March | 7

COMP41610 – Practical 3 – Neil Grogan - 13204052

Neil Grogan

University College Dublin

I. Matrix multiplication

Map

For the map function we could split the matrices in to the numbers they will multiplied by:

A = 2*2 matrix:
[1,2]
[3,4]

B = 2*2 matrix:
[5,6]
[7,8]

would become:
(A[0,0], B[0,0]) -> (0,0 [1,5])
(A[0,1], B[1,0]) -> (0,0 [2,7])

Reduce

Reduce function we could then multiply this line on each core (and every other line at the same time):

(0, 1), * (0,5) = (0,5)
(0, 2), * (0,7) = (0,14)

This could then be further reduced by adding the result of the multiplication:
(0,0) = (0,5) + (0,14)

∴ (0,0) = (19)

Mapper:
Input: <row, [values]>
Ex: <A, [1,5]> <B, [2,7]>

Emit: <row, [value]>
Ex: <0,0, [1]>, <0,0, [5]>, <0,1, [2]>, <1,1, [7]>

Reducer:
Input: <row, [value]>
Ex: <0,0, [1]>, <0,0, [5]>, <0,1, [2]>, <1,1, [7]>

Emit: <rows, [values]>
Ex. <0,0, [5]>, <0,1 [14]>

Keep reducing:
Input: <rows, [value]>
Ex. <0,0, [5]>, <0,1 [14]>

Emit: <row, [value]>
Ex. <0,0, [19]>

II. Dissimilarity Matrix

For the map function we could split the matrix in to the numbers they will operated on:

A = {x1(5, 3), x2(2,6), x3(4,1)}

would become:
(x1(5, 3), x2(2,6)) -> (x1i [2,5]), (x2i [6,3])
(x1(5, 3), x3(4,1)) -> (x1j [4,5]), (x3j [1,3])
(x2(2,6), x3(4,1)) -> (x2k [4,2]), (x3k [1,6])

Reduce
Reduce function we could then minus these numbers on each core (and every other line at the same time):

(x1i [5,2]) -> 2-5 -> (x1i [-3])
(x2i [6,3]) -> 6-3 -> (x2i [3])

Reduce
We could further reduce them by squaring these numbers on each core (and every other line at the same time):

(x1i [-3]) -> 9 ->(x1i [9])
(x2i [3]) -> 9 -> (x2i [9])

Reduce
We could further reduce them by adding these numbers on each core (and every other line at the same time):

(x1i [9]) + (x2i [9]) -> 18 -> (x12i [18])

This could then be further reduced by getting the square root as the result:
(x12i [18]) -> 4.242640687119285

∴ (x12i) = 4.242640687119285

Mapper:
Input: <segment1, [values]> <segment2, [values]>
Ex: <A, [1,5]> <B, [2,7]>

Emit: <segment_joined, [values]>
Ex: <A_B [1,5,2,7]>

Reducer:
Input: : <segment_joined, [values]>
Ex: <A_B [1,5,2,7]>

Emit: <seg1, [values]>, <seg2, [values]>

Ex. <A1 [2,1]>, <B1 [7,5]>

Keep reducing (minus):
Input: <seg1, [values]>, <seg2, [values]>
Ex. <A1 [2,1]>, <B1 [7,5]>

Emit: <segment, [value]>, <segment, [value]>
Ex. <A1 [1]>, <B1 [2]>

Keep reducing (square):
Input: <segment, [value]>, <segment, [value]>
Ex. <A1 [1]>, <B1 [2]>

Emit: <segment, [value]>, <segment, [value]>
Ex. <A1 [1]>, <B1 [4]>

Keep reducing (add):
Input: <segment, [value]>
Ex. <A1 [1]>, <B1 [4]>

Emit: <segment, [value]>
Ex. <A1B1 [5]>

Keep reducing (square root):
Input: <segment, [value]>
Ex. <A1B1 [5]>

Emit: <segment, [value]>

Ex. <A1B1 [2.23606797749979]>