Matrix Multiplication

Stephen Boyd

EE103
Stanford University

October 28, 2017

Outline

Matrix multiplication

- can multiply $m \times p$ matrix A and $p \times n$ matrix B to get $C = AB$:

$$C_{ij} = \sum_{k=1}^{p} A_{ik} B_{kj} = A_{i1} B_{1j} + \cdots + A_{ip} B_{pj}$$

  for $i = 1, \ldots, m$, $j = 1, \ldots, n$
- to get C_{ij}: move along ith row of A, jth column of B
- example:

$$\begin{bmatrix} -1.5 & 3 & 2 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} -1 & -1 \\ 0 & -2 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 3.5 & -4.5 \\ -1 & 1 \end{bmatrix}$$

Special cases of matrix multiplication

- scalar-vector product (with scalar on right!) $x\alpha$
- inner product a^Tb
- matrix-vector multiplication Ax
- *outer product*

$$ab^T = \begin{bmatrix} a_1b_1 & a_1b_2 & \cdots & a_1b_n \\ a_2b_1 & a_2b_2 & \cdots & a_2b_n \\ \vdots & \vdots & & \vdots \\ a_mb_1 & a_mb_2 & \cdots & a_mb_n \end{bmatrix}$$

Properties

- $(AB)C = A(BC)$, so both can be written ABC
- $A(B + C) = AB + AC$
- $(AB)^T = B^T A^T$
- $AI = A$; $IA = A$
- $AB = BA$ *does not hold in general*

Block matrices

- block matrices can be multiplied using the same formula, *e.g.*,

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} E & F \\ G & H \end{bmatrix} = \begin{bmatrix} AE+BG & AF+BH \\ CE+DG & CF+DH \end{bmatrix}$$

  (provided the products all make sense)

Column interpretation

- write $B = \left[\begin{array}{cccc} b_1 & b_2 & \cdots & b_n \end{array} \right]$ (b_i is ith column of B)
- then we have

$$AB = A \left[\begin{array}{cccc} b_1 & b_2 & \cdots & b_n \end{array} \right] = \left[\begin{array}{cccc} Ab_1 & Ab_2 & \cdots & Ab_n \end{array} \right]$$

- so AB is 'batch' multiply of A times columns of B

Inner product interpretation

- with a_i^T the rows of A, b_j the columns of B, we have

$$AB = \left[\begin{array}{cccc} a_1^T b_1 & a_1^T b_2 & \cdots & a_1^T b_n \\ a_2^T b_1 & a_2^T b_2 & \cdots & a_2^T b_n \\ \vdots & \vdots & & \vdots \\ a_m^T b_1 & a_m^T b_2 & \cdots & a_m^T b_n \end{array}\right]$$

- so matrix product is all inner products of rows of A and columns of B, arranged in a matrix

Gram matrix

- the *Gram matrix* of an $m \times n$ matrix A is

$$G = A^T A = \left[\begin{array}{cccc} a_1^T a_1 & a_1^T a_2 & \cdots & a_1^T a_n \\ a_2^T a_1 & a_2^T a_2 & \cdots & a_2^T a_n \\ \vdots & \vdots & \ddots & \vdots \\ a_n^T a_1 & a_n^T a_2 & \cdots & a_n^T a_n \end{array} \right]$$

- Gram matrix gives all inner products of columns of A
- example: $G = A^T A = I$ means columns of A are orthonormal

Complexity

- to compute $C_{ij} = (AB)_{ij}$ is inner product of p-vectors
- so total required flops is $(mn)(2p) = 2mnp$ flops
- multiplying two 1000×1000 matrices requires 2 billion flops
- ...and can be done in well under a second on current computers

Outline

Composition of linear functions

- A is an $m \times p$ matrix, B is $p \times n$
- define $f : \mathbf{R}^p \to \mathbf{R}^m$ as $f(u) = Au$, $g : \mathbf{R}^n \to \mathbf{R}^p$ as $g(v) = Bv$
- f and g are linear functions
- *composition* is $h : \mathbf{R}^n \to \mathbf{R}^m$, $h(x) = f(g(x))$
- we have
$$h(x) = f(g(x)) = A(Bx) = (AB)x$$
so
  - composition of linear functions is linear
  - associated matrix is product of matrices of the functions

Second difference matrix

- D_n is $(n-1) \times n$ difference matrix:

$$D_n x = (x_2 - x_1, \ldots, x_n - x_{n-1})$$

- D_{n-1} is $(n-2) \times (n-1)$ difference matrix:

$$D_n y = (y_2 - y_1, \ldots, y_{n-1} - y_{n-2})$$

- $\Delta = D_{n-1} D_n$ is $(n-2) \times n$ second difference matrix:

$$\Delta x = (x_1 - 2x_2 + x_3, x_2 - 2x_3 + x_4, \ldots, x_{n-2} - 2x_{n-1} + x_n)$$

Second difference matrix

for $n = 5$, $\Delta = D_{n-1}D_n$ is

$$\begin{bmatrix} 1 & -2 & -1 & 0 & 0 \\ 0 & 1 & -2 & -1 & 0 \\ 0 & 0 & 1 & -2 & -1 \end{bmatrix} =$$

$$= \begin{bmatrix} -1 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}$$

Outline

Matrix powers

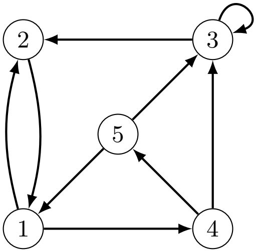
- for A square, A^2 means AA, and same for higher powers
- with convention $A^0 = I$ we have $A^k A^l = A^{k+l}$
- negative powers later; fractional powers in other courses

Directed graph

- $n \times n$ matrix A is adjacency matrix of directed graph:

$$A_{ij} = \begin{cases} 1 & \text{there is a edge from vertex } j \text{ to vertex } i \\ 0 & \text{otherwise} \end{cases}$$

- example:



$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Paths in directed graph

- $(A^2)_{ij} = \sum_{k=1}^{n} A_{ik}A_{kj}$ = number of paths of length 2 from j to i
- for the example,

$$A^2 = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 2 \\ 1 & 0 & 1 & 2 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

  e.g., there are two paths from 4 to 3 (via 3 and 5)
- more generally, $(A^\ell)_{ij}$ = number of paths of length ℓ from j

Outline

Matrix multiplication

Composition of linear functions

Matrix powers

QR factorization

Gram-Schmidt in matrix notation

- run Gram-Schmidt on columns $a_1, \ldots, a_k$ of $n \times k$ matrix A
- if columns are independent, get orthonormal $q_1, \ldots, q_k$
- define $n \times k$ matrix Q with columns $q_1, \ldots, q_k$
- $Q^T Q = I$
- from Gram-Schmidt algorithm

$$\begin{aligned} a_i &= (q_1^T a_i) q_1 + \cdots + (q_{i-1}^T a_i) q_{i-1} + \|\tilde{q}_i\| q_i \\ &= R_{1i} q_1 + \cdots + R_{ii} q_i \end{aligned}$$

  with $R_{ij} = q_i^T a_j$ for $i < j$, $R_{ii} = \|\tilde{q}_i\|$
- defining $R_{ij} = 0$ for $i > j$ we have $A = QR$
- R is upper triangular, with positive diagonal entries

QR factorization

- $A = QR$ is called *QR factorization* of A
- factors satisfy $Q^T Q = I$, R upper triangular with positive diagonal entries
- can be computed using Gram-Schmidt algorithm (or some variations)
- has a *huge* number of uses, which we'll see soon