An Efficient and Accurate Gated RNN for Execution under TFHE

Rickard Brännvall[1] and Andrei Stoian[2]

[1]Computer Science Department, RISE Research Institutes of Sweden, rickard.brannvall@ri.se
[2]Machine Learning Group, Zama, Paris, France, andrei.stoian@zama.ai

In Short

Basic idea: Let $u_t = Wx_t + Uh_{t-1} + b$ and change gated RNNs according to

$$\sigma(u_t) \odot h_{t-1} \rightarrow \left(u_t^- + h_{t-1}\right)^+ \tag{1}$$

such that multiplication is replaced with addition and sigmoid with ReLU.

Observation: In the limits the modified gates behaves like the conventional.

Result: The thus modified GRU and LSTM works just as well as the original.

Potential: Computate efficient gated RNNs with natural integer quantization.

Future work: Does this architecture generalize to transformers? Answer: Yes.

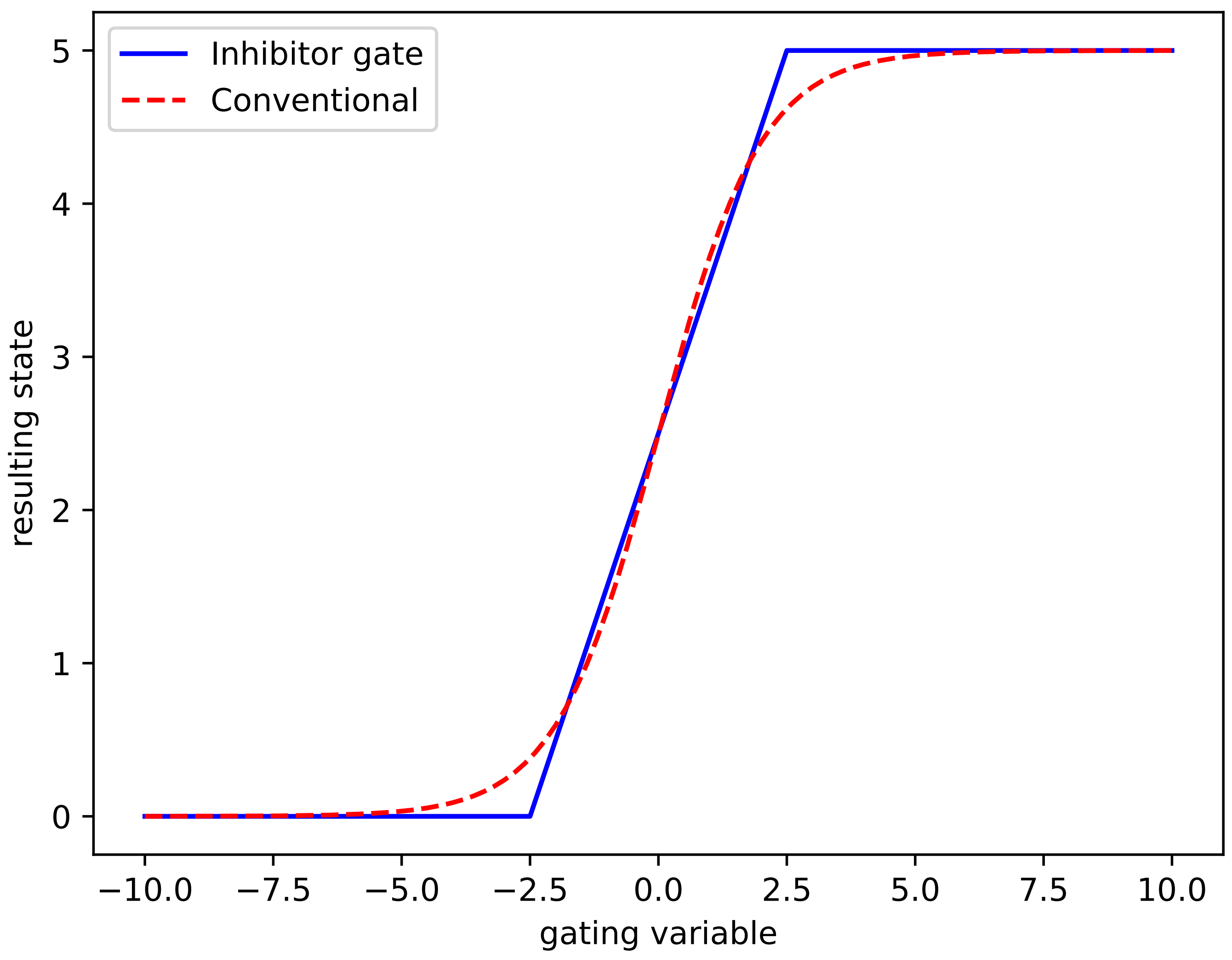

Figure 1: Comparison of the Inhibitor gate with the Sigmoid function and multiplication based gate used for the conventional GRU and LSTM.

Abstract

We implement and test the recently proposed Inhibitor gate for Recurrent Neural Networks (RNNs) that is both efficient and accurate under Homomorphic Encryption. Our numerical experiments shows order of magnitude improvements.

Background

Gated RNNs such as LSTM and GRU have applications in numerous real-world use cases for sequential data, such as time-series analysis and natural language processing, due to their ability to capture long-term dependencies.

Conventionally, the update based on current inputs and the previous state history is each multiplied with the dynamic gate values and combined to compute the next state. However, as it is a multiplication between two variables that depend on the input, it can be computationally expensive, especially under homomorphic encryption, where it involves a multiplication between two cipher text variables.

Method

The novel gating mechanism replaces the multiplication and sigmoid function of the conventional RNN with addition and ReLU activation. For the GRU we have

$$h_t = \sigma(u_t) \odot h_{t-1} + (1 - \sigma(u_t)) \odot \hat{h}_t \tag{2}$$

where u_t is defined as in equation 1, $\hat{h}_t$ is the proposal and h_{t-1} is the prevous state. The update rule for the i-GRU becomes

$$h_t = (h_{t-1} + u_t^-)^+ + (\hat{h}_t - u_t^+)^+ \tag{3}$$

where we write $\mathrm{ReLU(u)} = u^+$ for shorter notation.

Results

Numerical experiments using the TFHE Concrete library on three synthetic benchmark tests demonstrate that our algorithm outperforms a conventional gated RNN showing at least an order of magnitude faster execution already at 4-bit quantization. At a fixed computational budget, we get 7-bit precision with the novel gate that executes faster than the 4-bit conventional gate.

Table 1: Mean execution time in seconds on CPU over 100 trials for gated RNN solvers of the synthetic addition problem (task 1) and the copying memory problem (task 2). The table compares the proposed additive inhibitor gate (Add) with the conventional multiplicative (Mul) at different precision.

	Add 4b	Mul 1b	Mul 2b	Mul 3b	Mul 4b
Task 1	0.5	0.44	0.62	2.03	31.5
Task 2	1.13	1.61	2.33	10.1	104.

Table 2: Mean execution time in seconds on CPU for one step of a scalar gated RNN with 1-bit quantized weights and n bit precision for input and state (100 trials), where the proposed additive gate (Add) is compared with the conventional multiplicative gate (Mul) at different precision.

n	Add 4b	Mul 1b	Mul 2b	Mul 3b	Mul 4b
2	0.13	0.13	0.21	0.26	0.63
3	0.18	0.19	0.26	0.6	2.02
4	0.22	0.22	0.62	2.1	5.57
5	0.52	0.53	2.03	5.56	25.58
6	1.76	1.79	5.48	25.94	
7	4.64	4.84	26.39		
8	21.06	22.49			

Conclusion

The gating mechanism employed in this paper may enable privacy-preserving AI applications based on gated RNNs operating under homomorphic encryption by avoiding the multiplication of encrypted variables.

Related work. In parallel work we have proposed the Inhibitor mechanism also for Transformers and investigated its numerical performance under FHE.