SUPRISAL-DRIVEN ZONEOUT

KAMIL ROCKI*, TOMASZ KORNUTA*, TEGAN MAHARAJ†
*IBM Research, Almaden, 650 Harry Rd, San Jose, CA 95120, USA
† Ecole Polytechnique de Montreal, 2900 Boulevard Edouard-Montpetit, Montréal, QC H3T 1J4, Canada
{kmrocki,tkornut}@us.ibm.com, tegan.maharaj@polymtl.ca

MOTIVATION OF THE WORK

- Temporal dynamics can have very different time scales
- The more frequently occurring patterns in lower level neurons should trigger sparser activations in higher level ones.

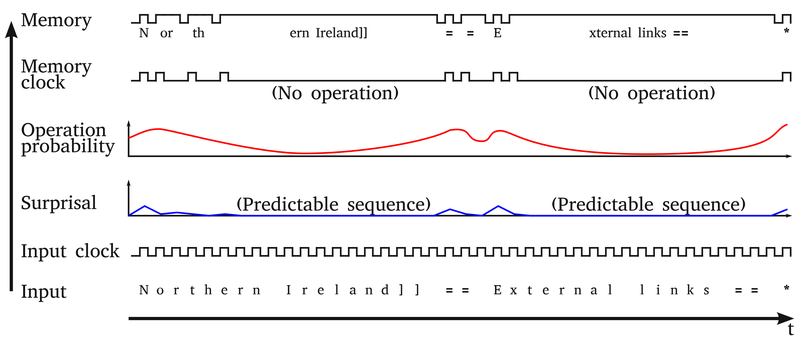
  $\Rightarrow$ **Minimization of a number of neurons being activated**.

MAIN CONTRIBUTIONS

- A new regularization technique based on surpisal-driven feedback,
- The method performs extremely well on the Hutter Prize Wikipedia (enwik8) and linux datasets:
  - beating the current state-of-the-art results,
  - significantly reducing the gap to the best known highly-engineered compression methods.

IDEA OF THE ADAPTIVE ZONEOUT

- Zoneout regularizes by zoning out activations, i.e. *freezes the state* of a cell for a time step with some fixed probability.
- The surprisal-driven feedback enables to change the zoneout rate online, within the scope of a given cell, allowing the zoneout rate to *adapt to current information*.
- As learning progresses, the activations of that cell become less frequent in time and more iterations will just skip memorization, thus the proposed mechanism in fact enables different memory cells to operate on *different time scales*.



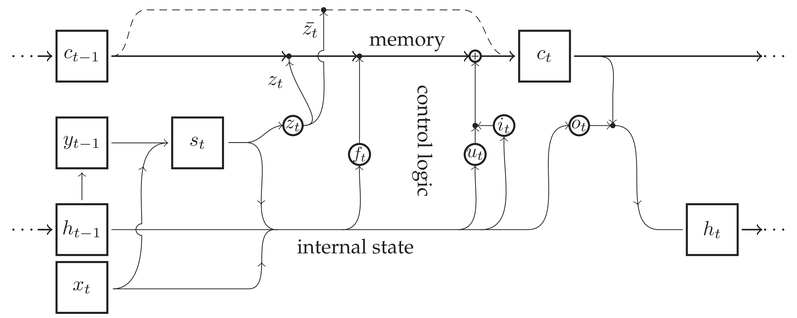
LINKS

The Linux dataset

Large text compression benchmark

cmix website

DIAGRAM OF COMPUTATIONS



ZONEOUT FORMULAS

Calculate the surprisal s_t on the basis of past predictions p_{t-1} and current observations x_t:

$$s_t = \log p_{t-1} \cdot x_t^T \tag{1}$$

Compute the gate activations:

$$f_t = \sigma(W_f \cdot x_t + U_f \cdot h_{t-1} + V_f \cdot s_t + b_f) \tag{2}$$

$$i_t = \sigma(W_i \cdot x_t + U_i \cdot h_{t-1} + V_i \cdot s_t + b_i) \tag{3}$$

$$o_t = \sigma(W_o \cdot x_t + U_o \cdot h_{t-1} + V_o \cdot s_t + b_o) \tag{4}$$

$$u_t = tanh(W_u \cdot x_t + U_u \cdot h_{t-1} + V_u \cdot s_t + b_u) \tag{5}$$

Compute the zoneout rate:

$$S_t = p_{t-1} - x_t \tag{6}$$

$$z_t = min(\tau + |S_t \cdot W_y^T|, 1) \tag{7}$$

Sample a binary mask Z_t according to probability z_t:

$$Z_t \sim z_t \tag{8}$$

New memory state depends on Z_t ($Z_t = 0$ means NOP):

$$c_t = (1 - f_t \odot Z_t) \odot c_{t-1} + Z_t \odot i_t \odot u_t \tag{9}$$

$$\hat{c}_t = tanh(c_t) \tag{10}$$

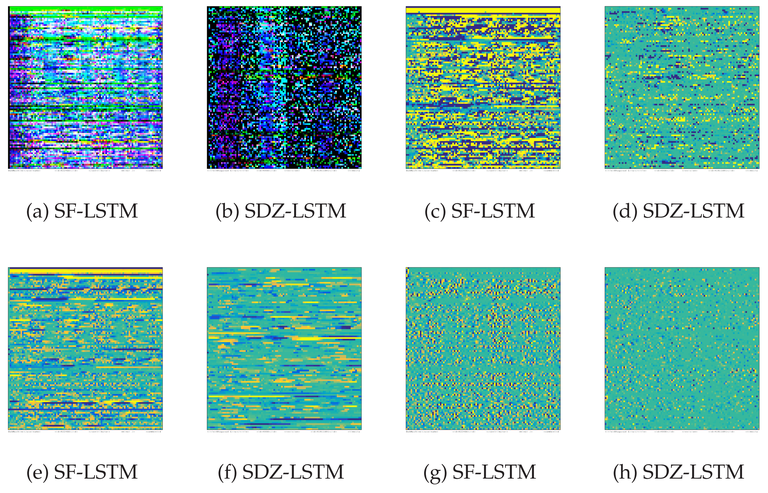
$$h_t = o_t \odot \hat{c}_t \tag{11}$$

Calculate and normalize the outputs:

$$y_t = W_y \cdot h_t + b_y \tag{12}$$

$$p_t^i = \frac{e^{y_t^i}}{\sum_i e^{y_t^i}} \tag{13}$$

COMPARISON OF ACTIVATIONS AND STATES



(a) SF-LSTM (b) SDZ-LSTM (c) SF-LSTM (d) SDZ-LSTM

(e) SF-LSTM (f) SDZ-LSTM (g) SF-LSTM (h) SDZ-LSTM

- left to right, 100 time steps, Y-axis represents cell index
- (a,b) i/o/f gate activations, colors indicate: Red – write (i), Green – read (o), Blue - erase (f), White – all, Black – no operation
- (c,d) Hidden state: Green (0) – no operation, Blue (-1), Yellow (+1)
- (e,f) Memory cell state
- (g,h) Memory cell change (L1-norm)

RESULTS: PBC ON THE ENWIK8 DATASET

mRNN	1.60	GF-RNN	1.58
Grid LSTM	1.47	Layer-normalized LSTM	1.46
Standard LSTM	1.45	MI-LSTM	1.44
Array LSTM	1.40	HM-LSTM	1.40
HyperNetworks	1.38	SF-LSTM	1.37
RHN	1.32	Surprisal-Driven Zoneout	**1.31**

Best lossless data compression program: cmix v11 1.245

RESULTS: PBC ON THE LINUX DATASET

SF-LSTM	1.38	Surprisal-Driven Zoneout	**1.18**

ACKNOWLEDGEMENTS

This work has been supported in part by the Defense Advanced Research Projects Agency (DARPA) program "Saccadic Vision and Hierarchical Temporal Memory" realised under the contract number N66001-15-C-4034.