Python プログラミング入門資料 1

中田 昌輝

2021 年 2 月 7 日

本資料では解説を特に行わず, 演習問題を載せる．今の世の中としては書き方を知識として体得していることはあまり評価されない．というのもプログラム書き方やスタイル, 言語は数年でがらりとかわってしまうからである．それゆえ大事なのは書き方を覚えることではなく, どのコードを組み合わせるかということを身につけることが大事である．出題する問題をネットのリファレンスなどを参考にして解いてもらいたい．

Python プログラミングは主に機械学習の分野で応用されている．これは Python が非常に書きやすいからではなく, いろいろなライブラリがそろえられているからである．またプログラムをコンパイルして実行を行う静的型付け言語ではなく, 実行時に意味解釈を行う動的型付け言語であることから途中実行も可能となる．このことによりコンパイラによる途中エラー導出は得にくいが, 途中までの結果をすぐに見ることもできる．

機械学習でのプログラミングはそれほど難しくはない．これは前述のとおりライブラリが豊富であるからだ．しかし, 機械学習が難しいとされている理由は実装内部として数学をメインに扱っているからであろう．数学のレベルとしては線形代数と微分積分の基礎が理解できていることが前提である．

問題 1

(1) 分類分けをする問題ではしばしば行列計算を用いることがある．というのも特徴をベクトルで表現するからである．人を特徴で分類するとき,「性別」,「年齢」,「体重」,「出生地」, ... など多くの分類の仕方が存在する．これを分類すればするほど共通する人が見つかったり, 相反する性質をもつ人間が見つかったりする．この分類分けをベクトルで表現するが内積がベクトルのノルムの積とほぼ同じである場合, それは同じような人といえるであろう．このような分類を教師なし学習というが, 教師あり学習でもベクトルを用いることはある．今回はこのようなベクトル・行列の計算をスムーズに行うために行列の積をプログラムしてもらう．

入力を $n \times m$ 行列 A と $m \times n$ 行列 B としてのその積を求めたい．つまり,

$$
\begin{array}{lllll}
n & m & & & \\
a_{11} & a_{12} & a_{13} & \cdots & a_{1m} \\
a_{21} & a_{22} & a_{23} & \cdots & a_{2m} \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nm} \\
b_{11} & b_{12} & b_{13} & \cdots & b_{1n} \\
b_{21} & b_{22} & b_{23} & \cdots & b_{2n} \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
b_{m1} & b_{m2} & b_{m3} & \cdots & b_{mn}
\end{array}
$$

という入力が与えられたときに, 積を出力するプログラムを書け.
出力形式は

$$
\begin{array}{ccccc}
c_{11} & c_{12} & c_{13} & \cdots & c_{1n} \\
c_{21} & c_{22} & c_{23} & \cdots & c_{2n} \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
c_{n1} & c_{n2} & c_{n3} & \cdots & c_{nn}
\end{array}
$$

とする. 数字を出力したあと, 空白をいれること. 行が変わる場合は改行を行い, 出力結果の最後に改行をいれることを忘れないこと.
ちなみに

$$
\begin{pmatrix} 10 & 1 & 4 & 32 \\ 3 & -2 & 41 & 1 \\ 43 & 100 & 21 & 134 \end{pmatrix}
\begin{pmatrix} -1 & -1 & -10 \\ 4 & 6 & 9 \\ 2 & 2 & 1 \\ 6 & 2 & 7 \end{pmatrix}
=
\begin{pmatrix} 194 & 68 & 137 \\ 77 & 69 & 0 \\ 1203 & 867 & 1429 \end{pmatrix}
$$

であるので, 検算に用いること.

(2) 転置行列もしばしば求めることが要求される. 転置行列とは列成分と行成分を入れ替えたものになる. まず多く見るのはベクトルの内積であろう. これを $n \times 1$ の行列 $A,\ B$ が与えられたときに, $A^\top B$ を求めるものと同じである. ここではベクトルを行列として認識することで計算を一般化することも可能となる.
入力を $n \times m$ 行列 A が与えられたときに転置行列 $A^\top$ を出力するプログラムを書け.
入力形式は

$$
\begin{array}{ccccc}
n & m & & & \\
a_{11} & a_{12} & a_{13} & \cdots & a_{1m} \\
a_{21} & a_{22} & a_{23} & \cdots & a_{2m} \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nm}
\end{array}
$$

出力形式は

$$
\begin{array}{ccccc}
c_{11} & c_{12} & c_{13} & \cdots & c_{1n} \\
c_{21} & c_{22} & c_{23} & \cdots & c_{2n} \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
c_{m1} & c_{m2} & c_{m3} & \cdots & c_{mn}
\end{array}
$$

ちなみに

$$
\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{pmatrix}^\top
=
\begin{pmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{pmatrix}
$$

であるので, 検算に用いること.

(3) 入力を $n \times m$ 行列 A と $n \times m$ 行列 B としてその各行列の成分ごとの積 (アムダール積) を求めたい. つ

まり

$$
\begin{array}{ccccc}
n & m & & & \\
a_{11} & a_{12} & a_{13} & \cdots & a_{1m} \\
a_{21} & a_{22} & a_{23} & \cdots & a_{2m} \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nm} \\
b_{11} & b_{12} & b_{13} & \cdots & b_{1m} \\
b_{21} & b_{22} & b_{23} & \cdots & b_{2m} \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
b_{n1} & b_{n2} & b_{n3} & \cdots & b_{nm}
\end{array}
$$

という入力が与えられたときに, アダマール積を出力するプログラムを書け.

出力形式は

$$
\begin{array}{ccccc}
c_{11} & c_{12} & c_{13} & \cdots & c_{1m} \\
c_{21} & c_{22} & c_{23} & \cdots & c_{2m} \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
c_{n1} & c_{n2} & c_{n3} & \cdots & c_{nm}
\end{array}
$$

ちなみに

$$
\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{pmatrix} \odot \begin{pmatrix} 2 & 4 & 1 \\ 2 & 5 & 1 \\ 4 & 1 & 5 \\ 42 & 15 & 12 \end{pmatrix} = \begin{pmatrix} 2 & 8 & 3 \\ 8 & 25 & 6 \\ 28 & 8 & 45 \\ 420 & 165 & 144 \end{pmatrix}
$$

であるので, 検算に用いること.

問題 2

最急降下法をベースとした問題を取り扱う. 最急降下法 (Gradient descent, steepest descent) とは関数の傾き (一階微分) のみから, 関数の最小値を探索する連続最適化問題の勾配法のアルゴリズムの一つである. n 次のベクトル $\boldsymbol{x}$ が与えられたときの関数 $f(\boldsymbol{x})$ の極小値を求める方法の一つである. 多くの場合, 関数 f というのは予測値と実際の値の二乗平均誤差である. 以下の式によって更新を行う.

$$
\boldsymbol{x}^{k+1} = \boldsymbol{x}^{k} - \alpha \mathrm{grad}\ f(\boldsymbol{x}^{(k)})
$$

このとき, α とはパラメータであり, 大きすぎると変動が大きいので収束が速いが最適解を得られることは難しい. 一方で, パラメータが小さいと最適解が得られる可能性は大きくなるが一回の更新に対する変動が小さいので収束が遅い.

(1) $f(x) = 3x^4 - 8x^3 - 6x^2 + 24x + 1$ とする. このとき, $f'(x)$ を求めることによって概形, 極値を得よ.

(2) $\alpha = 10^{-7}$ (`alpha = 1e-7`) とする. ここで, $|f'(x)| < 10^{-7}$ (`abs(grad) < 1e-7`) になるまで更新を行うこととする. 初期位置 $x = 0$ のときとして更新を行った際の最小値 $f(x_{\min})$ を得よ. またこの値と (1) で求めた極値と比較せよ.

(3) $x = -10$, $x = 10$ を初期位置として更新を行った際の最小値を得よ. また (2) との結果を比較し, 考察せよ.