

Neural Network and
optimization of
calculation

LSI Design Contest

The 21st LSI 2018 Design Contest In Okinawa

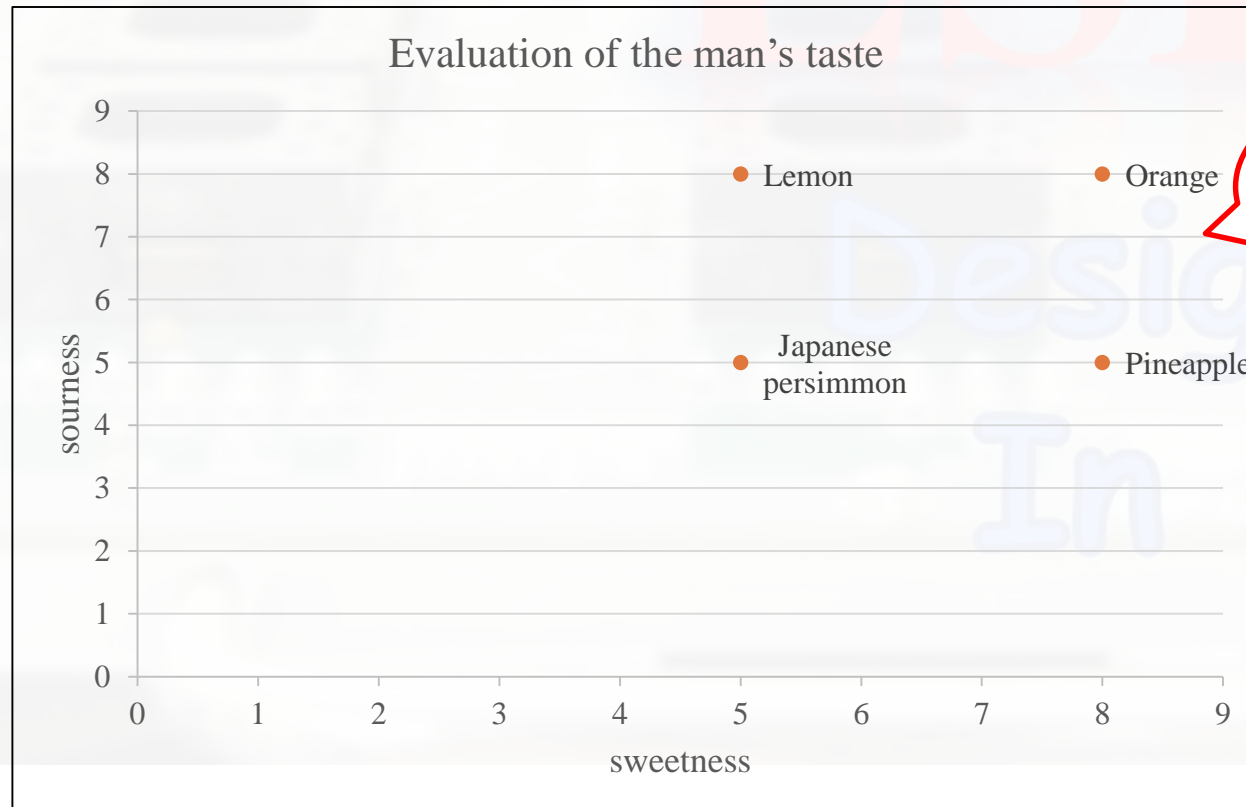


Introduction

- This is an example of Neural network calculation.
- The neural network structure used here is 3-layer structure.
- It consist of 2 input units, 3 hidden units and 2 output units.

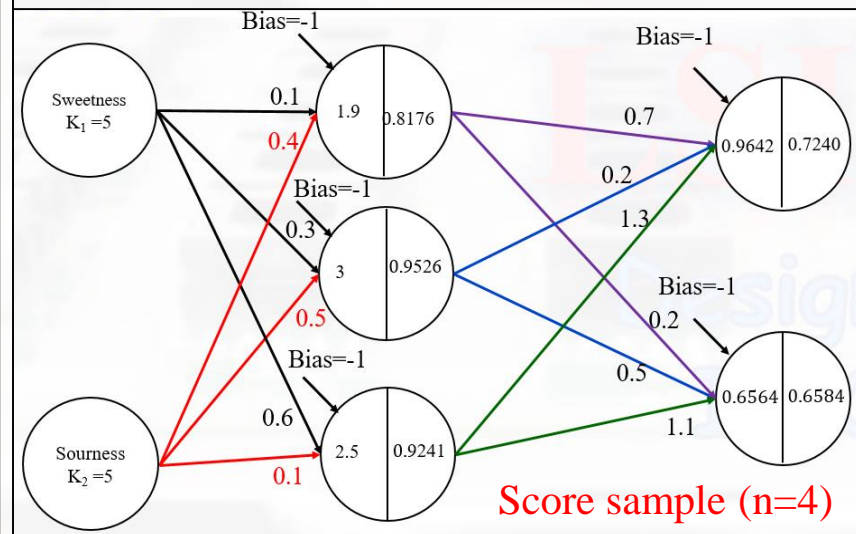
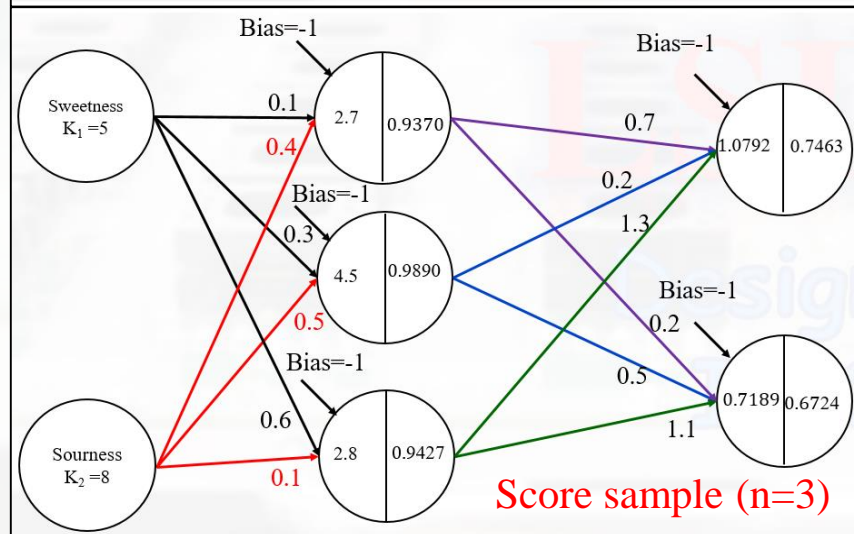
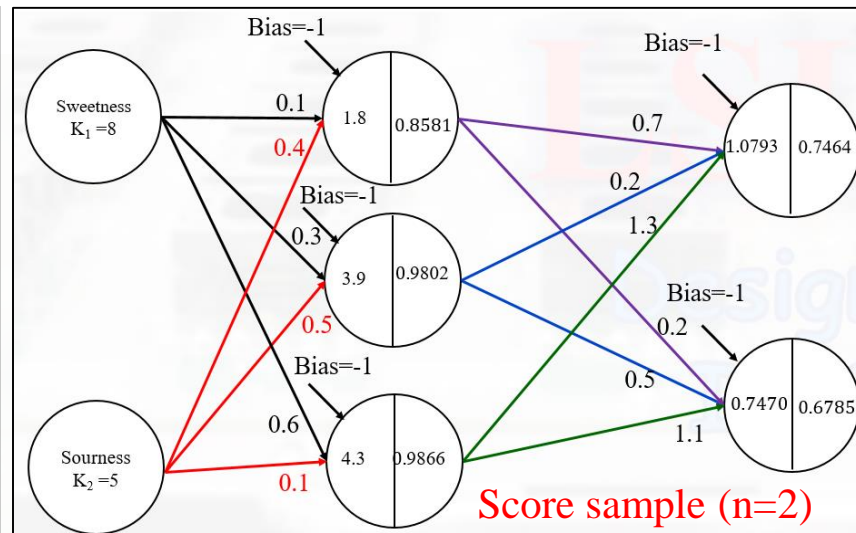
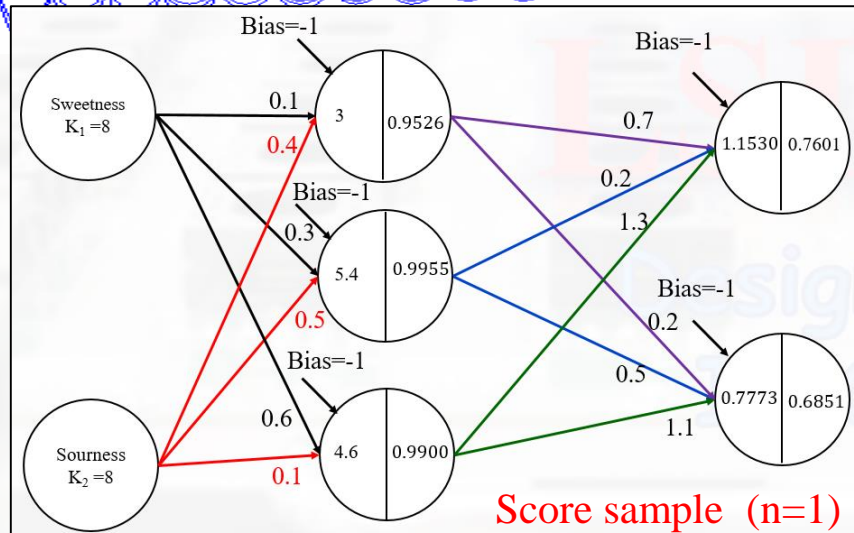
State condition

- 4 types of fruits : Orange, lemon, pineapple and Japanese persimmon
- A man eat these 4 types of fruits and decide the level of sweetness and sourness of the fruits from the range of 0 to 10
- After deciding the level of sweetness and sourness, he then decide which fruits he likes and not which fruit he dislikes
- So let's consider the fruits he likes as $[1,0]$ and the fruits he dislike as $[0,1]$



How to measure the man's taste using Neural Network?

Score sample (Initial parameter)



Rough Results

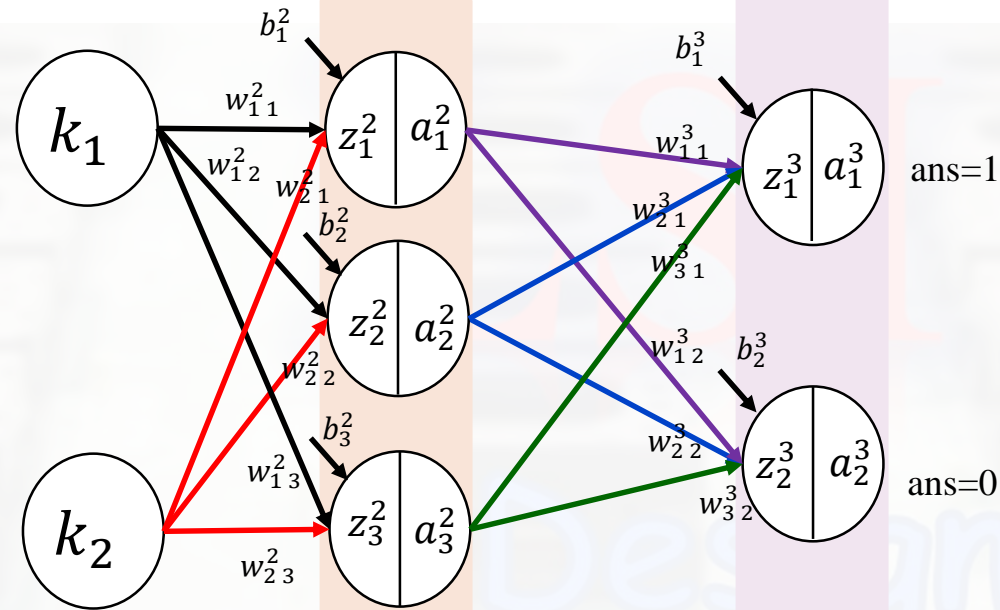
- From above example, the result for four students are as below.

Fruits n	Sweetness $K_1[n]$	Sourness $K_2[n]$	Supervisor $[t_1[n], t_2[n]]$	Taste	Output value $[a_1^3[n], a_2^3[n]]$
Orange	8	8	[1,0]	Like	[0.76, 0.69]
Lemon	8	5	[0,1]	Dislike	[0.75, 0.68]
Pineapple	5	8	[0,1]	Dislike	[0.75, 0.67]
Persimmon	5	5	[0,1]	Dislike	[0.72, 0.66]

- cost function: $C = C_1 + C_2 + \dots + C_n + \dots$
- square error: $C_n = \frac{1}{2} \{ (a_1^3[n] - t_1[n])^2 + (a_2^3[n] - t_2[n])^2 \}$
- $z_1^2 = w_{11}^2 k_1 + w_{21}^2 k_2 + b_1^2$
- $\delta_1^3 = (a_1^3 - t_1) a'(z_1^3)$
- $\delta_1^2 = (\delta_1^3 w_{11}^3 + \delta_2^3 w_{21}^3 + \dots) a'(z_1^2)$
- $\frac{\partial C}{\partial w_{ij}^m} = \delta_j^m a_i^{m-1},$ however, $a_i^1 = K_i$
- $\frac{\partial C}{\partial b_j^m} = \delta_j^m$

Omitting
[n]

Equation of z and a



$$z_i^2 = w_{1i}^2 k_1 + w_{2i}^2 k_2 + b_i^2$$

$$a_i^2 = a(z_i^2) = \frac{1}{1 + e^{-z_i^2}}$$

$$a'(z_i^2) = \frac{e^{-z_i^2}}{(e^{-z_i^2} + 1)^2} = ((1 - a_i^2)a_i^2)$$

$$z_i^3 = w_{1i}^3 a_i^2 + w_{2i}^2 a_i^2 + w_{3i}^2 a_i^2 + b_i^3$$

$$a_i^3 = a(z_i^3) = \frac{1}{1 + e^{-z_i^3}}$$

$$a'(z_i^3) = \frac{e^{-z_i^3}}{(e^{-z_i^3} + 1)^2} = ((1 - a_i^3)a_i^3)$$



Gradient descent

■ C is all learning data's error.

⇒ You find weight and bias that make minimize C to differentiate C with respect to weight and bias.

⇒ Gradient descent

$$(\Delta w_{11}^2 \dots, \Delta w_{11}^3 \dots, \Delta b_1^2 \dots, \Delta b_1^3 \dots) = -\eta \left(\frac{\partial C}{\partial w_{11}^2} \dots, \frac{\partial C}{\partial w_{11}^3} \dots, \frac{\partial C}{\partial b_1^2} \dots, \frac{\partial C}{\partial b_1^3} \dots \right)$$

If this numerical expression satisfy, C is the most smaller.

But it is too difficult to calculate this parameter.

If numbers of input units:10 , hidden units:10 , output units:3 ,

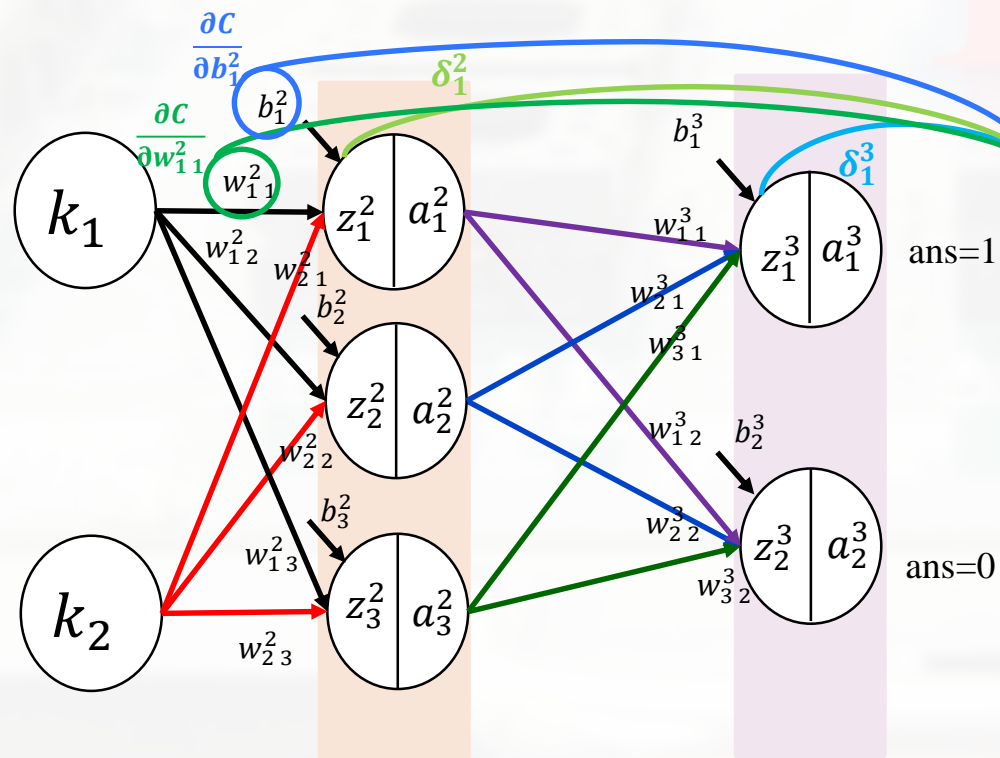
numbers of parameter is $10 \times 10 + 10(\text{bias}) + 10 \times 3 + 3(\text{bias}) = 143$.

⇒ want to calculate at all once, if it's possible decrease difference...

⇒ Back propagation.

Solve Gradient descent

■ Partial differential term can be generalized by below method.



$$\frac{\partial C}{\partial w_{11}^2} = \frac{\partial C}{\partial z_1^2} \frac{\partial z_1^2}{\partial w_{11}^2} = \frac{\partial C}{\partial z_1^2} k_1$$

$$(\because z_1^2 = w_{11}^2 k_1 + w_{12}^2 k_2 + w_{13}^2 k_3 + \dots + b_1^2)$$

$$\text{define } \frac{\partial C}{\partial z_1^2} = \delta_1^2 \rightarrow \frac{\partial C}{\partial w_{11}^2} = \delta_1^2 k_1$$

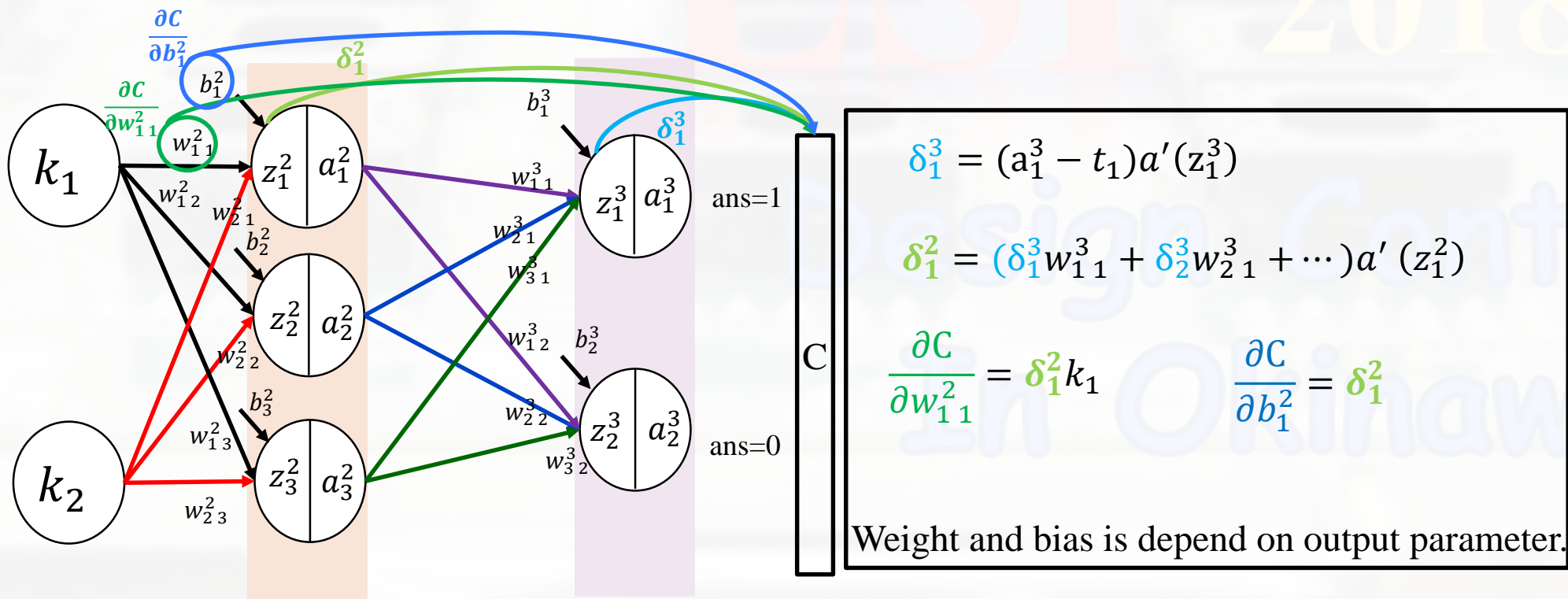
$$\text{while } \frac{\partial C}{\partial b_1^2} = \frac{\partial C}{\partial z_1^2} \frac{\partial z_1^2}{\partial b_1^2} = \delta_1^2$$

$$\therefore \frac{\partial C}{\partial w_{ij}^m} = \delta_j^m a_j^{m-1}, \frac{\partial C}{\partial b_j^m} = \delta_j^m \quad (m=2 \text{ or } 3)$$

$$(a_j^1 = k_j)$$

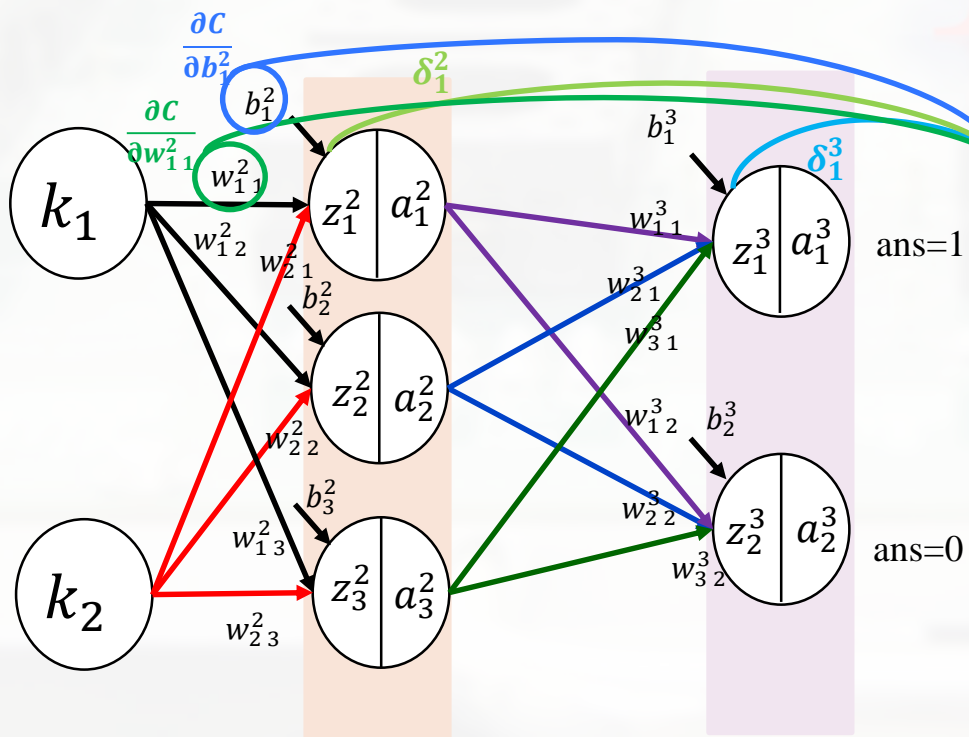
Back propagation

- Back propagation is optimization of calculation.
calculate all parameter to return output to input.



Calculation of δ

- δ is errors. it is unit's error. If you can express δ in known parameter, you don't have to differential the calculation.



It is easy to calculate output layer errors.

$$\delta_1^3 = \frac{\partial C}{\partial z_1^3} = \frac{\partial C}{\partial a_1^3} \frac{\partial a_1^3}{\partial z_1^3} = \frac{\partial C}{\partial a_1^3} a'(z_1^3)$$

$$\therefore \delta_1^3 = (a_1^3 - t_1) a'(z_1^3)$$

$\therefore C$ is made of the difference between t and a_3 .

But hidden layer errors is more difficult to calculate

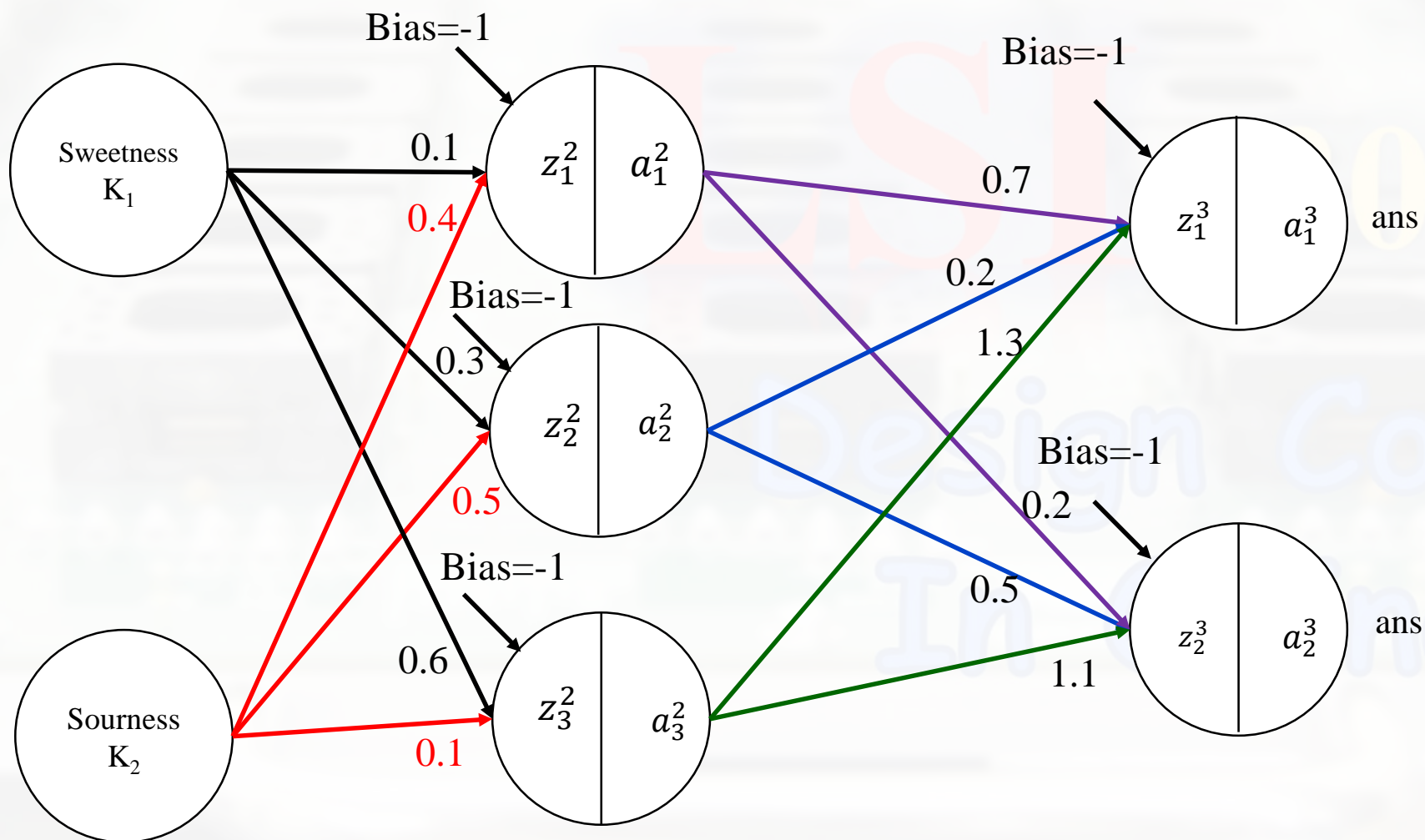
$$\delta_1^2 = \frac{\partial C}{\partial z_1^2} = \frac{\partial C}{\partial z_1^3} \frac{\partial z_1^3}{\partial a_1^2} \frac{\partial a_1^2}{\partial z_1^2} + \frac{\partial C}{\partial z_2^3} \frac{\partial z_2^3}{\partial a_1^2} \frac{\partial a_1^2}{\partial z_1^2} + \dots$$

$$(\frac{\partial C}{\partial z_1^3} = \delta_1^3, \frac{\partial C}{\partial z_2^3} = \delta_2^3, \dots \frac{\partial z_1^3}{\partial a_1^2} = w_{11}^3, \frac{\partial z_2^3}{\partial a_1^2} = w_{21}^3, \dots)$$

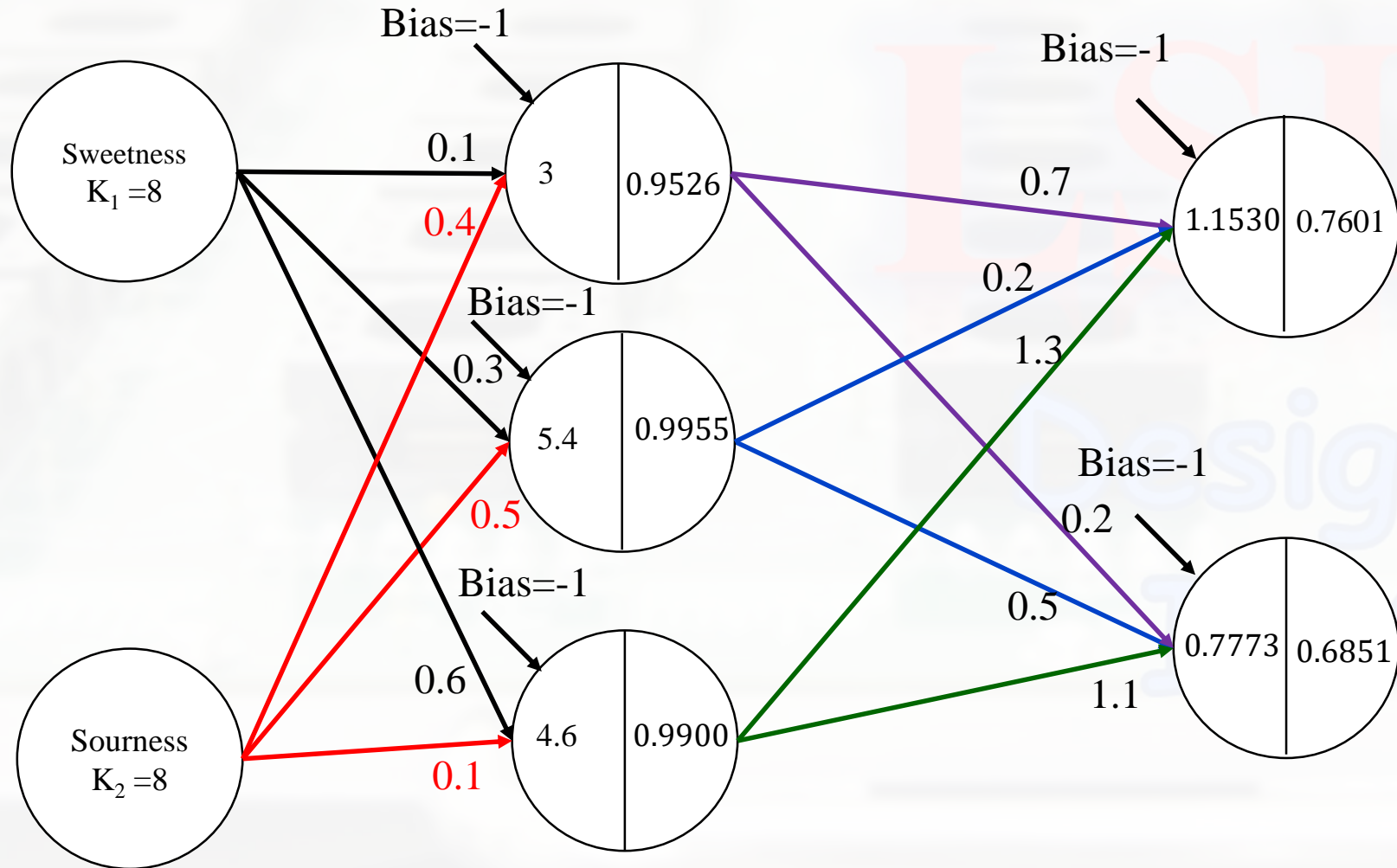
$$\delta_1^2 = \delta_1^3 w_{11}^3 a'(z_1^2) + \delta_2^3 w_{21}^3 a'(z_1^2) + \dots$$

$$\therefore \delta_1^2 = (\delta_1^3 w_{11}^3 + \delta_2^3 w_{21}^3 + \dots) a'(z_1^2)$$

Initial parameter



Score (n=1) : [8,8] (z2,a2,z3,a3)



$i = 1, 2, 3$

$$\therefore z_i^2 = w_{1i}^2 k_1 + w_{2i}^2 k_2 + b_i^2$$

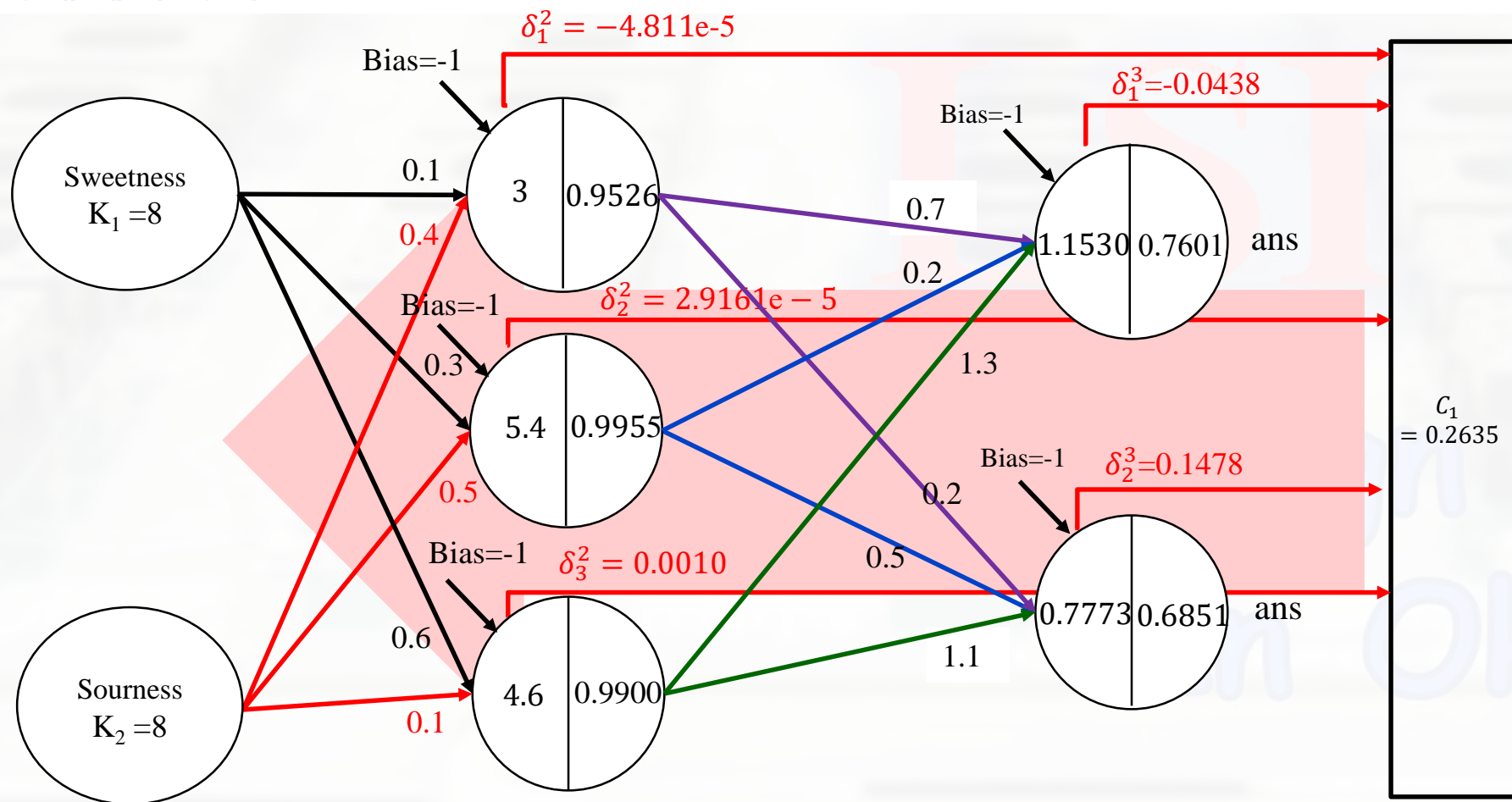
$$\therefore a_i^2 = a(z_i^2) = \frac{1}{1 + e^{-z_i^2}}$$

$i = 1, 2$

$$\therefore z_i^3 = w_{1i}^3 a_i^2 + w_{2i}^3 a_i^2 + w_{3i}^3 a_i^2 + b_i^3$$

$$\therefore a_i^3 = a(z_i^3) = \frac{1}{1 + e^{-z_i^3}}$$

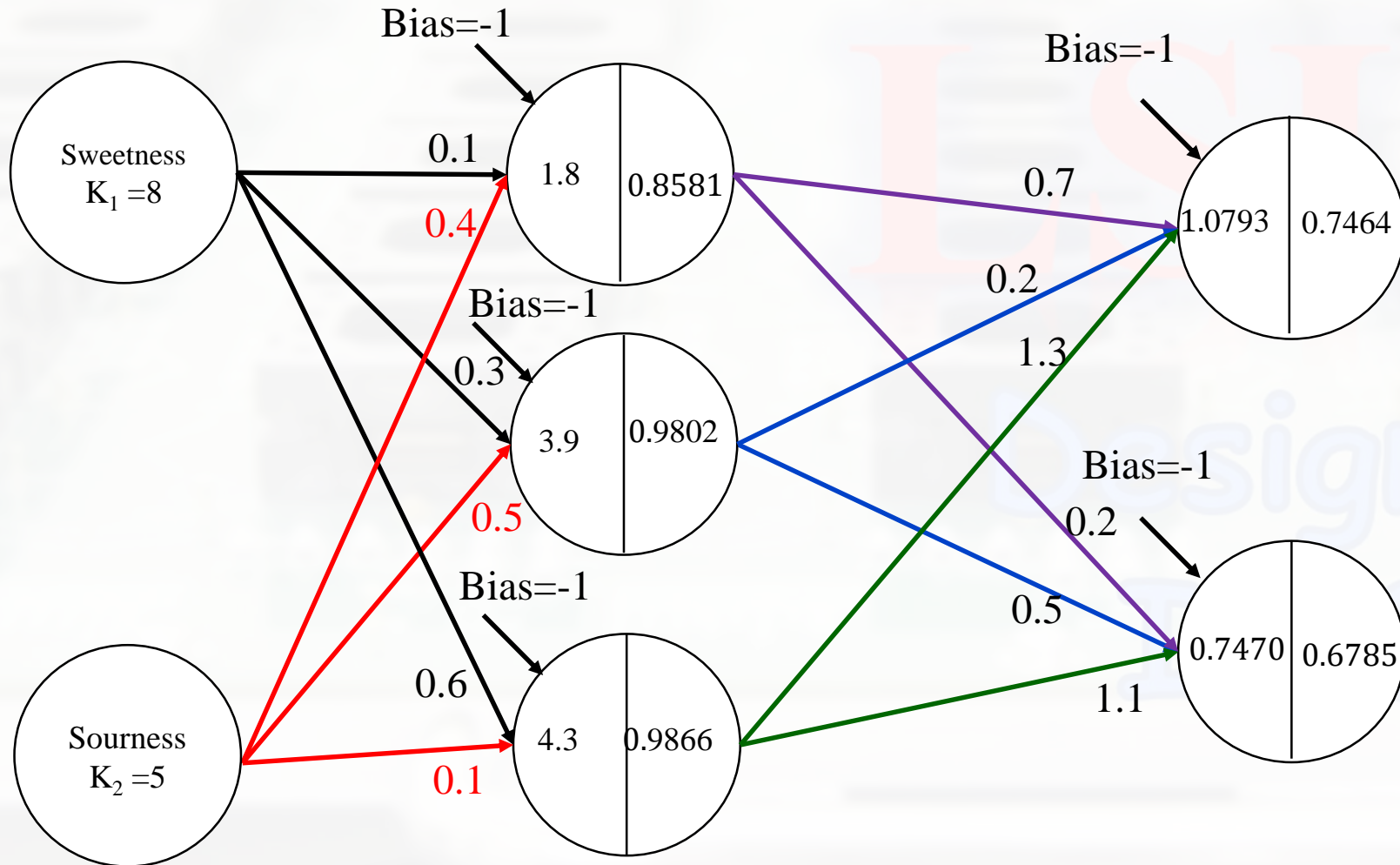
Score (n=1) : [8,8] ($\delta_1^2[1]$, $\delta_2^2[1]$, $\delta_3^2[1]$, $\delta_1^3[1]$, $\delta_2^3[1]$)



$$\therefore \delta_1^3 = (a_1^3 - t_1)a'(z_1^3)$$

$$\therefore \delta_1^2 = (\delta_1^3 w_{11}^3 + \delta_2^3 w_{21}^3)a'(z_1^2)$$

Score (n=2) : [8,5] (z2,a2,z3,a3)



$i = 1, 2, 3$

$$\therefore z_i^2 = w_{1i}^2 k_1 + w_{2i}^2 k_2 + b_i^2$$

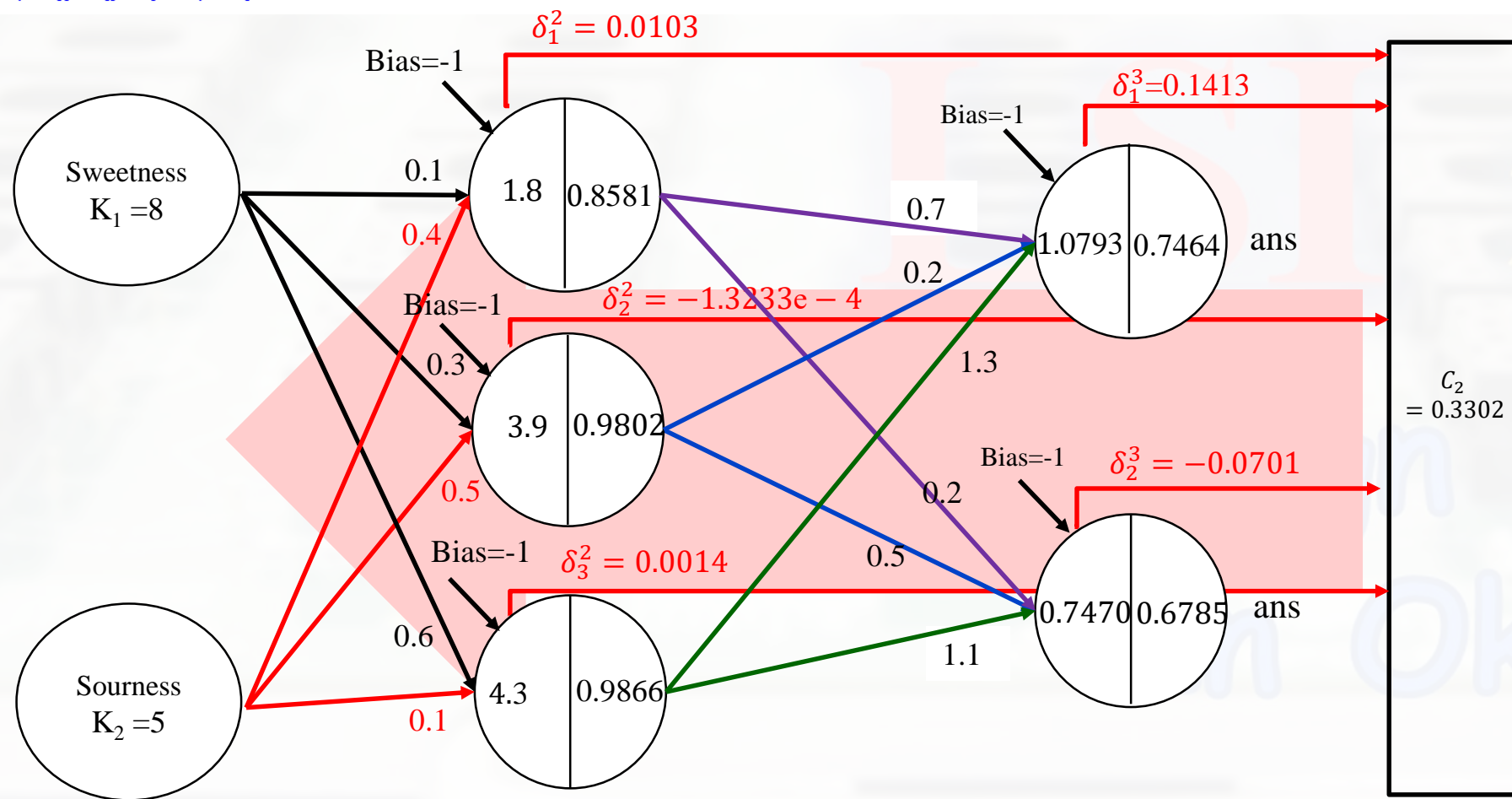
$$\therefore a_i^2 = a(z_i^2) = \frac{1}{1 + e^{-z_i^2}}$$

$i = 1, 2$

$$\therefore z_i^3 = w_{1i}^3 a_i^2 + w_{2i}^2 a_i^2 + w_{3i}^2 a_i^2 + b_i^3$$

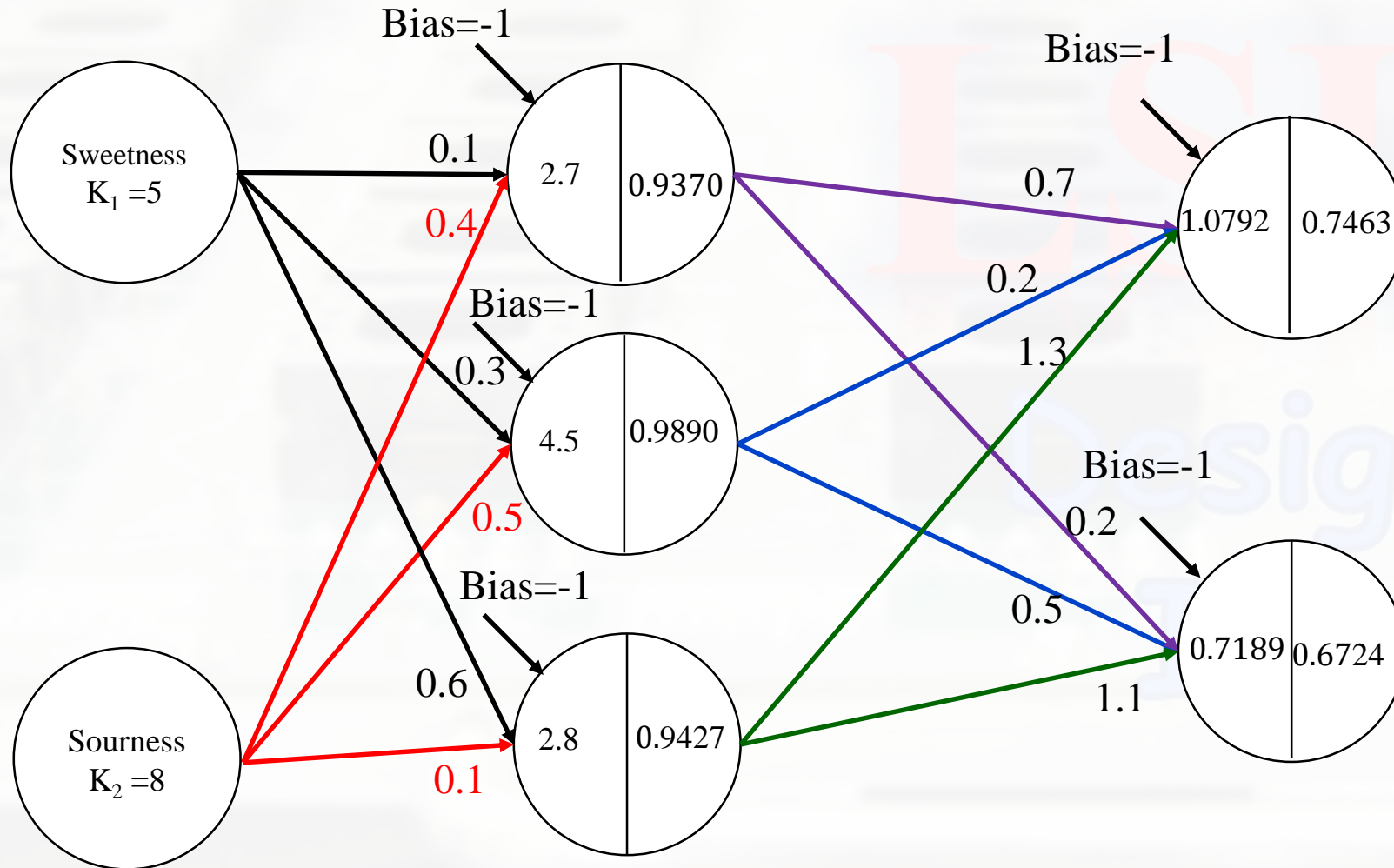
$$\therefore a_i^3 = a(z_i^3) = \frac{1}{1 + e^{-z_i^3}}$$

Score (n=2) : [8,5] ($\delta_1^2[2]$, $\delta_2^2[2]$, $\delta_3^2[2]$, $\delta_1^3[2]$, $\delta_2^3[2]$)



$$\begin{aligned} \therefore \delta_1^3 &= (a_1^3 - t_1)a'(z_1^3) \\ \therefore \delta_1^2 &= (\delta_1^3 w_{11}^3 + \delta_2^3 w_{21}^3)a'(z_1^2) \end{aligned}$$

Score (n=3) : [5,8] (z2,a2,z3,a3)



$i = 1, 2, 3$

$$\therefore z_i^2 = w_{1i}^2 k_1 + w_{2i}^2 k_2 + b_i^2$$

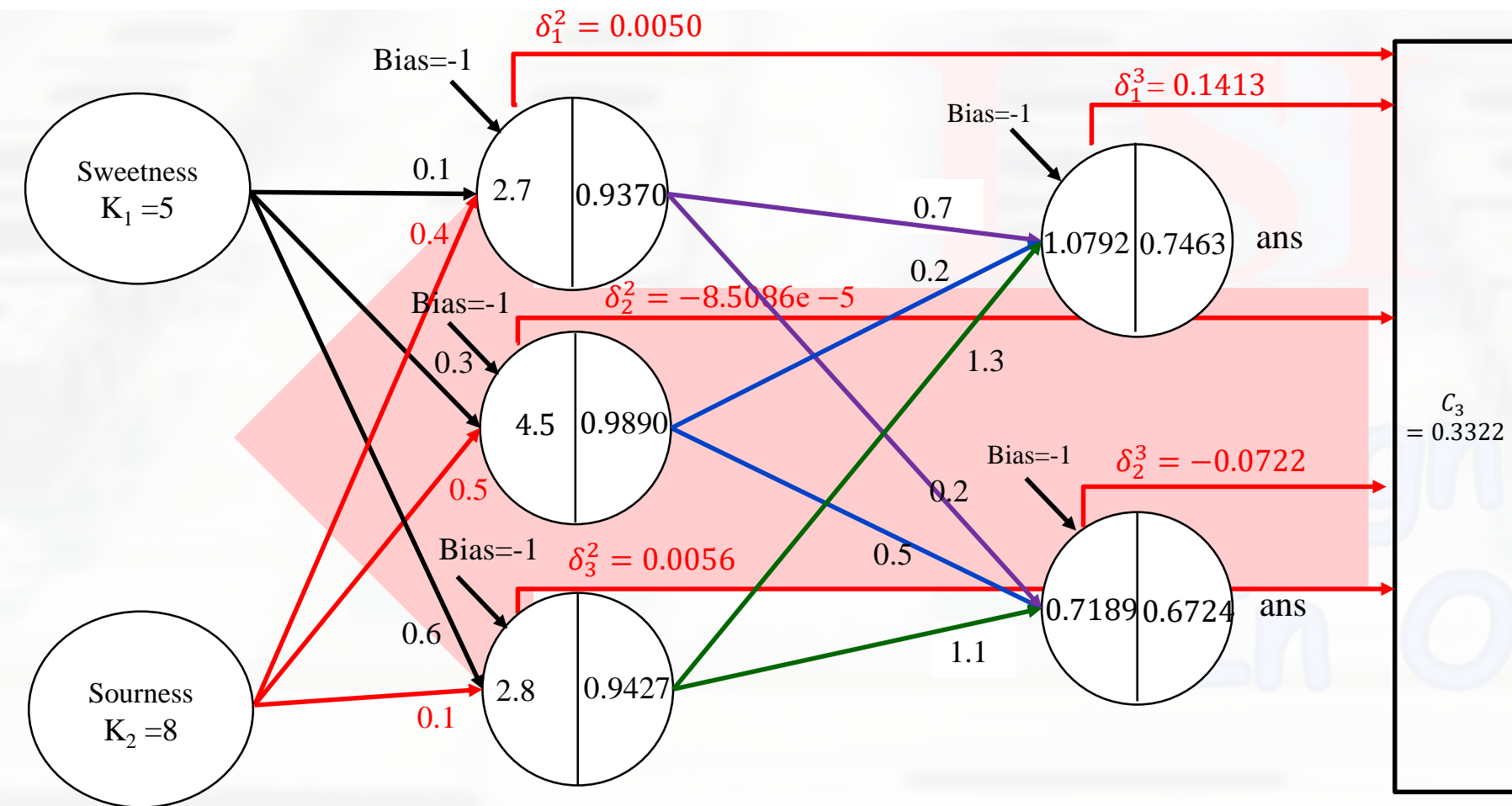
$$\therefore a_i^2 = a(z_i^2) = \frac{1}{1 + e^{-z_i^2}}$$

$i = 1, 2$

$$\therefore z_i^3 = w_{1i}^3 a_i^2 + w_{2i}^2 a_i^2 + w_{3i}^2 a_i^2 + b_i^3$$

$$\therefore a_i^3 = a(z_i^3) = \frac{1}{1 + e^{-z_i^3}}$$

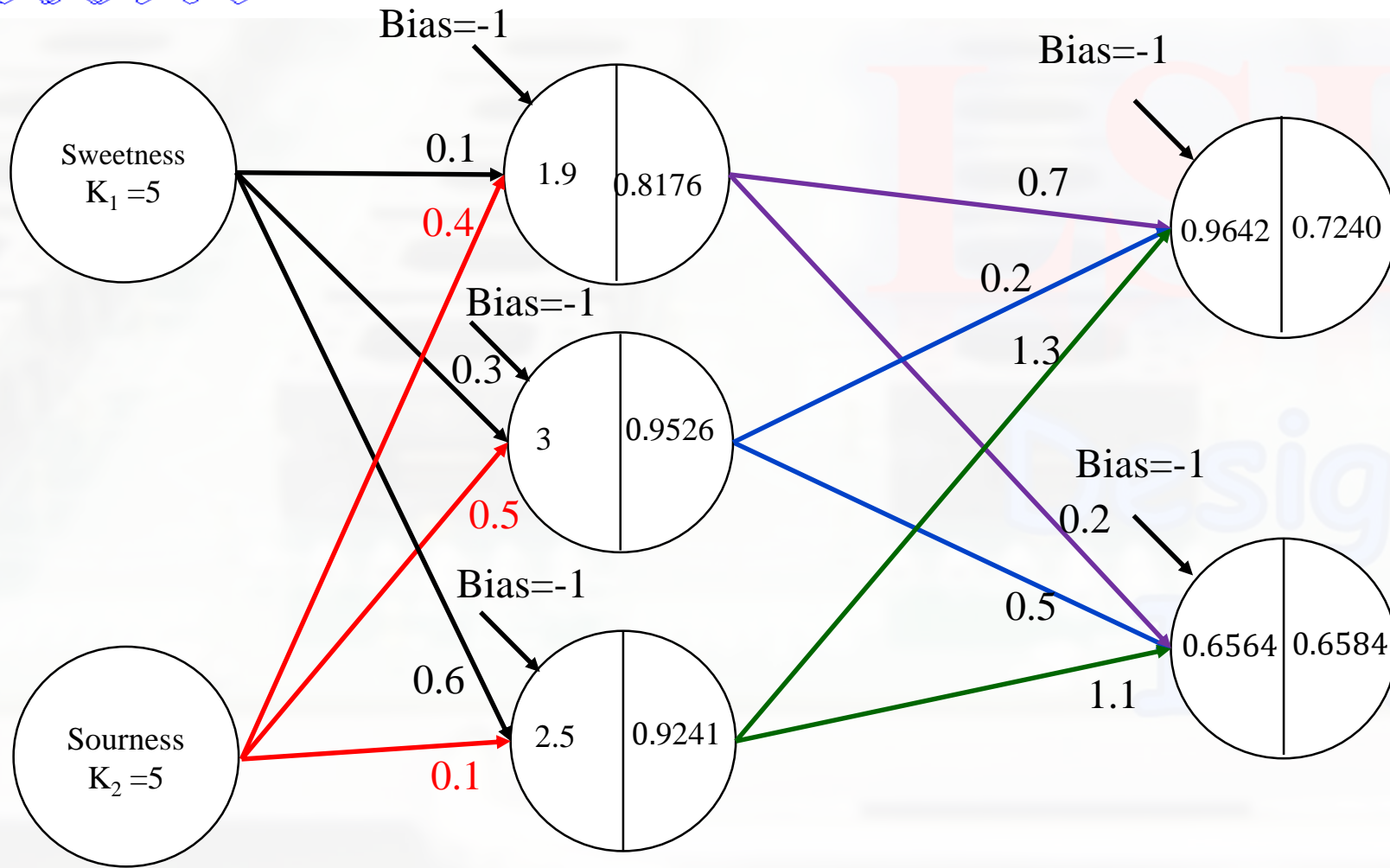
Score (n=3) : [5,8] ($\delta_1^2[3]$, $\delta_2^2[3]$, $\delta_3^2[3]$, $\delta_1^3[3]$, $\delta_2^3[3]$)



$$\therefore \delta_1^3 = (a_1^3 - t_1)a'(z_1^3)$$

$$\therefore \delta_1^2 = (\delta_1^3 w_{11}^3 + \delta_2^3 w_{21}^3)a'(z_1^2)$$

Score (n=4) : [5,5] (z2,a2,z3,a3)



$i = 1, 2, 3$

$$\therefore z_i^2 = w_{1i}^2 k_1 + w_{2i}^2 k_2 + b_i^2$$

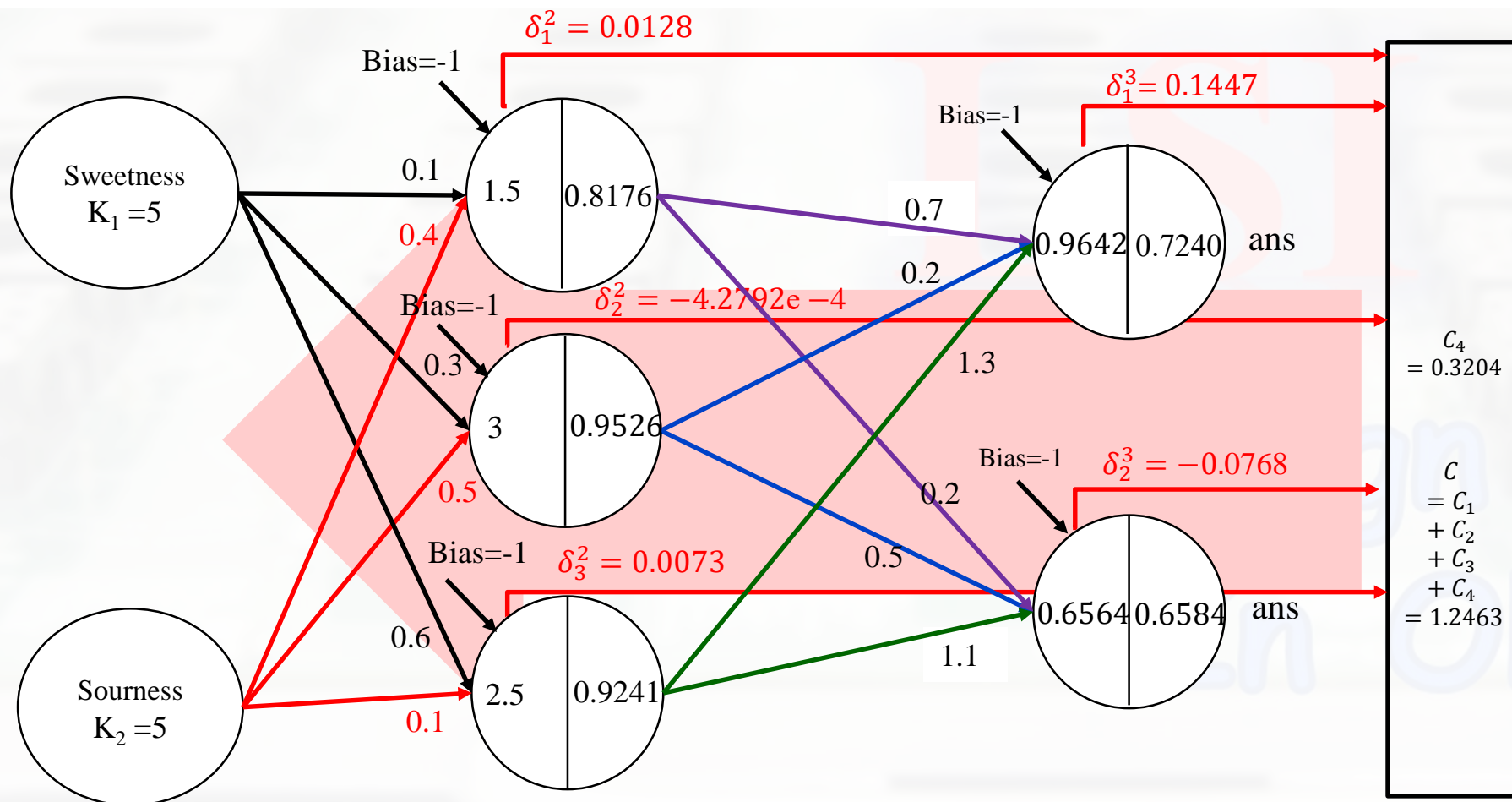
$$\therefore a_i^2 = a(z_i^2) = \frac{1}{1 + e^{-z_i^2}}$$

$i = 1, 2$

$$\therefore z_i^3 = w_{1i}^3 a_i^2 + w_{2i}^3 a_i^2 + w_{3i}^3 a_i^2 + b_i^3$$

$$\therefore a_i^3 = a(z_i^3) = \frac{1}{1 + e^{-z_i^3}}$$

Score (n=4) : [5,5] ($\delta_1^2[4]$, $\delta_2^2[4]$, $\delta_3^2[4]$, $\delta_1^3[4]$, $\delta_2^3[4]$)



$$\therefore \delta_1^3 = (a_1^3 - t_1)a'(z_1^3)$$

$$\therefore \delta_1^2 = (\delta_1^3 w_{11}^3 + \delta_2^3 w_{21}^3)a'(z_1^2)$$

Calculation to renew weight and bias (example 1/2)

To renew the weight and bias, we used the value of cost function that already calculated when we inserted the input. From that, we can calculate it using δ_j^i .

- For example, let's calculate how much weight of first hidden unit to first output unit (w_{11}^3) need to change.

$$\Delta w_{11}^3 = -\eta \frac{\partial C}{\partial w_{11}^3}$$

Δw_{11}^3 is the difference of old w_{11}^3 and new w_{11}^3

$$\frac{\partial C}{\partial w_{11}^3} = \left(\frac{\partial C_1}{\partial w_{11}^3} + \frac{\partial C_2}{\partial w_{11}^3} + \frac{\partial C_3}{\partial w_{11}^3} + \frac{\partial C_4}{\partial w_{11}^3} \right)$$
$$= (\delta_1^3[1]a_1^2[1] + \delta_1^3[2]a_1^2[2] + \delta_1^3[3]a_1^2[3] + \delta_1^3[4]a_1^2[4])$$

- Now let's calculate the difference of old b_1^3 and new b_1^3

$$\Delta b_1^3 = -\eta \frac{\partial C}{\partial b_1^3}$$
$$\frac{\partial C}{\partial b_1^3} = \left(\frac{\partial C_1}{\partial b_1^3} + \frac{\partial C_2}{\partial b_1^3} + \frac{\partial C_3}{\partial b_1^3} + \frac{\partial C_4}{\partial b_1^3} \right)$$
$$= (\delta_1^3[1] + \delta_1^3[2] + \delta_1^3[3] + \delta_1^3[4])$$

[1] → first supervisor value (8,8)
[2] → first supervisor value (8,5)
[3] → first supervisor value (5,8)
[4] → first supervisor value (5,5)

Calculation to renew weight and bias (example 2/2)

- Now, let's calculate how much weight of first hidden unit to second output unit (w_{12}^3) need to change.

$$\Delta w_{12}^3 = -\eta \frac{\partial C}{\partial w_{12}^3} \quad \leftarrow \quad \Delta w_{12}^3 \text{ is the difference of old } w_{12}^3 \text{ and new } w_{12}^3$$

$$\frac{\partial C}{\partial w_{12}^3} = \left(\frac{\partial C_1}{\partial w_{12}^3} + \frac{\partial C_2}{\partial w_{12}^3} + \frac{\partial C_3}{\partial w_{12}^3} + \frac{\partial C_4}{\partial w_{12}^3} \right)$$

$$= (\delta_2^3[1]a_1^2[1] + \delta_2^3[2]a_1^2[2] + \delta_2^3[3]a_1^2[3] + \delta_2^3[4]a_1^2[4])$$

- Now, let's calculate how much weight of second hidden unit to first output unit (w_{21}^3) need to change.

$$\Delta w_{21}^3 = -\eta \frac{\partial C}{\partial w_{21}^3} \quad \leftarrow \quad \Delta w_{21}^3 \text{ is the difference of old } w_{21}^3 \text{ and new } w_{21}^3$$

$$\frac{\partial C}{\partial w_{21}^3} = \left(\frac{\partial C_1}{\partial w_{21}^3} + \frac{\partial C_2}{\partial w_{21}^3} + \frac{\partial C_3}{\partial w_{21}^3} + \frac{\partial C_4}{\partial w_{21}^3} \right)$$

$$= (\delta_1^3[1]a_2^2[1] + \delta_1^3[2]a_2^2[2] + \delta_1^3[3]a_2^2[3] + \delta_1^3[4]a_2^2[4])$$

- Now let's calculate the difference of old b_2^3 and new b_2^3

$$\Delta b_2^3 = -\eta \frac{\partial C}{\partial b_2^3}$$

$$\frac{\partial C}{\partial b_2^3} = \left(\frac{\partial C_1}{\partial b_2^3} + \frac{\partial C_2}{\partial b_2^3} + \frac{\partial C_3}{\partial b_2^3} + \frac{\partial C_4}{\partial b_2^3} \right)$$

$$= (\delta_2^3[1] + \delta_2^3[2] + \delta_2^3[3] + \delta_2^3[4])$$

[1] → first supervisor value (8,8)
 [2] → first supervisor value (8,5)
 [3] → first supervisor value (5,8)
 [4] → first supervisor value (5,5)

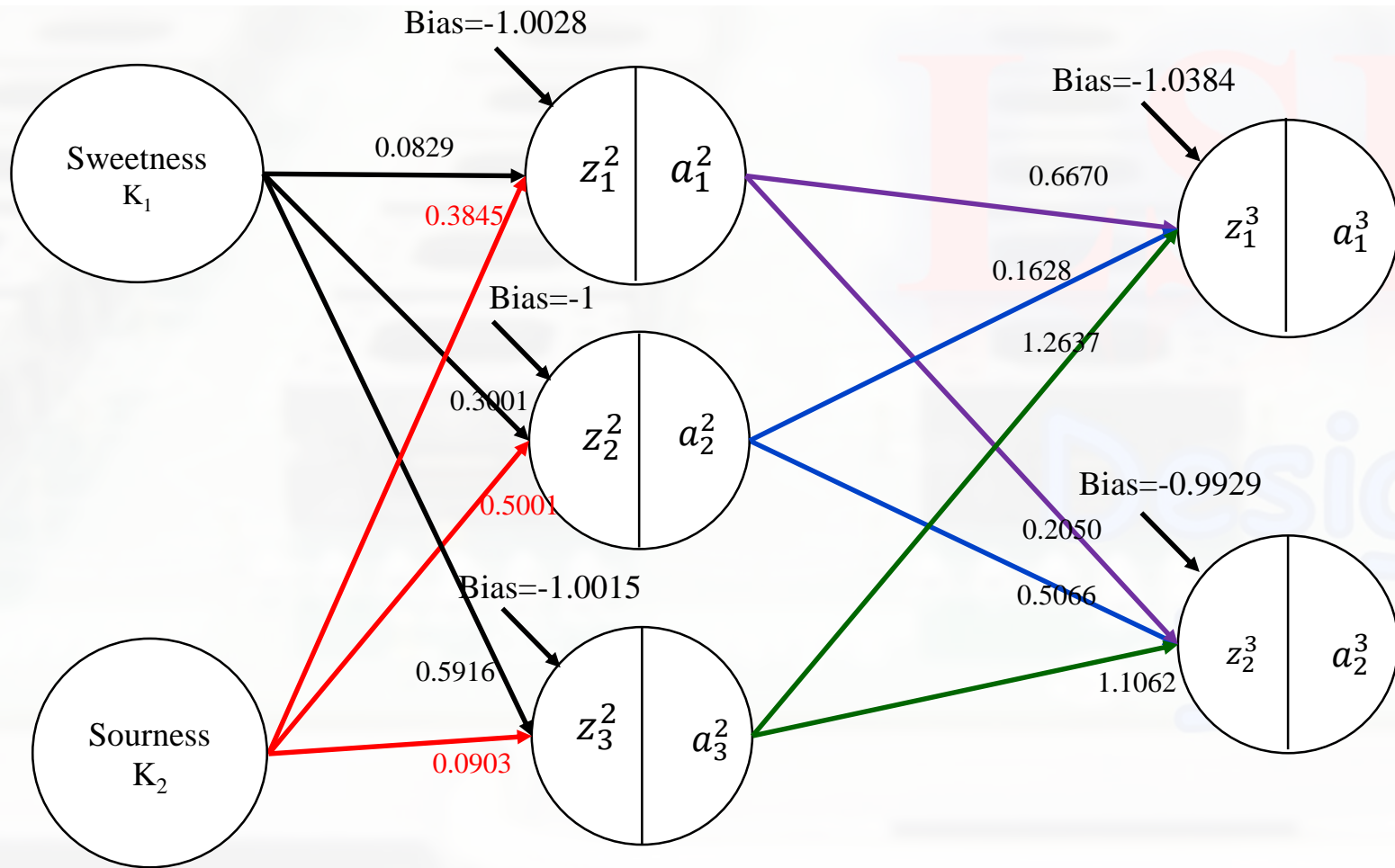


Calculation of Cost Func.

Input value $[K_1[n], K_2[n]]$	Supervisor $[t_1[n], t_2[n]]$	Output value $[a_3^1[n], a_3^2[n]]$	Square Error C_n	Cost func. C
[8,8]	[1,0]	[0.76, 0.69]	0.26	1.246
[8,5]	[0,1]	[0.75, 0.68]	0.33	
[5,8]	[0,1]	[0.75, 0.67]	0.33	
[5,5]	[0,1]	[0.72, 0.66]	0.32	

From the above result, we can see that the output value is far from the state value that we are finding. So, to obtain a suitable output value, we use Back Propagation

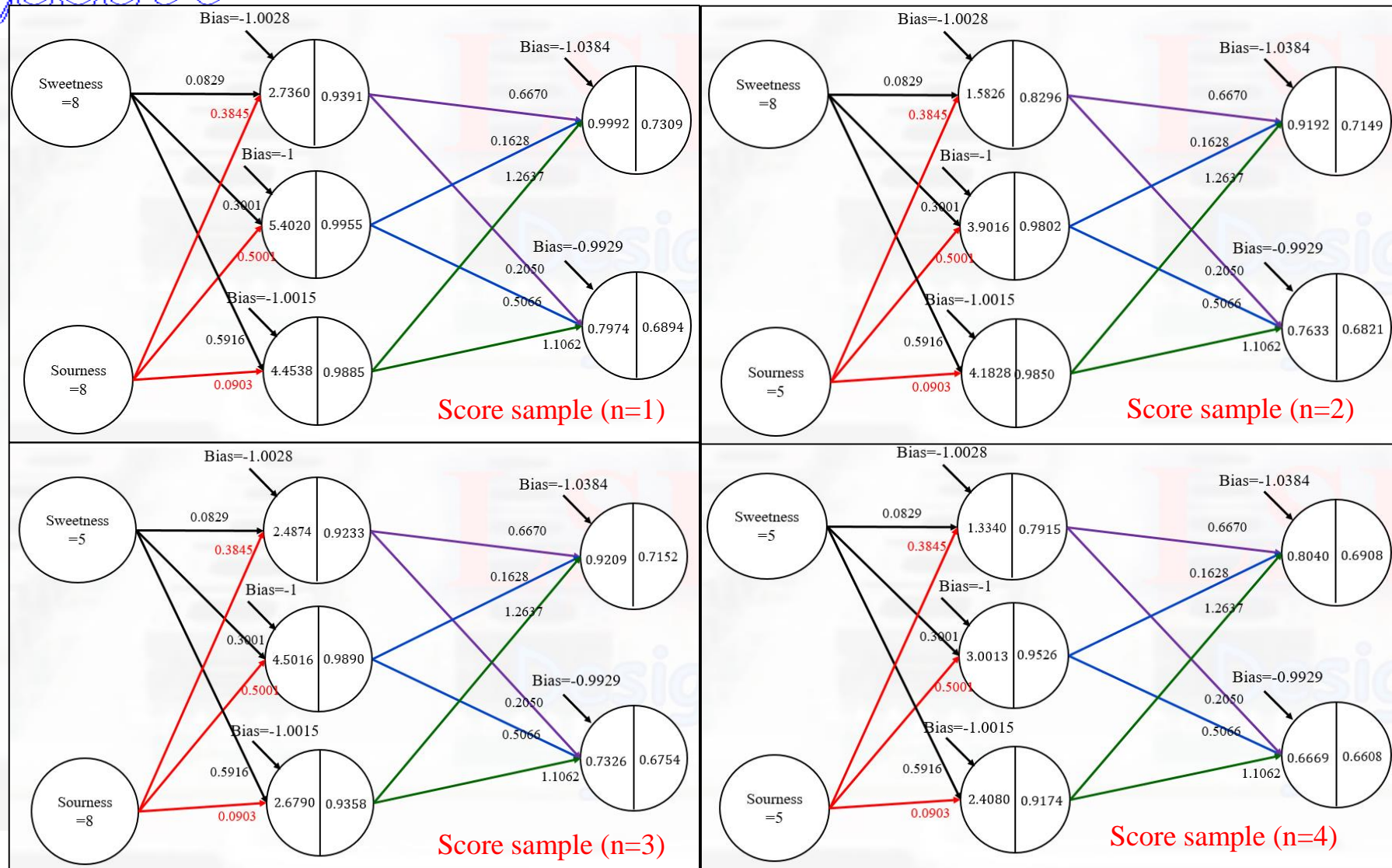
New parameter(after 1st renewal)



- $(new)w_{ij}^3 = (old)w_{ij}^3 + \Delta w_{ij}^3$
- $(new)w_{ij}^2 = (old)w_{ij}^2 + \Delta w_{ij}^2$
- $(new)b_i^3 = (old)b_i^3 + \Delta b_i^3$
- $(new)b_{ij}^2 = (old)b_{ij}^2 + \Delta b_{ij}^2$

$\therefore \eta = 0.1$

Score result (after 1st renewal)

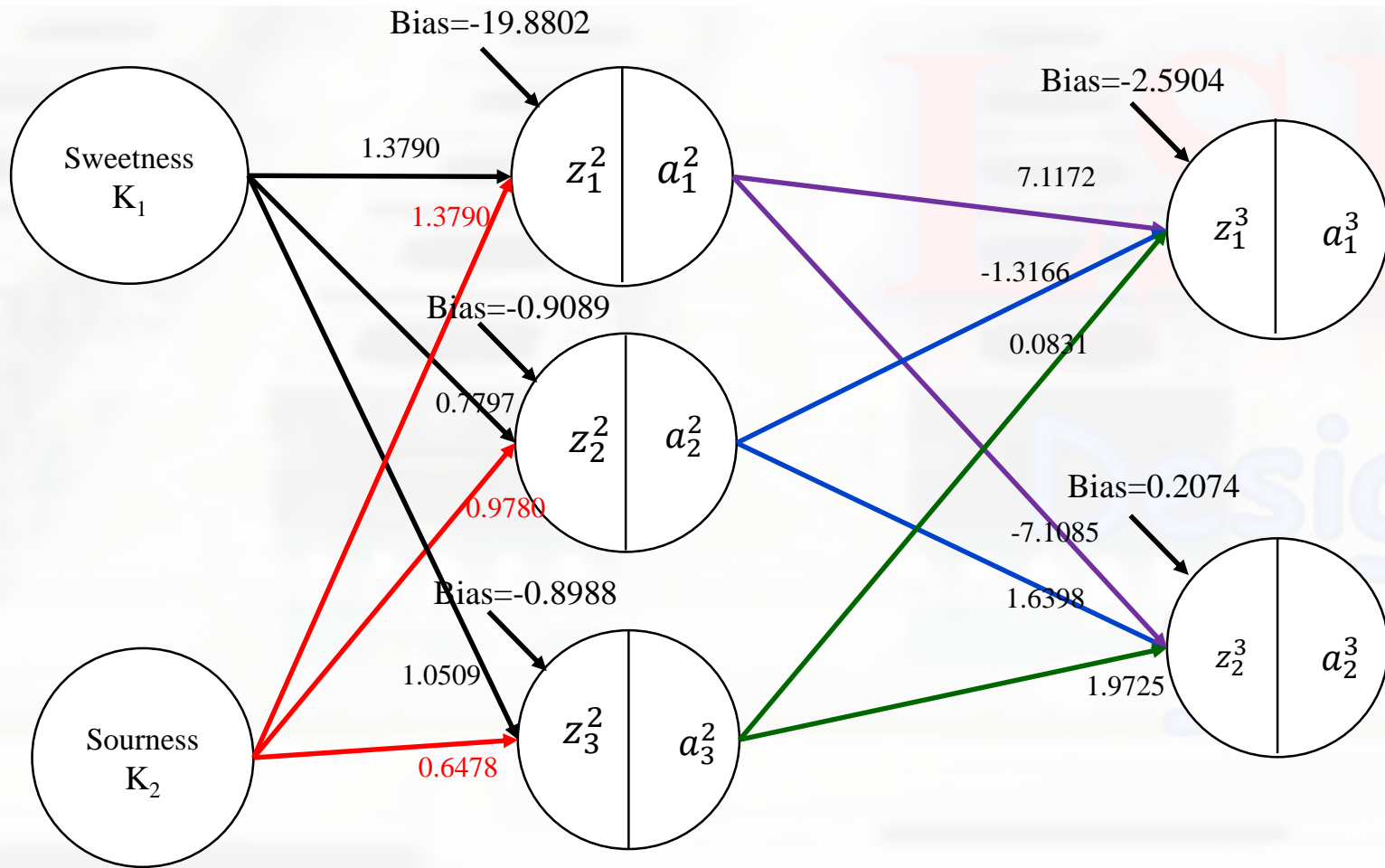




Calculation of Cost Func. (after 1st renewal)

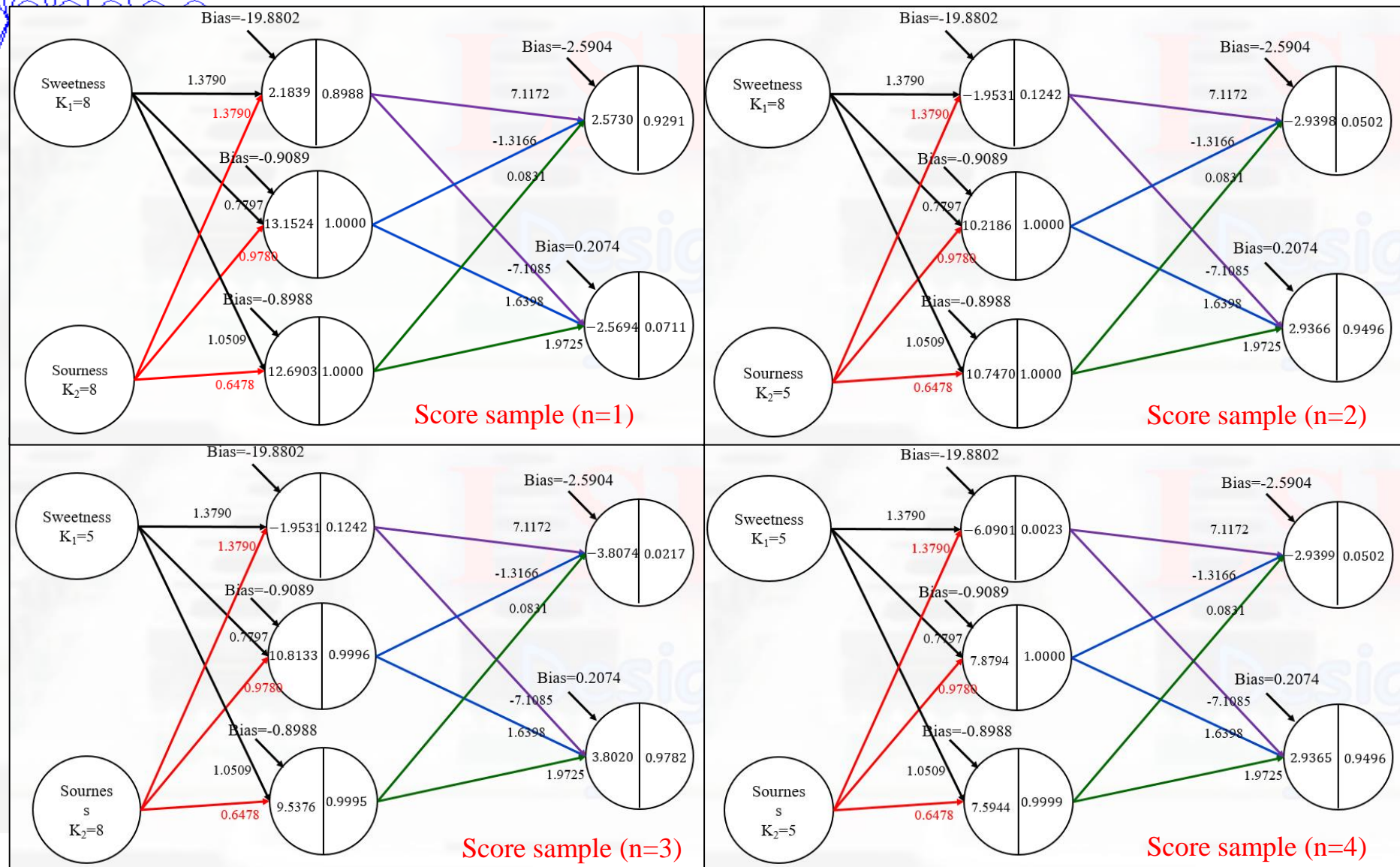
Input value $[K_1[n], K_2[n]]$	Supervisor $[t_1[n], t_2[n]]$	Output value $[a_3^1[n], a_3^2[n]]$	Square Error C_n	Cost func. C
[8,8]	[1,0]	[0.7309, 0.6894]	0.2739	1.1845
[8,5]	[0,1]	[0.7149, 0.6821]	0.3061	
[5,8]	[0,1]	[0.7152, 0.6754]	0.3085	
[5,5]	[0,1]	[0.6908, 0.6608]	0.2961	

New parameter(after 10000th renewal)



- $(new)w_{ij}^3 = (old)w_{ij}^3 + \Delta w_{ij}^3$
 - $(new)w_{ij}^2 = (old)w_{ij}^2 + \Delta w_{ij}^2$
 - $(new)b_i^3 = (old)b_i^3 + \Delta b_i^3$
 - $(new)b_{ij}^2 = (old)b_{ij}^2 + \Delta b_{ij}^2$
- $\therefore \eta = 0.1$

Score result (after 10000th renewal)



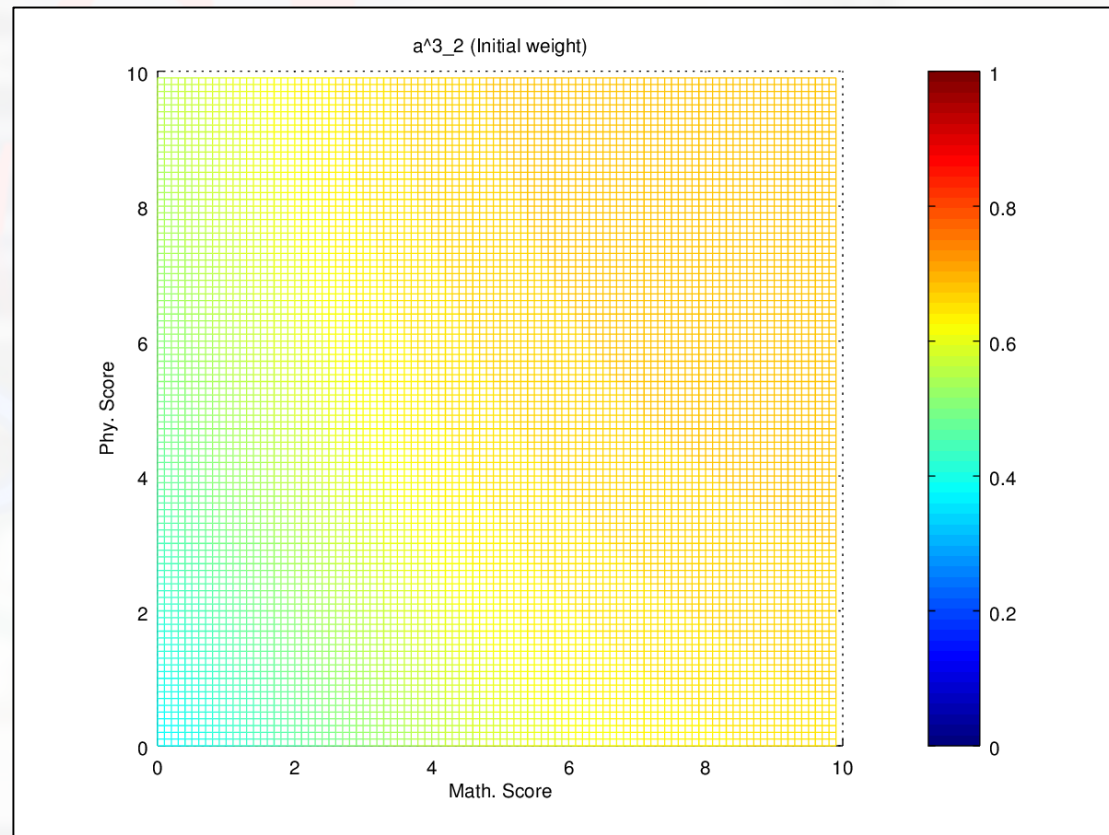
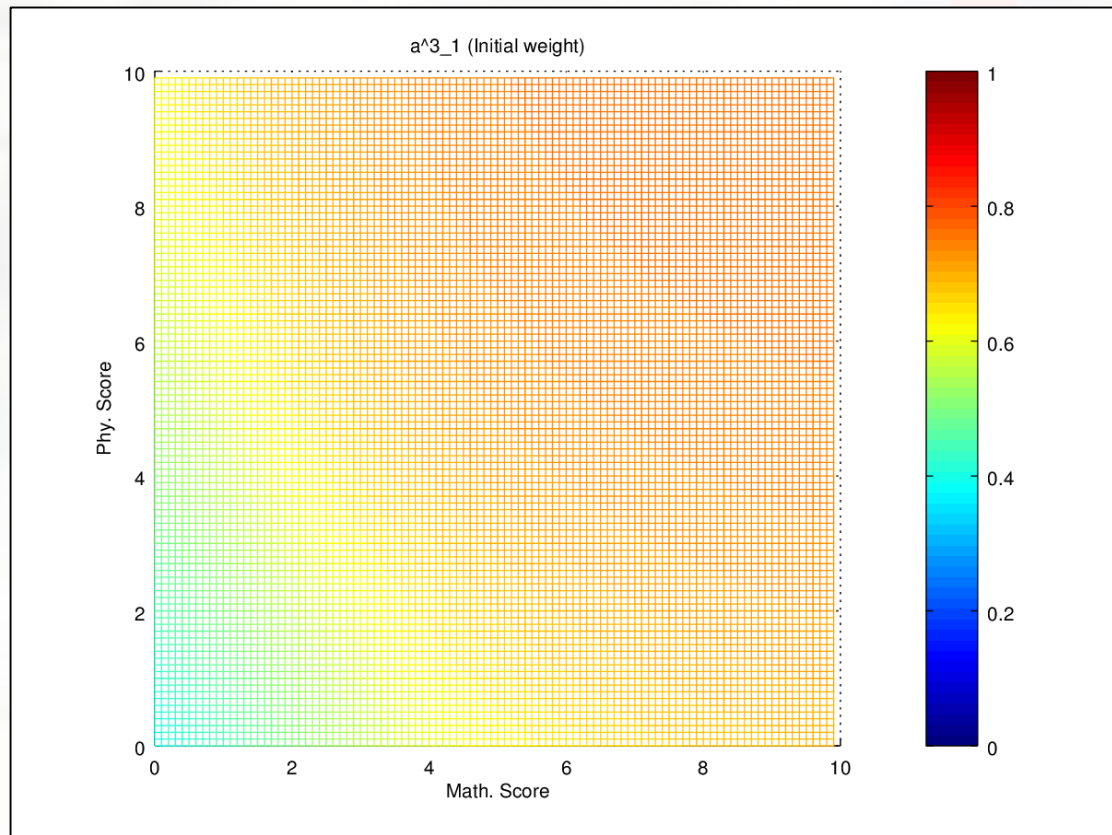


Calculation of Cost Func. (after 10000th renewal)

Input value $[K_1[n], K_2[n]]$	Supervisor $[t_1[n], t_2[n]]$	Output value $[a_3^1[n], a_3^2[n]]$	Square Error C_n	Cost func. C
[8,8]	[1,0]	[0.9291, 0.0711]	0.0050	0.0105
[8,5]	[0,1]	[0.0502, 0.9496]	0.0025	
[5,8]	[0,1]	[0.502, 0.9496]	0.0025	
[5,5]	[0,1]	[0.0217, 0.9782]	0.00047	

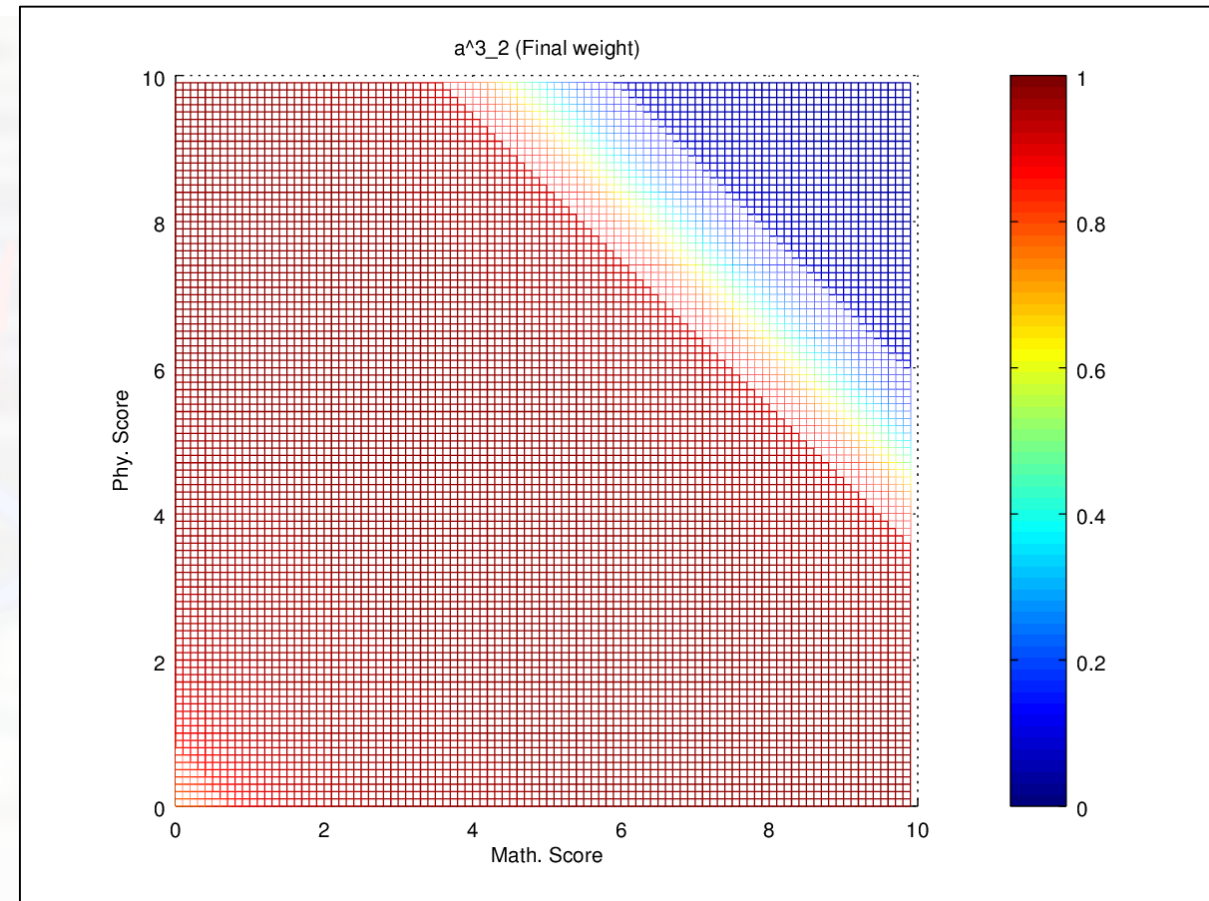
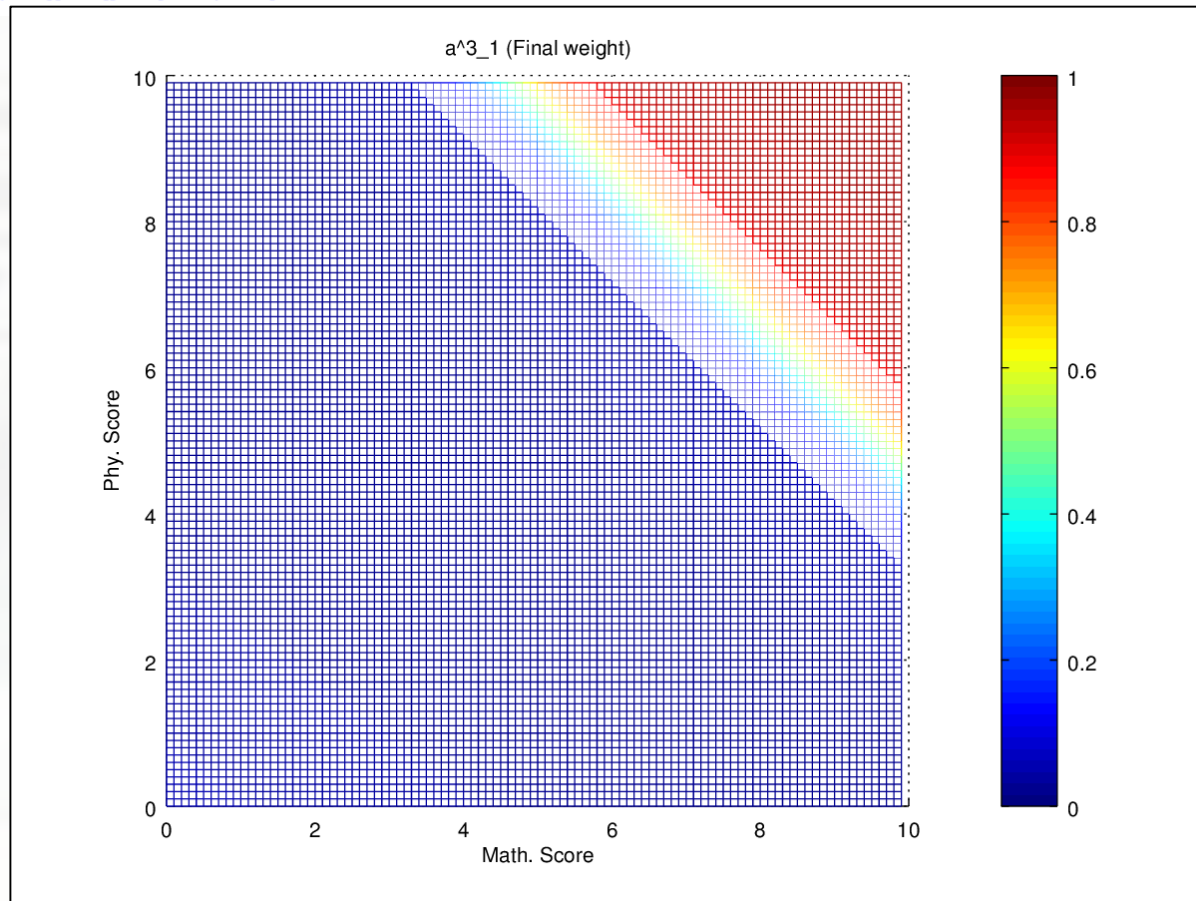
From above result, we can see the value of Cost function has decreased compared to the beginning of our calculation

Initial passing range graph



These graph shows the initial region of the man's taste

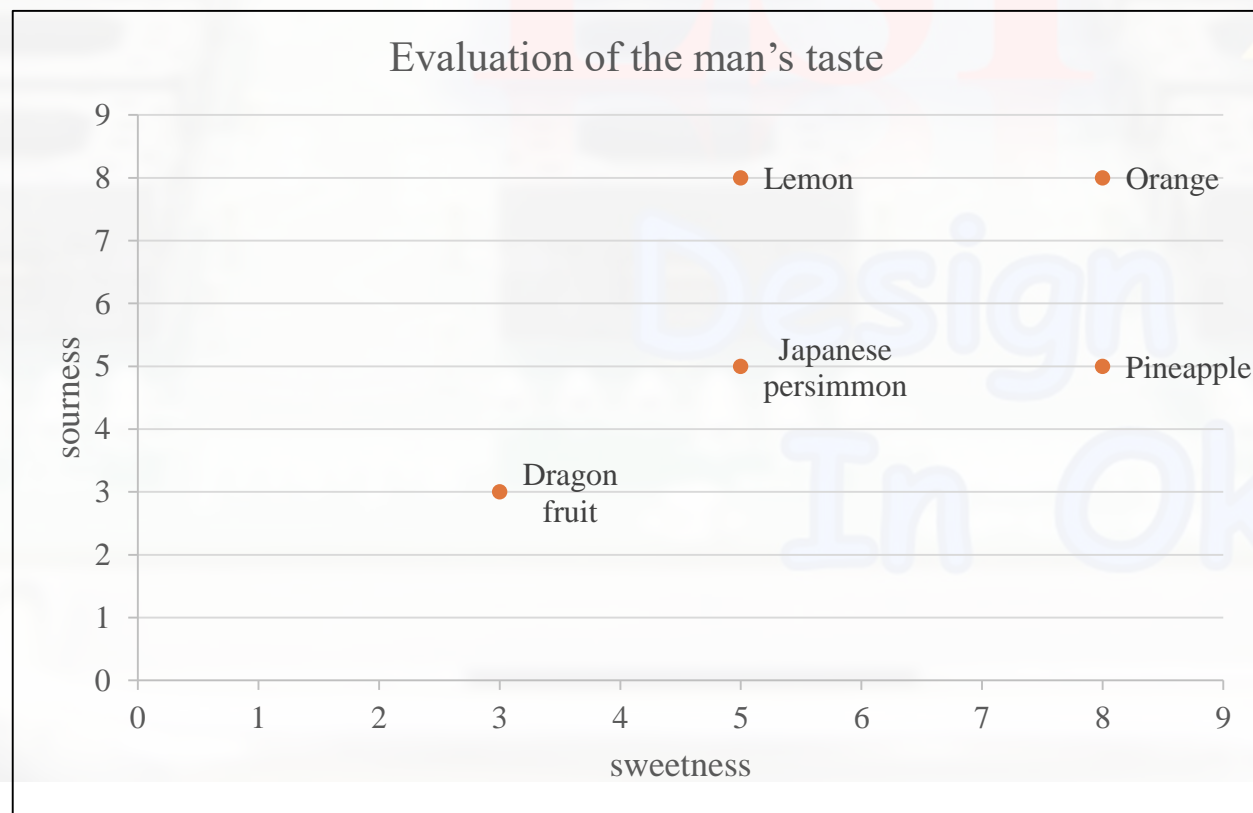
Final passing range graph



These graph shows the final region of the man's taste.

Task

- Now let's measure the man's taste when he eats another fruit which is dragon fruit
- He evaluate the sweetness and sourness of the dragon fruit as 3 both [$k_1=3$, $k_2=3$]
- The he said he likes the dragon fruit [$t_1=1$, $t_2=0$]





Conclusion

- After some calculation, we can see that the value of weight and bias is changing simultaneously with the input value. This calculation is repeated until the different value between output and supervisor value become smaller.
- From the graph, we can see the region of the man's taste is changing from unclear area to the area that we can estimated the result.