

Deep Learning 輪読会 2017

第2章 線形代数

2017.11.06

東京大学 松尾研究室

リサーチエンジニア 曾根岡侑也

2章 線形代数

2.1 スカラー, ベクトル, 行列, テンソル

2.2 行列とベクトルの乗算

2.3 単位行列と逆行列

2.4 線形従属とスパン

2.5 ノルム

2.6 特殊な行列とベクトル

2.7 固有値分解

2.8 特異値分解

2.9 ムーア・ペンローズ擬似逆行列

2.10 トレース演算子

2.11 行列式

2.12 例：主成分分析

はじめに

- 前章までの議論：DeepLearningの立ち位置とその歴史
 - 機械学習・表現学習などから見たDeepLearningの立ち位置
 - DeepLearningが注目されるまでの歴史ときっかけ
- 本章の議論：深層学習の理解に必要な線形代数
 - 行列の基礎、演算、ノルム
 - 固有値分解・特異値分解
 - 行列式・トレース演算子
 - 基礎知識を用いた主成分分析の導出

2.1 スカラ, ベクトル, 行列, テンソル

- **スカラ：単一の数**（表記：イタリックの小文字 s ）
 - 定義するときにはどんな種類なのか（実数, 自然数等）を明示する
- **ベクトル：数値の配列**（表記：イタリックの太字の小文字 \mathbf{x} ）
 - 順番に対応したインデックスを用いて個々の要素を表す $x_1 \ x_2$
 - 格納できる数値の種類も明示する R^n
 - ベクトル自体は右のような列で書き表す
 - ベクトルは空間上の点を表しているとみなせる

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} .$$

2.1 スカラ, ベクトル, 行列, テンソル

- **行列：2次元の数値配列**（表記：イタリックの太字の大文字 ***A***）

- 格納できる数値の種類と高さや幅も明示する $A \in R^{m \times n}$
- 行と列のインデックスで要素を表す $A_{m,n}$
- 要素を明示的に書くときは右のようになる $\begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix}$

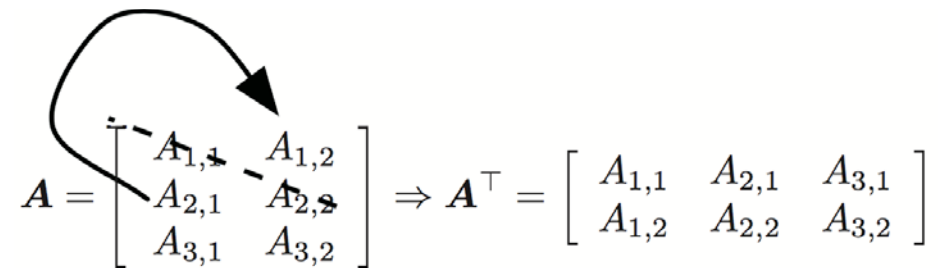
- **テンソル：3つ以上軸を持つ配列**（表記：太字の大文字 ***A***）

- 同様にインデックスで要素を表す $A_{m,n,l}$
- 軸の数が可変

2.1 行列の演算

- 転置：対角線上で鏡像にあたる行列

$$(\mathbf{A}^\top)_{i,j} = A_{j,i}.$$


$$\mathbf{A} = \begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \\ A_{3,1} & A_{3,2} \end{bmatrix} \Rightarrow \mathbf{A}^\top = \begin{bmatrix} A_{1,1} & A_{2,1} & A_{3,1} \\ A_{1,2} & A_{2,2} & A_{3,2} \end{bmatrix}$$

- 可算：要素ごとの足し算

- 同じ形の行列は要素同士の足し合わせができる

$$\mathbf{C} = \mathbf{A} + \mathbf{B} \quad \text{ただし、} C_{i,j} = A_{i,j} + B_{i,j}$$

- ブロードキャスト：行列とベクトルの加算方法

- 行列とベクトルの加算が可能で、 \mathbf{b} を各行に追加する

$$\mathbf{C} = \mathbf{A} + \mathbf{b}, \quad \text{ただし } C_{i,j} = A_{i,j} + b_j.$$

2.2 行列とベクトル演算

- 行列の積：2つの行列の掛け算

- Aの列数とBの行数が同じ必要がある

$$C = AB. \quad C_{i,j} = \sum_k A_{i,k} B_{k,j}.$$

- 線形方程式を下記のように書き表すことができる

$$Ax = b$$

- アダマール積：要素ごとの積

- 同じ形をしている必要がある

$$A \odot B$$

2.2 行列の積の法則

- 分配法則：満たしている

$$A(B + C) = AB + AC.$$

- 結合法則：満たしている

$$A(BC) = (AB)C.$$

- 交換法則：満たさない

$$AB = BA \text{ は必ず成立するわけではない}$$

2.3 単位行列と逆行列

- 単位行列：対角線の要素が1でそれ以外は全て0となっている行列
 - 任意のベクトルに掛け合わせても変化させないような行列 \mathbf{I}_n

$$\forall \mathbf{x} \in \mathbb{R}^n, \mathbf{I}_n \mathbf{x} = \mathbf{x}. \quad (\text{例}) \quad \mathbf{I}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- 逆行列：ある行列に掛け合わせると単位行列が得られる行列
 - 必ず存在するわけではないが、存在する場合求める計算方法がある

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

- これを用いることで線形方程式を解くことができる

$$\mathbf{A} \mathbf{x} = \mathbf{b} \quad \Rightarrow \quad \mathbf{x} = \mathbf{A}^{-1} \mathbf{b}.$$

2.4 線形従属とスパン

- **Aの逆行列を持つためには**

- どんな**b**に対しても解が1つだけである
- 連立方程式の解がない or 無限にある では×

- **線形結合：ベクトルの線形和**

- **Ax** は 下記のように**A**の列ベクトルをスカラー倍して足し合わせ

$$Ax = \sum_i x_i A_{:,i}.$$

- **スパン（列空間）**：行列の列ベクトルの線形結合でカバーされている空間

- **b** $\in \mathbb{R}^m$ の全てで解を持つには、Aのスパンが \mathbb{R}^m になる必要がある

2.4 線形従属とスパン

- **線形独立**

- ベクトルの集合において、どのベクトルも他のベクトルの線形結合で作ることができない状態
- できる状態を**線形従属**といい、その正方行列を**特異**という

- **Aが逆行列を持つために必要十分条件**

- 正方行列
- 全ての列が線形独立なベクトルを持つこと

2.5 ノルム

- ノルム：ベクトルの大きさ L^p
 - ベクトルを非負の値に写像する関数で原点から点 x までの距離

$$\|\mathbf{x}\|_p = \left(\sum_i |x_i|^p \right)^{\frac{1}{p}}.$$

- ノルムの性質
 - $f(\mathbf{x}) = 0 \Rightarrow \mathbf{x} = \mathbf{0}$
 - $f(\mathbf{x} + \mathbf{y}) \leq f(\mathbf{x}) + f(\mathbf{y})$ (三角不等式 (triangle inequality))
 - $\forall \alpha \in \mathbb{R}, f(\alpha \mathbf{x}) = |\alpha| f(\mathbf{x})$

2.5 ノルム

- **L2ノルム**：ユークリッドノルム $\|\mathbf{x}\|$
 - 原点から \mathbf{x} までのユークリッド距離と同値
 - 計算的に扱いやすい二乗の値を使うこともよくある
- **L1ノルム** $\|\mathbf{x}\|_1 = \sum_i |x_i|$.
 - 機械学習ではゼロと非ゼロを区別することが重要なことがあり、
数学的な単純性を保持したL1ノルムも使うことが多い
- **最大値ノルム** $\|\mathbf{x}\|_\infty = \max_i |x_i|$.
 - 要素の最大値を取る

2.5 行列のノルム

- フロベニウスノルム

- 行列のサイズでよく使う、L2ノルムに近いノルム

$$\|A\|_F = \sqrt{\sum_{i,j} A_{i,j}^2}.$$

- ノルムによるベクトルの内積表現

- 下記で表すことができる

$$\boldsymbol{x}^\top \boldsymbol{y} = \|\boldsymbol{x}\|_2 \|\boldsymbol{y}\|_2 \cos \theta.$$

2.6 特殊な行列とベクトル

- 対角行列：対角線上以外は0な行列
 - 単位行列は全ての対角成分が1の対角行列
 - ベクトル v が対角成分のものを $\text{diag}(v)$ と書き、計算効率がいい
 - 正方行列である必要はないが、その場合逆行列を持たない
- 対称行列：転置行列が元の行列と同じもの $A = A^T$.
 - 2つの引数を取る関数によって作られるもので出てくる（ex：距離）
- 単位ベクトル：L1ノルムが1
- 直交：ベクトルの内積が0 $x^T y = 0$
 - 互いに90度の角度である
 - ベクトルが単位ベクトルのとき正規直交という

2.6 特殊な行列とベクトル

- 直交行列

- 正方行列で、その行も列も互いに正規直交となる行列

$$\mathbf{A}^{\top} \mathbf{A} = \mathbf{A} \mathbf{A}^{\top} = \mathbf{I}.$$

- 逆行列は転置行列になるため簡単に計算ができる

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

2.7 固有値分解

- 数学では対象を構成部分に分解することで扱いやすくすることが多い
- 固有ベクトル v : A をかけた時に縮尺のみ変化する非ゼロベクトル
- 固有値 λ : 固有ベクトルに対応するスカラー値

$$Av = \lambda v$$

- 固有ベクトルの実数値倍したのもも固有ベクトルになるので
単位ベクトルのみを使用する

2.7 固有値分解

- 固有値分解：行列を固有ベクトルと固有値の集合に分解

- n 個の線形独立な固有ベクトルを列ごとに連結した行列 $\mathbf{V} = [\mathbf{v}^{(1)}, \dots, \mathbf{v}^{(n)}]$
- 固有値を連結したベクトル $\boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_n]^\top$

$$\mathbf{A} = \mathbf{V} \text{diag}(\boldsymbol{\lambda}) \mathbf{V}^{-1}.$$

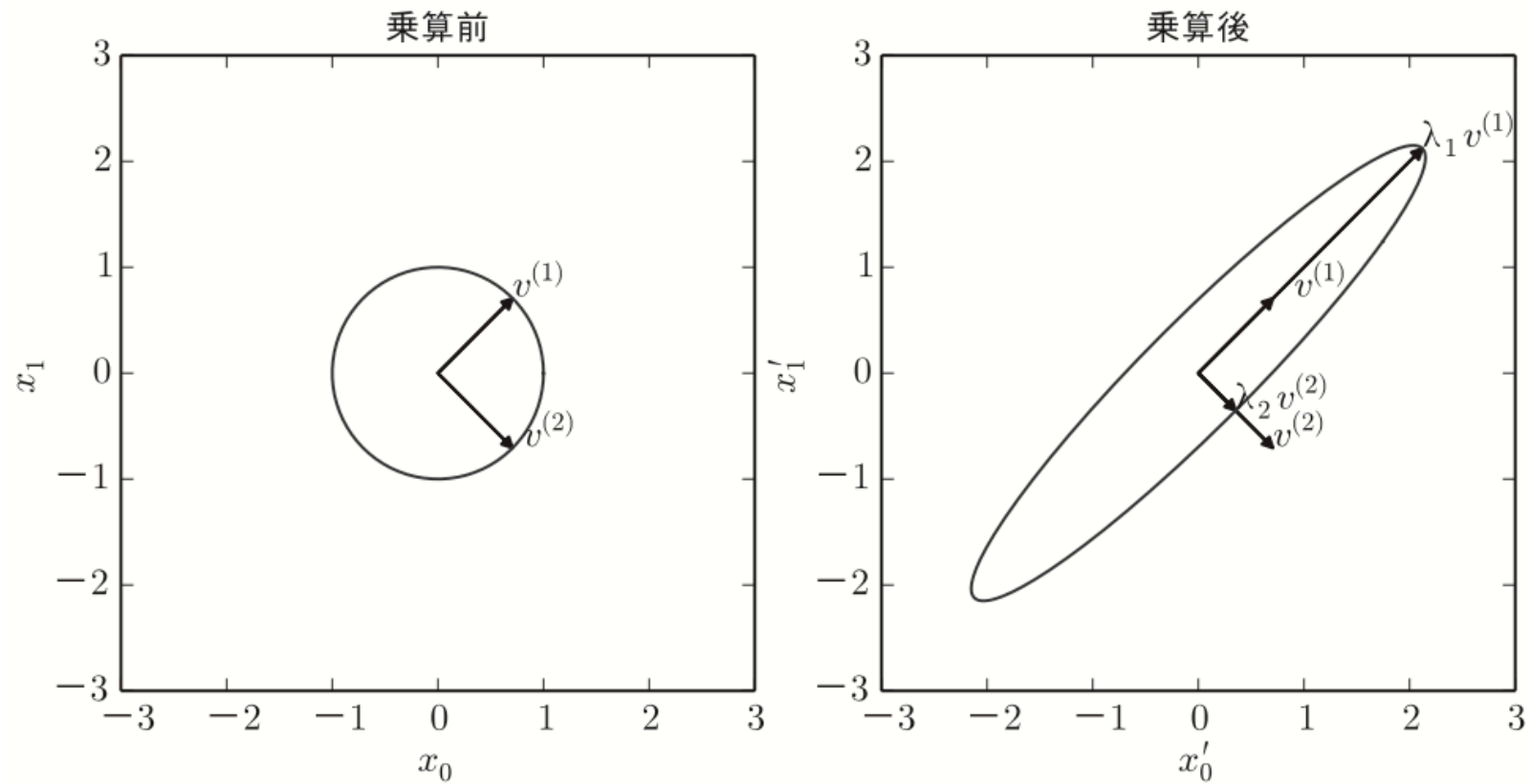
- 実対称行列：実数値だけからなる固有ベクトルと固有値で分解できる

- \mathbf{Q} ： \mathbf{A} の固有ベクトルからなる正交行列
- $\boldsymbol{\Lambda}$ ：対角行列

$$\mathbf{A} = \mathbf{Q} \boldsymbol{\Lambda} \mathbf{Q}^\top.$$

2.7 固有値分解

- 行列Aは固有ベクトル方向に固有値分だけ空間を縮小拡大だけ演算



2.7 固有値分解

- 固有値分解の活用方法
 - 固有値のいずれかがゼロ → 行列は特異
 - $\|x\|_2 = 1$ の場合、 $f(x) = x^T A x$ を最適化できる
- 正定値：固有値が全て正の行列
- 半正定値：固有値が正又はゼロの行列
 - $\forall x, x^T A x \geq 0$ が保証されている
- 負定値：固有値が全て負の行列
- 半負定値：固有値が負又はゼロの行列

2.8 特異値分解

- 特異値分解 (SVD) : 特異ベクトルと特異値に分解 $A = UDV^T$.
 - 実行列は全て特異値分解を持つが、固有値分解はそうでもない
 - A が $m \times n$ 行列のとき
 - U : $m \times m$ の行列で、各列は**左特異ベクトル**と呼ぶ
 - D : $m \times n$ の対角行列で、対角成分は**特異値**と呼ばれる
 - V : $n \times n$ の行列で、各列は**右特異ベクトル**と呼ぶ
 - U と V は直交行列
 - 左特異ベクトルは AA^T の固有ベクトルで、
右特異ベクトルは $A^T A$ の固有ベクトル

2.9 ムーア・ペンローズ擬似逆行列

- ムーア・ペンローズ擬似逆行列：

- 非正方行列において逆行列は一意に得られないことがある
- 下記の近似で擬似逆行列を得ることができる
 - 行 > 列の：解が存在しない可能性があるが、最も近いxが得られる
 - 列 > 行：解のうちの1つを得ることができる

$$\mathbf{A}^+ = \lim_{\alpha \searrow 0} (\mathbf{A}^\top \mathbf{A} + \alpha \mathbf{I})^{-1} \mathbf{A}^\top.$$

- 実際の計算は下記のアルゴリズムで行う
 - \mathbf{D}^+ ：特異値分解の対角行列 \mathbf{D} の擬似逆行列

(非ゼロ成分の逆数を求めた結果の行列を転置)

$$\mathbf{A}^+ = \mathbf{V} \mathbf{D}^+ \mathbf{U}^\top.$$

2.10 トレース演算子

- トレース演算子：対角成分の総和を得る演算子

$$\text{Tr}(\mathbf{A}) = \sum_i \mathbf{A}_{i,i}.$$

- トレース演算子の性質

- フロベニウスノルムで利用可能

$$\|\mathbf{A}\|_F = \sqrt{\text{Tr}(\mathbf{A}\mathbf{A}^\top)}.$$

- 転置行列で不変

$$\text{Tr}(\mathbf{A}) = \text{Tr}(\mathbf{A}^\top).$$

- 積の循環置換が可能（形式によらず）） $\text{Tr}(\prod_{i=1}^n \mathbf{F}^{(i)}) = \text{Tr}(\mathbf{F}^{(n)} \prod_{i=1}^{n-1} \mathbf{F}^{(i)}).$

- スカラーは不変

$$a = \text{Tr}(a)$$

2.11 行列式

- 行列式：行列を実数値のスカラーに写像する関数 $\det(\mathbf{A})$
 - 行列全ての固有値の積に等しい
 - 行列式の絶対値 = 行列の乗算で空間がどれだけ拡大縮小するかの尺度
 - 行列式 = 0：少なくとも1つの次元で完全に縮小
 - 行列式 = 1：体積が保存

2.12 例：主成分分析

- 有名な機械学習アルゴリズムの1つで線形代数のみで導出可能
- 目標：n次元のm個の点の集まりを出来る限り精度を保って不可逆圧縮を行う
 - $\mathbf{x}^{(i)} \in \mathbb{R}^n$ に対して、 $\mathbf{c}^{(i)} \in \mathbb{R}^l$ を見つける ($l < n$)
 - 符号化関数 $f(\mathbf{x}) = \mathbf{c}$ と復号化関数 $\mathbf{x} \approx g(f(\mathbf{x}))$ を作る
- PCA：復号化器を行列の乗算を用いて行う
 - 「Dの列が互いに直行する」「列が単位ノルムを持つ」条件でDを求める

$$g(\mathbf{c}) = \mathbf{D}\mathbf{c} \quad \text{ただし、} \quad \mathbf{D} \in \mathbb{R}^{n \times l}$$

2.12 例：主成分分析

- 符号化点 \mathbf{c} の求め方

- 復号化器で再構成したものと入力 \mathbf{x} のL2ノルムの二乗を最小化

$$\mathbf{c}^* = \arg \min_{\mathbf{c}} \|\mathbf{x} - g(\mathbf{c})\|_2^2.$$

$$= \arg \min_{\mathbf{c}} -2\mathbf{x}^\top g(\mathbf{c}) + g(\mathbf{c})^\top g(\mathbf{c}). \quad (\text{分配法則})$$

$$= \arg \min_{\mathbf{c}} -2\mathbf{x}^\top \mathbf{D}\mathbf{c} + \mathbf{c}^\top \mathbf{I}_l \mathbf{c} \quad (\text{置き換え})$$

$$= \arg \min_{\mathbf{c}} -2\mathbf{x}^\top \mathbf{D}\mathbf{c} + \mathbf{c}^\top \mathbf{c}. \quad (\mathbf{D} \text{ の直交性と単位ノルム})$$

- \mathbf{c} について微分して最小とする \mathbf{c} をもとめる

$$\nabla_{\mathbf{c}} (-2\mathbf{x}^\top \mathbf{D}\mathbf{c} + \mathbf{c}^\top \mathbf{c}) = \mathbf{0}$$

$$-2\mathbf{D}^\top \mathbf{x} + 2\mathbf{c} = \mathbf{0}$$

$$\mathbf{c} = \mathbf{D}^\top \mathbf{x}.$$

Dの転置が符号化の最適解

2.12 例：主成分分析

- 符号化関数と再構成演算の定義

$$f(\mathbf{x}) = \mathbf{D}^\top \mathbf{x}, \quad g(f(\mathbf{x})) = \mathbf{D}\mathbf{D}^\top \mathbf{x}.$$

- 符号化行列 \mathbf{D} の求め方
 - 復号化器で再構成したものと全入力 \mathbf{x} の誤差を最小化する

$$\mathbf{D}^* = \arg \min_{\mathbf{D}} \sqrt{\sum_{i,j} \left(x_j^{(i)} - r(\mathbf{x}^{(i)})_j \right)^2} \text{ subject to } \mathbf{D}^\top \mathbf{D} = \mathbf{I}_l.$$

2.12 例：主成分分析

- $l = 1$ の場合：第一主成分のみ再構成（ D はベクトル d とおける）

$$d^* = \arg \min_d \sum_i \|x^{(i)} - dd^\top x^{(i)}\|_2^2 \text{ subject to } \|d\|_2 = 1.$$

$$= \arg \min_d \|X - Xdd^\top\|_F^2 \text{ subject to } d^\top d = 1. \quad (\text{総和ではなく計画行列}X\text{に})$$

$$= \arg \min_d \text{Tr} \left((X - Xdd^\top)^\top (X - Xdd^\top) \right) \quad (\text{トレース演算})$$

$$= \arg \min_d -\text{Tr}(X^\top Xdd^\top) - \text{Tr}(dd^\top X^\top X) + \text{Tr}(dd^\top X^\top Xdd^\top)$$

$$= \arg \min_d -2 \text{Tr}(X^\top Xdd^\top) + \text{Tr}(dd^\top X^\top Xdd^\top) \quad (d\text{を含まないところの省略})$$

$$= \arg \min_d -2 \text{Tr}(X^\top Xdd^\top) + \text{Tr}(X^\top Xdd^\top) \text{ subject to } d^\top d = 1 \quad (\text{巡回と制約})$$

$$= \arg \max_d \text{Tr}(d^\top X^\top Xd) \text{ subject to } d^\top d = 1. \quad (\text{打ち消しと反転})$$

d の最適解は最大固有値に対応する XTX の固有ベクトル
一般解は最大固有値に対応する L 個の固有ベクトルで得られる

参考文献

- Deep Learning
 - Ian Goodfellow, Yoshua Bengio, Aaron Courville
 - 日本語版
 - <https://www.amazon.co.jp/%E6%B7%B1%E5%B1%A4%E5%AD%A6%E7%BF%92-lan-Goodfellow/dp/4048930621>