CSD101: Introduction to computing and programming (ICP)

Multi-dimensional arrays

- We used arrays to represent sequences. But arrays are a very general way to store multi-dimensional data. In particular 2D arrays are very useful in practice.
- 2D or two dimensional data is so common in applications that we have full packages to support them - for example spreadsheets.
- Just as 1D arrays represented sequences or vectors 2D arrays represent mathematical objects called matrices.
- A matrix is a mathematical object that can be visualized as a 2D table of values - typically floating point numbers - and is naturally represented by a 2D array.

Example of a matrix and its array representation I

- An $m \times n$ matrix (or array) has m rows and n columns. An example 3×4 matrix A is shown below:

$$A = \begin{bmatrix} 3.1 & 2.8 & 7.5 & 4.2 \\ 2.2 & 1.7 & 4.8 & 5.9 \\ 9.0 & 8.4 & 6.3 & 5.5 \end{bmatrix}$$

- The corresponding array representation in **C** is:

```
float a[3][4]={{3.1,2.8,7.5,4.2},{2.2,1.7,4.8,5.9},{9.0,8.4,6.3,5.5
```

Example of a matrix and its array representation II

- Elements in matrices and arrays are referred to by row, column indices. For example, the value 4.8 can be accessed by $a[1][2]$ - remember indices start at 0 in **C** . The matrix notation will normally write it as a subscript $A_{2,3}$ - matrix index count starts from 1 (by convention). Matrices are usually denoted by capital letters.
- Arrays are stored in row major form (i.e. row-wise). So, in passing arguments only the first argument can remain unspecified. The number of columns has to be specified.

Matrix operations I

- As for other math objects we can define operations on matrices. Let $A_{m \times n}$, $B_{m \times n}$, $C_{n \times p}$ be three matrices with the dimensions as shown and let $\alpha \in \mathbb{R}$ be a scalar (i.e. real number). Each element in a matrix is denoted by the corresponding small case letter. So, $a_{i,j}$ stands for the element in location (i, j) in matrix A.
- Multiplication by a scalar.
  $\alpha \times A = [\alpha a_{i,j}]$, for $1 \leq i \leq m$, $1 \leq j \leq n$. That is every element of A is multiplied by α.
- Addition, subtraction.
  $A \pm B = [a_{i,j} \pm b_{i,j}]$, for $1 \leq i \leq m$, $1 \leq j \leq n$. The corresponding elements are added or subtracted to get the result. Note A, B must be compatible (i.e. have the same dimensions).

Matrix operations II - multiplication

- Multiplication. $C_{m \times p} = A_{m \times n} \times B_{n \times p}$. The element $c_{i,j} = \sum_{k=1}^{n}(a_{i,k} \times b_{k,j})$. The product element is obtained by multiplying the ith row of A with the jth column of B and adding the corresponding products. So, the multiplicands A and B must be compatible for multiplication to be possible - that is - number of columns in A must equal number of rows in B.

Matrix operations III: transpose I

- Transpose of an $m \times n$ matrix A is the $n \times m$ matrix A^T which has the rows and columns of A inter changed. For example, if A is:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}$$

then A^T is given by (the element references are to A):

$$A^T = \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \\ a_{13} & a_{33} \end{bmatrix}$$

Determinant of a matrix I

- The **determinant** of an $n \times n$ square matrix A, denoted $det(A)$ is a scalar value computed from the entries of the matrix. The determinant describes some properties of the linear transformation represented by the matrix A.
- The determinant of A can be calulated by an inductive calculation as follows called the Laplace expansion.
  1. If A is a 1×1 matrix - that is just a value a - then $det(A) = a$.
  2. If A is a 2D matrix

  $$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

  then $det(A) = a \times d - b \times c$.

Determinant of a matrix II

3. If A is an $n \times n$ matrix then:
   $det(A) = \sum_{j=1..n}^{n} a_{1j} C_{1j}$ where C_{ij} is (ij) **cofactor** of A.
   Cofactor $C_{ij} = (-1)^{i+j} M_{ij}$, where M_{ij} is called the (ij) **minor** of A and is defined as $M_{ij} = det(A')$ where A' is obtained from A by deleting the i^{th} row and j^{th} column from A giving an $(n-1) \times (n-1)$ matrix.

- Using the above inductive definition we can recursively compute $det(A)$.
- An example. Let A be:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Determinant of a matrix III

$$\begin{aligned} det(A) &= a_{11}C_{11} + a_{12}C_{12} + a_{13}C_{13} \\ &= a_{11}M_{11} - a_{12}M_{12} + a_{13}M_{13} \\ &= a_{11}(a_{22}a_{33} - a_{23}a_{32}) - a_{12}(a_{21}a_{33} - a_{23}a_{31}) \\ &\quad + a_{13}(a_{21}a_{32} - a_{22}a_{31}) \end{aligned}$$

Inverse of a matrix

- An **identity** matrix I_n is an $n \times n$ square matrix where the right diagonal has only 1s and all the rest are 0s.
- The inverse of an $n \times n$ matrix A written A^{-1} is defined by the equation: $AA^{-1} = A^{-1}A = I_n$ provided $det(A) \neq 0$. If $det(A) = 0$ then A is said to be **singular** and the inverse does not exist.
- A definition of A^{-1} is:

$$A^{-1} = \frac{1}{det(A)}\text{Adjoint}(A)$$

  where Adjoint$(A = C^T)$, C is the cofactor matrix of A - that is each element a_{ij} is replaced by C_{ij} to obtain C.
- A calculation using the above formula is extremely inefficient.
- Gaussian elimination can be used to calculate $det(A)$ and A^{-1} efficiently.

Gaussian elimination to solve a system of linear equations I

- A set of n linear equations in n unknowns $x_1, x_2, \ldots, x_n$ can be represented using an $n \times n$ coefficient matrix A and $n \times 1$ column vector B representing the right hand sides of the equations. For example, the system of n linear equations:

$$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1$$
$$\ldots$$
$$\ldots$$
$$a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n$$

is represented by the $n \times n$ matrix $A = [a_{ij}]$, $i, j = 1..n$ and the right handside by the $n \times 1$ column vector $B = [b_j]$, $j = 1..n$.

Gaussian elimination to solve a system of linear equations II

- To solve the system of equations we do two steps i) elimination - converts A to an upper triangular matrix ii) backsubstitution which gives us the values for each unknown variable starting from x_n and moving to x_1.

Row reduction

Row reduction is done using elementary row operations:

- Swap any two rows.
- Multiply a row by a non-zero constant.
- Add a scalar multiple of one row to another.

- After row reduction is complete the orginal coefficient matrix becomes an upper triangular matrix.
- This can be easily used to find the values of the unknowns by the process of back substitution.
- Each elementary row operation is actually equivalent to pre-multiplying matrix A by a specific matrix corresponding to the elementary row operation. That is at the heart of why it works.

Example

Example:

$r_2 = r_2 + \frac{-a_{21}}{a_{11}} \times r_1$, $r_3 = r_3 + \frac{-a_{31}}{a_{11}} r_1$

x_1	x_2	x_3	
2	1	−1	8
−3	−1	2	−11
−2	1	2	−3

$\longrightarrow$

2	1	−1	8
0	$\frac{1}{2}$	$\frac{1}{2}$	1
−2	1	2	−3

$\longrightarrow$

2	1	−1	8
0	$\frac{1}{2}$	$\frac{1}{2}$	1
0	2	1	5

$r_3 = r_3 + \frac{-a_{32}}{a_{22}} \cdot r_3$

$\rightarrow$

2	1	−1	8
0	$\frac{1}{2}$	$\frac{1}{2}$	1
0	0	−1	1

backsubstitute $\longrightarrow$

$x_3 = -1$

$x_2 = \left(1 - \frac{x_3}{2}\right) 2 = 3$

$x_1 = (8 + x_3 - x_2) \frac{1}{2}$

$= 2$