A Case Study in Noise Enhanced Computing: Noisy Gradient Descent Bit Flip Decoding

Chris Winstead*[1], Gopalakrishnan Sundararajan†[1], and Emmanuel Boutillon‡[2]

[1]Department of Electrical and Computer Engineering, Utah State University, Logan, Utah, USA
[2]Lab-STICC, Université de Bretagne Sud, Lorient, France

1 Introduction

This abstract presents the authors' recent results on noise-enhanced computation, with specialized application to an error-correction decoding algorithm widely used in digital communication systems. Random defects are an increasingly troublesome problem for densely integrated electronic circuits. In this work we focus specifically on noise-induced transient upsets that can occur in nano-scale switching devices. There has recently been increasing interest in noise-tolerant design, since there is continual pressure to reduce the signal energy in electronic circuits, and eventually it will be necessary to perform computation at a very low signal-to-noise ratio. Some researchers propose sacrificing reliability in order to reduce energy consumption, which is acceptable for some applications that are inherently tolerant to momentary or intermittent faults (audio or image processing, for example).

In contrast to noise-tolerant design, some researchers argue for *noise enhanced* solutions by using algorithms that benefit in some way from random upsets. Noise-enhanced solutions are partly motivated by biological examples, particularly neural signal processing, in which there are several instances where neural function is enabled or improved by some form of noise [3]. In this work, we present a case study showing how one algorithm is transformed into a noise-enhanced form. The transformation is not quite automatic, but nevertheless sheds light on a heuristic procedure that may be applied more broadly. The transformation is based on the *stochastic gradient ascent* heuristic, which is a well-known method for constrained optimization problems. The challenge is to transform the problem into a form suitable for applying the stochastic gradient ascent heuristic.

2 Gradient Descent Bit-Flip (GDBF) Decoding

LDPC codes are a high-performance solution for error correction, used in many standards for wireless networking, mobile data communication, digital broadcast, and satellite communication. These codes have also been used in disc drive and solid-state memory applications, among others. The goal of using LDPC codes is to reduce the system's bit error rate (BER) when data is communicated across a noisy channel. A simplified version of the LDPC decoding problem is stated as follows (see MacKay [2], chapter 47, for a tutorial introduction). A vector of arbitrary data is encoded to produce a *codeword* c, which is a column vector of n bits generated subject to the constraint $Hc = 0$, where H is an $m \times n$ matrix of parity-check constraints — there are m parity constraints (rows) applied to n bits (columns) — and the multiplication Hc is performed modulo-2. The codeword is then modulated to obtain a message $\hat{c} = 1 - 2c$. After modulation, each parity-check constraint can be expressed as the product $s_j = \prod_{i \in \mathcal{N}(j)} \hat{c}_i = +1$, where s_j is referred to as the j^{th} *syndrome* and $\mathcal{N}(j) = \{k : h_{jk} = 1\}$.

The message is transmitted across a noisy channel that adds a vector of independent, identically distributed Gaussian noise, n, to the message. At the receiver, a vector of samples, y, is obtained, given by $y = \hat{c} + n$. The maximum-likelihood (ML) decoding problem is to find the decision d that

*chris.winstead@usu.edu
†gopal.sundar@aggiemail.usu.edu
‡emmanuel.boutillon@univ-ubs.fr

maximizes the correlation

$$d = \arg\max_{x \in \hat{C}} \sum_{i=1}^{n} y_i x_i, \tag{1}$$

where $\hat{C}$ is the *codebook*, i.e. the set of all possible messages subject to the parity constraint.

In order to solve the ML decoding problem with minimal complexity, many algorithms have been devised. One low-complexity method is the gradient descent bit-flip (GDBF) algorithm devised by Wadayama [5]. To obtain the GDBF algorithm, the ML problem is modified by adding the code's syndrome information as a penalty term

$$d = \arg\max_{x \in \hat{C}} \sum_{i=1}^{n} y_i x_i + \sum_{j=1}^{m} s_j \approx \arg\max_{x \in \{-1,+1\}^n} \sum_{i=1}^{n} y_i x_i + \sum_{j=1}^{m} s_j. \tag{2}$$

This approximation is motivated by the intuition that the summation $\sum_{j=1}^{m} s_j$ is maximized when $x \in \hat{C}$, since all s_j are equal to $+1$ in this case. When $x \notin \hat{C}$, at least one of the s_j is equal to -1 due to parity violation.

Thanks to the penalty term, the gradient descent heuristic may be applied to the objective function $f(x) = \sum_{i=1}^{n} y_i x_i + \sum_{j=1}^{m} s_j$. By taking the gradient of $f(x)$ with respect to each symbol, Wadayama defined the *inversion function* as

$$\Delta_i(x) \triangleq x_i \frac{\delta f}{\delta x_i} = x_i \left(y_i + \sum_{j=1}^{m} \frac{\delta s_j}{\delta x_i} \right) = x_i y_i + \sum_{j \in \mathcal{M}(i)} s_j, \tag{3}$$

where $\mathcal{M}(i) = \{j : h_{ij} = 1\}$.

Based on the gradient (3), Wadayama proposed a steepest-ascent procedure in which we iteratively flip the sign of bits for which $\Delta_i(x) < 0$. The most stable approach is to flip a single bit in each iteration, corresponding to the most negative value of $\Delta_i(x)$. In order to increase the speed of convergence, Wadayama described a *multi-bit* variant in which bits are flipped in parallel via a threshold operation. In the multi-bit GDBF algorithm, the sign of any bit x_i is flipped if $\Delta_i(x) < \theta$, where $\theta < 0$ is a threshold parameter introduced to improve stability. The authors have shown elsewhere [4] that performance is improved by using an adaptive threshold procedure in which θ is adjusted gradually toward zero.

The GDBF algorithm may be implemented as a message-passing algorithm consisting of n symbol node messages and m parity-check messages. In each iteration, these messages are exchanged between corresponding processors that perform the arithmetic update operations.

3 Noise Enhancement in the GDBF Algorithm

Since the GDBF algorithm is a form of gradient descent, it is natural to ask whether the heuristic of stochastic gradient ascent (SGA) can be applied. The SGA procedure introduces random perturbations in each step of the iterative search process. This heuristic is based on the intuition that naive gradient ascent may converge to an erroneous local maximum; the random perturbation is introduced with the intent to avoid or escape from such local maxima. In practice, SGA (or related methods) are known to be effective in various applications [1].

In the context of computing with noisy hardware, we may suppose that the random perturbations appear in the form of noise-induced upsets in the x_i and s_j messages. As a simple experiment, we allow each such message to be inverted with probability ϵ during every iteration. In this experiment we assume that there are no other faults. The results are shown in Fig. 3, which reveals that the algorithm benefits from a non-zero rate of message upsets when operating at high signal-to-noise ratio (SNR). The system's BER is improved when the internal upset rate is increased up to about 10^{-3}. The benefit is lost when the upset rate increases above this level.

4 Discussion

It is not necessarily surprising that the gradient descent decoding method benefits from stochastic perturbations, but this benefit is not guaranteed *a priori*. We may speculate about the particular

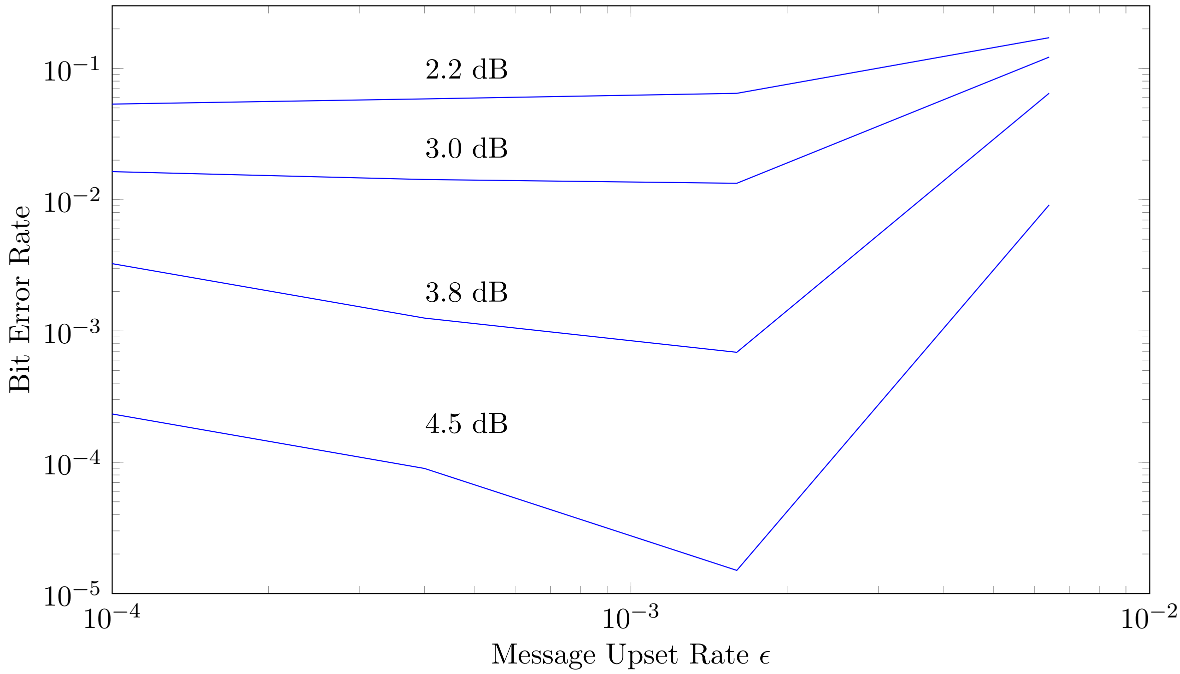

Figure 1: Simulation of the multi-bit GDBF algorithm with random message upsets. These results were obtained for a regular rate-1/2 (3,6) LDPC code of length $n = 1008$. In this simulation, threshold adaptation is assumed. The curves correspond to different values of the channel's signal-to-noise ratio, shown in decibels (dB).

features that allow GDBF to be enhanced by noise. The gradient descent formulation is made possible by adding the syndrome penalty term in (2). This penalty term, and its associated approximation, allow for bits to flipped independently based on the value of a local gradient. By virtue of this fact, the algorithm is able to tolerate independent sign-flips in arbitrary positions, such as may occur due to internal upsets. If a spurious upset occurs, then the algorithm can simply follow the gradient to restore the correct decision. Based on this reasoning, we propose that similar benefits can be realized for other problems if they can be formulated as gradient descent procedures that operate through the independent flipping of single bits.

References

[1] Chakrabartty, S., Shaga, R., and Aono, K. Noise-shaping gradient descent-based online adaptation algorithms for digital calibration of analog circuits. *Neural Networks and Learning Systems, IEEE Transactions on 24*, 4 (2013), 554–565.

[2] MacKay, D. J. C. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003. Available from `http://www.inference.phy.cam.ac.uk/mackay/itila/`.

[3] McDonnell, M. D., and Ward, L. M. The benefits of noise in neural systems: bridging theory and experiment. *Nat Rev Neurosci 12*, 7 (2011), 415–426.

[4] Sundararajan, G., Winstead, C., and Boutillon, E. Noisy gradient descent bit-flip decoding for LDPC codes. arXiv:1402.2773, 2014.

[5] Wadayama, T., Nakamura, K., Yagita, M., Funahashi, Y., Usami, S., and Takumi, I. Gradient descent bit flipping algorithms for decoding LDPC codes. *Communications, IEEE Transactions on 58*, 6 (2010), 1610–1614.