Factorization tricks for LSTM networks

Oleksii Kuchaiev (okuchaiev@nvidia.com) & Boris Ginsburg (bginsburg@nvidia.com)

We present two simple ways of reducing the number of parameters and accelerating the training of large Long Short-Term Memory (LSTM) networks: the first one is "matrix factorization by design" of LSTM matrix into the product of two smaller matrices, and the second one is partitioning of LSTM matrix, its inputs and states into the independent groups. Both approaches allow us to train large LSTM networks significantly faster to the state-of the art perplexity. On the One Billion Word Benchmark we improve single model perplexity down to **23.36**.

Factorized LSTM cell (F-LSTM)

Standard LSTM cell computation uses computationally expensive affine transform *T*.
F-LSTM cell attempts to approximate *T* using “matrix factorization by design”:

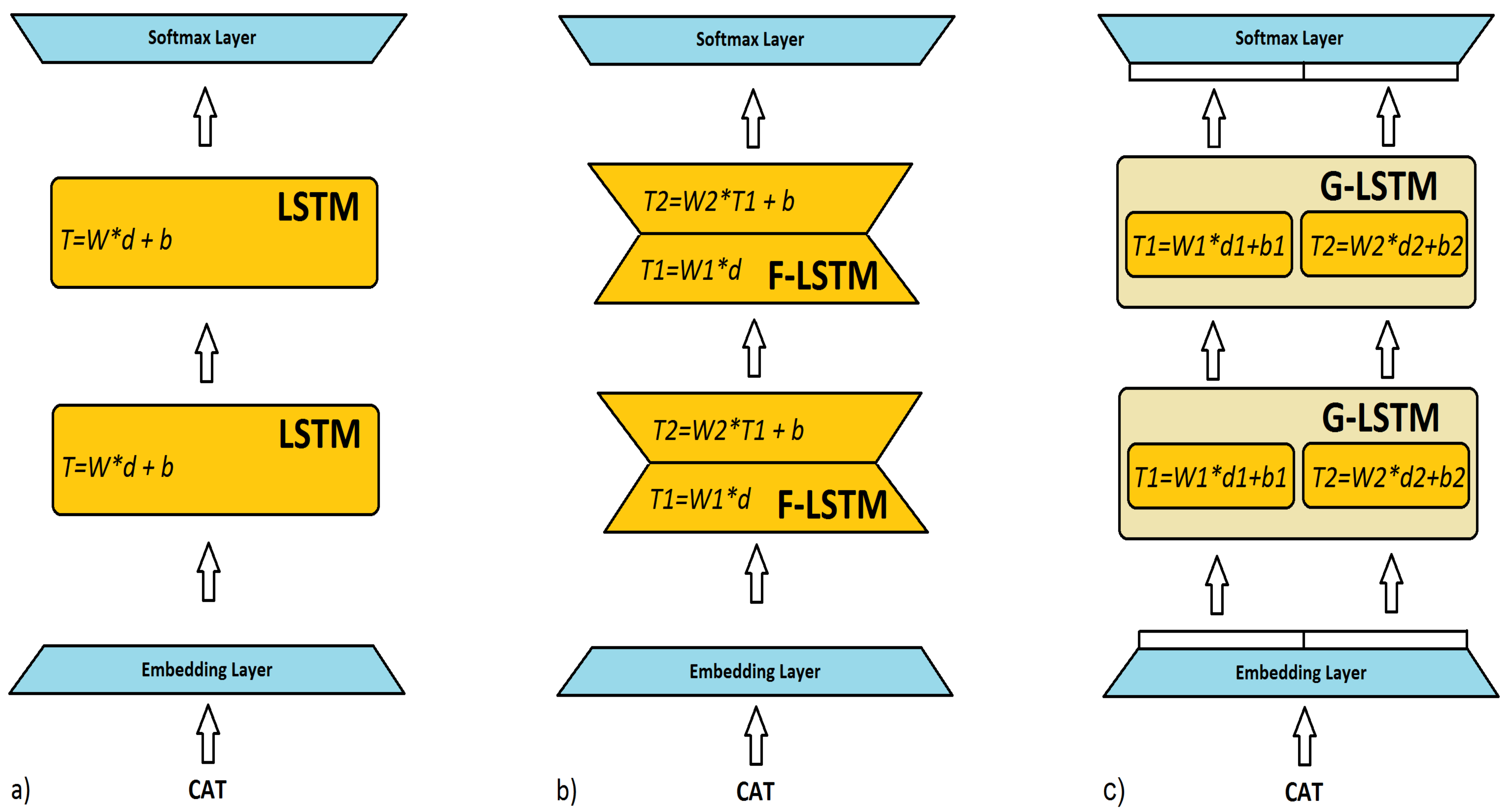
$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} sigm \\ sigm \\ sigm \\ tanh \end{pmatrix} T \begin{pmatrix} x_t \\ h_{t-1} \end{pmatrix} \quad \begin{array}{l} T = W * [x_t, h_{t-1}] + b \\ W \approx W2 * W1 \end{array}$$

Group LSTM cell (G-LSTM)

Group LSTM cell postulates that some parts of the input x and hidden state h can be thought of as independent feature groups.
Hence, it splits affine transform T into several smaller transforms responsible for separate groups:

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \left(\begin{pmatrix} sigm \\ sigm \\ sigm \\ tanh \end{pmatrix} T^1 \begin{pmatrix} x_t^1 \\ h_{t-1}^1 \end{pmatrix}, \ldots, \begin{pmatrix} sigm \\ sigm \\ sigm \\ tanh \end{pmatrix} T^k \begin{pmatrix} x_t^k \\ h_{t-1}^k \end{pmatrix} \right)$$

Language Models



(a) regular LSTM (b) F-LSTM (c) GLSTM layers with 2 groups

Equations inside cells show what kind of affine transforms are computed by those cells at each time step. Here d = (x, h) for models without groups and d1 = (x1 , h1), d2 = (x2 , h2) for model with two groups.

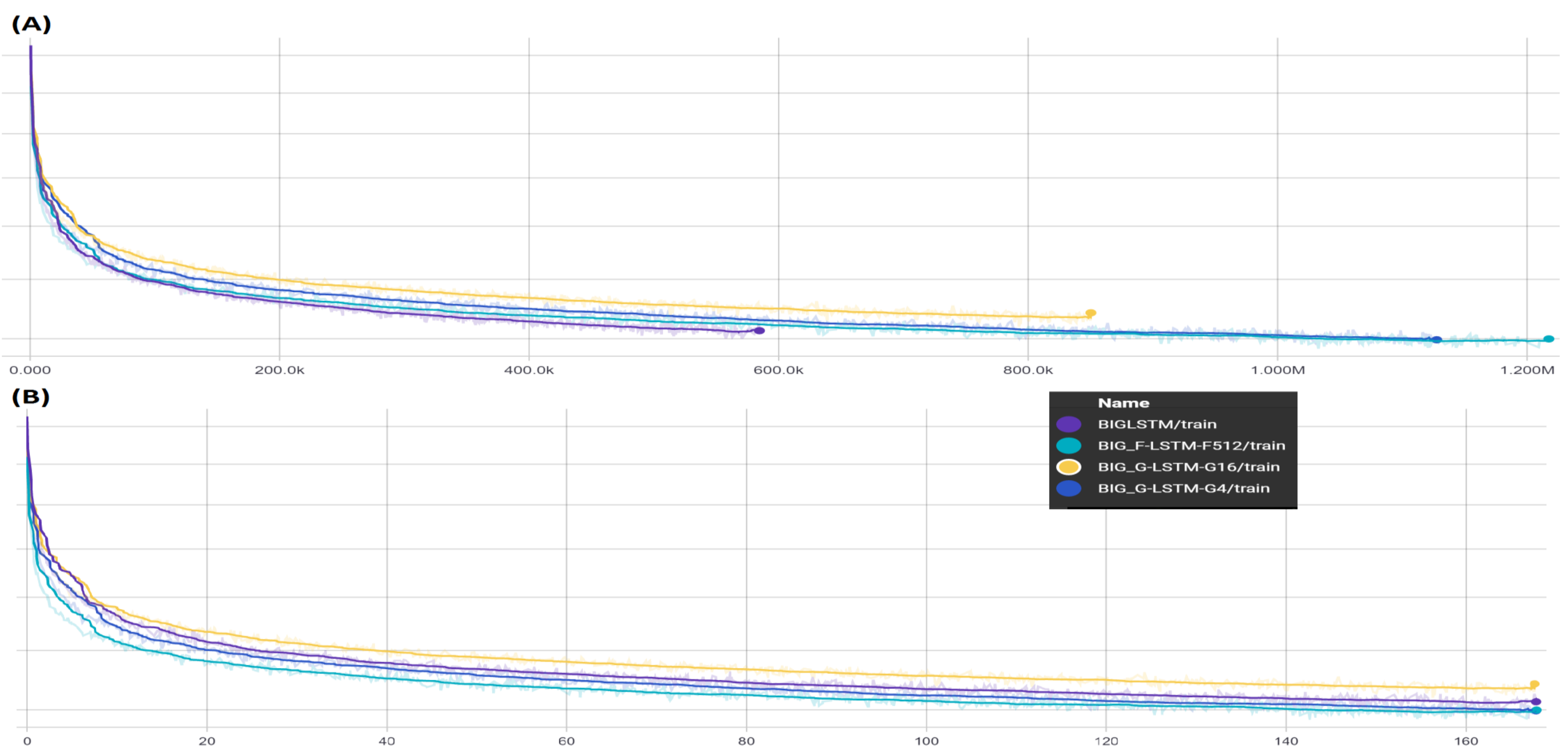
[1] “Exploring the limits of language modelling” Jozefowicz et al.

Results

One Billion Words benchmark after one week of training using DGX-1:

Model	Perplexity	Step	RNN parameters	Words/sec
BIGLSTM [1] baseline	31.001	584.6K	83,951,616	20.3K
BIG F-LSTM F512	28.11	1.217M	51,445,760	42.9K
BIG G-LSTM G-4	28.17	1.128M	33,619,968	41.1K
BIG G-LSTM G-16	34.789	850K	21,037,056	41.7K

BIG G-LSTM G-4 perplexity is 24.29 after 2 weeks and 23.36 after 3 weeks.



Y-axis: training loss log-scale, X-axis: for (A) - steps, for (B) - hours.
BIGLSTM baseline, BIG G-LSTM-G4, BIG G-LSTM-G16, and BIG F-LSTM-F512
All trained for one week.