**GI-LSTM: A Generalized and Interpretable LSTM architecture to capture long-term dependencies more efficiently.**

LSTMs have produced competitive results in a wide range of applications [RW3]–[RW7]. One of the main characteristics of the LSTM architecture is the element-wise explicit immediate recursive connection in the LSTM’s cell states, , a recursion that implicitly propagates into past values following an exponentially decay rate [RWW8-RWW10]. The latter exponentially weakening connectivity (EWC) could be partly the cause behind LSTM need from a significantly large number of parameters per layer [RW8]-[RW10], specifically having enough variance among the exponential rates of different cells to capture both, short-term and long-term dependencies.

Different approaches have been proposed to exploit long term dependencies using LSTM as basis, including stacked layers of one ore more NNs, RNNs or LSTMs, designed to handle a variety of datasets [RW11]–[RW14] or to solve specific practical problems [RW15], [RW16]; however, they do not modify the inner LSTM’s mechanism, leaving them potentially susceptible to the EWC problem.

Some approaches have been proposed to reduce the number of parameters needed to increase efficiency, such as the Clockwork RNN (CWRNN) [RW17]–[RW19], which reduces the connectivity between hidden units by dividing the network in modules that activate at different frequencies. Some other alternatives focus on mitigating the (EWC) problem by a more precise control over the decay rate [RWW19]-[RWW20] or the initialization of the LSTM forget gate value [RWW21]. Some recent approaches use the LSTM as basis and modify the inner gating mechanism to increase the propagation of the gradient during training while reducing the number of parameters. Nevertheless, none of these approaches have substantially reduce the number of parameters per layer while keeping or improving the accuracy of the proposed networks.

In a previous related work by the authors [RWW22] it was shown that extending the element-wise explicit connectivity between cell states in LSTM improve the accuracy among a variety of time-series datasets. This modified LSTM, named E-LSTM, resulted in an increase of parameters’ density per unity but reduced the number of units per layer needed, which caused in a substantial reduction in the number of parameters per layer, due to the quadratic dependency of parameters per unit, while matching or surpassing the LSTM’s accuracy. Based on the E-LSTM we proposed a new mechanism that generalizes the E-LSTM approach, named Generalized LSTM (GI-LSTM), by further increasing the explicit recursive connectivity among cell states to directly opposed the EWC, when needed. The GI-LSTM aims to extract more efficiently the very-long term dependencies even when their precise location is unknown. In addition, due to the specific method used to create this connectivity, the GI-LSTM is embedded with an easy-to-use interpretability component into that indicates the statistical relevance it gives to previous cell states.

Experiments using synthetic and real time series data are performed on the proposed GI-LSTM and the standard LSTM to identify and compare their accuracy under different number of parameters.

**GI-LSTM architecture.**

**Motivation for a Generalization.**

As stated in section II relevant information could be left uncaptured for long-term dependencies in the standard LSTM due to the lack of variety in the decay rate among state cells, partly addressed by the E-LSTM approach of increasing the explicit connectivity. Taking a step forward in this approach we increase the connectivity among cell states with respect to their previous values, although not on a specific location in time, in contrast to the E-LSTM, but in an interval of previous values. By using creating this higher connectivity we eliminate the need of the DC algorithm, which can be significantly time consuming when trying to identify very-long term statistical dependencies, even in univariate time series.

As stated in Chapter 4, relevant information for long-term dependencies could be left uncaptured in the standard LSTM due to the lack of variety in the decay rate among forget gates, an issue partly addressed by the E-LSTM approach of increasing the explicit connectivity between cell states. Taking a step forward in this approach the connectivity among cell states with respect to their previous values is increased, eliminating the need for specifying a previous location in time (in contrast to the E-LSTM) and replacing it by a flexible user-defined interval of previous values.

By creating this higher connectivity, the need for the DC algorithm is also removed, which can be time consuming when trying to identify very long-term dependencies, even in univariate time series. Furthermore, the proposed GI-LSTM architecture enables a semi-global interpretation, as defined in [38], specifying which parts of the time series the network gives relevance to, within the user-defined time intervals.

**5.1.2 Forward equations and conceptualization**

To keep the efficiency in the number of weights achieved by the E-LSTM while extending the reach into past values, the new increased connectivity is performed by introducing dynamic ‘memory groups’, , designed to create a balance between the explicit temporal connectivity and the number of parameters.

In more detail, the first memory group, contains explicit information of contiguous lagged cell states, starting at 1; the second memory group, contains information of non-contiguous lagged values of , a structure followed by higher-order memory groups, , up to a maximum number . Similar to the E-LSTM, a weighted forget gate, , is associated to each memory group to allow for dynamism during the information processing and to promote stability during the training phase.

(5.1.1)

(5.1.2)

(5.1.3)

(5.1.4)

(5.1.5)

(5.1.6)

(5.1.7)

(5.1.8)

where is the th row of the matrix ; is the 1-norm; is a vector of ones of dimension . The logic behind the choice of the 1-norm as constraint for the group weights , instead of the 2-norm, will be explained in detail in the next section. However, the following is worth noting: there is no constraint in the sign of the weights and the 1-norm constraint is embedded in the network through a nonunique parametrization with respect to learnable parameters , as shown in (5.1.9).

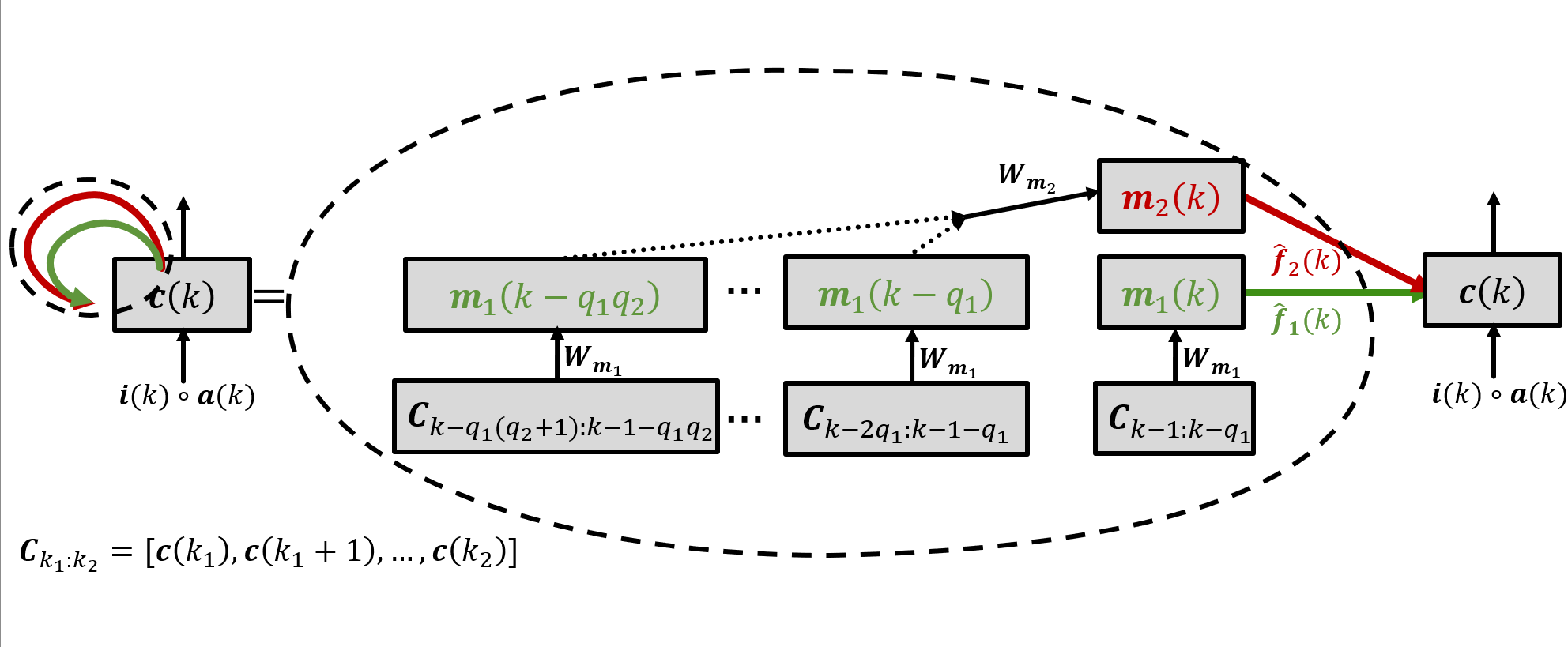
(5.1.9)

where .

In the definition of higher-order memory groups, the design choice of creating lag values that are multiplicative, instead of additive is intended to avoid information overlapping, as shown in Fig. 5.1.1, that could create potentially unnecessary redundancy.

There are three major advantages in the proposed memory-group approach, when compared to the E-LSTM approach. (i) It theoretically allows for reaching very-long term dependencies due to the multiplicative lags in the higher-order memory groups, i.e., , without the aid of the DC algorithm. (ii) Higher-order memory groups compress the information in lower-order memory groups, functioning as filters for past information. (iii) The architecture allows for a dynamic balance between short-term, long-term and very long-term dependencies due to the weighted gates associated to each memory group. Additionally, and following a similar strategy to the E-LSTM architecture, only one forget gate is added per memory group which, when considering the multiplicative reaching effect, maintains the number of parameters low in practice (we will see this in Section 5.3).

Fig. 5.1.1. Simplified graphical representation of the memory-group mechanism in the GI-LSTM, with .



**5.1.3 Backward equations and analysis**

The BP equations are derived following a similar approach to that described in Sections 3.1 and 3.2. Hence, the resulting backward equations linked to the GI-LSTM are as follows:

(5.1.10)

(5.1.11)

(5.1.12)

(5.1.13)

(5.1.14)

(5.1.15)

(5.1.16)

(5.1.17)

where is the index of the last memory group; is the th column of the matrix ; is a normalizing matrix, .

When analyzing the GI-LSTM backward equations it can be observed that in (5.1.14) the individual elements of the gradient do not vanish solely by the fact that the elements might have values close to or equal to 0, specifically due to the non-zero term resulting from the 1-norm, something that would occur if the 2-norm was used instead. Also, it can be noticed that when (5.1.15) is expanded the resulting term is a projection of the gradient over the

current values of . This projection, in the context of (5.1.9), can be interpreted as an opposition to the change in the learnable parameters of the memory groups, . In more detail, the opposition occurs when the change in the learnable parameters produces similar values for , due to a near-to-uniform scaling in .

Consequently, in (5.1.15) all elements in influence the gradient of each of its individual elements, partly producing the effect in the rescaling transformation (5.1.9) that embeds the lack of need for different representations of resulting in the same memory-group weights .

Upon further inspection of (5.1.14) and (5.1.16) two things can be noticed. First, the magnitude of is dependent on but not its direction; second, does not depend on the normalizing factor as seen in (5.1.9). Hence, the gradient could become unnecessarily affected by whenever any of its elements becomes significantly larger than 1, i.e., . Consequently, is removed from (5.1.16), resulting in (5.1.17), and instead a 1-norm row normalization is performed on during each batch (mini-batch) iteration in the training process to ensure , for each row .

(5.1.18)

It is important to notice that despite the removal of the parametrization of in terms of remains useful, due to the previously discussed projection effect in (5.1.15).

**5.1.4 GI-LSTM Interpretability**

The row-normalized memory-group matrices, , in the GI-LSTM architecture represent an easy-to-analyze option to partly access what the network has learned, specifically, what each individual unit,assigns greater relevance to in its memory groups, in terms of the temporal dimension. This interpretability can be achieved by observing, through different plots, the absolute values of the th-row memory-group matrices, i.e., .

The previous approach, although offering a substantial degree of interpretability, would not express how the temporal relevance is distributed across the memory groups when more than one is used, since the relevance is dependent on the dynamic behaviour of the normalized forget gates, . On the other hand, the forget gates’ dynamic behavior increases the difficulty of interpreting the temporal relevance, since it tends to change from one iteration to the next. Therefore, a middle ground between obtaining a more accurate insight into the distributed relevance and handling the dynamism of the forget gates can be achieved by using the time-averaged values of the forget gates (5.1.19), since they represent the overall effect the forget gates have across the forward pass. The process of incorporating the averaged values into the memory-group matrices’ effect, as described in (5.1.20), results in integrated memory-group matrices, , through which a more accurate interpretation of temporal relevance can be obtained. Plotting the row of the integrated memory-group matrices, , produces the desired interpretability for an individual unit .

(5.1.19)

(5.1.20)

where is the sequence length (previously mentioned).

By looking at the individual rows of the integrated matrices it is possible to access the temporal instances considered by individual units. Nevertheless, when interpreting one unit at a time, it might not be straightforward to identify what the whole network assigns relevance to; this is due to the units’ interdependence in the recurrence equations, affecting each other through the network gates. Therefore, if holistic interpretability is desired the mean value of the integrated row-normalized matrices (5.1.21), can be used instead.

(5.1.21)

where is the result of row-normalizing the matrix , i.e., .