On the Maximum Function in Stochastic Neural Networks

Florian Neugebauer[€], Ilia Polian[§] and John P. Hayes[$]

[€]Faculty of Computer Science & Mathematics
University of Passau,
Innstr. 43, D-94032 Passau, GER
florian.neugebauer@uni-passau.de

[§]Institute of Computer Architecture and
Computer Engineering, University of Stuttgart,
Pfaffenwaldring. 47, D-50679 Stuttgart, GER
ilia.polian@informatik.uni-stuttgart.de

[$]Computer Engineering Laboratory
University of Michigan,
Ann Arbor, MI 48109, USA
jhayes@umich.edu

***Abstract*—Stochastic circuits (SCs) offer significant area, power and energy benefits at the cost of computational inaccuracies. They have been used for various numerical applications such as image and signal processing. They have received particular attention recently in neural networks (NNs) where arithmetic functions can be efficiently implemented by stochastic computing. Many NNs use the maximum function, e.g., in the max-poling layer of convolutional NNs, and for the ReLU activation function. However, an efficient accurate SC for the max function has been lacking, and an approximate workaround is currently employed. We propose here an SC for max that is correlation insensitive and produces an exact result.**

Keywords: Emerging technologies, simulation, stochastic computing, neural networks.

I. Introduction

Stochastic computing [3][5] can provide compact, error-tolerant and low-power implementations of complex functions. It has been proposed for numerous applications such as low-density parity codes [6], image processing [2] and signal processing [8]. In recent years, it has gained a lot of attention in the field of neural networks [4][7]. A key operation in NNs is multiplication of connection weights by activation values and the summation of the resulting products in each neuron. Multiply and add can be performed very efficiently in stochastic computing. Other components of NNs such as activation functions require more sophisticated SCs. Non-linear activation functions (e.g., tanh) are usually realized as finite state machines in stochastic computing [4], which poses problems because they are sensitive to autocorrelation of stochastic numbers (SNs).

Convolutional NNs (CNNs) are a very successful type of NN, especially in the field of image classification. They consist of convolutional layers, pooling layers and dense layers. While the convolutional layers and dense layers can be implemented in a straightforward way with SCs, the pooling layers are often problematic. Max-pooling has shown better performance than average pooling, but no efficient SC to calculate the maximum of uncorrelated, i.e., independent SNs has been proposed so far. SCs for correlated SNs are known; however, they introduce significant hardware overhead, as other operations such as multiplication require uncorrelated SNs and the conversion between correlated and uncorrelated SNs is hardware intensive. Therefore, current stochastic NNs with max-pooling employ a workaround which only returns an approximate maximum [7]. We propose an SC for the maximum function that is correlation insensitive and provides exact results instead of approximations, while using comparable hardware area.

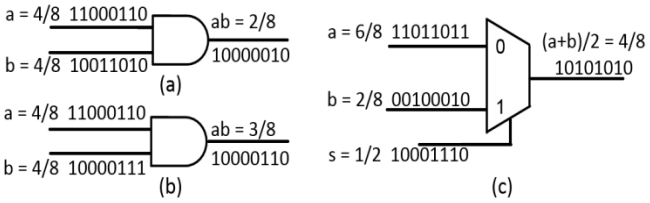

Fig. 1: (a) Unipolar multiplication; (b) inaccurate multiplication due to correlation; (c) scaled addition.

II. Stochastic Computing

Stochastic computing processes bit-streams called stochastic numbers (SNs). An SN of length n with n_1 1s and n_0 0s represents the value n_1/n in unipolar format and $(n_1 - n_0)/n$ in bipolar format. For example, the SN 00110100 has the unipolar value $3/8$ and the bipolar value $-2/8$. The order of 1s in the SN has no influence on its value. With this representation, multiplication and addition can be performed efficiently according to Fig. 1. Unipolar multiplication is done using a single AND gate, while scaled addition is implemented by a MUX. Fig. 1(b) shows how correlation between SNs can lead to inaccurate results for multiplication. Correlation describes the degree of similarity between two SNs and can be measured using the SCC metric [1]. A high ratio of overlapping 1s signifies high dependency and therefore high correlation, up until a maximum SCC of 1 when all 1s overlap.

This type of correlation can sometimes be exploited to implement certain functions such as the absolute difference of two SNs [2], but also for a maximum function, which can be

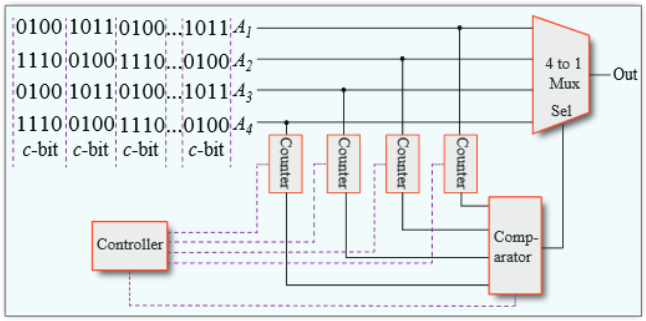

Fig. 2: Approximate maximum SC used in [7].

performed by an OR gate, when the input SNs have an SCC of 1. However, most SCs require uncorrelated SNs as inputs. It can be easily seen that the multiplication in Fig. 1 does not produce correct results when operated with maximally correlated SNs. Unfortunately, correlated SNs cannot be easily converted to uncorrelated SNs and vice-versa. The SNs have to be converted to binary first and then converted back to the stochastic domain again, which introduces significant delay and hardware overhead. This solution for the maximum function is therefore not practical in stochastic NNs.

III. STOCHASTIC MAXIMUM CIRCUIT

Fig. 2 shows an SC for calculating the maximum of four input SNs used by the max-pooling layer in [7]. The number of 1s in each c-bit block of the SNs is counted and the index of the block with the most 1s determines the SN from which the next c bit are passed through the MUX to the output. This method is based on the fact that, on average, the SN that is the global maximum is most of the time also the local maximum in each block. This circuit is however only approximate: the first block has to be chosen randomly (or through some other method that does not introduce a delay) and if the SN values do not differ by a large margin, it is unlikely that the largest SN also has the most 1s in every block.

Fig. 3 shows our proposed maximum circuit for an example with three input SNs. Each SN X_i has a counter assigned to it that tracks the difference $C_i = \max(X_{1,m}, X_{2,m}, X_{3,m}) - X_{i,m}$ where $X_{i,m}$ is the number of 1s in SN i after m bits. The circuit outputs a 1, when at least one of the inputs is 1 and at least one of the counters of those inputs is at zero (in which case $O_i = 1$ for this input). In other words, if an input SN has a 1 in clock cycle m and the value of this SN is equal to the maximum of all input SNs in clock cycle $m-1$, the value of this SN has to be the maximum in clock cycle m, so the circuit outputs a 1. The output Z is therefore always exactly the maximum of the inputs, independent of any correlation.

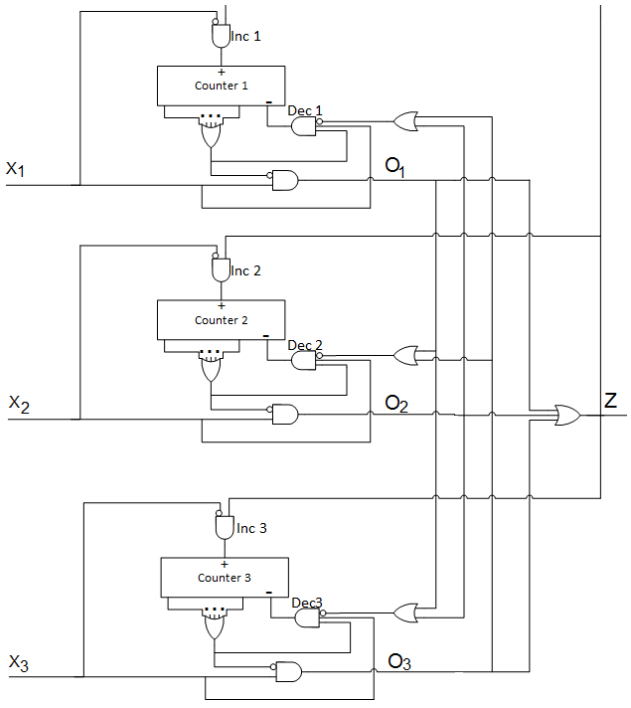

Fig. 3: Proposed maximum SC; an arbitrary number of inputs is possible by adding more counter blocks.

Network type	Classification accuracy
Binary	93.2
SC - Fig. 2, block size 32	87.0
SC - Fig. 2, block size 128	89.2
SC - Fig. 2, block size 256	90.6
SC - Fig. 2, block size 512	89.6
SC - Fig. 3	90.8

Table 1: Accuracy for handwritten digit classification using different SCs, fully binary CNN for comparison.

In order to evaluate the difference in performance between the circuit in Fig. 2 and our proposed circuit in an application, we simulated a small CNN for MNIST handwritten digit classification. The network consisted of one convolutional layer with 16 feature maps using 3×3 size filters, a ReLU activation layer, a 2×2 max-pooling layer (all implemented as SCs) and a fully connected layer, which was implemented using stochastic multipliers and parallel counters. The network was trained only once, every type of stochastic network was therefore using the same weights and biases. SN length was 4.096, this large number was chosen to reduce the impact of random effects. The results are shown in table 1.

It can be seen that our circuit performs better than the approximate version in a small CNN. Furthermore, block sizes that are too small or too large lead to significantly worse results for the latter. At this point, no convenient method to determine the optimal block size is known, it has to be found through trial and error.

IV. CONCLUSION AND FUTURE WORK

We have proposed a correlation insensitive SC for the maximum function. Our circuit calculates the exact maximum, unlike currently used circuits that only approximate it. Early simulation results for small CNNs show that our design outperforms the current maximum SCs in these applications. To further evaluate our circuit, we plan to simulate larger networks to assess the impact of an accurate max-pooling layer on the output accuracy in CNNs with more layers. Furthermore, we intend to perform a detailed analysis on the circuit's output SN, e.g., to address autocorrelation.

V. REFERENCES

[1] Alaghi, A. and J.P. Hayes. Exploiting correlation in stochastic circuit design. *ICCD,* 39-46, 2013.

[2] Alaghi A., C. Li and J.P. Hayes. Stochastic circuits for real-time image-processing applications. *DAC*, 2013.

[3] Alaghi A. and J.P. Hayes. Survey of stochastic computing. *ACM Trans. Embedded Comp. Syst.*, 12: article 92, 2013.

[4] Brown B.D. and H.C. Card. Stochastic neural computation I: Computational elements. *IEEE Trans.. Computers*, 50: 891-905, 2001.

[5] Gaines B.R. Stochastic computing systems. *Advances in Information Systems Science*, 2: 37-172, 1969.

[6] Naderi A. et al. Delayed stochastic decoding of LDPC codes. *IEEE Trans. Signal Proc.*, 59: 5617-5626, 2011.

[7] Ren, A. et al. SC-DCNN: highly-scalable deep convolutional neural network using stochastic computing. *ASPLOS*, 405-418, 2017.

[8] Wang, R. et al. Design, evaluation and fault-tolerance analysis of stochastic FIR filters. *Microelectronics Reliability* 57, 111-127, 2016.