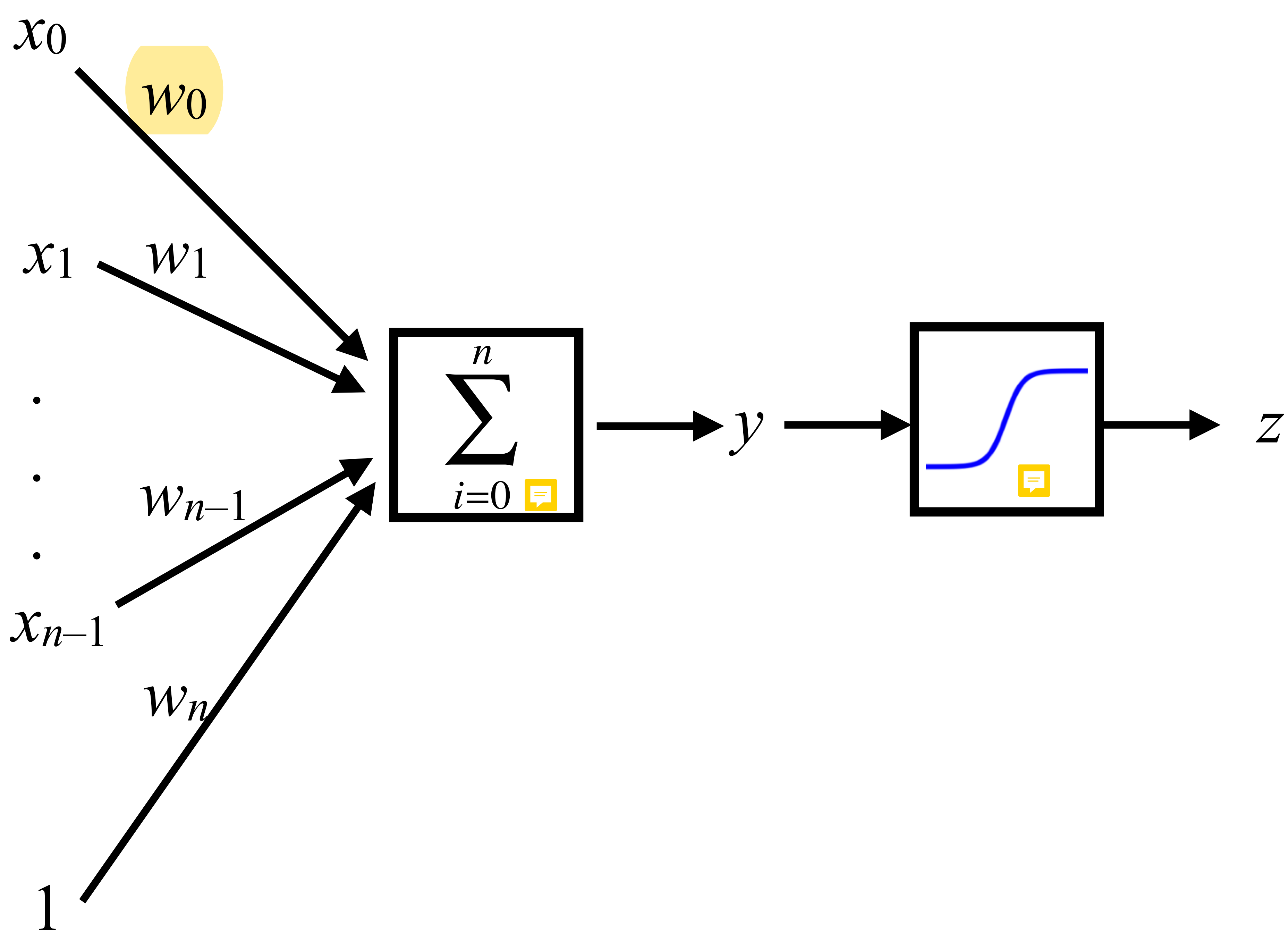
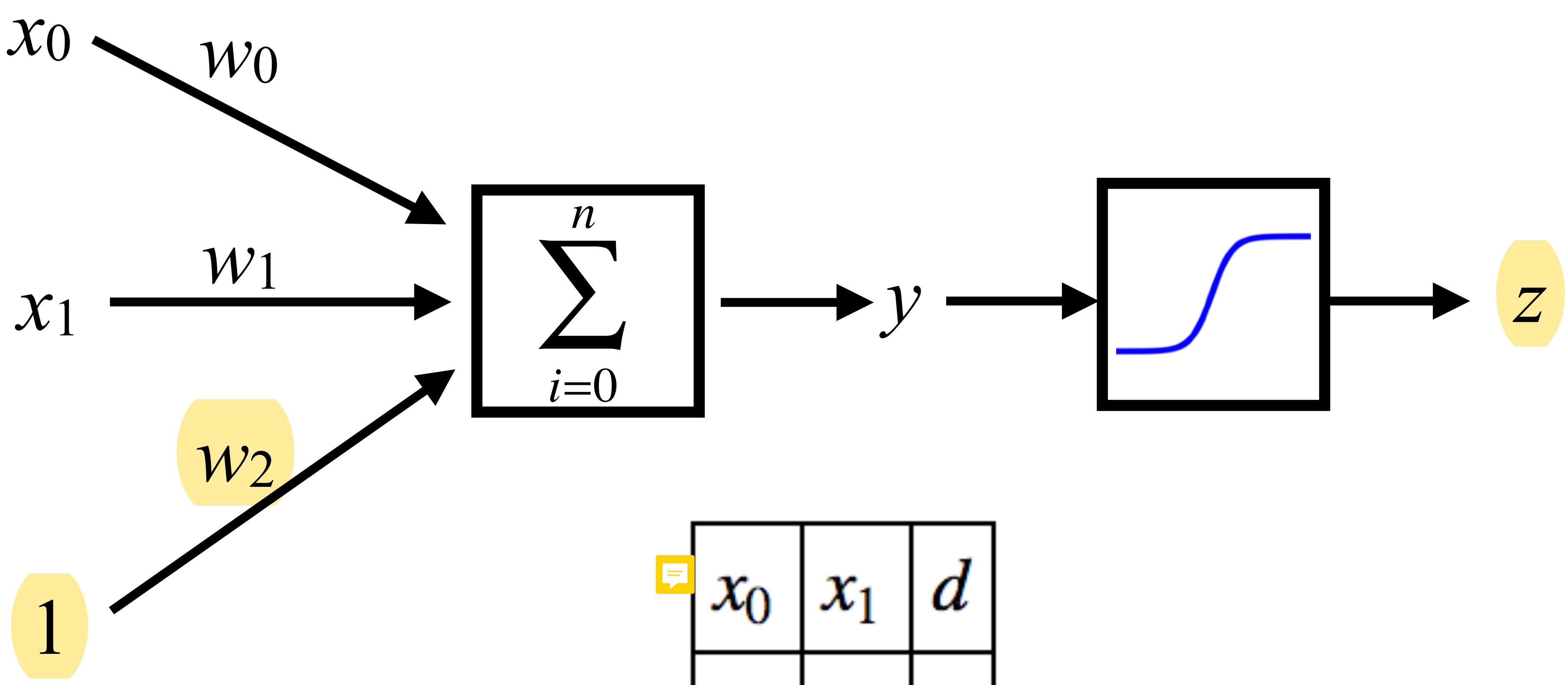
From the Perceptron to Neural Networks

The Perceptron — Single Layer Neural Network



The Perceptron — the NOR Gate Problem



x_0	x_1	d
0	0	1
0	1	0
1	0	0
1	1	0

The Utility of the Derivative and the Gradient

Suppose there is only one input *x*, and therefore one weight, *w*.

Then $$y = wx + b$$

And $$z = S(y)$$

Suppose the desired output is *d*. Then we can define a performance function (note the overall minus sign),

$$P = -\frac{1}{2}(d - z)^2$$

Derivative and Gradient

$$P = -\frac{1}{2}(d-z)^2 \qquad P \text{ is a function of } w,\ P(w).$$

$$\frac{dP}{dw} = \frac{dP}{dz}\frac{dz}{dy}\frac{dy}{dw} \qquad \frac{dP}{dz} = d - z \text{ (the “error”)} \qquad \frac{dy}{dw} = x \qquad \frac{dz}{dy} = \text{large}$$

$$\boxed{\frac{dP}{dw} \propto (d-z)x}$$

(for now, let’s think of the activation function as a very steep “step” function)

An analytical activation function:

$$z(y) = S(y) \qquad \frac{dz}{dy} = \frac{dS}{dy}$$

$$\frac{dz}{dy} = -(1+e^{-y})^{-2}e^{-y}(-1) = \frac{e^{-y}}{(1+e^{-y})^2} = \frac{1+e^{-y}-1}{(1+e^{-y})^2}$$

$$= \frac{1}{1+e^{-y}}\left[\frac{1+e^{-y}}{1+e^{-y}} - \frac{1}{1+e^{-y}}\right] = \boxed{z(1-z) = \frac{dz}{dy}}$$

What is unusual and turns out to be convenient: dz/dy can be expressed as the output!! (It’s like differentiating dy/dx and the answer turns out to depend on y rather than x!)

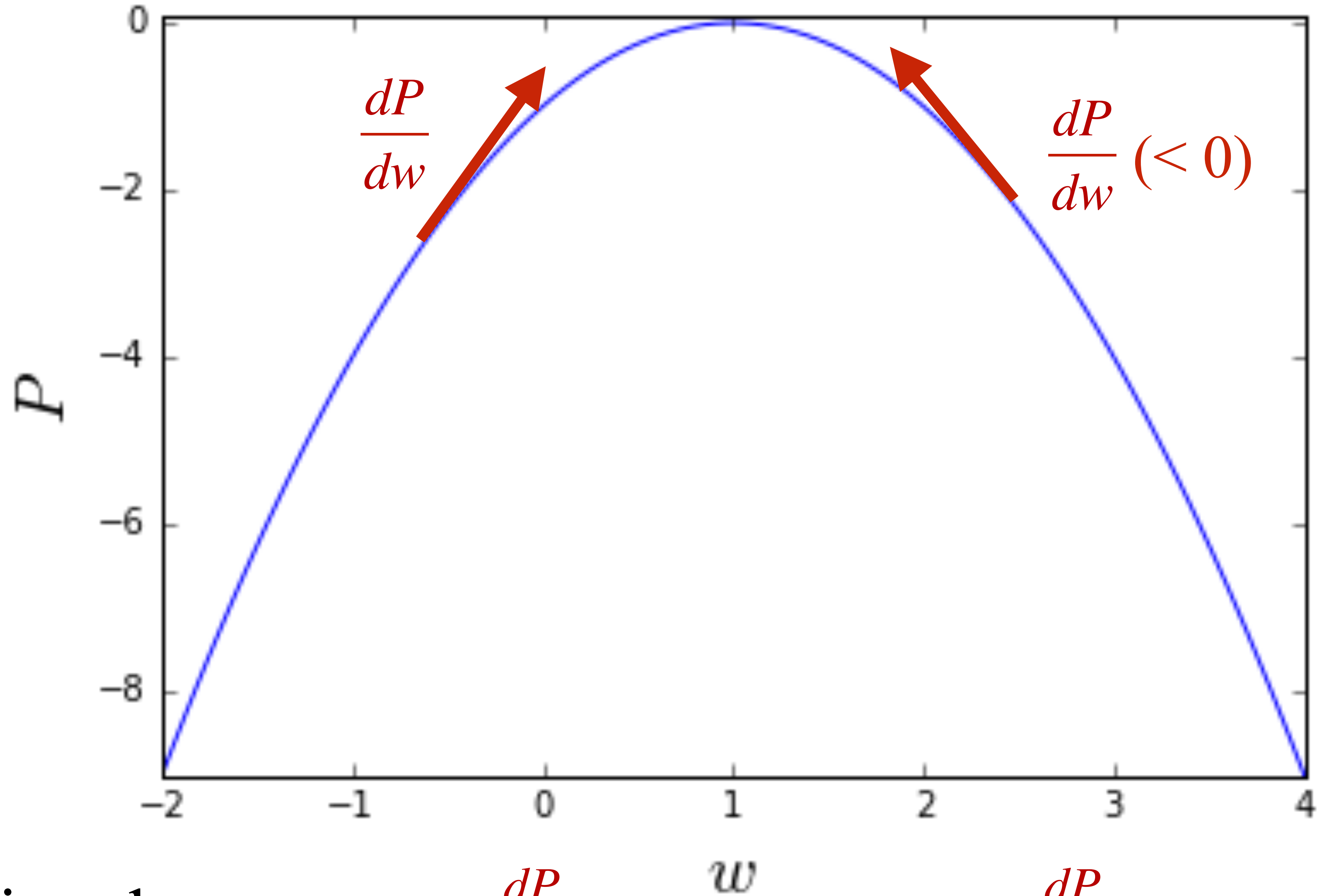
Derivative and Gradient

To find the optimal w:

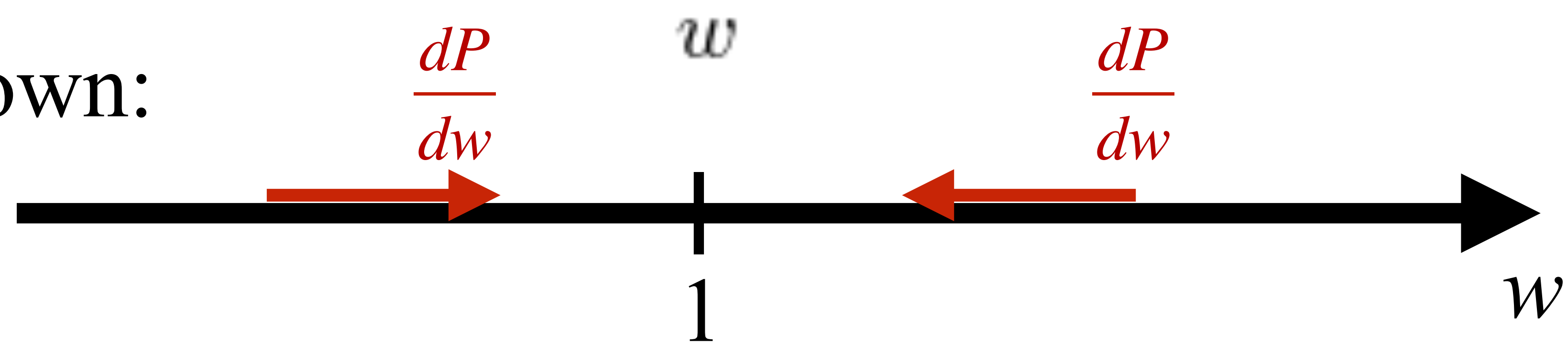
- Instead of setting dP/dw to 0, we want to use it to find out how to update w.

- Given the quadratic form of P, one can imagine the dependence of P on w looks something like this: (next slide)

- Given the quadratic form of P, one can imagine the dependence of P on w looks something like this



Looking down:

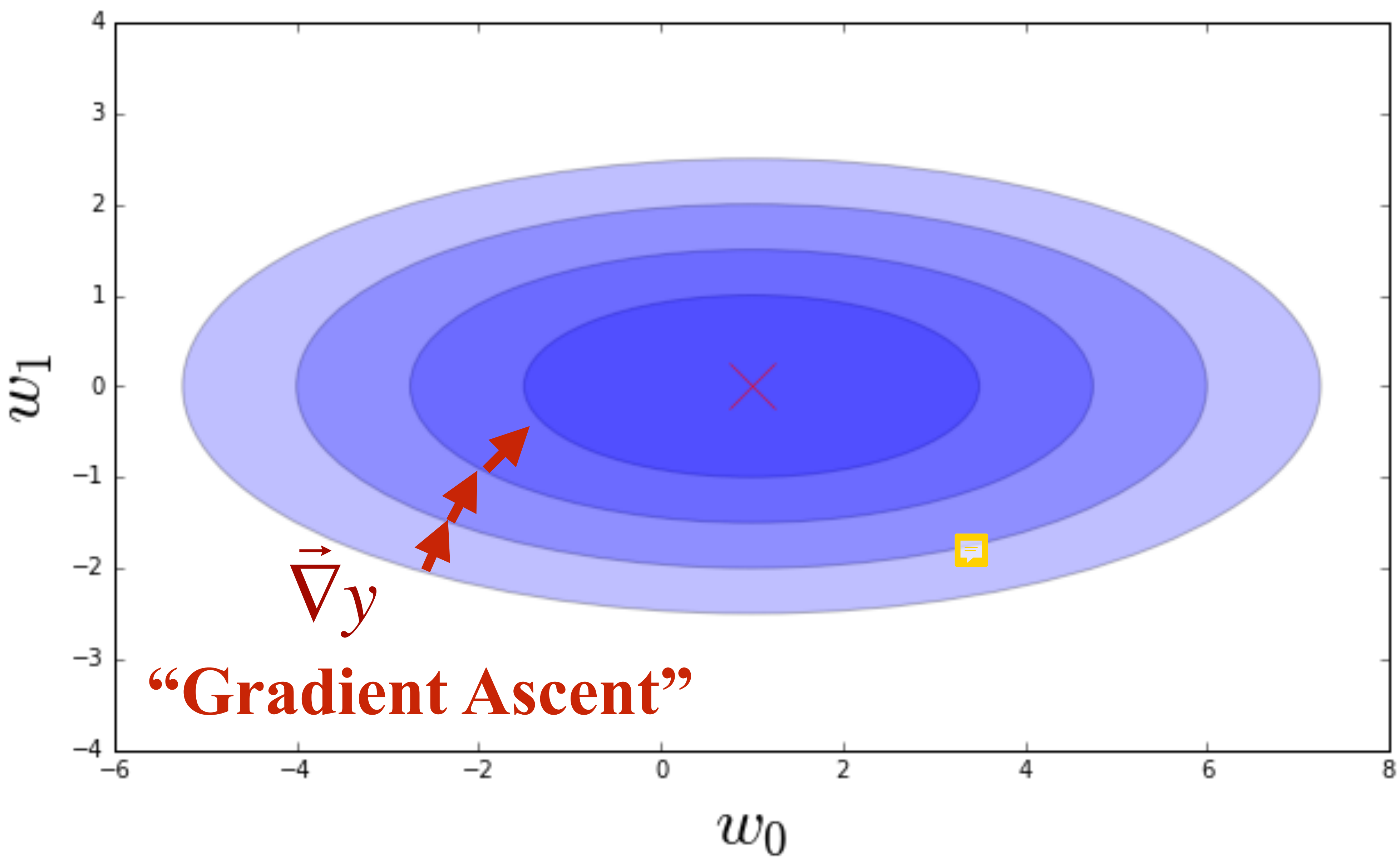


Thus dP/dw tells us the direction in which w should be adjusted. With learning rate set at α, $\Delta w = \alpha dP/dw$.

In 2D and higher Dimensions: The Gradient

$$\frac{dP}{dw} \rightarrow \vec{\nabla} P = \frac{\partial P}{\partial w_0}\hat{\imath} + \frac{\partial P}{\partial w_1}\hat{\jmath}$$

(From Calculus III and/or Methods.)



Adjustment in w_0-w_1 plane: direction set by the gradient of y, step size set by learning rate, α at each iteration.