Mathematical Foundations for Machine Learning

1. Importance of Linear Algebra

Linear algebra is the foundational language of machine learning. Data, models, and operations are all expressed using its concepts, making it indispensable for representing and solving problems efficiently.

Data Representation

- **Vectors**: Individual data points are often represented as **vectors**. A vector can represent a single observation with multiple features. For instance, a house can be represented as a vector like [square_footage, bedrooms, bathrooms]. In natural language processing, words or documents can be converted into vectors (word embeddings) to capture their semantic meaning.
- **Matrices**: Entire datasets are represented as **matrices**. Each row of the matrix is a data point (a vector), and each column is a feature. Images, for example, can be stored as matrices of pixel values. This structure allows for fast, parallel computation on the entire dataset at once.
- **Tensors**: In deep learning, data is often organized into **tensors**, which are generalizations of vectors and matrices to higher dimensions. For example, a color image can be a 3D tensor with dimensions for height, width, and color channels.

Model Representation and Operations

- **Linear Transformations**: Many machine learning algorithms, particularly deep neural networks, are essentially chains of **linear transformations**. Each layer of a neural network performs a matrix multiplication on the input data, followed by a non-linear activation function. This process transforms the data from one space to another, allowing the model to learn complex patterns.
- **Weights and Coefficients**: The parameters that a model learns are represented as vectors or matrices. In linear regression, the coefficients (or weights) that determine the slope and intercept of the line are a vector. In a neural network, the connections between neurons are represented as weight matrices.
- **Optimization**: Algorithms like **gradient descent**, used to train most machine learning models, rely on linear algebra to calculate the gradient (the direction of steepest descent) of the cost function. This process involves matrix calculus, which is a branch of linear algebra.

Key Applications & Algorithms

- **Linear Regression**: As its name suggests, this algorithm is built entirely on linear algebra. It solves a system of linear equations to find the best-fit line (or hyperplane) by minimizing the sum of squared errors, often using matrix factorization techniques.
- **Dimensionality Reduction**: Techniques like **Principal Component Analysis (PCA)** are based on linear algebra concepts such as **eigenvalues** and **eigenvectors**. PCA uses

these to find new axes (principal components) that capture the most variance in the data, allowing for the reduction of features while retaining important information.

- **Singular Value Decomposition (SVD)**: This is a powerful matrix decomposition technique with applications in recommender systems, where it's used to identify latent factors in user-item interactions, and in noise reduction and data compression.
- **Computer Vision**: Operations on images, such as resizing, rotating, and shearing, are all performed using linear transformations on matrices of pixel values.
- **Natural Language Processing (NLP)**: Linear algebra is fundamental to word embedding models, which represent words as vectors in a high-dimensional space. The relationships between words (e.g., similarity) can then be calculated using vector operations like dot products.

Ultimately, understanding linear algebra is crucial for more than just implementing algorithms; it allows you to understand the inner workings of models and build intuition about how they learn and process data.

Here's a playlist about how machine learning uses linear algebra to solve data problems. A friendly introduction to linear algebra for ML.