

Linear Algebra for Machine Learning

Essential operations

Multiplication

Multiplying $\mathbf{A} \in \mathbb{R}^{m,n}$ by $\mathbf{B} \in \mathbb{R}^{n,p}$ produces $\mathbf{C} \in \mathbb{R}^{m,p}$.

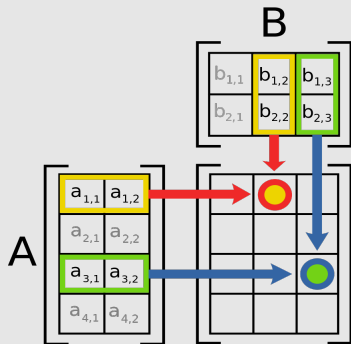
$$\mathbf{AB} = \mathbf{C}. \quad (3)$$

To compute \mathbf{C} the elements in the rows of \mathbf{A} are multiplied with the column elements of \mathbf{B} and the products added,

$$c_{ik} = \sum_{j=1}^n a_{ij} \cdot b_{jk}. \quad (4)$$

Define on the board:

- Dot product $\mathbf{a} \cdot \mathbf{b} = a_1 b_1 + a_2 b_2 + \cdots + a_n b_n$ for two vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$.
- Row times column view [Str+09]:



Linear Algebra for Machine Learning

└ Essential operations

└ The identity matrix

$$I = \begin{pmatrix} 1 & & \\ & 1 & \\ & & \ddots \\ & & & 1 \end{pmatrix} \quad (5)$$

Demonstrate multiplication with the inverse by hand.

$$\begin{pmatrix} -1 & 0 & 0 \\ 1 & -1 & 1 \\ 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} -1 & 0 & 0 \\ -2 & -1 & -1 \\ -1 & 0 & -1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (6)$$

Linear Algebra for Machine Learning

└ Essential operations

└ Matrix inverse

The inverse Matrix \mathbf{A}^{-1} undoes the effects of \mathbf{A} , or in mathematical notation,

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$$

(7)

The process of computing the inverse is called Gaussian elimination.

Example on the board:

$$\mathbf{A} = \begin{pmatrix} 2 & 0 \\ 1 & 3 \end{pmatrix} \rightsquigarrow \left(\begin{array}{cc|cc} 2 & 0 & 1 & 0 \\ 1 & 3 & 0 & 1 \end{array} \right) \rightsquigarrow \left(\begin{array}{cc|cc} 1 & 0 & \frac{1}{2} & 0 \\ 1 & 3 & 0 & 1 \end{array} \right) \quad (8)$$

$$\rightsquigarrow \left(\begin{array}{cc|cc} 1 & 0 & \frac{1}{2} & 0 \\ 0 & 3 & -\frac{1}{2} & 1 \end{array} \right) \rightsquigarrow \left(\begin{array}{cc|cc} 1 & 0 & \frac{1}{2} & 0 \\ 0 & 1 & -\frac{1}{6} & \frac{1}{3} \end{array} \right) \quad (9)$$

Test the result:

$$\begin{pmatrix} 2 & 0 \\ 1 & 3 \end{pmatrix} \begin{pmatrix} \frac{1}{2} & 0 \\ -\frac{1}{6} & \frac{1}{3} \end{pmatrix} = \begin{pmatrix} 2 \cdot \frac{1}{2} + 0 \cdot -\frac{1}{6} & 2 \cdot 0 + 0 \cdot \frac{1}{3} \\ 1 \cdot \frac{1}{2} + 3 \cdot -\frac{1}{6} & 0 \cdot 0 + 3 \cdot \frac{1}{3} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad (10)$$

Linear Algebra for Machine Learning

Essential operations

Computing determinants in two or three dimensions

The two-dimensional case:

$$\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11} \cdot a_{22} - a_{12} \cdot a_{21} \quad (12)$$

(13)

Computing the determinant of a three-dimensional matrix:

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11} \cdot \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12} \cdot \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13} \cdot \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} \quad (14)$$

Works for any row or column, as long as we respect the sign pattern.

Example computation on the board:

$$\begin{vmatrix} -1 & 0 & 0 \\ 1 & -1 & 1 \\ 1 & 0 & -1 \end{vmatrix} = -1 \cdot \begin{vmatrix} -1 & 1 \\ 0 & -1 \end{vmatrix} - 1 \cdot \begin{vmatrix} 0 & 0 \\ 0 & -1 \end{vmatrix} + 1 \cdot \begin{vmatrix} 0 & 0 \\ -1 & 1 \end{vmatrix} \quad (15)$$

$$= (-1) \cdot ((-1) \cdot (-1) - 0 \cdot 1)) - \quad (16)$$

$$(0 \cdot (-1) - 0 \cdot 0) + 0 \cdot 1 - (-1) \cdot 0 \quad (17)$$

$$= -1 \quad (18)$$

Linear Algebra for Machine Learning

└ Essential operations

└ Determinants in n-dimensions

$$\begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix} = a_{11} \begin{vmatrix} a_{22} & \dots & a_{2n} \\ \vdots & \ddots & \vdots \\ a_{n2} & \dots & a_{nn} \end{vmatrix} + a_{12} \begin{vmatrix} a_{21} & \dots & a_{2n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{vmatrix} + \dots + a_{1n} \begin{vmatrix} a_{21} & \dots & a_{2n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{vmatrix}$$

Draw the sign pattern on the board:

$$\begin{vmatrix} + & - & + & \dots \\ - & + & - & \dots \\ + & - & + & \dots \\ \vdots & \vdots & \vdots & \ddots \end{vmatrix}$$

(19)

The determinant can be expanded along any column as long as the sign pattern is respected.

Linear Algebra for Machine Learning

└ Linear curve fitting

└ The Pseudoinverse [Str+09; DFO20]

The inverse exists for square or n by n matrices. Nonsquare \mathbf{A} such as the one we just saw, require the pseudoinverse,

$$\mathbf{A}^\dagger = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T. \quad (21)$$

Sometimes solving $\mathbf{Ax} + \mathbf{b} = 0$ is impossible, the pseudoinverse considers,

$$\min_x \frac{1}{2} |\mathbf{Ax} - \mathbf{b}|^2 \quad (22)$$

(23)

instead. $\mathbf{A}^\dagger \mathbf{b} = \mathbf{x}$ yields the solution.

Sometimes solving $\mathbf{Ax} + \mathbf{b} = 0$ is impossible. One the board, derive:

$$\min_x \frac{1}{2} |\mathbf{Ax} - \mathbf{b}|^2 \quad (24)$$

$$(25)$$

At the optimum we expect,

$$0 = \nabla_x \frac{1}{2} |\mathbf{Ax} - \mathbf{b}|^2 \quad (26)$$

$$= \nabla_x \frac{1}{2} (\mathbf{Ax} - \mathbf{b})^T (\mathbf{Ax} - \mathbf{b}) \quad (27)$$

$$= \mathbf{A}^T (\mathbf{Ax} - \mathbf{b}) \quad (28)$$

$$= \mathbf{A}^T \mathbf{Ax} - \mathbf{A}^T \mathbf{b} \quad (29)$$

$$\mathbf{A}^T \mathbf{b} = \mathbf{A}^T \mathbf{Ax} \quad (30)$$

$$(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} = \mathbf{x} \quad (31)$$

Linear Algebra for Machine Learning

└ Regularization

└ Eigenvalues and Eigenvectors

Eigenvectors turn multiplication with a matrix into multiplication with a number,

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}. \quad (36)$$

Subtracting $\lambda\mathbf{x}$ leads to,

$$(\mathbf{A} - \lambda\mathbf{I})\mathbf{x} = \mathbf{0} \quad (37)$$

The interesting solutions are those where $\mathbf{x} \neq \mathbf{0}$, which means

$$\det(\mathbf{A} - \lambda\mathbf{I}) = 0 \quad (38)$$

On the board, compute the eigenvalues and vectors for the initial example.

$$\mathbf{A} = \begin{pmatrix} 1 & 4 \\ 0 & 2 \end{pmatrix} \rightarrow \begin{vmatrix} 1 - \lambda & 4 \\ 0 & 2 - \lambda \end{vmatrix} = (1 - \lambda) * (2 - \lambda) - 0 * 4 = 0 \quad (39)$$

$$\rightarrow \lambda_1 = 1, \lambda_2 = 2. \quad (40)$$

$$\begin{pmatrix} 1 - 1 & 4 \\ 0 & 2 - 1 \end{pmatrix} = \begin{pmatrix} 0 & 4 \\ 0 & 1 \end{pmatrix} \mathbf{x}_1 = \mathbf{0} \rightarrow \mathbf{x}_1 = \begin{pmatrix} p \\ 0 \end{pmatrix} \text{ for } p \in \mathbb{R} \quad (41)$$

$$\begin{pmatrix} 1 - 2 & 4 \\ 0 & 2 - 2 \end{pmatrix} = \begin{pmatrix} -1 & 4 \\ 0 & 0 \end{pmatrix} \mathbf{x}_1 = \mathbf{0} \rightarrow \mathbf{x}_2 = \begin{pmatrix} q \\ \frac{1}{4}q \end{pmatrix} \text{ for } q \in \mathbb{R} \quad (42)$$

Determinant not useful numerically, software packages use QR-Method.