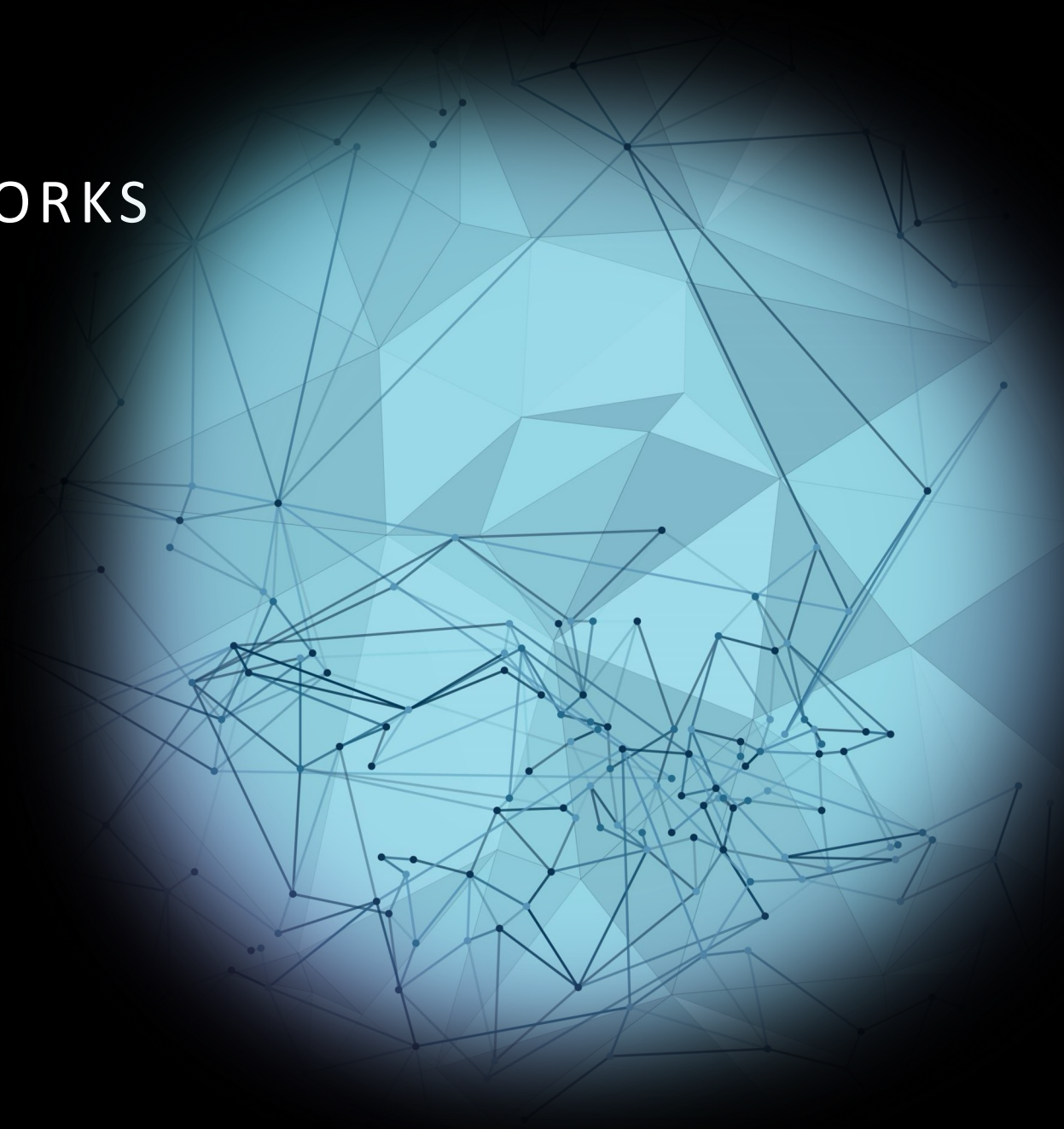
HIERARCHICAL MULTISCALE RECURRENT NEURAL NETWORKS

Davide Bulotta
596782



INTRODUCTION TO THE PROBLEM

Recurrent Neural Networks (RNNs) have trouble understanding complex time patterns in data. In particular the RNN can't learn well more complex time series because they already contains hieriarichal scale.

It's possibile use networks like multiscale RNN for resolve this kind of problem.

This type of networks are more computational efficiently (Less iteration on the high level).

MODEL DESCRIPTION - GENERAL

The networks used for this case of study is the ***hierarchical multiscale recurrent neural network (HM-RNN)***

The idea is learn the hierarchical multiscale structure from temporal data without explicit boundary information.

The boundary detector is a key component of the HM-LSTM model. It produces a binary value that indicates when a segment in the timeline end. When the boundary detector activates (state equal 1), the model treats this as the end of a data segment. The summary representation of the detected segment is passed to the upper layer of the network.

MODEL DESCRIPTION – Key equation (1)

They have implented three different operations: *UPDATE, COPY and FLUSH*

- *UPDATE* : Is similar to update operation on the LSTM except for the execution (according with boundaries).
- *COPY* : operation copies the cell and hidden states of the previous time step.
- *FLUSH* : operation is executed when a boundary is detected.

They compute the cell state with this:

$$\mathbf{c}_t^\ell = \begin{cases} \mathbf{f}_t^\ell \odot \mathbf{c}_{t-1}^\ell + \mathbf{i}_t^\ell \odot \mathbf{g}_t^\ell & \text{if } z_{t-1}^\ell = 0 \text{ and } z_t^{\ell-1} = 1 \text{ (UPDATE)} \\ \mathbf{c}_{t-1}^\ell & \text{if } z_{t-1}^\ell = 0 \text{ and } z_t^{\ell-1} = 0 \text{ (COPY)} \\ \mathbf{i}_t^\ell \odot \mathbf{g}_t^\ell & \text{if } z_{t-1}^\ell = 1 \text{ (FLUSH)}, \end{cases}$$

Finally, afther this, it's possible to compute the hidden state:

$$\mathbf{h}_t^\ell = \begin{cases} \mathbf{h}_{t-1}^\ell & \text{if COPY,} \\ \mathbf{o}_t^\ell \odot \tanh(\mathbf{c}_t^\ell) & \text{otherwise.} \end{cases}$$

MODEL DESCRIPTION – Key equation (2)

Gates values (f_t, i_t, o_t), new candidates value g_t and the boundaries $\tilde{z}_t$ are calculating with this formula:

$$\begin{pmatrix} \mathbf{f}_t^\ell \\ \mathbf{i}_t^\ell \\ \mathbf{o}_t^\ell \\ \mathbf{g}_t^\ell \\ \tilde{z}_t^\ell \end{pmatrix} = \begin{pmatrix} \texttt{sigm} \\ \texttt{sigm} \\ \texttt{sigm} \\ \texttt{tanh} \\ \texttt{hard sigm} \end{pmatrix} f_{\texttt{slice}} \left(\mathbf{s}_t^{\text{recurrent}(\ell)} + \mathbf{s}_t^{\text{top-down}(\ell)} + \mathbf{s}_t^{\text{bottom-up}(\ell)} + \mathbf{b}^{(\ell)} \right), \quad (4)$$

Activation functions:

- sigm: Sigmoid function used for f_t, i_t, o_t
- tanh: Hperbolic tangent function for g_t
- Hard sigm: Approximate sigmoid function used for $\tilde{z}_t$

State components:

- $s_t^{reccurent(l)} = U_l^l h_{t-1}^l$: Recurrent connection from the previous hidden state within the current layer.
- $s_t^{top-down(l)} = z_{t-1}^l U_{l+1}^l h_{t-1}^{l+1}$: Top-down connection from the upper layer, activated only if the boundary is detected at the previous time step.
- $s_t^{bottom-up(l)} = z_t^{l-1} w_{l-1}^l h_t^{l-1}$: Bottom-up connection from the lower layer, activated only if the boundary is detected at the current time step.

The boundary detector plays a crucial role in determining whether to update, copy, or flush the states in a hierarchical manner, thereby efficiently capturing the multiscale structure in temporal data.

MODEL EXPERIMENTS - METRICS

The HM-LSTM model has been evaluated on two main tasks: character-level language modelling and handwriting sequence generation.

The evaluation metric used for the experiments was the:

$$\text{(BPC) } E[\log_2 p(x_{t+1}|x_{\leq t})] \text{ - bits-per-character}$$

For the character-level language modelling the dataset used were:

- Penn Treebank - annotated corpus of American English, primarily derived from the Wall Street Journal.
- Text8 - consists of 100M characters extracted from the Wikipedia corpus.
- Hutter Prize Wikipedia - contains 205 symbols including XML markups and special characters.

Instead for the handwriting sequence generation they used the IAM-OnDB dataset.

MODEL EXPERIMENTS - RESULTS

Character-Level Language Modelling:

Peen Treebank: The model (HM-LSTM) achieved a test BPC score of [***1.24***]. Showing significant improvement when using the step function for hard boundary decisions and the slope annealing trick.

Hutter Prize Wikipedia: The model (HM-LSTM) obtained a test BPC of [***1.32***]. Near at the state-of-the-art results.

Text8: The model (HM-LSTM) obtained a state-of-the-art test BPC of [***1.29***]. Outperforming models like MI-LSTM and BatchNorm LSTM.

Handwriting Sequence Generation:

IAM-OnDB: The model (HM-LSTM) outperformed the standard LSTM on the IAM-OnDB dataset. The Average Log-Likelihood was 1167, the standard LSTM obtained 1081.

CONCLUSIONS

The HM-LSTM model brings new features, like learning hierarchical structures from time-based data without using boundaries, but implementing a special autonomous boundary detector to dynamically find boundaries.

Its strengths include top performance in character-level language modeling, better results than standard LSTMs in handwriting sequence generation, good handling of long-term dependencies, and lower computational demands by updating states only when needed.

However, the model is complex and hard to implement and tune, especially with techniques like the slope annealing trick. While it performs well on certain datasets, more testing on a wider range of tasks is needed to prove its robustness.