Bad Page Detector for NAND Flash Memory

Yi Liu, **Si Wu**, and **Paul H. Siegel**
Electrical and Computer Engineering Dept., University of California, San Diego, La Jolla, CA 92093 U.S.A
{*yil333, siw070, psiegel*}*@ucsd.edu*

***Abstract*—NAND flash memory cells gradually wear out during the program/erase (P/E) cycling, resulting in an increasing bit error count (BEC) at the page level. The BEC behavior varies significantly among pages. To increase the lifetime of a flash memory device, we propose a *bad page detector*, which predicts whether a page will become a "bad" page in the near future based on its current and previous BEC information. Two machine learning algorithms, based upon time-dependent neural network [1] and long-short term memory [2] architectures, were used to design the detector. Experimental results based on data collected from a TLC flash memory test platform quantify its effectiveness.**

I. Introduction

NAND flash memory cells are able to endure only a limited number of write operations. The program/erase process causes a deterioration of the oxide layer that traps electrons in the cell and eventually makes the cell unreliable. This causes the bit error count (BEC) of a page to increase. If the BEC exceeds a given threshold, for example the correction capability of the error correction code (ECC), some of the data stored in the page is lost, and we say the page is a "bad" page. To address this problem, we develop a *bad page detector*. It predicts the behavior of a page in the near future based on the current and previous BEC information. If a potentially bad page is detected, it can be retired from use and no future data will be written to it. The idea of bad page detection was mentioned in [3], where a detector based on a support-vector machine (SVM) was considered. In this paper, we extend that work and propose two algorithms based on machine learning, including a *time-dependent neural network* (TDNN) and a *long-short term memory* (LSTM).

II. Training Data Set

We measure the performance of TLC flash memory by performing a program and erase (P/E) cycling experiment. The TLC flash memory blocks are P/E cycled up to 10,000 P/E cycles and the BECs are recorded every 100 P/E cycles. The data collected consists of sequences of page BECs, $N(T) = [N(T_1), N(T_2), \dots, N(T_u)]$, associated with P/E cycle counts $[T_1, T_2, \dots, T_u]$.

We now formally define a $(T, T_{\mathrm{Of}}, N_{\mathrm{Th}})$–bad page.

Definition 1. *In a flash memory device, a physical page is called a* (T, T_{Of}, N_{Th})**-bad page** *if its BEC at some P/E cycle count less than* $T + T_{Of}$ *is equal to or greater than* N_{Th}*. Here* T_{Of} *is the P/E cycle count offset, and* N_{Th} *is the BEC threshold.*

For example, the BEC of a $(3000, 500, 1200)$-bad page surpasses 1200 at some P/E cycle count before 3500 . Notice that this page is also a $(T, 500, 1200)$-bad page for any $T \geq 3000$.

Based on this definition, we introduce a labeling function, which takes T, T_{Of}, N_{Th} and a page's BEC sequence $N(T)$ as input. It is defined as

$$L(T, T_{\mathrm{Of}}, N, N_{\mathrm{Th}}) = \begin{cases} 1, & N(T') \geq N_{\mathrm{Th}}, T' \leq T + T_{\mathrm{Of}} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

The label 1 indicates that a page is a $(T, T_{\mathrm{Of}}, N_{\mathrm{Th}})$-bad page.

The detector predicts $(T, 500, N_{\mathrm{Th}})$-bad pages based on BEC information before T. The input of this detector is the BECs vector $[N(T_1), N(T_2), \dots, N(T_5)]$ where $T_i - T_{i-1} = 100$ and $T_5 = T$. The training dataset $\mathcal{X}$ consists of pairs $(\mathbf{x}, t)$, where $\mathbf{x}$ is a length-5 BEC vector,

$$\begin{aligned} &\mathbf{x} = [N(T_1), N(T_2), N(T_3), N(T_4), N(T_5)], \\ &T_{i+1} - T_i = 100 \text{ and } 4000 \leq T_1 < T_5 \leq 10000, \end{aligned} \tag{2}$$

and t is defined by resizing T_5 according to

$$t = \frac{T_5 - 4000}{10000 - 4000} \tag{3}$$

such that t lies in the range $[0, 1]$. The P/E cycling experiment collected data from 12,855 pages, so the dataset size is $12,855 \times 52 = 668,460$ (covering P/E cycle count windows $4000 - 4400$ to $9100 - 9500$). The BEC vector length 5 and P/E cycle count offset 500 were chosen for illustrative purposes.

III. Time-Dependent Neural Network

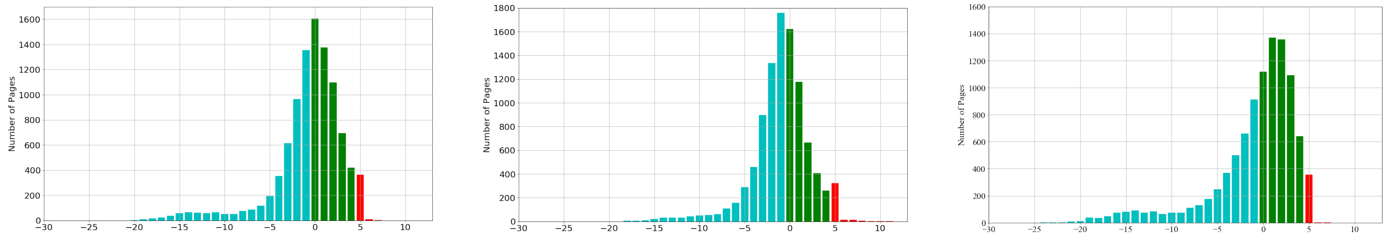
In [3], an SVM detector was trained on a subset of $\mathcal{X}$ with fixed t. In this section, we discuss the structure of a time-dependent neural network (TDNN), whose weights and biases are functions of t. Its motivation comes from the fact that BEC behavior varies as a function of P/E cycle count. The detector consists of three kinds of layers: time-dependent dense layer $\mathcal{D}(\mathbf{x}, t)$ (TDDL), leaky ReLU layer $\mathcal{L}(\mathbf{x})$ and softmax layer $\mathcal{S}(\mathbf{x})$. Given an input vector $\mathbf{x} \in \mathbb{R}^m$, TDDL defines following operation

$$\mathbf{y} = \mathcal{D}(\mathbf{x}, t) = \mathbf{W}(t)\mathbf{x} + \mathbf{B}(t) \tag{4}$$

where $\mathbf{y} \in \mathbb{R}^n$ is the output vector and $\mathbf{W} \in \mathbb{R}^{n \times m}$, $\mathbf{B} \in \mathbb{R}^n$ are network parameters to be determined during the training process. If the parameters are fixed, the layer is a dense layer, denoted simply by $\mathcal{D}(\mathbf{x})$.

The TDNN $\Phi(\mathbf{x}, t)$ contains two TDDLs $\mathcal{D}_0(\mathbf{x}, t)$ and $\mathcal{D}_1(\mathbf{x}, t)$ with input size 5 and output size 5; one dense layer $\mathcal{D}_2(\mathbf{x})$ with input size 5 and output size 2; two leaky ReLU layers $\mathcal{L}(\mathbf{x})$; and one softmax layer $\mathcal{S}(\mathbf{x})$. We build the network $\Phi(\mathbf{x}, t)$ as follows:

$$\mathbf{y} = \Phi(\mathbf{x}, t) = \mathcal{S}(\mathcal{D}_2(\mathcal{L}(\mathcal{D}_1(\mathcal{L}(\mathcal{D}_0(\mathbf{x}, t)), t)))). \tag{5}$$



(a) Time-dependent neural network (b) Long-short term memory network (c) Support vector machine

Fig. 1: Distribution of $Q-$bad page.

Network	Number of mispredicted pages	wasted P/E cycles($\times 100$)
TDNN	375 (2.9%)	15982
LSTM	376 (2.9%)	16084
SVM	358 (2.8%)	19761

TABLE I: Performances of bad page detectors

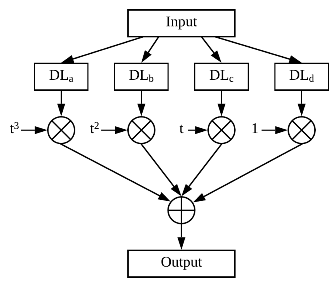


Fig. 2: The structure of time-dependent dense layer. DLs represent dense layers.

Output $\mathbf{y}$ is a length-2 vector $[p_1, p_2]$ that represents the probability of being a bad page ($p_1 + p_2 = 1$). We predict the page is a bad page if, for a specified threshold probability p_{Th} (usually set to 0.5), $p_1 \geq p_{\text{Th}}$. The loss function used in the training process is cross entropy.

We use degree-3 polynomial functions $f(t) = at^3 + bt^2 + ct + d$ to approximate the weight and bias functions in a TDDL, where a, b, c, d are control parameters that are determined during the training process. In general, higher degree polynomials would result in a more accurate approximation. the weight function $\mathbf{W}(t)$ in a TDDL can then be represented as

$$\begin{aligned} \mathbf{W}(t) &= \{w_{ij}(t)\} = \{a_{ij}t^3 + b_{ij}t^2 + c_{ij}t + d_{ij}\} \\ &= \{a_{ij}\}t^3 + \{b_{ij}\}t^2 + \{c_{ij}\}t + \{d_{ij}\} \\ &= \mathbf{W_a}t^3 + \mathbf{W_b}t^2 + \mathbf{W_c}t + \mathbf{W_d}. \end{aligned} \tag{6}$$

A dense layer $\mathcal{D}(\mathbf{x}, t)$ can be converted to the form

$$\mathcal{D}(\mathbf{x}, t) = \mathcal{D}_a(\mathbf{x})t^3 + \mathcal{D}_b(\mathbf{x})t^2 + \mathcal{D}_c(\mathbf{x})t + \mathcal{D}_d(\mathbf{x}). \tag{7}$$

Therefore, a TDDL can be realized by connecting 4 dense layers in parallel, as shown in Fig. 2.

IV. Long-Short Term Memory

The LSTM detector predicts bad pages by solving a regression problem. The network takes a sequence of $l-$consecutive BECs and predicts the the BEC at a future P/E cycle count. The labeling function L_{LSTM} is given by

$$L_{\text{LSTM}}(\mathbf{x}) = N(T_5 + 500) \tag{8}$$

for any $\mathbf{x} \in \mathcal{X}$. The LSTM network consists of a standard LSTM layer [2] $\mathcal{M} : \mathbb{R}^5 \to \mathbb{R}^4$ and a dense layer $\mathcal{D} : \mathbb{R}^4 \to \mathbb{R}$, represented as $y = \mathcal{D}(\mathcal{M}(\mathbf{x}))$. For a specified y_{Th} (usually N_{Th}), if $y \geq y_{\text{Th}}$, we predict the page is a bad page.

V. Experimental Results

Given a page and its BEC vector $N(T)$, we denote by T_l the first time the page is labeled as a bad page, which means $L(T_l, T_{\text{Of}}, N, N_{\text{Th}}) = 1$ and $L(T, T_{\text{Of}}, N, N_{\text{Th}}) = 0$ for any $T < T_l$. Similarly, we denote by T_d the first time the page is classified as a bad page by the detector. Letting $Q = T_d - T_l$, we say the page is a Q-*bad page*. All pages in the dataset can be categorized into three groups based on Q.

- Group I: when $Q < 0$, the page is labeled before it actually becomes a bad page and its storage capacity is wasted.
- Group II: when $0 < Q < 500$, even though the detector fails to label this page correctly at $T_d - 500$, the offset T_{Of} ensures that the detector still labels it bad *before* its BEC actually surpasses N_{Th}.
- Group III: when $Q \geq 500$, the detector fails to predict the bad page before it has become bad.

The goal of the bad page detector is to reduce the number of mispredicted bad pages (Group III) while also controlling the number of wasted P/E cycles (Group I).

Experimental results are shown in Fig. 1. Each bar in the histogram indicates the number of Q-bad pages in the dataset, for $-30 \leq Q \leq 12$. The histogram is divided into three parts. The light blue part on the left represents Group I, the green part represents Group II and the red part represents Group III. Summary results are shown in Table I. To compare the performances, we tuned the detectors such that the number of mispredicted pages is the same, representing about 3% of all pages. Both TDNN and LSTM outperform SVM detector, in the sense that the wasted P/E cycle totals are lower.

References

[1] D. Holden, T. Komura, and J. Saitou, "Phase-functioned neural networks for character control," *ACM Trans. Graphics (TOG)*, volume 36, issue 4, Jul. 2017.

[2] S. Hochreiter and J. Schmidhuber, " Long short-term memory," Neural Computation, 9(8):1735-1780, 1997.

[3] Y. Liu and X. Zhang, "Using machine learning algorithms to map out bad pages in NAND flash memory," presented at 2018 CMRR Spring Research Review, La Jolla, CA, Mar. 15, 2018.