

ベクトルや行列とその演算

Python 中級

鈴木 敬彦

本稿では、データとしてのベクトルや行列とこれらの演算について説明し、ベクトルや行列の幾何学的な空間での意味合いや、物理学的あるいは数学的な意味合い等については議論しません。

スカラー、ベクトル、行列、テンソル

スカラー (scalar) は単一の実数のみで構成されます。ベクトル (vector) は複数のスカラー値から構成される 1 次元配列です。行列 (matrix) は複数のスカラー値から構成される 2 次元配列です。テンソル (tensor) は複数のスカラー値から構成される 3 次元以上の配列です。

テンソルの次元数は階数と呼ばれ、3 次元の配列であれば 3 階のテンソル、11 次元の配列であれば 11 階のテンソルと呼びます。行列は 2 階のテンソル、ベクトルは 1 階のテンソル、スカラーは 0 階のテンソルでもあります。ベクトルや行列を表記する時には、各要素は空白やカンマで区切られ、全体を括弧（丸括弧や角括弧）で括ります。

スカラーの例: 3, 3.1415926

ベクトルの例: [0, 1], (x, y, z)

行列の例: $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$

テンソルについては、上記のような一般的な表記法はありません。

ベクトルの要素数は、一般的にベクトルが空間ベクトルであることが多いため、次元と称されることもあります。行列の形状は、その行列の行数 m と列数 n で $m \times n$ 行列と表されます。上記の行列の例は、どちらも 2×2 行列です。また、行列演算の中でのベクトルは、 $1 \times n$ 行列あるいは $m \times 1$ 行列として扱われ、上記のベクトルの例は、それぞれ 1×2 行列と 1×3 行列となります。 $1 \times n$ 行列は行ベクトル、 $m \times 1$ 行列を列ベクトルと呼びます。

行ベクトル

$$\mathbf{v}_{(1 \times n)} = (v_1, v_2, v_3, \dots, v_n)$$

列ベクトル

$$\mathbf{v}_{(m \times 1)} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_m \end{pmatrix}$$

行列

$$\mathbf{M}_{(m \times n)} = \begin{pmatrix} m_{(1,1)} & m_{(1,2)} & \cdots & m_{(1,n)} \\ m_{(2,1)} & m_{(2,2)} & \cdots & m_{(2,n)} \\ \vdots & \vdots & \ddots & \vdots \\ m_{(m,1)} & m_{(m,2)} & \cdots & m_{(m,n)} \end{pmatrix}$$

ベクトル、行列の演算

スカラー同士の演算については既知のものとしてここでは説明しません。また、テンソル同士の演算についても、基本的なニューラルネットワークの構築には不要であるため、説明しません。テンソルの演算について学びたい方は、アインシュタインの「総和規約」を理解しておく事を薦めます。

ベクトルの演算

ベクトルとスカラー

加算、減算はありません。除算は乗算の変形に過ぎず、乗算はベクトルをスカラー倍します。

$$k\mathbf{v} = k(v_1, v_2, v_3, \dots, v_n) = (kv_1, kv_2, kv_3, \dots, kv_n)$$

ベクトルとベクトル

加算、減算

加算、減算は、それぞれのベクトルの成分同士の演算で、それぞれのベクトルの要素数が同じ時だけ成立します。

$$\begin{aligned}\mathbf{u} + \mathbf{v} &= (u_1, u_2, u_3, \dots, u_n) + (v_1, v_2, v_3, \dots, v_n) \\ &= (u_1 + v_1, u_2 + v_2, u_3 + v_3, \dots, u_n + v_n)\end{aligned}$$

乗算

ベクトル同士の乗算（積）にはドット積 (dot product) とクロス積 (cross product) の2種類があり、除算は定義されません。また、加算や減算と同様に対応する成分同士の乗算も存在しますが、これは一般的な行列演算の一つとして定義されているため、ここではドット積のみを説明します。

ドット積は、それぞれのベクトルの要素数が同じ時だけ成立し、スカラー値を返します。演算子の記号には「 \cdot 」が使用されます。 n 次元実ユークリッド空間 \mathbb{R}^n で定義される内積と同じものです。

$$\mathbf{u} \cdot \mathbf{v} = u_1v_1 + u_2v_2 + u_3v_3 + \dots + u_nv_n = \sum_{i=1}^n u_i v_i$$

また、ベクトルの自身とのドット積の平方根はノルム (**norm**) と呼ばれ、ベクトルの「大きさ」を表します。

$$\|\mathbf{v}\| = \sqrt{\mathbf{v} \cdot \mathbf{v}}$$

行列の演算

ここでは、行列の演算について説明する前に、行列の基本的な事柄を紹介し、演算についても基本的なニューラルネットワークの理解に必要な範囲に留めます。

前述した通り、行列の形状はその行列の行数 m と列数 n で $m \times n$ 行列と表されます。特に $m = n$ の行列は正方行列と呼ばれます。行列の形状がわざわざ式中に記述されることは多くはありませんが、記述する場合は \times 記号等を用いて誤解されることのないように気を付けてください。

行列の i 行目 j 列目の成分は、行列の (i, j) 成分と呼ばれ、 a_{ij} のように表されることが一般的です。添字の表記については、誤解を避けるために i と j をカンマで区切ったり、更に括弧で括弧することもあります。行列 A の (i, j) 成分という意味で A_{ij} と書かれることもあります。

また、明確な規則はありませんが、数式を見やすくするために、数式中のベクトルや行列を表す文字を太字や大文字で記述する事は珍しくありません。

単位行列 (identity matrix)

単位行列と呼ばれる行列は、行列演算において最も特別な行列と言えます。単位行列は、対角成分（行列中の $i = j$ となる成分）が 1 で、それ以外は 0 である正方行列です。

$$a_{ij} = \begin{cases} 1 & (i = j) \\ 0 & (i \neq j) \end{cases}$$
$$I = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}, \quad I_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

行列の乗算（後述）において、単位行列はその単位元です。単位行列は、一般に I または E で記述され、また、形状を記述する場合には正方行列である事が自明のため I_3 のように記述されることも多くあります。

転置 (transpose)

行列の行と列を入れ替える操作を「転置」と呼び、転置された行列を転置行列と呼びます。とある行列 A が以下のような行列であれば、

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

その転置行列は、以下のようになります。

$$\mathbf{A}^T = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}^T = \begin{pmatrix} a_{11} & a_{21} & \cdots & a_{n1} \\ a_{12} & a_{22} & \cdots & a_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1m} & a_{2m} & \cdots & a_{nm} \end{pmatrix}$$

転置の記述は様々で、以下のように記述されることが多いです。

$${}^t\mathbf{A}, \mathbf{A}^t, \mathbf{A}^{tr}, \mathbf{A}^T, \mathbf{A}^\dagger, \mathbf{A}'$$

ただし、 ${}^{tr}\mathbf{A}$ は行列のトレース $\text{tr}(\mathbf{A})$ と紛らわしいため避けるべきです。

加算、減算

行列の加算、減算は、ベクトルの場合と同じで、成分同士の演算になります。双方の行列の形状が同じ時だけ成立します。

$$\begin{aligned} \mathbf{A} &= \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & \ddots & \vdots \\ b_{m1} & \cdots & b_{mn} \end{pmatrix} \\ \mathbf{A} + \mathbf{B} &= \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} + \begin{pmatrix} b_{11} & \cdots & b_{1n} \\ \vdots & \ddots & \vdots \\ b_{m1} & \cdots & b_{mn} \end{pmatrix} \\ &= \begin{pmatrix} a_{11} + b_{11} & \cdots & a_{1n} + b_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} + b_{m1} & \cdots & a_{mn} + b_{mn} \end{pmatrix} \end{aligned}$$

乗算

行列の乗算（積）には、基本的な演算の他に多岐に渡る積が定義されています。ここでは基本的な演算と、基本的なニューラルネットワークの理解に必須と思われるものだけを紹介します。

行列とスカラーの乗算

ベクトルをスカラー倍した時と同様に、行列をスカラー倍します。

$$k\mathbf{A} = k \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix} = \begin{pmatrix} ka_{11} & \cdots & ka_{1n} \\ \vdots & \ddots & \vdots \\ ka_{m1} & \cdots & ka_{mn} \end{pmatrix}$$

行列と行列の乗算

最も一般的な行列と行列の乗算で、行列の積と言ったほうが馴染みがあるかもしれません。演算子の記号は無く、単に \mathbf{AB} のように記述します。 \mathbf{A} が $m \times n$ 行列、 \mathbf{B} が $p \times q$ 行列の時、 $n = p$ でなければなりません。乗算は、左側の行列の行成分と右側の行列の列成分を対応する成分同士で乗じてその和を取ります。積は $m \times q$ 行列となります。

$$\mathbf{AB} = \underbrace{\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix}}_{2 \times 3} \underbrace{\begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{pmatrix}}_{3 \times 2} = \underbrace{\begin{pmatrix} a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} & a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32} \\ a_{21}b_{11} + a_{22}b_{21} + a_{23}b_{31} & a_{21}b_{12} + a_{22}b_{22} + a_{23}b_{32} \end{pmatrix}}_{2 \times 2}$$

スカラー同士の乗算のような交換法則は成立しないことに注意してください。左右を交換する場合は、転置します。

$$\mathbf{AB} \neq \mathbf{BA}$$

$$(\mathbf{AB})^T = \mathbf{B}^T \mathbf{A}^T, \quad \mathbf{AB} = (\mathbf{B}^T \mathbf{A}^T)^T$$

但し、どちらか一方の行列が前述の単位行列である場合のみ、例外的に交換法則は成立します。

$$\mathbf{AI} = \mathbf{IA} = \mathbf{A}$$

また、行列の積が $\mathbf{AB} = \mathbf{I}$ となるような行列 \mathbf{B} が存在する時、 \mathbf{A} は正則行列 (regular matrix) で「正則である」と言い、 \mathbf{B} を \mathbf{A} の逆行列 (inverse matrix) と呼び、 \mathbf{A}^{-1} と記述します。-1 乗ではないので注意してください。ただし、一般に \mathbf{A}^n と冪乗のように記述されているものは、 \mathbf{A} に \mathbf{A} を n 回乗ずることを意味し、 \mathbf{A}^3 であれば \mathbf{AAA} 意味し、通常の冪乗の概念と大きく変わりありません。但し、行列の冪は正方行列以外では成立しません。また、冪に関しては、成分毎に冪を計算することもあるので注意してください。

また、行列の乗算では以下の法則が成立します。覚えておくと便利なこともあるでしょう。

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

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

アダマール積 (Hadamard product/element-wise product)

行列の加算と同じように、行列同士を成分毎に乘じます。二つの行列の形状は同じでなくてはなりません。演算子の記号には「 \circ 」や「 \odot 」が用いられます。

$$\mathbf{A} \circ \mathbf{B} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \circ \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} = \begin{pmatrix} a_{11}b_{11} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{pmatrix}$$

クロネッカー積 (Kronecker product)

任意の形状の行列同士の演算で、演算子左側の行列の各成分と右側の行列を乗じます。 $m \times n$ 行列と $p \times q$ 行列のクロネッカー積は $mp \times nq$ 行列になります。演算子の記号には「 \otimes 」が用いられます。

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} \end{pmatrix} = \begin{pmatrix} a_{11}b_{11} & a_{11}b_{12} & a_{12}b_{11} & a_{12}b_{12} \\ a_{11}b_{21} & a_{11}b_{22} & a_{12}b_{21} & a_{12}b_{22} \\ a_{21}b_{11} & a_{21}b_{12} & a_{22}b_{11} & a_{22}b_{12} \\ a_{21}b_{21} & a_{21}b_{22} & a_{22}b_{21} & a_{22}b_{22} \end{pmatrix}$$