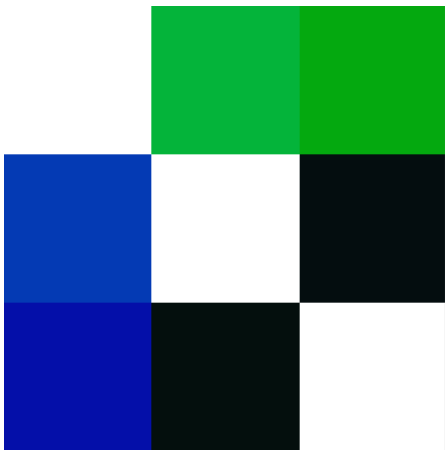
Black Tree

AutoML



NOTICES

SECURITIES: This booklet is ***not*** an offer, or a solicitation for an offer, to enter into any transaction. It is solely for informational purposes, only to describe a set of algorithms that implement machine learning and deep learning (the “algorithms”).

INTELLECTUAL PROPERTY: I (Charles Davi) retain all rights (copyright and otherwise) to all of the algorithms, data, charts, images, and other information presented in this booklet (the “information”). The information may not be used for any purpose whatsoever without my prior written consent, other than evaluating the information.

Vectorized Deep Learning

Charles Davi

March 16, 2021

Abstract

Many consumer devices can be used to perform parallel computations, and in a series of approximately five-hundred research notes,[1] I introduced a new and comprehensive model of artificial intelligence rooted in information theory and parallel computing that allows for classification and prediction in worst-case polynomial time, using what is effectively AutoML, since the user only needs to provide the datasets. This results in run-times that are simply incomparable to any other approach to A.I. of which I'm aware, with classifications at times taking seconds over datasets comprised of tens of millions of vectors, even when run on consumer devices. Below is a summary of the results of this model as applied to UCI and MNIST datasets, as well as several novel datasets rooted in thermodynamics. All of the code necessary to follow along is linked to below.

[1] All of the research notes and applicable code are publicly available on my ResearchGate homepage.

1 Vectorized Deep Learning

To follow along, simply download and install Octave, which is free, and download my A.I. library, which includes all of the code referenced in this article. Note that my A.I. library may NOT be used for commercial purposes.[2]

For a mathematically rigorous, theoretical explanation, of why these algorithms work, see my paper, "Analyzing Dataset Consistency". For an in-depth, practical explanation of why these algorithms work, including applications to other datasets, see my paper, "A New Model of Artificial Intelligence".

[2]I retain all rights (copyright and otherwise) to all of the materials, algorithms, and all other works published in this article. The algorithms in my A.I. library may NOT be used for commercial purposes without my express, prior, written consent. For the avoidance of doubt, you may NOT modify, or redistribute any of this material, in particular the algorithms, without my express, prior, written consent.

1.1 Data Classification

UCI Ionosphere Dataset

This algorithm effectively iterates through increasing levels of discernment, until it finds the level that generates the greatest change in what would be the perceived structure of the dataset. This instance of the algorithm is fully vectorized, with only a hypothetical number of iterations.

- **Size**: 351×34.
- **Task**: Unsupervised Clustering.

- **Average Accuracy**: 99.97%.
- **Runtime**: 0.14465 seconds.[3]

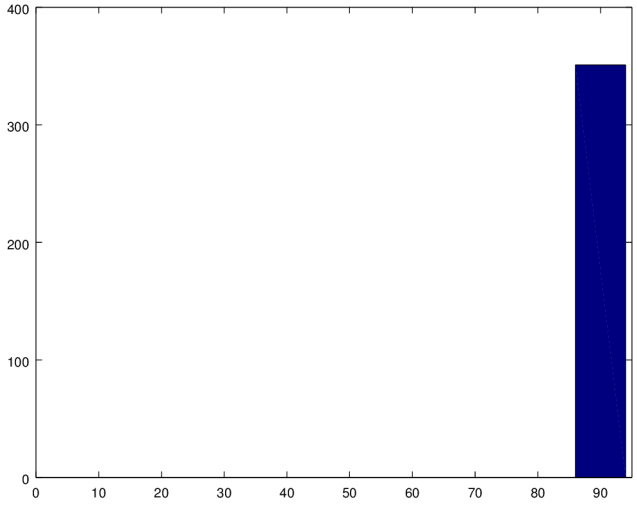


Figure 1: The number of rows with an accuracy of at least x%.

The average accuracy reported above is the average accuracy across all clusters. For a given cluster, the accuracy is calculated by counting the number of classification errors in the cluster, and dividing by the number of rows in the dataset, since the clusters are not mutually exclusive. This ratio is then

[3]All runtimes referenced in this article were generated on an iMac 3.2 GHz Intel Core i5.

subtracted from 1. The average number of elements per cluster is 6.9259, the minimum number of elements is 1, and the maximum is 46. The minimum accuracy is 97.7%, and the maximum accuracy is 100%.

UCI Wine Dataset

This algorithm is the same as the one used for the Ionosphere Dataset above, with an additional step that first normalizes the dataset.

- **Size**: 178×13.
- **Task**: Normalization; Unsupervised Clustering.
- **Average Accuracy**: 99.981%.
- **Runtime**: Normalization, 0.925937 seconds; Clustering, 0.0356669 seconds.

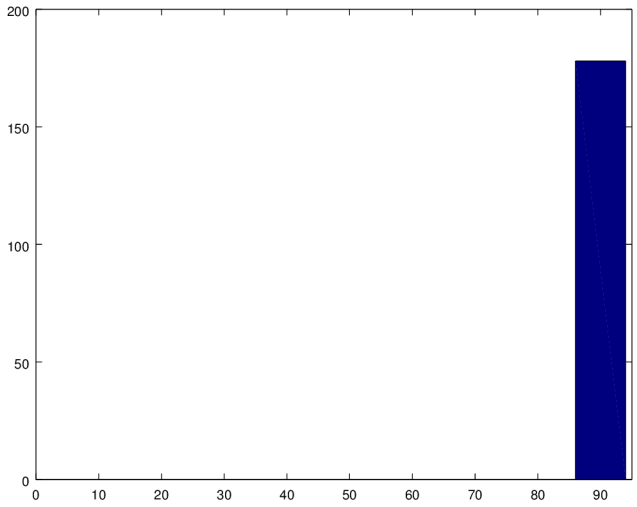


Figure 2: The number of rows with an accuracy of at least x%.

The average accuracy reported above is the average accuracy across all clusters. For a given cluster, the accuracy is calculated by counting the number of classification errors in the cluster, and dividing by the number of rows in the dataset, since the clusters are not mutually exclusive. This ratio is then subtracted from 1. The average number of elements per cluster is 2.3708, the

minimum number of elements is 1, and the maximum is 9. The minimum accuracy is 99.438%, and the maximum accuracy is 100%.

MNIST Numerical Dataset

This algorithm is a fully vectorized implementation of the nearest neighbor method.

- **Size**: $2,500 \times 121$.[4]
- **Task**: Unsupervised Classification Prediction.
- **Accuracy**: 93.60%.
- **Runtime**: 52.4577 seconds.

The accuracy reported above is the accuracy over all predictions, calculated by counting the total number of prediction errors, and dividing by the number of rows in the dataset. This ratio is then subtracted from 1.

1.2 Image Processing

I've also developed generalized image processing algorithms that allow any basic, single object image dataset, to be quickly and reliably transformed into a data structure, that can then be used for clustering and classification. Specifically, the algorithms generate a super-pixel representation of each image in the dataset, that can then be fed to a classifier or clustering algorithm. This is done by first processing a representative image from the dataset, and the information generated from the representative image is then used to process the remaining images in the dataset, at a much more efficient rate.

These algorithms would be particularly useful in real-time image classification problems, where a single object is presented to a camera, or otherwise fed to a computer for identification, and for whatever reason, the hardware is inexpensive or low-energy.

iPhone Photo Dataset

[4] $2,500$ images from the dataset are first loaded into memory, and then processed, prior to clustering, generating a $2,500 \times 121$ matrix. The runtime listed below is the runtime for only the prediction algorithm itself. For an explanation of the entire process, see, *"A New Model of Artificial Intelligence: Application to Data II"*.

- **Task**: Initial Analysis (Single Image).
- **Original Image Size**: 3264 × 2448 × 3 pixels.
- **Runtime**: 10.2044 seconds.

- **Task**: Process Dataset (30 images)
- **Original Image Size**: 3264 × 2448 × 3 pixels.
- **Runtime**: 0.069453 seconds, on average, per image.

Below are three images related to the iPhone Picture Dataset, for context, which are simply photos I took of grocery items from Whole Foods, using my iPhone: The first is an image of a grapefruit, the second is the super-pixel image fed to the classifier algorithm, and the third shows the boundary data that generated the super-pixel image.

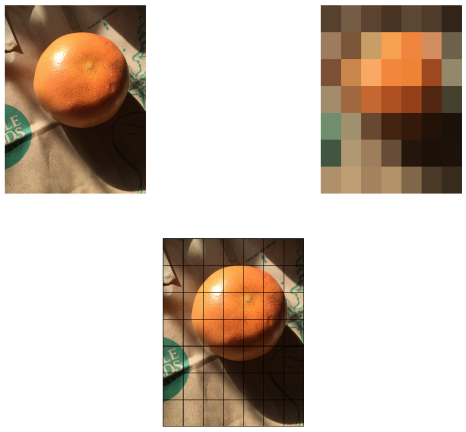


Figure 3: The original (left), the super-pixel image (right), and the boundaries (bottom).

MNIST Numerical Dataset

- **Task**: Initial Analysis (Single Image).
- **Original Image Size**: 28 × 28 pixels.
- **Runtime**: 0.168235 seconds.

- **Task**: Process Dataset (7,500 images)

- **Original Image Size**: 28 × 28 pixels.
- **Runtime**: 0.0031633 seconds, on average, per image.

MNIST Fashion Dataset

- **Task**: Initial Analysis (Single Image).
- **Original Image Size**: 28 × 28 pixels.
- **Runtime**: 0.16334 seconds.

- **Task**: Process Dataset (7,500 images)
- **Original Image Size**: 28 × 28 pixels.
- **Runtime**: 0.0036458 seconds, on average, per image.

1.3 Massive Datasets

The algorithms used in this section were developed to allow for deep learning techniques to be applied to thermodynamic systems, making use of both fully and partially vectorized algorithms. They are, however, generalized algorithms that likely have applications in other areas of study.

Two-State Gas Dataset

This dataset consists of two classes of collections of vectors:

(a) one representing the particles of a gas in a compressed volume, and

(b) another representing the particles of the gas in an expanded volume.

These two classes are intended to represent the two possible macrostates of the gas, compressed or expanded. Each class consists of 50 configurations, for a total of 100 configurations, intended to represent the microstates of the gas. Each configuration consists of $15,000$ vectors. The classification task is to cluster the 100 microstate configurations in a manner that is consistent with the two hidden macrostate classifiers, compressed or expanded.

The algorithm applied to this dataset also iterates through increasing levels of discernment, like the algorithms used in Section 1.1 above. However, this algorithm makes use of a vectorized operator that can quickly compare two large collections of vectors, as single operands, in turn allowing for the efficient comparison of microstates of complex systems.

- **Size**: $1,500,000 \times 3$.
- **Task**: Identify the macrostates of a gas.
- **Accuracy**: 100%.
- **Runtime**: 15.6721 seconds.

The accuracy is calculated by counting the number of classification errors, and dividing by the number of rows in the dataset. This ratio is then subtracted from 1.

Expanding Gas Dataset

This dataset consists of two classes of sequences:

(a) one representing the particles of a gas expanding at a slow rate, and

(b) another representing the particles of the gas expanding at a fast rate.

Each sequence of expansion consists of 15 observations, and there are 600 sequences. Each observation consists of $10,000$ vectors, representing the particles of the gas. The classification task is to cluster the 600 sequences in a manner that is consistent with the two hidden classifiers, slow or fast.

The algorithm applied to this dataset first compresses the dataset, by sorting and then embedding the dataset on the real number line. The sorting algorithm again makes use of a vectorized operator that can quickly compare two large collections of vectors. Then, a clustering algorithm similar to the one used in Section 1.1 above is applied to the embedded dataset. The bulk of the work done by the algorithm is sorting the dataset, ultimately allowing the dataset to be compressed from a $90,000,000 \times 3$ matrix, into a 600×15 matrix.

- **Size**: $90,000,000 \times 3$.
- **Task**: Identify the different rates at which a gas expands.
- **Accuracy**: 100%.
- **Runtime**: Sorting, 21.242 minutes; Embedding, 49.8727 seconds; Clustering, 0.0923202 seconds.

The accuracy is calculated by counting the number of classification errors, and dividing by the number of rows in the dataset. This ratio is then subtracted from 1.

Statistical Spheres Dataset

This dataset consists of some fixed number of K spheres in Euclidean 3-space, each of which consists of some number of points, producing shapes that are not solid, but nonetheless visually distinct. The classification task is to cluster the points in a manner that is consistent with the K hidden classifiers, representing the K distinct objects in the space.

The algorithm applied to this dataset also iterates through increasing levels of discernment, like the algorithms used in Section 1.1 above. However, this algorithm makes use of a different technique that can quickly cluster large collections of low-dimensional vectors.

- **Size**: $1,048,724 \times 3$.
- **Task**: Identify and cluster objects in Euclidean 3-Space.

- **Accuracy**: 100%.
- **Runtime**: 114.036 seconds.

The accuracy is calculated by counting the number of classification errors, and dividing by the number of rows in the dataset. This ratio is then subtracted from 1.

1.4 Other Algorithms

The balance of my work includes applications of these algorithms and others to image and video classifications, object tracking, shape classification, function approximation, as well as other algorithms specific to physics, including algorithms capable of estimating object velocities, and predicting projectile paths, given 3-D point data.

2 Financing Options

I would entertain financing for either -

1. An outright sale of my entire library of A.I. software, as it stands; or

2. Capital for a sales team and office space, to market the software, in exchange for equity.

In either case, I would likely insist on an opinion from counsel that the commercial distribution of the software does not violate all applicable laws, though I could be comfortable with a more reasoned opinion from competent counsel expert in the laws that relate to the distribution of software that could be used for military purposes.

Moreover, in either case, I would likely insist on the right to continue to develop my work, though I would entertain reasonable restrictions, such as a right of first refusal before any commercial offering, or perhaps developing my work outside of the public domain. In the latter case, I would likely insist on some form of explicit protection for the intellectual property I develop, since in that case, I would no longer be able to rely on copyright through publication.

3 About Me

I am a mathematician that worked in financial services for eight years, most recently at BlackRock, spending a significant portion of my free time conducting research in information theory. I spent the last five years conducting this research full-time, and the last two years coding and writing full-time.

In addition to my scientific writing, I've also published in The Atlantic,[5] and elsewhere, writing about banking, finance, and economics, which was widely cited by bank regulators, and other legal and financial professionals, including Judge Richard Posner.[6]

I'm a fairly prolific composer of both classical and contemporary music,[7] having studied piano and voice at Manhattan School of Music Prep; I was a professional audio engineer through an apprenticeship to a family friend; and I also wrote a loosely autobiographical epic poem during the Covid-19 quarantine in New York City.[8]

I received my J.D. from New York University School of Law, and my B.A. in Computer Science from Hunter College, City University of New York.

[5] My byline at *The Atlantic*.
[6] See page 50, footnote 5 of, "*The Crisis of Capitalist Democracy" (2011)*.
[7] A collection of my most recent recordings on *Soundcloud*.
[8] My book, *"Sketches of the Inchoate"*.