and we found a solution to have a NGEC for higher dimensions. Nevertheless it is possible that the internal state grows linearly over time. When we take a look at Definition 9, we see that the partial derivative for \( p_{out} \) depends on \( h'_c (s^p_c) \). So having the inequality

\[
\frac{\partial y^p}{\partial s^p_c} \leq h'_c (s^p_c) \quad \text{with} \quad h'_c (s^p_c) \xrightarrow{|s^p_c| \to \infty} 0
\]

the cell allows LTD defined in Definition 10 but actually we have \( \frac{\partial y^p}{\partial y^p_{in}} \xrightarrow{|s^p_c| \to \infty} 0 \) for arbitrary gate activations. Again, ideas like state decay, additional peephole connections or additional loss functions like mentioned in Section 4 either do not work or destroy the LTD of the LSTM and LSTM Stable cell. So, our solution is to change the architecture of the MD LSTM Stable cell, so that it fulfills an NGEC and allows LTD and COD. The key idea is to bound the internal state, so that for all inputs \(|y^p_{in}| \leq 1, p \in \mathbb{N}^D \) the internal state is bounded by \(|s^p_c| \leq 1\).

Note that this is comparable with the well-known Bounded-Input-Bounded-Output-Stability (BIBO-Stability). To create an MD cell that has an NGEC, allows LTD and has a bounded internal state, we take the MD LSTM Stable cell proposed in the last section and change its layout. Therefore we calculate the activation of the IG as function of the FG, so that we achieve \(|s^p_c| \leq 1\) by choosing \( y^p_c := 1 - y^p_\phi \). So the activation of the FG controls how much leaks from the previous states. The activation of the FG can also be interpreted as switch, if the internal activation, the new activation or a convex combination of these both activations should be stored in the cell. So the \( s_c \) can be seen as time-dependent exponential moving average of \( y_{c, in} \).

**Definition 18 (MD Leaky cell)** An MD Leaky cell is a cell of dimension \( D \) and order 1 where \( h_c = \tanh \) and

- \( \Gamma = \{(\lambda, 1), \ldots, (\lambda, D), \phi, \omega\} \)
- \( s^p_c = g_{conv} (y^p_{\Gamma}, s^p_{\Gamma}, \ldots, s^p_D) = \sum_{d=1}^{D} s^p_c - \frac{p^D_{\lambda,d}}{y^p_{\lambda,d}} \sum_{d'=1}^{D} y^p_{\lambda,d'} \)
- \( s^p_c = g_{int} (y^p_{\Gamma}, y^p_{\lambda, in}, s^p_{\Gamma}) = (1 - y^p_\phi) y^p_{\lambda, in} + s^p_{\Gamma} y^p_\phi \)
- \( y^p_c = g_{out} (y^p_{\Gamma}, s^p_c) = y^p_\omega h_c (s^p_c) \)

Now we can prove that the resulting cell has all benefits.

**Theorem 19** The MD Leaky cell has an NGEC and allows LTD and COD.
**Proof**  See [A.6] in appendix.

The MD Leaky cell can have one gate less than the MD LSTM cell and the MD LSTM Stable cell and because of this, the update path requires less computations.
7. General derivation of Leaky cells

In this section we introduce a more general way to create MD cells. So far we proposed cells for the MD case, which are able to provide long term memory. But especially in MDRNNs with more than one MD layer it is hard to measure if and how much long term dependencies are used and even if it is useful. Another way to interpret the cell is to consider them as kind of MD feature extractor like “feature maps” in Convolutional Neural Networks (Bengio and Lecun, 1995). Then the aim is to construct an MD cell which is able to generate useful features. Having a hierarchical Neural Network like in Bengio and Lecun (1995) and Graves and Schmidhuber (2008) over the hierarchies the number of features increases with a simultaneously decreasing feature resolution. Features in a layer with low resolution can be seen as low frequency features in comparison to features in a layer with high resolution. So it would be useful to construct a cell as feature extractor which produces a low frequency output in comparison to its input. Therefore we take a closer look at the theory of linear shift invariant (LSI)-systems and their frequency analysis and analyze a first order LSI-system regarding its free selectable parameters using the $F$-$Z$-transform (for a good overview and more details see Poularikas, 2000; Schlichthärle, 2000). Adding the knowledge of reducing the MD case to the 1D case (see section 5) we create new cell types for the MD case.

The update equations of an first order LSI-system with one input $u$, one internal state $x$ and one output $y$ can be written as

$$ x[n] = h_1(u[n], x[n-1]) = \alpha_0 u[n] + \alpha_1 x[n-1], \quad (12) $$

$$ y[n] = h_2(x[n], x[n-1]) = b_0 x[n] + b_1 x[n-1] \quad (13) $$

with the free selectable coefficients $\alpha_0, \alpha_1, b_0, b_1 \in \mathbb{R}$. Let $U(z) = \mathcal{Z}\{u[n]\}$ be the $Z$-transformed signal of $u[n]$ and $X(z), Y(z)$ respectively. Then we can write the so called transfer functions

$$ H_1(z) := \frac{X(z)}{U(z)} = \frac{\alpha_0}{1 - \alpha_1 z^{-1}}, $$

$$ H_2(z) := \frac{Y(z)}{X(z)} = b_0 + b_1 z^{-1}, $$

$$ H(z) := \frac{Y(z)}{U(z)} = H_2(z) H_1(z). $$

To analyze (12) and (13) according their frequency response we use the relationship between the $F$-transform and the $Z$-transform:

**Remark 20** Let $u[n] = e^{j\omega n}$ be a harmonic input sequence with the imaginary number $j^2 = -1$ and $H(z) = \frac{Y(z)}{U(z)}$ be a transfer functions of an LSI-system. When the poles of $H(z)$ are inside the circle $|z| = 1$, we can change from $Z$- to $F$-transform using the substitution

$$ H(\omega) = H(z) \bigg|_{z = e^{j\omega}} $$
with the harmonic sequence \( y[n] = H(\omega)u[n] = H(\omega)e^{j\omega n} \) with the same frequency \( \omega \) but with a different amplitude and a different phase dependent on the frequency \( \omega \).

We only want to analyze the amplitude of this harmonic sequence
\[
|y[n]| = |H(z)e^{j\omega n}| = |H(z)| = |H_2(z)||H_1(z)|
\]
and do that by analyzing both transfer functions \( H_1(z) \) and \( H_2(z) \) separately. The amplitude of \( H_1(\omega) = H_1(z)|_{z=e^{j\omega}} \) is calculated by
\[
|H_1(\omega)| = \frac{|\alpha_0|}{\sqrt{(1 - \alpha_1 \cos(\omega))^2 + \alpha_1^2 \sin^2(\omega)}}.
\]

Like mentioned before, in many tasks, the information signal has a low frequency. To have the largest amplitude at \( \omega = 0 \) we have to choose \( \alpha_1 \geq 0 \). As mentioned in Remark 20 the poles of \( H_1(z) = \frac{\alpha_0}{1 - \alpha_1 z^{-1}} = \frac{z\alpha_0}{z - \alpha_1} \) have to be in the circle \( |z| = 1 \), so we have the additional constraint \( |\alpha_1| < 1 \). This leads to the bounds \( \alpha_1 \in [0, 1) \). But for \( \alpha_1 \to 1 \) we have \( H_1(0) \to \infty \), so we have to choose \( \alpha_0 \) dependent on \( \alpha_1 \). We set a maximum gain of \( \max_\omega |H_1(\omega)| = |H_1(0)| = 1 \), so we get the constraint
\[
|\alpha_0| \leq 1 - \alpha_1. \tag{14}
\]

In the same way in analyze \( H_2(z) \):
\[
|H_2(\omega)| = |b_0 + b_1 e^{-j\omega}| = \sqrt{(b_0 + b_1 \cos(\omega))^2 + b_1^2 \sin^2(\omega)}
\]
To get the maximal gain at low frequency the parameters \( b_0 \) and \( b_1 \) must have the same sign.

### 7.1 Creating a first order cell

With these constraints for the parameters we now can define a new cell type. The parameters \( \alpha_0, \alpha_1, b_0, b_1 \) should be activations of gates like in LSTM cells. We have to find the right activation functions to fulfill the inequalities above. Using the weight-space symmetries in a network with at least one hidden layer (Bishop, 2006, 5.1.1), without loss of generality we set \( \alpha_0, \alpha_1, b_0, b_1 \geq 0 \). To fulfill the bounds for \( H_1 \), we set \( \alpha_1 \) as activation of a gate with activation function \( f_{log} \). So we have \( \alpha_1 \in (0, 1) \). This is comparable with the FG in the previous sections. To select the \( \alpha_0 \) we choose \( \alpha_0 := 1 - \alpha_1 \) (see (14)). So the value of \( \alpha_0 \) is comparable with the activation of the IG. For \( H_2 \) we set both values \( b_0, b_1 \) as activations of a gate with activation function \( f_{log} \) which leads to \( \max_\omega |H_2(\omega)| = \max \{b_0 + b_1\} = 2 \), so the amplitude response is bounded by 2.
Figure 4: **Schematic diagram of a one-dimensional LeakyLP cell:** The internal state is a convex combination of the new input $c_{in}$ and the previous state $s_c(t - 1)$. The previous state $s_c(t - 1)$ and the current state $s_c(t)$ are gated ($\omega_0$ and $\omega_1$) and accumulated afterwards. The output is squashed by tanh into the interval $[-1, 1]$. 
With these bounds we can define a cell with a cell input $y_{\text{ppp}}^{\text{c}} = u[n]$, a previous internal state $s_{\text{c}}^{-} = x[n - 1]$, an internal state $s_{\text{c}} = x[n]$ and a cell output $y_{\text{c}} = y[n]$. We substitute the coefficients by time dependent gate activations

\[\alpha_0 := 1 - y_{\phi}^{\text{P}} = y_{\text{IG}}^{\text{P}}\]
\[\alpha_1 := y_{\phi}^{\text{P}} = \text{FG}^{\text{P}}\]
\[b_0 := y_{\omega_0}^{\text{P}} = \text{OG}^{\text{P}}\]
\[b_1 := y_{\omega_1}^{\text{P}} = \text{OG of the previous internal state}\]

which leads to the transfer functions

\[H_{1}^{\phi,\text{P}}(z) = \frac{\alpha_0}{1 - \alpha_1 z^{-1}} = \frac{1 - y_{\phi}^{\text{P}}}{1 - y_{\phi}^{\text{P}} z^{-1}},\]
\[H_{2}^{\omega_0,\omega_1,\phi,\text{P}}(z) = b_0 + b_1 z^{-1} = y_{\omega_0}^{\text{P}} + y_{\omega_1}^{\text{P}} z^{-1},\]
\[H(z) = H_{\phi,\omega_0,\omega_1,\phi,\text{P}}(z) = \frac{Y(z)}{U(z)} = \alpha_0 \frac{b_0 + b_1 z^{-1}}{1 - \alpha_1 z^{-1}} = \frac{1 - y_{\phi}^{\text{P}}}{1 - y_{\phi}^{\text{P}} z^{-1}} \left( y_{\omega_0}^{\text{P}} + y_{\omega_1}^{\text{P}} z^{-1} \right).\] (15)

and the update equations

\[x[n] = \alpha_0 u[n] + \alpha_1 x[n - 1] \quad \iff s_{\text{c}} = (1 - y_{\phi}^{\text{P}}) y_{\text{c}_{\text{in}}}^{\text{P}} + y_{\phi}^{\text{P}} s_{\text{c}}^{-},\]
\[y[n] = b_0 x[n] + b_1 x[n - 1] \quad \iff y_{\text{c}} = y_{\omega_0}^{\text{P}} s_{\text{c}} + y_{\omega_1}^{\text{P}} s_{\text{c}}^{-}.\] (16)

The output of the cell is already bounded in $[-2, 2]$, but to fulfill Definition 8 we change (16) to

\[y_{\text{c}}^{\text{P}} = h_{c} \left( y_{\omega_0}^{\text{P}} s_{\text{c}} + y_{\omega_1}^{\text{P}} s_{\text{c}}^{-} \right)\] (17)

with $h_{c} = \tanh$ to ensure $y_{\text{c}}^{\text{P}} \in [-1, 1]$. This additional non-linearity is not necessary but leads to a better performance. This new cell type we call MD Leaky lowpass (LeakyLP) cell.

**Definition 21 (MD LeakyLP cell)** An MD LeakyLP cell is a cell of dimension $D$ and order 1 where $h_{c} = \tanh$ and

- $\Gamma = \{(\lambda, 1), \ldots, (\lambda, D), \phi, \omega_0, \omega_1\}$
- $s_{\text{c}}^{-} = g_{\text{conv}}(y_{\text{P}}^{\text{P}}, s_{\text{c}}^{-}, \ldots, s_{\text{c}}^{D}) = \sum_{d=1}^{D} s_{\text{c}}^{d} \frac{y_{\phi}^{\text{P}}}{\sum_{d'=1}^{D} y_{\phi}^{\text{P}} s_{\text{c}}^{d'}}$
- $s_{\text{c}}^{P} = g_{\text{int}}(y_{\text{P}}^{\text{P}}, y_{\text{c}_{\text{in}}}^{\text{P}}, s_{\text{c}}^{-}) = (1 - y_{\phi}^{\text{P}}) y_{\text{c}_{\text{in}}}^{\text{P}} + s_{\text{c}}^{-} y_{\phi}^{\text{P}}$
Figure 5: Frequency response of $H_1$ (dashed), $H_2$ (dotted) and $H$ (solid) for special parameters.

Top-left: The frequency response of an IIR filter is able to block even low frequency signals, but it cannot be zero at $f = 0.5$. Top-right: The frequency response of an FIR filter cannot be lower than the lightgray dotted line, but for $f = 0.5$ it can be zero. Bottom: When these both filters are concatenated, the resulting frequency response can combine the benefits of each filter.

$$y_c^p = g_{out} \left( y_{\Gamma}^p, s_c^p, s_c^p \right) = h_c \left( s_c^p y_{\omega_0} + s_c^p y_{\omega_1} \right)$$

A block diagram of a 1D LeakyLP cell is shown in Figure 4 and different frequency responses in Figure 5.
7.2 General first order MD cells

With the theory of this section we can easily create new cell types. In general, a cell has a number of gates $\gamma_1, \gamma_2, \ldots \in \Gamma_c$. For $D = 1$ a previous state $s^P_c$ is given directly. Otherwise the previous state is calculated as trainable convex combination of $D$ previous states, like described in Section 5. In Table 1 cell layouts are depicted whereby type A is the cell developed in Section 7 (compare to (15)). For the other types we briefly want to describe the main ideas.

The MD Butterworth lowpass filter. The cell of type B is a special case of the LeakyLP cell. When we set $y_{\phi}^{P} = 0$ there is a direct relation between the cutoff frequency of a discrete Butterworth lowpass filter and the activation of $y_{\phi}^{P}$. Let $f_{\text{cutoff}}$ be the frequency, where amplitude response is reduced to $\frac{1}{\sqrt{2}}$ of the maximal gain. We can calculate $f_{\text{cutoff}}$ by

$$f_{\text{cutoff}} = \frac{1}{\pi} \arctan \left( \frac{1 - y_{\phi}^{P}}{1 + y_{\phi}^{P}} \right) \quad (18)$$

with the bounds $f_{\text{cutoff}} \in (0, 0.5)$ and $y_{\phi}^{P} \in (-1, 1)$ (for more details see Schlichthärle, 2000, 2.2.6.4.2). For $y_{\phi}^{P} \in (0, 1)$ we get $f_{\text{cutoff}} \in (0, 0.25)$. In Figure 6 (left) we can see, that even for a negative value of $y_{\phi}^{P}$ and a highpass characteristic of $H_1(z)$ the impulse response $H(z)$ has a lowpass characteristic.

Adding an additional state gate. In 7.1 we fulfilled (14) for the MD LeakyLP cell by setting $\alpha_0 := 1 - \alpha_1$, so $\alpha_0$ is directly connected with $\alpha_1$. Another solution would be to add an additional value $\gamma \in (0, 1)$ and choose $\alpha_0 := \gamma (1 - \alpha_1)$. So we can extend the MD LeakyLP cell by adding an additional gate $\gamma_4$ for the previous state (see type C). Unfortunately this does not lead to a better performance and one more gate has to be calculated.

Another solution for the output. The cell of type D is another solution to choose $b_0$ and $b_1$ in Section 7.1. For the LeakyLP cell we calculate the output as described in (17). Now we set $b_0 = \gamma_2 \gamma_3^P$ and $b_1 = (1 - \gamma_2^P) \gamma_3^P$, and get

$$y_c^P = \gamma_3^P \left( \gamma_2^P s_c^P + (1 - \gamma_2^P) s_c^P \right).$$

This cell actually works as well as the MD LeakyLP cell and has the same number of gates. In this case we do not need a squashing function $h_c$, because we already have $y_c^P \in [-1, 1]$.

An MD cell as MD PID-controller. Type E has a completely different interpretation: In controlling engineering a PID-controller gets an error as input. In our case the gate activations have to
**Cells in MDRNNs <9.12.2014>**

| Type | $g_{int}(\cdot)$ | $g_{out}(\cdot)$ | $H(z)$ for $h_c(x) = x$ |
|------|------------------|-----------------|--------------------------|
| A    | $(1 - \gamma_1^p) y_{cin}^p + \gamma_1^p s_{c}^p$ | $h_c \left( \gamma_2^p s_{c}^p + \gamma_3^p s_{c}^p \right)$ | $\frac{(1-\gamma_1^p)(\gamma_2^p + \gamma_3^p z^{-1})}{1-\gamma_1^p}$ |
| B    | $(1 - \gamma_1^p) y_{cin}^p + \gamma_1^p s_{c}^p$ | $\frac{s_{c}^p + s_{c}^-}{2}$ | $\frac{(1-\gamma_1^p)(1+z^{-1})}{1-\gamma_1^p}$ |
| C    | $(1 - \gamma_1^p) y_{cin}^p + \gamma_1^p s_{c}^p$ | $h_c \left( \gamma_2^p s_{c}^p + \gamma_3^p s_{c}^p \right)$ | $\frac{(1-\gamma_1^p)(\gamma_2^p + \gamma_3^p z^{-1})}{1-\gamma_1^p}$ |
| D    | $(1 - \gamma_1^p) y_{cin}^p + \gamma_1^p s_{c}^p$ | $\gamma_3^p \left( \gamma_2^p s_{c}^p + (1 - \gamma_2^p) s_{c}^- \right)$ | $\frac{(1-\gamma_1^p)(\gamma_2^p + (1 - \gamma_2^p) z^{-1})}{1-\gamma_1^p z^{-1}}$ |
| E    | $(1 - \gamma_1^p) y_{cin}^p + \gamma_1^p s_{c}^p$ | $h_c \left( \gamma_2^p s_{c}^p + \gamma_3^p (s_{c}^- - s_{c}^p) \right)$ | $\frac{(1-\gamma_1^p)(\gamma_2^p + \gamma_3^p (1 - z^{-1}))}{1-\gamma_1^p z^{-1}}$ |

Table 1: Update equations and transfer function for different cell layouts. The column $s_{c}^p$ contains the update equations to calculate the internal state and column $y_{cin}^p$ contains the update equation for the output. These equations lead to the transfer function $H(z) = H_{\gamma_1^p; \gamma_2^p; \ldots}(z)$.

decide, if the proportional (P), the integral (I) or the derivative (D) term of the error is important for the output. When $\gamma_1^p \approx 0$ we have $y_{cin}^p \approx s_{c}^p$ so the internal state is proportional to the input. Then $\gamma_2^p$ gates the proportional part (P) of the input. The second gate $\gamma_3^p$ gates the difference between the last and the current input, which can be seen as a discrete derivative (D). If $\gamma_2^p \approx 1$ the internal state is an exponential moving average of $y_{cin}^p$ which is an integral term. So $\gamma_2^p$ gates a mainly integral part of the input (I), whereas $\gamma_3^p$ gates a mainly proportional part of the input (P). Dependent on $\gamma_1^p$ type E can be a PD-controller, a PI-controller or a mix of these both. In Figure 6(right) can be seen the frequency response of this cell for different gate activations.
Figure 6: Frequency response of $H_1$ (dashed), $H_2$ (dotted) and $H$ (solid) for special layouts and parameters of Table 1. Left (type B): A butterworth lowpass filter with a negative gate activation $\gamma_{0} = -0.5$ leads to the cutoff frequency $\frac{f_{\text{cutoff}}}{\pi} = \frac{1}{2} \arctan \left( \frac{1+0.5}{1-0.5} \right) \approx 0.3976$. Right (type E): Different frequency responses of a PID controller. Having a fixed $\gamma_{0} = 0.5$ the frequency response is dependent on the activations of $\gamma_{1}$ and $\gamma_{2}$ and can have lowpass (black), allpass (gray) and highpass (lightgray) characteristic.
8. Conclusion

In this paper we took a look at the one-dimensional LSTM cell and discussed the benefits of this cell. We found two properties, that probably make these cells so powerful in the one dimensional case. Expanding these properties to the multi dimensional case, we saw that the LSTM does not fulfill one of these properties any more. We solved this problem by changing the architecture of the cell. In addition we presented a more general idea how to create one dimensional or multi dimensional cells. We compare some newly developed cells with the LSTM cell on two datasets and we can improve the performance using the new cell types. Due to this we think that substituting the multi-dimensional LSTM cells by the multi-dimensional *LeakyLP cell* could improve the performance of any system working with a multi-dimensional space.
Appendix A. proofs

A.1 Proof of 6

Proof Let \( c \) be a 1D LSTM cell. To get the derivative \( \frac{\partial s_c(t_2)}{\partial s_c(t_1)} \) according the truncated gradient between two time steps \( t_1, t_2 \in \mathbb{N} \) we have to take a look at \( g_{int} \).

\[
\frac{\partial s_c(t_2)}{\partial s_c(t_1)} = \frac{\partial g_{int}(y_{t_2}, y_{c_{in}}(t_2), s_c(t_2 - 1))}{\partial s_c(t_1)}
\]

\[
= \frac{\partial (y_{c_{in}}(t_2)y_{t_2})}{\partial s_c(t_1)} + \frac{\partial s_c(t_2 - 1)}{\partial s_c(t_1)} y_{\phi}(t_2) + s_c(t_2 - 1) \frac{\partial y_{\phi}(t_2)}{\partial s_c(t_1)}
\]

\[
= \frac{\partial s_c(t_2 - 1)}{\partial s_c(t_1)} y_{\phi}(t_2)
\]

\[
= \frac{1}{tr} \prod_{t=t_1+1}^{t_2} y_{\phi}(t)
\]

(19)

In addition, \( \forall t \in \mathbb{N} \) we have

\[
\frac{\partial s_c(t)}{\partial y_{c_{in}}(t)} = y_c(t) \quad \text{and} \quad \frac{\partial y_c(t)}{\partial s_c(t)} = h_c^{'}(s_c(t)) y_{\omega}(t).
\]

(21)

We will prove the properties successively.

NGEC: For the LSTM cell the FG \( f_{\phi} = f_{log} \) ensures \( y_{\phi}(t) \in (0, 1) \), so using these bounds in (20) with

\[
\frac{\partial s_c(t)}{\partial s_c(t_{in})} = \frac{1}{tr} \prod_{t'=t_{in}+1}^{t} y_{\phi}(t') \in (0, 1)
\]

the LSTM cell has an NGEC.

LTD: Therefore, we choose

\[
y_c(t) \in \begin{cases} [1 - \varepsilon, 1] & \text{if } t = t_{in} \\ (0, \varepsilon] & \text{otherwise} \end{cases},
\]

\[
y_{\phi}(t) \in \begin{cases} [1 - \varepsilon, 1] & \text{if } t_{in} < t \leq t_{out} \\ (0, \varepsilon] & \text{otherwise} \end{cases}
\]

with a later chosen \( \varepsilon > 0 \). Let \( t_1, t_2 \in \mathbb{N}, t_1 \leq t_2 \) be two arbitrary dates, where we want to calculate the gradient \( \frac{\partial s_c(t_2)}{\partial y_{c_{in}}(t_1)} \). First, we want to show that the LSTM cell allows LTD for \( t_1 = t_{in} \wedge t_{in} \leq \)
\[ t_2 \leq t_{out} : \]
We have \( y_i(t_1) \in [1 - \varepsilon, 1) \) and \( \forall t = t_{in} + 1, \ldots, t_{out} : y_\phi(t) \in [1 - \varepsilon, 1) \). Then, we can estimate the derivative from (19) and (21) by

\[
\frac{\partial s_c(t_2)}{\partial y_{c_{in}}(t_1)} = \frac{\partial s_c(t_2)}{\partial s_c(t_1)} \frac{\partial s_c(t_1)}{\partial y_{c_{in}}(t_1)} = y_i(t_1) \prod_{t=t_1+1}^{t_2} y_\phi(t) \]

\[
\in \left[ (1 - \varepsilon) \prod_{t=t_1+1}^{t_2} (1 - \varepsilon), 1 \right) \subseteq \left[ (1 - \varepsilon)^{t_{out}-t_{in}+1}, 1 \right). \]

To fulfill the equation for LTD we choose \( \varepsilon \) depending on \( \delta \) such that

\[
1 - \delta \leq (1 - \varepsilon)^{t_{out}-t_{in}+1}
\]

\[
\Leftrightarrow \quad \varepsilon \leq 1 - (1 - \delta)^{t_{out}-t_{in}+1}
\]

holds. Second, we have to show, that the derivative is in \([0, \delta]\), when \( t_1 = t_{in} \land t_{in} \leq t_2 \leq t_{out} \) is not fulfilled.

In the case of \( t_1 \neq t_{in} \) when \( \varepsilon \leq \delta \) we can use the NGEC which leads to

\[
\frac{\partial s_c(t_2)}{\partial y_{c_{in}}(t_1)} = \frac{\partial s_c(t_2)}{\partial s_c(t_1)} \frac{\partial s_c(t_1)}{\partial y_{c_{in}}(t_1)} \in \left[ 0, \varepsilon \right] \subseteq [0, \delta].
\]

When \( t_1 = t_{in} \) we have two cases: \( t_2 < t_{in} \) or \( t_2 > t_{out} \). For the case \( t_2 < t_{in} \) the derivative is zero \((\subset [0, \delta])\), because the cell is causal. For \( t_2 > t_{out} \) we can split the derivative at \( t_{out} \) and get

\[
\frac{\partial s_c(t_2)}{\partial y_{c_{in}}(t_1)} = y_i(t_1) \prod_{t=t_1+1}^{t_{out}} y_\phi(t) \prod_{t=t_{out}+1}^{t_2} y_\phi(t) \in (0, \varepsilon) \subseteq (0, \delta).
\]

For \( \varepsilon \leq \min \left\{ \delta, 1 - (1 - \delta)^{t_{out}-t_{in}+1} \right\} \) the LSTM cell allows LTD.

COD: To prove that the LSTM cell has no COD, we show that there are gate activations such that in Definition 4 we get \( \delta_2 > \delta_1 \). Therefore, we assume that all gate activations are arbitrary \((y_\gamma(t) \in (0, 1))\), closed \((y_\gamma(t) \in (0, \varepsilon))\) or opened \((y_\gamma(t) \in [1 - \varepsilon, 1))\) with a later chosen \( \varepsilon > 0 \). We take a look at the right side of (21). For \( s_c(t) = 0 \) we get \( h'_c(s_c(t)) = 1 \). In Definition 4 we have to satisfy \( \exists y_\gamma(t) : \frac{\partial y_\gamma(t)}{\partial o(t)} \in [0, \delta_2] \) an choose the OG \( y_\omega(t) \in (0, \varepsilon) \) with

\[
\varepsilon \leq \delta_2.
\]

(22)
But then for \( t' = 1, \ldots, t - 1 \) we can choose the IG and FG open with the same \( \varepsilon \) so that
\[
y_{\phi}(t'), y_{i}(t') \in [1 - \varepsilon, 1).
\]
When for all time steps \( t' = 1, \ldots, t \) there is a positive input \( y_{c_{in}}(t') \in [c, 1), c \in (0, 1) \subset \mathbb{R} \) and an internal state \( s_{c}(t' - 1) < c \frac{(1 - \varepsilon)}{\varepsilon} \), the internal state is growing over time, because
\[
s_{c}(t') = y_{c_{in}}(t')y_{i}(t') + s_{c}(t' - 1)y_{\phi}(t') \\
\geq c(1 - \varepsilon) + s_{c}(t' - 1)(1 - \varepsilon) \\
\geq s_{c}(t' - 1) + c(1 - \varepsilon) - s_{c}(t' - 1)\varepsilon \\
> s_{c}(t' - 1) + c(1 - \varepsilon) - c \frac{(1 - \varepsilon)}{\varepsilon} \varepsilon \\
> s_{c}(t' - 1).
\]
For large \( s_{c}(t) \geq c \frac{(1 - \varepsilon)}{\varepsilon} \gg 1 \) we can estimate
\[
\tanh(s_{c}(t)) \leq \exp(-s_{c}(t)) \leq \exp \left(-c \frac{(1 - \varepsilon)}{\varepsilon}\right).
\]
This yields in (21) to the bound
\[
\left| \frac{\partial y_{c}(t)}{\partial s_{c}(t)} \right| = \left| h'_{c}(s_{c}(t)) y_{\omega}(t) \right| \leq \exp \left(-c \frac{(1 - \varepsilon)}{\varepsilon}\right) \tag{23}
\]
so in Definition 4 we get
\[
\delta_{1} \leq \exp \left(-c \frac{(1 - \varepsilon)}{\varepsilon}\right). \tag{24}
\]
But when we combine (22), (25) and the restriction in Definition 4 we have
\[
\varepsilon \leq \delta_{2} < \delta_{1} \leq \exp \left(-c \frac{(1 - \varepsilon)}{\varepsilon}\right),
\]
but there exist \( \varepsilon, c \), such that the inequality is not fulfilled, which is a contradiction.
Summarized, the 1D LSTM cell allows an LTD and has an NGEC, but does not allow COD. ■

32
A.2 Proof of 13

Proof  Let \( c \) be an MD LSTM cell of dimension \( D \), \( p, p_1, p_2, p_{\text{in}}, p_{\text{out}} \in \mathbb{N}^D \), \( p_{\text{in}} \leq p_{\text{out}} \) arbitrary dates and \( h_c = \tanh \) the sigmoid function. Besides \( \varepsilon > 0 \) is a later chosen value. In the first step we want to show that there are activations of the forget gates, so that

\[
\frac{\partial s_c^p}{\partial s_{c_{\text{in}}}^p} \in \left\{ \begin{array}{ll} [(1 - \varepsilon) \| p - p_{\text{in}} \|_1, 1] & \text{for } p_{\text{in}} \leq p \leq p_{\text{out}} \\ [0, D \varepsilon] & \text{otherwise} \end{array} \right. \tag{26}
\]

is fulfilled. The prove is done using induction over \( k = \| p - p_{\text{in}} \|_1 \) with \( p \geq p_{\text{in}} \). The base \( k = 0 \) is clear. Let be \( k \geq 1 \). We define

\[ P_p := \{ d \in \{1, \ldots, D\} \mid p_{\text{d}} \geq p_{\text{in}} \} \]

the set of dimensions \( d \), in which are \( p_{\text{in}}, p_{\text{d}} \)-paths. Note, that this set cannot be empty, because \( p > p_{\text{in}} \) for \( k \geq 1 \). When we have a dimension \( d \in P_p \) then \( \| p_{\text{d}} - p_{\text{in}} \|_1 = k - 1 \) and we assume

\[
\frac{\partial s_c^p}{\partial s_{c_{\text{in}}}^p} \in \left[ (1 - \varepsilon) \| p_{\text{d}} - p_{\text{in}} \|_1, 1 \right] = \left( 1 - \varepsilon \right)^{k-1}, 1 \right]. \tag{27}
\]

Then we choose the activations of the FG to be

\[
y_{p, d} \in \left\{ \begin{array}{ll} \left[ \frac{1 - \varepsilon}{|p_{\text{p}|}}, \frac{1}{|p_{\text{p}|}} \right] & \text{for } d \in P_p \wedge p_{\text{in}} < p \leq p_{\text{out}} \\ \left[ 0, \varepsilon \right] & \text{otherwise} \end{array} \right. \tag{28}
\]

Then we can estimate the derivative for \( p_{\text{in}} \leq p \leq p_{\text{out}} \) using (27) and (28) to

\[
\frac{\partial s_c^p}{\partial s_{c_{\text{in}}}^p} \approx \sum_{d \in P_p} \frac{\partial s_c^p}{\partial s_{c_{\text{in}}}^p} \sum_{d \in P_p} \frac{\partial s_c^p}{\partial s_{c_{\text{in}}}^p} \frac{1}{|p_{\text{p}|}} \sum_{d \in P_p} \frac{\partial s_c^p}{\partial s_{c_{\text{in}}}^p} \frac{1}{|p_{\text{p}|}} \left[ P_p \right] \left( 1 - \varepsilon \right)^{k-1} \frac{1}{|p_{\text{p}|}} \frac{1}{|p_{\text{p}|}} \frac{1}{|p_{\text{p}|}} \left( 1 - \varepsilon \right)^{k-1}, 1 \right], \tag{29}
\]

so (26) is fulfilled for \( p_{\text{in}} \leq p \leq p_{\text{out}} \).

If we have \( p < p_{\text{in}} \) in (26), the derivative is 0, because we have a causal system.

For \( p > p_{\text{out}} \) in (26), we choose \( \varepsilon \leq \frac{1}{D} \leq \frac{1}{|p_{\text{p}|}} \) in (28) to ensure \( \forall p \in \mathbb{N}^D \), \( \left| \frac{\partial s_c^p}{\partial s_{c_{\text{in}}}^p} \right| \leq 1 \) (see 29) and we get

\[
\frac{\partial s_c^p}{\partial s_{c_{\text{in}}}^p} = \sum_{d=1,...,D} \frac{\partial s_c^p}{\partial s_{c_{\text{in}}}^p} \sum_{d=1,...,D} \frac{\partial s_c^p}{\partial s_{c_{\text{in}}}^p} \left[ 0, D \varepsilon \right] \left( \max_{d=1,...,D} \frac{\partial s_c^p}{\partial s_{c_{\text{in}}}^p} \right) \subseteq (0, D \varepsilon], \tag{30}
\]
and (26) is fulfilled.
In the second step let \( p_1 \leq p_2 \) be the date, for which we want to calculate the truncated gradient \( \frac{\partial s_{p_2}}{\partial y_{c_{in}}} \). We choose the IG activation as

\[
y^p_c \in \begin{cases} 
[1 - \varepsilon, 1) & \text{if } p = p_{in} \\
(0, \varepsilon] & \text{otherwise}
\end{cases}
\]

and we get \( \frac{\partial s}{\partial y_{c_{in}}} = y^p_c \). Using (29), (30) and (31), we can estimate the partial derivative by

\[
\frac{\partial s_{p_2}}{\partial y_{c_{in}}} = \frac{\partial s_{p_2}}{\partial s_{c_{in}}} \frac{\partial s_{c_{in}}}{\partial y_{c_{in}}}
\]

\[
\Rightarrow \frac{\partial s_{p_2}}{\partial s_{c_{in}}} \in \begin{cases} 
[(1 - \varepsilon)(1 - \varepsilon)||p_{2} - p_{in}||_1, 1] & \text{for } p_1 = p_{in} \land p_{in} \leq p_2 \leq p_{out} \\
[0, D\varepsilon] & \text{otherwise}
\end{cases}
\]

and setting

\[
\varepsilon := \min \left\{ \frac{\delta}{D}, 1 - (1 - \delta) \frac{1}{||p_{in} - p_{out}||_1 + 1} \right\}
\]

the conditions of Definition 10 are fulfilled.

\[ \blacksquare \]

### A.3 Proof of 14

**Proof** Let \( c \) be an MD cell of dimension \( D \) with the internal state \( s_c \) and \( p_{in}, p_k \in \mathbb{N}^D, p_{in} \leq p_k \) two dates. Let \( p_k \) be a date \( k \) steps further in each dimension than a fixed date \( p_{in} \). So the distance between them is \( ||p_{in} - p_k||_1 = Dk \). Let \( \Pi \) be the set of all \( p_{in}, p_k \) paths, then there exist \( ||\Pi|| = \#\{p_{in}, p_k\} \) paths (see Definition 7). We assume

\[
y^p_{\phi, d} \in [\varepsilon, 1 - \varepsilon]
\]

with \( \varepsilon \in (0, 0.5) \) and we can estimate the partial derivative, using the truncated gradient, with

\[
\frac{\partial s_{p_k}}{\partial s_{c_{in}}} = \sum_{\pi \in \Pi} \frac{1}{\#\{p_{in}, p_k\}} \prod_{i=1}^k y^p_{\pi_i, d}
\]

\[
\in \text{tr} \left[ e^{k \#\{p_{in}, p_k\}} (1 - \varepsilon)^k \#\{p_{in}, p_k\} \right].
\]
For $D = 1$ we get $|\Pi| = 1$ and the cell has a NGEC. When $D \geq 2$ we can count the number of paths using the Stirling’s approximation and we can estimate the number of paths with

$$\#\{\overrightarrow{p_{in}p_k}\} = \left(\sum_{i=1}^{D} (\overrightarrow{p_{in}p_k})_i\right)! = \frac{(Dk)!}{(kl)^D} \approx \frac{\sqrt{2\pi Dk} \left(\frac{Dk}{e}\right)^D}{\left(\frac{2\pi k \left(\frac{k}{e}\right)^k}{D}\right)^D} = \frac{\sqrt{D}D^k}{\sqrt{2\pi k} D^{D-1}}.$$  

When we combine it with the FG activations we can estimate the derivative for great $k$ with the Stirling’s approximation and get

$$\frac{\partial s^p_c}{\partial s^p_{in}} \in tr \left[ \varepsilon^{Dk} \#\{\overrightarrow{p_{in}p_k}\}, (1 - \varepsilon)^{Dk} \#\{\overrightarrow{p_{in}p_k}\} \right]$$

$$k \gg 1 \in tr \left[ \frac{\sqrt{D}}{\sqrt{2\pi k} D^{D-1}} (D\varepsilon)^{Dk}, \frac{\sqrt{D}}{\sqrt{2\pi k} D^{D-1}} (D (1 - \varepsilon))^{Dk} \right].$$

The upper bound of this interval can grow for great $k$, if $[D (1 - \varepsilon)] > 1$ and this is the case for $D \geq 2$. So the MD LSTM cell can have a growing EC for $D \geq 2$. When the weights to the FGs are initialized with small values, we have $y^p_{\phi,d} \approx 0.5$. Then we have a growing EC when $D \geq 3$, when the training is starting. In the worst case we have $y^p_{\phi,d} \approx 1$ and the derivative in (32) goes for great $k$ to

$$\frac{\partial s^p_c}{\partial s^p_{in}} \approx \frac{\sqrt{D}}{\sqrt{2\pi} D^{D-1}} k^{D-\frac{1}{2}} (D)^{Dk}.$$  

\[\blacksquare\]

### A.4 Proof of [16]

**Proof** Let $c$ be an MD LSTM Stable cell of dimension $D \geq 2$ (for $D = 1$ the proof is equivalent to the 1D case of the LSTM cell), $p, p_1, p_2, p_{in}, p_{out} \in \mathbb{N}^D, p_{in} \leq p_{out}$ arbitrary dates and $h_c = \tanh$ the sigmoid function. Besides $\varepsilon > 0$ is a later chosen value.

In the first step we want to show that there are activations of the forget gates, so that

$$\frac{\partial s^p_c}{\partial s^p_{in}} \in tr \left\{ \left[(1 - (D - 1)\varepsilon)^2 \|p - p_{in}\|_1, 1\right] \right\} \text{ for } p_{in} \leq p \leq p_{out}$$

$$\left\{ 0, \varepsilon \right\} \text{ otherwise} \quad (33)$$

is fulfilled. The prove is done using induction over $k = \|p - p_{in}\|_1$. The base $k = 0$ is clear. Let be $k \geq 1$. We define

$$P_p := \{d \in \{1, \ldots, D\} \mid p_d \geq p_{m_d} \}$$
the set of dimensions \( d \), in which are \( p_{in} \rightarrow p_{d} \)-paths. Note, that this set cannot be empty, because \( p > p_{in} \) for \( k \geq 1 \). When we have a dimension \( d \in P \) then \( \| p_{d} \rightarrow p_{in} \|_1 = k - 1 \) and we assume

\[
\frac{\partial s_{c}^{p_{d}}}{\partial s_{c}^{p_{in}}} \in \text{tr} \left[ (1 - (D - 1)\varepsilon)^2 \| p_{d} \rightarrow p_{in} \|_1 , 1 \right] = \left[ (1 - (D - 1)\varepsilon)^2(2k-1) , 1 \right].
\] (34)

When we choose the activations of the LGs to be

\[
y_{\lambda,d}^{p} \in \begin{cases} [1 - \varepsilon, 1) & \text{for} \ d \in P \land P_{in} < p \leq P_{out} \\ (0, \varepsilon] & \text{otherwise} \end{cases}
\]

we can estimate

\[
\frac{\sum_{d \in P \setminus P_{in}} y_{\lambda,d}^{p}}{\sum_{d'=1}^{D} y_{\lambda,d'}^{p}} \in (1 - (D - 1)\varepsilon, 1],
\]

because

\[
1 \geq \frac{\sum_{d \in P} y_{\lambda,d}^{p}}{\sum_{d'=1}^{D} y_{\lambda,d'}^{p}} = \frac{\sum_{d \in P \setminus P_{in}} y_{\lambda,d}^{p}}{\sum_{d'=1}^{D} y_{\lambda,d'}^{p} + \sum_{d \in \{1, \ldots, D\} \setminus P \setminus P_{in}} y_{\lambda,d'}^{p}}
\]

\[
\geq \frac{| P_{p} | (1 - \varepsilon)}{| P_{p} | (1 - \varepsilon) + (D - | P_{p} |) \varepsilon}
\]

\[
\geq \frac{| P_{p} | (1 - (D - 1)\varepsilon)}{| P_{p} | (1 - (D - 1)\varepsilon) + (D - 1) \varepsilon}
\]

\[
\geq \frac{1 - (D - 1)\varepsilon}{| P_{p} | - \varepsilon(D - 1) (| P_{p} | - 1)}
\]

\[
\geq (1 - (D - 1)\varepsilon).
\]

Setting the FG to

\[
y_{\phi}^{p} \in \begin{cases} [1 - \varepsilon, 1) & \text{for} \ p_{in} < p \leq p_{out} \\ (0, \varepsilon] & \text{otherwise} \end{cases}
\]

we can estimate the derivative for \( p_{in} \leq p \leq p_{out} \) using (34), (35) and (36) to

\[
\frac{\partial s_{c}^{p_{in}}}{\partial s_{c}^{p_{in}}} \in \text{tr} \left( (1 - \varepsilon) (1 - (D - 1)\varepsilon)^2(2k-1) (1 - (D - 1)\varepsilon), 1 \right)
\]

\[
\Rightarrow \frac{\partial s_{c}^{p_{in}}}{\partial s_{c}^{p_{in}}} \in \text{tr} \left( (1 - (D - 1)\varepsilon)^{2k}, 1 \right)
\] (37)
so (33) is fulfilled for \( p_{in} \leq p \leq p_{out} \).

If we have \( p < p_{in} \) in (33), the derivative is 0, because we have a causal system.

For \( p > p_{out} \) the FG is closed (see (36)), and using the upper bounds of (34) and (35) we get

\[
\frac{\partial s^p_{p_{in}}}{\partial s^p_{c}} \mid_{tr} = y_{p_{in}}^p \left( \sum_{d=1}^{D} \frac{\partial s^p_{c}}{\partial s^p_{c}} \sum_{d'=1}^{D} \frac{y^p_{\lambda,d'}}{y^p_{\lambda,d'}} \right) 
\]

(38)

\[\in (0, \varepsilon)\]

\[\odot_{s^p_{p_{in}}}{c} \odot_{s^p_{c}} \in tr (0, \varepsilon)[\] and (33) is fulfilled.

In the second step let \( p_{1} \leq p_{2} \) be the date, for which we want to calculate the truncated gradient \( \frac{\partial s^p_{2}}{\partial s^p_{1}} \). We choose the IG activation as

\[
y_{p_{in}}^p \in \begin{cases} [1 - \varepsilon, 1) & \text{if } p = p_{in} \\ (0, \varepsilon) & \text{otherwise} \end{cases}
\]

(39)

\[\text{and we get } \frac{\partial s^p_{2}}{\partial y_{p_{in}}^p} = y_{p_{in}}^p. \]

Using (37), (38) and (39), we can estimate the partial derivative by

\[\Rightarrow \frac{\partial s^p_{2}}{\partial y_{p_{in}}^p} \in \begin{cases} [(1 - \varepsilon)(1 - (D - 1)\varepsilon)^2\|p_{2} - p_{in}\|_1, 1] & \text{for } p_{1} = p_{in} \land p_{in} \leq p_{2} \leq p_{out} \\ [0, \varepsilon] & \text{otherwise} \end{cases}
\]

and setting

\[\varepsilon := \min \left\{ \delta, \left( 1 - (1 - \delta)^2\|p_{in} - p_{out}\|_1 \right)^2 \left( \frac{1}{D - 1} \right) \right\}
\]

the conditions of Definition 10 are fulfilled.

A.5 Proof of 17

**Proof** Let \( c \) be a MD LSTM Stable cell of dimension \( D \) with the internal state \( s_c \) and \( p_{in}, p \in \mathbb{N}^D, p_{in} \leq p \) two arbitrary dates and \( \|p_{in} - p\|_1 = k \). Let all gate activations be arbitrary in \([0, 1] \). We show that

\[\frac{\partial s^p_{p_{in}}}{\partial s^p_{c}} \in tr [0, 1] \]

(40)

is fulfilled \( \forall k \in \mathbb{N} \) using induction over \( k \). For the base case \( k = 0 \) we get \( \frac{\partial s^p_{p_{in}}}{\partial s^p_{c}} = \frac{\partial s^p_{\infty}}{\partial s^p_{c}} = 1 \).

Let (40) be fulfilled for \( k - 1 \). That means if \( p_{in} \geq p_{in} \) we have \( \|p_{d} - p_{in}\|_1 = k - 1 \) and this
leads to \( \frac{\partial p_d^-}{\partial s_{c^{in}}} \in [0, 1] \). If \( p_d^- \neq p_{c^{in}} \) then there is no \( p_{c^{in}}-p_d^- \)-path and we have \( \frac{\partial p_d^-}{\partial s_{c^{in}}} = 0 \) for this dimension. Then we can calculate the derivative

\[
0 \frac{\partial s_c^{p}}{\partial s_{c^{in}}} = y_p^\phi \sum_{d=1}^D \frac{\partial s_c^{p_d}}{\partial s_{c^{in}}} \cdot \frac{y_{\lambda,d}^p}{\sum_{d'=1}^D y_{\lambda,d'}^p} \in \left[ 0, \max_{d \in P} \left\{ \frac{\partial s_c^{p_d}}{\partial s_{c^{in}}} \cdot \frac{y_{\lambda,d}^p}{\sum_{d'=1}^D y_{\lambda,d'}^p} \right\} \right]
\]

\[ \in [0, 1], \]

which gives us the desired interval.

A.6 Proof of 19

Proof

NGEC: The cell has an NGEC, because all gates have the same bounds as the MD Stable cell.

LTD: To prove the LTD, we use the proof of Theorem 16. The difference between the MD Stable cell and the MD Leaky cell is that the activations of the FG and IG are dependent on each other for the Leaky cell. Let \( p_{c^{in}}, p \in \mathbb{N}^D, p_{c^{in}} \leq p \) be two arbitrary dates like in Theorem 16. The IG has just the a restriction that for \( p = p_{c^{in}} \) it has to hold \( y_{p}^{c_{in}} \in [1 - \varepsilon, 1) \). Here, the FG can have an arbitrary activation, so we chose \( y_{p}^{c} = 1 - y_{p}^{c_{in}} \). For all \( p > p_{c^{in}} \) the FG have to be in the ranges, shown in (36), while the IG has no restriction and we choose \( y_{p}^{c_{in}} = 1 - y_{p}^{c} \), so the MD Leaky cell has the LTD.

COD: The proof that the MD Leaky cell allows COD can be done by estimating the bounds of \( s_c^{p} \). From the update equations of the cell we get

\[
|s_c^{p}| \leq \max_{i=1,...,D} |s_c^{p_d^-}|.
\]

Now we can estimate the internal state using the ranges \( y_{c^{in}}^{p} \in [-1, 1] \), recursion over \( p \)

\[
|s_c^{p}| = \left| \left(1 - y_{c_{in}}^{p} \right) y_{c_{in}}^{p} + y_{c_{in}}^{p_d^-} \right| \leq \max \left\{ |y_{c_{in}}^{p_d^-}|, |s_c^{p_d^-}|, \ldots, |s_c^{p_{c_{in}}}| \right\} \leq \max_{q \leq p} \left\{ |y_{c_{in}}^{q}| \right\} \leq 1
\]

and get \( s_c^{p} \in [-1, 1] \). To fulfill the derivatives in Definition 11 for \( \delta_1 \) we choose \( y_{c_{in}}^{p} \in [1 - \varepsilon, 1) \) and get

\[
\delta_1 \leq \min_{s_c^{p}} \left\{ h^{c}' \left(s_c^{p} \right) \right\} (1 - \varepsilon) = h^{c}' (1) (1 - \varepsilon). \quad (41)
\]
For \( \delta_2 \) we choose \( y_{\omega}^{\mathbf{p}} \in (0, \varepsilon] \) and get

\[
\delta_2 \geq \max_{s_{\mathbf{p}}} \{ h_c'(s_{\mathbf{p}}) \} \varepsilon = \varepsilon. \tag{42}
\]

To fulfill the derivatives in Definition 11 we use (41), (42) and \( h_c'(1) > \frac{1}{3} \) and with

\[
\varepsilon \leq \delta_2 < \delta_1 \leq h_c'(1)(1 - \varepsilon)
\Rightarrow \varepsilon \leq \frac{1}{4} < \frac{h_c'(1)}{h_c'(1) + 1}
\]

the COD is proven. \( \blacksquare \)
References

S. Fernandez A. Graves and J. Schmidhuber. Multi-dimensional recurrent neural networks. Technical report, IDSIA-04-07, 2007.

Emmanuel Augustin, Jean-marie Brodin, Matthieu Carré, Edouard Geoffrois, Emmanuèle Grosicki, and Françoise Prêteux. RIMES evaluation campaign for handwritten mail processing. In Proc. of the Workshop on Frontiers in Handwriting Recognition, number 1, 2006.

Y. Bengio. Practical recommendations for gradient-based training of deep architectures. CoRR, abs/1206.5533, 2012.

Yoshua Bengio and Yann Lecun. Convolutional networks for images, speech, and time-series. 1995.

C. M. Bishop. Pattern Recognition and Machine Learning. Springer, 2006.

J.-M. Brodin E. Grosicki, M. Carré and E. Geoffrois. RIMES evaluation campaign for handwritten mail processing. In Proc. of the Int. Conf. on Frontiers in Handwriting Recognition, 2008.

J. Schmidhuber F. A. Gers and F. Cummins. Learning to forget: Continual prediction with lstm. Technical report, IDSIA-01-99, 1999.

N. Schraudolph F. A. Gers and J. Schmidhuber. Learning precise timing with lstm recurrent networks. Journal of Machine Learning Research, 3:115–143, 2002.

A. Graves and J. Schmidhuber. Offline handwriting recognition with multidimensional recurrent neural networks. In NIPS, pages 545–552, 2008.

E. Grosicki and H. El-Abed. ICDAR 2011: French handwriting recognition competition. In Proc. of the Int. Conf. on Document Analysis and Recognition, pages 1459–1463, 2011.

Mario Pechwitz, S Snoussi Maddouri, Volker Märgner, Nouredinne Ellouze, Hamid Amiri, et al. Ifn/enit-database of handwritten arabic words. In Proc. of CIFED, volume 2, pages 127–136. Citeseer, 2002.

A. D. Poularikas. The Transforms and Applications Handbook. CRC Press, 2. edition edition, 2000.

J. Schmidhuber S. Hochreiter. Long short-term memory. Neural Computation, 9:1735–1780, 1997.

D. Schlichthärle. Digital Filters. Springer, 2000.