Article

Artificial Neural Network Assisted Error Correction for MLC NAND Flash Memory

Ruiquan He ¹, Haihua Hu ², Chunru Xiong ¹ and Guojun Han ^{1,*}

¹ ZTE School of Information Technology, Xinyu University, Xinyu 338025, China; Ray_HRQ@outlook.com (R.H.); xrcmcu@163.com (C.X.)

² School of Information Engineering, Guangdong University of Technology, Guangzhou 510006, China; haihua@mail2.gdut.edu.cn

* Correspondence: gjhan@gdut.edu.cn

Abstract: The multilevel per cell technology and continued scaling down process technology significantly improves the storage density of NAND flash memory but also brings about a challenge in that data reliability degrades due to the serious noise. To ensure the data reliability, many noise mitigation technologies have been proposed. However, they only mitigate one of the noises of the NAND flash memory channel. In this paper, we consider all the main noises and present a novel neural network-assisted error correction (ANNAEC) scheme to increase the reliability of multi-level cell (MLC) NAND flash memory. To avoid using retention time as an input parameter of the neural network, we propose a relative log-likelihood ratio (LLR) to estimate the actual LLR. Then, we transform the bit detection into a clustering problem and propose to employ a neural network to learn the error characteristics of the NAND flash memory channel. Therefore, the trained neural network has optimized performances of bit error detection. Simulation results show that our proposed scheme can significantly improve the performance of the bit error detection and increase the endurance of NAND flash memory.

Keywords: NAND flash memory; artificial neural network; error correction code; reliability

Citation: He, R.; Hu, H.; Xiong, C.; Han, G. Artificial Neural Network Assisted Error Correction for MLC NAND Flash Memory. *Micromachines* **2021**, *12*, 879. <https://doi.org/10.3390/mi12080879>

Academic Editors: Cristian Zambelli and Rino Micheloni

Received: 30 June 2021

Accepted: 20 July 2021

Published: 27 July 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

NAND flash memories have been widely used in smartphones, personal computers, data centers, etc. Thanks to these two key technologies: (1) continued scaling down process technology and (2) multilevel (e.g., MLC, TLC) cell data coding, the storage density of a NAND flash memory has been significantly increased over previous decades [1]. However, these two key technologies bring about a challenge in that the data stored in NAND flash memory may suffer from low reliability [2–4]. Furthermore, there are two major sources of noise in flash memory: cell-to-cell interference (CCI) and retention noise. Numerous works have been proposed to mitigate noises in NAND flash memory. For example, the data post compensation and predistortion technique [5] and detector design using a neighbor-a-priori information technique [6] exploit the a-priori information of the neighboring cells to mitigate the CCI. However, when considering retention noise, the voltage offset of flash memory cell tends to become unknown. It may be hard to use the a-priori information of the neighboring cells to compensate for the voltage shift caused by CCI. In addition, the CCI removal technique proposed by Lin [7] suffers from a similar problem in that the proposed technique ignores the impact of noise. In addition, Reference [8] proposed a retention-aware belief-propagation (BP) decoding scheme to mitigate the retention noise effect but did not take CCI into consideration.

Against the above background, the recent advances in neural networks and machine learning provide a new perspective to increase the reliability of MLC NAND flash memory. The key idea of the neural network is to learn an optimal network model from the massive training data, instead of using a definitive algorithm that is derived from a pre-defined

model [9]. A pioneering work is reported in [10,11], which utilizes an artificial neural network to predict the threshold voltage distribution of NAND flash memory. In the pretesting, the above method assumes that the prior information of the retention time is informed in advance. When the flash controller is powered off, we cannot obtain the retention time.

In this paper, we use the neural network to learn an optimal network model to detect the bits errors in the cells that are disturbed by both CCI and retention noise and propose a neural network-assisted error correction scheme. However, it is difficult to record the retention time in a practical system, which means that accurate LLR values cannot be calculated. Therefore, we propose using relative LLR to estimate the actual LLR. The relative LLR is affected little by retention time, so we do not require retention time as an input parameter of the neural network.

In this paper, we first model the threshold voltage distribution as a Gaussian mixture model, which is fairly close to the voltage distribution of the practical NAND flash memory, and we calculate the LLR of the theoretical threshold distribution using a quantization scheme. Then, the corresponding LLR of the actual threshold distribution is mapped according to the relative position of the optimal reading reference voltage. It is found that this idea makes the relative LLR values remain relatively steady throughout retention time, which allows us to avoid using retention time as an input parameter of the neural network. Finally, using the relative LLR to estimate the actual LLR, we train the neural network and use the trained network to recovery the bits that may be wrongly detected in the soft-decision detection or hard-decision detection.

The rest of this paper is organized as follows. The flash channel model is presented in Section 2. Section 3 introduces our proposed ANNAEC scheme. Numerical simulation results are presented in Section 4. The conclusions are drawn in Section 5.

2. Channel Model

Without loss of generality, the proposed ANNAEC is performed over a model-based MLC NAND flash memory. Based on [5,8,12], we can model threshold voltage, V_{th} , by

$$V_{th} = V + n_{RTN} + \Delta V_{CCI} - n_{retention}, \quad (1)$$

where V denotes the desired voltage level, n_{RTN} denotes random telegraph noise (RTN), ΔV_{CCI} denotes the shift caused by CCI noise, and $n_{retention}$ denotes retention noise.

2.1. The Voltage Distribution of Programmed and Erased Cell

The number of charges in the NAND flash memory cell can be altered in the program and erase operation. It is well known that before being programmed, a flash memory cell must be erased. In the erase operation, the charges in the memory cell are removed from the floating gate, and the threshold voltage of the erased cell will be set to the lowest voltage. The threshold voltage distribution of an erased cell follows a Gaussian distribution, which is given by

$$p_e(x) = \frac{1}{\sigma_e \sqrt{2\pi}} e^{-\frac{(x-\mu_e)^2}{2\sigma_e^2}} = \mathcal{N}(\mu_e, \sigma_e^2), \quad (2)$$

where σ_e and μ_e are the standard deviation and the mean of the threshold voltage of the erased cell, respectively.

According to [5,8], the threshold voltage of a programmed cell follows a Gaussian distribution shown below:

$$p_p(x) = \frac{1}{\sigma_p \sqrt{2\pi}} e^{-\frac{(x-\mu_p)^2}{2\sigma_p^2}} = \mathcal{N}(\mu_p, \sigma_p^2), \quad (3)$$

where σ_p and $\mu_p \in \{\mu_{p01}, \mu_{p00}, \mu_{p10}\}$ are the standard deviation and the mean of the threshold voltage of a programmed cell.

2.2. RTN

The electron capture and emission at the floating gate near the interface generate RTN, which is greatly impacted by flash memory P/E cycles [13]. As P/E cycles increase, the tunnel oxide of the floating gate transistor is gradually damaged and generates charge trapping in the oxide and interface states. RTN leads to a random fluctuation of cell threshold voltage and widens the voltage distribution. Hence, RTN is modeled with a Gaussian-like distribution [8], given as

$$p_r(x) = \frac{1}{\sigma_r \sqrt{2\pi}} e^{-\frac{x^2}{2\sigma_r^2}} = \mathcal{N}(0, \sigma_r^2), \quad (4)$$

where $\sigma_r = 0.00027 \times PE^{0.62}$, denotes the noise standard deviation.

2.3. CCI

Because of the parasitic capacitance-coupling effect among adjacent cells in flash memory, the threshold voltage of the victim cell increases as the threshold voltage of an adjacent cell increases. The immediate adjacent cells are the major noise source of the CCI. We consider an all bit-line structure. As shown in Figure 1, when the $(k+1)$ -th wordline (WL) has been programmed, the cell on the k -th WL can be programmed. Hence, the victim cell is influenced by three immediate adjacent cells. The threshold-voltage shift of the victim cell can be modeled as a linear combination of the threshold voltage changes of those immediate adjacent cells. We can estimate the threshold-voltage shift caused by CCI as

$$\Delta V_{victim} = \sum_n (\Delta V_t^{(n)} \cdot \gamma^{(n)}), \quad (5)$$

where $\Delta V_t^{(n)}$ is the change of an immediate adjacent cell, which is programmed after the victim cell and $\gamma^{(n)}$ represents the coupling ratio. We assume the vertical and the diagonal coupling ratio are γ_y and γ_{xy} , respectively. According to the cell-to-cell coupling strength factor s , we can set $\gamma_y = 0.08s$ and $\gamma_{xy} = 0.006s$ [12].

Figure 1. Illustration of the parasitic coupling capacitances among adjacent cells.

2.4. Retention

After a cell is programmed, the number of charges in the NAND flash memory cell continually reduce over time due to trap-assisted tunneling and charge detrapping [1]. Retention noise is modeled as a Gaussian distribution, i.e., $p_t(x) = \mathcal{N}(\mu_t, \sigma_t^2) = \frac{1}{\sqrt{2\pi}\sigma_t} e^{-\frac{(x-\mu_t)^2}{2\sigma_t^2}}$. The mean μ_t , and the standard deviation σ_t , are given by

$$\mu_t = \Delta V_t [A_t (PE)^{\alpha_i} + B_t (PE)^{\alpha_o}] \log(1 + T), \quad (6)$$

$$\sigma_t = 0.3 |\mu_t|, \quad (7)$$

where ΔV_t is the cell voltage change before and after being programmed, T donates memory retention time and PE is the number of PE cycles.

The conditional probability distribution function of the threshold voltage after being disturbed by RTN, CCI and retention are given as follows:

$$p(V_{th}|k \in \{11, 01, 00, 01\}) = \frac{1}{64} [\mathcal{N}(\mu_k - \mu_t, \sigma_k^2 + \sigma_t^2 + \sigma_r^2) + A + B + C], \quad (8)$$

$$A = \sum_{\mu_p} [2\mathcal{N}(\gamma_{xy}(\mu_p - \mu_e) + \mu_k - \mu_t, \gamma_{xy}^2(\sigma_p^2 + \sigma_e^2 + 2\sigma_r^2) + \sigma_k^2 + \sigma_t^2) + \mathcal{N}(\gamma_y(\mu_p - \mu_e) + \mu_k - \mu_t, \gamma_y^2(\sigma_p^2 + \sigma_e^2 + 2\sigma_r^2) + \sigma_k^2 + \sigma_t^2)], \quad (9)$$

$$B = \sum_{\mu_p^{(1)}} \sum_{\mu_p^{(2)}} \sum_{\mu_p^{(3)}} \mathcal{N}(\gamma_{xy}(\mu_p^{(1)} + \mu_p^{(2)} - 2\mu_e) + \gamma_y(\mu_p^{(2)} - \mu_e) + \mu_k - \mu_t, (2\gamma_{xy}^2 + \gamma_y^2)(\sigma_p^2 + \sigma_e^2 + 2\sigma_r^2) + \sigma_k^2 + \sigma_t^2), \quad (10)$$

$$C = \sum_{\mu_p^{(1)}} \sum_{\mu_p^{(2)}} \mathcal{N}(\gamma_{xy}(\mu_p^{(1)} - \mu_e) + \gamma_y(\mu_p^{(2)} - \mu_e) + \mu_k - \mu_t, (\gamma_{xy}^2 + \gamma_y^2)(\sigma_p^2 + \sigma_e^2 + 2\sigma_r^2 + \sigma_k^2 + \sigma_t^2)) + \sum_{\mu_p^{(2)}} \sum_{\mu_p^{(3)}} \mathcal{N}(\gamma_{xy}(\mu_p^{(3)} - \mu_e) + \gamma_y(\mu_p^{(2)} - \mu_e) + \mu_k - \mu_t, (\gamma_{xy}^2 + \gamma_y^2)(\sigma_p^2 + \sigma_e^2 + 2\sigma_r^2) + \sigma_k^2 + \sigma_t^2) + \sum_{\mu_p^{(1)}} \sum_{\mu_p^{(3)}} \mathcal{N}(\gamma_{xy}(\mu_p^{(1)} + \mu_p^{(3)} - 2\mu_e) + \mu_k - \mu_t, 2\gamma_{xy}^2(\sigma_p^2 + \sigma_e^2 + 2\sigma_r^2) + \sigma_k^2 + \sigma_t^2), \quad (11)$$

where $\mu_p^{(1)}$, $\mu_p^{(2)}$ and $\mu_p^{(3)}$ are the means of cells 1–3, respectively, which are shown in Figure 2, μ_k and σ_k are the mean and standard deviation of the victim cell.

Figure 2. Illustration of 15-level uniform sensing quantization for multi-level cell (MLC) flash memory.

In this paper, we set the flash memory parameters as follows: $\mu_{p11} = 1.2$, $\mu_{p01} = 2.55$, $\mu_{p00} = 3$, $\mu_{p10} = 3.45$, $\sigma_p = 0.05$, $\sigma_e = 0.35$, $A_t = 0.000035$, $B_t = 0.000235$, $\alpha_i = 0.62$ and $\alpha_o = 0.30$.

3. Artificial Neural Network-Assisted Error Correction

In this section, we first present the idea of relative LLR calculation. Then we explain why an artificial neural network is useful for NAND flash memory. Finally, we introduce our proposed ANNAEC scheme.

3.1. Relative LLR

For soft decision belief-propagation (BP) decoding, a soft quantization scheme has been proposed. As an example, Figure 2 shows a 15-level uniform sensing quantization [12].

The overlap region is obtained by the entropy of the cell's threshold voltage [12,14]. When the threshold voltage falls into the range $(R_{n-1}, R_n]$, where R_n is the n -th reference voltage, $R_0 = -\infty$ and $R_{16} = +\infty$, the LLR values of the least significant bit (LSB) and the most significant bit (MSB) in the i -th cell can be calculated by (12) and (13), respectively:

$$LLR_{lsb}(R_{n-1}, R_n) = \log \frac{\int_{R_{n-1}}^{R_n} p(V_{th}|11) + p(V_{th}|01) dx}{\int_{R_{n-1}}^{R_n} p(V_{th}|00) + p(V_{th}|10) dx}, \quad (12)$$

$$LLR_{msb}(R_{n-1}, R_n) = \log \frac{\int_{R_{n-1}}^{R_n} p(V_{th}|11) + p(V_{th}|10) dx}{\int_{R_{n-1}}^{R_n} p(V_{th}|01) + p(V_{th}|00) dx}. \quad (13)$$

However, it may be hard to accurately calculate the LLR values due to the retention noise. Even though retention noise is modeled as Gaussian distribution, the mean and the standard deviation are random, since ΔV_t is random as described in (6) and (7). Furthermore, it is difficult to obtain accurate retention time in a practical system. To deal with those problems, we can estimate LLR, based on the relative reference voltage positions, given as

$$\begin{aligned} & LLR'_{lsb}(R_{n-1} - V_{rv} + V'_{rv}, R_n - V_{rv} + V'_{rv}) \\ &= \log \frac{\int_{R_{n-1}-V_{rv}+V'_{rv}}^{R_n-V_{rv}+V'_{rv}} p'(V_{th}|11) + p'(V_{th}|01) dx}{\int_{R_{n-1}-V_{rv}+V'_{rv}}^{R_n-V_{rv}+V'_{rv}} p'(V_{th}|00) + p'(V_{th}|10) dx}, \end{aligned} \quad (14)$$

$$\begin{aligned} & LLR'_{msb}(R_{n-1} - V_{rv} + V'_{rv}, R_n - V_{rv} + V'_{rv}) \\ &= \log \frac{\int_{R_{n-1}-V_{rv}+V'_{rv}}^{R_n-V_{rv}+V'_{rv}} p'(V_{th}|11) + p'(V_{th}|10) dx}{\int_{R_{n-1}-V_{rv}+V'_{rv}}^{R_n-V_{rv}+V'_{rv}} p'(V_{th}|01) + p'(V_{th}|00) dx}, \end{aligned} \quad (15)$$

where p' means that we estimate ΔV_t in Equations (6) and (7) as $\Delta V_t \approx \mu_k - \mu_e$, V_{rv} and V'_{rv} are the reference voltages of the actual threshold distribution and the theoretical threshold distribution, respectively, as shown in Figure 3, where V_{rv} is obtained by voltage optimization [1] and V'_{rv} is obtained by theoretical calculations, such as minimizing entropy of the cell's threshold voltage [12,14]. In (14) and (15), we first calculate the LLR of the theoretical threshold distribution using a quantization scheme. Then, the corresponding LLR of the actual threshold distribution is mapped according to the relative position of the optimal reference voltage.

We depict the relative LLR versus data retention time in Figure 4. The relative LLR values remain relatively steady, which allows the neural network to not require retention time as an input parameter. In addition, LLR calculation is offline in a flash memory controller [15]. It may be difficult for a controller to estimate the characteristics of the memory channel because online estimation leads to a significant increase in the power consumption and read latency of the flash controller. Therefore, the proposed relative LLR can estimate the actual LLR over a time range, which can also help reduce the number of LLR tables stored in the controller.

Figure 3. Illustration of the statistic distribution and mathematical distribution at $s = 1$ and $PE = 1K$.

Figure 4. Plot of the relative log-likelihood ratio (LLR) versus data retention time at $PE = 1K$, $\Delta = 0.05$ and $s = 1$.

3.2. Why Are Artificial Neural Networks Useful for NAND Flash Memory?

To simplify the analysis, this subsection first discusses the case that the CCI is only generated by the vertical neighboring cell. In this case, the conditional probability distribution function of the threshold voltage, (8), is simplified to (16):

$$p(V_{th}|k \in \{11, 01, 00, 01\}) = \frac{1}{4} [\mathcal{N}(\mu_k - \mu_t, \sigma_k^2 + \sigma_t^2 + \sigma_r^2) + \sum_{\mu_p} \mathcal{N}(\mu_k + \gamma_y(\mu_p - \mu_e) - \mu_t, \sigma_k^2 + \gamma_y^2(\sigma_p^2 + \sigma_e^2 + 2\sigma_r^2) + \sigma_t^2 + \sigma_r^2)]. \quad (16)$$

In (16), it is seen that the threshold voltage distribution can be divided into four parts: the distribution of cells with CCI from “11”-state, “01”-state, “00”-state and “10”-state, which are also shown in Figure 4. In an overlap region, the bits with different CCI noise levels may have different error rates. For instance, in the overlap region between “01”-state and “00”-state, the bits of the cells in “00”-state with CCI from neighboring cells in “11”-state may be wrongly detected as “1” in LSB. In general, we want to find the optimal reading reference voltage at the intersecting point of the distributions of two states, such

as the red dotted line in Figure 5. However, once we know the programmed state or the threshold voltage of the cells that donate the CCI to victim cells, the optimal reading reference voltage may change. For example, the optimal reading reference voltage should be selected by the blue dotted line in Figure 5, when the vertical neighboring cell is in the erased state.

Figure 5. Illustration of the distribution of NAND flash memory at $s = 1.4$ (the cell-to-cell coupling strength factor), $PE = 1K$ and $Retention\ time = 10^5$.

In this paper, we expand the two-dimensional coordinates to three-dimensional, as shown in Figure 6a. The X-axis is the victim cell's voltage, and the Y-axis is the threshold voltage of vertical neighboring cell. By doing so, one can easily find the incorrectly detected cells, marked with red dots. Moreover, we have two important observations:

1. The correct cells (the blue dots) and the incorrect cells (the red dots) are not interlaced in the three-dimensional space. It means that the correct cells (or the incorrect cells) have similar features, which may be used for clustering them from the incorrect ones.
2. The hard decision may not be the optimal decision when the surrounding cells have been read. In Figure 6a, the gray plane is the hard-decision plane, but not optimal. Suppose that there is a decision plane, shown as Figure 6b, and then we apply this plane to the same data in Figure 6a. One can see that the decision performance by the plane gets significantly improved compared to the plane in Figure 6a.

Figure 6. Illustration of the decision of least significant bit (LSB) in the NAND flash memory. (a) The conventional hard-decision plane in the three-dimensional coordinates. (b) The optimal plane.

These two observations reveal that the detection of bits in a cell can be transformed into a clustering problem, which is to obtain an optimal classification hyperplane. When more surrounding cells are considered, the clustering problem will become more complex and the

dimensions of the classification hyperplane will increase beyond three. To address this issue, We propose to use the neural network, which is good at solving various clustering problems.

3.3. Proposed Artificial Neural Network-Assisted Error Correction (ANNAEC) Scheme

The main idea of the proposed ANNAEC scheme is shown in Figure 7. In general, the flash memory controller uses soft-decision error correction [12], read-retry [1,16] and voltage optimization, which has been widely used in practical systems, to ensure the reliability of data stored in NAND flash memory. When these techniques are not effective in suppressing flash channel noise, the flash memory controller attempts to operate the proposed ANNAEC scheme to correct error bits. Moreover, it can reduce the power consumption and computation burden of the controller, since the cells in an overlap region take a relatively small part of the cells on a page.

Figure 7. Block diagram of the proposed ANNAEC scheme in NAND flash memory.

In general, the host implements data writing and reading to the NAND flash memory chip by communicating with the memory controller, which communicates with the NAND flash memory chip. First, the host transfers data to the flash controller. The flash controller then encodes the data and writes it into the NAND flash memory chip. When the host reads the data, the flash controller communicates with the NAND flash chip. During this process,

the NAND flash chip reads the data from the cell and sends it to the flash controller by reading the sensing circuit. After that, the flash controller corrects and restores the original data through the decoding algorithm and sends it to the host. The proposed a neural network assisted error correction algorithm is used as an alternative decoding algorithm. When the decoding of the flash controller fails, the neural network model is used to first correct the data and then perform decoding.

We label the positions of the cells in an overlap region, which is at the N -th word-line and the M -th bit-line in the block as (N, M) , shown in Figure 7. The input parameters of the neural network are summarized in Table 1. X_1 and X_2 are the bits of cell- (N, M) in MLC memory, respectively. $X_3 \sim X_8$ are the LLRs of LSB and MSB of the immediate adjacent cells, i.e., cell- $(N + 1, M - 1)$, cell- $(N + 1, M)$ and cell- $(N + 1, M + 1)$. X_9 is the flag of page type. If the current reading page is LSB, we set X_9 to “0”; otherwise, X_9 is set to “1”. X_{10} is the number of PE cycles. There are two reasons for choosing those parameters: (1) the threshold voltage is difficult to be obtained in a practical system, but the LLR and bits in a cell can help to locate the range of threshold voltage; (2) the vertical and the diagonal neighboring cells contribute about 81% of the CCI [17,18].

Table 1. Summary of input parameters.

Notation	Physical Meaning
X_1, X_2	bit of the cell (N, M)
X_3, X_4	LLRs of LSB and MSB of the cell- $(N + 1, M - 1)$
X_5, X_6	LLRs of LSB and MSB of the cell- $(N + 1, M)$
X_7, X_8	LLRs of LSB and MSB of the cell- $(N + 1, M + 1)$
X_9	page type (LSB:0; MSB:1)
X_{10}	PE cycle

Afterward, we send the parameters into the back propagation neural network to correct error bits. The sigmoid function is selected as the activation function of the back propagation neural network, given as

$$f(x) = \frac{1}{1 + e^{-x}}. \quad (17)$$

The cost function is chosen as the typical mean square error (MSE) cost function [19], given by

$$E = \frac{1}{2} [(T_{y_0} - y_0)^2 + (T_{y_1} - y_1)^2], \quad (18)$$

where the outputs of neural networks y_0 and y_1 are the reliabilities of “0” and “1”, and T denotes the desired reliability in the data set. The relative LLR is calculated offline in the flash memory controller. It is difficult to recalculate the relative LLR, since the online characteristic estimation of the memory channel causes longer read latency. Since the accurate relative LLR is hard to recalculate, we update relative LLR by

$$LLR_{update} = (-1)^{\varepsilon+1} |LLR_{original}|, \quad (19)$$

where $LLR_{original}$ denotes original relative LLR obtained in the sensing operation, and ε is given by

$$\varepsilon = \begin{cases} 1 & \text{if } y_1 > y_0 \\ 0 & \text{else.} \end{cases} \quad (20)$$

Although (19) does not update the accurate LLR to decode, it can estimate the value of LLR. Moreover, (19) is used to correct the sign of LLR, which is more important than the absolute value of LLR, since fewer error signs of LLRs fewer less error bits.

4. Experiment Results

4.1. Training

Throughout all experiments, we used a rate-0.9 (4544, 4096) QC-LDPC code and the BP decoding algorithm. The experimental platform is implemented in Matlab. The channel parameters, which are used to generate the training dataset, are shown in Table 2. Since the parasitic coupling capacitances of CCI are invariable in a flash memory ship, without loss of generality, we set the cell-to-cell coupling strength factor to be $s = 1$. According to the raw bit error rate (RBER), we generate the dataset at $PE = \{3000, 4000, 5000\}$ and divide the dataset into two parts: error and correct bits, which are to be corrected, e.g., the cell- (N, M) in Figure 7. In total, the sizes of the training and validation data are 336,000 and 84,000, respectively. According to the performance of neural network versus the different numbers of hidden layer node, shown in Figure 8, the basic neural network structure is set to be $\{10, 3, 2\}$, meaning that there are 10 nodes in the input layer, 3 nodes in the hidden layer and 2 nodes in the output layer.

Table 2. Training dataset ($s = 1$).

Retention Time (h) \ PE	3000	4000	5000
RBER			
$\approx 6 \times 10^{-3}$	1×10^5	2×10^4	1×10^4
$\approx 7 \times 10^{-3}$	2×10^5	4×10^4	1.5×10^4
$\approx 8 \times 10^{-3}$	3×10^5	5×10^4	2×10^4
$\approx 9 \times 10^{-3}$	5×10^5	1×10^5	3×10^4
$\approx 1 \times 10^{-2}$	1×10^6	5×10^5	1×10^5
Size of the training data	336,000		
Size of the validation data	84,000		

Figure 8. Performance of neural network under the different numbers of hidden layer nodes.

4.2. Performance

In Figure 9a,b, we compare RBER and frame error rate (FER) using ANN-LDPC [11], the proposed method and the original method without the neural network versus data retention time at $s = 1$. We can observe that the proposed ANNAEC significantly reduces the RBER in comparison with the ANN-LDPC and original method.

For instance, in Figure 9a, the data retention time is about 3×10^4 h at $PE = 5000$ and $RBER = 2 \times 10^{-2}$, using the scheme without ANNAEC. Compared to the proposed ANNAEC scheme, Figure 9b shows that for the same performance, the ANN-LDPC can make the flash memory endure up to 3×10^5 h and the proposed method provides a performance gain of approximately 67% of data retention, which makes the retention time

of flash endure up to 5×10^5 h. In addition, the proposed method has a more stable error correction performance, when the memory suffers from a weak interference. Similarly, we can notice that the proposed ANNAEC improves the FER performance by up to an error rate of 1×10^{-3} at a retention time of 4×10^6 h and $PE = 3000$. The ANN-LDPC has a FER performance of approximately 5×10^{-3} .

Figure 9. (a) Comparison of the raw bit error rate (RBER) performance of NAND flash memory with and without ANNAEC scheme versus data retention time at $s = 1$. (b) Comparison of the frame error rate (FER) performance of low-density parity-check (LDPC) coded NAND flash memory with and without the ANNAEC scheme versus data retention time at $s = 1$.

5. Conclusions

In this paper, we have proposed to use the relative LLR calculation to estimate the actual LLR. Furthermore, in three-dimensional coordinates, we have transformed the bit detection problem into a clustering problem, which allows us to apply an artificial neural network in the memory channel. To solve the clustering problem, we proposed an artificial neural network-assisted error correction scheme, which has been shown by experiments to be effective in correcting the error bit when the conventional method without the neural network fails to decode. Simulation results have shown that the FER performance of our ANNAEC is significantly better than that of ANN-LDPC. For example, the ANN-LDPC can make the flash memory endure up to 3×10^5 h, and the proposed method provides the performance gain of approximately 67% of data retention, which makes the retention time of flash endure up to 5×10^5 h. Furthermore, our proposed approach can be extended to TLC or QLC flash memories.

Author Contributions: Conceptualization, R.H., H.H. and G.H.; methodology, R.H. and G.H.; software, R.H.; validation, R.H. and H.H.; formal analysis, R.H., H.H. and G.H.; investigation, R.H., H.H., G.H. and C.X.; writing—original draft preparation, R.H., G.H. and C.X.; writing—review and editing, H.H. and G.H.; visualization, C.X.; supervision, G.H.; project administration, R.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under grant 61871136.

Data Availability Statement: The study did not report any data.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cai, Y.; Ghose, S.; Haratsch, E.F.; Luo, Y.; Mutlu, O. Error Characterization, Mitigation, and Recovery in Flash-Memory-Based Solid-State Drives. *Proc. IEEE* **2017**, *105*, 1666–1704. [\[CrossRef\]](#) [\[CrossRef\]](#)
2. Lee, J.D.; Choi, J.H.; Park, D.; Kim, K. Data retention characteristics of sub-100 nm NAND flash memory cells. *IEEE Electron Device Lett.* **2003**, *24*, 748–750. [\[CrossRef\]](#)
3. Peng, Z.; He, R.; Han, G.; Cai, G.; Fang, Y. Neighbor-A-Posteriori Information Assisted Cell-State Adaptive Detector for NAND Flash Memory. *IEEE Commun. Lett.* **2019**, *23*, 1967–1971. [\[CrossRef\]](#) [\[CrossRef\]](#)
4. Xiong, Q.; Wu, F.; Lu, Z.; Zhu, Y.; Zhou, Y.; Chu, Y.; Xie, C.; Huang, P. Characterizing 3D Floating Gate NAND Flash. *ACM Trans. Storage* **2018**, *14*, 1–31. [\[CrossRef\]](#) [\[CrossRef\]](#)
5. Dong, G.; Li, S.; Zhang, T. Using Data Postcompensation and Predistortion to Tolerate Cell-to-Cell Interference in MLC nand Flash Memory. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2010**, *57*, 2718–2728. [\[CrossRef\]](#) [\[CrossRef\]](#)
6. Adnan Aslam, C.; Guan, Y.L.; Cai, K. Detector for MLC NAND Flash Memory Using Neighbor A-Priori Information. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2016**, *24*, 2827–2836. [\[CrossRef\]](#) [\[CrossRef\]](#)
7. Lin, X.; Han, G.; Ouyang, S.; Li, Y.; Fang, Y. Low-complexity detection and decoding scheme for LDPC-coded MLC NAND flash memory. *China Commun.* **2018**, *15*, 58–67. [\[CrossRef\]](#) [\[CrossRef\]](#)
8. Aslam, C.A.; Guan, Y.L.; Cai, K. Decision-Directed Retention-Failure Recovery With Channel Update for MLC NAND Flash Memory. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2018**, *65*, 353–365. [\[CrossRef\]](#) [\[CrossRef\]](#)
9. Riaz, H.; Park, J.; Choi, H.; Kim, H.; Kim, J. Deep and Densely Connected Networks for Classification of Diabetic Retinopathy. *Diagnostics* **2020**, *10*, 24. [\[CrossRef\]](#) [\[CrossRef\]](#) [\[PubMed\]](#)
10. Wei, D.; Qiao, L.; Hao, M.; Feng, H.; Peng, X. Reliability prediction model of NAND flash memory based on random forest algorithm. *Microelectron. Reliab.* **2019**, *100*, 100–101. [\[CrossRef\]](#) [\[CrossRef\]](#)
11. Nakamura, T.; Deguchi, Y.; Takeuchi, K. Adaptive Artificial Neural Network-Coupled LDPC ECC as Universal Solution for 3-D and 2-D, Charge-Trap and Floating-Gate NAND Flash Memories. *IEEE J. Solid State Circuits* **2019**, *54*, 745–754. [\[CrossRef\]](#) [\[CrossRef\]](#)
12. Dong, G.; Xie, N.; Zhang, T. On the Use of Soft-Decision Error-Correction Codes in nand Flash Memory. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2011**, *58*, 429–439. [\[CrossRef\]](#) [\[CrossRef\]](#)
13. Compagnoni, C.M.; Ghidotti, M.; Lacaita, A.L.; Spinelli, A.S.; Visconti, A. Random Telegraph Noise Effect on the Programmed Threshold-Voltage Distribution of Flash Memories. *IEEE Electron Device Lett.* **2009**, *30*, 984–986. [\[CrossRef\]](#) [\[CrossRef\]](#)
14. Aslam, C.A.; Guan, Y.L.; Cai, K. Read and Write Voltage Signal Optimization for Multi-Level-Cell (MLC) NAND Flash Memory. *IEEE Trans. Commun.* **2016**, *64*, 1613–1623. [\[CrossRef\]](#) [\[CrossRef\]](#)
15. Sandell, M.; Ismail, A. Machine learning for LLR estimation in flash memory with LDPC codes. *IEEE Trans. Circuits Syst. II Express Briefs* **2021**, *68*, 792–796. [\[CrossRef\]](#) [\[CrossRef\]](#)
16. Yong, K.-K.; Chang, L.-P. Error Diluting: Exploiting 3-D NAND Flash Process Variation for Efficient Read on LDPC-Based SSDs. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **2020**, *39*, 3467–3478. [\[CrossRef\]](#) [\[CrossRef\]](#)
17. Kim, T.; Kong, G.; Weiya, X.; Choi, S. Cell-to-Cell Interference Compensation Schemes Using Reduced Symbol Pattern of Interfering Cells for MLC NAND Flash Memory. *IEEE Trans. Magn.* **2013**, *49*, 2569–2573. [\[CrossRef\]](#) [\[CrossRef\]](#)
18. Park, S.K.; Moon, J. Characterization of Inter-Cell Interference in 3D NAND Flash Memory. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2021**, *68*, 1183–1192. [\[CrossRef\]](#) [\[CrossRef\]](#)
19. Žalik, K.R. An efficient k'-means clustering algorithm. *Pattern Recognit. Lett.* **2008**, *29*, 1385–1391. [\[CrossRef\]](#) [\[CrossRef\]](#)