General Computer Science for Engineers CISC 106 Lecture 24

Christopher Thorpe
Computer and Information Sciences
04/17/2009

Lecture Overview

- Structs (137-140)
- Vectorization
- Matrix Tricks

A Database Application

- Given:

Name: Chris
Credits: 27
Graduation: 12/15/2011

Name: Sola
Credits: 18
Graduation: 05/17/2011

Name: Roger
Credits: 55
Graduation: 06/10/2009

Name: Tom
Credits: 15
Graduation: 05/22/2012

A Database Application

- Given:

Name: Chris
Credits: 27
Graduation: 12/15/2011

Name: Sola
Credits: 18
Graduation: 05/17/2011

Name: Roger
Credits: 55
Graduation: 06/10/2009

Name: Tom
Credits: 15
Graduation: 05/22/2012

We can implement it with arrays like this:

Chris	27	12/15/2011
Sola	18	05/17/2011
Roger	55	06/10/2009
Tom	15	05/22/2012

A Database Application

- Given:

Name: Chris
Credits: 27
Graduation: 12/15/2011

Name: Sola
Credits: 18
Graduation: 05/17/2011

Name: Roger
Credits: 55
Graduation: 06/10/2009

Name: Tom
Credits: 15
Graduation: 05/22/2012

we can do it like this an array with structs:

Students (1). Name: Chris
Students (1).Credits: 27
Students (1). Graduation: 12/15/2011

Students (2).Name: Sola
Students (2).Credits: 18
Students (2).Graduation: 05/17/2011

Students (3). Name: Roger
Students (3). Credits: 55
Students (3). Graduation: 06/10/2009

Students (4). Name: Tom
Students (4). Credits: 15
Students (4). Graduation: 05/22/2012

Initializing an Array of Structs

- Students = struct{‘name’, {‘Chris’, ‘Sola’,’Roger’, Tom’},
  ‘credits’, {27, 18, 55, 15}, ‘graduation’,
  { ‘12/15/2011’,’ 05/17/2011’,’ 06/10/2009’,’05/22/2012’}

- record1.name = 'Me';
  record1.credits = 27;
  record1.age = 10;

  record2.name = 'Not Me';
  record2.credits = 30;
  record2.age = 14;

  record1
  record2

  record_array = [record1, record2];

Vectorization

- What is vectorization?
  - Functions that can be performed on the entire array instead of just one element in an array.
  - Needed because MATLAB is an interpreter not a compiler AND because speed is important.
- Advantages of vectorization
  - Fast and compact.
- Disadvantages of vectorization
  - Hard to look at what is going on 'inside'
  - Application to your current code is not always apparent

Vectorization

- Given an array of arbitrary size: Square each element in the array and put the result into a new array.

- The 'loopy' way
  - function output = square(input)

```
    n = length(input);
     for i = 1:n
         output(i) = input(i)^2;
    end
end
```

- The 'vector' way
  - Output = input.^2;

Vectorization

- Given an array of arbitrary size: Tell me how many elements of the given array are less than a given value.

- The 'loopy' way

```
function count = num_less_than(input, value)
    n = length(input);
    count = 0;
    for i = 1:n
    if (input(i) < value)
        count = count + 1;
    end
    end
```

- The 'vector' way

```
    vector_less = (input < value);
    count = sum(vector_less);
```

Vectorization

Given a grayscale image of arbitrary size: Make all pixels less than value1 equal to value2

- The 'loopy' way

```
function output = num_less_than(input, value1, value2)
    [rows, cols] = size(input);
    for i = 1:rows
        for j = 1:cols
            if(input(i, j) < value1)
                output(i, j) = value2;
            else
                output(i, j) = input(i, j);
            end
        end…..
```

- The 'vector' way

```
vector_less = (input < value);
vector_less = ~vector_less;
temp = vector_less.*input;
vector_less = vector_less.*5;
output = vector_less+temp;
```

Matrix Tricks

- Sometimes you want to know how many elements are in a matrix.
  - Useful for normalization problems.
  - Num_elements = numel(input);
- Sometimes you want to change the shape of a matrix.
  - For matrix multiplication
  - For simple output
  - New_matrix = reshape(matrix, m, n);

Matrix Tricks

- Sometimes your matrix is 'bloated' (seems larger than it really is)
  - Makes some mathematical operations impossible.
  - Makes accessing some dimensions tedious.
  - New_matrix = squeeze(matrix);
- Sometimes you need to change the order in which some dimensions appear.
  - For matrix multiplication
  - For simple output
  - New_matrix = shiftdim(matrix, n);