Convolutional Neural Network Based Symbol Detector for Two-Dimensional Magnetic Recording

Jinlu SHEN[1], Benjamin J. BELZER[1], Krishnamoorthy SIVAKUMAR[1], Kheong Sann CHAN[2] and Ashish JAMES[3]

1) Washington State University, Pullman, WA, USA, (jinlu.shen, belzer, siva)@wsu.edu

2) Nanjing Institute of Technology, Nanjing, China, kheongsann@ieee.org

3) Institute for Infocomm Research (I2R), A*STAR, Singapore, ashish_james@i2r.a-star.edu.sg

I. INTRODUCTION

Conventional detection systems in hard disk drives (HDD) typically include a 2D partial response (PR) equalizer that pre-processes the readback signals and shapes the output to a controlled target response, followed by a maximum likelihood (ML) or maximum *a posteriori* (MAP) detector which outputs log-likelihood ratios (LLRs) to be passed to a channel decoder. Pattern dependent noise prediction (PDNP) algorithm [1] is usually incorporated into the metric computation of the trellis in the ML/MAP detector to combat media noise intrinsic to the magnetic recording (MR) channel. For next generation two-dimensional magnetic recording (TDMR) HDDs, such conventional systems would suffer from impractically large trellis state cardinality when performing multi-track detection, and they may no longer be capable of handling the increased nonlinearities in high density recording channels. This work investigates applying advanced machine learning techniques to TDMR. Convolutional neural networks (ConvNets) are employed in place of the PR equalizer and ML/MAP detector with PDNP to directly process the un-equalized readback signals and output soft estimates. ConvNets are special deep neural networks (DNNs) that assume the inputs are images and perform convolution instead of affine function in the network forward pass [2]. This enables far fewer parameters in ConvNets than regular DNNs of the same depth and therefore allows for deeper networks. The motivation to use ConvNets is the resemblance between data detection problem in MR and typical image processing problems. In MR channels, the write process converts temporal data into spatial patterns recorded on a magnetic medium, which transforms sequential correlation into spatial ISI/ITI. Data detection can be viewed as an image processing problem, proceeding from the 2D image of the shingled bits (see Fig. 1), to higher level abstractions of features by means of convolutional layers that finally allow classification of individual bits. Several variations of ConvNets are compared in terms of network complexity and performance. The best performing ConvNet detector can provide data storage density of up to 3.7489 Terabits/in^2 on low track pitch TDMR channel simulated with a grain flipping probabilistic (GFP) model.

II. SYSTEM MODEL

The ConvNet detection system assumes a discrete channel model for the kth readback signal $r(k)$:

$$r(k) = (\mathbf{h} * \mathbf{u})(k) + n_e(k), \qquad (1)$$

where $\mathbf{h}$ is the channel response, $\mathbf{u}$ are the coded bits, $*$ indicates 2D convolution and $n_e(k)$ is reader electronics additive white Gaussian noise (AWGN). The channel response $\mathbf{h}$ is implicitly time varying and pattern dependent, because the channel is inherently nonlinear. Therefore, pattern-dependent media noise arises. The system is trained and tested using data from a GFP model, a realistic model which closely replicates output from micro-magnetic simulations but can be generated several orders of magnitude faster [3]. The simulated media has grain density of 11.4 Teragrains/in^2. Waveforms include 5 tracks (tracks 0 through 4), each of length 41,206 bits; only the coded bits for the three central tracks (tracks 1-3) are available. Fig. 1 shows a capture of the GFP readback signals. The blue and red stripes represent −1 and +1 coded bits; they are curved because of the shingled writing process. The ConvNet system is tested on two GFP data sets, each containing 100 blocks. Both data sets have bit length (BL) 11 nm, but different track pitches (TP): TP 15 nm (equivalently 2.916 grains per coded bit (GPB)) and TP 18 nm (equivalently 3.491 GPB).

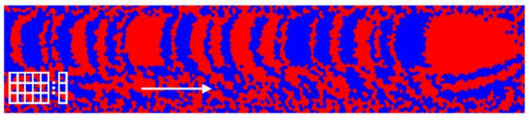

Fig. 1 A capture of the readback signals simulated from grain flipping probabilistic (GFP) model.

III. ConvNet DETECTOR

Fig. 2 shows the block diagram for the proposed ConvNet detection system. Three identically structured ConvNets estimate tracks 1, 2 and 3 simultaneously in a downtrack sliding window. The ConvNet detector processes a 2D patch of readback signals $\mathbf{r}$ from three tracks and outputs a probability estimate for the center bit

Benjamin J. BELZER
E-mail: belzer@wsu.edu
tel: +1-509-335-4970
Work funded by NSF grant CCF-1817083

of the patch. These soft estimates are converted to LLRs before being de-interleaved, weighted and passed to a low-density parity check (LDPC) decoder. The dotted lines and box in Fig. 2 indicate future work, i.e., iterative detection between the ConvNet detector and the LDPC decoder.

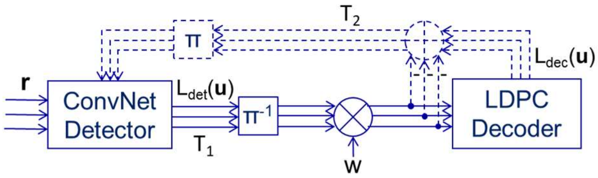

Fig. 2 Block diagram for the ConvNet detection system

The ConvNet detector has a general architecture of [INPUT - CONV - RELU - CONV - RELU - FC]. INPUT denotes the input to the network. It is a matrix representation of an image-like object: $[3 \times w_i]$, where w_i is the width of the input. CONV layer $\#k$ computes dot products between c_k different trained filters of size $[3 \times w_{c_k}]$ (white rectangles in Fig. 1) and same-sized patches in its input (assuming input dimension $[h \times w]$), with the filter moving in a sliding window fashion in both h and w direction of the layer input (white arrow in Fig. 1). The dimension of CONV layer output is $[h \times w]$ with the help of zero padding. RELU layer applies the rectified linear unit (ReLU) activation function $f(x) = \max(0, x)$ element wise on its input. Between CONV and RELU layer, a batch normalization layer is added to accelerate the training [4]. FC stands for fully connected layer, wherein each node is connected to all nodes in the previous layer output, and an affine function $z_j = \sum_i w_{ij} x_i$ is applied to them. Probability estimates are then formed using the softmax function $p_{ik} = e^{x_{ik}} / \sum_{k=1}^{K} e^{x_{ik}}$, and binary classification can be made. The network is optimized during training using gradient descent with cross entropy loss $J = \sum_i \sum_{k=1}^{2} 1(\hat{u}(i) = k) \times \log(p_{ik})$. Experiments show that two stacks of CONV-RELU layers are sufficient to yield a low bit error rate (BER) or 10^{-5} or lower.

IV. SIMULATION RESULTS

Table I summarizes the BER performance and computational complexity of ConvNets with different values of w_i, w_{c_k}, c_k, for track 2 of TP 15 nm data set. 80 blocks are used for training and the remaining 20 blocks for testing. Comparing the first four rows, input width $w_i = 21$ yields lowest BER. Comparing the last four rows, we conclude the number of filters c_k at each CONV layer has the most influence on both complexity and performance. Specifically, rows 4 and 5 show that reducing both c_k by half lowers the complexity by roughly a factor of 3.8 but only suffers a factor of 3 BER increase. Table II shows average user areal density (UAD) results over tracks 1, 2 and 3 from the ConvNets in rows 4 and 5 of Table I, for both TP 15 nm and TP 18 nm datasets. The UAD values in parentheses are scaled density values after a 6.4 nm squeeze margin. Complexity-performance tradeoff can be helpful in real-time hardware implementation.

Table I ConvNet BER and complexity, TP 15nm, track 2

Network Structure	BER train-test	# learnables
[3,11], [3,11]*128, [3,9]*64	0.0060-0.0118	230,210
[3,15], [3,11]*128, [3,9]*64	0.0021-0.0053	231,746
[3,17], [3,11]*128, [3,9]*64	0.0016-0.0040	232,514
[3,21], [3,11]*128, [3,9]*64	**0.0011-0.0029**	**234,050**
[3,21], [3,11]*64, [3,9]*32	**0.0035-0.0068**	**61,730**
[3,21], [3,9]*128, [3,7]*64	0.0017-0.0041	184,130
[3,21], [3,11]*32, [3,9]*16	0.0186-0.0213	17,042

Table II Overall ConvNet density performance

TP	Network Structure	Average Code Rate	Average User Areal Density
15	[3,21], [3,11]*128, [3,9]*64	0.9567	**3.7489** (2.6278)
15	[3,21], [3,11]*64, [3,9]*32	0.9367	**3.6706** (2.5728)
18	[3,21], [3,11]*128, [3,9]*64	0.9783	3.1949 (2.3569)
18	[3,21], [3,11]*64, [3,9]*32	0.9700	3.1677 (2.3368)

REFERENCES

1) J. Moon and J. Park, "Pattern-dependent noise prediction in signal dependent noise," *IEEE Jour. Sel. Areas Commun.*, vol. 19, no. 4, pp. 730–743, Apr 2001.
2) Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.
3) K. S. Chan, R. Radhakrishnan, K. Eason, M. R. Elidrissi, J. J. Miles, B. Vasic, and A. R. Krishnan, "Channel models and detectors for two-dimensional magnetic recording," *IEEE Trans. Mag.*, vol. 46, no. 3, pp. 804–811, March 2010.
4) S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift." arXiv preprint arXiv:1502.03167, 2015.