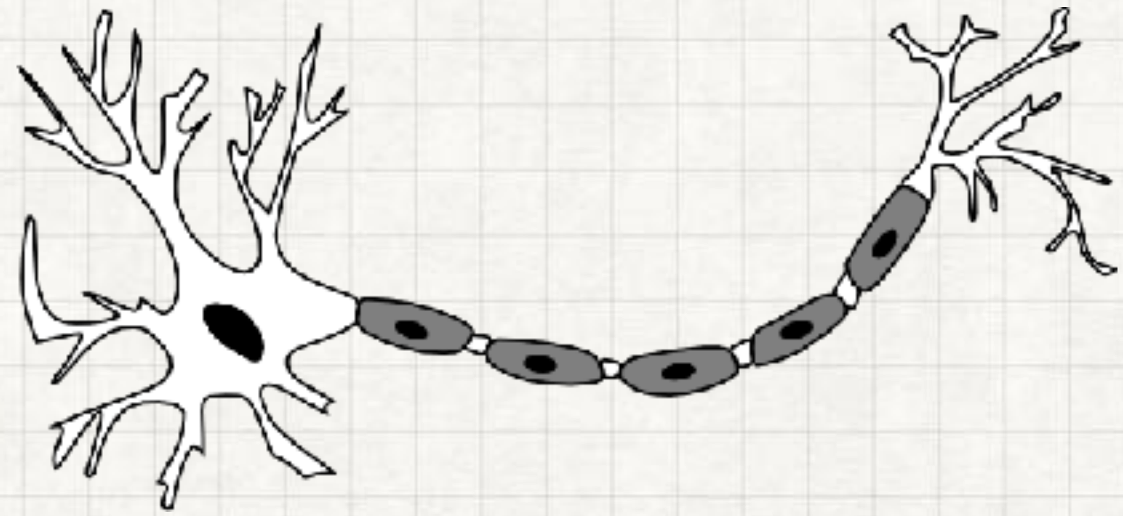
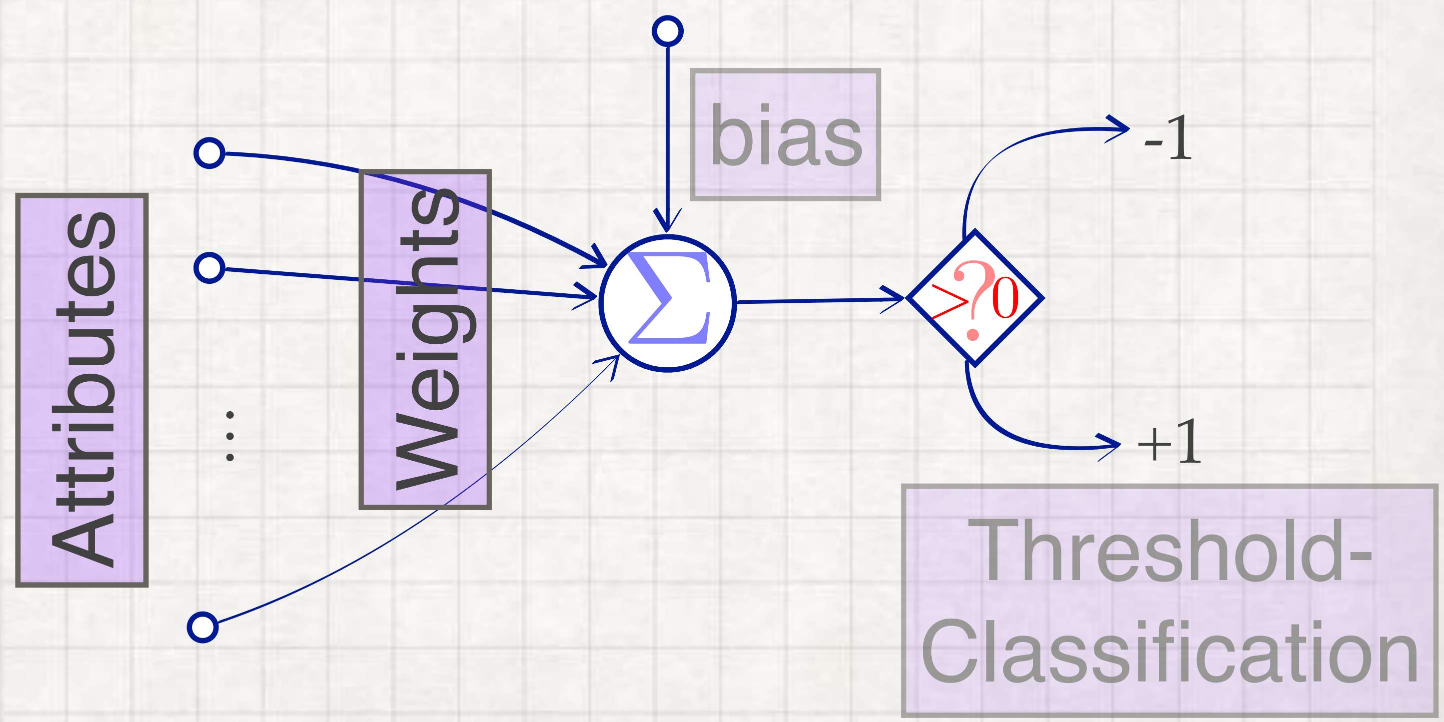
NEURAL NETWORK - I

MOD1: THE MAKE OF AN ARTIFICIAL NEURON

REVIEW: PERCEPTRON

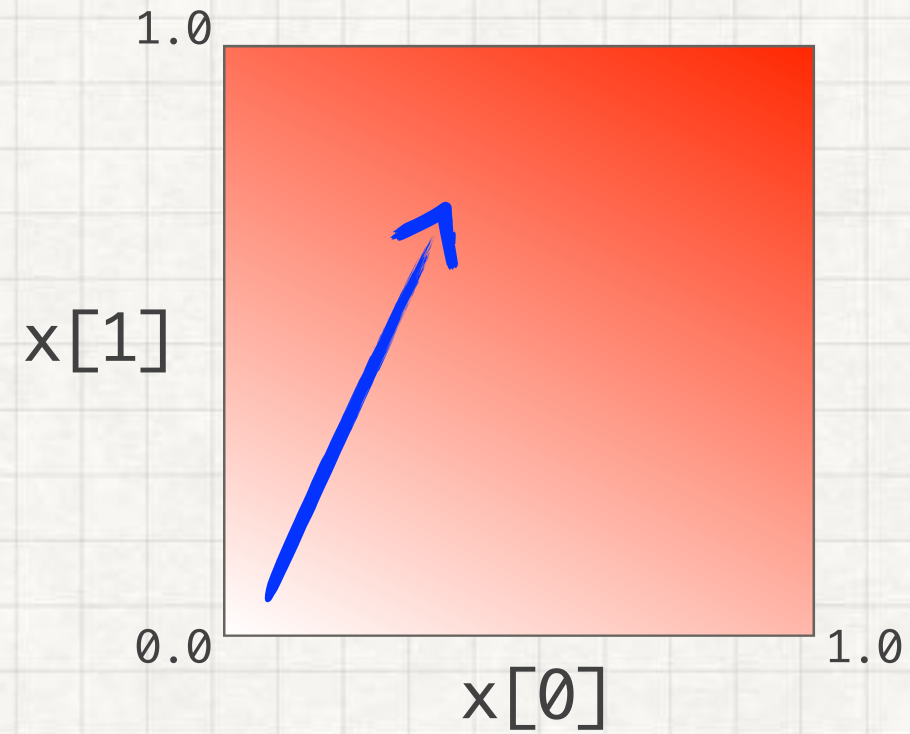
- A linear model
- $a = w_1x_1 + w_2x_2 + \ldots + w_px_p + b$
- Decision based on a:
- $y = -1,\ a < 0$
  $y = +1,\ a \geq 0$





PERCEPTRON IN A DATA *SPACE*

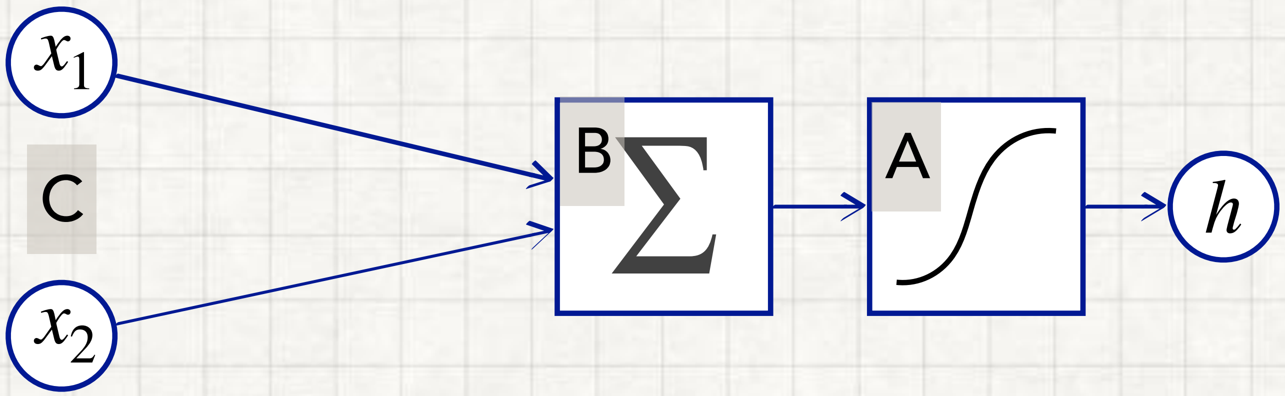
```
w[0] = 1.0
w[1] = 2.0
a = 1.0 * x[0] + 2.0 * x[1]
```



GEOMETRIC INTERPRETATION OF A-VALUES

- Colours are corresponding to “a” values at every possible point in the data space

GENERALISE A PERCEPTRON



Perceptron

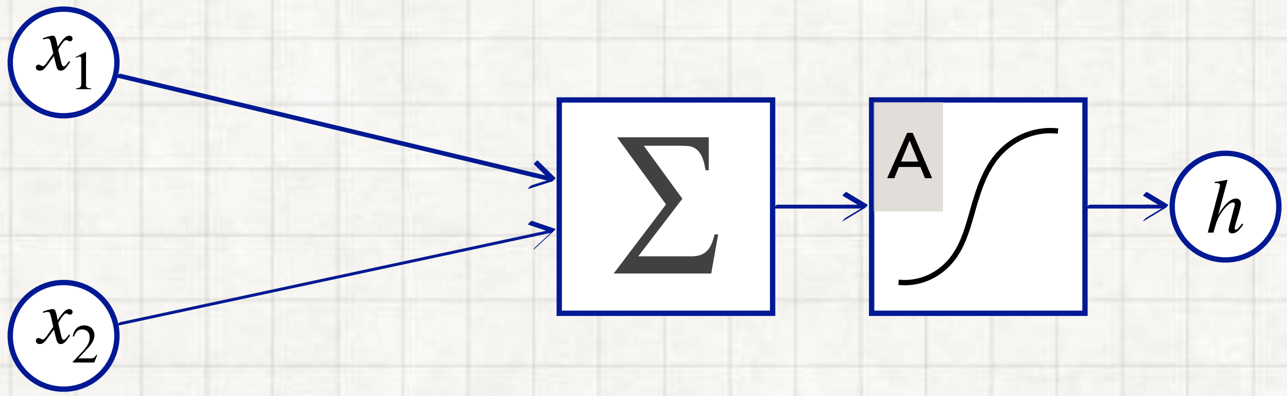
Q: Your idea on extending a perceptron?

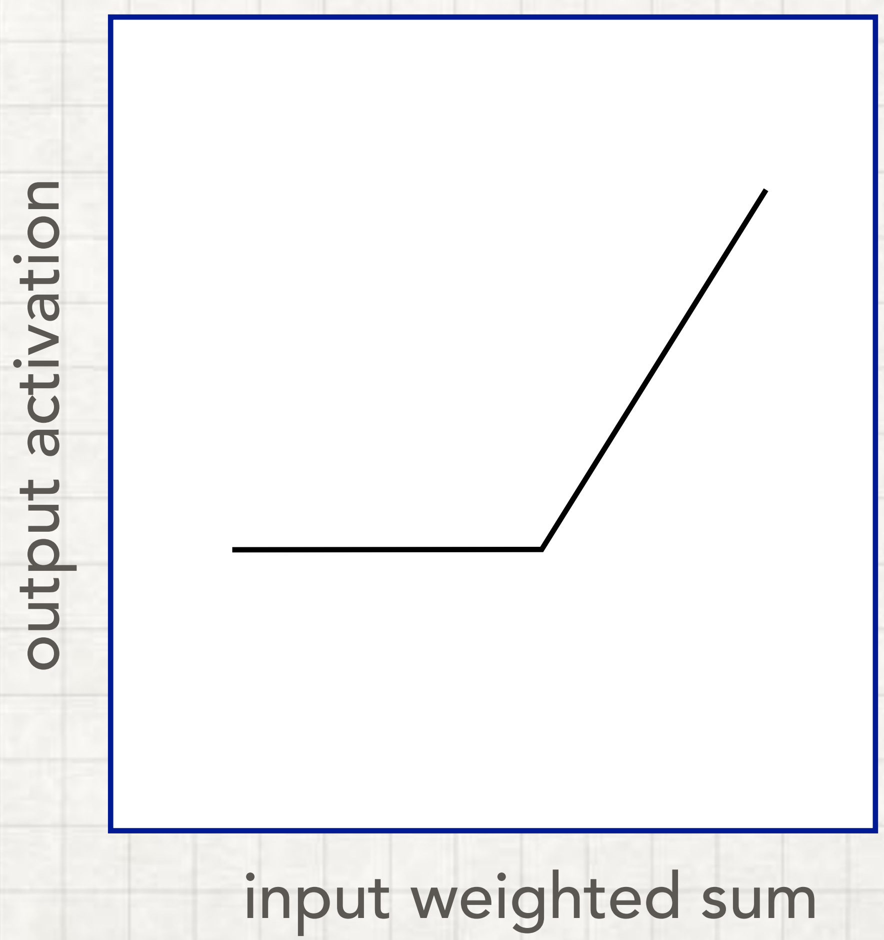
A. Activation

B. Sum

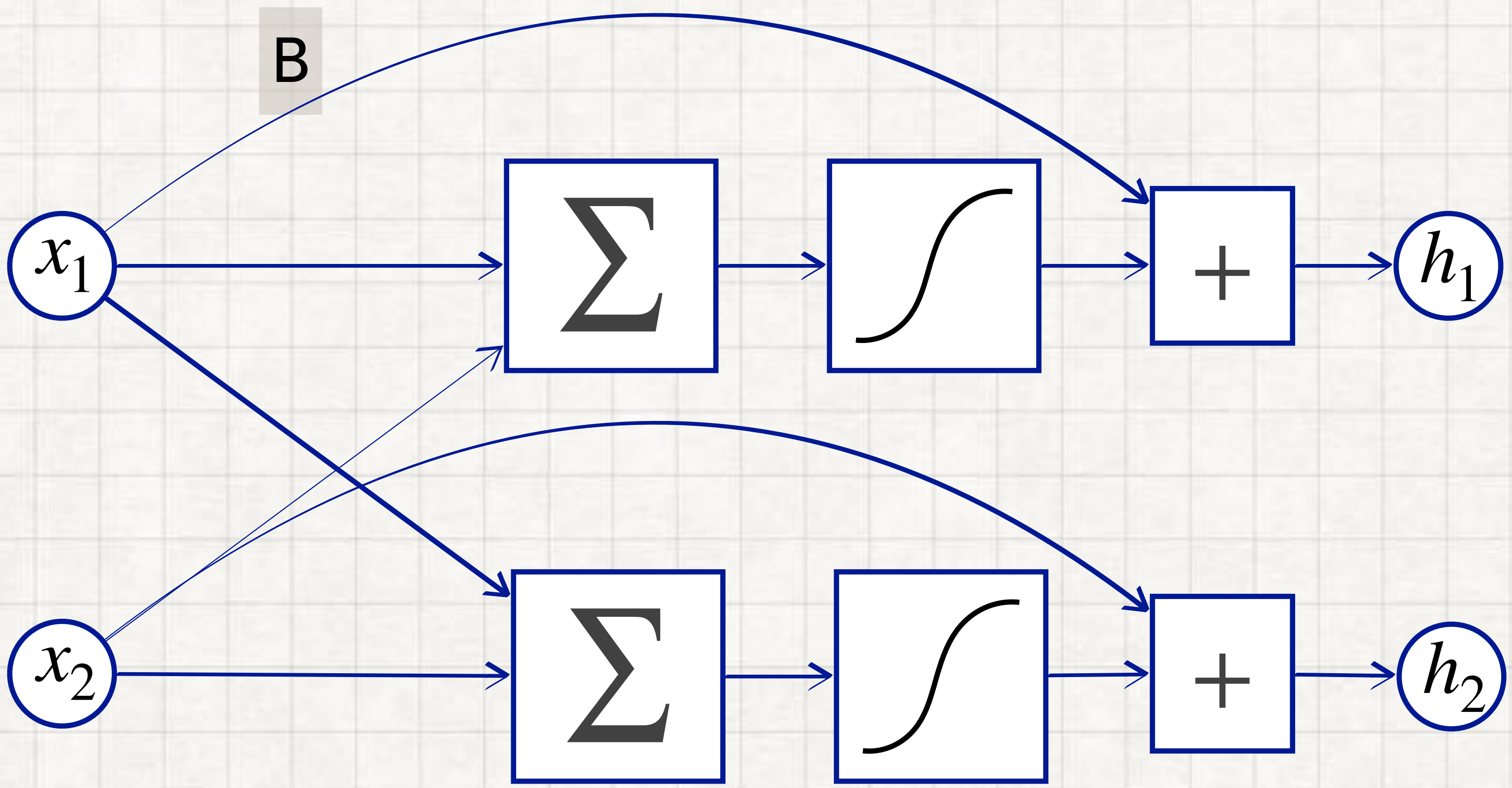
C. Transform $x's$

PERCEPTRON.X

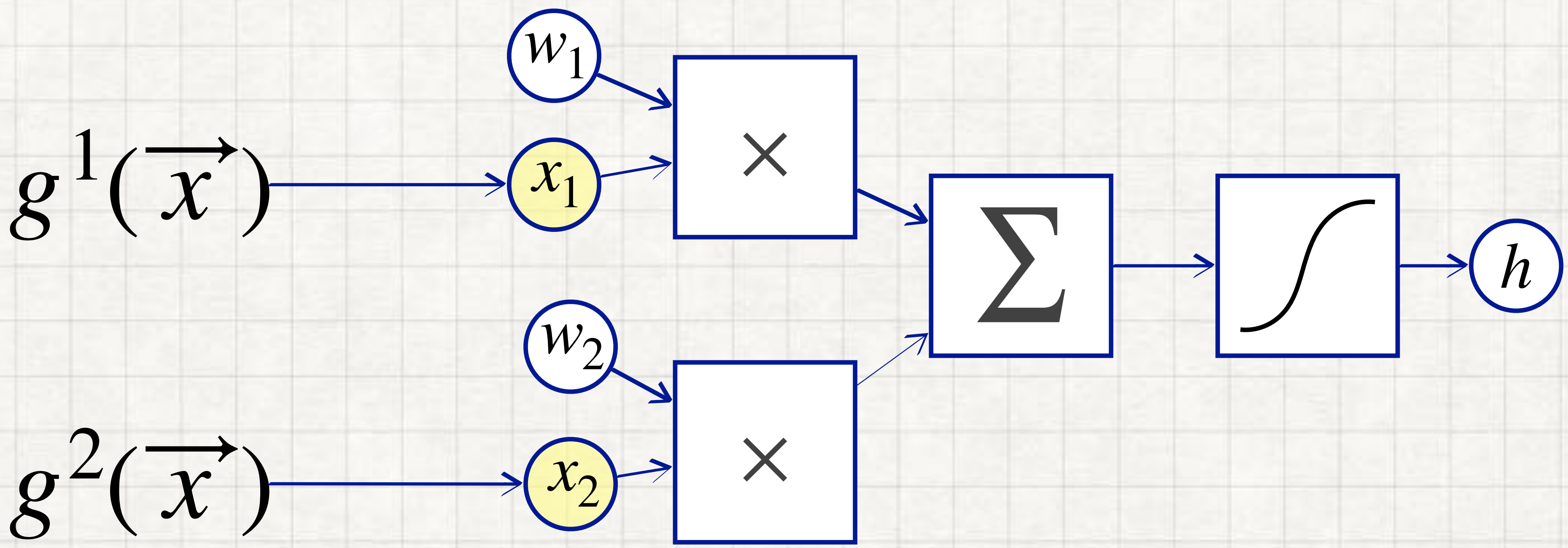




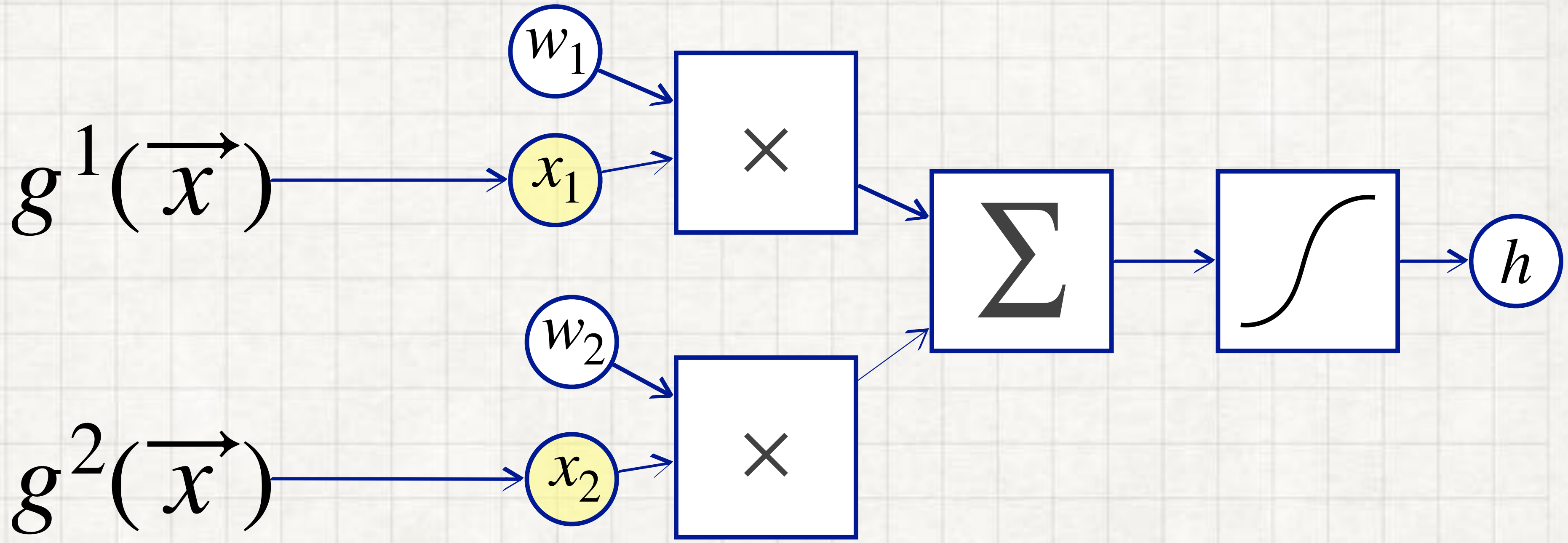
PERCEPTRON.X



INPUT-TRANSFORMED PERCEPTRON



INPUT-TRANSFORMED PERCEPTRON



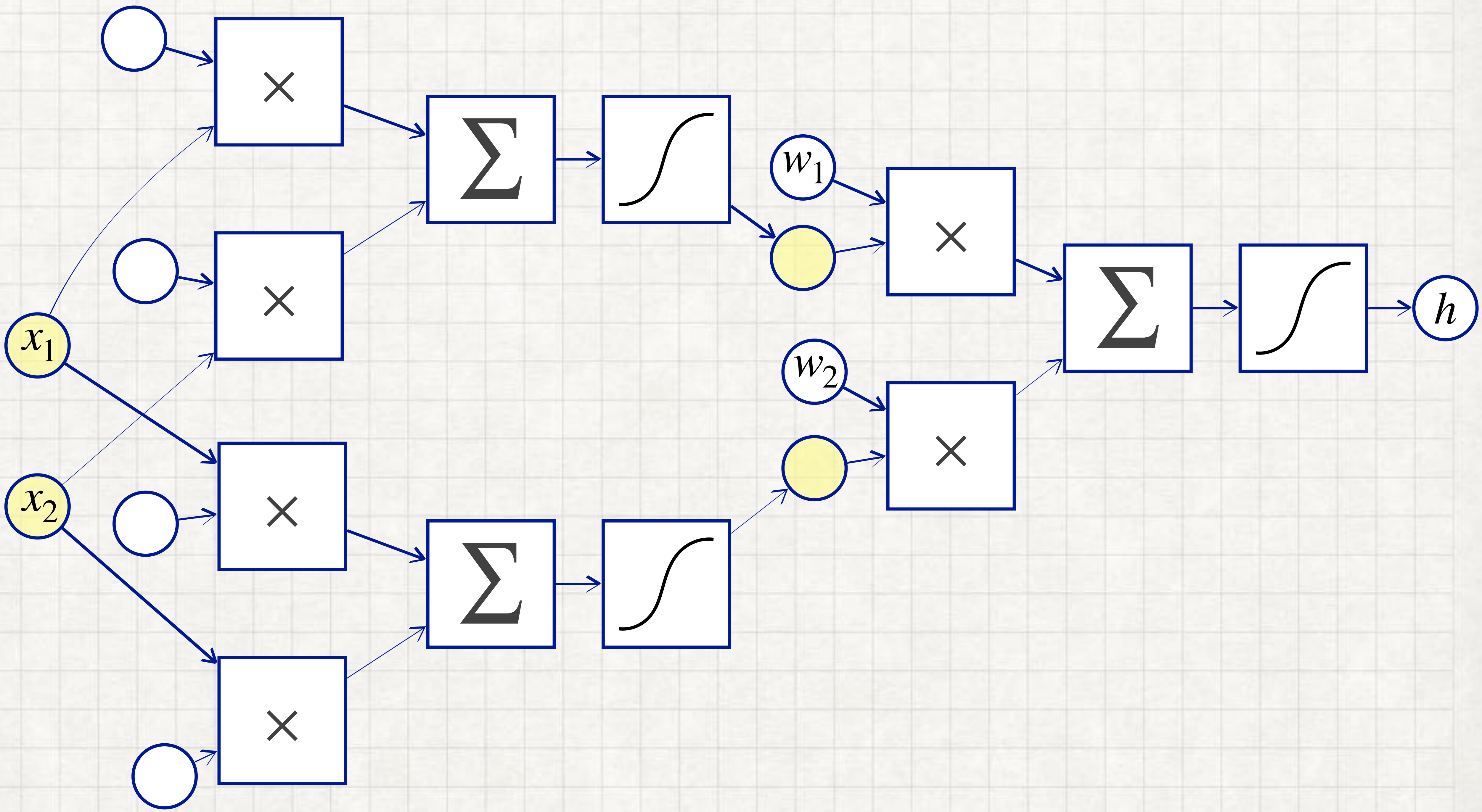
Q: In the example, when we use $\sin(\omega_1 \cdot x_1)$ as transform-1 and $\sin(\omega_2 \cdot x_2)$ as transform-2, how many degree of freedom in the new perceptron model with transformed inputs?

A. 2, w_1 and w_2.

B. 4, w_1, w_2, ω_1 and ω_2.

C. 6, w_1, w_2, ω_1, ω_2, x_1 and x_2

MULTI-LAYER PERCEPTRON



INPUT-TRANSFORMED PERCEPTRON

$$X \cdot W \overset{\varphi}{\rightarrow} H, \cdot U \overset{\varphi}{\rightarrow} Y = \varphi(\varphi(X \cdot W) \cdot U)$$

Collapsed:

$$Y = X \cdot W \cdot U = X \cdot (W \cdot U) = X \cdot V$$

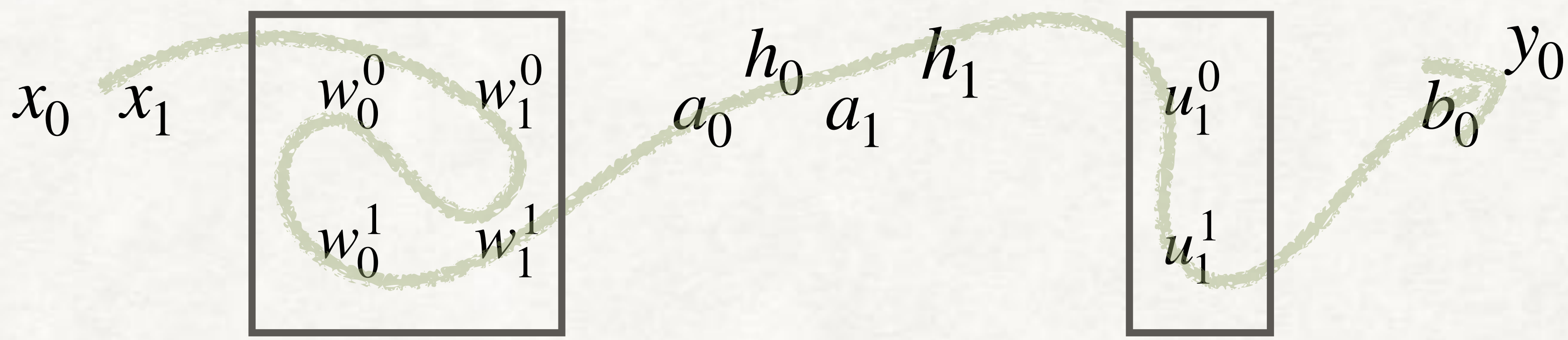
EX. MULTI-LAYER PERCEPTRON USING MATRIX

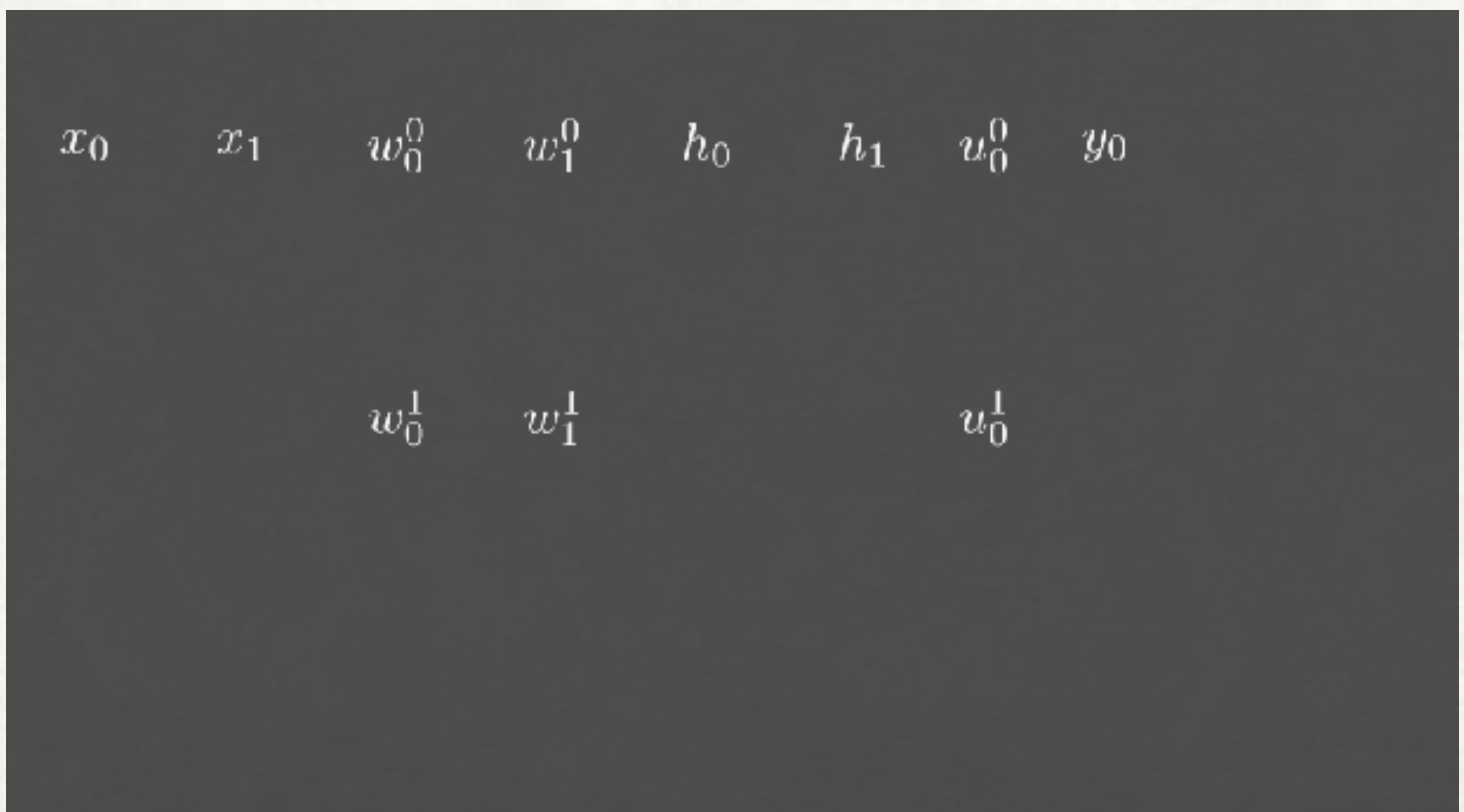
MOD2: TRAIN AN ARTIFICIAL NEURON

ADJUST NET OUTPUTS

- Gradient Definition
  - "What if .." for each parameter
  - Gradient Computation (Back-propagation)

AN EXAMPLE


x_0 x_1
w_0^0 w_1^0
w_0^1 w_1^1
a_0 h_0 a_1 h_1
u_1^0
u_1^1
b_0 y_0

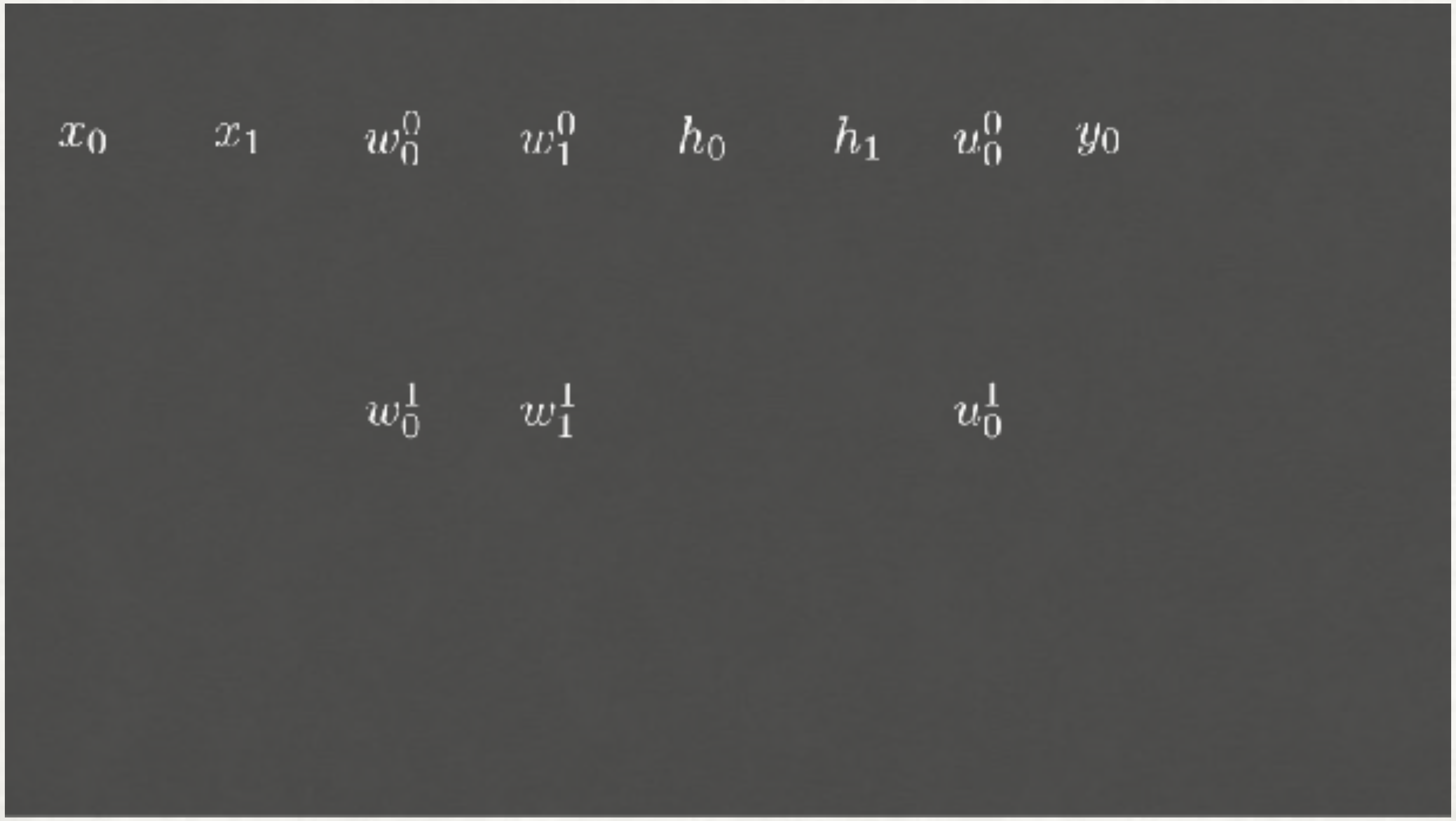

x_0 x_1 w_0^0 w_1^0 h_0 h_1 u_0^0 y_0
w_0^1 w_1^1 u_0^1

BP - 1: Y>>B

$x_0 \quad x_1 \qquad \boxed{\begin{matrix} w_0^0 & w_1^0 \\ w_0^1 & w_1^1 \end{matrix}} \qquad a_0 \quad h_0 \quad a_1 \quad h_1 \qquad \boxed{\begin{matrix} u_1^0 \\ u_1^1 \end{matrix}} \qquad b_0 \quad y_0$

$$\frac{dy}{db}$$

$$dy = ?db$$



$$dy = 1 \cdot df(b)$$

$$? = \frac{df(b)}{db}$$

$$dy = \frac{df(b)}{db} db$$

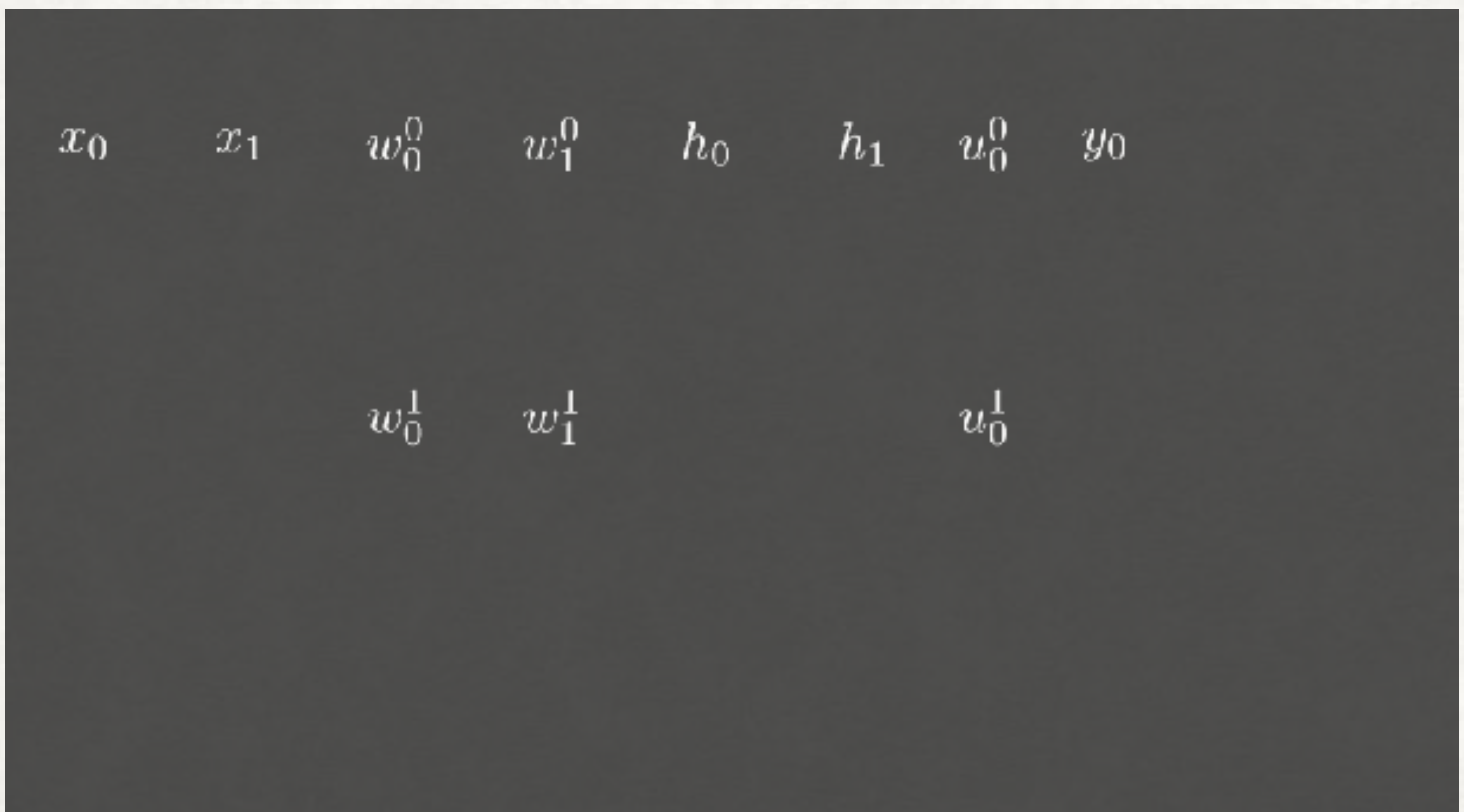
Computable Using B

BP - 1: Y>>B

$x_0 \quad x_1 \quad \boxed{\begin{matrix} w_0^0 & w_1^0 \\ w_0^1 & w_1^1 \end{matrix}} \quad a_0 \; h_0 \; a_1 \; h_1 \quad \boxed{\begin{matrix} u_1^0 \\ u_1^1 \end{matrix}} \quad b_0 \; y_0$

$$\frac{dy}{db}$$

$$dy = ?db$$



$$dy = 1 \cdot df(b)$$

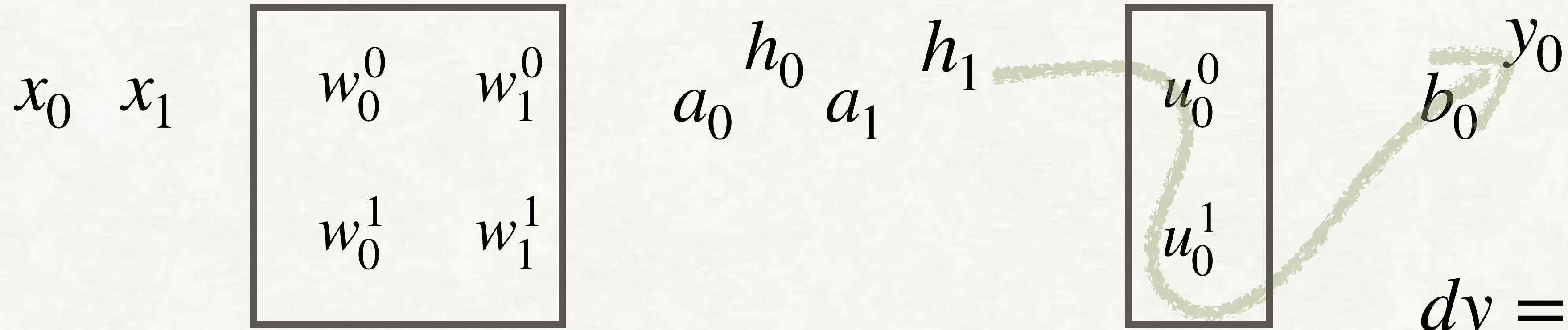
$$? = \frac{df(b)}{db}$$

$$dy = \frac{df(b)}{db}db$$

Computable Using B

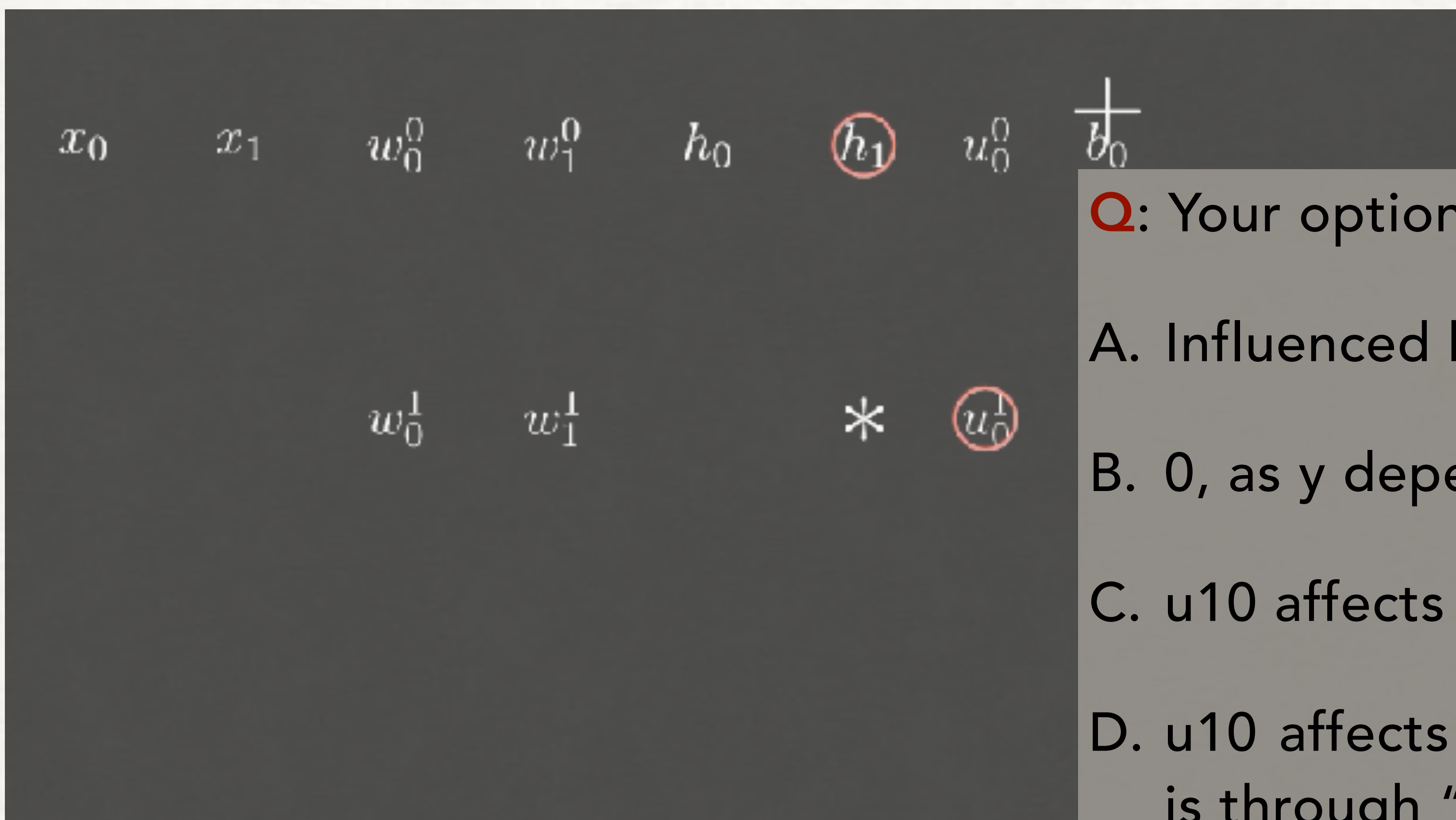
$$\frac{dSigmoid(b)}{db} = \frac{e^{-b}}{(1+e^{-b})^2}$$

BP - 2: B>>U



$$dy = 0.5db$$

$$dy = ?du_0^1$$



Q: Your option about the “?”:

A. Influenced by all x’s and w’s.

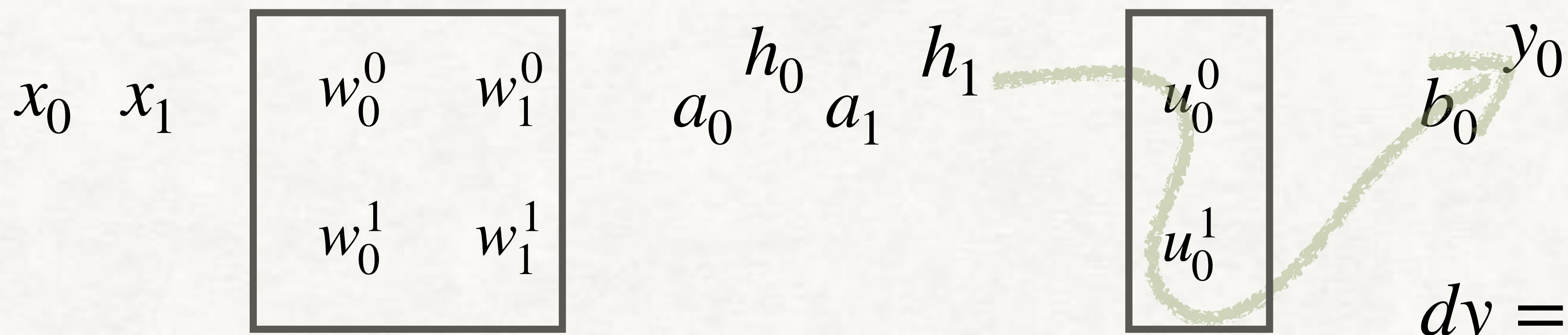
B. 0, as y depends on b, not u’s.

C. u10 affects b then y, so ?=0.5, same as b.

D. u10 affects b then y, its contribution to b is through “*h1” so ?=0.5*h1.

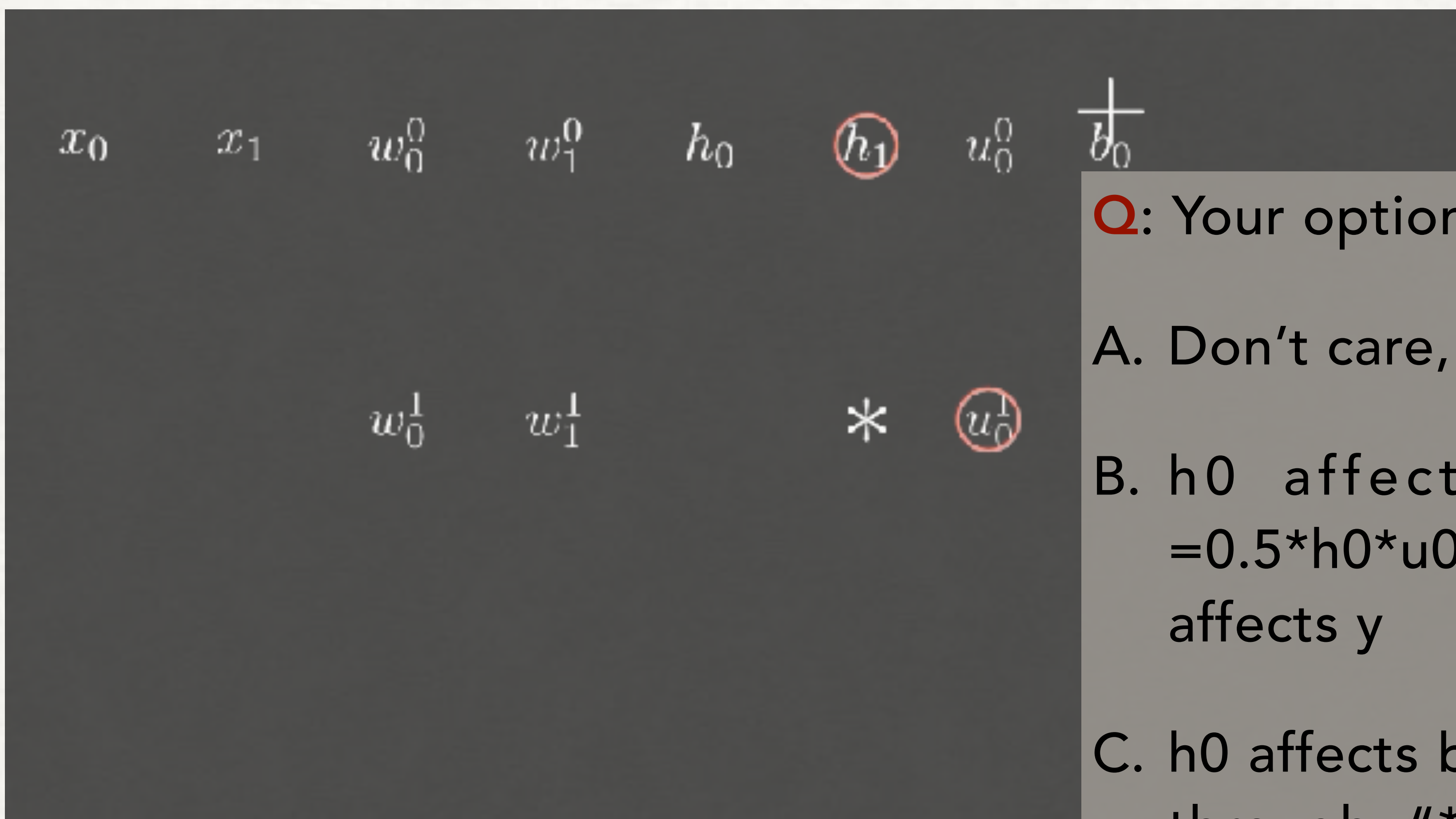
E. u10 contributes through “*h1”, so ?=h1

BP - 2A: B>>H



$$dy = 0.5db$$

$$dy = ?dh_0$$



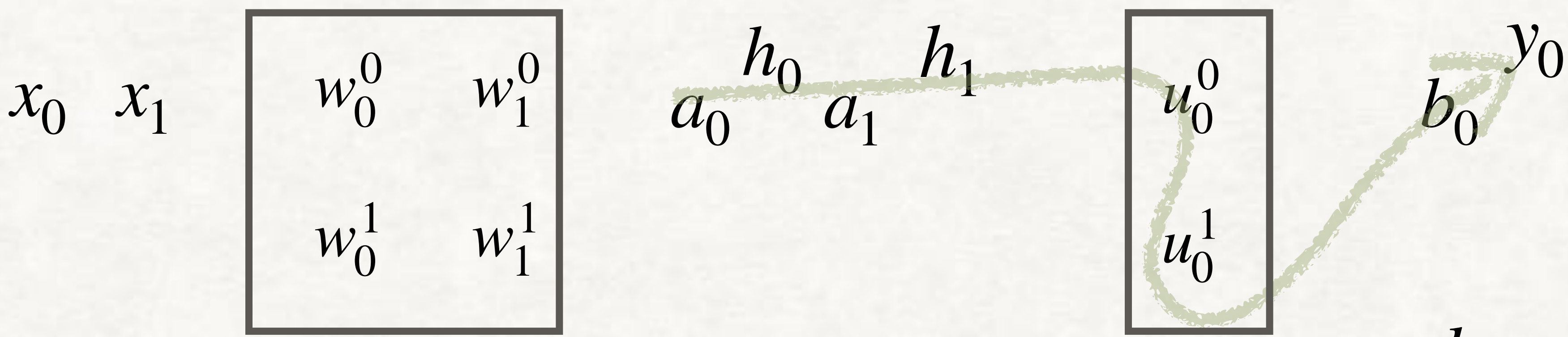
Q: Your option about the "?":

A. Don't care, h is not a learnable parameter.

B. h0 affects u00 then b then y, ? =0.5*h0*u00, where 0.5*h0 is how u00 affects y

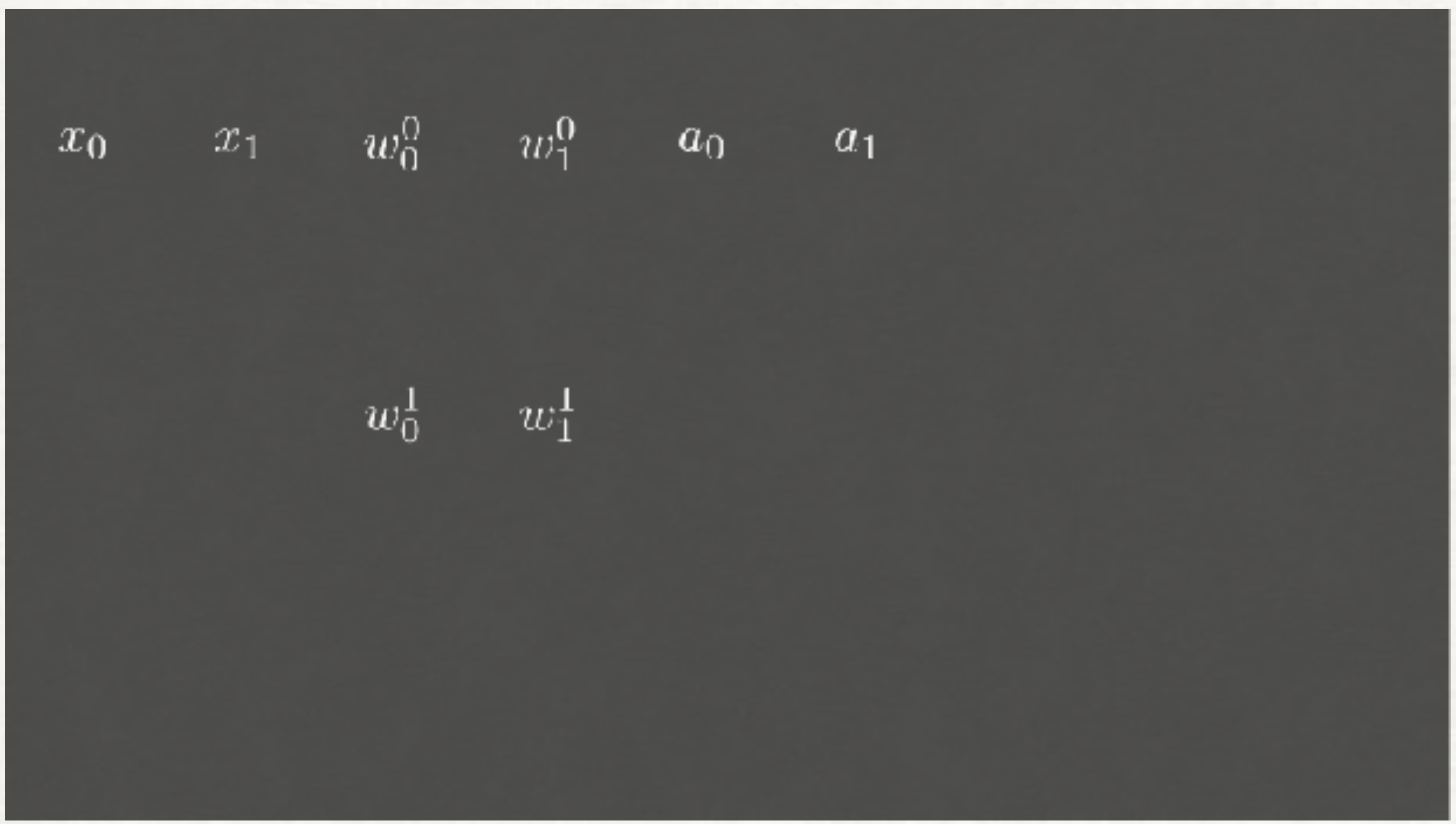
C. h0 affects b then y, its contribution to b is through "*u00" so ?=0.5*u00. And we DO need to compute h's effect, as we need to compute how to adjust parameters leading to h.

BP - 2A: H>>A



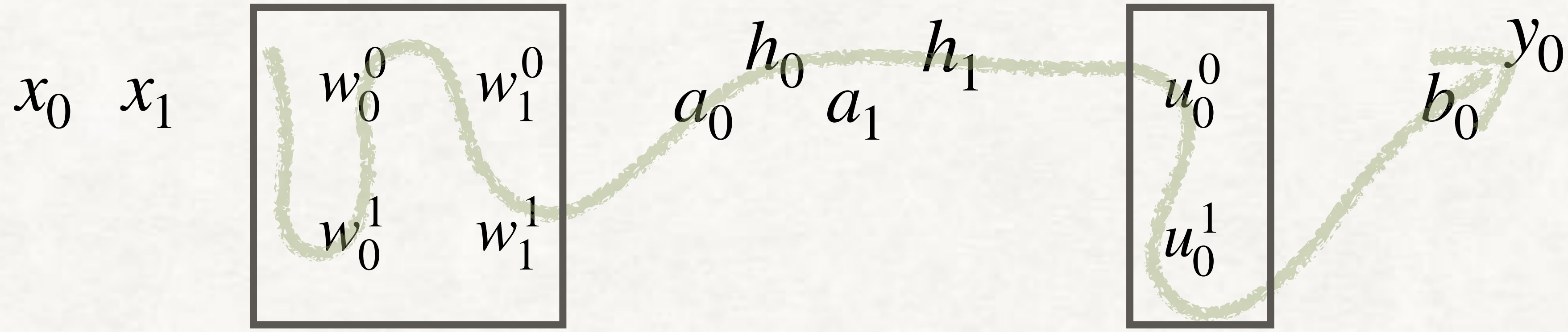
$$dy = -0.1 dh_0$$

$$dy = ?da_0$$

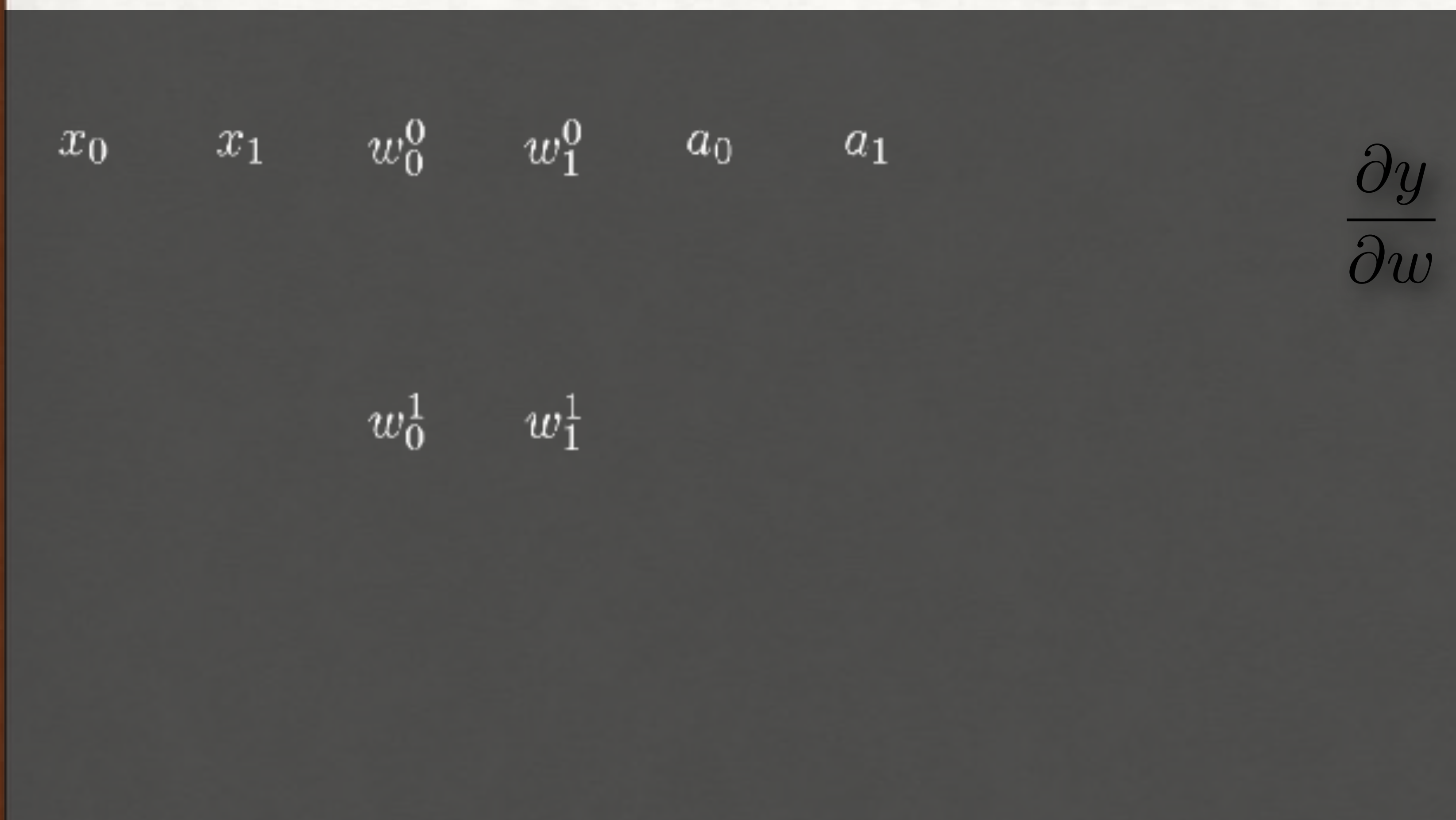


Similar to B<>Y, element-wise derivative.

BP - 2A: A>>W



$$\frac{\partial y}{\partial a} = [-1, 0.5]$$



$$\frac{\partial y}{\partial w} = \left[\begin{array}{c|c} \begin{array}{c} \frac{\partial y}{\partial w_0^0} = \frac{\partial y}{\partial a_0} x_0 \\ = -1 \cdot x_0 \end{array} & \frac{\partial y}{\partial w_1^0} \\ \hline \frac{\partial y}{\partial w_0^1} & \frac{\partial y}{\partial w_1^1} \end{array}\right]$$

$$= \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} \times \begin{bmatrix} \frac{\partial y}{\partial a_0} & \frac{\partial y}{\partial a_1} \end{bmatrix}$$

$$= X^T \cdot \frac{\partial y}{\partial a}$$

ESSENTIALS

- When Deployed (Working Condition/Forward Computation)
  - Connection Structure
  - Activation Mapping
- Training/Learning (Adjusting Parameters)
  - Gradient Definition
  - Gradient Computation (Back-propagation)
- Learning Material
  - Information Theory, Ch 39. Fig. 39.4

THANKS