Maths in ML

Why Maths ?

What Maths ?

How much Maths ?

Examples ?

Linear Algebra – Representation and manipulation of data

Statistics – Core of ML

Calculus – How to develop and optimize Algo’s

Probability – Likelihood of an event occuring

Examples where Linear Algebra is used -

1. Dataset and Data Files
2. Images and Photographs
3. One-Hot Encoding
4. Linear Regression
5. Regularization
6. Principal Component Analysis
7. Singular-Value Decomposition
8. Latent Semantic Analysis
9. Recommender Systems
10. Deep Learning

**1. Dataset and Data Files**

In machine learning, you fit a model on a dataset.

This is the table-like set of numbers where each row represents an observation and each column represents a feature of the observation.

For example, below is a snippet of the [Iris flowers dataset](http://archive.ics.uci.edu/ml/datasets/Iris):



|  |  |
| --- | --- |
| 1  2  3  4  5 | 5.1,3.5,1.4,0.2,Iris-setosa  4.9,3.0,1.4,0.2,Iris-setosa  4.7,3.2,1.3,0.2,Iris-setosa  4.6,3.1,1.5,0.2,Iris-setosa  5.0,3.6,1.4,0.2,Iris-setosa |

This data is in fact a matrix: a key data structure in linear algebra.

Further, when you split the data into inputs and outputs to fit a supervised machine learning model, such as the measurements and the flower species, you have a matrix (X) and a vector (y). The vector is another key data structure in linear algebra.

Each row has the same length, i.e. the same number of columns, therefore we can say that the data is vectorized where rows can be provided to a model one at a time or in a batch and the model can be pre-configured to expect rows of a fixed width.

**2. Images and Photographs**

Perhaps you are more used to working with images or photographs in computer vision applications.

Each image that you work with is itself a table structure with a width and height and one pixel value in each cell for black and white images or 3 pixel values in each cell for a color image.

A photo is yet another example of a matrix from linear algebra.

Operations on the image, such as cropping, scaling, shearing, and so on are all described using the notation and operations of linear algebra.

**3. One Hot Encoding**

Sometimes you work with categorical data in machine learning.

Perhaps the class labels for classification problems, or perhaps categorical input variables.

It is common to encode categorical variables to make them easier to work with and learn by some techniques. A popular encoding for categorical variables is the one hot encoding.

A one hot encoding is where a table is created to represent the variable with one column for each category and a row for each example in the dataset. A check, or one-value, is added in the column for the categorical value for a given row, and a zero-value is added to all other columns.

For example, the color variable with the 3 rows:



|  |  |
| --- | --- |
| 1  2  3  4 | red  green  blue  ... |

Might be encoded as:



|  |  |
| --- | --- |
| 1  2  3  4  5 | red, green, blue  1, 0, 0  0, 1, 0  0, 0, 1  ... |

Each row is encoded as a binary vector, a vector with zero or one values and this is an example of a sparse representation, a whole sub-field of linear algebra.

**4. Linear Regression**

Linear regression is an old method from statistics for describing the relationships between variables.

It is often used in machine learning for predicting numerical values in simpler regression problems.

There are many ways to describe and solve the linear regression problem, i.e. finding a set of coefficients that when multiplied by each of the input variables and added together results in the best prediction of the output variable.

If you have used a machine learning tool or library, the most common way of solving linear regression is via a least squares optimization that is solved using matrix factorization methods from linear regression, such as an LU decomposition or a singular-value decomposition, or SVD.

Even the common way of summarizing the linear regression equation uses linear algebra notation:



|  |  |
| --- | --- |
| 1 | y = A . b |

Where y is the output variable A is the dataset and b are the model coefficients.

**5. Regularization**

In applied machine learning, we often seek the simplest possible models that achieve the best skill on our problem.

Simpler models are often better at generalizing from specific examples to unseen data.

In many methods that involve coefficients, such as regression methods and artificial neural networks, simpler models are often characterized by models that have smaller coefficient values.

A technique that is often used to encourage a model to minimize the size of coefficients while it is being fit on data is called regularization. Common implementations include the L2 and L1 forms of regularization.

Both of these forms of regularization are in fact a measure of the magnitude or length of the coefficients as a vector and are methods lifted directly from linear algebra called the vector norm.

**6. Principal Component Analysis**

Often, a dataset has many columns, perhaps tens, hundreds, thousands, or more.

Modeling data with many features is challenging, and models built from data that include irrelevant features are often less skillful than models trained from the most relevant data.

It is hard to know which features of the data are relevant and which are not.

Methods for automatically reducing the number of columns of a dataset are called dimensionality reduction, and perhaps the most popular method is called the principal component analysis, or PCA for short.

This method is used in machine learning to create projections of high-dimensional data for both visualization and for training models.

The core of the PCA method is a matrix factorization method from linear algebra. The eigendecomposition can be used and more robust implementations may use the singular-value decomposition, or SVD.

**7. Singular-Value Decomposition**

Another popular dimensionality reduction method is the singular-value decomposition method, or SVD for short.

As mentioned, and as the name of the method suggests, it is a matrix factorization method from the field of linear algebra.

It has wide use in linear algebra and can be used directly in applications such as feature selection, visualization, noise reduction, and more.

We will see two more cases below of using the SVD in machine learning.

**8. Latent Semantic Analysis**

In the sub-field of machine learning for working with text data called natural language processing, it is common to represent documents as large matrices of word occurrences.

For example, the columns of the matrix may be the known words in the vocabulary and rows may be sentences, paragraphs, pages, or documents of text with cells in the matrix marked as the count or frequency of the number of times the word occurred.

This is a sparse matrix representation of the text. Matrix factorization methods, such as the singular-value decomposition can be applied to this sparse matrix, which has the effect of distilling the representation down to its most relevant essence. Documents processed in this way are much easier to compare, query, and use as the basis for a supervised machine learning model.

This form of data preparation is called Latent Semantic Analysis, or LSA for short, and is also known by the name Latent Semantic Indexing, or LSI.

**9. Recommender Systems**

Predictive modeling problems that involve the recommendation of products are called recommender systems, a sub-field of machine learning.

Examples include the recommendation of books based on previous purchases and purchases by customers like you on Amazon, and the recommendation of movies and TV shows to watch based on your viewing history and viewing history of subscribers like you on Netflix.

The development of recommender systems is primarily concerned with linear algebra methods. A simple example is in the calculation of the similarity between sparse customer behavior vectors using distance measures such as Euclidean distance or dot products.

Matrix factorization methods like the singular-value decomposition are used widely in recommender systems to distill item and user data to their essence for querying and searching and comparison.

**10. Deep Learning**

Artificial neural networks are nonlinear machine learning algorithms that are inspired by elements of the information processing in the brain and have proven effective at a range of problems, not the least of which is predictive modeling.

Deep learning is the recent resurgence in the use of artificial neural networks with newer methods and faster hardware that allow for the development and training of larger and deeper (more layers) networks on very large datasets. Deep learning methods are routinely achieving state-of-the-art results on a range of challenging problems such as machine translation, photo captioning, speech recognition, and much more.

At their core, the execution of neural networks involves linear algebra data structures multiplied and added together. Scaled up to multiple dimensions, deep learning methods work with vectors, matrices, and even tensors of inputs and coefficients, where a tensor is a matrix with more than two dimensions.

Linear algebra is central to the description of deep learning methods via matrix notation to the implementation of deep learning methods such as Google’s TensorFlow Python library that has the word “tensor” in its name.

**Summary**

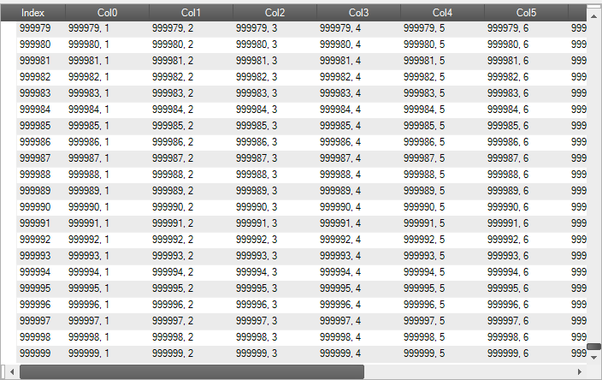
Specifically, you learned:

* The use of linear algebra structures when working with data such as tabular datasets and images.
* Linear algebra concepts when working with data preparation such as one hot encoding and dimensionality reduction.
* The ingrained use of linear algebra notation and methods in sub-fields such as deep learning, natural language processing, and recommender systems.

Applications of Linear Algebra

Let the below images show you the real application of Linear Algebra.

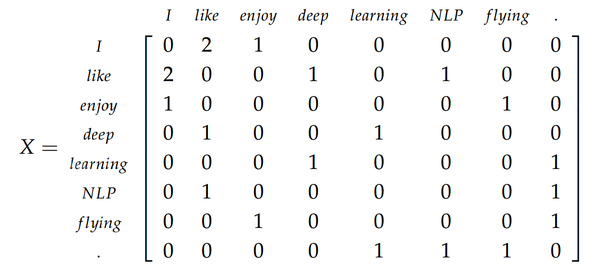
* **Dataset**- It is basically a matrix, **linear algebra** is here.



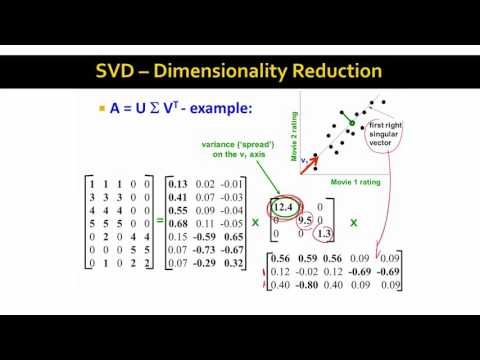
* **Image processing -** Ever heard of pixel in image, **linear algebra** is here.



* **Natural Language Processing -** Tokenization[[1]](https://www.quora.com/What-are-the-applications-of-linear-algebra-in-machine-learning#OUtMb), lemmanisation[[2]](https://www.quora.com/What-are-the-applications-of-linear-algebra-in-machine-learning#TbrAN) and normalization[[3]](https://www.quora.com/What-are-the-applications-of-linear-algebra-in-machine-learning#CeQgG) involves matrix. **Linear algebra** is here.



* **Dimensionality Reduction**[[4]](https://www.quora.com/What-are-the-applications-of-linear-algebra-in-machine-learning#yJgnT) **-** Read wiki. **Linear algebra** is also here.



* **Deep Learning** - The mathematics behind deep learning includes **Linear Algebra**. Google’s deep learning library TensorFlow itself includes word **Tensor** which can be thought as generalized matrices. There must be some reason behind giving this name.