Deep Learning: Applied Math and Machine Learning Basics

Bryan Lailey

May 2024

1 Linear Algebra

Linear algebra is a branch of continuous mathematics concerning linear equations of the type

$$a_1x_1 + ... + a_nx_n + b = 0. \tag{1}$$

Linear algebra is used extensively throughout science and engineering and is essential to understand the workings of many machine learning algorithms, particularly deep learning algorithms.

1.1 Scalars, Vectors, Matrices, and Tensors

Linear algebra involves several important types of mathematical objects:

- **Scalars:** A scalar is a single number. The convention for the document will be to represent scalars in lowercase italics and specify what kind of number they are. For example, we might say that $\{v \in \Re \mid 0 \leq v \leq c\}$ represents the speed of a particle.
- **Vectors:** A vector is an ordered array of numbers where we use an index to identify each individual number. The convention in this document will be to represent vectors in lowercase bold whose elements are written in lowercase italics with a subscript. Again, we need to specify what kind of numbers the elements of a vector are. For example, we might say that $\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$ represents the velocity of a particle where the vector elements v_1, v_2, and v_3 represent the x,y, and z components of the velocity, respectively.

  If each element of a vector is in $\Re$, and the vector has n elements, then the vector lies in $\Re^n$, the set formed by taking the Cartesian product of $\Re$ n times.

  We can define a set containing only some of the indices of a vector and write that set as a subscript of the vector. Supposing we wanted to include

only the x and y components of the velocity vector $\mathbf{v}$, we could create the set $S = \{1, 2\}$ and denote it $\mathbf{v_S}$. The complement of $\mathbf{v_S}$, $\mathbf{v_{-S}}$ contains all of the elements of $\mathbf{v}$ not included in S (i.e., v_3).

- **Matrices:** A matrix is a two dimensional array of numbers which requires that each element is identified by two indices rather than just one. The convention in this document is to represent matrices in uppercase bold whose elements are written in uppercase italics with two subscripts. If a matrix has m rows and n columns and each element of the matrix is in $\Re$, then we say that the matrix is in $\Re^{m \times n}$

  For example, we might say that $\mathbf{S} = \begin{bmatrix} S_{11} & S_{12} & S_{13} \\ S_{21} & S_{22} & S_{23} \\ S_{31} & S_{32} & S_{33} \end{bmatrix}$ represents the stress (force per unit area) on a cube of material that is being subjected to an arbitrary load. Here, the diagonal terms S_{11}, S_{22}, and S_{33} represent the stress applied to the cube from the positive x, y, and z directions respectively and the off-diagonal terms describe the shear. The pressure can also be calculated in a straight-forward way by dividing the trace of $\mathbf{S}$ by 3.

- **Tensors:** In general, we can describe an array that requires n indices to describe as a rank-n tensor. A scalar is a rank-0 tensor, a vector is a rank-1 tensor, and a matrix is a rank-2 tensor. Sometimes, especially in deep learning applications, we may need higher order tensors.

An important operation we can perform on matrices is the transpose. We can think of a transpose as a reflection across the main diagonal of a matrix. For matrix $\mathbf{A}$, its transpose $\mathbf{A^T}$ is defined such that

$$A^T_{i,j} = A_{j,i}. \tag{2}$$

Important special cases are that a scalar is its own transpose and the transpose of a row vector is a column vector (and vice-versa).

Matrix addition is performed on matrices of the same shape by adding their corresponding elements

$$\mathbf{C} = \mathbf{A} + \mathbf{B} = \sum_{i=1}^{m} \sum_{j=1}^{n} A_{i,j} + B_{i,j} \tag{3}$$

We can also add a scalar to a matrix or multiply a scalar by a matrix simply by performing that operation on each element of the matrix.

There is also a special convention in deep learning that is worth noting whereby we allow for the addition of a matrix and a vector,

$$\mathbf{C} = \mathbf{A} + \mathbf{b}, \tag{4}$$

by adding the vector $\mathbf{b}$ to each row of the matrix. The purpose of this shorthand, called broadcasting, is to implicitly copy $\mathbf{b}$ into many locations thus avoiding the need to create an entirely new matrix for that purpose.

1.2 Multiplying Matrices and Vectors

The matrix product of two matrices $\mathbf{A}$ and $\mathbf{B}$ is a third matrix, $\mathbf{C}$. For $\mathbf{C}$ to be defined, $\mathbf{A}$ must have the same number of columns as $\mathbf{B}$ has rows. If $\mathbf{A}$ is an $m \times n$ matrix and $\mathbf{B}$ is an $n \times p$ matrix, then $\mathbf{C}$ is an $m \times p$ matrix. We can define matrix multiplication element-wise as

$$C_{i,j} = \sum_k A_{i,k} B_{k,j}. \tag{5}$$

A less used matrix multiplication operation in linear algebra is the Hadamard product, which is simply the element-wise product of two matrices of the same shape

$$\mathbf{A} \odot \mathbf{B} = A_{i,j} B_{i,j} \tag{6}$$

The dot product between two vectors $\mathbf{x}$ and $\mathbf{y}$ of the same dimensionality is the matrix product $\mathbf{x}^T\mathbf{y}$. Equivalently, the matrix product $\mathbf{C} = \mathbf{AB}$ can be thought of as computing the dor product between the i^{th} row of $\mathbf{A}$ and the j^{th} column of $\mathbf{B}$.

$$Give examples of matrix product, Hadamard product, and dot product. \tag{7}$$

Matrix multiplication is both distributive

$$\mathbf{A}(\mathbf{B} + \mathbf{C}) = \mathbf{AB} + \mathbf{AC} \tag{8}$$

and associative

$$\mathbf{A}(\mathbf{BC}) = (\mathbf{AB})\mathbf{C} \tag{9}$$

but, unlike scalar multiplication, is not always commutative. The dot product between two vectors, however, is commutative,

$$\mathbf{x}^T\mathbf{y} = \mathbf{y}^T\mathbf{x}, \tag{10}$$

because a dot product is a scalar and a scalar is its own transpose which implies that

$$\mathbf{x}^T\mathbf{y} = \left(\mathbf{x}^T\mathbf{y}\right)^T = \mathbf{y}^T\mathbf{x}. \tag{11}$$

The system of linear equations

$$\mathbf{Ax} = \mathbf{b} \tag{12}$$

where $\mathbf{A} \in \Re^{m \times n}$, $\mathbf{x} \in \Re^n$, and $\mathbf{b} \in \Re^m$ describes a system of linear equations where $\mathbf{A}$ is a known matrix, $\mathbf{b}$ is a known vector, and $\mathbf{x}$ is a vector of unknown elements that we would like to solve for. We can write this system of linear equations out explicitly as

$$A_{1,1}x_1 + A_{1,2}x_2 + ... + A_{1,n}x_n = b_1 \tag{13}$$

$$... \tag{14}$$

$$A_{m,1}x_1 + A_{m,2}x_2 + ... + A_{m,n}x_n = b_m. \tag{15}$$

1.3 Identity and Inverse Matrices

Linear algebra provides a powerful tool for analytically solving Equation 12 called matrix inversion. To properly describe matrix inversion, we first need to introduce the concept of the identity matrix $\mathbf{I}_n$. The identity matrix is the matrix that preserves any vector when multiplied by that vector, i.e.,

$$\forall \mathbf{x} \in \Re^n, \mathbf{I}_n \mathbf{x} = \mathbf{x}. \tag{16}$$

The identity matrix has a simple structure, all off-diagonal elements are zero and all diagonal elements are one. For example, $\mathbf{I}_3$ is defined as

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{17}$$

The matrix inverse is defined such that a matrix multiplied by its inverse equals the identity matrix

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}_n. \tag{18}$$

We can use the concept of the matrix inverse to analytically solve Equation 12 via multiplying both sides by the inverse of $\mathbf{A}$:

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{19}$$

$$\mathbf{A}^{-1}\mathbf{A}\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} \tag{20}$$

$$\mathbf{I}_n\mathbf{x} = \mathbf{A}^{-1}\mathbf{b} \tag{21}$$

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}. \tag{22}$$

In principle, this process depends on there being an analytical solution for $\mathbf{A}^{-1}$, the conditions for which can be found in the following section. In practice, this method is primarily a theoretical tool and is rarely used for software applications because of inaccuracies introduced by the limited precision of digital computers.

1.4 Linear Dependence and Span