Advanced Data Recovery Techniques: Using machine learning in Data Recovery

Igor Sestanj | David Edwards

Data Analyzers LLC, Lake Mary, Florida
info@datanalyzers.com
January 2019

Abstract – Machine learning (ML) and artificial intelligence (AI) algorithms are finding their way into the data storage technology. Recent developments in data storage devices suggest machine learning have tremendous potential in solid state drives (SSD) and flash controllers. On the other side most of the latest hard disk drives are hybrids, although not directly used for data storage NAND flash memory has been widely popular among hard disk drive manufacturers as a worthy cache solution. In this paper we will try to elaborate ways we can use ML and AI technology in order to recover data once a data storage device failed for reasons other than hardware. Machine learning and AI based technologies appears to be extremely useful for certain operations that data recovery engineers use on a daily basis such as: sorting, carving or analysis.

Introduction - When Shingled Magnetic Recording (SMR) was introduced several years back not many of us in the data recovery industry were able to foresee future events that will lead us to DM-SMR hard drives (Drive-Managed SMR) which uses translator algorithms very similar to one we find on solid state devices. Similarity comes from data organization; sectors, tracks and bands which resembles structure found on NAND flash memory. Current data recovery techniques are all based on reverse engineering approach widely popular in Russia and China where most of the data recovery equipment has been developed and produced.

Few years back, in days when Federal Information and Protection Standard (FIPS) kicked in and manufacturers had to apply security measures in order to be able to sell to government and related organizations. Although not required for private parties security measures such as encryption offered that extra security for end users while in fact enabled manufacturers to "hide" what was under the hood. In the same time work of Geoffrey E Hinton[i] led to what we call ML and AI solution found in our cellphones, on the web, in language translation algorithms and elsewhere. Although in many ways this technology is still in its infancy one fact remains; ML and AI can be designed to learn without the aid of a human teacher! This may well be the start of autonomous intelligent "brain-like" machinery able to process, carve and sort all that "raw" data.

It has been established that artificial neural networks (ANN)[xiii], ML and AI designs, are capable of "understanding" information but only when large amount of data is available to process and therefore - learn.

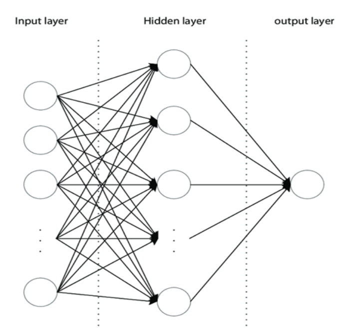

Fig 1.

Artificial Neural Network is divided into 3 segments (layers). Input Layer - The training observations are fed through these neurons; Hidden Layers - These are the intermediate layers between input and output which help the Neural Network learn the complicated relationships involved in data; Output Layer - The final output is extracted from previous two layers.

The very basic neural network (Fig 1) shows what is the general approach in modern artificial neural networks. Before moving further we'd like to point this work is currently conceptual. Purpose of this paper is to describe issues modern data recovery industry is facing, present finding related to ML and AI in this field and hopefully encourage others to continue.

Data recovery – process of salvaging inaccessible, lost, corrupted, damaged or formatted data from a secondary storage, removable media, when the data stored in them cannot be accessed. While in many cases recovery is required due to physical damage to the storage devices such cases are out of the scope of this paper. We'd like to focus on cases when there is no mechanical or electronic failure while storage device fails to access data due to reasons related to drive's firmware. Firmware in general can be described as specific class of computer software that provides the low-level control for the device's specific hardware.[ii] During data recovery process firmware sometimes has to be altered to allow the device to access data. Most data recovery enterprises around the globe use an industry standard equipment to perform such task. However there are many cases where these equipment is not able to complete this task mostly because that specific device inner workings has not been researched yet. One such example is adaptive or dynamic XOR. Importance of in-house research & development projects is in such cases critical. Although we will discuss SSD self-encryption further in this document let just say that for many self-encrypted SSDs solution is not yet available. In such cases partnerships with manufacturers may be the only reasonable way to retrieve user data. However, not all manufacturers work closely with data recovery service providers.

Introduction to ANN and ML – Discovered in early 50s Machine Learning (ML) is a subset of Artificial Neural Networking (ANN) which uses statistical techniques to give computers the ability to "learn" with data, without being explicitly programmed.[xiv] ANNs are computational models that are loosely inspired by their biological counterparts. [xv] Artificial intelligence is currently driving some of the most popular services and products of today. ANN lies in the heart of products such as self driving cars, image or voice recognition, home assistance devices, etc. Recently some data storage manufacturers are already employing ANNs and ML for their products[xvi.] This paper also explores options to recover data in such cases as we may not be able to reverse engineer those using traditional methods.

Lets start by understanding how a nervous system works. It comprises of millions of neurons a single neuron has the structure showed on (Fig. 2)

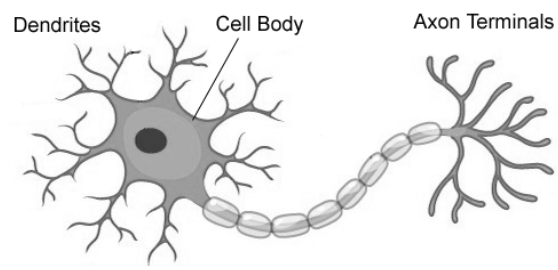

Fig 2.

The major components of a neuron are:

- **Dendrites** - Receives inputs from other neurons as an electrical impulse
- **Cell Body**- Generates inferences from those inputs and decide on actions
- **Axon terminals** - Transmit outputs again as an electrical impulse

In simple terms, a neuron takes input from other neurons through the dendrites, performs the required processing on the input and sends another electrical pulse through the axiom into the terminal nodes from where it is transmitted to other neurons. ANN works in a very similar fashion as it is able to receive input vector I=[i1, i2,… , in], and generate appropriate output vector O=[o1, o2, … , om]. The ANN contains several connected elementary calculation units, which are also called neurons.

Figure below (Fig. 3) shows a schematic representation of an artificial neuron with input vector (with r elements) and characteristic structure of the feed forward ANN with k hidden layers. Each of the input elements X1, X2, ..., Xr is multiplied with the corresponding weight of the connection Wi,1, Wi,2,..., Wi,r. The neuron sums these values and adds a bias bi (which is not present in all networks). The argument of the function (which is called transfer function) is given as follows:

$$\mathbf{a_i = x1w_{i,1} + x2w_{i,2} + \ldots + xrw_{i,r} + b_i}$$

while neuron produces output:

$$y_i = f(a_i) = f\left(\sum_{j=1}^{r} x_j w_{i,j} + b_i\right)$$

This output represents an input to the neurons of another layer, or an element of the output vector of the ANN.

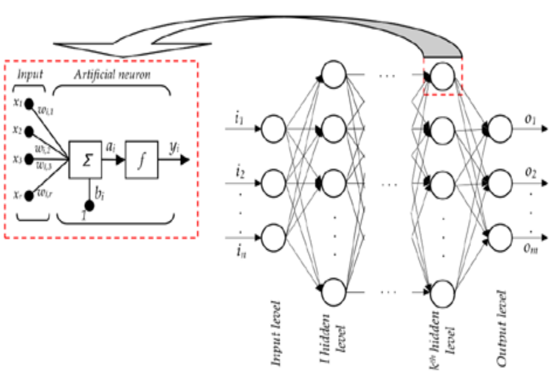

Fig 3.

This figure depicts a typical neural network. The input to each neuron are like the dendrites. Just like in biological nervous system, a neuron collates all the inputs and performs an operation on them after which, it transmits the output to all other neurons to which it is connected.

The different components of the neuron (Fig4.) are:

x1, x2,…, xN: Inputs to the neuron. These can either be the actual observations from input layer or an intermediate value from one of the hidden layers.

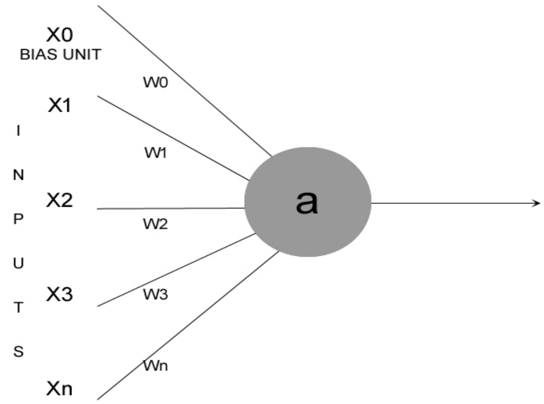

Fig 4.

x0: Bias unit. This is a constant value added to the input of the activation function. It works similar to an intercept term and typically has +1 value.

w0,w1, w2,…,wN: Weights on each input. Note that even bias unit has a weight.

a: Output of the neuron which is calculated as:

$$a = f\left(\sum_{i=0}^{N} w_i x_i\right)$$

Here f is known an activation function. This makes a Neural Network extremely flexible and imparts the capability to estimate complex non-linear relationships in data. It can be a Gaussian function, logistic function, hyperbolic function or even a linear function in simple cases.

Performance of ANN depends on the number of number of neurons, number of layers, transfer function, presence of a bias as well as on the way the neurons are connected. The speed of a neuron can further be increased by the use of Vedic multiplier.[xvii]

NAND flash data recovery issues (including SSD) – Flash based devices such as USB drives, memory cards or SSDs fail for a variety of reasons such as component failure, firmware corruption, and physical damage. Unplugging the drive while its writing firmware or developing an excessive amount of bit errors in the NAND chip which is the built in error correction control cannot correct are common issues.

When flash devices fail due to firmware our only way of recovery is to physically dump the NAND chip. For the purpose of this paper we will focus on firmware related issues.
Apart from other features one part of device's firmware is usually critical for data recovery, the FTL[iii]. Flash Translation Layer is the component that converts logical block addresses to physical address on the memory chip.

A common problem engineers face when reverse engineer a flash controller storage algorithm is XOR[iv] widely used in NAND flash controllers for writing purposes. Over 90% of flash drives we see uses XOR. When raw data is XOR'd it is very similar to it being encrypted. The main difference is XOR patterns are not unique, many devices by the same manufacturer can share the same XOR pattern. It is possible to manually search the raw data for blocks that resemble a XOR key. Once a XOR key has been discovered it can be extracted and then applied to all blocks of data in the NAND dump to reveal the original data. The process of locating and extracting XOR keys is time consuming and sometimes difficult. Some devices use multiple XOR keys as data can be XORed with one key while the Service area with another. However, if a XOR key cannot be located in the failed devices dump it is possible to use the exact same working device to assist in extracting a key.

If XOR was not hard enough, higher end devices like SSD's are using what we call dynamic XOR. Each block is XOR with a different key based on the block number.

Due to its complexity only recently have engineers developed solutions for some controllers that employ dynamic XOR. Self-encrypting SSDs use an encryption engine built into the controller and include a 256-bit AES encryption engine.

However, improving data security requires extra precautions many do not take into consideration until data is not accessible no more.

We are seeing people losing data due to controller failure while encryption keys are not shared with data recovery providers making them extremely difficult to recover data from.

Modern HDD recovery issues (including drives with Persistent cache and drives with SMR) – Please keep in mind this research paper does not cover data recovery cases caused by physical damage nor media damage related cases. Although modern drives are prone to these kind of failures goal of this research is to cover data recovery issues caused by firmware corruption (including media cache). Hard drive cache or disk buffer acts as temporary memory for the hard drive as it reads and writes data to the permanent storage on the platters. You can think of a hard drive's cache as being like RAM specifically for the hard drive. There are a bunch of different steps to retrieving data from a hard drive. Each one of them takes time, and it's rare that they sync up. Transfer from the hard drive via SATA usually moves much faster than the drive can read and write data to the platters. The disk buffer is often used to even out this flow of data and make the process much smoother.

SMR leverages the difference in the width of the disk's write head and read head to squeeze more tracks into the same area than Perpendicular recording (PMR).

In PMR, tracks have the width of the write head even though they are read with a narrow ad head, as seen in Figure 5 (a). In SMR, however, the tracks are written on top of each other, leaving just enough space for the read head to distinguish them, increasing track density, as seen in Figure 5 (b). Unlike PMR, however, overlapping writes cause the sector updates to corrupt data in adjacent tracks. Therefore, the surface of an SMR disk is divided into bands that are collections of narrow tracks divided by wide tracks called guard regions, as seen in Figure 5(c). A band in an SMR disk represents a unit that can be safely overwritten sequentially, beginning at the first track and ending at the last. A write to any sector in a band—except to sectors in the last track of the band—will require read-modify-write (RMW) of all the tracks forming the band.

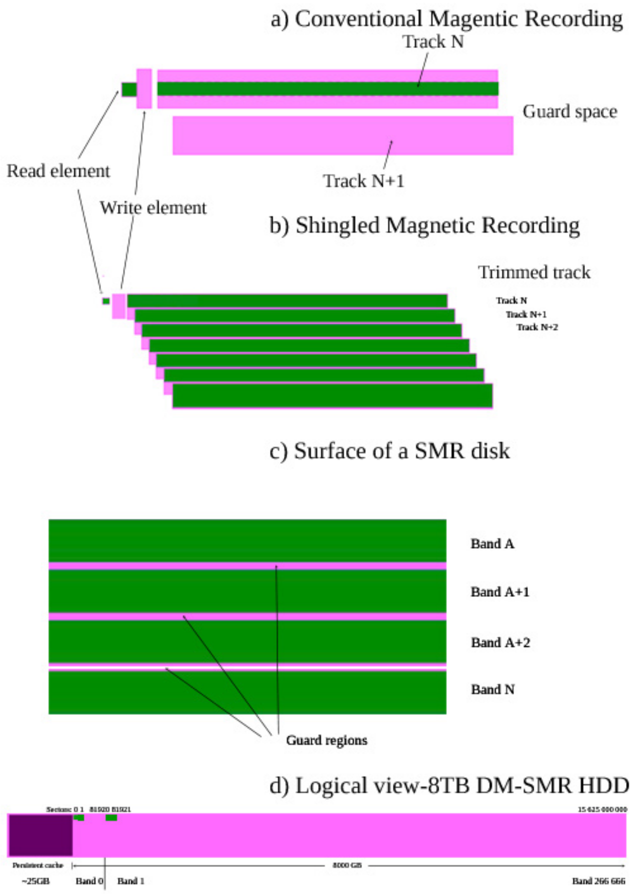

Fig. 5

(a) Conventional recording - tracks have the width of the write head.
(b) Shingled recording - tracks are laid partially on top of each other reducing the track width to the read head. This allows for more tracks, however, unlike with conventional recording, overwriting a sector corrupts other sectors.
(c) The surface of an SMR disk is divided into bands made up of multiple tracks separated by guard regions.
(d) An SMR disk also contains a persistent cache for absorbing random writes, in addition to a sequence of bands to whom a group of sectors are mapped.

Drive managed SMR (DM-SMR) disks, implement a Shingle Translation Layer (STL) - similar to the Flash Translation Layer (FTL) in SSDs - that manages data in firmware while presenting a drive as a block interface. All STLs proposed in the literature and found in actual DM-SMR disks contain one or more persistent caches for absorbing random writes, to avoid expensive RMW operation for each write. Consequently, when a write operation updates some part of a band, the STL writes the update to the persistent cache, and the band becomes “dirty”. The STL cleans a dirty band in the background or during idle times, by merging updates for the band from the persistent cache with unmodified data from the band, and writing back to the band, freeing space used by the updates in the persistent cache. The cost of cleaning a band may vary based on the type of the block mapping used. With dynamic mapping an STL can read a band, update it in memory, write the updated band to a different band, and fix the mapping, resulting in a read and a write of a band. With static mapping, however, an STL needs to persist the updated band to a scratch space first—directly overwriting the band can corrupt it in case of a power failure—resulting in a read and two writes of a band. Latest 8TB DM-SMR[xviii] drives with an average band size of 30MB, the disk has over 260,000 bands with sectors statically mapped to the bands, and about 25 GB of persistent cache, not visible to the host. When STL detects sequential writes it will starts streaming them directly to the bands, bypassing the persistent cache. Random writes, however, end up in the persistent cache, dirtying bands. Cleaning a single band takes about 2 seconds, or up to 45 seconds in extreme cases. STLs also differ in their cleaning strategies. Some STLs constantly clean in small amounts, while others clean during idle times. If the persistent cache fills before the workload completes, the STL is forced to interleave cleaning with work, reducing throughput.

T10 Zoned Block Commands (ZBC)[v] and T13 Zoned Device ATA Command Set (ZAC)[vi] standards abstract an SMR drive as a collection of zones (bands) which are logical non-overlapping collections of consecutive LBAs. There are two types of zones: conventional zones and write pointer zones which are conceptually mapped to bands consisting of CMR tracks and SMR tracks respectively. A write pointer zone is designed to be written in a log structure. It has an associated write pointer which indicates an LBA within the zone where the next write should target. By contrast, conventional zones have no write pointers. Majority of the zones in SMR drives are shingled.

In the real world Seagate disks are known to clean during idle times and to have static mapping[vii], meaning they have high throughput while the persistent cache is not full, and ultra-low throughput afterwards. Western Digital disks, on the other hand, are likely to clean constantly and have dynamic mapping, or they have lower throughput while the persistent cache is not full, but higher throughput after it fills. Resolving issues related to firmware corruption on these drive present a challenge due to new architecture, command set, lack of information, complexity, etc.

Machine learning and data recovery - Pattern recognition eventually evolved to machine learning and became popular between the 1970s and the 1980s. It focuses on how to make computer programs to perform intelligent "human-like" tasks, such as distinguishing an object from an image. Concepts such as decision tree, heuristic method, and quadratic discriminatory analysis were all introduced during this period. In the early 1990s, many realized that there was a more effective way to create pattern recognition algorithms, particularly replacing researchers with probability and statistics.[viii] This paradigm led to creation of machine learning. The goal of machine learning is to give a computer a collection of data and let the computer make its own conclusion with minimal human intervention. Meaning, that the 'machine' collects statistics from data, and then generating a probabilistic model to determine the most possible outcome. Properly designed machine learning algorithm always performs better than a person would because it is immune to cognitive bias and fatigue. This is where we'd like to pause for a moment and say that with latest improvements in data storage technologies[xix] it is clear human will no longer be able to deal with complex data structures. Although the greatest potential lays in deep learning, where there is minimal human intervention and bias because the parameters in the modes are learned from statistics. Unfortunately, deep learning is only possible with a considerable amount of statistics (big data) and strong arithmetic capabilities (such as: graphic processor) to optimize the mode. Therefore workings on such statistics may prove to be critical for future data recovery applications.

Machine learning could assist with automating process widely used when the goal is to retrieve data from a SMR drives, SSDs or NAND based data storage devices in general. The greatest potential might be distinguishing patterns such as locating xor keys in a dump and extract them. There are dozens of XOR patterns already extracted from previous cases that could be fed into the machine learning algorithm to help distinguish and locate new xor key patterns. Processes like sorting, carving and other data manipulations also can be improved using artificial intelligence.

Development of machine learning solutions and their applications on data recovery related problems requires collection of statistical data from raw data samples as well as from previously sorted/resolved cases. Collection and storage of such data must be done among many participating establishments using widely accessible cloud storage/sharing platforms. This way "machine" can compare same types of devices/models raw data to each other.

Understanding some of the tools available in Python programming language for data exploration may also prove to be critical. Python is an OOP (object-oriented programming) based, high level, interpreted programming language great for artificial intelligence. It works perfectly as a “glue” language to connect existing components together. Python’s support, ease of coding, ever evolving libraries make it a good choice for this project as it can implement the same logic with as much as 1/5th code as compared to other OOPs languages.[ix] Pyton libraries such as: Numpy (scientific computation), Scipy (advanced computing), Pybrain (machine learning) and AIMA which is Python implementation of algorithms from Russell and Norvig's 'Artificial Intelligence[x] all currently explored for the purpose of this project. At this point it is not clear how many epochs[xi] is necessary for any given process to be optimized and effectively used by AI driven machinery but it seems learning curve for such system if far shorter (narrower) than using any other technique.

Conclusion - It was not long ago, perhaps two years since Machine learning algorithms were proposed to deal with FTL complexity[xii]. Thus far we are seeing just a handful of such controllers. But it is expected this technology to catch up no latter than 2020. Traditional reverse engineering will not be capable to resolve issues and more innovative solutions are needed. Although industry is continuously shifting artificial intelligence has a great potential in data recovery as the next generation of data recovery engineers will have to deal with more complex math, adaptive electronics and “thinking machines”. Until then it is our job to lay the foundation as today’s machine learning designs are already adaptive enough, able to “learn on their own” and finally great at distinguishing data patterns.

Acknowledgements - Authors would like to thank Data Analyzers’s CEO, Mr. Andrew von Ramin Mapp for making this research possible and also other Global Data Recovery Alliance members for constructive discussions.

i - Geoffrey E. Hinton is internationally distinguished for his work on artificial neural nets, He has compared effects of brain damage with effects of losses in such a net, and found striking similarities with human impairment, such as for recognition of names and losses of categorization. | Anon (1998). "Certificate of election EC/1998/21: Geoffrey Everest Hinton". London: Royal Society.

ii - (Wikipedia) https://en.wikipedia.org/wiki/Firmware

iii - NAND Flash Data Recovery Cookbook - Part 1 | Igor Sestanj, June 2016 ISBN: 978-86-919771-0-8

iv - http://isweb.redwoods.edu/instruct/calderwoodd/diglogic/xor.htm

v - T10 draft | http://www.t10.org/ftp/zbcr01.pdf

vi - T13 draft | http://www.t13.org/Documents/UploadedDocuments/docs2015/di537r05-Zoned_Device_ATA_Command_Set_ZAC.pdf

vii - Shingled Magnetic Recording Areal Density Increase Requires New Data Management| Tim Feldman and Garth Gibson (June 2013)

viii - https://www.alibabacloud.com/blog/deep-learning-vs-machine-learning-vs-pattern-recognition_207110 (Alibaba bot)

ix - https://www.newgenapps.com/blog/python-for-ai-artificial-intelligence-ml

x - Artificial Intelligence: A Modern Approach | Peter Norvig, Stuart J. Russell (Dec 1994)

xi - http://www.fon.hum.uva.nl/praat/manual/epoch.html

xii - Machine Learning based Digital Signal Processor for NAND Flash Storage System | Wei Lin, 2017

xiii - Artificial neural networks (ANN) are computing systems vaguely inspired by the biological neural networks that constitute animal brains. - Artificial Neural Networks as Models of Neural Information Processing | Frontiers Research Topic

xiv - Supposedly paraphrased from: Samuel, Arthur (1959). "Some Studies in Machine Learning Using the Game of Checkers". IBM Journal of Research and Development. 3 (3): 210–229

xv - Research Topic-Artificial Neural Networks as Models of Neural Information Processing

xvi - A Self Learning Algorithm for NAND Flash Controllers | Hao Zhi, Lee | Phison Electronics Corp, 2017

xvii - Neuronal Logic gates realization using Vedic mathematics | Anshika, Yamuna SV , Nidhi Goee, S.Indu, 2015

xviii - Performance Evaluation of Host Aware Shingled Magnetic Recording (HA-SMR) Drives | Fenggang Wu, Ziqi Fan, Ming-Chang Yang, Baoquan Zhang, Xiongzi Ge, and David H.C. Du, Fellow, IEEE (Nov 2017)

xix - Using Machine Learning to Enhance Flash Endurance & Latency | Cloud Ze, 2017