Section 2.2 - Matrix-Matrix Multiplication: Definition

Maggie Myers
Robert A. van de Geijn

The University of Texas at Austin

Practical Linear Algebra – Fall 2009

Theorem

Let $L_A : \mathbb{R}^k \to \mathbb{R}^m$ and $L_B : \mathbb{R}^n \to \mathbb{R}^k$ both be linear transformations and, for all $x \in \mathbb{R}^n$, define the function $L_C : \mathbb{R}^n \to \mathbb{R}^m$ by $L_C(x) = L_A(L_B(x))$. Then $L_C(x)$ is a linear transformations.

Proof

Let $x, y \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$. Then

- $L_C(\alpha x) = L_A(L_B(\alpha x)) = L_A(\alpha L_B(x)) = \alpha L_A(L_B(x)) = \alpha L_C(x)$.
- $L_C(x + y) = L_A(L_B(x + y)) = L_A(L_B(x) + L_B(y)) = L_A(L_B(x)) + L_A(L_B(y)) = L_C(x) + L_C(y)$.

Towards a definition of matrix-matrix multiplication

- Let linear transformations L_A, L_B, and L_C be represented by matrices $A \in \mathbb{R}^{m \times k}$, $B \in \mathbb{R}^{k \times n}$, and $C \in \mathbb{R}^{m \times n}$, respectively.
- Then $Cx = L_C(x) = L_A(L_B(x)) = A(Bx)$.
- The matrix-matrix multiplication (product) is defined so that $C = A \times B = AB$.
- C is the composition of A and B just like L_C is the composition of L_A and L_B.

Dimensions must match

If

- A is $m_A \times n_A$ matrix,
- B is $m_B \times n_B$ matrix, and
- C is $m_C \times n_C$ matrix.

then for $C = AB$ to hold it must be the case that

- $m_C = m_A$,

How to compute the product of two matrices

Let

$$C = \begin{pmatrix} \gamma_{0,0} & \gamma_{0,1} & \cdots & \gamma_{0,n-1} \\ \gamma_{1,0} & \gamma_{1,1} & \cdots & \gamma_{1,n-1} \\ \vdots & \vdots & \vdots & \vdots \\ \gamma_{m-1,0} & \gamma_{m-1,1} & \cdots & \gamma_{m-1,n-1} \end{pmatrix}$$

$$A = \begin{pmatrix} \alpha_{0,0} & \alpha_{0,1} & \cdots & \alpha_{0,k-1} \\ \alpha_{1,0} & \alpha_{1,1} & \cdots & \alpha_{1,k-1} \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_{m-1,0} & \alpha_{m-1,1} & \cdots & \alpha_{m-1,k-1} \end{pmatrix}$$

$$B = \begin{pmatrix} \beta_{0,0} & \beta_{0,1} & \cdots & \beta_{0,n-1} \\ \beta_{1,0} & \beta_{1,1} & \cdots & \beta_{1,n-1} \\ \vdots & \vdots & \vdots & \vdots \\ \beta_{k-1,0} & \beta_{k-1,1} & \cdots & \beta_{k-1,n-1} \end{pmatrix}.$$

Recall

$\gamma_{i,j} = e_i^T(Ce_j)$: Ce_j "picks out" the jth column of C and $e_i^T(Ce_j)$ then picks out the ith element of the jth column.

- $C = AB$ is the matrix such that $Cx = A(Bx)$ for all x. Then

$$\gamma_{i,j} = e_i^T(Ce_j) = \underbrace{e_i^T(A\underbrace{(Be_j)}_{b_j})}_{i\text{th element of } Ab_j}.$$

- The ith element of vector Ab_j is given by the dot product of the ith row of A with the vector b_j.
- The ith row of A and jth column of B are given by

$$\begin{pmatrix} \alpha_{i,0} & \alpha_{i,1} & \cdots & \alpha_{i,k-1} \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} \beta_{0,j} \\ \beta_{1,j} \\ \vdots \\ \beta_{k-1,j} \end{pmatrix}.$$

- $\gamma_{i,j} = e_i^T(A(Bx))e_j =$
  $\alpha_{i,0}\beta_{0,j} + \alpha_{i,1}\beta_{1,j} + \cdots + \alpha_{i,k-1}\beta_{k-1,j} = \sum_{p=0}^{k-1} \alpha_{i,p}\beta_{p,j}$.

Definition of Matrix-matrix multiplication

Let $A \in \mathbb{R}^{m \times k}$, $B \in \mathbb{R}^{k \times n}$, and $C \in \mathbb{R}^{m \times n}$. Then the matrix-matrix multiplication (product)

$$C = AB$$

is computed by setting

$$\gamma_{i,j} = \sum_{p=0}^{k-1} \alpha_{i,p}\beta_{p,j} = \alpha_{i,0}\beta_{0,j} + \alpha_{i,1}\beta_{1,j} + \cdots + \alpha_{i,k-1}\beta_{k-1,j}.$$

Note

- $Cx = A(Bx) = (AB)x$.
- We can drop the parentheses, unless they are useful for clarity:

$$Cx = (AB)x = A(Bx) = ABx.$$

Example: Predicting the Weather

- Recall that

$$P = \begin{pmatrix} 0.4 & 0.3 & 0.1 \\ 0.4 & 0.3 & 0.6 \\ 0.2 & 0.4 & 0.3 \end{pmatrix}$$

  was the transition matrix so that if x was the prediction for today's weather, then Px was the prediction for tomorrow's weather.
- The transition matrix Q so that $z = Qx$ is the prediction for the day after tomorrow is given by $Q = P \times P$, since $z = P(Px) = Qx$:

$$\begin{aligned} Q = P \times P &= \begin{pmatrix} 0.4 & 0.3 & 0.1 \\ 0.4 & 0.3 & 0.6 \\ 0.2 & 0.4 & 0.3 \end{pmatrix} \begin{pmatrix} 0.4 & 0.3 & 0.1 \\ 0.4 & 0.3 & 0.6 \\ 0.2 & 0.4 & 0.3 \end{pmatrix} \\ &= \begin{pmatrix} 0.30 & 0.25 & 0.25 \\ 0.40 & 0.45 & 0.40 \\ 0.30 & 0.30 & 0.35 \end{pmatrix}. \end{aligned}$$

Example: Predicting the Weather

- The transition matrix W so that $z = Wx$ is the prediction for a week from today is given by

$$\begin{aligned} W &= \underbrace{P \times P \times P \times P \times P \times P \times P}_{\text{seven } P\text{s}} \\ &= P(P(P(P(P(P(P)))))) \\ &= (P \times P) \times (P \times P) \times (P \times P) \times P \\ &= (P \times P \times P) \times (P \times P \times P) \times P \end{aligned}$$

$$\begin{aligned} & P \times P \times P \\ &= \begin{pmatrix} 0.4 & 0.3 & 0.1 \\ 0.4 & 0.3 & 0.6 \\ 0.2 & 0.4 & 0.3 \end{pmatrix} \left(\begin{pmatrix} 0.4 & 0.3 & 0.1 \\ 0.4 & 0.3 & 0.6 \\ 0.2 & 0.4 & 0.3 \end{pmatrix} \begin{pmatrix} 0.4 & 0.3 & 0.1 \\ 0.4 & 0.3 & 0.6 \\ 0.2 & 0.4 & 0.3 \end{pmatrix} \right) \\ &= \begin{pmatrix} 0.4 & 0.3 & 0.1 \\ 0.4 & 0.3 & 0.6 \\ 0.2 & 0.4 & 0.3 \end{pmatrix} \begin{pmatrix} 0.30 & 0.25 & 0.25 \\ 0.40 & 0.45 & 0.40 \\ 0.30 & 0.30 & 0.35 \end{pmatrix} = \cdots \end{aligned}$$

Example: Composing Two Rotations

$$
\begin{aligned}
C &= \begin{pmatrix} \cos(\rho+\sigma) & -\sin(\rho+\sigma) \\ \sin(\rho+\sigma) & \cos(\rho+\sigma) \end{pmatrix} \\
&= \begin{pmatrix} \cos\rho & -\sin\rho \\ \sin\rho & \cos\rho \end{pmatrix} \begin{pmatrix} \cos\sigma & -\sin\sigma \\ \sin\sigma & \cos\sigma \end{pmatrix} \\
&= \begin{pmatrix} \cos\rho\cos\sigma - \sin\rho\sin\sigma & -\sin\rho\cos\sigma - \cos\rho\sin\sigma \\ \sin\rho\cos\sigma + \cos\rho\sin\sigma & \cos\rho\cos\sigma - \sin\rho\sin\sigma \end{pmatrix}.
\end{aligned}
$$

Conformal matrices

- For matrix-matrix multiplication to be a legal operations, the row and column dimensions of the matrices must obey the mentioned constraints.
- We say dimensions are *conformal* if the operation being performed with the matrices and/or vectors are legal.

Computing The Matrix-Matrix Product

The following triple-nested loops compute $C := AB + C$:

for $i = 0, \ldots, m-1$
  for $j = 0, \ldots, n-1$
    for $p = 0, \ldots, k-1$
      $\gamma_{i,j} := \alpha_{i,p}\beta_{p,j} + \gamma_{i,j}$
    endfor
  endfor
endfor

If originally $C = 0$, then the above algorithm computes $C := AB$.

Simple triple-nested loops

for $j = 0, \ldots, n-1$
  for $i = 0, \ldots, m-1$
    for $p = 0, \ldots, k-1$
      $\gamma_{i,j} := \alpha_{i,p}\beta_{p,j} + \gamma_{i,j}$
    endfor
  endfor
endfor

$$\left.\begin{array}{l} \text{for } p = 0, \ldots, k-1 \\ \quad \gamma_{i,j} := \alpha_{i,p}\beta_{p,j} + \gamma_{i,j} \\ \text{endfor} \end{array}\right\} \gamma_{i,j} := \sum_{p=0}^{k-1} \alpha_{i,p}\beta_{p,j} + \gamma_{i,j}$$

```
for j=1:n
  for i=1:m
    for p=1:k
      C(i,j) += A(i,p) * B(p,j);
    end
  end
end
```

Double Nested Loop, calling DOT

for $j = 0, \ldots, n-1$
  for $i = 0, \ldots, m-1$
    for $p = 0, \ldots, k-1$
      $\gamma_{i,j} := \alpha_{i,p}\beta_{p,j} + \gamma_{i,j}$
    endfor
  endfor
endfor

$\Big\} \gamma_{i,j} := \check{a}_i^T b_j + \gamma_{i,j}$

```
for j=1:n
  for i=1:m
    C(i,j) += SLAP_Dot( A(i,:), B(:,j) );
  end
end
```

Single loop with call to GEMV

for $j = 0, \ldots, n-1$
  for $i = 0, \ldots, m-1$
    for $p = 0, \ldots, k-1$
      $\gamma_{i,j} := \alpha_{i,p}\beta_{p,j} + \gamma_{i,j}$
    endfor
  endfor
endfor

$\Big\} \; c_j := Ab_j + c_j$

```
for j=1:n
  C(:,j) += ...
    SLAP_Gemv( SLAP_NO_TRANSPOSE, ...
           1, A, B(:,j), 1, C(:,j) );
end
```

Ordering the loops

- When computing $C = AB + C$ the three loops can be nested in $3! = 6$ different ways:

$$ijp, ipj, jip, jpi, pij, pji.$$

- We will examine this more, later.

for $= 0, \ldots,$
  for $= 0, \ldots,$
    for $= 0, \ldots,$
      $\gamma_{i,j} := \alpha_{i,p}\beta_{p,j} + \gamma_{i,j}$
    endfor
  endfor
endfor