Reduced Complexity Neural Network Equalizers for Two-Dimensional Magnetic Recording

Ahmed Aboutaleb and Nitin Nangare
Marvell Technology Inc., Santa Clara, CA 95054 USA

This article investigates the reduced complexity neural network (NN)-based architectures for equalization over the two-dimensional magnetic recording (TDMR) digital communication channel for data storage. We use realistic waveforms measured from a hard disk drive (HDD) with TDMR technology. We show that the multilayer perceptron (MLP) nonlinear equalizer achieves a 10.91% reduction in bit error rate (BER) over the linear equalizer with cross-entropy (CE)-based optimization. However, the MLP equalizer’s complexity is 6.6× the linear equalizer’s complexity. Thus, we propose the reduced complexity MLP (RC-MLP) equalizers. Each RC-MLP variant consists of finite-impulse response (FIR) filters, a nonlinear activation, and a hidden delay line. A proposed RC-MLP variant entails only 1.59× the linear equalizer’s complexity while achieving a 8.23% reduction in BER over the linear equalizer.

Index Terms—Equalization, neural network (NN), reduced complexity, two-dimensional magnetic recording (TDMR).

I. INTRODUCTION

HARD disk drives (HDDs) will continue to be cost-competitive, as user bit density increases. Data are stored on HDDs through magnetic recording digital communication channels. The conventional one-dimensional magnetic recording (1DMR) channel stores bits along a single track with sufficient spacing between tracks to prevent intertrack interference (ITI). Two-dimensional magnetic recording (TDMR) increases the density of information by decreasing the track pitch, resulting in significant ITI in the readback waveform [1]. TDMR currently uses two or more closely spaced read heads positioned in the HDD slider to help compensate for the increased ITI. Fig. 1 illustrates a TDMR reader with two heads.

A. Trellis-Based Viterbi Detection for TDMR

The read heads measure the waveforms that are readback from the magnetic media [2]. During readback, such waveforms are passed through a low-pass anti-aliasing filter. To acquire discrete-time samples, an analog-to-digital converter (ADC) samples the filter’s output at an appropriate rate. The intersymbol interference (ISI) in the ADC samples spans many bits in the down-track direction. Most commonly, bits are estimated using a trellis-based detector. With the Viterbi algorithm (VA), the maximum likelihood (ML) detection can be achieved with a trellis-based approach at an acceptable level of complexity [3]. Nevertheless, as the ISI length increases, the number of states in the trellis detector increases exponentially. In addition, the number of computations per bit estimate is directly proportional to the number of trellis states. Because of the ISI/ITI in TDMR, the number of trellis states may be high, which leads to a high level of implementation complexity in trellis-based detection.

Moreover, the canonical VA detector assumes Gaussian noise and linear ISI/ITI in the coded bits in order to achieve optimality in the ML sense. In practice, neither of these assumptions is generally true. In fact, the readback waveforms contain data-dependent noise, partial erasures, nonlinear ISI/ITI, and jitter noise [4], [5]. To shorten the ISI/ITI and whiten the noise, an equalizer is used before the trellis detector. Typical data recovery systems employ a 2-D linear minimum mean square error (2-D LMMSE) equalizer followed by a VA detector [4]. To simplify the implementation of the 2-D LMMSE equalizer, the equalizer is implemented as a finite impulse response (FIR) filter. Accordingly, the VA detector operates on the output samples of the equalizer while assuming a manageable number of trellis states.

B. Suboptimality of the Linear Equalizer

However, the 2-D LMMSE equalizer is optimal in the mean-square error (MSE) only if the noise is Gaussian and...
stationary and if the channel is linear in terms of the written bits [6]. Furthermore, an equalizer trained to minimize the MSE may not result in the best detector bit error rate (BER), which is the desired figure of merit. Furthermore, as the storage channel area density increases, nonlinear impairments, residual ISI and ITI, and signal-dependent noise become more significant [4]. Therefore, the actual channel conditions differ substantially from those required for optimality. In short, the 2-D LMMSE equalizer is not optimal in the desired metric (BER), and its MSE optimality conditions are not met in practice.

C. Relevant Works on Neural Networks for Magnetic Recording

Neural networks (NNs) are capable of compensating for various types of impairments, so long as they are trained with appropriate representative data. Recent studies have demonstrated that NNs can provide improved performance in high-density magnetic recording channels, despite the presence of higher amounts of media noise than in conventional 1-D channels. The use of NNs significantly improves the overall system performance of 1DMR [4], TDMR [5, 7, 8, 9], and multilayer magnetic recording (MLMR) [10] systems.

Nair and Moon [4] propose using a multilayer perceptron (MLP) as an equalizer for high-density 1DMR channels. The MLP is a fully connected feedforward NN with one or more hidden layers (see [11]). Their results demonstrate that, as a nonlinear equalizer, the MLP outperforms the conventional linear equalizer in terms of MSE and BER. Sayyafan et al. [8] have proposed to integrate a convolutional NN (CNN) with Bahl–Cocke–Jelinek–Raviv (BCJR) in order to iteratively estimate and cancel the data-dependent media noise. In both 1DMR and TDMR channels, their proposed system achieves significant information density gains over conventional pattern-dependent noise prediction (PDNP) systems. An investigation conducted by Luo et al. [9] demonstrates that the NN equalizer outperforms the 2-D LMMSE equalizer in a TDMR channel. In addition, Shen et al. [7] have presented a 2-D LMMSE equalizer followed by a CNN detector for application in TDMR. This system outperforms a conventional system comprised of a 2-D LMMSE equalizer along with a 2-D BCJR and a 2-D PDNP. Aboutaleb et al. [10] have proposed CNN-based equalization and detection systems for MLMR channels suffering from severe data-dependent noise. Compared with the conventional 2-D LMMSE equalizer-VA systems, the CNN-based systems achieve significant density gains. Shen and Nangare [5] have proposed an NN equalizer followed by a VA in which the parameters of the NN adapt to minimize the cross entropy (CE). By adapting the equalizer’s parameters to minimize the CE loss, the authors demonstrate lower detector BERs compared with the MSE-adapted equalizer parameters. Their study suggests that CE-based optimization correlates better with improved BERs, compared with MSE-based optimization. Furthermore, their study confirms that the NN equalizers can be used to compensate for the various impairments associated with high-density magnetic recording systems.

D. High Complexity of Neural Networks for Equalization and Detection

Despite the improvements in performance reported in [7], [8], [9], and [10], the mentioned NN-based equalizers require a much higher implementation complexity than the linear equalizer baseline. Indeed, the high complexity of NN-based methods hinders practical implementation. For example, among the lowest complexity NN equalizers proposed by previous studies, the complexity of the MLP in [5] is about 6.6× the complexity of the 2-D LMMSE. Moreover, recent studies on NN-based equalizers have not proposed any architectures with complexities that are 3× or lower than the complexity of the baseline linear equalizer.

E. Overview of the Proposed Equalizers

We propose four variants of a reduced complexity MLP (RC-MLP) architecture to facilitate practical implementation. The RC-MLP contains FIR filters, hidden delay lines, and nonlinear activation functions; these components can easily be implemented in practice. We demonstrate that the RC-MLP architectures deliver an excellent balance between performance and complexity. Among the proposed variants, one architecture offers the best performance-complexity trade-off.

The novel contributions of this article are summarized as follows.

1) We investigate reduced complexity NN architectures for equalization over TDMR channels. We propose four variants of RC-MLP that achieve most of the performance gains of high-complexity and high-performance MLPs.

2) In addition, we consider candidate equalizer architectures based on radial basis function NNs (RBFNNs). Our study examines the performance and complexity of these architectures.

3) Our experiments use data and readback waveforms measured from an HDD to evaluate the performance of the proposed and baseline methods. The data are obtained from an HDD with TDMR technology. Performance and complexity are compared for each method. Then, we identify the architecture with the best balance between performance and complexity.

II. SYSTEM MODEL

This article uses actual HDD waveforms with TDMR technology using two read heads for training and testing. The raw unequalized waveforms are sampled synchronously at one sample per bit per read head. In the next paragraph, we summarize a channel model that approximates the TDMR channel, where details can be found in [5, Sec. II] and [12].

A. Summary of a Channel Model

Let \( u = u_n \in \{-1, +1\} \) denote a binary input sequence to be written on the media. Define the transition sequence \( b \), such that \( b_n = (u_n - u_{n-1})/2 \). Let \( h(t, w) \) denote the 2-D channel response, modeled using the \( \text{erf}(\cdot) \) function as in [5],...
Then, the continuous-time readback waveform $r_n(t)$ is given by

$$r_n(t) = \sum b_n h(t - nT + \Delta t_n, w + \Delta w_n) + n(t)$$

(1)

where $T$ is the symbol interval, $\Delta t_n$ is the down-track jitter noise, modeled as a Gaussian random process using a truncated Gaussian distribution, such that $|\Delta t_n| < T/2$, $\Delta w_n$ is the cross-track jitter noise modeled as a truncated Gaussian distribution, and $n(t)$ is the additive white Gaussian noise (AWGN), which models the reader electronics noise.

Let $p(t, w) \triangleq h(t, w) - h(t - T, w)$ denote the channel bit response. Then, the equivalent discrete-time channel model is given by [5]

$$r_n^{(l)} = (p^{(l)} \ast u)_n + n^{(l)}$$

(2)

where $\ast$ denotes the discrete-time convolution, and $p^{(l)}$ is the 2-D response for reader $l$. It is worth noting that more accurate datasets can be synthesized using the grain flipping probability model to produce the magnetized bit cells as $u$ by simulating media–data interactions [14], [15]. This approach has been demonstrated to generated readback waveforms that closely match with readback measurements from HDDs. The equalizer’s parameters can be adapted using the CE or MSE loss.

### B. General Equalizer-Trellis Detector System

Fig. 2 shows the equalizer-detector system. The equalizer processes the ADC samples from the readers. We assume one reader per track and that the outputs of two readers are processed simultaneously. The equalizer’s output is passed to the soft output VA (SOVA) detector [16], which computes soft bit estimates as log-likelihood ratio (LLRs). The LLRs are then fed to the channel decoder to recover the information bits. The equalizer’s trainable parameters and target are adjusted using the MSE or the CE criteria.

### III. Adaptation

Let the equalizer’s output be denoted by $y_n$ and the noise-free partial response (PR) target signal by $\hat{y}_n$. The equalizer output $y_n$ depends on the equalizer’s architecture design and its learnable parameters, which include the weights $W$ and biases $B$. The equalizer’s design is discussed in Sections IV and V. The noise-free PR signal is given by

$$\hat{y}_n = (g \ast u)_n$$

(3)

where $g$ represents the PR target, and $u$ represents the written bit sequence.

#### A. Mean-Squared Error Loss Function

For a length-$N$ minibatch, the average MSE $J_{\text{MSE}}$ is computed as follows:

$$J_{\text{MSE}} = \frac{1}{N} \sum_{n=0}^{N-1} (\hat{y}_n - y_n)^2.$$  

(4)

For MSE adaptation, the optimization problem is given by

$$\min_{W, B, g} J_{\text{MSE}}$$  

(5a)

subject to $a^T g = 1$  

(5b)

where (5b) is the monic constraint (MC) on the target, which sets the first tap of the target to one, i.e., $g = [1, 0, 0, \ldots, 0]$. In MSE adaptation, including the MC prevents the target coefficients from converging to the trivial solution of $g = 0$ and has been shown to improve the BER performance [17].

#### B. Cross-Entropy Loss Function

Consider estimating the $n$th bit $u_n$. This estimate is denoted by $\hat{u}_n$. For CE adaptation, $\hat{u}_n$ is obtained by passing the equalizer’s output to the VA and computing the soft decisions, in the form of LLRs, using the SOVA [16]. Let $1_{u_n=i}$ denote the indicator function, such that $1_{u_n=i} = 1$ if $u_n = i$, $i \in \{-1, 1\}$ (and zero otherwise), and define $p_{0,n} \triangleq \Pr[\hat{u}_n = -1]$. Then, for the $n$th bit, the CE loss is computed as follows:

$$\mathcal{H}(u_n, \hat{u}_n) = -1_{u_n=-1} \log(p_{0,n}) - 1_{u_n=1} \log(1 - p_{0,n}).$$  

(6)

The estimate LLR$_n$ for the $n$th bit provided by the SOVA is defined as follows [5], [16]:

$$\text{LLR}_n \triangleq \log \left( \frac{\Pr[\hat{u}_n = 1]}{\Pr[\hat{u}_n = -1]} \right)$$  

(7)

$$= \log \left( \frac{1 - p_{0,n}}{p_{0,n}} \right).$$  

(8)

Then, $p_{0,n}$ is computed as follows:

$$p_{0,n} = \frac{1}{1 + e^{\text{LLR}_n}}.$$  

(9)

By substituting (9) in (6), it can be shown that the CE is given by

$$\mathcal{H}(u_n, \hat{u}_n) = \log \left( 1 + e^{-\text{LLR}_n} \right).$$  

(10)

The objective function to minimize is the average CE loss computed over a minibatch of length $N$ as

$$J_{\text{CE}} = \frac{1}{N} \sum_{k=0}^{N-1} \mathcal{H}(u_n, \hat{u}_n).$$  

(11)

Hence, the optimization problem can be written as follows:

$$\min_{W, B, g} J_{\text{CE}}.$$  

(12)

The optimization problems in (5a) and (12) are solved using the backpropagation algorithm with stochastic gradient descent (SGD) [11, Ch. 8].
Adapting the equalizer’s parameters using the CE criterion is equivalent to ML adaptation, since the BER and soft-bit information are jointly improved. Thus, for any given equalizer structure, the CE adaptation outperforms MSE adaptation in terms of BER [5].

IV. NEURAL NETWORK EQUALIZERS

The MLP and RBFNN are universal approximators under mild conditions [18], [19]. As such, they are capable of approximating any continuous input–output mapping provided they have sufficient hidden nodes and appropriate data to train them. The universality of the MLP and RBFNN makes them ideal candidates for mitigating the impairments and data-dependent noise observed in high-density magnetic recording.

A. Multilayer Perceptron

Fig. 3 shows the architecture of the MLP equalizer. To estimate the nth bit in the downtrack direction, the MLP uses the ADC samples observed over a sliding window of size 2M + 1 for each reader centered around the nth readback sample. Let \( r_n = [r_{n,0}, r_{n,1}] \) denote such ADC samples from the TDMR reader centered around the nth sample, where the ADC samples per reader are defined as follows: \( r_{n,0} = [r_{n-0,M}, r_{n-M+0}, \ldots, r_{n-M+1}, r_{n+M-1}, r_{n+1}, \ldots, r_{n+M}]^T, \) \( l = 0, 1 \). The MLP contains a matrix of weights \( W = \{w_{ij}\} \in \mathbb{R}^{(4M+2) \times K} \) that multiplies the vectorized input samples \( r_{n,\text{vec}} = \text{vec}(r_n) \) to provide K intermediate outputs. After adding the biases \( b_{0,k}, k = 1, \ldots, K \), to the intermediate outputs, the hidden outputs are obtained by applying an elementwise nonlinear activation function \( \Psi(\cdot) \) to the outcome. The examples of \( \Psi(\cdot) \) include the hyperbolic tangent, the sigmoid, and the rectified linear units (ReLU) activations. The MLP’s output is an affine combination of the K hidden outputs, where \( v = [v_j] \in \mathbb{R}^K \) is the vector of coefficients used in the linear combination, and \( b_1 \) is the added bias term. More precisely, the output of the MLP is given by

\[
y_n = \sum_{j=0}^{K-1} v_j \Psi \left( \sum_{i=0}^{4M+1} w_{ij,\text{vec}} r_{n,\text{vec},i} + b_{0,j} \right) + b_1.
\]

The learnable parameters for the MLP are the weights \( W = \{W, v\} \), and biases \( B = \{b_{0,k}, b_1\} \). Hence, the total number of parameters for the MLP is \( |W| + |B| = 4MK + 4K + 1 \). From (13), for each output sample, the MLP performs \( 4MK + 3K \) multiplications and additions, and \( K \) evaluations of \( \Psi(\cdot) \).

B. Radial Basis Function Neural Network

Fig. 4 shows the architecture of the RBFNN. The RBFNN replaces the matrix multiplication and nonlinear activation in the MLP with distances from cluster centroids and a basis function, respectively. The RBFNN is characterized by the centroids \( c_k \) with K biases \( b_{0,k}, k = 1, \ldots, K \), a length-K vector \( v \) with bias \( b_2 \), a norm \( \| \cdot \| \), typically the Euclidean norm, and a basis function \( \phi(\cdot) \). The length of the centroids \( c_k \) matches the number of the input samples. The canonical implementation of the RBFNN uses the Euclidean distance. Given an input instance, the distance between the input samples vectors and the centroids is computed. Then, the basis function is applied to the distances to provide the hidden output. Finally, the RBFNN output is computed as an affine combination of the hidden outputs. More specifically, the RBFNN output is described by

\[
y_n = \sum_{k=0}^{K-1} v_k \phi \left( \| r_{n,\text{vec}} - c_k \| + b_{0,k} \right) + b_1
\]

where \( c_k \in \mathbb{R}^{4M+2} \). To reduce the dimension of the centroids, we can use centroids that have the same length as the number of ADC samples per reader, instead of the vectorized ADC samples for both readers. If \( K \) is even, then this formulation gives the output of the RBFNN as

\[
y_n = \sum_{k=0}^{K/2-1} v_k \phi \left( \| r_{n,\text{vec}} - c'_k \| + b_{0,k} \right) + b_1
\]

where \( c'_k \in \mathbb{R}^{2M+1} \). In (15), \( K/2 \) centroids are assigned for each ADC sequence. The learnable parameters consist of the centers \( C = \{c_k\} \), weights vector \( v \), and biases \( B = \{b_{0,k}, b_1\} \). Thus, the RBFNN uses \( |C| + |V| + |B| = 2KM + 3K + 2 \) learnable parameters.
V. PROPOSED REDUCED COMPLEXITY NEURAL NETWORK-BASED EQUALIZERS

Many applications of NNs for computer vision utilize convolutional layers instead of fully connected matrix layers [11]. In addition, CNNs have also been shown to be universal approximators [20].

Nair and Moon [4], [21] observed that the rows of the trained matrix of weights in the MLP equalizer are approximately delayed replicas of one another for high-density TDMR. Therefore, the authors have proposed replacing the large matrix of weights in the MLP equalizer with an FIR filter.

A. FIR-RBFNN

To decrease the complexity of the RBFNN, we introduce an FIR filter $f_l \in \mathbb{R}^{2M+1}$, $l = 0, 1$, per ADC sequence, as shown in Fig. 5. The FIR filter maps the length-$(2M+1)$ ADC input to a length-$(2M' + 1)$ sequence, where $M' < M$. The FIR-RBFNN output is given by

$$y_n = \sum_{i=0}^{K} \sum_{k=0}^{K/2-1} d_k \phi\left(\|f_l(n) \ast r_{q(n)}\| + c_k\right) + b_{i,l} + b_l$$

(16)

where the dimension of the centroids is now $2M' + 1$.

B. RC-MLP

We propose four variants of RC-MLP architectures. RC-MLP employs FIR filters to implement matrix multiplication and affine combination operations in MLP, which results in the considerably lower complexity architectures. The input layer introduces two FIR filters, FIR-1 and FIR-2, one for each stream of the ADC samples. According to the specific architecture, the hidden outputs corresponding to each filter are then stacked in one or two delay lines, as exhibited in Figs. 6 and 7. An FIR filter can be viewed as a special case of a convolutional filter that performs only discrete-time convolutions.

1) RC-MLP1: Fig. 6 shows the architecture of RC-MLP1. Two intermediary delay lines are introduced to store temporary hidden outcomes from FIR-1 and FIR-2 separately. The last layer consists of two FIR filters, FIR-3 and FIR-4, that map the delayed hidden samples to the final output.

Let $f_l$ and $q_l$, $l = 0, 1$ denote the FIR filters interfacing the input ADC samples and hidden delayed outputs, respectively, shown in Fig. 7. The lengths of $f_l$ and $q_l$ are $2M + 1$ and $K$, respectively, where $K$ is the number of hidden delay samples per ADC path. Then, the equalizer output $y_n$ is given in terms of the hidden outputs $h_n$ as

$$h_n = \sum_{l=0}^{1} (q_l(n) \ast h_n) + b_0$$

(17)

where $b_0$ and $b_1$ are bias terms. RC-MLP1 uses $4M + 2K + 7$ learnable parameters and requires $4M + 2K + 2$ multipliers and two evaluations of $\psi(\cdot)$ per bit estimate.

2) RC-MLP2: RC-MLP2 sums the outputs of FIR-1 and FIR-2. A nonlinear activation function is then applied to the sum. Consequently, only one hidden delay line with $K$ samples is required, as shown in Fig. 8. FIR-3 combines the hidden delay line samples to provide the equalizer output. The hidden outputs and equalizer output can be expressed as follows:

$$y_n = \sum_{l=0}^{1} (f_l(n) \ast r_{q(n)}) + b_0$$

(18)

where $b_0$ and $b_1$ are bias terms. RC-MLP2 requires $4M + K + 4$ parameters, $4M + K$ multipliers, and one evaluation of $\psi(\cdot)$ per bit estimate.
3) RC-MLP3: The RC-MLP3 structure adds a linear connection between the linear output and the final output, as can be seen in Fig. 9. We added the linear connection in order to jump-start the system, using the FIR-1 and FIR-2 combination as a linear equalizer, while maintaining the improved generalization provided by the nonlinear NN equalizer. As a result of the direct connection between the first hidden output and the final output, errors are more directly propagated from the output to the adapted FIR-1 and FIR-2. The equalizer’s output for RC-MLP3 can be written as follows:

\[
h_{n, \text{Linear}} = \sum_{l=0}^{1} (f^{(l)} * r^{(l)})_n + b_0
\]

\[
h_{n} = \psi(h_{n, \text{Linear}})
\]

\[
y_n = (q * h_n)_n + c h_{n, \text{Linear}} + b_1.
\]

(19)

4) RC-MLP4: RC-MLP4 adds a linear connection to RC-MLP1, as shown in Fig. 10. The output is given by

\[
h_{n, \text{Linear}}^{(l)} = (f^{(l)} * r^{(l)})_n + b_{0,l}, \quad l = 0, 1
\]

\[
h_{n}^{(l)} = \psi(h_{n, \text{Linear}}^{(l)})
\]

\[
y_n = \left(\sum_{l=0}^{1} q^{(l)} * h_{n}^{(l)}\right)_n + c \sum_{l=0}^{1} h_{n, \text{Linear}}^{(l)} + b_1.
\]

(20)

VI. SIMULATION RESULTS

The performance and complexity of the discussed architectures have been assessed on raw ADC samples obtained from an HDD with TDMR technology. The data consists of 520 sectors, where each sector contains around 40,000 bits. The two read heads provide one ADC sample per bit, totaling two ADC samples per bit. The reader cross-track separation (CTS) is 52% of the 85-nm track pitch.

A. Linear Equalizers—20 Sectors

A comparison of BER performances and complexity across the first 20 sectors is provided in Table I. The equalizers’ trainable parameters and target adapt to the CE criterion, except for the 2-D LMMSE. For linear equalizers, 11 taps per ADC sequence are used, unless otherwise noted. The computational complexity of a method is quantified by the number of learnable parameters it requires. The number of parameters is directly proportional to the number of elementary operations required per bit estimation. Target taps \(g_i\) are optimized under an MC in all experiments. A special case is an experiment with an \([3, 7, 1]\) that uses 2-D LMMSE. In practice, such a configuration is used as a baseline. By adapting the target, the BER is reduced by 8.70% when the MSE is used as the criterion for adaptation. Furthermore, we consider a linear equalizer that is trained on CE (2-D LECE). Further reduction of the BER occurs when the linear equalizer is adapted with CE instead of MSE. By increasing the number of FIR taps (from 11 taps to 21 taps per ADC sequence), the linear equalizer’s BER decreases by about 1.77%. Compared with the linear equalizer that has 11 taps per ADC, 21 taps per ADC increases the linear equalizer’s complexity by about 90%.

B. RBFNN-Based Equalizers—20 Sectors

The conventional RBFNN equalizer increases complexity without significant improvement in the BER even with 20 and 30 cluster centroids per ADC. To improve the performance, the FIR-RBFNN is used. The two FIR filters, one per ADC sequence, interface the ADC samples and map the 11 samples per ADC to five samples per ADC. In this case, the centroids lie in 5-D space (instead of 11-D), making the distance computation more efficient. This FIR-RBFNN architecture with 6 centroids per ADC achieves BER performance that is only 1.67% higher than the RBFNN with 20 centroids, while requiring 4.87 less parameters. Nonetheless, the FIR-RBFNN’s BER is 5.23% lower than 2-D LECE, but its number of
equalizers are trained using CE. The MLP equalizer achieves improvements in BER by 9\% and target coefficients based on CE rather than MSE further by 6\% 

average over 520 sectors. In the case of MSE training of the linear, RC-MLP, and MLP equalizers when observed as an

D. Performance Versus Complexity on All 520 Sectors

Fig. 11 summarizes the performance versus complexity for the linear, RC-MLP, and MLP equalizers when observed as an average over 520 sectors. In the case of MSE training of the linear equalizer, adapting the target results in an improvement by 6.6\% in BER. Adapting the linear equalizer coefficients and target coefficients based on CE rather than MSE further improves the performance by 9.26\%. The RC-MLP and MLP equalizers are trained using CE. The MLP equalizer achieves

a 10.91\% lower BER than the linear equalizer trained on CE, at the cost of increasing complexity by 6.6×. In comparison, RC-MLP3 achieves a gain of 8.23\% over the linear equalizer trained on CE, at the cost of increasing complexity by 1.59×.

E. Mutual Information

It is instructive to examine whether the BER gain between the linear equalizer and RC-MLP3 translates into an increase in the mutual information between the LLRs and the written bits \( \mathbf{u} \), which is denoted by \( I(\text{LLR}, \mathbf{u}) \). The mutual information is computed using the nominal algorithm in [22] and [23].

Table II summarizes the mutual information for both equalizers. We observe a significantly lower BER with RC-MLP3 over 2-D LECE. This improvement in BER also translates into an increase in the mutual information \( I(\text{LLR}, \mathbf{u}) \). For CE adaptation, adapting the parameters of the equalizers improves both the sign and magnitude of the LLRs, because CE adaptation is an ML adaptation [5]. Consequently, we observe that both the BER and the mutual information have improved.

VII. Conclusion

We have examined the performance-complexity trade-off for different equalizer architectures. The MLP achieves significant performance gains over the linear equalizer. However, its complexity is about 6.6× that of the linear equalizer. We have proposed four variants of the RC-MLP to facilitate practical implementation of nonlinear NN equalizer. The RC-MLP3 architecture outperforms its variants and achieves most of the performance gains of the MLP while requiring only 1.59× the complexity of the linear equalizer. The complexities of MLP and RC-MLP scale as \( O(LK+K) \) and \( O(L+K) \), respectively, if \( L \) is the length of the input, and \( K \) is the length of the hidden output. In the future work, we will examine codeword error rates for different architectures.

Acknowledgment

Nitin Nangare is a Core Technical Contributor.

Table II

| Architecture | BER     | \( I(\text{LLR}, \mathbf{u}) \) |
|--------------|---------|-------------------------------|
| 2D-LMMSE     | 0.025436| 0.89798                       |
| 2D-LECE      | 0.023020| 0.90717                       |
| RC-MLP3      | 0.021196| 0.91436                       |

References

[1] R. Wood, M. Williams, A. Kavcic, and J. Miles, “The feasibility of magnetic recording at 10 Terabits per square inch on conventional media,” IEEE Trans. Magn., vol. 45, no. 2, pp. 917–923, Feb. 2009.

[2] B. Vasic and E. M. Kurtas, Coding and Signal Processing for Magnetic Recording Systems. Boca Raton, FL, USA: CRC Press, 2004.

[3] G. D. Forney, Jr., “The Viterbi algorithm,” Proc. IEEE, vol. 61, no. 3, pp. 268–278, Mar. 1973.

[4] S. K. Nair and J. Moon, “Data storage channel equalization using neural networks,” IEEE Trans. Neural Netw., vol. 8, no. 5, pp. 1037–1048, Sep. 1997.
[5] J. Shen and N. Nangare, “Nonlinear equalization for TDMR channels using neural networks,” in Proc. 54th Annu. Conf. Inf. Sci. Syst. (CISS), 2020, pp. 1–6.

[6] J. M. Mendel, Lessons in Estimation Theory for Signal Processing, Communications, and Control. London, U.K.: Pearson, 1995.

[7] J. Shen, A. Aboutaleb, K. Sivakumar, B. J. Belzer, K. S. Chan, and A. James, “Deep neural network a posteriori probability detector for two-dimensional magnetic recording,” IEEE Trans. Magn., vol. 56, no. 6, pp. 1–12, Jun. 2020.

[8] A. Sayyafan et al., “Deep neural network media noise predictor turbo-detection system for 1-D and 2-D high-density magnetic recording,” IEEE Trans. Magn., vol. 57, no. 3, pp. 1–13, Mar. 2021.

[9] K. Luo et al., “A study on block-based neural network equalization in TDMR system with LDPC coding,” IEEE Trans. Magn., vol. 55, no. 11, pp. 1–5, Nov. 2019.

[10] A. Aboutaleb et al., “Deep neural network-based detection and partial response equalization for multilayer magnetic recording,” IEEE Trans. Magn., vol. 57, no. 3, Mar. 2020, Art. no. 3101012.

[11] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, Deep Learning, vol. 1. Cambridge, MA, USA: MIT Press, 2016.

[12] R. Pighi, R. Raheli, and U. Amadei, “Multidimensional signal processing and detection for storage systems with data-dependent transition noise,” IEEE Trans. Magn., vol. 42, no. 7, pp. 1905–1916, Jul. 2006.

[13] P. Kovintavewat, I. Ozgunes, E. Kurtas, J. R. Barry, and S. W. McLaughlin, “Generalized partial-response targets for perpendicular recording with jitter noise,” IEEE Trans. Magn., vol. 38, no. 5, pp. 2340–2342, Sep. 2002.

[14] K. S. Chan et al., “Channel models and detectors for two-dimensional magnetic recording,” IEEE Trans. Magn., vol. 46, no. 3, pp. 804–811, Mar. 2010.

[15] S. J. Greaves, K. S. Chan, and Y. Kanai, “Areal density capability of dual-structure media for microwave-assisted magnetic recording,” IEEE Trans. Magn., vol. 55, no. 12, pp. 1–9, Dec. 2019.

[16] J. Hagenauer and P. Hoeher, “A Viterbi algorithm with soft-decision outputs and its applications,” in Proc. IEEE Global Telecommun. Conf. Exhib. Commun. Technol. Beyond, vol. 3, Nov. 1989, pp. 1680–1686.

[17] J. Moon and W. Zeng, “Equalization for maximum likelihood detectors,” IEEE Trans. Magn., vol. 31, no. 2, pp. 1083–1088, Mar. 1995.

[18] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” Neural Netw., vol. 2, no. 5, pp. 359–366, Jan. 1989.

[19] J. Park and I. W. Sandberg, “Universal approximation using radial-basis-function networks,” Neural Comput., vol. 3, no. 2, pp. 246–257, Mar. 1991.

[20] D.-X. Zhou, “Universality of deep convolutional neural networks,” Appl. Comput. Harmon. Anal., vol. 48, no. 2, pp. 787–794, 2020.

[21] S. K. Nair and J. Moon, “Simplified nonlinear equalizers,” IEEE Trans. Magn., vol. 31, no. 6, pp. 3051–3053, Nov. 1995.

[22] A. Kraskov, H. Stögbauer, and P. Grassberger, “Estimating mutual information,” Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top., vol. 69, Jun. 2004, Art. no. 066138.

[23] B. C. Ross, “Mutual information between discrete and continuous data sets,” PLoS ONE, vol. 9, no. 2, 2014, Art. no. e87357.