Neural Network and optimization of calculation

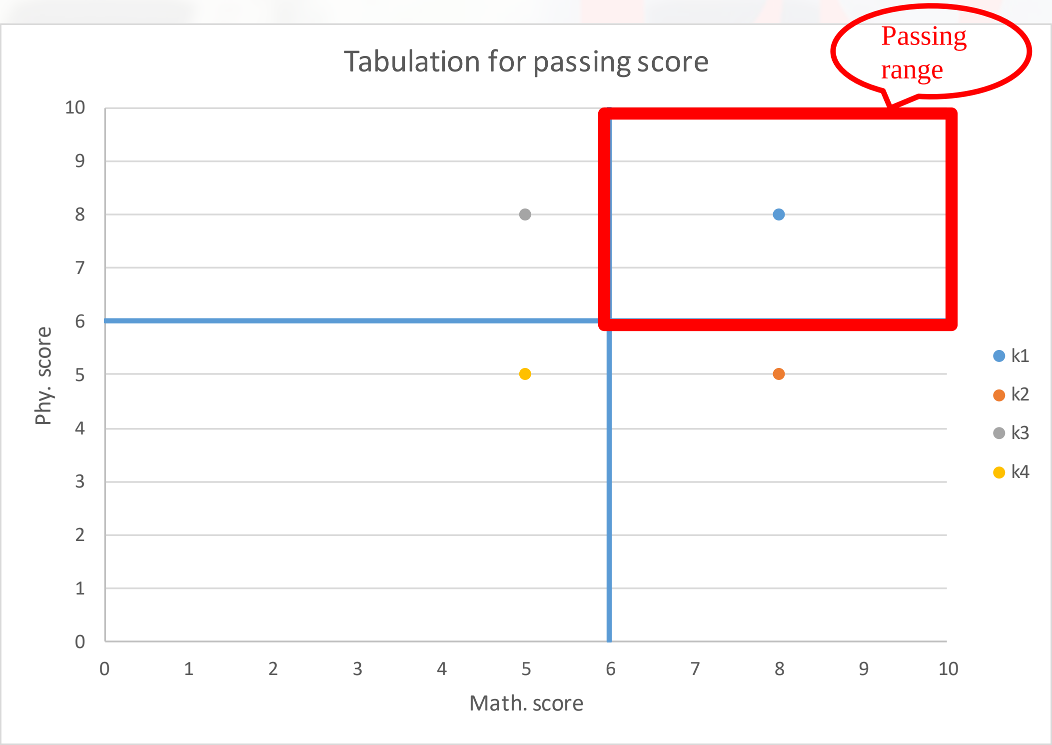
LSI Design Contest

Introduction

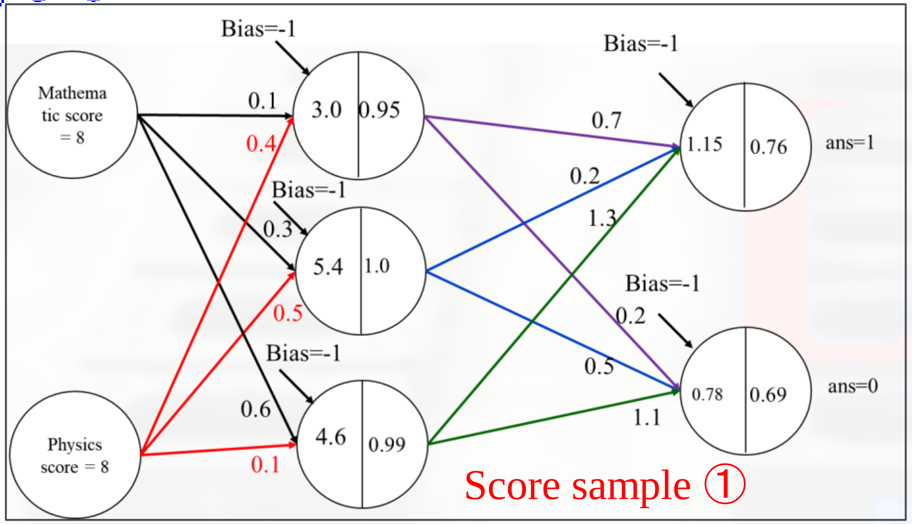
- This is an example of Neural network calculation.
- The neural network structure used here is 3-layer structure.
- It consist of 2 input units, 3 hidden units and 2 output units.

State condition

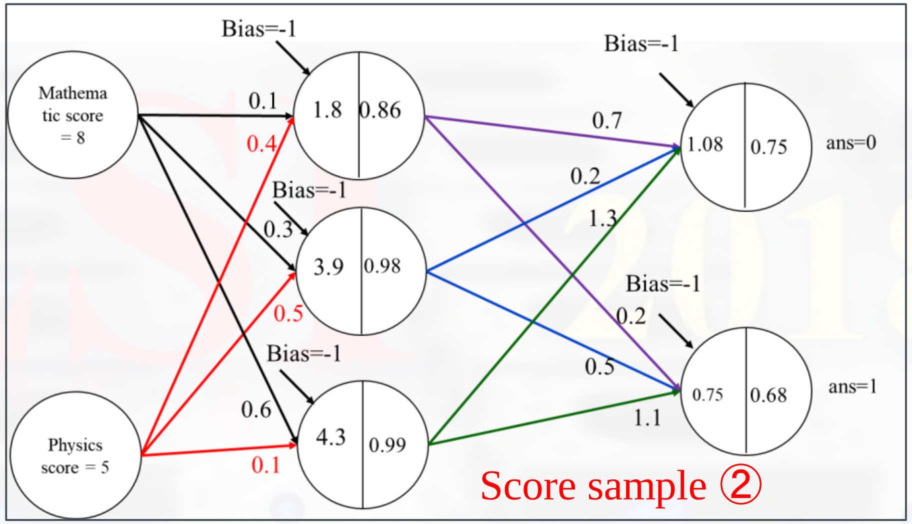
- Students need to pass both Mathematics and Physics examination in order to pass **[1,0]** the spring semester
- Passing mark for each exam is 6 and total mark is 10
- Students who only manage to pass only one exam is considered as failed **[0,1]**



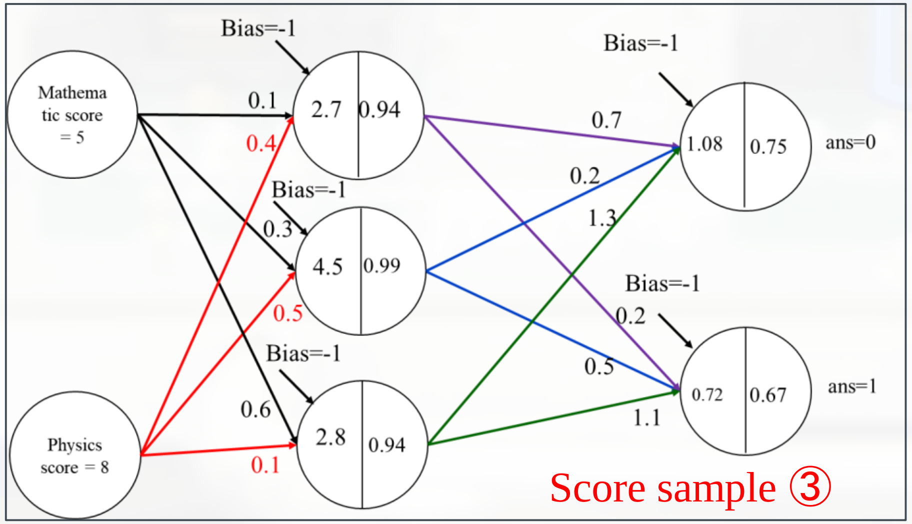
Score sample



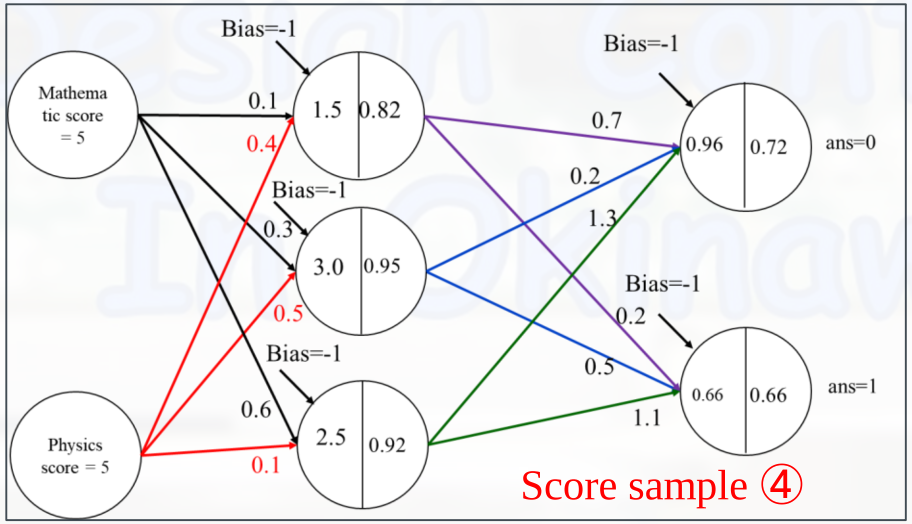
Score sample ①



Score sample ②



Score sample ③



Score sample ④

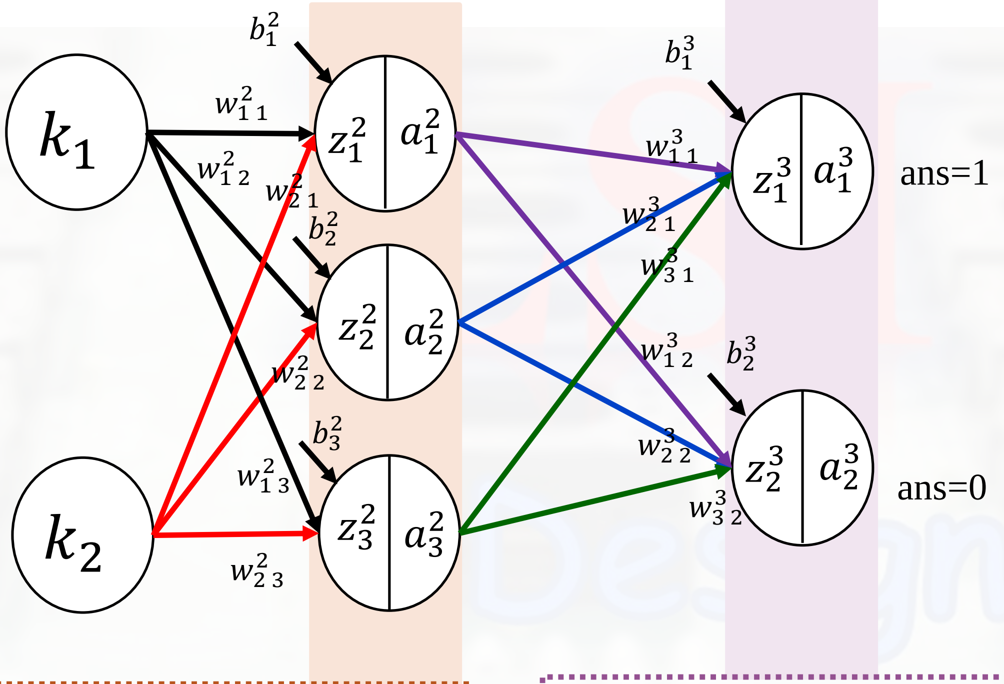
Rough Results

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

Students i	Math. Score K_1^i	Physics Score K_2^i	Supervisor $[t_1, t_2]$	Results	Output value $[a_1^3, a_2^3]$
A	8	8	[1,0]	Pass	[0.76, 0.69]
B	8	5	[0,1]	Fail	[0.75, 0.68]
C	5	8	[0,1]	Fail	[0.75, 0.67]
D	5	5	[0,1]	Fail	[0.72, 0.66]

- $z_1^2 = w_{1\,1}^2 k_1 + w_{2\,1}^2 k_2 + b_1^2$
- $square\ error$: $C_i = \frac{1}{2}\{(a_1^3[i] - t_1)^2 + (a_2^3[i] - t_2)^2\}$
- cost function: $C = C_1 + C_2 + \cdots + C_i + \cdots$
- $\delta_1^3 = (a_1^3 - t_1)a'(z_1^3)$
- $\delta_1^2 = (\delta_1^3 w_{1\,1}^3 + \delta_2^3 w_{2\,1}^3 + \cdots)a'(z_1^2)$
- $\frac{\partial C}{\partial w_{ij}^m} = \delta_j^m a_j^{m-1}$,
- $\frac{\partial C}{\partial b_j^m} = \delta_j^m$

Equation of z and a



$$z_i^2 = w^2_{1\,i}k_1 + w^2_{2\,i}k_2 + b_i^2$$

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

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

$$z_i^3 = w^3_{1\,i}a_i^2 + w^2_{2\,i}a_i^2 + w^2_{3\,i}a_i^2 + b_i^3$$

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

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

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

$$(\Delta w_{1\,1}^{2} \ldots, \Delta w_{1\,1}^{3} \ldots, \Delta \mathrm{b}_{1}^{2} \ldots, \Delta \mathrm{b}_{1}^{3} \ldots) = -\eta\left(\frac{\partial C}{\partial w_{1\,1}^{2}} \ldots, \frac{\partial C}{\partial w_{1\,1}^{3}} \ldots, \frac{\partial C}{\partial \mathrm{b}_{1}^{2}} \ldots, \frac{\partial C}{\partial \mathrm{b}_{1}^{3}} \ldots\right)$$

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 ,

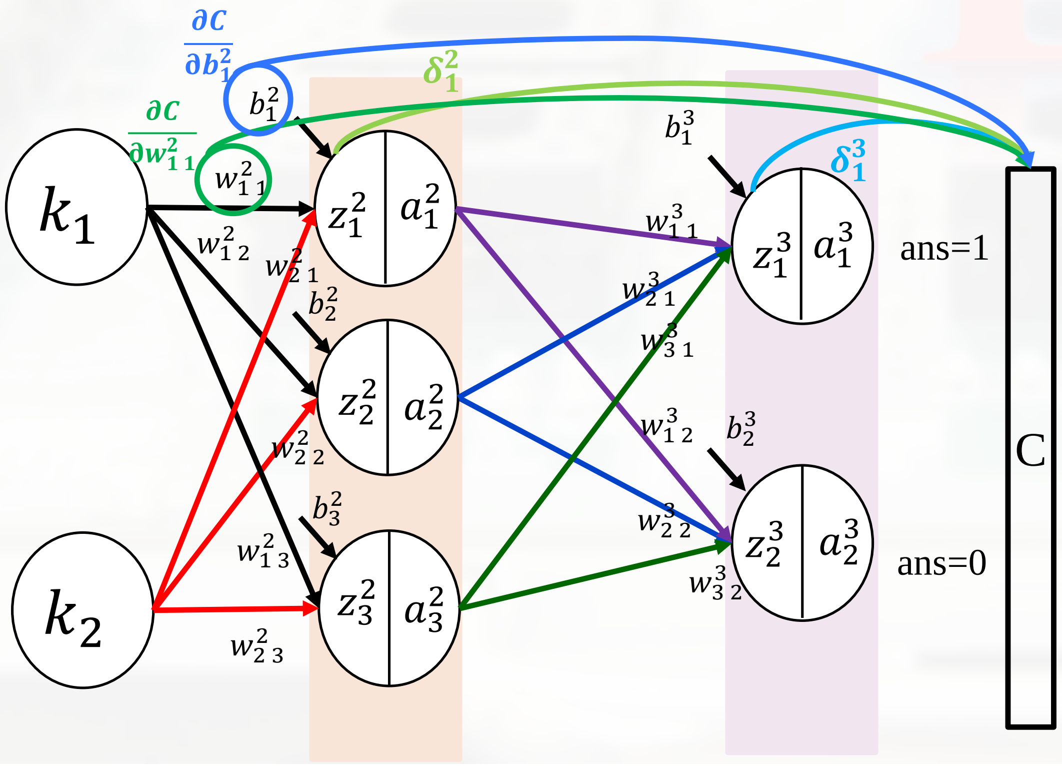
numbers of parameter is 10*10 + 10(bias) + 10*3 + 3(bias) =143.

⇒ 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.



$$\frac{\partial C}{\partial w_{1\,1}^2} = \frac{\partial C}{\partial z_1^2}\frac{\partial z_1^2}{\partial w_{1\,1}^2} = \frac{\partial C}{\partial z_1^2}k_1$$

$$(\because z_1^2 = w_{1\,1}^2 k_1 + w_{1\,2}^2 k_2 + w_{1\,3}^2 k_3 + \cdots + b_1^2)$$

$$\text{define } \frac{\partial C}{\partial z_1^2} = \delta_1^2 \rightarrow \frac{\partial C}{\partial w_{1\,1}^2} = \delta_1^2 k_1$$

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

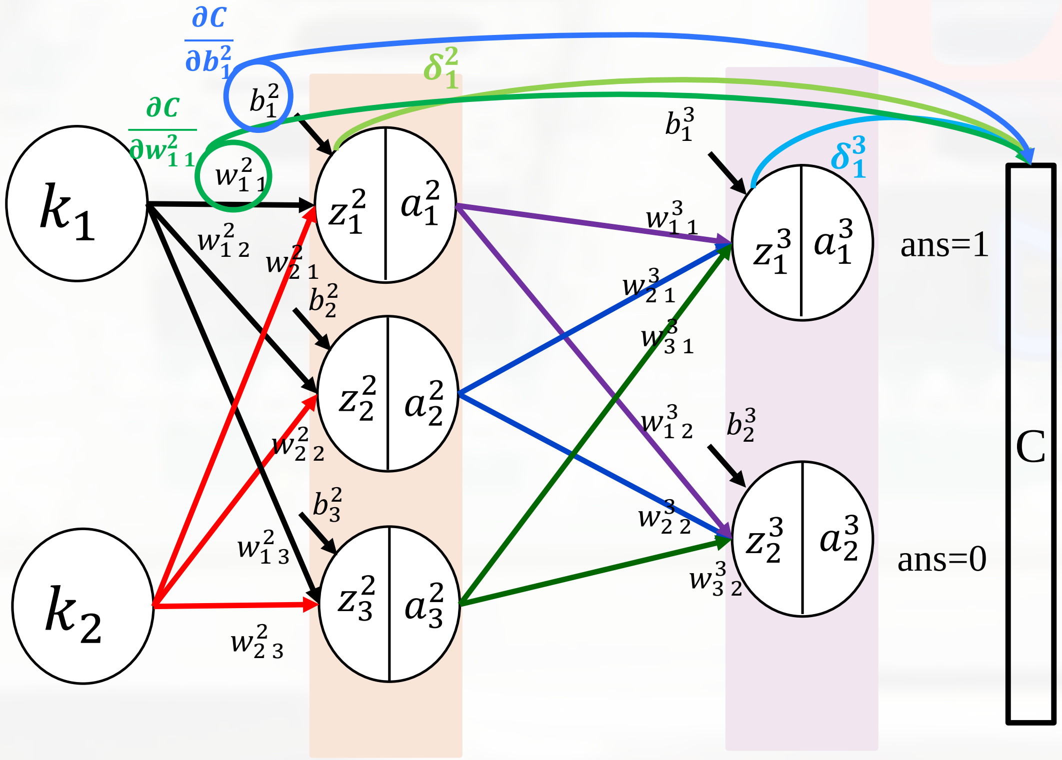
$$\therefore \frac{\partial C}{\partial w_{i\,j}^m} = \delta_j^m a_j^{m-1}, \frac{\partial C}{\partial b_j^m} = \delta_j^m \quad \text{(m=2 or 3)}$$

$$(a_j^1 = k_j)$$

Back propagation

- Back propagation is optimization of calculation.

  calculate all parameter to return output to input.



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

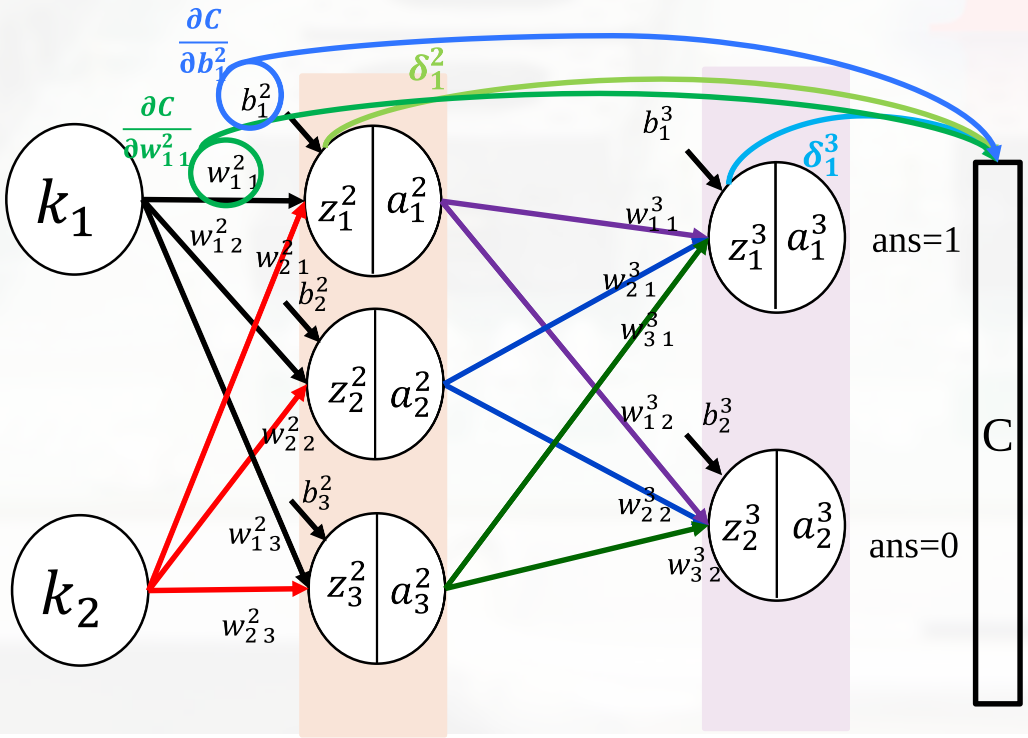
$$\delta_1^2 = (\delta_1^3 w_{1\,1}^3 + \delta_2^3 w_{2\,1}^3 + \cdots)a'(z_1^2)$$

$$\frac{\partial C}{\partial w_{1\,1}^2} = \delta_1^2 k_1 \qquad \frac{\partial C}{\partial b_1^2} = \delta_1^2$$

Weight and bias is depend on output parameter.

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.

$$\delta^3_1 = \frac{\partial C}{\partial z^3_1} = \frac{\partial C}{\partial a^3_1}\frac{\partial a^3_1}{\partial z^3_1} = \frac{\partial C}{\partial a^3_1} a'(z^3_1)$$

$$\therefore \delta^3_1 = (a^3_1 - t_1)a'(z^3_1)$$

∵ C is made of the difference between t and a_3.

But hidden layer errors is more difficult to calculate

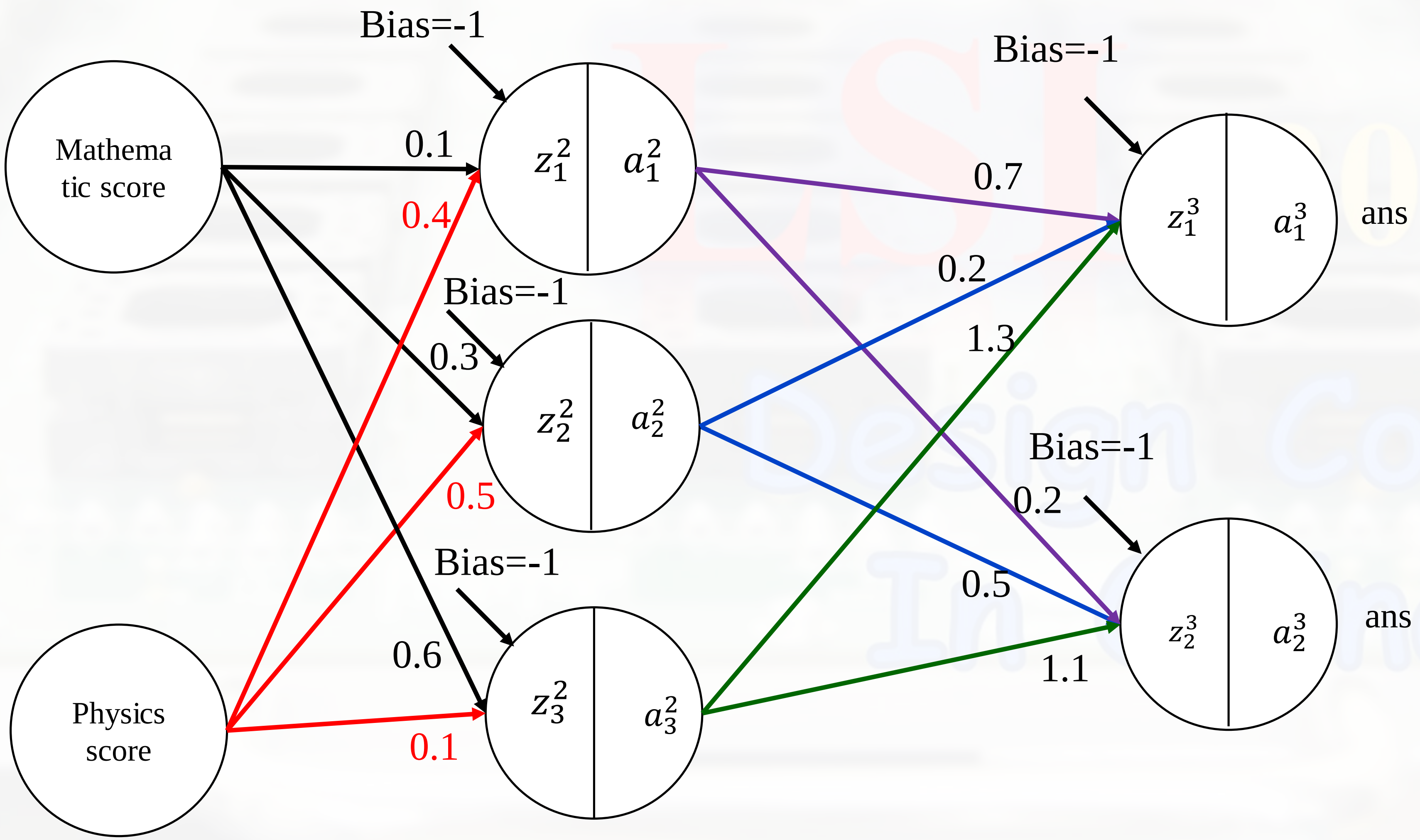
$$\delta^2_1 = \frac{\partial C}{\partial z^2_1} = \frac{\partial C}{\partial z^3_1}\frac{\partial z^3_1}{\partial a^2_1}\frac{\partial a^2_1}{\partial z^2_1} + \frac{\partial C}{\partial z^3_2}\frac{\partial z^3_2}{\partial a^2_1}\frac{\partial a^2_1}{\partial z^2_1} + \cdots$$

$$\left(\frac{\partial C}{\partial z^3_1} = \delta^3_1, \frac{\partial C}{\partial z^3_2} = \delta^3_2, \ldots \frac{\partial z^3_1}{\partial a^2_1} = w^3_{1\,1}, \frac{\partial z^3_2}{\partial a^2_1} = w^3_{2\,1}, \cdots\right)$$

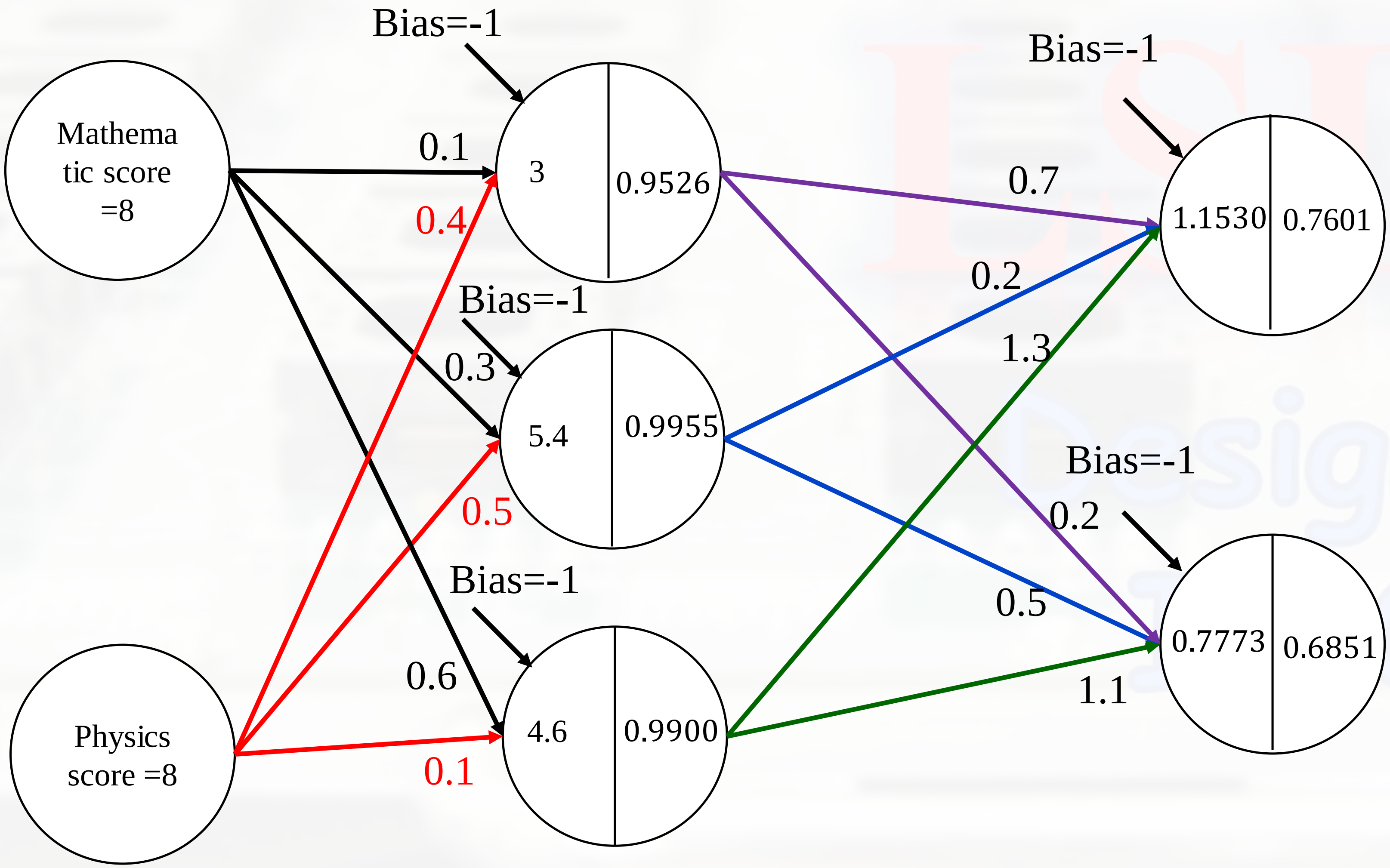
$$\delta^2_1 = \delta^3_1 w^3_{1\,1} a'(z^2_1) + \delta^3_2 w^3_{2\,1} a'(z^2_1) + \cdots$$

$$\therefore \delta^2_1 = (\delta^3_1 w^3_{1\,1} + \delta^3_2 w^3_{2\,1} + \cdots)a'(z^2_1)$$

Initial parameter



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



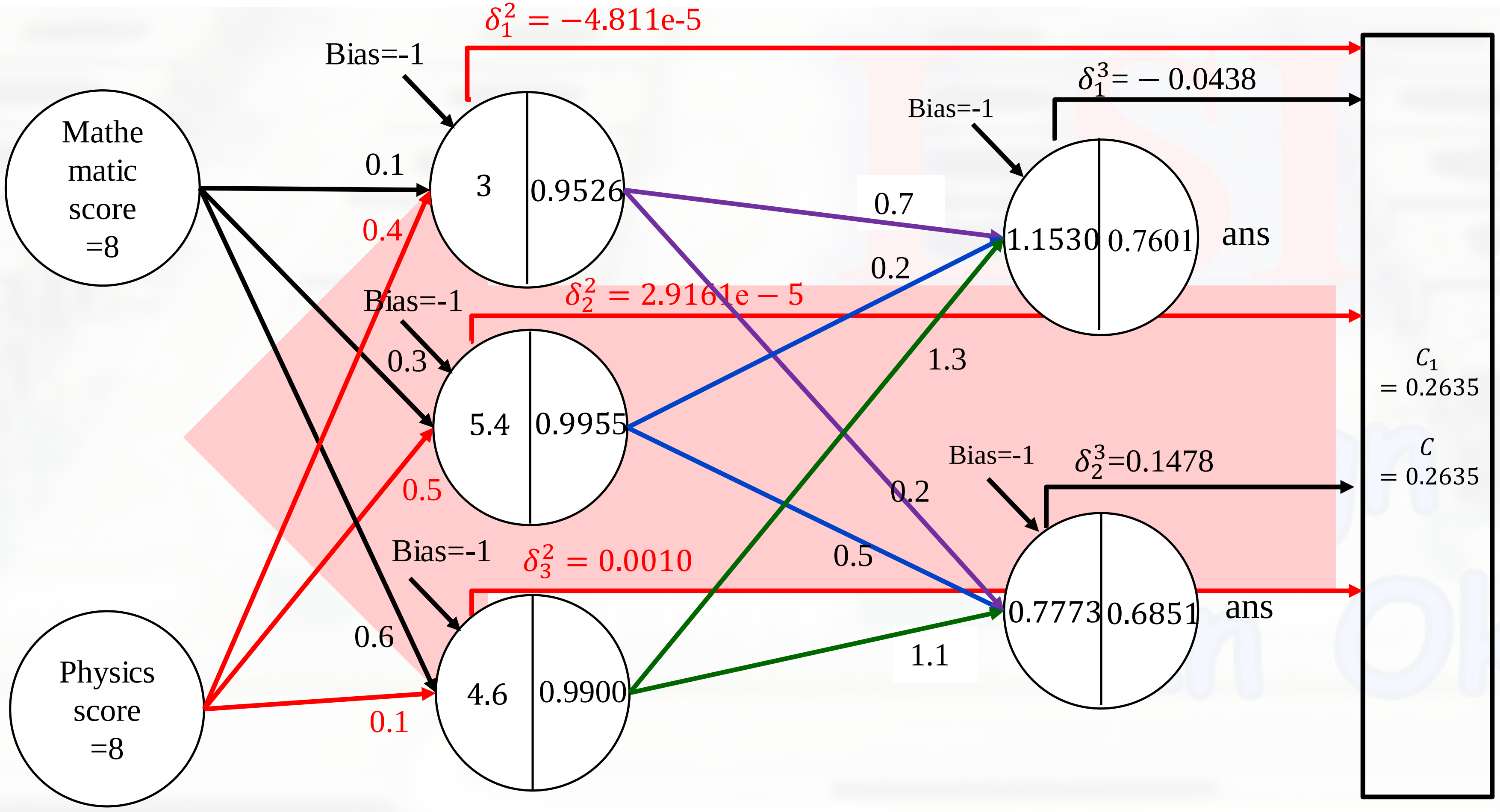
$\therefore z_i^2 = w_{1\,i}^2 k_1 + w_{2\,i}^2 k_2 + b_i^2$

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

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

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

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



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

$\therefore \delta_1^2 = (\delta_1^3 w_{1\,1}^3 + \delta_2^3 w_{2\,1}^3)a'(z_1^2)$

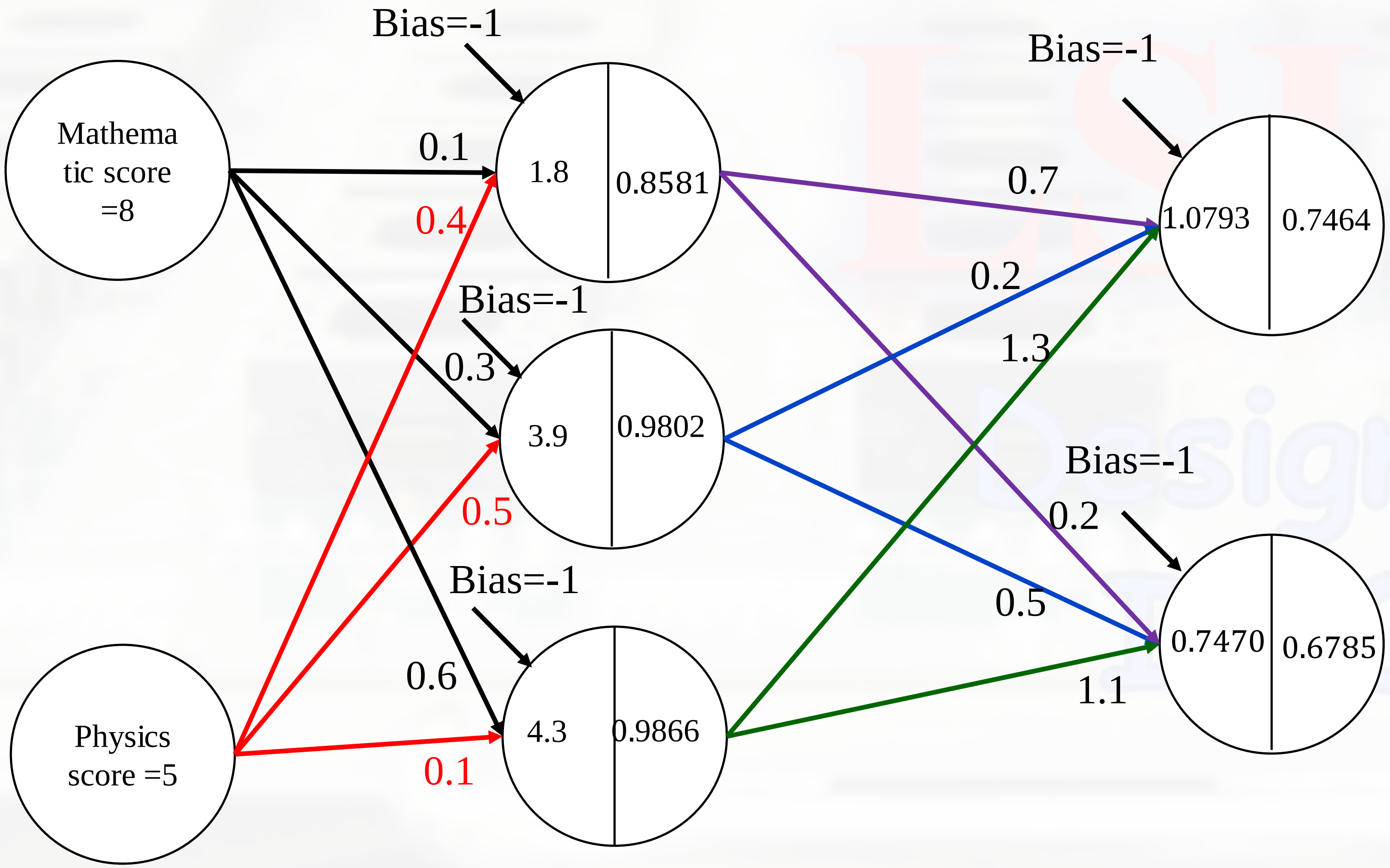
$\therefore \delta w_{i\,j}^3 = a_i^2 * \delta_i^3$

$\therefore \delta w_{i\,j}^2 = k_i * \delta_i^3$

$\therefore \delta b_{i\,j}^3 = \delta_i^3$

$\therefore \delta b_{i\,j}^2 = \delta_i^2$

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



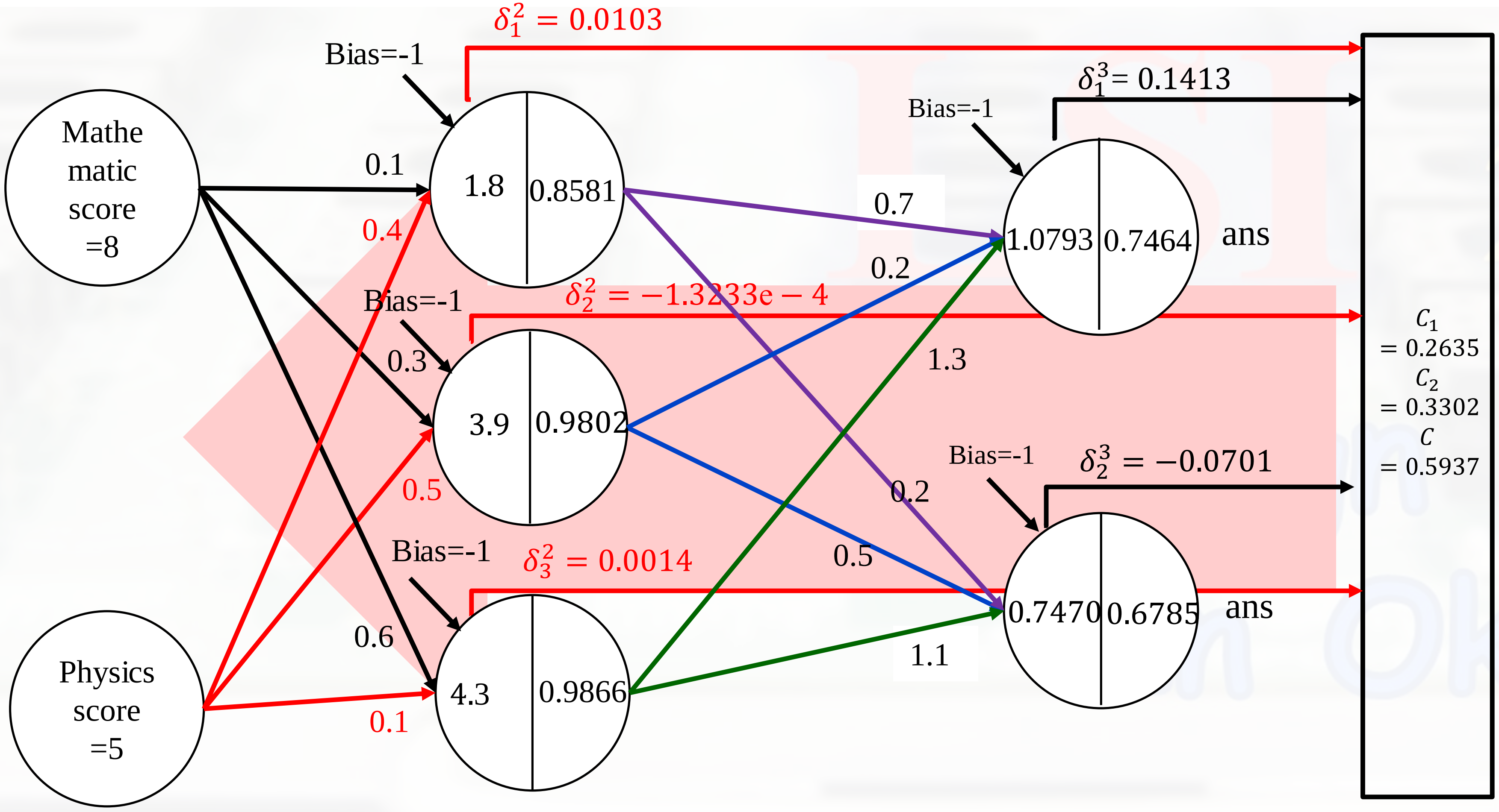
$$\therefore z_i^2 = w^2_{1\,i} k_1 + w^2_{2\,i} k_2 + b_i^2$$

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

$$\therefore z_i^3 = w^3_{1\,i} a_i^2 + w^2_{2\,i} a_i^2 + w^2_{3\,i} a_i^2 + b_i^3$$

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

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



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

$\therefore \delta_1^2 = (\delta_1^3 w_{1\,1}^3 + \delta_2^3 w_{2\,1}^3)a'(z_1^2)$

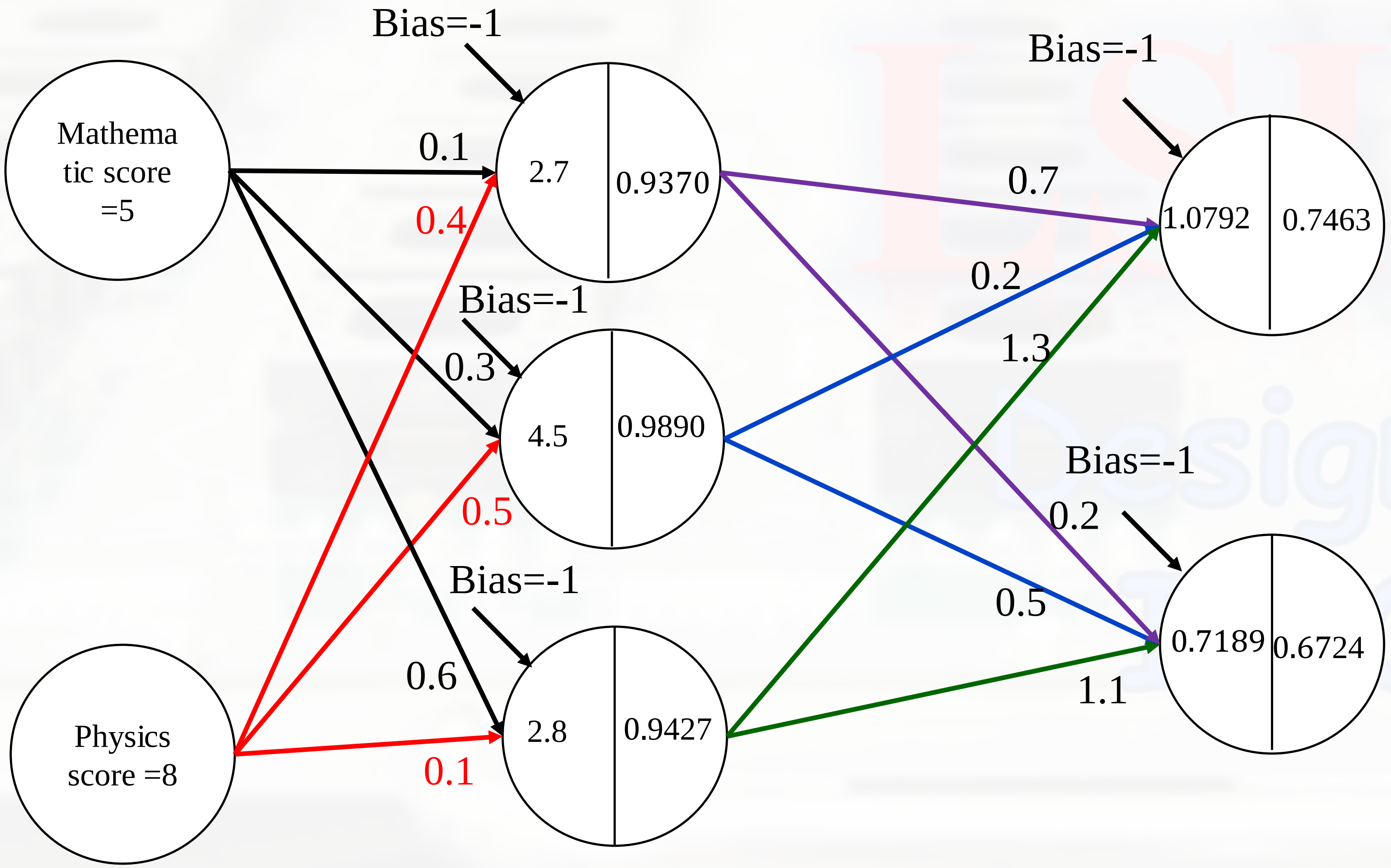
$\therefore \delta w_{i\,j}^3 = \delta w_{i-1\,j}^3 + a_i^2 * \delta_i^3$

$\therefore \delta w_{i\,j}^2 = \delta w_{i-1\,j}^2 + k_i * \delta_i^3$

$\therefore \delta b_{i\,j}^3 = b_{i-1\,j}^3 + \delta_i^3$

$\therefore \delta b_{i\,j}^2 = b_{i-1\,j}^2 + \delta_i^2$

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



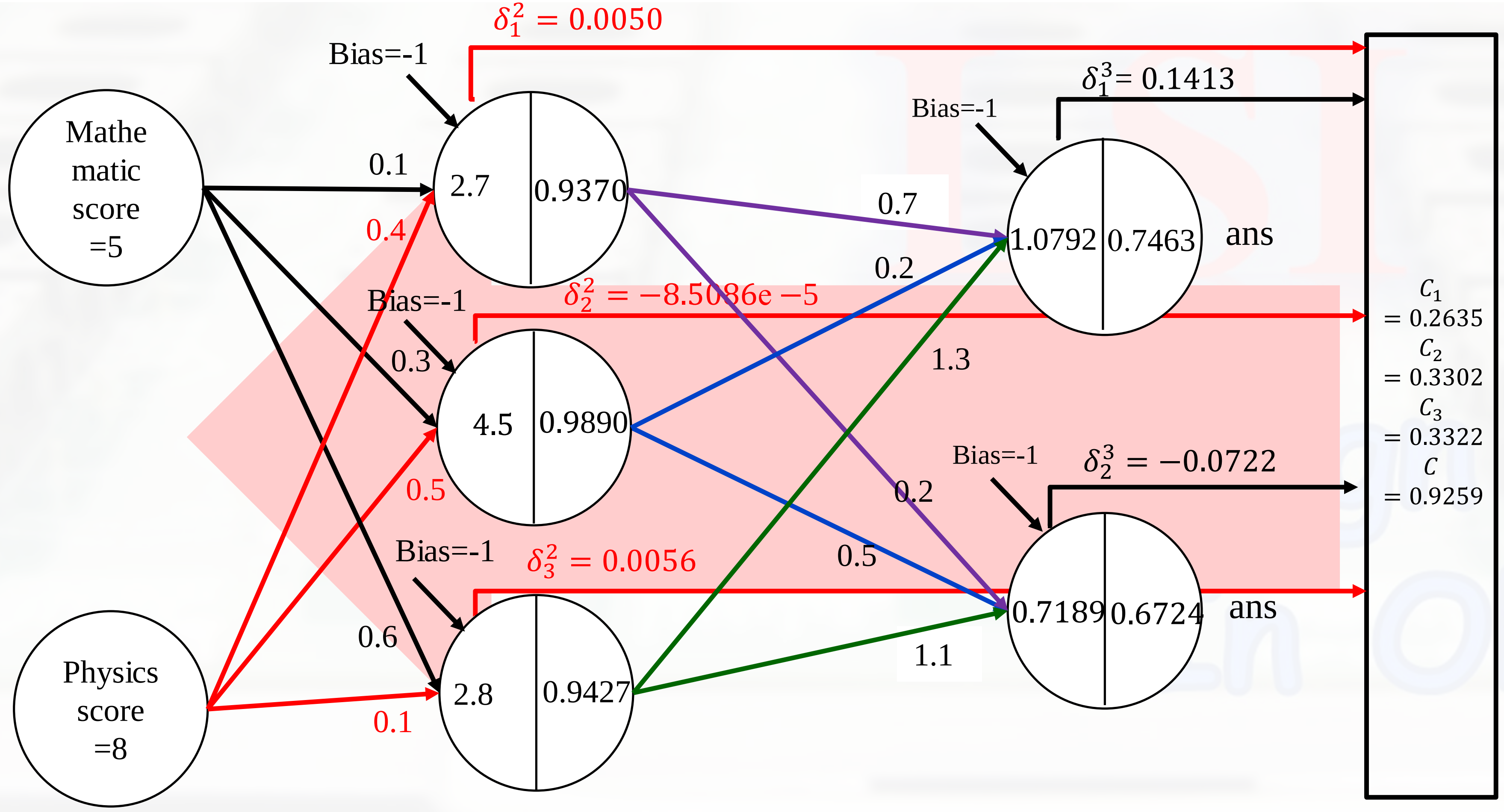
$$\therefore z_i^2 = w_{1\,i}^2 k_1 + w_{2\,i}^2 k_2 + b_i^2$$

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

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

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

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



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

$\therefore \delta_1^2 = (\delta_1^3 w^3_{1\,1} + \delta_2^3 w^3_{2\,1})a'(z_1^2)$

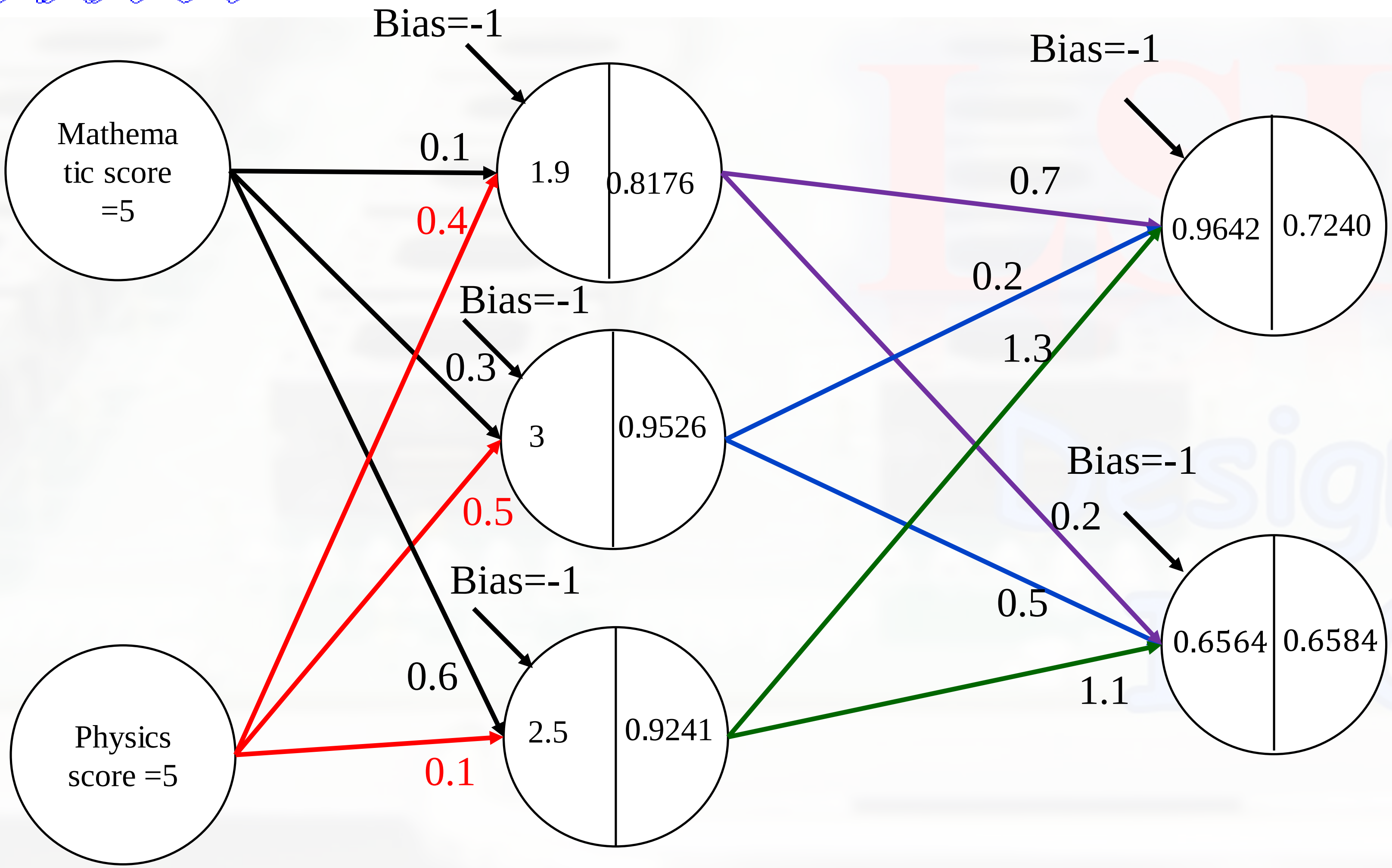
$\therefore \delta w^3_{i\,j} = \delta w^3_{i-1\,j} + a_i^2 * \delta_i^3$

$\therefore \delta w^2_{i\,j} = \delta w^2_{i-1\,j} + k_i * \delta_i^3$

$\therefore \delta b^3_{i\,j} = b^3_{i-1\,j} + \delta_i^3$

$\therefore \delta b^2_{i\,j} = b^2_{i-1\,j} + \delta_i^2$

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



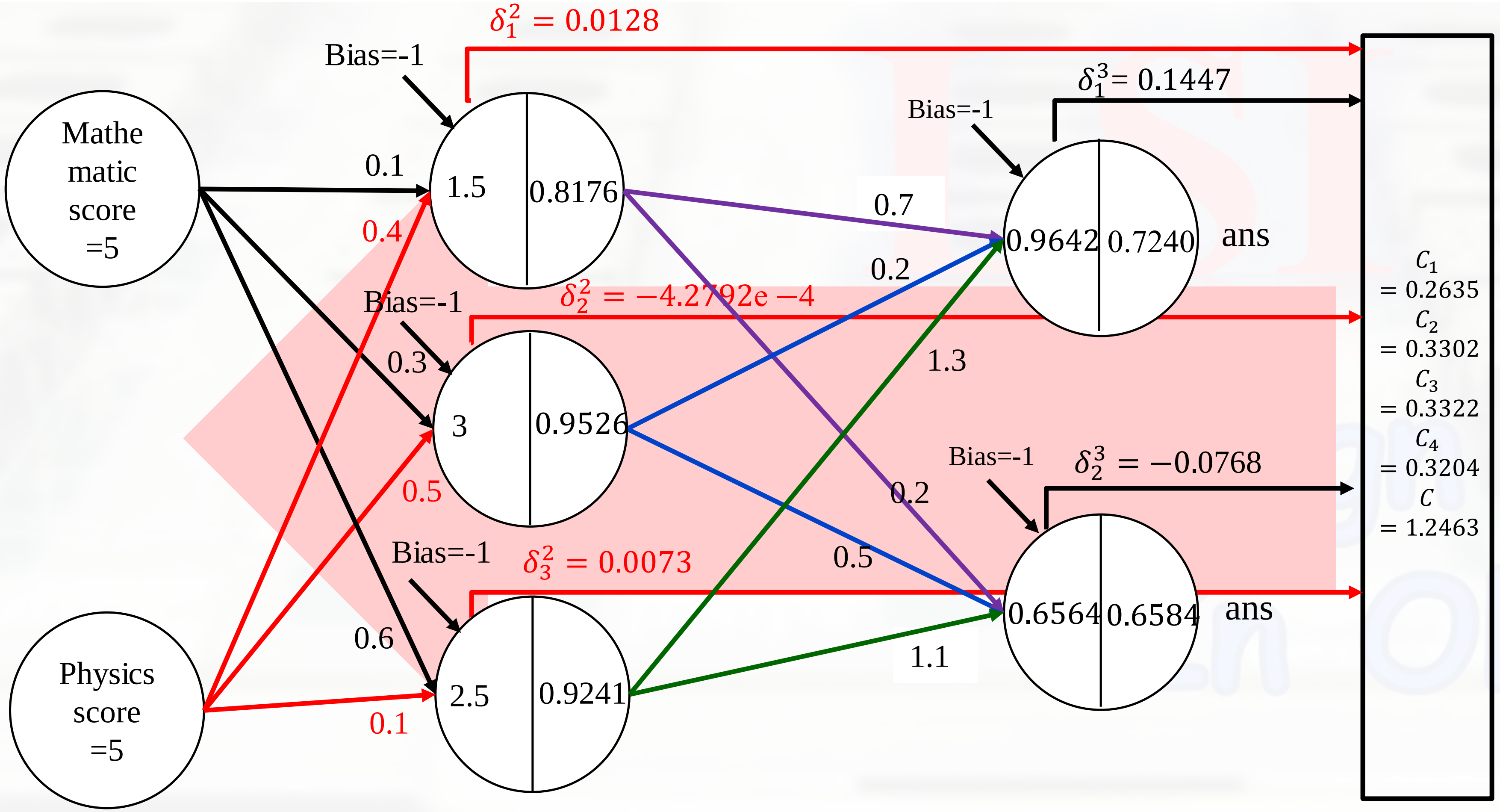
$\therefore z_i^2 = w_{1\,i}^2 k_1 + w_{2\,i}^2 k_2 + b_i^2$

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

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

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

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



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

$\therefore\ \delta_1^2 = (\delta_1^3 w_{1\,1}^3 + \delta_2^3 w_{2\,1}^3)a'(z_1^2)$

$\therefore\ \delta w_{i\,j}^3 = \delta w_{i-1\,j}^3 + a_i^2 * \delta_i^3$

$\therefore\ \delta w_{i\,j}^2 = \delta w_{i-1\,j}^2 + k_i * \delta_i^3$

$\therefore\ \delta b_{i\,j}^3 = b_{i-1\,j}^3 + \delta_i^3$

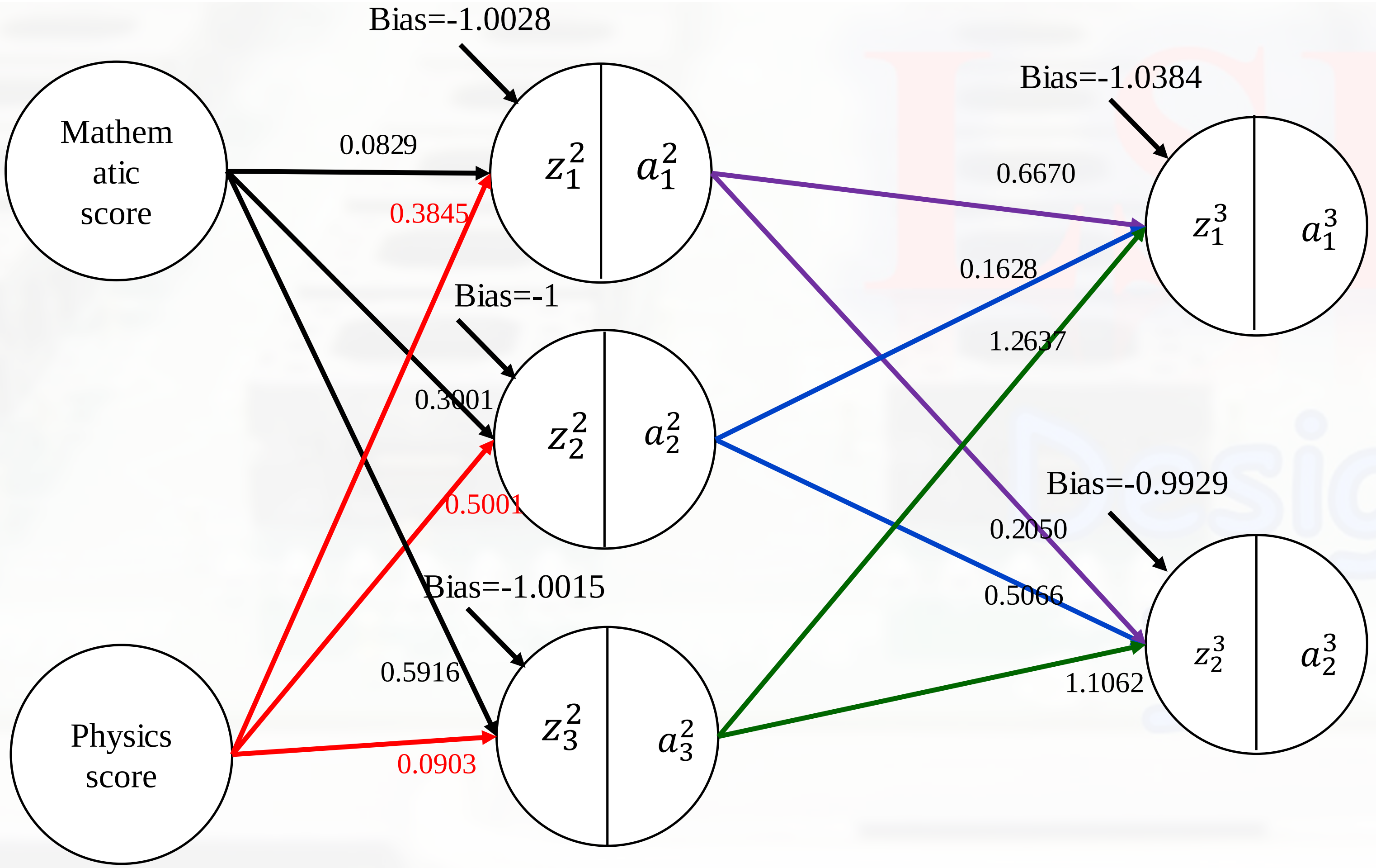
$\therefore\ \delta b_{i\,j}^2 = b_{i-1\,j}^2 + \delta_i^2$

Calculation of Cost Func.

Supervisor $[t_1, t_2]$	Output value $[a_3^1, a_3^2]$	Square Error C_i	Cost func. C
[1,0]	[0.76, 0.69]	0.26	1.246
[0,1]	[0.75, 0.68]	0.33	
[0,1]	[0.75, 0.67]	0.33	
[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

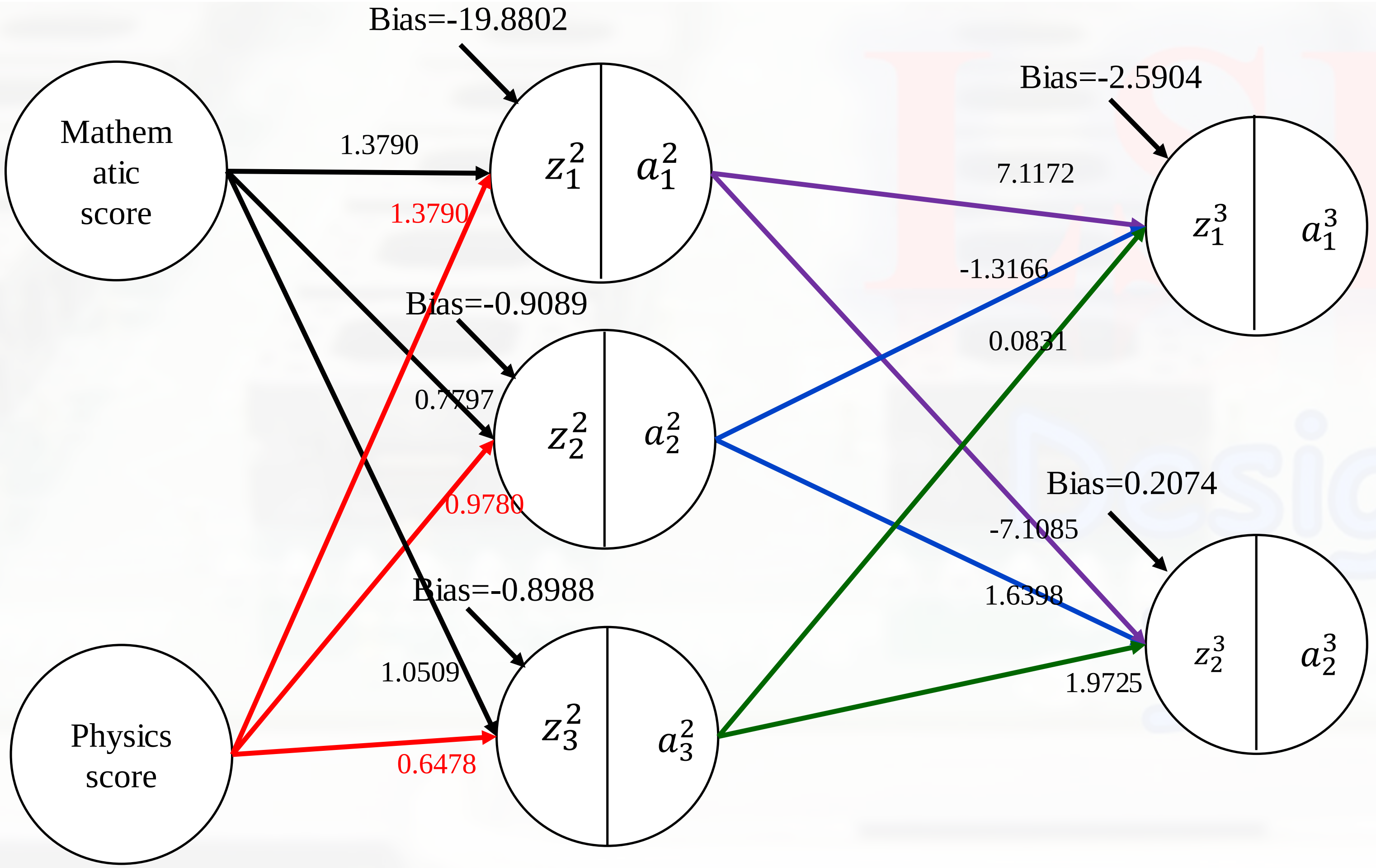
New parameter(after 1st renewal)



- $(new)w_{i\,j}^3 = (old)w_{i\,j}^3 - \eta * \delta w_{i\,j}^3$
- $(new)w_{i\,j}^2 = (old)w_{i\,j}^2 - \eta * \delta w_{i\,j}^2$
- $(new)b_{i\,j}^3 = (old)b_{i\,j}^3 - \eta * \delta b_{i\,j}^3$
- $(new)b_{i\,j}^2 = (old)b_{i\,j}^2 - \eta * \delta b_{i\,j}^2$

$\therefore \eta = 0.1$

New parameter(after 10000th renewal)



- $(new)w_{i\,j}^3 = (old)w_{i\,j}^3 - \eta * \delta w_{i\,j}^3$
- $(new)w_{i\,j}^2 = (old)w_{i\,j}^2 - \eta * \delta w_{i\,j}^2$
- $(new)b_{i\,j}^3 = (old)b_{i\,j}^3 - \eta * \delta b_{i\,j}^3$
- $(new)b_{i\,j}^2 = (old)b_{i\,j}^2 - \eta * \delta b_{i\,j}^2$

$\therefore \eta = 0.1$

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.