FastGRNN: A Fast, Accurate, Stable and Tiny Kilobyte Sized Gated Recurrent Neural Network

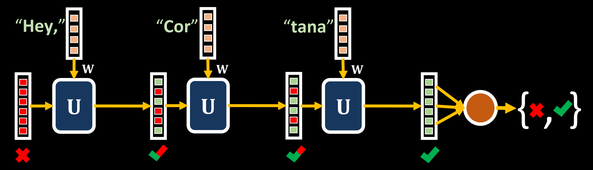
Aditya Kusupati † Manish Singh § Kush Bhatia‡ Ashish Kumar ‡ Prateek Jain † Manik Varma †

† Microsoft Research India § Indian Institute of Technology Delhi ‡ University of California Berkeley

Code: **https://github.com/Microsoft/EdgeML**

Recurrent Neural Networks (RNNs)

State-of-the-art for analyzing sequences & time series



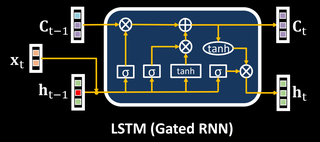
$$\nabla = \quad f(\dots, \mathbf{U}^T = \mathbf{Q}\begin{bmatrix} (1+\lambda)^T & & \\ & \ddots & \\ & & (1-\gamma)^T \end{bmatrix}\mathbf{Q}^\top, \dots)$$

Limitations of Existing RNNs

- *Traditional RNNs* : Training is unstable
- *Unitary RNNs* : Expensive to train and lack accuracy
- *Gated RNNs* : Large model size and prediction costs

$$\nabla = \quad f(\dots, \mathbf{U}^T = \mathbf{Q}\begin{bmatrix} 1^T & & \\ & \ddots & \\ & & 1^T \end{bmatrix}\mathbf{Q}^\top, \dots)$$

Unitary RNNs



LSTM (Gated RNN)

Our Solutions: **FastRNN** for provably stable training & **FastGRNN** for state-of-the-art performance in 1-6KB size models

Gradients & Theorems

$$\frac{\partial L}{\partial \mathbf{U}} = \alpha \sum_{t=0}^{T} \mathbf{D}_t \left(\prod_{k=t}^{T-1} (\alpha \mathbf{U}^\top \mathbf{D}_{k+1} + \beta \mathbf{I}) \right) (\nabla_{\mathbf{h}_T} L) \mathbf{h}_{t-1}^\top$$

$$\frac{\partial L}{\partial \mathbf{W}} = \alpha \sum_{t=0}^{T} \mathbf{D}_t \left(\prod_{k=t}^{T-1} (\alpha \mathbf{U}^\top \mathbf{D}_{k+1} + \beta \mathbf{I}) \right) (\nabla_{\mathbf{h}_T} L) \mathbf{x}_t^\top$$

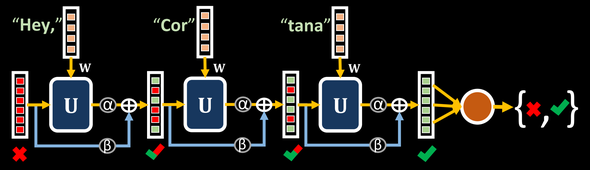
L is the loss function
T is the # timesteps

$$\mathbf{D}_k = \text{diag}(\sigma'(\mathbf{W}\mathbf{x}_k + \mathbf{U}\mathbf{h}_{k-1} + \mathbf{b}))$$

- Setting $\alpha \approx O(1/T), \beta = 1 - \alpha$ stabilizes the gradients
- FastRNN has convergence rate and generalization error upper bounds independent of T

FastRNN

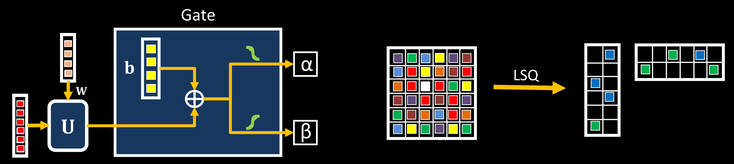
- Provably stable training with a residual connection having 2 additional scalars
- *Accuracy*: RNN ≪ Unitary RNNs < FastRNN < Gated RNNs



$$\nabla = \quad f(\dots, (\alpha\mathbf{UD} + \beta\mathbf{I})^T = \mathbf{Q}\begin{bmatrix} (\beta+\alpha\|\mathbf{UD}\|)^T & & \\ & \ddots & \\ & & (\beta-\alpha\|\mathbf{UD}\|)^T \end{bmatrix}\mathbf{Q}^\top, \dots)$$

FastGRNN

- Extend α & β from scalars to vector gates
- Make **U** and **W** low-rank (L), sparse (S) and quantized (Q)
- *Accuracy*: RNN ≪ Unitary RNNs < Gated RNNs ≈ FastGRNN



Architectures' Equations

Simple RNN

$$\mathbf{h}_t = \sigma(\mathbf{W}\mathbf{x}_t + \mathbf{U}\mathbf{h}_{t-1} + \mathbf{b})$$

FastRNN

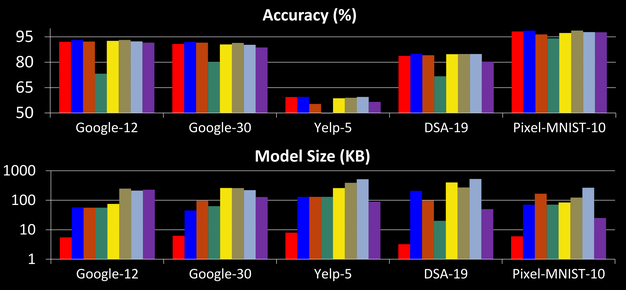
$$\tilde{\mathbf{h}}_t = \sigma(\mathbf{W}\mathbf{x}_t + \mathbf{U}\mathbf{h}_{t-1} + \mathbf{b})$$
$$\mathbf{h}_t = \alpha\tilde{\mathbf{h}}_t + \beta\mathbf{h}_{t-1}$$

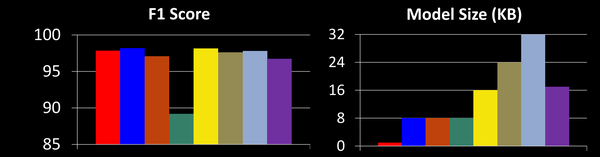
FastGRNN

$$\mathbf{z}_t = \sigma(\mathbf{W}\mathbf{x}_t + \mathbf{U}\mathbf{h}_{t-1} + \mathbf{b}_z)$$
$$\tilde{\mathbf{h}}_t = \tanh(\mathbf{W}\mathbf{x}_t + \mathbf{U}\mathbf{h}_{t-1} + \mathbf{b}_h)$$
$$\mathbf{h}_t = (\zeta(1-\mathbf{z}_t) + \nu) \odot \tilde{\mathbf{h}}_t + \mathbf{z}_t \odot \mathbf{h}_{t-1}$$

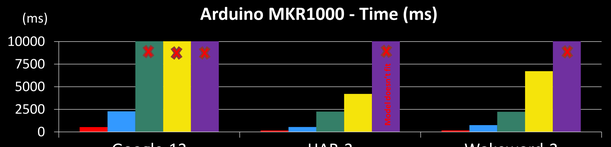
$0 \le \alpha, \beta, \zeta, \nu \le 1$, and are trainable scalars, parameterized by the sigmoid function
$\sigma(\cdot)$ can be any non-linearity
$\odot$ is Hadamard product

Results

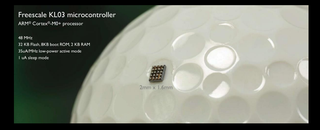




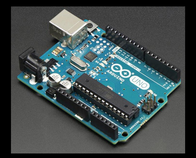
Wakeword-2: "Hey Cortana" in 1KB



Proposed : FastGRNN FastGRNN-LSQ FastGRNN-Q FastRNN
Existing : Simple RNN UGRNN GRU LSTM Spectral-RNN



ARM Cortex M0+ at 48 MHz & 35 μA/MHz with 2 KB RAM & 32 KB read only Flash



8 bit ATmega328P Processor at 16 MHz with 2 KB RAM & 32 KB read only Flash

Billions of these resource-constrained devices form the IoT ecosystem